Scalable dark siren cosmology with gwcosmo: GPU acceleration, validation, and systematics

Alexander Papadopoulos,[1,*] Christian E. A. Chapman-Bird,[2,1] Rachel Gray,[1] Christopher Messenger,[1] and Tom Bertheas[3,4]

[1]*Institute for Gravitational Research, University of Glasgow, Glasgow, G12 8QQ, United Kingdom*
[2]*Institute for Gravitational Wave Astronomy and School of Physics and Astronomy, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom*
[3]*Laboratoire des 2 Infinis - Toulouse (L2IT-IN2P3), Université de Toulouse, CNRS, F-31062 Toulouse Cedex 9, France*
[4]*Laboratoire de Physique de l'Ecole Normale Supérieure (LPENS), ENS, Université PSL, CNRS, 75005 Paris, France*

(Received 5 June 2026; accepted 25 August 2026; published 15 September 2026)

As the number of confident gravitational-wave detections grows, population-level hierarchical analyses face increasing computational costs. Dark-siren cosmological inference integrates over the localisation volume of each gravitational-wave source. To remain feasible without discarding information from the quieter but more numerous sources in the catalog, significant efficiency improvements are vital for analysis pipelines. In this work, we present an upgraded version of the cosmological inference pipeline `gwcosmo`, which leverages vectorisation on graphics processing units to process the entire gravitational-wave catalog in parallel with each iteration. This new implementation achieves a speed-up of $\sim 10^3$ over the previous version, facilitating analyses of O5-like numbers of GW events on wall-clock timescales of hours. Our results demonstrate the scalability of the `gwcosmo` pipeline, specifically its ability to handle the increasing computational load of expanding event catalogs, positioning it as a vital tool for future advances in dark-siren cosmology.

DOI: 10.1103/53c5-mtk5

I. INTRODUCTION

Inference of cosmological parameters from compact binary coalescences (CBCs) was first described by Schutz in 1986 [1], introducing the usage of gravitational wave (GW) signals as "standard sirens." Unlike electromagnetic distance ladders, standard sirens offer an independent probe of cosmology, making it possible to constrain parameters such as the Hubble constant, H_0, directly from GW signals. Cosmological inference with GW signals requires information on the redshift of the source (z), as well as the luminosity distance (d_L) [1–3]. GW observations directly measure the redshifted chirp mass $\mathcal{M}_z = \mathcal{M}(1+z)$, where $\mathcal{M}$ is the source-frame chirp mass, and the luminosity distance of the source, which depends on redshift through the cosmological model

$$d_L(z) = (1+z)\int_0^z \frac{c\,dz'}{H(z')}. \tag{1}$$

Here $H(z) = H_0\sqrt{\Omega_m(1+z)^3+\Omega_\Lambda}$, with $\Omega_{m,\Lambda}$ being the matter and dark energy density parameters assuming a ΛCDM(ColdDarkMatter) cosmological model. Because GW signals encode information on the redshifted rather than source-frame masses, the source redshift cannot be determined directly from GW data alone,[1] and additional information is required to break this degeneracy.

In this framework, GW signals as standard sirens can be split into a number of classes. In the presence of an electromagnetic counterpart, the redshift can be determined by the localization of the counterpart inside a host galaxy—earning the name *bright sirens* [2,5–8]. When no unique counterpart information is available, it is possible to break the degeneracy by either using information solely from the mass distribution of GW sources, or combining this with galaxy surveys to statistically determine the redshift of the sources—*spectral* [9–16] and *dark sirens*, respectively [1,17–24].

*Contact author: a.papadopoulos.1@research.gla.ac.uk

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

[1]In the case of binary neutron star mergers, it is possible to break this degeneracy using the tidal information of the signal [4].

Since the first detection in 2015, more than 200 GW events have been detected by the LIGO-Virgo-KAGRA Collaboration (LVK) [25–27] up to the end of the first part of the fourth observing run (O4a), collected in the fourth Gravitational-Wave Transient Catalog (GWTC-4.0) [28–31]. It is anticipated that this number will continue to rise in the near future, with both O4b and O4c data releases still to come, as well as future observing runs. This rapidly growing number of detections will be crucial for tightening cosmological constraints with GWs, since the precision of these measurements increases with the number of observed sources. Dark and spectral sirens will play an important role in constraining cosmological parameters, as to date there has been only one instance of a confidently associated bright siren, GW170817 [6].

The cosmological inference pipeline `gwcosmo` [22,32,33] implements the *dark siren* method via a hierarchical Bayesian framework to jointly infer cosmological and population properties of compact binary coalescences, additionally incorporating information from galaxy surveys. Other pipelines such as `icarogw` [34,35] and `CHIMERA` [24,36] employ similar hierarchical methods to perform cosmological inference with GWs. Previous versions of `gwcosmo` have been used by the LVK collaboration to analyze GW events and produce cosmological constraints, most recently in [37]. During cosmological inference, this version of `gwcosmo` operated on the GW catalog data in a highly serialized fashion, processing the pixelated sky-localization area of each GW event one at a time, working pixel-by-pixel to construct the hierarchical likelihood. This means that the wall-time for analysis scales approximately as $\mathcal{O}(N_{\rm events} \times N_{\rm pix} \times N_{\rm samples})$, denoting the number of events, pixels and samples in each pixel, respectively. Indeed, in [37], analyses using 141 GW events took $\mathcal{O}$(weeks) to reach convergence even when parallelized over 32 central processing units (CPUs). This presents a real challenge to the scalability of cosmological analyses with a growing catalog of GW events, which threatens to become intractable in the current framework.

In this work, we present an updated version of `gwcosmo`, which is capable of scalable cosmological inference using a graphics processing unit (GPU), as well as retaining the original ability to parallelize over multiple CPUs. This update allows us to perform fully vectorized evaluations of `gwcosmo`'s hierarchical likelihood, significantly decreasing the wall time taken to produce constraints on cosmological and GW population parameters. Even when using $\mathcal{O}(10^3)$ GW events, comparable to the numbers expected during the O5 observing run, the updated code is capable of outperforming its predecessor by a factor of 1000. This enables a wide range of new studies in GW cosmology in a fraction of the time, while producing results fully consistent with previous versions of the code and supporting future analyses with substantially expanded GW event catalogs through additional performance optimizations.

This paper is organized as follows. In Sec. II we describe the methods used which allow the likelihood to be evaluated on a GPU, as well as new features added to the pipeline. In Sec. III we demonstrate the consistency of the code with the previous version, and the ability to recover parameters from a simulated population of GW events. Additionally in this Section we investigate the effects of a number of implementation choices on the final cosmological constraints. We also include a number of Appendixes describing the GW data used in our analyses (Appendix A), validating our criterion for posterior agreement (Appendix B), comparing code performance on different devices and energy consumption (Appendix C).

II. METHODOLOGY

A. Hierarchical likelihood

For a population of GW sources described by hyperparameters $\Lambda = \Lambda_{\rm cosmo}, \Lambda_{\rm pop}$, where $\Lambda_{\rm cosmo}(\Lambda_{\rm pop})$ describe the cosmological (population) model, the posterior on the hyperparameters given the GW data $\{x_{\rm GW}\}$ from the $N_{\rm det}$ detected events, conditioned on the detection criterion $\{D_{\rm GW}\}$ is given by [38,39],

$$p(\Lambda|\{x_{\rm GW}\},\{D_{\rm GW}\},I) \propto p(\Lambda|I)p(N_{\rm det}|\Lambda,I) \times \prod_i^{N_{\rm det}} p(x_{{\rm GW}i}|D_{{\rm GW}i},\Lambda,I), \quad (2)$$

where the term $p(x_{{\rm GW}i}|D_{{\rm GW}i},\Lambda,I)$ is the likelihood from GW event i given the binary detection parameter $D_{{\rm GW}i} = 1$. The inclusion of I is a catch-all for the remaining information, cosmological or otherwise, not expressed explicitly. Using Bayes's theorem and by explicit marginalization over the source frame parameters of individual GW events θ (masses, spins, redshift, inclination etc.), this can be expressed as,

$$p(\Lambda|\{x_{\rm GW}\},\{D_{\rm GW}\},I) \propto p(\Lambda|I)p(N_{\rm det}|\Lambda,I) \times \alpha(\Lambda)^{-N_{\rm det}} \prod_i^{N_{\rm det}} \int p(x_{{\rm GW}i}|\theta,\Lambda,I)p(\theta|\Lambda,I)d\theta. \quad (3)$$

The term $\alpha(\Lambda)$ is known as the *selection function*, and is defined as

$$\alpha(\Lambda) = \int p(D_{\rm GW}|\theta,\Lambda,I)p(\theta|\Lambda,I)d\theta, \quad (4)$$

and is used to account for the finite detection horizon for GW events in the observed population, meaning that they represent an incomplete set of the true astrophysical population. As this term is identical for all events in the population, it can be removed from the product. While seemingly dependent on Λ, by using a scale-free Jeffrey's

prior on the expected rate of detected GW events, the cosmological dependence in $p(N_{\rm det}|\Lambda)$ drops out [40].

In order to explicitly incorporate galaxy catalog information, Eq. (3) can be rewritten to marginalize over GW event sky location (Ω) by dividing each event into $N_{\rm pix}$ equally sized pixels, and then marginalizing over z, as well as explicitly including the selection function in the denominator,

$$p(\Lambda|\{x_{\rm GW}\},\{D_{\rm GW}\},I) \propto p(\Lambda|I)p(N_{\rm det}|\Lambda,I)\left[\iint p(D_{\rm GW}|z,\theta',\Lambda,I)p(\theta'|\Lambda,I)\sum_j^{N_{\rm pix}} p(z|\Omega_j,\Lambda,I)d\theta' dz\right]^{-N_{\rm det}} \times \prod_i^{N_{\rm det}}\left[\iint \sum_j^{N_{\rm pix}} p(x_{{\rm GW}i}|\Omega_j,z,\theta',\Lambda,I)p(\theta'|\Lambda,I)p(z|\Omega_j,\Lambda,I)d\theta' dz\right], \quad (5)$$

where $\theta = \{z,\theta'\}$. We approximate in Eq. (5) that the probability of detection does not vary over the sky, and so the $p(D_{\rm GW}|z,\theta',\Lambda,I)$ term is removed from the sum over pixels.[2] Of particular interest is the likelihood term $p(x_{{\rm GW}i}|\Omega_j,z,\theta',\Lambda,I)$. This can be written in terms of samples to explicitly show the usage of a kernel density estimate (KDE) to construct a distribution of GW probability in each pixel,

$$p(x_{{\rm GW}i}|\Omega_j,z,\theta',\Lambda,I) \propto \sum_{k=1}^{N_{{\rm samps},j}} w_{ijk}K_{h_{ij}}(z-z_{ijk}). \quad (6)$$

Where $K_{h_{ij}}(\cdot)$ is a Gaussian kernel and bandwidth h_{ij}, and the weights of sample k in pixel j are defined as

$$w_{ijk} = \frac{p(z_{ijk},\theta'_{ijk}|\Lambda,I)}{\pi_{\rm PE}(z_{ijk},\theta'_{ijk})}, \quad (7)$$

which removes assumed parameter estimation priors, and reweighting by the cosmological and population models being analyzed. Lastly, we also include a numerical stability criterion on a pixel-by-pixel basis called the *effective number of samples*, $N_{\rm eff,PE}$, which for pixel j in event i using samples k is defined as [41,42]

$$N_{{\rm eff,PE},j} = \frac{(\sum_k w_{ijk})^2}{\sum_k w_{ijk}^2}, \quad (8)$$

using the weights calculated in Eq. (7). If a pixel does not pass this threshold for a given set of Λ, it does not contribute to the total likelihood. For further discussion of the statistical framework underpinning `gwcosmo`, see [33].

[2]This is because the probability of detection is averaged over an observing run, and the rotation of the earth blurs much of the dependence on sky position. There remains a mild dependence on declination which is neglected in this framework, consistently with the previous version of `gwcosmo` [33].

B. GPU acceleration

The primary computational cost when evaluating the `gwcosmo`'s likelihood is constructing the KDEs in Eq. (6). These are evaluated over the redshift samples—which are dependent on Λ via $\Lambda_{\rm cosmo}$—for all sky pixels in each GW event. For each likelihood evaluation Λ are updated, and so the KDEs must be recomputed on the fly for each event. Given the lack of an alternative method to compute these in parallel, this operation is sequential, meaning the cost increases linearly with the number of events in the analysis. This process, which was implemented in previous iterations of `gwcosmo`, is visualized in Fig. 1. In each likelihood call, the code must move through a set of nested loops, with the top level being a loop over $N_{\rm events}$, then through each of the $N_{\rm pixels}$ making up the GW event sky area, and finally through the $N_{\rm samples}$ in each pixel when constructing the KDE. These pixels are then integrated over redshift, before a summation on the event level, to produce a single hierarchical likelihood value. The cost of this process grows linearly as the number of GW events being analyzed increases, to the point that analyses become intractable.

We present a new approach to evaluate the `gwcosmo` likelihood, which leverages the massively parallel processing capabilities of GPUs to significantly improve the performance of cosmological analyses. The major structural change required to implement this is the construction of a three-dimensional tensor that contains event posterior samples, of shape $(N_{\rm events}, N_{\rm pix}, N_{\rm samples})$. Each tensor element corresponds to an unweighted posterior sample associated with a given sky pixel for a given event. To avoid the complexity of ragged arrays required by events with different numbers of pixels and samples, we pad our tensors based on the largest member's dimensions (max sky pixels and max samples). This ensures a dense, rectangular structure where padded indices do not contribute anything to the total likelihood. As visualized in Fig. 2, this allows the likelihood evaluation to be fully vectorized, with every operation contained in both numerator and denominator of Eq. (5) being computed in parallel over events, pixels and samples. This means that all explicit loops over events, pixels and samples are removed from the code, delivering a

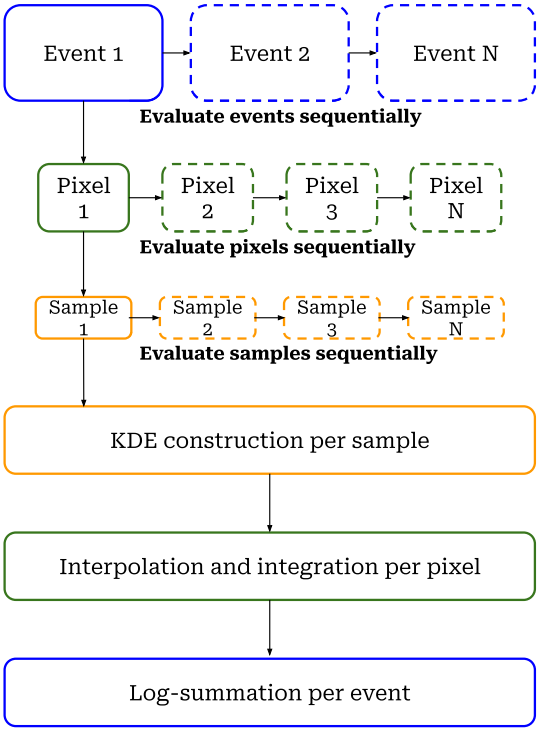


FIG. 1. Sequential CPU based likelihood evaluation scheme. The process uses serial evaluation over events, pixels and samples, whereby the KDE construction iterates over each sample, then pixel-level distributions are interpolated and integrated, and the single-event likelihoods are log summed. Dashed lines indicate elements which must wait to be evaluated in the sequential scheme.

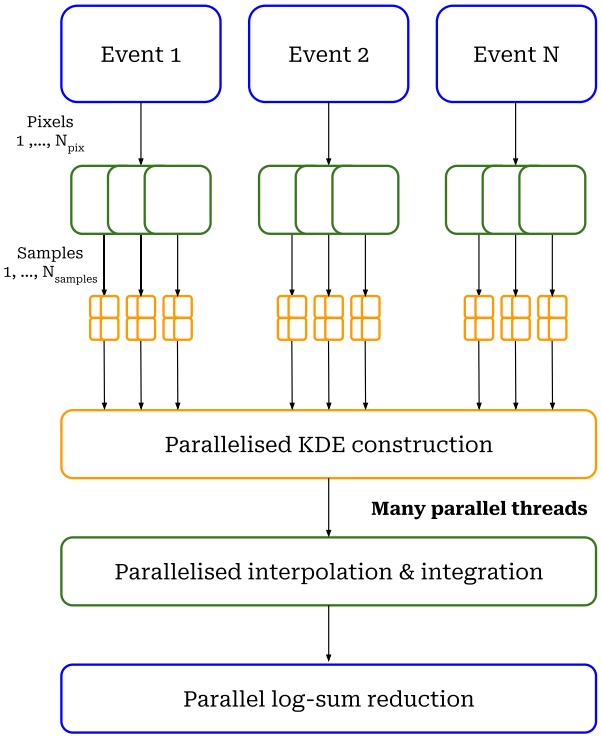


FIG. 2. Parallel GPU based likelihood evaluation scheme. The strength of this implementation is to leverage many parallel threads in order to simultaneous process events, pixels and samples. The parallelized KDE construction, interpolation and integration, and log-sum reduction work in tandem to produce a significant performance upgrade relative to the sequential implementation.

significant increase in computational performance. The cost of this process is in theory $\mathcal{O}(1)$, i.e., constant with respect to the size of the inputs, but in practice this will be limited by both the part of the likelihood which is not computed in parallel, and by the total GPU memory available.

To allow for this, we implement a PyTorch [43] framework for `gwcosmo`, making population models, cosmological calculations and selection effects computation PyTorch compatible. Additionally, a number of custom CUDA [44] kernels are implemented using the `Numba` [45] library. In particular, the vectorized computation of KDEs eliminate the computational bottleneck caused by sequential KDEs. These kernels allow for many operations to be carried out at low level using shared memory, removing the need to instantiate large intermediate arrays during the likelihood evaluation. Despite these substantial changes to allow the code to run in a massively parallel framework, it retains the CPU-only capability of previous versions—ensuring backward compatibility.

C. Memory reduction strategies

The structure of a three-dimensional rectangular array containing the GW event posterior samples is very effective in improving the performance of the pipeline, however this is a trade off in terms of the memory required. Given that GW posterior samples can number in the thousands for a given event, and the size of one array dimension is set by the maximum amount in *any* pixel in *any* event, the memory requirement of analysis can balloon with only a small number of events. In order to combat this, we implement the option to randomly downsample the GW posterior samples in a given pixel. In the limit of many thousands of samples, this still allows the KDE to accurately represent the distribution in each pixel.

Additionally, for a catalog of many thousands of events, we can directly decrease the size of the array being evaluated by batching the likelihood. This is particularly useful when using a device with less video random access memory (VRAM), but also sees a decrease in computational performance proportional to the fraction of events evaluated in each batch.

III. RESULTS

A. Acceleration

To demonstrate the improved computational performance of `gwcosmo`, we benchmark the code using 2000 simulated GW events, more than ten times the

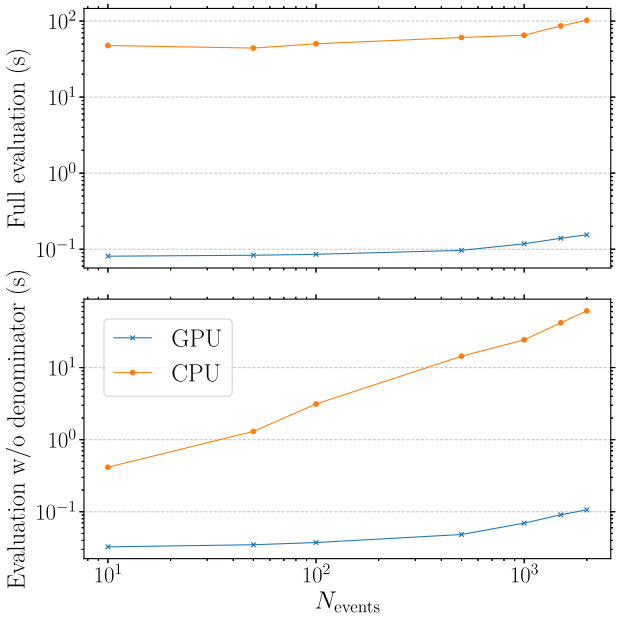


FIG. 3. Mean single likelihood evaluations times for the CPU and GPU implementations as a function of events, averaged over 100 evaluations. Both curves use the full set of GW event samples for a direct comparison. GPU values are calculated using an Nvidia H100 GPU. Top panel: Evaluations of both the numerator and denominator of the likelihood. For the legacy CPU implementation the evaluation of selection effects becomes a dominant cost. Bottom panel: Evaluations of the numerator only, where the linear scaling of the legacy CPU implementation is clear.

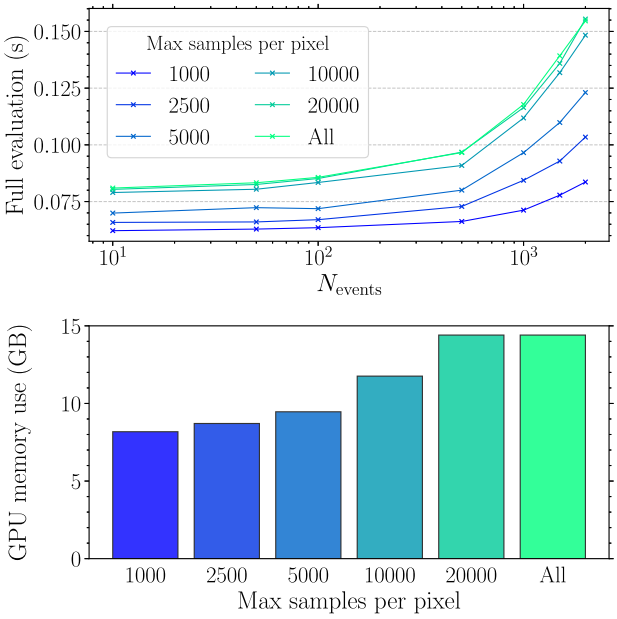


FIG. 4. Top panel: Likelihood evaluation times for the GPU code as a function of GW event catalog size, with different pixel downsampling limits, averaged over 100 evaluations. By decreasing the maximum number of samples used the mean evaluation time can be decrease by a further factor of 2 for the largest catalog size. Bottom panel: Maximum memory usage as a function of downsampling limits when using a full event catalog of 2000 GW sources. Event a conservative downsampling can significantly reduce the total memory required by the code. These values were produced enforcing a minimum number of pixels per event of 5.

maximum number used in an analysis to date. The generation of this data is described in Appendix A. We compare the GPU implementation to the CPU code used in [37], and all values are produced with an Nvidia H100 GPU using Hopper architecture unless explicitly stated otherwise. The CPU values presented here were produced using 2.4 GHz Xeon E5-2630v3 CPUs, with the results in Fig. 6 parallelized over 32 processes. When stating times we take the mean evaluation time over 100 evaluations of the likelihood for hyperparameters randomly drawn from the prior, after ten warm-up evaluations which are discarded.

In Fig. 3 we compare the two versions of `gwcosmo` as a function of the number of events in the catalog, ranging from 10 to 2000. The top panel includes the total evaluation time [numerator + denominator + parameter updates, from Eq. (5)], while the bottom panel is the full likelihood without the denominator included. This was motivated by the fact that for O5-like sensitivity [46], a very large number of injections are required to calculate selection effects—in the previous version of `gwcosmo` this becomes a dominant cost. This is due to the larger parameter volume which becomes accessible to the detector—in order to accurately calculate Eq. (4) we require many more injections. When excluding this, the CPU evaluation curve returns to an expected linear scaling with N_{events}. The key takeaway is the improvement of a factor of almost 1000 when $N_{\text{events}} = 2000$, ensuring the scalability of `gwcosmo` for O5-like analyses. Although in theory a perfectly vectorized implementation should not scale at all with N_{events}, the small increase in computational time taken on the GPU can be ascribed to a small number of operations which still require iterating over events, pixels or samples—as opposed to GPU saturation.

It is possible to further decrease the computational cost of evaluating the likelihood by using a subset of the total posterior samples available per pixel, as described in Sec. II C. In Fig. 4 we see that by decreasing this cap (effectively also a cap on the total number of samples), an improvement of a factor of 2 in performance can be achieved. In the lower panel of the same figure, the potential memory-saving effects of this are also clear. In Sec. III C we investigate the effects of this on the final posterior.

In Table I we break down the cost of a single likelihood evaluation by component and compare the improved performance relative to the legacy CPU implementation. We break evaluation into the numerator, denominator and other data processing operations, and compared these for the CPU, and two GPU settings—fiducial (with all samples) and downsampled (using a cap of 2500 samples per

TABLE I. Comparison of the evaluation cost for the legacy CPU likelihood and the GPU implementation, broken down by component when using a catalog of 2000 GW events. The "CPU" column profiles the legacy implementation, while "GPU fiducial" reports the GPU likelihood using all PE samples, and "GPU downsampled" reports the GPU likelihood with a maximum of 2500 samples per pixel. The KDE evaluation, interpolation and selection effect calculation dominate the total time in the CPU implementation, and are significantly accelerated by the GPU implementation.

	CPU	GPU fiducial		GPU downsampled	
Component	Time (ms)	Time (ms)	Speedup	Time (ms)	Speedup
Numerator					
KDE	29780.13	75.59	394.0×	28.22	1055.3×
Interpolation and rate	32156.28	1.04	30919.5×	1.02	31525.8×
Integration	200.45	0.19	1055.0×	0.19	1055.0×
Denominator					
Selection effect	41772.22	48.70	857.7×	48.76	856.7×
z Prior	< 0.01	0.53	0.33×	0.52	0.33×
Other Operations					
Sample reweighting	< 0.01	22.55	0.00005×	14.69	0.00008×
Parameter update	0.81	6.61	0.12×	6.43	0.13×
Fill Arrays	< 0.01	5.38	0.00004×	1.88	0.0001×
Redshift rate	0.12	0.11	1.10×	0.11	1.09×
Total	103910.31	160.70	646.6×	101.82	1020.5×

pixel). The main operations in the numerator are the construction of KDEs and interpolating and summing over pixels, before integrating over the expression. These operations are extremely costly in the CPU implementation, making up over half the total evaluation time. Even for the fiducial GPU case, the KDE is a factor of almost 400× faster, while the interpolation and summation are sped up by a factor of greater than 3×10^4. In the CPU implementation, the denominator becomes a more significant computational bottleneck (independent of the GW catalog size), when considering detectors with higher sensitivities, as a greater number of injections are required to evaluate the selection effects. The effect of this is clear in the CPU column of Table I. By instead handling this with a PyTorch framework the cost is reduced by a factor of almost 900. For more straightforward operations on flattened arrays such as reweighting of GW event samples or filling arrays, the GPU implementation actually increases the cost due to additional overhead, however the effect of this on the total evaluation time is negligible due to enormous savings in other operations. Overall, the improvement in performance is stark—with even the most conservative settings providing inference accelerated by a factor of greater than 600. In addition to this, introducing mild downsampling to the posterior samples in each pixel can result in a further acceleration, meaning a single likelihood evaluation is complete more than one thousand times quicker than with the CPU implementation.

B. Code validation

In order to verify the consistency of the GPU implementation of `gwcosmo`, we report two tests: firstly, recovering the injected hyperparameters of a simulated population of 2000 GW events—representing an O5-like magnitude—in order to demonstrate the self-consistency of our method. The generation of this simulation population is described in Appendix A. Secondly, we analyze 141 real GW events from GWTC-4.0 and look for agreement with the same analysis produced by the previous CPU-only implementation of the framework. For all analyses presented in this Section we employ the Bilby parameter estimation library, with the nested sampler `nessai` [47–49].

Figure 5 presents the results of our analysis with simulated data for two different settings, with all samples being used and when capping the maximum number allowed per sky pixel. Using 2000 mock GW events in a spectral siren analysis (without galaxy catalog information), we find that the new code version correctly recovers the injected hyperparameters for both cases, with the posteriors in good agreement. It also projects that such a spectral analysis with the BBH-only MULTIPEAK mass model produces a constraint of $H_0 = 64.01^{+5.20}_{-5.30}\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$, or with an uncertainty of 16%. This analysis with an O5-like event count was able to reach convergence in a sampling time of 21 hours on a GPU when using the maximum available number of event samples per pixel, and when capping this at 2500 samples, of 17 hours. This cap on maximum samples per pixel is motivated in Sec. III C. It is worth noting that this analysis would be infeasible with the previous CPU version of `gwcosmo`, due to the total computational cost.

To demonstrate the ability of the updated code to produce posteriors in agreement with previous iterations,

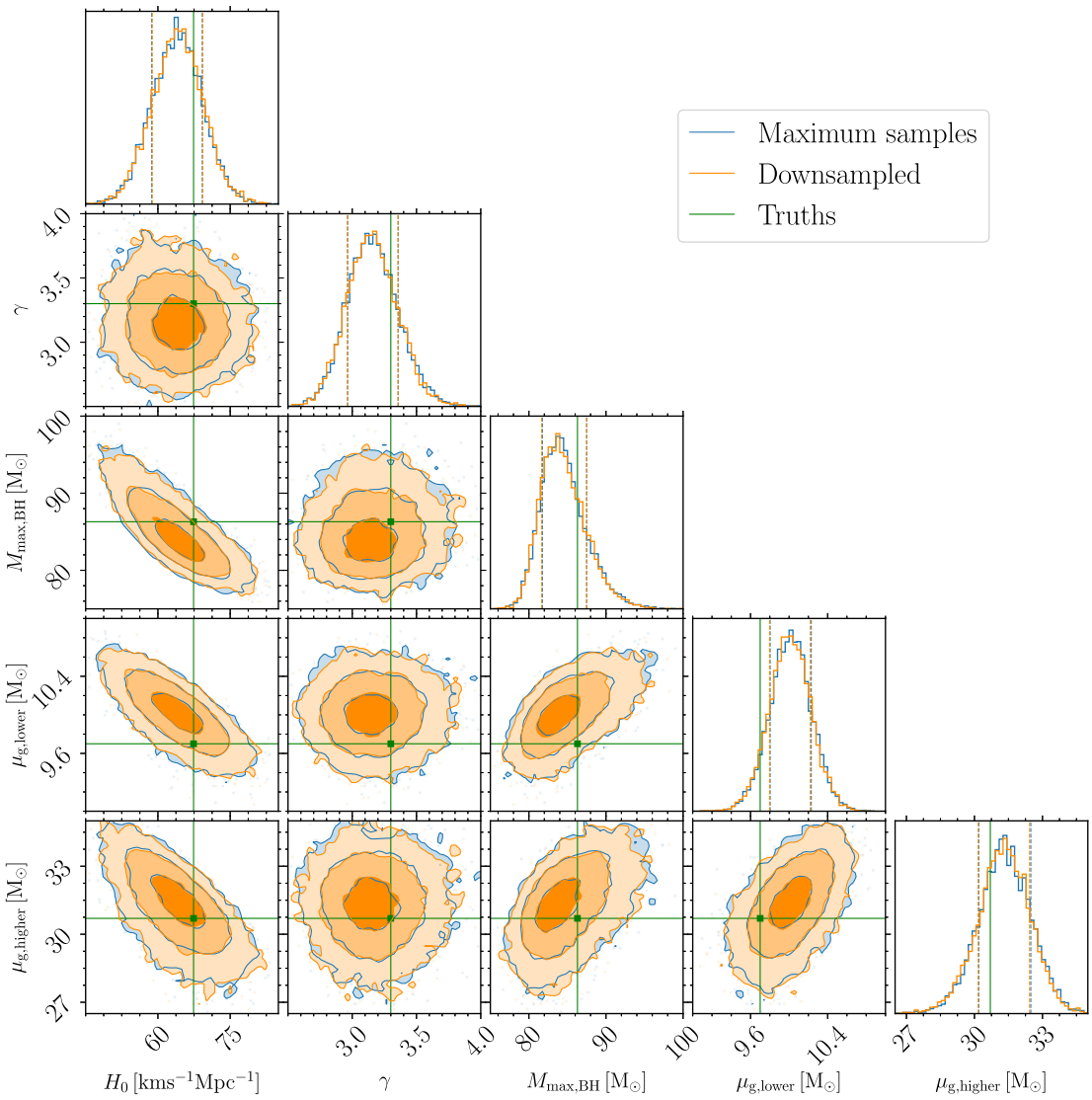


FIG. 5. Corner plot of recovered hyperparameters using a simulated set of 2000 GW events at O5 sensitivity using the MULTIPEAK mass model, analyzed with the GPU version. A subset of the total 15 hyperparameters are included here, of which the most cosmologically informative are chosen along with H_0. The blue (orange) posteriors pertain to analysis using all (2500) samples allowed in each pixel. Both sets of posteriors are in good agreement and recover the injected values, shown by the green lines.

we compare identical dark siren analyses using GWTC-4.0 events using the FULLPOP-4.0 [37,50] model. This is presented in Fig. 6, with the legacy CPU result in blue and two GPU results with different maximum sample caps in orange and green. These posteriors demonstrate excellent agreement, which is quantified using the Kullback-Leibler (KL) divergence between the marginal posteriors of each parameter [51]. The GPU posteriors are considered in agreement with the CPU reference when the maximum of the marginal KL divergences falls below a noise tolerance bound established from the CPU distribution itself. This process is described in more detail in Appendix B. For the full GPU analysis, the maximum marginal KL is 94.0 mbits, compared to a noise bound of 118.7 mbits, confirming agreement. The downsampled GPU analysis achieves a maximum marginal KL of 86.4 mbits, demonstrating that the approximation does not degrade the inference.

Beyond this, the improvement in performance between the two implementations is significant—the GPU using all samples analysis reached convergence in 4 days (96 hours) of sampling, compared to 32 days (768 hours, or 24,576 hours core-hours distributed over 32 cores) for the legacy CPU analysis. When decreasing the maximum number of samples per pixel to 2500, this is further improved to just 15 hours of sampling time—highlighting the performance gain from this approximation, while also showing that it does not degrade the inference. It is also worth noting that this comparison does not take into account the CPU analysis being parallelized over 32 cores,

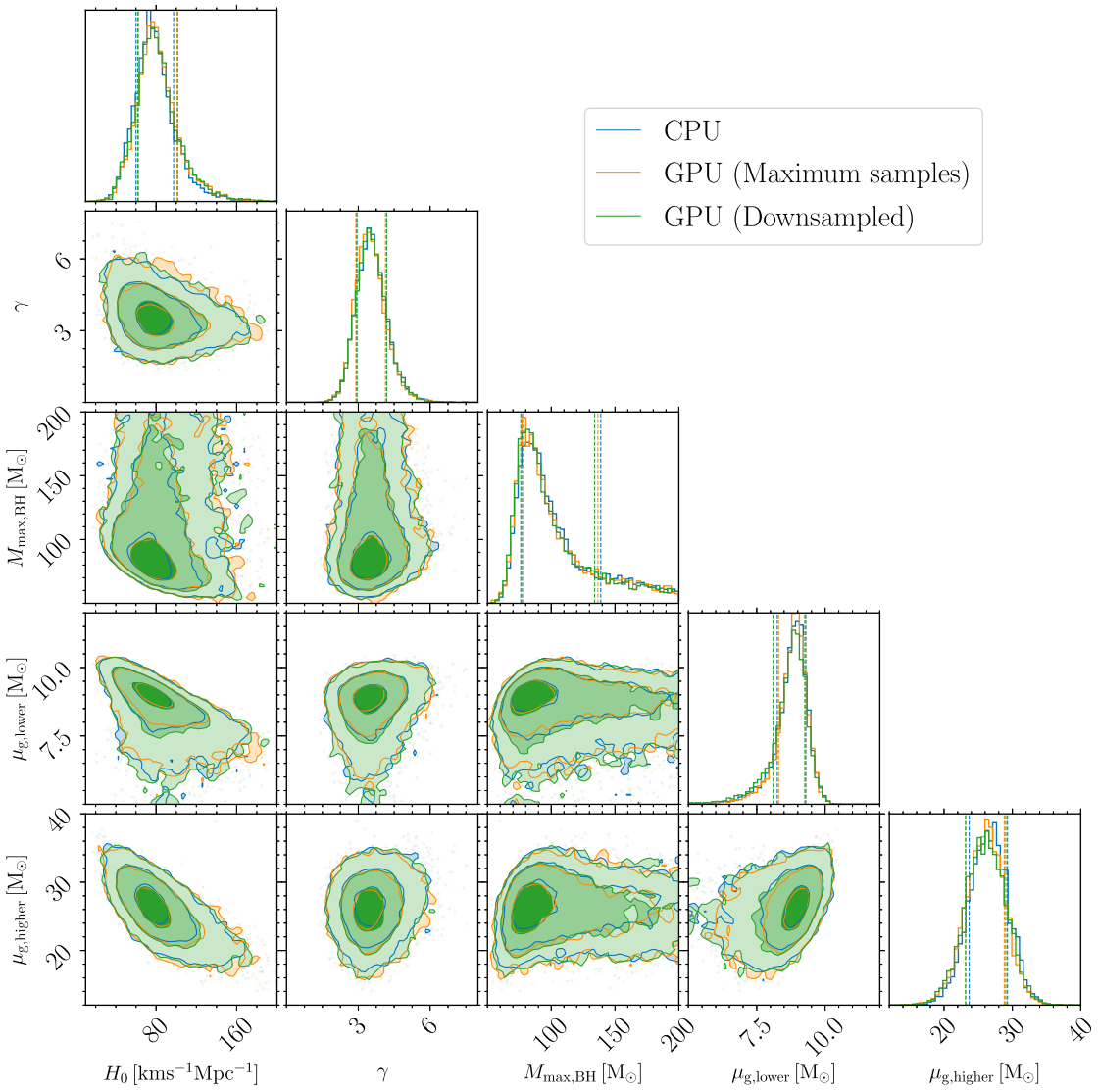


FIG. 6. Comparison of posteriors produced using the legacy CPU implementation and the GPU implementation of `gwcosmo` for a dark siren analysis with GWTC-4.0 using the FULLPOP-4.0 mass model. The blue posteriors represent the CPU analysis, while the orange (green) posteriors show analysis using all (2500) allowed samples per pixel. The posteriors demonstrate excellent agreement between all three analyses, with the GPU analysis using all samples. We determine agreement between the posteriors by calculating a bootstrap noise threshold using the CPU posterior (described in Appendix B), which we find to be 118.7 mbits. We find that in comparison with this, maximum marginal KL values for the GPU (94.0 mbits) and GPU downsampled (86.4 mbits) both fall beneath this, indicating agreement between the posteriors. These three analyses converged in 768 hours, 96 hours and 15 hours for the CPU, GPU and downsampled GPU results, respectively.

despite this GPU analysis being significantly faster in both cases. In Appendix C we also note that the energy consumption of the two different implementations is very different, with the CPU-bound inference requiring an order of magnitude more energy to reach convergence than the GPU implementation using downsampling.

These analyses cannot be directly compared to the time required in our mock data analyses however, due to the presence of an additional eight hyperparameters in the FULLPOP-4.0 mass model relative to the MULTIPEAK, as well as the FULLPOP-4.0 model being more complex and taking longer to evaluate for a given set of $\Lambda_{\rm pop}$. For more details on the construction of the two models, see [37].

C. Systematic tests

The improved performance of `gwcosmo` enabled a systematic validation of several implementation choices. Five configuration settings were varied to assess their impact on the resulting H_0 posterior: the previously introduced cap on the maximum number of samples per sky pixel, the bandwidth selection method for the sky-pixel KDEs, the resolution of the KDE evaluation grid, the

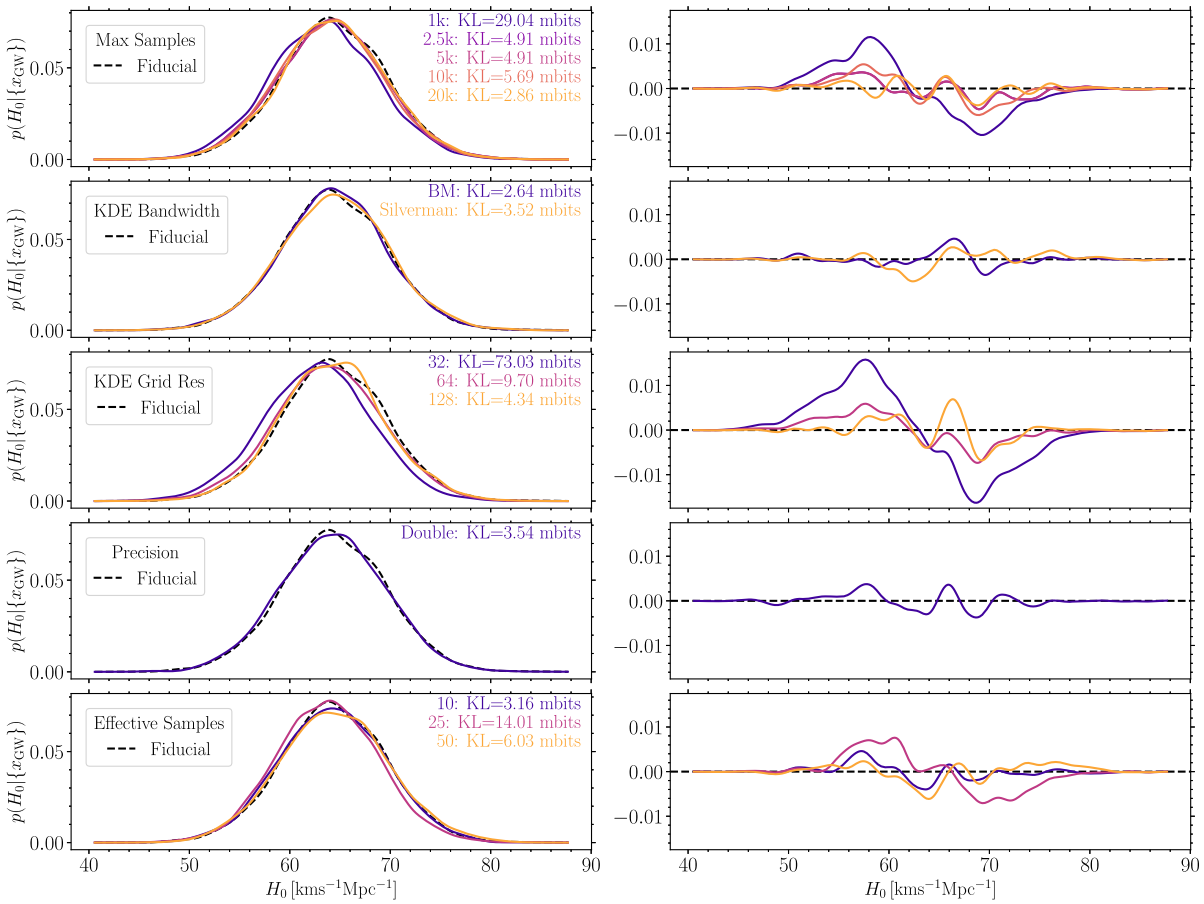


FIG. 7. Left panel: H_0 posteriors for systematic test cases, in descending order: maximum allowed samples per pixel, KDE bandwidth estimator, KDE grid resolution, floating point precision, minimum number of effective samples. The fiducial case is shown with the dashed black line, against which the KL divergence are calculated, whilst different tests are shown in different colors. Right panel: Residuals for H_0 posteriors for systematic test cases, calculated against the fiducial case represented by the black dashed line.

floating-point precision, and the minimum effective number of PE samples required per pixel. Each case was compared to the "fiducial" analysis settings, using all available GW event samples, single precision floats, the densest KDE grid, the default bandwidth estimator and $N_{\rm eff,PE} = 2$ as defined in Eq. (8).

The left panel of Fig. 7 shows the resulting H_0 posteriors for each test, as well as the KL divergence between the fiducial and test cases. We determine that a KL value below or equal to 5.96 mbits for the H_0 posterior indicates agreement, a choice which is motivated in Appendix B. Based on this criterion, we find that it is possible to place a cap of 2500 samples per sky pixel without seeing a significant shift in the final H_0 posterior or losing any constraining power relative to an analysis using all GW event samples. Below this value however we find a difference in the posterior no longer consistent with noise from a finite number of samples being use to calculate the KL divergence. The maximum sample cap does give an additional improvement in performance, with a run using a cap of 2500 samples converging in ∼15 hours, compared to ∼21 hours when using the complete set of GW event samples. There is a slight shift in the posterior at the most aggressive cut on samples. This occurs because uniform downsampling inside a pixel alters the representation of the asymmetric sample distributions typical of GW distance posteriors. The resulting loss of resolution in the KDE tails ultimately affects the shape and position of the H_0 posterior. In order to ensure that this is not a systematic effect, or will scale as a function of catalog size, we compute median H_0 values with 60% confidence intervals in Table II, with the relative shift from the maximum sample case. From these values it is clear that the shift is not systematic—it oscillates near the maximum sample value until the most aggressive cap. Our fiducial value of 2500 samples per pixel has a relative shift of −0.3% for $N_{\rm events} = 2000$. With the GWTC-4.0 catalog of 141 events, we find the same relative shift to be −0.4%, indicating that this is not an effect which scales proportionally to catalog size.

We also find that calculating KDE bandwidths with more conservative estimators does not significantly affect the resulting posteriors on H_0. The default estimator uses Scott's rule of thumb [52], but we additionally consider Silverman's rule of thumb [53], and the biweight-midvariance [54]. These were chosen to be more robust

TABLE II. Median H_0 and 68% credible interval as a function of the maximum number of samples retained per pixel, for the O5-like mock catalog of 2000 GW events. The relative shift column gives the fractional difference in the median relative to the full-sample case. The posterior shifts noticeably only at the most aggressive cap of 1000 samples, from 2500 samples upward the median is stable to within the scatter expected from finite-sample noise.

Max. samples	H_0 (km s^{-1} Mpc^{-1})	Rel. shift
1000	$63.43^{+5.22}_{-5.35}$	−1.38%
2500	$64.13^{+5.16}_{-5.31}$	−0.30%
5000	$64.25^{+5.18}_{-5.30}$	−0.11%
10 000	$64.07^{+5.14}_{-5.29}$	−0.39%
20 000	$64.33^{+5.18}_{-5.04}$	0.02%
All	$64.32^{+5.18}_{-5.04}$	...

to potential outliers in the sample set, and are outlined in more detail in Appendix D. Despite this, neither has a significant effect on the posterior, nor do they affect the total convergence time for analysis.

A greater effect is found by decreasing the resolution of the KDE evaluation redshift grid from the fiducial 256 points. This is the number of points over which the KDE is constructed, and by increasing or decreasing it the ability of the KDE to recreate structure is improved or worsened. As seen in the center row of Fig. 7, when this is below 128 points the posterior shifts relative to the fiducial case and no longer meets our KL divergence criterion. This can be explained by the fact that the grid is no longer dense enough to correctly capture the structure in the GW event samples. Indeed, this is one area in which the new version of `gwcosmo` improves over it's predecessor, which only used 100 points for the KDE evaluation grid due to linear computational cost. This obstacle is removed in the new, vectorized code and as such finer grids can be used.

The results also demonstrate that using single precision floating point numbers shows no disadvantages relative to double precision. Indeed, the final posteriors meet our criterion for agreement, while the double precision analysis took over double the total analysis time compared to single precision, with ∼46 hours compared to ∼21 hours. This test allows us to state that our analyses are unchanged by the use of single single precision floats.

Lastly, in the bottom left panel of Fig. 7, we test the effect of numerical stability criteria on our analysis. Each pixel in the likelihood must pass a threshold on $N_{\mathrm{eff,PE}}$, as defined in (8). If a pixel does not pass the threshold on $N_{\mathrm{eff,PE}}$ then it does not contribute to the likelihood. This threshold is also dependent on the minimum number of pixels enforced per event, as having fewer pixels with more samples means each is more likely to pass the threshold. However we do not vary this in this work, so that the systematic tests presented here remain consistent. We find that imposing a harsher threshold on the number of effective GW event samples in the bottom row of the left panel of Fig. 7 does show deviations in the posterior. Increasing this above $N_{\mathrm{eff,PE}} = 10$ introduces more variance to the shape of the posterior and it no longer passes our agreement criterion.

IV. CONCLUSIONS

We have developed a new implementation of `gwcosmo`, which takes advantage of the massively parallel computational ability of GPUs, by enabling near-fully vectorized evaluation of the likelihood over GW events, pixels and samples. By restructuring the hierarchical likelihood and using a dense tensor containing the GW data, as well as implementing PyTorch functionality and custom `CUDA` kernels for key operations, we have eliminated the dominant sequential bottlenecks which were present in previous CPU-only versions of `gwcosmo`. Beyond this, it also retains its predecessor's CPU capability, to allow analysis in different computing environments.

This results in a potential performance increase of $\mathcal{O}(10^3)$, reducing likelihood evaluation times from minutes to milliseconds for larger gravitational wave event catalogs. Given this, we demonstrated the ability of `gwcosmo` to analyze O5-like numbers of GW events [$\mathcal{O}(10^3)$] in significantly reduced time, making previous intractable analyses computationally feasible. The GPU implementation is also capable of analysing substantially larger catalogs, by using the batched likelihood as described in Sec. II C, at the cost of computational performance. In order to further optimize the memory efficiency of the code, and allow analyses of larger event catalogs, it will be necessary to eliminate padded arrays and replace them with flattened contiguous memory layouts and offset-based ragged tensor representations, reducing wasted memory and improving GPU utilization.

With this new implementation, we demonstrated consistency of the pipeline through both the recovery of injected hyperparameters from a simulated population of GW events, as well as agreement with posteriors produced with real data using the legacy CPU implementation. As shown in Appendix C this also provides much improved energy efficiency for cosmological inference, with a single analysis requiring an order of magnitude less energy when using a GPU.

Our testing also reveals that single-precision floating point arithmetic, or moderate downsampling of GW event samples do not introduce significant deviations in the final posteriors. Both of these are valuable tools with two purposes, allowing for mitigation of out-of-memory issues, as well as additional performance increases. However a decrease in the resolution of the KDE grid or an aggressive stability cut on sky pixels have a more significant effect on the final posterior.

The new vectorized implementation of gwcosmo also makes it competitive in performance with other cosmological inference pipelines. For example, icarogw [35] uses Monte Carlo integration rather than KDEs to evaluate GW posterior samples, yet achieves comparable runtimes. In the case of CHIMERA [24,36], the "many-1D" kernel which is most similar to that in gwcosmo has a similar evaluation time for O5-like event numbers. An important caveat for these comparisons is the use of different GPU devices, and so they are not a direct comparison.

Overall, this implementation will enable a new regime of dark siren cosmology using gwcosmo, ensuring cosmological inference remains computationally tractable, even in an era of ever-expanding GW event catalogs. This capability opens the door to more sophisticated population modeling, the incorporation of new galaxy catalogs (see for example [55], which made use of the version of gwcosmo presented in this paper), and extended studies of modified gravitational-wave propagation.

ACKNOWLEDGMENTS

A. P. is supported by UKRI STFC studentship 323353-01. C. E. A. Chapman-Bird thanks Bruno Borbely for providing access to GPU resources early in this project, and is supported by UKSA Grant No. UKRI971. R. G. and C. M. are supported by United Kingdom Research and Innovation Grant No. UKRI2487. T. B. acknowledges support from a CDSN Ph.D. grant from ENS-PSL. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459, as well as by the IN2P3 computing center (CC-IN2P3) in Lyon (Villeurbanne).

DATA AVAILABILITY

There are no publicly available research data or software supporting this manuscript. Requests for further information or data should be sent to the authors.

APPENDIX A: DATA

Simulated data. To test and benchmark the performance of gwcosmo, we simulate a catalog of mock gravitational wave events at O5 sensitivity [46]. These were generated assuming a realistic 4-detector LVK network complete with 70% uncorrelated duty cycles, and PE carried out using Bilby with the IMRPhenomXHM waveform and Gaussian noise. The population of GW events was drawn from the MULTIPEAK mass model, covering only the BBH range of the CBC mass spectrum, as first used in [56] and assuming a Madau-Dickinson distribution in redshift [57]. We generate 2000 GW event posteriors with an SNR > 12, which approximates the number expected for three years of observing at O5 design sensitivity. This is not intended to be a projection of the potential constraints on H_0 in this regime, but to demonstrate that gwcosmo can analyze such a magnitude of events. Injected hyperparameter values for the generated population can be found in Table III.

Real data. To demonstrate agreement between the current and previous versions of gwcosmo we also include analysis using GWTC-4.0 data. Considering a selection threshold of IFAR < 4 yr^{-1}, consistent with that in [37], we use 141 CBCs in a dark siren analysis with the FULLPOP-4.0 mass model and the GLADE+ K-band galaxy catalog with weighting proportional to galaxy luminosity. Additionally to evaluate our selection function we use the publicly available injections campaign [58].

APPENDIX B: KL VALIDATION

Single marginal posterior threshold. In order to determine a threshold to decide on agreement between single marginal posteriors with the KL divergence, we employ a toy model with two identical normal distributions. By drawing 7000 samples from both distributions (representative of the number of posterior samples produced in our analyses) and calculating the KL divergence for 1000 realizations. We find that 90% of the values fall below 5.96 mbits, which can be seen in Fig. 8. This is similar to methods described in [59,60] using another similar metric, the Jensen-Shannon divergence [61]. Given this, we elect to consider the marginal H_0 posteriors in Sec. III C with values in or below this range to be in statistical agreement with one another, accounting for a finite number of samples. In the case of N-dimensional posteriors, we multiply this value by N in order to assess

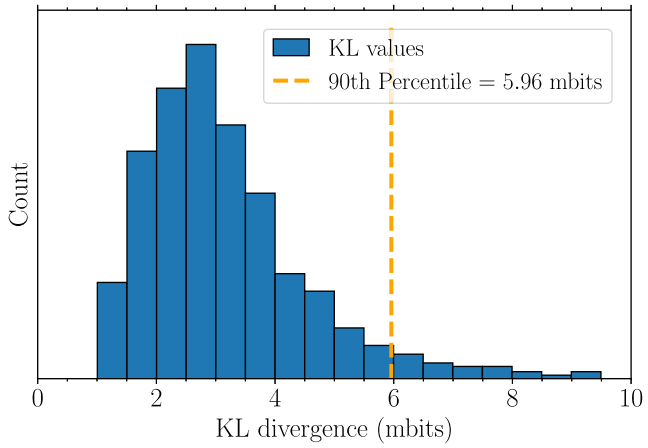


FIG. 8. KL divergences calculated from two identical normal distributions, using $N_{\text{samples}} = 7000$ in order to show the scatter from using a finite number of samples. This was calculated 1000 times, and the 90th percentile, denoted by the orange dashed line, was found to be 5.96 mbits even when comparing identical distributions.

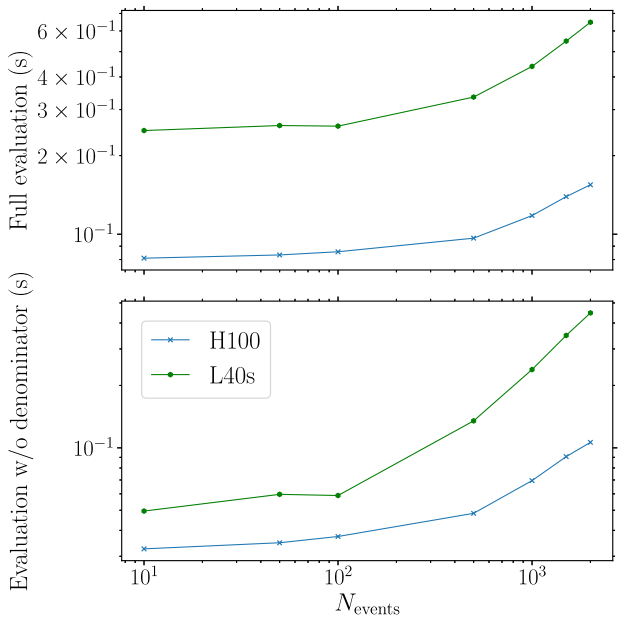


FIG. 9. Bayesian inference wall-clock time comparison using difference devices. The NVIDIA H100 outperforms the more affordable L40S by a factor of 4 when using a full catalog of mock GW events. The performance gap is primarily attributed to the H100's superior memory bandwidth, making the top-of-the-range hardware more efficient for memory-bound likelihood calculations required in cosmological parameter estimation.

agreement. Additionally, we repeated this metric test using a skewnormal distribution, which has a heavier tail on one side, similar to the asymmetry seen in some of our posteriors. We found that this gave a marginally more stringent value of 5.85 mbits, in good agreement with the value obtained from the normal distribution.

N-dimensional posterior threshold. Second, in order to quantify agreement between GPU and CPU posterior implementations in many dimensions, we establish a threshold on the noise from finite samples using symmetric resampling. This is relevant for the 23-dimensional consistency test in Sec. III B, and means we avoid any assumptions about parameter correlations or normality in posteriors. We generate a null distribution by resampling the CPU posterior 1000 times—at each iteration, two independent half-sample subsets are drawn without replacement and the marginal KL divergence is computed for each of the 23 parameters using histogram-based estimation. The maximum of the marginal KL values across all parameters is calculated at each iteration. The 90th percentile of this distribution defines the noise threshold used to determine agreement. To assess the GPU posterior, we draw half-sample subsets from both CPU and GPU posteriors and compute the maximum of the marginal KL divergences as our test statistic. The GPU posterior is considered in agreement with the CPU posterior if this test statistic falls below the noise tolerance bound.

TABLE III. Injected hyperparameter values for the simulated gravitational-wave event catalog used in this work. The BBH population parameters are taken from the maximum-likelihood values reported in [37], while the cosmological parameter values are those from Planck18 [63].

Parameter	Value	Parameter	Value
α	2.9	$\mu_{g,\text{high}}(M_\odot)$	30.7
β	1.0	$\sigma_{g,\text{high}}(M_\odot)$	6.3
$m_{\min}(M_\odot)$	4.6	λ_g	0.4
$m_{\max}(M_\odot)$	86.3	$\lambda_{g,\text{low}}$	0.8
$\delta_m(M_\odot)$	4.8	γ	3.3
$\mu_{g,\text{low}}(M_\odot)$	9.7	κ	2.9
$\sigma_{g,\text{low}}(M_\odot)$	0.7	z_p	2.5
$H_0(\text{km s}^{-1}\,\text{Mpc}^{-1})$	67.4	Ω_m	0.315

APPENDIX C: DEVICE COMPARISON AND ENERGY CONSIDERATIONS

Given that our analyses make use of a top-of-the-range GPU in the Nvidia H100 (Hopper architecture), we also compared this to the performance when using a more affordable device. This was chosen to be the Nvidia L40s (Ada Lovelace architecture), a device with a smaller overall memory and lower memory bandwidth, but improved single-precision optimisation. The results of which can be seen in Fig. 9. While still reporting a significant improvement in performance over the legacy CPU implementation, the L40s evaluates the likelihood a factor of ∼4 slower than the H100 when $N_{\text{events}} = 2000$. This points to the overall bottleneck on this device being memory bandwidth.

In Table IV, we compare the energy consumption required to complete an analysis of GWTC-4.0 using 141 CBCs and the FULLPOP-4.0 mass model with the legacy CPU and updated GPU implementation. Using figures from [62], the mean daily energy consumption for a residence in the European Union is taken to be 38.2 kWh, which is used to calculate the days equivalent row. The GPU implementation requires an order of magnitude less energy to complete this analysis when compared to its CPU-bound counterpart.

TABLE IV. Comparison of computational cost, energy consumption, and estimated carbon emissions between a 32-core CPU implementation and an H100 GPU implementation with a maximum of 2500 samples allowed per sky pixel. The wall time is that of completing one analysis of GWTC-4.0 with the FULLPOP-4.0 mass model, the same as those shown in Sec. III.

Metric	CPU (32 cores)	H100 GPU
Wall time	32 days (768 hours)	15 hours
Compute time	24,576 core-hours	15 GPU-hours
Energy consumption	230.4 kWh	10.5 kWh
Days equivalent	≈6	≈0.27

APPENDIX D: BANDWIDTH ESTIMATORS

Scott's rule provides an optimal bandwidth (h) for kernel density estimation, assuming the underlying data distribution is Gaussian [52]. While computationally efficient, it relies heavily on the sample standard deviation, $\hat{\sigma}$, making it sensitive to outliers which can artificially inflate the bandwidth and oversmooth the density estimate. It takes the form

$$h = 1.06\hat{\sigma}n^{-1/5}. \tag{D1}$$

Silverman's rule improves upon Scott's by introducing a more robust measure of spread [53]. By taking the minimum of the standard deviation and the scaled interquartile range (IQR), it prevents extreme outliers from excessively widening the bandwidth. It is more often used in cases where the distribution deviates from Gaussianity,

$$h = 0.9 \ \min\left(\hat{\sigma}, \frac{\mathrm{IQR}}{1.34}\right) n^{-1/5}. \tag{D2}$$

The biweight midvariance is a robust estimator of scale [54]. Unlike the previous rules, it employs an indicator function $I(|u_i| < 1)$ that acts as a "hard limit" assigning zero weight to any observations falling more than nine median absolute deviations (MAD) from the center. This makes it significantly more resistant to heavy-tailed distributions than either Scott's or Silverman's methods. It is implemented as

$$h = \frac{n\sum_{i=1}^{n}(x_i - Q)^2(1-u_i^2)^4 I(|u_i|<1)}{\left(\sum_i (1-u_i^2)(1-5u_i^2)I(|u_i|<1)\right)^2}, \tag{D3}$$

where the auxiliary variable u_i is standardized by the median, Q, and the MAD,

$$u_i = \frac{x_i - Q}{9 \cdot \mathrm{MAD}}. \tag{D4}$$

[1] B. F. Schutz, Determining the Hubble constant from gravitational wave observations, Nature (London) **323**, 310 (1986).
[2] D. E. Holz and S. A. Hughes, Using gravitational-wave standard sirens, Astrophys. J. **629**, 15 (2005).
[3] M. Moresco *et al.*, Unveiling the universe with emerging cosmological probes, Living Rev. Relativity **25**, 6 (2022).
[4] C. Messenger and J. Read, Measuring a cosmological distance-redshift relationship using only gravitational wave observations of binary neutron star coalescences, Phys. Rev. Lett. **108**, 091101 (2012).
[5] S. Nissanke, D. E. Holz, S. A. Hughes, N. Dalal, and J. L. Sievers, Exploring short gamma-ray bursts as gravitational-wave standard sirens, Astrophys. J. **725**, 496 (2010).
[6] B. P. Abbott *et al.*, A gravitational-wave standard siren measurement of the Hubble constant, Nature (London) **551**, 85–88 (2017).
[7] M. Mancarella, F. Iacovelli, S. Foffa, N. Muttoni, and M. Maggiore, Accurate standard siren cosmology with joint gravitational-wave and γ-ray burst observations, Phys. Rev. Lett. **133**, 261001 (2024).
[8] A. Cozzumbo, U. Dupletsa, R. Calderón, R. Murgia, G. Oganesyan, and M. Branchesi, Model-independent cosmology with joint observations of gravitational waves and γ-ray bursts, J. Cosmol. Astropart. Phys. 05 (2025) 021.
[9] D. F. Chernoff and L. S. Finn, Gravitational radiation, inspiraling binaries, and cosmology, Astrophys. J. Lett. **411**, L5 (1993).
[10] S. R. Taylor, J. R. Gair, and I. Mandel, Cosmology using advanced gravitational-wave detectors alone, Phys. Rev. D **85**, 023535 (2012).
[11] W. M. Farr, M. Fishbach, J. Ye, and D. E. Holz, A future percent-level measurement of the Hubble expansion at redshift 0.8 with Advanced LIGO, Astrophys. J. Lett. **883**, L42 (2019).
[12] J. M. Ezquiaga and D. E. Holz, Jumping the gap: Searching for LIGO's biggest black holes, Astrophys. J. Lett. **909**, L23 (2021).
[13] J. M. Ezquiaga and D. E. Holz, Spectral sirens: Cosmology from the full mass distribution of compact binaries, Phys. Rev. Lett. **129**, 061102 (2022).
[14] G. Pierra and A. Papadopoulos, Heavy black-holes also matter in standard siren cosmology, arXiv:2601.03257.
[15] M. Tagliazucchi, M. Moresco, A. Agapito, M. Mancarella, S. Ferraiuolo, S. Mastrogiovanni, N. Borghi, F. Pannarale, and D. Bonacorsi, Pushing spectral siren cosmology into the third-generation era: A blinded mock data challenge, Astron. Astrophys. **711**, A120 (2026).
[16] T. Bertheas, V. Gennari, D. Steer, and N. Tamanini, Spectral sirens cosmology from binary black holes populations with sharper mass features, arXiv:2603.06792.
[17] W. Del Pozzo, Inference of cosmological parameters from gravitational waves: Applications to second generation interferometers, Phys. Rev. D **86**, 043011 (2012).
[18] M. Soares-Santos *et al.*, First measurement of the Hubble constant from a dark standard siren using the dark energy survey galaxies and the LIGO/Virgo binary–black-hole merger GW170814, Astrophys. J. Lett. **876**, L7 (2019).
[19] A. Palmese *et al.*, A statistical standard siren measurement of the Hubble constant from the LIGO/Virgo gravitational wave compact object merger GW190814 and dark energy survey galaxies, Astrophys. J. Lett. **900**, L33 (2020).

[20] A. Finke, S. Foffa, F. Iacovelli, M. Maggiore, and M. Mancarella, Cosmology with LIGO/Virgo dark sirens: Hubble parameter and modified gravitational wave propagation, J. Cosmol. Astropart. Phys. 08 (2021) 026.
[21] S. Mastrogiovanni, K. Leyde, C. Karathanasis, E. Chassande-Mottin, D. A. Steer, J. Gair, A. Ghosh, R. Gray, S. Mukherjee, and S. Rinaldi, On the importance of source population models for gravitational-wave cosmology, Phys. Rev. D **104**, 062009 (2021).
[22] R. Gray, C. Messenger, and J. Veitch, A pixelated approach to galaxy catalogue incompleteness: Improving the dark siren measurement of the Hubble constant, Mon. Not. R. Astron. Soc. **512**, 1127 (2022).
[23] S. Mastrogiovanni, D. Laghi, R. Gray, G. C. Santoro, A. Ghosh, C. Karathanasis, K. Leyde, D. A. Steer, S. Perries, and G. Pierra, Joint population and cosmological properties inference with gravitational waves standard sirens and galaxy surveys, Phys. Rev. D **108**, 042002 (2023).
[24] N. Borghi, M. Mancarella, M. Moresco, M. Tagliazucchi, F. Iacovelli, A. Cimatti, and M. Maggiore, Cosmology and astrophysics with standard sirens and galaxy catalogs in view of future gravitational wave observations, Astrophys. J. **964**, 191 (2024).
[25] J. Aasi *et al.* (LIGO Scientific Collaboration), Advanced LIGO, Classical Quantum Gravity **32**, 074001 (2015).
[26] F. Acernese *et al.* (Virgo Collaboration), Advanced Virgo: A second-generation interferometric gravitational wave detector, Classical Quantum Gravity **32**, 024001 (2015).
[27] T. Akutsu *et al.*, Overview of KAGRA: Detector design and construction history, Prog. Theor. Exp. Phys. **2021**, 05A101 (2020).
[28] B. Abbott *et al.*, GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs, Phys. Rev. X **9**, 031040 (2019).
[29] R. Abbott *et al.*, GWTC-2: Compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run, Phys. Rev. X **11**, 021053 (2021).
[30] R. Abbott *et al.*, GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, Phys. Rev. X **13**, 041039 (2023).
[31] A. G. Abac *et al.*, GWTC-4.0: Updating the gravitational-wave transient catalog with observations from the first part of the fourth LIGO-Virgo-KAGRA observing run, Astrophys. J. Lett. **1004**, L22 (2026).
[32] R. Gray, I. M. n. Hernandez, H. Qi, A. Sur, P. R. Brady, H.-Y. Chen, W. M. Farr, M. Fishbach, J. R. Gair, A. Ghosh, D. E. Holz, S. Mastrogiovanni, C. Messenger, D. A. Steer, and J. Veitch, Cosmological inference using gravitational wave standard sirens: A Mock data analysis, Phys. Rev. D **101**, 122001 (2020).
[33] R. Gray, F. Beirnaert, C. Karathanasis, B. Revenu, C. Turski, A. Chen, T. Baker, S. Vallejo, A. E. Romano, T. Ghosh, A. Ghosh, K. Leyde, S. Mastrogiovanni, and S. More, Joint cosmological and gravitational-wave population inference using dark sirens and galaxy catalogues, J. Cosmol. Astropart. Phys. 12 (2023) 023.
[34] S. Mastrogiovanni, D. Laghi, R. Gray, G. C. Santoro, A. Ghosh, C. Karathanasis, K. Leyde, D. A. Steer, S. Perries, and G. Pierra, Joint population and cosmological properties inference with gravitational waves standard sirens and galaxy surveys, Phys. Rev. D **108**, 042002 (2023).
[35] S. Mastrogiovanni, G. Pierra, S. Perriès, D. Laghi, G. C. Santoro, A. Ghosh, R. Gray, C. Karathanasis, and K. Leyde, Icarogw: A Python package for inference of astrophysical population properties of noisy, heterogeneous and incomplete observations, Astron. Astrophys. **682**, A167 (2024).
[36] M. Tagliazucchi, M. Moresco, N. Borghi, and M. Fiebig, Accelerating the standard siren method: Improved constraints on modified gravitational-wave propagation with future data, Astron. Astrophys. **702**, A244 (2025).
[37] A. G. Abac *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), GWTC-4.0: Constraints on the cosmic expansion rate and modified gravitational-wave propagation, Astrophys. J. Lett. **1007**, L17 (2026).
[38] I. Mandel, W. M. Farr, and J. R. Gair, Extracting distribution parameters from multiple uncertain observations with selection biases, Mon. Not. R. Astron. Soc. **486**, 1086 (2019).
[39] S. Vitale, D. Gerosa, W. M. Farr, and S. R. Taylor, Inferring the properties of a population of compact binaries in presence of selection effects, in *Handbook of Gravitational Wave Astronomy* (Springer, Singapore, 2021), pp. 1–60.
[40] M. Fishbach, D. E. Holz, and W. M. Farr, Does the black hole merger rate evolve with redshift?, Astrophys. J. Lett. **863**, L41 (2018).
[41] W. M. Farr, Accuracy requirements for empirically measured selection functions, Res. Not. AAS **3**, 66 (2019).
[42] C. Talbot and J. Golomb, Growing pains: Understanding the impact of likelihood uncertainty on hierarchical Bayesian inference for gravitational-wave astronomy, Mon. Not. R. Astron. Soc. **526**, 3495 (2023).
[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, Pytorch: An imperative style, high-performance deep learning library, arXiv:1912.01703.
[44] J. Nickolls, I. Buck, M. Garland, and K. Skadron, Scalable parallel programming with CUDA, Queueing Syst. Theory Appl. **6**, 40 (2008).
[45] S. K. Lam, A. Pitrou, and S. Seibert, Numba: A LLVM-based Python JIT compiler, in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (New York, NY, Association for Computing Machinery, 2015), pp. 1–6.
[46] B. P. Abbott *et al.*, Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA, Living Rev. Relativity **23**, 3 (2020).
[47] G. Ashton *et al.*, BILBY: A user-friendly Bayesian inference library for gravitational-wave astronomy, Astrophys. J. Suppl. Ser. **241**, 27 (2019).
[48] M. J. Williams, Nested sampling with normalizing flows for gravitational-wave inference, Phys. Rev. D **103**, 103006 (2021).
[49] M. J. Williams, J. Veitch, and C. Messenger, Importance nested sampling with normalising flows, Mach. Learn. Sci. Tech. **4**, 035011 (2023).

[50] A. G. Abac *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), GWTC-4.0: Population properties of merging compact binaries, Astrophys. J. Lett. **1005**, L51 (2026).
[51] S. Kullback and R. A. Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79 (1951).
[52] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, Wiley Series in Probability and Statistics (Wiley, New York, 2009).
[53] B. W. Silverman, *Density estimation for statistics and data analysis* (Chapman and Hall, London, 1986).
[54] R. Wilcox, *Introduction to Robust Estimation and Hypothesis Testing*, 5th ed. (Academic Press, San Diego, CA, 2021).
[55] I. McMahon, D. Laghi, M. Soares-Santos, K. Ackley, G. Dálya, Y. Gençel, D. Sánchez-Cid, F. Andrade-Oliveira, S. MacBride, C. Chapman-Bird, R. Gray, and A. Papadopoulos, Measurement of the Hubble constant using the dark energy survey year 6 gold galaxy catalog and the fourth gravitational-wave transient catalog, Mon. Not. R. Astron. Soc. **550**, stag1167 (2026).
[56] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), Population properties of compact objects from the second LIGO–Virgo gravitational-wave transient catalog, Astrophys. J. Lett. **913**, L7 (2021).
[57] P. Madau and M. Dickinson, Cosmic star-formation history, Annu. Rev. Astron. Astrophys. **52**, 415 (2014).
[58] R. Essick *et al.*, Compact binary coalescence sensitivity estimates with injection campaigns during the LIGO-Virgo-KAGRA Collaborations' fourth observing run, Phys. Rev. D **112**, 102001 (2025).
[59] I. M. Romero-Shaw *et al.*, Bayesian inference for compact binary coalescences with Bilby: Validation and application to the first LIGO–Virgo gravitational-wave transient catalogue, Mon. Not. R. Astron. Soc. **499**, 3295 (2020).
[60] G. Ashton and C. Talbot, Bilby-MCMC: An MCMC sampler for gravitational-wave inference, Mon. Not. R. Astron. Soc. **507**, 2037 (2021).
[61] M. Menéndez, J. Pardo, L. Pardo, and M. Pardo, The Jensen-Shannon divergence, J. Franklin Inst. **334**, 307 (1997).
[62] ODYSSEE-MURE, Odyssee database: Energy efficiency indicators and consumption trends, https://www.odyssee-mure.eu/ (2023) (accessed: 2026-05-06).
[63] N. Aghanim *et al.*, Planck 2018 results: VI. Cosmological parameters, Astron. Astrophys. **641**, A6 (2020).