

## Adapting to noise distribution shifts in flow-based gravitational-wave inference

Jonas Wildberger<sup>1,\*</sup>, Maximilian Dax<sup>1,†</sup>, Stephen R. Green<sup>2,3,‡</sup>, Jonathan Gair,<sup>3</sup> Michael Pürrer<sup>3,4,5</sup>,  
Jakob H. Macke,<sup>1,6</sup> Alessandra Buonanno,<sup>3,7</sup> and Bernhard Schölkopf<sup>1</sup>

<sup>1</sup>Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, 72076 Tübingen, Germany

<sup>2</sup>School of Mathematical Sciences, University of Nottingham,  
University Park, Nottingham NG7 2RD, United Kingdom

<sup>3</sup>Max Planck Institute for Gravitational Physics (Albert Einstein Institute),  
Am Mühlenberg 1, 14476 Potsdam, Germany

<sup>4</sup>Department of Physics, East Hall, University of Rhode Island, Kingston, Rhode Island 02881, USA

<sup>5</sup>URI Research Computing, Tyler Hall, University of Rhode Island, Kingston, Rhode Island 02881, USA

<sup>6</sup>Machine Learning in Science, University of Tübingen, 72076 Tübingen, Germany

<sup>7</sup>Department of Physics, University of Maryland, College Park, Maryland 20742, USA

 (Received 25 November 2022; accepted 14 March 2023; published 26 April 2023)

Deep learning techniques for gravitational-wave parameter estimation have emerged as a fast alternative to standard samplers—producing results of comparable accuracy. These approaches (e.g., DINGO) enable amortized inference by training a normalizing flow to represent the Bayesian posterior conditional on observed data. By conditioning also on the noise power spectral density (PSD) they can even account for changing detector characteristics. However, training such networks requires knowing in advance the distribution of PSDs expected to be observed and therefore can only take place once all data to be analyzed have been gathered. Here, we develop a probabilistic model to forecast future PSDs, greatly increasing the temporal scope of trained deep learning models. Using PSDs from the second LIGO-Virgo observing run (O2)—plus just a *single* PSD from the beginning of the third (O3)—we show that we can train a DINGO network to perform accurate inference throughout O3 (on 37 real events). We therefore expect this approach to be a key component to enable the use of deep learning techniques for low-latency analyses of gravitational waves.

DOI: [10.1103/PhysRevD.107.084046](https://doi.org/10.1103/PhysRevD.107.084046)

### I. INTRODUCTION

Detector noise plays a crucial role in interpreting observations of gravitational waves (GWs). In its simplest form, noise is assumed to be additive, stationary, and Gaussian. This means that it is characterized in frequency domain by its power spectral density (PSD)  $S_n(f)$ . The GW likelihood for parameters  $\theta$  is then the probability that, after a proposed signal  $h(\theta)$  is subtracted from data  $d$ , the residual is noise satisfying these assumptions, i.e.,

$$p(d|\theta, S_n) \propto \exp\left(-\frac{1}{2}(d - h(\theta)|d - h(\theta))_{S_n}\right), \quad (1)$$

where  $(\cdot|\cdot)$  is the noise-weighted inner product,

$$(a|b)_{S_n} = 4 \sum_I \Re \int_{f_{\min}}^{f_{\max}} \frac{\hat{a}_I^*(f) \hat{b}_I(f)}{S_{n,I}(f)} df. \quad (2)$$

Here,  $\hat{a}$  denotes the Fourier transform of  $a$ , and the sum runs over interferometers  $I$ .

Although the LIGO [1], Virgo [2], and KAGRA [3–5] detectors are mostly stable during an observing run, the noise spectrum does vary over time and across detectors. Moreover, between observing runs, detectors are upgraded, resulting in reduced noise levels and increased sensitivity. The particular PSDs at the time of an event must therefore be estimated and taken into account when performing inference. Estimation of a PSD is typically carried out using either data adjacent to an event (off-source, Welch method [6]) or by jointly modeling the signal and noise from the on-source data (e.g., BayesWave [7,8]). The PSD is

\*wildberger.jonas@tuebingen.mpg.de

†maximilian.dax@tuebingen.mpg.de

‡stephen.green2@nottingham.ac.uk

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Open access publication funded by the Max Planck Society.

then inserted into the likelihood and samples are drawn from the posterior  $p(\theta|d, S_n)$  using Markov chain Monte Carlo [9] or nested sampling methods [10–12].

An emerging alternative to classical likelihood-based inference is simulation-based inference using probabilistic deep learning [13–22]. Technologies such as normalizing flows enable neural networks to describe complex conditional probability distributions. A conditional density estimator  $q(\theta|d)$  can be trained using simulated data to approximate  $p(\theta|d)$  such that once a detection is made, samples can be drawn in seconds. To incorporate varying noise PSDs into these methods, the estimate of  $S_n$  is provided as additional context to the network, i.e.,  $q(\theta|d, S_n)$ . During training, random PSDs are drawn from an empirical distribution  $p(S_n)$  estimated from signal-free data during an observing run. Simulated data are then constructed based on these PSDs, and the PSD is provided as context. At inference time, the network is effectively “tuned” to the interferometers at the time of detection. This approach (called DINGO [19]) fully amortizes training costs across all detections within an observing run and has been shown to produce results nearly indistinguishable from classical samplers.

The approach described requires access to the empirical distribution  $p(S_n)$  of PSDs during an observing run. This makes it unsuitable for *online* parameter estimation (e.g., to provide alerts to electromagnetic telescopes) since the distribution  $p(S_n)$  covering future observations is unavailable at the time of network training. Thus, a network trained with an empirical PSD distribution can only be used for a limited time—once the PSDs change too much, the measured data become out of distribution. Such a disagreement between training and inference distributions can lead to inaccurate results (Fig. 1). Here, we address this problem and provide a solution for training DINGO models robustly to better adapt to shifting distributions. Although our main motivation is the DINGO pipeline, the proposed method can in fact be used to improve the robustness of any machine learning method for gravitational-wave search or inference that requires noise PSDs for training.

During training, DINGO models  $q(\theta|d, S_n)$  estimate a distribution conditional on (as opposed to marginalized over) the PSD  $S_n$ . We are therefore not restricted to using (an approximation to) the real distribution  $p(S_n)$  of PSDs; instead, we can use a synthetic distribution  $q(S_n)$  whose support contains that of  $p(S_n)$ . In other words, if  $\text{supp } p(S_n) \subseteq \text{supp } q(S_n)$ , then the DINGO model trained with  $q(S_n)$  can be used for the entire observing run.

In this work, we develop a parametrized latent variable model for  $q(S_n)$ , which we fit to PSDs from a previous observing run. It is then straightforward to modify this distribution of PSDs via operations on the latent space. In particular, we use a one-shot observation from an upgraded detector to shift the latent space distribution coarsely to the expected noise level. Further, we broaden the spread of

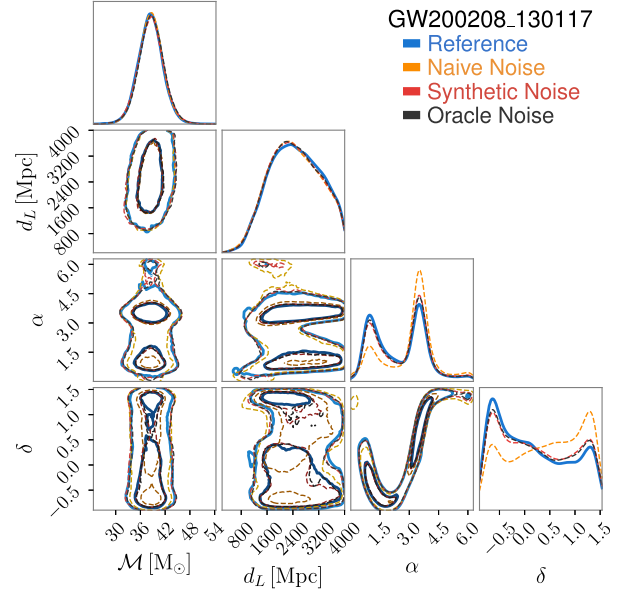


FIG. 1. Posterior for GW200208\_130117. Results from DINGO models trained only with empirically estimated detector-noise PSDs from the beginning of an observing run (orange) may deviate visibly from the reference (blue, obtained using importance sampling [23]). DINGO models trained with our proposed synthetic PSD model (red) achieve much more accurate results, on par with using PSDs estimated throughout the entire observing run (black). Our PSD model therefore enables DINGO to adapt to unseen drifts of the PSDs without retraining.

PSDs by blurring the latent space distribution. Our model  $q(S_n)$  thereby represents a broad distribution over PSDs, which is capable of capturing variations throughout an observing run.

We evaluate our approach on the third LIGO-Virgo-KAGRA (LVK) observing run (O3) by analyzing 390 simulated and 37 real GW events [24]. We train DINGO with our PSD model and consistently find similar performance to DINGO trained with real O3 PSDs. Since our PSD model only uses O2 data and a single PSD from the beginning of O3, this demonstrates that our approach is indeed capable of preparing DINGO for unseen PSD changes.

## II. METHODS

In probabilistic modeling, a latent variable model [25–27] enables efficient sampling of new data given an empirical distribution. We define a latent variable model for the PSDs as

$$q(S_n) = \int q(S_n|z)q_z(z)dz \quad (3)$$

with latent variables  $z$  that we describe below. We use a fixed parametrization for  $q(S_n|z)$  to integrate knowledge about the data generating process into the model. PSD data

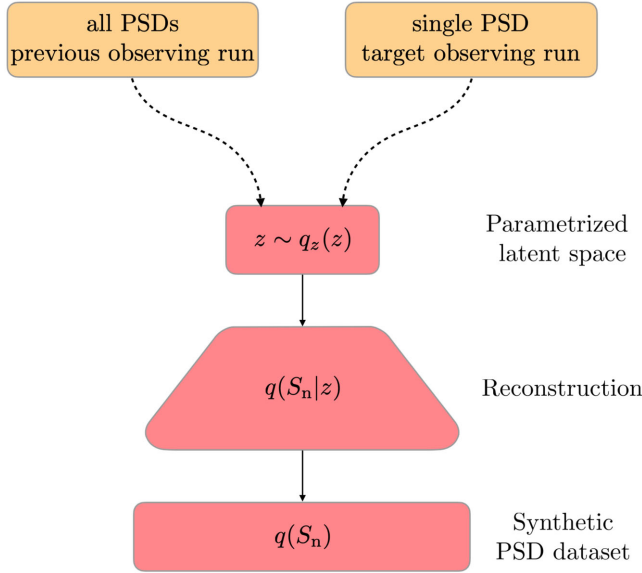


FIG. 2. Overview of the PSD generation pipeline. The PSDs from the previous observing run and a single PSD from the target observing run are used for fitting the latent distribution  $q_z(z)$ , as indicated by the dashed arrows. We can then generate synthetic PSDs by first sampling latent variables in  $z \sim q_z(z)$ . These are then reconstructed back to frequency space via  $q(S_n|z)$ , obtaining a synthetic PSD distribution  $q(S_n)$ .

can then be fitted via the distribution  $q_z(z)$  over the latent variables. This provides an interpretable framework, enabling systematic interventions on the PSD distribution. Synthetic PSDs can then be generated by sampling  $z \sim q_z(z)$  and reconstructing  $z$  back to frequency space via  $q(S_n|z)$ . This pipeline is visualized in Fig. 2; next we explain the individual components in detail.

### A. Parametrization of $q(S_n|z)$

We aim to design a model  $q(S_n)$  with sufficient expressiveness to capture every realistic PSD. On the other hand, the latent space should be low dimensional and maximally disentangled, such that the distribution  $q_z(z)$  partially factorizes. In particular, the latent features  $z$  should model those PSD characteristics that are likely to change over time. This increases the data efficiency and robustness of our model under distribution shifts. These requirements inform our design of  $q(S_n|z)$ . The following computations are carried out in log space.

We leverage domain knowledge to parametrize the latent feature space explicitly. Each PSD is represented by a  $n$ -dimensional vector  $S_n \in \mathbb{R}^n$  for uniformly distributed frequency bins  $f_1, \dots, f_n$  in the range  $[f_{\min}, f_{\max}]$ . According to [28],  $S_n$  can be decomposed into two main components, the smooth broadband noise  $b \in \mathbb{R}^n$  and a sum of high-power spectral lines  $\sum_i s_i \in \mathbb{R}^n$ . Improvements to the detectors are intended to reduce the

impact of the broadband noise, making it likely to change over time. The position and shape of the spectral lines may vary on much shorter timescales. We thus model these components independently. We further assume independent additive Gaussian noise with constant variance  $\sigma^2$  in each frequency bin.

With these assumptions, our model reads

$$q(S_n|z) = \mathcal{N}\left(b + \sum_{i=1}^l s_i, \sigma^2 I_n\right), \quad (4)$$

where  $I_n$  denotes the  $n \times n$  identity matrix. In latent space, the broadband noise is represented by  $k$  values  $y_1, \dots, y_k \in \mathbb{R}_+$  on fixed log-spaced frequency nodes  $x_1, \dots, x_k$ . The logarithmic distribution accounts for larger fluctuations of the broadband noise in lower frequency regions.  $b$  is then obtained from its latent representation via cubic spline interpolation between the nodes.

Each spectral line  $s_i$  is represented by parameters  $f_{0i}, A_i, Q_i$ , denoting the center, height, and width of a truncated Cauchy distribution, respectively, i.e.,

$$s_i[f_{0i}, A_i, Q_i](f) = \frac{w(f, f_{0i})A_i f_{0i}^4}{(f_{0i}f)^2 + (Q_i(f_{0i}^2 - f^2))^2}, \quad (5)$$

with

$$w(f, f_{0i}) = \begin{cases} 1, & \text{if } |f - f_{0i}| \leq \delta, \\ \exp\left(-\frac{|f - f_{0i}|}{\delta}\right), & \text{otherwise,} \end{cases} \quad (6)$$

similar to [28]. To efficiently treat a varying number of spectral lines per PSD, we segment the frequency range into  $l$  equally wide intervals and model a single spectral line within each. Absence of a spectral line is modeled with  $A_i \approx 0$ . We do not find that restricting to one line per interval hinders DINGO performance in practice, provided the number of intervals is sufficiently large; see Fig. 3 (upper panel). In our experiments, we used  $k = 30$  and  $l = 400$  over a frequency range of  $[f_{\min}, f_{\max}] = [20, 2048]$  Hz. If desired, the model could be extended to use overlapping intervals. In this case, each spectral distribution would no longer be independent, but rather conditional on the parameters of the preceding frequency interval.

We found that  $\sigma^2$  does not vary significantly between different PSDs. We therefore use a fixed value for  $\sigma^2$  (estimated from the variance of the observed broadband noise) and do not consider it as part of our latent space. Summarizing, our mapping from the latent space to frequency space reads

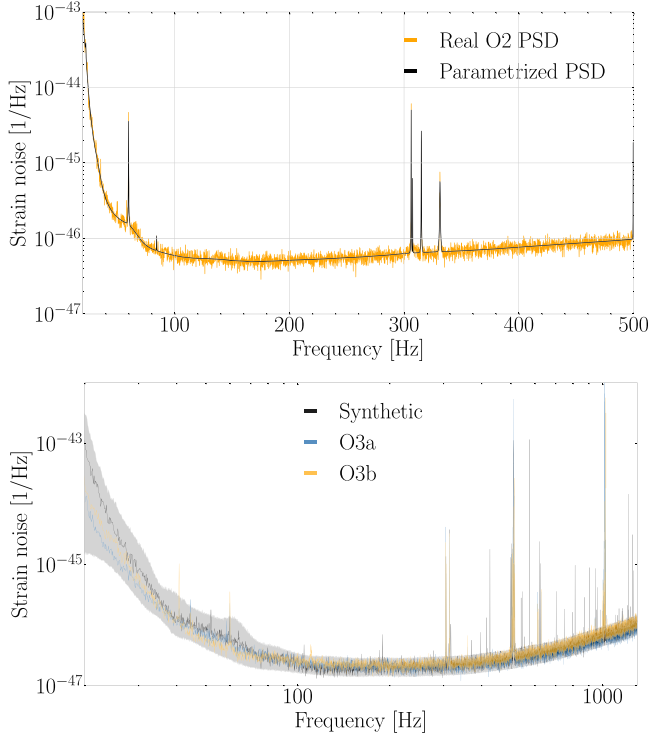


FIG. 3. Upper: comparison of a real O2 PSD and its representation under the latent variable model. Since  $l = 400$  is sufficiently large, we can accurately model spectral lines that are close together (at around 300 Hz). Lower: comparison of a synthetic PSD (dark gray) and two real PSDs from O3a (blue) and O3b (orange). The synthetic PSD was sampled from a model trained on O2 noise, rescaled to O3. In the background (light gray) we include the envelope (1st–99th percentile) of the synthetic broadband distribution. By shifting the broadband noise level (see Sec. II C), we match the overall scale of O3 PSDs. By broadening the distribution, we ensure that PSDs from both O3a and O3b are covered.

$$z = \begin{pmatrix} y_1, \dots, y_k, \\ f_{01}, A_1, Q_1, \\ \vdots \\ f_{0l}, A_l, Q_l \end{pmatrix} \mapsto S_n. \quad (7)$$

The dimensionality of our latent space is  $d = k + 3l$ .

### B. Fitting the latent distribution $q_z(z)$

We fit  $q_z(z)$  based on an empirical distribution of PSDs estimated from detector data. We estimate this collection of PSDs using Welch’s method applied to signal-free data stretches during the previous observing run. We then project these onto the latent space using maximum-likelihood estimation and obtain  $q_z(z)$  as the product of Gaussian kernel density estimates (KDEs) of these projections. This gives a tractable approximation to the true latent distribution. Importantly, having an analytic KDE provides

flexibility to widen  $q_z(z)$  by increasing the KDE bandwidth.

We perform the maximum-likelihood projection onto the latent space in two steps. We first estimate the spline parameters  $y_1, \dots, y_k$  for the broadband noise as the sample mean of the PSD in the vicinity of the corresponding  $x_i$ . Since the broadband noise should not model the spectral lines, we apply a filter before this step.<sup>1</sup> Once the broadband noise is fitted, we subtract it from the PSD to fit the spectral lines. Specifically, we obtain  $(f_{0i}, A_i, Q_i)$  by solving the least-squares problem

$$(f_{0i}, A_i, Q_i) = \arg \min_{(\tilde{f}, \tilde{A}, \tilde{Q})} \sigma^{-2} r r^\top, \quad (8)$$

$$r = r(f, A, Q) = S_n - (b + s_i(f, A, Q)),$$

over each interval  $i$ . Together, these two steps correspond to a maximum-likelihood estimate of  $z$  for (4) with fixed  $\sigma^2$ . On eight CPU cores, it takes less than a minute to obtain the latent maximum-likelihood estimate  $z_{\text{MLE}}$  for each PSD, and this procedure is straightforwardly parallelizable. Indeed, this fast calculation (compared to  $\approx 1$  h for BayesLine [28]) is one reason for choosing this simpler approach when building our PSD model.

Having computed  $z_{\text{MLE}}$  for each PSD from the empirical distribution, we next use KDEs to obtain  $q_z(z)$ . We use a separate KDE for each “independent” noise source and then combine them multiplicatively. For the broadband noise, we partition the frequency range into three intervals according to the most dominant noise source:  $\mathcal{F}_1 = [f_{\min}, 30 \text{ Hz}]$  for seismic noise,  $\mathcal{F}_2 = [30 \text{ Hz}, 100 \text{ Hz}]$  for thermal noise, and  $\mathcal{F}_3 = [100 \text{ Hz}, f_{\max}]$  for shot noise [28]. Since these noise sources should be largely independent of each other, we use individual KDEs for each subset  $\{y_i | x_i \in \mathcal{F}_j\}$ . These subsets do not overlap, so the reconstructed spline naturally agrees on the boundary between intervals  $\mathcal{F}_i$ . Similarly, the spectral lines are modeled independently, so we use a separate KDE for each set of parameters  $(f_{0i}, A_i, Q_i)$ . The latent distribution is then given as the product of the  $3 + l$  individual KDEs.

The independence assumption made above, and the corresponding partial factorization of  $q_z(z)$  into  $3 + l$  independent distributions, serves two purposes. First, it provides broader coverage of the latent space compared to an unfactorized KDE. Second, it decorrelates factors of variation that are independent so that trained networks do not learn to expect spurious correlations that exist in the empirical PSDs. In particular, this allows the PSD model to capture a greater variety of noise curves. Both of these effects ensure that trained networks are generally more

<sup>1</sup>We compute a running median over the neighborhood sets of each  $x_i$ . Every data point that deviates by more than 3 standard deviations from the running median is marked as an outlier and ignored for the fit of the broadband noise.

robust with respect to changes in the PSD so that they can adapt to unseen data.

### C. Interventions in latent space

The latent space is much lower dimensional than the raw PSD space and has features that correspond to the main components comprising a PSD. By intervening in this space (i.e., changing the PSD distribution at the level of the latent space) we can therefore naturally adapt the PSD distribution to changing detector-noise characteristics. We perform two such types of interventions: (1) we shift the broadband noise to account for improved detector sensitivity, and (2) we broaden the distribution to account for uncertainty in the estimated broadband shift and other unmodeled detector changes.

For (1), we rescale the latent distribution over  $y_1, \dots, y_k$  according to an estimated PSD from early in the target run. Given a single PSD  $S_n^{\text{target}}$  from the target run, we shift the distribution over  $y_i$  so that its mean corresponds to the target PSD, i.e.,

$$y_i \mapsto y_i - \mathbb{E}[y_i] + y_i^{\text{target}}. \quad (9)$$

Here,  $y_i^{\text{target}}$  refers to the latent features corresponding to  $S_n^{\text{target}}$ . The resulting distribution corresponds to variations in all latent variables inferred from past data, but with broadband noise level shifted to that of the target observing run. Note that this works even when  $S_n^{\text{target}}$  is not close to the mean of the target run, as long as the difference is small compared to the variance of the estimated distribution. Such shifts are particularly useful to flexibly change the shape of the noise curves, which may be required after major detector upgrades.

The broadening (2) is controlled by the KDE bandwidth parameter, and it improves the robustness of DINGO networks trained with the synthetic noise distribution. A larger bandwidth parameter can compensate for more significant shifts in the PSD distribution, although it may require higher learning capacity.

In Fig. 3, we illustrate how these interventions on the latent space ensure that synthetically generated PSDs match the broadband noise level of O3 PSDs (lower panel), despite the difference in scale between O2 and O3.

## III. RESULTS

We prepare DINGO networks with the settings from [19]. In particular, we use the waveform model IMRPhenomPv2 [29–31] with frequency-domain data in the range [20, 1024] Hz,  $\Delta f = 0.125$  Hz, and the same prior. We train three networks, each with a different noise PSD dataset: (1) the Oracle dataset consists of PSDs estimated from real O3a and O3b detector noise ( $\sim 5000$  PSDs per detector); (2) the Synthetic dataset is sampled with our proposed method, using only O2 data

and a single PSD from the start of O3 (50,000 PSDs per detector); and (3) the Naive baseline dataset consists of real PSDs from the first four days of O3a (100 PSDs per detector). The Oracle dataset encompasses the real PSDs that DINGO encounters at inference time, so this should be an upper bound for the performance of our method. On the other hand, the Naive dataset is based only on data available at the beginning of O3, so to be useful our method should significantly outperform this baseline.

We evaluate the three DINGO networks on simulated and real strain data. To assess performance, we compare against reference posteriors. Since we analyze more than 400 events, generating these using stochastic samplers is not feasible. Instead, we generate reference posteriors by importance weighting the DINGO results produced with the Oracle PSD dataset, which provides verified inference results at low computational cost (DINGO-IS) [23]. To save computational time, we further use phase marginalization when calculating the likelihoods [9,32,33]. We note that with precessing waveforms this is only an approximation, but for IMRPhenomPv2 the error introduced is small. See [23] for an alternative approach to generate the phase in the presence of higher modes. For a comparison of DINGO-IS against BILBY-DYNESTY see Appendix A.

### A. Simulated data

We first evaluate inference results when solely the PSD is varied in the strain data. To this end, we sample gravitational-wave parameters from the prior and inject the signal into identical noise realizations that are scaled with different PSDs. Specifically, we choose 26 real PSDs from O3, evenly spaced in time over the course of one year. The injection strains are then analyzed using each of the three DINGO networks.

Figure 4 shows the mean Jensen-Shannon divergence (JSD) [34] between DINGO and DINGO-IS posteriors for chirp mass ( $\mathcal{M}$ ), luminosity distance ( $d_L$ ), and sky position ( $\alpha, \delta$ ). These parameters have been found to be most significantly impacted by conditioning the network on an incorrect PSD at inference time [19]. (We find a similar trend for training with incomplete PSD information.) Results for all 15 parameters are provided in Appendix B, showing a similar qualitative behavior to the parameters reported here.

At the beginning of the observing run, we observe similar accuracy for all three networks. This is not surprising, as the PSDs are still captured by the Naive dataset at this point. However, as PSDs drift away from their initial distribution, the performance of the Naive baseline decreases substantially, with JSDs increasing by 1–2 orders of magnitude. The effect is particularly striking at the transition from O3a to O3b. This demonstrates that the Naive baseline indeed fails to generalize well to unseen PSDs. The Oracle network on the other hand shows excellent performance throughout the entire duration of

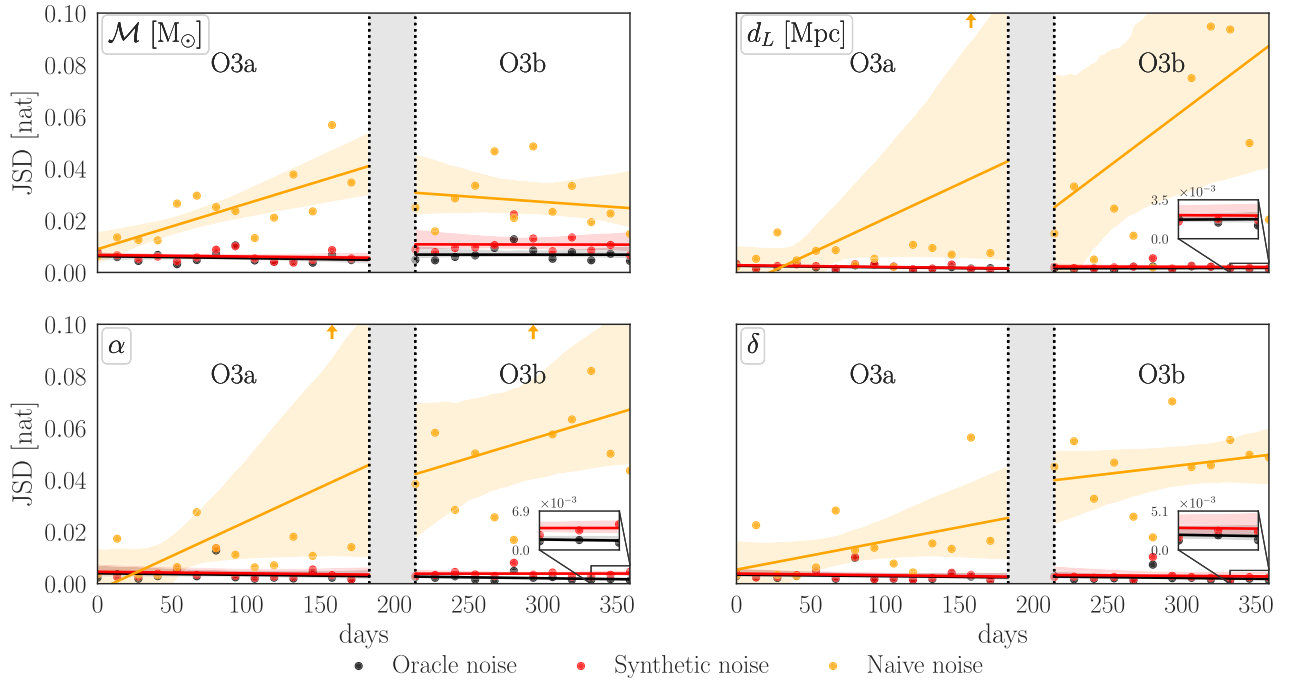


FIG. 4. JSDs between DINGO and reference samples. Injections use noise PSDs estimated at various times throughout O3 (indicated on horizontal axis). Results are averaged over injections with five random sets of source parameters (fixed for all PSDs). Day 0 marks the beginning of O3a. The gap indicates the period when detectors were off-line between O3a and O3b.

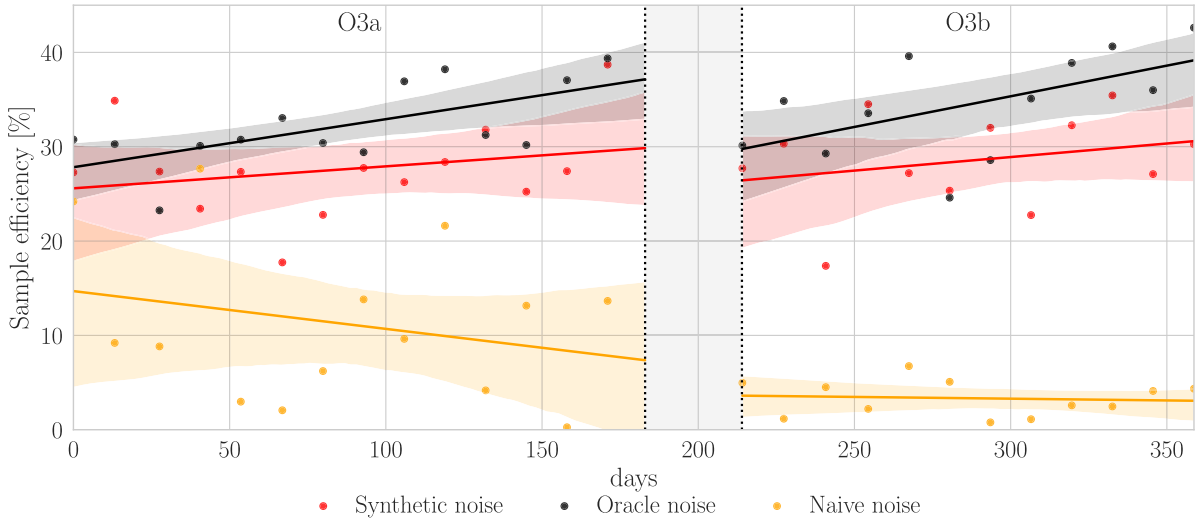


FIG. 5. Importance sampling efficiency  $\epsilon$  for DINGO models. Injections use noise PSDs estimated at various times throughout O3 (indicated on horizontal axis). Results are averaged over injections with five random sets of source parameters (fixed for all PSDs). Day 0 marks the beginning of O3a. The gap indicates the period when detectors were off-line between O3a and O3b.

O3, with JSDs around  $2 \times 10^{-3}$  nat.<sup>2</sup> This is also expected, since the Oracle network has access to PSDs from the entire duration of O3 during training.

Finally, with the Synthetic PSD dataset, DINGO accuracy approaches that of the Oracle network—even for PSDs

<sup>2</sup>For comparison, a maximum JSD of  $2 \times 10^{-3}$  nat has been established as a bound for indistinguishability in [10].

recorded almost a year after the PSD used to recalibrate the Synthetic dataset (enlarged parts in Fig. 4). It far outperforms the Naive baseline, demonstrating that, with our proposed method, DINGO can indeed be trained to generalize to unseen PSDs, with at most a small decrease in accuracy.

In the same experiment, we study the sample efficiency  $\epsilon$  when using the inferred DINGO posterior as a proposal

distribution for importance sampling [23] (Fig. 5). Small values of  $\epsilon$  have been found to flag failure cases of the inference network, e.g., caused by out-of-distribution data, establishing it as another quality measure of the DINGO posteriors. For the Naive baseline,  $\epsilon$  decreases with time and plummets after the transition from O3a to O3b. In contrast, the Oracle and Synthetic networks provide a consistently high sample efficiency of  $\epsilon \approx 30\%$ . As a performance metric,  $\epsilon$  is sensitive to deviations in full parameter space, including high-dimensional correlations. Figure 5 thus confirms the trend observed for the JSDs and demonstrates that the Synthetic PSD dataset can be used to ensure a high efficiency of DINGO-IS throughout an observing run.

### B. Real data

We now perform a large study on real data [24], analyzing all 37 binary black hole events from GWTC-2 and GWTC-3 that are consistent with the training prior. Here, we use four different DINGO models with distance priors [0.1, 2], [0.1, 4], [0.1, 6], and [0.1, 12] Gpc. The mean JSDs over all 15 parameters are reported in Table I. Generally, the scores fluctuate more compared to the highly controlled simulated data, in that the noise realizations vary and can be nonstationary or non-Gaussian, and real signals do not precisely match models. GW190527\_092055, for example, is more difficult for the Oracle network even though the PSD is covered by the training dataset. We see that the trend observed for simulated data translates to real events and all 15 parameters. The decreasing accuracy for the Naive baseline becomes particularly apparent after the transition from O3a to O3b. With our proposed Synthetic noise model, however, the accuracy is maintained throughout the entire observing run and on par with the Oracle performance. This showcases, once again, that our approach is indeed capable of forecasting shifts in the PSD distribution and to enable robust inference.

GW190517\_055101 has a high mean JSD for all three datasets, suggesting that the poor performance is not due to an out-of-distribution PSD. Across all events (except GW190517\_055101) and parameters, we obtain average JSDs of  $1.2 \times 10^{-3}$  nat for the Oracle noise dataset,  $1.4 \times 10^{-3}$  nat with the Synthetic noise dataset, and  $2.7 \times 10^{-3}$  nat with the Naive noise dataset. We thus conclude that our approach serves as a convincing replacement to the empirical PSD distribution when full PSD information is unavailable.

We report the importance sampling efficiency for all analyzed events in Table II. We see that high overall scores are achieved with all PSD datasets, although scores are usually lowest with the Naive noise dataset. Compared to the simulated events of Sec. III A, the downward trend in sampling efficiency for the Naive noise dataset is somewhat less clear. This is likely due to varying noise realizations and source parameters across real events, which introduce

TABLE I. Thirty-seven binary black hole events from GWTC-2 and GWTC-3 analyzed with DINGO trained with three different PSD datasets. We report the JSD (in units of  $10^{-3}$  nat) between DINGO and reference posteriors, averaged over all inferred source parameters.

| Event           | Noise datasets |           |       |
|-----------------|----------------|-----------|-------|
|                 | Oracle         | Synthetic | Naive |
| GW190408_181802 | 2.1            | 4.0       | 3.1   |
| GW190413_052954 | 0.2            | 0.5       | 0.4   |
| GW190413_134308 | 0.9            | 0.6       | 2.0   |
| GW190421_213856 | 0.3            | 0.4       | 0.5   |
| GW190503_185404 | 1.1            | 2.7       | 1.6   |
| GW190513_205428 | 2.1            | 3.7       | 2.9   |
| GW190514_065416 | 0.5            | 0.5       | 0.7   |
| GW190517_055101 | 21.3           | 25.4      | 14.2  |
| GW190519_153544 | 1.2            | 1.4       | 1.2   |
| GW190521_074359 | 1.4            | 2.4       | 2.7   |
| GW190527_092055 | 5.5            | 2.1       | 1.4   |
| GW190602_175927 | 1.9            | 1.9       | 3.3   |
| GW190701_203306 | 1.2            | 1.4       | 1.4   |
| GW190719_215514 | 0.7            | 0.3       | 0.4   |
| GW190727_060333 | 0.3            | 0.6       | 0.6   |
| GW190731_140936 | 0.3            | 0.4       | 1.1   |
| GW190803_022701 | 0.6            | 0.3       | 1.4   |
| GW190805_211137 | 0.3            | 0.4       | 1.1   |
| GW190828_063405 | 0.7            | 2.1       | 2.6   |

| Event           | Noise datasets |           |       |
|-----------------|----------------|-----------|-------|
|                 | Oracle         | Synthetic | Naive |
| GW190909_114149 | 0.4            | 0.5       | 0.9   |
| GW190915_235702 | 0.8            | 0.7       | 1.4   |
| GW190926_050336 | 1.7            | 1.8       | 1.8   |
| O3b ↓           |                |           |       |
| GW191127_050227 | 3.5            | 1.7       | 3.2   |
| GW191204_110529 | 3.1            | 4.4       | 6.1   |
| GW191215_223052 | 0.8            | 1.1       | 2.8   |
| GW191222_033537 | 0.4            | 0.5       | 1.4   |
| GW191230_180458 | 0.3            | 0.4       | 0.7   |
| GW200128_022011 | 0.6            | 1.0       | 2.9   |
| GW200129_065458 | 2.6            | 1.6       | 5.3   |
| GW200208_130117 | 0.5            | 0.6       | 5.5   |
| GW200208_222617 | 2.3            | 2.3       | 3.1   |
| GW200209_085452 | 1.3            | 1.8       | 2.2   |
| GW200216_220804 | 0.4            | 0.9       | 3.5   |
| GW200219_094415 | 0.8            | 1.0       | 4.0   |
| GW200220_124850 | 0.2            | 0.3       | 0.7   |
| GW200224_222234 | 0.7            | 1.4       | 14.5  |
| GW200311_115853 | 1.6            | 2.6       | 10.2  |

additional complications when comparing sampling efficiencies. For visual comparison, we include corner plots of selected events in Appendix B.

TABLE II. Importance sampling efficiency  $\epsilon$  for DINGO networks trained with three different PSD datasets, based on 100,000 samples per analysis. Since the variance of  $\epsilon$  between identical importance sampling runs can be high, we here report median scores over ten runs for each event.

| Event           | Noise datasets |           |       |
|-----------------|----------------|-----------|-------|
|                 | Oracle         | Synthetic | Naive |
| GW190408_181802 | 28.5           | 19.4      | 17.3  |
| GW190413_052954 | 44.6           | 45.0      | 41.9  |
| GW190413_134308 | 41.0           | 49.3      | 43.6  |
| GW190421_213856 | 51.1           | 46.2      | 44.7  |
| GW190503_185404 | 46.9           | 40.4      | 35.4  |
| GW190513_205428 | 28.4           | 28.0      | 16.0  |
| GW190514_065416 | 34.5           | 37.5      | 37.3  |
| GW190517_055101 | 20.5           | 13.6      | 23.0  |
| GW190519_153544 | 41.0           | 41.7      | 25.3  |
| GW190521_074359 | 30.9           | 27.8      | 18.9  |
| GW190527_092055 | 25.8           | 33.6      | 28.6  |
| GW190602_175927 | 41.7           | 46.0      | 38.8  |
| GW190701_203306 | 27.5           | 33.5      | 26.4  |
| GW190719_215514 | 24.3           | 22.9      | 23.4  |
| GW190727_060333 | 42.7           | 48.7      | 31.1  |
| GW190731_140936 | 46.5           | 47.7      | 33.9  |
| GW190803_022701 | 51.8           | 45.5      | 48.0  |
| GW190805_211137 | 50.1           | 51.8      | 39.2  |
| GW190828_063405 | 50.8           | 35.4      | 34.2  |

| Event           | Noise datasets |           |       |
|-----------------|----------------|-----------|-------|
|                 | Oracle         | Synthetic | Naive |
| GW190909_114149 | 32.9           | 17.6      | 22.5  |
| GW190915_235702 | 49.3           | 48.7      | 45.1  |
| GW190926_050336 | 30.8           | 22.1      | 27.7  |
| O3b ↓           |                |           |       |
| GW191127_050227 | 2.8            | 3.9       | 2.5   |
| GW191204_110529 | 19.3           | 20.1      | 17.1  |
| GW191215_223052 | 45.0           | 44.1      | 28.9  |
| GW191222_033537 | 46.6           | 47.6      | 39.7  |
| GW191230_180458 | 41.3           | 40.3      | 40.5  |
| GW200128_022011 | 33.9           | 29.8      | 29.8  |
| GW200129_065458 | 27.4           | 30.9      | 8.2   |
| GW200208_130117 | 48.7           | 47.8      | 41.7  |
| GW200208_222617 | 9.5            | 15.5      | 13.9  |
| GW200209_085452 | 16.7           | 13.1      | 16.6  |
| GW200216_220804 | 33.7           | 32.5      | 30.6  |
| GW200219_094415 | 25.4           | 22.6      | 23.3  |
| GW200220_124850 | 49.9           | 47.7      | 46.6  |
| GW200224_222234 | 49.5           | 36.0      | 9.5   |
| GW200311_115853 | 43.9           | 41.8      | 16.3  |

### C. Discussion

We compared the performance of DINGO inference networks trained with empirical and synthetic PSD distributions. In training, these distributions are represented as a finite set of PSD samples, and the empirical

distributions consist of far fewer samples per detector (Naive, 100 PSDs; Oracle,  $\sim 5000$  PSDs) than the Synthetic distribution (50,000 PSDs). The unbalanced nature of these datasets may impact our results, since more training samples naturally lead to better generalization. However, the effectively unlimited number of synthetic samples available should be viewed as an advantage, and in practice, the number of available samples will always depend on the estimation method. Indeed, for the Naive dataset, which consists of PSDs from the first four days of O3, the data scarcity severely limits the number of available PSDs. Therefore, using unbalanced datasets correctly reflects the practical considerations. Generally, however, our results showing a time-dependent performance decay for the Naive dataset indicate that the distribution shift has a far more significant effect on performance than the size of the dataset.

### IV. CONCLUSIONS

We developed a probabilistic model for detector-noise PSDs to improve training for flow-based GW inference and enable low-latency analyses. The PSD model can be fitted to empirical distributions and then rapidly sampled to obtain synthetic PSDs. By design, the PSD model operates on an interpretable latent space, such that samples can be modified in a physically meaningful way. This allows for data-efficient modeling of distribution shifts, as may occur between LVK observing runs.

In our experiments, we fitted a PSD distribution based on data from the second observing run (O2) and predicted the updated distribution for O3 based on only a single PSD from the beginning of O3. We demonstrated on simulated and real GW events that DINGO models trained with these synthetic O3 PSDs perform accurate inference throughout the entire run ( $\sim 1$  yr). We found comparable performance to DINGO models that have direct access to the entire O3 PSD distribution. This is in stark contrast to DINGO models trained with PSDs from the first few days of O3, for which the accuracy strongly degrades over time.

Our approach crucially hinges on the conditioning of DINGO models on the PSD. Indeed, we do not need to predict specific distribution shifts; instead, it is merely necessary that PSDs encountered at inference time lie within the support of the training distribution. Anticipated distribution shifts are addressed in two ways: first, the broadband noise is shifted to better match the target; second, the PSD distribution is artificially broadened to account for unknown (future) variations. Our synthetic distribution thereby captures unseen distribution shifts, at the cost of being somewhat broader than necessary. While this may generally require increased learning capacity, we have not observed this as a limiting factor in our experiments.

These systematic interventions on the PSD distribution are enabled by the design of our PSD model. We



disentangle independent factors of variations by separating spectral lines from the broadband noise, both of which are represented in a latent space.

Our framework further provides a tractable evidence of PSDs under the estimated distribution, which can be used to quantify distribution shifts. Going forward, this metric could be used to verify that our PSD model continues to work well between future observing runs, e.g., O3 and O4. A low evidence would then flag a distribution shift and indicate that retraining the network may be required. Moreover, for individual events, results can be validated using importance sampling [23].

Techniques for parametrizing and estimating PSDs such as BayesWave [7] and BayesLine [28] have been successfully applied to spectral estimation and accurate posterior inference, and our method indeed draws inspiration from these. However, our goal is to model a distribution of PSDs, whereas these methods are designed to estimate a single PSD. Our work in fact complements these methods: by setting  $\sigma^2 = 0$ , we can model smooth broadband noise, such that our synthetic PSDs closely resemble those of BayesWave (as opposed to noisier Welch PSDs). By training DINGO networks using  $\sigma^2 = 0$  synthetic PSDs, we therefore hope to enable inference with BayesWave PSDs as context.

Our PSD model represents a key component in training flow-based inference networks for low-latency analysis of GW data—enabling complete parameter estimation in real time, with no retraining even for unseen PSD distribution shifts (as expected during an observing run). Once DINGO is extended to binary neutron stars, this will play a crucial role in delivering rapid and accurate multimessenger alerts.

This research has made use of data or software obtained from the Gravitational Wave Open Science Center [24].

The code underlying the proposed method is available as part of the DINGO package at Ref. [35].

## ACKNOWLEDGMENTS

We thank N. Gupte, I. Harry, S. Ossokine, V. Raymond, and R. Smith for useful discussions. This material is based upon work supported by NSF’s LIGO Laboratory which is a major facility fully funded by the National Science Foundation. This research has made use of data or software obtained from the Gravitational Wave Open Science Center (gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research

Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain. The construction and operation of KAGRA are funded by Ministry of Education, Culture, Sports, Science and Technology (MEXT), and Japan Society for the Promotion of Science (JSPS), National Research Foundation (NRF) and Ministry of Science and ICT (MSIT) in Korea, Academia Sinica (AS) and the Ministry of Science and Technology (MoST) in Taiwan. M.D. thanks the Hector Fellow Academy for support. J.H.M. and B.S. are members of the MLCoe, EXC number 2064/1—Project number 390727645 and the Tübingen AI Center funded by the German Ministry for Science and Education (FKZ 01IS18039A). For the implementation of DINGO we use PYTORCH [36], NFLOWs [37], LALSIMULATION [38], and the Adam optimizer [39]. The plots are generated with MATPLOTLIB [40], CHAINCONSUMER [41], and SEABORN [42].

## APPENDIX A: COMPARISON TO OTHER SAMPLERS

In our experiments, we used importance sampling to verify the accuracy of DINGO networks trained with various noise PSD datasets. This approach, called DINGO-IS, has been shown to generate accurate posterior distributions, given a high effective sample size [23]. To validate this claim, and to compare our proposed method with nested sampling, we ran BILBY [11] with the DYNesty [12] sampler for all 37 real events in Sec. III B. The box-and-whisker plot of the JSDs between all one-dimensional marginal distributions is presented in Fig. 7. In most cases, the maximum divergences between DINGO-IS and BILBY are well below the indistinguishability threshold of  $2 \times 10^{-3}$  nat. For events exceeding this threshold, only a small number of parameters are problematic, and the effective sample size is relatively low; see Table II. Overall, we obtain an average maximum JSD of  $1.3 \times 10^{-3}$  nat and an average mean JSD of  $0.4 \times 10^{-3}$  nat between DINGO-IS and BILBY, confirming the effectiveness of DINGO-IS for producing reference posterior distributions. Figure 8, row 3, further compares typical- and worst-case deviations qualitatively. In the typical case, the deviation between DINGO-IS and BILBY is barely visible. While results with DINGO-IS are not perfectly aligned with BILBY’s prediction in the worst case, our proposed method still produces distributions that are very close to those of BILBY.

## APPENDIX B: ADDITIONAL RESULTS

Figure 6 shows the mean (top) and maximum (bottom) JSDs between various DINGO posteriors and

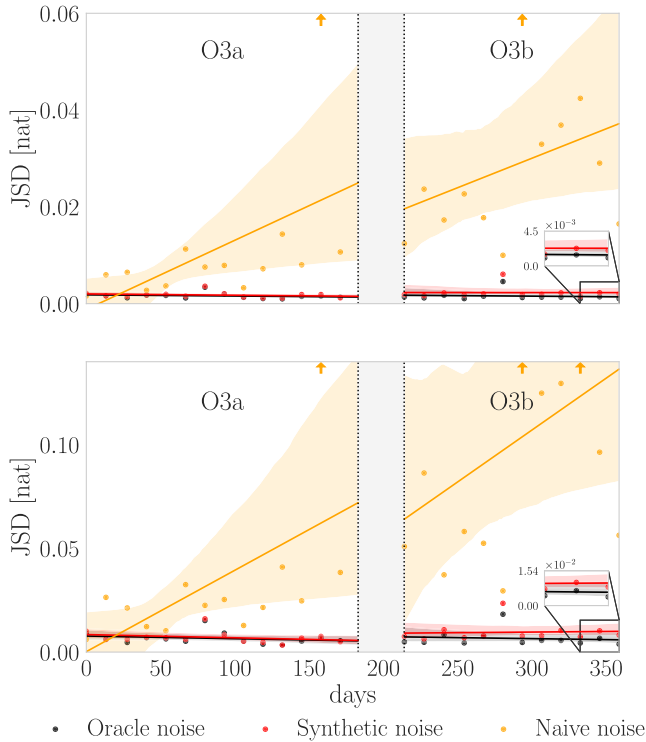


FIG. 6. Average (top) and maximum (bottom) JSDs over all parameters between DINGO samples and reference samples. Injections use noise PSDs estimated at various times throughout O3 (indicated on horizontal axis). Results are averaged over injections with five random sets of source parameters (fixed for all PSDs). Day 0 marks the beginning of O3a. The gap indicates the period when detectors were off-line between O3a and O3b.

reference samples, across all inferred parameters, for the study on simulated data in Sec. III A. We see that the network trained with Synthetic PSDs on average performs similar to that trained with full PSD information (Oracle), with minor deviations for those parameters that are estimated the worst. The average JSD for the Oracle dataset is  $(1.6 \pm 0.7) \times 10^{-3}$  nat, for Synthetic is  $(2.0 \pm 0.9) \times 10^{-3}$  nat, and for Naive is  $(19.1 \pm 18.8) \times 10^{-3}$  nat.

In Fig. 8, we compare posterior marginal distributions of selected O3 events between the reference posterior and our DINGO models. The events in the top row correspond to small and average JSDs between DINGO trained with the Synthetic noise PSD dataset and the reference posterior in Table I. With the Synthetic PSD distributions, we obtain excellent visual agreement with the reference, even improving upon the Oracle PSD distribution in the case of GW190413\_134308. By

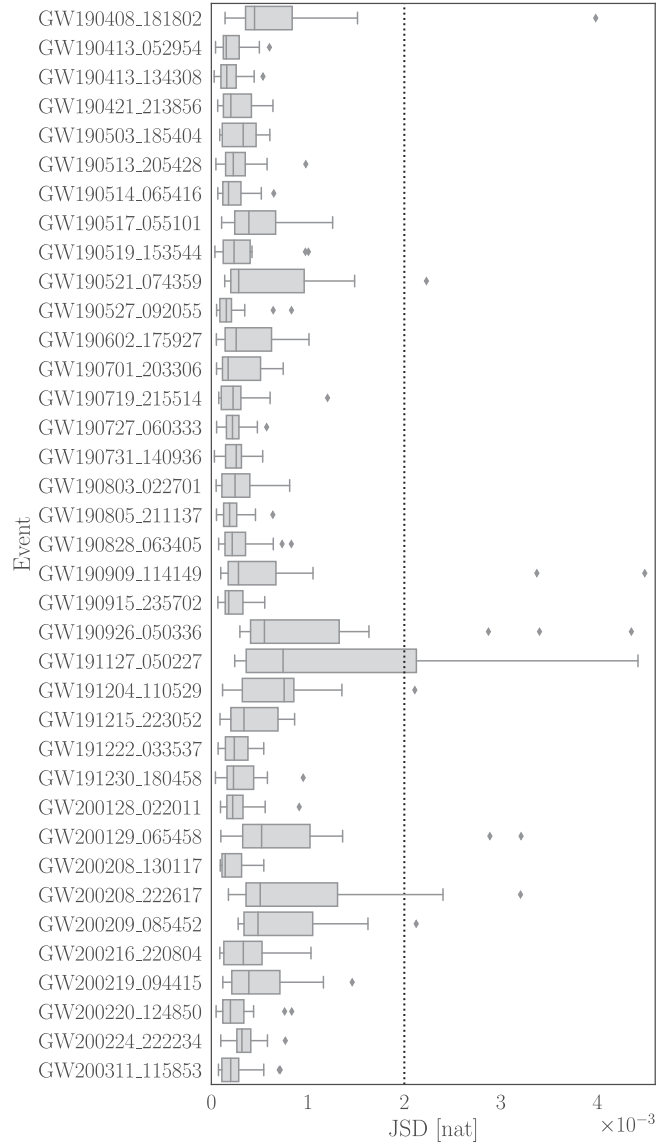


FIG. 7. Comparison of posterior distributions obtained with DINGO-IS and BILBY DYNESTY. We here report the median, quartiles and 1.5 interquartile range whiskers of the JSDs between all one-dimensional marginal distributions. The dotted line marks the indistinguishability threshold for the maximum JSD established in [10].

contrast, the Naive noise model can lead to significant deviations. In the middle row, we selected two events for which DINGO’s predictions with the Synthetic noise dataset deviate the most from the reference distributions. However, we do not believe that these inaccuracies are related to the PSD dataset since they are present even when using Oracle.

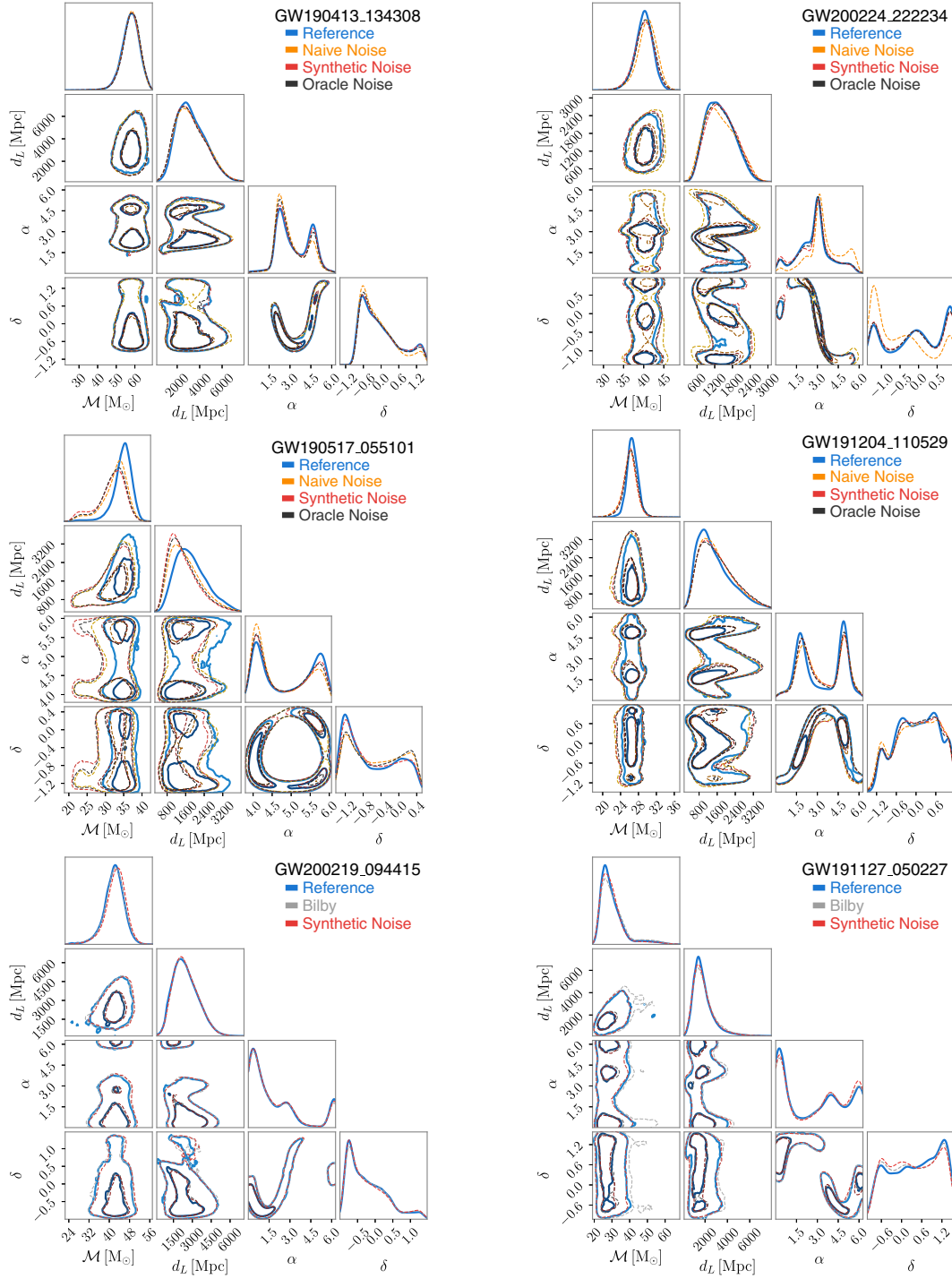


FIG. 8. Marginalized one- and two-dimensional posterior distributions for selected O3 events. Contours represent 50% and 90% credible regions. In the first two rows, we compare DINGO models trained with different PSD datasets to the reference posterior (blue, obtained with importance sampling [23]). Events were chosen to represent small to average deviations between Synthetic and the reference (top row) and worst-case deviations (middle row). In the bottom row, we further compare our reference results to BILBY (gray), showing an event with average deviation (left) and the event with the largest deviation (right).

- [1] J. Aasi *et al.* (LIGO Scientific Collaboration), Advanced LIGO, *Classical Quantum Gravity* **32**, 074001 (2015).
- [2] F. Acernese *et al.* (Virgo Collaboration), Advanced Virgo: A second-generation interferometric gravitational wave detector, *Classical Quantum Gravity* **32**, 024001 (2015).
- [3] K. Somiya (KAGRA Collaboration), Detector configuration of KAGRA: The Japanese cryogenic gravitational-wave detector, *Classical Quantum Gravity* **29**, 124007 (2012).
- [4] 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).
- [5] T. Akutsu *et al.* (KAGRA Collaboration), Overview of KAGRA: Detector design and construction history, *Prog. Theor. Exp. Phys.* **2021**, 05A101 (2021).
- [6] P. Welch, The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms, *IEEE Trans. Audio Electroacoust.* **15**, 70 (1967).
- [7] N. J. Cornish and T. B. Littenberg, BayesWave: Bayesian inference for gravitational wave bursts and instrument glitches, *Classical Quantum Gravity* **32**, 135012 (2015).
- [8] N. J. Cornish, T. B. Littenberg, B. Bécsy, K. Chatziioannou, J. A. Clark, S. Ghonge, and M. Millhouse, BayesWave analysis pipeline in the era of gravitational wave observations, *Phys. Rev. D* **103**, 044006 (2021).
- [9] J. Veitch, V. Raymond, B. Farr, W. Farr, P. Graff, S. Vitale *et al.*, Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library, *Phys. Rev. D* **91**, 042003 (2015).
- [10] G. Ashton *et al.*, BILBY: A user-friendly Bayesian inference library for gravitational-wave astronomy, *Astrophys. J. Suppl. Ser.* **241**, 27 (2019).
- [11] 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).
- [12] J. S. Speagle, DYNESTY: A dynamic nested sampling package for estimating Bayesian posteriors and evidences, *Mon. Not. R. Astron. Soc.* **493**, 3132 (2020).
- [13] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy, *Nat. Phys.* **18**, 112 (2022).
- [14] A. J. K. Chua and M. Vallisneri, Learning Bayesian Posteriors with Neural Networks for Gravitational-Wave Inference, *Phys. Rev. Lett.* **124**, 041102 (2020).
- [15] C. Chatterjee, L. Wen, K. Vinsen, M. Kovalam, and A. Datta, Using deep learning to localize gravitational wave sources, *Phys. Rev. D* **100**, 103025 (2019).
- [16] S. R. Green, C. Simpson, and J. Gair, Gravitational-wave parameter estimation with autoregressive neural network flows, *Phys. Rev. D* **102**, 104057 (2020).
- [17] S. R. Green and J. Gair, Complete parameter inference for GW150914 using deep learning, *Mach. Learn. Sci. Tech.* **2**, 03LT01 (2021).
- [18] 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](https://arxiv.org/abs/2010.12931).
- [19] 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).
- [20] 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 (ICLR 2022)* (2022), [arXiv:2111.13139](https://arxiv.org/abs/2111.13139).
- [21] P. G. Krastev, K. Gill, V. A. Villar, and E. Berger, Detection and parameter estimation of gravitational waves from binary neutron-star mergers in real LIGO data using deep learning, *Phys. Lett. B* **815**, 136161 (2021).
- [22] H. Shen, E. A. Huerta, E. O’Shea, P. Kumar, and Z. Zhao, Statistically-informed deep learning for gravitational wave parameter estimation, *Mach. Learn. Sci. Tech.* **3**, 015007 (2022).
- [23] M. Dax, S. R. Green, J. Gair, M. Pürrer, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, companion Letter, Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference, *Phys. Rev. Lett.* **130**, 171403 (2023).
- [24] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo, *SoftwareX* **13**, 100658 (2021).
- [25] D. J. Bartholomew, M. Knott, and I. Moustaki, *Latent Variable Models and Factor Analysis: A Unified Approach* (John Wiley & Sons, New York, 2011).
- [26] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 2014, Conference Track Proceedings* (2014), [arXiv:1312.6114v10](https://arxiv.org/abs/1312.6114v10).
- [27] D. J. Rezende, S. Mohamed, and D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in *International Conference on Machine Learning* (Curran Associates, Inc., New York, 2014), pp. 1278–1286, [arXiv:1401.4082](https://arxiv.org/abs/1401.4082).
- [28] T. B. Littenberg and N. J. Cornish, Bayesian inference for spectral estimation of gravitational wave detector noise, *Phys. Rev. D* **91**, 084034 (2015).
- [29] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. J. Forteza, and A. Bohé, Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era, *Phys. Rev. D* **93**, 044007 (2016).
- [30] M. Hannam, P. Schmidt, A. Bohé, L. Haegel, S. Husa, F. Ohme, G. Pratten, and M. Pürrer, Simple Model of Complete Precessing Black-Hole-Binary Gravitational Waveforms, *Phys. Rev. Lett.* **113**, 151101 (2014).
- [31] A. Bohé *et al.*, Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors, *Phys. Rev. D* **95**, 044028 (2017).
- [32] J. Veitch and W. Del Pozzo, Analytic marginalisation of phase parameter, <https://dcc.ligo.org/LIGO-T1300326/public> (2013).

- [33] E. Thrane and C. Talbot, An introduction to Bayesian inference in gravitational-wave astronomy: Parameter estimation, model selection, and hierarchical models, *Publ. Astron. Soc. Aust.* **36**, e010 (2019); **37**, e036(E) (2020).
- [34] J. Lin, Divergence measures based on the Shannon entropy, *IEEE Trans. Inf. Theory* **37**, 145 (1991).
- [35] <https://github.com/dingo-gw/dingo>.
- [36] A. Paszke *et al.*, PYTORCH: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (Curran Associates, Inc., Red Hook, NY, 2019), pp. 8024–8035.
- [37] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, NFFLOWS: Normalizing flows in PYTORCH (Zenodo, 2020), [10.5281/zenodo.4296287](https://zenodo.org/record/4296287).
- [38] LIGO Scientific Collaboration, LIGO Algorithm Library—LALSuite, free software (GPL) (2018).
- [39] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [40] J. D. Hunter, MATPLOTLIB: A 2d graphics environment, *Comput. Sci. Eng.* **9**, 90 (2007).
- [41] S. R. Hinton, CHAINCONSUMER, *J. Open Source Software* **1**, 45 (2016).
- [42] M. L. Waskom, SEABORN: Statistical data visualization, *J. Open Source Software* **6**, 3021 (2021).