Efficient gravitational wave searches with pulsar timing arrays using Hamiltonian Monte Carlo

Gabriel E. Freedman,[1] Aaron D. Johnson,[1,2] Rutger van Haasteren,[3] and Sarah J. Vigeland[1]

[1]*Center for Gravitation, Cosmology and Astrophysics, University of Wisconsin–Milwaukee,*
P.O. Box 413, Milwaukee Wisconsin 53201, USA
[2]*Theoretical AstroPhysics Including Relativity (TAPIR), MC 350-17, California Institute of Technology,*
Pasadena, California 91125, USA
[3]*Max-Planck-Institut für Gravitationsphysik (Albert-Einstein-Institut),*
Callinstrasse 38, D-30167, Hannover, Germany

(Received 2 November 2022; accepted 10 January 2023; published 10 February 2023)

Pulsar timing arrays (PTAs) detect low-frequency gravitational waves (GWs) by looking for correlated deviations in pulse arrival times. Current Bayesian searches use Markov chain Monte Carlo (MCMC) methods, which struggle to sample the large number of parameters needed to model the PTA and GW signals. As the data span and number of pulsars increase, this problem will only worsen. An alternative Monte Carlo sampling method, Hamiltonian Monte Carlo (HMC), utilizes Hamiltonian dynamics to produce sample proposals informed by first-order gradients of the model likelihood. This in turn allows it to converge faster to high dimensional distributions. We implement HMC as an alternative sampling method in our search for an isotropic stochastic GW background, and show that this method produces equivalent statistical results to similar analyses run with standard MCMC techniques, while requiring 100–200 times fewer samples. We show that the speed of HMC sample generation scales as $\mathcal{O}(N_{\mathrm{psr}}^{5/4})$ where N_{psr} is the number of pulsars, compared to $\mathcal{O}(N_{\mathrm{psr}}^{2})$ for MCMC methods. These factors offset the increased time required to generate a sample using HMC, demonstrating the value of adopting HMC techniques for PTAs.

DOI: 10.1103/PhysRevD.107.043013

I. INTRODUCTION

Pulsar timing arrays (PTAs) [1–3] seek to detect low-frequency gravitational waves (GWs) by looking for spatial correlations induced in the times of arrival (TOAs) pulses from millisecond pulsars. PTAs are most sensitive in the nanohertz frequency regime (∼1–100 nHz), where the dominant source of GWs is expected to be a stochastic gravitational wave background (GWB) originating from a cosmic population of supermassive black hole binaries (SMBHBs) [4–7]. The North American Nanohertz Observatory for Gravitational Waves (NANOGrav) [8] has been collecting pulsar TOA data since 2004. NANOGrav, along with the European Pulsar Timing Array [9], Parkes Pulsar Timing Array [10], and the Indian Pulsar Timing Array Project [11] form the International Pulsar Timing Array (IPTA) [12].

Detection of low-frequency GWs provides a valuable tool for studying parts of the dynamical universe not accessible through electromagnetic observations. Constraining the GWB shape and strength can provide useful constraints on properties of the SMBHB population including the black hole–host galaxy scaling relations [13,14] and the astrophysical environments of SMBHBs emitting GWs [15–19]. The GWB could also contain contributions from more speculative sources such as primordial GWs from inflation [20,21] and networks of cosmic strings [22,23].

GW signals can be extracted as a correlated signal from pulsar timing data only after subtracting the pulsar's timing model and accounting for underlying sources of noise in both the pulsar and observing instruments. These analyses are frequently done using Bayesian techniques [24–27], which we outline in Sec. II. In order to perform the Bayesian searches, NANOGrav makes use of the parallel-tempering Markov chain Monte Carlo (MCMC) code PTMCMCSAMPLER [28], which includes a variety of jump proposal schemes such as differential evolution, prior draws, and adaptive Metropolis.

MCMC methods work adequately for a large portion of statistical models, but simple MCMC algorithms such as random-walk Metropolis [29] or Gibbs sampling [30] become slow as the size and complexity of the model grow and take considerably longer to converge. Both of the aforementioned methods use random-walk proposals to generate samples and explore the parameter space, which tend to be increasingly inefficient when the target distribution includes correlations among the parameters [31]. Hamiltonian Monte Carlo (HMC) [31,32] removes the requirement to sample the model randomly, and replaces it with a simulation of Hamiltonian dynamics on the distribution itself. This scheme allows samples to be drawn at much further distances from one another, and explores the full parameter space in a more efficient way. For a target distribution of dimension d, the cost of drawing an independent sample with HMC goes roughly as $\mathcal{O}(d^{5/4})$, compared to $\mathcal{O}(d^2)$ for random-walk Metropolis [33]. The no-u-turn sampler (NUTS) [34] algorithm provides a basis for performing analysis with HMC without pretuning the sampling.

The HMC algorithm was initially developed for the problem of performing lattice field theory simulations of quantum chromodynamics [32]. The earliest approach applying HMC to PTA science was in the development of a model-independent method for performing Bayesian analyses on pulsar timing data [35]. The technique worked extremely well when applied to the IPTA Mock Data Challenge.[1] When applied to real data, however, the sampling could not fully explore the hierarchical model and became stuck in “Neal’s funnel” [36]. Applying data-aware coordinate transformations using the Cholesky decomposition helped deal with hierarchical funneling, and consequently there was a successful application of HMC to the targeted problem of outlier excision from PTA datasets [37]. The trade-off was that the additional transformations made sampling the hierarchical likelihood slower than the typical marginalized likelihood that was already used. As a result, HMC was not further explored in this context and has since remained largely underutilized towards the broad array of PTA science.

In this paper, we present a method for performing PTA GW searches using HMC as the underlying sampling algorithm. This represents the first attempt at applying HMC to the marginalized PTA likelihood, where we can avoid the funneling that plagues hierarchical models while still leveraging the benefits of HMC in exploring high-dimensional distributions. We test this method on the NANOGrav 11-year dataset [38], as well as realistic simulated data with similar red and white noise to the NANOGrav 11-year dataset. We demonstrate that performing a Bayesian GWB search with HMC results in a significant reduction in required sample generation to give equivalent results to current methods.

We also show that the additional gradient calculations necessary for HMC to operate scale roughly the same as the current likelihood evaluation with respect to the number of pulsars in a given dataset. Additionally we demonstrate that when comparing the time to generate independent samples, HMC outperforms traditional MCMC methods for PTA models of varying size in accordance with the expected scaling. This is a necessary consideration as the sizes of PTAs will continue to grow and with that the number of parameters needed to sample over.

This paper is organized as follows. In Sec. II, we describe the methods, signal models, and software used. In Sec. III we present the results of a GWB search using HMC, and compare the accuracy and efficiency of this method for both real and simulated PTA data. We conclude in Sec. IV and discuss how this method could be utilized for future PTA work.

II. METHODOLOGY AND SOFTWARE

In this section, we provide a brief outline of a typical PTA Bayesian GW search. We then give an overview of the HMC and NUTS algorithms, and discuss how to apply these methods to existing PTA work.

A. PTA signal model

We now discuss the PTA likelihood function. Following the outline provided in [27], we start by considering a single pulsar and its timing residual vector $\delta\mathbf{t}$ with length equal to the number of TOAs in our dataset, N_{TOA}. This timing residual data can be decomposed into individual components:

$$\delta\mathbf{t} = M\boldsymbol{\epsilon} + F\mathbf{a} + U\mathbf{j} + \mathbf{n}. \tag{1}$$

Each term describes a different inaccuracy or source of noise that contributes to the residual data. The term $M\boldsymbol{\epsilon}$ represents inaccuracies stemming from the subtraction of the pulsar’s timing model, with M the timing model design matrix, and $\boldsymbol{\epsilon}$ the vector of timing model parameter offsets. The effects due to low-frequency (“red”) noise are encoded in the term $F\mathbf{a}$. We choose to define this in a rank-reduced basis where F represents our matrix of basis functions, in this case alternating sine and cosine functions, and $\mathbf{a}$ represents a set of Fourier coefficients. The term $U\mathbf{j}$ describes noise that is completely uncorrelated in time but completely correlated across observations of a similar epoch. The matrix U maps between N_{TOA} residual data and N_{epoch} observation sessions, and $\mathbf{j}$ accounts for the correlated noise in each epoch. The final term, $\mathbf{n}$, includes any other high-frequency (“white”) noise that cannot be accounted for in the previous terms, such as radiometer noise.

Previous Bayesian analysis schemes [39–43] have described the white noise with EFAC (constant multiplier to TOA uncertainties) and EQUAD (white noise added in quadrature to EFAC) parameters and employed a power-law model to describe the red noise. The sum of these white noise covariances we describe via a matrix N. The parameters describing $\boldsymbol{\epsilon}$, $\mathbf{a}$, and $\mathbf{j}$ we group as follows:

$$T = [MFU], \qquad \mathbf{b} = \begin{bmatrix} \boldsymbol{\epsilon} \\ \mathbf{a} \\ \mathbf{j} \end{bmatrix}. \tag{2}$$

[1] The first IPTA Mock Data Challenge was developed by Fredrick Jenet, Kejia Lee, and Michael Keith and administered in 2012.

We place a Gaussian prior on these parameters with covariance:

$$B = \begin{bmatrix} \infty & 0 & 0 \\ 0 & \varphi & 0 \\ 0 & 0 & \mathcal{J} \end{bmatrix}, \tag{3}$$

where ∞ represents a diagonal matrix of infinities corresponding to unconstrained uniform priors on all timing model parameters. The parameters that describe $\mathcal{J}$ we refer to as ECORR and correspond to the epoch-correlated white noise signals per receiving back end. The matrix φ defines the parameters involving red noise signals, which includes low-frequency noise intrinsic to each pulsar, as well as the stochastic GWB. For this paper, we performed our analysis by modeling the GWB using a fiducial power-law spectrum of the characteristic GW strain h_c and cross-power spectral density S_{ab}:

$$h_c(f) = A_{\mathrm{gw}} \left(\frac{f}{f_{yr}} \right)^{\alpha}, \tag{4}$$

$$S_{ab}(f) = \Gamma_{ab} \frac{A_{\mathrm{gw}}^2}{12\pi^2} \left(\frac{f}{f_{\mathrm{yr}}} \right)^{-\gamma} f_{\mathrm{yr}}^{-3}, \tag{5}$$

where $\gamma = 3–2\alpha$. For a background generated by the GW emission from the evolution of a population of inspiraling SMBHBs in circular orbits, we have $\alpha = -2/3$, which implies $\gamma = 13/3$ [44]. The function Γ_{ab} is called the overlap reduction function (ORF) and describes the average correlations between any two pulsars a and b as a function of their angular separation. For an isotropic, stochastic GWB, this ORF is given by the Hellings-Downs correlation [45]:

$$\Gamma_{ab} = \frac{3}{2} x_{ab} \ln x_{ab} - \frac{x_{ab}}{4} + \frac{1}{2} + \frac{\delta_{ab}}{2}, \tag{6}$$

where $x_{ab} = (1 - \cos \xi_{ab})/2$ for two pulsars with angular separation ξ_{ab}.

We analytically marginalize over the timing model parameters to reduce the overall dimensionality of our posterior [35,46] and are left with the form of the likelihood that is used for the analysis in this paper:

$$\mathcal{P}(\delta\mathbf{t}|\phi) = \frac{\exp\left(-\frac{1}{2}\delta\mathbf{t}^T C^{-1} \delta\mathbf{t}\right)}{\sqrt{\det 2\pi C}}, \tag{7}$$

where $C = N + TBT^T$. We define ϕ as the set of all varying parameters in our model. We compute the likelihood and perform Bayesian searches using the NANOGrav package ENTERPRISE [47].

B. Hamiltonian Monte Carlo

We now provide a description of the HMC algorithm. In HMC [31,32], we start by introducing an auxiliary momentum variable p_i alongside each target parameter q_i. In most implementations, the momenta are chosen to be independent of the q_i and follow a zero-mean Gaussian distribution, with a covariance matrix M that is typically taken to be the identity. The log of the joint density of $\mathbf{p}$ and $\mathbf{q}$ defines our Hamiltonian:

$$H(\mathbf{p}, \mathbf{q}) = U(\mathbf{q}) + K(\mathbf{p}) = -\mathcal{L}(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T M^{-1} \mathbf{p}, \tag{8}$$

where $\mathcal{L}(\mathbf{q}) \equiv \log \mathcal{P}(\delta\mathbf{t}|\phi)$ is the log of the likelihood function for the distribution of our target parameters $\mathbf{q}$. Analogous to Hamiltonian dynamics, we have a potential energy term $U(\mathbf{q})$ and a kinetic energy term $K(\mathbf{p})$. We then simulate the evolution of this system over time according to Hamilton's equations:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \qquad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}. \tag{9}$$

This can be solved numerically using a symplectic integrator such as a "leapfrop" method, which for an integration step size ε uses an update scheme:

$$\mathbf{p}^{t+\varepsilon/2} = \mathbf{p}^t + \left(\frac{\varepsilon}{2}\right) \nabla_{\mathbf{q}} \mathcal{L}(\mathbf{q}^t), \tag{10a}$$

$$\mathbf{q}^{t+\varepsilon} = \mathbf{q}^t + \varepsilon \mathbf{p}^{t+\varepsilon/2}, \tag{10b}$$

$$\mathbf{p}^{t+\varepsilon} = \mathbf{p}^{t+\varepsilon/2} + \left(\frac{\varepsilon}{2}\right) \nabla_{\mathbf{q}} \mathcal{L}(\mathbf{q}^{t+\varepsilon}), \tag{10c}$$

where superscripts denote the time at which the particular quantity is evaluated. The standard method for producing a chain of samples using HMC then proceeds as follows: We first resample our momenta distribution. Then for a set number of leapfrog steps L, we use Eq. (10) to evolve our system through time and propose some final position and momentum vectors $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{p}}$. This proposal is accepted or rejected according to the Metropolis algorithm [29].

Mapping the path of the leapfrog integrator leads to a useful sanity check of HMC: trajectory divergences. These divergences occur when the trajectory taken via Hamiltonian simulation departs from the true trajectory, and risk biasing estimates or reducing HMC to random-walk behavior [48]. By tracking the trajectories and alerting the user of large divergences, HMC offers another diagnostic to detect unsuitably parametrized models that is not possible with Metropolis-Hastings (MH) MCMC methods.

There are limitations to HMC and the models under which it can be used properly. Due to its origins in Hamiltonian dynamics, HMC can only operate in continuous state spaces and contains no internal recourse to deal with discrete variables. In such cases, the discrete variables can be handled with separate algorithms such as Gibbs sampling [30]. HMC also requires that the log density of the target distribution is differentiable almost everywhere with respect to the model parameters, with the exception coming at points of probability 0 [31]. Additionally, HMC struggles when there is strong multimodality in the target distribution due to the modes being separated by regions of very low probability [49]. The PTA models used in this paper satisfy the above conditions, and HMC remains a valid choice of underlying sampling algorithm.

C. No-u-turn sampler

The performance of the HMC algorithm is particularly sensitive to two user-defined parameters: the number of leapfrog steps L and integration step size ϵ, defined in the above section. If these parameters are not properly tuned, the algorithm may waste computation time or begin to exhibit unwanted random walk behavior and in some cases may not even be ergodic [31]. In general, tuning these parameters appropriately would require multiple preliminary runs.

The no-u-turn sampler (NUTS) [34] offers an extension to the HMC algorithm that dynamically tunes the number of leapfrog steps L. NUTS uses a recursive doubling algorithm, similar to the one outlined in [36], to determine when the generated proposal trajectory begins to double back on itself, or make a “u turn.” The algorithm builds a binary tree, simulating Hamiltonian dynamics forwards and backwards randomly in time for 2^j steps, with j the height of the full tree. If we define $\mathbf{q}^+$, $\mathbf{p}^+$ and $\mathbf{q}^-$, $\mathbf{p}^-$ as the position-momenta pairs of the left- and rightmost nodes of the bottom subtree, then the stopping condition for NUTS can be written as

$$(\mathbf{q}^+ - \mathbf{q}^-)\cdot\mathbf{p}^- < 0 \quad \text{or} \quad (\mathbf{q}^+ - \mathbf{q}^-)\cdot\mathbf{p}^+ < 0. \tag{11}$$

The above procedure adaptively tunes the parameter L for each iteration in the chain. The step size parameter ϵ in NUTS is set using the method of stochastic optimization with varying adaptation [50]. In particular, Hoffman and Gelman utilize the primal-dual averaging algorithm proposed by [51]. With L and ϵ automatically tuned, NUTS can be run without any human intervention.

D. Coordinate transformations and software

Previous approaches to pulsar timing analyses with HMC utilized a hierarchical PTA likelihood. Initially these methods did not include coordinate transformations on the data, and as a result became stuck with hierarchical funneling. This funneling originates from the fact that within hierarchical models, random variables are very highly correlated when the data are sparse [52]. One can reduce the correlations between the random variables, and hence the funneling, by adopting a noncentered reparametrization of the data [53]. In regards to the hierarchical PTA likelihood, such a reparametrization using the Cholesky decomposition allowed HMC sampling to proceed but at the cost of slowing down the likelihood.

In this paper we are focused entirely on the marginalized PTA likelihood and can therefore leave behind the coordinate transformations designed for hierarchical models. We do employ a set of transformations designed to improve the performance of the NUTS algorithm. First we perform an interval transform, moving all parameters with bounded priors from their interval $[a, b]$ to the whole real line. We then whiten the data using Cholesky whitening to move to a set of transformed variables whose covariance matrix is the identity. This is accomplished through the Hessian calculated around the maximum *a posteriori* parameter vector. Neither transformation considerably alters the likelihood computation speed.

When determining the speed and efficiency of the HMC and NUTS pipeline, one must depend almost entirely on the ability to calculate gradients of the likelihood and do so as quickly as possible. Numerical derivatives are comparably easier to write but slow in practice and prone to errors from approximations. By-hand analytic derivatives are fast but difficult to write into concise code for all but the simplest of models. An excellent solution for arbitrary likelihood functions and their gradients is the package JAX [54], which leverages both automatic differentiation and just-in-time compilation to efficiently differentiate native PYTHON code and turn an otherwise slow gradient function into incredibly fast machine executables. The use of this technique is rather new, with JAX only recently becoming a mature code base, and consequently this marks the first time JAX and automatic differentiation have been utilized for HMC sampling of PTA data.

Summarizing the software used for the analyses to follow in this paper, the signal models and likelihood used in our analyses come from NANOGrav’s flagship PTA analysis suite ENTERPRISE [47]. We utilize the automatic differentiation capabilities in JAX [54] to calculate the likelihood derivatives required for HMC to operate. We perform two coordinate transformations on our data to better interface with the NUTS algorithm. Lastly, for the sampling we use a custom-built NUTS code that is freely and openly available in PICCARD.[2] The combination of these three codes leads to an end-to-end pipeline for performing PTA analyses with HMC sampling.

III. RESULTS

In this section, we study the HMC sampling method both in its ability to accurately perform Bayesian searches for a stochastic GWB using PTA data, as well the efficiency of

[2] https://www.github.com/vhaasteren/piccard.

such a method when compared against the existing techniques employed by NANOGrav.

The GWB model that is analyzed in this paper arises from a PTA consisting of data from 45 pulsars. The parameters encompassing the signal model closely mimic those outlined in Sec. II A. We fix white noise parameters to their maximum likelihood values as obtained from individual pulsar noise runs. We model pulsar-intrinsic red noise with a power-law power spectral density (PSD) containing two search parameters $\log_{10} A_{\rm red} \in U[-18, -11]$ and $\gamma_{\rm red} \in U[0, 7]$. We model the GWB as a power-law PSD process that is common amongst all the pulsars. The corresponding parameters are an amplitude with log-uniform prior $A_{\rm CP} \in U[-18, -12]$ and a spectral index $\gamma_{\rm CP}$ that we fix to 13/3. We do not include spatial correlations in our GWB model. This results in a total of $2N_{\rm psr} + 1$ varying parameters in the model.

We also generate a set of simulated PTA datasets using LIBSTEMPO [55]. We inject both per-pulsar white and red noise parameters at their maximum likelihood values. The injected values again originate from individual pulsar noise runs, where all parameters for a given pulsar are allowed to vary. Again we include a common process signal representing the GWB with both a fixed amplitude and spectral index at $\log_{10} A_{\rm CP} = -14.7$ and $\gamma_{\rm CP} = 13/3$, and do not include interpulsar spatial correlations. We repeat the above procedure for 100 realizations of the GWB which results in a collection of 100 realistic simulated PTA datasets. When analyzing the simulated data, we use a similar signal model to the one described above but this time allow the common-process spectral index to vary as $\gamma_{\rm CP} \in U[0, 7]$.

Runs conducted with the MH MCMC algorithm use the PTMCMCSAMPLER [28] code. The sampler is set up in similar fashion to the NANOGrav 11-year GWB search [56]. We include adaptive Metropolis and differential evolution jump proposals. For all varying parameters present in the model, we also add prior draw jump proposals. We do not utilize parallel tempering in this work.

A. NANOGrav 11-year data comparison

We perform a stochastic GWB search with both the HMC and MH MCMC algorithms on the NANOGrav 11-year dataset [38]. This dataset encompasses the timing data for 45 millisecond pulsars. Figure 1 shows the posterior distributions for the background amplitude $A_{\rm CP}$ calculated using both Monte Carlo methods. We calculate 95% upper limits on $A_{\rm CP}$ and estimate uncertainties with bootstrap methods [57]. The HMC algorithm produces results that are consistent with the base MH MCMC search, with corresponding 95% upper limits $A_{\rm CP,HMC} < 1.72(4) \times 10^{-15}$ and $A_{\rm CP,MH\ MCMC} < 1.74(3) \times 10^{-15}$.

The MH MCMC sampling routine was run for a total number of samples $M_{\rm MH\ MCMC} = 1{,}000{,}000$, whereas the HMC routine was run for $M_{\rm HMC} = 8{,}000$. The wall time for the MH run was approximately 4 hours, compared to just under 4 hours for the HMC run. Both sets of chains are checked for convergence using the Gelman-Rubin R-hat convergence test [34]. It is worth reinforcing that the benefit of generating fewer samples is partially outweighed by the increased computational cost of proposing a new HMC sample. We explore the scaling of sample generation time in Sec. III C.

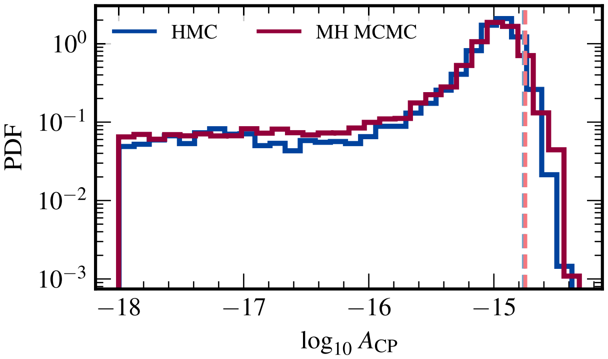

FIG. 1. Posterior probability distributions for the amplitude $\log_{10} A_{\rm CP}$ of a common-process signal run using either MH MCMC or HMC as the primary sampling method, computed using the NANOGrav 11-year dataset. The common-process amplitude parameter is set with a log-uniform prior, the common-process spectral index is fixed at 13/3, and no spatial correlations are included. Vertical lines represent 95% upper limits calculated for posteriors generated using HMC [blue; $A_{\rm CP,HMC} < 1.72(4) \times 10^{-15}$] and MH MCMC [red; $A_{\rm CP,HMC} < 1.74(3) \times 10^{-15}$], though the two lines will be difficult to individually resolve due to the similarity in upper limits. We conclude that the two procedures produce consistent posteriors when applied to identical models.

We also perform a direct comparison to the upper limit calculated in the NANOGrav 11-year GWB search [56]. In order to do such a comparison, we alter our signal model slightly to match that of the 11-year analysis and adjust the common-process amplitude from a log uniform to a uniform prior $A_{\rm CP} \in [10^{-18}, 10^{-12}]$. Performing this analysis with the HMC pipeline, again with $M = 8,\ 000$ samples, recovers a 95% upper limit of $A_{\rm CP,HMC} < 1.64(3) \times 10^{-15}$. This is in relative agreement with the result in [56] of $A_{\rm CP} < 1.61(2) \times 10^{-15}$ for a similar model with identical Jet Propulsion Laboratory (JPL) ephemeride DE436.

We further compare the efficiency of HMC sampling by looking at the autocorrelation lengths of the two sets of chains, measuring how far one must jump through the chain to find the next statistically significant sample. The autocorrelation lengths are calculated per parameter in the model. This was calculated for each set of chains generated with the two Monte Carlo sampling methods, and the results are shown in Fig. 2. We find that the HMC chains have autocorrelation lengths between 1 and 2 orders of magnitude smaller than those of identical parameters in the MH MCMC chains. This behavior is expected, as the HMC algorithm is designed to take larger, more-informed

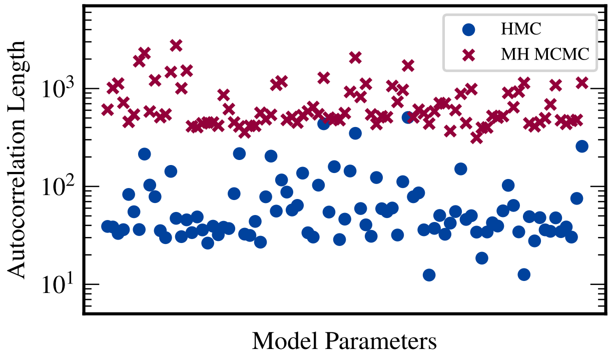

FIG. 2. Autocorrelation lengths for 91 parameters ($2N_{\mathrm{psr}}$ individual pulsar red-noise parameters and a common process signal parametrized with an amplitude A_{CP} and spectral index $\gamma_{\mathrm{CP}} = 13/3$) present in a standard GWB model. The autocorrelation lengths are calculated from two sets of chains generated from sampling this model: one sampled with HMC (blue) and one with MH MCMC (red). Each mark represents the approximate number of steps one must jump through that particular parameter's chain to reach an independent sample.

steps to avoid random walklike behavior and produce a higher ratio of independent samples.

B. Simulated data and parameter recovery

We also aim to test that the HMC algorithm behaves similarly to the standard MH MCMC technique when considering statistical coverage of a standard PTA model. To determine the capability of the sampling methods to accurately recover injected parameters, we consider 100 simulated PTA datasets and seek to verify if in $p\%$ of the realizations the injected parameter values fall within the $p\%$ credible region of the posteriors. We run standard Bayesian searches on all realizations using both sampling methods.

The results of the parameter recovery test described above are summarized in Fig. 3, with a particular focus on the two parameters describing the GWB. The HMC sampler recovers the injected GWB parameter values with the same consistency as the traditional analysis. Neither method recovers the injected parameters exactly, and therefore no line in Fig. 3 falls directly on the vertical line at $x = 0$. This is due to an inherent model mismatch present when simulating data with LIBSTEMPO and recovering the posteriors separately with ENTERPRISE. The simulated GWB is generated with more frequencies than is searched over during the analysis, leading to a natural bias in recovery.[3]

C. Scaling of gradient computation speed

The time per HMC sample generation is dominated by the time to calculate the gradient of the log-likelihood necessary for leapfrog integration. The evaluation time for the base likelihood calculation present in ENTERPRISE is calculated by averaging the evaluation time for 50 calls of the log-likelihood function. We first use a PTA with only a single pulsar, and repeat the above step adding one additional pulsar at a time up to $N_{\mathrm{psr}} = 45$. This produces an idea of how the base likelihood evaluation time, and by extension the MCMC sample generation time, scales with the number of pulsars present in a PTA (Fig. 4: dashed red line).

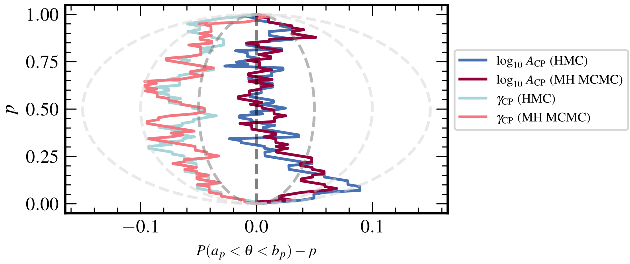

FIG. 3. *p*-*p* comparison of GWB parameter recovery for both the HMC and MH MCMC sampling methods operating on simulated PTA data. The x axis shows the difference between the fraction of realizations with which the injected values fall within the $p\%$ credible region of the posteriors and the $p\%$ credible region on the y axis. The vertical dark gray line at $x = 0$ represents a perfect recovery of the injected parameter values. The light gray lines represent 1σ, 2σ, and 3σ deviations.

In order to accurately scale the computation time necessary to draw a sample with NUTS, we must account for the dynamic tuning of the HMC hyperparameter L and note that we likely require multiple evaluations of the log likelihood and gradient to generate a sample. First we

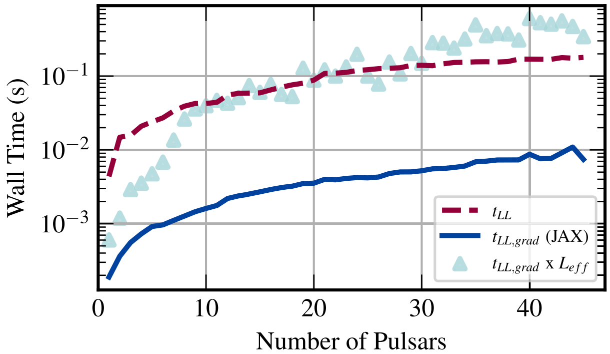

FIG. 4. Wall time for calculating implementations of both the log of the PTA likelihood as well as its gradient, scaled by the number of pulsars present in a given model. The red dashed line represents the log-likelihood evaluation as present in the standard PTA analysis suite ENTERPRISE. The solid blue line shows the evaluation of the log likelihood and gradient function after being precompiled with JAX. The cyan triangles denote the evaluation times present in the blue line multiplied by a value L_{eff} representing the effective number of gradient evaluations required to generate a new HMC sample.

[3] For further details, see documentation for GWB simulation in the TOASIM module of LIBSTEMPO.

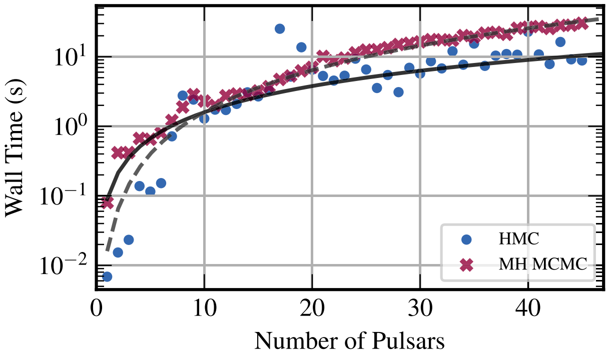

FIG. 5. Wall time to produce an independent sample in Markov chains generated using HMC and MH MCMC methods, scaled by the number of pulsars $N_{\rm psr}$ present in the model. The total number of parameters in a given model is $d = 2N_{\rm psr} + 1$. The solid black represents the expected scaling for HMC of $\mathcal{O}(d^{5/4})$. The dashed gray line denotes the expected scaling for MH MCMC of $\mathcal{O}(d^2)$.

consider the evaluation time of the log likelihood and gradient function compiled with JAX, and scale per pulsar following the same procedure defined above (Fig. 4: solid blue line). We then take the 45 separate PTA objects and run standard GWB analyses, with models defined in Sec. III, through the HMC pipeline for $M = 10,\ 000$ samples. The height j of the NUTS binary tree defines a total of $L = 2^j + 1$ gradient evaluations per new sample. By averaging this over the full run, we can approximate an $L_{\rm eff}$ and more accurately scale the time per HMC sample generation (Fig. 4: cyan triangles).

Finally, we look at the time to generate independent samples in our chain and how it scales with increasing PTA size. This is ultimately the most important metric for testing the efficiency of HMC as independent samples and thinned Markov chains are what inevitably drive the statistical inferences made on the data. Independent samples in this context are defined here as samples that are separated by one autocorrelation length.

We take the 45 PTA objects of increasing $N_{\rm psr}$ and generate Markov chains with HMC and MH MCMC of size $M = 8{,}000$ and $M = 1{,}000{,}000$, respectively. Taking the median autocorrelation length of each chain and the base sampling speed calculated previously, we create a scaling of the wall time for both sampling methods in making independent samples for PTA models of increasing size. The results are summarized in Fig. 5. It shows that in the long run HMC will outperform MH MCMC techniques in making statistically relevant samples, despite the likely increase in upfront computational cost.

IV. CONCLUSIONS

In this paper, we have implemented an efficient method for sampling the high dimensional distributions present in PTA GW Bayesian searches using the HMC algorithm. This method leverages a hybrid technique comprised of parts from both traditional stochastic Monte Carlo schemes as well as deterministic sampling methods derived from Hamiltonian dynamics. We show that utilizing HMC results in a reduction of approximately 2 orders of magnitude in the number of samples drawn to produce equivalent results to the existing Bayesian searches performed on PTA datasets.

The efficiency of this technique is largely defined by the speed at which derivatives of the log likelihood can be computed for the purpose of simulating Hamiltonian dynamics. We have shown that the current implementation of this calculation scales similarly to the present log-likelihood calculation with respect to the number of pulsars in a dataset, and improves upon traditional MCMC methods when comparing the production of independent samples of the distribution. This improvement in performance scaling is paramount because PTAs will continuously grow and add more pulsars to their data collection. The 11-year dataset featured in this paper contains 45 pulsars. Future NANOGrav datasets will have ≥ 60 pulsars and future IPTA datasets may contain close to $\sim$100 pulsars. Increasing the data volume will further strain our computational capabilities to perform large parameter GW searches. HMC provides a way of resolving these limitations in a way that is more favorable to future PTA analyses.

It is worth emphasizing that the $\mathcal{O}(d^{5/4})$ scaling of HMC is not just with respect to the number of pulsars, but with respect to the total number of parameters. We analyzed a model with $2N_{\rm psr} + 1$ parameters (with the GWB spectral index held fixed), but this represents only one of many different approaches for GW searches in PTAs. For example, one can parametrize the GWB with a free spectrum model, increasing its number of parameters from 2 to 30. Likewise, one can parametrize the individual pulsar red noise in a similar fashion, increasing the parameter count from 2 to 30 *per pulsar*. The favorable scaling of HMC opens the door for more flexible models that are currently prohibitive with current MH MCMC runs.

Currently we have only applied the HMC algorithm to the problem of sampling a stochastic GWB model. PTAs are also sensitive to certain deterministic GW signals, and work towards tailoring this method to such searches is under development. This technique is particularly promising for searches for GWs from individual SMBHBs because of the large number of parameters necessary to describe the GW signal ($2N_{\rm psr} + 8$ for a circular binary, more if the source is eccentric). In general, this technique can be adapted to the full suite of PTA searches, provided the underlying models adhere to the limitations outlined in Sec. II B. The ultimate goal is a general purpose pipeline for performing any such PTA analysis that leverages the benefits of the HMC algorithm towards exploring complicated, high-dimensional models.

ACKNOWLEDGMENTS

We thank Michele Vallisneri for useful discussions. We also thank Paul Baker for valuable comments. Lastly we thank the anonymous referees for their helpful comments and suggestions that improved the manuscript. This work was supported by National Science Foundation (NSF) Grant No. PHY-2011772. The authors are members of the North American Nanohertz Observatory for Gravitational Waves (NANOGrav) collaboration, which receives support from NSF Physics Frontiers Center Grants No. 1430284 and No. 2020265. G. E. F. is supported by National Aeronautics and Space Administration (NASA) Future Investigators in NASA Earth and Space Science and Technology Grant No. 80NSSC22K1591. A. D. J. and S. J. V. were supported by University of Wisconsin-Milwaukee (UWM) Discovery and Innovation Grant No. 101X410. A. D. J. acknowledges support from the Caltech and Jet Propulsion Laboratory President's and Director's Fund. This material is based upon work supported by NASA under Grant No. RFP22_5-0 issued through the Wisconsin Space Grant Consortium and the National Space Grant College and Fellowship Program.

[1] M. V. Sazhin, Sov. Astron. **22**, 36 (1978).

[2] S. Detweiler, Astrophys. J. **234**, 1100 (1979).

[3] R. S. Foster and D. C. Backer, Astrophys. J. **361**, 300 (1990).

[4] A. Sesana, F. Haardt, P. Madau, and M. Volonteri, Astrophys. J. **611**, 623 (2004).

[5] A. Sesana, F. Haardt, P. Madau, and M. Volonteri, Astrophys. J. **623**, 23 (2005).

[6] P. A. Rosado, A. Sesana, and J. Gair, Mon. Not. R. Astron. Soc. **451**, 2417 (2015).

[7] S. Burke-Spolaor, S. R. Taylor, M. Charisi, T. Dolch, J. S. Hazboun, A. M. Holgado, L. Z. Kelley, T. J. W. Lazio, D. R. Madison, N. McMann *et al.*, Astron. Astrophys. Rev. **27**, 5 (2019).

[8] S. Ransom, A. Brazier, S. Chatterjee, T. Cohen, J. M. Cordes, M. E. DeCesar, P. B. Demorest, J. S. Hazboun, M. T. Lam, R. S. Lynch *et al.*, Bull. Am. Astron. Soc. **51**, 195 (2019).

[9] G. Desvignes, R. N. Caballero, L. Lentati, J. P. W. Verbiest, D. J. Champion, B. W. Stappers, G. H. Janssen, P. Lazarus, S. Osłowski, S. Babak *et al.*, Mon. Not. R. Astron. Soc. **458**, 3341 (2016).

[10] M. Kerr, D. J. Reardon, G. Hobbs, R. M. Shannon, R. N. Manchester, S. Dai, C. J. Russell, S. Zhang, W. van Straten, S. Osłowski *et al.*, Pub. Astron. Soc. Aust. **37**, e020 (2020).

[11] P. Tarafdar, K. Nobleson, P. Rana, J. Singha, M. A. Krishnakumar, B. C. Joshi, A. K. Paladi, N. Kolhe, N. Dhanda Batra, N. Agarwal *et al.*, Pub. Astron. Soc. Aust. **39**, e053 (2022).

[12] B. B. P. Perera, M. E. DeCesar, P. B. Demorest, M. Kerr, L. Lentati, D. J. Nice, S. Osłowski, S. M. Ransom, M. J. Keith, Z. Arzoumanian *et al.*, Mon. Not. R. Astron. Soc. **490**, 4666 (2019).

[13] V. Ravi, J. S. B. Wyithe, R. M. Shannon, and G. Hobbs, Mon. Not. R. Astron. Soc. **447**, 2772 (2015).

[14] A. Sesana, Mon. Not. R. Astron. Soc. **433**, L1 (2013).

[15] G. D. Quinlan, New Astron. **1**, 35 (1996).

[16] A. Sesana, F. Haardt, and P. Madau, Astrophys. J. **651**, 392 (2006).

[17] Z. Haiman, B. Kocsis, and K. Menou, Astrophys. J. **700**, 1952 (2009).

[18] B. Kocsis and A. Sesana, Mon. Not. R. Astron. Soc. **411**, 1467 (2011).

[19] L. Sampson, N. J. Cornish, and S. T. McWilliams, Phys. Rev. D **91**, 084055 (2015).

[20] L. P. Grishchuk, Sov. J. Exp. Theor. Phys. Lett. **23**, 293 (1976).

[21] P. D. Lasky, C. M. F. Mingarelli, T. L. Smith, J. T. Giblin, E. Thrane, D. J. Reardon, R. Caldwell, M. Bailes, N. D. R. Bhat, S. Burke-Spolaor *et al.*, Phys. Rev. X **6**, 011035 (2016).

[22] X. Siemens, V. Mandic, and J. Creighton, Phys. Rev. Lett. **98**, 111101 (2007).

[23] J. J. Blanco-Pillado, K. D. Olum, and B. Shlaer, Phys. Rev. D **89**, 023512 (2014).

[24] R. van Haasteren, Y. Levin, P. McDonald, and T. Lu, Mon. Not. R. Astron. Soc. **395**, 1005 (2009).

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

[26] L. Lentati, P. Alexander, M. P. Hobson, S. Taylor, J. Gair, S. T. Balan, and R. van Haasteren, Phys. Rev. D **87**, 104021 (2013).

[27] Z. Arzoumanian, A. Brazier, S. Burke-Spolaor, S. J. Chamberlin, S. Chatterjee, B. Christy, J. M. Cordes, N. J. Cornish, K. Crowter, P. B. Demorest *et al.*, Astrophys. J. **821**, 13 (2016).

[28] J. Ellis and R. van Haasteren, jellis18/ptmcmcsampler: Official release (2017), 10.5281/zenodo.1037579.

[29] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[30] S. Geman and D. Geman, IEEE Trans. Pattern Anal. Mach. Intell. **6**, 721 (1984).

[31] R. Neal, in *Handbook of Markov Chain Monte Carlo* (Chapman & Hall/CRC, Boca Raton, FL, 2011), pp. 113–162.

[32] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).

[33] M. Creutz, Phys. Rev. D **38**, 1228 (1988).

[34] M. D. Hoffman and A. Gelman, arXiv:1111.4246.

[35] L. Lentati, P. Alexander, M. P. Hobson, S. Taylor, J. Gair, S. T. Balan, and R. van Haasteren, Phys. Rev. D **87**, 104021 (2013).

[36] R. M. Neal, Ann. Stat. **31**, 705 (2003).

[37] M. Vallisneri and R. van Haasteren, Mon. Not. R. Astron. Soc. **466**, 4954 (2017).

[38] Z. Arzoumanian, A. Brazier, S. Burke-Spolaor, S. Chamberlin, S. Chatterjee, B. Christy, J. M. Cordes, N. J. Cornish, F. Crawford, H. Thankful Cromartie *et al.*, Astrophys. J. Suppl. Ser. **235**, 37 (2018).

[39] R. van Haasteren, Y. Levin, P. McDonald, and T. Lu, Mon. Not. R. Astron. Soc. **395**, 1005 (2009).

[40] R. van Haasteren and Y. Levin, Mon. Not. R. Astron. Soc. **401**, 2372 (2010).

[41] R. van Haasteren, Y. Levin, G. H. Janssen, K. Lazaridis, M. Kramer, B. W. Stappers, G. Desvignes, M. B. Purver, A. G. Lyne, R. D. Ferdman *et al.*, Mon. Not. R. Astron. Soc. **414**, 3117 (2011).

[42] J. A. Ellis, Classical Quantum Gravity **30**, 224004 (2013).

[43] J. A. Ellis, X. Siemens, and R. van Haasteren, Astrophys. J. **769**, 63 (2013).

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

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

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

[47] J. A. Ellis, M. Vallisneri, S. R. Taylor, and P. T. Baker, ENTERPRISE: Enhanced Numerical Toolbox Enabling a Robust PulsaR Inference SuitE (Zenodo, 2020), 10.5281/zenodo.4059815.

[48] M. Betancourt, arXiv:1604.00695.

[49] C. Sminchisescu and M. Welling, Pattern Recognit. **44**, 2738 (2011).

[50] C. Andrieu and J. A. Thoms, Stat. Comput. **18**, 343 (2008).

[51] Y. Nesterov, Math. Program. **120**, 221 (2009).

[52] M. J. Betancourt and M. Girolami, arXiv:1312.0906.

[53] O. Papaspiliopoulos, G. Roberts, and M. Sköld, Stat. Sci. **22** (2007).

[54] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne *et al.*, JAX: Composable Transformations of Python + NumPy Programs (2018), http://github.com/google/jax.

[55] M. Vallisneri, Astrophysics Source Code Library p. ascl:2002.017 (2020), https://ui.adsabs.harvard.edu/abs/2020ascl.soft02017V.

[56] Z. Arzoumanian, P. T. Baker, A. Brazier, S. Burke-Spolaor, S. J. Chamberlin, S. Chatterjee, B. Christy, J. M. Cordes, N. J. Cornish, F. Crawford *et al.*, Astrophys. J. **859**, 47 (2018).

[57] B. Efron, Ann. Stat. **7**, 1 (1979).