Machine Learning: Science and Technology

LETTER • **OPEN ACCESS**

Complete parameter inference for GW150914 using deep learning

To cite this article: Stephen R Green and Jonathan Gair 2021 *Mach. Learn.: Sci. Technol.* **2** 03LT01

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 03LT01 https://doi.org/10.1088/2632-2153/abfaed

MACHINE LEARNING Science and Technology

OPEN ACCESS

RECEIVED
26 February 2021

REVISED
8 April 2021

ACCEPTED FOR PUBLICATION
22 April 2021

PUBLISHED
16 June 2021

LETTER

Complete parameter inference for GW150914 using deep learning

Stephen R Green* and **Jonathan Gair**

Max Planck Institute for Gravitational Physics (Albert Einstein Institute), Am Mühlenberg 1, 14476 Potsdam, Germany

* Author to whom any correspondence should be addressed.

E-mail: stephen.green@aei.mpg.de and jonathan.gair@aei.mpg.de

Keywords: gravitational waves, black holes, parameter estimation, deep learning

Abstract

The LIGO and Virgo gravitational-wave observatories have detected many exciting events over the past 5 years. To infer the system parameters, iterative sampling algorithms such as MCMC are typically used with Bayes' theorem to obtain posterior samples—by repeatedly generating waveforms and comparing to measured strain data. However, as the rate of detections grows with detector sensitivity, this poses a growing computational challenge. To confront this challenge, as well as that of fast multimessenger alerts, in this study we apply deep learning to learn non-iterative surrogate models for the Bayesian posterior. We train a neural-network conditional density estimator to model posterior probability distributions over the full 15-dimensional space of binary black hole system parameters, given detector strain data from multiple detectors. We use the method of normalizing flows—specifically, a *neural spline* flow—which allows for rapid sampling and density estimation. Training the network is likelihood-free, requiring samples from the data generative process, but no likelihood evaluations. Through training, the network learns a *global* set of posteriors: it can generate thousands of independent posterior samples per second for any strain data consistent with the training distribution. We demonstrate our method by performing inference on GW150914, and obtain results in close agreement with standard techniques.

1. Introduction

Since the first detection in September 2015 [1], the LIGO/Virgo Collaboration has published observations of gravitational waves from 50 compact binary coalescences [2–6], primarily binary black hole mergers, but also two binary neutron star mergers. In addition, the LIGO/Virgo Collaboration has publicly released around two dozen additional triggers [7] of events of interest, the details of which have so far not been published. These observations have had a transformative impact on our understanding of compact objects in the Universe, facilitated by inferring the parameters of the system (masses, spins, etc) using accurate physical models of the emitted gravitational waves.

This inference is extremely computationally expensive. LIGO/Virgo currently employ Markov Chain Monte Carlo and nested-sampling algorithms to obtain samples from the Bayesian posterior distribution over the parameters [8, 9]. These algorithms are iterative, requiring many waveform simulations for each independent posterior sample. Although fast waveform models have been developed, run times for single detections typically take days for binary black holes and weeks for binary neutron stars [2, 10]. In addition, runs are usually performed using several waveform models to probe any systematic effects, and using the physically most complete (and thus computationally most costly) waveforms available. These long run times will become increasingly problematic as instrument sensitivity improves and event rates reach one per day or higher [11]. For events with multimessenger counterparts (such as binary neutron stars) it is especially important to have fast sky localization to direct follow-up electromagnetic observations.

There is an urgent need for new approaches that can generate scientific inferences much more rapidly than existing pipelines [8, 9]. Deep learning is a promising direction to increase the speed of gravitational-wave inference by several orders of magnitude, which has received increasing focus in recent years [12–14]. The goal of these approaches is to build a non-iterative inverse model for the system parameters given the detector data, which, once trained, can rapidly generate posterior samples without having to perform waveform simulations. In other words, the idea is to train a neural-network conditional density estimator $q(\theta|s)$ to approximate the Bayesian posterior distribution $p(\theta|s)$ of parameter values θ given detector strain data s.

Neural networks typically have millions of parameters, which are optimized stochastically during training to minimize an appropriate loss function. With a 'likelihood-free' training algorithm, it is never necessary to draw samples from the posterior or evaluate a likelihood, rather the procedure is generative and just requires an ability to simulate data sets to construct the training set. Consequently, the training time is comparable to the sampling time using a standard method. There have been several previous studies on this topic, but these either simplified the description of the posterior, e.g. by using a Gaussian mixture approximation [15], or simplified the input, e.g. using a reduced space of parameters and a single detector [16].

In a previous paper [17], we used a neural-network architecture called a conditional variational autoencoder (CVAE) [18, 19] combined with normalizing flows [20–23] to learn the posterior distribution over all parameters of an aligned-spin quasi-circular merger observed with a single gravitational-wave detector. With a single detector we could not recover the full set of waveform parameters, and all data sets analyzed were artificially generated with advanced LIGO design-sensitivity noise [24].

In this study, we develop an approach that can, for the first time, be used to fully analyze real data from the LIGO/Virgo interferometers. We describe a neural-network architecture, based on normalizing flows alone, that can generate posteriors over the full $D = 15$ dimensional parameter space of quasi-circular binary inspirals, using input data from multiple gravitational-wave detectors. We apply this network to analyze observed interferometer data surrounding the first gravitational-wave detection, GW150914. We show that we obtain an accurate Bayesian posterior distribution over the system parameters, in the sense that it is in close agreement with results of conventional methods. This is the first demonstration that deep-learning methods can be used in a realistic setting to produce fast-and-accurate scientific inference on real gravitational-wave interferometer data. We thereby establish a new benchmark in fast-and-accurate gravitational-wave inference, and also describe methods that could be applied to other inference problems in experimental physics.

2. Neural network model

Our aim is to train a neural-network conditional density estimator $q(\theta|s)$ to approximate the gravitational-wave posterior $p(\theta|s)$. To this end, $q(\theta|s)$ must have sufficient flexibility to capture the detailed shape of the true posterior over parameters θ, as well as the dependence on the complicated strain data s. We use the method of *normalizing flows*.

A normalizing flow f is an invertible mapping on a sample space with simple Jacobian determinant [20]. For a conditional distribution, the flow must depend on s, so we denote it f_s. The idea is to train the flow so that it maps a simple 'base' distribution $\pi(u)$ into the far more complex $q(\theta|s)$. We define the conditional distribution in terms of the flow by

$$q(\theta|s) = \pi(f_s^{-1}(\theta)) \left|\det J_{f_s}^{-1}\right|, \tag{1}$$

which is based on the change of variables rule for probability distributions. $\pi(u)$ should be chosen such that it can be easily sampled and its density evaluated; we will always take it to be standard multivariate normal of the same dimension D as the sample space.

By the properties of a normalizing flow, $q(\theta|s)$ inherits the nice properties of $\pi(u)$. Indeed, to draw a sample, one first samples $u \sim \pi(u)$, and then sets $\theta = f_s(u)$; it follows that $\theta \sim q(\theta|s)$. To evaluate the conditional density, one uses (1); the right hand side may be evaluated by the defining properties that f_s is invertible and has simple Jacobian determinant.

Normalizing flows are under active development in computer science, and are usually represented by neural networks. Neural networks are very flexible function approximators, so they can give rise to complex conditional densities. Our previous study [17] used a masked autoregressive flow [22] with affine transformations. In the present study, we use a much more powerful *neural spline* flow [25]. We use the original neural spline flow implementation [29], illustrated in figure 1. We now give a brief summary.

The flow is a composition of 'coupling transforms' $c_s(u)$, each of which transform elementwise half of the parameters (say, $u_{d+1:D}$) conditional upon the other half ($u_{1:d}$) as well as s [31], i.e.:

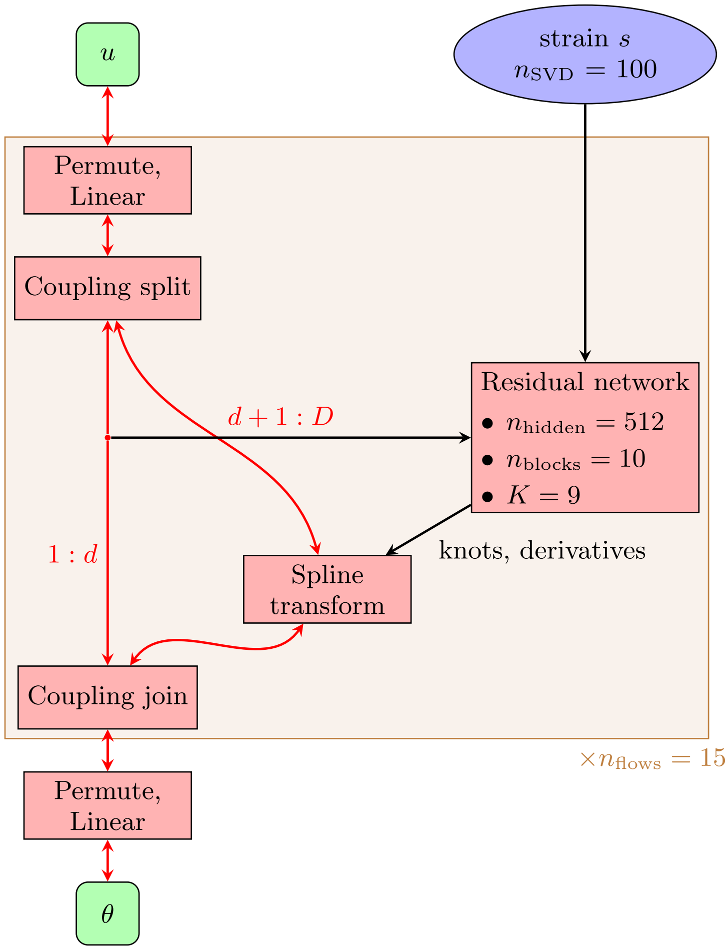

Figure 1. Overall structure of the normalizing flow [25] from $u \mapsto \theta$, with optimal hyperparameter choices indicated. Red connections are invertible. The residual network is made up of n_{blocks} residual blocks, each with two fully-connected hidden layers of n_{hidden} units. Prior to each linear transformation [26], we inserted batch-normalization layers to speed training [27] and exponential linear units for nonlinearity [28]. Each block is also conditioned on s.

$$c_{s,i}(u) = \begin{cases} u_i & \text{if } i \leqslant d, \\ c_i(u_i; u_{1:d}, s) & \text{if } i > d. \end{cases} \tag{2}$$

If c_i is invertible and differentiable with respect to u_i, then it follows immediately that the coupling transform is a normalizing flow. By composing n_{flows} such transforms, and permuting the indices of u in between, a very flexible flow is obtained.

The neural spline coupling transform [25] takes each c_i to be a monotonically-increasing piecewise function, defined by a set of knots $\{(u_i^{(k)}, c_i^{(k)})\}_{k=0}^K$ and positive-valued derivatives $\{\delta_i^{(k)}\}_{k=0}^K$, between which are interpolated rational-quadratic (RQ) functions. The knots and derivatives are output from a residual neural network [32], which takes as input $u_{1:d}$ and s; details are given in figure 1. The RQ spline is differentiable and has analytic inverse, so it satisfies the properties of a coupling transform.

3. Training

The conditional density estimator $q(\theta|s)$ must be trained to approximate as closely as possible the gravitational-wave posterior $p(\theta|s)$. We do this by tuning the neural-network parameters to minimize a loss function, the expected value (over s) of the cross-entropy between the true and model distributions,

$$\begin{aligned} L &= -\int \mathrm{d}s\, p(s) \int \mathrm{d}\theta\, p(\theta|s) \log q(\theta|s) \\ &= -\int \mathrm{d}\theta\, p(\theta) \int \mathrm{d}s\, p(s|\theta) \log q(\theta|s). \end{aligned} \tag{3}$$

On the second line we used Bayes' theorem to express L in a form that involves an integral over the likelihood rather than the posterior; this is a key simplification which means posterior samples are

Table 1. Parameters characterizing quasicircular black hole binaries.

Parameter	Description	Prior	Extrinsic
(m_1, m_2)	Component masses	$(10\,\mathrm{M}_\odot, 80\,\mathrm{M}_\odot)$, $m_1 \geqslant m_2$	No
ϕ_c	Reference phase	$[0, 2\pi]$	No
$t_{c,\mathrm{geocent}}$	Time of coalescence	$(-0.1\,\mathrm{s}, 0.1\,\mathrm{s})$	Yes
d_L	Luminosity distance	(100 Mpc, 1000 Mpc)	Yes
(a_1, a_2)	Dimensionless spin magnitudes	$[0, 0.88]$	No
$(\theta_1, \theta_2, \phi_{12}, \phi_{JL})$	Spin angles	standard [30]	No
θ_{JN}	Inclination relative to line-of-sight	$[0, \pi]$, uniform in sine	No
ψ	Polarization angle	$[0, \pi]$	Yes
(α, δ)	Sky position	Uniform over sky	Yes

not needed for training. We evaluate the integral (3) on a minibatch of training data with a Monte Carlo approximation,

$$L \approx -\frac{1}{N}\sum_{i=1}^{N} \log q(\theta^{(i)}|s^{(i)}), \tag{4}$$

where $\theta^{(i)} \sim p(\theta)$, $s^{(i)} \sim p(s|\theta^{(i)})$, and N is the number of samples in the minibatch. We then use backpropagation (the chain rule) to compute the gradient with respect to network parameters, and minimize L stochastically on minibatches using the Adam optimizer [33].

To obtain a training pair $(\theta^{(i)}, s^{(i)})$, we draw $\theta^{(i)}$ from the prior, we generate a waveform $h(\theta^{(i)})$, and we add a noise realization to obtain $s^{(i)}$. Waveform generation is too costly to perform in real time during training, so we adopt a hybrid approach where we perform the expensive calculations in advance. Before training, we sample 'intrinsic' parameters (see table 1) and we generate the associated plus and cross waveform polarizations $h^{(i)}_{+,\times}$. At each training epoch, we sample new 'extrinsic' parameters for every $h^{(i)}_{+,\times}$. Applying the extrinsic parameters (which is cheap) we obtain waveforms in the detectors. Finally, we also add noise realizations in real time during training. By drawing extrinsic parameters and noise realizations during training, we effectively increase the size of the training set. We used 10^6 sets of intrinsic parameters, which was sufficient to avoid overfitting.

3.1. Prior

We perform inference over the full $D = 15$ dimensional set of precessing quasi-circular binary black hole parameters. These parameters are listed in table 1. As indicated in the table, five of these parameters are considered extrinsic, so they are sampled during training; the remaining intrinsic parameters are sampled in advance of training.

Our choice of prior ranges is based on astrophysical as well as computational considerations. For example, for the component masses, the prior range covers most of the binary black hole coalescences detected to date [2, 6]. The lower bound of $10\,\mathrm{M}_\odot$ arises because the in-band inspiral time scales as $T \propto \mathcal{M}^{-5/3}$, where $\mathcal{M} = (m_1 m_2)^{3/5}/(m_1 + m_2)^{1/5}$ is the chirp mass, so lighter binaries would require much longer data segments, which is more costly. Likewise, the time of coalescence prior is determined by the accuracy of detection pipelines. Although a distance prior uniform in the comoving source frame [9] would be most physical, we adopted a uniform prior over d_L and an upper bound of 1000 Mpc to more uniformly cover the parameter space and improve training. We applied the physical prior at inference time by reweighting samples. We also rescaled all parameters to have zero mean and unit variance.

3.2. Strain data

For likelihood-free training, we require simulated $s^{(i)}$ that arise from the data generative process, $s^{(i)} \sim p(s|\theta^{(i)})$. We assume stationary Gaussian noise, so the gravitational-wave likelihood function is known explicitly, but the method applies even when this is not the case—e.g. with non-Gaussian noise—as long as one can simulate data.

We generate training waveforms using the `IMRPhenomPv2` frequency-domain precessing model [34–36]. We take a frequency range of [20, 1024] Hz, and a waveform duration of 8 s. This frequency range is based on the sensitivity curve of the LIGO and Virgo instruments, and the 8 s waveform duration is adequate for binary black hole mergers consistent with our prior. We then whiten $h^{(i)}_{+,\times}$ using the noise power spectral density (PSD) estimated from 1024 s of detector data prior to GW150914 (separately for each detector), as is standard in LIGO/Virgo analyses [8]. Following [15], we compress the whitened waveforms to a

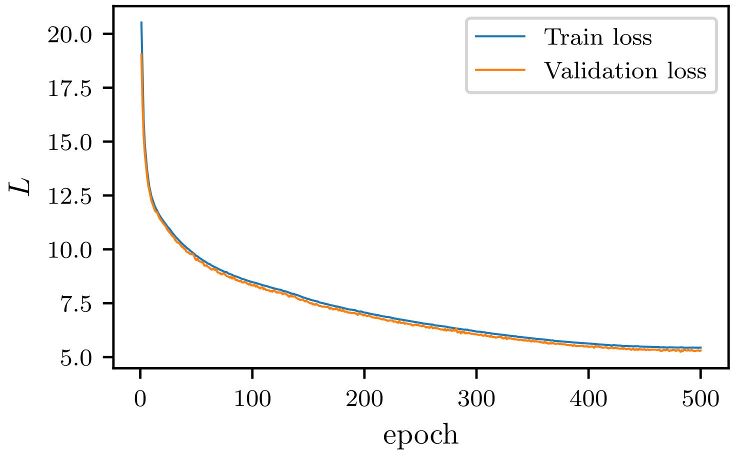

Figure 2. Training and validation loss for each epoch. The closeness of these curves indicates that there is very little overfitting. (Note that the validation loss is in fact slightly lower than the training loss; this is because the batch normalization layers have different behavior during training and testing.)

reduced-order representation; we use a singular value decomposition (SVD), and keep the first $n_{\mathrm{SVD}} = 100$ components in each detector.

During training, we sample extrinsic parameters and generate detector signals. This requires a trivial rescaling to apply d_L, and linear combinations of $h^{(i)}_{+,\times}$ to project onto the antenna patterns for the two LIGO detectors. To apply time delays in the reduced-basis (RB) representation, we follow of [37, 38] and pre-prepare a grid of time-translation matrix operators that act on vectors of RB coefficients, using cubic interpolation for intermediate times. Since the transformation to RB space is a rotation, we add white noise directly to the RB coefficients of the whitened waveforms to obtain $s^{(i)}$. Finally, the noisy strain data is standardized to have zero mean and unit variance in each component.

4. Results

We trained for 500 epochs with a batch size of 512. The initial learning rate for the Adam optimizer was 0.0002, and we annealed to zero using cosine annealing [39]. We performed a search over network hyperparameters, and listed those with best performance (measured by final validation loss) in figure 1. We reserved 10% of the training set for validation, and found no evidence of overfitting (see figure 2). With an NVIDIA Quadro P4000 GPU, training took ≈6 days.

To perform inference on GW150914, we took 8 s of detector data containing the signal and expressed it in the RB representation. We then drew samples from the base space, and applied the normalizing flow conditioned on the strain data to obtain samples from $q(\theta|s)$. This produced samples at a rate of 5000 per second. We benchmarked these against samples produced by `bilby` [9, 40] with the `dynesty` sampler [41].

Our main result is presented in figure 3, which compares the neural network and `bilby` posteriors. Both distributions are in very close agreement, with minor differences in θ_{JN} and the sky position, where the neural network gives more support to secondary modes. With more training or a larger network, we expect even better convergence.

Although our demonstration has focused on GW150914, the network was trained to generate any posterior consistent with the training distribution. In figure 4 we show a P–P plot constructed from 100 artificial injections with parameters drawn from the prior, and noise realizations consistent with the training PSD. For each parameter, this plots the cumulative distribution function of the percentile scores of the true parameters within the marginalized 1D posteriors. Since the percentiles should be distributed uniformly, the diagonal curves confirm that the network is properly sampling the posteriors.

In our experiments, we also varied n_{SVD}, and we found slightly reduced performance as this was increased. This indicates that, although with less compression it should be possible to produce tighter posteriors, better network optimization is required to take full advantage. Indeed, subleading SVD elements contain mostly noise, which makes training more difficult.

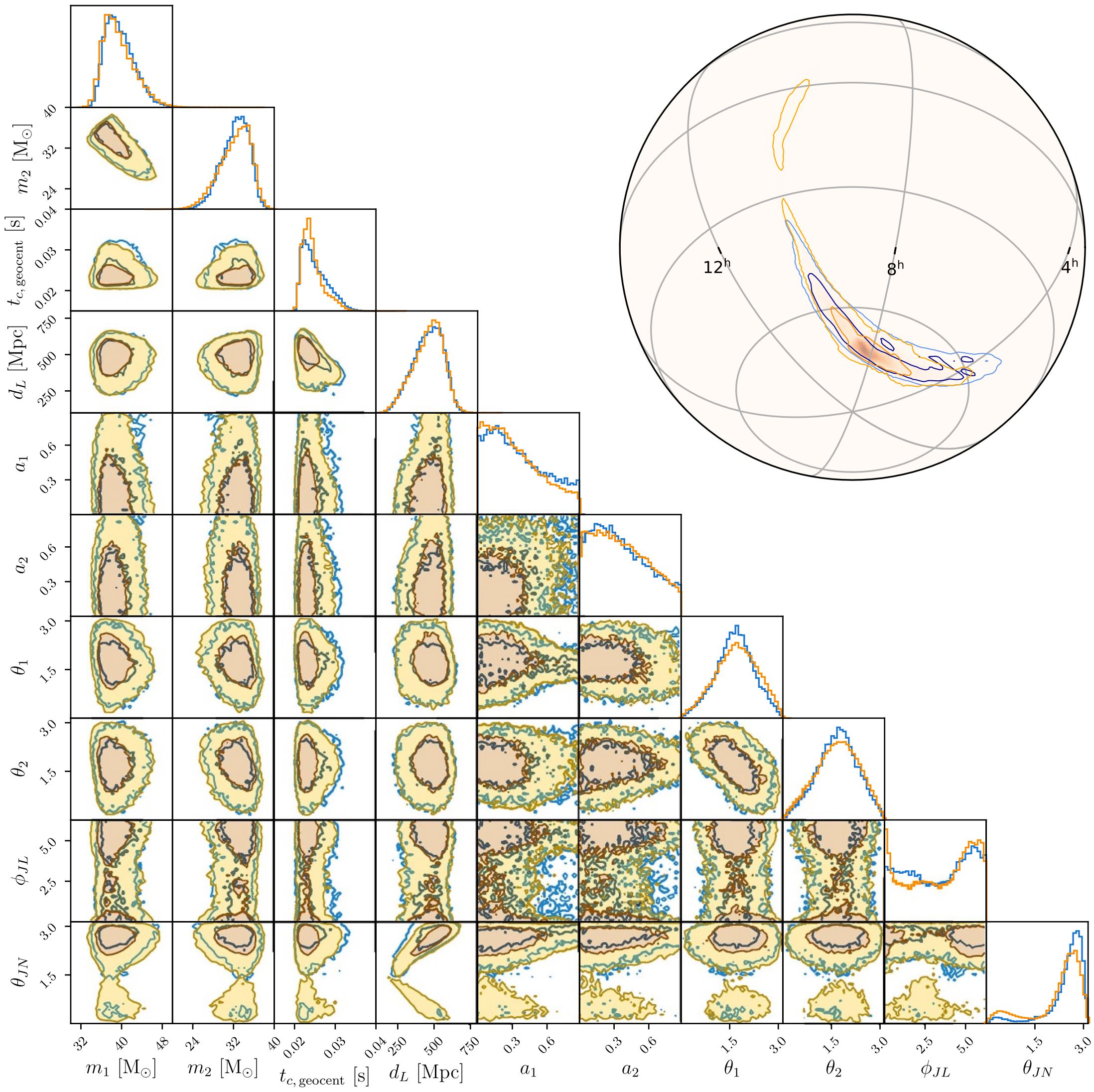

Figure 3. Marginalized 1D and 2D posterior distributions over a subset of parameters, comparing the normalizing flow (orange) and `bilby dynesty` (blue). Contours represent 50% and 90% credible regions. The inset shows the sky position, with rejection sampling used to obtain unweighted neural network samples.

5. Conclusions

In this study, we demonstrated for the first time that deep neural networks can accurately infer all 15 binary black hole parameters from real gravitational-wave strain data, including accurate estimates of uncertainties. The network learns a surrogate for the inverse model, so that generating a posterior sample requires just a forward pass through the neural network, which takes as input the detector data. Once the network is trained, inference is therefore extremely fast, generating 5000 independent samples per second.

This study is the first to demonstrate the use of neural networks for gravitational-wave inference in a realistic setting. Past studies [15–17] have all been subject to a number of restrictions: they all restricted analysis to a subset of parameters, they assumed design sensitivity noise, and they did not analyze real data. The form of the posteriors was also restricted in all of these studies, either by explicit restriction to Gaussian distributions [15], or by not having sufficiently flexible neural density estimators [16, 17]. All of these restrictions have been lifted in the present study, and we have demonstrated results in very close agreement with standard samplers. This came about in large part because of the use of the neural spline flow architecture, but also due to the careful strain data compression using SVD and sampling of extrinsic parameters during training.

This work should have significant impact on gravitational-wave data analysis: rapid parameter estimation is critical for multimessenger follow-up and for confronting the expected high rate of future detections. An advantage of our approach is that waveform generation is done in advance of training and inference, rather than at sampling time as for conventional methods. Thus, waveform models that include more physics but

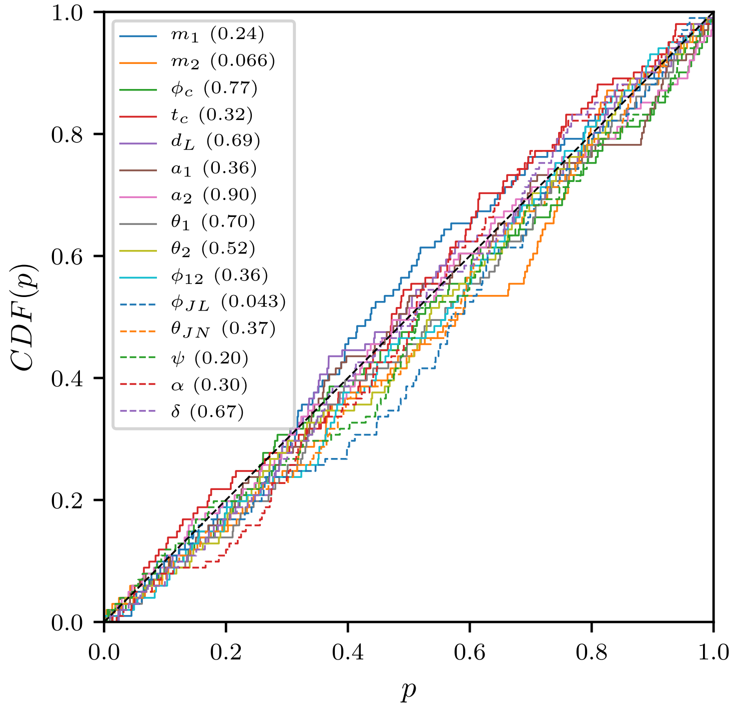

Figure 4. P–P plot for 100 artificial strain data sets analyzed by the neural network. For each injection and 1D posterior distribution, we compute the percentile value of the injected parameter. The figure shows the cumulative distribution function of the injections for each parameter, which should lie close to the diagonal if the network is performing properly. KS test p-values are given in the legend.

may be slower to evaluate [42] can be used to analyze data in the same time as faster models. Going forward, a major priority will be to extend the prior range over masses to include binary neutron stars [43], since rapid sky localization for these events is important for finding electromagnetic counterparts. This will require improved compression of the strain data, possibly involving convolutional embedding networks.

The network we presented is tuned to a particular noise PSD—in this case, estimated just prior to GW150914. Although the noise characteristics of the LIGO and Virgo detectors are mostly stable during an observing run, they do vary slightly from event to event, so in future studies these variations should be taken into account. Indeed, we would like to fully amortize training costs by building a conditional density estimator that can do inference on any event without retraining for the change in PSD. One approach would be to condition the model on PSD information: during training, waveforms would be whitened with respect to a PSD drawn from a distribution representing the variation in detector noise from event to event, and (a summary of) this PSD information would be passed to the network as additional context. (PSD samples can be obtained from detector data at random times.) For inference, PSD information would then be passed along with the whitened strain data.

Likelihood-free inference methods are also an avenue to move beyond the idealization of stationary Gaussian noise. This approximation is imposed in standard approaches by the requirement of having a likelihood function. However, since training the conditional density estimator only requires the ability to simulate data, it should be possible to do inference even in the presence of non-Gaussian noise artifacts such as glitches. One could, for instance, generate training data by injecting simulated waveforms into real detector noise.

In contrast to CVAEs used in past studies [16, 17], normalizing flows have the advantage of estimating the density directly, without any need to marginalize over latent variables. This means that the loss function can be taken to be the cross-entropy (4) rather than an upper bound [18, 19]. Moreover, since $q(\theta|s)$ is a normalized probability distribution, the Bayesian evidence can be obtained as a byproduct. The performance we achieved without latent variables in this study was made possible by the use of a more powerful normalizing flow [25] compared to [17]. As new and more powerful normalizing flows are developed by computer scientists in the future, they will be straightforward to deploy to further improve the performance and capabilities of deep learning for gravitational-wave parameter estimation.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: 10.5281/zenodo.4558988. Other data (including trained neural networks) that support the findings of this study are available upon request from the authors.

Acknowledgments

We would like to thank C Simpson for helpful discussions and for performing training runs during the early stages of this study. Our code was implemented in `PyTorch` [44], and with the neural spline flow implementation of [29]. Plots were produced with `matplotlib` [45], `ChainConsumer` [46] and `ligo.skymap` [47]. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation.

ORCID iDs

Stephen R Green https://orcid.org/0000-0002-6987-6313
Jonathan Gair https://orcid.org/0000-0002-1671-3668

References

[1] Abbott B P *et al* (LIGO Scientific, Virgo) 2016 Observation of gravitational waves from a binary black hole merger *Phys. Rev. Lett.* **116** 061102
[2] Abbott B P *et al* (LIGO Scientific, Virgo) 2019 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** 031040
[3] Abbott R *et al* (LIGO Scientific, Virgo) 2020 GW190412: observation of a binary-black-hole coalescence with asymmetric masses *Phys. Rev. D* **102** 043015
[4] Abbott B P *et al* (collaborationLIGO Scientific, Virgo) 2020 GW190425: observation of a compact binary coalescence with total mass $\sim 3.4 M_\odot$ *Astrophys. J. Lett.* **892** L3
[5] Abbott R *et al* (collaborationLIGO Scientific, Virgo) 2020 GW190814: gravitational waves from the coalescence of a 23 solar mass black hole with a 2.6 solar mass compact object *Astrophys. J.* **896** L44
[6] Abbott R *et al* (collaborationLIGO Scientific, Virgo) 2020 GWTC-2: compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run (arXiv:2010.14527) [gr-qc]
[7] Gravitational-wave candidate event database (https://gracedb.ligo.org/superevents/public/O3/) (accessed 22 August 2020)
[8] Veitch J *et al*, 2015 Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library *Phys. Rev.* **D91** 042003
[9] Romero-Shaw I M *et al* 2020 Bayesian inference for compact binary coalescences with BILBY: validation and application to the first LIGO–Virgo gravitational-wave transient catalogue *Mon. Notices Royal Astron. Soc* **499** 3295–319
[10] Abbott B P *et al* (collaborationLIGO Scientific, Virgo) 2019 Properties of the binary neutron star merger GW170817 *Phys. Rev.* **X9** 011001
[11] Abbott B P *et al* (collaborationKAGRA, LIGO Scientific, VIRGO) 2018 Prospects for observing and localizing gravitational-wave transients with advanced LIGO, advanced Virgo and KAGRA *Living Rev. Rel.* **21** 3
[12] George D and Huerta E A 2018 Deep learning for real-time gravitational wave detection and parameter estimation: results with advanced LIGO data *Phys. Lett.* B **778** 64–70
[13] Allen G *et al* 2019 Deep learning for multi-messenger astrophysics: a gateway for discovery in the big data era (arXiv:1902.00522) [astro-ph.IM]
[14] Cuoco E *et al* 2020 Enhancing gravitational-wave science with machine learning *Mach. Learn.: Sci. Technol.* **2** 011002
[15] Chua A J K and Vallisneri M 2020 Learning Bayesian posteriors with neural networks for gravitational-wave inference *Phys. Rev. Lett.* **124** 041102
[16] Gabbard H, Messenger C, Heng I S, Tonolini F, and Murray-Smith R, 2019 Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy (arXiv:1909.06296) [astro-ph.IM]
[17] Green S R, Simpson C and Gair J 2020 Gravitational-wave parameter estimation with autoregressive neural network flows *Phys. Rev.* D **102** 104057
[18] Kingma D P and Welling M 2013 Auto-encoding variational Bayes (arXiv:1312.6114) [stat.ML]
[19] Rezende D J, Mohamed S and Wierstra D 2014 Stochastic backpropagation and approximate inference in deep generative models *Int. Conf. Machine Learning* pp 1278–86 (http://proceedings.mlr.press/v32/rezende14.html)
[20] Rezende D and Mohamed S 2015 Variational inference with normalizing flows *Int. Conf. Machine Learning* pp 1530–8 (http://proceedings.mlr.press/v37/rezende15.html)
[21] Kingma D P, Salimans T, Jozefowicz R, Chen Xi, Sutskever I and Welling M 2016 Improved variational inference with inverse autoregressive flow *Advances in Neural Information Processing Systems* pp 4743–51 (http://papers.nips.cc/paper/6581-improved-variational-inference-with-inverse-autoregressive-flow)
[22] Papamakarios G, Pavlakou T and Murray I 2017 Masked autoregressive flow for density estimation *Advances in Neural Information Processing Systems* pp 2338–47 (http://papers.nips.cc/paper/6828-masked-autoregressive-flow-for-density-estimation)
[23] Chen Xi, Kingma D P, Salimans T, Duan Y, Dhariwal P, Schulman J, Sutskever I, and Abbeel P 2016 Variational Lossy autoencoder (arXiv:1611.02731) [cs.LG]
[24] Advanced LIGO anticipated sensitivity curves (https://dcc.ligo.org/LIGO-T0900288-v2/public) LIGO Technical Document, LIGO-T0900288-v2 2009
[25] Durkan C, Bekasov A, Murray I, and Papamakarios G 2019 Neural spline flows (arXiv:1906.04032) [stat.ML]

[26] Kaiming H, Zhang X, Ren S, and Sun J 2016 Identity mappings in deep residual networks (arXiv:1603.05027) [cs.CV]
[27] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *Int. Conf. Machine Learning* pp 448–56 (http://proceedings.mlr.press/v37/ioffe15.html)
[28] Clevert D-A, Unterthiner T, and Hochreiter S 2015 Fast and accurate deep network learning by exponential linear units (elus) (arXiv:1511.07289) [cs.LG]
[29] Durkan C, Bekasov A, Murray I, and Papamakarios G 2019 Neural spline flows (https://github.com/bayesiains/nsf)
[30] Farr B, Ochsner E, Farr W M and O'Shaughnessy R 2014 A more effective coordinate system for parameter estimation of precessing compact binaries from gravitational waves *Phys. Rev.* D **90** 024018
[31] Dinh L, Krueger D, and Bengio Y 2014 Nice: non-linear independent components estimation (arXiv:1410.8516) [cs.LG]
[32] Kaiming H, Zhang X, Ren S, and Sun J 2015 Deep residual learning for image recognition (arXiv:1512.03385) [cs.CV]
[33] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980) [cs.LG]
[34] Hannam M, Schmidt P, Boh'e A, Haegel L, Husa S, Ohme F, Pratten G and Pürrer M 2014 Simple model of complete precessing black-hole-binary gravitational waveforms *Phys. Rev. Lett.* **113** 151101
[35] Khan S, Husa S, Hannam M, Ohme F, Pürrer M, Forteza X J and Bohé A 2016 Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era *Phys. Rev.* **D93** 044007
[36] Bohé A, Hannam M, Husa S, Ohme F, Pürrer M, and Schmidt P 2016 PhenomPv2—technical notes for the LAL implementation LIGO Technical Document, LIGO-T1500602-v4 (https://dcc.ligo.org/LIGO-T1500602/public)
[37] Canizares P, Field S E, Gair J R and Tiglio M 2013 Gravitational wave parameter estimation with compressed likelihood evaluations *Phys. Rev.* D **87** 124005
[38] Rory Smith S E Field K B, Haster C-J, Pürrer M, Raymond V and Schmidt P 2016 Fast and accurate inference on gravitational waves from precessing compact binaries *Phys. Rev.* D **94** 044031
[39] Loshchilov I and Hutter F 2016 Sgdr: stochastic gradient descent with warm restarts (arXiv:1608.03983) [cs.LG]
[40] Ashton G *et al*„ 2019 BILBY: a user-friendly Bayesian inference library for gravitational-wave astronomy *Astrophys. J. Suppl.* **241** 27
[41] Speagle J S 2020 Dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences *Mon. Not. R. Astron. Soc.* **493** 3132–58
[42] Ossokine S *et al* 2020 Multipolar effective-one-body waveforms for precessing binary black holes: construction and validation *Phys. Rev. D* **102** 044055
[43] Krastev P G, Gill K, Ashley Villar V and Berger E 2021 Detection and parameter estimation of gravitational waves from binary neutron-star mergers in real LIGO data using deep learning *Phys. Lett.* B **815** 136161
[44] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems 32*, ed H Wallach, H Larochelle, A Beygelzimer, F d'Alché Buc, E Fox and R Garnett (Red Hook, NY: Curran Associates, Inc.) pp 8024–35
[45] Hunter J D 2007 Matplotlib: a 2d graphics environment *Comput. Sci. Eng.* **9** 90–5 (http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library)
[46] Hinton S R 2016 ChainConsumer *J. Open Source Softw.* **1** 00045
[47] Singer L 2020 ligo.skymap (https://lscsoft.docs.ligo.org/ligo.skymap/)