Amortized Bayesian inference of gravitational waves with normalizing flows

Maximilian Dax
MPI for Intelligent Systems,
72076 Tübingen, Germany
maximilian.dax@tuebingen.mpg.de

Stephen R. Green
MPI for Gravitational Physics,
14476 Potsdam, Germany
stephen.green@aei.mpg.de

Jonathan Gair
MPI for Gravitational Physics,
14476 Potsdam, Germany

Jakob H. Macke
MPI for Intelligent Systems,
72076 Tübingen, Germany

Alessandra Buonanno
MPI for Gravitational Physics,
14476 Potsdam, Germany

Bernhard Schölkopf
MPI for Intelligent Systems,
72076 Tübingen, Germany

Abstract

Gravitational waves (GWs) detected by the LIGO and Virgo observatories encode descriptions of their astrophysical progenitors. To characterize these systems, physical GW signal models are inverted using Bayesian inference coupled with stochastic samplers—a task that can take days for a typical binary black hole. Several recent efforts have attempted to speed this up by using normalizing flows to estimate the posterior distribution conditioned on the observed data. In this study, we further develop these techniques to achieve results nearly indistinguishable from standard samplers when evaluated on real GW data, with inference times of one minute per event. This is enabled by (i) incorporating detector nonstationarity from event to event by conditioning on a summary of the noise characteristics, (ii) using an embedding network adapted to GW signals to compress data, and (iii) adopting a new inference algorithm that makes use of underlying physical equivariances.

1 Introduction

Since 2015, the LIGO and Virgo gravitational-wave observatories [1, 2] have together detected signals from over 50 mergers of black holes and/or neutron stars [3–5]. Given a detection, the data are compared against theoretical predictions using Bayesian inference to characterize the system. In this study we focus on inference of binary black hole (BBH) mergers, as these are most common and have shorter observable signals than mergers involving neutron stars.

Signal models take the form of time-series waveform predictions $h(\theta)$ as a function of system parameters θ (including the black-hole masses and spins, the orientation of the binary, and its localization in space and time). These are made on the basis of Einstein's theory of general relativity. To this is added detector noise n, assumed to be stationary and Gaussian to a good approximation [although, importantly, the noise power spectral density (PSD) S_n can vary from event to event]. This gives the data generative process (or likelihood) $p(d|\theta, S_\mathrm{n})$, where $d = h(\theta) + n$. Combined with a prior, Bayes' theorem gives the posterior distribution over parameters,

$$p(\theta|d, S_\mathrm{n}) = \frac{p(d|\theta, S_\mathrm{n})p(\theta)}{p(d|S_\mathrm{n})}, \tag{1}$$

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

where the normalization factor $p(d|S_\mathrm{n})$ is the Bayesian evidence.

Standard codes for gravitational-wave inference [6, 7] use stochastic methods such as Markov chain Monte Carlo (MCMC) to draw samples from the posterior. Typically, $\sim 10^4$ posterior samples are desired, which requires millions of likelihood evaluations (waveform simulations). Even using highly optimized models, this can take days per event. With an ever-increasing event rate due to improvements in detector sensitivity, the computational cost of analyzing all events is becoming very significant.

Many studies have attempted to address the challenges of GW inference using machine learning [8–15]. These approaches all give *fast* inference, but the challenge remains to produce *accurate* and *complete* results competitive with conventional algorithms. The present extended abstract (a workshop version of [16]) describes an approach that finally moves beyond proof-of-concept by producing results nearly indistinguishable from conventional algorithms. We build upon a previous study [12] by some of us using neural posterior estimation (NPE) [17] with conditional normalizing flows to estimate the GW posterior. However, we now achieve full amortization of training costs across observations by accounting for detector noise nonstationarity from event to event. We also introduce an embedding network to treat the high-dimensional GW data sets, and we develop a new algorithm to incorporate model equivariances into our framework. We demonstrate our approach (called "DINGO") by analyzing eight GW events from the first Gravitational-Wave Transient Catalog (GWTC-1) [3], achieving excellent quantitative agreement with standard codes. Inference on each event with a fully-trained network takes approximately one minute.

2 Method

DINGO trains a flexible conditional density estimator $q(\theta|d, S_\mathrm{n})$ to approximate $p(\theta|d, S_\mathrm{n})$. We use a normalizing flow [18–20], to define $q(\theta|d, S_\mathrm{n})$ via a mapping $f_{d,S_\mathrm{n}} : u \mapsto \theta$ from a standard-normal "base" distribution $\mathcal{N}(0, 1)^D(u)$,

$$q(\theta|d, S_\mathrm{n}) = \mathcal{N}(0, 1)^D \left(f^{-1}_{d,S_\mathrm{n}}(\theta) \right) \left| \det J_{f^{-1}_{d,S_\mathrm{n}}} \right| , \tag{2}$$

where $D = 15$ is the dimension of the parameter space. This is simply the change-of-variables rule for probability distributions. f_{d,S_n} is chosen to be invertible with simple Jacobian determinant, so $q(\theta|d, S_\mathrm{n})$ can be rapidly evaluated and sampled from. Our basic approach is based on that of [12], and we now highlight the main improvements; for further details see [16].

2.1 Noise conditioning

Noise in GW detectors is to a good approximation stationary and Gaussian. However, it does vary to some degree from event to event, and this must be taken into account for inference at the desired accuracy. The noise PSD is typically estimated from signal-free detector data in the vicinity of an event, and it enters into conventional analyses via the likelihood. For NPE, we use the PSD to whiten the strain data before passing it to the flow. However, as shown in [16], this is not sufficient to fully account for PSD drifts, since information is lost if the PSD is discarded after whitening.[1] We resolve this problem by providing the PSD as additional context information to the flow.

Past studies applying similar approaches to GWs [8–15] used a fixed PSD for training. In this study, we augment our training data with a collection of PSDs estimated during LIGO/Virgo observing runs ($\sim 10^3$ PSDs per detector). These are drawn randomly during training, and are used to generate noise realizations that are added to simulated signals. At inference time, the estimated PSD at the time of the event is provided as context along with the signal data.

2.2 Embedding network

We represent data in the frequency domain since this is the natural representation for stationary Gaussian noise PSDs. For 8 s waveforms (suitable for BBHs), and frequencies between 20 and

[1] For instance, when working with whitened data, scaling the PSDs in all detectors with an overall factor α^2 has an effect equivalent to scaling the luminosity distance d_L by α. If no information about the PSD is provided to the inference network this degeneracy impedes correct inference of d_L.

1024 Hz, we have 24,096 (real) input dimensions per detector. This must be heavily compressed using an embedding network before conditioning the flow. We first apply a linear layer to project the data to 400 components per detector. To provide an inductive bias to extract signal information, we initialize this with the principal components [obtained via a singular value decomposition (SVD)] of clean waveforms from our training set. Implementing the SVD compression with a learnable layer (instead of a fixed projection as done in previous works) allows the inference network to learn a more useful representation while retaining the inductive bias provided by the SVD initialization. Following this, a fully-connected residual network [21] compresses to 128 features.

The flow itself is a composition of 30 rational-quadratic spline coupling flows [22], each of which is made up of 5 two-layer residual blocks, and is conditioned on the output of the embedding network. The embedding network and flow are trained alongside each other.

2.3 Group equivariant neural posterior estimation

One of the main performance impediments encountered in our experiments had to do with estimating the time of binary coalescence t_I measured in each detector I. The relative time of arrival of the signal in each detector is related through triangulation to the sky position of the source, and we also infer the overall time of coalescence at geocenter, so the prior includes training data with varying t_I. In frequency domain, time translations correspond to local phase shifts, which, although well understood, can be challenging for neural networks to learn based on simulations alone. Indeed, this occupied much of the network capacity in [12].

If we had precise knowledge of t_I then we could manually time shift the data to a fixed coalescence time to simplify the task of the network. However, $t_I = t_I(\theta)$ is a function of the parameters θ, which are *a priori* unknown at inference time. Our new approach—called *group equivariant* neural posterior estimation (GNPE) [23]—resolves this problem. GNPE is a general method that enables to self-consistently apply θ-dependent transformations to observational data d despite unknown θ. As discussed in detail in [23], this can be used to integrate exact equivariances of a forward model by construction, and also allows for approximate equivariances.

In the context of GW parameter inference GNPE works as follows. For each θ-dependent variable t_I we define a proxy

$$\hat{t}_I = t_I(\theta) + \epsilon, \quad \epsilon \sim \kappa(\epsilon) \tag{3}$$

as a perturbed version of t_I. We choose a narrow, uniform kernel $\kappa = U[-1\text{ ms}, 1\text{ ms}]$ for the perturbation ϵ, therefore the proxy $\hat{t}_I$ is a good approximation of t_I. With that definition, we aim to sample from the extended posterior[2]

$$\left(\theta, \hat{t}\right) \sim p\left(\theta, \hat{t}|d\right), \tag{4}$$

where $\hat{t}$ is a short hand notation collecting the proxies $\hat{t}_I$ for all detectors I. Samples from (4) can be easily turned into samples $\theta \sim p(\theta|d)$ from the desired posterior by simply marginalizing over $\hat{t}$ (i.e., dropping the corresponding axes). As becomes apparent below, it is easier to learn a representation of the extended posterior $p(\theta, \hat{t}|d)$ than to directly learn $p(\theta|d)$, since the inference network for the extended posterior can be trained on GW data with (almost) no time shifts.

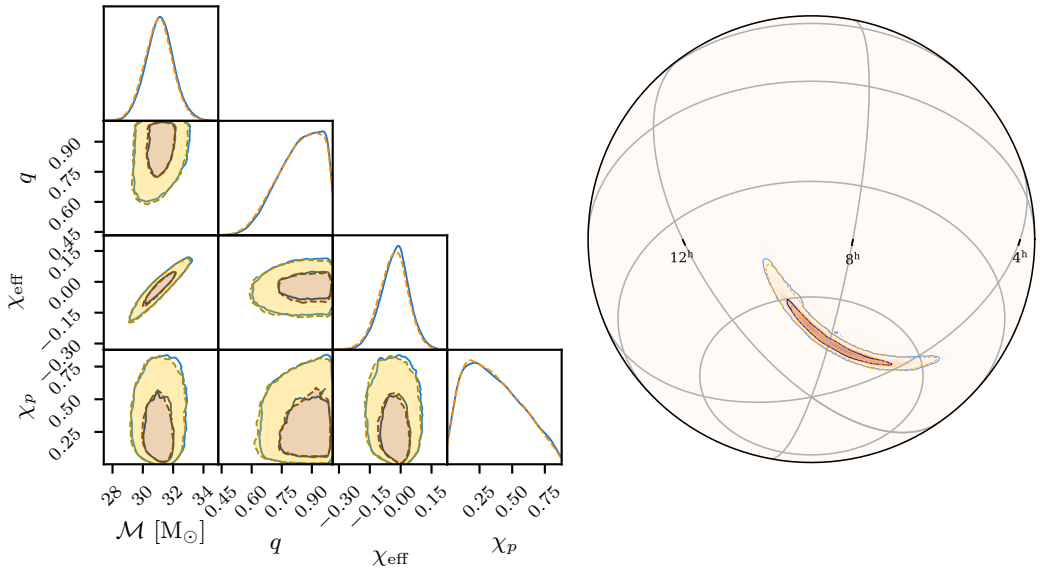
With GNPE, we infer the extended posterior (4) with Gibbs sampling [24, 25]. Specifically, we iteratively sample θ and $\hat{t}$ conditioned on the respective other parameter as well as on d,

$$\hat{t} \sim p\left(\hat{t}|d, \theta\right) \qquad \Longleftrightarrow \qquad \hat{t} = t(\theta) + \epsilon, \quad \epsilon \sim \kappa(\epsilon), \tag{5}$$

$$\theta \sim p\left(\theta|d, \hat{t}\right) \qquad \Longleftrightarrow \qquad \theta \sim q\left(\theta|d_{-\hat{t}}, \hat{t}\right). \tag{6}$$

Here the left-hand side describes the sampling operation and the right-hand side how it is performed in practice. While sampling $\hat{t}$ for given θ in (5) is trivial with its definition (3), we employ a normalizing flow q in (6) to estimate $p\left(\theta|d, \hat{t}\right)$ and sample θ. Importantly, due to the conditioning on $\hat{t}$ (and the invertibility of time shifts), we can time shift the strain data by $-\hat{t}$ before providing it to the flow, which we denote with $d_{-\hat{t}}$. The normalizing flow is thus trained with simulations with almost no time shift, since $t_I - \hat{t}_I$ is restricted to the range $[-1\text{ ms}, 1\text{ ms}]$ of the kernel κ. We found that this re-alignment of the data d is crucial for the accuracy of the flow.

[2]For readability, we leave the conditioning of the posterior on the PSD S_n implicit in this subsection.



(a) (b)

Figure 1: Comparison of GW150914 posterior distributions produced by LALINFERENCE MCMC (blue) and DINGO (orange). Contours represent 50% and 90% credible regions. Panel (a) shows the chirp mass $\mathcal{M}$, mass ratio q, and two effective spin parameters $\chi_{\rm eff}$, χ_p; (b) shows the sky position.

Iterative application of (5) and (6) results in a Markov chain that asymptotically converges to the extended posterior (4). In practice, we speed up convergence by constructing N Markov chains to obtain N samples in parallel. We initialize each of these by sampling from a flow $q(t_I|d)$ trained with standard NPE to infer initial t_I estimates. From each of the N Markov chains we extract only the final sample. As shown in [23], this significantly improves the convergence behaviour, such that only ~ 30 GNPE iterations are required.

Training

The GNPE inference network is trained by minimizing the loss

$$L = \mathbb{E}_{p(\theta)}\mathbb{E}_{p(S_{\rm n})}\mathbb{E}_{p(d|\theta,S_{\rm n})}\mathbb{E}_{\kappa(\epsilon)}\left[-\log q\left(\theta|d_{-(t(\theta)+\epsilon)}, S_{\rm n}, t(\theta)+\epsilon\right)\right]. \tag{7}$$

Estimating (7) requires sampling $\theta^{(i)} \sim p(\theta)$ and $S_{\rm n}^{(i)} \sim p(S_{\rm n})$, and then simulating data $d^{(i)} \sim p(d|\theta^{(i)}, S_{\rm n}^{(i)})$. We choose a prior suitable for BBH systems, with uniform component masses $m_1, m_2 \in [10, 80]$ M$_\odot$, luminosity distance $d_L \in [100, 6000]$ Mpc, and standard uninformative priors for sky position, orientation, and spins. We train separate networks for detector noise levels in the first (O1) and second (O2) observing runs of LIGO and Virgo, with PSD samples estimated empirically from noise data [26]. As in [12], training data are generated from a fixed set of spin-precessing frequency-domain waveforms, described by the IMRPhenomPv2 [27–29] model, but with extrinsic parameters and noise realizations drawn randomly during training. We train for 450 epochs using the Adam optimizer [30], and reserve 2% of the training data for validation. With training sets of 5×10^6 waveforms, there is no indication of overfitting. Training takes between 16 and 18 days on a single NVIDIA Tesla V100. We use `PyTorch` [31] and `nflows` [32] for the implementation of our neural networks. The plots are generated with `ChainConsumer` [33] and `ligo.skymap` [34]. All software used in this project is freely available under MIT or BSD license.

3 Results

We evaluate DINGO on all GWTC-1 events with component masses greater than 10 M$_\odot$ (our prior bound). For all events, we use the data from the two LIGO detectors for the analysis; for GW170814 (the first 3-detector event) we additionally use data from the Virgo detector. We compare

Table 1: Deviation between DINGO and LALINFERENCE posteriors, quantified by the Jensen-Shannon divergence (JSD) between 1D marginals. We report the mean JSD across all parameters, and the maximal JSD, including the corresponding parameter. All JSDs are reported in units of 10^{-3} nat.

event	mean JSD	max JSD
GW150914	0.6	1.4 (δ)
GW151012	0.7	2.7 (m_1)
GW170104	1.0	6.4 (m_1)
GW170729	1.3	6.3 (d_L)

event	mean JSD	max JSD
GW170809	0.9	5.5 (δ)
GW170814	1.0	2.5 (α)
GW170818	1.3	3.8 (α)
GW170823	0.4	0.9 (d_L)

against MCMC posteriors produced using the standard LIGO/Virgo parameter estimation code LALINFERENCE [6]. For DINGO, generation of 50,000 sample points with 30 GNPE iterations takes roughly 1 minute. A qualitative comparison for the first GW detection is displayed in Fig. 1, showing excellent overlap.

For quantitative comparisons, we compute the Jenson-Shannon divergence (JSD) [35] between 1D marginalized posteriors, a divergence which lies between 0 and $\ln(2) \approx 0.69$ nat. Across all events and parameters,[3] we find a mean JSD of 0.0009 nat, which is only slightly higher than the variation (0.0007 nat) found between LALINFERENCE runs with identical settings but different random seeds [37]. Moreover, for GW samplers, a maximum JSD across parameters of 0.002 nat is regarded as indistinguishable [37]; our results approach this, with two events below for all parameters, and the others with one to three parameters above; see Tab. 1. For comparison, variations in the PSD and the choice of waveform model [37] both impact the JSD at a much higher level of ≈ 0.02 nat.

4 Conclusions

In this work, we achieved unprecedented accuracy for rapid GW parameter inference using normalizing flows. We analyzed eight GWTC-1 events, and showed excellent agreement with standard codes, with inference times reduced by over three orders of magnitude. This improvement in performance compared to past studies was achieved by conditioning on the noise PSD, introducing a powerful embedding network, and using the novel GNPE algorithm to simplify the learning task by aligning the signal waveforms in each detector. The latter is a general approach to treating equivariances, which we hope will be useful in other inference applications as well.

The present study is limited to analyzing BBH systems, with relatively simple waveform models, and with stationary Gaussian noise. To extend to longer signals for binary neutron stars and more complex signals incorporating more physics [38] will require somewhat larger networks and improved data representation or compression. DINGO should scale better than conventional algorithms to expensive waveforms since generation of training data can be fully paralellelized. Moreover, as a likelihood-free method, it does not impose any fundamental restrictions on the noise model and should ultimately reduce systematic errors of current analyses (although in this study we used stationary Gaussian noise to compare against conventional samplers). We plan to address these limitations in future studies.

Deep learning is now able to analyze the vast majority of LIGO/Virgo events at comparable accuracy to standard algorithms. Through future extensions we expect that DINGO could become one of the leading approaches to GW inference.

References

[1] J. Aasi et al. Advanced LIGO. *Class. Quant. Grav.*, 32:074001, 2015. doi: 10.1088/0264-9381/32/7/074001.

[2] F. Acernese et al. Advanced Virgo: a second-generation interferometric gravitational wave detector. *Class. Quant. Grav.*, 32(2):024001, 2015. doi: 10.1088/0264-9381/32/2/024001.

[3] Parameters consist of detector-frame component masses (m_1, m_2), time of coalescence at geocenter t_c, reference phase ϕ_c, sky position (α, δ), luminosity distance d_L, inclination angle θ_{JN}, spin magnitudes (a_1, a_2), spin angles $(\theta_1, \theta_2, \phi_{12}, \phi_{JL})$ [36], and polarization angle ψ.

[3] B. P. Abbott et al. GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs. *Phys. Rev. X*, 9(3):031040, 2019. doi: 10.1103/PhysRevX.9.031040.

[4] R. Abbott et al. GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run. *Phys. Rev. X*, 11:021053, 2021. doi: 10.1103/PhysRevX.11.021053.

[5] R. Abbott et al. GWTC-2.1: Deep Extended Catalog of Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run. 8 2021.

[6] J. Veitch et al. Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library. *Phys. Rev.*, D91(4):042003, 2015. doi: 10.1103/PhysRevD.91.042003.

[7] Gregory Ashton et al. BILBY: A user-friendly Bayesian inference library for gravitational-wave astronomy. *Astrophys. J. Suppl.*, 241(2):27, 2019. doi: 10.3847/1538-4365/ab06fc.

[8] Hunter Gabbard, Chris Messenger, Ik Siong Heng, Francesco Tonolini, and Roderick Murray-Smith. Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy, 2019.

[9] Alvin J. K. Chua and Michele Vallisneri. Learning Bayesian posteriors with neural networks for gravitational-wave inference. *Phys. Rev. Lett.*, 124(4):041102, 2020. doi: 10.1103/PhysRevLett.124.041102.

[10] Chayan Chatterjee, Linqing Wen, Kevin Vinsen, Manoj Kovalam, and Amitava Datta. Using Deep Learning to Localize Gravitational Wave Sources. *Phys. Rev. D*, 100(10):103025, 2019. doi: 10.1103/PhysRevD.100.103025.

[11] Stephen R. Green, Christine Simpson, and Jonathan Gair. Gravitational-wave parameter estimation with autoregressive neural network flows. *Phys. Rev. D*, 102(10):104057, 2020. doi: 10.1103/PhysRevD.102.104057.

[12] Stephen R. Green and Jonathan Gair. Complete parameter inference for GW150914 using deep learning. *Mach. Learn. Sci. Tech.*, 2(3):03LT01, 2021. doi: 10.1088/2632-2153/abfaed.

[13] Arnaud Delaunoy, Antoine Wehenkel, Tanja Hinderer, Samaya Nissanke, Christoph Weniger, Andrew R. Williamson, and Gilles Louppe. Lightning-Fast Gravitational Wave Parameter Inference through Neural Amortization. 10 2020.

[14] Plamen G. Krastev, Kiranjyot Gill, V. Ashley Villar, and Edo 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. doi: 10.1016/j.physletb.2021.136161.

[15] Hongyu Shen, E. A. Huerta, Eamonn O'Shea, Prayush Kumar, and Zhizhen Zhao. Statistically-informed deep learning for gravitational wave parameter estimation. 3 2021.

[16] Maximilian Dax, Stephen R Green, Jonathan Gair, Jakob H Macke, Alessandra Buonanno, and Bernhard Schölkopf. Real-time gravitational-wave science with neural posterior estimation. *arXiv preprint arXiv:2106.12594*, 2021.

[17] George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation, 2016.

[18] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

[19] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[20] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[22] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems*, pages 7509–7520, 2019.

[23] Maximilian Dax, Stephen R Green, Jonathan Gair, Michael Deistler, Bernhard Schölkopf, and Jakob H Macke. Group equivariant neural posterior estimation. 2021.

[24] Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic processes and their applications*, 49(2): 207–216, 1994.

[25] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[26] Rich Abbott et al. Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo. *SoftwareX*, 13:100658, 2021. doi: 10.1016/j.softx.2021.100658.

[27] Mark Hannam, Patricia Schmidt, Alejandro Bohé, Leïla Haegel, Sascha Husa, Frank Ohme, Geraint Pratten, and Michael Pürrer. Simple Model of Complete Precessing Black-Hole-Binary Gravitational Waveforms. *Phys. Rev. Lett.*, 113(15):151101, 2014. doi: 10.1103/PhysRevLett.113.151101.

[28] Sebastian Khan, Sascha Husa, Mark Hannam, Frank Ohme, Michael Pürrer, Xisco Jimńez Forteza, and Alejandro Bohé. Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era. *Phys. Rev.*, D93(4):044007, 2016. doi: 10.1103/PhysRevD.93.044007.

[29] Alejandro Bohé, Mark Hannam, Sascha Husa, Frank Ohme, Michael Pürrer, and Patricia Schmidt. PhenomPv2 – technical notes for the LAL implementation. *LIGO Technical Document, LIGO-T1500602-v4*, 2016. URL `https://dcc.ligo.org/LIGO-T1500602/public`.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[32] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. nflows: normalizing flows in PyTorch, November 2020. URL `https://doi.org/10.5281/zenodo.4296287`.

[33] S. R. Hinton. ChainConsumer. *The Journal of Open Source Software*, 1:00045, August 2016. doi: 10.21105/joss.00045.

[34] Leo Singer. ligo.skymap. `https://lscsoft.docs.ligo.org/ligo.skymap/`, 2020.

[35] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

[36] Benjamin Farr, Evan Ochsner, Will M. Farr, and Richard O'Shaughnessy. A more effective coordinate system for parameter estimation of precessing compact binaries from gravitational waves. *Phys. Rev. D*, 90(2):024018, 2014. doi: 10.1103/PhysRevD.90.024018.

[37] 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. Roy. Astron. Soc.*, 499(3):3295–3319, 2020. doi: 10.1093/mnras/staa2850.

[38] Serguei Ossokine et al. Multipolar Effective-One-Body Waveforms for Precessing Binary Black Holes: Construction and Validation. *Phys. Rev. D*, 102(4):044055, 2020. doi: 10.1103/PhysRevD.102.044055.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 4.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
   [No] We plan to release a production-level python package for DINGO in the very near future, including scripts to reproduce our results. The current research code has substantial dependencies on local infrastructure.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?
   [Yes] The most important details are given in section 2, for full information we refer to the corresponding paper [16].
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
   [No] The computational cost of training prohibits multiple identical runs.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See end of section 2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] End of section 2.
   (b) Did you mention the license of the assets? [Yes] See end of section 2.
   (c) Did you include any new assets either in the supplemental material or as a URL?
   [No] All code and models will be included in the DINGO package, see question 3(a).
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]