Accelerating parameter inference with graphics processing units

D. Wysocki, R. O'Shaughnessy, and Jacob Lange
Center for Computational Relativity and Gravitation, Rochester Institute of Technology, Rochester, New York 14623, USA

Yao-Lung L. Fang
Computational Science Initiative, Brookhaven National Laboratory, Upton, New York 11973, USA and National Synchrotron Light Source II, Brookhaven National Laboratory, Upton, New York 11973, USA

(Received 14 February 2019; published 16 April 2019)

Gravitational wave Bayesian parameter inference involves repeated comparisons of gravitational wave data to generic candidate predictions. Even with algorithmically efficient methods such as RIFT or reduced-order quadrature, the time needed to perform these calculations and the overall computational cost can be significant compared to the minutes to hours needed to achieve the goals of low-latency multimessenger astronomy. By translating some elements of the RIFT algorithm to operate on graphics processing units, we demonstrate substantial performance improvements, enabling dramatically reduced overall cost and latency.

DOI: 10.1103/PhysRevD.99.084026

I. INTRODUCTION

The Advanced LIGO [1] and Virgo [2] ground-based gravitational wave (GW) detectors have identified several coalescing compact binaries [3–8]. The properties of the sources responsible have been inferred via the Bayesian comparison of each source with candidate gravitational wave signals [3–12]. Many more are expected as observatories reach design sensitivity [13]. Both to handle the rapid pace of discovery and particularly to enable coordinated multimessenger follow-up, GW observatories should be able to reconstruct the evidence for a signal being present in the data along with the full source parameters of coalescing binaries as fast as possible [14,15]. However, particularly when using the best-available waveform models, these calculations can be very costly. For binary neutron stars (BNS), detailed inferences even using simplified waveforms take weeks. Even for massive binary black holes (BBH), these calculations can take months for costly time-domain effective-one-body models which incorporate the effects of precession.

Several strategies have been developed to reduce the computational cost of parameter estimation [9,16–20]. Approaches that have appeared in the literature include generating the approximate solutions more quickly [21–26]; interpolating some combination of the waveform or likelihood [9,23,24,27–31]; or adopting a sparse representation to reduce the computational cost of data handling [9,16,18,20,26,32,33]. Not published but also important are code optimization projects that improve infrastructure, such as better parallelization for LALINFERENCE [12,34]. Some methods, however, achieve rapid turnaround through simplifying approximations. The RIFT/rapidPE strategy described in [9,10,35,36] eschews these simplifications, performing embarrassingly parallel inferences even for costly models. However, the method as previously implemented still had a significant net computational cost and noticeable startup time, limiting its diverse potential applications. Previously, B. Miller developed a custom-coded implementation of the relevant likelihood on graphics processing units (GPUs), suggesting substantial performance improvements were possible [37,38]. The PYCBC search code [39] and its PYCBC-inference extension [40] can also make use of several GPU hardware-accelerated operations. In this paper, we describe a variant of the RIFT approach which flexibly translates one of its algorithms to operate on GPUs and adjusts its workflow to exploit the speed improvements this reimplementation affords.

This paper is organized as follows. In Sec. II, we review the underlying marginalized likelihood calculations used by RIFT and their updated GPU implementation. In Sec. III, we quantify the improved performance of our GPU-accelerated code, while assessing operating settings which facilitate increased performance. In Sec. IV, we describe the performance of the end-to-end pipeline on the synthetic nonspinning signals used in Sec. III. In Sec. V, we summarize our results and discuss their potential applications to future GW source and population inference.

II. METHODS

A. Parameter inference with the RIFT likelihood

ILE—a specific algorithm to "integrate (the likelihood) over extrinsic parameters"—provides a straightforward and efficient mechanism to compare any specific candidate gravitational wave source with real or synthetic data [9,35,36,41], by marginalizing the likelihood of the data over the seven coordinates characterizing the spacetime coordinates and orientation of the binary relative to the Earth. Specifically the likelihood of the data given Gaussian noise, relative to Gaussian noise, has the form (up to normalization)

$$\ln \mathcal{L}(\boldsymbol{\lambda},\theta) = -\frac{1}{2}\sum_k \langle h_k(\boldsymbol{\lambda},\theta) - d_k | h_k(\boldsymbol{\lambda},\theta) - d_k\rangle_k - \langle d_k|d_k\rangle_k, \tag{1}$$

where h_k are the predicted responses of the kth detector due to a source with parameters ($\boldsymbol{\lambda}$, θ) and d_k are the detector data in each instrument k; $\boldsymbol{\lambda}$ denotes the combination of redshifted mass M_z and the remaining parameters needed to uniquely specify the binary's dynamics; θ represents the seven extrinsic parameters (four spacetime coordinates for the coalescence event and three Euler angles for the binary's orientation relative to the Earth); and $\langle a|b\rangle_k \equiv \int_{-\infty}^{\infty} 2df\tilde{a}(f)^*\tilde{b}(f)/S_{h,k}(|f|)$ is an inner product implied by the kth detector's noise power spectrum $S_{h,k}(f)$. In practice we adopt both low- and high-frequency cutoffs $f_{\max}$, $f_{\min}$ so all inner products are modified to

$$\langle a|b\rangle_k \equiv 2\int_{|f|>f_{\min},|f|<f_{\max}} df \frac{[\tilde{a}(f)]^*\tilde{b}(f)}{S_{h,k}(|f|)}. \tag{2}$$

The joint posterior probability of $\boldsymbol{\lambda}$, θ follows from Bayes' theorem:

$$p_{\rm post}(\boldsymbol{\lambda},\theta) = \frac{\mathcal{L}(\boldsymbol{\lambda},\theta)p(\theta)p(\boldsymbol{\lambda})}{\int d\boldsymbol{\lambda}d\theta\mathcal{L}(\boldsymbol{\lambda},\theta)p(\boldsymbol{\lambda})p(\theta)}, \tag{3}$$

where $p(\theta)$ and $p(\boldsymbol{\lambda})$ are priors on the (independent) variables θ, $\boldsymbol{\lambda}$. For each $\boldsymbol{\lambda}$, we evaluate the marginalized likelihood

$$\mathcal{L}_{\rm marg} \equiv \int \mathcal{L}(\boldsymbol{\lambda},\theta)p(\theta)d\theta \tag{4}$$

via direct Monte Carlo integration over almost all parameters θ, where $p(\theta)$ is uniform in four-volume and source orientation. For the event time parameter, we marginalize by direct quadrature, for each choice of the remaining Monte Carlo parameters. For the remaining dimensions, to evaluate the likelihood in regions of high importance, we use an adaptive Monte Carlo integrator as described in [9].

This marginalized likelihood can be evaluated efficiently by generating the dynamics and outgoing radiation in all possible directions once and for all for fixed $\boldsymbol{\lambda}$, using a spherical harmonic decomposition. Using this cached information effectively, the likelihood can be evaluated as a function of θ at a very low computational cost. A dimensionless, complex gravitational-wave strain

$$h(t,\vartheta,\phi;\boldsymbol{\lambda}) = h_+(t,\vartheta,\phi;\boldsymbol{\lambda}) - ih_\times(t,\vartheta,\phi;\boldsymbol{\lambda}) \tag{5}$$

can be expressed in terms of its two fundamental polarizations h_+ and $h_\times$. Here, t denotes time, and ϑ and ϕ are the polar and azimuthal angles for the direction of gravitational wave propagation away from the source. The complex gravitational-wave strain can be written in terms of spin-weighted spherical harmonics ${}_{-2}Y_{\ell m}(\vartheta,\phi)$ as

$$h(t,\vartheta,\phi;\boldsymbol{\lambda}) = \sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{\ell}\frac{D_{\rm ref}}{D}h^{\ell m}(t;\boldsymbol{\lambda})_{-2}Y_{\ell m}(\vartheta,\phi), \tag{6}$$

where the sum includes all available harmonic modes $h^{\ell m}(t;\boldsymbol{\lambda})$ made available by the model; where $D_{\rm ref}$ is a fiducial reference distance; and where D, the luminosity distance to the source, is one of the extrinsic parameters.

Following Pankow *et al.* [9], we substitute expression (6) for $h_{\ell m}$ into the expression $h_k(t) = F_{+,k}h_+(t_k) + F_{\times,k}h_\times(t_k)$ for the detector response h_k, where $t_k = t_c - \vec{x}_k\cdot\hat{n}$ is the arrival time at the kth detector (at position $\vec{x}_k$) for a plane wave propagating along $\hat{n}$ [9]. We then substitute these expressions for h_k into the likelihood function (1) thereby generating [9]

$$\begin{aligned}\ln \mathcal{L}(\boldsymbol{\lambda},\theta) = {}& (D_{\rm ref}/D){\rm Re}\sum_k\sum_{\ell m}(F_{k-2}Y_{\ell m})^*Q_{k,lm}(\boldsymbol{\lambda},t_k)\\ &-\frac{(D_{\rm ref}/D)^2}{4}\sum_k\sum_{\ell m\ell' m'}\\ &\times[|F_k|^2[{}_{-2}Y_{\ell m}]^*{}_{-2}Y_{\ell' m'}U_{k,\ell m,\ell' m'}(\boldsymbol{\lambda})\\ &+{\rm Re}(F_{k-2}^2Y_{\ell m-2}Y_{\ell' m'}V_{k,\ell m,\ell' m'})],\end{aligned} \tag{7}$$

where $F_k = F_{+,k} - iF_{\times,k}$ are the complex-valued detector response functions of the kth detector [9] and the quantities Q, U, V depend on h and the data as

$$\begin{aligned}Q_{k,\ell m}(\boldsymbol{\lambda},t_k) &\equiv \langle h_{\ell m}(\boldsymbol{\lambda},t_k)|d\rangle_k\\ &= 2\int_{|f|>f_{\rm low}}\frac{df}{S_{n,k}(|f|)}e^{2\pi i f t_k}\tilde{h}^*_{\ell m}(\boldsymbol{\lambda};f)\tilde{d}(f),\end{aligned} \tag{8a}$$

$$U_{k,\ell m,\ell' m'}(\boldsymbol{\lambda}) = \langle h_{\ell m}|h_{\ell' m'}\rangle_k, \tag{8b}$$

$$V_{k,\ell m,\ell' m'}(\boldsymbol{\lambda}) = \langle h^*_{\ell m}|h_{\ell' m'}\rangle_k. \tag{8c}$$

In RIFT, the marginalization in Eq. (4) over extrinsic parameters is performed by evaluating this likelihood $\ln\mathcal{L}(\boldsymbol{\lambda},\theta)$ on long arrays θ_α of extrinsic parameters, including D_α but excluding time. Treating the block of quantities that arise in one sequence of evaluations together, the expressions Q, F, U, V, and D can be considered as multidimensional arrays. For example, $F_k(\theta_\alpha)$ is a matrix indexed over k (detectors) and α (extrinsic parameters); ${}_{-2}Y_{lm}(\theta_k)$ is a matrix of shape (modes) $\times$ (extrinsic); and D_α is a one-dimensional array over α. By contrast, the U, V are arrays of shape (detectors) $\times$ (modes)^2 and are independent of extrinsic parameters. While Q depends explicitly on time, and varies rapidly on short timescales, F varies on the Earth's rotation timescale, so we approximate F as time independent.

At a fixed set of intrinsic parameters $\boldsymbol{\lambda}$, ILE will repeatedly evaluate the time-marginalized likelihood for each candidate Monte Carlo set of extrinsic parameters θ:

$$\mathcal{L}_{\mathrm{margT}} \equiv \int \mathcal{L}\frac{dt}{T}, \tag{9}$$

where T is a small coincidence window consistent with the candidate event time; here and previously, $T = 0.3$ s. Because we treat the parameter asymmetrically when marginalizing over all extrinsic parameters, we likewise organize a multidimensional array representation of Q to emphasize the special role of time: for each k, l, m we construct a uniformly sampled time series $Q_{k,lm}(\boldsymbol{\lambda}, \tau)$ vs τ, truncating it to a narrow coincidence window. In effect, we represent Q by a matrix $Q_{k,lm}(\boldsymbol{\lambda}, t + \hat{n}\cdot x_k)$ with shape (detectors) $\times$ (modes) $\times$ (extrinsic) $\times$ (time). In terms of these multidimensional arrays, rewriting sums as matrix operations, the likelihood can be equivalently expressed as

$$\begin{aligned}\ln\mathcal{L} = &\frac{D_{\mathrm{ref}}}{D}\mathrm{Re}[(FY)^\dagger Q]\\ &- \frac{D_{\mathrm{ref}}^2}{4D^2}[(FY)^\dagger UFY + (FY)^T VFY],\end{aligned} \tag{10}$$

where this symbolic expression employs an implicit index-summation convention such that all naturally paired indices are contracted. The result is an array of shape (time) $\times$ (extrinsic).

B. Accelerated evaluation via efficient multiplication

To accelerate the code, after precomputing the inner products U, V, Q, we simply shift them to the graphics card, and then carry out all calculations necessary to implement Eqs. (10) and (9) on the GPU. These arrays are only a few kilobytes. We then construct blocks of 10^4 random extrinsic parameters θ_α with the CPU; transfer them to the board; and use on-board code to construct 10^4 values for $\ln\mathcal{L}_{\mathrm{marg}}$. To enable this implementation with portable code, we use CUPY [42], a drop-in replacement for equivalent NUMPY code used for the CPU-based version of ILE. For the most costly part of the calculation—the inner products necessary to evaluate $Q_{lm}(t)$, accounting for distinct time-of-flight-dependent time windows for each interferometer's data—we use a custom Compute Unified Device Architecture (CUDA) kernel to perform the necessary matrix multiplication. With these changes alone, the likelihood evaluation is roughly 60× faster on equivalent hardware. After this update, individual likelihood evaluation costs are not the performance- or cost-limiting feature of RIFT-based source parameter inference. Instead, the overhead associated with the adaptive Monte Carlo integrator and with the (CPU-based) inner product evaluations for Q, U, V dominate our computational cost.

C. Tradeoff between Monte Carlo integration and accuracy

Each marginalized likelihood evaluation has a relative uncertainty of order $1/\sqrt{n_{\mathrm{eff}}}$, where the number of effective samples $n_{\mathrm{eff}} = \epsilon N_{\mathrm{it}}$ increases linearly with the total number N_{it} of Monte Carlo evaluations performed by ILE. Therefore, to increase (decrease) our accuracy for each likelihood evaluation by a factor of A will require a factor of A^2 more (fewer) iterations. We adopt a fixed threshold on n_{eff}.

We do benefit slightly by reusing the adaptive Monte Carlo integrator for each extrinsic point $\boldsymbol{\lambda}$. Since the integrator has already identified the likely range of sky locations and distances on the first iteration, each subsequent evaluation of the marginalized likelihood can converge marginally more quickly.

III. ANALYSIS OF MARGINALIZED LIKELIHOOD EVALUATION COST

We illustrate the code using two synthetic signals: a binary black hole with masses $m_{1,2} = 35$, $30\ M_\odot$ and a binary neutron star with masses $1.4\ M_\odot$ and $1.35\ M_\odot$. We perform this analysis on heterogeneous LVC Collaboration computing resources described in more detail in the Appendix, to assess our variable performance across architectures. Notably, we investigate the following GPU options: (a) GTX 1050 Ti at LIGO-CIT, available in large numbers; (b) V100, available on selected high-performance machines; and (c) GTX 1050 at LIGO-LHO. Unless otherwise noted in the text, we will discuss code configurations using CPU only and GPU (b) using 4096 Hz sampling and only the $\ell = 2$, $m = \pm 2$ modes. In this section, we conservatively report computational costs for parameters $\boldsymbol{\lambda}$ which are close to or within the posterior. Because of their consistency with the data, the marginalized likelihood calculations converge the most slowly, as they have the best-determined extrinsic parameters.

A. Synthetic source generation and analysis settings

We generate synthetic data for a two-detector LIGO configuration, assuming operation at initial (BBH) or advanced (BNS) design sensitivity. Both signals are generated with SEOBNRv4 effective-one-body waveform approximation [43], with a zero-noise data realization. To qualitatively reproduce the noise power spectrum and amplitude of typical binary black hole observations in O1 and O2, the binary black hole signal has source distance 200 Mpc, chosen so SNR $\simeq 14$. Similarly, to qualitatively reproduce the analysis of GW170817, our synthetic binary neutron star is placed 100 Mpc away, so the signal amplitude is SNR $\simeq 31$.

We perform a multistage iterative RIFT analysis of these signals with the time-domain SEOBNRv4 (BBH) or TaylorT4 [44] (BNS) approximation, under the simplifying assumption that both objects are point particles with zero spin. For both sources, we adopt the fiducial distance prior $p(D_L) \propto D_L^2$, selecting a maximum distance roughly 5 times larger than the known source distance: 1 Gpc for the binary black hole and 500 Mpc for the neutron star. We adopt a uniform prior on the component (redshifted) masses $m_{i,z}$ (and, when appropriate, spins $\chi_{i,z}$). Unless otherwise noted, all mass quantities described in this work are redshifted masses $m_{k,z} = (1+z)m_k$. For the binary black hole we generate the signal with $f_{\mathrm{min}} = 8$ Hz and analyze the signal with $f_{\mathrm{min}} = 10$ Hz; for the binary neutron star, we generate the signal with $f_{\mathrm{min}} = 20$ Hz and analyze it with $f_{\mathrm{min}} = 23$ Hz.

B. Massive compact binary black holes

Previously, each instance of ILE examined one intrinsic point $\boldsymbol{\lambda}$. The overall cost of this ILE evaluation involved three parts: a startup cost, a setup cost, and a Monte Carlo integration cost. The startup cost τ_{start} is associated with code setup followed by reading, data conditioning, and Fourier transforming the data. The setup cost τ_{setup} arises from waveform generation and the inner products U, V, Q. Finally, the Monte Carlo cost $\tau_{\mathrm{mc}} = N_{\mathrm{ad}}\tau_{\mathrm{ad}} + N_{\mathrm{it}}\tau_{\mathrm{eval}}$ increases with the number of Monte Carlo iterations N_{it} with and the number of times N_{ad} the sampling prior used in adaptive Monte Carlo (MC) integrator is regenerated from the most recent n_{chunk} data points, in proportion to their cost. In the standard configuration, the adaptive sampling prior is regenerated every n_{chunk} iterations (i.e., $N_{\mathrm{ad}} \simeq N_{\mathrm{it}}/n_{\mathrm{chunk}}$), regenerating a sampling prior in two sky location parameters and distance. The choice of one $\boldsymbol{\lambda}$ for each instance of ILE was due to the substantial time τ_{MC}, which vastly dominated the overall computational cost. For example, for a typical analysis of a short signal—a typical binary black hole signal with $f_{\mathrm{min}} = 20$ Hz and $m_1 \simeq m_2 \simeq 30\ M_\odot$—this version has cost elements as shown in Table I, based on the assumption that the MC terminates after $N_{\mathrm{it}} \simeq 2 \times 10^6$ iterations using $n_{\mathrm{chunk}} \simeq 10^4$. The marginalized likelihood described above converges incredibly rapidly to a small value if the candidate model is inconsistent with a signal (or the absence thereof) in the data, with only $N_{\mathrm{it}} \simeq O(10^4)$ evaluations needed.

With the new low-cost $\mathcal{L}_{\mathrm{margT}}$ evaluations, however, the startup and setup costs τ_{start}, τ_{setup} now can be comparable to or in excess of the total time used to evaluate the marginalized likelihood $\mathcal{L}_{\mathrm{marg}}$. In fact, the total time per Monte Carlo evaluation $\tau_{\mathrm{it}} = \tau_{\mathrm{manage}} + \tau_{\mathrm{eval}}$ spent in general-purpose overhead τ_{manage} will be much larger than the time τ_{eval} spent carrying out scientific calculations by evaluating the likelihood. For these reasons, we reorganize the workflow, so each instance of ILE loops over N_{eval} different choices for $\boldsymbol{\lambda}$. Additionally, particularly for massive binary black holes, we investigate two additional performance improvements. First and foremost, we lower the sampling rate, thus lowering the startup cost τ_{start} and particularly setup cost τ_{setup}. Previously and out of an abundance of caution, ILE employed a sampling rate $1/\Delta t = 16384$ Hz for all calculations, but terminated all inner products in Eq. (2) at roughly $f_{\mathrm{max}} = 2048$ Hz or less. Reducing the sampling rate by a factor s will reduce the cost of all operations with time series—they are shorter.

TABLE I. Profiling performance: Binary black holes. Evaluation costs for the marginalized likelihood on default hardware, for a two-mode system $(l, m) = \pm 2$ analyzing $T = 8$ s of data with a massive binary black hole $m_1 = 35\ M_\odot$, $M_2 = 30\ M_\odot$. The last column indicates peak GPU utilization.

Version	S rate [Hz]	Modes [m]	τ_{start} [s]	τ_{setup} [s]	τ_{ad}	$\tau_{\mathrm{it,like}}$ [μs]	$\tau_{\mathrm{it,rest}}$ [μs]	$\frac{T_{\mathrm{ILE}}}{N_{\mathrm{eval}}}$ [s]	GPU use [%]
CPU	16384	± 2	20	2.4		540	20	690	
	4096	± 2	20				20		
CPU	16384	$\pm 2, \pm 1$	20	1.5		680	20	1060	
	4096	$\pm 2, \pm 1$	20				20		
GPU (a)	16384	± 2	20					270	
	4096	± 2	20					45	
GPU (b)	16384	± 2	20	1.8	1	0.85	20	28	15
	4096	± 2	20	1.2	1	0.75	20	25	
GPU (b)	16384	$\pm 2, \pm 1$	20	1		4.2	20	38	
	4096	$\pm 2, \pm 1$	20	1		2.5	20	35	
GPU (c)	16384	± 2	20	6		18	58	160	
	4096	± 2	20	3.7		11	58	140	$\simeq 50$

Depending on the relative cost of overhead vs array operations, $\tau_{\rm setup}$ *and* the cost per evaluation $\tau_{\rm eval}$ decrease modestly because of this effect. Also following Pankow *et al.*, to ensure safe inner products over short data sets in the presence of very narrow spectral lines, we operated on a significantly padded data buffer and modified the noise power spectrum by truncating the inverse power spectrum to a finite response time. For binary black holes, the degree of padding was significantly in excess of the signal duration. Instead and following LALINFERENCE [12], we adopt a much shorter inverse spectrum truncation length, a much shorter padding buffer, and a Tukey window applied to the data to remove discontinuities at the start and end of each segment. Reducing the duration of data analyzed by a factor s' will reduce the cost $\tau_{\rm setup}$ by a factor s'. Combining these factors, the overall computational cost $T_{\rm ILE}/N_{\rm eval}$ of each marginalized likelihood evaluation is

$$T_{\rm ILE}/N_{\rm eval} = \frac{\tau_{\rm start}}{N_{\rm eval}} + [\tau_{\rm setup} + N_{\rm ad}\tau_{\rm ad} + N_{\rm it}\tau_{\rm it}]. \quad (11)$$

The first rows in Table I show our estimated breakdown of these elements of the computational cost, for a typical binary black hole signal with $f_{\rm min} = 20$ Hz and $m_1 \simeq m_2 \simeq 30\ M_\odot$. Because of the extremely high cost of adaptive overhead, we only adapt in two parameters, corresponding to sky location; and we only adapt until a marginalized likelihood evaluation returns significant $\ln \mathcal{L}_{\rm marg}$. As a result, $N_{\rm ad}$ (the number of adaptive stages *per likelihood evaluation*) scales as $N_{\rm it}/n_{\rm chunk}/N_{\rm eval}$ and does not increase with the overall number of samples, so the overall evaluation time scales as

$$T_{\rm ILE,mod}\,/N_{\rm eval} = \frac{\tau_{\rm start} + \tau_{\rm ad}N_{\rm it}/n_{\rm chunk}}{N_{\rm eval}} + [\tau_{\rm setup} + N_{\rm it}\tau_{\rm it}] \quad (12)$$

$$\simeq (\tau_{\rm start}/20\ {\rm s}) + 10(\tau_{\rm ad}/\mu{\rm s}) + 2[(\tau_{\rm it,liike}/\mu{\rm s}) + \tau_{\rm it,rest}/\mu{\rm s}], \quad (13)$$

where in the latter expression we substitute the usual values of $N_{\rm it} \simeq 2\times 10^6$, $n_{\rm chunk} \simeq 10^4$, and $N_{\rm eval} \simeq 20$ conservatively adopted for problems involving many degrees of freedom d (between 6 to 8). We target $N_{\rm eval} \simeq 20$ so that startup and other one-time costs are not a significant contributor to the overall run-time. In this configuration, the run-time per marginalized likelihood is around $T_{\rm ILE}/N_{\rm eval} \simeq 25$ s for a binary black hole. Of this 25 s, only roughly 1.5 s corresponds to evaluating the likelihood, implying the cost $\tau_{\rm it} \simeq 1.5/N_{\rm it} \simeq 7.5\times 10^{-7}$. Likewise, because overhead dominates over array manipulations, an analysis with 16 kHz time series requires only marginally more time than the corresponding 4 kHz analysis. Conversely, the overhead of performing the Monte Carlo integrator (including both $\tau_{\rm ad}$ and $\tau_{\rm it,rest}$) is a quite substantial contribution to the overall run-time for the best-available hardware.

For low-latency analyses, we need to assess the overall wall clock time needed to perform an analysis, often with a restricted number $N_{\rm GPU}$ of available GPU-enabled machines. To complete $N_{\rm net}$ likelihood evaluations with these resources requires

$$\begin{aligned} T_{\rm net,mod} &= \frac{N_{\rm net}}{N_{\rm GPU}N_{\rm eval}} T_{\rm ILE,mod} \\ &= \frac{N_{\rm net}}{N_{\rm GPU}}\left[\frac{\tau_{\rm start} + \tau_{\rm ad}N_{\rm it}/n_{\rm chunk}}{N_{\rm eval}} + [\tau_{\rm setup} + N_{\rm it}\tau_{\rm it}]\right]. \end{aligned} \quad (14)$$

Typically we target $N_{\rm eval} \lesssim 3\times 10^4$, $N_{\rm GPU} \simeq 100$, or roughly 300× the cost per individual marginalized likelihood evaluation. For the binary black hole configuration described above, marginalized likelihood evaluations will complete in about 100 min on a cluster of (b), or more realistically 200 min on a cluster of (a). This number can be reduced to tens of minutes with a modestly larger GPU pool, or by marginally more conservative convergence thresholds $n_{\rm eff}$ or $N_{\rm it}$. This discussion ignores the latency introduced by the gaussian process (GP) interpolation stage at the end. We will revisit accelerated GP interpolation in subsequent work.

The ILE likelihood generates waveform dynamics and $h_{\rm lm}(t)$ once per evaluation $\boldsymbol{\lambda}$. Waveform generation can contribute significantly to the time needed to evaluate $\ln \mathcal{L}_{\rm marg}$ for extremely costly waveform models which require $\tau_{\rm wf}$ a significant fraction of $\tau_{\rm setup}$. That said, the tests described above used relatively costly waveform generation with $h_{\rm lm}(t)$ allocation requiring $\tau_{\rm wf} \simeq 1.2$ s at 4 kHz and 1.9 s at 16 kHz.

The ILE likelihood $\ln \mathcal{L}_{\rm margT}$ involves sums over modes, with the most expensive likelihood and setup operations (involving Q) growing linearly with the number of modes used. This increased cost is most apparent in Table I when comparing the columns corresponding $\tau_{\rm it,like}$ for GPU (b) between the ± 2 and $(\pm 2, \pm 1)$ rows. The cost of evaluating $\ln \mathcal{L}_{\rm margT}$ with four modes is roughly twice that of runs with two modes, as expected.

In a heterogeneous computing environment, code performance varies substantially with available resources, as seen by contrasting corresponding results for (a), (b), and (c) in Table I. For the slowest hardware, the overall run-time will be only a factor of a few smaller than the corresponding CPU-only run-time.

C. Binary neutron stars

Table II shows the corresponding performance breakdown for our fiducial binary neutron star, which has signal SNR $= 32$. In this analysis, we have adopted the TaylorT4 model with $\ell = 2$ and all $m = \pm 2$, ± 1 modes as our

TABLE II. Profiling performance: Binary neutron stars. Evaluation costs for the marginalized likelihood on default hardware, analyzing $T = 8$ s of data with a binary neutron stars $m_1 = 1.4\ M_\odot$, $M_2 = 1.35\ M_\odot$, with convergence threshold $n_{\rm eff} > 50$.

Version	S rate [Hz]	Modes [m]	$\tau_{\rm start}$ [s]	$\tau_{\rm setup}$ [s]	$\tau_{\rm ad}$	$\tau_{\rm it,like}$ [μs]	$\tau_{\rm it,rest}$ [μs]	$\frac{T_{\rm ILE}}{N_{\rm eval}}$ [s]	GPU use [%]
CPU	16384	$\pm 2, \pm 1$	35	26	0.14	680	25	590	
	4096	$\pm 2, \pm 1$	35					25	
GPU (a)	16384	$\pm 2, \pm 1$	35						
	4096	$\pm 2, \pm 1$	35					60–450	
GPU (b)	16384	$\pm 2, \pm 1$	35	26	0.07	2.4		35	16
	4096	$\pm 2, \pm 1$	35	11.5	0.1	2.4		24	
GPU (c)	16384	$\pm 2, \pm 1$	35	71		20	38	105	
	4096	$\pm 2, \pm 1$	35	28		12		60	

fiducial analysis template. This source has a significantly longer duration and a higher amplitude, both of which contribute to slower convergence and longer run-time.

Low-mass compact binaries such as binary neutron stars have much longer inspiral times from the same starting frequency of 20 Hz: roughly 160 s. Nominally, the manipulation of a corresponding power-of-two duration data (256 s) involves 32× more costly Fourier transforms and time integrations than the BBH analysis described above. (An array of 256 s of data at 16 kHz corresponds to 32 Mb.) Comparing $\tau_{\rm setup}$ appearing in Tables I and II, we indeed find our BNS analysis requires a factor of order 32× longer to set up all necessary inner products. Unlike the BBH analysis, the inner product evaluation costs in $\tau_{\rm setup}$ now dominate the overall evaluation time $T_{\rm ILE}/N_{\rm eval}$.

Improved performance arises in part from reusing our adaptive integrator, which (after the first marginalized likelihood evaluation) can exploit tight localization afforded by the many BNS cycles available in band and the high amplitude of our fiducial BNS signal. In some cases, the Monte Carlo integral converges to our target accuracy in of order 10× fewer steps than for BBH. Additionally, particularly for first-stage RIFT grids, the improved performance arises from our choice of initial grid: many points fit poorly and are identified as such well before the fiducial 2×10^6 Monte Carlo iterations. Both sources of improved performance relative to the BBH analysis are independent of the choice of hardware.

For the closed-form post-Newtonian approximation used in the study above, waveform costs are a modest contribution to overall run-time, contributing only 4.1 s to $\tau_{\rm setup}$ at 16 kHz. For other approximations not available in closed form, the waveform generation cost for binary neutron star inspirals can be much more substantial, particularly as $f_{\rm min}$ decreases below the 20 Hz used in this study.

While GPU cards have finite memory, the modest size of our underlying data arrays and intermediate data products on-board ensures that we are unlikely to saturate this bound, even for 16 kHz data, unless we investigate significantly lower starting frequencies or employ vastly more angular modes $h_{\rm lm}$.

IV. PERFORMANCE AND VALIDATION DEMONSTRATIONS WITH FULL PIPELINE

To better illustrate code performance in realistic settings, we also describe end-to-end analyses with the original and modified code. In a full analysis, many of the initially proposed evaluation points $\boldsymbol{\lambda}$ either are not or are marginally consistent with the data. As a result, most marginalized likelihood evaluations in the first iteration proceed extremely rapidly. As the high-cost and low-cost evaluations are generally well mixed between different instances, the overall time to complete the full first iteration is typically much lower than subsequent iterations. With a well-positioned and sufficiently large initial grid, few follow-on iterations are needed.

A. Binary black hole analysis

Modest-amplitude short-duration binary black holes empirically constitute the most frequent detection candidates for current ground-based GW observatories [11]. Because of their brevity and hence broad posterior, the RIFT code's interpolation-based method converges rapidly. Combined with the low cost of each iteration, these sources require an exceptionally low commitment of resources and can be performed in extremely low latency, as desired. To demonstrate this, we use the GPU-accelerated code with $N_{\rm eval} = 20$, analyzing data with $f_{\rm sample} = 4096$ Hz. We use 100 points in the first iteration, in a very coarse mass grid (i.e., spacing comparable to typical astrophysical mass scales), followed by 20 points in each subsequent iteration. Figure 1 shows our results. The final posterior is already well explored with the initial grid points, and converged by the second iteration. Conversely, on average all iterations required roughly 45 s per $\boldsymbol{\lambda}$ to evaluate their grid on GPU (a) hardware. This configuration therefore converged within the 30 min needed for the first two iterations. We can achieve a smaller turnaround time for an otherwise identical analysis by adjusting $N_{\rm eval}$ appropriate to the available hardware. While this low-dimensional problem does not capture all fitting and automation challenges associated with high-dimensional fully precessing binaries,

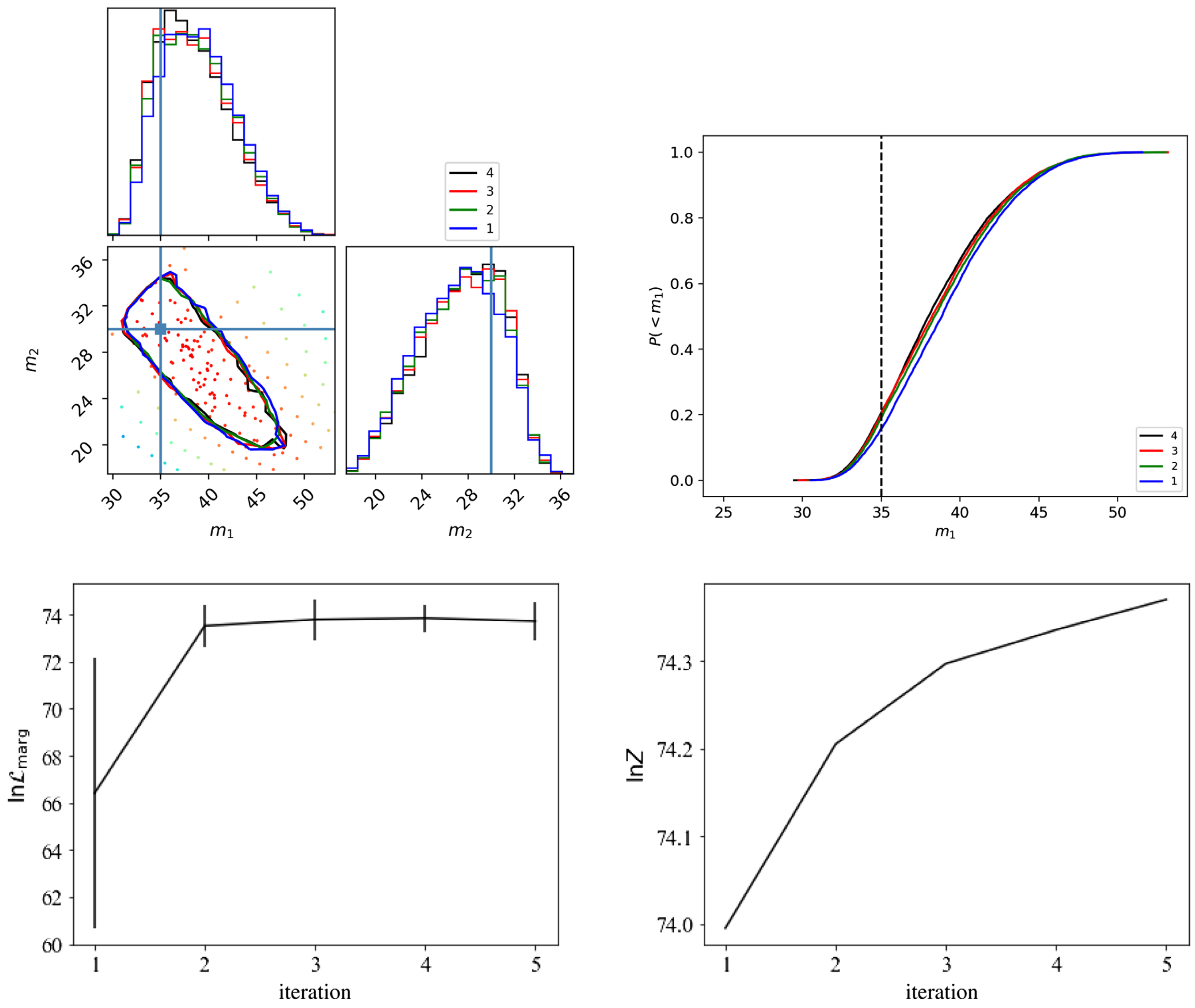

FIG. 1. Convergence of BBH analysis: Zero spin. Results for marginal posterior distributions of our fiducial synthetic binary black hole. Solid contours show credible intervals; solid one-dimensional distributions show marginal cumulative distribution functions (CDFs) and probability density functions (PDFs) for the corresponding variables; and colored points indicate the location $\boldsymbol{\lambda}$ and value of the underlying marginalized likelihood evaluations. Left panel: Posterior distribution over $\mathcal{M}$ and $\delta = (m_1 - m_2)/M$. Right panel: Marginal 1d CDFs of $\mathcal{M}$, showing convergence. Bottom left panel: Mean and variance of the array $\ln \mathcal{L}_{\rm marg}(\boldsymbol{\lambda}_j)$ for $j = 1, 2, \ldots, N_{\rm eval}$ indexing all candidate sets of intrinsic parameters $\boldsymbol{\lambda}_j$ performed in that iteration, showing that after the first iteration the candidate points are consistent with the posterior (i.e., no proposed point has very low $\ln \mathcal{L}_{\rm marg}$). Bottom right panel: The estimated evidence $Z = \int d\lambda \mathcal{L}_{\rm marg}$ vs the iteration number. As a systematic fitting error dominates our error budget, the Monte Carlo error is not shown.

it does capture the low cost and rapid response possible with RIFT.

B. Binary neutron star analysis: Assuming zero spin

Significant-amplitude binary neutron star mergers are the most important scenario for rapid parameter inference, as low latency can enable multimessenger follow-up [11]. In our second test we compare two workflows, one with the original embarrassingly parallel RIFT code using $N_{\rm eval} = 1$ and one with the "batched" GPU-accelerated code with $N_{\rm eval} = 20$, analyzing data with $f_{\rm sample} = 4096$ Hz. In this example, we use 100 initial points spread over the two mass dimensions $\mathcal{M}_z$, δ, adding 20 evaluations per iteration. We choose this simple low-dimension, small-size, and slowly converging configuration to facilitate visualization of the grid, posterior, and convergence. The initial coarse grid covers a region $\mathcal{M}_z \in [1.1962, 1.1970]\ M_\odot$ and $\delta \in [0, 0.25]$, ensuring the posterior was smaller than our initial coarse grid spacing. The top panels of Fig. 2 show posterior distributions derived from the first several RIFT iterations, while the bottom panels show convergence diagnostics. Because our analysis uses a post-Newtonian

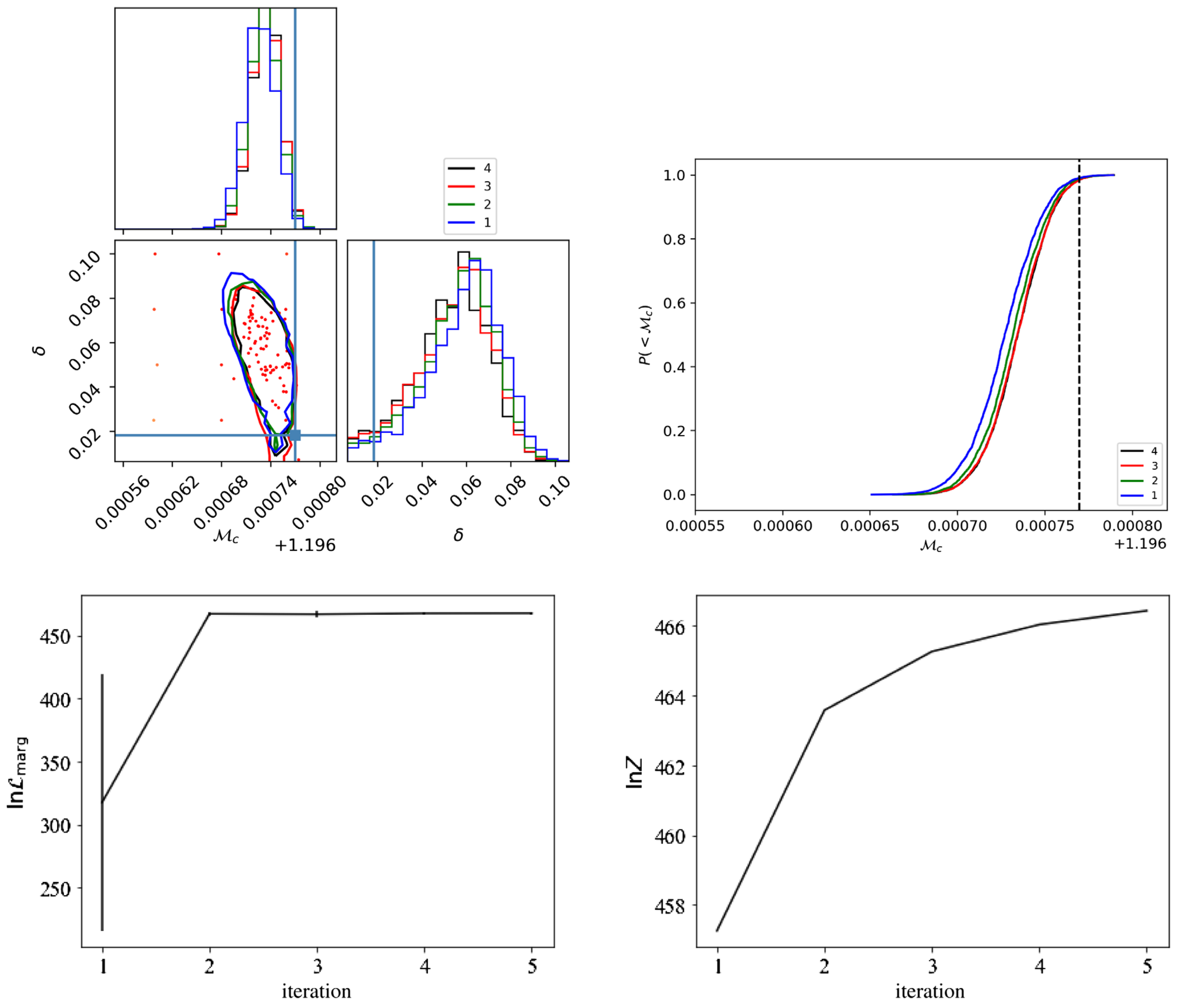

FIG. 2. Convergence of BNS analysis: Zero spin. Results for marginal posterior distributions of our fiducial synthetic neutron star. Solid contours show credible intervals; solid one-dimensional distributions show marginal CDFs and PDFs for the corresponding variables; and colored points indicate the location $\boldsymbol{\lambda}$ and value of the underlying marginalized likelihood evaluations. Left panel: Posterior distribution over $\mathcal{M}$ and $\delta = (m_1 - m_2)/M$. Right panel: Marginal 1d CDFs of $\mathcal{M}$, showing convergence. Bottom left panel: Mean and variance of $\ln \mathcal{L}_{\rm marg}$ on the evaluation points, showing that after the first iteration the candidate points are consistent with the posterior (i.e., no proposed point has very low $\ln \mathcal{L}_{\rm marg}$). Bottom right panel: $Z = \int d\lambda \mathcal{L}_{\rm marg}$ vs the iteration number. As a systematic fitting error dominates our error budget early on, the Monte Carlo error is not shown.

model to interpret a signal generated with SEOBNRv4, the peak posterior density is slightly offset from the synthetic signal's parameters

This specific workflow configuration reduces overall core usage by maximizing GPU use per iteration: after the first iteration, only one GPU is active. Specifically, with the updated RIFT workflow used here, with one instance analyzing each 20 evaluations, we use five core + GPU pairs in the first iteration, followed by one core + GPU for remaining iterations. By contrast, with the original CPU-only RIFT code analyzing each $\boldsymbol{\lambda}$ in parallel, this process requires roughly 20 core-minutes per $\boldsymbol{\lambda}$, using 100 cores in the first iteration and 20 cores in each subsequent evaluation. Note that because $N_{\rm eval} = 20$ for the GPU is larger than the (hardware and $f_{\rm sample}$ dependent) speed-up factor between the CPU and GPU implementation, the overall wall clock time needed for an end-to-end analysis is larger for the GPU workflow. We can achieve a comparable or smaller turnaround time for an otherwise identical analysis by adjusting $N_{\rm eval}$ appropriate to the available hardware.

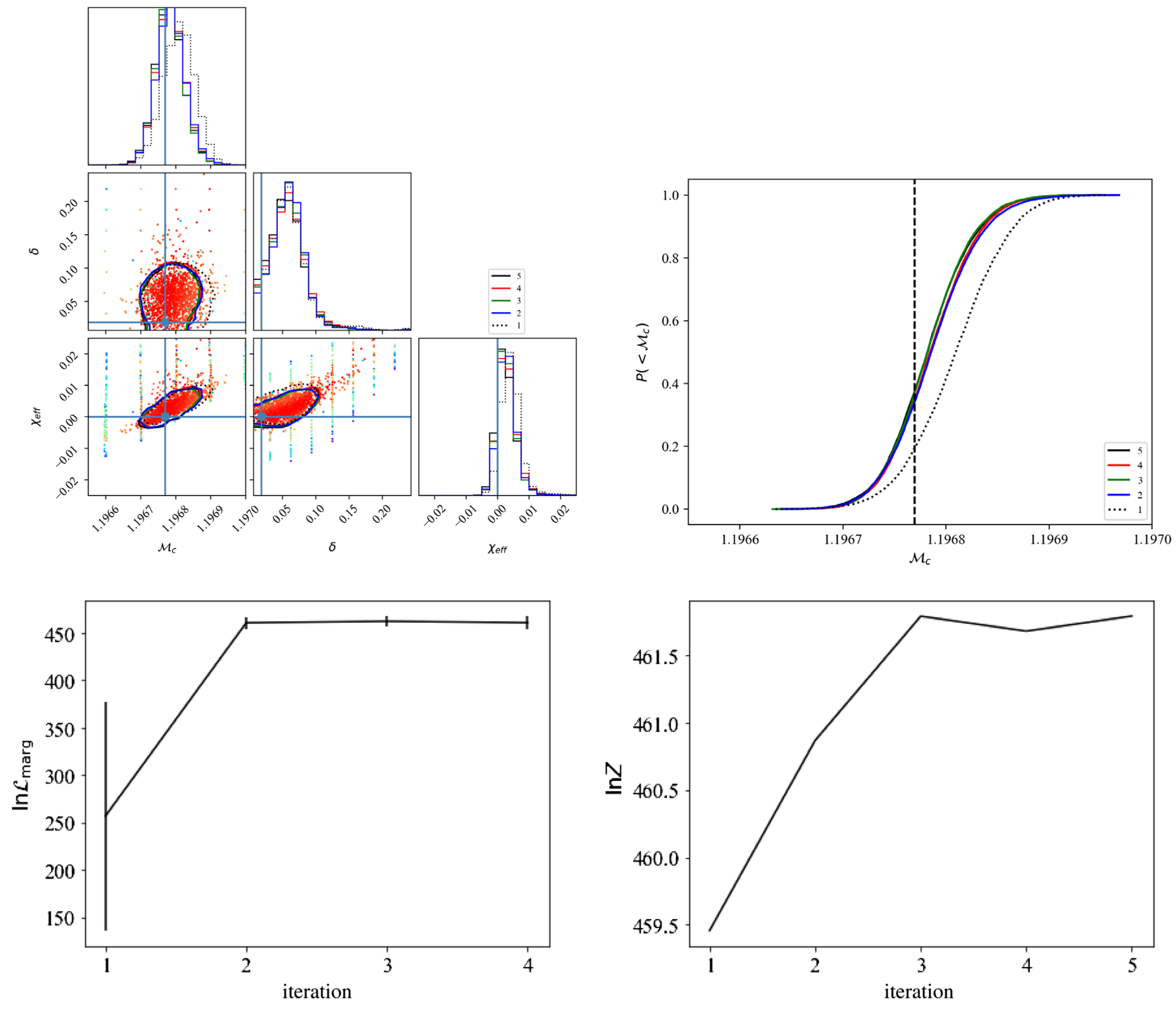

FIG. 3. Demonstration of low-latency spinning analysis: Results for marginal posterior distributions of our fiducial synthetic neutron star, assuming the binary has nonprecessing spins. Solid contours show credible intervals; solid one-dimensional distributions show marginal CDFs and PDFs for the corresponding variables; and colored points indicate the location λ and value of the underlying marginalized likelihood evaluations. Only evaluations with marginalized log-likelihoods within 30 of the maximum are shown, to increase contrast. The solid curves show an analysis with SEOBNRv4_ROM. Left panel: Posterior distribution over $\mathcal{M}$, $\delta = (m_1 - m_2)/M$, and χ_{eff}. Right panel: Marginal 1d CDFs of δ, showing convergence. Bottom left panel: Mean and variance of $\ln \mathcal{L}_{\mathrm{marg}}$ on the evaluation points, showing that after the first iteration the candidate points are consistent with the posterior (i.e., no proposed point has very low $\ln \mathcal{L}_{\mathrm{marg}}$). Bottom right panel: $Z = \int d\lambda \mathcal{L}_{\mathrm{marg}}$ vs the iteration number. As a systematic fitting error dominates our error budget early on, the Monte Carlo error is not shown.

C. Binary neutron star analysis: Assuming nonprecessing spin

Binary neutron star models with more parameters such as spin and tides require correspondingly larger numbers of points in the initial grid and per iteration. Fortunately, the number of observationally significant and accessible dimensions is often substantially less than the prior dimensionality. In practice,we use roughly 20× more points per iteration for precessing massive BBH systems ($d = 8$) or for spinning binary neutron stars with tides ($d = 6$). As a result, we can still achieve relatively rapid turnaround on a high-dimensional binary neutron star analysis even in resource-constrained environments.

As a concrete demonstration of a realistic modest-latency analysis using the GPU-accelerated code, we reanalyze our fiducial binary neutron star signal, accounting for the possibility of nonzero (aligned) neutron star spins. Our initial grid consists of 5000 points, spread approximately uniformly across a four-dimensional hypercube $\mathcal{M}_z \in [1.1962, 1.1970]\ M_\odot$, $\delta \in [0, 0.25]$, and $\chi_{i,z} \in [-0.05, 0.05]$. Subsequent iterations use 500 random draws from the estimated posterior samples produced from the previous iteration. Because our fiducial post-Newtonian model (TaylorT4) as implemented in LALSUITE [34] does not include spin, we employ both the SEOBNRv4_ROM and SpinTaylorT4 waveform approximations for nonprecessing binaries, omitting higher-order modes. For SEOBNRv4_ROM, we use 48 GPU (c) enabled nodes; for SpinTaylorT4, we use 100 GPU (a) nodes. Figure 3 shows our results.

Due to our self-imposed resource constraints on $N_{\rm eval} = 20$ and the number of GPUs, the first iteration is resource-limited and requires of order $(5000/20/100) \simeq 2.5$ to $(5000/20/48) \simeq 5$ times as long as the first iteration of the zero-spin BNS analysis described above. Each marginalized likelihood evaluation in the first iteration (as well as subsequent iterations) requires roughly 60 s for both SpinTaylorT4 and SEOBNRv4ROM, on average. Subsequent marginalized likelihood iterations are not resource limited and complete in roughly 35 min, depending on hardware. As previously, we can achieve a comparable or smaller turnaround time for an otherwise identical analysis by adjusting $N_{\rm eval}$ appropriate to the available hardware.

V. CONCLUSIONS

We have demonstrated that the marginalized likelihood $\ln \mathcal{L}_{\rm marg}(\boldsymbol{\lambda})$ appearing in the RIFT/rapidPE parameter inference calculation can be evaluated at fixed $\boldsymbol{\lambda}$ in tens of seconds on average for both binary black holes and binary neutron stars. This performance improvement could enable very low latency source parameter inference for compact binaries, which can be of use for targeting multimessenger follow-up observations via sky localization and precise source characterization. This prospect is particularly interesting because RIFT can often achieve this performance using computationally costly models for binary merger with rich physics such as higher modes or eccentricity, as the waveform generation cost does not usually limit code performance.

In addition to producing results rapidly, RIFT results can be produced with a noticeably smaller overall resource footprint than loosely similar lalinference (LI) analyses, even without tuning to optimize RIFT pipeline settings. As a concrete and nonoptimized example, for all five of the iterations of the spinning binary neutron star parameter inference with SEOBNRv4_ROM from 20 Hz described in this work, our RIFT analysis expended roughly 14 core-days. By contrast, a TaylorF2 analysis of GW170817 with LI in Markov-chain Monte Carlo mode starting from 23 Hz required 228 core-days. For precessing binary black holes using SEOBNRv3, the improvement is equally substantial: 10 core-days for a 10-iteration investigation of a synthetic GW150914-like source with RIFT, vs 291 core-days for a LI analysis of GW170729. We defer detailed relative benchmarking using comparably converged parameter inference to future work.

The overall code performance and thus latency can be further decreased substantially, notably by converting the Monte Carlo random number generation and inner products to GPU-based operations. In such a configuration, the marginalized likelihood code would perform almost all calculations (except waveform generation) on a GPU, with minimal communication off the board. This configuration should further reduce the average time needed to compute $\ln \mathcal{L}_{\rm marg}(\boldsymbol{\lambda})$ for both binary black holes (which are Monte Carlo limited) and binary neutron stars (which are inner-product limited). We anticipate a further factor of roughly 10 reduction in overall evaluation time can be achieved soon. At that level of performance, a single 8-GPU machine with contemporary hardware could perform parameter inference less than 10 min. Since binary compact objects intersect our past light cone only once every roughly 15 min, accounting for all past history, such a configuration would be able to address low-latency parameter inference for the duration of second-generation ground-based observing. Alternatively, if larger resource pools are available in low latency, both the original and the now GPU-accelerated RIFT can perform extremely rapid parameter inference if large iterations are performed completely in parallel (i.e., $N_{\rm eval} \simeq 1$).

By allowing models with higher-order modes to be used in low latency, our code can exploit the tighter constraints which higher modes can enable on the properties of low-mass binaries with suitable amplitudes and orientations. These tighter constraints could better inform low-latency source classification and hence multimessenger follow-up observations of compact binary mergers.

Beyond low-latency multimessenger astronomy, rapid parameter inference enables new applications. For example, every parameter inference provides *evidence* for a signal being present in the data; with rapid parameter inference, this evidence could be used as (the last stage in a hierarchical pipeline for) a detection statistic [45]. This approach can identify individual events and even a population. Alternatively and in many ways equivalently, one can identify a population of GW sources without assuming any one is real, by applying parameter inference to more candidate events and self-consistently separating foreground and background [46,47].

When suitable surrogate models are available, the overall code performance and thus latency could yet again be further reduced by eliminating the iteration and fitting stages entirely, performing one Monte Carlo integration at once [48]. This approach exploits a linear representation of $h_{\rm lm}(t)$ via basis functions, to enable rapid likelihood evaluation as a function of both extrinsic parameters and $\boldsymbol{\lambda}$. Though not necessarily or compactly available for all surrogate models, particularly for the small basis sizes necessary to fit onboard GPUs, this approach could enable exceedingly low latency at a small computational cost. This alternative architecture would be exceptionally well suited to the alternative applications of low-latency PE described above.

The use of CUPY enables our code to be highly portable across architectures and heterogeneous GPU environments, while transitioning smoothly between GPU and CPU modes. The techniques we used here will be transported to other Bayesian inference modeling codes used to interpret GW observations [49,50].

In this work, we have focused exclusively on profiling a simplified pipeline to produce posterior distributions for detector-frame intrinsic parameters. The code also produces reliable extrinsic parameter distributions [9]. With some postprocessing, this pipeline provides joint posterior distributions for all intrinsic and extrinsic parameter distributions together; examples of these distributions have been published elsewhere [11]. Presently, rather than harvest extrinsic information from every iteration, we harvest joint intrinsic and extrinsic information with a single final iteration, which we will implement shortly in our production pipeline.

ACKNOWLEDGMENTS

We thank our anonymous referee for helpful feedback. D. W. thanks the RIT COS and CCRG for support, and NSF-1707965. J. L. is supported by NSF PHY-1707965. R.O'S. is supported by NSF PHY-1707965 and PHY-1607520. Y.-L. L. F. was supported in part by BNL LDRD Projects No. 17-029 and No. 19-002, and New York State Urban Development Corporation, d/b/a Empire State Development, under Contract No. AA289. We thank the Computational Science Initiative at BNL for hosting a GPU Hackathon, during which a part of this work was conducted. We also thank Jeff Layton at NVIDIA for bringing CuPy to our attention.

APPENDIX: PROPERTIES OF RESOURCES USED

LIGO-CIT worker nodes, denoted by (a) in the text, are principally S6 nodes with 2-CPU ×8 core Opteron 2.3 GHz machines with 16 Gb of RAM, with GTX 1050 Ti cards with 4 Gb of RAM. For LIGO-CIT, profiling reports reflect performance averaged over the whole cluster and hence lacks the detailed reporting produced for the other configurations. LIGO-LHO worker nodes with GPUs, denoted by (c) in the text, are a heterogeneous configuration mostly consisting of (a). Unlike profiling at LIGO-CIT, the profiling for LIGO-LHO reflects controlled tests on a single node. The V100 machine (ldas-pcdev13, denoted by (b) in the text) is a 24-core ES-2650 v4 machine with 4 GPUs, only one of which is active in our tests: Tesla V100 with 16 Gb of RAM. All non-GPU profiling was also performed on this machine.

[1] B. Abbott *et al.* (LIGO Scientific Collaboration), Classical Quantum Gravity **32**, 074001 (2015).
[2] T. Accadia *et al.*, J. Instrum. **7**, P03012 (2012).
[3] LIGO Scientific and Virgo Collaborations, Phys. Rev. Lett. **116**, 061102 (2016).
[4] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. Lett. **118**, 221101 (2017).
[5] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. X **6**, 041015 (2016).
[6] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. Lett. **119**, 141101 (2017).
[7] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Astrophys. J. **851**, L35 (2017).
[8] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. Lett. **119**, 161101 (2017).
[9] C. Pankow, P. Brady, E. Ochsner, and R. O'Shaughnessy, Phys. Rev. D **92**, 023002 (2015).
[10] J. Lange, R. O'Shaughnessy, and M. Rizzo, arXiv:1805.10457.
[11] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), https://dcc.ligo.org/LIGO-P1800307 (2018).
[12] J. Veitch *et al.*, Phys. Rev. D **91**, 042003 (2015).
[13] B. P. Abbott *et al.*, Living Rev. Relativity **19**, 1 (2016).
[14] LIGO Scientific Collaboration, https://dcc.ligo.org/LIGO-T1600115.
[15] LIGO Scientific Collaboration, https://dcc.ligo.org/LIGO-M1800085/public.
[16] R. Smith, S. E. Field, K. Blackburn, C.-J. Haster, M. Pürrer, V. Raymond, and P. Schmidt, Phys. Rev. D **94**, 044031 (2016).
[17] B. Miller, R. O'Shaughnessy, B. Farr, and T. Littenberg, Phys. Rev. D **92**, 044056 (2015).
[18] P. Canizares, S. E. Field, J. Gair, V. Raymond, R. Smith, and M. Tiglio, Phys. Rev. Lett. **114**, 071104 (2015).
[19] S. Vinciguerra, J. Veitch, and I. Mandel, Classical Quantum Gravity **34**, 115006 (2017).
[20] B. Zackay, L. Dai, and T. Venumadhav, arXiv:1806.08792.
[21] S. E. Field, C. R. Galley, J. S. Hesthaven, J. Kaye, and M. Tiglio, Phys. Rev. X **4**, 031006 (2014).
[22] M. Hannam, P. Schmidt, A. Bohé, L. Haegel, S. Husa, F. Ohme, G. Pratten, and M. Pürrer, Phys. Rev. Lett. **113**, 151101 (2014).
[23] R. J. E. Smith, K. Cannon, C. Hanna, D. Keppel, and I. Mandel, Phys. Rev. D **87**, 122002 (2013).
[24] K. Cannon, J. D. Emberson, C. Hanna, D. Keppel, and H. P. Pfeiffer, Phys. Rev. D **87**, 044008 (2013).

[25] A. Lundgren and R. O'Shaughnessy, Phys. Rev. D **89**, 044021 (2014).

[26] P. Canizares, S. E. Field, J. R. Gair, and M. Tiglio, Phys. Rev. D **87**, 124005 (2013).

[27] M. Pürrer, Classical Quantum Gravity **31**, 195010 (2014).

[28] P. Graff, F. Feroz, M. P. Hobson, and A. Lasenby, Mon. Not. R. Astron. Soc. **421**, 169 (2012).

[29] R. J. E. Smith, C. Hanna, I. Mandel, and A. Vecchio, Phys. Rev. D **90**, 044074 (2014).

[30] J. Blackman, S. E. Field, C. R. Galley, B. Szilágyi, M. A. Scheel, M. Tiglio, and D. A. Hemberger, Phys. Rev. Lett. **115**, 121102 (2015).

[31] R. H. Cole and J. R. Gair, Phys. Rev. D **90**, 124043 (2014).

[32] H. Antil, S. E. Field, F. Herrmann, R. H. Nochetto, and M. Tiglio, J. Sci. Comput. **57**, 604 (2013).

[33] N. J. Cornish, arXiv:1007.4820.

[34] LIGO Scientific Collaboration, LIGO Algorithm Library-LALSuite, Free Software (GPL) (2018), https://lscsoft.docs.ligo.org/lalsuite/lalsimulation/index.html.

[35] R. O'Shaughnessy, J. Blackman, and S. E. Field, Classical Quantum Gravity **34**, 144002 (2017).

[36] B. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. D **94**, 064035 (2016).

[37] B. Miller, thesis, Northwestern University, 2016.

[38] B. Miller and C. Pankow (private communication).

[39] S. A. Usman *et al.*, Classical Quantum Gravity **33**, 215004 (2016).

[40] C. M. Biwer, C. D. Capano, S. De, M. Cabero, D. A. Brown, A. H. Nitz, and V. Raymond, Publ. Astron. Soc. Pac. **131**, 024503 (2019).

[41] J. Lange *et al.*, Phys. Rev. D **96**, 104041 (2017).

[42] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017, http://learningsys.org/nips17/assets/papers/paper_16.pdf.

[43] A. Bohé *et al.*, Phys. Rev. D **95**, 044028 (2017).

[44] A. Buonanno, B. R. Iyer, E. Ochsner, Y. Pan, and B. S. Sathyaprakash, Phys. Rev. D **80**, 084043 (2009).

[45] R. Smith and E. Thrane, Phys. Rev. X **8**, 021019 (2018).

[46] W. M. Farr, J. R. Gair, I. Mandel, and C. Cutler, Phys. Rev. D **91**, 023005 (2015).

[47] S. M. Gaebel, J. Veitch, T. Dent, and W. M. Farr, Mon. Not. R. Astron. Soc. **484**, 4008 (2019).

[48] R. O'Shaughnessy, J. Blackman, and S. Field, Classical Quantum Gravity **34**, 144002 (2017).

[49] D. Wysocki, J. Lange, and R. O'Shaughnessy, arXiv:1805.06442.

[50] D. Wysocki and R. O'Shaughnessy, https://ccrg.rit.edu/content/software/pop-models.