SAPIENZA
UNIVERSITÀ DI ROMA

Machine Learning enhanced Cosmological and Astrophysical inference with Binary Black Holes observed by LIGO-Virgo-KAGRA

Physics Department
Master of Science in Physics (LM-17)

Candidate
Arianna Scarpa
ID number 1871231

Thesis Advisor
Dr. Simone Mastrogiovanni

Thesis Referee
Prof.ssa Raffaella Schneider

Academic year
2024-2025

Graduation Session
29th October 2025

Master's Thesis. Sapienza University of Rome.
Author: Arianna Scarpa

Email: scarpa.1871231@studenti.uniroma1.it

Dedicated to
the loving memory of
Daniele C.

Abstract

Although we have been studying the cosmic expansion since about 100 years, many unsolved puzzles are yet to be solved. One of the most famous problems is the so-called Hubble tension: measurements of the Hubble constant H_0, the universe expansion rate today, are in a 5σ tension between direct observations of supernovae and indirect measures from the Cosmic Microwave Background. Gravitational Waves (GWs) from Compact Binary Coalescences (CBCs) are emergent cosmological probes that can help to leverage this tension. Gravitational Waves allow us to measure directly the luminosity distance of the source, that if provided by a redshift estimation, can be used to measure the cosmic expansion, including H_0. However, the redshift can not be directly extracted from the GW signal as it is totally degenerate with the source masses. Current cosmological analyses using Binary Black Holes (BBHs), jointly fit for the cosmic expansion and their source mass spectrum. Typically, the mass spectrum is modeled using analytical prescriptions such as power laws and Gaussians that could not capture the mass-redshift interplay of BBHs. In this work, I derive and introduce a non-parametric model based on Normalizing Flows, a machine learning tool to approximate complex distributions that can accommodate for non-trivial relations between the mass and redshift of BBHs. Normalizing Flows trained on realistic simulated catalogs accurately reconstruct underlying population features, eliminating the H_0 bias observed with traditional phenomenological models. Validation on mock catalogs and astrophysical synthesis catalogs demonstrates that this approach recovers the injected value for the Hubble constant. I also show that this methodology can be used to estimate formation channels for BBHs.

Contents

Introduction

Gravitational Wave (GW) sources are rapidly becoming promising sources to measure the cosmic expansion. This is particularly relevant, given unsolved puzzles related to cosmology such as the "Hubble tension": measurements of the Hubble constant (the expansion rate of the Universe today) differs between direct measurements in the local universe and indirect measurements with the Cosmic Microwave Background. Differently from Standard Candles, such as Supernovae, GWs from Compact Binary Coalescences (CBCs) provide a direct estimate of the luminosity distance to the source, without the need for electromagnetic distance measurements. Indeed, when no electromagnetic counterpart is detected, the redshift can be estimated statistically and inference on cosmological parameters can be performed, especially the Hubble constant. Recent results from the GWTC-4 catalog, which includes 153 Binary Black Hole (BBH) mergers, report a measurement of H_0 compatible with Cosmic Microwave Background and Supernovae estimates.

The LIGO-Virgo-KAGRA (LVK) collaboration has shown that the primary mass distribution of BBHs is far from trivial: it features a peak around 33 solar masses, followed by a sharp decline. Only a few percent of the sources have masses above 45 solar masses. The spin distributions and merger rates also carry astrophysical information and population studies are becoming very precise thanks to the growing number of detections.

Despite these advances, population modeling remains a challenge because modeling compact binaries formation channels is intrinsically complex. Recent works have highlighted tensions in the literature regarding correlations between binary parameters such as mass, spin, and redshift. These correlations have a non negligible impact on cosmological inference, yet they are often neglected or oversimplified in current models.

Inference results on cosmological parameters are highly dependent on the assumed model for the underlying population of compact binaries. Phenomenological models such as power laws and Gaussian peaks or broken power laws do not capture the possible redshift evolution of the mass distribution and this mismodeling can introduce a systematic bias in the inferred value of H_0. This is already visible in real data: estimates of H_0 vary depending on the assumed population model.

This work directly addresses this issue. I developed a method to model the population of compact binaries within the ICAROGW pipeline that uses Normalizing Flows,

which are powerful machine learning techniques capable of learning complex distributions. The goal is to let the model learn the true structure of the population from data, instead of imposing a predefined shape. Data might come from mock catalogs or astrophysically simulated fiducial catalogs. This flexibility turns out to be crucial for accurate cosmological inference.

The thesis is organized as follows. In Chapter 1 I introduce the scientific framework, in particular I show how CBCs can be used as cosmological probes and the possible systematics that can be introduced with current parametric models. In Chapter 2 I present the Normalizing Flows technique, I propose a non-parametric model to replace the current parametric models and I implement it into the ICAROGW pipeline. In Chapter 3 I validate the model using a mock catalog with a redshift evolving population. Finally, in Chapter 4 I test the model on an astrophysical synthesis fiducial catalog by constraining cosmology and population properties with both GWTC-3 and simulated data.

Chapter 1

Scientific Framework

1.1 Observing Compact Binary Coalescences

Compact Binary Coalescences (CBCs) are observed via gravitational wave (GW) emission. Besides confirming Einstein's General Relativity, and directly testing it, populations of CBCs opened a new window to study cosmology and astrophysics.

Gravitational wave events have been collected since the first observation GW150914 on September 14th of 2015 [1]. The Advanced LIGO, Advanced Virgo and KAGRA (LVK) detectors have been conducting coordinated runs named O1, O2, O3, O4 [1] and collecting data of the signals in *Gravitational Wave Transient Catalogs* (GWTC). The most recent and public CBCs catalog is GWTC-4.0 [3], which contains all gravitational wave detection data up to the first part of the O4 run, which ended in January 2024 and that is called O4a. A total of 158 gravitational wave events have been detected [4]. Fig. 1.1 shows the cumulative number of detections as a function of time. The increased number of detections is due to the increased sensitivity of GW detectors.

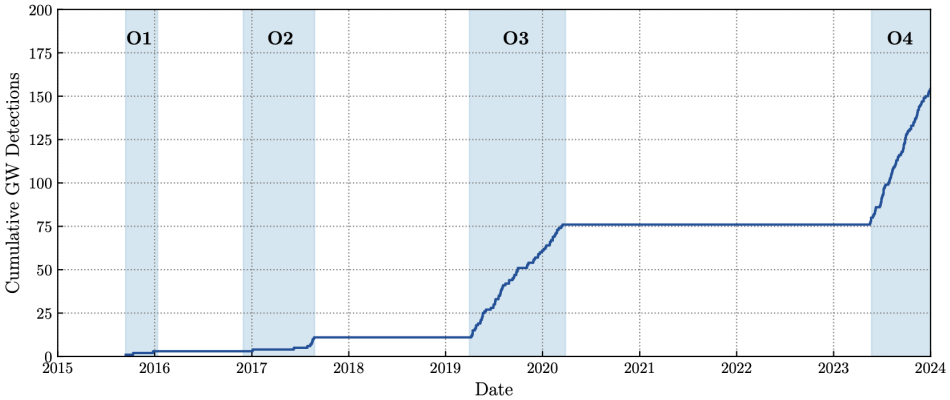

Figure 1.1: Image taken from [5] showing the cumulative number of observations during O1, O2, O3, O4a runs. Between each run, all detectors are upgraded and their sensitivity is increased.

[1] The last part of the O4 run, which is called O4c, is about to end in November 2025 [2].

CBCs are typically classified into three categories: Binary Black Holes (BBHs), Binary Neutron Stars (BNSs) and Neutron Star Black Holes (NSBHs). This classification is set on mass boundaries. Sources with masses between 1-3 $M_\odot$ are classified as Neutron Stars (NSs), while if masses are above 5 $M_\odot$, they are classified as Black Holes (BHs). This classification assumes a perfect knowledge of formation channels and it is never used when fitting for astrophysical and cosmological properties of CBCs.
This thesis aims at building new data analysis techniques to jointly infer cosmological and astrophysical properties of BBHs. In the following sections I will lay the basic knowledge from which this thesis is beginning.

1.2 Compact binaries as cosmological probes

Cosmology is the branch of Physics that studies the Universe expansion and its evolution on large scales. We will now introduce the background concepts and notation that are used in this work.

1.2.1 Friedmann's equations and ΛCDM model

Observations of nearby galaxies have shown that their spectral lines are systematically shifted towards longer wavelengths. This effect, called *redshift*, increases with distance and it indicates that the Universe is expanding. In 1929 Hubble and Lemaître observed a linear relation between the recession velocity of galaxies and their distance. This law is known as the *Hubble-Lemaître* law and it is valid at low redshift:

$$v \simeq H_0\, d \qquad \text{for } \; z \lesssim 0.1 \tag{1.1}$$

At larger distances, the dynamics of the cosmic expansion must be described with Einstein's General Relativity, in particular by the Friedmann equations. Indeed, the Hubble law can be formally derived from the *Friedmann-Lemaître-Robertson-Walker* (FLRW) metric [6]:

$$ds^2 = -dt^2 + a^2(t)\left[\frac{dr^2}{1-kr^2} + r^2 d\Omega^2\right] \tag{1.2}$$

where k is the spatial curvature, r is the radial coordinate and $d\Omega^2 = d\theta^2 + \sin^2\theta d\phi^2$ is the infinitesimal metric of the unit 2-sphere. $a(t)$ is the scale factor that evolves with cosmic time t and it is defined through the redshift:

$$z = \frac{\lambda^{\text{obs}} - \lambda^{\text{emitt}}}{\lambda^{\text{emitt}}} \equiv \frac{1}{a(t)} - 1 \tag{1.3}$$

where $\lambda^{\text{obs}} - \lambda^{\text{emitt}}$ represents the shift in the observed and the emitted spectral lines, caused by the Universe expansion.
By imposing Einstein's equations, the behavior of the scale factor $a(t)$ can be found. It is described by the first and the second *Friedmann equations*:

$$\begin{cases} \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho - \frac{k}{a^2} \\ \frac{\ddot{a}}{a} = -\frac{4\pi G}{3}\left(\rho + 3p\right) \end{cases} \tag{1.4}$$

where ρ is the total energy density of the Universe, including matter (dominant in the late universe), radiation (dominant in the early universe) and dark energy (dominant today). p is the total pressure and k represents the spatial curvature of the universe: $k = 0$ corresponds to a flat geometry, $k > 0$ to a closed Universe and $k < 0$ to an open Universe.
The *density parameter* Ω_i for a given component i (such as matter, radiation or dark energy) can be defined as:

$$\Omega_i \equiv \frac{8\pi G}{3H_0^2}\rho_i = \frac{\rho_i}{\rho_{\rm crit}} \tag{1.5}$$

where $\rho_{\rm crit}$ is the critical density:

$$\rho_{\rm crit} = \frac{3H_0^2}{8\pi G} \tag{1.6}$$

In the following, we chose the *flat Cold Dark Matter* (flat ΛCDM) cosmological model in which the spatial curvature is set to $k = 0$ and the dominant form of dark energy is given by the cosmological constant Λ. The radiation component is neglected. Thus one can define a dimensionless Hubble parameter as:

$$\begin{aligned} E(z)^2 \equiv \left(\frac{H(z)}{H_0}\right)^2 &= \Omega_m(1+z)^3 + \Omega_\Lambda \\ &= \Omega_m(1+z)^3 + (1-\Omega_m) \end{aligned}$$

In order to convert the observed redshift into a distance, we need astrophysical sources that can act as distance indicators. *Standard candles* are astrophysical sources whose intrinsic luminosity L can be accurately measured. By comparing the intrinsic luminosity with the observed flux of photons from the candles, we can get a distance measurement using the relation [6] [7]:

$$F = \frac{L}{4\pi d_L^2} \tag{1.7}$$

where $d_{\rm L}$ is the *luminosity distance*, which is a distance that does not change with the cosmic expansion, indeed it can be used to compare distances and volumes at different cosmic epochs. $d_{\rm L}$ is a fundamental quantity since we directly measure it from standard candles. It is defined as:

$$d_{\rm L}(z) = \frac{c\,(1+z)}{H_0}\int_0^z \frac{dz'}{E(z')} \tag{1.8}$$

and its redshift derivative is:

$$\frac{\partial d_{\rm L}(z)}{\partial z} = \frac{d_{\rm L}(z)}{1+z} + \frac{c(1+z)}{H_0 E(z)} \tag{1.9}$$

The *comoving distance* $d_c(z)$, that represents the distance between the observer and a source at redshift z, is related to the luminosity distance $d_{\rm L}$ with:

$$d_c(z) = \frac{d_L(z)}{1+z} \tag{1.10}$$

It furnishes a direct definition for the *comoving volume* V_c, enclosed up to a redshift z:

$$V_c(z) = \frac{4}{3}\,\pi\, d_c^3(z) \tag{1.11}$$

its differential with respect to the redshift is given by:

$$\frac{dV_c}{dz} = 4\pi \frac{c}{H_0} \frac{d_c^2(z)}{\sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}} \tag{1.12}$$

Current Hubble constant estimations use early and late Universe measurements. Early Universe measurements are inferred indirectly from the Cosmic Microwave Background by the *Planck* experiment [8]. Late Universe measurements are based on Cepheids and type Ia Supernovae and are obtained from the SH_0ES experiments [9]. The measurements estimates different values of H_0, leading to a significant tension, known as the *Hubble tension* [10]. Fig. 1.2 illustrates the Hubble tension by comparing the evolution through the years of the H_0 measurements.

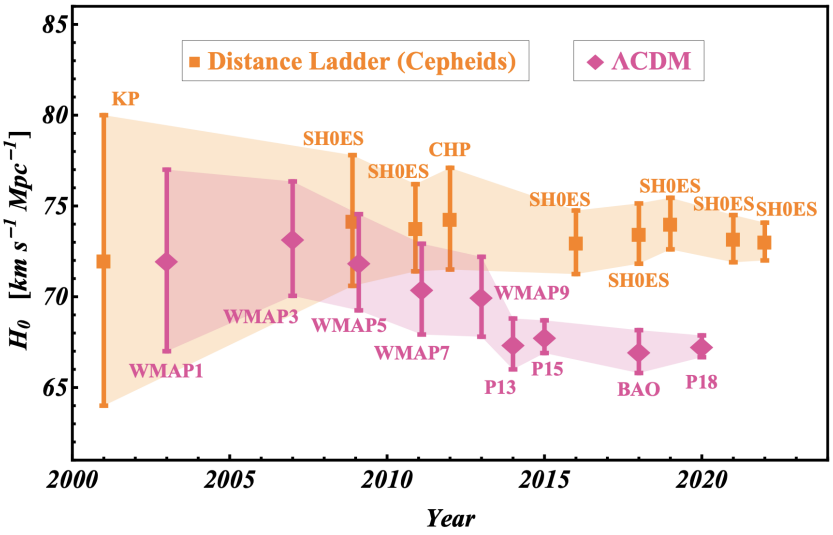

Figure 1.2: Image taken from [10] which compares the early Universe estimates (in pink) from the CMB estimation, based on the ΛCDM model, and the late Universe estimates (in orange) based on Cepheids and type Ia supernovae.

1.2.2 Introduction to Gravitational Wave Cosmology

The GW strain of CBCs signals is measured at the detector and it is caused by the distortion of spacetime produced by the propagation of the GW itself. We provide a more detailed discussion in Sec. A.1 of GW theory. Both of the two GW polarizations can be decomposed into an amplitude part and a phase part [11] [7]:

$$h(t_d) = h_0(t_d, d_\mathrm{L})\; e^{i\phi(t_d, \mathcal{M}_d)} \tag{1.13}$$

here, for brevity, we have not reported the dependence of the waveform on other parameters such as spins and sky position. t_d represents the observing detector

time. h_0 is the signal amplitude and it depends on t_d and on the luminosity distance of the source d_{L}. ϕ is the signal phase and it depends on t_d and on the detector chirp mass $\mathcal{M}_d$, which is defined as:

$$\mathcal{M}_d = \frac{(m_{1d}\ m_{2d})^{3/5}}{(m_{1d} + m_{2d})^{1/5}} \tag{1.14}$$

where m_{1d} and m_{2d} are the CBC masses, observed at the detector. This implies that we can extract from the CBC signal, the direct estimation of $\mathcal{M}_d$ (from the signal phase evolution) and of d_{L} (from the signal amplitude).
Since the CBC happens at cosmological distances, then the detector chirp mass is the redshifted source chirp mass $\mathcal{M}_s$:

$$\mathcal{M}_d = (1+z)\mathcal{M}_s \tag{1.15}$$

Eq. 1.15 clearly highlights a crucial point: it is not possible to disentangle the redshift from the detector chirp mass. Thus, to use GW signals as standard sirens, additional redshift information is needed because it can not be determined from the GW signal itself.
Three methodologies allow us to estimate the redshift, hence allowing to exploit GWs signals as cosmological probes [7]:

- *spectral sirens* analysis uses the redshift information coming from the gravitational wave signal itself [12]. It exploits the detector-source mass relation. By assuming that the mass spectrum of BBHs does not evolve in redshift, we can jointly fit for cosmology and the mass spectrum itself. This is the method on which this thesis focuses.

- *spectral sirens with galaxy catalog* (or *dark sirens*) analysis obtains redshift information using galaxy surveys [12]. This method associates the GW sky localization with host galaxies catalogs whose redshifts are estimated with spectroscopic or photometric measurements.

- *bright sirens* analysis is allowed only if an EM counterpart is detected, in such a way that the host galaxy can be identified and the redshift can be measured. Currently, only one GW event has a measured EM counterpart and it is a BNS merger, GW170817 [13].

Gravitational Wave cosmology is particularly relevant in the current context of the *Hubble tension* as it provides an independent methodology to measure cosmological parameters and possibly help to better understand the mismatching estimations of H_0.

1.2.3 Phenomenological model for the masses distributions of BBHs

The detector rate of CBCs mergers can be written as:

$$\frac{dN}{dtd\theta_d}(\Lambda) = \frac{dN}{dd_L dm_{1d} dm_{2d} d\chi dt_d}(\Lambda) \tag{1.16}$$

where N is the number of CBCs sources in the Universe, Λ are the population parameters, θ_d are the GW parameters, observed at the detector: the luminosity

distance d_L, the detector frame masses m_{1d} m_{2d} and the detector time t_d. In this work, the spins χ are neglected. The *spectral sirens* analysis is performed exploiting the change of variables from the detector frame to the source frame:

$$m_{1d} = m_{1s}(1+z) \tag{1.17}$$
$$m_{2d} = m_{2s}(1+z) \tag{1.18}$$
$$dt_d = dt_s(1+z) \tag{1.19}$$

The luminosity distance d_L can be expressed as a function of the redshift z, following Eq. 1.8. In this way, the CBCs rate becomes:

$$\frac{dN}{dd_L dm_{1d} dm_{2d} dt_d} = \frac{dN}{dz dm_{1s} dm_{2s} dt_s} \frac{1}{1+z} \frac{1}{|J_{d\to s}|} \tag{1.20}$$

where the determinant of the Jacobian of the change of variables is given by:

$$|J_{d\to s}| = \left| \frac{\partial d_L}{\partial z}(1+z)^2 \right| \tag{1.21}$$

the derivative of the luminosity distance with respect to the redshift follows Eq. 1.9. Expressing the rate as a function of the comoving volume (Eq. 1.11), its deriavtive with respect to the redshift appears and follows Eq. 1.12. Then, Eq. 1.20 becomes:

$$\frac{dN}{dz dm_{1s} dm_{2s} dt_s} \frac{1}{1+z} \frac{1}{|J_{d\to s}|} = \frac{dN}{dV_c dm_{1s} dm_{2s} dt_s} \frac{dV_c}{dz} \frac{1}{1+z} \frac{1}{|J_{d\to s}|} \tag{1.22}$$

By defining the following quantities:

$$\frac{1}{N} \frac{dN}{dm_{1s} dm_{2s}} \coloneqq p_{\rm pop}(m_{1s} m_{2s} | \Lambda) \tag{1.23}$$
$$\frac{dN}{dV_c dt_s} \coloneqq R_0 \Psi(z, \Lambda) \tag{1.24}$$

where $p_{\rm pop}(m_{1s} m_{2s}|\Lambda)$ is a normalized prior distribution for the primary and secondary CBCs masses. R_0 is the CBCs merger rate per year and comoving volume. $\Psi(z, \Lambda)$ describes the rate evolution in redshift, and it is such that $\Psi(z=0, \Lambda) = 1$. Note that Λ represents the population parameters, but each distribution (e.g. the masses distribution or the redshift distributions) depends on different subset of these parameters. For simplicity, they are denoted as Λ, but they vary depending on the quantity and they never intersect.

The final CBCs rate can be written as:

$$\frac{dN}{dd_L dm_{1d} dm_{2d} dt_d} = R_0 \; p_{\rm pop}(m_{1s} m_{2s}|\Lambda) \Psi(z, \Lambda) \frac{dV_c}{dz} \frac{1}{1+z} \frac{1}{|J_{d\to s}|} \tag{1.25}$$

It is worth noticing that the masses distribution does not depend on the redshift. When modeling $p_{\rm pop}$, one can choose analytical parametric models such as combination of Broken-PowerLaw and Gaussian Peaks [14]; this approximation is the so-called *phenomenological model*. For some of the benchmarking in this thesis, we adopt the POWERLAW + PEAK model.

The joint prior distribuiton for the primary and secondary source masses takes into account the fact that the secondary mass is by definition smaller than the primary mass. The joint distribution is thus given by the product of the prior of the primary mass and the conditional prior for the secondary mass, given the primary mass:

$$\pi\left(m_{1\mathrm{S}}, m_{2\mathrm{S}}|\Lambda\right) = \pi\left(m_{1\mathrm{S}}|\Lambda\right)\pi\left(m_{2\mathrm{S}}|m_{1\mathrm{S}}, \Lambda\right) \tag{1.26}$$

The prior for the primary mass can be chosen to be a mixture of a truncated POWERLAW and a Gaussian PEAK, weighted by the *mixture fraction* ϕ. On the other hand, the conditional prior for the secondary mass is given by a truncated PowerLaw:

$$\begin{aligned}\pi\left(m_{1\mathrm{S}} \mid m_{\min}, m_{\max}, \alpha\right) = (1-\phi)\,\mathcal{P}\left(m_{1\mathrm{S}} \mid m_{\min}, m_{\max}, -\alpha\right) \\ + \phi\,\mathcal{G}_{[m_{\min}, m_{\max}]}\left(m_{1\mathrm{S}} \mid \mu, \sigma\right)\end{aligned} \tag{1.27}$$

$$\pi\left(m_{2\mathrm{S}}|m_{\min}, m_{1\mathrm{S}}, \beta\right) = \mathcal{P}\left(m_{2\mathrm{S}}|m_{\min}, m_{1\mathrm{S}}, \beta\right) \tag{1.28}$$

where

$$\mathcal{P}(x|m_{\min}, m_{\max}, \alpha) = \begin{cases} \frac{x^{\alpha}}{\mathcal{N}_{PL}} & \text{if} \quad m_{\min} < x < m_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{1.29}$$

$$\mathcal{G}_{[m_{\min}, m_{\max}]}(x|\mu\sigma) = \begin{cases} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\mathcal{N}_G \sigma\sqrt{2\pi}} & \text{if} \quad m_{\min} < x < m_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{1.30}$$

here, $\mathcal{N}_{PL}$ and $\mathcal{N}_G$ are the two normalization factors that can be found in [15]. A smoothing factor can be applied to gradually suppress the low mass end of the distribution at its minimum to obtain a smoother transition instead of a steep cutoff. It is implemented with a sigmoid window function with smoothing parameter δ_m [15]. A visualization of this model can be found in Fig. 1.3.

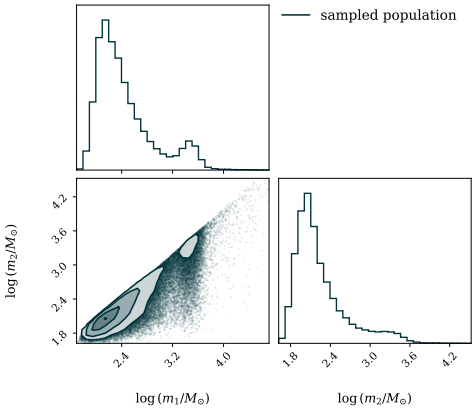

Figure 1.3: Samples for the BBH primary and secondary source masses distribution following the phenomenological model POWERLAW + PEAK (Eq. 1.27). We chose the parameters to be compatible with the GWTC-3 results [16]: $\alpha = 3.78$, $\beta = 0.81$, $m_{\min} = 4.98$, $m_{\max} = 112.5$, $\mu = 32.27$, $\sigma = 3.88$, $\lambda = 0.03$.

1.3 Latest spectral sirens results

In this section I summarize the most recent cosmological results, obtained with the spectral sirens method on the GWTC-4.0 catalog [17].

The spectral siren analysis was based on different parametric descriptions of the primary source mass. In particular, two phenomenological models were employed: the POWERLAW+PEAK and the MULTI PEAK, in which the latter is given by a power law and two Gaussian peaks, instead of one.

These models, as all current parametric models, are not assumed to evolve in redshift. Indeed, in spectral sirens method, redshift information comes from the fraction of events lying near sharp features in the source frame mass distribution. These sharp features are the Gaussian peaks or the high mass cutoff. They provide stable reference points in the mapping between the detector and source frame masses.

The latest GW cosmology study [17], as well as the previous study on the GWTC-3 catalog [13], show that, in spectral analysis, the H_0 estimation is strongly influenced by the choice of the population model. Indeed, Fig. 1.4 shows the difference in the H_0 estimations, obtained assuming the POWERLAW+PEAK and the MULTI PEAK. In particular, the POWERLAW+PEAK pushes H_0 to higher values and this is caused by the single peak, that alone is unable to correctly model the population.

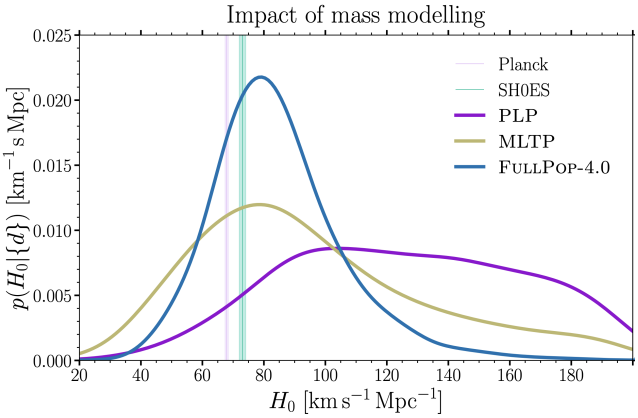

Figure 1.4: Image taken from [17], showing posterior distributions for the Hubble constant obtained with the spectral sirens method. Different population models were used for the primary mass distribution: the POWERLAW+PEAK in purple the and MULTI PEAK in yellow. The blue line represents the FULLPOP-4.0 model, which models the full CBCs population and that we do not focus on in this work.

For completeness, we also report the best results found in the latest cosmological study [17], in which H_0 was estimated using 142 GW events (141 dark sirens and 1 bright siren) from the GWTC-4.0 catalog. At the 68.3% credibility interval, H_0 was constrained to:

$$H_0 = 76.6^{+13.0}_{-9.5} \text{ km s}^{-1}\text{Mpc}^{-1} \tag{1.31}$$

To conclude our discussion, we also report the results of the latest population studies on the GWTC-4.0 data [4]. Different parametric descriptions of the source frame primary mass distributions were chosen for the BBH masses distributions. In particular, the best models were the BROKEN POWERLAW + 2 PEAKS and a more flexible model called B-SPLINE, which we will not focus on in this work. Fig. 1.5 shows the merger rate as a function of the primary source mass, described by the two best parametric models. Finally, they are compared with the fiducial model of GWTC-3, the POWERLAW+PEAK [16].

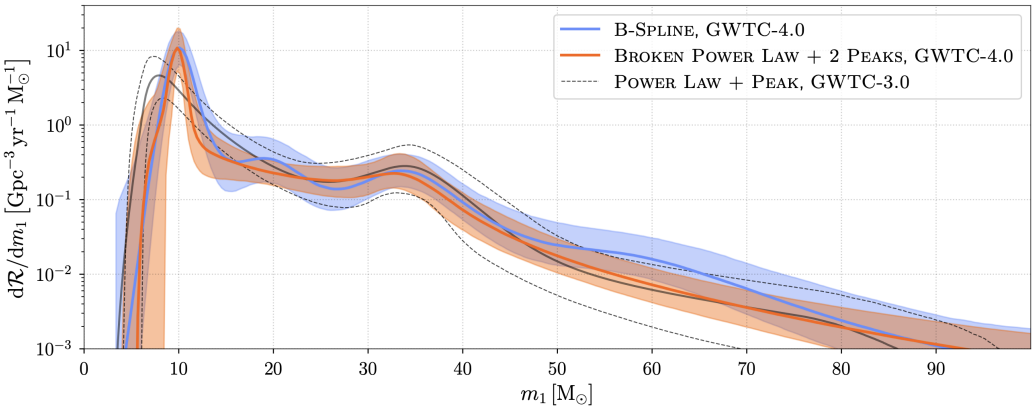

Figure 1.5: Image taken from [4] showing the BBH merger rate with 90% credible interval as a function of the primary source mass. Three parametric models are considered: the BROKEN POWERLAW + 2 PEAKS, the B-SPLINE and the POWERLAW + PEAK. The latter was the best model found in GWTC-3 population studies [16].

The source mass spectrum shows some interesting features [5]:

- the majority of black hole mergers are measured for small primary masses and the merger rate drops as the primary mass increases;
- the presence of BBH in the " forbidden" $50-100M_\odot$ *pair-instability gap* which is a stellar mass range where electron and positron pair production triggers the complete disruption of the star, preventing the formation of BHs [18]. A possible explanation to the population in the gap are the so-called *dynamical mergers*, which are formed in dense stellar environments and possibly include also *hierarchical mergers*. The latter are consecutive mergers of BHs formed in previous coalescences (first generation BHs). These second (or higher) generation BHs can reach masses above the pair instability gap threshold, thus accounting for the observations;
- the abundance of BBH mergers with primary mass $\sim 35M_\odot$, represented by a Gaussian peak. The peak can be explained by the pulsational pair instability supernovae (PISN) [19], which are very massive stars that, without these pulsations, would have retained enough mass to fall into the pair instability gap. Instead they undergo mass ejections before collapsing, resulting in an accumulation of BHs around $\sim 35M_\odot$. We remark that the association of the $35M_\odot$ peak to the PISN is still speculative and under debate.

1.4 Systematics on the Hubble constant estimation

A recent study (see [20]) has shown that systematic biases in the H_0 estimation arise when astrophysically synthetic BBH populations are modeled following phenomenological models, thus without considering the possible redshift evolution. In the study, GW signals are simulated from a fiducial astrophysical synthesis catalogs and the Hubble constant H_0 is estimated with hierarchical Bayesian inference using the spectral siren method, assuming three different parametric models for the BBH masses: the BROKEN POWER LAW, the POWERLAW + PEAK and the MULTI PEAK.

The first analysis they conducted was to consider BBH populations from the the POWERLAW + PEAK model, but including a linear evolution in redshift of the Gaussian peak with slope $\mu_g^1 - \mu_g^0$, where μ_g^0 and μ_g^1 represent the position of the Gaussian peak respectively at z=0 and at z=1. Seven different values of μ_g^1 were considered and H_0 was inferred using hierarchical Bayesian inference for all 7 values. Inference results can be seen in Fig. 1.6 and they show that the redshift evolution of the peak can strongly bias H_0, even if the redshift evolution is very light.

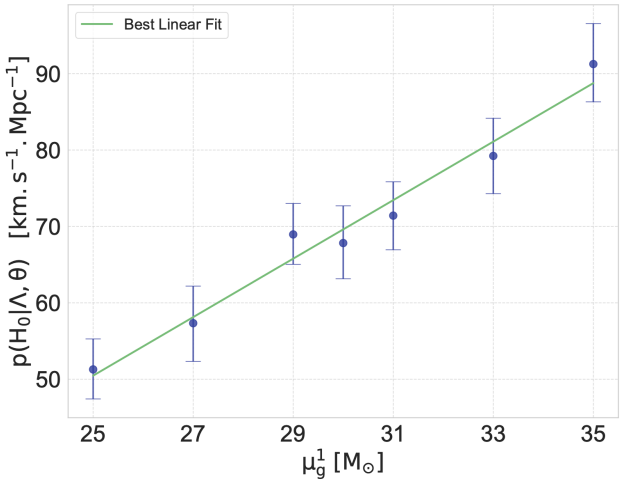

Figure 1.6: Image taken from [20] showing the different H_0 estimates with respect to the Gaussian peak position at μ_g^1 at z=1: as the peak position moves towards higher masses, the bias on H_0 increases.

Fig. 1.7 shows the evolution in redshift of the synthetic catalog that was employed in the study. The overall shape of the distribution changes as the redshift increases and some peaks appear and shift, thus systematic effects are expected since the parametric models do not take into account that redshift evolution and do not model the more fine structures of the distribution.

Simulating $\sim$ 2000 GW detections from the synthetic astrophysical fiducial catalog, the study showed that the inferred H_0 value can be biased up to 3σ, as Fig. 1.8 shows.

The study concluded that the bias is caused by the oversimplified phenomenological models that do not take into account:

- The redshift evolution in the mass spectrum, which is expected because the BBH populations are likely to evolve over cosmic time [7].
- Sharp features in the mass spectrum.

This causes a mismatch between the real BBH astrophysical population and the chosen parametric model. The study suggests that future GW cosmology should make use of population models that are able to capture those features, otherwise large GW samples will cause systematically distorted measurement of H_0. Up to GWTC-4.0, no significant redshift evolution is measured in the observed population because a better sensitivity in GW detectors will be required [5].

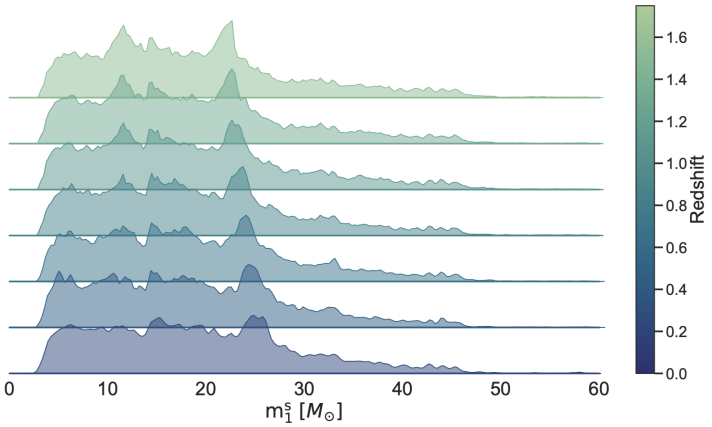

Figure 1.7: Image taken from [20] showing the evolution in redshift of the primary BBH source mass, from a synthetic astrophysical fiducial catalog.

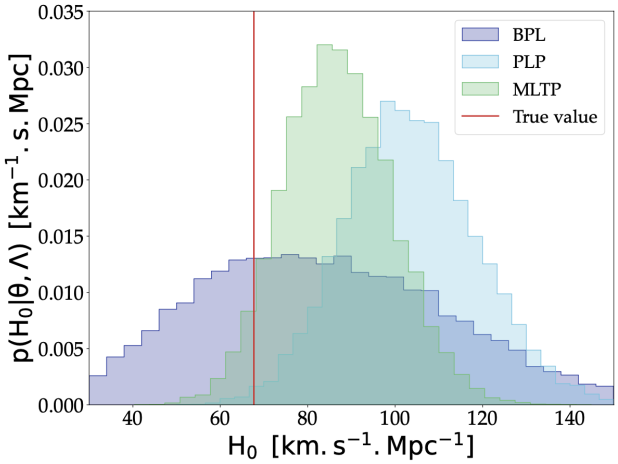

Figure 1.8: Image taken from [20] showing the biased posterior distribution for H_0 that are recovered when the synthetic catalog is modeled with the BROKEN POWER LAW (in purple), the POWERLAW + PEAK (in light blue) and the MULTI PEAK (in green).

In the study that we have just discussed, an astrophysical synthetic BBH catalog was used. Indeed, synthetic BBH catalogs are simulated by astrophysicists in order to recover the BBHs formation channels [18], this is why they are fiducial catalogs. Every simulation is achieved assuming that different astrophysical processes give rise to two main formation channels: the *isolated binary evolution* (static) and the *dynamical* channels. To avoid systematic biases when modeling those simulated population with phenomenological models, our aim is to implement machine learning tools capable to fit the mass and redshift relations predicted by those synthetic codes and see if the bias is corrected.

There is an extensive literature about the formation channels, here we just provide an overview on how these formation channels are used when parametric mass models. The first one is regulated by binary stellar physics, while the second one is governed by few body gravitational interactions:

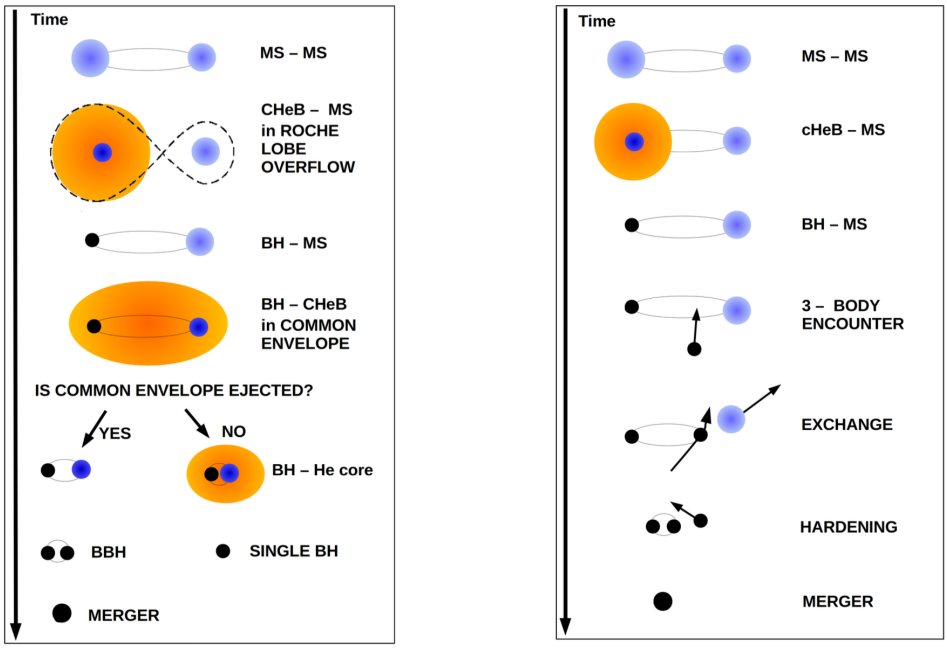

Figure 1.9: Image taken from [18] showing the evolution of the two main BBH formation chennels: on the left the isolated binary formation channel, on the right the dynamical formation channel.

- Isolated binary channel evolution: BBHs form from the evolution of massive isolated stellar binaries and it is expected to dominate in low density environments. The evolutionary path is depicted in the left panel of Fig. 1.9 and it is explained as follows:

  1. the process starts with two massive stars on their main sequence (MS);
  2. as one of the two stars evolves off the MS, it enters into its giant phase by expanding and filling its Roche lobe, which is the region within the

orbiting material is gravitationally bounded to the star. When the stellar radius exceeds the Roche lobe radius, mass transfer to the other star occurs (Roche lobe overflow);

3. the giant collapses into a BH;
4. when the other MS star enters its giant phase, the common envelope (CE) phase starts in which the BH and the helium core of the other star are both in the same envelope of gas;
5. when the CE is ejected, the loss of orbital energy and angular momentum leads to a tighter binary, evolving in two closer BHs. If the CE is not ejected, the binary merges creating a single BH;
6. a BBH is formed and will merge producing a GW emission;

- Dynamical channel evoluiton: BBHs form through a three body hierarchical capture and this is the reason why they dominate in dense stellar envirnonments. The dynamical channel evolution path, represented in the left panel of Fig. 1.9, is the following:

  1. initially, in a binary made of two stars in their MS phase, one of the two stars evolves into a BH;
  2. in dense stellar environment a three body interaction can occur when a single BH interacts with the binary;
  3. these interactions lead to a an exchange in which the single BH replaces the MS stars, forming a BBH system;
  4. following encounters with other stars or binaries further harden the BBH system, making it more tightly bound. The BBH will eventually merge and emit GWs.

  The three most relevant types of clusters are [18, 21]:

  – Globular clusters: they are old and dense systems (central density $\rho_C \geq 10^4 M_\odot pc^{-3}$), containing $\geq 10^5 M_\odot$ stars with aged $\sim 12 Gyr$. Their age implies low metallicities, which favor the formation of more massive BHs due to weaker stellar winds. They represent the $\leq 1\%$ of the baryonic mass of the Univese, this is why they are the most important dynamical formation channel.
  – Nuclear clusters: they are located at the center of galaxies where stars reach up to 10^7–$10^8\, M_\odot$, causing the presence of supermassive BHs.
  – Young clusters: they are $\lesssim 100$ Myr old and contain $\geq 10^4 M_\odot$ stars ($\rho_C > 10^3 M_\odot pc^{-3}$).

Chapter 2

Data Analysis Techniques for Population Inference

In order to extract astrophysical and cosmological information from gravitational wave observations, a set of statistical and computational tools is needed. In this Chapter I will describe the data analysis techniques implemented in this work, which combine hierarchical Bayesian inference and the machine learning technique of Normalizing Flows. Firstly, I will present the ICAROGW pipeline, which allows for hierarchical Bayesian inference, and then I will focus on the non-parametric modeling of the BBH masses and redshift distributions with Normalizing Flows.

2.1 Hierarchical Bayesian Inference with ICAROGW

The detection process of CBCs can be described as an inhomogeneous Poisson process. Its likelihood is the *hierarchical likelihood*:

$$\mathcal{L}(\{x\}|\Lambda) \propto e^{-N_{\rm exp}(\Lambda)} \prod_{i=1}^{N_{\rm obs}} \int \mathcal{L}(x_i|\theta,\Lambda) \frac{dN}{d\theta dt}(\Lambda) dt d\theta \tag{2.1}$$

$$= e^{-N_{\rm exp}(\Lambda)} \prod_{i=1}^{N_{\rm obs}} T_{\rm obs} \int \mathcal{L}(x_i|\theta,\Lambda) \frac{dN}{d\theta dt}(\Lambda) d\theta \tag{2.2}$$

where θ are the GW parameters, while Λ are the population parameters. $\mathcal{L}(x_i|\theta,\Lambda)$ is the *single-event likelihood*, in which i runs on the number of observed GW events $N_{\rm obs}$. $\frac{dN}{d\theta dt}(\Lambda)$ is the *rate*, which describes the number of CBCs as a function of the astrophysical and cosmological parameters. Finally, $N_{\rm exp}$ is the expected number of events.

The hierarchical likelihood in Eq. 2.2 can not be calculated analytically. Therefore, currents methodologies allow for numerical techniques to obtain an approximated value of it. My work is developed inside ICAROGW [1] [15], an official LVK pipeline, aimed at infering population and cosmological properties from GW observations. I will now discuss each component of Eq. 2.2 and the way they are computed.

[1] https://github.com/simone-mastrogiovanni/icarogw

- The integral of the *single-event likelihood* is computed using a set of N_s *parameter estimation samples* (PE) from GW signals, which contain information on the source parameters. Every detected GW event is affected by measurement uncertanties on the binary parameters and they are taken into account by drawing from the event posterior distribution:

$$p(\theta|x_i,\Lambda) \propto \mathcal{L}(x_i|\theta,\Lambda)\pi_{\rm PE}(\theta|\Lambda) \tag{2.3}$$

  where $\pi_{\rm PE}(\theta|\Lambda)$ is the samples prior.

  To obtain an approximation of the integral, a Monte Carlo integration sum can be performed over the *PE posterior samples*:

$$\int \mathcal{L}(x_i|\theta,\Lambda)\frac{dN}{dtd\theta}(\Lambda)d\theta = \frac{T_{\rm obs}}{N_{s,i}}\sum_{j=1}^{N_{s,i}}\frac{1}{\pi_{\rm PE}(\theta_{i,j})}\frac{dN}{dtd\theta}(\Lambda)\bigg|_{i,j} \tag{2.4}$$

- The *number of expected events* takes into account the selection bias and it is computed using a set of *injections*. Injections are simulated signals that are tested with the detector sensitivity to check which would be detectable. They correct the *selection bias* that arises from the fact that the detector has a given sensitivity, which implies that only some CBC signals are detected with a detection probability given by:

$$p_{\rm det}(\theta,\Lambda) = \int_{x\in\text{detectable}} \mathcal{L}(x_i|\theta,\Lambda)\ dx \tag{2.5}$$

  The number of expected events $N_{\rm exp}$ can be defined as:

$$N_{\rm exp}(\Lambda) = T_{\rm obs}\int p_{\rm det}(\theta,\Lambda)\frac{dN}{dtd\theta}d\theta \tag{2.6}$$

  in which $T_{\rm obs}$ is the observation time (measured in yr). The integral is approximated with a Monte Carlo integration over the injections samples:

$$N_{\rm exp} = \frac{T_{\rm obs}}{N_{\rm gen}}\sum_{j=1}^{N}\frac{1}{\pi_{\rm inj}(\theta_j)}\frac{dN}{dtd\theta}\bigg|_j \tag{2.7}$$

  In Sec A.2, we show two stability estimators for the two Monte Carlo integrations.

Finally, Eq. 2.2 can be written as follows:

$$\log\left(\mathcal{L}(\{x\}|\Lambda)\right) \sim \frac{T_{\rm obs}}{N_{\rm gen}}\sum_{j=1}^{N_{\rm det}} s_j + \sum_{i}^{N_{\rm gen}}\log\left(\frac{T_{\rm obs}}{N_s,i}\sum_{j=1}^{N_{\rm det}} w_{i,j}\right) \tag{2.8}$$

Since the logarithm[2] of the hierarchical likelihood is equal up to a constant, if that constant does not depend on the population parameters that can be inferred, the

[2]In this work, we indicate the log to be the logarithm in base e.

shape of the posterior and the location of its maximum are preserved. That's why a *scale-free* hierarchical likelihood can be defined, in which the normalization factor is neglected.
If a non informative prior distribution (Jeffreys prior [22]) for parameter space, i.e a prior free distribution is assumed for $N_{\rm eff}(\Lambda)$, then:

$$\pi\left(N_{\rm exp}(\Lambda)\right) = \frac{1}{N_{\rm exp}(\Lambda)} \tag{2.9}$$

Then the likelihood in Eq. 2.2 can be marginalized over $N_{\rm exp}$ and it takes the following form:

$$\mathcal{L}(\{x\}|\Lambda) \propto \prod_{i=1}^{N_{\rm obs}} \frac{\int \mathcal{L}(x_i|\theta,\Lambda)\frac{dN}{dtd\theta}(\Lambda)d\theta}{\int p_{\rm det}(\theta,\Lambda)\frac{dN}{dtd\theta}d\theta} \tag{2.10}$$

2.2 Normalizing Flows to replace the Parametric Model

The central idea of this work is to replace the complex astrophysical distributions, namely the CBCs merger rate, with a Normalizing Flow, which is trained using BBH masses and redshift samples from BBH synthesis catalogs.
Eq. 1.25 is computed in the following way:

$$\frac{dN}{dd_L dm_{1d} dm_{2d} dt_d} = p_{\rm NF}(m_{1s}, m_{2s}, z)\frac{dV_c}{dz}\frac{1}{1+z}\frac{1}{|J_{d\to s}|}\frac{N_{\rm astro}}{T_{\rm obs}} \tag{2.11}$$

where $p_{\rm NF}(m_{1s}, m_{2s}, z)$ is a probability assigned to the values m_{1s}, m_{2s}, z obtained by training on samples coming from mock or astrophysical catalogs and $N_{\rm astro}$ is the number of astrophysical events.
In the case in which the mock catalogs contain two formation channels, we can also define a composite rate by defining a mixture parameter ϕ [23]:

$$\begin{aligned}\frac{dN}{dd_L\, dm_{1d}\, dm_{2d}\, dt_d} = &\Big[\ \phi\ p_{\rm NF}^{\rm POP1}(m_{1s}, m_{2s}, z)\ N_{\rm astro}^{\rm POP1} \\ &+ (1-\phi)\ p_{\rm NF}^{\rm POP2}(m_{1s}, m_{2s}, z)\ N_{\rm astro}^{\rm POP2}\Big] \\ &\cdot \frac{dV_c}{dz}\ \frac{1}{1+z}\ \frac{1}{|J_{d\to s}|}\frac{1}{T_{\rm obs}}\end{aligned} \tag{2.12}$$

However, we should not limit ourselves to two formation channels. Let us assume that we are given with $N_{\rm channels}$. Then, the BBH detector rate, can be written as the sum of the rate of the single channel, each of which is multiplied by the *multiplicative factor* λ_i, with $i \in [1, ..., N_{\rm channels}]$. Here, λ_i are not normalized quantities such as ϕ, indeed they are coefficients that regulate the abundance of the single channel. Thus, we can write the detector rate as:

$$\frac{dN}{dd_L dm_{1d} dm_{2d} dt_d} = \sum_{i=1}^{N_{\rm channels}} \lambda_i\ p_{\rm NF}^{i}(m_{1s}, m_{2s}, z)\ \frac{dV_c}{dz}\frac{1}{1+z}\frac{1}{|J_{d\to s}|}\frac{N_{\rm astro}^{i}}{T_{\rm obs}} \tag{2.13}$$

The astrophysical catalogs that we used in this work sampled the redshift from the distribution:

$$R_0 \ \Psi(z,\Lambda)\frac{dV_c}{dz}\frac{1}{1+z} \tag{2.14}$$

where R_0 is the CBCs merger rate per comoving volume per year. Then, by defining $R(z,\Lambda) \equiv R_0\Psi(z,\Lambda)$, the CBCs rate in the case of one channel can be written as:

$$\begin{aligned}
\frac{dN}{dd_L\, dm_{1d}\, dm_{2d}\, dt_d} &= \frac{1}{|J_{d\to s}|}\, p_{\rm NF}(m_{1s}, m_{2s}, z) \\
&\quad \cdot \iint dz\, dm_{1s}\ dm_{2s}\ \ p_{\rm pop}(m_{1s}, m_{2s} \mid \Lambda)\ R(z,\Lambda)\ \frac{1}{1+z}\ \frac{dV_c}{dz} \\
&= \frac{1}{|J_{d\to s}|}\, p_{\rm NF}(m_{1s}, m_{2s}, z) \\
&\quad \cdot \int dm_{1s}\ dm_{2s}\ \ p_{\rm pop}(m_{1s}, m_{2s} \mid \Lambda)\ \int dz\ R(z,\Lambda)\ \frac{1}{1+z}\ \frac{dV_c}{dz} \\
&= \frac{1}{|J_{d\to s}|}\, p_{\rm NF}(m_{1s}, m_{2s}, z) \\
&\quad \cdot \int dz\ R(z,\Lambda)\ \frac{1}{1+z}\ \frac{dV_c}{dz}
\end{aligned} \tag{2.15}$$

The number of astrophysical observations $N_{\rm astro}$ is obtained through:

$$N_{\rm astro} = T_{\rm obs} \int dz\ R(z,\Lambda)\frac{1}{1+z}\frac{dV_c}{dz} \tag{2.16}$$

While the expected number of detections N_{exp} follows Eq. 2.6. It will be smaller than $N_{\rm astro}$ because it takes into account the detector selection bias:

$$\begin{aligned}
N_{\rm exp} = T_{\rm obs} &\int dm_{1s}\ dm_{2s}\ \ p_{\rm det}(m_{1s}, m_{2s}, z)\ p_{\rm NF}(m_{1s}, m_{2s}, z)\ \frac{1}{1+z}\ \frac{dV_c}{dz} \\
&\cdot \int dz\ R(z,\Lambda)\ \frac{1}{1+z}\ \frac{dV_c}{dz}
\end{aligned} \tag{2.17}$$

If we extend the result in the case of multiple catalogs:

$$\begin{aligned}
\frac{dN}{dd_L dm_{1d} dm_{2d} dt_d} &= \frac{1}{|J_{d\to s}|} \sum_{i=1}^{N_{\rm channel}} \lambda_i\ \ p^i_{\rm NF}(m_{1s}, m_{2s}, z) \int dz\ R^i(z,\Lambda)\frac{1}{1+z}\frac{dV_c}{dz} \\
&= \frac{1}{|J_{d\to s}|} \sum_{i=1}^{N_{\rm channel}} \lambda_i\ \ p^i_{\rm NF}(m_{1s}, m_{2s}, z)\ \frac{N^i_{\rm astro}}{T_{\rm obs}}
\end{aligned} \tag{2.18}$$

2.3 Normalizing Flows architecture

Normalizing Flows (NF) are unsupervised generative machine learning models that can estimate the probability density function of a distribution, given samples from that distribution as input. They work applying a sequence of invertible and differentiable transformations that map a simple base distribution into the target one. In this section, I will describe the specific NF architectures that I used in this work: Masked Affine Transformations and Neural Spline Flows. Finally, I will show the way they are build to efficiently model the joint distribution of the BBH source frame masses and redshift.

2.3.1 Masked Affine Transformations and Neural Spline Flows

Let $Z \in \mathbb{R}^D$ be a random variable with known probability density function (pdf) $p_Z : \mathbb{R}^D \to \mathbb{R}$ and let $X \in \mathbb{R}^D$ be a random variable with unknown pdf $p_X : \mathbb{R}^D \to \mathbb{R}$. Z is called the latent variable, while X is the observable variable. If there exists a diffeomorphism[3] $f : \mathbb{R}^D \to \mathbb{R}^D$ with inverse f^{-1} such that

$$x = f(z) \tag{2.19}$$
$$z = f^{-1}(x) \tag{2.20}$$

then, p_X is obtained from p_Z using the change of variable formula:

$$p_X(x) = p_Z\big(f^{-1}(x)\big) \left| \det J_{f^{-1}}(x) \right| \tag{2.21}$$
$$= p_Z(z) \left| \det J_f(z) \right|^{-1} \tag{2.22}$$

where $J_f(z) = \partial f(z)/\partial z$ is the Jacobian matrix of f at z and $z = f^{-1}(x)$ [24].
There are many ways in which the function f and its inverse f^{-1} can be built. In general, f can be a potentially complex mapping, so it is not practical to parametrize it directly. Indeed, f is decomposed into a sequence of simpler transformations. I chose to work with the *Masked Autoregressive Flow* (MAF) [25] architecture, in which the D-dimensional mapping is defined coordinate by coordinate. Rather than constructing a single transformation f that acts on the entire D-dimensional vector, f is a built using scalar bijections h, each of which transforms only the component z_i. The scalar map is the following:

$$h(\cdot; \Theta) : \mathbb{R} \to \mathbb{R} \tag{2.23}$$

it is a strictly monotonic and invertible function. It takes as input a single latent variable z_i and the parameters Θ. It outputs the transformed observable variable x_i. For each dimension i, the parameters Θ_i are generated conditionally on the previous coordinates $(x_1, ..., x_{i-1})$. This is done by a neural network called the *conditioner*, because it conditions the transformation h on the previously transformed variables. Then, an autoregressive layer $T : \mathbb{R}^D \to \mathbb{R}^D$, can be written as:

$$x_i \;=\; h\left(z_i;\; \Theta_i(x_{1:i-1})\right) \qquad i = 1, \ldots, D. \tag{2.24}$$

[3] i.e. f and its inverse f^{-1} are both differentiable.

In Fig. 2.1, the autoregressive architecture of a single layer MAT is schematically shown.

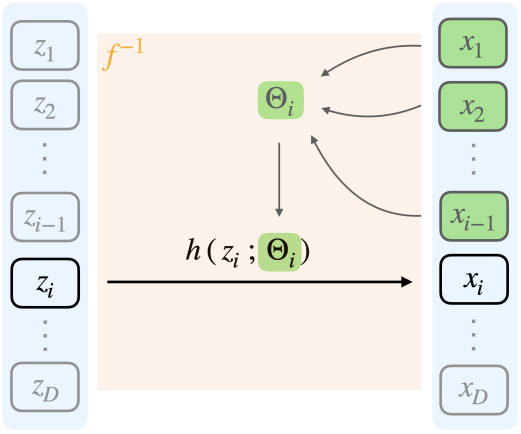

Figure 2.1: Architecture scheme of a MAT transform acting on the i^{th} component of the latent variable z_i, with $i \in [1, .., D]$. The scalar bijection h is built conditioned on the previous outputs. The parameters Θ_i of the transformation are computed with a neural network. The composition of all the scalar bijections in every dimension defines the invertible transform f.

In this work the observable variables are the BBH source frame masses and the merger redshift. Thus, for $D = 3$, the autoregressive layer has the following explicit form:

$$\begin{cases} x_1 = h\left(z_1;\ \Theta_1(\cdot)\right) \\ x_2 = h\left(z_2;\ \Theta_2(x_1)\right) \\ x_3 = h\left(z_3;\ \Theta_3(x_1, x_2)\right) \end{cases} \tag{2.25}$$

Since each output x_i depends only on z_i and on the already computed outputs $x_{<i}$, the Jacobian $J_T = \partial x/\partial z$ is lower triangular and the absolute value of its determinant factorizes as the product of the diagonal entries:

$$|\det J_T(z)| \ = \ \prod_{i=1}^{D} \left|\frac{\partial x_i}{\partial z_i}\right| = \ \prod_{i=1}^{D} \left|\partial_{z_i} h\big(z_i; \Theta_i(x_{1:i-1})\big)\right| \tag{2.26}$$

This makes the logarithm of the determinant very efficient to evaluate. The final transformation f is obtained as the composition of multiple such autoregressive layers T.

The scalar bijection h must be easy to invert and differentiate. This ensures that the map between the latent variable z and the observable variable x is efficient. Moreover, having a simple derivative allows to compute the Jacobian determinant quickly and accurately. It is sufficent therefore to provide the explicit expressions of the scalar map $h(\cdot;\Theta)$, its derivative $\partial_z h$, and its inverse h^{-1} in order to compute Eq. 2.22.
The easiest scalar bijection h that can be chosen is an affine transformation whose parameters are computed using a conditional neural network:

$$h(z_i;\ \Theta_i(x_{1:i-1})) \ = \ \mu_i(x_{1:i-1}) \ + \ \sigma_i(x_{1:i-1})\, z_i \tag{2.27}$$

with $\sigma_i(x_{1:i-1}) > 0$ in such a way that monotonicity is ensured. This choice is computationally efficient since the derivative and inverse are analytical and easy to invert and differentiate:

$$\frac{\partial h}{\partial z_i} = \sigma_i(x_{1:i-1}), \tag{2.28}$$

$$h^{-1}(x_i;\, \Theta_i) = \frac{x_i - \mu_i(x_{1:i-1})}{\sigma_i(x_{1:i-1})}. \tag{2.29}$$

This architecture works well with unimodal and simple distributions.

Another choice that can be made are *Neural Spline Flow* (NSF) [26]. In this case, the scalar bijection h is defined using *piecewise rational quadratic spline* (RQS), which works better with multi modal and complex distributions. For this architecture, the input is divided into K bins [4] and in each of them, a monotonically increasing rational quadratic function is defined and it takes the form of the ratio of two quadratic polynomials. The bin boundaries are given by $K+1$ coordinates called *knots*: $\{(u^{(k)}, v^{(k)})\}_{k=0}^{K}$ where $u^{(k)}$ is the input of the spline and $v^{(k)}$ is the output. If a single bin k is considered with domain knots $\{u^{(k)}, u^{(k+1)}\}$ and corresponding image knots $\{v^{(k)}, v^{(k+1)}\}$, then a local coordinate can be defined [5]:

$$\xi \;=\; \frac{z - u^{(k)}}{u^{(k+1)} - u^{(k)}} \in [0, 1] \tag{2.30}$$

where $\Delta u^{(k)} \equiv u^{(k+1)} - u^{(k)}$ and $\Delta v^{(k)} \equiv v^{(k+1)} - v^{(k)}$. In the range $\{(u^{(0)}, v^{(0)})\} = (-B, -B)$ and $\{(u^{(K)}, v^{(K)})\} = (B, B)$ the knots monotonically increase. The default value for the boundaries is $B = 5$, which implies that the spline is applied only in the interval $[-B, B]$. Thus, variables need to be correctly normalized in such a way that the distribution fits within that interval. Outside of the spline range, linear tails are used. In every bin, the knot derivative is defined and set positive $\{\delta^{(k)}\}_{k=1}^{K-1}$ and at the boundaries $\delta^{(0)} = \delta^{(K)} = 1$ to match the linear tails. Fig. 2.2 shows a plot of the monotonicity of the spline and of its inverse, of the spline itself and of its linear tails.

$$s^{(k)} = \frac{\Delta v^{(k)}}{\Delta u^{(k)}} \tag{2.31}$$

In each layer of the transform, the neural network conditioner estimates the input, output and derivative of the spline: $(u^{(k)}, v^{(k)}, \delta^{(k)}) = \Theta_i(x_{1:i-1})$. In each bin the RQS scalar map is written as a quotient of two quadratics in the local coordinate ξ:

$$x \;=\; h(z; \Theta) \;=\; v^{(k)} + \Delta v^{(k)}\, \frac{N(\xi)}{D(\xi)} \tag{2.32}$$

with

$$\begin{aligned} N(\xi) &= s^{(k)}\, \xi^2 + \delta^{(k)}\, \xi(1-\xi) \\ D(\xi) &= s^{(k)} + \left(\delta^{(k+1)} + \delta^{(k)} - 2s^{(k)}\right) \xi(1-\xi) \end{aligned} \tag{2.33}$$

[4] The number of bins that has been used in this work is $K = 8$.

[5] Here, the coordinate index i is omitted to lighten the notation.

It can be shown [26] that:

$$\frac{dh}{dz} = \frac{(s^{(k)})^2 \left[\delta^{(k+1)}\xi^2 + 2s^{(k)}\xi(1-\xi) + \delta^{(k)}(1-\xi)^2 \right]}{\left[s^{(k)} + (\delta^{(k+1)} + \delta^{(k)} - 2s^{(k)})\,\xi(1-\xi) \right]^2} \tag{2.34}$$

Finally, the inverse can be found inverting analitically Eq. 2.33.

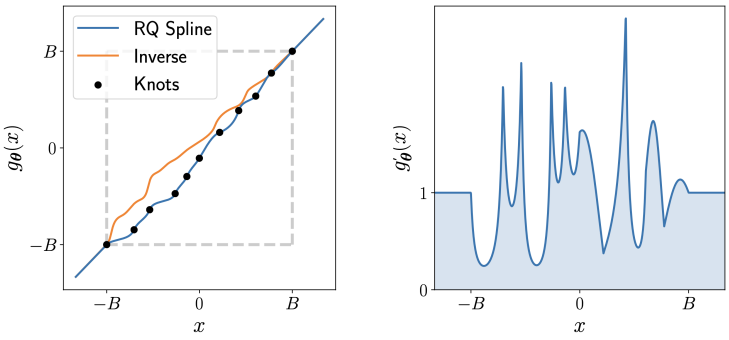

Figure 2.2: Image taken from [26] with K=10 bins showing the RQS bijection h, its inverse and its derivative. The notation is different since in the paper formalism the input domain is x and the bijection is $g_\theta(x)$. It is clear that they are continuous functions and that thanks to the knots, multi modality distributions are likely to be well modeled.

2.3.2 Training and Validation

I have employed two different Normalizing Flows models: the first one uses the easiest Masked Affine Transformation model, described in Eq. 2.27, to fit simpler unimodal distributions and it utilizes the *nflows* [27] library. The second one is a NSF and it works better for multimodal, complex distributions; it uses the *zuko* [28] library. Two `python` classes, both called `one_catalog_wrapper`, were built to be compatible with the *icarogw* pipeline. In the case of two populations, the class `two_catalogs_wrapper` was build in such a way to follow Eq. 2.12. The classes are reported in Sec. A.4.

What follows is the step by step description of the train and test of the flow and the estimation of the $p_{\rm NF}(m_1, m_2, z)$, needed to compute Eq. 2.11. For both models, the constructor takes as an input a dictionary containing the binary parameters (m_1, m_2, z) and stacks them into a matrix X. Then, for every variable, the mean μ and the standard deviation σ are computed and the normalized variables are found with $(X - \mu)/\sigma$. Data are split into train (80%) and validation (20%) sets.

The models are build with a given number of transformations and a given width of the neural network that estimates the transformation parameters. Both flows are trained by miminizing the *negative log likelihood* (NLL) on the normalized data. In each iteration during the training, a random mini batch of size B is drawn from the

training set, and the loss function is computed as:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\log p_\theta(y_i) \tag{2.35}$$

where $p_\theta(y_i)$ is the probability assigned by the flow with parameters θ to the sample y_i. Minimizing the NLL is equivalent to minimize the Kullback Leibler (KL) divergence [29]. This implies that the flow is trained in such a way to fit the data distribution as close as possible.

After the loss is computed, the *backpropagation* is performed: the gradient of the loss is computed with respect to the neural network parameters θ that define the transformation inside the flow. The gradient shows how much each parameter contribuited to the error and in which direction it should be adjusted to minimize the NLL. The optimizer that we chose for the backpropagation is *Adam* (Adaptive Moment Estimation) [30] that works with a given *learning rate* which determines the size of each update step: if it is too large, the training may diverge, if it is too small, the training becomes less efficient.

After each update, the model is evaluated on the validation set and the validation NLL is computed. A learning rate scheduler reduces the learning rate when the validation loss stalls and an early stopping stops the training if after several consecutive iterations there is no improvement; the number of consecutive iterations is called *patience*. The best validation loss parameters are kept and saved as the best model. To have a direct evaluation of the performance of the flow, we implemented two diagnostics procedures [31]:

1. The training history of the flow is visualized with a plot that shows the training and validation losses as a function of the iteration number. The plot provides a check of the behaviour of the flow during the training process. In a well performing model, the training loss is ecpected to decrease steadily and to eventually plateau, while the validation loss should follow a similar trend and stabilize around a value close to the training one. The training loss typically shows large oscillations, while the validation loss appears to be much smoother. This is a consequence of the optimization procedure. The training loss is computed at every step over a random mini batch of the training set, which provides only a noisy estimate of the full loss, then introducing significant variance. On the other hand, the validation loss is always evaluated on the entire validation dataset and this implies a stable estimate of the NLL, resulting in a much smoother curve.

2. The best value of validation loss, the best value of train loss and their absolute difference are computed. They are extracted from the best model, which is the model that corresponds to the minumum validation loss. A well trained model is expected to reach comparable values of train and validation loss, with a small absolute difference. If the validation loss stops decreasing and the training loss continues to improve, then the model is overfitting: it is "memorizing" the training set without capturing the true distribution, thus it can not be used as

a predictable model. On the other hand, if both losses remain high and do not decrease during training, the model is underfitting, meaninig that it has not learned an accurate representation of the data. Therefore, the printed values give a diagnostic of both the quality of the fit and the generalization capability of the flow.

The NF returns the estimation of the combined pdf of the three variables m_1, m_2 and z. To correctly take into account the change of variables from the normalized space to the observable space, the following change of variables correction is applied:

$$\log p_X(x) = \log p_Z\left(\frac{x-\mu}{\sigma}\right) - \sum_{j=1}^{3} \log \sigma_j \tag{2.36}$$

The *nflows* models is faster and simpler since it does not include the learning rate scheduler. An example of the two Normalizing Flows architecture follows.

In Tab. 2.1 and Tab. 2.2 two examples of the architectures of the *zuko* and *nflows* models are showed. In particular, the input and output features of every layer of the transformation are summarized. For both models, the example reports a Normalizing Flow with 3 layers and 20 hidden features, in such a way to better understand the architecture differences between them.

zuko model architecture

		input features	output features
layer1	MaskedLinear	3	20
	ReLU	20	20
	MaskedLinear	20	69
layer2	MaskedLinear	3	20
	ReLU	20	20
	MaskedLinear	20	69
base distribution: 3D diagonal Gaussian			

Table 2.1: Architecture of the *zuko* model in which every layer is made of three transformations. In the first one, a MaskedLinear neural nework [25] maps 3 input features into 20 hidden features. A nonlinear activation (ReLU) [32] follows. The final transformation outputs 69 features, which correspond to the parameters of the RQS. For each variable, 8 bin widths, 8 bin heights and 7 derivatives at the internal knots are computed (the derivatives at the bounary knots are fixed, as discussed in Sec. 2.3). The 23 parameters for all the 3 variables are equal to a total of 69 parameters, that the neural network computes.

nflows model architecture

		input features	output features
layer1	MaskedLinear	3	20
	ReLU	20	20
	MaskedLinear	20	6
layer2	MaskedLinear	3	20
	ReLU	20	20
	MaskedLinear	20	6
base distribution: 3D diagonal Gaussian			

Table 2.2: Architecture of the *nflows* model in which every layer is made of three transformations, just as in the *zuko* model. In this model, the final transformation output features are equal to 6. For each of the 3 input variables, the 2 affine transformation parameters, described in Eq. 2.27, are computed by the neural network, thus having a total of 6 parameters to be computed.

Chapter 3

Validation of the Methodology

Before applying the developed framework to synthetic BBH distributions, it is crucial to validate the model to the parametric BBH distributions that have been shown in literature to be able to introduce a systematic bias on the estimation of H_0. In this Chapter, I introduce a simulation pipeline for benchmarking gravitational wave population analysis. We then use this pipeline to study if Normalizing Flows can approximate populations models from redshift evolving distributions.

3.1 Redshift evolving population

The POWERLAW+ EVOLVING PEAK mock catalog was generated by constructing a `python` class compatible with the ICAROGW pipeline. The combined probability density function for the primary masses and redshift distribution is:

$$p_{\rm pop}(m_1, m_2, z) = p(m_1 \mid z)\; p(m_2 \mid m_1)\; p_z(z) \tag{3.1}$$

Firstly the primary BBH source mass m_1 at a given redshift was assumed to be the mixture of a smoothed power law and a truncated gaussian:

$$p(m_1|z, \Lambda) = \phi\; p_{\rm PL}^{\rm smooth}(m_1|\Lambda) + (1-\phi)\; p_{\rm G}(m_1|z, \Lambda) \tag{3.2}$$

where the POWERLAW term follows [15]:

$$p_{\rm PL}^{\rm smooth}(m_1|\Lambda) = p_{\rm PL}(m_1|\Lambda)\frac{W(m_1)}{\mathcal{N}_{\rm smooth}} \tag{3.3}$$

in which

- $p_{\rm PL}(m_1|\Lambda)$ is the pdf of a stationary power law between $[m_{\rm min}, m_{\rm max}]$:

$$p_{\rm PL}(m_1|\Lambda) = \frac{m_1^{-\alpha}}{\mathcal{N}_{\rm PL}(\alpha)} \tag{3.4}$$

  here, $\mathcal{N}_{\rm PL}$ is the power law normalization constant:

$$\mathcal{N}_{\rm PL}(\alpha) = \frac{m_{\rm max}^{1-\alpha} - m_{\rm min}^{1-\alpha}}{1-\alpha} \qquad \text{if } \alpha \neq 1 \tag{3.5}$$

$$\mathcal{N}_{\rm PL}(\alpha) = \log\left(\frac{m_{\rm max}}{m_{\rm min}}\right) \qquad \text{if } \alpha = 1 \tag{3.6}$$

- $W(m_1)$ is a piecewise window function that transitions from 0 to 1, ensuring a smooth low masses cutoff:

$$W(m) = \begin{cases} 0 & m \leq m_{\min}, \\ \left[1 + \exp\left(\frac{\delta m}{m - m_{\min}} + \frac{\delta m}{m - m_{\min} - \delta m}\right)\right]^{-1} & m_{\min} < m < m_{\min} + \delta m, \\ 1 & m \geq m_{\min} + \delta m. \end{cases} \tag{3.7}$$

- $\mathcal{N}_{\rm smooth}$ takes into account the renormalization of the pdf after applying the smoothing window:

$$\mathcal{N}_{\rm smooth} = \int_{m_{1\min}}^{\infty} p_{\rm PL}(m_1|\Lambda)\ W(m_1)\ dm_1 \tag{3.8}$$

The EVOLVING PEAK term is a truncated Gaussian with lower truncation at $m_{\min}$ and upper truncation at ∞; it follows [33]:

$$p_{\rm G}(m_1|z,\Lambda) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\big(m_1 - \mu(z)\big)^2}{\sigma^2}\right] \frac{1}{1 - cdf\left(\frac{m_{\min} - \mu(z)}{\sigma}\right)} \tag{3.9}$$

here:

- $\mu(z)$ is the mean value of the Gaussian that evolves linearly with redshift z:

$$\mu(z) = z\mu_1 + \mu_0 \tag{3.10}$$

- $cdf(t)$ is the truncated gaussian cumulative density function:

$$cdf(t) = \int_{-\infty}^{t} \frac{\exp\big(-t'^2/2\big)}{\sqrt{2\pi}} dt' \tag{3.11}$$

The secondary BBH source mass m_2 is a truncated power law on $[m_{\min}, m_1]$ with a low mass smoothing:

$$p(m_2 \mid m_1) = \begin{cases} \dfrac{m_2^{-\beta}}{\mathcal{N}_{\rm PL}(\beta)} & m_{\min} \leq m_2 \leq m_1 \quad \text{(outside smoothing region)} \\ \dfrac{m_2^{-\beta}\, H(m_2)}{\mathcal{N}_{\rm PL}(\beta)} & m_{\min} \leq m_2 < m_{\min} + \delta m \quad \text{(inside smoothing)} \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

where:

- $\mathcal{N}_{\rm PL}$ follows Eq. 3.6 with $m_{\max} = m_1$
- $H(m)$ is a smoothing Hann function [34]:

$$H(m) = \frac{1}{2}\big[1 - \cos(\pi s)\big] \qquad s \equiv \frac{m - m_{\min}}{\delta m} \in [0, 1) \tag{3.13}$$

3.2 Description of the synthetic redshift evolving catalog

A POWERLAW+EVOLVING PEAK catalog was simulated with the population parameters described in Tab. 3.1. The redshift distribution follows the Madau Dickinson [35] star formation rate:

$$\Psi\left(z|\gamma,k\right) = \left[1+(1+z_{\rm P})^{-\gamma-k}\right]\frac{(1+z)^{\gamma}}{1+\left(\frac{1+z}{1+z_{\rm P}}\right)^{\gamma+k}} \tag{3.14}$$

This functional form captures the characteristic behavior of the cosmic star formation history with a growth phase modeled by the slope parameter γ, followed by a maximum indicated by $z_{\rm P}$ and a subsequent decline parametrized by the slope parameter k.

Parameter	Value	Description
α	4.09	Exponent of the primary mass POWERLAW
β	-1.23	Exponent of the secondary mass POWERLAW
μ_0	32.27 $M_\odot$	PEAK mean at $z = 0$
μ_1	2.5 $M_\odot$	Linear redshift evolution coefficient of the PEAK mean
σ	3.88	Standard deviation of the PEAK component
ϕ	0.04	Mixture fraction between PEAK and POWERLAW component
$m_{\rm max}$	112.50 $M_\odot$	Maximum of the BBH mass
$m_{\rm min}$	4.98 $M_\odot$	Minimum of the BBH mass
δ_m	3.70 $M_\odot$	Smoothing parameter for the low masses cutoff
γ	2.70	
k	6	Madau Dickinson star formation rate parameters [35]
$z_{\rm P}$	0.50	

Table 3.1: Population parameters for the POWERLAW+EVOLVING PEAK mock catalog.

Fig. 3.1 and Fig. 3.2 show the mock redshift evolving catalog in the source frame and in the detector frame. The source frame plot shows the true astrophysical distribution of BBH source masses and merger redshift. The linear evolution of the Gaussian peak stands out significantly, in particular in the bottom left plot of Fig. 3.1. The detector frame population of BBH source masses and merger redshift is obtained taking into account the selection effects. The redshift evolution of the peak is more visible in the astrophysical corner plot (Fig. 3.1) because of the selection biases that let us detect more massive black holes, in the redshift evolving Gaussian peak, to higher distances.

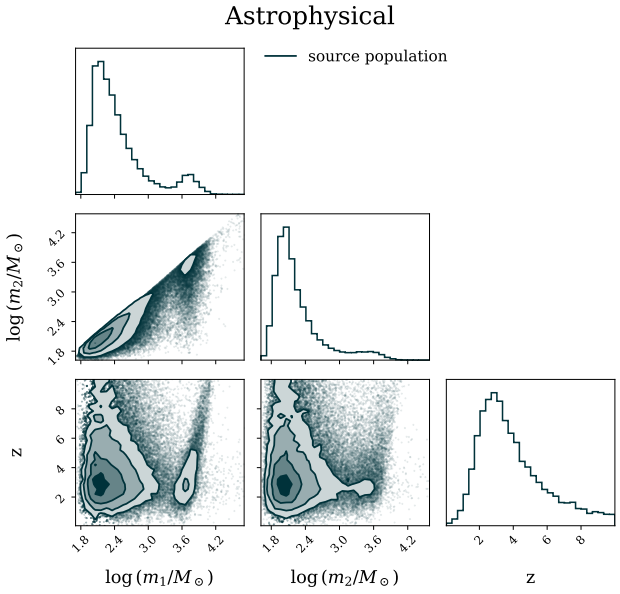

Figure 3.1: Astrophysical POWERLAW+EVOLVING PEAK population. The peak evolves linearly following Eq. 3.10 with slope $\mu_1 = 2.5\ M_\odot$. The logarithm of the masses is plotted instead of the masses themselves to better visualize the distribution.

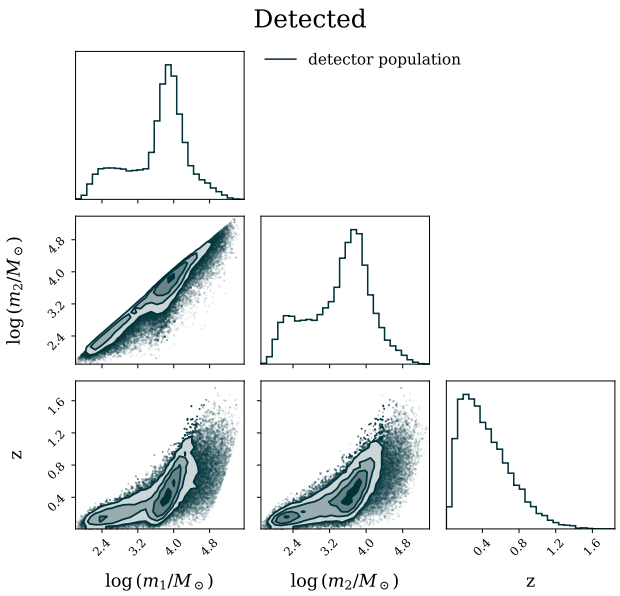

Figure 3.2: Detected POWERLAW+EVOLVING PEAK population, with selection effects taken into account. Detected masses are redshifted as expected, since they follow Eq 1.17.

Another visualization of the redshift dependence can be seen by plotting the primary source mass m_1 distribution at different redshift ranges. In Fig. 3.3, the linear evolution of the Gaussian peak is noticeable. The plot on the left shows the evolution of the spectrum up to $z \sim 2.37$ to better enhance the linear trend of Eq. 3.10, even if mergers at those redshifts are not measured because of the ground based detectors limitations. The right plot shows the evolution of the detected masses, the evolution stops at $z \sim 1.2$ because it is a value near the most distant detected source. The detected primary masses are redshifted, as expected from Eq. 1.17.

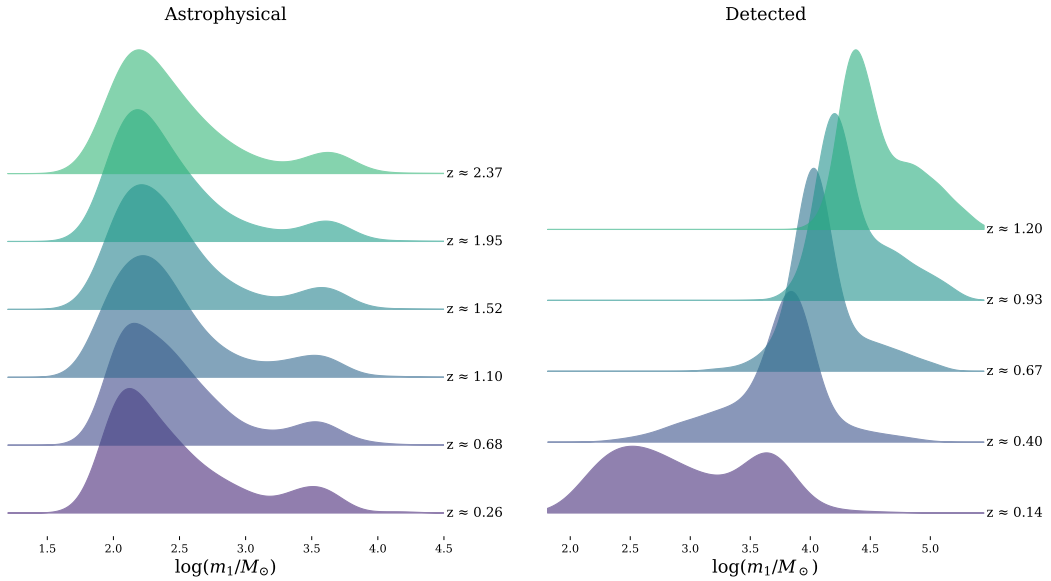

Figure 3.3: Source and detector plot of the distribution of the logarithm of the primary source mass m_1 at different redshift ranges. The population samples were fitted with a Gaussian Kernel Density Estimator (KDE) [36] to obtain a smooth distribution.

We generated the PE posterior samples from a POWERLAW + EVOLVING PEAK, fixing the population parameters following Tab. 3.1 and the cosmological quantities to $H_0 = 67.7$ km s^{-1}Mpc^{-1}, $\Omega_m = 0.308$. We used the GWTC-3 injection samples [1] and finally we modeled the masses distribution with a POWERLAW+PEAK, thus ignoring the redshift evolution. We inferred [2] H_0 using the Markov chain Monte Carlo (MCMC) [38] technique, fixing the matter density parameter at $\Omega_m = 0.308$ [39].

The H_0 posterior distribution is showed in Fig. 3.4 and it recovers the results obtained in [20] and discussed in Sec. 1.4: H_0 is biased up to 3σ. The analytical form of the POWERLAW+PEAK phenomenological model, which does not evolve with redshift, is the cause of the bias. This implies that whenever a redshift evolving population is

[1] More information about GWTC-3 injections samples and PE posterior samples is found in Sec. A.3.

[2] The BILBY [37] library and the ICAROGW [15] likelihood were used to perform inference.

modeled with a population that is fixed in redshift, a systematic error comes out as a result of the mismatch between the redshift evolving population and the wrong approximation of the phenomenological model.

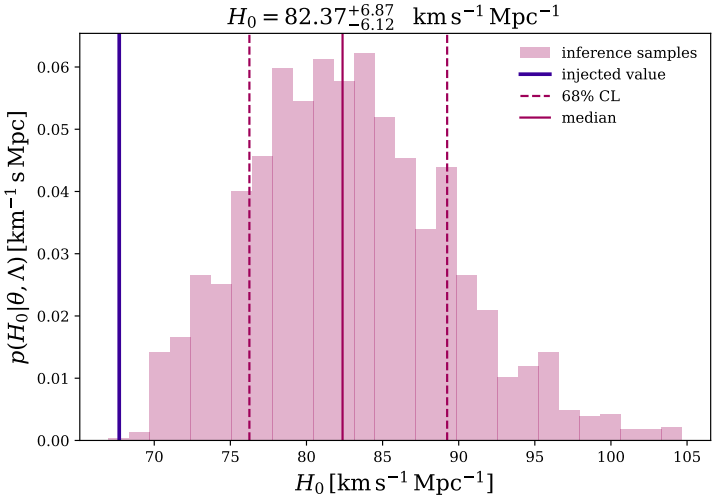

Figure 3.4: Samples from the posterior probability distribution of H_0, obtained with MCMC inference, performed with 600 steps and 4 walkers. The median value of the posterior is compatible within the true value within 3σ of error, showing a bias. This is compatible with results obtained in [20].

3.3 Redshift evolving population modeled with Normalizing Flows

The purpose of this section is to show that by substituting the phenomenological model with the Normalizing Flows, the bias on the H_0 estimation is fully corrected. The Normalizing Flows fits in $\log(m_1)$, $\log(m_2)$, z, by accounting for the Jacobian J correction:

$$p_{\rm NF}(m_1, m_2, z) = p_{\rm NF}\Big(\log(m_1), \log(m_2), z\Big)\frac{1}{|J|} \tag{3.15}$$

$$= p_{\rm NF}\Big(\log(m_1), \log(m_2), z\Big)\frac{1}{m_1 m_2} \tag{3.16}$$

We use two models for the BBH merger rate with the Normalizing Flows.

- ONE CATALOG: we draw samples from the POWERLAW + EVOLVING PEAK population as a single catalog and we fit it with a single Normalizing Flow. We use the *zuko* model and finally we perform inference on H_0 to verify the bias correction. The *zuko* model performs well in modeling and fitting multimodal and complex distributions, as discussed in Sec. 2.3.2.

- TWO CATALOGS: we draw samples separately from the POWERLAW and from the EVOLVING PEAK and we fit them using two different *nflows* models. The

two populations are mixed using the mixture fraction coefficient ϕ, following Eq. 1.27. We chose the *nflows* model since it fits very well unimodal and simple distributions, such as the POWERLAW and the EVOLVING PEAK, as described in Sec 2.3.2.

This last methodology is useful because it allows us to infer the mixture fraction between the formation channels. This implies that there is the possibility to constraint two different populations coming for example from two different formation channels, using GW signals and thus having astrophysical information.

3.3.1 Calibration of ONE CATALOG architecture and H_0 inference

The first step in the full ONE CATALOG analysis was to let the Normalizing Flows fit the source masses and redshift distribution in order to estimate $p_{\rm NF}(m_1, m_2, z)$ directly from data, without the need of an analytical phenomenological model.

The architecture that was used is the *zuko* `one_catalog_wrapper` (described in Sec. 2.3.2), with the following parameters:

number of layers	2
hidden features	40
learning rate	0.002
batch size	1000
number of iterations	1000
patience	40

The diagnostics of the training are reported in Fig. 3.5. The model performance varies depending on the Normalizing Flows model, especially the number of layers, the hidden features and the learning rate. A compromise between the best (but most of the times slower) and the fastest (but most of the times less accurate) model was chosen. For example, we preferred a lighter Normalizing Flows whose best validation loss was only 2% worse with respect to the heavier, but more precise Normalizing Flows. This choice was made to maintain the precision of the fit but ensuring a fast fit to improve the MCMC inference time. The parameters choice we made brought to a hierarchical likelihood evaluation time of $\sim 0.5s$.

best training loss (BTL)	2.4260
best validation loss (BVL)	2.4270
|BTL - BVL|	0.0190

Figure 3.5: Normalizing Flows diagnostics show that the model has learned the distribution without overfitting or underfitting, since both the training loss and validation loss curves decrease as expected and the difference between the best model training loss and validation loss is of the order of 10^{-2}.

The Normalizing Flows fit of the population can be seen in Fig. 3.6 and it was obtained with rejection sampling using the pdf estimated by the Normalizing Flows itself. It correctly captures the redshift evolution in the POWERLAW+EVOLVING PEAK distribution. Thus, to check if the bias on H_0 is corrected, we performed MCMC inference with the PE posterior samples simulated from the POWERLAW+EVOLVING PEAK and the GWTC-3 injection set. The results are visible in Fig. 3.7: the bias has been removed in the estimation of H_0.
This analysis shows the goodness of the Normalizing flow fit: it is perfectly able to reconstruct the distribution of a population starting directly from data. It also shows that the Normalizing Flow model is correctly implemented into the ICAROGW pipeline and ready to be tested on real astrophysically simulated fiducial catalogs.

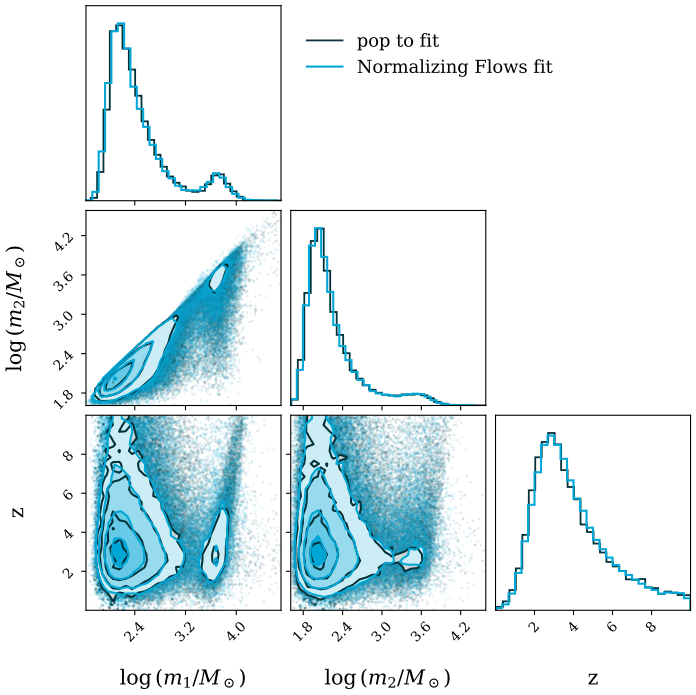

Figure 3.6: Normalizing Flow fit (in light blue) of the POWERLAW+EVOLVING PEAK population (in black). The model captures the multi modality of the distribution, in particular the evolving Gaussian peak. One could notice that the tails of the distribution are very well fitted.

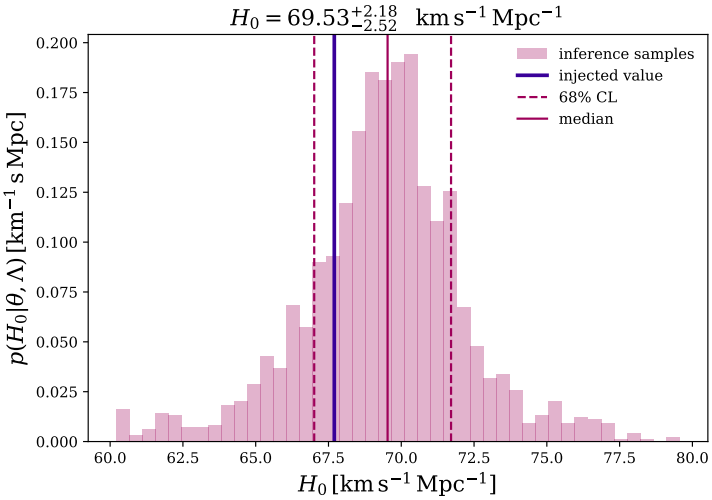

Figure 3.7: H_0 posterior distribution obtained with MCMC inference with 600 steps and 4 walkers. The median of the distribution is compatible within 1σ of error with the injected value. The systematic bias has been corrected.

3.3.2 Calibration of TWO CATALOGS architecture and H_0 inference

Before moving to astrophysically simulated fiducial catalogs, we developed a further methodology to test another potentiality of the Normalizing Flows analysis. The purpose is to verify if two different populations (that could represent two different formation channels) can be fitted separately with a Normalizing Flows and then mixed with a mixture fraction coefficient ϕ, following Eq. 2.12. The two different populations are the POWERLAW and the EVOLVING PEAK, they are singularly fitted using the *nflows* model `one_catalog_wrapper` (described in Sec. 2.3.2) and they are put together in the `two_catalogs_wrapper`, which estimates the following probability:

$$p_{\rm NF}(m_1, m_2, z) = \phi \; p_{\rm NF}^{\rm PL}(m_1, m_2, z) + (1-\phi)\, p_{\rm NF}^{\rm Peak}(m_1, m_2, z) \tag{3.17}$$

The POWERLAW and EVOLVING PEAK Normalizing Flows fits are reported in Fig. 3.8 and Fig. 3.9 and their parameters are reported as follows.
For the POWERLAW population:

number of layers	2
hidden features	32
learning rate	0.001
batch size	1000
number of iterations	1000
patience	40

for the EVOLVING PEAK population:

number of layers	3
hidden features	32
learning rate	0.001
batch size	1000
number of iterations	1000
patience	40

In this case as well, we chose the best performing light model over the heaviest but most precise one, since the loss in performance was minimal while the computational time was significantly higher for the heavier model. The hierarchical likelihood evaluation time took $\sim 0.5s$. The diagnostics follow the expected behavior (described in Sec. 2.3.2) and will be omitted from now on to lighten the discussion.

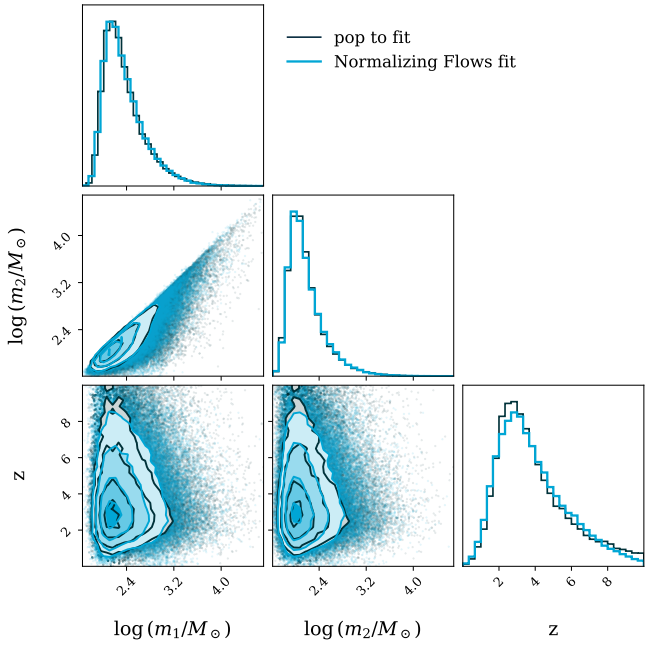

Figure 3.8: POWERLAW Normalizing Flows fit (in light blue) of the underlying injected population (in black). The tails of the distribution are well fitted and the whole distribution is well approximated by the Normalizing Flows.

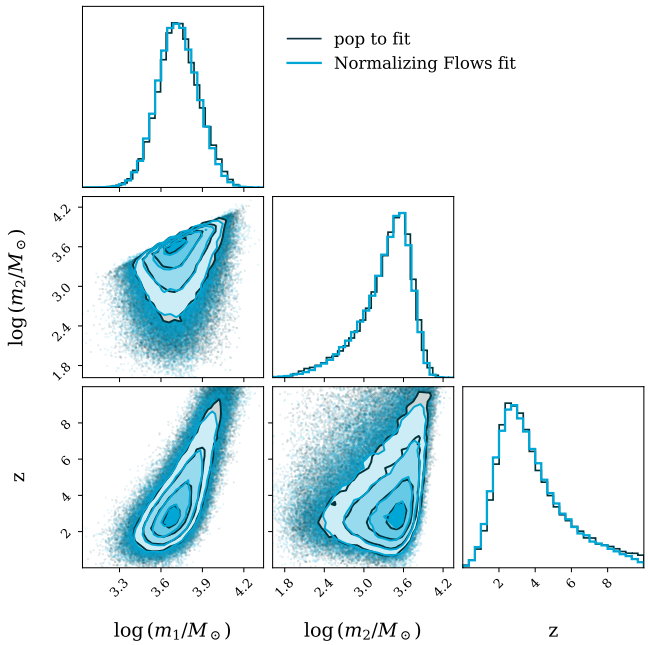

Figure 3.9: PEAK Normalizing Flows fit (in light blue) of the redshift evolving population (in black), which is perfectly captured by the model.

In order to perform hierarchical inference and verify the H_0 bias correction, an estimation of the mixture fraction ϕ in Eq. 3.17, is needed. We inferred its value and we fixed it at $\phi = 0.07$.
Finally, the MCMC inference on H_0 has been performed and the results are shown in Fig. 3.10. Once again, the systematic bias on H_0 has been fully corrected thanks to the Normalizing Flows that was able to accurately fit the redshift evolving population.

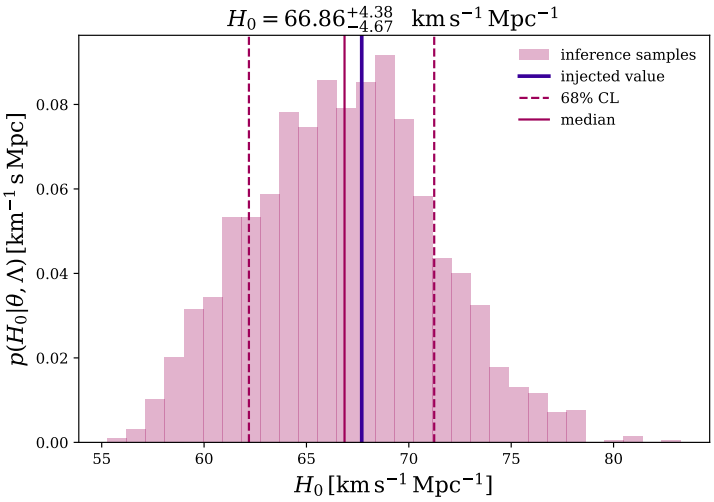

Figure 3.10: Results of the MCMC inference on H_0 performed with 4 walkers and 600 steps: the bias has disappeared and the posterior median of H_0 is compatible with the true value.

This analysis shows that the Normalizing Flows model is able to capture redshift evolving population and can correctly manage different population probabilities into the ICAROGW pipeline. Thus, the last step is to test the model on an astrophysically simulated fiducial catalogs to infer population and cosmological quantities.

Chapter 4

Cosmology with Synthetic Gravitatioanal Wave Populations

The purpose of this Chapter is to apply the methodology described in the previous Chapters to an astrophysically simulated fiducial catalog, to infer population and cosmological properties. Starting from a synthetic catalog made by four sub catalogs, I generated PE posterior samples and injection samples. Then I chose two different approaches: I firstly conducted an analysis on the full catalog and then I focused on two catalogs, which represent the two main BBH formation channel. By studying the catalogs separately, not only we can infer cosmological quantities, but we can also put constraints in the absolute abundance of each channel.

4.1 Description of a B-POP catalog

The catalog we chose was simulated by the researchers at the Gran Sasso Science Institute (GSSI) using the population synthesis tool B-POP [40]. The full catalog (called *global*) is made of four catalogs representing the four main formation channels discussed in Sec. 1.4. The first catalog represents the *isolated* binaries channel, while the *globular clusters* (GCs), *nuclear clusters* (NCs) and *young clusters* (YCs) catalogs contain the dynamical formation channels. For each channel, the following quantities, that correspond to 1 year, are provided:

- BBH source masses m_1, m_2
- merger redshift z
- merger rate $R(z)$ measured in yr^{-1} Gpc^{-3} and defined in Sec. 2.2. To obtain the total merger rate the single catalog merger rates need to be summed up.

To better understand the B-POP catalog, we report in Fig. 4.1 the plot of the source and detector primary and secondary masses, and of the source and detected redshift and luminosity distance. The quantities at the detector were simulated by taking into account the selection effects caused by the detector sensitivity. The B-POP distribution is non trivial and mergers are detected up to $z \sim 1.6$.

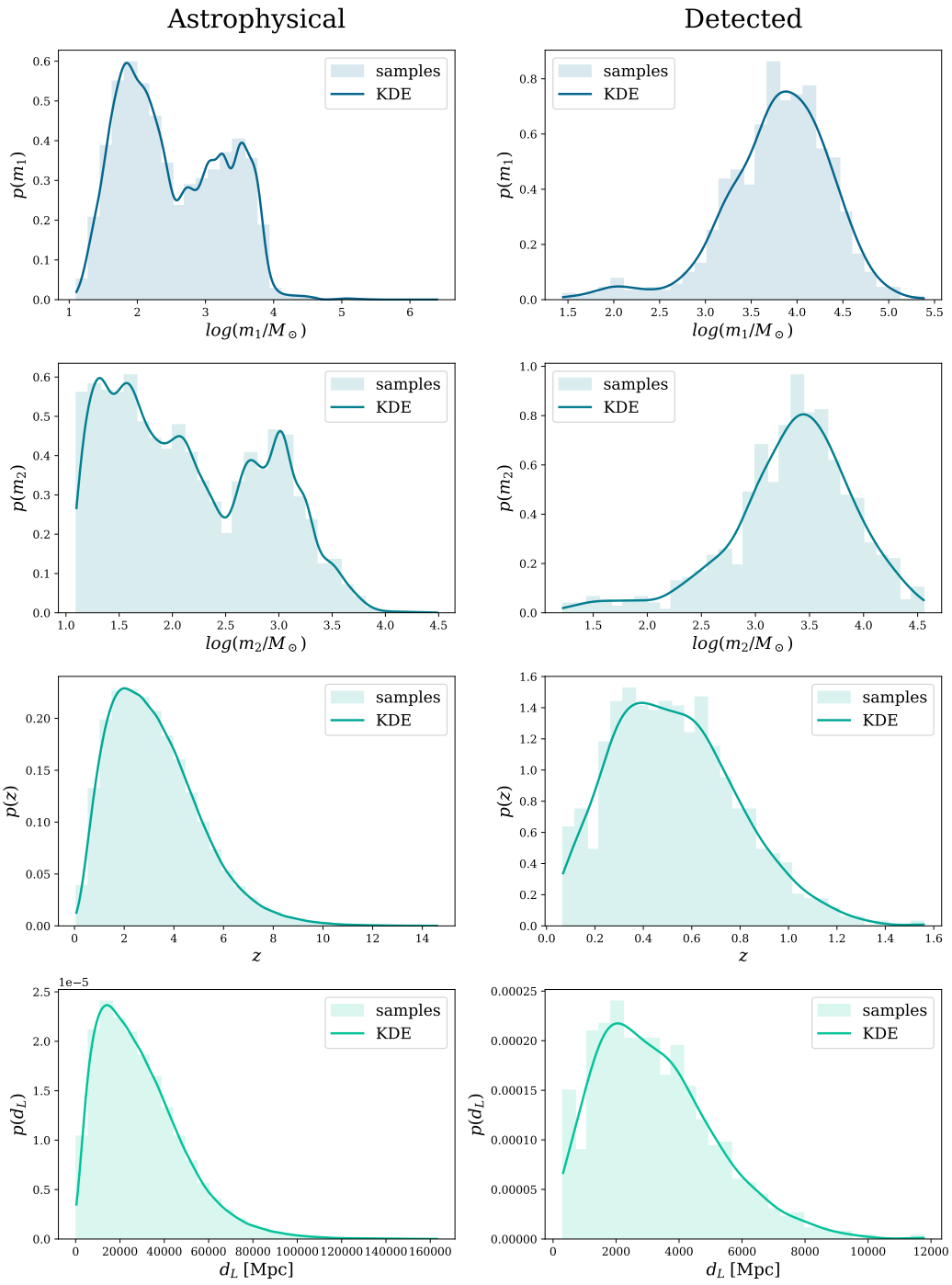

Figure 4.1: B-POP source and detector distributions for the BBH masses, redshift and luminosity distance. Because of selection effects, in O5 like era, the BBH lower masses will be less frequent to detect. Also, only mergers up to $z \sim 1.6$ are detected. This implies that in the case of this methodology, only the Normalizing Flows fit at low redshifts will be used during inference.

Fig. 4.2 shows the distributions of the BBH primary source and detector masses. The plot clearly demonstrates that, in astrophysical simulations, the masses distribution is expected to evolve in redshift over cosmic time. The evolution is non trivial and it exhibits redshift evolution as well as more complex and fine structures that the Normalizing Flows will be required to capture.

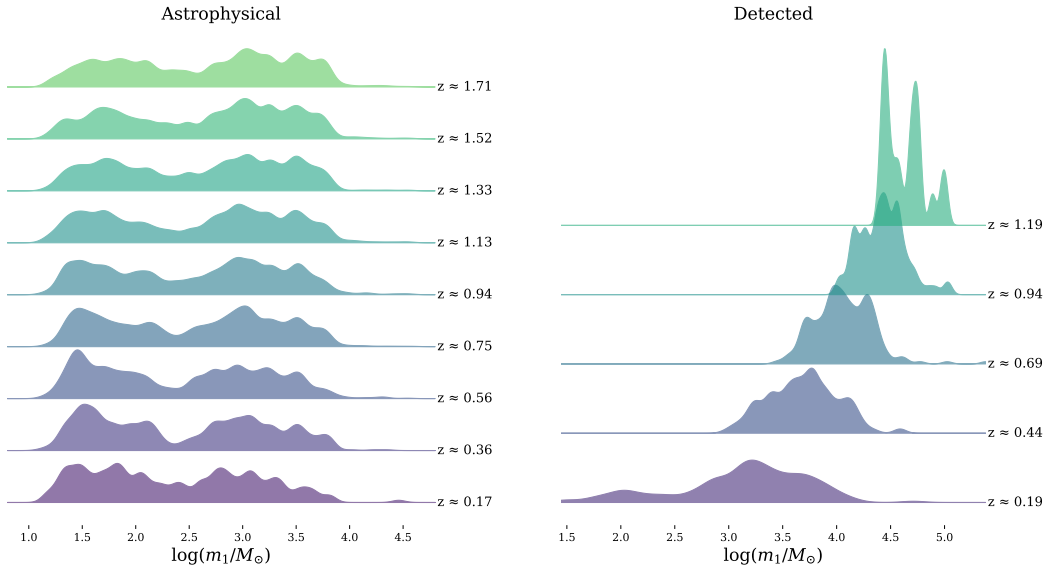

Figure 4.2: Distribution of the logarithm of the primary source and detector mass of the B-POP global catalog at different redshift ranges. The left plot shows the source mass, while in the right one shows the detector mass. Both distributions are smoothed using Gaussian KDE [36]. The redshift evolution is far from being trivial and both distributions features complex structures.

In the first part of the analysis the full catalog will be used to infer H_0, while in the second part, two catalogs will be analyzed to obtain population and cosmological constraints.

4.2 Generation of PE posterior samples and injection samples

To perform hierarchical Bayesian inference and obtain population and cosmological information, we generated 700 PE posterior samples and $1.278 \cdot 10^8$ injection samples, assuming the O5 sensitivity [41, 42, 43].
Starting from a catalog of source frame parameters m_{1s}, m_{2s} and merger redshift z, our goal is to generate a set of detector frame GW events specified by the detector frame masses m_{1d}, m_{2d} and the luminosity distance d_L, in such a way to be compatible with the ICAROGW pipeline and its PE posterior samples and injection samples.

Fig 4.3 shows a flowchart of the steps needed to generate PE posterior samples and injection samples, starting from some population samples.

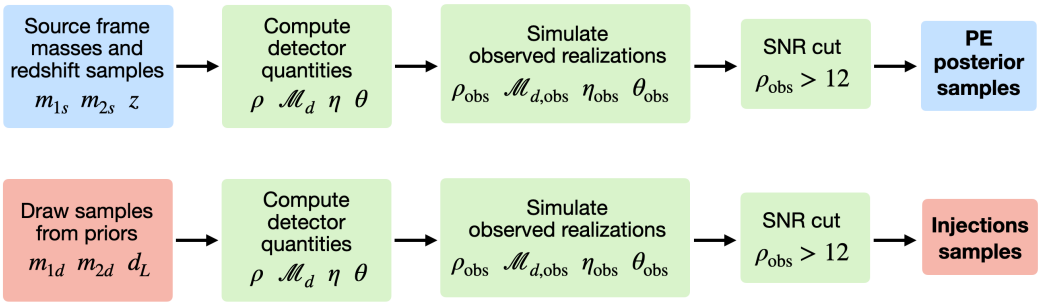

Figure 4.3: Logical structure of the generation of PE posterior samples (in the first line) and of injection samples (in the second line)

So the first thing we did was to convert source frame masses to detector frame masses, using the source frame conversions formulas in Eq. 1.17, Eq. 1.18 and Eq. 1.8.
Rather than working directly with (m_{1d}, m_{2d}, d_L), we will use the following variables:

- the detector frame chirp mass $\mathcal{M}_d$ (which is introduced in Sec. A.1 and in Sec. 1.2):

$$\mathcal{M}_d = \frac{(m_{1d}\ m_{2d})^{3/5}}{(m_{1d} + m_{2d})^{1/5}} \tag{4.1}$$

- the symmetric mass ratio:

$$\eta = \frac{m_{1d} m_{2d}}{(m_{1d} + m_{2d})^2} \tag{4.2}$$

- the matched filtering signal to noise ratio (SNR):

$$\rho \ = \ \rho_0 \ \theta \left(\frac{\mathcal{M}_d}{\mathcal{M}_{d,0}} \right)^{5/6} \frac{d_{L,0}}{d_L} \ S\left(f_{\rm ISCO}(m_{1d}, m_{2d}) \right) \tag{4.3}$$

  where θ is a projection factor that is defined as the quadratic sum of the antenna response functions F_+ and $F_{\rm x}$; it takes into account the orientation of the binary relative to the detector network. ρ_0, $\mathcal{M}_{d,0}$ and $d_{L,0}$ are reference

dimensional scaling factors which ensure that the SNR remains dimensionless. They are set to $\rho_0 = 12$, $\mathcal{M}_{d,0} = 25M_\odot$ and $d_{L,0} = 2.5Mpc$ [41]. Finally, $S\left(f_{\rm ISCO}(m_{1d}, m_{2d})\right)$ is a smooth attenuation obtained with a sigmoid function that is introduced because astrophysical simulations contain very massive stars that could make the SNR diverge or grow high, thus appearing as a loud signal. But detectors are not as sensitive as they should be in order to measure those kind of signals, which appear at low frequencies. This is why to signals with innermost stable circular orbit (ISCO) frequency [11] $f_{\rm ISCO} < 20Hz$, it is assigned SNR=0.

The following step is needed to simulate the observed realizations $(\rho_{\rm obs}, \mathcal{M}_{d,\rm obs}, \eta_{\rm obs}, \theta_{\rm obs})$, by sampling from likelihoods that model the measurement process, conditioned on the true values $(\rho, \ln \mathcal{M}_d, \eta, \theta)$. The following conditional distributions are used:

- The SNR is modeled with a non central χ^2 distribution with parameter ρ^2 and degrees of freedom equal to twice the number of detectors N_d:

$$\rho^2_{\rm obs} \sim \chi^2\left(2N_d, \lambda = \rho^2\right) \tag{4.4}$$

- The chirps mass is modelled in log space as a Gaussian:

$$\log\left(\mathcal{M}_{d,\rm obs}\right) \sim \mathcal{G}\Big(\log\left(\mathcal{M}_d\right), \sigma_{\log\mathcal{M}}\left(\rho_{\rm obs}\right)^2\Big) \tag{4.5}$$

  where the standard deviation $\sigma_{\log\mathcal{M}}(\rho_{\rm obs}) \propto \rho^{-1}_{\rm obs}$.

- The symmetric mass ratio is modelled using a truncated Gaussians, conditioned on its true value and on $\rho_{\rm obs}$:

$$\eta_{\rm obs} \sim \mathcal{G}_{[\eta_{\rm min},\eta_{\rm max}]}\Big(\eta, \sigma_\eta\left(\rho_{\rm obs}\right)^2\Big) \tag{4.6}$$

  in which $\sigma_\eta(\rho_{\rm obs}) \propto \rho^{-1}_{\rm obs}$.

- The projection factor θ follows a truncated Gaussian, conditioned on its true value and on $\rho_{\rm obs}$:

$$\theta_{\rm obs} \sim \mathcal{G}_{[\theta_{\rm min},\theta_{\rm max}]}\Big(\theta, \sigma_\theta\left(\rho_{\rm obs}\right)^2\Big) \tag{4.7}$$

  with $\sigma_\theta(\rho_{\rm obs}) \propto \rho^{-1}_{\rm obs}$

Afterwards, a detection threshold is applied to the simulated observed SNR:

$$\rho_{\rm obs} \geq \rho_{\rm thr} \equiv 12 \tag{4.8}$$

All the events that do not satisfy this condition are rejected, all the others are kept as detection events. PE posterior samples and injection samples sets are generated in the following way:

PE posterior samples: Given the observed realizations that survived the cut, $\mathrm{ev} = (\rho_{\mathrm{obs}}, \mathcal{M}_{\mathrm{d,obs}}, \eta_{\mathrm{obs}}, \theta_{\mathrm{obs}})$, the posterior samples are drawn from the following distribution:

$$p(\rho, \mathcal{M}_d, \eta, \theta \,|\, \mathrm{ev}) \;\propto\; \mathcal{L}(\,\mathrm{ev}\,|\rho, \mathcal{M}_{\mathrm{d}}, \eta, \theta)\; \pi(\rho, \mathcal{M}_{\mathrm{d}}, \eta, \theta) \tag{4.9}$$

where $\mathcal{L}(d \mid \rho, \mathcal{M}_d, \eta, \theta)$ represents the product of the distributions defined above. $\pi(\rho, \mathcal{M}_d, \eta, \theta)$ takes into account the change of variables from (m_{1d}, m_{2d}, d_L) to $(\mathcal{M}_d, \eta, \rho_t rue)$ and it is implemented in the code through the Jacobian factor:

$$\pi(\rho, \mathcal{M}_d, \eta, \theta) \;\sim\; \frac{1}{J(\rho, \mathcal{M}_d, \eta, \theta)} \tag{4.10}$$

Finally, the PE posterior samples are drawn from the set of candidates $(\rho, \mathcal{M}_d, \eta, \theta)$, by importance resampling using the weight $1/\pi(\rho, \mathcal{M}_d, \eta, \theta)$ and the conversion to the variables (m_{1d}, m_{2d}, d_L).

Injection set: Synthetic events are generated by drawing the primary and secondary masses from power law distributions and the luminosity distance from a power law distribution proportional to d_L^2, as discussed in Sec. A.3. Then the SNR cut is applied and the events that respect the threshold condition are kept as injections.

4.3 Global catalog Normalizing Flows fit

In this section, I conduct the analysis on the global catalog, so without differentiating between the different formation channels.
The first quantity that can be extracted from the rate $R(z)$ is the number of predicted astrophysical observations, which follows Eq. 2.16:

$$N_{\mathrm{astro}} \sim 6.6 \cdot 10^4 \text{ events/year} \tag{4.11}$$

Afterwards, the global catalog samples $\log(m_1), \log(m_2), z$ were given as an input to the *zuko* library Normalizing Flows model, in order for it to fit the distributions. We chose the logarithm of the masses instead of the direct masses, as this choice improves the performance of the Normalizing Flows. We finally implemented the Jacobian J correction to the variable change:

$$\begin{aligned} p_{\mathrm{NF}}(m_1, m_2, z) &= p_{\mathrm{NF}}\Big(\log(m_1), \log(m_2), z\Big) \frac{1}{|J|} \\ &= p_{\mathrm{NF}}\Big(\log(m_1), \log(m_2), z\Big) \frac{1}{m_1 m_2} \end{aligned} \tag{4.12}$$

The model was trained and validated following the methodology introduced in Sec. 2.3.2 with the following architecture parameters:

number of layers	5
hidden features	56
learning rate	0.0015
batch size	3000
number of iterations	200
patience	5

The flow structure is deeper and more complex with respect to the ones used for the POWERLAW + EVOLVING PEAK discussed in Sec. 3.1. This is reasonable since the B-POP distribution is complex and it shows some fine structures that are more difficult to capture. Still, a compromise between the lightest but less precise model and the heaviest but most precise model, had to be made in order to preserve computational times, but ensuring an exhaustive description of the population. Indeed, the hierarchical likelihood computation time was of $\sim 5s$. The fit in Fig. 4.4 shows that the most important B-POP features are correctly captured, but the fit could be improved on more fine structures increasing the number of splines in the NSF. It is set to the default *zuko* value of 8 (as discussed in Sec. 2.3.2) and increasing it has a substantial worsening of the computation time and on the overheating of the computing device [1].
Thanks to the computation of $p_{\rm NF}(m_1, m_2, z)$, we estimated the expected number of detections, following Eq. 2.17. We assigned a Poissonian uncertainty to the measurement, because the detection process is modeled to be Poissonian, as described in Sec. 2.1. Then, since we generated 700 PE posterior samples, the expected number of detections uncertainty is equal to $\sqrt{700} \sim 26$:

$$N_{\rm exp} = 713 \pm 26 \ \text{events/year} \tag{4.13}$$

The obtained value is compatible with the expected value within 1σ of error. This result shows that:

- $N_{\rm exp} < N_{\rm astro}$ since the simulated O5 like detectors will not be able to measure every astrophysical event, because of the *selection bias* described in Sec. 2.1;
- the Normalizing Flows architecture has been correctly implemented into the ICAROGW pipeline since the number of expected detections is compatible with the number of PE posterior samples;

Finally, I performed inference on H_0 (the results are visible in Fig. 4.5) and we recovered a H_0 unbiased estimate. Thus demonstrating that the systematic bias studied in [20] can be completely removed thanks to the Normalizing Flows. Moreover, this also suggests a new methodology to test astrophysically simulated fiducial catalogs, using GW observations. Indeed, in a O5 era, GW signals can be used to validate a synthetic astrophysical catalogs though inference on cosmological quantities.

[1] Indeed, all the computations obtained in this work were performed on a MacBook Air M1 2020.

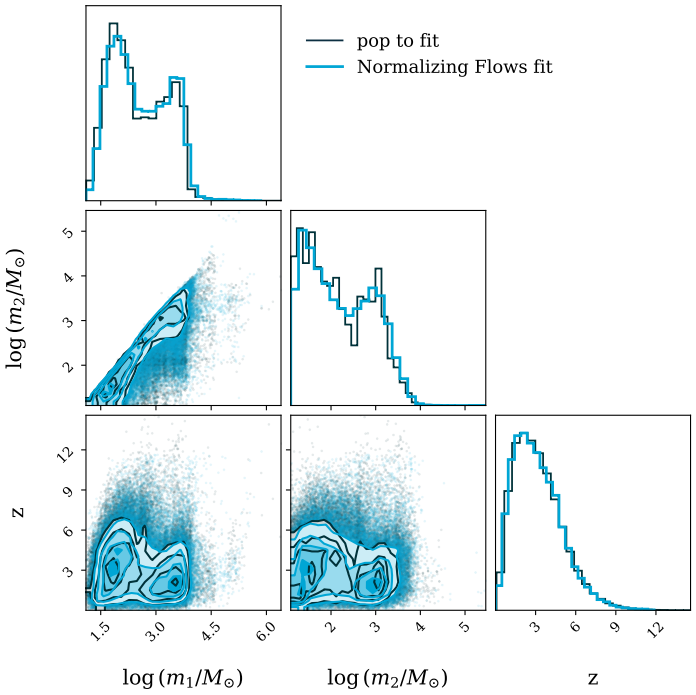

Figure 4.4: B-POP global catalog (in black) and Normalizing Flows fit (in light blue): the main population features are well modeled by the Normalizing Flows, however it provides a less precise fit of the fine structures of the distribution.

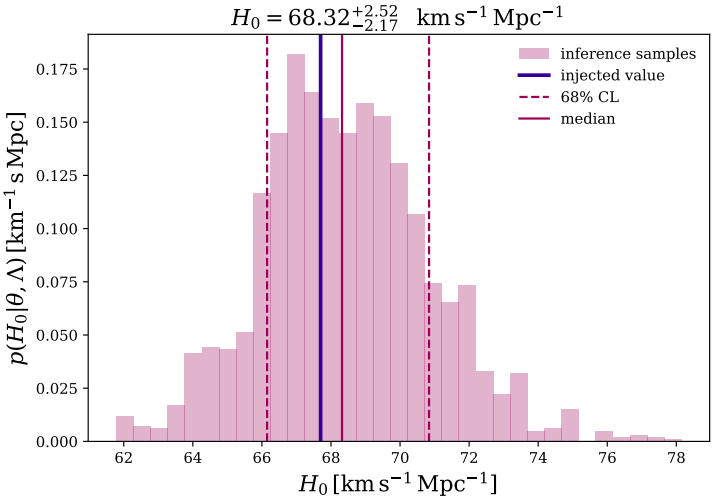

Figure 4.5: MCMC inference on H_0 performed with 4 walkers and 600 steps. The median of the posterior distribution is compatible with the true value for H_0.

4.4 Two catalogs Normalizing Flows fit

To extend the discussion on the B-POP catalog, I conducted a further analysis considering two different populations, which represent the isolated binaries channel and the dynamical channel. Using the methodology illustrated in Sec. 2.2, it is possible to constrain not only cosmological parameters, but also population parameters, such as the abundance of one catalog with respect to the others.

To have a better understanding of the four catalogs, I report some useful quantities in Tab. 4.1. We notice that the catalogs have a different number of samples. Furthermore, the ratios between the number of samples in a single catalog N_i with respect to the number of samples in the total catalog $N_{\rm tot}$, is equal to the ratio between the merger rate of the single catalog $\tilde{R}_i \equiv \int dz\ R_i(z) \frac{dV_c}{dz} \frac{1}{1+z}$ with respect of the merger rate of the full catalog $\tilde{R} \equiv \int dz\ R(z) \frac{dV_c}{dz} \frac{1}{1+z}$. This shows that the isolated binaries channel is the one containing more events and which mostly contributes to the total merger rate. The second most populated channel is the globular clusters one, which is the most important dynamical BBH channel, as discussed in Sec. 1.4.

catalog	N_i	$N_{\rm astro}$	$N_{\rm i}/N_{\rm tot} = \tilde{R}_i/\tilde{R}$ [%]
isolated binaries	782800	41866	62.77
globular clusters	210349	11292	16.87
nuclear clusters	136914	7397	10.98
young clusters	116964	6331	9.38

Table 4.1: Characteristic of the four B-POP catalogs, representing the main BBH formation channels: the isolated binaries channels dominates over the other ones, thus it is the one contributing the most to the total merger rate. Indeed, the isolated binaries catalog has the bigger number of astrophysical events, followed by the globular, nuclear and young clusters catalogs.

The choice we made was to work with two catalogs: the isolated binaries catalog and the dynamical catalog, which is given by the sum of the globular clusters, nuclear clusters and young clusters catalogs. Then, the analysis was conducted following Eq. 2.18, in which λ_i represent the *multiplicative factors*. They act as absolute weights for each astrophysical formation channel: they are not normalized fractions, but free scaling parameters that determine the overall number of events attributed to each channel in the total population.

Eq. 2.18 shows that the single catalog needs to be modeled by the Normalizing Flows, in order to estimate $p^i_{\rm NF}(m_1, m_2, z)$. Thus, two different models have been trained using the *zuko* Normalizing Flows, with the following architectures:

for the isolated binaries catalog:

number of layers	3
hidden features	16
learning rate	0.01
batch size	2048
number of iterations	200
patience	150

for the dynamical channel catalog:

number of layers	4
hidden features	24
learning rate	0.05
batch size	2048
number of iterations	200
patience	90

This choice was made in order to improve the computation times as much as possible, since performing a precise MCMC inference on many parameters requires a lot of time and engine power; which in this work were limited. In this case too, we chose to fit the logarithm of the primary and secondary masses in order improve the performance of the NSF. Then we corrected with the change of variables Jacobian J of the transformation, which is the same as Eq. 4.12.

In Fig. 4.6 and Fig. 4.7 the two different Normalizing Flows fit of the isolated binaries and dynamical catalogs are reported. The plots show that the architecture is deep enough to capture all the main features of the distribution, it worsen slightly on some minor structures. The most quickly, so the lightest, but precise Normalizing Flows was thus found: the hierarchical likelihood estimation time was of $\sim 6s$.

In the last part of the work, I will describe the tests that we have conducted on the B-POP synthetic population, in the case of two catalogs. Indeed, this methodology suggests that with astrophysically simulated fiducial catalogs we can perform not only cosmological studies, but also population analysis.

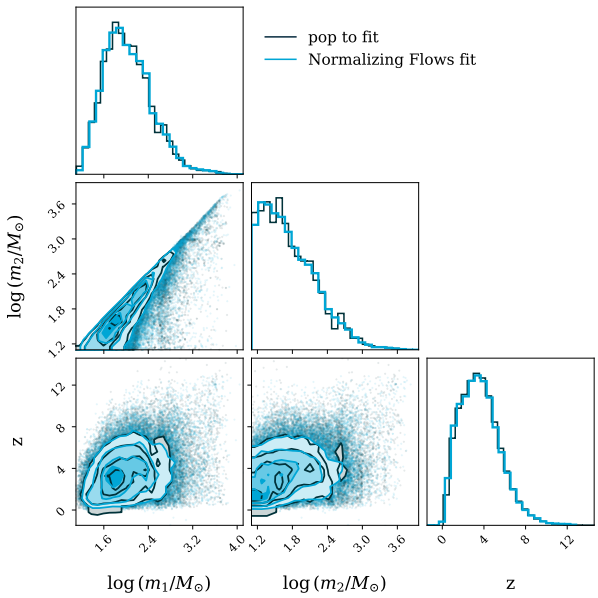

Figure 4.6: Isolated binaries channel Normalizing Flows fit (in light blue). The true population (in black) has more events at lower primary and secondary masses with respect to the other channels. The Normalizing Flows correctly fits the main features of the distributions.

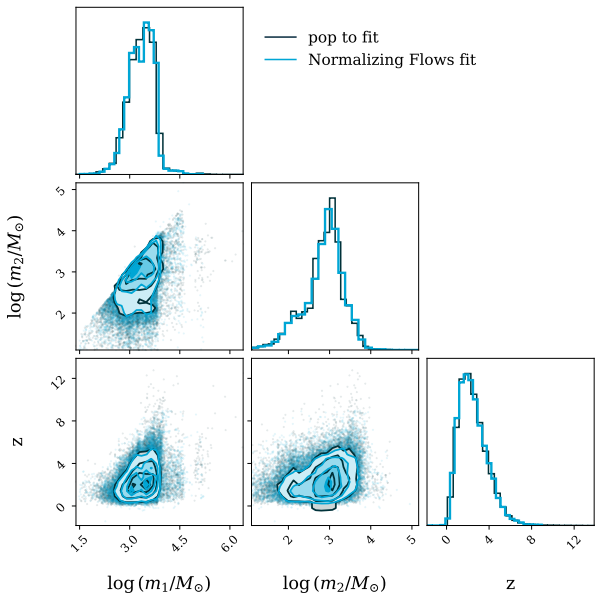

Figure 4.7: Normalizing Flows fit (in light blue) of the dynamical formation channel (in black) that captures the main features of the distributions. By changing the number of bins in the NSF, the more fine structures are better fitted, but the computation time increases in a non negligible way.

4.4.1 Inference on multiplicative factors λ_i

The first test that I did was to perform inference on the two multiplicative factor $\lambda_{\text{isolated}}$ and $\lambda_{\text{dynamical}}$ separately, using the B-POP simulated data, described in Sec. 4.2. I did this to verify if the fiducial values $\lambda_i = 1$ (with $i \in [\text{isolated}, \text{dynamical}]$), which are the values with which the catalog is built, are supported in the posterior distributions. During this inference process, I fixed the cosmological quantities to $H_0 = 67.74$ km s^{-1}Mpc^{-1}, $\Omega_m = 0.3089$ and the other multiplicative factor to its fiducial value $\lambda_i = 1$. In this way, the goodness of the Normalizing Flows can be tested: if the posterior distributions for λ_i do not support the fiducial values, then the fit is not correct. Results are visible at Fig. 4.8, which shows that the fiducial values are supported in both cases.

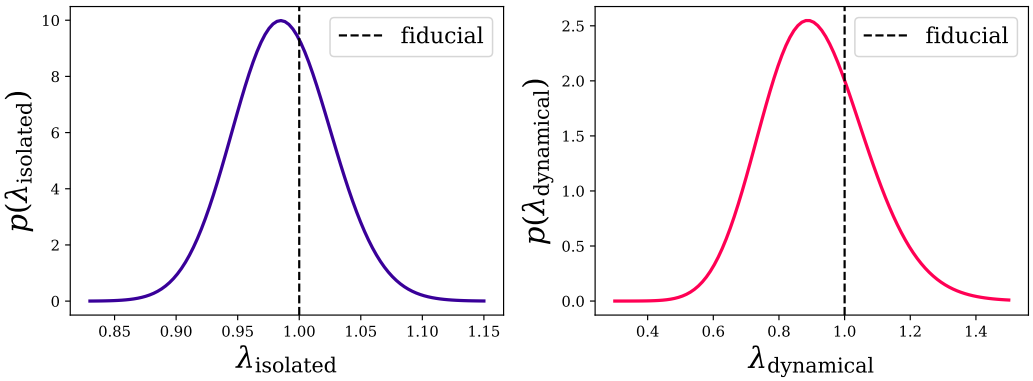

Figure 4.8: Posterior distributions for the multiplicative factors λ_i obtained with hierarchical Bayesian inference. The distributions show that the population has been correctly modeled: the fiducial values λ_i are correctly supported by the posteriors.

Then, I analyzed the correlations and constraints between the two formation channels, through combined inference on both the multiplicative factors $\lambda_{\text{isolated}}$ and $\lambda_{\text{dynamical}}$. Results are visible in Fig. 4.9 and show a negative correlation pattern. This behavior is expected: when the contribution of one catalog increases, the other one must correspondingly decrease in order to keep the overall number of detected GW events fixed. In other words, if the two catalogs have to explain the same set of events, then an excess in one catalog reduces the multiplicity of the other.

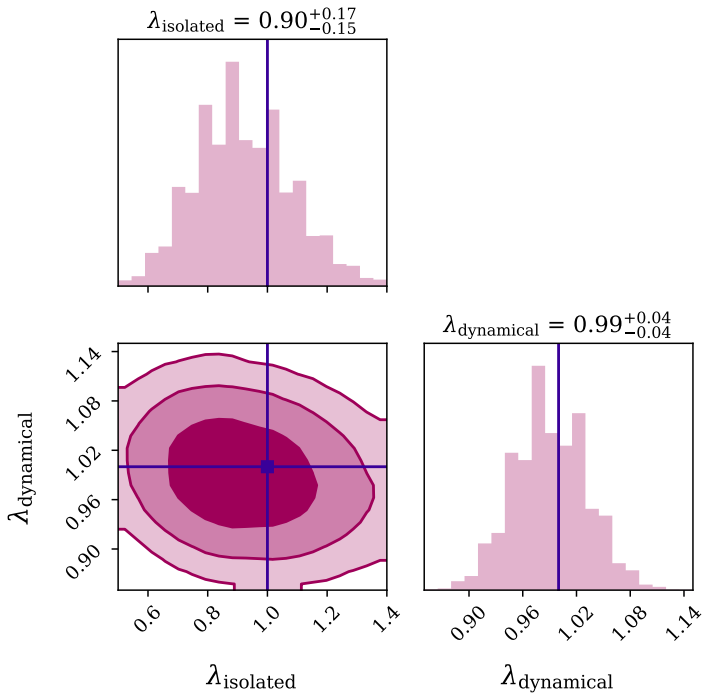

Figure 4.9: Posterior distributions obtained performing inference on the two catalogs: $\lambda_{\rm isolated}$ and $\lambda_{\rm dynamical}$. The fiducial values λ_i are supported and compatible within 1σ of error. A negative correlation pattern is visible and coherent with the expected results.

These results are the first proof that the Normalizing Flows fit was good. Yet, it is not sufficient to show that the fiducial values for the multiplicative factors are supported, we also need to make sure that the fit is also recovering the injected value of H_0. In the case of B-POP, the catalogs were simulated assuming $H_0 = 67.74$ km s^{-1}Mpc^{-1}[40].

4.4.2 Combined inference on the Hubble constant H_0 and multiplicative factors λ_i

The second test that we conducted was to perform combined inference on the Hubble constant H_0 and on the multiplicative factors of the isolated binaries catalog $\lambda_{\rm isolated}$ and of the dynamical formation catalog $\lambda_{\rm dynamical}$. In this way, a population constraint, validated on the H_0 estimate, is obtained. Results are visible in Fig. 4.10. The value of H_0 is compatible with the injected value within 1.4σ of error. $\lambda_{\rm isolated}$ is compatible with its fiducial value within 0.9σ of error and also $\lambda_{\rm dynamical}$ is compatible with its fiducial value within 1.8σ of error. This is the second proof that shows that the Normalizing Flows fit was correctly implemented.

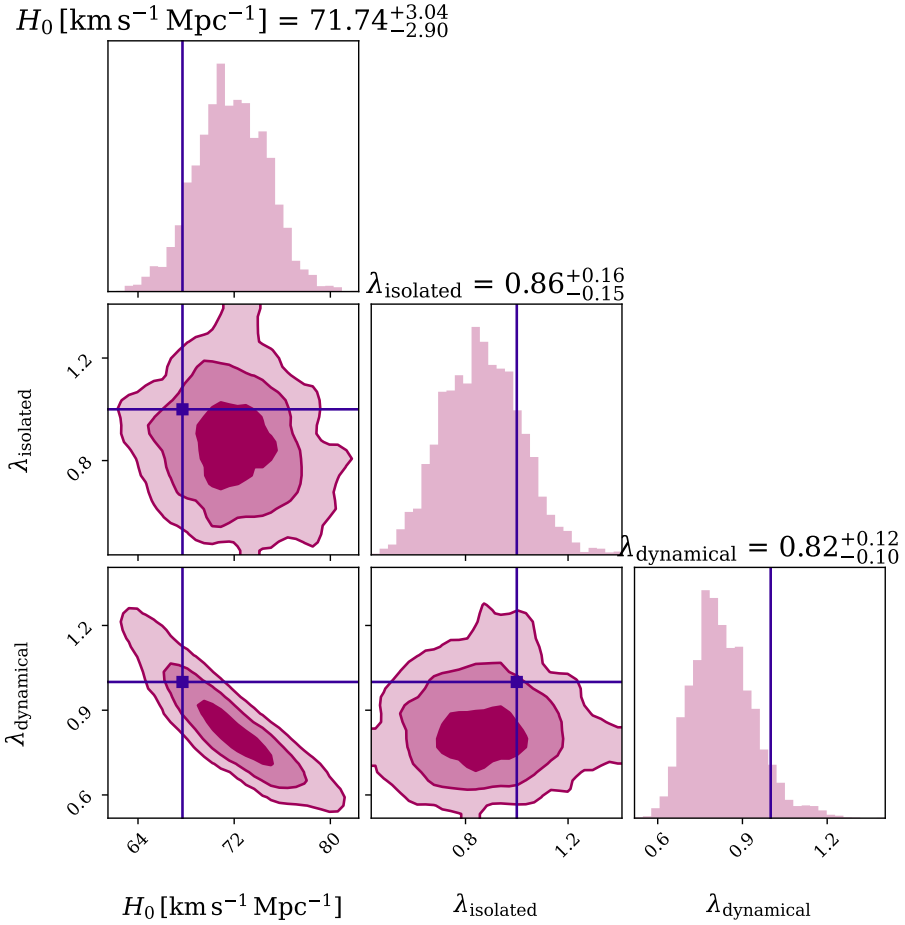

Figure 4.10: Combined inference on H_0, $\lambda_{\mathrm{isolated}}$ and $\lambda_{\mathrm{dynamical}}$, obtained with 13 walkers and 600 steps.

4.4.3 Inference on H_0 using O3 data

Finally, the model was tested on the real O3 data, by using the GWTC-3 PE posterior and injections samples, both described in Sec. A.3. I did not consider the latest GWTC-4.0 catalog as the dataset was only released on August 26th 2025, while this thesis was being finalized.
The value of the Hubble constant H_0 was inferred, fixing the multiplicative factors at their fiducial values, since they were supported by all the posterior distributions we recovered. In Fig. 4.11 the H_0 posterior is shown and we find the following estimation for H_0 at 68% of CL:

$$H_0 = 65.95^{+2.99}_{-3.14}\ \mathrm{km\ s^{-1}Mpc^{-1}} \tag{4.14}$$

Thus, if we assume that the B-POP catalog correctly describes GWTC-3, then the H_0 posterior is more consistent with the Planck measurement. This proves that the B-POP model furnishes a good description for the masses and redshift distributions of BBHs observed by LIGO-Virgo-KAGRA.

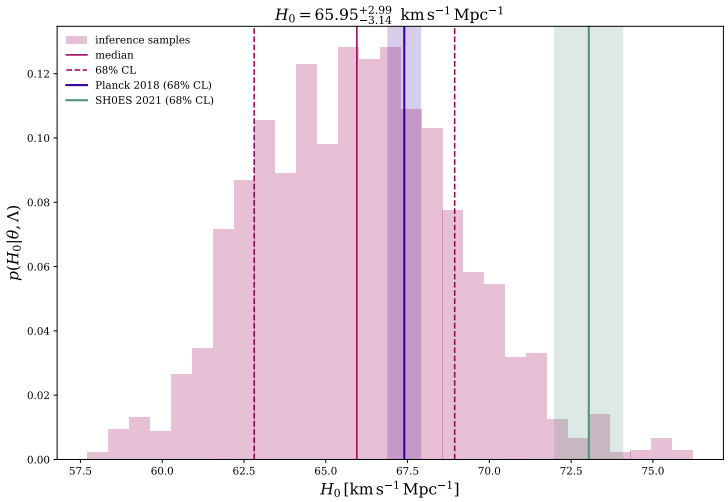

Figure 4.11: Posterior distribution of H_0 obtained through MCMC inference with 13 walkers and 600 steps using the Normalizing Flows model to fit the B-POP population on GWTC-3 PE posterior and injection samples. The magenta lines correspond to the median and 68% credible interval of the inferred posterior from the MCMC samples. The purple line represents the Planck 2018 measurement ($H_0 = 67.4 \pm 0.5$ km s^{-1} Mpc^{-1}) [8] with its 68% CL uncertainty (shaded band), while the red line shows the SH0ES 2021 result ($H_0 = 73.04 \pm 1.04$ km s^{-1} Mpc^{-1}) [9] with its 68% CL uncertainty.

4.5 Summary of the findings

The results of the methodology that we applied in this Chapter show that:

- When simulating GW events from the B-POP fiducial catalog and fitting the different populations with the Normalizing Flows, the injected astrophysical model is correctly reconstructed.
- We found a criterion to establish the goodness of the Normalizing Flows fit which works with astrophysical fiducial populations.
- We also found a criterion to establish the goodness of the synthetic astrophysical fiducial catalog and it worked with the B-POP catalog.
- In O5 like era, population constraints on the formation channels can be put. In O3 like era, only cosmological quantities can inferred using synthetic astrophysical fiducial catalogs.

Conclusions

The main objective of this work was to devise non-parametric machine learning tools to alleviate systematic biases for GW cosmology. Indeed, I started with the purpose of solving the systematic bias that appears when the BBH masses distributions are modeled with the phenomenological model. To address this issue, I developed a new methodology based on Normalizing Flows and I implemented it into the ICAROGW pipeline. The use of Normalizing Flows allows for the implementation of non-parametric models of the masses and redshift distributions, that can correctly recover the unbiased H_0 estimation, thanks to their ability to capture redshift evolutions and complex features in the distributions.

I validated the model using a synthetic redshift evolving mock catalog sampled from a POWERLAW+EVOLVING PEAK distribution. Three different analyses have been conducted. In the first one, a BBH population with masses evolving in redshift has been analyzed, with parametric models non evolving in redshift. I performed inference on H_0 and, as expected, I found the H_0 estimation biased up to 3σ. Then I approximated the evolution in redshift of the mass spectrum with a Normalizing Flows, which corrected the H_0 bias thanks to its ability to fit the redshift evolving populations. Finally, I used two different Normalizing Flows to fit two different populations: the POWERLAW and the redshift EVOLVING PEAK. In this case too, I recovered the injected estimation of H_0 and no bias was measured. This showed that the Normalizing Flows flexibility in capturing redshift evolving populations has successfully replaced the phenomenological model in this discussion.

I tested the model on an astrophysical fiducial catalog, which was simulated by researchers at the GSSI using the B-POP population synthesis code. The global catalog is made of four distinct catalogs representing the BBH formation channels: the isolated binaries channel, the globular clusters channel, the nuclear clusters channel and the young clusters channel. Firstly, I fitted the full catalog using the Normalizing Flows which successfully modeled the global population features and led to an unbiased estimate of H_0. Then I studied two catalogs individually representing the isolated binaries channel and the dynamical channel. This allowed not only for cosmological inference, but it also allowed us to constrain the relative contribution of the two different catalogs through their multiplicative factors. The results showed that the fiducial values used to generate the catalog are supported by the posteriors, confirming the goodness of the Normalizing Flows fit. Moreover, I analyzed the correlations between the two channels and, as expected, I found an anti correlation pattern. I then performed a combined inference on H_0 and on the two multiplicative

factors and I recovered the expected values for all three quantities. Finally, I inferred H_0 using the GWTC-3 real data obtaining the following estimation for the Hubble constant:

$$H_0 = 65.95^{+2.99}_{-3.14}\ \mathrm{km\ s^{-1}Mpc^{-1}}$$

This result shows that, if we describe B-POP as BHs formed in isolated stellar binaries and in dynamical environments, then the simulated catalog represents a good fit for GWTC-3.

I developed a methodology which demonstrated that Normalizing Flows are not only capable of reproducing astrophysically complex populations but they can also be used to disentangle the contributions from different formation channels. This methodology can be used to validate astrophysically simulated fiducial catalogs with GW observations. Finally, since the simulations have been obtained under realistic observational conditions corresponding to the O5 era expected sensitivity, then GW cosmology and population studies will also be able to constrain astrophysical quantities such as the relative abundance of every BBH formation channel.

Acknowledgments

I would like to thank Dr. Simone Mastrogiovanni for his guidance throughout my thesis work. His support and advice have been invaluable both scientifically and on the human level. This experience has been highly formative and will remain a fundamental part of my growth, hopefully guiding me in my future career. I am also grateful to him for giving me the opportunity to gain direct research experience by welcoming me into the GravitySirens group.

My sincere thanks also go to Dr.ssa Sarah Ferraiuolo, Dr. Grégoire Pierra, Dr. Leonardo Iampieri, Dr. Alberto Colombo and Dr. Alessandro Buchicchio for their advice and guidance during the course of my thesis work.

I would also like to thank Prof.ssa Raffaella Schneider for her insightful feedback and the stimulating discussion during my defense.

I am deeply thankful to my parents, Alessandra and Sandro, for always encouraging and supporting me in pursuing both my studies and my passions.

I would finally like to thank my loved ones: Michela, Francesca, Sara, Noemi, Romeo, and Rica, for always being supportive and by my side.

Appendix A

Appendix

A.1 Gravitational Waves in General Relativity

Assuming the perturbed metric tensor in a flat spacetime background to be:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \tag{A.1}$$

where $\eta_{\mu\nu}$ is the Minkowski background metric:

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & +1 \end{pmatrix}. \tag{A.2}$$

and $h_{\mu\nu}$ is the perturbation and it is considered to be small: $h_{\mu\nu} << 1$ (weak field approximation). To obtain information on the dynamics of $h_{\mu\nu}$, Einstein's field equations are considered:

$$R_{\mu\nu} = \frac{8\pi G}{c^4}\left(T_{\mu\nu} - \frac{1}{2}g_{\mu\nu}T\right) \tag{A.3}$$

where $R_{\mu\nu}$ is the Ricci tensor, $T_{\mu\nu}$ is the stress-energy tensor and $T = g^{\mu\nu}T_{\mu\nu}$ is its trace. It is found [1] as a solution for the perturbation:

$$\begin{cases} \Box\bar{h}_{\mu\nu} = -\frac{16\pi G}{c^4}T_{\mu\nu} \\ \frac{\partial}{\partial x^\mu}\bar{h}^\mu_\nu = 0 \end{cases} \tag{A.4}$$

where $\Box$ is the d'Alambertian in flat spacetime:

$$\Box = \eta^{\alpha\beta}\frac{\partial}{\partial x^\alpha}\frac{\partial}{\partial x^\beta} = -\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2 \tag{A.5}$$

[1] In this Section, the main results showed in [11] are summarized.

and the second equation in A.4 is the harmonic gauge condition in which the trace reversed tensor $\bar{h}_{\mu\nu}$ is defined as:

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h \tag{A.6}$$

and where $h = h^{\mu}_{\ \mu} = \eta^{\mu\nu}h_{\mu\nu}$ is the trace of $h_{\mu\nu}$.
When considering $T_{\mu\nu} = 0$ (outside of the source) the first equation in A.4 becomes:

$$\left(-\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2\right)\bar{h}_{\mu\nu} = 0 \tag{A.7}$$

which clearly shows that a small perturbation of a flat spacetime follows a D'Alambertian like equation, behaving like a wave travelling at the speed of light.

The solution to Einstein's equations for $\bar{h}_{\mu\nu}$ are:

$$\bar{h}_{\mu\nu} = \frac{4G}{c^4}\int_V \frac{T_{\mu\nu}(t - \frac{|\vec{x}-\vec{x'}|}{c})}{|\vec{x}-\vec{x'}|}d^3x' \tag{A.8}$$

where the integral is performed among the 3D source volume, $\vec{x}$ is the position of the observer with respect to the source and $\vec{x'}$ is the position of an element of the source within a coordinate system centered on the source itself. This solution has the form of a retarded potential and it reflects the causal propagation of gravitational effects at the speed of light.

To understand the physical degrees of freedom of the system we chose the TT (Transverse-Traceless) gauge. In the TT gauge $h_{\mu\nu}$ and $\bar{h}_{\mu\nu}$ coincide and are traceless:

$$h_{\mu\nu} = \bar{h}_{\mu\nu} \tag{A.9}$$

It can be shown that for a plane gravitational wave propagating along the x axis, the small perturbation tensor has the following form:

$$h^{TT}_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & h_{yy} & h_{yz} \\ 0 & 0 & h_{yz} & -h_{yy} \end{pmatrix} \tag{A.10}$$

the gravitational wave has only 2 degrees of freedom that corresponds to 2 polarization states: if $h_{yz} = 0$ and $h_{yy} \equiv h_+ \neq 0$ the wave has a *plus* polarization, while if $h_{yy} = 0$ and $h_{yz} \equiv h_{\mathrm{x}} \neq 0$ the *cross* polarization is found.
Assuming that the size ϵ of the source is much smaller than the wavelength of the emitted gravitational wave $\lambda_{\mathrm{GW}} = \frac{2\pi c}{\omega}$, then: $\omega\epsilon << c$. Since the velocity of the system emitting the gravitational wave can be written as $v \sim \omega\epsilon$, then $v << c$ (slow motion approximation).

When solving Eq. A.4 in the weak field, slow motion approximation, defining $|\vec{x}| \equiv r$, the following solution is obtained:

$$\begin{cases} \bar{h}^{\mu 0} = 0 \\ \bar{h}^{ik}(t,r) = \frac{2G}{c^4 r} \frac{d^2}{dt^2}\, q^{ik}(t - \frac{r}{c}) \end{cases} \tag{A.11}$$

where q^{ik} is the quadrupole moment tensor and it is defined as:

$$q^{kn}(t) = \frac{1}{c^2} \int_V T^{00}(t, \vec{x}) x^k x^n d^3x \qquad k, n = 1, 2, 3 \tag{A.12}$$

It is worth noticing two things: first of all, Eq. A.11 shows that any form of mass or energy can generate gravitational waves if and only if the second time derivative of the system's quadrupole moment does not vanish. Moreover the factor that regulates the intensity of the gravitational wave is very small: $\frac{G}{c^4} \sim 8 \cdot 10^{-50} \frac{\mathrm{s}^2}{\mathrm{g\ cm}}$ and this justifies why gravitational waves are extremely weak.

This formalism can be applied to a compact binary system to study the inspiral, the merging and the ringdown phase of two coalescing compact objects (shown in Fig. A.1). The information from which the masses of the two coalescing objects and the luminosity distance of the merger can be inferred is the gravitational wave phase and amplitude.

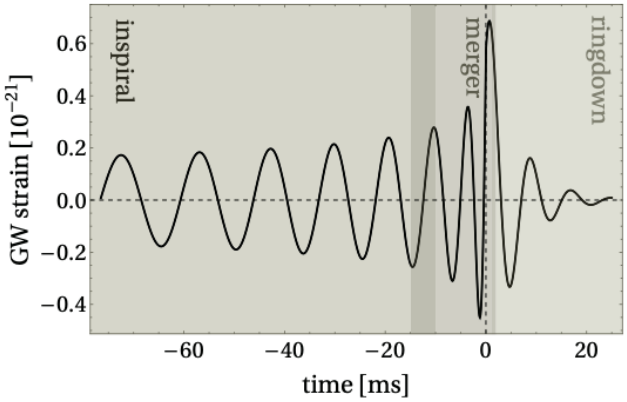

Figure A.1: Gravitational Wave signal from the inspiral, merger and ringdown phases of a compact binary system coalescence. During the inspiral phase the two compact objects start orbiting around each other and their orbital frequency increases as the two get closer. Then they merge into a new black hole, which oscillates in its ringdown phase. Image taken from[11].

The wave phase of the GW signal has the following form:

$$\Phi(t) = \int_o^t 2\pi \nu_{GW}(t) dt + \Phi(t = 0) \tag{A.13}$$

in which the wave frequency ν_{GW} is equal to:

$$\nu_{GW}(t) = \frac{5^{3/8}}{8\pi} \left(\frac{c^3}{G\mathcal{M}} \right)^{5/8} \left(\frac{1}{t_c - t} \right)^{3/8} \tag{A.14}$$

where t_c represents the *coalescence time* and the term $\mathcal{M}$ is the *chirp mass* and it depends on the binary source masses:

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}} \tag{A.15}$$

The chirp mass can be measured from the integrated gravitational wave phase:

$$\Phi(t) = -\frac{1}{16}\left(\frac{c^3}{G}\right)^{5/3} (\pi \mathcal{M} \nu_{GW})^{-5/3} - 2\left(\frac{c^3 t_c}{5G\mathcal{M}}\right)^{5/8} + \Phi(t=0) \tag{A.16}$$

Since the expansion of the Universe needs to be taken into account, the quantities measured at the detector are *redshifted*, this implies that the *detector chirp mass* is the following:

$$\mathcal{M}_d = \mathcal{M}\,(1+z) \tag{A.17}$$

The second quantity that can be measured is the *luminosity distance* d_L of the source from the gravitational wave amplitude h_0 measured by the detector at the time t_d:

$$h_0(t_d) = \frac{4\pi^{2/3} G^{5/3}}{c^4} \frac{\mathcal{M}_d}{d_L(t_d)} \left[\mathcal{M}_d \nu_{GW}\,(1+z)\right]^{2/3} \tag{A.18}$$

A.2 Numerical stability of the ICAROGW pipeline

The Monte Carlo integration sum over the PE posterior samples in Eq. 2.4, can be written as:

$$\int \mathcal{L}(x_i|\theta,\Lambda) \frac{dN}{dtd\theta}(\Lambda) d\theta = \frac{T_{\rm obs}}{N_{s,i}} \sum_{j=1}^{N_{s,i}} \frac{1}{\pi_{\rm PE}(\theta_{i,j})} \frac{dN}{dtd\theta}(\Lambda)\bigg|_{i,j} \tag{A.19}$$

$$= \frac{T_{\rm obs}}{N_{s,i}} \sum_{j=1}^{N_{s,i}} w_{i,j} \tag{A.20}$$

where $w_{i,j} = \frac{1}{\pi_{\rm PE}(\theta_{i,j})} \frac{dN(\Lambda)}{dtd\theta}\big|_{i,j}$. In order to guarantee numerical stability during the computation, the *effective number of samples* is defined as follows:

$$N_{{\rm eff},i} = \frac{\sum_{j=1}^{N_{s,i}} (w_{i,j})^2}{\sum_{j=1}^{N_{s,i}} w_{i,j}^2} \geq 20 \tag{A.21}$$

if that condition is not satisfied, the model cannot be trusted because it would imply that the integral is computed with less than 20 points.

The Monte Carlo integration over the injections samples in Eq. 2.7, can be written as:

$$N_{\rm exp} = \frac{T_{\rm obs}}{N_{\rm gen}} \sum_{j=1}^{N} \frac{1}{\pi_{\rm inj}(\theta_j)} \frac{dN}{dtd\theta}\bigg|_j \tag{A.22}$$

$$= \frac{T_{\rm obs}}{N_{\rm gen}} \sum_{j=1}^{N} s_j \tag{A.23}$$

where $s_j = \frac{1}{\pi_{\rm inj}(\theta_j)} \frac{dN}{dtd\theta}\Big|_j$, $N_{\rm gen}$ is the total number of generated injection samples and $N_{\rm det}$ is the number of detected injection samples. Once again, to guarantee numerical stability the *effective number of injections* is defined and constrained as it follows:

$$N_{{\rm eff},i} = \frac{\left[\sum_j^{\rm N_{det}} s_j\right]^2}{\sum_j^{\rm N_{det}} s_j^2 - N_{\rm gen}^{-1}\left(\sum_j^{\rm N_{det}} s_j\right)^2} \geq 4N_{\rm obs} \tag{A.24}$$

A.3 PE posterior samples and injection samples from GWTC-3

PE posterior samples and injection samples are essential to perform hierarchical bayesian inference, as described in Sec. 2.1. We chose to work with posterior and injections samples from the GWTC-3 catalog [44] data release on Zenodo [45, 46, 47].
GWTC-3 PE posterior samples furnish a set of posterior draws for every GW event (for a total number of 69 events, each with 1286 samples), as described by Eq. 2.3. The common choice for the PE posterior samples prior is the following:

$$\pi_{\rm PE}(\theta|\Lambda) \propto d_L^2 \tag{A.25}$$

and it derives from the small redshift approximation in which the volume element scales as $dV \propto d_L^2 dd_L$ [48]. The PE posterior set is obtained by importance reweighting every event by the astrophysical rate:

$$w_{i,j} \propto \frac{1}{\pi_{\rm PE}(\theta_{i,j})} \frac{dN}{dtd\theta}(\Lambda)\Big|_{\theta_{i,j}} \tag{A.26}$$

On the other hand 81117 GWTC-3 injection samples are Monte Carlo drawn from the injection distribution $\pi_{\rm inj}(\theta)$, described in Eq. 2.7. They are chosen in such a way to respect the following condition:

$$\text{IFAR}_{max} \geq 1 \text{ yr} \tag{A.27}$$

IFAR (Inverse False Alarm Rate) quantifies the expected time interval between false alarms (caused by noise trigger) with the same significance of the considered event. So an IFAR of 1 yr means that a background trigger of equal or higher significance as the considered event is expected once per year. Then the detected injections are the only signals that are equal or exceed that threshold. This choice matches the GWTC-3 catalog: a candidate event is inserted in the catalog only if its IFAR $\geq$ 1yr [44]. Each surviving injection is importance reweighted by the astrophysical rate:

$$s_j \propto \frac{1}{\pi_{\rm inj}(\theta_j)} \frac{dN}{dtd\theta}(\Lambda)\Big|_{\theta_j} \tag{A.28}$$

A.4 Normalizing Flows python class

`one_catalog_wrapper` zuko class:

```
class one_catalog_wrapper(object):
    def __init__(self, dicto_pop, n_layers, hidden_features, lr,
                 batch_size, n_iter, patience, plot, seed, device="cpu"):

        self.device = torch.device(device)
        torch.manual_seed(seed)
        random.seed(seed)
        np.random.seed(seed)
        raw_pop = np.vstack([dicto_pop['mass1'].flatten(),
                             dicto_pop['mass2'].flatten(),
                             dicto_pop['redshift'].flatten()]).T
        self.mean = raw_pop.mean(axis=0)
        self.std = raw_pop.std(axis=0)
        self.norm_pop = (raw_pop - self.mean) / self.std
        self.train_data, self.vali_data = train_test_split(self.norm_pop,
                                                           test_size=0.2,
                                                           random_state=seed)
        self.plot = plot
        self.flow = self.build_flow(n_layers=n_layers,
                                    hidden_features=hidden_features).to(self.device)
        self.train_flow(self.flow,
                        self.train_data,
                        self.vali_data,
                        n_iter=n_iter,
                        lr=lr,
                        batch_size=batch_size,
                        patience=patience)

    def build_flow(self, n_layers, hidden_features):
        return NSF(features=3,
                   context=0,
                   transforms=n_layers,
                   hidden_features=[hidden_features])

    def train_flow(self, flow, train_data, vali_data, n_iter, lr, batch_size, patience):
        optimizer = torch.optim.Adam(flow.parameters(),
                                     lr=lr)
        scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(optimizer,
                                                               mode='min',
                                                               factor=0.5,
                                                               patience=10)
        flow.train()
        max_vali_loss = 1e10
        min_vali_model = None
        p = 0
        train_losses, vali_losses = [], []
        for i in range(n_iter):
            idx = np.random.choice(len(train_data),
                                   size=batch_size,
                                   replace=False)
```

```
51              x_batch = torch.tensor(train_data[idx], dtype=torch.float32).to(self.device)
52              optimizer.zero_grad()
53              loss = -flow().log_prob(x_batch).mean()
54              loss.backward()
55              optimizer.step()
56              with torch.no_grad():
57                  vali_tensor = torch.tensor(vali_data, dtype=torch.float32).to(self.device)
58                  vali_loss = -flow().log_prob(vali_tensor).mean().item()
59              train_losses.append(loss.item())
60              vali_losses.append(vali_loss)
61              scheduler.step(vali_loss)
62              if vali_loss < max_vali_loss:
63                  max_vali_loss = vali_loss
64                  min_vali_model = copy.deepcopy(flow.state_dict())
65                  p = 0
66              else:
67                  p += 1
68              if p >= patience:
69                  break
70          flow.load_state_dict(min_vali_model)
71          flow.eval()
72           with torch.no_grad():
73              train_tensor = torch.tensor(train_data, dtype=torch.float32).to("cpu")
74              best_train_loss = -flow().log_prob(train_tensor).mean().item()
75              vali_tensor = torch.tensor(vali_data, dtype=torch.float32).to("cpu")
76              best_vali_loss = -flow().log_prob(vali_tensor).mean().item()
77              print(f"best train loss: {best_train_loss:.4f}")
78              print(f"best validation loss: {best_vali_loss:.4f}")
79              print(f"|train - validation|: {np.abs(best_train_loss - best_vali_loss):.4f}")
80          if self.plot:
81              plt.figure(figsize=(8, 4))
82              plt.plot(train_losses, label='Training loss')
83              plt.plot(vali_losses, label='Validation loss')
84              plt.title("Train and validation losses")
85              plt.xlabel("Iteration")
86              plt.ylabel("Negative Log Likelihood (NLL)")
87              plt.legend()
88              plt.grid()
89              plt.tight_layout()
90              plt.show()
91
92      def pdf(self, mass_1, mass_2, z):
93          or_shape = mass_1.shape
94          data = np.vstack([mass_1.flatten(),
95                            mass_2.flatten(),
96                            z.flatten()]).T
97          norm_data = (data - self.mean) / self.std
98          data_tensor = torch.tensor(norm_data, dtype=torch.float32).to(self.device)
99          with torch.no_grad():
100             log_probs = self.flow().log_prob(data_tensor).cpu().numpy()
101         log_det_jacobian = -np.log(self.std).sum()
102         log_probs += log_det_jacobian
103         probs = np.exp(log_probs)
104         mask = ((mass_1 <= 0) | (mass_2 <= 0) | (z <= 0)).flatten()
```

```
105             probs[mask] = 0.
106             return np.reshape(probs, or_shape)
107
108         def log_pdf(self, mass_1, mass_2, z):
109             return np.log(self.pdf(mass_1, mass_2, z))
110
111         def sample(self, Nsamples, mmin, mmax, zmax):
112             m1_samples = np.random.uniform(mmin, mmax, size=Nsamples*100)
113             m2_samples = np.random.uniform(mmin, mmax, size=Nsamples*100)
114             z_samples = np.random.uniform(0, zmax, size=Nsamples*100)
115             w = self.pdf(m1_samples, m2_samples, z_samples)
116             idx = np.random.choice(len(w), size=Nsamples, replace=True, p=w/w.sum())
117             return m1_samples[idx], m2_samples[idx], z_samples[idx]
```

`one_catalog_wrapper` nflows class:

```
class one_catalog_wrapper(object):
    def __init__(self, dicto_pop,
                 n_layers,
                 hidden_features,
                 lr,
                 batch_size,
                 n_iter=1000,
                 patience=40,
                 seed=42,
                 plot=True,
                 device="cpu"):
        self.device = torch.device(device)
        self.plot = plot
        self.seed = seed
        self.n_iter = n_iter
        self.patience = patience

        torch.manual_seed(seed)
        random.seed(seed)
        np.random.seed(seed)
        raw_pop = np.vstack([dicto_pop['mass1'].flatten(),
                             dicto_pop['mass2'].flatten(),
                             dicto_pop['redshift'].flatten()]).T
        self.mean = raw_pop.mean(axis=0)
        self.std = raw_pop.std(axis=0)
        self.norm_pop = (raw_pop - self.mean) / self.std
        self.train_data, self.vali_data = train_test_split(self.norm_pop,
                                                           test_size=0.2,
                                                           random_state=seed)
        self.flow = self.build_flow(n_layers,
                                    hidden_features).to(self.device)
        self.train_flow(self.train_data,
                        self.vali_data,
                        n_iter=self.n_iter,
                        lr=lr,
                        batch_size=batch_size,
                        patience=self.patience)

    def build_flow(self, n_layers, hidden_features):
        transforms = []
        for _ in range(n_layers):
            transforms.append(ReversePermutation(features=3))
            transforms.append(MaskedAffineAutoregressiveTransform(features=3,
                                                  hidden_features=hidden_features))
        transform = CompositeTransform(transforms)
        base_distribution = StandardNormal([3])
        return Flow(transform, base_distribution)

    def train_flow(self, train_data,
                   vali_data,
                   n_iter,
                   lr,
```

```
                batch_size,
                patience):
        optimizer = torch.optim.Adam(self.flow.parameters(),
                                     lr=lr)
        self.flow.train()
        max_vali_loss = 1e10
        min_vali_model = None
        p = 0
        train_losses, vali_losses = [], []
        for i in range(n_iter):
            idx = np.random.choice(len(train_data),
                                   size=batch_size,
                                   replace=False)
            x_batch = torch.tensor(train_data[idx],
                                   dtype=torch.float32).to(self.device)
            optimizer.zero_grad()
            loss = -self.flow.log_prob(x_batch).mean()
            loss.backward()
            optimizer.step()
            with torch.no_grad():
                vali_tensor = torch.tensor(vali_data,
                                           dtype=torch.float32).to(self.device)
                vali_loss = -self.flow.log_prob(vali_tensor).mean().item()
            train_losses.append(loss.item())
            vali_losses.append(vali_loss)
            if vali_loss < max_vali_loss:
                max_vali_loss = vali_loss
                min_vali_model = copy.deepcopy(self.flow.state_dict())
                p = 0
            else:
                p += 1
            if p >= patience:
                break
        self.flow.load_state_dict(min_vali_model)
        self.flow.eval()
        with torch.no_grad():
            train_tensor = torch.tensor(train_data, dtype=torch.float32).to("cpu")
            best_train_loss = -self.flow.log_prob(train_tensor).mean().item()
            vali_tensor = torch.tensor(vali_data, dtype=torch.float32).to("cpu")
            best_vali_loss = -self.flow.log_prob(vali_tensor).mean().item()
            print(f"best training loss: {best_train_loss:.4f}")
            print(f"best validation loss: {best_vali_loss:.4f}")
            print(f"|train - validation|: {np.abs(best_train_loss - best_vali_loss):.4f}")
        if self.plot:
            plt.figure(figsize=(8, 4))
            plt.plot(train_losses, label='Train loss')
            plt.plot(vali_losses, label='Validation loss')
            plt.xlabel('Iteration')
            plt.ylabel('Negative Log Likelihood (NLL)')
            plt.title('Train and validation losses')
            plt.legend()
            plt.grid()
            plt.tight_layout()
            plt.show()
```

```
107
108     def pdf(self, mass_1, mass_2, z):
109         or_shape = mass_1.shape
110         data = np.vstack([mass_1.flatten(),
111                           mass_2.flatten(),
112                           z.flatten()]).T
113         norm_data = (data - self.mean) / self.std
114         data_tensor = torch.tensor(norm_data,
115                                    dtype=torch.float32).to(self.device)
116         with torch.no_grad():
117             log_probs = self.flow.log_prob(data_tensor).cpu().numpy()
118         log_det_jacobian = -np.sum(np.log(self.std))
119         log_probs += log_det_jacobian
120         probs = np.exp(log_probs)
121         mask = ((mass_1 <= 0) | (mass_2 <= 0) | (z <= 0)).flatten()
122         probs[mask] = 0.
123         return np.reshape(probs, or_shape)
124
125     def log_pdf(self, mass_1, mass_2, z):
126         return np.log(self.pdf(mass_1, mass_2, z))
127
128     def sample(self, Nsamples,
129                mmin,
130                mmax,
131                zmax):
132         m1_samples = np.random.uniform(mmin, mmax, size=Nsamples * 100)
133         m2_samples = np.random.uniform(mmin, mmax, size=Nsamples * 100)
134         z_samples = np.random.uniform(0, zmax, size=Nsamples * 100)
135         w = self.pdf(m1_samples, m2_samples, z_samples)
136         idx = np.random.choice(len(w),
137                                size=Nsamples,
138                                replace=True,
139                                p=w / w.sum())
140         return m1_samples[idx], m2_samples[idx], z_samples[idx]
141
142
```

`two_catalogs_wrapper`

```
1
2  class two_catalogs_wrapper(object):
3      def __init__(self, one_catalog_wrapper_pop1, one_catalog_wrapper_pop2):
4          self.population_parameters = ['mixture']
5          self.model1 = one_catalog_wrapper_pop1
6          self.model2 = one_catalog_wrapper_pop2
7
8      def update(self, **kwargs):
9          self.mixture = kwargs['mixture']
10
11     def pdf(self, mass_1, mass_2, z):
12         or_shape = mass_1.shape
13         p1 = self.model1.pdf(mass_1, mass_2, z).flatten()
14         p2 = self.model2.pdf(mass_1, mass_2, z).flatten()
15         probs = (1 - self.mixture) * p1 + self.mixture * p2
16         mask = ((mass_1 <= 0) | (mass_2 <= 0) | (z <= 0)).flatten()
17         probs[mask] = 0.
18         return np.reshape(probs, or_shape)
19
20     def log_pdf(self, mass_1, mass_2, z):
21         return np.log(self.pdf(mass_1, mass_2, z))
22
23     def sample(self, Nsamples, mmin, mmax, zmax):
24         mass_1_samples = np.random.uniform(mmin, mmax, size=Nsamples * 100)
25         mass_2_samples = np.random.uniform(mmin, mmax, size=Nsamples * 100)
26         z_samples = np.random.uniform(0, zmax, size=Nsamples * 100)
27         w = self.pdf(mass_1_samples, mass_2_samples,z_samples)
28         idx = np.random.choice(len(z_samples), size=Nsamples, replace=True, p=w/w.sum())
29         return mass_1_samples[idx], mass_2_samples[idx], z_samples[idx]
```

Bibliography

[1] B.P. Abbott et al. "GW150914: First results from the search for binary black hole coalescence with Advanced LIGO". In: *Physical Review D* 93.12 (June 2016). ISSN: 2470-0029. DOI: 10.1103/physrevd.93.122003. URL: http://dx.doi.org/10.1103/PhysRevD.93.122003.

[2] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration. *LIGO, VIRGO AND KAGRA Observing Run Plans*. URL: https://observing.docs.ligo.org/plan/.

[3] LVK Collaboration. *GWTC-4: Gravitational-Wave Transient Catalog*. URL: https://gwosc.org/O4/O4a/.

[4] The LIGO Scientific Collaboration, the Virgo Collaboration, and the KAGRA Collaboration. *GWTC-4.0: Population Properties of Merging Compact Binaries*. 2025. arXiv: 2508.18083 [astro-ph.HE]. URL: https://arxiv.org/abs/2508.18083.

[5] T. A. Callister. *Observed Gravitational-Wave Populations*. 2024. arXiv: 2410.19145 [astro-ph.HE]. URL: https://arxiv.org/abs/2410.19145.

[6] Sean M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. Cambridge University Press, 2019.

[7] Grégoire Pierra and Simone Mastrogiovanni. *Gravitational wave cosmology : an introduction*. 2025. arXiv: 2507.10597 [gr-qc]. URL: https://arxiv.org/abs/2507.10597.

[8] N. Aghanim et al. "Planck2018 results: VI. Cosmological parameters". In: *Astronomy & Astrophysics* 641 (Sept. 2020), A6. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201833910. URL: http://dx.doi.org/10.1051/0004-6361/201833910.

[9] Adam G. Riess et al. "A Comprehensive Measurement of the Local Value of the Hubble Constant with 1 km s^{-1} Mpc^{-1} Uncertainty from the Hubble Space Telescope and the SH0ES Team". In: *The Astrophysical Journal Letters* 934.1 (July 2022), p. L7. DOI: 10.3847/2041-8213/ac5c5b. URL: https://doi.org/10.3847/2041-8213/ac5c5b.

[10] Leandros Perivolaropoulos and Foteini Skara. "Hubble tension or a transition of the Cepheid SnIa calibrator parameters?" In: *Phys. Rev. D* 104 (12 Dec. 2021), p. 123511. DOI: 10.1103/PhysRevD.104.123511. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.123511.

[11] Valeria Ferrari, Leonardo Gualtieri, and Paolo Pani. *General Relativity and its Applications: Black Holes, Compact Stars and Gravitational Waves*. 1st. CRC Press, 2020. DOI: 10.1201/9780429491405.

[12] Rachel Gray et al. "Joint cosmological and gravitational-wave population inference using dark sirens and galaxy catalogues". In: *Journal of Cosmology and Astroparticle Physics* 2023.12 (Dec. 2023), p. 023. DOI: 10.1088/1475-7516/2023/12/023. URL: https://dx.doi.org/10.1088/1475-7516/2023/12/023.

[13] R. Abbott et al. "Constraints on the Cosmic Expansion History from GWTC–3". In: *The Astrophysical Journal* 949.2 (June 2023), p. 76. DOI: 10.3847/1538-4357/ac74bb. URL: https://dx.doi.org/10.3847/1538-4357/ac74bb.

[14] R. Abbott et al. "Population Properties of Compact Objects from the Second LIGO–Virgo Gravitational-Wave Transient Catalog". In: *The Astrophysical Journal Letters* 913.1 (May 2021), p. L7. DOI: 10.3847/2041-8213/abe949. URL: https://dx.doi.org/10.3847/2041-8213/abe949.

[15] Simone Mastrogiovanni et al. "ICAROGW: A python package for inference of astrophysical population properties of noisy, heterogeneous, and incomplete observations". In: *A&A* 682 (2024), A167. DOI: 10.1051/0004-6361/202347007. URL: https://doi.org/10.1051/0004-6361/202347007.

[16] R. Abbott et al. "Population of Merging Compact Binaries Inferred Using Gravitational Waves through GWTC-3". In: *Phys. Rev. X* 13 (1 Mar. 2023), p. 011048. DOI: 10.1103/PhysRevX.13.011048. URL: https://link.aps.org/doi/10.1103/PhysRevX.13.011048.

[17] The LIGO Scientific Collaboration, the Virgo Collaboration, and the KAGRA Collaboration. *GWTC-4.0: Constraints on the Cosmic Expansion Rate and Modified Gravitational-wave Propagation*. 2025. arXiv: 2509.04348 [astro-ph.CO]. URL: https://arxiv.org/abs/2509.04348.

[18] Michela Mapelli. "Formation Channels of Single and Binary Stellar-Mass Black Holes". In: 2021. DOI: 10.1007/978-981-15-4702-7_16-1. arXiv: 2106.00699 [astro-ph.HE].

[19] Djuna Croon and Jeremy Sakstein. *Prediction of Multiple Features in the Black Hole Mass Function due to Pulsational Pair-Instability Supernovae*. 2023. arXiv: 2312.13459 [astro-ph.HE]. URL: https://arxiv.org/abs/2312.13459.

[20] Grégoire Pierra et al. "Study of systematics on the cosmological inference of the Hubble constant from gravitational wave standard sirens". In: *Phys. Rev. D* 109.8 (2024), p. 083504. DOI: 10.1103/PhysRevD.109.083504. arXiv: 2312.11627 [astro-ph.CO].

[21] Simon F. Portegies Zwart, Stephen L.W. McMillan, and Mark Gieles. "Young Massive Star Clusters". In: *Annual Review of Astronomy and Astrophysics* 48.1 (Aug. 2010), pp. 431–493. ISSN: 1545-4282. DOI: 10.1146/annurev-astro-081309-130834. URL: http://dx.doi.org/10.1146/annurev-astro-081309-130834.

[22] Harold Jeffreys. "An invariant form for the prior probability in estimation problems". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 186 (1946), pp. 453–461. DOI: 10.1098/rspa.1946.0056. URL: http://doi.org/10.1098/rspa.1946.0056.

[23] S Mastrogiovanni et al. "Inferring binary black holes stellar progenitors with gravitational wave sources". In: *Monthly Notices of the Royal Astronomical Society* 517.3 (Oct. 2022), pp. 3432–3444. ISSN: 0035-8711. DOI: 10.1093/mnras/stac2850. eprint: https://academic.oup.com/mnras/article-pdf/517/3/3432/46649718/stac2850.pdf. URL: https://doi.org/10.1093/mnras/stac2850.

[24] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2021), pp. 3964–3979. DOI: 10.1109/TPAMI.2020.2992934.

[25] George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked Autoregressive Flow for Density Estimation". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf.

[26] Conor Durkan et al. "Neural spline flows". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[27] *nflows Documentation*. https://github.com/bayesiains/nflows.

[28] *Zuko Documentation*. https://zuko.readthedocs.io/stable/tutorials/basics.html.

[29] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. ISBN: 9780387310732. URL: https://books.google.it/books?id=kTNoQgAACAAJ.

[30] 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.

[31] Matias Valdenegro-Toro and Matthia Sabatelli. *Machine Learning Students Overfit to Overfitting*. 2022. arXiv: 2209.03032 [cs.LG]. URL: https://arxiv.org/abs/2209.03032.

[32] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2019. arXiv: 1803.08375 [cs.NE]. URL: https://arxiv.org/abs/1803.08375.

[33] Instituto Superior Técnico. *Lecture 4: Selection*. http://web.ist.utl.pt. PDF file. Accessed 14 July 2015. Nov. 2002.

[34] Oskar M. Essenwanger. *Elements of Statistical Analysis*. Elsevier, 1986. ISBN: 0444424261.

[35] Piero Madau and Mark Dickinson. "Cosmic Star-Formation History". In: *Annual Review of Astronomy and Astrophysics* 52.1 (Aug. 2014), pp. 415–486. ISSN: 1545-4282. DOI: 10.1146/annurev-astro-081811-125615. URL: http://dx.doi.org/10.1146/annurev-astro-081811-125615.

[36] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall, 1986.

[37] Gregory Ashton and othters. "Bilby: A User-friendly Bayesian Inference Library for Gravitational-wave Astronomy". In: *The Astrophysical Journal Supplement Series* 241.2 (Apr. 2019), p. 27. DOI: 10.3847/1538-4365/ab06fc. URL: https://dx.doi.org/10.3847/1538-4365/ab06fc.

[38] Joshua S. Speagle. *A Conceptual Introduction to Markov Chain Monte Carlo Methods*. 2020. arXiv: 1909.12313 [stat.OT]. URL: https://arxiv.org/abs/1909.12313.

[39] P. A. R. Ade et al. "Planck2015 results: XIII. Cosmological parameters". In: *Astronomy & Astrophysics* 594 (Sept. 2016), A13. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201525830. URL: http://dx.doi.org/10.1051/0004-6361/201525830.

[40] Manuel Arca Sedda et al. "Isolated and dynamical black hole mergers with B-POP: the role of star formation and dynamics, star cluster evolution, natal kicks, mass and spins, and hierarchical mergers". In: *Monthly Notices of the Royal Astronomical Society* 520.4 (Jan. 2023), pp. 5259–5282. ISSN: 0035-8711. DOI: 10.1093/mnras/stad331. eprint: https://academic.oup.com/mnras/article-pdf/520/4/5259/49330065/stad331.pdf. URL: https://doi.org/10.1093/mnras/stad331.

[41] S. et al. Ferraiuolo. "Inferring astrophysics and cosmology with individual compact binary coalescences and their gravitational-wave stochastic background". In: *Astronomy & Astrophysics* 701 (Sept. 2025). Published online 02 September 2025, Section "Cosmology (including clusters of galaxies)", A36. DOI: 10.1051/0004-6361/202555124.

[42] S. Mastrogiovanni et al. "On the importance of source population models for gravitational-wave cosmology". In: *Phys. Rev. D* 104 (6 Sept. 2021), p. 062009. DOI: 10.1103/PhysRevD.104.062009. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.062009.

[43] Maya Fishbach, Will M. Farr, and Daniel E. Holz. "The Most Massive Binary Black Hole Detections and the Identification of Population Outliers". In: *The Astrophysical Journal Letters* 891.2 (Mar. 2020), p. L31. DOI: 10.3847/2041-8213/ab77c9.

[44] R. Abbott et al. "GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run". In: *Phys. Rev. X* 13 (4 Dec. 2023), p. 041039. DOI: 10.1103/PhysRevX.13.041039. URL: https://link.aps.org/doi/10.1103/PhysRevX.13.041039.

[45] LIGO Scientific Collaboration and 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*. 2021. DOI: 10.5281/zenodo.5546663. URL: https://doi.org/10.5281/zenodo.5546663.

[46] LIGO Scientific Collaboration and Virgo Collaboration and KAGRA Collaboration. *O3 search sensitivity estimates*. 2021. DOI: 10.5281/zenodo.5546676. URL: https://doi.org/10.5281/zenodo.5546676.

[47] LVK Collaboration. *GWTC-3: Gravitational-Wave Transient Catalog*. URL: https://gwosc.org/GWTC-3/.

[48] D. W. Hogg. "Distance measures in cosmology". In: *arXiv e-prints* (1999). arXiv: astro-ph/9905116 [astro-ph].