Testing a new emulator for binary black hole mergers

by
Jake Jackson

The University of Padova
Department of Physics
October 2023

Submitted to the Department of Physics in partial fulfillment of the requirements the Master's degree in Physics of Data

Supervisors: Dr Giuliano Iorio & Prof Michela Mapelli

Student Number: 2046805

Abstract

Sampling the relevant parameter space for binary black hole mergers is a conundrum in current gravitational-wave astronomy. Hence, one of the most wanted tools in the field of gravitational-wave astronomy is a simulation emulator, i.e. a tool that allows us to up-sample the simulations done with a population-synthesis code but with much larger statistics and lower computational costs. The aim of this thesis project is to test the code presented by Cheung et al.[1] (2022, https://ui.adsabs.harvard.edu/abs/2022PhRvD.106h3014C/abstract) onto a new set of population-synthesis simulations, performed with our code SEVN (Iorio et al. 2023, https://ui.adsabs.harvard.edu/abs/2023MNRAS.tmp.1606I/abstract). This algorithm is based on normalizing flows and will be used to up-sample the simulations with SEVN. We will use the new tool to perform a study of the parameter space (common-envelope ejection efficiency, natal kicks, metallicity). Our goal is to use this data set to obtain an insight into the formation channels of binary black holes observed by LIGO and Virgo during the Third Gravitational-Wave Transient Catalog (GWTC3).

Acknowledgments

Personal Acknowledgments

I would like to express my sincere gratitude to Dr Giuliano Iorio and Prof Michela Mapelli for their guidance, support, and encouragement throughout my research.

I am also very grateful to,

- Kaze W. K. Wong: Center for Computational Astrophysics, Flatiron Institute
- Damon H. T. Cheung: Department of Physics, The Chinese University of Hong Kong
- Otto A. Hannuksela: Institute for Theoretical Physics, KU Leuven

for kindly sharing their normalized flows simulation emulator implementation.

Organizations

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1916 Albert Einstein predicted the existence of gravitational waves [5]. This was only one year after the final form of the general relativity field equations. Approximately 100 years later, in 2016 the LIGO collaboration announced the first direct observation of gravitational waves with a significance greater than 5.1 σ[6]. The signal was from a double black hole binary (BHBs) system with masses of 29 and 36 $M_\odot$ merging $\sim$ 1.3 billion light-years away. Many subsequent detections have followed including the first binary neutron star (BNS) merger in 2017 [7]. The latest periodically released catalogue GWTC-3 lists 35 probable events including three possible neutron star-black hole coalescences [8].

This has opened up a new era of astronomy and allows us to observe some of the most exciting and extreme events in the universe. The implications of these observations give insights into previously unknown astrophysics. A new detection method not reliant on light from the system offers a unique observational window, undisturbed by the traditional problems of delay and scattering from dust, gas, or other obstacles.

Given these insights we have an exciting space to explore between new gravitational wave events and our understanding of stellar evolution. Here we use the the population synthesis code SEVN, to evolve a group of stars and we analyse the resulting merged binaries. The preliminary focus of this project is to utilize the Normalizing Flows (NF) machine learning technique to develop a tool to aid this research.

Previous results from Cheung et al (2022) suggest that NF has potential to be able to learn the complex distributions and produce accurate population synthesis samples in a significantly faster time. This project is focused utilize NF for SEVN and in depth exploration of the parameter space. This is particularly exciting as it offers a clear path way to implement the power of machine learning in astrophysics in a way that is additive to the scientific process.

This chapter aims to briefly outline the scientific landscape, and explain the key parameters explored in this investigation. As well as capture the scientific excitement for gravitational waves a binary merger events.

1.1 Binary systems

A binary system is a system of two astronomical bodies that are close enough that the gravitational force causes them to orbit each other around their common centre of mass in a setup similar to that of Figure 1-1. Binary stars are usually created as a bound system during star formation . Whilst it's possible a star may capture another, in the presence of a third body, into a bound state, this is highly unlikely even in a densely populated region [9].

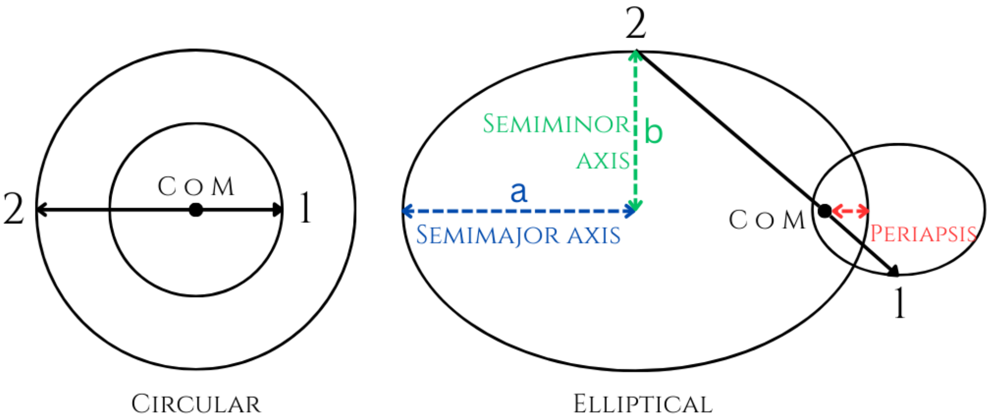


Figure 1-1: Illustration of the orbit of two masses around their centre of mass (CoM), the elliptical case is with an eccentricity ~ 0.8. Where mass M_1 is the largest mass.

Here we can see two arrangements one in a circular orbit and one in an elliptical configuration. A system like either of the examples in Figure 1-1 is a snapshot in time. The stars will continue to evolve and as they lose or gain mass they will change radius and perhaps undergo a merger event. For the context of gravitational waves, it's important to remember that the force attracting them is due to the warping of the fabric of spacetime as described by general relativity.

1.2 Gravitational waves

Gravitational waves are a consequence of General Relativity. In simple terms, gravitational waves are ripples in the curvature of spacetime that propagate as waves . They permeate outward from a given source such as a compact binary merger at the speed of light. They are derived from of the famous field equations of General Relativity, such as 1.1 below.

$$R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R = \frac{8\pi G}{c^4}T_{\alpha\beta} \tag{1.1}$$

Where $R_{\alpha\beta}$, $g_{\alpha\beta}$ and $T_{\alpha\beta}$ are all 4 x 4 tensors. $R_{\alpha\beta}$ is the Ricci tensor obtained by contracting the indices of the Riemann curvature tensor. In addition, $g_{\alpha\beta}$ is the metric tensor and $T_{\alpha\beta}$ is the stress energy tensor.

The specific steps to derive the formal solution depend on the situation we are trying to model strong curvature requires a numerical solution of fully nonlinear Einstein equations. In a weak field (almost flat curvature) with ”slow ”sources the analytic solution of the linearized wave equations is used [10]. In this case the quadratic terms in $g_{\alpha\beta}$ don’t significantly contribute, this leads to the wave equation below,

$$\Box \bar{h}_{\alpha\beta} = \frac{16\pi G}{c^4} T_{\alpha\beta} \tag{1.2}$$

Where $\Box = \eta^{\alpha\beta}\partial_\alpha\partial_\beta$ is the d’Alembert operator. The gravitational wave solution must be a solution of the wave equation 1.2. For a binary system, the emission of gravitational results in a loss of orbital energy therefore the binary shrinks as it emits until it merges. The binaries in circular orbits the radius between the masses is given by,

$$r(t) = r_0 \left(1 - \frac{t}{t_{\text{coalesce}}}\right)^{1/4}. \tag{1.3}$$

Where r_0 is the initial radius and t_{coalesce} is the total time to coalesce [11]. Furthermore, the rate of orbital decay can be estimated with the equation,

$$\frac{\mathrm{d}r}{\mathrm{d}t} = -\frac{64}{5}\frac{G^3}{c^5}\frac{(M_1 M_2)(M_1 + M_2)}{r^3}, \tag{1.4}$$

.

We can also determine the expected merger time as follows,

$$t = \frac{5}{256}\frac{c^5}{G^3}\frac{r^4}{(M_1 M_2)(M_1 + M_2)} \tag{1.5}$$

1.3 Mass transfer

1.3.1 Wind mass transfer

All main sequence stars produce stellar wind. This stellar wind is comprised of particles, including protons, electrons and atoms. This wind is illustrated by the solar winds photograph in Figure 1-2. Various processes give rise to this wind such as internal fusion driven outward pressure, and the star’s magnetic field. For low to intermediate mass stars like the Sun, this tends to be driven by stellar magnetic fields. Only a small mass fraction of the star is lost prior to becoming red giants. Massive stars with $M_{ZAMS} \gtrsim 30\odot$ and $T > 10^4 K$ can however lose up to half of their mass through stellar winds. They have strong winds driven by the outward pressure of their own radiation.

Stellar winds remove mass and angular momentum from one or both stars in a binary system. This mass loss can affect the evolution of the stars and their future mass transfer dynamics. Furthermore, SEVN actually takes into account the possibility that some of the wind mass and momentum can be accreted by its stellar companion [12].

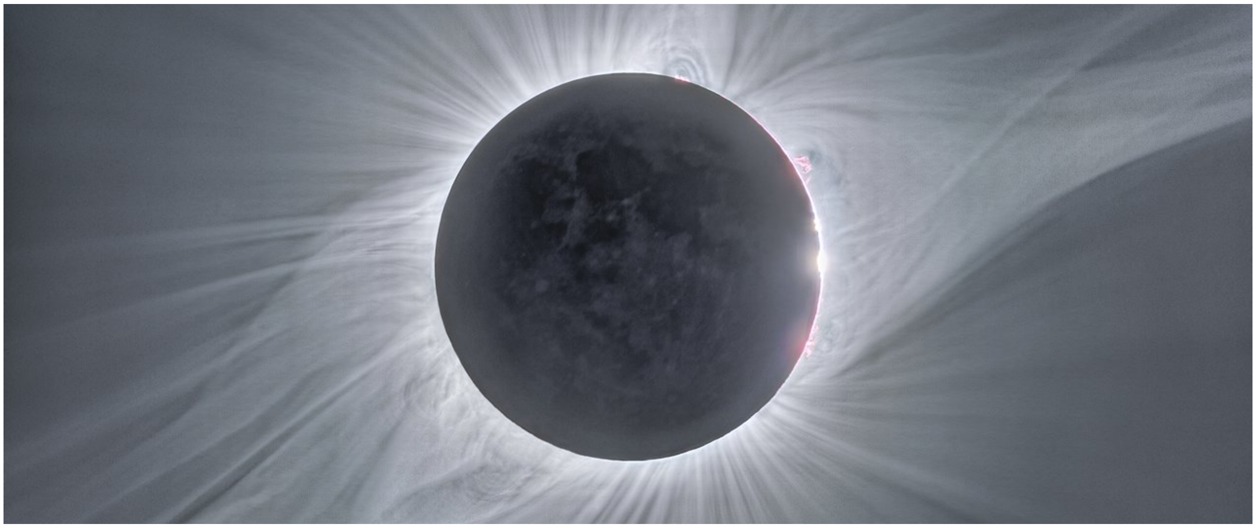


Figure 1-2: Image of solar wind revealed during eclipse. Credit:ESO/P. Horálek/Solar Wind Sherpas project [2]

A very small proportion of the stellar wind undergoes mass transfer from donor to accretor. Interestingly the ratio between the rate of mass accretion and wind mass loss by the donor is as follows.

$$\left|\frac{\dot{M}_{\mathrm{a}}}{\dot{M}_{\mathrm{d}}}\right| \propto \left(\frac{v_{\mathrm{orb}}}{v_{\mathrm{wind}}}\right)^4 \tag{1.6}$$

Where v_{orb} and v_{wind} are the orbit and wind velocities. In the fast wind approximation the $v_{\mathrm{wind}} >> v_{\mathrm{orb}}$. This leads to a small ratio in equation 1.6. The winds from the star are ejected spherically not just towards the secondary, in terms of mass transfer this is a very inefficient process that does not conserve mass in the binary.

SEVN implements the mass accretion following Bond and Hoyle (1944) [13] and assuming fast wind. The rate of mass accretion is given by,

$$\dot{M}_{\mathrm{a}} = -\frac{1}{\sqrt{1-e^2}}\left(\frac{GM_{\mathrm{a}}}{v_{\mathrm{wind}}^2}\right)^2 \frac{\alpha_{\mathrm{wind}}}{2a^2}\frac{1}{\left(1+v_f^2\right)^{3/2}}\dot{M}_{\mathrm{d}}, \tag{1.7}$$

where $\dot{M}_{\mathrm{d}}$ is the rate of wind mass loss by the donor, a is the binaries semimajor axis. In addition,

$$v_{\mathrm{orb}}^2 = \frac{G\left(M_{\mathrm{d}}+M_{\mathrm{a}}\right)}{a}, \qquad v_{\mathrm{wind}}^2 = 2\beta_{\mathrm{wind}}\frac{GM_{\mathrm{d}}}{R_{eff}}, \qquad v_f^2 = \frac{v_{\mathrm{orb}}^2}{v_{\mathrm{wind}}^2} = \frac{G\left(M_{\mathrm{d}}+M_{\mathrm{a}}\right)}{av_{\mathrm{wind}}^2} \tag{1.8}$$

R_{eff} is the minimum difference between the radius of the star and its Roche lobe radius known as the effective stellar radius. The dimensionless. wind parameters in SEVN are set to $\alpha_{\mathrm{wind}} = 1.5$ and $\beta_{\mathrm{wind}} = 0.125$ [12].

1.3.2 Roche lobes

The term ”Roche-lobe” refers to the distinct ”tear shaped” bound by the in binary systems. The lines in Figure 1-3 represent gravitational equipotentials. The surfaces surrounding each mass touch at the inner Lagrangian point L1 resulting in a dumb-bell shaped configuration. Equipotentials, having constant potential allow a given test particle to move freely on the potential without acquiring or loosing additional energy. The Roche lobes are the volumes enclosed by the limit equipotential surface i.e the ones that pass through L1. Physically they represent the volume in which material is gravitationally bound to only one of the stars [14]. The gravitational potential below is given the assumption that point masses and the binary is circular. To simplify the problem a rotating frame of reference is set with rotation period set to match the orbital period.

$$\Phi = -\frac{GM_1}{\left(x^2+y^2+z^2\right)^{1/2}} - \frac{GM_2}{\left[(x-a)^2+y^2+z^2\right]^{1/2}} - \frac{1}{2}\omega^2\left[\left(x-\frac{M_2 a}{M_1+M_2}\right)^2+y^2\right] \tag{1.9}$$

The first two contributions to equation 1.9 correspond to the gravitational potentials from M_1 and M_2 respectively, whereas the final term is from the fictitious centrifugal force in the rotating system.

This can be further simplified by introducing the mass ratio $q = \frac{M_2}{M_1}$ Given that the strength of the gravitational force is proportional to their masses, so is the Roche lobes form. In Figure 1-4 it can be seen that when masses are similar sizes the Roche lobe is more symmetrical, conversely a low q results in a less symmetric Roche lobe as seen in 1-3a. Factoring q into equation 1.9 gives,

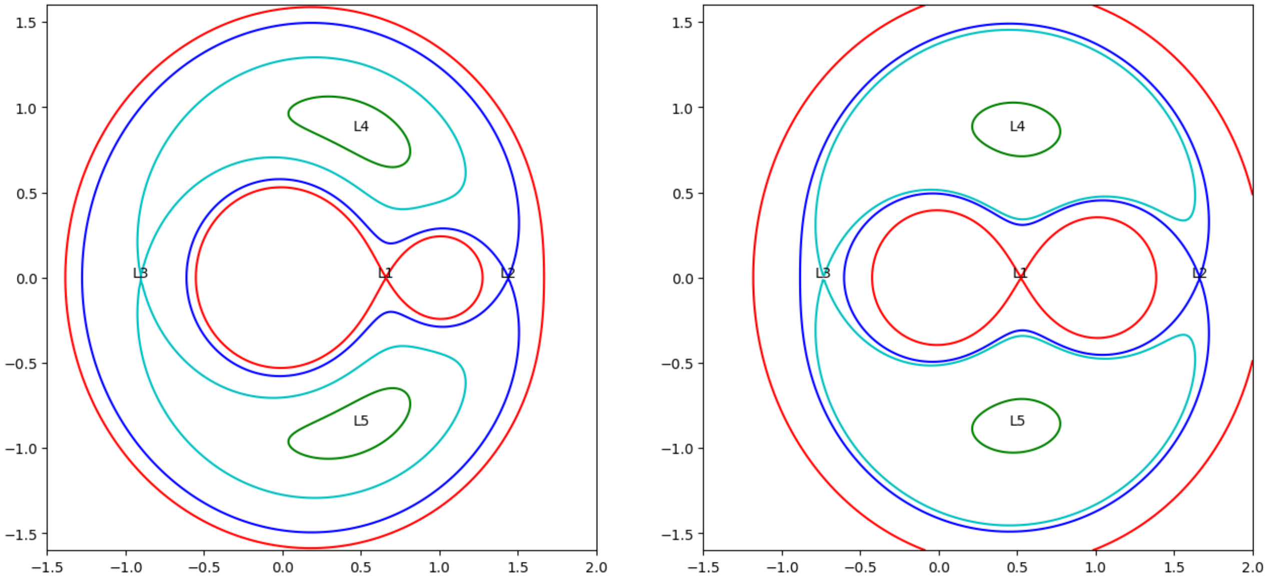
$$\Phi' = \frac{2}{(1+q)\left(x^2+y^2+z^2\right)^{1/2}} + \frac{2q}{(1+q)\left[(x-1)^2+y^2+z^2\right]^{1/2}} + \left[\left(x-\frac{q}{1+q}^{2}\right)+y^2\right] \tag{1.10}$$

$$\Phi' = \left[-\frac{G\left(M_1+M_2\right)}{2a}\right]^{-1}\Phi \tag{1.11}$$

For a more realistic (non point like) star there may be a case in which the stars physical volume extends beyond the Roche lobe. The stars material with a radius greater than the Roche lobe this material can escape from the object and pass to the accretor through L1, this event is known as a Roche lobe overflow (RLO). During the RLO, the transferring material gains angular momentum due to the Coriolis force, this material can form an accretion disk around the star gaining mass.

These Roche lobe considerations outlined above result in three distinct binary outcomes.

1. Detached binaries: neither star fill its Roche lobe
2. Semidetached binaries: one of the pair fills its Roche lobe and can pass its mass to the accompanying object.
3. Contact binaries: both stars exceed the Roche lobe and form a common envelope. This makes both



(a) q = 0.2

(b) q = 0.8

Figure 1-3: Roche lobes and the normalized gravitational potentials on the orbital plane $z = 0$ (created using PyAstronomy [3])

stars difficult to distinguish from each other.

1.3.3 Mass transfer stability

Mass transfer will change both the Roche lobes and the radii of the stars in the binary system. If the radius of a star shrinks faster than its Roche lobe the mass transfer event will gradually stop and the mass transfer is considered to be unstable [15]. Conversely if the donor stars radius shrinks faster or doesn't not expand as fast as the Roche lobe, mass transfer will be stable and should last a long time.

The stars with more radiative envelopes usually shrink in response to mass loss, conversely those with deep convective envelopes tends to maintain their radius or even slightly expand [16].

The donor's mass loss response ζ when compared to the Roche lobe variation to ζ_{L} can be used in order to evaluate the mass transfer stability. They are given by,

$$\zeta = \frac{d \log R}{d \log M} \qquad \zeta_{\mathrm{L}} = \frac{d \log R_{\mathrm{L}}}{d \log M} \tag{1.12}$$

This is difficult to evaluate and population synthesis codes simplify the problem by comparing the mass ratio to a mass ratio value critical value,

$$q_{\mathrm{c}} = 0.362 + \frac{1}{3\left(1 - \frac{M_{\mathrm{He,d}}}{M_{\mathrm{d}}}\right)} \tag{1.13}$$

where $M_{He,d}$ is the donor helium mass. The the SEVN fiducial was used to generate the training data for this project. This option is the same as the Hurley et al(2002) implementation with the exception the the donors mass transfer is always stable if it has a radiative envelope [12].

1.3.4 Common envelope efficiency

The case for the contact binaries is very interesting. The common envelope contains the dense cores of the stars. When inside the common envelope drag forces cause them to spiral inward. The orbital energy of the binary plays a crucial role in the outcome. If the orbital energy is efficiently transferred to the envelope and used to expel it, the binary may survive in an albeit tighter orbit. If this transfer is inefficient the stars spiral closer leading to a merger or more compact system.

In order to parameterise this energy removal efficiency a dimensionless parameter known as common envelope efficiency α is used. Typically in the range $0 < \alpha < 1$, with the higher values representing more efficient energy transfer. Values of $\alpha > 1$ are used in SEVN to account for additional sources of energy. An example of this is the envelope recombination energy or the uncertainty from using this relatively simple formalism.

The α formalism first introduced in 1984 by Webbink [17] it's a useful analytic approach however it does not capture underlying physics. In SEVN the Hurley+ standard energy prescription is used. Here we give a basic overview of the regime. In this approach, alpha is introduced as the free parameter for energy removal efficiency. Usually $0 < \alpha < 1$

$$E_{bind} = -\frac{G}{\lambda}\left(\frac{M_1 M_{env,1}}{r_1} + \frac{M_2 M_{env,2}}{r_2}\right) \tag{1.14}$$

Where the subscripts i f, 1 and 2 denote the initial state, final state, and two interacting objects respectively. E is energy, M is mass, r is radius, λ is a factor that allows us to incorporate geometry. The shorthand's *env* is the envelope *bind* is binding. The orbital binding energy is given by

$$E_{orb} = \frac{1}{2}\frac{GM_{c,1}M_{c,2}}{a} \tag{1.15}$$

Where c subscript denotes the cores and a is the semimajor axis. Finally, alpha is introduced in the formula for the change in orbital energy needed to unbind the envelope.

$$E_{bind,i} = \Delta E_{orb} = \alpha(E_{orb,f} - E_{orb,i}) \tag{1.16}$$

It is also important to note that a merger occurs in the event that the final semimajor axis is less than the sum of the core radii.

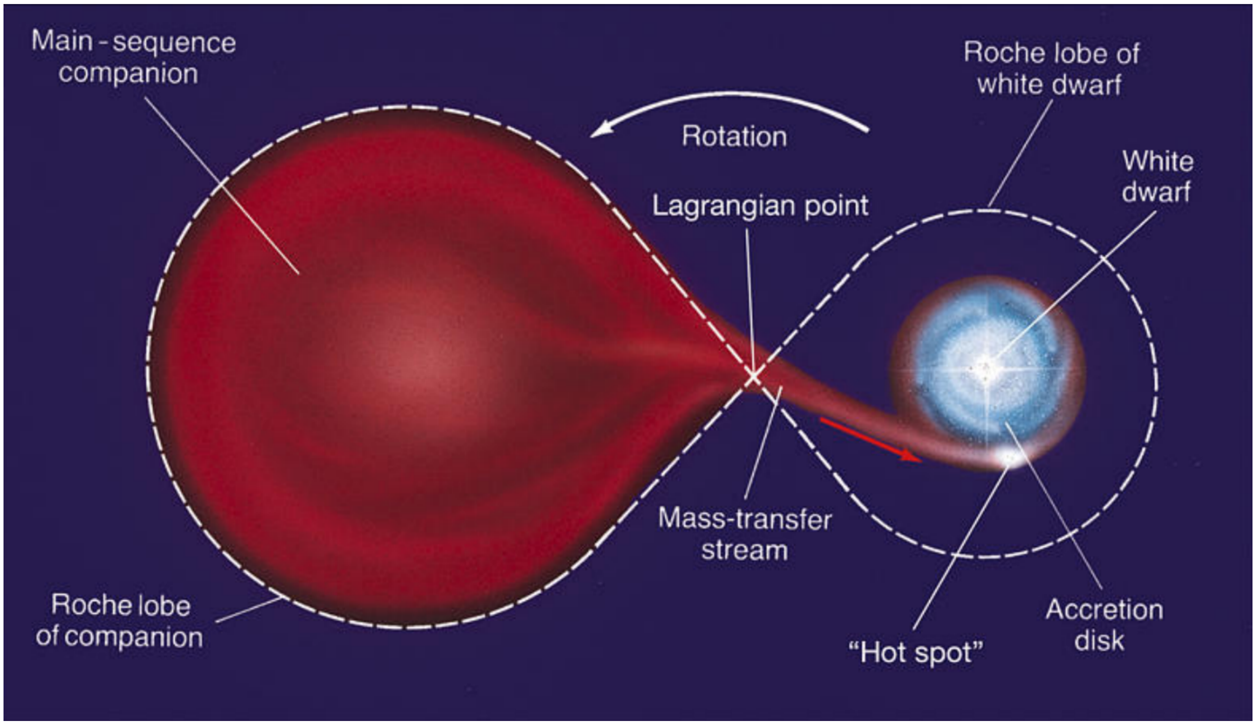


Figure 1-4: Roche Lobe overflow illustration by Person Prentice Hall, Inc ©2005

1.4 Metallicity

The Astrophysical definition of a metal varies somewhat from the chemical definition, every element heavier than Helium (He) is considered a metal. Usually, this concept is parameterized by Z which is the mass fraction of elements heavier than He. The metallicity of a star is an important characteristic because it can provide insights into the star's history, age, and formation. When the Universe was first created in the Big Bang, it consisted primarily of hydrogen and helium, with trace amounts of other elements. Stars are formed from these primordial materials. The first generation of stars had very low metallicity because they were essentially composed of hydrogen and helium. As stars go through their life cycles, they undergo nuclear fusion in their cores, converting hydrogen into helium and eventually into heavier elements through a process called nucleosynthesis. This enriches the star's composition with heavier elements. Stars are often classified into three broad metallicity categories [18]:

- Population I: These are stars with high metallicity, meaning they have a relatively high abundance of heavy elements[19]. Population I stars, like our Sun, are typically found in the spiral arms of galaxies and are associated with younger stellar populations .

- Population II: These stars have lower metallicity compared to Population I stars[19]. They are generally found in the galactic halo and the central bulge of galaxies. Population II stars are older and formed when the Universe had fewer heavy elements.

- Population III: These are hypothetical, extremely old and metal-poor stars that are believed to have formed in the very early Universe. They have virtually no heavy elements and have not been observed directly. The existence of Population III stars is inferred from theoretical models [20].

High-metallicity stars are thought to have stronger solar winds than that of low-metallicity stars, driven by there own radiation pressure [21]. This traditionally leads to the expectation that high metallicity progenitors will release too much mass via wind to be able to form black holes. This has an exception in the case of stars with particularly strong magnetic fields.

Chapter 2

Methodology

2.1 SEVN : Stellar EVolution for N-body

The acquisition of direct experimental evidence in Astrophysics is deeply observational, these observations capture a small moment in time of objects with vast evolutionary timescales. In order to explore a full evolutionary pathway its essential to use model these systems. Population synthesis codes are a powerful tool that allows us to apply our current analytical understanding and predict the stellar evolution. These models are invaluable for understanding the collective behavior of stars in various astronomical contexts and are particularly useful for studying galaxies, star clusters, and the overall evolution of stellar populations.

This project utilizes the population synthesis code SEVN. The SEVN code allows users to inputs the stars and binaries with initial conditions such (masses, spin, semi-major axis, eccentricity) and then evolves them through time. SEVN uses interpolation of pre-calculated stellar tracks in order to compute stellar evolution. For binary evolution, it relies on analytic and semi-analytic formulas. The primary benefit of this approach lies in its capacity to enhance generality and adaptability: the stellar evolution models used in SEVN can be readily modified or upgraded by loading new sets of reference tables. SEVN permits the selection of stellar tables without the need for alterations to the code's internal structure or recompilation. The most use simple case is for a single stellar track like that in Figure 2-1.

SEVN is very advanced in terms of binary evolution processes. This makes it an extremely exciting population synthesis code for research into the sources of gravitational waves. SEVN incorporates a wide array if mass transfer regimes such as common envelope, Roche-lobe overflow, wind mass transfer, stellar tides, RLO onset circularisation and collision events at periastron.

2.1.1 Input data: Fiducial 5 million

The input data for this project was the fiducial model for 5 million binaries this data was first produced for the paper ”Compact object mergers: exploring uncertainties from stellar and binary evolution with SEVN”

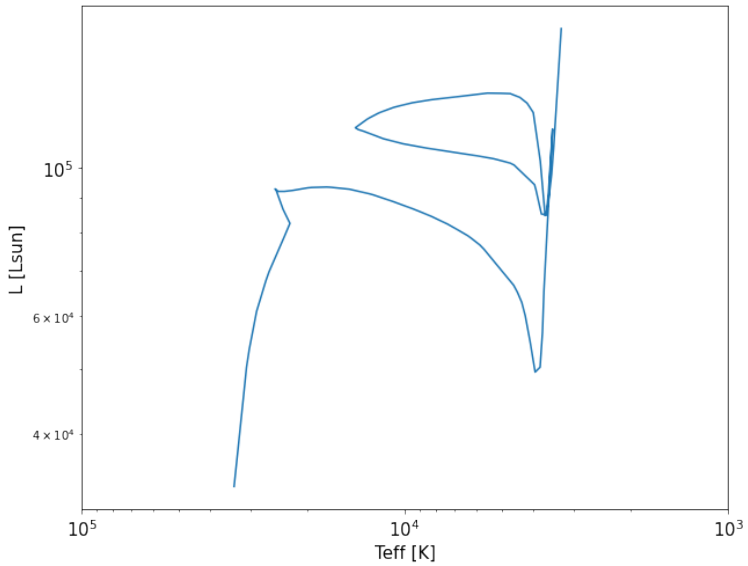


Figure 2-1: SEVN single stellar evolution (SSE) example from the SEVN Python Wrapper adaptation of the SSE tracks tutorial.

by Iorio, G. et al 2022 [12]. The run contains a vast collection of data $\sim$ 1.8 TB. For the purposes of this project we are interested in the 801038 BHBH mergers. This fortunately saves a lot of computational costs.

The normalized flows simulation emulator is trained on the following parameters i.e. masses M_0 and M_1, metallicity Z and common envelope (CE) efficiency α. M_0 is the mass of the black hole that was initially the most massive star in the binary. Conversely M_1 is the mass of the black hole that was initially the least massive star of the system.

In terms of training a machine learning technique like that of normalized flows it is important to get an idea of the balance of the data. Figures 2-2 and 2-3 give us an idea of this distribution. There is a significant data spike in the mass distributions as shown in Figure 2-3 in fact $\sim 50\%$ of all the BHBH mergers are predicted to be in the range 5.5 to 9.5 $M_\odot$. A data imbalance makes learning the distribution through machine learning challenging as the model can learn that it can achieve high accuracy by consistently predicting the region with the majority even if recognizing the minority regions are important when applying the model to the full range of possible physics. Given the purpose this emulator is to generate samples distribution of the data we cannot adjust the balance of the data as is customary in other machine learning applications.

2.2 Normalizing flows

The name of the Normalizing Flows (NF) generative model is comprised of two components. Normalizing as when transforming the distribution it needs to be normalized by a change of variables. Flows refer to the series of invertible transformations that make up the more complex transformations. In simple terms, a basic probability density is transformed into a complex target probability density. For instance, a simple

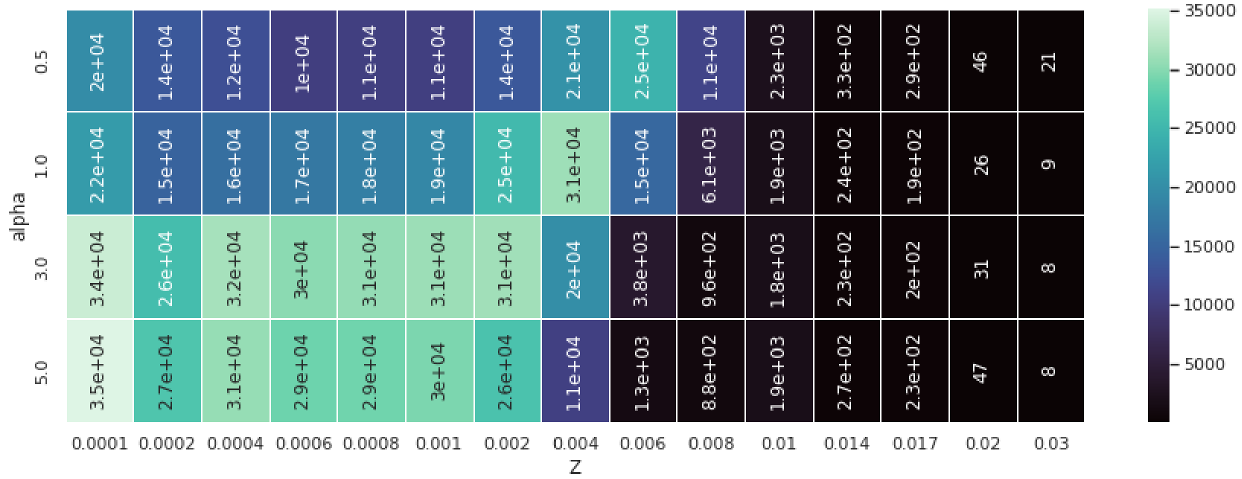


Figure 2-2: Frequency of merged BHBH binaries from SEVN data grouped by each combination of α and Z.

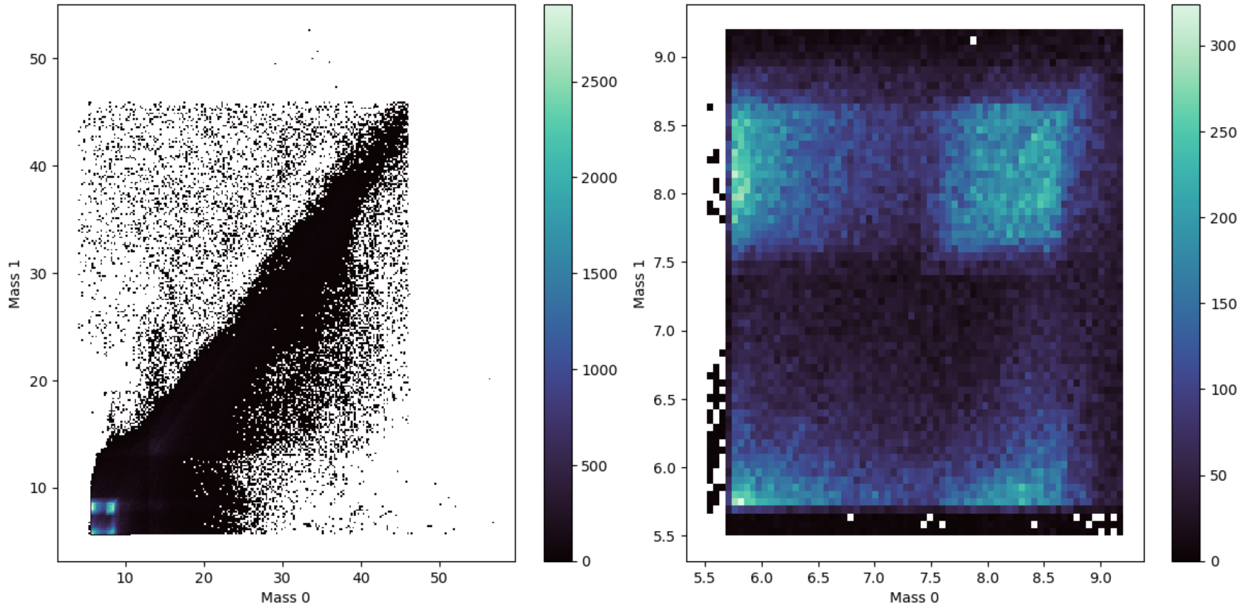


Figure 2-3: 2D histogram of the SEVN data showing the data spike $\sim 50\%$ of all the BHBH mergers are predicted to be in the range mass range 5.5 to 9.5 $M_\odot$

.

Gaussian $z \sim p_z$ to complex target probability density function $x \sim p_x$. This is done through the use of an invertible transformation easy to compute (i.e. tractable) Jacobian [22].

This is achieved through the use of transformations g such that $g : \mathbb{R}^d \to \mathbb{R}^d$ for $x, z \in \mathbb{R}^d$. The key idea of normalizing flows is simple, probabilities must be normalized to allow us to get the change of variables formula below,

$$\int p_x(x)dx = \int p_z(z)dz = 1, \qquad \therefore \qquad p_x(x) = p_z(z)\left|\det\frac{\partial z}{\partial x}\right| \tag{2.1}$$

where the Jacobian, $J = \left|\det\frac{\partial z}{\partial x}\right|$ scales the distribution in order to ensure the enclosed area is always

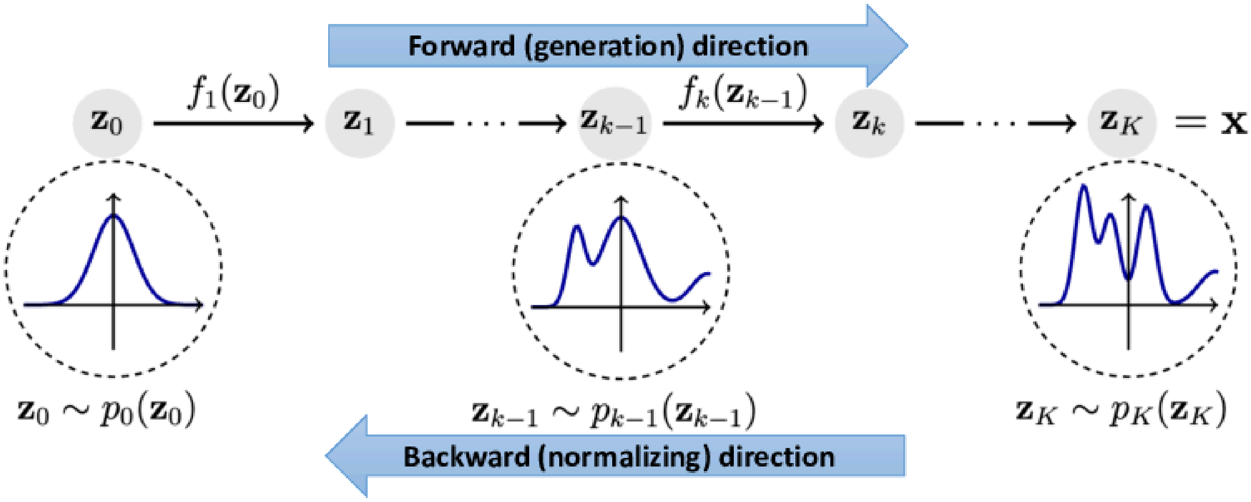

Forward (generation) direction
$\mathbf{z}_0$
$f_1(\mathbf{z}_0)$
$\mathbf{z}_1$
$\mathbf{z}_{k-1}$
$f_k(\mathbf{z}_{k-1})$
$\mathbf{z}_k$
$\mathbf{z}_K = \mathbf{x}$
$\mathbf{z}_0 \sim p_0(\mathbf{z}_0)$
$\mathbf{z}_{k-1} \sim p_{k-1}(\mathbf{z}_{k-1})$
$\mathbf{z}_K \sim p_K(\mathbf{z}_K)$
Backward (normalizing) direction

Figure 2-4: Normalizing flows diagram showing the forwards and backwards directions (Source [4])

equal to one. In the normalizing flows technique it is important that these transformations are invertible in order to allow both the forwards (from simple to complex distribution) and backwards directions. Explicitly this property can be wrote like $z = g^{-1}(x)$ and substituting into 2.1 gives the equation below.

$$p_x(x) = p_z(g^{-1}(x)) \left| \det \frac{\partial g^{-1}(x)}{\partial x} \right| \tag{2.2}$$

In order to estimate complex distributions many of these transformations can be chained together into a ”flow”. Stacking together multiple invertible functions after each other, as all together, they still represent a single, invertible function. This chain of transformations may look something like $z_K = g_K \cdot \cdots \cdot g_2 \cdot g_1(z_0)$ where K represents the number of blocks. In order to train a normalized flows model the loss is calculated, this is done using the log likelihood as follows,

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log(p_x(x)) = -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log \left(p_z(z) \left| \det \frac{\partial z}{\partial x} \right| \right) = -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log(p_z(z)) + \log \left| \det \frac{\partial z}{\partial x} \right| \tag{2.3}$$

Expanding the formula out as in the right hand side of equation 2.3 less us see that the metric is made up of two key components the first base measure sample log likelihood and the second term, the log-determinant or (volume correction), accounts for the change of volume induced by the transformation of the normalizing flows. In training the flow parameters are adjusted in order to maximize the log likelihood (or reduce negative log likelihood).

In a similar manner to that of (Cheung et al 2023) this project utilizes the masked autoencoder for distribution estimation (MADE) as a building block instead of the fully-connected layer. Using has many advantages including parallel density evaluations without typical sequential loop. This makes MAF fast to evaluate and train on parallel computing architectures such as Graphics Processing Units (GPUs).

Autoregressive models are a class of models where the conditional distribution of each variable in a

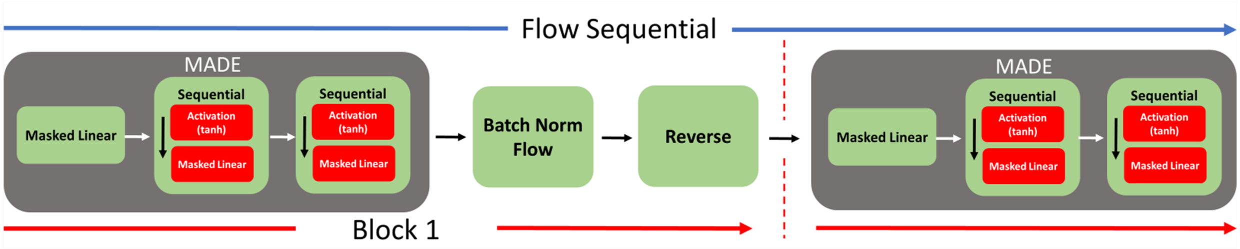


Figure 2-5: Flow diagram produced to illustrate the building blocks of MAF.

sequence depends on the previous variables. In the context of MAF, it means that the transformation of each element in the input data depends on the elements that came before it. In MAF, the autoregressive property is introduced through a masking mechanism. This means that for each input dimension (feature), only a subset of previous dimensions can influence it. This masking enforces a specific order in which the dimensions are transformed, preserving the invertibility of the flow. MAF uses an inverse autoregressive flow, meaning it models the inverse transformation, which maps data from the complex distribution back to the simple distribution (e.g., from data space to a Gaussian space). This inverse transformation allows for easy sampling from the learned complex distribution.

2.2.1 Fundamental Statistics

The normalizing flows technique can be powerful for Hierarchical Bayesian analysis. Bayes theorem[23] is a fundamental tool that allows us to update our knowledge based on new evidence. If we want to know the probability of A given event B we can find this distribution known as the posterior via the equation below.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.4}$$

Where $P(B|A)$ is the conditional probability of B given A is true. This also known as the likelihood due to the fact that $P(B|A) = L(A|B)$. $P(A)$ and $P(B)$ are the probabilities of A and B respectively, we think of $P(A)$ as the prior "our assumption" and $P(B)$ as the normalization. In order to link this our emulation to SEVN we rewrite Bayes theorem in terms of the input model hyperparameters $\lambda = (\alpha, Z, ...)$ giving the equation below,

$$p(\lambda|d) = \frac{p(d|\lambda) \cdot \pi(\lambda)}{p(d)} \tag{2.5}$$

- d: The data
- $p(d)$:The evidence
- $p(\lambda|d)$: Population posterior
- $p(d|\lambda)$: Likelihood of observing data given population model with hyperparameters λ
- $\pi(\lambda)$: The prior of the hyperparameters

We can expand the marginalized likelihood as follows.

$$p(d|\lambda) = \int p(d|\theta) p_{pop}(\theta|\lambda) d\theta \tag{2.6}$$

Where $p_{pop}(\theta|\lambda)$ represents the population probability of observing an event given the population model characterized by the hyperparameters. It is computationally expensive to calculate $p_{pop}(\theta|\lambda)$, this is why its we use machine learning technique of normalizing flows.

2.3 Modifications and optimizations

First, a pipeline was written to make the normalized flows compatible with SEVN BHBH merger data. The original implementation was too intensive to run locally, therefore supercomputer cluster resources were allocated from the DEMOBLACK group. However, after optimisation, almost all of the training could occur locally.

- **Preprocessing:** The SEVN BHBHm dataset was trimmed down to only M_0, M_1, Z and α. Since Z and α are both made up of a set repeating fixed values they were saved employing dictionary encoding. This almost eliminated the overhead of 1600000 numbers. The data was saved in Parquet files, the combined effect was that the input went from 237.9MB down to 11.4MB. This had a measurable effect on the training time.

- **Data loader:** The data set would not fit into local GPU RAM. A data loader was added which allows training in batches and lowered the overhead enough that CUDA could be used. This reduced training time by $\sim \times 5$. These optimizations enabled the subsequent hyperparameter tuning and deeper 1000 epoch training runs.

- **Early stopping:** The rate of loss change was monitored with a threshold to stop the model when it was not learning fast enough. This has the benefit of saving computation time and preventing over-training.

- **Run information storage(in JSON):** The training script was rewritten to store every variable in a dictionary including all user options from the argument parser(which was also altered to enable fully customizable runs).

- **SBATCH submission script**: This can automate the production and submission of SBATCH scripts on the cluster. This can be useful when running multiple training's simultaneously on the cluster like in hyperparameter tuning.

2.3.1 Hyperparameter tuning

Development of normalized flow tool set

Firstly the training script was rewritten in order to allow many different configurations to be specified to the argument parser. The complex setup of the run configuration was tracked each run gets a unique name and a JSON. In addition to this, a tool set was developed that can scan through all of the runs and find the best run, or search for specific configurations for example evaluate the run times or loss change per epoch or unit time.

In preparation for hyperparameter tuning and the complexity of tracking many runs a toolset was developed that would systematically log parameter combinations and the full set of run metrics. Firstly training script was rewritten in order to allow many different configurations to be specified to the argument parser. The results were labelled based on the user-varied parameters and a JSON was saved to the output directory. This systematic procedure enabled the development of tools that offer ease of use. For instance, tools to select the best run based on a metric like loss and configure PyTorch accordingly. Another example is the tool to select runs based on parameter criteria i.e. Z > 0.1 or simply compile a table of various previous runs.

Essential tools were also written that allow the user to easily generate samples, or probability distributions with just the original run JSON as mandatory input as well as various associated plotting tools. Tools were also written to apply masks i.e. remove a certain alpha value and also set the input and also apply logs to the input data.

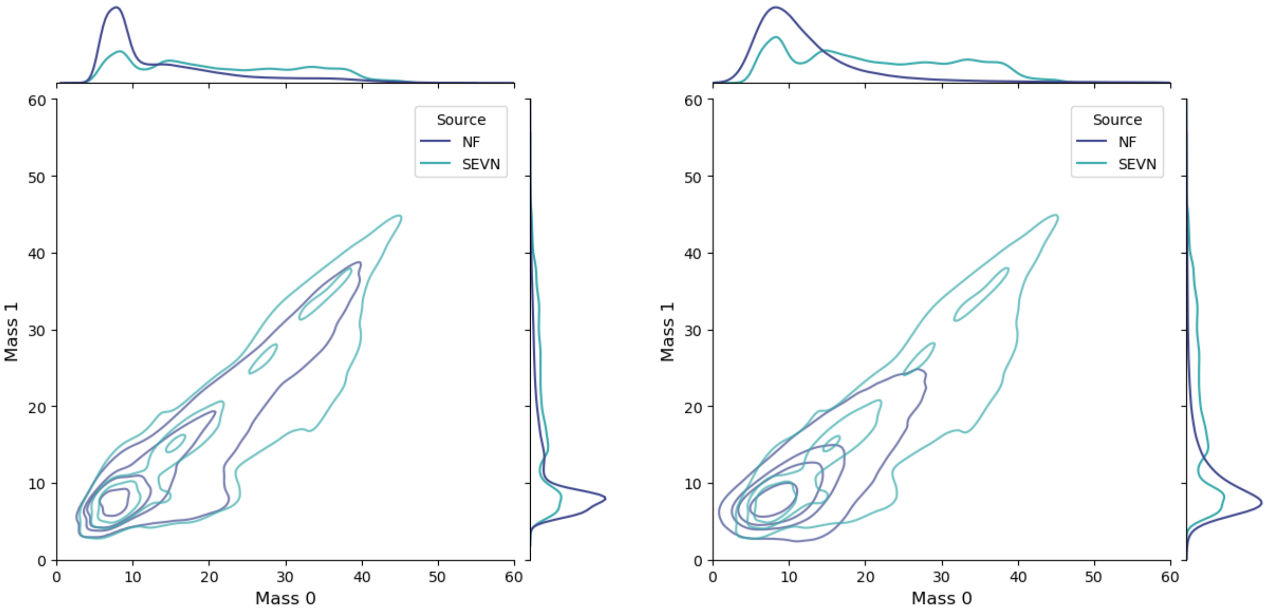
Tuning

The preliminary objective of this project to test the application of NF emulator for binary black hole mergers of SEVN, in order to tailor the algorithm to the problem it's is important to be quantitative in the way in which different configurations are run. The right set of hyperparameters can significantly increase the accuracy, and precision [24]. Users now have access to valuable metrics such as the rate of loss change per epoch, runtime and what epoch and time the training plateaued and the run had been stopped.

Chapter 3

Results

Over the course of this research, the normalized flows emulator for SEVN has developed significantly. This section shall detail the results, that in cumulation have tracked the progress that has helped make the technique more applicable to the SEVN use case. Earlier in the project the predicted distributions tended to be smooth and diffuse and a problem arose that negative masses would be predicted. This tightening of the learnt distribution is best illustrated by Figure 3-1a.



(a) Mass > 60 = 0.00024%, Mass < 0 = 0.0 %

(b) Mass > 60 = 0.135 % Mass < 0 = 0.0004 %

Figure 3-1: Mass distribution contours for 10%, 50%, 70% and 90% contour levels. Plot (a) is from the 1000 epoch 20 block run. (b) is the default model. The contour levels map to iso-proportions of the density. The contour levels are drawn at percentage density of the input data for example 80% would correspond to a small dense region.

3.1 Preliminary results

Table 3.1 gives the key default model run parameters. This model approximates the distribution of masses well as can be seen comparing the input data Figure 2-3 with the NF samples generated by 3-2. This is an exciting result, confirming the effectiveness of this Normalizing flows emulator. With this technique samples can be generated very quickly for instance it takes only 1.31s to emulate 20000 samples. This time is minuscule compared to the typical population synthesis stellar evolution even with a fast and advanced code like SEVN. Here the properties of these first optimized results are explored further.

Epochs	Activation	Hidden Layers	Blocks	Data Split
1000	Tanh	128	10	0.6, 0.2, 0.2

Table 3.1: The default values of the model after optimization. Dataspilt (train, validation, test) this split was used for all trainings.

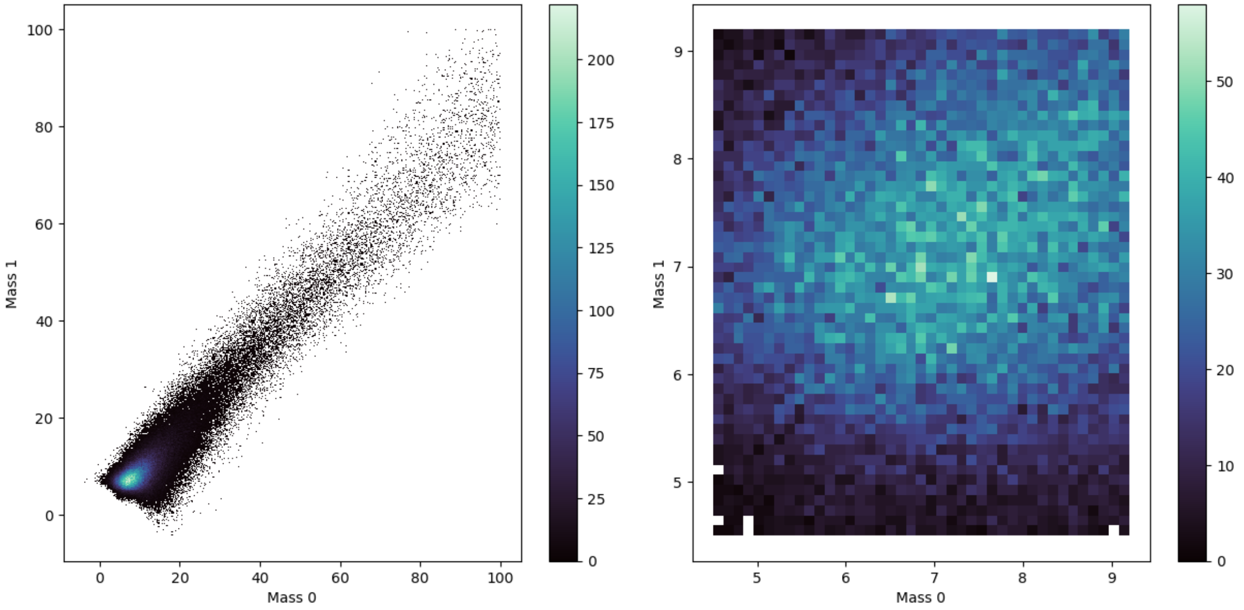


Figure 3-2: Histogram (2D) of NF samples ran with default values for 1000 epochs

The right plot of Figure 3.1 shows the previously reported smoothing effect of the emulator and the four distinct regions within the peak are not resolved. This is important as given the proximity of the frequency spike to zero, it can be difficult for the method not to produce any non-physical negative masses. This smoothing problem is known to be difficult for the normalizing flows technique. To some degree, it is expected that the samples lack the full complexity of the population synthesis derived samples. However, given the fact that we can learn more complex results from more training, hidden layers and blocks. The feasibility of utilizing the normalizing flow emulator with different configurations is explored further in the hyperparameter tuning results below.

Epochs	Activation	Blocks	Hidden	$\mathcal{L}_{valid}$	$\frac{d\mathcal{L}_{valid}}{dt}$ $[s^-1]$	Runtime $[s]$
276	relu	10	128	5.09	-0.000105	1200.0
116	tanh	20	200	5.29	-0.000116	1210.0
216	tanh	10	200	5.25	-0.000149	1200.0
151	tanh	20	128	5.31	-0.00015	1200.0
104	tanh	30	128	5.43	-0.000173	1210.0
79	tanh	30	200	5.56	-0.00029	1210.0
105	relu	30	128	5.45	-0.000382	1200.0
276	tanh	10	128	5.19	-9.17e-05	1200.0
218	relu	10	200	5.09	-9.81e-05	1200.0
81	relu	30	200	6.06	NA	1210.0
154	relu	20	128	5.99	NA	1200.0
117	relu	20	200	5.27	NA	1200.0

Table 3.2: Hyperparameter tuning results for run time $\sim$ 20 minutes. $\mathcal{L}_{valid}$ is the validation loss

3.2 Hyperparameter tuning

Attempts to solve the low model resolution lead naturally to a hyperparameter search to find a model that could offer greater resolution but also be trained effectively. The results of this search are in 3.2. The loss rate is calculated in order to gain insight into which models have plateaued are are beginning to in the time window. The intuition behind this table is that a simpler model will train faster and reach more epochs however won't have the resolving power to understand the complexity of the model. Conversely, while a complex model will have a greater resolving power in practical terms it may not be reasonable to use or train. As with any approach, this is a trade-off based on user demands and the resources available at the time. To approximate the effectiveness the models were all given 20 minutes to train.

As expected, Table 3.2 shows adding more blocks or hidden layers adds more computational complexity and therefore slows down training. Therefore it seems reasonable to assume that the ideal model is the one that has just enough complexity to capture the relationships in the data.

For the final model selection the run with the parameters given by Table 3.3. this model was selected as it had good results across the various metrics and had a good loss rate suggesting that it had not yet plateaued. The model had a low loss and will train relatively fast.

Epochs	Activation	Hidden Layers	Blocks	Data Split
1000	Tanh	128	20	0.6, 0.2, 0.2

Table 3.3: Final model parameter set

3.3 Final Model

It is interesting that in both plots of Figure 3-1 it looks as though there is an overestimation of the spike. This may be in part due to the preference of machine learning algorithms to prioritize high-density regions.

It's also important to note that for the comparison SEVN test data was used, the contours are a kernel density estimate of this true data and will struggle with spike smoothing. The 2D histogram in 2-3 of the full SEVN data is the closest reflection of what we should expect here. This is an exciting result that should be tested against a deeper physics-based likelihood estimation. The plots clearly show a significantly reduced problem of tail distribution, which caused a use case problem as when predicting unphysical samples with negative masses.

Chapter 4

Conclusion

In this thesis, the ability of the normalizing flows technique to learn the complex distributions of the parameters of binary black hole merger SEVN has been evaluated. This involved first setting up a pipeline that would allow the emulator used by Cheung et al.[1] to work with SEVN. Following this numerous optimizations were made increasing performance around $\sim \times 5$ for the same model on the same data. This boost in performance has enabled the deep 1000 epoch model to be used and trained. A tool set has also been developed that allows for in-depth hyperparameter tuning across many metrics. This work has resulted in an accurate model that can rapidly generate (20000 samples in 1.31s) SEVN mass distribution samples as illustrated in Figure 3-1a. This is a significant step that is useful for many applications including Monte Carlo experiments and Hierarchical Bayesian Analysis.

4.1 Future Developments

Going forward, there are many fascinating potential use cases of the work carried out during this thesis. The learned likelihood can be used to explore the relationships between the mass distributions given different combinations of the hyperparameters alpha and Z. Using the Kullback–Leibler divergence [25] would give us a measure of the statistical distance between these distributions. Also comparing a particular distribution to an amalgamated average distribution could give a systematic way to measure more unusual results. Furthermore, it would be very interesting to how this model parameter configuration performs on data from neutron star-black hole coalescences. The tuning toolbox has also enabled access to key metrics, it would be an interesting next step to implement a Bayesian hyperparameter optimization to further refine the model selection process.

Bibliography

[1] Damon H. T. Cheung, Kaze W. K. Wong, Otto A. Hannuksela, Tjonnie G. F. Li, and Shirley Ho. Testing the robustness of simulation-based gravitational-wave population inference. *Physical Review D*, 106(8), oct 2022.

[2] Shadia Habbal and Nathalia Alzate. Total solar eclipse chasers in search of the physics of the corona and the solar wind.

[3] Stefan Czesla, Sebastian Schröter, Christian P. Schneider, Klaus F. Huber, Fabian Pfeifer, Daniel T. Andreasen, and Mathias Zechmeister. PyA: Python astronomy-related packages, Jun 2019.

[4] Simon Boehm. Conditional density estimation with normalizing flows, Aug 2023.

[5] Albert Einstein. Näherungsweise Integration der Feldgleichungen der Gravitation. *Sitzungsberichte der K&ouml;niglich Preussischen Akademie der Wissenschaften*, pages 688–696, January 1916.

[6] B. P et al Abbott. Observation of gravitational waves from a binary black hole merger. *Phys. Rev. Lett.*, 116:061102, Feb 2016.

[7] B. P. Abbott et al. Multi-messenger observations of a binary neutron star merger. *The Astrophysical Journal*, 848(2):L12, oct 2017.

[8] The LIGO Scientific Collaboration, the Virgo Collaboration, the KAGRA Collaboration, and R. Abbott et al. Gwtc-3: Compact binary coalescences observed by ligo and virgo during the second part of the third observing run, 2021.

[9] Peter Bodenheimer. *Principles of Star Formation.* Astronomy and Astrophysics Library. Springer Nature, Berlin, Heidelberg, 1. aufl. edition, 2011.

[10] Steven A. Balbus. Simplified derivation of the gravitational wave stress tensor from the linearized einstein field equations. *Proceedings of the National Academy of Sciences - PNAS*, 113(42):11662–11666, 2016.

[11] Michele Maggiore. *Gravitational Waves: Volume 1: Theory and Experiments.* Oxford University Press, 10 2007.

[12] Giuliano Iorio, Michela Mapelli, Guglielmo Costa, Mario Spera, Gastó n J Escobar, Cecilia Sgalletta, Alessandro A Trani, Erika Korb, Filippo Santoliquido, Marco Dall'Amico, Nicola Gaspari, and Alessandro Bressan. Compact object mergers: exploring uncertainties from stellar and binary evolution with scpsevn/scp. *Monthly Notices of the Royal Astronomical Society*, 524(1):426–470, jun 2023.

[13] H. Bondi and F. Hoyle. On the mechanism of accretion by stars. , 104:273, January 1944.

[14] Dina. Prialnik. *An introduction to the theory of stellar structure and evolution / Dina Prialnik.* Cambridge University Press, Cambridge, 2nd ed edition, 2010.

[15] G. E. Soberman, E. S. Phinney, and E. P. J. van den Heuvel. Stability criteria for mass transfer in binary stellar evolution, 1997.

[16] Hongwei Ge, Michael S. Hjellming, Ronald F. Webbink, Xuefei Chen, and Zhanwen Han. ADIABATIC MASS LOSS IN BINARY STARS. i. COMPUTATIONAL METHOD. *The Astrophysical Journal*, 717(2):724–738, jun 2010.

[17] R. F. Webbink. Double white dwarfs as progenitors of R Coronae Borealis stars and type I supernovae. , 277:355–360, February 1984.

[18] Erika Böhm-Vitense. *Introduction to Stellar Astrophysics*. Cambridge University Press, Cambridge, GBR, 1989.

[19] W. Baade. The Resolution of Messier 32, NGC 205, and the Central Region of the Andromeda Nebula. , 100:137, September 1944.

[20] J. L. Puget and J. Heyvaerts. Population III stars and the shape of the cosmological black body radiation. , 83(3):L10–L12, March 1980.

[21] Jorick S. Vink, A. de Koter, and H. J. G. L. M. Lamers. Mass-loss predictions for o and b stars as a function of metallicity. *Astronomy &amp Astrophysics*, 369(2):574–588, apr 2001.

[22] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation, 2018.

[23] Thomas Bayes and null Price. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

[24] Hilde J. P Weerts, Andreas C Mueller, and Joaquin Vanschoren. Importance of tuning hyperparameters of machine learning algorithms. 2020.

[25] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

Appendix A

Appendix

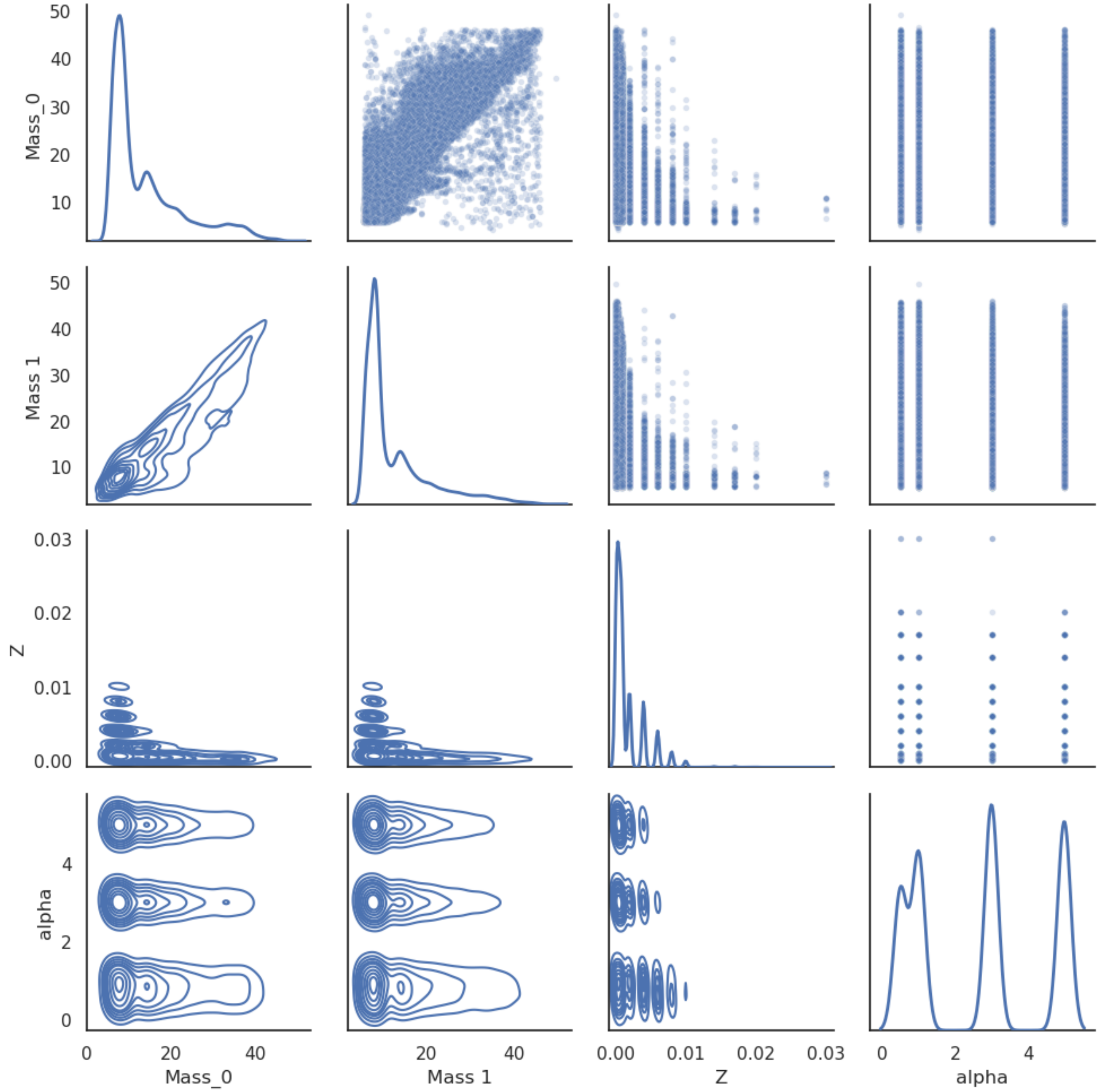


Figure A-1: Joint plot comparing M_0, M_1, Z and α for SEVN input data. Diagonal region (top left to bottom right) 1D density plots. Below diagonal is 2D density plots. Finally above the main diagonal are the scatter plots.

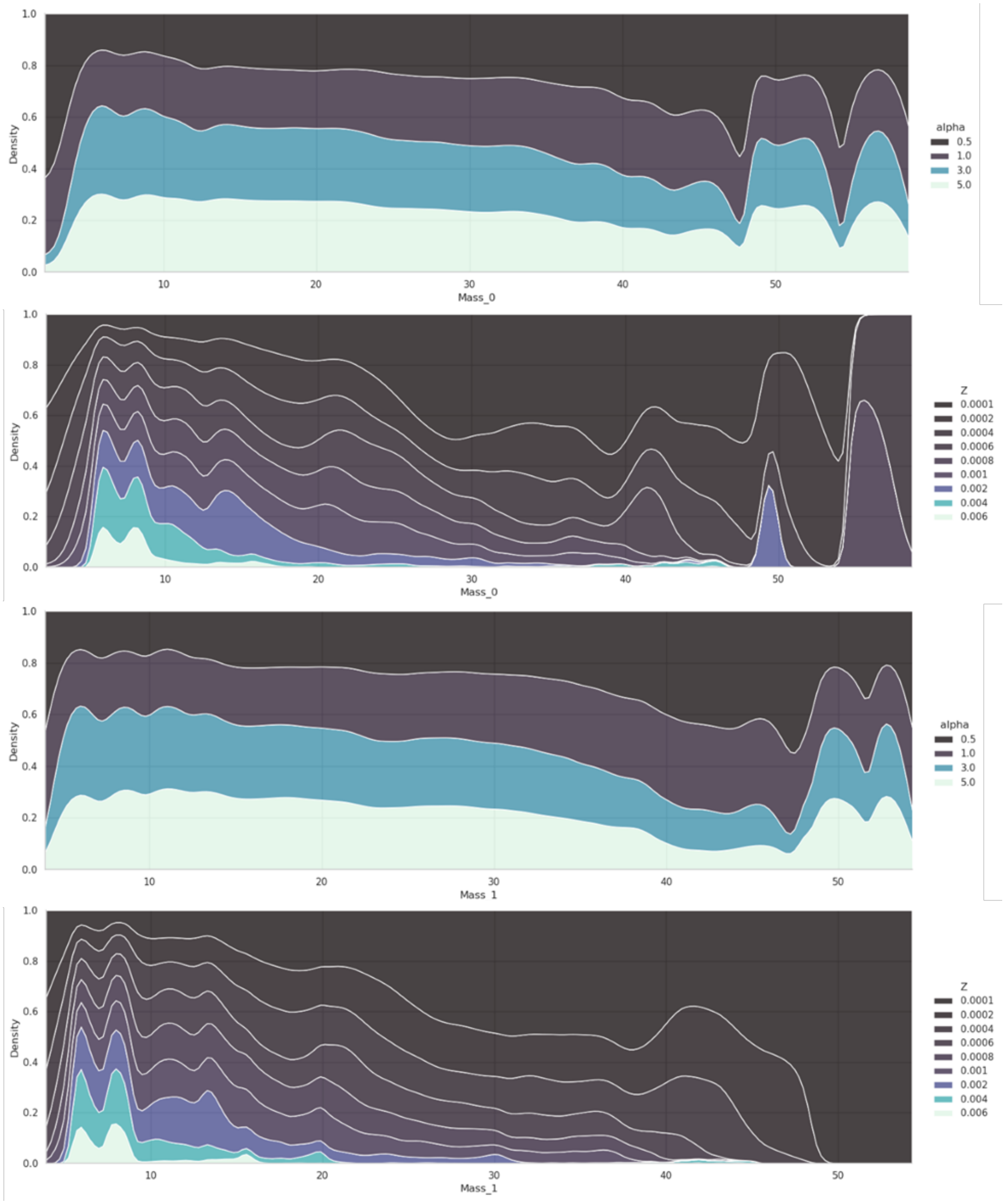


Figure A-2: Plots showing input data density distributions from SEVN BHBHm. The largest Z values had very small amounts of data so were cut in order to make plots readable

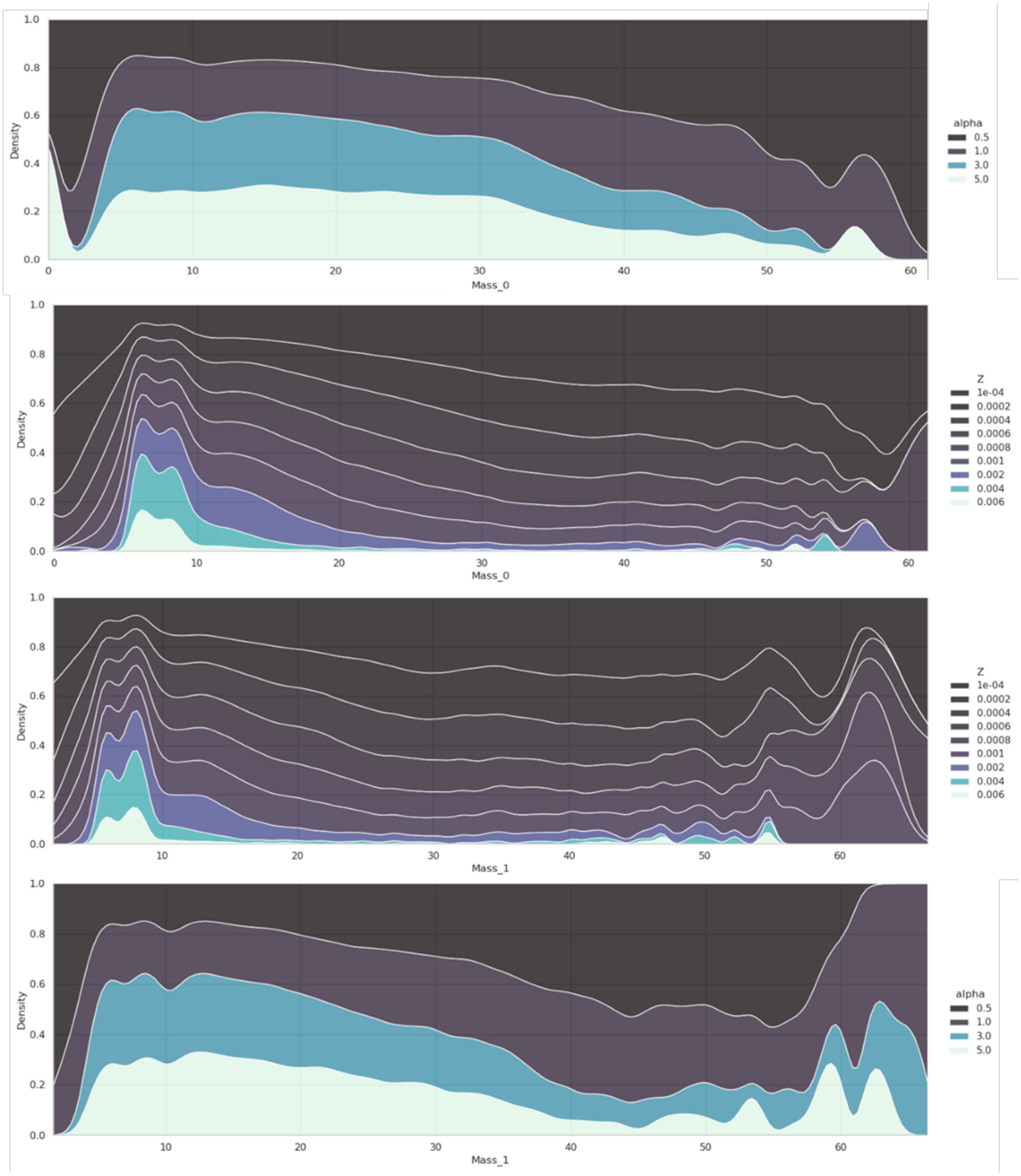


Figure A-3: Plots showing input data density distributions from NF BHBHm. The largest Z values had very small amounts of data so were cut in order to make plots readable

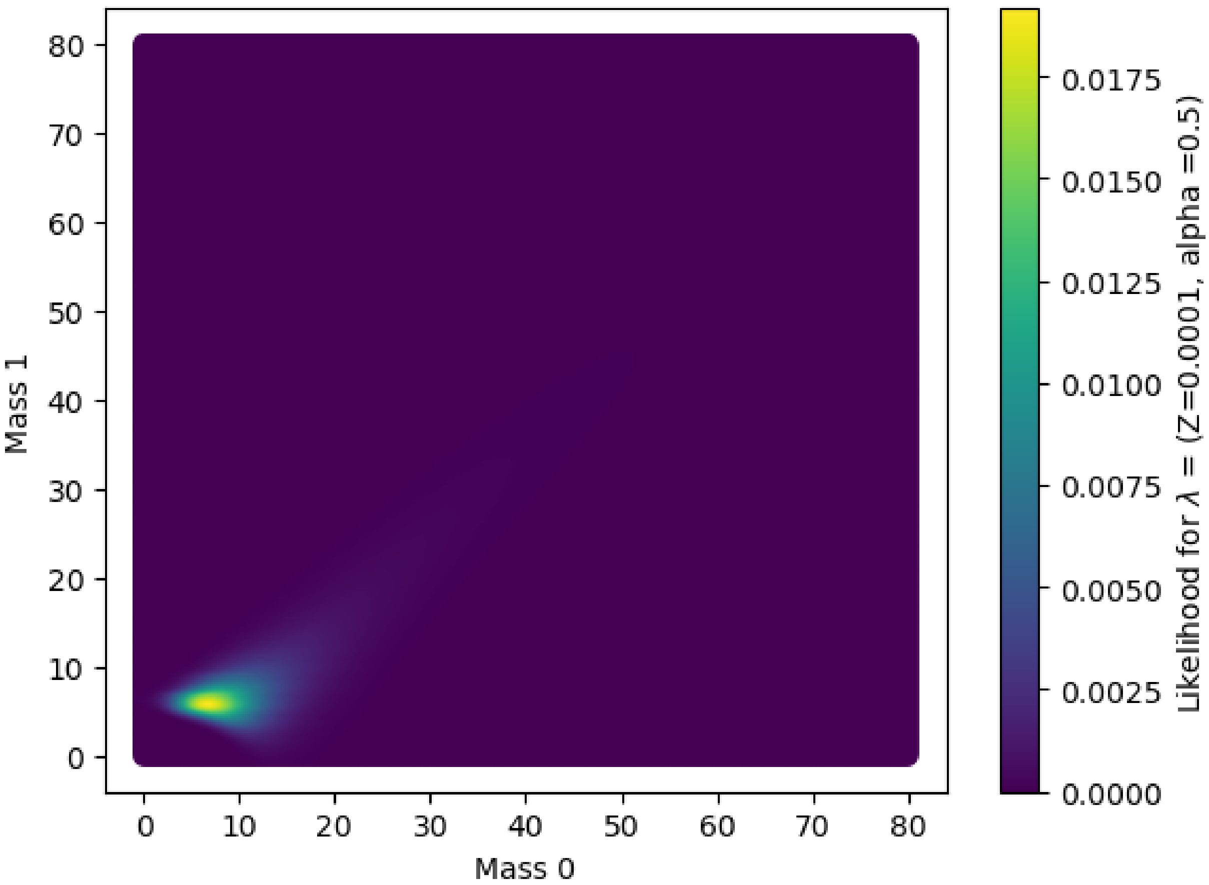


Figure A-4: Plot of Likelihood from default 1000 epoch model