Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **507,** 2037–2051 (2021) https://doi.org/10.1093/mnras/stab2236
Advance Access publication 2021 August 5

BILBY-MCMC: an MCMC sampler for gravitational-wave inference

G. Ashton [1]★ and C. Talbot[2]

[1]*Department of Physics, Royal Holloway, University of London, London TW20 0EX, UK*
[2]*LIGO Laboratory, California Institute of Technology, Pasadena, CA 91125, USA*

Accepted 2021 July 23. Received 2021 July 7; in original form 2021 June 16

ABSTRACT

We introduce BILBY-MCMC, a Markov chain Monte Carlo sampling algorithm tuned for the analysis of gravitational waves from merging compact objects. BILBY-MCMC provides a parallel-tempered ensemble Metropolis-Hastings sampler with access to a block-updating proposal library including problem-specific and machine learning proposals. We demonstrate that learning proposals can produce over a 10-fold improvement in efficiency by reducing the autocorrelation time. Using a variety of standard and problem-specific tests, we validate the ability of the BILBY-MCMC sampler to produce independent posterior samples and estimate the Bayesian evidence. Compared to the widely used DYNESTY nested sampling algorithm, BILBY-MCMC is less efficient in producing independent posterior samples and less accurate in its estimation of the evidence. However, we find that posterior samples drawn from the BILBY-MCMC sampler are more robust: never failing to pass our validation tests. Meanwhile, the DYNESTY sampler fails the difficult-to-sample Rosenbrock likelihood test, over constraining the posterior. For CBC problems, this highlights the importance of cross-sampler comparisons to ensure results are robust to sampling error. Finally, BILBY-MCMC can be embarrassingly and asynchronously parallelized making it highly suitable for reducing the analysis wall-time using a High Throughput Computing environment. BILBY-MCMC may be a useful tool for the rapid and robust analysis of gravitational-wave signals during the advanced detector era and we expect it to have utility throughout astrophysics.

Key words: gravitational waves – methods: data analysis – stars: neutron.

1 INTRODUCTION

Gravitational-wave astronomy has enabled the first measurements of masses of merging binary black holes (BBHs; Abbott et al. 2016), new constraints on the equation of state of nuclear matter (Abbott et al. 2018), and offers a new opportunity to measure the expansion rate of the Universe (Abbott et al. 2017) and break the existing measurement tension (Feeney et al. 2019). In the coming years, the Advanced-LIGO (Aasi et al. 2015), Virgo (Acernese et al. 2015), and KAGRA (Aso et al. 2013) detectors are expected to start a fourth observing run that will see the rate of observed signals from binary mergers increase by up to an order of magnitude. This brings with it challenges in data analysis: we need software that is rapid, reliable, and can take advantage of available large-scale computing.

Many of the science goals of gravitational-wave astronomy rely on the ability to robustly draw samples from the posterior distribution $p(\theta|d)$, where θ are the model parameters and d is the data, and estimate the Bayesian evidence $\mathcal{Z}$. [For an introduction to Bayesian inference for gravitational-wave astronomy, see, e.g. Thrane & Talbot (2019).]. For compact binary coalescence (CBC) signals, the posterior distribution is highly non-Gaussian with complicated correlations. Because of the complicated structure of the posterior, stochastic sampling is one of the only viable processes that can robustly generate these quantities for the full complexity of the model (though see Green, Simpson & Gair 2020 and Gabbard et al. 2019 for machine learning based approaches and Pankow et al. 2015 and Lange, O'Shaughnessy & Rizzo 2018 for iterative-fitting based approaches).

Stochastic sampling algorithms to analyse CBC signals have predominately applied either a Markov chain Monte Carlo (MCMC) approach (Metropolis et al. 1953; Hastings 1970), as introduced by Christensen & Meyer (1998) or Nested Sampling (Skilling 2006), as introduced by Veitch & Vecchio (2008). The dominant software used since the first observing run (O1) has been LALINFERENCE (Veitch et al. 2015), which provides three independent stochastic samplers: two Nested Sampling algorithms, LALINFERENCE-NEST and BAMBI (Graff et al. 2012)) and a Metropolis–Hastings MCMC (LALINFERENCE-MCMC) algorithm. Veitch et al. (2015) provided a variety of standard analytical test cases to demonstrate the validity of each of the samplers. But, in the absence of analytical posterior distributions for CBC signals, cross-sampler comparisons, especially between different sampling algorithms, are an important check that results are robust. That LALINFERENCE offered multiple samplers was critical to its success.

The LALINFERENCE software had been widely used, well tested, and become a benchmark for other packages developed since. Since its development, a number of high-quality general-use stochastic sampling software packages are now available. Modern gravitational-wave data-analysis software has been developed that provides an interface to use these off-the-shelf samplers for gravitational-wave astronomy. For example, both BILBY (Ashton et al. 2019) and PYCBC-INFERENCE (Biwer et al. 2019) utilize the DYNESTY (Speagle 2020) Nested Sampling algorithm, and the PTEMCEE (Vousden, Farr

★ E-mail: gregory.ashton@ligo.org

& Mandel 2016) MCMC algorithm (amongst others). In Romero-Shaw et al. (2020), a detailed cross-sampler comparison was performed between the BILBY-implementation of the DYNESTY sampler and LALINFERENCE and the two where found to agreement to within statistical uncertainties. However, such samplers rarely work out of the box and often need some customization and validation to handle CBC signals. For example, a study by Kulkarni & Capano (2020) compared the PTEMCEE and DYNESTY samplers and found the PTEMCEE sampler unable to produce unbiased results for BBH systems.

The BILBY package provides a modular interface to several stochastic samplers and the ability to implement arbitrary likelihoods and priors. This flexibility has made BILBY a popular choice across astrophysics. However, our testing with BILBY has revealed that the implemented MCMC algorithms do not produce results that match those of either the DYNESTY or LALINFERENCE packages for CBC-like use cases. This has a significant future impact: the ability to cross-check between samplers remains a critical test for robustness. In addition, MCMC-based methods are nearly embarrassingly-parallelizable, making them ideal for use in a High Throughput Computing (HTC) environment.[1] With these two motivations, we verify an MCMC-based algorithm implemented within BILBY as of version 1.1.3.

We began by looking at off-the-shelf options. These are preferable as they are well tested and often actively maintained and improved by the open-source community. An obvious choice that has demonstrated performance for CBC inference (Biwer et al. 2019) is the PTEMCEE MCMC sampler, an adaptive parallel-tempered version of the EMCEE (Foreman-Mackey et al. 2013) algorithm. [The multimodal posterior distributions inherent to CBC problems necessitate the use of the PTEMCEE parallel-tempering approach (Gilks, Roberts & Sahu 1998; Earl & Deem 2005)]. Both of these algorithms use ensemble-sampling in which $n_{\rm ens}$ MCMC chains evolve in tandem, new points are proposed based on the position of other chains in the ensemble (cf. Section 2.7). However, in testing we found PTEMCEE to be inefficient compared to the DYNESTY sampler (by up to a factor of 100). In comparison, the LALINFERENCE-MCMC sampler has a demonstrated efficiency similar to that of the Nest algorithm (Veitch et al. 2015). Unlike the off-the-shelf samplers, LALINFERENCE-MCMC utilizes a *proposal distribution* (cf. Section 2) that takes advantage of knowledge about the problem in hand (in our case, CBC signals). This suggests the need for a parallel-tempered MCMC sampler with access to problem-specific proposals.

In this work, we develop BILBY-MCMC, a from-scratch MCMC algorithm with adaptive parallel tempering and ensemble sampling. We develop BILBY-MCMC to take advantage of a wide variety of proposal distributions including standard, problem-specific and machine learning based proposals. We validate the BILBY-MCMC sampler against the DYNESTY nested sampler (as described in Romero-Shaw et al. (2020)) for its use in both standard validation problems and CBC inference. In this paper, we discuss the tuning and validation of BILBY-MCMC for efficient inference of CBC signals. However, as part of the BILBY package, BILBY-MCMC can be used as a sampler for any inference problem, and includes access to an interface to define custom problem-specific proposal distributions.

We introduce the BILBY-MCMC sampler in Section 2, then discuss the validation for a set of standardized tests in Section 3. In Section 4, we apply a set of CBC-specific validation tests and describe the performance before concluding in Section 5.

[1]HTC environments differ from High Performance Computing in that the interconnect speeds between nodes is slow. This makes HTC environments sub-optimal for algorithms that require regular inter-node communication (e.g, the massively parallel methods explored in Smith et al. (2020)). As discussed later in Section 2.9, multiple independent MCMC algorithms can be run that continuously produce independent samples, making them ideal for an HTC environment.

2 BILBY-MCMC

MCMC algorithms generate correlated samples from the target distribution, in our case the posterior distribution, by a sequential stepping process. We now describe the details of the algorithm relevant to the BILBY-MCMC implementation, for a more thorough introduction to MCMC algorithms in astrophysics we refer the reader to recent reviews (Sharma 2017; Hogg & Foreman-Mackey 2018).

We apply the Metropolis–Hastings algorithm (Metropolis et al. 1953; Hastings 1970) to draw samples from the target density,

$$p(\theta|d) \propto \mathcal{L}(d|\theta)\pi(\theta), \tag{1}$$

where $\mathcal{L}(d|\theta)$ is the *likelihood* of the model parameters θ and $\pi(\theta)$ is the *prior* probability of the model parameters. Throughout, we assume a fixed model M, though formally we note that both the likelihood and priors are model-dependent [i.e. $\mathcal{L}(d|\theta)$ is more completely written as $\mathcal{L}(d|\theta, M)$ and similarly for the prior].

Given a current sample θ, a proposed sample θ' is generated from a *proposal distribution* $Q(\theta'|\theta)$. (We discuss proposal distributions in Section 2.1.) We accept the proposed sample θ' with a probability

$$\alpha = \min\left(1, \frac{Q(\theta|\theta')\,\mathcal{L}(d|\theta')\,\pi(\theta')}{Q(\theta'|\theta)\,\mathcal{L}(d|\theta)\,\pi(\theta)}\right), \tag{2}$$

and append θ' to a *chain* of samples. If the proposal is rejected, the current sample, θ, is appended to the chain. We implement the Metropolis–Hastings step in practice by drawing a random number u from a uniform distribution on the unit interval; if $\alpha \geq u$, the proposal is accepted, otherwise it is rejected.

We initialize the chain with a random draw from the prior distribution $\pi(\theta)$ and iterate the Metropolis–Hasting algorithm to generate a chain of m samples $\{\theta_i\}$ where $i \in [0, m)$. Samples in the chain are generally correlated. Independent samples can be obtained from $\{\theta_i\}$ by selecting a subset of m/τ samples where τ is the autocorrelation time (ACT) of the chain. We select the subset by taking a sample every τ steps. We iterate the algorithm until reaching the stopping criteria,

$$n_{\rm samples} \geq \frac{m - n_{\rm burn}}{\gamma\tau}, \tag{3}$$

where $n_{\rm burn}$ is the number of samples discarded to remove the chain initialization, known as the *burn-in* period, and $\gamma \leq 1$ is a *thinning* factor (in the PYTHON interface, γ is `thin_by_nact`).

To estimate τ, we use the autocorrelation module provided by the EMCEE `v3.0` package (Foreman-Mackey et al. 2013). This method improves on the traditional approach by adding an automated function to choose the window (cf. Hogg & Foreman-Mackey 2018 along with the software documentation; https://emcee.readthedocs.io/en/stable/tutorials/autocorr/).

We provide a number of automated approaches to estimate the burn-in period $n_{\rm burn}$. The primary method is a simple scaling: we discard $r_{\rm burn}$ ACTs, i.e. $n_{\rm burn} = r_{\rm burn}\tau$. By default, we use $r_{\rm burn} = 10$, but this scaling factor can be varied by the user through the `burn_in_nact` option. In addition, proposal methods that violate the assumptions of the MCMC algorithm (e.g. using dynamic tuning

to improve convergence) set minimum values for $n_{\rm burn}$ and the user can also specify $n_{\rm burn}$ directly if these automated approaches fail.

To produce independent samples (matching the behaviour of the LALInference-MCMC sampler), we can set the thinning factor $\gamma = 1$. However, thinning is inefficient in the sense that unthinned samples ($\gamma < 1$) are unbiased and provide greater precision for summary statistics (Link & Eaton 2012). In cases where $\gamma < 1$, we differentiate between the number of samples equation (3), and the effective number of samples $n^{\rm eff}_{\rm samples} = \gamma n_{\rm samples}$. For the standard validation tests in this work, we use $\gamma = 1$. For the CBC validation tests, we use $\gamma = 1/5$ and $n_{\rm samples} \geq 25\,000$. This ensures a minimum number of 5000 independent samples while smoothing posterior plots and providing more accurate summary statistics.

Having introduced the simple Metropolis-Hastings algorithm, we now turn to the specifics of the Bilby-MCMC implementation that enable it to efficiently perform CBC parameter estimation. In Section 2.1, we define the standard library of proposal distributions then introduce the learning proposals in Section 2.2 and the gravitational-wave specific proposals in Section 2.3. We note that Bilby-MCMC provides a flexible interface to define and use new proposals. As such, this list is not an exhaustive set of all available proposals. In Section 2.4, we describe how the proposals are used together in a block-updating sampling approach. In Section 2.5, we describe the extension to a parallel-tempered sampler required to analyse multimodal distributions then describe how ensemble-sampling is implemented in Section 2.7. In Section 2.8, we provide a model for the efficiency of the sampler, then introduce a timing model, and discuss computational parallelization in Section 2.9.

2.1 Standard proposal distributions

Broadly speaking, the performance of an MCMC sampler is determined by the ACT of the chains it produces. Chains with smaller ACTs taker fewer steps to traverse the target distribution and hence produce more independent samples for a fixed number of MCMC steps (or equivalently, computational cost). The ACT itself depends on how efficient the proposal distribution $Q(\theta'|\theta)$ is in proposing points that enable the chains to traverse the posterior.

For the Metropolis–Hastings algorithm, there are two ways to optimize a stochastic sampler to reduce the ACT. First, we can choose a parametrization that reduces the complexity of the parameter space. If under a transformation T, the posterior distribution has a simpler form (e.g, if T maps a Banana-like distribution to a multivariate Gaussian, or softens hard edges), then sampling in $T(\theta)$ rather than θ is the most straight forward approach to improving the algorithm performance (Hogg & Foreman-Mackey 2018). In Section 4.1, we discuss the best known parametrization for CBC signals. Secondly, once the best known parametrization is chosen, we optimize the choice of proposal distributions.

In this section, we introduce the standard library of proposals implemented in Bilby-MCMC and discuss their performance and utility. For each proposal, we also provide the *Hastings factor*,

$$\mathcal{H} = \frac{Q(\theta|\theta')}{Q(\theta'|\theta)}, \tag{4}$$

which ensures detailed balance is met (Hastings 1970) and enables unbiased sampling using asymmetric proposals.

2.1.1 FG: fixed Gaussian

The Fixed Gaussian proposal implemented in Bilby-MCMC is a generalization of the zero-mean multivariate Gaussian proposal (Gelman et al. 1996) in which a proposal for the ith parameter is generated from

$$\theta_i' = \theta_i + \sigma_i w_i \epsilon, \tag{5}$$

where σ_i is a user-defined scale parameter, w_i is the prior support (if the prior has infinite support, we set $w_i = 1$) for θ_i, and ϵ is a draw from a standard normal distribution. The introduction of the scaling by the prior support enables some automatic tuning to the anticipated scale of the problem, while the σ_i enables the user to define varying length-scales for each parameter. Of note, our implementation does not allow the user to change the spatial orientation of the proposal (i.e. through correlations between parameters). For the Fixed Gaussian proposal, which is symmetric, $\mathcal{H} = 1$.

In practice, the Fixed Gaussian proposals have limited use and require manual tuning (through the choice of σ_i) to achieve meaningful performance on realistic problems. As such, we do not enable this proposal by default.

2.1.2 AG: adaptive Gaussian

To circumvent the tuning requirements of a Fixed Gaussian proposal, Haario, Saksman & Inen (2001) introduced the notion of an adaptive proposal that uses past performance of the sampler to drive the sampler to a target acceptance rate.

Such adaptive proposal are non-Markovian and may lead to the generated samples not being representative of the posterior. However, as discussed in Haario et al. (2001) and Veitch et al. (2015) (in the context of CBC signals), if the adaptation rate decays throughout the run or the adaptation is halted sufficiently early in the run, the equilibrium distribution may be sufficiently close to the posterior. We verify that, within the statistical uncertainties relevant for typical CBC problems, this is true for our Adaptive Gaussian proposal.

To dynamically adapt the proposal, we use the acceptance ratio,

$$a = \frac{n_{\rm accepted}}{n_{\rm accepted} + n_{\rm rejected}}, \tag{6}$$

to quantify the current performance (Gelman et al. 1996). If $a \sim 1$, proposals are accepted too often: this suggests slow exploration of the posterior. If $a \ll 1$, proposals are infrequently accepted: the proposed points tend to jump away from areas of high posterior support. In both cases, this leads to large ACTs. For well-tuned proposals (which reduce the ACT relative to poorly tuned proposals) and under idealized settings, Roberts, Gelman & Gilks (1997) demonstrated that $a \sim 0.23$. We set this as a target acceptance rate and dynamically adapt the proposal to achieve it.

Our implementation of the Adaptive Gaussian proposal extends equation (5),

$$\theta_i' = \theta_i + s\sigma_i w_i \epsilon, \tag{7}$$

adding a factor of s, the adapting scale parameter. Initially, $s = 1$, then on each iteration where the proposal is applied, we update s following Veitch et al. (2015):

$$s \to s + s_\gamma \frac{(1 - a')}{100}, \tag{8}$$

if the previous proposed point was accepted or

$$s \to s - s_\gamma \frac{a'}{100}, \tag{9}$$

if the previous proposed point was not accepted. Here $a^{'} = 0.234$ is the target acceptance rate and the quantity,

$$s_\gamma = \left(\frac{N}{n}\right)^{1/5} - 1, \tag{10}$$

is the adaptation decay rate with n the number of points proposed and N a user-specified number of steps after which to stop adapting, the default value is $N = 10^5$. We set a minimum scale $s \geq N^{-1}$. For the Adaptive Gaussian proposal, which is symmetric, $\mathcal{H} = 1$.

2.1.3 DE: differential evolution

We implement the Differential Evolution proposal (Ter Braak 2006; ter Braak & Vrugt 2008) as described in Veitch et al. (2015). Two samples θ^a and θ^b are drawn at random from the chain. Then

$$\theta' = \theta + \gamma(\theta^a - \theta^b), \tag{11}$$

where γ is chosen randomly from $\gamma = \left\{1, N(0, 2.38/\sqrt{2n_{\rm dim}}\right\}$. When $\gamma = 1$, this acts as a mode-hopping proposal improving the performance of the sampler in multimodal problems. When γ is drawn from the normal distribution [as proposed by Ter Braak 2006; Roberts & Rosenthal 2001), the proposed points lie along the line passing through θ and $\theta^{'}$. As such, the proposal is well suited to posterior distribution with linear correlations in which the line passing through θ and $\theta^{'}$ lies along the principle axes. As with the Adaptive Gaussian proposal, formally this proposal makes the chain non-Markovian. Later, in Section 3, we verify that the equilibrium distribution is statistically identical to the posterior, i.e the posterior is unbiased. Like the Gaussian proposals, the Differential Evolution proposal is symmetric, such that $\mathcal{H} = 1$.

2.1.4 PR: prior proposal

The prior proposal draws $\theta^{'}$ from the prior $\pi(\theta)$. For well-measured parameters, in which the posterior is much narrower than the prior, this proposal is highly inefficient. However, we find it to be effective when used as part of a block-updating set of proposals applied to poorly measured parameters (e.g. the spin and tidal parameters of the secondary lower mass object in a CBC inference problem). It also aids mode-mixing in high-temperature chains (see Section 2.5). For the Prior proposal, the Hastings factor is $\mathcal{H} = \pi(\theta)/\pi(\theta')$.

2.1.5 UN: uniform proposal

A simplification of the Prior Proposal, this proposal proposes points uniformly within the prior bounds. We utilize this proposal as a robust and simpler variant of the Prior Proposal with similar performance. For the Uniform proposal, which is symmetric, the Hastings factor is $\mathcal{H} = 1$.

2.2 Machine learning proposal distributions

In BILBY-MCMC, we introduce a class of *learning* proposals that, as we show in Section 3, dramatically decrease the ACT while producing statistically identical posterior distributions. learning proposals use a random sampling from the past MCMC chain to learn the distribution and then generate new samples. For all learning proposals, during an initialization stage (during which the MCMC chain has not yet been explored), they fall back to an Adaptive Gaussian proposal. Once the initialization stage is complete, they sample the MCMC chain, use the samples to fit the proposal distribution, and then this distribution is used to propose new points. Periodically, the proposal distribution is refitted using new samples from the MCMC chain to circumvent premature learning. As with the Adaptive Gaussian proposal, the use of the past chain again breaks the Markovian property of the chain, but we verify in Section 3 that the resulting posterior remains unbiased using validation tests.

2.2.1 KD: Gaussian kernel density estimate

We fit a Gaussian Kernel Density Estimate (KDE; Rosenblatt 1956; Parzen 1962) to a random draw of samples from the MCMC chain. We find this non-parametric multivariate density estimate to be both rapid in learning (typical learning times are fractions of a second) and flexible enough to fit complicated features. KDE methods have previously been used in the context of CBC inference by the KOMBINE (Farr & Farr, in preparation) ensemble sampler.

When used to estimate a probability density from a set of samples, KDE methods suffer a subtle dependence on a tuneable 'bandwidth' parameter and typically over-smooth hard edges and multimodal distributions. However, when used as a learning proposal density, these issues only result in loss of efficiency, and do not bias results. To understand why, consider a parameter with a hard edge (e.g. the lower bound on the spin of a black hole that cannot be negative). A KDE proposal fitted to a chain will over-smooth the hard edge and propose non-physical points with negative spin. However, the MCMC algorithm will never accept these points. This results in a small loss of efficiency, but no bias.

We utilize the standard implementation of Gaussian KDE in the SCIPY (Virtanen et al. 2020) package with bandwidth estimated using 'Scotts rule' (Scott 2015). Once the KDE $k(\theta)$ is fitted, proposal samples can be drawn directly and the Hastings factor calculated by $\mathcal{H} = k(\theta)/k(\theta')$. We find that fitting the KDE takes fractions of a second while the proposal time is negligible compared to typical CBC likelihood evaluation times.

2.2.2 GM: Gaussian mixture model

While KDEs smooth a set of samples as a Gaussian centred on each sample, in a Gaussian mixture model (GMM) the density is estimated using a finite number of Gaussian distributions. As with KDE methods, this model is not good at fitting distributions with hard edges. The means and covariance matrices of these Gaussian distributions are chosen using an expectation-maximization algorithm. We use the SKLEARN (Pedregosa et al. 2012) package to fit the GMM. In this work, we use 10 components in the mixture. Fitting the GMM takes slightly more time than fitting the KDE; however, it is typically <1 s and sampling from/evaluating the GMM is faster than sampling from the KDE as there are fewer components. As with the KDE proposal, the Hastings factor is calculated from $\mathcal{H} = g(\theta)/g(\theta')$, where $g(\theta)$ is the fitted GMM.

2.2.3 NF: Normalizing flows

The *normalizing flows* class of machine learning algorithms (Papamakarios et al. 2019) learn a bijective map from the target density (the set of training samples drawn from the MCMC sampler) to a latent space, in our case a multivariate Gaussian. Normalizing flows have previously been used in gravitational-wave astronomy to directly sample the CBC posterior distribution (Green et al. 2020; Green & Gair 2021) and as way to propose new points in a nested sampler (Williams, Veitch & Messenger 2021). Following the work of Hoffman et al. (2019), Moss (2020), we use the NFLOWS

package (Durkan et al. 2020), which implements the normalizing flows algorithm in PyTorch (Paszke et al. 2019), to learn the proposal distribution. We periodically optimize the normalizing flow using the Jensen–Shannon divergence (JSD) test (cf. Appendix A) between samples drawn from the learnt map and a set of independent validation samples. Unlike the KDE and GMM proposals, the normalizing flows proposals can take several tens of seconds to minutes to train. In practice (cf. Section 3), we find the normalizing flows method has a similar performance to the GMM method, but at an increased computational cost. Therefore, we do not utilize it for CBC inference problems. The Hastings factor is again given by the ratio of the normalizing flow density at the initial and proposed points.

2.3 Gravitational-wave proposal distributions

We implement the gravitational-wave-specific *polarization and phase correlation*, *phase reversal*, and *phase and polarization reversal* proposals as described in Veitch et al. (2015). We find these proposals dramatically improve the sampling for analyses that do not utilize analytic marginalization of the binary phase (cf. Section 4.2). We do not implement the *sky reflection*, *extrinsic-parameter*, and *Gibbs sampling of distance* proposals described in Veitch et al. (2015), Raymond & Farr (2014). While we expect these to be general improvements to the algorithm, the use of distance marginalization (cf. Section 4.2), and our choice of parametrization (cf. Section 4.1) diminish the expected utility of these proposals.

2.4 Block sampling

Each of the proposal distributions described in the last three sections can update either all parameters in the set of model parameters θ, or only a subset of those parameters. The Bilby-MCMC sampler is initialized with a list of individual proposals, the subset of θ that they are to update, and their unnormalized weighting. We then use the weighting to create a cyclic *proposal cycle*. At each step of the sampler, the next proposal in the cycle is chosen, a point is proposed and accepted/rejected based on the condition described in equation (2). The proposal cycle enables weighted block-updating of proposals and ensures the detailed balance condition is met as described in Veitch et al. (2015).

For non-CBC inference problems (i.e. the standard tests considered in Section 3), we default to an equal-weighted set of the Adaptive Gaussian, Differential Evolution, Uniform, KDE, GMM, and Normalizing Flow proposals. Though, this can be customized by users. For CBC inference problems, we define a proposal cycle described in Table 1. We arrived at this choice by hand tuning: running analyses on simulated signals and identifying opportunities for improvement. As such, we do not anticipate that the proposals selected in Table 1 are optimal and we expect improvements to be made in the future. Users can modify and extend proposal cycle using the flexible interface.

2.5 Parallel-tempering

The standard Metropolis Hastings algorithm does not produce estimates of the evidence, and fails when attempting to sample from multimodal distributions. Parallel-tempering (Gilks et al. 1998; Earl & Deem 2005) addresses both of these issues.

As the name suggests, $n_{\rm temps}$ parallel MCMC chains are run. (In practice, these can be updated sequentially, i.e. stepping each chain in turn, or using the parallelization techniques described later in Section 2.9. However, the chains must remain pseudo-synchronized to enable swaps between chains). For the jth chain, the likelihood in equation (2), is modified:

$$\mathcal{L}(d|\theta, M) \rightarrow \mathcal{L}(d|\theta, M)^{1/T_j}, \tag{12}$$

where $T_j \geq 1$ is the chain 'temperature'. Note that the ladder of temperatures $\{T_j\}$ is ordered $T_{j+1} > T_j$. The $T_0 = 1$ 'cold' chain samples from the target posterior distribution. But, for 'hot' chains with $T_j > 1$, the likelihood is flattened out and easier to sample.

Periodically, swaps are proposed between adjacent chains and accepted with a probability,

$$\min\left[1, \left(\frac{\mathcal{L}(d|\theta_m)}{\mathcal{L}(d|\theta_n)}\right)^{(1/T_n)-(1/T_m)}\right]. \tag{13}$$

These swaps provide a mechanism for the cold temperature chain (which generates posterior samples) to explore multimodal likelihoods. We utilize the dynamic temperature adaption methods described in Vousden et al. (2016) to optimize the choice of the temperature ladder $\{T_j\}$. Samples taken during this optimization period are automatically labelled as part of the burn-in epoch.

Table 1. The gravitational-wave proposals set used in this work.

Proposal	θ-subset	Weight
Adaptive Gaussian	All	10
Differential evolution	All	10
Adaptive GaussianI	Intrinsic	10
Differential evolution	Intrinsic	10
KDE	Intrinsic	10
GMM	Intrinsic	10
Differential evolution	Extrinsic	10
KDE	Extrinsic	10
GMM	Extrinsic	10
Adaptive Gaussian	Extrinsic	5
Differential evolution	Mass	5
GMM	Mass	5
Differential evolution	Spin	5
GMM	Spin	5
Adaptive Gaussian	Measured-spin	5
Differential evolution	Mass ratio and primary spin	5
Differential evolution	Tidal deformability	5
Prior proposal	Tidal deformability	5
Phase reversal	Phase	0.1
Phase and polarization reversal	Phase and polarization	0.1
Correlated phase/polarization	Phase and polarization	0.1
Prior	$\psi, \phi_{12}, \theta_2, \Lambda_1, \Lambda_2, t_j$	0.1

Notes. For a description of the proposals themselves, see Section 2.1. In cases where the θ-subset is 'all', the whole set of θ is updated. Where a subset is listed, see Section 4.1, only that subset is updated by the proposal. The weights are unnormalized and determine the relative frequency of each proposal. In the final row of 'Prior' proposals, each is updated individually, not as a set.

2.6 Evidence calculation

In addition to resolving the problem of sampling multimodal distributions, parallel-tempering also enables an estimate to be made of $\ln \mathcal{Z}$, the natural logarithm of the Bayesian evidence. To estimate the evidence in Bilby-MCMC, we implement *thermodynamic integration* (Goggans & Chi 2004; Lartillot & Philippe 2006) as described in Littenberg & Cornish (2009) and Veitch et al. (2015),

and the *stepping-stone algorithm* (Xie et al. 2010; Maturana-Russel et al. 2019). In testing, we verify the findings of Maturana-Russel et al. (2019): The stepping stone method is superior, producing more accurate results for the same computational cost. As such, while BILBY-MCMC calculates both methods, we report only the stepping stone evidence throughout this work.

2.7 Ensemble sampling

In recent years, ensemble-sampling algorithms have been highly successful in astrophysics (see, e.g. Foreman-Mackey et al. 2013; Vousden et al. 2016; Farr & Farr, in preparation). These algorithms use an ensemble of interacting MCMC samplers. New points are proposed based on the current distribution of the ensemble of points enabling automatic tuning of the proposals to the target density. That these algorithms self-tune has been paramount to their versatility and use throughout astrophysics.

In BILBY-MCMC, an ensemble of $n_{\rm ens}$ chains can be utilized with inter-chain swaps proposed by an ensemble stretch proposal (Goodman & Weare 2010). In comparison to the EMCEE and PTEMCEE samplers, BILBY-MCMC is poorly vectorized and does not scale to many hundreds of chains.

If used in conjunction with parallel-tempering, one can either use $n_{\rm temps}$ ensembles (for a total of $n_{\rm temps} \times n_{\rm ens}$ samplers) or with one parallel-tempered chain (for a total of $n_{\rm temps} + n_{\rm ens} - 1$ samplers). The former configuration mirrors how the PTEMCEE sampler operates while the latter configuration may be useful, for example, to provide an estimate of the evidence with a reduced computational cost. For thermodynamic integration, each set of parallel-tempered chains is used to calculate an estimate of the evidence, then the results are averaged between chains. In the validation tests described in Sections 3 and 4, we do not utilize the ensemble sampler as it was found to provide no practical improvement in efficiency.

2.8 Efficiency

Throughout this work, we will quantify and compare the posterior sampling *efficiency* of samplers by the ratio of the number of independent samples to the number of likelihood evaluations:

$$\epsilon = \frac{n^{\rm eff}_{\rm samples}}{n_\ell}. \tag{14}$$

For a simple MCMC sampler, the number of steps is equal to the number of likelihood evaluations. However, in BILBY-MCMC we do not evaluate the likelihood if the proposed sample is outside the prior bounds. Nevertheless, the number of likelihood evaluations is the relevant weighting as it is the dominant computational operation.

We calculate the efficiency directly for the validation tests in Section 3, but here we first derive an efficiency model. For a parallel-tempered ensemble sampler with $n_{\rm temps} \times n_{\rm ens}$ chains where $n_{\rm burn} = r_{\rm burn}\tau$ are discarded for burn-in (in practice, there are several alternative methods that can determine $n_{\rm burn}$ as described in Section 2), the efficiency is

$$\epsilon = \frac{1}{\tau n_{\rm temps}(1-\xi)}, \tag{15}$$

where

$$\xi = \frac{r_{\rm burn} n_{\rm ens}}{n^{\rm eff}_{\rm samples}} \tag{16}$$

is the *burn-in inefficiency*, the fraction of 'wasted' samples due to the burn in process.

When configuring the sampler, care should be taken to ensure $\xi \ll 1$ to avoid significant wasted computation. For example, drawing 1000 independent samples using $n_{\rm ens} = 1$ and the default $r_{\rm burn} = 10$, the burn-in inefficiency is a reasonable 1 per cent. However, if we attempt to use 10 co-evolving ensembles $n_{\rm ens} = 10$, the burn-in inefficiency also increases to 10 per cent. (The same logic equally applies to configurations that combine independent runs as discussed in Section 2.9, replacing $n_{\rm ens}$ with the number of independent runs and $n^{\rm eff}_{\rm samples}$ with the number of samples per run.).

Provided $\xi \ll 1$, the efficiency is determined by the ACT τ and the number of parallel-tempered chains $n_{\rm temps}$. The ACT is a property of the sampling algorithm, which can be reduced using the methods discussed in Section 2.1. Naively, reducing $n_{\rm temps}$ appears to improve the posterior sampling efficiency. However, $n_{\rm temps} > 1$ is required to sample from multimodal distributions, calculate the evidence, and can reduce τ. In Section 3 we will demonstrate with a specific example, but as a rough rule of thumb about $n_{\rm temps} = 8$ is sufficient for the multimodal posteriors of CBC inference problems and provides a reasonable estimation of the evidence. However, if a refined estimate of the evidence is required, more temperatures are needed, decreasing the posterior sampling efficiency.

We develop a resampling approach to reclaim some of this lost efficiency. For the jth chain with temperature T_j, we define $\{\theta_i\}_{(j)}$ as the set of posterior samples it produces from the tempered posterior distribution:

$$p(\theta|d) \propto \mathcal{L}(d|\theta)^{1/T_j}\pi(\theta). \tag{17}$$

For each sample θ from the tempered posterior distribution, we calculate a weight,

$$w = \mathcal{L}(d|\theta)^{1-(1/T_j)}, \tag{18}$$

from the ratio of the hot likelihood to the $T = 0$ likelihood. Then, we rejection sample[2] (MacKay 2003) the tempered posterior samples resulting in a set of posterior samples from the cold posterior distribution. For low dimensional problems, we find this produces a modest gain in efficiency at no additional cost. As an example, analysing a CBC signal using a non-spinning model and using the analytic marginalization of the distance, phase and time (cf. Section 4.2), rejection sampling the hot chains produces an $\sim$20 per cent efficiency improvement. However, for fully-precessing CBC problems, we find the rejection sampling does not accept any new points (i.e. the efficiency remains unchanged). In Sections 3 and 4, we do not utilize the rejection sampling method.

In this work, we will compare the efficiency of BILBY-MCMC with that of the DYNESTY sampler using the random walk proposal method described in Romero-Shaw et al. (2020). This proposal method has a tuning parameter $n_{\rm act}$ that determines the number of internal MCMC steps to take based on the estimated ACT. Following Speagle (2020) (in which the DYNESTY sampler was shown to be more efficient than the EMCEE sampler), we calculate the efficiency from equation (14), with $n^{\rm eff}_{\rm samples}$ calculated from the *effective sample size* as estimated from the nested sampling weights. We note that this assumes that new points proposed during nested sampling are independent, however this is not required (Salomone et al. 2018) or guaranteed in practice. If the points are correlated, $n^{\rm eff}_{\rm samples}$ will significantly overestimate the efficiency of the DYNESTY sampler. To guard against this (and to investigate the potential impact on posterior estimation), for the Rosenbrock and Unimodal Gaussian validation tests, we run the

[2]We draw u from a uniform distribution on the unit interval, if $u < w$, the sample is accepted, otherwise it is rejected.

DYNESTY sampler with two different values of $n_{\rm act}$ and verify that the efficiency approximately scales with $n_{\rm act}$. This demonstrates that the $n_{\rm act}$ value chosen is sufficiently large to generate independent samples and hence that the efficiency of the DYNESTY is not overestimated.

2.9 Timing model and parallelization

We distinguish two levels of computational parallelization that can reduce the wall time: *combining independent runs* and *multiprocessing* an individual run.

Combining independent runs is embarrassingly parallel: we simply repeat N copies of the analysis using an identical data and configuration, but a different random seed. If each analysis produces $n^{\rm eff}_{\rm samples}$ independent samples, then, in total, we end up with $Nn^{\rm eff}_{\rm samples}$. Such a configuration is ideal for use in an HTC environment and has the added advantage that one can cross-compare between chains. However, this approach is limited by the increase in burn-in inefficiency (cf. equation 16): Each independent run has to burn-in. This may be worthwhile to decrease the wall-time for important and time-sensitive results.

Before discussing the use of multiprocessing, we introduce a timing model to understand the wall-time required to produce $n^{\rm eff}_{\rm samples}$ independent samples from the posterior of a single serial run. For CBC inference problems, the compute-time is determined by t_ℓ, the time required to evaluate the likelihood[3] and the number of likelihood evaluations required. In a serial-processing model, the total time T can be estimated by

$$T = n_\ell t_\ell = \frac{n_{\rm samples} t_\ell}{\epsilon} \approx 28\,{\rm h}\left(\frac{n^{\rm eff}_{\rm samples}}{1000}\right)\left(\frac{t_\ell}{10\,{\rm ms}}\right)\left(\frac{\epsilon}{0.01\ {\rm per\ cent}}\right)^{-1}, \quad (19)$$

where we have taken a typical efficiency from Section 4.3 for an CBC analysis using $n_{\rm temps} = 8$.

If either $n_{\rm temps} > 1$ or $n_{\rm ens} > 1$, BILBY-MCMC can be trivially parallelized leveraging the multicore processors typically available in modern processors. We implement this parallelization using the PYTHON standard-library `multiprocessing` module. In this model of multiprocessing, there is an overhead cost to transferring the data (i.e. any data products required by the likelihood). For typical CBC problems, this can be as much as a few milliseconds. [We note that the LALINFERENCE MCMC sampler (Veitch et al. 2015) mitigates this by the use of a distributed computing model with a Message Passing Interface]. This overhead time is comparable to the likelihood evaluation time t_ℓ and results in imperfect scaling of the timing model equation (19). We model this by introducing $m \leq 1$, a parallelization *inefficiency* that we will measure empirically. Then, the timing model for an analysis parallelized over $n_{\rm cores}$ is

$$T = \frac{n^{\rm eff}_{\rm samples} t_\ell}{\epsilon}\frac{1}{m n_{\rm cores}}. \quad (20)$$

The number of cores should be matched to the number of parallelizable jobs, i.e. $n_{\rm cores} = n_{\rm temps} n_{\rm ens}/m$, where m is a non-zero natural number. If the number of cores is mismatched with the number of parallelizable jobs, i.e. $n_{\rm cores} > n_{\rm temps} n_{\rm ens}$, this will always leave one or more cores idle. When $n_{\rm cores} = n_{\rm temps} n_{\rm ens}$, we refer to this as *perfect matching*.

[3]Typically, t_ℓ ranges from a few to many hundreds of milliseconds and is dominated by the cost to evaluate the waveform. Longer duration signals typically take longer to evaluate. However, when calculating the likelihood during an MCMC analysis, cached waveform evaluations can be used, e.g. when proposing a move only in the extrinsic parameters. The discussion in this section assumes a fixed t_ℓ, resulting in a conservative timing estimate that ignores this potential computational saving.

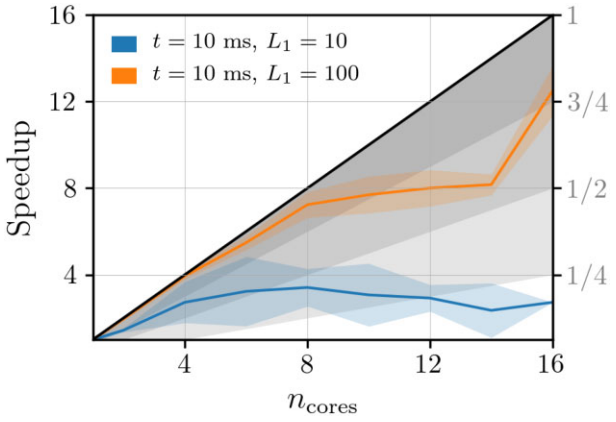

Figure 1. Empirically measured speed-up for a test analysis with $n_{\rm temps} = 16$ and $n_{\rm ens} = 1$. Solid lines indicate the mean while shaded region indicates the standard deviation as measured over three identical runs. The right-hand axis provides the speed-up factor m at perfect matching.

We can measure m empirically by looking at the speed-up for an identical analysis as a function of $n_{\rm cores}$. We find that direct parallelization results in values of m that are as small as (or in the worst case smaller than) $1/n_{\rm cores}$, i.e. the parallelization can be slower than a serial run. This is because of the substantial data-transfer overhead. To mitigate the data transfer overhead, in parallel analyses, we transfer data and then take a fixed number, L_1, of 'internal' MCMC steps. To further improve the efficiency, we do not store these internal steps. In effect, this pre-thins the MCMC chains by a factor of L_1. When $L_1 > 1$, the ACT and other associated quantities are calculated on the stored chain, but can be re-scaled.

In Fig. 1, we determine the speed-up factor m for a test case in which $n_{\rm temps} = 16$ and the per-likelihood evaluation time t_ℓ is held fixed at 10 ms. We run the experiment twice. First, we use $L_1 = 10$, which demonstrates poor parallelization scaling with an overall speed-up factor of $m \sim 1/8$ for perfect matching. Then, we increase L_1 to 100 and see improved scaling with $m \sim 3/4$ for perfect matching. For $n_{\rm cores} = 8$ or less, the performance is near-optimal $m \sim 1$. The marginal improvement in speed between $n_{\rm cores}$ of 10, 12, and 14 demonstrates the effect of imperfect matching.

Using the empirically measured m from Fig. 1, if our analysis is using $n_{\rm temps} = 8$, $n_{\rm ens} = 1$, we can see the rough timing predicted by equation (20) for perfectly matched parallelized runs:

$$T \approx 5\,{\rm h}\left(\frac{n^{\rm eff}_{\rm samples}}{1000}\right)\left(\frac{t_\ell}{10\,{\rm ms}}\right)\left(\frac{\epsilon}{0.01\ {\rm per\ cent}}\right)^{-1}\left(\frac{m}{0.75}\right)^{-1} \times \left(\frac{n_{\rm cores}}{8}\right)^{-1}. \quad (21)$$

It is worth pointing out two caveats to this timing model. First, while increasing L_1 improves m, if L_1 is greater than the typical ACT, this itself introduces a new type of inefficiency (namely an over-thinned chain). Second, these quantities are not independent. For example, if one wants to combine a large number of independent runs, each producing $n^{\rm eff}_{\rm samples} = 10$ samples, it may appear that equation (21) would predict an ∼3-min analysis time. However, the burn-in inefficiency would be increased leading to a decrease in the efficiency and hence increase in the overall run time. This

Table 2. Validation tests reported in this work.

Test	Sampler	Configuration	$n_{\rm temps}$	Evidence	JSD [mb]	ACT (τ)	Efficiency (per cent)	$n_\ell/10^6$	$n^{\rm eff}_{\rm samples}$
Standard Normal	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 50$	–	0.02 ± 0.04	0.9	–	0.58	1.1	6200
	BILBY-MCMC	AG-DE-UN	1	–	0.5	6	15.0	0.05	8000
	BILBY-MCMC	AG-DE-UN	16	0.03 ± 0.01	0.5	5	1.2	0.5	6000
	BILBY-MCMC	AG-DE-UN	32	-0.007 ± 0.01	0.5	5	0.6	0.8	5000
Rosenbrock	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 10$	–	0.08 ± 0.07	2.7	–	0.7	1.1	7500
	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 50$	–	0.04 ± 0.07	1.6	–	0.1	6.7	7500
	BILBY-MCMC	AG-DE-UN-GM-NF-KD	16	-0.02 ± 0.01	0.7	10	0.6	3.3	20 000
	BILBY-MCMC	AG-DE-UN-GM-NF-KD	1	–	0.3	16	6.2	0.34	20 000
	BILBY-MCMC	AG-DE-UN-NF	1	–	0.3	19	5.2	0.40	21 000
	BILBY-MCMC	AG-DE-UN-KD	1	–	0.3	110	0.9	2.2	20 000
	BILBY-MCMC	AG-DE-UN-GM	1	–	0.3	17	6.1	0.37	22 000
	BILBY-MCMC	AG-DE-UN	1	–	0.5	171	0.6	3.4	20 000
Unimodal Gaussian	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 10$		-0.05 ± 0.2	0.8	–	0.09	22	20000
	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 50$	–	0.07 ± 0.2	0.7	–	0.01	150	20 000
	BILBY-MCMC	AG-DE-UN-GM-NF-KD	1	–	0.05	85	1.2	0.45	5000
	BILBY-MCMC	AG-DE-UN-GM-NF-KD	16	-0.25 ± 0.13	0.006	70	0.09	5.7	5000
	BILBY-MCMC	AG-DE-UN-GM-NF-KD	32	-0.03 ± 0.06	0.003	61	0.05	10	5000
Bimodal Gaussian	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 50$	–	0.02 ± 0.2	0.4	–	0.005	390	20 000
	BILBY-MCMC	AG-DE-UN-GM-KD	16	-0.05 ± 0.1	0.3	3	0.02	32	5000
	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 50$	–	51.4 ± 0.2	–	–	0.010	160	16 000
BBH A	BILBY-MCMC	–	8	49.7 ± 0.2	–	7×10^3	0.0018	300	5000
	DYNESTY	$n_{\rm live} = 2000$, $n_{\rm act} = 50$	–	133.4 ± 0.2	–	–	0.006 86	220	15 000
BNS A	BILBY-MCMC	–	8	132.6 ± 0.3	–	60×10^3	0.000 21	2500	5000

White-shaded rows are those from the standard validation tests (Section 3) while grey-shaded rows are tests from gravitational-wave validation tests (Section 4). For the standard validation tests, we give the BILBY-MCMC configuration by the set of proposals (described in Section 2.1), while for the DYNESTY sampler, we give $n_{\rm live}$ and $n_{\rm act}$ (cf. Romero-Shaw et al. 2020). For the gravitational-wave validation tests, we use the proposal set described in Table 1. In the Evidence column, we report $\Delta \ln \mathcal{Z} = \ln \mathcal{Z} - \ln \mathcal{Z}'$ for the standard validation tests where the exact evidence $\ln \mathcal{Z}'$ is known; for the gravitational-wave validation tests (grey rows), where the evidence is not known, we report the natural logarithm of the signal versus Gaussian noise Bayes factor. Where the posterior distribution can be directly sampled from, we report the maximum JSD (see Section A) in milli-bits [mb]. For the MCMC configurations, we list the final-estimated ACT τ; this is always given in raw steps (i.e. we re-scale runs that use $L_1 > 1$). For the gravitational-wave validation tests, τ is given by the mean valued averaged over all independent runs; typically, this varies by several tens of per cent. We also report the posterior sampling efficiency described in Section 2.8, the total number of likelihood evaluations, and the number of independent samples the analysis produced.

may still be preferable, but we urge users to give consideration to the efficiency before starting analyses. We comment that the inefficiencies commented on above are specific to the naive approach of combining multiple independent runs. The fundamental issue arises that a standard MCMC chain produces unbiased samples from the target density as the number of steps tends to infinity. There are sophisticated approaches that will overcome these inefficiencies by instead aiming to obtain unbiased samples as the number of MCMC chains tends to infinity (Jacob, O'Leary & Atchadé 2020). These approaches could be used in future to improve the efficiency of BILBY-MCMC when parallelized over many cores.

In this section, we have seen that BILBY-MCMC can be parallelized both by combining independent runs and utilizing multiprocessing. By comparison, the run-time of the DYNESTY nested sampler can only be reduced by the use of multiprocessing. This is because it is not possible to configure a nested sampler to run part of the full analysis (i.e. only to produce a small subset of the required total number of independent samples). In this work, we utilize multiprocessing of the DYNESTY sampler in Section 4 that can take advantage of multicore processors. We note that DYNESTY can also be used in High Performance Computing (HPC) environments by multiprocessing using many multicore processors (Smith et al. 2020). We discuss the relative merits of these two approaches with reference to a specific example in Section 4.3.

3 STANDARD VALIDATION TESTS

In this section, we outline a suite of tests designed to validate the BILBY-MCMC package for standardized problems. These tests build on previous validation tests of gravitational-wave samplers (Veitch et al. 2015; Biwer et al. 2019) and tests of the DYNESTY sampler (Speagle 2020) implemented in BILBY (Ashton et al. 2019; Romero-Shaw et al. 2020). Though not reported here, we additionally perform integration checks on individual aspects of the sampler and verify that when the likelihood is uninformative the prior is properly recovered. The scripts used to perform all verification checks and additional figures are available from git.ligo.org/gregory.ashton/bilby_mcmc_validation; in Table 2, we also link to the individual tests.

3.1 Standard normal distribution

As an initial test, we evaluate a 1D standard-normal likelihood, where

$$\mathcal{L}(\theta) = \frac{\mathrm{e}^{-\theta^2/2}}{\sqrt{2\pi}}, \tag{22}$$

and the prior is uniform between -10 and 10:

$$\pi(\theta) = U(-10, 10). \tag{23}$$

In this case, the evidence can be estimated as

$$\mathcal{Z} = \int_{-\infty}^{\infty} \mathcal{L}(\theta)\pi(\theta)\,\mathrm{d}\theta \approx \frac{1}{20}. \tag{24}$$

and the posterior $p(\theta)$ is a standard-normal distribution.

Running the Bilby-MCMC and Dynesty samplers on this problem, in Table 2, we report the configurations, the difference in log-evidence, and quantities related to the performance.

To verify the posterior sampling, we calculate the JSD (see Appendix A for an extended discussion) between 5000 independent posterior samples drawn using the sampler and samples drawn directly from the known posterior. For all configurations, we report JSD values below a threshold of 2 mb (where mb is the shorthand for a milli-bit of information): this demonstrates the posteriors are *statistically identical*. As such, we conclude that both the Dynesty and Bilby-MCMC samplers are able to sample this simple inference problem without bias and report accurate estimates of the uncertainty.

To verify the estimates of the Bayesian evidence, we compare with the known evidence calculated in equation (24). Both the Dynesty and Bilby-MCMC sampler using $n_{\mathrm{temps}} = 32$ produce estimates of the evidence that agree with equation (24) to within the stated uncertainties. However, it is known that parallel-tempered evidence estimates have a bias that can be reduced by increasing the number of temperatures (Xie et al. 2010; Maturana-Russel et al. 2019). This point is demonstrated by the Bilby-MCMC analyses with $n_{\mathrm{temps}} = 16$, which does not produce a result consistent with the known evidence.

3.2 Rosenbrock likelihood

We analyse the Rosenbrock likelihood (Rosenbrock 1960), taking the explicit form and priors from equation (C2) of Fowlie, Handley & Su (2020). The banana-shaped posterior is challenging to sample from and representative of the types of posteriors seen in CBC inference problems. This makes it an ideal validation test. Results for several configurations of both samples are listed in Table 2.

We sample directly from the posterior distribution of the Rosenbrock likelihood using a re-parametrization. This enables us to calculate the maximum JSD between samples drawn using different configurations of the Bilby-MCMC and Dynesty samplers and the directly sampled posterior. The maximum JSD for the Bilby-MCMC analyses all fall below the 2-mb threshold for statistically identical posteriors. However, we find that the samples from the Dynesty sampler using $n_{\mathrm{act}} = 10$ are marginally above this threshold while the analysis with $n_{\mathrm{act}} = 50$ is below. In re-running these analyses, we find variations in the JSD value of the order of$\sim$50 per cent: this indicates the Dynesty analyses are subtly biased. n_{act} is a user-controlled parameter described in Romero-Shaw et al. (2020), which determines the number of internal MCMC steps to take based on the estimated ACT. A value of 10 was previously found to be sufficient for BBH analyses (Romero-Shaw et al. 2020), but this test demonstrates larger values may be necessary to ensure convergence for the Rosenbrock likelihood. The dependence on n_{act} indicates the cause is likely to be the MCMC-within-nested-sampling algorithm itself (we used the version in Bilby v1.1.3 for the analyses in this work); investigation is needed to determine if this is failing and to resolve this bias.

We visualize the results in Fig. 2: Bilby-MCMC and the 'direct' samples agree, but samples from the Dynesty analyses (with $n_{\mathrm{act}} = 50$) are overly constrained. This is a typical failure mode of posterior samples generated by nested sampling methods that use bounding ellipsoids to improve performance of the sampler. We note that we do not see similar issues for CBC inference problems (see Sections 4.3 and 4.4). This failure requires further investigation and highlights the need for cross-sampler comparisons.

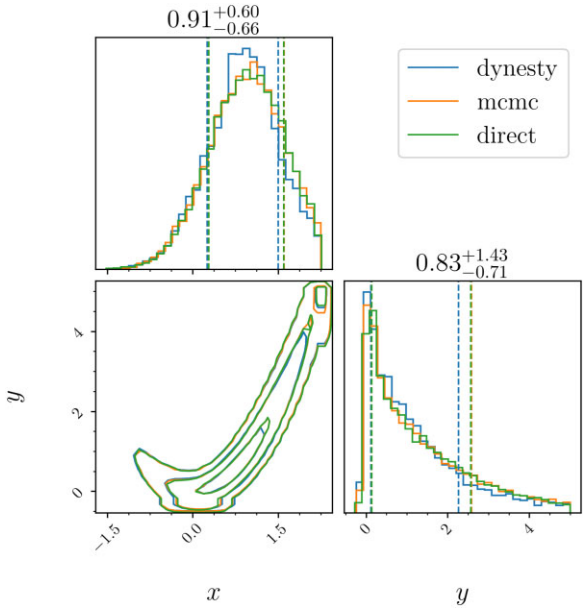

Figure 2. Comparison of the Dynesty sampler (with $n_{\mathrm{live}} = 2000$ and $n_{\mathrm{act}} = 50$), the Bilby-MCMC sampler (with $n_{\mathrm{temps}} = 1$ and the full set of learning proposals), and samples drawn directly from the posterior for the Rosenbrock test. The JSD test between each of the samplers and the direct samples (see Table 2) quantifies that the Bilby-MCMC sampler produces statistically identical posterior samples while the Dynesty sampler produces JSD values at the failure threshold. Visually, we see that the posterior samples produced by the Dynesty sampler are overly constrained.

The various MCMC configurations in Table 2 enable a comparison of the impact of the learning proposals. Using all three learning proposals (AG-DE-UN-GM-NF-KD), reduces the ACT by a factor of $\gtrsim 10$ with respect to the analysis without any learning proposals (AG-DE-UN). By running each of the learning proposals individually, we see that the normalizing flows and GMM proposals both have a similar performance improvement (with respect to the AG-DE-UN proposals alone) to all three together. Meanwhile the KDE proposal alone provides only a factor of $\sim$2 reduction in the ACT. This demonstrates that learning-proposals are a powerful tool in improving the efficiency of the MCMC algorithm.

Finally, we turn to evidence estimation. The Rosenbrock likelihood used in this work has an analytically approximated evidence of $\ln \mathcal{Z}' = -5.804$ (Fowlie et al. 2020). In Table 2, we provide evidence estimates for the Dynesty and Bilby-MCMC sampler with $n_{\mathrm{temps}} = 16$. For the Dynesty sampler, the evidences agree to within the stated uncertainties. For the Bilby-MCMC sampler, the evidence estimate disagrees at the level of 1 standard deviation. This performance is consistent with the findings of Veitch et al. (2015) in which the LALInference MCMC sampler similarly struggled to consistently estimate the evidence of the Rosenbrock likelihood.

3.3 15D unimodal Gaussian

We analyse the 15D unimodal multivariate Gaussian distribution originally proposed in Veitch et al. (2015) using the specific configuration from Romero-Shaw et al. (2020). We report the results in Table 2, varying the number of parallel-tempered chains, but utilizing the standard proposal sets. For all samplers and configurations,

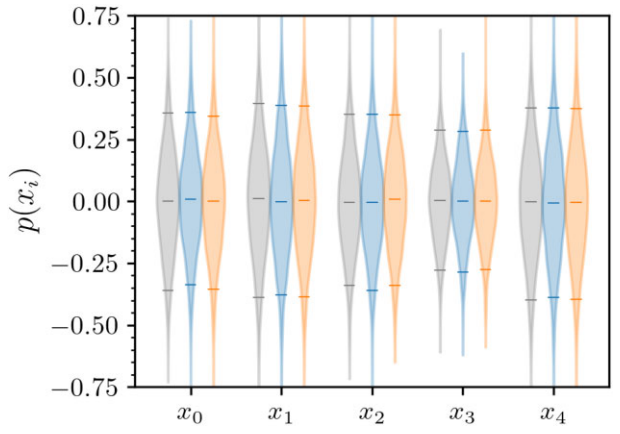

Figure 3. Violin plot showing posteriors from five parameters of the 15D unimodal Gaussian test. Each violin shows 5000 samples drawn directly from the posterior (grey), from the DYNESTY analysis with $n_{\rm act} = 50$ (blue) and the BILBY-MCMC analysis with $n_{\rm temps} = 1$ (orange). Vertical lines denote the median and 90 per cent credible interval.

the maximum JSD falls below the nominal 2-mb threshold for statistically identical samples. To visualize the results, in Fig. 3, we plot a subset of five posteriors in a violin plot. This shows strong agreement between the samplers and with samples drawn directly from the posterior.

The evidence for this 15D unimodal Gaussian test case can be estimated directly (Romero-Shaw et al. 2020) as $\ln \mathcal{Z}' \approx -34.54$. The DYNESTY sampler correctly estimates the evidence to within the stated uncertainty in both configurations. Meanwhile, the BILBY-MCMC sampler gets close to the true evidence, with $n_{\rm temps} = 32$, but suffers the previously discussed bias when $n_{\rm temps}$ is small.

We study the performance of BILBY-MCMC while varying $n_{\rm temps}$. In this unimodal case, even a single-temperature sampler can sample the posterior. Increasing the number of parallel temperatures from 1 to 16 marginally reduces the ACT. As $n_{\rm temps}$ is increased, the uncertainty on the evidence estimate decreases, but the ACT does not significantly change. This is expected since it is a unimodal target density that does not require parallel-tempering to hop between modes. As such, increasing the number of temperatures (to achieve a meaningful estimate of the evidence) results in a reduction in the efficiency as predicted by equation (16).

3.4 15D bimodal Gaussian

We analyse a bimodal Gaussian distribution consisting of two copies of the unimodal Gaussian distribution (cf. Section 3.3) and means separated by 8 standard deviations in each dimension (as used in Romero-Shaw et al. (2020)). This test probes the ability of the sampler to efficiently hop between modes. With a single cold chain, the MCMC sampler is unable to find both modes (in other words, the ACT is infinite). With $n_{\rm temps} = 16$, BILBY-MCMC is able to sample from both modes. Comparing to samples drawn directly from the posterior, the maximum JSD for the DYNESTY sampler and BILBY-MCMC sampler both fall below the threshold for statistically identical samples. Romero-Shaw et al. (2020) noted that the DYNESTY sampler tends to overweight one or other of the two modes in this test. But, that combining over many runs the effect averages out. We confirm this in our individual run of the DYNESTY sampler. For the BILBY-MCMC sampler, we find that the effect is weaker. Quantifying the effect by the number of samples in each mode, the BILBY-MCMC tends to produce more equal-weighted posteriors (in agreement with the true posterior). This can be understood because the MCMC sampler is proposing jumps between modes while for the DYNESTY sampler the relative weights of the two modes is determined by the bounding ellipsoids.

The evidence for the 15D bimodal Gaussian be directly estimated (Romero-Shaw et al. 2020) as $\ln \mathcal{Z}' \approx -34.54$. Comparing the evidence estimated by the samplers to this direct estimation, we find similar performance to that of the 15D unimodal Gaussian studied in Section 3.3. Namely, the DYNESTY sampler outperforms BILBY-MCMC in accuracy and uncertainty.

4 GRAVITATIONAL-WAVE VALIDATION TESTS

In this section, we discuss the specifics and validation of the BILBY-MCMC sampler for CBC gravitational-wave inference. The inference of CBC coalescence signals has been well studied in the literature. The fundamentals can be found in Veitch et al. (2015), a recent review in Thrane & Talbot (2019), and the specifics of the BILBY interface in Ashton et al. (2019) and Romero-Shaw et al. (2020). In Section 4.1, we introduce the basics of the CBC model and describe the best-known parametrization of θ to reduce the ACT. Then, in Section 4.2, we discuss the use of analytic marginalization methods that reduce the dimensionality of θ in sampling.

4.1 Models, optimal parametrization, and priors

A circularized gravitational-wave signal from a CBC can be described by a set of 17 model parameters θ. We can partition θ into 11 intrinsic parameters (two mass, six spin parameters, the binary phase, and up to two tidal deformability parameters) and 6 extrinsic parameters (the 3D localization, polarization, merger time, and the angle between the total angular momentum and the line of sight).

There are many ways to choose these 17 parameters in the literature. These different parametrizations offer varying levels of computational convenience and interpretability. In this work, we use the following parametrization for CBC analyses based on which parameters lead to the shortest auto-correlation lengths in our tests.

4.1.1 Mass

Labelling the detector-frame mass of the two objects in the binary m_1 and m_2, we sample in the detector-frame chirp mass,

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}, \tag{25}$$

and mass ratio $q = m_2/m_1$. We apply prior cuts, discussed below, such that $q \leq 1$. This is the standard choice employed for compact binary analyses as the chirp mass is the best measured parameter for binary inspirals, followed by the mass ratio (Cutler & Flanagan 1994).

4.1.2 Spin

The spin of the compact objects contribute 6 degrees of freedom to the problem. Following Farr et al. (2014), we sample in the *magnitudes and tilts* parametrized in spherical coordinates with the z-axis aligned with the total angular momentum using the magnitude a_i and tilt θ_i (where $i \in [1, 2]$ labels the primary and secondary objects) along with two azimuthal parameters $\{\phi_{jl}, \phi_{12}\}$.

4.1.3 Tides

Tidal deformability of neutron stars is typically described in terms of either two dimensionless deformability parameters Λ_i or combinations of these two combinations of these parameters that directly determine the contribution to the phase contribution $\{\tilde{\Lambda}, \delta\tilde{\Lambda}\}$ (Flanagan & Hinderer 2008; Favata 2014; Wade et al. 2014). We sample in the latter set as these reduce the ACT.

4.1.4 Location

The location of the binary is uniquely described by four parameters, the distance to the source, the 2D sky location, and the merger time. We choose these parameters as in LALInference. We specify the merger time as the time of arrival of the merger signal at one of the detectors, ideally the one where we expect the highest signal-to-noise ratio (S/N). We characterize the sky location using a reference frame based on the separation vector of two of the detectors; see Romero-Shaw et al. (2020) section 3.1.7 for an explicit definition. For all of the results presented here, we marginalize over the distance to the source (see Section 4.2), so the specific choice of distance parameter is irrelevant.

4.1.5 Orientation

Finally, we require three Euler angles to convert the binary frame to the galactic reference frame. These are the inclination angle between the binary angular momentum and the line-of-sight from the source to the observer $\theta_{\rm JN}$, the binary phase at a reference frequency ϕ (typically the merger frequency), and the polarization of the source ψ. Throughout, we sample in $\cos(\theta_{\rm JN})$ and ψ. In practice, there is a strong correlation between the phase and the polarization and so we sample in a phase offset parameter:

$$\delta\phi = \begin{cases} \phi + \psi & \theta_{\rm JN} \leq \frac{\pi}{2} \\ \phi - \psi & \theta_{\rm JN} > \frac{\pi}{2} \end{cases}. \tag{26}$$

The change of sign is due to a change in the direction of the degeneracy when observing from above/below the orbital plane. This parametrization introduces a discontinuity in the likelihood at $\theta_{\rm JN} = \pi/2$. However, the proposal schemes outlined in Section 2.1, including the machine-learned proposals, do not depend on assumptions of smoothness. In practice we find that the parametrization improves the performance relative to analyses that use the phase directly.

Following LALInference, we apply a prior uniform on the component masses m_1 and m_2 with cuts in the chirp mass and mass ratio. We then apply the non-informative priors on all other parameters and a uniform in the source-frame prior for the luminosity distance (Romero-Shaw et al. 2020).

4.2 Analytic likelihood marginalization

Of the 17D parameter space described in Section 4.1, there are three, namely the luminosity distance, geocentric time, and binary phase, over which we are able to efficiently marginalize the gravitational-wave likelihood [see Veitch & Del Pozzo (2013), Farr (2014), Veitch et al. (2015), Singer & Price (2016), Singer et al. (2016), and Thrane & Talbot (2019) for a review]. In the context of an MCMC sampler, the marginalized likelihood has a shorter ACT relative to the non-marginalized likelihood. This is both due to the reduction in dimensionality and to the reduction in the complexity of the posterior. Since it is possible to reconstruct the marginalized parameters after analysis (Thrane & Talbot 2019), where possible marginalized likelihoods are strongly recommended. For the luminosity distance, we always marginalize the likelihood. For the geocentric time, we marginalize the likelihood [and add the time jitter, t_j, as described in Romero-Shaw et al. (2020)] except in instances where the reduced-order-quadrature method ROQ is used in which time-marginalization has not yet been implemented. The assumptions made in marginalizing the binary phase are invalid for precessing CBC systems or models that include higher order emission modes. Therefore, we do not marginalize the binary phase in this work. But, in future use cases, where a non-pressing waveform without higher order emission modes is considered, we do recommend using phase marginalization.

Table 3. Simulation parameters for the three fiducial events analysed in Section 4.

	Parameters		BBH A	BNS A
Intrinsic	Mass	$\mathcal{M}$	17.1	1.4875
		q	0.62	0.950
	Spin	a_1	0.296	0.01
		a_2	0.393	0.01
		θ_1	0.09	0
		θ_2	1.20	0
		ϕ_{12}	1.10	0
		ϕ_{jl}	0.52	0
	Tidal	Λ_1	0	1500
		Λ_2	0	750
Extrinsic	Loc.	RA	3.95	1.67
		Dec.	0.22	−1.22
		$d_{\rm L}$	497	180
	Orient.	$\theta_{\rm JN}$	1.88	−0.88
		ψ	2.70	2.70
		ϕ	3.69	3.69

Note. In the two left columns, we provide the parameter groups names as described in Section 4.1.

4.3 Fiducial BBH: BBH A

We simulate a fiducial (reference) BBH signal observed by the LIGO Hanford and Livingston detectors (Aasi et al. 2015) at their design sensitivity (Abbott et al. 2020). The simulation parameters, labelled as BBH A, are given in Table 3. We use the `IMRPhenomPv2` (Schmidt, Hannam & Husa 2012; Hannam et al. 2014) waveform approximant to both simulate and analyse the signal. In this noise realization, the simulated signal has a network matched-filter S/N of ∼13.

We analyse 4 s of simulated data with the Dynesty and Bilby-MCMC samplers using the configurations described in Table 2, the priors described in Section 4, and distance and time marginalization. For the Bilby-MCMC sampler, we use 13 independent chains, a thinning factor of $\gamma = 0.2$, and run each chain until it produces 2000 samples. In total, this produces 25 000 samples with $n^{\rm eff}_{\rm samples} = 5000$. For the Dynesty sampler, we use the standardized configuration listed in Table 2, but use two independent run to enable a robustness check.

It is not possible to sample directly from the posterior in this case, so we resort to cross-sampler comparisons to verify posterior sampling. Across all CBC parameters, we find that the maximum JSD between the samplers falls below the 2-mb threshold, i.e. we find statistically identical posteriors between Dynesty and Bilby-MCMC. To visualize these difference in Fig. 4, we plot histograms of

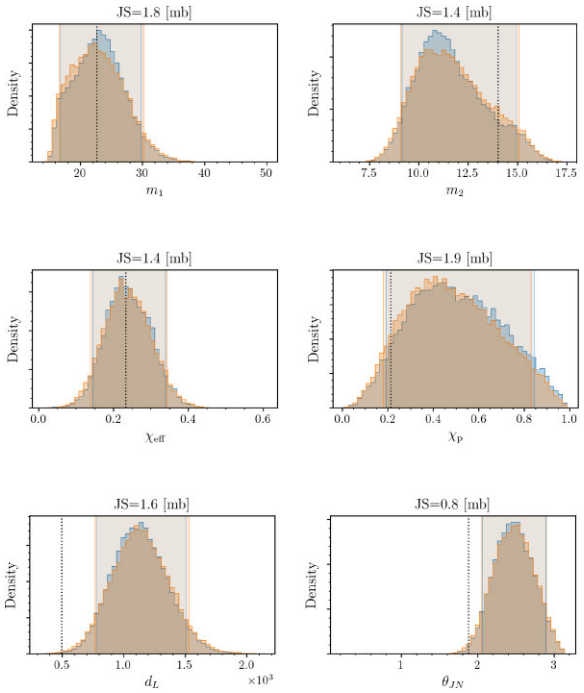

Figure 4. Histograms of the posteriors from the DYNESTY (blue) and BILBY-MCMC (orange) analyses of the fiducial BBH. Configurations and summary statistics are given in Table 2. Vertical lines mark the edges of the 90 per cent credible interval for each sampler and black dotted lines mark the value used to simulate the data. Note that we do not expect the posteriors to peak at the simulation values due to the influence of the simulated noise and the Bayesian prior. In the title of each figure, we give the JSD; across all parameters, this JSD is found to be below the 2-mb threshold we use to determine if the two sets of posteriors are statistically different.

several quantities of astrophysical interest, along with their individual JSD.

In the evidence column of Table 2, we report the Bayes factor between the signal evidence and the Gaussian noise evidence (for a fixed realization of the noise and power spectral density this is a fixed quantity). The evidence estimates disagree at the 1σ level of the quotes uncertainties. The difference is likely explained by the known bias in parallel-tempered evidence (cf. Section 2.6) when $n_{\rm temps}$ is small. Here, we tune $n_{\rm temps}$ for efficient sampling of the posterior, rather than evidence estimation. In such a configuration, we recommend that the evidence estimate only be used as a rough guide, but not be used for quantitative analysis. To reduce the bias, $n_{\rm temps}$ can be increased at the cost of posterior sampling efficiency.

Comparing the performance of the two samplers, the DYNESTY sampler is an order of magnitude more efficient than the BILBY-MCMC sampler. However, this efficiency does not directly translate into an order of magnitude reduction in wall-time. To understand why, we need to discuss the parallelization strategies available.

As discussed in Section 2.9, we have two available levels of parallelization: combining independent runs and multiprocessing using $n_{\rm cores}$ processors. For the DYNESTY sampler, reductions in wall-time can only be achieved via multiprocessing. This is because it is not possible to configure a nested sampler to run part of the full analysis (i.e. to only produce a small subset of the required total number of independent samples). A simple model for the wall-time of the DYNESTY run that agrees with our measured wall-time is

$$T = 28\,\mathrm{h}\left(\frac{n_\ell}{160\times10^6}\right)\left(\frac{t_\ell}{10\,\mathrm{ms}}\right)\left(\frac{n_{\rm cores}}{16}\right)^{-1}, \tag{27}$$

where n_ℓ is the number of likelihood evaluations (cf. Table 2) and t_ℓ is the approximate per-likelihood evaluation time for the BBH A likelihood. Here, we use 16-core processors: below we will discuss the potential scaling to larger multiprocessing pools.

On the other hand, for BILBY-MCMC we can parallelize using independent runs and multiprocessing. We run several independent runs, each producing 400 independent samples. In an HTC environment (and assuming access to resources is not limited), these can be run at the same time so that the total analysis wall time is given by the wall-time of any individual run. Using equation (20) and perfectly matching $n_{\rm cores}$ to $n_{\rm temps}$,

$$T \approx 10\,\mathrm{h}\left(\frac{n^{\rm eff}_{\rm samples}}{400}\right)\left(\frac{t_\ell}{10\,\mathrm{ms}}\right)\left(\frac{\epsilon}{0.0017\text{ per cent}}\right)^{-1} \times \left(\frac{m}{0.75}\right)^{-1}\left(\frac{n_{\rm cores}}{8}\right)^{-1}, \tag{28}$$

where we use the actual efficiency from Table 2 and multiprocessing speed-up factor from Section 2.9. Both equations (27) and (28) agree with the empirically measured values (up to errors expected for varying access to resources in an HTC environment).

The net result is that the BILBY-MCMC sampler is less efficient, but can be set up to enable a shorter wall time by utilizing independent runs. Some of this inefficiency arises from the sampler itself, some from the burn-in inefficiency. For this configuration, the burn-in inefficiency (equation 16) is a few per cent; further parallelization (in terms of more independent runs) would increase this inefficiency.

For the DYNESTY sampler, reducing the wall-time can only be achieved via access to a larger multiprocessing pool. The ability to do this is restricted by the available hardware: $n_{\rm cores}$ of 8–16 are typical in most HTC environments though modern CPUs with up to 128 cores do exist, which could provide significant speed ups. Beyond this, massively parallelized nested sampling can leverage multiple CPUs in an HPC environment: in Smith et al. (2020), processing pools including several hundred cores have been used providing two orders of magnitude of speed up. (We caution that we have not verified the validity of equation 27) for such massively-parallel environments.) However, access to such resources requires synchronized usage of a dedicated HPC environment.

To investigate the potential for bias in the BILBY-MCMC sampler, in Fig. 5, we show the results of a parameter-parameter (PP) test (Cook, Gelman & Rubin 2006; Talts et al. 2018) for BBH systems. This is an important test, typically it fails when one or more of the proposal distributions does not respect detailed balance. In this test, we simulate 100 BBH signals drawn from an astrophysical prior distribution, analyse each using the BILBY-MCMC sampler, and then check the consistency of the reported credible intervals. Specifically, Fig. 5 shows the number of events in a given confidence interval as a function of the confidence interval. We find that the BILBY-MCMC sampler is unbiased at the level probed by this test.

4.4 Fiducial binary neutron star: BNS A

We simulate a fiducial binary neutron star (BNS) merger using the `IMRPhenomPv2_NRTidal` waveform (Dietrich, Bernuzzi & Tichy 2017; Dietrich et al. 2019) that includes matter effects from the two neutron stars. The simulation parameters of the system, BNS

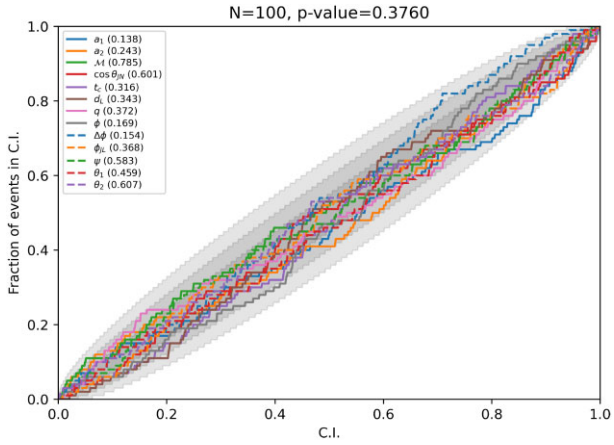

Figure 5. A parameter-parameter test for the Bilby-MCMC sampler for simulated BBH systems. We plot the fraction of simulated events found within the confidence interval (CI) as a function of the C.I. For an unbiased sampling from the posterior distribution, lines of this plot are diagonal: we add three grey shaded regions showing the 1σ, 2σ, and 3σ quantiles. To quantify if the results are consistent with an unbiased sampling, we calculate a p-value of the probability that they are unbiased. The p-value for each individual parameter is given in the legend and a combined p-value is given in the title. Under an unbiased result, we would expect the p-value to be a draw from uniform distribution on [0, 1]. Since all individual parameters (and the combined result) are greater than 1/15 (a nominal threshold based on the number of parameters), we conclude the sampler is unbiased, at least at the level probed by 100 simulations.

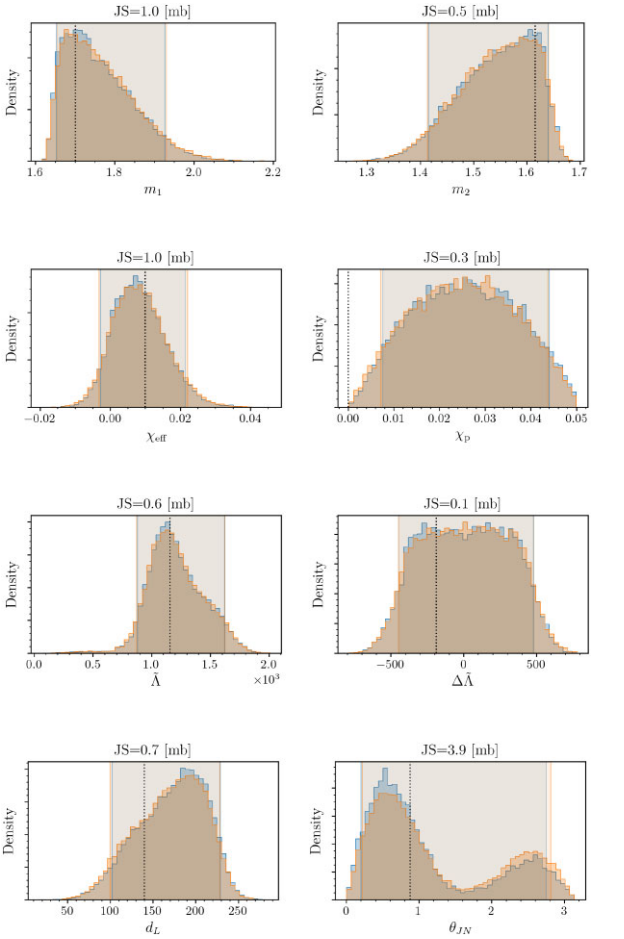

Figure 6. Histograms of selected posteriors from the Dynesty (blue) and Bilby-MCMC (orange) analyses of the fiducial BNS A. Configurations and summary statistics are given in Table 2. Vertical lines mark the edges of the 90 per cent credible interval for each sampler and a black dotted lines marks the value used to simulate the data. Note that we do not expect the posteriors to peak at the simulation values due to the influence of the simulated noise and the Bayesian prior. The largest JS-divergence reported across all parameters occurs for the inclination, $\theta_{\rm JN}$, above the threshold of 2 mb (see Appendix A).

A, listed in Table 3 are much lower in mass than that of the BBH systems previously studied. The result of this lower mass is that the signal spends a longer duration in the observable band of the detectors (typically, above 20 Hz). To capture this, we analyse 128 s of data. Necessarily, this results in a significant increase in the time required to analyse the likelihood and hence overall wall-time. To mitigate this, we use the Reduced-Order-Quadrature (ROQ) method (Antil et al. 2012; Canizares et al. 2013, 2015; Smith et al. 2016; Qi & Raymond 2020) with the basis provided by Baylor, Smith & Chase (2019) to decrease the per-likelihood evaluation cost.

The simulated signal has small spin components aligned along the angular momentum axis, an arbitrarily selected choice of tidal deformability parameters, and nearly equal-mass components. In the specific noise realization used, the network matched-filter S/N is ∼18. We analyse the signal using both the Dynesty and Bilby-MCMC samplers using the configurations described in Table 2. The analyses are identical to those of the BBH A analysis, except, we use the `IMRPhenomPv2_NRTidal`waveform model (through the ROQ basis), use only distance marginalization, and restrict the spins to a low-spin configuration (dimensionless spin magnitude less than 0.05; Abbott et al. 2019).

As with the BBH case, the Bayesian evidence estimates (see Table 2 for the signal versus noise Bayes factor) disagree. Again, we conclude this is due to the known bias in the parallel-tempered evidence estimate. The posterior distributions from the Dynesty and Bilby-MCMC are statistically identical, except for the inclination parameter $\theta_{\rm JN}$. In Fig. 6, we reproduce histograms for selected parameters of typical astrophysical inference visually demonstrating the agreement and inclination difference. The cause of the difference in inferred inclination is not yet fully understood, but we note that the difference is only marginally above our threshold for statistically identical. Comparing individual re-analyses between the two samplers, the difference persists suggesting it is systematic and not a random fluctuation. We note this is an instance where the posterior is bimodal and speculate this could be a symptom of the Dynesty nested sampling failing to fully explore both modes. However, the difference is sufficiently small for us to conclude the underlying conclusions about the source (i.e. the 90 per cent credible intervals) are robust, while the posterior shape is subject to some sampling error (from one or both samplers).

Due to the larger S/N of the fiducial BNS, and the increase in dimension of the prior, the efficiency of both the Dynesty and Bilby-MCMC samplers is reduced compared to that of the fiducial BBH. The ratio of efficiencies is also increased: the Dynesty sampler is ∼30 times more efficient in this case. As with the BBH analyses, this efficiency does not directly translate into wall-time savings due to the different parallelization approaches. However, the efficiency is significant. In future work, we aim to improve the choice of parametrization and proposals to improve the efficiency of the Bilby-MCMC sampler.

5 SUMMARY

We introduce BILBY-MCMC, a parallel-tempered ensemble sampler with problem-specific and machine learning based proposals. The BILBY-MCMC sampler is the first MCMC sampler implemented in the BILBY (Ashton et al. 2019) inference package with demonstrated performance for analysing CBC events observed by ground-based gravitational-wave detectors. We demonstrate, using both comparisons to known results and cross-sampler comparisons, that the posterior samples are unbiased. Compared to the DYNESTY nested sampling algorithm, BILBY-MCMC suffers a known bias in its estimation of the Bayesian evidence when the number of parallel-tempered chains, $n_{\rm temps}$, is small. Increasing $n_{\rm temps}$ reduces the bias, but at the cost of posterior-sampling efficiency. We introduce a method to re-sample from the tempered chains, recovering some of this inefficiency, but find it provides little improvement for typical CBC inference problems. We conclude that BILBY-MCMC is ideal for problems in which only the posterior distribution is of interest, but that nested sampling approaches should be preferred when evidence calculations are required. This makes BILBY-MCMC unsuitable for model-comparison via a Bayes factor (MacKay 2003). Instead, one may wish to develop a hyper-model where the model is treated as a random variable (see e.g. the Reverse Jump Markov Chain Monte Carlo approach described in Cornish & Littenberg 2007)).

BILBY-MCMC can be trivially and asynchronously parallelized. This enables it to be configured to leverage High-Throughput Computing environments to reduce the wall-time. That the parallelization is asynchronous makes it ideal for utilizing non-interacting distributed computing such as the Open Science Grid (Pordes et al. 2007; Sfiligoi et al. 2009). By comparison, nested sampling approaches can be parallelized solely through the use of multiprocessing. Smith et al. (2020) demonstrated massive scaling of the DYNESTY (Speagle 2020) sampler to many hundreds of cores; BILBY-MCMC cannot similarly be scaled due to the fundamental limit of the burn-in inefficiency. However, the Smith et al. (2020) approach requires synchronized access to a High-Performance Computing environment in which the communication times between cores is rapid.

BILBY-MCMC provides the user access to a modular library of proposal distributions that can be chained together. The choice of parametrization and proposals has a significant effect on the efficiency of the sampler. We anticipate further development in both these aspects will improve the sampler efficiency resulting in reduced wall-time. Users adapting BILBY-MCMC to other astrophysical inference problems can define their own sets of proposal distributions and easily implement new problem-specific proposals by sub-classing the existing software.

ACKNOWLEDGEMENTS

We thank Michael Williams, John Veitch, Ben Farr, and Moritz Hübner for useful comments during the development of this work. We thank Will Farr, whose review of this work led to several improvements and clarifications in our exposition. CT acknowledges support of the National Science Foundation, and the LIGO Laboratory. We are grateful for computational resources provided by Cardiff University, and funded by an STFC grant ST/I006285/1 supporting UK Involvement in the Operation of Advanced LIGO. We are also grateful to computing resourced provided by the LIGO Laboratory computing clusters at California Institute of Technology and LIGO Hanford Observatory supported by National Science Foundation Grants PHY-0757058 and PHY-0823459. This work makes use of the SCIPY (Virtanen et al. 2020), NUMPY (Oliphant 2006; Van Der Walt, Colbert & Varoquaux 2011; Harris et al. 2020), and PESUMMARY (Hoy & Raymond 2020) packages for data analysis and visualization.

DATA AVAILABILITY

No new data were generated or analysed in support of this research. The scripts used to perform all verification checks and additional figures are available from git.ligo.org/gregory.ashton/bilby_mcmc_validation.

REFERENCES

Aasi J. et al., 2015, Class. Quantum Gravity, 32, 074001
Abbott B. P. et al., 2016, Phys. Rev. Lett., 116, 241102
Abbott B. P. et al., 2017, Nature, 551, 85
Abbott B. P. et al., 2018, Phys. Rev. Lett., 121, 161101
Abbott B. P. et al., 2019, Phys. Rev. X, 9, 011001
Abbott B. P. et al., 2020, Living Rev. Relativ., 23, 3
Acernese F. et al., 2015, Class. Quantum Gravity, 32, 024001
Antil H., Field S. E., Herrmann F., Nochetto R. H., Tiglio M., 2012, preprint (arXiv:1210.0577)
Ashton G. et al., 2019, ApJS, 241, 27
Aso Y. et al., 2013, Phys. Rev. D, 88, 043007
Baylor A., Smith R., Chase E., 2019, Imrphenompv2_nrtidal_gw190425_narrow_mc. Available at https://zenodo.org/record/3478659#.YSReu_ozbIU
Biwer C. M., Capano C. D., De S., Cabero M., Brown D. A., Nitz A. H., Raymond V., 2019, PASP, 131, 024503
Canizares P., Field S. E., Gair J. R., Tiglio M., 2013, Phys. Rev. D, 87, 124005
Canizares P., Field S. E., Gair J., Raymond V., Smith R., Tiglio M., 2015, Phys. Rev. Lett., 114, 071104
Christensen N., Meyer R., 1998, Phys. Rev. D, 58, 082001
Cook S. R., Gelman A., Rubin D. B., 2006, J. Comput. Graph. Stat., 15, 675
Cornish N. J., Littenberg T. B., 2007, Phys. Rev. D, 76, 083006
Cutler C., Flanagan E. E., 1994, Phys. Rev. D, 49, 2658
Dietrich T., Bernuzzi S., Tichy W., 2017, Phys. Rev. D, 96, 121501
Dietrich T. et al., 2019, Phys. Rev. D, 99, 024029
Durkan C., Bekasov A., Murray I., Papamakarios G., 2020, nflows: Normalizing Flows in PyTorch. Available at https://github.com/bayesiains/nflows
Earl D. J., Deem M. W., 2005, Phys. Chem. Chem. Phys., 7, 3910
Farr W. M., 2014, Technical Report LIGO-T1400460, Marginalisation of the Time and Phase Parameters in CBC Parameter Estimation
Farr B., Ochsner E., Farr W. M., O'Shaughnessy R., 2014, Phys. Rev. D, 90, 024018
Favata M., 2014, Phys. Rev. Lett., 112, 101101
Feeney S. M., Peiris H. V., Williamson A. R., Nissanke S. M., Mortlock D. J., Alsing J., Scolnic D., 2019, Phys. Rev. Lett., 122, 061105
Flanagan É. É., Hinderer T., 2008, Phys. Rev. D, 77, 021502
Foreman-Mackey D., Hogg D. W., Lang D., Goodman J., 2013, PASP, 125, 306
Fowlie A., Handley W., Su L., 2020, MNRAS, 497, 5256
Gabbard H., Messenger C., Heng I. S., Tonolini F., Murray-Smith R., 2019, preprint (arXiv:1909.06296)
Gelman A. et al., 1996, Bayesian Stat., 5, 42
Gilks W. R., Roberts G. O., Sahu S. K., 1998, J. Am. Stat. Assoc., 93, 1045
Goggans P. M., Chi Y., 2004, in Gary J. E., Yuxiang Z., eds, AIP Conf. Ser. Vol. 707, 23rd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering. Am. Inst. Phys., New York, p. 59
Goodman J., Weare J., 2010, Commun. Appl. Math. Comput. Sci., 5, 65
Graff P., Feroz F., Hobson M. P., Lasenby A., 2012, MNRAS, 421, 169
Green S. R., Gair J., 2021, Machine Learning: Science and Technology. IOP Publishing
Green S. R., Simpson C., Gair J., 2020, Phys. Rev. D, 102, 104057
Haario H., Saksman E., Inen J. T., 2001, Bernoulli, 7, 223
Hannam M., Schmidt P., Bohé A., Haegel L., Husa S., Ohme F., Pratten G., Pürrer M., 2014, Phys. Rev. Lett., 113, 151101

Harris C. R. et al., 2020, Nature, 585, 357
Hastings W. K., 1970, Biometrika, 57, 97
Hoffman M., Sountsov P., Dillon J. V., Langmore I., Tran D., Vasudevan S., 2019, preprint (arXiv:1903.03704)
Hogg D. W., Foreman-Mackey D., 2018, ApJS, 236, 11
Hoy C., Raymond V., 2020, preprint (arXiv:2006.06639)
Jacob P. E., O'Leary J., Atchadé Y. F., 2020, J. R. Stat. Soc. B, 82, 543
Kulkarni S., Capano C. D., 2020, preprint (arXiv:2011.13764)
Lange J., O'Shaughnessy R., Rizzo M., 2018, preprint (arXiv:1805.10457)
Lartillot N., Philippe H., 2006, Systematic Biol., 55, 195
Link W. A., Eaton M. J., 2012, Methods Ecol. Evol., 3, 112
Littenberg T. B., Cornish N. J., 2009, Phys. Rev. D, 80, 063007
MacKay D. J., 2003, Information Theory, Inference and Learning Algorithms. Cambridge Univ. Press, Cambridge
Maturana-Russel P., Meyer R., Veitch J., Christensen N., 2019, Phys. Rev. D, 99, 084006
Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., Teller E., 1953, J. Chem. Phys., 21, 1087
Moss A., 2020, MNRAS, 496, 328
Oliphant T. E., 2006, A Guide to NumPy. Trelgol Publishing, USA
Pankow C., Brady P., Ochsner E., O'Shaughnessy R., 2015, Phys. Rev. D, 92, 023002
Papamakarios G., Nalisnick E., Rezende D. J., Mohamed S., Lakshminarayanan B., 2019, preprint (arXiv:1912.02762)
Parzen E., 1962, Ann. Math. Stat., 33, 1065
Paszke A. et al., 2019, Advances in Neural Information Processing Systems 32. Curran Associates, Inc., New York
Pedregosa F. et al., 2012, preprint (arXiv:1201.0490)
Pordes R. et al., 2007, J. Phys. Conf. Ser., 78, 012057
Qi H., Raymond V., 2020, preprint (arXiv:2009.13812)
Raymond V., Farr W. M., 2014, preprint (arXiv:1402.0053)
Roberts G. O., Rosenthal J. S., 2001, Stat. Sci., 16, 351
Roberts G. O., Gelman A., Gilks W. R., 1997, Ann. Appl. Probab., 7, 110
Romero-Shaw I. M. et al., 2020, MNRAS, 499, 3295
Rosenblatt M., 1956, Ann. Math. Stat., 27, 832
Rosenbrock H. H., 1960, Comput. J., 3, 175
Salomone R., South L., Drovandi C., Kroese D., 2018, preprint (arXiv:1805.03924)
Schmidt P., Hannam M., Husa S., 2012, Phys. Rev. D, 86, 104063
Scott D. W., 2015, Multivariate Density Estimation: Theory, Practice, and Visualization. John Wiley & Sons, Hoboken, New Jersey
Sfiligoi I., Bradley D. C., Holzman B., Mhashilkar P., Padhi S., Wurthwein F., 2009, Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering, Vol. 2. p. 428
Sharma S., 2017, ARA&A, 55, 213
Singer L. P., Price L. R., 2016, Phys. Rev. D, 93, 024013
Singer L. P. et al., 2016, ApJ, 829, L15
Skilling J., 2006, Bayesian Anal., 1, 833
Smith R., Field S. E., Blackburn K., Haster C.-J., Pürrer M., Raymond V., Schmidt P., 2016, Phys. Rev. D, 94, 044031
Smith R. J. E., Ashton G., Vajpeyi A., Talbot C., 2020, MNRAS, 498, 4492
Speagle J. S., 2020, MNRAS, 493, 3132
Talts S., Betancourt M., Simpson D., Vehtari A., Gelman A., 2018, preprint (arXiv:1804.06788)
Ter Braak C. J. F., 2006, Stat. Comput., 16, 239
ter Braak C. J., Vrugt J. A., 2008, Stat. Comput., 18, 435
Thrane E., Talbot C., 2019, Publ. Astron. Soc. Aust., 36, e010
Van Der Walt S., Colbert S. C., Varoquaux G., 2011, Comput. Sci. Eng., 13, 22
Veitch J., Del Pozzo W., 2013, Technical Report LIGO-T1300326, Analytic Marginalisation of Phase Parameter
Veitch J., Vecchio A., 2008, Phys. Rev. D, 78, 022001
Veitch J. et al., 2015, Phys. Rev. D, 91, 042003
Virtanen P. et al., 2020, Nat. Methods, 17, 261
Vousden W. D., Farr W. M., Mandel I., 2016, MNRAS, 455, 1919
Wade L., Creighton J. D. E., Ochsner E., Lackey B. D., Farr B. F., Littenberg T. B., Raymond V., 2014, Phys. Rev. D, 89, 103012
Williams M. J., Veitch J., Messenger C., 2021, Phys. Rev. D, 103, 103006
Xie W., Lewis P. O., Fan Y., Kuo L., Chen M.-H., 2010, Systematic Biol., 60, 150

APPENDIX A: THE JS-DIVERGENCE CRITERIA

As described in Romero-Shaw et al. (2020), we use the 1D JSD (maximized over all dimensions) to quantify the agreement between sets of posterior samples. In that work, a threshold of 2 mb was established for the maximum JSD[4]: Above this value, the differences between posteriors where deemed statistically significant. Here, we extend that analysis. We simulate pairs of posterior samples from the 15D unimodal Gaussian distribution (cf. Section 3.3) varying the number of samples drawn in each case. We find a strong correlation between the number of samples and the inverse of the maximum JSD (Fig. A1). This demonstrates that, while appropriate for sample sizes

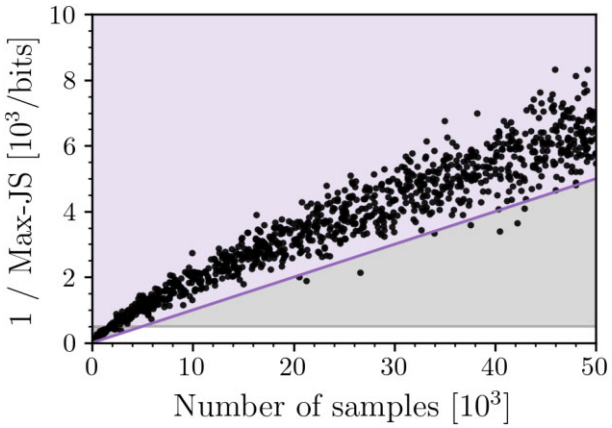

Figure A1. The maximum JSD for 1000 pairs of posteriors drawn form a 15D unimodal Gaussian distribution. We vary the number of samples drawn in each simulation. The horizontal grey line indicates the 2-mb threshold established in Romero-Shaw et al. (2020). The purple curve is the new threshold given in equation (A1).

of a few thousand, the original threshold is overly conservative for small samples sized and too liberal for larger sample sizes.

To better capture the correlations observed in the simulated data, we introduce a new threshold:

$$\text{maximum JSD} \leq \frac{10}{n^{\text{eff}}_{\text{samples}}}. \tag{A1}$$

This threshold is demonstrated in Fig. A1 as the purple shaded region.

For the simulated 15-D system, we see maximum JSD values as large as equation (A1) a few times in the 1000 simulations. This threshold falsely identifies statistical differences between the sets of posterior samples in our simulation as a rate of ∼0.1 per cent. In this sense, it can be used as a conservative bound: If the maximum JSD between samplers is found to be larger than the prediction of equation (A1), this highlights an area of concern warranting further study.

We note that a better fit to the lower bound on the inverse maximum JSD could be found (e.g. by a probability-of-failure based rule), but equation (A1) is easy to remember and hence provides a good rule of thumb.

[4]The maximum over the set of sampled parameters.

This paper has been typeset from a TEX/LATEX file prepared by the author.