BAYESIAN INFERENCE WITH HAMILTONIAN NORMALIZING FLOWS

Maximilian Gartz

Matr. No.: 972736

University of Osnabrück

March 7, 2021

Bachelor's Thesis

submitted to the Institute of Cognitive Science

Advisors

Prof. Dr. Michael Franke	Prof. Dr. Michael Gnewuch
Institute of Cognitive Science	Institute of Mathematics
University of Osnabrück	University of Osnabrück

Abstract

Bayesian inference is becoming more and more prevalent, not only for statistical models in fields like psychology, but in Machine Learning as well. It not only allows to incorporate domain knowledge for a given inference problem, but also to quantify uncertainties about the inferred model parameters and, in turn, the model's predictions. Computing the posterior distribution over model parameters, however, turns out to be intractable for most practical purposes.

Sophisticated approaches like Markov Chain Monte Carlo methods, in particular based on the state of the art Hamiltonian Monte Carlo kernel, have been applied to reliably generate samples from the posterior distribution even for complex models. However, there are still some drawbacks to those methods, when considering their application to models with exceedingly high numbers of parameters and large datasets, as is typical for machine learning problems.

In the past several years, variational Bayesian inference, which formulates the Bayesian inference problem as an optimization task, became popular due to recent advances in optimization techniques. More specifically, it allows to apply stochastic optimization and thus scale Bayesian inference to large datasets. The idea is to define a parameterized family of distributions and find the member minimizing a divergence measure to the true posterior distribution. One approach to define such a family is to make use of normalizing flows, which define a variational family as all the push-forwards of some base distribution along a parameterized diffeomorphism. Much research has focused on the question of which transformations allow for efficient computations and simultaneously expressive variational families.

Recently, efficient normalizing flows based on Hamiltonian dynamics were proposed in the context of modeling the underlying distribution of some dataset, i.e., density estimation. The focus of this thesis is to summarize the relevant mathematical foundations of this approach and adapt it for an application to variational Bayesian inference. Using two simple Bayesian models—a univariate Gaussian and a linear regression model—the goal is to show qualitatively that Hamiltonian normalizing flows can be used for variational Bayesian inference. An additional contribution is the development of a Python software package for Bayesian inference, based on TensorFlow Probability, which allow for a flexible definition of Bayesian models and choice of inference algorithms. In particular, it is used to implement the experiments discussed in this thesis.

Acknowledgements

First, I am grateful to my advisors, Prof. Michael Franke and Prof. Michael Gnewuch, for the opportunity to work on this subject for my Bachelor's thesis. I especially want to thank Dr. Marcin Wnuk from the Institute of Mathematics at the University of Osnabrück for his repeated feedback and helpful discussions regarding the theoretical aspects of this thesis. Polina Tsvilodub also deserves a special mention for providing invaluable feedback and mental support during the final weeks.

Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

Signature: ______________________________

City, date: Osnabrück, 08.03.2021

Contents

Abstract **III**

Acknowledgements **V**

Declaration of Authorship **VII**

Contents **XI**

1 Introduction **1**

1.1 Probability Distributions and Their Representations 1

1.1.1 Measurable Spaces and Measures 2

1.1.2 Probability Measures and Distributions 3

1.1.3 Probability Kernels and Conditional Distributions 3

1.1.4 Joint and Marginal Distributions 4

1.1.5 Representing Distributions with Densities 5

1.1.6 Representing Distributions with Samples 7

1.2 Statistical Models and Inference 8

1.2.1 Observation and Covariate Spaces 8

1.2.2 Data Generating Processes 9

1.2.3 The Model Configuration Space and Its Parameterizations 9

1.2.4 Parameter Inference - Fitting Models to Data 11

1.3 The Bayesian Inference Problem 11

1.3.1 Bayesian Models 12

1.3.2 Bayes' Rule 12

1.3.3 Bayesian Inference and Expectations 13

1.3.4 Approximating Expectations with Samples 15

1.3.5 Sampling Methods vs. Variational Methods 16

1.4 Sampling Methods - State of the Art 16

1.4.1 Naive Sampling Approaches 17

1.4.2 Markov Chain Monte Carlo Methods 18
1.4.3 Benefits and Drawbacks 21
1.5 Thesis Goals and Structure 21

2 Variational Inference 23
2.1 Variational Inference as Optimization 23
2.1.1 Divergences 23
2.1.2 The General Optimization Problem 25
2.1.3 Choosing a Variational Family 27
2.2 Variational Bayesian Inference 28
2.2.1 Variational Free Energy 29

3 Normalizing Flows 31
3.1 Normalizing Flows and Variational Families 31
3.1.1 Transformation of Samples 32
3.1.2 Transformation of Densities 32
3.1.3 Compositions of Normalizing Flows 33
3.2 Variational Bayesian Inference with Normalizing Flows 34
3.2.1 Variational Free Energy for Normalizing Flows 35
3.3 Practical Challenges 36
3.4 Continuous Normalizing Flows 37
3.4.1 Residual Flows 37
3.4.2 Continuous Time Limit of Residual Flow Compositions 37
3.4.3 Defining Continuous Normalizing Flows 38
3.4.4 Benefits and Drawbacks 40
3.5 Extensions to Continuous Normalizing Flows 40
3.5.1 Augmented Normalizing Flows 41
3.5.2 Volume Preserving Flows 45

4 Hamiltonian Normalizing Flows 47
4.1 The Hamiltonian Formalism 47
4.1.1 From Newtonian to Lagrangian Mechanics 47
4.1.2 Phase Space and the Hamiltonian 49
4.1.3 Hamilton's Equations and Hamiltonian Flows 50
4.2 Hamiltonian Flows as Normalizing Flows 52
4.2.1 Hamiltonian Normalizing Flows for Variational Inference 53

5 Evaluation **55**
5.1 Implementation 55
5.1.1 MCMC and VI Implementation 56
5.1.2 ADVI and Normalizing Flows Implementation 57
5.1.3 Hamiltonian Normalizing Flows Implementation 58
5.2 Experiments 59
5.2.1 HMC Configuration 60
5.2.2 VI Configurations 61
5.2.3 Univariate Gaussian Model 62
5.2.4 Linear Regression Model 63
5.3 Inference Results 63
5.3.1 Univariate Gaussian Model - Inference Results 63
5.3.2 Linear Regression Model - Inference Results 69
5.4 Discussion 75

6 Conclusion **77**

A Experiment Datasets **79**
A.1 Univariate Gaussian Model Datasets 79
A.2 Linear Regression Model Datasets 80

B HMC Inference Results **81**
B.1 Univariate Gaussian Model - HMC Results 81
B.2 Linear Regression Model - HMC Results 83

C Variational Inference Results **85**
C.1 Univariate Gaussian Model - VI Results 85
C.2 Linear Regression Model - VI Results 90

Bibliography **97**

Chapter 1

Introduction

The scientific endeavour in general is based on the idea of modeling the world and the processes within it mathematically on different levels of description. In some domains the uncertainties involved become dominant. They can be either fundamental or accounted for by methodological imprecision. This is where statistical models come into play and allow for a more precise quantification of those uncertainties.

After a brief review of the relevant mathematics and notation involved (Section 1.1), this introduction will discuss classical statistical models and the parameter inference problem (Section 1.2). This will lead up to the notion of Bayesian models and the fundamental problem of Bayesian inference (Section 1.3). Two different approaches to resolve this problem are introduced and it follows a high level description of the state of the art methods for Bayesian inference (Section 1.4). Finally, the structure and goals of the thesis are outlined to provide an overview (Section 1.5).

1.1 Probability Distributions and Their Representations

Before getting into statistical models, a few remarks on the mathematical foundations and notation are in order. Most of the theory regarding statistical models is best understood in the context of *measure theory*. The objective for this section is, in particular, to highlight the distinction between probability distributions and their representation via densities or samples. There will not be precise definitions provided for every mathematical object, since those are available in any standard reference. The focus lies on an intuitive review, which highlights the relationship between the relevant objects. An introduction to measure and probability theory is for example provided by Çinlar (2011) and Kallenberg (1997), which will be the main references for the theory summarized below. In some respects the notation will diverge slightly from the conventions used there, following Betancourt (2018b, 2018a) in his online case studies on probability theory. Folland (2009) is another useful reference for an introduction to measure theory from a real analysis perspective.

1.1.1 Measurable Spaces and Measures

The arena of statistical models, and more specifically probability distributions, are measurable spaces. A *measurable space* (X, Σ_X) is a non-empty set X equipped with a σ-algebra Σ_X. A *σ-algebra* is a subset of the power set $\mathcal{P}(X)$, which satisfies certain closure conditions w.r.t. to set operations. It comprises the so called *measurable sets* of X and thus defines a structure on this set. For this thesis it will later be useful to additionally assume at least an underlying topology on the sets of interest. This structure defines a notion of neighborhoods for the elements of a set, which, in particular, allows for the concept of continuity of maps between such spaces. A choice of topology $\mathcal{O}_X$ on a set X may induce a choice of σ-algebra $\mathcal{B}_X$, called a *Borel σ-algebra.* Accordingly, any time a measurable space is considered to have an underlying topological structure, where the measurable sets are induced by the topology, it will be denoted $(X, \mathcal{B}_X)$.

Measurable maps are maps $f : X \longrightarrow Y$ between measurable spaces $(X, \Sigma_X), (Y, \Sigma_Y)$, where the pre-image $A = f^{-1}(B)$ of any measurable set $B \in \Sigma_Y$ is again a measurable set $A \in \Sigma_X$. In other words, measurable maps preserve the measurable set structures and are thus called the *morphisms* of measurable spaces. Two measurable spaces that permit the existence of a bijection between them, which is measurable in both directions, are called *isomorphic.* That is, they are in a sense equivalent.

A *measure* on a measurable space (X, Σ_X) is a map $\mu_X : \Sigma_X \longrightarrow [0, \infty]$ from the measurable sets into the extended non-negative real numbers, which consistently assigns a generalized notion of mass to each measurable subset $A \in \Sigma_X$. Together, the measurable space and a choice of measure define what is called a *measure space* (X, Σ_X, μ_X). Measurable maps that also preserve the assigned mass are then the morphisms of measure spaces and are usually referred to as *measure preserving maps.* Two measure spaces that permit a bijection with this property in both directions are again called isomorphic. Such isomorphic measure spaces are equivalent from the point of view of measure theory, that is, they share all their relevant properties.

Measure spaces allow for the notion of *Lebesgue integration* of measurable functions $f : X \longrightarrow \mathbb{R}$ w.r.t. the measure μ_X on that space:

$$\mu_X f := \int_X f \; d\mu_X = \int_X f(x)\mu_X(dx). \tag{1.1}$$

For countable underlying sets X, this corresponds to a sum:

$$\mu_X f := \sum_{x \in X} f(x)\mu_X(\{x\}). \tag{1.2}$$

Intuitively, the Lebesgue integral $\mu_X f$ computes a weighted sum of the function values.

1.1.2 Probability Measures and Distributions

To get to probability distributions, one has to consider normalized finite measures on measurable spaces. Let π_X have those properties, i.e., $\pi_X(X) = 1$. Then π_X is referred to as a *probability measure* and the triple (X, Σ_X, π_X) is called a *probability space.* Note that a probability measure thus distributes a conserved quantity, the probability mass, onto the measurable sets. This is why they are also referred to as *probability distributions.* One can obviously also consider Lebesgue integrals w.r.t. probability distributions, since they are just a special case of measures. Such an integral is also referred to as an *expectation* of a measurable function f w.r.t. to the distribution π_X, which justifies the notation:

$$\mathbb{E}_{\pi_X}[f] = \int_X f(x)\pi_X(dx). \tag{1.3}$$

In this particular case, the Lebesgue integral denotes a weighted average instead of just any weighted sum, hence the term 'expectation'. The following will mostly restrict to considering probability distributions, although some things may be true for more general measures as well.

1.1.3 Probability Kernels and Conditional Distributions

Probability kernels are important objects, not only because they can represent regular conditional distributions and be used to construct joint distributions on product spaces, but also in the theory of *Markov Chain Monte Carlo* methods, which are the state of the art approach to Bayesian inference.

A map $K : X \times \Sigma_Y \longrightarrow [0, \infty]$, which defines measurable functions $K_B : X \longrightarrow [0, \infty]\ \ \forall B \in \Sigma_Y$ in the first and measures $K_x : \Sigma_Y \longrightarrow [0, \infty]\ \ \forall x \in X$ in the second argument, is called a *transition kernel* from (X, Σ_X) to (Y, Σ_Y). *Probability kernels* are those transition kernels, for which the measures K_x are probability measures, while a *Markov kernel* on (X, Σ_X) is a probability kernel from (X, Σ_X) to itself. It is useful to note that, by construction, a transition kernel K defines a map:

$$\kappa : X \longrightarrow \mathcal{M}(Y) \tag{1.4}$$

from the origin space into the set of all measures on the target space. That is, it defines a measure $K_x \in \mathcal{M}(Y)$ on the target space for each point $x \in X$ in the origin space.

In general, a transition kernel can push forward a measure μ_X on the origin space (X, Σ_X) to define a measure $\mu_X K$ on the target space (Y, Σ_Y) according to:

$$(\mu_X K)(B) = \int_X K_B(x)\mu_X(dx) = \int_X K(x, B)\mu_X(dx) \quad \forall B \in \Sigma_Y. \tag{1.5}$$

Moreover, it can pull back a measurable function $f : Y \longrightarrow \mathbb{R}^+$ on Y to define a function

$Kf : X \longrightarrow \mathbb{R}^+$ on X according to:

$$(Kf)(x) = \int_Y f(y)K_x(dy) = \int_Y f(y)K(x, dy) \quad \forall x \in X. \tag{1.6}$$

Without further explanation, the following chapters will assume that probability kernels can represent *regular conditional distributions* (see Çinlar (2011) or Kallenberg (1997) for technical details). In particular, kernels $K : X \times \Sigma_Y \longrightarrow [0, 1]$ and the corresponding conditional distributions $\pi_{Y|X} : \Sigma_Y \times X \longrightarrow [0, 1]$, $(B, x) \mapsto \pi_{Y|X}(B|x) = K(x, B)$ may be used interchangeably.

Accordingly, for probability kernels from (X, Σ_X) to (Y, Σ_Y), i.e., conditional distributions $\pi_{Y|X}(dy|x)$, the push-forward (1.5) and pull-back (1.6) can be written in terms of expectations. That is, the push-forward of a distribution π_X w.r.t. such a kernel yields:

$$\pi_Y(B) = \mathbb{E}_{\pi_X}[\pi_{Y|X}(B|x)] = \int_X \pi_{Y|X}(B|x)\pi_X(dx) \quad \forall B \in \Sigma_Y. \tag{1.7}$$

The pull-back of a measurable function then can be considered a *conditional expectation*:

$$\mathbb{E}_{\pi_{Y|X}}[f](x) = \int_Y f(y)\pi_{Y|X}(dy|x) \quad \forall x \in X. \tag{1.8}$$

1.1.4 Joint and Marginal Distributions

A distribution over a product space, e.g., $\pi_{X\times Y} : \Sigma_X \otimes \Sigma_Y \longrightarrow [0, 1]$ on $(X \times Y, \Sigma_X \otimes \Sigma_Y)$, is called a *joint distribution*. A *marginal distribution* is any push-forward of a joint distribution along a projection onto a component space. For example, let $\omega_X : X \times Y \longrightarrow X$ be a projection, then $\pi_X = \pi_{X\times Y} \circ \omega_X^{-1}$ is the *marginal distribution* over the component space (X, Σ_X), where ω_X^{-1} denotes the pre-image under the map ω_X. Some joint distributions are recovered as the product of their marginal distributions, i.e.:

$$\pi_{X\times Y}(A, B) := \pi_{X\times Y}(A \times B) = \pi_X(A)\pi_Y(B) \;\; \forall A \in \Sigma_X, B \in \Sigma_Y. \tag{1.9}$$

In this case, they are said to have *independent component distributions*. Product spaces, however, allow much richer joint distributions to be defined with inherent dependencies between the components, in which case the joint distribution is not recovered as a product of the marginals. Starting from a distribution π_X on the component space (X, Σ_X), one can construct such joint distributions $\pi_{X\times Y}$ using probability kernels $K : X \times \Sigma_Y \longrightarrow [0, 1]$ according to:

$$\pi_{X\times Y}(A, B) = \int_A K(x, B)\pi_X(dx) \quad \forall A \in \Sigma_X, B \in \Sigma_Y. \tag{1.10}$$

Equivalently, this can be written in terms of a conditional distribution:

$$\pi_{X\times Y}(A,B) = \int_A \pi_{Y|X}(B|x)\pi_X(dx), \tag{1.11}$$

or in a short differential notation based on Çinlar (2011):

$$\pi_{X\times Y}(dx,dy) = \pi_{Y|X}(dy|x)\pi_X(dx). \tag{1.12}$$

The inverse to the problem of constructing joint distributions is called disintegration. A *disintegration* decomposes a joint distribution $\pi_{X\times Y}$ w.r.t. a projection map, e.g. ω_X, into the corresponding marginal distribution π_X and a kernel or conditional distribution $\pi_{Y|X}$. For any given projection map, this decomposition is almost surely unique.

If the disintegration $\pi_{X\times Y}(dx,dy) = \pi_{Y|X}(dy|x)\pi_X(dx)$ is known, it is trivial to construct the remaining marginal distribution π_Y as a push-forward of π_X along the probability kernel:

$$\pi_Y(B) = \mathbb{E}_{\pi_X}\left[\pi_{Y|X}(B|x)\right] = \int_X \pi_{Y|X}(B|x)\pi_X(dx). \tag{1.13}$$

Note that this requires integration over the complete component space X.

1.1.5 Representing Distributions with Densities

In practice distributions are not particularly useful when it comes to computations, since they are defined only on the measurable subsets of the underlying space and are of an highly abstract nature. This is where different representations of distributions come into play. In particular, probability distributions may be represented uniquely by densities.

More generally, consider any choice of measure μ_X in relationship to some *base measure* ν on a given measurable space (X, Σ_X). If μ_X is *absolutely continuous* w.r.t. ν, it can be represented by a ν-measurable function $p : X \longrightarrow \mathbb{R}^+$ according to the *Radon-Nikodym theorem* (Çinlar, 2011):

$$\mu_X(A) = \int_A \mu_X(dx) = \int_A p(x)\nu(dx) \quad \forall A \in \Sigma_X. \tag{1.14}$$

This can also be expressed in the aforementioned short differential notation as:

$$\mu_X(dx) = p(x)\nu(dx). \tag{1.15}$$

Since the function p is unique up to ν-negligible sets, this justifies calling it the *Radon-Nikodym derivative* or the *density* of μ_X w.r.t. ν denoted by:

$$p(x) := \frac{\mu_X(dx)}{\nu(dx)}(x) = \frac{d\mu_X}{d\nu}(x) \quad \forall x \in X. \tag{1.16}$$

If two measures are absolutely continuous w.r.t. each other, they are called *equivalent* and the Radon-Nikodym derivative exists in both directions.

Note that there are some natural choices for base measures on uncountably infinite and countable underlying sets. Let $(X, \mathcal{B}_X)$ be a countable set equipped with the Borel σ-algebra induced by the discrete topology $\mathcal{O}_X = \mathcal{P}(X)$, such that it may be referred to as a *discrete space*. A natural choice of base measure on such a discrete space is the *counting measure* χ, which assigns to each measurable subset $A \in \mathcal{B}_X$ its magnitude $\chi(A) = |A|$. For a measurable space $(X, \mathcal{B}_X)$ with an uncountably infinite underlying set $X = \mathbb{R}^n$, for some $n \in \mathbb{N}$, with the Borel σ-algebra induced by the standard topology on $\mathbb{R}^n$, a natural choice is the so called *Lebesgue measure* λ^n. It is the unique translation invariant measure which assigns $\lambda^n(Q) = 1$ to any unit hyper-cube $Q \subseteq \mathbb{R}^n$ and thus corresponds to the usual notion of uniform mass distribution. For a more technical discussion on the construction of the Lebesgue measure refer to Folland (2009).

An important aspect of the density representation is that it recovers the underlying measure. That is, it consistently breaks down the measure's assignment of generalized mass $\mu_X(A)$ to measurable sets $A \in \Sigma_X$ into an assignment of density $p(x)$ to the individual elements $x \in X$. This allows to recover the mass of any measurable set by integrating over the density of the elements of that set. Choosing the abovementioned natural measures as base measures provides useful examples:

1. for discrete spaces $(X, \mathcal{B}_X)$ with the counting measure χ as a base measure:

$$\mu_X(A) = \sum_{x \in A} \frac{d\mu_X}{d\chi}(x)\chi(\{x\}) = \sum_{x \in A} p(x) \quad \forall A \in \mathcal{B}_X, \tag{1.17}$$

2. for uncountably infinite spaces $(X = \mathbb{R}^n, \mathcal{B}_X)$ with the Lebesgue measure λ^n as a base measure:

$$\mu_X(A) = \int_A \mu_X(dx) = \int_A \frac{d\mu_X}{d\lambda^n}\ \lambda^n(dx) = \int_A p(x)\ \lambda^n(dx) = \int_A p(x)\ dx \quad \forall A \in \mathcal{B}_X. \tag{1.18}$$

In particular, note that all of this applies to probability distributions and yields the typical notions of probability density and mass functions, where mass functions refer to densities on discrete spaces. This is also true for joint and conditional distributions, yielding joint and conditional probability density or mass functions:

$$\begin{aligned} \pi_X(dx) &= p(x)\lambda(dx) \\ \pi_{Y|X}(dy|x) &= p(y|x)\mu(dy) \\ \pi_{X \times Y}(dx, dy) &= p(x, y)\nu(dx, dy), \end{aligned} \tag{1.19}$$

where λ, μ and ν are some base measures on the corresponding measurable spaces. The following chapters will restrict to uncountably infinite underlying spaces, in particular, to those that are equal to some $\mathbb{R}^n$, if not specified otherwise. Any considered measure or distribution is assumed to be equivalent to the Lebesgue measure, such that they have valid density functions on $\mathbb{R}^n$ in relation to the usual notion of uniform mass distribution. It is useful to assume equivalence to the base measure, to ensure that the Radon-Nikodym derivative between any two measures exists in both directions.

1.1.6 Representing Distributions with Samples

Given the representation of distributions π_X as densities $p(x)$, w.r.t. the Lebesgue measure λ, the computation of expectations of measurable functions $f : X \longrightarrow \mathbb{R}$ can be done in practice according to:

$$\mathbb{E}_{\pi_X}[f] = \int_X f(x)\pi_X(dx) = \int_X f(x)p(x)\lambda(dx) = \int_X f(x)p(x)dx. \tag{1.20}$$

Note, however, that these integrals do not necessarily have a closed form solution and thus in many cases need to be approximated via numerical integration. For high dimensional uncountably infinite spaces $(X = \mathbb{R}^n, \mathcal{B}_X)$, those integrals cannot be efficiently computed with standard quadrature rules. Moreover, in practice usually most of the probability mass is associated with only a small, not necessarily connected region of the total space. Accordingly, most function evaluations do not contribute much to the expectation, if the evaluation points lie outside of this distinguished region, making them rather inefficient (Betancourt, 2017a).

A reasonable alternative to the representation of a distribution π_X via a density function, therefore, seems to be a procedure that is capable of generating a finite sequence of points in X, which allows for accurate estimates of expectations w.r.t. this distribution. Let $D_N = (x_1, x_2, ..., x_N)$ be a sequence of points in X and $f : X \longrightarrow \mathbb{R}$ any measurable function. Then the term:

$$\hat{f}(D_N) = \frac{1}{N}\sum_{n=1}^{N} f(x_n) \tag{1.21}$$

is called the *empirical expectation* of f. Any process that is capable of generating sequences D_N of points, for which this empirical expectation is a consistent estimator for the expectation of any measurable function f w.r.t. distribution π_X, i.e.:

$$\lim_{N\to\infty} \hat{f}(D_N) = \mathbb{E}_{\pi_X}[f], \tag{1.22}$$

will be called a *sampling procedure* representing the distribution π_X. The generated finite sequences are then referred to as *samples* of π_X, which are called *exact*, if every point in the sequence was generated independently of the others. Accordingly, any sub-sequence of exact samples is also an exact sample, as is any individual point. If a point $x \in X$ is a sample generated from a sampling procedure representing π_X, then this will be denoted by $x \sim \pi_X$ (Betancourt, 2017b, 2018b).

There exist elaborate algorithms, which are, at least for all practical purposes, capable of generating such exact samples for standard distributions (Devroye, 1986), for example those that can be represented by a uniform or Gaussian density w.r.t. the Lebesgue measure. Sampling from joint distributions of such standard distributions can be realized using the so called *ancestral sampling* approach (Bishop, 2006).

1.2 Statistical Models and Inference

This section briefly introduces statistical models and the problem of parameter inference. In particular, it will specify the kinds of models considered in the following chapters. Useful references for the theory of statistics are Lehmann and Casella (1998) and Keener (2010). McCullagh (2002) also gives a good account of what the generally accepted definition of a statistical model is, while pointing out some resulting problems and providing a treatment on the level of category theory. This section will make use of all the above references, but mostly follows Betancourt (2019) in his case study on probabilistic modeling and statistical inference.

1.2.1 Observation and Covariate Spaces

To be as general as possible, consider some *domain* or *population* M. Usually, one is interested in some particular abstract characteristics of the elements of this domain. To quantify them, those characteristics are operationalized, which allows to measure them in experiments. This choice of operationalization induces a mapping $y : M \longrightarrow Y$ from the elements of the domain into a so called *observation* or *target space* Y. Similarly, it induces a map $x : M \longrightarrow X$ from the domain into an additional space called *covariate* or *feature space* X (McCullagh, 2002).

Any statistical experiment now typically assumes an independently drawn set of experimental units $U \subset M$ from this domain. The idea then is that any experimental unit $u \in U$ can be mapped onto an element $y(u) \in Y$, as well as an element $x(u) \in X$. This allows to associate each one of them with a set of values, quantifying the covariates and targets of interest (McCullagh, 2002).

Consider, for example, a psychological experiment, where the influence of age on intelligence in men is investigated. In this case, the domain M is the set of all men. Now assume age is measured in years, while intelligence is operationalized and measured as the IQ of any given man. This choice defines the map $y : M \longrightarrow Y$ from the set of all men into the possible IQ values, as well as the map $x : M \longrightarrow X$ into the set of possible age values. Let the subset $U \subset M$ be the subjects of the experiment, i.e. the experimental units, which are assumed to be randomly selected men. The above maps then allow to assign to each subject $u \in U$ the corresponding age $x(u)$ and IQ value $y(u)$.

The theory of this thesis will restrict to the case of so called *generative models*, that is, models for which the covariate space is empty. Relating this to the above example, the goal is to model only the underlying distribution of IQ in men, without considering additional explanatory factors like age. This simplifies the notation significantly. Any algorithm discussed in later chapters, however, can be applied to so called *regression models* as well, where the covariate space is non-empty, by considering an additional conditioning on the covariates. This will be verified with the linear regression example in Chapter 5.

Finally, the following chapters assume the map $y : M \longrightarrow Y$ to be fixed by some purposefully chosen operationalization, such that Y is a Euclidean space $\mathbb{R}^n$. A dataset of N independent experimental units $U = \{u_i\}_{i=1}^N$ obtained from M can thus, for the purposes of the theory presented in this thesis, be represented by a tuple $D = (y(u_i))_{i=1}^N \in Y^N$ consisting only of the corresponding elements in a fixed observation space.

1.2.2 Data Generating Processes

To be mathematically more precise, the observation or target space is assumed to be a measurable space $(Y, \mathcal{B}_Y)$. This allows to establish probability distributions π_Y on it. In the context of statistical modeling any such distribution will sometimes be referred to as a *data generating process.* The set of all possible data generating processes then will be denoted $\Pi(Y)$.

At the heart of classical statistical modeling lies the assumption of the existence of a true underlying data generating process $\pi_Y^\dagger \in \Pi(Y)$, such that the elements of the dataset $D \in Y^N$ are assumed to be exact samples of $\pi_Y^\dagger$. Alternatively, D itself can be considered a single sample of the corresponding unique joint distribution $\pi_{Y^N}^\dagger = \bigotimes_{i=1}^N \pi_Y^\dagger$. Finding $\pi_Y^\dagger \in \Pi(Y)$, however, is clearly intractable for all practical cases. More importantly, there is not even an obvious objective w.r.t. which one could judge which $\pi_Y \in \Pi(Y)$ is in fact the true data generating process (Betancourt, 2019).

1.2.3 The Model Configuration Space and Its Parameterizations

A possible approach is to restrict the search space of all generating processes $\pi(Y)$ to a tractable subset. In general, this subset will be called the *model configuration space* and denoted by $\mathcal{S} \subseteq \Pi(Y)$. Space in this context is used in a colloquial sense, mathematically it is just a set with no additional structures. Such a subset $\mathcal{S}$ also defines a *statistical model* in the classical sense (Betancourt, 2019; McCullagh, 2002).

Note, however, that there is no canonical way to choose $\mathcal{S}$. In particular, since $\Pi(Y)$ is just the set of probability measures on $(Y, \mathcal{B}_Y)$, the elements $\pi_Y \in \mathcal{S}$ are only abstract mathematical objects, which are defined by exactly how much probability mass they assign to each measurable set.

A map $P : \Theta \longrightarrow \Pi(Y)$ from some parameter set or *parameter space* $\Theta \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$, into the set of all possible distributions on Y, defines a so called *parameterized model* simply as the image of the parameter space $\mathcal{S} := P(\Theta)$ (McCullagh, 2002; Lehmann & Casella, 1998; Keener, 2010). In particular, any family of density functions $p(y; \theta)$ with $\theta \in \Theta$ represents a set of probability measures on Y and thus induces a model configuration space choice.

From another perspective, given a fixed choice of model configuration space $\mathcal{S} \subseteq \Pi(Y)$, a surjective map $P : \Theta \longrightarrow \mathcal{S}$ is called a *parameterization* of $\mathcal{S}$. In case the map is injective as well, and thus a bijection, the parameterization P is called *identifyable.* That is, every parameter tuple $\theta \in \Theta$ can be uniquely identified with an element $\pi_Y \in \mathcal{S}$. Intuitively, a parameterization allows to assign parameter values to each model configuration (Betancourt, 2019; McCullagh, 2002).

In general, there may exist many possibly equivalent parameterizations, i.e., distinct surjective maps $P_i : \Theta_i \longrightarrow \mathcal{S}$ from different parameter spaces $\Theta_i \ \forall i \in I$ onto the model configuration space $\mathcal{S}$, where $I \subseteq \mathbb{N}$ is some index set. They may be equivalent in the sense that they are isomorphic as sets, that is, there exists at least one bijective map between them. If only identifyable parameterizations are considered, all parameter spaces are isomorphic to each other, since they are all isomorphic to the model configuration space $\mathcal{S}$. Set isomorphisms $g : \Theta_j \longrightarrow \Theta_k$ between parameter spaces Θ_j, Θ_k, for $j, k \in I$, associated with different parameterization maps P_j, P_k are called *reparameterizations.* The following commutative diagram visualizes the relationship between different parameterizations as described above.

$$\begin{tikzcd}
 & \mathcal{S} & \\
g^{-1}(\Theta_k) = \Theta_j \arrow[ur, "P_j \ = \ P_k \circ g"] \arrow[rr, "g"] & & \Theta_k = g(\Theta_j) \arrow[ul, "P_k \ = \ P_j \circ g^{-1}"']
\end{tikzcd}$$

Figure 1.1: Parameterizations P_j, P_k of the model configuration space $\mathcal{S}$.

For simplicity, the theoretic discussions of this thesis will restrict to parameterizations with parameter spaces $\Theta = \mathbb{R}^d$ for some $d \in \mathbb{N}$. This will later on allow to assume much more structure on the spaces involved without having to discuss differentiable manifolds and even more advanced concepts. Under this assumption, it is also clear what is meant by the dimension of a parameter space and it is possible to consider differentiable maps between different parameter spaces, which will become relevant in the context of normalizing flows. For classical statistical models it is, however, sufficient to consider parameter spaces to be just sets.

The univariate Gaussian family of density functions provides a simple example for a map defining a parameterized model with two parameters, $\theta = (\mu, \sigma) \in \Theta_0 = \mathbb{R} \times \mathbb{R}^+ \subset \mathbb{R}^2$. This corresponds to restricting the set of all probability distributions to the subset that permits a Gaussian density representation on $(Y, \mathcal{B}_Y)$ w.r.t. the Lebesgue measure. It is easy to see that there exists a reparameterization to $\Theta_1 = \mathbb{R}^2$ via a logarithmic transformation of the standard deviation parameter σ. This particular parameterization would then be valid according to the restrictions of this thesis mentioned above. Note, that the reparameterization does only effect the functional form of the density, but not the model configuration space $\mathcal{S}$, it simply results in a different 'labeling' of the elements of $\mathcal{S}$.

1.2.4 Parameter Inference - Fitting Models to Data

Having chosen a model configuration space $\mathcal{S}$, based on a family of probability density functions $p(y;\theta)$ in some parameterization $P : \Theta \longrightarrow \mathcal{S}$, the problem of finding the true underlying data generating process $\pi_Y^\dagger$ is far from solved. The restriction $\mathcal{S}$ might not even contain $\pi_Y^\dagger$. In practice, this is indeed the rule rather than the exception (Betancourt, 2019).

Moreover, a finite dataset cannot uniquely determine the true underlying distribution because of the uncertainties involved in drawing samples from the underlying population, such that the problem of finding $\pi_Y^\dagger$ is in general ill-posed. Eventually, one is left with the problem of at least finding a member $\pi_Y^* \in \mathcal{S}$, represented by $p(y;\theta^*)$ w.r.t. the Lebesgue measure, which is closest to $\pi_Y^\dagger$ in terms of being something like a best guess. A useful heuristic here is to choose the member, for which the density of the product distribution maximizes the probability density of the dataset D, i.e.:

$$\theta^* := \underset{\theta\in\Theta}{\arg\max}\; p(D;\theta) = \underset{\theta\in\Theta}{\arg\max} \prod_{i=1}^{N} p(y_i;\theta). \tag{1.23}$$

This is equivalent to maximizing the monotone logarithmic transformation, i.e., the log probability density of the data, which in practice also provides better numerical stability.

$$\theta^* := \underset{\theta\in\Theta}{\arg\max}\; \log p(D;\theta) = \underset{\theta\in\Theta}{\arg\max} \sum_{i=1}^{N} \log p(y_i;\theta). \tag{1.24}$$

The parameters θ^* are called the *maximum likelihood* solution to the parameter inference problem. They constitute a point estimate for the model parameters that maximizes the probability density of the data (Keener, 2010).

For some special cases it is possible to solve this problem analytically, but typically gradient based optimization approaches are used in practice. The optimization problem is in general non-convex, such that there is no guarantee to find an optimal solution to this problem, even if it exists.

1.3 The Bayesian Inference Problem

The Bayesian approach to statistical modeling is based on the realization that fundamentally all distributions over the observation space may have been the true generating process, but based on prior knowledge and the observed data this is more likely for some than for others. It embraces the fact, that the inference problem has no unique solution, in the sense of a particular model configuration being the true one. According to this perspective, the classical maximum likelihood heuristic, although useful in practice, only makes sense if the inference problem is uniquely solvable, i.e., in the case of infinite available data. This is exactly the regime where the classical and the Bayesian approach agree (van der Vaart, 1998; Betancourt, 2019).

1.3.1 Bayesian Models

To define Bayesian models formally, consider the observation space $(Y, \mathcal{B}_Y)$ and some parameter space Θ. This time, however, assume an underlying topological structure on Θ and the induced Borel σ-algebra $\mathcal{B}_\Theta$, such that the parameter space is a measurable space $(\Theta, \mathcal{B}_\Theta)$.

A map $P : \Theta \longrightarrow \Pi(Y)$ now obviously defines some transition kernel according to (1.4) and thus corresponds to a choice of conditional distribution $\pi_{Y|\Theta}$, which may be defined via a conditional density function $p(y|\theta)$ w.r.t. the Lebesgue measure on Y. It not only determines a model configuration space $\mathcal{S} := P(\Theta)$, but also a σ-algebra $\Sigma_\mathcal{S} := \{A \in \mathcal{P}(\mathcal{S}) :\ P^{-1}(A) \in \mathcal{B}_\Theta\}$, such that $\mathcal{S}$ can be considered a measurable space $(\mathcal{S}, \Sigma_\mathcal{S})$ as well.

A Bayesian model is then constructed by introducing a so called *prior distribution* $\pi_\Theta \in \Pi(\Theta)$ over the parameter space, via a selection of a prior density function $p(\theta)$. This distribution has a corresponding push-forward $\pi_\mathcal{S} = \pi_\Theta \circ P^{-1} \in \Pi(\mathcal{S})$ on the model configuration space, which will be referred to as prior distribution as well. It encapsulates a priori assumptions about how likely the different model configurations are. A choice of prior distribution π_Θ finally induces a joint distribution:

$$\pi_{Y\times\Theta}(dy, d\theta) = \pi_\Theta(d\theta)\pi_{Y|\Theta}(dy|\theta) \tag{1.25}$$

on the product space $(Y \times \Theta, \mathcal{B}_Y \otimes \mathcal{B}_\Theta)$. This joint distribution, or more generally any joint distribution over this product space, defines a *Bayesian model* in the chosen parameterization $P : \Theta \longrightarrow \mathcal{S} \subseteq \Pi(Y)$.

When discussing Bayesian inference, everything will play out in the frame of some parameterization. It is useful, however, to keep in mind that distributions over a parameter space have a more abstract counterpart, that is their push-forward distribution on $(\mathcal{S}, \Sigma_\mathcal{S})$ along the respective parameterization. Note that, by construction, the initial parameterization map, in which the model is defined, is measurable. For any other parameterization, and consequently for reparameterizations, this will be an additional requirement.

Very simply put, the goal of Bayesian inference is to find the disintegration of the joint distribution defining the model w.r.t. the projection onto the observation space Y, which is not a trivial problem. Bayes' rule, however, provides a recipe to solve it.

1.3.2 Bayes' Rule

The widely known *Bayes' rule* is based on the observation that there exist two canonical disintegrations of any Bayesian model, i.e., joint distribution $\pi_{Y\times\Theta}$, according to the projections onto the component spaces Y and Θ. In the short differential notation used before, the disintegrations can be written as:

$$\begin{aligned} \pi_{Y\times\Theta}(dy, d\theta) &= \pi_\Theta(d\theta)\pi_{Y|\Theta}(dy|\theta) \\ \pi_{Y\times\Theta}(dy, d\theta) &= \pi_Y(dy)\pi_{\Theta|Y}(d\theta|y). \end{aligned} \tag{1.26}$$

While the first disintegration is known by construction of the model, the second one is unknown. Neither the marginal distribution π_Y over the observation space, nor the so called *posterior distribution* $\pi_{\Theta|Y}$ over parameter space are available. The equations (1.26), however, imply the identity:

$$\pi_Y(dy)\pi_{\Theta|Y}(d\theta|y) = \pi_\Theta(d\theta)\pi_{Y|\Theta}(dy|\theta) \quad \Longrightarrow \quad \frac{\pi_{\Theta|Y}(d\theta|y)}{\pi_\Theta(d\theta)} = \frac{\pi_{Y|\Theta}(dy|\theta)}{\pi_Y(dy)} =: f(y,\theta). \tag{1.27}$$

This expresses the equality of the Radon-Nikodym derivatives of the conditional distributions w.r.t. the marginal distributions on each component space. The application of Bayes' rule can be thought of as a transformation of the prior distribution π_Θ into the posterior distribution $\pi_{\Theta|Y}$. The posterior distribution has again a corresponding push-forward $\pi_{\mathcal{S}|Y} = \pi_{\Theta|Y} \circ P^{-1}$ along the parameterization, which captures the data-informed a posteriori beliefs about how likely the different model configurations are. The transformation interpretation becomes apparent, when considering the following simple rearrangement of (1.27):

$$\pi_{\Theta|Y}(d\theta|y) = \frac{\pi_{Y|\Theta}(dy|\theta)}{\pi_Y(dy)}\pi_\Theta(d\theta) = f(y,\theta)\pi_\Theta(d\theta). \tag{1.28}$$

The Radon-Nikodym derivative f intuitively corrects the prior into the posterior distribution based on the observed data $y \in Y$. The function is, however, only partially known because of $\pi_Y(dy)$ and is thus also not directly accessible.

Given some base measures on $(Y, \mathcal{B}_Y)$ and $(\Theta, \mathcal{B}_\Theta)$, the equality of the Radon-Nikodym derivatives in (1.27) implies the same for the corresponding probability density ratios:

$$\frac{p(\theta|y)}{p(\theta)} = \frac{p(y|\theta)}{p(y)}. \tag{1.29}$$

The reason this is true, is that the base measure obviously cancels out when considering ratios of densities. Bayes' rule in terms of densities is then usually put in the form (Gelman et al., 2013):

$$p(\theta|y) = \frac{p(y|\theta)}{p(y)}p(\theta). \tag{1.30}$$

1.3.3 Bayesian Inference and Expectations

In Bayesian inference, the idea is to apply Bayes' rule in order to infer the posterior distribution over model configuration space. That is, the goal is to update the prior distribution, informed by the observed dataset. In general the benefit of Bayesian inference is that the posterior distribution provides more information than a simple point estimate for the model parameters obtained from the maximum likelihood or the related maximum a posteriori approach. It allows for the quantification of uncertainties about the parameter values, for example, summarized via so called *credible regions* or, more specifically, *highest density regions* (Gelman et al., 2013).

Just as in parameter inference for classical statistical models, Bayes' rule generalizes straightforwardly from single observations $y \in Y$ to complete datasets $D \in Y^N$. With the typical assumption of all samples being independently drawn from the same underlying distribution, any Bayesian model on $(Y \times \Theta, \mathcal{B}_Y \otimes \mathcal{B}_\Theta)$ induces a unique Bayesian model on $(Y^N \times \Theta, \mathcal{B}_{Y^N} \otimes \mathcal{B}_\Theta)$ according to:

$$\pi_{Y^N \times \Theta}(dD, d\theta) = \pi_\Theta(d\theta)\pi_{Y^N|\Theta}(dD|\theta) = \pi_\Theta(d\theta) \bigotimes_{i=1}^{N} \pi_{Y|\Theta}(dy_i|\theta). \tag{1.31}$$

Bayes' rule then takes the form:

$$\pi_{\Theta|Y^N}(d\theta|D) = \frac{\pi_{Y^N|\Theta}(dD|\theta)}{\pi_{Y^N}(dD)}\pi_\Theta(d\theta) = \frac{p(D|\theta)}{p(D)}\pi_\Theta(d\theta) \tag{1.32}$$

Noting that the dataset D is fixed for any given Bayesian inference problem it is obvious that the so called *model evidence* $p(D)$, i.e., the probability density of the data under the defined model, is just a constant. Moreover, the conditional density of the data $p(D|\theta)$ can be considered only a function of the parameters, which is sometimes emphasized by calling it the *likelihood function* denoted by $\mathcal{L}_D(\theta) := p(D|\theta)$. Those observations allow to consider a Bayesian model, in the context of a particular inference problem, not to be a joint distribution over observation and parameter space, but an unnormalized finite measure $\mu_\Theta^D(d\theta) = \mathcal{L}_D(\theta)\pi_\Theta(d\theta)$ over parameter space. In fact, this measure is equal to the posterior distribution up to normalization with constant $Z := p(D)$. This allows to formulate Bayes' rule as:

$$\pi_{\Theta|Y^N}(d\theta|D) = \frac{1}{p(D)}\mathcal{L}_D(\theta)\pi_\Theta(d\theta) = \frac{1}{Z}\mu_\Theta^D(d\theta). \tag{1.33}$$

The normalizing constant, i.e., the model evidence, is however not easily accessible, since it requires integration over the complete parameter space:

$$Z = \int_\Theta \mu_\Theta^D(d\theta) = \int_\Theta \mathcal{L}_D(\theta)\pi_\Theta(d\theta) = \mathbb{E}_{\pi_\Theta}\left[\mathcal{L}_D(\theta)\right]. \tag{1.34}$$

This integral will be intractable in most cases, especially for high dimensional parameter spaces. In particular, this indicates that in most practical cases there is no closed form solution for the posterior distribution $\pi_{\Theta|Y^N}(d\theta|D)$ (Gelman et al., 2013).

To get a clear path towards a satisfying solution for the Bayesian inference problem, it is useful to consider what the posterior is used for in practice. The distribution and its density are themselves not easily interpreted, especially for high dimensional parameter spaces, so that it is necessary to condense the information into summary statistics. Most relevant computations can therefore be formulated in terms of an expectation of a measurable function $f : \Theta \longrightarrow \mathbb{R}$ w.r.t. the posterior distribution:

$$\mathbb{E}_{\pi_{\Theta|Y^N}}\left[f\right](D) = \int_\Theta f(\theta)\pi_{\Theta|Y^N}(d\theta|D) = \int_\Theta f(\theta)p(\theta|D)d\theta. \tag{1.35}$$

The moments of the posterior distribution, for example, provide information about its structure and can be computed according to:

$$\mathbb{E}_{\pi_{\Theta|Y^N}}\left[\theta^n\right](D) = \int_\Theta \theta^n\pi_{\Theta|Y^N}(d\theta|D) = \int_\Theta \theta^n p(\theta|D)d\theta. \tag{1.36}$$

Making predictions about the probability of new data $\tilde{y} \in Y$ in the Bayesian framework can be formulated as:

$$\mathbb{E}_{\pi_{\Theta|Y^N}}\left[p(\tilde{y}|\theta)\right](D) = \int_\Theta p(\tilde{y}|\theta)\pi_{\Theta|Y^N}(d\theta|D) = \int_\Theta p(\tilde{y}|\theta)p(\theta|D)d\theta. \tag{1.37}$$

Intuitively therefore, the idea of the Bayesian framework is to compute averages over all possible model configurations, which are weighted according to the posterior distribution, while the classical approach provides a single final model configuration as a solution. The integrals involved in computing those averages, however, will usually neither have closed form solutions, nor will they be easily evaluated numerically because of the possibly high dimension of the parameter space. Bayesian inference thus effectively reduces to the general integration problem, which is one of the main subjects of numerical mathematics (Gelman et al., 2013; Betancourt, 2019).

1.3.4 Approximating Expectations with Samples

The *Monte Carlo method* provides an efficient approach to approximate expectations of measurable function $f : \Theta \longrightarrow \mathbb{R}$ like those mentioned in the previous subsection 1.3.3. It is based on evaluating f on exact samples $(\theta_1, \theta_2, ..., \theta_N)$ from the distribution w.r.t. which the expectation is evaluated, in this case the posterior $\pi_{\Theta|Y^N}$ (Bishop, 2006; Betancourt, 2018b):

$$\hat{f}_N^{MC} = \frac{1}{N}\sum_{n=1}^{N} f(\theta_n) \approx \mathbb{E}_{\pi_{\Theta|Y^N}}\left[f\right], \qquad \theta_n \sim \pi_{\Theta|Y^N}. \tag{1.38}$$

As covered in the introductory section 1.1.6 on representing distributions with samples, this is a consistent estimator, due to the evaluation points being samples of the relevant distribution. Moreover, this estimator has particularly nice convergence guarantees due to the samples being exact. In practice therefore, it seems sufficient to be able to generate samples from the posterior distribution to allow for efficient evaluation of expectations of interest. The closer those samples are to being exact, the better the convergence of the estimator (Betancourt, 2019).

Fitting a Bayesian model to data can thus be interpreted to be about finding a way to generate posterior samples. The problem of generating samples from distributions however, is itself not a trivial one, even if the density is available in a closed form. Note that this thesis will take the developed algorithms for practically exact sampling from distributions with standard density representations for granted. Such standard densities are, for example, the uniform density or those belonging to the *exponential family.*

Usually the prior distribution in the construction of a Bayesian model is defined by such a standard density w.r.t. the chosen parameterization. If this is the case, the model evidence $p(D) = \mathbb{E}_{\pi_\Theta}\left[\mathcal{L}_D(\theta)\right]$, i.e., the relevant normalizing constant in Bayes' rule, can be approximated efficiently via samples from the prior distribution. Even with such an approximation, however, Bayes' rule only allows for the evaluation of the posterior distribution or the corresponding density, not the generation of posterior samples, which are required for efficient approximation of the expectations of interest.

1.3.5 Sampling Methods vs. Variational Methods

There are two prominent approaches to resolve the Bayesian inference problem. They will be referred to as the sampling and the variational approach, respectively (Blei et al., 2017).

Sampling methods rely on generating posterior samples directly based on the information contained in the unnormalized finite measure μ_Θ^D representing the Bayesian model. It is trivial to see that this measure has to contain all the necessary information to be able to generate samples from the posterior distribution. This approach will be introduced in some more detail in the next Section 1.4.

Variational methods on the other hand take the same approach that underlies classical statistical modeling itself. The idea is to define a family of density functions corresponding to distributions on the parameter space. This is usually called the variational family. The goal then is to find the member of that family, which is closest to representing the true posterior distribution according to some measure of divergence. The variational approach will be described in much more detail in Chapter 2.

A notable difference between sampling and variational methods is that the variational approach actually yields an effectively closed form approximation to the posterior, which can not only be used to generate approximate posterior samples, but also to evaluate posterior densities. Sampling methods only produce posterior samples and require additional methods like kernel density estimation to approximate posterior densities based on those samples.

1.4 Sampling Methods - State of the Art

Sampling methods, in particular so called Markov Chain Monte Carlo methods, are the state of the art approach to solving the Bayesian inference problem (Gelman et al., 2013). As mentioned in the previous section, they generate samples from the posterior based on the information contained in the Bayesian model, which can be considered an unnormalized version of the posterior distribution. This section will briefly introduce naive sampling approaches and the more advanced Markov Chain Monte Carlo methods with the objective of highlighting the benefits and drawbacks of the current powerhouse of Bayesian inference. These considerations will elucidate the need for variational methods.

The goal of any sampling method is to generate samples from a *target distribution* π_Q over some generic measurable space $(Q, \mathcal{B}_Q)$, where $\mathcal{B}_Q$ is the Borel σ-algebra, induced by the underlying topology on this space. In the Bayesian inference context this is the posterior distribution $\pi_{\Theta|Y^N}$ over a parameter space $(\Theta, \mathcal{B}_\Theta)$.

What makes those approaches feasible for Bayesian inference is that they only require knowledge of the target distribution up to normalization. That is, it is sufficient to have access to the corresponding *unnormalized target measure* $\mu_Q = Z\pi_Q$ with $Z \in \mathbb{R}^+$, which corresponds to the Bayesian model $\mu_\Theta^D(d\theta) = \mathcal{L}_D(\theta)\pi_\Theta(d\theta)$ interpreted as an unnormalized measure over the parameter space. They could, however, just as well be applied to the problem of sampling from any complex distribution that can be evaluated up to a normalizing constant (Bishop, 2006; Gelman et al., 2013).

Most sampling approaches rely on an *auxiliary distribution* η_Q over the same space. This distribution is usually defined by some standard density function $h(q)$ for which exact sampling procedures are available.

1.4.1 Naive Sampling Approaches

There are two naive sampling approaches worth mentioning, both of which are explained in detail by Gelman et al. (2013) and Bishop (2006). The first one is called *importance sampling.* This method is special in the sense that it does not explicitly generate samples from the target distribution, but allows for immediate evaluation of expectations of functions $f : Q \longrightarrow \mathbb{R}$ w.r.t. the target distribution. This is achieved by reformulating the expectation of interest with a *change of measure* to the auxiliary distribution:

$$\mathbb{E}_{\pi_Q}[f] = \int_Q f(q)\pi_Q(dq) = \int_Q f(q)\frac{d\pi_Q}{d\eta_Q}(q)\eta_Q(dq) = \mathbb{E}_{\eta_Q}\left[\frac{d\pi_Q}{d\eta_Q} \cdot f\right], \tag{1.39}$$

which requires the auxiliary distribution to be chosen such that the target distribution is absolutely continuous w.r.t. it. If π_Q is not accessible directly, but only through the unnormalized measure μ_Q, this can be accounted for by also approximating the normalization constant Z via a change of measure. It is obvious that the expectation then takes the form:

$$\mathbb{E}_{\pi_Q}[f] = \mathbb{E}_{\eta_Q}\left[\frac{1}{Z} \cdot \frac{d\mu_Q}{d\eta_Q} \cdot f\right] \approx \frac{1}{N}\sum_{n=1}^{N} \frac{1}{Z}\frac{d\mu_Q}{d\eta_Q}(q_n)f(q_n) \qquad q_n \sim \eta_Q, \tag{1.40}$$

while the constant Z is just:

$$Z = \int_Q \mu_Q(dq) = \int_Q \frac{d\mu_Q}{d\eta_Q}\eta_Q(dq) = \mathbb{E}_{\eta_Q}\left[\frac{d\mu_Q}{d\eta_Q}\right] \approx \frac{1}{N}\sum_{n=1}^{N}\frac{d\mu_Q}{d\eta_Q}(q_n) \qquad q_n \sim \eta_Q. \tag{1.41}$$

The Radon-Nikodym derivative of μ_Q w.r.t. η_Q can be called the *importance function*, which returns a so called importance weight for each sample $q_n \sim \eta_Q$. Those weights account for the fact that the expectation is computed w.r.t. the auxiliary distribution. Samples from neighborhoods which are assigned much more probability mass by the auxiliary distribution than by the target distribution, will have very little impact on the expectation. It is noteworthy however, that all the generated samples are used in the approximation and if the target and auxiliary distribution are equal, this approach reduces to Monte Carlo approximation.

In the *rejection sampling* method the idea is somewhat similar. To generate samples from the target distribution, this approach first generates samples from the auxiliary distribution and rejects those with small importance weights with correspondingly high probability. The details are not important for the following considerations. Note however, that the samples produced by rejection sampling are exact samples from the target distribution, because they are generated independent of each other, which allows to use Monte Carlo approximation to evaluate expectations of interest.

Finally, it is apparent that the introduced naive sampling approaches become inefficient rather quickly, if the chosen auxiliary distribution is not sufficiently similar to the target distribution. In the case of importance sampling this is reflected in small weights for a large number of samples, whereas in rejection sampling it results in a large number of samples being rejected. In practice, it is usually hard to find appropriate auxiliary distributions especially for the Bayesian inference problem of complex models.

1.4.2 Markov Chain Monte Carlo Methods

It is clear from the Monte Carlo method introduced earlier, that exact samples are the holy grail for efficient approximation of expectations w.r.t. some target distribution π_Q. Since methods for generating exact posterior samples are, however, extremely costly, *Markov chain Monte Carlo* (MCMC) methods settle for the next best thing. In particular, they explore the idea of generating close to exact samples, more specifically, samples from a Markov chain. This section is mostly based on Betancourt et al. (2014), which not only provides a summary of the theory of Markov Chain Monte Carlo methods, but in particular a geometric analysis of the state of the art Hamiltonian Monte Carlo method.

In Subsection 1.1.3, Markov kernels were introduced as probability kernels from a measurable space $(Q, \mathcal{B}_Q)$ to itself. Just as any other transition kernel, a Markov kernel $\mathcal{T} : Q \times \mathcal{B}_Q \longrightarrow [0, 1]$ can push forward a probability distribution. This operation can be considered a map from the space of probability distributions on $(Q, \mathcal{B}_Q)$ onto itself, defined by:

$$\tau : \Pi(Q) \longrightarrow \Pi(Q), \qquad \varpi_Q \longmapsto (\varpi_Q \mathcal{T})(\,\cdot\,) = \int_Q \mathcal{T}(q,\,\cdot\,)\varpi(dq). \tag{1.42}$$

This is usually called the *Markov transition* defined by the Markov kernel. A *Markov chain* thus is a sequence of probability distributions $(\varpi_Q^1, \varpi_Q^2, ...\varpi_Q^N)$, generated from some initial distribution ϖ_Q^1, by repeated application of a Markov transition. Samples from a Markov chain are then a sequence $(q_1, q_2, ...q_N)$ with $q_i \sim \varpi_Q^i$. Since two successive distributions are related through the application of the Markov transition, the samples can be generated via *ancestral sampling* according to:

$$\begin{aligned} q_1 &\sim \varpi_Q^1 \\ q_n &\sim \mathcal{T}(q_{n-1},\,\cdot\,), \end{aligned} \tag{1.43}$$

where $\mathcal{T}(q_{n-1},\,\cdot\,)$ defines a probability distribution over $(Q, \mathcal{B}_Q)$ at $q_{n-1} \in Q$. If $\varpi_Q = \varpi_Q \mathcal{T}$ holds for some probability distribution, then the Markov kernel and the corresponding transition are said to preserve this distribution and ϖ_Q is referred to as a *stationary distribution* w.r.t. this kernel.

To generate samples from the target distribution π_Q, the idea is to define a transition kernel $\mathcal{T}$ that preserves it, i.e., $\pi_Q = \pi_Q \mathcal{T}$. Then most of the distributions in the generated Markov chain will eventually be sufficiently close to the target distribution and yield the desired samples. Those will, however, not be independent but correlated to some extend. The degree of correlation of successive samples can be quantified by the *autocorrelation* of the Markov chain, the details of which will not be covered here.

This approach allows for more efficient but sequential generation of samples, while sacrificing some of the convergence speed of the corresponding MCMC-estimator of expectations of functions $f : Q \longrightarrow \mathbb{R}$ w.r.t. the target distribution:

$$\hat{f}_N^{MCMC}(q_1) = \frac{1}{N} \sum_{n=1}^{N} f(q_n), \tag{1.44}$$

where the evaluation points $(q_1, q_2, ...q_N)$ are the samples from a generated Markov chain.

Any particular MCMC method is therefore defined by some Markov transition kernel that preserves the target distribution. The construction of such kernels is, however, not trivial. In particular, the goal is to find kernels which generate Markov chains that efficiently explore the space of probability distributions as to find the target distribution quickly, while minimizing the autocorrelation. The well known *Gaussian Random Walk Metropolis* kernel is a simple example that can be explicitly constructed via conditional densities. For more complex target distributions and high dimensional spaces Q, however, it yields only diffuse local exploration of $\Pi(Q)$ in finite time.

A more natural way of constructing Markov kernels is to make use of automorphisms on the measure space $(Q, \mathcal{B}_Q, \pi_Q)$, that is, continuous measure preserving bijections $g : Q \longrightarrow Q$. The idea here is to define a family Γ of such maps, all of which preserve the target distribution, i.e., $\pi_Q = \pi_Q \circ g^{-1} \quad \forall g \in \Gamma$. With the choice of a σ-algebra and a probability measure they form a probability space $(\Gamma, \Sigma_\Gamma, \gamma)$, which induces a Markov kernel according to:

$$\mathcal{T}(q, A) := \int_\Gamma \mathbb{I}_A(g(q))\gamma(dg) \quad q \in Q, A \in \mathcal{B}_Q, \tag{1.45}$$

where $\mathbb{I}_A$ is the indicator function for the set A. This kernel, and thus the induced transition, will then preserve the target distribution, because it is the convolution of maps that preserve it. The Gaussian Random Walk Metropolis kernel, for example, can be constructed from random independent translation on Q.

To ensure efficient exploration of $\Pi(Q)$ and small autocorrelation of the resulting Markov chains, a more coherent and less diffuse behavior of the chosen family of automorphisms is required. Flows potentially fulfill those requirements and provide the foundation for the *Hamiltonian Monte Carlo (HMC)* method.

A *flow* on a Space Q is a group action of the additive group of the real numbers on that space. That is, it is a family of automorphisms $\phi_t : Q \longrightarrow Q$ parameterized by a real parameter $t \in \mathbb{R}$, which is often interpreted as time, with the following properties:

$$\begin{aligned} \phi_r \circ \phi_s &= \phi_{r+s} \\ \phi_t^{-1} &= \phi_{-t} \\ \phi_0 &= \mathrm{Id}_Q. \end{aligned} \tag{1.46}$$

Since the inversion of a flow is only achieved by inversion of the time parameter, increasing the parameter will coherently push away the points $q \in Q$ from their initial positions and Markov chains constructed on their basis will, in turn, yield efficient exploration of $\Pi(Q)$ with small autocorrelation.

The HMC method provides a canonical way of constructing such flows ϕ_t^H, called Hamiltonian flows, on the *cotangent bundle* T^*Q of the space Q. The notion of a cotangent bundle will not be discussed in detail, but it should be noted that it requires $(Q, \mathcal{B}_Q)$ to be at least a differentiable manifold.

In principle, the HMC method utilizes Hamiltonian dynamics to generate a Markov chain, which, by construction of the respective Hamiltonian, preserves the target distribution. The method is covered in detail by (Betancourt et al., 2014) and on a more intuitive level by (Betancourt, 2017a; Neal, 2011; Gelman et al., 2013; Bishop, 2006). Although the details of the exact implementation are beyond the scope of this thesis, Hamiltonian systems and flows will be reviewed in a simplified version in chapter 4, when introducing Hamiltonian normalizing flows.

1.4.3 Benefits and Drawbacks

Markov chain Monte Carlo methods provide a coherent approach to solving the Bayesian inference problem by generating correlated posterior samples. A major benefit are the theoretic convergence guarantees of MCMC estimators for approximating expectations. Although the convergence speed suffers due to the samples not being exact samples of the posterior distribution, they are still a significant improvement to naive sampling approaches which try to generate independent posterior samples at prohibitively high costs.

The Hamiltonian Monte Carlo method is the state of the art sampling method and provides a theoretically efficient approach to explore the space of distributions and minimize the autocorrelation of the generated Markov chain and corresponding samples. Even though the HMC method and its more sophisticated extensions (Hoffman & Gelman, 2011; Girolami & Calderhead, 2011) provide some significant improvements, they also increase the computation costs per transition and generating posterior samples remains costly. Note that the costs per transition, in particular, strongly depend on the size of the dataset, since the unnormalized target measure, i.e., the corresponding density, has to be evaluated on each transition. In case of the HMC method, this increases to one evaluation of the density for each integration step in the simulation of the Hamiltonian dynamics. They also introduce hyperparameters[1] that need to be tuned to every individual problem via sophisticated techniques.

Note also, that MCMC method are not parallelizable, because the samples are generated sequentially and each sample depends on the previous one by design. Although it is possible to run multiple chains in parallel, which is obviously useful, even a single chain may incur prohibitively high computational cost.

1.5 Thesis Goals and Structure

The subject of this thesis will be variational Bayesian inference, in particular, the application of so called normalizing flows for this purpose. More specifically, after this introductory review of the relevant theory regarding Bayesian inference and a brief discussion of sampling methods, the following chapters will lead up to Hamiltonian normalizing flows, which were first introduced by Toth et al. (2019) in the context of density estimation.

Chapter 2 will provide a precise formulation of the variational inference approach as an optimization problem, the goal of which is to minimizes the divergence between the members of a variational family to some target distribution. Variational Bayesian inference then is variational inference applied to the Bayesian inference problem with the posterior as the target distribution.

Normalizing flows are introduced in Chapter 3. After briefly noting the more general domains of application for normalizing flows they are discussed in the context of variational Bayesian inference.

[1] A mass matrix M, integration time L and time step size ϵ for computing the Hamiltonian dynamics

As a particularly interesting example, residual flows will be highlighted. The limiting case of compositions of infinitely many residual flows leads to the notion of so called continuous normalizing flows.

Chapter 4 will first summarize the Hamiltonian formalism of classical mechanics, specifically the notion of a phase space and Hamiltonian flows on it, before discussing Hamiltonian normalizing flows as a special instance of continuous normalizing flows.

In the context of this thesis, a software package for the purpose of Bayesian inference is developed, utilizing TensorFlow Probability (Dillon et al., 2017). Although there are already many such frameworks available, this one focuses on providing a flexible way of defining normalizing flows to be used in the context of variational Bayesian inference. The package will be used to apply Hamiltonian normalizing flows to selected Bayesian inference problems in Chapter 5. To verify and evaluate the results, they are compared to those from the state of the art Hamiltonian Monte Carlo method. The goal is to qualitatively show that Hamiltonian normalizing flows can be used for variational Bayesian inference.

Finally, Chapter 6 will present a conclusion regarding the application of Hamiltonian normalizing flows for the purpose of Bayesian inference. This will be accompanied by a discussion of the potentials of this approach and prospects for future research.

Chapter 2

Variational Inference

As eluded to in the introduction, variational inference follows the philosophy of classical statistical modeling by selecting a so called variational family of distributions, that is a subset $\Gamma \subseteq \Pi(Q)$ of the distributions over a measurable space $(Q, \mathcal{B}_Q)$ of interest. The idea is to find the member of this family $\gamma^\dagger \in \Gamma$, that is closest to a target distribution $\pi_Q \in \Pi(Q)$ according to some measure of divergence $\mathcal{D}$. Note that, in the literature (compare for example Blei et al. (2017), C. Zhang et al. (2017), Hoffman et al. (2013) and Ranganath et al. (2013)), variational inference is commonly restricted to refer to variational Bayesian inference in particular. This thesis, however, will consider variational inference from a more general perspective. In some sense, it can be thought of as generalizing the maximum likelihood approach underlying classical parameter inference, such that it is applicable to the Bayesian inference problem as well.

2.1 Variational Inference as Optimization

The variational inference approach can be formulated as an optimization task, specifically, as the problem of minimizing a divergence measure between a variational family and a target distribution. To understand this in detail it is necessary to briefly introduce divergences, before discussing the optimization problem itself.

2.1.1 Divergences

Divergences are measures of distance between probability distributions. They are, however, a weaker notion than that of a metric, since they are not necessarily symmetric nor do they have to satisfy the triangle inequality. More specifically, a divergence on $\Pi(Q)$ is a map $\mathcal{D} : \Pi(Q) \times \Pi(Q) \longrightarrow \mathbb{R}$, mapping any two probability distributions onto the non-negative real numbers, where:

$$\mathcal{D}(\pi||\gamma) \geq 0 \quad \forall \pi, \gamma \in \Pi(Q) \tag{2.1}$$

and:

$$\mathcal{D}(\pi||\gamma) = 0 \qquad \Longleftrightarrow \qquad \pi = \gamma \qquad \pi, \gamma \in \Pi(Q). \tag{2.2}$$

In other words, a divergence between two distributions is zero if and only if they are the same and positive otherwise. To emphasize the asymmetry of a divergence $\mathcal{D}(\pi||\gamma)$, it is usually read as the 'divergence of π from γ'.

There is a particularly interesting class of divergences called *f-divergences* (Ali & Silvey, 1966; Amari, 2009). They are of the general form:

$$\mathcal{D}_f(\pi||\gamma) := \mathbb{E}_\gamma\left[f \circ \frac{d\pi}{d\gamma}\right] = \int_Q \left(f \circ \frac{d\pi}{d\gamma}\right)(q)\gamma(dq) \qquad \pi, \gamma \in \Pi(Q) \tag{2.3}$$

for any convex function $f : \mathbb{R} \longrightarrow \mathbb{R}$ with $f(1) = 0$, which ensures $\mathcal{D}_f(\pi||\gamma) = 0$ if and only if π and γ are equal. Note that all f-divergences can immediately be interpreted as an expectation w.r.t. the second argument, while a change of measure allows to express it as an expectation w.r.t. the first argument:

$$\mathcal{D}_f(\pi||\gamma) := \mathbb{E}_\gamma\left[f \circ \frac{d\pi}{d\gamma}\right] = \mathbb{E}_\pi\left[\frac{d\gamma}{d\pi} \cdot f \circ \frac{d\pi}{d\gamma}\right]. \tag{2.4}$$

A mathematically convenient member of those f-divergences is the *Kullback-Leibler (KL) divergence* $\mathcal{D}_{KL}$ defined by:

$$\mathcal{D}_{KL}(\pi||\gamma) := \mathbb{E}_\gamma\left[\frac{d\pi}{d\gamma} \cdot \left(\log \circ \frac{d\pi}{d\gamma}\right)\right] = \int_Q \frac{d\pi}{d\gamma}(q)\left(\log \circ \frac{d\pi}{d\gamma}\right)(q)\gamma(dq), \tag{2.5}$$

where obviously $f(u) = u \log u$. Using a change of measure, the usual form of the KL-divergence is recovered as:

$$\mathcal{D}_{KL}(\pi||\gamma) = \mathbb{E}_\gamma\left[\frac{d\pi}{d\gamma} \cdot \left(\log \circ \frac{d\pi}{d\gamma}\right)\right] = \mathbb{E}_\pi\left[\log \circ \frac{d\pi}{d\gamma}\right] = -\mathbb{E}_\pi\left[\log \circ \frac{d\gamma}{d\pi}\right] =: \mathcal{D}^*_{KL}(\gamma||\pi). \tag{2.6}$$

For the sake of completeness it is interesting to note that this form actually corresponds to the *dual KL-divergence* $\mathcal{D}^*_{KL}$, which is itself an f-divergence with $f(u) = -\log u$. More generally, for each divergence $\mathcal{D}$, the map $\mathcal{D}^*$ with $\mathcal{D}^*(\gamma||\pi) = \mathcal{D}(\pi||\gamma)$ is called its dual.

The usual form of the KL-divergence emphasizes its interpretation as *relative entropy* (Bishop, 2006). Note that, as opposed to entropy, it is a structure defined on the space of distributions. The (differential) *entropy* of a distribution π only exists w.r.t. some base measure λ:

$$\mathcal{H}\left(\frac{d\pi}{d\lambda}\right) := -\mathbb{E}_{\pi}\left[\log\circ\frac{d\pi}{d\lambda}\right]. \tag{2.7}$$

That is, entropy is only defined for densities or, more typically, for random variables with a particular associated density (Bishop, 2006).

The use of the KL-divergence for the purpose of variational inference is pervasive (Blei et al., 2017; C. Zhang et al., 2017; Hoffman et al., 2013; Ranganath et al., 2013). The reason is mostly that the logarithm is a *group homomorphism* from the multiplicative to the additive group of the real numbers and very small values are mapped to large negative values. Both properties are highly advantageous for actual numerical computations involving probabilities, which in practice often come down to multiplication of a large number of small values leading to underflow problems. Those problems can generally be avoided in the logarithmic domain.

2.1.2 The General Optimization Problem

The term variational inference is used in reference to the *calculus of variations*, which covers optimization of real *functionals*, that is, maps from a space of functions to the real numbers (Bishop, 2006). Since the target distribution $\pi_Q \in \Pi(Q)$ in variational inference is fixed, any divergence $\mathcal{D} : \Pi(Q) \times \Pi(Q) \longrightarrow \mathbb{R}$ defines two functionals:

$$\begin{aligned} \mathcal{D}(\pi_Q||\,\cdot\,) &: \Pi(Q) \longrightarrow \mathbb{R}, \quad \varpi \mapsto \mathcal{D}(\pi_Q||\varpi), \\ \mathcal{D}(\,\cdot\,||\pi_Q) &: \Pi(Q) \longrightarrow \mathbb{R}, \quad \varpi \mapsto \mathcal{D}(\varpi||\pi_Q). \end{aligned} \tag{2.8}$$

Because of the asymmetry of divergences they are obviously not equivalent and therefore suggest two possible optimization problems associated with a divergence choice. Both functionals by construction assign a non-negative real number to each distribution $\varpi \in \Pi(Q)$. Naively, finding the target distribution in the space of all distributions on the relevant space simply requires searching for the element $\varpi^\dagger \in \Pi(Q)$ for which $\mathcal{D}(\pi_Q||\varpi^\dagger) = \mathcal{D}(\varpi^\dagger||\pi_Q) = 0$. Then, according to the properties of divergences, it is apparent that $\varpi^\dagger = \pi_Q$. Minimization of either functional over the total space of distributions thus amounts to exact inference, since the target distribution necessarily is contained in $\Pi(Q)$.

Just as in the case of classical statistical inference, however, this optimization problem is typically intractable. The remedy again is to choose an appropriate subset $\Gamma \subseteq \Pi(Q)$, called the *variational family*, which is sufficiently small or has a convenient structure as to ensure tractability of the optimization problem. Variational inference then reduces to approximate inference, since there is no guarantee that the target distribution is contained in the selected variational family. With this restriction, the goal is to find the member $\gamma_Q^\dagger \in \Gamma$ that minimizes the functional of interest and is in this sense closest to the target distribution.

For the reasons eluded to in the previous subsection, the following discussion will restrict to the KL-divergence. Since evaluating the functionals associated with the KL-divergence, or any f-divergence for that matter, requires approximation of an expectation, it is useful to distinguish the cases in which samples of the target distribution are either accessible or not.

If the goal is to fit a variational family to an assumed true underlying distribution of a dataset, then this dataset by assumption constitutes samples from the target distribution. In this case the natural choice of functional is $\mathcal{D}_{KL}(\pi_Q || \cdot)$, such that the optimization problem is put in terms of an expectation w.r.t. the target distribution. The goal then is to find $\gamma_Q^\dagger \in \Gamma$ such that:

$$\gamma_Q^\dagger := \underset{\gamma_Q \in \Gamma}{\arg\min}\, \mathcal{D}_{KL}(\pi_Q || \gamma_Q) = \underset{\gamma_Q \in \Gamma}{\arg\min} -\mathbb{E}_{\pi_Q} \left[\log \circ \frac{d\gamma_Q}{d\pi_Q} \right]. \tag{2.9}$$

As a convention it is useful to follow Papamakarios et al. (2019) and call $\mathcal{D}_{KL}(\pi_Q || \cdot)$ the *forward KL-divergence*, where the target distribution is in the first argument. For a choice of base measure λ, this can easily be shown to be equivalent to the maximum likelihood approach introduced in Subsection 1.2.4. To illustrate this, see how the KL-divergence can be deconstructed into:

$$\begin{aligned}
\mathcal{D}_{KL}(\pi_Q || \gamma_Q) &= -\mathbb{E}_{\pi_Q} \left[\log \circ \frac{d\gamma_Q}{d\pi_Q} \right] \\
&= -\mathbb{E}_{\pi_Q} \left[\log \circ \left(\frac{d\gamma_Q}{d\lambda} \cdot \frac{d\lambda}{d\pi_Q} \right) \right] \\
&= -\mathbb{E}_{\pi_Q} \left[\log \circ \frac{d\gamma_Q}{d\lambda} \right] + \mathbb{E}_{\pi_Q} \left[\log \circ \frac{d\pi_Q}{d\lambda} \right] \\
&= -\mathbb{E}_{\pi_Q} \left[\log \circ \frac{d\gamma_Q}{d\lambda} \right] - \mathcal{H} \left(\frac{d\pi_Q}{d\lambda} \right).
\end{aligned} \tag{2.10}$$

Since the entropy of the density of a fixed distribution is just a constant, it can be ignored for the purpose of optimization. Moreover, the expectation term can be approximated using the Monte Carlo method with the target distribution's samples contained in the dataset:

$$- \mathbb{E}_{\pi_Q} \left[\log \circ \frac{d\gamma_Q}{d\lambda} \right] \approx -\frac{1}{N} \sum_{n=1}^{N} \left(\log \circ \frac{d\gamma_Q}{d\lambda} \right) (q_n) \quad \text{with} \quad q_n \sim \pi_Q. \tag{2.11}$$

Since the Radon-Nikodym derivative is the density of the variational family member, minimizing the term within the family is equivalent to finding the member that maximizes the probability density of the data, which is exactly the maximum likelihood approach.

If, on the other hand, the goal is to apply variational inference in the context of Bayesian inference, then samples of the target distribution, i.e. the posterior, are obviously not available. In this case it is convenient to choose the functional $\mathcal{D}_{KL}(\,\cdot\,||\pi_Q)$, putting the optimization problem in terms of an expectation w.r.t. the variational family members, which are usually chosen such that samples can be easily generated. The goal then is to find $\gamma_Q^\dagger \in \Gamma$, such that:

$$\gamma_Q^\dagger := \underset{\gamma_Q \in \Gamma}{\arg\min}\, \mathcal{D}_{KL}(\gamma_Q||\pi_Q) = \underset{\gamma_Q \in \Gamma}{\arg\min} -\mathbb{E}_{\gamma_Q}\left[\log \circ \frac{d\pi_Q}{d\gamma_Q}\right], \tag{2.12}$$

where $\mathcal{D}_{KL}(\,\cdot\,||\pi_Q)$ is called the *reverse KL-divergence* (Papamakarios et al., 2019), with the target distribution in the second argument. Applying variational inference to solve the Bayesian inference problem is discussed in more detail in Section 2.2.

2.1.3 Choosing a Variational Family

In practice, an important question is how to choose an appropriate variational family $\Gamma \subseteq \Pi(Q)$. In the best case Γ covers a large number of different distributions, that is, it is a particularly rich family of distributions, for which the optimization remains tractable (Bishop, 2006).

Generally, optimization over a space of functions is impractical. The usual approach therefore is to choose a parameterization for the set Γ, such that it can be associated with some variational parameter space $\Psi \subseteq \mathbb{R}^r$ for $r \in \mathbb{N}$. This allows to select a variational family associated with a family of density functions $q^\psi(q) := q(q;\psi)\ \ \forall \psi \in \Psi$ over the underlying space Q. For clarity, any member of the variational family will be indexed with the associated variational parameter, i.e. $\gamma_Q^\psi \in \Gamma$ is the distribution represented by $q^\psi(q)$ w.r.t. some fixed base measure.

As a very simple example, a Gaussian or any other standard family of density functions can be used to define a variational family. Another prominent approach is called *mean-field approximation*, where the variational family is chosen to be the set of all factorizable distributions, that is, their multivariate densities factorize completely into univariate densities. Since this set is not readily usable in black box variational inference approaches, there is usually an additional assumption about the density of the component distributions w.r.t. some base measure. The variational family will thus usually consist of all factorizable distributions, where the components can, for example, all be represented via univariate Gaussian densities. This is obviously a smaller family than that of the distributions with multivariate Gaussian density representation, typically leading to worse but computationally more efficient approximations (Bishop, 2006; Blei et al., 2017; Kucukelbir et al., 2016).

Note that those examples are all variants of the same idea - utilizing standard families of density functions to define variational families. There are, however, other approaches to define variational families, one of which makes use of normalizing flows and will be discussed in Chapter 3.

If the variational family is assumed to have a fixed structure, e.g., by being defined via a family of density functions $q^\psi(q)$ in some parameterization, the forward and reverse KL-divergence can be considered just functions of the variational parameters, instead of functionals of the variational family members. The optimization problems can then be reformulated as:

$$\begin{aligned} \psi^\dagger &:= \operatorname*{arg\,min}_{\psi\in\Psi} \mathcal{D}_{KL}(\pi_Q||\gamma_Q^\psi) = \operatorname*{arg\,min}_{\psi\in\Psi} -\mathbb{E}_{\pi_Q}\left[\log\frac{d\gamma_Q^\psi}{d\pi_Q}\right], \\ \psi^\dagger &:= \operatorname*{arg\,min}_{\psi\in\Psi} \mathcal{D}_{KL}(\gamma_Q^\psi||\pi_Q) = \operatorname*{arg\,min}_{\psi\in\Psi} -\mathbb{E}_{\gamma_Q^\psi}\left[\log\frac{d\pi_Q}{d\gamma_Q^\psi}\right], \end{aligned} \tag{2.13}$$

where the solution $\psi^\dagger$ corresponds to the associated distribution $\gamma_Q^\dagger := \gamma_Q^{\psi^\dagger}$, which is the closest match to the target distribution within the chosen family.

2.2 Variational Bayesian Inference

In variational Bayesian inference, the relevant measurable space is the parameter space $(\Theta, \mathcal{B}_\Theta)$ associated with some parameterization of the model configuration space $\mathcal{S} \subseteq \Pi(Y)$. The target distribution in this context is the posterior $\pi_{\Theta|Y^N}$, whereas the unnormalized target measure $\mu_\Theta^D(d\theta) = Z\pi_{\Theta|Y^N}(d\theta)$ corresponds to the Bayesian model $\mu_\Theta^D(d\theta) = \mathcal{L}_D(\theta)\pi_\Theta(d\theta)$ interpreted as an unnormalized measure over the parameter space. Recall that π_Θ is the prior distribution, $\mathcal{L}_D$ the likelihood function and the constant Z is the so called model evidence $p(D)$, the computation of which is usually intractable.

A variational family Γ, in this context, is a subset of the distributions over the parameter space, i.e., $\Gamma \subseteq \Pi(\Theta)$. As described in the previous section, the reverse KL-divergence $\mathcal{D}_{KL}(\,\cdot\,||\pi_{\Theta|Y^N})$ is the typical choice of functional to consider for optimization in variational Bayesian inference:

$$\gamma_\Theta^\dagger := \operatorname*{arg\,min}_{\gamma_\Theta\in\Gamma} \mathcal{D}_{KL}(\gamma_\Theta||\pi_{\Theta|Y^N}) = \operatorname*{arg\,min}_{\gamma_\Theta\in\Gamma} -\mathbb{E}_{\gamma_\Theta}\left[\log\circ\frac{d\pi_{\Theta|Y^N}}{d\gamma_\Theta}\right]. \tag{2.14}$$

It becomes clear immediately that minimizing the reverse KL-divergence is not possible, because it cannot be evaluated without access to the posterior distribution and its density. This is the same fundamental problem as encountered in sampling approaches, and again the solution is to notice that it is sufficient to have access to the unnormalized target measure μ_Θ^D (Blei et al., 2017).

2.2.1 Variational Free Energy

With the above observations about the target distribution, the reverse KL-divergence of a variational family member from the posterior distribution can be rewritten as:

$$\begin{aligned}\mathcal{D}_{KL}(\gamma_\Theta||\pi_{\Theta|Y^N}) &= -\mathbb{E}_{\gamma_\Theta}\left[\log\circ\frac{d\pi_{\Theta|Y^N}}{d\gamma_\Theta}\right]\\ &= -\mathbb{E}_{\gamma_\Theta}\left[\log\circ\left(\frac{d\pi_{\Theta|Y^N}}{d\mu_\Theta^D}\cdot\frac{d\mu_\Theta^D}{d\gamma_\Theta}\right)\right]\\ &= -\mathbb{E}_{\gamma_\Theta}\left[\log\circ\left(\frac{1}{Z}\cdot\frac{d\mu_\Theta^D}{d\gamma_\Theta}\right)\right]\\ &= -\mathbb{E}_{\gamma_\Theta}\left[\log\circ\frac{d\mu_\Theta^D}{d\gamma_\Theta}\right]+\log Z\\ &= \mathcal{F}_{\pi_\Theta,\mathcal{L}_D}(\gamma_\Theta)+\log Z\end{aligned} \tag{2.15}$$

The term $\mathcal{F}(\gamma_\Theta) := \mathcal{F}_{\pi_\Theta,\mathcal{L}_D}(\gamma_\Theta)$ is usually called the *variational free energy* in an analogy to statistical physics, which will not be discussed further. It can be assumed to only depend on the variational family member γ_Θ, since the prior π_Θ and the likelihood $\mathcal{L}_D$ are fixed for any given Bayesian inference problem as they define the Bayesian model. Similar to the reverse KL-divergence, it is therefore a functional of the variational family members, but can readily be evaluated with the information provided by the model. Since the normalization constant Z turns out to be irrelevant from an optimization perspective, the optimization problem can be reformulated as:

$$\begin{aligned}\gamma_\Theta^\dagger :&= \underset{\gamma_\Theta\in\Gamma}{\arg\min}\,\mathcal{D}_{KL}(\gamma_\Theta||\pi_{\Theta|Y^N})\\ &= \underset{\gamma_\Theta\in\Gamma}{\arg\min}\,\mathcal{F}(\gamma_\Theta)\\ &= \underset{\gamma_\Theta\in\Gamma}{\arg\min}\,-\mathbb{E}_{\gamma_\Theta}\left[\log\circ\frac{d\mu_\Theta^D}{d\gamma_\Theta}\right]\end{aligned} \tag{2.16}$$

In much of the literature the term *Evidence Lower BOund (ELBO)* is used to refer to the negative variational free energy (Blei et al., 2017; C. Zhang et al., 2017; Rezende & Mohamed, 2015). This terminology is easily understood by recognizing that the constant Z corresponds to the model evidence, while:

$$-\mathcal{F}(\gamma_\Theta) = \log Z - \mathcal{D}_{KL}(\gamma_\Theta||\pi_{\Theta|Y^N}), \tag{2.17}$$

following from equation (2.15), and the non-negativity of divergences, imply that $-\mathcal{F}(\gamma_\Theta)$ is a tight lower bound for the log model evidence. It is a tight bound in the sense that it is equal to the log model evidence only in case the variational family member is equal to the posterior distribution. It is trivial to see that minimizing the variational free energy is equivalent to maximizing the ELBO.

To gain a deeper understanding of what it means to minimize the variational free energy, it can be decomposed further according to:

$$\begin{aligned}
\mathcal{F}(\gamma_\Theta) &= -\mathbb{E}_{\gamma_\Theta}\left[\log \circ \frac{d\mu_\Theta^D}{d\gamma_\Theta}\right] \\
&= -\mathbb{E}_{\gamma_\Theta}\left[\log \circ \left(\frac{d\mu_\Theta^D}{d\pi_\Theta} \cdot \frac{d\pi_\Theta}{d\gamma_\Theta}\right)\right] \\
&= -\mathbb{E}_{\gamma_\Theta}\left[\log \circ \left(\mathcal{L}_D \cdot \frac{d\pi_\Theta}{d\gamma_\Theta}\right)\right] \\
&= -\mathbb{E}_{\gamma_\Theta}\left[\log \circ \frac{d\pi_\Theta}{d\gamma_\Theta}\right] - \mathbb{E}_{\gamma_\Theta}\left[\log \circ \mathcal{L}_D\right] \\
&= \mathcal{D}_{KL}(\gamma_\Theta || \pi_\Theta) - \mathbb{E}_{\gamma_\Theta}\left[\log \circ \mathcal{L}_D\right].
\end{aligned} \tag{2.18}$$

By inspection, it is clear that minimizing the expected negative log likelihood $-\mathbb{E}_{\gamma_\Theta}[\log \circ \mathcal{L}_D]$ is equivalent to finding the variational family member that maximizes the expected probability of the data. The term $\mathcal{D}_{KL}(\gamma_\Theta || \pi_\Theta)$ on the other hand has a regularizing effect, penalizing strong divergence of the variational family member from the prior distribution. The latter can be interpreted as a version of *Occam's razor* (Baker, 2016), where a parsimonious choice of variational family member as an approximation to the posterior is preferred. Parsimonious in this context is to be understood as preferring posterior beliefs that only moderately deviate from the prior assumptions.

Chapter 3

Normalizing Flows

This chapter introduces normalizing flows, which are the central subject of this thesis. After a brief discussion of their general application in variational inference, the focus will again be on variational Bayesian inference. In particular, their structure allows a reformulation of the variational free energy introduced in the previous chapter. There are some challenges associated with the construction and application of normalizing flows, which will be highlighted before focusing specifically on residual flows. They will lead to the notion of continuous normalizing flows and form the foundation of the Hamiltonian normalizing flows discussed in Chapter 4. The following sections are primarily based on the extensive reviews provided by Kobyzev et al. (2019) and Papamakarios et al. (2019).

3.1 Normalizing Flows and Variational Families

As in the previous chapters, assume some target distribution π_Q on a generic measurable space $(Q, \mathcal{B}_Q)$, which in the Bayesian inference context again will be the posterior distribution over parameter space. Given the choice of some base distribution η_Z on another measurable space $(Z, \mathcal{B}_Z)$, the idea of normalizing flows is to construct a variational family:

$$\Gamma(\eta_Z, f^\psi) := \Big\{ \eta_Z \circ (f^\psi)^{-1} \mid \psi \in \Psi \subseteq \mathbb{R}^r \Big\} \subseteq \Pi(Q) \quad \text{with} \quad f^\psi : Z \longrightarrow Q \tag{3.1}$$

as all the push-forwards of the base distribution along a parameterized family of isomorphisms, i.e. continuous measurable bijective transformations f^ψ, which are colloquially referred to as *normalizing flows* (Kobyzev et al., 2019). Flows in the strict mathematical sense, however, only play a role in the particular case of continuous normalizing flows, which are introduced in Section 3.4. The variational inference optimization problem here amounts to finding the flow parameters, for which the push-forward of the base distribution most accurately corresponds to the target distribution.

3.1.1 Transformation of Samples

The construction of variational families from normalizing flows provides a convenient way of producing samples for their family members. Let the base distribution η_Z be defined by a standard density function w.r.t. the base measure λ_Z, for which samples are easily generated. In practice, this will typically be a Gaussian or uniform density. Since every member of the variational family $\Gamma(\eta_Z, f^\psi)$, induced by the normalizing flow f^ψ, is a push-forward distribution $\gamma_Q^\psi = \eta_Z \circ (f^\psi)^{-1}$, samples are trivially produced by transforming samples from the base distribution (Papamakarios et al., 2019):

$$f^\psi(z) = q \sim \gamma_Q^\psi \qquad z \sim \eta_Z. \tag{3.2}$$

In particular, if the variational inference optimization problem is solved and the distribution $\gamma_Q^\dagger$ closest to the target π_Q is obtained, generating approximate samples for the target distribution is as simple as applying the corresponding transformation to samples from the base distribution.

3.1.2 Transformation of Densities

Since in practice computations are done in the realm of densities, it is interesting to investigate how exactly the density of the base distribution η_Z transforms under a particular continuous measurable bijection $f : Z \longrightarrow Q$. For this purpose let λ_Z and λ_Q be base measures on the spaces $(Z, \mathcal{B}_Z)$ and $(Q, \mathcal{B}_Q)$, respectively. It is obvious from the definition of a push-forward distribution, that the probability mass assigned by the base distribution η_Z to any measurable set $U \in \mathcal{B}_Z$, will be conserved under the transformation f:

$$\begin{aligned} \int_{U\subseteq Z} \eta_Z(dz) &= \int_{f(U)\subseteq Q} \left(\eta_Z \circ f^{-1}\right)(dq) \\ \iff \quad \int_{U\subseteq Z} h(z)\lambda_Z(dz) &= \int_{f(U)\subseteq Q} (h\circ f^{-1})(q) \left(\lambda_Z \circ f^{-1}\right)(dq). \end{aligned} \tag{3.3}$$

Here, h is the density of the base distribution w.r.t. λ_Z, while $h \circ f^{-1}$ denotes the density of the push-forward distribution w.r.t. the push-forward of λ_Z. Note that there is no guarantee for the push-forward distribution to also have a density w.r.t. the base measure λ_Q on $(Q, \mathcal{B}_Q)$.

Consider the case $Q = Z = \mathbb{R}^d$ for some $d \in \mathbb{N}$, where the base measures $\lambda_Z = \lambda_Q$ correspond to the Lebesgue measure on $\mathbb{R}^d$. This allows to assume much more structure on Q and Z, in particular a Euclidean vector space structure, such that it is possible to talk about differentiable maps between those spaces. If $f : Z \longrightarrow Q$ is now differentiable in both directions, then it is called a *diffeomorphism.* At each point $z \in Z$, the map f is locally approximated by a linear transformation called the Jacobian $J_f(z) : Z \longrightarrow Q$, as is its inverse $f^{-1} : Q \longrightarrow Z$ by the inverse Jacobian $J_{f^{-1}}(q) = J_f(z)^{-1}$ for every $f(z) = q \in Q$.

In this specific regime, the push-forward distribution is guaranteed to have a corresponding density w.r.t. the base measure λ_Q (Kobyzev et al., 2019):

$$\begin{aligned}(\eta_Z \circ f^{-1})(dq) &= (h \circ f^{-1})(q)\ (\lambda_Z \circ f^{-1})(dq) \\ &= (h \circ f^{-1})(q)\ \frac{d(\lambda_Z \circ f^{-1})}{d\lambda_Q}(q)\lambda_Q(dq) \\ &= (h \circ f^{-1})(q)\ \left|\det J_{f^{-1}}(q)\right| \lambda_Q(dq)\end{aligned} \tag{3.4}$$

and (3.3) recovers the familiar equation for a change of variable. A useful interpretation here is, that the transformed densities are obtained simply by applying a correction term based on the Jacobian determinant of the transformation f. The following chapters will restrict to this special case.

Finally, consider again a target distribution π_Q on $(Q, \mathcal{B}_Q)$. Since $Q = Z = \mathbb{R}^d$, a base distribution η_Q and a parameterized measurable diffeomorphism:

$$f^\psi : Q \longrightarrow Q, \quad q_0 \mapsto f^\psi(q_0) = q_T \tag{3.5}$$

induce a variational family $\Gamma(\eta_Q, f^\psi) \subseteq \Pi(Q)$. According to equation (3.4), the density $q^\psi(q_T)$ of each variational family member $\gamma_Q^\psi \in \Gamma(\eta_Q, f^\psi)$ can be summarized as:

$$\begin{aligned} & q^\psi(q_T) = \left(h \circ (f^\psi)^{-1}\right)(q_T)\ \left|\det J_{(f^\psi)^{-1}}(q_T)\right| \\ \iff \quad & q^\psi(q_T) = h(q_0)\ \left|\det J_{f^\psi}(q_0)\right|^{-1} \\ \iff \quad & \log q^\psi(q_T) = \log h(q_0) - \log\left|\det J_{f^\psi}(q_0)\right| \qquad \forall\, q_0, f^\psi(q_0) = q_T\ \in Q. \end{aligned} \tag{3.6}$$

Since in practice it is convenient to work with densities in the logarithmic domain, this will be the convention for the remaining discussion.

3.1.3 Compositions of Normalizing Flows

As mentioned earlier, it is desirable to be able to define rich variational families to allow for arbitrarily good approximations even of complex target distributions π_Q. Similar to how neural networks in machine learning are able to approximate arbitrary functions by composing parameterized affine transformations with intermediate nonlinear maps, composing parameterized diffeomorphisms allows for the construction of sufficiently rich variational families as to represent any target distribution. This has been proven formally for some choices of parameterized diffeomorphisms (Bogachev et al., 2005).

The observation that compositions of diffeomorphisms are still diffeomorphisms makes the idea of composing normalizing flows trivial. Let f^ψ be a parameterized family of diffeomorphisms constructed as such a composition, then this can be written as:

$$f^\psi = f_T^{\psi_T} \circ f_{T-1}^{\psi_{T-1}} \circ \ldots \circ f_2^{\psi_2} \circ f_1^{\psi_1} \quad \text{with} \quad f_k^{\psi_k}(q_{k-1}) = q_k,\ f^\psi(q_0) = q_T, \tag{3.7}$$

where all the composed transformations can be defined independently with completely unrelated parameter sets Ψ_k. The determinant of the Jacobian then distributes onto the Jacobians of the individual transformations according to the chain rule and determinant properties (Papamakarios et al., 2019; Kobyzev et al., 2019):

$$\begin{aligned}\det J_{f^{\psi}}(q_0) &= \det\left(J_{f^T_{\psi_T}}(q_{T-1}) \circ J_{f^{T-1}_{\psi_{T-1}}}(q_{T-2}) \circ \ldots \circ J_{f^2_{\psi_2}}(q_1) \circ J_{f^1_{\psi_1}}(q_0)\right) \\ &= \prod_{k=1}^{T} \det J_{f^k_{\psi_k}}(q_{k-1}).\end{aligned} \tag{3.8}$$

It is easy to see how this manifests in the density of the respective variational family members:

$$\begin{aligned}\log q^{\psi}(q_T) &= \log h(q_0) - \log\left|\det J_{f^{\psi}}(q_0)\right| \\ &= \log h(q_0) - \sum_{k=1}^{N} \log\left|\det J_{f^k_{\psi_k}}(q_{k-1})\right|,\end{aligned} \tag{3.9}$$

while samples are still trivially generated by applying all the component transformations successively.

3.2 Variational Bayesian Inference with Normalizing Flows

In the case of variational Bayesian inference the target distribution is the posterior $\pi_{\Theta|Y^N}$ on a parameter space $(\Theta = \mathbb{R}^d, \mathcal{B}_\Theta)$, with the usual notation for the prior distribution, likelihood function and the unnormalized posterior $\mu^D_\Theta(d\theta) = \mathcal{L}_D(\theta)\pi_\Theta(d\theta)$ defining the Bayesian model. A normalizing flow then is a parameterized measurable diffeomorphism $f^\psi : \Theta \longrightarrow \Theta$ with $\psi \in \Psi \subseteq \mathbb{R}^r$. Together with some base distribution η_Θ, a normalizing flow induces a variational family $\Gamma(\eta_\Theta, f^\psi)$. The optimization problem clearly is:

$$\gamma^\dagger_\Theta := \underset{\gamma_\Theta \in \Gamma}{\arg\min}\, \mathcal{F}(\gamma_\Theta) = \underset{\gamma_\Theta \in \Gamma}{\arg\min} -\mathbb{E}_{\gamma_\Theta}\left[\log \circ \frac{d\mu^D_\Theta}{d\gamma_\Theta}\right], \tag{3.10}$$

as derived in Section 2.2.

3.2.1 Variational Free Energy for Normalizing Flows

The specific structure of the variational family members $\gamma_\Theta^\psi = \eta_\Theta \circ (f^\psi)^{-1} \in \Gamma(\eta_\Theta, f^\psi)$ allows a more convenient reformulation of the variational free energy (Papamakarios et al., 2019):

$$\begin{aligned}
\mathcal{F}(\gamma_\Theta^\psi) &= -\mathbb{E}_{\gamma_\Theta^\psi}\left[\log \circ \frac{d\mu_\Theta^D}{d\gamma_\Theta^\psi}\right] \\
&= -\mathbb{E}_{\eta_\Theta}\left[\log \circ \frac{d\mu_\Theta^D}{d\gamma_\Theta^\psi} \circ f^\psi\right] \\
&= -\mathbb{E}_{\eta_\Theta}\left[\log \circ \frac{d(\mu_\Theta^D \circ f^\psi)}{d(\gamma_\Theta^\psi \circ f^\psi)}\right] \\
&= -\mathbb{E}_{\eta_\Theta}\left[\log \circ \frac{d(\mu_\Theta^D \circ f^\psi)}{d\eta_\Theta}\right] \\
&= -\mathbb{E}_{\eta_\Theta}\left[\log \circ \frac{d(\mu_\Theta^D \circ f^\psi)}{d(\lambda \circ f^\psi)} \frac{d(\lambda \circ f^\psi)}{d\lambda} \frac{d\lambda}{d\eta_\Theta}\right] \\
&= -\mathbb{E}_{\eta_\Theta}\left[\log \circ \frac{d\mu_\Theta^D}{d\lambda} \circ f^\psi + \log \circ \left|\det J_{f^\psi}(\,\cdot\,)\right|\right] - \mathcal{H}\left(\frac{d\eta_\Theta}{d\lambda}\right),
\end{aligned} \tag{3.11}$$

where λ is the Lebesgue measure on $(\Theta, \mathcal{B}_\Theta)$ and $\mu_\Theta^D, \gamma_\Theta^\psi$ are equivalent to λ. In this case the Radon-Nikodym derivative is equal to the ratio of the respective densities and the identity follows from the point-wise definition of ratios of functions and the properties of f^ψ. The final line can be interpreted as a function of only the variational parameters ψ instead of a functional of the variational family members, such that:

$$\mathcal{F}(\psi) = -\mathbb{E}_{\eta_\Theta}\left[\log \circ \frac{d\mu_\Theta^D}{d\lambda} \circ f^\psi + \log \circ \left|\det J_{f^\psi}(\,\cdot\,)\right|\right] - \mathcal{H}\left(\frac{d\eta_\Theta}{d\lambda}\right). \tag{3.12}$$

There are some further clarifications required:

i) The term J_{f^ψ} is a map $J_{f^\psi} : \Theta \longrightarrow \mathrm{Hom}(\Theta, \Theta)$ from the underlying space onto the set of linear maps on it. In particular, it returns for each $\theta_0 \in \Theta$ the linear approximation $J_{f^\psi}(\theta_0) : \Theta \longrightarrow \Theta$ to the transformation f^ψ at that point.

ii) The Radon-Nikodym derivative of the unnormalized posterior μ_Θ^D w.r.t. the base measure λ corresponds to the density $p(D, \theta)$ of the unnormalized posterior, which is accessible via the definition of the Bayesian model.

iii) The entropy $\mathcal{H}\left(\frac{d\eta_\Theta}{d\lambda}\right)$ obviously only depends on the choice of base distribution and is thus irrelevant for the optimization.

The optimization problem can then be rewritten as:

$$\begin{aligned}\psi^\dagger :&= \operatorname*{arg\,min}_{\psi\in\Psi} \mathcal{F}(\psi)\\ &= \operatorname*{arg\,min}_{\psi\in\Psi} -\mathbb{E}_{\eta_\Theta}\left[\log\circ\frac{d\mu_\Theta^D}{d\lambda}\circ f^\psi + \log\circ\left|\det J_{f^\psi}(\,\cdot\,)\right|\right],\end{aligned} \tag{3.13}$$

where the corresponding member of the variational family that is closest to the target distribution is constructed as $\gamma_\Theta^\dagger := \gamma_\Theta^{\psi^\dagger} = \eta_\Theta\circ(f^{\psi^\dagger})^{-1}$. Although this reformulation does not immediately seem to be advantageous, it resolves the dependency of the expectation on the variational parameters. This allows to use Monte Carlo sampling to approximate the expectation before computing the gradient of the variational free energy in gradient based optimization methods (Papamakarios et al., 2019):

$$\nabla_\psi\mathcal{F}(\psi) \approx -\frac{1}{N}\sum_{n=1}^{N}\nabla_\psi \log p(D, f^\psi(\theta_0^n)) + \nabla_\psi \log\left|\det J_{f^\psi}(\theta_0^n)\right| \quad \theta_0^n \sim \eta_\Theta. \tag{3.14}$$

Mohamed et al. (2019) call this a *pathwise gradient estimator*. They also discuss other possible gradient estimators for cases in which the expectation can not be uncoupled from the variational parameters via a reparameterization, which might be relevant for differently constructed types of variational families.

3.3 Practical Challenges

There are some practical challenges associated with constructing and applying normalizing flows. First, it is generally not trivial to construct appropriate parameterized diffeomorphisms, although even simple ones allow the construction of more complex transformations by composition.

The main challenge when constructing normalizing flows is to ensure bijectivity. By the structure of normalizing flows it is clear that the forward transformation is required for generating samples from a push-forward distribution, while evaluating their probability densities requires the inverse transformation. Although bijectivity has to be guaranteed, in some applications it is not necessary to be able to explicitly compute the inverse, which would be an even stronger requirement (Papamakarios et al., 2019).

The main problem regarding efficient application of normalizing flows is the computation of the Jacobian determinant. Even with sophisticated methods for computing the Jacobian, at least the determinant evaluation is problematic for high-dimensional underlying spaces. Much research in the area of normalizing flows therefore focuses on constructing flows that allow for efficient computation of the Jacobian determinant, while still having high expressive power. One example would be to ensure a triangular matrix representation of the Jacobian, such that the determinant is just the product of the diagonal elements (Papamakarios et al., 2019).

Both, Kobyzev et al. (2019) and Papamakarios et al. (2019), include reviews of different normalizing flow structures, highlighting benefits and drawbacks. This thesis will thus not get into any more detail in this regard, but focus on a particular kind of normalizing flows, called residual flows, which lead to the notion of continuous normalizing flows.

3.4 Continuous Normalizing Flows

This section discusses residual flows, their composition, and derives continuous normalizing flows as a limiting case. Moreover, benefits and drawbacks of continuous normalizing flows will be highlighted. Finally, augmented and volume preserving flows will be investigated, since they are fundamental to the idea of Hamiltonian normalizing flows, which are discussed in the next chapter.

3.4.1 Residual Flows

As before, consider a measurable space $(Q = \mathbb{R}^d, \mathcal{B}_Q)$, such that Q is a Euclidean vector space. *Residual flows* then are a particular type of normalizing flow, where the transformation $f^\psi : Q \longrightarrow Q$ is constructed as:

$$f^\psi(q_0) = q_0 + V^\psi(q_0) = q_T \quad q_0, q_T \in Q. \tag{3.15}$$

It is apparent that residual flows are defined by a family of vector fields $V^\psi : Q \longrightarrow Q$ that completely determine f^ψ. One approach that ensures bijectivity of the constructed transformation f^ψ, is to require V^ψ to be Lipschitz continuous with Lipschitz constant $L < 1$, such that it is a contractive map and the invertibility follows from the Banach fixed-point theorem (Papamakarios et al., 2019).

Intuitively the vector field V^ψ defines the difference vectors along which each point is translated due to the transformation f^ψ. Again, it is easy to construct more expressive flows by composition, such that:

$$f^\psi(q_0) = q_0 + \sum_{k=1}^{T} V_k^{\psi_k}(q_{k-1}) = q_T \quad \text{with} \quad q_k = q_{k-1} + V_k^{\psi_k}(q_{k-1}) \quad q_k \in Q. \tag{3.16}$$

This can the be interpreted as a path along which each point is translated through Q, or alternatively again a single translation defined by the sum of difference vectors.

The name residual flows was coined by Papamakarios et al. (2019) to highlight the strong similarity to residual neural networks, which were introduced by He et al. (2015).

3.4.2 Continuous Time Limit of Residual Flow Compositions

There is an interesting limiting case of compositions of residual flows, in which infinitely many differential translations are considered. To derive this case, first introduce a discrete time parameter

t with step size Δt to replace the index k in equation (3.16):

$$q_t = q_{t-\Delta t} + \Delta t\, V_t^{\psi_t}(q_{t-\Delta t}), \tag{3.17}$$

where the original form is recovered for $\Delta t = 1$. This allows to consider the continuous time limit, i.e. $\Delta t \to 0$, after rearrangement:

$$\lim_{\Delta t \to 0} \frac{q_t - q_{t-\Delta t}}{\Delta t} = \lim_{\Delta t \to 0} V_t^{\psi_t}(q_{t-\Delta t}) \qquad \text{s.t.} \qquad \frac{dq_t}{dt} = V_t^{\psi_t}(q_t). \tag{3.18}$$

The result is a differential equation describing the change of q_t over time. This has a convenient physics analogy, where q_t denotes the position of a particle at time $t \in \mathbb{R}$, which changes over time according to a time dependent velocity vector field V_t. Note that the solution to such an equation, for an initial state $q_0 \in Q$, is a trajectory with final state $q_T \in Q$ at time T.

3.4.3 Defining Continuous Normalizing Flows

To define so called *continuous normalizing flows*, also known as infinitesimal or continuous time flows (Papamakarios et al., 2019; Kobyzev et al., 2019), it is necessary for practical purposes to restrict to cases in which the variational parameters are time independent, that is $\psi_t = \psi \;\; \forall t \in \mathbb{R}$. Consider any differential equation:

$$\frac{dq_t}{dt} = V_t^{\psi}(q_t), \tag{3.19}$$

where for every $\psi \in \Psi$, the vector field V_t^{ψ} is continuous in t and uniformly Lipschitz continuous in q_t, i.e., there is a Lipschitz constant for all $t \in \mathbb{R}$. In this case, the equation is locally uniquely solvable for an initial $q_0 \in Q$, according to the Picard-Lindelöf theorem, and induces a smooth reversible flow on Q:

$$\phi_t^{\psi} : Q \longrightarrow Q, \qquad q_0 \longrightarrow q_0 + \int_0^t V_u^{\psi}(q_u)du = q_t. \tag{3.20}$$

This really is a flow in the mathematical sense, i.e., a group action of the additive group of the real numbers on Q with the properties described in (1.46), which can easily be verified. It is also effectively parameterized by the parameters of the vector field defining the differential equation and is thus a normalizing flow $f^{\psi} := \phi_T^{\psi}$ for some fixed evolution time $T \in \mathbb{R}$. Trivially, there is no need to worry about the invertibility of such normalizing flows and the computational costs are identical in both directions. This allows to make use of neural networks to define the derivative function, i.e., the vector field V_t^{ψ} (Papamakarios et al., 2019; Kobyzev et al., 2019). The general idea originated with Chen et al. (2018) under the name *Neural ODEs (NODE)*.

As already derived for general normalizing flows in equation (3.9), it is clear that the density of a variational family member $\gamma_Q^\psi \in \Gamma(\eta_Q, f^\psi)$ is given by:

$$\log q^\psi(q_T) = \log h(q_0) - \log\left|\det J_{f^\psi}(q_0)\right| . \tag{3.21}$$

Chen et al. (2018), however, in their paper 'Neural Ordinary Differential Equations', show that under a continuous flow ϕ_t^ψ, determining the trajectory q_t, the logarithmic density also follows the differential equation:

$$\frac{d \log q^\psi(q_t)}{dt} = -Tr\left\{J_{V_t^\psi}(\phi_t^\psi(q_0))\right\}, \tag{3.22}$$

where the initial density $q^\psi(q_0) := h(q_0)$ is the density of the base distribution η_Q and the final density $q^\psi(q_T)$ is that of the push-forward $\eta_Q \circ (f^\psi)^{-1}$. Furthermore, $Tr\{\,\cdot\,\}$ denotes the trace of a linear map. From this differential equation it is immediately clear that the density of the variational family member γ_Q^ψ can also be expressed as:

$$\log q^\psi(q_T) = \log h(q_0) - \int_0^T Tr\left\{J_{V_t^\psi}(\phi_t^\psi(q_0))\right\} dt, \tag{3.23}$$

such that for $f^\psi := \phi_T^\psi$ the identity:

$$\log\left|\det J_{f^\psi}(q_0)\right| = \int_0^T Tr\left\{J_{V_t^\psi}(\phi_t^\psi(q_0))\right\} dt \tag{3.24}$$

directly follows from equations (3.21) and (3.23).

In practice, this allows to simultaneously transform samples $q_0 \sim \eta_Q$ and compute their transformed logarithmic densities $\log q^\psi(q_T)$ via the combined integral (Papamakarios et al., 2019):

$$\begin{bmatrix} q_T \\ \log q^\psi(q_T) \end{bmatrix} = \begin{bmatrix} q_0 \\ \log h(q_0) \end{bmatrix} + \int_0^T \begin{bmatrix} V_t^\psi(q_t) \\ -Tr\left\{J_{V_t^\psi}(q_t)\right\} \end{bmatrix} dt. \tag{3.25}$$

With the usual notational conventions for variational Bayesian inference, this allows to rewrite the variational free energy and thus the optimization problem (3.13) as:

$$\begin{aligned} \psi^\dagger :&= \underset{\psi\in\Psi}{\arg\min}\, \mathcal{F}(\psi) \\ &= \underset{\psi\in\Psi}{\arg\min} -\mathbb{E}_{\eta_\Theta}\left[\log\circ\frac{d\mu_\Theta^D}{d\lambda}\circ f^\psi + \log\circ\left|\det J_{f^\psi}(\,\cdot\,)\right|\right] \\ &= \underset{\psi\in\Psi}{\arg\min} -\mathbb{E}_{\eta_\Theta}\left[\log\circ\frac{d\mu_\Theta^D}{d\lambda}\circ f^\psi + \int_0^T Tr\left\{J_{V_t^\psi}(\phi_t^\psi(\,\cdot\,))\right\} dt\right], \end{aligned} \tag{3.26}$$

such that the relevant gradient approximation for optimization takes the form:

$$\nabla_\psi \mathcal{F}(\psi) \approx - \sum_{n=1}^{N} \nabla_\psi \log p(D, f^\psi(\theta_0^n)) + \nabla_\psi \int_0^T Tr \left\{ J_{V_t^\psi}(\phi_t^\psi(\theta_0^n)) \right\} dt \qquad \theta_0^n \sim \eta_\Theta. \tag{3.27}$$

While the relevant gradients can be computed via automatic differentiation through the numerical integration, this is not particularly efficient. Chen et al. (2018) propose to use the so called *adjoint sensitivity method* instead, where the gradient is shown to follow another differential equation, such that it can be computed using a standard ODE solver. The details of this method are beyond the scope of this thesis.

Finally, it should also be noted that continuous normalizing flows can still be combined further into compositions of such flows. To do this, simply consider multiple differential equations, each of which induces a flow. For fixed integration times those flows can be combined via composition.

3.4.4 Benefits and Drawbacks

There are some obvious benefits provided by continuous normalizing flows. The main advantage is that the parameterized vector field $V_t^\psi(q_t)$, which defines the continuous normalizing flow $f^\psi := \phi_T^\psi$, has no hard constraints except for the continuity condition mentioned before. This allows for the use of, for example, neural networks to define this vector field. At the same time the structure of continuous normalizing flows allows for simple inversion of the flow (Papamakarios et al., 2019; Kobyzev et al., 2019). Moreover, the computation of the trace is much more efficient than that of the determinant.

The trace, on the other hand, has to be computed for every integration step and not just once. One should, however, not loose sight of the fact that a continuous normalizing flow corresponds to a composition of infinitely many residual flows and has an accordingly high expressive power. For an approximation of the integration with Euler's method, the continuous normalizing flow in fact breaks down into a composition of a large number of residual flows (Papamakarios et al., 2019).

Finally, the integrals involved generally increase the cost for the transformation of samples, the computation of transformed densities and the gradient required for optimization. Everything, therefore, comes down to a trade off between the increased computational costs on the one hand and the increase in expressive power plus the simple construction and inversion on the other hand.

3.5 Extensions to Continuous Normalizing Flows

There are two interesting extensions of continuous normalizing flows worth considering. First, it has been shown that continuous normalizing flows can represent any diffeomorphism when lifting the problem onto a higher dimensional space by adding auxiliary variables (H. Zhang et al., 2019). Accordingly, they are then capable of approximating any target distribution (Kobyzev et al., 2019).

This idea was introduced by Dupont et al. (2019), as *Augmented Neural ODEs (ANODE)*, which may define *augmented normalizing flows.*

Second, although the trace computation can be efficiently approximated using, e.g., the Hutchinson's trace estimator (Hutchinson, 1990), it would be useful to avoid those computations entirely. This is possible with what will be referred to as *continuous volume preserving flows.* Volume preserving flows are simply based on the idea of constructing normalizing flows with a unit Jacobian determinant or vanishing trace respectively. This is obviously also possible for normalizing flows more generally, not only for continuous normalizing flows (Rezende & Mohamed, 2015). Both approaches are introduced in the next two subsections.

3.5.1 Augmented Normalizing Flows

Intuitively, the idea behind *augmented normalizing flows* is that, if the relevant space is thought of as being embedded in a higher dimensional space, the flows are constrained to this subspace. If the evolution is, however, computed on the total space, by solving the corresponding augmented ODE, the flows become in general more flexible and can find more efficient solutions, because they can exploit the additional dimensions (Dupont et al., 2019; H. Zhang et al., 2019; Huang et al., 2020).

In practice, therefore, this amounts to defining a product space $Q \times A$, where $A = \mathbb{R}^p$ for some $p \in \mathbb{N}$, on which the augmented flows are defined. Note that $Q = \mathbb{R}^d$ is still assumed to be a Euclidean vector space. The ODE, augmented with $a_t \in A$, thus takes the form:

$$\frac{d}{dt}(q_t, a_t) = V_t^{\psi}(q_t, a_t). \tag{3.28}$$

This differential equation then induces the corresponding augmented flow:

$$\phi_t^{\psi} : Q \times A \longrightarrow Q \times A, \qquad (q_0, a_0) \longrightarrow (q_0, a_0) + \int_0^t V_u^{\psi}(q_u, a_u) du = (q_t, a_t). \tag{3.29}$$

Naive Augmented Normalizing Flows

For some fixed time T this flow can be used to define a normalizing flow on the base space Q. This requires that the initial values for the augmentation are always set to zero, such that the initial state is restricted to $Q \times \{0\} \subseteq Q \times A$. Moreover, by adding a corresponding term to the objective function of the optimization problem, one tries to enforce that $\phi_T^{\psi}(q_0, 0) = (q_T, 0) \;\; \forall q_0 \in Q$, i.e., the final state is also confined to the subspace $Q \times \{0\}$, which ensures invertibility (H. Zhang et al., 2019). This is usually only approximately possible, since it is subject to the optimization. Then a normalizing flow on Q can be defined as:

$$f^{\psi} : Q \longrightarrow Q, \qquad q_0 \longrightarrow (\omega_Q \circ \phi_T^{\psi})(q_0, 0) = q_T, \tag{3.30}$$

where $\omega_Q : Q \times A \longrightarrow Q$ is a projection onto the base space. This normalizing flow, together with a base distribution η_Q induces a variational family $\Gamma(\eta_Q, f^\psi)$. A corresponding objective function for the variational Bayesian inference problem with the unnormalized target measure μ_Q could look like:

$$\mathcal{F}_{\text{aug}}(\psi) = -\mathbb{E}_{\eta_Q}\left[\log \circ \frac{d\mu_Q}{d\lambda} \circ f^\psi + \int_0^T Tr\left\{J_{V_t^\psi}(\phi_t^\psi(\,\cdot\,, 0))\right\} dt + \left\|(\omega_A \circ \phi_T^\psi)(\,\cdot\,, 0)\right\|_A\right], \tag{3.31}$$

with the last term being a norm penalty on deviations of the final states of the flow ϕ_T^ψ from the subspace $Q \times \{0\}$. This approach is extremely restrictive and causes a number of problems, such that it cannot really be made rigorous. It can, however, be interpreted as a degenerate case of the following idea.

General Augmented Normalizing Flows

Instead of restricting the initial and final state of the flow on the augmented space $Q \times A$ to the subspace $Q \times \{0\}$, one could lift the whole problem onto the augmented space. To do this, assume $(A, \mathcal{B}_A)$ to be a measurable space, such that the augmented space is the product space $(Q \times A, \mathcal{B}_Q \otimes \mathcal{B}_A)$. Now introduce a fixed base distribution on this space as:

$$\eta_{Q\times A}(dq, da) = \eta_Q(dq)\eta_{A|Q}(da|q). \tag{3.32}$$

In this case, the augmented flow itself defines a normalizing flow $f^\psi := \phi_T^\psi$ for a fixed time T and induces a variational family $\Gamma(\eta_{Q\times A}, f^\psi)$ with members $\gamma_{Q\times A}^\psi = \eta_{Q\times A} \circ (f^\psi)^{-1}$ on the augmented space. The goal then would be to minimize the divergence between the marginals of the those members, i.e. $\gamma_{Q\times A}^\psi \circ \omega_Q^{-1}$, and the target distribution π_Q.

Because marginalization over the augmented space will typically be intractable, one could instead assume that the target distribution π_Q has a corresponding distribution $\pi_{Q\times A}$ on the augmented space, such that its disintegration w.r.t. the projection ω_Q takes the form:

$$\pi_{Q\times A}(dq, da) = \pi_Q(dq)\pi_{A|Q}(da|q). \tag{3.33}$$

Note that every choice of conditional distribution $\pi_{A|Q}$ induces a target distribution on the augmented space, such that the optimization problem can be lifted onto it. Equivalently, the conditional distribution lifts the corresponding unnormalized target measure μ_Q according to:

$$\mu_{Q\times A}(dq, da) = \mu_Q(dq)\pi_{A|Q}(da|q). \tag{3.34}$$

The goal then is to minimize the divergence between the variational family members and the lifted target distribution. Considering the reverse KL-divergence, as usual for the case of variational

Bayesian inference, the optimization problem can be formulated as:

$$\psi^{\dagger} := \underset{\psi\in\Psi}{\arg\min}\, \mathcal{D}_{KL}\left(\gamma^{\psi}_{Q\times A}||\pi_{Q\times A}\right). \tag{3.35}$$

As in equation (3.11), the augmented variational free energy then takes the form:

$$\begin{aligned}\mathcal{F}_{\text{aug}}(\psi) &= -\mathbb{E}_{\gamma^{\psi}_{Q\times A}}\left[\log\circ\frac{d\mu_{Q\times A}}{d\gamma^{\psi}_{Q\times A}}\right]\\ &= -\mathbb{E}_{\eta_{Q\times A}}\left[\log\circ\frac{d\mu_{Q\times A}}{d\lambda_{Q\times A}}\circ f^{\psi} + \int_0^T Tr\left\{J_{V^{\psi}_t}(\phi^{\psi}_t(\,\cdot\,))\right\}dt\right] - \mathcal{H}\left(\frac{d\eta_{Q\times A}}{d\lambda_{Q\times A}}\right).\end{aligned} \tag{3.36}$$

This is exactly analogous to the simple continuous normalizing flow case, except everything is happening on the augmented space. Since minimization of the KL-divergence is equivalent to the minimization of the augmented variational free energy, the optimization problem finally is:

$$\begin{aligned}\psi^{\dagger} :&= \underset{\psi\in\Psi}{\arg\min}\, \mathcal{D}_{KL}(\gamma^{\psi}_{Q\times A}||\pi_{Q\times A})\\ &= \underset{\psi\in\Psi}{\arg\min}\, \mathcal{F}_{\text{aug}}(\psi)\\ &= \underset{\psi\in\Psi}{\arg\min} -\mathbb{E}_{\eta_{Q\times A}}\left[\log\circ\frac{d\mu_{Q\times A}}{d\lambda_{Q\times A}}\circ f^{\psi} + \int_0^T Tr\left\{J_{V^{\psi}_t}(\phi^{\psi}_t(\,\cdot\,))\right\}dt\right].\end{aligned} \tag{3.37}$$

Deconstructing the Augmented Variational Free Energy

To further investigate the augmented variational free energy $\mathcal{F}_{\text{aug}}(\psi)$ and get a better intuition, consider now additionally the disintegration of the variational family members w.r.t. the projection onto the base space:

$$\gamma^{\psi}_{Q\times A}(dq, da) = \gamma^{\psi}_{Q}(dq)\gamma^{\psi}_{A|Q}(da|q). \tag{3.38}$$

Following (Salimans et al., 2014), the augmented variational free energy $\mathcal{F}_{\text{aug}}(\psi)$ can then be deconstructed according to:

$$\begin{aligned}
\mathcal{F}_{\text{aug}}(\psi) &= -\mathbb{E}_{\gamma^{\psi}_{Q\times A}}\left[\log\circ\frac{d\mu_{Q\times A}}{d\gamma^{\psi}_{Q\times A}}\right] \\
&= -\mathbb{E}_{\gamma^{\psi}_{Q\times A}}\left[\log\circ\left(\frac{d\mu_Q(dq)}{d\gamma^{\psi}_Q(dq)}\cdot\frac{d\pi_{A|Q}(da|q)}{d\gamma^{\psi}_{A|Q}(da|q)}\right)\right] \\
&= -\mathbb{E}_{\gamma^{\psi}_Q}\left[\mathbb{E}_{\gamma^{\psi}_{A|Q}}\left[\log\circ\frac{d\mu_Q(dq)}{d\gamma^{\psi}_Q(dq)}+\log\circ\frac{d\pi_{A|Q}(da|q)}{d\gamma^{\psi}_{A|Q}(da|q)}\right]\right] \\
&= -\mathbb{E}_{\gamma^{\psi}_Q}\left[\log\circ\frac{d\mu_Q(dq)}{d\gamma^{\psi}_Q(dq)}+\mathbb{E}_{\gamma^{\psi}_{A|Q}}\left[\log\circ\frac{d\pi_{A|Q}(da|q)}{d\gamma^{\psi}_{A|Q}(da|q)}\right]\right] \\
&= -\mathbb{E}_{\gamma^{\psi}_Q}\left[\log\circ\frac{d\mu_Q(dq)}{d\gamma^{\psi}_Q(dq)}\right]-\mathbb{E}_{\gamma^{\psi}_Q}\left[\mathbb{E}_{\gamma^{\psi}_{A|Q}}\left[\log\circ\frac{d\pi_{A|Q}(da|q)}{d\gamma^{\psi}_{A|Q}(da|q)}\right]\right] \\
&= \mathcal{F}(\psi)+\mathbb{E}_{\gamma^{\psi}_Q}\left[\mathcal{D}_{KL}\left(\gamma^{\psi}_{A|Q}||\pi_{A|Q}\right)\right].
\end{aligned} \tag{3.39}$$

Note that all of this assumes the existence of the relevant Radon-Nikodym derivatives, such that they correspond to the respective density ratios. The final line can be recognized as the variational free energy $\mathcal{F}(\psi)$ between the actual target distribution and the marginals of the variational family members, plus an expected KL-divergence term, which Huang et al. (2020) refer to as the *augmentation gap.* This augmentation gap is obviously non-negative and plays a role similar to the norm penalty in (3.31). The initially presented naive approach thus seems to be something like a limiting case of this more general version, in which the distributions $\pi_{A|Q}$ and $\eta_{A|Q}$ are chosen to be the Dirac probability measure on zero.

Sampling and Density Evaluation

Similar to the continuous normalizing flow case one can sample from the variational family members and evaluate densities simultaneously according to:

$$\begin{bmatrix}(q_T,a_T)\\ \log q^{\psi}(q_T,a_T)\end{bmatrix}=\begin{bmatrix}(q_0,a_0)\\ \log h(q_0,a_0)\end{bmatrix}+\int_0^T\begin{bmatrix}V^{\psi}_t(q_t,a_t)\\ -Tr\left\{J_{V^{\psi}_t}(q_t,a_t)\right\}\end{bmatrix}dt, \tag{3.40}$$

where $h(q_0,a_0)$ is the density of the lifted base distribution $\eta_{Q\times A}$ and $q^{\psi}(q_T,a_T)$ is the density of the variational family members. To obtain approximate samples from the actual target distribution

π_Q, one can simply discard a_T and keep only q_T from the generated samples. The density of the target distribution for the generated samples can be approximated as:

$$q^\psi(q_T) = \frac{q^\psi(q_T, a_T)}{p(a_T|q_T)}, \tag{3.41}$$

for $p(a_T|q_T)$ being the density of the chosen conditional distribution $\pi_{A|Q}$, which defined the lift of the target onto the augmented space. This is reasonable, because the optimization minimizes the augmentation gap, such that $q^\psi(a_T|q_T)$ will be close to $p(a_T|q_T)$. This approximation may be improved further by defining the lift of the target onto the augmented space as a variational family $\pi^\xi_{A|Q}$, via some family of density functions $p^\xi(a_T|q_T)$, and optimizing those parameters along with the others.

To approximate the density of an arbitrary sample $q_T \sim \pi_Q$, it is not possible to simply use the inverse flow and evaluate the density of the base distribution, because the lift onto the augmented space involves sampling. It thus has to be evaluated as an expectation:

$$q^\psi(q_T) = \mathbb{E}_{\pi_{A|Q}}\left[\frac{q^\psi(q_T, \cdot\,)}{p(\,\cdot\,|q_T)}\right] \approx \frac{1}{N}\sum_{n=1}^{N} \frac{q^\psi(q_T, a_T^n)}{p(a_T^n|q_T)} \qquad a_T^n \sim \pi_{A|Q}(da|q_T), \tag{3.42}$$

where $q^\psi(q_T, a_T^n)$ can now be evaluated using the inverse flow and the base distribution.

3.5.2 Volume Preserving Flows

Another interesting improvement of continuous normalizing flows would be to define *volume preserving flows*, for which the trace of the Jacobian vanishes, to spare even more computational costs. In general, note that the Jacobian determinant and Jacobian trace corrections ensure the volume preserving property of normalizing flows. If, however, a flow by construction is volume preserving, the computation of this correction term can be avoided. For continuous normalizing flows this is exactly the case, if the vector field V_t^ψ inducing the flow ϕ_t^ψ is divergence free:

$$Tr\left\{J_{V_t^\psi}(q_t)\right\} = \nabla \cdot V_t^\psi(q_t) = 0. \tag{3.43}$$

This is immediately clear from the definition of the divergence of vector fields. When inspecting equation (3.23), this implies that the density of the variational family members can trivially be evaluated as:

$$\begin{aligned} \log q^\psi(q_T) &= \log h(q_0) - \int_0^T Tr\left\{J_{V_t^\psi}(\phi_t^\psi(q_0))\right\} dt \\ &= \log h(q_0) - 0 \\ &= \log h(q_0). \end{aligned} \tag{3.44}$$

This can yield a significant reduction in computational costs, while restricting the expressiveness of corresponding normalizing flows. The main question then is how to construct a parameterized divergence free vector field.

One approach would be to note that the curl of any vector field is divergence free. Then a parameterized divergence free vector field $V^\psi = \nabla \times A^\psi$ can be defined as the curl of a parameterized vector potential A^ψ. The generalizations of the curl operator to more than three dimensions, however, let alone the corresponding computations, are not easily understood and go far beyond the scope of this thesis. It is not clear that this is a viable approach.

Hamiltonian normalizing flows provide a different path to defining volume preserving flows and will be introduced in the next chapter.

Chapter 4

Hamiltonian Normalizing Flows

This chapter introduces Hamiltonian normalizing flows, motivated by Toth et al. (2019). To understand the underlying idea, it is necessary to first introduce the Hamiltonian formalism of classical mechanics along with the notion of a phase space, a Hamiltonian and Hamiltonian flows on this space. Significant simplifications are necessary, since the mathematical details are far beyond the scope of this thesis. Finally, Hamiltonian flows can be used to define Hamiltonian normalizing flows as volume preserving augmented continuous normalizing flows, which is the main goal of this thesis.

4.1 The Hamiltonian Formalism

Beyond the well known Newtonian formulation of classical mechanics there are two relevant reformulations of the theory—the Lagrangian and the Hamiltonian formalism. Since this thesis avoids the discussion on the level of smooth manifolds and the associated language of differential geometry, the main reference for this section will be Goldstein et al. (2001), which is a standard text on classical mechanics. The next subsections will introduce some foundations regarding Newtonian and Lagrangian mechanics, before discussing the relevant notions of the Hamiltonian formalism.

4.1.1 From Newtonian to Lagrangian Mechanics

First, note that *Newton's equations of motion* describe the dynamics of a single particle of mass m in terms of a second order differential equation of the general form:

$$F\left(r, \dot{r}, t\right) = m\ddot{r}, \tag{4.1}$$

where t denotes time, the dots denote time derivatives and $r \in \mathbb{R}^3$ corresponds to the position of the particle in a Cartesian coordinate system.

In the Newtonian formalism, the dynamics of a particle, i.e., its trajectory $r(t)$ given some initial position $r(0)$ and velocity $\dot{r}(0)$, is determined by a force $F(r, \dot{r}, t)$. In particular, the equation itself expresses that the change of the particles momentum $p = m\dot{r}$ exactly corresponds to the force acting on it.

From another perspective, the force specifies a particular type of physical system. The harmonic oscillator in one dimension, for example, is exactly characterized by a force $F(x) = -kx$, where the force acting on the particle is proportional to its position with constant $-k$, such that $k \in \mathbb{R}^+$. Forces that are velocity independent, i.e. $F(r, t)$, can be considered time dependent vector fields and are referred to as *force fields*. For time independent force fields $F(r)$, those that can be derived as the negative gradient field of an underlying scalar function $U(r)$ are called *conservative*. For those forces, Newton's equations can be reformulated as:

$$F(r) = -\nabla_r U(r) \quad \Longrightarrow \quad -\nabla_r U(r) = m\ddot{r}. \tag{4.2}$$

In this case, a physical system is therefore determined by a scalar function $U(r)$, called the *potential energy*. Furthermore, the notion of the velocity dependent *kinetic energy* of the system is an important quantity defined as a scalar function:

$$T(\dot{r}) = \frac{1}{2} m \langle \dot{r}, \dot{r} \rangle. \tag{4.3}$$

Now consider a more general setting, where a system consists of N particles interacting with each other. Then there will be $3N$ coupled equations of motion, one for each dimension and particle, and the momentum of each particle changes according to the net force acting on it. There will usually also be constraining forces involved in any physical system, such that it has a reduced number of degrees of freedom in which it can actually change.

If a suitable coordinate system is chosen, then the so called configuration of the system can be described by $d \leq 3N$ independent *generalized coordinates* denoted $q = (q^1, q^2, ..., q^d) \in Q \subseteq \mathbb{R}^d$, where Q is called the *configuration space* of the system. The relationship between the old Cartesian coordinate components $r^1, r^2, ..., r^{3N}$ and new coordinates $q^1, q^2, ..., q^d$ can be summarized in transformation equations $r^i = f^i(q^1, q^2, ..., q^d, t)$. For the purposes of this thesis the configuration space will be considered equal to some $\mathbb{R}^d$, that is, all the coordinates are unconstrained real values. The time derivative of the generalized coordinates yields the *generalized velocities* $\dot{q} = (\dot{q}^1, \dot{q}^2, ..., \dot{q}^d) \in V \subseteq \mathbb{R}^d$, where V is called the velocity space. Again, for the purpose of this thesis, consider this velocity space to be equivalent to the same $\mathbb{R}^d$. With those simplifications, the so called *tangent bundle* TQ of the configuration space is in fact equivalent to a product space $TQ = Q \times V = \mathbb{R}^d \times \mathbb{R}^d$, together with the associated projection $\omega_Q : TQ \longrightarrow Q, \;\; (q, v) \longmapsto q$ onto the base space.

With the above notion, it is possible to derive generalized equations of motion, which are formulated in terms of a so called *Lagrangian*, a function $\mathcal{L} : TQ \times \mathbb{R} \longrightarrow \mathbb{R}$, which is the difference of the kinetic and potential energies $\mathcal{L}(q, \dot{q}, t) = T(q, \dot{q}, t) - U(q, \dot{q}, t)$. A Lagrangian completely specifies a physical system, although it is not unique—very similar to how potential energies are only determined up to an additive constant. Note that the kinetic and potential energy in terms of the generalized coordinates and velocities are obtained via the transformation functions f^i.

One approach to derive the generalized equations of motion for conservative systems, is to make use of Hamilton's principle and some variational calculus. *Hamilton's principle* states that a physical system, between times t_0 and t_1, takes a trajectory between two fixed configurations for which the action:

$$S[q(t)] = \int_{t_0}^{t_1} \mathcal{L}(q, \dot{q}, t) dt, \tag{4.4}$$

a functional of the possible trajectories through configuration space, is stationary. A trajectory is called a stationary state w.r.t. the action S, if the action to first order remains unchanged for small deviations from this trajectory. From variational calculus it is known that the stationary states of any functional are exactly the solutions to the *Euler-Lagrange equations*:

$$\frac{d}{dt} \frac{\partial \mathcal{L}(q, \dot{q}, t)}{\partial \dot{q}^i} = \frac{\partial \mathcal{L}(q, \dot{q}, t)}{q^i}, \qquad i = 1, 2, ..., d \tag{4.5}$$

which are a system of $d \leq 3N$ coupled second order differential equations. A benefit here is, that the equations are now reduced to a minimum by making use of independent coordinates, but also that these equations are linear in the Lagrangian, which allows to easily derive the equations of motion for more complicated composite systems.

4.1.2 Phase Space and the Hamiltonian

Starting from the Lagrange formalism, a *Legendre transform* of the Lagrangian in the velocity parameters yields:

$$p_i := \frac{\partial \mathcal{L}(q, \dot{q}, t)}{\partial \dot{q}^i} \qquad \dot{q} := \dot{q}(q, p, t) \tag{4.6}$$

$$\begin{aligned} \mathcal{H}(q, p, t) &= \sum_{i=1}^{d} \dot{q}^i \frac{\partial \mathcal{L}(q, \dot{q}, t)}{\partial \dot{q}^i} - \mathcal{L}(q, \dot{q}, t) \\ &= \sum_{i=1}^{d} \dot{q}^i(q, p, t)\, p_i - \mathcal{L}(q, \dot{q}(q, p, t), t). \end{aligned} \tag{4.7}$$

The parameters $p \in P = V^* = (\mathbb{R}^d)^*$ are called the *conjugate momenta* to the generalized coordinates $q \in Q = \mathbb{R}^d$ and are elements of the dual of the tangent space, at that point. They can be represented by row vectors, for which the components are denoted by subscript indices. Since the dual space $(\mathbb{R}^n)^*$ is equivalent to $\mathbb{R}^n$ itself, the distinction is not strictly necessary for the purposes of this thesis.

With the simplifications of the previous subsection, the so called cotangent bundle T^*Q of the configuration space is again equivalent to a product space $T^*Q = Q \times P = \mathbb{R}^d \times (\mathbb{R}^d)^*$, together with the associated projection $\omega_Q : T^*Q \longrightarrow Q, \ \ (q,p) \longmapsto q$ onto the base space. The space T^*Q deserves a special name and is referred to as the *phase space* of the system. In the Hamiltonian formalism, the two kinds of parameters p, q are now considered independent and on equal footing, as opposed to being related via a time derivative as was the case in the Lagrangian formulation.

The function $\mathcal{H} : T^*Q \times \mathbb{R} \longrightarrow \mathbb{R}$, resulting from the Legendre transform, is called the *Hamiltonian* of the system. Naturally, it contains the same information as the Lagrangian and thus completely specifies a physical system. In case the Hamiltonian is time independent, it is just a function on phase space and corresponds to the total energy function of the system. Note that time independence in this case only refers to the Hamiltonian not explicitly depending on time. Implicitly, it will always depend on time due to the configuration and momentum being functions of time. This thesis restricts to considering such time independent Hamiltonians, in particular those that are also separable:

$$H(q,p) = T(p) + U(q), \tag{4.8}$$

since they are sufficient for the purpose of Hamiltonian normalizing flows and have a convenient form for numerical integration (Toth et al., 2019).

4.1.3 Hamilton's Equations and Hamiltonian Flows

Hamilton's equations of motion can then be derived again using Hamilton's principle. First, review the action functional and substitute the Hamiltonian for the Lagrangian according to their relation due to the Legendre transform (4.7):

$$\begin{aligned} S[q(t), p(t)] &= \int_{t_0}^{t_1} \mathcal{L}(q,p,t)dt \\ &= \int_{t_0}^{t_1} \sum_{i=1}^{d} \dot{q}^i(q,p,t)p_i - \mathcal{H}(q,p,t)dt. \end{aligned} \tag{4.9}$$

For convenience, consider the Lagrangian as a function of the parameters q, p and their derivatives:

$$\mathcal{L}(q, \dot{q}, p, \dot{p}, t) = \sum_{i=1}^{d} \dot{q}^i p_i - \mathcal{H}(q,p,t). \tag{4.10}$$

The corresponding $2d$ Euler-Lagrange equations then take the form:

$$\begin{aligned} \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{p}_i} - \frac{\partial \mathcal{L}}{\partial p_i} &= 0 \quad i = 1, 2, ..., d \\ \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}^i} - \frac{\partial \mathcal{L}}{\partial q^i} &= 0 \quad i = 1, 2, ..., d. \end{aligned} \tag{4.11}$$

Computing the actual derivatives in those equations simplifies them to:

$$\begin{aligned} \frac{dq^i}{dt} &= \frac{\partial \mathcal{H}(q,p,t)}{\partial p_i} \qquad i = 1,2,...,d, \\ \frac{dp_i}{dt} &= -\frac{\partial \mathcal{H}(q,p,t)}{\partial q^i} \qquad i = 1,2,...,d. \end{aligned} \tag{4.12}$$

Those $2d$ first order differential equations are called *Hamilton's equations* of motion. The Hamiltonian formalism thus doubles the amount of equations, while decreasing the order. The solution to those equations yields the continuous time evolution of a system in phase space for a given initial state $s_0 = (q(0), p(0)) \in T^*Q = \mathbb{R}^d \times (\mathbb{R}^d)^*$.

Considering the restriction to time independent and separable Hamiltonians in the previous subsection, the final relevant form of the equations is:

$$\begin{aligned} \frac{dq^i}{dt} &= \frac{\partial T(p)}{\partial p_i} \qquad i = 1,2,...,d, \\ \frac{dp_i}{dt} &= -\frac{\partial U(q)}{\partial q^i} \qquad i = 1,2,...,d. \end{aligned} \tag{4.13}$$

For simplicity, the component indexing will be dropped from now on and it is implicitly assumed that all operations are done over all d components of q, p. Instead subscripts will be used to indicate the time variable for a more concise notation, that is, $(q_t, p_t) := (q(t), p(t))$ denotes configuration and momentum values at time t.

It is clear that Hamilton's equations in general (4.12), similar to any ODE, induce a flow on phase space according to:

$$\begin{aligned} &\phi_t^H : T^*Q \longrightarrow T^*Q, \\ &\qquad (q_0, p_0) \longmapsto (q_0, p_0) + \int_0^t \left(\frac{\partial \mathcal{H}(q_u, p_u, u)}{\partial p} \; , \; -\frac{\partial \mathcal{H}(q_u, p_u, u)}{\partial q} \right) du = (q_t, p_t), \end{aligned} \tag{4.14}$$

which is referred to as a *Hamiltonian flow.*

If the Hamiltonian is time independent, i.e. $\mathcal{H}(q,p)$, there are two interesting properties worth mentioning. First, the Hamiltonian is a conserved quantity over the evolution of the system and thus corresponds to the total energy of the system. This is shown by considering the time derivative of the Hamiltonian:

$$\begin{aligned} \frac{d\mathcal{H}(q,p)}{dt} &= \frac{\partial \mathcal{H}(q,p)}{\partial q}\frac{dq}{dt} + \frac{\partial \mathcal{H}(q,p)}{\partial p}\frac{dp}{dt} \\ &= \frac{\partial \mathcal{H}(q,p)}{\partial q}\frac{\partial \mathcal{H}(q,p)}{\partial p} - \frac{\partial \mathcal{H}(q,p)}{\partial p}\frac{\partial \mathcal{H}(q,p)}{\partial q} \\ &= 0. \end{aligned} \tag{4.15}$$

Since the time derivative vanishes, there are no changes in the Hamiltonian over time. Accordingly, this implies that the Hamiltonian is constant along the corresponding Hamiltonian flow:

$$\mathcal{H}(q_0, p_0) = \mathcal{H}(\phi_T^H(q_0, p_0)) = \mathcal{H}(q_T, p_T). \tag{4.16}$$

Secondly, the divergence of the vector field defined by Hamilton's equations vanishes, which is related to *Liouville's theorem*. This can be easily seen:

$$\nabla_{(q,p)} \cdot \left(\frac{\partial \mathcal{H}(q,p)}{\partial p} \; , \; -\frac{\partial \mathcal{H}(q,p)}{\partial q} \right) = \frac{\partial^2 \mathcal{H}(q,p)}{\partial q \partial p} - \frac{\partial^2 \mathcal{H}(q,p)}{\partial p \partial q} = 0, \tag{4.17}$$

due to the symmetry of second partial derivatives, that is, *Schwarz's theorem*, and the asymmetry in Hamilton's equations.

In summary, notice how this formalism allows to define a physical system via a scalar function $\mathcal{H}(q,p) = T(p) + U(q)$, the Hamiltonian, that induces a Hamiltonian flow ϕ_t^H on phase space $T^*Q = Q \times P = \mathbb{R} \times (\mathbb{R})^*$ according to Hamilton's equations, which is volume preserving by construction. In practice, so called *symplectic integrators* are used to numerically compute those flows. They have the special property of exactly preserving a slightly perturbed version of the Hamiltonian, which is of significance especially for computing flows over longer time intervals (Donnelly & Rogers, 2005).

4.2 Hamiltonian Flows as Normalizing Flows

Constructing Hamiltonian normalizing flows is now almost trivial. The straightforward solution to defining a parameterized Hamiltonian flow is simply via a parameterized Hamiltonian. As mentioned before, it is especially convenient to restrict to time independent and separable Hamiltonians:

$$\mathcal{H}^\psi(q,p) = T^\alpha(p) + U^\beta(q), \quad \text{with} \quad \psi = (\alpha, \beta) \in \Psi = \mathbb{R}^r. \tag{4.18}$$

Intuitively, this defines a family of Hamiltonian systems, that is, physical systems that exhibit varying behavior. If $T^\alpha(p)$ and $U^\beta(q)$ are differentiable, then $\mathcal{H}^\psi(q,p)$ induces a differentiable family of flows on phase space as in (4.14):

$$\begin{aligned} \phi_t^\psi &: T^*Q \longrightarrow T^*Q, \\ &(q_0, p_0) \longmapsto (q_0, p_0) + \int_0^t \left(\frac{\partial \mathcal{H}^\psi(q_u, p_u)}{\partial p} \; , \; -\frac{\partial \mathcal{H}^\psi(q_u, p_u)}{\partial q} \right) du = (q_t, p_t). \end{aligned} \tag{4.19}$$

For some fixed time T, this family of Hamiltonian flows defines a volume preserving continuous normalizing flow $f^\psi := \phi_T^\psi$, which will be referred to as a *Hamiltonian normalizing flow (HNF)*. For any initial state (q_0, p_0) this flow computes the evolution of the system $\mathcal{H}^\psi$ defined by the value of the parameters ψ.

4.2.1 Hamiltonian Normalizing Flows for Variational Inference

Recall that in variational inference there is an underlying measurable space $(Q, \mathcal{B}_Q)$ and some target distribution π_Q, which in variational Bayesian inference corresponds to the posterior on a parameter space.

Following Toth et al. (2019), it seems arbitrary to split the dimensions of this space into configuration and momentum parameters as to identify it with the phase space in the Hamiltonian formalism, if it has an even number of dimensions at all. Similar to the approach taken in the Hamiltonian Monte Carlo method it is instead useful to identify the space $(Q, \mathcal{B}_Q)$ with the configuration space, as the notation already indicates. In the Bayesian inference context the configuration space of the Hamiltonian formalism with this choice conveniently corresponds to some parameterization of the model configuration space.

The idea then is to introduce an auxiliary parameter for each configuration parameter, such that those auxiliary parameters correspond to dimensions of the momentum space $(P, \mathcal{B}_P)$ (Toth et al., 2019). Phase space is thus just the product space $(Q \times P,\ \mathcal{B}_Q \otimes \mathcal{B}_P)$. Moreover, let $\omega_Q : Q \times P \longrightarrow Q$ be a projection back onto configuration space and $\omega_P : Q \times P \longrightarrow P$ the corresponding projection onto momentum space.

HNFs then are just augmented normalizing flows, where the augmentation space is the momentum space. In contrast to general augmented normalizing flows, the additional parameters can, however, not evolve independent from the original ones. They are constrained by the preservation of the Hamiltonian along a flow. This, at least intuitively, makes HNFs less expressive (Toth et al., 2019). The interesting aspect is that this exact property is what makes HNFs volume preserving and thus more computationally efficient.

HNFs, therefore, combine the properties of augmented and volume preserving normalizing flows, such that the optimization problem for variational Bayesian inference simplifies to:

$$\begin{aligned}
\psi^\dagger :&= \underset{\psi \in \Psi}{\arg\min}\, \mathcal{D}_{KL}(\gamma^{\psi}_{Q\times P} || \pi_{Q\times P}) \\
&= \underset{\psi \in \Psi}{\arg\min}\, \mathcal{F}_{\text{aug}}(\psi) \\
&= \underset{\psi \in \Psi}{\arg\min} -\mathbb{E}_{\eta_{Q\times P}}\left[\log \circ \frac{d\mu_{Q\times P}}{d\lambda_{Q\times P}} \circ f^{\psi} + \int_0^T Tr\left\{J_{V_t^{\psi}}(\phi_t^{\psi}(\,\cdot\,))\right\} dt\right] \\
&= \underset{\psi \in \Psi}{\arg\min} -\mathbb{E}_{\eta_{Q\times P}}\left[\log \circ \frac{d\mu_{Q\times P}}{d\lambda_{Q\times P}} \circ f^{\psi}\right].
\end{aligned} \tag{4.20}$$

In summary, HNFs trade some additional cost for computing the corresponding transformation for less costs in the evaluation of the Jacobian trace correction of the transformation. More specifically, HNFs double the parameters involved and sacrifice expressive power to avoid this computation of the correction term entirely (Toth et al., 2019).

The question is, whether or not this trade off is beneficial in practice. At least in the case of density estimation, for which HNFs were initially introduced, they show performance comparable to *Real-valued Non-Volume Preserving (RNVP) flows* (Dinh et al., 2016) on a Gaussian mixture dataset, but are more computationally efficient (Toth et al., 2019). RNVP flows were used as a baseline, because they are capable of reproducing state of the art level image generation performance on various benchmark datasets and are thus highly expressive.

Chapter 5

Evaluation

The goal of this chapter is to introduce a possible implementation of Hamiltonian Normalizing flows for variational Bayesian inference and to evaluate qualitatively, whether or not they are capable of solving simple variational Bayesian inference problems. The objective is to provide a proof of concept for their applicability to Bayesian inference, not to definitively evaluate their performance. To do this, this chapter investigates the inference problem for two simple Bayesian models—a univariate Gaussian and a linear regression model.

5.1 Implementation

In the context of this thesis a Python software package for Bayesian inference is being developed on the basis of TensorFlow Probability[1] (TFP) (Dillon et al., 2017). It is used for the implementation of the experiments described in Section 5.2. Discussing the structure and documentation of the package in detail is beyond the scope of this thesis, but the respective GitHub repository[2] contains the documentation and example code.

In general, the package allows for a simple definition of Bayesian models using TFP distributions. It utilizes TFP bijectors, among other things, to allow for computations on an unconstrained parameter space that is equal to some $\mathbb{R}^d$, as was assumed in the theory of this thesis. The bijectors implement diffeomorphisms by specifying their forward and inverse transformations, as well as the respective Jacobian determinant corrections.

In addition to allowing for the flexible definition of Bayesian models, the package implements two general inference algorithms, Markov Chain Monte Carlo (MCMC) and variational Bayesian inference (VI). Every one of those algorithms at least requires the model and a dataset as an input, and implements a fit method that performs the parameter inference.

[1] url: `https://www.tensorflow.org/probability`

[2] url: `https://github.com/MaxGrtz/bayesian-inference/tree/thesis`

5.1.1 MCMC and VI Implementation

The MCMC algorithm is of a very simple structure. In addition to the Bayesian model and a dataset, it requires a *transition kernel*, e.g., the HMC kernel or variants thereof. The fit method basically is just a high level wrapper around the MCMC sampling function implemented by TFP. Given a number of chains, a number of samples and initial states for each chain, the method returns the sample results in a wrapper providing access to relevant statistics and diagnostics.

The VI algorithm, on the other hand, requires a so called *surrogate posterior* γ_Θ^ψ and a *discrepancy function* $d : \mathbb{R} \longrightarrow \mathbb{R}$. The surrogate posterior implements a variational family as a TFP distribution with trainable parameters. It is fitted to the true posterior distribution $\pi_{\Theta|Y^N}$ using gradient descent based optimization techniques. A discrepancy function d represents a convex function f in logarithm space, such that $d \circ \log = f$. Such functions can then in turn be used to define f-divergences (2.3):

$$D_f(\pi||\gamma) := \mathbb{E}_\gamma \left[f \circ \frac{d\pi}{d\gamma} \right] = \mathbb{E}_\gamma \left[d \circ \log \circ \frac{d\pi}{d\gamma} \right], \tag{5.1}$$

where π and γ are probability distributions. The reverse KL-divergence is recovered with the choice $d(u) = -u$. In the documentation of TFP, there is a simple proof provided for a general variational loss function derived from any f-divergence, which for the Bayesian inference problem takes the form:

$$\mathcal{F}(\psi) := \mathbb{E}_{\gamma_\Theta^\psi} \left[d \circ \log \circ \frac{d\mu_\Theta^D}{d\gamma_\Theta^\psi} \right], \tag{5.2}$$

where μ_Θ^D is the unnormalized posterior. It includes the variational free energy derived in this thesis as a special case. For computations it is convenient to introduce the Lebesgue base measure λ, such that the variational loss can be expressed in terms of densities:

$$\mathcal{F}(\psi) = \mathbb{E}_{\gamma_\Theta^\psi} \left[d \circ \log \circ \frac{d\mu_\Theta^D}{d\gamma_\Theta^\psi} \right] = \mathbb{E}_{\gamma_\Theta^\psi} \left[d \circ \log \circ \left(\frac{d\mu_\Theta^D}{d\lambda} \cdot \frac{d\lambda}{d\gamma_\Theta^\psi} \right) \right] = \mathbb{E}_{\gamma_\Theta^\psi} \left[d \circ (\log p(D, \cdot) - \log q^\psi(\cdot)) \right], \tag{5.3}$$

where $p(D, \cdot)$ and $q^\psi(\cdot)$ are the densities of μ_Θ^D and γ_Θ^ψ, respectively. Note that TFP not only implements various discrepancy functions, but also provides a general method for the corresponding optimization. The developed package, however, implements a custom fit method, additionally handling frequently occurring numerical problems and, for convenience, allowing for the display of a progress bar. Moreover, it provides a stochastic fit method, which is particularly useful for large datasets, implementing stochastic variational inference following Hoffman et al. (2013). In either case, the fit methods return a TFP distribution that approximates the true posterior and the history of the losses over the optimization.

5.1.2 ADVI and Normalizing Flows Implementation

As mentioned in the previous section, the VI algorithm requires the choice of a surrogate posterior. Using, for example, the Gaussian family to define a surrogate posterior leads to so called full rank *automatic differentiation variational inference* (ADVI). If the Gaussian family is restricted to have a diagonal covariance matrix instead, this implements mean field ADVI (Kucukelbir et al., 2016).

Surrogate posteriors based on normalizing flows are simply implemented as TFP transformed distributions, which consist of a base distribution and a bijector with trainable parameters. The density computations here automatically use the inverse transformation of the bijector, evaluate the log density of the base distribution and add the log Jacobian determinant correction term. It is trivial to define different normalizing flows by writing custom trainable bijectors, sub-classing the TFP bijector base class. Compositions of normalizing flows are easily implemented using the chain bijector provided by TFP, which wraps a list of bijectors and applies them sequentially. TFP even provides a bijector for defining continuous normalizing flows, called FFJORD (Grathwohl et al., 2018), which uses Hutchinson's trace estimator and implements the adjoint sensitivity method for the gradient computations.

For all of the above surrogate posteriors, and more generally for all those that allow for pathwise gradient estimators as described in equation (3.14), variational Bayesian inference can be implemented by Algorithm 1. Surrogate posteriors that are, in the language of TFP, not reparameterizable, i.e. do not allow for pathwise gradient estimators, require an appropriate adaptation of the gradient approximation.

Algorithm 1: General Variational Bayesian Inference

```
Require: dataset $D$ and unnormalized posterior $\mu_\Theta^D$ with density $p(D, \theta)$,
         surrogate posterior $\gamma_\Theta^\psi = \eta_{\Theta \times A} \circ (f^\psi)^{-1}$ with density $q^\psi(\theta)$,
         discrepancy function d,
         number of optimization steps L,
         sample size N,
         optimizer opt
for $L$ steps do
    $(\theta^n)_{n=1}^N \sim \gamma_\Theta^\psi$;   // draw N samples from surrogate posterior
    $\mathcal{F} \leftarrow \frac{1}{N} \sum_{n=1}^N d\left(\log p(D, \theta^n) - \log q^\psi(\theta^n)\right)$;   // approx. loss
    $grads \leftarrow \nabla_\psi \mathcal{F}$;   // compute gradients w.r.t. variational parameters
    $opt.apply\_gradients(grads, \psi)$;   // update variational parameters
end
Return : approximate posterior $\gamma_\Theta^\dagger$ with the final variational parameters $\psi^\dagger$
```

To implement augmented normalizing flows, the base distribution and the normalizing flow are defined on an augmented space. The unnormalized target measure is lifted onto this space, as described in Section 3.5.1, by a conditional distribution, which will be referred to as the posterior lift distribution. A general implementation of Bayesian inference with augmented normalizing flows thus is provided by Algorithm 2.

Algorithm 2: Variational Bayesian Inference with Augmented Normalizing Flows

```
Require: dataset $D$ and unnormalized posterior $\mu_\Theta^D$ with density $p(D,\theta)$,
         posterior lift distribution $\pi_{A|\Theta}$ with density $p(a|\theta)$,
         surrogate posterior $\gamma^\psi_{\Theta\times A} = \eta_{\Theta\times A} \circ (f^\psi)^{-1}$ with density $q^\psi(\theta,a)$,
         discrepancy function d,
         number of optimization steps L,
         sample size N,
         optimizer opt
for $L$ steps do
    $(\theta^n, a^n)_{n=1}^N \sim \gamma^\psi_{\Theta\times A}$;        // draw N samples from surrogate posterior
    $\mathcal{F} \leftarrow \frac{1}{N}\sum_{n=1}^N d\left(\log p(D,\theta^n) + \log p(a^n|\theta^n) - \log q^\psi(\theta^n,a^n)\right)$;        // approx. loss
    $grads \leftarrow \nabla_\psi \mathcal{F}$;        // compute gradients w.r.t. variational parameters
    $opt.apply\_gradients(grads, \psi)$;        // update variational parameters
end
$(\theta^n, a^n)_{n=1}^K \sim \gamma^\dagger_{\Theta\times A}$;        // draw K samples from trained surrogate posterior
Return : $empirical\_distribution((\theta^n)_{n=1}^K)$,
         approximating the marginal distribution over the model parameters
```

5.1.3 Hamiltonian Normalizing Flows Implementation

As discussed in the theory of this thesis, HNFs can be considered volume preserving augmented normalizing flows, where the number of dimensions of the augmentation space is equal to the number of dimensions of the parameter space. The augmented space then corresponds to phase space. A HNF finally is implemented as a bijector, which on construction is supplied trainable kinetic and potential energy functions, e.g. via neural networks, and a symplectic integrator, e.g. the leapfrog integrator, with information about the step size and number of integration steps. The forward and inverse transformations of this bijector are then simply the evolution of the system defined by the energy functions, forward and backwards in time, as computed by the symplectic integrator. In particular, the bijector just applies zero as the log Jacobian determinant correction, since it is by construction volume preserving.

The general implementation of the Hamiltonian flow bijector was verified by reproducing the results for density estimation reported by Toth et al. (2019). Their implementation is however only applicable in the density estimation context, which is quite different from the Bayesian inference case.

Algorithm 3 shows a possible implementation of variational Bayesian inference with HNFs for the particular case of choosing the variational loss derived from the reverse KL-divergence, which simplifies the computation significantly. It should be noted, however, that the actual implementation used for the experiments realizes the more general version shown in Algorithm 2, which allows for the choice of any discrepancy function, sacrificing some efficiency by computing an additional optimization irrelevant term in the KL-divergence case.

Algorithm 3: Bayesian Inference with Hamiltonian Normalizing Flows

```
Require: dataset $D$ and unnormalized posterior $\mu_\Theta^D$ with density $p(D, \theta)$,
         posterior lift distribution $\pi_{P|\Theta}$ with density $p(p|\theta)$,
         surrogate posterior $\gamma_{\Theta \times P}^{\psi} = \eta_{\Theta \times P} \circ (f^{\psi})^{-1}$ with density $q^{\psi}(\theta, p)$,
         where $f^{\psi}$ implements a HNF or a composition of them,
         number of optimization steps L,
         sample size N,
         optimizer opt

for L steps do
    $(\theta^n, p^n)_{n=1}^{N} \sim \gamma_{\Theta \times P}^{\psi}$;        // draw N samples from surrogate posterior
    $\mathcal{F} \leftarrow -\frac{1}{N} \sum_{n=1}^{N} \log p(D, \theta^n) + \log p(p^n|\theta^n)$;        // approx. loss
    $grads \leftarrow \nabla_{\psi} \mathcal{F}$;        // compute gradients w.r.t. variational parameters
    $opt.apply\_gradients(grads, \psi)$;        // update parameters
end
$(\theta^n, p^n)_{n=1}^{K} \sim \gamma_{\Theta \times P}^{\dagger}$;        // draw K samples from trained surrogate posterior
Return : $empirical\_distribution((\theta^n)_{n=1}^{K})$,
         approximating the marginal distribution over the model parameters
```

5.2 Experiments

To test the implementation of HNFs for Bayesian inference as described above, this section investigates two examples. The first will be a univariate Gaussian generative model. The second is a simple linear regression, to show that the algorithm works for regression models as well. The results will be compared to those of the Hamiltonian Monte Carlo method, which is assumed to yield the true posterior for such simple models and a sufficiently high number of samples per Markov chain. Additionally, the experiments will involve a comparison to other surrogate posteriors like mean field ADVI and various normalizing flows.

For each surrogate posterior the average runtime, final loss $\mathcal{F}$, posterior mean and standard deviation error, as well as the Wasserstein distance $\mathcal{W}_2$ of the posterior approximation to the true posterior, as provided by the HMC method, are reported.

In general, the Wasserstein distance $\mathcal{W}_p$ is a metric on the space of probability distribution over a metric space (Q, d). It is defined as:

$$\mathcal{W}_p(\pi, \gamma) := \left(\inf_{\rho_{Q\times Q} \in \Lambda(\pi,\gamma)} \int_{Q\times Q} d(q_1, q_2)^p \rho_{Q\times Q}(dq_1, dq_2) \right)^{\frac{1}{p}} \qquad \pi, \gamma \in \Pi(Q),\ p \geq 1 \in \mathbb{R}, \tag{5.4}$$

where $\Lambda(\pi, \gamma)$ denotes the set of all joint distribution over $Q \times Q$ with marginal distributions π, γ (Villani, 2009). In the present case Q is a Euclidean space, i.e. a vector space $Q = \mathbb{R}^d$ equipped with the standard inner product. This inner product not only induces a corresponding norm, but in turn a metric:

$$d_2(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}. \tag{5.5}$$

The Wasserstein metric $\mathcal{W}_2$ on the space of distributions, based on the metric d_2 of the underlying space, will be computed using the linear programming solution provided by the Python Optimal Transport (POT) package (Flamary & Courty, 2017).

Intuitively, think about the so called transport cost of transforming one distribution into the other according to a particular transport map, where the cost is determined by how much mass this map moves how far. The optimal transport map is then the one with minimal cost. The Wasserstein metric corresponds exactly to this minimal transport cost. As a side note, the whole theory of normalizing flows can be formulated in terms of optimal transport theory instead of taking the change of variable perspective used in this thesis (Papamakarios et al., 2019).

5.2.1 HMC Configuration

To be more specific, the experiments will use the state of the art HMC kernel with dynamic integration time to get a baseline solution for the posterior distribution. This variant is referred to as the No-U-Turn sampler (NUTS) and was proposed by Hoffman and Gelman (2011). The dual averaging step size adaptation, proposed in the same paper, is used to optimize the step size choice for the geometry of the given problem during the warm-up phase of the sampling process. The experiments will run 5 chains with $1,000$ samples each, with an additional warm-up phase of $10,000$ samples per chain. This will yield a total of $5,000$ samples to approximate and visualize the posterior distributions for each parameter.

5.2.2 VI Configurations

For all surrogate posteriors the variational loss will be based on the KL-divergence. The optimization is run for $L = 10,000$ iterations, using the Adam optimizer. The initial learning rate will be chosen for each surrogate posterior individually, balancing convergence speed and stability, and reported along with the results. On each iteration the sample size to approximate the variational loss is $N = 50$. The marginal posteriors for each parameter are approximated with $K = 10,000$ samples. Finally, those distributions can then be used to generate arbitrarily many posterior samples per parameter. The experiments will again use $5,000$ samples to visualize the posteriors and compute the defined error metrics.

ADVI and Normalizing Flow Configurations

A good baseline reference for variational inference solutions is provided by mean field ADVI. This surrogate posterior requires no further configuration. For all surrogate posteriors based on normalizing flows the base distribution is chosen to be a standard Gaussian. The experiments will consider an affine normalizing flow, which effectively yields a full rank ADVI, a masked autoregressive flow (Papamakarios et al., 2017) and a continuous normalizing flow. The masked autoregressive flow (maf) is defined by an autoregressive neural network with two hidden layers $[128, 128]$ and $tanh$ non-linearities, while the continuous normalizing flow (cnf) is defined by a FFJORD bijector, where the derivative function is defined by a neural network with layers $[d + 1, 128, 128, d]$ and $tanh$ non-linearities. Note that d corresponds to the dimension of the parameter space, while the additional input dimension effectively allows to model a time dependent derivative function. The FFJORD bijector uses a Dormand-Prince ODE solver (Shampine, 1986) with an initial step size of 0.1 and the trace of the Jacobian is approximated using Hutchinson's trace estimator.

Hamiltonian Normalizing Flow Configuration

The exact configuration of the HNFs for both experiments will be as follows. Every experiment will consider a single (hnf(1)), a composition of two (hnf(2)) and a composition of three (hnf(3)) HNFs. The posterior lift distribution is defined via a conditional diagonal Gaussian, where the location and scale depend on the parameters via neural networks with layers $[d, 128, 128, d]$ and $ReLu$ non-linearities for the hidden layers. Note that d again is the dimension of the parameter space. The network parameterizing the location has a linear output layer, while the scale network has a softplus output layer. The kinetic and potential energies are defined via neural networks with layers $[d, 128, 128, 1]$. The networks have $tanh$ non-linearities for the hidden layers and a softplus non-linearity at the output to enforce positivity of the energy functions. Note that $ReLu$ non-linearities would not be an option here, because the energy functions need to be twice continuously differentiable. This is the case, because the symplectic integration already uses first order derivatives, such that the gradient computation requires second order derivatives of the energy functions. The initial step size for the symplectic integration is set to 0.1, but will be optimized along with the variational parameters. The number of integration steps for each flow is fixed to 2.

The composition of multiple HNFs simulates something like a time dependence, similar to the extra time parameter in continuous normalizing flows. It will be investigated whether this has a positive effect on performance as measured by the defined error metrics. In general, note that the HNFs have considerably more parameters than all the other defined flows. This aims at compensating for the reduced expressive power.

5.2.3 Univariate Gaussian Model

Starting with the univariate Gaussian example, it is first necessary to create a test dataset. To test how reliable the inference results are, this experiment will consider 10 different datasets. Those will be generated by known distributions defined by Gaussian densities with different location and scale parameters μ^*, σ^*. For the evaluation of the inference results it is convenient to know these ground truth parameter values. Each dataset will consist of 100 examples, i.e. $D = (y_i)_{i=1}^{100}$ with $y_i \sim \mathcal{N}(\mu^*, \sigma^*)$. The next step is to define a full Bayesian model:

$$\begin{aligned} \mu &\sim \mathcal{N}(0, 10) \\ \sigma &\sim \text{Half-Normal}(10) \\ y_i &\sim \mathcal{N}(\mu, \sigma). \end{aligned} \tag{5.6}$$

Defining this model in the developed package is as simple as providing an ordered dictionary of the prior distributions for μ and σ and a likelihood function taking the parameters and returning a normal distribution. To allow computations on an unconstrained parameter space, it is necessary to define a list of bijectors, which unconstrain each parameter. If no list is provided, bijectors for each parameter are chosen based on the defaults defined by TFP for each distribution. The location parameter μ is already unconstrained, such that the corresponding bijector is simply the identity map. The scale parameter σ however is constrained to be positive, such that the default choice of bijector is the softplus transformation $s(x) = \log(1 + \exp(x))$. The softplus bijector takes any real number and returns a positive real number, such that its inverse transformation unconstrains the scale parameter to the complete real line. Note, that the exponential map would be a possible choice as well, but is much more prone to numerical instabilities.

The ground truth parameters for each test dataset will just be sampled from the prior distributions of the defined model and reported along with the inference results. Detailed overviews are provided in Appendix A.

5.2.4 Linear Regression Model

For this simple linear regression example, the experiment will consider again 10 datasets of 100 examples each, where every example is a tuple of features and targets, i.e. $D = (x_i, y_i)_{i=1}^{100}$. A regression model then tries to model the relationship between x and y. In this example we consider a linear relationship $y_i = \beta_0 + x_i\beta_1 + \epsilon_i$, where the ϵ_i capture the uncertainties and are assumed to be normally distributed with scale σ, i.e. $\epsilon_i \sim \mathcal{N}(0, \sigma)$. Each test dataset is generated by different ground truth parameters $\beta_0^*, \beta_1^*, \sigma^*$ and the full Bayesian model considered in this experiment can be summarized as:

$$\begin{aligned} \beta_0 &\sim \mathcal{N}(0, 10) \\ \beta_1 &\sim \mathcal{N}(0, 10) \\ \sigma &\sim \text{Half-Normal}(10) \\ y_i &\sim \mathcal{N}(\beta_0 + x_i\beta_1, \sigma), \end{aligned} \tag{5.7}$$

where the features are sampled from a Gaussian with a fixed scale, that is $x_i \sim \mathcal{N}(0, 10)$. As in the univariate Gaussian model, it is necessary to define the bijectors, which allow computations to be done on an unconstrained parameter space. It is obvious, that for β_0, β_1 the identity map is sufficient, whereas the scale σ can be unconstrained using the softplus bijector.

The ground truth parameters for each test dataset will again just be sampled from the prior distributions of the defined model and reported along with the inference results. Detailed overviews are provided in Appendix A.

5.3 Inference Results

This section summarizes the experiment results. The visualizations and detailed statistics are restricted to dataset 1, while the final summary tables aggregate the results of all datasets. The corresponding plots and tables for the remaining datasets can be found in the GitHub repository[3] associated with this thesis.

5.3.1 Univariate Gaussian Model - Inference Results

First, consider the visualization of dataset 1 in Figure 5.1. It shows a histogram and the estimated density of the dataset together with the true generating location parameter. More detailed information about the dataset is summarized in Table 5.1. The corresponding information about all the other datasets is available in Appendix A.

[3] url: `https://github.com/MaxGrtz/bayesian-inference/tree/thesis`

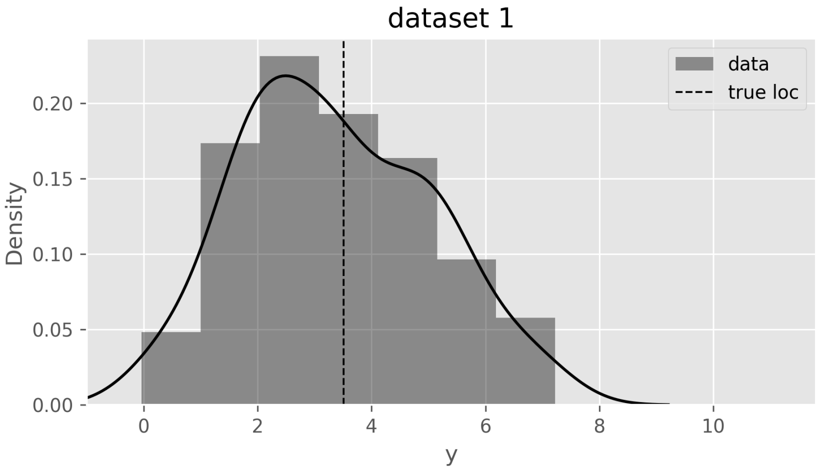

Figure 5.1: Histogram and density of the data with true location (univariate Gaussian dataset 1).

Table 5.1: Detailed summary of univariate Gaussian dataset 1.

	true		sample statistics				
dataset	loc	scale	# samples	mean	sd	HDI 3%	HDI 97%
dataset 1	3.51	1.66	100	3.37	1.65	0.42	6.64

The next paragraphs will summarize the inference results of the HMC method and the variational inference results for each surrogate posterior.

HMC Inference Results

The baseline solution to the Bayesian inference problem is provided by the HMC method. Figure 5.2 shows a plot of the sample trace and approximate posterior density for each parameter, while Table 5.2 and Table 5.3 summarize the inference results and relevant diagnostics to evaluate convergence of the generated Markov chains.

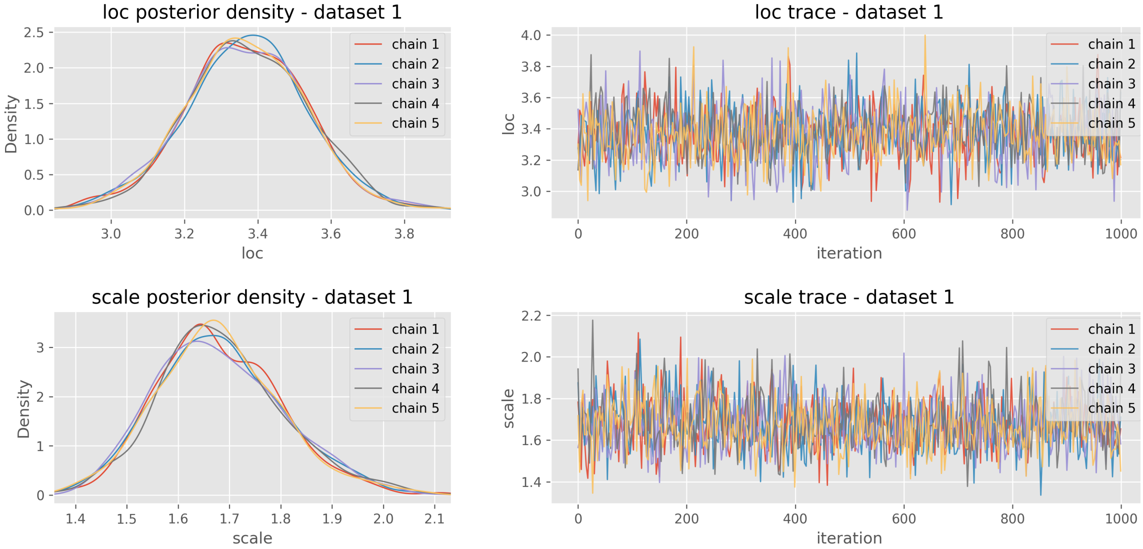

Figure 5.2: Density and trace plot for posterior parameter samples (univariate Gaussian dataset 1).

Table 5.2: HMC summary statistics and diagnostics (univariate Gaussian dataset 1).

	statistics					diagnostics		
param	mean	sd	HDI 3%	mode	HDI 97%	mcse-mean	ess-mean	$\hat{R}$
loc	3.37	0.16	3.05	3.37	3.67	0.00	3885.56	1.00
scale	1.68	0.12	1.47	1.67	1.94	0.00	4056.92	1.00

Table 5.3: HMC runtime and acceptance ratios per chain (univariate Gaussian dataset 1).

		acceptance ratio				
dataset	runtime [s]	chain 1	chain 2	chain 3	chain 4	chain 5
dataset 1	36.40	0.851	0.857	0.872	0.850	0.842

Table 5.3 shows high acceptance rates for the HMC proposed samples in each chain and a total runtime of 36.4 seconds. The statistics of Table 5.2 summarize the information about the estimated posterior distribution for each parameter, which are visualized in the density plots of Figure 5.2. They show that the posterior approximations contain the true parameter values in their highest density intervals and the posterior sample means are close to the true values. The diagnostics, in particular the *potential scale reduction factor* $\hat{R}$, computed according to Vehtari et al. (2019), clearly indicate the successful convergence of the generated Markov chains, which is corroborated visually by the trace plots of Figure 5.2. The *ess-mean* is the total estimated effective sample size

over all 5 chains, given the 5000 correlated samples generated. Intuitively, it measures the number of effectively independent samples, considering the autocorrelation of the Markov chains. Together with the *mcse-mean*, i.e. the Monte Carlo standard error, which is a measure for how close the posterior mean estimate is to the true posterior mean, this is a good indicator for the chains to have, in fact, converged to the true posterior. It is thus reasonable to assume that, at least for dataset 1, the generated posterior samples are representative of the true posterior distribution for both parameters.

Appendix B extends the above tables to all datasets. It is apparent from those summaries that the generated Markov chains on datasets 5 and 7 did not converge sufficiently well, such that the results for those datasets will be discarded, since they cannot provide a reliable baseline solution.

Variational Inference Results

To investigate the variational inference results for each surrogate posterior, it is useful to first consider the convergence behavior of the respective optimization processes. Figure 5.3 visualizes the variational loss history for each surrogate posterior on dataset 1 as a smoothed curve. The smoothing is done for the purpose of clarity, using an exponentially weighted moving average with smoothing factor $\alpha = 0.1$. The x-axis, representing the number of iterations, is logarithmically scaled, such that the early phase of the optimization is more clearly visible.

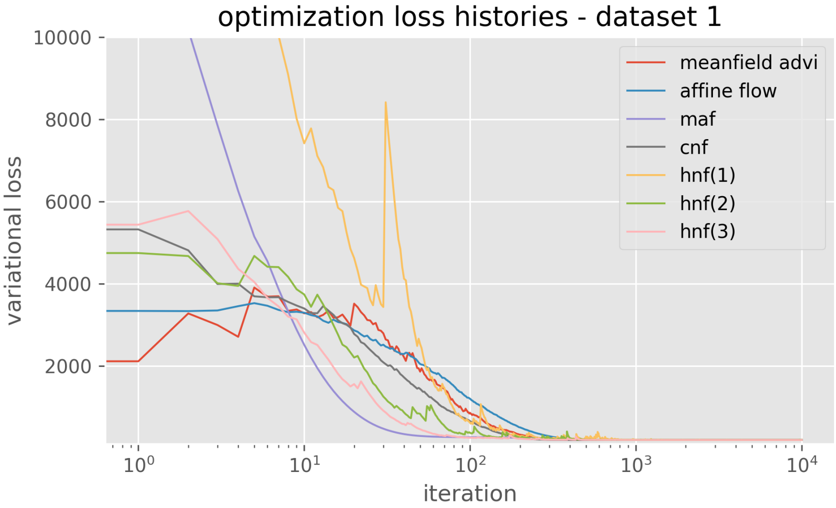

Figure 5.3: Variational loss history for each surrogate posterior (univariate Gaussian dataset 1).

The plot shows a stable optimization behavior and convergence for all surrogate posteriors on dataset 1. It is, however, clearly noticeable on other datasets, and in a non-smoothed version of the plot, that HNFs show a much less stable learning behavior than ADVI and the other normalizing flows (see Appendix C). Figure 5.4 visualizes the approximate posterior density according to each surrogate posterior, while Table 5.4 summarizes the posterior statistics and error metrics w.r.t. the true posterior, as provided by the HMC solution.

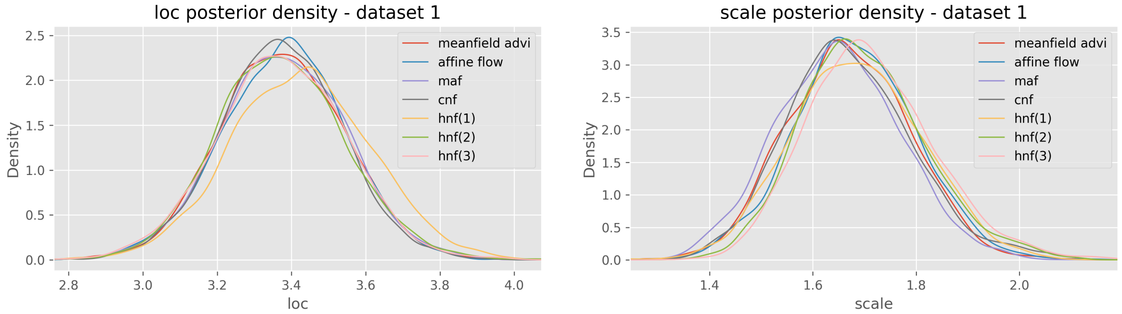

Figure 5.4: Posterior density estimates for each surrogate posterior (univariate Gaussian dataset 1).

Table 5.4: Statistics and error metrics for each surrogate posterior (univariate Gaussian dataset 1).

		statistics					error metrics		
param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
loc	meanfield advi	3.38	0.17	3.07	3.37	3.69	0.00616	0.00353	0.00907
	affine flow	3.38	0.17	3.06	3.38	3.68	0.01073	0.00219	0.01280
	maf	3.38	0.17	3.06	3.38	3.70	0.01099	0.00543	0.01299
	cnf	3.37	0.16	3.07	3.37	3.68	0.00499	0.00246	0.00697
	hnf(1)	3.42	0.19	3.08	3.43	3.78	0.05635	0.02307	0.06148
	hnf(2)	3.37	0.17	3.06	3.37	3.71	0.00415	0.01041	0.01410
	hnf(3)	3.37	0.17	3.04	3.37	3.69	0.00514	0.01028	0.01211
scale	meanfield advi	1.67	0.12	1.46	1.67	1.89	0.01218	0.00430	0.01625
	affine flow	1.68	0.12	1.46	1.68	1.90	0.00082	0.00489	0.01045
	maf	1.65	0.12	1.43	1.65	1.88	0.03180	0.00365	0.03348
	cnf	1.67	0.12	1.46	1.66	1.93	0.00949	0.00139	0.01046
	hnf(1)	1.69	0.13	1.46	1.68	1.92	0.00412	0.00300	0.01214
	hnf(2)	1.70	0.12	1.49	1.69	1.96	0.01700	0.00198	0.01756
	hnf(3)	1.71	0.12	1.50	1.70	1.97	0.03062	0.00113	0.03110

On dataset 1, all surrogate posteriors provided reasonably good approximations to the true posterior. The single Hamiltonian flow (hnf(1)) showed clearly worse performance for the location parameter's posterior approximation, when compared to the other surrogate posteriors. For the results on the remaining datasets refer to Appendix C.

Since the performance on a single dataset is not necessarily representative, a possible approach is to average the performance metrics over all datasets (excluding the discarded datasets 5 and

7) and parameters. This, however, renders the error metrics themselves non-interpretable due to the different scales of the parameter values involved. Only the comparison between the different surrogate posteriors remains meaningful. Note also that, with this approach, the datasets will have differing impact on the average performance metrics, depending on the magnitude of the parameter values. A better approach would be to compute a relative distance measure, that is independent of the scale of the values. This effect, however, is probably negligible for this experiment, since the results in Appendix C show that the error metrics are mostly in the same order of magnitude for all datasets (except for the discarded ones). Table 5.5 summarizes the average performance of each surrogate posterior.

Table 5.5: Average performance metrics for each surrogate posterior.

surr. posterior	learning rate	runtime [s]	final loss	mean err.	sd err.	$\mathcal{W}_2$
meanfield advi	5×10^{-3}	10.135	261.165	0.01453	0.01011	0.02919
affine flow	5×10^{-3}	13.468	261.165	0.01190	0.00890	0.02810
maf	5×10^{-3}	22.179	261.259	0.02709	0.02802	0.04726
cnf	1×10^{-4}	870.719	261.185	0.02807	0.00723	0.03527
hnf(1)	1×10^{-3}	57.935	262.489	0.10938	0.03661	0.12462
hnf(2)	1×10^{-3}	93.509	261.396	0.09005	0.01888	0.10394
hnf(3)	1×10^{-3}	132.161	261.364	0.07962	0.01573	0.08624

There are a few relevant observations to make. First, meanfield ADVI and the affine flow, which is equivalent to full rank ADVI, perform about equally well and better than all other surrogate posteriors. Due to the simple structure of the univariate Gaussian model, it is expected that they perform comparably well. They are also the most efficient approximations, with the lowest average runtimes.

HNFs show the worst performance as measured by the average posterior mean and standard deviation errors, as well as the Wasserstein distance to the HMC baseline solution. There is, however, a clearly noticeable performance improvement from a single HNF (hnf(1)) to a composition of three (hnf(3)). It is also interesting to note that even the composition of three Hamiltonian flows has a lower average runtime than the continuous normalizing flows (cnf), almost by a factor 7, although the Hamiltonian flows have much more parameters. This is probably due to the fact that Hamiltonian flows are restricted to two integration steps per flow with trainable step size. The computational costs are therefore fixed for each integration. The continuous normalizing flow on the other hand also uses an adaptive step size, but always integrates for a total time of 1, such that the integration cost depends on the step size. This makes it hard to meaningfully compare the runtimes of Hamiltonian and continuous normalizing flows, at least given the implementations used for the current experiments.

Posterior Predictive Check

Finally, another useful way of evaluating the inference results for models with such low dimensional target spaces, is to produce posterior predictive samples and visually inspect whether or not those samples are close to the underlying data. The predictive samples are generated by taking 1000 approximate posterior samples and generating a data sample for each one of them, based on the data distribution. Figure 5.5 shows the approximate posterior predictive densities for HMC and all variational inference results.

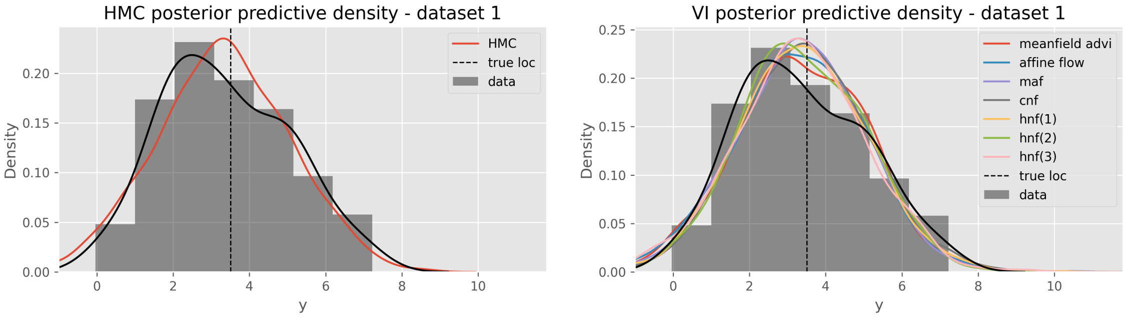

Figure 5.5: Posterior predictive check (univariate Gaussian dataset 1).

Visual inspection shows no clear differences between the predicted data based on the posterior estimate of HMC and all variational inference results. This allows at least to conclude that HNFs, as implemented, are generally capable of solving the Bayesian inference problem.

5.3.2 Linear Regression Model - Inference Results

For the second experiment, again consider the visualization of dataset 1 in Figure 5.6. It shows a scatter plot of the data, together with the true linear relationship between the variables x and y. More detailed information about the dataset is summarized in Table 5.6.

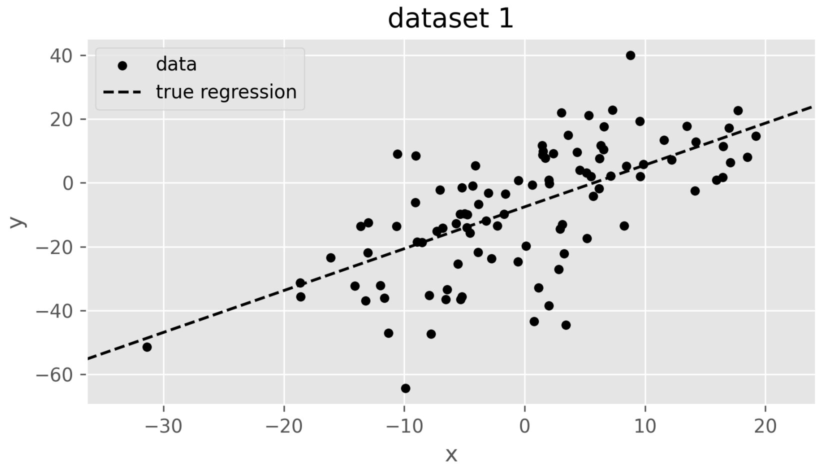

Figure 5.6: Scatter plot of the data with true regression line (linear regression dataset 1).

Table 5.6: Detailed information about linear regression dataset 1.

	true			sample statistics			
dataset	beta0	beta1	scale	# samples	$\widehat{\text{beta0}}$	$\widehat{\text{beta1}}$	$\widehat{\text{scale}}$
dataset 1	-7.53	1.31	15.03	100	-8.30	1.39	15.04

The sample statistics $\widehat{\text{beta0}}$ and $\widehat{\text{beta1}}$ in Table 5.6 are computed using standard linear regression[4] on the data. The estimate $\widehat{\text{scale}}$ is simply the standard deviation of the prediction error of this regression line w.r.t. the data. The corresponding tables for the remaining datasets can be found in Appendix A.

HMC Inference Results

The baseline solution to this Bayesian inference problem is again provided by the HMC method.

[4] using numpy.polyfit with degree 1

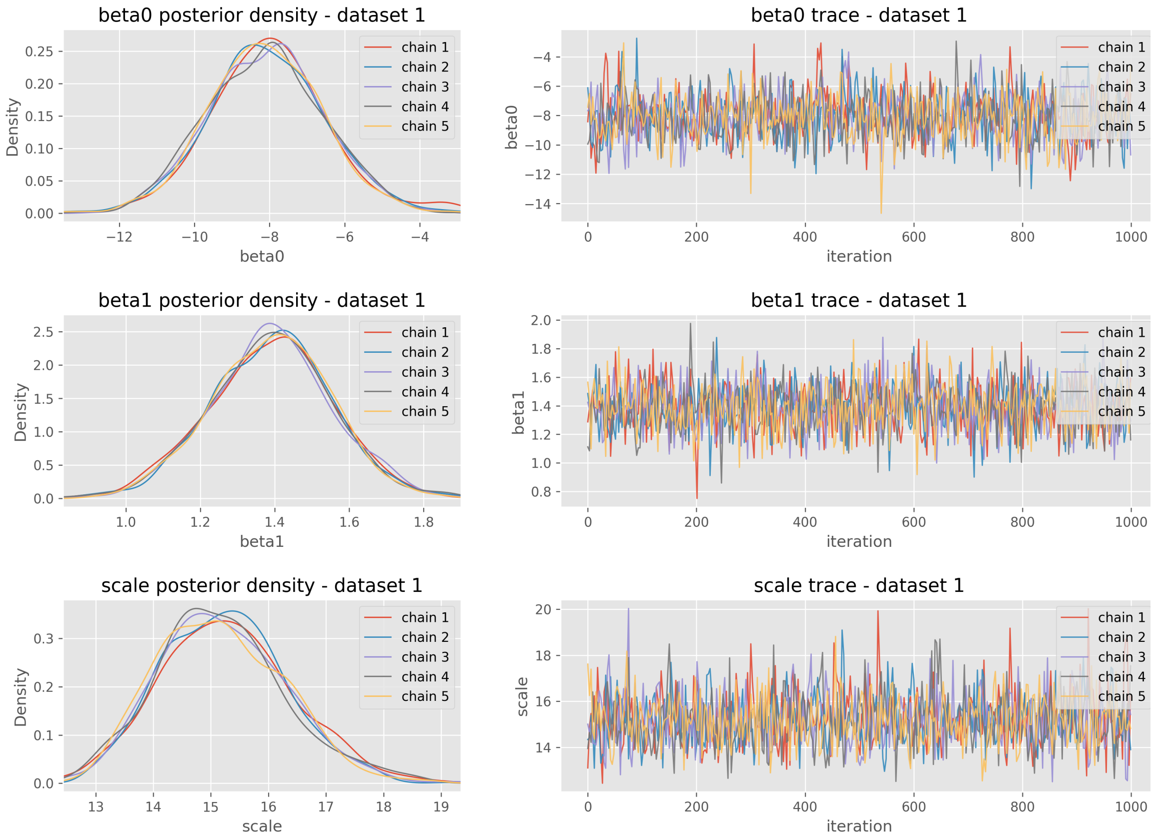

Figure 5.7: Density and trace plot for posterior parameter samples (linear regression dataset 1).

Figure 5.7 shows a plot for the posterior trace and density of each model parameter, while Table 5.7 and Table 5.8 summarizes the inference results and relevant diagnostics to evaluate convergence of the generated Markov chains.

Table 5.7: HMC summary statistics and diagnostics (linear regression dataset 1).

		statistics					diagnostics		
dataset	param	mean	sd	HDI 3%	mode	HDI 97%	mcse-mean	ess-mean	$\hat{R}$
dataset 1	beta0	-8.08	1.54	-10.92	-8.09	-5.11	0.04	1750.19	1.00
	beta1	1.39	0.16	1.08	1.40	1.70	0.00	4673.37	1.00
	scale	15.21	1.11	13.29	15.16	17.44	0.02	2285.59	1.00

Table 5.8: HMC runtime and acceptance ratios per chain (linear regression dataset 1).

		acceptance ratio				
dataset	runtime [s]	chain 1	chain 2	chain 3	chain 4	chain 5
dataset 1	203.48	0.939	0.956	0.945	0.957	0.952

Table 5.8 shows a total HMC runtime of 203.48 seconds, which is considerably higher than for the univariate Gaussian experiment. Table 5.7 again shows that the posterior approximations contain the true parameter values in their highest density intervals and the posterior sample means are close to the true values. Together with the diagnostics, this indicates convergence of the chains to the true posteriors, following the same reasoning as in the previous experiment.

Refer to Appendix B for the extension of the above tables to all datasets. Those summaries suggest that datasets 2 and 6 should be discarded, because of insufficient convergence of the Markov chains. The results for all other dataset are again used as baselines for an evaluation of the variational inference results.

Variational Inference Results

The variational inference optimization process for each surrogate posterior is visualized in Figure 5.8 by a smoothed loss curve. Again the curve is smoothed using an exponentially weighted moving average with the same smoothing factor $\alpha = 0.1$ as in the previous experiment. The less stable optimization behavior of HNFs mentioned before, is even more obvious in this experiment.

Figure 5.9 shows the posterior density plots for each surrogate posterior. Note that the plots for dataset 1 suggest that the posterior estimates for hnf(3) seem to diverge slightly from those of the other surrogate posteriors. This is also clearly noticeable in the statistics and error metrics summarized in Table 5.9.

Finally, Table 5.10 averages the performance of each surrogate posterior over all datasets (excluding the discarded datasets 2 and 6) and parameters. As in the previous experiment meanfield ADVI and the affine flow still obviously have not only the lowest runtimes, but also perform best according to the error metrics. While HNFs again show the worst performance, it is clear that the composition of multiple flows provides significant improvements. The runtime of hnf(3) is again lower than that of continuous normalizing flow (cnf) by a factor of 7.

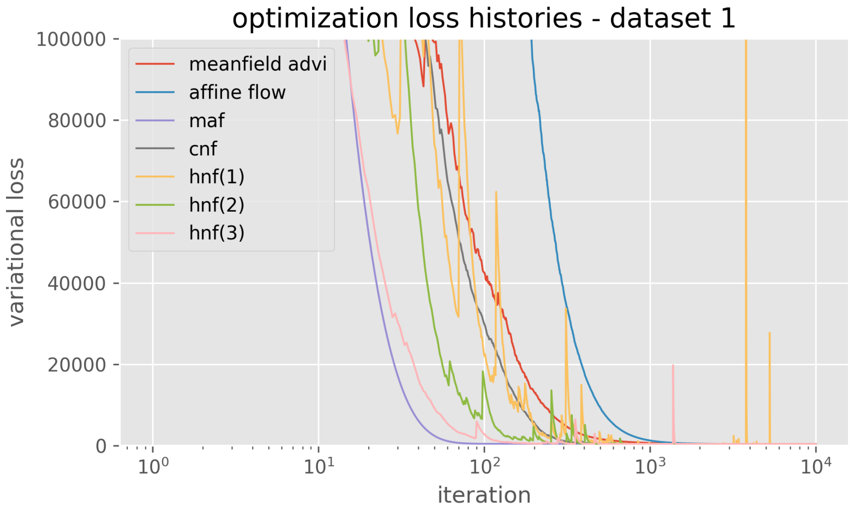

Figure 5.8: Variational loss history for each surrogate posterior (linear regression dataset 1).

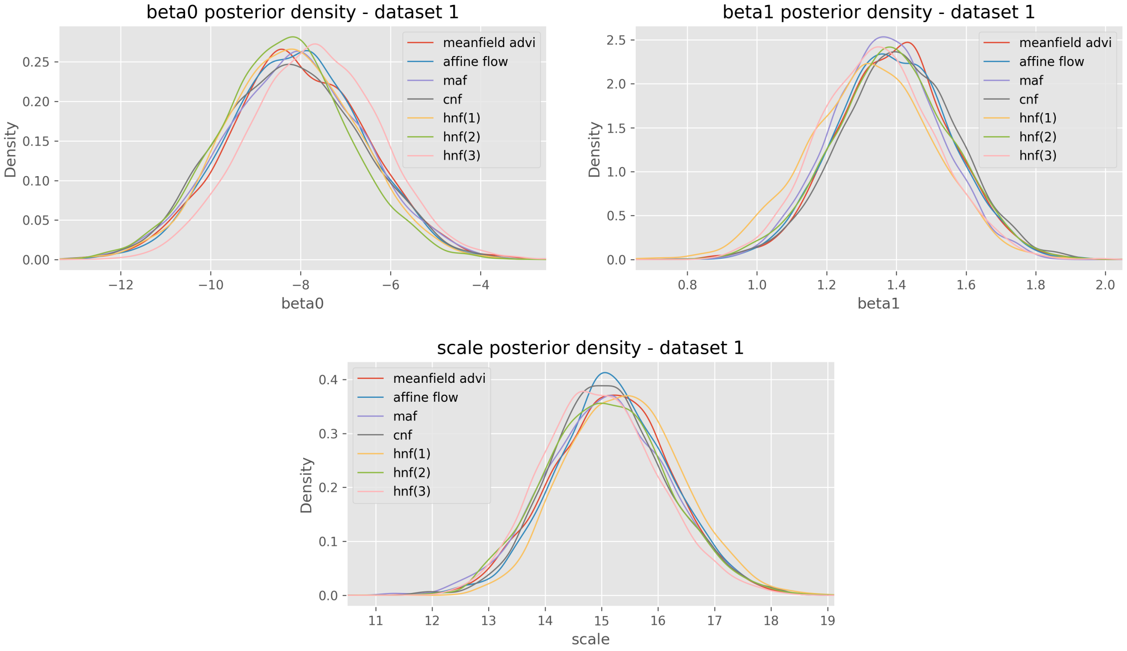

Figure 5.9: Posterior density estimates for each surrogate posterior (linear regression dataset 1).

Table 5.9: Statistics and error metrics for each surrogate posterior (linear regression dataset 1).

		statistics					error metrics		
param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
beta0	meanfield advi	-8.07	1.53	-10.93	-8.13	-5.16	0.00794	0.00925	0.06954
	affine flow	-8.10	1.47	-10.84	-8.10	-5.37	0.02449	0.07048	0.10737
	maf	-8.15	1.54	-11.07	-8.13	-5.20	0.07078	0.00179	0.11805
	cnf	-8.22	1.55	-11.02	-8.24	-5.27	0.14640	0.00993	0.17920
	hnf(1)	-8.20	1.46	-10.91	-8.21	-5.43	0.12297	0.07989	0.16314
	hnf(2)	-8.38	1.44	-11.14	-8.35	-5.62	0.30453	0.10555	0.33426
	hnf(3)	-7.70	1.45	-10.43	-7.69	-5.01	0.37770	0.08911	0.39520
beta1	meanfield advi	1.39	0.16	1.08	1.40	1.70	0.00379	0.00154	0.00624
	affine flow	1.39	0.16	1.09	1.39	1.70	0.00120	0.00055	0.00723
	maf	1.37	0.16	1.07	1.37	1.66	0.02273	0.00659	0.02452
	cnf	1.41	0.17	1.08	1.41	1.72	0.01669	0.00590	0.01854
	hnf(1)	1.32	0.18	0.98	1.33	1.65	0.06637	0.02195	0.07171
	hnf(2)	1.39	0.17	1.06	1.39	1.71	0.00136	0.00809	0.01042
	hnf(3)	1.35	0.16	1.05	1.34	1.67	0.04308	0.00190	0.04467
scale	meanfield advi	15.18	1.05	13.22	15.19	17.17	0.03200	0.05981	0.11950
	affine flow	15.22	1.02	13.36	15.19	17.18	0.00828	0.08827	0.12765
	maf	15.09	1.10	13.00	15.10	17.17	0.11895	0.00347	0.16111
	cnf	15.14	1.03	13.27	15.10	17.18	0.07328	0.07545	0.12484
	hnf(1)	15.40	1.07	13.60	15.39	17.38	0.19106	0.03727	0.30292
	hnf(2)	15.10	1.07	13.15	15.07	17.18	0.11467	0.04092	0.13654
	hnf(3)	14.98	1.02	13.18	14.96	17.00	0.22803	0.08990	0.25141

Table 5.10: Average performance metrics for each surrogate posterior.

surr. posterior	learning rate	runtime [s]	final loss	mean err.	sd err.	$\mathcal{W}_2$
meanfield advi	5×10^{-3}	14.542	374.279	0.01911	0.02077	0.05113
affine flow	5×10^{-3}	19.796	374.278	0.01626	0.02568	0.05282
maf	5×10^{-3}	31.312	374.670	0.05430	0.02523	0.07828
cnf	1×10^{-4}	1218.760	374.272	0.04846	0.01371	0.06782
hnf(1)	1×10^{-3}	71.550	379.370	0.13793	0.07234	0.18420
hnf(2)	1×10^{-3}	118.916	375.091	0.10836	0.04503	0.14793
hnf(3)	1×10^{-3}	160.656	374.820	0.09341	0.02792	0.10949

Posterior Predictive Check

Figure 5.10 shows standard regression lines computed on posterior predictive samples for each inference algorithm. The predictive samples are generated by taking 100 posterior samples and, for each one, producing 100 samples of target value predictions for each feature value of the dataset.

Although this choice of visualization does not capture the variance of the data around the linear relationship, it still is a good visual indication for the fact that all inference methods provide reliable solutions to this inference problem. In particular, in allows to conclude that HNFs are straightforwardly applicable to regression models as well.

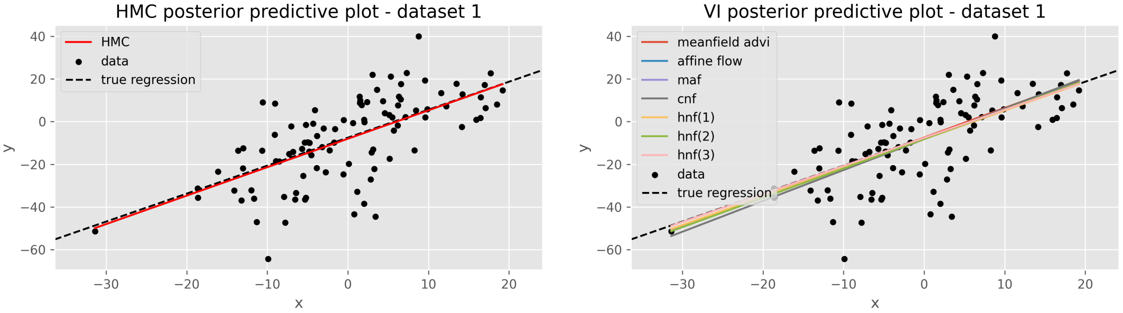

Figure 5.10: Posterior predictive check (linear regression dataset 1)

5.4 Discussion

The experiment results presented in the previous section conclusively show that HNFs, as described in this thesis, provide reliably good results for simple Bayesian inference problems. They are applicable for generative and regression models alike and are capable of very closely recovering the baseline solutions provided by the HMC method, at least according to the error metrics used for the performance evaluation. Although HNFs perform worse, in this respect, than the other surrogate posteriors, they show improved results for compositions of multiple flows. Besides the visual posterior predictive check, the quality of the inference results was measured primarily based on the Wasserstein distance W_2 of the posterior approximation to the baseline result of the HMC method.

A notable disadvantage of HNFs is the aforementioned less stable optimization behavior. Although this might be an artifact of the implementation, it can be seen in exploratory experiments, accessible in the associated GitHub repository, that they show a more stable optimization behavior for variational Bayesian inference on a univariate Gaussian mixture model. This is somewhat surprising, because mixture models are usually considered more complex due to having multiple modes. In particular, they pose a significant challenge to the HMC method (Betancourt et al., 2014). In attempts to apply Hamiltonian flows to generalized linear models with random effects, it was not possible to achieve convergence with any of the tested configurations. More research is therefore necessary to investigate their applicability to statistical models, which are more typically used in practice.

The results also show high runtimes for HNFs compared to other surrogate posteriors except for continuous normalizing flows. The HMC method had lower runtimes as well, when considering the fact that the reported time includes the unusually long warm-up phase of 10.000 samples per chain, which was only necessarily to provide sufficiently reliable baseline results. It is not clear whether or not the runtimes can be improved significantly with more efficient implementations. An advantage over the HMC method is only provided by the opportunity for applying stochastic optimization techniques in case of exceedingly large datasets. This is, however, not investigated further in this thesis.

Finally, it should be noted that the experiments, as they are designed, do not allow conclusive judgement about the performance and efficiency of any of the surrogate posteriors used for variational inference, because the configurations were arbitrarily chosen and no effort was made to assess the actual convergence speed. The primary objective of the experiments was to qualitatively evaluate whether or not HNFs are capable of solving the Bayesian inference problem for simple models.

Chapter 6

Conclusion

The goal of this thesis was to present the mathematical foundations of Hamiltonian normalizing flows on the level of measure theory and consider their application in the context of variational Bayesian inference.

Chapter 1 introduced the necessary mathematical notation and foundations of Bayesian inference. The primary realization was that the problem of fitting a Bayesian model to a dataset, can be reduced to the problem of generating posterior samples. This is true because one is mostly interested in evaluating expectations with respect to the posterior distribution, e.g., for the purpose of summarizing its structure or making predictions about new data, which is most efficiently done via Monte Carlo approximations. The chapter closed with a brief review of sampling approaches and the state of the art Hamiltonian Monte Carlo method for generating posterior samples.

Variational inference was discussed as an optimization based approach to the Bayesian inference problem in Chapter 2. In general, it relies on the idea of defining a variational family of distributions and minimizing a divergence measure to a target distribution. In particular, this is applicable to the Bayesian inference problem, where the target distribution corresponds to the posterior. Variational inference allows to scale Bayesian inference to large datasets by making use of stochastic optimization techniques.

Normalizing flows provide a means of defining rich variational families as all the push-forwards of some base distribution along a parameterized diffeomorphism. They were discussed in detail in Chapter 3, highlighting the application to variational Bayesian inference. Residual normalizing flows have a very specific structure, resembling that of residual neural networks. Considering the limit of infinitely many residual flows lead to the idea of continuous normalizing flows, which are defined by ordinary differential equations. This chapter concluded with two extensions or variants of continuous normalizing flows—augmented and volume preserving flows.

After outlining the Hamiltonian formalism of classical mechanics, Chapter 4 introduced Hamiltonian normalizing flows as volume preserving augmented continuous normalizing flows and discussed their application to variational Bayesian inference. They are fundamentally based on defining a family of Hamiltonian systems via a parameterized Hamiltonian, such that the associated Hamiltonian flows define diffeomorphisms on phase space, which induce a variational family given some base distribution.

Chapter 5 discussed the implementation and application of Hamiltonian normalizing flows to two particular Bayesian models—a univariate Gaussian and a simple linear regression model. In two experiments the Bayesian inference problem for both models was solved using the Hamiltonian Monte Carlo method to provide a baseline solution. The performance of variational inference with different variational families, i.e., surrogate posteriors, was investigated by comparing the inference results to those baseline solutions. The experiments were done using a Python software package for Bayesian inference based on TensorFlow Probability, which was developed in the context of this thesis. The experiment results clearly indicate that Hamiltonian normalizing flows yield reliably good solutions to the Bayesian inference problem for simple models. Although the composition of multiple Hamiltonian normalizing flows improves the quality of the results significantly, ADVI and other flow based surrogate posteriors still provide better results. Moreover, variational inference with the current implementation of Hamiltonian normalizing flows shows less stable optimization behavior than for other surrogate posteriors.

In conclusion, this thesis provides a more rigorous treatment of the theory of Hamiltonian normalizing flows than was provided by (Toth et al., 2019), who initially introduced them in the context of density estimation. The thesis successfully adapts their approach to allow for the application of Hamiltonian normalizing flows to Bayesian inference. More generally, the summarized theory and the developed software package provide a framework for flexibly defining and applying augmented normalizing flows to Bayesian inference, where Hamiltonian flows are just a special case. Additional effort is required, however, to improve on the current implementations of the software package.

Finally, to meaningfully assess the performance of Hamiltonian normalizing flows, more rigorous experiments with more complex models are required. In particular it is necessary to investigate how restrictive the volume preserving property of Hamiltonian flows is in practice—that is, how much expressiveness is sacrificed for computational efficiency. Intuitively, volume preserving flows only allow for permutations of the base density values over the underlying space. This is similar to what Papamakarios et al. (2019) observed for discrete flows. This leads to a tight coupling between the base and transformed densities. An interesting direction for future research thus might be to evaluate how defining the base distribution itself as a parameterized family of density functions influences performance and the stability of the optimization process. This is in fact equivalent to investigating the benefits of combining Hamiltonian normalizing flows with non-volume preserving flows.

Appendix A

Experiment Datasets

This appendix reports detailed information for every dataset generated in each experiments.

A.1 Univariate Gaussian Model Datasets

Table A.1 summarizes the true generating location and scale parameters as well as sample statistics for each dataset.

Table A.1: Univariate Gaussian dataset summaries.

	true		sample statistics				
dataset	loc	scale	# samples	mean	sd	HDI 3%	HDI 97%
dataset 1	3.51	1.66	100	3.37	1.65	0.42	6.64
dataset 2	-2.14	2.39	100	-2.03	2.45	-6.40	2.79
dataset 3	11.25	0.53	100	11.33	0.62	10.03	12.36
dataset 4	4.47	3.92	100	4.59	3.69	-2.59	10.29
dataset 5	-5.57	6.97	100	-5.81	7.62	-17.92	8.51
dataset 6	-17.89	11.14	100	-17.12	10.86	-34.69	2.05
dataset 7	4.93	5.56	100	5.14	5.47	-5.77	15.33
dataset 8	-6.42	4.13	100	-7.42	4.24	-15.94	0.02
dataset 9	14.19	1.82	100	13.94	1.82	10.69	17.67
dataset 10	0.82	11.23	100	-0.84	10.13	-19.63	15.15

A.2 Linear Regression Model Datasets

Table A.2 summarizes the true generating parameters as well as sample statistics for each dataset. The sample statistics $\widehat{\text{beta0}}$ and $\widehat{\text{beta1}}$ in Table 5.6 are computed using standard linear regression on the data. The estimate $\widehat{\text{scale}}$ is simply the standard deviation of the prediction error of this regression line w.r.t. the data.

Table A.2: Linear regression dataset summaries

	true			sample statistics			
dataset	beta0	beta1	scale	# samples	$\widehat{\text{beta0}}$	$\widehat{\text{beta1}}$	$\widehat{\text{scale}}$
dataset 1	-7.53	1.31	15.03	100	-8.30	1.39	15.04
dataset 2	16.77	-4.83	2.40	100	17.30	-4.85	2.69
dataset 3	0.30	-4.74	2.34	100	0.18	-4.75	2.11
dataset 4	6.48	14.41	8.92	100	5.13	14.30	9.32
dataset 5	15.75	2.53	15.65	100	17.61	2.64	16.11
dataset 6	5.50	-1.32	16.76	100	4.65	-1.31	16.96
dataset 7	-7.48	3.66	9.57	100	-8.21	3.61	8.52
dataset 8	13.85	-0.74	10.62	100	13.34	-0.83	11.55
dataset 9	0.24	-7.60	7.18	100	1.34	-7.63	6.94
dataset 10	-2.80	5.03	15.82	100	-2.69	5.07	15.78

Appendix B

HMC Inference Results

This appendix reports the HMC inference results for both experiments.

B.1 Univariate Gaussian Model - HMC Results

Table B.1 shows the total runtimes and acceptance ratios of HMC proposals for each dataset. The highlighted cells indicate problems, where the initial state never changed, that is all proposals were rejected. The same problem manifests in Table B.2, where the diagnostics suggest bad convergence behavior of the Markov chains for the respective datasets, at least when considering all 5 chains. Since the posterior distribution approximations for those datasets will be unreliable, they cannot be used as baseline results to which the variational inference solutions are compared. For this reason, datasets 5 and 7 are ignored in further evaluations.

Table B.1: HMC runtime in seconds and acceptance ratios per chain.

		acceptance ratio				
dataset	runtime [s]	chain 1	chain 2	chain 3	chain 4	chain 5
dataset 1	36.40	0.851	0.857	0.872	0.850	0.842
dataset 2	34.70	0.864	0.857	0.865	0.846	0.844
dataset 3	41.37	0.845	0.876	0.848	0.850	0.861
dataset 4	42.81	0.860	0.853	0.868	0.856	0.881
dataset 5	65.74	0.991	0.986	0.000	0.989	0.993
dataset 6	46.62	0.853	0.858	0.847	0.858	0.865
dataset 7	55.34	0.988	0.988	0.989	0.000	0.985
dataset 8	39.90	0.857	0.854	0.852	0.867	0.863
dataset 9	37.62	0.847	0.847	0.843	0.854	0.862
dataset 10	49.15	0.858	0.847	0.848	0.832	0.847

Table B.2: HMC posterior statistics and diagnostics.

		statistics					diagnostics		
dataset	param	mean	sd	HDI 3%	mode	HDI 97%	mcse-mean	ess-mean	$\hat{R}$
dataset 1	loc	3.37	0.16	3.05	3.37	3.67	0.00	3885.56	1.00
	scale	1.68	0.12	1.47	1.67	1.94	0.00	4056.92	1.00
dataset 2	loc	-2.03	0.25	-2.50	-2.03	-1.55	0.00	3695.61	1.00
	scale	2.49	0.18	2.18	2.48	2.85	0.00	3746.80	1.00
dataset 3	loc	11.33	0.06	11.21	11.33	11.45	0.00	3907.13	1.00
	scale	0.63	0.05	0.55	0.62	0.72	0.00	2680.51	1.00
dataset 4	loc	4.59	0.38	3.87	4.59	5.29	0.01	3546.16	1.00
	scale	3.76	0.27	3.28	3.74	4.31	0.00	3429.60	1.00
dataset 5	loc	-4.79	2.18	-7.17	-5.60	-0.66	0.92	5.57	1.42
	scale	6.27	2.98	0.39	7.55	8.80	1.31	5.18	1.42
dataset 6	loc	-16.93	1.07	-18.90	-16.95	-14.86	0.02	3189.86	1.00
	scale	10.97	0.77	9.63	10.92	12.51	0.01	3329.51	1.00
dataset 7	loc	3.93	2.49	-0.96	4.97	6.16	1.09	5.24	1.42
	scale	4.52	2.12	0.35	5.42	6.34	0.93	5.18	1.42
dataset 8	loc	-7.41	0.44	-8.21	-7.41	-6.58	0.01	3397.10	1.00
	scale	4.32	0.31	3.79	4.30	4.94	0.01	3416.06	1.00
dataset 9	loc	13.94	0.18	13.59	13.93	14.29	0.00	3684.96	1.00
	scale	1.85	0.13	1.62	1.84	2.11	0.00	3517.58	1.00
dataset 10	loc	-0.83	1.02	-2.73	-0.84	1.06	0.02	3133.42	1.00
	scale	10.24	0.74	8.98	10.18	11.70	0.01	3543.26	1.00

B.2 Linear Regression Model - HMC Results

Table B.4 shows the total runtimes and acceptance ratios of HMC proposals for each dataset, while Table B.3 contains the corresponding sampling result statistics and diagnostics. Following the same reasoning as before, datasets 2 and 6 are ignored in further evaluations.

Table B.3: HMC posterior statistics and diagnostics

		statistics					diagnostics		
dataset	param	mean	sd	HDI 3%	mode	HDI 97%	mcse-mean	ess-mean	$\hat{R}$
dataset 1	beta0	-8.08	1.54	-10.92	-8.09	-5.11	0.04	1750.19	1.00
	beta1	1.39	0.16	1.08	1.40	1.70	0.00	4673.37	1.00
	scale	15.21	1.11	13.29	15.16	17.44	0.02	2285.59	1.00
dataset 2	beta0	14.01	6.59	0.84	17.22	17.81	2.93	5.05	1.42
	beta1	-3.69	2.32	-4.90	-4.84	0.95	1.03	5.04	1.42
	scale	2.26	0.98	0.33	2.68	3.12	0.43	5.20	1.42
dataset 3	beta0	0.18	0.22	-0.24	0.18	0.59	0.01	1837.95	1.00
	beta1	-4.75	0.02	-4.79	-4.75	-4.71	0.00	4313.89	1.00
	scale	2.16	0.16	1.89	2.16	2.49	0.00	2517.17	1.00
dataset 4	beta0	5.08	0.93	3.29	5.09	6.85	0.02	1774.23	1.00
	beta1	14.30	0.09	14.14	14.30	14.46	0.00	4540.18	1.00
	scale	9.50	0.69	8.34	9.45	10.88	0.01	2502.84	1.00
dataset 5	beta0	17.17	1.64	14.11	17.15	20.34	0.04	2026.71	1.00
	beta1	2.65	0.19	2.29	2.65	3.01	0.00	4644.12	1.00
	scale	16.25	1.17	14.29	16.17	18.57	0.02	2694.28	1.00
dataset 6	beta0	3.66	2.27	0.23	3.98	7.49	0.77	8.71	1.38
	beta1	-1.10	0.45	-1.65	-1.24	-0.26	0.19	5.84	1.42
	scale	13.81	6.71	0.56	16.68	19.44	2.95	5.17	1.43
dataset 7	beta0	-8.12	0.87	-9.80	-8.11	-6.55	0.02	1936.37	1.00
	beta1	3.61	0.09	3.43	3.61	3.78	0.00	4154.23	1.00
	scale	8.66	0.63	7.58	8.61	9.96	0.01	2193.39	1.00
dataset 8	beta0	13.20	1.16	11.00	13.19	15.38	0.03	1968.01	1.00
	beta1	-0.83	0.12	-1.04	-0.83	-0.61	0.00	4522.25	1.00
	scale	11.74	0.86	10.28	11.67	13.53	0.02	2348.91	1.00
dataset 9	beta0	1.33	0.70	-0.02	1.34	2.66	0.02	1851.88	1.00
	beta1	-7.63	0.06	-7.75	-7.63	-7.51	0.00	4452.62	1.00
	scale	7.09	0.52	6.22	7.06	8.14	0.01	2272.63	1.00
dataset 10	beta0	-2.56	1.55	-5.38	-2.56	0.34	0.04	1948.46	1.00
	beta1	5.07	0.15	4.78	5.07	5.36	0.00	4575.32	1.00
	scale	15.93	1.13	14.04	15.88	18.24	0.02	2311.21	1.00

Table B.4: runtime and acceptance ratios per chain for each dataset

		acceptance ratio				
dataset	runtime [s]	chain 1	chain 2	chain 3	chain 4	chain 5
dataset 1	203.48	0.939	0.956	0.945	0.957	0.952
dataset 2	265.99	0.996	0.995	0.993	0.000	0.996
dataset 3	244.64	0.944	0.953	0.962	0.967	0.962
dataset 4	255.26	0.962	0.949	0.939	0.955	0.959
dataset 5	184.29	0.949	0.952	0.939	0.940	0.958
dataset 6	283.88	0.996	0.994	0.000	0.995	0.995
dataset 7	221.83	0.943	0.942	0.959	0.944	0.951
dataset 8	235.78	0.947	0.956	0.956	0.954	0.946
dataset 9	262.53	0.959	0.953	0.938	0.951	0.957
dataset 10	222.94	0.945	0.944	0.948	0.944	0.941

Appendix C

Variational Inference Results

This appendix reports the variational inference results for both experiments.

C.1 Univariate Gaussian Model - VI Results

First, Figure C.1 shows the optimization loss histories for every dataset. The curves are smoothed using an exponential moving average with smoothing factor $\alpha = 0.1$. Table C.1 summarizes the statistics of the posterior approximation and the corresponding error metrics for each surrogate posterior on every dataset.

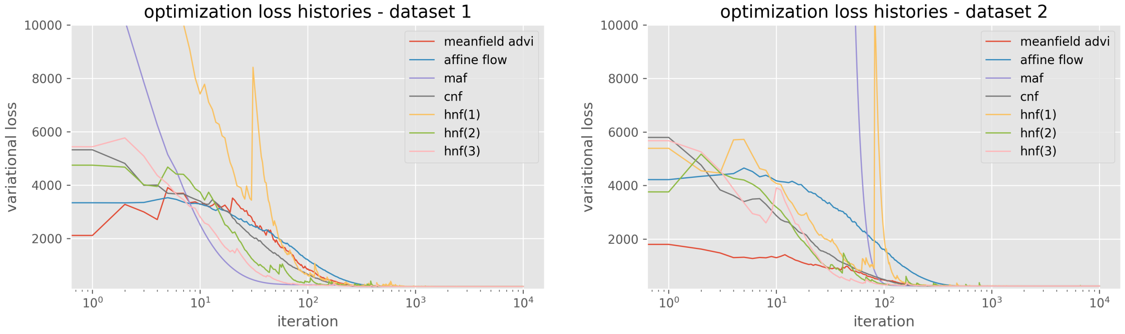

Figure C.1: Optimization loss histories (univariate Gaussian datasets).

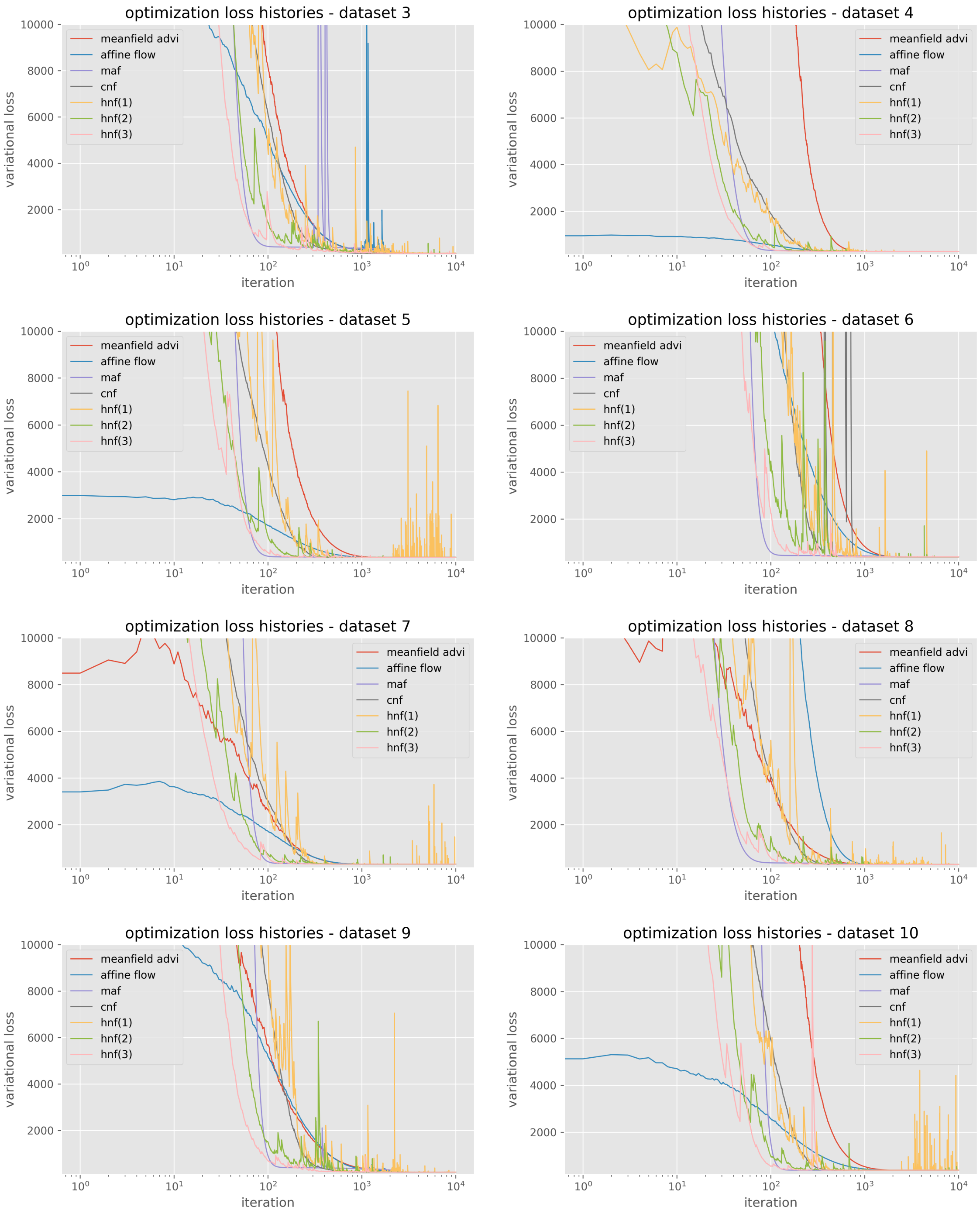

Figure C.1: Optimization loss histories (univariate Gaussian datasets).

Table C.1: Statistics and error metrics for each surrogate posterior (univariate Gaussian datasets).

			statistics					error metrics		
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 1	loc	meanfield advi	3.38	0.17	3.07	3.37	3.69	0.00616	0.00353	0.00907
		affine flow	3.38	0.17	3.06	3.38	3.68	0.01073	0.00219	0.01280
		maf	3.38	0.17	3.06	3.38	3.70	0.01099	0.00543	0.01299
		cnf	3.37	0.16	3.07	3.37	3.68	0.00499	0.00246	0.00697
		hnf(1)	3.42	0.19	3.08	3.43	3.78	0.05635	0.02307	0.06148
		hnf(2)	3.37	0.17	3.06	3.37	3.71	0.00415	0.01041	0.01410
		hnf(3)	3.37	0.17	3.04	3.37	3.69	0.00514	0.01028	0.01211
	scale	meanfield advi	1.67	0.12	1.46	1.67	1.89	0.01218	0.00430	0.01625
		affine flow	1.68	0.12	1.46	1.68	1.90	0.00082	0.00489	0.01045
		maf	1.65	0.12	1.43	1.65	1.88	0.03180	0.00365	0.03348
		cnf	1.67	0.12	1.46	1.66	1.93	0.00949	0.00139	0.01046
		hnf(1)	1.69	0.13	1.46	1.68	1.92	0.00412	0.00300	0.01214
		hnf(2)	1.70	0.12	1.49	1.69	1.96	0.01700	0.00198	0.01756
		hnf(3)	1.71	0.12	1.50	1.70	1.97	0.03062	0.00113	0.03110
dataset 2	loc	meanfield advi	-2.02	0.25	-2.48	-2.02	-1.55	0.00464	0.00323	0.00961
		affine flow	-2.02	0.25	-2.50	-2.01	-1.54	0.01189	0.00679	0.01575
		maf	-2.01	0.24	-2.47	-2.01	-1.56	0.01439	0.00243	0.01720
		cnf	-2.02	0.26	-2.48	-2.02	-1.52	0.00926	0.00967	0.01837
		hnf(1)	-1.91	0.25	-2.38	-1.92	-1.45	0.11248	0.00047	0.11341
		hnf(2)	-2.09	0.26	-2.58	-2.08	-1.59	0.06211	0.01840	0.07172
		hnf(3)	-2.00	0.25	-2.47	-2.00	-1.54	0.02242	0.00399	0.02513
	scale	meanfield advi	2.48	0.18	2.14	2.48	2.83	0.00783	0.00157	0.01792
		affine flow	2.49	0.17	2.17	2.49	2.83	0.00194	0.00797	0.01553
		maf	2.47	0.17	2.14	2.48	2.80	0.01656	0.00992	0.02623
		cnf	2.50	0.18	2.20	2.49	2.87	0.01258	0.00502	0.01578
		hnf(1)	2.57	0.19	2.22	2.57	2.95	0.08399	0.01251	0.08666
		hnf(2)	2.56	0.19	2.22	2.55	2.93	0.06712	0.00764	0.06820
		hnf(3)	2.48	0.19	2.14	2.47	2.87	0.00524	0.01326	0.01508
dataset 3	loc	meanfield advi	11.33	0.06	11.22	11.33	11.45	0.00097	0.00047	0.00141
		affine flow	11.33	0.06	11.22	11.33	11.45	0.00097	0.00136	0.00208
		maf	11.33	0.06	11.21	11.33	11.45	0.00231	0.00214	0.00296
		cnf	11.35	0.07	11.23	11.35	11.48	0.02149	0.00408	0.02220
		hnf(1)	11.58	0.32	11.17	11.54	12.17	0.24447	0.25489	0.35969
		hnf(2)	11.30	0.09	11.15	11.30	11.45	0.02991	0.02539	0.04470
		hnf(3)	11.33	0.07	11.20	11.33	11.47	0.00321	0.00992	0.01303
	scale	meanfield advi	0.63	0.04	0.55	0.63	0.72	0.00354	0.00099	0.00460
		affine flow	0.62	0.04	0.55	0.62	0.71	0.00102	0.00236	0.00370
		maf	0.64	0.06	0.53	0.63	0.74	0.00813	0.00921	0.01266
		cnf	0.63	0.05	0.55	0.62	0.73	0.00066	0.00013	0.00392
		hnf(1)	0.58	0.08	0.47	0.57	0.74	0.04185	0.03657	0.06001
		hnf(2)	0.66	0.17	0.56	0.65	0.77	0.03069	0.12591	0.16120
		hnf(3)	0.60	0.04	0.53	0.60	0.70	0.02116	0.00237	0.02170

Continued on next page

			statistics					error metrics		
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 4	loc	meanfield advi	4.59	0.37	3.89	4.60	5.27	0.00311	0.01138	0.01598
		affine flow	4.59	0.36	3.88	4.60	5.26	0.00098	0.01496	0.01813
		maf	4.59	0.36	3.90	4.59	5.25	0.00428	0.01559	0.02229
		cnf	4.58	0.37	3.90	4.58	5.28	0.01115	0.00998	0.02228
		hnf(1)	4.55	0.40	3.82	4.56	5.29	0.03965	0.01884	0.04578
		hnf(2)	4.62	0.39	3.90	4.61	5.36	0.02496	0.00913	0.03231
		hnf(3)	4.50	0.36	3.84	4.50	5.20	0.08665	0.01384	0.08940
	scale	meanfield advi	3.75	0.25	3.28	3.76	4.23	0.00158	0.02051	0.03092
		affine flow	3.76	0.26	3.28	3.75	4.22	0.00130	0.01393	0.02948
		maf	3.79	0.26	3.31	3.79	4.28	0.03042	0.01733	0.04047
		cnf	3.73	0.27	3.25	3.72	4.29	0.02228	0.00512	0.02495
		hnf(1)	3.84	0.27	3.36	3.84	4.38	0.08808	0.00273	0.08962
		hnf(2)	3.75	0.26	3.28	3.74	4.27	0.00669	0.01095	0.01722
		hnf(3)	3.64	0.27	3.18	3.63	4.17	0.11258	0.00741	0.11367
dataset 5	loc	meanfield advi	-5.74	0.76	-7.18	-5.74	-4.34	0.95447	1.42127	1.83072
		affine flow	-5.78	0.75	-7.21	-5.79	-4.37	0.99564	1.42877	1.85304
		maf	-5.76	0.75	-7.16	-5.75	-4.35	0.97233	1.43331	1.84796
		cnf	-5.74	0.76	-7.16	-5.73	-4.33	0.95248	1.42312	1.83106
		hnf(1)	-5.78	0.85	-7.39	-5.78	-4.19	0.98913	1.32835	1.79854
		hnf(2)	-5.96	0.78	-7.44	-5.97	-4.44	1.17117	1.40362	1.94231
		hnf(3)	-5.68	0.79	-7.17	-5.68	-4.19	0.88861	1.39164	1.77616
	scale	meanfield advi	7.74	0.54	6.75	7.73	8.80	1.46317	2.44011	2.95682
		affine flow	7.72	0.53	6.71	7.72	8.73	1.45145	2.45081	2.95576
		maf	7.77	0.53	6.78	7.77	8.77	1.49539	2.45403	2.98149
		cnf	7.69	0.56	6.68	7.67	8.80	1.42068	2.42638	2.93958
		hnf(1)	7.75	0.64	6.67	7.71	8.99	1.47552	2.34596	2.93827
		hnf(2)	7.76	0.52	6.83	7.72	8.83	1.48144	2.45832	2.99170
		hnf(3)	7.71	0.58	6.69	7.68	8.87	1.43632	2.39984	2.93024
dataset 6	loc	meanfield advi	-17.00	1.10	-19.06	-16.98	-14.94	0.06578	0.03301	0.08163
		affine flow	-16.90	1.08	-19.03	-16.87	-14.92	0.02659	0.01399	0.05924
		maf	-16.94	1.06	-18.89	-16.93	-14.92	0.01412	0.01402	0.03855
		cnf	-16.87	1.07	-18.86	-16.88	-14.88	0.05815	0.00070	0.06727
		hnf(1)	-17.04	1.03	-18.92	-17.05	-15.05	0.11252	0.03958	0.12527
		hnf(2)	-16.59	1.05	-18.57	-16.58	-14.61	0.34282	0.01926	0.34783
		hnf(3)	-17.29	1.05	-19.27	-17.28	-15.27	0.35751	0.02380	0.36127
	scale	meanfield advi	11.00	0.75	9.62	11.00	12.39	0.03362	0.02321	0.07176
		affine flow	11.00	0.77	9.57	11.00	12.43	0.03284	0.00366	0.07396
		maf	11.05	0.64	9.86	11.06	12.27	0.07599	0.13280	0.16408
		cnf	10.99	0.80	9.64	10.95	12.61	0.01799	0.03091	0.04530
		hnf(1)	10.89	0.78	9.53	10.83	12.50	0.07951	0.01382	0.08741
		hnf(2)	11.00	0.77	9.62	10.97	12.50	0.03329	0.00446	0.04371
		hnf(3)	10.89	0.77	9.51	10.88	12.37	0.08318	0.00217	0.08884

Continued on next page

			statistics					error metrics		
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 7	loc	meanfield advi	5.12	0.56	4.04	5.13	6.20	1.19964	1.92978	2.38087
		affine flow	5.13	0.55	4.10	5.12	6.13	1.20342	1.94993	2.39584
		maf	5.12	0.54	4.09	5.10	6.12	1.19051	1.95618	2.39378
		cnf	5.17	0.54	4.15	5.17	6.17	1.24435	1.95571	2.42223
		hnf(1)	5.10	0.56	4.09	5.07	6.20	1.16929	1.92966	2.38062
		hnf(2)	5.24	0.56	4.20	5.24	6.30	1.30984	1.93243	2.43977
		hnf(3)	5.22	0.54	4.18	5.22	6.21	1.29774	1.95441	2.44572
	scale	meanfield advi	5.57	0.40	4.81	5.57	6.32	1.04335	1.72200	2.09215
		affine flow	5.57	0.38	4.87	5.57	6.30	1.04884	1.73416	2.10583
		maf	5.45	0.38	4.71	5.44	6.14	0.92265	1.74147	2.05003
		cnf	5.61	0.41	4.88	5.59	6.43	1.08566	1.70553	2.11476
		hnf(1)	5.53	0.39	4.82	5.52	6.32	1.00118	1.72646	2.08166
		hnf(2)	5.63	0.40	4.93	5.61	6.44	1.10517	1.71578	2.13285
		hnf(3)	5.49	0.39	4.83	5.47	6.29	0.96517	1.72723	2.07538
dataset 8	loc	meanfield advi	-7.38	0.42	-8.16	-7.38	-6.56	0.03376	0.01702	0.04137
		affine flow	-7.38	0.42	-8.18	-7.38	-6.59	0.02494	0.01262	0.03168
		maf	-7.41	0.43	-8.22	-7.42	-6.62	0.00324	0.00611	0.01635
		cnf	-7.44	0.45	-8.26	-7.44	-6.56	0.02822	0.00992	0.03454
		hnf(1)	-7.42	0.45	-8.26	-7.41	-6.59	0.00894	0.00966	0.02539
		hnf(2)	-7.25	0.42	-8.02	-7.26	-6.43	0.15572	0.01421	0.15747
		hnf(3)	-7.40	0.47	-8.28	-7.41	-6.50	0.00386	0.03269	0.03634
	scale	meanfield advi	4.30	0.31	3.73	4.30	4.88	0.01281	0.00078	0.03030
		affine flow	4.30	0.30	3.73	4.30	4.86	0.01798	0.00691	0.03499
		maf	4.36	0.34	3.71	4.36	5.00	0.04038	0.03544	0.06201
		cnf	4.36	0.32	3.83	4.35	4.99	0.04716	0.00832	0.04923
		hnf(1)	4.20	0.32	3.61	4.19	4.84	0.12078	0.01717	0.12385
		hnf(2)	4.45	0.32	3.89	4.43	5.09	0.12850	0.01227	0.12984
		hnf(3)	4.39	0.33	3.80	4.38	5.04	0.07223	0.02397	0.07863
dataset 9	loc	meanfield advi	13.94	0.18	13.59	13.94	14.29	0.00278	0.00074	0.00441
		affine flow	13.93	0.18	13.59	13.92	14.26	0.01183	0.00547	0.01380
		maf	13.94	0.18	13.62	13.94	14.29	0.00413	0.00366	0.00753
		cnf	13.96	0.19	13.59	13.96	14.32	0.01980	0.00505	0.02435
		hnf(1)	14.19	0.26	13.70	14.19	14.67	0.24982	0.07751	0.26271
		hnf(2)	13.84	0.22	13.43	13.85	14.23	0.09332	0.03261	0.10247
		hnf(3)	13.86	0.21	13.47	13.86	14.26	0.07889	0.02700	0.08427
	scale	meanfield advi	1.84	0.13	1.60	1.84	2.08	0.00652	0.00588	0.01323
		affine flow	1.84	0.13	1.60	1.84	2.09	0.00595	0.00333	0.01237
		maf	1.78	0.17	1.48	1.78	2.10	0.06698	0.03539	0.07658
		cnf	1.85	0.14	1.63	1.84	2.15	0.00596	0.00555	0.01078
		hnf(1)	1.86	0.15	1.58	1.86	2.14	0.01024	0.01587	0.02193
		hnf(2)	1.89	0.13	1.63	1.88	2.13	0.03756	0.00205	0.03907
		hnf(3)	1.80	0.14	1.55	1.79	2.06	0.05059	0.00339	0.05146

Continued on next page

			statistics					error metrics		
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 10	loc	meanfield advi	-0.83	1.02	-2.79	-0.82	1.09	0.00699	0.00176	0.03653
		affine flow	-0.81	1.03	-2.74	-0.83	1.13	0.02550	0.01036	0.04133
		maf	-0.84	1.05	-2.86	-0.84	1.11	0.01082	0.03193	0.04427
		cnf	-0.88	1.03	-2.78	-0.91	1.17	0.04260	0.00888	0.06672
		hnf(1)	-0.70	1.02	-2.63	-0.69	1.25	0.13792	0.00305	0.14399
		hnf(2)	-0.57	1.01	-2.48	-0.58	1.28	0.25806	0.00735	0.26119
		hnf(3)	-0.64	0.98	-2.52	-0.64	1.21	0.19068	0.03791	0.19806
	scale	meanfield advi	10.20	0.70	8.85	10.20	11.58	0.03020	0.03331	0.08199
		affine flow	10.25	0.71	8.92	10.23	11.59	0.01519	0.03168	0.07438
		maf	10.14	0.86	8.46	10.15	11.75	0.09883	0.12320	0.17849
		cnf	10.10	0.73	8.80	10.06	11.55	0.13736	0.00854	0.14122
		hnf(1)	9.88	0.80	8.48	9.85	11.47	0.35942	0.05696	0.37460
		hnf(2)	10.38	0.74	9.08	10.36	11.84	0.14884	0.00002	0.15441
		hnf(3)	10.38	0.70	9.13	10.36	11.72	0.15003	0.03853	0.15970

C.2 Linear Regression Model - VI Results

Figure C.2 shows the optimization loss histories for all linear regression datasets. The curves are smoothed using an exponential moving average with smoothing factor $\alpha = 0.1$. Table C.2 summarizes the statistics of the posterior approximation and the corresponding error metrics for each surrogate posterior on every dataset.

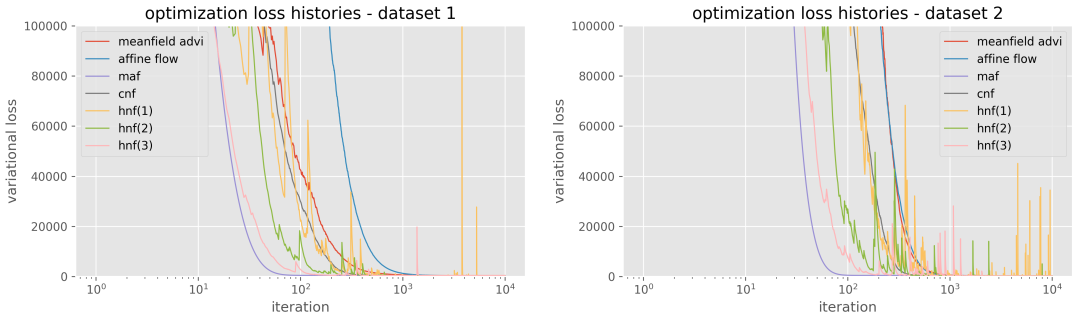

Figure C.2: Optimization loss histories (linear regression datasets).

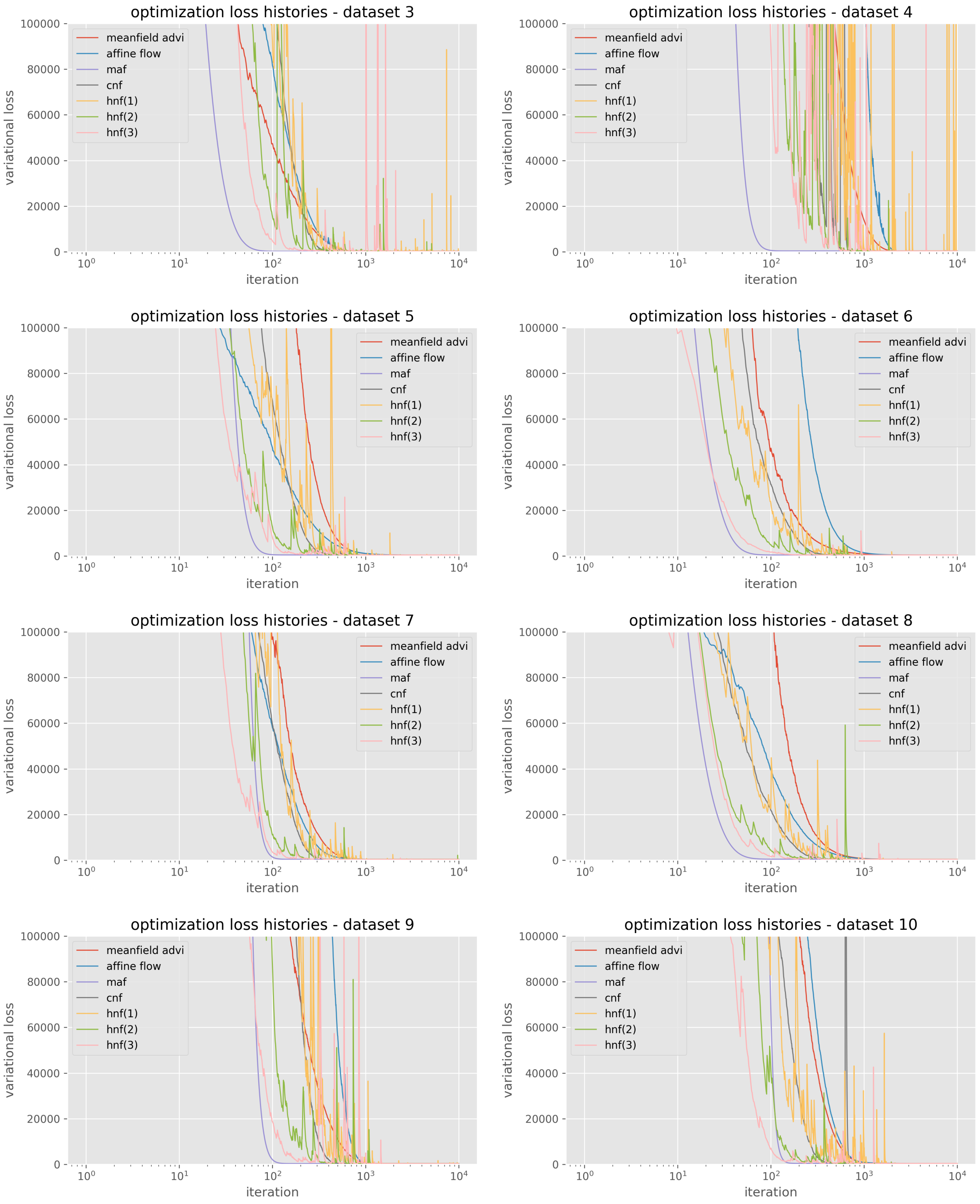

Figure C.2: Optimization loss histories (linear regression datasets).

Table C.2: Statistics and error metrics for each surrogate posterior (linear regression datasets).

					statistics				error metrics	
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 1	beta0	meanfield advi	-8.07	1.53	-10.93	-8.13	-5.16	0.00794	0.00925	0.06954
		affine flow	-8.10	1.47	-10.84	-8.10	-5.37	0.02449	0.07048	0.10737
		maf	-8.15	1.54	-11.07	-8.13	-5.20	0.07078	0.00179	0.11805
		cnf	-8.22	1.55	-11.02	-8.24	-5.27	0.14640	0.00993	0.17920
		hnf(1)	-8.20	1.46	-10.91	-8.21	-5.43	0.12297	0.07989	0.16314
		hnf(2)	-8.38	1.44	-11.14	-8.35	-5.62	0.30453	0.10555	0.33426
		hnf(3)	-7.70	1.45	-10.43	-7.69	-5.01	0.37770	0.08911	0.39520
	beta1	meanfield advi	1.39	0.16	1.08	1.40	1.70	0.00379	0.00154	0.00624
		affine flow	1.39	0.16	1.09	1.39	1.70	0.00120	0.00055	0.00723
		maf	1.37	0.16	1.07	1.37	1.66	0.02273	0.00659	0.02452
		cnf	1.41	0.17	1.08	1.41	1.72	0.01669	0.00590	0.01854
		hnf(1)	1.32	0.18	0.98	1.33	1.65	0.06637	0.02195	0.07171
		hnf(2)	1.39	0.17	1.06	1.39	1.71	0.00136	0.00809	0.01042
		hnf(3)	1.35	0.16	1.05	1.34	1.67	0.04308	0.00190	0.04467
	scale	meanfield advi	15.18	1.05	13.22	15.19	17.17	0.03200	0.05981	0.11950
		affine flow	15.22	1.02	13.36	15.19	17.18	0.00828	0.08827	0.12765
		maf	15.09	1.10	13.00	15.10	17.17	0.11895	0.00347	0.16111
		cnf	15.14	1.03	13.27	15.10	17.18	0.07328	0.07545	0.12484
		hnf(1)	15.40	1.07	13.60	15.39	17.38	0.19106	0.03727	0.30292
		hnf(2)	15.10	1.07	13.15	15.07	17.18	0.11467	0.04092	0.13654
		hnf(3)	14.98	1.02	13.18	14.96	17.00	0.22803	0.08990	0.25141
dataset 2	beta0	meanfield advi	17.30	0.27	16.80	17.30	17.80	3.29371	6.32153	7.19564
		affine flow	17.29	0.29	16.73	17.29	17.84	3.28347	6.29827	7.17476
		maf	17.30	0.27	16.80	17.30	17.82	3.29067	6.31486	7.19194
		cnf	17.34	0.28	16.79	17.34	17.84	3.32664	6.30947	7.20260
		hnf(1)	17.17	0.44	16.44	17.16	17.96	3.15905	6.14617	7.05757
		hnf(2)	17.10	0.30	16.53	17.10	17.65	3.09352	6.28380	7.08257
		hnf(3)	17.19	0.32	16.59	17.19	17.77	3.18306	6.27359	7.11513
	beta1	meanfield advi	-4.85	0.03	-4.90	-4.85	-4.80	1.15766	2.29068	2.57378
		affine flow	-4.85	0.03	-4.90	-4.85	-4.80	1.15884	2.29139	2.57464
		maf	-4.87	0.03	-4.92	-4.87	-4.82	1.18270	2.29185	2.58581
		cnf	-4.85	0.03	-4.90	-4.85	-4.80	1.16078	2.28913	2.57452
		hnf(1)	-4.99	0.11	-5.14	-4.98	-4.87	1.29968	2.20761	2.61968
		hnf(2)	-4.80	0.04	-4.86	-4.80	-4.72	1.10597	2.27575	2.54303
		hnf(3)	-4.86	0.04	-4.93	-4.86	-4.79	1.17444	2.27958	2.57474
	scale	meanfield advi	2.75	0.19	2.40	2.75	3.12	0.48665	0.78847	0.96391
		affine flow	2.73	0.20	2.37	2.73	3.11	0.46706	0.78593	0.95368
		maf	2.74	0.18	2.41	2.73	3.09	0.47100	0.79800	0.96584
		cnf	2.78	0.20	2.44	2.77	3.21	0.51652	0.78047	0.98387
		hnf(1)	3.11	0.35	2.56	3.08	3.82	0.84857	0.63387	1.14116
		hnf(2)	2.78	0.23	2.39	2.76	3.25	0.51242	0.75244	0.96553
		hnf(3)	2.81	0.23	2.42	2.79	3.29	0.54061	0.75183	0.98250

Continued on next page

			statistics					error metrics		
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 3	beta0	meanfield advi	0.19	0.22	-0.21	0.19	0.61	0.01548	0.00398	0.01775
		affine flow	0.22	0.22	-0.20	0.22	0.61	0.03777	0.00278	0.03886
		maf	0.17	0.22	-0.25	0.17	0.57	0.00816	0.00091	0.01071
		cnf	0.17	0.22	-0.23	0.18	0.57	0.00424	0.00295	0.01021
		hnf(1)	0.23	0.26	-0.24	0.23	0.70	0.05502	0.03851	0.07358
		hnf(2)	0.15	0.25	-0.28	0.15	0.59	0.02610	0.03078	0.08741
		hnf(3)	0.14	0.22	-0.27	0.14	0.54	0.03489	0.00417	0.03643
	beta1	meanfield advi	-4.75	0.02	-4.79	-4.75	-4.72	0.00453	0.00029	0.00452
		affine flow	-4.75	0.02	-4.79	-4.75	-4.71	0.00203	0.00052	0.00212
		maf	-4.76	0.02	-4.80	-4.76	-4.73	0.01495	0.00020	0.01498
		cnf	-4.74	0.02	-4.78	-4.74	-4.70	0.00591	0.00004	0.00598
		hnf(1)	-4.78	0.16	-4.89	-4.78	-4.65	0.02782	0.13503	0.14990
		hnf(2)	-4.75	0.08	-4.80	-4.75	-4.69	0.00266	0.05770	0.07092
		hnf(3)	-4.77	0.03	-4.82	-4.76	-4.72	0.01609	0.00727	0.01773
	scale	meanfield advi	2.15	0.15	1.88	2.15	2.44	0.01001	0.01075	0.01842
		affine flow	2.17	0.15	1.89	2.16	2.47	0.00544	0.00474	0.01486
		maf	2.26	0.16	1.96	2.26	2.57	0.09375	0.00153	0.09484
		cnf	2.17	0.16	1.90	2.16	2.49	0.00254	0.00057	0.00470
		hnf(1)	2.37	0.21	2.03	2.36	2.76	0.20179	0.04775	0.20899
		hnf(2)	2.15	0.27	1.82	2.12	2.54	0.01854	0.11013	0.16661
		hnf(3)	2.19	0.18	1.88	2.19	2.53	0.02556	0.01779	0.03282
dataset 4	beta0	meanfield advi	5.08	0.96	3.27	5.09	6.83	0.00580	0.02570	0.04796
		affine flow	5.08	0.94	3.32	5.08	6.84	0.00143	0.00733	0.03596
		maf	5.07	0.95	3.30	5.07	6.87	0.01417	0.01645	0.04206
		cnf	5.03	0.95	3.23	5.07	6.74	0.04653	0.01594	0.07029
		hnf(1)	5.06	1.10	2.94	5.10	7.06	0.01753	0.16544	0.17431
		hnf(2)	5.15	1.04	3.23	5.13	7.18	0.07126	0.11068	0.13781
		hnf(3)	4.86	0.98	3.01	4.85	6.68	0.22068	0.05059	0.23074
	beta1	meanfield advi	14.30	0.08	14.14	14.30	14.46	0.00184	0.00095	0.00235
		affine flow	14.30	0.09	14.15	14.30	14.48	0.00614	0.00346	0.00712
		maf	14.30	0.07	14.17	14.30	14.44	0.00249	0.01512	0.01537
		cnf	14.25	0.09	14.08	14.25	14.40	0.05235	0.00142	0.05265
		hnf(1)	14.39	0.24	13.99	14.38	14.83	0.08853	0.15859	0.18772
		hnf(2)	14.37	0.23	14.01	14.38	14.72	0.07506	0.14940	0.18197
		hnf(3)	14.32	0.14	14.07	14.32	14.55	0.01917	0.05173	0.06669
	scale	meanfield advi	9.48	0.66	8.28	9.48	10.73	0.01672	0.02655	0.06238
		affine flow	9.49	0.66	8.26	9.48	10.72	0.01077	0.02692	0.06380
		maf	9.44	0.71	8.14	9.42	10.83	0.05962	0.02776	0.08227
		cnf	9.44	0.69	8.24	9.40	10.77	0.05838	0.00568	0.06273
		hnf(1)	9.96	0.93	8.48	9.85	11.87	0.45617	0.24802	0.53339
		hnf(2)	9.79	0.80	8.49	9.73	11.40	0.28998	0.11035	0.31340
		hnf(3)	9.68	0.73	8.41	9.64	11.14	0.18338	0.04704	0.19381

Continued on next page

			statistics					error metrics		
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 5	beta0	meanfield advi	17.11	1.60	14.06	17.12	20.20	0.05293	0.04642	0.08672
		affine flow	17.15	1.57	14.30	17.11	20.08	0.01717	0.07567	0.09991
		maf	17.12	1.62	14.10	17.11	20.13	0.04059	0.02109	0.06985
		cnf	17.03	1.63	13.72	17.11	19.91	0.13161	0.01259	0.20674
		hnf(1)	17.36	1.53	14.35	17.38	20.16	0.19483	0.11481	0.23897
		hnf(2)	17.02	1.58	13.98	17.05	19.94	0.14102	0.05726	0.18029
		hnf(3)	17.47	1.64	14.34	17.48	20.49	0.30442	0.00363	0.31562
	beta1	meanfield advi	2.63	0.19	2.27	2.63	2.98	0.01846	0.00028	0.01972
		affine flow	2.64	0.19	2.27	2.64	3.00	0.00790	0.00041	0.01083
		maf	2.71	0.21	2.32	2.71	3.10	0.06569	0.01535	0.06808
		cnf	2.64	0.20	2.27	2.64	3.01	0.00364	0.00394	0.01191
		hnf(1)	2.58	0.23	2.17	2.58	3.04	0.06442	0.03919	0.07611
		hnf(2)	2.59	0.19	2.23	2.59	2.96	0.05756	0.00210	0.05835
		hnf(3)	2.67	0.20	2.29	2.66	3.05	0.02073	0.01229	0.02525
	scale	meanfield advi	16.25	1.11	14.18	16.25	18.30	0.00437	0.06629	0.12483
		affine flow	16.30	1.08	14.30	16.26	18.37	0.04681	0.09549	0.13652
		maf	16.29	1.27	13.94	16.28	18.70	0.04397	0.09322	0.13780
		cnf	16.35	1.12	14.38	16.29	18.57	0.09896	0.05314	0.12054
		hnf(1)	16.14	1.01	14.23	16.14	18.03	0.10742	0.15896	0.21346
		hnf(2)	16.34	1.14	14.33	16.29	18.55	0.08707	0.03607	0.10823
		hnf(3)	16.26	1.14	14.25	16.22	18.56	0.01319	0.03090	0.05335
dataset 6	beta0	meanfield advi	4.54	1.73	1.40	4.50	7.84	0.88013	0.54109	1.14157
		affine flow	4.57	1.69	1.38	4.59	7.71	0.91412	0.58086	1.18392
		maf	4.50	1.64	1.61	4.45	7.71	0.84058	0.63143	1.14916
		cnf	4.46	1.66	1.31	4.47	7.48	0.80086	0.61004	1.09417
		hnf(1)	4.52	1.67	1.38	4.51	7.58	0.86039	0.60031	1.15250
		hnf(2)	4.58	1.71	1.40	4.55	7.80	0.92655	0.56213	1.19259
		hnf(3)	4.45	1.64	1.34	4.42	7.45	0.78930	0.63352	1.11120
	beta1	meanfield advi	-1.28	0.18	-1.62	-1.29	-0.95	0.18746	0.27663	0.35674
		affine flow	-1.31	0.18	-1.67	-1.31	-0.97	0.21527	0.26960	0.36727
		maf	-1.29	0.19	-1.65	-1.29	-0.92	0.19074	0.26313	0.34789
		cnf	-1.29	0.20	-1.68	-1.29	-0.92	0.19591	0.25581	0.34892
		hnf(1)	-1.32	0.18	-1.67	-1.32	-0.97	0.21895	0.26943	0.36965
		hnf(2)	-1.33	0.20	-1.70	-1.33	-0.95	0.23183	0.25399	0.36758
		hnf(3)	-1.30	0.19	-1.66	-1.29	-0.94	0.19841	0.26457	0.35630
	scale	meanfield advi	17.11	1.18	14.94	17.11	19.37	3.29788	5.52787	6.68153
		affine flow	17.06	1.19	14.83	17.04	19.32	3.24603	5.52552	6.64949
		maf	16.83	1.19	14.53	16.84	19.12	3.02309	5.52209	6.54016
		cnf	17.15	1.19	14.95	17.14	19.44	3.34044	5.52653	6.70826
		hnf(1)	17.33	1.01	15.45	17.33	19.21	3.51438	5.70675	6.89733
		hnf(2)	17.50	1.20	15.30	17.47	19.78	3.68500	5.51164	6.87773
		hnf(3)	17.14	1.13	15.12	17.09	19.38	3.32642	5.57926	6.74112

Continued on next page

			statistics					error metrics		
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 7	beta0	meanfield advi	-8.14	0.84	-9.75	-8.13	-6.56	0.01426	0.02378	0.03935
		affine flow	-8.15	0.85	-9.74	-8.16	-6.51	0.02878	0.01416	0.04159
		maf	-8.15	0.86	-9.78	-8.14	-6.56	0.02633	0.01105	0.04320
		cnf	-8.13	0.86	-9.69	-8.15	-6.48	0.01010	0.00532	0.04355
		hnf(1)	-7.95	0.88	-9.59	-7.96	-6.30	0.17396	0.01256	0.17882
		hnf(2)	-8.12	0.89	-9.80	-8.12	-6.44	0.00007	0.02645	0.03915
		hnf(3)	-8.05	0.89	-9.72	-8.05	-6.38	0.07252	0.02567	0.08256
	beta1	meanfield advi	3.62	0.09	3.45	3.62	3.79	0.01273	0.00055	0.01299
		affine flow	3.61	0.09	3.45	3.61	3.78	0.00612	0.00224	0.00686
		maf	3.58	0.08	3.42	3.58	3.74	0.02668	0.00633	0.02769
		cnf	3.57	0.09	3.39	3.58	3.75	0.03405	0.00295	0.03457
		hnf(1)	3.63	0.10	3.44	3.64	3.82	0.02654	0.01414	0.03093
		hnf(2)	3.66	0.09	3.49	3.66	3.83	0.05026	0.00155	0.05070
		hnf(3)	3.62	0.10	3.44	3.62	3.80	0.01226	0.00565	0.01385
	scale	meanfield advi	8.70	0.59	7.59	8.69	9.82	0.03987	0.03808	0.07633
		affine flow	8.69	0.60	7.57	8.70	9.80	0.02517	0.03337	0.07565
		maf	8.54	0.73	7.17	8.53	9.91	0.11794	0.10292	0.16923
		cnf	8.62	0.61	7.49	8.61	9.79	0.04309	0.01995	0.06891
		hnf(1)	8.46	0.62	7.32	8.44	9.72	0.19905	0.00915	0.20443
		hnf(2)	8.51	0.61	7.44	8.48	9.74	0.14711	0.01892	0.15282
		hnf(3)	8.67	0.60	7.61	8.64	9.85	0.00666	0.03025	0.04513
dataset 8	beta0	meanfield advi	13.13	1.13	10.96	13.15	15.24	0.07331	0.03266	0.08878
		affine flow	13.19	1.18	11.04	13.18	15.44	0.01505	0.01604	0.05169
		maf	13.12	1.16	10.93	13.12	15.29	0.07576	0.00549	0.08437
		cnf	13.23	1.18	10.83	13.26	15.36	0.03246	0.02052	0.07421
		hnf(1)	13.32	1.19	11.02	13.35	15.54	0.12076	0.02580	0.13231
		hnf(2)	12.92	1.14	10.79	12.93	15.04	0.28272	0.02494	0.28836
		hnf(3)	13.08	1.12	10.91	13.11	15.14	0.11834	0.04396	0.13133
	beta1	meanfield advi	-0.84	0.12	-1.06	-0.84	-0.63	0.01247	0.00237	0.01335
		affine flow	-0.82	0.12	-1.04	-0.82	-0.59	0.01265	0.00232	0.01370
		maf	-0.82	0.11	-1.04	-0.82	-0.62	0.00391	0.00502	0.00757
		cnf	-0.81	0.12	-1.03	-0.81	-0.59	0.01595	0.00147	0.01662
		hnf(1)	-0.81	0.13	-1.06	-0.81	-0.57	0.01992	0.01309	0.02514
		hnf(2)	-0.81	0.12	-1.03	-0.81	-0.59	0.01720	0.00100	0.01785
		hnf(3)	-0.82	0.12	-1.04	-0.82	-0.59	0.01312	0.00583	0.01579
	scale	meanfield advi	11.73	0.80	10.24	11.72	13.26	0.01170	0.06027	0.09624
		affine flow	11.73	0.80	10.26	11.73	13.23	0.00972	0.05767	0.09497
		maf	11.64	0.70	10.34	11.64	12.97	0.09567	0.15212	0.19495
		cnf	11.78	0.88	10.22	11.75	13.52	0.04545	0.01994	0.07071
		hnf(1)	11.86	0.82	10.41	11.83	13.50	0.12419	0.03571	0.14738
		hnf(2)	11.65	0.84	10.17	11.63	13.32	0.08597	0.01577	0.10205
		hnf(3)	11.67	0.83	10.18	11.64	13.27	0.07268	0.02773	0.09305

Continued on next page

					statistics				error metrics	
dataset	param	surr. posterior	mean	sd	HDI 3%	mode	HDI 97%	mean err.	sd err.	$\mathcal{W}_2$
dataset 9	beta0	meanfield advi	1.32	0.69	-0.00	1.32	2.59	0.01306	0.01445	0.02981
		affine flow	1.34	0.70	0.06	1.33	2.65	0.00597	0.00846	0.02772
		maf	1.36	0.72	0.04	1.36	2.70	0.02697	0.01659	0.03783
		cnf	1.26	0.71	-0.03	1.24	2.61	0.07069	0.00198	0.07690
		hnf(1)	1.41	0.80	-0.11	1.41	2.91	0.07767	0.09019	0.12125
		hnf(2)	1.39	0.82	-0.02	1.37	2.81	0.05480	0.11515	0.28468
		hnf(3)	1.46	0.70	0.22	1.46	2.79	0.13302	0.00951	0.13645
	beta1	meanfield advi	-7.64	0.06	-7.75	-7.64	-7.52	0.00713	0.00338	0.00815
		affine flow	-7.62	0.06	-7.74	-7.62	-7.50	0.00737	0.00219	0.00801
		maf	-7.64	0.06	-7.76	-7.64	-7.52	0.01059	0.00115	0.01107
		cnf	-7.65	0.07	-7.76	-7.65	-7.52	0.01844	0.00125	0.01913
		hnf(1)	-7.53	0.11	-7.74	-7.53	-7.34	0.09522	0.04285	0.10510
		hnf(2)	-7.67	0.07	-7.80	-7.67	-7.54	0.04012	0.00628	0.04126
		hnf(3)	-7.66	0.07	-7.79	-7.66	-7.52	0.02980	0.00859	0.03306
	scale	meanfield advi	7.09	0.50	6.15	7.09	8.02	0.00640	0.01966	0.06928
		affine flow	7.08	0.49	6.17	7.08	8.01	0.00815	0.02662	0.06637
		maf	7.02	0.53	6.03	7.02	8.01	0.06868	0.00703	0.09360
		cnf	7.16	0.53	6.27	7.14	8.18	0.07288	0.00642	0.07687
		hnf(1)	7.23	0.58	6.22	7.21	8.32	0.13409	0.06321	0.18546
		hnf(2)	7.07	0.51	6.17	7.04	8.08	0.02300	0.00889	0.03445
		hnf(3)	7.24	0.55	6.34	7.20	8.28	0.14411	0.03137	0.19309
dataset 10	beta0	meanfield advi	-2.62	1.56	-5.56	-2.63	0.26	0.06240	0.00944	0.08908
		affine flow	-2.63	1.51	-5.45	-2.65	0.23	0.06422	0.03909	0.09251
		maf	-2.64	1.53	-5.50	-2.66	0.28	0.07821	0.01387	0.09586
		cnf	-2.64	1.57	-5.54	-2.65	0.42	0.08265	0.02242	0.10031
		hnf(1)	-2.87	1.62	-5.83	-2.84	0.20	0.30643	0.07046	0.32129
		hnf(2)	-3.13	1.55	-6.05	-3.12	-0.24	0.56610	0.00403	0.57215
		hnf(3)	-2.68	1.50	-5.40	-2.70	0.19	0.11575	0.04887	0.13749
	beta1	meanfield advi	5.07	0.15	4.78	5.07	5.35	0.00438	0.00252	0.00621
		affine flow	5.08	0.15	4.82	5.08	5.37	0.01146	0.00227	0.01307
		maf	5.14	0.13	4.89	5.14	5.39	0.06340	0.02105	0.06712
		cnf	5.04	0.15	4.74	5.04	5.31	0.03827	0.00052	0.03893
		hnf(1)	5.08	0.22	4.76	5.08	5.40	0.00346	0.07095	0.13321
		hnf(2)	5.10	0.16	4.80	5.10	5.39	0.02456	0.00341	0.02538
		hnf(3)	5.06	0.16	4.75	5.06	5.37	0.01712	0.01010	0.02048
	scale	meanfield advi	15.91	1.09	13.87	15.88	17.99	0.02700	0.03953	0.11773
		affine flow	15.91	1.09	13.85	15.91	17.95	0.02624	0.03527	0.12340
		maf	15.78	1.07	13.74	15.81	17.77	0.15318	0.05950	0.20666
		cnf	15.88	1.09	13.82	15.87	17.91	0.05839	0.03881	0.13856
		hnf(1)	16.37	1.08	14.46	16.30	18.47	0.43517	0.04254	0.44116
		hnf(2)	16.05	1.09	14.06	16.03	18.12	0.11893	0.03522	0.15521
		hnf(3)	15.92	1.11	14.00	15.87	18.13	0.01953	0.01632	0.06165

Bibliography

Ali, S. M., & Silvey, S. D. (1966). A General Class of Coefficients of Divergence of One Distribution from Another. *Journal of the Royal Statistical Society: Series B (Methodological)*, *28*(1), 131–142. doi: https://doi.org/10.1111/j.2517-6161.1966.tb00626.x

Amari, S. (2009). Divergence, Optimization and Geometry. In C. S. Leung, M. Lee, & J. H. Chan (Eds.), *Neural information processing* (pp. 185–193). Bangkok, Thailand: Springer Berlin Heidelberg. doi: https://doi.org/10.1007/978-3-642-10677-4_21

Baker, A. (2016). Simplicity. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Winter2016 ed.). Metaphysics Research Lab, Stanford University. Retrieved from `https://plato.stanford.edu/archives/win2016/entries/simplicity/`

Betancourt, M. (2017a). A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv*. Retrieved from `http://arxiv.org/abs/1701.02434`

Betancourt, M. (2017b). The Convergence of Markov chain Monte Carlo Methods: From the Metropolis method to Hamiltonian Monte Carlo. *arXiv*. Retrieved from `http://arxiv.org/abs/1706.01520`

Betancourt, M. (2018a). *Conditional Probability Theory (For Scientists and Engineers).* Retrieved 2021-01-20, from `https://github.com/betanalpha/knitr_case_studies/tree/master/conditional_probability_theory` (commit b474ec1a5a79347f7c9634376c866fe3294d657a)

Betancourt, M. (2018b). *Probability Theory (For Scientists and Engineers).* Retrieved 2021-01-20, from `https://github.com/betanalpha/knitr_case_studies/tree/master/probability_theory` (commit b474ec1a5a79347f7c9634376c866fe3294d657a)

Betancourt, M. (2019). *Probabilistic Modeling and Statistical Inference.* Retrieved 2021-01-20, from `https://github.com/betanalpha/knitr_case_studies/tree/master/modeling_and_inference` (commit b474ec1a5a79347f7c9634376c866fe3294d657a)

Betancourt, M., Byrne, S., Livingstone, S., & Girolami, M. (2014). The Geometric Foundations of Hamiltonian Monte Carlo. *arXiv*. Retrieved from `http://arxiv.org/abs/1410.5110`

Bishop, C. M. (2006). *Pattern Recogniton and Machine Learning.* Springer New York.

Blei, D. M., Kucukelbir, A., & Mcauliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877. doi: https://doi.org/10.1080/01621459.2017.1285773

Bogachev, V. I., Kolesnikov, A., & Medvedev, K. (2005). Triangular transformations of measures. *Sbornik: Mathematics*, *196*(3), 309–335. doi: https://doi.org/10.1070/sm2005v196n03abeh000882

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural Ordinary Differential Equations. *arXiv*. Retrieved from `http://arxiv.org/abs/1806.07366`

Çinlar, E. (2011). *Probability and Stochastics.* Springer New York.

Devroye, L. (1986). *Non-Uniform Random Variate Generation.* Springer New York. doi: https://doi.org/10.1007/978-1-4613-8643-8

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., ... Saurous, R. A. (2017). TensorFlow Distributions. *arXiv*. Retrieved from `http://arxiv.org/abs/1711.10604`

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using Real NVP. *arXiv*. Retrieved from `http://arxiv.org/abs/1605.08803`

Donnelly, D., & Rogers, E. (2005). Symplectic integrators: An introduction. *American Journal of Physics*, *73*(10), 938–945. doi: https://doi.org/10.1119/1.2034523

Dupont, E., Doucet, A., & Teh, Y. W. (2019). Augmented neural ODEs. *arXiv*. Retrieved from `https://arxiv.org/abs/1904.01681`

Flamary, R., & Courty, N. (2017). *POT Python Optimal Transport library.* Retrieved from `https://pythonot.github.io/`

Folland, G. (2009). *A Guide to Advanced Real Analysis.* The Mathematical Association of America. doi: https://doi.org/10.5948/upo9780883859155

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). Chapman & Hall/CRC.

Girolami, M., & Calderhead, B. (2011). Riemannian Manifold Hamiltonian Monte Carlo. *Statistical Methodology*, *73*(2), 123–214. doi: https://doi.org/10.1111/j.1467-9868.2010.00765.x

Goldstein, H., Poole, C., Safko, J., & Addison, S. R. (2001). *Classical Mechanics* (3rd ed.). Pearson.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., & Duvenaud, D. (2018). *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models.* arXiv. Retrieved from `https://arxiv.org/abs/1810.01367`

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv*. Retrieved from `https://arxiv.org/abs/1512.03385`

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, *14*, 1303–1347. doi: https://dl.acm.org/doi/10.5555/2567709.2502622

Hoffman, M. D., & Gelman, A. (2011). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *arXiv*. Retrieved from `http://arxiv.org/abs/1111.4246`

Huang, C.-W., Dinh, L., & Courville, A. (2020). Augmented Normalizing Flows: Bridging the Gap Between Generative Flows and Latent Variable Models. *arXiv*. Retrieved from `http://arxiv.org/abs/2002.07101`

Hutchinson, M. F. (1990). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, *19*(2), 433–450. doi: https://doi.org/10.1080/03610919008812866

Kallenberg, O. (1997). *Foundations of Modern Probability*. Springer New York.

Keener, R. W. (2010). *Theoretical Statistics*. Springer New York.

Kobyzev, I., Prince, S., & Brubaker, M. (2019). Normalizing Flows: An Introduction and Review of Current Methods. *arXiv*. Retrieved from `https://arxiv.org/abs/1908.09257`

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., & Blei, D. M. (2016). Automatic Differentiation Variational Inference. *arXiv*. Retrieved from `http://arxiv.org/abs/1603.00788`

Lehmann, E. L., & Casella, G. (1998). *Theory of Point Estimation* (2nd ed.). Springer New York.

McCullagh, P. (2002). What is a statistical model? *Annals of Statistics*, *30*(5), 1225–1310. doi: https://doi.org/10.1214/aos/1035844977

Mohamed, S., Rosca, M., Figurnov, M., & Mnih, A. (2019). Monte Carlo Gradient Estimation in Machine Learning. *arXiv*. Retrieved from `http://arxiv.org/abs/1906.10652`

Neal, R. M. (2011). MCMC using Hamiltonian Dynamics. In *Handbook of markov chain monte carlo* (pp. 113–162). Chapman and Hall/CRC.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2019). Normalizing Flows for Probabilistic Modeling and Inference. *arXiv*. Retrieved from `http://arxiv.org/abs/1912.02762`

Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked Autoregressive Flow for Density Estimation. *arXiv*. Retrieved from `http://arxiv.org/abs/1705.07057`

Ranganath, R., Gerrish, S., & Blei, D. M. (2013). Black Box Variational Inference. *arXiv*. Retrieved from `https://arxiv.org/abs/1401.0118`

Rezende, D. J., & Mohamed, S. (2015). Variational Inference with Normalizing Flows. *arXiv*. Retrieved from `https://arxiv.org/abs/1505.05770`

Salimans, T., Kingma, D. P., & Welling, M. (2014). Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. *arXiv*. Retrieved from `https://arxiv.org/abs/1410.6460`

Shampine, L. F. (1986). Some Practical Runge-Kutta Formulas. *Mathematics of Computation*, *46*(173), 135–150. doi: https://doi.org/10.2307/2008219

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., & Higgins, I. (2019). Hamiltonian Generative Networks. *arXiv*. Retrieved from `http://arxiv.org/abs/1909.13789` van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge University Press. doi: https://doi.org/10.1017/CBO9780511802256

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2019). Rank-normalization, folding, and localization: An improved $\widehat{R}$ assessing convergence of MCMC. *arXiv*. Retrieved from `http://arxiv.org/abs/1903.08008` doi: 10.1214/20-BA1221

Villani, C. (2009). *Optimal Transport*. Springer Berlin Heidelberg.

Zhang, C., Butepage, J., Kjellstrom, H., & Mandt, S. (2017). Advances in Variational Inference. *arXiv*. Retrieved from `https://arxiv.org/abs/1711.05597`

Zhang, H., Gao, X., Unterman, J., & Arodz, T. (2019). Approximation Capabilities of Neural ODEs and Invertible Residual Networks. *arXiv*. Retrieved from `http://arxiv.org/abs/1907.12998`