Utrecht University

Institute for Gravitational and Subatomic Physics

Applied Data Science Master's Thesis

l

Candidate: name
First Examiner: name
Second Examiner: name

Date: July 2025

Contents

1 Abstract

2 Introduction

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.
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.

The above can be represented mathematically as Formula 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]

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

where

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

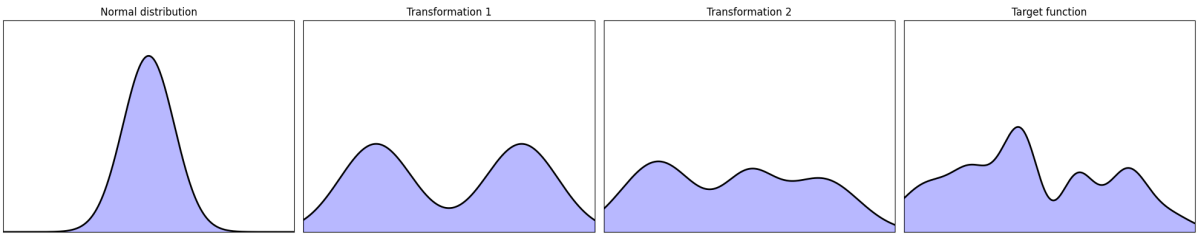
and

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

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.
- 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: the introduction of variational approximation as a technique for inference; methodological enhancements aimed at improving VI performance; and 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 $\{x_1, x_2, \ldots, x_n\}$, 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 $p(y \mid x)$ [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 $p(y \mid x)$, 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:

$$p(x) = \sum_z p(x \mid z) * p(z) \tag{2}$$

if a discrete distribution is given, or

$$p(x) = \int p(x \mid z) * p(z) dz \tag{3}$$

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). $p(x \mid z)$ is the conditional likelihood, also chosen to be a simple distribution, but with complex parameters that may vary with z. Despite both $p(x \mid z)$ 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 let us represent 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 $p(x \mid z)$. [26][27][28] Figure 2 shows how a Gaussian mixture model was used to model the density of a more complicated distribution.

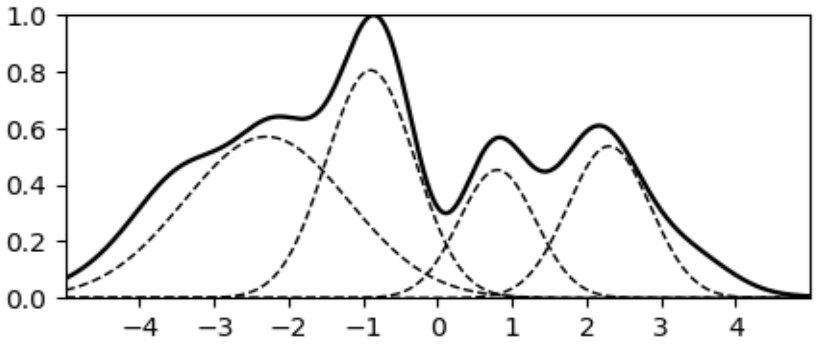


Figure 2: Latent variable model approximation of a complex distribution

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 errors indicate dependency. [29][30]

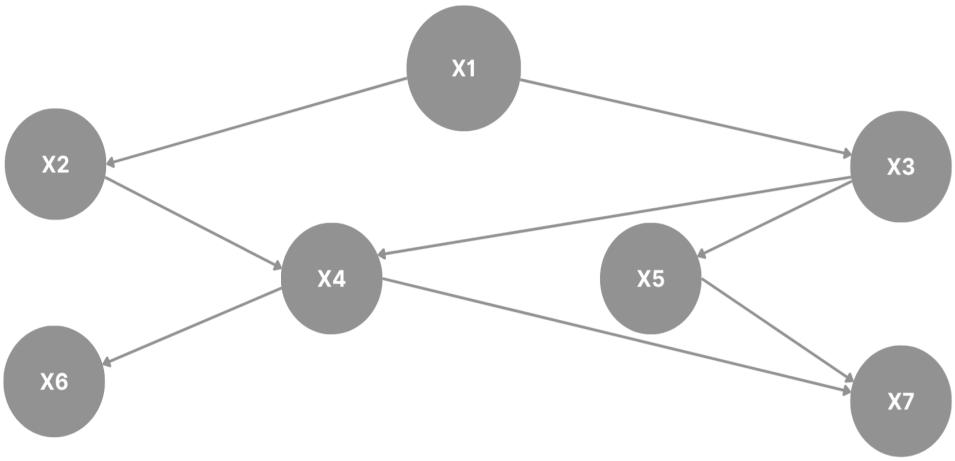


Figure 3: Graphical model representation

If joint probability density is relatively easy to compute:

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

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

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

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:

$$p(z \mid x) = \frac{p(x,z)}{p(x)} \tag{6}$$

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 Kullback-Leibler (KL) which is the measure of the difference between two probability distributions and discussed in Appendix A, be presented as:

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

then,

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

so that

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

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

$$L = \sum q(z) * \log \frac{p(x,z)}{q(z)} \tag{9}$$

so that

$$KL + L = lnp(x) \tag{10}$$

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 $p(z \mid x)$ typically lies outside Q. The idea is to find a distribution (target point) within $Q(z)$ that is as close as possible to $p(z \mid x)$ using optimization techniques to minimize the distance between them. [31]

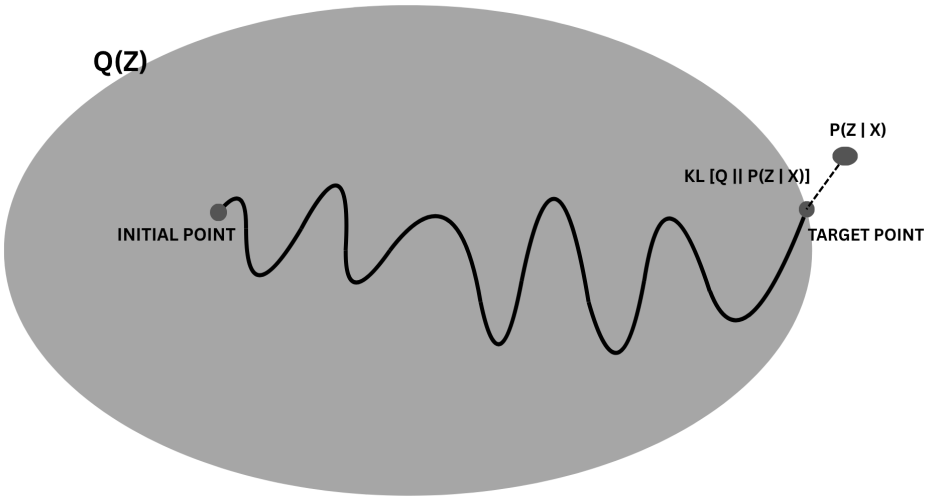


Figure 4: Basics of Variational Inference[31]

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 coreset 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]

4.3 Presenting literature review related to the comparison between VI and MCMC

5 Toy problem

5.1 Problem Statement

5.2 Proposed solution

5.3 Research question

5.4 Research setup

6 Results

7 Conclusion

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 [42][43]

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

Then Kullback-Leibler Divergence of $p(x)$ with respect to $q(x)$ is given as: [43]

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

It is important to recognize that, for multivariate models, where p and q are multivariate normal distributions, modified Kullback-Leibler Divergence takes the form: [43][44][45]

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

Then,

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

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

if

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

then

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

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

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

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. [46][47][48]

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:

$$p(x, z) = p\left(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5\right) \tag{1}$$

so that

$$p(z \mid x) = \frac{p(x, z)}{p(x)} = \frac{p\left(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5\right)}{\iiiint \iint p\left(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4, z_5\right) dz_1 dz_2 dz_3 dz_4 dz_5}. \tag{2}$$

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 $p(z \mid x)$, so that $KL+L = lnp(x)$, and the lower bound L can be shown as:

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

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:

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

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

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

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

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, "¡tt¿i- flow¡/tt¿: 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. Gabrié, 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. Hübner, K. S. Phukon, M. Pitkin, M. Rizzo, N. Sarin, R. Smith, S. Stevenson, A. Vajpeyi, M. Arène, K. Athar, S. Banagiri, N. Bose, M. Carney, K. Chatziioannou, J. A. Clark, M. Colleoni, R. Cotesta, B. Edelman, H. Estellés, C. García-Quirós, A. Ghosh, R. Green, C.-J. Haster, S. Husa, D. Keitel, A. X. Kim, F. Hernandez-Vivanco, I. Magaña 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. Pürrer, A. Ramos-Buades, V. Raymond, E. Thrane, J. Veitch, D. Williams, M. J. Williams, and L. Xiao, "Bayesian inference for compact binary coalescences with ¡scp¿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. Hübner, 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. Gabrié, 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 Asso-*

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] A. Bulinski and D. Dimitrov, "Statistical estimation of the kullback-leibler divergence," 2019. [Online]. Available: https://arxiv.org/abs/1907.00196

[43] J. Soch, "StatProofBook/StatProofBook.github.io: StatProofBook 2024," https://zenodo.org/record/14646799, Jan. 2025, zenodo, Jan. 14, 2025.

[44] F. Perez-Cruz, "Kullback-leibler divergence estimation of continuous distributions," 08 2008, pp. 1666 – 1670.

[45] 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

[46] 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.

[47] 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.

[48] T. B. Berrett, "Modern k-nearest neighbour methods in entropy estimation, independence testing and classification," 2017.