Fast Parameter Inference on Pulsar Timing Arrays with Normalizing Flows

David Shih,[1] Marat Freytsis,[2] Stephen R. Taylor,[3] Jeff A. Dror,[4,5] and Nolan Smyth[5]

[1]*New High Energy Theory Center, Rutgers University, Piscataway, New Jersey 08854-8019, USA*
[2]*LEPP, Department of Physics, Cornell University, Ithaca, New York 14853, USA*
[3]*Department of Physics and Astronomy, Vanderbilt University, 2301 Vanderbilt Place, Nashville, Tennessee 37235, USA*
[4]*Department of Physics, University of Florida, Gainesville, Florida 32611, USA*
[5]*Department of Physics, University of California Santa Cruz, 1156 High Street, Santa Cruz, California 95064, USA*

(Received 12 February 2024; revised 1 May 2024; accepted 21 May 2024; published 2 July 2024)

Pulsar timing arrays perform Bayesian posterior inference with expensive Markov chain Monte Carlo (MCMC) methods. Given a dataset of ~10–100 pulsars and $\mathcal{O}(10^3)$ timing residuals each, producing a posterior distribution for the stochastic gravitational wave background (SGWB) can take days to a week. The computational bottleneck arises because the likelihood evaluation required for MCMC is extremely costly when considering the dimensionality of the search space. Fortunately, generating simulated data is fast, so modern simulation-based inference techniques can be brought to bear on the problem. In this Letter, we demonstrate how conditional normalizing flows trained on simulated data can be used for extremely fast and accurate estimation of the SGWB posteriors, reducing the sampling time from weeks to a matter of seconds.

DOI: 10.1103/PhysRevLett.133.011402

Introduction.—Pulsar timing array (PTA) experiments have recently announced evidence for an all-sky background of gravitational waves (GWs) in a frequency window of ~1–100 nHz [1–4]. These experiments leverage the exceptional timing regularity of millisecond pulsars to search for quadrupolarlike correlated arrival-time deviations of radio pulses, thereby signaling the presence of GWs. The origin of these GWs is still uncertain, but a known source in this frequency range is from a population of subparsec-separated supermassive black-hole binaries whose individual GW signals superpose incoherently to produce a stochastic background (see, e.g., [5,6], and references therein). Additionally, there could be gravitational waves arising from new physics in the early Universe [7]. The origin of this signal, and the underlying astrophysics and cosmology leading to it, will become better understood as existing pulsars are timed longer and more pulsars are added to arrays [8]. Yet this will lead to existing PTA inference strategies becoming ever more taxed, lengthening analysis times and reducing the scope of studies that can be tackled with available computational resources. There are ongoing efforts to resolve this problem, including techniques that refit on intermediate analysis products and condense them into sufficient statistics [9]. However, these efforts have yet to employ deep learning methods, which hold great potential to accelerate and improve the sensitivity of PTA GW inference. Deep learning has already shown great promise in ground-based GW data analysis (e.g., [10–15]).

One of the essential aims of PTA inference is to learn the posterior density $p(\theta|r)$ in parameter space [e.g., the amplitude and power-law slope of the stochastic gravitational wave background (SGWB) signal] given the data r. The standard approach to deriving this posterior density is Markov chain Monte Carlo (MCMC) sampling of the true likelihood $p(r|\theta)$, which is modeled as a Gaussian distribution over the data. The covariance for individual pulsars is determined by red and white noise processes, while the interpulsar covariance is based on the Hellings-Down curve [16] of the SGWB. To evaluate this likelihood, one must invert a large covariance matrix (with dimensions of the product of the number of pulsars and twice the number of GW frequencies searched) for each parameter vector visited in an MCMC chain. Deriving a single posterior density for the NANOGrav 12.5 year dataset [17] requires $\gtrsim 5$ days of computation to explore a Hellings-Downs-correlated model of the SGWB, while simultaneously sampling the intrinsic pulsar red noise processes. (Likelihood evaluation times on this dataset for a Hellings–Downs-correlated model can be ~0.1–1 s. At least 10^6 likelihood iterations are usually performed in an MCMC exploration of the model space, resulting in ~5 days of computational wall time.)

Although the PTA likelihood $p(r|\theta)$ is slow to evaluate, it is extremely fast to sample from, since the time series have diagonal covariance in Fourier space, and the Fourier transform is fast to evaluate. This is an ideal situation for simulation-based inference (SBI) (see, e.g., [18] for a review). By generating a large training dataset consisting of pairs of parameters and time series (θ, r) (the dataset is distributed according to the prior on θ), the SBI technique known as *neural posterior estimation* [19–21] can then be used to learn a fast-sampling posterior density $p(\theta|r)$ from the samples. Since this learned posterior is conditioned on the data r, it can be used to quickly analyze any new data that is covered by the training dataset. (There are many other SBI techniques for learning the posterior or likelihood from samples, e.g., neural likelihood estimation or neural ratio estimation. We refer to the reader to [18] for more details.) Neural posterior estimation has already proven to be highly successful across a wide range of domains (see [22,23] for curated and continuously updated bibliographies). Here, we demonstrate for the first time the power of using neural posterior estimation to analyze pulsar timing data.

In order to learn the posterior from samples, we will use the method of *normalizing flows* (NFs). Normalizing flows are a powerful method for density estimation and generative modeling (see [24,25] for reviews and original references). Using highly expressive neural networks, they aim to learn an invertible transformation with tractable Jacobian between any data distribution to a latent space following a simple prespecified distribution (such as Gaussian or uniform). By running this transformation in one direction, one can estimate the probability density of any point in the dataset; running it in the other direction, one can generate more samples that follow the same distribution as the data.

Data.—Following the recent PTA literature [26,27], pulsar timing observations are modeled with a leading contribution due to a timing ephemeris, which upon fitting and subtracting from the observations leaves a set of timing residuals. In SGWB searches—i.e., ignoring possible deterministic GW signals—these residuals are modeled as random Gaussian processes, fully characterized by their power spectra in frequency space. The relevant parameters of the different random Gaussian components are white noise; individual "red noise" for each pulsar I with $A_r^{(I)}$ and $\gamma_r^{(I)}$ describing, respectively, the amplitude and exponent of a power-law model in frequency space; and the SGWB which is common to all pulsars, with amplitude A_{GW} and γ_{GW}, again for a power-law spectral model in frequency space. If supermassive black-hole mergers are the dominant source of gravitational waves in the nanohertz range, then it is expected that pulsar timing will observe $\gamma_{\mathrm{GW}} \approx 13/3$ [28]. We will keep it as a free parameter in this study and seek to infer it from the data.

Our goal is to build mock pulsar timing datasets that model the key sources of signal and noise. Although common software frameworks exist for pulsar timing analysis (most notably LIBSTEMPO [29] and PINT [30]), we choose to generate our training data using our own code for greater simulation efficiency, control over the data, and understanding of the results. (For details of our methods, see Supplemental Material [31].) Using our own framework, we generate one million mock PTA residual time series for $N_p = 10$ pulsars with observation times drawn from the epoch-averaged NANOGrav 12.5 year dataset [17]. We use the ten pulsars which were found to contribute the most evidence toward an (isotropic) SGWB signal in the NANOGrav analysis of their dataset [35]. Table I describes these ten pulsars and their best-fit red noise parameters. The white noise is fixed to 100 ns for all pulsars; red noise is sampled independently for each pulsar, from a uniform distribution, $\log_{10}A_r^{(i)} \in [-19, -13]$, $\gamma_r^{(i)} \in [1, 7]$; and the SGWB is sampled uniformly from $\log_{10}A_{\mathrm{GW}} \in [-18, -13]$, $\gamma_{\mathrm{GW}} \in [1, 7]$. Red noise and SGWB contributions to the residuals are generated in frequency space and then Fourier transformed to the time domain; see Supplemental Material [31] for details. We include a minimal pulsar timing model for each pulsar which accounts for a time offset, the pulsar period, and the rate of change of the pulsar period. To remove the dependence on the Fourier transform base frequency and the pulsar timing model, we apply the G-matrix projection, following [36]; this projects the timing residuals into the null space of the timing-model design matrix, which is equivalent to marginalizing over linear deviations to the timing-model parameters [37]. After the G-matrix projection, there are a total of 2940 projected residuals across the ten pulsars. Of the 10^6 generated time series, we reserve 90% for training the normalizing flow (to be described in the next section) and 10% for validation (model selection).

TABLE I. The ten pulsars used in this analysis, their number of residuals, and best-fit red noise parameters A_r and γ_r.

Name	No. residuals	Best-fit $\log_{10} A_r$	Best-fit γ_r
J1909 − 3744	408	−15.08	1.73
J2317 + 1439	447	−17.08	3.20
J2043 + 1711	302	−16.39	2.94
J1600 − 3053	236	−13.54	0.61
J1918 − 0642	262	−16.38	2.68
J0613 − 0200	278	−14.46	2.16
J1944 + 0907	136	−16.51	3.06
J1744 − 1134	268	−13.62	2.45
J1910 + 1256	170	−16.70	3.25
J0030 + 0451	463	−15.08	4.89

It was a challenge to preprocess the residuals into a form that enabled the flow to learn effectively. They spanned an enormous range—nearly 14 orders of magnitude—due to the wide log-uniform priors taken for the red noise and SGWB amplitudes. However, a simple log-transform of the residuals was not possible, since they could take either sign. Instead, we found that rescaling the residuals, $r \to 10^7 \times r$, followed by a clipping ± 1000, worked well to make the inputs of the neural network $\mathcal{O}(1)$. This focuses on the part of the parameter space of greatest interest (the weakly detectable SGWB regime) and might lose sensitivity to the part of parameter space of less interest (a huge SGWB signal, which is anyways incompatible with current observations). Finally, each time series $r \in \mathbb{R}^{2940}$ is paired with

TABLE II. The neural network architectures that define our embedding and our posterior density models.

E_I network	LSTM	Num_layers = 2
		Hidden_size = 100
		Output_dim = 400

E_I network	MLP	Hidden_size = 200,100
		Output_dim = 20

F network	MLP	Hidden_size = 100,100
		Output_dim = 50

Posterior network	MAF	$N_{\mathrm{MADE}} = 8$ Hidden_size = 200,200 Base: uniform Transform: RQS (8 bins)

the set of parameters which characterize the stochastic noise, $\theta \in \mathbb{R}^{22}$; these parameters are

$$\theta = (\log_{10} A_{\mathrm{GW}}, \gamma_{\mathrm{GW}}, \log_{10} A_r^{(1)}, \gamma_r^{(1)}, \ldots, \log_{10} A_r^{(10)}, \gamma_r^{(10)}). \tag{1}$$

Since the training data are generated with a uniform distribution, we preprocess θ with a simple shift and rescaling so $\theta \in [-1, 1]^{22}$.

Maximum likelihood (ML) setup.—We fit a conditional normalizing flow to samples (θ, r) using the maximum likelihood loss objective in order to estimate the posterior density $p(\theta|r)$. We use masked autoregressive flows (MAFs) [38] with rational quadratic spline (RQS) transformations [39,40]; for the details of the hyperparameter choices, see Table II. We note that our choice of base distribution was motivated by the uniformly sampled training data; in early tests, we found the flow performed better this way compared to using a Gaussian base distribution.

Rather than feeding all 2940 residuals directly to the NF, we first pass them through an auxiliary embedding network. This is a popular trick for improving the performance of neural posterior estimation (see, e.g., [15,41])—by compressing the inputs down to a more informative feature vector, the embedding network mitigates the curse of dimensionality.

We found the following multistage architecture worked well:

$$r' = F\big(E_1(r_1), E_2(r_2), \ldots, E_{10}(r_{10})\big). \tag{2}$$

Here, each E_I takes the (preprocessed) residuals of pulsar I and returns a per-pulsar embedding; then, F takes the concatenation of these embeddings and returns an overall embedding. Each E_I consists of a two-layer long short-term memory (LSTM) network [42] followed by an multi-layer perceptron (MLP). (The MLP takes as input the concatenation of the hidden and cell states from each LSTM layer.) Meanwhile, F is just a simple two-layer MLP that takes as input the concatenation of the per-pulsar embeddings and outputs a final 50-dimensional embedding vector. For details of the architecture, we again refer the reader to Table II. We found that using an LSTM in the E_I instead of just an MLP improved the performance of the network significantly. So did using a two-stage per-pulsar structure instead of feeding all 2940 residuals to a single LSTM or MLP.

To implement our normalizing flow and embedding network, we use the NFLOWS package [43] and PyTorch [44]. The entire setup (NF plus embedding network) is trained concurrently using the log-likelihood objective and the RAdam optimizer [45] with default parameters and a batch size of 256. The networks are trained for up to 100 epochs, and the epoch with the best validation loss is chosen for the final demonstration. The training took 15 min per epoch (totalling 25 h for 100 epochs) on three Nvidia P100 GPUs, while sampling the flow to produce the posteriors takes approximately 5.6 s per 100 000 samples. Meanwhile, sampling the equivalent number of posterior draws using the traditional MCMC pipeline takes approximately 5.5 h on an Apple M3 Pro chip. Although the flow may take longer to train than a single run of the MCMC, it can be reused to quickly generate posteriors for as many datasets as needed. For the purposes of large-scale simulation studies on a PTA with fixed design, this will be enormously beneficial.

Results.—We first show in Fig. 1 the posteriors recovered by our normalizing flow for a 2 × 2 grid of $(\log_{10} A_{\mathrm{GW}}, \gamma_{\mathrm{GW}})$ values. For each parameter choice, a single instance of pulsar residuals is generated and fed as conditional labels to the trained flow, and 100 000 samples in parameter space are generated from the flow. (In this example and all the subsequent ones, we fix the injected red noise values to their nominal best-fit values shown in Table I.) We use CHAIN_CONSUMER [46] to plot the posteriors from the samples. In Fig. 1, we also show posteriors obtained from the exact PTA likelihood, using pulsar dataset simulations passed through the ENTERPRISE [47] PTA data analysis pipeline, and sampled using MCMC. We can see that the flow-generated posteriors are already quite accurate, matching the true MCMC posteriors reasonably well across the parameter space. In particular, the flow posteriors more or less cover the MCMC ones and correctly indicate when the SGWB parameters can be recovered vs when the amplitude or the slope are too small and the posterior corresponds to only an upper limit. [Indeed, the flow correctly reports that the posterior in these cases carries no information below an approximately diagonal line in $\log_{10}(A_{\mathrm{GW}})$ vs γ_{GW}. This is expected from the nature of the PTA, whereby the bulk of the constraint on the SGWB comes from the lowest frequencies, and there one can trade off amplitude for slope as indicated in the third panel in Fig. 1.]

Next, we drill down to the case of nominal SGWB values of $\gamma_{\mathrm{GW}} = 13/3$ and $A_{\mathrm{GW}} = 10^{-15}$. We can repeatedly generate timing residuals with these SGWB parameters

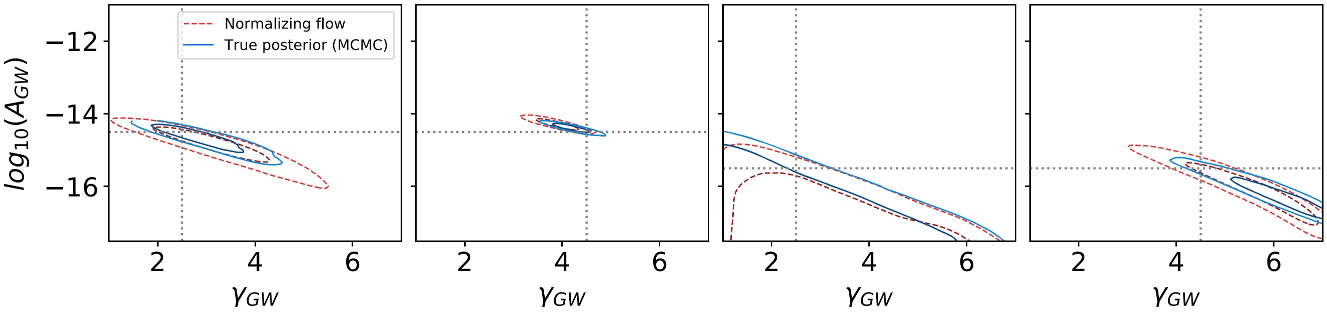

FIG. 1. A comparison of the flow-estimated posteriors and the MCMC-derived ground truth posteriors, for a grid of four different pairs of SGWB parameters: $(\gamma_{\rm GW}, \log_{10}A_{\rm GW}) = (2.5, -14.5), (4.5, -14.5), (2.5, -15.5), (4.5, -15.5)$ from left to right, respectively.

and see the variation in posteriors that result. Four instances are shown in Fig. 2. We again observe qualitatively good coverage of the true (MCMC) posterior. We quantify the agreement with the true posteriors using the Hellinger distance [48]. This is a distance measure between probability distributions which becomes tractable when the distributions are Gaussian—an approximation that empirically describes our two-dimensional posteriors quite well. We find that the mean and standard deviation in Hellinger distances between flow and true posteriors calculated across ten instances is 0.33 ± 0.04.

Finally, the flow-generated samples can be made more precise by reweighting them with the true likelihoods. This reweighting technique has been explored previously in the PTA literature by Hourihane *et al.* [49], who studied the efficacy of reweighting an approximate posterior obtained by ignoring cross-correlations between pulsars. (See also [50], which explored a very similar reweighting technique starting from flow-based posteriors, in the context of gravitational wave interferometry.) Given a parameter point $\theta_a \sim p_{\rm flow}(\theta|r)$ sampled from the flow, we can calculate the true likelihood of these parameter points $p_{\rm true}(r|\theta_a)$. Up to an overall normalization (the Bayesian evidence), this can be used to determine the weights required to turn the flow samples into samples following the true posterior:

$$w_a = \frac{p_{\rm true}(r|\theta_a)p(\theta_a)}{p_{\rm flow}(\theta_a|r)}. \tag{3}$$

(Furthermore, the average of the weights provides an estimate for the Bayesian evidence. This can be used as a metric to compare two different models; if the importance sampling with a given N does not accurately approximate the integral over the true posterior, the estimated evidence will be biased to lower values [50].)

The reweighted flow posteriors are shown in Fig. 3 compared with the MCMC posteriors for a single instance of timing residuals generated from the nominal SGWB parameters. We see the reweighted posteriors are significantly improved over the posteriors sampled from the uncorrected flow, basically in perfect agreement with the MCMC.

With the reweighted posteriors, the Hellinger distances (again, calculated over ten instances) improve to 0.22 ± 0.15. This can be benchmarked against the Hellinger distances between the posteriors obtained through MCMC with different random seeds; evaluating 100 additional random realizations of the MCMC against the one that we have been using here as the point of comparison, we find 0.01 ± 0.01.

Meanwhile, another common measure of the quality of importance sampling (used, e.g., in [49,50]) is the *weighting efficiency* (which is closely related to the *effective sample size* [51])

$$\epsilon = \frac{1}{N}\frac{\left(\sum_{a=1}^N w_a\right)^2}{\sum_{a=1}^N w_a^2}. \tag{4}$$

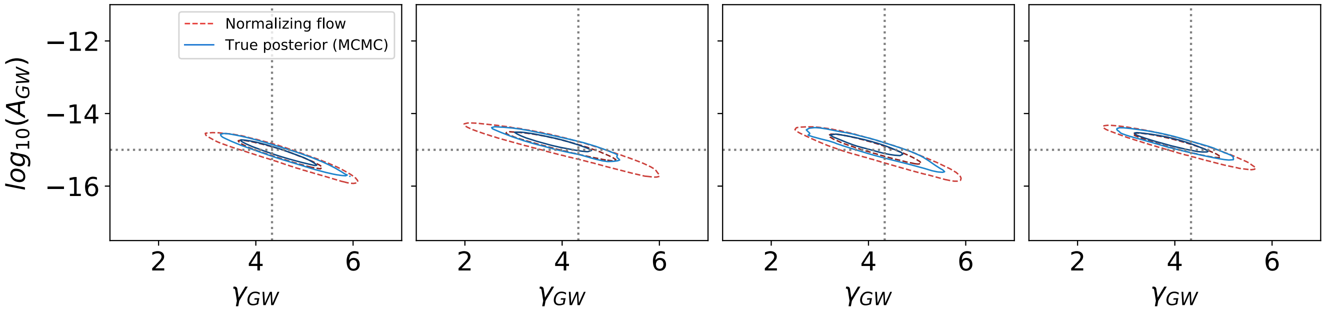

FIG. 2. A comparison of the flow-estimated posteriors and the MCMC-derived ground truth posteriors, for four time series sampled from a single choice of SGWB parameters, $(\gamma_{\rm GW}, \log_{10}A_{\rm GW}) = (13/3, -15)$.

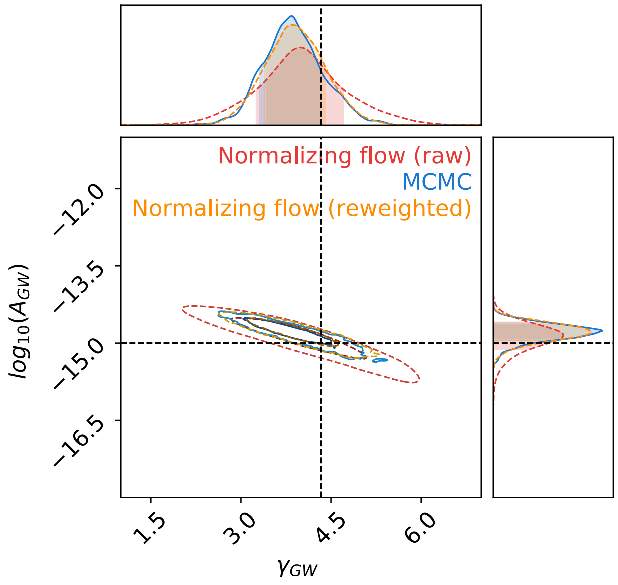

FIG. 3. Posteriors from the raw normalizing flow (red line), the reweighted normalizing flow (orange line), and the MCMC-derived ground truth (blue line), for a single realization of the PTA residuals with the same nominal SGWB parameters as in Fig. 2.

This is a statistical measure of the fraction of unweighted samples that would be required to match the variance of the weighted sample. The weighting efficiency is estimated (using $N = 10^6$ samples) to be $\log_{10}(\varepsilon) = -3.2 \pm 0.7$.

Clearly, both of these metrics indicate that there could be room for further improvement of the neural posterior estimation quality. This is entirely to be expected, since we did not extensively optimize the hyperparameters for this first proof-of-concept work.

Although reweighting the flow posteriors also requires evaluating the true likelihoods, it is an interesting alternative to the MCMC, for several reasons. First, these calls are fully uncorrelated and, hence, fully parallelizable, whereas the MCMC samples always suffer from some correlation and need to be evaluated (at least to some degree) sequentially. Second, the specific numbers and comparisons presented here are not set in stone—the quality of the flow can likely be improved systematically with additional improvements to the neural network architecture. This will reduce the number of likelihood evaluations required for the reweighting, further improving the comparison with the MCMC (which is a stable, mature technique).

Conclusions.—We have shown, using simulations that capture some of the challenging aspects of analyzing real PTA datasets—e.g., uneven cadence, different noise properties, and timing models in each pulsar—that normalizing flows offer enormous potential to vastly accelerate PTA data analysis and parameter inference with almost no loss in accuracy or sensitivity. Going forward, we expect these techniques enabled by modern machine learning to revolutionize the PTA field, complement the traditional MCMC-based techniques, and, with the influx of high-cadence data and new pulsars from forthcoming flagship radio facilities, ultimately replace the status quo pipelines. Machine-learning techniques, like the one we have studied here, will safeguard the future tractability and scalability of nanohertz-frequency GW analyses, ushering in a new era of discovery with PTA data.

There is much that can be improved in our pilot study. On the purely ML side, the specific architecture taken here (MAF-RQS normalizing flow with LSTM-based embedding) was not heavily optimized for performance, and it is likely that, with a more dedicated hyperparameter scan, the performance of the flow-based posterior estimation could be greatly improved. It would also be fruitful to explore different architectures, e.g., embeddings based on transformers, or more expressive alternatives to ordinary normalizing flows such as diffusion models [52–56] or continuous normalizing flows [57]. We should also point out that posteriors obtained via simulation-based inference are not guaranteed to be conservative [58] (which has implications for the effectiveness of importance sampling), and it is an interesting future direction to explore improved techniques such as [59] that guarantee more conservative posteriors. In any event, the results shown here should not be taken as the ultimate limit of what modern ML techniques can achieve but just the starting point.

Pulsar timing data are highly heterogeneous in quality and regularity, as the limitations of legacy data are joined with the ever-improving sensitivity of modern data. Furthermore, there are many processes associated with the propagation of radio pulses in the ionized interstellar medium that leave their imprint on pulsar timing data, and it is known that a one-size-fits-all approach to modeling these effects in pulsars is not appropriate. Machine-learning strategies must be able to accommodate the rich variety of noise processes with which pulsar timing data must contend and be able to do so on a per-pulsar basis. Additionally, GW signals in the PTA band are a combination of stochastic (e.g., the GW background) and deterministic (e.g., individually resolvable binary signals, or bursts), and machine-learning pipelines need to be able to model these with the same or better flexibility as current likelihood-centered approaches. Perhaps the most important improvement that must be made is the fact that the datasets are continually growing. Ideally, a neural network would not need to be completely retrained to incorporate the extension of existing datasets or, indeed, their expansion with additional pulsars.

Out of the improvements we have identified, there also lie opportunities. Simultaneous characterization of a GW background and a (perhaps variable) number of single resolvable GW signals remains challenging for current pipelines. Iterative refinement of pulsar noise models is also time consuming and somewhat *ad hoc*, in that it may depend on the assumed base model from which iteration is begun, and it currently does not take place at the level of the full array but rather independently in each pulsar. Deep learning, while not a panacea, could be well placed to tackle such complicated, high-dimensional decisions. If so, the discovery potential of PTAs will continue to grow, belying the long-timescale nature of the experiment to offer regular GW, pulsar, and interstellar-medium breakthroughs.

We thank Gabriel Freedman and members of the NANOGrav Detection Working Group for their useful comments on a preliminary form of this work. D. S. is supported by Department of Energy (DOE) Grant No. DOE-SC0010008. M. F. was supported by the DOE under Grant No. DE-SC0010008 and the National Science Foundation (NSF) under Grant No. PHY1316222. S. R. T. acknowledges support from NSF AST-2007993, the NANOGrav NSF Physics Frontier Center No. 2020265, and an NSF CAREER No. 2146016. J. A. D. was supported in part by U.S. Department of Energy Grant No. DE-SC0023093. N. S. is supported in part by the NSF Graduate Research Fellowships Program (GRFP) under Grant No. DGE-1842400. This work was conducted in part using the resources of the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University, Nashville, Tennessee.

[1] G. Agazie *et al.* (NANOGrav Collaboration), Astrophys. J. Lett. **951**, L8 (2023).

[2] J. Antoniadis *et al.* (EPTA Collaboration), Astron. Astrophys. **678**, A50 (2023).

[3] D. J. Reardon *et al.*, Astrophys. J. Lett. **951**, L6 (2023).

[4] H. Xu *et al.*, Res. Astron. Astrophys. **23**, 075024 (2023).

[5] G. Agazie *et al.* (NANOGrav Collaboration), Astrophys. J. Lett. **952**, L37 (2023).

[6] J. Antoniadis *et al.* (EPTA Collaboration), arXiv:2306.16227.

[7] A. Afzal *et al.* (NANOGrav Collaboration), Astrophys. J. Lett. **951**, L11 (2023).

[8] N. S. Pol *et al.* (NANOGrav), Astrophys. J. Lett. **911**, L34 (2021).

[9] W. G. Lamb, S. R. Taylor, and R. van Haasteren, Phys. Rev. D **108**, 103019 (2023).

[10] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip, Phys. Rev. D **95**, 104059 (2017).

[11] M. Razzano and E. Cuoco, Classical Quantum Gravity **35**, 095016 (2018).

[12] D. George and E. A. Huerta, Phys. Lett. B **778**, 64 (2018).

[13] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Phys. Rev. D **100**, 063015 (2019).

[14] G. Vajente, Y. Huang, M. Isi, J. C. Driggers, J. S. Kissel, M. J. Szczepańczyk, and S. Vitale, Phys. Rev. D **101**, 042003 (2020).

[15] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Phys. Rev. Lett. **127**, 241103 (2021).

[16] R. W. Hellings and G. S. Downs, Astrophys. J. Lett. **265**, L39 (1983).

[17] M. F. Alam *et al.* (NANOGrav Collaboration), Astrophys. J. Suppl. Ser. **252**, 4 (2021).

[18] K. Cranmer, J. Brehmer, and G. Louppe, Proc. Natl. Acad. Sci. U.S.A. **117**, 30055 (2020).

[19] G. Papamakarios and I. Murray, arXiv:1605.06376.

[20] J.-M. Lueckmann, P. J. Goncalves, G. Bassetto, K. Öcal, M. Nonnenmacher, and J. H. Macke, arXiv:1711.01861.

[21] D. S. Greenberg, M. Nonnenmacher, and J. H. Macke, arXiv:1905.07488.

[22] K. Cranmer and J. Lo, Simulation-based inference, https://simulation-based-inference.org/.

[23] S. Mishra-Sharma, Awesome neural SBI, https://github.com/smsharma/awesome-neural-sbi.

[24] I. Kobyzev, S. J. Prince, and M. A. Brubaker, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3964 (2021).

[25] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, J. Mach. Learn. Res. **22**, 1 (2021).

[26] S. R. Taylor, *Nanohertz Gravitational Wave Astronomy* (CRC Press, Boca Raton, 2021).

[27] A. D. Johnson *et al.*, Phys. Rev. D **109**, 103012 (2024).

[28] E. S. Phinney, arXiv:astro-ph/0108028.

[29] M. Vallisneri, LIBSTEMPO: PYTHON wrapper for Tempo2, Astrophysics Source Code Library, record ascl:2002.017 (2020), ascl:2002.017.

[30] J. Luo, S. Ransom, P. Demorest, P. S. Ray, A. Archibald, M. Kerr, R. J. Jennings, M. Bachetti, R. van Haasteren, C. A. Champagne, J. Colen, C. Phillips, J. Zimmerman, K. Stovall, M. T. Lam, and F. A. Jenet, Astrophys. J. **911**, 45 (2021).

[31] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.133.011402 for additional details regarding the PTA simulations, which includes Refs. [32–34].

[32] G. B. Hobbs, R. T. Edwards, and R. N. Manchester, Mon. Not. R. Astron. Soc. **369**, 655 (2006).

[33] R. T. Edwards, G. B. Hobbs, and R. N. Manchester, Mon. Not. R. Astron. Soc. **372**, 1549 (2006).

[34] J. Luo, S. Ransom, P. Demorest, R. van Haasteren, P. Ray, K. Stovall, M. Bachetti, A. Archibald, M. Kerr, J. Colen, and F. Jenet, PINT: High-precision pulsar timing analysis package, Astrophysics Source Code Library, record ascl:1902.007 (2019), ascl:1902.007.

[35] Z. Arzoumanian *et al.* (NANOGrav Collaboration), Astrophys. J. Lett. **905**, L34 (2020).

[36] R. van Haasteren and Y. Levin, Mon. Not. R. Astron. Soc. **428**, 1147 (2013).

[37] R. van Haasteren and M. Vallisneri, Phys. Rev. D **90**, 104012 (2014).

[38] G. Papamakarios, T. Pavlakou, and I. Murray, arXiv:1705.07057.

[39] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, arXiv:1906.04032.

[40] J. A. Gregory and R. Delbourgo, IMA J. Numer. Anal. **2**, 123 (1982).

[41] S. Mishra-Sharma and K. Cranmer, Phys. Rev. D **105**, 063017 (2022).

[42] H. Sak, A. Senior, and F. Beaufays, arXiv:1402.1128.

[43] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, NFLOWS: Normalizing flows in PyTorch, https://zenodo.org/records/4296287 (2020).

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Curran Associates Inc., Red Hook, NY, 2019).

[45] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, arXiv:1908.03265.

[46] S. R. Hinton, J. Open Source Software **1**, 00045 (2016).

[47] J. A. Ellis, M. Vallisneri, S. R. Taylor, and P. T. Baker, ENTERPRISE: Enhanced Numerical Toolbox Enabling a Robust PulsaR Inference SuitE, https://zenodo.org/records/4059815 (2020).

[48] E. Hellinger, J. Reine Angew. Math. **136**, 210 (1909).

[49] S. Hourihane, P. Meyers, A. Johnson, K. Chatziioannou, and M. Vallisneri, Phys. Rev. D **107**, 084045 (2023).

[50] M. Dax, S. R. Green, J. Gair, M. Pürrer, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, Phys. Rev. Lett. **130**, 171403 (2023).

[51] A. Kong, A note on importance sampling using standardized weights, Technical Report No. 348, University of Chicago, Department of Statistics, 1992.

[52] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, arXiv:1503.03585.

[53] Y. Song and S. Ermon, arXiv:1907.05600.

[54] Y. Song and S. Ermon, arXiv:2006.09011.

[55] J. Ho, A. Jain, and P. Abbeel, arXiv:2006.11239.

[56] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, arXiv:2011.13456.

[57] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, arXiv:1806.07366.

[58] J. Hermans, A. Delaunoy, F. Rozet, A. Wehenkel, V. Begy, and G. Louppe, arXiv:2110.06581.

[59] A. Delaunoy, J. Hermans, F. Rozet, A. Wehenkel, and G. Louppe, arXiv:2208.13624.