Unbiased likelihood-free inference of the Hubble constant from light standard sirens

Francesca Gerardi,[1,*] Stephen M. Feeney,[1] and Justin Alsing[2,3]

[1]*Department of Physics & Astronomy, University College London,*
Gower Street, London WC1E 6BT, United Kingdom

[2]*Oskar Klein Centre for Cosmoparticle Physics, Department of Physics, Stockholm University,*
Stockholm SE-106 91, Sweden

[3]*Imperial Centre for Inference and Cosmology, Astrophysics Group, Imperial College London,*
Blackett Laboratory, Prince Consort Road, London SW7 2AZ, United Kingdom

(Received 15 April 2021; accepted 16 September 2021; published 22 October 2021)

Multimessenger observations of binary neutron star mergers offer a promising path toward resolution of the Hubble constant (H_0) tension, provided their constraints are shown to be free from systematics such as the Malmquist bias. In the traditional Bayesian framework, accounting for selection effects in the likelihood requires calculation of the expected number (or fraction) of detections as a function of the parameters describing the population and cosmology; a potentially costly and/or inaccurate process. This calculation can, however, be bypassed completely by performing the inference in a framework in which the likelihood is never explicitly calculated, but instead fit using forward simulations of the data, which naturally include the selection. This is likelihood-free inference (LFI). Here, we use density-estimation LFI, coupled to neural-network-based data compression, to infer H_0 from mock catalogues of binary neutron star mergers, given noisy redshift, distance and peculiar velocity estimates for each object. We demonstrate that LFI yields statistically unbiased estimates of H_0 in the presence of selection effects, with precision matching that of sampling the full Bayesian hierarchical model. Marginalizing over the bias increases the H_0 uncertainty by only 6% for training sets consisting of $O(10^4)$ populations. The resulting LFI framework is applicable to population-level inference problems with selection effects across astrophysics.

DOI: 10.1103/PhysRevD.104.083531

I. INTRODUCTION

In recent years, late-time measurements [1–3] of the Hubble constant, H_0, have diverged from estimates provided by early-time probes [4–7] (see Refs. [8–10] for a summary). At the heart of the discrepancy is a 4.2σ tension between the latest direct measurement of $H_0 = (73.2 \pm 1.3)\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ by the SH0ES Team's Cepheid-supernova distance ladder [1] and the model-dependent value of $H_0 = (67.4 \pm 0.5)\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ inferred from observations of the cosmic microwave background (CMB) anisotropies by the Planck satellite [4]. While unforeseen systematic effects [11–16] might be the cause of this disagreement, it is possible that this is a hint for new physics beyond the standard ΛCDM model (see Ref. [17] for a comprehensive summary of potential theoretical solutions). Despite considerable effort, however, no consensus on an explanation has been reached. This strongly motivates the need for a new, independent, direct probe of H_0. Gravitational waves (GWs) emitted by compact-object mergers—so-called *standard sirens*—are very promising in this regard [18–32], since their amplitude provides a self-calibrated estimate of the luminosity distance, d, depending only on general relativity.

There are three types of compact-object systems typically considered for H_0 studies [33]: binary black holes (BBH), binary neutron stars (BNS) and neutron star-black hole (NSBH) systems. The potential for BNS and NSBH systems to have electromagnetic (EM) counterparts makes them particularly promising, as if an EM counterpart can be detected, the merger's host galaxy can be identified and its redshift measured, yielding H_0 when combined with d [20,21,24,27–29,34,35]. The first BNS system detected by the LIGO-Virgo Consortium, GW170817 [36], also produced an EM counterpart [37], constraining H_0 to $70.0^{+12.0}_{-8.0}\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ [35]. The 10% constraints produced by this single event are expected to shrink to ∼1% in the next 5–10 years once $O(100)$ events have been observed [28,29,34].

For standard siren estimates of H_0 to resolve the current tension, they must be shown to be free from systematic errors. Standard siren datasets suffer from Malmquist bias [38,39] which, left untreated, results in H_0 being overestimated. Traditional Bayesian methods must therefore take

[*] francesca.gerardi.19@ucl.ac.uk

this effect into account by including in the likelihood terms involving the number (or, equivalently, fraction) of mergers that are expected to be detected given a set of population and cosmological parameters, $\bar{N}(\mathbf{\Omega})$ [32,35,40–42]. The simplest method for calculating the expected number of detections is through Monte Carlo integration, i.e., repeated simulations of the dataset. Implementing this directly within a posterior sampling algorithm is, however, completely unfeasible, given the sheer number of simulations that would be needed. Instead, a single large catalogue of detected mergers can be generated using a fiducial set of population parameters and then reweighted to approximate $\bar{N}$ for any value of population parameters sampled [43]. If the distribution of object parameters changes rapidly as a function of population parameters, however, a large (potentially computationally unfeasible) number of fiducial-population simulations are required to guarantee there are enough non-zero weights for the estimate of $\bar{N}$ to be reliable (the *effective* number of detected mergers must be at least four times the measured number [44]). Alternatively, $\bar{N}$ can be evaluated on a grid of $\mathbf{\Omega}$ and interpolated to generic population parameters [31,42]. While no reweighting is necessary in this case, the dependence on gridded computations means this method scales very poorly with parameter-set dimensionality.

Recently, Ref. [45] proposed a machine-learning based approach to this problem. The authors use a Gaussian mixture model to fit the distribution of object parameters found using a set of detected mergers drawn from a fiducial population. By dividing out the prior on the object parameters for the fiducial population, they obtain an estimate of the probability of detecting a merger given its parameters. This estimate can be combined with the prior on the object parameters for a generic population to calculate $\bar{N}$ at any point sampled, either directly or via a neural-network-based interpolation. This approach suffers less bias than the reweighting method due to the assumption of a fiducial population, and comes at a cost of only $O(1000)$ simulated populations. However, the estimate of the detection probability as a function of object parameters is only defined over the range of parameters supported by the fiducial population; should this range change rapidly with the population parameters, the method's $\bar{N}$ estimates will lose accuracy.

Here, we take a different approach, demonstrating that the computation of $\bar{N}$ can be completely bypassed using likelihood-free inference (LFI), which requires no analytic knowledge of the likelihood function. Specifically, we use density-estimation LFI (DELFI) [46–49], in which the distribution of data as a function of the parameters that generated them is fit by supplying density estimators with a training set of simulated datasets. This fit is then used as a proxy likelihood to obtain posteriors on the parameters of interest. As the simulated data include the selection function, LFI automatically accounts for the Malmquist bias.

LFI's ability to accelerate the inference of the properties of individual BBH mergers has been demonstrated in a number of recent works [50–56]. Here, we apply LFI to population-level inference, taking as our example the inference of H_0 from 100 simulated GW-selected BNS mergers with EM counterparts. In this particular setting, traditional Bayesian inference (with $\bar{N}$ interpolated from a grid of cosmological values [42]) is feasible, and we take this approach as a ground truth from which we can robustly quantify any systematic errors introduced by LFI. We take as our inputs sets of individual mergers' observed redshifts, distances (generated via traditional [36] or likelihood-free analyses e.g., [55]) and peculiar velocities, performing our LFI analysis with the aid of pydelfi [49]. While we concentrate here on the inference of H_0 from BNS, the technique is applicable to population studies in general e.g., [57,58].

We describe the hierarchical model we use to simulate our BNS mergers in Sec. II, and explain our inference method in Sec. III, highlighting the importance of data compression. Results are discussed in Sec. IV, and conclusions are drawn in Sec. V.

II. SIMULATIONS

In this work we assume we possess noisy estimates of redshift $\hat{z}$, distance $\hat{d}$ and peculiar velocity $\hat{v}$ for each BNS merger. The mergers' $[\hat{\mathbf{z}}, \hat{\mathbf{d}}, \hat{\mathbf{v}}]$ are generated via the hierarchical model in Fig. 1, which is loosely based on the model used in Ref. [42]. We assume that the strain data have been precompressed into estimates of $\hat{\mathbf{d}}$, which can be done rapidly using the likelihood-free method of Ref. [55]. Given the aforementioned prospects for solving the H_0 tension, we fix the number of mergers to $N = 100$. We consider two test cases, both assuming the same set of observables, but distinguished by whether GW selection is applied. Considering these two cases allows us to differentiate the impact of LFI alone from LFI specifically in the presence of selection effects.

In the following we wish to infer two cosmological parameters—the Hubble constant, H_0, and the deceleration parameter, q_0—which we denote by $\mathbf{\Omega} = [H_0, q_0]$. For a given choice of $\mathbf{\Omega}$, true redshifts are randomly sampled from

$$\begin{aligned} P(z_i|\mathbf{\Omega}, z_{\max}) &= \frac{1}{(1+z_i)} \frac{dV}{dz}(\mathbf{\Omega}) \mathcal{H}(z_{\max} - z_i) \\ &\simeq \frac{4\pi}{(1+z_i)} \frac{c^3 z^2}{H_0^3} [1 - 2(1+q_0) z_i] \mathcal{H}(z_{\max} - z_i), \end{aligned} \tag{1}$$

where $\mathcal{H}$ is a Heaviside step function. The final line is a good approximation for $z_{\max} \ll 1$. Given a single cosmological redshift draw, the ith distance is given by [59]

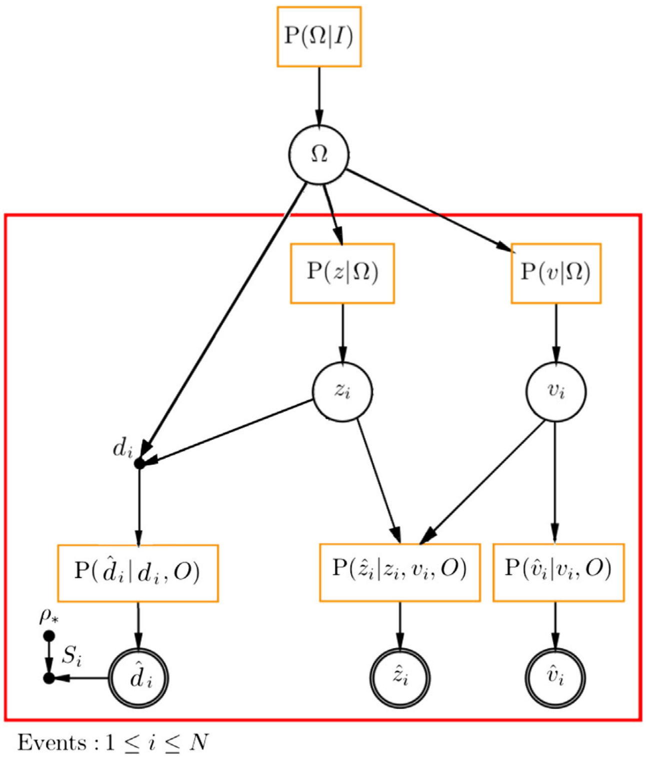

FIG. 1. The hierarchical model used to describe our BNS population and data, adapted from Ref. [42]. Read top-to-bottom, parameters (circles) are drawn from probability distributions (orange rectangles) to generate observed quantities (double circles). I represents the prior information assumed about the cosmological parameters, $\boldsymbol{\Omega} = [H_0, q_0]$, and quantities within the red plate are specific to an individual merger.

$$d_i(z_i, H_0, q_0) = \frac{cz_i}{H_0}\left[1 + \frac{1}{2}(1 - q_0)z_i\right]. \tag{2}$$

Denoting as $\mathcal{N}(\mu, \sigma)$ the normal distribution of mean μ and standard deviation σ, peculiar velocities are sampled from

$$\begin{aligned} P(v_i) &= \mathcal{N}(\mu_{v_\parallel}, \sigma_{v_\parallel}) \\ &= \mathcal{N}(0\ \mathrm{km\,s^{-1}}, 500\ \mathrm{km\,s^{-1}}). \end{aligned} \tag{3}$$

We convert our true redshifts, distances and peculiar velocities into observed quantities $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{z}}, \hat{\boldsymbol{d}}, \hat{\boldsymbol{v}}]$ assuming, for simplicity, the marginal likelihoods are Gaussian,[1] as follows

$$P(\hat{z}|z, v) = \mathcal{N}(z + v/c, \sigma_{\hat{z}} = 1.2 \times 10^{-3}) \tag{4}$$

$$P(\hat{d}|d) = \mathcal{N}(d, \sigma_{\hat{d}} = d/10) \tag{5}$$

$$P(\hat{v}|v) = \mathcal{N}(v, \sigma_{\hat{v}} = 200\ \mathrm{km\,s^{-1}}). \tag{6}$$

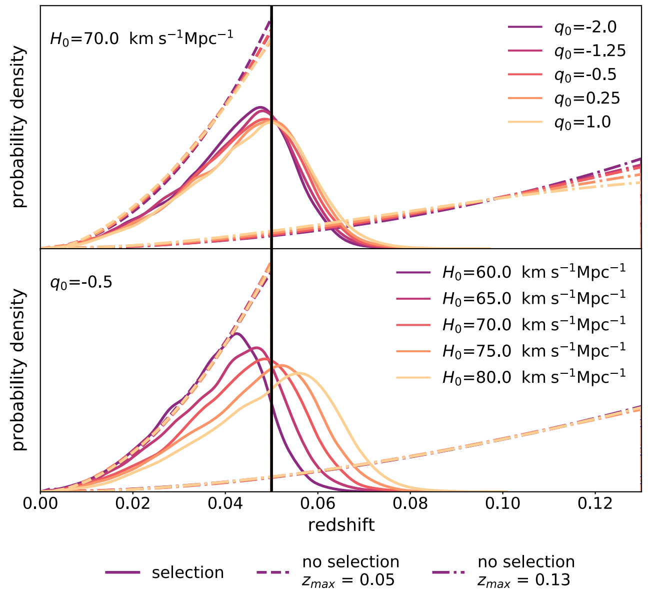

FIG. 2. The dependence of BNS redshift distributions on q_0 (top) and H_0 (bottom) for our no-selection (dashed) and selection datasets (solid). To obtain comparable constraints on H_0 from the two datasets, we impose a cutoff at $z_{\mathrm{max}} = 0.05$ for the no-selection case, while using $z_{\mathrm{max}} = 0.13$ for the selection case. The input distribution for the selection dataset is shown as a dot-dashed line.

When GW selection is not applied, we simulate populations by simply drawing from the above distributions N times. When using GW selection, we require that the signal-to-noise ratio (SNR), defined as

$$\rho_i(\hat{d}_i) = 12\left(\frac{250\ \mathrm{Mpc}}{\hat{d}_i}\right), \tag{7}$$

is greater than $\rho_* = 12$ for $i = [1, N]$. Introducing the GW selection changes the distribution of GW sources, reducing the effective upper redshift limit in a cosmology-dependent way, as shown in Fig. 2; the peak of the redshift distribution broadens and shifts to higher z for increasing H_0, while q_0 has a much smaller impact over this redshift range. For values of $H_0 \in [60, 80]\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ and $q_0 \in [-2, 1]$, the redshift distribution is peaked at $z \simeq 0.05$. To ensure we generate sources at similar redshifts for our selection and no-selection populations (and consequently obtain similar constraints on cosmological parameters) we set z_{max} equal to 0.05 and 0.13 for the no-selection and selection cases, respectively.

III. METHOD

A. Traditional inference

We begin by outlining the traditional approach to inferring parameters from GW-selected populations, before describing our adopted likelihood-free methodology.

[1] The $\sigma_{\hat{d}} \propto d$ scaling of the distance uncertainty is chosen for simplicity. A better motivated choice would be $\sigma_{\hat{d}} \propto d^2$, given that the signal-to-noise ratio, Eq. (7), scales as $1/d$ e.g., [42].

The traditional framework has been set out in numerous references [18,20–22,24,29,30,32,34,35,41,42], but we will follow the notation of Ref. [42] here. For simplicity, in this work we set aside the inference of the BNS properties (e.g., the NS mass distribution) and focus on the cosmology. As we are considering a fixed sample size here, the posterior on the cosmological parameters given a catalogue $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{z}}, \hat{\boldsymbol{d}}, \hat{\boldsymbol{v}}]$ can be written as

$$P(\boldsymbol{z}, \boldsymbol{v}, H_0, q_0 | \hat{\boldsymbol{x}}) \propto \frac{P(H_0)P(q_0)}{[\bar{N}(H_0, q_0)]^N} \prod_{i=1}^{N} P(z_i | H_0, q_0, z_{\max}) P(v_i) \times P(\hat{z}_i | z_i, v_i) P(\hat{d}_i | d_i) P(\hat{v}_i | v_i). \tag{8}$$

We assume truncated Gaussian priors on the cosmological parameters

$$P(H_0) = \mathcal{H}(H_0 - 60)\mathcal{H}(80 - H_0) \times \mathcal{N}(70\ \mathrm{km\,s^{-1}\ Mpc^{-1}}, 20\ \mathrm{km\,s^{-1}\ Mpc^{-1}})$$
$$P(q_0) = \mathcal{H}(q_0 + 2)\mathcal{H}(1 - q_0)\mathcal{N}(-0.55, 0.5). \tag{9}$$

All other distributions are taken to match those set out in Sec. II.

The impact of the selection function is captured by the factor of $[\bar{N}(H_0, q_0)]^{-N}$. $\bar{N}$ (which, recall, denotes the expected number of *detected* mergers) must be evaluated at every point in parameter space sampled by a particular inference tool. Here, we follow Ref. [42] in evaluating $\bar{N}$ on a 10×10 grid in H_0 and q_0 (boosting the fiducial detection rate $\Gamma = 1540\ \mathrm{Gpc^{-3}\ yr^{-1}}$ [36] by a factor of 130 to reduce sample variance), and then fitting using a fourth-order (15-coefficient) polynomial. Following Ref. [42], we then perform traditional Bayesian Inference using No-U-Turn-Sampling [60] as implemented in the PyStan package [61,62], explicitly sampling each merger's true redshift and peculiar velocity along with H_0 and q_0. We take the marginal posteriors on H_0 and q_0 output by PyStan as the ground truth in the tests that follow.

B. Likelihood-free inference

Explicitly calculating $\bar{N}(H_0, q_0)$ at each point of parameter space sampled is computationally unfeasible. The methods proposed to circumvent this issue must balance computational cost and accuracy. The standard method of estimating $\bar{N}$ via a reweighted sum over a set of detected mergers generated using a fiducial population [43,44,57] works well provided the object-level parameter distribution for generic population parameters does not differ too strongly from that of the fiducial population [44]. To counter this, the fiducial detected merger population must be oversampled, increasing the cost of both generating the detected sample and evaluating the likelihood. The cost of the former will become prohibitive in any setting where the distributions of object parameters have finite (or strongly suppressed) support which changes with the population parameters. Reference [45] estimates $\bar{N}$ by fitting the distribution of object parameters found in the fiducial detection set and from this obtaining an estimate of the probability of detecting a merger given its parameters. This reduces both the computational cost and the bias due to estimating the detection probability from a fiducial population that might differ strongly from the underlying truth; however, it still fundamentally depends on the assumption of a fiducial population. The gridded approximation [42] we use for our traditional Bayesian analysis here does not require a fiducial population but is computationally expensive, requiring $\sim 130 \times N$ selected mergers for each single point of the grid, hence $\sim$13000 detected samples in total. It can not be scaled to problems with a large number of population parameters.

Here we demonstrate that we can bypass the $\bar{N}$ calculation entirely using likelihood-free methods, which are based solely on simulations and therefore naturally account for selection effects. In particular, we use density-estimation likelihood-free inference (DELFI) [46–49], in which synthetic mergers sampling the joint parameter-data space $(\boldsymbol{\Omega}, \hat{\boldsymbol{x}})$ are used to train neural density estimators (NDEs) to fit $P(\hat{\boldsymbol{x}} | \boldsymbol{\Omega})$, the probability of obtaining GW-selected data given the population parameters. By fitting this distribution, we implicitly marginalize over the mergers' true redshifts and peculiar velocities. The fit is evaluated at the observed data $\hat{\boldsymbol{x}}_{\mathrm{obs}}$ to obtain $P(\hat{\boldsymbol{x}}_{\mathrm{obs}} | \boldsymbol{\Omega}; \mathbf{w})$, a parametric model for the likelihood depending on the trained weights $\mathbf{w}$ of the neural density estimators. This is then multiplied by the prior to yield the final posterior $P(\boldsymbol{\Omega} | \hat{\boldsymbol{x}}_{\mathrm{obs}}) \propto P(\boldsymbol{\Omega}) P(\hat{\boldsymbol{x}}_{\mathrm{obs}} | \boldsymbol{\Omega}; \mathbf{w})$.

Our LFI analysis uses pydelfi,[2] an implementation of DELFI developed by Ref. [49], based on Refs [46–48]. pydelfi learns a parametric model to the conditional distribution $P(\hat{\boldsymbol{x}} | \boldsymbol{\Omega})$—via *on-the-fly* or precomputed simulations —using a set of NDEs. The NDE components can be freely chosen as a combination of mixture density networks (MDNs) and masked autoregressive flows (MAFs) (see Refs. [46,49,63,64] for details on the NDEs). To reduce the possibility of pathological behavior from one particular NDE affecting our results, we create an ensemble of estimators by stacking together five MDNs (with one to five Gaussian components) and one MAF. We use the same ensemble of NDEs for all pydelfi runs. To reduce variance in our results, we train all of the NDEs using a fixed set of 2000 simulated training populations, rather than letting the algorithm generate on-the-fly simulations. These training samples are obtained by uniformly drawing from $H_0 \in [60, 80]\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ and $q_0 \in [-2, 1]$. The choice of the training-set size is empirically driven by the estimators' efficiency: there exists a (setting-specific) limiting

[2] https://github.com/justinalsing/pydelfi.

training-set size beyond which there is no significant improvement in the training [49]. Reducing the training set to 1000 populations significantly impacts the quality of our results; boosting it to 10000 does not improve the results enough to justify the higher computational cost.

1. Data compression method

As the simulated catalogues consist of $N = 100$ sources, performing LFI on the raw data would require fitting a 302-dimensional probability distribution, which is unfeasible (given the available resources in terms of number of simulations and our fidelity requirements). In order to reduce the dimensionality of the inference space, the data must be compressed to a set of summary statistics $\hat{\boldsymbol{t}}$, a vector of $\dim(\hat{\boldsymbol{t}}) \equiv \dim(\boldsymbol{\Omega})$ components (i.e., one compressed summary per parameter of interest). Identifying suitable summary statistics translates into finding a map $f\colon \hat{\boldsymbol{x}} \to \hat{\boldsymbol{t}}$ that compresses the data while retaining as much information as possible. Methods capable of performing such a mapping include score compression [48,65,66], information maximizing neural networks [67] and regression neural networks (NNs) [68]. In this work, we train regression neural networks to compress generic merger data into estimates of the generative cosmological parameters. For training purposes, we need to construct a set of training and validation datasets, for which the underlying cosmology is known and will constitute the target. The network will ultimately compress the noisy data to a set of summary statistics which correspond to a prediction about the generative cosmological model. To avoid any dependence on the particular training initialization of a single network, we create an ensemble of 9 trained neural networks, all defined by the same settings and trained on the same exact data but using different random initial weights.

The raw observables span a broad range of magnitudes —$\hat{z} \simeq O(10^{-2})$, $\hat{d} \simeq O(10^{2})$ and $\hat{v} \simeq O(10^{3})$—which can cause problems in the training process. If there are large differences in scale between different components of the data vector, the NN will naturally prioritize the larger components, effectively ignoring part of the dataset. Moreover, the magnitude of the data vector determines the update rate, so large values might lead to stability problems. Prior to feeding data into any neural network, therefore, we normalize the data to ensure they are all at roughly the same scale. We first sort all merger catalogues by redshift to reduce the variability to which each NN input node is exposed. We then concatenate each catalogue's $\hat{\boldsymbol{z}}$, $\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{v}}$ to create a single 300-element raw-input vector. Finally we shift and scale by the mean and standard deviation of 100 catalogues generated at our fiducial cosmology $[H_0, q_0] = [70, -0.5]$ to create the normalized inputs for our regression networks. We also normalize the target parameters which generated the training and validation datasets, shifting and scaling their distributions to be within 0 and 1. The NN predictions—our summary statistics—are hence normalized estimates of the cosmological parameters.

2. Data compression optimization

The choice of architecture and settings for our neural networks is completely free, which poses an intimidating optimization problem over the vast number of possible NN architectures and settings. To define a NN we must choose an architecture, its activation function and training, by tuning batch size, learning rate and potentially employing regularization methods. We cannot reasonably explore all of these choices, and we therefore consider neural networks composed of two hidden layers, each made of 128 hidden units, fix the activation function to be a leaky relu [69] with `alpha` $= 0.01$,[3] and focus on finding the best combination of batch size n_{batch} and learning rate α from a small set of choices, namely $n_{\mathrm{batch}} = [100, 500]$ and $\alpha = [10^{-4}, 5\times 10^{-4}, 10^{-3}]$. To avoid potential overfitting, we consider regularization terms, which control the training while acting on the loss function, set to be the mean squared error (MSE). We toggle between ridge and lasso regression methods, which use L2 and L1 regularizations respectively [70], and explore a few values of the parameters weighting the regularization term, $\lambda_{1,2}$, namely $\{\lambda_{1,2} = 0\}$,$\{\lambda_1 = 0, \lambda_2 = [10^{-4}, 2\times 10^{-4}]\}$ and $\{\lambda_1 = [10^{-4}, 2\times 10^{-4}], \lambda_2 = 0\}$. We define the optimal compressor as the NN for which pydelfi most faithfully reproduces PyStan's results for a range of $[H_0, q_0]$. The process by which we determine the optimal NN settings is described in the following.

For each combination of batch size, learning rate and regularization, we first train the regression NN on a set of n_{train} samples of known cosmology, validating with a further n_{val} datasets. To determine the impact of the amount of training data available on the final inference, we consider two training set sizes, the first with $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [5000, 2000]$ and the second with [500000, 100000]. In all cases, the generative cosmologies are sampled from $H_0 \in [60, 80]\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ and $q_0 \in [-2, 1]$ using the Latin hypercube method.

To determine the NN parameters that optimize LFI performance for a range of underlying cosmologies, we generate 100 test catalogues for cosmological parameters sampled from $H_0 \in [65, 75]\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ and $q_0 \in [-0.7, -0.3]$ using the Latin hypercube method (the reason for this restricted range will be explained in Sec. IV). We then perform traditional Bayesian inference and LFI on each test catalogue, for each choice of NN parameters. Given these results, we compute the differences $b_{H_0} = \hat{H}_0^{\mathrm{trad}} - \hat{H}_0^{\mathrm{LFI}}$ and $b_{q_0} = \hat{q}_0^{\mathrm{trad}} - \hat{q}_0^{\mathrm{LFI}}$ between the maximum-posterior estimates of the cosmological parameters from the traditional and LFI approaches, which we define

[3] https://keras.io/api/layers/activation_layers/leaky_relu/.

as “biases”.[4] Compiling the results from all of the test catalogues, we calculate the means ($\bar{b}_{H_0,q_0}$) and standard deviations ($\sigma_{b_{H_0,q_0}}$) of the biases injected by LFI for each compression NN. The optimal compression network is chosen to be that which minimizes the standard deviation of the H_0 bias, provided its mean bias is consistent with zero.

In addition to requiring LFI produces unbiased estimates of the cosmological parameters, we also want to ensure our compression is as lossless as possible, i.e., that the LFI and traditional constraints have similar H_0 uncertainties. To do so, we need the total uncertainty in the LFI parameter constraints, which we approximate as the quadrature sum of the “raw” uncertainty of the LFI posteriors and the additional uncertainty due to the bias.[5] We estimate the former by calculating the mean variance of the LFI cosmological parameter posteriors over all 100 test catalogues; the uncertainty on the bias is simply $\sigma_{b_{H_0}}$. Hence, the increase in the H_0 uncertainty expected from replacing traditional Bayesian inference with LFI in this setting can be estimated by calculating

$$\%\hat{\sigma}^{H_0}_{\mathrm{incr}} = 100 \times \left(\frac{\sqrt{(\sigma^{H_0}_{\mathrm{LFI}})^2 + \sigma^2_{b_{H_0}}}}{\sigma^{H_0}_{\mathrm{trad}}} - 1 \right). \tag{10}$$

IV. RESULTS

We first consider the no-selection case to demonstrate the feasibility of LFI in this setting and obtain a baseline for its impact on the precision and accuracy of the inference. We then add in GW selection to determine whether selection specifically affects LFI’s performance, and to provide a final estimate of the systematics.

A. No-selection case

Considering the no-selection case first gives us a baseline for gauging LFI’s performance in the more complex setting with selection, allowing us to determine whether selection specifically has any impact on LFI. We train our compression NNs for all combinations of the aforementioned batch size, learning rate and regularizer choices, for both training-set sizes $[n_{\mathrm{train}}, n_{\mathrm{val}}]$. Each of these neural networks provides different compression performance and thus all are tested as compressors in the LFI workflow. An example of compression performance for $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [500000, 200000]$ is given in Fig. 3, which shows the summary statistics $\hat{\boldsymbol{t}}$ output by the regression NN against the generative cosmological parameters for the validation set. Focusing on the $\hat{t}_1 - H_0$ and $\hat{t}_2 - H_0$ plots for now, we notice that the width and slope of the distribution change at the edges of the training set, shaded in grey. As the NN behavior might be suboptimal in these ranges, we generate the test samples used to optimize the compressor settings from values of H_0 within $[65, 75]$ km s^{-1} Mpc^{-1}, lying in the unshaded area.

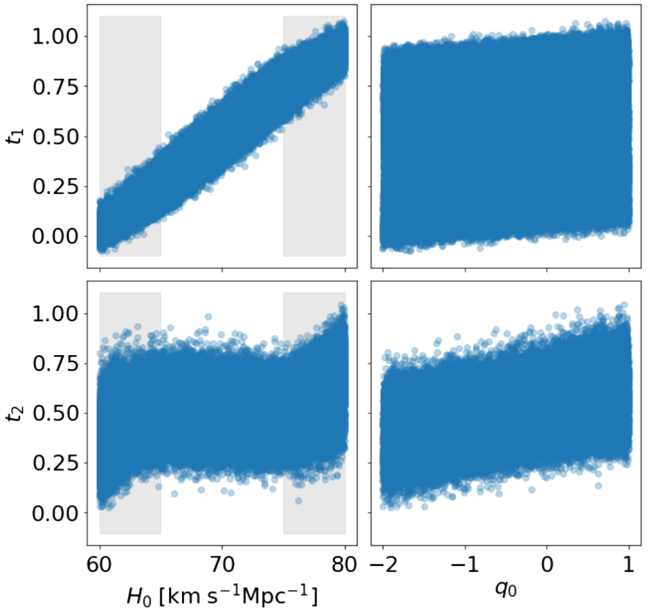

FIG. 3. The summary statistics $\hat{\boldsymbol{t}} = (\hat{t}_1, \hat{t}_2)$ output by our compression NN plotted against the cosmological parameters at which the corresponding data were generated. This NN was trained with $[n_{\mathrm{batch}}, \alpha, \lambda_{1,2}] = [100, 10^{-4}, 0]$, and the points correspond to the validation dataset for the $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [500000, 200000]$ setup. The shaded areas indicate the regions of H_0 where the slopes of the summary statistics change with respect to the central trend.

We identify the best regularization for each combination of batch size and learning rate using the b_{H_0} distribution. The b_{H_0} and b_{q_0} probability densities are respectively shown as blue and orange violin plots in Fig. 4, for $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [5000, 2000]$, and summarized in Table I. Results for all NN parameter choices can be found in Tables III and IV. From the violin plots we see that the likelihood-free inference of both H_0 and q_0 is unbiased, since the bias is consistent with zero for all choices of NN parameters. For the best models, independent of the specific NN parameters and data realization, LFI’s maximum posterior estimate for both parameters is typically well within PyStan’s 1σ posterior uncertainty ($\geq$0.89 km s^{-1} Mpc^{-1} for these test populations).

We observe that for our smaller training set, regularization greatly improves performance. As an example,

[4] As such, these biases contain contributions from any inaccuracies in the traditional $\bar{N}$ estimation and inference (expected to be small) and loss of information through imperfect compression. If the compression is lossless and the PyStan inference introduces no error, the PyStan and LFI posteriors should match perfectly.

[5] This is equivalent to marginalizing over an unknown additive bias, assuming the parameters and bias are independent and Gaussian-distributed.

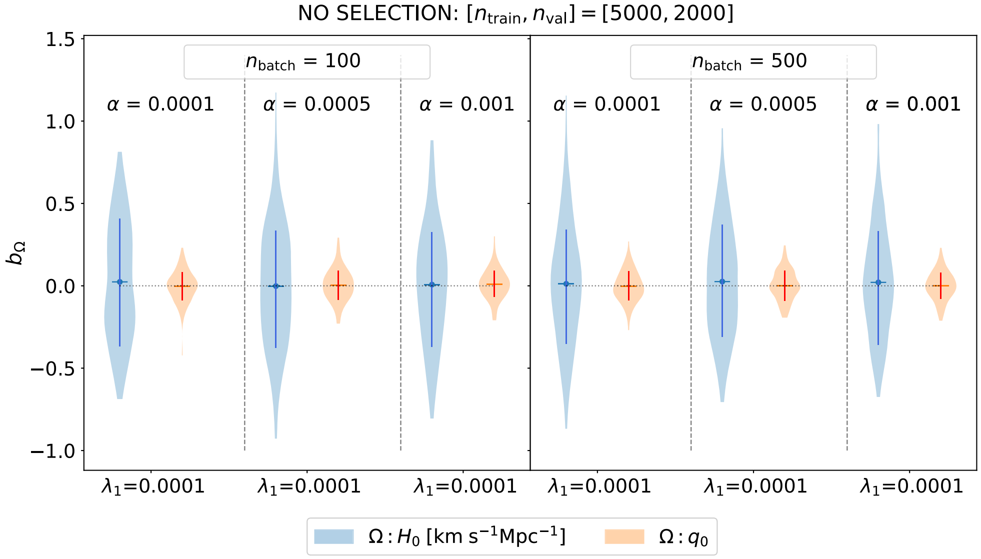

FIG. 4. Violin plots for the $b_{H_0} = \hat{H}_0^{\rm trad} - \hat{H}_0^{\rm LFI}$ (blue) and $b_{q_0} = \hat{q}_0^{\rm trad} - \hat{q}_0^{\rm LFI}$ (orange) bias distributions for the no-selection setting. Results are shown for the NNs whose regularization choice minimizes the bias for each combination of batch size $n_{\rm batch}$ and learning rate α. Dots represent the mean biases, and lines the 1σ error bars. The mean biases are consistent with zero, and the bias distributions are considerably narrower than the relevant parameter posteriors, for all NNs plotted.

considering $[n_{\rm batch}, \alpha] = [100, 10^{-4}]$ we find that adding a regularization term $\lambda_1 = 10^{-4}$ reduces $\sigma_{b_{H_0}}$ from 1.75 to 0.35 and markedly increases the H_0 constraining power, reducing $f_\sigma^{H_0} = \sigma_{H_0}^{\rm LFI}/\sigma_{H_0}^{\rm trad}$ from 1.95 to 1.06. With regularization added, the width of the LFI H_0 posterior is compatible with PyStan's. Considering the larger training set reduces the impact of the regularizer and significantly reduces the H_0 LFI posterior's uncertainty, which we find to be systematically ∼2–3% smaller than PyStan's: we suspect that this is due to slight overfitting by pydelfi. The LFI q_0 constraints are also ∼5% tighter than PyStan's, independent of the size of the training set.

For the $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$ setup, the network with $[n_{\rm batch}, \alpha, \lambda_1] = [500, 10^{-3}, 10^{-4}]$ imparts the smallest bias in the H_0 posterior, with $\sigma_{b_{H_0}} = 0.32$. The H_0 bias

TABLE I. Means and standard deviations for the biases b_{H_0,q_0}, posterior-width ratios f_{H_0,q_0} and percentage increase in H_0 uncertainty for the NNs whose regularization choice minimizes the bias for each combination of batch size $n_{\rm batch}$ and learning rate α in the no-selection case.

NO SELECTION CASE							
$n_{\rm batch}$	α	Regularizer	b_{H_0}[km s^{-1} Mpc^{-1}]	b_{q_0}	$f_\sigma^{H_0}$	$f_\sigma^{q_0}$	$\%\hat{\sigma}_{\rm incr}^{H_0}$
TRAINING and VALIDATION parameters: $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$							
100	10^{-4}	$\lambda_1 = 10^{-4}$	0.024 ± 0.35	-0.003 ± 0.095	1.014 ± 0.045	0.95 ± 0.035	7.64%
	5×10^{-4}	$\lambda_1 = 10^{-4}$	-0.002 ± 0.365	0.004 ± 0.098	1.028 ± 0.042	0.952 ± 0.032	9.43%
	10^{-3}	$\lambda_1 = 10^{-4}$	0.007 ± 0.352	0.009 ± 0.09	1.024 ± 0.048	0.952 ± 0.038	8.58%
500	10^{-4}	$\lambda_1 = 10^{-4}$	0.012 ± 0.358	-0.003 ± 0.092	1.003 ± 0.043	0.947 ± 0.036	6.81%
	5×10^{-4}	$\lambda_1 = 10^{-4}$	0.026 ± 0.328	0.001 ± 0.091	1.018 ± 0.051	0.948 ± 0.026	7.3%
	10^{-3}	$\lambda_1 = 10^{-4}$	0.021 ± 0.322	-0.0 ± 0.087	1.012 ± 0.054	0.943 ± 0.036	6.45%
TRAINING and VALIDATION parameters: $[n_{\rm train}, n_{\rm val}] = [500000, 100000]$							
100	10^{-4}	$\lambda_2 = 2 \times 10^{-4}$	-0.073 ± 0.193	0.015 ± 0.061	0.979 ± 0.042	0.945 ± 0.038	−0.05%
	5×10^{-4}	None	-0.061 ± 0.218	0.014 ± 0.071	0.978 ± 0.048	0.948 ± 0.04	0.35%
	10^{-3}	None	-0.058 ± 0.21	0.02 ± 0.058	0.973 ± 0.042	0.945 ± 0.04	−0.35%
500	10^{-4}	$\lambda_2 = 10^{-4}$	-0.043 ± 0.193	0.017 ± 0.066	0.972 ± 0.041	0.944 ± 0.039	−0.77%
	5×10^{-4}	$\lambda_2 = 2 \times 10^{-4}$	-0.053 ± 0.208	0.015 ± 0.061	0.975 ± 0.046	0.944 ± 0.037	−0.15%
	10^{-3}	$\lambda_2 = 10^{-4}$	-0.062 ± 0.189	0.012 ± 0.06	0.979 ± 0.048	0.946 ± 0.035	−0.22%

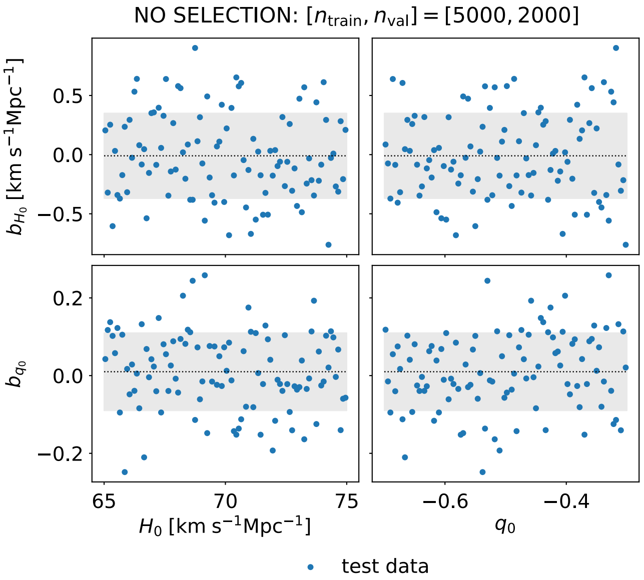

FIG. 5. Distribution of generative parameters and LFI posterior biases. The one-sigma range of the bias is shaded grey. The neural network model used to perform the compression and generate this plot corresponds to the NN parameters combination $[n_{\rm batch}, \alpha, \lambda_1] = [500, 10^{-3}, 10^{-4}]$ for $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$.

shrinks further when using our larger training set, with $\sigma_{b_{H_0}} = 0.19$. As the bias is small and consistent with zero it could be ignored when doing population-level inference; here, however, we marginalize over it and find that it would impart a 6.45% and −0.05% increase in the quoted H_0 uncertainty, respectively: well within any reasonable tolerance. We note here that this slight increase in uncertainty is entirely down to imperfect compression, since in tests pydelfi provides the same posteriors when rerunning on the same compressed data.

One advantage of using a regression neural network for compression is that it only relies on a fiducial model for the computation of the mean and standard deviations used to normalize the neural network inputs. Nevertheless, the compression is sensitive to the choice of the training and validation data, as well as the range of sampled $\mathbf{\Omega}$ values. To investigate the randomness of the H_0 bias with respect to the sampled parameter space, we plot the biases against the generative parameters for all 100 test catalogues for our best compression network in Fig. 5. We find there is no major correlation between the true parameters and the biases (for example, for the best model of the $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$ setup, we find correlation coefficients of $C(H_0, b_{H_0}) = -0.13$ and $C(q_0, b_{H_0}) = -0.023$).

B. Selection case

We now proceed to determine the impact of selection on the compression. As in the no-selection case, we first optimize the regularization for each combination of batch size and learning rate. We compute the distributions of the H_0 and q_0 biases, plotting the results for the best compressors in Fig. 6 and tabulating their performance in Table II. Results for all the NN parameters can be found in Tables V and VI. As in the no-selection case, the LFI maximum-posterior parameter estimates are unbiased when compared to the PyStan baseline.

As before, for our smaller training set regularization overall largely improves the performance. Considering

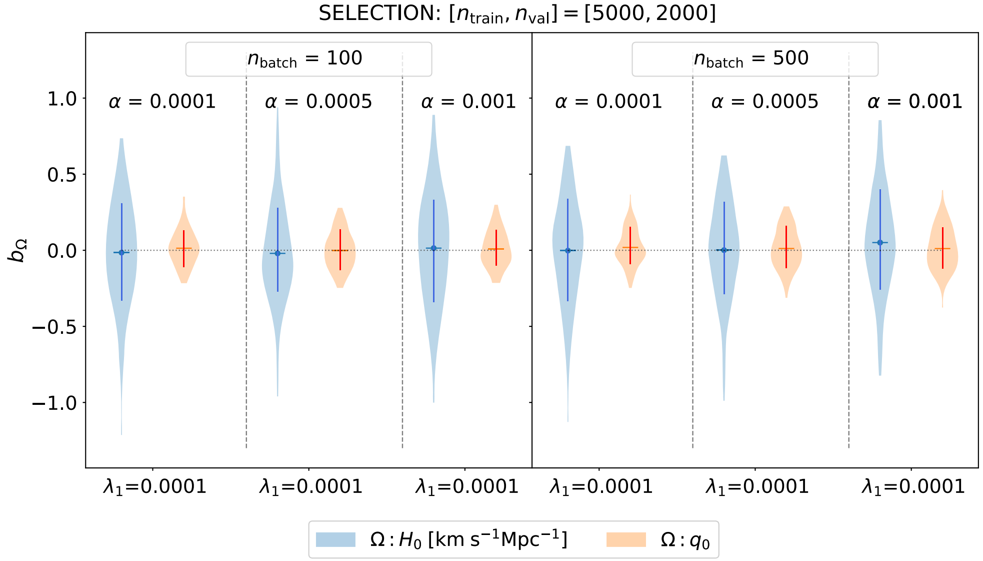

FIG. 6. Violin plots for the b_{H_0} (blue) and b_{q_0} (orange) bias distributions for the setting with GW selection. Results are shown for the NNs whose regularization choice minimizes the bias for each combination of batch size $n_{\rm batch}$ and learning rate α. Dots represent the mean biases, and lines the 1σ error bars. As in the no-selection case, the mean biases are all consistent with zero, and the bias distributions are all considerably narrower than the relevant parameter posteriors.

TABLE II. Means and standard deviations for the biases b_{H_0,q_0}, posterior-width ratios f_{H_0,q_0} and percentage increase in H_0 uncertainty for the NNs whose regularization choice minimizes the bias for each combination of batch size n_{batch} and learning rate α in the selection case.

SELECTION CASE							
n_{batch}	α	Regularizer	b_{H_0}[km s^{-1} Mpc^{-1}]	b_{q_0}	$f_\sigma^{H_0}$	$f_\sigma^{q_0}$	$\%\hat{\sigma}_{\mathrm{incr}}^{H_0}$
TRAINING and VALIDATION parameters: $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [5000, 2000]$							
100	10^{-4}	$\lambda_1 = 10^{-4}$	-0.015 ± 0.338	0.014 ± 0.115	1.013 ± 0.039	1.005 ± 0.044	6.53%
	5×10^{-4}	$\lambda_1 = 10^{-4}$	-0.02 ± 0.313	-0.001 ± 0.122	1.014 ± 0.041	1.005 ± 0.058	5.9%
	10^{-3}	$\lambda_1 = 10^{-4}$	0.014 ± 0.357	0.008 ± 0.119	1.018 ± 0.042	1.006 ± 0.051	7.67%
500	10^{-4}	$\lambda_1 = 10^{-4}$	-0.002 ± 0.334	0.019 ± 0.116	1.025 ± 0.04	1.01 ± 0.049	7.63%
	5×10^{-4}	$\lambda_1 = 10^{-4}$	0.001 ± 0.313	0.012 ± 0.128	1.025 ± 0.038	1.013 ± 0.036	6.99%
	10^{-3}	$\lambda_1 = 10^{-4}$	0.051 ± 0.329	0.011 ± 0.137	1.019 ± 0.053	1.011 ± 0.058	6.91%
TRAINING and VALIDATION parameters: $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [500000, 100000]$							
100	10^{-4}	$\lambda_2 = 10^{-4}$	-0.032 ± 0.184	0.022 ± 0.092	0.976 ± 0.031	1.006 ± 0.036	−0.73%
	5×10^{-4}	$\lambda_2 = 10^{-4}$	-0.033 ± 0.177	0.02 ± 0.092	0.979 ± 0.039	1.003 ± 0.043	−0.56%
	10^{-3}	None	0.0 ± 0.183	0.026 ± 0.091	0.965 ± 0.03	1.004 ± 0.038	−1.88%
500	10^{-4}	$\lambda_1 = 10^{-4}$	-0.022 ± 0.178	0.015 ± 0.093	0.978 ± 0.036	1.003 ± 0.039	−0.7%
	5×10^{-4}	$\lambda_2 = 10^{-4}$	-0.013 ± 0.18	0.019 ± 0.086	0.977 ± 0.035	1.006 ± 0.038	−0.68%
	10^{-3}	$\lambda_2 = 2 \times 10^{-4}$	-0.01 ± 0.199	0.021 ± 0.083	0.979 ± 0.043	1.003 ± 0.042	−0.18%

$[n_{\mathrm{batch}}, \alpha] = [100, 10^{-4}]$ as an example as before, we find that regularizing the training for $\lambda_1 = 10^{-4}$ reduces the uncertainty on the H_0 bias from 1.71 to 0.34 and greatly improves the H_0 constraining power, from $f_\sigma^{H_0} = 1.77$ to 1.06. As in the no-selection case, the LFI posteriors produced using the optimal compressors are completely compatible with PyStan's. Again, increasing the training set size reduces the impact of the regularizer and significantly reduces the LFI H_0 posterior's uncertainty, to ∼2.5% smaller than PyStan's.

For the $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [5000, 2000]$ setup two NN compressors minimize the H_0 bias, with $\sigma_{b_{H_0}} = 0.31$. These are defined by $[n_{\mathrm{batch}}, \alpha, \lambda_1] = \{[100, 5 \times 10^{-4}, 10^{-4}], [500, 5 \times 10^{-4}, 10^{-4}]\}$. As in the no-selection case the best models compressors use λ_1 regularization. For the larger $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [500000, 100000]$ setup, the smallest standard deviation for the H_0 bias is again considerably smaller: $\sigma_{b_{H_0}} = 0.18$ for the compressor with $[n_{\mathrm{batch}}, \alpha, \lambda_2] = [100, 5 \times 10^{-4}, 10^{-4}]$. As in the no-selection case, we compute the percentage increase in uncertainty on H_0 imparted by replacing traditional inference with LFI, marginalizing over the bias. For the aforementioned three best compressors, these percentage increases are $\{5.9\%, 6.99\%\}$ and −0.56%, respectively, compatible with that determined for the no-selection case. Including GW selection does not impact LFI performance on a statistical level. Illustrative examples of the $H_0 - q_0$ joint posteriors produced by pydelfi and PyStan can be found in Fig. 8.

In Fig. 7 we plot the values of the b_{H_0} and b_{q_0} distributions against true input cosmology parameters. Unlike in Fig. 5, there is a clear dependence of b_{H_0} and b_{q_0} on the true value of H_0 that generated the data. The strongest correlation is between the q_0 bias and the generative H_0, with a correlation coefficient of −0.47 for the best model $[n_{\mathrm{batch}}, \alpha, \lambda_1] = [100, 5 \times 10^{-4}, 10^{-4}]$

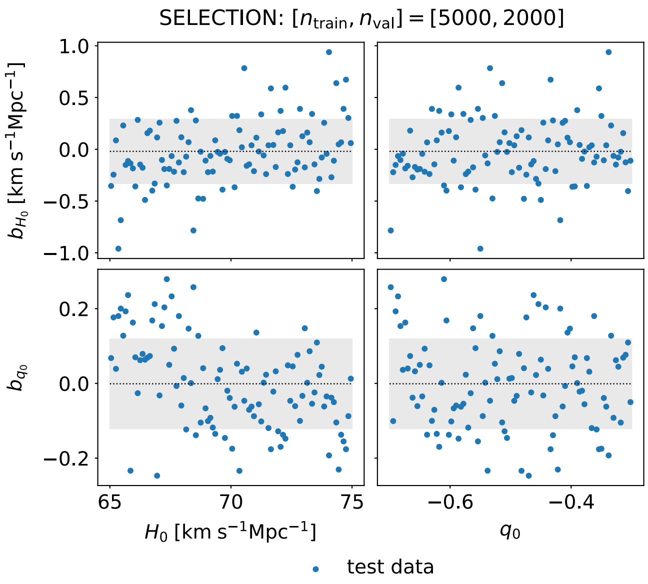

FIG. 7. Distribution of generative parameters and LFI posterior biases for the GW selection setting. The one-sigma range of the bias is shaded grey. The neural network model used to perform the compression and generate this plot corresponds to the NN parameters combination $[n_{\mathrm{batch}}, \alpha, \lambda_1] = [100, 5 \times 10^{-4}, 10^{-4}]$ for $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [5000, 2000]$.

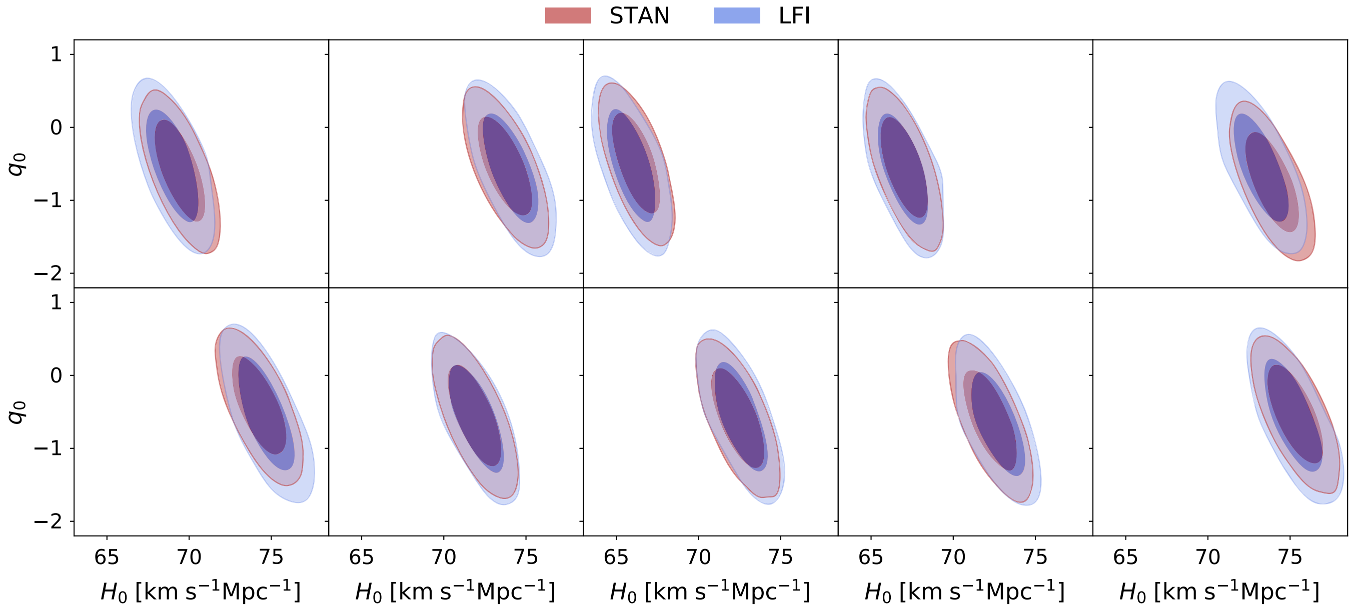

FIG. 8. Example posterior contour plots produced by LFI (blue) and traditional Bayesian sampling (red) for test datasets with GW selection.

of the smaller training set $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$. Increasing the size of the training sample generates stronger correlation (−0.66 for the best model). This indicates the regression (or, indeed, an imperfect fit to PyStanS $\bar{N}$) is not capturing the selection function perfectly, and that other compression methods may fare better. Nevertheless, for the optimal compressors the biases on the cosmological parameters are consistent with zero, and have standard deviations which are a small fraction of the full posterior uncertainty.

V. CONCLUSIONS

We have investigated the ability of likelihood-free inference (LFI) to estimate the cosmological expansion from GW-selected populations of binary neutron star mergers with EM counterparts. When computing the parameter posterior using traditional Bayesian inference, selection effects must be taken into account through the computation of the expected number of detected sources, $\bar{N}$. This is a computationally expensive (and potentially inaccurate) process, even in approximate forms [42–44]. As LFI does not explicitly evaluate the posterior, instead building a proxy likelihood using neural density estimator fits to parameter–simulated-dataset pairs, there is no need to calculate $\bar{N}$ when performing LFI. Instead, the selection is naturally built into the simulations on which the method is based.

The goal of this work was to compare the precision and accuracy achievable using LFI to that of traditional Bayesian inference in the presence of selection effects. We note that improvements to the traditional found-injection approach broadening injection-set coverage (through, e.g., designer injection sets covering a range of populations) have the potential to improve the resulting $\bar{N}$ estimates. A quantitative comparison (in terms of precision, accuracy and computational cost, and considering more complete data models) of the LFI approach with improved found-injection methods is strongly motivated by the findings of this proof-of-concept work.

In this work we considered GW selection only; adding EM selection would increase the computational burden, making accounting for selection effects even more expensive. We employed "preprocessed" 100-merger datasets, consisting of noisy estimates of redshift, distance and peculiar velocity for each merger, assuming the distances have already been inferred from GW strains (which can be performed rapidly as in Ref. [55] to yield a fully LFI-based pipeline). Given the high dimensionality of the input data, LFI methods require the data to be compressed to a set of summary statistics. We trained ensembles of regression neural networks for this purpose, passing their outputs to the density-estimation likelihood-free-inference package pydelfi to infer the cosmological parameters. Both of these stages require the provision of training data: we have presented results for compression networks trained using $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$ and $[n_{\rm train}, n_{\rm val}] = [500000, 100000]$ populations; in all cases pydelfi was trained using 2000 simulated populations. Given each population contains 100 mergers, the total number of detected mergers required to train the two setups was 9×10^5 and 6×10^8, respectively.

LFI's precision and accuracy depends sensitively on the compression method's ability to retain salient information about the parameters of interest. We trained a large suite of regression networks (each containing two hidden layers of

128 hidden units) for compression, optimizing the learning rate, batch size and regularization based on pydelfi's ability to infer H_0 using the networks' outputs. Specifically, we selected the network whose resulting H_0 inference best reproduced the traditional Bayesian "ground truth" (as implemented using PyStan) for a set of 100 test datasets, taking the differences between maximum-posterior H_0 estimates for the two methods as our metric.

Testing the method first on datasets in which no GW selection was made, we demonstrated that LFI provides unbiased H_0 estimates when using suitably optimized regression-network data compression. For our optimal combination of training variables, we found a bias (defined as the difference between the maximum-posterior PyStan and pydelfi estimates) on H_0 of $b_{H_0} = 0.021 \pm 0.322\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$: consistent with zero and with a standard deviation a factor of roughly three smaller than the posterior uncertainty on H_0. Marginalizing over this bias would lead to an increase of only 6.45% in the uncertainty on H_0. Adding in GW selection, we find no impact on LFI's performance: LFI is still able to provide unbiased estimates of H_0 in the presence of selection effects. For the best model we obtain $b_{H_0} = -0.02 \pm 0.313\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$, which would yield an increase in uncertainty on H_0 of only 5.9% when marginalized over. Increasing the number of samples used to train the compression networks results in LFI posteriors that are statistically indistinguishable from their traditional Bayesian counterparts in mean and variance; however, this comes with a significant increase in computational cost. When processing GW-selected data, we note a small but significant correlation between the H_0 and q_0 biases and the generative H_0 values. This indicates a different choice of compressor architecture and setup might improve results, but investigating alternative compression methods is left for future work.

As this method is simulation-based, having a trustworthy and sufficient generative model is critical. This analysis has been conducted on simplified mock data, for which we know the underlying model. In the context of real observations, more-realistic simulations, such as those implemented in LALSuite [71], are needed. As current ground-based interferometers enhance their sensitivity [33], third-generation GW detectors such as Einstein Telescope [72] and Cosmic Explorer [73] come online, and the BNS sample builds, including instrumental systematics [74] and an as-yet elusive model of joint EM-GW selection e.g., [31,75–81] will become ever more important. In this work we have focused on inferring the cosmological parameters only, but complete inference of the population properties of BNS catalogues must include parameters fixed here, such as the merger rate, mass distributions and equation of state e.g., [57,82–86]. Extending the analysis to incorporate these parameters is left to future work. Finally, we note that, though we have focused on the inference of the cosmological expansion from GW-selected catalogues of binary neutron star mergers with EM counterparts here, this method can be applied to a broad range of population analyses in the presence of selection effects e.g., [57,58].

The code is provided at [87].

ACKNOWLEDGMENTS

We thank Tom Charnock, Will Farr, Colm Talbot and Niall Jeffrey for helpful discussions. This work was partially enabled by funding from the UCL Cosmoparticle Initiative. S. M. F. is supported by the Royal Society. J. A. was supported by the research project grant *Fundamental Physics from Cosmological Surveys* funded by the Swedish Research Council (VR) under Dnr 2017-04212.

APPENDIX: FULL TABLES

For completeness, in the following we tabulate the results for all combinations of learning rate, batch size and regularization explored for both no-selection and selection analyses.

TABLE III. Means and standard deviations for the biases b_{H_0,q_0}, posterior-width ratios f_{H_0,q_0} and percentage increase in H_0 uncertainty for all combinations of batch size, learning rate and regularization in the no-selection case, using $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$.

NO SELECTION CASE							
$n_{\rm batch}$	α	Regularizer	$b_{H_0}[{\rm km\,s^{-1}\ Mpc^{-1}}]$	b_{q_0}	$f_\sigma^{H_0}$	$f_\sigma^{q_0}$	$\%\hat{\sigma}_{\rm incr}^{H_0}$
TRAINING and VALIDATION parameters: $[n_{\rm train}, n_{\rm val}] = [5000, 2000]$							
100	10^{-4}	None	0.369 ± 1.752	0.002 ± 0.098	1.951 ± 0.124	0.948 ± 0.023	165.74%
		$\lambda_2 = 10^{-4}$	-0.002 ± 0.459	0.008 ± 0.089	1.055 ± 0.043	0.95 ± 0.028	15.59%
		$\lambda_2 = 2 \times 10^{-4}$	0.054 ± 0.415	0.008 ± 0.095	1.056 ± 0.05	0.95 ± 0.028	13.95%
		$\lambda_1 = 10^{-4}$	0.024 ± 0.35	-0.003 ± 0.095	1.014 ± 0.045	0.95 ± 0.035	7.64%
		$\lambda_1 = 2 \times 10^{-4}$	0.012 ± 0.398	0.009 ± 0.1	1.038 ± 0.049	0.95 ± 0.037	11.58%
	5×10^{-4}	None	-0.101 ± 1.612	-0.003 ± 0.103	1.855 ± 0.116	0.948 ± 0.031	148.89%
		$\lambda_2 = 10^{-4}$	0.008 ± 0.404	0.012 ± 0.082	1.043 ± 0.039	0.948 ± 0.031	12.24%
		$\lambda_2 = 2 \times 10^{-4}$	-0.003 ± 0.423	0.011 ± 0.084	1.04 ± 0.042	0.948 ± 0.029	12.76%
		$\lambda_1 = 10^{-4}$	-0.002 ± 0.365	0.004 ± 0.098	1.028 ± 0.042	0.952 ± 0.032	9.43%
		$\lambda_1 = 2 \times 10^{-4}$	-0.006 ± 0.415	0.011 ± 0.083	1.027 ± 0.042	0.947 ± 0.03	11.27%
	10^{-3}	None	0.053 ± 0.498	0.005 ± 0.09	1.058 ± 0.037	0.954 ± 0.028	17.55%
		$\lambda_2 = 10^{-4}$	-0.014 ± 0.385	0.009 ± 0.083	1.04 ± 0.045	0.952 ± 0.032	11.32%
		$\lambda_2 = 2 \times 10^{-4}$	-0.012 ± 0.391	0.005 ± 0.09	1.036 ± 0.048	0.949 ± 0.031	11.15%
		$\lambda_1 = 10^{-4}$	0.007 ± 0.352	0.009 ± 0.09	1.024 ± 0.048	0.952 ± 0.038	8.58%
		$\lambda_1 = 2 \times 10^{-4}$	-0.026 ± 0.418	0.011 ± 0.086	1.022 ± 0.048	0.948 ± 0.035	10.94%
500	10^{-4}	None	0.087 ± 2.066	-0.007 ± 0.105	2.047 ± 0.163	0.948 ± 0.025	195.21%
		$\lambda_2 = 10^{-4}$	-0.035 ± 0.41	0.003 ± 0.099	1.049 ± 0.047	0.948 ± 0.032	13.05%
		$\lambda_2 = 2 \times 10^{-4}$	-0.007 ± 0.375	-0.004 ± 0.087	1.04 ± 0.044	0.95 ± 0.04	10.91%
		$\lambda_1 = 10^{-4}$	0.012 ± 0.358	-0.003 ± 0.092	1.003 ± 0.043	0.947 ± 0.036	6.81%
		$\lambda_1 = 2 \times 10^{-4}$	0.01 ± 0.399	0.002 ± 0.096	1.041 ± 0.043	0.948 ± 0.032	11.94%
	5×10^{-4}	None	-0.195 ± 1.807	0.003 ± 0.099	2.068 ± 0.157	0.948 ± 0.022	178.17%
		$\lambda_2 = 10^{-4}$	0.021 ± 0.427	0.003 ± 0.096	1.041 ± 0.042	0.949 ± 0.026	13.01%
		$\lambda_2 = 2 \times 10^{-4}$	-0.027 ± 0.388	0.009 ± 0.099	1.038 ± 0.053	0.953 ± 0.035	11.19%
		$\lambda_1 = 10^{-4}$	0.026 ± 0.328	0.001 ± 0.091	1.018 ± 0.051	0.948 ± 0.026	7.3%
		$\lambda_1 = 2 \times 10^{-4}$	0.015 ± 0.361	-0.002 ± 0.087	1.022 ± 0.043	0.951 ± 0.032	8.73%
	10^{-3}	None	0.593 ± 1.892	0.001 ± 0.101	2.078 ± 0.172	0.951 ± 0.022	184.82%
		$\lambda_2 = 10^{-4}$	0.01 ± 0.413	0.01 ± 0.088	1.052 ± 0.044	0.949 ± 0.034	13.42%
		$\lambda_2 = 2 \times 10^{-4}$	0.003 ± 0.413	0.001 ± 0.084	1.038 ± 0.04	0.947 ± 0.036	12.13%
		$\lambda_1 = 10^{-4}$	0.021 ± 0.322	-0.0 ± 0.087	1.012 ± 0.054	0.943 ± 0.036	6.45%
		$\lambda_1 = 2 \times 10^{-4}$	-0.01 ± 0.362	0.01 ± 0.101	1.03 ± 0.053	0.95 ± 0.041	9.59%

TABLE IV. Means and standard deviations for the biases b_{H_0,q_0}, posterior-width ratios f_{H_0,q_0} and percentage increase in H_0 uncertainty for all combinations of batch size, learning rate and regularization in the no-selection case, using $[n_{\text{train}}, n_{\text{val}}] = [500000, 100000]$.

NO SELECTION CASE							
n_{batch}	α	Regularizer	$b_{H_0}[\text{km s}^{-1}\text{ Mpc}^{-1}]$	b_{q_0}	$f_\sigma^{H_0}$	$f_\sigma^{q_0}$	$\%\hat{\sigma}_{\text{incr}}^{H_0}$
TRAINING and VALIDATION parameters: $[n_{\text{train}}, n_{\text{val}}] = [500000, 100000]$							
100	10^{-4}	None	-0.063 ± 0.253	0.016 ± 0.065	0.969 ± 0.042	0.945 ± 0.038	0.3%
		$\lambda_2 = 10^{-4}$	-0.053 ± 0.193	0.018 ± 0.065	0.981 ± 0.047	0.951 ± 0.042	0.11%
		$\lambda_2 = 2 \times 10^{-4}$	-0.073 ± 0.193	0.015 ± 0.061	0.979 ± 0.042	0.945 ± 0.038	−0.05%
		$\lambda_1 = 10^{-4}$	-0.073 ± 0.243	0.023 ± 0.062	0.97 ± 0.044	0.944 ± 0.038	0.13%
		$\lambda_1 = 2 \times 10^{-4}$	-0.075 ± 0.254	0.006 ± 0.064	0.975 ± 0.038	0.943 ± 0.034	0.93%
	5×10^{-4}	None	-0.061 ± 0.218	0.014 ± 0.071	0.978 ± 0.048	0.948 ± 0.04	0.35%
		$\lambda_2 = 10^{-4}$	-0.058 ± 0.222	0.015 ± 0.063	0.972 ± 0.045	0.945 ± 0.041	−0.18%
		$\lambda_2 = 2 \times 10^{-4}$	-0.044 ± 0.224	0.016 ± 0.058	0.981 ± 0.049	0.945 ± 0.037	0.79%
		$\lambda_1 = 10^{-4}$	-0.032 ± 0.267	0.009 ± 0.067	0.968 ± 0.045	0.94 ± 0.041	0.57%
		$\lambda_1 = 2 \times 10^{-4}$	-0.072 ± 0.293	0.024 ± 0.062	0.973 ± 0.043	0.947 ± 0.04	1.89%
	10^{-3}	None	-0.058 ± 0.21	0.02 ± 0.058	0.973 ± 0.042	0.945 ± 0.04	−0.35%
		$\lambda_2 = 10^{-4}$	-0.066 ± 0.224	0.024 ± 0.063	0.979 ± 0.044	0.944 ± 0.035	0.54%
		$\lambda_2 = 2 \times 10^{-4}$	-0.039 ± 0.252	0.022 ± 0.063	0.979 ± 0.047	0.946 ± 0.039	1.23%
		$\lambda_1 = 10^{-4}$	-0.074 ± 0.281	0.021 ± 0.062	0.98 ± 0.044	0.947 ± 0.038	2.14%
		$\lambda_1 = 2 \times 10^{-4}$	-0.057 ± 0.3	0.014 ± 0.062	0.974 ± 0.044	0.946 ± 0.039	2.12%
500	10^{-4}	None	-0.034 ± 0.264	0.017 ± 0.065	0.974 ± 0.044	0.948 ± 0.042	1.15%
		$\lambda_2 = 10^{-4}$	-0.043 ± 0.193	0.017 ± 0.066	0.972 ± 0.041	0.944 ± 0.039	−0.77%
		$\lambda_2 = 2 \times 10^{-4}$	-0.047 ± 0.196	0.016 ± 0.059	0.976 ± 0.044	0.945 ± 0.034	−0.37%
		$\lambda_1 = 10^{-4}$	-0.064 ± 0.225	0.012 ± 0.061	0.969 ± 0.041	0.947 ± 0.038	−0.41%
		$\lambda_1 = 2 \times 10^{-4}$	-0.069 ± 0.246	0.022 ± 0.064	0.976 ± 0.044	0.945 ± 0.041	0.88%
	5×10^{-4}	None	-0.052 ± 0.243	0.008 ± 0.06	0.974 ± 0.042	0.946 ± 0.04	0.54%
		$\lambda_2 = 10^{-4}$	-0.064 ± 0.208	0.016 ± 0.057	0.969 ± 0.04	0.942 ± 0.037	−0.79%
		$\lambda_2 = 2 \times 10^{-4}$	-0.053 ± 0.208	0.015 ± 0.061	0.975 ± 0.046	0.944 ± 0.037	−0.15%
		$\lambda_1 = 10^{-4}$	-0.056 ± 0.249	0.022 ± 0.065	0.967 ± 0.039	0.944 ± 0.034	0.0%
		$\lambda_1 = 2 \times 10^{-4}$	-0.057 ± 0.273	0.022 ± 0.07	0.974 ± 0.043	0.946 ± 0.038	1.37%
	10^{-3}	None	-0.056 ± 0.227	0.019 ± 0.056	0.975 ± 0.054	0.944 ± 0.04	0.3%
		$\lambda_2 = 10^{-4}$	-0.062 ± 0.189	0.012 ± 0.06	0.979 ± 0.048	0.946 ± 0.035	−0.22%
		$\lambda_2 = 2 \times 10^{-4}$	-0.044 ± 0.229	0.015 ± 0.066	0.982 ± 0.045	0.946 ± 0.04	0.96%
		$\lambda_1 = 10^{-4}$	-0.076 ± 0.27	0.014 ± 0.063	0.97 ± 0.04	0.946 ± 0.033	0.88%
		$\lambda_1 = 2 \times 10^{-4}$	-0.078 ± 0.285	0.011 ± 0.064	0.974 ± 0.041	0.947 ± 0.037	1.73%

TABLE V. Means and standard deviations for the biases b_{H_0,q_0}, posterior-width ratios f_{H_0,q_0} and percentage increase in H_0 uncertainty for all combinations of batch size, learning rate and regularization in the selection case, using $[n_{\text{train}}, n_{\text{val}}] = [5000, 2000]$.

SELECTION CASE							
n_{batch}	α	Regularizer	$b_{H_0}[\text{km s}^{-1}\text{ Mpc}^{-1}]$	b_{q_0}	$f_\sigma^{H_0}$	$f_\sigma^{q_0}$	$\%\hat{\sigma}^{H_0}_{\text{incr}}$
TRAINING and VALIDATION parameters: $[n_{\text{train}}, n_{\text{val}}] = [5000, 2000]$							
100	10^{-4}	None	-0.153 ± 1.714	0.014 ± 0.136	1.77 ± 0.139	1.019 ± 0.076	144.02%
		$\lambda_2 = 10^{-4}$	0.043 ± 0.403	0.02 ± 0.123	1.059 ± 0.044	1.012 ± 0.04	13.05%
		$\lambda_2 = 2 \times 10^{-4}$	0.021 ± 0.426	0.016 ± 0.114	1.047 ± 0.037	1.013 ± 0.044	12.74%
		$\lambda_1 = 10^{-4}$	-0.015 ± 0.338	0.014 ± 0.115	1.013 ± 0.039	1.005 ± 0.044	6.53%
		$\lambda_1 = 2 \times 10^{-4}$	-0.029 ± 0.365	0.008 ± 0.119	1.029 ± 0.038	1.008 ± 0.044	8.97%
	5×10^{-4}	None	-0.309 ± 1.657	0.007 ± 0.126	1.838 ± 0.153	1.013 ± 0.032	145.28%
		$\lambda_2 = 10^{-4}$	0.059 ± 0.423	0.009 ± 0.116	1.018 ± 0.043	1.007 ± 0.057	9.95%
		$\lambda_2 = 2 \times 10^{-4}$	-0.015 ± 0.4	0.022 ± 0.118	1.033 ± 0.041	1.009 ± 0.044	10.47%
		$\lambda_1 = 10^{-4}$	-0.02 ± 0.313	-0.001 ± 0.122	1.014 ± 0.041	1.005 ± 0.058	5.9%
		$\lambda_1 = 2 \times 10^{-4}$	0.0 ± 0.393	0.003 ± 0.115	1.036 ± 0.038	1.009 ± 0.042	10.5%
	10^{-3}	None	-0.006 ± 0.526	0.011 ± 0.132	1.055 ± 0.055	1.015 ± 0.089	17.38%
		$\lambda_2 = 10^{-4}$	-0.03 ± 0.408	0.011 ± 0.114	1.013 ± 0.044	1.005 ± 0.056	8.93%
		$\lambda_2 = 2 \times 10^{-4}$	0.017 ± 0.387	0.02 ± 0.121	1.023 ± 0.046	1.009 ± 0.063	9.09%
		$\lambda_1 = 10^{-4}$	0.014 ± 0.357	0.008 ± 0.119	1.018 ± 0.042	1.006 ± 0.051	7.67%
		$\lambda_1 = 2 \times 10^{-4}$	-0.003 ± 0.476	0.0 ± 0.12	1.055 ± 0.048	1.011 ± 0.056	15.32%
500	10^{-4}	None	0.022 ± 1.961	0.005 ± 0.136	2.028 ± 0.176	1.005 ± 0.028	179.37%
		$\lambda_2 = 10^{-4}$	0.011 ± 0.455	0.016 ± 0.119	1.044 ± 0.035	1.009 ± 0.051	13.52%
		$\lambda_2 = 2 \times 10^{-4}$	-0.009 ± 0.417	0.011 ± 0.114	1.056 ± 0.04	1.008 ± 0.048	13.25%
		$\lambda_1 = 10^{-4}$	-0.002 ± 0.334	0.019 ± 0.116	1.025 ± 0.04	1.01 ± 0.049	7.63%
		$\lambda_1 = 2 \times 10^{-4}$	-0.018 ± 0.377	0.013 ± 0.125	1.033 ± 0.037	1.016 ± 0.03	9.66%
	5×10^{-4}	None	0.189 ± 2.115	0.027 ± 0.127	1.97 ± 0.134	1.017 ± 0.042	185.92%
		$\lambda_2 = 10^{-4}$	-0.039 ± 0.453	0.02 ± 0.12	1.048 ± 0.039	1.012 ± 0.056	13.75%
		$\lambda_2 = 2 \times 10^{-4}$	0.011 ± 0.408	0.015 ± 0.119	1.039 ± 0.035	1.01 ± 0.046	11.34%
		$\lambda_1 = 10^{-4}$	0.001 ± 0.313	0.012 ± 0.128	1.025 ± 0.038	1.013 ± 0.036	6.99%
		$\lambda_1 = 2 \times 10^{-4}$	-0.037 ± 0.359	0.014 ± 0.12	1.038 ± 0.044	1.01 ± 0.058	9.62%
	10^{-3}	None	-0.082 ± 1.972	0.016 ± 0.117	2.027 ± 0.173	1.012 ± 0.031	180.05%
		$\lambda_2 = 10^{-4}$	-0.006 ± 0.451	0.011 ± 0.119	1.046 ± 0.036	1.005 ± 0.062	13.57%
		$\lambda_2 = 2 \times 10^{-4}$	-0.027 ± 0.371	0.017 ± 0.124	1.046 ± 0.042	1.005 ± 0.058	10.71%
		$\lambda_1 = 10^{-4}$	0.051 ± 0.329	0.011 ± 0.137	1.019 ± 0.053	1.011 ± 0.058	6.91%
		$\lambda_1 = 2 \times 10^{-4}$	0.004 ± 0.348	0.011 ± 0.113	1.027 ± 0.038	1.003 ± 0.05	8.2%

TABLE VI. Means and standard deviations for the biases b_{H_0,q_0}, posterior-width ratios f_{H_0,q_0} and percentage increase in H_0 uncertainty for all combinations of batch size, learning rate and regularization in the selection case, using $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [500000, 100000]$.

SELECTION CASE							
n_{batch}	α	Regularizer	$b_{H_0}[\mathrm{km\,s^{-1}\ Mpc^{-1}}]$	b_{q_0}	$f_{\sigma}^{H_0}$	$f_{\sigma}^{q_0}$	$\%\hat{\sigma}_{incr}^{H_0}$
TRAINING and VALIDATION parameters: $[n_{\mathrm{train}}, n_{\mathrm{val}}] = [500000, 100000]$							
100	10^{-4}	None	-0.033 ± 0.278	0.023 ± 0.082	0.97 ± 0.037	0.999 ± 0.045	0.78%
		$\lambda_2 = 10^{-4}$	-0.032 ± 0.184	0.022 ± 0.092	0.976 ± 0.031	1.006 ± 0.036	−0.73%
		$\lambda_2 = 2 \times 10^{-4}$	-0.019 ± 0.195	0.017 ± 0.085	0.981 ± 0.036	1.002 ± 0.037	−0.04%
		$\lambda_1 = 10^{-4}$	-0.025 ± 0.186	0.021 ± 0.085	0.978 ± 0.033	1.003 ± 0.039	−0.55%
		$\lambda_1 = 2 \times 10^{-4}$	-0.044 ± 0.214	0.018 ± 0.087	0.984 ± 0.036	1.005 ± 0.042	0.57%
	5×10^{-4}	None	0.01 ± 0.207	0.02 ± 0.087	0.968 ± 0.038	1.001 ± 0.04	−1.12%
		$\lambda_2 = 10^{-4}$	-0.033 ± 0.177	0.02 ± 0.092	0.979 ± 0.039	1.003 ± 0.043	−0.56%
		$\lambda_2 = 2 \times 10^{-4}$	-0.026 ± 0.199	0.015 ± 0.088	0.979 ± 0.037	1.002 ± 0.04	−0.21%
		$\lambda_1 = 10^{-4}$	-0.028 ± 0.198	0.019 ± 0.081	0.988 ± 0.037	1.001 ± 0.042	0.64%
		$\lambda_1 = 2 \times 10^{-4}$	-0.047 ± 0.269	0.018 ± 0.084	0.989 ± 0.034	1.0 ± 0.045	2.37%
	10^{-3}	None	0.0 ± 0.183	0.026 ± 0.091	0.965 ± 0.03	1.004 ± 0.038	−1.88%
		$\lambda_2 = 10^{-4}$	-0.007 ± 0.184	0.014 ± 0.088	0.98 ± 0.034	1.005 ± 0.044	−0.35%
		$\lambda_2 = 2 \times 10^{-4}$	-0.015 ± 0.193	0.015 ± 0.093	0.982 ± 0.037	1.004 ± 0.044	−0.0%
		$\lambda_1 = 10^{-4}$	-0.053 ± 0.242	0.015 ± 0.087	0.996 ± 0.033	1.001 ± 0.045	2.39%
		$\lambda_1 = 2 \times 10^{-4}$	-0.031 ± 0.263	0.015 ± 0.095	0.991 ± 0.036	1.001 ± 0.048	2.44%
500	10^{-4}	None	-0.037 ± 0.267	0.022 ± 0.084	0.98 ± 0.041	0.998 ± 0.036	1.42%
		$\lambda_2 = 10^{-4}$	-0.038 ± 0.199	0.028 ± 0.109	0.976 ± 0.034	1.01 ± 0.035	−0.51%
		$\lambda_2 = 2 \times 10^{-4}$	-0.02 ± 0.194	0.021 ± 0.095	0.971 ± 0.034	1.005 ± 0.036	−1.05%
		$\lambda_1 = 10^{-4}$	-0.022 ± 0.178	0.015 ± 0.093	0.978 ± 0.036	1.003 ± 0.039	−0.7%
		$\lambda_1 = 2 \times 10^{-4}$	-0.025 ± 0.186	0.012 ± 0.089	0.979 ± 0.036	1.002 ± 0.035	−0.47%
	5×10^{-4}	None	-0.045 ± 0.277	0.019 ± 0.09	0.982 ± 0.039	1.002 ± 0.037	1.83%
		$\lambda_2 = 10^{-4}$	-0.013 ± 0.18	0.019 ± 0.086	0.977 ± 0.035	1.006 ± 0.038	−0.68%
		$\lambda_2 = 2 \times 10^{-4}$	-0.02 ± 0.182	0.014 ± 0.087	0.981 ± 0.032	1.005 ± 0.039	−0.32%
		$\lambda_1 = 10^{-4}$	-0.03 ± 0.196	0.018 ± 0.079	0.982 ± 0.032	1.001 ± 0.036	0.04%
		$\lambda_1 = 2 \times 10^{-4}$	-0.047 ± 0.233	0.012 ± 0.089	0.987 ± 0.038	1.004 ± 0.036	1.33%
	10^{-3}	None	-0.013 ± 0.22	0.021 ± 0.089	0.962 ± 0.042	1.003 ± 0.044	−1.43%
		$\lambda_2 = 10^{-4}$	-0.006 ± 0.201	0.015 ± 0.086	0.98 ± 0.033	1.004 ± 0.041	−0.07%
		$\lambda_2 = 2 \times 10^{-4}$	-0.01 ± 0.199	0.021 ± 0.083	0.979 ± 0.043	1.003 ± 0.042	−0.18%
		$\lambda_1 = 10^{-4}$	-0.026 ± 0.212	0.015 ± 0.084	0.986 ± 0.037	1.0 ± 0.044	0.77%
		$\lambda_1 = 2 \times 10^{-4}$	-0.036 ± 0.25	0.012 ± 0.09	0.993 ± 0.037	1.005 ± 0.043	2.28%

[1] A. G. Riess, S. Casertano, W. Yuan, J. B. Bowers, L. Macri, J. C. Zinn, and D. Scolnic, Cosmic distances calibrated to 1% precision with Gaia EDR3 parallaxes and Hubble space telescope photometry of 75 milky way cepheids confirm tension with ΛCDM, Astrophys. J. Lett. **908**, L6 (2021).

[2] S. Birrer *et al.*, H0LiCOW—IX. Cosmographic analysis of the doubly imaged quasar SDSS 1206 + 4332 and a new measurement of the Hubble constant, Mon. Not. R. Astron. Soc. **484**, 4726 (2019).

[3] K. C. Wong *et al.*, H0LiCOW—XIII. A 2.4 percent measurement of H_0 from lensed quasars: 5.3σ tension between early- and late-Universe probes, Mon. Not. R. Astron. Soc. **498**, 1420 (2020).

[4] Planck Collaboration, Planck 2018 results. VI. Cosmological parameters, Astron. Astrophys. **641**, A6 (2020).

[5] G. E. Addison, D. J. Watts, C. L. Bennett, M. Halpern, G. Hinshaw, and J. L. Weiland, Elucidating ΛCDM: Impact of baryon acoustic oscillation measurements on the Hubble constant discrepancy, Astrophys. J. **853**, 119 (2018).

[6] Dark Energy Survey and South Pole Telescope Collaborations, Dark energy survey year 1 results: A precise H_0 estimate from DES Y1, BAO, and D/H data, Mon. Not. R. Astron. Soc. **480**, 3879 (2018).

[7] O. H. E. Philcox, M. M. Ivanov, M. Simonović, and M. Zaldarriaga, Combining full-shape and BAO analyses of galaxy power spectra: A 1.6% CMB-independent constraint on H_0, J. Cosmol. Astropart. Phys. 05 (2020) 032.

[8] J. L. Bernal, L. Verde, and A. G. Riess, The trouble with H_0, J. Cosmol. Astropart. Phys. 10 (2016) 019.

[9] L. Verde, T. Treu, and A. Riess, Tensions between the early and the late Universe (2019), Nat. Astron. **3**, 891 (2019).

[10] J. L. Bernal, L. Verde, R. Jimenez, M. Kamionkowski, D. Valcin, and B. D. Wandelt, The trouble beyond H_0 and the new cosmic triangles, Phys. Rev. D **103**, 103533 (2021).

[11] M. Rigault *et al.*, Confirmation of a star formation bias in type ia supernova distances and its effect on measurement of the Hubble constant, Astrophys. J. **802**, 20 (2015).

[12] D. O. Jones, A. G. Riess, and D. M. Scolnic, Reconsidering the effects of local star formation on type ia supernova cosmology, Astrophys. J. **812**, 31 (2015).

[13] M. Rigault *et al.* (Nearby Supernova Factory Collaboration), Strong dependence of type ia supernova standardization on the local specific star formation rate, Astron. Astrophys. **644**, A176 (2020).

[14] D. O. Jones, A. G. Riess, D. M. Scolnic, Y. C. Pan, E. Johnson, D. A. Coulter, K. G. Dettman, M. M. Foley, R. J. Foley, M. E. Huber, S. W. Jha, C. D. Kilpatrick, R. P. Kirshner, A. Rest, A. S. B. Schultz, and M. R. Siebert, Should type ia supernova distances be corrected for their local environments? Astrophys. J. **867**, 108 (2018).

[15] W. L. Freedman, B. F. Madore, T. Hoyt, I. S. Jang, R. Beaton, M. G. Lee, A. Monson, J. Neeley, and J. Rich, Calibration of the tip of the red giant branch, Astrophys. J. **891**, 57 (2020).

[16] D. Brout and D. Scolnic, It's dust: Solving the mysteries of the intrinsic scatter and host-galaxy dependence of standardized type ia supernova brightnesses, Astrophys. J. **909**, 26 (2021).

[17] E. Di Valentino, O. Mena, S. Pan, L. Visinelli, W. Yang, A. Melchiorri, D. F. Mota, A. G. Riess, and J. Silk, In the realm of the Hubble tension—A review of solutions, Classical Quantum Gravity **38**, 153001 (2021).

[18] B. F. Schutz, Determining the Hubble constant from gravitational wave observations, Nature (London) **323**, 310 (1986).

[19] D. E. Holz and S. A. Hughes, Using gravitational-wave standard sirens, Astrophys. J. **629**, 15 (2005).

[20] N. Dalal, D. E. Holz, S. A. Hughes, and B. Jain, Short GRB and binary black hole standard sirens as a probe of dark energy, Phys. Rev. D **74**, 063006 (2006).

[21] S. Nissanke, D. E. Holz, S. A. Hughes, N. Dalal, and J. L. Sievers, Exploring short gamma-ray bursts as gravitational-wave standard sirens, Astrophys. J. **725**, 496 (2010).

[22] S. R. Taylor, J. R. Gair, and I. Mandel, Cosmology using advanced gravitational-wave detectors alone, Phys. Rev. D **85**, 023535 (2012).

[23] C. Messenger and J. Read, Measuring a Cosmological Distance-Redshift Relationship Using only Gravitational Wave Observations of Binary Neutron Star Coalescences, Phys. Rev. Lett. **108**, 091101 (2012).

[24] S. Nissanke, D. E. Holz, N. Dalal, S. A. Hughes, J. L. Sievers, and C. M. Hirata, Determining the Hubble constant from gravitational wave observations of merging compact binaries, arXiv:1307.2638.

[25] M. Oguri, Measuring the distance-redshift relation with the cross-correlation of gravitational wave standard sirens and galaxies, Phys. Rev. D **93**, 083511 (2016).

[26] W. Del Pozzo, T. G. F. Li, and C. Messenger, Cosmological inference using only gravitational wave observations of binary neutron stars, Phys. Rev. D **95**, 043502 (2017).

[27] S. Vitale and H.-Y. Chen, Measuring the Hubble Constant with Neutron Star Black Hole Mergers, Phys. Rev. Lett. **121**, 021303 (2018).

[28] N. Seto and K. Kyutoku, Prospects of the local Hubble parameter measurement using gravitational waves from double neutron stars, Mon. Not. R. Astron. Soc. **475**, 4133 (2018).

[29] S. M. Feeney, H. V. Peiris, A. R. Williamson, S. M. Nissanke, D. J. Mortlock, J. Alsing, and D. Scolnic, Prospects for Resolving the Hubble Constant Tension with Standard Sirens, Phys. Rev. Lett. **122**, 061105 (2019).

[30] R. Gray *et al.*, Cosmological inference using gravitational wave standard sirens: A mock data analysis, Phys. Rev. D **101**, 122001 (2020).

[31] S. M. Feeney, H. V. Peiris, S. M. Nissanke, and D. J. Mortlock, Prospects for Measuring the Hubble Constant with Neutron-Star-Black-Hole Mergers, Phys. Rev. Lett. **126**, 171102 (2021).

[32] 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, arXiv:2007.05579.

[33] B. Abbott *et al.* (KAGRA, LIGO Scientific, VIRGO Collaborations), Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA, Living Rev. Relativity **21**, 3 (2018).

[34] H.-Y. Chen, M. Fishbach, and D. E. Holz, A two percent Hubble constant measurement from standard sirens within five years, Nature (London) **562**, 545 (2018).

[35] B. 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).

[36] B. P. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, Phys. Rev. Lett. **119**, 161101 (2017).

[37] B. P. Abbott *et al.* (LIGO Scientific, Virgo, Fermi GBM, INTEGRAL, IceCube, AstroSat Cadmium Zinc Telluride Imager Team, IPN, Insight-Hxmt, ANTARES, Swift, AGILE Team, 1M2H Team, Dark Energy Camera GW-EM, DES, DLT40, GRAWITA, Fermi-LAT, ATCA, ASKAP, Las Cumbres Observatory Group, OzGrav, DWF (Deeper Wider Faster Program), AST3, CAASTRO, VINROUGE, MASTER, J-GEM, GROWTH, JAGWAR, CaltechNRAO, TTU-NRAO, NuSTAR, Pan-STARRS, MAXI Team, TZAC Consortium, KU, Nordic Optical Telescope, ePESSTO, GROND, Texas Tech University, SALT Group, TOROS, BOOTES, MWA, CALET, IKI-GW Follow-up, H.E.S.S., LOFAR, LWA, HAWC, Pierre Auger, ALMA, Euro VLBI Team, Pi of Sky, Chandra Team at McGill University, DFN, ATLAS Telescopes, High Time

Resolution Universe Survey, RIMAS, RATIR, SKA South Africa/MeerKAT Collaborations), Multi-messenger observations of a binary neutron star merger, Astrophys. J. Lett. **848**, L12 (2017).

[38] K. G. Malmquist, On some relations in stellar statistics, Medd. Lunds Astron. Obs. Ser. I **100**, 1 (1922), https://ui.adsabs.harvard.edu/abs/1922MeLuF.100....1M.

[39] K. G. Malmquist, A contribution to the problem of determining the distribution in space of the stars, Medd. Lunds Astron. Obs. Ser. I **106**, 1 (1925), https://ui.adsabs.harvard.edu/abs/1925MeLuF.106....1M.

[40] T. J. Loredo, Accounting for source uncertainties in analyses of astronomical survey data, AIP Conf. Proc. **735**, 195 (2004).

[41] 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).

[42] D. J. Mortlock, S. M. Feeney, H. V. Peiris, A. R. Williamson, and S. M. Nissanke, Unbiased Hubble constant estimation from binary neutron star mergers, Phys. Rev. D **100**, 103523 (2019).

[43] V. Tiwari, Estimation of the sensitive volume for gravitational-wave source populations using weighted Monte Carlo integration, Classical Quantum Gravity **35**, 145009 (2018).

[44] W. M. Farr, Accuracy requirements for empirically measured selection functions, Res. Notes Am. Astron. Soc. **3**, 66 (2019).

[45] C. Talbot and E. Thrane, Fast, flexible, and accurate evaluation of Malmquist bias with machine learning: Preparing for the pending flood of gravitational-wave detections, arXiv:2012.01317.

[46] G. Papamakarios, D. C. Sterratt, and I. Murray, Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows, arXiv:1805.07226.

[47] J.-M. Lueckmann, G. Bassetto, T. Karaletsos, and J. H. Macke, Likelihood-free inference with emulator networks, Proc. Mech. Learn. Res. **96**, 32 (2019), https://ui.adsabs.harvard.edu/abs/2018arXiv180509294L.

[48] J. Alsing, B. Wandelt, and S. Feeney, Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology, Mon. Not. R. Astron. Soc. **477**, 2874 (2018).

[49] J. Alsing, T. Charnock, S. Feeney, and B. Wandelt, Fast likelihood-free cosmology with neural density estimators and active learning, Mon. Not. R. Astron. Soc. **488**, 4440 (2019).

[50] D. George and E. A. Huerta, Deep learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data, Phys. Lett. B **778**, 64 (2018).

[51] H. Shen, E. A. Huerta, E. O’Shea, P. Kumar, and Z. Zhao, Statistically-informed deep learning for gravitational wave parameter estimation, arXiv:1903.01998.

[52] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy, arXiv:1909.06296.

[53] A. J. K. Chua and M. Vallisneri, Learning Bayesian Posteriors with Neural Networks for Gravitational-Wave Inference, Phys. Rev. Lett. **124**, 041102 (2020).

[54] S. R. Green, C. Simpson, and J. Gair, Gravitational-wave parameter estimation with autoregressive neural network flows, Phys. Rev. D **102**, 104057 (2020).

[55] S. R. Green and J. Gair, Complete parameter inference for GW150914 using deep learning, Mach. Learn. Sci. Technol. **2**, 03LT01 (2021).

[56] A. Delaunoy, A. Wehenkel, T. Hinderer, S. Nissanke, C. Weniger, A. R. Williamson, and G. Louppe, Lightning-fast gravitational wave parameter inference through neural amortization, arXiv:2010.12931.

[57] R. Abbott *et al.* (The LIGO Scientific Collaboration, the Virgo Collaboration), Population properties of compact objects from the second LIGO-Virgo gravitational-wave transient catalog, arXiv:2010.14533.

[58] A. G. Kim, Characterizing the sample selection for supernova cosmology, Open J. Astrophys. **4**, 2 (2021).

[59] M. Visser, Jerk and the cosmological equation of state, Classical Quantum Gravity **21**, 2603 (2004).

[60] M. D. Hoffman and A. Gelman, The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo, arXiv:1111.4246.

[61] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, Stan: A probabilistic programming language, J. Stat. Softw. **76**, 1 (2017).

[62] S. D. Team, PyStan: The PYTHON interface to Stan, Version 2.17.1.0. (2018), https://pystan.readthedocs.io/en/latest/.

[63] C. M. Bishop, Mixture density networks, Technical Report, 1994.

[64] G. Papamakarios, T. Pavlakou, and I. Murray, Masked autoregressive flow for density estimation, arXiv:1705.07057.

[65] J. Alsing and B. Wandelt, Generalized massive optimal data compression, Mon. Not. R. Astron. Soc. **476**, L60 (2018).

[66] J. Alsing and B. Wandelt, Nuisance hardened data compression for fast likelihood-free inference, Mon. Not. R. Astron. Soc. **488**, 5093 (2019).

[67] T. Charnock, G. Lavaux, and B. D. Wandelt, Automatic physical inference with information maximizing neural networks, Phys. Rev. D **97**, 083004 (2018).

[68] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).

[69] A. L. Maas, A. Y. Hannun, and A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in *Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA* (JMLR, 2013).

[70] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference,and Prediction* (Springer, 2009).

[71] LIGO Scientific Collaboration, LIGO Algorithm Library—LALSuite, free software (GPL) (2018).

[72] B. Sathyaprakash *et al.*, Scientific objectives of Einstein telescope, Classical Quantum Gravity **29**, 124013 (2012); **30**, 079501(E) (2013).

[73] B. P. Abbott *et al.* (LIGO Scientific Collaboration), Exploring the sensitivity of next generation gravitational wave detectors, Classical Quantum Gravity **34**, 044001 (2017).

[74] L. Sun *et al.*, Characterization of systematic error in Advanced LIGO calibration, Classical Quantum Gravity **37**, 225008 (2020).

[75] S. Rosswog, U. Feindt, O. Korobkin, M. R. Wu, J. Sollerman, A. Goobar, and G. Martinez-Pinedo, Detectability of compact binary merger macronovae, Classical Quantum Gravity **34**, 104001 (2017).

[76] D. Scolnic *et al.* (DES Collaboration), How many kilonovae can be found in past, present, and future survey data sets?, Astrophys. J. Lett. **852**, L3 (2018).

[77] P. S. Cowperthwaite, V. A. Villar, D. M. Scolnic, and E. Berger, LSST target-of-opportunity observations of gravitational-wave events: Essential and efficient, Astrophys. J. **874**, 88 (2019).

[78] C. N. Setzer, R. Biswas, H. V. Peiris, S. Rosswog, O. Korobkin, and R. T. Wollaeger (LSST Dark Energy Science Collaboration), Serendipitous discoveries of kilonovae in the LSST main survey: Maximizing detections of subthreshold gravitational wave events, Mon. Not. R. Astron. Soc. **485**, 4260 (2019).

[79] H.-Y. Chen, Systematic Uncertainty of Standard Sirens from the Viewing Angle of Binary Neutron Star Inspirals, Phys. Rev. Lett. **125**, 201301 (2020).

[80] S. Mastrogiovanni, R. Duque, E. Chassande-Mottin, F. Daigne, and R. Mochkovitch, What role will binary neutron star merger afterglows play in multimessenger cosmology?, arXiv:2012.12836.

[81] G. Raaijmakers, S. Nissanke, F. Foucart, M. M. Kasliwal, M. Bulla, R. Fernandez, A. Henkel, T. Hinderer, K. Hotokezaka, K. Lukošiūtė, T. Venumadhav, S. Antier, M. W. Coughlin, T. Dietrich, and T. D. P. Edwards, The challenges ahead for multimessenger analyses of gravitational waves and kilonova: A case study on GW190425, arXiv:2102.11569.

[82] LIGO Scientific and Virgo Collaborations, GW170817: Measurements of Neutron Star Radii and Equation of State, Phys. Rev. Lett. **121**, 161101 (2018).

[83] N. Farrow, X.-J. Zhu, and E. Thrane, The mass distribution of galactic double neutron stars, Astrophys. J. **876**, 18 (2019).

[84] P. Landry, R. Essick, and K. Chatziioannou, Nonparametric constraints on neutron star matter with existing and upcoming gravitational wave and pulsar observations, Phys. Rev. D **101**, 123007 (2020).

[85] S. Galaudage, C. Adamcewicz, X.-J. Zhu, S. Stevenson, and E. Thrane, Heavy double neutron stars: Birth, midlife, and death, Astrophys. J. Lett. **909**, L19 (2021).

[86] S. Mastrogiovanni, K. Leyde, C. Karathanasis, E. Chassande-Mottin, D. A. Steer, J. Gair, A. Ghosh, R. Gray, S. Mukherjee, and S. Rinaldi, Cosmology in the dark: On the importance of source population models for gravitational-wave cosmology, arXiv:2103.14663.

[87] https://github.com/frgerardi/LFIH0_BNS.git.