Normalizing Flows as an Avenue to Studying Overlapping Gravitational Wave Signals

Jurriaan Langendorff,[1,*] Alex Kolmus,[2,†] Justin Janquart,[1,3] and Chris Van Den Broeck[1,3]
[1]Institute for Gravitational and Subatomic Physics (GRASP), Department of Physics, Utrecht University, Princetonplein 1, 3584 CC Utrecht, Netherlands
[2]Institute for Computing and Information Sciences (ICIS), Radboud University Nijmegen, Toernooiveld 212, 6525 EC Nijmegen, Netherlands
[3]Nikhef, Science Park 105, 1098 XG Amsterdam, Netherlands

(Received 28 November 2022; revised 8 March 2023; accepted 5 April 2023; published 26 April 2023)

Because of its speed after training, machine learning is often envisaged as a solution to a manifold of the issues faced in gravitational-wave astronomy. Demonstrations have been given for various applications in gravitational-wave data analysis. In this Letter, we focus on a challenging problem faced by third-generation detectors: parameter inference for overlapping signals. Because of the high detection rate and increased duration of the signals, they will start to overlap, possibly making traditional parameter inference techniques difficult to use. Here, we show a proof-of-concept application of normalizing flows to perform parameter estimation on overlapped binary black hole systems.

DOI: 10.1103/PhysRevLett.130.171402

Introduction.—Over the last few years, the improved sensitivity of the LIGO [1] and Virgo [2] detectors has made the detection of gravitational waves (GWs) originating from compact binary coalescences (CBCs) more and more common, with over 90 detections reported after the third observation run [3]. Soon, the upgrade of the current detectors and the addition of KAGRA [4–8] and LIGO India [9] to the network of ground-based interferometers will lead to even more detections. In addition, the passage from second-generation (2G) to third-generation (3G) detectors [Einstein Telescope (ET) [10,11] and Cosmic Explorer (CE) [12–14]] will lead to an important increase in the number of observed CBCs. These detectors are also projected to have a reduced lower frequency cutoff [15], leading to longer signal durations. Therefore, CBC signals will overlap in 3G detectors [16–20].

Analyzing one of the overlapping signals without accounting for the presence of the other can lead to biases in the recovered posteriors, especially when the merger times of the two events are close [17–21]. These could impact any direct science case for CBCs (e.g., tests of general relativity [22]), but also indirectly related ones such as the hunt for primordial black holes [23–28]. In Ref. [29], the authors demonstrate on two overlapped binary black holes (BBHs) how adapted Bayesian inference can help reduce the biases. In particular, they perform joint parameter estimation, where the two signals are analyzed jointly. While accounting for all the noise characteristics, their analysis also suffers from some instabilities, and further upgrades are needed for it to be entirely reliable. An issue also mentioned in this Letter is the computational time. With hundreds of thousands of CBC mergers expected in the 3G era [17], analyses taking several weeks are not a realistic alternative.

Even if traditional methods can be sped-up [30–33], or quantum computing [34] could potentially be used in the future, the development of frameworks capable of doing complete analyses in short timescales is crucial for the development of 3G detectors. Therefore, in this Letter, we propose the first step in that direction, showing how overlapping BBHs can be analyzed with a normalizing flow (NF) approach [35–37].

Machine learning for overlapping gravitational waves.—The use of machine learning (ML) in GW data analysis has been growing over the last years, having a wide range of applications [38]. A subset of these methods falls under the umbrella of simulation-based inference [39], and is being developed to perform parameter estimation for CBCs [40–47]. References [43–45] use NFs to get posterior distributions for BBH parameters, obtaining results close to those from traditional Bayesian methods. Our approach is somewhat similar to theirs, with some notable differences explained below.

Our approach uses *continuous conditional NFs* [48,49] (CCNFs), a variant of NFs suited for probabilistic modeling and Bayesian inference. Because of the recursive and continuous nature of these models, their memory footprint can be quite small [50], allowing for extensive training on home-grade GPUs while retaining the ability to capture complex distributions.

NFs are a method in ML through which a neural network can learn the mapping from some simple base distribution $p_u(\boldsymbol{u})$ to a more complex final distribution $q(\boldsymbol{\theta})$. This is done through a series of invertible and differentiable transformations, summarized by a function $g(\boldsymbol{\theta})$. However, in our case, the final distribution we seek depends on the GW data to analyze. Therefore, we use *conditional NFs* [51], where

the transformation functions are dependent on the data $\boldsymbol{d}$ (hence, $g = g(\boldsymbol{\theta}, \boldsymbol{d})$). A major difference with [51] is that our base distributions are kept static. Thus our model $g(\boldsymbol{\theta}, \boldsymbol{d})$ is a trainable conditional bijective function transforming a simple 30-D Gaussian into a 30-D complex distribution. The bijectivity allows us to express and sample $q(\boldsymbol{\theta}|\boldsymbol{d})$ in terms of $g(\boldsymbol{\theta}, \boldsymbol{d})$ and $p_u(\boldsymbol{u})$ via

$$q(\boldsymbol{\theta}|\boldsymbol{d}) = |\det[J_{g^{-1}}(\boldsymbol{\theta}, \boldsymbol{d})]| p_u[g^{-1}(\boldsymbol{\theta}, \boldsymbol{d})], \tag{1}$$

where $\det[J_{g^{-1}}(\boldsymbol{\theta}, \boldsymbol{d})]$ is the determinant of the Jacobian $J_{g^{-1}}(\boldsymbol{\theta}, \boldsymbol{d})$ of the transformation. We train the model by minimizing the forward KL divergence, which is equivalent to maximum likelihood estimation [37,52]. As noted by [45], $q(\boldsymbol{\theta}|\boldsymbol{d})$ should cover the actual (Bayesian) posterior $p(\boldsymbol{\theta}|\boldsymbol{d})$, and asymptotically approach it as training progresses due to the mode-covering nature of the forward KL divergence.

A distinctive choice of our method is the continuous nature of the flow, which is linked to the transformation function itself. Neural ordinary differential equations (neural ODEs) [50] are the foundation of continuous NFs; they are not represented by a stack of discrete layers but by a hypernetwork [53]. Hypernetworks can be understood as regular networks where "external" inputs such as a time or depth variable smoothly change the output of the network for identical inputs. They can thus represent multiple transformations. In [50], hypernetworks are used to represent ODEs and are trained by using ODE solvers and clever use of the adjoint sensitivity method. A continuous NF uses neural ODEs as its transformations.

We will now explain the training of a continuous flow. For clarity, we will use h to refer to a continuous transformation and g for a discrete one. If $\boldsymbol{\theta}(t)$ represents the samples from the distribution at a given time t, when going from t_1 to t_2, the continuous NF obeys

$$\frac{d\boldsymbol{\theta}(t)}{dt} = h(t, \boldsymbol{\theta}(t)). \tag{2}$$

The change in likelihood associated with this "step" differs slightly from Eq. (1) due to the continuous nature of the flow:

$$\log[p(\boldsymbol{\theta}(t_1))] = \log[p(\boldsymbol{\theta}(t_0))] - \int_{t_0}^{t_1} \mathrm{Tr}[J_g(\boldsymbol{\theta}(t))]. \tag{3}$$

Assuming a nonstiff ODE the integration can be performed rapidly with state-of-the-art ODE solvers, MALI [54] in our case. In addition, we have to calculate a trace instead of a determinant, speeding up the computation which reduces the complexity, going from $\mathcal{O}(D^3)$ to at most $\mathcal{O}(D^2)$ with D being the dimensionality of posterior space, speeding-up the computation [55,56]. Moreover, using continuous NFs removes the need to use coupling layers between transformations, instead, all parameter dimensions can be dependent on each other throughout the flow. Combining the continuous and conditional flows leads to CCNFs, where the conditional consists of the GW $\boldsymbol{d}$ and the time t.

We also need a better data representation than the raw strain to train and analyze the data. Therefore, we follow a similar approach as in [41–45], using a single value decomposition (SVD) [57] as summary statistics, reducing the dimension and the noise content of the data while retaining at least 99% of the original signal. Each of the 256 generated basis vectors is used as a kernel in 1D convolutions used as an initial layer in a ten-layer residual convolutional neural network (CNN), enabling one to capture the time variance of the signal. Therefore, we do not need to use a Gibbs sampler to estimate the time of the signal as done in [43–45], and can sample over time like any other variable. The CCNF itself is represented by two multilayer perceptrons with three hidden layers of 512 units. Furthermore, we use a different representation for the angles. Instead of directly using their values, we project them onto a sphere for the sky location and onto a circle for the other angles. This makes for a better-posed domain for these angles, and plays on the strong interpolation capacities of the network, making the training step easier.

In the end, our framework combines data representation as a hybrid between SVD and CNN, followed by the CCNF network. A representation of our analysis framework is given in Fig. 1. Our entire framework is relatively small compared to the ones presented in [43], both the residual network and CNF network. Therefore, it can run on lower-end GPUs, but could also be limited in its capacity to model the problem.

Data and setup.—To test our framework, we start with a simplified setup, considering a network made of the two LIGO detectors and the Virgo detector, at design sensitivity [2,58], and with a lower sensitive frequency of 20 Hz. We generate stationary Gaussian noise from their power spectral density (PSD) and inject two precessing BBH mergers using the IMRPhenomPv2 waveform [59]. Our data frames have an 8 s duration and are whitened after

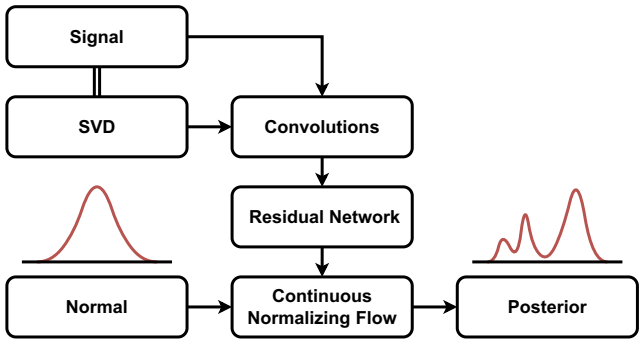

FIG. 1. Representation of our analysis framework. It is made of a preprocessing part where we build a SVD basis to filter the data, followed by a normalizing-flow-based neural network.

the signals are injected. The chirp mass [$\mathcal{M}_c = (M_1 + M_2)^{3/5}/(M_1 M_2)^{1/5}$] and mass ratio ($q = M_1/M_2$) are sampled from uniform distributions, between $10M_\odot$ and $100M_\odot$ and 0.125 and 1, respectively. The individual component masses are constrained between $5M_\odot$ and $100M_\odot$. During the data generation, the luminosity distance is kept fixed. It is then rescaled to result in a network signal-to-noise ratio value taken randomly between 10 and 50 from a beta distribution with a central value of 20. The coalescence time for the two events is set randomly around a time of reference, with $t_c \in [t_{\rm ref} - 0.05, t_{\rm ref} + 0.05]$ s, ensuring that the two BBHs merge in the high bias regime [18]. The other parameters are drawn from their usual domain. Table I gives an overview of the parameters and the function from which they are sampled.

During the training, we continuously generate data by sampling the prior distributions for the events and making a new noise realization for each frame. The training is stopped when convergence is reached and before overfitting occurs. Our model was trained for about 12 days on a single *Nvidia GeForce GTX 1080*.

Results.—To demonstrate the method's reliability, a *P-P* plot for the recovered parameters is shown in Fig. 2. It is constructed by sampling the posteriors of 1000 overlapped events with parameters drawn from the distributions detailed in Table I [60]. Since the cumulative density aligns along the diagonal, our network is reliable. Comparing this to the results given in [43] for single signals, there is a broadening of the shell around the diagonal, showing more variability in signal recovery, meaning our inference is less accurate than for single signals. Possible origins are the degenerate posteriors, increased complexity of the problem, and the reduced size of our network. This increased variability when going from single to joint parameter estimation has also been noted in Bayesian approaches [29].

TABLE I. Summary of the parameters considered and the function used to generate the BBHs.

Parameter	Function
Chirp mass ($\mathcal{M}$)	$\mathcal{U}(10, 100)M_\odot$
Mass ratio (q)	$\mathcal{U}(0.125, 1)$
Component masses ($m_{1,2}$)	Constrained in $[5, 100]M_\odot$
Luminosity distance (D_L)	Rescaled to follow SNR
SNR	$\mathcal{B}(10, 50)$
Coalescence time (t_c)	$\mathcal{U}(t_{\rm ref} - 0.05, t_{\rm ref} + 0.05)$
Spin Amplitudes ($a_{1,2}$)	$\mathcal{U}(0, 1)$
Spin tilt angles ($\theta_{1,2}$)	Uniform in sine
Spin vector azimuthal angle (ϕ_{jl})	$\mathcal{U}(0, 2\pi)$
Spin precession angle (ϕ_{12})	$\mathcal{U}(0, 2\pi)$
Inclination angle (θ_{jn})	Uniform in sine
Wave polarization (ψ)	$\mathcal{U}(0, \pi)$
Phase of coalescence (ϕ)	$\mathcal{U}(0, 2\pi)$
Right ascension (RA)	$\mathcal{U}(0, 2\pi)$
Declination (DEC)	Uniform in cosine

While Bayesian methods have been developed in [29], they are not yet fully stable and take a long time to analyze a BBH system. Therefore, making a statistically significant study comparing the two approaches seems a bit premature at this stage. However, to have some sense of the performances of our network compared to traditional methods, we make 15 injections complying with our network's setup and analyze them with the framework presented in [29]. Using these analyses, we can already identify some trends between the two pipelines. The first is that our ML pipeline typically has broader posteriors than the Bayesian approach. As mentioned in Ref. [29], the classical joint parameter estimation approach can sometimes get overconfident—see Ref. [29] for a discussion on the Bayesian algorithm—where the recovered injected value lies outside of the 90% confidence interval. Our method is not confronted with this bottleneck as the broader posterior encapsulates the injected value. Figure 3 illustrates the two representative situations: one where the Bayesian approach finds the event correctly, and one where we see that our ML approach covers the injected values while it does not for the classical approach. Bias in the posterior, similar to the one noted in Ref. [29], can exist in our method and would not be seen because of the broad posteriors. However, because we are using the forward KL divergence, we expect the posteriors to have some support for the injected values. The origin of the larger posterior, which is not observed in the single parameter estimation machine learning-based methods, is probably due to the increased complexity of the problem combined with the small residual and CNF network sizes. One

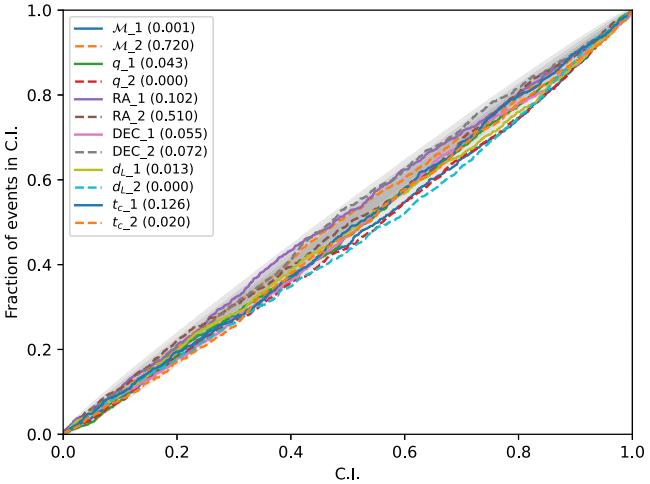

FIG. 2. *P-P* plots for a subset of the recovered parameters for the two events in the data. The parameters shown are representative of all the BBH parameters for the two events. In both cases, the lines align along the diagonal, showing that our method can be trusted. The legend indicates which line corresponds to which parameters. The parameters for event 1 (resp. 2) are noted $P1$ (resp. $P2$), where P are the usual parameter symbols as presented in Table I. The values between the brackets are the KS test statistic.

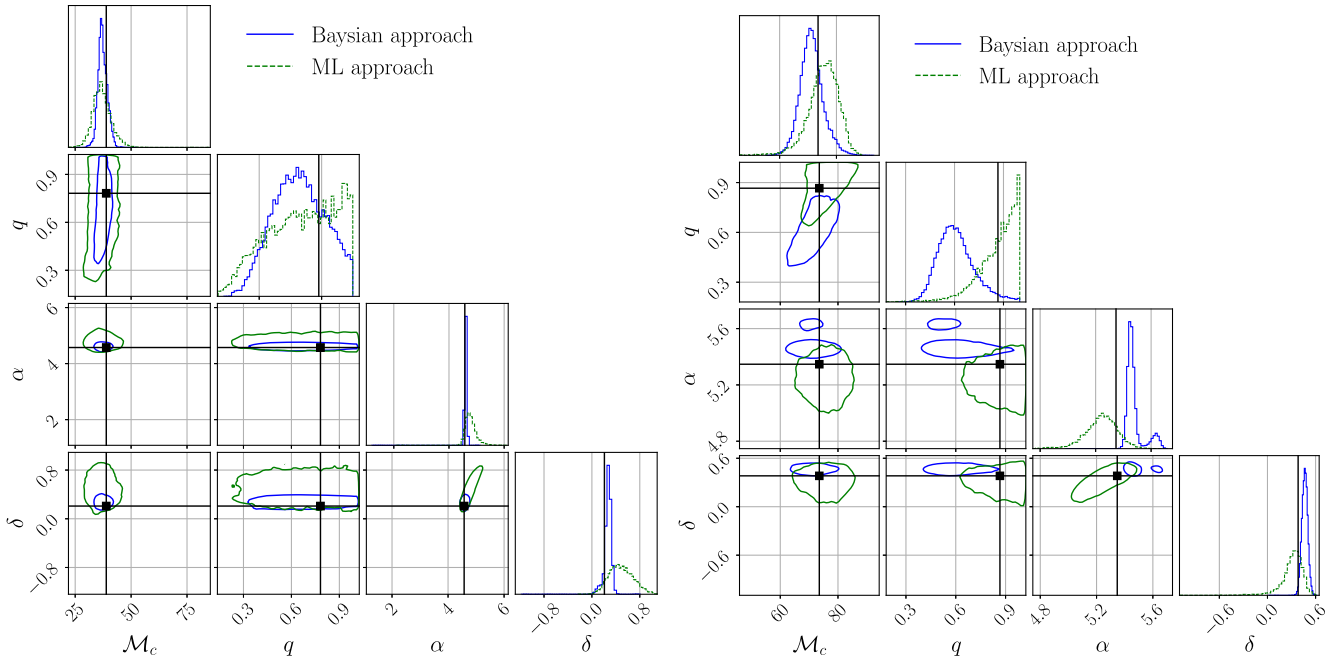

FIG. 3. Comparison between our approach and the one from [29] for two separate events and for the chirp mass, mass ratio, right ascension, and declination. The injected values are given by the black lines. For the left event the true value is encapsulated by the posteriors of both methods, for the right event this is only the case for our method. Our posteriors are generally broader but include the injected value within the 90% confidence interval. This could be corrected by applying importance sampling on the output samples.

possible avenue is applying importance sampling after the normalizing flow [45,61]. However, such methods can be tricky, and additional modifications to our network could be needed.

Finally, an important advantage of our method is its speed. After being trained, it can analyze two overlapping BBH signals in about a second, to compare with $\mathcal{O}(20$ days) reported in [29]. While it is difficult to estimate the time gain for other CBC signals, we can expect the inference time after training not to be significantly larger than for BBHs. Since computational time is a crucial aspect of studies in the 3G era, ML approaches seem to be more suited to study realistic scenarios for these detectors.

Conclusions and perspectives.—In this Letter, we have presented a proof-of-concept machine learning-based method to analyze overlapping BBH signals. We focused on a 2G detector scenario with the two LIGO detectors, and the Virgo detector at design sensitivity, with a lower frequency cutoff of 20 Hz. Our approach is based on continuous normalizing flows.

While also using normalizing flows, as in [41–45], we bring extra modifications that seem to help in the inference task. We represent the data through a mixture of SVD and convolutions, enabling us to sample directly over the events' arrival time, retaining the ability to access the likelihood of a sample. We also move to continuous conditional normalizing flows, reducing the computational cost of the method as we need to solve a trace instead of a determinant when going from one step to the other in the transformation. Finally, we also use a particular representation of the angles, projecting them onto circles (for the phase, the polarization, …) and spheres (for the sky location). We believe that these modifications make our network more flexible, enabling it to deal with overlapping signals even in a reduced form.

With this simplified setup, we have shown that our approach is reliable, with posteriors consistent with the injected values. Our method takes about one week to train on a single GPU. After that, it only takes about a second to analyze two overlapped BBHs. While, in reality, other types of CBC mergers can happen, their inference after training should not be significantly longer than for BBHs. We also compared our machine learning method with classical Bayesian methods for overlapping signals. While our scheme leads to wider posteriors, it can correctly recover the injected values, even when the Bayesian approach gets overconfident and misses the injection. A possibility to correct for the widened posteriors is to use importance sampling.

Our method's combined reliability and speed show that machine learning is a viable approach to analyzing CBC mergers in the 3G era. More interestingly, it would even be possible without needing to account for the development of more powerful computational means and could enable some science-case studies for ET and CE soon. For example, once trained for all possible BBH systems, it could help study the BBH mass function in the 3G era.

Still, one should note that extra improvements are needed before using our method in realistic 3G scenarios.

One would first need to change our setup to the 3G detectors, where a lower frequency cutoff and extreme SNRs could be encountered. In addition, a wider range of objects should be accounted for. One should include higher-order modes and eccentricity as they could play a crucial role in the 3G era. Other modifications could also be implemented. Additionally, we need to account for the change in noise realization from one event to the other. Some of these steps, like changing the detector configuration, should be relatively easy. Others are more complex, as it is hard to perform parameter inference for long-lasting mergers due to the computational burden. So, extra developments in parameter estimation using machine learning would be required to get to the realistic 3G scenario. For overlapping signals, one would also benefit from developments in the classical study of the 3G scenario, such as how to deal with the noise characterization or the types of other events that could come into the data.

In the end, there is still work to be done before machine learning can be used in realistic 3G scenarios. However, we believe that this Letter shows it is an interesting avenue and could be practical on a relatively short timescale.

The authors are thankful to Tomasz Baka, Tom Heskes, and Twan van Laarhoven for discussions on related topics. The authors also thank Maximilian Dax for the careful rereading of the manuscript. J. L., J. J., and C. V. D. B. are supported by the research program of the Netherlands Organisation for Scientific Research (NWO). A. K. is supported by the NWO under the CORTEX project (NWA.1160.18.316). The authors are grateful for computational resources provided by the LIGO Laboratory and supported by the National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459. We are grateful for computational resources provided by Cardiff University, and funded by an STFC grant supporting UK Involvement in the Operation of Advanced LIGO.

*j.w.langendorff@uu.nl
†alex.kolmus@ru.nl

[1] The LIGO Scientific Collaboration, Classical Quantum Gravity **32**, 074001 (2015).
[2] F. Acernese *et al.* (Virgo Collaboration), Classical Quantum Gravity **32**, 024001 (2015).
[3] R. Abbott *et al.* (LIGO Scientific, Virgo, KAGRA Collaborations), arXiv:2111.03606.
[4] K. Somiya (KAGRA Collaboration), Classical Quantum Gravity **29**, 124007 (2012).
[5] Y. Aso, Y. Michimura, K. Somiya, M. Ando, O. Miyakawa, T. Sekiguchi, D. Tatsumi, and H. Yamamoto (KAGRA Collaboration), Phys. Rev. D **88**, 043007 (2013).
[6] T. Akutsu *et al.* (KAGRA Collaboration), Nat. Astron. **3**, 35 (2019).
[7] T. Akutsu, M. Ando, K. Arai *et al.*, arXiv:2005.05574.
[8] T. Akutsu *et al.* (KAGRA Collaboration), Prog. Theor. Exp. Phys. **2021**, 05A101 (2021).
[9] B. Iyer, T. Souradeep, C. Unnikrishnan, S. Dhurandhar, S. Raja, and A. Sengupta, LIGO-India Technical Report, https://dcc.ligo.org/LIGO-M1100296/public (2011).
[10] M. Punturo *et al.*, Classical Quantum Gravity **27**, 194002 (2010).
[11] S. Hild *et al.*, Classical Quantum Gravity **28**, 094013 (2011).
[12] D. Reitze *et al.*, Bull. Am. Astron. Soc. **51**, 035 (2019).
[13] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, K. Ackley, and C. Adams, Classical Quantum Gravity **34**, 044001 (2017).
[14] T. Regimbau, M. Evans, N. Christensen, E. Katsavounidis, B. Sathyaprakash, and S. Vitale, Phys. Rev. Lett. **118**, 151105 (2017).
[15] B. Sathyaprakash *et al.*, Classical Quantum Gravity **29**, 124013 (2012); **30**, 079501(E) (2013).
[16] T. Regimbau and S. A. Hughes, Phys. Rev. D **79**, 062002 (2009).
[17] A. Samajdar, J. Janquart, C. Van Den Broeck, and T. Dietrich, Phys. Rev. D **104**, 044003 (2021).
[18] E. Pizzati, S. Sachdev, A. Gupta, and B. S. Sathyaprakash, Phys. Rev. D **105**, 104016 (2022).
[19] P. Relton and V. Raymond, Phys. Rev. D **104**, 084039 (2021).
[20] Y. Himemoto, A. Nishizawa, and A. Taruya, Phys. Rev. D **104**, 044010 (2021).
[21] A. Antonelli, O. Burke, and J. R. Gair, Mon. Not. R. Astron. Soc. **507**, 5069 (2021).
[22] S. Wu and A. H. Nitz, Phys. Rev. D **107**, 063022 (2023).
[23] S. Sachdev, T. Regimbau, and B. S. Sathyaprakash, Phys. Rev. D **102**, 024051 (2020).
[24] A. Sharma and J. Harms, Phys. Rev. D **102**, 063009 (2020).
[25] S. Biscoveanu, C. Talbot, E. Thrane, and R. Smith, Phys. Rev. Lett. **125**, 241101 (2020).
[26] B. Zhou, L. Reali, E. Berti, M. Çalışkan, C. Creque-Sarbinowski, M. Kamionkowski, and B. S. Sathyaprakash, arXiv:2209.01310.
[27] B. Zhou, L. Reali, E. Berti, M. Çalışkan, C. Creque-Sarbinowski, M. Kamionkowski, and B. S. Sathyaprakash, arXiv:2209.01221.
[28] L. Reali, A. Antonelli, R. Cotesta, S. Borhanian, M. Çalışkan, E. Berti, and B. S. Sathyaprakash, arXiv:2209.13452.
[29] J. Janquart, T. Baka, A. Samajdar, T. Dietrich, and C. Van Den Broeck, arXiv:2211.01304.
[30] B. Zackay, L. Dai, and T. Venumadhav, arXiv:1806.08792.
[31] L. Dai, T. Venumadhav, and B. Zackay, arXiv:1806.08793.
[32] N. Leslie, L. Dai, and G. Pratten, Phys. Rev. D **104**, 123030 (2021).
[33] S. Morisaki, Phys. Rev. D **104**, 044062 (2021).
[34] S. Gao, F. Hayes, S. Croke, C. Messenger, and J. Veitch, Phys. Rev. Res. **4**, 023006 (2022).
[35] D. Jimenez Rezende and S. Mohamed, arXiv:1505.05770.
[36] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, arXiv:1606.04934.
[37] G. Papamakarios, T. Pavlakou, and I. Murray, arXiv:1705.07057.
[38] E. Cuoco *et al.*, Mach. Learn. Sci. Tech. **2**, 011002 (2021).
[39] K. Cranmer, J. Brehmer, and G. Louppe, Proc. Natl. Acad. Sci. U.S.A. **117**, 30055 (2020).

[40] A. Delaunoy, A. Wehenkel, T. Hinderer, S. Nissanke, C. Weniger, A. R. Williamson, and G. Louppe, arXiv:2010.12931.
[41] S. R. Green, C. Simpson, and J. Gair, Phys. Rev. D **102**, 104057 (2020).
[42] S. R. Green and J. Gair, Mach. Learn. Sci. Tech. **2**, 03LT01 (2021).
[43] M. Dax, S. R. Green, J. Gair, M. Deistler, B. Schölkopf, and J. H. Macke, arXiv:2111.13139.
[44] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Phys. Rev. Lett. **127**, 241103 (2021).
[45] M. Dax, S. R. Green, J. Gair, M. Pürrer, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, arXiv:2210.05686.
[46] M. J. Williams, J. Veitch, and C. Messenger, Phys. Rev. D **103**, 103006 (2021).
[47] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, Nat. Phys. **18**, 112 (2022).
[48] I. Kobyzev, S. J. Prince, and M. A. Brubaker, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3964 (2021).
[49] G. Papamakarios, E. Nalisnick, D. Jimenez Rezende, S. Mohamed, and B. Lakshminarayanan, J. Mach. Learn. Res. **22**, 1 (2021).
[50] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Adv. Neural Inf. Process. Syst. **31** (2018).
[51] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling, arXiv:1912.00042.
[52] G. Papamakarios and I. Murray, arXiv:1605.06376.
[53] D. Ha, A. M. Dai, and Q. V. Le, in *International Conference on Learning Representations, Toulon* (2017).
[54] J. Zhuang, N. C. Dvornek, sekhar tatikonda, and J. s Duncan, in *International Conference on Learning Representations, Vienna* (2021).
[55] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, arXiv:1810.01367.
[56] T. M. Nguyen, A. Garg, R. G. Baraniuk, and A. Anandkumar, arXiv:1912.03978.
[57] N. Halko, P.-G. Martinsson, and J. A. Tropp, SIAM Rev. **53**, 217 (2011).
[58] L. Barsotti, P. Fritschel, M. Evans, and S. Gras, Advanced LIGO anticipated sensitivity curves, https://dcc.ligo.org/LIGO-T1800044/public (2021).
[59] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. J. Forteza, and A. Bohé, Phys. Rev. D **93**, 044007 (2016).
[60] We refer the reader to Fig. 1in Ref. [29] for an illustration of overlapping BBH signals.
[61] A. Kolmus, G. Baltus, J. Janquart, T. van Laarhoven, S. Caudill, and T. Heskes, Phys. Rev. D **106**, 023032 (2022).