NANOGrav 15-year gravitational-wave background methods

Aaron D. Johnson,[1,2] Patrick M. Meyers,[2] Paul T. Baker,[3] Neil J. Cornish,[4] Jeffrey S. Hazboun,[5] Tyson B. Littenberg,[6] Joseph D. Romano,[7] Stephen R. Taylor,[8] Michele Vallisneri,[9,2] Sarah J. Vigeland,[1] Ken D. Olum,[10] Xavier Siemens,[5,1] Justin A. Ellis,[11,12,13,†] Rutger van Haasteren,[14] Sophie Hourihane,[2] Gabriella Agazie,[1] Akash Anumarlapudi,[1] Anne M. Archibald,[15] Zaven Arzoumanian,[16] Laura Blecha,[17] Adam Brazier,[18,19] Paul R. Brook,[20] Sarah Burke-Spolaor,[21,22] Bence Bécsy,[5] J. Andrew Casey-Clyde,[23] Maria Charisi,[8] Shami Chatterjee,[18] Katerina Chatziioannou,[2] Tyler Cohen,[24] James M. Cordes,[18] Fronefield Crawford,[25] H. Thankful Cromartie,[18] Kathryn Crowter,[26] Megan E. DeCesar,[27] Paul B. Demorest,[28] Timothy Dolch,[29,30] Brendan Drachler,[31,32] Elizabeth C. Ferrara,[11,12,13,†] William Fiore,[21,22] Emmanuel Fonseca,[21,22] Gabriel E. Freedman,[1] Nate Garver-Daniels,[21,22] Peter A. Gentile,[21,22] Joseph Glaser,[21,22] Deborah C. Good,[23,33] Kayhan Gültekin,[34] Ross J. Jennings,[21,22] Megan L. Jones,[1] Andrew R. Kaiser,[21,22] David L. Kaplan,[1] Luke Zoltan Kelley,[35] Matthew Kerr,[36] Joey S. Key,[37] Nima Laal,[5] Michael T. Lam,[38,31,32] William G. Lamb,[8] T. Joseph W. Lazio,[9] Natalia Lewandowska,[39] Tingting Liu,[21,22] Duncan R. Lorimer,[21,22] Jing Luo,[40,*] Ryan S. Lynch,[41] Chung-Pei Ma,[35,42] Dustin R. Madison,[43] Alexander McEwen,[1] James W. McKee,[44,45] Maura A. McLaughlin,[21,22] Natasha McMann,[8] Bradley W. Meyers,[26,46] Chiara M. F. Mingarelli,[33,23,47] Andrea Mitridate,[48] Cherry Ng,[49] David J. Nice,[50] Stella Koch Ocker,[18] Timothy T. Pennucci,[51] Benetge B. P. Perera,[52] Nihan S. Pol,[8] Henri A. Radovan,[53] Scott M. Ransom,[54] Paul S. Ray,[36] Shashwat C. Sardesai,[1] Carl Schmiedekamp,[55] Ann Schmiedekamp,[55] Kai Schmitz,[56] Brent J. Shapiro-Albert,[21,22,57] Joseph Simon,[58] Magdalena S. Siwek,[59] Ingrid H. Stairs,[26] Daniel R. Stinebring,[60] Kevin Stovall,[28] Abhimanyu Susobhanan,[1] Joseph K. Swiggum,[50] Jacob E. Turner,[21,22] Caner Unal,[61,62] Haley M. Wahl,[21,22] Caitlin A. Witt,[63,64] and Olivia Young[31,32]

(NANOGrav Collaboration)

[1]*Center for Gravitation, Cosmology and Astrophysics, Department of Physics, University of Wisconsin-Milwaukee, P.O. Box 413, Milwaukee, Wisconsin 53201, USA*
[2]*Division of Physics, Mathematics, and Astronomy, California Institute of Technology, Pasadena, California 91125, USA*
[3]*Department of Physics and Astronomy, Widener University, One University Place, Chester, Pennsylvania 19013, USA*
[4]*Department of Physics, Montana State University, Bozeman, Montana 59717, USA*
[5]*Department of Physics, Oregon State University, Corvallis, Oregon 97331, USA*
[6]*NASA Marshall Space Flight Center, Huntsville, Alabama 35812, USA*
[7]*Department of Physics, Texas Tech University, Box 41051, Lubbock, Texas 79409, USA*
[8]*Department of Physics and Astronomy, Vanderbilt University, 2301 Vanderbilt Place, Nashville, Tennessee 37235, USA*
[9]*Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109, USA*
[10]*Institute of Cosmology, Department of Physics and Astronomy, Tufts University, Medford, Massachusetts 02155, USA*
[11]*Department of Astronomy, University of Maryland, College Park, Maryland 20742*
[12]*Center for Research and Exploration in Space Science and Technology, NASA/GSFC, Greenbelt, Maryland 20771*
[13]*NASA Goddard Space Flight Center, Greenbelt, Maryland 20771, USA*
[14]*Max-Planck-Institut für Gravitationsphysik (Albert-Einstein-Institut), Callinstrasse 38, D-30167, Hannover, Germany*
[15]*Newcastle University, NE1 7RU, Newcastle, United Kingdom*
[16]*X-Ray Astrophysics Laboratory, NASA Goddard Space Flight Center, Code 662, Greenbelt, Maryland 20771, USA*
[17]*Physics Department, University of Florida, Gainesville, Florida 32611, USA*
[18]*Cornell Center for Astrophysics and Planetary Science and Department of Astronomy, Cornell University, Ithaca, New York 14853, USA*
[19]*Cornell Center for Advanced Computing, Cornell University, Ithaca, New York 14853, USA*
[20]*Institute for Gravitational Wave Astronomy and School of Physics and Astronomy, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom*

[21]*Department of Physics and Astronomy, West Virginia University, P.O. Box 6315, Morgantown, West Virginia 26506, USA*
[22]*Center for Gravitational Waves and Cosmology, West Virginia University, Chestnut Ridge Research Building, Morgantown, West Virginia 26505, USA*
[23]*Department of Physics, University of Connecticut, 196 Auditorium Road, U-3046, Storrs, Connecticut 06269-3046, USA*
[24]*Department of Physics, New Mexico Institute of Mining and Technology, 801 Leroy Place, Socorro, New Mexico 87801, USA*
[25]*Department of Physics and Astronomy, Franklin & Marshall College, P.O. Box 3003, Lancaster, Pennsylvania 17604, USA*
[26]*Department of Physics and Astronomy, University of British Columbia, 6224 Agricultural Road, Vancouver, British Columbia V6T 1Z1, Canada*
[27]*George Mason University, resident at the Naval Research Laboratory, Washington, DC 20375, USA*
[28]*National Radio Astronomy Observatory, 1003 Lopezville Road, Socorro, New Mexico 87801, USA*
[29]*Department of Physics, Hillsdale College, 33 East College Street, Hillsdale, Michigan 49242, USA*
[30]*Eureka Scientific, 2452 Delmer Street, Suite 100, Oakland, California 94602-3017, USA*
[31]*School of Physics and Astronomy, Rochester Institute of Technology, Rochester, New York 14623, USA*
[32]*Laboratory for Multiwavelength Astrophysics, Rochester Institute of Technology, Rochester, New York 14623, USA*
[33]*Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, New York 10010, USA*
[34]*Department of Astronomy and Astrophysics, University of Michigan, Ann Arbor, Michigan 48109, USA*
[35]*Department of Astronomy, University of California, Berkeley, 501 Campbell Hall #3411, Berkeley, California 94720, USA*
[36]*Space Science Division, Naval Research Laboratory, Washington, DC 20375-5352, USA*
[37]*University of Washington Bothell, 18115 Campus Way NE, Bothell, Washington 98011, USA*
[38]*SETI Institute, 339 North Bernardo Avenue Suite 200, Mountain View, California 94043, USA*
[39]*Department of Physics, State University of New York at Oswego, Oswego, New York, 13126, USA*
[40]*Department of Astronomy and Astrophysics, University of Toronto, 50 Saint George Street, Toronto, Ontario M5S 3H4, Canada*
[41]*Green Bank Observatory, P.O. Box 2, Green Bank, West Virginia 24944, USA*
[42]*Department of Physics, University of California, Berkeley, California 94720, USA*
[43]*Department of Physics, University of the Pacific, 3601 Pacific Avenue, Stockton, California 95211, USA*
[44]*E.A. Milne Centre for Astrophysics, University of Hull, Cottingham Road, Kingston-upon-Hull, HU6 7RX, United Kingdom*
[45]*Centre of Excellence for Data Science, Artificial Intelligence and Modelling (DAIM), University of Hull, Cottingham Road, Kingston-upon-Hull, HU6 7RX, Unite Kingdom*
[46]*International Centre for Radio Astronomy Research, Curtin University, Bentley, WA 6102, Australia*
[47]*Department of Physics, Yale University, New Haven, Connecticut 06520, USA*
[48]*Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany*
[49]*Dunlap Institute for Astronomy and Astrophysics, University of Toronto, 50 St. George Street, Toronto, Ontario M5S 3H4, Canada*
[50]*Department of Physics, Lafayette College, Easton, Pennsylvania 18042, USA*
[51]*Institute of Physics and Astronomy, Eötvös Loránd University, Pázmány P. s. 1/A, 1117 Budapest, Hungary*
[52]*Arecibo Observatory, HC3 Box 53995, Arecibo, Puerto Rico 00612, USA*
[53]*Department of Physics, University of Puerto Rico, Mayagüez, Puerto Rico 00681, USA*
[54]*National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, Virginia 22903, USA*
[55]*Department of Physics, Penn State Abington, Abington, Pennsylvania 19001, USA*
[56]*Institute for Theoretical Physics, University of Münster, 48149 Münster, Germany*
[57]*Giant Army, 915A 17th Avenue, Seattle, Washington 98122, USA*
[58]*Department of Astrophysical and Planetary Sciences, University of Colorado, Boulder, Colorado 80309, USA*
[59]*Center for Astrophysics, Harvard University, 60 Garden Street, Cambridge, Massachusetts 02138*
[60]*Department of Physics and Astronomy, Oberlin College, Oberlin, Ohio 44074, USA*
[61]*Department of Physics, Ben-Gurion University of the Negev, Be'er Sheva 84105, Israel*
[62]*Feza Gursey Institute, Bogazici University, Kandilli, 34684, Istanbul, Turkey*

[63]Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA), Northwestern University, Evanston, Illinois 60208, USA
[64]Adler Planetarium, 1300 South DuSable Lake Shore Drive, Chicago, Illinois 60605, USA

(Received 7 July 2023; accepted 21 March 2024; published 9 May 2024)

Pulsar timing arrays (PTAs) use an array of millisecond pulsars to search for gravitational waves in the nanohertz regime in pulse time of arrival data. This paper presents rigorous tests of PTA methods, examining their consistency across the relevant parameter space. We discuss updates to the 15-year isotropic gravitational-wave background analyses and their corresponding code representations. Descriptions of the internal structure of the flagship algorithms Enterprise and PTMCMCSampler are given to facilitate understanding of the PTA likelihood structure, how models are built, and what methods are currently used in sampling the high-dimensional PTA parameter space. We introduce a novel version of the PTA likelihood that uses a two-step marginalization procedure that performs much faster in gravitational wave searches, reducing the required resources facilitating the computation of Bayes factors via thermodynamic integration and sampling a large number of realizations for computing Bayesian false-alarm probabilities. We perform stringent tests of consistency and correctness of the Bayesian and frequentist analysis methods. For the Bayesian analysis, we test prior recovery, simulation recovery, and Bayes factors. For the frequentist analysis, we test that the optimal statistic, when modified to account for a non-negligible gravitational-wave background, accurately recovers the amplitude of the background. We also summarize recent advances and tests performed on the optimal statistic in the literature from both gravitational wave background detection and parameter estimation perspectives. The tests presented here validate current analyses of PTA data.

DOI: 10.1103/PhysRevD.109.103012

I. INTRODUCTION

Since the first detection of gravitational waves (GWs) from a stellar mass black hole binary in 2015 [1], the field of GW astronomy has flourished with dozens more detections of transient signals [2]. Besides these signals at $\mathcal{O}(100)$ Hz frequencies, detected via ground-based laser interferometry, other methods can detect GWs across a wide range of frequencies. Pulsar timing arrays (PTAs) create a galactic-scale GW detector using radio telescopes to collect pulse times of arrival (TOAs) from an array of millisecond pulsars. These pulsars exhibit exceptional long-timescale arrival-time stability, allowing PTAs to use them as galactic scale clocks that are sensitive to perturbations at frequencies 1–100 nHz. The expected target signal is the stochastic GW background (GWB), possibly from an ensemble of merging supermassive black-hole binaries that could result from galactic mergers [3].

Previously, the European Pulsar Timing Array [4], the Parkes Pulsar Timing Array in Australia [5], and the North American Nanohertz Observatory for Gravitational waves (NANOGrav) [6,7] all reported detection of a red-noise process with common spectrum among pulsars, but no evidence either way for inter-pulsar correlations [8–10]. Such a process is known as CURN, for common-spectrum uncorrelated red noise. The International Pulsar Timing Array (IPTA) [11] consists of these collaborations along with the Indian Pulsar Timing Array [12] and the MeerKAT Pulsar Timing Array [13]. Combining data from older datasets from NANOGrav, the European Pulsar Timing Array, and the Parkes Pulsar Timing Array in Australia, the IPTA also found a consistent CURN in their second data release [14]. Such a signal might arise from some currently unknown noise processes shared by the measured pulsars [10,15,16]. Thus, to claim a detection of GWs we require evidence of the telltale correlations between pulsar pairs, known as Hellings and Downs (HD) correlations [17].

Among the 67 pulsars used in the GWB analysis in the 15-year data [18], NANOGrav reports a stronger detection of the CURN process. Additionally, NANOGrav reports evidence for HD correlations suggesting that the common signal seen by the three PTAs is indeed a gravitational-wave background [19]. While the parameter estimation and subsequent astrophysical interpretation remains somewhat dependent on noise models (see, e.g., [20] for more information about the noise models used), the GWB remains consistent with an origin of an ensemble of supermassive black hole binaries [21]. In addition to searching for a GWB, other analyses have been performed looking for single continuous-wave sources [22], anisotropy in the GWB [23], and possible hints of new physics [24]. Future joint analyses in the IPTA will work toward combining data to improve sensitivity to GW sources and their corresponding parameters for astrophysical interpretation.

*Deceased.
†Infinia ML, 202 Rigsbee Avenue, Durham, North Carolina 27701, USA.

In performing analyses for HD correlations, PTA collaborations employ a variety of techniques to process and analyze their data. NANOGrav uses a modular pipeline that starts with radio telescope data and ends with astrophysical GWB inference. Radio telescope data are processed into TOAs, analyzed for outliers, and fitted with a timing model [18]. Subtracting the predicted TOAs from the observed TOAs results in the TOA residuals. These residuals are broken down into terms associated with deterministic signals, as one might expect from an individual binary, as well as stochastic signals such as those from a GWB. Stochastic signals may be further broken into different types of noise processes: white and red. White noise has a flat power spectral density and is associated with noise in telescope observations. Red noise has a larger amplitude at lower frequencies and originates from pulsars themselves in the form of spin noise [25], fluctuations in the dispersion measure caused by the interstellar medium in between us and the pulsar, and a potential GWB [26]. A single-pulsar white and red noise analysis is performed at the end of this part in the pipeline. The TOAs, timing model, residuals, and noise analyses comprise the initial input for any GW analysis.

Unlike other regimes of GW data analysis, most current PTA gravitational-wave analyses are performed in the time domain [27,28]. In this paper, we specialize to the case of a GWB and describe the analysis implementations as they currently exist. The Bayesian GWB analysis started with the pioneering work of van Haasteren *et al.* [28]. While the brute force inversion of a full $N_{\mathrm{TOA}} \times N_{\mathrm{TOA}}$ matrix was possible with $\sim 10^3$ TOAs at the time of publication, it would not work today with $\sim 5 \times 10^5$ TOAs. The modern PTA likelihood was introduced in [29] where the authors expanded the red noise in a set of Fourier coefficients. Lentati *et al.* [29] marginalized over the timing model as in [28] and added an additional marginalization over the Fourier coefficients. van Haasteren and Vallisneri [30,31] used the connection between these methods and the theory of Gaussian processes leading to further optimizations in the likelihood computation and sampling.

Complementary to Bayesian approaches, the NANOGrav frequentist GWB analysis uses the so-called optimal statistic (OS), an estimator of the amplitude and significance of a GWB. Initially, this statistic was formulated in the frequency domain [32], and later it was reimplemented in the time domain [27]. Traditionally, it has been formulated in the regime where the amplitude of the GWB is much lower than the amplitude of the intrinsic red noise in the pulsars, an assumption that has been relaxed recently [33]. Additionally, the optimal statistic relies upon an estimate of the intrinsic red noise in each pulsar. Our estimates of intrinsic red noise are uncertain, and using a specific choice of the red noise can result in a biased statistic [34]. The solution is to create a hybrid Bayesian-frequentist "noise-marginalized" optimal statistic, in which the optimal statistic is computed over many posterior draws for our red noise model, creating a distribution for the amplitude estimate of the GWB and its associated significance. It is common practice to average the optimal statistic over this distribution [34], although recently alternative approaches have also been proposed [35].

The traditional optimal statistic assumes that there is only one type of spatial correlation in the data. Monopolar and dipolar correlations are also possible, resulting from systematic issues such as clock errors [36] or solar system ephemeris errors [37], respectively, and so potentially there will be multiple spatially correlated signals in the data simultaneously. When searching for only one type of spatially correlated signal at a time, one might make spurious detections due to the contribution of other spatially correlated signals. The solution to this is to simultaneously fit for multiple spatial correlation patterns using the multiple-correlation optimal statistic (MCOS) [38].

These methods have evolved conceptually over the past decade, and so have their software implementations. NANOGrav maintains a set of publicly available software packages that are written in the Python programming language and make extensive use of NumPy and SciPy [39,40]. The packages that we focus on in this work include Enterprise,[1] `enterprise_extensions`,[2] PTMCMCSampler,[3] and `la_forge`[4] [41–44]. In this paper, we review these packages and perform a campaign of simulations to validate them for the specific case of searching for a GWB with NANOGrav data, as performed in NANOGrav's 15-year GWB analysis [19]. We base all simulations in this paper on the NANOGrav 15-year dataset, using all 67 pulsars that have been timed for more than three years [18].

Inspired by probabilistic programming packages such as Stan [45] and PyMC [46], Enterprise allows the specification of probabilistic data models for PTAs, and the evaluation of the resulting priors and likelihoods. By contrast, `enterprise_extensions` contains prebuilt models that are commonly used in PTA analyses. It also includes hypermodels used in Bayesian model selection by way of product-space sampling, as well as implementations of the OS, MCOS, and a noise-averaged version of these known as the "noise-marginalized" optimal statistic (NMOS). We use PTMCMCSampler, a parallel-tempering enabled Markov-chain Monte Carlo (MCMC) sampler, to approximate the posterior of the models created with Enterprise. Finally, once we finish sampling, we use `la_forge` to compress and post-process the resulting chains.

We address two main issues in this work. First, as the number of pulsars in our dataset increases, our array gains sensitivity and the computations become longer due to an increased number of parameters and the increased cost of

[1] https://github.com/nanograv/enterprise.
[2] https://github.com/nanograv/enterprise_extensions.
[3] https://github.com/jellis18/PTMCMCSampler.
[4] https://github.com/nanograv/la_forge.

likelihood calculation. Robust Bayes factor calculations such as thermodynamic integration also require large amounts of computational resources which were previously prohibitive, taking months to complete before speeding up the likelihood calculation. Bayes factors between many of the models computed in the 15-year analysis provide an intractable challenge for the standard product-space model comparisons which have been used in previous analyses due to the large Bayes factors and difficulty of finding a good weight to allow for the chains to switch between the models easily. Thermodynamic integration solves this issue and is especially necessary in the cases where the Bayes factor is very large or very small. Performing a large series of Bayesian analyses, such as are used in the Bayesian false-alarm probability calculations [19], is also computationally prohibitive. To speed up these computations, we implement a ∼5× faster method of likelihood computation, which is equivalent mathematically to the previous method. Second, the original optimal statistic for PTAs [27] works well as a detection statistic under the no-signal null hypothesis, but remains biased as an estimator for the GWB amplitude. We check that this bias has been reduced by accounting for a GWB. Both of these methods provide crucial adjustments to the NANOGrav methods for future analyses, as the number of pulsars increases and the background becomes even more prominent in our dataset.

In Sec. II, we describe the traditional likelihood computation and a new faster implementation for situations where the white-noise parameters remain constant. We also describe the computational scaling associated with each calculation. In Sec. III we discuss using Enterprise along with PTMCMCSampler to explore the high dimensional parameter spaces in GWB analyses. Frequentist methods implemented in `enterprise_extensions` are discussed in Sec. IV. Next, we present tests on the Bayesian methods in Sec. V and frequentist methods in Sec. VI. Finally, we discuss possible future directions and conclude in Sec. VII.

II. THE PTA LIKELIHOOD CALCULATIONS

First, we describe the standard and "fast" PTA likelihoods. Both of these likelihoods use Enterprise, a pure Python package built to analyze pulsar noise, timing models, and to search for GWs in PTA data. For a discussion of Enterprise and its structure, see Appendix A.

A. The PTA likelihood

Our TOAs $\boldsymbol{t}$ can be written as

$$\boldsymbol{t} = \boldsymbol{t}_{\rm det} + \boldsymbol{t}_{\rm stoc}, \tag{1}$$

where $\boldsymbol{t}_{\rm det}$ is the deterministic part of the TOAs, $\boldsymbol{t}_{\rm stoc}$ is the stochastic part of the TOAs. After fitting the deterministic part of the signal with a least squares fit, $\boldsymbol{t}_{\rm M}$,

$$\boldsymbol{t}_{\rm det} \approx \boldsymbol{t}_{\rm M} + \boldsymbol{M\epsilon}, \tag{2}$$

where $\boldsymbol{M\epsilon}$ is a Taylor expansion of the residual deterministic part where $\boldsymbol{M}$ consists of derivatives in the Taylor expansion. The details of the process used to produce the timing model in the NANOGrav 15-year dataset can be found in the dataset paper [18]. The stochastic part of the TOAs

$$\boldsymbol{t}_{\rm stoc} = \boldsymbol{Fc} + \boldsymbol{n}, \tag{3}$$

where $\boldsymbol{c}$ represent the Fourier coefficients, $\boldsymbol{F}$ represents a discrete Fourier transform of the red noise processes in the data, and $\boldsymbol{n}$ consists of white noise. Combining all of the above and subtracting off the timing model fit, we find

$$\boldsymbol{\delta t} = \boldsymbol{t} - \boldsymbol{t}_{\rm M} \approx \boldsymbol{M\epsilon} + \boldsymbol{Fc} + \boldsymbol{n}. \tag{4}$$

As in [29], we expand red noise in a set of Fourier coefficients that determine a specific random realization of a stochastic process,

$$\boldsymbol{Fc} = \sum_{j=1}^{N} [X_j \sin(2\pi f_j t) + Y_j \cos(2\pi f_j t)], \tag{5}$$

where alternating $\boldsymbol{X}$, $\boldsymbol{Y}$ make up $\boldsymbol{c}$, $\boldsymbol{F}$ contains alternating columns of sine and cosine components, and $f_i = i/T$ with T the observing time span of the entire dataset (16.03 years in the 15-year dataset[5]). Red noise may consist of components intrinsic to the pulsar, such as spin noise, or even a gravitational-wave background. We limit the number of frequency bins N used based on the model of each red noise signal. For pulsar intrinsic red noise, we limit ourselves to 30 frequency bins, which is sufficient to capture the high-frequency content of the data. This corresponds to frequencies from about 2 to about 60 nHz. The number of frequencies used in the GWB analyses depends on how we model it. For example, a two-parameter power law with amplitude and spectral index in the 15-year dataset shows that the red noise dips below the white noise at around 14 frequency bins or around 28 nHz [19]. While we could include more frequency bins (50 were included in [47]), adding more frequencies on this model would bias the common power law spectral index and amplitude, unless a more advanced red noise model is used [48]. Using a model that allows each frequency bin to vary independently is known as a "free spectrum model." For this model, bins are not affected significantly by adjacent bins, and we use the same 30 frequency bins on the GWB that we use on the intrinsic red noise. Therefore, including more frequency bins on common signals has negligible influence on a free

[5]The dataset has been so named because the pulsar with the longest observation time span contains 15.8 years of data.

spectrum analysis, but this model also includes an additional parameter per frequency bin.

Several types of white noise parameters are searched over that adjust the TOA uncertainties found during template fitting, e.g., [49]. EQUAD $\boldsymbol{Q}$ is an extra factor added in quadrature to the TOA uncertainties. EFAC $\boldsymbol{G}$ rescales the TOA uncertainties and EQUAD together. Finally, ECORR $\boldsymbol{J}$ is an extra term that correlates different frequency bands within the same epoch, a term which here means a single observation. A single observation (epoch) could be a single day or it could be a combination of multiband data from separate days combined. Separate epochs remain completely uncorrelated. ECORR can account for pulse jitter [20]. Each of these white noise signals add one dimension per pulsar per observing backend per frequency band used to acquire data. The white noise covariance matrix can be written as

$$\boldsymbol{N} = \langle \boldsymbol{n}\boldsymbol{n}^T \rangle = \sum_{\mu} [G_{\mu}^2(\sigma_i^2 + Q_{\mu}^2)\delta_{ij} + J_{\mu}^2 \delta_{e(i)e(j)}], \quad (6)$$

where the ith TOA belongs to the backend μ, δ_{ij} denotes the Kronecker delta, and $\delta_{e(i)e(j)}$ denotes another Kronecker delta that equals 1 only when the epochs are the same for both TOAs considered and 0 otherwise.

Subtracting the deterministic and stochastic models results in the residual,

$$\boldsymbol{r} = \boldsymbol{\delta t} - \boldsymbol{M}\boldsymbol{\epsilon} - \boldsymbol{F}\boldsymbol{c}. \quad (7)$$

Under a multivariate Gaussian assumption for the noise the full likelihood can then be written as

$$p(\boldsymbol{\delta t}|\boldsymbol{c}, \boldsymbol{\epsilon}) = \frac{1}{\sqrt{\det(2\pi \boldsymbol{N})}} \exp\left(-\frac{1}{2}\boldsymbol{r}^T \boldsymbol{N}^{-1} \boldsymbol{r}\right). \quad (8)$$

We then group the matrices and vectors into a more compact notation,

$$\boldsymbol{T} = [\boldsymbol{M} \quad \boldsymbol{F}], \qquad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{c} \end{bmatrix}, \quad (9)$$

and place a Gaussian prior on the Fourier coefficients with covariance

$$\boldsymbol{B}(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{E} & 0 \\ 0 & \boldsymbol{\phi}(\boldsymbol{\eta}) \end{bmatrix} = \begin{bmatrix} \infty & 0 \\ 0 & \boldsymbol{\phi}(\boldsymbol{\eta}) \end{bmatrix}, \quad (10)$$

where $\boldsymbol{\eta}$ contains the hyperparameters, i.e., the parameters of the prior such as amplitudes or spectral indices of the GWB or red noise power laws. We let $\boldsymbol{E}$ be a diagonal matrix of very large values (10^{40} by default) in $\boldsymbol{B}$. This places an improper, almost-infinite variance Gaussian prior on the timing model parameters. However, these parameters are well determined by pulsar timing observations and thus likelihood dominated. Upon inversion, this choice marginalizes over the timing model uncertainties [29–31].

The covariance matrix for the red noise coefficients,

$$\boldsymbol{\phi} = \langle \boldsymbol{c}\boldsymbol{c}^T \rangle, \quad (11)$$

can be constructed via blocks. Each block contains

$$N_{\text{freq}} = \max(N_{\text{IRN}}, N_{\text{GWB}}), \quad (12)$$

frequencies where N_{IRN} is the number of frequencies used on the intrinsic red noise, and N_{GWB} is the number of frequencies used on a red noise process common among pulsars (correlated or not). If we denote the block containing the frequencies of pulsars as (a, b), and we further specify each frequency as (i, j), then

$$[\phi]_{(ai)(bj)} = \langle c_{ai} c_{bj} \rangle = \delta_{ij}(\delta_{ab}\varphi_{ai} + \Phi_{ab,i}), \quad (13)$$

with

$$\Phi_{ab,i} = \Gamma_{ab}\Phi_i, \quad (14)$$

where the overlap reduction function (ORF), Γ_{ab}, describes the correlations between pulsar pairs. In the isotropic GWB analysis, the ORFs we can search for include a no-correlation ORF for CURN, monopolar correlations that could be caused by clock corrections [36], dipolar correlations that could be caused by solar system ephemeris errors [37], and the HD correlations that are characteristic of a GWB:

$$\Gamma_{ab}^{\text{CURN}} = \delta_{ab}, \quad (15)$$

$$\Gamma_{ab}^{\text{MON}} = 1, \quad (16)$$

$$\Gamma_{ab}^{\text{DIP}} = \cos\theta_{ab}, \quad (17)$$

$$\Gamma_{ab}^{\text{HD}} = \frac{1}{2}\delta_{ab} + \frac{3}{2}x_{ab} \ln x_{ab} - \frac{1}{4}x_{ab} + \frac{1}{2}, \quad (18)$$

where $x_{ab} = (1 - \cos\xi_{ab})/2$ and ξ_{ab} is the angle between pulsars a and b on the sky. The terms ρ and κ_a are related to the power spectral density, $S(f)$ of the time delay caused by intrinsic red noise or a GWB, respectively, as

$$\varphi_{ai}, \Phi_i = S(f_i)\Delta f = S(f)/T. \quad (19)$$

A typical model choice for these spectra is a power law,

$$\varphi_{ai}, \Phi_i = \frac{A^2}{12\pi^2}\frac{1}{T}\left(\frac{f_i}{f_{\text{ref}}}\right)^{-\gamma} \text{yr}^2, \quad (20)$$

where the GWB characteristic strain is

$$h_c = A\left(\frac{f}{f_{\rm ref}}\right)^{\alpha}, \tag{21}$$

with $\gamma = 3 - 2\alpha$. Provided that the GWB is made up of signals from an ensemble of supermassive black hole binaries, we expect $\gamma = 13/3$ ($\alpha = -2/3$) [50]. The reference frequency, $f_{\rm ref}$, traditionally has been set to $1/\mathrm{yr}^{-1}$, and this is the value that we use in our simulations. For the intrinsic red noise and common uncorrelated red process, A and γ, A and γ are the hyperparameters, $\boldsymbol{\eta}$, for a power law spectrum model.

In the traditional form of the likelihood, we analytically marginalize over $\boldsymbol{b}$. This is a simultaneous marginalization over the timing model uncertainty and red noise coefficients,

$$p(\boldsymbol{\delta t}|\boldsymbol{\eta}) = \int p(\boldsymbol{\delta t}|\boldsymbol{b})p(\boldsymbol{b}|\boldsymbol{\eta})d\boldsymbol{b}, \tag{22}$$

where

$$p(\boldsymbol{b}|\boldsymbol{\eta}) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{b}^T\boldsymbol{B}^{-1}\boldsymbol{b}\right)}{\sqrt{\det(2\pi\boldsymbol{B})}}, \tag{23}$$

is a Gaussian prior with hyperparameters $\boldsymbol{\eta}$. Evaluating the integral gives

$$p(\boldsymbol{\delta t}|\boldsymbol{\eta}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{C})}}\exp\left(-\frac{1}{2}\boldsymbol{\delta t}^{\rm T}\boldsymbol{C}^{-1}\boldsymbol{\delta t}\right), \tag{24}$$

where the covariance matrix is now

$$\boldsymbol{C} = \boldsymbol{N} + \boldsymbol{T}\boldsymbol{B}\boldsymbol{T}^T. \tag{25}$$

Such an inversion can be efficiently evaluated with the Woodbury matrix identity,

$$\boldsymbol{C}^{-1} = \boldsymbol{N}^{-1} - \boldsymbol{N}^{-1}\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^T\boldsymbol{N}^{q-1}, \tag{26}$$

with

$$\boldsymbol{\Sigma} = (\boldsymbol{B}^{-1} + \boldsymbol{T}^T\boldsymbol{N}^{-1}\boldsymbol{T})^{-1}. \tag{27}$$

This formulation reduces the number of operations required for the computational bottleneck from an inversion of the full covariance matrix $\boldsymbol{C}$, which takes $\mathcal{O}(N_{\rm p}^3 N_{\rm TOA}^3)$ operations to the inversion of $\boldsymbol{\Sigma}$, which takes $\mathcal{O}(N_{\rm p}^3(2N_{\rm freq} + N_{\rm M})^3)$. Given that the number of TOAs is $\sim 10^5$, $N_{\rm freq} = 30$, $N_{\rm M} \sim 10^2$, the savings are significant.

The white noise covariance matrix is block-diagonal with a block for each observation epoch. We invert each block efficiently using the Sherman-Morrison formula

$$\boldsymbol{N}_b^{-1} = \boldsymbol{H}_b^{-1} - \frac{\boldsymbol{H}_b^{-1}\boldsymbol{u}\boldsymbol{u}^{\rm T}\boldsymbol{H}_b^{-1}}{J^{-2} + \boldsymbol{u}^{\rm T}\boldsymbol{N}_b^{-1}\boldsymbol{u}}, \tag{28}$$

where $\boldsymbol{H}_b$ contains the diagonal elements of the white noise covariance matrix, and

$$\boldsymbol{u}^T = (1, 1, \ldots, 1), \tag{29}$$

with length of the number of TOAs in the epoch. While this could be a non-negligible part of the computation, we fix the white noise in the GWB analysis, and therefore we do not consider the speed of evaluation here.

B. Faster likelihood for GWB analyses

White-noise parameters are varied in noise runs prior to performing any GWB analyses. After these initial runs, all white noise parameters are set to their median marginalized posterior values to reduce the total number of parameters from $\sim 10^3$ to $\sim 10^2$. When modeling the stochastic processes as a power law, this results in 136 parameters: two parameters for each of the 67 pulsar's intrinsic red noise parameters, and two for the GWB. White noise parameters in Enterprise are cached to speed up evaluation time significantly after the initial likelihood evaluation.

Starting with the same multivariate Gaussian in Eq. (8), we now marginalize in two steps instead of simultaneously. Marginalizing over the timing model,

$$p(\boldsymbol{\delta t}|\boldsymbol{c}, \boldsymbol{E}) = \int p(\boldsymbol{\delta t}|\boldsymbol{\epsilon}, \boldsymbol{c})p(\boldsymbol{\epsilon}|\boldsymbol{\eta})d\boldsymbol{\epsilon}, \tag{30}$$

with the Gaussian prior

$$p(\boldsymbol{\epsilon}|\boldsymbol{\eta}) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{E}^{-1}\boldsymbol{\epsilon}\right)}{\sqrt{\det(2\pi\boldsymbol{E})}}, \tag{31}$$

and $\boldsymbol{E} = \langle\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\rangle$. This integral results in

$$p(\boldsymbol{\delta t}|\boldsymbol{c}) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{\delta t} - \boldsymbol{F}\boldsymbol{c})^T\boldsymbol{D}^{-1}(\boldsymbol{\delta t} - \boldsymbol{F}\boldsymbol{c})\right)}{\sqrt{\det(2\pi\boldsymbol{D})}}, \tag{32}$$

with

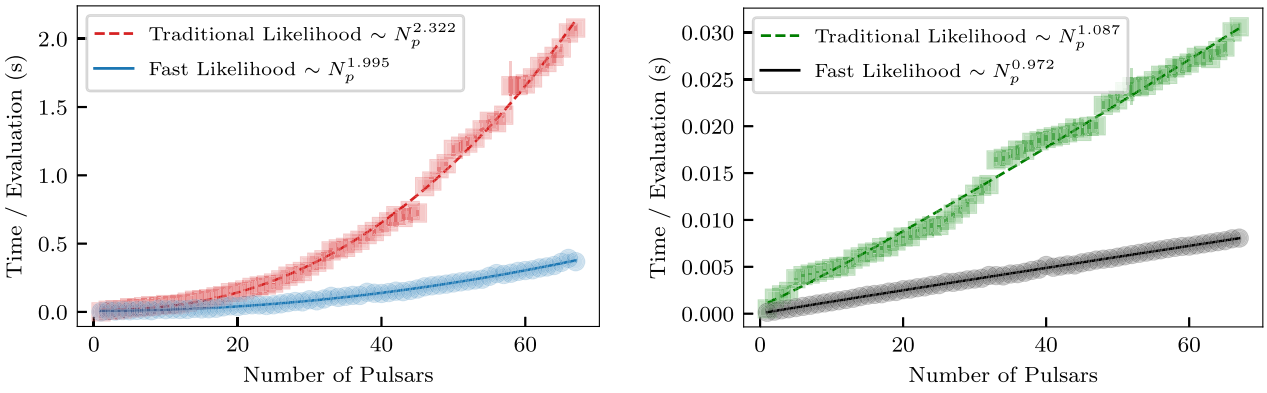
$$\boldsymbol{D} = \boldsymbol{N} + \boldsymbol{M}\boldsymbol{E}\boldsymbol{M}^T. \tag{33}$$

We use the Woodbury matrix identity for this inversion,

$$\boldsymbol{D}^{-1} = \boldsymbol{N}^{-1} - \boldsymbol{N}^{-1}\boldsymbol{M}\boldsymbol{\Lambda}^{-1}\boldsymbol{M}^T\boldsymbol{N}^{-1}, \tag{34}$$

with

$$\boldsymbol{\Lambda} = \boldsymbol{E}^{-1} + \boldsymbol{M}^T\boldsymbol{N}^{-1}\boldsymbol{M} = \boldsymbol{M}^T\boldsymbol{N}^{-1}\boldsymbol{M}, \tag{35}$$



(a) HD Model Comparison (b) CURN Model Comparison

FIG. 1. (a) Comparison between the empirical scaling of HD-correlated models for both the simultaneous marginalization procedure (labeled "traditional likelihood") and the new two-step marginalization procedure (labeled "fast likelihood") which runs faster for models where the white noise is not being varied. In the latter case, we can cache the $\boldsymbol{D}$ matrix resulting in a drastic reduction in computation time at the cost of some memory. Inversions in both methods use a sparse Cholesky decomposition method on a single CPU core. This new marginalization procedure results in an average speed increase per evaluation of 5.65× for 67 pulsars. (b) Comparison between the CURN models using the different marginalization procedures. Inverting the now diagonal matrix of these models is trivial. The reduction in number of elements required to be inverted results in a speed increase of 3.80× for 67 pulsars. Individual points and uncertainties are found by averaging over 100 evaluations of a model and taking the standard deviation. Fits to the points have been made with a nonlinear least squares algorithm fitting A, B, C to a function $f(N_p) = AN_p^B + C$.

because $\mathrm{diag}(\boldsymbol{E}) \to \infty$.[6] Now, we complete the two-step marginalization procedure by marginalizing over the Fourier coefficients,

$$p(\boldsymbol{\delta t}|\boldsymbol{\eta}) = \int p(\boldsymbol{\delta t}|\boldsymbol{c}) p(\boldsymbol{c}|\boldsymbol{\eta}) d\boldsymbol{c}, \tag{37}$$

with the Gaussian prior

$$p(\boldsymbol{c}|\boldsymbol{\eta}) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{c}^T\boldsymbol{\phi}^{-1}\boldsymbol{c}\right)}{\sqrt{\det(2\pi\boldsymbol{\phi})}}, \tag{38}$$

resulting in

$$p(\boldsymbol{\delta t}|\boldsymbol{\eta}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{K})}} \exp\left(-\frac{1}{2}\boldsymbol{\delta t}^T\boldsymbol{K}^{-1}\boldsymbol{\delta t}\right), \tag{39}$$

where

$$\boldsymbol{K} = \boldsymbol{D} + \boldsymbol{F}\boldsymbol{\phi}\boldsymbol{F}^T, \tag{40}$$

and once again we use the Woodbury matrix identity to invert this covariance matrix. We find

$$\boldsymbol{K}^{-1} = \boldsymbol{D}^{-1} - \boldsymbol{D}^{-1}\boldsymbol{F}\boldsymbol{\Theta}\boldsymbol{F}^T\boldsymbol{D}^{-1}, \tag{41}$$

with

$$\boldsymbol{\Theta} = (\boldsymbol{\phi}^{-1} + \boldsymbol{F}^T\boldsymbol{D}^{-1}\boldsymbol{F})^{-1}. \tag{42}$$

As long as $\boldsymbol{D}^{-1}$ can be cached in its entirety, the computation will be faster with the two-step marginalization, Eq. (39), over the simultaneous marginalization procedure, Eq. (24). Now, the $\boldsymbol{\Theta}$ inversion in Eq. (42) dominates the likelihood computation with $\mathcal{O}(N_p^3(2N_{\mathrm{freq}})^3)$ operations.

C. Empirical computational scaling for likelihood evaluations

As shown in Fig. 1, we find a significant improvement in the fast version of the likelihood, Eq. (39), over the traditional likelihood Eq. (24). With 67 pulsars, in the HD-correlated model, the fast likelihood leads to 5.65 times faster evaluation than the traditional computation. The speed up factor is reduced to 3.8 in the CURN model. Empirically, as the number of pulsars increases, the difference in evaluation times also increases for the HD-correlated case.

Empirically, we do not find that all computations currently scale as $\mathcal{O}(N_p^3)$. By fixing the number of frequency bins used in each model and including pulsars in alphanumeric order, we can compare the matrix inversion cost among models and check how the computations

[6]Formally, we use the matrix determinant lemma to evaluate the determinant in Eq. (32) as

$$\det(\boldsymbol{D}) = \det\left(\boldsymbol{E}^{-1} + \boldsymbol{M}^T\boldsymbol{N}^{-1}\boldsymbol{M}\right)\det(\boldsymbol{E})\det(\boldsymbol{N}), \tag{36}$$

where $\mathrm{diag}(\boldsymbol{E}) = \infty$. However, in practice we can use 10^{40} as an effectively infinite value and avoid dealing with the infinite determinant term.

scale with number of pulsars. There are two separate likelihood implementations: a sparse version that uses `scipy.sparse.csc` compressed sparse column matrices along with a sparse Cholesky decomposition `scikit-sparse` [51–53] and a version that uses a dense Cholesky decomposition. Through profiling, we find that the dense computation indeed scales as $\mathcal{O}(N_p^3)$ and is generally slower than the sparse method on a single core. The sparse method instead scales as $\mathcal{O}(N_p^2)$ using the fast likelihood, Eq. (39), or as $\mathcal{O}(N_p^{2.3})$ when using the slower version of the likelihood, Eq. (24).

This $\mathcal{O}(N_p^{2.3})$ empirical scaling of the traditional likelihood displayed in Fig. 1(a) can be traced to the sparse Cholesky inversion $\mathbf{\Sigma}^{-1}$. In general, the sparse Cholesky inversion does not have a strict scaling but depends on how the matrix is structured. Because of this, the scaling of the sparse Cholesky inversion may have some dependence on the number of frequencies and pulsars used. By profiling the code to see which parts use the largest fraction of the total evaluation time, we find that in the sparse cases, changing from a dense to a sparse matrix takes up the majority of the computation time, and the inversion is performed very quickly by comparison. This means that our inversion has been sped up to a point where it is no longer the bottleneck of the computation, and the $\mathcal{O}(N_p^2)$ pieces of the computation dominate the overall scaling for Eq. (39) evaluations. Future work will aim to reduce the computation time required in these $\mathcal{O}(N_p^2)$ operations, perhaps by working with sparse matrices from the start.

Indeed, this increased speed will be important in future analyses. The addition of pulsar specific noise models to the analysis may require an increase in the number of frequencies $N_{\rm freq}$ used in the Fourier basis. Increasing the number of frequencies means that the bottleneck of the calculation, the matrix inversion, now takes even longer. With a hundred pulsars and additional frequencies included, a single likelihood evaluation can take seconds to evaluate. GWB analyses currently require millions of samples, implying that, without the faster likelihood, a single analysis would take a significant fraction of a year or more to complete.

III. BAYESIAN METHODS AND SAMPLING THE PTA PARAMETER SPACE

GW data analyses over the past decade have made use of both Bayesian and frequentist statistical techniques. Bayesian methods rely on Bayes' theorem for parameter estimation and model selection,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta})}{p(\boldsymbol{y})}. \tag{43}$$

Given a descriptive model of the relevant data as a likelihood $p(\boldsymbol{y}|\boldsymbol{\theta})$ and a prior distribution $p(\boldsymbol{\theta})$ on the parameters, we can sample the (unnormalized) posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$, thus reallocating credibility from the prior across parameter values. The normalizing factor in the denominator is known as the marginal likelihood or evidence, and it may be used in deciding which of a set of models is preferred.

Numerical methods to estimate $p(\boldsymbol{\theta}|\boldsymbol{y})$ when the dimension of $\boldsymbol{\theta}$ is large typically depend on a form of stochastic sampling to perform high-dimensional integrals for both parameter estimation and model selection. In NANOGrav, we use PTMCMCSampler [43], a parallel-tempering (PT) enabled Metropolis-Hastings MCMC sampler. This has been the NANOGrav sampler of choice for many years. It has also recently been compared to other options and recommended for use in PTA data analysis [54]. To sample the high-dimensional PTA parameter space, we use the default proposal distributions that come with PTMCMCSampler, and some other custom proposals which will be enumerated and described below. By default, PTMCMCSampler uses single-component adaptive metropolis (SCAM), adaptive metropolis (AM), and differential evolution (DE) proposal distributions, referred to in the code as "jump proposals." Additionally, the sampler supports adding custom distributions. Details of this sampler were also discussed in [55], but we reiterate them here with any changes that have since been made.

A. Metropolis-Hastings algorithm

We use the Metropolis-Hastings algorithm [56,57] to stochastically sample the posterior distribution given by the likelihood and chosen priors. After enough iterations, the sampler converges to a stable distribution that approximates the posterior well regardless of where in the parameter space we start. We will show tests of this in Sec. V. Under certain circumstances convergence is guaranteed with an infinite number of samples, but in the high-dimensional parameter space that we search over, the amount of time required for convergence can be prohibitive without well-chosen proposal distributions.

First, we start with an initial set of parameters θ at iteration $t = 0$. Then, we draw a new point $\boldsymbol{\theta}_*$ from a proposal distribution for each iteration $J_t(\boldsymbol{\theta}_*, \boldsymbol{\theta})$. In the Metropolis algorithm [56], such proposal distributions are required to be symmetric, but the Metropolis-Hastings (MH) algorithm [57] allows for asymmetric distributions by the inclusion of the Hastings ratio. The MH algorithm leads to the proposed point being accepted with log probability [58]

$$\ln A = \min(0, \ln R + \ln H), \tag{44}$$

where R is the ratio of probabilities between the proposed point and the old one,

$$\ln R = \ln p(\boldsymbol{\theta}_*|\boldsymbol{y}) - \ln p(\boldsymbol{\theta}|\boldsymbol{y}), \tag{45}$$

and H is a ratio which accounts for asymmetric proposal distributions

$$\ln H = \ln J_t(\boldsymbol{\theta}|\boldsymbol{\theta}_*) - \ln J_t(\boldsymbol{\theta}_*|\boldsymbol{\theta}). \tag{46}$$

Iterating for each t returns a chain of samples from the posterior distribution. However, the number of samples required to settle into a stationary distribution that approximates the posterior well depends on several factors including the autocorrelation length, which is related to the number of parameters, and whether the distribution is multimodal where the chain might get stuck in a single mode. At each iteration, we draw a point from the proposal distributions which is either accepted or rejected. If the new point is accepted, then it is added to the chain. Otherwise, if the new point is rejected, then the current point is added to the chain instead. Chains produced in this way contain samples that are not completely independent, and they are correlated with themselves. In the case of a MH sampler, the autocorrelation length is typically of the same order as the number of parameters.

B. Parallel tempering

To improve exploration and mixing of the sampler, we sample multiple chains with different exponents, known as temperatures, and propose swaps between them in a sampling scheme known as PT [59]. The posterior now contains the likelihood raised to some power,

$$p(\boldsymbol{\theta}|\boldsymbol{y},\beta) = p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta})^{\beta}, \tag{47}$$

where $\beta = 1/T$ and T is known as the chain's temperature. Samples from one chain are propagated to the next via swap proposals between chains of different temperatures. The higher the temperature is, the more the posterior becomes like the prior, enabling exploration of the parameter space and reducing the autocorrelation length of all chains via swaps, increasing the number of effectively independent samples. Though the cost of this scheme is more evaluations of the likelihood, we often find that this is more efficient because of the increased number of effectively independent samples returned. It can also be used to find the evidence as will be discussed in a later section. We set a temperature for each of the chains using a geometrically spaced ladder,

$$T_i = \left(\frac{T_{\max}}{T_{\min}}\right)\exp(i), \tag{48}$$

where $i = 0, 1, 2, \ldots$, which should result in a $\sim$25% temperature swap acceptance rate between adjacent chains when sampling a multivariate Gaussian distribution [55]. In PTMCMCSampler, swaps are proposed between adjacent chains, though in general swaps could be proposed between any two chains in the ladder. Swaps are accepted between temperatures T_i and T_j with log probability,

$$\ln A_{ij} = \min(0, (\beta_j - \beta_i)\ln L_{ij}), \tag{49}$$

where

$$\ln L_{ij} = \ln p(\boldsymbol{y}|\boldsymbol{\theta}_i) - \ln p(\boldsymbol{y}|\boldsymbol{\theta}_j), \tag{50}$$

is the log likelihood ratio. Other temperature ladders are possible including adaptive temperature spacing based on a constant acceptance rate [60]. This improves the PT scheme when sampling a posterior that is not a multivariate Gaussian. While these different temperature spacings certainly have advantages, they are not currently implemented in PTMCMCSampler.

To perform parallel tempering swaps, we use multiple cores to sample each chain simultaneously through the use of `MPI` [61] and `MPI4Py` [62,63]. Previously, the sampler used an asynchronous model for the temperature swaps, so that chains could sample at their own paces and swap as soon as the next one down reaches a specified interval. These processes are now synchronized using blocking commands, which are necessary for the standard product space sampling method that NANOGrav uses for model selection [64] (see Appendix B for a discussion of the necessity of synchronizing the sampler).

C. Jump proposals

The GWB analysis uses several proposal distributions. These distributions are critical for timely convergence and exploration of the parameter space. Ideally, jump proposals match the posterior closely to minimize the autocorrelation length of the chain and thus reduce the number of samples that need to be taken. The combination of all of these proposals has proven to work well for the problem at hand, even with the $\mathcal{O}(100)$ parameters that we work with in the 15-year dataset. The proposal distributions discussed here consist of the default in PTMCMCSampler along with empirical distributions and prior draws.

1. Adaptive metropolis

Upon initializing the sampler, PTMCMCSampler takes a list of "parameter groups" as an argument. If we believe that multiple parameters will be correlated, then we can add them as a group, and the sampler will propose jumps in this subspace. The full sample space of all parameters together is always a group regardless of new groups. However, if no groups are given, the entire sample space is considered the only group. Typically, power law amplitude and spectral indices are grouped together with a sampling group for each individual pulsar.

PTMCMCSampler also requires a sample covariance matrix at initialization. Periodically, we compute a sample covariance matrix $\boldsymbol{C}_s$ from the chain's history for each sample

group using an online algorithm (see Appendix D) to avoid storing the entirety of the chain. $\boldsymbol{C}_s$ is then decomposed via a singular-value decomposition,

$$\boldsymbol{C}_s = \boldsymbol{U}_s \boldsymbol{\Sigma}_s \boldsymbol{V}_s{}^T. \tag{51}$$

This provides a robust generalization of the eigenvalue decomposition and is used to jump along correlated directions in the parameter space.

The adaptive Metropolis proposal distribution [65] uses the sample covariance matrix to propose jumps in uncorrelated directions of the parameter space. First, we move the usual parameters at the ith step of the chain $\boldsymbol{\theta}_i$ into the parameter combinations using

$$\boldsymbol{\zeta}_i = \boldsymbol{U}_s{}^T \boldsymbol{\theta}_i. \tag{52}$$

Each jump proposed is given by a normal distribution,

$$\boldsymbol{\zeta}_{i+1} = \boldsymbol{\zeta}_i + \sqrt{\boldsymbol{\Sigma}_g}\mathcal{N}(0, c_d), \tag{53}$$

where $\boldsymbol{\Sigma}_g$ is the sample covariance matrix for the specific group in which the jump is proposed, and

$$c_d = \frac{2.4s}{\sqrt{2n_{\rm dim}}}, \tag{54}$$

where s is a scale parameter and $n_{\rm dim}$ depends on the number of dimensions of the group that the jump is proposed in.. For each default jump, 3% of jumps have their scale multiplied by 10, 7% of jumps have their scale multiplied by 0.2, and the other 90% have unmodified scale. In all cases, the relative scale of the jump is adjusted based on the temperature of the chain. The mix of small, medium, and large jumps helps the sampler find the scale of the parameter space being explored. To project back into a jump in the original parameters, we use

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{U}\boldsymbol{\zeta}_{i+1}. \tag{55}$$

2. Single-component adaptive metropolis

Similar to the adaptive Metropolis jump, the SCAM jump [66] uses the sample covariance matrix, but only moves along one uncorrelated direction in the parameter space. Once again, we start by projecting onto the uncorrelated combinations of parameters $\boldsymbol{\zeta} = \boldsymbol{U}^T\boldsymbol{\theta}$. We then propose a jump in a single parameter direction as

$$\zeta^j_{i+1} = \zeta^j_i + \sigma^j_s \mathcal{N}(0, c_d), \tag{56}$$

where σ^j_s is the jth diagonal element of the diagonal matrix $\sqrt{\boldsymbol{\Sigma}_s}$, and j labels the uncorrelated parameter for which we are proposing a new point. Finally, we move back to the original set of parameters using

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{U}\boldsymbol{\zeta}_{i+1}. \tag{57}$$

3. Differential evolution

The final default proposal distribution in PTMCMCSampler uses a simple genetic algorithm known as DE [67]. This algorithm takes two samples from the history of the chain, subtracts them, and proposes a jump along that direction. In the current version of PTMCMCMSampler, the full chain is not stored, and it instead draws from a buffer formed over many unthinned iterations of the sampler. By keeping this buffer much longer than the autocorrelation length of the sampler, we ensure that the draws come from a stationary distribution. The DE jump draws two samples $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_n$ and then proposes a jump

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + s_{\rm DE}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n), \tag{58}$$

where $s_{\rm DE} = 1$ or

$$s_{\rm DE} \sim \text{Uniform}\left[0, \frac{2.4}{\sqrt{2\beta n_{\rm dim}}}\right], \tag{59}$$

each with 50% probability with $\beta = 1/T$ and $n_{\rm dim}$ the number of dimensions in which the jump is proposed.

4. Empirical distributions

Before the GWB analysis, we run a noise analysis on each pulsar individually. This run includes white noise (EFAC, EQUAD, ECORR) and a power-law intrinsic red noise (amplitude and spectral index). From each of these runs, we find a posterior for the intrinsic red noise, and we create white noise dictionaries with which to set the white noise values constant during the full analyses. Out of these posteriors, we can make 1D or 2D histograms that we can then sample from during the full GWB analysis to propose as new points. During the creation of these histograms, all bins' counts are incremented by one to allow exploration of the entire prior. They are then checked to make sure they cover the prior before starting any sampling that includes them so that they do not bias parameter recovery. Such histograms are known as empirical distributions (see Appendix A in [68]). Typically, we use 2D empirical distributions on the intrinsic red noise parameters for each pulsar. This provides excellent proposals if the empirical distributions are somewhat close to the posteriors on the parameters that we propose jumps in. Empirical proposal distributions reduce the number of samples required to achieve stationarity, sometimes called the burn-in, significantly. Importantly, empirical distributions are only a small part of our overall mix of proposal distributions that include

many proposals that suggest points across the entire prior space.

D. Parameter estimation

Parameter estimation provides crucial information for astrophysical inference. The full multidimensional posterior, can be used to find the marginalized posteriors for each of the parameters. Convergence and exploration of the parameter space is critical to finding the maxima of the likelihood function and to sampling them effectively. GWB analyses typically involve millions of likelihood evaluations due to the large autocorrelation lengths of the chains. It can take many evaluations to get a reasonable number of effective samples. As a test of the procedure, we have verified the different runs on a single simulation return the same result regardless of where we start in the parameter space.

1. Gelman-Rubin $\hat{R}$ diagnostic

We use the Gelman-Rubin $\hat{R}$ diagnostic [69] to check the stationarity of the chain. This does not necessarily mean that the chains have converged, but it tells us that the samples are coming from one part of the parameter space. The diagnostic splits the MCMC chain into multiple segments and checks the within-chain and between-chain statistics to confirm that the chain is in a stationary state. A threshold is required to tell whether the chain passes or fails the diagnostic. As suggested by [70], we use 1.01 as the $\hat{R}$ threshold. In all tests performed in this work, we use the $\hat{R}$ diagnostic to make sure the chains are stationary before performing tests. This diagnostic is implemented in `la_forge` for ease of use in PTA data analysis.

E. Model selection: Bayes factor calculations

To compare models, we need to compute the marginal likelihood or evidences for each model. Division of these evidences return Bayes factors. The standard method of calculation used in NANOGrav is a form of product space sampling [64,71]. We refer to this method as the hypermodel framework (henceforth hypermodel), and it is implemented in `enterprise_extensions`. Other methods for finding the model evidence include reweighting the posterior [72], thermodynamic integration [73,74], and nested sampling [75].

1. Hypermodel framework

The hypermodel concatenates different models into a single model by combining their joint parameters into a set of parameters that contains only the unique parameters between the models. During sampling, a continuous "switch" parameter called `nmodel` changes between the models turning on only the parameters that belong to the "on" model. By sampling the models and this switch between them, we can compute an odds ratio comparing how many times each of the models were sampled. This corresponds directly to a Bayes factor between the two models. The uncertainties are then computed using a standard bootstrap in which we resample the thinned `nmodel` marginalized posterior with replacement and recompute the odds ratio. The odds ratio is averaged and a standard deviation calculated over a number of realizations to give the final Bayes factor[7] with uncertainties.

In current data, we often find the situation where Bayes factors for one model or another are significantly disfavored. Adding a log weight to the log likelihood can remedy such a situation. To accomplish this, we add a constant value to the log likelihood, scaling the likelihood by the exponential of the log weight. We can find a weight estimate by subtracting the maximum posterior values between the two models being compared. This results in a more even mixing between the two models, and the weight can be undone in postprocessing by multiplying the Bayes factor by the exponential of the log weight.

2. Reweighting

Reweighting is a simple technique that utilizes existing samples from a probability distribution, the approximate, to obtain an estimate of some other probability distribution, the target, which shares support with the approximate. Each existing sample is "weighted" by the ratio of the target and approximate probability densities. These weighted samples are an estimate of the target distribution, whereas the weights can be used to estimate Bayes factors and uncertainties. Since the samples of the distribution have already been produced, each weight can be calculated in parallel, increasing the speed at which the target space can be evaluated. Reweighting results in a reduction in the number of effectively independent samples based on how disjointed the posteriors are between the approximate and target, but it often is still much faster than directly sampling the target posterior directly. This technique is particularly effective in cases where the two distributions have similar support on similar regions in the parameter space but one distribution is significantly faster to evaluate. In the context of PTAs, reweighting has been used to generate HD posteriors from the faster-to-evaluate CURN posteriors [72].

3. Nested sampling

Nested sampling [76], a staple for GW analyses in current ground-based detectors, returns a Bayes factor and posterior samples. It computes the evidence integral by turning the multidimensional integral into a one-dimensional integral. N "live" points are sampled from the priors and the space outside of the lowest likelihood

[7]Note that the odds ratio is only equal to the Bayes factor if the prior odds are the same for each model in consideration.

point is removed, thereby reducing the volume considered by approximately $1/N$. In this way, nested sampling climbs the likelihood distribution in a global way and eventually reaches a stopping criterion set by the error on the log evidence. Overall it has a reputation for being easy to use, for being good at finding multimodality, and for having stopping conditions that do not require much input from the user [77]. In PTA data, we find it difficult to use nested sampling on the full GWB analysis due to the high number of parameters. However, we use nested sampling with a reduced set of the data below to check our Bayes factors with 30 parameters. In this case, using Ultranest [75] returns the evidence for each model, but with much larger uncertainty over the same computation time span as the hypermodel.

4. Thermodynamic integration

If we use enough chains over a broad enough set of temperatures, then parallel tempering also can be used to compute the evidence. By taking the average of the log likelihood on each sampled temperature, we can integrate over this to yield the model evidence. The log evidence is given by [73,74],

$$\ln\, p(\boldsymbol{\delta t}|M) = \int_{-\infty}^{0} \beta \mathrm{E}_{\beta}[\ln\, p(\boldsymbol{\delta t}|\boldsymbol{\eta}, M)] d\, \ln\beta, \quad (60)$$

where $\beta = 1/T$, M is the model of interest, and we are integrating with respect to $\ln\beta$. We use two separate methods to evaluate uncertainties on evidence estimates. In one, we use a cubic spline which is fit using a trans-dimensional algorithm known as the reversible jump MCMC algorithm [78], and in the other we use a bootstrap on an interpolation between points in the integrand of Eq. (60). There are two types of uncertainty here: the discretization error that is determined by the number of temperatures that we choose, and the sampling error that is determined by the number of independent samples.

IV. FREQUENTIST METHODS AND THE OPTIMAL STATISTIC

Fully Bayesian methods, especially when used for model selection, can be computationally expensive. In this section, we consider a detection statistic and an estimator for the GWB that are built from directly cross-correlating arrival times between pulsars. We first consider the statistic in the case where the amplitude of the GWB is small compared to the noise, which is what has traditionally been assumed. This statistic is still important for null hypothesis testing, although it can be improved upon when used as an estimator for the strength of the background. We then move on to consider the situation where the background cannot be neglected, and present an estimator that properly accounts for the background itself. We also discuss how to construct a "binned" estimator across the sky to yield a HD reconstruction that takes into account the size of the background. We finish by presenting a version of the OS that simultaneously fits for multiple correlation patterns. We also highlight the effect of the choice of noise parameters used in constructing the OS.

A. Traditional optimal statistic

We begin by considering the noise-weighted match between the correlation of data in pulsar a with data in pulsar b:

$$\rho_{ab} = \frac{\boldsymbol{\delta t}_a^T \mathbf{P}_a^{-1} \tilde{\boldsymbol{\Phi}}_{ab} \mathbf{P}_b^{-1} \boldsymbol{\delta t}_b}{\mathrm{tr}(\mathbf{P}_a^{-1} \tilde{\boldsymbol{\Phi}}_{ab} \mathbf{P}_b^{-1} \tilde{\boldsymbol{\Phi}}_{ba})} \equiv \boldsymbol{\delta t}_a^T \boldsymbol{Q}_{ab} \boldsymbol{\delta t}_b, \quad (61)$$

$$\boldsymbol{Q}_{ab} = \frac{\mathbf{P}_a^{-1} \tilde{\boldsymbol{\Phi}}_{ab} \mathbf{P}_b^{-1}}{\mathrm{tr}(\mathbf{P}_a^{-1} \tilde{\boldsymbol{\Phi}}_{ab} \mathbf{P}_b^{-1} \tilde{\boldsymbol{\Phi}}_{ba})}, \quad (62)$$

where for two different pulsars, a and b, we have

$$\mathbf{P}_a = \langle \boldsymbol{\delta t}_a \boldsymbol{\delta t}_a^T \rangle = \boldsymbol{D}_a + \boldsymbol{F}_a \boldsymbol{\phi}_{aa} \boldsymbol{F}_a^T, \quad (63)$$

$$\tilde{\boldsymbol{\Phi}}_{ab} = \frac{\boldsymbol{F}_a \boldsymbol{\phi}_{ab} \boldsymbol{F}_b^T}{\Gamma_{ab} A_{gw}^2}. \quad (64)$$

The normalization of $\tilde{\boldsymbol{\Phi}}_{ab}$ is chosen such that $\langle \rho_{ab} \rangle = \Gamma_{ab} A_{\mathrm{gw}}^2$. In the small signal regime, the variance of this correlation is $\sigma_{ab}^2 = (\mathrm{tr}\mathbf{P}_a^{-1} \tilde{\boldsymbol{\Phi}}_{ab} \mathbf{P}_b^{-1} \tilde{\boldsymbol{\Phi}}_{ba})^{-1}$. If we assume that the cross-correlation for one pair of pulsars is not correlated with the cross-correlation of another pair of pulsars, i.e., $\langle \rho_{ab} \rho_{cd} \rangle \propto \delta_{ac} \delta_{bd}$, then we can perform a variance-weighted, HD-matched sum over these correlations to estimate the amplitude of the GWB from all pairs,

$$\hat{A}_{\mathrm{gw}}^2 = \frac{\sum_a \sum_{b>a} \rho_{ab} \Gamma_{ab} \sigma_{ab}^{-2}}{\sum_a \sum_{b>a} \Gamma_{ab}^2 \sigma_{ab}^{-2}}, \quad (65)$$

$$\sigma_{\mathrm{gw}}^2 = \left(\sum_a \sum_{b>a} \sigma_{ab}^{-2} \Gamma_{ab}^2 \right)^{-1}. \quad (66)$$

This statistic is often used for null hypothesis testing for GWB detection. Under the (null) assumption that $A_{\mathrm{gw}}^2 = 0$, the variance of this estimator can be used to construct a signal-to-noise ratio (S/N),

$$\mathrm{S/N} = \frac{\sum_a \sum_{b>a} \rho_{ab} \Gamma_{ab} \sigma_{ab}^{-2}}{(\sum_a \sum_{b>a} \Gamma_{ab}^2 \sigma_{ab}^{-2})^{1/2}}, \quad (67)$$

which we calculate on the data and then compare to its expected distribution under the null hypothesis. The distribution for this statistic is a generalized χ^2 distribution [79], but it is often estimated empirically using methods that destroy correlations but preserve potential mismodeling. Two such methods are sky scrambles [80] and phase shifts [81].

Construction of the matrices in Eqs. (63) and (64) requires a choice of hyperparameters $\boldsymbol{\eta}$. Specifically, the red noise parameters for each pulsar are used to construct $\boldsymbol{D}_a$, and the CURN amplitude and spectral index are used to construct $\boldsymbol{\phi}_{ab}$. The natural choice for these parameters are those taken from the Bayesian analysis. However, choosing the maximum likelihood parameters for $\boldsymbol{\eta}$ from a fully Bayesian run, or jointly maximizing the individual 1D posteriors for each parameter, leads to a bias in the recovered value of $A^2_{\rm gw}$ [34]. Part of this bias is due to making a single choice of noise parameters. By averaging the statistic calculated over draws of $\boldsymbol{\eta}$ from a posterior chain, resulting in what is referred to as the NMOS, this bias can be partially alleviated. Additionally, a single choice of hyperparameters could result in a larger value of the S/N than is representative of the dataset. In general, therefore, the S/N is averaged over many draws from $p(\boldsymbol{\eta}|\boldsymbol{\delta t})$, and this average is used as a detection statistic. Other approaches have also been proposed, e.g., averaging the p value associated with the S/N for each draw, instead of averaging the S/N [35].

The OS defined in Eqs. (65) and (66). It is implemented in the `compute_os` method of the `OptimalStatistic` class, which is found in the `frequentist.optimal_statistic` module of `enterprise_extensions`. The NMOS (described in the previous paragraph) is implemented as the `compute_noise_marginalized_os` of the same class, and takes a MCMC chain and a list of parameter names as input.

B. Optimal statistic with a non-negligible GWB

In the case where $A^2_{\rm gw}$ is comparable to the red noise level in some pulsars, the assumption that $\langle\rho_{ab}\rho_{cd}\rangle \propto \delta_{ac}\delta_{bd}$ breaks down. We must account for the covariance between correlations when constructing both $\hat{A}^2_{\rm gw}$ and especially its variance, which will be dominated by the background itself. When accounting for the covariance between correlations due to the GWB we find

$$\boldsymbol{\Sigma}_{\boldsymbol{ab,cd}} = \langle\rho_{ab}\rho_{cd}\rangle - \langle\rho_{ab}\rangle\langle\rho_{cd}\rangle, \tag{68}$$

$$= \mathrm{tr}(\boldsymbol{Q}_{ba}\mathbf{P}_{ac}\boldsymbol{Q}_{cd}\mathbf{P}_{db}) + \mathrm{tr}(\boldsymbol{Q}_{ba}\mathbf{P}_{ad}\boldsymbol{Q}_{dc}\mathbf{P}_{cb}), \tag{69}$$

where

$$\mathbf{P}_{ab} = \langle\boldsymbol{\delta t}_a\boldsymbol{\delta t}_b^T\rangle = \delta_{ab}\boldsymbol{D}_a + \boldsymbol{F}_a\boldsymbol{\phi}_{ab}\boldsymbol{F}_b^T. \tag{70}$$

We can then construct a least-squares estimator for the background using this covariance matrix,

$$\hat{A}^2_{\rm gw} = \frac{\boldsymbol{\Gamma}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\rho}}{\boldsymbol{\Gamma}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}}, \tag{71}$$

$$\sigma^2_{A^2_{\rm gw}} = (\boldsymbol{\Gamma}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma})^{-1}, \tag{72}$$

where $\boldsymbol{\Sigma}$ is given by Eq. (68), $\boldsymbol{\rho}$ is a vector of paired correlations, and $\boldsymbol{\Gamma}$ is a vector of the HD correlation coefficients corresponding to each pair. It is important to note here that construction of $\mathbf{P}_{ab}$ and therefore $\boldsymbol{\Sigma}$ requires a choice of $\boldsymbol{\eta}$. That is, some choice of $A^2_{\rm gw}$ is needed to actually construct our estimator. One can take an iterative approach, where we begin with a choice of $\boldsymbol{\eta}$ [e.g., the maximum *a posteriori* draw from $p(\boldsymbol{\eta}|\boldsymbol{\delta t})$], evaluate Eq. (71), and then use the resulting $\hat{A}^2_{\rm gw}$ to construct $\boldsymbol{\Sigma}$, iterating until convergence. In practice, we have found this converges rapidly, and is consistent with results that use a single iteration with an initial choice of $A^2_{\rm gw}$ estimated from the posterior $p(\boldsymbol{\eta}|\boldsymbol{\delta t})$.

It is also common to estimate the strength of the observed background using subsets of paired correlations that have similar separations on the sky. The individual bins should then trace the HD curve. We collect pairs that fall into a single bin, initially choosing the number of bins we would like, and then assigning pulsar pairs to bins such that there are roughly the same number of pairs in each bin. We label the average angular separation between pulsars in the ith bin as ξ_i, and construct an estimator for the correlated power in each bin, $\rho_{{\rm opt},i}$, whose expectation is given by $\langle\rho_{{\rm opt},i}\rangle = \Gamma_{\xi_i}A^2_{\rm gw}$. In this case Γ_{ξ_i} is the HD correlation coefficient evaluated at the average angular separation for pulsar pairs in the ith bin. Other choices could also be made, and would slightly change the results [33]. Motivated by our choice of binning, we can imagine $\boldsymbol{\Sigma}$ taking a block form where the i, i block corresponds to correlations between pairs in bin ξ_i, and the i, j block corresponds to correlations between pairs in ξ_i and pairs in ξ_j. The resulting estimator is given by [33]

$$\rho_{{\rm opt},i} = \Gamma_{\xi_i}\frac{\boldsymbol{\Gamma}_i^T\boldsymbol{\Sigma}_{ii}^{-1}\boldsymbol{\rho}_i}{\boldsymbol{\Gamma}_i^T\boldsymbol{\Sigma}_{ii}^{-1}\boldsymbol{\Gamma}_i}, \tag{73}$$

where $\boldsymbol{\Gamma}_i$ is a vector of the overlap reduction function for all pairs in bin i, $\boldsymbol{\rho}_i$ is the vector of paired correlations in bin i. The nonzero GWB induces correlations between these bins as well, and this covariance matrix is given by [33]

$$B_{ij} = \langle\rho_{{\rm opt},i}\rho_{{\rm opt},j}\rangle - \langle\rho_{{\rm opt},i}\rangle\langle\rho_{{\rm opt},j}\rangle, \tag{74}$$

$$= \Gamma_{\xi_i}\Gamma_{\xi_j}\frac{\boldsymbol{\Gamma}_i^T\boldsymbol{\Sigma}_{ii}^{-1}\boldsymbol{\Sigma}_{ij}\boldsymbol{\Sigma}_{jj}^{-1}\boldsymbol{\Gamma}_j}{(\boldsymbol{\Gamma}_i^T\boldsymbol{\Sigma}_{ii}^{-1}\boldsymbol{\Gamma}_i)(\boldsymbol{\Gamma}_j^T\boldsymbol{\Sigma}_{jj}^{-1}\boldsymbol{\Gamma}_j)}. \tag{75}$$

C. Optimal statistic for multiple correlation patterns

Reference [38] arrived at the MCOS through a χ^2 approach, with

$$\chi^2 = \sum_{ab}\left(\frac{\rho_{ab} - \sum_\alpha A^2_\alpha\Gamma^\alpha_{ab}}{\sigma_{ab}}\right)^2, \tag{76}$$

ρ_{ab} given by Eq. (61), and σ_{ab} its associated uncertainty. The label α now indexes different spatial correlation patterns. For example, one can jointly minimize χ^2 with respect to A^2 for both a HD-correlated process and a monopole process, and calculate the associated covariance matrix between those estimators.

By generalizing the optimal statistic to include more than one ORF simultaneously, one arrives at the MCOS,

$$\hat{A}^2_\alpha = \sum_\beta B_{\alpha\beta} C^\beta, \tag{77}$$

where α and β are individual ORFs, and

$$B^{\alpha\beta} \equiv \sum_a \sum_{b>a} \frac{\Gamma^\alpha_{ab}\Gamma^\beta_{ab}}{\sigma^2_{ab}}, \tag{78}$$

$$C^\beta \equiv \sum_a \sum_{b>a} \frac{\rho_{ab}\Gamma^\beta_{ab}}{\sigma^2_{ab}}. \tag{79}$$

When denoting matrices in this notation, upper and lower indices indicate matrix inverses with respect to one another. The variance on the individual estimators for each spatially correlated process is given by

$$\sigma^2_{\hat{A}^2_\alpha} = B_{\alpha\alpha}. \tag{80}$$

The noise-marginalized version of the MCOS follows a similar structure as the noise-marginalized version of the original optimal statistic. By drawing parameters from the MCMC chains, we average over the noise and return a distribution of values.

The MCOS is implemented in the `compute_multiple_corr_os` method of the `OptimalStatistic` class. The noise marginalized version of the MCOS is implemented as the `compute_noise_marginalized_multiple_corr_os` method.

V. TESTS OF BAYESIAN METHODS: PTMCMCSampler AND Enterprise

Here, we perform tests robustness of our Bayesian methods. The tests performed here use the faster likelihood with the two-step marginalization procedure. We begin these tests by checking prior recovery. This tests that our proposal distributions satisfy detailed balance, a condition required for the chain samples to reflect the posterior. Next, we create simulations based on the 15-year NANOGrav data and check for unbiased posteriors. Simulations are produced directly from Enterprise models using the TOAs from the 15-year data. Therefore, good posterior recovery implies that the models we use are self-consistent and that the recovered posteriors are in the correct place with the right width. Finally, we use a reduced version of the data to check that Bayes factors agree among different methods of computation. If the different methods agree, we conclude that our calculations are working properly.

A. Prior recovery tests

To test the proposal distributions incorporated in PTMCMCSampler and `enterprise_extensions`, we sample a posterior that is equal to the priors by setting the likelihood equal to the prior and the prior to a constant. PT only tempers the likelihood in PTMCMCSampler, so it is necessary to sample the prior as the likelihood to also test this part of the sampler. If the proposal distributions satisfy detailed balance, then the recovered posterior equals the input priors within sampling uncertainties. PTMCMCSampler contains three default proposal distributions known inside the code as "jump proposals." Additional jump proposals come from `enterprise_extensions`, and which proposals get used depends on the type of search being performed. The isotropic GWB analyses includes AM, SCAM, and DE proposals. Additionally, this search includes prior draws and two-dimensional empirical distributions on the intrinsic red noise amplitudes and spectral indices for each pulsar.

To assess prior recovery, we use a quantile-quantile (Q-Q) plot. Q-Q plots compare the quantiles of the distribution of our recovered prior with samples drawn from a simulated distribution of the input. We subtract the mean of the simulated distribution from every point in our plot so that the mean falls along zero on the vertical axis. The expected result is a set of lines, one for each parameter, falling within the given uncertainties with few venturing outside the 3σ bounds. Bias could appear as a line remaining significantly above or below the mean for the entire interval indicating that more samples exist in one quantile of the distribution.

We use the same prior for each spectral index parameter γ and for each log amplitude parameter $\log_{10} A$,

$$\gamma \sim \text{Uniform}[0, 7], \tag{81}$$

$$\log_{10} A \sim \text{Uniform}[-18, -11]. \tag{82}$$

These priors are chosen specifically for this test and may be slightly different in "production grade" analyses between the intrinsic red noise amplitudes and the GWB amplitude, because we can rule out very large GWB amplitudes since they have not been observed in previous datasets. Spectral index priors are typically the same for all parameters. The NANOGrav 15-year GWB analysis uses 67 pulsars. Assuming a power-law spectrum, each pulsar adds a spectral index and an intrinsic red noise amplitude parameter. Along with the common process spectral index and amplitude parameters, each of the plots in Fig. 2 contains 68 lines. The majority of the lines remain inside three sigma

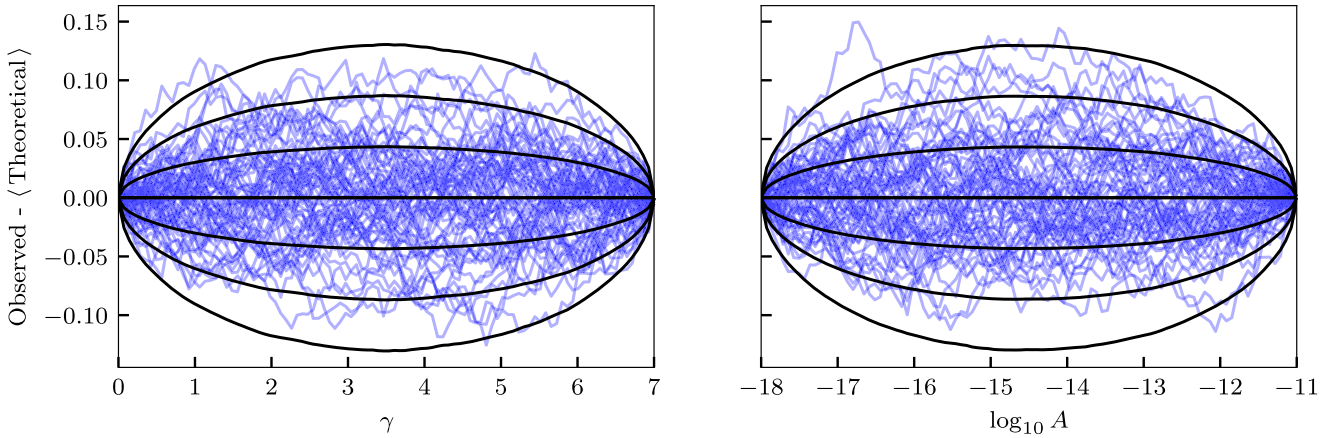


FIG. 2. Quantile-quantile plot showing the recovery of the prior. To produce this plot, we sample the prior in place of the likelihood and set the prior to a constant. This is not the same as setting the likelihood to a constant, because we use a parallel tempering method that only tempers the likelihood. Priors on the γ parameters are Uniform[0, 7], and the priors on the $\log_{10} A$ parameters are Uniform$[-18, -11]$. The chains produced should be equal to the input prior distribution within the sampling uncertainties after thinning. On the horizontal axis, we plot the parameter value associated with the quantiles of the simulated uniform distribution. On the vertical axis, we plot the parameters associated with the quantiles of the distribution output from the sampling process minus the mean of simulated parameters so that the mean lies along the zero of the vertical axis. The curved, solid lines show 1σ, 2σ, and 3σ uncertainties. The uncertainty lines are created by taking the average and standard deviation over 10,000 realizations of a uniform distribution with the same number of samples as the observed distribution. This plot shows that the priors were recovered correctly when using parallel tempering, AM, SCAM, DE, empirical, and prior proposal distributions.

uncertainty bounds and only occasionally do lines venture outside and then back toward zero.

B. Simulation recovery tests

Next, we create simulations to check that PTMCMCSampler returns unbiased posteriors. To quantify whether our posteriors are unbiased, we use a P-P plot.[8] We make these plots by creating 100 realizations of data, sampling the posteriors, and finding the p value at which the simulated value falls. These p values should follow a uniform distribution. By taking the CDF of these values and plotting them against the p values, we expect each parameter (shown as a line on the plot) to follow a diagonal line within some confidence interval. Since we model the red noise as a power law and have 67 pulsars, there are 136 parameters searched over in each simulation: an amplitude and spectral index for each pulsar's intrinsic red noise and an amplitude and spectral index for the common red process that is shared among all pulsars. For this P-P plot, we use the CURN model, because the computational expense is small compared to the HD correlated model, and the posteriors between the two models are similar. Furthermore, Hourihane *et al.* [72] found that reweighting the CURN plots using the HD model's likelihood also resulted in diagonal P-P plots. Given the similarity between the two models' posteriors, PTMCMCSampler should have no extra problems with exploring the HD parameter space given its effective exploration of the CURN parameter space.

We make simulations differently here than in other papers which have used either LIBSTEMPO [82] and TEMPO2 or PINT. The question of whether our models match the simulations of realistic PTA data is outside the scope of this study. Instead, we use the models in Enterprise to simulate pulsar residuals with timing model uncertainties, specified white noise, intrinsic red noise, and either a CURN or a HD correlated GWB. The priors set on these values are as shown in Table I. Crucially, the distributions which we draw values from and the priors we search over must be the same. If they are not, then the P-P test will fail. We have reduced the prior space from the full production analyses to limit ourselves to a detectable part of the parameter space. While this is not required to make good P-P plots, our purpose is to test whether we get

TABLE I. Priors sampled for the simulated values of simulation realizations. IRN indicates an intrinsic red noise parameter and CURN indicates a common uncorrelated parameter which is common to all pulsars. These priors are also used in the search of the simulations to recover posteriors for use in P-P tests.

Parameter	Prior	Values
IRN amplitude	Uniform	$[-15, -12]$
IRN spectral index	Uniform	[2, 6]
CURN amplitude	Uniform	$[-16, -14]$
CURN spectral index	Uniform	[2, 6]

[8]We could use Q-Q plots, which are quite general, for this too. However, P-P plots are somewhat standard among the literature for testing simulation recovery.

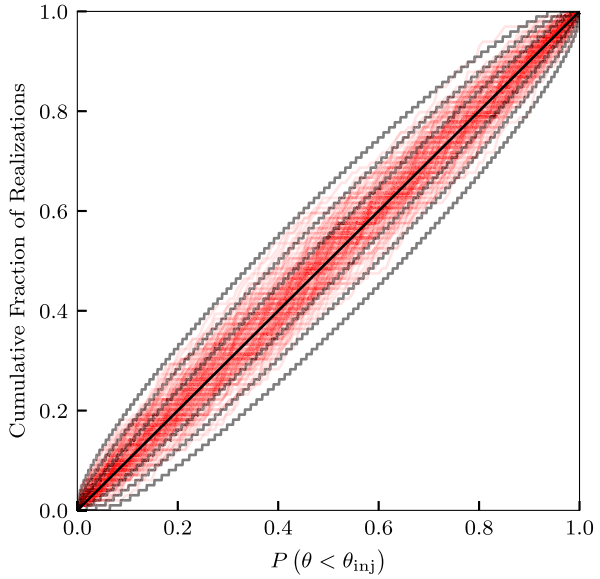


FIG. 3. A P-P plot shows the recovery of simulated parameters for 100 realizations simulated using the 15-year NANOGrav data with simulated values pulled from the priors in Table I. The model being simulated includes power law intrinsic red noise parameters and a power law CURN. Each of the 136 parameters are presented as an individual line on the P-P plot. A diagonal black line indicates perfect recovery, and the 68.27%, 95.45%, 99.73% confidence intervals, found using an inverse CDF of a binomial distribution [83], appear as curved black lines with 68.27% being the closest to the diagonal and 99.73% being farthest away. The plot indicates no significant bias in recovery of the posterior by PTMCMCSampler when the simulation and recovery is performed via Enterprise.

unbiased results in the event of a set of parameters that have strong signals visible.

The P-P plot produced in Fig. 3 indicates little bias in the recovery of the simulated values. Lines that remain above and below the diagonal could indicate that the recovery is biased so that we find the simulated values either too low or too high consistently. A signature "S" shape in which the line goes above (below) and then below (above) the diagonal indicates that the width of the posterior distribution is over or under estimated. After checking each of these 136 parameters individually, we find no evidence of either of these issues.

C. Bayes factor recovery tests

In the final segment of the Bayesian tests, we check how well our Bayes factors are recovered. Here, we use a few different techniques of computing the Bayes factors and compare between them. Due to computational limitations, we reduce our dataset to a set of 14 pulsars that have been timed for greater than 15 years. This allows us to use nested sampling that does not converge quickly for high dimensional spaces such as with the full 67 pulsar parameter space. Simulations were made with varying noise realizations and GWB amplitudes across a broad range of log Bayes factors,

$$\log_{10}\mathrm{BF} = \log_{10} Z_{\rm HD} - \log_{10} Z_{\rm CURN}, \tag{83}$$

from −2.5 to 17, where Z is the evidence. Bayes factors are computed for the HD correlated model against the CURN model. We check the hypermodel using PTMCMCSampler against nested sampling with Ultranest, and the hypermodel results against Bayes factors returned with reweighting [72]. By running these methods on the same 100 realizations with each method, we show that they return consistent answers, although at different levels of uncertainty.

In the comparison between reweighting and the hypermodel, we take the resulting chain from the hypermodel and reweight any of the uncorrelated model samples to the HD correlated model. This gives us a set of weights that can be used to compute a Bayes factor and uncertainties. We find that the two methods give results on the log ratio that are consistent with zero in every realization within 3σ, as shown in the bottom panel of Fig. 4. In every case, reweighting gives a larger uncertainty than the hypermodel and dominates in the subtraction of Bayes factors in which errors are propagated in quadrature.

Nested sampling requires a stopping condition in terms of the uncertainty on the log evidence. Unfortunately, we find that even with the reduced parameter space, setting the stopping condition to $d\log Z < 0.5$ led to week-long run times. Once again, the uncertainties on the hypermodel are overwhelmed by the uncertainties on nested sampling without requiring more computational resources for nested sampling, and all realizations are consistent with zero within 3σ, as shown in the top panel of Fig. 4.

As a final test of the Bayes factors, we run the hypermodel on a CURN model against itself on the same 100 simulations that were used above. In this case, we know that the Bayes factor must equal 1, because a model should not be preferred over itself. On top of checking whether we get an answer consistent with the known value, this test shows whether the uncertainties are being estimated properly. As shown in Fig. 5, the Bayes factor of 1 is recovered in every realization within 3σ. This method represents an easy check that can be performed for any situation to make sure that the hypermodel calculation is working. However, this test is not sufficient to claim that the method will work for all scenarios. The case of a model against itself does not take into account the situation where the posteriors between two models are very different. Therefore, the test of consistency against other samplers remains necessary.

VI. TESTS OF FREQUENTIST METHODS: `enterprise_extensions` AND THE OPTIMAL STATISTIC

In this section we present a series of tests on the optimal statistic presented in Sec. IV. We begin by

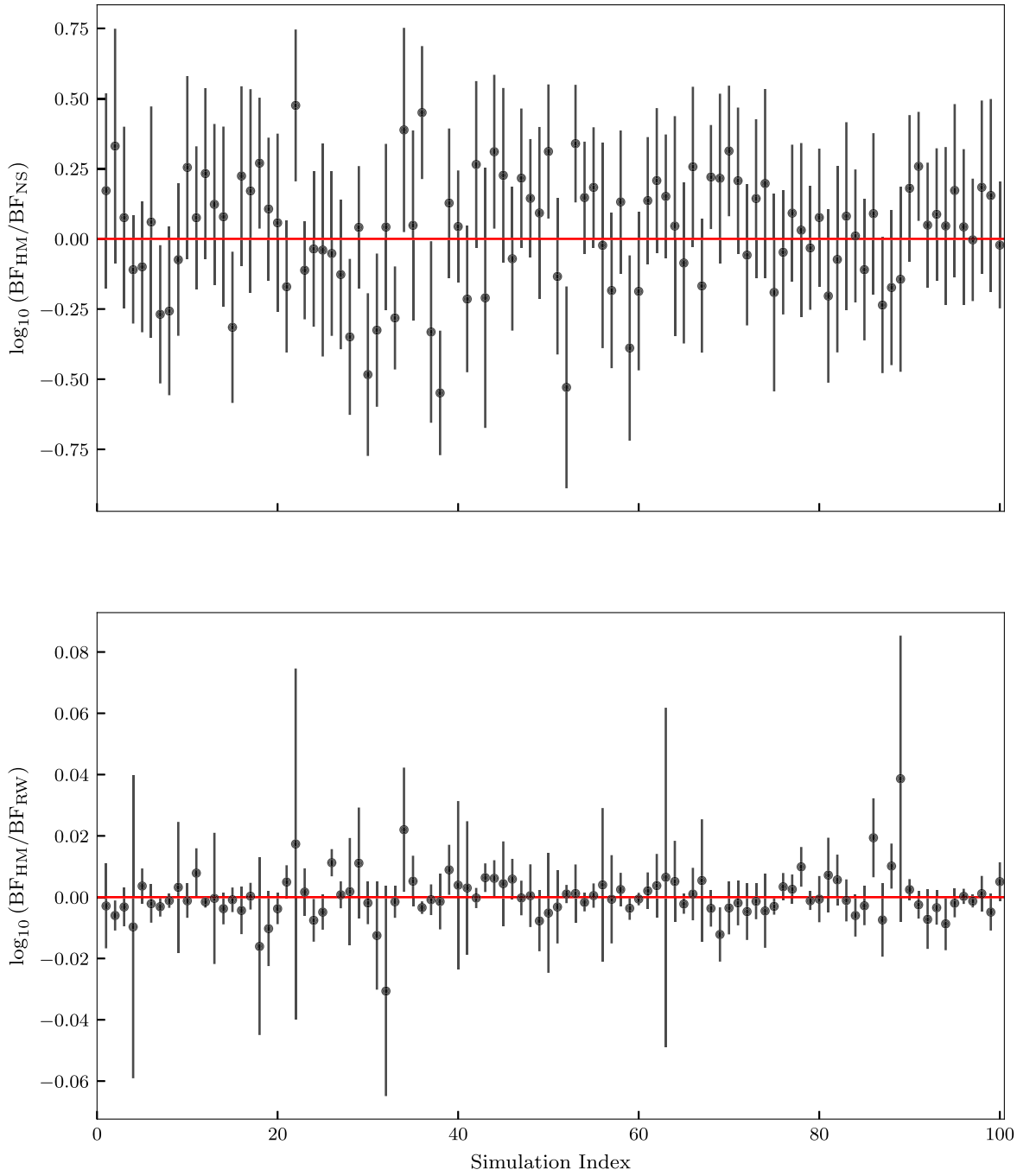


FIG. 4. Logarithmic ratio of the Bayes factor computed for each of 100 simulations using the hypermodel framework and nested sampling in the top panel and the hypermodel framework and reweighting in the bottom panel. Each point indicates a mean value of the ratio and the 1σ uncertainties are given as a vertical bar on each point. The red line indicates zero on the vertical axis, where we expect these values to fall if the Bayes factors returned from each method are consistent. Uncertainties are dominated by the nested sampling and reweighting methods. The values are consistent with zero within 3σ across all simulations.

using simulated datasets to compare the "traditional" optimal statistic and the one that accounts for covariance between pulsar pair correlations. We use those same datasets to evaluate how the revised binned estimator performs as well. This is followed by a discussion of the distinction between using these statistics as estimators for the amplitude of the GWB vs using them as detection statistics in a classical null hypothesis testing scenario. We finish by summarizing recent work in the literature on the MCOS and constructing empirical and analytic distributions for the optimal statistic.

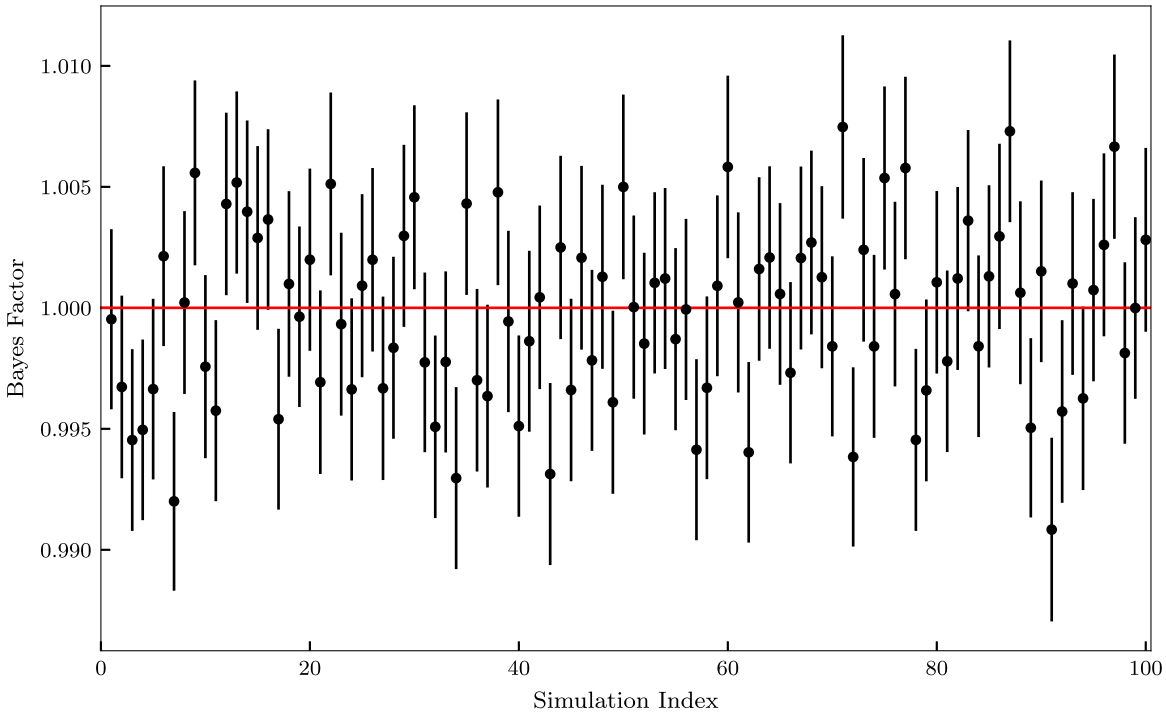


FIG. 5. Bayes factors computed for a CURN model against itself for the same 100 simulations as used in the other tests of Bayes factors. The red line in this figure indicates the true Bayes factor between the two models. All realizations recover the true value within 3σ. This represents a quick check to see if the Bayes factor calculation using a hypermodel returns correct answers in the ideal scenario of posteriors that are exactly the same between the two models being sampled.

A. The optimal statistic as an estimator

In the case where $A^2_{\rm gw}$ is small compared to intrinsic red noise in all pulsars, the distribution on $\hat{A}^2_{\rm gw}$ in Eq. (65) is approximately Gaussian with a variance given by Eq. (65) except in the tails [79]. In present analyses, $A^2_{\rm gw}$ is not smaller than the corresponding intrinsic red noise amplitude for at least a few pulsars [19,20]. Therefore, to test how well the small signal approximation works, we perform 200 simulations with the length and cadence of the dataset given in [47] and with red noise drawn from $p(\boldsymbol{\eta}|\boldsymbol{\delta t})$ from [8]. We draw $A_{\rm gw}$ from a uniform distribution, $\log_{10}A_{\rm gw,inj} \in [-17,-13]$. On each simulated dataset we calculate $\hat{A}^2_{\rm gw}$ and its variance using Eqs. (65) and (66), and we use the method described in Appendix C to construct P-P-like plots, replacing the posterior samples with a Gaussian, $\mathcal{N}(\hat{A}^2_{\rm gw}, \sigma^2_{\rm gw})$. These are not traditional P-P plots, as there is no well-defined prior from which we draw our simulations and sample our posterior. However, performing the same processing as in Appendix C does test how frequently the Gaussian distribution centered on $\hat{A}^2_{\rm gw}$ with variance $\sigma^2_{\rm gw}$ includes the simulated value of $A^2_{\rm gw}$. We do the same thing for the estimator that includes covariances between pulsar pairs, defined in Eqs. (71) and (72).

The results of this test are shown in Fig. 6, with the "traditional" optimal statistic results shown in the blue, solid curve and the corrected results shown in the orange, solid curve. The blue curve is consistent with the traditional optimal statistic underestimating the error on the estimator by not accounting for the GWB, and therefore not capturing the simulated value in its credible intervals as frequently as it should. The orange curve corrects this, as it follows the expected line more closely. Therefore, if one is to use the optimal statistic as an estimator for the GWB, the corrected statistic performs significantly better.

We can perform the same test for the binned optimal statistic in Eqs. (73) and (74) as well. For each simulation, we evaluate the cumulative distribution function at the simulated value, similar to the test in Appendix C, but on each individual binned estimator in Eqs. (73) and (74). We plot the results of the P-P-like test in Fig. 7, where we see similar results to Fig. 6. In this case, we show deviations from the predicted line, and so expectation is zero, as opposed to $y = x$. The blue curves show excess cases where the CDF is zero and close to one, indicating misestimation. The orange curves, which includes the GWB in its variance, perform better, especially near zero and one. The orange curves do reach the 3-sigma level more than one might normally expect, and this is likely because we have assumed that the distribution on the binned estimators is a Gaussian. In practice, the distribution on the estimators is a generalized chi-squared distribution, which can be well approximated by a Gaussian under certain circumstances [79].

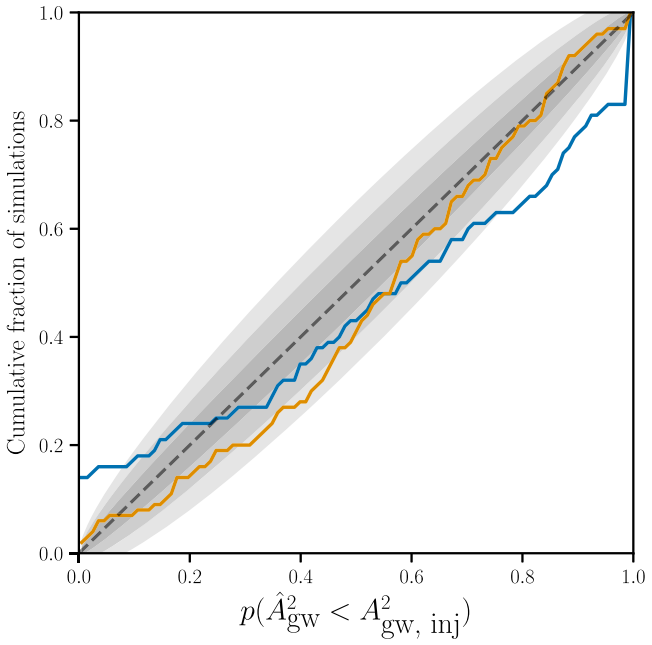


FIG. 6. P-P-like plot that characterizes how well the optimal statistic functions as an estimator for the amplitude of the GWB. The blue, solid curve, which uses the traditional OS does not follow the expected diagonal line, indicating that it underestimates the variance on the estimator for the background. The orange curve, meanwhile, follows the diagonal line with its expected error bars (shaded region) because it properly estimates the variance by including the contribution from the GWB.

In these simulations we have used the simulated amplitude of the background when calculating the covariance between correlations. We do this for practical purposes—the goal of these P-P-like tests is to show that our estimator is unbiased and its error bars are correct when we take into account the amplitude of the GWB. In practice, we do not have access to the GWB amplitude *a priori*—we either use values drawn from a MCMC chain (e.g., the noise-marginalized optimal statistic), or we could employ an iterative approach, where we calculate the GWB estimator using the optimal statistic, and then use the estimator in the covariance matrix to properly estimate the covariance matrix from the correlations, and then repeat until convergence.

B. The optimal statistic for detection

We have shown that the estimators which include the strength of the GWB in their construction are better than the ones that do not, especially in the case where the amplitude of the GWB is of a similar size to that of the intrinsic red noise. However, the S/N calculated in Eq. (67) is calculated under the null hypothesis. Therefore, when making a detection, we construct a distribution for this statistic under the null hypothesis (i.e., no correlated power), and then we compare the same statistic calculated on the original data to

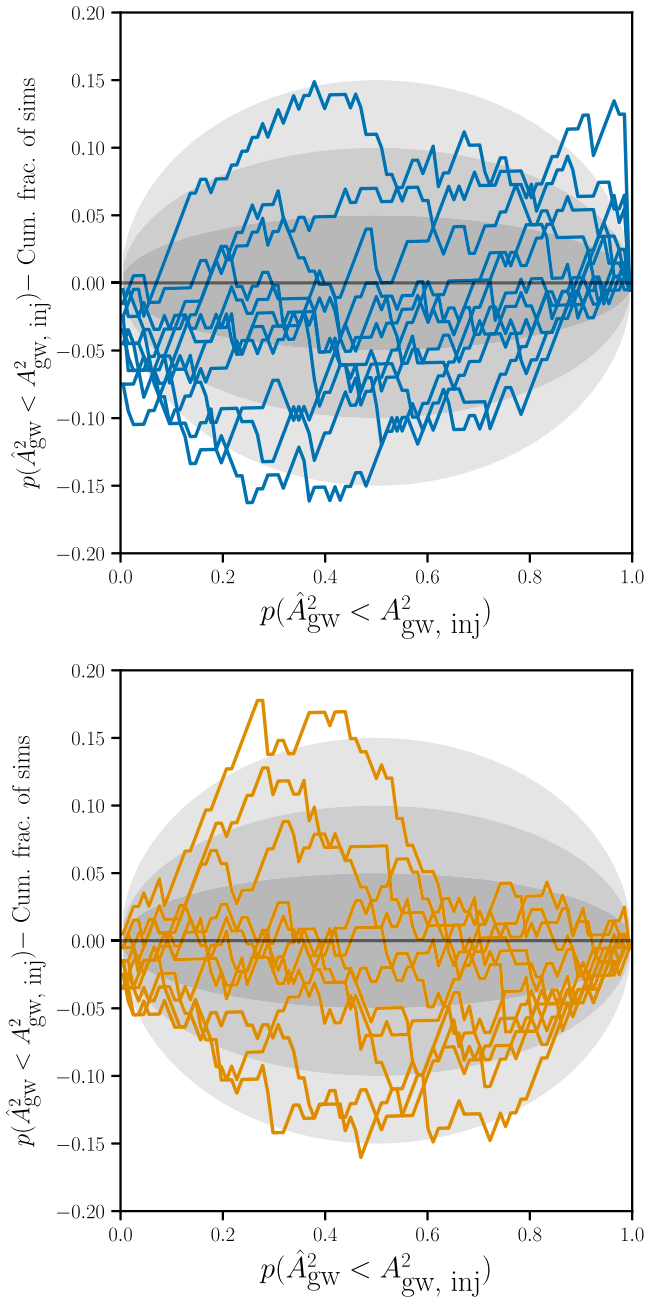


FIG. 7. P-P-like plots (with diagonal subtracted off) that characterize how well the angular-binned optimal statistic functions as an estimator for the amplitude of the GWB, modulated by the Hellings-Downs correlations. We show results estimating the amplitude in each of 11 individual bins. Top: the blue, solid curve, uses the traditional binned OS does not follow the expected horizontal line, indicating that it underestimates the variance on the estimator for the background. This is especially obvious looking near zero and near one, where we see the curves diverge from the expected range (shaded regions) Bottom: the orange curve follows the horizontal line better than the blue curves in the top because it properly estimates the variance by including the contribution from the GWB.

that distribution. Therefore, Eq. (67) is the correct expression for a detection statistic. It is common to use a noise-marginalized version of this statistic, as discussed previously. That is, we take the average of this S/N over many draws from $p(\boldsymbol{\eta}|\boldsymbol{\delta t})$ and compare this to a null distribution.

Construction of a distribution for the null statistic takes a few forms. The analytic distribution of this statistic was calculated in [79] and can be used. However, pathologies in the data that are not correctly modeled would not be accounted for in an analytic calculation. To preserve potential mismodeling, but still approximate the null distribution of the detection statistic, it is common to use sky scrambling [80] and phase shifting [81]. Sky scrambling involves assigning random sky positions to each pulsar, and calculating the optimal statistic, while phase shifting applies random phases to each frequency and each pulsar in the $\boldsymbol{F}$ matrix used to construct Eq. (64). Comparisons between the analytic distribution and distributions constructed by sky scrambling and phase shifting are still in progress. Additionally, some concerns have been raised about whether performing sky scrambling and phase shifting without producing independent[9] scrambles or shifts results in oversampling parts of the null distribution, and undersampling other parts [84]. Some tests of this have been done to explore this in [19] and found that more stringent "orthogonality" conditions do not produce meaningfully different distributions than those calculated with less stringent conditions. However, more tests are in progress.

C. MCOS tests

We do not present novel tests of the MCOS here, as tests have been presented in separate work that more completely presents the statistic itself and characterizes its behavior [38]. We summarize those results here, for completeness.

In [38,85] it is shown that, due to the nonisotropic distribution of pulsars on the sky, Hellings-Downs correlations are not orthogonal to monopolar or dipolar correlations, as one might expect. This is what motivates the construction of a statistic that simultaneously fits for multiple spatial correlation patterns simultaneously. In [38], they perform simulations of non-Hellings-Downs correlation patterns, and show that it is possible using the statistic in Eq. (67) to find a spurious detection of Hellings-Downs correlations. Using Eqs. (77) and (80), remedies this situation, as the new statistic correctly finds that the non-Hellings-Downs correlations are preferred. Additionally, the authors consider the MCOS as an estimator for the strength of the background associated with each spatial correlation pattern. They find that, when multiple correlation patterns are simulated into the data, the MCOS estimator for the strength of each pattern performs better than individually estimating the strength of each process separately. However, there are issues with the estimation of signals and their uncertainties owing to the fact that the MCOS is based on the traditional optimal statistic, which does not take into account covariance between pulsar pairs (see Sec. VI A).

[9]A statistic one could use to assess independence is presented in [84].

VII. CONCLUSIONS

Here we have checked both Bayesian and frequentist methods used in the 15-year GWB analysis. These methods were outlined as they are used in the 15-year analysis. We subjected the method implementations of each part of the analysis to the most stringent tests of correctness performed so far, and we find that each analysis returns unbiased, self-consistent results.

In the Bayesian tests, we find that the priors are recovered correctly while including parallel tempering, AM, SCAM, DE, empirical distributions, and prior proposal distributions that are used in the full analysis. This indicates that all proposal distributions work properly and are not biasing our results. We also find that the simulations made with simulations pulled from a prior distribution return diagonal P-P plots within acceptable uncertainties. By creating realizations of the 15-year data with various amplitudes and spectral indices of a GWB including HD correlations in them, we perform tests using reweighting, nested sampling, and the hypermodel to return Bayes factors between an HD correlated and CURN model. Each comparison between these methods return log Bayes factor ratios consistent with zero. Finally, we perform model comparison between a model and itself on these same simulations and find the expected result of Gaussian distributed Bayes factors centered around unity.

In the frequentist tests, we show through simulating datasets that properly including the GWB when constructing the optimal statistic is necessary in the situation where the GWB is not small compared to the intrinsic red noise. The estimator that accounts for the size of the GWB, however, is not consistent with the null hypothesis of no correlated power in the timing residuals, and therefore the "traditional" optimal statistic should be used as a detection statistic. Such a detection statistic should be calculated on the observed timing residuals and compared to a null distribution, which can be calculated either using the analytic distribution [79] or a method that preserves potential mismodeling but suppresses correlations [80,81]. Finally, we summarize the recently proposed MCOS, which simultaneously fits for multiple spatial correlation patterns, and helps prevent, e.g., monopolar correlations from producing a spurious detection of Hellings-Downs correlations and vice versa.

Currently, the NANOGrav software collection allows one to perform inference without a steep learning curve. As our data volume and parameter space increase with each dataset, we look for additional methods to improve

efficiency of memory management and the likelihood evaluation speed in Enterprise. This includes possible data compression techniques to reduce the number of points of data required to fit to perform inference (e.g., [86]). Additionally, reducing the autocorrelation of our chains produced with PTMCMCSampler could reduce computation time significantly. To this end, some recent work has used JAX [87], a Python package that includes just-in-time compilation to speed up evaluation of loops and autodifferentiation to take derivatives quickly and accurately, among other convenient features. With this, Freedman *et al.* [88] were able to use a No-U-Turn sampler, a Hamiltonian Monte Carlo sampling technique, which reduced autocorrelation significantly compared to PTMCMCSampler. In Bécsy *et al.* [89], just-in-time compilation is used through the NUMBA Python package alongside techniques to sample some parameters more often than others to vastly increase the speed of evaluation of continuous GW searches. While these papers have shown the incredible speed gains that one may achieve via these new technologies, they cannot be easily implemented in the current paradigm used by Enterprise. This is primarily due to subclassing of NumPy arrays used in Enterprise, which is not supported by these other programs.

Both the Bayesian and frequentist methods and their software representations pass all the tests they were subjected to. As PTA datasets increase in sensitivity, testing that our methods are reliable proves paramount. Here, we validate the results of current and future analyses that use the methods examined. Through rigorous testing of our software and methods, we form a foundation on which astrophysical results may stand.

ACKNOWLEDGMENTS

The NANOGrav project receives support from National Science Foundation (NSF) Physics Frontiers Center Award No. 1430284. A. D. J., K. C., and M. V. acknowledge support from the Caltech and Jet Propulsion Laboratory President's and Director's Research and Development Fund. A. D. J. and K. C. acknowledge support from the Sloan Foundation. S. H. is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1745301. L. B. acknowledges support from the National Science Foundation under Award No. AST-1909933 and from the Research Corporation for Science Advancement under Cottrell Scholar Award No. 27553. P. R. B. is supported by the Science and Technology Facilities Council, Grant No. ST/W000946/1. S. B. gratefully acknowledges the support of a Sloan Fellowship, and the support of NSF under Award No. 1815664. M. C. and S. R. T. acknowledge support from Grant No. NSF AST-2007993. M. C. and N. S. P. were supported by the Vanderbilt Initiative in Data Intensive Astrophysics (VIDA) Fellowship. Support for this work was provided by the NSF through the Grote Reber Fellowship Program administered by Associated Universities, Inc./National Radio Astronomy Observatory. Support for H. T. C. is provided by NASA through the NASA Hubble Fellowship Program Grant No. HST-HF2-51453.001 awarded by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., for NASA, under Contract No. NAS5-26555. M. E. D. acknowledges support from the Naval Research Laboratory by NASA under Contract No. S-15633Y. T. D. and M. T. L. are supported by an NSF Astronomy and Astrophysics Grant (AAG) Award No. 2009468. E. C. F. is supported by NASA under Award No. 80GSFC21M0002. G. E. F., S. C. S., and S. J. V. are supported by NSF Award No. PHY-2011772. The Flatiron Institute is supported by the Simons Foundation. The work of N. L. and X. S. is partly supported by the George and Hannah Bolinger Memorial Fund in the College of Science at Oregon State University. N. L. acknowledges the support from Larry W. Martin and Joyce B. O'Neill Endowed Fellowship in the College of Science at Oregon State University. Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (Grant No. 80NM0018D0004). M. A. M. is supported by NSF Grants No. 1458952 and 2009425. C. M. F. M. was supported in part by the National Science Foundation under Grants No. NSF PHY-1748958 and No. AST-2106552. A. Mi. is supported by the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy—EXC 2121 Quantum Universe—Grant No. 390833306. K. D. O. was supported in part by NSF Grant No. 2207267. T. T. P. acknowledges support from the Extragalactic Astrophysics Research Group at Eötvös Loránd University, funded by the Eötvös Loránd Research Network (ELKH), which was used during the development of this research. S. M. R. and I. H. S. are CIFAR Fellows. Portions of this work performed at N. R. L. were supported by ONR 6.1 basic research funding. J. D. R. also acknowledges support from start-up funds from Texas Tech University. J. S. is supported by an NSF Astronomy and Astrophysics Postdoctoral Fellowship under Award No. AST-2202388, and acknowledges previous support by the NSF under Award No. 1847938. Pulsar research at U. B. C. is supported by an NSERC Discovery Grant and by CIFAR. S. R. T. acknowledges support from an NSF CAREER Award No. 2146016. C. U. acknowledges support from BGU (Kreitman fellowship), and the Council for Higher Education and Israel Academy of Sciences and Humanities (Excellence fellowship). C. A. W. acknowledges support from CIERA, the Adler Planetarium, and the Brinson Foundation through a CIERA-Adler postdoctoral fellowship. O. Y. is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2139292. J. A. C. C. was supported in part by NASA CT Space Grant PTE Federal Award

No. 80NSSC20M0129, and also supported in part by the National Science Foundation's NANOGrav Physics Frontier Center, Award No. 2020265. H. T. C. has NASA Hubble Fellowship: Einstein Postdoctoral Fellow. R. J. J. and J. K. S. is NANOGrav Physics Frontiers Center Postdoctoral Fellow.

APPENDIX A: STRUCTURE OF Enterprise

The Enhanced Numerical Toolbox Enabling a Robust PulsaR Inference SuitE (Enterprise) [41] is a Python package built to analyze pulsar noise and timing models, and to search for GWs in PTA data. It grew out of previous PTA data analysis packages such as NX01 [90], PAL2 [91], and PICCARD [92]. In the context of this paper, we are using Enterprise to search for an isotropic GWB in simulations of the NANOGrav 15-year dataset. Development of Enterprise began in the interest of making a suite of tools that could analyze data from any PTA consortium, and support international collaboration without requiring significant knowledge of programming.

Enterprise is a thoroughly object-oriented package that defines a PTA data model as a hierarchical structure (see Fig. 8). The top-level object is PTA, which interfaces with the sampler by way of the `get_lnlikelihood` and `get_lnprior` methods: these evaluate the (log) Gaussian-process-marginalized likelihood of Eq. (24) and the total prior respectively, taking as input a Python dictionary of parameter values, or alternatively a NumPy vector with the parameters in the same order as in the property `PTA.params`.

The PTA object is created from a sequence of `SignalCollection` objects, each of which corresponds to a pulsar in the array, and represents the pulsar's complete data model. `SignalCollection` implements a set of methods that return the vector and matrix constituents used by `PTA.get_lnlikelihood`: for instance, `get_residuals` for the residuals $\boldsymbol{\delta t}$, `get_delay` for any deterministic delays, `get_ndiag` for the white-noise matrix $\boldsymbol{N}$, `get_basis` and `get_phi` for Gaussian-process bases $\boldsymbol{T}$ and prior matrix $\boldsymbol{B}$. All these methods take a Python dictionary of parameter values.

Each `SignalCollection` object consists of one `Pulsar` object and any number of `Signal` objects. `Pulsar` uses the PINT or LIBSTEMPO packages to read pulsar data from `par` and `tim` files, standard formats for storing timing model parameters and timing data (see, e.g., [93]), and it provides the vectors `residuals` (for $\boldsymbol{\delta t}$), `toaerrs` (for $\boldsymbol{\sigma}$), and `freqs` (for the pulse radio frequencies), as well as the timing-model design matrix `Mmat` (i.e., $\boldsymbol{M}$), and several more pulsar properties. The `Signal` objects, which come in a variety of subtypes, implement deterministic signals and noise components, and provide the building blocks that `SignalCollection` and (downstream) PTA assemble into a full likelihood. Specifically, deterministic signals define `get_delay`, white-noise components define `get_ndiag`, and Gaussian processes define `get_basis` and `get_phi`. Common Gaussian processes, used most notably for the HD-correlated background, have parallel bases for each pulsar (with the same n_{GP} but different n_{obs}), provide a `get_phicross` method in addition to `get_phi` to fill

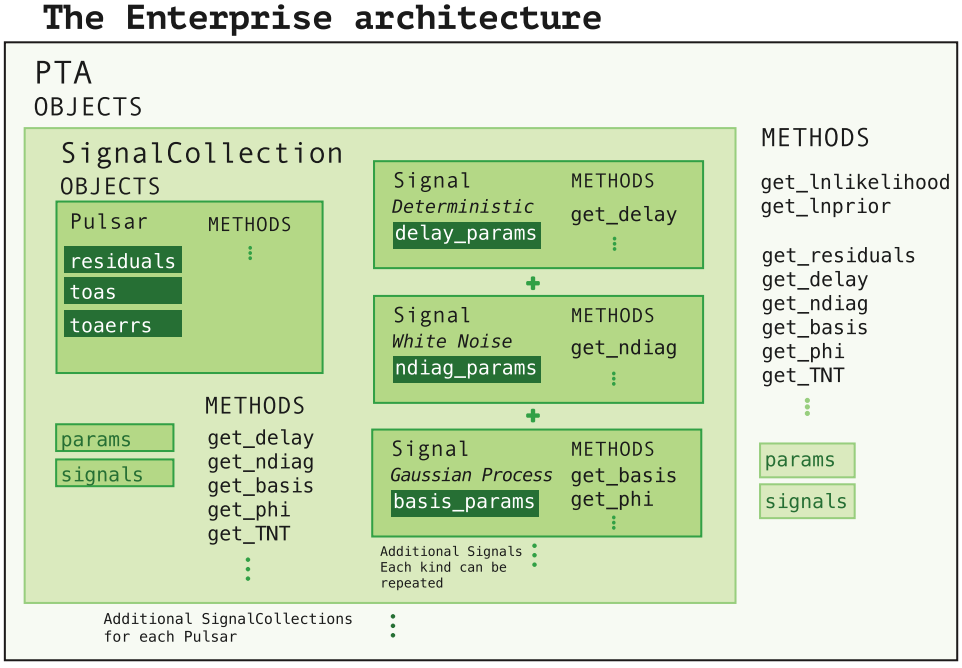


FIG. 8. The hierarchical structure of Enterprise.

off-diagonal prior-matrix elements and are meant to be used with common hyperparameters.

The package includes a number of optimizations for speed, memory, and ease of configuration:

(1) All objects keep track of which parameters affect the output of their methods, and cache results if those parameters are not changed, or are set as constants. This is useful, for instance, in analyses that freeze the white-noise parameters, or in stochastic schemes that update parameters in blocks. The `SignalCollection` and PTA objects have methods for intermediate matrix combinations such as $\boldsymbol{T}^T\boldsymbol{N}^{-1}\boldsymbol{T}$, and these are also cached on the basis of the relevant parameters.

(2) The `SignalCollection` object combines delays from all deterministic signals into a single vector, assembles white-noise matrices into a single matrix, and stacks Gaussian-process bases and priors into two combined matrices. The object has the ability of reusing basis vectors that appear identically in multiple processes, such as Fourier vectors for red noise and the GWB.

(3) The `get_ndiag` methods return custom "kernel" objects that represent the constituents of the measurement-noise matrix $\boldsymbol{N}$. Whether the kernels are represented internally as a vector for EFAC/EQUAD noise, a Sherman-Morrison decomposition for ECORR [see Eq. (28)], or a sparse matrix for even more general cases, they all provide "solve" methods that return combinations such as $\boldsymbol{T}^T\boldsymbol{N}^{-1}\boldsymbol{y}$ and $\boldsymbol{T}^T\boldsymbol{N}^{-1}\boldsymbol{T}$ without actually computing and storing $\boldsymbol{N}^{-1}$. Furthermore, the kernel objects know how to combine themselves with other kernel objects, creating an optimized object used in the likelihood calculation.

(4) The PTA object defines an optimized `get_phiinv` method that can choose between multiple inversion strategies to build the global $\boldsymbol{B}^{-1}$ depending on the structure of $\boldsymbol{B}$.

Enterprise is configured by building a `SignalCollection` template from a sequence of `Signal` templates. Enterprise includes "factories" that build templates for commonly used noise components such as `MeasurementNoise` (for radiometer noise), `EcorrKernelNoise` (for jitter-like noise), `TimingModel` (for the Gaussian process with improper prior used to marginalize over timing-model corrections), `FourierBasisGP` (for Gaussian processes with a sine/cosine Fourier basis), and `FourierBasisCommonGP` (for Fourier Gaussian processes with correlations across pulsars), and more. Each template must be assigned one or more `Parameter` objects that encode the priors chosen for the relevant parameters. For instance, `MeasurementNoise` may be passed `efac = Normal(1, 0.25)` and `log10_t2equad = Uniform(-8.5, -5)` to indicate that in Eq. (6) $F \sim \mathcal{N}(1, 0.25)$ and $\log_{10} Q \sim U(-8, -5)$. Parameters may also be passed as `Constant` if they will not be sampled stochastically.

The `SignalCollection` template is then applied to each `Pulsar` object in turn, generating instantiated `SignalCollection` and `Signal` objects, in which parameters are specialized to the pulsar (e.g., `efac` becomes `B1855+09_efac`) and the `Pulsar` quantities are made available locally. For most `Signals`, a further kind of specialization is possible in which a selection function is passed to the `Signal`, effectively splitting it into multiple copies, each applied with different parameters to a different section of the data. For instance, the selection `by_backend`, when given to `MeasurementNoise`, would split EFAC parameters for the NANOGrav B1855+09 pulsar into `B1855+09_430_ASP_efac`, `B1855+09_430_PUPPI_efac`, `B1855+09_L-wide_ASP_efac`, `B1855+09_L-wide_PUPPI_efac`, each applied to the subset of the residuals obtained with that receiver backend.

Configuration is completed by collecting the set of instantiated `SignalCollection` into a single PTA object. See the NANOGrav 12.5-yr and 15-yr tutorials for examples of creating a standard PTA object that is ready to compute log likelihoods and priors.

Enterprise includes a number of additional facilities and utilities that simplify the task of creating a model. For instance, the prior for a Fourier Gaussian processes can be written simply as Python functions that take a vector of frequencies and return the diagonal components of $\boldsymbol{\phi}$. By wrapping the prior in `parameter.function`, the arguments of the Python function are automatically book kept as model parameters and specialized to pulsar and selections. Furthermore, if the Python function includes arguments that are defined in the `Pulsar` object (such as `residuals` or `toaerrs`), these are passed to the function automatically once the `Signal` that uses the function has been specified.

Of course, Enterprise already defines a number of commonly used Gaussian-process priors and ORFs. It also implements the deterministic model of solar-system-ephemeris uncertainties [37] used for older NANOGrav data releases (`utils.FourierBasisCommonGP_physicalephem`); it can draw random values of model parameters according to their prior (`parameter.sample`); it can sample the conditional distributions of Gaussian-process weights given their hyperparameters (`utils.ConditionalGP`); it can simulate full realizations of the data model (`simulate`); it can handle the faster likelihood of Sec. II B with a special noise kernel `MarginalizingTimingModel` that holds the components of $\boldsymbol{D}$ internally and performs the Woodbury inversion of Eq. (34) transparently. See the Enterprise documentation for this and much more.

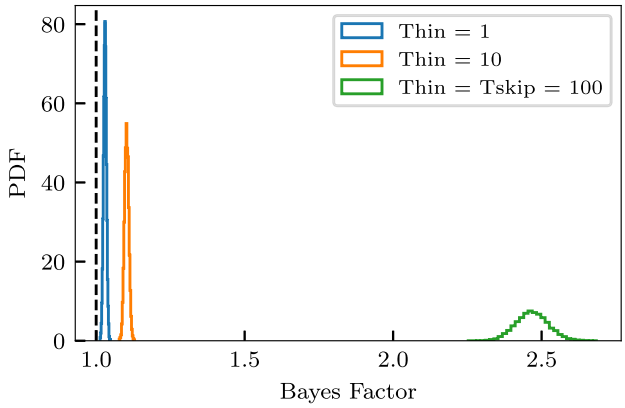


FIG. 9. The Bayes factor for a model over itself should be one, which is shown as a dashed, black line on this plot. Instead, we find that the Bayes factor is inconsistent with a value of one. Parallel tempering swaps are proposed every 100 samples, labeled "Tskip" in this plot (as it is in PTMCMCSampler). Thinning increases the contamination from swaps by increasing the number of samples that come from swaps out of the total number. Uncertainties on the Bayes factors were computed via bootstrapping.

APPENDIX B: ASYNCHRONOUS PARALLEL TEMPERING AND THE HYPERMODEL FRAMEWORK

PTMCMCSampler previously used an asynchronous parallelization model for parallel tempering in which each chain sampled a set number of steps on its own unless interrupted by another chain asking to propose a swap between chains. During our tests, we noticed that this asynchronous parallel-tempered MCMC biased Bayes factors computed with hypermodel in favor of the model that took the longest evaluation time per iteration. As a simple example of this issue, we simulate data with a sinusoidal signal $h(t)$ and noise $n(t)$,

$$\mathbf{d}(t) = \mathbf{h}(t) + \mathbf{n}(t). \tag{B1}$$

The sinusoid signal is described by amplitude, angular frequency, and phase,

$$h_i(t, A, \omega, \phi) = A \sin(\omega t_i + \phi), \tag{B2}$$

and the noise $n(t) \sim \mathcal{N}(0, 1)$. The log likelihood we use for this signal model is

$$p(d|A, \omega, \phi) \propto -\frac{1}{2}\sum_i (d_i(t) - h_i(t, A, \omega, \phi))^2. \tag{B3}$$

Priors are all chosen to be uniform with

$$A \sim \text{Uniform}[0, 5], \tag{B4}$$

$$\omega \sim \text{Uniform}[0, 3], \tag{B5}$$

$$\phi \sim \text{Uniform}[0, \pi]. \tag{B6}$$

Priors remain the same across each model used, and parameter estimation of the simulated values returns consistently with the posteriors.

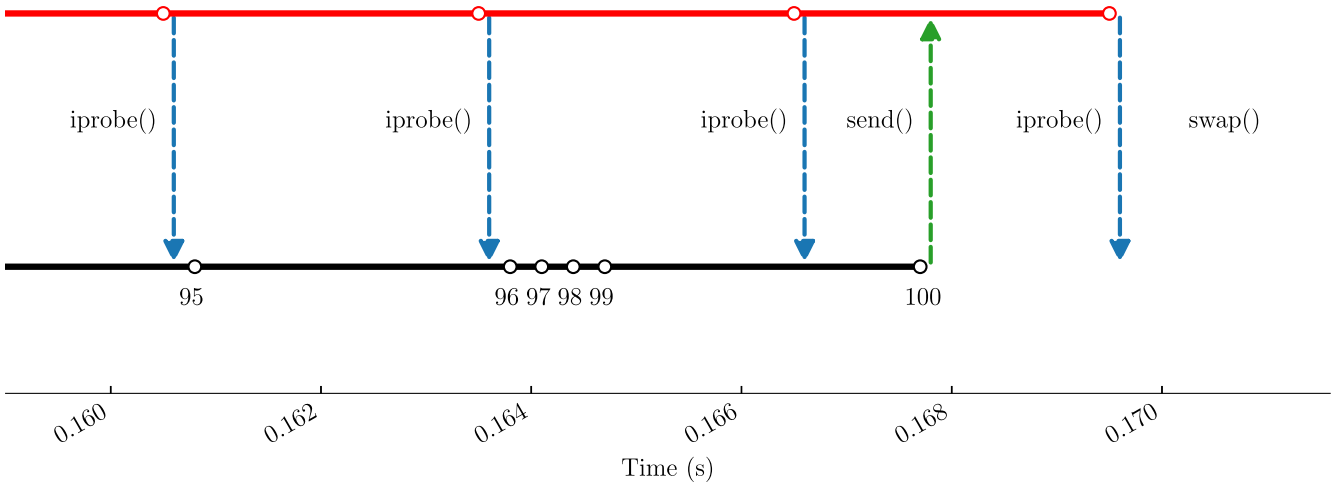


FIG. 10. Two chains, one at $T_1 = 1$ and the other at $T_1 = 2$, progress through iterations. Iterations start at the white circles and last until the next circle. Two models are being considered, one of which takes 10 times longer to evaluate. After each iteration, the hot chain, shown in red on top, runs a nonblocking probe to check if anything has been sent from the cold chain, shown in black on bottom. Once the cold chain reaches 100 samples, a blocking send pushes data required for a swap to the hot chain. These data get accepted by the hot chain after the current iteration finishes and a swap is proposed. Because of this disparity in evaluation times, we find that the hot chain is 10 times more likely to be in the model that takes longer to evaluate. This results in a bias in the model that goes into the swap proposals. Fundamentally, this means that the swaps proposed are more likely to happen in certain parts of the hypermodel parameter space, breaking detailed balance.

The hypermodel framework consists of multiple models concatenated with a continuous "switch" parameter between them. This framework chooses models based on which of two bins the switch parameter falls into. In this case, the first model is the sinusoidal model as described above, and the second model is the same model, but with a `time.sleep()` call to increase the evaluation time by a factor of a few.

Upon computing the odds ratios with the asynchronously updated sampler, we find that the odds ratio is not consistent with the anticipated value of one, as seen in Fig. 9. Parallel tempering swap proposals occur every 100 samples. Thinning by multiples of 10 cause the samples to become increasingly contaminated with swaps that pull the odds ratio away from one.

Syncing the model evaluation times gives odds ratios consistent with one, contrary to what we see in Fig. 9. Therefore, the problem appears to be caused by the evaluation time of one of the models being much longer than the other. The exact timeline of when these swaps are proposed is shown in Fig. 10 and proves critical to figuring out what went wrong.

In asynchronous parallel tempering, swaps are proposed whenever a chain gets to a set number of samples. The chain sends data to the next chain up in the temperature ladder, halting the chain until the other chain is done with its current sample evaluation. While waiting for this signal, the hotter chain continues collecting samples, probing the lower chain for data after each sample. Once the signal has been received, the chains swap with their most recent sample iteration.

Upon proposing a swap, the cold chain finds the hot chain in the model that has the longer evaluation time more often than not. This means that the swaps are proposed with a dependence on where in the switch parameter space we are, violating detailed balance for parallel tempering swap proposals.

One option to fix this involves synchronizing the swap proposals so that all chains propose swaps at the same iterations. The other option is to weight the proposals with weights proportional to the ratio of the evaluation times. In some situations this option is not possible due to dependence of evaluation time on where we are in the parameter space. We opted to synchronize the sampler for simplicity and to keep the sampler as generic as possible.

APPENDIX C: UNDERSTANDING CONFIDENCE INTERVALS OF P-P PLOTS

In Fig. 3, the confidence intervals are found as in Ashton and Talbot [83]. Unlike in the Q-Q plots where we have tens of thousands of samples, the standard bootstrap approximated confidence intervals (which estimates the confidence intervals as Gaussian) overshoot the exact bounds near p values of 0 and 1. For a given realization of the data, we have a simulated value for a single parameter $\theta_{\rm inj}$, and we can compute a CDF of this value,

$$F(\theta_{\rm inj}) = \int_{-\infty}^{\theta_{\rm inj}} p(x|\boldsymbol{\delta t})dx. \tag{C1}$$

If we take many new realizations, their associated CDFs should be uniformly distributed between 0 and 1. Picking a particular p value on the horizontal axis will split the uniform distribution into two segments. Let us define a success as $F(\theta_{\rm inj}) \leq s$ and a failure as $F(\theta_{\rm inj}) > s$, where s is the chosen horizontal axis value. From N draws, the probability of k successes can be found with a binomial distribution,

$$p(k\,\text{successes}) = \binom{N}{k} s^k (1-s)^{N-k}. \tag{C2}$$

We would like to have coverage $(1-\alpha)/2$ on both sides of the mean of the distribution with

$$\alpha = (0.6827, 0.9545, 0.9973) \tag{C3}$$

for the three bounds that we show. Using the quantile function for the binomial distribution, $F^{-1}(s)$ (`binom.ppf` in `scipy.stats`) on each of the horizontal axis values, the offset from the mean is

$$\sigma = F^{-1}(s)/N. \tag{C4}$$

APPENDIX D: BATCH UPDATES OF THE SAMPLE MEAN AND COVARIANCE

Let $\boldsymbol{x_0}$ be an $N \times n_{\rm param}$ matrix of samples from the history of an MCMC chain and let $\boldsymbol{x_1}$ be the next M samples of the $n_{\rm param}$ parameters. Finally, let $\boldsymbol{x}$ be the concatenation of $\boldsymbol{x_0}$ and $\boldsymbol{x_1}$ with total length $N+M$. The mean for $\boldsymbol{x}$ along the $N+M$ axis can be computed as

$$\bar{x}^j = \bar{x}_0^j + \frac{1}{M+N}\sum_{i=1}^{N}(x_1^{ij} - \bar{x}_0^j), \tag{D1}$$

where an over bar denotes an average. Next, we can compute the sample covariance through a batch update as

$$\Sigma = \frac{(N-1)\Sigma_0 + \Sigma_1}{N+M-1}, \tag{D2}$$

where

$$\Sigma_1^{ij} = (x_1^{ji} - \bar{x}^j)(x_1^{ij} - \bar{x}_0^j). \tag{D3}$$

In PTMCMCSampler, these methods use $M = 1$ and iterate over the whole chain.

[1] R. Abbott *et al.*, Observation of gravitational waves from a binary black hole merger, Phys. Rev. Lett. **116**, 061102 (2016).

[2] R. Abbott *et al.*, GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, Phys. Rev. X **13**, 041039 (2023).

[3] 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, C. M. F. Mingarelli, A. Rasskazov, X. Siemens, J. J. Simon, and T. L. Smith, The astrophysics of nanohertz gravitational waves, Astron. Astrophys. Rev. **27**, 5 (2019).

[4] G. Desvignes *et al.*, High-precision timing of 42 millisecond pulsars with the European pulsar timing array, Mon. Not. R. Astron. Soc. **458**, 3341 (2016).

[5] R. N. Manchester *et al.*, The parkes pulsar timing array project, Publ. Astron. Soc. Aust. **30**, e017 (2013).

[6] M. A. McLaughlin, The North American nanohertz observatory for gravitational waves, Classical Quantum Gravity **30**, 224008 (2013).

[7] A. Brazier, S. Chatterjee, T. Cohen, J. M. Cordes, M. E. DeCesar, P. B. Demorest, J. S. Hazboun, M. T. Lam, R. S. Lynch, M. A. McLaughlin, S. M. Ransom, X. Siemens, S. R. Taylor, and S. J. Vigeland, The NANOGrav program for gravitational waves and fundamental physics, arXiv: 1908.05356.

[8] Z. Arzoumanian *et al.* (NANOGrav Collaboration), The NANOGrav 12.5 yr data set: Search for an isotropic stochastic gravitational-wave background, Astrophys. J. Lett. **905**, L34 (2020).

[9] S. Chen *et al.*, Common-red-signal analysis with 24-yr high-precision timing of the European pulsar timing array: Inferences in the stochastic gravitational-wave background search, Mon. Not. R. Astron. Soc. **508**, 4970 (2022).

[10] B. Goncharov *et al.*, On the evidence for a common-spectrum process in the search for the nanohertz gravitational-wave background with the parkes pulsar timing array, Astrophys. J. Lett. **917**, L19 (2021).

[11] B. B. P. Perera *et al.*, The international pulsar timing array: Second data release, Mon. Not. R. Astron. Soc. **490**, 4666 (2019).

[12] B. C. Joshi, P. Arumugasamy, M. Bagchi, D. Bandyopadhyay, A. Basu, N. Dhanda Batra, S. Bethapudi, A. Choudhary, K. De, L. Dey, A. Gopakumar, Y. Gupta, M. A. Krishnakumar, Y. Maan, P. K. Manoharan, A. Naidu, R. Nandi, D. Pathak, M. Surnis, and A. Susobhanan, Precision pulsar timing with the ORT and the GMRT and its applications in pulsar astrophysics, J. Astrophys. Astron. **39**, 51 (2018).

[13] R. Spiewak, M. Bailes, M. T. Miles, A. Parthasarathy, D. J. Reardon, M. Shamohammadi, R. M. Shannon, N. D. R. Bhat, S. Buchner, A. D. Cameron, F. Camilo, M. Geyer, S. Johnston, A. Karastergiou, M. Keith, M. Kramer, M. Serylak, W. Van Straten, G. Theureau, and V. Venkatraman Krishnan, The MeerTime pulsar timing array: A census of emission properties and timing potential, Publ. Astron. Soc. Aust. **39**, e027 (2022).

[14] J. Antoniadis *et al.*, The international pulsar timing array second data release: Search for an isotropic gravitational wave background, Mon. Not. R. Astron. Soc. **510**, 4873 (2022).

[15] A. Zic, G. Hobbs, R. M. Shannon, D. Reardon, B. Goncharov, N. D. R. Bhat, A. Cameron, S. Dai, J. R. Dawson, M. Kerr, R. N. Manchester, R. Mandow, T. Marshman, C. J. Russell, N. Thyagarajan, and X.-J. Zhu, Evaluating the prevalence of spurious correlations in pulsar timing array data sets, Mon. Not. R. Astron. Soc. **516**, 410 (2023).

[16] B. Goncharov, E. Thrane, R. M. Shannon, J. Harms, N. D. R. Bhat, G. Hobbs, M. Kerr, R. N. Manchester, D. J. Reardon, C. J. Russell, X.-J. Zhu, and A. Zic, Consistency of the parkes pulsar timing array signal with a nanohertz gravitational-wave background, Astrophys. J. Lett. **932**, L22 (2022).

[17] R. W. Hellings and G. S. Downs, Upper limits on the isotropic gravitational radiation background from pulsar timing analysis, Astrophys. J. **265**, L39 (1983).

[18] G. Agazie *et al.*, The NANOGrav 15-year data set: Observations and timing of 68 millisecond pulsars, Astrophys. J. Lett. **951**, L9 (2023).

[19] G. Agazie *et al.*, The NANOGrav 15-year data set: Evidence for a gravitational-wave background, Astrophys. J. Lett. **951**, L8 (2023).

[20] G. Agazie *et al.*, The NANOGrav 15-year data set: Detector characterization and noise budget, Astrophys. J. Lett. **951**, L10 (2023).

[21] G. Agazie *et al.*, The NANOGrav 15-year data set: An astrophysical interpretation of a gravitational wave background from supermassive black hole binaries, Astrophys. J. Lett. **952**, L37 (2023).

[22] G. Agazie *et al.*, The NANOGrav 15-year data set: Bayesian limits on gravitational waves from individual supermassive black hole binaries, Astrophys. J. Lett. **951**, L50 (2023).

[23] G. Agazie *et al.*, The NANOGrav 15-year data set: Search for anisotropy in the gravitational-wave background, Astrophys. J. Lett. **956**, L3 (2023).

[24] A. Afzal *et al.*, The NANOGrav 15-year data set: Search for signals of new physics, Astrophys. J. Lett. **951**, L11 (2023).

[25] R. M. Shannon and J. M. Cordes, Assessing the role of spin noise in the precision timing of millisecond pulsars, Astrophys. J. **725**, 1607 (2010).

[26] X. P. You, G. Hobbs, W. A. Coles, R. N. Manchester, R. Edwards, M. Bailes, J. Sarkissian, J. P. W. Verbiest, W. Van Straten, A. Hotan, S. Ord, F. Jenet, N. D. R. Bhat, and A. Teoh, Dispersion measure variations and their effect on precision pulsar timing, Mon. Not. R. Astron. Soc. **378**, 493 (2007).

[27] S. J. Chamberlin, J. D. E. Creighton, X. Siemens, P. Demorest, J. Ellis, L. R. Price, and J. D. Romano, Time-domain implementation of the optimal cross-correlation statistic for stochastic gravitational-wave background searches in pulsar timing data, Phys. Rev. D **91**, 044048 (2015).

[28] R. V. Haasteren, Y. Levin, P. McDonald, and T. Lu, On measuring the gravitational-wave background using pulsar timing arrays, Mon. Not. R. Astron. Soc. **395**, 1005 (2009).

[29] L. Lentati, P. Alexander, M. P. Hobson, S. Taylor, J. Gair, S. T. Balan, and R. van Haasteren, Hyper-efficient

model-independent Bayesian method for the analysis of pulsar timing data, Phys. Rev. D **87**, 104021 (2013).
[30] R. van Haasteren and M. Vallisneri, New advances in the Gaussian-process approach to pulsar-timing data analysis, Phys. Rev. D **90**, 104012 (2014).
[31] R. van Haasteren and M. Vallisneri, Low-rank approximations for large stationary covariance matrices, as used in the Bayesian and generalized-least-squares analysis..., Mon. Not. R. Astron. Soc. **446**, 1170 (2015).
[32] M. Anholm, S. Ballmer, J. D. E. Creighton, L. R. Price, and X. Siemens, Optimal strategies for gravitational wave stochastic background searches in pulsar timing data, Phys. Rev. D **79**, 084030 (2009).
[33] B. Allen and J. D. Romano, Hellings and Downs correlation of an arbitrary set of pulsars, Phys. Rev. D **108**, 043026 (2023).
[34] S. J. Vigeland, K. Islo, S. R. Taylor, and J. A. Ellis, Noise-marginalized optimal statistic: A robust hybrid frequentist-Bayesian statistic for the stochastic gravitational-wave background in pulsar timing arrays, Phys. Rev. D **98**, 044003 (2023).
[35] M. Vallisneri, P. M. Meyers, K. Chatziioannou, and A. J. K. Chua, Posterior predictive checking for gravitational-wave detection with pulsar timing arrays: I. The optimal statistic, Phys. Rev. D **108**, 123007 (2023).
[36] C. Tiburzi, G. Hobbs, M. Kerr, W. Coles, S. Dai, R. Manchester, A. Possenti, R. Shannon, and X. You, A study of spatial correlations in pulsar timing array data, Mon. Not. R. Astron. Soc. **455**, 4339 (2016).
[37] M. Vallisneri *et al.*, Modeling the uncertainties of solar-system ephemerides for robust gravitational-wave searches with pulsar timing arrays, Astrophys. J. **893**, 112 (2020).
[38] S. C. Sardesai and S. J. Vigeland, Generalized optimal statistic for characterizing multiple correlated signals in pulsar timing arrays, Phys. Rev. D **108**, 124081 (2023).
[39] C. R. Harris *et al.*, Array programming with NumPy, Nature (London) **585**, 357 (2020).
[40] P. Virtanen *et al.* (SciPy 1.0 Contributors), SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nat. Methods **17**, 261 (2020).
[41] J. A. Ellis, M. Vallisneri, S. R. Taylor, and P. T. Baker, Enterprise: Enhanced Numerical Toolbox Enabling a Robust PulsaR Inference SuitE (2020), https://dx.doi.org/10.5281/ZENODO.4059815.
[42] S. R. Taylor *et al.*, `nanograv/enterprise_extensions`, v2.4.3 (2024), https://zenodo.org/doi/10.5281/zenodo.10976003.
[43] J. Ellis and R. V. Haasteren, JELLIS18/PTMCMCSampler: Official release (2017), https://dx.doi.org/10.5281/ZENODO.1037579.
[44] J. S. Hazboun, `La_Forge` (2020), https://dx.doi.org/10.5281/ZENODO.4152550.
[45] D. Lee *et al.*, Stan-dev/stan: V2.14.0 (2016), https://dx.doi.org/10.5281/ZENODO.221370.
[46] T. Wiecki *et al.*, PyMC-devs/PyMC: V5.4.0 (2023), https://dx.doi.org/10.5281/ZENODO.7961775.
[47] Z. Arzoumanian *et al.* (NANOGrav Collaboration), The NANOGrav 11 year data set: Pulsar-timing constraints on the stochastic gravitational-wave background, Astrophys. J. **859**, 47 (2018).
[48] J. Simon *et al.*, The NANOGrav 12.5-year data set: Chromatic noise characterization & mitigation (to be published).
[49] Z. Arzoumanian *et al.*, The NANOGrav 11-year data set: High-precision timing of 45 millisecond pulsars, Astrophys. J. Suppl. Ser. **235**, 37 (2023).
[50] E. S. Phinney, A practical theorem on gravitational wave backgrounds, arXiv:astro-ph/0108028.
[51] D. Cournapeau *et al.*, `scikit-sparse` (2024), https://dx.doi.org/10.5281/ZENODO.10976214.
[52] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, Algorithm 887: Cholmod, supernodal sparse Cholesky factorization and update/downdate, ACM Trans. Math. Softw. **35**, 22 (2008).
[53] T. A. Davis and W. W. Hager, Dynamic supernodes in sparse Cholesky update/downdate and triangular solves, ACM Trans. Math. Softw. **35**, 27 (2009).
[54] A. Samajdar *et al.*, Robust parameter estimation from pulsar timing data, Mon. Not. R. Astron. Soc. **517**, 1460 (2022).
[55] Z. Arzoumanian *et al.*, NANOGrav limits on gravitational waves from individual supermassive black hole binaries in circular orbits, Astrophys. J. **794**, 141 (2014).
[56] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087 (1953).
[57] W. K. Hastings, Monte Carlo sampling methods using Markov Chains and their applications, Biometrika **57**, 97 (1970).
[58] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3th ed. (Chapman and Hall/CRC, London, 2013).
[59] R. M. Neal, Sampling from multimodal distributions using tempered transitions, Stat. Comput. **6**, 353 (1996).
[60] W. Vousden, W. M. Farr, and I. Mandel, Dynamic temperature selection for parallel-tempering in Markov Chain Monte Carlo simulations, Mon. Not. R. Astron. Soc. **455**, 1919 (2016).
[61] Message Passing Interface Forum, *MPI: A Message-Passing Interface Standard* (University of Tennessee, 1994).
[62] L. Dalcin and Y.-L. L. Fang, `MPI4Py`: Status update after 12 years of development, Comput. Sci. Eng. **23**, 47 (2021).
[63] L. Dalcín, R. Paz, and M. Storti, `MPI` for Python, J. Parallel Distrib. Comput. **65**, 1108 (2005).
[64] T. Lodewyckx, W. Kim, M. D. Lee, F. Tuerlinckx, P. Kuppens, and E.-J. Wagenmakers, A tutorial on Bayes factor estimation with the product space method, J. Math. Psychol. **55**, 331 (2011).
[65] H. Haario, E. Saksman, and J. Tamminen, An adaptive metropolis algorithm, Bernoulli **7**, 223 (2001).
[66] H. Haario, E. Saksman, and J. Tamminen, Componentwise adaptation for high dimensional MCMC, Comput. Stat. **20**, 265 (2005).
[67] C. J. F. T. Braak, A Markov Chain Monte Carlo version of the genetic algorithm differential evolution: Easy Bayesian computing for real parameter spaces, Stat. Comput. **16**, 239 (2006).
[68] K. Aggarwal *et al.* (NANOGrav Collaboration), The NANOGrav 11 yr data set: Limits on gravitational waves from individual supermassive black hole binaries, Astrophys. J. **880**, 116 (2019).

[69] A. Gelman and D. B. Rubin, Inference from iterative simulation using multiple sequences, Stat. Sci. **7**, 457 (1992).
[70] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner, Rank-normalization, folding, and localization: An improved Ŕ for assessing convergence of MCMC (with discussion), Bayesian Anal. **16**, 667 (2021).
[71] B. P. Carlin and S. Chib, Bayesian model choice via Markov Chain Monte Carlo methods, J. R. Stat. Soc. **57**, 473 (1995).
[72] S. Hourihane, P. Meyers, A. Johnson, K. Chatziioannou, and M. Vallisneri, Accurate characterization of the stochastic gravitational-wave background with pulsar timing arrays by likelihood reweighting, Phys. Rev. D **107**, 084045 (2023).
[73] Y. Ogata, A Monte Carlo method for high dimensional integration, Numer. Math. **55**, 137 (1989).
[74] A. Gelman and X.-L. Meng, Simulating normalizing constants: From importance sampling to bridge sampling to path sampling, Stat. Sci. **13**, 163 (1998).
[75] J. Buchner, Ultranest—A robust, general purpose Bayesian inference engine, J. Open Source Software **6**, 3001 (2021).
[76] J. Skilling, Nested sampling for general Bayesian computation, Bayesian Anal. **1**, 833 (2006).
[77] J. Buchner, Nested sampling methods, Stat. Surv. **17**, 169 (2023).
[78] N. J. Cornish and T. B. Littenberg, BayesWave: Bayesian inference for gravitational wave bursts and instrument glitches, Classical Quantum Gravity **32**, 135012 (2015).
[79] J. S. Hazboun, P. M. Meyers, J. D. Romano, X. Siemens, and A. M. Archibald, Analytic distribution of the optimal cross-correlation statistic for stochastic gravitational-wave-background searches using pulsar timing arrays, Phys. Rev. D **108**, 104050 (2023).
[80] N. J. Cornish and L. Sampson, Towards robust gravitational wave detection with pulsar timing arrays, Phys. Rev. D **93**, 104047 (2016).
[81] S. R. Taylor, L. Lentati, S. Babak, P. Brem, J. R. Gair, A. Sesana, and A. Vecchio, All correlations must die: Assessing the significance of a stochastic gravitational-wave background in pulsar-timing arrays, Phys. Rev. D **95**, 042002 (2017).
[82] M. Vallisneri, LIBSTEMPO: Python wrapper for TEMPO2, Astrophysics Source Code Library, ascl:2002.017 (2020).
[83] G. Ashton and C. Talbot, Bilby-MCMC: An MCMC sampler for gravitational-wave inference, Mon. Not. R. Astron. Soc. **507**, 2037 (2021).
[84] V. Di Marco, A. Zic, M. T. Miles, D. J. Reardon, E. Thrane, and R. M. Shannon, Toward robust detections of nanohertz gravitational waves, Astrophys. J. **956**, 14 (2023).
[85] Z. Arzoumanian *et al.* (NANOGRAV Collaboration), The NANOGrav 12.5-year data set: Search for non-Einsteinian polarization modes in the gravitational-wave background, Astrophys. J. Lett. **923**, L22 (2021).
[86] N. Laal *et al.*, Exploring the capabilities of Gibbs sampling in single-pulsar analyses of pulsar timing arrays, Phys. Rev. D **108**, 063008 (2023).
[87] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python + NumPy programs (2018).
[88] G. E. Freedman, A. D. Johnson, R. Van Haasteren, and S. J. Vigeland, Efficient gravitational wave searches with pulsar timing arrays using Hamiltonian Monte Carlo, Phys. Rev. D **107**, 043013 (2023).
[89] B. Bécsy, N. J. Cornish, and M. C. Digman, Fast Bayesian analysis of individual binaries in pulsar timing array data, Phys. Rev. D **105**, 122003 (2022).
[90] S. Taylor, stevertaylor/{NX01} 1.2 (2017), https://dx.doi.org/10.5281/ZENODO.250258.
[91] J. Ellis and R. van Haasteren, JELLIS18/PAL2: PAL2 (2017), https://dx.doi.org/10.5281/ZENODO.251456.
[92] R. van Haasteren, PICCARD: Pulsar timing data analysis package, Astrophysics Source Code Library, ascl:1610.001 (2016), https://ui.adsabs.harvard.edu/abs/2016ascl.soft10001V.
[93] 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, PINT: A modern software package for pulsar timing, Astrophys. J. **911**, 45 (2021).