Accelerating gravitational-wave inference with machine learning

Michael J. Williams

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Physics & Astronomy
College of Science and Engineering
University of Glasgow

October 2023

Abstract

The future for gravitational-wave astronomy is bright, with improvements for existing ground-based interferometers of the LIGO-Virgo-KAGRA Collaboration (LVK) and new ground- and space-based interferometers planned for the near future. As a result, there will imminently be an abundance of data to analyse from these detectors, which will bring with it the chances to probe new regimes. However, this will also bring with it new challenges to address, such as the volume of data and need for new analysis techniques.

Leveraging this data hinges on our ability to determine the characteristics of the sources that produce the observed gravitational-wave signals, and Bayesian inference is the method of choice. The main algorithms that have been used in these analyses are Markov Chain Monte Carlo and Nested Sampling. Each have their own advantages and disadvantages. However, both are computationally expensive when applied to gravitational-wave inference, typically taking of order days to weeks for shorter signals and up to months for longer signals, such as those from binary neutron star mergers. Furthermore, the cost of these analyses increases as additional physics is included, such as higher-order modes, precession and eccentricity. These factors, combined with the previously mentioned increase in data, and therefore number of signals, pose a significant challenge. As such, there is a need for faster and more efficient algorithms for gravitational-wave inference. In this work, we present novel algorithms that serve as drop-in replacements for existing approaches but can accelerate inference by an order of magnitude.

Our initial approach is to incorporate machine learning into an existing algorithm, namely nested sampling, with the aim of accelerating it whilst leaving the underlying algorithm unchanged. To this end, we introduce `nessai`, a nested sampling algorithm that includes a novel method for sampling from the likelihood-constrained prior that leverages normalizing flows, a type of machine learning algorithm. Normalizing flows can approximate the distribution of live points during a nested sampling run, and allow for new points to be drawn from it. They are also flexible and can learn complex correlations, thus eliminating the need to use a random walk to propose new samples.

We validate `nessai` for gravitational-wave inference by analysing a population of simulated binary black holes (BBHs) and demonstrate that it produces statistically consistent results. We also compare `nessai` to `dynesty`, the standard nested sampling algorithm used by the LVK, and find that, after some improvements, it is on average ~ 6 times more efficient and enables inference in time scales of order 10 hours on a single core. We also highlight other advantages of `nessai`, such as the included diagnostics and simple parallelization of the likelihood evaluation. However, we also find that the rejection sampling step necessary to ensure new samples are distributed according to the prior can be a significant computational bottleneck.

We then take the opposite approach and design a custom nested sampling algorithm tailored to normalizing flows, which we call `i-nessai`. This algorithm is based on importance nested sampling and incorporates elements from existing variants of nested sampling. In contrast to the standard algorithm, samples no longer have to be ordered by increasing likelihood nor distributed according to the prior, thus addressing the aforementioned bottleneck in `nessai`. Furthermore, the formulation of the evidence allows for it to be updated with batches of samples rather than one-by-one. The algorithm we design is centred around constructing a meta-proposal that approximates the posterior distribution, which is achieved by iteratively adding normalizing flows until a stopping criterion is met.

We validate `i-nessai` on a range of toy test problems which allows us to verify the algorithm is consistent with both `nessai` and, when available, the analytic results. We then repeat a similar analysis to that performed previously, and analyse a population of simulated BBH signals with `i-nessai`. The results show that `i-nessai` produces consistent results, but is up to 3 times more efficient than `nessai` and more than an order of magnitude more efficient (13 times) than `dynesty`. We also apply `i-nessai` to a binary neutron star (BNS) analysis and find that it can yield results in less than 30 minutes whilst only requiring $\mathcal{O}(10^6)$ likelihood evaluations.

Having developed tools to accelerate parameter estimation, we then apply them to real data from LVK observing runs. We choose to analyse all 11 events from O1 and small selection of events from O2 and O3 and find good agreement between our results and those published by the LVK This demonstrates that `nessai` can be used to analyse real gravitational-wave data. However, it also highlights aspects that could be improved to further accelerate the algorithm, such as how the orbital phase and multimodal likelihood surfaces are handled. We also show how `i-nessai` can be applied to real data, but ultimately conclude that further work is required to determine if the settings used are robust. Finally, we consider `nessai` in the context of next generation ground-based interferometers and highlight some of the challenges such analyses present.

As a whole, the algorithms introduced in this work pave the way for faster gravitational-wave inference, offering speed-ups of up to an order of magnitude compared to existing approaches. Furthermore, they demonstrate how machine learning can be incorporated into existing analyses to accelerate them, which has the additional benefit of providing drop-in replacements for existing tools.

Contents

List of Figures

List of Tables

Preface

In this thesis, I use Oxford English spelling, which follows the convention used in the Oxford English Dictionary. Most notably, this uses the suffix *-ize* for some words since “it reflects both the origin and the pronunciation more closely”[1] whilst recognizing *-ise* as an alternative spelling.

[1] Oxford English Dictionary. *Frequently asked questions: Why does the OED spell verbs such as organize and recognize in this way?* `https://www.oed.com/page/faqs/Frequently+asked+questions#spell`. Accessed: 2023/06/28.

Acknowledgements

> They share a lot, astronomy and childhood. Both are voyages across huge distances. Both search for facts beyond their grasp. Both theorize wildly and let possibilities multiply without limits. Both are humbled every few weeks. Both operate out of ignorance. Both are mystified by time. Both are forever starting out.
>
> Richard Powers, *Bewilderment*

I am grateful to both my supervisors, John and Chris, for the support and guidance they provided me throughout my PhD and for instilling in me the fundamentals of being 'Bayesian'. To John, for the many hours spent contemplating the fundamentals of nested sampling whilst staring at a whiteboard, in person or virtually, and for the hacking sessions that lead to many an idea and fixed many a bug. To Chris, for introducing me to machine learning, sparking my passion for it and for the countless discussions on different machine learning ideas that lead to many opportunities.

I am a grateful to the members of the IGR for the environment they promote that is welcoming and friendly but also intellectually stimulating and enjoyable. For the many coffee breaks and evenings in the pub spent discussing anything and everything one could imagine. And to my office-mates for putting up with me, for input on plots, figures and code, and for creating a space I looked forwarding to coming to every day.

I am grateful to the members of the PHAS Machine Learning Journal Club and IGR Machine learning Meeting, past and present, for the opportunity to discuss such a wide range of machine learning papers, applications and problems and for the knowledge I have gained as a result.

I am grateful to my examiners Vivien and Norman, for an enjoyable viva, for the comments that have undoubtedly improved my thesis and for spotting many typos.

I am grateful to my family and friends, for your support, for the chances you gave me to disconnect from my PhD and being there even when I myself was distant.

Last, but by no means least, I am forever grateful to my partner, Julia, for your support, encouragement and patience, for putting up with me when I was stressed, for listening to many a practice presentation and proofreading many a document, and for the many mind-clearing hikes up a Munro, or four.

My research was supported by the Science and Technologies Funding Council (STFC) and made extensive use of the computational resources provided by the Institute for Gravitational Research, LIGO Laboratory and Cardiff University.

Declaration

All the material presented in this thesis was carried out by the author under the supervision of Dr. John Veitch and Dr. Chris Messenger in the Institute for Gravitational Research at the University of Glasgow. It has not been submitted for any other degree at the University of Glasgow, or at any other institution. This work was carried out as a part of the LIGO-Virgo-KAGRA Collaboration (LVK) and has therefore benefited from interactions with other members of the collaboration.

Chapters 1 to 3 introduce the relevant background theory and review the current literature on gravitational waves, Bayesian inference and machine learning. These chapters were written with input from Dr. John Veitch and Dr. Chris Messenger.

Chapter 4 presents work that was written under the supervision of Dr. John Veitch and Dr. Chris Messenger and has been published in *Physical Review D* [1] and is presented almost verbatim. The introductory sections have been omitted and certain symbols and spellings have been updated to be consistent with the other chapters. Some additional results that were originally included in the appendices have been moved to the main body of the text. The remaining appendices are included in appendix D where the explanation on training normalizing flows has been removed since this is already covered in chapter 3.

Chapters 5 and 6 present work that was conducted under the supervision of Dr. John Veitch and Dr. Chris Messenger and were written by the author with their input.

Chapter 7 presents work that was written under the supervision of Dr. John Veitch and Dr. Chris Messenger and has been accepted for publication in *Machine Learning: Science and Technology* [2] and, similarly to chapter 4, is presented almost verbatim. The introduction and background sections have been replaced with a condensed introduction, since the material is covered in chapter 2 and chapter 3. The appendices are reproduced in appendix F where the appendix on training normalizing flows with weights has been omitted, since it is covered in section 3.3.5.

Chapter 8 presents work that was conducted under the supervision of Dr. John Veitch and Dr. Chris Messenger and was written by the author with their input.

Chapter 9 reviews all the work presented in this thesis and was written by the author with input from Dr. John Veitch and Dr. Chris Messenger.

Acronyms

ABC approximate Bayesian computation 43, 54

Adam adaptive moment estimation 64, 89, 222–224

AdVirgo Advanced Virgo 14, 22, 91, 121, 203

aLIGO Advanced LIGO 13, 14, 22, 91, 121

ASD amplitude spectral density ix, 16, 17

BBH binary black hole i, ii, ix–xii, 6, 7, 18, 22, 23, 27, 50, 51, 114, 119, 132, 135, 138, 139, 167, 170, 174, 178, 179, 183–188, 191, 195, 200, 203, 206–208, 210–216, 219, 225, 241, 242, 250

BNS binary neutron star ii, xiv, 6, 7, 18, 22, 23, 50, 51, 53–55, 102, 167, 170, 188, 190–192, 194, 207, 208, 214, 215, 225, 250

CBC compact binary coalescence iii, 1, 6–8, 15–17, 21–24, 47–54, 56, 57, 77–79, 82, 90, 200, 210, 211, 217, 226

CDF cumulative density function 27, 107

CE Cosmic Explorer 15, 167, 203–206, 208, 216

CMB cosmic microwave background 8

CMF cumulative mass function 190, 194, 233, 240

CNN convolutional neural network 78, 79

DECIGO deci-hertz gravitational-wave observatory 15

EOB effective-one-body 18, 19, 23

EOS equation of state 22, *Glossary:* equation of state

ESS effective sample size 31, 144, 151, 155–157, 162, 164, 165, 200, 202

ET Einstein Telescope 15, 167, 203–206, 208, 209, 216

GAN generative adversarial network 43, 66, 76, 78, *Glossary:* generative adversarial network

GRB gamma-ray burst 7, 8, 22, 217, *Glossary:* gamma-ray burst

GWTC-1 Gravitational-Wave Transient Catalog 1 xi, xiii, 89, 90, 167, 170, 173–179, 181, 185, 188, 189, 205–207, 215, 241–244

GWTC-2 Gravitational-Wave Transient Catalog 2 xiii, 54, 89

GWTC-2.1 Gravitational-Wave Transient Catalog 2.1 xii, 55, 167–170, 173, 175, 178, 180–185, 191–196, 205–208, 215, 216, 241–246

GWTC-3 Gravitational-Wave Transient Catalog 3 xii, 55, 168, 191, 198, 206, 207

i.i.d. independently and identically distributed 37, 38, 56, 83, 142, 145, 146, 148, 152, 213

IMR inspiral-merger-ringdown 18

ISCO innermost stable circular orbit 6, 7

JSD Jensen-Shannon divergence x, xi, xiv, 27, 28, 46, 47, 94, 96, 160, 168, 173–175, 178, 179, 181, 200, 202, 215, 239

KAGRA Kamioka Gravitational Wave Detector 14, 22

KLD Kullback-Leibler divergence 27, 28, 37, 44, 46, 47, 63, 72–74, 86, 150, 223, 224, 251

KS test Kolmogorov-Smirnov test 42, 44, 46, 97, 252

LIGO Laser Interferometer Gravitational-wave Observatory 13, 14, 22, 203

LISA Laser Interferometer Space Antenna 15, 217, 219

LSC LIGO Scientific Collaboration 252

LVC LIGO-Virgo Collaboration 2, 170, 252

LVK LIGO-Virgo-KAGRA Collaboration i, ii, xviii, 2, 8, 22, 140, 142, 167, 168, 188, 190, 191, 195, 208, 215–217, 219, 252

MCMC Markov Chain Monte Carlo 25, 31, 32, 36, 38, 39, 41, 51, 54, 79, 89, 90, 142, 143, 210, 217, 253

MLP multi-layer perceptron 65, 68

NS-SMC Nested Sampling via Sequential Monte Carlo 41

NSBH neutron star-black hole 6, 7, 22, 23, 55, 102

P-P plot probability-probability plot ix–xiii, 44–46, 74, 75, 91, 92, 101, 114, 115, 137, 138, 158, 168, 211, 230, 234, 236, 238, 252, *Glossary:* probability-probability plot

PDF probability density function 25, 27, 29, 30, 48, 66, 67, 73, 74, 77, 85, 143, 223, 224, 248, 250, 251, 253

PMF probability mass function 27, 231, 253

PSD power spectral density 16, 17, 23, 48–51, 56, 90, 91, 175, 203, 215, 248, 254

RealNVP real non-volume preserving 69, 76, 153

ReLU Rectified Linear Unit 61, 62, 222, 224, 229

ResNet residual neural network 65, 68, 89, 222, 224, 229

ROM reduced order model 53

ROQ Reduced-Order-Quadrature xiv, 53, 56, 159, 162, 165, 188, 192, 207, 214

SGWB stochastic gravitational-wave background 8, 9, 15, 23

SMC Sequential Monte Carlo 41, 143, 149, 151, 153, 163

SNR signal-to-noise ratio ix, 50, 53, 90, 91, 95, 114, 116, 117, 119, 121, 123, 135, 137, 138, 185, 203, 205, 212

SVD singular value decomposition 53

tanh hyperbolic tangent 61, 62

TT gauge Transverse-Traceless gauge 3–5

VAE variational autoencoder 43, 44, 66, 76, 90, 102, *Glossary:* variational autoencoder

Index of Notation

Number sets

$\mathbb{R}$	Real numbers
$\mathbb{Z}$	Integers

General

$\boldsymbol{a}$	vector with components $\{a_1, a_2, ..., a_d\}$
$\tilde{a}$	frequency-domain representation
a^*	complex conjugate
$\hat{a}$	esimate of a
$\langle a\|b\rangle$	noise-weighted inner product, see eq. (2.49)
$\equiv$	equivalent to

Probability & sampling

H	hypothesis/model
$\boldsymbol{\theta}$	parameters of the model
$\boldsymbol{d}$	observed data
$\mathcal{L}(\boldsymbol{\theta})$	likelihood
$\pi(\boldsymbol{\theta})$	prior
Z	Bayesian evidence
$\mathcal{B}$	Bayes factor
$\mathcal{H}$	information content
X	prior volume
$\sim$	distributed as
$\{\boldsymbol{x}_i\}_{i=1}^N$	set of N samples
$\mathcal{N}(a, b)$	normal distribution with mean a and standard deviation b
$\mathcal{U}[a, b]$	uniform distribution with limits a and b
$\chi(\mu)$	χ-distribution with μ degrees of freedom

Machine learning

$\boldsymbol{\phi}$	trainable parameters
η	learning rate
$\Theta(\cdot)$	conditioner
$\mathcal{Z}$	latent space
$\mathcal{X}$	physical (data) space

Chapter 1

Introduction

This chapter describes gravitational waves and the sources that produce them, how gravitational waves are detected, the models used to describe gravitational-wave data and the current state of gravitational-wave astronomy; focusing predominantly on the setting of current generation ground-based interferometers. This serves to provide context and motivation for the work presented in later chapters.

Section 1.1 reviews how gravitational-waves are predicted by Einstein's general relativity and their nature. Section 1.2 introduces categories of gravitational-wave sources that are relevant for current ground-based detectors. Section 1.3 then describes gravitational-wave detectors, focusing on ground-based interferometers, their response to gravitational-waves and the existing network of gravitational-wave detectors. Section 1.4 discusses the data from gravitational-wave detectors is modelled, including specifically how the signals from compact binary coalescence (CBC) are modelled and the physical parameters the describe them. Finally, the state of gravitational-wave astronomy and the motivation for the work presented in this thesis are reviewed in section 1.5.

Gravitational waves were theorized long before they were detected. In 1916, Einstein, in his theory of General Relativity [3], showed that massive accelerating bodies should produce wave-like perturbations in spacetime. These would propagate at the speed of light and could provide information about the sources that produce them and the nature of gravity. The first evidence for the existence of gravitational waves came in 1974 when Russell Hulse and Joseph Taylor discovered a binary pulsar system [4]. Driven by Einstein's prediction, they tracked the time of arrival of the emission from the system and found that the decay in the orbital separation was consistent with the loss of energy via the emission of gravitational waves predicted by general relativity. This provided indirect evidence for gravitational waves, and they later received the Nobel Prize for this discovery [5].

Almost a hundred years after Einstein first formulated General Relativity, the first direct detection of gravitational waves was made. On September 15th 2015 the two laser interferometers of the LIGO-Virgo Collaboration (LVC) detected the perturbations caused by the gravitational waves emitted by the collision of two black holes 1.3 billion light years away [6]. This detection marked the start of gravitational-wave astronomy and set the stage for a new means of understanding the Universe. Since this first detection, the LIGO-Virgo-KAGRA Collaboration (LVK) has continued to detect gravitational waves and the total number of confidently detected events now stands at 90 [7–10].

We now introduce gravitational waves, discuss how they interact with matter and the classes of sources that emit them. From here, we then provide an overview of gravitational-wave detectors, focused on ground-based interferometers and their response to gravitational waves. We then discuss how to model data from gravitational-wave detectors, including both the noise and signal components. Finally, the state of gravitational-wave astronomy is summarized, including some of the key detections and prospects for the future.

1.1 Gravitational waves

If we assume a nearly flat spacetime, the metric tensor $g_{\mu\nu}$ can be written as

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \tag{1.1}$$

where $\eta_{\mu\nu}$ is the metric in flat Minkowski spacetime with signature $(-,+,+,+,)$ and $h_{\mu\nu}$ is a small perturbation $|h_{\mu\nu}| \ll 0$. Our aim is to define the linearized Einstein field equations and consider the solutions. If instead of the metric perturbation $h_{\mu\nu}$, we consider the *trace-reversed metric perturbation*

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h, \tag{1.2}$$

where $h = \eta^{\mu\nu}h_{\mu\nu}$ is the trace, and choose a specific coordinate system, then the field equations can be simplified [11, 12]. If we choose the *Lorenz gauge*, which requires that the divergence of the trace-reversed metric perturbation be zero, that is

$$\frac{\partial \bar{h}^{\mu\alpha}}{\partial x^{\mu}} = 0, \tag{1.3}$$

it can be shown that the linearized field equations simplify to

$$\Box \bar{h}_{\mu\nu} \equiv \left(-\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2 \right) \bar{h}_{\mu\nu} = -\frac{16\pi G}{c^2} T_{\mu\nu}, \tag{1.4}$$

where $\Box$ is the d'Alembert operator, c is the speed of light in a vacuum, ∇^2 is the Laplace operator, G is the gravitational constant, and $T_{\mu\nu}$ is the stress-energy tensor [11, 12].

However, we are more interested in the propagation of gravitational waves in a vacuum, i.e. where $T_{\mu\nu} = 0$ and eq. (1.4) simplifies to

$$\left(-\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2 \right) \bar{h}_{\mu\nu} = 0. \tag{1.5}$$

The simplest solution to this equation is a plane wave of the form

$$\bar{h}_{\mu\nu} = \mathrm{Re}\left[A_{\mu\nu} \exp\left(i k_{\alpha} x^{\alpha} \right) \right], \tag{1.6}$$

where $A_{\mu\nu}$ is the polarization tensor, and k_{α} is the wave vector, all of which are constant [11–13]. This describes a wave that travels at the speed of light.

We now consider the degrees of freedom – $h_{\mu\nu}$ is symmetric, so it has 10 components and the Lorenz gauge places four constraints on $h_{\mu\nu}$, thus reducing the degrees-of-freedom to six. However, we can reduce the degrees-of-freedom further by introducing an infinitesimal coordinate transform: $x^{\mu} \rightarrow x^{\mu} + \xi^{\mu}$ with $\Box \xi_{\mu} = 0$ [11–13]. This reduces the degrees of freedom to two and implies that $\bar{h}_{\mu\nu}$ is traceless, so $\bar{h}_{\mu\nu} = h_{\mu\nu}$, and is also transverse ($\bar{h}_{0\nu} = 0$). This is referred to as the Transverse-Traceless gauge (TT gauge) and the metric in this gauge is written as h_{ij}^{TT}, where ij is used to refer to only the spatial components since $\bar{h}_{0\nu} = 0$, i.e. it is constant in time.

If we consider eq. (1.6) in the TT gauge and define the direction of propagation to be the $z = x^{(3)}$-direction, then the wave vector will be $k_{\alpha} = (-\omega, 0, 0, \omega)$ and the only non-zero elements of the polarization tensor will be $A_{11} = -A_{22} \equiv a$ and $A_{12} = A_{21} \equiv b$. The wave can be written in the TT gauge as

$$h_{ij}^{\mathrm{TT}} = \begin{pmatrix} a & b & 0 \\ b & -a & 0 \\ 0 & 0 & 0 \end{pmatrix} \exp\left[\frac{-i\omega}{c}(t - z) \right], \tag{1.7}$$

where a and b are two amplitude terms. If we introduce two separate polarization tensors

$$\mathbf{e}_+ = \left\{e_{ij}^+\right\} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{1.8}$$

and

$$\mathbf{e}_\times = \left\{e_{ij}^\times\right\} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{1.9}$$

which are known as the *plus* and *cross polarizations* and are related by a $\pi/2$ rotation about the z-axis, we can then define

$$h_+ = a \exp\left[\frac{-i\omega}{c}(t-z)\right], \tag{1.10}$$

$$h_\times = b \exp\left[\frac{-i\omega}{c}(t-z)\right], \tag{1.11}$$

which allows us to rewrite eq. (1.7) as

$$h_{ij}^{\mathrm{TT}} = h_+ e_{ij}^+ + h_\times e_{ij}^\times. \tag{1.12}$$

The relevance of this formulation will become apparent when we consider how the metric perturbation interacts with test masses. We can equally introduce other polarizations, such as circular or elliptical polarizations, that are better suited to different applications [14].

We now consider the physical effect of a gravitational wave in the TT gauge on a set of particles. The gravitational wave will stretch the coordinate frame itself, so we cannot directly measure the change in the positions of the particles. Instead, to observe the effect of a gravitational wave, we should consider the proper distance, $ds^2 = g_{\mu\nu}dx^\mu dx^\nu$, which in the TT gauge we can write as

$$ds^2 = -c^2dt^2 + (\delta_{ij} + h_{ij}^{\mathrm{TT}})dx^i dx^j \tag{1.13}$$

where δ_{ij} is the Kronecker delta. We can visualize the effect of a gravitational wave propagating in the z-direction by considering the change in proper distance between a series of particles (or test masses) arranged in a ring in the transverse plane (x–y plane). Figure 1.1 shows how each of the polarizations of a passing gravitational wave will deform the ring of particles; the plus polarization will stretch and squeeze the ring along the x and y axes whilst the cross polarization will stretch and squeeze along the bisection of the x and y axes. Therefore, if we wish to detect gravitational waves, we must measure this change in the proper distance. We can do this by leveraging the relationship between the

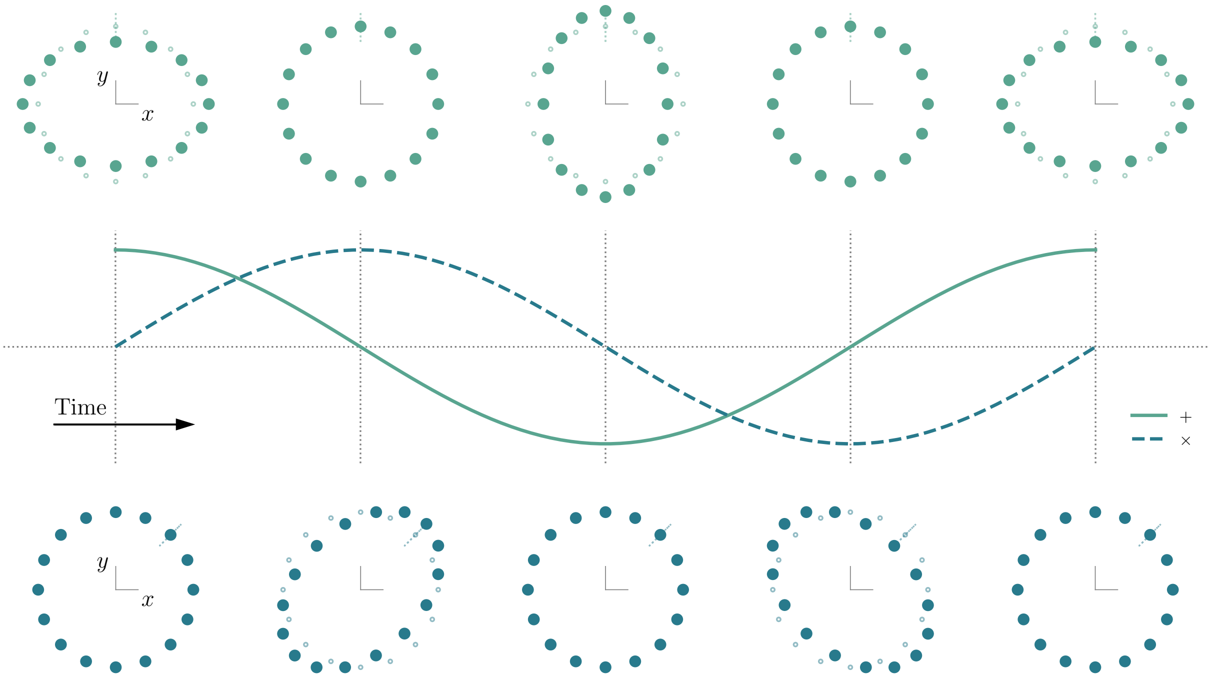


Figure 1.1: Effect of the plus and cross polarizations on a ring of test masses as function of time (left to right) for one complete wavelength. The amplitude of the individual plus and cross polarizations is also shown and highlights the $\pi/2$ phase difference between the two polarizations. Figure based on fig. 2 in Isi [14] and produced using the accompanying code [15].

proper distance and the light travel time, as we will see in section 1.3.

We can now discuss how gravitational waves are produced and their typical amplitudes. This requires us to consider solutions to the Einstein field equations in non-vacuum spacetime, which are expressed as an integral over the stress-energy tensor $T_{\mu\nu}$ [11]. Let us first discuss the nature of gravitational radiation by drawing parallels to electromagnetic radiation. For electromagnetic radiation, due to conservation of charge the monopole moment must be zero but the dipole moment can be non-zero, so dipole radiation is possible. However, for gravitational radiation, conservation of mass and conservation of momentum imply that the monopole and dipole moments must be zero. There is no conservation law for the quadrupole moment, so we can expect quadrupolar (and higher) gravitational radiation [11]. If we assume the weak-field slow-motion ($v \ll c$) limit, then the solutions to the Einstein field equations in non-vacuum spacetime reduce to the quadrupole formula

$$\bar{h}^{ij} = \frac{2G}{c^4}\frac{1}{d_\mathrm{L}}\ddot{Q}^{ij} \tag{1.14}$$

where Q^{ij} is the quadrupole moment and d_L is the luminosity distance to the source. This can be then projected into the TT gauge via a projection operator [11, 12].

Equation (1.14) gives important insight into the nature of gravitational radiation; most importantly, it is at the leading-order quadrupolar. This informs us about what type of motion will emit gravitational radiation. Spherical and axisymmetric motion will not produce radiation since the quadrupole moment will be zero, instead the motion must be non-axisymmetric. We can also use eq. (1.14) to estimate the amplitude of gravitational radiation. If we assume that $\ddot{Q} \sim E_k$, where E_k is the non-axisymmetric kinetic energy, then for a system with $E_k = 1000\,\mathrm{J}$ the gravitational-wave amplitude at $1\,\mathrm{Mpc}$ will be $\sim 1 \times 10^{-64}$. This implies that to detect gravitational waves, we will need to detect very small perturbations or observe systems that are highly energetic, or both. This hints at potential sources of gravitational waves, which are now discussed.

1.2 Gravitational-wave sources

There is a plethora of potential gravitational-wave sources [16] that range from known, well-modelled sources to potentially unknown, exotic sources. It is therefore informative to classify gravitational waves according to the nature of the signal and the type of source that we expected to produce said signal. Each of the categories can in turn have various sub-categories. We now introduce four broad categories of gravitational waves, these are informed by the frequencies ranges current ground-based interferometers are sensitive to (see section 1.3) and in other contexts, e.g. space-based interferometers [17, 18], one may consider a different classification.

1.2.1 Gravitational waves from coalescing binaries

Compact binary coalescences (CBCs) are systems of two compact objects, such as black holes or neutron stars, that are in orbit about a common centre of mass. As they orbit, the system loses energy via gravitational-wave emission and the orbital separation between them decreases until the two objects plunge together and collide, leaving behind a remnant. The nature of this remnant will depend on the compact objects in the original binary [19–22]. For ground-based interferometers, we consider three possibles classes of CBC: binary black holes (BBHs), binary neutron stars (BNSs) and neutron star-black holes (NSBHs).

The signals produced by such binaries are chirp-like; the frequency initially changes very slowly but increases over time as the orbital separation decreases. An example signal is shown in fig. 1.2. These signals can be divided into three phases: inspiral, merger and ringdown. The inspiral is the initial phase where the binaries are in orbit and the strength of the gravitational-wave emission increases as they get closer until they reach the ISCO [28], which for a non-spinning (Schwarzchild) black hole is at $r = 6GM/c^2$ [28]

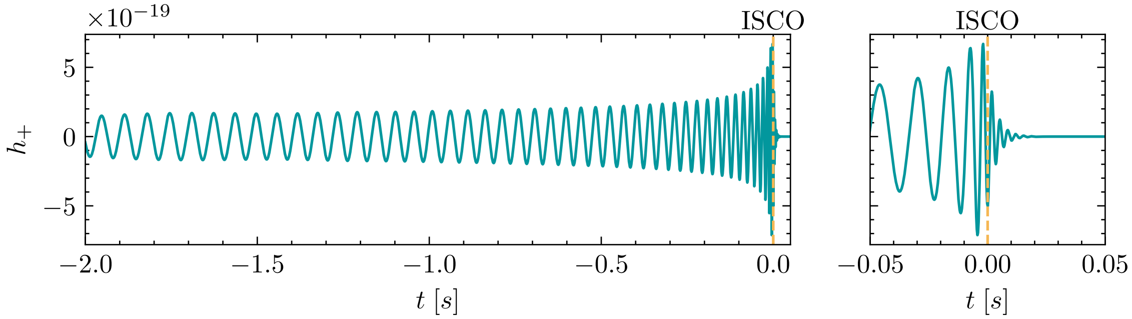


Figure 1.2: Example of the signal from a non-spinning BBH merger with source frame component masses $m_1 = 32$ M$_\odot$ and $m_2 = 29$ M$_\odot$ at a luminosity distance $d_\mathrm{L} = 1$ Mpc from the observer. The plus polarization is shown for a waveform produced using `IMRPhenomD` [23, 24] which includes the initial inspiral, merger at $t = 0$ and final ringdown. The vertical dashed line indicates the time corresponding to the innermost stable circular orbit (ISCO) with frequency $f_\mathrm{ISCO} \approx 73$ Hz. Figure produced using `PyCBC` [25] and `LALSuite` [26, 27].

and corresponds to a frequency of

$$f_\mathrm{ISCO} = \frac{c^3}{6^{3/2}\pi MG}, \tag{1.15}$$

where M is the total mass of the system [28]. The merger is the phase in which the objects collide, and the ringdown is the final stage where the remnant "rings down" into a final stationary equilibrium state. This evolution is shown in fig. 1.2.

To date, these are the only source of gravitational waves that have been detected. The first detection of gravitational waves was of a BBH in 2015 [6] and there have since been subsequent detections of BBH systems and BNS and NSBH systems [29, 30]. These detections are discussed further in section 1.5.

1.2.2 Burst gravitational waves

Burst gravitational waves are a broader category of transient gravitational waves that may be produced by a range of sources. Whilst this could include signals from CBCs, the notion of a gravitational-wave burst generally refers to signals without well-defined waveforms or that are harder to model [31–33]. Bursts can be classed as *short*–duration $\lesssim 1$ s or *long*–duration $\gtrsim 1$ s [16, 34] and different methods are used to detect each.

Potential sources of short-duration bursts include, but are not limited to, gamma-ray bursts (GRBs), core-collapse supernovae; pulsar glitches; and cosmic strings [35–37]. GRBs are of particular interest, since BNS mergers are a known progenitor of short GRBs [38] and NSBH mergers likely also produce them [39]. Long-duration burst are expected to originate from a similarly broad range of sources, including accretion onto black holes and long GRBs [34]. Searches for such signals are typically unmodelled searches that search for coincident excess power in multiple detectors [40]. They can be targeted, such as for GRBs [35, 41], or all-sky searches [32, 42]. More recently, targeted modelled searches have also been applied to search for cosmic strings [37]. To date, the confident candidates that have been detected by burst searches have all been identified as signals from CBCs [7–10, 33, 43].

1.2.3 Continuous gravitational waves

Continuous gravitational waves are long-duration quasi-monochromatic signals. There are different possible sources of such emissions; the conventional source is rapidly rotating neutron stars that have a non-axisymmetric mass distribution [44], however more exotic sources have also been proposed [45].

The signals from rapidly rotating neutron stars are modelled as sinusoids with a frequency decay due to the energy lost to gravitational-wave emission. Detecting these gravitational waves requires integrating the signal over long durations (months to years) in order to distinguish them from the noise. Searches for such signals can be categorized into three distinct categories: *targeted* searches that search for signals from neutron stars with known locations and frequency evolutions, [46, 47], *directed* searches that search for signals from neutron stars with known locations but unknown frequencies [48–50], and all-sky searches that search for unknown neutron stars [51–54]. For example, one could perform a targeted search for the continuous gravitational waves emitted by the pulsar in the Hulse-Taylor system, which has a known location and frequency, or a directed search for emission from the neutron star companion, which does not have a known frequency. As of writing, continuous gravitational waves have yet to be detected, but upper limits have been placed on the maximum possible gravitational-wave strain amplitude [55]. For a recent review of continuous gravitational waves, see Riles [44].

1.2.4 Stochastic gravitational-wave background

The stochastic gravitational-wave background (SGWB) is analogous to the cosmic microwave background (CMB) [56] and is produced by the superposition of unresolved sources of gravitational radiation. Like the CMB, it should come from all directions and can be assumed to be isotropic [57], though ultimately may be anisotropic at small levels [58]. There are different possible sources for the SGWB: cosmological sources, such as inflation, phase transitions and cosmic strings, or an astrophysical background [58]. The latter would be produced by numerous unresolved sources, including transient events, such as supernovae and compact binary coalescence, that have occurred since the start of the Universe and sources that continually emit gravitational radiation, such as rotating neutron stars [59]. The detections made by the LVK [7–10] suggest that the astrophysical background will be dominant over the cosmological background in LVK detectors. The signal from the SGWB will be a stochastic process and, in a single detector, will be indistinguishable from the noise. However, it will be coherent in two detectors (identical if they are co-located). Therefore, if we have two or more detectors with uncorrelated noise, we can cross-correlate the noise for each detector [60] to detect the SGWB. Current searches for the SGWB target both an isotropic background [61–63] and a directional background [64–66] and currently establish upper limits on the energy density of gravitational waves. For a more complete review of the SGWB see Christensen [58].

1.3 Gravitational-wave detectors

The first efforts to detect gravitational waves started in the 1960s and used resonant mass gravitational-wave detectors [67], which rely on mechanical resonance. These detectors consist of a bar with a resonant frequency close to that expected of gravitational waves (around 1 kHz [16]), hence they are often known as *bar detectors*. The bar in such a detector will stretch and contract as a gravitational-waves passes through and, if the gravitational wave has the right frequency, it will excite the resonant frequency of the bar, resulting in mechanical oscillations that can be measured. These detectors have largely fallen out of favour, and whilst some still operate [68], the focus is now instead on using laser interferometry [69–72].

1.3.1 Laser interferometry

Laser interferometry detects small changes in the lengths of optical paths, which in this context are caused by gravitational waves. There are many different types of interferometers but here we focus on the design currently used in ground-based gravitational wave detectors [70–72] which is based on the *Michelson* interferometer.

In a Michelson interferometer, light from a source hits a beam splitter and is sent down two orthogonal arms that each have a reflective mirror (test mass) at the end. The light from each arm returns, recombines at the beam-splitter, and part of the light travels towards a photodetector and the other part towards the source. The fraction of light that hits the photodetector depends on how the beams interfere when they recombine. If the arms are the same length, then there will be destructive interference and no light will reach the photodetector, but if the arms differ in length, then there will be partial destructive interference and some light will reach the photodetector. If an interferometer is designed such that initially no light reaches the photodetector, then if a gravitational wave interacts with it, it will cause a similar effect to what was seen in fig. 1.1 for a ring of particles, one arm will stretch and the other will squeeze which will lead to light hitting the photodetector. This difference in arm length will cause light to hit the photodetector,

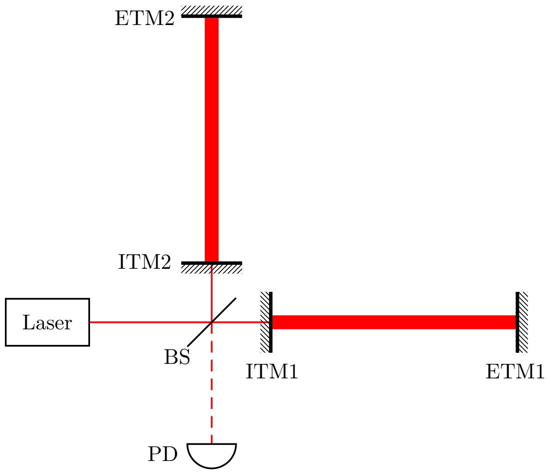


Figure 1.3: Simplified diagram of the Michelson interferometer with Fabry-Perot cavities as used by current generation ground-based interferometers [70–72]. The beamspliter (BS) directs light towards the two cavities comprised of an input test mass (ITM) and end test mass (ETM). A portion of the light exits the cavity and is detected by the photodetector (PD). This diagram is simplified and omits various components, see e.g. Aasi et al. [70] for a more complete schematic.

allowing us to detect the gravitational wave by measuring the relative change in the arm length, known as strain. A schematic of a Michelson interferometer is shown in fig. 1.3.

If the interferometer arms have lengths L_1 and L_2 and a passing gravitational wave changes the length of each arm by ΔL_1 and ΔL_2, then the overall differential change will be $\Delta L = \Delta L_1 - \Delta L_2$. We can then relate the differential change in length, assuming $L_1 = L_2 = L$, to the strain produced by the gravitational wave in the interferometer [12]

$$h = \frac{\Delta L}{L}. \tag{1.16}$$

Ground-based interferometers have arm lengths of order $L \sim 1\,\mathrm{km}$ and the gravitational-wave sources we expected to detect with these interferometers will have amplitudes of order 10^{-23} at frequencies of 10 Hz–1000 Hz [16]. However, a basic Michelson interferometer with these arm lengths would not be sensitive to such small perturbations, so a number of modifications to the basic Michelson design are needed. These include incorporating a Fabry-Perot optical cavity in each arm which causes the light to bounce back and forth in the arms before exiting, increasing the effective length of the arms, and the use of power recycling which increases the laser power and sharpens the interference pattern [73]. This, and various other improvements [70–73], allow current generation detectors to reach strain sensitivities of the order $h \sim 10^{-23}$ which are necessary for detecting gravitational waves [74].

Another important consideration in the design of the interferometers is noise that is present in the detector and will impact the measured strain. There are many different sources of noise that will impact measurements, some of these are intrinsic to the design of the detector, such as quantum noise or thermal noise whilst others are environmental, such as seismic noise or anthropogenic noise [70]. Furthermore, these noise sources will have different amplitudes and occur at different frequencies. Understanding and characterizing the different sources of noise is therefore key to determining the sensitivity of the detectors to gravitational waves. We revisit this in section 1.4 when we discuss modelling the data from gravitational-wave detectors.

1.3.2 Detector response

The response of a gravitational-wave detector to a gravitational wave depends on its geometry and orientation with respect to the source. The response is defined in terms of the response tensor, D^{ij}, which for an interferometric detector in the long-wavelength limit is

$$D^{ij} = \frac{1}{2}(\hat{p}^i\hat{p}^j - \hat{q}^i\hat{q}^j), \tag{1.17}$$

where $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ are unit vectors along each arm of the interferometer [12].

To describe a gravitational wave, we consider two coordinate systems: the first is defined in terms of Earth-fixed coordinates and the second in the reference frame of the source propagating in direction $\hat{\boldsymbol{n}}$. In the Earth reference frame, we define the location of the gravitational-wave source on the sky with two polar angles (θ, ϕ), where θ is the angle relative to the z-axis and ϕ is the rotation in the $\hat{\boldsymbol{p}}$–$\hat{\boldsymbol{q}}$ plane measured about the z-axis. This geometry is shown in fig. 1.4.

In the source frame, we define the basis vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3 = \hat{\boldsymbol{n}}$. We relate these to the Earth-fixed coordinates using the sky location (θ, ϕ) and a third angle, ψ, which describes the counter-clockwise rotation of the $\boldsymbol{e}_1$–$\boldsymbol{e}_2$ plane in the source frame about the line-of-sight $(-\hat{\boldsymbol{n}})$ and is known as the polarization. The two coordinate systems and the three angles are shown in fig. 1.4. Using the angles, the components of the basis vectors, in terms of the Earth-fixed coordinates, are [12]

$$\boldsymbol{e}_1 = \{+\cos\psi\sin\phi - \sin\psi\cos\phi\cos\theta, -\cos\psi\cos\phi - \sin\psi\sin\phi\cos\theta, +\sin\psi\sin\theta\}, \tag{1.18a}$$

$$\boldsymbol{e}_2 = \{+\sin\psi\sin\phi - \cos\psi\cos\phi\cos\theta, -\sin\psi\cos\phi - \cos\psi\sin\phi\cos\theta, +\cos\psi\sin\theta\}, \tag{1.18b}$$

$$\boldsymbol{e}_3 = \{-\sin\theta\cos\phi, -\sin\theta\sin\phi, -\cos\theta\}. \tag{1.18c}$$

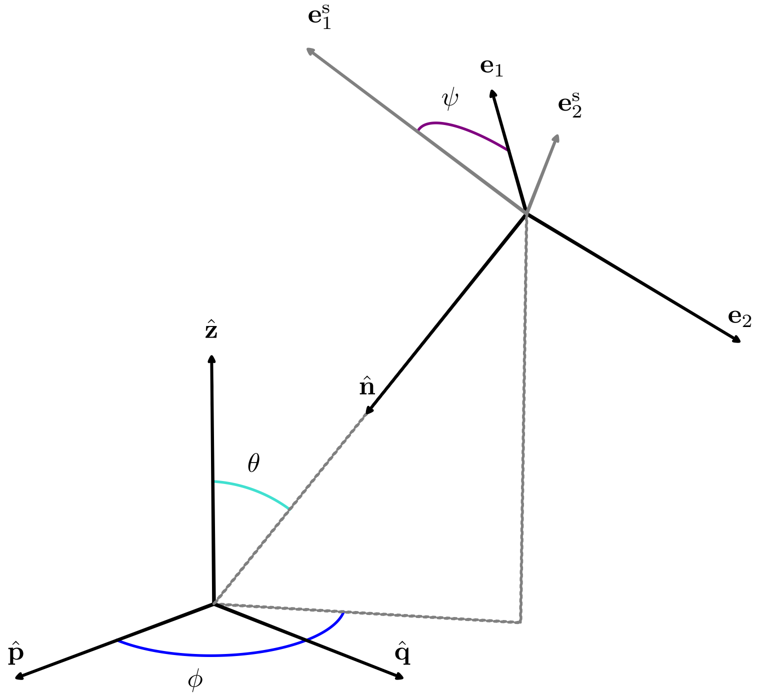


Figure 1.4: Diagram showing the Earth-fixed coordinate system and the source frame coordinate system. The Earth-fixed coordinate system has its origin at the intersection of $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ which denote the orientation of the interferometer arms. The location of the source on the sky is described by the polar angles (θ, ϕ). The source frame is oriented such that $\hat{\boldsymbol{n}}$ denotes the direction of propagation of the gravitational-wave and the polarization ψ describes the rotation of the source frame in $\boldsymbol{e}_1$–$\boldsymbol{e}_2$ plane. $\boldsymbol{e}_1^s$ and $\boldsymbol{e}_2^s$ show the orientation of the source frame before applying the rotation by ψ.

which are related to the plus and cross polarization tensors by [12]

$$\boldsymbol{e}_+ = \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 - \boldsymbol{e}_2 \otimes \boldsymbol{e}_2, \tag{1.19a}$$

$$\boldsymbol{e}_\times = \boldsymbol{e}_1 \otimes \boldsymbol{e}_2 + \boldsymbol{e}_2 \otimes \boldsymbol{e}_1, \tag{1.19b}$$

where $\otimes$ is the tensor product. We can express the strain h produced in an interferometer with detector tensor D_{ij} by the metric perturbation h_{ij}^{TT} in the transverse-traceless gauge as[1]

$$h = D^{ij} h_{ij}^{\mathrm{TT}}. \tag{1.20}$$

Using eq. (1.12), we can re-write this as

$$\begin{aligned} h &= D^{ij} e_{ij}^{+} h_+ + D^{ij} e_{ij}^{\times} h_\times, \\ &= F_+ h_+ + F_\times h_\times, \end{aligned} \tag{1.21}$$

where F_+ and $F_\times$ are the antenna response beam patterns. If we assume an interferometer with orthogonal arms orientated in the Earth-fixed frame such that the x-axis is along one arm and the y-axis is along the other, then the antenna response beam patterns will be [75]

$$F_+(\theta, \phi, \psi) = -\frac{1}{2}(1 + \cos^2\theta) \cos 2\phi \cos 2\psi + \cos\theta \sin 2\phi \sin 2\psi, \tag{1.22a}$$

$$F_\times(\theta, \phi, \psi) = +\frac{1}{2}(1 + \cos^2\theta) \cos 2\phi \sin 2\psi + \cos\theta \sin 2\phi \cos 2\psi. \tag{1.22b}$$

It is also informative to consider the combination, often known as the power, of the antenna beam pattern functions

$$P(\theta, \phi) = F_+^2(\theta, \phi, \psi) + F_\times^2(\theta, \phi, \psi) \tag{1.23}$$

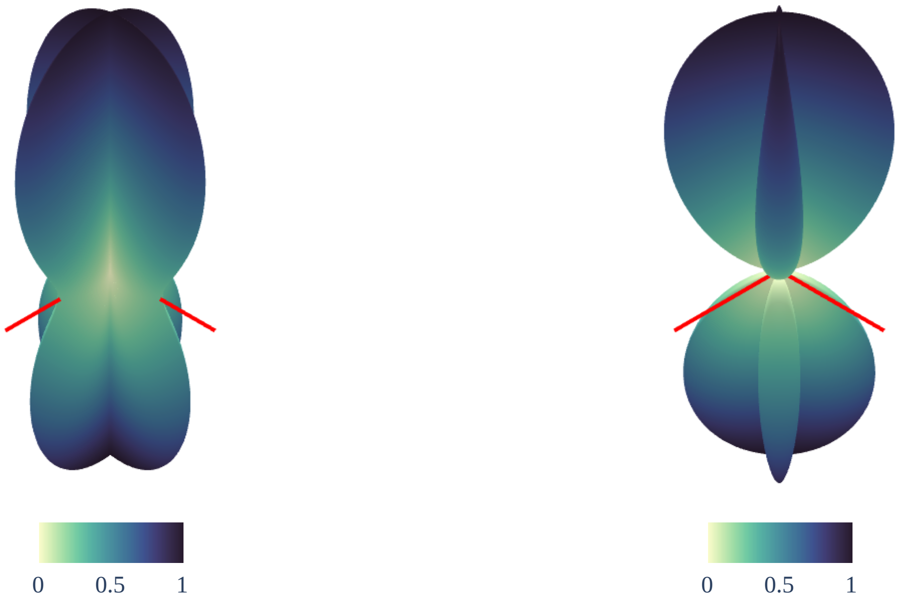
since it is independent of the polarization angle ψ and will be invariant under any change of polarization basis [75]. The antenna response beam patterns for a specific polarization angle are shown in fig. 1.5

1.3.3 Current & future ground-based interferometers

The current network of gravitational-wave interferometers is made up of five ground-based interferometers. These all share the same core design centred around a Michelson interferometer but differ in other key design aspects. The interferometers are

- Advanced LIGO (aLIGO) [70] Hanford and Livingston in the USA, these are second-generation detectors that improve upon the initial Laser Interferometer Gravitational-

[1] The strain h should not be confused with the trace defined in section 1.1 which is also denoted h.



(a) Plus antenna response beam pattern

(b) Cross antenna response beam pattern

Figure 1.5: Plus and cross antenna response beam patterns for a ground-based interferometer as defined in eq. (1.22) with $\psi = 0$. The directions of the arms of the interferometer are shown in red.

wave Observatory (LIGO) design [76]. They are Fabry-Perot Michelson interferometers with 4 km arms and 40 kg test masses and operate at room temperature.

- Advanced Virgo (AdVirgo) [77] in Italy is the second iteration of the Virgo design [78]. It is also a Fabry-Perot Michelson interferometer that operates at room temperature and has 40 kg test masses, but its arm length is slightly shorter at 3 km.
- GEO600 [69] in Germany is a standard Michelson interferometer with 600 m arms and ∼5.6 kg tests masses and is used as a development platform for the LIGO interferometers.
- Kamioka Gravitational Wave Detector (KAGRA) [72] in Japan, is a Fabry-Perot Michelson interferometer that is cryogenically cooled, has 3 km arms and 23 kg test masses.

The different designs result in different sensitivities. During the third observing run, the aLIGO detectors were the most sensitive in the network, followed by AdVirgo, KAGRA and GEO600 [8, 10, 79, 80]. Various improvements are also planned for the detectors, which aim to improve their sensitives for future observing runs, these are detailed in Abbott et al. [74].

In the longer-term, third generation ground-based interferometers are planned. These are designed to be significantly more sensitive than the current generation detectors and include the triangular interferometer Einstein Telescope (ET) [81] and the 40 km Michelson interferometer Cosmic Explorer (CE) [82].

1.3.4 Other gravitational-wave detectors

So far, we have focused solely on ground-based interferometers, however there are other types of existing and planned gravitational-wave detectors that are sensitive to different frequencies and therefore different sources.

Pulsar Timing Arrays Pulsar timing arrays measure the time of arrival of the radio pulses from an array of pulsars, which, due to the stable rotation of pulsars, arrive at regular intervals and can be measured and predicted with high accuracy [83, 84]. They can therefore be used to detected perturbations in Earth's orbit due to gravitational waves or the effects of gravitational waves interacting with the emission from the pulsars. Such arrays are sensitive to low-frequency gravitational waves (10^{-9} Hz to 10^{-6} Hz) [83, 85], making them well-suited to detecting the SGWB [86], though other sources such as massive black hole binaries have also been considered [85, 87].

Space-based Detectors Planned spaced-based detectors will use triangular constellations of spacecraft in either heliocentric or geocentric orbits and perform laser interferometry between the pairs of spacecraft to track the separation between the free-falling test masses on board. There are three proposed space-based interferometers: Laser Interferometer Space Antenna (LISA) [17] which will have a heliocentric orbit and sensitive to gravitational waves between 10^{-4} Hz and 1 Hz [88, 89], TianQin [18, 90] which will have a geocentric orbit and be sensitive to gravitational waves between 10^{-4} Hz and 1 Hz [90] and finally deci-hertz gravitational-wave observatory (DECIGO) [91, 92] which will comprise four constellations in heliocentric orbits and be sensitive in the range of 10^{-1} Hz and 10 Hz [92]. At these frequency ranges, spaced-based detectors will be sensitive to a range of different sources including galactic binaries, massive black hole binariess, extreme mass ratio insprials, primordial black holes and even some stellar mass CBC signals [93].

1.4 Modelling gravitational-wave data

We saw previously that, in general, the input to a gravitational-wave detector is given by eq. (1.20). The output strain from the detector as a function of time $d(t)$ will depend on the input but also on the inherent noise in the detector. We therefore model the strain data as the combination of two components: the signal from a gravitational wave $h(t)$ and the detector noise $n(t)$, which is the combination from the different noise sources in the detector, so we define the data as

$$d(t) = h(t) + n(t), \tag{1.24}$$

or equivalently in the frequency-domain as $\tilde{d}(f) = \tilde{h}(f) + \tilde{n}(f)$. In order to extract information about the gravitational-wave signal $h(t)$, we must be able to distinguish it from the noise $n(t)$. To do so, we must be able to characterize the noise in the detector and, in some cases, model the signal. We now review how we characterize the noise and model the signal in the context of analysing signals from CBCs.

1.4.1 Characterizing the noise

We can model the noise in an interferometer as a random process in the time domain, which we represent as a vector $\boldsymbol{n}$, with components $n_i = n(t_i)$. This allows us to infer statistical properties of the noise and model it as a probability distribution $p(\boldsymbol{n})$. In most cases, we will assume that properties of the noise do not change over time scales being considered, in this case the noise is referred to as *stationary* noise and the correlations between different noise samples will only depend on the time lag between them $\tau = |t_i - t_j|$. We will also assume that the noise is Gaussian, so $p(\boldsymbol{n})$ is described by a multivariate Gaussian distribution which is parameterized by a mean and a covariance matrix. We revisit this in section 2.10, after having introduced probability and Bayesian inference.

Alternatively, we can consider the properties of the noise in the frequency domain. Since we assume the noise is stationary over a duration T and Gaussian, the noise in different frequency bins will be independent, so we can use the power spectral density (PSD), $S_n(f)$, to characterize the noise. It is defined as

$$S_n(f) = \lim_{T\to\infty} \frac{2}{T} \left| \int_{-T/2}^{T/2} n(t) \exp\left(-2\pi i f t\right) \mathrm{d}t \right|^2, \tag{1.25}$$

with units of Hz^{-1} and the square root of it is known as the amplitude spectral density (ASD). For a complete derivation of the PSD, see Creighton and Anderson, Chapter 7, Section 1 [12]. Noise that can be described by a frequency dependent PSD is known as coloured noise, whereas if the PSD is constant in time and frequency, it is known as white noise.

In practice, the PSD is not known a priori, so it must be estimated from the data and different methods are used [94, 95] which are discussed in the context of parameter estimation in section 2.10.3. In fig. 1.6, we show an example of the estimated ASDs for LIGO Hanford, LIGO Livingston and Virgo from the first half of third observing run [8]. This shows how the sensitivity of the detectors varies as function of frequency, and how they are designed to be most sensitive in the 100 Hz–1000 Hz range.

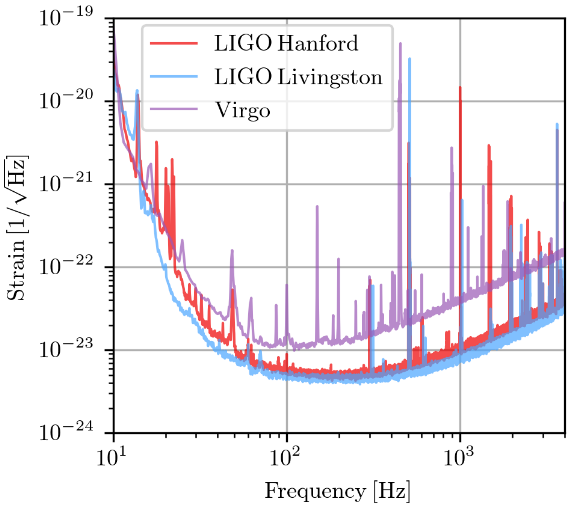


Figure 1.6: Example ASDs for LIGO Hanford, LIGO Livingston and Virgo for O3a. These show the strain in each detector, assuming only noise is present, as a function of frequency in units of $1/\sqrt{\mathrm{Hz}}$. See eq. (1.25) for the definition of the ASD and PSD. Original figure from Abbott et al. [8].

1.4.2 Modelling signals for CBC

We have already seen how the effect of a gravitational-wave signal on an interferometer can be modelled as a function of the cross and plus polarizations in eq. (1.21),

$$h(t) = F_{+}(\theta, \phi, \psi)h_{+}(t) + F_{\times}(\theta, \phi, \psi)h_{\times}(t) \tag{1.26}$$

where we have now included a time dependency. The antenna patterns $\{F_{+}, F_{\times}\}$ also vary with time due to the rotation of the Earth, however for short duration signals such as those from CBCs detected by current ground-based detectors, this effect is ignored [96]; so what remains is to define a model for the signal polarizations $\{h_{+}(t), h_{\times}(t)\}$.

The models that define the signal are generally referred to as waveforms or waveform approximants and they define the strain for a CBC signal as a function of a set of parameters $\boldsymbol{\theta}$. As discussed in section 1.2, these signals are comprised of three stages: the inspiral, merger and ringdown. There are different strategies for modelling CBC waveforms, these use different approximations, can be valid in different regions of the parameter space and model different parts of the signal. Typically, they do not model $\{h_{+}(t), h_{\times}(t)\}$ directly but instead decompose the signal into different spin-weighted spherical harmonic modes $h_{lm}(t)$, such that

$$h_{+}(t) - ih_{\times}(t) = \sum_{l=2}^{\infty}\sum_{m=-l}^{l} h_{lm}(t)_{-2}Y_{lm}(\theta, \phi), \tag{1.27}$$

where ${}_{-2}Y_{lm}$ are −2-spin-weighted spherical harmonics (see e.g. Breuer et al. [97]). The dominant modes are the $l = 2, m = \pm 2$ (or 2-2) modes and the other subdominant modes are referred to collectively as higher-order modes, see e.g. García-Quirós et al. [98] or Ramos-Buades et al. [99].

We now review the different methods for generating waveforms following Schmidt [100] but focusing on those that most relevant to the work presented in this thesis.

- **Post-Newtonian Theory:** these methods use the post-Newtonian formalism of General Relativity, which provides approximate solutions to the Einstein field equations and are valid in weak fields and for slow motion. These solutions systematically incorporate corrections to the Newtonian solution as a function of a small expansion parameters in powers of v/c [101]. This formalism is well suited to modelling the inspiral but not the merger or ringdown.

- **Numerical Relativity:** these methods numerically simulate the full relativistic two-body problem. The complexity of the simulations depends on the system being considered; simulations for BNSs include the full inspiral-merger-ringdown (IMR) [102, 103] whereas BBH simulations often only included the merger and ringdown [104–106]. Such simulations are computationally expensive and are therefore often used to validate other methods rather than being used directly for analyses [107]. Recent improvements to numerical relativity have allowed for simulations of more complex systems, for example, including precession or eccentricity [108–110].

- **Effective-one-body (EOB):** these methods treat the two-body system as a single test particle and solve a series of ordinary differential equations that describe the motion of the particle in a Kerr spacetime. They also incorporate post-Newtonian results to obtain a description of the complete IMR and are calibrated against numerical relativity simulations [111, 112]. Various improvements to the standard effective-one-body waveform exist, including treatments for high-order modes and precession [113–116].

- **Phenomenological Waveforms:** these methods directly model the IMR waveform in the frequency domain, rather than modelling the dynamics of the system and then calculating the gravitational-wave emission. The amplitude and phase evolution of each part of the IMR waveform are treated independently and use different models, which are based on Post-Newtonian terms at early times and are calibrated against numerical relativity for the merger and ringdown [107, 117]. The models are then combined to produce the final waveform. Different phenomenological waveforms have been developed, each incorporating different physical effects, the most recent of which is the `IMRPhenomX` family of waveforms [98, 118] which include precession and higher-order modes [119, 120].

These methods have vastly different computational costs, and this largely dictates their application. Phenomenological waveforms have been used extensively for stochastic sampling based analyses, whereas the typical cost of computing EOB waveforms has necessitated alternative approaches [7–10], however recent developments might allow for sampling with EOB waveforms [99, 121]. These analyses techniques are discussed in section 2.10.

1.4.3 Parameters

Irrespective of the waveform model that is used, they are all described by a set of common parameters $\boldsymbol{\theta}$. We can then denote the waveform at a given time t as $h(\boldsymbol{\theta}; t)$. The number of parameters will depend on the physics being described but they can generally be grouped into two distinct sets: intrinsic parameters that dictate the evolution in time of the waveform and extrinsic parameters that change the response of the detector to the signal.

We start with the extrinsic parameters since these are independent of the physics being modelled and therefore common to any waveform being used. First we define the parameters that determine the source's location and time: the sky location, the time of arrival of the signal and the distance to the source. The sky location is defined in terms of two angles, and different coordinates systems can be used. The equatorial system is described by right ascension α and declination δ, which are related to the angles defined in eq. (1.22) by $(\alpha, \delta) = (\phi - t_{\mathrm{GMST}}, \pi/2 - \theta)$ where t_{GMST} is the Greenwich mean sidereal time of arrival of the signal. Other systems, such as azimuthal angles defined based on the location of detectors, are also used [122]. These angles are used alongside a reference time, such as the time of the merger as measured at the centre of the Earth t_{c} or at a given detector t_{IFO} [122]. The use-cases for each are discussed in Romero-Shaw et al. [122]. For the distance to the source, we consider the luminosity distance d_{L}.

Having defined the parameters that are used to localize a source, what remains is to define the parameters that determine its orientation. This requires three angles that are analogous to Euler angles; this is discussed in more depth in section 6.2. We have already defined one of these angles in section 1.3.2: the polarization angle ψ, which describes the counter-clockwise rotation of the x–y plane in the source frame about the line-of-sight. It is important to highlight that the gravitational wave produced by a source that is rotated by $\psi = 2\pi$ is equivalent to that produced by a source with $\psi = \pi$, or more formally, gravitational waves are *spin-2* in nature. Therefore, we only consider the polarization over a range of π. Next, we consider the orientation of the orbital plane with respect to the x–y plane in the source frame. This can be described by either the inclination angle between the system's orbital angular momentum $\boldsymbol{L}$ and the line-of-sight, denoted as ι, or the inclination angle between the total angular momentum $\boldsymbol{J}$ and the line-of-sight,

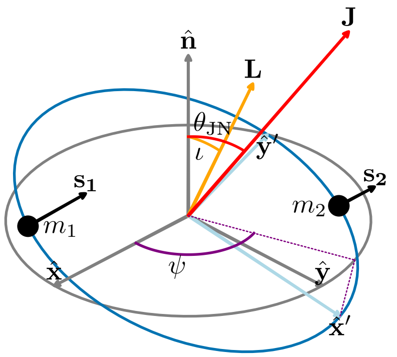


Figure 1.7: Diagram showing the geometry of a binary system of spinning compact objects in the source reference frame (non-prime), where $\hat{\boldsymbol{n}}$ is the line-of-sight vector to the observer. The prime coordinate system is defined such that the $\hat{\boldsymbol{x}}'$–$\hat{\boldsymbol{y}}'$ plane lies in the orbital plane and the orbital angular momentum $\boldsymbol{L}$ is pointed in the z-direction. The prime coordinate system is related to the source frame by the polarization angle ψ and an inclination angle, either ι or θ_{JN}. ι is defined between $\hat{\boldsymbol{n}}$ and $\boldsymbol{L}$ whilst θ_{JN} is defined between $\hat{\boldsymbol{n}}$ and the total angular momentum $\boldsymbol{J}$. m_1 and m_2 the masses of each compact object and $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ denote their spin vectors. The phase φ_c is not shown, but is measured in the $\hat{\boldsymbol{x}}'$–$\hat{\boldsymbol{y}}'$ plane about $\boldsymbol{L}$.

denoted θ_{JN} [123]. Finally, we define an orbital phase φ_c with respect to a reference time, which in most cases is the time of coalescence t_c. This geometry is shown in fig. 1.7 for a spinning system and shows the relevant angles.

Now we consider the intrinsic parameters; at the very least, we must define the component masses of the compact objects in the system, m_1 and m_2. These are defined such that the more massive object is labelled as 1, that is $m_1 \geq m_2$. This, alongside the extrinsic parameters, are enough to define a system of two non-spinning point masses. We can also define various alternative parameterizations of the masses, these include the total mass of the system

$$M = m_1 + m_2, \tag{1.28}$$

the asymmetric mass ratio

$$q = \frac{m_2}{m_1}, \tag{1.29}$$

and the chirp mass

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}. \tag{1.30}$$

The relationship between these parameters and the component masses is visualized in fig. 1.8, where contours of equal chirp mass and mass ratio are shown. We will revisit these parameters when discussing gravitational-wave data analysis in section 2.10.

More often, we will consider systems with spinning binaries. The spin for each compact object is a vector $\boldsymbol{s}_i$, with components $\boldsymbol{s}_i = (s_{ix}, s_{iy}, s_{iz})$, these are shown in fig. 1.7. Alternatively, the spin can be described using a dimensionless spin magnitude $a_i = |\boldsymbol{s}_i|/m_1^2 \in$

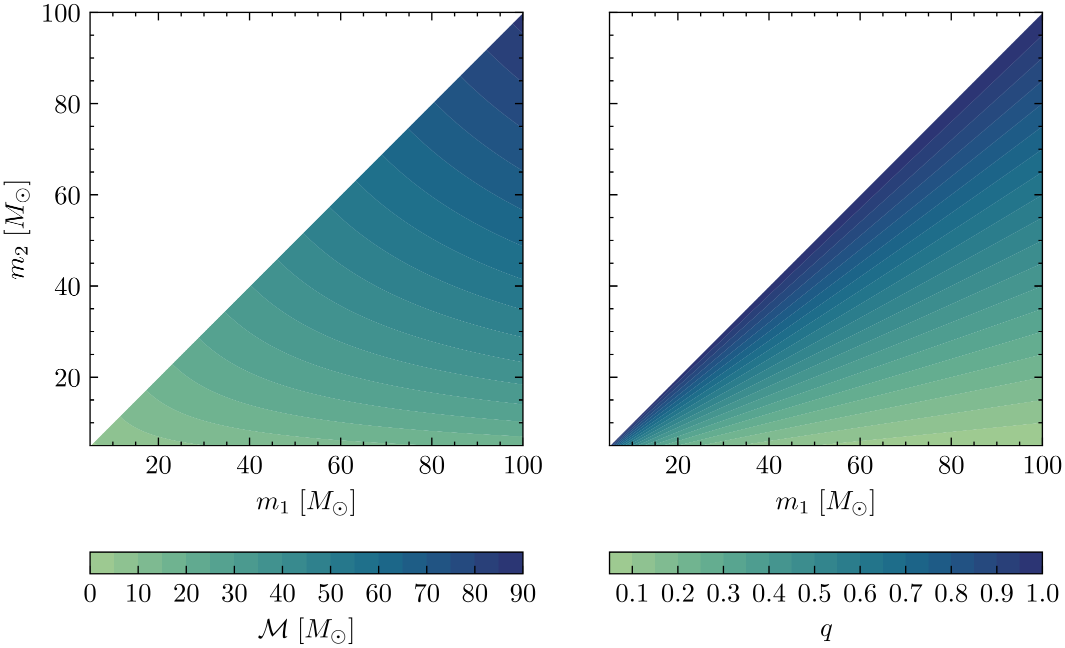


Figure 1.8: Contours of equal chirp mass $\mathcal{M}$ and mass ratio q in the component mass space where we have enforced $m_1 \geq m_2$.

$[0, 1]$ and a set of angles used to define the orientation [96, 123]. This is referred to as the *system frame* parameterization [123]. The angles used in this parameterization are the inclination angle θ_{JN} as defined previously, the inclination of the spins (or tilt angles) $\theta_{1,2}$, which are defined relative to $\boldsymbol{L}$, and two azimuthal angles: the first ϕ_{12} is the angle of $\boldsymbol{s}_2 - \boldsymbol{s}_1$ relative to $\boldsymbol{L}$ and the second, ϕ_{JL}, specifies the angle between $\boldsymbol{L}$ and $\boldsymbol{J}$. In addition to these parameters, we can define several derived parameters. The effective inspiral spin χ_{eff} [124, 125] is defined as

$$\chi_{\mathrm{eff}} = \frac{a_1 \cos\theta_1 + q a_2 \cos\theta_2}{1+q}, \tag{1.31}$$

where M is the total mass and, unlike the spin magnitudes, is reasonably well constrained [7]. When considering precessing systems, it can be informative to consider the effective precessing spin parameter χ_{p} [126, 127], which is typically defined as

$$\chi_{\mathrm{p}} = \max\left\{a_1 \sin\theta_1, q\frac{4q+3}{(4+3q)}a_2 \sin\theta_2\right\}, \tag{1.32}$$

though other definitions do exist, e.g. Gerosa et al. [128]. This parameter can capture the effect of spin-precession and can therefore be used to identify signs of precession in detected CBC events [8].

If additional physical effects are incorporated into the waveform, then additional intrinsic parameters may also be required. If analysing systems where at least one of the compact objects is a neutron star, then the effects of the tidal deformation can be included by introducing a tidal deformability parameter per neutron star Λ_i [129, 130].

1.5 The current state of gravitational-wave astronomy

The first direct detection of gravitational waves was made on September 14th 2015 by the two LIGO interferometers when they detected the gravitational radiation emitted by a stellar-mass BBH merger and heralded the beginning of gravitational-wave astronomy [6, 131]. This event is referred to as GW150914 and occurred at the start of the first aLIGO observing run, known as O1. This was followed by two subsequent detections also from BBH systems [7, 132].

The second observing run, O2, began in late 2016, initially with the two aLIGO detectors and later on, the AdVirgo detector. This observing run yielded a further seven BBH detections [8, 133–135] and the first detection of gravitational waves from a BNS system, GW170817 [29]. This event was also the first and, to date, only firm multi-messenger gravitational-wave observation [35, 38, 41, 136–139]: the LVK interferometers detected the gravitational waves emitted by the system and optical observatories observed the electromagnetic radiation emitted by the GRB and kilonova that followed the merger. This observation had implications for both fundamental physics and astrophysics [38, 140, 141], including for the speed of gravity [38], sources of GRBs [142], the Hubble constant [143] and neutron star equation of state (EOS) [141, 144, 145].

The third observing run, O3, began in April 2019, ran until March 2020, was divided in two parts: O3a and O3b, and saw KAGRA observing for the first time [146]. This observing run brought the total number of confident gravitational-wave candidates to 90 [8–10]. Amongst these were many interesting systems, including, but not limited to, the heaviest BBH to date GW190521 [147, 148], NSBH systems [149] and the first detection of precession [150].

The ever-increasing number of gravitational-wave detections enable a range of analyses, once the sources have been detected and characterized. These include analyses of the rates of CBCs, the astrophysical populations that the compact objects belong to and the formation channels that produce them [151–153], cosmological analyses that inform us about the fundamental structure of the Universe [143, 154–157] and analyses that test the validity of general relativity and determine if it is consistent with the observed gravitational-wave signals [158–162].

Looking forward, the next observing run, O4, started in May 2023 and is expected to last 18 months [163]. Various improvements have been made to the detectors, which are expected to increase the number of detected events by up to 3.3 times [74]. Past this there are plans for a fifth observing run with further improvements to all existing interferometers [164–166] and the potential for a fifth instrument in India [167].

These prospects bring with them a series of challenges. The last observing run, O3, showed us that the Universe can surprise us [148, 168]; new gravitational-wave observations will enable us to probe new physics [150, 169], detect potentially new sources and push the boundary of our understanding [170, 171]. However, this adds complexity to the analyses that must be performed and increases their computational cost [96, 172, 173]. Add to this the expected increase in the rate of observed events [74], and it is clear that there is a need for new and improved analysis techniques that can handle the increased volume and complexity of data that must be analysed. The work presented in this thesis explores one particular avenue of achieving this goal: machine learning applied to Bayesian inference.

1.6 Summary

This chapter introduces gravitational waves, starting from the linearized Einstein field equations. The solution to these equations in a vacuum is a transverse plane wave, which can be described in terms of two polarizations bases: the cross and plus polarizations. Such waves will stretch and squeeze spacetime in the plane perpendicular to their direction of propagation. The quadrupolar nature of gravitational waves means that only non-axisymmetric systems in motion will produce gravitational waves and their typical amplitudes will be $\mathcal{O}(10^{-23})$.

Four different types of gravitational-wave signals are then discussed: CBC, bursts, continuous gravitational waves and the SGWB. These are each produced by astrophysical sources and have distinct morphologies. The work presented in this thesis will focus on gravitational waves from CBCs which include: BBHs, BNSs and NSBHs. These sources produce distinct chirp-like signals that increase in frequency and amplitude as the systems evolve.

These gravitational-wave signals are detected using ground-based laser interferometers that are sensitive to strains $\mathcal{O}(10^{-23})$. Leveraging the data from such detectors requires a detailed understanding of how they respond to gravitational waves, which determines the antenna response patterns shown in fig. 1.5, and characterizing the noise present in the detectors, which we assume to stationary Gaussian noise described by a PSD. The data is then modelled as a combination of detector noise and a signal. Different waveforms are used to model the signals from CBC, such as phenomenological and EOB waveforms, and are parameterized by intrinsic parameters and extrinsic parameters.

Finally, we have summarized the state of gravitational-wave astronomy. There have been 90 confident detections to date, and these include all three expected CBC source types. These detections provide us with new means to understand the Universe and have implications for our understanding of fundamental physics and astrophysics. Future observations have similar prospects but bring with them a range of challenges that will need to addressed.

In this thesis, we consider on one particular challenge: improving Bayesian inference techniques for the analysis of gravitational waves from CBCs. Our aim is to investigate and develop means to accelerate Bayesian inference using machine learning and to bridge the gap between current techniques and novel end-to-end machine learning methods that are being developed. We focus on augmenting a specific algorithm, called nested sampling, which is widely used but can be computationally expensive. We centre our work around the novel idea of incorporating normalizing flows, a type of machine learning algorithm, into nested sampling; this idea is presented in chapter 4. We then iteratively develop and test this idea in chapters 5 to 8. With this in mind, we now review two key topics: Bayesian inference in chapter 2 and machine learning in chapter 3. In the former, we will start from Bayesian probability and work towards an understanding of nested sampling and how it is applied to gravitational-wave astrophysics. In the latter, we introduce the basics of machine learning and neural networks before focusing on normalizing flows.

Chapter 2

Bayesian inference

This chapter introduces the fundamentals of Bayesian inference, the techniques used to perform this inference and its application to gravitational-wave data analysis, with a particular focus on nested sampling,

Starting from the notion of Bayesian probability and Bayes' theorem, the key concepts needed to understand nested sampling are introduced; sections 2.1 to 2.3 introduce probability, probability density functions (PDFs) and Bayesian inference. Section 2.4 introduces the notion of sampling and resampling and section 2.5 extends this to Markov Chain Monte Carlo (MCMC). Section 2.6 reviews nested sampling in depth, including its formulation, design, implementation and variants; and provides context for the work presented in later chapters. From here, other methods for performing Bayesian inference are introduced, namely simulation based inference in section 2.7 and variational inference in section 2.8. Section 2.9 highlights methods for validating results from Bayesian inference that will be used in later chapters. Finally, section 2.10 details the application of Bayesian inference to gravitational-wave data analysis, first defining the gravitational-wave likelihood in section 2.10.1 before focusing on its application to gravitational-wave parameter estimation and various implementations that have been used to perform Bayesian inference in this field.

Throughout this chapter and later chapters, certain terms will be used with subject-specific meanings, such as using 'evidence' to refer specifically to the Bayesian evidence that arises from Bayes' theorem rather than to mean support for a particular hypothesis. Such terms are defined in the main text when they first appear and also in the glossary.

2.1 Probability

Probability encodes our degree of belief in a proposition, X, as a numerical value between 0 and 1. If the probability is 1, then the proposition is true, if it is 0, then it is false and values in between define how much we believe X to be true. Probability also incorporates the assumptions that have made to determine our degree of belief, which we denote I. This understanding of probability is known as *Bayesian* probability. It was first conceived by the Rev Thomas Bayes in the 1700s [174] and then developed by Pierre-Simon Laplace [175], but it was not until the 1900s that Sir Harold Jeffreys, Richard Cox and others formulated the Bayesian approach as we know it today [176, 177].

From this definition of probability, it follows that

$$p(X|I) + p(\bar{X}|I) = 1 \tag{2.1}$$

where $p(X|I)$ is the probability of the proposition X given the assumptions I and $\bar{X}$ is the negation of X, i.e. that proposition that X is false. This is known as the *sum rule*.

Now consider the case where we have two propositions X and Y. If we can specify the probability of Y being true and, subsequently, we can specify the probability of X being true given Y is also true, then we arrive at the *product rule*

$$p(X, Y|I) = p(X|Y, I)p(Y|I). \tag{2.2}$$

This defines the probability that both X and Y are true. These rules form the foundation of the Bayesian formulation of probability theory, and various results can be derived from them.

One such result is Bayes theorem [174]. Starting from the product rule, eq. (2.2), we note that it could equally be expressed as $p(X, Y|I) = p(Y|X, I)p(X|I)$. If we equate these two expressions and rearrange them, we arrive at

$$p(X|Y, I) = \frac{p(Y|X, I)p(X|I)}{p(Y|I)}, \tag{2.3}$$

which is known as Bayes theorem. These rules also lead to the concept of *marginalization*:

$$p(X|I) = \int_{-\infty}^{\infty} p(X, Y|I)\mathrm{d}Y, \tag{2.4}$$

which defines the probability of X irrespective of the value of Y. The importance of Bayes theorem and marginalization will become apparent when considering its application to data analysis, where we will no longer consider propositions X and Y, but data and hypothesis. Before this, we introduce the notion of probability density functions.

2.2 Probability density functions

Thus far, we have considered probability in the context of propositions X and Y. However, in data analysis applications, we more typically consider an arbitrarily large number of propositions, for example the possible distances to a binary black hole (BBH) system. This is known as the *continuum limit* [178] and, in this context, X is now a continuous numerical value instead of a single proposition. As such, $p(X|I)$ is now no longer a probability, but a probability density function (PDF) and, strictly speaking, should have different notation, however, here we will simply refer to probability and PDF as $p(\cdot)$[1]. Additionally, we will now refer to X as a random variable and use x to denote a value of X, that is $p(X = x|I)$ is the probability that X will take the value x, or simply $p(x)$.

For the PDF of a continuous random variable $p(X)$, the cumulative density function (CDF) is the probability that X is less than or equal to x

$$\mathrm{CDF}_X(x) = p(X < x) = \int_{-\infty}^{x} p(y)\mathrm{d}y, \tag{2.5}$$

and $\mathrm{CDF}_X(x) \in [0, 1]$.

We can also relate the PDFs for two random variables X and Y. Given a bijective function $f : X \to Y$ such that $y = f(x)$, the PDF for the random variable y is written using the change of variable as

$$p(y) = p(x)\left|\frac{\partial x}{\partial y}\right|. \tag{2.6}$$

2.2.1 A note on notation

Through this section, we will discuss conditional probabilities and functions with multiple arguments. These have similar notations but their interpretation is subtly different. A conditional probability $p(x|y)$ is the probability of x given y, i.e. a distribution on x, not y. Whereas in some cases where we have a function of two variables $f(x, y)$, we will write $f(x; y)$ to imply the value of f at x for a chosen value of y, that is, y is fixed.

2.2.2 Distance measures

Distance measures quantify the statistical difference between probability distributions. In this section, we will focus on two related distance measures: the Kullback-Leibler divergence (KLD) and the Jensen-Shannon divergence (JSD).

The KLD [179] is a measure of the distance between a probability distribution $p(x)$

[1] If X is discrete instead of continuous, e.g. the integers $\mathbb{Z}$, then $p(X|I)$ is a probability mass function (PMF) instead of a PDF.

and a reference probability distribution $q(x)$

$$D_{\mathrm{KL}}(p||q) = \int p(x) \log_b \frac{p(x)}{q(x)} \mathrm{d}x, \tag{2.7}$$

where b is the base of the logarithm, which we assume to be $b = e$ if not specified. Its units depend on the base being used, the most common are bits for $b = 2$ and nats for $b = e$. It is important to note that the KLD is not symmetric, i.e. $D_{\mathrm{KL}}(p||q) \neq D_{\mathrm{KL}}(q||p)$, which means it is not a metric. The KLD is defined such that for two correctly normalized probability distributions $D_{\mathrm{KL}}(p||q) \in [0, \infty)$, where 0 represents no difference and ∞ represents maximal difference.

The JSD is another distance measure that is commonly used, it is a symmetric version of the KLD and measures the similarity between two distributions

$$D_{\mathrm{JS}}(p||q) = \frac{1}{2} D_{\mathrm{KL}}(p||m) + \frac{1}{2} D_{\mathrm{KL}}(q||m), \tag{2.8}$$

where $m(x) = (1/2)[p(x) + q(x)]$. It can therefore be used to compare distributions when there is no known reference. Unlike the KLD, the JSD is bounded [180] and its bounds depend on the base b of the logarithm used

$$0 \leq D_{\mathrm{JS}}(p||q) \leq \log_b(2). \tag{2.9}$$

As a result, for $b = 2$ the JSD is bounded on $[0, 1]$ which, as will be seen later, is often used when comparing posterior distributions. Whilst the JSD is also not a metric, the square root of it is and is known as the Jensen-Shannon distance.

2.3 Bayesian inference

We now consider Bayes' theorem eq. (2.3) in the context of data analysis, specifically parameter estimation. We will replace the propositions X and Y with a parameter $\boldsymbol{\theta}$ that describes a model H, which replaces I, and some data $\boldsymbol{d}$. We will then write Bayes' theorem as

$$p(\boldsymbol{\theta}|\boldsymbol{d}, H) = \frac{p(\boldsymbol{d}|\boldsymbol{\theta}, H) p(\boldsymbol{\theta}|H)}{p(\boldsymbol{d}|H)}. \tag{2.10}$$

We can now see the importance of Bayes theorem: it allows us to relate the probability of a hypothesis (that some parameter $\boldsymbol{\theta}$ of a given model H describes the data) given some data $\boldsymbol{d}$ ($p(\boldsymbol{\theta}|\boldsymbol{d}, H)$) to the probability that we observed the data $\boldsymbol{d}$ given the hypothesis was true ($p(\boldsymbol{d}|\boldsymbol{\theta}, H)$). This is the foundation of Bayesian inference.

The terms in Bayes' theorem each have formal names: $p(\boldsymbol{\theta}|H)$ is known as the prior probability or $\pi(\boldsymbol{\theta})$ and describes the state of knowledge prior to obtaining the data;

$p(\boldsymbol{d}|\boldsymbol{\theta}, H)$ is the probability of the data given the parameter and the model, and is known as the likelihood and often shortened to $\mathcal{L}(\boldsymbol{\theta})$; $p(\boldsymbol{\theta}|\boldsymbol{d}, H)$ is the probability of the parameter given the model and the data, known as the posterior probability; and $p(\boldsymbol{d}|H)$ is the probability of data given only the model, known as the Bayesian evidence Z or marginalized likelihood. We can define the evidence using eq. (2.4):

$$Z = p(\boldsymbol{d}|H) = \int p(\boldsymbol{d}, \boldsymbol{\theta}|H) p(\boldsymbol{\theta}|H) \mathrm{d}\boldsymbol{\theta}. \tag{2.11}$$

Typically, we extend this formulation to include multiple parameters, denoted by $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_N\}$. Thus, the probabilities describe the state of knowledge of all the parameters and their relationships. If we are then interested in a particular parameter, we can marginalize over the other parameters:

$$p(\theta_1|\boldsymbol{d}, H) = \int \mathrm{d}\theta_2 ... \mathrm{d}\theta_N p(\boldsymbol{\theta}|\boldsymbol{d}, H). \tag{2.12}$$

Directly computing the posterior probability would require closed forms for the likelihood, prior and evidence. In different scenarios, defining each of these can be challenging or even impossible. In this thesis, we focus on the case where one can define a likelihood and prior, but the evidence integral eq. (2.11) does not have an analytic solution, i.e. it is analytically intractable. However, we discuss the case where the likelihood does not have a closed-formed, also referred to as being intractable, in section 2.7. Estimating the posterior, or drawing samples from it, requires at the very least sampling from the numerator in eq. (2.10), however a closed form for this does not typically exist, so we instead have to resort to different techniques for sampling from a PDF or an unnormalized PDF.

In sections 2.4 to 2.6, we introduce different methods for sampling from PDFs and, is some cases, computing the normalization constant. Before that, we briefly introduce another application of Bayesian inference: model selection.

2.3.1 Model selection

Whilst we have considered Bayesian inference in the context of parameter estimation, it can also be applied to model selection, where the aim is to compare different models that describe the same data. Generally, we can compare pairs of models (H_i and H_j) using the posterior odds

$$O_{ij} = \frac{p(H_i|\boldsymbol{d})}{p(H_j|\boldsymbol{d})}, \tag{2.13}$$

which, using Bayes' theorem, we can rewrite as

$$O_{ij} = \frac{p(\boldsymbol{d}|H_i)}{p(\boldsymbol{d}|H_j)} \frac{p(H_i)}{p(H_j)} = \frac{Z_i}{Z_j} \frac{p(H_i)}{p(H_j)}, \tag{2.14}$$

where the term $p(\boldsymbol{d})$ cancels out. The ratio of the evidences is known as the Bayes factor ($\mathcal{B}$), and if the prior odds for the two models are the same, then it is equal to the posterior odds. Posterior odds greater than one indicate that H_1 is favoured over H_2, however interpreting the degree to which a model is favoured is challenging. Jeffreys [176] provided grades based on the value of O_{ij} which correspond to levels of significance.

2.4 Sampling from distributions & resampling

There are cases where we need to draw samples from a particular distribution $p(\boldsymbol{\theta})$ but $p(\boldsymbol{\theta})$ cannot be sampled from directly, for example an unnormalized posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{d}, H) \propto p(\boldsymbol{d}|\boldsymbol{\theta}, H)p(\boldsymbol{\theta}|H)$. However, if $p(\boldsymbol{\theta})$ can be computed, and we have another distribution $q(\boldsymbol{\theta})$, known as the proposal distribution, which can be sampled from directly and its PDF computed, then we can use resampling to obtain samples from $p(\boldsymbol{\theta})$. There are many different resampling methods [181], however in this work we focus on two that have been used extensively in gravitational-wave data analysis: rejection sampling and multinomial resampling. For a set of samples $\Theta \equiv \{\boldsymbol{\theta}_i\}_{i=1}^{N}$ drawn from the proposal distribution, i.e. $\boldsymbol{\theta}_i \sim q(\boldsymbol{\theta})$, both methods make use of weights

$$\alpha_i \propto \frac{p(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}, \tag{2.15}$$

which are normalized such that $\sum_{i=1}^{N} \alpha_i = 1$. These methods are well suited to cases where a proposal distribution that closely 'resembles' the target, i.e. the PDF $q(\boldsymbol{\theta})$ has similar support and shape as $p(\theta)$, can be constructed and sampled from. If the proposal distribution and target are significantly different, then these methods will become inefficient due to a low number of samples having large weights. Similarly, these methods also become increasingly inefficient as the number of dimensions and complexity of $p(\boldsymbol{\theta})$ increases. In section 2.4.3, we will see how to quantify the efficiency using the effective sample size.

2.4.1 Multinomial resampling

In multinomial resampling [182], the core idea is to return N samples by sampling N times from a uniform distribution $u_i \sim \mathcal{U}[0, 1]$ and then use the values of u_i to select samples from Θ, where $\Theta \equiv \{\boldsymbol{\theta}_i\}_{i=1}^{N}$ and $\boldsymbol{\theta}_i \sim q(\boldsymbol{\theta})$. The samples are selected by first sorting Θ

given the normalized weights α_i from eq. (2.15) and then computing the cumulative sum

$$C_i = \sum_{j=1}^{i} \alpha_j. \tag{2.16}$$

For each u_i, the corresponding sample from Θ is the first sample for which $C_i \geq u_i$.

The advantage of this method is that it returns a specified number of samples, however, these samples will likely contain duplicates and this must be accounted for in any analyses that make use of these samples.

2.4.2 Rejection sampling

Rejection sampling is a resampling method where samples from the proposal distribution are accepted or rejected based on an acceptance probability. For a given sample, the acceptance probability is compared to a random draw from a uniform distribution $u_i \sim \mathcal{U}[0, 1]$ and the sample is accepted when

$$\frac{p(\boldsymbol{\theta}_i)}{Mq(\boldsymbol{\theta}_i)} = \frac{\alpha_i}{M} \geq u_i, \tag{2.17}$$

where $M = \sum_{i=1} \alpha_i$ is a normalization constant, and rejected otherwise.

In contrast to multinomial resampling, for a given set of samples Θ, rejection sampling returns a random number of samples from the target distribution. However, these samples are statistically independent (i.e. they do not contain duplicates) and are therefore more comparable to samples drawn using other methods, for example, directly from a distribution or using other Monte Carlo methods such as Markov Chain Monte Carlo (MCMC) and nested sampling.

2.4.3 Effective sample size

When sampling from a target distribution $p(\boldsymbol{\theta})$ using a proposal distribution $q(\boldsymbol{\theta})$ and corresponding weights α_i it can be informative to compute Kish's effective sample size (ESS) [183]

$$\text{ESS} = \frac{\left[\sum_{i=1}^{N} \alpha_i\right]^2}{\sum_{i=1}^{N} \alpha_i^2}, \tag{2.18}$$

for a set of samples $\Theta \equiv \{\boldsymbol{\theta}_i\}_{i=1}^{N} \sim q(\boldsymbol{\theta})$. This gives an indication of how statistically significant (or independent) the samples are given the weights. A low ESS indicates that the samples drawn from $q(\boldsymbol{\theta})$ are a poor match for $p(\boldsymbol{\theta})$, i.e. the sampling is inefficient. The ESS can also serve as an upper limit for how many samples would be obtained when using rejection sampling.

2.5 Markov Chain Monte Carlo

MCMC algorithms are a method for sampling from a probability distribution $p(\boldsymbol{\theta})$ by constructing a Markov chain. They iteratively explore the parameter space and produce samples that are distributed proportionally to the target distribution. In the context of Bayesian inference, the target distribution is the unnormalized posterior probability $p(\boldsymbol{\theta}) \propto \mathcal{L}(\boldsymbol{\theta})\pi(\boldsymbol{\theta})$, which avoids having to compute the normalizing integral.

There are many different variants of MCMC, such as Gibbs sampling [184] and Hamiltonian Monte Carlo [185], but for the purpose of understanding how these algorithms work, we will focus on the Metropolis-Hastings algorithm [186, 187]. For a complete review of MCMC algorithms see Hogg and Foreman-Mackey [188].

In the Metropolis-Hastings algorithm, a new sample $\boldsymbol{\theta}'$ is drawn from a conditional density (or proposal) $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ that depends on the current sample $\boldsymbol{\theta}$. At a given step, the Markov chain can transition from the current sample to the new sample with probability

$$r(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left\{\frac{p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}, 1\right\}. \tag{2.19}$$

If it is accepted, $\boldsymbol{\theta}'$ is added to the chain otherwise the existing sample $\boldsymbol{\theta}$ is added. This allows for the use of asymmetric proposal distributions, i.e. $q(\boldsymbol{\theta}'|\boldsymbol{\theta})/q(\boldsymbol{\theta}'|\boldsymbol{\theta}) \neq 1$, which can guide the chain towards regions of higher probability, such as in Hamiltonian Monte Carlo [185]. If the proposal is symmetric, then $q(\boldsymbol{\theta}'|\boldsymbol{\theta})/q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = 1$ and the algorithm reduces to the Metropolis algorithm [189].

MCMC algorithms do not estimate the normalization factor of the target density; the Bayesian evidence Z when the target is the posterior. However, the use of *tempering* [190] enables the use of thermodynamic integration for computing the integral. This introduces a 'temperature' dependency $T = 1/\beta$ to the target distribution

$$p(\boldsymbol{\theta}|\beta) \propto \pi(\boldsymbol{\theta})\mathcal{L}(\boldsymbol{\theta})^{\beta}, \tag{2.20}$$

and allows for the logarithm of the normalization constant to be estimated via

$$\ln Z = \int_0^1 E_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\beta)}\left[\ln \mathcal{L}(\boldsymbol{\theta})\right] \mathrm{d}\beta, \tag{2.21}$$

where the integrand is the expectation value of the log-likelihood for samples drawn from $p(\boldsymbol{\theta}|\beta)$. This requires sampling multiple chains with different temperatures, which can be done in parallel, and then combining them to obtain the final estimate. For the complete derivation, see e.g. Friel and Pettitt [191]. The inclusion of a temperature also changes the nature of the likelihood surface and can improve the sampling of complex likelihoods. However, the evidence estimates obtained via this method often have larger uncertainties since they depend on the number of temperatures [96].

2.6 Nested sampling

Nested sampling is a stochastic algorithm that was first proposed by Skilling [192, 193] and is designed to compute the multidimensional integrals. In this work we focus on its application to Bayesian inference, where it is used to compute the Bayesian evidence Z and obtain posterior samples, the latter are by-product of the algorithm, however it also has other applications, see e.g.Pártay et al. [194]. We now review the mathematical formulation and core algorithm, and then discuss different variants that are relevant to the work presented in later chapters. For more details on nested sampling, see Skilling [193], which introduces the algorithm, and Ashton et al. [195], which is a comprehensive review for physical scientists; both complement the explanation in this work.

The Bayesian evidence Z as defined in eq. (2.11) requires marginalizing over the likelihood, which quickly becomes expensive for multidimensional parameter spaces. In nested sampling, instead of computing the n-dimensional integral, the evidence is redefined in terms of a new quantity known as the prior volume X which is defined on the unit interval, where a prior volume of 1 contains all the prior probability and decreasing values of prior volume contain less. The evidence, eq. (2.11), is then written (using the shorthand notation for the likelihood $\mathcal{L}(\boldsymbol{\theta})$ and prior $\pi(\boldsymbol{\theta})$) as

$$Z = \int_{\Theta} \mathcal{L}(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \int_0^1 \bar{\mathcal{L}}(X)\mathrm{d}X, \tag{2.22}$$

where, Θ is the domain of the parameters, $\mathrm{d}X = \pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$ and $\bar{\mathcal{L}}(X)$ is the likelihood as a function of prior volume. The integrand is this equation is positive and decreasing, so the function must be well-behaved [193]. This idea is illustrated in fig. 2.1 where a two–dimensional parameter space is shown with four likelihood contours. The corresponding plot of prior volume versus likelihood is also shown.

We arrive at eq. (2.22) by considering the prior volume contained with a likelihood contour with likelihood $\lambda \equiv \mathcal{L}(\boldsymbol{\theta})$, which we can define as

$$X(\lambda) = \int_{\mathcal{L}(\boldsymbol{\theta})>\lambda} \pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{2.23}$$

such that

$$Z = \int_{-\infty}^{+\infty} X(\lambda)\mathrm{d}\lambda. \tag{2.24}$$

Equation (2.22) can then be derived using integration by parts, assuming that $\bar{\mathcal{L}}(X)$ is the inverse of $X(\lambda)$; see Ashton et al. [195, Box 2] for the proof.

To approximate the evidence, we must compute the area under the curve shown in

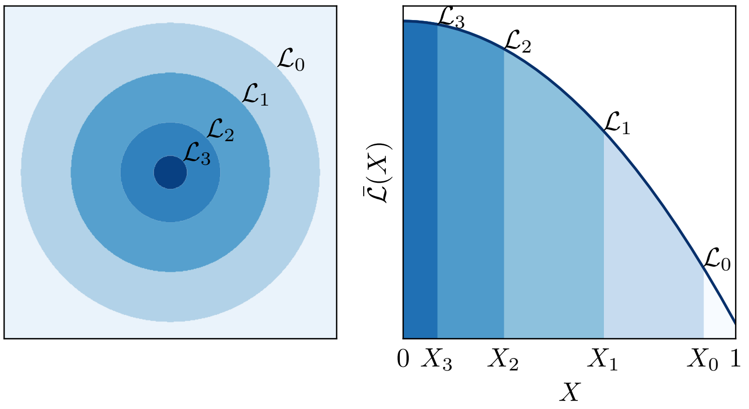


Figure 2.1: Visualization of the core nested sampling idea adapted from Skilling [193]. **Left:** a two–dimensional parameter space showing four iso-likelihood contours labelled by increasing likelihood and coloured such that a darker colour corresponds to a higher likelihood. **Right:** plot of the prior volume X versus likelihood $\bar{\mathcal{L}}(X)$. The same four likelihood contours are shown in the same colours and contain, from lowest to highest likelihood ($\mathcal{L}_0$ to $\mathcal{L}_3$), 90%, 60%, 30% and 10% of the prior volume.

fig. 2.1. If the likelihood is a well-behaved function, we can consider an ordered sequence of M point with decreasing the prior volumes X_i:

$$0 < X_M < \cdots < X_2 < X_1 = 1, \tag{2.25}$$

and evaluate the likelihood at each point $\mathcal{L}_i = \bar{\mathcal{L}}(X_i)$. Each prior volume then, in the original parameter space, corresponds to a contour of equal likelihood, this is shown in fig. 2.1. The evidence integral from eq. (2.22) can then be approximated by

$$Z = \sum_{i=1}^{M} w_i \mathcal{L}_i, \tag{2.26}$$

where the weights w_i can be approximated, for example using the Trapezoidal rule

$$w_i = \frac{1}{2}(X_{i-1} - X_{i+1}), \tag{2.27}$$

with reflecting boundary conditions $X_0 = 2 - X_1$ and $X_{M+1} = -X_M$.

What remains is how to compute the prior volume X. We required the prior volumes to be ordered and decreasing, so we can relate them via a shrinkage factor $t_i \in (0, 1)$

$$X_i = t_i X_{i-1}, \tag{2.28}$$

with $X_1 = 1$. But how do we compute the shrinkage between prior volumes?

This is where the core nested sampling idea comes into play; we do so statistically. At a given step i, we can randomly draw a new point X_i such that $X_i < X_{i-1}$ by sampling

$t_i \sim \mathcal{U}(0,1)$ and then computing X_i using eq. (2.28). However, this would be impractical since we would need to a function to map from prior volume to likelihood and then to parameters $\boldsymbol{\theta}$. Instead, we can use the relationship between the prior volume and likelihood, and sample a new random point $\boldsymbol{\theta}_i$ from the prior with the constraint that $\mathcal{L}(\boldsymbol{\theta}_i) > \mathcal{L}_{i-1}$. We must then compute the prior volume X_i for a given likelihood.

If we consider N_{live} random samples from the prior within a likelihood contour, which we call *live points*, each will have a corresponding prior volume and likelihood. If we remove the *worst* point (lowest $\mathcal{L}$, highest X), then the next prior volume will be related via the shrinkage t_i which will be distributed according to a beta distribution given by [193, 196]

$$t_i \sim \beta(t_i; N_{\text{live}}, 1) = N_{\text{live}} t_i^{N_{\text{live}}-1}. \tag{2.29}$$

We can then use this knowledge to estimate the mean and variance for t_i, given N_{live},

$$\text{Mean}[t_i] = \frac{N_{\text{live}}}{N_{\text{live}}+1}, \quad \text{Var}[t_i] = \frac{N_{\text{live}}}{(N_{\text{live}}+1)^2(N_{\text{live}}+2)}, \tag{2.30}$$

and in turn compute the prior volume X_i. However, Skilling [193] proposes using the expectation value of $\log t_i$ since it dominates the geometrical exploration

$$\text{Mean}[\log t_i] = -\frac{1}{N_{\text{live}}}, \quad \text{Var}[\log t_i] = \frac{1}{N_{\text{live}}^2}. \tag{2.31}$$

Therefore, if we ignore the uncertainty, $\log X_i = \sum_{j=1}^{i} \log t_j = -i/N_{\text{live}}$.

Having determined the prior volume, we then draw a new point from the prior subject to the likelihood constraint given by the worst point and we end with N_{live} points again, so we can repeat the process. This step if the crux of nested sampling, since it most cases the likelihood-constrained prior is not defined analytically.

2.6.1 Nested sampling algorithm

We now have all the building blocks to construct the nested sampling algorithm. The complete algorithm is outlined in alg. 1 which we will now describe.

The algorithm starts by drawing N_{live} points from the prior $\pi(\boldsymbol{\theta})$. The worst point $\boldsymbol{\theta}^*$, with the lowest likelihood, is then selected and used to update the prior volume and in turn the evidence. A replacement point is then drawn from prior with the likelihood constraint $\mathcal{L}(\boldsymbol{\theta}) > \mathcal{L}_i$. The process is then repeated; the worst point is used to update evidence and then replaced with a new point. This process continues until a termination criterion is reached.

Once the termination criterion is reached, we are left with a final set of live points. These points are uniformly distributed with the final prior volume and can be used to improve the evidence estimate by sequentially adding them in order of increasing likelihood. The shrinkage in prior volume can be estimated via eq. (2.31), but N_{live} must be decreased by 1 each time a point is added, since it is not replaced.

Algorithm 1: Nested sampling algorithm

```
Data: Prior $\pi$, likelihood $\mathcal{L}$
Result: Evidence $Z$
$Z = 0$, $X_0 = 1$, $\mathcal{L}_0 = 0$, $i = 0$;
Sample $N_{\text{live}}$ points $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_{\text{live}}}\}$ from the prior;
while termination criterion not reached do
    $i \leftarrow i + 1$;
    Set $\mathcal{L}_i$ to the lowest likelihood value in the current samples;
    Estimate $\log t_i$ and then $X_i$ via eqs. (2.28) and (2.31);
    Compute $w_i$ via eq. (2.27);
    Update the evidence $Z$ by $\mathcal{L}_i w_i$;
    Replace worst point with a new point drawn the prior with $\mathcal{L}(\boldsymbol{\theta}) > \mathcal{L}_i$;
end
for $\boldsymbol{\theta}_i$ in final live points do
    Set $\mathcal{L}_i$;
    $N_{\text{live}} \leftarrow N_{\text{live}} - 1$;
    Estimate $X_i$ via eq. (2.28) with updated $N_{\text{live}}$;
    Compute $w_i$ via eq. (2.27); Update evidence $Z$ by $\mathcal{L} w_i$;
end
```

2.6.2 Termination criteria

Given the nested sampling formulation, we must then determine when to terminate sampling. Unlike MCMC, nested sampling does not continuously produce posterior samples, so we cannot typically use the number of samples as a termination criterion. If we have domain knowledge, for example the maximum likelihood is known a priori then this could be used. However, this is not generally possible, so more often the termination is determined at runtime.

Skilling [193] proposes using the maximum likelihood of the current live points and computing a maximum contribution to the evidence given the current prior volume $\hat{Z}_{\text{live}} = \mathcal{L}_{\max} X_i$. The ratio

$$\Delta \hat{Z}_i = \frac{\hat{Z}_{\text{live}} + \hat{Z}_i}{\hat{Z}_i}, \tag{2.32}$$

where $\hat{Z}_i$ is the evidence at the current iteration, can then be used as a termination criterion [197]; when it becomes very small the change in the evidence by adding more samples should be minimal. Different values have been used in the literature, but $\ln \Delta \hat{Z}_i \leq$

0.1 has been widely used within gravitational-wave data analysis [96, 198].

2.6.3 Uncertainty estimates

The statistical uncertainty in nested sampling arises from estimating the shrinkage t_i. It has been shown that this can be quantified by considering the information content $\mathcal{H}$ of the posterior $p(\boldsymbol{\theta})$ compared to the prior $\pi(\boldsymbol{\theta})$ [193, 195], computed using the KLD

$$\mathcal{H} = \int p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta}. \tag{2.33}$$

If one rewrites this in terms of the prior volume and assumes the uncertainty is dominated by the region around the posterior mass, then the uncertainty on the log-evidence can be taken to be approximately Gaussian [193]

$$\sigma[\ln Z] \approx \sqrt{\frac{\mathcal{H}}{N_{\text{live}}}}. \tag{2.34}$$

However, this estimate has been shown to be unreliable in some cases [199], for example when the nested samples are not independently and identically distributed (i.i.d.), and more reliable estimates can be obtained via repeated simulation [200].

2.6.4 Posterior estimation

Nested sampling can also be used to perform parameter inference. Posterior weights can be computed for a given set of nested samples $\Theta \equiv \{\boldsymbol{\theta}_i\}_{i=1}^{N}$, these will be proportional to the prior volume of each sample

$$p_i = \frac{\mathcal{L}_i w_i}{Z}, \tag{2.35}$$

where the weights w_i are the weights from eq. (2.26). The weights are in fact random variables described by the distribution of shrinkage ratios. This introduces an additional uncertainty which is explored in Higson et al. [200], however, in practice, most implementations of nested sampling use the expectations values for the weights Additionally, rather than using the nested samples and their corresponding weights, it is more common to resample the nested samples to obtain a set of posterior samples since it can be difficult to incorporate weighted samples into subsequent analyses. For the resampling, the weights, i.e. α_i from eq. (2.15), are the posterior weights given by eq. (2.35) and these are used with either of the methods outlined in section 2.4.

2.6.5 Likelihood constrained sampling

During the nested sampling algorithm (alg. 1), the worst point $\boldsymbol{\theta}^*$ is discarded and a replacement $\boldsymbol{\theta}_i$ is then be drawn. This replacement should be i.i.d., drawn from the prior and must satisfy the likelihood constraint $\mathcal{L}(\boldsymbol{\theta}_i) > \mathcal{L}(\boldsymbol{\theta}^*)$. However, this can be challenging since a closed-form for sampling from the likelihood-constrained prior does not typically exist and the likelihood contour may be complex; for example, it may have correlations between certain parameters or be multimodal. Skilling [193] proposed using MCMC but did not explore the method in much depth. Three main families of methods have since been developed to address this challenge.

Random walk sampling

Random walk-based nested sampling algorithms use MCMC algorithms to draw new samples from the likelihood-constrained prior. This requires accepting only those points for which $\mathcal{L}(\boldsymbol{\theta}) > \mathcal{L}^*$ until the correlation with the starting point (one of the existing samples) has been lost. This method also requires a random walk that can adapt to the continuously shrinking constrained prior and a method for determining the number of steps to take [196]. Further modifications are often needed to handle multi-modality and complex correlations between parameters, for example, as implemented in [96].

Similar to this is the use of slice sampling [201], where samples are drawn from a randomly oriented line within the prior volume. The challenge in this case is choosing the direction of the line and how to sample from it.

Nested samplers that implement random walk based sampling include `cpnest` [202], `DNest` [203, 204], `dynesty` [197], `LALInference` [96], `nestle` [205], `PolyChord` [206] and `ultranest` [207].

Direct (region) sampling methods

Alternatively, various methods have been proposed for approximating the likelihood constrained prior or a superset of it with a proposal distribution. The challenge in this case is how to construct the proposal distribution. These are typically bounding ellipsoids that contain the live points and are then enlarged by some factor to avoid missing regions of the parameter space. In cases where the problem is more or complex or multi-modal, the bounding distribution can be constructed using multiple ellipsoids [208]. More recent variants of this idea include the RadFriends [209] and MLFriends [210] algorithms. Nested samplers that implement direct sampling methods include `DIAMONDS` [211], `dynesty` [197], `multinest` [208, 212], `nestle` [205] and `ultranest` [207].

Hybrid methods

Finally, there are algorithms that use a combination of random walk and direct sampling methods. These aim to improve the efficiency by leverage the advantages of both methods, namely the direct sampling methods reduce the volume of space that needs to be sampled whereas the random walk methods have better scaling as the number of dimensions increases. Nested samplers that implement a combination of both methods includes `dynesty` [197] and `ultranest` [207]

2.6.6 Variants of nested sampling

Numerous variants of nested sampling have been proposed that modify parts of the core algorithm. For example, some employ different methods for computing the weights used to compute the evidence [212, 213] and others modify how samples are distributed throughout the run [203]. We now review four variants of nested sampling that we will draw upon in this thesis.

Diffusive nested sampling

Diffusive nested sampling [213] uses a multi-level exploration method where a mixture of constrained distributions is sampled from at each iteration using MCMC. The constrained distributions are added sequentially, and each contains approximately e^{-1} of the prior volume of the previous. In contrast to standard nested sampling approaches, all the samples from the MCMC chain are kept and those that do not meet the current likelihood criteria are added to the previous level. The values for the prior volume X are estimated using the fraction of samples above the likelihood threshold compared to the total number of samples.

Dynamic nested sampling

Dynamic nested sampling was proposed in Higson et al. [203] and introduces a framework for varying the number of live points throughout a nested sampling run. Since the shrinkage in prior volume is related to the number of live points (eq. (2.29)), this allows us to control the shrinkage over the course of the run. This can be used to focus on specific regions of the parameter space, depending on the focus of the analysis. For example, focusing on the region of the parameter space where the bulk of the posterior probability is located or focusing on regions that improve the evidence estimate. Higson et al. [203] propose two importance functions that measure the relative importance of the ith sample for the evidence and posterior calculation

$$I_Z(i) \propto \frac{Z_{\geq i}}{N_{\text{live}}^{(i)}}, \quad I_{\text{Post}}(i) \propto \mathcal{L}_i w_i, \tag{2.36}$$

where $Z_{\geq i}$ is the evidence contribution from samples with likelihoods greater than or equal to $\mathcal{L}_i$ and $N_{\text{live}}^{(i)}$ is the number of live points at ith iteration.

The dynamic nested sampling algorithm is a two stage process. It starts with a standard exploratory nested sampling run with a constant number of live points, which results in a set of nested samples. The relative importance of the samples is then defined as the combination of the two terms from eq. (2.36) with a user-defined weight $\gamma \in [0, 1]$ known as the goal [203] (G in Higson et al. [203]),

$$I(\gamma, i) = (1 - \gamma)\frac{I_Z(i)}{\sum_j I_Z(j)} + \gamma\frac{I_{\text{Post}}(i)}{\sum_j I_{\text{Post}}(j)}. \tag{2.37}$$

Additional samples are then added in regions with high importance $I(\gamma, i)$.

Importance nested sampling

Importance nested sampling was proposed in Cameron and Pettitt [214] and expanded upon in Feroz et al. [212]. In this version of nested sampling, the evidence integral is approximated in terms of a *pseudo-importance sampling density* $Q(\boldsymbol{\theta})$

$$\hat{Z} = \frac{1}{N_{\text{Total}}}\sum_{i=1}^{N_{\text{Total}}}\frac{\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{Q(\boldsymbol{\theta}_i)}, \tag{2.38}$$

where N_{Total} is the total number of nested samples. Posterior weights are then computed using

$$p_i = \frac{\mathcal{L}_i(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{N_{\text{Total}}Q(\boldsymbol{\theta}_i)}, \tag{2.39}$$

and these can be used as you would the posterior weights in a normal nested sampling algorithm.

In standard importance nested sampling, the unbiased estimator for the variance of the evidence is given by [212, 214]

$$\sigma^2[\hat{Z}] = \frac{1}{N_{\text{Total}}(N_{\text{Total}} - 1)}\sum_{i=1}^{N_{\text{Total}}}\left[\frac{\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{Q(\boldsymbol{\theta}_i)} - \hat{Z}\right]^2, \tag{2.40}$$

however, as discussed in [212], this does not apply when using a pseudo-importance sampling density, which is the case in `multinest`.

In `multinest` [208, 212], one or more ellipsoidal distributions are used to construct an approximation of the current likelihood contour defined by $\mathcal{L}^*$. New points are then drawn from within this proposal distribution and their likelihood evaluated until $\mathcal{L}(\hat{\boldsymbol{\theta}}) > \mathcal{L}^*$ and, similarly to diffusive nested sampling, all of these points are used in the evidence summation and define the number of points within a level n_i. The pseudo-importance sampling density for each point is given by

$$Q(\boldsymbol{\theta}) = \frac{1}{N_{\text{Total}}} \sum_{i=1}^{N_{\text{iter}}} \frac{n_i E_i(\boldsymbol{\theta})}{V_{\text{tot},i}}, \tag{2.41}$$

where $V_{\text{tot},i}$ is the volume of the bounding distribution, E_i is an indicator function that is 1 if the point lies within the i'th ellipsoidal decomposition and 0 otherwise, N_{iter} is the number of iterations, where an iteration is an instance of the ellipsoidal decomposition and N_{Total} is the total number of points $N_{\text{Total}} = \sum_{i=1}^{N_{\text{iter}}} n_i$.

Nested Sampling via Sequential Monte Carlo

Sequential Monte Carlo (SMC) is a general extension of importance sampling where random samples with corresponding weights are drawn from a sequence of probability densities such that they converge towards a target density [215]. These algorithms are typically comprised of three main steps: *mutation* in which the samples are moved towards the target density via a Markov kernel, *correction* where the weights of the samples are updated, and *selection* where the samples are resampled according to their weights.

In Salomone et al. [216], the authors draw parallels between nested sampling and SMC and show that nested sampling is a type of adaptive SMC algorithm where weights are assigned sub-optimally. They also highlight several limitations of the standard nested sampling algorithm, including the assumption of independent samples. They propose a new class of SMC algorithms called Nested Sampling via Sequential Monte Carlo (NS-SMC) and demonstrate that it is equivalent to nested sampling but address the aforementioned limitations. This formulation bears similarities to the importance nested sampling as described in [212, 214] but removes batches of live points at each iteration and includes the mutation and selection steps that are typical in SMC.

2.6.7 Nested sampling diagnostics

Nested sampling algorithms have a range of tuneable settings, some of these will be common across most implementations, for example the number of live points N_{live}, whilst others will be implementation specific, such as the choice of proposal kernel in an implementation that uses an MCMC based random walk for drawing new points. The choice of settings can impact the results obtained, and inappropriate setting may lead to biases in the results; e.g. missing a mode in the posterior distribution or over-estimating the evidence. General purpose checks, such as those discussed later in section 2.9 can be used to validate results, but a series of nested sampling-specific checks have also been proposed, see e.g Higson et al. [217]. We now review two methods that we will make use of throughout the work presented in this thesis.

Evidence uncertainty

Nested sampling provides an uncertainty estimate for the estimated log-evidence. For a given model, one can compare the estimated uncertainty given by eq. (2.34) with the observed distribution of log-evidences for repeated analyses of the same model. If the sampling is unbiased, then the estimated uncertainty should agree with the observed distribution. This is particularly useful in cases where the model being analysed has an analytic evidence Z (eq. (2.11)) and information content $\mathcal{H}$ (eq. (2.33)).

Order statistic checks

Fowlie et al. [218] proposed a nested sampling cross-check based on order statistics. They consider the *insertion index*, the index at which a new element is inserted into a list to preserve the ordering, of a new live point in a nested sampling run with respect to the existing live points sorted by enclosed prior volume. Since the relationship between prior volume and likelihood is monotonic, this is the same as sorting by likelihood.

In a nested sampling run, if the new live points are in fact drawn from the likelihood constrained prior then, this should result in them being uniformly distributed in prior volume. Given the relationship between prior volume and likelihood, the discrete insertion index of the new points should therefore be uniformly distributed between 0 and $N_{\text{live}} - 1$.

A statistical test can then be performed on the insertion indices to quantify if they deviate from the expected uniform distribution or not. Fowlie et al. use the Kolmogorov-Smirnov test (KS test) [219, 220] and compute a corresponding p-value which, since the insertion index is discrete, will be a conservative estimate [221].

Fowlie et al. also propose computing a rolling p-value by breaking the nested sampling run into various chunks, finding the smallest p-value, and then adjusting the p-value for multiple testing. They suggest using chunks of size N_{live} since the prior volume should decrease by e every N_{live} iterations.

In practice, the only modification required to implement this check is to record the insertion index of each new live point. The final and rolling p-values can then be computed using a KS test.

2.7 Simulation based inference

Complex simulations have been developed for many different science domains, however, these simulations are not always suited to the previously described techniques for Bayesian inference. For example, whilst we may be able to simulate the data, the likelihood may be intractable or prohibitively expensive when considering the typical number of evaluations required. In such cases, simulation based inference can be employed to circumvent computing the likelihood [222]. In broad terms, methods for simulation based inference can be classified into *classical* methods and machine learning-based methods, which we will now briefly review.

Classical methods fall into two categories: approximate Bayesian computation (ABC) and methods that create a proxy for the likelihood. ABC compares the simulated and observed data using a distance measure. Data is simulated with parameters $\boldsymbol{\theta}$ and if it is sufficiently close to the observed data, the parameters are kept as a posterior sample. This is then repeated until a set of posterior samples is obtained. ABC's effectiveness hinges on the distance measure used and the tolerance used to determine if a simulation should be accepted or rejected. For a more thorough review of ABC see Sisson et al. [223]. The second method relies on creating a surrogate for the likelihood (or posterior) by estimating the distribution of the simulated data. Unlike ABC, this method is amortized; once the distribution has been approximated, it can be reevaluated for different observed data.

Machine learning-based methods leverage advances in machine learning to improve the classical methods, in most cases by using it to create surrogates for a target distribution. These replace the traditional methods, such as kernel density estimates or histograms, and are more flexible and have better scaling to large numbers of dimensions. They can also easily leverage other advances, such as active learning to further improve inference and tackle more expensive simulations [222]. A wide range of machine learning algorithms can be employed for simulation-based inference, these include density estimation algorithms, such as normalizing flows, and other generative algorithms, such as generative adversarial networks (GANs) and variational autoencoders (VAEs). More details about machine learning and these algorithms is provided in chapter 3.

2.8 Variational inference

Variational inference, like simulation based inference (section 2.7), is a technique that can be applied to Bayesian inference that does not rely on stochastic sampling, it is sometimes referred to as *Variational Bayes*. It frames the estimation of the posterior as an optimization problem over parameters $\boldsymbol{\phi}$ that parameterize a distribution $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ such that the distribution approximates the true posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{d})$ [224]. The parameters are optimized based on a distance measure D, such as the KLD described in section 2.2.2,

$$\boldsymbol{\phi} = \text{arg min } D(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{d})), \tag{2.42}$$

or the reverse measure.

Variational inference is closely related to simulated-based inference; it can be applied to simulation-based inference [225], but not all applications of variational inference are simulation-based inference [226]. However, similarly to simulation-based inference, variational inference has developed significantly thanks to advances in machine learning, such as VAEs [227] and normalizing flows [226].

2.9 Validating results from Bayesian inference

When performing Bayesian inference, we often need to validate the results that have been obtained by one of the previously mentioned methods. For example, before applying a method to real data, we might verify the method using simulated data that is well understood, or we may want to compare results obtained using two different methods.

2.9.1 Probability-probability plots

Probability-probability plots (P-P plots) are a visual method for assessing the similarity of two probability distributions by plotting their cumulative distributions against each other [228]. If the distributions are similar, the resulting line will be close to a diagonal. They are similar to quantile-quantile (QQ) plots, where the quantiles of two distributions are plotted against each other. The similarity between the distributions can also be quantified by computing the distance between the two cumulative distributions using, for example, the KS test [219, 220].

A common use for P-P plots is for comparing an empirical distribution to a theoretical distribution. They are particularly useful for validating methods for producing posteriors distributions, since they can help identify overarching issues with the method, such as under- or over-constraining [229, 230]. For a given model, this done by simulating N fiducial data $\boldsymbol{d}$ with parameters $\boldsymbol{\theta}$ drawn from a prior $\pi(\boldsymbol{\theta})$ and then producing posterior distributions for each data. For each posterior distribution, we then determine the fraction of the posterior probability ρ below or above the true value[2]. In this case, the theoretical distribution is uniform and we can produce a P-P plot using the observed distribution of ρ.

Four example P-P plots are shown in fig. 2.2; each is generated with 100 simulated posterior Gaussian distributions, each of which has 5000 independent posterior samples

[2] The choice of below or above will mirror the P-P plot.

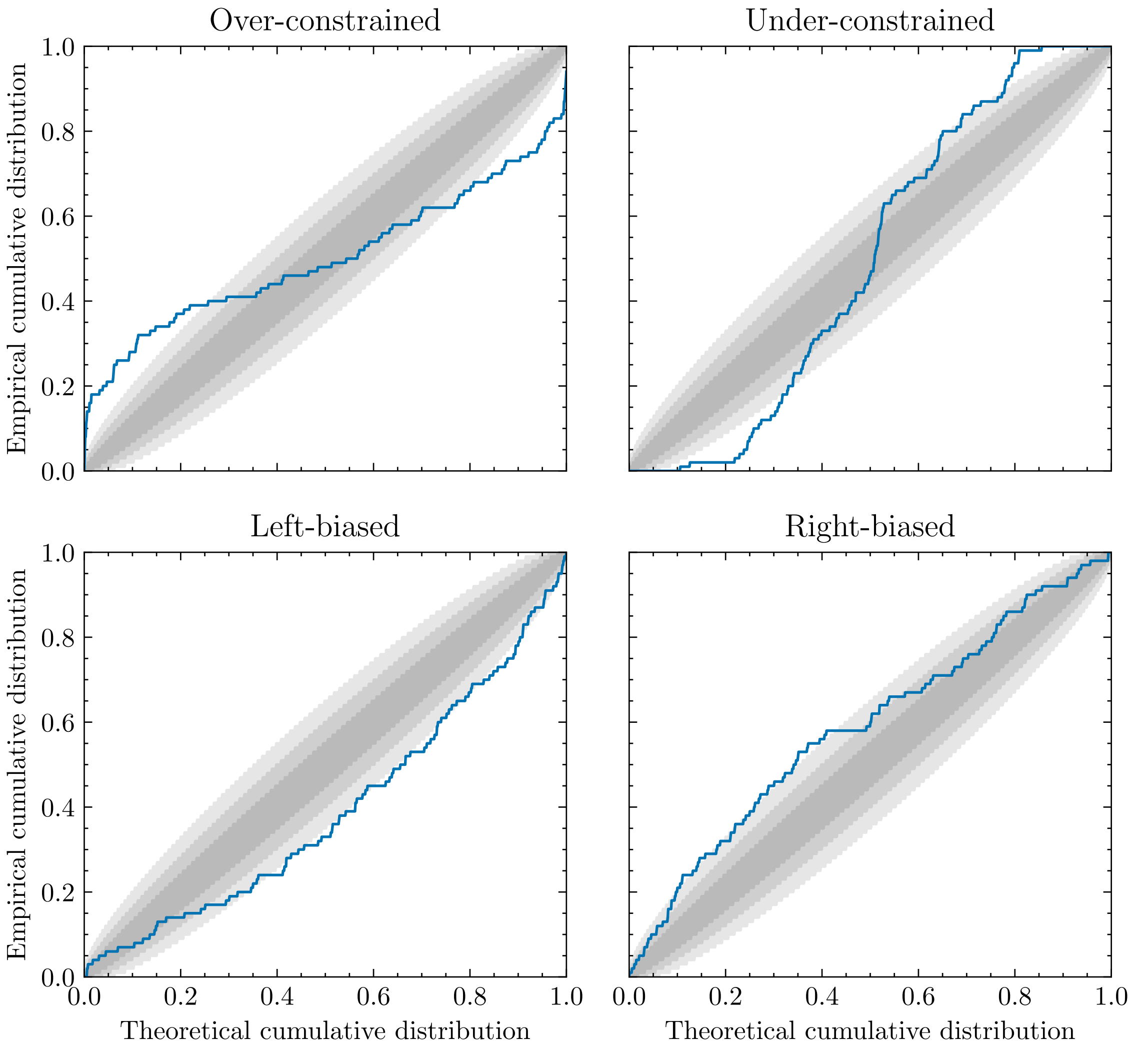


Figure 2.2: Example P-P plots that show the four most common issues that can be identified using a P-P plot. The plots were produced for a set of 100 posterior distributions each with 5000 posteriors samples using the fraction of the posterior probability ρ below the true value. The 1-, 2- and 3-σ cumulative confidence intervals are shown with shaded regions.

and has had either the mean or the standard deviation adjusted to be biased. These highlight the four most common problems that can be identified using a P-P plot:

- **Over-constrained**: the posterior distributions are over-constrained, i.e. narrower than expected because the standard deviation of the samples is, on average, smaller than expected,
- **Under-constrained**: the posterior distributions are under-constrained, i.e. broader than expected because the standard deviation of the samples is, on average, larger than expected,
- **Left-biased**: the posterior distributions are consistently shifted towards the left of the true value, i.e. because the mean of the samples is, on average, biased towards lower values than expected,
- **Right-biased**: the posterior distributions are consistently shifted towards the right of the true value, i.e. because the mean of the samples is, on average, biased towards lower values than expected.

The P-P plots in fig. 2.2 also include the 1-, 2- and 3-σ cumulative confidence intervals which depend on the total number of data being analysed.

In cases where the posterior distribution is n-dimensional a P-P plot with multiple lines, one for each parameter, can be produced. Additionally, since a P-P plot shows the comparison between two distributions, one can also compute a p-value for each of the parameters included in the plot using, for example, the KS test. An overall value can also be computed by combining the individual p-values, for example using Fisher's method [231]. This provides a simple method to quantify whether a P-P plot *passes* or *fails* based on a p-value threshold, typically 0.05 [229], but it does not provide information on how it failed, e.g. by being over-constrained. This overall test is often referred to as a P-P test.

One important caveat to note when using P-P plots is that if there are more complex biases or correlations that lead to incorrect posterior distributions, these can be missed, especially if the number of data-points is low. Therefore, P-P plots can be a useful method for identifying significant biases but a passing P-P plot is not always an indication that the obtained posterior distribution is unbiased.

2.9.2 Comparing posterior distributions

The need to compare posterior distributions often arises when comparing different sampling methods. If the posterior probability can be evaluated directly, then distance measures such as the KLD and JSD can be used. However, in practice, we typically only have samples from the posterior distributions. Various methods have been proposed for computing KLDs from n-dimensional samples [232], however these are not always reliable.

Instead, it has become standard practice in gravitational-wave data analysis to compute the JSD for the one–dimensional marginal posterior distributions and determine a threshold below which two sets of posterior samples can be considered statistically indistinguishable [122, 233–236]. Romero-Shaw et al. [122] proposed maximizing the JSD over all dimensions and established a threshold of 2 mbits. Ashton and Talbot [233] expanded this and introduced a new threshold based on the effective number of samples

$$\text{maximum JSD} \leq \frac{10}{n_{\text{effective}}}. \tag{2.43}$$

The authors compare this threshold to the 2 mbits threshold and find that the latter is valid for samples sizes $\mathcal{O}(10^3)$ but breaks down for more samples, whereas the proposed threshold is a conservative bound for up to 5×10^4 samples [233].

2.10 Bayesian inference for gravitational-wave data analysis

Having introduced Bayesian inference and the related techniques, we now revisit gravitational-wave data analysis. Previously, we described how the noise in a detector and signals are modelled and now we introduce the framework for analysing such data. We start by defining the gravitational-wave likelihood that is typically used to analyse signals from compact binary coalescences (CBCs) and then discuss how it is used for detection and parameter estimation, with a particular focus on the latter.

Throughout this section, we will consider a time series of duration T, sampled at a frequency f_s such that the number of discrete samples will be $N_s = T f_\text{s}$ and the time interval between samples is $\Delta t = 1/f_\text{s}$. We follow Veitch and Vecchio [237] and, to remain consistent, use the following conventions for the Fourier Transform:

$$\tilde{a}(f) = \int_{-\infty}^{+\infty} a(t) \exp(-2\pi i f t) \mathrm{d}t, \tag{2.44}$$

and its inverse

$$a(t) = \int_{-\infty}^{+\infty} \tilde{a}(f) \exp(2\pi i f t) \mathrm{d}f, \tag{2.45}$$

such that the discrete samples in either time or frequency are defined as

$$a(t_j) = a(j\Delta t) = a_j, \tag{2.46a}$$

$$\tilde{a}(f_k) = \tilde{a}(k/T) = \Delta t \tilde{a}_k, \tag{2.46b}$$

with

$$\tilde{a}_k = \sum_j a_j \exp(-2\pi i jk/N_s), \tag{2.47a}$$

$$a_j = \frac{1}{N_s}\sum_j \tilde{a}_k \exp(2\pi i jk/N_s). \tag{2.47b}$$

We will also use the shorthand notation $\boldsymbol{a} = \{a_j\}_{j=1}^{N_s}$ to refer to the vector of discrete samples, i.e. $\boldsymbol{d}$ for the data, $\boldsymbol{n}$ for noise and $\boldsymbol{h}_{\boldsymbol{\theta}}$ for the signal with parameters $\boldsymbol{\theta}$.

2.10.1 The gravitational-wave likelihood

We start by considering the noise-only hypothesis H_n. We assume the noise is stationary Gaussian coloured noise with zero mean and power spectral density (PSD) $S_n(f)$, so the likelihood will be a multivariate Gaussian distribution with the PDF given by

$$p(\boldsymbol{n}_0|H_n, \boldsymbol{C}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{C})}} \exp\left\{-\frac{1}{2}\boldsymbol{n}_0^T \cdot \boldsymbol{C}^{-1} \cdot \boldsymbol{n}_0\right\}, \tag{2.48}$$

where $\boldsymbol{C}$ is a covariance matrix that describes the correlations in the noise and $\boldsymbol{n}_0$ denotes the specific realization of Gaussian noise that is being analysed, hence $p(\boldsymbol{n}_0|H_n, \boldsymbol{C})$ is the probability of that specific realization. However, if we transform to the Fourier domain, the stationary noise will have a diagonal covariance matrix given by $\tilde{\boldsymbol{C}}_{ij} = \delta_{ij} S_n(f_i)$ [238]. So, if we define the noise-weighted inner product as [237]

$$\langle a|b\rangle = 2\int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f) + \tilde{a}^*(f)\tilde{b}(f)}{S_n(f)} \mathrm{d}f, \tag{2.49a}$$

$$\approx \frac{2}{T}\sum_{k>0} \frac{\tilde{a}(f_k)\tilde{b}^*(f_k) + \tilde{a}^*(f_k)\tilde{b}(f_k)}{S_n(f_k)}, \tag{2.49b}$$

then the noise-only likelihood can be written as

$$p(\boldsymbol{n}_0|H_n, S_n(f)) \propto \exp\left\{-\frac{1}{2}\langle \boldsymbol{n}_0|\boldsymbol{n}_0\rangle\right\}. \tag{2.50}$$

In practice, for the analysis of CBC signals, we will use this formulation of the likelihood instead of eq. (2.48), in part because it requires inverting the covariance matrix $\boldsymbol{C}$.

We then consider the hypothesis H_s that the data includes a gravitational-wave signal $\boldsymbol{h}_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ in a specific noise realization $\boldsymbol{n}_0$, and is therefore modelled as $\boldsymbol{d} = \boldsymbol{h}_{\boldsymbol{\theta}} + \boldsymbol{n}_0$. So we can express the noise as $\boldsymbol{n}_0 = \boldsymbol{d} - \boldsymbol{h}_{\boldsymbol{\theta}}$ and rewrite eq. (2.50) as

$$p(\boldsymbol{d}|H_s, S_n(f), \boldsymbol{\theta}) \propto \exp\left\{-\frac{1}{2}\langle \boldsymbol{d} - \boldsymbol{h}_{\boldsymbol{\theta}}|\boldsymbol{d} - \boldsymbol{h}_{\boldsymbol{\theta}}\rangle\right\}. \tag{2.51}$$

This is the gravitational-wave likelihood for a signal in a single detector with Gaussian noise described by a PSD $S_n(f)$. This can also be re-written as [239]

$$p(\boldsymbol{d}|H_s, S_n(f), \boldsymbol{\theta}) \propto \exp\left\{\langle \boldsymbol{d}|\boldsymbol{h_\theta}\rangle - \frac{1}{2}\langle \boldsymbol{h_\theta}|\boldsymbol{h_\theta}\rangle - \frac{1}{2}\langle \boldsymbol{d}|\boldsymbol{d}\rangle\right\}, \tag{2.52}$$

where the final term $\langle \boldsymbol{d}|\boldsymbol{d}\rangle$ is constant for given data $\boldsymbol{d}$.

This formalism can be easily extended to multiple detector if we assumed the noise to be independent in each detector. For N detectors, the likelihood is then the product of N multivariate Gaussian distributions

$$p(\boldsymbol{d}_{\{H,L,V\}}|H_s, S_n^{\{H,L,V\}}(f), \boldsymbol{\theta}) = \prod_{k\in\{H,L,V\}} p(\boldsymbol{d}_k|H_s, S_n^k(f), \boldsymbol{\theta}). \tag{2.53}$$

This likelihood has been used extensively to perform parameter estimation for CBC signals [96, 122, 198]. This is discussed further in section 2.10.3 but first we briefly discuss how eq. (2.51) can be used to derive a detection statistic for gravitational-wave signals.

2.10.2 Gravitational-wave detection

Thus far, we have focused on the Bayesian inference in the context of parameter estimation, e.g. via nested sampling, however, we also use a Bayesian framework for detecting gravitational waves. In detection, we aim to determine between two hypotheses: the null-hypothesis H_n that the data does not contain a gravitational wave, and the hypothesis H_s that the data contains noise and a gravitational-wave signal. This is therefore a problem in model selection.

We can write the Bayes factor between the two hypothesis following section 2.3.1 as

$$\begin{aligned}\mathcal{B}_{s/n} = \frac{p(\boldsymbol{d}|H_s, S_n(f), \boldsymbol{\theta})}{p(\boldsymbol{d}|H_n, S_n(f))} &= \frac{\exp\left\{\langle \boldsymbol{d}|\boldsymbol{h_\theta}\rangle - \frac{1}{2}\langle \boldsymbol{h_\theta}|\boldsymbol{h_\theta}\rangle - \frac{1}{2}\langle \boldsymbol{d}|\boldsymbol{d}\rangle\right\}}{\exp\left\{-\frac{1}{2}\langle \boldsymbol{d}|\boldsymbol{d}\rangle\right\}} \\ &= \exp\left\{\langle \boldsymbol{d}|\boldsymbol{h_\theta}\rangle - \frac{1}{2}\langle \boldsymbol{h_\theta}|\boldsymbol{h_\theta}\rangle\right\},\end{aligned} \tag{2.54}$$

where the constants of proportionality have cancelled. For detection, we are not necessarily interested in the values of the parameters $\boldsymbol{\theta}$ and would rather marginalize over $\boldsymbol{\theta}$ to obtain $p(\boldsymbol{d}|H_s, S_n(f))/p(\boldsymbol{d}|H_n, S_n(f))$. However, as discussed in the context of nested sampling (recall that the evidence is the likelihood marginalized over $\boldsymbol{\theta}$), marginalizing over the parameter space can be challenging and computationally expensive. Instead, the Bayes factor can be approximated by computing the maximum likelihood statistic for a discrete set of N parameter values $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N\}$ [12]. In the context of gravitational-wave detection, this corresponds to constructing a template bank of N possible waveforms and computing a detection statistic over all waveforms. It can be shown that, from eq. (2.54), the optimal detection statistic for a fixed $\boldsymbol{\theta}$ and Gaussian noise is [12, 239]

$$\langle \boldsymbol{d} | \boldsymbol{h}_{\boldsymbol{\theta}} \rangle = 2 \int_0^\infty \frac{\tilde{\boldsymbol{d}}(f) \tilde{\boldsymbol{h}_{\boldsymbol{\theta}}}^*(f) + \boldsymbol{d}^*(f) \tilde{\boldsymbol{h}_{\boldsymbol{\theta}}}(f)}{S(f)} \mathrm{d}f, \tag{2.55}$$

which is known as the *matched filter*. The maximum likelihood statistic for a set of N parameter values, known as the matched filter signal-to-noise ratio (SNR), is then defined as [12]

$$\rho^2 = \frac{\langle \boldsymbol{d} | \boldsymbol{h}_{\boldsymbol{\theta}} \rangle}{\sqrt{\langle \boldsymbol{h}_{\boldsymbol{\theta}} | \boldsymbol{h}_{\boldsymbol{\theta}} \rangle}}, \tag{2.56}$$

where $\langle \boldsymbol{h}_{\boldsymbol{\theta}} | \boldsymbol{h}_{\boldsymbol{\theta}} \rangle^{1/2}$ is often known as the *optimal (matched filter)* SNR.

2.10.3 Gravitational-wave parameter estimation

Gravitational-wave parameter estimation is typically carried out in a Bayesian framework using the likelihood as defined in eq. (2.51), or some variant of it. We must therefore define a signal model H_s, PSD $S_n(f)$ and prior distributions over the model parameters $p(\boldsymbol{\theta}|H_s)$ in order to be able to perform inference with the likelihood.

Parameters

The model H_s will include one of the waveform approximants discussed in section 1.4.2. This will model certain physics and determine which parameters we will perform inference on. Current analyses typically include 15 parameters for BBHs and 17 for binary neutron stars (BNSs), though as discussed previously, additional parameters are often added to, for example, probe additional physics [129, 130] or study possible deviations from general relativity [160, 162].

The waveform approximant will be defined in terms of specific parameters, however it is common to use a different parameter space for inference. This reparameterization is included because the parameter space can be highly correlated and multimodal, and certain reparameterizations can make inference more efficient [96]. Common reparameterizations include sampling in chirp mass and mass ratio instead of component masses [96], using the system-frame parameters [123] to describe the spins and employing different angles for the sky location as discussed in section 1.4.2. In spite of these reparameterizations, the parameter space is still highly complex. Figure 2.3 shows an example of the correlations between mass ratio and chirp mass for an example CBC signal.

Estimating the noise

The noise present in the detector must be modelled via the PSD $S_n(f)$. For analyses that use simulated data, a PSD that is known a priori can be used [74]. However, whilst the

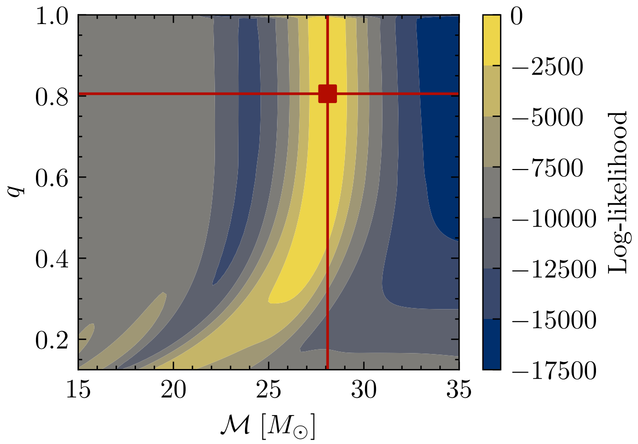


Figure 2.3: The gravitational-wave log-likelihood in a zero-noise case as a function of chirp mass $\mathcal{M}$ and mass ratio q for a GW150914-like signal (see appendix C.1 for the exact parameters) in a two-detector network. The crosshairs show the parameter values for the injected signal.

PSD is assumed to be stationary over the timescales analysed for CBC signals ($\mathcal{O}(1-10)$ seconds for BBHs and $\mathcal{O}(100)$ seconds for BNSs), in practice it will vary over the course of an observing run, so it must be estimated for each event being analysed. Different methods can be employed for estimating the PSD and this is typically done using the data which is adjacent in time to the data being analysed, e.g. up to 1024 s before the time of event being analysed [96]. The most basic method, known as Welch's method [238], divides the adjacent data into separate non-overlapping time segments, applies a Tukey window, computes an estimate of the PSD for each segment using the discrete Fourier transform, and then estimates a final PSD using the median power in each frequency bin [94, 96, 198]. Alternatively, `BayesWave` [95] can be used to fit a parameterized model to the PSD using MCMC and obtain a posterior distribution over the model parameters [240]. This method has been shown to be more robust than the basic method described previously and has been used extensively [9, 10].

In some cases, there may be one or more glitches surrounding and/or overlapping a signal. This presents a challenge as if the glitches are not accounted for, they may introduce biases in the parameter estimation results [241–244], such as incorrect estimates of the chirp mass or luminosity distance due to the excess power [241]. Depending on the type of glitch [245], different methods may be employed to mitigate its effect, these include excluding certain frequencies from analyses or using Bayesian inference to model the glitch and then subtract it [95, 246, 247]. These techniques have been applied to GW170817 [29, 246] and various events in the third LIGO-Virgo observing run [9, 10, 248].

Priors

The results obtained using Bayesian inference depend on the choice of priors [249]. We now review common choices of priors used for analysing signals from CBCs in current ground-based interferometers.

We start with the component masses: most analyses use a joint prior on the components masses that is uniform in $m_1 - m_2$ subject to the constraint $m_1 \geq m_2$, where the limits on m_1 and m_2 depend on prior assumptions about the type of source being analysed. These limits are informed by the initial estimates returned by detection pipelines [238] and the length of the data segment being analysed. This is not strictly correct in a Bayesian sense since the priors are therefore conditional on the observed $\boldsymbol{d}$ rather than representing our state of knowledge prior to analysing the data [178]. However, assuming the priors are sufficiently broad, it should not significantly bias the recovered parameters but will impact the evidence estimate. Since sampling normally occurs in the chirp mass-mass ratio space, these priors are converted to corresponding priors in this parameter space: $p(\mathcal{M}, q|H_s) \propto \mathcal{M} m_1^{-2}$ [96, 198]. Alternatively, the prior can be defined to be uniform in the chirp mass-mass ratio space with constraints on the component masses, e.g. in Romero-Shaw et al. [122], and we will do so in chapters 4 to 7.

The priors over the angles that describe the sky location of the source are typically isotropic, the exact definition will depend on the convention being used, but for right ascension and declination these are $p(\alpha, \delta|H_s) \propto \cos(\delta)$. The prior on the luminosity distance will depend on the distance scales being considered, some analyses have employed a distance-squared prior $p(d_\mathrm{L}|H_s) \propto d_\mathrm{L}^2$ [7, 96] but more recent analysis have used a prior that is uniform in comoving volume $p(z|H_s) \propto \mathrm{d}V_c/\mathrm{d}z$ or uniform in comoving volume and source frame time $p(z|H_s) \propto 1/(1+z)\mathrm{d}V_c/\mathrm{d}z$ [122]. These priors are comparable at low distances but differ greatly at larger distances [122].

The prior over the orientation of the binary as described by $(\theta_\mathrm{JN}, \psi, \varphi_c)$ is normally taken to be isotropic $p(\theta_\mathrm{JN}, \psi, \varphi_c|H_s) \propto \sin\theta_\mathrm{JN}$. When spins are included, the priors are uniform over the spin magnitudes $a_{\{1,2\}}$ on $[0, 1]$, and isotropic over the spin angles $p(\theta_1, \theta_2, \phi_{12}, \phi_\mathrm{JL}|H_s) \propto \sin\theta_1 \sin\theta_2$. If the spins are assumed to be aligned, then we only consider two parameters: the z-components of the spin vectors $\chi_i \equiv s_{iz}$, which are defined on $[-1, 1]$ [96, 250].

The prior over the coalescence time is informed by the trigger time returned by one of the detection pipelines [238]. Typically, the prior is uniform over a 200 ms window around the trigger time [96, 198, 251] which accounts for any uncertainties in the times provided by the detection pipelines. Once again, this not strictly Bayesian but should not impact the estimated coalescence time so long as the prior is sufficiently broad, though it will change the evidence.

Marginalization

The gravitational-wave likelihood eq. (2.51) can also be written such that certain parameters are marginalized over rather than being sampled. This reduces the number of parameters that are sampled which can accelerate inference. Furthermore, the posterior distributions for these parameters can be reconstructed post-sampling [251]. Currently, three main marginalizations are used for gravitational-wave parameter estimation [251]:

- **Phase marginalization:** marginalizes over the phase of coalescence φ_c by considering only the 2-2 modes of the gravitational-wave signal [96, 252]. This is only valid for waveform approximants that do not include higher-order-modes or where the effect of precession is not measurable, which we discuss in chapter 6,
- **Distance marginalization:** marginalizes over the luminosity distances d_{L} by leveraging $\bar{h}^{ij} \propto d_{\mathrm{L}}^{-1}$ (see eq. (1.14)) which allows for the likelihood to be integrated numerically [253, 254]. This marginalization is valid for all gravitational-wave signals, however, care must be taken if considering distances at cosmological scales due to cosmological redshift and its effect on the observed mass [251],
- **Time marginalization:** marginalizes over coalescence time t_{c} using a reference coalescence time, discrete Fourier transforms and a discretized coalescence time [255]. This marginalization is known to breakdown for high SNR and long-duration signals [251].

All three marginalizations, or combinations of them, can be use simultaneously, however care must be taken to ensure the correct order is used, see, for example, Thrane and Talbot [251, Appendix C.5].

Modifications to the likelihood

The cost of evaluating the likelihood, as defined in eq. (2.51), scales with the length of the observation T and sampling frequency f_{s}. This means that performing parameter estimation for long-duration signals can be costly. There exist various modifications to the likelihood that aim to reduce its computational cost.

Reduced order models (ROMs) build computationally efficient surrogates that approximate a given waveform, therefore reducing the cost of evaluating the likelihood. These methods are typically based on either singular value decomposition (SVD) [256, 257] or a greedy reduced basis method and empirical interpolation [258]. Reduced-Order-Quadrature (ROQ) based techniques [259–261] use the latter method and have been widely applied to CBC analyses [9, 262] where they can provide speed-ups of order $\mathcal{O}(10^2 - 10^4)$ when applied to BNS signals [259, 260, 263]. More recently, machine learning has also been applied to constructing surrogates; we discuss this, after having introduced machine learning, in section 3.4.

Alternatively, methods based on heterodyning [264, 265] and relative binning [266–268] make use of knowledge about the signal being analysed to reduce the cost of computing subsequent waveforms. These use the template from a detection pipeline as a starting point, which is then used to approximate the waveform during inference. These techniques are best suited to long signals, where the speed-up can be up to $\mathcal{O}(10^4)$ [267].

Finally, there are techniques that use multi-banding to leverage the chirp-like nature of signals from CBC [263, 269–271]. This enables the use of different sampling frequencies in the time domain; since during the inspiral a lower sampling frequency is need than during the merger. When applied to BNS signals, this technique can provide speed-ups of order $\mathcal{O}(10)$ times [263] without the need to precompute bases or have prior knowledge about the signal being analysed.

Implementations

Various implementations exist for performing gravitational-wave parameter estimation, some employ stochastic sampling techniques, such as MCMC and nested sampling, and others employ techniques based on ABC and simulation based inference. However, they all aim to address the same challenges: computing the likelihood as defined in eq. (2.51) can be computationally expensive, mostly due to the cost of computing the waveform; and the likelihood surface is high-dimensional[3], multimodal and, in cases, highly correlated, which can make it hard to sample or approximate.

`LALInference` The first widely-used stochastic-sampling package for analysing gravitational-wave signals from CBC was `LALInference` [96]. It implements both an MCMC sampler and a nested sampler, both of which include specific modifications to handle the complex nature of the gravitational-wave likelihood space [96], and builds upon a wide range of existing work [237, 272–282]. It was used to analyse all the events detected up to and including Gravitational-Wave Transient Catalog 2 (GWTC-2) [6–8, 29] and it is still used for validating other samplers [10, 122].

`bilby` More recently, `bilby` has been employed instead of `LALInference`. Unlike its predecessor, it uses off-the-shelf nested samplers and MCMC samplers. It includes support for `bilby_mcmc` [233], `dynesty` [197], `emcee` [283], `kombine` [284], `multinest` [208, 212], `polychord` [206], `ultranest` [207] and more. It also supports `nessai` [1], which we introduce in chapter 4. However, the default nested sampler is `dynesty` [197] with several custom proposal methods based on the methods implemented in `LALInference`

[3]

[122, 198]. It was used in Gravitational-Wave Transient Catalog 2.1 (GWTC-2.1) and Gravitational-Wave Transient Catalog 3 (GWTC-3) [9, 10].

Related to `bilby` is `parallel_bilby` [285], which is a highly parallelized version of `bilby` and designed for use with high-throughput computing cluster. It is uses `dynesty` and parallelizes the proposal process in the nested sampling algorithm. It has been predominantly used for expensive analyses such as performing inference for BNS and neutron star-black hole (NSBH) signals [10, 30].

`RIFT` Unlike the implementations mentioned so far, `RIFT` leverages Gaussian processes and stochastic sampling techniques [250, 286]. It uses an iterative procedure that alternates between proposing posterior samples and then improving upon them. Various improvements have been made since it was initially introduced [287–291] and it has been used in GWTC-2.1 [9] and GWTC-3 [10] to analyse signals using waveform approximants that are too computationally expensive for use with standard sampling methods, such as `SEOBNRv4PHM` [116].

Other implementations Besides `LALInference` and `bilby`, other implementations that use stochastic sampling have been developed and applied to gravitational-wave inference. These include `PyCBC Inference` [25, 292], which has been used to produce four *Open Gravitational-wave Catalogues* [293–296], and `bajes` [297], which is specifically designed for analysing multi-messenger transients. Similarly to `bilby`, both these implementations support for various off-the-shelf samplers.

There are also implementations that do not rely on standard sampling techniques, `simple-pe` [298] uses simple arguments and understanding of the underlying physics to produce approximate posteriors in times of order minutes, and the non-Markovian sampler `VARAHA` [299] iteratively discards regions of the parameter space whilst focusing on regions of high-likelihood. The latter, whilst not framed as a nested sampling algorithm, has significant parallels to `i-nessai` [2] which we present in chapter 7.

Simulation-based inference More recently, simulation-based inference has been applied to gravitational-wave inference. This technique leverages advances in machine learning to accelerate inference and, in some cases, enable analyses that are not possible with standard sampling-based techniques since it does not require evaluating the likelihood. We will discuss these methods in more details in chapter 3 where we introduce machine learning and discuss its applications to gravitational-wave data analysis.

2.11 Summary

In this chapter, we have introduced the fundamentals of probability theory and Bayesian inference. Our focus is on Bayes' Theorem, eq. (2.10), with allows us to relate the probability of a hypothesis given some data (the posterior probability) to the probability we observed the data given the hypothesis was true (the likelihood) and is the foundation of Bayesian inference. Our aim is therefore to estimate the posterior distribution, or draw samples from it, and, is some cases, estimate the normalization constant known as the evidence. However, doing so is not trivial since we typically cannot sample the posterior directly and estimating the evidence requires computing an n-dimensional integral. Different techniques exist to address this challenge, such as simulation based inference and variational inference (see sections 2.7 and 2.8), however our focus in this work is on stochastic sampling algorithms, specifically nested sampling.

Nested sampling is designed to compute the evidence integral by recasting it as a one-dimensional problem in terms of the prior volume, we review the details in section 2.6. The algorithm considers the fraction of the prior (volume) contained with contours of equal likelihood, starting with the lowest likelihood which contains the entire prior and then shrinking to higher likelihoods that contain progressively smaller fractions of the prior. The prior volume is determined by estimating the shrinkage between subsequent likelihood levels using a set of live points. At a given iteration the point with the lowest likelihood is discarded, thus shrinking the prior volume, and the number of point remaining is used to estimate the shrinkage. A replacement point is then drawn within the new likelihood contour and next iterations begins. The crux of the algorithm is drawing new samples with the likelihood contour, since they must be i.i.d. and distributed according to the prior. This, alongside computing the likelihood, is typically the main bottleneck in nested sampling algorithms. We review different methods and implementations in section 2.6.5.

We then shift our focus to Bayesian inference in the context of gravitational-wave data analysis, where nested sampling is the standard algorithm for performing inference. We define the gravitational-wave likelihood, eq. (2.51), which will be central to all the analyses presented in this thesis. We highlight its importance in gravitational-wave detection before concentrating on parameter estimation, specifically for CBC signals. We review the methods used to estimate the PSD, discuss the typical choices for priors for sampling and summarize variations modifications that can be made to the likelihood, such as marginalizations and methods to accelerate the likelihood, such as ROQs. Finally, we discuss the variance implementations that exist for performing gravitational-wave parameter estimation, including `bilby` which will be used throughout the work presented in later chapters.

This second introductory chapter provides further context for the work presented in this thesis; performing gravitational-wave parameter estimation typically relies upon the use of stochastic sampling algorithms, predominantly nested sampling, and such analyses are computationally expensive, taking order days to weeks [96, 122]. This is due to the cost of evaluating the likelihood coupled with the complexity of the parameter space, which makes the fundamental step in nested sampling from the likelihood-constrained prior more challenging and leads to a large number of likelihood evaluations. Developing new methods for sampling from the likelihood-constrained prior therefore has the potential to improve the efficiency of nested sampling algorithms and in turn reduce the cost of performing CBC parameter estimation.

Chapter 3

Machine Learning

This chapter introduces machine learning, highlighting different types and applications, before reviewing neural networks. The main focus is on normalizing flows, their formulation, construction and training, since these will be used throughout subsequent chapters. It concludes by reviewing applications of machine learning to gravitational-wave data analysis.

Section 3.1 provides an overview of machine learning and reviews different categories of machine learning algorithms based on how they are trained and how they are applied. The basics of neural networks are then introduced in section 3.2, including how they are trained and the different types that will be used in this work. Finally, section 3.4 reviews applications of machine learning to gravitational-wave data analysis for data quality, source modelling, searches and determining source properties. These fall into two broad categories: end-to-end machine learning approaches and approaches that augment existing techniques with machine learning. This provides the final piece of context for the work presented in this thesis and concludes the review of the existing literature.

3.1 An introduction to machine learning

Machine learning is the study of algorithms that are modified via an automatic process that is akin to 'learning'. This is achieved by designing and updating a model using training data, or *training* the model [300]. There are a vast number of algorithms that can fall under the umbrella of machine learning and it can therefore be useful to broadly categorize them.

Machine learning algorithms can be categorized based by how they are trained [300]:

- **Supervised learning:** in supervised learning, the dataset consists of the set of training data $\{x^{(i)}\}$ and corresponding labels (or classes) $\{y^{(i)}\}$. The aim is to train an algorithm to predict the labels given the training data such that, once it is trained, it can be applied to unlabelled data. An example of supervised learning would be learning to distinguish between pictures of cats and dogs, where a picture is an instance of the training data $x^{(i)}$ and the label $y^{(i)}$ is either 'cat' or 'dog'.

- **Unsupervised learning:** in unsupervised learning, the dataset consists of a series of unlabelled examples $\{x^{(i)}\}$. The aim is to train an algorithm that maps the dataset to a representation that can be used to solve a problem. In the context of pictures of cats and dogs, the pictures are once again the training data $\{x^{(i)}\}$, however, we do not provide labels, instead the objective is to have the algorithm identify the classes present in the data without being told they exist.

- **Semi-supervised learning:** in semi-supervised learning, the dataset contains a mix of labelled and unlabelled examples and the known labels are often 'unreliable'. The aim is the same as supervised learning but leveraging the additional information that the unlabelled data can provide. Considering again pictures of cats and dogs, the dataset is now a mix of pictures $\{x^{(i)}\}$ with corresponding labels and those without $\{y^{(i)}\}$. The unlabelled pictures provide additional information for an algorithm to learn from compared to using just the labelled data.

- **Reinforcement learning:** in reinforcement learning, rather than a dataset, a machine learning algorithm, often known as an agent, is trained using a reward-based system. The agent takes an action which results either in a reward and penalty; the overall aim is to maximize the cumulative reward. It can be thought of as learning via 'trial-and-error', such as by playing a game and learning based on the outcome of an action which, for example, could be quantified by a score.

In this thesis, we will focus on applications of unsupervised machine learning algorithms, therefore the following explanation will be aimed predominantly at their application. The information will be broadly applicable to supervised and reinforcement algorithms, but there may be subtle differences.

Machine learning algorithms can also be roughly classified on the type of task they can perform, however, this classification is somewhat arbitrary and algorithms can fall in multiple categories, so we highlight categories that are of note for gravitational-wave data analysis:

- **Classification:** classification algorithms are trained to predict a label (or labels) for input data such that it can be classified in to discrete classes [300]. Distinguishing between pictures of cats and dogs is a typical example of a classification task [301].

- **Regression:** regression algorithms are trained to predict a continuous variable (also known as a target) given some input data [300]. Predicting house prices is a common example of a regression task [302].

- **Generation:** generative algorithms are trained to generate outputs that resemble the training dataset [300, 303]. Generating images of human faces is a typical example of a generative task [304].

- **Anomaly/outlier detection:** anomaly detection algorithms are trained to identify inputs that deviate from the training dataset [305]. An example of anomaly detection is identifying changes in a monitoring system, e.g. fluctuations in a power grid [306].

- **Clustering:** clustering algorithms are designed to split a dataset in to similar groups, where the number of groups can be defined a priori or learnt from the data set [307]. For example, grouping customers with similar interests in recommendation systems [308].

Neural networks are amongst the most broadly application machine learning algorithms, thanks it part to the wide number of variants. They can be used to perform all the categories of task mentioned previously and they can also be trained using all four of the previously mentioned methods. We now review neural networks.

3.2 Neural networks

Neural networks are a subset of machine learning algorithms that have seen extensive use in a wide variety of fields. They were first conceived in the 1940s [309] and saw some development in the 80s [310] but advances in hardware which enable faster training [311, 312] led to a surge in their use in the last two decades. Many variations of neural networks have been developed and they can be applied to a wide range of problems in all areas of machine learning.

In its simplest form, a neural network is a function that maps from an input $\boldsymbol{x}$ to an output y:

$$y = f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\phi}), \tag{3.1}$$

parameterized by a set of trainable parameters $\boldsymbol{\phi}$ that are updated in the training process. The core building blocks of a neural network are nodes usually known as neurons, these apply a simple operation to an input $\boldsymbol{x}$ (usually a vector) and produce a scalar output y:

$$y = \boldsymbol{w} \cdot \boldsymbol{x} + b, \tag{3.2}$$

where $\boldsymbol{w}$ and b are the trainable parameters known as the weights and biases respectively. A single neural network, as the name implies, will be made up of multiple neurons which are organized in layers such that a layer f with m neurons takes an n-dimensional $\boldsymbol{x}$ and returns an m-dimensional vector output $\boldsymbol{y}$:

$$\boldsymbol{y} = f(x; W, \boldsymbol{b}) = \sigma\left(W\boldsymbol{x} + \boldsymbol{b}\right), \tag{3.3}$$

where the weights W are now an $m \times n$ matrix and the biases an m-dimensional vector $\boldsymbol{b}$. We have also included an activation function σ, this is typically a non-linear function that allows the layer to learn a more complex mapping from the inputs to the output. Its importance will become apparent when we consider multiple layers.

A layer is a vector function $f_i : \mathbb{R}^n \to \mathbb{R}^m$, e.g. as defined in eq. (3.3), that learns a non-linear mapping between an input and an output and is parameterized by parameters $\boldsymbol{\phi}_i$, in the case of eq. (3.3) $\boldsymbol{\phi}_i \equiv \{W, \boldsymbol{b}\}$. A neural network is a set of nested vector functions such that the output of the i-layer is the input to the $i+1$-layer, e.g. for three layers

$$f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\phi}) = f_3(f_2(f_1(\boldsymbol{x}; \boldsymbol{\phi}_1); \boldsymbol{\phi}_2); \boldsymbol{\phi}_3). \tag{3.4}$$

The importance of the non-linear activation function σ is now apparent, without it this three-layer neural network would reduce to a single weight matrix and bias vector. Activation functions are therefore key for a neural network to be able to represent more complex functions. Furthermore, it has been proven that a two-layer neural network is a universal function approximator, i.e. it can in theory approximate any possible continuous function, this is known as the Universal Approximation Theorem [313]. There are many different activation functions that are used, generally they are all continuously differentiable which allows gradient-based optimization, common examples include the hyperbolic tangent (tanh), sigmoid, Rectified Linear Unit (ReLU) [314] and LeakyReLU [315] which are shown in fig. 3.1.

This simple formulation of a neural network can be extended to included various modifications, some of which will be reviewed in section 3.2.3.

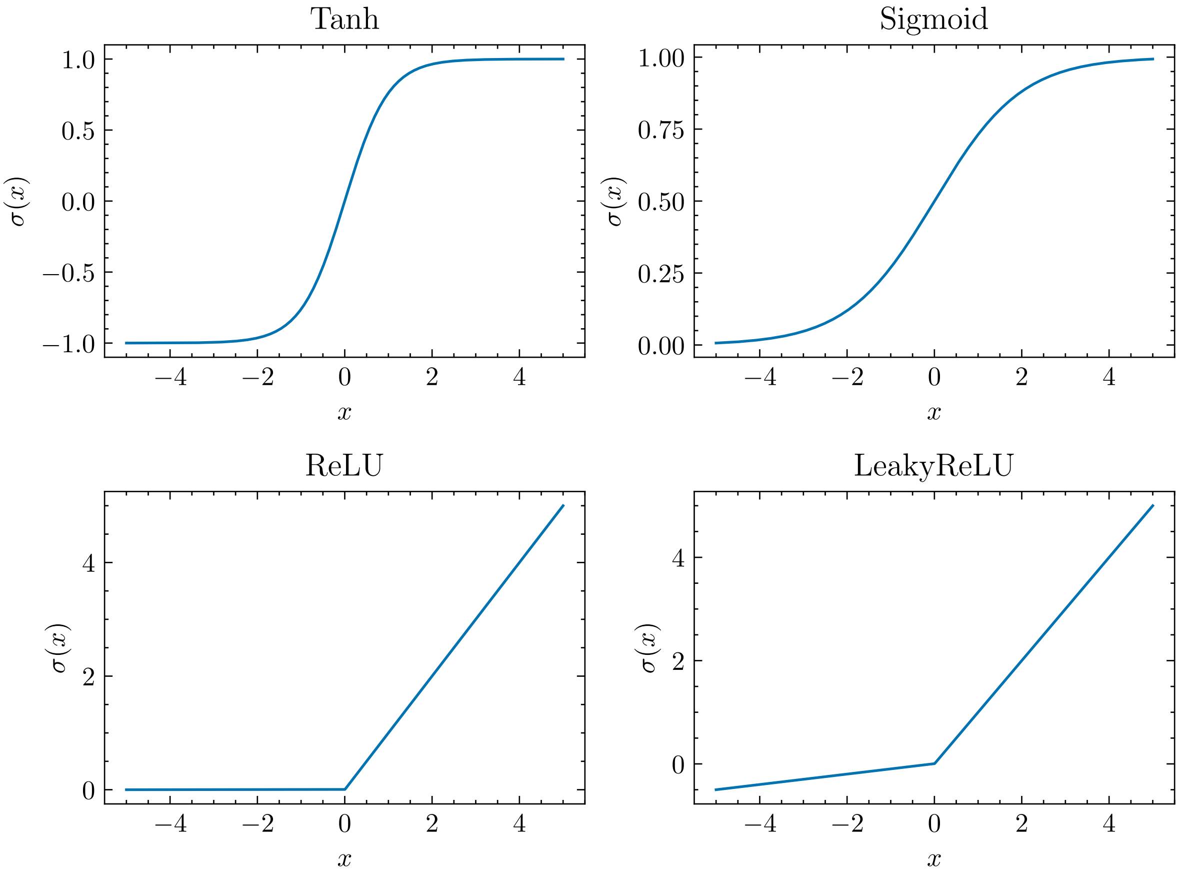


Figure 3.1: Examples of four commonly used activation functions: tanh, Sigmoid, ReLU and LeakyReLU with a slope of 0.01

3.2.1 Training

The final piece that is necessary to understand neural networks, and, in general, most machine learning algorithms, is the training process used to update the trainable parameters. This is an optimization problem and, as such, there are many different methods that can be used [316–318], however the most common approaches are based on gradient descent [319, 320].

We must first define a function which will be used in the optimization process. This will quantify the algorithm's performance given a set of inputs and is commonly referred to as a loss or cost function. We will denote the loss function $L(\boldsymbol{x}; \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ are the parameters of the neural network. The particular loss function will depend on the problem in question, but it should be a scalar function. Example loss functions include the mean squared error, categorical cross-entropy and Kullback-Leibler divergence (KLD).

The optimization process is typically iterative; at each iteration, the value of the loss function is computed for a set of inputs. We must then compute the gradients of the parameters with respect to the value of the loss function. This could be done directly with respect to each parameter in the network, but this would be inefficient. Instead, backpropagation [312, 321] is used to efficiently compute the gradients [320]. Once the gradient has been computed, gradient descent is used to take a step in the parameter space in a direction that reduces the loss and update the values of the weights. In practice, stochastic gradient descent is used instead of standard gradient descent, since the latter becomes prohibitively expensive for large datasets. Stochastic gradient descent uses a subset of the training data, known as a batch, to approximate the actual gradient. Given a batch of data of length n_{BS} (known as the batch size), the parameters at a given iteration i are updated following

$$\boldsymbol{\phi}_{i+1} = \boldsymbol{\phi}_i + \eta \nabla L(\boldsymbol{\phi}_i) = \boldsymbol{\phi}_i + \frac{\eta}{n_{\mathrm{BS}}} \sum_{j=1}^{n_{\mathrm{BS}}} \nabla L(\boldsymbol{x}_j; \boldsymbol{\phi}_i), \tag{3.5}$$

where η is the step size, usually known as the learning rate, and ∇L is the gradient of the loss function with respect to the parameters $\boldsymbol{\phi}$.

A typical training regime will involve shuffling the training data and dividing it into batches. We then iterate over the batches and for each batch compute the loss L and update the parameters of the network. A complete loop of over all the batches of training data is often referred to as an epoch.

What remains is to define a starting point for the optimization process, i.e. an initial set of parameters. This is known as *initializing* the weights of the neural network. The standard approach is to initialize the weights with random values that are close to, but not exactly, zero. However, it has been shown that choice of distribution can impact both the overall performance and time to converge [322] and various different methods have been proposed [323].

Equation (3.5) is the simplest version of stochastic gradient descent. There are various extensions and variants of stochastic gradient descent that have been used extensively for training neural networks. Notable examples include stochastic gradient descent with momentum [324] and adaptive moment estimation (Adam) [325]. It has also become standard practice to adjust the learning rate during training [326]; proposed methods include scheduling specific values of the learning rate [327], decaying the learning rate [328–330] and cyclic learning rates [329, 331].

3.2.2 Data and its importance

Designing and training a neural network is dependent on the data being used; the data can dictate the design of the network, how it is trained, and the validity and applicability of the final trained network.

Data pre-processing

Data pre-processing prior to training is crucial for training neural networks [332], it can improve the quality of the results and make training more stable. There are many ways to pre-process data and this can include augmenting available training data [333], but we do not cover that in this overview as it is typically problem specific.

Due to how the weights in neural networks are initialized and optimized, ensuring that data is zero-centred and all dimensions have variances close to one can greatly improve training [334]. It is therefore standard practice to scale data prior to training; commonly used methods included rescaling to the unit-interval and rescaling to have zero mean and unit standard deviation, known as standardization.

Splitting data

Given a dataset, it is common practice [335], particularly in supervised applications, to divide the data into three categories: training, validation and testing:

- **Training data:** is used to train the neural network and is often the largest of the three categories.

- **Validation data:** is not used to train the neural network, but instead is used to evaluate its performance as it trains.

- **Testing data:** is reserved until after the final model is trained. It serves to verify that the trained neural network generalizes to unseen data. This category of data is sometimes omitted [335].

This split is key to ensuring a neural network is properly trained and can generalize to unseen data. During training, this is done using the validation data. At a given iteration, the loss function is computed using the validation but the weights are not updated, giving the validation loss. The validation loss is tracked over the course of training. If the neural network is learning general properties of the training data, then the validation loss should decrease at a similar rate to the training loss. However, if the neural network is instead learning properties specific to the training data, or even memorizing the training data, then the network is overfitting and the validation loss will increase.

3.2.3 Types of neural networks

There are various different types of neural networks, some are named based on the type of layers they use and others based on how they are trained or their application. We will now highlight two types of neural networks that are relevant to work presented in this thesis.

Multi-layer perceptrons

Multi-layer perceptrons (MLPs) [310], also known as fully connected or dense neural networks, are the most basic type of neural network and they consist solely of layers as described in eq. (3.3). They are well suited to a wide variety of problems. Their main limitation is how the number of trainable parameters scales with the input size; for a single layer if the input size m doubles, then, since the weight matrix W is an $m \times n$ matrix, so will the number of trainable parameters in that layer (ignoring the bias term). This can be inefficient for high-dimensional datasets, such as images. It is therefore common to see MLPs used in conjunction with other types of layers that can reduce dimensionality [301].

Residual neural networks

Residual neural networks (ResNets) are a modification to the overall architecture of a standard neural network. In a residual neural network, there are additional connections between layers such that the input to a given layer is added to the output of a later layer (not necessarily the following layer). This was proposed in He et al. [328] and has been shown to mitigate the effect of vanishing gradients (when the gradients tend to zero) [336] and allow for the training of deeper neural networks. It has since been extended and improved upon [337–339] and ResNets are now a machine learning mainstay [340].

These neural networks are typically constructed in blocks that each consist of multiple layers and the residual connection is between the input of the first layer in the block and the output of the last layer [328]. Following the formulation used previously, a 3–layer residual block is defined as

$$f_{\text{Block}}(\boldsymbol{x};\boldsymbol{\phi}) = f_3(f_2(f_1(\boldsymbol{x};\boldsymbol{\phi}_1);\boldsymbol{\phi}_2);\boldsymbol{\phi}_3) + \boldsymbol{x}, \tag{3.6}$$

where each f_i is a layer. This requires careful design of the output layer, in this case f_3, so that the shape of the output matches the shape of $\boldsymbol{x}$ and they can be added together.

3.3 Normalizing flows

Normalizing flows are a type of generative machine learning algorithm in a similar vein to generative adversarial networks (GANs) [341] and variational autoencoders (VAEs) [227]. There are different ways to understand and motivate normalizing flows: from a machine learning perspective or from a more statistical perspective.

Starting with the latter, normalizing flows are a family of trainable parameterized distributions over parameters $\boldsymbol{x}$, that are typically defined in the n-dimensional real space $\mathbb{R}^n$. They are highly flexible and can have complex correlations between the parameters. Similar to standard distributions, they typically have a well-defined probability density function and can be sampled from. This means they can be trained to approximate the distribution of some arbitrary data and then provide a means to evaluate an approximate probability density function (PDF) for said data, and produce new data points that are drawn from the distribution.

An alternative view of normalizing flows is as an invertible parameterized mapping from a complex distribution in the data space $\mathcal{X}$, labelled $p_{\mathcal{X}}(\boldsymbol{x})$, to a simpler distribution in the latent space $\mathcal{Z}$, labelled $p_{\mathcal{Z}}(\boldsymbol{z})$ and often known as the latent distribution. The mapping is constructed via a series of transforms and overall is typically represented as $f : \mathcal{X} \rightarrow \mathcal{Z}$. In this interpretation, the aim is to learn a mapping to a space (the latent space $\mathcal{Z}$) where certain calculations or operations are simpler, e.g. computing the PDF, drawing new samples or performing a random walk [342]. The invertible nature of the mapping then allows you to, for example, transform samples back to the original space and obtain new samples from the complex distribution.

Both interpretations of normalizing flows are useful in different contexts, and both are used throughout this thesis. We will now review the fundamentals of normalizing flows, with a focus on aspects directly relevant to work in this thesis. For a complete review, see Kobyzev et al. [343] and Papamakarios et al. [344]

3.3.1 Mathematical formulation

The mathematical formulation of normalizing flows is centred around the change of variable formula between the latent distribution $p_{\mathcal{Z}}(\boldsymbol{z})$ and the data distribution $p_{\mathcal{X}}(\boldsymbol{x})$

$$p_{\mathcal{X}}(\boldsymbol{x}) = p_{\mathcal{Z}}(f(\boldsymbol{x};\boldsymbol{\phi}))\left|\det\left(\frac{\partial f(\boldsymbol{x};\boldsymbol{\phi})}{\partial \boldsymbol{x}}\right)\right|, \tag{3.7}$$

where $f(\boldsymbol{x};\boldsymbol{\phi})$ is the normalizing flow mapping parameterized by trainable parameters $\boldsymbol{\phi}$ and $|\det(\partial f(\boldsymbol{x};\boldsymbol{\phi})/\partial \boldsymbol{x})|$ is the absolute value of the Jacobian determinant defined as

$$\left|\det\left(\frac{\partial f(\boldsymbol{x};\boldsymbol{\phi})}{\partial \boldsymbol{x}}\right)\right| = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{vmatrix}, \tag{3.8}$$

where x_i is the i'th component and $f_i \equiv f(x_i, \boldsymbol{\phi})$. The trainable parameters $\boldsymbol{\phi}$ are typically the weights of a neural network (or a series of neural networks) and their dimensionality will depend on the dimensionality of the data space, design of the flow and complexity of the neural network. The latent distribution is typically chosen to be a distribution which can be sampled from directly and has a tractable PDF, $p_{\mathcal{Z}}(\boldsymbol{z})$, i.e. it can be evaluated analytically. This leads to two distinctive features of normalizing flows:

- The PDF of the learnt distribution $p_{\mathcal{X}}(\boldsymbol{x})$ is also tractable. This contrasts with other generative algorithms, where the PDF is often intractable.
- The data distribution $p_{\mathcal{X}}(\boldsymbol{x})$ can be sampled from directly by sampling from the latent distribution $\boldsymbol{z} \sim p_{\mathcal{Z}}(\boldsymbol{z})$ and applying the inverse mapping $\boldsymbol{x} = f^{-1}(\boldsymbol{z};\boldsymbol{\phi})$ such that $\boldsymbol{x} \sim p_{\mathcal{X}}(\boldsymbol{x})$.

However, this imposes constraints on the types of functions that can be used to construct a normalizing flow. Specifically, the mapping $f : \mathcal{X} \rightarrow \mathcal{Z}$ must be invertible and the Jacobian determinant must be tractable. There are two predominant methods for constructing normalizing flows such that they meet these criteria: *coupling flows* or *autoregressive flows.*

3.3.2 Coupling flows

Coupling flows are constructed using coupling transforms, which were proposed in Dinh et al. [345]. Given an D-dimensional input vector $\boldsymbol{x} \in \mathbb{R}^D$, a coupling transforms splits the vector into two subspaces: $(\boldsymbol{x}_a, \boldsymbol{x}_b) \in \mathbb{R}^d \times \mathbb{R}^{D-d}$. One subspace is then transformed using a bijection $g(\cdot;\boldsymbol{\lambda}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ parameterized by $\boldsymbol{\lambda}$ which determines how the space is transformed. The dimensions of $\boldsymbol{\lambda}$ will depend on the dimension of the transformed subspace d and the form of $g(\cdot;\boldsymbol{\lambda})$, e.g. if the subspace is two–dimensional and the bijection is described by three parameters per dimension, then $\boldsymbol{\lambda}$ will comprise three vectors of length two $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \boldsymbol{\lambda}_3\}$. We will see an example of this later in this section when we discuss affine coupling transforms. The overall coupling transform $f : \mathbb{R}^D \to \mathbb{R}^D$ is then

$$f(\boldsymbol{x}) = \boldsymbol{y} = \begin{cases} \boldsymbol{y}_a & = \boldsymbol{x}_a, \\ \boldsymbol{y}_b & = g(\boldsymbol{x}_b; \Theta(\boldsymbol{x}_a)), \end{cases} \tag{3.9}$$

where $\Theta(\boldsymbol{x}_a)$ is a function that maps $\boldsymbol{x}_a$ to corresponding $\boldsymbol{\lambda}$, i.e. $\boldsymbol{\lambda} = \Theta(\boldsymbol{x}_a)$, and is parameterized by $\boldsymbol{\phi}$; it is often known as the conditioner. This transform is easily invertible, assuming g is also invertible,

$$f^{-1}(\boldsymbol{y}) = \boldsymbol{x} = \begin{cases} \boldsymbol{x}_a & = \boldsymbol{y}_a, \\ \boldsymbol{x}_b & = g^{-1}(\boldsymbol{y}_b; \Theta(\boldsymbol{x}_a)), \end{cases} \tag{3.10}$$

Typically, the bijection $g(\cdot;\boldsymbol{\lambda})$ is applied element-wise, so the Jacobian matrix is lower triangular

$$J = \begin{bmatrix} \mathbb{I}_d & \mathbf{0}_{d\times(D-d)} \\ \frac{\partial \boldsymbol{y}_b}{\partial \boldsymbol{x}_a} & \mathrm{diag}(\Theta(\boldsymbol{x}_a)) \end{bmatrix} \tag{3.11}$$

and the Jacobian determinant is simply the product of diagonal entries

$$\det(J) = \prod_{i=1}^{D-d} \Theta(\boldsymbol{x}_a)_i. \tag{3.12}$$

Notice how we do not need the derivative of the conditioner $\Theta(\boldsymbol{x}_a)$, so it can be arbitrarily complex. In practice, it is normally a neural network, such as an MLP or a ResNet [346], with weights $\boldsymbol{\phi}$ which are optimized given a loss function; this is discussed in section 3.3.5.

This class of transform only transforms one of the subspaces, however, if the order of the two subspaces is reversed, such that g is applied to $\boldsymbol{x}_a$, then the other subspace can also be updated. It is therefore common to stack coupling transforms and alternate which subspace is updated. The method used to split the subspaces varies by implementation, but it is common to split the dimensions in half ($\boldsymbol{x}_a = x_{1:d}$, $\boldsymbol{x}_b = x_{d+1:D}$) or use a chequerboard pattern such that every other element dimension in the overall space is updated [347]. Going forward, we will assume the input vector is split in half.

Affine coupling transforms

One specific type of coupling transform that we will use extensively throughout this work is the *affine* coupling transform, where the bijective function g that is applied to the i'th

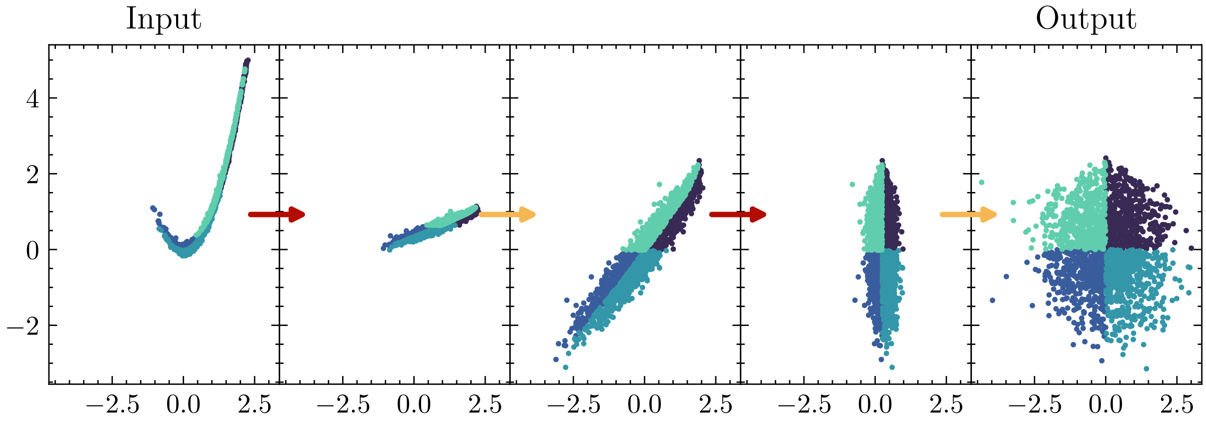


Figure 3.2: Example of sequential transforms for an affine coupling transform-based normalizing flows with 4 transforms: two affine transforms and two batch normalization transforms, which scale the input data by a running mean and variance (see section 3.3.4). The input samples are drawn from the Rosenbrock function and transformed to a Gaussian distribution. Red arrows show when an affine transform has been applied, and yellows arrows shown when batch normalization has been applied. Each colour of sample corresponds to a quadrant in the latent space and the colour of each sample is fixed, this shows how the latent space maps to the original data space. More details about this example are provided in appendix B.

element is an affine transformation

$$g(x_i; \lambda_1, \lambda_2) = x_i \exp(\lambda_1) + \lambda_2, \tag{3.13}$$

where λ_1 and λ_2 are output by the conditioner. This transform is trivial to invert, and the Jacobian determinant is $\det(J) = \exp(\lambda_1)$.

Therefore, for a D-dimensional input vector where the $\boldsymbol{x}_b$ subspace is updated conditioned on the $\boldsymbol{x}_a$, the conditioner will output two vectors, $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2\}$ of length $D - d$. The Jacobian determinant will then be

$$\det(\boldsymbol{J}) = \prod_{i=1}^{D-d} \exp(\boldsymbol{\lambda}_1)_i. \tag{3.14}$$

Again, we do not need the derivative of the conditioner, so it can be a neural network.

Normalizing flows constructed with this type of transform were proposed by Dinh et al. in [347] and are known as real non-volume preserving (RealNVP). In fig. 3.2, we show an example of using a normalizing flow constructed with coupling transforms to map samples from the 2-dimensional Rosenbrock distribution to a Gaussian distribution. More details about this example are provided in appendix B.

3.3.3 Autoregressive flows

Autoregressive flows were proposed by Kingma and Welling [227] and are constructed such that for a D-dimensional input vector $\boldsymbol{x} \in \mathbb{R}^D$, the i'th input is transformed conditioned on the previous $i-1$ inputs, denoted as $\boldsymbol{x}_{1:i-1}$,

$$y_i = g(x_i; \Theta_i(\boldsymbol{x}_{1:i-1})), \tag{3.15}$$

where $i = 2, ...D$ and Θ_1 is a constant. The overall transform f can be expressed as

$$f(\boldsymbol{x}) = \boldsymbol{y} = \begin{cases} y_1 & = g(x_1; \Theta_1), \\ y_2 & = g(x_2; \Theta_2(\boldsymbol{x}_1)), \\ y_3 & = g(x_3; \Theta_3(\boldsymbol{x}_{1:2})), \\ & \vdots \\ y_{D-1} & = g(x_{D-1}; \Theta_{D-1}(\boldsymbol{x}_{1:D-2})), \\ y_D & = g^{-1}(x_D; \Theta_D(\boldsymbol{x}_{1:D-1})). \end{cases} \tag{3.16}$$

Note that computing y_i, does not require having already computed y_{i-1}. Similarly to coupling flows, the Jacobian matrix is triangular, so the Jacobian determinant is the product of the diagonal entries.

Computing the inverse is, however, more difficult, since computing the inverse for y_i requires having already computed the values for $\boldsymbol{x}_{1:i-1}$,

$$f^{-1}(\boldsymbol{y}) = \boldsymbol{x} = \begin{cases} x_1 & = g^{-1}(y_1; \Theta_1), \\ x_2 & = g^{-1}(y_2; \Theta_2(\boldsymbol{x}_1)), \\ x_3 & = g^{-1}(y_3; \Theta_3(\boldsymbol{x}_{1:2})), \\ & \vdots \\ x_{D-1} & = g^{-1}(y_{D-1}; \Theta_{D-1}(\boldsymbol{x}_{1:D-2})), \\ x_D & = g(y_D; \Theta_D(\boldsymbol{x}_{1:D-1})). \end{cases} \tag{3.17}$$

This process is therefore inherently sequential, and makes computing the inverse more computationally expensive than the forward direction $\boldsymbol{y} = f(\boldsymbol{x})$.

Variations have also been proposed where the i'th input is transformed conditioned on previous $i-1$ outputs [348],

$$y_i = g(x_i; \Theta_i(\boldsymbol{y}_{1:i-1})). \tag{3.18}$$

This flips the design and allows for the inverse to be computed more efficiently, whilst making the forward pass more expensive.

The bijective functions g used to construct autoregressive flows are the same as those used for coupling flows, for example *masked autoregressive flows* [349] use the affine transform as defined in eq. (3.13).

3.3.4 Including additional transforms

Additional invertible transforms can be included in between the transforms in a normalizing flows. These can be transforms that either have unit-Jacobian determinants, or transforms for which the Jacobian determinant can be easily computed. These are often elementwise transforms, common examples are batch normalization and *LU* factorization.

Batch normalization

Batch normalization (or batch norm) has been used extensively when training neural networks [350]. It applies an affine scaling to each dimension of the input data based on a running mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$

$$f(x) = \frac{x - \tilde{\mu}}{\sqrt{\tilde{\sigma}^2 + \epsilon}}, \tag{3.19}$$

where $\epsilon \ll 1$ and is included for numerical stability. The estimates of the mean and variance are updated every time a new batch of training is used. Dinh et al. [347] proposed its use in normalizing flows where it can be used between transforms and the running statistics are computed per dimension; fig. 3.2 shows an example of this. Other variants have also been proposed, such as *ActNorm* [346].

LU factorization

Kingma and Dhariwal [346] proposed including a transform using the LU factorization (or LU decomposition)

$$f(\boldsymbol{x}) = \mathbf{PLU}\boldsymbol{x} + \boldsymbol{b} \tag{3.20}$$

where $\mathbf{P}$ is a fixed permutation matrix, $\mathbf{L}$ is a lower triangular matrix with ones on the diagonals, $\mathbf{U}$ is a learnable upper triangular matrix with non-zero diagonal entries and $\boldsymbol{b}$ is a learnable vector. This applies a rotation and shift that can be useful when parameters are correlated. Figure 3.3 shows an example of how *LU* factorization can be trained to transform two correlated Gaussian parameters, for more details see appendix B.

3.3.5 Training normalizing flows

When training a normalizing flow, the aim is to fit the distribution described by $p_{\mathcal{X}}(x)$ to some target distribution. Alternatively, we can think of this as learning the parameters

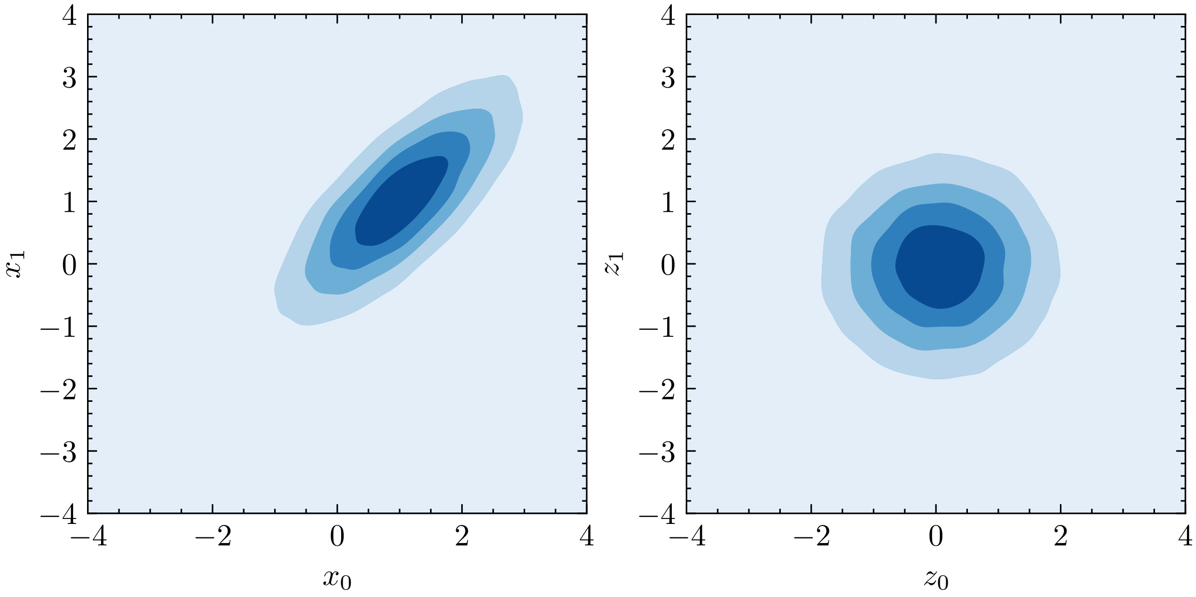


Figure 3.3: Example of how an *LU* factorization transform can be trained to transform correlated parameters to uncorrelated parameters via a rotation and shift. This transform was trained by minimizing the forward KLD using samples from the correlated distribution and a parameterized distribution comprised of an *LU* factorization transform and a standard Gaussian distribution. Each shaded contours contains 20% of the probability mass. See appendix B for details about how this figure was produced.

that describe the change of variables from the latent distribution to an approximation of the target distribution. This is typically done by minimizing the KLD divergence between the flow distribution and the target, though other methods have been proposed [351, 352]. Depending on the nature of the target density, one can choose to train a normalizing flow using the forward or reverse KLD.

Forward KLD

Given a set of samples $\{\boldsymbol{x}\}_{i=1}^{K}$ drawn from a target distribution $p^*(\boldsymbol{x})$ and a normalizing flow $p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})$, the KLD can be written as

$$\begin{aligned} L(\boldsymbol{\phi}) &= D_{\mathrm{KL}}(p^*(\boldsymbol{x})||p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})), \\ &= \mathbb{E}_{p^*(\boldsymbol{x})}[\ln p^*(\boldsymbol{x})] - \mathbb{E}_{p^*(\boldsymbol{x})}[\ln p_{\mathcal{X}}(x|\boldsymbol{\phi})], \\ &= \mathbb{E}_{p^*(\boldsymbol{x})}[\ln p^*(\boldsymbol{x})] - \mathbb{E}_{p^*(\boldsymbol{x})}\left[\ln p_{\mathcal{Z}}(f(\boldsymbol{x};\boldsymbol{\phi})) + \ln\left|\det\frac{\partial f(\boldsymbol{x};\boldsymbol{\phi})}{\partial \boldsymbol{x}}\right|\right], \end{aligned} \tag{3.21}$$

where we have used eq. (3.7) to expand $p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})$. For a fixed set of samples, the first term is constant, so it can be ignored when minimizing the forward KLD. Then, since the samples are drawn from $p^*(\boldsymbol{x})$ the expectation value can be approximated as

$$L(\boldsymbol{\phi}) \approx -\frac{1}{K}\sum_{i=1}^{K} \ln p_{\mathcal{Z}}(f(\boldsymbol{x}_i;\boldsymbol{\phi})) + \ln\left|\det\frac{\partial f(\boldsymbol{x}_i;\boldsymbol{\phi})}{\partial \boldsymbol{x}}\right|, \tag{3.22}$$

which can be used to train the normalizing flow. This is equivalent to using maximum likelihood estimation to fit the flow to the samples $\{\boldsymbol{x}\}_{i=1}^{K}$ [344].

This method is useful in cases where the target distribution cannot be evaluated directly but it can be sampled from.

Reverse KLD

The reverse KLD between a target distribution $p^*(\boldsymbol{x})$ and a normalizing flows $p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})$ can be written

$$\begin{aligned} L(\boldsymbol{\phi}) &= D_{\mathrm{KL}}(p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})||p^*(\boldsymbol{x})), \\ &= \mathbb{E}_{p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})}[\ln p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi}) - \ln p^*(\boldsymbol{x})], \\ &= \mathbb{E}_{p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})}\left[\ln p_{\mathcal{Z}}(f(\boldsymbol{x};\boldsymbol{\phi})) + \ln\left|\det\frac{\partial f(\boldsymbol{x};\boldsymbol{\phi})}{\partial \boldsymbol{x}}\right| - \ln p^*(\boldsymbol{x})\right], \\ &= \mathbb{E}_{p_{\mathcal{Z}}(\boldsymbol{z})}\left[\ln p_{\mathcal{Z}}(\boldsymbol{z}) + \ln\left|\det\frac{\partial f^{-1}(\boldsymbol{z};\boldsymbol{\phi})}{\partial \boldsymbol{z}}\right| - \ln p^*(f^{-1}(\boldsymbol{z};\boldsymbol{\phi}))\right], \end{aligned} \tag{3.23}$$

where eq. (3.7) has been used to change the expectation value from samples from $p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{\phi})$, to samples from $p_{\mathcal{Z}}(\boldsymbol{z})$. This method is useful in cases where the PDF of the target distribution can be evaluated directly, or up to a normalization constant, and is often used in variational inference [226].

Training normalizing flows with weighted measures

The KLD of two distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ was defined in section 2.2 as

$$D_{\mathrm{KL}}(p||q) = \int p(\boldsymbol{x}) \log\left[\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] \mathrm{d}\boldsymbol{x}. \tag{3.24}$$

If we consider the case of minimizing the KLD between two distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ where $p(\boldsymbol{x})$ is fixed, then

$$\begin{aligned} D_{\mathrm{KL}}(p||q) &= -\int p(\boldsymbol{x}) \log q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \int p(\boldsymbol{x}) \log p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \\ &= -\int p(\boldsymbol{x}) \log q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \text{constant}. \end{aligned} \tag{3.25}$$

The constant term is independent of $q(\boldsymbol{x})$ so we only need to compute the first term when minimizing the KLD. If we introduce a distribution $r(\boldsymbol{x})$ from which samples $\{\boldsymbol{x}\}_{i=1}^{K}$ are drawn, the integral can be approximated using Monte Carlo integration by

$$D_{\mathrm{KL}}(p||q) \approx -\frac{1}{K}\sum_{i=1}^{K}\frac{p(\boldsymbol{x}_i)}{r(\boldsymbol{x}_i)}\log q(\boldsymbol{x}_i) + \text{constant}. \tag{3.26}$$

If $r \equiv p$, this reduces to

$$D_{\mathrm{KL}}(p||q) \approx -\frac{1}{K}\sum_{i=1}^{K}\log q(\boldsymbol{x}_i) + \text{constant}, \tag{3.27}$$

and we can ignore the constant when optimizing $q(\boldsymbol{x})$. However, if $r \not\equiv p$ and both $p(\boldsymbol{x})$ and $r(\boldsymbol{x})$ are tractable, then we can define

$$D_{\mathrm{KL}}(p||q) \approx -\frac{1}{N}\sum_{i=1}^{K} w_i \log q(\boldsymbol{x}_i) + \text{constant}, \tag{3.28}$$

where $w_i \equiv p(\boldsymbol{x}_i)/r(\boldsymbol{x}_i)$. This allows for the KLD to be minimized using samples that are not from the target distribution.

Example of training a normalizing flow with weights We now demonstrate this using a simple example. We use $K = 10,000$ training samples from a 2-dimensional Gaussian with mean and covariance matrix

$$\mu_d = \begin{pmatrix} 0 & 0 \end{pmatrix}, \quad \Sigma_d = \begin{pmatrix} 25 & 0 \\ 0 & 25 \end{pmatrix}, \tag{3.29}$$

and use as a target distribution a another 2-dimensional Gaussian with mean and covariance matrix

$$\mu_t = \begin{pmatrix} 2 & 2 \end{pmatrix}, \quad \Sigma_t = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}. \tag{3.30}$$

Both distributions are shown in fig. 3.4. We then train a normalizing flow using the training samples and weights to approximate the target distribution, where the weights are given by the ratio of the PDF of the target distribution and the PDF of the training data distribution. Further details can be found in appendix B.

The results are presented in fig. 3.4 and show how, once trained, the flow produces samples that match the target distribution despite being trained with samples from a different, broader distribution. The probability-probability plot (P-P plot) in fig. 3.4 compares the theoretical cumulative distributions for each dimension to those estimated from the samples and shows that the samples pass, i.e. they are statistically consistent with being drawn from the target distribution.

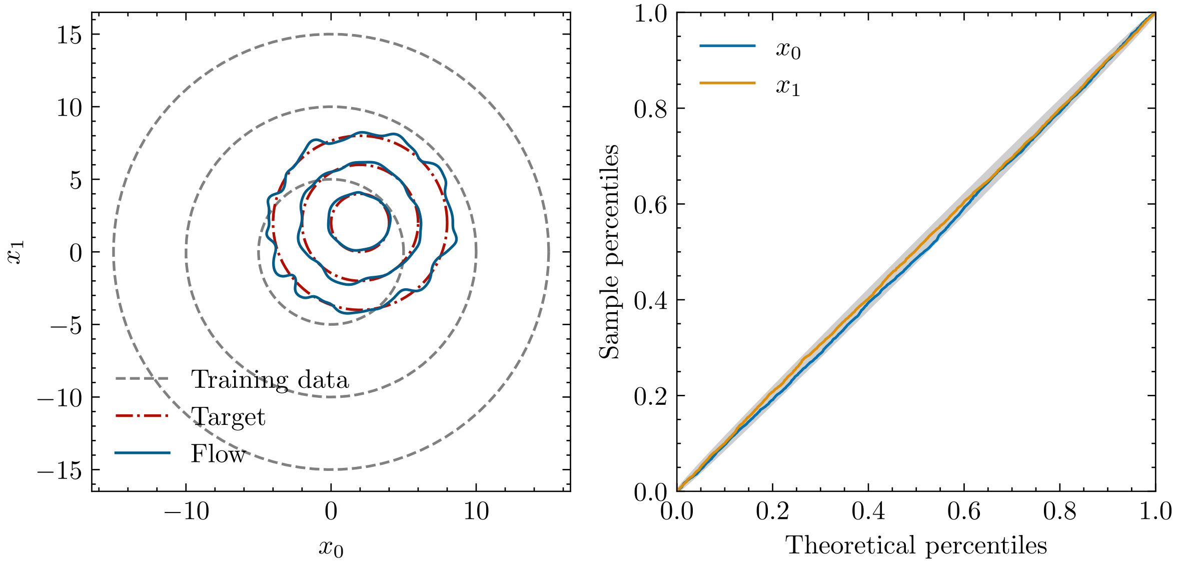


Figure 3.4: Example of training a normalizing flow with weights instead of samples from the target distribution. **Left**: the 1-, 2-, 3-σ contours for the training data distribution (grey dashed line), target distribution (red dot-dashed line) and the distribution learnt by the flow (blue solid line). The contours are computed analytically for the target and training distributions and estimated using kernel density estimate for the flow. **Right:** P-P plot for each of the dimensions (x_0, x_1) comparing the target distribution and 5000 samples from the distribution learnt by the flow. The shaded region indicates the 3-σ confidence interval. See appendix B for details about how this figure was produced.

3.3.6 Conditional normalizing flows

So far, we have only discussed normalizing flows as parameterized distributions or invertible parameterized mappings, however, they can be extended to be conditional distributions or mappings. That is, the transform applied $f : \mathcal{X} \to \mathcal{Z}$ is conditioned on an additional input $c \in \mathcal{C}$ such that

$$p_{\mathcal{X}|\mathcal{C}}(x|c) = p_{\mathcal{Z}}(f(x;c)) \left| \det \left(\frac{\partial f(x;c)}{\partial x} \right) \right| . \tag{3.31}$$

In practice, this is implemented by including the conditional input $\boldsymbol{c}$[1] in the conditioner, e.g. $\Theta(\cdot, \boldsymbol{c})$. For example, for a coupling flow, a conditional coupling transform $f(\boldsymbol{x}; c)$ is formulated as

$$\boldsymbol{y}_a = \boldsymbol{x}_a, \tag{3.32a}$$

$$\boldsymbol{y}_b = g(\boldsymbol{x}_b; \Theta(\boldsymbol{x}_a, \boldsymbol{c})), \tag{3.32b}$$

[1] $\boldsymbol{c}$ is sometimes referred to as the *context*.

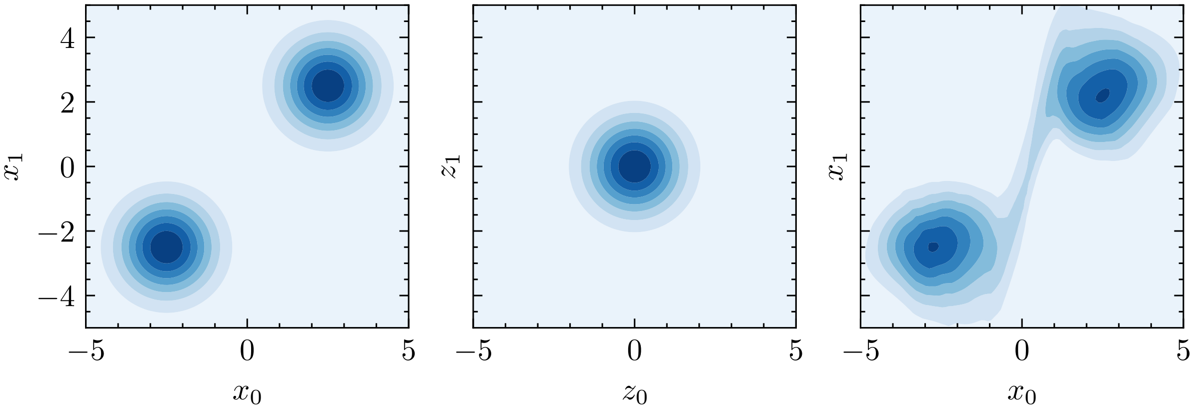


Figure 3.5: Example of the topological limitations of normalizing flows. Each shaded region contains 1/8 of the total probability mass. a) A multimodal target distribution to be learnt with a normalizing flow. b) The unimodal Gaussian latent distribution of a normalizing flow. c) The distribution learnt using RealNVP-like flow. The 'bridge' connecting the two models is clearly visible. See appendix B for details about how this figure was produced.

where the bijection g remains unchanged. Since $\boldsymbol{c}$ is not transformed, the transform is still invertible and its Jacobian determinant istractable.

This allows normalizing flows to be used as you would any other type of conditional generative algorithm, such as Conditional GANs [353] or Conditional VAEs [354]. More importantly, this is key for their use in simulated-based inference [355] since it enables training a single normalizing flow to approximate the true posterior distribution over all possible datasets, i.e. allowing for amortized inference. For more detail, see Papamakarios and Murray [356].

3.3.7 Limitations and variants of normalizing flows

The limitations of normalizing flows arise from their inherent architecture. The transformations used in normalizing flows do not change the topology of the space they map to or from, therefore the support of the latent space and data space is the same. For example, when using a Gaussian latent distribution, if the target distribution is multimodal with zero density between the modes, the normalizing flow cannot learn the exact distribution since it has inherently different topology. Instead, the normalizing flow will bridge between the two modes, this is shown in fig. 3.5.

Various methods have been proposed for addressing these topological constraints, these include using multimodal latent distributions [349, 357–359], designing normalizing flows to work on manifolds [360, 361], augmenting the data space with additional dimensions [362], creating surjective flows [363], including stochasticity via sampling [364] and using learned accept/reject sampling [365]. Each of the methods have various advantages and disadvantages, such as sacrificing the bijectivity or exact PDF, and are best suited to different applications.

Aside from topological constraints, the flexibility of a normalizing flow depends on the bijective function g to being used. Numerous variants that use different bijections have been proposed [366–369], perhaps the most popular of which are Neural Spline Flows [370] which use Rational Quadratic Splines to construct both coupling and autoregressive flows.

3.4 Applications of machine learning to gravitational-wave data analysis

Machine learning has been applied to a range of problems within gravitational-wave data analysis, which speaks to its flexibility and versatility [371]. These applications have leveraged a wide range of machine learning algorithms and have taken vastly different approaches. Broadly speaking, they fall into three broad groups: methods that use machine learning to augment existing techniques, methods that entirely replace existing techniques and methods that neither augment nor replace existing techniques and instead focus on entirely new analyses.

3.4.1 Data quality

Machine learning algorithms have been used to improve the quality of, and characterize the data from gravitational-wave interferometers.

Glitches frequently occur in the strain data. Detecting and classifying these glitches can help to understand their origin and account for them in analyses, and there has been extensive research into using machine learning to do so [372]. Two notable algorithms are: `GravitySpy` [373, 374] uses neural networks and citizen science to classify glitches directly in strain data and `iDQ` [375–377] which uses statistical machine learning to detect non-Gaussian noise artefacts using auxiliary channels and has been used in O2 and O3 [377].

There has also been a significant effort to develop algorithms that can denoise gravitational-wave times series. These techniques have been applied to denoising compact binary coalescence (CBC) signals [378, 379]. However, these are challenging to apply to inference since current techniques require defining a likelihood function, which would require modelling the noise properties of the denoising algorithm.

3.4.2 Source modelling

Accurate models of gravitational-wave sources are required for both detecting and characterizing gravitational-wave signals. In some cases, these models are well-defined and in other cases there are limited or no models for the sources.

Neural networks have been used to construct surrogates for computationally expensive CBC waveforms [380, 381]. These reduce the time to simulate waveforms, are inherently vectorized and differentiable, which all could significantly reduce the cost of inference. However, the latter two features have yet to be leveraged by widely used parameter estimation pipelines, such as `bilby` [198] and `PyCBC Inference` [25, 292], though Wong et al. [382] demonstrated their potential. Similarly, Gaussian processes have been used to model CBC waveforms with the additional benefit of including model uncertainty [383] which can also be included in parameter estimation.

Outside of CBC signals, GANs have been applied to generating signals that do not have well-defined mathematical models. In McGinn et al. [384], a GAN is trained to generate different classes of burst-like signals and interpolation in the latent space is then used to generate signals that are a mixture of the different classes. Other works have leveraged the ability to train GAN on limited datasets to learn to generate transient noise artefacts similar to those observed in current generation ground-based detectors [385, 386]. Such generative models could be used to train or validate detections algorithms.

3.4.3 Searches

Searches for gravitational-wave signals generally fall into two broad categories: modelled searches and unmodelled searches, and machine learning has been applied to both.

Existing modelled searches for CBC use matched filtering to identify signals in the data. There has been extensive research into constructing machine-learning based methods to replace matched filtering [387–389]. Many of these are based on convolutional neural network (CNN) architectures and aim to be faster than matched filtering whilst matching the sensitivity. In the past, such approaches have often been limited by the false alarm rates required for confident detection, however, recent studies have shown that machine learning-based approaches may be suitable for applications in certain regions of the search parameter space [390, 391].

Burst searches are predominantly unmodelled, relying instead on detecting excess power present in detectors. Machine learning has been used to augment existing unmodelled burst searches [392–395] and to augment burst searches targeting specific types of sources [396, 397]. In both cases, machine algorithms, such as Gaussian Mixture Models and CNNs, are trained on the output of an existing search pipeline and used as a post-processing step to improve the detection efficiency of the search. Various pure machine-learning burst searches have also been proposed, some modelled [398, 399] and some unmodelled [400, 401]. The former use approaches similar to those use for detecting CBC signals and train supervised models based label data. The latter instead treat detecting signals as an anomaly detection problem, and train models to identify outliers in the data.

Continuous gravitational-wave searches are computationally limited [44], making them a prime target for applications of machine learning. Applications to modelled searches [402, 403] have, like CBC searches, used CNNs to conduct wide parameter searches for unknown spinning neutron stars and have found such searches are effective for times-spans of $\mathcal{O}(1\,\text{day})$ but do not generalize to longer durations, e.g. $\mathcal{O}(10\,\text{days})$. However, if one instead focuses on a targetted search, Joshi and Prix [404] showed that a CNN-based approached can scale to durations $\mathcal{O}(10\,\text{days})$. Machine learning has also been used to augment existing searches for continuous gravitational waves, both modelled [405, 406] and unmodelled [407, 408]. Similar to some burst applications, these use machine learning as a post-processing step to increase search sensitivity by, for example, clustering simple candidates together or, in modelled searches, vetoing candidates that are likely of terrestrial origin.

3.4.4 Source properties

When detecting transient gravitational-wave signals, knowledge of the type of source is key, since this hints at the presence or not of an electromagnetic counterpart. Rapidly determining the type of source is there for key for electromagnetic follow-up. Clustering algorithms have been used to enabled determining the source type in low-latency [409].

Machine learning has also been used to augment existing gravitational-wave Bayesian inference algorithms for CBC. `LALInference` [96] includes the option to use `BAMBI` [410] to approximate the likelihood function using a neural network, which accelerate inference when the likelihood is computationally expensive. `bilby_mcmc` [233] implements an Markov Chain Monte Carlo (MCMC) sampler that includes various trainable proposal methods that are used alongside standard proposal methods to improve the efficiency of proposing new samples. Such approaches serve to bridge the gap between standard sampling techniques and the more recently developed end-to-end machine learning approaches.

End-to-end machine learning approaches entirely replace existing sampling algorithms and are capable of performing inference in time-scales on the order of seconds. These predominantly use variational inference to construct an approximate Bayesian posterior distribution and amortize the cost of performing inference [234–236, 411–415], though other non-variational approaches have also been proposed [416, 417]. As well as the potential speed-up once trained, another attractive property of these techniques is the ability to perform likelihood-free inference, which could enable analyses where the likelihood is intractable. Most applications have focused on applications to inference for CBCs [234, 236, 411–416], however the principles can, and have been, applied to other types of sources, such as continuous gravitational waves [235].

Machine learning has also been applied to the analyses that follow the initial parameter estimation, such as rates and populations analyses [418–421]. In these examples, density estimations algorithms, such as Gaussian Mixture models and normalizing flows, are used to accelerate analyses by approximating distributions that are otherwise intractable or computationally expensive to evaluate. Such techniques are widely applicable and can be translated to other problems that rely on complex multidimensional probability distributions.

3.5 Summary

In this chapter, we have introduced machine learning and highlighted the range of types and possible applications, and then focused on neural networks. We described the fundamentals of neural networks, starting from a single neuron before introducing the notion of layer and then activation functions. We covered how neural networks are trained using a loss function and backpropagation, and the importance of data in this training process.

With the fundamentals in hand, we reviewed the different types of neural networks that we will use going forward before focusing on a specific type of machine learning algorithm that incorporates neural networks: normalizing flows. They are a type of density estimation algorithm that can be used to approximate complex multidimensional distributions via a series of invertible transforms and are a cornerstone of the worked presented throughout the subsequent chapters of this thesis. We described their mathematical formulation, which relies on the use of parameterized bijective transforms with tractable Jacobian determinants and is centred around the change of variable formula. We detailed the different methods used to construct these transforms, the additional types of transforms that can be included and the loss functions used to train them. We discussed different variants of normalizing flows and their limitations, which will influence how we will leverage them and, as we will see in later chapters, can be a limiting factor in some cases.

Finally, we provided an overview of applications of machine learning to gravitational-wave data analysis, highlighting it use for improving data quality, modelling and detecting gravitational-wave sources and performing Bayesian inference. These methods include end-to-end machine learning approaches that enabled new analyses or replace existing ones and machine-learning augmented methods that improve upon existing techniques. We will focus on the latter, in particular how normalizing flows can be used to improve stochastic sampling techniques for Bayesian inference, which will be the focus of the following chapters.

Chapter 4

Nested sampling with normalizing flows for gravitational-wave inference

This chapter introduces `nessai`: nested sampling with artificial intelligence, a novel nested sampling algorithm that incorporates machine learning to accelerate Bayesian inference and demonstrates its application to gravitational-wave parameter estimation.

This work has been published in Williams et al. [1] and it is presented almost verbatim. We have omitted the introductory sections and updated certain symbols and spellings to be consistent with the other chapters. We have also included and new figure, fig. 4.7, and included results that were originally included in the appendices in the main text. The remaining appendices are included in appendix D where we have also removed the explanation on training normalizing flows since this is already covered in chapter 3.

The symbols used for certain parameters have also been updated for consistency withe rest of the thesis. Figures 4.3 and D.1 have also been remade to ensure the symbols used throughout the text are consistent. As a result, the p-values have changed slightly but this does not impact the overall conclusions. Figure 4.6 was also regenerated for the same reason.

The chapter is structured as follows: sections 4.1 and 4.2 present the method and overall algorithm, and section 4.3 describes problem-specific modifications for gravitational-wave inference. In section 4.4, we compare our approach to existing methods. Section 4.5 presents results used to validate the algorithm and comparisons to an existing widely-used sampler called `dynesty`. Finally, section 4.6 summarizes our findings and discusses avenues for further work.

Nested sampling is one of the cornerstones of gravitational-wave inference for compact binary coalescence (CBC) signals; it provides estimates of the Bayesian evidence and posterior samples, which inform us about the nature of systems that produce such signals. However, such analyses are computationally expensive, and their cost is directly related to the cost of evaluating the likelihood (see section 2.6) and the length of data being analysed. Typical CBC analyses in current generation ground-based interferometers take of order days to weeks with current inference algorithms [96, 122, 198]. Given the ever-increasing rate of CBC detections [74] and increasing complexity of the physics being modelled [99, 120], there is therefore a necessity for more efficient inference algorithms.

Machine learning is being applied to increasingly more problems in physics and astronomy, e.g. [422–426], including gravitational-wave data analysis; see section 3.4 for a review. It has many appealing attributes that make it well suited to various problems, such as the inference speed of trained algorithms or the ability to learn complex features in data without a pre-defined model. In the context of parameter estimation for CBC signals, it has predominantly been leveraged to develop algorithms that do not rely on stochastic sampling techniques and can instead, once trained, directly produce samples from an approximate posterior distribution [234, 236, 411–416]. Such algorithms have the potential to perform inference on time-scales of order milliseconds but the apparent opaqueness of these techniques and the need to redesign existing workflows has somewhat slowed their widespread adoption.

In this work, we consider a different approach to leveraging machine learning for gravitational-wave inference: augmenting an existing technique. In particular, we focus on nested sampling and propose a novel modified algorithm that incorporates machine learning in the process of drawing new samples from the likelihood-constrained prior. As discussed in section 2.6, this is the predominant bottleneck in nested sampling, and improving the efficiency of this step would reduce the overall cost of performing inference. Our focus is on normalizing flows (see section 3.3 for details), a type of generative machine learning algorithm that can model complex probability distributions, which we train to approximate the distributions of live points at a given iteration.

4.1 Method

We present a novel method for sampling within a given iso-likelihood using a normalizing flow. The normalizing flow learns the distribution of a set of live points and is constructed such that the learnt distribution can be sampled from analytically. We introduce additional steps to ensure that the samples are distributed according to the *sampling prior*[1] and

[1] We use sampling prior to denote the prior used for nested sampling and to distinguish it from the latent prior used in normalizing flows.

bounded by the iso-likelihood contour. This eliminates the need to evolve new samples and allows us to efficiently draw new samples within a complex iso-likelihood contour. A more efficient proposal equates to fewer rejected points and, since the likelihood must be computed before a point can be rejected, this is also equivalent to a reduction in the number of likelihood evaluations. We first describe our method in the isolated case of a single set of live points at a given iteration and then present a nested sampling algorithm which incorporates it.

4.1.1 Sampling within an iso-likelihood contour.

At any given point in the nested sampling algorithm there is a current set of live points which by definition is contained within the iso-likelihood contour defined by the *worst point* with likelihood $\mathcal{L}^*$. We must then draw new samples within that contour that are independently and identically distributed (i.i.d.) according to the prior and we do so using an algorithm centred around a normalizing flow. We describe the implementation of our algorithm in terms of four steps:

How to define an iso-likelihood contour using a normalizing flow. If we treat the sampling space as the physical space $\mathcal{X}$ and the live points as the data $\{\boldsymbol{x}\}_{i=1}^{N_{\text{live}}}$ we can then train a normalizing flow to approximate the distribution of the live points to within some error and use it to draw new samples in $\mathcal{X}$. This requires sampling from the latent prior $p_{\mathcal{Z}}$, which we choose to be an n-dimensional Gaussian, and then applying the inverse mapping learned by flow f^{-1}. However, since our choice of latent prior $p_{\mathcal{Z}}$ has an infinite domain, the resulting distribution of samples $p_{\mathcal{X}}$ will not be bounded by the iso-likelihood contour or distributed according to the sampling prior. We therefore examine the notion of an iso-likelihood contour in the context of the normalizing flow with an n-dimensional Gaussian latent prior.

How to determine the contour given the current set of live points. Once trained, the normalizing flow can be used to map the current *worst point* in the physical space $\boldsymbol{x}^*$ to the latent space $\mathcal{Z}$ as $\boldsymbol{z}^*$. This point has a likelihood in the latent space $\mathcal{L}_{\mathcal{Z}}^*$ given by $p_{\mathcal{Z}}(\boldsymbol{z}^*)$ and, since $p_{\mathcal{Z}}$ is an n-dimensional Gaussian, points of equal likelihood lie on the $(n-1)$-sphere with radius r^* given by $\boldsymbol{z}^*$. If we assume a perfect mapping, then this iso-likelihood contour in the latent space can be mapped to an iso-likelihood contour in the physical space. We can therefore sample within the contour in latent space and use the inverse mapping to produce samples within the contour in the data space.

How to sample within the contour. We use two approaches for drawing K new samples $\boldsymbol{z}_i$ in the latent space given a radius r^*, these produce normally and uniformly

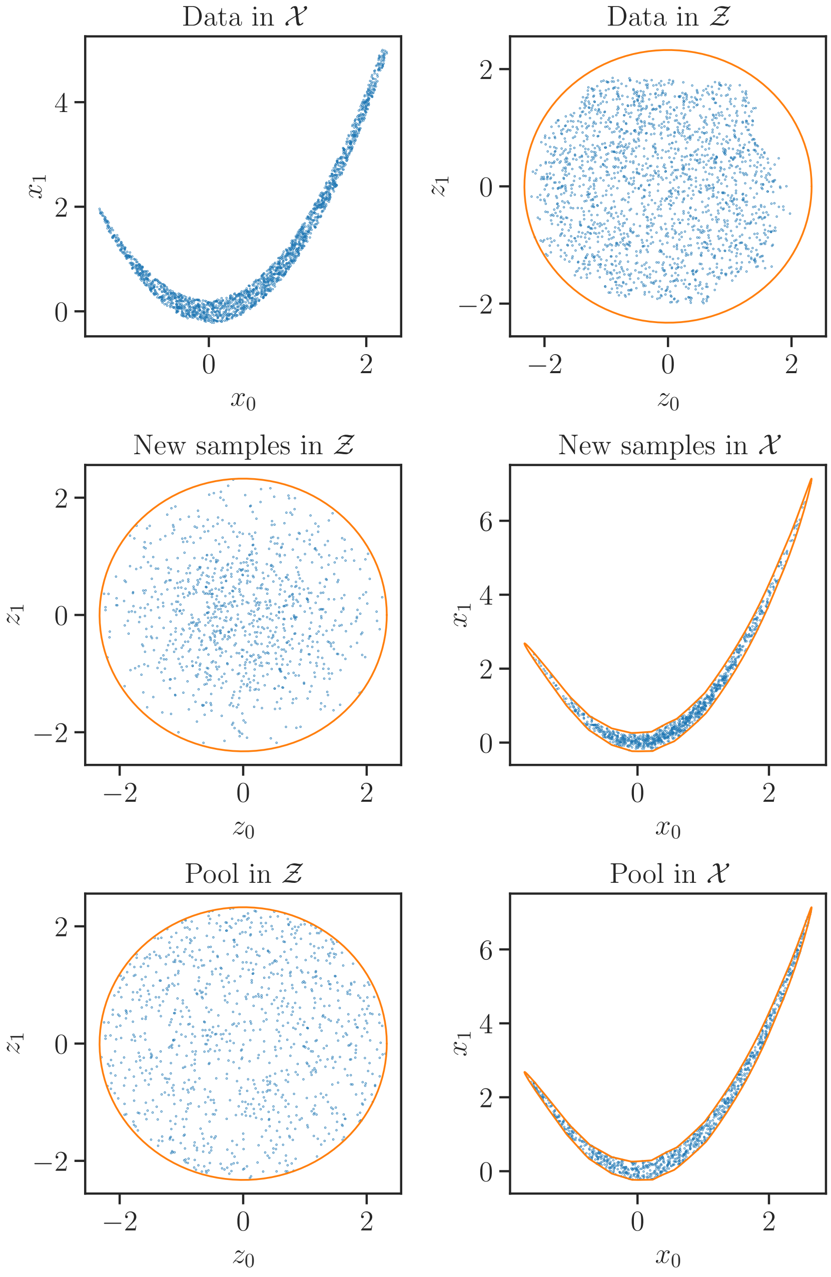


Figure 4.1: Example of how a normalizing flow trained on a set of live points can produce samples within current likelihood contour for a simple two-dimensional parameter space. **Top:** example of training samples in the physical space $\mathcal{X}$ and learned mapping to the latent space $\mathcal{Z}$ with the likelihood contour for the current *worst point* shown in orange. **Middle:** samples drawn from a truncated Gaussian within the likelihood contour in $\mathcal{Z}$ and mapped to $\mathcal{X}$ using the inverse mapping. **Bottom:** pool of accepted samples after applying rejection sampling until 1000 points are obtained, shown in both $\mathcal{Z}$ and $\mathcal{X}$.

distributed samples respectively. Both start by drawing K samples on the $(n-1)$-sphere using the algorithm proposed in [427, 428] where K-dimensional vectors are drawn from an n-dimensional unit Gaussian and then normalized using the Euclidean norm. These samples $\boldsymbol{y}_i$ can then be rescaled to obtain samples within in the n-ball

$$\boldsymbol{z}_i = \rho_i \frac{\boldsymbol{y}_i}{||\boldsymbol{y}_i||_2}, \tag{4.1}$$

where the choice of distribution for ρ_i determines how the resulting samples $\boldsymbol{z}_i$ are distributed in $\mathcal{Z}$. For uniformly distributed samples $\rho_i = u^{1/n}$ where $u \sim \mathcal{U}(0, r^*)$ and for normally distributed samples $\rho_i \sim \chi(n)$ where $\chi(n)$ is a χ-distribution with n degrees of freedom truncated at r. The inverse mapping of the normalizing flow can then be applied to $\boldsymbol{z}_i$ to obtain samples in the physical space. We consider two approaches because sampling from a truncated Gaussian in high-dimensional spaces[2] can become inefficient for large values of r^*, though we found this has minimal impact on the analyses considered in section 4.5.

How to ensure new samples are drawn according to the prior. The samples obtained in the previous step must be re-sampled such that they are distributed according to the sampling prior. We use rejection sampling and compute weights α_i for each sample

$$\alpha_i = \frac{p(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i)}, \tag{4.2}$$

where $\boldsymbol{x}_i = f^{-1}(\boldsymbol{z}_i)$, $p(\boldsymbol{x})$ is the sampling prior and $q(\boldsymbol{x})$ is the probability density function (PDF) of the proposal distribution which is computed using the inverse of eq. (2.6)

$$q(\boldsymbol{x}) = q(f^{-1}(\boldsymbol{z})) = p_{\mathcal{Z}}(\boldsymbol{z}) \left| \det \left(\frac{\partial f^{-1}(\boldsymbol{z})}{\partial \boldsymbol{x}} \right) \right|^{-1}. \tag{4.3}$$

The choice of latent prior $p_{\mathcal{Z}}(\boldsymbol{x})$ will depend on which method was used to draw the samples in the latent space. The weights eq. (4.2) are then rescaled such that their maximum value is one. We then draw N samples $u \sim \mathcal{U}[0, 1]$ and accept samples for which $\alpha_i/u_i > 1$. In fig. 4.1, we show an example of this process for a simple two-dimensional case.

[2] The notion of a high-dimensional space is relative, in the context of gravitational-wave inference we often consider a problem with tens of dimensions to be high-dimensional, since the parameter spaces are typically highly correlated meaning many methods do not scale to the full parameter space. Conversely, in other contexts, such as machine learning, we may only consider a problem to be high-dimensional once it has hundreds or thousands of dimensions, though the parameter space may lie on a lower dimensional manifold in the high-dimensional space.

4.2 Algorithm outline

We identify three key stages of the sampling approach detailed in section 4.1.1 that we then incorporate into the nested sampling algorithm:

- **Training:** a normalizing flow is trained on the current set of N_{live} live points using forward Kullback-Leibler divergence (KLD) as previously described in section 3.3.5.
- **Population:** once the normalizing flow is trained, samples are drawn within the n-ball of radius r determined as described in section 4.1.1. These samples are then mapped to $\mathcal{X}$, re-sampled according to the prior and finally stored in the pool of new samples.
- **Proposal**: once the pool of new samples has been populated, new live points are drawn at random from the pool and then removed until the pool is empty or the normalizing flow is retrained.

The start of the algorithm remains unchanged: N_{live} live points are drawn from the prior distribution and their log-likelihoods computed. We then start the iterative process of determining the *worst* live point with log-likelihood $\mathcal{L}^* = \mathcal{L}(\theta^*)$ and drawing a replacement live point that lies within the likelihood contour. In our modified algorithm, we use standard rejection sampling from the prior for the first M points (typically $2N_{\text{live}}$) or until it becomes inefficient. The normalizing flow is then trained on the current N_{live} live points, allowing us to sample from an approximation of the likelihood constrained priors and populate the pool. Once the pool is populated, a replacement point is drawn from the pool, its log-likelihood is computed and if it is greater than $\mathcal{L}^*$, the point is accepted; if not, more points are drawn until one is accepted. The proposal stage is then repeated for subsequent worst live points until one of four criteria is met:

- **the proposal pool is depleted:** the normalizing flow is retrained using the current live points and *worst* live point is used to compute a new radius and the population stage is repeated,
- **the acceptance rate falls below a user-defined criterion:** the current proposal pool is discarded and the normalizing flow is retrained. This threshold is defined by the user,
- **the normalizing flow is retrained with the current live points:** the criterion for this is a number of iterations, by default this every N_{live} iterations, but it can also be disabled such that flow is only trained when the proposal pool is depleted,
- **the nested sampling convergence criterion is met:** the algorithm terminates.

4.3 Reparameterizations

Since this algorithm relies on the normalizing flows' ability to approximate the distribution of live points at various stages throughout the sampling, we include a series of reparameterizations of $\mathcal{X}$ to reduce the complexity of the data space and removing certain features. We denote this reparameterized space $\mathcal{X}'$ and include the Jacobian for each reparameterization in eq. (4.3). These reparameterizations are:

- **Rescaling:** we add the option to rescale the input data according to either the *sampling priors* or the current minimum and maximum values such that all the parameters in $\mathcal{X}'$ are defined over the same domain. As a default, we use $[-1, 1]^n$.
- **Boundary inversion:** we observe that asymmetric distributions with high density regions near the prior bounds are often under-sampled. To mitigate this effect, we add the option to *mirror* the live points around such bounds and train on the resulting symmetric distribution. Further details are provided in appendix D.1.

We also introduce additional settings which help with convergence and sampling efficiency, some of these and discussed in section 4.5.5 and a comprehensive list can be found in the online documentation for our sampler [429].

4.3.1 Gravitational-wave reparameterizations

The gravitational-wave parameter space is typically 15-dimensional and contains various degeneracies between parameters such as the masses, inclination and luminosity distance which can make sampling inefficient. Previous work has shown that certain reparameterizations can improve sampling efficiency [96]. We use two of these: chirp mass $\mathcal{M}$ and asymmetric mass ratio q replace the component masses and we use the *system-frame* parameterization in place of the *radiation-frame* to describe the orientation of the binary [123].

Angular parameters

More than half of the parameters to sample are angles and we note that the periodicity of these angles is not encoded in the mapping learned by the normalizing flow since the latent space $\mathcal{Z}$ is continuous and unbounded. We therefore include a further reparameterization specifically for the angular parameters θ_i. We assume that each angle has a corresponding radial component ρ_{θ_i} and together they describe a position in a two-dimensional plane. We can therefore use standard transformations to express this position in Cartesian coordinates $(x_{\theta_i}, y_{\theta_i})$:

$$\begin{aligned} x_{\theta_i} &= \rho_{\theta_i} \cos \theta_i, \\ y_{\theta_i} &= \rho_{\theta_i} \sin \theta_i. \end{aligned} \tag{4.4}$$

If we choose the distribution of radial components such that $\rho_{\theta_i} \in [0, \infty)$ then $x_{\theta_i}, y_{\theta_i} \in (-\infty, \infty)$. Since we are using a Gaussian latent prior $p_{\mathcal{Z}}$, we sample ρ_{θ_i} from a χ-distribution with two degrees of freedom such that, if the angle is uniformly distributed on $[0, 2\pi]$, the resulting distribution of $(x_{\theta_i}, y_{\theta_i})$ is Gaussian. We use this treatment for the phase, inclination, polarization and all four spin angles, for polarization we rescale the angles to $[0, 2\pi]$ before applying the transformation to Cartesian coordinates. This reparameterization also naturally includes periodic boundary conditions for the angles with uniform priors.

The sky location is described by a further two angles, right ascension α and declination δ. For these angles, we extend the previous treatment from two-dimensional to three-dimensional Cartesian coordinates (x, y, z) and draw the radial component ρ from a χ-distribution with three degrees of freedom. For the standard priors, $p(\alpha) \sim \mathcal{U}[0, 2\pi]$ and $p(\delta) \sim \cos \delta$, the resulting distribution of (x, y, z) is again Gaussian.

Spins

The spin magnitudes a_1 and a_2 also require a specific treatment. They are typically defined on $[0, 0.99]$ with uniform priors and, importantly, the posterior distributions are often broad and span the entire prior range. We consider applying the boundary inversion to both bounds but in practice find this ineffective. We instead opt to map χ_i into a two-dimensional plane with positions described using Cartesian coordinates x_{a_i} and y_{a_i}. We achieve this by first defining a rescaled magnitude $\hat{a}_i \in [0, 1]$ which is obtained using the corresponding priors. Then, we consider the angle defined by $\hat{a}_i \pi$ and, again, introduce a radial component $\rho_{a_i} \sim \chi(2)$. The corresponding Cartesian coordinates (x_{a_i}, y_{a_i}) are defined on $[0, \infty)$ and $(-\infty, \infty)$ respectively. However, we know that the coupling transforms we have chosen to use are better suited to unbounded domains. To avoid this, we introduce a random variable k which is drawn from a Rademacher distribution and include it in the Cartesian coordinate transform

$$\begin{aligned} x_{\chi_i} &= \rho_{\chi_i} \cos \hat{\chi}_i \pi, \\ y_{\chi_i} &= k \rho_{\chi_i} \sin \hat{\chi}_i \pi. \end{aligned} \tag{4.5}$$

As a result of including k, $(x_{\chi_i}, y_{\chi_i}) \in (-\infty, \infty)$ and the hard boundary at $x_{\chi_i} = 0$ has been avoided.

Distance

We choose to reparameterize the luminosity distance d_{L} such that the prior for the resulting parameter d_{U} is uniform. The exact reparameterization therefore depends on the prior used for d_{L}. In this work we choose to use a prior on d_{L} that is uniform in co-moving volume so we first convert the luminosity distance to a co-moving distance d_{C} and then the uniform parameter is simply $d_{\mathrm{U}} = d_{\mathrm{C}}^3$. Similar reparameterizations can be determined for other commonly used distances priors such as a power law. Additionally, we allow boundary inversion as described in section 4.2 but limit it to only the upper bound since in practice the luminosity distance posterior will not rail against the lower bound.

Other parameters

The remaining parameters are the chirp mass $\mathcal{M}$, mass ratio q and time of coalescence t_{c}, for which we use the reparameterizations from $\mathcal{X}$ to $\mathcal{X}'$ mentioned in section 4.2, allowing boundary inversion for q.

4.3.2 Implementation

We use the implementation of normalizing flows in `PyTorch` [430] available in `nflows` [431] which allows for a wide variety of normalizing flows to be used. However, we choose to use coupling transforms [347] because of the tractable Jacobian and ease of computing the inverse mapping. As suggested in [345, 346], we include invertible linear transforms that randomly permute the parameters before each coupling transform, allowing all the parameters to interact with each other. We also include batch normalization [350] after each coupling transform, as described in Dinh et al. [347]. We use a residual neural network (ResNet) [328, 337] for computing the parameters for each transform. We train the normalizing flows with the adaptive moment estimation (Adam) optimizer [325]. In appendix D.3, we detail the specific parameters used for the results presented in section 4.5.

Our sampler, `nessai` (Nested Sampling with Artificial Intelligence), is available as an open source package [432] and documentation is also available online [429].

4.4 Related Work

Different frameworks and samplers have been developed for gravitational-wave inference. `LALInference` [96] implements nested sampling and Markov Chain Monte Carlo (MCMC) with specific proposal methods for the gravitational-wave parameter space and has been used extensively for analyses of the first gravitational wave detection [6] and Gravitational-Wave Transient Catalog 1 (GWTC-1) and Gravitational-Wave Transient Catalog 2 (GWTC-2) [7, 8]. More recently, the `Python` package `bilby` [198] has been developed to use off-

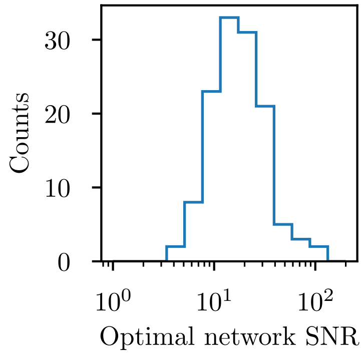


Figure 4.2: Distribution of the optimal network signal-to-noise ratio (SNR) for the 128 simulated gravitational-wave injections in simulated Gaussian noise. The priors on chirp mass and luminosity distance appendix D.2 were chosen such that the ringdown frequency [436] does not exceed the Nyquist frequency and that the majority of signals have detectable optimal network SNRs

the-shelf samplers, such as `dynesty` [197], and been shown to achieve comparable results to `LALInference` on GWTC-1 [122].

Machine learning has previously been incorporated into stochastic sampling algorithms; in [410] the likelihood function is approximated with a neural network, and in [433] neural networks are used to generalize Hamiltonian Monte Carlo. More closely related to our work, normalizing flows have been used to improve the efficiency of MCMC methods by reparameterizing the sampling space [434] and a similar approach has also been extended to MCMC sampling in nested sampling in [435].

Recent work has shown that likelihood-free inference using conditional variational autoencoders (VAEs) [234, 416] and normalizing flows [411, 412] can produce posterior distributions for compact binary coalescence of binary black holes. These approaches promise to drastically reduce the cost of producing posterior samples when compared to traditional stochastic sampling methods. However, they require large amounts of training data and they currently lack the flexibility to deal with, for example, different power spectral densitys (PSDs), high sampling frequencies and long duration signals.

4.5 Results

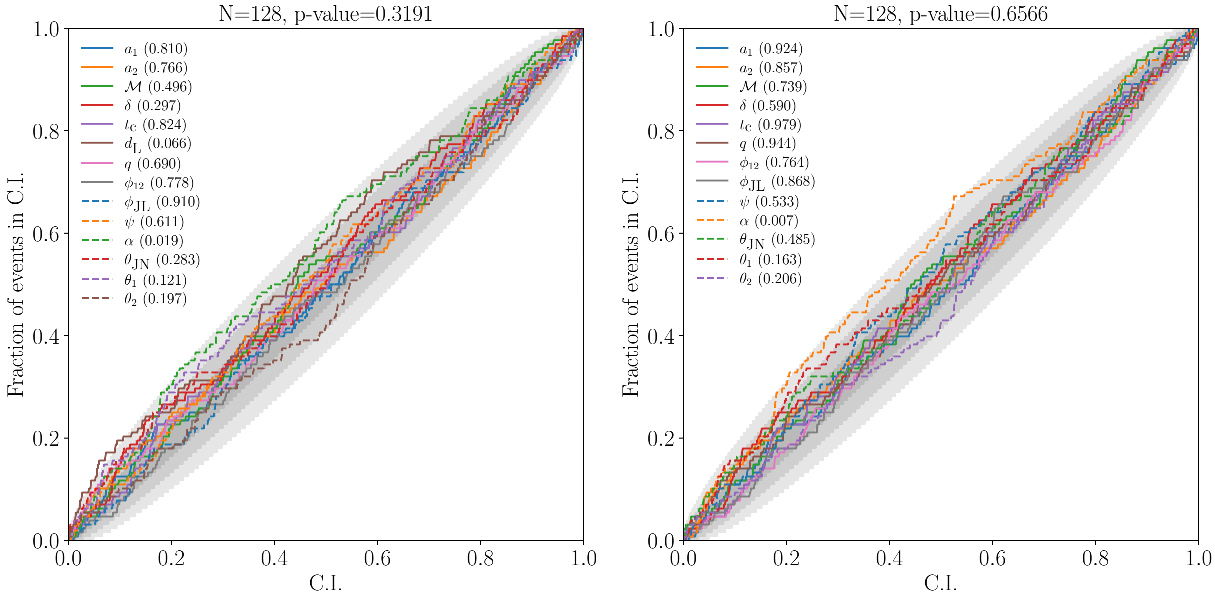
We chose to evaluate the performance of our sampling algorithm, `nessai`, with simulated gravitational-wave signals from CBC. The parameter is multidimensional and various parameters are correlated, which can prove challenging when sampling. The likelihood, as defined in section 2.10.1, is also typically computationally costly to evaluate, predominantly because of the cost of generating the gravitational-wave waveform $\boldsymbol{h}_{\boldsymbol{\theta}}$ which scales with the length of the observation. This makes it well suited to our sampler since this can partially offset the cost of training the normalizing flow. We first use probability-probability plots to check the consistency of our sampler and then compare our results to those obtained using `dynesty`. We then highlight how the likelihood computation can be parallelized in `nessai` before finally discussing various diagnostics that can be used to identify problems during sampling and tune the sampler settings.

We use `bilby_pipe` and `bilby` [198] to simulate 128 injections using `IMRPhenomPv2` [119, 437] sampled at 2048 Hz with 4-second observing time in a three-detector network with Advanced LIGO (aLIGO) Hanford, aLIGO Livingston and Advanced Virgo (AdVirgo) at design sensitivity [70, 77]. We set the minimum frequency to 20 Hz and Gaussian noise is added to the injections using the PSDs for each detector. We choose uniform priors on chirp mass $\mathcal{M} \sim \mathcal{U}[25, 35]$ $\mathrm{M}_{\odot}$ and asymmetric mass ratio $q \sim \mathcal{U}$ $[0.125, 1.0]$, a prior on luminosity distance that is uniform in co-moving volume on (100–2000) Mpc, a uniform prior for the reference time at the geocentre with width 0.2 s and the remaining priors are set to the defaults for precessing binary black holes in `bilby` [122], see appendix D.2 for a complete list. The specific priors on chirp mass and luminosity distance are chosen such that ringdown frequency [436] does not exceed the Nyquist frequency and the majority of signals have detectable optimal network SNRs, the distribution of SNRs is shown in fig. 4.2.

We analyse the injections with our sampling algorithm, `nessai`, outlined in section 4.2 and include the specific reparameterizations for gravitational wave analyses described in section 4.3.1. We choose to analyse each injection twice: once with just phase marginalization, and once with both phase and distance marginalization. Further details of the exact settings used for `nessai` are provided in appendix D.3.

4.5.1 Result validation

P-P plots are a standard method of verifying the performance of sampling algorithms [229, 230]. They test whether the correct proportion of injected values are recovered at a given confidence interval for a specific prior distribution. These tests are particularly useful when using a Gaussian likelihood, such as eq. (2.51), since the fraction of events within a given confidence interval should be uniformly distributed and we can therefore compute p-values for each parameter and a combined p-value for all the parameters. We produce P-P plots for both of our analyses using `bilby` and present the results in fig. 4.3. For an idealized sampler for the $p\%$ confidence interval, $p\%$ of the events should be recovered, this would correspond to a diagonal line. In practice, we expect to see deviation from the diagonal, as such the 1-, 2- and 3-σ confidence intervals are also shown in fig. 4.3. These results show that `nessai` consistently recovers for the posteriors for the 128 injections but also indicate that the luminosity distance is consistently harder to sample. The combined p-values of 0.3391 and 0.6566 for our analyses without and with distance marginalization



(a) With phase marginalization

(b) With phase and distance marginalization

Figure 4.3: Probability-probability plots (P-P plots) showing the confidence interval versus the fraction of the events within that confidence interval for the posterior distributions obtained using our analysis `nessai` for 128 simulated compact binary coalescence signals produced with `bilby` and `bilby_pipe`. The 1-, 2- and 3-σ confidence intervals are indicated by the shaded regions and p-values are shown for each of the parameters and the combined p-value is also shown.

serve as further verification.

4.5.2 Comparison to `dynesty`

To further validate our results, we compare them to those obtained with `dynesty` [197], another nested sampling algorithm commonly used in gravitational-wave inference [9, 10, 122, 198]. We use the configuration described in [122] but increase the number of live points to 2000 and run on a single thread to ensure as direct of a comparison with `nessai` as possible. With these settings `dynesty` passes the P-P test (see appendix D.4) but we note that these settings are the minimum required to produce reliable results and in practice more conservative settings are often used. Additionally, several injections required a second analysis with a different sampling-seed in order to reach convergence. The results obtained with `dynesty` allow us to verify the log-evidences returned by `nessai` since these cannot be computed analytically and provide a point of reference when considering the number of likelihood evaluations and total computational time.

In fig. 4.4, we compare the log-evidences returned by `dynesty` and `nessai`. If `nessai` was consistently over or under-estimating the log-evidence when compared to `dynesty`, this would indicate a potential problem during sampling, such as over- or under-constraining,

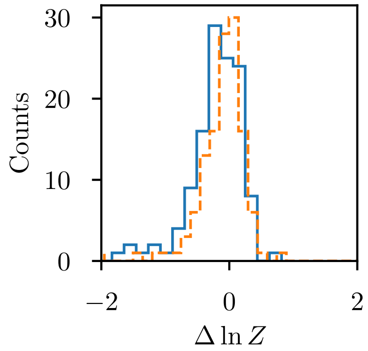


Figure 4.4: Difference between the log-evidences $\Delta \ln Z$ obtained using `dynesty` and `nessai` for all 128 injections with distance marginalization (dashed line) and without distance marginalization (solid line).

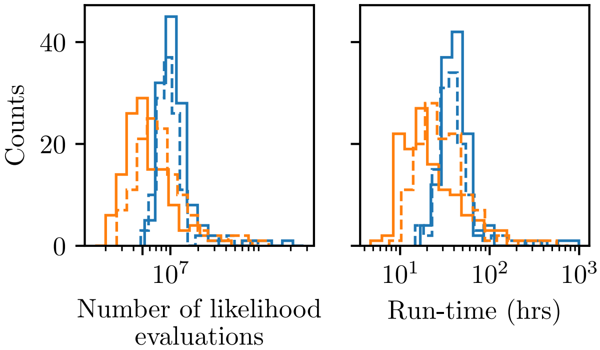


Figure 4.5: Distribution of the total number of likelihood evaluations required to reach convergence for and total run time for `dynesty` (blue) and `nessai` (orange) when applied to 128 simulated signals from compact binary coalescence with the priors and sampler settings described in section 4.5 and appendix D.3. The results with distance marginalization disabled are shown with solid lines and those with distance marginalization enabled are shown with dashed lines.

which would lead to biased results. The results in fig. 4.4 show no such bias. However, since sampling is a stochastic process there is an error associated with the computed log-evidence. The theoretical error can be approximated using the information content $\mathcal{H}$ and the number of live points N_{live}, $\delta \log Z \approx \sqrt{H/N_{\text{live}}}$. To quantify this error, we repeat the analysis on a single injection with 50 different sampling seeds and compute an approximate error $\delta \log Z \approx 0.092$. In practice, we observe a wider spread of log-evidences of 0.11, this is consistent with previous analyses which determined that there are additional sources of uncertainty [237].

`nessai` is designed with the aim of improving the efficiency of drawing replacement live points at the cost of repeatedly training a normalizing flow and populating a pool of live points. An improvement in the efficiency translates to a reduction in the total number of likelihood evaluations, since the likelihood must be computed for each rejected point. We therefore compare the total number of likelihood evaluations required to reach convergence for each sampler in fig. 4.5 with and without distance marginalization. `nessai` requires a median of 5.04×10^6 and 7.22×10^6 likelihood evaluations to converge with and without distance marginalization respectively, and `dynesty` requires 10.44×10^6 and 9.67×10^6. In contrast to `dynesty`, the results in fig. 4.5 show that sampling with `nessai` is more efficient without distance marginalization. We attribute this to a combination of the reparameterization used for luminosity distance and the specific sampler settings used in these experiments, which were tuned for analyses without distance marginalization. Subsequent investigations presented in section 5.4 have shown that this result does not hold generally, and with improved settings, sampling with distance marginalization is more efficient than without.

This, however, does not directly translate to the run-times for each sampler, since they each have different additional computational costs associated with sampling. In fig. 4.5 we show the total run-time for each sampler and when comparing the median run-times we observe that `nessai` is 2.32 times faster than `dynesty` without distance marginalization and 1.40 times faster with it. Additionally, we examine the proportion of the run-time spent on training and population and find that on average population-time accounts for approximately 40% of the total run-time and training-time accounts for a further 8%. We also note that the cost of training and population does not depend on the cost of evaluating the likelihood, as such, the fraction of the total run-time will decrease as the cost of evaluating the likelihood cost increases.

For each injection, we can also compare the posterior distributions produced by each sampler. These allow us to quickly identify discrepancies between samplers for specific injections or regions of the parameter space. We show an example of such a comparison is fig. 4.6. We can also quantify the differences between the posterior distributions obtained with `dynesty` and `nessai` by computing the Jensen-Shannon divergence (JSD) between the 1-dimensional marginal posterior distributions as described in section 2.9.2. To do so, we take 100 random subsets of 1000 samples from the marginal distributions we are comparing and compute the median JSD over the subsets. Figure 4.7 shows the distribution of JSDs per sampled parameter and the distribution of the maximum JSD across all parameters per injection. We also include the value for the posterior distributions show in fig. 4.6 as a point of reference. Since we are using 1000 samples, the threshold for the JSD that indicates a statistically significant difference is 10 mbits. These results show that the majority of the marginal posterior distributions are statistically consistent between `dynesty` and `nessai`. In approximately a third of cases, with or without, distance marginalization, the maximum JSD is above the 10 mbits threshold suggesting that at least one of the pairs of marginal distributions are statistically different. However, since

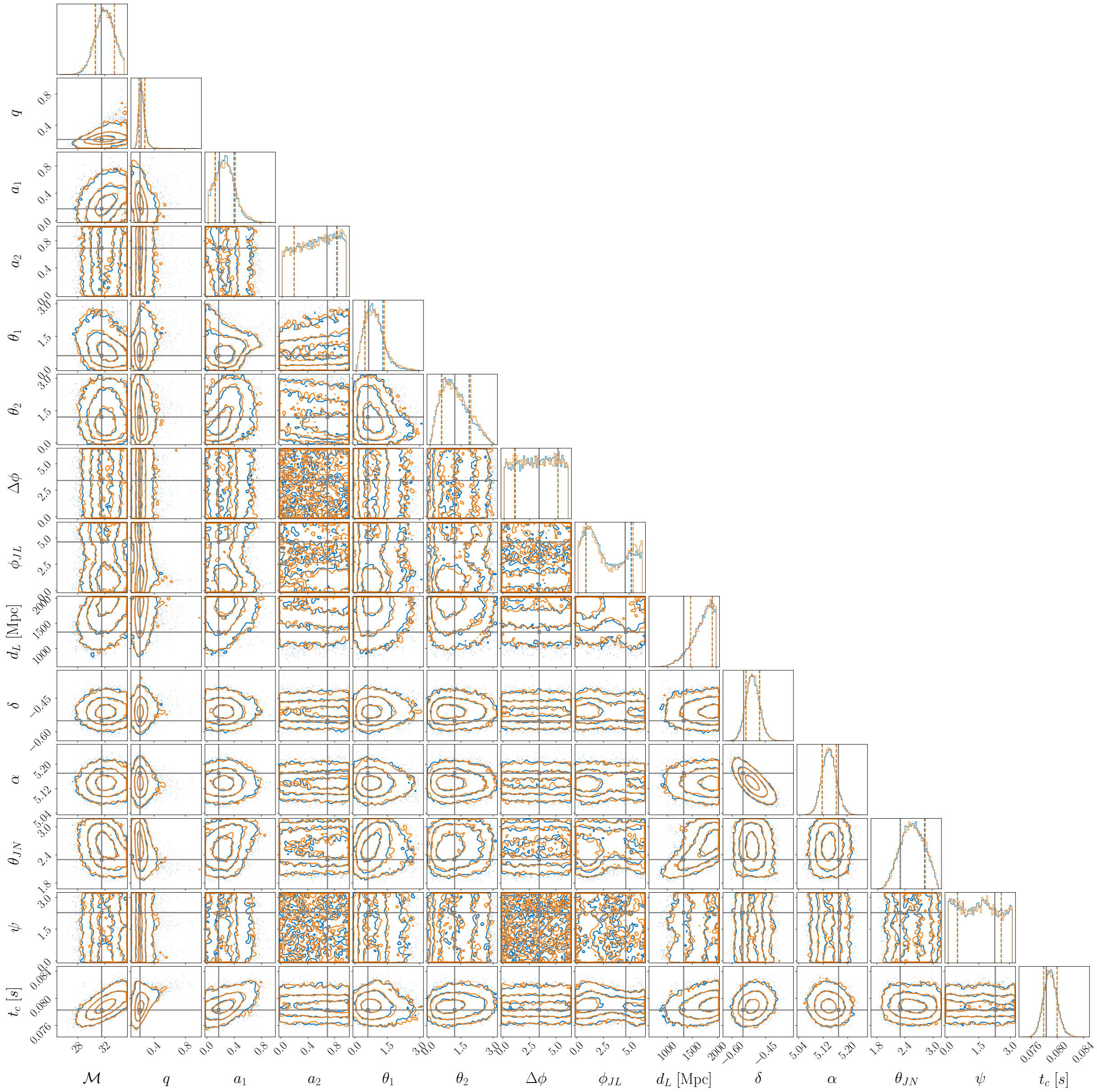


Figure 4.6: Corner plot comparing the posterior distributions produced with `dynesty` (blue) and our sampler `nessai` (orange) for an injection with an optimal network SNR of 15.54. The phase is marginalized and remaining 14 parameters are shown, see appendix D.2 for details about the parameters. The injected value is indicated by the cross-hairs in each subplot and the respective 16% and 84% percentiles are also shown in the 1-dimensional marginalized posteriors.

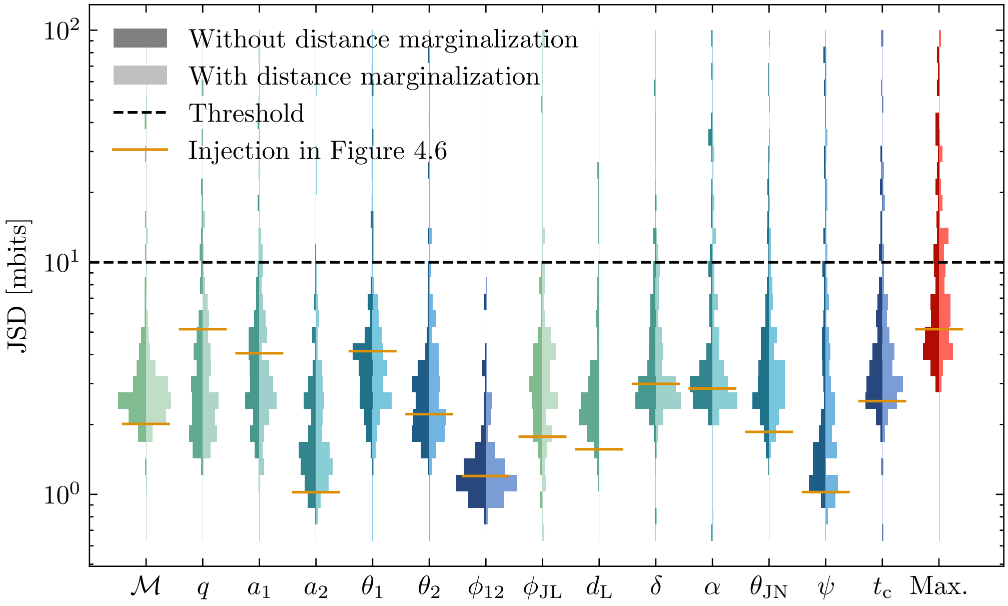


Figure 4.7: Distribution of JSDs in mbits between the 1-dimensional marginal posterior distributions obtained using `dynesty` and `nessai`. The JSD reported is the median value of 100 random subsets with 1000 samples each per parameter. Results are shown per parameter and for the runs with (left-hand side) and without (right-hand side) distance marginalization. The maximum JSD over all parameters per injection is also show in red. The dashed lined denotes the 10 mbits threshold determined using the method described in Ashton and Talbot [233] and the orange line over each distribution shows the JSD between the posterior distributions show in fig. 4.6.

there is no ground truth for these analyses and both samplers pass the P-P test we cannot determine which is 'correct' and which is biased.

4.5.3 Parallelization of the likelihood computation

Our sampler is designed such that candidate live points are drawn simultaneously in the population stage. This allows for simple parallelization of the likelihood computation, since the pool of candidate live points can be distributed over a number of threads and likelihood values computed and stored until needed for the proposal stage. In fig. 4.8, we compare the run-time and time spent evaluating the likelihood for the same injection using an increasing number of threads for the likelihood computation. We use an additional thread for the main sampling process. The time spent evaluating the likelihood is inversely proportional to the number of threads allocated, although the overall run-time is not. With a single thread it accounts for 54% of the total run-time and this decreases to 9% when using 16 threads. As mentioned previously, there is a cost associated with the population stage and further smaller cost associated with training, for this injection these are 36% and 9% respectively. These remain approximately constant when increasing the number of threads available and act as a lower limit on the theoretical minimum run-time, this

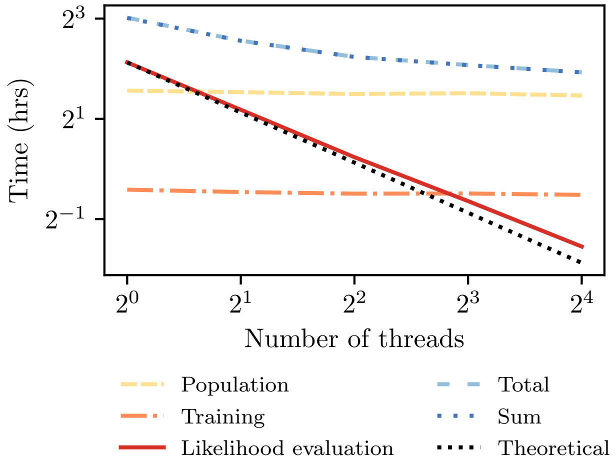


Figure 4.8: Comparison of the total time (in hours) spent on each stage of the algorithm for an increasing number of threads for a single injection with a fixed noise seed. The time spent evaluating the likelihood decreases as the number of threads increases, the theoretical reduction is shown in black. The costs of training and population stages remain approximately constant, as such they act as a lower bound on the minimum run-time. The sum of the time spent on likelihood evaluation, training and population is approximately equal to the total time spent sampling, indicating minimal overhead.

is shown in fig. 4.8. The remaining $< 1\%$ of the run-time is general overhead associated with running the sampler.

4.5.4 Diagnostics

As mentioned previously, there are various challenges when implementing a sampling algorithm. `nessai` is designed to sample from within the constrained prior, in this case care must be taken to ensure that the prior is not over-constrained since this will lead to regions of parameter being under-sampled which in turn will bias the results. There are also specific problems that arise from the nature of the parameter space, such as multimodality and correlations. We use a series of diagnostics to identify possible problems during sampling. In section 4.5.5, we also discuss how some of these diagnostics can be used to tune the sampler settings described in section 4.2 and appendix D.3.

We use the cross-checks proposed in [218] as a heuristic for determining if the nested sampling algorithm has converged without over or under-constraining the posterior distributions. These checks rely on order statistics and the assumption that new live points should be inserted uniformly into the existing live points which allows for a p-value to be computed using a Kolmogorov-Smirnov test (KS test) [219, 220] with the additional consideration that the underlying distribution is discrete [221]. In fig. 4.9, we show an example of the distribution of the indices of newly inserted live points and in fig. 4.10 we show the p-values computed every N_{live} iterations. The histogram shows the final dis-

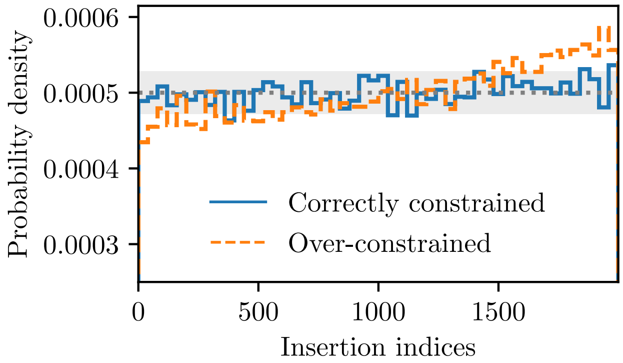


Figure 4.9: Example of the distribution of insertion indices for two nested sampling runs with 2000 live points. Uniformly distributed indices indicate no under- or over-constraining and deviations from uniformity indicate the opposite. The result with an orange dashed line shows over-constraining and the result with a solid blue line shows the correctly converged run. The shaded region indicates the 2-σ errors on the expected distribution.

tribution of insertion indices for all the nested samples, this may not highlight specific problematic regions of the parameter space but if it is not uniform, it is a clear indication that the sampler is consistently over- or under-constraining. If the distribution of p-values in fig. 4.10 is non-uniform then this is another clear indication of problems during sampling.

The acceptance is another important statistic to monitor during sampling since we aim to develop a more efficient sampler. There are two acceptances we can monitor in `nessai`, the proposal acceptance (the acceptance based on the likelihood threshold) and the population acceptance (the acceptance of the rejection sampling step). The first has a direct effect on the number of likelihood evaluations whilst the second only affects the total run-time, both quantities are shown in fig. 4.10. This figure also highlights how periodically retraining the normalizing flow leads to an increase in the proposal efficiency. It also shows how the population process is typically inefficient which explains why on average 40% percent of the total run-time is spent on the population stage.

We also track the minimum and maximum log-likelihoods, number of log-likelihood evaluations, log-evidence and fractional change in evidence. The combination of these statistics allows the user to quickly understand the current state of the sampler and identify potential issues such as plateaus in the likelihood space and regions which are inefficient to sample. The complete set of statistics is shown in fig. 4.10.

4.5.5 Tuning `nessai`

`nessai` includes various settings, a comprehensive list and description of each can be found in the documentation [429]. In practice, we find that a small subset of the settings predominantly determine whether the algorithm converges without any bias. We use the

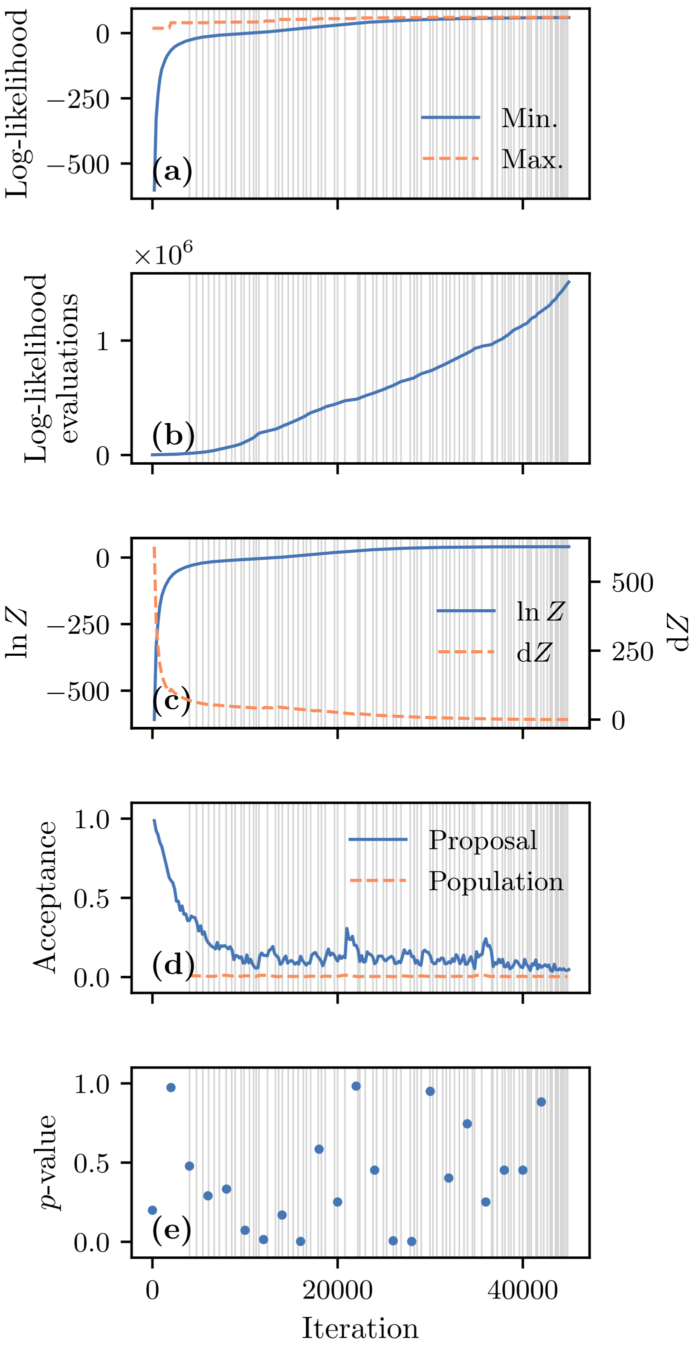


Figure 4.10: Example of the state plot produced by our sampler that tracks statistics as a function of sampling iteration: **(a)** minimum (blue solid) and maximum (orange dashed) log-likelihood, **(b)** cumulative number of likelihood evaluations, **(c)** log-evidence $\log Z$ (blue solid) and fractional change in evidence $\mathrm{d}Z$ (orange dashed), **(d)** proposal (blue solid) and population (orange dashed) acceptance and **(e)** p-value for cross-checks of every N_{live} live points. The iterations at which the normalizing flow is trained are indicated with vertical lines, for this injection these total 87.

validation method described in section 4.5.1 and the diagnostics from section 4.5.4 to understand how these settings affect convergence.

As expected, the number of live points N_{live} is an important setting but it is even more crucial in `nessai` since it limits the amount of training data available. We find that a minimum of 1000 live points is required and for more complex problems, such as gravitational-wave inference, at least 2000 live points should be used.

There are a large number of settings which relate to the complexity of the normalizing flow. While tuning the sampler we found that the number of coupling transformations greatly affected convergence. If too many transforms were used the algorithm was prone to over-constraining the posterior distribution. We attribute this to the complexity of the likelihood contour learnt by the flow, if the flow has too many trainable parameters it can over-fit the distribution and exclude regions of the parameter space which should be sampled. At the other extreme, if the model is too simple, then the resulting contour can "smooth" fine details and more samples are drawn outside the initial likelihood constraint. These will not be accepted and the sampling process is therefore less efficient. We use a similar logic for the number of neurons and layers in the neural network that parameterizes the flow but we find that these parameters predominantly affect training time with a lesser effect on overall convergence. Another parameter that is important to consider is the batch size, during sampling the normalizing flow can be training upwards of 100 times. Hence, a larger batch size is recommended since it can greatly reduce training time, we also recommend increasing the batch size when using reparameterizations that increase the amount of training data, such as the boundary inversion described in section 4.2 and appendix D.1.

We note that the size of the pool of new samples affects the efficiency of the algorithm and the total run-time. If the pool-size is small, then the normalizing flow is frequently retrained, in the extreme case where the proposal is inefficient due to, for example, the complexity of the parameter space, then the normalizing can be retrained multiple times during a single iteration. Conversely, if the pool-size is large then if the flow is force-ably retrained a number of points are discarded or, if the flow is only retrained once the pool is empty, then the rejection sampling becomes in-efficient since a large fraction of the potential new points will lie outside the likelihood bound. We instead opt to inversely scale the pool-size given the mean acceptance of the sampler since the last iteration the flow was trained. We recommend setting the base pool-size to the number of live points, only retraining the model when the pool is empty and setting the maximum pool-size to be ten times the base pool-size. We use these settings for the results in section 4.5 and find that this results in a median of 263 training instances required to reach convergence.

As mentioned previously, approximately 40% of the run-time is spent on populating the pool of new samples. This is directly attributable to the efficiency of the rejection sampling required to ensure samples are distributed according to the prior. In section 4.1.1 we propose two methods for drawing samples within the contour in the latent space, these produce uniformly and normally distributed samples respectively. In practice, we find the two methods comparable in most cases except for when the latent radius lies in the tail of the χ-distribution that corresponds to the latent prior $p_{\mathcal{Z}}$. In this case, using the uniform distribution results in lower population and proposal acceptances, which leads to longer run-times.

4.6 Conclusions

We have proposed a novel method for sampling within a given likelihood contour according to the prior that can be incorporated into the standard nested sampling algorithm. Our method employs normalizing flows to learn the density of the current set of live points, which, once trained, allows us to produce points within the contour by sampling from a simple distribution and using rejection sampling. The use of normalizing flows allows us to avoid using multiple bounding distributions and since new samples are independent of the previous samples we eliminate the need to use a random walk. We implement this proposal method in our sampler, `nessai`, and conduct a series of tests to verify that it recovers the correct Bayesian posteriors and then compare our results to those obtained with another sampler to determine if our design does in fact result in a more efficient sampler.

We apply our sampler to 128 four-second duration simulated signals from the coalescence of binary black hole systems sampled at 2048 Hz and we run two separate analyses, one with distance marginalization and another without. The resulting P-P plots (fig. 4.3) show that our sampler more reliably recovers the posterior distributions with distance marginalization than without, however both pass the P-P test. This indicates that our proposal method does not introduce any inherent biases.

We use `dynesty` for the comparison, which has been shown to produce results consistent with those used in previous LVK analyses [122]. We find that our sampler returns evidences consistent with `dynesty`, which serves as further verification of our results. Since we aim to produce a more efficient sampler, we also compare the likelihood evaluations required to reach convergence. When not using distance marginalization, we find that `nessai` requires 5.04×10^6 likelihood evaluations, 2.07 times fewer than `dynesty`. When distance marginalization is enabled `nessai` requires 7.22×10^6, which, whilst still 1.34 fewer than `dynesty`, is more than with the marginalization disabled. We revisit this in the following chapter.

However, this reduction in likelihood evaluations does not relate directly to the total computation time because of the additional costs associated with sampling, which for nessai are associated with training the normalizing flow and populating the pool of new samples. We find that the fraction of the time spent on each stage changes when using distance marginalization. Without the marginalization, on average, 8% of the total computation time is spent on training and a further 40% on population. When using distance marginalization, this changes to 5% spent on training and 42% on population. We attribute the difference in population time to the efficiency of the rejection sampling, which is improved when including the reparameterization for distance discussed in section 4.3.1. We find that without distance marginalization, the median run-time for nessai is 2.32 times faster than dynesty. However, when distance marginalization is enabled we observe that, on average, nessai is only 1.40 times faster than dynesty. This further reinforces our recommendation to use nessai with distance marginalization disabled.

We also show how our sampler can make use of parallelized likelihood functions by evaluating the likelihood of new live points during the population stage. We repeat the previous analysis for a single injection without distance marginalization and parallelize the likelihood computation with increasing number of threads up to 16. We observe that the reduction time evaluating the likelihood does not quite match the theoretical values, indicating that there is a small overhead associated with it. This also highlights how the limiting factor is the time spent training the normalizing flow and populating the pool of new live points.

To aid in diagnosing potential biases during sampling, we include a series of diagnostics in our sampler which allow us to easily identify under and over-constraining. These diagnostics also help to tune the sampling settings and highlight how periodically retraining the normalizing flow during sampling prevents the proposal from becoming inefficient during sampling.

We find that our algorithm is susceptible to under-sampling regions of the parameter space which are close to the prior bounds. We consequently introduce the previously described reparameterizations to mitigate this, and a series of diagnostics to aid in diagnosing biases and correctly tuning the settings. We aim to address this in further work with changes to the design of the normalizing flows we have used.

It is natural to compare this work to simulation-based inference approaches [234, 411, 412, 416] which use VAEs and normalizing flows to produce posterior distributions. Our approach differs from these in that it requires no prior computation since training occurs during sampling and we do not introduce any assumptions about the data other than those necessary to apply a nested sampling algorithm. nessai is therefore a drop-in replacement for existing sampling algorithms that does not require changes to existing pipelines.

In future work we aim to evaluate our sampler using more expensive waveform models including those for longer duration signals, such as those from binary neutron star (BNS) or neutron star-black hole (NSBH) systems, and models which include higher-order modes.

We will also investigate the suitability of other types of normalizing flow transforms, such as the spline based transforms from Durkan et al. [370] and flows which allow for specifying a manifold [360]. These changes could improve the efficiency of the population stage, which is currently the slowest part of the algorithm. Another possible approach for reducing the cost of population is using alternative reparameterizations for parameters such as the spin magnitudes, which we observe to be two of the most challenging parameters to sample.

In summary, we have proposed a novel variation of the standard nested sampling algorithm that incorporates normalizing flows specifically designed for inference with computationally expensive likelihood functions. We have applied our sampler to the problem of gravitational wave inference and shown that it consistently recovers the Bayesian posteriors distributions and evidences with 2.07 times fewer total likelihood evaluations than `dynesty`, another commonly used sampler, which translates to a 2.32 times reduction in computation time. Our sampler therefore serves as a more efficient drop-in replacement for existing samplers.

Chapter 5

Improvements to nessai

In this chapter, we discuss potential sources of bias in nessai and minor modifications to the algorithm that are designed to avoid them and have the potential to further accelerate the algorithm. We test these modifications on toy examples and repeat the gravitational-wave analyses described in chapter 4.

Section 5.1 describes constant volume mode, a modification to how new samples are drawn in nessai that is designed to avoid over-constraining. Section 5.2 discusses biases that may arise from repeatedly training the normalizing flow and means to avoid this. We then test these modifications in section 5.3 using a series of toy examples. Section 5.4 then presents results from repeating the analysis described in chapter 4 using the aforementioned modifications and compares these results to those obtained previously. Finally, section 5.5 summarizes the findings and makes recommendations for what settings to use in analyses.

When applying nested sampling, the most common sources of biases in the results are correlations between subsequent nested samples and over- or under-shrinking the likelihood-constrained prior at a given iteration [200]. These can lead to biases in both the estimated evidences and posterior distributions [218].

In `nessai`, samples are not drawn using a current live point as a starting point, so new samples are not directly correlated to existing samples. However, these samples are used to train subsequent normalizing flows, so there are instead correlations between subsequent distributions of samples. Additionally, if the radius used to construct the latent contour used for drawing new samples is too small, this can lead to likelihood contours being over-constrained. In both cases, these issues can be diagnosed using the insertion indices as described in chapter 4. We now discuss specific methods for combating both these sources of potential biases in `nessai` and present results demonstrating these methods. We then repeat the analysis from chapter 4 and compare the results.

5.1 Constant volume mode

In the original version of `nessai` [1], the radius of the latent contour was determined based on the worst point by mapping it to the latent space. This relied on the point being mapped to a region of low-probability (larger radius) in the latent space. However, since the flow is trained with samples rather than the likelihood of the points, the regions with the lowest probability in the latent space (largest radius) will not always correspond to regions with the lowest likelihood. This can result in the worst point mapping to a small radius.

In practice, we observe that this leads to large variations in the radius used between training, as shown in fig. 5.2. Whilst this is not inherently an issue, if the volume of the latent contour is consistently underestimated, then this can lead to the likelihood-constrained prior being over-constrained. This results in an over-estimate of the final evidence and an over-constrained posterior distribution. The effect is most noticeable for posterior distributions where the true posterior should be uniform but the estimated posterior is under-sampled at the prior bounds. An example of this is shown in fig. 5.1 which shows an over-constrained run alongside increasingly well-sampled runs and is discussed further in section 5.3.1.

Originally, we counteracted this by including a *fuzz factor* (ϵ_{FF}) that increased the volume of the latent contour, by reparameterising certain parameters, such as those we expect to have broadly uniform posterior distributions, and by extensively tuning the sampling settings. To avoid this and make `nessai` more reliable, we introduce an alternative to using the worst point, which we call constant volume mode.

In constant volume mode, the volume of the latent contour, and therefore radius r, is

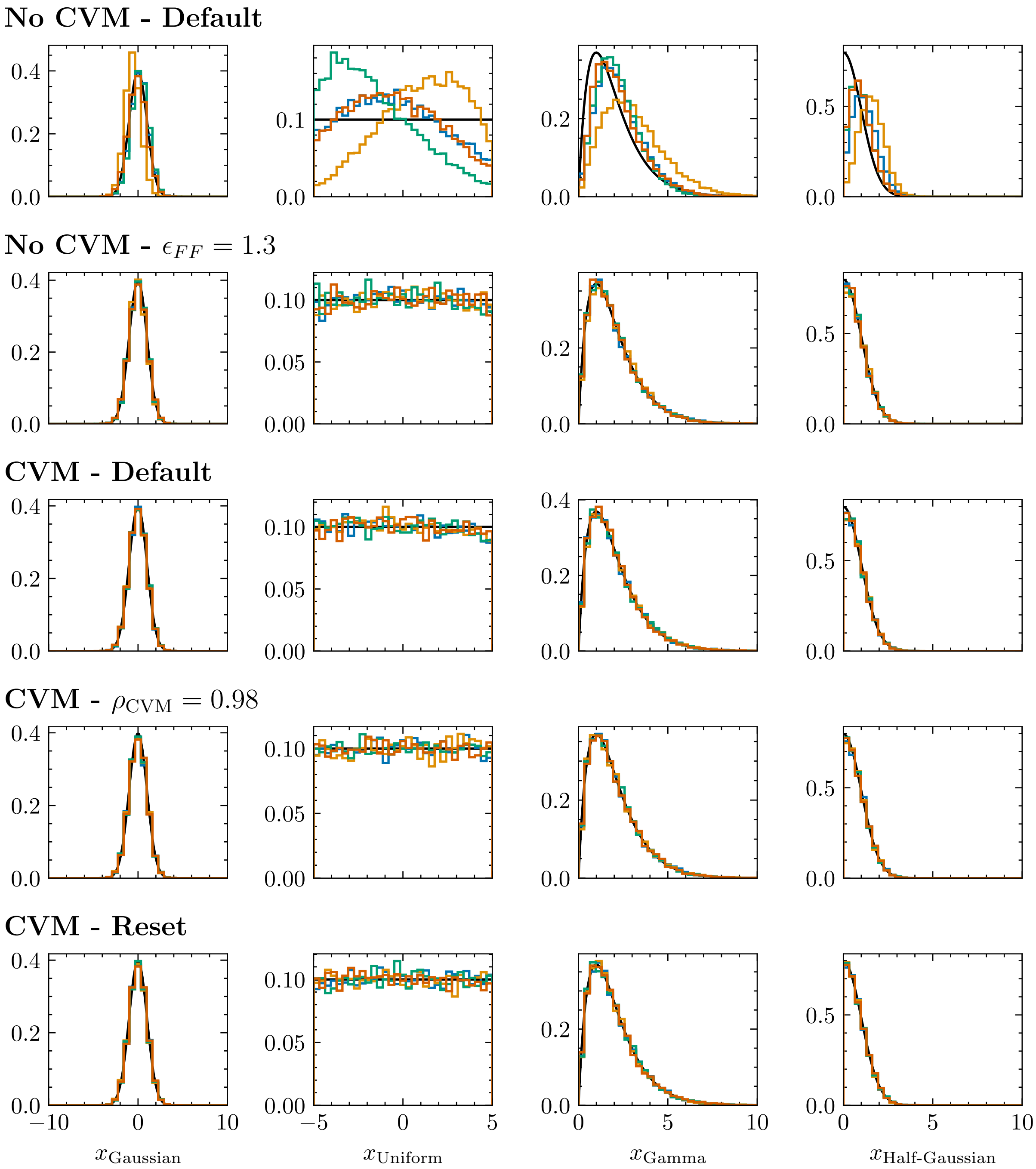


Figure 5.1: Posterior distribution for 16-dimensional likelihood described in appendix A.3 with four Gaussian parameters, four Half-Gaussian parameters, four Gamma distribution parameters and four uniform parameters. The true posterior distributions are shown with solid black lines, the results obtained using nessai with different settings are shown on each row. From top to bottom they are: without constant volume mode with the default fuzz factor ϵ_{FF}, without constant volume mode with the fuzz factor manually specified to $\epsilon_{\text{FF}} = 1.3$, with constant volume mode and the default value $\rho_{\text{CVM}} = 0.95$, with constant volume mode and $\rho_{\text{CVM}} = 0.98$, and with constant volume mode, $\rho_{\text{CVM}} = 0.95$ and resetting the flow every fourth time it is trained.

fixed for the entire sampling run which avoids variations in the latent radius. Since, for an n-dimensional Gaussian, the distribution of the radius follows a χ-distribution, this corresponds to truncating the χ-distribution at a specific value of r. The value is determined using the inverse cumulative density function (CDF) and a user-defined fraction defined on $\rho_{\mathrm{CVM}} \in (0, 1)$, which defaults to $\rho_{\mathrm{CVM}} = 0.95$. This method is tested in section 5.3.1

5.2 Correlations between subsequent normalizing flows

We tested various different methods for determining when to train the normalizing flow when sampling with `nessai`, but ultimately we found that training when the pool of new points is depleted to be the most reliable. This is the method that was used in chapter 4 and, whilst it can lead to the flow being frequently retrained, this is often preferable since populating the pool of points with a flow trained on a previous set of live points will lead to an inefficient proposal if using constant volume mode, since the same latent radius will be used.

At a given iteration, a flow is trained with the current live points. By design, these live points will be either draws from the previous flow or samples that were used to train the previous flow. Therefore, the weights of the previous flow should be a reasonable starting point for training the next flow. This is the method used in the original version of `nessai` and it has the advantage of speeding up training since the flow is not starting from scratch each time it is trained. The downside to this approach is that if the previous flow is poorly trained, for example stuck in a local minimum, then the subsequent flow will be starting from a 'bad' starting point and may fail to recover.

In the context of `nessai`, this can translate to the previous flow over- or under-constraining the likelihood contour and the subsequent flow showing the same bias. Over-constraining arises in the final result as a bias in the evidence and, often, missing modes in the posterior if it is multimodal. Under-constraining is less apparent in the final result, and instead is more noticeable when examining the efficiency of the run, since it will lead to more proposed points being rejected. Both cases can be easily mitigated by periodically training the flow from scratch rather than using the previous flow as a starting point.

5.3 Results

We present results comparing runs without and without constant volume mode and resetting the normalizing flow. Code to reproduce all the results is available at [438] and uses the likelihoods implemented in `nessai-models` [439].

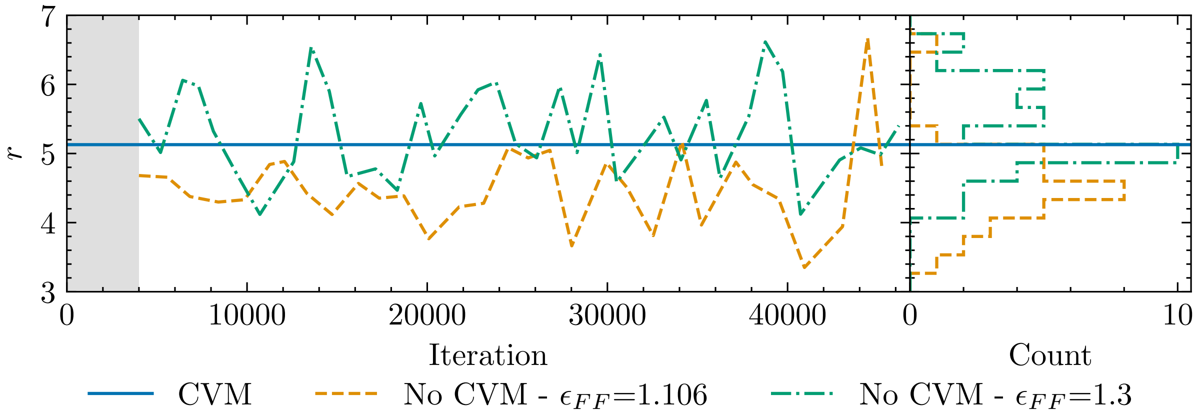


Figure 5.2: Radius of the latent distribution for three nested sampling runs with `nessai`: one with constant volume mode and two without. The two runs without constant volume have different values for the fuzz factor, the first has the default which in 16 dimensions is 1.106 and the second has the value manually set to 1.3. The shaded region shows the iterations before the normalizing flow was trained.

5.3.1 Constant volume mode

To test constant volume mode, we use a 16-dimensional likelihood where the likelihoods is the product of 16 independent distributions, each corresponding to one dimension. These are: four Gaussian distributions, four Half-Gaussian distributions, four γ-distributions and four uniform distributions. This likelihood contains a mixture of features that will test whether the likelihood contours are over-constrained or not, these include likelihoods that are maximum at the prior bounds and likelihoods that are uniform. More details about the likelihood is provided in appendix A.3.

We perform three runs: one with constant volume mode and two without. For the runs without constant volume mode, we consider two values for the fuzz factor ϵ_{FF}: the first is the default that is determined based on the expansion fraction $\epsilon_{\text{VF}} = 4$, which is related to the fuzz factor by $\epsilon_{\text{FF}} = (1 + \epsilon_{\text{VF}})^{1/n_{flow}}$, and the second we set to $\epsilon_{\text{FF}} = 1.3$.

The radii of the latent distributions for the three runs are shown in fig. 5.2. This highlights how the radius can vary greatly over the course of a run. Increasing the fuzz-factor can help offset this, but it does noes entirely prevent small radii and also has the downside of decreasing the rejection sampling efficiency when larger radii are used. Both small and large radii are avoided with constant volume mode which leads to more efficient sampling.

The posterior distributions produced by the four runs are shown in fig. 5.1. These highlight how over-constrained likelihood contours can lead to biased posterior distributions, which are most noticeable when the default the fuzz-factor is used (top row). It also shows how this effect is most apparent for parameters where the likelihood contour does not shrink, like the uniform parameters (second column), or where the posterior rails

Table 5.1: p-values computed between the true marginal posterior distributions and estimated distributions for the 16-dimensional mixture of distributions described in section 5.1. Results are shown for five runs with different settings, from left to right without constant volume mode with the default fuzz factor ϵ_{FF}, without constant volume mode with the fuzz factor manually specified to $\epsilon_{\mathrm{FF}} = 1.3$, with constant volume mode and the default value $\rho_{\mathrm{CVM}} = 0.95$, with constant volume mode and $\rho_{\mathrm{CVM}} = 0.98$, and with constant volume mode, $\rho_{\mathrm{CVM}} = 0.95$ and resetting the flow every fourth time it is trained. The combined p-value for each run is also shown in the final row.

Parameter	No CVM	No CVM - $\epsilon_{\mathrm{FF}} = 1.3$	CVM	CVM - $\rho_{\mathrm{CVM}} = 0.98$	CVM reset
Gaussian 1	0.00	0.55	0.57	0.03	0.81
Gaussian 2	0.00	0.00	0.25	0.15	0.11
Gaussian 3	0.00	0.24	0.56	0.60	1.00
Gaussian 4	0.00	0.67	0.38	0.47	0.23
Uniform 1	0.00	0.31	0.00	0.61	0.33
Uniform 2	0.00	0.08	0.00	0.02	0.74
Uniform 3	0.00	0.00	0.06	0.99	0.38
Uniform 4	0.00	0.15	0.06	0.96	0.17
Gamma 1	0.00	0.13	0.49	0.55	0.92
Gamma 2	0.00	0.00	0.00	0.72	0.52
Gamma 3	0.00	0.04	0.32	0.55	0.19
Gamma 4	0.00	0.16	0.10	0.94	0.37
Half-Gaussian 1	0.00	0.36	0.09	0.00	0.28
Half-Gaussian 2	0.00	0.00	0.00	0.01	0.26
Half-Gaussian 3	0.00	0.03	0.00	0.75	0.91
Half-Gaussian 4	0.00	0.25	0.23	0.56	0.88
Combined p-value	0.00	0.00	0.00	0.01	0.64

up against the prior bounds, like the half-Gaussian parameters (fourth column). Enabling constant volume mode or increasing the fuzz factor both significantly reduce the biases in the final posterior distributions, as seen in rows two and three of fig. 5.1. However, if we compute p-values between the posterior samples and the true posterior distributions, shown in table 5.1, we see that there are still biases, most notably in the uniform and half-Gaussian distributions. Increasing the volume fraction used in constant volume mode, row 4 four in fig. 5.1, further reduces the biases in the posteriors but the combined p-value in table 5.1 is still below the nominal threshold of 0.05, which we attribute to the half-Gaussian parameters being undersampled at the prior bounds.

These results demonstrate how constant volume mode can mitigate over-constraining but additional adjustments to the settings may be required to sample particular distributions, such as those that have maximum posterior probability at the priors bounds. In the following section, we repeat this test but periodically reset the normalizing flow and show how this can be used alongside constant volume mode.

5.3.2 Resetting the normalizing flow

We test three applications of resetting the flow. First, we repeat the analysis of the 16-dimensional mixture of distributions from the previous section, then we analyse the 8-dimensional Rosenbrock likelihood as defined in appendix A.1. Finally, we analyse a 50-dimensional Gaussian likelihood with uniform priors on $[-10, 10]^n$. In all three cases, we enable constant volume mode with the default value $\rho_{\mathrm{CVM}} = 0.95$ and in the two new tests we perform one run without resetting the flow and another where the flow is reset every 4th time it trains.

Results for the 16-dimensional mixture of distributions are presented in the final row of fig. 5.1 and are almost indistinguishable from the results obtained without resetting the flow with $\rho_{\mathrm{CVM}} = 0.98$. However, the p-values for each marginal posterior distribution in table 5.1 show that this change has addressed the remaining biases, and the combined p-value now passes. This demonstrates that in some scenarios, it is better to periodically reset the flow instead of increasing ρ_{CVM}.

The Rosenbrock likelihood is challenging to sample from due to its 'banana-like' shape and multiple modes [440], which can often lead to over-constraining. Figure 5.3 shows the 1- and 2-dimensional posteriors obtained when sampling with `nessai` with and without resetting the normalizing flow. This demonstrates how the run where the flow is not reset is over-constrained; it misses part of the posterior mass in the second mode and the long tail that is present in the later dimensions. This bias is also present in the insertion indices, which have a corresponding p-value of 0.0038. Since the initial result is over-constrained, it is to be expected that the run where the flow is reset will require more likelihood evaluations, however, the training time also increases since the flow takes longer to train when starting from random weights. In this particular example, the run time doubles from 3.25 minutes to 6.5 minutes. This is the main disadvantage to resetting the flow: it can increase the wall time.

Since the evidence for a Gaussian likelihood is analytic, we can compare the estimated evidences to the true value, which is $\ln Z = -50 \times \ln 20 \approx -149.79$. These results are shown in table 5.2 alongside the posterior effective sample size, the number of likelihood evaluations and wall time. From these results, we see that the run where the flow is not reset significantly over-estimates the log-evidence, i.e. it is over-constrained. On the other hand, the log-evidence estimate for the run where the flow is periodically reset agrees with the analytic value. The number of likelihood evaluations required is consistent with the first run being over-constrained. However, despite requiring half as many likelihood evaluations, the run without resetting is 50% slower than the run with it. This is in part due to using a vectorized likelihood that accounts for less than 0.1% of total wall time, so the cost is dominated by rejection sampling. We present the rejection sampling acceptance as a function of iteration in fig. 5.4, this shows how resetting the flow results

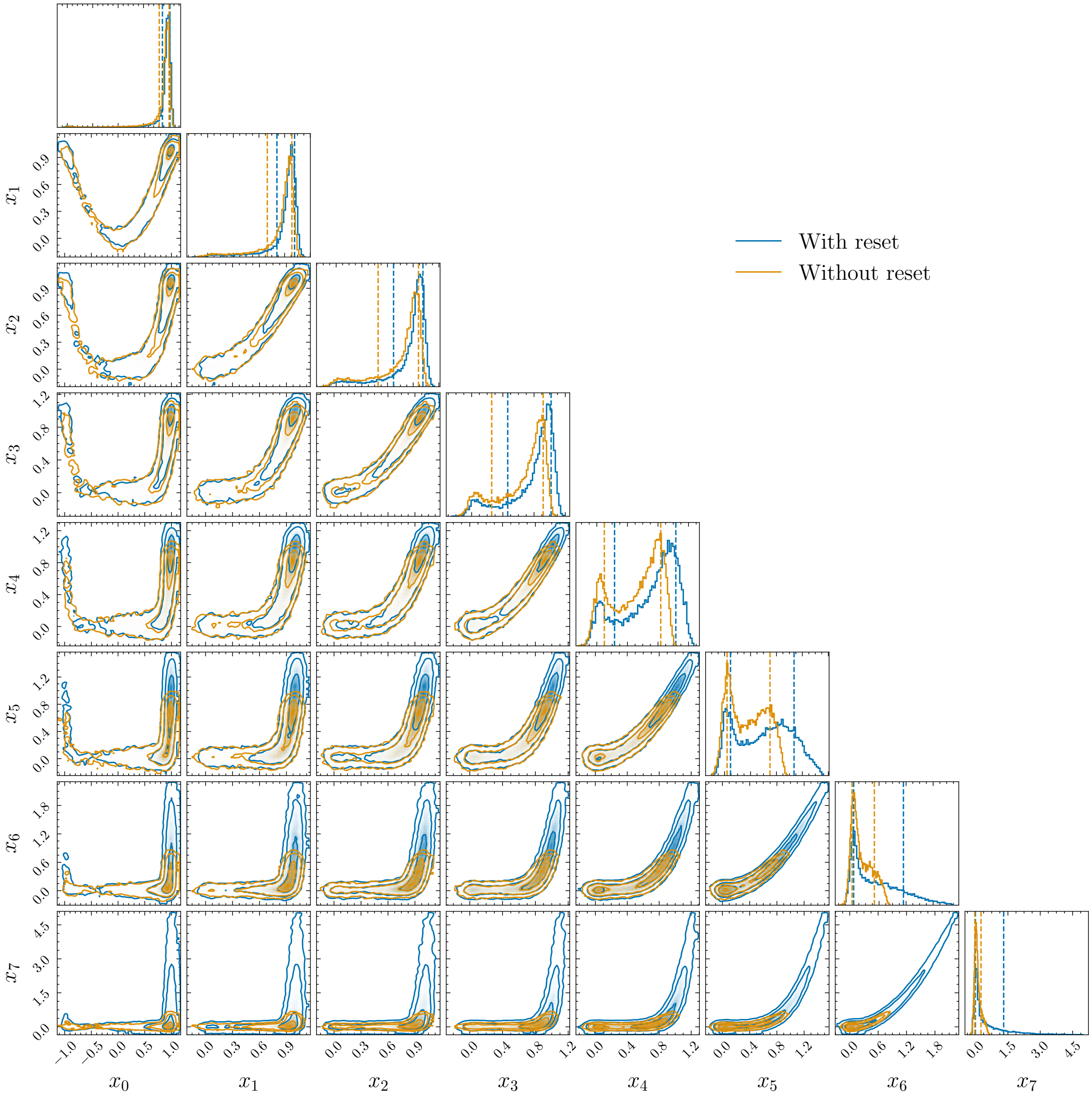


Figure 5.3: Corner plot showing the posterior distribution for the 8-dimensional Rosenbrock sampled using `nessai` with (blue) and without (orange) resetting the normalizing before training. The 1-σ confidence intervals are shown alongside the marginal histograms.

Table 5.2: Comparison of the results obtained when sampling a 50-dimensional Gaussian with and without resetting the normalizing flow between training.

	$\ln Z$	ESS	Likelihood evaluations	Wall time [min]	Likelihood time [sec]
Analytic	-149.79	-	-	-	-
No resetting	-147.78 ± 0.20	19 868	31 376 997	889	10.6
Reset every 4	-149.93 ± 0.20	19 840	62 534 452	510	21.7

in significantly higher acceptance, which explains the difference in wall times. Figure 5.4 also shows that training becomes less frequent when the flow is not reset, which is again consistent with over-constraining: if the contour is over-constrained, then fewer samples will be rejected by the likelihood threshold and the flow will be trained less frequently.

The results in fig. 5.4 also highlight a limitation of the normalizing flows being used in `nessai`. After first training the flow, the sampling efficiency given the likelihood threshold increases, however the rejection sampling is significantly less efficient than the later stages of the run. This is a result of the flow proposing samples that are outside the prior bounds and are therefore rejected, as the likelihood contour shrinks fewer of the proposed samples are outside the prior and the rejection sampling becomes more efficient.

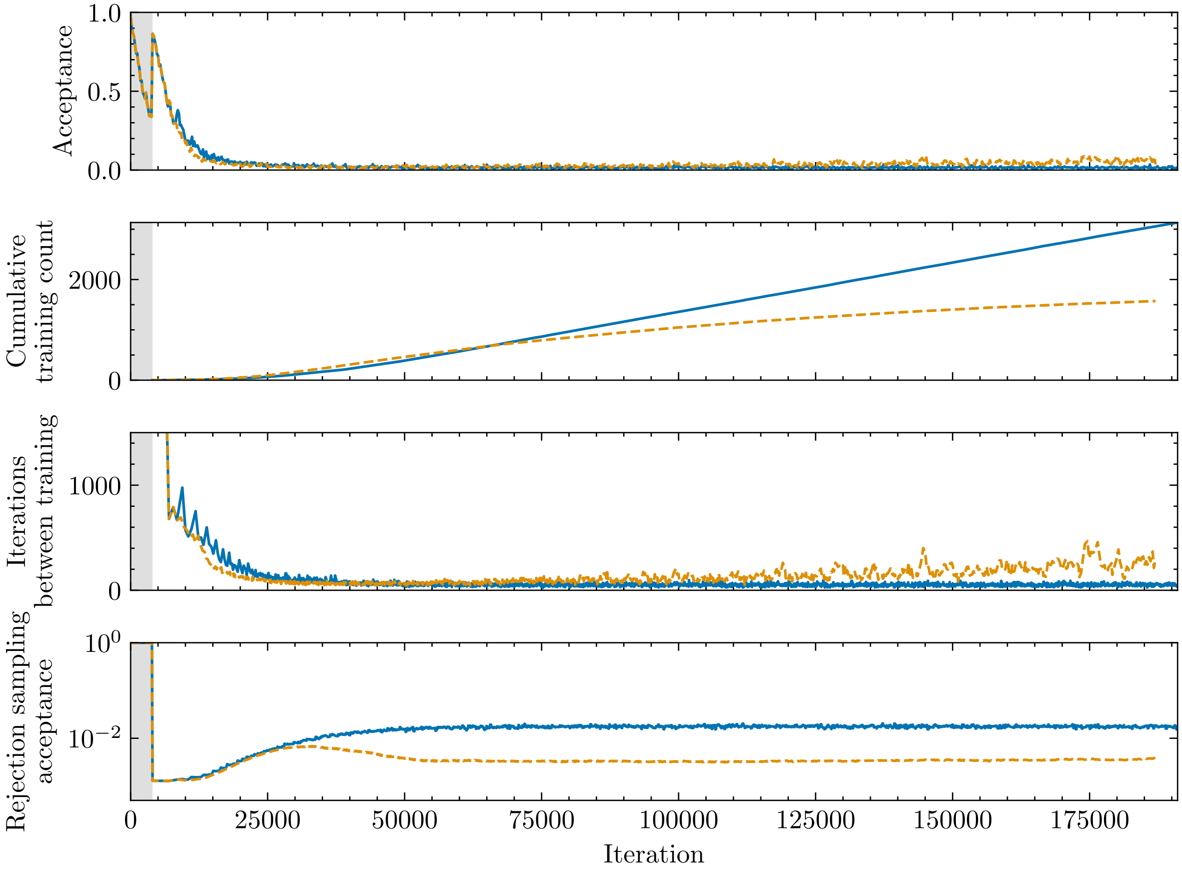


Figure 5.4: Comparison of the sampling efficiency, training frequency and rejection sampling efficiency when sampling a 50-dimensional Gaussian likelihood with (blue) and without (orange) resetting the normalizing flow. The shaded region indicates the iterations before the normalizing flow was first retrained.

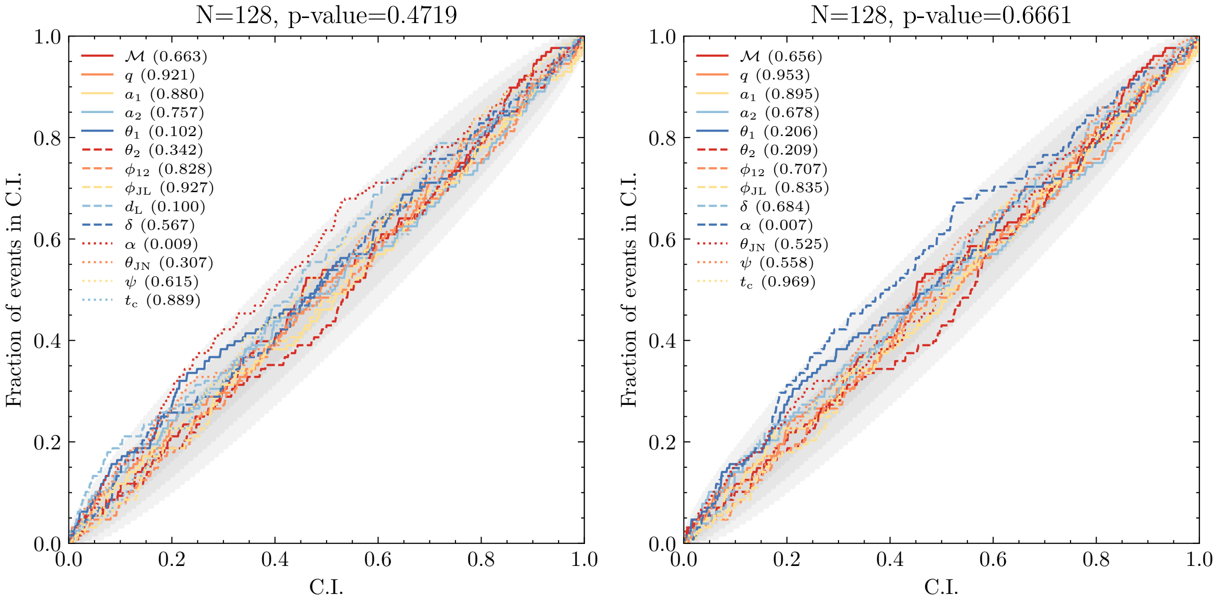
5.4 Improving previous gravitational-wave results

We repeat the analysis of 128 binary black holes (BBHs) performed in chapter 4 with updated settings and compare the results. The main change is enabling constant volume mode with $\rho_{\mathrm{CVM}} = 0.95$. As shown in section 5.1, this can reduce issues during sampling that lead to under-sampling regions near the prior bounds. As a result, we find that the reparameterization introduced for the spin magnitudes a_1 and a_2 is no longer necessary and the default rescaling can be used instead. We keep all the other settings unchanged. The probability-probability plots (P-P plots) for the updated analyses is included in fig. 5.5 and shows that with these settings `nessai` still produces reliable results with and without distance marginalization.

We then compare the number of likelihood evaluations required to reach convergence and total wall time in fig. 5.6. These results show that this simple change improves the sampling efficiency both with and without distance marginalization. With distance marginalization enabled, the number of likelihood evaluations and wall time improve by 2.7 and 3.0 times respectively, whereas without distance marginalization the improvement is only 1.6 and 2.0 time respectively. This translates to an overall average reduction in likelihood evaluations of ~ 6 times compared to `dynesty`.

We also investigate these improvements on a per-injection basis, focusing on the runs with distance marginalization. In fig. 5.7, we present a comparison of change in number of likelihood evaluations and wall time as a function of network signal-to-noise ratio (SNR) for the original results obtained with `nessai` compared to `nessai` with constant volume mode. These results improve for all 128 injections but show that the largest improvements come at larger SNRs. We present similar results comparing both sets of `nessai` run to those performed with `dynesty` in fig. 5.8. These results show that the original version of `nessai` is more efficient than `dynesty` at intermediate SNRs but less efficient and high and low SNRs. In contrast, when constant volume mode is enabled, `nessai`'s efficiency improves at higher SNRs and, as a result, it is more efficient than `dynesty` at high SNRs. However, it remains worse at low SNRs. We now explore these specific runs.

The runs that are less efficient than `dynesty` (i.e. require more likelihood evaluations) are largely below a network SNR of 10 and, as the SNR decreases, they get comparatively slower. We examine the 16 ‘worst’ runs compared to `dynesty`, of which 14 require more likelihood evaluations, and find that they are all highly multimodal likelihoods, particularly in the sky angles (α, δ). This includes some injections with higher SNRs, where the signal lies in a sky location where one of the detectors is insensitive due to the antenna patterns. The nested samples for the 16 runs are presented in fig. 5.9 and show this multimodality. As discussed in section 3.3.7, when normalizing flows are trained on multimodal distributions, the resulting flow has non-zero probability between the modes.



(a) With phase marginalization (b) With phase and distance marginalization

Figure 5.5: P-P plots showing the confidence interval versus the fraction of the events within that confidence interval for the posterior distributions obtained using **nessai** with constant volume mode for 128 simulated compact binary coalescence signals produced with **bilby** and **bilby_pipe**. The 1-, 2- and 3-σ confidence intervals are indicated by the shaded regions and p-values are shown for each of the parameters and the combined p-value is also shown.

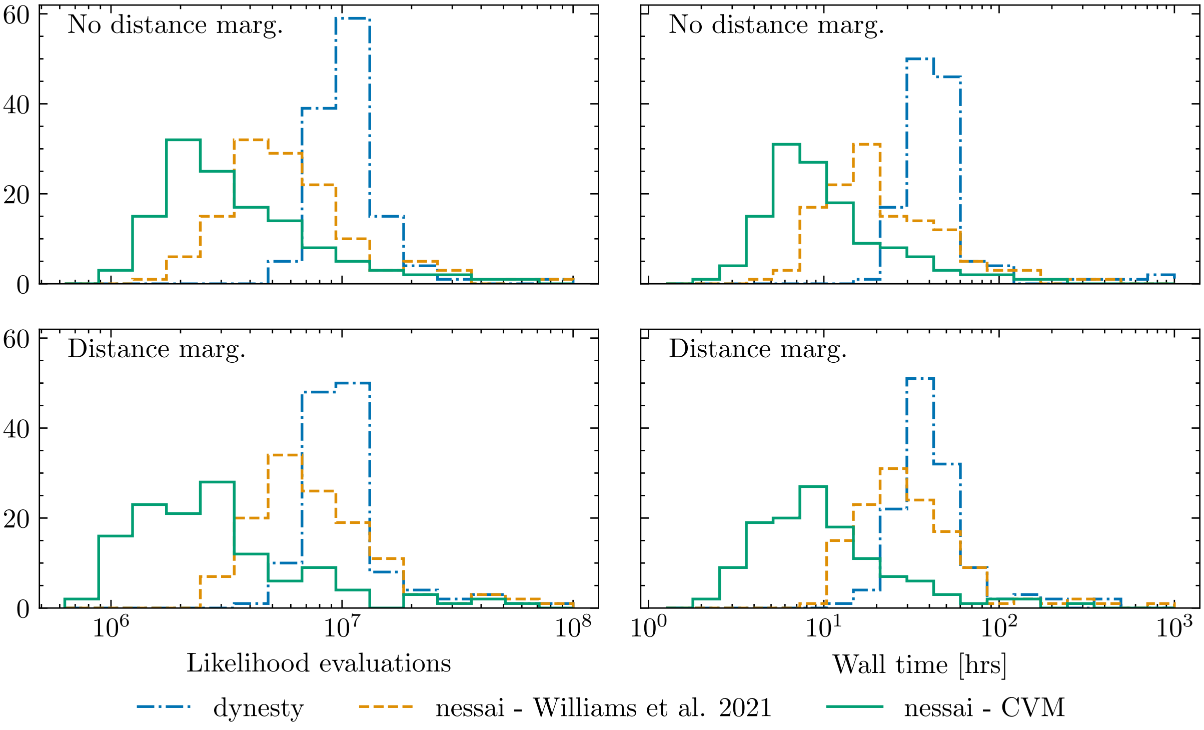


Figure 5.6: Updated version of figure 6 from Williams et al. [1] (fig. 4.5) which includes results obtained using constant volume mode.

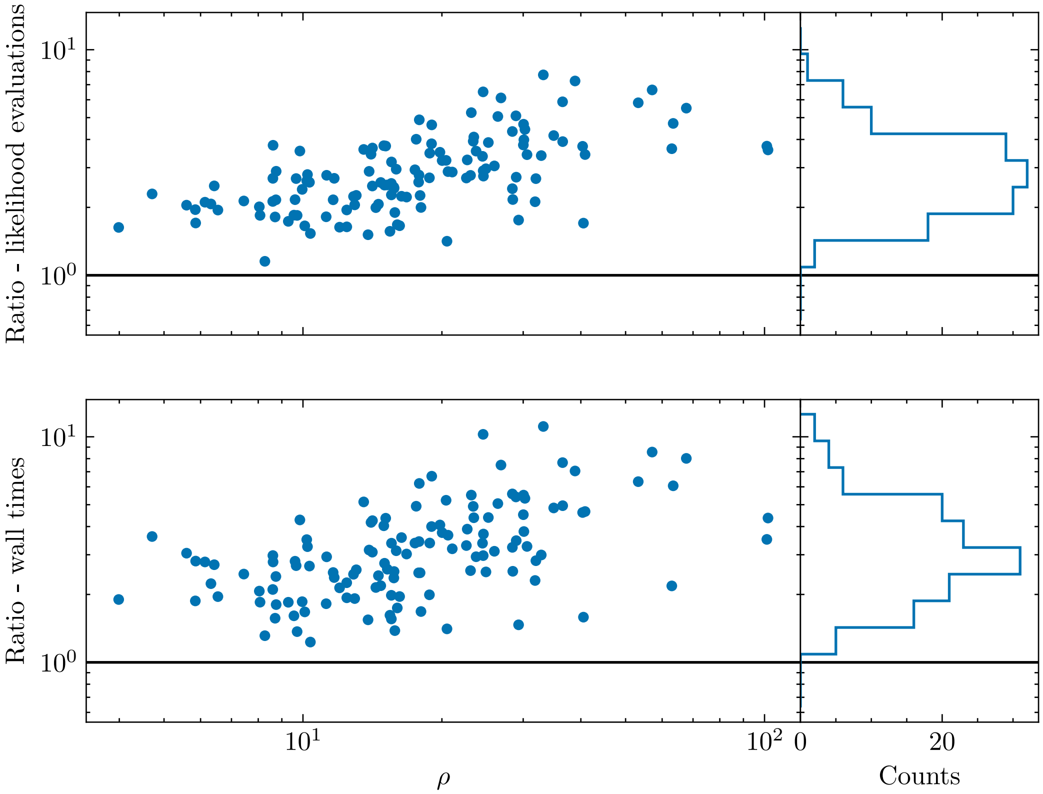


Figure 5.7: Comparison of the number of likelihood evaluations and wall times for `nessai` with and without constant volume mode as function of network SNR. The results shown are for runs with distance marginalization enabled.

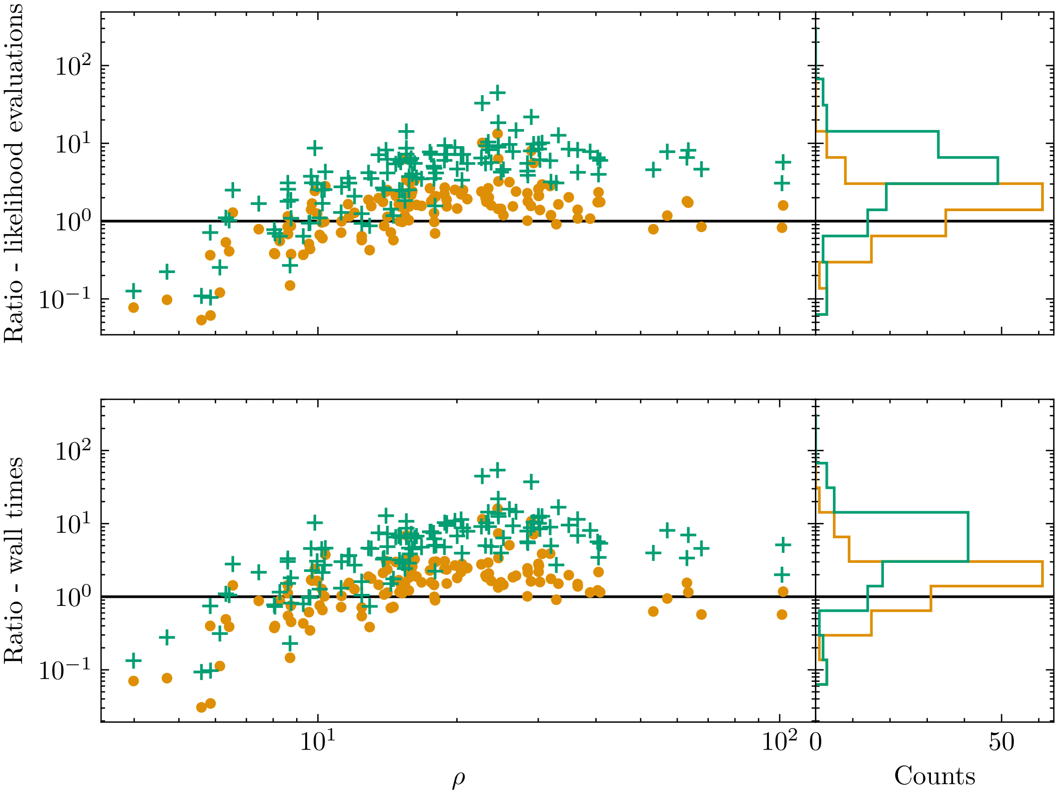


Figure 5.8: Comparison of the number of likelihood evaluations and wall times for `nessai` and `dynesty` with and without constant volume mode as function of network SNR. The orange points show the results for the original `nessai` settings (without constant volume mode) and the green crosses show the results with constant volume mode. The results shown are for runs with distance marginalization enabled.

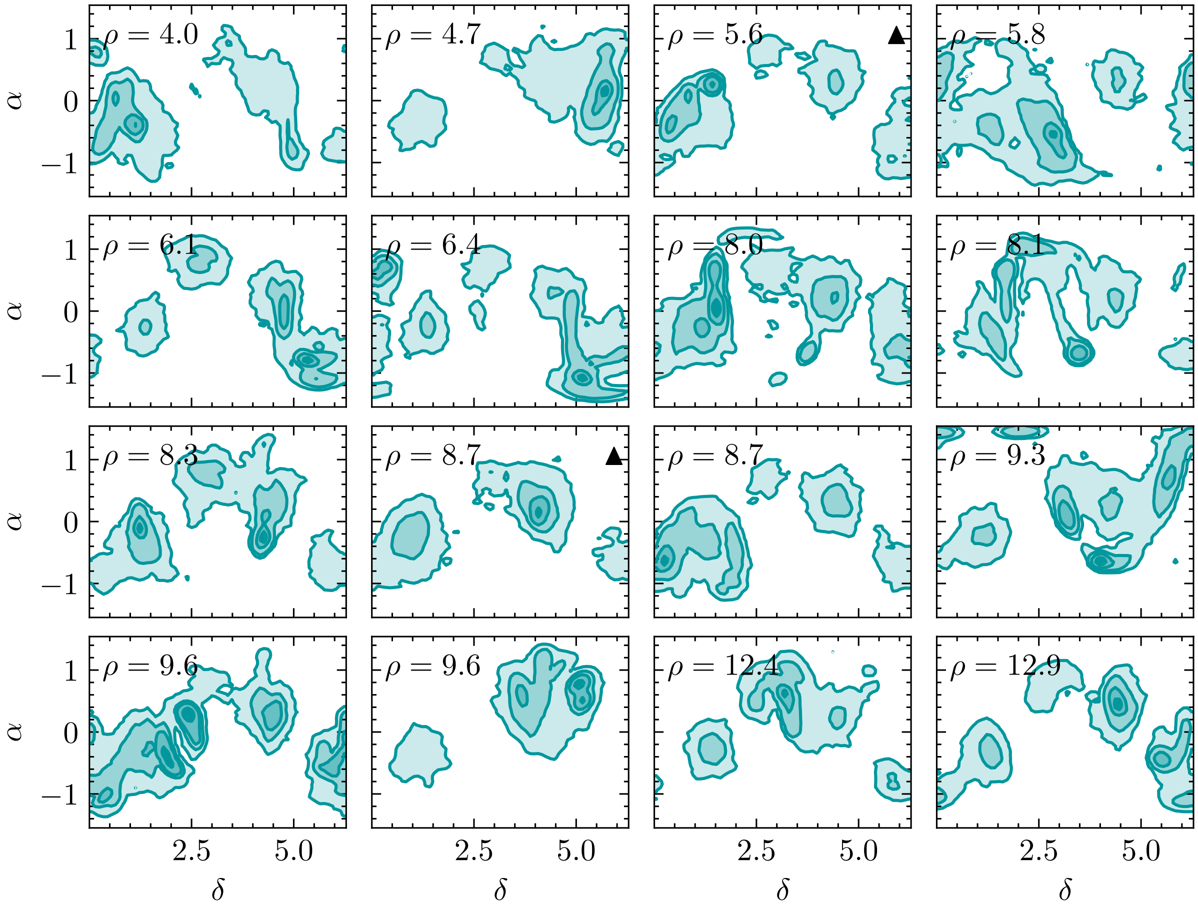


Figure 5.9: Nested samples for right ascension α and declination δ for the 16 injections, for which `nessai` with constant volume mode is the least efficient compared to `dynesty`. Runs which required fewer likelihood evaluations than `dynesty` are marked with a black triangle.

In `nessai`, this is exacerbated by the low number of samples used during training and will lead to learnt likelihood contours that contain regions below the likelihood threshold. Any samples drawn from these will be rejected, leading to low sampling efficiency and increasing the number of likelihood evaluations required.

5.5 Conclusions

In this chapter, we have introduced and tested two main modifications to the standard settings in `nessai`: constant volume mode, which fixes the latent radius, and periodically resetting the normalizing flow. These modifications aim to mitigate biases in the results obtained with `nessai` by reducing the potential for over-constraining the likelihood constrained prior when drawing new samples. Such biases arise in the evidence and posterior distributions and be diagnosed using the insertion indices discussed in section 2.6.7. Furthermore, these changes make `nessai` more robust to the choice of settings.

The results presented in section 5.1 show that without constant volume mode, the latent radius can vary greatly between iterations leading to over-constrained posterior distributions, particularly for parameters with posterior mass at the prior bounds, see fig. 5.1. Constant volume mode mitigates this by fixing the latent radius, thus reducing the over-constraining, as shown in fig. 5.1. However, biases can still arise in certain cases, such as when the posterior probability is maximal at the prior bounds.

The second modification, resetting the normalizing flow, is designed to avoid cumulative biases from poor training at a given iteration by periodically retraining from scratch. We show that this improves results for posteriors with maximum probability at the prior bounds, see table 5.1, for complex posterior distributions with long tails and multiple modes, such as the Rosenbrock likelihood (fig. 5.3), and for high-dimensional problems (table 5.2).

With these improvements in hand, we repeat the analysis of 128 BBHs presented in chapter 4 and find that `nessai` still passes the P-P tests without and with distance marginalization. Comparing these results to those obtained with the original settings, we find that the new settings improve sampling efficiency (measured in terms of likelihood evaluations) by up to 2.7 times whilst also reducing the wall time by 3.0 times. This results in median wall time of just 8.5 h on a single core with distance marginalization enabled.

We also explore how sampling efficiency compares to `dynesty` as a function of SNR and present the results in fig. 5.8. We find that `nessai` performs worse than `dynesty` at low network SNRs or when the SNR in a single detector is low. Such signals with have likelihood surfaces that multimodal in the sky angles, which normalizing flows are ill-suited to. This could be partially address by sampling in the detector-frame [96, 198] and we test this in chapter 8.

In summary, the modifications to the original version of `nessai` used in chapter 4 make sampling significantly more robust whilst also improving the efficiency for both gravitational-wave and non-gravitational-wave applications. We therefore recommend their use for analyses that use `nessai`.

Chapter 6

Sampling phase with nessai

This chapter considers applications of nessai to gravitational-wave analyses where the orbital phase cannot be marginalized and the challenges that arise from the increased complexity of the parameter space being sampled. This includes examining alternative reparameterizations of the phase and other source angles, such as the quaternion parameterization, and evaluating their suitability for use with nessai.

Section 6.1 presents a series of baseline analyses that highlight the issues that arise when sampling the phase with nessai. Section 6.2 describes how the source angles, inclination, polarization and phase, can be reparameterized using the unit-quaternions and defines equations for transforming between them. Section 6.3 reviews other reparameterizations of the phase that exist in the literature. These different reparameterizations are then tested and evaluated in section 6.4, starting with validating the quaternions and comparing the reparameterizations in sections 6.4.1 and 6.4.2 before performing analyses similar to those from chapters 4 and 5, but including phase, in section 6.4.3. Finally, section 6.5 summarizes the results and highlights important considerations to account for when using nessai for analyses without phase marginalization.

The orbital phase φ_c is hard to sample due to the correlations in the parameter space between it and the polarization and inclination [96]. The gravitational-wave analyses presented so far all avoid sampling the phase by using analytic phase marginalization. This marginalization was introduced in Veitch and Del Pozzo [252] relies on the assumption that the waveform can be factorized as $\boldsymbol{h}_{\boldsymbol{\theta}} = \boldsymbol{h}_{\boldsymbol{\theta},0}\exp(i\varphi_c)$, where $\boldsymbol{h}_{\boldsymbol{\theta},0}$ is the waveform evaluated at a reference frequency, e.g. $\varphi_c = 0$. This allows for the marginalized likelihood to be written as

$$p(\boldsymbol{d}|H) \propto \exp\left\{-\frac{1}{2}\left\langle \boldsymbol{h}_{\boldsymbol{\theta},0}|\boldsymbol{h}_{\boldsymbol{\theta},0}\right\rangle\right\}\exp\left\{-\frac{1}{2}\left\langle \boldsymbol{d}|\boldsymbol{d}\right\rangle\right\} \\ \times \frac{1}{2\pi}\int_0^{2\pi}\exp\frac{1}{2}\left\{\left\langle \boldsymbol{d}|\boldsymbol{h}_{\boldsymbol{\theta},0}\right\rangle e^{-i\varphi_c} + \left\langle \boldsymbol{h}_{\boldsymbol{\theta},0}|\boldsymbol{d}\right\rangle e^{i\varphi_c}\right\}\mathrm{d}\varphi_c, \quad (6.1)$$

where we have assumed $p(\varphi_c|H) = 1/2\pi$ [252]. This is a standard integral that yields a modified Bessel function of the first kind I_0:

$$\frac{1}{2\pi}\int_0^{2\pi}\exp\frac{1}{2}\left\{\left\langle \boldsymbol{d}|\boldsymbol{h}_{\boldsymbol{\theta},0}\right\rangle e^{-i\varphi_c} + \left\langle \boldsymbol{h}_{\boldsymbol{\theta},0}|\boldsymbol{d}\right\rangle e^{i\varphi_c}\right\}\mathrm{d}\varphi_c = I_0\left(\left|\left\langle \boldsymbol{d}|\boldsymbol{h}_{\boldsymbol{\theta},0}\right\rangle\right|\right), \quad (6.2)$$

so the marginalized likelihood can then be written as

$$p(\boldsymbol{d}|H) \propto \exp\left\{-\frac{1}{2}\left\langle \boldsymbol{h}_{\boldsymbol{\theta},0}|\boldsymbol{h}_{\boldsymbol{\theta},0}\right\rangle\right\}\exp\left\{-\frac{1}{2}\left\langle \boldsymbol{d}|\boldsymbol{d}\right\rangle\right\} I_0\left(\left|\left\langle \boldsymbol{d}|\boldsymbol{h}_{\boldsymbol{\theta},0}\right\rangle\right|\right). \quad (6.3)$$

However, as discussed in section 2.10.3, this is only valid for waveforms that solely contain the dominant 2-2 modes or when the effects of precession are minimal because of the assumption that $\boldsymbol{h}_{\boldsymbol{\theta}} = \boldsymbol{h}_{\boldsymbol{\theta},0}\exp(i\varphi_c)$. Therefore, since current analyses use approximants that include higher-order modes [9, 10, 98, 118] and this marginalization is no longer valid, we must instead sample the phase. We now explore the modifications necessary to efficiently sample phase with `nessai`.

6.1 Baseline analyses

As a point of reference, we now present a baseline analysis performed using settings based on the results presented in chapter 5. These use the gravitational-wave specific reparameterizations, minus the reparameterization for the spins, as described previously in section 5.4. We apply the same reparameterization to phase as was used for the other periodic angles, where they are mapped to Cartesian coordinates by introducing a radial parameter.

We analyse a GW150914-like injection with `IMRPhenomPv2` in a three-detector network with Advanced LIGO (aLIGO) and Advanced Virgo (AdVirgo) design sensitivity [70, 74, 77] with an optimal network signal-to-noise ratio (SNR) of ~ 40. We perform two analyses: one where all 15 parameters are sampled (or marginalized) and another where the spin parameters $\{a_1, a_2, \theta_1, \theta_2, \phi_{\mathrm{JL}}, \phi_{12}\}$ are fixed to the injected values. We repeat each analysis with and without phase marginalization, and use distance marginalization in both cases. When the spins are fixed, the extrinsic parameters are better constrained, making the correlations between polarization and phase more apparent, see for example fig. 6.6. We use this when testing the quaternion parameterization and demonstrating other parameterizations.

In fig. 6.1, we present a subset of the posterior distributions for the complete analysis with and without phase marginalization. These results show reasonable agreement for the phase, polarization and inclination but the remaining parameters are over-constrained when sampling without phase marginalization, most notably the spin magnitudes. The insertion indices, shown in appendix E.1, reinforce that the results are over-constrained and overall this is a clear indication that these initial settings are inadequate for sampling phase.

We also compare a subset of the diagnostics produced by `nessai`, shown in fig. E.2, and find that disabling phase marginalization significantly reduces the acceptance, which in turn results in the normalizing flow being trained more frequently and the pool of samples being repopulated more frequently. This is in-line with previous studies that find that disabling phase marginalization significantly reduces sampling efficiency [252], however the effect is significantly larger for `nessai`: we observe a ~ 10 times increase in the number of likelihood evaluations and wall time, whereas previous studies report closer to a factor of 4 [252].

Results for the runs with fixed spin parameters are included in appendix E.1. Unlike the previous results, these do not show clear signs of over-constraining and the posterior distributions obtained with and without phase marginalization agree, including for the phase and polarization. However, there is still a 10-fold increase in the number of likelihood evaluations and, therefore, total wall time, and the rejection sampling efficiency drops significantly.

These results imply that the `nessai` settings used in previous analyses can be used to sample phase, polarization and inclination, however, not in the complete precessing case. Therefore, revised settings or alternative reparemeterizations, or both, will be required to produced unbiased results when sampling phase. We now explore possible reparameterizations and settings in the following sections.

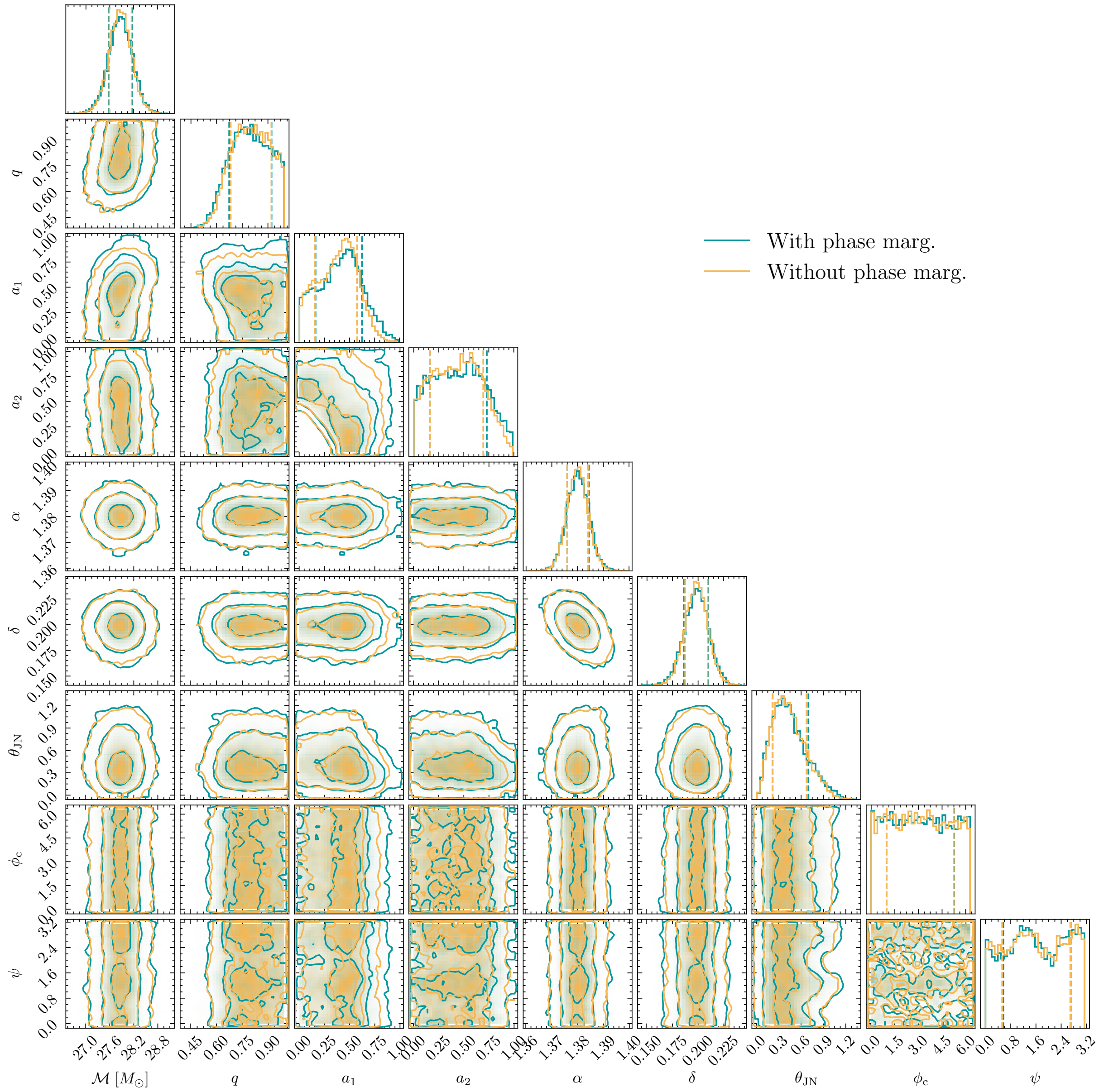


Figure 6.1: Corner plots comparing posterior distributions for a subset of parameters sampled with (blue) and without (orange) phase marginalization for GW150914-like signal in a three-detector network with an optimal network SNR of approximately 40. For the run with phase marginalization, the φ_c posterior has been reconstructed [198]. Both results were produced using `nessai` with the same sampling settings. The 2-dimensional contours contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

6.2 Representing the orientation of binaries with quaternions

The three source angles that parameterize the orientation of binaries, orbital phase φ_c, inclination $\theta_{\rm JN}$ and polarization ψ, describe a series of rotations and can be treated as Euler angles. They can therefore be transformed into unit-quaternions, which represent the 3-dimensional rotation group SO(3) [441]. The unit-quaternions are comparatively simpler to construct than Euler angles and eliminate their inherent degeneracies. We now examine the relationship between the source angles $(\varphi_c, \theta_{\rm JN}, \psi)$ and the unit-quaternions and determine if it is possible to sample in the unit-quaternions instead of the source angles.

6.2.1 Unit-quaternions

The quaternions are akin to the complex numbers but defined in four–dimensional space and introduce not one but three 'imaginary parts' $\boldsymbol{q} = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$ where a, b, c, d are real coefficients [442]. However, since our focus is on rotations, we will use the unit-quaternions:

$$\hat{\boldsymbol{q}} = [q_0, q_1, q_2, q_3]^T, \tag{6.4}$$

where

$$|\hat{\boldsymbol{q}}|^2 = q_0^2 + q_1^1 + q_2^2 + q_3^2 = 1, \tag{6.5}$$

which lie on the 3-sphere S^3 and can describe the rotation group SO(3) [443]. The rotations describe the Euler angles and the unit-quaternions can be related to equivalent rotation matrices, the exact form of which will depend on the convention used [444]. In this work, we use the ZYZ convention for the Euler angles which starts with a rotation through γ about the z-axis, followed by a rotation through β about the y-axis and finally a rotation about the z-axis by α, where y and z are the fixed frame axes. This allows us to define the transformation from the Euler angles to the unit-quaternions [444]:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(\frac{q_3}{q_0}\right) + \operatorname{atan2}\left(\frac{-q_1}{q_2}\right) \\ 2\arccos\left(\sqrt{\frac{q_0^2+q_3^2}{|\hat{\boldsymbol{q}}|}}\right) \\ \operatorname{atan2}\left(\frac{q_3}{q_0}\right) - \operatorname{atan2}\left(\frac{-q_1}{q_2}\right) \end{bmatrix}, \tag{6.6}$$

where the atan2 function [445] allows for distinguishing between quadrants. The inverse transformation is then

$$\hat{\boldsymbol{q}} = \begin{bmatrix} \cos\dfrac{\beta}{2}\cos\dfrac{\alpha+\gamma}{2} \\ -\sin\dfrac{\beta}{2}\sin\dfrac{\alpha-\gamma}{2} \\ \sin\dfrac{\beta}{2}\cos\dfrac{(\alpha-\gamma)}{2} \\ \cos\dfrac{\beta}{2}\sin\dfrac{(\alpha+\gamma)}{2} \end{bmatrix}. \tag{6.7}$$

We now consider how the source angles are related to this convention.

6.2.2 From unit-quaternions to source angles and back

Our goal is to sample in the parameter space defined by the unit-quaternions and transform to source angles when evaluating the likelihood, however we must first address the inverse of this transformation: from source angles to quaternions. For this, we must relate $[\varphi_c, \theta_{\mathrm{JN}}, \psi]$ to $[\alpha, \beta, \gamma]$ and define equivalent priors on the unit-quaternions that can be used for sampling.

We follow the convention used in section 1.3.2, and define the line-of-sight to the observer $\hat{\boldsymbol{n}}$ to be aligned with the z-axis such that x–y plane lies perpendicular to $\hat{\boldsymbol{n}}$. The source angles describe a rotation through φ_c about the line-of-sight, followed by a rotation through θ_{JN} about the y-axis and finally a rotation about the line-of-sight through ψ. Therefore, the equivalence is $[\psi, \theta_{\mathrm{JN}}, \varphi_c] \equiv [\alpha, \beta, \gamma]$ which allows us to relate the source angles to the unit-quaternions with eqs. (6.6) and (6.7).

The prior distributions for the sources angles are $\varphi_c \sim \mathcal{U}[0, 2\pi]$, $\theta_{\mathrm{JN}} \sim \sin\theta_{\mathrm{JN}}$ and $\psi \sim \mathcal{U}[0, \pi]$ where polarization is defined on the interval $[0, \pi]$ because gravitational waves are spin-2 in nature[1]. To determine the equivalent priors on the components of the unit-quaternions, we first consider the domain over which they are defined.

As mentioned previously, the unit-quaternions lie on the 3-sphere S^3 so we consider the uniform distribution on S^3 and transform to the Euler angles using eq. (6.6). However, S^3 double-covers SO(3) [441], so the resulting distributions will double-cover the Euler angles. This is remedied by mapping back to the closed region $[0, 2\pi] \times [0, \pi] \times [0, 2\pi]$ using modulo 2π. We then take one more step and apply $\alpha \mod \pi$ to map it back to $[0, \pi]$ to match the interval used for ψ. This transformation is two–to–one and means that once samples are transformed to the source angle-space, the original quaternions cannot

[1] This implies that a gravitational wave produced by a source that is rotated by $\psi = 2\pi$, will be equal to those produced by a source with $\psi = \pi$.

be recovered. The final transformation from unit-quaternions to source angles is therefore

$$\begin{bmatrix} \varphi_c \\ \theta_{\mathrm{JN}} \\ \psi \end{bmatrix} = \begin{bmatrix} \left[\mathrm{atan2}\left(\frac{q_3}{q_0}\right) + \mathrm{atan2}\left(\frac{-q_1}{q_2}\right)\right] \mod 2\pi \\ 2\arccos\left(\sqrt{\frac{q_0^2+q_3^2}{|\hat{\boldsymbol{q}}|}}\right) \\ \left[\mathrm{atan2}\left(\frac{q_3}{q_0}\right) - \mathrm{atan2}\left(\frac{-q_1}{q_2}\right)\right] \mod \pi \end{bmatrix}. \tag{6.8}$$

What remains is to define the priors on the unit-quaternions, eq. (6.4). However, since these are defined on the unit-3-sphere, the prior will be joint on the four components

$$p(\hat{\boldsymbol{q}}) = \frac{1}{S}\delta(|\hat{\boldsymbol{q}}| - 1) \tag{6.9}$$

where $\delta(\cdot)$ is the delta function and $S = 2\pi^{3/2}/\Gamma(3/2)$ is the surface area of the unit-3--sphere. Whilst one could use joint priors for sampling, in practice it is easier to instead consider the quaternions and define the priors in the non unit-quaternion space. Muller and Marsaglia [427, 428] showed that points can be drawn uniformly on the n-sphere by generating an n-dimensional vector of normal deviates $\boldsymbol{x}$ and normalizing by the length of the vector. We can therefore define the priors on each of the quaternion components such that $p(q_i) \equiv \mathcal{N}(q_i; 0, 1)$ which is equivalent to defining a joint uniform prior on all four components. During sampling, a quaternion can be normalized to a unit-quaternion and then used to compute the source angles. This also has the added benefit of being better suited for use with `nessai` since the priors are unbounded, so the normalizing flow cannot propose samples outwith the priors bounds.

6.3 Other reparameterizations for phase and the source angles

We now consider alternative reparameterizations of the phase, and in some cases, other source angles. We compare them to the default reparameterization used for angles in `nessai` using the posterior distribution from the analysis with the fixed spin parameters described in section 6.1. The posterior distribution in the source angles is presented in fig. 6.2 and shows the typically correlations that are observed between the three source angles.

The approach used in the baseline run is described in detail section 4.3, in summary: a radial component ρ_θ is sampled from a χ-distribution with two degrees of freedom and

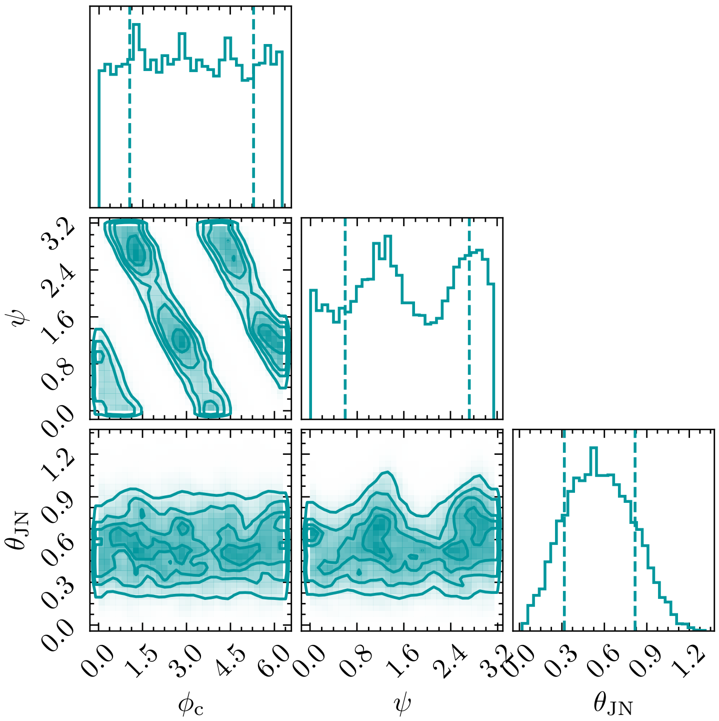


Figure 6.2: Posterior distribution for the source angles for a GW150914-like signal with all the spin parameters fixed to the injected values. This shows the highly-correlated and multimodal nature of these parameters. The 2-dimensional contours contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

used to convert the angle θ to Cartesian coordinates:

$$
\begin{aligned}
x_\theta &= \rho_\theta \cos\theta, \\
y_\theta &= \rho_\theta \sin\theta.
\end{aligned}
\tag{6.10}
$$

This treatment is used for angles that are periodic and the resulting parameters have unit-Gaussian priors. For angles, such as the inclination, that are not periodic and have Sine or Cosine priors we instead use the default rescaling in `nessai` which rescales the samples at a given iteration to $[-1, 1]$ since the probability at the prior bounds is zero. The posterior distribution after applying this reparameterization is shown in fig. 6.3. This distribution is free of hard bounds. However, some of the parameters appear 'folded' and have other, more complex shapes or multiple modes. Whilst the flow can learn approximations of these shapes, the regions between the various modes will lead to samples being rejected based on the likelihood criterion.

`LALInference` [96] introduced two replacement parameters for polarization and phase that leverage the correlations between the three source angles, they are

$$
\begin{aligned}
\alpha &= \psi + \varphi_c, \\
\beta &= \psi - \varphi_c.
\end{aligned}
\tag{6.11}
$$

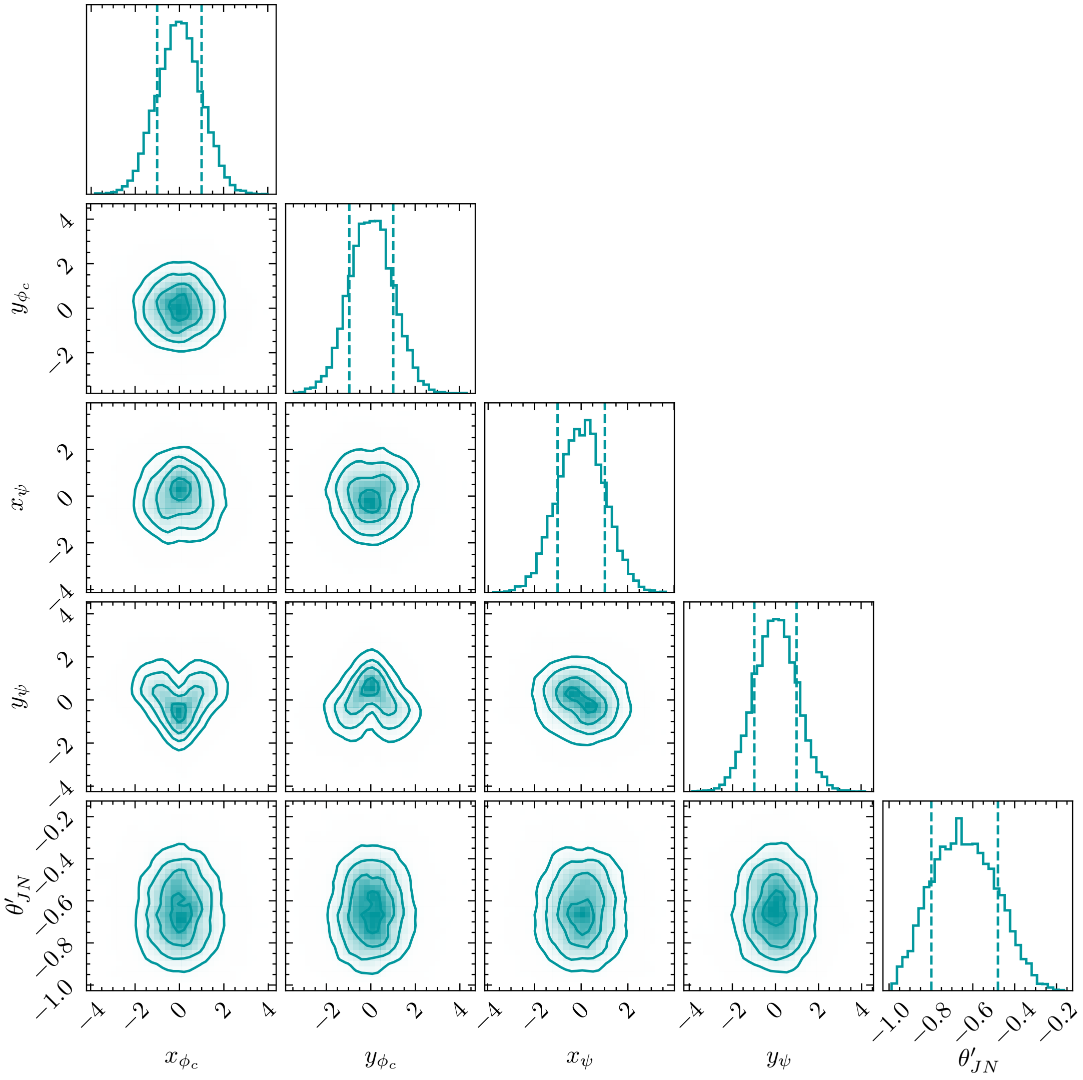


Figure 6.3: Posterior distribution for the source angles for a GW150914-like signal, with all the spin parameters fixed to the injected values, reparameterized using the default treatment in the original version of `nessai`. This method converts the angles to Cartesian coordinates by introducing radial parameters. The 2-dimensional contours contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

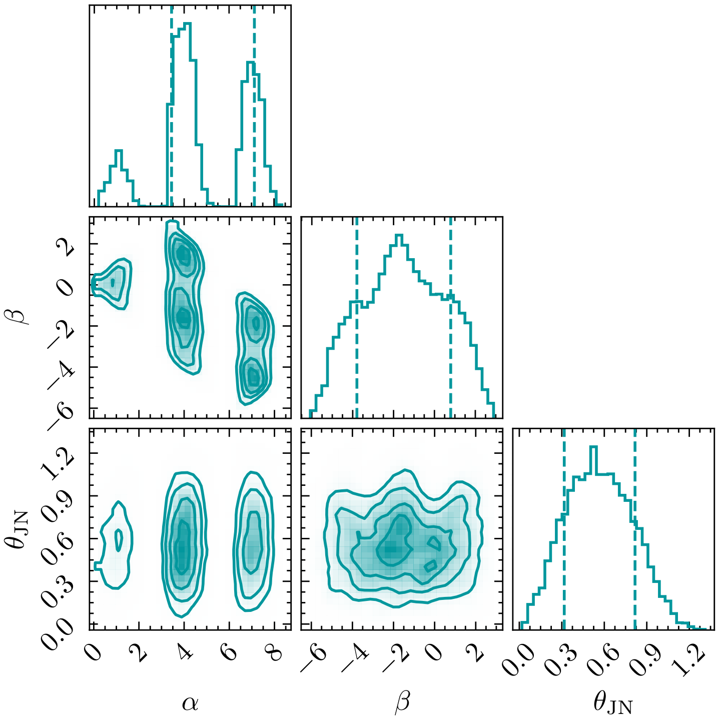


Figure 6.4: Posterior distribution for source angles using the parameters α and β introduced in `LALInference` and defined in eq. (6.11) for a GW150914-like signal with all the spin parameters fixed to the injected values. The 2-dimensional contours contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

These parameters remove the correlation between the polarization and phase and result in a posterior distribution with disconnected modes in the $\alpha - \theta_{\rm JN}$ space as shown in fig. 6.4. Closely related to this is the use $\Delta\varphi_c$ which was introduced in `bilby` [198] and is defined as

$$\Delta\varphi_c = \varphi_c + \operatorname{sgn}(\cos\theta_{\rm JN})\psi, \tag{6.12}$$

where sgn returns the sign $\{+, -\}$ of the input. This single parameter has a similar effect to α and β and is not correlated with the inclination. The corresponding posterior is shown in fig. 6.5.

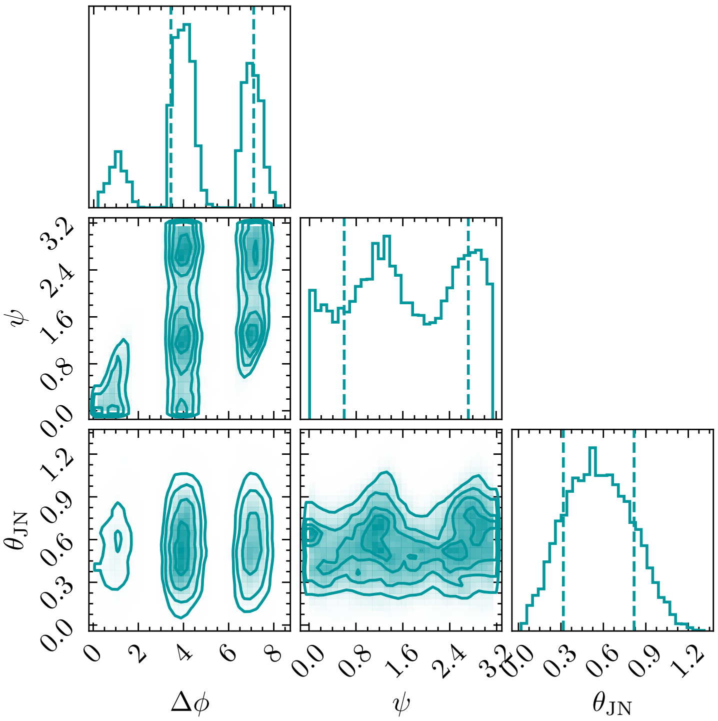


Figure 6.5: Posterior distribution for source angles where the phase has been replaced with $\Delta\varphi_c$ as defined in eq. (6.12) for a GW150914-like signal with all the spin parameters fixed to the injected values. The 2-dimensional contours contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

6.4 Results

We test the different reparameterizations using the baseline analyses described in section 6.1 and then based on these findings perform P-P tests to validate the settings. We consider four different methods of sampling the phase: sampling the standard phase and applying the reparameterization for periodic parameters, sampling $\Delta\varphi_c$ and applying the reparameterization for periodic parameters, sampling the unit-quaternions and sampling without any gravitational-wave specific reparameterizations. We omit the (α, β) reparameterization since when the angles are transformed to Cartesian coordinates the resulting space only differs by a rotation to that obtained when using $\Delta\varphi_c$. All analyses are performed using `bilby` [198] with a custom likelihood implementation for the quaternions.

6.4.1 Quaternions

We validate sampling the quaternions by analysing the GW150914-like injection with fixed spins twice: once with the default parameters and once sampling the quaternions, and comparing the results. In the latter, the quaternions are transformed to unit-quaternions and then source angles during the likelihood calculation.

We start by comparing the posterior distributions, which are presented in fig. 6.6.

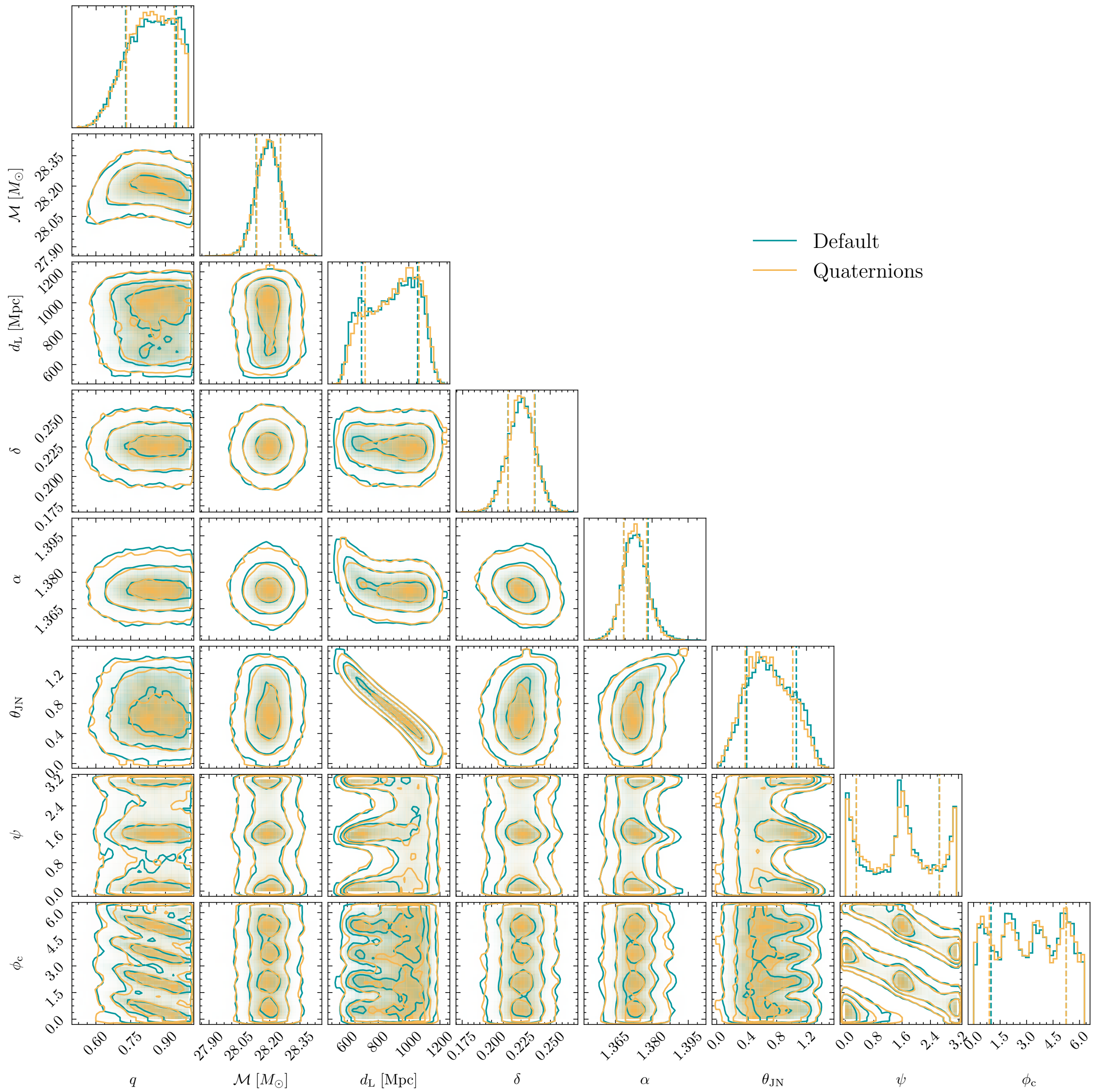


Figure 6.6: Comparison of the posterior distributions obtained when sampling with the standard source angles $(\varphi_c, \theta_{\mathrm{JN}}, \psi)$ versus the quaternions. The results for the quaternions are shown in yellow and those for the default source angles are shown in teal. The 1-σ confidence intervals are shown for the marginal distributions and 2-dimensional contours shown contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

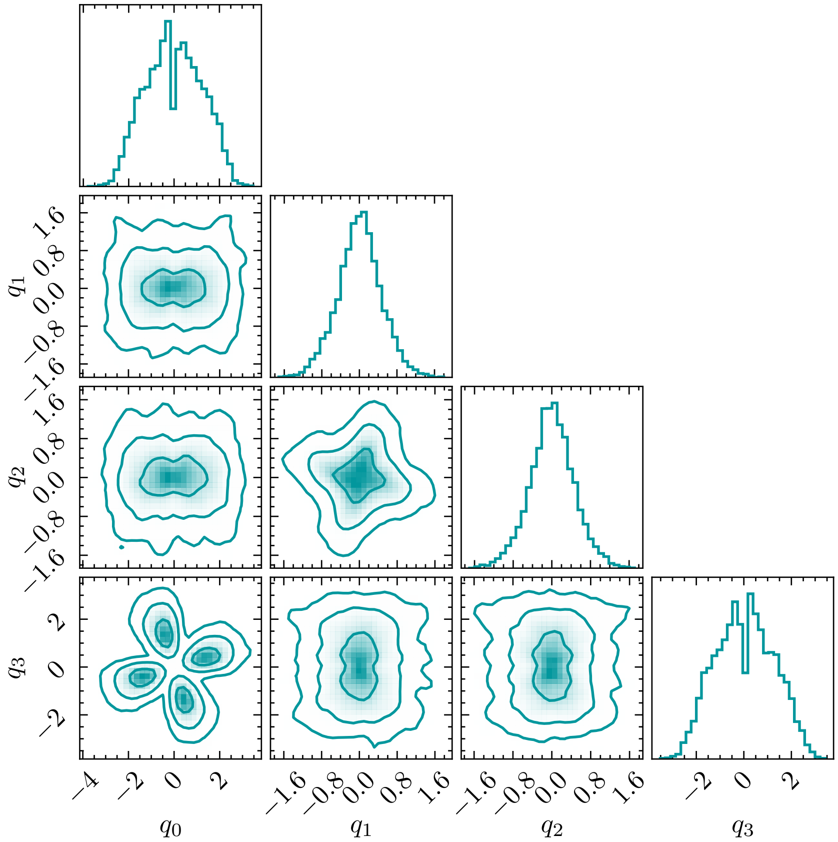


Figure 6.7: Posterior distribution for a GW150914-like binary black hole (BBH) injection in the unnormalized quaternion parameter space. The 2-dimensional contours contain 39.3%, 86.5% and 99.8% of the posterior probability respectively

These show that sampling the quaternions allows `nessai` to correctly capture the correlations between the phase and polarization and verifies that the transforms and prior distributions described are indeed equivalent to the standard parameters and priors. The posterior distribution in the unnormalized quaternion space is shown in fig. 6.7. This is the space in which `nessai` is sampling and the normalizing flows are being trained. We observe a cloverleaf-like structure in the q_0–q_3 plot; this arises because the quaternions double-cover the Euler angles which in turn double-cover the source angles, and is a particularly challenging structure for the normalizing flows to learn since it is multimodal and the marginal distributions have a narrow trough around the origin. These features may lead to inefficient sampling or a particular mode being undersampled when compared to the others. We therefore examine the effect this has on sampling efficiency.

The log-evidences, number of likelihood evaluations and wall times for both analyses are presented alongside the baseline analysis with phase marginalization in table 6.1. The quaternion analysis over-estimates the log-evidence compared to the baseline runs, implying the run is over-constrained and the settings may need fine-tuning, however we leave this for complete analysis in section 6.4.2. The number of likelihood evaluations between

Table 6.1: Comparison of the results obtained when sampling using the quaternions compared to the default reparameterisation.

	$\ln Z$	ESS	Likelihood evaluations	Wall time [min]
Default (marg.)	-12145.37 ± 0.11	11 763	358 916	6.6
Default	-12145.23 ± 0.12	12 629	3 544 511	72
Quaternions	-12144.64 ± 0.12	12 726	3 433 878	38

the two runs without phase marginalization are comparable, meaning that the sampling in the quaternion space has not significantly improved sampling efficiency. However, the wall time has decreased by $\sim 50\%$. We investigate the cause of the difference and find that the rejection sampling efficiency that ensures samples are distributed according to the prior is more efficient when using the quaternions.

We have shown that the unit-quaternions are an alternative parameterization of the source angles. When sampling in a reduced parameter space, this parameterization is as efficient as sampling the source angles directly. However, that parameter space presents various features that may prove difficult to sample in some cases. We test the quaternions on the complete parameter space in the next section.

6.4.2 Comparing reparameterizations

We first consider the different reparameterizations: the default, $\Delta\varphi_c$ and the quaternions. Changing the parameters being sampled to $\Delta\varphi_c$ or the quaternions is insufficient to prevent the over-constraining described previously, therefore irrespective of which parameters are sampled other settings will need changing. Given the results obtained in chapter 5, we consider increasing the volume of the latent contour by increasing ρ_{CVM} and periodically resetting the flow.

We iteratively increase latent volume up to $\rho_{\mathrm{CVM}} = 0.995$ and find that this progressively reduces the over-constraining at the cost of reducing the efficiency of the rejection sampling, which leads to longer wall times. However, even at $\rho_{\mathrm{CVM}} = 0.995$ the insertion indices show signs of over-constraining, suggesting further changes may be needed. We present a subset of the diagnostics produced by `nessai` in fig. 6.8 for the three different parameters being sampled. These results show that sampling in $\Delta\varphi_c$ or the quaternions is largely comparable to the default reparameterization. The run sampling $\Delta\varphi_c$ is slightly more efficient than the other two, however, it is also the most over-constrained of the three runs, see appendix E.2. These results suggest that there is little benefit to sampling in these parameters compared to default reparameterization in `nessai`. We attribute this to the multimodality present in the parameter spaces, as seen in figs. 6.5 and 6.7, which we know reduces sampling efficiency. Therefore, we do not consider these reparameterizations in further analyses and use the default reparameterization.

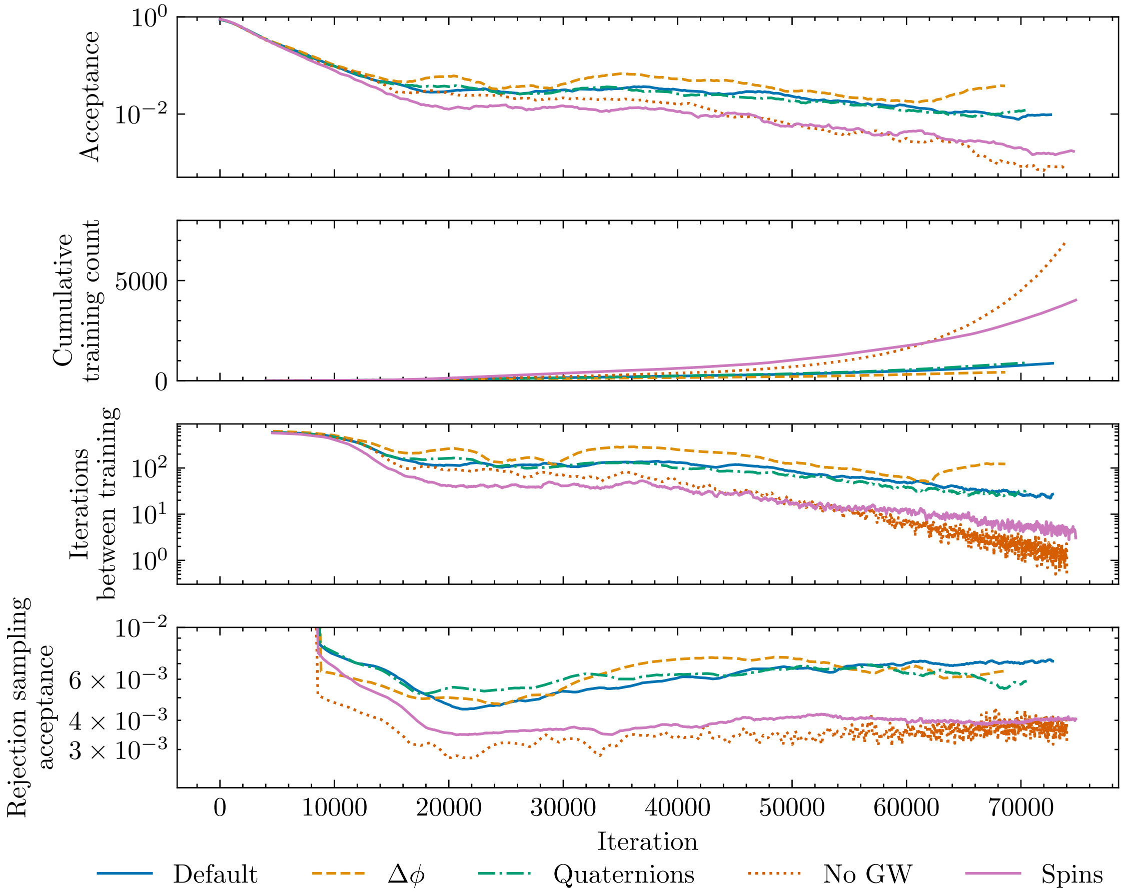


Figure 6.8: Subset of the diagnostic plots for runs with `nessai` using different reparameterizations to sample phase. From top to bottom these are: the acceptance base on the likelihood threshold at a given iteration, the cumulative training count, number of iterations between training and the rejection sampling efficiency. To improve readability, the rolling mean with a window length of 16 has been used to smooth the acceptances and iterations between training. Results are shown for a run sampling the default phase parameter with the default reparameterizations, a run sampling with quaternions, a run sampling $\Delta\varphi_c$ instead of phase, a run sampling the default phase with no gravitational-wave reparameterizations and a run sampling the default phase parameter with additional reparameterizations for the spin magnitudes.

The over-constraining seen in fig. 6.1 is similar to what was described in section 5.3, so one might expect that resetting the flow would address the over-constraining. However, we find that the increase in computational cost is prohibitive; more than 10-fold that of the baseline analysis without phase marginalization irrespective of the parameters being sampled. We attribute this to the reparameterizations used to map between the $\mathcal{X}$-space and the $\mathcal{X}'$-space which change how the likelihood-constrained prior shrinks, most notably the reparameterization for mass ratio that mirrors samples about the prior bound.

In lieu of resetting the flow, we consider alternative reparameterizations for the spin parameters. In the original version of `nessai`, the spin magnitudes were mapped to a two–dimensional parameter space (x_{a_i}, y_{a_i}) where a uniform prior corresponded to a two–dimensional Gaussian. In chapter 5, we found that this reparameterization was no longer needed when using constant volume mode, which sped up inference. We re-test the reparameterization when sampling phase and find that it helps mitigate the over-constraining with lower values of ρ_{CVM} however it significantly decreases the sampling efficiency as shown in fig. 6.8. We test these settings further in the following section.

As an alternative, we also test disabling the gravitational-wave specific reparameterizations and use instead the default rescaling in `nessai` which rescales the samples at a given iteration to $[-1, 1]$ in each dimension based on the current minimum and maximum. This presents different challenges, for example there is not dedicated treatment for periodic parameters, but we find that periodically resetting the flow every 8'th training and increasing the volume fraction to $\rho_{\mathrm{CVM}} = 0.98$ are enough to produce results that do not show signs of over-constraining, see fig. 6.9. The various diagnostics for this run are shown alongside the previous analyses in fig. 6.8. The run without any problem-specific reparameterizations has significantly lower acceptance, particularly at early iterations where a large fraction of the proposed samples lie outside the prior bounds.

6.4.3 Probability-probability tests

The tests conducted so far serve as a starting point for conducting P-P tests. We perform P-P tests using the same configuration as in chapter 4. The simulated four-second BBH signals in a three-detector network with design sensitivity [70, 77] sampled at 2048 Hz. We use the same priors but, due to the increased computational cost, reduce the number of injections from 128 to 64 and use 16 cores per analysis. The median network SNR of the resulting injections is 18.

We perform three separate analyses. The first uses the defaults reparameterizations but increases the latent contour to $\rho_{\mathrm{CVM}} = 0.995$, the second uses the reparameterizations for the spin magnitudes introduced in chapter 4 with $\rho_{\mathrm{CVM}} = 0.98$ and the third uses the default reparameterizations in `nessai` with $\rho_{\mathrm{CVM}} = 0.98$ and the flow is reset every 8th training.

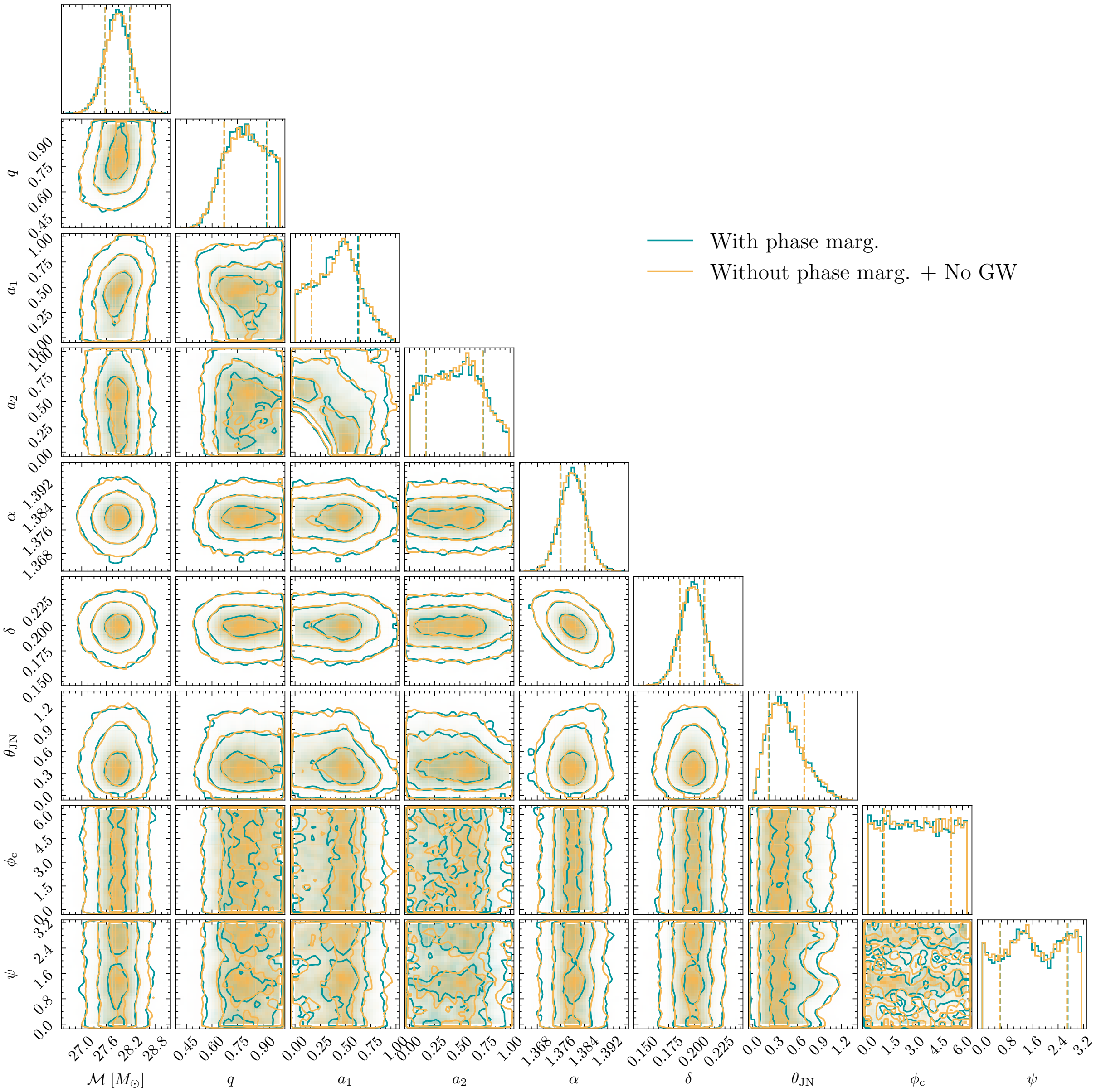
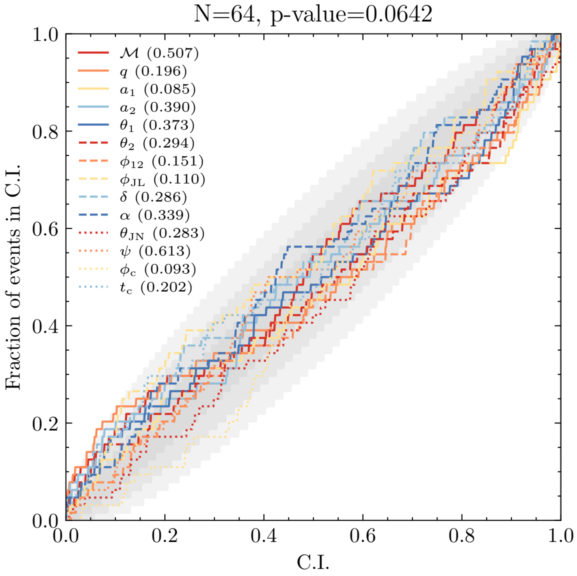


Figure 6.9: Corner plot comparing the posterior distributions for the baseline analysis using `nessai` with phase marginalization described in section 6.1 and for an analysis using `nessai` without any problem-specific reparameterizations. The 1-σ confidence intervals are shown for the marginal distributions and 2-dimensional contours shown contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.



(a) With default reparameterizations and $\rho_{\mathrm{CVM}} = 0.995$

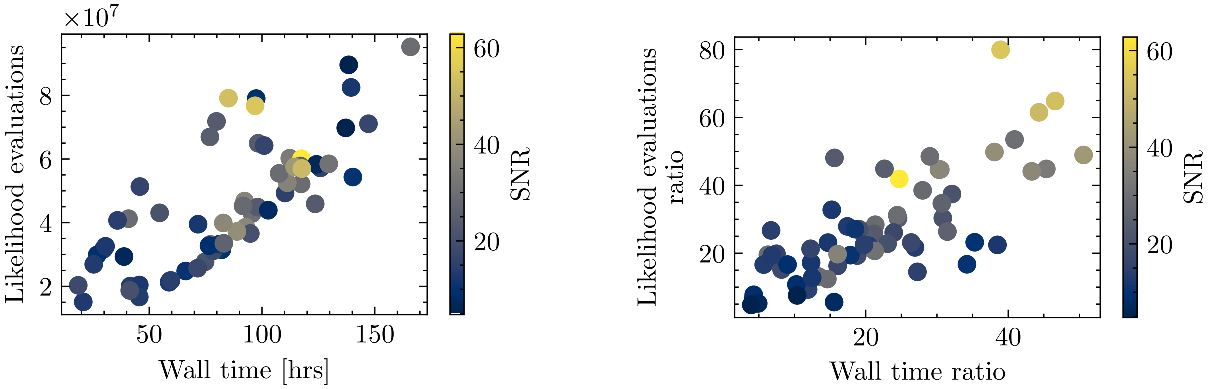
(b) With spin reparameterizations and $\rho_{\mathrm{CVM}} = 0.98$

Figure 6.10: P-P plots showing the confidence interval versus the fraction of the events within that confidence interval for the posterior distributions obtained using our analysis `nessai` for 64 simulated compact binary coalescence signals produced with `bilby` and `bilby_pipe`. The 1-, 2- and 3-σ confidence intervals are indicated by the shaded regions and p-values are shown for each of the parameters and the combined p-value is also shown.

We present probability-probability plots (P-P plots) for the first two analyses in fig. 6.10. The P-P plot for the first analysis with $\rho_{\mathrm{CVM}} = 0.995$ indicate that there is a systematic bias in the posterior distributions, most notably in the spin magnitude a_1 and phase φ_c. The results for the second analysis with the additional reparameterizations pass the P-P test with a p-value of 0.3997. This suggests that the additional reparameterization for the spins is needed to reliably sample without phase marginalization.

The analyses without any problem-specific reparameterizations show significantly different behaviour. A fraction of the analyses complete in comparable times to the previous analyses, however, the remaining analyses take significantly longer and in some cases failing to complete after two weeks. This is a result of the rejection sampling being increasingly inefficient at later iterations due to multimodality in various parameters, including the sky location and geocentre time, similar to what was observed in section 5.4.

We examine the number of likelihood evaluations and wall times as a function of network SNR for the analyses that passes the P-P test and present the results as a function of network SNR in fig. 6.11. Higher SNRs require more likelihood evaluations and, as expected, take longer to sampler. In some cases, low SNR injections can take longer than higher SNRs, this occurs when the likelihood surface is multimodal which, as discussed previously, can lead to inefficient sampling. The average number of likelihood evaluations required is 43.0×10^6 which corresponds to an average wall time of 84 hours. We show a



(a) Sampling without phase marginalization

(b) Compared to sampling with phase marginalization

Figure 6.11: Wall time and number of likelihood evaluations when analysing the 64 BBH injections using `nessai` without phase marginalization. The colour of each marker corresponds to the optimal network SNR of each injection. **Left:** the wall times in hours versus the total number of likelihood evaluations. **Right:** increase in the wall time and the number of likelihood evaluations compared to sampling with phase marginalization enabled.

breakdown of the total wall time in terms of time spent evaluating the likelihood and time spent populating the pool of new samples, which is dominated by the cost of rejection sampling, in fig. 6.12. In spite of using 16 cores to parallelize the likelihood evaluation, it is still the dominant cost and, on average, accounts for 71% of the wall time whilst populating the pool accounts for 27% and training accounts for less than 1%. We also compare the wall time and number of likelihood evaluations to equivalent analyses with phase marginalization enabled, also run with 16 cores. The P-P plot for this analysis is included in appendix E.3. We present the comparison in fig. 6.11. On average, sampling phase is 20 times slower than marginalizing it whilst requiring 23 times as many likelihood evaluations.

6.5 Conclusions

The orbital phase is challenging to sample, and the same applies when using `nessai`. The settings used in previous gravitational-wave analysis with `nessai` are not suitable for applications where phase cannot be marginalized and result in over-constrained results, as seen in fig. 6.1. Therefore, revised settings or alternative reparameterizations are required for such as analyses.

We relate the source angles, phase, polarization and inclination, to the unit-quaternions and demonstrate that they are equivalent. However, due to the spin-2 nature of gravitational waves, the mapping from source angles to unit-quaternions is one-to-many and the unit-quaternion space double covers the source angle space. We introduce equivalent

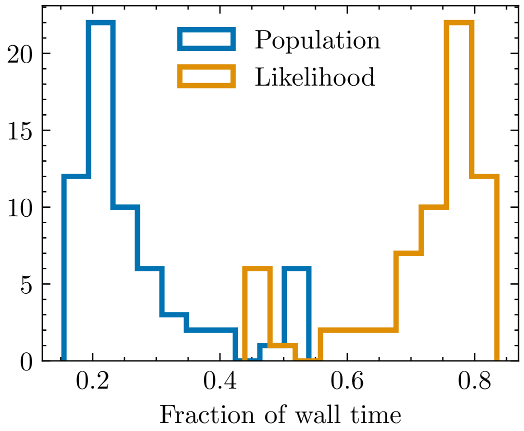


Figure 6.12: Fraction of the wall time spent evaluating the likelihood (orange) and populating the pool of new samples (blue) when analysing 64 BBH injections without phase marginalization.

priors on the quaternions and show that it is possible to sample in the quaternion space to obtain results that are consistent with sampling in the standard parameter space, see fig. 6.6.

We compare sampling the default phase parameter, to sampling the quaternions or sampling $\Delta\varphi_c$ and find that irrespective of the parameter that is sampled, the results are over-constrained, suggesting that other settings need changing. We find that over a range of settings, sampling the quaternions or $\Delta\varphi_c$ leads to lower efficiency than sampling the default parameters, which we attribute to the multimodal likelihood surface.

Following the success using constant volume mode and resetting the flow seen in chapter 5, we test both when sampling phase. Increasing the volume of the latent contour helps reduce over-constraining but is insufficient to pass a P-P test. We find that resetting the flow is prohibitively expensive with the gravitational-wave specific reparameterizations and, as such, we do not recommend it in this case. However, if one disables the these reparameterizations, increasing ρ_{CVM} and resetting the flow produces unbiased results for a GW150914-like injection. The final change we test is using different reparameterizations for the spin magnitudes, as described in section 4.3.1.

We perform P-P tests with three different configurations: setting $\rho_{\mathrm{CVM}} = 0.995$, using additional spin reparameterizations and disabling all gravitational-wave specific reparameterizations. Of the three, only the analysis with additional spin reparameterizations passes the P-P test. The analysis with $\rho_{\mathrm{CVM}} = 0.995$ produces over-constrained results, and the analysis without gravitational-wave specific reparameterizations fails to converge for a number of the injections.

These results show that `nessai` can be used in scenarios where phase marginalization is no longer applicable. However, even when running with 16 cores, the analyses take on average 84 hours and this is dominated by the time spent evaluating the likelihood. The equivalent analyses with phase marginalization are, on average, 20 times faster and require 23 times fewer likelihood evaluations. This suggests that, whilst these settings pass the P-P test, they could likely be improved. These improvements could focus on either finding better reparameterizations for the spins, which are often over-constrained, or developing methods for handling multimodality, which becomes the limiting factor when not using the gravitational-wave specific reparameterizations.

In summary, the results presented in this chapter serve as a starting point for performing inference in scenarios where phase marginalization is not applicable. Most importantly, such analyses require different settings and care should be taken to ensure results are correctly converged. In chapter 8, we perform analyses on real LIGO-Virgo-KAGRA Collaboration (LVK) data, including without phase marginalization, which serves to further validate the settings we have proposed so far.

Chapter 7

Importance nested sampling with normalizing flows

This chapter introduces i-nessai: importance nested sampling with artificial intelligence, a novel modified nested sampling algorithm that is designed around normalizing flows and addresses the main bottlenecks in nessai. We validate i-nessai on toy problems and for gravitational-wave parameter estimation and demonstrate the speed-ups it can provide.

This work has been published in Williams et al. [2] and it is presented almost verbatim. We have omitted the introductory sections since the material is already covered in the previous chapters. The appendices are included in appendix F but the appendix on the training normalizing flows with weights has been omitted since it already covered in section 3.3.5. Certain spellings have been updated to be consistent with other chapters, but the symbols have not been changed, as they were already consistent with other chapters.

This chapter is structured as follows: section 7.1 reviews the various alternative formulations of nested sampling that this work builds upon; these are described in-depth in section 2.6.6. A simplified version of the modified nested sampling is then described and validated in section 7.2. This is followed by a description of the complete method and algorithm, including all its components, in section 7.3. Section 7.4 reviews existing related approaches in the literature. Section 7.5 presents a range of results from analyses of increasing complexity; these include analyses of the Rosenbrock likelihood and two gravitational-wave analyses. The results are also compared to nessai and dynesty. Finally, section 7.6 reviews the results, summarizes the main finding and discusses potential directions for future work.

When implementing nested sampling, the main challenge is drawing new points from the likelihood-constrained prior at a given iteration. There are different approaches to this such as using Markov Chain Monte Carlo (MCMC), slice sampling or sampling from bounding distributions [196]. There have also been efforts to incorporate machine learning into nested sampling for approximating the likelihood [410], in the proposal process [1, 435] and for sampling from arbitrary priors [446].

In Williams et al. [1], we proposed `nessai`, a nested sampling algorithm that uses normalizing flows to approximate the likelihood-constrained prior at different iterations. We showed that this approach could speed up convergence and allowed for natural parallelization of the likelihood. However, we noted that a significant portion of the wall time was being spent performing rejection sampling to ensure points were distributed according to the prior, and this, alongside the inherently serial nature of nested sampling, set an upper limit on how fast the algorithm could be.

In this work, we present a modified nested sampling algorithm based on importance sampling that addresses the aforementioned bottlenecks. In particular, this modified algorithm:

- incorporates normalizing flows similarly to Williams et al. [1],
- removes the requirement for samples to be independently and identically distributed (i.i.d.) and distributed according to the prior,
- allows samples to be added in any order independent of a likelihood constraint,
- allows the evidence to be updated for batches of samples.

Taken together, these changes improve the efficiency of the algorithm, reducing the number of required likelihood evaluations by up to an order of magnitude over our previous version, and greatly increasing the scalability of the algorithm.

This is especially relevant in the context of gravitational-wave data analysis, where nested sampling is the de facto analysis algorithm [96, 198]. As of the last LIGO-Virgo-KAGRA Collaboration (LVK) [71–73] observing run, there are 90 confirmed detected compact binaries [8–10] and this number is expected to increase by a factor of ~ 3.3 in the fourth observing run [74]. This presents a significant computational challenge since typical analyses take of order days to weeks. Furthermore, a subset of these analyses are currently only possible at great computational cost [250, 285]. The algorithm we present brings the possibility of tackling these challenging analyses and dramatically reduces the wall-time required to complete an analysis.

7.1 Alternative formulations of nested sampling

This work draws on three existing formulations of nested sampling which were introduced previously in section 2.6.6. We now summarize each of them.

- **Diffusive nested sampling:** this variant of nested sampling explores multi-level likelihood levels at once and retains all samples from the MCMC chains irrespective of the likelihood threshold. The main similarity to our work is the lack of a likelihood constraint and how all the samples that are drawn, are used.
- **Importance nested sampling:** this formulation of nested sampling defines a *pseudo-importance sampling density* $Q(\boldsymbol{\theta})$ which is used to compute the evidence instead of the usual weights computed using the prior volume. Similarly to diffusive nested sampling, it does not reject samples based on a likelihood threshold.
- **Nested Sampling via Sequential Monte Carlo:** this framework reformulates nested sampling as a version of Sequential Monte Carlo (SMC) which has similarities to importance nested sampling. Batches of samples are drawn from successive probability density functions (PDFs) until the overall distribution approximates the posterior.

The keys elements of these three versions of nested sampling that we draw upon are: removing the likelihood threshold, considering batches of samples and computing the evidence using a proposal distribution Q. In this formulation, the evidence is computed using

$$\hat{Z} = \frac{1}{N_{\text{Total}}} \sum_{i=1}^{N_{\text{Total}}} \frac{\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{Q(\boldsymbol{\theta}_i)}, \tag{7.1}$$

and the corresponding uncertainty using

$$\sigma^2[\hat{Z}] = \frac{1}{N_{\text{Total}}(N_{\text{Total}}-1)} \sum_{i=1}^{N_{\text{Total}}} \left[\frac{\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{Q(\boldsymbol{\theta}_i)} - \hat{Z}\right]^2. \tag{7.2}$$

From eq. (7.1), it then follows that the posterior weights are given by

$$p_i = \frac{\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{N_{\text{Total}}Q(\boldsymbol{\theta}_i)}. \tag{7.3}$$

We now combine all of these elements and describe our proposed algorithm.

7.2 Core importance nested algorithm

In this section, we motivate and present the core importance nested sampling algorithm used in `nessai`. We extend the formulation of importance nested sampling described in section 7.1 to allow the use of normalizing flows instead of ellipsoidal bounding distributions. We also draw on the design of diffusive nested sampling where the likelihood constraint is relaxed such that samples are not rejected based on their likelihood.

We start by considering the definition of the evidence from eq. (7.1). In importance nested sampling, the aim is to construct an importance sampling density $Q(\boldsymbol{\theta})$, which we will call *meta-proposal*, from which samples can be drawn, and used to estimate the evidence. The error on this estimate is given by eq. (7.2) and depends on the number of samples N_{Total} and $Q(\boldsymbol{\theta})$. If we consider a fixed number of samples, the meta-proposal that maximizes the effective sample size (ESS) of the set of summands $\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)/Q(\boldsymbol{\theta}_i)$, and therefore provides the most precise evidence estimate, will be $Q(\boldsymbol{\theta}) \equiv \mathcal{L}(\boldsymbol{\theta})\pi(\boldsymbol{\theta})/Z$, i.e. when $Q(\boldsymbol{\theta})$ is equal to the target posterior. Since the evidence is unknown a-priori, the aim is to construct the meta-proposal such that $Q(\boldsymbol{\theta}) \propto \mathcal{L}(\boldsymbol{\theta})(\pi)(\boldsymbol{\theta})$.

This formulation of nested sampling is closely related to Variational Inference [224], where the goal is to approximate a target probability density. In this case, the target density is $\mathcal{L}(\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ and the approximate distribution is the meta-proposal $Q(\boldsymbol{\theta})$. The difference is in how the approximate distribution is obtained. In variational inference, the approximate distribution is optimized by minimizing a variational objective, whereas in this algorithm the distribution is constructed by progressively sampling and adding proposal distributions.

We now consider how to construct the meta-proposal using normalizing flows. An important difference between the ellipsoidal bounds used in `multinest` and normalizing flows is the space over which they are defined. For a normalizing flow, this depends on the domain of the latent distribution $p_{\mathcal{Z}}$. For the typical case of a n-dimensional Gaussian the mapping is defined such that $f : \mathbb{R}^n \to \mathbb{R}^n$, so the flow will have infinite support. We need the meta-proposal to have the same support as the prior, so we include an additional invertible transform that maps from $\mathbb{R}^n$ to a bounded space, such as the Sigmoid $s(x) = [1 - \exp(-x)]^{-1}$. We denote the bounded space $\mathcal{X}$ and the unbounded space $\mathcal{X}'$.

Therefore, instead of considering a series of bounded distributions, we consider a set of N normalized proposal distributions (normalizing flows) $\{q_1, ..., q_N\}$ all defined over the entire prior volume and with corresponding weights α_j defined such that $\sum_{j=1}^{N} \alpha_j = 1$. The overall proposal density as a function of $\boldsymbol{\theta}$ is given by

$$Q(\boldsymbol{\theta}) = \sum_{j=1}^{N} \alpha_j q_j(\boldsymbol{\theta}). \tag{7.4}$$

In practice, in order to sample from $Q(\boldsymbol{\theta})$ we first draw a proposal $k \in \{1, ..., N\}$, drawn from a categorical distribution with category weights $\{\alpha_1, ..., \alpha_N\}$, then a sample is drawn from the sub-proposal $q_k(\boldsymbol{\theta})$.

With this formulation, we can compute an estimate of the evidence for a set of samples drawn from $Q(\boldsymbol{\theta})$ using eq. (7.1) and, as noted in Feroz et al. [212], we no longer require new samples that have monotonically increasing likelihood values. Furthermore, as described in Salomone et al. [216], we do not require that new samples be i.i.d. or distributed according to the likelihood-constrained prior. This removes the need for the rejection sampling that was a bottleneck in the version of `nessai` we described in Williams et al. [1].

We now outline a simplified importance nested sampling algorithm, which we build upon in later sections. The main changes are to the process of removing and replacing the worst point at each iteration. Instead of removing a point and finding a single replacement point, we construct a proposal distribution $q_j(\boldsymbol{\theta})$ based on the points sampled thus far and draw a set of N_j new points $\Theta_j = \{\boldsymbol{\theta}_i\}_{i=1}^{N_j}$ which are added to the overall set of points $\{\Theta_1, ..., \Theta_{j-1}\}$. The meta-proposal $Q(\boldsymbol{\theta})$ is then updated to include $q_j(\boldsymbol{\theta})$ and the evidence is updated. The new importance nested sampling algorithm therefore consists of the following steps:

1. Draw N_{live} points $\{\boldsymbol{\theta}_i\}_{i=1}^{N_{\text{live}}} \sim \pi(\boldsymbol{\theta})$ and compute the likelihood $\mathcal{L}_i = \mathcal{L}(\boldsymbol{\theta}_i)$ of each point,
2. add the next proposal distribution $q_j(\boldsymbol{\theta})$,
3. draw N_j samples from $\Theta_j = \{\boldsymbol{\theta}_i\}_{i=1}^{N_j} \sim q_j(\boldsymbol{\theta})$ and compute the corresponding likelihoods,
4. update the meta-proposal $Q(\boldsymbol{\theta})$ to include $q_j(\boldsymbol{\theta})$,
5. compute the evidence and the corresponding error via eqs. (7.1) and (7.2)
6. repeat steps 2-5 until a stopping criterion is met,
7. compute the final evidence and posterior weights using the independent samples and eqs. (2.38) and (2.39).

This includes an additional step not present in standard nested sampling: redrawing independent samples from the final meta-proposal. Since subsequent proposals are constructed using samples from the previous iterations, new samples are not i.i.d. and eqs. (2.38) to (2.40) do not strictly apply. However, once the meta-proposal is finalized, i.i.d. samples can be sampled and used to compute unbiased estimates of the evidence and posterior weights.

The design of the algorithm hinges on how the next proposal distribution is added, how the number of samples drawn from each proposal (N_j) is determined and how the weights in the meta-proposal $Q(\boldsymbol{\theta})$ are determined. Note that the first proposal distribution $q_0(\boldsymbol{\theta})$ will typically be the prior. We now apply this simplified algorithm to a toy example.

7.2.1 Toy example

In this toy example, we consider a simple problem with an analytic evidence and posterior distribution. We apply the algorithm described in section 7.2 but with some simplifications. This allows us to validate the core algorithm.

We use a 2-dimensional Gaussian likelihood with mean $\mu_{\mathcal{L}} = 0$ and standard deviation $\sigma_{\mathcal{L}} = 1$ and a Gaussian prior a with mean $\mu_{\pi} = 0$ and standard deviation $\sigma_{\pi} = 2$. The posterior distribution is therefore another Gaussian distribution with mean $\mu_{Post} = 0$ and standard deviation $\sigma_{Post} = \sqrt{1/[(1/\sigma_{\mathcal{L}}^2) + (1/\sigma_{\pi}^2)]}$. The evidence is given by a Gaussian distribution with mean μ_{π} and standard deviation $\sqrt{\sigma_{\mathcal{L}}^2 + \sigma_{\pi}^2}$ evaluated at $\mu_{\mathcal{L}}$, so $Z_{\text{Analytic}} = 0.03183$.

To make the comparison between the true and sampled posterior distributions easier, we express the posterior distribution in terms of the log-likelihood $p(\ln \mathcal{L})$. To do this, we note that the posterior distribution defined in terms of the radius squared is $p(r^2) = \chi_2^2(r^2)/\sigma_{Post}^2$ where χ_2^2 is a chi-squared distribution with two degrees of freedom. Then

$$p(\ln \mathcal{L}) = p(r^2) \left| \frac{\partial r^2}{\partial \ln \mathcal{L}} \right|, \tag{7.5}$$

where

$$r^2 = -2\sigma_{\mathcal{L}}^2 \left[\ln(2\pi\sigma_{\mathcal{L}}^2) + \ln \mathcal{L} \right], \tag{7.6}$$

which is defined on $[0, \infty)$ since the maximum possible value of the log-likelihood is $\ln \mathcal{L} = -\ln(2\pi\sigma_{\mathcal{L}}^2)$.

The four steps we must define for the simplified algorithm are: how to construct each proposal distribution, how to determine the number of samples to draw from each proposal, how to determine the weights for each proposal in the meta-proposal and a stopping criterion. For the proposals, instead of normalizing flows, we use 2-dimensional Gaussian distributions $q_j(\boldsymbol{\theta})$ with mean zero and different standard deviations. We determine the standard deviation of each proposal by setting a likelihood threshold $\mathcal{L}_{\text{t}}$ such that 50% of the points from the previous iteration are discarded and then compute the standard deviation of the remaining points. We set the number of samples drawn from each proposal to constant $N_j = N_{\text{live}} = 500$ and set the weights for the meta-proposal α_j to be equal. This means that each proposal will contribute equally to the meta-proposal. Finally, instead of using a stopping criterion, we define a fixed number of proposal distributions (iterations) $N = 4$ where the first is the prior distribution $q_0(\boldsymbol{\theta}) \equiv \pi(\boldsymbol{\theta})$. This is akin to fixing the number of iterations in a normal nested sampling algorithm.Once the final proposal has been added, we draw i.i.d. samples from the finalized meta-proposal and compute the final unbiased evidence estimate and posterior weights.

We present the results obtained with this algorithm in fig. 7.1. This shows the samples

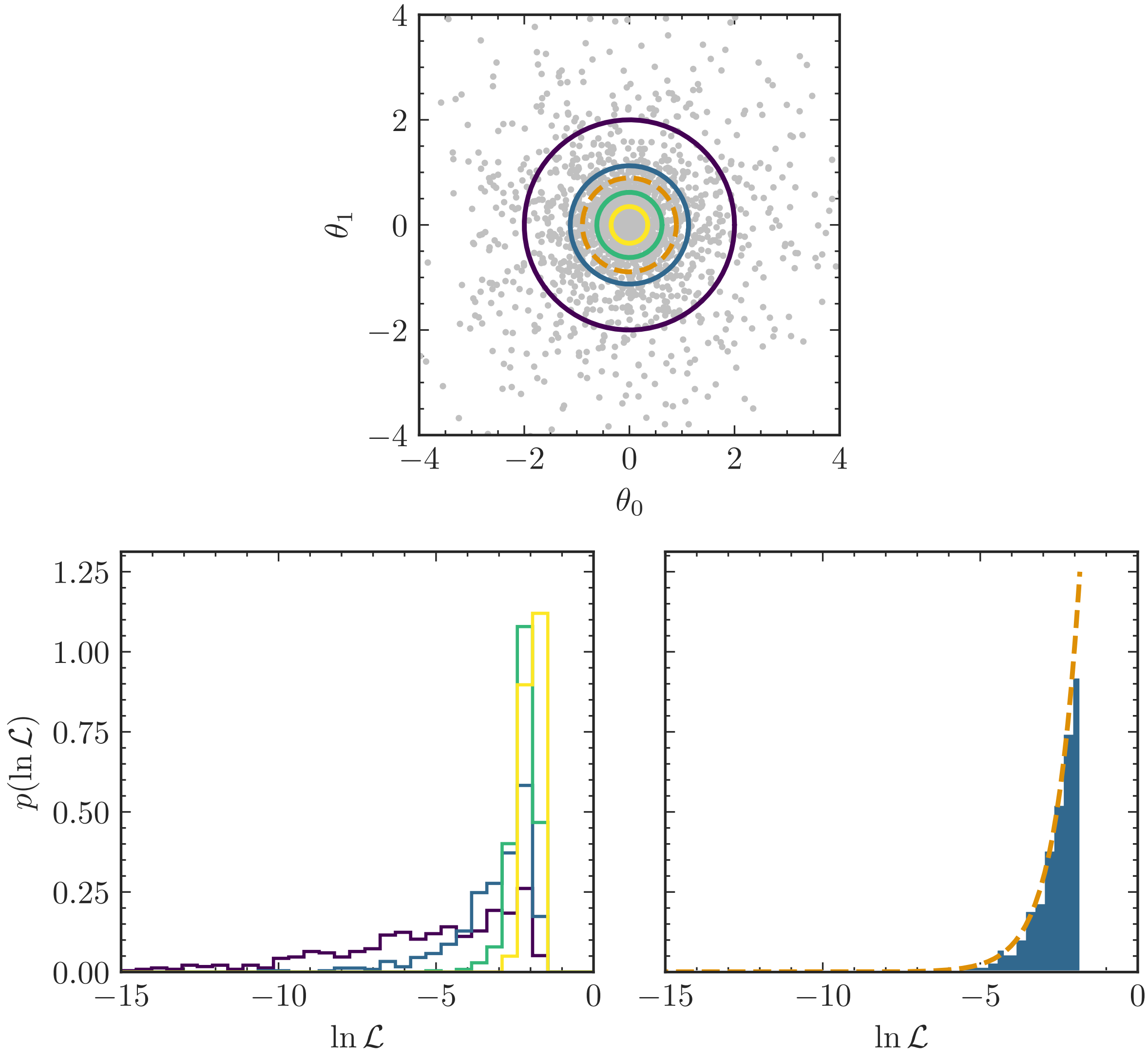


Figure 7.1: Results for the toy example described in section 7.2.1. **Top:** the final samples are shown in grey, the solid lines show the 1-σ contour for each proposal distribution starting with the prior, lighter colours indicate later iterations. The orange dashed line shows the 1-σ contour for the analytic posterior distribution. **Bottom left:** distribution of log-likelihoods for the final samples drawn from each proposal distribution. **Bottom right:** distribution of the log-likelihoods of the final samples weighted by their corresponding posterior weights. The orange dashed line indicates the analytic posterior distribution computed using eq. (7.5).

and the 1-σ contours for each of the proposal distributions, along with the corresponding distribution of log-likelihoods. We compute two evidence estimates: one with the initial samples that are not i.i.d. $\hat{Z} = 0.03177 \pm 0.00042$ and the other with the final i.i.d. samples $\hat{Z} = 0.03191 \pm 0.00042$. We find that both are in agreement with the analytic value, $Z = 0.03183$, but, as we will see in section 7.5.1, the initial estimate will be biased, the bias is just very small in this simple example. This demonstrates that the underlying algorithm can reliably estimate the evidence. We also compute the posterior weights using eq. (7.3) and plot the weighted histogram in log-likelihood space, which shows good agreement with the analytic expression from eq. (7.5). Overall, these results demonstrate the principles of the proposed algorithm and that, for a simple toy example, it converges to the expected result.

7.3 Method

Having outlined the underlying algorithm, we now describe each of the steps in the complete algorithm in detail.

7.3.1 Constructing proposal distributions

With this formulation of nested sampling, the main design choice is how to construct the proposal distribution $q_j(\boldsymbol{\theta})$ at each iteration (step 2). This is akin to drawing new samples in standard nested sampling however, since we no longer require an ordered sequence of points with decreasing prior volume, new points no longer need strictly increasing likelihood values.

The new proposal $q_j(\boldsymbol{\theta})$ at each iteration is defined in terms of a likelihood threshold $\mathcal{L}_\mathrm{t}$: of the current N_live points, M_j are discarded based on a likelihood threshold and the remaining $N_\mathrm{live} - M_j$ points are used to construct the next proposal distribution $q_j(\boldsymbol{\theta})$. In our implementation, this is done by training a normalizing flow. The result is a series of increasingly dense proposal distributions, which is equivalent to the distributions becoming narrower in the log-likelihood space. This is shown in fig. 7.1.

We therefore require a method for determining the likelihood threshold $\mathcal{L}_\mathrm{t}$ used to determine how many points will be discarded before constructing the next proposal distribution. We consider two methods, both of which use weights

$$w_i = \frac{\pi(\boldsymbol{\theta}_i)}{Q(\boldsymbol{\theta}_i)}, \tag{7.7}$$

which quantify the relative importance of each sample $\boldsymbol{\theta}_j$ compared to the prior. Additionally, one could include the likelihood in the weights, however, we leave this for future work.

In the first method, the threshold $\mathcal{L}_\mathrm{t}$ is determined using the $(1-\rho)$ quantile of the likelihood values of the samples from the previous iteration, where ρ is set by the user. To account for non-prior distributed samples used in our algorithm, we use a weighted quantile, where the weights are given by eq. (7.7). This method is based on the standard method used in SMC [216] and diffusive nested sampling [213], but with the addition of the weighted quantile.

The second method we consider is closely related to the first but uses log-weights $\log w_i$ instead of w_i. We consider the normalized sum of $\log w_i$ for the set of N live points ordered by increasing likelihood

$$\lambda(M) = \frac{\sum_{m=1}^{M} \log w_m}{\sum_{i=1}^{N} \log w_i}, \tag{7.8}$$

where M is the number of live points to be discarded. We then determine the value of M at which $\lambda(M) \geq \rho$, for $\rho \in [0,1]$ and set $\mathcal{L}_\mathrm{t} \equiv \mathcal{L}(\boldsymbol{\theta}_M)$. This is analogous to shrinking the log-prior volume by a factor ρ at each iteration whilst also accounting for the different weights of each sample. In practice, since the normalizing flows have support over the entire prior volume, this results in increasing the entropy of $q_j(\boldsymbol{\theta})$. We therefore denote this as the *entropy-based* method to distinguish it from the quantile-based method.

For both methods, we employ a maximum number of live points that can be removed - this prevents the remaining live points being too few to robustly train the next normalizing flow. This maximum together with the value of ρ will determine the total number of samples used in the algorithm. We also employ a minimum number of samples to ensure a minimum change in distribution of training data between subsequent proposals. We discuss the advantages and disadvantages of both methods in appendix F.1.

7.3.2 Training normalizing flows with weights

As discussed in section 2.6.6, it is common practice in SMC to resample at each iteration prior to the mutation step. Different sampling methods can be used, but they all keep the total number of samples constant by including repeated samples. This works when the mutation step is a Markov kernel, but in this work we use a normalizing flow to perform the equivalent of the mutation step and, when training a normalizing flow duplicates in the training data, can be problematic. In extreme cases, where only a few samples are representative, the training data could contain tens of copies of the same sample, which will make training unstable.

Without a step that is equivalent to resampling, deficiencies in training can have a cumulative effect. For example, if the mapping learnt by the normalizing flow $q_j(\boldsymbol{\theta})$ under-samples a region of the space compared to the target, then if another normalizing flow $q_{j+1}(\boldsymbol{\theta})$ is trained with samples drawn using $q_j(\boldsymbol{\theta})$ then $q_{j+1}(\boldsymbol{\theta})$ will also under-sample the same region. To counteract this effect, we include weights in the approximation of

Kullback-Leibler divergence (KLD) used to train the normalizing flow as described in section 3.3.5. To train the j-th flow, we use all samples from the current meta-proposal $Q_{j-1}(\boldsymbol{\theta})$ that satisfy the likelihood constraint $\mathcal{L}(\boldsymbol{\theta}) > \mathcal{L}_{\mathrm{t}}$ and then minimize

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} w_i \log q_j(\boldsymbol{\theta}_i), \tag{7.9}$$

where $q_j(\boldsymbol{\theta})$ is given by eq. (2.6) and w_i are the weights for each sample. In principle these weights could include the likelihood, however in this work we use the weights given by eq. (7.7) which are proportional to the ratio of the likelihood-constrained prior and the likelihood-constrained meta-proposal.

7.3.3 Drawing samples from the proposal distributions

At a given iteration j, once the normalizing flow $q_j(\boldsymbol{\theta})$ has been trained (step 2), we sample from the flow (step 3) and evaluate the likelihood for each new sample. This involves sampling from the latent distribution $p_{\mathcal{Z}}(z)$ and then applying the inverse flow mapping f^{-1} to obtain samples in $\mathcal{X}'$. These samples must then be mapped backed to the original space $\mathcal{X}$, where the likelihood can be computed.

The number of samples drawn at a given iteration N_j should be determined by drawing from a multinomial distribution with N possible outcomes (the number of proposal distributions) and $N_{\text{Total}} = \sum_{j=1}^{N} N_j$ trials, however the weights for each outcome are not known prior to sampling. Instead, we set N_j and determine the weight for the current iteration α_j based on its value. We allow N_j to either be equal to the number of samples removed at that iteration (M_j) or kept constant ($N_j = N_{\text{live}}$). The former will maintain a fixed number of live points N_{live} throughout the run whereas the latter allows for N_{live} to vary. We discuss the consequences of this approximation in sections 7.3.4 and 7.3.7.

Similarly to diffusive nested sampling, all the samples are kept irrespective of their likelihood, which means that samples can 'leak' below the current likelihood threshold.

7.3.4 Updating the meta-proposal

Having drawn samples from the current proposal distribution, the meta-proposal $Q(\boldsymbol{\theta})$ must be updated. The overall form of $Q(\boldsymbol{\theta})$ will depend on the weights α_j that are assigned to each proposal. Whilst adding proposals, we approximate the weights as $\alpha_j \propto N_j$ and normalize them such that they sum to one. This approximation can be corrected for once the sampling has been terminated by fixing the weights to their values from sampling, recomputing N_j by sampling from a multinomial distribution with weights $\{\alpha_0, ..., \alpha_{N_j}\}$ and drawing new samples from each $q_j(\boldsymbol{\theta})$ according to N_j. However, in practice, we find error introduced by this approximation to be significantly smaller than the overall error of the estimated evidence.

7.3.5 Stopping criterion

We define the stopping criterion to be the ratio of the evidence between the live points and the current evidence

$$\text{Condition} = \frac{\hat{Z}_{\text{LP}}}{\hat{Z}}, \tag{7.10}$$

where $\hat{Z}_{\text{LP}}$ is computed using eq. (7.1) and including only the live points in the sum. The algorithm will then terminate when the condition is less than a user-defined threshold τ.

This is more suitable than the fractional change in the evidence between iterations, that is used in standard nested sampling algorithms, because multiple points are removed simultaneously at each iteration, the number of points can vary between iterations and points can leak below the current $\mathcal{L}_{\text{t}}$, which all mean fractional change does not decrease smoothly and instead can fluctuate significantly between iterations.

7.3.6 Posterior samples

Similarly to SMC and `multinest`, our algorithm returns samples $\{\boldsymbol{\theta}_i\}_{i=1}^{N_{\text{Total}}}$ and their corresponding posterior weights p_i given by eq. (7.3). Different methods can then be employed to draw posterior samples. The standard approach in nested sampling is to use rejection sampling [1] or multinomial resampling [197] to resample the nested samples using the posterior weights. Alternatively, the weights can be used directly in weighted histograms or kernel density estimates.

When using multinomial resampling or the weights directly, the posterior samples are not statistically independent, so it is informative to compute Kish's ESS [183] with the posterior weights. This gives an indication of the effective number of posterior samples in the posterior and allows for comparing results obtained via different sampling methods. It can also be used to diagnose poorly converged runs, since a low ESS is an indication that the samples and their corresponding weights are a poor match for the true posterior distribution.

7.3.7 Post-processing

Once sampling is complete, we correct for the approximation of the meta-proposal $Q(\boldsymbol{\theta})$ discussed in section 7.3.4 by redrawing N_{Final} samples from the meta-proposal according the draws from the multinomial distribution. The number of samples can be equal to N_{Total} or can be increased or decreased depending on the desired output.

This has the additional benefit of allowing more samples to be drawn after sampling has completed and can be used to obtain more posterior samples or decrease the estimated error on the evidence.

7.3.8 Complete algorithm

We can now combine all these elements into a complete algorithm, which is shown in alg. 2. The algorithm incorporates normalizing flows but no longer requires that samples drawn from them be i.i.d. according to the prior. Furthermore, samples are drawn and their likelihoods evaluated in batches and all the samples are kept irrespective of their likelihood. Finally, the evidence is a simple sum, so it can be updated for batches of samples. Thus, this algorithm meets all the criteria that were initially set out.

Algorithm 2: Overview of `i-nessai`

```
Input: Likelihood $\mathcal{L}$, Prior $\pi$, Tolerance $\tau$, Method for determining $N_j$, $N_{\text{Final}}$
Output: Evidence $\hat{Z}$, samples $\{\Theta_1, ..., \Theta_j\}$ and posterior weights $W$
$j \leftarrow 1$ ;
$\Theta_1 \leftarrow \{\boldsymbol{\theta}_i \sim \pi\}_{i=1}^{N_1}$;
$N_{\text{Total}} \leftarrow N_1, q_1 \leftarrow \pi$ ;
while $condition \geq \tau$ do
    $j \leftarrow j + 1$ ;
    $q_j \leftarrow$ trained normalizing flow;
    $N_j \leftarrow$ determined via specified method;
    $\Theta_j \leftarrow \{\boldsymbol{\theta}_i \sim q_j\}_{i=1}^{N_j}$;
    $N_{\text{Total}} \leftarrow N_{\text{Total}} + N_j$;
    $\hat{Z} \leftarrow \frac{1}{N_{\text{Total}}} \sum_{i=1}^{N_{\text{Total}}} \frac{\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{Q(\boldsymbol{\theta}_i)}$;
    $W \leftarrow \left\{ \frac{\mathcal{L}(\boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_i)}{N_{\text{Total}} Q(\boldsymbol{\theta}_i)} \right\}_{i=1}^{N_{\text{Total}}}$;
end
Redraw $N_{\text{Final}}$ samples from the final meta-proposal and compute the final
  evidence estimate and posterior weights.
```

7.3.9 Biases

In our algorithm, the proposal distributions (normalizing flows) are trained and then sampled from, rather than being constructed post sampling. This means that, unlike in `multinest`, the meta-proposal distribution is an importance sampling density and eq. (7.2) should give a reliable estimate of the evidence error. We verify this in section 7.5.1.

We also note that a different bias in the evidence arises from evaluating each normalizing flow with samples that were also used to train it. This is necessary since the meta-proposal requires evaluating each normalizing flow on every sample. This is a side effect of the small amount of training data available to each flow and difficulty in setting the hyperparameters for N different normalizing flows prior to sampling. This bias is corrected for when the samples are redrawn as described in section 7.3.7 which we demonstrate in section 7.5.

7.4 Related work

As described in section 7.1, the proposed method draws from existing variations of nested sampling: the soft likelihood constraint from diffusive nested sampling [213], the formulation of importance nested sampling used in `multinest` [212] and the use of normalizing flows as described in Williams et al. [1] and Moss [435]. However, it also has parallels to standard importance sampling and the methods derived from it.

Considering the use of a sequence of normalizing flows to approximate a target (or posterior) distribution, the most closely related works are Nested Variational Inference [447], Annealed Flow Transport Monte Carlo [448] and Preconditioned Monte Carlo [449]. The first is a hybrid between Variational Inference and SMC where a series of parameterized distributions are simultaneously optimized using an annealed version of the target distribution. In the latter two works, the standard SMC algorithm is modified to include an additional step that uses a normalizing flow. Additionally, in Karamanis et al. [449] the authors apply their algorithm to gravitational-wave inference, however only a single simulated event is analysed rather than a set of events.

As with any stochastic sampling algorithm for Bayesian inference, this work can also be compared to simulation-based or likelihood-free inference [355] where the posterior distribution is approximated using repeated simulations of the data instead of evaluating the likelihood. This technique has been applied to data analysis in physics and astrophysics, including but not limited to gravitational-wave data analysis [234, 412, 414, 416], cosmology [450, 451] and particle physics [452]. The approach used in these methods involves training on a dataset that is representative of the entire parameter space and then being able to perform inference for any given point in that space. This is the opposite to the approach employed in this work, where the algorithm is general purpose and is not trained for a specific task but instead is trained on the fly, removing the need for expensive initial training at the cost of being slower when performing inference.

7.5 Results

We present results obtained using the algorithm described in section 7.3.8 on range of problems. We implement the algorithm in the `nessai` software package and it is available at 10.5281/zenodo.4550693 [432]. To distinguish it from the version of `nessai` described in Williams et al. [1], we will refer to it as `i-nessai`.

We run all our experiments using normalizing flows based on real non-volume preserv-

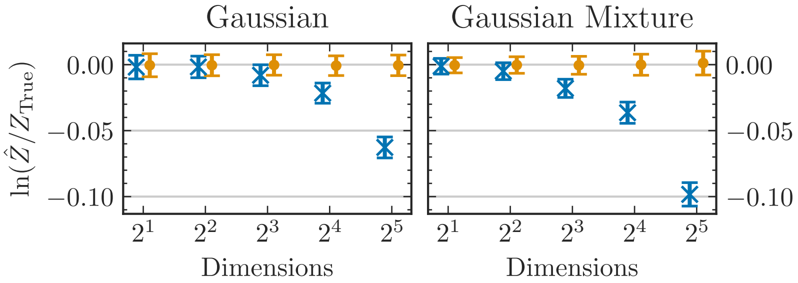


Figure 7.2: Mean estimated log-evidence before (blue cross) and after (orange dot) the resampling step described in section 7.3.7 for an n-dimensional Gaussian and Gaussian Mixture. The error-bars show the mean estimated error for the log-evidence. The estimated evidence has been rescaled using the true value such that the distributions of log-evidences should be centred around zero. The number of samples drawn during the resampling step is set such that is equal to the number of samples accumulated during the initial sampling.

ing (RealNVP) [347] as we find that more complex flows, such as Neural Spline Flows [370], over-fit to the small amount of data available[1] and, compared to the other components of the algorithm, are too computationally expensive to justify using. Furthermore, `i-nessai` requires storing the normalizing flow for each level so using a flow with more parameters can significantly increase the memory footprint of the algorithm.

We start with a series of tests using analytic likelihoods followed by a test using a more challenging likelihood and compare these results to those obtained with `nessai`. We then apply `i-nessai` to two different gravitational-wave analyses. Finally, we investigate parallelization of the algorithm and how it scales with the number of live points.

For all experiments, we use the entropy-based method for constructing each proposal distribution described in section 7.3.1 with $\rho = 0.5$. We discuss this choice in appendix F.1. We also set the number of samples per flow to a constant $N_j = N_{\text{live}}$. Code to reproduce all the experiments is available at 10.5281/zenodo.8124198 [453].

7.5.1 Validation using analytic likelihoods

We start by validating `i-nessai` using likelihoods for which the evidence can be computed analytically in n dimensions. We choose to analyse the simple case of an n-dimensional Gaussian. For a more complex case, we employ the n-dimensional M-component Gaussian mixture likelihood described and used in Moss [435] and Higson et al. [203] which is documented in appendix A.2.

For both likelihoods, we consider $n = \{2, 4, 8, 16, 32\}$ and use uniform priors on $[-10, 10]^n$. The analytical log-evidence for both models is $\ln Z = -n \log 20$. We analyse

[1] A single instance of over-fitting across all the flows will not significantly impact results, however, if the flows consistently over-fit then the final result will be over-constrained.

each likelihood 50 times, including redrawing the samples as described in section 7.3.7, and examine the distribution of the log-evidence estimates and the corresponding estimated error. In fig. 7.2, we include the result of the redrawing of the samples and recomputing the final log-evidence estimate. This shows that without redrawing the samples there is a bias in the estimated log-evidence, however this bias is small compared to the value of the log-evidence, for example, for the 32-dimensional Gaussian and Gaussian Mixture the true log-evidence is -95.86 and the average biases are 0.6% and 0.9% respectively. After redrawing the samples, `i-nessai` reliably estimates the evidence for both models for all values of n. We also compare the distribution of the re-computed log-evidences alongside the expected distribution computed using eq. (7.2) in appendix F.2 and observe that the estimated log-evidence errors agree with the observed distributions.

7.5.2 Comparison with standard nested sampling

We now compare `i-nessai` with standard nested sampling, in particular the standard version of `nessai`. This allows us to verify the results obtained with `i-nessai`, compare the observed and estimated evidences and evidence errors, the number of likelihood evaluations, the wall time and ESS of the posterior distribution. We repeat the analyses described in section 7.5.1 using `nessai` and present the results for both likelihoods in fig. 7.3.

Figure 7.3 shows that `i-nessai` produces estimates of the log-evidence for the Gaussian and Gaussian Mixture that are consistent with `nessai` but have significantly lower variances and the corresponding estimates of the error are correspondingly smaller. We explore how the error on the log-evidence estimate scales in section 7.5.7. Furthermore, fig. 7.3 shows that `i-nessai` requires a comparable number of likelihood evaluations in lower dimensions but more than an order of magnitude less in higher dimensions and a similar trend is seen with the wall time. However, this behaviour is highly dependent on the user-defined settings, which in these experiments were set based on the requirements for the high-dimensional analyses. The ESS of the posterior distribution highlights a notable difference between the two samplers; with `nessai` the ESS increases as the number of dimensions increase for both likelihoods whereas with `i-nessai`, for the Gaussian Mixture likelihood, it decreases in higher dimensions but is still of order 10^4. Since in importance nested sampling the ESS depends on how well the meta-proposal approximates the likelihood times the prior, a lower ESS indicates a 'worse' approximation. In contrast, in standard nested sampling, and therefore `nessai`, the ESS does not depend on the convergence of the sampler and an under- or over-constrained result can still have a large ESS.

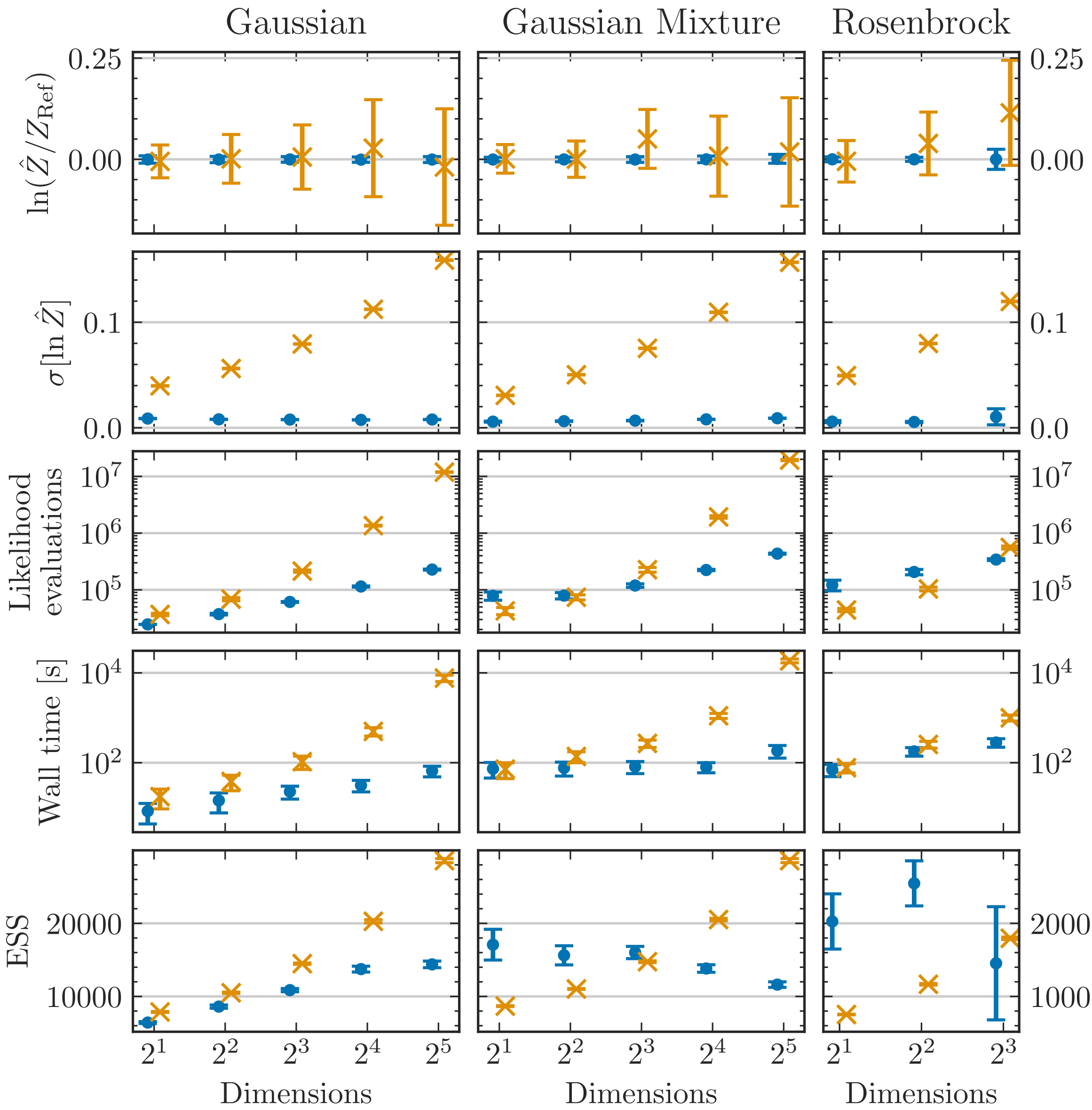


Figure 7.3: Comparison of results produced using `nessai` (orange) and `i-nessai` (blue) when applied to the n-dimensional Gaussian, Gaussian Mixture and Rosenbrock likelihoods as described in sections 7.5.1 and 7.5.3. From top to bottom, results are shown for the final estimated log-evidence rescaled by a reference evidence (the true value for the Gaussian and Gaussian Mixture and the mean value obtained with `i-nessai` for the Rosenbrock), the estimated log-evidence error, the total number of likelihood evaluations, the total wall time in seconds and the ESS of the posterior distribution. Results are averaged over 50 runs with different random seeds for both samplers and the error bars show the standard deviation.

7.5.3 Testing on more challenging likelihoods

To further test `i-nessai`, we consider the n-dimensional Rosenbrock likelihood [454] which has highly correlated parameters and is recognized as a challenging function to sample. We use the more involved variant [440, 455] as defined in appendix A.1 with a uniform prior on $[-5, 5]^n$. We test for $n = \{2, 4, 8\}$ and run `i-nessai` 50 times for each n. Above $n = 2$ there is no analytical solution for the log-evidence of the Rosenbrock likelihood, so we compare results to those obtained with `nessai`. We present these results in fig. 7.3. We observe that `i-nessai` is consistent with `nessai` for $n = 2$ but for $n = \{4, 8\}$ predicts a lower evidence than `nessai`, however the relative difference is less than 1%. The number of likelihood evaluations and wall times are comparable between both samplers but `i-nessai` has a larger ESS in $n = \{2, 4\}$ and lower in $n = 8$. To better understand these differences, we inspect the results obtained with `nessai` and find that the insertion indices [1, 218] are consistent with the results being over-constrained (see appendix F.3). This corresponds to the log-evidence being marginally over-estimated which agrees with the differences in estimated log-evidence observed in fig. 7.3.

7.5.4 Probability-probability test with binary black hole signals

As a more practical test for `i-nessai`, we repeat the analysis used to validate `nessai` in Williams et al. [1], where we used `bilby` [198] and `nessai` to analyse simulated signals from compact binary coalescence of binary black holes injected into 4 seconds of data sampled at 2048 Hz in a three-detector network. For this analysis, we use the same priors (described in Appendix C of Williams et al. [1]) and enable phase, distance and time marginalization in the likelihood. This reduces the parameter space to 12 parameters. We analyse 64 injections simulated from the same priors and produce a probability-probability (P-P) plot and corresponding p-values using `bilby`. This analysis includes the resampling step described in section 7.3.7 and we re-draw the same number of samples that were used in the initial sampling, doubling the number of likelihood evaluations. The probability-probability plot is presented in fig. 7.4 with individual and combined p-values. The combined p-value is 0.3798 which demonstrates that `i-nessai` reliably recovers all 12 parameters. Furthermore, these results are obtained without introducing any of reparameterizations used in standard `nessai` [1] to handle, for example, angles and spin magnitudes.

In fig. 7.5, we show the sampling time and the number of likelihood evaluations required to reach convergence. The median number of likelihood evaluations is 6.5×10^5 and the median wall time is 119 minutes. We also include results obtained using `nessai` and `dynesty` [197], which has been used extensively for gravitational-wave inference [8–10, 122]. Probability-probability plots for both samplers are shown in appendix F.4. We observe that the median reduction in the number of likelihood evaluations are 3 and 13

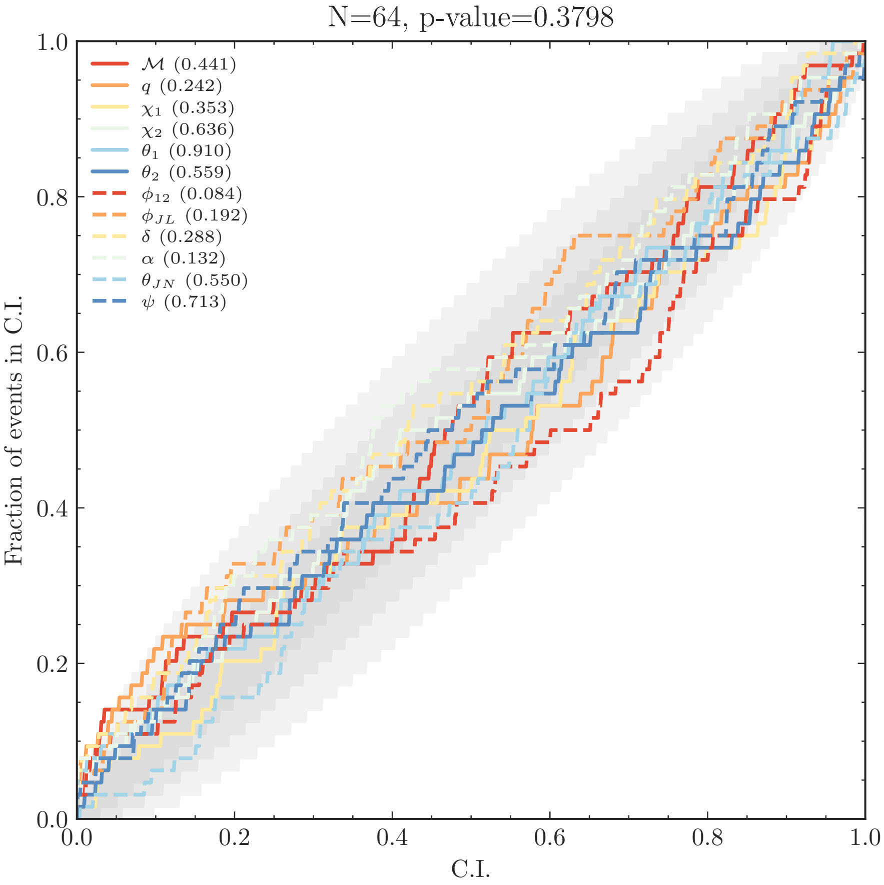


Figure 7.4: Probability-probability plot (P-P plot) for 64 simulated signals from binary black hole coalescence analysed using `i-nessai`. The shaded regions indicated the 1-, 2- and 3-σ confidence intervals. Individual p-values are shown for each parameter, and the combined p-value is also shown.

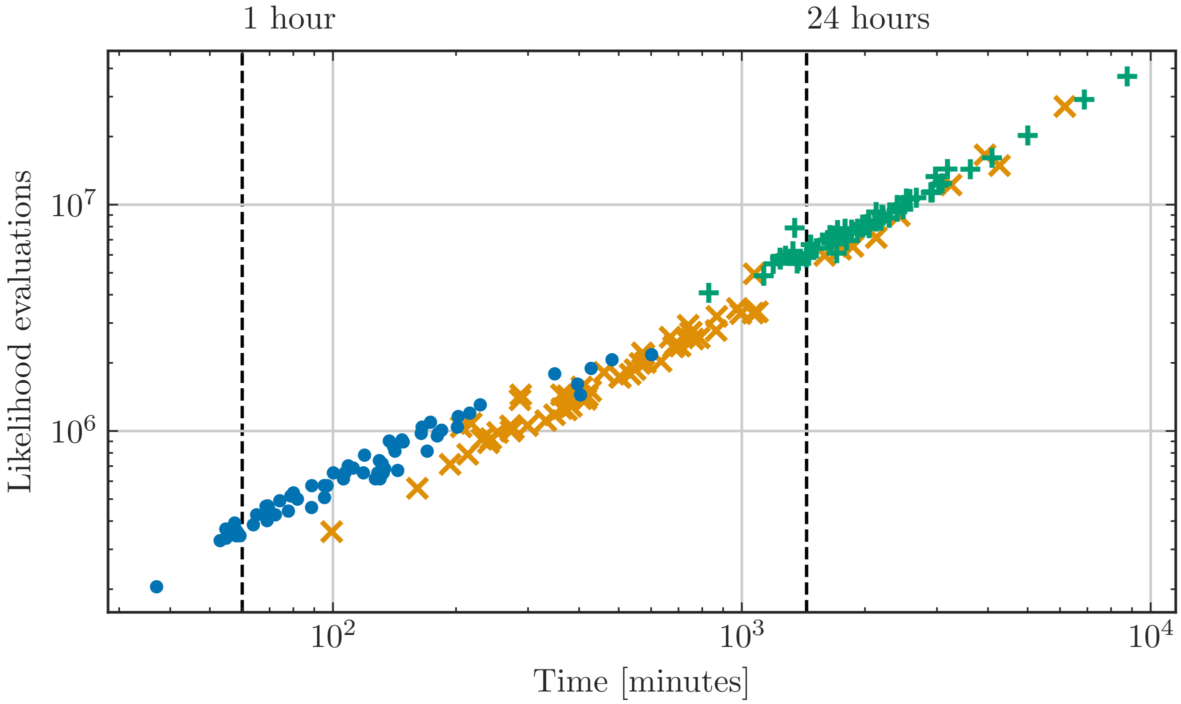


Figure 7.5: Total sampling time versus number of likelihood evaluations for `i-nessai` (blue dots), `nessai` (orange crosses) and `dynesty` (green plus signs) for the 64 binary black hole injections described in section 7.5.4.

for `nessai` and `dynesty` respectively. These equate to reductions in the total wall time of 4 times and 17 times.

7.5.5 Binary neutron star analysis with reduced order quadrature bases

We simulate the signal from a binary neutron star merger similar to GW190425 [262] at a distance of 45 Mpc using `IMRPhenomPv2_NRtidalv2` [456] and inject it into 80 seconds of simulated noise from a two-detector network with aLIGO noise spectral density sensitivity [70] sampled at 8192 Hz. The resulting signal has an optimal network SNR of 30.12.

To analyse the signal, we use `IMRPhenomPv2` [126, 437, 457, 458] with an Reduced-Order-Quadrature (ROQ) basis [260] to reduce the cost of evaluating the likelihood[2]. We also limit the analysis to assume aligned spins and use a low-spin prior a described in Abbott et al. [262]. We run the analysis using `i-nessai`, `nessai` and `dynesty`. We repeat each analysis with four different random seeds and combine the posterior distributions for each seed into a single distribution. We use 16 cores for each analysis to decrease the overall wall time. The settings for `i-nessai` are tuned to ensure that the effective number

[2] We use the ROQ data available at `https://git.ligo.org/lscsoft/ROQ_data`.

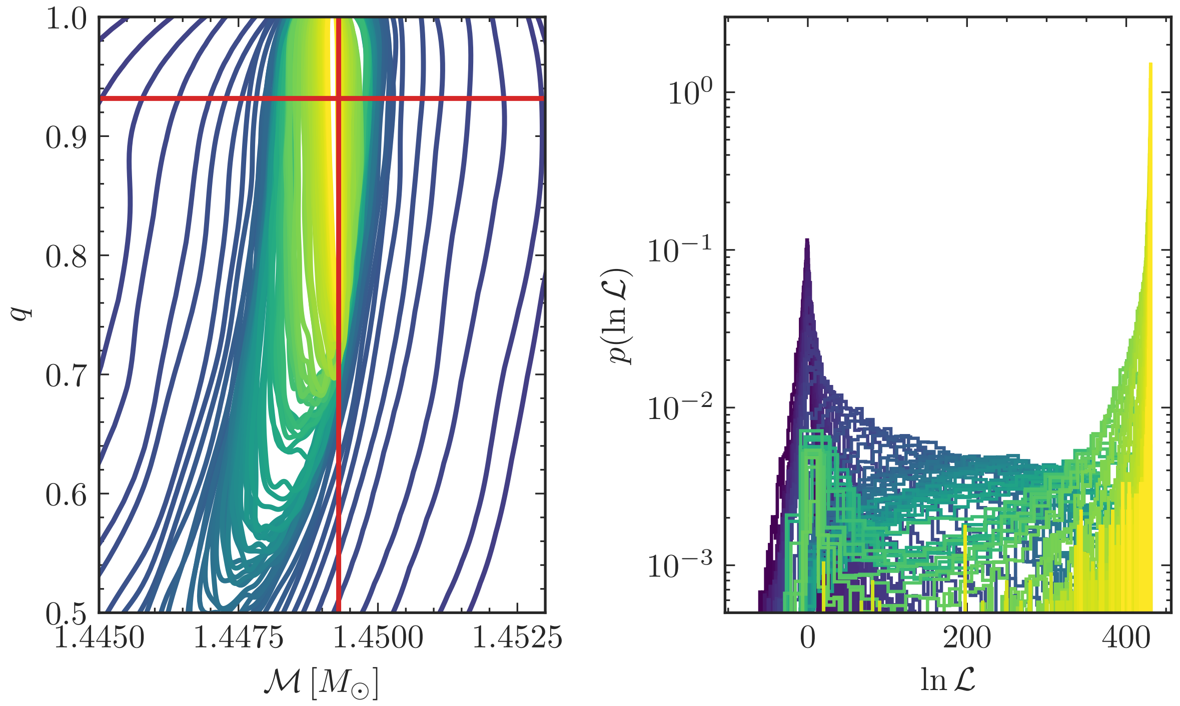


Figure 7.6: Evolution of the proposal distributions ($q_i(\boldsymbol{\theta})$) included in the meta-proposal when performing inference on the binary neutron star injection described in section 7.5.5. Brighter colours indicate later iterations in the algorithm. **Left:** the 90% contours for each of the proposal distributions in the chirp mass-mass ratio space. Only a small region of the parameter space around the highest likelihood is shown. The cross-hair indicates the injected value. **Right:** the distribution of log-likelihoods for each of the proposal distributions.

of posterior samples are comparable to the other samplers.

In fig. 7.6, we show how the meta-proposal evolves as more proposal distributions (normalizing flows) are added over the course of sampling. This shows how the proposals converge around the parameters of the injected signal which correspond to the region with the highest log-likelihood.

To quantify the differences between the results, we compute the Jensen-Shannon divergence (JSD) between the marginal posterior distributions for each parameter as described in Romero-Shaw et al. [122]. We use the threshold described in Ashton and Talbot [233] to determine if the JSD indicate significant statistical differences between the results. We find that all the divergences are below the threshold, except for the in-plane spin χ_1, for which `i-nessai` and `nessai` agree but `dynesty` marginally disagrees with both. We include the complete set of JSDs in appendix F.5 and a corner plot comparing the distributions in fig. 7.7.

We also compare the total number of likelihood evaluations and wall time for each sampler in table 7.1. From these results we see that, on average, `i-nessai` requires 1.4

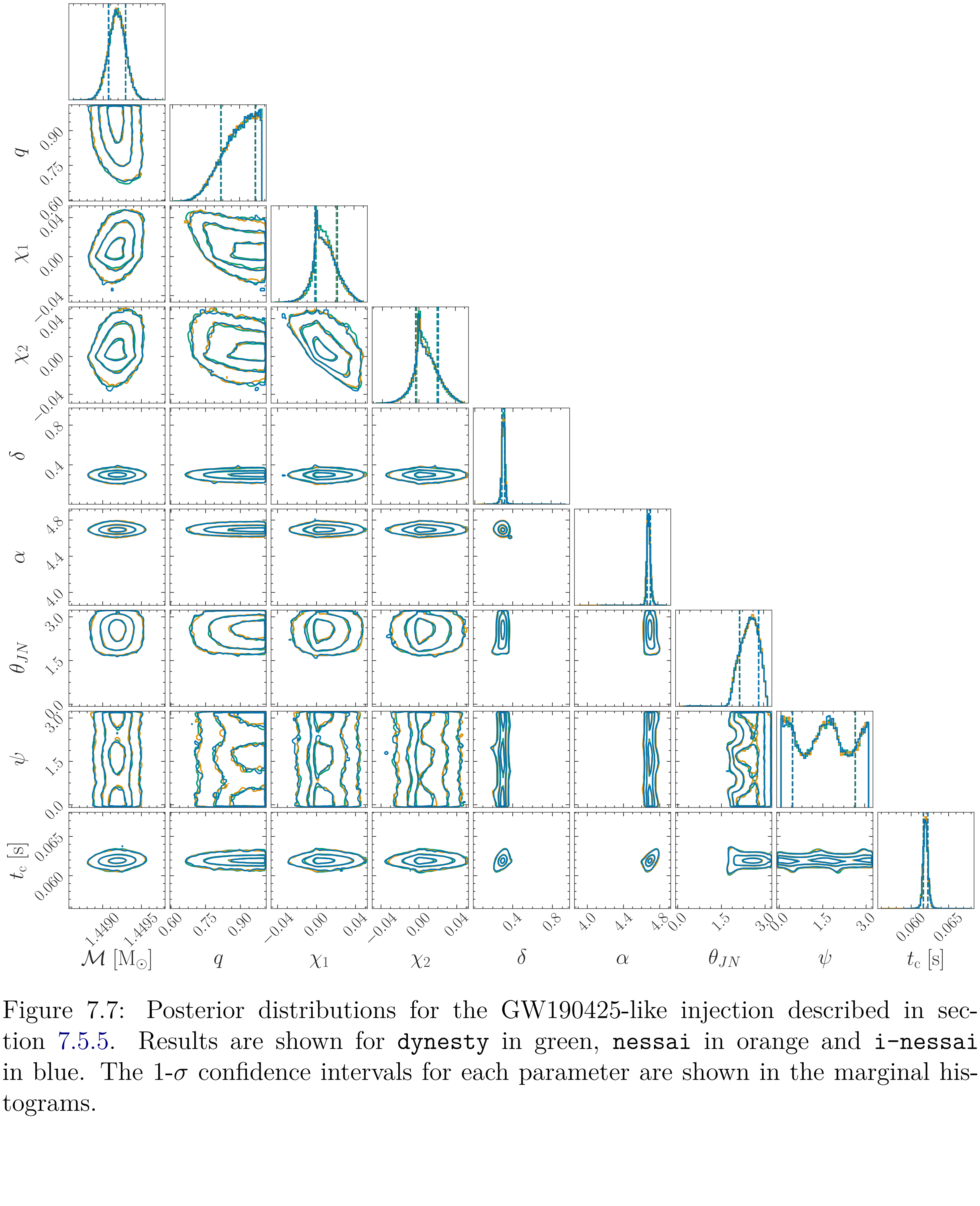


Figure 7.7: Posterior distributions for the GW190425-like injection described in section 7.5.5. Results are shown for `dynesty` in green, `nessai` in orange and `i-nessai` in blue. The 1-σ confidence intervals for each parameter are shown in the marginal histograms.

Table 7.1: Total likelihood evaluations, wall time in minutes and ESS from the posterior distribution for the binary neutron star analysis with ROQs as described in section 7.5.5 for `dynesty`, `nessai` and `i-nessai`. Results are averaged over four runs and the mean and standard deviations are quoted. All analyses were run with 16 cores.

	Wall time [min]	Likelihood evaluations	Effective sample size
`dynesty`	376.3 ± 8.1	$4.30 \times 10^7 \pm 7.12 \times 10^4$	13098 ± 131
`nessai`	57.9 ± 8.9	$1.42 \times 10^6 \pm 1.74 \times 10^5$	13036 ± 45
`i-nessai`	24.3 ± 3.0	$1.01 \times 10^6 \pm 8.99 \times 10^4$	14625 ± 3539

and 42.5 times fewer likelihood evaluations than `nessai` and `dynesty` respectively.

7.5.6 Parallelization

As mentioned previously, the formulation of nested sampling used in this work does not have the same serial limitations of standard nested sampling. The algorithm we present is designed around drawing new samples and evaluating their likelihood in parallel. This leverages the inherently parallelized nature of the normalizing flows. However, the process of training subsequent proposals to add to the meta-proposal is still a serial process.

In standard `nessai`, the costs of rejection sampling and training set an upper limit for the reduction in wall time that can be achieved by parallelizing the likelihood evaluation. However, the total cost of training typically accounted for less than 8% of the total wall time [1]. In `i-nessai`, the rejection sampling step is no longer necessary, so the training is now the main limiting factor and the potential reduction in wall time is far greater. In fig. 7.8, we present results showing how the wall time decreases for an increasing number of cores for one of the binary black holes injections used in section 7.5.4. This shows how initially the wall time is dominated by the cost of evaluating the likelihood but as more cores are added the inherent cost of sampling, which includes training the flows and drawing new samples, becomes the dominant cost. However, in this example, it only accounts for 13% of the total wall time when running on a single core.

7.5.7 Algorithm scaling

In `i-nessai` the number of live points has a different function to that in a typical nested sampler since, in combination with the method used to determine new levels, it will determine how many points are removed at an iteration and how many remain to train the normalizing flow. We previously noted that, for `nessai`, 2000 points were needed for reliable results [1]. We now test `i-nessai` with different values of N_{live} and set the number of samples per flow $N_j = N_{\text{live}}$

We evaluate the scaling of `i-nessai` as a function of N_{live} and present the results in

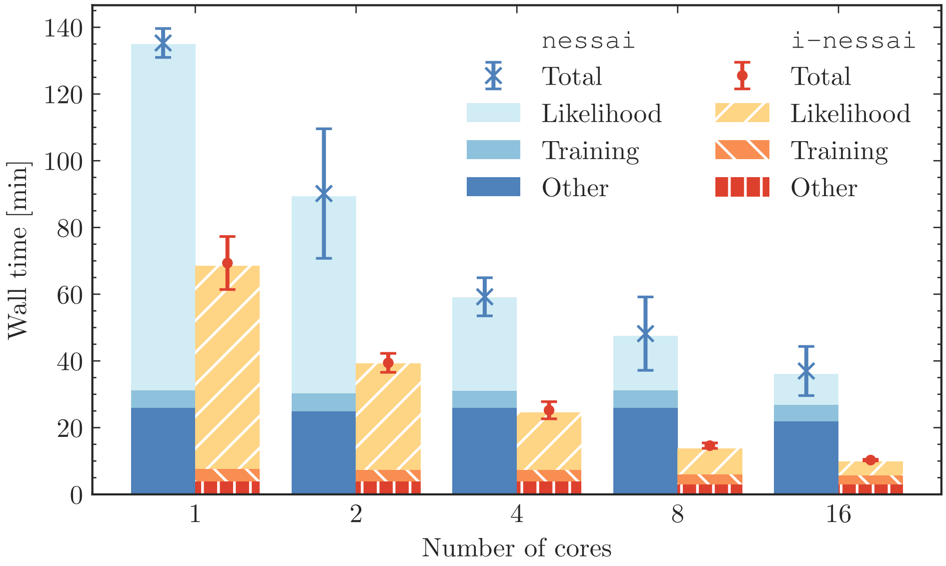


Figure 7.8: Comparison of the wall time spent training the normalizing flows and evaluating the likelihood in `nessai` and `i-nessai` as a function of the number of cores. Results are shown for one of the binary black hole injections described in section 7.5.4 and are averaged over four runs.

fig. 7.9 for a 16-dimensional Gaussian likelihood sampled with $N_{\text{live}} = \{100, 500, 1000, 2000, 4000, 60$ The estimated log-evidence is consistent with the true value for all values of N_{live} and both the observed and estimated standard deviations decrease as N_{live} increases, which is consistent with eqs. (7.1) and (7.2). We observe that the number of likelihood evaluations scales approximately linearly with the number of live points. This contrasts with the wall time which, for a 100 times increase in the number of live points, only increases by 22 times. This is the result of using a likelihood that has a low computational cost, so the cost of running the sampler is dominated by the operations related to the normalizing flow: training, drawing new samples and computing the meta-proposal probability as given by eq. (7.4). In practice, most likelihoods will have a higher computational cost and the wall time will scale approximately linearly with N_{live}.

7.6 Discussion and conclusions

In this work, we present an importance sampling-based nested sampling algorithm, `i-nessai`, that builds on existing work [212, 213, 216] to incorporate normalizing flows and overcome the main bottlenecks in `nessai` described in Williams et al. [1]. The resulting algorithm is a hybrid between standard nested sampling and SMC, where normalizing flows are successively trained and added to an overall meta-proposal that describes the distribution of samples.

We demonstrate that `i-nessai` reliably estimates the log-evidence and associated error

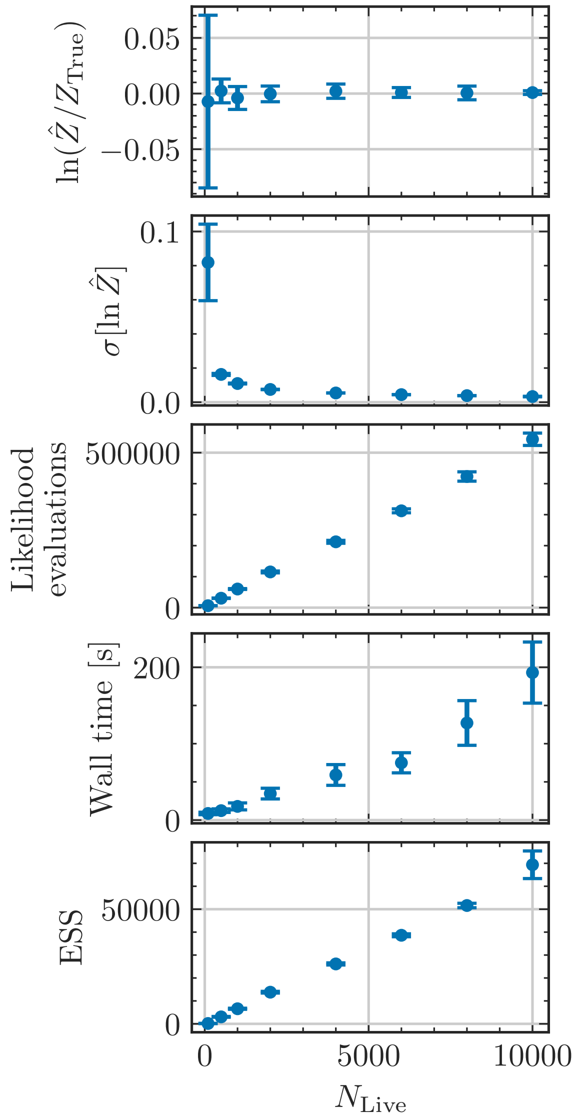


Figure 7.9: Scaling of `i-nessai` as a function of the number of live points N_{live} for an 16-dimensional Gaussian likelihood, as described in section 7.5.1. Results are averaged over 10 runs and the error-bars show the observed standard deviation. From top to bottom the results show the mean estimated log-evidence rescaled by the true value, the mean estimated standard deviation for the log-evidence, the total number of likelihood evaluations, the total wall time and the ESS of the posterior distribution as defined in eq. (2.18).

for Gaussian and Gaussian Mixture likelihoods in up to 32 dimensions. When we compare these results to those obtained with standard `nessai`, we observe that `i-nessai` converges significantly faster and requires fewer overall likelihood evaluations. Furthermore, the observed variance in the estimated log-evidence is consistently less than for `nessai`. This demonstrates that `i-nessai` produces consistent evidence estimates at a fraction of computational cost while also being more precise.

We perform inference on 64 simulated gravitational-wave signals from binary black hole coalescence using `i-nessai` and show that it passes a probability-probability test (fig. 7.4) which indicates that it produces unbiased estimates of the system parameters. Furthermore, these results are obtained without introducing problem specific reparameterizations. Similarly to the analytic likelihoods, we compare these results to those obtained with `nessai` and `dynesty` and observe a median reduction in the number of likelihood evaluations of 3 and 13 times respectively, which equates to a 4 and 17 times reduction in the total wall time.

To further demonstrate the advantages of `i-nessai` compared to standard samplers, we perform inference on a simulated GW190425-like binary neutron star merger using ROQ bases [260] and aligned low-spin priors. The inference completes in just 24 minutes, 2.4 and 15.5 times faster than `nessai` and `dynesty` respectively, while also producing consistent posterior distributions and only requiring 1.01×10^6 likelihood evaluations compared to 1.42×10^6 and 4.30×10^7 respectively.

We also show how the likelihood evaluation can be parallelized in `i-nessai` and find that, once of the cost of evaluating the likelihood becomes negligible, training the normalizing flows and drawing new samples are the main limiting factors. This is in contrast to `nessai`, where performing rejection sampling is the main limiting factor, accounting for approximately 40% of the time when running on a single core. In `i-nessai` training and drawing new samples account for significantly less of the total time. It therefore has improved scaling with respect to the number of cores compared to `nessai`, as shown in fig. 7.8.

A downside of this approach when compared to `nessai` is that the order statistics-based tests proposed in Fowlie et al. [218] and included in `nessai` are no longer applicable since we no longer require points be distributed according to the likelihood-constrained prior. It is therefore harder to identify under- or over-constraining in `i-nessai`. The ESS (eq. (2.18)) can be used to diagnose issues during sampling, however it is not always a reliable diagnostic.

In future work we will consider alternative methods for constructing the meta-proposal which do not rely on discard samples, for example using only the weights in eq. (7.9) and we will explore optimizing the meta-proposal weights after sampling. We will also explore applications of `i-nessai` more complete gravitational-wave analyses like those described in

[8–10] which included calibration uncertainties and waveforms with higher-order modes. Another possible application to explore is model comparison; typically, if we want to obtain a posterior distribution for a different prior than that used for the sampling, the existing posterior samples must be re-weighted using an alternative prior. However, the formulation of the nested sampling in this work would allow for the prior to be changed post-sampling and the evidence recomputed by updating eq. (7.1), so long as the new prior does not extend the boundaries of the prior using during the initial sampling.

In summary, we have introduced an importance nested sampling algorithm, `i-nessai`, that leverages normalizing flows and addresses the bottlenecks in `nessai` [1]. We have demonstrated that `i-nessai` produces results that are consistent with standard nested sampling for a range of problems, whilst requiring up to an order-of-magnitude fewer likelihood evaluations and having improved scalability. Similarly to `nessai`, `i-nessai` is a drop-in replacement for existing samplers, meaning it can easily be used to accelerate existing analyses.

Chapter 8

Applications of nessai

In this chapter, we presented results from analysing gravitational-wave data from O1, O2 and O3 using nessai and i-nessai. We discuss the challenges associated with these analyses, compare our results to those published by the LIGO-Virgo-KAGRA Collaboration (LVK) and highlight some of the challenges associated with analysing data from future ground-based gravitational-wave detectors.

Section 8.1 reviews how calibration errors are accounted for in parameter estimation and presents results demonstrating the most suitable methods for use with nessai. Section 8.2 presents results from analysing binary black holes (BBHs) mergers and binary neutron star (BNS) merger from O1 and O2 using nessai and discusses how these compare to results from Gravitational-Wave Transient Catalog 1 (GWTC-1) and Gravitational-Wave Transient Catalog 2.1 (GWTC-2.1). Section 8.3 presents results for 4 events from O3 that are challenging to analyse: GW190412, GW190425, GW190521 and GW191109, and examines how nessai performs. A limited analysis with i-nessai is then presented in section 8.4 which highlights its potential but also the need for further investigation. Section 8.5 presents results for analyses of simulated data from Einstein Telescope (ET) and Cosmic Explorer (CE) and discussed the challenges associated with these analyses. Finally, section 8.6 summarizes the results and reviews the pros and cons of using nessai when analysing real data.

Thus far, we have introduced `nessai` and various modifications to it, including `i-nessai`. These algorithms have been validated using a mixture of non-gravitational-wave likelihoods, such as the Rosenbrock and Gaussian mixture model, and fiducial gravitational-wave injections. These tests are particularly useful because we can either compute an analytic solution or perform other tests, such as probability-probability plots (P-P plots) to validate the posterior distributions. However, real data presents different challenges that are not captured in these tests and that can make analyses more challenging [94]. In this chapter, we explore the applications of `nessai` and `i-nessai` to real data from LIGO-Virgo-KAGRA Collaboration (LVK) observing runs. We focus predominantly on `nessai` since we have performed more extensive testing with it and have determined the settings necessary for analyses without phase marginalization, as detailed in chapters 5 and 6. In contrast, we have not performed the same ranges of tests with `i-nessai` and the tests presented in chapter 7 were limited to applications with phase marginalization. This is discussed further in section 8.4.

When analysing real data, we will use previous analyses produced by the LVK as point of reference to compare to. In most cases, this involves comparing the posterior distributions using Jensen-Shannon divergence (JSD) as described in section 2.9.2. However, such comparisons are not always possible due to differences in the underlying assumptions that have been made, such as priors or choices of waveform approximants.

8.1 Calibration errors

When considering real data, we must account for the systematic errors caused by the imperfect nature of the detector calibration, which will affect the observed strain. Different approaches can be used to model the error [96, 122, 459], but recent analyses, such as Gravitational-Wave Transient Catalog 2.1 (GWTC-2.1) [9] and Gravitational-Wave Transient Catalog 3 (GWTC-3) [10] follow Farr et al. [460] where the errors are treated as frequency dependent phase and amplitudes errors ($\delta\phi(f)$ and $\delta A(f)$) such that the observed strain as function of frequency is

$$\tilde{d}_{\mathrm{obs}}(f) = \tilde{d}(f)[1 + \delta A(f)] \exp[i\delta\phi(f)], \tag{8.1}$$

where $\tilde{h}(f)$ is the true strain. These errors are expected to be small and vary smoothly [460] and must be included in the likelihood when analysing real data. They are typically modelled using cubic spline polynomials with 10 nodes for phase and amplitude per detector [122], thus adding 20 parameters to sample per detector in the network. Since the errors are expected to be small, the priors are assumed to be Gaussian with means and standard deviations given by a calibration envelope [461, 462]. Recent studies suggest that the effects of these errors are negligible with current detector sensitives but will be

Table 8.1: Run statistics for the analysis of GW150914 using `nessai` when not accounting for calibration errors compared to sampling the calibration parameters or using calibration reweighting. The number of posterior samples is shown for the combined results, whereas the number of likelihood evaluations and wall time are shown for both parallel runs.

	Posterior samples	Likelihood evaluations	Wall time [min]	Reweighting time [min]
No calibration	8230	2 713 780 2 997 256	99 117	-
Sample	8839	7 983 618 7 018 733	574 864	-
Reweight	3369	1 967 937 2 623 297	74 96	< 1

more relevant in next generation detectors [463, 464]. Payne et al. [464] also proposed using importance sampling to reweight results obtained without modelling calibration errors and thereby include the errors. This allows analysis to be performed without sampling the additional parameters that are necessary when modelling the calibration errors, and can reduce computational cost. We consider both reweighting and sampling for use with `nessai`.

8.1.1 Calibration errors with `nessai`

The analyses with `nessai` presented so far have not accounted for calibration errors since they used simulated data. However, tests on non-gravitational-wave likelihoods, e.g. in section 7.5.1, have shown that increasing the number of dimensions being sampled requires adjustments to the settings being used. We now test sampling the calibration parameters with `nessai` and using calibration reweighting and use these results to inform further analyses.

We analyse GW150914 using `nessai`; once sampling the calibration errors, once using calibration reweighting and once without accounting for calibration. We model the signal with `IMRPhenomPv2` and use the calibration data provided with GWTC-2.1 [9]. We run the analyses with `bilby` and `bilby_pipe` [198] on 16 cores. We perform two parallel runs per analysis and combine the final results to obtain a single set of the posterior samples for each of treatment of the calibration errors.

We compare the posterior samples and find that the reweighting step has more than halved the final number of samples compared to the other runs, as shown in table 8.1. However, it has minimal impact on the overall wall time and number of likelihood evaluations, whereas sampling the calibration parameters significantly increases both, see table 8.1. A subset of the posterior distributions for each run are shown in fig. 8.1. As expected, the differences between the posterior distributions that account for calibration errors and those that do not are minimal. The most notable, though overall minor, change when accounting for calibration is to the sky location (α, δ) which is due to the correlations between the location and other parameters, such as the inclination, that are affected by the amplitude calibration errors [465].

We also examine the diagnostics produced by `nessai` and compare them with and without sampling the calibration errors. The insertion indices for one of the runs sampling the calibration errors indicate the result may be slightly over-constrained, whereas the other does not, see appendix G.1. This suggests that whilst the results presented in fig. 8.1 show reasonable agreement, the settings may not be generally robust. In fig. 8.2, we compare a subset of the statistics tracked by `nessai` over the course of sampling. These show how the additional parameters that are included when sampling the calibration errors significantly reduce sampling efficiency. Both the acceptance based on the likelihood threshold and for the rejection sampling are consistently lower, which increases the overall wall time, shown in table 8.1.

In summary, sampling the calibration parameters significantly increases the cost of performing analyses with `nessai` whilst also requiring different sampling settings. In contrast, calibration reweighting does not impact sampling, allowing for the use of the same settings and the reweighting state can be parallelized to reduced wall time. This comes at the cost of the final number of posterior samples. However, the reduction in number of samples can be offset by running more parallel analyses whilst still remaining faster than sampling the parameters directly. We therefore opt to perform the analyses using calibration reweighting instead of sampling the calibration parameters

8.2 Analysis of O1 and O2 events

Eleven gravitational-wave transients were detected in the first two LIGO-Virgo Collaboration (LVC) observing runs, O1 and O2 [7]. Ten of these were identified as binary black holes (BBHs) and the remaining signal was the first binary neutron star (BNS) detection, GW170817 [29]. These events have since be been reanalysed multiple times with different inference algorithms and waveform approximants [7, 9]. Here, we focus on the results from Gravitational-Wave Transient Catalog 1 (GWTC-1) [7] and GWTC-2.1 [9] and compare them to results obtains with `nessai`. For all analyses, we use the data and calibration envelopes described in GWTC-2.1 [9]

8.2.1 BBH analyses

The analyses in GWTC-1 used `LALInference` [96] with `IMRPhenomPv2` [457, 458] for the BBHs events. In contrast, the analyses in GWTC-2.1 used `bilby` and `bilby_pipe` [198], with the `dynesty` sampler [197], and the more recent `IMRPhenomX` family of waveforms [118], specifically, `IMRPhenomXPHM` which includes precession and higher-order modes [118]. This

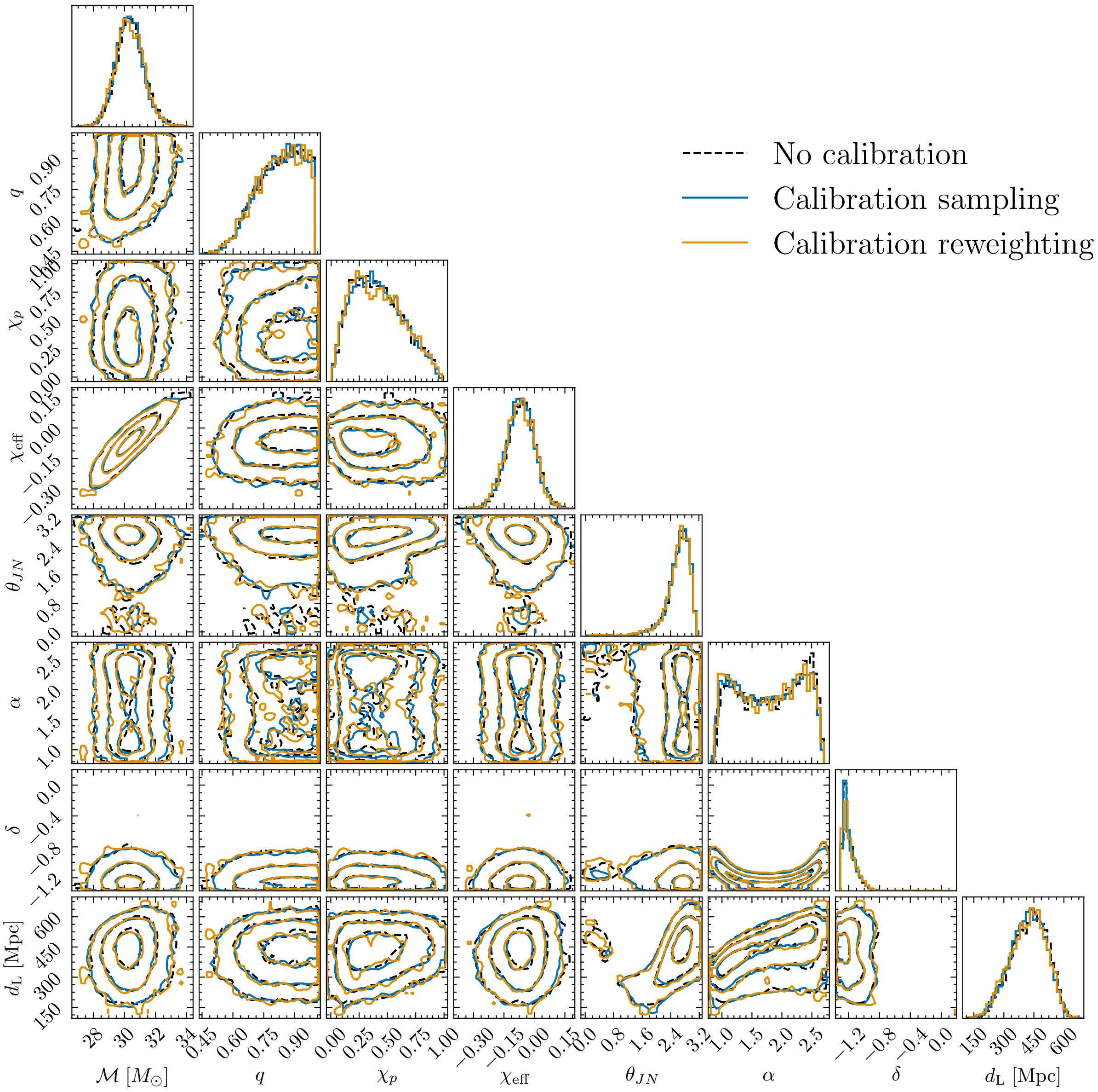


Figure 8.1: Corner plot showing a subset of the posterior distributions for GW150914 obtained using `nessai` with three different treatments for the calibration errors: sampling the calibration errors (blue), calibration reweighting (orange), and not accounting for calibration errors (black dashed line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

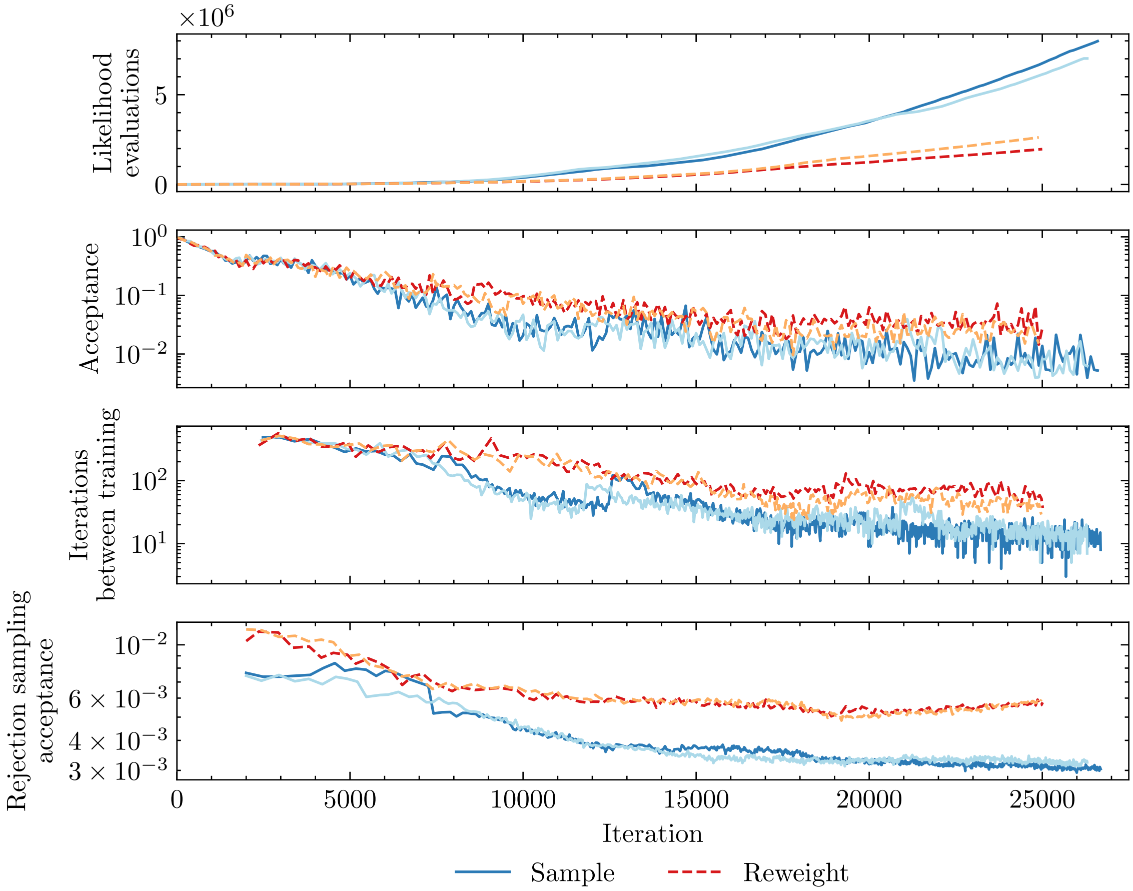


Figure 8.2: Comparison of diagnostics produced by `nessai` when sampling the calibration errors (blue solid lines) compared to using reweighting (red dashed lines). Results are shown for two different runs. From top to bottom: the cumulative number of likelihood evaluations, the acceptance based on the likelihood threshold, the number of iterations between training the normalizing flow and the acceptance when performing rejection sampling to ensure samples are distributed according to the prior. The flow is trained for the first time at the 2000th iteration.

gives us a range of results to compare to and provides context for the statistical variations we expect to see between different inference codes and samplers.

We use bilby and bilby_pipe for the analyses with nessai and base the configuration for each event on analyses performed in GWTC-2.1 since it also used bilby and bilby_pipe. We use calibration reweighting to account for the calibration error and perform two parallel runs per event which are then combined to obtain a single result. We analyse each event using three different waveform approximants: IMRPhenomPv2, IMRPhenomXP and IMRPhenomXPHM. We opt to use phase marginalization with IMRPhenomPv2 and IMRPhenomXP, under the assumption that the effects of precession will be negligible, and disable it for IMRPhenomXPHM since it contains higher-order modes. Results obtained with IMRPhenomPv2 will mostly closely match the results in GWTC-1, however, differences may arise due to differences between LALInference and bilby and the use of phase marginalization, which the GWTC-1 analyses did not use. Results obtained with IMRPhenomXPHM should be consistent with GWTC-2.1 since it also used bilby, so any differences can be attributed to differences between nessai and dynesty. Finally, results obtained with IMRPhenomXP will differ from GWTC-1 because of changes to the waveform and inference pipeline, but will also differ from GWTC-2.1 in cases where higher-order modes are present or the effect of precession is large.

Comparison to GWTC-1 results

We compare the results obtained using nessai with IMRPhenomPv2 to the samples released with GWTC-1 [466]. These only contain a subset of the usual 15 parameters over which inference is performed, which limits the comparisons that can be made. We compute JSDs between the nessai results with IMRPhenomPv2 and IMRPhenomXP and the GWTC-1 results and present them in fig. 8.3. The average overall JSD is 0.011 bits whilst, following Ashton and Talbot [233], the average threshold that indicates agreement between the posteriors, given the number of posterior samples, is almost an order of magnitude smaller at 0.0017 bits. These results show that there are significant differences between the GWTC-1 and IMRPhenomPv2 results. We note, however, that the priors on chirp mass used for the two analyses are different; they differ in the range of chirp masses but also in the allowed component and total mass. More details are provided in appendix G.2. The event with the lowest average JSD is GW170809, and the event with the highest average JSD is GW150914. We present the corner plots for these two events in figs. 8.4 and 8.5, and also include the posteriors obtained with IMRPhenomXP.

For GW150914 there is a noticeable difference in posterior distribution for the chirp mass and mass ratio, with the GWTC-1 results favouring higher chirp masses and equal masses. To explore this difference in more detail, we perform additional analyses with priors that are consistent with GWTC-1 and disable phase marginalization, but the results

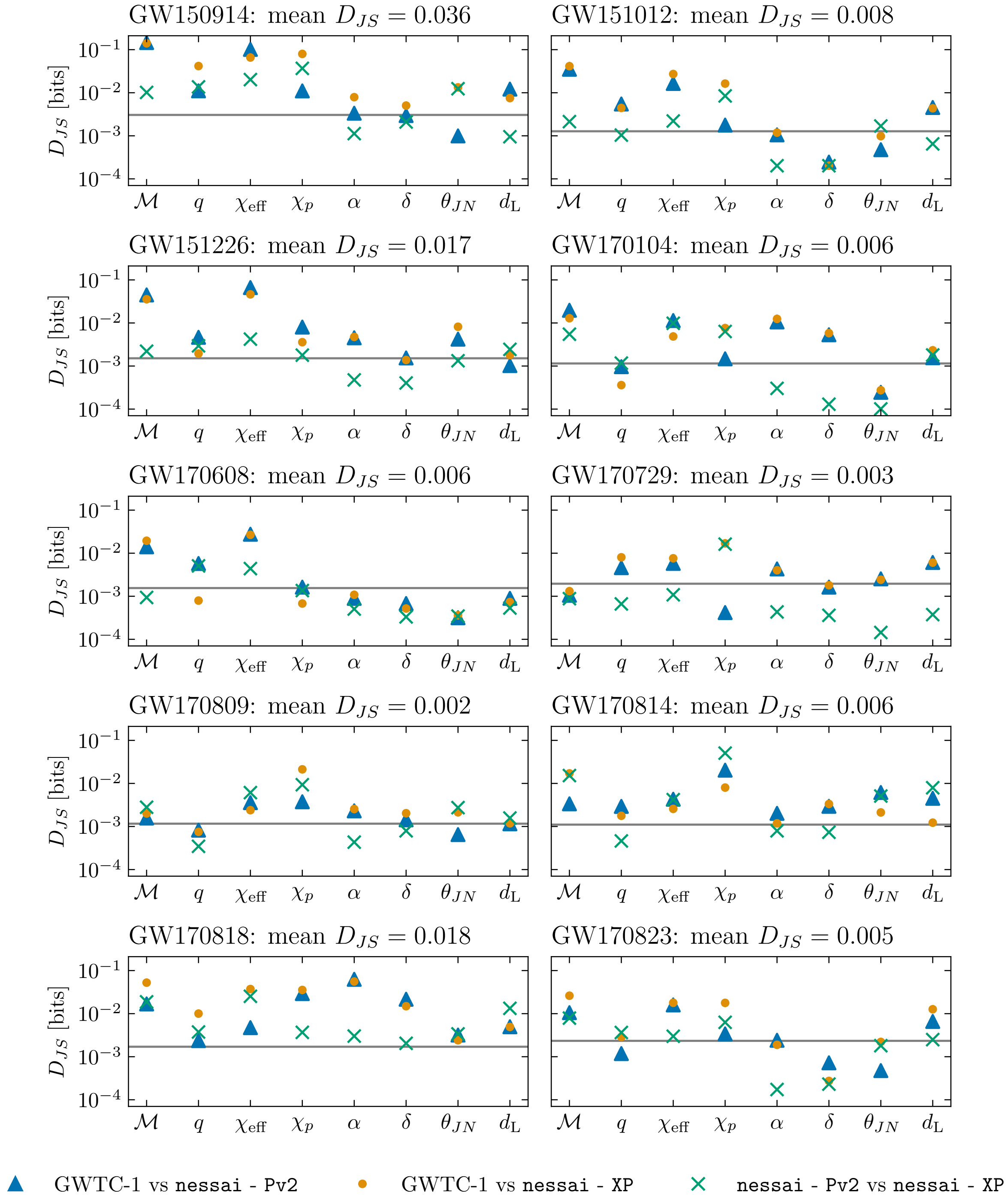


Figure 8.3: JSD in bits between GWTC-1 posterior samples and posterior samples produced by `nessai` with `IMRPhenomXP` and `IMRPhenomXP` for the 10 BBH events from O1 and O2. The mean JSD in bits for each event is shown above each plot. The horizontal grey line shows the $10/n$ samples threshold, where n is the minimum number of samples between the three sets of posterior samples for each event.

obtained are still inconsistent with those with GWTC-1, see appendix G.3. We therefore attribute this to differences in the configuration, data (e.g. power spectral densitys (PSDs)) and overall pipelines being used (`LALInference` vs `bilby`). This result also highlights a difference between `IMRPhenomPv2` and `IMRPhenomXP`; the latter has more support for larger values of χ_p, whereas the former is broad agreement with the GWTC-1 result. This is consistent with the results reported in Pratten et al. [120], where the authors observe similar differences between `IMRPhenomPv2` and `IMRPhenomXP`.

For GW170809, as quantified by the JSD, the posterior distributions show significantly better agreement than for GW150914. The estimates of the masses are consistent across both GWTC-1 and `nessai` with `IMRPhenomPv2` but also between `IMRPhenomPv2` and `IMRPhenomXP`. There are however visible differences in the sky location returned by `nessai` and GWTC-1, though this may be in part due to large differences in the number of samples being plotted; more than 50 000 for GWTC-1 and only ~ 8000 for both `nessai` analyses. Whilst the results for `IMRPhenomPv2` and `IMRPhenomXP` share the same support for the masses, the `IMRPhenomXP` results again show more support for higher values of χ_p.

If instead of per event, we consider the mean JSD per parameter, we find that chirp mass has the highest JSD of 0.0294 bits whilst the inclination θ_JN has the lowest at 0.0019 bits. The 1-dimensional marginal chirp mass posterior distributions for each event are shown in fig. 8.6. Most of the posteriors show minor differences; slight shifts in the median values or differences in width of the posteriors, the one exception to this is GW151012 where the secondary higher-mass mode has more posterior mass compared to GWTC-1.

Overall, we find consistent differences between the results obtain with `nessai` and those from GWTC-1, most notably in the estimated chirp mass. For GW150914, we explore these discrepancies in more detail and find that changing the priors and marginalizations cannot fully account for them. We therefore attribute them to differences in the configuration, the analysis pipelines and possibly data being used. Comparing runs with `IMRPhenomPv2` and `IMRPhenomXP` highlights cases where changes in the waveform lead to noticeable differences in the posterior distributions, most notably for the spin parameters.

Comparison to GWTC-2.1 results

We present the results obtained using `nessai` with `IMRPhenomXP` and `IMRPhenomXPHM` and compare them to samples released alongside GWTC-2.1 [467]. We compute JSDs between the three results for a subset of the parameters and present them in fig. 8.7. The results using `IMRPhenomXPHM` are directly comparable to those from GWTC-2.1 and, unlike the previous comparison to GWTC-1, all the results have a mean JSD below 0.005 bits. This implies that the results obtained with `nessai` are consistent with those published in GWTC-2.1. Since phase was sampled rather than marginalized, this also implies that

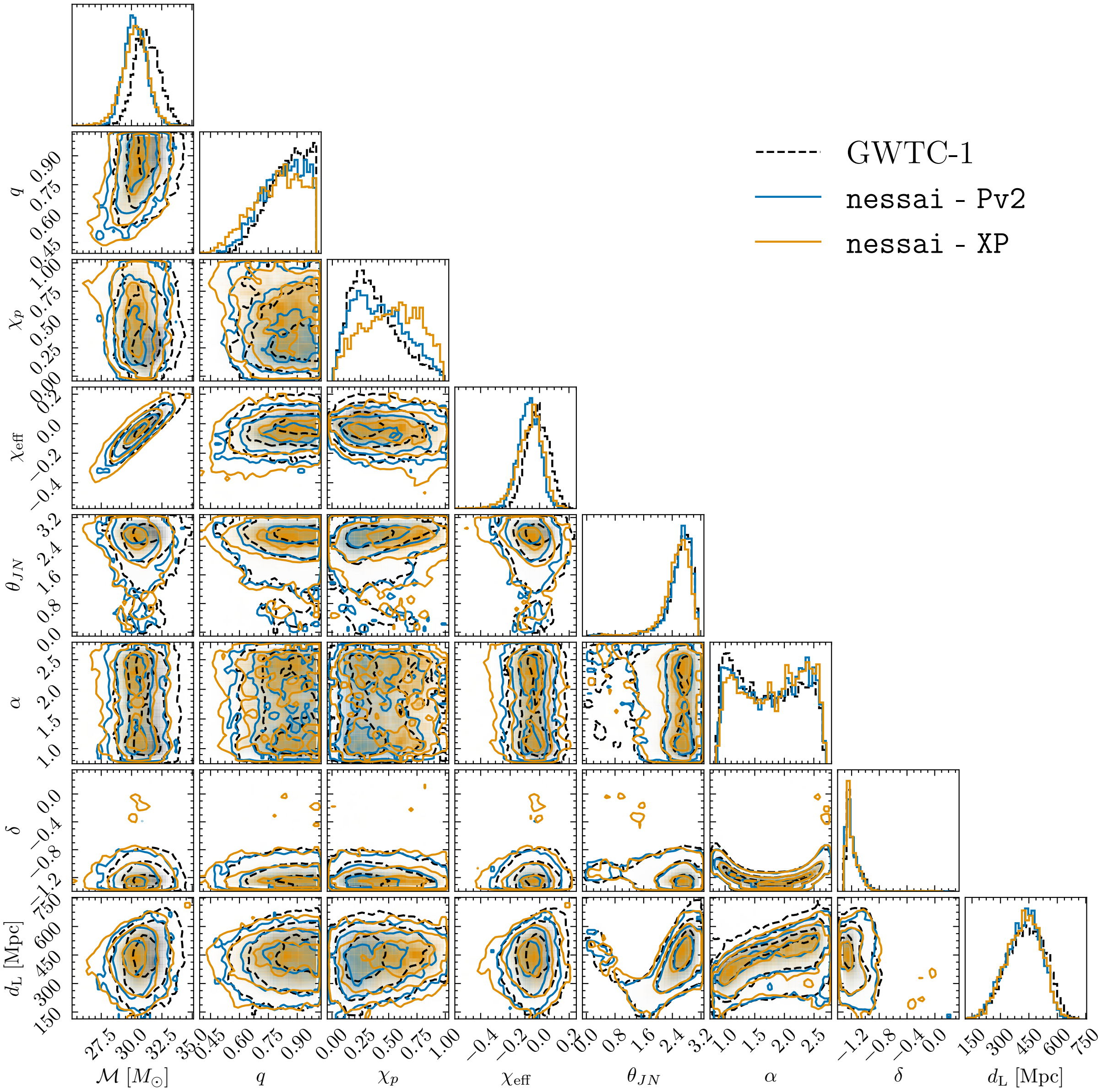


Figure 8.4: Corner plot showing the posterior distributions for GW150914 from GWTC-1 (black dashed line), and two analyses using `nessai`: one with `IMRPhenomPv2` (blue solid line) and another with `IMRPhenomXP` (orange solid line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

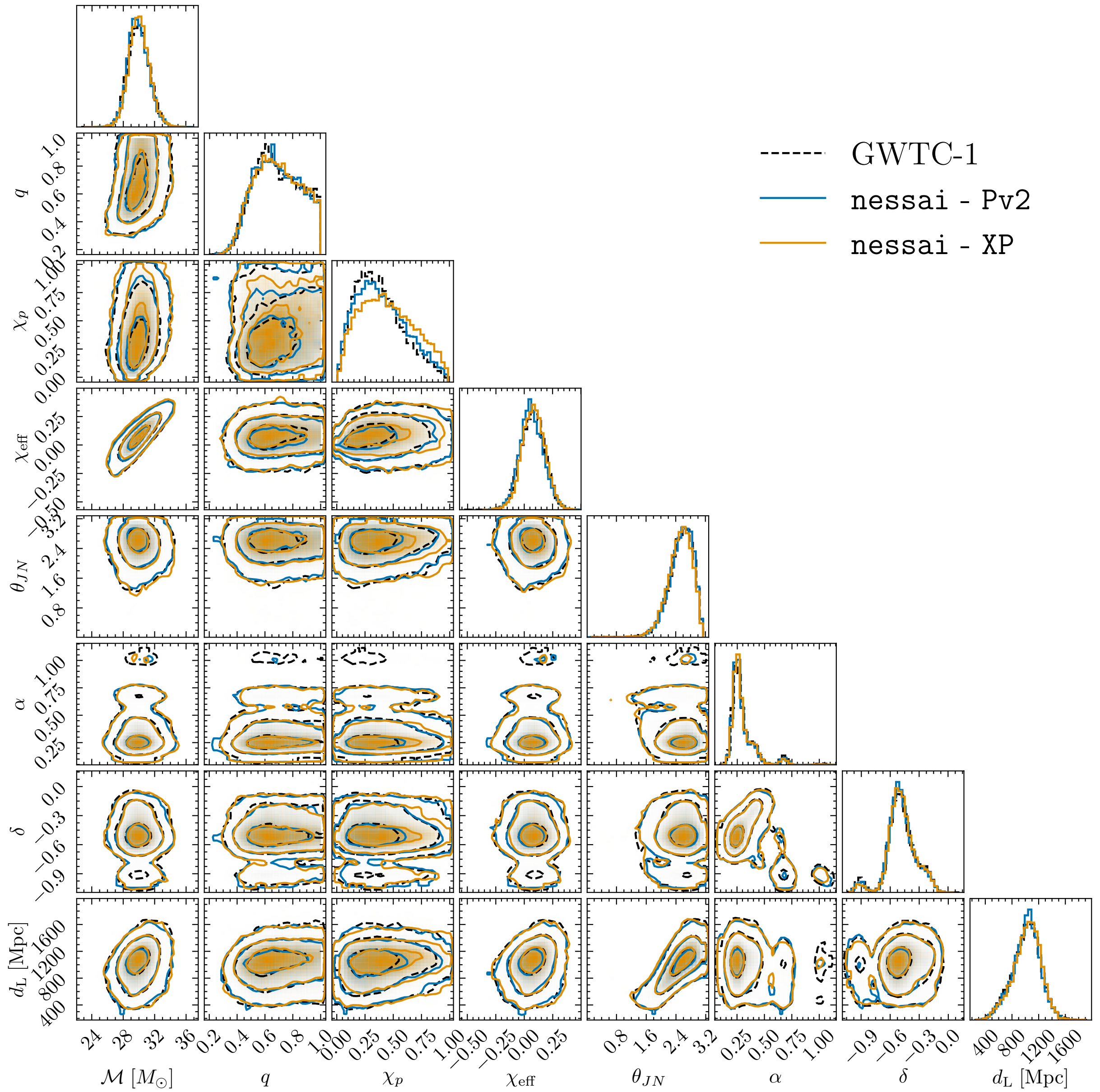


Figure 8.5: Corner plot showing the posterior distributions for GW170809 from GWTC-1 (black dashed line), and two analyses using `nessai`: one with `IMRPhenomPv2` (solid blue line) and another with `IMRPhenomXP` (orange solid line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

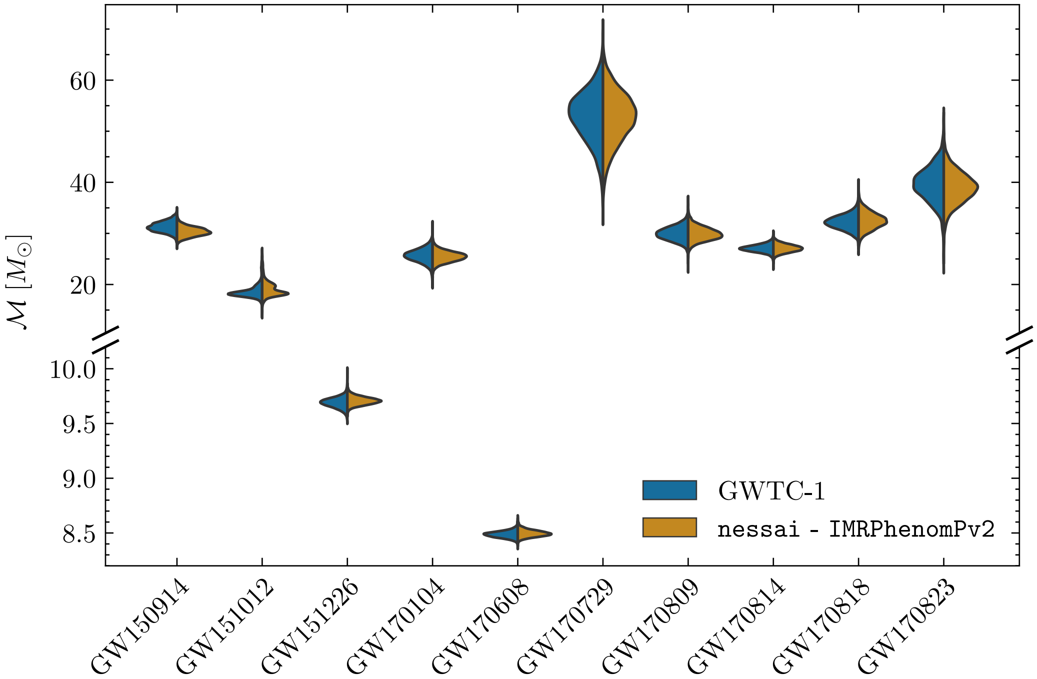


Figure 8.6: Posterior distributions for chirp mass for the 10 BBH events from O1 and O2 comparing GWTC-1 and an analysis using `nessai` with `IMRPhenomPv2`. The GWTC-1 samples are shown in blue on the left of each violin plot and the `nessai` sample are shown in orange on the right.

the settings proposed in chapter 6 are suitable for such analyses. However, similar to the results for GWTC-1, the choice of settings and use of calibration reweighting has resulted in comparatively low number of posterior samples. The event with the highest average JSD is GW170729 and lowest corresponds to GW170814. Corner plots for both events are shown in figs. 8.8 and 8.9 and include results obtained with `IMRPhenomXP` with phase marginalization.

GW170729 has the highest average JSD of all the events at 0.0046 bits, with the highest individual JSD of 0.0154 bits for the right ascension, a pattern that repeats across multiple events. Despite this, the posterior distribution in fig. 8.8 still show good agreement between results from `nessai` with `IMRPhenomXPHM` and GWTC-2.1, including for the highly multimodal posterior over right ascension and declination. However, the posteriors for `nessai` with `IMRPhenomXPHM` appear, in both the 1- and 2-dimensional marginal distributions, to be narrower than the results from GWTC-2.1. This suggests that the results from `nessai` may be slightly over-constrained. However, these differences result correspond to only slight variations in the confidences intervals, for example the median detector frame mass of the heavier component is estimated to be $m_1 = 77.21^{+15.72}_{-14.50}$ M$_\odot$ for GWTC-2.1 and $m_1 = 77.29^{+15.30}_{-14.20}$ M$_\odot$ for `nessai` with `IMRPhenomXPHM`. Conversely, the results with `IMRPhenomXP` and phase marginalization once again have more support for equal masses and larger values of χ_p.

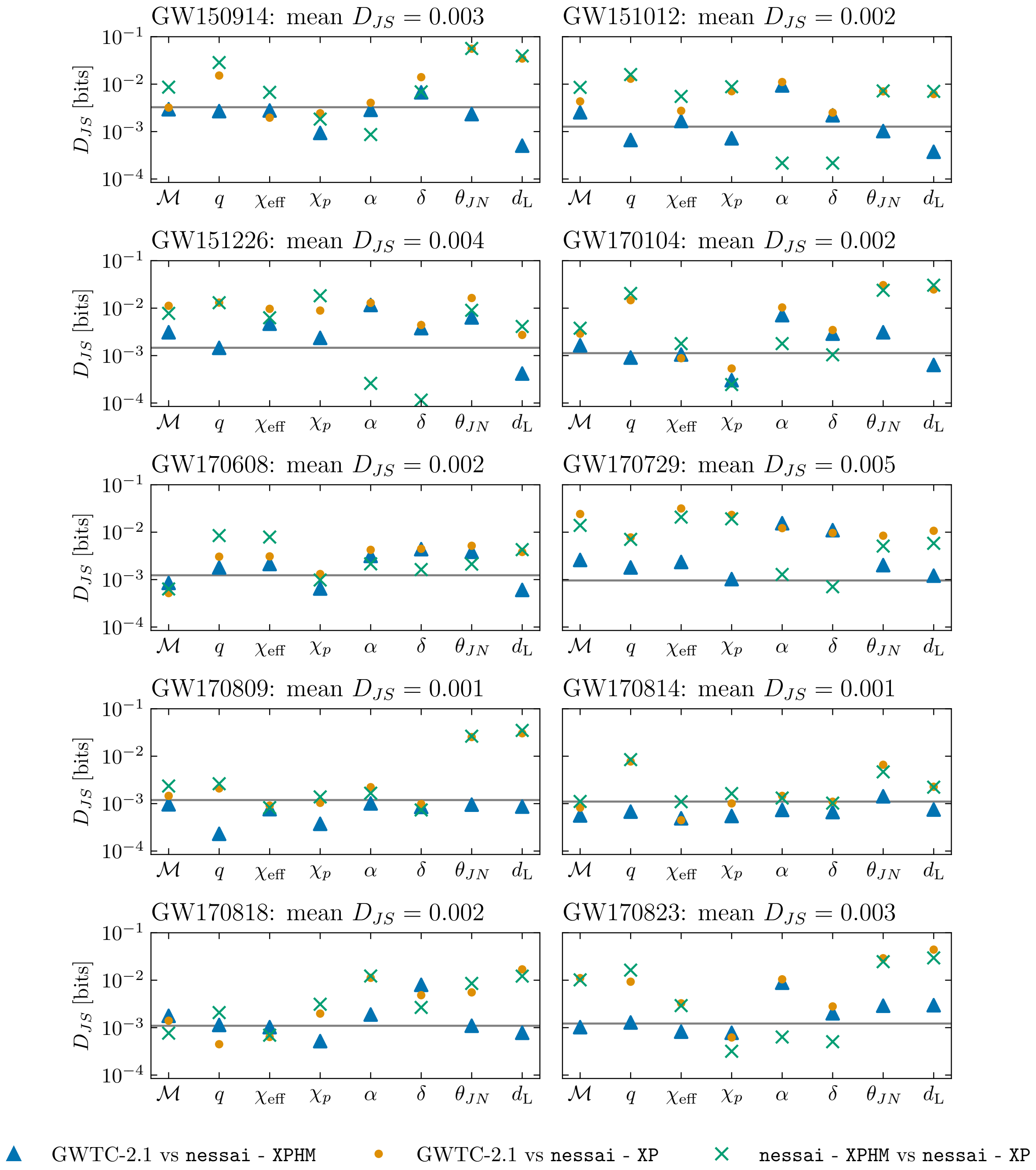


Figure 8.7: JSD in bits between GWTC-1 posterior samples and posterior samples produced by `nessai` with `IMRPhenomXPHM` and `IMRPhenomXP` for the 10 BBH events from O1 and O2. The mean JSD for each event is shown above each plot. The horizontal grey line shows the $10/n$ samples threshold, where n is the minimum number of samples between the three sets of posterior samples for each event.

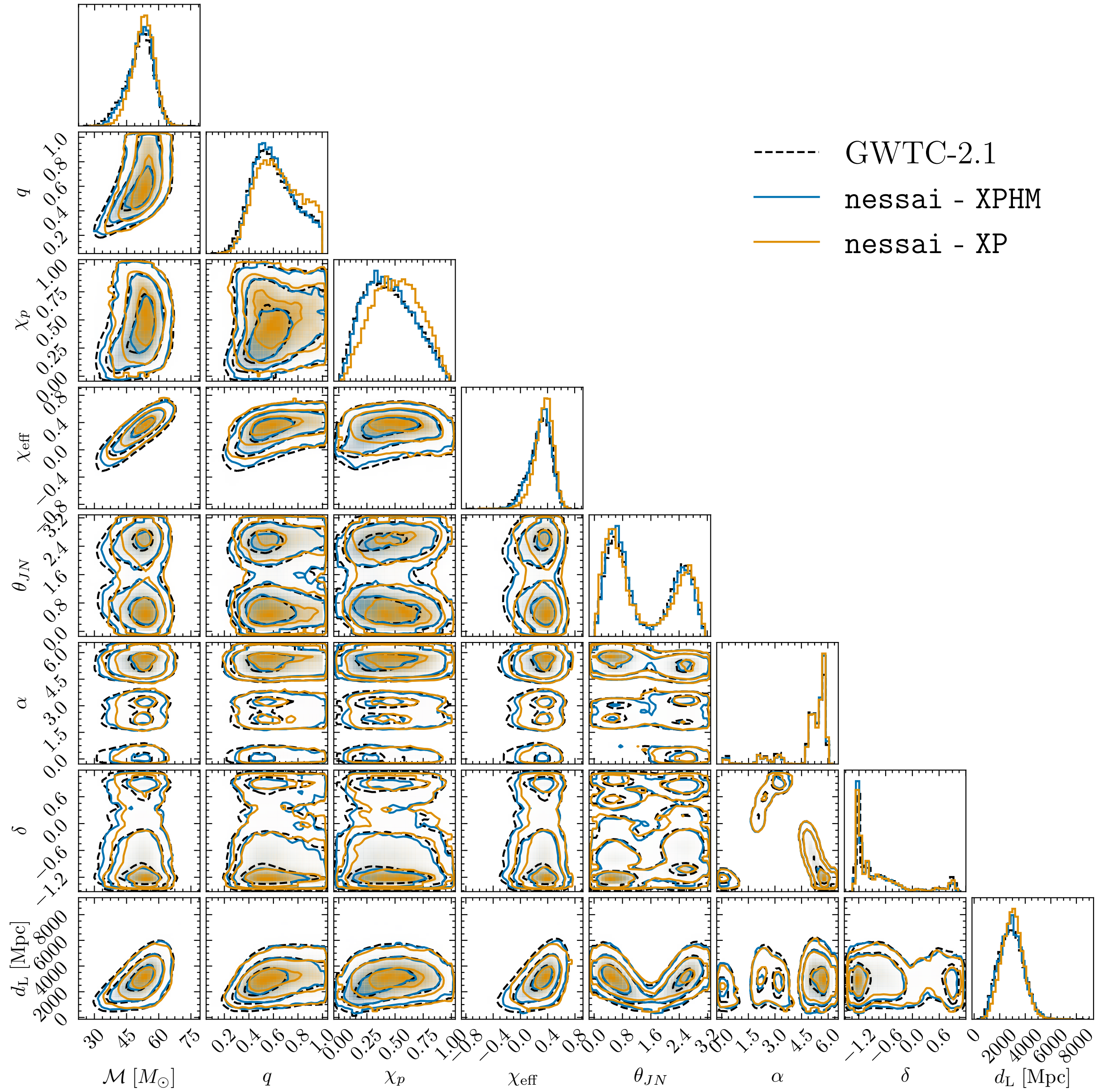


Figure 8.8: Corner plot showing the posterior distributions for GW170729 from GWTC-2.1 (black dashed line), and two analyses using `nessai`: one with `IMRPhenomXPHM` (blue solid line) and another with `IMRPhenomXP` (orange solid line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

GW170814 has the lowest average JSD of all events at 0.0007 bits. All but one of the parameters have JSD below 0.001 bits, the exception being the inclination angle θ_{JN} with $D_{\mathrm{JS}} = 0.0014$ bits. As expected, fig. 8.9 shows that posteriors between the analyses are consistent, however the lower number of samples for the `nessai` analyses is apparent. The result obtained with `IMRPhenomXP` and phase marginalization also shows good agreement with the results with `IMRPhenomXPHM`, suggesting there is little evidence for precession or higher-order modes in this signal.

Of all the parameters, right ascension and declination (α, δ) have the highest average JSDs at 0.0062 bits and 0.0042 bits respectively. Once again, χ_{p} has the lowest average JSD at just 0.0008 bits. We present the sky maps produced using the right ascension and declination posterior samples from each analysis in figs. 8.10 and 8.11, these also include the area in square degrees contained within the 50% and 90% contours for each analysis. The 50% contour for the `nessai` is smaller than the corresponding contour from GWTC-2.1 for eight of the events. Similarly, for the 90% contours, `nessai` produces smaller sky maps for seven of the events. Furthermore, GW170729 and GW17809, the posterior from `nessai` is missing small secondary modes that are present in the GWTC-2.1 results. If we assume the GWTC-2.1 results to be unbiased, this suggests that the `nessai` sky maps are over-constrained, though the low number of samples may also contribute to this.

In summary, the results produced using `nessai` with `IMRPhenomXPHM` are broadly consistent with the samples released alongside GWTC-2.1; the average JSD between the samples is 0.0025 bits. GW170729 has the highest JSD, but the posterior distributions presented in fig. 8.8 still show good visual agreement. Of all the parameters, the right ascension and declination show the largest disagreement and the sky maps in figs. 8.10 and 8.11 suggest that some of the results with `nessai` may be slightly over-constrained, though the lower number of samples may also contribute to this. When considering these results alongside those presented for GWTC-1, they suggest the differences are not due to `nessai`, but rather how the different analyses are configured.

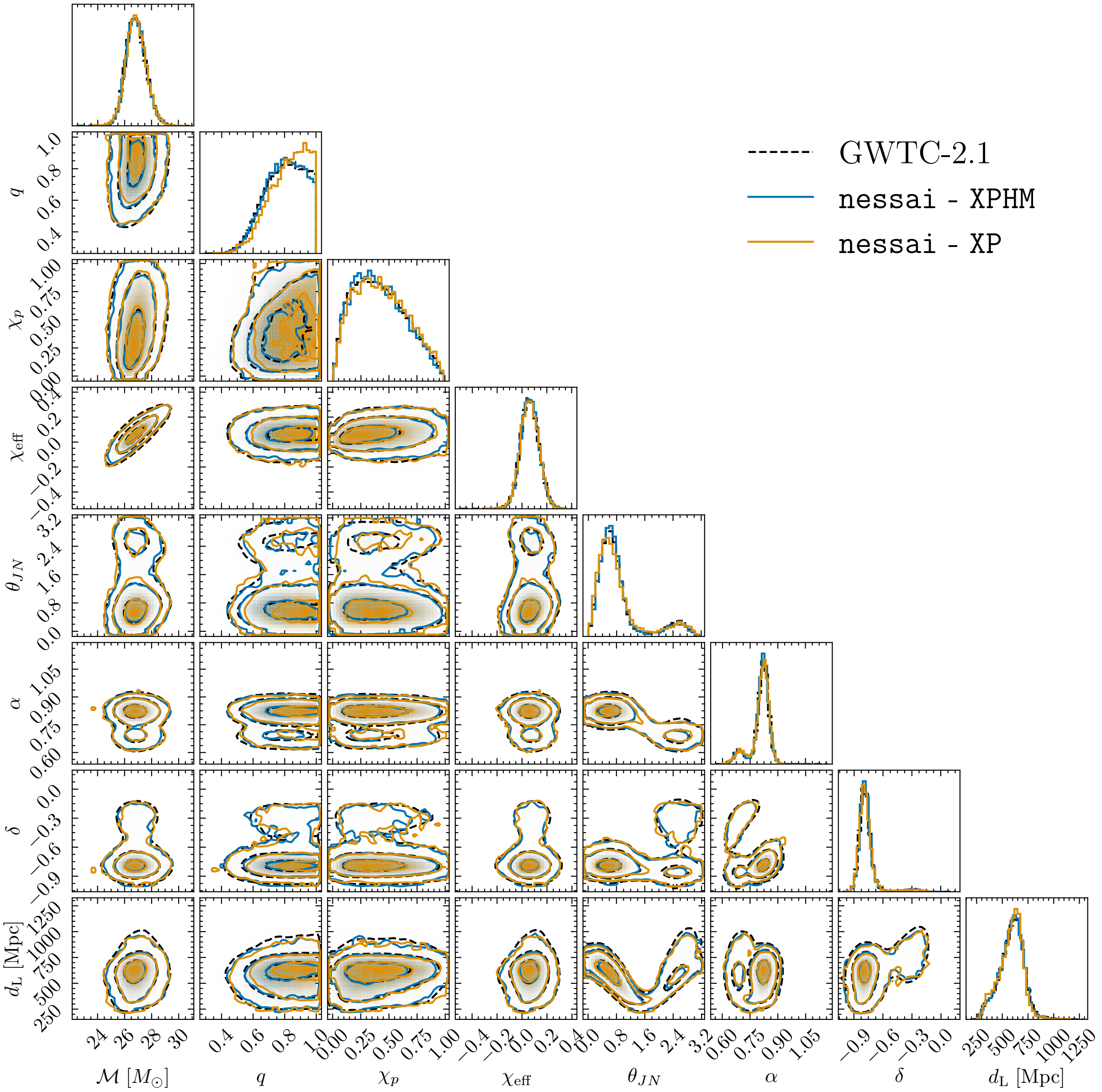


Figure 8.9: Corner plot showing the posterior distributions for GW170814 from GWTC-2.1 (black dashed line), and two analyses using `nessai`: one with `IMRPhenomXPHM` (blue solid line) and another with `IMRPhenomXP` (orange solid line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

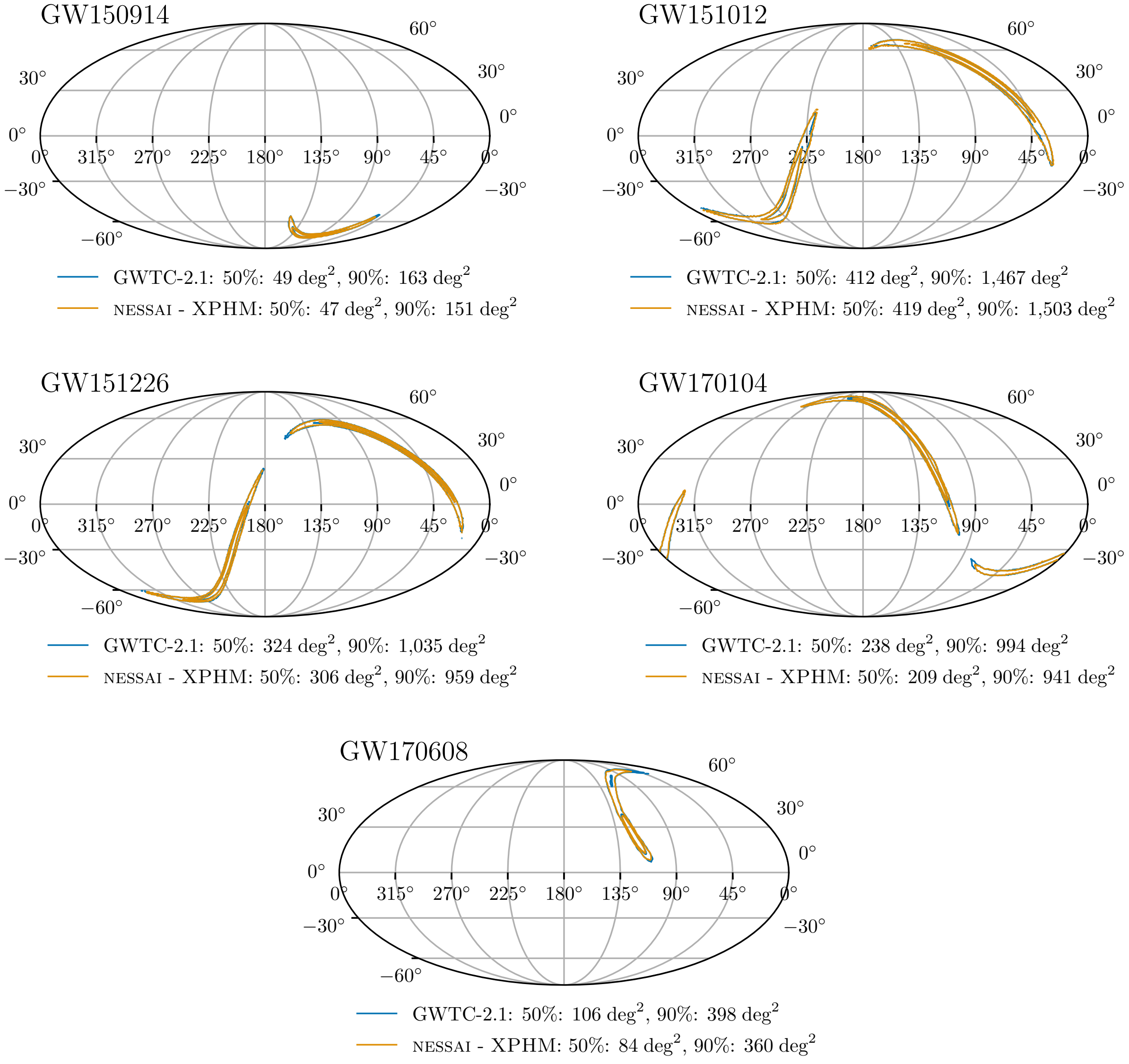

GW150914
GWTC-2.1: 50%: 49 deg^2, 90%: 163 deg^2
NESSAI - XPHM: 50%: 47 deg^2, 90%: 151 deg^2
GW151012
GWTC-2.1: 50%: 412 deg^2, 90%: 1,467 deg^2
NESSAI - XPHM: 50%: 419 deg^2, 90%: 1,503 deg^2
GW151226
GWTC-2.1: 50%: 324 deg^2, 90%: 1,035 deg^2
NESSAI - XPHM: 50%: 306 deg^2, 90%: 959 deg^2
GW170104
GWTC-2.1: 50%: 238 deg^2, 90%: 994 deg^2
NESSAI - XPHM: 50%: 209 deg^2, 90%: 941 deg^2
GW170608
GWTC-2.1: 50%: 106 deg^2, 90%: 398 deg^2
NESSAI - XPHM: 50%: 84 deg^2, 90%: 360 deg^2
60°
30°
0°
−30°
−60°
0°
315°
270°
225°
180°
135°
90°
45°
0°

Figure 8.10: Sky maps for the first half of the BBH events from O1 and O2. Sky maps are shown for the analysis from GWTC-2.1 and for the analysis using `nessai` with `IMRPhenomXPHM`. The 50% and 90% contours are shown for both analyses, and the area of each contour in square degrees is also quoted.

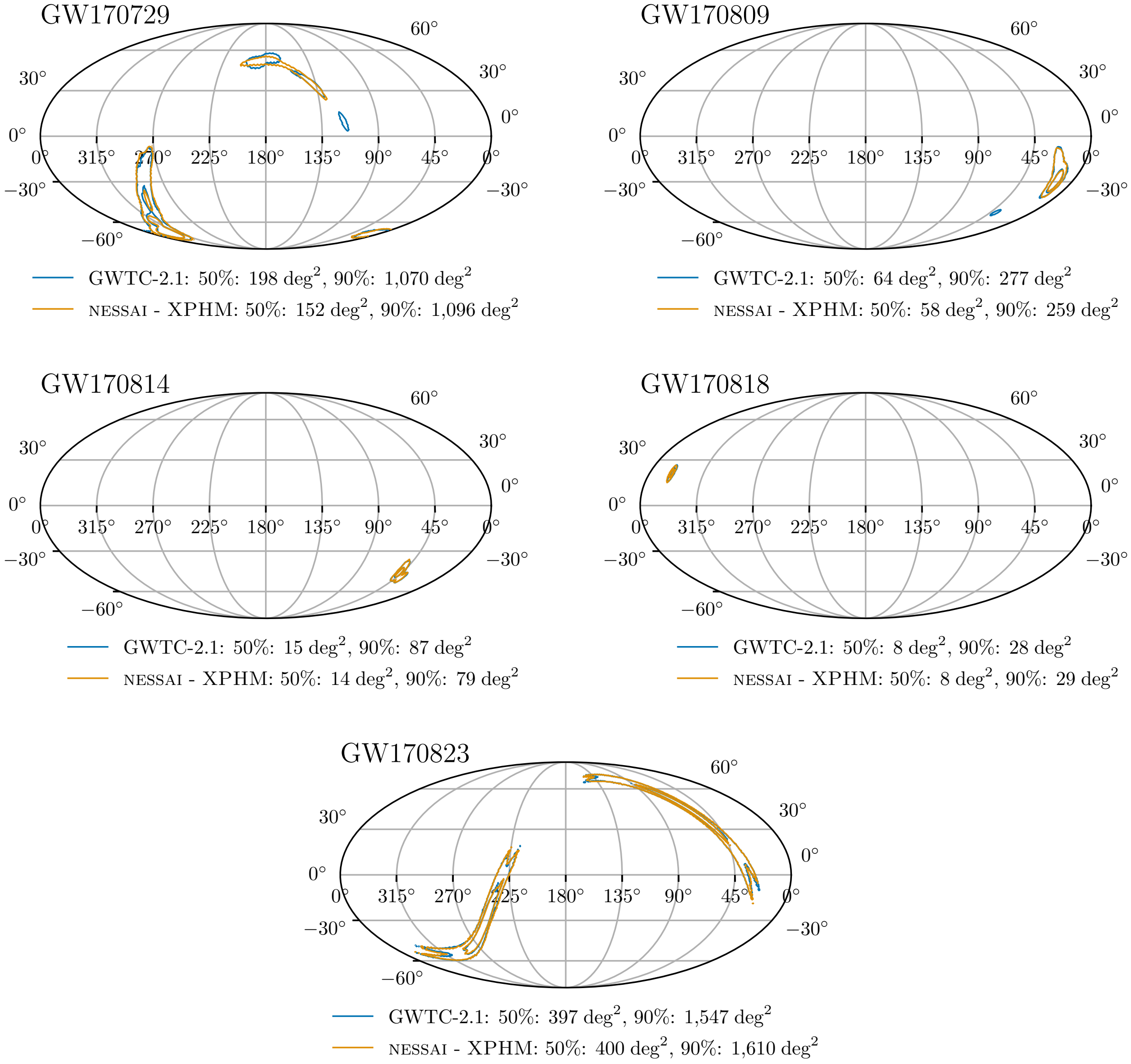

GW170729
60°
30°
0°
−30°
−60°
0° 315° 270° 225° 180° 135° 90° 45° 0°
GWTC-2.1: 50%: 198 deg^2, 90%: 1,070 deg^2
NESSAI - XPHM: 50%: 152 deg^2, 90%: 1,096 deg^2
GW170809
60°
30°
0°
−30°
−60°
0° 315° 270° 225° 180° 135° 90° 45° 0°
GWTC-2.1: 50%: 64 deg^2, 90%: 277 deg^2
NESSAI - XPHM: 50%: 58 deg^2, 90%: 259 deg^2
GW170814
60°
30°
0°
−30°
−60°
0° 315° 270° 225° 180° 135° 90° 45° 0°
GWTC-2.1: 50%: 15 deg^2, 90%: 87 deg^2
NESSAI - XPHM: 50%: 14 deg^2, 90%: 79 deg^2
GW170818
60°
30°
0°
−30°
−60°
0° 315° 270° 225° 180° 135° 90° 45° 0°
GWTC-2.1: 50%: 8 deg^2, 90%: 28 deg^2
NESSAI - XPHM: 50%: 8 deg^2, 90%: 29 deg^2
GW170823
60°
30°
0°
−30°
−60°
0° 315° 270° 225° 180° 135° 90° 45° 0°
GWTC-2.1: 50%: 397 deg^2, 90%: 1,547 deg^2
NESSAI - XPHM: 50%: 400 deg^2, 90%: 1,610 deg^2

Figure 8.11: Sky maps for the second half of the BBH events from O1 and O2. Sky maps are shown for the analysis from GWTC-2.1 and for the analysis using `nessai` with `IMRPhenomXPHM`. The 50% and 90% contours are shown for both analyses, and the area of each contour in square degrees is also quoted.

Run statistics

We now examine the run statistics for the BBH analyses described in the previous sections. Equivalent statistics are not included in the data releases that accompany GWTC-1 and GWTC-2.1 [467], so we cannot compare them to `nessai`. However, we can still compare the runs with `nessai` with different waveform approximants and determine which part of the algorithm is the dominant cost.

For each event, we track the number of likelihood evaluations and total wall time per parallel run. However, the wall time is highly dependent on the hardware allocated to each run, so it is a less reliable indication of sampling efficiency than the number of likelihood evaluations. In fig. 8.12, we present the wall times and total likelihood evaluations for each event. There is an approximately linear relationship between the number of likelihood evaluations and wall time, which suggests that the likelihood accounts for a significant portion of the total wall time. We revisit this later in this section. As expected, the analyses with phase marginalization require an order of magnitude fewer likelihood evaluations and there are similar differences in the wall time. We also observe that the number of likelihood evaluations is strongly correlated to duration of the signal and signal-to-noise ratio (SNR). GW151012 and GW170104 used 8 s segment lengths and GW151226 and GW170608 use 16 s segment lengths and these are amongst the most costly analyses, alongside GW150914 which is the loudest BBH event from O1 and O2 [7].

We compute the fraction of the total wall time spent evaluating the likelihood, training the normalizing flow and populating the pool of new samples and present the results per event in fig. 8.13. The fraction of the time spent evaluating the likelihood is higher in the runs with `IMRPhenomXPHM` and phase marginalization disabled, but a clear trend is visible across both sets of runs: analyses for events that have a longer duration spend a larger portion of the time evaluating the likelihood, since the cost of evaluating the likelihood scales with duration. This is particularly noticeable for the two 16-second events, GW151226 and GW170608, where more than 50% of the time was spent evaluating the likelihood and suggests that these analyses could be performed with more cores.

The run statistics for the analyses with `nessai` show expected correlation between signal duration and number of likelihood evaluations and wall time. As seen in chapter 6, disabling phase marginalization for the runs with `IMRPhenomXPHM` increases the cost of performing analyses by an order of magnitude. For shorter signals, this is dominated by the cost of populating the pool of new samples, which includes rejection sampling and highlights how `i-nessai` could be used to reduce the cost of performing analyses. However, for longer signals, even when using 16 cores, the cost of evaluating the likelihood still accounts for more than 50% of the wall time, suggesting that such analyses may be able to make use of more cores.

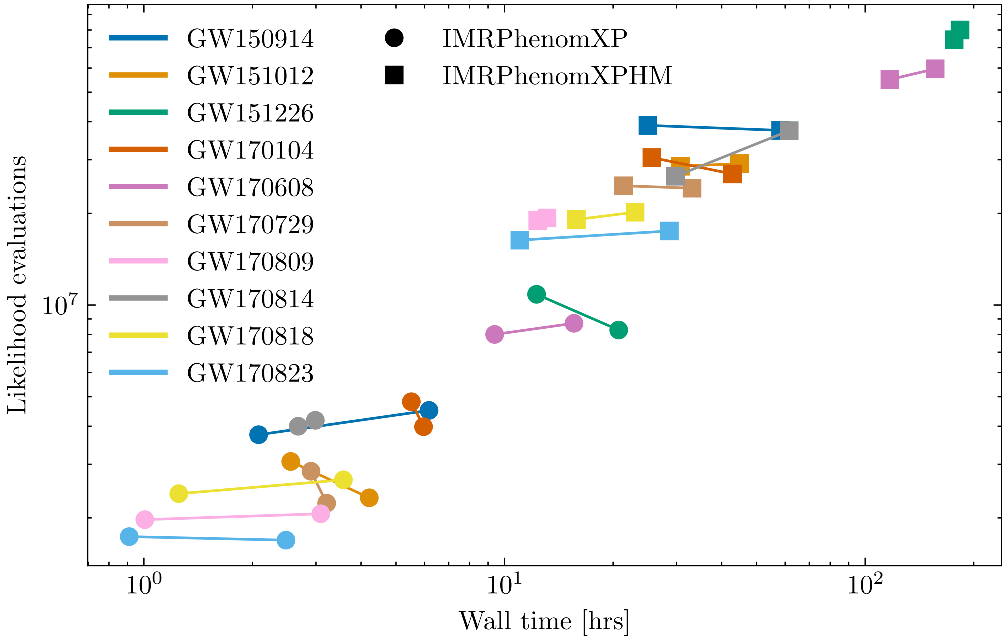


Figure 8.12: Wall time in hours versus the total number of likelihoods evaluations for each of BBH events from O1 and O2 when analysed using `nessai` with either `IMRPhenomXP` with phase marginalization (circles) or `IMRPhenomXPHM` without phase marginalization (squares). Two parallel runs are performed for each combination of event and waveform, and each pair is joined by a line.

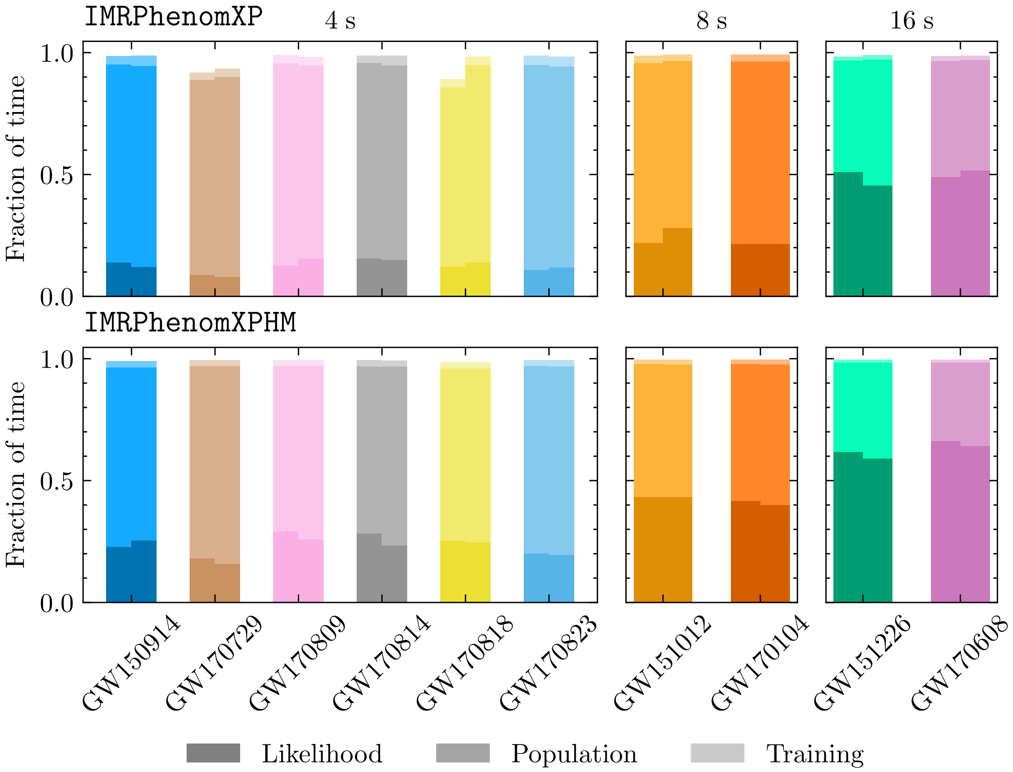


Figure 8.13: Fraction of the total wall time spent of evaluating the likelihood, populating the pool of new samples and training the normalizing flow for each BBH analysis with `nessai`. Events are grouped by the length of the data analysed, from left to right: 4 s, 8 s and 16 s. Results are shown for `IMRPhenomXP` with phase marginalization (top) and `IMRPhenomXPHM` without phase marginalization (bottom). Separate bar charts are shown for the two parallel runs per event.

8.2.2 BNS analysis

Unlike the BBH events from O1 and O2, the sole BNS event GW170817 [29] has not been reanalysed by the LVK and the most recent public samples are those from GWTC-1 [7, 466], it has however been reanalysed by other groups [122, 296]. We reanalyse the event using `nessai` and `bilby` with `IMRPhenomPv2_NRtidalv2` [456] and compare our results to those from GWTC-1, which used `LALInference` and `IMRPhenomPv2_NRTidal` [129].

In GWTC-1, two analyses were performed: one with low-spin priors ($a_i < 0.05$) and another with high-spin priors ($a_i < 0.89$) [29] and we use the same treatment here. For the tidal parameters, we sample in the dimensionless parameters $(\tilde{\Lambda}, \delta\tilde{\Lambda})$ [468, 469] instead of sampling (Λ_1, Λ_2) directly, since this is what was used in the GWTC-1 analysis [7]. We employ Reduced-Order-Quadratures (ROQs) to reduce the cost of evaluating the likelihood, however this requires that we analyse 256 s of data instead of the 128 s used for GWTC-1 and that we use a different prior range for the chirp mass: $[0.92, 1.7]\ \mathrm{M}_{\odot}$. Furthermore, the electromagnetic emission that followed the merger [38] allowed for its sky location to be precisely determined, meaning that one can also fix the sky location instead of sampling it, further reducing the computational cost. Due to the length of the signal being analyses, calibration reweighting has increased computational cost for BNS analyses, we therefore also perform the analyses twice: once sampling over the calibration parameters and a second time using calibration reweighting.

We compare the posterior samples produced by our analyses to those released with GWTC-1 in fig. 8.14. For the low-spin priors, the two `nessai` analyses are broadly consistent with each other and the with GWTC-1 results. The insertion indices for the analyses, see fig. 8.15, show clearly that the runs where the calibration parameters are sampled are over-constrained whilst the runs with calibration reweighting are not. This suggests that, despite the apparent agreement with GWTC-1, the settings used are only appropriate when used with calibration reweighting. The high-spin posteriors, unlike for the low-spin analyses, show clear differences to those from GWTC-1, most notably in the 2-dimensional posteriors for the masses and tidal parameters. This may be in part due to the priors used in our analysis which, due to the use of ROQs, have additional constraints on the masses compared to the GWTC-1 priors. Furthermore, the updated treatment for tidal parameters (`NRTidal` versus `NRTidalv2`) and use of phase marginalization, may also account for some of the differences. However, even when using calibration reweighting, the insertion indices show slight signs of over-constraining, suggesting that the high-spin posteriors are slightly over-constrained.

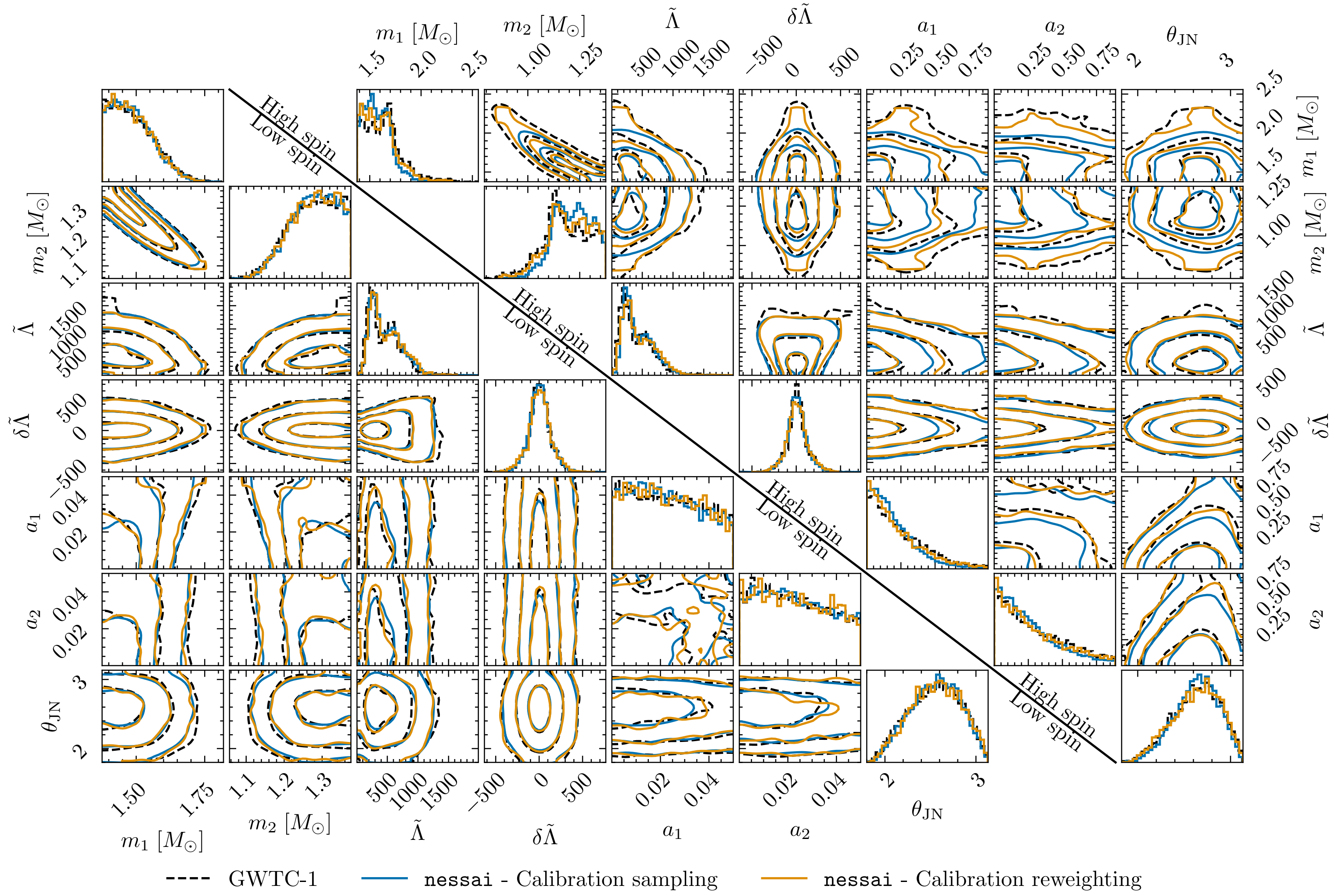
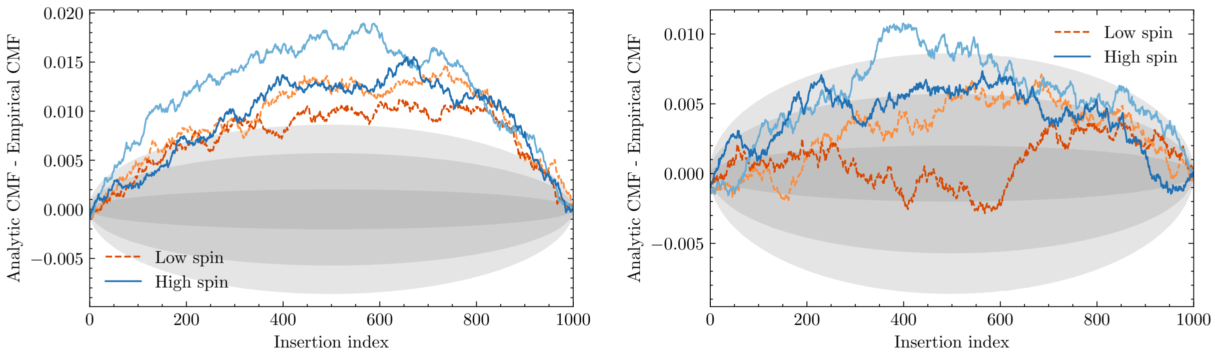


Figure 8.14: Subset of the posterior distributions for the low- and high-spin analyses of GW170817. Posteriors are shown for the samples from GWTC-1 (dashed black line), for the two analyses with `nessai`: one sampling the calibration errors (solid orange line) and the other using calibration reweighting (solid blue line). The 2-dimensional contours shown contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.



(a) Calibration sampling

(b) Calibration reweighting

Figure 8.15: Comparison of analytic cumulative mass function (CMF) and the empirical CMF for the insertion indices corresponding to the low- and high-spin analyses of GW170817 with `nessai`. The two parallel runs for each analysis are shown in different shades of orange (low-spin) and blue (high-spin). The shaded regions indicate the 1-, 2- and 3-σ confidence regions.

Whilst run statistics for the published results are not available, we can still compare the number of likelihood evaluations and wall times between different analyses, which we present in table 8.2. As observed in section 8.1.1, sampling the calibration parameters increases the wall time and number of likelihood evaluations, however for these long-duration signals, calibration reweighting is also more expensive, taking of order 5 minutes per run when spread across 16 cores. Furthermore, the analyses that sampled the calibration parameters are over-constrained, see fig. 8.15, and will have therefore required fewer likelihood evaluations than equivalent analyses that were not over-constrained. As such, the corresponding run statistics in table 8.2 under-represent the cost of obtaining unbiased results when sampling the calibration parameters with `nessai`.

These results show that the results from analysing GW170817 with `nessai` are broadly consistent with previous LVK analyses. The largest differences arise in the high-spin analyses, however, these differences are still smaller than the statistical differences we would expect from e.g. a different noise realization and well within the uncertainty of the previous analyses. This, alongside the diagnostics produced by `nessai`, implies that the settings used are suitable for low-spin analyses of BNS signals when the sky location is fixed and calibration reweighting is used. However, despite being broadly consistent with previous analyses, the high-spin results show signs of over-constraining, even when using calibration reweighting, suggesting that such analyses may require different settings. We emphasize though that the insertion index diagnostic test is more stringent than some other tests, e.g. P-P tests, and it has not been used in the LVK analyses, so we cannot be sure that previous LVK analyses would pass it[1]. We revisit this application in the

[1] The insertion index diagnostic test has been applied to some events from O1, O2 and O3a in Klinger and Agathos [470], however this has some limitations since the insertion indices were computed post-sampling rather than being recorded during sampling.

Table 8.2: Run statistics for the low- and high-spin analyses of GW170817 using nessai. The number of posterior samples is shown for the combined results, whereas the number of likelihood evaluations and wall time are shown for both parallel runs. Note that the number of likelihood evaluations does not include the evaluations required in the reweighting step, however, there will only be $\mathcal{O}(10^4)$.

Analysis	Calibration method	Posterior samples	Likelihood evaluations	Wall time [min]	Reweighting time [min]
Low-spin	Sample	9176	9.1×10^5 12.0×10^5	82 110	-
	Reweight	3474	6.2×10^5 5.6×10^5	17 65	4.7 5.0
High-spin	Sample	9319	11.1×10^5 12.1×10^5	1158 2176	-
	Reweight	4209	8.2×10^5 8.6×10^5	409 607	4.8 6.5

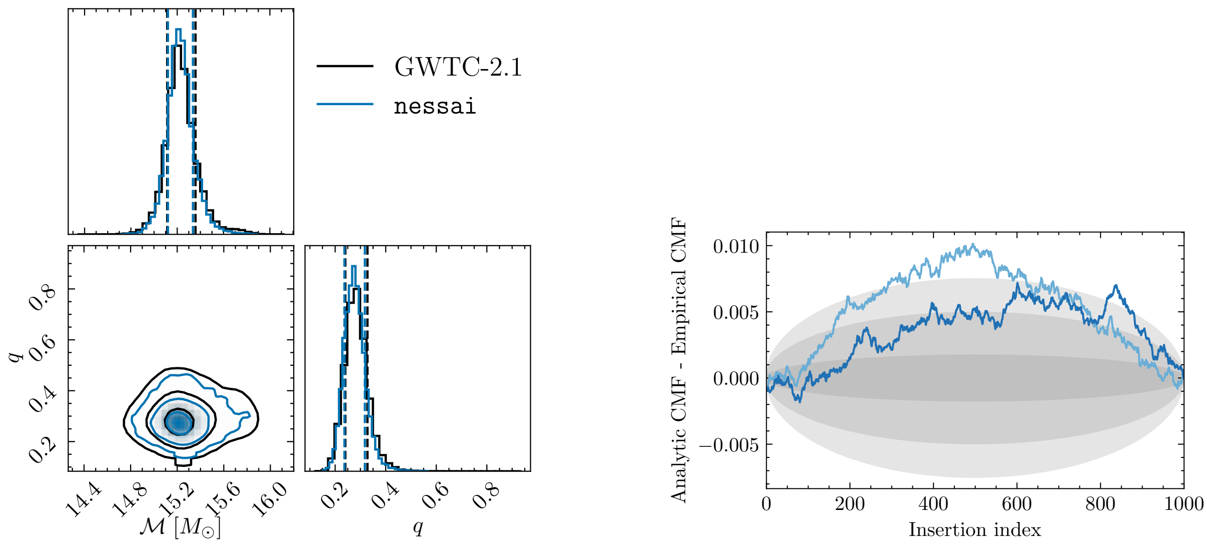
following section when we reanalyse another BNS event, GW190425.

8.3 Analysis of selected O3 events

We now analyse a subset of the events detected in O3 and compare the results obtained using `nessai` with `IMRPhenomXPHM` to the samples released alongside GWTC-2.1 [9, 467] and [10, 471]. We choose events that are particularly challenging to sample with the aim of identifying deficiencies during sampling when using the settings determined in chapter 6. All non-sampler-specific settings were set based on the settings used in GWTC-2.1 and GWTC-3.

8.3.1 GW190412

GW190412 was the first confident detection of BBH with asymmetric masses, and also showed significant contribution from higher-order modes in the waveform [472]. We repeat the analysis with `nessai` using the same settings as the analyses in section 8.2 with two parallel runs and present the results in fig. 8.16. The posterior distributions produced by our analyses are broadly consistent with those previously published by the LVK; the mass ratio and chirp mass are estimated to be $q = 0.276^{+0.76}_{-0.61}$ and $\mathcal{M} = 15.22^{+0.21}_{-0.18}$, however, the posteriors are narrower, suggesting the two parallel runs may be over-constrained. In fig. 8.16b, we compare the distribution of insertion indices to the expected distribution and find that both runs show signs of over-constraining. This suggests that these settings may not be suitable for more challenging to analyse events, however it once again highlights the value of checking the insertion indices.



(a) Posterior distributions

(b) Insertion indices

Figure 8.16: Results for the reanalysis of GW190412 with `nessai`. (a) Comparison of the posterior distribution for chirp mass and mass ratio obtained using `nessai` (blue) and the GWTC-2.1 analysis (black). The 2-dimensional contours shown contain 39.3%, 86.5% and 99.8% of the posterior probability respectively. (b) Comparison of the empirical distribution of the insertion indices for the two parallel runs to the analytic distribution. The shaded regions indicate the 1-, 2- and 3-σ confidence regions.

8.3.2 GW190425

GW190425 was detected in O3a and has component masses that are consistent with being a BNS, however the source-frame total mass is larger than the known population of BNS systems [262]. Unlike the first BNS detection, GW170817, this was a two-detector event and the higher mass results in a shorter signal. We reanalyse the GW190425 with `nessai` using `IMRPhenomPv2_NRtidalv2` [456] with ROQ bases and compare the results to the samples released with the discovery paper [262] and the reanalysis in GWTC-2.1. Similarly to our analysis of GW170817, we perform both a low spin ($a_i < 0.05$) and a high spin ($a_i < 0.89$) analysis and perform runs sampling the calibration parameters and using calibration reweighting. However, unlike for GW170817, the analyses in discovery paper and GWTC-2.1 sampled in Λ_1 and Λ_2 rather than $\tilde{\Lambda}$ and $\delta\tilde{\Lambda}$ [262]; we do the same in our analysis.

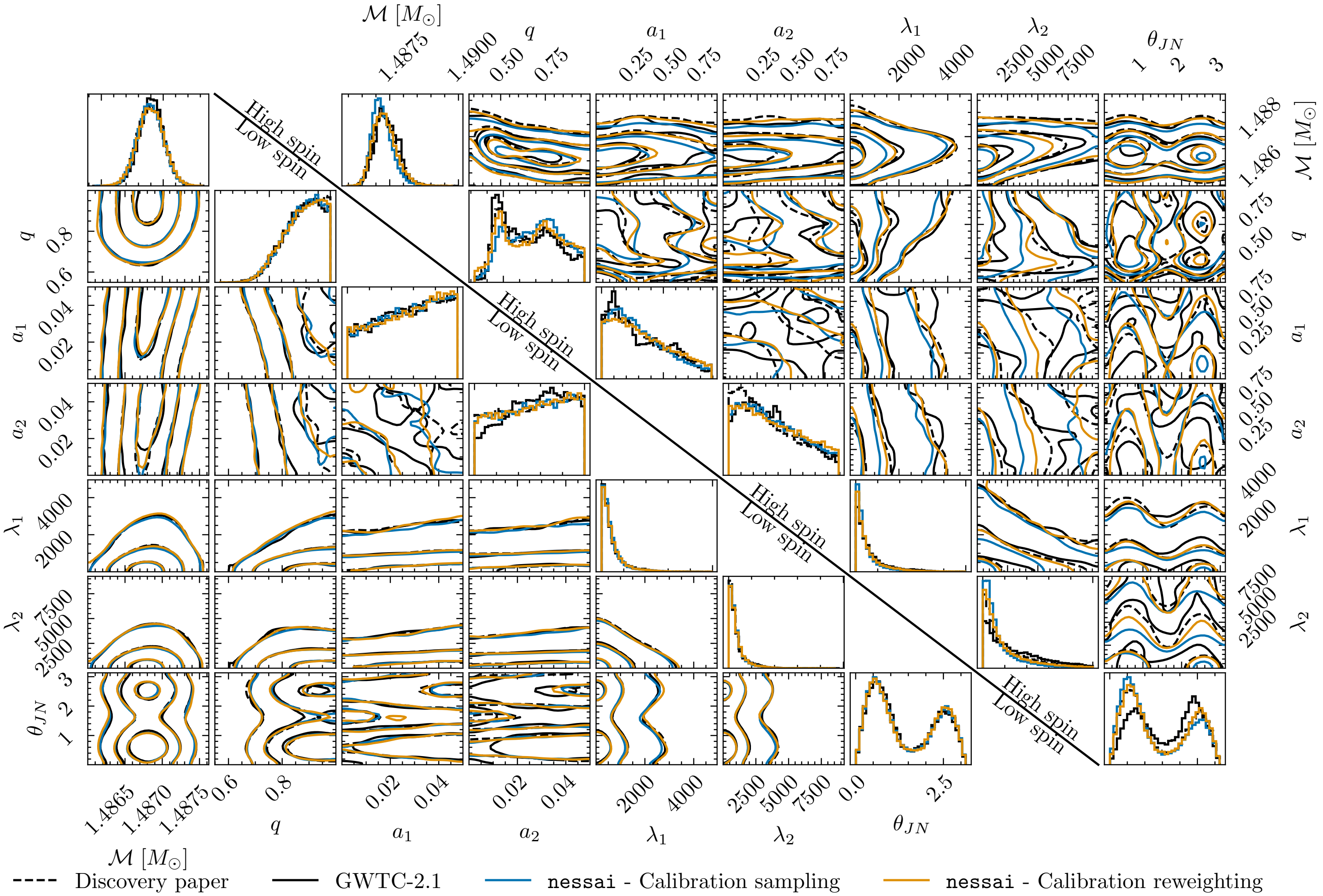
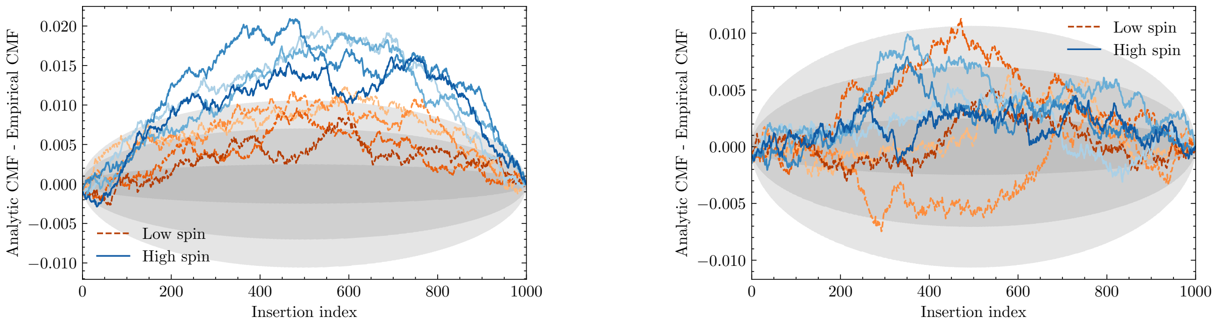


Figure 8.17: Subset of the posterior distributions for the low- and high-spin analyses of GW190425. Posteriors are shown for the samples released with the discovery paper (black dashed line), for the samples from GWTC-2.1 (solid black line), for the two analyses with `nessai`: one sampling the calibration errors (solid orange line) and the other using calibration reweighting (solid blue line). The 2-dimensional contours shown contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.



(a) Calibration sampling

(b) Calibration reweighting

Figure 8.18: Comparison of analytic CMF and the empirical CMF for the insertion indices corresponding to the low- and high-spin analyses of GW190425 with `nessai`. The four parallel runs for each analysis are shown in different shades of orange (low-spin) and blue (high-spin). The shaded regions indicate the 1-, 2- and 3-σ confidence regions.

In fig. 8.17, we compare the posterior distributions for the chirp mass, mass ratio, tidal deformabilities, spin magnitudes and inclination angle θ_{JN} for both the low and high spin analyses. We see mostly good agreement between all three low-spin results, including the analysis that sampled the calibration parameters, the exception is the spin magnitude of secondary mass a_2, where `nessai` is consistent with the discovery paper but the GWTC-2.1 result has less support for low spin magnitudes. However, in the high-spin case we see significant differences between the results from `nessai` compared to the GWTC-2.1 results, instead, the `nessai` results show better agreement with the discovery paper results, for example in the inclination angle θ_{JN}. The two `nessai` runs are also inconsistent, suggesting one or both may be over- or under-constrained.

We examine the insertion indices for these runs, see fig. 8.18, and find that the indices for the analyses where the calibration parameters are sampled show signs of over-constraining, particularly for the high-spin analyses. The runs that use calibration reweighting however do not show clear signs of over- or under-constraining, so the differences observed in fig. 8.17 cannot unequivocally be attributed to sampling biases, suggesting over differences are at play. This contrasts with our analyses of GW170817, where the high-spin analyses showed signs of over-constraining even using calibration reweighting

The results alongside those for GW170817 provide further evidence that `nessai`, with the correct settings, can produce reliable results for low-spin BNS analyses. Moreover, unlike GW170817, these analyses did not fix the sky locations, showing that `nessai` is not limited to BNS analyses where the sky location is known. High-spin analyses remain challenging due to the increased complexity of the parameter space.

8.3.3 GW190521

GW190521 is the most massive BBH detected to date, with a source frame total mass of $\sim 150\ \mathrm{M}_{\odot}$ [147]. The LVK analysis also found that posterior obtained with IMRPhenomXPHM has support for a secondary lower and unequal mass mode, fig. 8.19 shows the IMRPhenomXPHM samples release with GWTC-2.1 [467]. The posterior distribution produced by our reanalysis with nessai and IMRPhenomXPHM is shown alongside the result from GWTC-2.1 in fig. 8.19 and only shows support for higher, equal mass mode. The analysis with nessai does not recover the second mode that is present in the original analysis, which may be a result of using $N_{\mathrm{live}} = 1000$ for the analysis or over-constraining of likelihood contours during sampling. We therefore perform another analysis with $N_{\mathrm{live}} = 2000$, which is also included in fig. 8.19, and examine the insertion indices for both runs, see fig. 8.20. The posterior distribution obtained after increasing N_{live} is still inconsistent with the GWTC-2.1 results – it shows better agreement in the inclination but the more extreme mass ratio mode is still not present. The insertion indices in fig. 8.20 do not show clear signs of over-constraining, though overall the $N_{\mathrm{live}} = 1000$ do appear slightly worse. Given that this inconsistency cannot be clearly attributed to biases in sampling, this suggests there is another difference in either the configuration used or the pre-processing of the data.

8.3.4 GW191109

GW191109 is another event from O3 with a high-mass source (median total source frame of mass of $M = 112^{+20}_{-16}\ \mathrm{M}_{\odot}$ [10]) and it also has significant support for negative values of χ_{eff}. We reanalyse the data using nessai with IMRPhenomXPHM and compare the results to the IMRPhenomXPHM samples released with GWTC-2.1 [467]. In fig. 8.21, we compare the posterior distribution produced by our analysis with nessai to the GWTC-2.1 samples. The nessai result has more support for negative χ_{eff}, however, it is missing a secondary mode in the sky that is present in the GWTC-2.1 result and it lacks support for the higher-mass region. This suggests the posterior distribution may be over-constrained, so we compare the insertion indices for the two parallel analyses to the analytic distribution and present the result in fig. 8.22. The insertion indices, whilst not as clearly over-constrained as those for GW190412 or some of the GW190425 analyses, do show signs of slight over-constraining, which is most likely the cause of the differences seen in fig. 8.21.

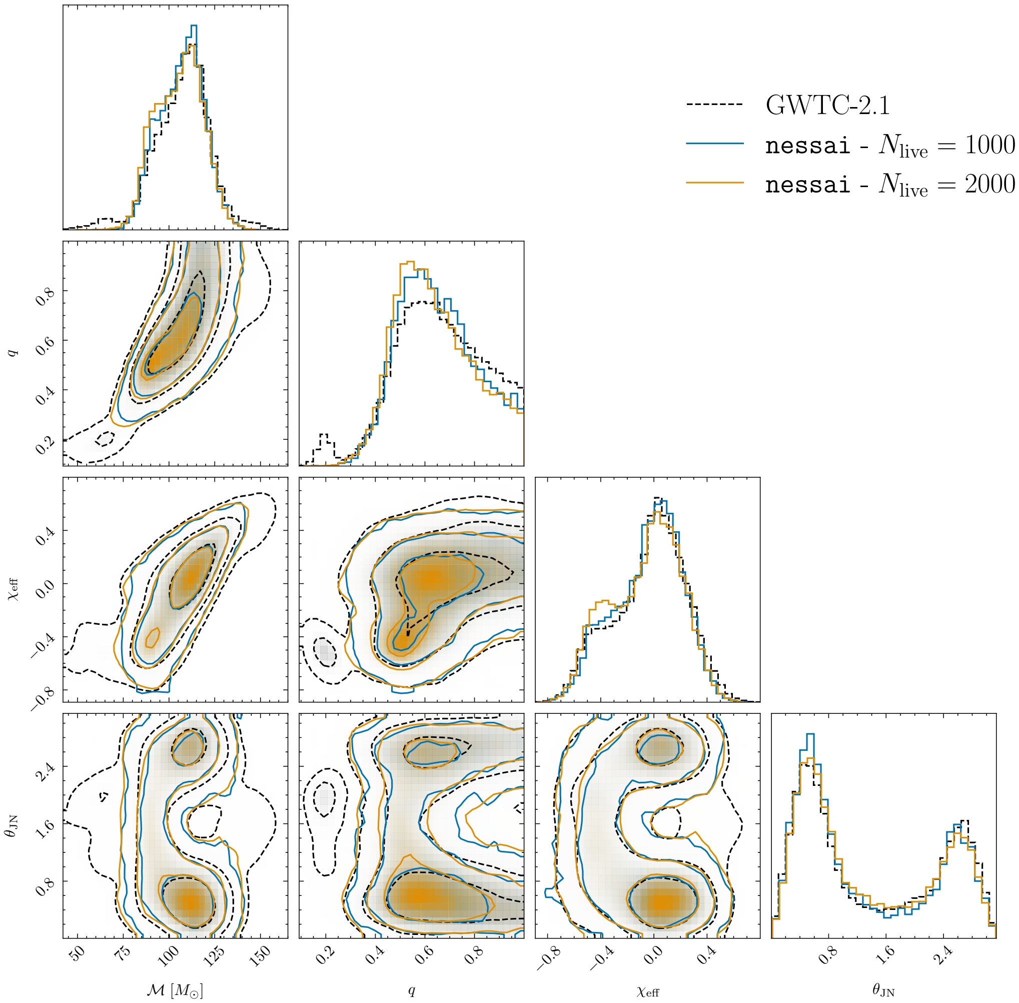


Figure 8.19: Comparison of the posterior distribution for GW190521 from GWTC-2.1 (black dashed line) and the posterior from our two reanalyses using `nessai`: using 1000 live points (blue solid line) and 2000 live points (orange solid line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

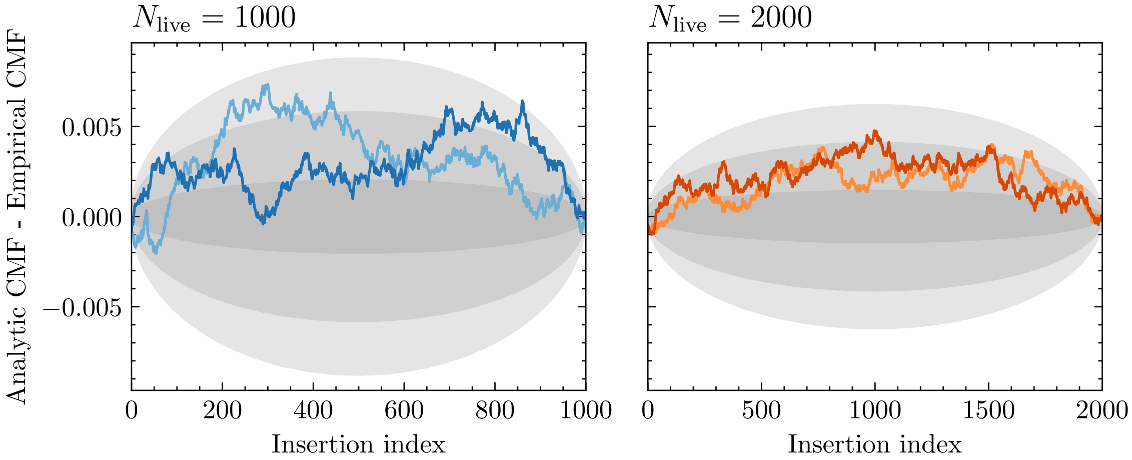


Figure 8.20: Comparison of the empirical distribution of the insertion indices to the analytic distribution for the two analyses of GW190521 with `nessai`. The indices for $N_{\text{live}} = 1000$ are shown on the left and those for $N_{\text{live}} = 2000$ on the right. The shaded regions indicate the 1-, 2- and 3-σ confidence regions and differ for the two analyses because they depend on the number of live points.

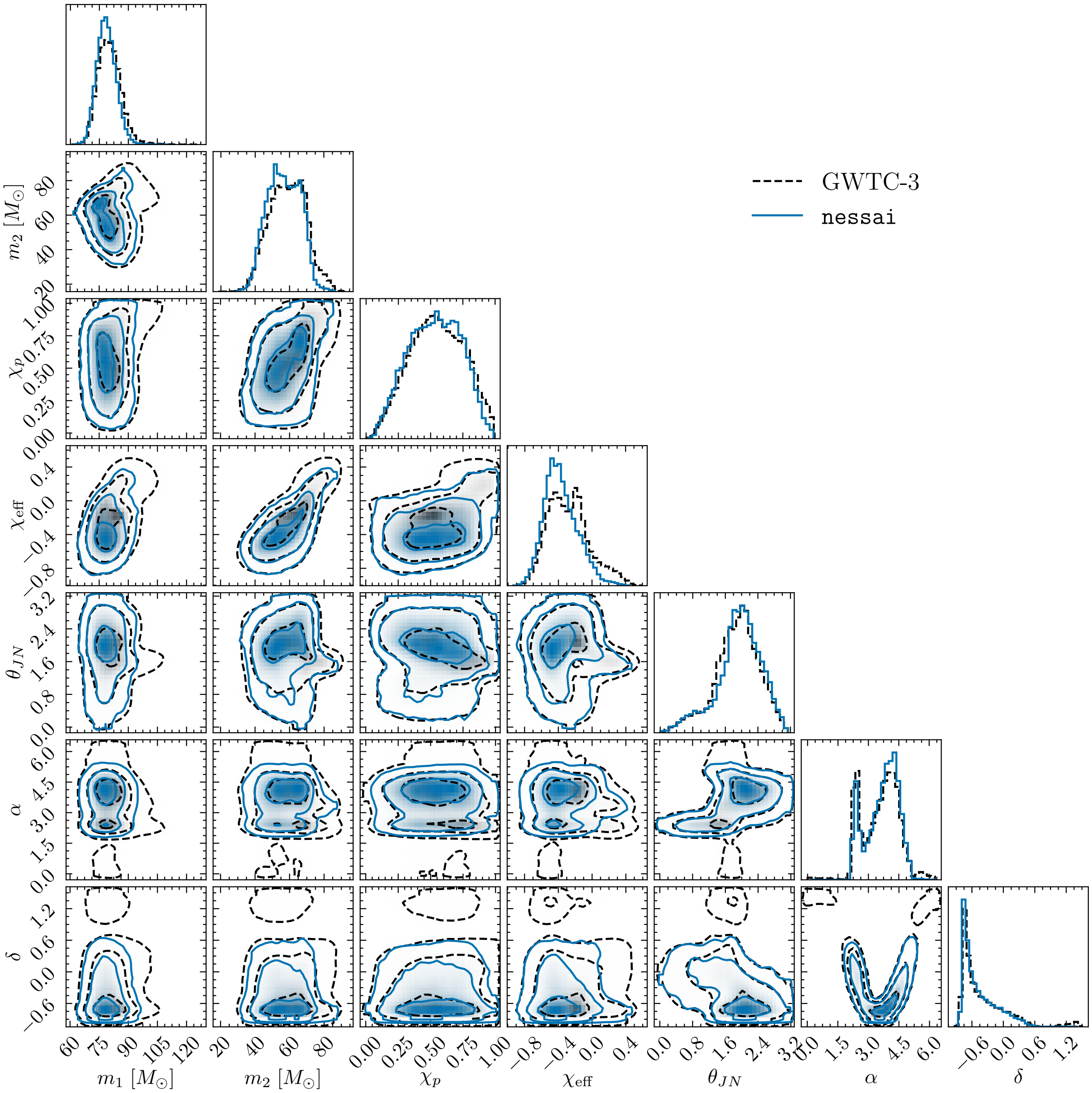


Figure 8.21: Comparison of a subset of the posterior distribution for GW190521 from GWTC-3 (black dashed line) and the posterior from our reanalysis using `nessai` (blue solid line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

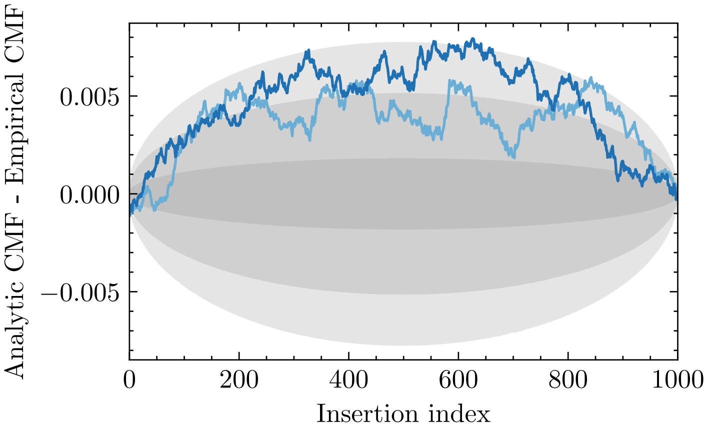


Figure 8.22: Comparison of the empirical distribution of the insertion indices for the two parallel runs for GW191109 to the analytic distribution. The shaded regions indicate the 1-, 2- and 3-σ confidence regions.

8.4 Analysis with i-nessai

The analysis of the BBH events from O1 and O2 highlighted once again how rejection sampling can be a bottleneck in compact binary coalescence (CBC) analyses with nessai, see fig. 8.13. We presented i-nessai in chapter 7 and demonstrated that it can address this bottleneck and reduce inference times. We now repeat the analyses described in section 8.2.1 for GW170823 using i-nessai and compare the results and run statistics to those presented previously.

We perform analyses with IMRPhenomPv2 with phase marginalization and we use two parallel runs per analysis which are combined to obtain a single set of posteriors samples per analysis. For both analyses we keep N_j constant and set it to $N_j = 10\,000$ and we set $\rho = 0.5$ with the entropy-based method for determining the next proposal, see chapter 7 for details. We also include consider similar analyses with IMRPhenomXPHM but find that, similar to what is discussed in chapter 6 for nessai, this would require changes to the sampler settings and the current settings do not reliably converge or produce a low number of effective samples. We show an example of this in appendix G.4.

In section 8.1.1, we showed that sampling the calibration errors significantly increases the cost of analyses with nessai, due to both increased number of likelihood evaluations and lower rejection sampling efficiency. Since i-nessai no longer requires rejection sampling, we also perform an analysis with i-nessai where the calibration parameters are sampled rather than using calibration reweighting and include this in our comparisons.

We compare the two analyses using i-nessai with IMRPhenomPv2 (calibration reweighting and sampling calibrtion parameters) to the analysis with nessai from section 8.2. Figure 8.23 shows the posterior distributions for all three analyses, we see that all three analyses produce consistent results, including the analysis that sampled the calibration parameters. This shows that i-nessai can produce consistent results when applied to real data and suggests that sampling the calibration parameters does not lead to the same biases that are seen in nessai, however more extensive testing is required. To quantify the differences, we compute the JSD divergences between the three runs and present the results in table 8.3. These divergences reinforce that the results with i-nessai are consistent with nessai and that sampling the calibration parameters does not necessarily bias the results.

Given the results with i-nessai show good agreement with those obtained with nessai, we compare the run statistics. Table 8.4 contains the average number of likelihood evaluations, average effective sample size (ESS) rounded to the nearest integer and average wall time in minutes for the three analyses. If we consider the two i-nessai analyses, these results show that, whilst the inclusion of calibration parameters may not bias the sampling, it still has a significant impact on the overall efficiency, be that in terms

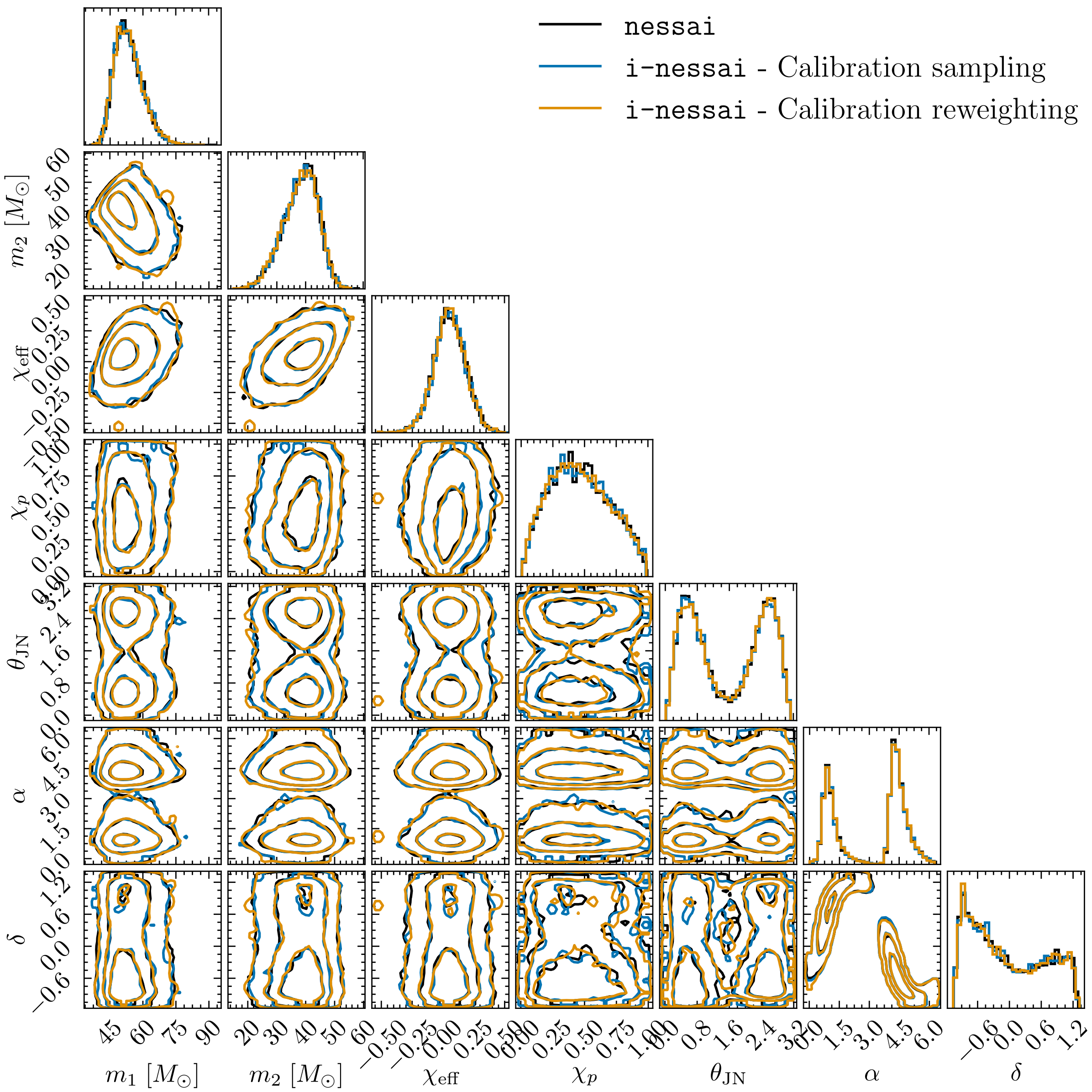


Figure 8.23: Comparison of the posterior distribution for GW170823 obtained `i-nessai` to results obtained previously with `nessai` in section 8.2 (black solid line). The 2-dimensional posteriors show the 39.3%, 86.5% and 99.8% contours.

Table 8.3: JSD in mbits between the posterior distributions for GW170823 from three analyses: the original analysis with nessai from section 8.2, an analysis using i-nessai with calibration reweighting and an analysis using i-nessai whilst sampling the calibration parameters. Since all three analyses use identical configurations, the divergences are shown for the 12 parameters that are sampled, rather than other derived parameters. These include the azimuth and zenith (ϵ, κ) instead of right ascension and declination and the time of coalescence measured at L1, t_{L1}.

	$\mathcal{M}$	q	a_1	a_2	θ_1	θ_2	ϕ_{12}	ϕ_{JL}	θ_{JN}	ψ	ϵ	κ	t_{L1}
nessai-i-nessai	0.83	0.39	0.16	0.26	0.41	0.38	0.22	0.42	0.10	0.43	0.30	0.76	0.44
nessai-i-nessai w. cal	0.63	0.30	0.38	0.56	0.67	0.56	0.12	0.28	0.12	0.36	0.18	0.86	1.00
i-nessai-i-nessai w. cal	1.42	0.42	0.47	0.25	0.33	0.23	0.20	0.37	0.21	0.11	0.44	0.75	0.48

Table 8.4: Run statistics for three analyses of GW170823: nessai with calibration reweighting, i-nessai with calibration reweighting and i-nessai with calibration sampling. The statistics are average over the two parallel runs performed for each analysis. The ESS has been rounded to the nearest integer.

	Likelihood evaluations	ESS	Wall time [min]
nessai (reweighting)	14.3×10^5	6759	91.4
i-nessai (reweighting)	7.7×10^5	15 305	17.5
i-nessai (sampling)	29.8×10^5	5319	236.5

of the number of likelihood evaluations to ESS or the wall time. This is no longer due to the inefficiency of rejection sampling in higher numbers of dimensions but instead due to poor convergence when training the flow, which we attribute to a combination of the low number of training samples and mixture of a few informative parameters with far more uninformative parameters (calibration parameters). On the other hand, if we compare the result using i-nessai with reweighting to nessai, we see that i-nessai produces more than double the number of effective samples in less than a fifth of the time. This speed-up comes predominantly from no longer requiring rejection sampling, which for this event accounted for more than 70% of the wall time (see fig. 8.13), though the reduction in the total number of likelihood evaluations also contributes.

The results show that i-nessai can be applied to real data and still provides the speed-ups seen for simulated data. They highlight that i-nessai has different failure modes to standard nessai; instead of producing over-constrained results when the calibration parameters are sampled, it yields fewer effective posterior samples and takes longer to converge. However, the recommendation is the same as for standard nessai: calibration reweighting is the preferred method for handling calibration uncertainty. Whilst this shows the potential of i-nessai, we find that the settings used here do not scale to analyses with higher-order modes, i.e. without phase marginalization, and further investigation is needed.

8.5 Looking forward

The next generation of ground-based interferometers will be significantly more sensitive than current generation detectors [473]. This will enable us to probe the early Universe and increases the expected event rate by orders of magnitude, but also brings new challenges [81]. For example, signals from the population of BBH systems detected by the current generation ground-based interferometers will have order-of-magnitude higher SNRs and be in the sensitive band of the detectors for 10 s of seconds [81]. This will significantly increase the cost of performing parameter estimation, because of the increased cost of computing the likelihood and the increased complexity of the likelihood surface due to e.g. higher-order modes [474], and require improved waveform approximants [475]. In this section, we demonstrate some of these challenges by applying `nessai` to simulated data for a network of next-generation ground-based detectors and highlight the need for further investigation into performing parameter estimation with data from these detectors.

We consider two network configurations: a network where Einstein Telescope (ET) is the only interferometer and a network with ET and Cosmic Explorer (CE). The former is unlikely but highlights some of the challenges posed by the triangular nature of ET. For ET we use the ET-D PSD described in Hild et al. [476] and place the detector at the current location of Advanced Virgo (AdVirgo) and for CE we use the *pessimistic* PSD from Abbott et al. [473] and place it at the current location of Laser Interferometer Gravitational-wave Observatory (LIGO) Hanford[2]. We simulate coloured Gaussian noise based on the PSDs and inject a GW150914-like signal at a distance of 4000 Mpc using `IMRPhenomXP` [118, 120]. Based on a minimum frequency of 5 Hz for ET, we analyse 40 s of data sampled at 40 Hz, the injected signal has a combined SNR of 104 in ET and 230 in CE.

We enable phase and distance marginalization but disable time marginalization since it is known to break down at high SNRs [251]. Given the SNR of the signal, phase marginalization is no longer valid since the effect of precession will no longer be negligible [477], however, given the challenges associated with sampling phase with `nessai` (chapter 6) and, as we will see, the complexity of the problem, we ignore this. We run our analysis using `bilby` [198] and use the settings for `nessai` described in chapter 5, including the use of constant volume mode, we also parallelize the likelihood evaluation over eight cores.

[2] These locations are based on what is currently implemented in `bilby`.

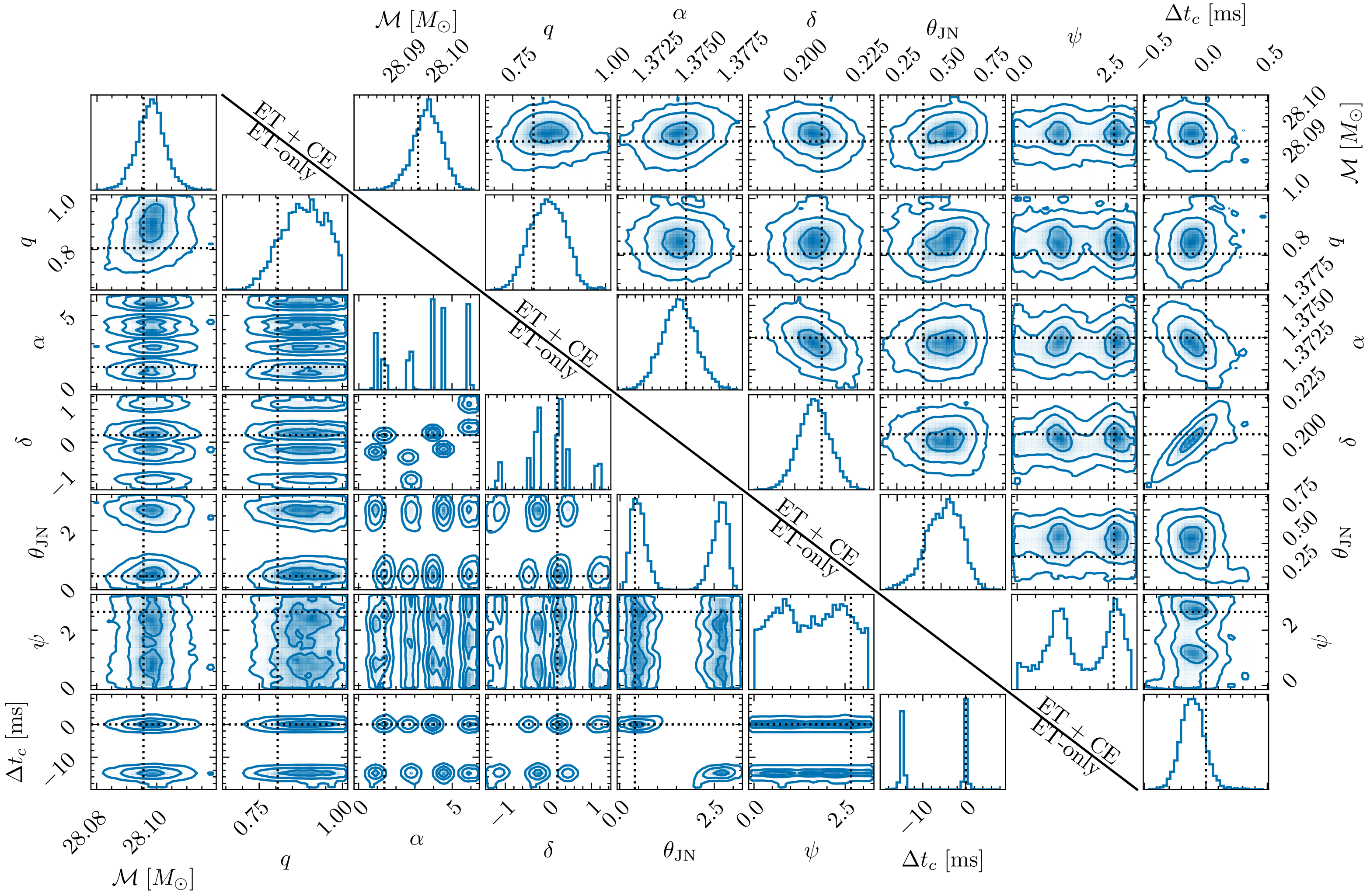


Figure 8.24: Subset of the posterior distributions for two analyses of a GW150914-like injection: the lower corner plot shows the results for an ET-only network and the upper corner plots shows the result for an ET plus CE network. $\Delta t_{\rm c}$ denotes the difference between the recovered time at the geocentre and the time of the injection. The 2-dimensional contours shown contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

Figures 8.24 and 8.25 show the posterior distributions and sky maps for the two network configurations. In the ET-only case, the sky angles (α, δ) are constrained to eight discrete modes on the sky, one of which contains the injected location. We also see two modes in the inclination angle that differ by $\pi/2$ and correspond to two different times; one at the injected time and another $\sim 15\,\mathrm{ms}$ before, and half of modes on the sky correspond to each. These two modes in inclination are mirror images of each other, reflected about the plane of the detector; this is visible in the sky map shown in fig. 8.25. Overall, the injected parameters have been recovered, however, the multimodal nature of many of the parameters makes sampling challenging. On the other hand, the inclusion of CE resolves the degeneracies seen in the ET-only case, and there is only a single sky-inclination-time mode.

The posterior distributions in fig. 8.24 hint at the difficulty of performing parameter estimation. When considering analyses with `nessai`, the biggest challenge is the highly multimodal parameter space, which will lead to inefficient rejection sampling. The ET-only run required $\mathcal{O}(2 \times 10^7)$ likelihood evaluations, compared to $\mathcal{O}(5 \times 10^6)$ for the two interferometer case, and it also shows signs of over-constraining which implies that more likelihood evaluations would be required for an unbiased run. Less specifically to `nessai`, the high SNR of the signals also poses a challenge; the ET plus CE analyses took ~ 23 hours, of which 21.3 hours were spent evaluating the likelihood. Compare this to the analyses of GW150914 in section 8.2 that took less than 10 hours, of which $\sim 15\%$ of the time was spent evaluating the likelihood, and increased computational cost of the longer-duration, higher-SNR is clear.

These results highlight some of the challenges associated with parameter estimation for next generation ground-based detectors. In particular, it highlights that the triangular nature of ET can result in a challenging to sample likelihood surface but also emphasizes the importance of having a second detector. We also show that `nessai` in its current state can be used to analyse such signals, but similar to what was observed in previous analyses, multimodality is a challenge and an improved treatment could reduce the cost of performing inference.

8.6 Conclusions

In this chapter, we have explored analysing real strain data from the current ground-based interferometers with `nessai` and `i-nessai`, focusing predominantly on the former since we have already determined the necessary settings in chapters 5 and 6. We start by considering how to handle calibration errors, and then reanalyse the events detected in O1 and O2 using a range of waveform approximants and compare our results to those from GWTC-1 and GWTC-2.1. As a further stress test, we also reanalyse a subset of

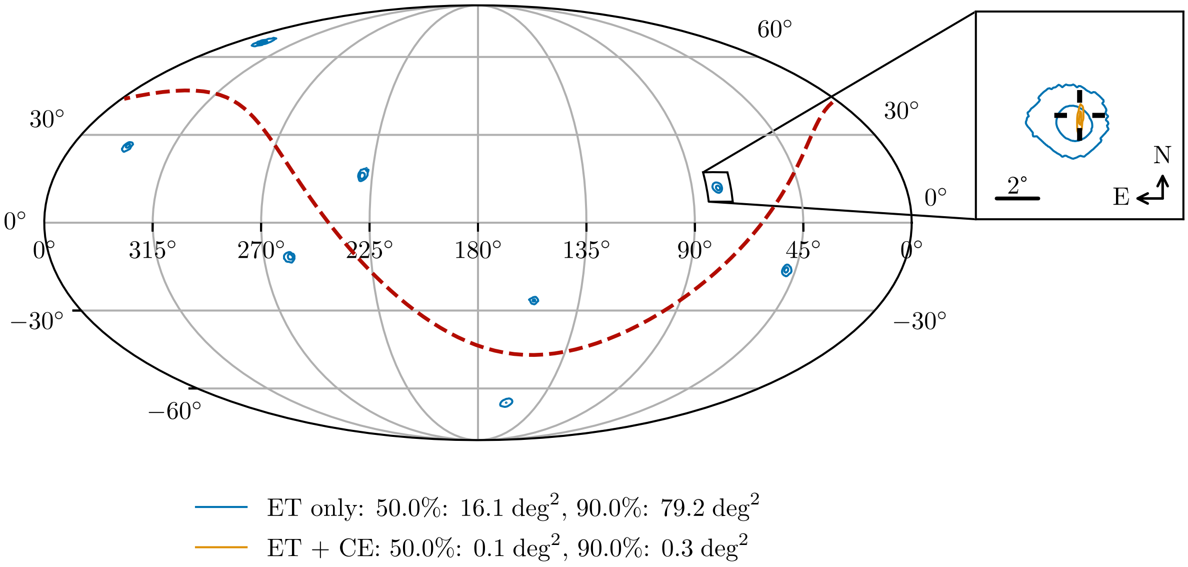

60°
30°
0°
−30°
−60°
0°
315°
270°
225°
180°
135°
90°
45°
0°
30°
0°
−30°
2°
N
E
ET only: 50.0%: 16.1 deg^2, 90.0%: 79.2 deg^2
ET + CE: 50.0%: 0.1 deg^2, 90.0%: 0.3 deg^2

Figure 8.25: Sky maps for two analyses of a GW150914-like injection in third-generation detector networks. Results for an ET-only network are shown in blue and those for an ET plus CE network in orange. The 50% and 90% contours are shown for both analyses, and the area of each contour in square degrees is also quoted. The cross-hair marks the right ascension and declination of the injected signal, and the red-dashed line shows the plane of the detector projected onto the sky for the ET-only case.

the events detected in O3 and compare our results to corresponding results from GWTC-2.1 and GWTC-3. Finally, we present two preliminary analyses; the first demonstrates how `i-nessai` can be used to analyse real data and the second highlights the challenges associated with applying `nessai` to next generation ground-based detectors.

The first hurdle to overcome when applying `nessai` to real data is handling the additional parameters introduced to model the imperfect nature of the detector calibration. This introduces upwards of 50 additional parameters that must either be sampled or otherwise accounted for. We find that sampling these parameters is inefficient with `nessai` due to how normalizing flows scale and can lead to over-constraining. Instead, we find success using calibrating reweighting as a post-processing step which avoids the need to sample additional parameters, but does result in fewer posterior samples. We test this further in subsequent analyses of real events from O1, O2 and O3.

We start by applying `nessai` to the BBH events from O1 and O2. These events have been analysed as a part of both GWTC-1 and GWTC-2.1, which allow us to compare runs with both `IMRPhenomPv2` and `IMRPhenomXPHM`. For the former we enable phase marginalization but disable it for the latter. When compared to results released with GWTC-1, we find that the results obtained using `nessai` with `IMRPhenomPv2` show systematic differences for some events, including GW150914. We investigate this further, considering the choice of priors and the use of phase marginalization, but conclude that these do not account for the differences, implying they must arise due to another discrepancy, for example changes between LALInference and bilby. When we reanalyse the same events using nessai and IMRPhenomXPHM, we see that nessai is broadly consistent with the results from GWTC-2.1. The sky angles show the largest difference, with the sky maps for the nessai analyses tending to be smaller than those for GWTC-2.1, suggesting some of the results with nessai may be slightly over-constrained. Since the settings were unchanged between the two analyses with nessai, the results with IMRPhenomXPHM suggest that the differences seen compared to GWTC-1 are not caused by problems during sampling.

We examine the run statistics from our analyses of the O1 and O2 BBHs and compare the analyses with and without phase marginalization, which correspond to the analyses with IMRPhenomPv2 and IMRPhenomXPHM respectively. These statistics are consistent with the results presented in chapter 6, and show that sampling the phase increases the number of likelihood evaluations and wall time by roughly an order of magnitude. We also observed a clear correlation between the length of the data being analyses and the fraction of the time spent computing the likelihood. This suggests that analyses with longer data lengths could make use of more cores compared to the four-second analyses performed in previous chapters.

GW170817 was the first detection of gravitational waves from BNS system and it was also the first multi-messenger gravitational-wave observation. We reanalyse the event using nessai and, following previous analyses, perform low- and high-spin analyses and compare our results to those released with GWTC-1 [7, 466]. Our analyses differ slightly from those in GWTC-1 since they use ROQs which require the use of different mass priors and analysing 256 seconds of data rather than 128 seconds. We perform analyses with calibration reweighting but, given the increased computational cost for long-duration signals, also perform runs where the calibration parameters are sampled. In both the low- and high-spin, analyses the insertion indices show sigs of over-constraining when sampling the calibration parameter whereas the analyses that used calibration reweighting only show slightly signs of over-constraining in one of the high-spin runs. The posterior distributions for the low-spin analyses are consistent with the GWTC-1 results. The high-spin analyses show differences in the posteriors for the masses and tidal parameters, these are likely due to of or a combination of the more up-to-date waveform model, the different priors and the use of phase marginalization. However, these differences are small and within uncertainty in the results from GWTC-1.

To identify failure modes with nessai, we reanalyse four events from O3 which we identify as being particularly challenging and compare the results to those published in GWTC-2.1 and GWTC-3. These events we choose to analyse are three BBH events: GW190412, GW190521 and GW191109 and the BNS event GW190425. GW190412 is a BBH with asymmetric masses that showed significant contribution from higher-order modes, our reanalysis produces results that are consistent with those from GWTC-2.1 but the insertion indices show signs of over-constraining. GW190521 is the most massive BBH detected to date, and the analysis from GWTC-2.1 found support for a secondary lower, unequal mass mode. The initial posterior distribution from our reanalysis does not have support for this secondary mode, so we repeat the analyses with double the number of live points but this second analysis also lacks support for the secondary more. Unlike for GW190412, the insertion indices for these runs do not show clear signs of over-constraining, suggesting there may be other differences at play. The final BBH event, GW191109, is also a high-mass source but also shows significant support for $\chi_{\mathrm{eff}} < 0$. The results from our reanalysis also show support for $\chi_{\mathrm{eff}} < 0$, however, a secondary mode in the sky location is absent from the posterior distribution and the insertion indices show slight signs of over-constraining.

GW190425 was the second confident detection that was consistent with being a BNS. We repeat the low- and high-spin analyses described in the discovery paper [262] and GWTC-2.1 [9] and compare our results to the samples released alongside each publication. We test both sampling the calibration parameters and calibration reweighting, but once again find that sampling them leads to over-constraining, so focus on the reweighted results. In the low-spin case, we observe good agreement between `nessai` and both sets of LVK results, except for a_2 where `nessai` is consistent with the discovery paper but disagrees with GWTC-2.1. In the high-spin case, the results produced by `nessai` are not consistent with either analyses; agreeing with the discovery paper for some parameters but with GWTC-2.1 for others. The insertion indices for the high-spin runs are not clearly over-constrained, so we cannot confidently attribute these differences to solely `nessai`. Instead, we attribute these differences to a combination of different priors, an updated waveform and different analysis framework (`LALInference` versus `bilby`).

We also perform limited tests of `i-nessai` applied to real data, specifically analysing GW170823 with `IMRPhenomPv2` and phase marginalization enabled. We find that unlike standard `nessai`, the inclusion of calibration parameters does not bias the results but does lead to a significant increase in the overall wall time and we therefore recommend using calibration reweighting for `i-nessai`. The results produced by `i-nessai` are consistent with `nessai` but, when using calibration reweighting, they are produced approximately five times faster whilst also yielding more than double the number of effective post samples. However, we find the settings used do not scale to analyses without phase marginalization, and this therefore requires further investigation.

Looking forward to next generation ground-based interferometers, there will be similar challenges when performing parameter estimation. As an example, we analyse a GW150914-like injection in an ET-only network and an ET plus CE network using `nessai`. We find that even when the analysis is simplified by including phase marginalization, it is significantly more costly than comparable analyses in the current generation of ground-based detectors. The ET-only case is particularly challenging since we observe a degeneracy in the sky angles and time of coalescence, which leads to multimodal posterior distributions. This highlights the need for further investigation into improved sampling techniques for next generation ground-based interferometers.

Overall, these results show that, when applied to real data, `nessai` can produce results that are consistent with published results or with differences that can be attributed to other differences in the analyses. Furthermore, in most cases, the observed differences are smaller than the statistical uncertainties from e.g. a different noise realization and are within the uncertainties of the published results. Additionally, these results demonstrate the value of the diagnostics included in `nessai`, particularly the insertion indices; they can be used to reliably identify biases during sampling and are more stringent than other commonly used tests, such as P-P tests. Therefore, in cases where they do not show signs of biases, they are an indicator of unbiased results that does not require repeating analyses with e.g. a different sampler. However, certain caveats still apply when using `nessai` to analyse real data:

- calibration errors should be handled via calibration reweighting rather than by sampling additional parameters since the latter increases the computational cost and can lead to biases in the results,
- sampling with phase marginalization remains an order of magnitude faster than without and should be used where possible,
- the insertion indices recorded by `nessai` should be checked for signs of over- or under-constraining,
- multimodality remains a challenge and can significantly impact sampling efficiency.

These caveats also highlight aspects that could be the subject of further investigation, for example: improved treatment of the phase parameter in `nessai` and `i-nessai` or novel methods for handling multimodality.

Chapter 9

Conclusions

The overarching aim of the work presented in this thesis is to investigate and develop means for accelerating gravitational-wave data analysis using machine learning. Our specific focus is on augmenting stochastic sampling techniques used to perform Bayesian Inference, thus bridging the gap between current techniques and the end-to-end machine learning techniques that are being developed. We have demonstrated that the methods proposed can accelerate gravitational-wave inference for compact binary coalescence (CBC) by up to and, in some cases, more than an order of magnitude. This improvement stems from the core idea of incorporating normalizing flows into nested sampling and the many iterative improvements that built upon it. We now review the main contributions and conclusions from chapters 4 to 8 before considering the broader impact of the work presented and overall implications.

nessai The first stepping stone to achieve this was the development of the algorithm nessai: nested sampling with artificial intelligence, (pronounced /'nes.i/) which is introduced in Williams et al. [1]. nessai is centred around the standard nested sampling algorithm first proposed by Skilling [193] that has since be built upon in various works [195]. Unlike previous implementations, it uses normalizing flows to learn the likelihood-constrained prior and sample directly from it without the need for Markov Chain Monte Carlo (MCMC). At a given iteration, a normalizing flow is trained to map the current distribution of live points to a Gaussian distribution in the latent space, a contour is then constructed in the latent space, samples are drawn from within the latent contour and then mapped back to the original space to obtain new samples within the likelihood contour. The flexibility of normalizing flows allows nessai to learn complex contours and sample from them efficiently, though this process requires the use of rejection sampling to ensure samples are distributed according to the priors.

Our first demonstration of nessai is for analysing simulated 4-second binary black hole (BBH) signals in a three-detector network. We analyse 128 signals with nessai;

we use phase marginalization but also repeat the analyses with and without distance marginalization. The resulting probability-probability plots (P-P plots) show that in the both cases nessai consistently recovers the posterior distributions for the injections.

We repeat the analyses using another nested sampler, dynesty, which has been used extensively for analysis of CBC signals [9, 10, 296] and use this a baseline to compare our results. We find that, on average, the number of likelihood evaluations required by nessai is smaller than dynesty by a factor of 2.07 when not using distance marginalization, and a factor of 1.34 with distance marginalization enabled. This improvement also translates to the wall time; we find that nessai is 2.32 times faster than dynesty without distance marginalization enabled and 1.40 with it disabled; demonstrating that nessai can reduce both the number of likelihood evaluations and wall time. We therefore recommend using nessai with distance marginalization since, after the improvements presented in chapter 5, it is faster, but in this initial work we recommended using it without distance marginalization.

We also consider other aspects of nessai that distinguish it from other nested sampling implementations. The design of the algorithm allows for natural parallelization of the likelihood computation, since new samples are drawn in batches rather than sequentially, and this means it is trivial to reduce the wall time of analyses using more cores, particularly for analyses where computing the likelihood is the dominant cost. However, we also find that the rejection sampling necessary to ensure samples are distributed according to the prior becomes an increasing bottleneck as more cores are used. We revisit this in chapter 7 we when proposed an alternative formulation of nested sampling that avoids this. Finally, we also introduce the diagnostics available in nessai, most important of which are the insertion indices. These allow for over- or under-constrained runs to be identified without the need for a point of reference, e.g. a result from another sampler. We will see throughout this thesis the utility of these diagnostics when understanding results and tuning settings.

This work set the foundation for the work that is presented in the subsequent chapters. It demonstrated that nessai can be used to accelerate Bayesian inference for BBH signals but also highlighted areas of improvement; we found that this version was highly dependent on the choice of settings, we did not investigate applications without phase marginalization, we noted that rejection sampling can be a significant bottleneck and we did not test nessai on real gravitational-wave data. These are all aspects we investigate in the remainder of this thesis.

Improvements to nessai The initial analyses with nessai presented in chapter 4 made use of problem-specific reparametrizations and settings to mitigate bias during. In nessai, these biases arise predominantly from over- or under-constrained contours learnt by the normalizing flow and we therefore investigate specific methods of combating them.

We introduce constant volume mode, which fixes the radius of the latent contour for the duration of sampling based on a user-defined fraction $\rho_{\text{CVM}} \in (0, 1)$. This prevents small radii that can lead to over-constrained contours and consequently, biased results. We test constant volume mode on a likelihood that combines 1-dimensional distributions for which the previous settings tend to produce biased results, this includes uniform distributions and distributions that have maximal posterior probability at the prior bounds. The results show that the use of constant volume mode can reduce these biases, although they can still arise in some scenarios.

The second change we introduce is periodically resetting the flow, that is: periodically training from randomly initialized weights rather than starting from the previous trained flow. This minimizes biases that can arise when a training instance fails to converge on a suitable contour, which can otherwise have a cumulative effect over the course of sampling. We find this improves results for posteriors with maximal posterior probability at the prior bounds and for complex posterior distributions with long tails and multiple modes.

We reanalyse the 128 BBH injections from chapter 4 using `nessai` with constant volume mode and compare the results to our original results with `nessai`. With these changes, `nessai` still passes the P-P tests with and without distance marginalization but has improved sampling efficiency compared to previous results. This results in, on average, a 2.7 times reduction in the number of likelihood evaluations when distance marginalization is enabled and, 1.6 times when it is disabled. This means that with the updated settings using distance marginalization is more efficient than not, and we therefore recommend its use in analyses with `nessai` using constant volume mode. Furthermore, we find that there is an improvement in efficiency for all injections rather than just a subset. The largest improvements are for events with signal-to-noise ratios (SNRs) greater than 20 which still remain amongst the most costly analyses overall but show relative improvements of up to $\mathcal{O}(100)$ in wall time. In contrast, the events that show the least improvement are those with lower SNRs and those that exhibit multimodality in the likelihood surface, particularly in the right ascension and declination.

Sampling phase with `nessai` Next we turn to tackling more complex analyses with `nessai`, namely those where phase marginalization is no longer valid, for example because of the inclusion of higher-order modes, and the phase must instead be sampled. Preliminary studies find that the settings used in previous analyses with `nessai` are not suitable for such analyses and, and we therefore explore alternate parameterizations and settings.

Part of the challenge when sampling phase is the correlations between it and the other source angles, polarization and inclination. With this is mind, we propose sampling using the quaternions and define a mapping between them and the three source angles. We verify that the two sets of parameters are equivalent and test this parameterization alongside others introduced in previous works. Compared to the default parameterizations, these parameterizations show at best marginal improvement, however we find they are also more prone to over-constraining. We attribute this to the multimodality that is observed in the reparameterized space, particularly for the quaternions, see fig. 6.7, which `nessai` is not well suited to. Based on these findings, we perform all subsequent tests using the standard parameterization for the source angles. Future work could consider applications of the quaternions for other sampling algorithms better suited to multimodality.

We perform similar tests to those described in chapters 4 and 5; we analyse simulated BBH signals in a three-detector network. However, due to the increased computational cost, we only analyse 64 injections. The settings we converge upon leverage the problem-specific reparameterizations in order to pass a P-P test as we find simply tweaking the general settings available in `nessai` insufficient. With these updated settings, sampling phase requires on average 23 times more likelihood evaluations and takes 20 times longer than the equivalent analyses with phase marginalization. Whilst these results demonstrate that `nessai` can sample phase, the settings could likely be improved in a similar way to that shown in chapter 5 for analyses with phase marginalization.

`i-nessai` One of the fundamental bottlenecks in `nessai` is the rejection sampling required to ensure new samples are distributed according to the prior, as discussed in chapters 4 and 6. Removing the need for this step would therefore enable even faster inference. In Williams et al. [2], reproduced in chapter 7, we propose a modified nested sampling algorithm that is designed around the use of normalizing flows and importance weights, which we call `i-nessai`: importance nested sampling with normalizing flows. `i-nessai` incorporates normalizing flows similarly to `nessai` but removes the requirements for samples to be independently and identically distributed (i.i.d.) and distributed according to the priors, and added in order of increasing likelihood, whilst also allowing for samples to be added in batches. It draws upon existing variants of nested sampling including diffusive nested sampling [213], importance nested sampling [212, 332] and nested sampling via sequential Monte Carlo [216].

We outline the algorithm in section 7.3.8; it revolves around iteratively adding proposal distributions to construct a meta-proposal which should be proportional to the posterior. In the complete algorithm, each proposal distribution is a normalizing flow but we first demonstrate a simplified version using 2-dimensional Gaussian distributions on a toy example in section 7.2.1 before detailing the algorithm in section 7.3.

We first test `i-nessai` on a range of non-gravitational-wave problems: a Gaussian likelihood, Gaussian mixture likelihood and the Rosenbrock likelihood. These tests allow us to verify that the algorithm produces results that are consistent with standard nested sampling, i.e. `nessai`. Compared to `nessai`, we find that `i-nessai` produces more precise evidence estimates whilst requiring fewer likelihood evaluations.

Returning to gravitational-wave inference, we perform analyses similar to those described in chapters 4 and 5 and analyse 64 simulated signals from BBH coalescences in a three-detector network. The posterior distributions from these analyses pass a P-P test, further verifying that `i-nessai` can produce statistically consistent results. We analyse the same injections using `nessai` and `dynesty` and present the likelihood evaluations and wall times for these analyses in fig. 7.5. With a median of 6.5×10^5 likelihood evaluations, `i-nessai` is more than an order of magnitude more efficient than `dynesty`. This also translates to the wall time, where `i-nessai` is an order of magnitude faster, with a median wall time of 119 minutes minutes on a single core.

We apply `i-nessai` to the analysis of a GW190425-like binary neutron star (BNS) signal in a two-detector network. We use Reduced-Order-Quadratures (ROQs) to accelerate the likelihood calculation, assume low-spin priors and compare the results to those obtained with `dynesty` and `nessai`. The results from `i-nessai` show good agreement with the other two samplers and required more than 40 times fewer likelihood evaluations than `dynesty`. However, the improvement compared to `nessai` is less than 1.5 times, suggesting than `nessai` may be more efficient when applied to BNSs than BBH, though this requires further investigation.

Given that rejection sampling is no longer required, we revisit parallelizing the likelihood calculation. For a BBH signal, we find that even when using 16 cores to evaluate the likelihood, it is still the dominant cost, whereas for `nessai`, populating the pool of new samples becomes the dominant cost for 8 cores and above. In this particular example, when using 16 cores, inferences takes of order 15 minutes using `i-nessai`, compared to 40 minutes for `nessai`.

The algorithm we introduce, `i-nessai`, addresses the largest bottleneck in `nessai`; rejection sampling, whilst simultaneously removing sequential elements of the standard nested sampling algorithm. `i-nessai` has significant differences to standard nested sampling, such as what constitutes an iteration, how the evidence is calculated or when the algorithm terminates, all of which are designed around normalizing flows. This means that existing understanding of standard nested sampling does not necessarily translate to `i-nessai`, instead, this is an opportunity for further research into improving `i-nessai` and our understanding of the differences between it and standard nested sampling.

Applications of `nessai` Having developed and tested `nessai` and `i-nessai`, we finally focus on exploring applications of both algorithms, though predominantly `nessai`, to a range of different problems, focusing primarily on gravitational-wave events detected in O1, O2 and O3. This allows us to thoroughly test the proposed algorithm to make general recommendations about its application and use.

When analysing real data, there are systematic calibration errors that must be modelled and included in the inference, either by sampling the additional parameters or reweighting the final result. We find that nessai is better suited to using calibration reweighting, since sampling the calibration parameters can require including 50+ extra parameters and the normalizing flow-based proposal scales poorly when these are included and can result in over-constraining The downside to using reweighting is that it reduces the final number of samples obtained and can be computationally costly for long-duration signals.

We analyse the BBH events from O1 and O2 using three different waveform approximants: IMRPhenomPv2, IMRPhenomXP and IMRPhenomXPHM. For IMRPhenomPv2 and IMRPhenomXP, we enable phase marginalization, whereas for IMRPhenomXPHM we disable it. We compare our results to those released with Gravitational-Wave Transient Catalog 1 (GWTC-1), which used IMRPhenomPv2, and Gravitational-Wave Transient Catalog 2.1 (GWTC-2.1), which used IMRPhenomXPHM. We find nessai shows better agreement with GWTC-2.1 than with GWTC-1, with mean Jensen-Shannon divergences (JSDs) across all events of 0.005 bits compared to 0.011 bits. The agreement with GWTC-2.1 implies that nessai can produce results that are consistent with previous LIGO-Virgo-KAGRA Collaboration (LVK) analyses, however, the lack of publicly available run statistics prevents us from drawing conclusions about sampling efficiency. We can though compare between the different analyses with nessai; the analyses with IMRPhenomXPHM take an order of magnitude longer than those without, highlighting once more how sampling phase significantly impacts sampling efficiency with nessai. We investigate the differences between the results from nessai and GWTC-1 and find that whilst the difference in priors and choice of marginalization cannot account for the differences. Given this, and the agreement with GWTC-2.1 results, we attribute these differences to the use of bilby instead of LALInference and other possible changes in the data, such as to the power spectral densitys (PSDs) used.

There are two confident BNS detections from O1, O2 and O3: GW170817 and GW190425 and we reanalyse both using nessai. Following previous analyses, we perform low- and high-spin analyses for both events, however for GW170817 we fix the sky location based on the corresponding electromagnetic observations whereas for GW190425 we sample the sky location. We also repeat each analysis twice: once with calibration reweighting and once sampling the calibration errors. For both events, the low-spin analyses with nessai agree with the published results, however the analyses that use calibration reweighting show less over-constraining in the insertion indices than those that sample calibration errors. The high-spin results show larger differences; for GW170817 we observe differences in the estimated masses and tidal parameters when compared to GWTC-1. For GW190425, the results show better agreement with the initial results published in Abbott et al. [262]

than those released with GWTC-2.1 [9]. Furthermore, the insertion indices are not clearly over-constrained, so we attribute these difference to other factors, e.g. differences in the waveform and analysis framework.

We reanalyse three BBH events from O3: GW190412, GW190521, and GW191109. These events all contain features in the posterior distributions that may be challenging to sample and therefore serve to identify failure modes in `nessai`. The results for GW190521 do not show support for the secondary mode at more extreme mass ratios present in the LVK results, however the insertion indices do not show signs of over-constraining, suggesting these differences are not caused by `nessai`. Meanwhile, the results for GW190412 and GW191109 show slight signs of over-constraining, which is reflected in the corresponding posteriors. This suggests that whilst the settings used are suitable for more "vanilla" BBHs, they may need changing for more challenging analyses. It also demonstrates the importance of checking the insertion indices to identifying but that they cannot always explain the difference observed between results.

We also analyse an event using `i-nessai`, specifically GW170823 from O1 [7], and compare the results to those obtain with `nessai`. We once again perform analyses with `IMRPhenomPv2` and `IMRPhenomXPHM`, but find that for the latter `i-nessai`, the current settings do not converge or produce a low number of effective posterior samples. However, when using `IMRPhenomPv2`, the results are consistent with `nessai`, even when sampling the calibration parameters rather than using reweighting. This suggests that sampling the calibration parameters does not bias `i-nessai` in the same way it does `nessai`, though it does result in less effective posterior samples and leads to an increase in the wall time and number of likelihood evaluations. When using calibration reweighting, `i-nessai` produces results five times faster than `nessai` and yields more effective samples. This demonstrates the potential benefits of using `i-nessai` over `nessai`, however further testing and validation is required, which we leave for future work.

The next generation of ground-based detectors will bring new challenges for parameter estimation. With this in mind, we perform inference on a simulated GW150914-like signal in two different network configurations: Einstein Telescope (ET)-only and ET plus Cosmic Explorer (CE). In both cases, the analyses take significantly longer than the equivalent analyses in current generation detectors and they are dominated by the cost of evaluating the likelihood. Furthermore, in the ET-only case, the likelihood surface is highly multimodal, with up to eight distinct modes in some parameters. These results highlight the need for further investigation in parameter estimation techniques for next generation detectors and emphasize the importance of a multi-detector network.

These results from applying `nessai` to real gravitational-wave detections demonstrate that it can produce results that are consistent with previous analyses, and the observed differences are within the uncertainties of the previous analyses. However, there are cases where the settings used lead to over-constraining; this emphasizes the importance of validating results, e.g. via the insertion indices, which are a more stringent test than, for example, a P-P test. Furthermore, the results once again demonstrate that sampling phase and multimodal likelihood surfaces remain challenging with `nessai` and could be the focus of future investigations.

On the impact of `nessai` The discussion thus far has focused on demonstrable and quantifiable results produced with `nessai` and `i-nessai`. However, this overlooks an aspect of the work that is harder to evaluate: the time and effort spent developing the `nessai` software package [429] and the impact this had on its use outside the work presented in this thesis. Our aim was to produce a stable and user-friendly package that could be used in existing analyses without the need for extensive modifications to existing codes. With this in mind, we highlight research that has and is making use of `nessai`.

Starting with other applications for LVK data analysis; `nessai` has been used in waveform development, specifically in the development of `pySEOBNR`, where it was using to perform parameter estimation for validating the waveform model [478]. In Mérou et al. [479], `nessai` was used alongside `jax` [480] to search for long-duration transient continuous gravitational waves [481, 482]. Similarly, `ContinuousFUN` [483] leverages `nessai` and `i-nessai` as alternatives to MCMC to perform continuous gravitational-wave follow-up [484].

Laser Interferometer Space Antenna (LISA) data analysis poses different challenges to analyses for ground-based detectors [93]. Nonetheless, `nessai` has successfully been applied to a range of problems in LISA data analysis. In Linley [485], `nessai` is used in conjunction with newly developed "downsampling" approach to evaluate the feasibility of multi-band analyses. In Finch et al. [486], `nessai` is used to analyse galactic white dwarf binaries. Finally, in Chapman-Bird et al. [487], `nessai` is used to validate a machine learning-based approach for estimating the selection function in LISA.

Beyond gravitational-wave parameter estimation, `nessai` has also been used to analyze other astronomical transients. In Hayes et al. [488] it used to perform model comparison between different jet-structure models for short gamma-ray burst (GRB) and it was used alongside Gaussian processes to perform inference for kilonova light curves in Datrier [489]. Both of these applications rely heavily on parallelizing the likelihood evaluation using `nessai` to accelerate inference. Even further still from its original application, in Tait et al. [490] `nessai` is used to diagnose issues in experimental measurements of mechanical disk resonators.

Many of these studies are not applying `nessai` to CBC parameter estimation for current generation ground-based interferometers, or even gravitational-wave inference. This emphasizes that the `nessai` algorithm and code base are flexible and versatile. Furthermore, it can be used to accelerate inference in a wide range of applications and its impact is not limited to gravitational-wave astrophysics.

Broader conclusions There has been a Bayesian revolution and Bayesian inference is becoming an integral part of data analysis in many fields [491–494], including physics and astronomy [495, 496]. Although these fields are diverse and aim to answer different questions, they often share the same underlying challenges when leveraging Bayesian techniques. In many cases, the volume of data along coupled with the cost of performing inference can be a challenge and developing solutions to this is an ongoing effort, including in gravitational-wave data analysis [96, 198, 234]. One area of research that may prove key for developments in Bayesian inference is machine learning.

Machine learning is experiencing a similar revolution; it ballooned in popularity, it has already revolutionized certain fields, e.g. Jumper et al. [497], and current research is continually redefining what was thought to be possible [498]. Physics and astronomy are no exception to this, machine learning has the potential to revolutionize data analysis in these fields, and it is already seeing use. However, there are various hurdles that may hinder its widespread adoption, we now highlight some examples. The apparent opaqueness of machine learning algorithms can make results hard to interpret, though this is an active area of research and recent advances may help address this [499]. It can also be challenging to determine when machine learning is well suited to a problem; one can train such an algorithm to perform a given task, but that does not guarantee that it is applicable in practice. This can lead to an abundance of methods that do not address a particular problem, or the use of which over existing methods is difficult to justify. Finally, machine learning algorithms are often inherently general purpose, in the context of physics and astronomy this means that, whilst they can be applied to a plethora of problems, they do not always leverage all the domain knowledge available. Developing problem specific algorithms, i.e. that inherently incorporate domain knowledge rather than trying to learn it, may therefore be key but may require more in-depth machine-learning knowledge.

In this research, we choose an approach to leveraging machine learning that avoids some of the aforementioned hurdles: incorporating machine learning into existing analyses. This allows one to build on existing domain knowledge and expertise, rather than having to develop this from scratch and can help avoid the apparent opaqueness of end-to-end approaches, since the overall method is still based on familiar and well-understood concepts. Furthermore, such hybrid methods can be easier to use than bespoke machine learning approaches since they are drop-in replacements. They also serve as an intermediary between existing analyses and end-to-end machine learning analyses that, whilst often faster, can be harder to adopt due to the changes needed to existing infrastructure.

We have focused on the intersection of Bayesian Inference and machine learning and developed machine-learning augmented methods that can accelerate inference for gravitational-wave data analysis. Both methods are based on nested sampling and incorporate normalizing flows, however there are fundamental differences in how they are designed. The first is `nessai`, which is built around the standard nested sampling algorithm with minimal changes and uses normalizing flows to propose new samples. We have shown that, with various improvements, `nessai` produces unbiased results and can accelerate inference by up to four times compared to the standard analyses used by the LVK. The second is `i-nessai`, which is a novel nested sampling algorithm designed entirely around normalizing flows and addresses the bottlenecks in `nessai`. The results presented here show that `i-nessai` is up to an order-of-magnitude faster than the standard analyses, enabling analyses of BBH events on the order of 10 minutes.

These hybrid methods complement the end-to-end machine learning approaches for Bayesian inference that are being developed, e.g. Gabbard et al. [234] and Dax et al. [236], and present a tradeoff between speed, flexibility and ease-of-use. End-to-end machine learning methods can perform inference in fractions of a second but require extensive training and validation beforehand, whereas our methods may be slower but can be applied immediately and without the need for new infrastructure.

Overall, we believe that these methods can serve as a key component for future gravitational-wave analyses. They can be employed directly to enable faster and more efficient inference in existing analyses, which will be key for rapid follow-up of multi-messenger observations and for handling the ever-increasing number of events [74]. They also have the potential to aid in detecting new physics in the signals we observe by enabling analyses that could otherwise being computationally prohibitive. There is also scope to expand upon these methods and adapt them for applications to more challenging data analysis problems, such as performing inference for signals in LISA [486, 500, 501]. Finally, we hope that this work serves as an example of how machine learning can be leveraged to augment existing techniques and prompts others to consider applying such approaches to more data analysis problems.

Appendix A

Likelihood functions for testing samplers

We now describe the various likelihood functions that have been used for testing and validating `nessai` and `i-nessai`. All the likelihoods are implemented in `nessai-models` [439]. The QR codes included in this appendix link to interactive visualizations of the likelihoods in two dimensions.

A.1 Rosenbrock

The n-dimensional Rosenbrock function [454] has highly correlated parameters and is recognized as a challenging function to sample. We use the more involved variant [440, 455] where the log-likelihood is defined as

$$\ln \mathcal{L}_{\text{Rosenbrock}}(\boldsymbol{\theta}) = -\sum_{i=1}^{n-1} [100(\theta_{i+1} - \theta_i^2)^2 + (1 - \theta_i)^2]. \tag{A.1}$$

The evidence for this likelihood can only be computed analytically in two dimensions: $\ln Z = -5.804$ [208].

A.2 Gaussian mixture

An n-dimensional M-component Gaussian mixture likelihood has been used to validate nested sampling algorithms in various works. Here, we describe the version used in Moss [435] and Higson et al. [203]. The likelihood is defined as

$$\mathcal{L}_{\text{GM}}(\boldsymbol{\theta}) = \sum_{m=1}^{M} W^{(m)} \left(2\pi {\sigma^{(m)}}^2\right)^{-n/2} \exp\left\{\frac{-|\boldsymbol{\theta} - \mu^{(m)}|^2}{2{\sigma^{(m)}}^2}\right\}, \tag{A.2}$$

where $\mu^{(m)}$ and $\sigma^{(m)}$ are the mean and standard deviation of each component in all dimensions and $\sum_{m=1}^{M} W^{(m)} = 1$. We use the same hyperparameters [203, 435]: $M = 4$, $W^{(m)} = \{0.4, 0.3, 0.2, 0.1\}$, $\mu_1^{(m)} = \{0, 0, 4, -4\}$, $\mu_2^{(m)} = \{4, -4, 0, 0\}$, $\mu_n^{(m)} = 0 \; \forall \; n \in \{3, ..., n\}$ and $m \in \{1, ..., M\}$, and $\sigma^{(m)} = 1 \; \forall \; m \in 1, ..., M$.

A.3 Mixture model

This likelihood is designed to include various different characteristics that can lead to issues during sampling that lead to over-constrained or biased results. The model comprises independent variables which are distributed according to one of four distributions:

- Gaussian distribution with default parameters $\mu = 0$, $\sigma = 1$, with default prior $\mathcal{U}[-10, 10]$,
- Half-Gaussian distribution with parameters $\mu = 0$, $\sigma = 1$, with default prior $\mathcal{U}[0, 10]$,
- γ-distribution with parameters $k = 1.99$, with default prior $\mathcal{U}[0, 10]$,
- uniform distribution $\mathcal{U}[-5, 5]$, with default prior $\mathcal{U}[-5, 5]$.

The model is then constructed to contain an independent mixture of these distributions.

A.4 Skilling's statistical model

In Skilling [193], the author proposes a "statistical" example with a likelihood that represents a narrow Gaussian "spike" atop a wider Gaussian "plateau" both centred at the origin. The narrow spike has 100 times the posterior mass of the wide plateau. Skilling also considers a variant where the narrow peak is off-centre, and Brewer et al. [213] also consider a variant of this.

For this model, the priors are uniform on $\mathcal{U}[-0.5, 0.5]^n$ with $n = 20$ by default, and the likelihood is given by

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi} v} \exp\left(-\frac{\theta_i^2}{2v^2}\right) + 100 \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi} u} \exp\left(-\frac{(\theta_i - \mu_u)^2}{2u^2}\right), \tag{A.3}$$

with $v = 0.1$, $u = 0.01$ and μ_u is the offset of the narrow peak which Skilling sets to $\mu_u = 0$ and $\mu_u = 0.2$ and Brewer et al. set to $\mu_u = 0.031$. To distinguish it from Skilling's variant, we refer to the latter as the *Brewer likelihood*. For all three variants of this likelihood, the log-evidence is $\ln Z = \ln(101) \approx 4.615$ [193, 213].

Appendix B

Details of normalizing flow examples

This appendix contains further details about of the examples included in section 3.3. Code to reproduce all the examples is available at [438].

B.1 Example of an affine coupling-based normalizing flow

In this example, an affine coupling-based normalizing flow is trained to transform samples from the 2-dimensional Rosenbrock function, as defined in appendix A.1, to samples from a unit Gaussian.

The normalizing flow comprises two affine coupling transforms with alternating masks. After each transform is a batch normalization transform (see section 3.3.4). Each affine coupling transform contains a conditioner network that uses a residual neural network (ResNet) style architecture with four blocks, each containing 2 layers with 64 neurons and a Rectified Linear Unit (ReLU) activation function.

We train the flow using 10 000 samples from the 2-dimensional Rosenbrock, which we obtain using rejection sampling, and reserve 10% of the samples for computing the validation loss. We use a batch size of 1000 with the adaptive moment estimation (Adam) optimizer [325] and a learning rate of 0.001. We train for a maximum of 500 epochs but stop training early if there is no improvement in validation loss after 20 epochs.

Figure 3.2 is produced by passing 2000 samples from the Rosenbrock function through each of the transforms and plotting the output at each stage.

B.2 Example of training a flow with weights

In this example, a normalizing flow is trained to approximate a 2-dimensional multivariate Gaussian distribution G_t with mean and covariance

$$\boldsymbol{\mu_t} = \begin{pmatrix} 2 & 2 \end{pmatrix}, \quad \boldsymbol{\Sigma_t} = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}. \tag{B.1}$$

However, instead of using samples from the target distribution, we use samples from a different training distribution and use weights in Kullback-Leibler divergence (KLD). The training data is drawn from another 2-dimensional multivariate Gaussian distribution with mean and covariance

$$\boldsymbol{\mu}_d = \begin{pmatrix} 0 & 0 \end{pmatrix}, \quad \boldsymbol{\Sigma}_d = \begin{pmatrix} 25 & 0 \\ 0 & 25 \end{pmatrix}. \tag{B.2}$$

The weights w_i are then given by the ratio of the probability density functions (PDFs)

$$w = \frac{p_{G_t}(\boldsymbol{x})}{p_{G_d}(\boldsymbol{x})}, \tag{B.3}$$

where p_{G_t} is the PDF of the target and p_{G_d} is the PDF of the training data.

The normalizing flow comprises two affine coupling transforms with alternating masks. Note that for this example, we do not include batch normalization or *LU* factorization. It is trained using the Adam optimizer [325] with batch size of 1000 and learning rate $\eta = 0.001$. We 10 000 random samples from the target distribution and use 10% for validation. We train for a maximum of 500 epochs but stop training early if there is no improvement in validation loss after 50 epochs. We draw 5000 samples from the trained flow and use these to produce fig. 3.4.

B.3 Application of LU factorization

In this example, the initial distribution is a 2-dimensional multivariate Gaussian distribution with mean and covariance matrix

$$\boldsymbol{\mu} = \begin{pmatrix} 1.0 & 1.0 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1.2 & 0.9 \\ 0.9 & 1.2 \end{pmatrix}. \tag{B.4}$$

The aim is to transform samples from the initial distribution using the *LU* factorization such that the resulting samples are approximately normally distributed. To do so, we construct a distribution consisting of 2-dimensional unit Gaussian and an *LU* factorization layer, implemented using `glasflow` [502]. The only trainable parameters are those of the *LU* factorization layer.

We train the *LU* factorization layer using the Adam optimizer [325] for 1000 iterations with learning rate $\eta = 0.05$. At each iteration, 1000 samples are drawn from the initial distribution. The forward KLD is computed using the samples and eq. (3.22) and weights are then updated using backpropagation. Figure 3.3 is produced by drawing 50 000 samples from the initial distribution and applying the learnt *LU* factorization.

B.4 Example of the topological limitations of normalizing flows

In this example, an affine-coupling based normalizing flow is trained using samples from a bimodal distribution. The distribution comprises two equal weighted 2-dimensional multivariate Gaussian distributions with means $\boldsymbol{\mu}_1 = (-2.5, -2.5)$ and $\boldsymbol{\mu_2} = (2.5, 2.5)$ and unit variances.

The normalizing flow comprises four affine coupling transforms with alternating masks. Before each transform is an *LU* factorization transform (see section 3.3.4) and after each is a batch normalization transform (see section 3.3.4). Each affine coupling transform contains a conditioner network that uses a ResNet style architecture with four blocks, each containing 2 layers with 16 neurons and a ReLU activation function.

The flow is trained using the Adam optimizer [325] with batch size of 1000 and learning rate $\eta = 0.001$. We 4000 random samples from the target distribution and use 10% for validation. We train for a maximum of 500 epochs but stop training early if there is no improvement in validation loss after 20 epochs. Cosine annealing [329] is used to reduce the learning rate of the course of training.

Each subfigure in fig. 3.5 is produced by evaluating the different PDFs on a 100×100 grid. From left to right: the PDF of the target distribution, the PDF of the Gaussian latent distribution and normalizing flow PDF computed using eq. (3.7).

Appendix C

Gravitational-wave parameter & injections reference

C.1 Injection parameters

Table C.1: Intrinsic parameters for injected signals used in various chapters. The masses quoted are detector-frame masses. For details of the different parameters, see table C.3

	$m_1 \mathrm{M}_\odot$	m_2 $\mathrm{M}_\odot$	a_1	a_2	θ_1	θ_2	ϕ_{JL}	ϕ_{12}	Λ_1	Λ_2
GW150914-like BBH	36	29	0.4	0.3	0.5	1.0	0.3	1.7	-	-
GW190425-like BNS	1.725	1.607	0.0092	0.0102	0.2839	0.2978	0.4823	0.3266	149.3519	984.6168

Table C.2: Extrinsic parameters for different injections. For details of the different parameters, see table C.3

	α	δ	d_{L}	ψ	θ_{JN}	φ_c
GW150914-like BBH	1.375	-1.2108	350 Mpc	2.659	0.4	1.3
GW190425-like BNS	4.6980	0.3122	45 Mpc	0.0264	0.6688	0.0264

C.2 Parameter reference

Table C.3: Reference of parameter used to described compact binary coalescence (CBC) signals and the corresponding symbols

Parameters	Label	Unit
Detector-frame mass of the ith component	m_i	$\mathrm{M}_\odot$
Detector-frame chirp mass	$\mathcal{M}$	$\mathrm{M}_\odot$
Detector-frame total mass	M	$\mathrm{M}_\odot$
Asymmetric mass ratio (≤ 1)	q	-
Luminosity distance	d_L	Mpc
Right ascension	α	rad
Declination	δ	rad
Azimuth	ϵ	rad
Zenith	κ	rad
Time of coalescence measured at the geocentre	t_c	s
Time of coalescence measured at an interferometer (IFO)	t_IFO	s
Inclination angle between the total angular momentum $\boldsymbol{J}$ and the direction of propagation $\hat{n}$	θ_JN	rad
Inclination angle between the orbital angular momentum $\boldsymbol{L}$ and the direction of propagation $\hat{n}$	ι	rad
Polarization angle	ψ	rad
Phase	φ_c	rad
Dimensionless spin magnitude of the ith component	a_i	-
Aligned spin of the ith component	χ_i	-
Spin tilt angles	θ_i	rad
Difference between the azimuthal angles of each spin vector relative to the orbital angular momentum	ϕ_{12}	rad
Difference between the azimuthal angles of the total and orbital angular momentum	ϕ_JL	rad
Effective inspiral spin [124, 125]	χ_eff	-
Effective precessing spin [126, 127]	χ_p	-
Dimensionless tidal deformability of the ith component	Λ_i	-
Combined dimensionless tidal deformability [468, 469]	$\tilde{\Lambda}$	-
Relative difference in the combined tidal deformability [468, 469]	$\Delta\tilde{\Lambda}$	-

Appendix D

Appendices for Nested sampling with normalizing flows

D.1 Boundary inversion

In section 4.3 we describe *boundary inversion* which we introduce to avoid under-sampling regions which are close to the prior bounds. The user defines which parameters the inversion can be applied to and before training the sampler determines if it should be applied to each parameter using the following steps:

1. Compute the density of samples over the specified range and find the maximum value.

2. Compute the fraction of the density that lies within the initial and final $m\%$ of the bounds, i.e. $[0, 0.1]$ and $[0.9, 1.0]$ if the parameter is defined on $[0, 1]$.

3. Choose to apply inversion to the bound with the highest density if it is at least $n\%$ of the maximum density and the density at the bound is non-zero.

From our testing the percentages m and n default to 10% and 50% respectively but can be changed by the user. We consider two methods for applying the inversion:

- **duplication:** duplicate the set of points and apply the inversion to the duplicates,

- **splitting:** randomly select half of the points to apply the inversion to.

We find that duplication generally provides more consistent results but at the cost of the increasing the training time. As such, we recommend using splitting when inversion is applied to more than two parameters.

D.2 Gravitational-wave priors

Table D.1: Prior distributions used for each parameter for gravitational-wave parameter estimation. Their corresponding labels and the lower and upper bounds are included where applicable.

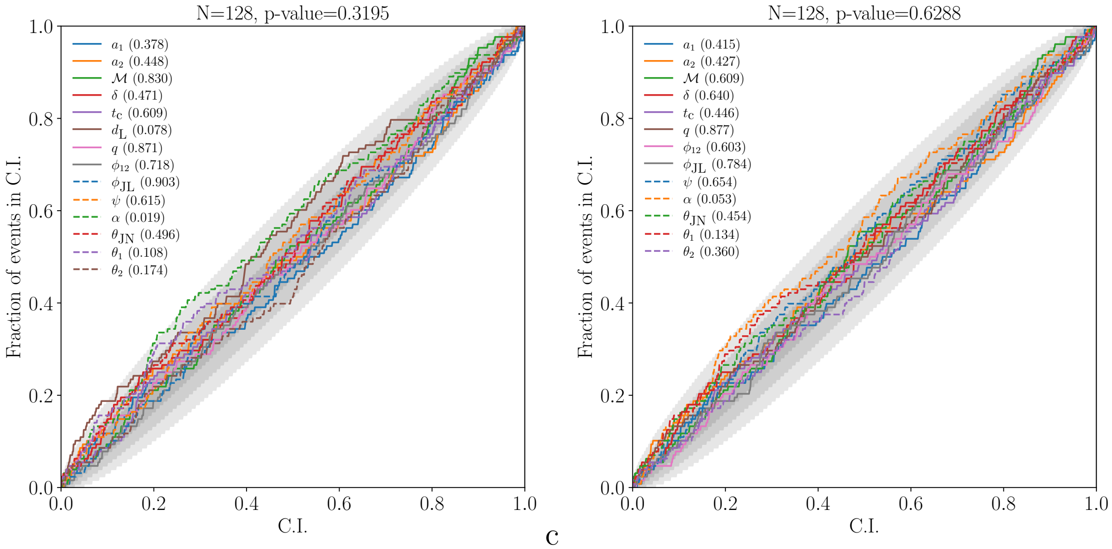
Parameters	Label	Prior	Bounds
Chirp mass	$\mathcal{M}$	Uniform	(25–35) $\mathrm{M}_\odot$
Asymmetric mass ratio	q	Uniform	$[0.125, 1.0]$
Luminosity distance	d_L	Uniform in co-moving volume	(100–2000) Mpc
Right ascension	α	Uniform	$[0, 2\pi]$
Declination	δ	Cosine	-
Reference time at geocentre	t_c	Uniform around trigger time	$[-0.1, 0.1]$
Inclination	θ_JN	Sine	-
Polarization	ψ	Uniform	$[0, \pi]$
Phase	ϕ_c	Uniform	$[0, 2\pi]$
Dimensionless spin magnitudes	a_i	Uniform	$[0, 0.99]$
Spin tilt angles	θ_i	Sine	-
Difference between the azimuthal angles of each spin vector relative to the orbital angular momentum	ϕ_{12}	Uniform	$[0, 2\pi]$
Difference between the azimuthal angles of the total and orbital angular momentum	ϕ_JL	Uniform	$[0, 2\pi]$

D.3 nessai sampling settings

Table D.2: Settings used for nessai for gravitational-wave inference in Williams et al. [1]. These are split into three categories: general settings which control aspects of the sampler such as the choice of latent prior or pool-size, flow hyperparameters which determine the configuration of the normalizing flow and flow training settings which control the training process. Different batch sizes were used for runs with and without distance marginalization, and this is shown in parentheses. For a complete description of each, see the documentation [429].

nessai settings					
General settings		Flow hyperparameters		Flow training settings	
Training frequency	None	Coupling transformations	6	Optimizer	Adam
Cooldown	200	Linear transformation	LU	Learning rate	0.001
Base pool-size	2000	Network type	residual neural network (ResNet)	Batch size	2000 (4000)
Update pool-size	True	Layers per network	2	Max. epochs	500
Draw-size	2000	Neurons per layer	32	Patience	50
Train on empty	True	Activation	Rectified Linear Unit (ReLU)		
Weights reset	False	Batch-Normalisation	Intra-transforms		
Latent prior	Truncated Gaussian				
Rescale	True				
Update bounds	True				

D.4 P-P tests for `dynesty`



c

(a) With phase marginalization

(b) With phase and distance marginalization

Figure D.1: Probability-probability plots (P-P plots) showing the confidence interval versus the fraction of the events within that confidence interval for the posterior distributions obtained using `dynesty` for 128 simulated compact binary coalescence signals produced with `bilby` and `bilby_pipe`. The 1-, 2- and 3-σ confidence intervals are indicated by the shaded regions and p-values are shown for each of the parameters and the combined p-value is also shown. We use the settings described in Romero-Shaw et al. [122] except for the number of live points which we increase to 2000.

Appendix E

Additional results for analyses with phase

E.1 Baseline analysis when sampling phase

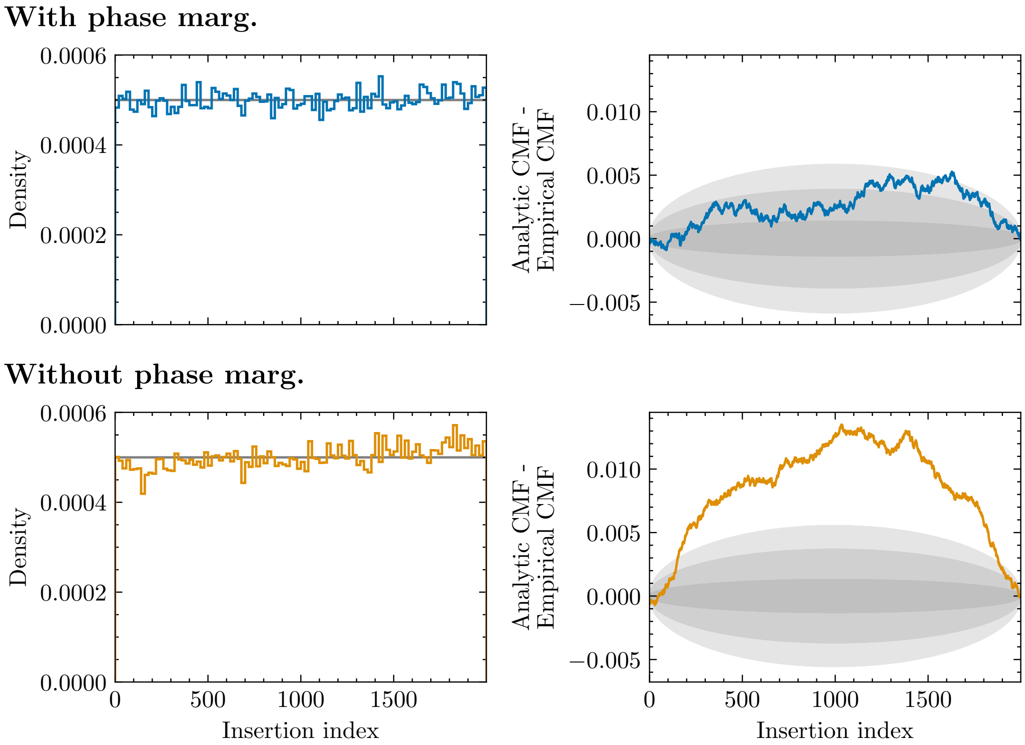


Figure E.1: Insertion indices produced by `nessai` when sampling with (top) and without (bottom) phase marginalization with the baseline settings discussed in section 6.1. **Left:** histogram of the insertion indices, with the analytic probability mass function (PMF) shown in grey. **Right:** comparison of the empirical distribution of the insertion indices to the analytic distribution. The shaded regions indicate the 1-, 2- and 3-σ confidence regions.

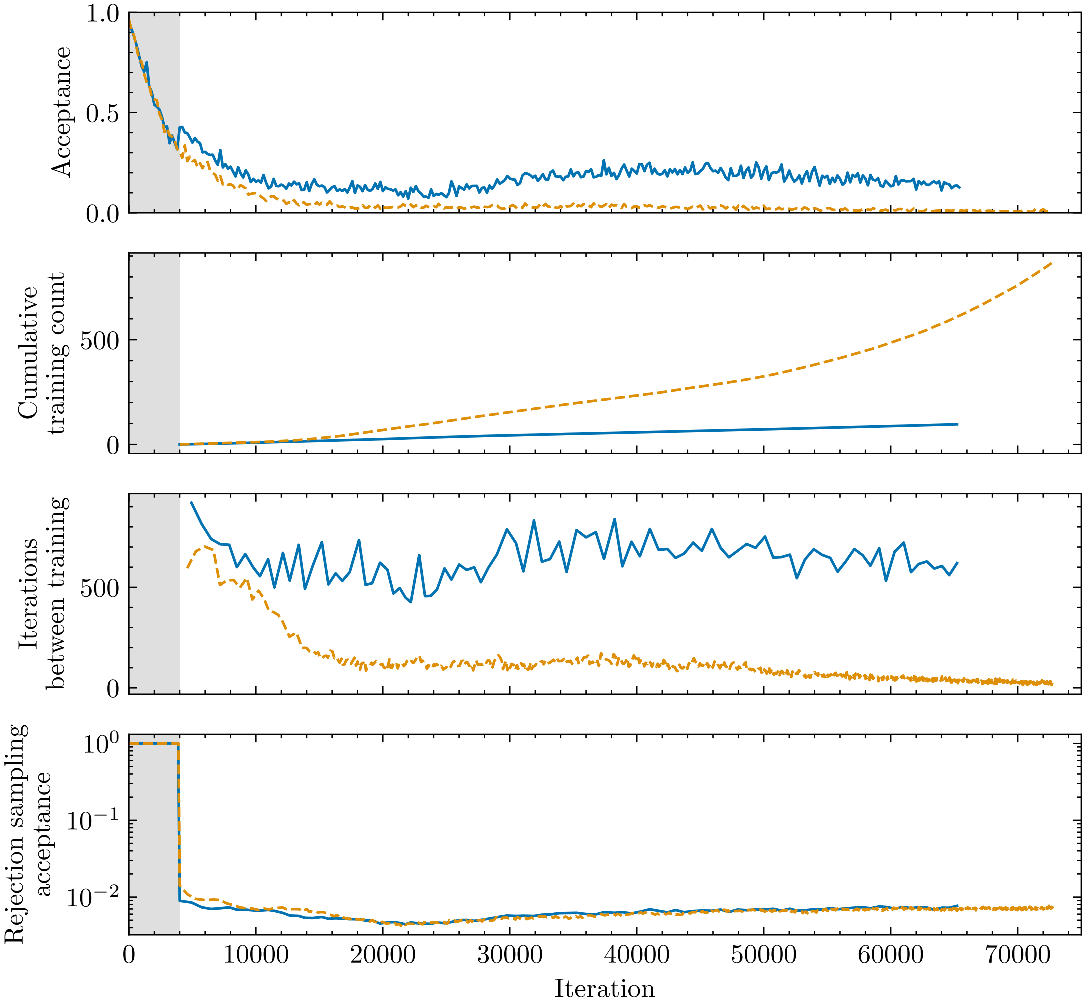


Figure E.2: Subset of the diagnostic plots from the baseline run with `nessai`, with (blue) and without (orange) phase marginalization. From top to bottom these are: the acceptance based on the likelihood threshold at a given iteration, the cumulative training count, number of iterations between training and rejection sampling efficiency. These highlight how disabling phase marginalization significantly reduces the acceptance when sampling with the default settings used in previous analyses.

E.2 Insertion indices for runs with different source angle reparameterizations

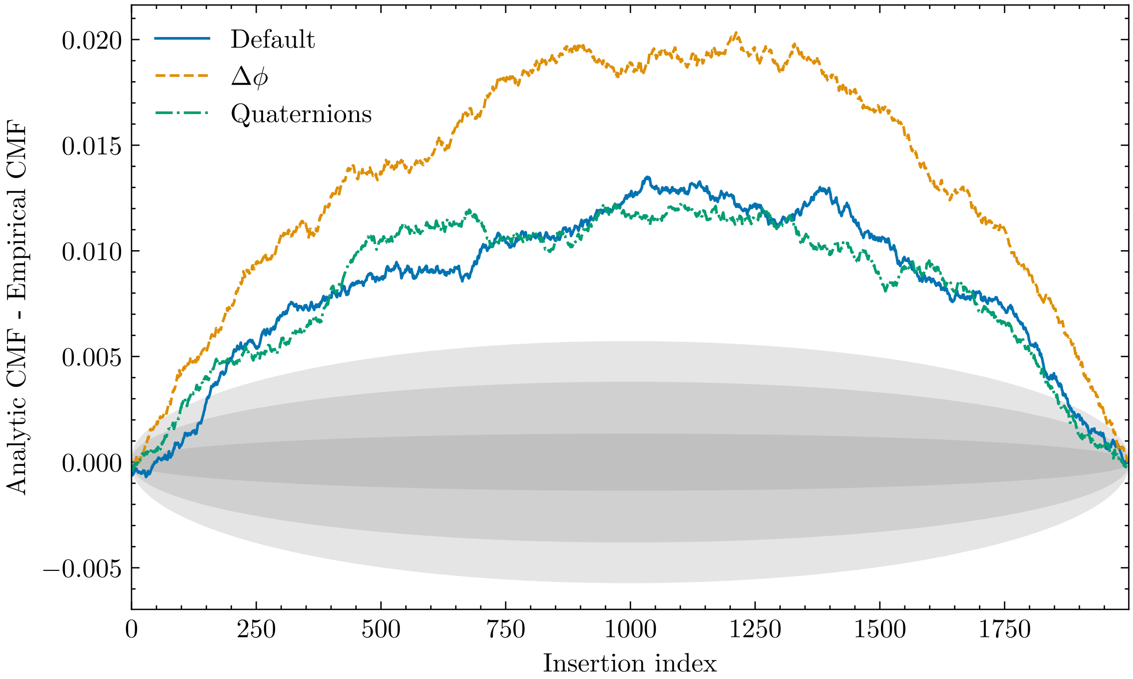


Figure E.3: Comparison of analytic cumulative mass function (CMF) and the empirical CMF for the insertion indices corresponding to three analyses using `nessai` with different reparameterizations for the source angles: the default in `nessai` (blue), $\Delta\varphi_c$ (orange) and the quaternions (green). See sections 6.2 and 6.3 for details about the reparameterizations. The shaded regions indicate the 1-, 2- and 3-σ confidence regions.

E.3 Probability-probability with phase marginalization

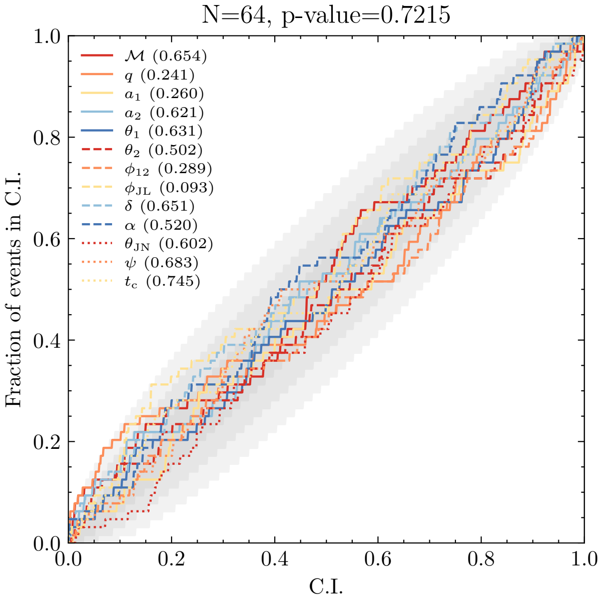


Figure E.4: Probability-probability plot (P-P plot) showing the confidence interval versus the fraction of the events within that confidence interval for the posterior distributions obtained using **`nessai`** with phase marginalization for 64 simulated compact binary coalescence signals produced with `bilby` and `bilby_pipe`. The 1-, 2- and 3-σ confidence intervals are indicated by the shaded regions and p-values are shown for each of the parameters and the combined p-value is also shown.

Appendix F

Appendices for Importance nested sampling with normalizing flows

F.1 Methods for constructing the next proposal distribution

We test the quantile-based method and the entropy-based methods for constructing the next proposal distribution described in section 7.3.1 and consider the stability and number of iterations required to converge. We find that the quantile-based method for determining the next level is sensitive to outliers in the meta-proposal $Q(\theta)$. This leads to large changes in the number of discarded samples M_j between iterations which in turn can make the algorithm unstable. In contrast, the entropy-based approach is far more stable and leads to smoother variations in the number of discarded samples which we attribute to the use of the log-weights. Additionally, we find that the entropy-based method converges quicker than the quantile-based because the prior volume shrinks faster. As such, we use the entropy-based method for all our experiments.

F.2 Validating the variance estimator

We validate the unbiased estimator for the variance of the evidence from eq. (7.2) for `i-nessai` using the Gaussian and Gaussian Mixture likelihoods described in section 7.5.1. We use the results from the analyses described in section 7.5.1 and produce probability-probability (P-P) plots comparing the observed distribution of evidences and a Gaussian distribution with the mean equal to the true evidence and the standard deviation estimated using eq. (7.2) averaged over the 50 runs per dimensions. The results are presented in fig. F.1 and show good agreement between the estimated and observed distributions.

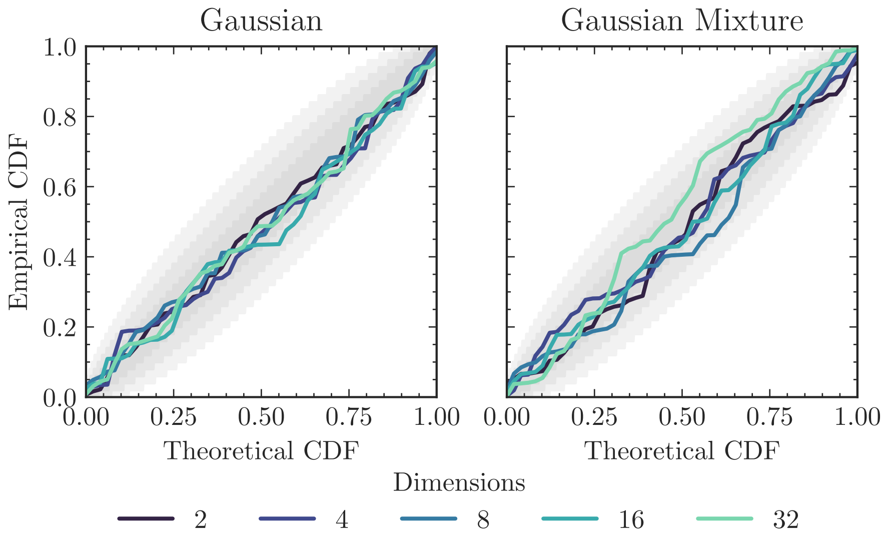


Figure F.1: probability-probability plots (P-P plots) for the estimated evidences for the Gaussian and Gaussian Mixture models described in section 7.5.1 for $n = \{2, 4, 8, 16, 32\}$. The theoretical distribution is assumed to be a Gaussian centred around the true evidence with the standard deviation given by the estimated standard deviation eq. (7.2) averaged over 50 analyses per dimension. The 1-, 2- and 3-σ confidence intervals are indicated by the shaded regions.

F.3 Insertion indices test for the Rosenbrock likelihood

In section 7.5.3, we analyse the Rosenbrock likelihood for $n = \{2, 4, 8\}$ using `nessai` and `i-nessai` and find that the estimated log-evidence disagreed as shown in fig. 7.3. In Fowlie et al. [218], the authors proposed using order-statistics to check the convergence of nested sampling runs. This involves computing an insertion index for each new sample according to where it is inserted into the current ordered set of live points. If new samples are distributed according to the prior, then the overall distribution of the insertion indices should be uniform. This can be quantified by computing a p-value for the overall distribution using the Kolmogorov-Smirnov statistic [219, 220] for discrete distributions [221]. We compute p-values for each analysis and presented the results in fig. F.2. If the results are unbiased then the distribution of p-values should be uniform on $[0, 1]$, however we observe that for $n > 2$ the distributions are not uniform, indicating problems during sampling. This agrees with the observation that for $n = \{4, 8\}$, with the settings used, `nessai` over-estimates the log-evidence.

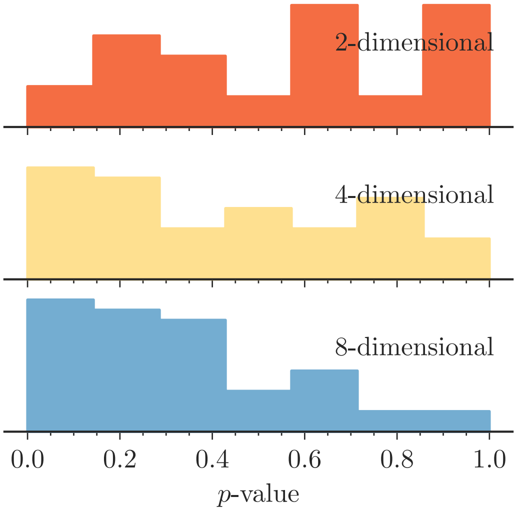
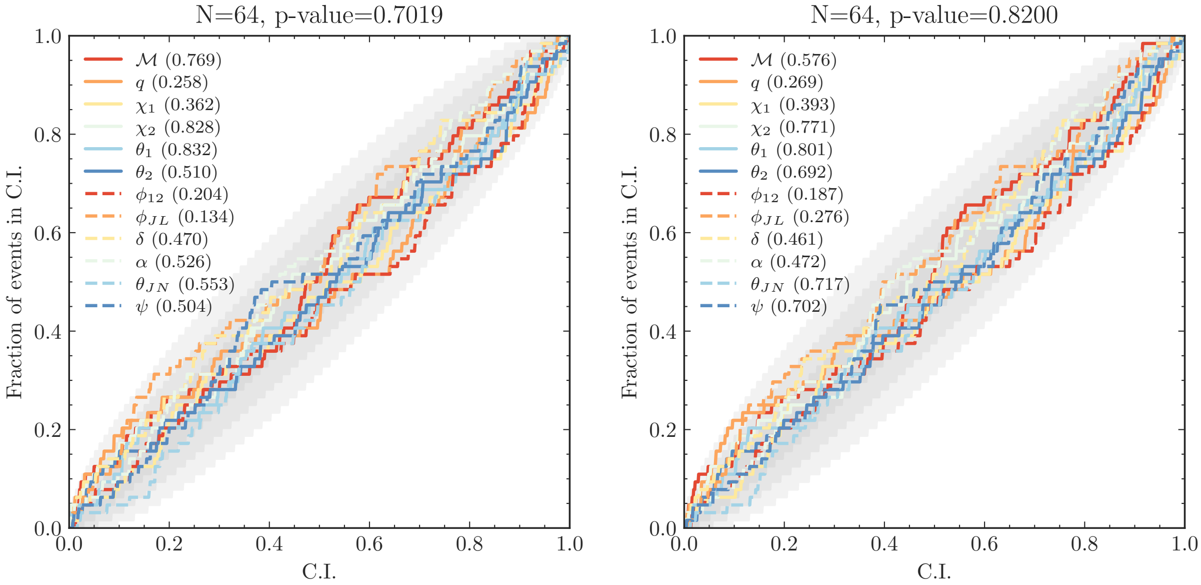


Figure F.2: Distribution of the p-values for the insertion indices [218] when analysing the Rosenbrock likelihood 50 times using `nessai` with $n = \{2, 4, 8\}$.

F.4 Probability-probability plots for other samplers



(a) `nessai`

(b) `dynesty`

Figure F.3: P-P plots showing the confidence interval versus the fraction of the events within that confidence interval for the posterior distributions obtained using `nessai` and `dynesty` for 64 simulated compact binary coalescence signals produced with `bilby` and `bilby_pipe`. The 1-, 2- and 3-σ confidence intervals are indicated by the shaded regions and p-values are shown for each of the parameters and the combined p-value is also shown.

F.5 Jensen-Shannon divergence for comparing marginal posterior distributions

We compute the Jensen-Shannon divergence (JSD) between the marginal posterior distributions obtained in section 7.5.5 as described in Romero-Shaw et al.[122]. We use bootstrapping to compare 100 different realizations of 5,000 samples from each posterior and quote the mean JSD and standard deviation in table F.1. Following Ashton and Talbot [233], for 5,000 posterior samples, the JSD threshold is 2×10^{-3} nats. The divergences between `i-nessai` and `nessai` agree for all the parameters, whereas for `dynesty` there is marginal disagreement in the posteriors for the aligned spin χ_1. However, since `nessai` and `i-nessai` are in agreement, we do not investigate this further in this work.

Table F.1: JSD in units of 1×10^{-3} nats for the marginal posterior distributions between `nessai`, `i-nessai` and `dynesty`. Values shown are the mean and the 1-σ quantiles computed over 100 different realisations of 5,000 samples.

	`dynesty-nessai`	`dynesty-i-nessai`	`nessai-i-nessai`
$\mathcal{M}$	$0.61^{0.20}_{-0.20}$	$0.69^{0.22}_{-0.19}$	$0.53^{0.21}_{-0.13}$
q	$0.52^{0.29}_{-0.16}$	$0.36^{0.22}_{-0.11}$	$0.30^{0.18}_{-0.08}$
χ_1	$2.24^{0.78}_{-0.55}$	$2.61^{0.77}_{-0.59}$	$0.53^{0.27}_{-0.17}$
χ_2	$1.68^{0.60}_{-0.46}$	$1.93^{0.47}_{-0.54}$	$0.73^{0.22}_{-0.22}$
δ	$1.37^{0.29}_{-0.28}$	$1.47^{0.34}_{-0.28}$	$1.59^{0.38}_{-0.31}$
α	$1.04^{0.22}_{-0.25}$	$1.15^{0.25}_{-0.27}$	$1.37^{0.30}_{-0.28}$
θ_{JN}	$0.71^{0.22}_{-0.17}$	$0.74^{0.21}_{-0.21}$	$0.79^{0.26}_{-0.23}$
ψ	$0.18^{0.13}_{-0.06}$	$0.21^{0.15}_{-0.09}$	$0.19^{0.10}_{-0.09}$
t_{c}	$1.29^{0.42}_{-0.25}$	$1.56^{0.31}_{-0.39}$	$1.57^{0.39}_{-0.33}$

Appendix G

Additional results for applications of nessai

G.1 Insertion indices for analyses with calibration errors

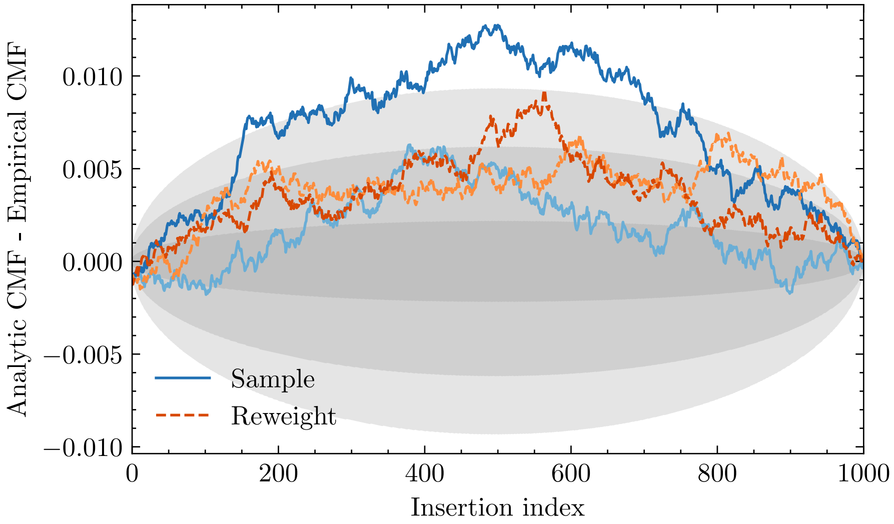


Figure G.1: Comparison of analytic cumulative mass function (CMF) and the empirical CMF for the insertion indices corresponding to two analyses of GW150914 with `nessai`: sampling the calibration parameters (blue) and calibration reweighting (orange). The two parallel runs for each analysis are shown in different shades of blue and orange. The shaded regions indicate the 1-, 2- and 3-σ confidence regions.

G.2 Prior distributions for O1 and O2 events in GWTC-1 and GWTC-2.1

We compare the prior distributions used to analyse binary black hole (BBH) events from O1 and O2 in the Gravitational-Wave Transient Catalog 1 (GWTC-1) [7] and Gravitational-Wave Transient Catalog 2.1 (GWTC-2.1) analyses [9]. The priors for the extrinsic parameters and six spin parameters are consistent between both analyses, whereas the priors on the masses are different. Both analyses use priors that are defined in chirp mass and mass ratio, but are uniform in component mass (see section 2.10.3). However, as a result of using an improved waveform model, GWTC-2.1 allows for mass ratios as low as 0.05 for all events whereas GWTC-1 only allows values as low as 0.125 [7, 9]. Furthermore, the two analyses use different constraints on the component masses, GWTC-1 uses event-specific constraints on component masses, and for some events total mass, whereas GWTC-2.1 constrains the component mass to $m_i \in [1, 1000]$ $\mathrm{M}_{\odot}$. The priors on component masses for each event are shown in fig. G.2.

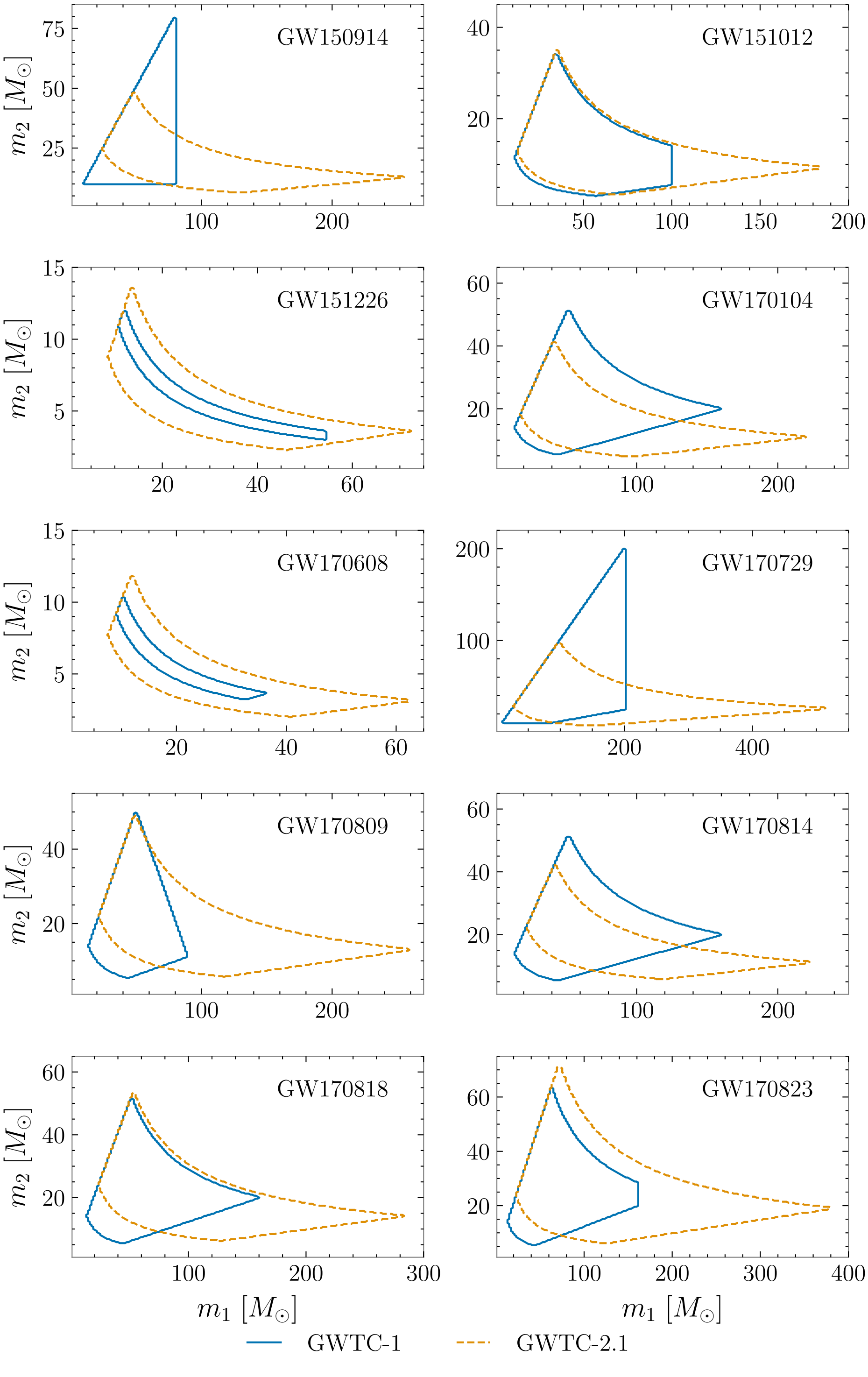


Figure G.2: Priors on component masses used in GWTC-1 and GWTC-2.1 for the analysis of the 10 BBH events from O1 and O2.

G.3 Further analysis of GW150914

In section 8.2, we observe sizeable differences between the posterior distributions for GW150914 obtained using `bilby` and `nessai` with `IMRPhenomPv2` compared to the samples released with GWTC-1 [7] that used the same waveform. To better understand this difference, we perform multiple analyses with difference priors and marginalizations and compare the results to those presented in 8.2. The analysis in section 8.2 used priors and other settings based on the reanalysis of GW150914 from GWTC-2.1 [9]. As shown in appendix G.2, these priors differ significantly from those used in GWTC-1. We therefore repeat the analysis with priors that match GWTC-1. We also perform an analysis with phase marginalization disabled, as it was not enabled in the GWTC-1 analyses. All other settings are left unchanged.

We present the posterior distributions for chirp mass and mass ratio in fig. G.3, this includes five sets of samples: the samples from GWTC-1, the samples from GWTC-2.1 [9, 467], samples from an analysis using `nessai` with the GWTC-2.1 priors, `IMRPhenomPv2` and phase marginalization enabled, samples from an analysis using `nessai` with the GWTC-1 priors, `IMRPhenomPv2` and phase marginalization enabled and samples from an analysis using `nessai` with the GWTC-1 priors, `IMRPhenomPv2` and phase marginalization disabled. The results from GWTC-1 are the clear outlier, favouring higher chirp masses and more equal masses. Changing the priors and phase marginalization does not significantly impact the posterior distributions, suggesting that the discrepancy is caused by another difference between the runs. The results obtained with `nessai` more closely resemble those from GWTC-2.1, however this analysis used `IMRPhenomXPHM` which includes higher-order modes, so some differences in the results are expected.

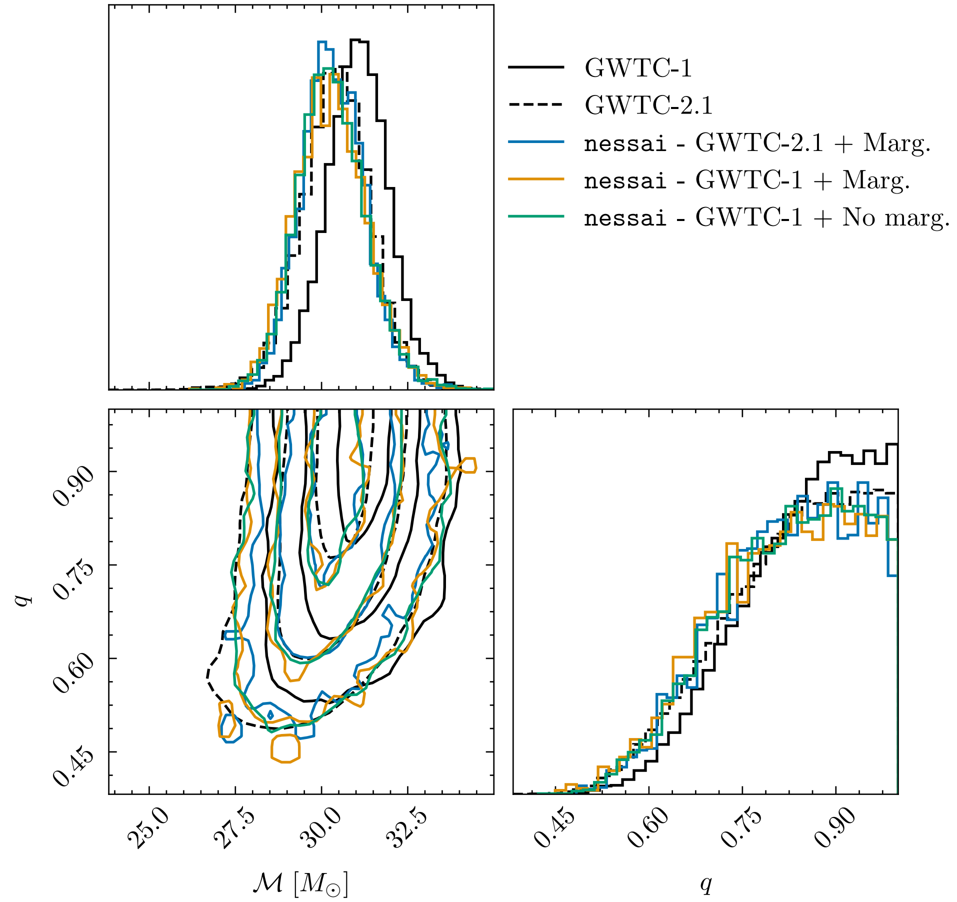


Figure G.3: Comparison of posterior distributions GW150914 for chirp mass and mass ratio for samples from five different analyses. The sets of samples are: the samples from GWTC-1 (black solid line), the samples from GWTC-2.1(black dashed line), samples from an analysis using `nessai` with the GWTC-2.1 priors, `IMRPhenomPv2` and phase marginalization enabled (blue solid line), samples from an analysis using `nessai` with the GWTC-1 priors, `IMRPhenomPv2` and phase marginalization enabled (orange solid line) and samples from an analysis using `nessai` with the GWTC-1 priors, `IMRPhenomPv2` and phase marginalization disabled (green solid line). The 2-dimensional posteriors show the 50%, 86% and 98% contours.

G.4 i-nessai analysis with higher-order modes

We present a preliminary analysis of GW170823 using i-nessai with IMRPhenomXPHM and compare to results obtained with nessai in section 8.2.1 and to results released with GWTC-2.1 [9]. We use bilby and bilby_pipe [198] with the same data and priors as used in section 8.2.1 and, based on the results presented in section 8.4, use calibration reweighting to handle the calibration uncertainty. In order for i-nessai to converge, we increase the number of samples drawn per proposal to $N_j = 50\,000$ and set $\rho = 0.9$ with the entropy-based method for determining the next proposal, without these changes we find that i-nessai fails to converge.

In fig. G.4, we present the posterior distribution obtained with i-nessai and compare it to the results from nessai and GWTC-2.1. The results from i-nessai are consistent with the other analysis but there is significant statistical noise from the comparatively low number of posterior samples. Similarly to the analyses presented in section 8.4, these results are produced by combining two individual runs, these took 5 and 7 hours respectively, compared to 11 and 28.6 hours for the equivalent runs with nessai. This suggests that i-nessai can be used to accelerate such analyses, but would likely require more robust settings to reliably produce sufficient posterior samples.

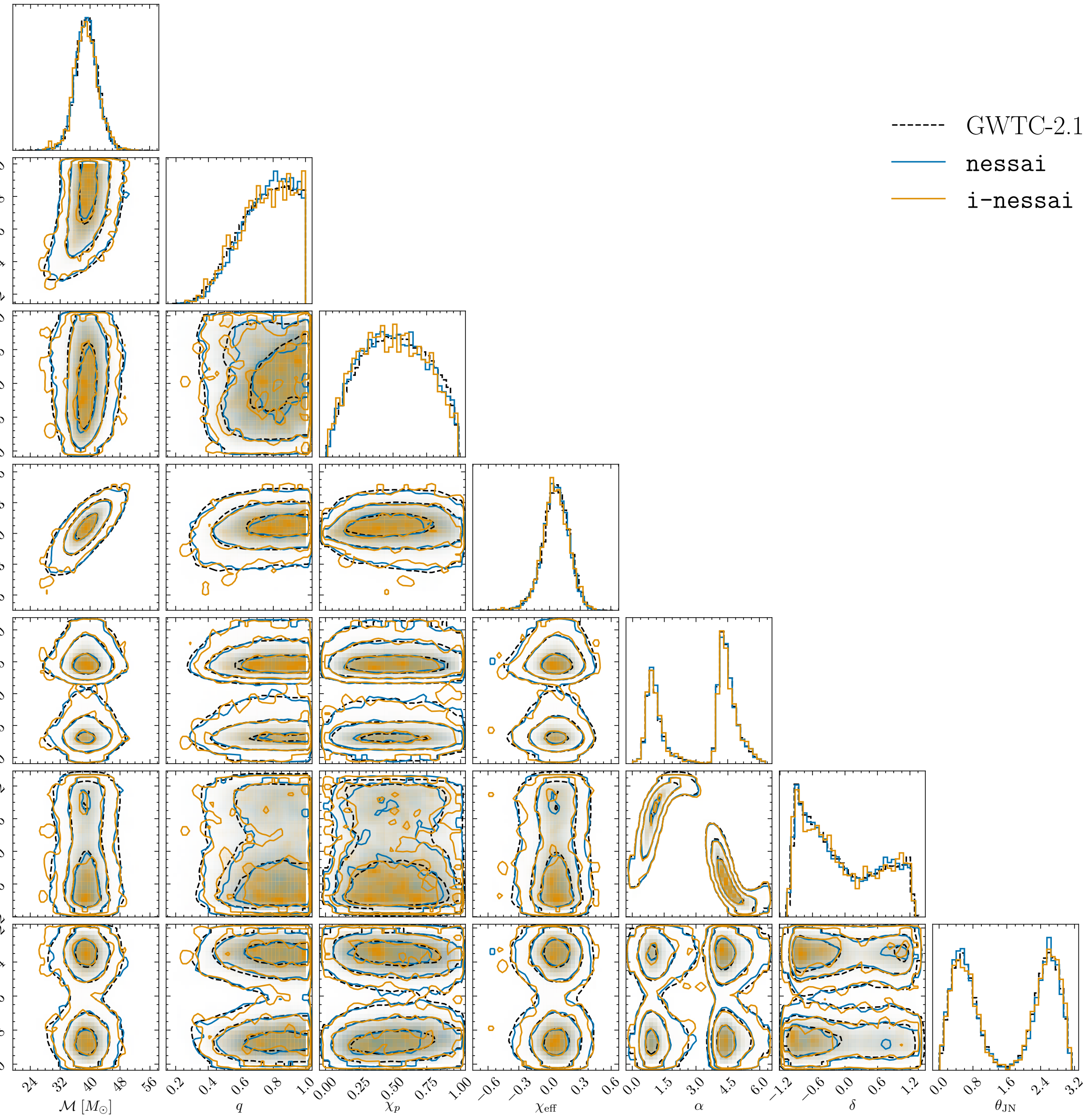


Figure G.4: Corner plot showing a subset of the posterior distributions for GW190823 from different analyses using `IMRPhenomXPHM`. Results are shown for GWTC-2.1 (black dashed line), the analysis with `nessai` described in section 8.2.1 (blue solid line) and the analysis with `i-nessai` described in appendix G.4 (orange solid line). The 2-dimensional contours shown contain 39.3%, 86.5% and 99.8% of the posterior probability respectively.

Software & Code

Software The research presented in this thesis made extensive use of the Python programming language and various open-source software packages, without which, this research would not have been possible. In particular, I wish to highlight the use of: bilby and bilby_pipe [198], corner [503], cpnest [202], dynesty [197], glasflow [502], LALSuite [26, 27], matplotlib [504], nflows [431], NumPy [505], pandas [506, 507], PyTorch [430], SciPy [508], and seaborn [509].

Code Code to reproduce all the results and figures presented in this thesis is available under the MIT licence at the following locations:

- nessai: main codebase: 10.5281/zenodo.4550693 [432] and documentation: https://nessai.readthedocs.io [429],
- nessai-models: codebase with likelihoods for testing nessai: 10.5281/zenodo.7105559 [439],
- code to reproduce the figures from chapters 1 to 3: https://github.com/mj-will/thesis-code [438],
- code to reproduce the results from chapter 4: https://github.com/mj-will/nessai-experiments [510],
- code to reproduce the results from chapters 5 and 6: https://github.com/mj-will/thesis-code [438],
- code to reproduce the results from chapter 7: 10.5281/zenodo.8124198 [453],
- code to reproduce the results from chapter 8: https://github.com/mj-will/nessai-gwtc-analysis [511].

Glossary

χ-distribution the probability distribution that describes the distribution of the square root of the sum of the squares of ν random normal variables, i.e $x = \sqrt{\sum_{i=1}^{\nu} y_i^2} \mid y_i \sim \mathcal{N}(0,1)$. ν is known as the degrees of freedom and the probability density function (PDF) is given by

$$\chi(x;\nu) \equiv p(x;\nu) = \frac{x^{\nu-1}e^{-\nu^2/2}}{\nu^{k/2-1}\Gamma(k/2)} \ ; \ x \geq 0, \tag{G.1}$$

where $\Gamma(k)$ is the gamma function. 85, 88, 101, 107, 126

γ-distribution a gamma distribution is parameterised by a shape parameter k and a scale parameter α. Its PDF is

$$p(x;k,\alpha) = \frac{1}{\Gamma(k)\alpha^k} x^{k-1} e^{-\frac{x}{\alpha}},$$

where $\Gamma(k)$ is the gamma function. 108, 221

active learning in machine learning, introducing more training samples at specific points in the parameter space that will have the largest benefit to the performance. 43

all-sky in gravitational-wave searches, a search that does not target specific (known) locations on the sky. 8

amortized In the context of simulation-based inference, the inference is amortized if, once the approximate likelihood (or posterior) has been learnt, performing inference does not require updating the approximate posterior. 43

batch size the number of data samples used to update the parameters of a neural network with stochastic gradient descent. 63, 222–224, 229

coloured noise Gaussian noise where the power spectral density (PSD) is frequency dependent. 16, 48

conditioner In the context of normalising flows, the function that returns the parameters that parameterise a transform. Typically a neural network. In this thesis it is denoted by $\Theta(\cdot)$ 68, 69, 222, 224

constant volume mode setting in the standard version of `nessai` that fixes the volume of the latent contour to a user-defined fraction $\rho_{\mathrm{CVM}} \in (0,1)$ for the duration of the sampling run. v, x, 104–110, 114–119, 135, 139, 203, 212

epoch when training a neural network, an epoch is a complete loop over all the batches of training data. 63, 222–224

equation of state equation that governs the pressure-density of matter; in the context of neutron stars this will determine their composition. xix, 22

evidence (or Bayesian evidence or marginalized likelihood) is the probability of some data $\boldsymbol{d}$ given a model H, $(p(\boldsymbol{d}|H))$. It is the denominator in Bayes' theorem eq. (2.10), is often denoted Z and is defined in eq. (2.11). xi, xxii, 29, 30, 32, 33, 49, 52, 56, 82, 92, 93, 98, 99, 101, 103, 105, 107, 110, 118, 132, 142–146, 148, 151, 152, 154–157, 163–166

expansion fraction (ϵ_{VF}) setting in the standard version of `nessai` that is used to set the fuzz factor for the latent radius (ϵ_{FF}) based on the number of dimensions in the normalising flow n_{flow}, $\epsilon_{\mathrm{FF}} = (1+\epsilon_{\mathrm{VF}})^{1/n_{flow}}$. 108

extreme mass ratio insprial system comprised of a massive black hole ($10^5\,\mathrm{M}_\odot$–$10^9\,\mathrm{M}_\odot$) and stellar-mass compact object ($\sim 1\,\mathrm{M}_\odot$) that will be observed by space-based gravitational-wave detectors [512]. 15

extrinsic parameters when modelling gravitational waves, extrinsic parameters determine how the detector will respond to the gravitational-wave signal. Examples include parameters that descrived the sky location and orientation of a source with respect to a reference point. xiv, 19, 23, 225, 241

galactic binary binary system where both components are white dwarfs and is located within the Milky Way [513]. 15

gamma-ray burst highly energetic short–duration emission from events such as supernovae . xx, 7, 217

Gaussian *See:* Gaussian distribution x, 83, 84, 112, 113

Gaussian distribution a Gaussian distribution, also known as a normal distribution, is parameterised by a mean μ and a standard deviation σ. Its PDF is

$$\mathcal{N}(x;\mu,\sigma) \equiv p(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$

When $\mu = 0$ and $\sigma = 1$ it is often referred to as a *unit* Gaussian. *See also:* multivariate Gaussian distribution. 49, 72, 108, 221, 249–251

Gaussian noise *See:* white noise 91

Gaussian process a stochastic process modelled by a multivariate Gaussian distribution, where the mean and covariance are functions rather than a vector and matrix. 55, 217

generative adversarial network a neural network that comprises of two components: a generator and a discriminator, that 'compete' against each other during training [341]. xx, 43, 66

GEO600 600 m Michelson interferometer located in Germany and operator by the GEO collaboration [69]. 14

GW150914 the first detected gravitational-wave event from a binary black hole (BBH) merger [6]. x–xiv, 22, 132, 139, 169, 171, 173, 175, 176, 185, 203–206, 208, 216, 240, 243, 244

GW170817 the first detection of gravitational waves from a binary neutron star (BNS) system and the first multi-messenger observation [29, 38]. xii, xiv, 22, 170, 188, 190–192, 194, 207, 215

GW190412 BBH system detected in O3a with support for asymmetric masses [472] 167, 191, 192, 195, 207, 208, 216

GW190425 BNS event detected during O3a with a total mass $\sim 3.4; \mathrm{M}_\odot$ [262] xii, 167, 191–195, 207, 208, 215

GW190521 heaviest binary system detected up to the end of O3 with a total mass $\sim 150\ \mathrm{M}_\odot$ [9, 10, 147] xii, 22, 167, 195–198, 207, 208, 216

GW191109 high-mass BBH event from O3b with total mass $\sim 112\mathrm{M}_\odot$ and support for negative χ_{eff} [10] xii, 167, 195, 199, 207, 208, 216

Hubble constant constant of proportionality that describes the rate of expansion of the universe usually denoted H_0 [514]. 22

information content in the context of nested sampling, the information (typically in bits or nats) contained in the posterior distribution compared to the prior. This is denoted $\mathcal{H}$ and is computed using the Kullback-Leibler divergence (KLD), see eq. (2.33). xxii, 37, 42, 93

intractable lack of a closed-form and/or an analytic solution. 29, 43, 67

intrinsic parameters when modelling gravitational waves, intrinsic parameters determine the evolution of the waveform as a function of time. Examples include the masses and spins of a binary system. xiv, 19, 20, 23, 225

Kerr spacetime Spacetime around a spinning axially symmetric black hole. 18

Kronecker delta a function defined for two positive integers that is 1 when the integers are equal and 0 otherwise,

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \tag{G.2}$$

4

long-wavelength limit the assumption that $(fL)/c \ll 1$, where f is the frequency of a gravitational wave, L is the length of the interferometer arm and c is the speed of light. 11

massive black hole binaries binary systems comprised of black holes with masses in the range $10^4\,\mathrm{M}_\odot$–$10^8\,\mathrm{M}_\odot$ and redshifts up to $z \sim 10$ that will be observed by space-based gravitational-wave detectors [515]. 15

metric (or distance function) is a non-negative function $f : D \times D \to \mathbb{R}$ that describes the 'distance' between points and, for a series of points series of points $x, y, z \in D$, must

- satisfy the triangle inequality $f(x, y) + f(y, z) \geq f(x, z)$,
- be symmetric so $f(x, y) = d(y, x)$,
- satisfy $f(x, x) = 0$.

28

multivariate Gaussian distribution generalization of the Gaussian distribution to n dimensions. It is parameterized by a mean $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ and its PDF is

$$p(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right\},$$

where $\mathbf{\Sigma}^{-1}$ is the inverse of the covariance matrix. 48, 223, 224, 250

neutron star a compact object formed from the collapse of supergiant stars with a typical mass $1\,\mathrm{M}_{\odot}$–$2\,\mathrm{M}_{\odot}$ and radius $\mathcal{O}(10\,\mathrm{km})$ [516]. 6, 8, 22, 249, 253

O1 the first LIGO Scientific Collaboration (LSC) observing run that ran from September 2015 to January 2016. ii, xi, xii, 22, 167, 170, 174, 178, 179, 183–186, 188, 190, 200, 205–207, 214–216, 241, 242

O2 the second LIGO-Virgo Collaboration (LVC) observing run that ran from Novemember 2016 until August 2017. ii, xi, xii, 22, 167, 170, 174, 178, 179, 183–186, 188, 190, 200, 205–207, 214, 215, 241, 242

O3 the third LIGO-Virgo-KAGRA Collaboration (LVK) observing run that started in April 2019 and ran until March 2020 with a one-month break in October 2019. ii, 22, 23, 167, 191, 195, 206, 207, 214–216, 250, 252

O3a the first half of the third LVK observing run (O3) that started in April 2019 and ran until October 2019. ix, 17, 190, 192, 250

O3b the second half of the third LVK observing run (O3) that started in November 2019 and ran until March 2020. 250

O4 the current LVK observing run, starting in May 2023. 22

overfitting when training a neural network, if the network learns characteristics specific to the training data rather than general characteristics, it is said to be overfitting. 65

P-P test *See:* probability-probability plot (P-P plot) 46, 96, 119, 130, 135, 137, 139, 140, 190, 209, 212–214, 217

primordial black hole black hole that formed in the early Universe with masses anywhere between $10^{-8}\,\mathrm{M}_{\odot}$ and $10^{4}\,\mathrm{M}_{\odot}$ and could also be one of the components of dark matter [517]. 15

probability-probability plot Probability-probability (P-P) plots are a visual method for assessing the similarity of two distributions by plotting their cumulative distributions against each other. The difference can be quantified using, for example, a Kolmogorov-Smirnov test (KS test) and a corresponding *p*-value can be computed; this constitutes a P-P test. See fig. 2.2 for example P-P plots. ix, xxi, 44, 74, 92, 114, 137, 158, 168, 211, 230, 234, 236, 252

proposal *See:* proposal distribution. 32

proposal distribution in sampling, a distribution with a known PDF from which samples can be drawn (or proposed) and, typically, accepted or rejeceted based on an acceptance criterion, see e.g. rejection sampling (section 2.4) or Markov Chain Monte Carlo (MCMC) (section 2.5). 30–32, 38, 41, 85, 143–148, 150, 152, 154, 160, 253

pulsar rapidly rotating neutron star that emit radios pulses at regular pulses. 7, 15

Rademacher distribution a discrete probability distribution with support $k \in \{-1, 1\}$ and the two outcomes have equal probability. The probability mass function (PMF) is

$$p(k) = \begin{cases} 0.5 & \text{if } k = -1, \\ 0.5 & \text{if } k = 1, \\ 0 & \text{if } k \notin \{-1, 1\}. \end{cases} \tag{G.3}$$

88

Rosenbrock *See:* appendix A.1. x, xii, 69, 110, 111, 119, 141, 156, 157, 168, 222, 237

state plot plot produced live by `nessai` that includes various diagnostics including the minimum and maximum likelihood, log-evidence, acceptance and other metrics that are useful when examining a run. x, 99

template bank a bank of gravitational-wave waveforms over which a detection statistic is maximized to detect gravitational waves. 49

traceless (tensor) a tensor where the sum of the diagonals elements (the trace) is zero, e.g.

$$h = \eta^{\mu\nu} h_{\mu\nu} = 0. \tag{G.4}$$

3

tractable has a closed form and/or an anlytic solution 67, 74, 76

Tukey window also known as a cosine-tapered window, is the convolution of a cosine lobe of width $(\alpha/2)N$ and a rectangular window of width $(1 - \alpha/2)N$, where N is the number of samples and $\alpha \in [0, 1]$ [518]. 51

uniform distribution a uniform distribution is parameterized by a minimum and a maximum (a, b). Its PDF is

$$\mathcal{U}(x; a, b) \equiv p(x; a, b) = \begin{cases} \frac{1}{b-a} & \text{if } x \in [a, b], \\ 0 & \text{if } x \notin [a, b]. \end{cases} \tag{G.5}$$

30, 31, 42, 108, 221

variational autoencoder a neural network where the input is compressed to a latent representation via an encoder and then reconstruted to match the original input via a decoder [227]. xxi, 43, 66, 90

well-behaved *See:* well-behaved function 33

well-behaved function a function that is single-valued, continuous and continuously differentiable. 34, 254

white dwarf compact object similar in size to the Earth that is formed after the death of solar mass stars. 249

white noise noise where the PSD is constant in frequency. 16, 250

Bibliography

[1] Michael J. Williams, John Veitch, and Chris Messenger. “Nested sampling with normalizing flows for gravitational-wave inference”. In: *Phys. Rev. D* 103.10 (2021), p. 103006. DOI: 10.1103/PhysRevD.103.103006. arXiv: 2102.11056 [gr-qc] (cit. on pp. xviii, 54, 81, 105, 115, 142, 145, 151, 153, 157, 162, 163, 166, 210, 229).

[2] Michael J. Williams, John Veitch, and Chris Messenger. “Importance nested sampling with normalising flows”. In: *Machine Learning: Science and Technology* (May 2023). DOI: 10.1088/2632-2153/acd5aa. arXiv: 2302.08526 [astro-ph.IM] (cit. on pp. xviii, 55, 141, 213).

[3] Albert Einstein. “Näherungsweise Integration der Feldgleichungen der Gravitation”. In: *Sitzungsberichte der Königlich PreuSSischen Akademie der Wissenschaften (Berlin* (Jan. 1916), pp. 688–696 (cit. on p. 2).

[4] R. A. Hulse and J. H. Taylor. “Discovery of a pulsar in a binary system.” In: *Astrophysical Journal* 195 (Jan. 1975), pp. L51–L53. DOI: 10.1086/181708 (cit. on p. 2).

[5] Nobel Prize Outreach. *The Nobel Prize in Physics 1993.* 2023. URL: https://www.nobelprize.org/prizes/physics/1993/press-release/ (visited on 02/06/2023) (cit. on p. 2).

[6] B.P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116.6 (2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. arXiv: 1602.03837 [gr-qc] (cit. on pp. 2, 7, 22, 54, 89, 250).

[7] B.P. Abbott et al. “GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs”. In: *Phys. Rev. X* 9.3 (2019), p. 031040. DOI: 10.1103/PhysRevX.9.031040. arXiv: 1811.12907 [astro-ph.HE] (cit. on pp. 2, 8, 19, 21, 22, 52, 54, 89, 170, 185, 188, 207, 216, 241, 243).

[8] R. Abbott et al. “GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run”. In: *Phys. Rev. X* 11 (2021), p. 021053. DOI: 10.1103/PhysRevX.11.021053. arXiv: 2010.14527 [gr-qc] (cit. on pp. 2, 8, 14, 16, 17, 19, 21, 22, 54, 89, 142, 157, 166).

[9] R. Abbott et al. “GWTC-2.1: Deep Extended Catalog of Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run”. In: (Aug. 2021). arXiv: 2108.01045 [gr-qc] (cit. on pp. 2, 8, 19, 22, 51, 53, 55, 92, 121, 142, 157, 166, 168–170, 191, 208, 211, 216, 241, 243, 245, 250).

[10] R. Abbott et al. “GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run”. In: (Nov. 2021). arXiv: 2111.03606 [gr-qc] (cit. on pp. 2, 8, 14, 19, 22, 51, 54, 55, 92, 121, 142, 157, 166, 168, 191, 195, 211, 250).

[11] Michele Maggiore. “31 The geometric approach to GWs”. In: *Gravitational Waves: Volume 1: Theory and Experiments*. Oxford University Press, Oct. 2007. ISBN: 9780198570745. DOI: 10.1093/acprof:oso/9780198570745.003.0001. eprint: https://academic.oup.com/book/0/chapter/353712130/chapter-pdf/43657914/acprof-9780198570745-chapter-1.pdf (cit. on pp. 3, 5).

[12] Jolien Creighton and Warren Anderson. *Gravitational-Wave Physics and Astronomy : An Introduction to Theory, Experiment and Data Analysis*. John Wiley & Sons, Ltd, 2011. ISBN: 9783527636037. DOI: https://doi.org/10.1002/9783527636037.ch6. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9783527636037 (cit. on pp. 3, 5, 10, 11, 13, 16, 49, 50).

[13] Bernard F Schutz and Franco Ricci. “Gravitational Waves, Sources, and Detectors”. In: (May 2010). arXiv: 1005.4735 [gr-qc] (cit. on p. 3).

[14] Maximiliano Isi. “Parametrizing gravitational-wave polarizations”. In: (Aug. 2022). arXiv: 2208.03372 [gr-qc] (cit. on pp. 4, 5).

[15] Maximiliano Isi. *Parametrizing gravitational-wave polarizations*. Version v1.0.0. Aug. 2022. DOI: 10.5281/zenodo.7016058 (cit. on p. 5).

[16] Curt Cutler and Kip S. Thorne. “An Overview of gravitational wave sources”. In: *16th International Conference on General Relativity and Gravitation (GR16)*. 2013, pp. 72–111. DOI: 10.1142/9789812776556_0004. arXiv: gr-qc/0204090 (cit. on pp. 6, 7, 9, 10).

[17] Pau Amaro-Seoane et al. “Laser Interferometer Space Antenna”. In: *arXiv e-prints*, arXiv:1702.00786 (Feb. 2017), arXiv:1702.00786. DOI: 10.48550/arXiv.1702.00786. arXiv: 1702.00786 [astro-ph.IM] (cit. on pp. 6, 15).

[18] Jun Luo et al. “TianQin: a space-borne gravitational wave detector”. In: *Class. Quant. Grav.* 33.3 (2016), p. 035010. DOI: 10.1088/0264-9381/33/3/035010. arXiv: 1512.02076 [astro-ph.IM] (cit. on pp. 6, 15).

[19] Benjamin P Abbott et al. “Model comparison from LIGO–Virgo data on GW170817’s binary components and consequences for the merger remnant”. In: *Class. Quant. Grav.* 37.4 (2020), p. 045006. DOI: 10.1088/1361-6382/ab5f7c. arXiv: 1908.01012 [gr-qc] (cit. on p. 6).

[20] B. P. Abbott et al. “Search for Post-merger Gravitational Waves from the Remnant of the Binary Neutron Star Merger GW170817”. In: *Astrophys. J. Lett.* 851.1 (2017), p. L16. DOI: 10.3847/2041-8213/aa9a35. arXiv: 1710.09320 [astro-ph.HE] (cit. on p. 6).

[21] Francesco Zappa et al. “Black-Hole Remnants from Black-HoleNeutron-Star Mergers”. In: *Phys. Rev. Lett.* 123.4 (2019), p. 041102. DOI: 10.1103/PhysRevLett.123.041102. arXiv: 1903.11622 [gr-qc] (cit. on p. 6).

[22] Zoheyr Doctor, Ben Farr, and Daniel E. Holz. “Black Hole Leftovers: The Remnant Population from Binary Black Hole Mergers”. In: *Astrophys. J. Lett.* 914.1 (2021), p. L18. DOI: 10.3847/2041-8213/ac0334. arXiv: 2103.04001 [astro-ph.HE] (cit. on p. 6).

[23] Sascha Husa et al. “Frequency-domain gravitational waves from nonprecessing black-hole binaries. I. New numerical waveforms and anatomy of the signal”. In: *Physical Review D* 93.4, 044006 (Feb. 2016), p. 044006. DOI: 10.1103/PhysRevD.93.044006. arXiv: 1508.07250 [gr-qc] (cit. on p. 7).

[24] Sebastian Khan et al. “Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era”. In: *Physical Review D* 93.4, 044007 (Feb. 2016), p. 044007. DOI: 10.1103/PhysRevD.93.044007. arXiv: 1508.07253 [gr-qc] (cit. on p. 7).

[25] Alex Nitz et al. *gwastro/pycbc*. Mar. 2023. DOI: 10.5281/zenodo.7692098 (cit. on pp. 7, 55, 78).

[26] LIGO Scientific Collaboration. *LIGO Algorithm Library - LALSuite*. free software (GPL). 2018. DOI: 10.7935/GT1W-FZ16 (cit. on pp. 7, 247).

[27] Karl Wette. “SWIGLAL: Python and Octave interfaces to the LALSuite gravitational-wave data analysis libraries”. In: *SoftwareX* 12 (2020), p. 100634. DOI: 10.1016/j.softx.2020.100634 (cit. on pp. 7, 247).

[28] Curt Cutler and Eanna E. Flanagan. “Gravitational waves from merging compact binaries: How accurately can one extract the binary’s parameters from the inspiral wave form?” In: *Phys. Rev. D* 49 (1994), pp. 2658–2697. DOI: 10.1103/PhysRevD.49.2658. arXiv: gr-qc/9402014 (cit. on pp. 6, 7).

[29] B.P. Abbott et al. "GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral". In: *Phys. Rev. Lett.* 119.16 (2017), p. 161101. DOI: 10.1103/PhysRevLett.119.161101. arXiv: 1710.05832 [gr-qc] (cit. on pp. 7, 22, 51, 54, 170, 188, 250).

[30] R. Abbott et al. "Observation of Gravitational Waves from Two Neutron StarBlack Hole Coalescences". In: *Astrophys. J. Lett.* 915.1 (2021), p. L5. DOI: 10.3847/2041-8213/ac082e. arXiv: 2106.15163 [astro-ph.HE] (cit. on pp. 7, 55).

[31] Benjamin P. Abbott et al. "All-sky search for short gravitational-wave bursts in the first Advanced LIGO run". In: *Phys. Rev. D* 95.4 (2017), p. 042003. DOI: 10.1103/PhysRevD.95.042003. arXiv: 1611.02972 [gr-qc] (cit. on p. 7).

[32] B.P. Abbott et al. "All-Sky Search for Short Gravitational-Wave Bursts in the Second Advanced LIGO and Advanced Virgo Run". In: *Phys. Rev. D* 100.2 (2019), p. 024017. DOI: 10.1103/PhysRevD.100.024017. arXiv: 1905.03457 [gr-qc] (cit. on pp. 7, 8).

[33] R. Abbott et al. "All-sky search for short gravitational-wave bursts in the third Advanced LIGO and Advanced Virgo run". In: *Phys. Rev. D* 104.12 (2021), p. 122004. DOI: 10.1103/PhysRevD.104.122004. arXiv: 2107.03701 [gr-qc] (cit. on pp. 7, 8).

[34] Eric Thrane et al. "Long gravitational-wave transients and associated detection strategies for a network of terrestrial interferometers". In: *Phys. Rev. D* 83 (2011), p. 083004. DOI: 10.1103/PhysRevD.83.083004. arXiv: 1012.2150 [astro-ph.IM] (cit. on p. 7).

[35] R. Abbott et al. "Search for Gravitational Waves Associated with Gamma-Ray Bursts Detected by Fermi and Swift during the LIGOVirgo Run O3b". In: *Astrophys. J.* 928.2 (2022), p. 186. DOI: 10.3847/1538-4357/ac532b. arXiv: 2111.03608 [astro-ph.HE] (cit. on pp. 7, 8, 22).

[36] B.P. Abbott et al. "Optically targeted search for gravitational waves emitted by core-collapse supernovae during the first and second observing runs of advanced LIGO and advanced Virgo". In: *Phys. Rev. D* 101.8 (2020), p. 084002. DOI: 10.1103/PhysRevD.101.084002. arXiv: 1908.03584 [astro-ph.HE] (cit. on p. 7).

[37] R. Abbott et al. "Constraints on cosmic strings using data from the third Advanced LIGO-Virgo observing run". In: *Phys. Rev. Lett.* 126 (2021), p. 241102. DOI: 10.1103/PhysRevLett.126.241102. arXiv: 2101.12248 [gr-qc] (cit. on pp. 7, 8).

[38] B.P. Abbott et al. “Gravitational Waves and Gamma-rays from a Binary Neutron Star Merger: GW170817 and GRB 170817A”. In: *Astrophys. J. Lett.* 848.2 (2017), p. L13. DOI: 10.3847/2041-8213/aa920c. arXiv: 1710.05834 [astro-ph.HE] (cit. on pp. 7, 22, 188, 250).

[39] R. Mochkovitch et al. “Gamma-ray bursts as collimated jets from neutron star/black hole mergers”. In: *Nature* 361.6409 (Jan. 1993), pp. 236–238. DOI: 10.1038/361236a0 (cit. on p. 7).

[40] M. Drago et al. “Coherent WaveBurst, a pipeline for unmodeled gravitational-wave data analysis”. In: (June 2020). arXiv: 2006.12604 [gr-qc] (cit. on p. 7).

[41] R. Abbott et al. “Search for Gravitational Waves Associated with Gamma-Ray Bursts Detected by Fermi and Swift During the LIGO-Virgo Run O3a”. In: *Astrophys. J.* 915.2 (2021), p. 86. DOI: 10.3847/1538-4357/abee15. arXiv: 2010.14550 [astro-ph.HE] (cit. on pp. 8, 22).

[42] J. Abadie et al. “All-sky search for gravitational-wave bursts in the second joint LIGO-Virgo run”. In: *Phys. Rev. D* 85 (2012), p. 122007. DOI: 10.1103/PhysRevD.85.122007. arXiv: 1202.2788 [gr-qc] (cit. on p. 8).

[43] R. Abbott et al. “All-sky search for long-duration gravitational-wave bursts in the third Advanced LIGO and Advanced Virgo run”. In: *Phys. Rev. D* 104.10 (2021), p. 102001. DOI: 10.1103/PhysRevD.104.102001. arXiv: 2107.13796 [gr-qc] (cit. on p. 8).

[44] Keith Riles. “Searches for continuous-wave gravitational radiation”. In: *Living Rev. Rel.* 26.1 (2023), p. 3. DOI: 10.1007/s41114-023-00044-3. arXiv: 2206.06447 [astro-ph.HE] (cit. on pp. 8, 78).

[45] Asimina Arvanitaki, Masha Baryakhtar, and Xinlu Huang. “Discovering the QCD Axion with Black Holes and Gravitational Waves”. In: *Phys. Rev. D* 91.8 (2015), p. 084011. DOI: 10.1103/PhysRevD.91.084011. arXiv: 1411.2263 [hep-ph] (cit. on p. 8).

[46] R. Abbott et al. “Gravitational-wave constraints on the equatorial ellipticity of millisecond pulsars”. In: *Astrophys. J. Lett.* 902 (2020), p. L21. DOI: 10.3847/2041-8213/abb655. arXiv: 2007.14251 [astro-ph.HE] (cit. on p. 8).

[47] R. Abbott et al. “Search for continuous gravitational waves from 20 accreting millisecond X-ray pulsars in O3 LIGO data”. In: *Phys. Rev. D* 105.2 (2021), p. 022002. DOI: 10.1103/PhysRevD.105.022002. arXiv: 2109.09255 [astro-ph.HE] (cit. on p. 8).

[48] J. Aasi et al. “Directed search for continuous gravitational waves from the Galactic center”. In: *Phys. Rev. D* 88.10 (2013), p. 102002. DOI: 10.1103/PhysRevD.88.102002. arXiv: 1309.6221 [gr-qc] (cit. on p. 8).

[49] J. Aasi et al. “Directed search for gravitational waves from Scorpius X-1 with initial LIGO data”. In: *Phys. Rev. D* 91.6 (2015), p. 062008. DOI: 10.1103/PhysRevD.91.062008. arXiv: 1412.0605 [gr-qc] (cit. on p. 8).

[50] R. Abbott et al. “Search for continuous gravitational wave emission from the Milky Way center in O3 LIGO-Virgo data”. In: *Phys. Rev. D* 106.4 (2022), p. 042003. DOI: 10.1103/PhysRevD.106.042003. arXiv: 2204.04523 [astro-ph.HE] (cit. on p. 8).

[51] B. Abbott et al. “First all-sky upper limits from LIGO on the strength of periodic gravitational waves using the Hough transform”. In: *Phys. Rev. D* 72 (2005), p. 102004. DOI: 10.1103/PhysRevD.72.102004. arXiv: gr-qc/0508065 (cit. on p. 8).

[52] B.P. Abbott et al. “All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO O2 data”. In: *Phys. Rev. D* 100.2 (2019), p. 024004. DOI: 10.1103/PhysRevD.100.024004. arXiv: 1903.01901 [astro-ph.HE] (cit. on p. 8).

[53] R. Abbott et al. “All-sky search for continuous gravitational waves from isolated neutron stars in the early O3 LIGO data”. In: *Phys. Rev. D* 104.8 (2021), p. 082004. DOI: 10.1103/PhysRevD.104.082004. arXiv: 2107.00600 [gr-qc] (cit. on p. 8).

[54] R. Abbott et al. “All-sky, all-frequency directional search for persistent gravitational waves from Advanced LIGO’s and Advanced Virgo’s first three observing runs”. In: *Phys. Rev. D* 105.12 (2022), p. 122001. DOI: 10.1103/PhysRevD.105.122001. arXiv: 2110.09834 [gr-qc] (cit. on p. 8).

[55] R. Abbott et al. “All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO and Advanced Virgo O3 data”. In: *Phys. Rev. D* 106.10 (2022), p. 102008. DOI: 10.1103/PhysRevD.106.102008. arXiv: 2201.00697 [gr-qc] (cit. on p. 8).

[56] P. A. R. Ade et al. “Planck 2015 results. XIII. Cosmological parameters”. In: *Astron. Astrophys.* 594 (2016), A13. DOI: 10.1051/0004-6361/201525830. arXiv: 1502.01589 [astro-ph.CO] (cit. on p. 8).

[57] Nelson Christensen. “Measuring the stochastic gravitational-radiation background with laser-interferometric antennas”. In: *Phys. Rev. D* 46 (12 Dec. 1992), pp. 5250–5266. DOI: 10.1103/PhysRevD.46.5250 (cit. on p. 8).

[58] Nelson Christensen. “Stochastic Gravitational Wave Backgrounds”. In: *Rept. Prog. Phys.* 82.1 (2019), p. 016903. DOI: 10.1088/1361-6633/aae6b5. arXiv: 1811.08797 [gr-qc] (cit. on pp. 8, 9).

[59] Tania Regimbau. “The astrophysical gravitational wave stochastic background”. In: *Res. Astron. Astrophys.* 11 (2011), pp. 369–390. DOI: 10.1088/1674-4527/11/4/001. arXiv: 1101.2762 [astro-ph.CO] (cit. on p. 8).

[60] B. Abbott et al. “Searching for a Stochastic Background of Gravitational Waves with LIGO”. In: *Astrophys. J.* 659 (2007), pp. 918–930. DOI: 10.1086/511329. arXiv: astro-ph/0608606 (cit. on p. 9).

[61] Benjamin P. Abbott et al. “Upper Limits on the Stochastic Gravitational-Wave Background from Advanced LIGO’s First Observing Run”. In: *Phys. Rev. Lett.* 118.12 (2017). [Erratum: Phys.Rev.Lett. 119, 029901 (2017)], p. 121101. DOI: 10.1103/PhysRevLett.118.121101. arXiv: 1612.02029 [gr-qc] (cit. on p. 9).

[62] B.P. Abbott et al. “Search for the isotropic stochastic background using data from Advanced LIGO’s second observing run”. In: *Phys. Rev. D* 100.6 (2019), p. 061101. DOI: 10.1103/PhysRevD.100.061101. arXiv: 1903.02886 [gr-qc] (cit. on p. 9).

[63] B. Abbott et al. “Upper limits on a stochastic background of gravitational waves”. In: *Phys. Rev. Lett.* 95 (2005), p. 221101. DOI: 10.1103/PhysRevLett.95.221101. arXiv: astro-ph/0507254 (cit. on p. 9).

[64] J. Abadie et al. “Directional limits on persistent gravitational waves using LIGO S5 science data”. In: *Phys. Rev. Lett.* 107 (2011), p. 271102. DOI: 10.1103/PhysRevLett.107.271102. arXiv: 1109.1809 [astro-ph.CO] (cit. on p. 9).

[65] Benjamin P. Abbott et al. “Directional Limits on Persistent Gravitational Waves from Advanced LIGO’s First Observing Run”. In: *Phys. Rev. Lett.* 118.12 (2017), p. 121102. DOI: 10.1103/PhysRevLett.118.121102. arXiv: 1612.02030 [gr-qc] (cit. on p. 9).

[66] B.P. Abbott et al. “Directional limits on persistent gravitational waves using data from Advanced LIGO’s first two observing runs”. In: *Phys. Rev. D* 100.6 (2019), p. 062001. DOI: 10.1103/PhysRevD.100.062001. arXiv: 1903.08844 [gr-qc] (cit. on p. 9).

[67] James L. Levine. “Early Gravity-Wave Detection Experiments, 1960-1975”. In: *Physics in Perspective* 6.1 (Jan. 2004), pp. 42–75. DOI: 10.1007/s00016-003-0179-6 (cit. on p. 9).

[68] Odylio Denys Aguiar. “The Past, Present and Future of the Resonant-Mass Gravitational Wave Detectors”. In: *Res. Astron. Astrophys.* 11 (2011), pp. 1–42. DOI: 10.1088/1674-4527/11/1/001. arXiv: 1009.1138 [astro-ph.IM] (cit. on p. 9).

[69] Harald Luck et al. "The upgrade of GEO600". In: *J. Phys. Conf. Ser.* 228 (2010). Ed. by Zsuzsa Marka and Szabolcs Marka, p. 012012. DOI: 10.1088/1742-6596/228/1/012012. arXiv: 1004.0339 [gr-qc] (cit. on pp. 9, 14, 250).

[70] J. Aasi et al. "Advanced LIGO". In: *Class. Quant. Grav.* 32 (2015), p. 074001. DOI: 10.1088/0264-9381/32/7/074001. arXiv: 1411.4547 [gr-qc] (cit. on pp. 9–11, 13, 91, 121, 135, 159).

[71] F. Acernese et al. "Advanced Virgo: a second-generation interferometric gravitational wave detector". In: *Class. Quant. Grav.* 32.2 (2015), p. 024001. DOI: 10.1088/0264-9381/32/2/024001. arXiv: 1408.3978 [gr-qc] (cit. on pp. 9, 10, 142).

[72] Yoichi Aso et al. "Interferometer design of the KAGRA gravitational wave detector". In: *Phys. Rev. D* 88 (4 Aug. 2013), p. 043007. DOI: 10.1103/PhysRevD.88.043007 (cit. on pp. 9, 10, 14, 142).

[73] B. P. Abbott et al. "LIGO: The Laser interferometer gravitational-wave observatory". In: *Rept. Prog. Phys.* 72 (2009), p. 076901. DOI: 10.1088/0034-4885/72/7/076901. arXiv: 0711.3041 [gr-qc] (cit. on pp. 10, 142).

[74] B.P. Abbott et al. "Prospects for Observing and Localizing Gravitational-Wave Transients with Advanced LIGO, Advanced Virgo and KAGRA". In: *Living Rev. Rel.* 21.1 (2018), p. 3. DOI: 10.1007/s41114-018-0012-9. arXiv: 1304.0670 [gr-qc] (cit. on pp. 10, 14, 22, 23, 50, 82, 121, 142, 219).

[75] Bernard F. Schutz. "Networks of gravitational wave detectors and three figures of merit". In: *Classical and Quantum Gravity* 28.12, 125023 (June 2011), p. 125023. DOI: 10.1088/0264-9381/28/12/125023. arXiv: 1102.5421 [astro-ph.IM] (cit. on p. 13).

[76] Caltech/MIT/LIGO Lab. *Diagram of a basic interferometer design.* [Online; accessed February 2023]. 2023. URL: %5Curl%7Bhttps://www.ligo.org/science/GW-Overview/images/IFO.jpg%7D (cit. on p. 14).

[77] F. Acernese et al. "Advanced Virgo: a second-generation interferometric gravitational wave detector". In: *Class. Quant. Grav.* 32.2 (2015), p. 024001. DOI: 10.1088/0264-9381/32/2/024001. arXiv: 1408.3978 [gr-qc] (cit. on pp. 14, 91, 121, 135).

[78] T. Accadia et al. "Virgo: a laser interferometer to detect gravitational waves". In: *JINST* 7 (2012), P03012. DOI: 10.1088/1748-0221/7/03/P03012 (cit. on p. 14).

[79] H. Abe et al. "Performance of the KAGRA detector during the first joint observation with GEO 600 (O3GK)". In: (Mar. 2022). DOI: 10.1093/ptep/ptac093. arXiv: 2203.07011 [astro-ph.IM] (cit. on p. 14).

[80] F. Acernese et al. “Virgo Detector Characterization and Data Quality: results from the O3 run”. In: (Oct. 2022). arXiv: 2210.15633 [gr-qc] (cit. on p. 14).

[81] Michele Maggiore et al. “Science Case for the Einstein Telescope”. In: *JCAP* 03 (2020), p. 050. DOI: 10.1088/1475-7516/2020/03/050. arXiv: 1912.02622 [astro-ph.CO] (cit. on pp. 15, 203).

[82] David Reitze et al. “Cosmic Explorer: The U.S. Contribution to Gravitational-Wave Astronomy beyond LIGO”. In: *Bull. Am. Astron. Soc.* 51.7 (2019), p. 035. arXiv: 1907.04833 [astro-ph.IM] (cit. on p. 15).

[83] G Hobbs et al. “The International Pulsar Timing Array project: using pulsars as a gravitational wave detector”. In: *Classical and Quantum Gravity* 27.8 (Apr. 2010), p. 084013. DOI: 10.1088/0264-9381/27/8/084013 (cit. on p. 15).

[84] George Hobbs and Shi Dai. “Gravitational wave research using pulsar timing arrays”. In: *Natl. Sci. Rev.* 4.5 (2017), pp. 707–717. DOI: 10.1093/nsr/nwx126. arXiv: 1707.01615 [astro-ph.IM] (cit. on p. 15).

[85] Alberto Sesana and Alberto Vecchio. “Measuring the parameters of massive black hole binary systems with Pulsar Timing Array observations of gravitational waves”. In: *Phys. Rev. D* 81 (2010), p. 104008. DOI: 10.1103/PhysRevD.81.104008. arXiv: 1003.0677 [astro-ph.CO] (cit. on p. 15).

[86] F. A. Jenet et al. “Upper bounds on the low-frequency stochastic gravitational wave background from pulsar timing observations: Current limits and future prospects”. In: *Astrophys. J.* 653 (2006), pp. 1571–1576. DOI: 10.1086/508702. arXiv: astro-ph/0609013 (cit. on p. 15).

[87] Andrea N. Lommen. “Pulsar timing arrays: the promise of gravitational wave detection”. In: *Rept. Prog. Phys.* 78.12 (2015), p. 124901. DOI: 10.1088/0034-4885/78/12/124901 (cit. on p. 15).

[88] Travis Robson, Neil J. Cornish, and Chang Liu. “The construction and use of LISA sensitivity curves”. In: *Class. Quant. Grav.* 36.10 (2019), p. 105011. DOI: 10.1088/1361-6382/ab1101. arXiv: 1803.01944 [astro-ph.HE] (cit. on p. 15).

[89] Stanislav Babak, Antoine Petiteau, and Martin Hewitson. “LISA Sensitivity and SNR Calculations”. In: (Aug. 2021). arXiv: 2108.01167 [astro-ph.IM] (cit. on p. 15).

[90] Jianwei Mei et al. “The TianQin project: current progress on science and technology”. In: *PTEP* 2021.5 (2021), 05A107. DOI: 10.1093/ptep/ptaa114. arXiv: 2008.10332 [gr-qc] (cit. on p. 15).

[91] S. Kawamura et al. "The Japanese space gravitational wave antenna DECIGO". In: *Class. Quant. Grav.* 23 (2006). Ed. by N. Mio, S125–S132. DOI: 10.1088/0264-9381/23/8/S17 (cit. on p. 15).

[92] Seiji Kawamura et al. "Current status of space gravitational wave antenna DECIGO and B-DECIGO". In: *PTEP* 2021.5 (2021), 05A105. DOI: 10.1093/ptep/ptab019. arXiv: 2006.13545 [gr-qc] (cit. on p. 15).

[93] Pau Amaro-Seoane et al. "Astrophysics with the Laser Interferometer Space Antenna". In: *Living Rev. Rel.* 26.1 (2023), p. 2. DOI: 10.1007/s41114-022-00041-y. arXiv: 2203.06016 [gr-qc] (cit. on pp. 15, 217).

[94] J. Aasi et al. "Parameter estimation for compact binary coalescence signals with the first generation gravitational-wave detector network". In: *Phys. Rev. D* 88 (2013), p. 062001. DOI: 10.1103/PhysRevD.88.062001. arXiv: 1304.1775 [gr-qc] (cit. on pp. 16, 51, 168).

[95] Neil J. Cornish and Tyson B. Littenberg. "BayesWave: Bayesian Inference for Gravitational Wave Bursts and Instrument Glitches". In: *Class. Quant. Grav.* 32.13 (2015), p. 135012. DOI: 10.1088/0264-9381/32/13/135012. arXiv: 1410.3835 [gr-qc] (cit. on pp. 16, 51).

[96] J. Veitch et al. "Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library". In: *Phys. Rev. D* 91.4 (2015), p. 042003. DOI: 10.1103/PhysRevD.91.042003. arXiv: 1409.7215 [gr-qc] (cit. on pp. 17, 21, 23, 33, 37, 38, 49–54, 57, 79, 82, 87, 89, 119, 121, 127, 142, 168, 170, 218).

[97] R. A. Breuer et al. "Some properties of spin-weighted spheroidal harmonics". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 358.1692 (1977), pp. 71–86. DOI: 10.1098/rspa.1977.0187. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1977.0187 (cit. on p. 17).

[98] Cecilio García-Quirós et al. "Multimode frequency-domain model for the gravitational wave signal from nonprecessing black-hole binaries". In: *Phys. Rev. D* 102.6 (2020), p. 064002. DOI: 10.1103/PhysRevD.102.064002. arXiv: 2001.10914 [gr-qc] (cit. on pp. 18, 121).

[99] Antoni Ramos-Buades et al. "SEOBNRv5PHM: Next generation of accurate and efficient multipolar precessing-spin effective-one-body waveforms for binary black holes". In: (Mar. 2023). arXiv: 2303.18046 [gr-qc] (cit. on pp. 18, 19, 82).

[100] Patricia Schmidt. “Gravitational Waves From Binary Black Hole Mergers: Modeling and Observations”. In: *Front. Astron. Space Sci.* 7 (2020), p. 28. DOI: 10.3389/fspas.2020.00028 (cit. on p. 18).

[101] Luc Blanchet. “Gravitational Radiation from Post-Newtonian Sources and Inspiralling Compact Binaries”. In: *Living Rev. Rel.* 17 (2014), p. 2. DOI: 10.12942/lrr-2014-2. arXiv: 1310.1528 [gr-qc] (cit. on p. 18).

[102] Masaru Shibata and Koji Uryu. “Simulation of merging binary neutron stars in full general relativity: Gamma = two case”. In: *Phys. Rev. D* 61 (2000), p. 064001. DOI: 10.1103/PhysRevD.61.064001. arXiv: gr-qc/9911058 (cit. on p. 18).

[103] Matthew Anderson et al. “Simulating binary neutron stars: Dynamics and gravitational waves”. In: *Phys. Rev. D* 77 (2008), p. 024006. DOI: 10.1103/PhysRevD.77.024006. arXiv: 0708.2720 [gr-qc] (cit. on p. 18).

[104] Frans Pretorius. “Evolution of binary black hole spacetimes”. In: *Phys. Rev. Lett.* 95 (2005), p. 121101. DOI: 10.1103/PhysRevLett.95.121101. arXiv: gr-qc/0507014 (cit. on p. 18).

[105] Manuela Campanelli, C. O. Lousto, and Y. Zlochower. “Spinning-black-hole binaries: The orbital hang up”. In: *Phys. Rev. D* 74 (2006), p. 041501. DOI: 10.1103/PhysRevD.74.041501. arXiv: gr-qc/0604012 (cit. on p. 18).

[106] John G. Baker et al. “Consistency of post-Newtonian waveforms with numerical relativity”. In: *Phys. Rev. Lett.* 99 (2007), p. 181101. DOI: 10.1103/PhysRevLett.99.181101. arXiv: gr-qc/0612024 (cit. on p. 18).

[107] L. Santamaria et al. “Matching post-Newtonian and numerical relativity waveforms: systematic errors and a new phenomenological model for non-precessing black hole binaries”. In: *Phys. Rev. D* 82 (2010), p. 064016. DOI: 10.1103/PhysRevD.82.064016. arXiv: 1005.3306 [gr-qc] (cit. on p. 18).

[108] Adam G. M. Lewis, Aaron Zimmerman, and Harald P. Pfeiffer. “Fundamental frequencies and resonances from eccentric and precessing binary black hole inspirals”. In: *Class. Quant. Grav.* 34.12 (2017), p. 124001. DOI: 10.1088/1361-6382/aa66f4. arXiv: 1611.03418 [gr-qc] (cit. on p. 18).

[109] Tim Dietrich et al. “High-resolution numerical relativity simulations of spinning binary neutron star mergers”. In: *26th Euromicro International Conference on Parallel, Distributed and Network-based Processing.* 2018, pp. 682–689. DOI: 10.1109/PDP2018.2018.00113. arXiv: 1803.07965 [gr-qc] (cit. on p. 18).

[110] Antoni Ramos-Buades et al. “First survey of spinning eccentric black hole mergers: Numerical relativity simulations, hybrid waveforms, and parameter estimation”. In: *Phys. Rev. D* 101.8 (2020), p. 083015. DOI: 10.1103/PhysRevD.101.083015. arXiv: 1909.11011 [gr-qc] (cit. on p. 18).

[111] A. Buonanno and T. Damour. “Effective one-body approach to general relativistic two-body dynamics”. In: *Phys. Rev. D* 59 (1999), p. 084006. DOI: 10.1103/PhysRevD.59.084006. arXiv: gr-qc/9811091 (cit. on p. 18).

[112] Alessandra Buonanno and Thibault Damour. “Transition from inspiral to plunge in binary black hole coalescences”. In: *Phys. Rev. D* 62 (2000), p. 064015. DOI: 10.1103/PhysRevD.62.064015. arXiv: gr-qc/0001013 (cit. on p. 18).

[113] Yi Pan et al. “Inspiral-merger-ringdown waveforms of spinning, precessing black-hole binaries in the effective-one-body formalism”. In: *Phys. Rev. D* 89.8 (2014), p. 084006. DOI: 10.1103/PhysRevD.89.084006. arXiv: 1307.6232 [gr-qc] (cit. on p. 18).

[114] Roberto Cotesta et al. “Enriching the Symphony of Gravitational Waves from Binary Black Holes by Tuning Higher Harmonics”. In: *Phys. Rev. D* 98.8 (2018), p. 084028. DOI: 10.1103/PhysRevD.98.084028. arXiv: 1803.10701 [gr-qc] (cit. on p. 18).

[115] Alessandro Nagar et al. “Multipolar effective one body model for nonspinning black hole binaries”. In: *Phys. Rev. D* 101.2 (2020), p. 024041. DOI: 10.1103/PhysRevD.101.024041. arXiv: 1904.09550 [gr-qc] (cit. on p. 18).

[116] Serguei Ossokine et al. “Multipolar Effective-One-Body Waveforms for Precessing Binary Black Holes: Construction and Validation”. In: *Phys. Rev. D* 102.4 (2020), p. 044055. DOI: 10.1103/PhysRevD.102.044055. arXiv: 2004.09442 [gr-qc] (cit. on pp. 18, 55).

[117] Parameswaran Ajith et al. “Phenomenological template family for black-hole coalescence waveforms”. In: *Class. Quant. Grav.* 24 (2007). Ed. by B. Krishnan, M. A. Papa, and Bernard F. Schutz, S689–S700. DOI: 10.1088/0264-9381/24/19/S31. arXiv: 0704.3764 [gr-qc] (cit. on p. 18).

[118] Geraint Pratten et al. “Setting the cornerstone for a family of models for gravitational waves from compact binaries: The dominant harmonic for nonprecessing quasicircular black holes”. In: *Phys. Rev. D* 102.6 (2020), p. 064001. DOI: 10.1103/PhysRevD.102.064001. arXiv: 2001.11412 [gr-qc] (cit. on pp. 18, 121, 170, 203).

[119] Sebastian Khan et al. “Phenomenological model for the gravitational-wave signal from precessing binary black holes with two-spin effects”. In: *Phys. Rev. D* 100.2 (2019), p. 024059. DOI: 10.1103/PhysRevD.100.024059. arXiv: 1809.10113 [gr-qc] (cit. on pp. 18, 91).

[120] Geraint Pratten et al. “Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes”. In: *Phys. Rev. D* 103.10 (2021), p. 104056. DOI: 10.1103/PhysRevD.103.104056. arXiv: 2004.06503 [gr-qc] (cit. on pp. 18, 82, 175, 203).

[121] Lorenzo Pompili et al. “Laying the foundation of the effective-one-body waveform models SEOBNRv5: improved accuracy and efficiency for spinning non-precessing binary black holes”. In: (Mar. 2023). arXiv: 2303.18039 [gr-qc] (cit. on p. 19).

[122] I. M. Romero-Shaw et al. “Bayesian inference for compact binary coalescences with bilby: validation and application to the first LIGOVirgo gravitational-wave transient catalogue”. In: *Mon. Not. Roy. Astron. Soc.* 499.3 (2020), pp. 3295–3319. DOI: 10.1093/mnras/staa2850. arXiv: 2006.00714 [astro-ph.IM] (cit. on pp. 19, 47, 49, 52, 54, 55, 57, 82, 90–92, 101, 157, 160, 168, 188, 230, 239).

[123] Benjamin Farr et al. “A more effective coordinate system for parameter estimation of precessing compact binaries from gravitational waves”. In: *Phys. Rev. D* 90 (2 July 2014), p. 024018. DOI: 10.1103/PhysRevD.90.024018 (cit. on pp. 20, 21, 50, 87).

[124] Thibault Damour. “Coalescence of two spinning black holes: an effective one-body approach”. In: *Phys. Rev. D* 64 (2001), p. 124013. DOI: 10.1103/PhysRevD.64.124013. arXiv: gr-qc/0103018 (cit. on pp. 21, 226).

[125] Etienne Racine. “Analysis of spin precession in binary black hole systems including quadrupole-monopole interaction”. In: *Phys. Rev. D* 78 (2008), p. 044021. DOI: 10.1103/PhysRevD.78.044021. arXiv: 0803.1820 [gr-qc] (cit. on pp. 21, 226).

[126] Mark Hannam et al. “Simple Model of Complete Precessing Black-Hole-Binary Gravitational Waveforms”. In: *Phys. Rev. Lett.* 113.15 (2014), p. 151101. DOI: 10.1103/PhysRevLett.113.151101. arXiv: 1308.3271 [gr-qc] (cit. on pp. 21, 159, 226).

[127] Patricia Schmidt, Frank Ohme, and Mark Hannam. “Towards models of gravitational waveforms from generic binaries II: Modelling precession effects with a single effective precession parameter”. In: *Phys. Rev. D* 91.2 (2015), p. 024043. DOI: 10.1103/PhysRevD.91.024043. arXiv: 1408.1810 [gr-qc] (cit. on pp. 21, 226).

[128] Davide Gerosa et al. “A generalized precession parameter χ_p to interpret gravitational-wave data”. In: *Phys. Rev. D* 103.6 (2021), p. 064067. DOI: 10.1103/PhysRevD.103.064067. arXiv: 2011.11948 [gr-qc] (cit. on p. 21).

[129] Tim Dietrich, Sebastiano Bernuzzi, and Wolfgang Tichy. “Closed-form tidal approximants for binary neutron star gravitational waveforms constructed from high-resolution numerical relativity simulations”. In: *Phys. Rev. D* 96.12 (2017), p. 121501. DOI: 10.1103/PhysRevD.96.121501. arXiv: 1706.02969 [gr-qc] (cit. on pp. 22, 50, 188).

[130] Tim Dietrich et al. “Matter imprints in waveform models for neutron star binaries: Tidal and self-spin effects”. In: *Phys. Rev. D* 99.2 (2019), p. 024029. DOI: 10.1103/PhysRevD.99.024029. arXiv: 1804.02235 [gr-qc] (cit. on pp. 22, 50).

[131] B.P. Abbott et al. “Properties of the Binary Black Hole Merger GW150914”. In: *Phys. Rev. Lett.* 116.24 (2016), p. 241102. DOI: 10.1103/PhysRevLett.116.241102. arXiv: 1602.03840 [gr-qc] (cit. on p. 22).

[132] B. P. Abbott et al. “GW151226: Observation of Gravitational Waves from a 22-Solar-Mass Binary Black Hole Coalescence”. In: *Phys. Rev. Lett.* 116.24 (2016), p. 241103. DOI: 10.1103/PhysRevLett.116.241103. arXiv: 1606.04855 [gr-qc] (cit. on p. 22).

[133] Benjamin P. Abbott et al. “GW170104: Observation of a 50-Solar-Mass Binary Black Hole Coalescence at Redshift 0.2”. In: *Phys. Rev. Lett.* 118.22 (2017). [Erratum: Phys.Rev.Lett. 121, 129901 (2018)], p. 221101. DOI: 10.1103/PhysRevLett.118.221101. arXiv: 1706.01812 [gr-qc] (cit. on p. 22).

[134] B.P. Abbott et al. “GW170814: A Three-Detector Observation of Gravitational Waves from a Binary Black Hole Coalescence”. In: *Phys. Rev. Lett.* 119.14 (2017), p. 141101. DOI: 10.1103/PhysRevLett.119.141101. arXiv: 1709.09660 [gr-qc] (cit. on p. 22).

[135] B.. P.. Abbott et al. “GW170608: Observation of a 19-solar-mass Binary Black Hole Coalescence”. In: *Astrophys. J.* 851.2 (2017), p. L35. DOI: 10.3847/2041-8213/aa9f0c. arXiv: 1711.05578 [astro-ph.HE] (cit. on p. 22).

[136] B.P. Abbott et al. “Multi-messenger Observations of a Binary Neutron Star Merger”. In: *Astrophys. J. Lett.* 848.2 (2017), p. L12. DOI: 10.3847/2041-8213/aa91c9. arXiv: 1710.05833 [astro-ph.HE] (cit. on p. 22).

[137] E. Burns et al. “A Fermi Gamma-ray Burst Monitor Search for Electromagnetic Signals Coincident with Gravitational-Wave Candidates in Advanced LIGO’s First Observing Run”. In: *Astrophys. J.* 871.1 (2019), p. 90. DOI: 10.3847/1538-4357/aaf726. arXiv: 1810.02764 [astro-ph.HE] (cit. on p. 22).

[138] G. P. Lamb et al. “The optical afterglow of GW170817 at one year post-merger”. In: *Astrophys. J. Lett.* 870.2 (2019), p. L15. DOI: 10.3847/2041-8213/aaf96b. arXiv: 1811.11491 [astro-ph.HE] (cit. on p. 22).

[139] R. Hamburg et al. “A Joint Fermi-GBM and LIGO/Virgo Analysis of Compact Binary Mergers From the First and Second Gravitational-wave Observing Runs”. In: *Astrophys. J.* 893 (2020), p. 100. DOI: 10.3847/1538-4357/ab7d3e. arXiv: 2001.00923 [astro-ph.HE] (cit. on p. 22).

[140] B.P. Abbott et al. “On the Progenitor of Binary Neutron Star Merger GW170817”. In: *Astrophys. J. Lett.* 850.2 (2017), p. L40. DOI: 10.3847/2041-8213/aa93fc. arXiv: 1710.05838 [astro-ph.HE] (cit. on p. 22).

[141] B.P. Abbott et al. “Properties of the binary neutron star merger GW170817”. In: *Phys. Rev. X* 9.1 (2019), p. 011001. DOI: 10.1103/PhysRevX.9.011001. arXiv: 1805.11579 [gr-qc] (cit. on p. 22).

[142] J. D. Lyman et al. “The optical afterglow of the short gamma-ray burst associated with GW170817”. In: *Nature Astron.* 2.9 (2018), pp. 751–754. DOI: 10.1038/s41550-018-0511-3. arXiv: 1801.02669 [astro-ph.HE] (cit. on p. 22).

[143] B.P. Abbott et al. “A gravitational-wave standard siren measurement of the Hubble constant”. In: *Nature* 551.7678 (2017), pp. 85–88. DOI: 10.1038/nature24471. arXiv: 1710.05835 [astro-ph.CO] (cit. on p. 22).

[144] B.P. Abbott et al. “GW170817: Measurements of neutron star radii and equation of state”. In: *Phys. Rev. Lett.* 121.16 (2018), p. 161101. DOI: 10.1103/PhysRevLett.121.161101. arXiv: 1805.11581 [gr-qc] (cit. on p. 22).

[145] B.P. Abbott et al. “Constraining the p-Mode–g-Mode Tidal Instability with GW170817”. In: *Phys. Rev. Lett.* 122.6 (2019), p. 061104. DOI: 10.1103/PhysRevLett.122.061104. arXiv: 1808.08676 [astro-ph.HE] (cit. on p. 22).

[146] R. Abbott et al. “First joint observation by the underground gravitational-wave detector KAGRA with GEO 600”. In: *Prog. Theor. Exp. Phys.* 2022.6 (Apr. 2022). 063F01. ISSN: 2050-3911. DOI: 10.1093/ptep/ptac073. arXiv: 2203.01270 [gr-qc] (cit. on p. 22).

[147] R. Abbott et al. “GW190521: A Binary Black Hole Merger with a Total Mass of 150 $M_{\odot}$”. In: *Phys. Rev. Lett.* 125 (2020), p. 101102. DOI: 10.1103/PhysRevLett.125.101102. arXiv: 2009.01075 [gr-qc] (cit. on pp. 22, 195, 250).

[148] R. Abbott et al. “Properties and astrophysical implications of the 150 Msun binary black hole merger GW190521”. In: *Astrophys. J. Lett.* 900 (2020), p. L13. DOI: 10.3847/2041-8213/aba493. arXiv: 2009.01190 [astro-ph.HE] (cit. on pp. 22, 23).

[149] R. Abbott et al. “Observation of gravitational waves from two neutron star-black hole coalescences”. In: *Astrophys. J. Lett.* 915 (2021), p. L5. DOI: 10.3847/2041-8213/ac082e. arXiv: 2106.15163 [astro-ph.HE] (cit. on p. 22).

[150] Mark Hannam et al. “General-relativistic precession in a black-hole binary”. In: *Nature* 610.7933 (Oct. 2022), pp. 652–655. DOI: 10.1038/s41586-022-05212-z (cit. on pp. 22, 23).

[151] B.P. Abbott et al. “Binary Black Hole Population Properties Inferred from the First and Second Observing Runs of Advanced LIGO and Advanced Virgo”. In: *Astrophys. J. Lett.* 882.2 (2019), p. L24. DOI: 10.3847/2041-8213/ab3800. arXiv: 1811.12940 [astro-ph.HE] (cit. on p. 22).

[152] R. Abbott et al. “Population Properties of Compact Objects from the Second LIGO-Virgo Gravitational-Wave Transient Catalog”. In: *Astrophys. J. Lett.* 913.1 (2021), p. L7. DOI: 10.3847/2041-8213/abe949. arXiv: 2010.14533 [astro-ph.HE] (cit. on p. 22).

[153] R. Abbott et al. “The population of merging compact binaries inferred using gravitational waves through GWTC-3”. In: (Nov. 2021). arXiv: 2111.03634 [astro-ph.HE] (cit. on p. 22).

[154] B. P. Abbott et al. “A Gravitational-wave Measurement of the Hubble Constant Following the Second Observing Run of Advanced LIGO and Virgo”. In: *Astrophys. J.* 909.2 (2021), p. 218. DOI: 10.3847/1538-4357/abdcb7. arXiv: 1908.06060 [astro-ph.CO] (cit. on p. 22).

[155] M. Soares-Santos et al. “First Measurement of the Hubble Constant from a Dark Standard Siren using the Dark Energy Survey Galaxies and the LIGO/Virgo Binary–Black-hole Merger GW170814”. In: *Astrophys. J. Lett.* 876.1 (2019), p. L7. DOI: 10.3847/2041-8213/ab14f1. arXiv: 1901.01540 [astro-ph.CO] (cit. on p. 22).

[156] Rachel Gray et al. “Cosmological inference using gravitational wave standard sirens: A mock data analysis”. In: *Phys. Rev. D* 101.12 (2020), p. 122001. DOI: 10.1103/PhysRevD.101.122001. arXiv: 1908.06050 [gr-qc] (cit. on p. 22).

[157] R. Abbott et al. “Constraints on the cosmic expansion history from GWTC-3”. In: (Nov. 2021). arXiv: 2111.03604 [astro-ph.CO] (cit. on p. 22).

[158] B.P. Abbott et al. “Tests of general relativity with GW150914”. In: *Phys. Rev. Lett.* 116.22 (2016). [Erratum: Phys.Rev.Lett. 121, 129902 (2018)], p. 221101. DOI: 10.1103/PhysRevLett.116.221101. arXiv: 1602.03841 [gr-qc] (cit. on p. 22).

[159] B.P. Abbott et al. “Tests of General Relativity with GW170817”. In: *Phys. Rev. Lett.* 123.1 (2019), p. 011102. DOI: 10.1103/PhysRevLett.123.011102. arXiv: 1811.00364 [gr-qc] (cit. on p. 22).

[160] B.P. Abbott et al. "Tests of General Relativity with the Binary Black Hole Signals from the LIGO-Virgo Catalog GWTC-1". In: *Phys. Rev. D* 100.10 (2019), p. 104036. DOI: 10.1103/PhysRevD.100.104036. arXiv: 1903.04467 [gr-qc] (cit. on pp. 22, 50).

[161] R. Abbott et al. "Tests of general relativity with binary black holes from the second LIGO-Virgo gravitational-wave transient catalog". In: *Phys. Rev. D* 103.12 (2021), p. 122002. DOI: 10.1103/PhysRevD.103.122002. arXiv: 2010.14529 [gr-qc] (cit. on p. 22).

[162] R. Abbott et al. "Tests of General Relativity with GWTC-3". In: (Dec. 2021). arXiv: 2112.06861 [gr-qc] (cit. on pp. 22, 50).

[163] *LIGO, VIRGO AND KAGRA OBSERVING RUN PLANS*. Accessed: 2023/27/03. URL: %5Curl%7Bhttps://observing.docs.ligo.org/plan/%7D (cit. on p. 22).

[164] John Miller et al. "Prospects for doubling the range of Advanced LIGO". In: *Phys. Rev. D* 91 (2015), p. 062005. DOI: 10.1103/PhysRevD.91.062005. arXiv: 1410.5882 [gr-qc] (cit. on p. 23).

[165] T. Akutsu et al. "Overview of KAGRA : KAGRA science". In: *PTEP* 2021.5 (2021), 05A103. DOI: 10.1093/ptep/ptaa120. arXiv: 2008.02921 [gr-qc] (cit. on p. 23).

[166] F. Acernese et al. "Advanced Virgo Plus: Future Perspectives". In: *J. Phys. Conf. Ser.* 2429.1 (2023), p. 012040. DOI: 10.1088/1742-6596/2429/1/012040 (cit. on p. 23).

[167] Bala Iyer et al. *LIGO-India, Proposal of the Consortium for Indian Initiative in Gravitational-wave Observations (IndIGO)*. Tech. rep. LIGO Scientific Collaboration, Nov. 2011. URL: https://dcc.ligo.org/LIGO-M1100296/public (cit. on p. 23).

[168] R. Abbott et al. "GW190814: Gravitational Waves from the Coalescence of a 23 Solar Mass Black Hole with a 2.6 Solar Mass Compact Object". In: *Astrophys. J.* 896.2 (2020), p. L44. DOI: 10.3847/2041-8213/ab960f. arXiv: 2006.12611 [astro-ph.HE] (cit. on p. 23).

[169] Isobel M. Romero-Shaw, Paul D. Lasky, and Eric Thrane. "Four Eccentric Mergers Increase the Evidence that LIGOVirgoKAGRAs Binary Black Holes Form Dynamically". In: *Astrophys. J.* 940.2 (2022), p. 171. DOI: 10.3847/1538-4357/ac9798. arXiv: 2206.14695 [astro-ph.HE] (cit. on p. 23).

[170] Michael Ebersold et al. "Observational limits on the rate of radiation-driven binary black hole capture events". In: *Phys. Rev. D* 106.10 (2022), p. 104014. DOI: 10.1103/PhysRevD.106.104014. arXiv: 2208.07762 [gr-qc] (cit. on p. 23).

[171] Weichangfeng Guo et al. "Mimicking mergers: mistaking black hole captures as mergers". In: *Mon. Not. Roy. Astron. Soc.* 516.3 (2022), pp. 3847–3860. DOI: 10.1093/mnras/stac2385. arXiv: 2203.06969 [gr-qc] (cit. on p. 23).

[172] Ethan Payne, Colm Talbot, and Eric Thrane. "Higher order gravitational-wave modes with likelihood reweighting". In: *Phys. Rev. D* 100.12 (2019), p. 123017. DOI: 10.1103/PhysRevD.100.123017. arXiv: 1905.05477 [astro-ph.IM] (cit. on p. 23).

[173] Eamonn O'Shea and Prayush Kumar. "Correlations in parameter estimation of low-mass eccentric binaries: GW151226 & GW170608". In: (July 2021). arXiv: 2107.07981 [astro-ph.HE] (cit. on p. 23).

[174] Thomas Bayes Rev. "An essay toward solving a problem in the doctrine of chances". In: *Phil. Trans. Roy. Soc. Lond.* 53 (1764), pp. 370–418. DOI: 10.1098/rstl.1763.0053 (cit. on p. 26).

[175] A. I. Dale. "Bayes or Laplace? An Examination of the Origin and Early Applications of Bayes' Theorem". In: *Archive for History of Exact Sciences* 27.1 (1982), pp. 23–47. ISSN: 00039519, 14320657. URL: http://www.jstor.org/stable/41133662 (visited on 10/12/2023) (cit. on p. 26).

[176] Harold Jeffreys. *The Theory of Probability*. Oxford Classic Texts in the Physical Sciences. 1939. ISBN: 978-0-19-850368-2, 978-0-19-853193-7 (cit. on pp. 26, 30).

[177] R. T. Cox. "Probability, Frequency and Reasonable Expectation". In: *American Journal of Physics* 14.1 (1946), pp. 1–13. DOI: 10.1119/1.1990764. eprint: https://doi.org/10.1119/1.1990764 (cit. on p. 26).

[178] Devinderjit Sivia and John Skilling. *Data analysis: a Bayesian tutorial*. OUP Oxford, 2006 (cit. on pp. 27, 52).

[179] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. DOI: 10.1214/aoms/1177729694 (cit. on p. 27).

[180] J. Lin. "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151. DOI: 10.1109/18.61115 (cit. on p. 28).

[181] Tiancheng Li, Miodrag Bolic, and Petar M. Djuric. "Resampling Methods for Particle Filtering: Classification, implementation, and strategies". In: *IEEE Signal Processing Magazine* 32.3 (2015), pp. 70–86. DOI: 10.1109/MSP.2014.2330626 (cit. on p. 30).

[182] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”. In: vol. 140. 2. 1993, pp. 107–113 (cit. on p. 30).

[183] L. Kish. *Survey Sampling.* 1965 (cit. on pp. 31, 151).

[184] George Casella and Edward I. George. “Explaining the Gibbs Sampler”. In: *The American Statistician* 46.3 (1992), pp. 167–174. ISSN: 00031305. URL: http://www.jstor.org/stable/2685208 (visited on 01/24/2023) (cit. on p. 32).

[185] Michael Betancourt. “A Conceptual Introduction to Hamiltonian Monte Carlo”. In: *arXiv e-prints*, arXiv:1701.02434 (Jan. 2017), arXiv:1701.02434. DOI: 10.48550/arXiv.1701.02434. arXiv: 1701.02434 [stat.ME] (cit. on p. 32).

[186] Nicholas Metropolis et al. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. DOI: 10.1063/1.1699114. eprint: https://doi.org/10.1063/1.1699114 (cit. on p. 32).

[187] W. K. Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109. ISSN: 0006-3444. DOI: 10.1093/biomet/57.1.97. eprint: https://academic.oup.com/biomet/article-pdf/57/1/97/23940249/57-1-97.pdf (cit. on p. 32).

[188] David W. Hogg and Daniel Foreman-Mackey. “Data analysis recipes: Using Markov Chain Monte Carlo”. In: *Astrophys. J. Suppl.* 236.1 (2018), p. 11. DOI: 10.3847/1538-4365/aab76e. arXiv: 1710.06068 [astro-ph.IM] (cit. on p. 32).

[189] Luke Tierney. “Markov Chains for Exploring Posterior Distributions”. In: *The Annals of Statistics* 22.4 (1994), pp. 1701–1728. DOI: 10.1214/aos/1176325750 (cit. on p. 32).

[190] Paul M. Goggans and Ying Chi. “Using Thermodynamic Integration to Calculate the Posterior Probability in Bayesian Model Selection Problems”. In: *AIP Conference Proceedings* 707.1 (2004), pp. 59–66. DOI: 10.1063/1.1751356. eprint: https://aip.scitation.org/doi/pdf/10.1063/1.1751356 (cit. on p. 32).

[191] N. Friel and A. N. Pettitt. “Marginal Likelihood Estimation via Power Posteriors”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 70.3 (2008), pp. 589–607. ISSN: 13697412, 14679868. URL: http://www.jstor.org/stable/20203843 (cit. on p. 32).

[192] John Skilling. “Nested Sampling”. In: *AIP Conference Proceedings* 735.1 (2004), pp. 395–405. DOI: 10.1063/1.1835238. eprint: https://aip.scitation.org/doi/pdf/10.1063/1.1835238 (cit. on p. 33).

[193] John Skilling. “Nested sampling for general Bayesian computation”. In: *Bayesian Analysis* 1.4 (2006), pp. 833–859. DOI: 10.1214/06-BA127 (cit. on pp. 33–38, 210, 221).

[194] Livia B. Pártay, Gábor Csányi, and Noam Bernstein. “Nested sampling for materials”. In: *The European Physical Journal B* 94.8 (Aug. 2021). DOI: 10.1140/epjb/s10051-021-00172-1 (cit. on p. 33).

[195] Greg Ashton et al. “Nested sampling for physical scientists”. In: *Nature* 2 (2022). DOI: 10.1038/s43586-022-00121-x. arXiv: 2205.15570 [stat.CO] (cit. on pp. 33, 37, 210).

[196] Johannes Buchner. “Nested Sampling Methods”. In: *arXiv e-prints*, arXiv:2101.09675 (Jan. 2021), arXiv:2101.09675. DOI: 10.48550/arXiv.2101.09675. arXiv: 2101.09675 [stat.CO] (cit. on pp. 35, 38, 142).

[197] Joshua S. Speagle. “DYNESTY: a dynamic nested sampling package for estimating Bayesian posteriors and evidences”. In: *Monthly Notices of the Royal Astronomical Society* 493.3 (Apr. 2020), pp. 3132–3158. DOI: 10.1093/mnras/staa278. arXiv: 1904.02180 [astro-ph.IM] (cit. on pp. 36, 38, 39, 54, 90, 92, 151, 157, 170, 247).

[198] Gregory Ashton et al. “BILBY: A User-friendly Bayesian Inference Library for Gravitational-wave Astronomy”. In: *Astrophysical Journal Supplement Series* 241.2, 27 (Apr. 2019), p. 27. DOI: 10.3847/1538-4365/ab06fc. arXiv: 1811.02042 [astro-ph.IM] (cit. on pp. 37, 49, 51, 52, 55, 78, 82, 89, 91, 92, 119, 123, 129, 130, 142, 157, 169, 170, 203, 218, 245, 247).

[199] Charles R. Keeton. “On statistical uncertainty in nested sampling”. In: *Monthly Notices of the Royal Astronomical Society* 414.2 (Apr. 2011), pp. 1418–1426. DOI: 10.1111/j.1365-2966.2011.18474.x (cit. on p. 37).

[200] Edward Higson et al. “Sampling Errors in Nested Sampling Parameter Estimation”. In: *Bayesian Analysis* 13.3 (2018), pp. 873–896. DOI: 10.1214/17-BA1075 (cit. on pp. 37, 105).

[201] Radford M. Neal. “Slice Sampling”. In: *The Annals of Statistics* 31.3 (2003), pp. 705–741. ISSN: 00905364. URL: http://www.jstor.org/stable/3448413 (cit. on p. 38).

[202] John Veitch et al. *johnveitch/cpnest*. DOI: 10.5281/zenodo.7385221 (cit. on pp. 38, 247).

[203] Edward Higson et al. “Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation”. In: *Statistics and Computing* 29.5 (Sept. 2019), pp. 891–913. DOI: 10.1007/s11222-018-9844-0. arXiv: 1704.03459 [stat.CO] (cit. on pp. 38–40, 154, 220, 221).

[204] Brendon J. Brewer and Daniel Foreman-Mackey. “DNest4: Diffusive Nested Sampling in C++ and Python”. In: *Journal of Statistical Software* 86.7 (2018), pp. 1–33. DOI: 10.18637/jss.v086.i07 (cit. on p. 38).

[205] K Barbary. *kbarbary/nestle: v0.2.0*. Version v0.2.0. 2016. URL: https://github.com/kbarbary/nestle (cit. on p. 38).

[206] W. J. Handley, M. P. Hobson, and A. N. Lasenby. “polychord: nested sampling for cosmology.” In: *Monthly Notices of the Royal Astronomical Society* 450 (June 2015), pp. L61–L65. DOI: 10.1093/mnrasl/slv047. arXiv: 1502.01856 [astro-ph.CO] (cit. on pp. 38, 54).

[207] Johannes Buchner. “UltraNest - a robust, general purpose Bayesian inference engine”. In: *Journal of Open Source Software* 6.60 (2021), p. 3001. DOI: 10.21105/joss.03001 (cit. on pp. 38, 39, 54).

[208] F. Feroz, M. P. Hobson, and M. Bridges. “MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics”. In: *Mon. Not. Roy. Astron. Soc.* 398 (2009), pp. 1601–1614. DOI: 10.1111/j.1365-2966.2009.14548.x. arXiv: 0809.3437 [astro-ph] (cit. on pp. 38, 40, 54, 220).

[209] Johannes Buchner. “A statistical test for Nested Sampling algorithms”. In: *Statistics and Computing* 26.1-2 (Jan. 2016), pp. 383–392. DOI: 10.1007/s11222-014-9512-y (cit. on p. 38).

[210] Johannes Buchner. “Collaborative Nested Sampling: Big Data versus Complex Physical Models”. In: *Publications of the Astronomical Society of the Pacific* 131.1004 (Aug. 2019), p. 108005. DOI: 10.1088/1538-3873/aae7fc (cit. on p. 38).

[211] Enrico Corsaro and Joris De Ridder. “DIAMONDS: a new Bayesian nested sampling tool”. In: *European Physical Journal Web of Conferences*. Vol. 101. European Physical Journal Web of Conferences. Sept. 2015, 06019, p. 06019. DOI: 10.1051/epjconf/201510106019. arXiv: 1509.08311 [astro-ph.IM] (cit. on p. 38).

[212] Farhan Feroz et al. “Importance Nested Sampling and the MultiNest Algorithm”. In: *The Open Journal of Astrophysics* 2.1, 10 (Nov. 2019), p. 10. DOI: 10.21105/astro.1306.2144. arXiv: 1306.2144 [astro-ph.IM] (cit. on pp. 38–41, 54, 145, 153, 163, 213).

[213] Brendon J. Brewer, Livia B. Pártay, and Gábor Csányi. “Diffusive Nested Sampling”. In: *arXiv e-prints*, arXiv:0912.2380 (Dec. 2009), arXiv:0912.2380. arXiv: 0912.2380 [stat.CO] (cit. on pp. 39, 149, 153, 163, 213, 221).

[214] Ewan Cameron and Anthony Pettitt. “Recursive Pathways to Marginal Likelihood Estimation with Prior-Sensitivity Analysis”. In: *Statistical Science* 29.3 (2014), pp. 397–419. DOI: 10.1214/13-STS465 (cit. on pp. 40, 41).

[215] Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. “Elements of Sequential Monte Carlo”. In: *arXiv e-prints*, arXiv:1903.04797 (Mar. 2019), arXiv:1903.04797. arXiv: 1903.04797 [stat.ML] (cit. on p. 41).

[216] Robert Salomone et al. “Unbiased and Consistent Nested Sampling via Sequential Monte Carlo”. In: *arXiv e-prints*, arXiv:1805.03924 (May 2018), arXiv:1805.03924. arXiv: 1805.03924 [stat.CO] (cit. on pp. 41, 145, 149, 163, 213).

[217] Edward Higson et al. “Nestcheck: diagnostic tests for nested sampling calculations”. In: *Mon. Not. Roy. Astron. Soc.* 483.2 (2019), pp. 2044–2056. DOI: 10.1093/mnras/sty3090. arXiv: 1804.06406 [stat.CO] (cit. on p. 42).

[218] Andrew Fowlie, Will Handley, and Liangliang Su. “Nested sampling cross-checks using order statistics”. In: *Mon. Not. Roy. Astron. Soc.* 497.4 (2020), pp. 5256–5263. DOI: 10.1093/mnras/staa2345. arXiv: 2006.03371 [stat.CO] (cit. on pp. 42, 97, 105, 157, 165, 237).

[219] A. Kolmogorov. “Sulla determinazione empirica di una legge di distribuzione”. In: *Giornale dell’Istituto Italiano degli Attuari* 4 (1933), pp. 83–91 (cit. on pp. 42, 44, 97, 237).

[220] N. Smirnov. “Table for Estimating the Goodness of Fit of Empirical Distributions”. In: *The Annals of Mathematical Statistics* 19.2 (1948), pp. 279–281. DOI: 10.1214/aoms/1177730256 (cit. on pp. 42, 44, 97, 237).

[221] Taylor B. Arnold and John W. Emerson. “Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions”. In: *The R Journal* 3.2 (2011), pp. 34–39. DOI: 10.32614/RJ-2011-016 (cit. on pp. 42, 97, 237).

[222] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. “The frontier of simulation-based inference”. In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30055–30062. DOI: 10.1073/pnas.1912789117. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1912789117 (cit. on p. 43).

[223] S. A. Sisson, Y. Fan, and M. A. Beaumont. “Overview of Approximate Bayesian Computation”. In: *arXiv e-prints*, arXiv:1802.09720 (Feb. 2018), arXiv:1802.09720. DOI: 10.48550/arXiv.1802.09720. arXiv: 1802.09720 [stat.CO] (cit. on p. 43).

[224] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877. DOI: 10.1080/01621459.2017.1285773. eprint: https://doi.org/10.1080/01621459.2017.1285773 (cit. on pp. 43, 144).

[225] Manuel Glöckler, Michael Deistler, and Jakob H. Macke. “Variational methods for simulation-based inference”. In: *arXiv e-prints*, arXiv:2203.04176 (Mar. 2022), arXiv:2203.04176. DOI: 10.48550/arXiv.2203.04176. arXiv: 2203.04176 [stat.ML] (cit. on p. 44).

[226] Danilo Jimenez Rezende and Shakir Mohamed. “Variational Inference with Normalizing Flows”. In: *arXiv e-prints*, arXiv:1505.05770 (May 2015), arXiv:1505.05770. DOI: 10.48550/arXiv.1505.05770. arXiv: 1505.05770 [stat.ML] (cit. on pp. 44, 73).

[227] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: (Dec. 2013). arXiv: 1312.6114 [stat.ML] (cit. on pp. 44, 66, 70, 254).

[228] John R. Michael. “The Stabilized Probability Plot”. In: *Biometrika* 70.1 (1983), pp. 11–17. ISSN: 00063444. URL: http://www.jstor.org/stable/2335939 (visited on 06/14/2023) (cit. on p. 44).

[229] Samantha R Cook, Andrew Gelman, and Donald B Rubin. “Validation of Software for Bayesian Models Using Posterior Quantiles”. In: *Journal of Computational and Graphical Statistics* 15.3 (2006), pp. 675–692. DOI: 10.1198/106186006X136976. eprint: https://doi.org/10.1198/106186006X136976 (cit. on pp. 44, 46, 91).

[230] Sean Talts et al. “Validating Bayesian Inference Algorithms with Simulation-Based Calibration”. In: *arXiv e-prints*, arXiv:1804.06788 (Apr. 2018), arXiv:1804.06788. arXiv: 1804.06788 [stat.ME] (cit. on pp. 44, 91).

[231] R. A. Fisher. *Statistical Methods for Research Workers*. Ed. by Samuel Kotz and Norman L. Johnson. New York, NY: Springer New York, 1992, pp. 66–70. ISBN: 978-1-4612-4380-9. DOI: 10.1007/978-1-4612-4380-9_6 (cit. on p. 46).

[232] Fernando Perez-Cruz. “Kullback-Leibler divergence estimation of continuous distributions”. In: *2008 IEEE International Symposium on Information Theory*. 2008, pp. 1666–1670. DOI: 10.1109/ISIT.2008.4595271 (cit. on p. 47).

[233] Gregory Ashton and Colm Talbot. “B ilby-MCMC: an MCMC sampler for gravitational-wave inference”. In: *Mon. Not. Roy. Astron. Soc.* 507.2 (2021), pp. 2037–2051. DOI: 10.1093/mnras/stab2236. arXiv: 2106.08730 [gr-qc] (cit. on pp. 47, 54, 79, 96, 160, 173, 239).

[234] Hunter Gabbard et al. “Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy”. In: *Nature Phys.* 18.1 (2022), pp. 112–117. DOI: 10.1038/s41567-021-01425-7. arXiv: 1909.06296 [astro-ph.IM] (cit. on pp. 47, 79, 82, 90, 102, 153, 218, 219).

[235] Joe Bayley, Chris Messenger, and Graham Woan. “Rapid parameter estimation for an all-sky continuous gravitational wave search using conditional varitational auto-encoders”. In: *Physical Review D* 106.8, 083022 (Oct. 2022), p. 083022. DOI: 10.1103/PhysRevD.106.083022. arXiv: 2209.02031 [astro-ph.IM] (cit. on pp. 47, 79).

[236] Maximilian Dax et al. “Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference”. In: (Oct. 2022). arXiv: 2210.05686 [gr-qc] (cit. on pp. 47, 79, 82, 219).

[237] J. Veitch and A. Vecchio. “Bayesian coherent analysis of in-spiral gravitational wave signals with a detector network”. In: *Phys. Rev. D* 81 (2010), p. 062003. DOI: 10.1103/PhysRevD.81.062003. arXiv: 0911.3820 [astro-ph.CO] (cit. on pp. 47, 48, 54, 93).

[238] Benjamin P Abbott et al. “A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals”. In: *Class. Quant. Grav.* 37.5 (2020), p. 055002. DOI: 10.1088/1361-6382/ab685e. arXiv: 1908.11170 [gr-qc] (cit. on pp. 48, 51, 52).

[239] Michele Maggiore. “3357 Data analysis techniques”. In: *Gravitational Waves: Volume 1: Theory and Experiments*. Oxford University Press, Oct. 2007. ISBN: 9780198570745. DOI: 10.1093/acprof:oso/9780198570745.003.0007. eprint: https://academic.oup.com/book/0/chapter/353716628/chapter-pdf/43657980/acprof-9780198570745-chapter-7.pdf (cit. on pp. 49, 50).

[240] Tyson B. Littenberg and Neil J. Cornish. “Bayesian inference for spectral estimation of gravitational wave detector noise”. In: *Phys. Rev. D* 91.8 (2015), p. 084034. DOI: 10.1103/PhysRevD.91.084034. arXiv: 1410.3852 [gr-qc] (cit. on p. 51).

[241] Jade Powell. “Parameter Estimation and Model Selection of Gravitational Wave Signals Contaminated by Transient Detector Noise Glitches”. In: *Class. Quant. Grav.* 35.15 (2018), p. 155017. DOI: 10.1088/1361-6382/aacf18. arXiv: 1803.11346 [astro-ph.IM] (cit. on p. 51).

[242] Simone Mozzon et al. “Does nonstationary noise in LIGO and Virgo affect the estimation of H0?” In: *Phys. Rev. D* 106.4 (2022), p. 043504. DOI: 10.1103/PhysRevD.106.043504. arXiv: 2110.11731 [astro-ph.CO] (cit. on p. 51).

[243] Jack Y. L. Kwok et al. “Investigation of the effects of non-Gaussian noise transients and their mitigation in parameterized gravitational-wave tests of general relativity”. In: *Phys. Rev. D* 105.2 (2022), p. 024066. DOI: 10.1103/PhysRevD.105.024066. arXiv: 2109.07642 [gr-qc] (cit. on p. 51).

[244] Ronaldas Macas et al. "Impact of noise transients on low latency gravitational-wave event localization". In: *Phys. Rev. D* 105.10 (2022), p. 103021. DOI: 10.1103/PhysRevD.105.103021. arXiv: 2202.00344 [astro-ph.HE] (cit. on p. 51).

[245] S. Mukherjee, R. Obaid, and B. Matkarimov. "Classification of glitch waveforms in gravitational wave detector characterization". In: *J. Phys. Conf. Ser.* 243 (2010). Ed. by Fulvio Ricci, p. 012006. DOI: 10.1088/1742-6596/243/1/012006 (cit. on p. 51).

[246] Chris Pankow et al. "Mitigation of the instrumental noise transient in gravitational-wave data surrounding GW170817". In: *Phys. Rev. D* 98.8 (2018), p. 084016. DOI: 10.1103/PhysRevD.98.084016. arXiv: 1808.03619 [gr-qc] (cit. on p. 51).

[247] Jonathan Merritt et al. "Transient glitch mitigation in Advanced LIGO data". In: *Phys. Rev. D* 104.10 (2021), p. 102004. DOI: 10.1103/PhysRevD.104.102004. arXiv: 2108.12044 [gr-qc] (cit. on p. 51).

[248] D. Davis et al. "Subtracting glitches from gravitational-wave detector data during the third LIGO-Virgo observing run". In: *Class. Quant. Grav.* 39.24 (2022), p. 245013. DOI: 10.1088/1361-6382/aca238. arXiv: 2207.03429 [astro-ph.IM] (cit. on p. 51).

[249] Salvatore Vitale et al. "Impact of Bayesian Priors on the Characterization of Binary Black Hole Coalescences". In: *Phys. Rev. Lett.* 119.25 (2017), p. 251103. DOI: 10.1103/PhysRevLett.119.251103. arXiv: 1707.04637 [gr-qc] (cit. on p. 52).

[250] Jacob Lange, Richard O'Shaughnessy, and Monica Rizzo. "Rapid and accurate parameter inference for coalescing, precessing compact binaries". In: (May 2018). arXiv: 1805.10457 [gr-qc] (cit. on pp. 52, 55, 142).

[251] Eric Thrane and Colm Talbot. "An introduction to Bayesian inference in gravitational-wave astronomy: Parameter estimation, model selection, and hierarchical models". In: *Publications of the Astronomical Society of Australia* 36 (2019). DOI: 10.1017/pasa.2019.2 (cit. on pp. 52, 53, 203).

[252] John Veitch and Walter Del Pozzo. *Analytic Marginalisation of Phase Parameter*. Tech. rep. T1300326. LIGO, Apr. 2013 (cit. on pp. 53, 121, 122).

[253] Leo P. Singer and Larry R. Price. "Rapid Bayesian position reconstruction for gravitational-wave transients". In: *Phys. Rev. D* 93.2 (2016), p. 024013. DOI: 10.1103/PhysRevD.93.024013. arXiv: 1508.03634 [gr-qc] (cit. on p. 53).

[254] Leo P. Singer et al. "Going the Distance: Mapping Host Galaxies of LIGO and Virgo Sources in Three Dimensions Using Local Cosmography and Targeted Follow-up". In: *Astrophys. J. Lett.* 829.1 (2016), p. L15. DOI: 10.3847/2041-8205/829/1/L15. arXiv: 1603.07333 [astro-ph.HE] (cit. on p. 53).

[255] Will M. Farr. *Marginalisation of the Time Parameter in Gravitational Wave Parameter Estimation.* Tech. rep. T1400460. LIGO, Oct. 2014 (cit. on p. 53).

[256] Michael Pürrer. "Frequency domain reduced order models for gravitational waves from aligned-spin compact binaries". In: *Class. Quant. Grav.* 31.19 (2014), p. 195010. DOI: 10.1088/0264-9381/31/19/195010. arXiv: 1402.4146 [gr-qc] (cit. on p. 53).

[257] Michael Pürrer. "Frequency domain reduced order model of aligned-spin effective-one-body waveforms with generic mass-ratios and spins". In: *Phys. Rev. D* 93.6 (2016), p. 064041. DOI: 10.1103/PhysRevD.93.064041. arXiv: 1512.02248 [gr-qc] (cit. on p. 53).

[258] Maxime Barrault et al. "An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672. ISSN: 1631-073X. DOI: https://doi.org/10.1016/j.crma.2004.08.006 (cit. on p. 53).

[259] Priscilla Canizares et al. "Accelerated gravitational-wave parameter estimation with reduced order modeling". In: *Phys. Rev. Lett.* 114.7 (2015), p. 071104. DOI: 10.1103/PhysRevLett.114.071104. arXiv: 1404.6284 [gr-qc] (cit. on p. 53).

[260] Rory Smith et al. "Fast and accurate inference on gravitational waves from precessing compact binaries". In: *Phys. Rev. D* 94.4 (2016), p. 044031. DOI: 10.1103/PhysRevD.94.044031. arXiv: 1604.08253 [gr-qc] (cit. on pp. 53, 159, 165).

[261] Soichiro Morisaki and Vivien Raymond. "Rapid Parameter Estimation of Gravitational Waves from Binary Neutron Star Coalescence using Focused Reduced Order Quadrature". In: *Phys. Rev. D* 102.10 (2020), p. 104020. DOI: 10.1103/PhysRevD.102.104020. arXiv: 2007.09108 [gr-qc] (cit. on p. 53).

[262] B.P. Abbott et al. "GW190425: Observation of a Compact Binary Coalescence with Total Mass $\sim 3.4 M_{\odot}$". In: *Astrophys. J. Lett.* 892 (2020), p. L3. DOI: 10.3847/2041-8213/ab75f5. arXiv: 2001.01761 [astro-ph.HE] (cit. on pp. 53, 159, 192, 208, 215, 250).

[263] Soichiro Morisaki. "Accelerating parameter estimation of gravitational waves from compact binary coalescence using adaptive frequency resolutions". In: *Phys. Rev. D* 104.4 (2021), p. 044062. DOI: 10.1103/PhysRevD.104.044062. arXiv: 2104.07813 [gr-qc] (cit. on pp. 53, 54).

[264] Neil J. Cornish. "Fast Fisher Matrices and Lazy Likelihoods". In: (July 2010). arXiv: 1007.4820 [gr-qc] (cit. on p. 54).

[265] Neil J. Cornish. “Rapid and Robust Parameter Inference for Binary Mergers”. In: *Phys. Rev. D* 103.10 (2021), p. 104057. DOI: 10.1103/PhysRevD.103.104057. arXiv: 2101.01188 [gr-qc] (cit. on p. 54).

[266] Barak Zackay, Liang Dai, and Tejaswi Venumadhav. “Relative Binning and Fast Likelihood Evaluation for Gravitational Wave Parameter Estimation”. In: (June 2018). arXiv: 1806.08792 [astro-ph.IM] (cit. on p. 54).

[267] Liang Dai, Tejaswi Venumadhav, and Barak Zackay. “Parameter Estimation for GW170817 using Relative Binning”. In: (June 2018). arXiv: 1806.08793 [gr-qc] (cit. on p. 54).

[268] Nathaniel Leslie, Liang Dai, and Geraint Pratten. “Mode-by-mode relative binning: Fast likelihood estimation for gravitational waveforms with spin-orbit precession and multiple harmonics”. In: *Phys. Rev. D* 104.12 (2021), p. 123030. DOI: 10.1103/PhysRevD.104.123030. arXiv: 2109.09872 [astro-ph.IM] (cit. on p. 54).

[269] Kipp Cannon et al. “Toward Early-Warning Detection of Gravitational Waves from Compact Binary Coalescence”. In: *Astrophys. J.* 748 (2012), p. 136. DOI: 10.1088/0004-637X/748/2/136. arXiv: 1107.2665 [astro-ph.IM] (cit. on p. 54).

[270] D. Buskulic. “Very low latency search pipeline for low mass compact binary coalescences in the LIGO S6 and Virgo VSR2 data”. In: *Class. Quant. Grav.* 27.19 (2010). Ed. by Fulvio Ricci, p. 194013. DOI: 10.1088/0264-9381/27/19/194013 (cit. on p. 54).

[271] Serena Vinciguerra, John Veitch, and Ilya Mandel. “Accelerating gravitational wave parameter estimation with multi-band template interpolation”. In: *Class. Quant. Grav.* 34.11 (2017), p. 115006. DOI: 10.1088/1361-6382/aa6d44. arXiv: 1703.02062 [gr-qc] (cit. on p. 54).

[272] Nelson Christensen and Renate Meyer. “Using Markov chain Monte Carlo methods for estimating parameters with gravitational radiation data”. In: *Phys. Rev. D* 64 (2001), p. 022001. DOI: 10.1103/PhysRevD.64.022001. arXiv: gr-qc/0102018 (cit. on p. 54).

[273] Nelson Christensen, Renate Meyer, and Adam Libson. “A MetropolisHastings routine for estimating parameters from compact binary inspiral events with laser interferometric gravitational radiation data”. In: *Classical and Quantum Gravity* 21.1 (Dec. 2003), p. 317. DOI: 10.1088/0264-9381/21/1/023 (cit. on p. 54).

[274] Christian Rover, Renate Meyer, and Nelson Christensen. “Bayesian inference on compact binary inspiral gravitational radiation signals in interferometric data”. In: *Class. Quant. Grav.* 23 (2006), pp. 4895–4906. DOI: 10.1088/0264-9381/23/15/009. arXiv: gr-qc/0602067 (cit. on p. 54).

[275] Christian Rover, Renate Meyer, and Nelson Christensen. “Coherent Bayesian inference on compact binary inspirals using a network of interferometric gravitational wave detectors”. In: *Phys. Rev. D* 75 (2007), p. 062004. DOI: 10.1103/PhysRevD.75.062004. arXiv: gr-qc/0609131 (cit. on p. 54).

[276] M. V. van der Sluys et al. “Gravitational-Wave Astronomy with Inspiral Signals of Spinning Compact-Object Binaries”. In: *Astrophys. J. Lett.* 688 (2008), p. L61. DOI: 10.1086/595279. arXiv: 0710.1897 [astro-ph] (cit. on p. 54).

[277] Marc van der Sluys et al. “Parameter estimation of spinning binary inspirals using Markov-chain Monte Carlo”. In: *Class. Quant. Grav.* 25 (2008). Ed. by S. Hughes and E. Katsavounidis, p. 184011. DOI: 10.1088/0264-9381/25/18/184011. arXiv: 0805.1689 [gr-qc] (cit. on p. 54).

[278] V. Raymond et al. “Degeneracies in Sky Localisation Determination from a Spinning Coalescing Binary through Gravitational Wave Observations: A Markov-Chain Monte-Carlo Analysis for two Detectors”. In: *Class. Quant. Grav.* 26 (2009). Ed. by Patrick Sutton and Deirdre Shoemaker, p. 114007. DOI: 10.1088/0264-9381/26/11/114007. arXiv: 0812.4302 [gr-qc] (cit. on p. 54).

[279] Marc van der Sluys et al. “Parameter estimation for signals from compact binary inspirals injected into LIGO data”. In: *Class. Quant. Grav.* 26 (2009). Ed. by Mario Diaz, Fredrick Jenet, and Soumya Mohanty, p. 204010. DOI: 10.1088/0264-9381/26/20/204010. arXiv: 0905.1323 [gr-qc] (cit. on p. 54).

[280] V. Raymond et al. “The Effects of LIGO detector noise on a 15-dimensional Markov-chain Monte-Carlo analysis of gravitational-wave signals”. In: *Class. Quant. Grav.* 27 (2010). Ed. by Sascha Husa and Badri Krishnan, p. 114009. DOI: 10.1088/0264-9381/27/11/114009. arXiv: 0912.3746 [gr-qc] (cit. on p. 54).

[281] Carl L. Rodriguez et al. “Basic Parameter Estimation of Binary Neutron Star Systems by the Advanced LIGO/Virgo Network”. In: *Astrophys. J.* 784 (2014), p. 119. DOI: 10.1088/0004-637X/784/2/119. arXiv: 1309.3273 [astro-ph.HE] (cit. on p. 54).

[282] Salvatore Vitale et al. “Measuring the spin of black holes in binary systems using gravitational waves”. In: *Phys. Rev. Lett.* 112.25 (2014), p. 251101. DOI: 10.1103/PhysRevLett.112.251101. arXiv: 1403.0129 [gr-qc] (cit. on p. 54).

[283] Daniel Foreman-Mackey et al. “emcee: The MCMC Hammer”. In: *Publications of the Astronomical Society of the Pacific* 125.925 (Mar. 2013), p. 306. DOI: 10.1086/670067. arXiv: 1202.3665 [astro-ph.IM] (cit. on p. 54).

[284] Benjamin Farr, Vicky Kalogera, and Erik Luijten. "A more efficient approach to parallel-tempered Markov-chain Monte Carlo for the highly structured posteriors of gravitational-wave signals". In: *Phys. Rev. D* 90.2 (2014), p. 024014. DOI: 10.1103/PhysRevD.90.024014. arXiv: 1309.7709 [astro-ph.IM] (cit. on p. 54).

[285] Rory J. E. Smith et al. "Massively parallel Bayesian inference for transient gravitational-wave astronomy". In: *Mon. Not. Roy. Astron. Soc.* 498.3 (2020), pp. 4492–4502. DOI: 10.1093/mnras/staa2483. arXiv: 1909.11873 [gr-qc] (cit. on pp. 55, 142).

[286] C. Pankow et al. "Novel scheme for rapid parallel parameter estimation of gravitational waves from compact binary coalescences". In: *Phys. Rev. D* 92.2 (2015), p. 023002. DOI: 10.1103/PhysRevD.92.023002. arXiv: 1502.04370 [gr-qc] (cit. on p. 55).

[287] D. Wysocki et al. "Accelerating parameter inference with graphics processing units". In: *Phys. Rev. D* 99.8 (2019), p. 084026. DOI: 10.1103/PhysRevD.99.084026. arXiv: 1902.04934 [astro-ph.IM] (cit. on p. 55).

[288] Chad Henshaw, Richard O'Shaughnessy, and Laura Cadonati. "Implementation of a generalized precession parameter in the RIFT parameter estimation algorithm". In: *Class. Quant. Grav.* 39.12 (2022), p. 125003. DOI: 10.1088/1361-6382/ac6cc0. arXiv: 2201.05220 [gr-qc] (cit. on p. 55).

[289] J. Wofford et al. "Expanding RIFT: Improving performance for GW parameter inference". In: (Oct. 2022). arXiv: 2210.07912 [gr-qc] (cit. on p. 55).

[290] A. B. Yelikar, V. Delfavero, and R. O'Shaughnessy. "Low-latency parameter inference enabled by a Gaussian likelihood approximation for RIFT". In: (Jan. 2023). arXiv: 2301.01337 [gr-qc] (cit. on p. 55).

[291] Richard Udall et al. "RUNMON-RIFT: Adaptive configuration and healing for large-scale parameter inference". In: *Astron. Comput.* 42 (2023), p. 100664. DOI: 10.1016/j.ascom.2022.100664. arXiv: 2110.10243 [gr-qc] (cit. on p. 55).

[292] C. M. Biwer et al. "PyCBC Inference: A Python-based parameter estimation toolkit for compact binary coalescence signals". In: *Publ. Astron. Soc. Pac.* 131.996 (2019), p. 024503. DOI: 10.1088/1538-3873/aaef0b. arXiv: 1807.10312 [astro-ph.IM] (cit. on pp. 55, 78).

[293] Alexander H. Nitz et al. "1-OGC: The first open gravitational-wave catalog of binary mergers from analysis of public Advanced LIGO data". In: *Astrophys. J.* 872.2 (2019), p. 195. DOI: 10.3847/1538-4357/ab0108. arXiv: 1811.01921 [gr-qc] (cit. on p. 55).

[294] Alexander H. Nitz et al. “2-OGC: Open Gravitational-wave Catalog of binary mergers from analysis of public Advanced LIGO and Virgo data”. In: *Astrophys. J.* 891 (Mar. 2020), p. 123. DOI: 10.3847/1538-4357/ab733f. arXiv: 1910.05331 [astro-ph.HE] (cit. on p. 55).

[295] Alexander H. Nitz et al. “3-OGC: Catalog of Gravitational Waves from Compact-binary Mergers”. In: *Astrophys. J.* 922.1 (2021), p. 76. DOI: 10.3847/1538-4357/ac1c03. arXiv: 2105.09151 [astro-ph.HE] (cit. on p. 55).

[296] Alexander H. Nitz et al. “4-OGC: Catalog of gravitational waves from compact-binary mergers”. In: (Dec. 2021). arXiv: 2112.06878 [astro-ph.HE] (cit. on pp. 55, 188, 211).

[297] Matteo Breschi, Rossella Gamba, and Sebastiano Bernuzzi. “Bayesian inference of multimessenger astrophysical data: Methods and applications to gravitational waves”. In: *Phys. Rev. D* 104.4 (2021), p. 042001. DOI: 10.1103/PhysRevD.104.042001. arXiv: 2102.00017 [gr-qc] (cit. on p. 55).

[298] Stephen Fairhurst et al. “Fast inference of binary merger properties using the information encoded in the gravitational-wave signal”. In: (Apr. 2023). arXiv: 2304.03731 [gr-qc] (cit. on p. 55).

[299] Vaibhav Tiwari et al. “VARAHA: A Fast Non-Markovian sampler for estimating Gravitational-Wave posteriors”. In: (Mar. 2023). arXiv: 2303.01463 [astro-ph.HE] (cit. on p. 55).

[300] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. ISBN: 9781999579517 (cit. on pp. 59, 60).

[301] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e92 Paper.pdf (cit. on pp. 60, 65).

[302] David Harrison and Daniel L Rubinfeld. “Hedonic housing prices and the demand for clean air”. In: *Journal of Environmental Economics and Management* 5.1 (1978), pp. 81–102. ISSN: 0095-0696. DOI: https://doi.org/10.1016/0095-0696(78)90006-2 (cit. on p. 60).

[303] Sam Bond-Taylor et al. “Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.11 (2022), pp. 7327–7347. DOI: 10.1109/TPAMI.2021.3116668 (cit. on p. 60).

[304] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *arXiv e-prints*, arXiv:1511.06434 (Nov. 2015), arXiv:1511.06434. DOI: 10.48550/arXiv.1511.06434. arXiv: 1511.06434 [cs.LG] (cit. on p. 60).

[305] Guansong Pang et al. "Deep Learning for Anomaly Detection: A Review". In: *ACM Comput. Surv.* 54.2 (Mar. 2021). ISSN: 0360-0300. DOI: 10.1145/3439950 (cit. on p. 60).

[306] Enyan Dai and Jie Chen. "Graph-Augmented Normalizing Flows for Anomaly Detection of Multiple Time Series". In: *arXiv e-prints*, arXiv:2202.07857 (Feb. 2022), arXiv:2202.07857. DOI: 10.48550/arXiv.2202.07857. arXiv: 2202.07857 [cs.LG] (cit. on p. 60).

[307] Absalom E. Ezugwu et al. "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects". In: *Engineering Applications of Artificial Intelligence* 110 (2022), p. 104743. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2022.104743 (cit. on p. 60).

[308] Joydeep Das et al. "Clustering-based recommender system using principles of voting theory". In: *2014 International Conference on Contemporary Computing and Informatics (IC3I)*. 2014, pp. 230–235. DOI: 10.1109/IC3I.2014.7019655 (cit. on p. 60).

[309] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *The Bulletin of Mathematical Biophysics* 5.4 (1943), pp. 115–133. DOI: 10.1007/bf02478259 (cit. on p. 60).

[310] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. ISBN: 026268053X (cit. on pp. 60, 65).

[311] Dan Claudiu Cirean et al. "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". In: *Neural Computation* 22.12 (Dec. 2010), pp. 3207–3220. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00052. eprint: https://direct.mit.edu/neco/article-pdf/22/12/3207/842857/neco_a_00052.pdf (cit. on p. 60).

[312] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988, pp. 696–699. ISBN: 0262010976 (cit. on pp. 60, 63).

[313] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert L. White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2 (1989), pp. 359–366 (cit. on p. 61).

[314] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks”. In: *International Conference on Artificial Intelligence and Statistics*. 2011 (cit. on p. 61).

[315] Andrew L. Maas. “Rectifier Nonlinearities Improve Neural Network Acoustic Models”. In: 2013 (cit. on p. 61).

[316] M.N.H. Siddique and M.O. Tokhi. “Training neural networks: backpropagation vs. genetic algorithms”. In: *IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*. Vol. 4. 2001, 2673–2678 vol.4. DOI: `10.1109/IJCNN.2001.938792` (cit. on p. 63).

[317] Yann Dauphin et al. “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization”. In: *arXiv e-prints*, arXiv:1406.2572 (June 2014), arXiv:1406.2572. arXiv: `1406.2572 [cs.LG]` (cit. on p. 63).

[318] Donald Goldfarb, Yi Ren, and Achraf Bahamou. “Practical Quasi-Newton Methods for Training Deep Neural Networks”. In: *arXiv e-prints*, arXiv:2006.08877 (June 2020), arXiv:2006.08877. arXiv: `2006.08877 [cs.LG]` (cit. on p. 63).

[319] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. “Optimization Methods for Large-Scale Machine Learning”. In: *arXiv e-prints*, arXiv:1606.04838 (June 2016), arXiv:1606.04838. arXiv: `1606.04838 [stat.ML]` (cit. on p. 63).

[320] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv e-prints*, arXiv:1609.04747 (Sept. 2016), arXiv:1609.04747. arXiv: `1609.04747 [cs.LG]` (cit. on p. 63).

[321] Robert Hecht-Nielsen. “Theory of the backpropagation neural network”. In: *International 1989 Joint Conference on Neural Networks* (1989), 593–605 vol.1 (cit. on p. 63).

[322] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256. URL: `https://proceedings.mlr.press/v9/glorot10a.html` (cit. on p. 63).

[323] Meenal V. Narkhede, Prashant P. Bartakke, and Mukul S. Sutaone. “A review on weight initialization strategies for neural networks”. In: *Artificial Intelligence Review* 55.1 (Jan. 2022), pp. 291–322. ISSN: 1573-7462. DOI: 10.1007/s10462-021-10033-z (cit. on p. 64).

[324] Ilya Sutskever et al. “On the Importance of Initialization and Momentum in Deep Learning”. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML’13. Atlanta, GA, USA: JMLR.org, 2013, pp. III1139III–1147 (cit. on p. 64).

[325] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980 (cit. on pp. 64, 89, 222–224).

[326] Herbert Robbins and Sutton Monro. “A Stochastic Approximation Method”. In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. DOI: 10.1214/aoms/1177729586 (cit. on p. 64).

[327] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv e-prints*, arXiv:1409.1556 (Sept. 2014), arXiv:1409.1556. DOI: 10.48550/arXiv.1409.1556. arXiv: 1409.1556 [cs.CV] (cit. on p. 64).

[328] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on pp. 64, 65, 89).

[329] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic Gradient Descent with Warm Restarts”. In: *arXiv e-prints*, arXiv:1608.03983 (Aug. 2016), arXiv:1608.03983. arXiv: 1608.03983 [cs.LG] (cit. on pp. 64, 224).

[330] Kaichao You et al. “How Does Learning Rate Decay Help Modern Neural Networks?” In: *arXiv e-prints*, arXiv:1908.01878 (Aug. 2019), arXiv:1908.01878. arXiv: 1908.01878 [cs.LG] (cit. on p. 64).

[331] Leslie N. Smith. “Cyclical Learning Rates for Training Neural Networks”. In: *arXiv e-prints*, arXiv:1506.01186 (June 2015), arXiv:1506.01186. DOI: 10.48550/arXiv.1506.01186. arXiv: 1506.01186 [cs.CV] (cit. on p. 64).

[332] J. Sola and J. Sevilla. “Importance of input data normalization for the application of neural networks to complex industrial problems”. In: *IEEE Transactions on Nuclear Science* 44.3 (1997), pp. 1464–1468. DOI: 10.1109/23.589532 (cit. on pp. 64, 213).

[333] Luis Perez and Jason Wang. “The Effectiveness of Data Augmentation in Image Classification using Deep Learning”. In: *arXiv e-prints*, arXiv:1712.04621 (Dec. 2017), arXiv:1712.04621. arXiv: 1712.04621 [cs.CV] (cit. on p. 64).

[334] Yann A. LeCun et al. “Efficient BackProp”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_3 (cit. on p. 64).

[335] Yun Xu and Royston Goodacre. “On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning”. In: *Journal of Analysis and Testing* 2 (2018), pp. 249–262 (cit. on p. 64).

[336] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training Recurrent Neural Networks”. In: *arXiv e-prints*, arXiv:1211.5063 (Nov. 2012), arXiv:1211.5063. DOI: 10.48550/arXiv.1211.5063. arXiv: 1211.5063 [cs.LG] (cit. on p. 65).

[337] Kaiming He et al. “Identity Mappings in Deep Residual Networks”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 630–645 (cit. on pp. 65, 89).

[338] Saining Xie et al. “Aggregated Residual Transformations for Deep Neural Networks”. In: *arXiv e-prints*, arXiv:1611.05431 (Nov. 2016), arXiv:1611.05431. DOI: 10.48550/arXiv.1611.05431. arXiv: 1611.05431 [cs.CV] (cit. on p. 65).

[339] Ross Wightman, Hugo Touvron, and Hervé Jégou. “ResNet strikes back: An improved training procedure in timm”. In: *arXiv e-prints*, arXiv:2110.00476 (Oct. 2021), arXiv:2110.00476. DOI: 10.48550/arXiv.2110.00476. arXiv: 2110.00476 [cs.CV] (cit. on p. 65).

[340] Omar Elharrouss et al. “Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches”. In: *arXiv e-prints*, arXiv:2206.0801 (June 2022), arXiv:2206.08016. DOI: 10.48550/arXiv.2206.08016. arXiv: 2206.08016 [cs.CV] (cit. on p. 65).

[341] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: *arXiv e-prints*, arXiv:1406.2661 (June 2014), arXiv:1406.2661. DOI: 10.48550/arXiv.1406.2661. arXiv: 1406.2661 [stat.ML] (cit. on pp. 66, 250).

[342] Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. “Adaptive Monte Carlo augmented with normalizing flows”. In: *Proc. Nat. Acad. Sci.* 119.10 (2022), e2109420119. DOI: 10.1073/pnas.2109420119. arXiv: 2105.12603 [physics.data-an] (cit. on p. 66).

[343] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". In: *arXiv e-prints*, arXiv:1908.09257 (Aug. 2019), arXiv:1908.09257. arXiv: 1908.09257 [stat.ML] (cit. on p. 66).

[344] George Papamakarios et al. "Normalizing Flows for Probabilistic Modeling and Inference". In: *J. Mach. Learn. Res.* 22.1 (July 2022). ISSN: 1532-4435 (cit. on pp. 66, 73).

[345] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear Independent Components Estimation". In: *arXiv e-prints*, arXiv:1410.8516 (Oct. 2014), arXiv:1410.8516. arXiv: 1410.8516 [cs.LG] (cit. on pp. 67, 89).

[346] Diederik P. Kingma and Prafulla Dhariwal. "Glow: Generative Flow with Invertible 1x1 Convolutions". In: *arXiv e-prints*, arXiv:1807.03039 (July 2018), arXiv:1807.03039. arXiv: 1807.03039 [stat.ML] (cit. on pp. 68, 71, 89).

[347] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using Real NVP". In: *arXiv e-prints*, arXiv:1605.08803 (May 2016), arXiv:1605.08803. arXiv: 1605.08803 [cs.LG] (cit. on pp. 68, 69, 71, 89, 154).

[348] Diederik P. Kingma et al. "Improving Variational Inference with Inverse Autoregressive Flow". In: *arXiv e-prints*, arXiv:1606.04934 (June 2016), arXiv:1606.04934. arXiv: 1606.04934 [cs.LG] (cit. on p. 70).

[349] George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked Autoregressive Flow for Density Estimation". In: *arXiv e-prints*, arXiv:1705.07057 (May 2017), arXiv:1705.07057. arXiv: 1705.07057 [stat.ML] (cit. on pp. 71, 76).

[350] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *arXiv e-prints*, arXiv:1502.03167 (Feb. 2015), arXiv:1502.03167. DOI: 10.48550/arXiv.1502.03167. arXiv: 1502.03167 [cs.LG] (cit. on pp. 71, 89).

[351] Ivo Danihelka et al. "Comparison of Maximum Likelihood and GAN-based training of Real NVPs". In: *arXiv e-prints*, arXiv:1705.05263 (May 2017), arXiv:1705.05263. arXiv: 1705.05263 [cs.LG] (cit. on p. 72).

[352] Aditya Grover, Manik Dhar, and Stefano Ermon. "Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models". In: *arXiv e-prints*, arXiv:1705.08868 (May 2017), arXiv:1705.08868. arXiv: 1705.08868 [cs.LG] (cit. on p. 72).

[353] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *arXiv e-prints*, arXiv:1411.1784 (Nov. 2014), arXiv:1411.1784. arXiv: 1411.1784 [cs.LG] (cit. on p. 76).

[354] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. “Learning Structured Output Representation Using Deep Conditional Generative Models”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’15. Montreal, Canada: MIT Press, 2015, pp. 3483–3491 (cit. on p. 76).

[355] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. “The frontier of simulation-based inference”. In: *Proceedings of the National Academy of Science* 117.48 (Dec. 2020), pp. 30055–30062. DOI: `10.1073/pnas.1912789117`. arXiv: `1911.01429 [stat.ML]` (cit. on pp. 76, 153).

[356] George Papamakarios and Iain Murray. “Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation”. In: *arXiv e-prints*, arXiv:1605.06376 (May 2016), arXiv:1605.06376. DOI: `10.48550/arXiv.1605.06376`. arXiv: `1605.06376 [stat.ML]` (cit. on p. 76).

[357] Pavel Izmailov et al. “Semi-Supervised Learning with Normalizing Flows”. In: *arXiv e-prints*, arXiv:1912.13025 (Dec. 2019), arXiv:1912.13025. arXiv: `1912.13025 [cs.LG]` (cit. on p. 76).

[358] Lynton Ardizzone et al. “Training Normalizing Flows with the Information Bottleneck for Competitive Generative Classification”. In: *arXiv e-prints*, arXiv:2001.06448 (Jan. 2020), arXiv:2001.06448. arXiv: `2001.06448 [cs.LG]` (cit. on p. 76).

[359] Paul Hagemann and Sebastian Neumayer. “Stabilizing invertible neural networks using mixture models”. In: *Inverse Problems* 37.8, 085002 (Aug. 2021), p. 085002. DOI: `10.1088/1361-6420/abe928`. arXiv: `2009.02994 [cs.LG]` (cit. on p. 76).

[360] Johann Brehmer and Kyle Cranmer. “Flows for simultaneous manifold learning and density estimation”. In: *arXiv e-prints*, arXiv:2003.13913 (Mar. 2020), arXiv:2003.13913. arXiv: `2003.13913 [stat.ML]` (cit. on pp. 76, 103).

[361] Brendan Leigh Ross and Jesse C. Cresswell. “Tractable Density Estimation on Learned Manifolds with Conformal Embedding Flows”. In: *arXiv e-prints*, arXiv:2106.05275 (June 2021), arXiv:2106.05275. arXiv: `2106.05275 [stat.ML]` (cit. on p. 76).

[362] Chin-Wei Huang, Laurent Dinh, and Aaron Courville. “Augmented Normalizing Flows: Bridging the Gap Between Generative Flows and Latent Variable Models”. In: *arXiv e-prints*, arXiv:2002.07101 (Feb. 2020), arXiv:2002.07101. arXiv: `2002.07101 [cs.LG]` (cit. on p. 76).

[363] Didrik Nielsen et al. “SurVAE Flows: Surjections to Bridge the Gap between VAEs and Flows”. In: *arXiv e-prints*, arXiv:2007.02731 (July 2020), arXiv:2007.02731. arXiv: `2007.02731 [cs.LG]` (cit. on p. 76).

[364] Hao Wu, Jonas Köhler, and Frank Noé. “Stochastic Normalizing Flows”. In: *arXiv e-prints*, arXiv:2002.06707 (Feb. 2020), arXiv:2002.06707. arXiv: 2002.06707 [stat.ML] (cit. on p. 76).

[365] Vincent Stimper, Bernhard Schölkopf, and José Miguel Hernández-Lobato. “Resampling Base Distributions of Normalizing Flows”. In: *arXiv e-prints*, arXiv:2110.15828 (Oct. 2021), arXiv:2110.15828. arXiv: 2110.15828 [stat.ML] (cit. on p. 76).

[366] Chin-Wei Huang et al. “Neural Autoregressive Flows”. In: *arXiv e-prints*, arXiv:1804.00779 (Apr. 2018), arXiv:1804.00779. arXiv: 1804.00779 [cs.LG] (cit. on p. 77).

[367] Thomas Müller et al. “Neural Importance Sampling”. In: *arXiv e-prints*, arXiv:1808.03856 (Aug. 2018), arXiv:1808.03856. arXiv: 1808.03856 [cs.LG] (cit. on p. 77).

[368] Zachary M. Ziegler and Alexander M. Rush. “Latent Normalizing Flows for Discrete Sequences”. In: *arXiv e-prints*, arXiv:1901.10548 (Jan. 2019), arXiv:1901.10548. arXiv: 1901.10548 [stat.ML] (cit. on p. 77).

[369] Jonathan Ho et al. “Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design”. In: *arXiv e-prints*, arXiv:1902.00275 (Feb. 2019), arXiv:1902.00275. arXiv: 1902.00275 [cs.LG] (cit. on p. 77).

[370] Conor Durkan et al. “Neural Spline Flows”. In: *arXiv e-prints*, arXiv:1906.04032 (June 2019), arXiv:1906.04032. arXiv: 1906.04032 [stat.ML] (cit. on pp. 77, 103, 154).

[371] Elena Cuoco et al. “Enhancing Gravitational-Wave Science with Machine Learning”. In: *Mach. Learn. Sci. Tech.* 2.1 (2021), p. 011002. DOI: 10.1088/2632-2153/abb93a. arXiv: 2005.03745 [astro-ph.HE] (cit. on p. 77).

[372] Filip Morawski et al. “Anomaly detection in gravitational waves data using convolutional autoencoders”. In: *Mach. Learn. Sci. Tech.* 2.4 (2021), p. 045014. DOI: 10.1088/2632-2153/abf3d0. arXiv: 2103.07688 [astro-ph.IM] (cit. on p. 77).

[373] Michael Zevin et al. “Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science”. In: *Class. Quant. Grav.* 34.6 (2017), p. 064003. DOI: 10.1088/1361-6382/aa5cea. arXiv: 1611.04596 [gr-qc] (cit. on p. 77).

[374] J. Glanzer et al. “Data quality up to the third observing run of Advanced LIGO: Gravity Spy glitch classifications”. In: (Aug. 2022). arXiv: 2208.12849 [gr-qc] (cit. on p. 77).

[375] R. Essick, L. Blackburn, and E. Katsavounidis. “Optimizing Vetoes for Gravitational-Wave Transient Searches”. In: *Class. Quant. Grav.* 30 (2013), p. 155010. DOI: 10.1088/0264-9381/30/15/155010. arXiv: 1303.7159 [astro-ph.IM] (cit. on p. 77).

[376] Rahul Biswas et al. “Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data”. In: *Phys. Rev. D* 88.6 (2013), p. 062003. DOI: 10.1103/PhysRevD.88.062003. arXiv: 1303.6984 [astro-ph.IM] (cit. on p. 77).

[377] Reed Essick et al. “iDQ: Statistical inference of non-gaussian noise with auxiliary degrees of freedom in gravitational-wave detectors”. In: *Machine Learning: Science and Technology* 2.1 (Dec. 2020), p. 015004. DOI: 10.1088/2632-2153/abab5f (cit. on p. 77).

[378] Hongyu Shen et al. “Denoising Gravitational Waves with Enhanced Deep Recurrent Denoising Auto-encoders”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 3237–3241. DOI: 10.1109/ICASSP.2019.8683061 (cit. on p. 77).

[379] Kentaro Mogushi et al. “NNETFIX: an artificial neural network-based denoising engine for gravitational-wave signals”. In: *Mach. Learn. Sci. Tech.* 2.3 (2021), p. 035018. DOI: 10.1088/2632-2153/abea69. arXiv: 2101.04712 [gr-qc] (cit. on p. 77).

[380] Sebastian Khan and Rhys Green. “Gravitational-wave surrogate models powered by artificial neural networks”. In: *Phys. Rev. D* 103.6 (2021), p. 064015. DOI: 10.1103/PhysRevD.103.064015. arXiv: 2008.12932 [gr-qc] (cit. on p. 78).

[381] Lucy M. Thomas, Geraint Pratten, and Patricia Schmidt. “Accelerating multimodal gravitational waveforms from precessing compact binaries with artificial neural networks”. In: *Phys. Rev. D* 106.10 (2022), p. 104029. DOI: 10.1103/PhysRevD.106.104029. arXiv: 2205.14066 [gr-qc] (cit. on p. 78).

[382] Kaze W. K. Wong, Maximiliano Isi, and Thomas D. P. Edwards. “Fast gravitational wave parameter estimation without compromises”. In: (Feb. 2023). arXiv: 2302.05333 [astro-ph.IM] (cit. on p. 78).

[383] Daniel Williams et al. “Precessing numerical relativity waveform surrogate model for binary black holes: A Gaussian process regression approach”. In: *Phys. Rev. D* 101.6 (2020), p. 063011. DOI: 10.1103/PhysRevD.101.063011. arXiv: 1903.09204 [gr-qc] (cit. on p. 78).

[384] J. McGinn et al. “Generalised gravitational wave burst generation with generative adversarial networks”. In: *Class. Quant. Grav.* 38.15 (2021), p. 155005. DOI: 10.1088/1361-6382/ac09cc. arXiv: 2103.01641 [astro-ph.IM] (cit. on p. 78).

[385] Melissa Lopez et al. “Simulating transient noise bursts in LIGO with generative adversarial networks”. In: *Physical Review D* 106.2, 023027 (July 2022), p. 023027. DOI: 10.1103/PhysRevD.106.023027. arXiv: 2203.06494 [astro-ph.IM] (cit. on p. 78).

[386] Jade Powell et al. “Generating transient noise artefacts in gravitational-wave detector data with generative adversarial networks”. In: *Class. Quant. Grav.* 40.3 (2023), p. 035006. DOI: 10.1088/1361-6382/acb038. arXiv: 2207.00207 [astro-ph.IM] (cit. on p. 78).

[387] Hunter Gabbard et al. “Matching matched filtering with deep networks for gravitational-wave astronomy”. In: *Phys. Rev. Lett.* 120.14 (2018), p. 141103. DOI: 10.1103/PhysRevLett.120.141103. arXiv: 1712.06041 [astro-ph.IM] (cit. on p. 78).

[388] Daniel George and E. A. Huerta. “Deep Neural Networks to Enable Real-time Multimessenger Astrophysics”. In: *Phys. Rev. D* 97.4 (2018), p. 044039. DOI: 10.1103/PhysRevD.97.044039. arXiv: 1701.00008 [astro-ph.IM] (cit. on p. 78).

[389] Marlin B. Schäfer, Frank Ohme, and Alexander H. Nitz. “Detection of gravitational-wave signals from binary neutron star mergers using machine learning”. In: *Phys. Rev. D* 102.6 (2020), p. 063015. DOI: 10.1103/PhysRevD.102.063015. arXiv: 2006.01509 [astro-ph.HE] (cit. on p. 78).

[390] Marlin B. Schäfer et al. “First machine learning gravitational-wave search mock data challenge”. In: *Phys. Rev. D* 107.2 (2023), p. 023021. DOI: 10.1103/PhysRevD.107.023021. arXiv: 2209.11146 [astro-ph.IM] (cit. on p. 78).

[391] Paraskevi Nousi et al. “Deep Residual Networks for Gravitational Wave Detection”. In: (Nov. 2022). arXiv: 2211.01520 [gr-qc] (cit. on p. 78).

[392] V. Gayathri et al. “Enhancing the sensitivity of transient gravitational wave searches with Gaussian mixture models”. In: *Phys. Rev. D* 102.10 (2020), p. 104023. DOI: 10.1103/PhysRevD.102.104023. arXiv: 2008.01262 [gr-qc] (cit. on p. 78).

[393] Tanmaya Mishra et al. “Optimization of model independent gravitational wave search for binary black hole mergers using machine learning”. In: *Phys. Rev. D* 104.2 (2021), p. 023014. DOI: 10.1103/PhysRevD.104.023014. arXiv: 2105.04739 [gr-qc] (cit. on p. 78).

[394] T. Mishra et al. “Search for binary black hole mergers in the third observing run of Advanced LIGO-Virgo using coherent WaveBurst enhanced with machine learning”. In: *Phys. Rev. D* 105.8 (2022), p. 083018. DOI: 10.1103/PhysRevD.105.083018. arXiv: 2201.01495 [gr-qc] (cit. on p. 78).

[395] Marek J. Szczepaczyk et al. “Search for gravitational-wave bursts in the third Advanced LIGO-Virgo run with coherent WaveBurst enhanced by Machine Learning”. In: (Oct. 2022). arXiv: 2210.01754 [gr-qc] (cit. on p. 78).

[396] P. Astone et al. “New method to observe gravitational waves emitted by core collapse supernovae”. In: *Phys. Rev. D* 98.12 (2018), p. 122002. DOI: 10.1103/PhysRevD.98.122002. arXiv: 1812.05363 [astro-ph.IM] (cit. on p. 78).

[397] Marco Cavaglia et al. “Improving the background of gravitational-wave searches for core collapse supernovae: A machine learning approach”. In: *Mach. Learn. Sci. Tech.* 1 (2020), p. 015005. DOI: 10.1088/2632-2153/ab527d. arXiv: 2002.04591 [astro-ph.IM] (cit. on p. 78).

[398] Man Leong Chan, Ik Siong Heng, and Chris Messenger. “Detection and classification of supernova gravitational wave signals: A deep learning approach”. In: *Phys. Rev. D* 102.4 (2020), p. 043022. DOI: 10.1103/PhysRevD.102.043022. arXiv: 1912.13517 [astro-ph.HE] (cit. on p. 78).

[399] M. López Portilla et al. “Deep learning for core-collapse supernova detection”. In: *Phys. Rev. D* 103.6 (2021), p. 063011. DOI: 10.1103/PhysRevD.103.063011. arXiv: 2011.13733 [astro-ph.IM] (cit. on p. 78).

[400] Tom Marianer, Dovi Poznanski, and J Xavier Prochaska. “A semisupervised machine learning search for never-seen gravitational-wave sources”. In: *Monthly Notices of the Royal Astronomical Society* 500.4 (Nov. 2020), pp. 5408–5419. ISSN: 0035-8711. DOI: 10.1093/mnras/staa3550. eprint: https://academic.oup.com/mnras/article-pdf/500/4/5408/34925881/staa3550.pdf (cit. on p. 78).

[401] Vincent Boudart and Maxime Fays. “Machine learning algorithm for minute-long burst searches”. In: *Phys. Rev. D* 105.8 (2022), p. 083007. DOI: 10.1103/PhysRevD.105.083007. arXiv: 2201.08727 [gr-qc] (cit. on p. 78).

[402] Christoph Dreissigacker et al. “Deep-Learning Continuous Gravitational Waves”. In: *Phys. Rev. D* 100.4 (2019), p. 044009. DOI: 10.1103/PhysRevD.100.044009. arXiv: 1904.13291 [gr-qc] (cit. on p. 79).

[403] Christoph Dreissigacker and Reinhard Prix. “Deep-Learning Continuous Gravitational Waves: Multiple detectors and realistic noise”. In: *Phys. Rev. D* 102.2 (2020), p. 022005. DOI: 10.1103/PhysRevD.102.022005. arXiv: 2005.04140 [gr-qc] (cit. on p. 79).

[404] Prasanna M. Joshi and Reinhard Prix. “A novel neural-network architecture for continuous gravitational waves”. In: (May 2023). arXiv: 2305.01057 [gr-qc] (cit. on p. 79).

[405] Banafsheh Beheshtipour and Maria Alessandra Papa. "Deep learning for clustering of continuous gravitational wave candidates". In: *Phys. Rev. D* 101.6 (2020), p. 064009. DOI: 10.1103/PhysRevD.101.064009. arXiv: 2001.03116 [gr-qc] (cit. on p. 79).

[406] B. Beheshtipour and M. A. Papa. "Deep learning for clustering of continuous gravitational wave candidates II: identification of low-SNR candidates". In: *Phys. Rev. D* 103.6 (2021), p. 064027. DOI: 10.1103/PhysRevD.103.064027. arXiv: 2012.04381 [gr-qc] (cit. on p. 79).

[407] Joe Bayley, Graham Woan, and Chris Messenger. "Generalized application of the Viterbi algorithm to searches for continuous gravitational-wave signals". In: *Phys. Rev. D* 100.2 (2019), p. 023006. DOI: 10.1103/PhysRevD.100.023006. arXiv: 1903.12614 [astro-ph.IM] (cit. on p. 79).

[408] Joseph Bayley, Chris Messenger, and Graham Woan. "Robust machine learning algorithm to search for continuous gravitational waves". In: *Phys. Rev. D* 102.8 (2020), p. 083024. DOI: 10.1103/PhysRevD.102.083024. arXiv: 2007.08207 [astro-ph.IM] (cit. on p. 79).

[409] Deep Chatterjee et al. "A Machine Learning Based Source Property Inference for Compact Binary Mergers". In: *Astrophys. J.* 896.1 (2020), p. 54. DOI: 10.3847/1538-4357/ab8dbe. arXiv: 1911.00116 [astro-ph.IM] (cit. on p. 79).

[410] Philip Graff et al. "BAMBI: blind accelerated multimodal Bayesian inference". In: *Mon. Not. Roy. Astron. Soc* 421.1 (Mar. 2012), pp. 169–180. DOI: 10.1111/j.1365-2966.2011.20288.x. arXiv: 1110.2997 [astro-ph.IM] (cit. on pp. 79, 90, 142).

[411] Stephen R. Green and Jonathan Gair. "Complete parameter inference for GW150914 using deep learning". In: *Mach. Learn. Sci. Tech.* 2.3 (2021), 03LT01. DOI: 10.1088/2632-2153/abfaed. arXiv: 2008.03312 [astro-ph.IM] (cit. on pp. 79, 82, 90, 102).

[412] Stephen R. Green, Christine Simpson, and Jonathan Gair. "Gravitational-wave parameter estimation with autoregressive neural network flows". In: *Phys. Rev. D* 102.10 (2020), p. 104057. DOI: 10.1103/PhysRevD.102.104057. arXiv: 2002.07656 [astro-ph.IM] (cit. on pp. 79, 82, 90, 102, 153).

[413] Maximilian Dax et al. "Group equivariant neural posterior estimation". In: (Nov. 2021). arXiv: 2111.13139 [cs.LG] (cit. on pp. 79, 82).

[414] Maximilian Dax et al. "Real-Time Gravitational Wave Science with Neural Posterior Estimation". In: *Phys. Rev. Lett.* 127.24 (2021), p. 241103. DOI: 10.1103/PhysRevLett.127.241103. arXiv: 2106.12594 [gr-qc] (cit. on pp. 79, 82, 153).

[415] Jurriaan Langendorff et al. “Normalizing Flows as an Avenue to Studying Overlapping Gravitational Wave Signals”. In: *Phys. Rev. Lett.* 130.17 (2023), p. 171402. DOI: 10.1103/PhysRevLett.130.171402. arXiv: 2211.15097 [gr-qc] (cit. on pp. 79, 82).

[416] Alvin J. K. Chua and Michele Vallisneri. “Learning Bayesian posteriors with neural networks for gravitational-wave inference”. In: *Phys. Rev. Lett.* 124.4 (2020), p. 041102. DOI: 10.1103/PhysRevLett.124.041102. arXiv: 1909.05966 [gr-qc] (cit. on pp. 79, 82, 90, 102, 153).

[417] Arnaud Delaunoy et al. “Lightning-Fast Gravitational Wave Parameter Inference through Neural Amortization”. In: (Oct. 2020). arXiv: 2010.12931 [astro-ph.IM] (cit. on p. 79).

[418] Colm Talbot and Eric Thrane. “Flexible and Accurate Evaluation of Gravitational-wave Malmquist Bias with Machine Learning”. In: *Astrophys. J.* 927.1 (2022), p. 76. DOI: 10.3847/1538-4357/ac4bc0. arXiv: 2012.01317 [gr-qc] (cit. on p. 79).

[419] Kaze W. K. Wong, Gabriella Contardo, and Shirley Ho. “Gravitational-wave population inference with deep flow-based generative network”. In: *Phys. Rev. D* 101 (12 June 2020), p. 123005. DOI: 10.1103/PhysRevD.101.123005 (cit. on p. 79).

[420] David Ruhe et al. “Normalizing Flows for Hierarchical Bayesian Analysis: A Gravitational Wave Population Study”. In: (Nov. 2022). arXiv: 2211.09008 [astro-ph.IM] (cit. on p. 79).

[421] Kaze W. K. Wong and Miles Cranmer. “Automated discovery of interpretable gravitational-wave population models”. In: *39th International Conference on Machine Learning Conference*. July 2022. arXiv: 2207.12409 [astro-ph.IM] (cit. on p. 79).

[422] J. S. Bloom et al. “Automating Discovery and Classification of Transients and Variable Stars in the Synoptic Survey Era”. In: *Publications of the Astronomical Society of the Pacific* 124.921 (Oct. 2012), p. 1175. DOI: 10.1086/668468 (cit. on p. 82).

[423] Mike Walmsley et al. “Galaxy Zoo: probabilistic morphology through Bayesian CNNs and active learning”. In: *Monthly Notices of the Royal Astronomical Society* 491.2 (Oct. 2019), pp. 1554–1574. DOI: 10.1093/mnras/stz2816 (cit. on p. 82).

[424] Christopher M. J. Osborne, John A. Armstrong, and Lyndsay Fletcher. “RADYNVERSION: Learning to Invert a Solar Flare Atmosphere with Invertible Neural Networks”. In: *The Astrophysical Journal* 873.2 (Mar. 2019), p. 128. DOI: 10.3847/1538-4357/ab07b4 (cit. on p. 82).

[425] Claudionor N. Coelho et al. “Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors”. In: *Nature Mach. Intell.* 3 (2021), pp. 675–686. DOI: 10.1038/s42256-021-00356-5. arXiv: 2006.10159 [physics.ins-det] (cit. on p. 82).

[426] George Em Karniadakis et al. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (May 2021). ISSN: 2522-5820. DOI: 10.1038/s42254-021-00314-5 (cit. on p. 82).

[427] Mervin E. Muller. “A Note on a Method for Generating Points Uniformly on N-Dimensional Spheres”. In: *Commun. ACM* 2.4 (Apr. 1959), pp. 19–20. ISSN: 0001-0782. DOI: 10.1145/377939.377946 (cit. on pp. 85, 126).

[428] George Marsaglia. “Choosing a Point from the Surface of a Sphere”. In: *Annals of Mathematical Statistics* 43 (1972), pp. 645–646 (cit. on pp. 85, 126).

[429] Michael J. Williams. *Nessai Documentation.* 2023. URL: https://nessai.readthedocs.io/ (cit. on pp. 87, 89, 98, 217, 229, 247).

[430] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32.* Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf (cit. on pp. 89, 247).

[431] Conor Durkan et al. *nflows: normalizing flows in PyTorch.* Version v0.14. Nov. 2020. DOI: 10.5281/zenodo.4296287 (cit. on pp. 89, 247).

[432] Michael J. Williams. *mj-will/nessai.* Version latest. DOI: 10.5281/zenodo.4550693 (cit. on pp. 89, 153, 247).

[433] Daniel Levy, Matthew D. Hoffman, and Jascha Sohl-Dickstein. “Generalizing Hamiltonian Monte Carlo with Neural Networks”. In: *arXiv e-prints*, arXiv:1711.09268 (Nov. 2017), arXiv:1711.09268. DOI: 10.48550/arXiv.1711.09268. arXiv: 1711.09268 [stat.ML] (cit. on p. 90).

[434] Matthew Hoffman et al. “NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport”. In: *arXiv e-prints*, arXiv:1903.03704 (Mar. 2019), arXiv:1903.03704. DOI: 10.48550/arXiv.1903.03704. arXiv: 1903.03704 [stat.CO] (cit. on p. 90).

[435] Adam Moss. “Accelerated Bayesian inference using deep learning”. In: *Mon. Not. Roy. Astron. Soc.* 496.1 (2020), pp. 328–338. DOI: 10.1093/mnras/staa1469. arXiv: 1903.10860 [astro-ph.CO] (cit. on pp. 90, 142, 153, 154, 220, 221).

[436] Benjamin P. Abbott et al. "The basic physics of the binary black hole merger GW150914". In: *Annalen Phys.* 529.1-2 (2017), p. 1600209. DOI: 10.1002/andp.201600209. arXiv: 1608.01940 [gr-qc] (cit. on pp. 90, 91).

[437] Patricia Schmidt, Mark Hannam, and Sascha Husa. "Towards models of gravitational waveforms from generic binaries: A simple approximate mapping between precessing and non-precessing inspiral signals". In: *Phys. Rev. D* 86 (2012), p. 104063. DOI: 10.1103/PhysRevD.86.104063. arXiv: 1207.3088 [gr-qc] (cit. on pp. 91, 159).

[438] Michael J. Williams. *Thesis Code*. 2023 (cit. on pp. 107, 222, 247).

[439] Michael J. Williams. *mj-will/nessai-models*. DOI: 10.5281/zenodo.7105559 (cit. on pp. 107, 220, 247).

[440] Yun-Wei Shang and Yu-Huang Qiu. "A Note on the Extended Rosenbrock Function". In: *Evolutionary Computation* 14.1 (Mar. 2006), pp. 119–126. ISSN: 1063-6560. DOI: 10.1162/evco.2006.14.1.119. eprint: https://direct.mit.edu/evco/article-pdf/14/1/119/1493646/evco.2006.14.1.119.pdf (cit. on pp. 110, 157, 220).

[441] J. H. Conway and D. A. Smith. *On Quaternions and Octonions*. New York: A K Peters/CRC Press. ISBN: 9780429185861. DOI: 10.1201/9781439864180 (cit. on pp. 124, 125).

[442] Wm. R. Hamilton. "Theory of Quaternions". In: *Proceedings of the Royal Irish Academy (1836-1869)* 3 (1844), pp. 1–16. ISSN: 03027597. URL: http://www.jstor.org/stable/20489494 (visited on 06/13/2023) (cit. on p. 124).

[443] Jean Gallier. "The Quaternions and the Spaces S3, SU(2), SO(3), and $\mathbb{R}P3$". In: *Geometric Methods and Applications: For Computer Science and Engineering*. New York, NY: Springer New York, 2001, pp. 248–266. ISBN: 978-1-4613-0137-0. DOI: 10.1007/978-1-4613-0137-0_8 (cit. on p. 124).

[444] D. M. Henderson. *Euler angles, quaternions, and transformation matrices for space shuttle analysis*. Tech. rep. DN-1.4-8-020. NASA, June 1977 (cit. on p. 124).

[445] "IEEE Standard for Floating-Point Arithmetic". In: *IEEE Std 754-2019 (Revision of IEEE 754-2008)* (2019), pp. 1–84. DOI: 10.1109/IEEESTD.2019.8766229 (cit. on p. 125).

[446] Justin Alsing and Will Handley. "Nested sampling with any prior you like". In: *Mon. Not. Roy. Astron. Soc.* 505.1 (2021), pp. L95–L99. DOI: 10.1093/mnrasl/slab057. arXiv: 2102.12478 [astro-ph.IM] (cit. on p. 142).

[447] Heiko Zimmermann et al. "Nested Variational Inference". In: *arXiv e-prints*, arXiv:2106.1130 (June 2021), arXiv:2106.11302. DOI: 10.48550/arXiv.2106.11302. arXiv: 2106.11302 [stat.ML] (cit. on p. 153).

[448] Michael Arbel, Alexander G. D. G. Matthews, and Arnaud Doucet. "Annealed Flow Transport Monte Carlo". In: *arXiv e-prints*, arXiv:2102.07501 (Feb. 2021), arXiv:2102.07501. DOI: 10.48550/arXiv.2102.07501. arXiv: 2102.07501 [stat.ML] (cit. on p. 153).

[449] Minas Karamanis et al. "Accelerating astronomical and cosmological inference with preconditioned Monte Carlo". In: *Mon. Not. Roy. Astron. Soc.* 516.2 (2022), pp. 1644–1653. DOI: 10.1093/mnras/stac2272. arXiv: 2207.05652 [astro-ph.IM] (cit. on p. 153).

[450] Justin Alsing et al. "Fast likelihood-free cosmology with neural density estimators and active learning". In: *Mon. Not. Roy. Astron. Soc.* 488.3 (2019), pp. 4440–4458. DOI: 10.1093/mnras/stz1960. arXiv: 1903.00007 [astro-ph.CO] (cit. on p. 153).

[451] Niall Jeffrey, Justin Alsing, and François Lanusse. "Likelihood-free inference with neural compression of DES SV weak lensing map statistics". In: *Mon. Not. Roy. Astron. Soc.* 501.1 (2021), pp. 954–969. DOI: 10.1093/mnras/staa3594. arXiv: 2009.08459 [astro-ph.CO] (cit. on p. 153).

[452] Johann Brehmer and Kyle Cranmer. "Simulation-based inference methods for particle physics". In: (Oct. 2020). arXiv: 2010.06439 [hep-ph] (cit. on p. 153).

[453] Michael J. Williams. *mj-will/nessai-ins-paper: Accepted version*. Version v1.0.0. July 2023. DOI: 10.5281/zenodo.8124198 (cit. on pp. 154, 247).

[454] H. H. Rosenbrock. "An Automatic Method for Finding the Greatest or Least Value of a Function". In: *The Computer Journal* 3.3 (Jan. 1960), pp. 175–184. ISSN: 0010-4620. DOI: 10.1093/comjnl/3.3.175. eprint: https://academic.oup.com/comjnl/article-pdf/3/3/175/988633/030175.pdf (cit. on pp. 157, 220).

[455] David E. Goldberg and John H. Holland. "Genetic Algorithms and Machine Learning". In: *Machine Learning* 3.2 (Oct. 1988), pp. 95–99. ISSN: 1573-0565. DOI: 10.1023/A:1022602019183 (cit. on pp. 157, 220).

[456] Tim Dietrich et al. "Improving the NRTidal model for binary neutron star systems". In: *Phys. Rev. D* 100.4 (2019), p. 044003. DOI: 10.1103/PhysRevD.100.044003. arXiv: 1905.06011 [gr-qc] (cit. on pp. 159, 188, 192).

[457] Sascha Husa et al. "Frequency-domain gravitational waves from nonprecessing black-hole binaries. I. New numerical waveforms and anatomy of the signal". In: *Phys. Rev. D* 93.4 (2016), p. 044006. DOI: 10.1103/PhysRevD.93.044006. arXiv: 1508.07250 [gr-qc] (cit. on pp. 159, 170).

[458] Sebastian Khan et al. "Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era". In: *Phys. Rev. D* 93.4 (2016), p. 044007. DOI: 10.1103/PhysRevD.93.044007. arXiv: 1508.07253 [gr-qc] (cit. on pp. 159, 170).

[459] Tyson B. Littenberg et al. "Fortifying the characterization of binary mergers in LIGO data". In: *Physical Review D* 88.8, 084044 (Oct. 2013), p. 084044. DOI: 10.1103/PhysRevD.88.084044. arXiv: 1307.8195 [astro-ph.IM] (cit. on p. 168).

[460] Will M. Farr, Benjamin Farr, and Tyson Littenberg. *Modelling Calibration Errors In CBC Waveforms.* Tech. rep. T1400682. LIGO, Oct. 2014 (cit. on p. 168).

[461] Aaron Viets et al. "Reconstructing the calibrated strain signal in the Advanced LIGO detectors". In: *Class. Quant. Grav.* 35.9 (2018), p. 095015. DOI: 10.1088/1361-6382/aab658. arXiv: 1710.09973 [astro-ph.IM] (cit. on p. 168).

[462] Craig Cahillane et al. "Calibration uncertainty for Advanced LIGO's first and second observing runs". In: *Physical Review D* 96.10, 102001 (Nov. 2017), p. 102001. DOI: 10.1103/PhysRevD.96.102001. arXiv: 1708.03023 [astro-ph.IM] (cit. on p. 168).

[463] Salvatore Vitale et al. "Physical approach to the marginalization of LIGO calibration uncertainties". In: *Phys. Rev. D* 103.6 (2021), p. 063016. DOI: 10.1103/PhysRevD.103.063016. arXiv: 2009.10192 [gr-qc] (cit. on p. 169).

[464] Ethan Payne et al. "Gravitational-wave astronomy with a physical calibration model". In: *Phys. Rev. D* 102 (2020), p. 122004. DOI: 10.1103/PhysRevD.102.122004. arXiv: 2009.10193 [astro-ph.IM] (cit. on p. 169).

[465] Salvatore Vitale et al. "Effect of calibration errors on Bayesian parameter estimation for gravitational wave signals from inspiral binary systems in the Advanced Detectors era". In: *Phys. Rev. D* 85 (2012), p. 064034. DOI: 10.1103/PhysRevD.85.064034. arXiv: 1111.3044 [gr-qc] (cit. on p. 170).

[466] LIGO Scientific Collaboration and Virgo Collaboration. *Parameter estimation sample release for GWTC-1.* Version v5. LIGO DCC, 2020. DOI: 10.5281/zenodo.6513631 (cit. on pp. 173, 188, 207).

[467] LIGO Scientific Collaboration and Virgo Collaboration. *GWTC-2.1: Deep Extended Catalog of Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run - Parameter Estimation Data Release.* Version v2. Zenodo, May 2022. DOI: 10.5281/zenodo.6513631 (cit. on pp. 175, 185, 191, 195, 243).

[468] Marc Favata. “Systematic Parameter Errors in Inspiraling Neutron Star Binaries”. In: *Phys. Rev. Lett.* 112 (10 Mar. 2014), p. 101101. DOI: 10.1103/PhysRevLett.112.101101 (cit. on pp. 188, 226).

[469] Leslie Wade et al. “Systematic and statistical errors in a bayesian approach to the estimation of the neutron-star equation of state using advanced gravitational wave detectors”. In: *Phys. Rev. D* 89.10 (2014), p. 103012. DOI: 10.1103/PhysRevD.89.103012. arXiv: 1402.5156 [gr-qc] (cit. on pp. 188, 226).

[470] Talya Klinger and Michalis Agathos. “An Order Statistics Post-Mortem on LIGO-Virgo GWTC-2 Events Analysed with Nested Sampling”. In: *Annalen Phys.* 2022 (Sept. 2022), p. 2200271. DOI: 10.1002/andp.202200271. arXiv: 2210.00110 [gr-qc] (cit. on p. 190).

[471] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration. *GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run Parameter estimation data release*. Zenodo, Nov. 2021. DOI: 10.5281/zenodo.5546663 (cit. on p. 191).

[472] R. Abbott et al. “GW190412: Observation of a Binary-Black-Hole Coalescence with Asymmetric Masses”. In: *Phys. Rev. D* 102.4 (2020), p. 043015. DOI: 10.1103/PhysRevD.102.043015. arXiv: 2004.08342 [astro-ph.HE] (cit. on pp. 191, 250).

[473] Benjamin P Abbott et al. “Exploring the Sensitivity of Next Generation Gravitational Wave Detectors”. In: *Class. Quant. Grav.* 34.4 (2017), p. 044001. DOI: 10.1088/1361-6382/aa51f4. arXiv: 1607.08697 [astro-ph.IM] (cit. on p. 203).

[474] Divyajyoti et al. “Detectability of gravitational higher order modes in the third-generation era”. In: *Phys. Rev. D* 104.8 (2021), p. 084080. DOI: 10.1103/PhysRevD.104.084080. arXiv: 2103.03241 [gr-qc] (cit. on p. 203).

[475] Michael Pürrer and Carl-Johan Haster. “Gravitational waveform accuracy requirements for future ground-based detectors”. In: *Phys. Rev. Res.* 2.2 (2020), p. 023151. DOI: 10.1103/PhysRevResearch.2.023151. arXiv: 1912.10055 [gr-qc] (cit. on p. 203).

[476] S. Hild et al. “Sensitivity Studies for Third-Generation Gravitational Wave Observatories”. In: *Class. Quant. Grav.* 28 (2011), p. 094013. DOI: 10.1088/0264-9381/28/9/094013. arXiv: 1012.0908 [gr-qc] (cit. on p. 203).

[477] Rhys Green et al. “Identifying when Precession can be Measured in Gravitational Waveforms”. In: *Phys. Rev. D* 103.12 (2021), p. 124023. DOI: 10.1103/PhysRevD.103.124023. arXiv: 2010.04131 [gr-qc] (cit. on p. 203).

[478] Deyan P. Mihaylov et al. "pySEOBNR: a software package for the next generation of effective-one-body multipolar waveform models". In: (Mar. 2023). arXiv: 2303.18203 [gr-qc] (cit. on p. 217).

[479] Joan-René Mérou et al. *Analysing long GW transients from newborn neutron stars using JAX*. Tech. rep. G2300320. https://dcc.ligo.org/LIGO-G2300320. LIGO, Mar. 2023 (cit. on p. 217).

[480] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs.* Version 0.3.13. 2018. URL: http://github.com/google/jax (cit. on p. 217).

[481] Reinhard Prix, Stefanos Giampanis, and Chris Messenger. "Search method for long-duration gravitational-wave transients from neutron stars". In: *Phys. Rev. D* 84 (2011), p. 023007. DOI: 10.1103/PhysRevD.84.023007. arXiv: 1104.1704 [gr-qc] (cit. on p. 217).

[482] David Keitel et al. "First search for long-duration transient gravitational waves after glitches in the Vela and Crab pulsars". In: *Phys. Rev. D* 100.6 (2019), p. 064058. DOI: 10.1103/PhysRevD.100.064058. arXiv: 1907.04717 [gr-qc] (cit. on p. 217).

[483] Rodrigo Tenorio and Michael J. Williams. "Continuous gravitational-wave follow-up using nessai ". *In prep.* (cit. on p. 217).

[484] Rodrigo Tenorio, David Keitel, and Alicia M. Sintes. "Application of a hierarchical MCMC follow-up to Advanced LIGO continuous gravitational-wave candidates". In: *Phys. Rev. D* 104.8 (2021), p. 084012. DOI: 10.1103/PhysRevD.104.084012. arXiv: 2105.13860 [gr-qc] (cit. on p. 217).

[485] Jethro Linley. "Probing the general dynamics of compact binary systems with multi-band gravitational-wave observations". PhD thesis. University of Glasgow, 2022 (cit. on p. 217).

[486] Eliot Finch et al. "Identifying LISA verification binaries amongst the Galactic population of double white dwarfs". In: (Oct. 2022). arXiv: 2210.10812 [astro-ph.SR] (cit. on pp. 217, 219).

[487] Christian E. A. Chapman-Bird, Christopher P. L. Berry, and Graham Woan. "Rapid determination of LISA sensitivity to extreme mass ratio inspirals with machine learning". In: (Dec. 2022). arXiv: 2212.06166 [astro-ph.HE] (cit. on p. 217).

[488] Fergus Hayes et al. "Unpacking merger jets: a Bayesian analysis of GW170817, GW190425 and electromagnetic observations of short gamma-ray bursts". In: (May 2023). arXiv: 2305.06275 [astro-ph.HE] (cit. on p. 217).

[489] Laurence Élise Hélène Datrier. “Exploring the calibration of cosmological probes used in gravitational-wave and multi-messenger astronomy”. PhD thesis. University of Glasgow, 2023 (cit. on p. 217).

[490] Simon C. Tait et al. “Use of Nested-Sampling Methods to Diagnose Issues in Experimental Measurements of Mechanical Disk Resonators”. *In prep.* (cit. on p. 217).

[491] Stephen P. Brooks. “Bayesian Computation: A Statistical Revolution”. In: *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 361.1813 (2003), pp. 2681–2697. ISSN: 1364503X. URL: `http://www.jstor.org/stable/3559268` (visited on 06/28/2023) (cit. on p. 218).

[492] Mark A. Beaumont and Bruce Rannala. “The Bayesian revolution in genetics”. In: *Nature Reviews Genetics* 5 (2004), pp. 251–261 (cit. on p. 218).

[493] Thomas R. Shultz. “The Bayesian revolution approaches psychological development”. In: *Developmental Science* 10.3 (2007), pp. 357–364. DOI: `https://doi.org/10.1111/j.1467-7687.2007.00588.x`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-7687.2007.00588.x` (cit. on p. 218).

[494] Ulrike Hahn. “The Bayesian boom: good thing or bad?” In: *Frontiers in Psychology* 5 (2014). ISSN: 1664-1078. DOI: `10.3389/fpsyg.2014.00765` (cit. on p. 218).

[495] T. J. Loredo. “Promise of Bayesian inference for astrophysics.” In: *Statistical Challenges in Modern Astronomy.* Jan. 1992, pp. 275–297. DOI: `10.1007/978-1-4613-9290-3_31` (cit. on p. 218).

[496] Udo von Toussaint. “Bayesian inference in physics”. In: *Rev. Mod. Phys.* 83 (3 Sept. 2011), pp. 943–999. DOI: `10.1103/RevModPhys.83.943` (cit. on p. 218).

[497] John M. Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (2021), pp. 583–589 (cit. on p. 218).

[498] OpenAI. *GPT-4 Technical Report.* 2023. arXiv: `2303.08774 [cs.CL]` (cit. on p. 218).

[499] Cynthia Rudin et al. “Interpretable machine learning: Fundamental principles and 10 grand challenges”. In: *Statistics Surveys* 16.none (2022), pp. 1–85. DOI: `10.1214/21-SS133` (cit. on p. 218).

[500] Sylvain Marsat, John G. Baker, and Tito Dal Canton. “Exploring the Bayesian parameter estimation of binary black holes with LISA”. In: *Phys. Rev. D* 103 (8 Apr. 2021), p. 083011. DOI: `10.1103/PhysRevD.103.083011` (cit. on p. 219).

[501] Riccardo Buscicchio et al. “Bayesian parameter estimation of stellar-mass black-hole binaries with LISA”. In: *Phys. Rev. D* 104.4 (2021), p. 044065. DOI: `10.1103/PhysRevD.104.044065`. arXiv: `2106.05259 [astro-ph.HE]` (cit. on p. 219).

[502] Jordan Mcginn et al. *glasflow*. Version latest. Oct. 2022. DOI: 10.5281/zenodo.7108558. (cit. on pp. 223, 247).

[503] Daniel Foreman-Mackey. "corner.py: Scatterplot matrices in Python". In: *The Journal of Open Source Software* 1.2 (June 2016), p. 24. DOI: 10.21105/joss.00024 (cit. on p. 247).

[504] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55 (cit. on p. 247).

[505] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2 (cit. on p. 247).

[506] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134 (cit. on p. 247).

[507] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a (cit. on p. 247).

[508] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2 (cit. on p. 247).

[509] Michael L. Waskom. "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021 (cit. on p. 247).

[510] Michael J. Williams. ***nessai** Experiments*. 2023. URL: https://github.com/mj-will/nessai-experiments (cit. on p. 247).

[511] Michael J. Williams. ***nessai** GWTC Analysis*. 2023. URL: https://github.com/mj-will/nessai-gwtc-analysis (cit. on p. 247).

[512] Stanislav Babak et al. "Science with the space-based interferometer LISA. V: Extreme mass-ratio inspirals". In: *Phys. Rev. D* 95.10 (2017), p. 103012. DOI: 10.1103/PhysRevD.95.103012. arXiv: 1703.09722 [gr-qc] (cit. on p. 249).

[513] G. Nelemans. "Galactic binaries as sources of gravitational waves". In: *AIP Conf. Proc.* 686.1 (2003). Ed. by J. M. Centrella, pp. 263–272. DOI: 10.1063/1.1629441. arXiv: astro-ph/0310800 (cit. on p. 249).

[514] David W. Hogg. "Distance measures in cosmology". In: (May 1999). arXiv: astro-ph/9905116 (cit. on p. 250).

[515] Antoine Klein et al. "Science with the space-based interferometer eLISA: Supermassive black hole binaries". In: *Phys. Rev. D* 93.2 (2016), p. 024003. DOI: 10.1103/PhysRevD.93.024003. arXiv: 1511.05581 [gr-qc] (cit. on p. 251).

[516] Feryal Özel and Paulo Freire. “Masses, Radii, and the Equation of State of Neutron Stars”. In: *Ann. Rev. Astron. Astrophys.* 54 (2016), pp. 401–440. DOI: 10.1146/annurev-astro-081915-023322. arXiv: 1603.02698 [astro-ph.HE] (cit. on p. 252).

[517] Pablo Villanueva-Domingo, Olga Mena, and Sergio Palomares-Ruiz. “A brief review on primordial black holes as dark matter”. In: *Front. Astron. Space Sci.* 8 (2021), p. 87. DOI: 10.3389/fspas.2021.681084. arXiv: 2103.12087 [astro-ph.CO] (cit. on p. 252).

[518] F.J. Harris. “On the use of windows for harmonic analysis with the discrete Fourier transform”. In: *Proceedings of the IEEE* 66.1 (1978), pp. 51–83. DOI: 10.1109/PROC.1978.10837 (cit. on p. 253).

Index