Gravitational wave populations and cosmology with neural posterior estimation

Konstantin Leyde,[1,2,*] Stephen R. Green,[3,†] Alexandre Toubiana,[4,‡] and Jonathan Gair[4,§]

[1]*Université Paris Cité, CNRS, Astroparticule et Cosmologie, F-75013 Paris, France*
[2]*Institute of Cosmology and Gravitation, University of Portsmouth, Burnaby Road, Portsmouth PO1 3FX, United Kingdom*
[3]*School of Mathematical Sciences, University of Nottingham University Park, Nottingham NG7 2RD, United Kingdom*
[4]*Max Planck Institute for Gravitational Physics (Albert Einstein Institute) Am Mühlenberg 1, 14476 Potsdam, Germany*

(Received 6 January 2024; accepted 21 February 2024; published 18 March 2024)

We apply neural posterior estimation for fast-and-accurate hierarchical Bayesian inference of gravitational wave populations. We use a normalizing flow to estimate directly the population hyperparameters from a collection of individual source observations. This approach provides complete freedom in event representation, automatic inclusion of selection effects, and (in contrast to likelihood estimation) without the need for stochastic samplers to obtain posterior samples. Since the number of events may be unknown when the network is trained, we split into subpopulation analyses that we later recombine; this allows for fast sequential analyses as additional events are observed. We demonstrate our method on a toy problem of dark siren cosmology, and show that inference takes just a few minutes and scales to ∼600 events before performance degrades. We argue that neural posterior estimation therefore represents a promising avenue for population inference with large numbers of events.

DOI: 10.1103/PhysRevD.109.064056

I. INTRODUCTION

Hierarchical Bayesian analysis (HBA) provides the statistical framework to combine individual gravitational wave (GW) observations to answer questions about entire populations. Starting from a population model $p_{\rm pop}(\theta|\Lambda)$ for source parameters θ depending on population *hyperparameters* Λ, with prior $p(\Lambda)$, HBA characterizes the population in terms of the posterior distribution $p(\Lambda|\mathcal{D}_C)$, where $\mathcal{D}_C$ is a catalog of GW observations. With over 100 observations by the LIGO-Virgo-KAGRA Collaboration [1–7] to date [8], HBA has been used to constrain a wide variety of population properties including mass and spin distributions [9–33], and fundamental physics [34–47].

When combined with redshift information, GWs can also be used to constrain cosmology. Indeed, the joint GW and electromagnetic observation of GW170817—a standard *siren*—constrained the Hubble constant H_0 to within ∼20% [48,49]. However, the vast majority of observations are of binary black holes, with no electromagnetic counterpart. In these cases, statistical *dark siren* methods using HBA can nevertheless still place constraints on cosmology. This can be done by either correlating GW signals with galaxy catalogs [45,50–56] or by involving assumptions on the source binary mass distribution [39,40,57–61].

Here, we focus on the mass spectrum method: given a population model in the source frame, the predicted distribution of detector frame masses[1] and luminosity distance depends on the population and the cosmological parameters. By comparing this predicted distribution to the one observed with gravitational waves (GWs), we can therefore jointly constrain population and cosmological parameters.

However, the current uncertainty on H_0 from GW observations is much larger than from studies of the cosmic microwave background [62] or supernovae [63] and it will not be before $\mathcal{O}(10^4)$ binary black hole mergers [39,59–61], or several hundred binary neutron stars [64] that their uncertainty will be comparable. Networks of future detectors, such as the Einstein telescope (ET) and Cosmic Explorer (CE), will provide the requisite large number of observed events, reaching far into the cosmic past. This will

*konstantin.leyde@port.ac.uk
†stephen.green2@nottingham.ac.uk
‡alexandre.toubiana@aei.mpg.de
§jonathan.gair@aei.mpg.de

[1]Recall the relation between detector frame masses m_d and source frame masses m_s are related as $m_d = (1+z)m_s$. Throughout, we assume the contribution from proper motion to be negligible against the cosmological redshift.

allow for the precise inference of cosmological parameters, using bright sirens [65], in conjunction with galaxy catalogs [66–71] and features in the mass spectrum [57,58,71–73]. Conventional population analyses (hierarchical Bayesian inference methods) require an analytic population model, and are slow when analyzing a large number of events. The large number of events of the upcoming detector networks calls for new methods for the measurement of the hyperparameters (e.g., H_0) with GW events.[2]

In this work, we apply neural posterior estimation to population inference of GW signals. The specific illustrative problem we set out to solve is to obtain constraints on cosmological parameters through the dark siren mass spectrum method,[3] addressing the aforementioned issues. In addition to the gain in computational speed,[4] simulation-based approaches can, in principle, directly incorporate predictions from astrophysical simulations, without having to resort to phenomenological descriptions of the resulting source parameter distributions. We summarize the analyzed GW data by posterior samples of the parameters of the individual events,[5] but the method could be applied to any input data that summarizes the GW observations sufficiently well. It is therefore particularly adapted to future analysis chains that rely on other deep learning algorithms. In principle, our method can also account for additional uncertainty from latent variables, which are difficult to account for in conventional methods or when modeling the population likelihood. For example, this could include the use of different waveforms for the production of single-event posterior samples.

The learning task is to approximate the posterior $p(\Lambda|\mathcal{D}_C)$, where Λ is the set of hyperparameters describing the population model and the cosmological parameters, and $\mathcal{D}_C$ is the GW catalog data. We propose a deep neural network scheme that learns directly the posterior distribution of the population parameters—including the selection effect. In particular, this approach allows us to infer population properties in a likelihood-free way (also referred to as *simulation-based inference*), requiring solely the simulation of observed event data.

A number of previous studies have applied machine learning techniques to aspects of the population inference problem. In [74], machine learning was used to estimate the selection function, while [75–78] used machine learning to represent the *population likelihood* (including selection effects in the latter two cases). By contrast, in our approach we directly model the *population posterior distribution*, which circumvents the need for an additional MCMC analysis to obtain the hyperparameter posterior since posterior samples are produced directly through importance sampling. Additionally, [76] learned the population likelihood (in the bright siren case—assuming an EM counterpart), but used a toy model for the single-event posterior distribution, whereas our method uses posterior samples generated with the realistic deep learning model `dingo` [79]. It has been shown that this model agrees very well with the true posterior distribution in the parameter ranges we consider.

The network's architecture used here is that of a *conditional normalizing flow* [80–84]. This framework allows one to generate a distribution conditioned on data, and to draw samples from the distribution efficiently. This method has been applied to a large variety of problems in science [75,79,85–91]. In particular, it has accelerated single-event parameter estimation for compact binary coalescences by several orders of magnitude, see `dingo` [79,89–91]. Whereas the latter model outputs posterior samples of individual event parameters given the estimated noise spectral density and measured strain for that event, the model described in this work outputs the distribution of the hyperparameters given posterior samples of the individual events.

The structure of this paper is as follows. In Sec. II A, we begin by revisiting the classical approach to population inference utilizing Bayesian statistics. Following this, in Sec. II B 1, we outline our divide-and-conquer strategy, which splits the population into smaller subpopulations for independent analysis, subsequently merging them to obtain the final result analysing the complete catalog. In Sec. III, we then provide an overview of the astrophysical assumptions that underlie our study. The training dataset, along with its number of entries, is then presented in detail in Sec. IV. From these training datasets, we train our models and present the results, which are described in Sec. V, accompanied by a comparison against the traditional Bayesian approach. Finally, in Sec. VI, we discuss our results and possible extensions to our work.

II. METHODS

We now outline the conventional hierarchical Bayesian population analysis and relate it to the deep neural network approach in Sec. II B. The classical approach will function as our reference point against which we will compare the outcomes with the normalizing flow (NF) method. We refer to the classical method as HBA (hierarchical Bayesian analysis) and to the neural network model as neural posterior estimation (NPE).

[2]Since the hyperparameters describe the overall distribution of source parameters rather than the single-event ones, the extraction of the hyperparameters is also referred to as *hierarchical inference*.

[3]We note, however, that the proposed method of population analysis with deep neural networks is not limited to this application. In principle, our scheme could use electromagnetic, or GW data or both to produce constraints on the cosmological or population parameters.

[4]Note that just-in-time compilation, the use of GPUs and gradient-based sampling algorithms can achieve similar speeds.

[5]In the following, we refer to these as single-event posterior samples.

TABLE I. Overview of the variables and quantities used.

Variable	Description
	GW data
θ	Single-event BBH parameters
$p(\hat{\theta}\|\mathcal{D})$	Single-event posterior distribution
	GW catalog
$\mathcal{D}_C$	A catalog C of GW observations
$N_{\rm obs}$	Number of observed GW events
	Population parameters
Λ	Hyperparameters
$p_{\rm pop}(\theta\|\Lambda)$	Population model
$p(\Lambda\|\mathcal{D}_C)$	Hyperparameter posterior from a catalog C, cf. Sec. (2)
$\xi(\Lambda)$	Selection bias
	Machine learning
$n_{\rm sub}$	Number of events per subpopulation
$q(\Lambda\|\mathcal{D}_C)$	Hyperparameters posterior estimate from a GW catalog C

To facilitate the following discussion, we introduce some notation (see also Table I for a summary of the variables used). We denote the set of hyperparameters as Λ—this can include cosmological parameters such as H_0 and the parameters describing the mass, spin and redshift distribution of individual events. The true source parameters are written as θ, and the distribution of data, $\mathcal{D}$, given the true parameters as $p(\mathcal{D}|\theta)$. The latter term is the single-event GW likelihood. The population model is denoted as $p_{\rm pop}(\theta|\Lambda)$, and we use K to denote a collection of events. For instance, $\theta_K \coloneqq \{\theta_i\}_{i\in K}$ is the set of true parameters of events in the set K. In this notation, the probability of drawing the true parameters θ_K from the population model is then

$$p_{\rm pop}(\theta_K|\Lambda) = \prod_{j\in K} p_{\rm pop}(\theta_j|\Lambda), \tag{1}$$

since individual sample draws are independent. The number of events in the GW catalog is denoted as $N_{\rm obs} \coloneqq \mathrm{card}(C)$, with $\mathrm{card}(X)$ the number of elements in the set X.

A. Hierarchical Bayesian population method

The goal of extracting population and cosmological parameters from GW data is classically approached with a hierarchical Bayesian analysis (HBA) [10,12,16,59,60,92–97]. We wish to infer the posterior distribution of Λ, based on a set of GW events $\mathcal{D}_C \coloneqq \{\mathcal{D}_i\}_{i\in C}$. With the catalog $\mathcal{D}_C$ the posterior of Λ can be rewritten with Bayes's theorem as $p(\Lambda|\mathcal{D}_C) = p(\Lambda)p(\mathcal{D}_C|\Lambda)/p(\mathcal{D}_C)$, where $p(\Lambda)$ denotes the prior knowledge of Λ, $p(\mathcal{D}_C|\Lambda)$ is the hierarchical likelihood, and $p(\mathcal{D}_C)$ is the evidence, the probability of observing data $\mathcal{D}_C$.

Using then the HBA scheme, the posterior of the hyperparameters informed from $N_{\rm obs}$ events is given by (marginalizing over the overall rate of events) [98–100]

$$\begin{aligned} p(\Lambda|\mathcal{D}_C) &= \frac{p(\Lambda)}{p(\mathcal{D}_C)}\, p(\mathcal{D}_C|\Lambda) = \frac{p(\Lambda)}{p(\mathcal{D}_C)} \prod_{j=1}^{N_{\rm obs}} p(\mathcal{D}_j|\Lambda) \\ &= \frac{p(\Lambda)}{p(\mathcal{D}_C)} \prod_{j=1}^{N_{\rm obs}} \frac{\int p(\mathcal{D}_j|\theta_j)\, p_{\rm pop}(\theta_j|\Lambda)\mathrm{d}\theta_j}{\int p_{\rm det}(\theta_j)\, p_{\rm pop}(\theta_j|\Lambda)\mathrm{d}\theta_j}, \end{aligned} \tag{2}$$

the prior on Λ is denoted as $p(\Lambda)$ and the prior probability of the data as $p(\mathcal{D}_C)$. The uncertainty in our knowledge of single-event parameters is encoded in the likelihood $p(\mathcal{D}_j|\theta_j)$ of obtaining data $\mathcal{D}_j$, given the true parameters θ_j. Finally, the probability of detection, given the source parameters θ, is denoted by $p_{\rm det}(\theta)$ and depends (among other factors) on the detector sensitivity, the number of detectors and the detection threshold. This encodes the fact that not all data is included in the set $\mathcal{D}_C$, but we choose segments of data in which we are confident that signals of astrophysical origin are present. This selection is a property of the data alone. The data is the sum $\mathcal{D} = h(\theta) + n$ of the pure signal $h(\theta)$ and the noise n. The detection probability is the probability that this data lies in the region we define as a detected source, i.e., $p_{\rm det}(\theta) = \int_{\mathcal{D}\,\rm detected} \mathrm{d}n\, p(\mathcal{D}|\theta)$. The denominator (in the product) of the above equation accounts for this *selection effect*—not all GW sources have the same probability of detection. It is common to define the detected fraction of the population $\xi(\Lambda) \coloneqq \int p_{\rm det}(\theta)\, p_{\rm pop}(\theta|\Lambda)\mathrm{d}\theta$. In general, it is difficult to evaluate this term, and one usually relies on an injection campaign to produce a set of detected GW signals. We will show that our method accounts for the selection effect, bypassing the explicit computation of $\xi(\Lambda)$. Effectively, we perform an injection campaign during the generation of the training data and hence, the cost is amortized over the repeated evaluation of the neural population posterior.

There is some freedom in the representation of the GW data $\mathcal{D}_i$: we focus here on posterior samples, that approximate the uncertainty of the source parameters θ (such as the component source frame masses, or the luminosity distance). The posterior samples follow the distribution $\hat{\theta} \sim p(\hat{\theta}|\mathcal{D}_k)$, and we denote the assumed prior under which the posterior samples were created as $\pi_{\rm MCMC}$. In the following, we use $\hat{\theta}_{ik}$ to denote the ith posterior sample from the GW event k [compare to Eq. (9)], and $n_{{\rm post},k}$ is the number of posterior samples for this event. The numerator of Eq. (2) is usually approximated by summing over posterior samples of the individual GW events. The population likelihood as informed by *one* GW event $\mathcal{D}_k$ can then be rewritten as

$$p(\mathcal{D}_k|\Lambda) \approx \frac{p(\mathcal{D}_k)}{\xi(\Lambda) n_{\mathrm{post},k}} \sum_{i=1}^{n_{\mathrm{post},k}} \frac{p_{\mathrm{pop}}(\hat{\theta}_{ik}|\Lambda)}{\pi_{\mathrm{MCMC}}(\hat{\theta}_{ik})}, \tag{3}$$

where the sum above is taken over the posterior samples $\hat{\theta}_{ik} \sim p(\hat{\theta}_{ik}|\mathcal{D}_k)$. To evaluate the full population posterior of Eq. (2), one multiplies the individual contributions of Eq. (3).

B. Neural posterior estimation (NPE) methods

Hierarchical Bayesian analysis becomes increasingly expensive as the number of sources included in the analysis increases, due both to the cost of obtaining the posterior samples for each event, and the cost of combining the events to obtain the population posterior. The use of machine learning approaches is becoming increasingly widespread in the physical sciences, as these often provide a fast and efficient way to complete complex analysis tasks. In a gravitational wave context, `dingo` has been shown to generate posterior distributions nearly indistinguishable from those produced by standard sampling algorithms in a small fraction of the time [91], while residual differences can be efficiently eliminated through importance sampling [101]. We hope to see similar benefits from the application of machine learning methods to population inference. A major complication is that the number of events that will be observed is not typically known *a priori*. Not only does this present the difficulty of generating an arbitrarily large training dataset, but neural networks typically have fixed input dimension. We overcome this problem by implementing a strategy that divides the GW catalog into smaller subpopulations, each containing $\mathcal{O}(10\text{–}100)$ signals. Our model then learns the posterior distribution *analyzing a subpopulation of events*. We combine the intermediate results (the population posterior of each subpopulation) to derive the population posterior of the entire catalog.[6] We will now elaborate on the model loss, how to combine subpopulations of events, the NF's architecture and the generation of the training dataset.

1. Subpopulation analysis

To simplify the problem, we split the GW catalog into smaller subpopulations. Calling one of these subpopulations $\mathcal{D}_K \coloneqq \{\mathcal{D}_k\}_{k\in K}$, the model we propose then approximates the population posterior from analyzing $\mathcal{D}_K$, converging to the term $p(\Lambda|\{\mathcal{D}_k\}_{k\in K})$. One then obtains the complete posterior analyzing all events by combining the individual posteriors of each of the subpopulations. This approach ensures the computational cost to generate the training dataset is not too large.

The catalog C is divided into subpopulations of events, $\{K_i\}$, where each of the K_i contains n_{sub} events.[7] That is, the K_i, for $i \in \{1, 2, \ldots, n_{\mathrm{b}}\}$, define a (random) distinct partition of C, i.e.,

$$C = K_1 \dot{\cup} K_2 \dot{\cup} \ldots \dot{\cup} K_{n_{\mathrm{b}}}, \tag{4}$$

with $n_{\mathrm{b}} \coloneqq N_{\mathrm{obs}}/n_{\mathrm{sub}}$. The machine learning model produces a population posterior $q(\Lambda|\mathcal{D}_{K_i})$ for each of the subpopulations, which approximates $p(\Lambda|\mathcal{D}_{K_i})$. The repeated application of Bayes's theorem yields the complete posterior informed by all events in C, i.e.

$$q(\Lambda|\mathcal{D}_C) \coloneqq \frac{\mathcal{N}}{p(\Lambda)^{n_{\mathrm{b}}-1}} \prod_{i=1}^{n_{\mathrm{b}}} q(\Lambda|\mathcal{D}_{K_i}), \tag{5}$$

with $p(\Lambda)$ the prior on the hyperparameters and $\mathcal{N}$ is a normalization constant given by $\mathcal{N}^{-1} \coloneqq \int [\prod_{i=1}^{n_{\mathrm{b}}} q(\Lambda|\mathcal{D}_{K_i})/p(\Lambda)^{n_{\mathrm{b}}-1}]\mathrm{d}\Lambda$. In the limit $q(\Lambda|\mathcal{D}_C) \coloneqq p(\Lambda|\mathcal{D}_C)$, $\mathcal{N} = (\prod_{i=1}^{n_{\mathrm{b}}} p(\mathcal{D}_{K_i}))/p(\mathcal{D}_C)$, where

$$\begin{aligned} p(\mathcal{D}_{K_i}) &= \int p(\mathcal{D}_{K_i}|\Lambda) p(\Lambda) \mathrm{d}\Lambda \\ p(\mathcal{D}_C) &= \int \left[\prod_{i=1}^{n_{\mathrm{b}}} p(\mathcal{D}_{K_i}|\Lambda)\right] p(\Lambda) \mathrm{d}\Lambda. \end{aligned} \tag{6}$$

In Sec. IV, we assume a uniform prior of the hyperparameters Λ so that the denominator in Eq. (5) also amounts to a normalization constant. If the model correctly learns the posterior distribution that analyzes a subpopulation of events, we should have the approximation

$$q(\Lambda|\mathcal{D}_C) \approx p(\Lambda|\mathcal{D}_C). \tag{7}$$

The target distribution is conditioned on the observed data. In general, this could be a large space, making the learning task more complex. However, not all components of the data are informative about the target distribution. It is clear from the form of a standard HBA, Eq. (2), that one possible summary of the data for each event is the set of samples from the individual event parameter posterior distribution. Therefore, we make the choice to represent the input data via a set of posterior samples for the GW events. The neural network (NN) then learns the population posterior from the posterior samples of the individual signals in one subpopulation. We denote the set of posterior samples of the events in K as $\hat{\theta}_K$ and the number of posterior samples per event as n_{post}, assumed to be equal for all events. We define,

$$\hat{\theta}_K = \{\hat{\theta}_{ij} \colon i \in K; j = 1, 2, \ldots, n_{\mathrm{post}} - 1, n_{\mathrm{post}}\}, \tag{8}$$

[6]The hyperparameters samples are combined via importance sampling, as detailed in Sec. II B 1.

[7]Throughout, we assume the length of the subset of events n_{sub} to divide the total number of events N_{obs}.

where

$$\hat{\theta}_{i\bullet} \sim p(\theta|\mathcal{D}_i), \tag{9}$$

for i an event in the subpopulation K. From our choice of the data representation, we can then schematically write

$$q(\Lambda|\mathcal{D}_K) \approx q(\Lambda|\hat{\theta}_K). \tag{10}$$

In principle, however, the network could learn the population posterior from any representation of the data $\mathcal{D}_K$ that is sufficiently informative; this could be the Fourier-transformed or the time-domain strain data. Of course, no matter the representation of the data, the resulting posterior distribution should be the same.

The neural networks used in this work have $\mathcal{O}(10^{6-8})$ parameters that are optimized during the training process to minimize the chosen loss function, ensuring that the learned function converges to the desired distribution. We take the loss function to be proportional to the Kullback-Leibler (KL) divergence (up to an additive constant), which is defined as [102]

$$D_{\mathrm{KL}}(p\|q) \coloneqq \int p(x) \log\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x. \tag{11}$$

The KL divergence is positive semi-definite, and is zero only if $p = q$. Also, note that the KL divergence is not symmetric in the distributions p and q. Thus, it can be seen as a (generalized) distance between the target distribution and the one learned by the network.

The objective is thus to minimize $D_{\mathrm{KL}}(p(\Lambda|\mathcal{D}_K)\| q(\Lambda|\mathcal{D}_K))$. In reality, we will be approximating $p(\Lambda|\mathcal{D}_K)$ by $p(\Lambda|\hat{\theta}_K)$, since we assume that the data $\mathcal{D}_K$ is summarized accurately by the single-event posterior samples $\hat{\theta}_K$. This can be done by minimizing the loss

$$\mathbb{L} \coloneqq \mathbb{E}_{p(\Lambda)}\mathbb{E}_{p_{\mathrm{pop}}(\theta_K|\Lambda)}\mathbb{E}_{p(\mathcal{D}_K|\theta_K)}\mathbb{E}_{p(\hat{\theta}_K|\mathcal{D}_K)}\left[-\log\left(q(\Lambda|\hat{\theta}_K)\right)\right], \tag{12}$$

where we have introduced the expectation value

$$\mathbb{E}_{p(x|y)}[f(x,y)] \coloneqq \int \mathrm{d}x p(x|y) f(x,y). \tag{13}$$

The right-hand side of Eq. (12) is the expectation value over four distributions. Averaging over noise realizations, $\mathcal{D}_K$, and population draw, θ_K, we can apply Bayes's theorem successively to obtain the equality (see Appendix B)

$$\mathbb{L} = -\mathbb{E}_{p(\hat{\theta}_K)}\mathbb{E}_{p(\Lambda|\hat{\theta}_K)} \log\left[q(\Lambda|\hat{\theta}_K)\right]. \tag{14}$$

From the definition of the KL divergence in Eq. (11), we rewrite the above equation as

$$\mathbb{L} = \mathbb{E}_{p(\hat{\theta}_K)}\left[D_{\mathrm{KL}}(p(\Lambda|\hat{\theta}_K)\|q(\Lambda|\hat{\theta}_K)) - D_{\mathrm{KL}}(p(\Lambda|\hat{\theta}_K)\|1)\right]. \tag{15}$$

Thus, this expression is [up to a constant and the expectation value over $p(\hat{\theta}_K)$] the KL divergence between the model $q(\Lambda|\hat{\theta}_K)$ and the target distribution $p(\Lambda|\hat{\theta}_K)$. Since the KL divergence is minimized for $p = q$, it follows that the above loss is also minimized for $p(\Lambda|\hat{\theta}_K) = q(\Lambda|\hat{\theta}_K)$, and if the network is properly trained, $q(\Lambda|\hat{\theta}_K)$ will approximate $p(\Lambda|\hat{\theta}_K)$. If a network achieved the minimum loss for every possible choice of input parameters, $\hat{\theta}_K$, then it would perfectly represent the population posterior. In practice, this will not be achievable. By averaging the loss over noise realizations, $\mathcal{D}_K$, and population draws, θ_K, we ensure that learning effort is expended to represent the distribution best for values of $\hat{\theta}_K$ that are more likely to be observed in practice.

To evaluate the loss value of Eq. (12) one has to evaluate an expectation value over four distributions. We approximate these expectation values by Monte Carlo averaging, i.e.

$$\mathbb{L} \approx \frac{1}{N} \sum_{\{\Lambda_\nu, \mathcal{D}_{K,\nu}\}} q(\Lambda_\nu|\mathcal{D}_{K,\nu}), \tag{16}$$

where N is the number of samples drawn as follows: according to the prior $p(\Lambda)$ we draw population parameters. For each sample Λ, we create the cosmological model, draw n_{sub} true events, simulate n_{sub} observed strains (passing some specified selection threshold) and produce n_{post} posterior samples. For computational reasons, we precompute the samples $\{\Lambda_\nu, \mathcal{D}_{K,\nu}\}$ and call the resulting data the *training dataset*. The loss is then minimized over choices for the NN parameters during the training process.

Note that at no point in the process is the (true) population posterior explicitly evaluated. The above scheme relies solely on the *simulation of data* rather than on evaluating the hierarchical Bayesian likelihood in Eq. (2). As a consequence, it does not require an analytic expression for the population prior, but solely relies on a forward model to generate training data (i.e. samples from the population likelihood). This differs from most HBAs, with the exception of [103,104], which instead use Monte Carlo integration to evaluate the population likelihood, using the population prior to draw samples. In turn, that approach requires us to be able to efficiently estimate the single-event likelihood.

Also, by construction, the model contains the selection effect term $\xi(\Lambda)$ appearing in the denominator of Eq. (2). We thus avoid the computation of this term during inference.[8]

In some cases the NPE results differ from the HBA approach for reasons we elaborate below. These differences can be corrected by reweighting the NPE samples to the target HBA posterior using importance sampling weights

$$w(\Lambda) = \frac{p(\Lambda|\mathcal{D}_C)}{q(\Lambda|\mathcal{D}_C)}. \tag{17}$$

This is possible because we have access to the learned NPE posterior density, and have an explicit expression for the target HBA density. We show this procedure on one example in Sec. V A. Importance sampling can also provide a validation: an unchanged posterior (after reweighting) implies that the model has learned the correct HBA distribution.

2. Combining subpopulations of events

In the previous section, we have subdivided the complex problem of obtaining the posterior distribution from catalogs of GW events into multiple simpler problems, namely to obtain the posterior distribution from a subpopulation of GW events. We thus train a model q to approximate the population parameter posterior informed by a *subset* of events $\mathcal{D}_{K_i}$, i.e. $p(\Lambda|\mathcal{D}_{K_i})$. One is eventually interested in the posterior as informed by the event catalog $C = \dot{\cup}_{i=1}^{n_b} K_i$. To obtain this distribution we apply the following procedure:

(1) With the model, we draw N_{prop} Λ samples from each of the posteriors $q(\Lambda|\mathcal{D}_{K_i})$ analyzing a subpopulation of GW events—these are our proposal samples. In total, we have $n_b \times N_{\text{prop}}$ samples.
(2) Out of these, we randomly choose N_{prop} samples. The chosen samples follow the distribution $q_{\text{init}}(\Lambda|\hat{\theta}_C) := \frac{1}{n_b}\sum_{i=1}^{n_b} q(\Lambda|\mathcal{D}_{K_i})$.
(3) We evaluate the combined population posterior according to Eq. (5) for the proposal samples with our model; to obtain $q(\Lambda|\hat{\theta}_C)$. From this, we can compute the weights w as

$$w(\Lambda|\hat{\theta}_C) = \frac{q(\Lambda|\hat{\theta}_C)}{q_{\text{init}}(\Lambda|\hat{\theta}_C)} = \frac{\frac{\mathcal{N}}{p(\Lambda)^{n_b-1}}\prod_{i=1}^{n_b} q(\Lambda|\mathcal{D}_{K_i})}{\frac{1}{n_b}\sum_{i=1}^{n_b} q(\Lambda|\mathcal{D}_{K_i})}, \tag{18}$$

where we applied the definition of $q(\Lambda|\hat{\theta}_C)$ in Eq. (5) in the second equality.
(4) The samples are importance weighted according to $w(\Lambda|\hat{\theta}_C)$ above. The reweighted samples follow the desired distribution $q(\Lambda|\hat{\theta}_C)$.

In order to apply this procedure it is vital that one can sample from the distribution and that one has access to the probability with which the samples are created. The architecture of a normalizing flow allows for this. The generation of random samples with normalizing flows is rapid, making the scheme fast. We will apply the procedure in practice and compare it to the conventional HBA method in Sec. V.

Other scheme are also possible: one could multiply the hierarchical (neural) posterior [dividing out the prior, cf. Eq. (5)] that analyze each of the subpopulations (the probability of which is given by the flow) and use MCMC sampling to recover the combined posterior (that analyzes the entire catalog). The sampling method we outlined above avoids running a full MCMC analysis, which would further increase the computing time. We have compared the two approaches for selected cases and found very good agreement.

C. Flow architecture

The following section summarizes the building blocks that make up our NPE model. The proposed machine learning model combines two embedding neural networks for data compression and a normalizing flow for population posterior generation as described in Fig. 1. In the following, we refer to the full algorithm simply as the "model."

The posterior samples from all events in one subpopulation represent a large dataset that we seek to reduce with two embedding networks that summarize (i) the individual events in a first stage and (ii) the set of all summaries of n_{sub} events produced by the first embedding network. The flow is then conditioned on the output of the second embedding network. Figure 1 shows a schematic overview of the model architecture.

The first embedding network summarizes each single-event posterior.[9] This network takes as input data the collection of n_{post} posterior samples of the component masses and the luminosity distance (following Sec. II A); that is a three-dimensional posterior distribution for the n_{sub} events in the subpopulation. We have found that 16 "summary" parameters for the single-event posterior are sufficient to recover the population posterior. The first embedding network is *identical* for each event. If the flow analyzes n_{sub} events, we thus have $16 \times n_{\text{sub}}$ scalars describing the input data after applying the first embedding

[8]To obtain constraints on a different population distribution requires the training of a new model, which also entails the generation of new training data. This is different from conventional analysis, where the original injection set (to evaluate the selection effect) can be recycled, provided that it covers the parameter space of the new population model sufficiently well.

[9]As input data of the first embedding network, we use the standardized posterior samples (subtracting the mean and dividing by the standard deviation of the respective variable). This standardization of the input data is a common practice in machine learning and allows for faster convergence of the model.

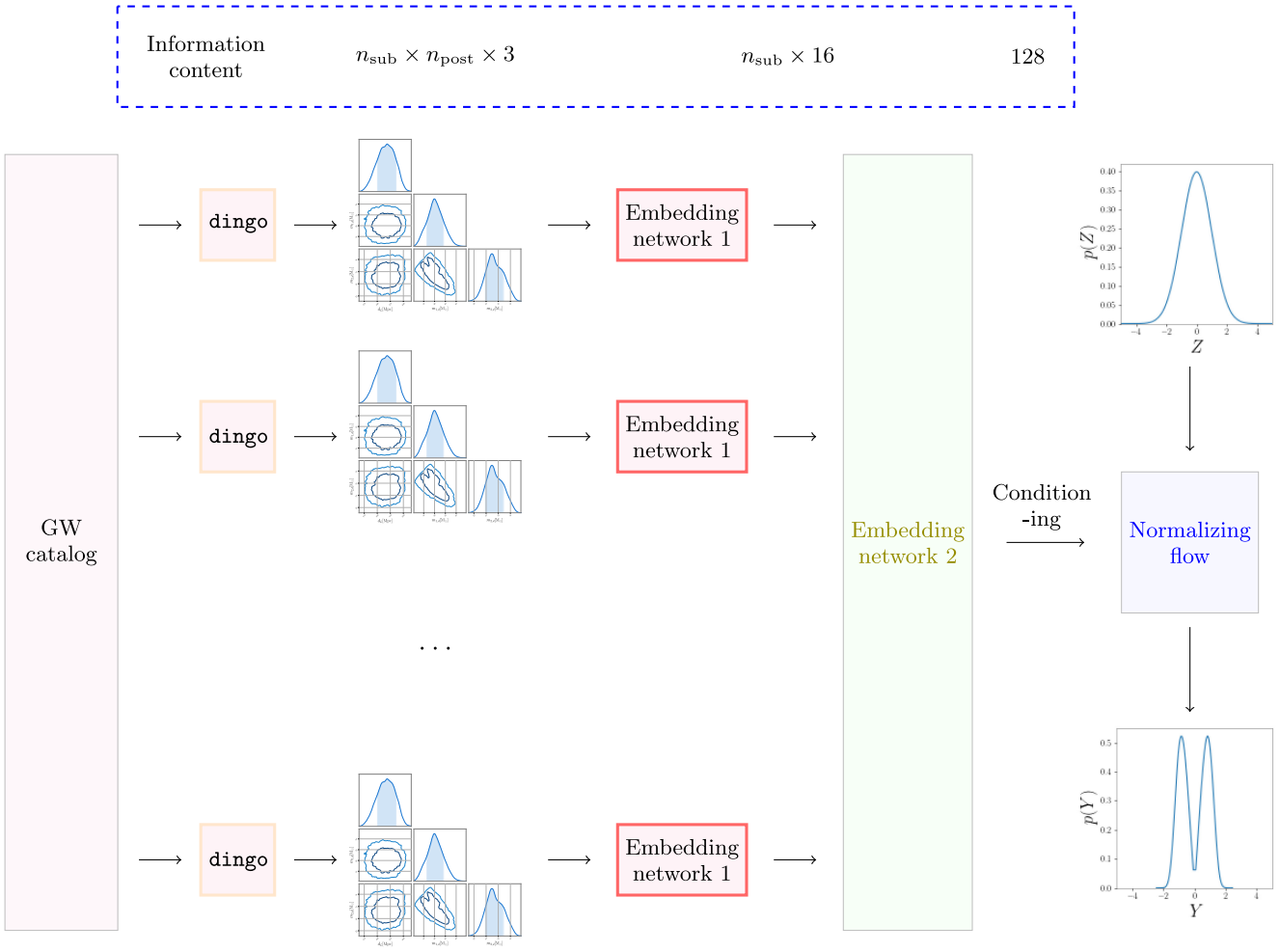

FIG. 1. Overview of the data reduction with the embedding networks and the conditioning of the normalizing flow. This method reduces the data dimension from initially $n_{\rm sub} \times n_{\rm post} \times 3$ to 128. Note that the embedding network 1 is identical for all GW events. This data summary reduces the number of adjustable NN parameters and hence simplifies the training process.

network. These data are then further summarized by an additional embedding network, whose output feeds into the flow. We choose this second summary to have 64 and 256 parameters for the two models we train in the result section below. The embedding network parameters (summarizing the input data) also implicitly appear in the loss [cf. Eq. (12)] and are therefore optimized jointly with the parameters that define the flow transformation as discussed below. The two embedding networks significantly reduce the number of free parameters in the model, leading to less overfitting.

As anticipated, the normalizing flow is conditioned on the output of the second embedding network and computes an approximation of the true population posterior $p(\Lambda|\mathcal{D}_K)$. In general, the normalizing flow performs a transformation that maps the physical variables (here the hyperparameters $\mathbf{Y} \coloneqq \Lambda$) to an unphysical variable $\mathbf{Z}$ that follows the normal distribution.[10] One can rapidly generate samples in $\Lambda \sim q(\Lambda|\mathcal{D}_{K_i})$ by drawing samples from the normal distribution $[\mathbf{Z} \sim \mathcal{N}(\mu = 0, \sigma = 1)]$ and applying the flow transformation to them, i.e. $\Lambda = \mathbf{g}(\mathbf{Z})$.

We use the `nflows` [105] package to construct this transformation, where the flow transformation from $\mathbf{Z}$ to $\mathbf{Y}$ is constructed from a sequence of simple transformations. In our case, these are piecewise rational quadratic coupling transforms [106] in analogy to those implemented in [90].[11] The number of coupling transforms is referred to as the number of *flow steps*. The parameters governing the coupling transforms are trained to minimize the loss defined in Eq. (12). We have investigated different choices of parameters and found that four flow steps, each parametrized by a fully connected residual network with 32 parameters and five to fifteen layers provided the best results. Table IV summarizes the details of the specific network architecture. Throughout, we use graphics

[10]We follow the standard notation as used in the review of normalizing flows of [84].

[11]See Appendix A for additional details on these transformations.

processing units (GPUs) to accelerate both the generation of single-event posterior samples and the training of the normalizing flow.

III. ASTROPHYSICAL AND INSTRUMENTAL SETUP

In the following we describe our assumptions on the astrophysical population of binary black holes (BBHs), the detector network, the detection criterion and the generation of waveforms.

A. Assumptions on the population distribution

Throughout this work, we model sources as uniformly distributed in comoving volume in a flat ΛCDM universe described by the Hubble constant, H_0, and the matter content, $\Omega_{\rm m}$. We fix the latter to $\Omega_{\rm m} = 0.3$ and assume it to be known. This assumption is straightforwardly relaxed, but this is beyond the scope of our work here.

The two training datasets we construct follow the POWER LAW source frame mass distribution.[12] This source frame mass model is characterized by four parameters. The minimum mass $m_{\rm min}$ and the maximum mass $m_{\rm max}$ limit both source frame masses from below and above, respectively. In addition, we have two power law slope parameters α and β characterizing the distribution of masses according to

$$\begin{aligned} p(m_{1,s}|\Lambda_m) &= \mathcal{N} m_{1,s}^{-\alpha} \chi_{[m_{\rm min}, m_{\rm max}]}(m_{1,s}), \\ p(m_{2,s}|m_{1,s}, \Lambda_m) &= \mathcal{N}' m_{2,s}^{\beta} \chi_{[m_{\rm min}, m_{1,s}]}(m_{2,s}), \end{aligned} \tag{19}$$

where we have defined the normalization constants $\mathcal{N}$ and $\mathcal{N}'$, as well as the set of hyperparameters $\Lambda_m = \{m_{\rm min}, m_{\rm max}, \alpha, \beta\}$. Finally, χ is the characteristic function, defined as

$$\chi_{[a,b]}(x) \coloneqq \begin{cases} 1 & \text{if } x \in [a, b] \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

B. Assumptions on BBH sources

We focus on precessing BBHs in quasi-circular orbits, characterized by 15 parameters: the component (detector frame) masses of the BHs, $m_{1,d}, m_{2,d}$, the two spin vectors described by their magnitudes, a_1, a_2, angles of reference, ϕ_{12}, ϕ_{jl}, and tilts, θ_1, θ_2, as well as the time of coalescence, t_c, the phase at coalescence, Φ_c, declination and right ascension of the sky position, $\delta_{\rm SP}$ and $\alpha_{\rm SP}$, luminosity distance, d_L, the inclination of the orbital plane with respect to the line of sight, θ_{jn} and polarization angle, ψ. We assume sources to be distributed uniformly in the sky, we draw the spins isotropically over the sphere and the spin magnitudes uniformly between 0 and 0.99.

The IMRPHENOMXPHM waveform [107] is used to model the gravitational wave signal of the BBH coalescences in the frequency domain.

C. Assumptions on waveform and the detector network

We assume the O1 sensitivity curve [1] for the Laser Interferometer Gravitational wave Observatory (LIGO) Hanford and LIGO Livingston detectors and impose as a selection criterion a signal-to-noise ratio (SNR) threshold of 12.[13] Given a detected GW signal, we then use the deep learning tool `dingo` [79,91] to analyze the strain data and produce single-event posterior samples. Due to limitations in the size of the parameter space of signals that can be reliably analyzed we restrict the mass range of signals to a conservative mass cutoff *in source frame mass* of $m_{1,s} \geq m_{2,s} \geq 18 M_\odot$.[14] This lower bound on the mass range directly implies a lower bound on the prior of $m_{\rm min}$ we can explore. Also, recall that the mass spectrum method uses only the component masses and luminosity distance. We thus discard the remaining single-event parameters. See Table III for the priors on the training set and, by extension, the prior learned by the model.

IV. TRAINING DATASETS

The number of events per subpopulation, $n_{\rm sub}$, is a free parameter of this approach, the optimal value of which we would like to determine. To study this, we build two different training sets and train models on each one, assuming the same detector network and with the population model, described in Sec. III. In the following, we refer to a *hypersample*, as one population of events that share the same hyperparameters. Training set `low` has 6.7×10^5 hypersamples, which is ten times more hypersamples than that of training set `high` (which has 4.4×10^4 hypersamples). However, training set `low` has only ten events per population, which is 20 times less than training set `high` (which has 200 events per hypersample). Overall, the two datasets contain approximately the same number of GW events (and posterior samples) and hence, their information content (and their computational cost) is also approximately equal. For this reason, the performance of the models trained on the respective datasets can be directly compared. Training dataset `low` only allows small event subpopulations ($\leq$10), but with an in-depth training on many population examples, whereas training dataset `high`

[12]This is the simplest of the four source frame mass distribution considered currently by the LIGO-Virgo-KAGRA Collaboration [10,12,16].

[13]This is an approximation, since the application of this method to real data will make the selection criteria more intricate, e.g. incorporating the false alarm rate. To fully account for selection effects, an injection campaign would be needed.

[14]In the near future, the lower bound of `dingo`'s mass range is expected to decrease to $5\,M_\odot$.

TABLE II. Properties of the two generated training datasets.

Summary of training datasets		
Study	`low`	`high`
Number of training population samples	6.7×10^5	4.4×10^4
Number of available events per population	10	200
Number of available posterior samples per event	200	200

TABLE III. Summary of priors assumed for the two training datasets. The uniform prior is denoted as $\mathcal{U}$.

Summary priors		
Metaparameter	Prior	Unit
H_0	$\mathcal{U}(40, 140)$	km s^{-1} Mpc^{-1}
$m_{\min}$	$\mathcal{U}(18, 30)$	$M_\odot$
$m_{\max}$	$\mathcal{U}(37, 47)$	$M_\odot$
α	$\mathcal{U}(-2, 2)$	⋯
β	$\mathcal{U}(-2, 2)$	⋯

allows for large subpopulations at the price of a limited number of populations.

A. Training set `low`

Each hypersample of the training data contains the true value of the hyperparameters, Λ, ten events and 200 associated posterior samples in three variables: the (detector frame) component masses and luminosity distance. In total, the data associated to one population hypersample thus contains $6000 = 10 \times 200 \times 3$ scalars.

During one training epoch, we randomly choose n_{sub} events among the ten events for each hypersample with n_{post} random posterior samples each. This sampling method increases the variability of the input data. After several trials, we have found that the model provides a good approximation to the population posterior if it analyzes six events per subpopulation ($n_{\text{sub}} = 6$).[15] The same reasoning applies to the number of posterior samples, 100 posterior samples per event seem to be sufficient to produce a faithful approximation (although see the discussion below in Sec. V B).

A summary of the training dataset `low` can be found in table II.

B. Training set `high`

With future GW detector networks in mind, we also construct a training dataset that allows for models that analyze a much larger number of events. Each population hypersample of the training set `high` includes 200 events with 200 posterior samples per event. The data associated to each hypersample thus includes $120,000 = 200 \times 200 \times 3$ scalars. The model we train below selects randomly 100 out of the 200 available events during each training epoch. For each of these events, the flow chooses 100 out of 200 posterior samples at random. Thus, each input population sample includes $30,000 = 100 \times 100 \times 3$ scalars. Again, this method of drawing a subset of the available data is introduced to reduce overfitting. For training set `high` we find that $n_{\text{sub}} = 100$ gives the best performing models. This is the result of a trade-off; given that one hypersample contains 200 GW events, if n_{sub} is set higher the variability of the training data is not high enough, and if n_{sub} is too low we cannot analyze a large number of events, since too many subpopulations have to be combined. Empirically, we find that the model does not produce reliable results above $\mathcal{O}(10\text{–}20)$ combinations of different subpopulations.

C. Training the networks

Given the training data, we train different models by minimizing the loss we have introduced in Eq. (12), varying the network parameters, as well as the number of GW events that are taken as input parameters. The parameters describing the flows that yield the best agreement with standard HBA results are summarized in Table IV. The network trained on dataset `low` (which we

TABLE IV. Architecture of the embedding networks and the normalizing flow. For the hidden layers of the embedding networks we use the tuple notation $X^n := (X, X, \ldots, X)$, with X repeated n times.

Summary of the normalizing flow parameters		
	Model	
Variable	`low`	`high`
Events per batch n_{sub}	6	100
Posterior samples per event n_{post}	100	100
Dimensions embedding network 1	(512, 256^5, 128, 64)	(512, 256, 128, 64)
Dimensions embedding network 2	(512^4, 256^4, 128^2, 64^3)	(1024^2, 512^2, 256)
Flow steps	3	4
Spline points	8	6
Hidden dimensions(spline network)	32	32
Hidden layers(spline network)	5	15
Training epochs	200	300
Learning rate	0.0001	0.0001
Scheduler	Plateau	Plateau
Batch size	1024	1024

[15]The combination of subpopulations of events delivers more reliable posterior distributions if the number of events per subpopulation (n_{sub}) is high. However, the time to generate the training dataset limits the maximum value n_{sub} which therefore, cannot be set arbitrarily high.

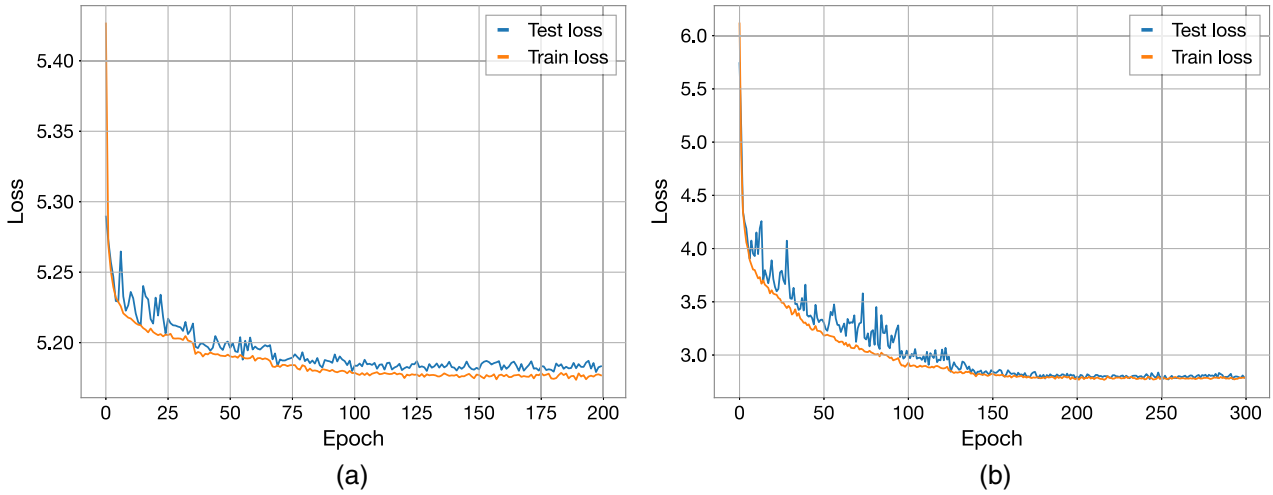

FIG. 2. Loss for the (left) model `low` and (right) model `high`. Since the test loss (blue) and train loss (orange) do not differ much, we conclude that the models generalize well to unseen data.

refer to in the following as model `low`) had a training time of ∼5.5 h. In Fig. 2(a), we present the training and test loss curves for the model. Based on the loss curves, we conclude that the model can generalize effectively to data that were not included in the optimization process.

The training time of model `high` was 82 min. The shorter training time (compared to model `low`) is due to the smaller number of hypersamples in training dataset `high`. The associated training and test loss curves of model `high` are plotted in Fig. 2(b). We discuss the resulting respective posterior distributions in Sec. V A for the training set `low` and Sec. V B for training set `high`.

V. RESULTS

A. Results with model `low`

As a first validation step, we generate the P-P-plot of the model. To this end, we draw 1000 population hypersamples from the training dataset, input the corresponding posterior samples in the model, and sample $q(\Lambda|\mathcal{D}_{K_i})$ for each. From the Λ samples, we compute the percentile in which the true value of the population lies and sort the resulting percentiles by value. The cumulative density of the percentiles is shown in Fig. 3(a). If the model correctly infers the hyperparameters, the figure should follow the diagonal within a reasonable error interval. From the Kolmogorov-Smirnov

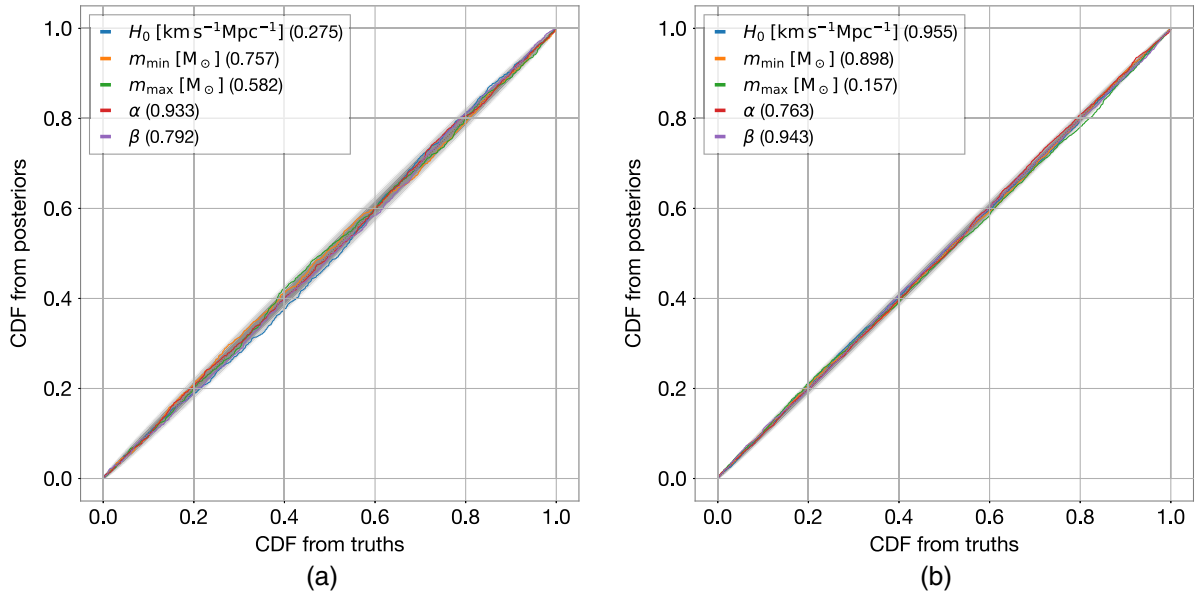

FIG. 3. The P-P-plot for 1000 injections for model `low` (left) and 2500 injections for model `high` (right). We find p values as indicated in the legend, indicating that the models reconstruct the population posterior correctly. However, the lowest p values of model `high` are slightly lower than the model `low`. The $m_{\max}$ parameter of model `high` has the lowest value, with 15.7%.

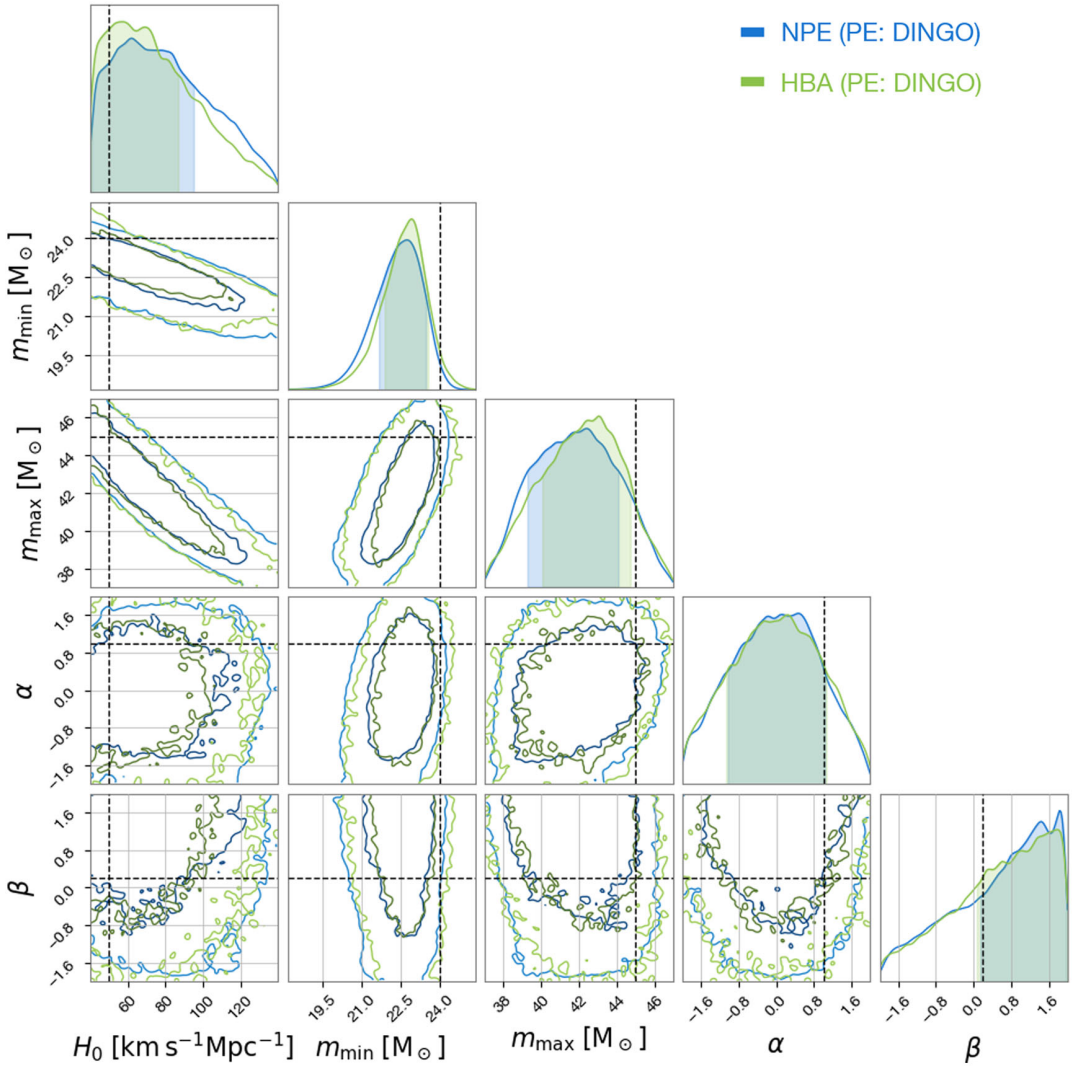

FIG. 4. Results from model `low` (NPE, blue) compared to a conventional hierarchical Bayesian analysis (HBA, green). The posterior analyzes 60 GW events of population No. 4 (cf. Appendix C). The one and two sigma intervals are indicated as two-dimensional contours and dashed lines mark the true values of Λ. We could produce an effective sample size of $\sim 8 \times 10^4$ posterior samples in 2.5 min of computation time. In total, we have verified our results on a total of twelve populations. For the largest discrepancy between our model and the conventional approach consider the result in Fig. 5. The "PE" stands for individual event parameter estimation which, in both cases, uses the `dingo` algorithm.

(KS) test (comparing the computed percentiles against the uniform distribution), we obtain p values between 27% and 93%, which is expected for five variables, indicating that the model reconstructs the population posterior correctly.

The above tests only include $n_{\rm sub} = 6$ events per population. To further validate our results, we generate a detected population with 60 GW events. With `dingo`, we produce posterior samples for each of these events and run a classical Monte Carlo Markov chain (MCMC) analysis (with an analytical likelihood, using `icarogw` [60]) on these samples. The resulting hyperparameter posterior distribution serves as our ground truth. This scheme differs from the classical approach (for instance in [16,92]) as the computationally expensive parameter estimation (PE) has been carried out with `dingo`. This "crossing" from the likelihood-free inference part to the classical inference (with `icarogw`) is a simplification and comes at a cost. If `dingo` does not correctly estimate the single-event posterior distribution, the resulting distribution of the hyperparameters with `icarogw` will not represent the ground truth. We will discuss below for model `high` the possible consequences of this assumption. Note that it is possible to correct for possible inaccuracies of `dingo` by importance sampling. However, since this significantly increases the computation time we do not choose to pursue this here.

In parallel, we apply our model to the same single-event posterior samples and combine the results with the importance sampling step that was outlined in Sec. II B 2.

TABLE V. JS divergence (in units of 10^{-3}nat) for all hyperparameters and all twelve populations. The mean and median values over all populations are also presented. The most problematic parameters are $m_{\rm min}$ and $m_{\rm max}$. We have indicated which population are plotted in later sections, corresponding to cases where our model differs weakly or strongly to the conventional approach.

	JS divergence (10^{-3} nat)				
Population	H_0	$m_{\rm min}$	$m_{\rm max}$	α	β
0	6.4	26.4	4.9	2.9	1.6
1 (Fig. 5)	10.0	70.9	15.3	17.7	4.0
2	3.8	6.3	11.9	4.1	5.8
3	3.2	4.3	4.2	8.5	1.5
4 (Fig. 4)	7.0	10.3	3.5	0.9	1.4
5	0.9	4.2	7.2	6.8	0.8
6	1.8	6.6	11.6	8.1	3.5
7	2.6	16.4	18.4	3.4	21.3
8	8.9	4.6	4.5	3.4	1.0
9	2.3	10.1	17.2	6.2	2.1
10	30.6	10.2	21.8	4.0	0.8
11	3.4	4.1	11.1	6.9	2.6
Mean	6.73	14.53	10.97	6.07	3.86
Median	3.61	8.36	11.35	5.16	1.81

This procedure is applied for twelve different populations.[16] Figure 4 shows one out of these twelve distributions, with a model that analyzes 60 events in total. Since the network was trained for $n_{\rm sub} = 6$ events, we divide the input data in $n_{\rm b} = 10$ subpopulations. The result shows that it is possible to combine the output of multiple model evaluations and obtain the correct population posterior. This figure is representative of the majority of cases—we generally see good agreement between the two methods. We have also verified that the (arbitrary) division of events in the different subpopulations does not impact the resulting population posterior.

To make this comparison more quantitative, we compute the Jensen-Shannon (JS) divergence[17] between the NPE and HBA results for each of the variables in Λ. Table V collects these values. The lower the JS divergence, the better the agreement between two distributions. The JS divergence for single-event PE with LALINFERENCE (for identical runs with different random seeds) is $\sim 7 \times 10^{-4}$nat [108]. For two `icarogw` runs with the same settings, we find JS divergences between 3×10^{-3}nat and 10^{-4}nat. The JS divergences observed in our experiments are an order of magnitude higher than these baselines. Thus, there exists potential for further refinement in our approach. The problems appear almost exclusively for two hyperparameters: the minimum and maximum mass of the population, $m_{\rm min}$ and $m_{\rm max}$, and out of the two, the minimum mass proves to be the more difficult parameter. Out of the 60 JS divergences we have analyzed (twelve populations with five hyperparameters each), 16 had a JS divergence larger than 0.01. The population models which proved to be most difficult to reconstruct were populations 1, 7 and 10, where population 1 and 7 both have low Hubble constant. Additionally, population 7 has $m_{\rm max} = 38M_\odot$, very close to the boundary of the prior for which the model was trained ($m_{\rm max} = 37M_\odot$). It is well known the neural network performance decreases close to the prior boundary. We show in the next section that it is possible to recover the HBA result by applying importance sampling to the samples produced from our model.

1. Failing of the model and recovery from importance sampling

In certain cases the NPE samples do not agree with the HBA samples. However, we have access to the probability associated with which each population sample was generated through the construction of the NF. We can therefore obtain the HBA result by calculating the (conventional) population likelihood for each sample, $p(\Lambda|\mathcal{D}_C)$, and reweighting the NF samples to this target likelihood, using the weights determined by Eq. (17). Figure 5 shows this procedure on the example of population 1, and model `low`. For comparison, the current LVK cosmological inference code produces the classical result (with 24,000 samples and parallelizing on 16 cores) in ∼8 h, whereas the flow produced 300,000 samples in 2.3 min. Applying an additional importance sampling step (parallelizing on 16 cores) generated an effective sample size of 10^4 in ∼3.3 h. This gain in computation time is due to the reduced number of likelihood evaluations with the NPE (3×10^5) when compared to the HBA (1.7×10^6). These computation times do not include the times for single-event parameter estimation (here carried out with `dingo` and therefore, within minutes). Also note that one can implement hierarchical inference codes with just-in-time compilation, using GPUs and faster sampling algorithms [109] that generate hyperparameter posteriors in minutes.

When combining more than 10–20 subpopulations of events, the resulting NPE posterior becomes unreliable.[18] Consequentially, model `low` cannot analyze more than ∼100 events. This might be caused by model `low` not resolving the fine structure in the posterior distribution that

[16]Each population has a different Hubble constant, and variables parametrizing the source frame mass distribution. The details of the populations are given in Table VI.

[17]The JS divergence is a symmetrized version of the KL divergence that was defined in Eq. (11).

[18]We occasionally find a nonsmooth posterior distribution for models that analyze one subpopulation. While this does not significantly impact the result, these discontinuities accumulate when we compute the product of several posteriors that each analyze one subpopulation, respectively. This is likely a consequence of the chosen model architecture and not intrinsic to the method.

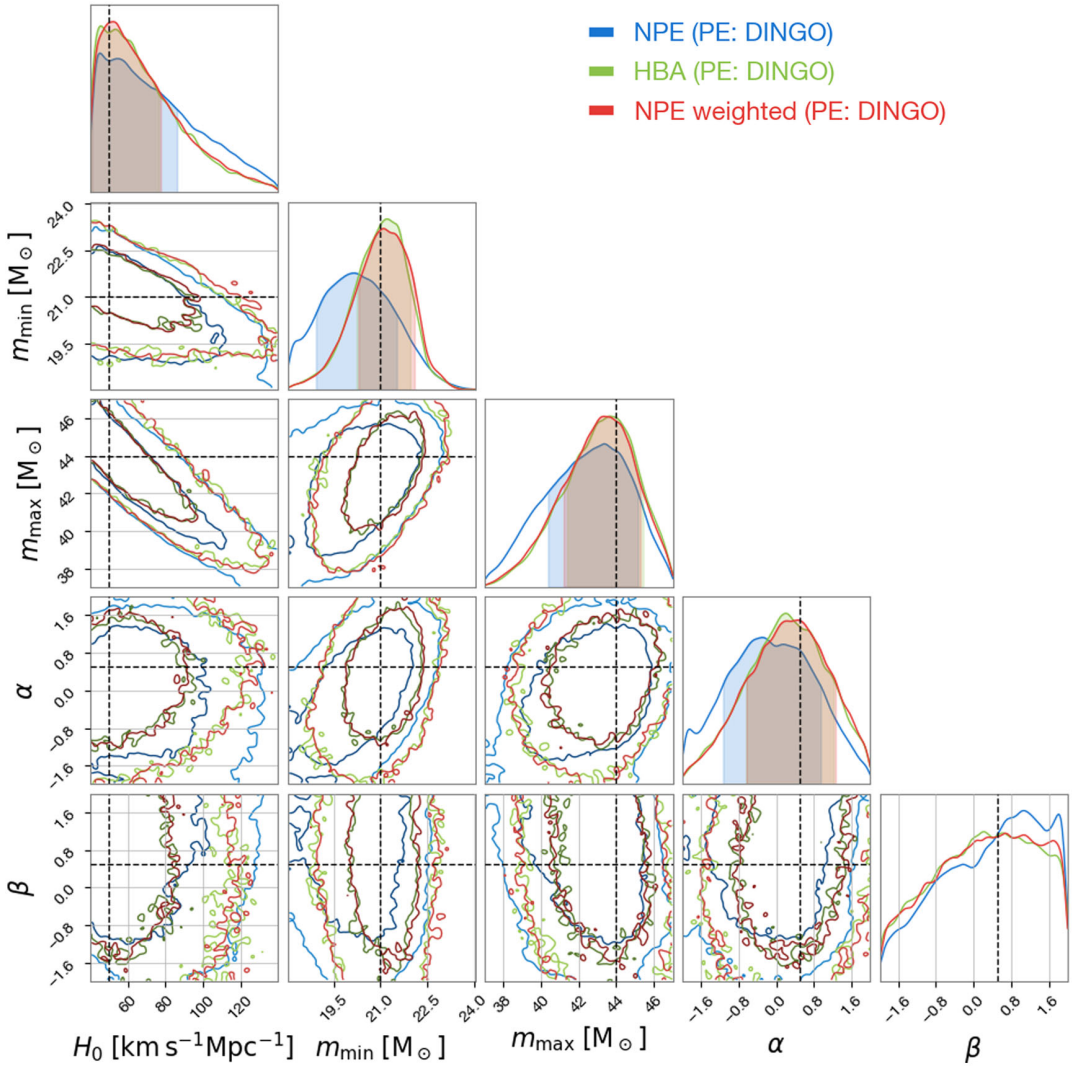

FIG. 5. Results from model `low` [NPE (PE: DINGO), blue] compared to a conventional hierarchical Bayesian analysis (HBA) (green). The posterior analyzes 60 GW events of population No. 1 (cf. Appendix C). The one and two sigma intervals are indicated as two-dimensional contours and dashed lines mark the true values of Λ. As the NF and the classical analysis differ, most notably in the variable $m_{\rm min}$, we perform an importance sampling with the classical likelihood. The resulting posterior is shown in red [NPE (weighted)] and agrees well with the classical result.

becomes important when combining large event sets. We thus rely on training set `high` to construct a model that can analyze a larger number of events as we now elaborate.

B. Results with model `high`

To show the capability of the model to reconstruct the population posterior given a large number of observed events, we use training set `high` to construct a model analyzing $n_{\rm sub} = 100$ GW events. Figure 2(b) shows the resulting loss curves of the training and test dataset, respectively. The train and test loss coincide, suggesting that the model can process unseen input data and generate accurate hyperparameter posterior distributions. Figure 3(b) shows the P-P-plot of model `high` with 2500 population realizations, implying that the network has correctly learned the desired posterior distribution.

We have verified that when analyzing 100 GW events that the NN is in good agreement with the HBA for all the populations described in Table VI. Considering a larger number of events, we focus on one specific population, with the parameters $H_0 = 67$ km s^{-1} Mpc^{-1}, $m_{\rm min} = 20.1 M_\odot$, $m_{\rm max} = 42.9 M_\odot$, $\alpha = 0.6$ and $\beta = -0.5$. Figure 6 compares the posterior of our model and the classical posterior, analyzing 600 events. Although the posterior distributions overlap, they show a significant deviation.[19]

[19]As previously, we can successfully recover the HBA result through importance sampling.

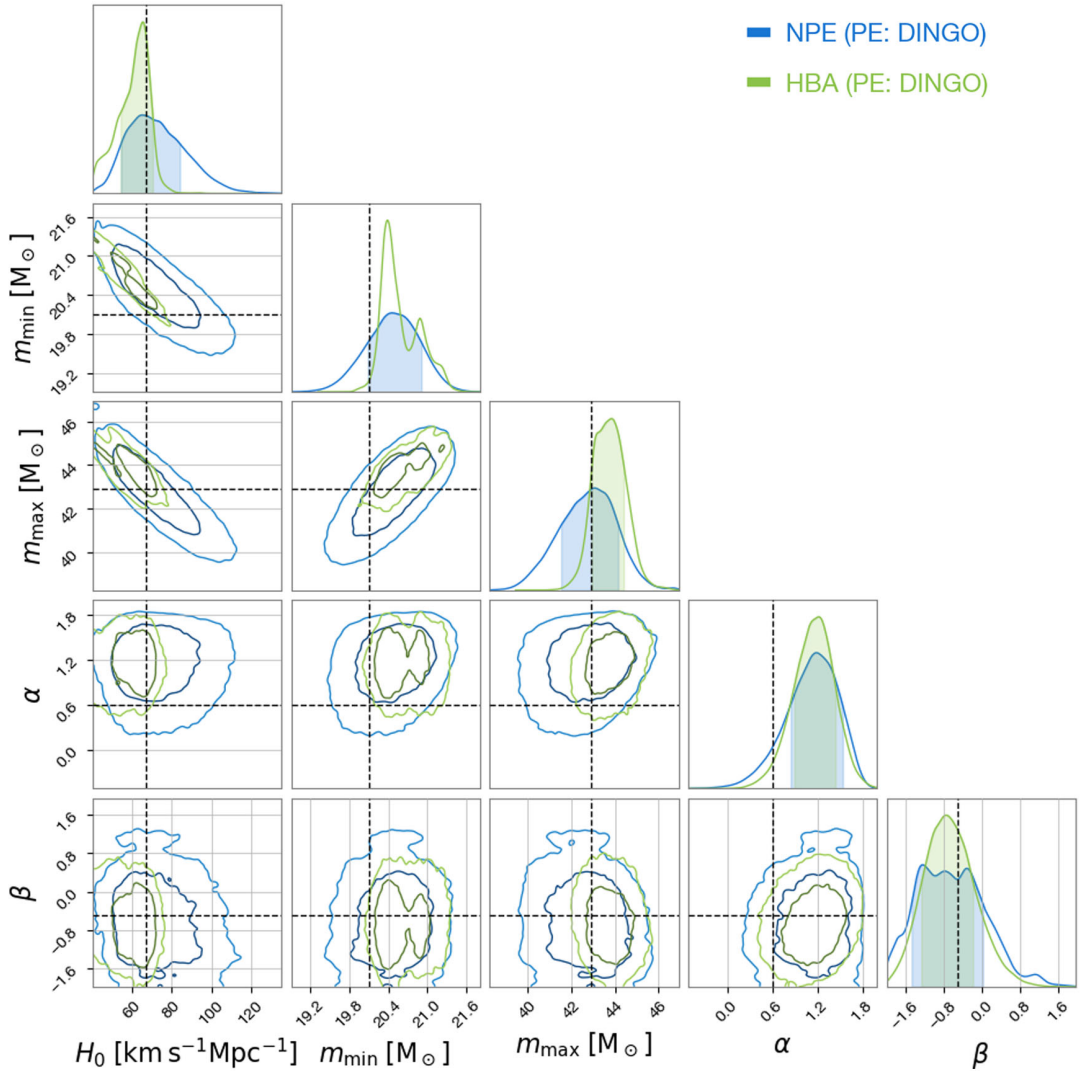

FIG. 6. Results from model `high` (blue) compared to a conventional hierarchical Bayesian analysis (HBA) (green). Both analyses use the `dingo` samples as input data. The posterior analyzes 600 GW events. The one and two sigma intervals are indicated as two-dimensional contours and dashed lines mark the true values of Λ. The model and the conventional analysis show a discrepancy. However, the model is generally closer to the injected values than the HBA method that relies on `dingo` PE samples.

The computation time for the conventional approach was 127 h[20] and for our model 7 min.

The reason for the discrepancy in Fig. 6 is still an open question: as anticipated above, we make an approximation of the ground truth. We use `dingo` to estimate the single-event posterior distributions, and these samples are then subsequently analyzed by `icarogw` to derive the hyperparameter posterior. To decrease the computation time we did not perform importance sampling on the generated `dingo` samples for the individual event posteriors. This can degrade the performance of the estimation of the single-event posterior. From a preliminary analysis, we find that some posterior distributions as inferred by `dingo` differ from the PE samples of `bilby`. In the near future, we hope to perform a full PE on all events and compute the ground truth from conventional analyses alone. However, we emphasize that even if the `dingo` algorithm is not a perfect approximation to the single-event posterior, this does not invalidate our approach—by construction the NPE model learns the posterior distribution marginalized over the `dingo` uncertainty.

There are other potential sources of the discrepancy, such as the smaller number of hypersamples represented in the training dataset `high`. We have checked that the posterior resulting from a model trained with training set `low` is compatible with a posterior trained with training set `high`.

Moreover, we find a strong dependence of the HBA results on the number of posterior samples per event (if the number of posterior is not "high enough"). This potentially additional source of uncertainty is now discussed.

[20]This computation time was for an injection set (used to compute the selection effect) of 1.4×10^5 detected GW signals.

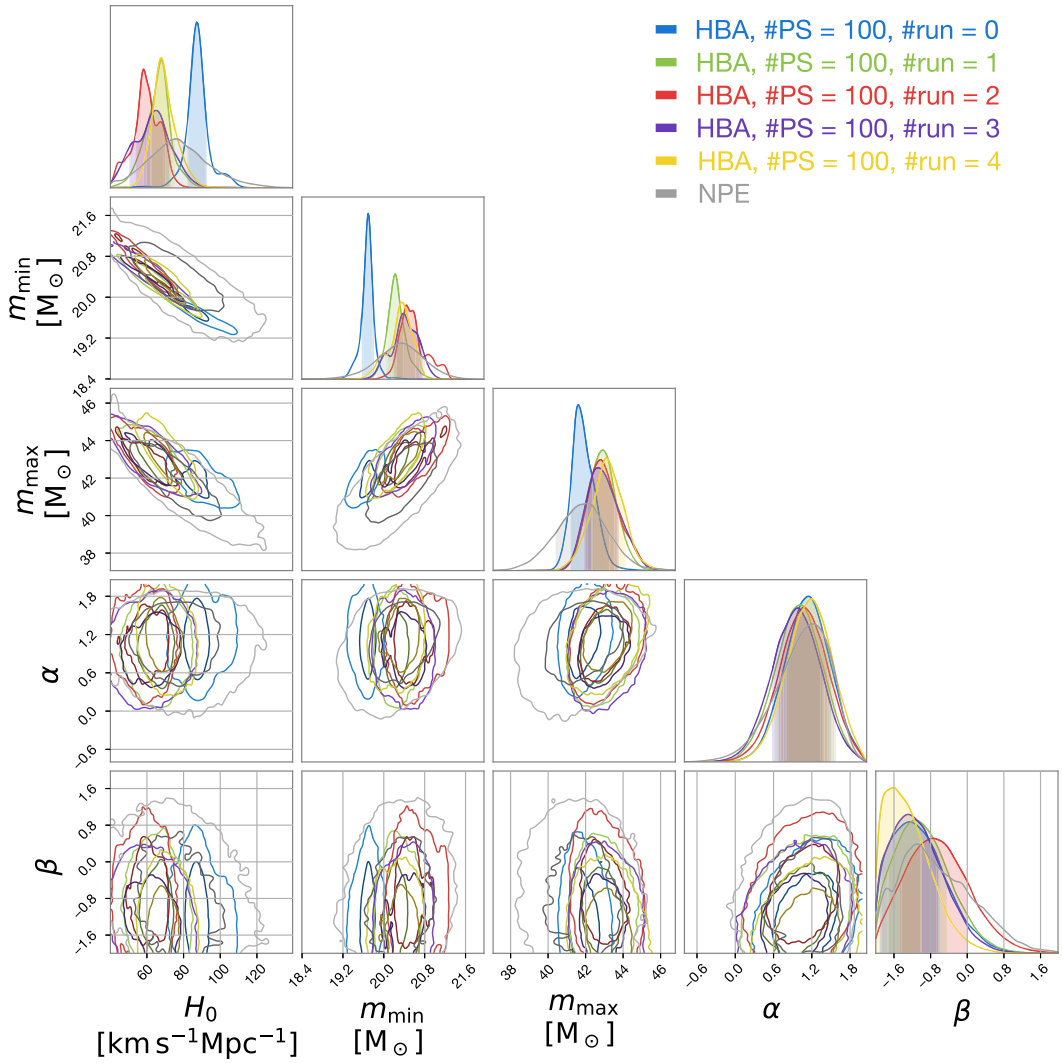

FIG. 7. Results from repeating the HBA analysis with different random sets of 100 samples from each single-event posterior. We compare this to the NPE scheme, plotted in gray. The posterior here was computed from 300 events, and the true population parameters are indicated by the dashed lines.

1. Impact of the number of posterior samples

The HBA scheme usually processes $\mathcal{O}(10^3$–$10^4)$ posterior samples per event. Since the NPE model works with a significantly lower number, this section explores the consequences of this approximation.

We carry out the HBA using 100 posterior samples per event, for 300 events in total. The population analyzed is the same as in the previous paragraph. We repeat the HBA many times, using a different set of posterior samples for each event each time. This leads to a scatter of the population posterior, as shown in Fig. 7.

For the hyperparameters with more scatter (in particular H_0 and m_{min}) the NPE differs more from the individual HBAs and gives posteriors that are broader, covering the range over which the individual HBAs vary. So, the NPE is mass covering, i.e. it has support across the scatter of the posterior that arises from the limited number of posterior samples. This indicates that the NPE marginalizes over the additional uncertainty arising from this approximation. This behavior is also expected from the construction of the loss in Eq. (14)—if the model $q(\Lambda|\mathcal{D}_K)$ has no support on the support of $p(\Lambda|\mathcal{D}_K)$, the loss diverges. As a consequence, future analyses will either have to increase the number of posterior samples or use a more complete summary of the GW signal. In this limit, we expect to find a closer agreement between these two methods.[21]

[21]Note that the works of [110,111] have explored these consequences for conventional analyses. An insufficient number of posterior samples per event was found to lead to narrow, incorrect population posteriors.

VI. CONCLUSION

Future planned detector networks will detect up to a hundred thousand GW sources each year, allowing for high-precision measurements of the parameters characterizing the population, including cosmological parameters. With these many events, fast methods, such as machine learning, will be essential for population inference and related analyses, e.g., tests of general relativity (GR).

In this work, we have demonstrated that normalizing flows can rapidly produce the posterior distribution of cosmological and population parameters inferred from observed GW dark sirens. We have introduced a loss function one has to maximize for the NF to converge to the true posterior distribution. Within this setup, the posterior learned naturally incorporates selection effects. Normalizing flows prove to be flexible enough to approximate the posterior distribution of up to 600 GW events.

However, there are instances where the results from the flow do not align with the standard results from the HBA. The work of [110,111] has shown for conventional HBAs that incorrect population posteriors can arise due to insufficient samples per Monte Carlo integral, an effect which could contribute to the observed differences. An almost perfect agreement can still be obtained by performing importance sampling on the samples output by the neural network, using the standard HBA likelihood for the weights. This process increases the computational time of our method, but still requires $\mathcal{O}(10)$ fewer likelihood evaluations than in the standard HBA approach.

The reasons for the discrepancies between HBA and our method are still unresolved. The single-event posterior samples generated by `dingo` could deviate from the true single-event posteriors, possibly leading to biases in the HBA results. Our method relies on an arbitrary data summary (provided the summary contains the majority of the signal information).[22] As long as the flow trains on this summary data, the model should recover the true hyperparameter posterior. Indeed, the network could, in principle, compensate for the eventual incorrect representation of the GW data, but we have not explicitly shown this in the present work. Moreover, we currently use 100 posterior samples per GW event which also leads to an additional uncertainty. To produce reliable posterior distributions one has to use a sufficient number of posterior samples per event, posing a potential bottleneck for analyses of 3G detector data. Indeed, future analyses might have to limit the number of posterior samples for computing efficiency, resulting in an additional uncertainty our approach can marginalize over. Alternatively, a possibility is that the model has not accurately learned the population posterior, but we have performed several tests that make this scenario unlikely.

Our approach requires us to divide the observed population into subpopulations of fixed dimensionality to use as input for the network. Errors in the learning of the posterior accumulate when combining the results of subpopulations. In practice, we find that when the model is repeatedly evaluated to combine a large number $[\mathcal{O}(10\text{–}20) \times n_{\rm sub}]$ of subpopulations of events, instabilities appear that prevent the production of an accurate posterior, i.e. GW catalogs with $N_{\rm obs}/n_{\rm sub} \gtrapprox 10$ cannot be robustly analyzed with the current framework. A more robust network architecture might be needed to learn the posterior with the necessary accuracy to analyse $\mathcal{O}(1000)$ events. We leave this for future work, as well as more complex mass and redshift distributions.

Finally, the method proposed can test population models with source frame mass distributions that are difficult to parametrize analytically since it relies on simulation-based inference, with (in principle) no explicit likelihood needed. For instance, one could include stellar evolution codes, circumventing the choice of a analytic distribution describing the source frame mass distribution.

ACKNOWLEDGMENTS

K. L. is grateful to the Fondation CFM pour la Recherche in France for support during his doctorate. We thank Eric Chassande-Mottin for comments on the manuscript. Numerical computations were performed on the DANTE platform, APC, France. Numerical computations were partly performed on the S-CAPAD/DANTE platform, IPGP, France. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation.

APPENDIX A: COUPLING FLOWS

We provide details of the deep NN in this appendix. The flow transformation of Sec. II C, $\mathbf{g}$, is given by a sequence of coupling transforms [82] that are each parametrized by monotonic rational quadratic splines [106]. As a reminder, $\mathbf{g}$ applies a coordinate transformation on $\mathbf{Z} \in \mathbb{R}^D$.

One can write the sequence of coupling transforms as

$$\mathbf{g} = \mathbf{g}_{(n_{\rm block})} \circ \mathbf{g}_{(n_{\rm block}-1)} \circ \ldots \circ \mathbf{g}_{(2)} \circ \mathbf{g}_{(1)}, \tag{A1}$$

where $n_{\rm block}$ is the *number of blocks*. Each of the functions $\mathbf{g}_{(i)}$ depends on the data, a set of NN parameters and the latent variable $\mathbf{Z}$. The $\mathbf{g}_i$ all share the same functional form

[22]The choice we made here, to represent data by physical posterior samples, θ, is not the only possibility. For instance, another approach would be to directly compress the strain data. This compressed data can then provide the input for the population inference with a NF. As such, our approach is particularly adapted to a population analysis relying on other NN summaries.

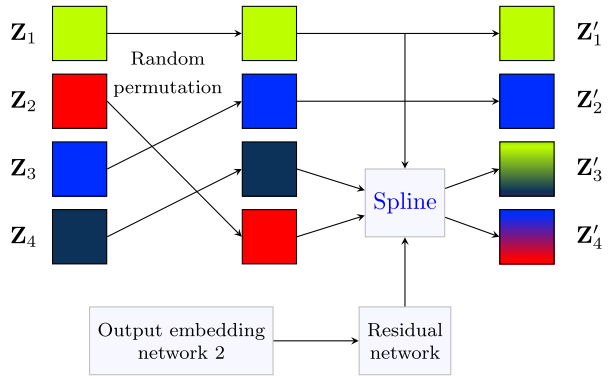

FIG. 8. Schematic overview of an elementary cell of the coupling transform on the example of a four-dimensional flow. The compressed data that is summarized by the second embedding network serves as the input data for the spline through a residual neural network.

(but have different model parameters). First, one applies a random (but different for each coupling transform, fixed during training) permutation of $\mathbf{Z}$. Then the first k components of the reshuffled variable $\tilde{\mathbf{Z}}$ are left unchanged. The remaining parameters undergo an invertible (spline) transformation $\mathbf{s}_i$ that is parametrized by the NN parameters Θ_i. One can write the transformation component-wise as [82]

$$[\mathbf{g}_{(i)}(\mathbf{Z})]_j = \begin{cases} \mathbf{Z}_j & \text{if } 1 \le j \le k, \\ [\mathbf{s}(\mathbf{Z};\Theta_{(i)})]_j & \text{if } k+1 \le j \le D, \end{cases} \tag{A2}$$

where $\Theta_{(i)}$ are the NN parameters of the ith coupling transformation. A fully connected residual network governs the spline function. Each transformation is invertible, and its inverse and Jacobian is simple to compute. Figure 8 shows schematically the transformation of one such elementary cells that make up the flow. In our case, the transformation $\mathbf{s}:\mathbb{R}^D \to \mathbb{R}^{D-j}$ corresponds to a monotonic,[23] quadratic, rational spline function [106] (see Fig. 1 of [106] for an example of a one-dimensional spline function). The parameters of this function are governed by a NN. Note that $\mathbf{s}$ depends on all variables $\mathbf{Z}$ and can hence incorporate correlations between different parameters.

APPENDIX B: LOSS IDENTITY

In this appendix, we show that the loss function proposed in Eq. (12) equals the expectation value of the KL divergence between the true posterior and the model. We proceed in two steps: we demonstrate that the four expectation values can be rewritten in terms of two expectation values. These two expectation values can then be exchanged (from Bayes's theorem), yielding the desired result.

Let us consider the following expectation value of the function $f(w,z)$

$$\mathbb{L} = \mathbb{E}_{p(w)}\mathbb{E}_{p(x|w)}\mathbb{E}_{p(y|x)}\mathbb{E}_{p(z|y)}f(w,z). \tag{B1}$$

This can be written from definition as a fourfold integration

$$\mathbb{L} = \int \mathrm{d}w\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\,p(w)p(x|w)p(y|x)p(z|y)f(w,z). \tag{B2}$$

Making the additional assumption that y and w are conditionally independent given x, i.e.,

$$p(y,w|x) = p(y|x)p(w|x) \;\Leftrightarrow\; p(y|x,w) = p(y|x), \tag{B3}$$

and similarly that z and (x,w) are conditionally independent given y, so that $p(z|y) = p(z|y,x,w)$, and applying the law of conditional probability $p(s|t)p(t) = p(s,t)$, the above expression can be rewritten as

$$\mathbb{L} = \int \mathrm{d}w\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\,p(w,x,y,z)f(w,z). \tag{B4}$$

Since the function f is independent of the random variables x, y, we can perform the integration

$$\mathbb{L} = \int \mathrm{d}w\,\mathrm{d}z\,p(w)p(z|w)f(w,z). \tag{B5}$$

After applying Bayes's theorem and changing the order of the integration, we obtain

$$\mathbb{L} = \int \mathrm{d}w\,\mathrm{d}z\,p(z)p(w|z)f(w,z) = \mathbb{E}_{p(z)}\mathbb{E}_{p(w|z)}f(w,z). \tag{B6}$$

The identity claimed in Sec. II B 1 of Eqs. (12) and (14) can be obtained for $w=\Lambda$, $x=\theta_K$, $y=\mathcal{D}_K$, $z=\hat{\theta}_K$, $f = -\log[q(\Lambda|\hat{\theta}_K)]$ and $p(w|z)$ is the target distribution, namely the population posterior given a set of GW events $p(\Lambda|\{\mathcal{D}_k\}_{k\in K})$. The conditional independence conditions reduce to assuming $p(\hat{\theta}_K,\Lambda,\theta_K|\mathcal{D}_K) = p(\hat{\theta}_K|\mathcal{D}_K)$ $p(\Lambda,\theta_K|\mathcal{D}_K)$, which holds because the distribution of posterior samples depends only on the observed data, and $p(\mathcal{D}_K,\Lambda|\theta_K) = p(\mathcal{D}_K|\theta_K)p(\Lambda|\theta_K)$, which holds because the observed data depends only on the parameters of the sources in the data.

APPENDIX C: POPULATION DETAILS

In the results section we analyze 13 different populations, each of which has a different underlying set of hyperparameters. Table VI lists the value of all parameters for each population.

[23]If the spline was not monotonic the function would be not invertible, making it unsuitable for NFs.

TABLE VI. The hyperparameters for the fourteen populations considered.

Λ	H_0	$m_{\min}$	$m_{\max}$	α	β
	Units				
Number of Pop	$\mathrm{km\,s^{-1} \times Mpc^{-1}}$	$M_\odot$	$M_\odot$	...	...
0	60.0	22.0	46.0	0.0	0.0
1	50.0	21.0	44.0	0.5	0.5
2	55.0	23.0	42.0	1.0	−0.5
3	100.0	18.5	46.0	1.0	0.0
4	50.0	24.0	45.0	1.0	0.2
5	90.0	23.0	44.0	0.4	0.0
6	80.0	24.0	43.0	−0.3	−0.7
7	45.0	19.0	38.0	1.5	0.5
8	135.0	19.0	46.0	1.5	−0.5
9	60.0	19.0	45.0	−1.6	0.0
10	70.0	22.0	43.0	−0.4	0.4
11	80.0	19.4	42.3	0.6	−0.3
12	67.0	20.1	42.9	0.6	−0.5
13	70.0	21.0	43.3	0.8	0.3

[1] J. Aasi *et al.*, Advanced LIGO, Classical Quantum Gravity **32**, 074001 (2015).

[2] A. Buikema *et al.* (aLIGO Collaboration), Sensitivity and performance of the Advanced LIGO detectors in the third observing run, Phys. Rev. D **102**, 062003 (2020).

[3] M. Tse *et al.*, Quantum-enhanced Advanced LIGO detectors in the era of gravitational-wave astronomy, Phys. Rev. Lett. **123**, 231107 (2019).

[4] F. Acernese *et al.* (Virgo Collaboration), Advanced Virgo: A second-generation interferometric gravitational wave detector, Classical Quantum Gravity **32**, 024001 (2015).

[5] F. Acernese *et al.* (Virgo Collaboration), Increasing the astrophysical reach of the Advanced Virgo detector via the application of squeezed vacuum states of light, Phys. Rev. Lett. **123**, 231108 (2019).

[6] T. Akutsu *et al.* (KAGRA Collaboration), Overview of KAGRA: Detector design and construction history, Prog. Theor. Exp. Phys. **2021**, 05A101 (2021).

[7] Y. Aso, Y. Michimura, K. Somiya, M. Ando, O. Miyakawa, T. Sekiguchi, D. Tatsumi, and H. Yamamoto (KAGRA Collaboration), Interferometer design of the KAGRA gravitational wave detector, Phys. Rev. D **88**, 043007 (2013).

[8] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, Phys. Rev. X **13**, 041039 (2023).

[9] M. Fishbach, D. E. Holz, and B. Farr, Are LIGO's black holes made from smaller black holes?, Astrophys. J. Lett. **840**, L24 (2017).

[10] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Binary black hole population properties inferred from the first and second observing runs of Advanced LIGO and Advanced Virgo, Astrophys. J. Lett. **882**, L24 (2019).

[11] M. Isi, K. Chatziioannou, and W. M. Farr, Hierarchical test of general relativity with gravitational waves, Phys. Rev. Lett. **123**, 121101 (2019).

[12] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Population properties of compact objects from the second LIGO-Virgo gravitational-wave transient catalog, Astrophys. J. Lett. **913**, L7 (2021).

[13] I. M. Romero-Shaw, P. D. Lasky, E. Thrane, and J. C. Bustillo, GW190521: Orbital eccentricity and signatures of dynamical formation in a binary black hole merger signal, Astrophys. J. Lett. **903**, L5 (2020).

[14] I. M. Romero-Shaw, N. Farrow, S. Stevenson, E. Thrane, and X.-J. Zhu, On the origin of GW190425, Mon. Not. R. Astron. Soc. **496**, L64 (2020).

[15] V. Tiwari and S. Fairhurst, The emergence of structure in the binary black hole mass distribution, Astrophys. J. Lett. **913**, L19 (2021).

[16] R. Abbott *et al.* (KAGRA, Virgo, and LIGO Scientific Collaborations), Population of merging compact binaries

inferred using gravitational waves through GWTC-3, Phys. Rev. X **13**, 011048 (2023).
[17] B. Edelman, Z. Doctor, J. Godfrey, and B. Farr, Ain't no mountain high enough: Semiparametric modeling of LIGO–Virgo's binary black hole mass distribution, Astrophys. J. **924**, 101 (2022).
[18] C. Hoy, S. Fairhurst, M. Hannam, and V. Tiwari, Understanding how fast black holes spin by analyzing data from the second gravitational-wave catalogue, Astrophys. J. **928**, 75 (2022).
[19] S. Biscoveanu, T. A. Callister, C.-J. Haster, K. K. Y. Ng, S. Vitale, and W. M. Farr, The binary black hole spin distribution likely broadens with redshift, Astrophys. J. Lett. **932**, L19 (2022).
[20] G. Ashton, S. Thiele, Y. Lecoeuche, J. McIver, and L. K. Nuttall, Parameterised population models of transient non-Gaussian noise in the LIGO gravitational-wave detectors, Classical Quantum Gravity **39**, 175004 (2022).
[21] S. S. Bavera, M. Fishbach, M. Zevin, E. Zapartas, and T. Fragos, The χeff− z correlation of field binary black hole mergers and how 3G gravitational-wave detectors can constrain it, Astron. Astrophys. **665**, A59 (2022).
[22] C. Karathanasis, S. Mukherjee, and S. Mastrogiovanni, Binary black holes population and cosmology in new lights: Signature of PISN mass and formation channel in GWTC-3, Mon. Not. R. Astron. Soc. **523**, 4539 (2023).
[23] M. Fishbach, C. Kimball, and V. Kalogera, Limits on hierarchical black hole mergers from the most negative $\chi_{\rm eff}$ systems, Astrophys. J. Lett. **935**, L26 (2022).
[24] S. Mastrogiovanni, A. Lamberts, R. Srinivasan, T. Bruel, and N. Christensen, Inferring binary black holes stellar progenitors with gravitational wave sources, Mon. Not. R. Astron. Soc. **517**, 3432 (2022).
[25] C. Ye and M. Fishbach, Inferring the neutron star maximum mass and lower mass gap in neutron star–black hole systems with spin, Astrophys. J. **937**, 73 (2022).
[26] F. Antonini, M. Gieles, F. Dosopoulou, and D. Chattopadhyay, Coalescing black hole binaries from globular clusters: Mass distributions and comparison to gravitational wave data from GWTC-3, Mon. Not. R. Astron. Soc. **522**, 466 (2023).
[27] C. Adamcewicz and E. Thrane, Do unequal-mass binary black hole systems have larger χeff? Probing correlations with copulas in gravitational-wave astronomy, Mon. Not. R. Astron. Soc. **517**, 3928 (2022).
[28] A. M. Farah, B. Edelman, M. Zevin, M. Fishbach, J. M. Ezquiaga, B. Farr, and D. E. Holz, Things that might go bump in the night: Assessing structure in the binary black hole mass spectrum, Astrophys. J. **955**, 107 (2023).
[29] T. A. Callister and W. M. Farr, A parameter-free tour of the binary black hole population, arXiv:2302.07289.
[30] M. Fishbach and G. Fragione, Globular cluster formation histories, masses, and radii inferred from gravitational waves, Mon. Not. R. Astron. Soc. **522**, 5546 (2023).
[31] M. Mould, D. Gerosa, M. Dall'Amico, and M. Mapelli, One to many: Comparing single gravitational-wave events to astrophysical populations, Mon. Not. R. Astron. Soc. **525**, 3986 (2023).
[32] J. Sadiq, T. Dent, and M. Gieles, Binary vision: The merging black hole binary mass distribution via iterative density estimation, Astrophys. J. **960**, 65 (2024).
[33] S. Rinaldi, W. Del Pozzo, M. Mapelli, A. L. Medina, and T. Dent, Evidence for the evolution of black hole mass function with redshift, arXiv:2310.03074.
[34] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Tests of general relativity with the binary black hole signals from the LIGO-Virgo catalog GWTC-1, Phys. Rev. D **100**, 104036 (2019).
[35] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Tests of general relativity with binary black holes from the second LIGO-Virgo gravitational-wave transient catalog, Phys. Rev. D **103**, 122002 (2021).
[36] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), Tests of general relativity with GWTC-3, arXiv:2112.06861.
[37] I. Magana Hernandez, Constraining the number of spacetime dimensions from GWTC-3 binary black hole mergers, Phys. Rev. D **107**, 084033 (2023).
[38] J. M. Ezquiaga, Hearing gravity from the cosmos: GWTC-2 probes general relativity at cosmological scales, Phys. Lett. B **822**, 136665 (2021).
[39] M. Mancarella, E. Genoud-Prachex, and M. Maggiore, Cosmology and modified gravitational wave propagation from binary black hole population models, Phys. Rev. D **105**, 064030 (2022).
[40] K. Leyde, S. Mastrogiovanni, D. A. Steer, E. Chassande-Mottin, and C. Karathanasis, Current and future constraints on cosmology and modified gravitational wave friction from binary black holes, J. Cosmol. Astropart. Phys. 09 (2022) 012.
[41] M. Isi, W. M. Farr, and K. Chatziioannou, Comparing Bayes factors and hierarchical inference for testing general relativity with gravitational waves, Phys. Rev. D **106**, 024048 (2022).
[42] R. Niu, T. Zhu, and W. Zhao, Testing Lorentz invariance of gravity in the standard-model extension with GWTC-3, J. Cosmol. Astropart. Phys. 12 (2022) 011.
[43] A. Chen, R. Gray, and T. Baker, Testing the nature of gravitational wave propagation using dark sirens and galaxy catalogues, arXiv:2309.03833.
[44] A. Ray, P. Fan, V. F. He, M. Bloom, S. M. Yang, J. D. Tasson, and J. D. E. Creighton, Measuring gravitational wave speed and Lorentz violation with the first three gravitational-wave catalogs, arXiv:2307.13099.
[45] S. Mastrogiovanni, D. Laghi, R. Gray, G. C. Santoro, A. Ghosh, C. Karathanasis, K. Leyde, D. A. Steer, S. Perries, and G. Pierra, A novel approach to infer population and cosmological properties with gravitational waves standard sirens and galaxy surveys, Phys. Rev. D **108**, 042002 (2023).
[46] R. Magee, M. Isi, E. Payne, K. Chatziioannou, W. M. Farr, G. Pratten, and S. Vitale, The impact of selection biases on tests of general relativity with gravitational-wave inspirals, Phys. Rev. D **109**, 023014 (2024).
[47] E. Payne, M. Isi, K. Chatziioannou, and W. M. Farr, Fortifying gravitational-wave tests of general relativity against astrophysical assumptions, Phys. Rev. D **108**, 124060 (2023).

[48] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW170817: Observation of gravitational waves from a binary neutron star inspiral, Phys. Rev. Lett. **119**, 161101 (2017).
[49] B. P. Abbott *et al.* (LIGO Scientific, Virgo, 1M2H, Dark Energy Camera GW-E, DES, DLT40, Las Cumbres Observatory, VINROUGE, MASTER Collaborations), A gravitational-wave standard siren measurement of the Hubble constant, Nature (London) **551**, 85 (2017).
[50] B. F. Schutz, Determining the Hubble constant from gravitational wave observations, Nature (London) **323**, 310 (1986).
[51] W. Del Pozzo, Inference of the cosmological parameters from gravitational waves: Application to second generation interferometers, Phys. Rev. D **86**, 043011 (2012).
[52] R. Gray *et al.*, Cosmological inference using gravitational wave standard sirens: A mock data analysis, Phys. Rev. D **101**, 122001 (2020).
[53] R. Gray, C. Messenger, and J. Veitch, A pixelated approach to galaxy catalogue incompleteness: Improving the dark siren measurement of the Hubble constant, Mon. Not. R. Astron. Soc. **512**, 1127 (2022).
[54] A. Finke, S. Foffa, F. Iacovelli, M. Maggiore, and M. Mancarella, Cosmology with LIGO/Virgo dark sirens: Hubble parameter and modified gravitational wave propagation, J. Cosmol. Astropart. Phys. 08 (2021) 026.
[55] C. Turski, M. Bilicki, G. Dálya, R. Gray, and A. Ghosh, Impact of modelling galaxy redshift uncertainties on the gravitational-wave dark standard siren measurement of the Hubble constant, Mon. Not. R. Astron. Soc. **526**, 6224 (2023).
[56] R. Gray *et al.*, Joint cosmological and gravitational-wave population inference using dark sirens and galaxy catalogues, J. Cosmol. Astropart. Phys. 12 (2023) 023.
[57] S. R. Taylor, J. R. Gair, and I. Mandel, Hubble without the Hubble: Cosmology using advanced gravitational-wave detectors alone, Phys. Rev. D **85**, 023535 (2012).
[58] S. R. Taylor and J. R. Gair, Cosmology with the lights off: Standard sirens in the Einstein Telescope era, Phys. Rev. D **86**, 023502 (2012).
[59] W. M. Farr, M. Fishbach, J. Ye, and D. Holz, A future percent-level measurement of the Hubble expansion at redshift 0.8 with Advanced LIGO, Astrophys. J. Lett. **883**, L42 (2019).
[60] S. Mastrogiovanni, K. Leyde, C. Karathanasis, E. Chassande-Mottin, D. A. Steer, J. Gair, A. Ghosh, R. Gray, S. Mukherjee, and S. Rinaldi, On the importance of source population models for gravitational-wave cosmology, Phys. Rev. D **104**, 062009 (2021).
[61] J. M. Ezquiaga and D. E. Holz, Spectral sirens: Cosmology from the full mass distribution of compact binaries, Phys. Rev. Lett. **129**, 061102 (2022).
[62] N. Aghanim *et al.* (Planck Collaboration), Planck 2018 results. VI. Cosmological parameters, Astron. Astrophys. **641**, A6 (2020); **652**, C4(E) (2021).
[63] A. G. Riess, S. Casertano, W. Yuan, L. M. Macri, and D. Scolnic, Large magellanic cloud cepheid standards provide a 1% foundation for the determination of the Hubble constant and stronger evidence for physics beyond ΛCDM, Astrophys. J. **876**, 85 (2019).
[64] H.-Y. Chen, M. Fishbach, and D. E. Holz, A two per cent Hubble constant measurement from standard sirens within five years, Nature (London) **562**, 545 (2018).
[65] E. Belgacem, Y. Dirian, S. Foffa, E. J. Howell, M. Maggiore, and T. Regimbau, Cosmology and dark energy from joint gravitational wave-GRB observations, J. Cosmol. Astropart. Phys. 08 (2019) 015.
[66] J. Yu, Y. Wang, W. Zhao, and Y. Lu, Hunting for the host galaxy groups of binary black holes and the application in constraining Hubble constant, Mon. Not. R. Astron. Soc. **498**, 1786 (2020).
[67] S. Borhanian, A. Dhani, A. Gupta, K. G. Arun, and B. S. Sathyaprakash, Dark sirens to resolve the Hubble–Lemaître tension, Astrophys. J. Lett. **905**, L28 (2020).
[68] J.-Y. Song, L.-F. Wang, Y. Li, Z.-W. Zhao, J.-F. Zhang, W. Zhao, and X. Zhang, Synergy between CSST galaxy survey and gravitational-wave observation: Inferring the Hubble constant from dark standard sirens, China Phys. Mech. Astron. **67**, 230411 (2024).
[69] I. Gupta, Using gray sirens to resolve the Hubble-Lemaître tension, Mon. Not. R. Astron. Soc. **524**, 3537 (2023).
[70] L.-G. Zhu and X. Chen, The dark side of using dark sirens to constrain the Hubble-Lemaître constant, Astrophys. J. **948**, 26 (2023).
[71] N. Muttoni, D. Laghi, N. Tamanini, S. Marsat, and D. Izquierdo-Villalba, Dark siren cosmology with binary black holes in the era of third-generation gravitational wave detectors, Phys. Rev. D **108**, 043543 (2023).
[72] Z.-Q. You, X.-J. Zhu, G. Ashton, E. Thrane, and Z.-H. Zhu, Standard-siren cosmology using gravitational waves from binary black holes, Astrophys. J. **908**, 215 (2021).
[73] H. Leandro, V. Marra, and R. Sturani, Measuring the Hubble constant with black sirens, Phys. Rev. D **105**, 023523 (2022).
[74] C. Talbot and E. Thrane, Flexible and accurate evaluation of gravitational-wave Malmquist bias with machine learning, Astrophys. J. **927**, 76 (2022).
[75] K. W. K. Wong, G. Contardo, and S. Ho, Gravitational wave population inference with deep flow-based generative network, Phys. Rev. D **101**, 123005 (2020).
[76] F. Gerardi, S. M. Feeney, and J. Alsing, Unbiased likelihood-free inference of the Hubble constant from light standard sirens, Phys. Rev. D **104**, 083531 (2021).
[77] M. Mould, D. Gerosa, and S. R. Taylor, Deep learning and Bayesian inference of gravitational-wave populations: Hierarchical black-hole mergers, Phys. Rev. D **106**, 103013 (2022).
[78] D. Ruhe, K. Wong, M. Cranmer, and P. Forré, Normalizing flows for hierarchical Bayesian analysis: A gravitational wave population study, arXiv:2211.09008.
[79] M. Dax, S. R. Green, J. Gair, M. Deistler, B. Schölkopf, and J. H. Macke, Group equivariant neural posterior estimation, in *International Conference on Learning Representations* (2022), arXiv:2111.13139.
[80] E. G. Tabak and E. Vanden-Eijnden, Density estimation by dual ascent of the log-likelihood, Commun. Math. Sci. **8**, 217 (2010).

[81] E. G. Tabak and C. V. Turner, A family of nonparametric density estimation algorithms, Commun. Pure Appl. Math. **66**, 145 (2013).
[82] L. Dinh, D. Krueger, and Y. Bengio, NICE: Non-linear independent components estimation, arXiv:1410.8516.
[83] D. Jimenez Rezende and S. Mohamed, Variational inference with normalizing flows, arXiv:1505.05770.
[84] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3964 (2021).
[85] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Equivariant flow-based sampling for lattice gauge theory, Phys. Rev. Lett. **125**, 121601 (2020).
[86] F. Noé, S. Olsson, J. Köhler, and H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning, Science **365**, eaaw1147 (2019).
[87] P. Wirnsberger, A. J. Ballard, G. Papamakarios, S. Abercrombie, S. Racanière, A. Pritzel, D. Jimenez Rezende, and C. Blundell, Targeted free energy estimation via learned mappings, J. Chem. Phys. **153**, 144112 (2020).
[88] J. Köhler, L. Klein, and F. Noé, Equivariant flows: Sampling configurations for multi-body systems with symmetric energies, arXiv:1910.00753.
[89] S. R. Green, C. Simpson, and J. Gair, Gravitational-wave parameter estimation with autoregressive neural network flows, Phys. Rev. D **102**, 104057 (2020).
[90] S. R. Green and J. Gair, Complete parameter inference for GW150914 using deep learning, Mach. Learn. Sci. Tech. **2**, 03LT01 (2021).
[91] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Real-time gravitational wave science with neural posterior estimation, Phys. Rev. Lett. **127**, 241103 (2021).
[92] R. Abbott *et al.* (LIGO Scientific, Virgo, KAGRA, and Virgo Collaborations), Constraints on the cosmic expansion history from GWTC–3, Astrophys. J. **949**, 76 (2023).
[93] J. M. Ezquiaga and M. Zumalacárregui, Dark energy in light of multi-messenger gravitational-wave astronomy, Front. Astron. Space Sci. **5**, 44 (2018).
[94] J. M. Ezquiaga and D. E. Holz, Jumping the gap: Searching for LIGO's biggest black holes, Astrophys. J. Lett. **909**, L23 (2021).
[95] M. Corman, A. Ghosh, C. Escamilla-Rivera, M. A. Hendry, S. Marsat, and N. Tamanini, Constraining cosmological extra dimensions with gravitational wave standard sirens: From theory to current and future multi-messenger observations, Phys. Rev. D **105**, 064061 (2022).
[96] J. M. Ezquiaga, Hearing gravity from the cosmos: GWTC-2 probes general relativity at cosmological scales, Phys. Lett. B **822**, 136665 (2021).
[97] I. Magana Hernandez, Constraining the number of spacetime dimensions from GWTC-3 binary black hole mergers, Phys. Rev. D **107**, 084033 (2023).
[98] I. Mandel, W. M. Farr, and J. R. Gair, Extracting distribution parameters from multiple uncertain observations with selection biases, Mon. Not. R. Astron. Soc. **486**, 1086 (2019).
[99] E. Thrane and C. Talbot, An introduction to Bayesian inference in gravitational-wave astronomy: Parameter estimation, model selection, and hierarchical models, Pub. Astron. Soc. Aust. **36**, e010 (2019).
[100] S. Vitale, D. Gerosa, W. M. Farr, and S. R. Taylor, Inferring the properties of a population of compact binaries in presence of selection effects, *Handbook of Gravitational Wave Astronomy* (Springer, Singapore, 2020).
[101] M. Dax, S. R. Green, J. Gair, M. Pürrer, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, Neural importance sampling for rapid and reliable gravitational-wave inference, Phys. Rev. Lett. **130**, 171403 (2023).
[102] S. Kullback and R. A. Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79 (1951).
[103] D. Wysocki, R. O'Shaughnessy, L. Wade, and J. Lange, Inferring the neutron star equation of state simultaneously with the population of merging neutron stars, arXiv:2001.01747.
[104] J. Golomb and C. Talbot, Hierarchical inference of binary neutron star mass distribution and equation of state with gravitational waves, Astrophys. J. **926**, 79 (2022).
[105] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, nflows: Normalizing flows in PyTorch, Zenodo (2020), https://doi.org/10.5281/zenodo.4296287.
[106] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Neural spline flows, in *Advances in Neural Information Processing Systems, Vancouver, Canada*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32.
[107] G. Pratten *et al.*, Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes, Phys. Rev. D **103**, 104056 (2021).
[108] I. M. Romero-Shaw *et al.*, Bayesian inference for compact binary coalescences with bilby: Validation and application to the first LIGO–Virgo gravitational-wave transient catalogue, Mon. Not. R. Astron. Soc. **499**, 3295 (2020).
[109] B. Edelman, B. Farr, and Z. Doctor, Cover your basis: Comprehensive data-driven characterization of the binary black hole population, Astrophys. J. **946**, 16 (2023).
[110] J. Golomb and C. Talbot, Searching for structure in the binary black hole spin distribution, Phys. Rev. D **108**, 103009 (2023).
[111] C. Talbot and J. Golomb, Growing pains: Understanding the impact of likelihood uncertainty on hierarchical Bayesian inference for gravitational-wave astronomy, Mon. Not. R. Astron. Soc. **526**, 3495 (2023).