

**TITLE**

**SUBTITLE**

---

**Harsh Narola**

ISBN: XXX-XX-XXX-XXXX-X

DOI: <https://doi.org/12.12345/1234>

Printed by: Gildeprint – [www.gildeprint.nl](http://www.gildeprint.nl)

Copyright: © 2025 Harsh Narola



Universiteit  
Utrecht



This work originates as part of the research program of the Foundation for Fundamental Research on Matter (FOM), and falls as of April 1, 2017 under the responsibility of the Foundation for Nederlandse Wetenschappelijk Onderzoek Instituten (NWO-I), which is part of the Dutch Research Council (NWO).

**Title**

Subtitle

**Dutch title**

Dutch subtitle

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector  
magnificus, prof.dr. H.R.B.M. Kummeling, ingevolge het besluit van het college voor promoties in  
het openbaar te verdedigen op woensdag 7 februari 2024 des middags te 4.15 uur

door

**Harsh Narola**

geboren op 24 Juli 1998  
te Manvilas, India

Promotor:

Prof. dr. C.F.F. van den Broeck

Co-promotor:

Dr. Anuradha Samajdar

Assessment Committee:

Prof. dr. J.F.J. van den Brand

Dr. N.E. Chisari

Prof. dr. S. Hild

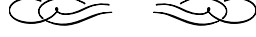
Prof. dr. R.J.M. Snellings

Prof. dr. N. Stergioulas

*Lorem ipsum dolor*

—Lorem, Ipsum.





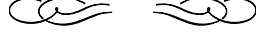
# Contents

---

<b>Introduction</b>	<b>1</b>
<b>1 Gravitational waves in general relativity: from sources to detection</b>	<b>3</b>
<b>2 Measuring the parameters of compact binaries from a gravitational wave signal</b>	<b>5</b>
2.1 Bayesian analysis . . . . .	5
2.2 Gravitational wave likelihood . . . . .	6
2.3 Sampling algorithms . . . . .	7
2.3.1 Nested sampling . . . . .	7
2.3.2 Markov chain Monte Carlo . . . . .	11
<b>3 Null stream inspired noise transient elimination</b>	<b>15</b>
<b>4 Combining deep learning with null stream inspired noise transient elimination</b>	<b>19</b>
<b>5 Measuring the source parameters of GW signals with third-generation telescope</b>	<b>21</b>
<b>6 Strong lensing to constrain modified propagation</b>	<b>23</b>
<b>Public summary</b>	<b>25</b>
<b>Openbare samenvatting</b>	<b>27</b>
<b>Summary in Gujarati</b>	<b>29</b>
<b>Curriculum vitae</b>	<b>31</b>
<b>List of acronyms</b>	<b>33</b>
<b>Acknowledgments</b>	<b>35</b>
<b>Bibliography</b>	<b>36</b>



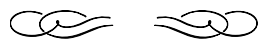




# List of Figures

2.1	An illustration of probability distribution of the prior (blue), likelihood (red), and posterior (green) of the luminosity distance $D_L$ . Measuring a source parameters amounts to transforming the prior distribution to the posterior distribution using the likelihood of the parameter at the given value. The prior on the luminosity distance is assumed to be uniform in comoving volume therefore its values increases with $D_L$ . . . . .	6
2.2	To test the Gaussian-stationary nature, we compare the distribution of whitened data (blue) with the normal distribution (red). We used 16 seconds of data segment from LIGO-Livingstone detector and the PSD is generated using 2048 seconds of data near the segment. . . . .	7
2.3	<i>Left</i> : Constant likelihood surfaces for a two dimensional parameter space $\vec{\theta} = \{\theta_1, \theta_2\}$ , illustrating the nested contours. <i>Right</i> : Evolution of the prior mass ( $X$ ) enclosed by the constant likelihood surfaces shown in the left plot. . . . .	8
2.4	<i>Bottom</i> : Progresion of a Markov chain in the parameter space of $\vec{\theta}$ . Each arrow represents an iteration, filled dots represent accepted proposal and empty dots represents rejected proposals. <i>Top</i> : Probability distribution function constructed from the accepted samples of the Markov chain from the bottom plot. . . . .	12
3.1	Glitch panel comparison . . . . .	15
3.2	Glitch overlap . . . . .	16
3.3	Mismatch . . . . .	16
3.4	Mass Distance Sky . . . . .	17
3.5	Confidence interval . . . . .	17

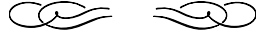




## List of Tables

---





# Introduction

---

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



# *Chapter* **1**



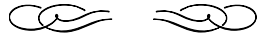
## Gravitational waves in general relativity: from sources to detection

---





# Chapter 2



---

## Measuring the parameters of compact binaries from a gravitational wave signal

---

A GW signal emitted during the merger of two compact objects can be typically modelled using 15 parameters. Eight of them are considered intrinsic parameters; two for the masses and six for three dimensional spins of the two objects. The rest of them, e.g. sky location, distance, are considered extrinsic parameters. If one of the compact object is a neutron star, an additional intrinsic parameter is added to model its tidal deformability. Measurement of these parameters is a key step which enables various physics related insights from the data, e.g. testing the prediction of the general relativity, constraining the neutron star equation of state. In this chapter, I illustrate how the parameters of merging binary black holes and/or neutron stars are measured from a GW signal.

### 2.1 Bayesian analysis

---

Measuring the parameters of a GW source amounts to treating each parameter as a random variable and estimating its probability distribution which is conditioned on the observed data. For the sake of the argument, let's say we are interested in measuring the luminosity distance ( $D_L$ ) to the source. Before the GW signal is observed, the measurement cannot be made or we can say that all values of  $D_L$  are equally likely or the *prior* probability distribution of the random variable  $M_{\text{total}}$  is uniform. After the data is collected i.e. a GW signal is observed, we can update the prior probability distribution using the data to obtain the *posterior* probability distribution. The posterior probability distribution or the *measurement* of the parameter can be thought of as the probability distribution which is conditioned on the observed data.

We formalize the problem of converting the prior probability distribution to the posterior probability distribution using the Bayes theorem. Our aim is to estimate the posterior distribution  $p(\vec{\theta}|\vec{d})$  where  $p$  denotes the probability distribution function,  $\vec{\theta}$  denotes the set of GW source parameters, and

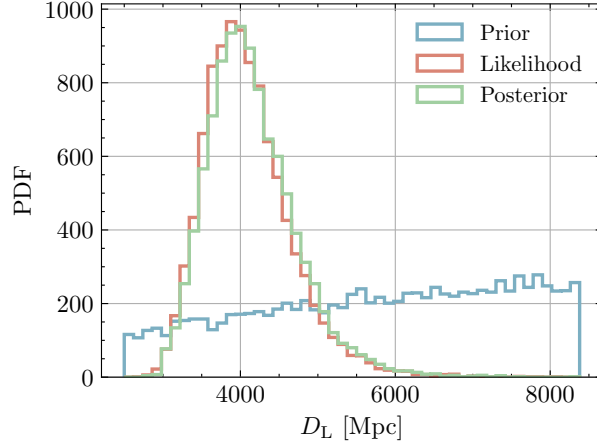


Figure 2.1: An illustration of probability distribution of the prior (blue), likelihood (red), and posterior (green) of the luminosity distance  $D_L$ . Measuring a source parameters amounts to transforming the prior distribution to the posterior distribution using the likelihood of the parameter at the given value. The prior on the luminosity distance is assumed to be uniform in comoving volume therefore its values increases with  $D_L$ .

$\vec{d}$  denotes the data. Using Bayes' theorem we can invert the conditional probability  $p(\vec{\theta}|\vec{d})$  as

$$p(\vec{\theta}|\vec{d}) = \frac{\pi(\vec{\theta}) p(\vec{d}|\vec{\theta})}{p(\vec{d})}, \quad (2.1)$$

where  $\pi(\vec{\theta})$  is the prior probability distribution of  $\vec{\theta}$  which is typically assumed to be uniform. The term  $p(\vec{d}|\vec{\theta})$  is the likelihood of observing data given the parameters  $\vec{\theta}$ . The term  $p(\vec{d})$  can be treated as the normalization term for the time being. This term is typically known as the evidence if we use Bayes theorem to test competing hypothesis. Figure 2.1 shows an illustration of the prior probability distribution (blue) and the posterior probability distribution (red) of the parameter ( $D_L$ ) where the latter is obtained by updating the prior weights by the weight of likelihood (green) for the corresponding value.

## 2.2 Gravitational wave likelihood

While the Bayesian framework described above is conceptually easy, it is a computationally challenging task to do parameter estimation due to the dimensionality of the space of  $\vec{\theta}$  (typically 15 to 17), length of the  $\vec{d}$ , and complexity of simulating of GW signal models. Before diving into how we tackle these issues, let us define a few standard quantities used during the computation, starting with the likelihood function  $p(\vec{d}|\vec{\theta})$  itself

$$\ln p(\vec{d}|\vec{\theta}) = -\frac{1}{2} \langle \vec{d} - \vec{h}(\vec{\theta}) | \vec{d} - \vec{h}(\vec{\theta}) \rangle, \quad (2.2)$$

where  $\vec{h}$  denotes the GW signal model computed as a function of the source parameters  $\vec{\theta}$ . The brackets  $\langle . | . \rangle$  denote the noise weighted inner product

$$\langle \vec{a} | \vec{b} \rangle = \frac{4}{T} \sum_f \frac{\tilde{a}(f) \tilde{b}^*(f)}{S_n(f)}, \quad (2.3)$$

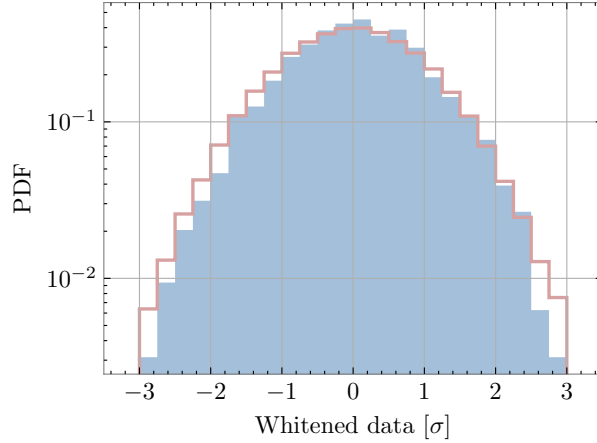


Figure 2.2: To test the Gaussian-stationary nature, we compare the distribution of whitened data (blue) with the normal distribution (red). We used 16 seconds of data segment from LIGO-Livingstone detector and the PSD is generated using 2048 seconds of data near the segment.

where  $T$  is the duration of segment  $\vec{d}$  we want to analyze,  $\tilde{b}^*$  denotes the complex conjugate of the Fourier transform of time domain vector  $\vec{b}$ , and  $f$  is the frequency. The term  $S_n(f)$  is the variance of the noise at frequency  $f$ , also known as the power spectral density (PSD).

The expression of the likelihood function is inspired from the so-called Whittle likelihood which is typically used to analyze Gaussian and stationary time series. While the data  $\vec{d}$  cannot be expected to be Gaussian and stationary at all times, for the short stretches of data that we analyze, it can be approximated with a Gaussian distribution with zero mean and known variance, making the Whittle likelihood a suitable choice. In fact, the likelihood expression in Eq. (2.2) is simply the product (or sum when operated with natural logarithm on both sides) of Gaussian distributions with zero mean and variance equal to  $S_n(f)$  for each frequency. We show a qualitative test of the Gaussian and stationary nature of the data in Figure 2.2. The distribution of the whitened data  $\frac{\tilde{d}}{\sqrt{S_n}}$  tends to follow the normal distribution. The data segment used to make this plot lies near the GW150914 signal but does not contain the signal.

## 2.3 Sampling algorithms

When performing parameter estimation, we need to explore the likelihood function on large parameter space. For example, if we choose only 10 points along each dimension of the parameter space, we need to perform about  $10^{15}$  to  $10^{17}$  likelihood evaluation where each evaluation takes  $\mathcal{O}(10)$  milliseconds. Indeed, 10 points may be too sparse to cover each dimension. Therefore, we rely on various sampling techniques to efficiently explore the likelihood function on large space. I give a brief introduction of the two sampling methods used to obtain the results of the later chapters.

### 2.3.1 Nested sampling

Nested sampling algorithm was introduced in 2004 by Skilling. The name derives from the fact that the progression of the algorithm relies on *nested* contours of likelihood (Figure 2.3 (left))

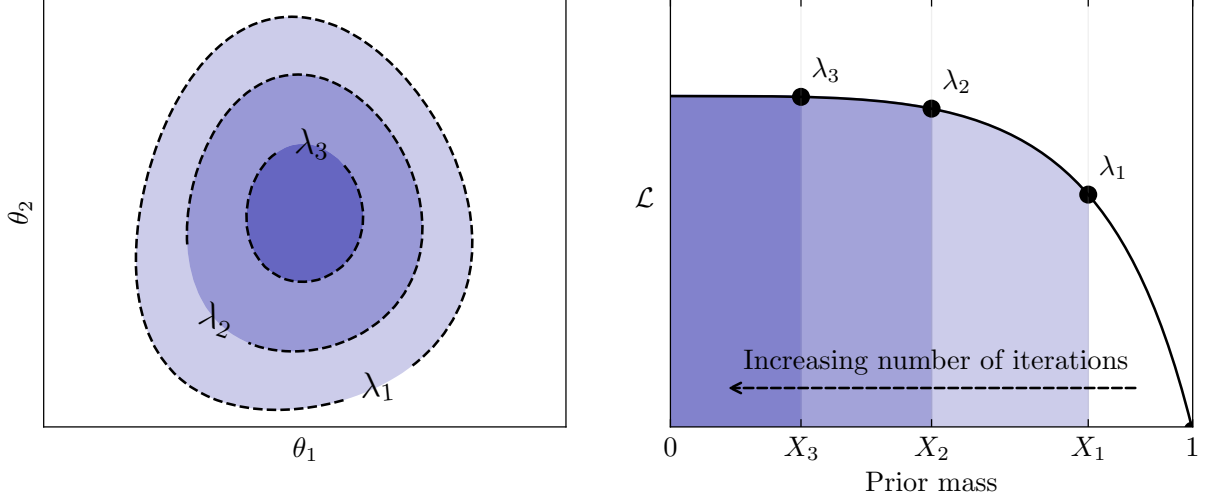


Figure 2.3: *Left*: Constant likelihood surfaces for a two dimensional parameter space  $\vec{\theta} = \{\theta_1, \theta_2\}$ , illustrating the nested contours. *Right*: Evolution of the prior mass ( $X$ ) enclosed by the constant likelihood surfaces shown in the left plot.

and not the individual likelihood values. The aim of the algorithm is to compute the evidence  $Z$ ,

$$Z = \int d\vec{\theta} \pi(\vec{\theta}) p(\vec{d}|\vec{\theta}) \quad (2.4)$$

where the integrand on the right hand side is simply the product of the likelihood and the prior defined in Eq. (2.1), i.e.,  $Z$  is the normalization constant from that equation. The posterior distribution on the model parameters,  $p(\vec{\theta}|\vec{d})$ , is obtained as a byproduct. In this sense, Nested sampling is an algorithm to perform multidimensional integral.

Typically, one can divide the parameter space of  $\Delta\vec{\theta}$  into smaller cubes such that the integrand nearly remains constant over the cube and then sum over the whole space to obtain  $Z$ . However, this approach quickly becomes intractable. Here, we instead divide the space into contours created by constant likelihood surfaces (Figure 2.3). For simplicity let us denote the likelihood term  $p(\vec{d}|\vec{\theta})$  by  $\mathcal{L}(\vec{\theta})$ . Let us define  $X(\lambda)$  as,

$$X(\lambda) = \int_{\mathcal{L}(\vec{\theta}) > \lambda} \pi(\vec{\theta}) d\vec{\theta}, \quad (2.5)$$

which is the prior mass obtained by integrating  $\pi(\vec{\theta})$  over the region of parameter space where  $\mathcal{L}(\vec{\theta}) > \lambda$ . Since the prior is normalized probability distribution, its integration over the entire space spanned by  $\vec{\theta}$  is 1, therefore maximum value of  $X$  is also 1. As  $\lambda$  increases,  $X$  tends to zero as the volume over which prior needs to be integrated grows smaller (Figure 2.3) and smaller. Using Eq. (2.5), we can convert the multidimensional integral of Eq. (2.4) into one dimensional integration,

$$Z = \int_0^1 dX \hat{\mathcal{L}}(X), \quad (2.6)$$

where  $\hat{\mathcal{L}}$  is the inverse of  $X(\lambda)$ ,

$$\hat{\mathcal{L}}(X(\lambda)) \equiv \lambda.$$

We can approximate the integral in Eq. (2.6) with the Riemann sum to obtain

$$Z \approx \sum_{k=0}^N \left( \frac{\hat{\mathcal{L}}_k + \hat{\mathcal{L}}_{k+1}}{2} \right) \Delta X_k \equiv \sum_{k=0}^N w_k. \quad (2.7)$$

Once we calculate  $Z$ , the posterior probability of point  $\vec{\theta}_k$  can be computed by

$$p(\vec{\theta}_k | \vec{d}) = \frac{\pi(\theta) \mathcal{L}(\theta)}{Z} = \frac{w_k}{\sum_{k=0}^N w_k}, \quad (2.8)$$

as mentioned near the beginning of this sub-section.

Although, we have defined prior mass  $X$ , we have so far not specified how it is calculated once the sampling starts. The quantity  $X$  is non-increasing in nature and its value ranges from 1 to 0. Treating it as a random variable, its probability distribution can be approximated with the uniform distribution,

$$X \sim U(0, 1) \quad (2.9)$$

and the corresponding cumulative distribution function is,

$$F(X) = \int_0^X dX' = X. \quad (2.10)$$

The probability that a random variable  $\chi$  is greater than the prior masses from the full set of prior masses  $\{X_i\}$  is

$$P(\chi > \{X_i\}) = \prod_{i=0}^N F(X_i = \chi) = \chi^K, \quad (2.11)$$

where we assume that the samples are independent of each other. The probability density  $p(\chi)$  is then given by

$$p(\chi) = \frac{d}{d\chi} P(\chi > \{X_i\}) = K \chi^{K-1}. \quad (2.12)$$

The above equality suggests that random variable  $\chi$  indeed follows the Beta distribution with

$$\chi \sim B(K, 1). \quad (2.13)$$

We have assumed that  $X$  and  $\chi$  range from 0 and 1. However, as the algorithm progresses, the prior mass  $X$  and, in turn,  $\chi$  shrinks. Let's say they have an upper bounded of  $X^*$ , then we can define a shrinkage ratio  $t \equiv \chi/X^*$  where

$$t \sim B(K, 1). \quad (2.14)$$

We provide the pseudocode how the nested sampling progress in Algorithm block 1. The pseudocode is generalized for  $K$  number of live points. For a typical parameter estimation run, one uses  $\mathcal{O}(10^3)$  live points.

### Terminating criterion

Since the aim of the Nested sampling algorithm is to accumulate the evidence  $Z$  in small increment of  $\Delta Z$ , it may be terminate when the increment is smaller than a user specified number  $\epsilon$ ,

$$\ln(Z_i + X_i \mathcal{L}_{\max}) - \ln(Z_i) < \epsilon, \quad (2.15)$$

where  $\mathcal{L}_{\max}$  is the maximum value of likelihood encountered during sampling.

---

**Algorithm 1:** Nested Sampling

---

**Input** : Likelihood function  $\mathcal{L}(\vec{\theta})$ , prior distribution  $\pi(\vec{\theta})$ , and number of live points  $K$

**Output:** Evidence estimate  $Z$  and posterior samples  $p(\vec{\theta}|\vec{d})$

// Initialize the algorithm

Choose  $K$  live points  $\{\vec{\theta}_1, \vec{\theta}_2, \dots, \vec{\theta}_K\}$  from the prior  $\pi(\vec{\theta})$ , compute likelihoods  $\{\mathcal{L}_k = \mathcal{L}(\vec{\theta}_k)\}$ ,  
 set initial prior volume  $X_0 = 1$ , iteration  $i = 0$ , and  $Z = 0$ ;

// Starting main loop

**while** *termination criterion not met* **do**

Identify the point  $\vec{\theta}_i$  corresponding lowest likelihood,  $\mathcal{L}_i = \min\{\mathcal{L}_k\}$ ;

**if**  $i == 0$  **then**

Draw  $X_i$  from  $\mathcal{B}(K, 1)$ ;

**else**

Draw  $t$  from  $\mathcal{B}(K, 1)$ ;

$X_i = tX_{i-1}$ ;

**end**

Discard the  $\vec{\theta}_i$  as a *dead* point, collect the tuple  $(\mathcal{L}_i, X_i)$  corresponding to  $\vec{\theta}_i$ ; Increase  $Z$   
 by adding  $\Delta Z = \frac{1}{2} (\mathcal{L}_i + \mathcal{L}_{i-1}) (X_i - X_{i-1})$ ;

Evaluate termination criterion;

**if** *criterion not met* **then**

Sample a new point from the prior such that its likelihood is higher than the point that  
 was just discarded. Add it to the set of live points so that number of live points are  
 equal to  $K$  again;

$i = i + 1$ ;

**else**

criterion met

**end**

**end**

---

### Limitations of nested sampling

I will come to this if time permits. Perhaps it is better to do a table of differences at the end of chapter.

### 2.3.2 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a subclass of Monte Carlo methods, a class of computational methods that rely on repeated random sampling to address the problem. Markov chain in particular refers to an iterative process where the probability of accepting a proposal at  $t^{\text{th}}$  iteration depends only on the point from previous iteration. If we denote the probability by  $p_t$ ,

$$p_t(\vec{\theta}') = \int d\vec{\theta}_{t-1} T(\vec{\theta}', \vec{\theta}_{t-1}) \vec{\theta}_{t-1} \quad (2.16)$$

where  $\vec{\theta}'$  is the proposed point and  $\vec{\theta}_{t-1}$  is the point from previous iteration and  $T$  is the transition probability. Standard MCMC algorithms aim to obtain the samples from posterior probability distribution which is in contrast to the Nested sampling algorithm where the aim is to compute the evidence and posterior distributions are the byproduct. Obtaining the evidence after an MCMC algorithm is terminated requires additional computations. Nevertheless, here we outline one such MCMC algorithm called the Metropolis–Hastings algorithm.

#### Metropolis-Hastings algorithm

Metropolis-Hastings algorithm makes use of a proposal density function  $Q$  to determine whether a proposal point  $\vec{\theta}'$  should be accepted. The probability that a proposal  $\vec{\theta}'$  is accepted is given by  $\alpha$

$$\alpha = \min \left( 1, \frac{p(\vec{\theta}'|\vec{d})}{p(\vec{\theta}_{t-1}|\vec{d})} \frac{Q(\vec{\theta}_{t-1}|\vec{\theta}')}{Q(\vec{\theta}'|\vec{\theta}_{t-1})} \right). \quad (2.17)$$

If the proposal is accepted, it will become  $\vec{\theta}_t$  and the step is repeated. The quantity  $Q$  can be a Gaussian distribution or any distribution from which we know how to draw samples. Figure 2.4 shows an illustration for the progression of a Markov chain (bottom) and posterior distribution constructed from the accepted samples (top).

#### Computing the evidence

The evidence when using Metropolis-Hastings algorithm is computed in post-processing. Let us introduce  $T$  which is usually known as temperature

$$p_T(\vec{\theta}|\vec{d}) \propto p(\vec{d}|\vec{\theta})^{1/T} \pi(\vec{\theta}). \quad (2.18)$$

When  $T = 1$ , the right hand side yields posterior weights and when  $T = \infty$ , it represents the prior. Samples from all chains with  $T > 1$  are eventually discarded. Writing the evidence as a function of  $\beta \equiv 1/T$  to get,

$$Z(\beta) = p_\beta(\vec{d}) = \int d\vec{\theta} p(\vec{d}|\vec{\theta})^\beta \pi(\vec{\theta}). \quad (2.19)$$

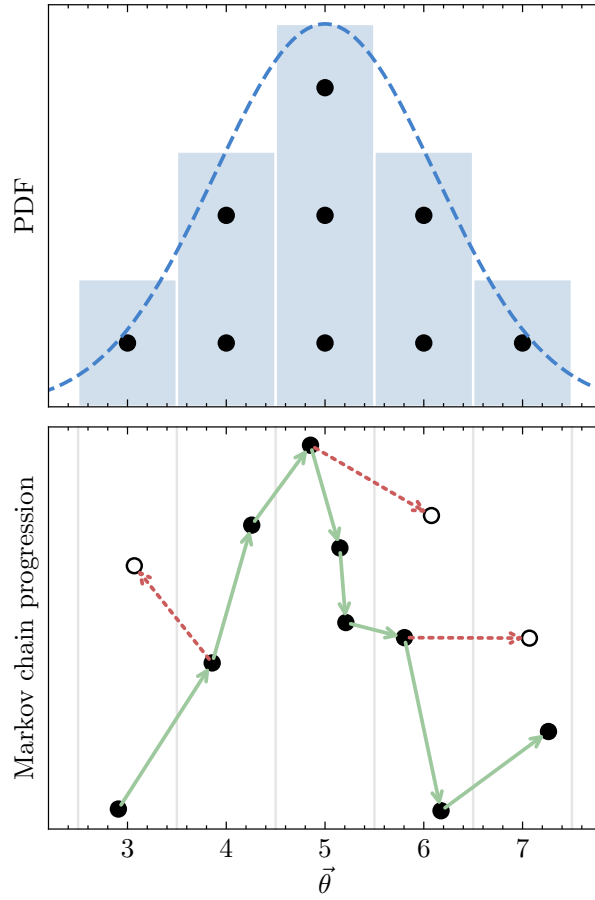


Figure 2.4: *Bottom:* Progression of a Markov chain in the parameter space of  $\vec{\theta}$ . Each arrow represents an iteration, filled dots represent accepted proposal and empty dots represents rejected proposals. *Top:* Probability distribution function constructed from the accepted samples of the Markov chain from the bottom plot.



Let us operate with log and then take derivative w.r.t.  $\beta$  on both sides,

$$\frac{d}{d\beta} \ln p_\beta(\vec{d}) = \left\langle \ln p(\vec{d}|\vec{\theta}) \right\rangle_\beta \quad (2.20)$$

where  $\langle \cdot \rangle_\beta$  stands for the expectation value with respect to the posterior distribution at temperature  $T = 1/\beta$ . The log evidence is then computed as

$$\ln p(\vec{d}) = \ln p_{\beta=1}(\vec{d}) \quad (2.21)$$

$$= \int_0^1 \frac{d}{d\beta} \ln p_\beta(\vec{d}) + \ln p_{\beta=0}(\vec{d}) \quad (2.22)$$

$$= \int_0^1 \frac{d}{d\beta} \left\langle \ln p(\vec{d}|\vec{\theta}) \right\rangle_\beta + \ln \int d\vec{\theta} \pi(\vec{\theta}) \quad (2.23)$$

$$= \int_0^1 \frac{d}{d\beta} \left\langle \ln p(\vec{d}|\vec{\theta}) \right\rangle_\beta \quad (2.24)$$

$$\approx \sum_{\beta=1/T_{\max}}^1 \Delta\beta \left\langle \ln p(\vec{d}|\vec{\theta}) \right\rangle_\beta \quad (2.25)$$

Hence, the expectation value of the log-likelihood with respect to the posterior distribution integrated over the temperature range leads to the evidence.

### Limitations of Metropolis-Hastings algorithm

Indeed, the efficiency of this algorithm depends on the choice of  $Q$ . In addition, each chain is started at random point and we need to discard certain number of iterations at the beginning so that the dependence on the starting point is lost. This period is called the *burn-in* period which again depends on the choice of  $Q$ . Moreover, adjacent samples for a given chain maybe correlated which is undesirable. We can choose to store the samples every  $n^{\text{th}}$  iterations to mitigate this issue, where  $n$  is greater then or equal to the autocorrelation time of the chain. This process is often called thinning.

### Reversible Jump Markov chain Monte Carlo



# Chapter 3



## Null stream inspired noise transient elimination

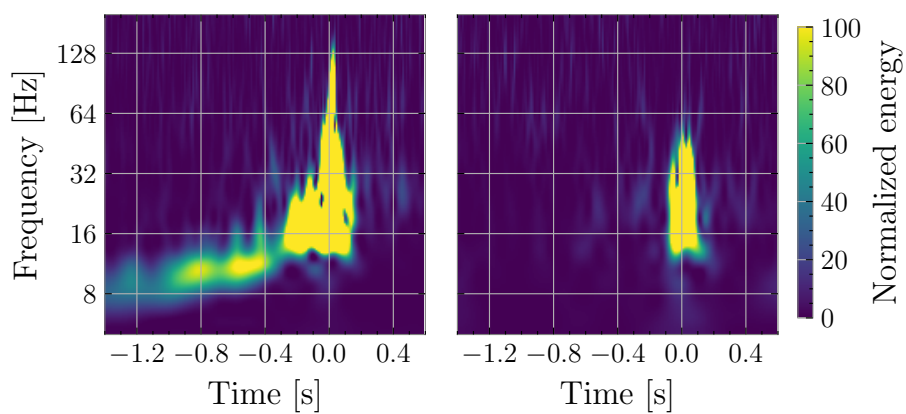


Figure 3.1: Glitch panel comparison

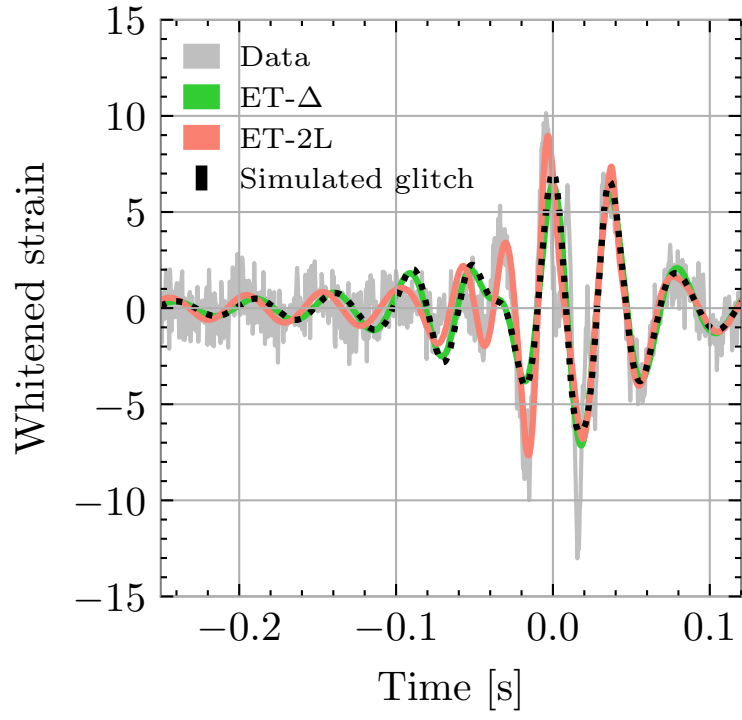


Figure 3.2: Glitch overlap

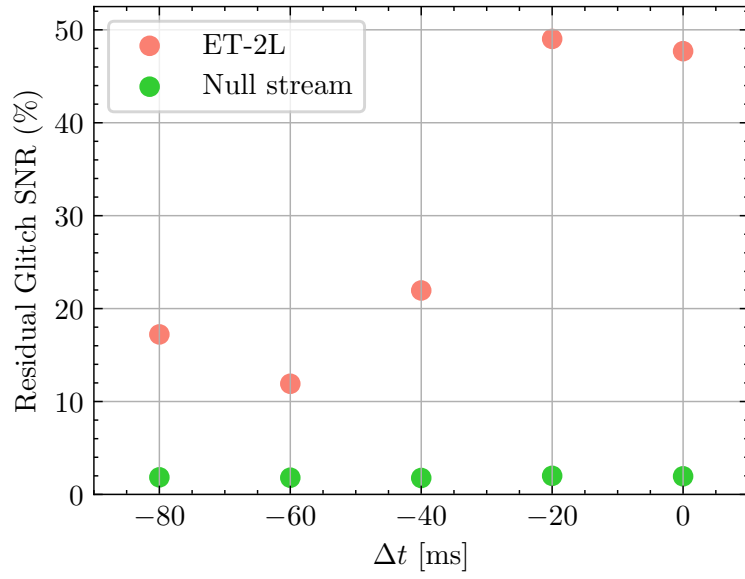
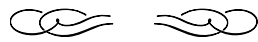


Figure 3.3: Mismatch





# Chapter 4



## Combining deep learning with null stream inspired noise transient elimination

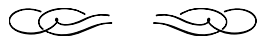
---

FIXME: What will this chapter contain? Combining null stream with deep extractor





# *Chapter* 5



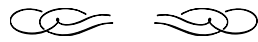
## Measuring the source parameters of GW signals with third-generation telescope

---

Relative binning with higher orders modes and precession FIXME: What will this chapter contain?



# *Chapter* **6**

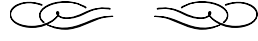


## **Strong lensing to constrain modified propagation**

---

Strong lensing to constrain modified propagation FIXME: What will this chapter contain?

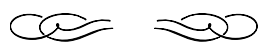




## Public summary

---



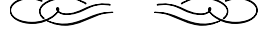


## Openbare samenvatting

---



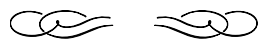




## Summary in Gujarati

---

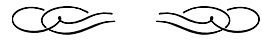




## Curriculum vitae

---

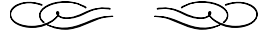




## Acronyms

---



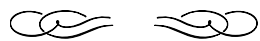


## Acknowledgments

---







## Bibliography

---

