



Utrecht University

Institute for Gravitational and
Subatomic Physics

Applied Data Science Master's Thesis

A variational inference scheme for gravitational-wave parameter
estimation

Candidate: Oleksii Bevza
First Examiner: Chris Van Den Broeck
Second Examiner: Thibaud Wouters

Date: August 2025

Contents

1	Abstract	3
2	Introduction	3
3	Computing challenges and currently available methods for the Einstein Telescope project	4
3.1	Computing challenges of the gravitational wave signals	4
3.2	Normalizing flows as a method of improving computational efficiency . .	5
3.3	Currently available methods of parameter estimation of Compact Binary Coalescence signals	8
4	Variational Inference as alternative to Markov Chain Monte Carlo	9
4.1	Theoretical aspects of Variational Inference	9
4.2	Advances in Variational Inference	14
4.3	Comparison of Markov Chain Monte Carlo and Variational Inference . .	18
4.4	Conclusion of literature review	19
5	Toy problem	20
5.1	Experiment setup	20
5.2	Results	22
5.2.1	Visual inspection	22
5.2.2	Comparison of the means, variances, and standard deviations of the true and VI distributions	23
5.2.3	Examination of the parametric and nonparametric KL divergences.	24
6	Conclusion	25
7	Future work	25
8	Appendixes	26
8.1	Appendix A	26
8.2	Appendix B	28
8.3	Appendix C	31
8.4	Appendix D	32

1 Abstract

Approximating complex probability densities via optimization using flow-based variational inference (VI) could potentially be an avenue to address current computational challenges in gravitational-wave astronomy. This thesis presents a thorough literature review on the conceptual fundamentals of VI, including its enhancements and limitations in high-dimensional spaces. A VI sampler using a Block Neural Autoregressive Flow was applied to a 15-dimensional toy problem to test the applicability of this approximation technique to gravitational-wave data analysis. Although the results of the experiment are promising, additional validation on more complex toy problems should be performed before VI can be applied directly to gravitational-wave analysis due to identified conceptual constraints related to variance underestimation. Nevertheless, the theoretical framework and recent research indicate potential methods to alleviate these constraints.

2 Introduction

Gravitational-wave astronomy is on the brink of a data-driven revolution. The third-generation signal detectors are expected to be at least ten times more sensitive than current, vastly improving the ability to explore the Universe. This leap in sensitivity, however, comes with formidable computational challenges that must be addressed.

Recent work in machine learning suggests a promising way forward. Normalizing-flow architectures can turn simple base distributions into complex, high-dimensional densities through a sequence of invertible transformations, delivering fast likelihood evaluations that are well suited to big data analysis. When these flows are trained within the variational-inference (VI) framework, they offer a significant speed-up over Markov Chain Monte Carlo (MCMC) while still providing reasonably accurate posterior estimates, a property that has already been demonstrated on toy problems and different datasets in multiple fields.

This thesis explores whether flow-based VI can be used to address gravitational-wave data analysis challenges. In Section 3, the problem is described and the basic idea of normalizing-flow architectures is explained. In Section 4, the fundamental concept of VI is presented, and recent improvements to VI are reviewed alongside a comparison with MCMC methods. In Section 5, a simple 15-dimensional toy problem is introduced: a single-component multivariate normal distribution with varying mean vector μ and covariance matrix Σ where VI samples are compared to the true theoretical distribution. The results demonstrate both the promise and the current limits of VI for gravitational-wave data analysis. Furthermore, this toy problem is viewed as a stepping stone that motivates specific future developments.

3 Computing challenges and currently available methods for the Einstein Telescope project

3.1 Computing challenges of the gravitational wave signals

Gravitational wave signals are currently the only available source that encodes information about energetic processes in the Universe. With the advent of the Einstein Telescope, a next-generation observatory, researchers are on the cusp of a revolutionary advance in the detection and analysis of both transient and continuous gravitational waves. The detectors of this state-of-the-art facility are expected to be at least ten times more sensitive than current second-generation detectors, vastly improving the ability to explore the Universe and significantly increasing the number of detectable events.

Gravitational wave research is typically divided into three key stages:

- Online (real-time) data acquisition: this stage involves the collection, filtering, and initial reduction of raw data from gravitational wave detectors.
- Low-latency processing: at this stage, transient detections and multi-messenger astronomical analyses are performed by rapidly analyzing the incoming data stream. The primary goal here is real-time event detection, which requires extremely fast parameter estimation to enable timely follow-up observations.
- Offline (post-processing) analysis: this phase is dedicated to detailed analysis after an event has occurred. It includes high-fidelity parameter estimation and comprehensive analysis of signals received.

However, the deployment of third-generation detectors such as the Einstein Telescope introduces new computational challenges:

- Increased event volume: the enhanced sensitivity will lead to a dramatic rise in the number of detectable events. Current methods are insufficient for processing this volume with the required low latency - especially for real - time parameter estimation. Simply scaling up existing approaches will not be adequate to meet these demands.
- Higher computational requirements: without significant advancements in algorithm design and optimization, third-generation detectors will face limitations imposed by available computing resources. Hence, improving computational efficiency is not optional - it is essential.[1]

3.2 Normalizing flows as a method of improving computational efficiency

Compact Binary Coalescence signals are defined by the characteristics of their sources. Estimating these sources' properties involves inferring parameters - typically starting from fifteen (the masses of the components, spin-related parameters, extrinsic parameters, inclination angle, polarization angle, the phase, and time of coalescence). In some cases, additional parameters related to tidal deformability, orbital eccentricity, anomalies, and modifications in neutron stars may be necessary to model the characteristics of their sources.[2]

Under the above circumstances, parameter estimation is performed within a Bayesian analysis framework where the probability distribution is not easily analyzable due to the dimensionality. Markov Chain Monte Carlo and Nested Sampling, which are not computationally efficient, are believed to be the most popular methods of parameter estimation in the field.[2]

Since a significant portion of the literature describes normalizing flows as a flexible tool applicable not only to probabilistic modeling but also to improving the efficiency of inference, it is important to take a closer look at this architecture. From the discussion above, it is clear that low-latency processing is the primary challenge, as data must be processed rapidly. Another important point is that high-precision accuracy is not required at this stage, since the goal is to identify events rather than to analyze them in detail, as is done in offline (post-processing) analysis.

Normalizing flows is a flexible method which allows a simple distribution to be transformed multiple times in order to achieve a complex distribution. Each transformation in the sequence must be both invertible and differentiable. To calculate the probability density of a point under the transformed distribution, it is important to map the point back to the original distribution and then adjust for how the transformations have changed the volume of space. This adjustment involves multiplying the density of the mapped point in the original space by the absolute value of the determinant of the Jacobian matrix for each transformation.[3]

The above can be represented mathematically as (1), where \mathbf{x} denotes a real-valued vector, $p_x(\mathbf{x})$ is the target distribution, and $p_u(\mathbf{u})$ is the base distribution that is transformed to match the shape of $p_x(\mathbf{x})$. Here, $J_T(\mathbf{u})$ denotes the Jacobian matrix of the transformation T , containing all partial derivatives of T with respect to \mathbf{u} . [3]

$$p_x(x) = p_u(u) |\det J_T(u)|^{-1} \quad (1)$$

where

$$J_{T(u)} = \begin{bmatrix} \frac{\partial T_1}{\partial u_1} & \dots & \frac{\partial T_1}{\partial u_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_D}{\partial u_1} & \dots & \frac{\partial T_D}{\partial u_D} \end{bmatrix}$$

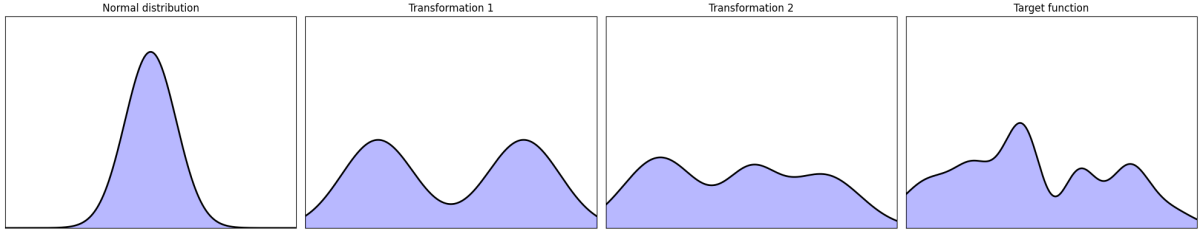
and

$$u = T^{-1}(x)$$

In this way, normalizing flows provide a flexible way to build complex distributions by starting with a simple distribution and applying a sequence of carefully chosen trans-

formations. New samples can be generated by sampling from the base distribution and applying the transformations. Similarly, the probability of any sample can be computed by reversing the process and accounting for the change in volume. Figure 1 illustrates a primitive example where a normal distribution has been transformed to obtain a target distribution.[4][3]

Figure 1: Sequence of transformations to obtain a target distribution from a standard normal distribution.



The theory of the subject emphasizes 2 types of normalizing flow architectures: flows based on neural networks are summarized in Table 1, while flows based on parameterized transformations are summarized in Table 2.

Table 1: Families of flows based on neural network

Flow Type	Description
Neural Autoregressive Flow	Represents the coupling function using a deep neural network. While neural networks are not generally invertible, it has been proven that under specific conditions, they can be bijective.[5]
Neural Splines	Model expressive distributions by transforming simple noise into complex ones using continuous, differentiable monotonic rational-quadratic splines.[6]
Circular Neural Spline Flow	Constructs normalizing flows on non-Euclidean manifolds such as circles, tori, and spheres, enabling the modeling of more expressive distributions on geometric domains.[7]
Residual Flows	Improve flow-based generative models by allowing unbiased, memory-efficient training and by introducing better activation functions for smoother gradient flow.[8]
Continuous Normalizing Flow	Uses neural ordinary differential equations to transform a simple base distribution into a complex target distribution over continuous time.[9]

Table 2: Families of flows based on parameterized transformations

Flow Type	Description
Elementwise Flows	Apply bijective non-linear transformations to each input element independently. While flexible and invertible, they lack interaction between variables, which limits their ability to model correlations.[4]
Linear Flows	Use matrix-based transformations to capture correlations. However, they remain within the same distribution family (e.g., Gaussian remains Gaussian) and are often used as components in more expressive flows.[4]
Planar Flows	Perform localized expansions or contractions via a simple transformation. Invertibility is ensured by parameter constraints. Expressiveness is limited due to the low-rank Jacobian.[4][10]
Radial Flows	Deform distributions by expanding or contracting around a specific point. The Jacobian determinant is efficient to compute, but the inverse has no closed-form (though it exists under certain conditions).[4][10]
Coupling and Autoregressive Flows	Coupling flows split the input vector $x \in R^D$ into parts $A(x^A, x^B) \in R^d \times R^{D-d}$, transforming one part while the other (via a conditioner) controls the transformation. The conditioner can be highly complex.[4] Autoregressive flows compute output variables sequentially, each depending only on earlier input values. They generalize triangular matrix multiplication and can approximate any density with sufficient data. Used in NICE[11], Real NVP [12]
Splines	Piecewise polynomial or rational coupling functions. Piecewise linear and quadratic splines restrict input to $[0, 1]$ and use uniform domain segments. Cubic splines allow flexible mapping but can suffer from inversion issues. Monotonic rational-quadratic splines support exact inversion and stable Jacobian computation.[4]
Glow	Defines invertible transformations between simple latent distributions and observed data distributions.[13]

The newest study evaluates how generative models like INNs and hybrid VAE+INN frameworks can efficiently replicate particle interactions in high-dimensional detector data, achieving strong accuracy and speed, particularly for straightforward simulation tasks. However, increased complexity in particle types reveals limitations in current architectures, underlining the importance of further refinements to boost model capacity and manage sparse data representations.[14] Another study on applying normalizing flows in high-dimensional spaces, i-flow, proposes enhancements such as using convolutional layers, enabling transfer across related tasks, and optimizing memory usage to improve efficiency and flexibility.[15]

3.3 Currently available methods of parameter estimation of Compact Binary Coalescence signals

Building on the aforementioned acceleration techniques, ongoing work on low-latency processing includes the following findings.

On the one hand, researchers are focusing on improving the computational efficiency of existing sampling techniques. Some researchers analyze the computational cost of Bayesian parameter estimation by comparing the standard method with three acceleration techniques: relative binning, multibanding, and reduced order quadrature. The results show that reduced order quadrature offers the best speed-accuracy balance, although improvements are still needed.[2] Another study validates a parameter estimation pipeline using normalizing flow by injecting simulated gravitational-wave signals into synthetic Gaussian noise and verifying that the recovered parameter quantiles follow a uniform distribution, confirming statistical consistency.[16] Other researchers have investigated improving sampling using normalizing flows, addressing tail underestimation issues of reverse-KL training by incorporating adaptive MCMC methods that blend global generative proposals with local sampling steps. Building on previous efforts like Markov Score Climbing and Boltzmann generators, their approach replaces simple variational families with more flexible flows.[17]

On the other hand, researchers are focusing on sampling techniques and have introduced the nested sampling package Bilby as a Bayesian computational method for gravitational wave astronomy that prioritizes calculating the evidence directly, evolving a set of samples constrained by likelihood levels to explore the prior mass efficiently. Unlike traditional MCMC approaches, these methods focus on the structure of likelihood contours.[18][19] Also, the nested sampling algorithm Nessai that uses normalizing flows to efficiently sample within contours was introduced, eliminating the need for random walks or bounding distributions. Future work will focus on extending Nessai to more complex waveform models and improving the efficiency of the algorithm.[20]

It is important to highlight a recent development in the field: a new package called flowMC. This framework enhances traditional MCMC sampling by integrating deep generative models, particularly normalizing flows, to perform global changes across parameter space. Built on JAX and Flax, the library combines local MCMC chains with a trainable global sampler, enabling efficient exploration of complex, high-dimensional, and multimodal posterior distributions.[21]

One approach combines relative binning, gradient-based MCMC, JAX, and normalizing flows to generate accurate posterior samples in under thirty minutes without requiring precomputed inputs. In other words, flowMC achieves strong performance without costly offline training, making it well-suited for low-latency follow-up of events such as binary neutron star mergers.[22]

4 Variational Inference as alternative to Markov Chain Monte Carlo

4.1 Theoretical aspects of Variational Inference

While reviewing the summary of available methods for parameter estimation of Compact Binary Coalescence signals, a careful reader may notice a common challenge faced by researchers when using Markov Chain Monte Carlo (MCMC) methods:

- MCMC methods have very high computational demands;
- Although MCMC can produce asymptotically exact results by sampling the chain, making them well-suited for offline or post-processing analysis, they are less appropriate for low-latency processing applications[23][24]

Given the challenges associated with MCMC and the urgent need for low-latency processing, it is important to consider alternative approaches. One such approach is Variational Inference (VI), which aims to approximate the target posterior distribution using optimization techniques. The concept of VI was first introduced in 1999.[25] VI may prove to be a viable alternative for several key reasons:

- Compared to MCMC, VI is typically much faster, as it relies on optimization rather than sampling from a chain. It is important to note that it can only provide similar densities to the target distribution and not asymptotically exact results.[23][24]
- Although still a relatively underexplored area in inference, there is emerging evidence suggesting that VI is well-suited for applications involving large datasets.[23][24]

Given these considerations and the pressing need for time-efficient, low-latency processing, a thorough investigation into the applicability of VI for Compact Binary Coalescence signal analysis is needed. A comprehensive literature review reveals three main streams of current research on variational inference: (i) the introduction of variational approximation as a technique for inference; (ii) methodological enhancements aimed at improving VI performance; and (iii) comparative analyses of MCMC and VI performance. Each of these research streams will be discussed in detail below. However, before introducing the theory of variational inference, it is important to understand the background behind it.

To start with, a model represents an approximation of a real-world scenario at a certain level, incorporating some assumptions. In statistics, it is a mathematical framework that describes a process using both data and assumptions. These models include parameters that are unknown and must be estimated from the available data to reveal hidden factors or predict outcomes. If a model does not align with the data, it can be rejected in favor of a better one. Statistical inference is the process of using observed data to draw conclusions about the underlying probability distribution. In conclusion, a probabilistic model $p(x)$ represents a probability distribution over a representative set of data $\{x_1, x_2, \dots, x_n\}$, and modeling $p(x)$ means fitting a distribution to this data. It is also important to note that probabilistic models can be conditional, such as $p(y | x)$ [26][27]

Another important topic that needs to be discussed is latent variable models, a framework used to represent complex probability distributions by introducing unobserved (latent) variables. These variables are not directly observed in the data but play a crucial role in modeling the underlying structure of the distribution. In the marginal distribution $p(x)$,

the variable x is the observed data of interest. In the conditional distribution $p(y | x)$, x is treated as observed evidence, and y is the prediction target. [27]

If there is a situation where modeling $p(x)$ directly is difficult due to its complexity, perhaps it's multi-modal or just does not fit any simple distributional family. In such cases, latent variable z can be introduced to simplify the modeling process. The latent variable model then expresses the observed distribution as:

$$p(x) = \sum_z p(x | z) * p(z) \quad (2)$$

if a discrete distribution is given, or

$$p(x) = \int p(x | z) * p(z) dz \quad (3)$$

if continuous distribution is given, where

$p(z)$ is the prior distribution over the latent variable z , chosen to be simple (a standard Gaussian or mixture distribution). $p(x | z)$ is the conditional likelihood, also chosen to be a simple distribution, but with complex parameters that may vary with z . Despite both $p(x | z)$ and $p(z)$ being simple, the marginal distribution $p(x)$ becomes complex after integrating out z . This technique enables to model complex distributions using tractable blocks. So, latent variable models a complex observed distribution $p(x)$ as the result of integrating over a product of simple distributions, namely, a prior $p(z)$ and a conditional likelihood $p(x | z)$. [26][27][28] Figure 2 shows how a Gaussian mixture model was used to model the density of a more complicated distribution.

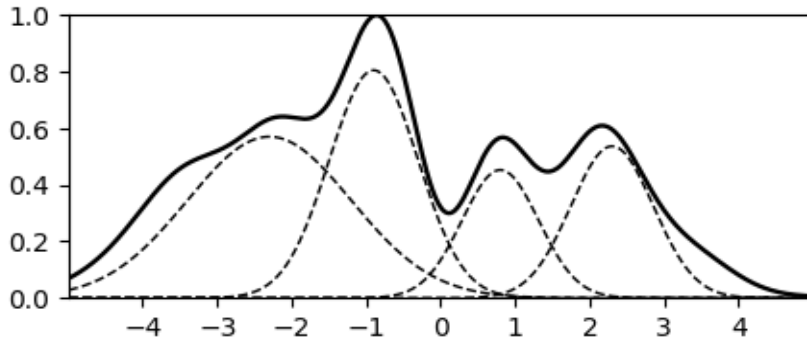


Figure 2: Latent variable model approximation of a complex distribution

The main issue arises in continuous cases because posterior distributions often become highly complex and high-dimensional, making analytical computations challenging and intractable. For discrete cases, summing over all latent variables is computationally intensive, as the number of hidden variables is believed to grow exponentially. It can also be represented as a graphical model (Figure 3) where arrows indicate dependency. [29][30]

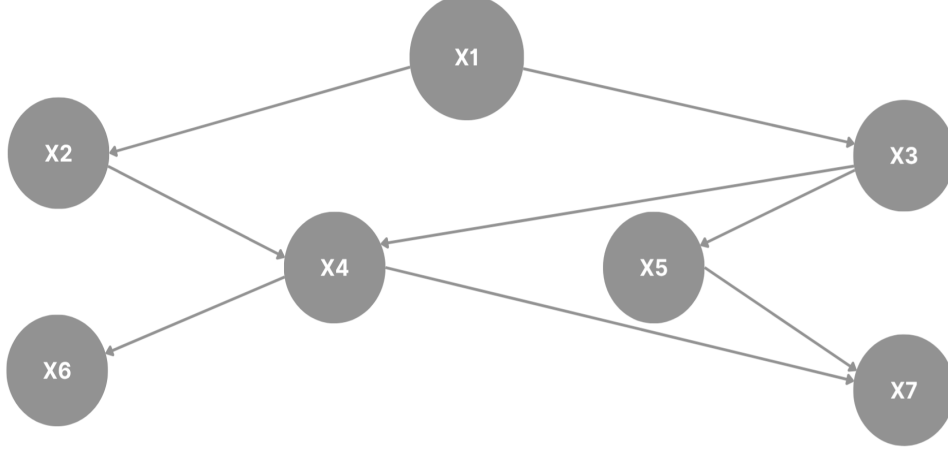


Figure 3: Graphical model representation

If joint probability density is relatively easy to compute:

$$P(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = P(x_7 | x_4, x_5) \cdot P(x_6 | x_4) \cdot P(x_5 | x_3) \cdot P(x_4 | x_2, x_3) \cdot P(x_3 | x_1) \cdot P(x_2 | x_1) \cdot P(x_1) \quad (4)$$

Computing conditional probability considering the example shown in Figure 3 might be intractable:

$$p(z | x) = \frac{p(x, z)}{p(x)} = p(x_4, x_7 | x_1, x_2, x_3, x_5, x_6) = \frac{p(x_1, x_2, x_3, x_4, x_5, x_6, x_7)}{\iint p(x_1, x_2, x_3, x_4, x_5, x_6, x_7) dx_4 dx_7} \quad (5)$$

One practical solution to this problem is approximation. A function can be understood as a mapping that takes an input variable and returns an output value. Its derivative indicates how the output value changes with small adjustments to the input. Extending this idea, we consider an operator (or functional) that accepts an entire function as input and returns a numerical value as output. Many problems can thus be expressed as optimization problems involving these operators, where the objective is to find the input function that minimizes or maximizes the operator's output. Approximate solutions typically limit the class of functions explored during optimization, simplifying the computational process. One of these solutions is variational inference. [29][31]

The objective of variational inference is to approximate the conditional distribution of latent variables given observed data. [23] This is achieved by formulating inference as an optimization problem. Specifically, one posits a family of tractable distributions (e.g., Gaussians) over the latent variables, which have a set of parameter values within that family. The optimal member of this family is identified by minimizing the Kullback-Leibler (KL) divergence between the candidate distribution and the true posterior distribution of interest. The resulting variational distribution then serves as a surrogate for the exact posterior, enabling approximate inference that is computationally efficient and scalable. [23][29][30][31]

By taking a closer look at the following equation:

$$p(z | x) = \frac{p(x, z)}{p(x)} \quad (6)$$

Where

- $p(z|x)$ is unknown
- $p(x, z)$ known
- $p(x)$ intractable to compute

Despite the intractability of $p(x)$, it is still possible to approximate the unknown $p(z|x)$ by introducing a function $q(z)$ which aims to approximate $p(z|x)$. Let KL-divergence, which is the measure of the difference between two probability distributions and discussed in Appendix A, be presented as:

$$KL = - \sum q(z) * \log \frac{p(z | x)}{q(z)} \quad (7)$$

then,

$$\begin{aligned} KL(q(z) | p(z | x)) &= - \sum q(z) \log \left\{ \frac{p(z | x)}{q(z)} \right\} = - \sum q(z) \log \left\{ \frac{\frac{p(x, z)}{p(x)}}{q(z)} \right\} \\ &= - \sum q(z) \log \left\{ \frac{p(x, z)}{p(x)} \cdot \frac{1}{q(z)} \right\} = - \sum q(z) \log \left\{ \frac{p(x, z)}{q(z)} \cdot \frac{1}{p(x)} \right\} \\ &= - \sum q(z) \left[\log \left\{ \frac{p(x, z)}{q(z)} \right\} + \log \left\{ \frac{1}{p(x)} \right\} \right] \\ &= - \sum q(z) \left[\log \left\{ \frac{p(x, z)}{q(z)} \right\} - \log \{p(x)\} \right] \\ &= - \sum q(z) \log \left\{ \frac{p(x, z)}{q(z)} \right\} + \sum q(z) \log \{p(x)\} \end{aligned} \quad (8)$$

so that

$$- \sum q(z) * \log \left\{ \frac{p(z | x)}{q(z)} \right\} + \sum q(z) * \log \left\{ \frac{p(x, z)}{q(z)} \right\} = \log \{p(x)\} \quad (9)$$

Let L be the evidence lower bound on the log-likelihood of the observed data discussed in Appendix B, and be presented as:

$$L = \sum q(z) * \log \frac{p(x, z)}{q(z)} \quad (10)$$

so that

$$KL + L = \ln p(x) \quad (11)$$

By increasing the lower bound, we reduce the KL divergence. Maximizing the lower bound is often more practical, since the KL divergence involves the joint distribution, whereas the lower bound only requires the joint probability in its numerator. Because we want the KL divergence to be as small as possible, the goal is to make the lower bound as large as possible. Therefore, the key idea is to find a distribution $q(z)$ that maximizes the lower bound. This approach forms the foundation of variational inference. By selecting a tractable form for $q(z)$, the inference problem becomes computationally feasible.

Graphically, the idea of Variational Inference is represented in Figure 4. $Q(z)$ is the set of all possible latent distributions within a chosen family (e.g., Gaussians), meaning the search is over all possible parameter values within that family. The true distribution denoted $p(z | x)$ typically lies outside Q . The idea is to find a distribution (target point) within $Q(z)$ that is as close as possible to $p(z | x)$ using optimization techniques to minimize the distance between them. [31]

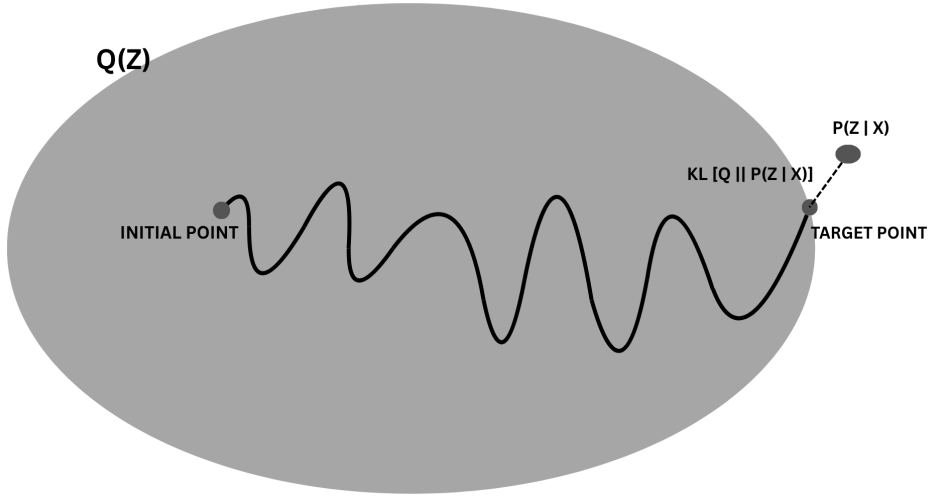


Figure 4: Basics of Variational Inference[31]

4.2 Advances in Variational Inference

A second stream of research focuses on recent developments and enhancements aimed at improving the accuracy and computational efficiency of variational inference. One of these is Stochastic Variational Inference (SVI), which applies stochastic optimization (e.g., stochastic gradient descent), allowing VI to process massive datasets. SVI works by processing random mini-batches of the data at each iteration, rather than the full dataset, to create a noisy but unbiased estimate of the Evidence Lower Bound. This means that, rather than computing the full sum over all data points (which is computationally expensive), only a subset is used per iteration, leading to significant computational savings. Gradients are also approximated with respect to a randomly sampled mini-batch. [32][33][34]

Some researchers focus on time efficiency and have applied the SVI algorithm to probabilistic topic models, where latent distributions are used to identify the most frequently used words. The algorithm was tested on a large collection of words, and it was shown that SVI improves time efficiency. Moreover, larger mini-batch sizes perform better, and higher forgetting rates (close to 1) lead to better convergence.[33] Other researchers have explained why the quality of the variational approximation degrades as dimensionality increases. The optimization process becomes noisier, as each parameter contributes to the total variance. The variational approximation becomes less accurate because the noisy parameters result in a poor fit to the true posterior. This suggests that in high-dimensional settings, the variance of the optimization noise accumulates linearly with the number of parameters, making the average solution noisier and further from the optimum. To address this problem, they propose an algorithm that ensures iterations with finite variance. In addition, R statistic was introduced to assess convergence, and the Monte Carlo standard error (MCSE) was used for iterative averaging. Their goal was to ensure that the final parameter estimates are statistically reliable despite the presence of noise.[34]

Some in the field see the process of optimizing the variational approximation to the true posterior as analogous to optimizing a policy in reinforcement learning, where:

- The variational distribution acts like a policy;
- Sampling latent variables is analogous to taking actions;
- The reward is determined by how well the sampled variables explain the data;

Those techniques can be borrowed to reduce the variance of gradient estimates. This makes the optimization more stable and efficient, enabling VI to be applied to a wider class of models. [32][35]

Other advances include Collapsed VI which improves upon traditional variational methods by first marginalizing out model parameters, which reduces dependencies among latent variables and makes the mean-field approximation much more accurate [32][36];

Sparse VI compresses datasets before performing inference by assuming that large datasets contain a substantial amount of redundancy. This is achieved through Bayesian coresets construction, a new approach that automatically creates small, representative summaries of large datasets. [32][37]

Distributed VI tackles the challenge of scalable, real-time probabilistic inference in sensor

networks. The algorithm is designed to address intractable posteriors and large-scale real-time data. Instead of the traditional evidence lower bound (ELBO), the paper introduces a distributed evidence lower bound (DELBO), where node-specific objective functions are optimized separately. [38]

The idea behind Boosting VI is to improve the approximation by mixing it with components from a base distribution, resulting in even higher accuracy. Toy examples using the Cauchy distribution, a mixture of four univariate Gaussians, and a mixture of five bivariate Gaussians with random means and covariances demonstrate that this approach can capture multimodality and various distributional shapes. Experiments with the banana distribution also showed that it is possible to achieve a trade-off between running time and accuracy. [39]

Importance sampling in VI allows to use samples from the variational approximation $q(z)$ and correct for the mismatch between $q(z)$ and the true posterior $p(z|x)$ by reweighting samples, resulting in better approximations of both marginal likelihoods and posterior expectations. Experiments were performed on Dirichlet distributions, mixtures of Gaussians (clutter model), and Bayesian logistic regression models with a Cauchy prior. It was proved that importance sampling gives a tighter lower bound in comparison to simple lower bound. [40]

Amortized VI uses a shared inference function to map each observation to its corresponding latent variable’s approximate posterior. This method “amortizes” the process of fitting separate variational parameters for each data point across all data points, reducing computational cost. However, this efficiency comes at the cost of flexibility, and this approach may not match the optimal solution. Experiments on a wide range of hierarchical models—including linear and nonlinear probabilistic models, Bayesian neural networks, and saw time-series models—demonstrate that Amortized VI typically converges faster than traditional Factorized VI. Nevertheless, amortized VI can be less stable, sometimes failing to converge if the inference network is not sufficiently expressive or is poorly initialized. [41]

A particle-based approximation is represented by a set of samples (particles) that are updated in order to minimize the KL divergence. This approach allows variational inference to be more flexible—that is to say, non-parametric. The chosen metric (e.g., Stein Variational Gradient Descent, Stein Variational Newton, Gradient Flow with Stein Discrepancy, or Gradient Flow with Stein Fisher Information) is used to iteratively update the particles so that their empirical distribution gets closer to the posterior. [42][43] On the other hand, Energetic Particle Variational (EPV) approximates the density with particles and then takes the variation for particle updates. This method tends to outperform traditional particle-based methods. Moreover, experiments on toy problems—such as a modified Gaussian with a nonlinear transformation, a two-component “banana-shaped” distribution, a sinusoidal-modulated Gaussian, and a star-shaped distribution in $2D$ showed good performance. However, the more complex the distribution, the larger the number of particles and iterations needed. [44]

A variational autoencoders (VAE) couples two neural networks in a single probabilistic model. The encoder $q(z|x)$ is an amortised variational inference network that outputs a mean and variance providing a tractable approximation to the true posterior $p(x|z)$. The decoder using a simple prior (e.g., Gaussian) defines the joint distribution $p(x, z)$.

Training proceeds by maximizing the Evidence Lower Bound (ELBO) and keeping KL divergence small. In practice, training consists of encoding each data point x and reconstructing it as \hat{x} . Once trained, the encoder is ignored, and the decoder tries to produce realistic data points drawn from the learned distribution. [45] In terms of multi-source neural variational inference, the latent variable set z is shared. That is, each encoder maps its raw observation x to a $q(z|x)$ and all $q(z|x)$ are in the same latent space z . [46] The shortcomings of the method are that the global optimum of the ELBO can choose such $q(z|x)$ that allows only for easy approximation; therefore, the true posterior $p(x|z)$ will be modeled poorly. Also, the reconstruction from x to \hat{x} produces errors which adds latent dimensions. Although this tightens the bound, it does not genuinely improve the model’s fit to the data. [47]

Nested Variational Inference (NVI) introducing a sequence of intermediate distributions that gradually transform an easy-to-sample distribution into the complex target distribution. At each stage of transformation, NVI defines and optimizes forward and reverse proposal kernels to efficiently move samples through the sequence. By maximizing ELBO and minimizing KL divergences between these proposals and their corresponding intermediate targets, NVI ensures that samples remain representative. [48]

Boosting Variational Inference (BVI) approximates complex distributions by gradually building up a mixture of simple distributions (e.g., Gaussians). Instead of using just one simple approximation, BVI adds new components one by one, each time improving the overall fit. At each step, BVI chooses the next component—the one that, if added, would most reduce the difference between the current mixture and the true target distribution—by solving an optimization problem. It then updates the component’s weight to reduce the error between the mixture and the true distribution. [49]

One shortcoming of the above-mentioned methods is that if $q(z|x)$ is restricted to a simple family of distributions (e.g., Gaussians), it may not be flexible enough to approximate more complex distributions, which can influence model performance. Normalizing flows address this limitation by constructing complex densities through a sequence of invertible and differentiable transformations, as discussed in Section 3.2. Variational inference with normalizing flows therefore replaces $q(z|x)$ with a transformed base distribution that remains computationally cheap while being far more expressive. An important note is that any normalizing flow architecture that is invertible and differentiable can be used with variational inference. [10] Although there is a considerable body of research examining the performance of variational inference with different normalizing flow architectures, such as Singularity-Free Planar Flows [50], Sylvester Normalizing Flows [51], Inverse Autoregressive Flows [52], Normalizing Flows with Adaptive Surrogate [53], Neural Spline Flows [54], Block Neural Autoregressive Flows [55], and Mixtures of Normalizing Flows [56] it is believed that a more comprehensive and systematic investigation is still needed. A thorough comparative study could offer deeper insights into the strengths and limitations of the effectiveness of variational inference in diverse settings.

It is also noticeable that not much research has been conducted on high-dimensional variational inference. Many papers use only low-dimensional toy problems, which demonstrate some improvements in accuracy, do not necessarily push the potential of variational inference to its limits. The weaknesses of variational inference in high-dimensional spaces are determined by several factors:

Table 3: Challenges of variational inference in high-dimensional space

Challenge	Description
Simple variational posterior	The variational posterior is simple (e.g., Gaussian, with all dimensions assumed independent), which makes it difficult to capture the complexity of the true posterior. As a result, the probabilistic model can only fit data within the limits of the variational approximation [57].
Single-sample approximation	The ELBO is estimated from a single-sample approximation of the marginal likelihood. Samples from low-probability regions that do not explain the observation are discarded, so only the high-probability regions of the true posterior are covered. This results in overly simple estimated distributions [57].
Multi-sampling	Averaging over multiple samples (multi-sampling) can escape the single-sample trap, making the ELBO tighter and improving log-likelihood. However, using more samples tends to make deeper layers activate more latent variables. When the model has discrete latent variables, the gradient of this objective has variance, so the proposal distribution is not updated with that gradient [57].
Design choice	In the scientific literature, it is difficult to find guidance on the selection of the variational objective and the choice of the approximating family. [58].

To address the problems mentioned above, researchers have proposed methods for maximizing posterior-approximation accuracy by analyzing the tails of the density ratio between the joint distribution and its approximation. When this density ratio is heavy-tailed, it indicates a mismatch between the approximating and target distributions. In high-dimensional spaces, using heavy-tailed families of approximate distributions can help reduce this mismatch. The findings are limited and suggest that normalizing flows with the Pareto-k diagnostic and exclusive KL divergence should be used as metrics to measure the mismatch. [58]

Another paper outlines the problem of high variance in the stochastic gradients of normalizing flows. During training, samples are expected to exhibit high variance, which is difficult to stabilize. The authors show that soft clipping of the conditional factors and a bijective soft log transformation of the samples (LOFT) improve estimates of the ELBO and the marginal likelihood. Experiments were performed on the funnel distribution, multivariate Student-t distribution, multivariate Gaussian mixture, conjugate linear regression, and horseshoe logistic regression models, in dimensions ranging from 10 up to 4002. [59]

4.3 Comparison of Markov Chain Monte Carlo and Variational Inference

The third stream of research focuses on comparing MCMC and VI. Early comparisons have generally been successful, although they span very different settings. Some studies evaluate VI versus MCMC by measuring predictive log-likelihood on held-out data for simple models. [60] Others employ the total-variation error metric on simulated data to quantify the distance between the estimated and true distributions. [61] A further paper returns to predictive log-likelihood on held-out image datasets. [62] In all of these cases, VI and MCMC achieved similar accuracy, but VI was substantially faster. On the other hand, initial experiments on several models were conducted, namely, the Normal-Poisson GLMM, a random-slope linear model, a simulated Gaussian mixture model, and a finite Gaussian mixture model applied to real MNIST data using MCMC results as ground truth. Each method’s covariance estimates, standard deviation and runtimes were compared. Mean-field variational Bayes notoriously underestimates both covariance and standard deviation, whereas linear-response variational Bayes yields slightly more accurate estimates. [63]

In 2018, a study tested the applicability of VI to physics by comparing VI with MCMC across five problems:

- Annual-modulation search in COSINE-100 dark-matter data (6 parameters);
- Orbital-parameter recovery from a synthetic radial-velocity curve (7 parameters);
- Periodic-G claim (5 parameters);
- Spectral-lag break in GRB 160625B (3 parameters);
- Galaxy-cluster mass from a weak-lensing shear catalogue (2 parameters)

The results showed that VI produced parameter estimates that were virtually identical to those from MCMC, while requiring only a fraction of the computation time. [64]

Next study tests VI on three nonlinear ODE models. For the SIR epidemic and a protein-signalling pathway, VI reproduces the No-U-Turn Sampler (NUTS) posteriors almost exactly, while significantly reducing wall-clock time. For the stochastic Lotka-Volterra model, it matches the MCMC estimates. [65] Following paper investigates VI with coupling-layer normalizing flows against MCMC on two synthetic geophysical problems: 2-D travel-time tomography and transmission full-waveform inversion. In both cases, VI reproduces the MCMC posterior means and standard deviations almost exactly, yet requires far fewer forward simulations, about 400 times faster in the tomography test and still decisively quicker in the full-waveform case. [66] Attempts to analyze VI, MCMC, and Laplace approximation on synthetic light curves and the Foundation Supernova Survey DR1 are also presented. VI achieves MCMC-level accuracy in distance estimates while significantly reducing runtimes. [67] Another proposal evaluates VI with isotropic-Gaussian mixtures on two benchmarks — a 5-component mixture in 2-D and a 10-component mixture in 20-D - using reverse KL divergence as the evaluation metric; the proposed Isotropic Bures-Wasserstein (IBW) update, which projects the full Wasserstein-2 gradient onto the isotropic BW manifold, achieves accuracy on par with other VI baselines and Hamiltonian MC while reducing runtime. [68]

4.4 Conclusion of literature review

Current literature on VI can be grouped into three main streams:

- Introduction and applications across diverse domains: showing that VI can generate posterior samples faster than traditional methods while maintaining comparable accuracy.
- Methodological enhancements: extensions and refinements that build on the core approximation framework.
- Comparative studies versus MCMC: systematic evaluations that highlight where VI excels and where it still falls short.

Early work framed VI primarily as a remedy for big-data challenges—such as large document corpora, image processing, and even reinforcement-learning tasks. Only recently has it been applied to scientific problems in physics and biology. Yet many of these “new” demonstrations rely on simplified, toy examples, underscoring the need for deeper mathematical advances. The literature suggests several promising strategies for addressing VI’s current limitations, especially its handling of high dimensionality and multimodal posteriors, so that it can potentially meet the demands of gravitational-wave data analysis.

One possible solution is to test alternative divergences. The KL divergence is known to underestimate posterior variance; moreover, it can easily lose the bound and may struggle to capture symmetry when multimodal components lie close together.[32] For this reason, researchers have introduced metrics that, under certain conditions, tighten the lower bound and yield gradients with as little variance as possible: α -divergence[69], Renyi divergence[70], the generalized-divergence family[71] and Stein discrepancy.[72]

Another promising direction is to develop a more expressive approximation framework. Although VI is optimal relative to a specific objective function, it often proves inadequate for complex models. To address this, some researchers proposed a three-component optimization framework they call Generalized Variational Inference (GVI), and noted that much work remains to be done in this area.[73] Additionally, Implicit Variational Inference (IVI) defines variational distributions implicitly, offering greater flexibility in approximating complex multimodal and correlated posteriors, but it struggles with intractable density functions and unstable adversarial training. To address these issues, Linearised Implicit Variational Inference (LIVI) was introduced, which employs a novel approximate bound by locally linearizing the neural sampler. LIVI avoids unstable adversarial objectives and introduces a new sampler architecture capable of modeling implicit distributions in very high-dimensional latent-variable spaces. Future work could further extend LIVI by exploring methods such as independent hypernetworks at different layers to enable more accurate sampling in even higher-dimensional settings.[74]

Returning to normalizing flows, relatively little work has systematically compared the performance of VI across different normalizing-flow architectures. A careful investigation of the respective advantages and disadvantages would therefore be highly beneficial. Finally, tackling the problem of high variance in the stochastic gradients of normalizing flows also appears to be a promising avenue for future research.

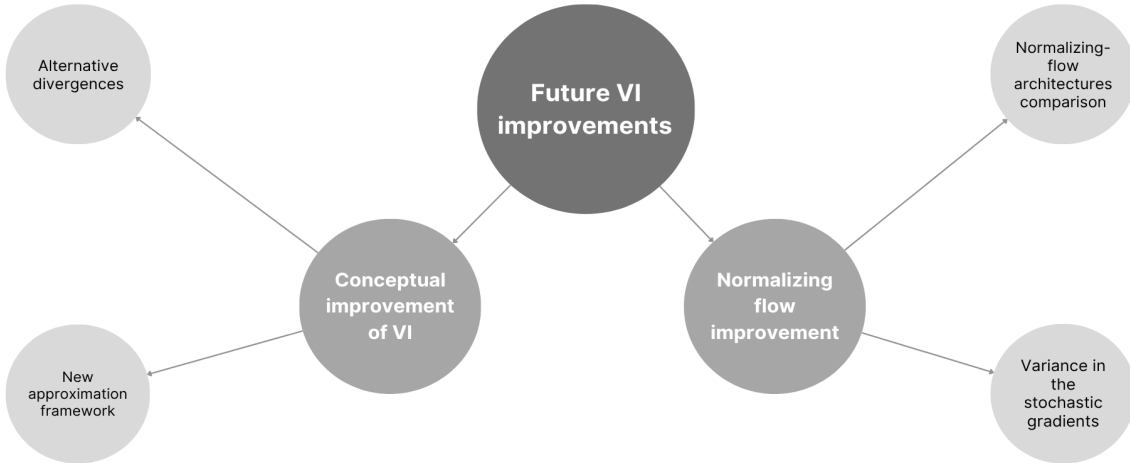


Figure 5: Possible areas for improving variational inference

5 Toy problem

5.1 Experiment setup

During the literature review, it became clear that:

- No previous work has applied VI to the analysis of gravitational-wave data.
- When a target distribution is complicated, researchers usually demonstrate VI only on low-dimensional toy problems.
- The benchmark datasets used to evaluate VI are far less complex than gravitational-wave signals.

These observations suggest that experimenting on VI on challenging, high-dimensional distributions could reveal the point at which (i) the conceptual limits of VI identified in the literature and (ii) the conceptual limits of normalizing flows begin to matter. Therefore, rather than tackling the computational challenges of gravitational-wave signals immediately, it was decided to carry out a simple toy experiment. In this experiment, the VI sampler is based on a Block Neural Autoregressive Flow (B-NAF) which is an extension of the NAF. B-NAF omits the separate “conditioner” network and instead learns the full transformation in a single masked network. In toy experiments, it was shown that B-NAF is able to capture both multi-modality and discontinuities, and shows better results than its competitors.[75] The flow was trained using the next parameters: (i) the depth of the internal neural network: 5, (ii) the width of each hidden layer, per block: 124, (iii) the number of full B-NAF transformations that are composed: 5.

The analytically specified log-likelihood (14) of a multivariate normal distribution with mean vector μ and covariance matrix Σ is used in two complementary ways: (i) to define a synthetic ‘ground-truth’ target distribution—allowing controlled experiments with varying μ and Σ ; (ii) to yield a closed-form objective ELBO that guides optimization

of the VI flow.

If the joint probability density function of the j -th term of the sequence can be shown as:[76]

$$f_X(x_j) = (2\pi)^{-K/2} |\det(V_0)|^{-1/2} \exp\left(-\frac{1}{2} (x_j - \mu_0)^\top V_0^{-1} (x_j - \mu_0)\right) \quad (12)$$

then the likelihood function is shown as: [76]

$$\begin{aligned} L(\mu, V; x_1, \dots, x_n) &= \prod_{j=1}^n f_X(x_j; \mu, V) \\ &= \prod_{j=1}^n (2\pi)^{-K/2} |\det(V)|^{-1/2} \exp\left(-\frac{1}{2} (x_j - \mu)^\top V^{-1} (x_j - \mu)\right) \\ &= (2\pi)^{-nK/2} |\det(V)|^{-n/2} \exp\left(-\frac{1}{2} \sum_{j=1}^n (x_j - \mu)^\top V^{-1} (x_j - \mu)\right) \end{aligned} \quad (13)$$

and the log-likelihood function shown as: [76]

$$\begin{aligned} l(\mu, V; x_1, \dots, x_n) &= \ln(L(\mu, V; x_1, \dots, x_n)) \\ &= \ln\left((2\pi)^{-nK/2} |\det(V)|^{-n/2} \exp\left(-\frac{1}{2} \sum_{j=1}^n (x_j - \mu)^\top V^{-1} (x_j - \mu)\right)\right) \\ &= -\frac{nK}{2} \ln(2\pi) - \frac{n}{2} \ln(\det(V)) - \frac{1}{2} \sum_{j=1}^n (x_j - \mu)^\top V^{-1} (x_j - \mu) \end{aligned} \quad (14)$$

To reduce the risk of the optimiser getting trapped in poor local optima and to obtain a tighter posterior approximation, annealed variational inference (AVI) was adapted. An inverse-temperature parameter $\beta \in (\beta_{\min}, 1]$ where β_{\min} was chosen as 0.49 that gradually sharpens ELBO during training.[77] Other chosen parameters for a toy problem are specified in Appendix C.

5.2 Results

The analysis of the results will consist of three parts:

- Visual inspection of the true and VI distributions.
- Comparison of the means, variances, and standard deviations of the true and VI distributions.
- Examination of the parametric and nonparametric KL divergences.

5.2.1 Visual inspection

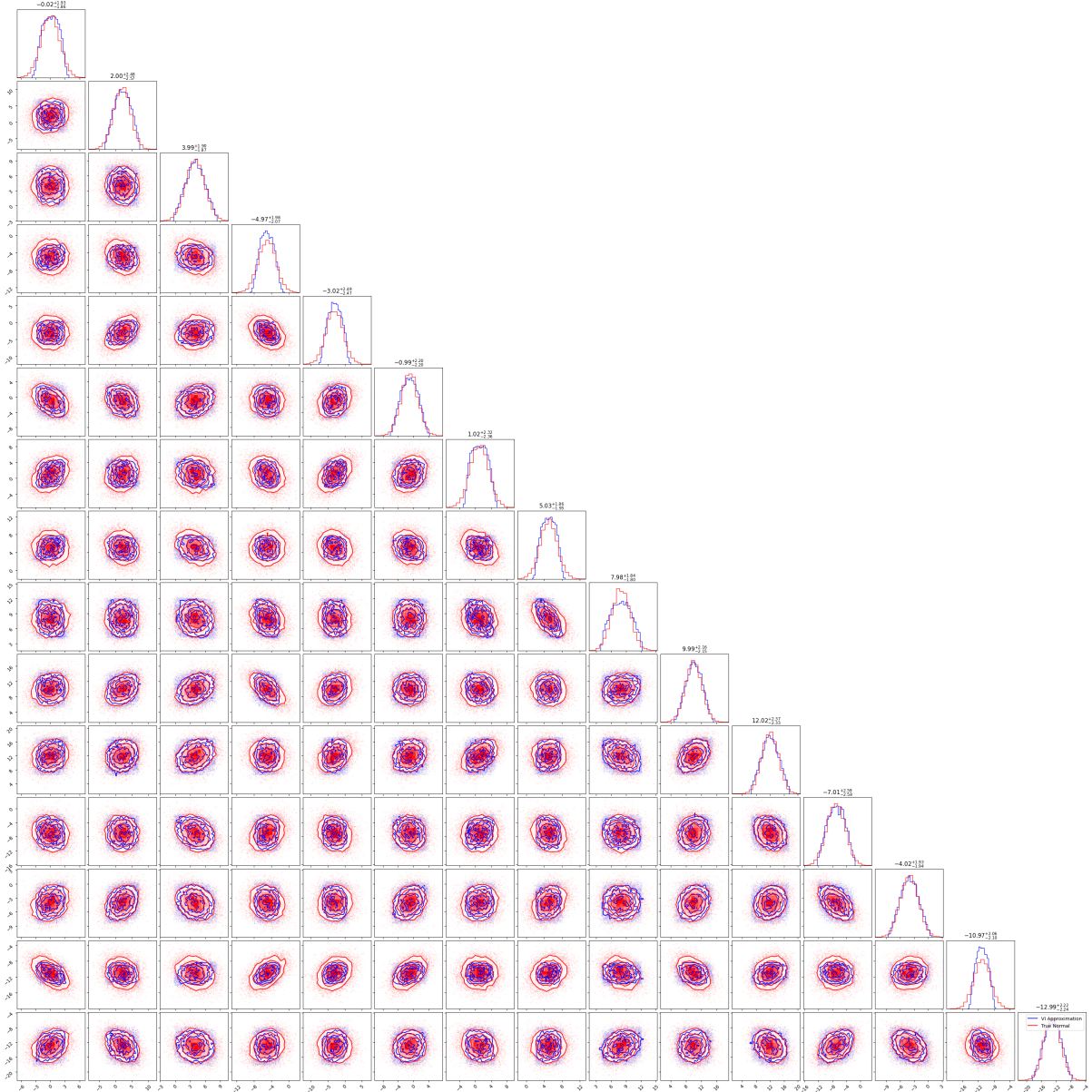


Figure 6: Possible areas for improving variational inference

Figure 6 shows that, with the training hyper-parameters listed in Appendix C, the VI sampler with a B-NAF is able to fit its VI samples to the true theoretical distribution. This result is expected: a single B-NAF layer is expressively rich enough to map one

single-component normal distribution to another single-component normal distribution. Even in this deliberately simple setting, however, a fundamental limitation of the KL divergence objective is already visible: the fitted VI posterior systematically underestimates the marginal variances. Minimising KL divergence penalises the variational density where the true posterior p is nearly zero, yet applies only a mild penalty when q fails to cover the tails of p . The optimiser therefore prefers a compact distribution that catches the mode, producing overly confident low-variance samples.

5.2.2 Comparison of the means, variances, and standard deviations of the true and VI distributions

Table 4: Comparison of VI and true sample statistics

Dimension	VI samples			True samples		
	Mean	Var	SD	Mean	Var	SD
1	0.0614	2.5555	1.5986	-0.0112	3.6635	1.9140
2	2.0533	5.6505	2.3771	1.9612	6.6014	2.5693
3	3.9897	3.3859	1.8401	4.0035	3.6895	1.9208
4	-5.0651	2.2688	1.5063	-4.9962	4.1922	2.0475
5	-2.9925	3.6618	1.9136	-3.0005	6.3682	2.5235
6	-1.0461	4.5242	2.1270	-0.9931	4.9123	2.2164
7	1.0550	3.6699	1.9157	1.0018	5.4890	2.3429
8	4.9813	2.2343	1.4948	5.0134	3.8343	1.9581
9	8.0078	3.8837	1.9707	7.9951	3.3659	1.8346
10	10.0725	4.1932	2.0477	9.9790	4.6700	2.1610
11	12.0035	5.7336	2.3945	12.0289	5.5879	2.3639
12	-6.9325	5.8011	2.4085	-7.0155	6.7481	2.5977
13	-4.0024	3.5331	1.8797	-4.0233	3.7473	1.9358
14	-11.0983	2.0343	1.4263	-10.9813	4.4284	2.1044
15	-12.9630	4.8264	2.1969	-12.9927	5.0670	2.2510

Taking into consideration the results from Table 4, which compares the means, variances, and standard deviations of the true and VI distributions, and from Table 5, which reports the absolute differences in these statistics, the following can be stated: (i) across all 15 dimensions, the absolute difference between the VI and true sample means never exceeds 0.07; (ii) except for dimensions 9 and 11, every VI variance is smaller than its true counterpart, and the same pattern holds for the standard deviations. These empirical findings reinforce the discussion following Figure 6 about the nature of KL divergence and its limitations.

Table 5: Per-dimension absolute differences between VI and true sample statistics.

Dim.	$ \mu_{\text{VI}} - \mu_{\text{true}} $	$ \sigma_{\text{VI}}^2 - \sigma_{\text{true}}^2 $	$ \sigma_{\text{VI}} - \sigma_{\text{true}} $
1	0.0726	1.1080	0.3154
2	0.0921	0.9509	0.1922
3	0.0138	0.3036	0.0807
4	0.0689	1.9234	0.5412
5	0.0080	2.7064	0.6099
6	0.0530	0.3881	0.0894
7	0.0532	1.8191	0.4272
8	0.0321	1.6000	0.4633
9	0.0127	0.5178	0.1361
10	0.0935	0.4768	0.1133
11	0.0254	0.1457	0.0306
12	0.0830	0.9470	0.1892
13	0.0209	0.2142	0.0561
14	0.1170	2.3941	0.6781
15	0.0297	0.2406	0.0541

5.2.3 Examination of the parametric and nonparametric KL divergences.

Table 6: Kullback–Leibler divergence estimates

Type of KL divergence	Result (nats)
Parametric	0.1511
kNN naïve	0.0208
kNN (SciPy)	0.0208
kNN (scikit-learn)	0.0208
kNN (scikit-learn, fast)	0.0208

To quantify the difference between the VI approximation and the true distribution, both parametric and non-parametric KL divergence were evaluated. Because the reliability of parametric KL estimates in high-dimensional spaces has been questioned, recent work has introduced k-nearest-neighbour (k-NN) non-parametric KL metrics which were taken into account. With annealed variational inference, the KL divergence was reduced from 0.95 to 0.15. Training experiments also revealed that the learning rate was the hyper-parameter to which the KL metrics were most sensitive. Job wall-clock time of the experiment is: 00:01:19. The code and all relevant information can be found at the following link: <https://github.com/OleksiiBevza/Variational-inference-scheme-for-gravitational-waves-parameter-estimation>

6 Conclusion

After a thorough review of the literature, it became clear that VI cannot be applied directly to the gravitational-wave analysis challenges of the Einstein Telescope. Nor can VI be used straightforwardly on synthetic gravitational-wave data, owing to conceptual constraints that were identified. To pinpoint where these limitations emerge, a toy problem was explored: a 15-dimensional, single-component multivariate normal distribution with different means vector μ and covariances matrix Σ . This experiment exposed a fundamental shortcoming of KL-divergence-based VI: the fitted VI posteriors systematically underestimate the marginal variances. Although this issue could be alleviated in the toy setting by using annealed variational inference, doing so will likely prove much harder for more complex problems.

7 Future work

Based on the research conducted so far, the direct application of VI to gravitational-wave data analysis is not recommended at present moment. A more promising strategy is to apply VI to progressively more complex toy problems in order to map out its limits. The literature review (Section 4.4) proposes several conceptual improvements that could be incorporated. At present, the biggest challenge is to remedy the variance underestimation exhibited by VI. Some authors advocate replacing the KL divergence with alternative measures.[69][70][71][72] Recent theoretical and empirical work likewise points to two main strategies for mitigating this: (i) changing the optimization objective and (ii) adopting a different variational family.[78]

8 Appendixes

8.1 Appendix A

Kullback-Leibler Divergence

Kullback-Leibler Divergence, also known as relative entropy, is a metric that measures the distance between two given distributions. Assuming the distributions $p(x)$ and $q(x)$ are given, the relative entropy denoted as H can be calculated as follows [79][80]

$$\begin{aligned} Hp(x) &= -\sum p(x) * \log p(x) \\ Hq(z) &= -\sum q(z) * \log q(z) \end{aligned} \quad (1)$$

Then Kullback-Leibler Divergence of $p(x)$ with respect to $q(x)$ is given as: [80]

$$\begin{aligned} KL(p(x)||q(x)) &= \left[-\sum q(x) * \log\{q(x)\} \right] - \left[-\sum p(x) * \log\{p(x)\} \right] \\ &= \sum p(x) * \log\{p(x)\} - \sum p(x) * \log\{q(x)\} \\ &= \sum p(x) [\log\{p(x)\} - \log\{q(x)\}] \\ &= \sum p(x) * \log \left\{ \frac{p(x)}{q(x)} \right\} \end{aligned} \quad (2)$$

It is important to recognize that, for multivariate models, where p and q are multivariate normal distributions, modified Kullback-Leibler Divergence takes the form: [80][81][82]

$$\begin{aligned} p : x &\sim \mathcal{N}(\mu_1, \Sigma_1) \\ q : x &\sim \mathcal{N}(\mu_2, \Sigma_2) \end{aligned} \quad (3)$$

Then,

$$\begin{aligned} KL[P||Q] &= \int_{\mathcal{X}} p(x) \ln \frac{p(x)}{q(x)} dx \\ &= \int_{R^n} \mathcal{N}(x; \mu_1, \Sigma_1) \ln \frac{\mathcal{N}(x; \mu_1, \Sigma_1)}{\mathcal{N}(x; \mu_2, \Sigma_2)} dx \\ &= \left\langle \ln \frac{\mathcal{N}(x; \mu_1, \Sigma_1)}{\mathcal{N}(x; \mu_2, \Sigma_2)} \right\rangle_{p(x)} \\ &= \left\langle \ln \frac{\frac{1}{\sqrt{(2\pi)^n |\Sigma_1|}} \cdot \exp \left[-\frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right]}{\frac{1}{\sqrt{(2\pi)^n |\Sigma_2|}} \cdot \exp \left[-\frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right]} \right\rangle_{p(x)} \\ &= \left\langle \frac{1}{2} \ln \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) + \frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right\rangle_{p(x)} \\ &= \frac{1}{2} \left\langle \ln \frac{|\Sigma_2|}{|\Sigma_1|} - (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) + (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right\rangle_{p(x)}. \end{aligned} \quad (4)$$

$$\begin{aligned}
\text{KL}[P\|Q] &= \frac{1}{2} \left(\ln \frac{|\Sigma_2|}{|\Sigma_1|} - \text{tr} \left[\Sigma_1^{-1} \left\langle (x - \mu_1)(x - \mu_1)^T \right\rangle_{p(x)} \right] \right. \\
&\quad \left. + \text{tr} \left[\Sigma_2^{-1} \left\langle xx^T - 2\mu_2 x^T + \mu_2 \mu_2^T \right\rangle_{p(x)} \right] \right) \\
&= \frac{1}{2} \left(\ln \frac{|\Sigma_2|}{|\Sigma_1|} - \text{tr} \left[\Sigma_1^{-1} \left\langle (x - \mu_1)(x - \mu_1)^T \right\rangle_{p(x)} \right] \right. \\
&\quad \left. + \text{tr} \left[\Sigma_2^{-1} \left(\left\langle xx^T \right\rangle_{p(x)} - \left\langle 2\mu_2 x^T \right\rangle_{p(x)} + \left\langle \mu_2 \mu_2^T \right\rangle_{p(x)} \right) \right] \right)
\end{aligned} \tag{5}$$

if

$$x \sim \mathcal{N}(\mu, \Sigma) \quad \Rightarrow \quad \langle x^T A x \rangle = \mu^T A \mu + \text{tr}(A \Sigma) \tag{6}$$

then

$$\begin{aligned}
\text{KL}[P\|Q] &= \frac{1}{2} \left(\ln \frac{|\Sigma_2|}{|\Sigma_1|} - \text{tr} [\Sigma_1^{-1} \Sigma_1] + \text{tr} [\Sigma_2^{-1} (\Sigma_1 + \mu_1 \mu_1^T - 2\mu_2 \mu_1^T + \mu_2 \mu_2^T)] \right) \\
&= \frac{1}{2} \left(\ln \frac{|\Sigma_2|}{|\Sigma_1|} - \text{tr} [I_n] + \text{tr} [\Sigma_2^{-1} \Sigma_1] + \text{tr} [\Sigma_2^{-1} (\mu_1 \mu_1^T - 2\mu_2 \mu_1^T + \mu_2 \mu_2^T)] \right) \\
&= \frac{1}{2} \left(\ln \frac{|\Sigma_2|}{|\Sigma_1|} - n + \text{tr} [\Sigma_2^{-1} \Sigma_1] + \text{tr} [\mu_1^T \Sigma_2^{-1} \mu_1 - 2\mu_1^T \Sigma_2^{-1} \mu_2 + \mu_2^T \Sigma_2^{-1} \mu_2] \right) \\
&= \frac{1}{2} \left[\ln \frac{|\Sigma_2|}{|\Sigma_1|} - n + \text{tr} [\Sigma_2^{-1} \Sigma_1] + (\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) \right].
\end{aligned} \tag{7}$$

so that the final Kullback-Leibler divergence for the multivariate normal distribution can be shown as:

$$\text{KL}[P\|Q] = \frac{1}{2} \left[(\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) + \text{tr} (\Sigma_2^{-1} \Sigma_1) - \ln \frac{|\Sigma_1|}{|\Sigma_2|} - n \right]. \tag{8}$$

Since histogram-based density estimation tends to struggle when evaluating high-dimensional distributions, some researchers have proposed a method that estimates divergence directly from k-nearest-neighbor distances between samples by:

- Adaptive kNN : Choosing k adaptively at different sample points.
- Whitening: Applying a linear transformation to the data to reduce bias.

Even though k-nearest-neighbor methods can be implemented easily for high-dimensional data, they still become unreliable as the number of dimensions grows too large. To address this, it has been suggested that instead of treating all dimensions equally, one should use a criterion to select only the most relevant dimensions for each query. [83][84][85]

8.2 Appendix B

Lower bound

Assuming two sets of variables,

- 1) $x = (x_1, x_2, x_3, x_4, x_5)$
- 2) $z = (z_1, z_2, z_3, z_4, z_5)$

the joint probability density function is presented as:

$$p(x, z) = p(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5) \quad (1)$$

so that

$$p(z | x) = \frac{p(x, z)}{p(x)} = \frac{p(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5)}{\iiint \iint p(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5) dz_1 dz_2 dz_3 dz_4 dz_5}. \quad (2)$$

where

- $p(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5)$ is known
- and $p(z_1, z_2, z_3, z_4, z_5 | x_1, x_2, x_3, x_4, x_5)$ is unknown.

Let $q(z_1, z_2, z_3, z_4, z_5)$ be used to estimate the unknown $p(z | x)$, so that $KL + L = \ln p(x)$, and the lower bound L can be shown as:

$$\begin{aligned} L &= \sum_z q(z) \ln \left\{ \frac{p(x, z)}{q(z)} \right\} \\ &= \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1, z_2, z_3, z_4, z_5) \ln \left\{ \frac{p(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5)}{q(z_1, z_2, z_3, z_4, z_5)} \right\} \end{aligned} \quad (3)$$

The idea is to find such a $q(z_1, z_2, z_3, z_4, z_5)$ that maximizes L by assuming independence of the variables z_1, z_2, z_3, z_4 , and z_5 , so that $q(z_1, z_2, z_3, z_4, z_5)$ can be written as:

$$q(z_1, z_2, z_3, z_4, z_5) = q(z_1) * q(z_2) * q(z_3) * q(z_4) * q(z_5) = \prod_{i=1}^5 q(z_i) \quad (4)$$

Substituting $q(z_1, z_2, z_3, z_4, z_5)$ back into the original equation gives:

$$\begin{aligned}
L &= \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1)q(z_2)q(z_3)q(z_4)q(z_5) \ln \left\{ \frac{p(x, z)}{q(z_1)q(z_2)q(z_3)q(z_4)q(z_5)} \right\} \\
&= \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1)q(z_2)q(z_3)q(z_4)q(z_5) \left[\ln\{p(x, z)\} - \ln\{q(z_1)q(z_2)q(z_3)q(z_4)q(z_5)\} \right] \\
&= \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1)q(z_2)q(z_3)q(z_4)q(z_5) \left[\ln\{p(x, z)\} - \ln\{q(z_1)\} - \ln\{q(z_2)\} - \ln\{q(z_3)\} \right. \\
&\quad \left. - \ln\{q(z_4)\} - \ln\{q(z_5)\} \right] \\
&= \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1)q(z_2)q(z_3)q(z_4)q(z_5) \left[\ln\{p(x, z)\} - \ln\{q(z_1)\} - (\ln\{q(z_2)\} + \ln\{q(z_3)\} \right. \\
&\quad \left. + \ln\{q(z_4)\} + \ln\{q(z_5)\}) \right] \\
&= \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1)q(z_2)q(z_3)q(z_4)q(z_5) \ln\{p(x, z)\} \\
&\quad - \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1)q(z_2)q(z_3)q(z_4)q(z_5) \ln\{q(z_1)\} \\
&\quad - \sum_{z_1} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_1)q(z_2)q(z_3)q(z_4)q(z_5) \left[\ln\{q(z_2)\} + \ln\{q(z_3)\} + \ln\{q(z_4)\} + \ln\{q(z_5)\} \right] \\
&= \sum_{z_1} q(z_1) \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_2)q(z_3)q(z_4)q(z_5) \ln\{p(x, z)\} \\
&\quad - \sum_{z_1} q(z_1) \ln\{q(z_1)\} \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_2)q(z_3)q(z_4)q(z_5) \\
&\quad - \sum_{z_1} q(z_1) \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_2)q(z_3)q(z_4)q(z_5) \left[\ln\{q(z_2)\} + \ln\{q(z_3)\} + \ln\{q(z_4)\} + \ln\{q(z_5)\} \right] \\
&= \sum_{z_1} q(z_1) \sum_{z_2} \sum_{z_3} \sum_{z_4} \sum_{z_5} q(z_2)q(z_3)q(z_4)q(z_5) \ln\{p(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5)\} \\
&\quad - \sum_{z_1} q(z_1) \ln\{q(z_1)\} - \sum_{z_1} q(z_1) * constant
\end{aligned} \tag{9}$$

$$\begin{aligned}
L &= \sum_{z_1} q(z_1) E_{z_2 z_3 z_4 z_5} [\ln \{p(z_1, z_2, z_3, z_4, z_5, x_1, x_2, x_3, x_4, x_5)\}] \\
&\quad - \sum_{z_1} q(z_1) \ln\{q(z_1)\} - \sum_{z_1} q(z_1) * \text{constant} \\
&= \sum_{z_1} q(z_1) E_{z_2 z_3 z_4 z_5} [\ln \{p(z_1, z_2, z_3, z_4, z_5, x_1, x_2, x_3, x_4, x_5)\} - \text{constant}] \\
&\quad - \sum_{z_1} q(z_1) \ln\{q(z_1)\} \\
&= \sum_{z_1} q(z_1) [E_{z_2 z_3 z_4 z_5} [\ln \{p(z_1, z_2, z_3, z_4, z_5, x_1, x_2, x_3, x_4, x_5)\}] + \text{constant}_1 + \text{constant}_2] \\
&\quad - \sum_{z_1} q(z_1) \ln\{q(z_1)\} \\
&= \sum_{z_1} q(z_1) [\ln\{f(x, z)\} - \text{constant}_2] - \sum_{z_1} q(z_1) \ln\{q(z_1)\} \\
&= \sum_{z_1} q(z_1) \ln\{f(x, z)\} - \sum_{z_1} q(z_1) \ln\{q(z_1)\} - \text{constant}_2 \sum_{z_1} q(z_1) \\
&= \sum_{z_1} q(z_1) * [\ln \{f(x, z) - \ln \{q(z_1)\} - \text{constant}_2] \\
&= \sum_{z_1} q(z_1) \left[\ln \left\{ \frac{f(x, z)}{q(z_1)} \right\} \right] + \text{constant} \tag{10}
\end{aligned}$$

8.3 Appendix C

Toy problem parameter specification

Table 7: VI training parameters

Parameter name	Chosen parameter	Description
Dimensions	15	The number of dimensions taken into account during the experiment.
Steps	10000	The number of optimisation updates to be performed during VI training.
Batch size	2000	Number of samples drawn from the normalising flow at each optimisation step.
Learning rate	0.065	Step size for the optimiser used to train the normalising flow.

Table 8: Experiment parameters

Parameter name	Mean	Prior bounds	
		Lower	Upper
dim1	0	-5.5	5.5
dim2	2	-3.5	7.5
dim3	4	-1.5	9.5
dim4	-5	-10.5	0.5
dim5	-3	-8.5	2.5
dim6	-1	-6.5	4.5
dim7	1	-4.5	6.5
dim8	5	-0.5	10.5
dim9	8	2.5	13.5
dim10	10	4.5	15.5
dim11	12	6.5	17.5
dim12	-7	-12.5	-1.5
dim13	-4	-9.5	1.5
dim14	-11	-16.5	-5.5
dim15	-13	-18.5	-7.5

8.4 Appendix D

Glossary

Table 9: Abbreviations and their meanings

Abbreviation	Meaning
VI	variational inference
MCMC	Markov Chain Monte Carlo
ELBO	the evidence lower bound
DELBO	distributed evidence lower bound
KL divergence	Kullback–Leibler divergence
SVI	stochastic variational inference
VAE	variational autoencoders
NVI	nested variational inference
BVI	boosting variational inference
ODE	ordinary differential equations
IBW	Isotropic Bures-Wasserstein
GVI	Generalized variational inference
LIVI	linearised implicit variational inference
B-NAF	Block Neural Autoregressive Flow
AVI	annealed variational inference

References

- [1] S. Bagnasco, A. Bozzi, T. Fragos, A. Gonzalvez, S. Hahn, G. Hemming, L. Lavezzi, P. Laycock, G. Merino, S. Pardi, S. Schramm, A. Stahl, A. Tanasijczuk, N. Tonello, S. Vallero, J. Veitch, and P. Verdier, “Computing challenges for the einstein telescope project,” 2023. [Online]. Available: <https://arxiv.org/abs/2312.11103>
- [2] Q. Hu and J. Veitch, “Costs of bayesian parameter estimation in third-generation gravitational wave detectors: a review of acceleration methods,” 2025. [Online]. Available: <https://arxiv.org/abs/2412.02651>
- [3] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” 2021. [Online]. Available: <https://arxiv.org/abs/1912.02762>
- [4] I. Kobyzev, S. J. Prince, and M. A. Brubaker, “Normalizing flows: An introduction and review of current methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, p. 3964–3979, Nov. 2021. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2020.2992934>
- [5] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, “Neural autoregressive flows,” 2018. [Online]. Available: <https://arxiv.org/abs/1804.00779>
- [6] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, “Neural spline flows,” 2019. [Online]. Available: <https://arxiv.org/abs/1906.04032>
- [7] D. J. Rezende, G. Papamakarios, S. Racanière, M. S. Albergo, G. Kanwar, P. E. Shanahan, and K. Cranmer, “Normalizing flows on tori and spheres,” 2020. [Online]. Available: <https://arxiv.org/abs/2002.02428>
- [8] R. T. Q. Chen, J. Behrmann, D. Duvenaud, and J.-H. Jacobsen, “Residual flows for invertible generative modeling,” 2020. [Online]. Available: <https://arxiv.org/abs/1906.02735>
- [9] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, “Ot-flow: Fast and accurate continuous normalizing flows via optimal transport,” 2021. [Online]. Available: <https://arxiv.org/abs/2006.00104>
- [10] D. J. Rezende and S. Mohamed, “Variational inference with normalizing flows,” 2016. [Online]. Available: <https://arxiv.org/abs/1505.05770>
- [11] L. Dinh, D. Krueger, and Y. Bengio, “Nice: Non-linear independent components estimation,” 2015. [Online]. Available: <https://arxiv.org/abs/1410.8516>
- [12] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp,” 2017. [Online]. Available: <https://arxiv.org/abs/1605.08803>
- [13] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” 2018. [Online]. Available: <https://arxiv.org/abs/1807.03039>
- [14] F. Ernst, L. Favaro, C. Krause, T. Plehn, and D. Shih, “Normalizing flows for high-dimensional detector simulations,” *SciPost Physics*, vol. 18, no. 3, Mar. 2025. [Online]. Available: <http://dx.doi.org/10.21468/SciPostPhys.18.3.081>

- [15] C. Gao, J. Isaacson, and C. Krause, “`flowi/ttgi`: High-dimensional integration and sampling with normalizing flows,” *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045023, Nov. 2020. [Online]. Available: <http://dx.doi.org/10.1088/2632-2153/abab62>
- [16] K. W. K. Wong, M. Isi, and T. D. P. Edwards, “Fast gravitational wave parameter estimation without compromises,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.05333>
- [17] M. Gabri  , G. M. Rotskoff, and E. Vanden-Eijnden, “Adaptive monte carlo augmented with normalizing flows,” *Proceedings of the National Academy of Sciences*, vol. 119, no. 10, Mar. 2022. [Online]. Available: <http://dx.doi.org/10.1073/pnas.2109420119>
- [18] I. M. Romero-Shaw, C. Talbot, S. Biscoveanu, V. D’Emilio, G. Ashton, C. P. L. Berry, S. Coughlin, S. Galaudage, C. Hoy, M. H  bner, K. S. Phukon, M. Pitkin, M. Rizzo, N. Sarin, R. Smith, S. Stevenson, A. Vajpeyi, M. Ar  ne, K. Athar, S. Banagiri, N. Bose, M. Carney, K. Chatziioannou, J. A. Clark, M. Colleoni, R. Cotesta, B. Edelman, H. Estell  s, C. Garc  a-Quir  s, A. Ghosh, R. Green, C.-J. Haster, S. Husa, D. Keitel, A. X. Kim, F. Hernandez-Vivanco, I. Maga  a Hernandez, C. Karathanasis, P. D. Lasky, N. De Lillo, M. E. Lower, D. Macleod, M. Mateu-Lucena, A. Miller, M. Millhouse, S. Morisaki, S. H. Oh, S. Ossokine, E. Payne, J. Powell, G. Pratten, M. P  rrer, A. Ramos-Buades, V. Raymond, E. Thrane, J. Veitch, D. Williams, M. J. Williams, and L. Xiao, “Bayesian inference for compact binary coalescences with `scipy/bilby/scp`: validation and application to the first ligo–virgo gravitational-wave transient catalogue,” *Monthly Notices of the Royal Astronomical Society*, vol. 499, no. 3, p. 3295–3319, Sep. 2020. [Online]. Available: <http://dx.doi.org/10.1093/mnras/staa2850>
- [19] G. Ashton, M. H  bner, P. D. Lasky, C. Talbot, K. Ackley, S. Biscoveanu, Q. Chu, A. Divakarla, P. J. Easter, B. Goncharov, F. H. Vivanco, J. Harms, M. E. Lower, G. D. Meadors, D. Melchor, E. Payne, M. D. Pitkin, J. Powell, N. Sarin, R. J. E. Smith, and E. Thrane, “Bilby: A user-friendly bayesian inference library for gravitational-wave astronomy,” *The Astrophysical Journal Supplement Series*, vol. 241, no. 2, p. 27, Apr. 2019. [Online]. Available: <http://dx.doi.org/10.3847/1538-4365/ab06fc>
- [20] M. J. Williams, J. Veitch, and C. Messenger, “Nested sampling with normalizing flows for gravitational-wave inference,” *Physical Review D*, vol. 103, no. 10, May 2021. [Online]. Available: <http://dx.doi.org/10.1103/PhysRevD.103.103006>
- [21] K. W. K. Wong, M. Gabri  , and D. Foreman-Mackey, “flowmc: Normalizing-flow enhanced sampling package for probabilistic inference in jax,” 2022. [Online]. Available: <https://arxiv.org/abs/2211.06397>
- [22] T. Wouters, P. T. H. Pang, T. Dietrich, and C. V. D. Broeck, “Robust parameter estimation within minutes on gravitational wave signals from binary neutron star inspirals,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.11397>
- [23] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American Statistical Association*

- ciation*, vol. 112, no. 518, p. 859–877, Apr. 2017. [Online]. Available: <http://dx.doi.org/10.1080/01621459.2017.1285773>
- [24] J. Grimmer, “An introduction to bayesian inference via variational approximations,” *Political Analysis*, vol. 19, no. 1, pp. 32–47, 2011.
 - [25] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine learning*, vol. 37, pp. 183–233, 1999.
 - [26] L. P. Cinelli, M. A. Marins, E. A. B. Da Silva, and S. L. Netto, *Variational methods for machine learning with applications to deep networks*. Springer, 2021, vol. 15.
 - [27] D. P. Kingma *et al.*, *Variational inference & deep learning: A new synthesis*, 2017.
 - [28] S. Nakajima, K. Watanabe, and M. Sugiyama, *Variational Bayesian learning theory*. Cambridge University Press, 2019.
 - [29] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.
 - [30] A. Ganguly and S. W. F. Earp, “An introduction to variational inference,” 2021. [Online]. Available: <https://arxiv.org/abs/2108.13083>
 - [31] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
 - [32] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, “Advances in variational inference,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 2008–2026, 2018.
 - [33] M. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” 2013. [Online]. Available: <https://arxiv.org/abs/1206.7051>
 - [34] A. K. Dhaka, A. Catalina, M. R. Andersen, M. Magnusson, J. H. Huggins, and A. Vehtari, “Robust, accurate stochastic optimization for variational inference,” 2020. [Online]. Available: <https://arxiv.org/abs/2009.00666>
 - [35] T. Weber, N. Heess, A. Eslami, J. Schulman, D. Wingate, and D. Silver, “Reinforced variational inference,” in *Advances in Neural Information Processing Systems (NIPS) Workshops*, 2015, pp. 33–90.
 - [36] Y. Teh, D. Newman, and M. Welling, “A collapsed variational bayesian inference algorithm for latent dirichlet allocation,” *Advances in neural information processing systems*, vol. 19, 2006.
 - [37] T. Campbell and B. Beronov, “Sparse variational inference: Bayesian coresets from scratch,” 2019. [Online]. Available: <https://arxiv.org/abs/1906.03329>
 - [38] P. Paritosh, N. Atanasov, and S. Martinez, “Distributed variational inference for online supervised learning,” *arXiv preprint arXiv:2309.02606*, 2023.
 - [39] F. Guo, X. Wang, K. Fan, T. Broderick, and D. B. Dunson, “Boosting variational inference,” 2017. [Online]. Available: <https://arxiv.org/abs/1611.05559>
 - [40] J. Domke and D. Sheldon, “Importance weighting and variational inference,” 2018. [Online]. Available: <https://arxiv.org/abs/1808.09034>

- [41] C. C. Margossian and D. M. Blei, “Amortized variational inference: When and why?” 2024. [Online]. Available: <https://arxiv.org/abs/2307.11018>
- [42] C. Liu, J. Zhuo, P. Cheng, R. Zhang, J. Zhu, and L. Carin, “Understanding and accelerating particle-based variational inference,” 2019. [Online]. Available: <https://arxiv.org/abs/1807.01750>
- [43] Z. Cheng, S. Zhang, L. Yu, and C. Zhang, “Particle-based variational inference with generalized wasserstein gradient flow,” 2023. [Online]. Available: <https://arxiv.org/abs/2310.16516>
- [44] Y. Wang, J. Chen, C. Liu, and L. Kang, “Particle-based energetic variational inference,” *Statistics and Computing*, vol. 31, no. 3, Apr. 2021. [Online]. Available: <http://dx.doi.org/10.1007/s11222-021-10009-7>
- [45] D. P. Kingma and M. Welling, “An introduction to variational autoencoders,” *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, p. 307–392, 2019. [Online]. Available: <http://dx.doi.org/10.1561/22000000056>
- [46] R. Kurle, S. Günnemann, and P. van der Smagt, “Multi-source neural variational inference,” 2018. [Online]. Available: <https://arxiv.org/abs/1811.04451>
- [47] Y. Yacoby, W. Pan, and F. Doshi-Velez, “Failure modes of variational autoencoders and their effects on downstream tasks,” 2022. [Online]. Available: <https://arxiv.org/abs/2007.07124>
- [48] H. Zimmermann, H. Wu, B. Esmaeili, and J.-W. van de Meent, “Nested variational inference,” 2021. [Online]. Available: <https://arxiv.org/abs/2106.11302>
- [49] T. Campbell and X. Li, “Universal boosting variational inference,” 2019. [Online]. Available: <https://arxiv.org/abs/1906.01235>
- [50] J. Ouyang, M. Gong, and H. Bondell, “Variational inference with singularity-free planar flows,” 2024. [Online]. Available: <https://openreview.net/forum?id=Gk75gOjtQh>
- [51] R. van den Berg, L. Hasenclever, J. M. Tomczak, and M. Welling, “Sylvester normalizing flows for variational inference,” 2019. [Online]. Available: <https://arxiv.org/abs/1803.05649>
- [52] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, “Improving variational inference with inverse autoregressive flow,” 2017. [Online]. Available: <https://arxiv.org/abs/1606.04934>
- [53] Y. Wang, F. Liu, and D. E. Schiavazzi, “Variational inference with nofas: Normalizing flow with adaptive surrogate for computationally expensive models,” *Journal of Computational Physics*, vol. 467, p. 111454, 2022.
- [54] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, “Neural spline flows,” *Advances in neural information processing systems*, vol. 32, 2019.
- [55] N. De Cao, W. Aziz, and I. Titov, “Block neural autoregressive flow,” in *Uncertainty in artificial intelligence*. PMLR, 2020, pp. 1263–1273.

- [56] G. G. P. F. Pires and M. A. T. Figueiredo, “Variational mixture of normalizing flows,” 2020. [Online]. Available: <https://arxiv.org/abs/2009.00585>
- [57] A. Mnih and D. J. Rezende, “Variational inference for monte carlo objectives,” 2016. [Online]. Available: <https://arxiv.org/abs/1602.06725>
- [58] A. K. Dhaka, A. Catalina, M. Welandawe, M. R. Andersen, J. Huggins, and A. Vehtari, “Challenges and opportunities in high-dimensional variational inference,” 2021. [Online]. Available: <https://arxiv.org/abs/2103.01085>
- [59] D. Andrade, “Stable training of normalizing flows for high-dimensional variational inference,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.16408>
- [60] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, “Automatic differentiation variational inference,” 2016. [Online]. Available: <https://arxiv.org/abs/1603.00788>
- [61] M. Braun and J. McAuliffe, “Variational inference for large-scale models of discrete choice,” *Journal of the American Statistical Association*, vol. 105, no. 489, p. 324–335, Mar. 2010. [Online]. Available: <http://dx.doi.org/10.1198/jasa.2009.tm08030>
- [62] D. M. Blei and M. I. Jordan, “Variational inference for dirichlet process mixtures,” *Bayesian Analysis, Bayesian Anal.*, 2006.
- [63] R. Giordano, T. Broderick, and M. Jordan, “Linear response methods for accurate covariance estimates from mean field variational bayes,” 2015. [Online]. Available: <https://arxiv.org/abs/1506.04088>
- [64] G. Gunapati, A. Jain, P. K. Srijith, and S. Desai, “Variational inference as an alternative to mcmc for parameter estimation and model selection,” *Publications of the Astronomical Society of Australia*, vol. 39, 2022. [Online]. Available: <http://dx.doi.org/10.1017/pasa.2021.64>
- [65] S. Ghosh, P. Birrell, and D. De Angelis, “Variational inference for nonlinear ordinary differential equations,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2719–2727.
- [66] X. Zhang, M. A. Nawaz, X. Zhao, and A. Curtis, *An introduction to variational inference in geophysical inverse problems*. Elsevier, 2021, p. 73–140. [Online]. Available: <http://dx.doi.org/10.1016/bs.agph.2021.06.003>
- [67] A. S. M. Uzsoy, S. Thorp, M. Grayling, and K. S. Mandel, “Variational inference for acceleration of sn ia photometric distance estimation with bayesn,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.06013>
- [68] M. Petit-Talamon, M. Lambert, and A. Korba, “Variational inference with mixtures of isotropic gaussians,” 2025. [Online]. Available: <https://arxiv.org/abs/2506.13613>
- [69] J. M. Hernández-Lobato, Y. Li, M. Rowland, D. Hernández-Lobato, T. Bui, and R. E. Turner, “Black-box alpha-divergence minimization,” 2016. [Online]. Available: <https://arxiv.org/abs/1511.03243>
- [70] Y. Li and R. E. Turner, “Rényi divergence variational inference,” 2016. [Online]. Available: <https://arxiv.org/abs/1602.02311>

- [71] R. Bamler, C. Zhang, M. Oppen, and S. Mandt, “Perturbative black box variational inference,” 2018. [Online]. Available: <https://arxiv.org/abs/1709.07433>
- [72] Y. Liu, P. Ramachandran, Q. Liu, and J. Peng, “Stein variational policy gradient,” 2017. [Online]. Available: <https://arxiv.org/abs/1704.02399>
- [73] J. Knoblauch, J. Jewson, and T. Damoulas, “Generalized variational inference: Three arguments for deriving new posteriors,” 2019. [Online]. Available: <https://arxiv.org/abs/1904.02063>
- [74] A. Uppal, K. Stensbo-Smidt, W. Boomsma, and J. Frellsen, “Implicit variational inference for high-dimensional posteriors,” in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: <https://openreview.net/forum?id=Sxu7xlUJGx>
- [75] N. D. Cao, I. Titov, and W. Aziz, “Block neural autoregressive flow,” 2019. [Online]. Available: <https://arxiv.org/abs/1904.04676>
- [76] M. Taboga, “Normal distribution—maximum likelihood estimation,” *Lectures on Probability Theory and Mathematical Statistics, 3rd ed.*; Kindle Direct Publishing, 2017.
- [77] C.-W. Huang, S. Tan, A. Lacoste, and A. Courville, “Improving explorability in variational inference with annealed variational objectives,” 2018. [Online]. Available: <https://arxiv.org/abs/1809.01818>
- [78] C. C. Margossian and L. K. Saul, “The shrinkage-delinkage trade-off: An analysis of factorized gaussian approximations for variational inference,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.09163>
- [79] A. Bulinski and D. Dimitrov, “Statistical estimation of the kullback-leibler divergence,” 2019. [Online]. Available: <https://arxiv.org/abs/1907.00196>
- [80] J. Soch, “StatProofBook/StatProofBook.github.io: StatProofBook 2024,” <https://zenodo.org/record/14646799>, Jan. 2025, zenodo, Jan. 14, 2025.
- [81] F. Perez-Cruz, “Kullback-leibler divergence estimation of continuous distributions,” 08 2008, pp. 1666 – 1670.
- [82] Y. Zhang, W. Liu, Z. Chen, J. Wang, and K. Li, “On the properties of kullback-leibler divergence between multivariate gaussian distributions,” 2023. [Online]. Available: <https://arxiv.org/abs/2102.05485>
- [83] Q. Wang, S. R. Kulkarni, and S. Verdú, “Divergence estimation for multidimensional densities via k -nearest-neighbor distances,” *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2392–2405, 2009.
- [84] M. Noshad, K. R. Moon, S. Y. Sekeh, and A. O. Hero, “Direct estimation of information divergence using nearest neighbor ratios,” in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 903–907.
- [85] T. B. Berrett, “Modern k -nearest neighbour methods in entropy estimation, independence testing and classification,” 2017.