PAPER • **OPEN ACCESS**

The SXS collaboration's third catalog of binary black hole simulations

OPEN ACCESS
IOP Publishing
Classical and Quantum Gravity
Class. Quantum Grav. **42** (2025) 195017 (56pp)
https://doi.org/10.1088/1361-6382/adfd34

The SXS collaboration's third catalog of binary black hole simulations

Mark A Scheel[1,*], Michael Boyle[2], Keefe Mitman[2], Nils Deppe[2], Leo C Stein[3], Cristóbal Armaza[2], Marceline S Bonilla[4], Luisa T Buchman[5], Andrea Ceja[4], Himanshu Chaudhary[1], Yitian Chen[2], Maxence Corman[6], Károly Zoltán Csukás[7], C Melize Ferrus[8], Scott E Field[9], Matthew Giesler[2], Sarah Habib[1], François Hébert[1], Daniel A Hemberger[1], Dante A B Iozzo[2], Tousif Islam[9,10], Ken Z Jones[4], Aniket Khairnar[3], Lawrence E Kidder[2], Taylor Knapp[1], Prayush Kumar[11], Guillermo Lara[6], Oliver Long[6], Geoffrey Lovelace[1,4], Sizheng Ma[1], Denyz Melchor[4], Marlo Morales[4], Jordan Moxon[1], Peter James Nee[6], Kyle C Nelli[1], Eamonn O'Shea[2], Serguei Ossokine[6], Robert Owen[12], Harald P Pfeiffer[6], Isabella G Pretto[1], Teresita Ramirez-Aguilar[4], Antoni Ramos-Buades[13], Adhrit Ravichandran[9], Abhishek Ravishankar[9], Samuel Rodriguez[4], Hannes R Rüter[14], Jennifer Sanchez[4], Md Arif Shaikh[15], Dongze Sun[1], Béla Szilágyi[1], Daniel Tellez[4], Saul A Teukolsky[1,2], Sierra Thomas[4], William Throwe[2], Vijay Varma[9], Nils L Vu[1], Marissa Walker[4,16], Nikolas A Wittek[6] and Jooheon Yoo[2]

[1] Theoretical Astrophysics 350-17, California Institute of Technology, Pasadena, CA 91125, United States of America
[2] Cornell Center for Astrophysics and Planetary Science, Cornell University, Ithaca, NY 14853, United States of America
[3] Department of Physics and Astronomy, University of Mississippi, University, MS 38677, United States of America
[4] Nicholas and Lee Begovich Center for Gravitational-Wave Physics and Astronomy, California State University Fullerton, Fullerton, CA 92834, United States of America

[*] Author to whom any correspondence should be addressed.

[5] Department of Physics and Astronomy, Washington State University, Pullman, WA 99164, United States of America
[6] Max Planck Institute for Gravitational Physics (Albert Einstein Institute), Am Mühlenberg 1, 14476 Potsdam, Germany
[7] HUN-REN Wigner RCP, Konkoly Thege Miklós út 29-33, Budapest H-1121, Hungary
[8] Theoretical Particle Physics and Cosmology Group, Physics Department, King's College London, Strand, London WC2R 2LS, United Kingdom
[9] Department of Mathematics, Center for Scientific Computing and Data Science Research, University of Massachusetts, Dartmouth, MA 02747, United States of America
[10] Kavli Institute for Theoretical Physics, University of California Santa Barbara, Kohn Hall, Lagoon Rd, Santa Barbara, CA 93106, United States of America
[11] International Centre for Theoretical Sciences, Tata Institute of Fundamental Research, Bangalore 560089, India
[12] Department of Physics and Astronomy, Oberlin College, Oberlin, OH 44074, United States of America
[13] Departament de Física, Universitat de les Illes Balears, IAC3 – IEEC, Crta. Valldemossa km 7.5, E-07122 Palma, Spain
[14] CENTRA, Departamento de Física, Instituto Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal
[15] Department of Physics, Vivekananda Satavarshiki Mahavidyalaya (affiliated to Vidyasagar University), Manikpara 721513, West Bengal, India
[16] Christopher Newport University, Newport News, VA 23606, United States of America

E-mail: scheel@tapir.caltech.edu

Received 23 May 2025; revised 1 August 2025
Accepted for publication 19 August 2025
Published 6 October 2025

Abstract

We present a major update to the Simulating eXtreme Spacetimes (SXSs) Collaboration's catalog of binary black hole (BBH) simulations. Using highly efficient spectral methods implemented in the Spectral Einstein Code (SpEC), we have nearly doubled the total number of binary configurations from 2018 to 3756. The catalog now more densely covers the parameter space with precessing simulations up to mass ratio $q = 8$ and dimensionless spins up to $|\vec{\chi}| \leqslant 0.8$ with near-zero eccentricity. The catalog also includes some simulations at higher mass ratios with moderate spin and more than 250 eccentric simulations. We have also deprecated and rerun some simulations from our previous catalog (e.g. simulations run with a much older version of SpEC or that had anomalously high errors in the waveform). The median waveform difference (which is similar to the mismatch) between resolutions over the simulations in the catalog is 4×10^{-4}. The simulations have a median of 22 orbits, while the longest simulation has 148 orbits. We have corrected each waveform in the catalog to be in the binary's center-of-mass frame and exhibit gravitational-wave memory. We estimate the total CPU cost of all simulations in the catalog to be 480 000 000 core-hours. We find that using spectral methods for BBH

simulations is over 1000 times more efficient than previously published finite-difference simulations. The full catalog is publicly available through the `sxs` Python package and at https://data.black-holes.org .

Keywords: binaries, numerical relativity, gravitational waves, black hole, general relativity

Contents

1. Introduction

Since the discovery of gravitational waves (GWs) from binary black holes (BBHs) in 2015 [1–4], the laser interferometer GW observatory (LIGO) [5] and Virgo [6] have observed GWs from the inspiral, merger, and ringdown of dozens of BBHs [7–11]. Inferring the properties of the black holes (BHs) that emitted these waves has revealed a population of stellar-mass BBHs in a variety of different configurations. Understanding these black-hole properties, such as their masses and spins, constrains stellar evolution models and enables tests of general relativity.

Achieving this understanding requires comparison of the observed GW strain with highly accurate models of the observed BBHs and the GWs they emit. Because of the nonlinear nature of general relativity, analytic approximations like post-Newtonian [12–24] and post-Minkowskian [25, 26] break down near the merger. As a result, waveform models rely on numerical-relativity simulations to provide a ground truth to build on. Following the 2005 breakthroughs in numerical-relativity calculation of the inspiral, merger, and ringdown of two BHs [27–29], several research groups have used different codes to create catalogs of numerical gravitational waveforms for BBHs in a variety of configurations. These include the NINJA [30, 31], NRAR [32], MAYA [33, 34], RIT [35–38], NCSA [39], BAM [40], and GR-Athena++[41] catalogs.

Since numerical relativity simulations are both computationally expensive and performed for specific parameter values (e.g. mass ratio and spins), they are typically not used in data analysis directly (although parameter estimation can directly use numerical relativity waveforms [42]). Instead, GW model makers typically use numerical-relativity waveforms to calibrate, validate, and build their models; specific examples include calibrating effective-one-body (EOB)[43–50] models, validating phenomenological models [51–67], and enabling the construction of surrogate models [68–72] that directly interpolate between numerical-relativity waveforms across parameter space.

Future GW detectors, like cosmic explorer (CE) [73, 74], Einstein telescope (ET) [75–77], the laser interferometer space antenna (LISA) [78], TianQin [79], Taiji [80], DECIGO [81], and the lunar GW antenna (LGWA) [82], will detect GWs from BBHs much more often and with much higher precision. Next-generation detectors on Earth will require waveform models that are approximately one order of magnitude better than today's [83], while detectors in space pose an even greater challenge, possibly requiring several orders of magnitude increases in accuracy [84]. The challenge is compounded in that numerical waveforms must also be both longer and more accurate. Numerical relativity alone is too computationally expensive to cover the needed frequency bands, so waveform models will rely on hybridization procedures [59, 72, 85–91] that combine post-Newtonian and numerical relativity waveforms. And since future detectors will see drastically more events, they might see rarer BBHs, such as those with high eccentricity or extreme spins. Accurately modeling such configurations requires catalogs that better cover the extreme corners of the BBH parameter space.

In this paper, we present a major update to the Simulating eXtreme Spacetimes (SXSs) Catalog of BBH waveforms, accessible via a python package, `sxs` [92].The previous SXS catalog update [93] has been cited by over 292 publications, including work involving LVK data analysis, analytic waveform models, surrogate waveform models, beyond-GR studies, prospects for future detectors, and high-energy scattering studies [10, 37, 38, 50, 54, 62, 70, 83, 94–157]

The new catalog now has a total of 3756 simulations and more densely covers the parameter space, with precessing simulations up to mass ratio $q \leqslant 8$ and dimensionless spins up to $|\chi| \leqslant 0.8$ with near-zero eccentricity [158]. The catalog also includes some simulations at higher mass ratios with moderate spins, and it includes over 250 eccentric simulations. Figure 1 shows

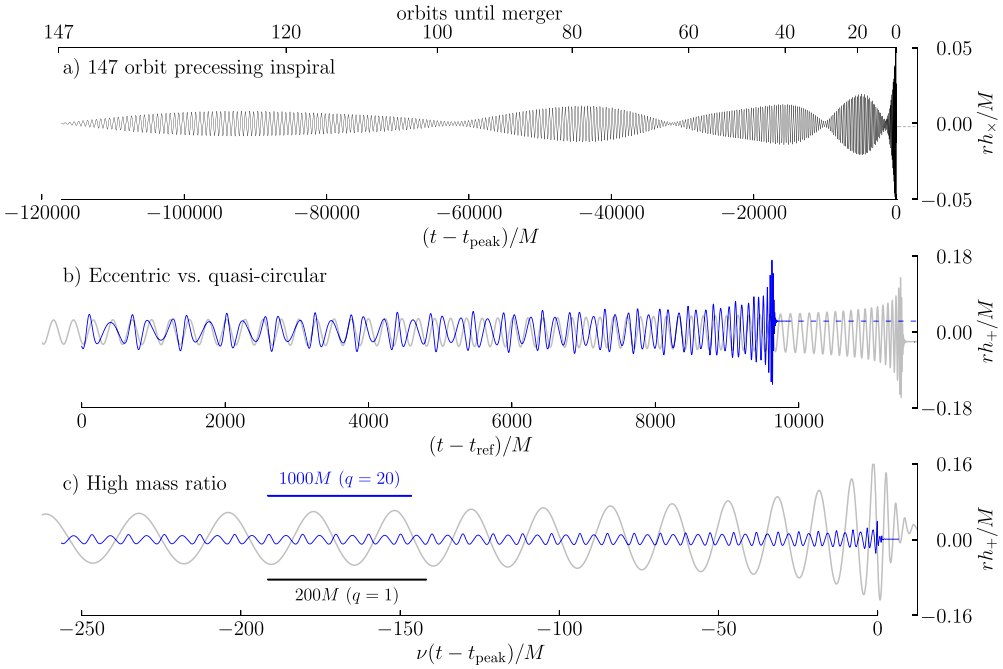

Figure 1. An overview of some of the most extreme systems in the updated catalog. (a) SXS:BBH:2621, a very long, 147-orbit simulation viewed from the emission direction $(\theta,\phi)=(\pi/2,\pi/2)$. For visibility, we extend the final steady-state value of the strain waveform as a dashed line to the right of $t-t_{\rm peak}=0$. This dashed line is nonzero because of gravitational-wave memory that is included in this updated catalog. (b) Blue: eccentric system SXS:BBH:2607 ($e_{\rm ref}\approx 0.31$), viewed from $(\theta,\phi)=(\pi/3,\pi/2)$. The faint gray trace is a circular system SXS:BBH:1153 with the same mass ratio ($q=1$), time-shifted to approximately agree in orbit-averaged frequency at $t_{\rm ref}$ of the eccentric waveform. Note the asymmetry, higher amplitude, and faster merger time of the eccentric system. Gravitational-wave memory is again indicated by the nonzero value of the strain after ringdown. (c) Blue: mass-ratio $q=20$ system SXS:BBH:2516, viewed from $(\theta,\phi)=(\pi/3,\pi/2)$. The faint gray trace is a $q=1$ reference system SXS:BBH:4434. The horizontal axis is scaled with the symmetric mass ratio ν so that the radiation-reaction timescale is the same horizontal distance on the plot for both waveforms. Note the smaller amplitude and the much longer inspiral time of the high mass-ratio system.

examples of some of the most extreme systems present in the updated catalog, including a simulation of a precessing system for over 147 orbits, an eccentric system with $e_{\rm ref}\approx 0.31$, and a system with mass ratio of 20.

The median waveform difference (which is similar to the mismatch, see section 2.4) between resolutions over all simulations is 4×10^{-4}, with a median of 22 orbits, while the longest simulation is 148 orbits. All the waveforms in the catalog are center-of-mass and GW-memory corrected [118, 159]. We have also deprecated and rerun simulations created with a much older version of SpEC and some simulations with anomalously high errors in the waveform.

We estimate the total CPU cost of all the simulations in the catalog to be only 480 000 000 core-hours. Using spectral methods for long, precessing BBH inspiral-merger-ringdown simulations is over 1000 times more efficient than using finite-difference methods for a few orbits of non-spinning BBHs at comparable accuracies; see, e.g. [41]. This performance gap is so large

that even GPU-based finite-difference codes have yet to prove competitive with CPU-based codes using spectral methods.

The rest of this paper is organized as follows. In section 2, we present an overview of the catalog, including our catalog's coverage in terms of parameter space and length, as well as estimates of the accuracy of the catalog's waveforms. In section 3, we summarize the methods that we use in the Spectral Einstein Code (SpEC), highlighting improvements since our 2019 catalog update [93]. In section 4, we discuss details of how we manage the catalog data. We briefly conclude in section 5. In the appendices, we document our waveform metadata format in appendix A, describe the algorithm for determining which simulations supersede deprecated ones in appendix B, document our current waveform format, including recent improvements, in appendix C, and list a few individual simulations with large errors in appendix D.

For those looking for a review on numerical relativity, see, e.g. [160–163].

2. Catalog overview

2.1. Available data

The catalog currently consists of 3756 simulations. Each simulation has a unique identifier of the form `SXS:BBH:1234`[17]. Each simulation typically includes multiple otherwise-identical runs at different spatial resolutions, which are denoted by `Lev`*N*. These resolution numbers do not necessarily have a consistent meaning across the catalog, but *for a given simulation* greater numbers represent higher resolution. See section 2.4 for how resolution numbers are defined for the newest simulations. The catalog currently contains 10 557 `Levs`, which gives an average of 2.8 per simulation.

In the catalog, we include metadata (see appendix A for details) for each resolution, which provides information like masses and spins, computed both for the initial data and at a *reference time* after initial transients have decayed away. We describe how we define the reference time in section 3.5.

We also provide apparent-horizon data, such as trajectories and spins as functions of time, for both the inspiral and ringdown in the same format as in earlier releases of the SXS catalog. See [93] for details. At time $t=0$, in the coordinates in which we measure trajectories and spins, the larger BH is on the positive x axis, the smaller BH is on the negative x axis, and the orbital angular momentum is in the positive z direction. However, at the reference time the BHs have moved from the positions they had at $t=0$, so, e.g. a reference-time spin in the x direction does not lie along the line segment separating the BHs.

We extract gravitational waveforms on a series of spheres surrounding the binary. We *directly* extract the strain h using Sarbach and Tiglio's formulation [164] of the Regge–Wheeler and Zerilli equations [165, 166], with implementation details described in [93, 167]. We separately extract the complex Weyl component Ψ_4, as explained in detail in [93]. We emphasize that these really are separate quantities; the extraction of h does not involve integrating Ψ_4; we sometimes compare two time derivatives of h versus Ψ_4 as one of our error estimates (e.g. figure 8). Note that sign conventions for quantities like h and Ψ_4 vary in the literature; see appendix C. of [93] for a detailed discussion of our sign conventions. We compute the quantities h and Ψ_4 on multiple coordinate spheres, typically 24 of them, that are spaced roughly uniformly in inverse radius and extend most of the way to the outer boundary. We then extrapolate

[17] These numbers are not necessarily consecutive. Also note that in this paper, we only discuss BBH systems; we will discuss black-hole–neutron-star (`BHNS`) and neutron-star–neutron-star (`NSNS`) systems in the SXS catalog in future work.

the waveforms at these finite-radius locations to future null infinity $\mathscr{I}^+$. See [93] for details of the extraction and extrapolation procedure, except we now use the `scri` package [168] for extrapolation. In addition to the center-of-mass correction [159, 169], we now also apply a memory correction described in section 3.6. We no longer supply finite-radius waveforms since they are contaminated by gauge and near-field effects.

All simulations in the catalog include initial transients, including a burst of non-astrophysical, high-frequency GWs commonly called *junk radiation*. Such initial transients appear in all numerical-relativity simulations of BBHs, because all known methods for constructing constraint-satisfying initial data do not yield a BBH in equilibrium emitting physically correct GWs. For a recent discussion of junk radiation, see section I of [170]. Unless specifically studying these transients, users of the catalog should remove them by discarding early times from all time-dependent catalog data. Section 3.5 discusses how we compute when junk radiation is no longer present.

There is no overall mass scale for the BBH problem, so each of our simulations can be scaled to any desired total mass. The units of strain waveforms in the catalog are rh/M as a function of u/M, where rh is the product of the areal radius and the strain evaluated at $\mathscr{I}^+$, u is the retarded time at $\mathscr{I}^+$, and M is the sum of the Christodoulou masses of the BHs at `reference_time` (see section 3.5 for definition of `reference_time`). The dimensionful metadata and horizon quantities are in code units, which are slightly different from the units of the waveforms. For example, the Christodoulou masses of the individual horizons at `reference_time`, which correspond to the quantities `reference_mass1` and `reference_mass2` in the metadata, do not sum exactly to unity (but the difference from unity is typically in the sixth digit). It is straightforward to rescale the metadata or horizons to the same units as the waveforms, or to any desired total mass.

All simulation data are available publicly [158], and we provide a Python package `sxs` [92] to simplify obtaining, managing, and analyzing the data, as described in section 4.

2.2. Deprecated simulations

Because of many improvements to `SpEC` over the years (see, e.g. section 3.1), newer simulations in the catalog are generally more accurate than older ones. Furthermore, we and others have identified problems with some older waveforms that were not evident until the numerical-relativity community began studying waveforms in greater detail, including higher-order modes, ringdown spectroscopy, GW memory, etc. Many of the largest problems occur in the 174 simulations from the first SXS catalog paper (published in 2013 [171]). For example, [172] identified—among others—waveforms with what they described as 'rippled-ringdown' and 'asymptotic-ringdown' anomalies, all of which came from this early group. We believe these are consistent with a lack of resolution during ringdown that we did not correct until April 2015, well after we produced the affected simulations. As a result, we have deprecated many early simulations. We have run newer simulations with the same physical parameters but with improved techniques and higher resolutions.

We and others have also occasionally found problems in a few newer simulations; we have deprecated these simulations as well. These problems were usually issues fixed by later updates to `SpEC`, but some involved missing or corrupted files due to filesystem problems or human error, noticed only after uploading the simulation to the catalog. We have recently put considerable effort into adding validation stages to our pipeline that postprocesses and archives simulations, so that we can identify more problems like these automatically as they occur. Some problematic simulations that we have deprecated or fixed have also been pointed out by [173–175]. One simulation, SXS:BBH:1131, had an error in the metadata caused by a bug

in SpEC's metadata writer that was present from September–November 2014. That simulation has been corrected in the current catalog, and remains non-deprecated.

We continue to include the deprecated simulations in the catalog, to enable interested users to study these problematic simulations. But the `sxs` package will produce an error when loading deprecated simulations, unless the user passes an option to ignore the deprecation. In both cases, the `sxs` package will suggest to the user a newer simulation with similar parameters, either through the error message or a warning if loading is forced. See appendix B for an overview of our algorithm for choosing the superseding simulation. Once a simulation has been deprecated, we no longer put effort into keeping that simulation up to date with the rest of the catalog. As a result, deprecated simulations may not have metadata fields consistent with new simulations or with other deprecated simulations. We encourage community members who make use of our simulations to report any concerns; if necessary we can deprecate old simulations and possibly rerun them if needed. To report a concern, open an issue at https://github.com/sxs-collaboration/sxs/issues/new?template=catalog-data-issue-template.md.

2.3. Parameter space

Since the release of the 2019 catalog [93], most of our effort for expanding the catalog has focused on higher mass ratios and spins. The BBH parameter space is quantified by the Christodoulou masses of the individual BHs m_1 and m_2 with mass ratio $q = m_1/m_2 \geqslant 1$, dimensionless spins $\vec{\chi}_1$ and $\vec{\chi}_2$, eccentricity e, and mean anomaly ℓ (see section 3.4). The dimensionless spins have magnitudes $|\vec{\chi}_i| \leqslant 1$; see section 2.2 of [93] for the definitions of masses and spins. Figure 2 shows mass ratios of all systems in the catalog, along with projections of the spins into the orbital plane

$$\chi_{1\perp} = \left|\vec{\chi}_1 \times \hat{L}\right| \qquad \chi_{2\perp} = \left|\vec{\chi}_2 \times \hat{L}\right|, \tag{1}$$

and the effective spin [53, 58, 59]

$$\chi_{\rm eff} \equiv \frac{(m_1\vec{\chi}_1 + m_2\vec{\chi}_2)\cdot\hat{L}}{m_1+m_2} = \frac{m_1\chi_{1\parallel} + m_2\chi_{2\parallel}}{m_1+m_2}, \tag{2}$$

where $\hat{L}$ is the direction of the instantaneous Newtonian orbital angular momentum. We extract all quantities at `reference_time`. The difference between `reference_time` and the time of merger varies widely between different simulations, both because the `reference_time` varies and more importantly because different simulations have different numbers of orbits. Therefore the points in the figure do not compare the parameters of all simulations at the same time relative to merger. Comparing simulation parameters at different times is most significant for spin directions in precessing cases. In particular, figure 2 shows that we now have dense coverage in mass ratios up to $q = 8$, including large spins and significant precession. We have also performed over 100 simulations between $q = 8$ and $q = 20$—most with essentially no spin on the smaller BH, but significant (usually precessing) spin on the larger BH.

We have continued to increase the number of cycles of our simulations, even for computationally challenging cases with high mass ratios and spins. A new subset of about 1000 precessing simulations with $q \leqslant 8$ has approximately 33 orbits per simulation, corresponding to ≈ 66 GW cycles. Figure 3 shows, for all simulations in the catalog, a histogram of the number of simulations binned by the number of GW cycles. Compared to the previous catalog, the

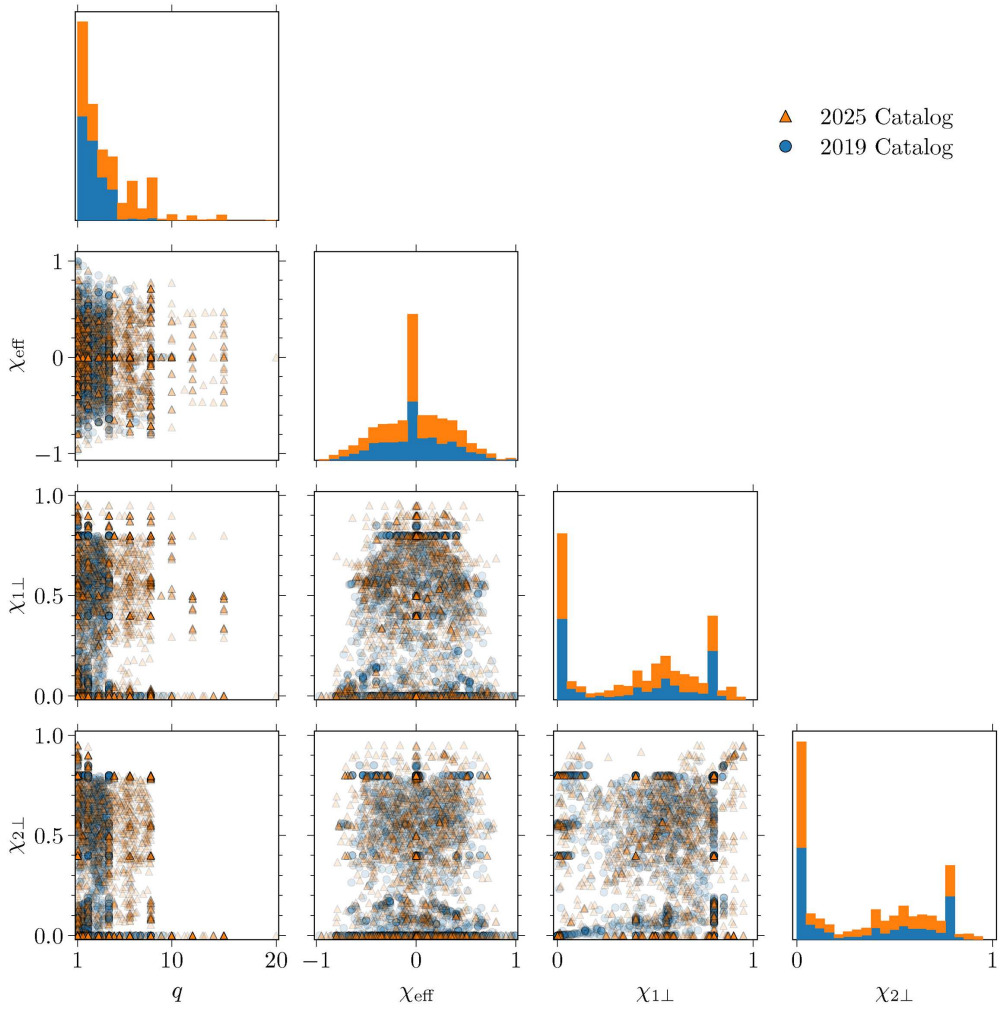

Figure 2. Distribution of reference mass ratios q and spins χ in the catalog. Each panel shows a projection of the 7-dimensional space. Each point is one simulation. We plot the effective spin $\chi_{\rm eff}$ (a combination of spins that has a strong effect on the phasing of the gravitational waves; defined in equation (2)) and the magnitudes of the spins in the orbital plane. Blue circles correspond to simulations that were released as part of the 2019 catalog, while orange triangles correspond to simulations new in this release. Darker regions are more densely covered. Deprecated simulations are omitted.

figure highlights an increase in the number of simulations that have 50–70 GW cycles[18]. There are also a number of new simulations with very few orbits that represent nearly-head-on collisions and scattering scenarios that are now in the catalog. We have produced a small number of simulations with even more cycles, but we have primarily focused on improved parameter-space coverage rather than on producing longer waveforms. This is for several reasons: first,

[18] We deprecated the longest simulation in the 2019 catalog, SXS:BBH:1110, because the waveform was contaminated with a center-of-mass acceleration as reported in [176].

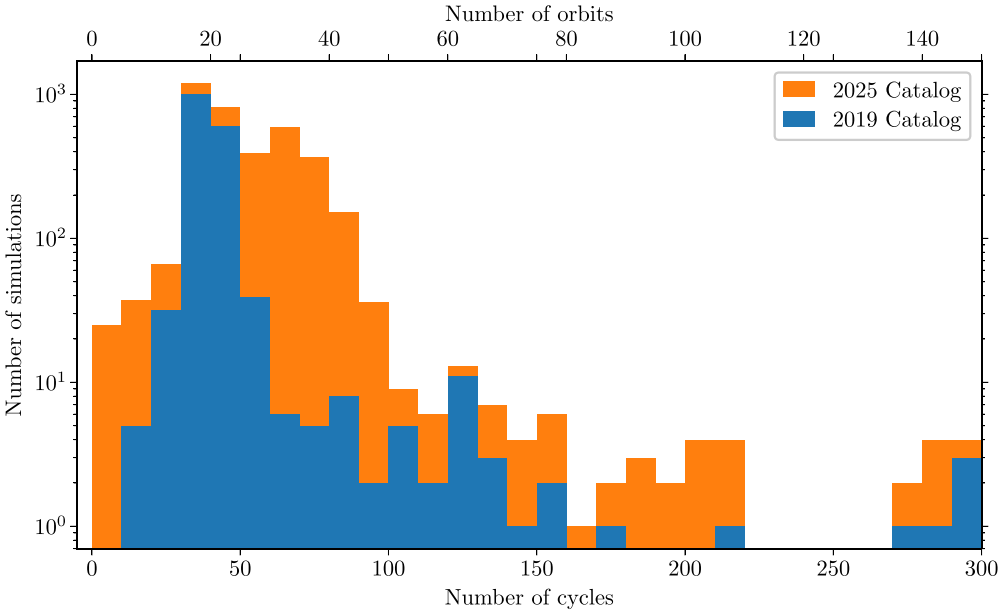

Figure 3. The number of orbits (top axis) and number of cycles (bottom axis) of the $\ell = m = 2$ GWs from the start of the simulation until the formation of a common apparent horizon for the simulations in the catalog, as determined by the coordinate trajectories of the black holes. Bin edges are multiples of five orbits and ten cycles. Deprecated simulations are omitted.

50–70 GW cycles is sufficient for most GW applications today, especially for systems with higher total masses ($\gtrsim 50M_\odot$) where LIGO-virgo-KAGRA (LVK) detectors are not sensitive enough at low frequencies to detect a longer inspiral. Second, for low-mass systems that can have many more GW cycles in band than would be feasible for us to simulate, alternative approaches (e.g. hybridizing numerical relativity and post-Newtonian models) can yield waveforms with sufficient length, although it remains unclear how long the numerical relativity waveforms must be for accurate hybridization, especially for precessing systems [91]. Finally, as demonstrated in [177] and discussed in section 2.4 below, the mismatch and waveform difference increase as $\sim t^2$, meaning longer simulations require significantly higher resolution to maintain reasonable phase errors. Nonetheless, in figure 4 we plot the reference dimensionless orbital angular frequency $M\Omega_{\rm orb}$ for our simulations to show what our low frequencies are. The top axis shows the dimensionful frequency of the $(2, 2)$ mode at the reference time for a system with a total mass of $50M_\odot$. Only simulations with reference eccentricities $< 10^{-2}$ are shown in figure 4, since for eccentric cases the instantaneous reference frequency $\Omega_{\rm orb}$ does not necessarily correspond to the lowest frequency in the waveform.

Figure 5 shows a histogram illustrating the number of simulations with different orbital eccentricities. In section 3.4, we describe the algorithm we use to measure the eccentricities, which is based on the BH trajectories, not the waveforms. Because the method uses trajectories, and because of the lack of a unique definition of eccentricity in general relativity (though efforts have been made to relate and understand different definitions, e.g. [100, 178, 179]), we recommend using the 'reference eccentricity' value in the simulation metadata, as computed according to section 3.4, only as a rough estimate. For more detailed analyses that depend on precise values of eccentricity, users should choose a definition of eccentricity and consistently

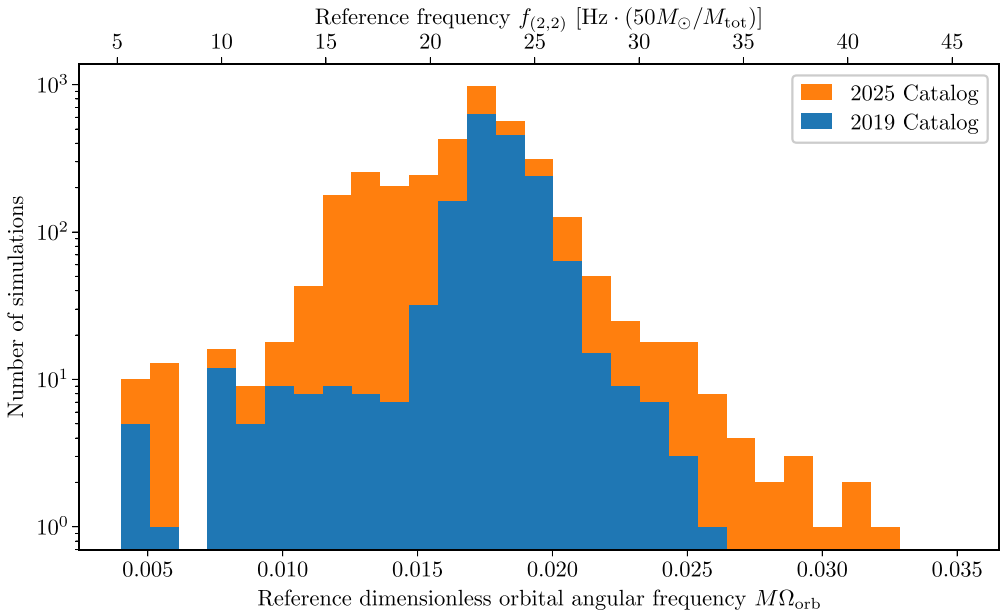

Figure 4. The reference dimensionless orbital angular frequency $M\Omega_{\rm orb}$ for the simulations in the catalog. The top axis is the frequency of the (2, 2) mode at the reference time for a binary with a total mass of $50M_\odot$. Only simulations with a reference eccentricity $< 10^{-2}$ are shown. Deprecated simulations are omitted.

measure it from the gravitational waveform. In general, we intend simulations with reference eccentricity below 10^{-3} in figure 5 to be quasi-circular; for these, we have used an iterative eccentricity-reduction scheme. In contrast, we intend simulations with reference eccentricity above 10^{-3} to represent eccentric systems. Figure 5 shows that the majority of waveforms in the catalog are quasi-circular, as in our 2019 catalog. However, we are continuing our efforts to extend the catalog to more eccentric and precessing systems [145, 180]. We will include more eccentric waveforms in future updates and releases of the catalog.

In summary, since 2019, we have nearly doubled the size of the catalog, with most new simulations having precessing spins with mass ratios $q > 4$. We also include a significant number of eccentric systems, as well as some simulations of hyperbolic encounters and scattering scenarios.

2.4. Waveform comparison

For measuring the accuracy of our waveforms, we compare the strain computed by (at least) two different resolutions for each simulation. For the newest simulations, we define different resolutions such that the target relative truncation error of the metric and its derivatives in the wavezone for `Lev`k is given by $2.17 \times 10^{-4} \times 4^{-k}$, and near the BHs it is approximately two orders of magnitude smaller. We set the projected constraint error (see [181]) tolerance to be four orders of magnitude smaller than the wavezone truncation error tolerance. Note that resolution numbers should be directly compared only for the same simulation. Resolution numbers for different simulations do not necessarily correspond to the same final errors, since different simulations (even newer ones) sometimes vary in choices of initial data or gauge (see section 3), as well as masses, spins, and number of orbits.

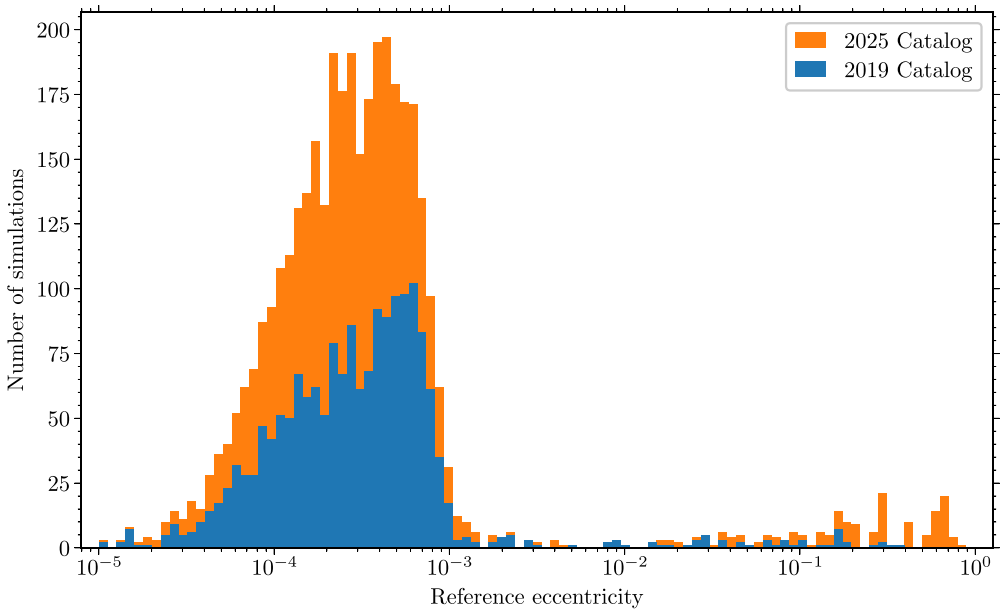

Figure 5. The number of simulations at different reference eccentricities $e_{\rm ref}$ in the catalog. The main population shows simulations using eccentricity reduction, while we have also completed several campaigns targeted at high $e_{\rm ref}$, yielding the tail. Deprecated simulations are omitted.

When presenting errors, we consider two quantities. The first is the waveform difference[19]

$$\Delta\left(h_1,h_2\right)=\frac{1}{2}\frac{\|h_1-h_2\|^2}{\sqrt{\|h_1\|^2\|h_2\|^2}}, \tag{3}$$

where

$$||h_1||=\sqrt{\int_{t_1}^{t_2}\int_{S^2}|h_1|^2\,\mathrm{d}\Omega\,\mathrm{d}t}. \tag{4}$$

The second is the waveform mismatch averaged over the two-sphere (which we hereafter refer to as the averaged mismatch)

$$\overline{\mathcal{M}}\left(h_1,h_2\right)=1-\frac{\langle h_1,h_2\rangle}{\sqrt{\langle h_1,h_1\rangle\langle h_2,h_2\rangle}}, \tag{5}$$

where the inner product $\langle h_1,h_2\rangle$ is

$$\langle h_1,h_2\rangle=\mathrm{Re}\left[\int_{f_1}^{\infty}\int_{S^2}\frac{\tilde{h}_1\left(f\right)\tilde{h}_2\left(f\right)^*}{S_n\left(f\right)}\,\mathrm{d}\Omega\,\mathrm{d}f\right]. \tag{6}$$

In equation (4), t_1 is the relaxation time defined in section 3.5, and $t_2=t_{\rm merger}+0.6(t_{\rm end}-t_{\rm merger})$, where $t_{\rm merger}$ is the time at which the strain's L^2 norm over the two-sphere achieves

[19] We choose this form so that the difference and mismatch agree in the limit of a flat power spectral density and infinite numerical resolution.

its maximum value and $t_{\rm end}$ is the end time of the simulation. For those simulations without mergers (e.g. hyperbolic encounters), we choose $t_2 = t_{\rm end} - 100M$. In equation (6), $\tilde{h}_1(f)$ is the Fourier transform of $h_1(t)$, * represents complex conjugation, f_1 is twice the orbital frequency at the relaxation time, and $S_n(f)$ is the power spectral density (PSD) of a GW detector. Note that $\overline{\mathcal{M}}$ differs from the mismatch used in [93] and in typical LVK analyses, which instead compute mismatches at various points on the two-sphere, minimizing the mismatch over time and phase shifts at each point on the sky independently, and then report the largest mismatch. The averaged mismatch in equation (5), however, corresponds to an average mismatch over the whole two-sphere since no point evaluation is performed. We use this new quantity because it better quantifies our waveform differences by including correlations between different points on the two-sphere that are important for getting spherical-harmonic modes of the waveform correct for modeling purposes. All of the code to compute mismatches and reproduce the subsequent analysis is available via the `sxs.simulations.analyze.analyze_simulation` function.

It is customary to consider two waveforms to be equal if they differ only by an overall phase or time shift, since the two waveforms represent the same physics. Similarly, two waveforms should also be considered equal if they differ only by an $SO(3)$ rotation, Lorentz boost, or supertranslation. There are a few ways in which one can incorporate these frame freedoms into a comparison between waveforms at different resolutions. First, one could fix the frame freedom of the waveform at each resolution independently, e.g. make $t = 0$ correspond to the time at which the strain's L^2 norm achieves its maximum value and fix the other Bondi-van der Burg–Metzner–Sachs (BMS) freedoms in a well-prescribed way (for a review of the BMS group in relation to numerical relativity simulations, see, e.g. [169, 182]); we call this option the 'independent alignment' method. Second, one could find the optimal BMS transformation that makes the waveform from one resolution best agree with the waveform from the other resolution; we call this the 'minimal difference' method since the optimization yields the smallest difference.

For the majority of our convergence tests, we use the independent alignment method, as it fixes the coordinate freedom of each resolution independently and will still converge in the limit of infinite resolution. This method is appropriate for waveform models that claim to be in some well-defined coordinate frame, because failures of the model to accurately fix that frame will translate into differences between waveforms. However, because no optimization over the coordinate transformations is performed, the difference produced by the independent alignment method will be larger than the difference resulting from methods that do optimize over these transformations. Thus, for illustrative purposes, for some cases we will compare with the minimal difference method, for which we perform a four-dimensional optimization over a time translation and $SO(3)$ rotation to minimize the waveform difference (see equation (3)). Note that we do not optimize over Lorentz boosts and supertranslations, since these optimizations can be very expensive and tend to be less important than optimizing the time and rotation freedoms, which are more directly related to the phase error of the simulation. Nonetheless, methods for efficiently optimizing over all of the BMS freedoms are in development and will be performed for future catalog analyses.

For the independent alignment method, we fix the frame freedom as follows. First, we perform a time translation so that $t = 0$ of each waveform corresponds to the merger, which here we define as the time at which the strain's L^2 norm over the two-sphere achieves its maximum value. Then, we find the rotation that aligns the angular velocity with the positive z axis [183], which makes the phase of the $(2, 2)$ mode zero and the real part of the $(2, 1)$ mode positive. We do this at the time that is 10% the length of inspiral (not including the times before

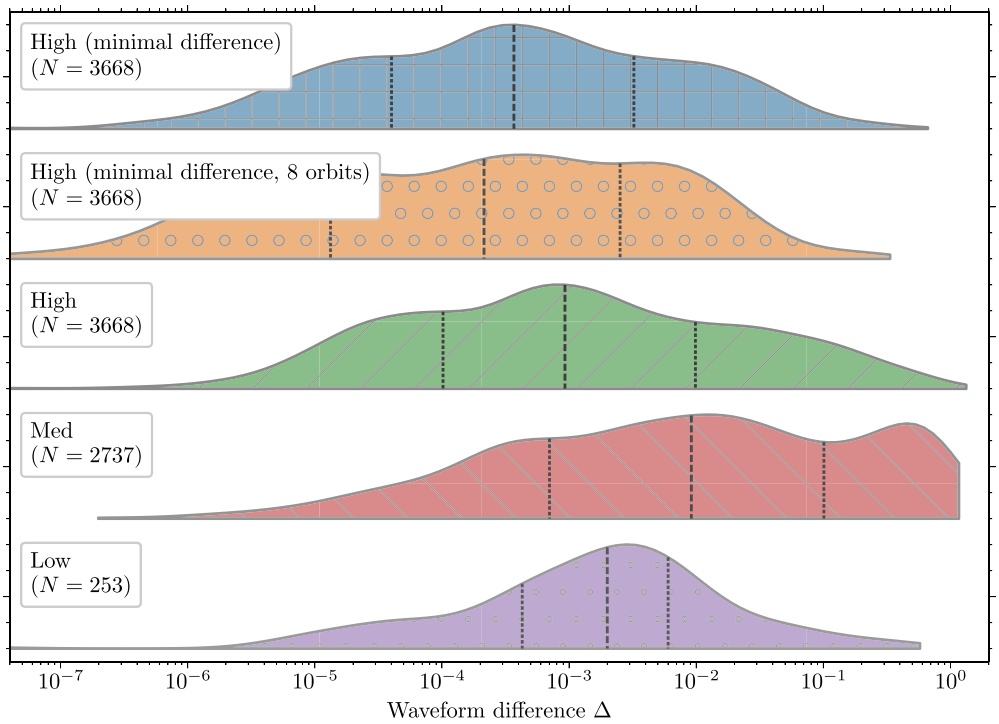

Figure 6. Waveform difference (see equation (3)) between strain waveforms for different simulation resolutions. Blue corresponds to the difference between the two highest resolutions available, using the minimal difference alignment method. Orange is identical to blue, but with the waveform truncated so that it starts 8 orbits before merger. Green is identical to blue, but with the independent alignment method. Red and purple are identical to green, but for the next two highest pairs of resolutions. Vertical lines show the quartiles of each distribution. The number of simulations in each distribution is shown in the legend. While some simulations have differences as low as 10^{-10}, we truncate the horizontal axis to make the figure more readable.

the relaxation time) before $t = 0$. For example, if the post-relaxation time inspiral is $5000M$ long, then we fix the frame at $-500M$.

In figure 6 we show the waveform difference between the resolutions available for each simulation. All waveform differences and mismatches used to produce figures 6–9 are available in the supplementary material [184]. Note that we have a handful of simulations with a single resolution, which do not contribute to this figure. The blue (green) curve in figure 6 shows the difference between the two highest resolutions when using the minimal difference (independent alignment) method, while the red and purple curves show the differences between the next highest resolutions with the independent alignment method. The vertical dashed lines show the quartiles of each distribution. As can be seen by comparing the blue and green curves, the minimal difference method improves the median difference by a factor of two and lessens the high-difference tail, which results from some of the more precessing systems requiring a much finer tuning of the rotation transformation. The reason why the purple distribution appears to have smaller differences than the red distribution is simply because there are fewer data points for the purple distribution, since we have fewer simulations that have four separate resolutions. Also, those simulations with four separate resolutions tend to be in less extreme

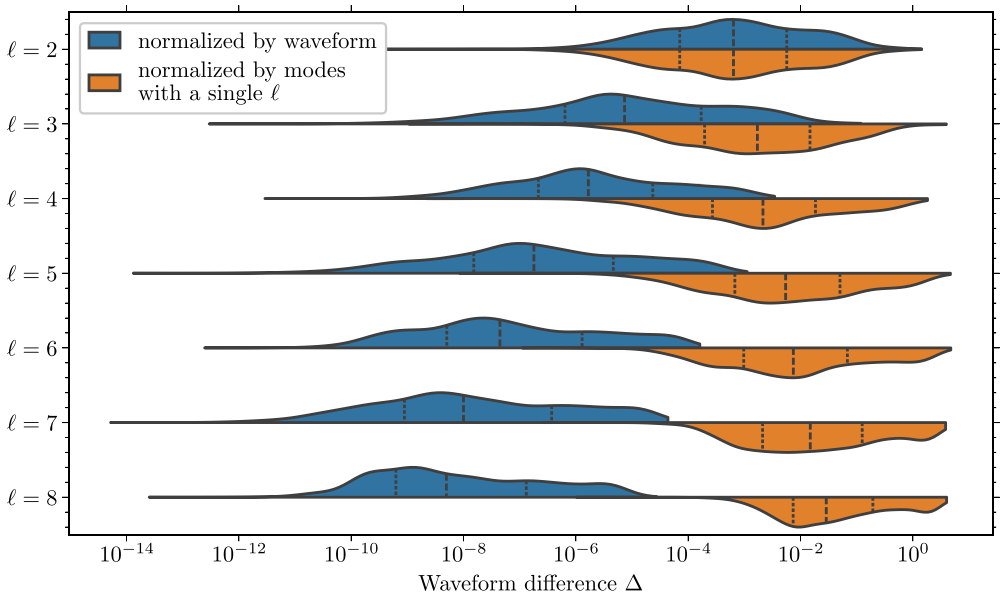

Figure 7. Waveform difference (per ℓ) (see equation (3)) between strain waveforms for the two highest-resolution simulations. In blue we show waveform differences normalized by the full waveform, while in orange we show differences normalized by only the ℓ' harmonics with $\ell' = \ell$. The waveforms used for these errors have been aligned using the minimal difference method. Vertical lines show the quartiles of each distribution.

regions of parameter space, where computational cost is lower and where it is easier to obtain higher accuracy. This figure shows that the relative error in our waveforms over the entire catalog, on average, is $\mathcal{O}(1\%)$. In terms of mismatches (see also figure 9), this median waveform difference corresponds to a averaged mismatch of $\mathcal{O}(10^{-4})$.

The differences shown in figure 6, as well as in figures 7 and 9 below, are taken between simulations at the two highest resolutions, and thus should be dominated by the numerical truncation error of the second-highest resolution simulation. The differences shown in the figures therefore represent an upper bound on the numerical truncation error of the highest-resolution simulation, so we expect the accuracy of the highest-resolution simulations to be better than what is shown.

An important note regarding these waveform differences, however, is that they are highly dependent on the length of the simulation. In particular, in [177] it was shown that the mismatch (and therefore also the waveform difference) between numerical relativity waveforms from different resolutions, whose relative error is dominated by some phase difference $\delta\phi$, tends to scale as

$$\overline{\mathcal{M}}\left(h, \mathrm{e}^{\mathrm{i}\delta\phi t}h\right) \sim \delta\phi^2 |t_2 - t_1|^2. \tag{7}$$

Therefore, while the differences in the blue curve figure 6 may seem high compared to other catalogs', this is mainly because our simulations tend to be very long (see figure 3 for an overview of our lengths). In particular, any numerical relativity waveform dominated by phase error can be made to have a smaller difference simply by shortening it by removing the beginning of the waveform. We demonstrate this with the orange curve, for which we truncate our

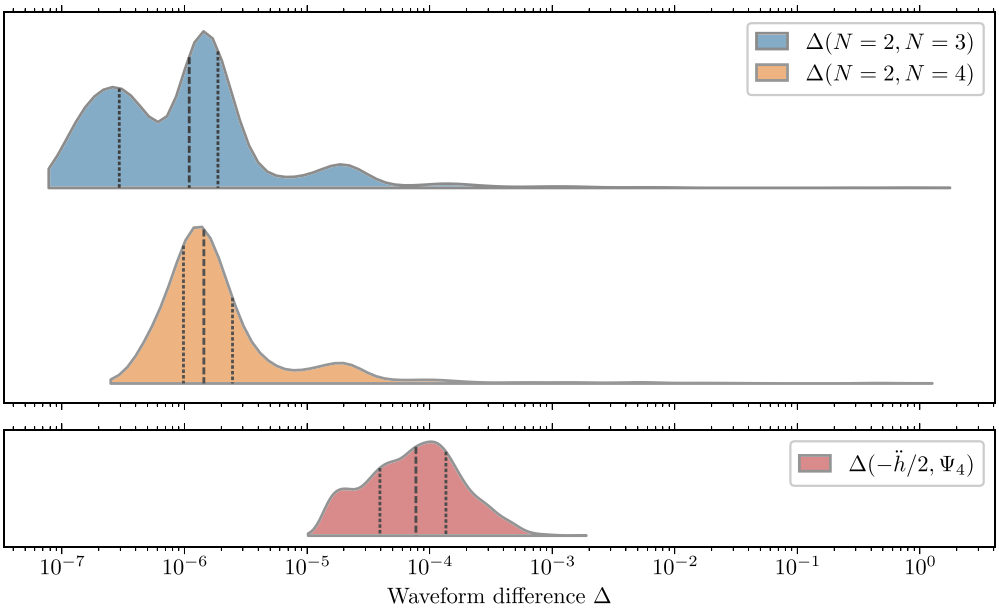

Figure 8. Top panel: waveform difference (see equation (3)) between different extrapolation orders. Bottom panel: difference between Ψ_4 and $-\ddot{h}/2$, which are computed independently by different methods but which should agree. Differences are taken for the highest resolution for each simulation. Waveforms are aligned using the independent alignment method. Vertical lines show the quartiles of each distribution.

simulations to be 8 orbits before computing waveform differences using the minimal difference method. In the most extreme case, by truncating one of our simulations to only be 8 orbits, we can improve the waveform difference between resolutions by four orders of magnitude. The remaining high tail in the orange curve is due to a few highly precessing systems that are underresolved, so that the two resolutions end up describing systems that are slightly physically different (e.g. different spin directions).

For building waveform models, the errors in individual spherical-harmonic modes of the strain are often of interest. In figure 7 we show the waveform difference between the two highest resolutions when restricted to spherical-harmonic modes with a certain ℓ. In blue we show the difference normalized by the norm of the entire waveform while in orange we show the difference normalized by the norm of only the modes with that particular value of ℓ. All of the errors are calculated from waveforms aligned with the minimal difference method. As may be expected because of their higher frequency and more complicated sourcing, the larger ℓ modes exhibit larger errors relative to their amplitudes. However, the larger ℓ modes contribute much less to the overall strain, so the errors in these modes have a small contribution to the overall waveform. Comparing figure 7 to the blue distribution in figure 6 makes it apparent that our dominant source of error in the overall waveform is still from the $\ell = 2$ modes.

The procedure [93] used to extrapolate a series of finite-radius waveforms to produce a waveform at $\mathscr{I}^+$ uses a polynomial fit of order N to extrapolate in $1/r$. Varying this order N can be used to quantify the error of the extrapolation procedure. When downloading waveforms from the catalog, the `sxs` package chooses $N = 2$ by default, but can also download $N = 3$ or $N = 4$ if requested. In figure 8, we show the waveform difference between different extrapolation orders for the highest resolution. The blue curve shows $N = 2$ vs. $N = 3$ and the

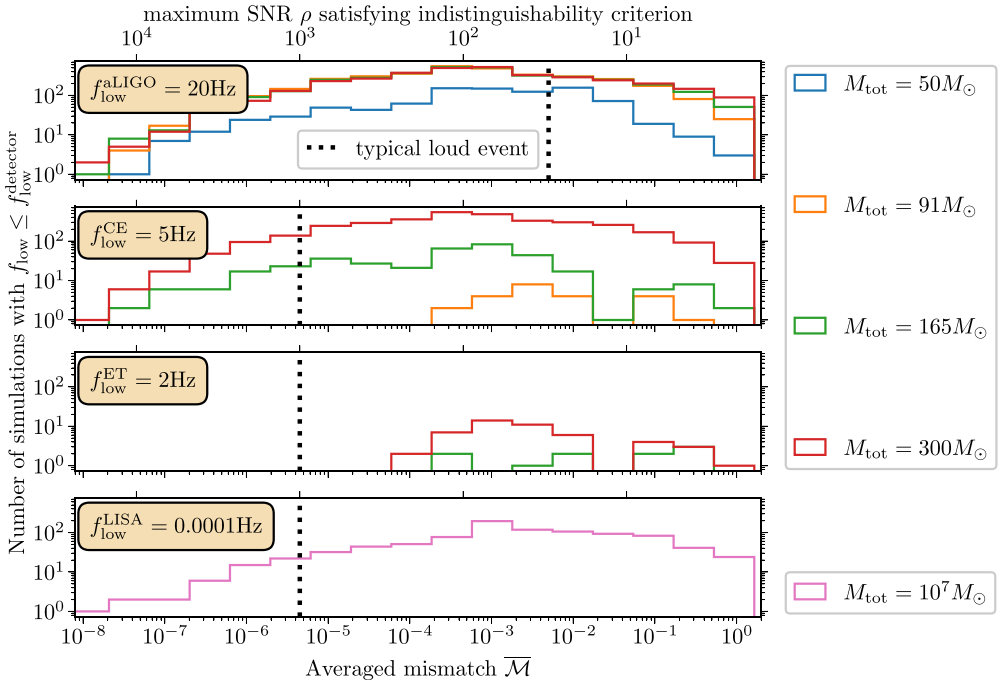

Figure 9. Number of simulations with $f_{\rm low} \leqslant f_{\rm low}^{\rm detector}$ vs. averaged mismatch (equation (5)) between the strain waveforms for the two highest resolutions of that simulation. Here $f_{\rm low}$ is twice the orbital frequency at the simulation's relaxation time and $f_{\rm low}^{\rm detector}$ is the low-frequency cutoff of each detector. The top horizontal axis represents the signal-to-noise ratio (SNR) at which the corresponding mismatch will not bias data analyses, according to equation (8). All simulations to the left of a specific mismatch have truncation error sufficiently small to not bias data analyses of signals with SNR smaller than ρ on the top axis. Different colors correspond to different total masses, logarithmically spaced from $50M_\odot$ to $300M_\odot$ for ground-based detectors. The vertical dotted lines indicate the SNR of typical loud events observed/expected in each detector [10, 74, 77, 84]. Only simulations with reference eccentricities $< 10^{-2}$ are shown.

orange curve shows $N = 2$ vs. $N = 4$. Waveforms are aligned using the independent alignment method in this figure. When comparing orders $N = 2$ and $N = 3$, the differences peak near $\mathcal{O}(10^{-6})$ with a tail that extends to $\sim$1. Overall this result demonstrates that by this measure, errors in our extrapolation procedure are on average unimportant when compared to the numerical truncation error. That is, the extrapolation errors in Figure 8 are typically an order of magnitude less than the errors shown in Figure 6 and are therefore not our dominant source of waveform error. The high-Δ tail is due to a few nearly-head-on or scattering simulations, which we list in appendix D. One reason for the large waveform differences is junk radiation, which causes large differences between extrapolation orders. The actual amount of junk radiation is similar to that in inspiral waveforms, but in these short simulations we do not have the luxury of waiting longer for junk to decay before the interesting physics occurs, so the relative contribution of the initial transient junk radiation to the waveform difference is larger. Another reason for the large waveform difference is that some of these runs have waveform-extraction

radii that are very close together resulting in an inaccurate extrapolation. Simulations with narrow distributions of extraction radii will be deprecated and rerun in the future.

In the bottom panel of figure 8, we also show the difference between $-\frac{1}{2}\ddot{h}$ and Ψ_4, two quantities that we compute independently in our code by completely different methods, but which should agree. The agreement between $-\frac{1}{2}\ddot{h}$ and Ψ_4 shows how well our simulations respect the Bianchi identity for Ψ_4 on future null infinity $\mathscr{I}^+$.

Finally, we assess the readiness of our catalog for next-generation detectors, specifically, LIGO Design, CE, ET, and LISA. In figure 9, we show the number of simulations in the catalog whose lowest frequency—taken to be twice the reference orbital frequency—is below the low-frequency cutoff of each detector. Only simulations with reference eccentricities $< 10^{-2}$ are shown in figure 9, since for eccentric cases the instantaneous reference frequency does not necessarily correspond to the lowest frequency in the waveform. On the bottom horizontal axis of figure 9 we plot the averaged mismatch (see equation (5)) between the two highest resolutions when using the PSDs for the four detectors, while on the top horizontal axis we plot the signal-to-noise ratio (SNR) at which the corresponding mismatch is sufficient to not bias data analyses. Specifically, we use the well-known sufficient condition for two waveform models to be indistinguishable [185–190],

$$\overline{\mathcal{M}} < \frac{D}{2\rho^2}, \tag{8}$$

where $\overline{\mathcal{M}}$ is the averaged mismatch (see equation (5)), D is the number of intrinsic parameters with $D = 9$, and ρ is the SNR of the observation the models are describing. Note that equation (8) is a sufficient, but not necessary, condition; i.e. there may be some situations where an averaged mismatch larger than the right hand side of equation (8) will still not lead to large biases in data analyses [190]. In figure 9, different colors correspond to different total masses and vertical dotted lines show SNRs of typical loud events expected for each detector. The PSDs for the four detectors come from the following: For aLIGO, we use the PSD presented in [191] and published in [192]; for CE, we use the PSD presented in [193, 194] and published in [195]; for ET, we use the ET-D PSD presented in [77] and published in [196]; and for LISA, we use the PSD presented in [197] and published in [198].

To interpret the figure, choose a SNR for a given detector, and choose a total mass. Then all simulations to the left of that SNR in the figure have enough cycles to cover the frequency band of the detector, and sufficiently small truncation error so as to not bias data analysis of a signal with that SNR, or with any smaller SNR. For example, for a CE event with SNR $\sim$200, the corresponding mismatch is about 2×10^{-4}, there are hundreds of simulations in the catalog sufficiently long and accurate for a total mass of $165M_\odot$ and larger, but for that SNR there are no simulations in the catalog sufficient for a total mass of only $91M_\odot$.

The figure shows that, for LIGO, which has a much higher low-frequency cutoff, the majority of our catalog has a sufficiently small averaged mismatch for events with $M_{\rm tot} > 90M_\odot$. However, for lower total masses or for future detectors with lower low-frequency cutoffs, many of our simulations are too short to span the entire frequency band, and most of the simulations that do span the entire frequency band may have mismatches too large to not bias the data analysis. In particular, for ET, which has a low-frequency cut off 2 Hz, we find that we have very few simulations long enough to span the frequency band for $M_{\rm tot} < 300M_\odot$, while for LISA we have no simulations long enough for to span the frequency band for $M_{\rm tot} < 10^6 M_\odot$. While hybridization with post-Newtonian waveforms may reduce the need for extremely long numerical relativity waveforms, the exact requirements for hybridization at next-generation accuracies have yet to be fully determined [59, 72, 85–91]. Understanding this is crucial to

prepare the community for the large number of $M_{\rm tot} < 10^6 M_\odot$ binaries that are expected to be observed by LISA [199].

3. Numerical methods

All the simulations in the catalog are performed using the SpEC [200]. We construct constraint-satisfying initial data using the extended conformal thin sandwich (XCTS) [201, 202] formulation. In the XCTS formulation, the conformal three-metric and the trace of the extrinsic curvature, as well as their time derivatives, can be freely chosen. Typically we choose these quantities to be weighted superpositions of the analytic solutions for two single BHs in Kerr–Schild coordinates [203], in time-independent horizon-penetrating harmonic coordinates [204, 205], in time-independent damped harmonic coordinates [206], or in a spherical version of Kerr–Schild coordinates [207]; each of these choices has its own advantages and drawbacks [207, 208]. A few simulations in the catalog are conformally flat and have a vanishing trace of the extrinsic curvature (i.e. maximal slicing) [209]. We choose the time derivatives of the conformal metric and trace of extrinsic curvature to be zero. We then solve the XCTS equations on a grid with two excised regions using a spectral elliptic solver [210]. The boundary conditions on the excision boundaries are chosen to ensure that these boundaries are either apparent horizons (AHs) [203, 209], or that they are surfaces of constant expansion with a small negative value [208] and thus are slightly inside AHs. The outer boundary is at a radius around $\sim 10^9 M$, and the boundary conditions are derived by requiring asymptotic flatness. The solution of the XCTS equations provides constraint-satisfying initial data on the initial slice.

The initial data depend on a set of input parameters, such as the spin vectors and masses associated with the analytic single-black-hole solutions used to build the conformal metric, and also the initial coordinate positions and velocities of the BHs. After the constraints are solved, we compute physical parameters such as the spin vectors and masses of the BHs, and frame quantities such as the initial total linear momentum and center-of-mass of the binary. These computed physical and frame quantities are not the same as the input parameters, and cannot be computed from the input parameters without solving the nonlinear XCTS equations. After measuring the physical and frame quantities, we adjust the input parameters and re-solve the XCTS equations. This process is iterated until the BHs have the desired masses and spins, the initial linear momentum is zero, and the initial center-of-mass is at the origin [211, 212].

For most of the simulations in the SXS catalog, we carry out an additional iteration that briefly (typically for a few orbits) evolves the initial data, and adjusts the initial coordinate velocities to yield a BBH with small orbital eccentricity [213–216], typically $e_0 \sim 10^{-3} - 10^{-4}$. For some simulations in the catalog, we intentionally omit the eccentricity-reduction iteration, to obtain initial data for BBHs with non-negligible orbital eccentricity. For other simulations we include an eccentricity iteration to tune the eccentricity to desired nonzero values [217, 218]. See section 3.4 for details of eccentricity measurement.

For the evolution, we use a first-order version of the generalized harmonic formulation [219–222] of Einstein's equations [221–223]. For most simulations we choose an initial gauge that approximates a time-independent solution in a co-rotating frame, and then we smoothly change to damped harmonic gauge [224–226], which works well near merger. A few simulations start in harmonic gauge at $t = 0$ and smoothly change to damped harmonic gauge. Some simulations start in damped harmonic gauge and remain in that gauge for the full evolution [208].

We evolve Einstein's equations using a multidomain spectral method [222, 226–229], with a method-of-lines timestepper that uses a fifth-order Dormand–Prince integrator and a

proportional-integral adaptive timestepping control system [230]. The computational domain extends from excision boundaries, located just inside the AHs and conforming to their shapes [226, 228, 229, 231], to an artificial outer boundary. No boundary conditions are needed or imposed on the excision boundaries. On the outer boundary we impose constraint-preserving boundary conditions [222, 232, 233] on most of the fields, we impose an approximate no-incoming-wave condition on two physical degrees of freedom by freezing the Newman–Penrose Ψ_0 at the boundary [222], and we impose a Sommerfeld condition on the four remaining (gauge) degrees of freedom [222]. The outer boundary is typically placed at a radius of order $1000M$, but this varies for different simulations. For extremely long simulations, the outer boundary is automatically placed farther outward to avoid the center-of-mass gauge effect reported in [176]. We also impose boundary conditions on the incoming characteristic fields at each inter-domain boundary using an upwind penalty method [234, 235]. After a common AH forms, the simulation automatically stops, interpolates onto a new grid with a single excision boundary inside the new common horizon [228, 229], and continues evolving until after ringdown.

Spectral methods are exponentially convergent, meaning that spatial truncation errors decrease exponentially with the number of grid points in a particular subdomain, if the size and shape of the subdomain remain fixed. However, our spectral adaptive mesh refinement (AMR) procedure [181, 236] dynamically changes the size, shape, and the number of subdomains during the simulation. In addition, we choose the resolution not by choosing a number of grid points, but by specifying an error tolerance that governs when grid points should be added or subtracted in a given subdomain (p-refinement), and when different subdomains should be split and joined (h-refinement). Because of this complicated AMR procedure, we should not expect strict convergence as a function of the AMR tolerance parameter. To see why, consider two otherwise-identical simulations with different AMR tolerances. These simulations may happen to have the same number of grid points in a particular subdomain at a particular time, because their local truncation errors happen to be within the appropriate thresholds. In this case they will not exhibit strict convergence. Similarly, strict convergence will be lost if these two simulations happen to have entirely different subdomain boundaries (because of h-refinement) at the same time. Strict convergence can also be lost because our adaptive control system that adjusts the size and shape of the excision boundaries [229] involves thresholds. One way that this can happen is if two otherwise-identical simulations with different AMR tolerances end up with excision boundaries in slightly different locations. Finally, decisions by both AMR and by the control system exhibit hysteresis. Despite these issues, most of our simulations do exhibit convergence with AMR tolerance, as shown in section 2.4.

3.1. Recent improvements to SpEC

Since the time of the previous catalog paper [93], there have been numerous improvements to SpEC, with approximately 2000 new commits in the git history. SpEC has a total of about 700k lines of code, excluding blank lines and comments. While some of these recent improvements have been rather mundane (like upgrading to C++17 or providing support for additional compute clusters), many of the changes have targeted performance, robustness, and new features.

All changes in SpEC are submitted as GitHub pull requests. These undergo code review, where each pull request is reviewed by a senior developer considering correctness, clarity, documentation, and efficiency. Before merging to the main branch, a pull request must pass a test suite (currently 422 tests) run on different compilers and different build options. Many of these tests are small unit tests: for example, checking that transforming points through a complicated coordinate map and then transforming back recovers the original points. Some tests are

more comprehensive: for example, a checkpoint/restart test runs a BBH for four timesteps, and compares the result to another BBH run for two timesteps, is checkpointed, restarted, and run for two more timesteps; the results must be bitwise identical. There are also many regression tests that demand, for example, that a few timesteps of a BBH give bitwise-identical results when run before the pull request compared to when run after the pull request; if the pull request is expected to change the output of a regression test, then as part of the pull request the user flags that test and writes a brief explanation of why the output of that test should change.

SpEC includes code that does not legally allow us to distribute it as open source. A newer code SpECTRE [237], under development, is open source, and will eventually include all relevant features of SpEC plus a large number of improvements, especially in scalability.

Many of the performance improvements have focused on reducing SpEC's memory usage by eliminating a large number of temporary variables. Given that memory bandwidth is the limiting factor on today's HPC systems, we have also worked on eliminating memory allocations and data copying. Other improvements sped up computations that were prohibitively slow for certain edge cases. An example of this kind is the code that computes the spin of an AH using the method of [238]. This method requires solving a generalized eigenvalue problem involving matrices of size $\sim L^2 \times L^2$, where L is the maximum ℓ retained in the spherical-harmonic $Y_{\ell,m}$ expansion of the horizon surface. Formerly, all eigenvalues and eigenvectors were found directly, but this was prohibitively slow for simulations with large mass ratios where L of a highly distorted horizon could be 80 or more. This is now done with an iterative method that finds only the eigenvectors with the three smallest eigenvalues, which are the ones needed to compute the spin of the AH.

Robustness improvements were mostly driven by the need to simulate BBHs with larger mass ratios and spins. At the time of [93], most simulations had mass ratio less than 4, with higher mass ratio simulations being expensive and not infrequently failing. Mass ratios up to about 8 are now straightforward even with spin magnitudes of 0.8 and precession. Some of the improvements include better eccentricity measurement [216, 217] that handles cases where our previous algorithm failed to robustly measure the eccentricity of the system, especially if the eccentricity was small.

A change that helps both efficiency and robustness is improvements in the coordinate maps that connect the grid frame to the inertial frame. In particular, several coordinate maps described in [229] were inverted numerically because they involved functions like Gaussians. The numerical inversion occasionally failed for edge cases (e.g. unlucky machine roundoff caused the wrong root to be found), and more of these failures happened for high mass ratios, eccentric systems, and hyperbolic encounters, when the map parameters became more extreme. Although these failures were rare, we often run hundreds of simulations at once, so diagnosing and fixing each new edge case took considerable human time. The solution was to replace these maps with new maps that use piecewise linear functions with kinks at subdomain boundaries so that the maps are quicker to compute, more accurate, and can be inverted analytically. Restricting the kinks to subdomain boundaries retains spectral convergence, since the map inside each subdomain remains smooth. Additionally, several bugs and inaccuracies in the time-dependent coordinate maps were fixed. These issues do not appear to impact the waveforms.

A failure mode that rarely but regularly appeared either at the beginning of the simulation (for high spins) or at the beginning of ringdown was caused by a mismatch between the shape of the excision boundaries and the shape of the horizons. The origin of this problem is the use of different grids before and after the transition, either from initial data to inspiral or inspiral to ringdown. This mismatch sometimes prevented the control system from locking and the simulation failed after a few timesteps. Now, if such a failure occurs, the horizon shape is

measured during the first few timesteps and then the simulation is automatically restarted with new excision boundaries that better match the horizon shapes. This process is repeated several times if necessary, typically up to three.

There have also been many improvements to the initial data solver since [212], primarily to make the solver more robust for very large initial distances between the two BHs. Among the changes are (i) a radial split of the inner spherical shells surrounding either excision boundary, if the radial coverage is too large; (ii) streamlining the criteria used to decide when to increase the AMR resolution and how to measure the error in root-finding; (iii) fixes to bugs that are triggered by initial data sets with particularly large separations or BH velocities; (iv) changes to the preconditioning that lead to more robust and more efficient solution of the discretized linear system; (v) the possibility to specify ADM energy and angular momentum of the initial-data set instead of orbital frequency and radial velocity; and (vi) various code-cleanup and refactoring to improve maintainability. In addition, because `SpEC` was originally written and tested with binaries on quasicircular orbits in mind, running simulations of highly eccentric and hyperbolic-like orbits required many adjustments, which we outline in section 3.3.

We have also implemented different initial data that provide improvements for reducing junk radiation and improving performance of high-spin simulation. For newer simulations we typically use SHK [208] initial data for spins below 0.8 and SphSKS [207] for larger spins. Which type of initial data was used is available in the metadata field `initial_data_type` (see appendix A for metadata details). We also typically use a negative expansion boundary condition in the initial data. This allows us to place the excision boundary inside the AH in the initial data and eliminates the need to extrapolate to the horizon interior when starting the evolution [208].

New features include a new method for measuring spin directions based on [239], and new methods of constructing free fields for initial data [206, 207]. Additionally, we have tuned AMR tolerances to produce a more uniform error across the grid and throughout the simulation. In particular, we have increased the resolution in the wavezone spherical shells to ensure that they are not the dominant source of error. As part of this, we improved the way we handle spherical harmonic filtering. `SpEC` uses a scalar spherical harmonic decomposition, which causes aliasing errors when involving higher rank tensors. A tedious but straightforward derivation shows that this aliasing can be eliminated by transforming to a tensor spherical harmonic basis, zeroing the top four ℓ modes, and then transforming back to the scalar spherical harmonic basis [240]. We precompute and cache the transformation rules on disk, then load them for the current L in each spherical shell. This storage algorithm has seen several improvements, including copying the cache on disk to node-local SSDs to reduce load time.

Improvements to the feedback control systems [228, 229, 241] were required for robustly simulating higher mass ratio configurations, mostly requiring tightening control system tolerances to avoid drift from the optimal solution. A major overhaul of the rotation control system was also done, so that it now has the same interface as the other control systems. This eliminated many special cases in the control system code, reducing the complexity of that code considerably.

Extracting all the Weyl components and extrapolating them to infinity is extremely inaccurate unless junk radiation is very small. We can significantly reduce the junk radiation by allowing the initial junk radiation pulse to evolve to the outer spherical shells, and then dropping the shells containing the junk radiation [242]. This causes a discontinuous change in the outer boundary, but drastically reduces junk radiation reflections off the outer boundary. While effective, this procedure is not done by default in our simulations.

Finally, two key new features are a way to reduce the effect of junk radiation on errors, as explained in detail in section 3.2, and a new algorithm for determining the end of the junk phase (the relaxation time), as described in section 3.5.

3.2. Perform branching after junk

Assessing the convergence of numerical relativity simulations is complicated by the presence of initial transients, usually referred to as junk radiation. We intentionally configure AMR so that it does not resolve the initial junk transients. This is for two reasons: First, the junk radiation is not astrophysically interesting, so the beginning of the simulation that contains significant junk is eventually discarded. Second, resolving the junk is computationally expensive because the transients have small wavelengths and large frequencies.

Because the junk is not resolved, two otherwise-identical simulations starting at $t=0$ with different AMR tolerances have different, nonconvergent junk transients. As a result, they have slightly different masses, spins, and orbital parameters even at $t=t_{\rm junk}$, where $t_{\rm junk}$ is some time after which the primary junk transients are deemed to have sufficiently decayed away, though residual reflections may remain (see section 3.5). In other words, the unresolved junk transients leave behind an effectively random small change in the physical observables. Because of this, otherwise-identical simulations starting at $t=0$ with different AMR tolerances can fail to converge even at times $t>t_{\rm junk}$.

Our solution to this problem is to modify the way in which we specify simulations with multiple resolutions. We call this method 'perform branching after junk' (PBandJ). Instead of starting multiple simulations with different AMR tolerances at $t=0$, we start a single simulation with one AMR tolerance at $t=0$. We then choose some time $t_{\rm PBandJ}>t_{\rm junk}$, which is independent of the reference time chosen by the algorithm in section 3.5. When that simulation gets to $t=t_{\rm PBandJ}$ we fork several copies of that simulation, each copy having a different AMR tolerance for $t>t_{\rm PBandJ}$. Effectively, a snapshot of the first simulation at $t=t_{\rm PBandJ}$ serves as initial data for the copies. This procedure provides better convergence results and error estimates, by simply ignoring the initial part of the simulation that is astrophysically uninteresting and that we will discard anyway for waveform analysis. Most simulations in the catalog carried out after year 2020 use this procedure, and $t_{\rm PBandJ}$ is typically chosen to be about 2.5 orbits after $t=0$. Typically, the first simulation that evolves from $t=0$ to $t=t_{\rm PBandJ}$ is the highest-resolution simulation, and the forked copies are at lower resolution. However, occasionally we decide that our resolution is insufficient after the first simulation has finished, and in this case we fork an additional copy at higher resolution starting at $t=t_{\rm PBandJ}$.

3.3. Adjustments for highly eccentric and hyperbolic-like orbits

To evolve highly eccentric ($e_{\rm ref}\gtrsim 0.5$) BBHs and hyperbolic-like encounters in `SpEC`, several adjustments are made to resolve the fast dynamics characteristic of these systems. Because eccentric binaries emit pulses of GWs, the algorithm to choose the output frequency of extracted GWs was changed to be proportional to the highest instantaneous orbital frequency achieved during the entire past of the simulation. For hyperbolic events, it is not possible to meaningfully define an orbital frequency, so instead waveforms are exported in timesteps of $0.5M$, which we find to be sufficient to capture the characteristic burst of radiation emitted during the periastron passage.

Junk radiation requires special care for these types of simulations, since we usually do not have extra early orbits that we can discard. Instead, scattering and highly eccentric encounters start at very large initial distance (typically, $D_0=250M$) to make sure that junk radiation has

sufficiently decayed before the first periastron passage. Junk radiation is also what necessitated some of the changes to the initial data code detailed above. Previous versions of SpEC suppressed AMR in the outer spherical shell grids until the pulse of junk radiation passed through the outer boundary. However, because of the fast dynamics of hyperbolic or extremely eccentric cases, inner shells need refinement already fairly early on, usually before the pulse crosses the outer boundary. To allow inner shells to refine as soon as possible, AMR is no longer suppressed globally, but individual spectral elements are allowed to refine as soon as the junk pulse passes through them.

Another adjustment for these systems is the choice of shorter time scales to trigger AMR, which is necessary for AMR to adapt quickly enough to the rapid change in the radial separation. Furthermore, significant refinement is needed during the close periastron passage. We have therefore added the ability for AMR to split (i. e. *h*-refine) hollow cylindrical elements in the angular direction.

The gauge source functions in SpEC start from an initial gauge that is slowly rolled off towards the damped harmonic gauge. If the initial separation between the BHs is very small, we shorten this rolloff time to make sure the system is in purely damped harmonic gauge well before the first periastron passage. Likewise, the time for PBandJ (see section 3.2) is chosen such that it occurs before the first periastron passage. Typically, junk radiation is still present in the outer shells at the PBandJ time, but the most severe impact of junk radiation is mitigated. More details on the adjustments for highly eccentric and hyperbolic-like orbits will be presented in [243].

3.4. Eccentricity and mean anomaly

Each simulation in the catalog contains metadata fields for eccentricity and mean anomaly (see appendix A for a list of all metadata fields.). Because there are multiple definitions of eccentricity in general relativity, and because both eccentricity and mean anomaly vary as the simulation proceeds, we emphasize that users who care about particular definitions and precise values of these quantities should compute eccentricity and mean anomaly themselves in a consistent manner, using either waveforms or trajectories according to whatever method they choose. In particular, when comparing the eccentricities and mean anomalies of a series of simulations, those comparisons are most physically meaningful when the eccentricities and mean anomalies are computed at some time relative to merger. In contrast, here we compute eccentricities and mean anomalies at some time early in the inspiral after junk, since the number of orbits in our simulations varies widely from case to case.

Nevertheless, we describe here how we compute eccentricities and mean anomalies. We use three methods. The first is the method described in [216], where a post-Newtonian-based functional form (equation (10) of [216]) is fit to the time derivative of the angular frequency of the orbital trajectory. This functional form is based on an expansion in small e, and includes spin-spin corrections and radiation reaction.

The second method is the one described in [217], which is similar to the first method, except that the functional form (equation (9) of [217]) allows for general e that can be large. For general e, it is usually necessary to invert the Kepler equation to solve for the eccentric anomaly $u(t)$ as a function of time t. Here the Kepler equation at 1PN order reads

$$t - t_{\rm ref} = a^{3/2} M^{-1/2} \left(1 + \frac{9-\eta}{2aM^{-1}}\right) \left[u(t) - e\left(1 + \frac{3\eta - 8}{2aM^{-1}}\right) \sin u(t) - \ell\right], \tag{9}$$

where e is the (Newtonian) eccentricity, ℓ is the mean anomaly at time $t = t_{\rm ref}$, a is the (Newtonian) semimajor axis, M is the total mass $m_1 + m_2$, and η is the symmetric mass ratio

$m_1 m_2/(m_1+m_2)^2$. Here we use geometric units $G = c = 1$. However, in [217] the inverse of the Kepler equation is approximated as the closed-form expression

$$u(t) = a^{-3/2} M^{1/2} \left(1 - \frac{9-\eta}{2aM^{-1}}\right)(t - t_{\rm ref}) + \ell, \tag{10}$$

which is a reasonable approximation for $e \lesssim 0.5$, but not for larger eccentricities.

The third method works best for large eccentricities, when eccentricity dominates the trajectory. This method is the same as the second method, except the functional form that is fit to the trajectory omits spin-spin terms and radiation-reaction terms, and the Kepler equation is inverted numerically. In other words, the functional form is equation (9) of [217] with $C_1 = C_2 = C_3 = C_4 = 0$, with $u(t)$ determined numerically from equation (9).

We use the first method for all cases with $e < 0.01$. For very small eccentricities the mean anomaly is degenerate and has no physical meaning. For larger eccentricities $e > 0.01$, we try both the second and third method, and choose the result with the smallest value of the L^2 norm of the fit residual divided by the number of fit parameters.

Note that there are some simulations with empty eccentricity or mean anomaly metadata fields, and other simulations where the eccentricity metadata field is a string (where the string describes a reason for lack of a numerical eccentricity such as 'simulation too short'). These simulations are either head-on or nearly-head-on collisions, hyperbolic encounters where the objects are unbound and escape to infinity, or simulations with too few orbits to reliably measure an eccentricity.

3.5. *Junk radiation, relaxation time, and reference time*

Although users are free to use their own methods to remove initial 'junk radiation' transients from simulations, we provide several metadata fields to assist with this. The metadata provides a suggested `relaxation_time` in units of total mass M. This is our estimate of the amount of time that should be removed from the beginning of the time series so that junk remains acceptably small. Note that `relaxation_time` is simulation-dependent and resolution-dependent. Users with applications that are particularly sensitive to junk are encouraged to measure the junk themselves in a manner of their choosing and truncate the time series appropriately for their use case.

We recently presented a new algorithm for determining `relaxation_time` [170]. This method, which we call HHT for Hilbert–Huang Transform [244], involves constructing and analyzing the empirical mode decomposition [244] of a signal and determining when high-frequency content has decayed to a desired tolerance. In our case, the signal we use is the irreducible horizon mass as a function of time, because it exhibits junk effects and settles to a constant at late times. This HHT method is used for most of our simulations. However, the new algorithm sometimes fails for some shorter and older simulations. For example, sometimes the simulation is so short that junk does not fully decay before the simulation ends; this is especially problematic for head-on and scattering simulations. When the HHT method fails, we revert back to the old RMS (root-mean-squared) algorithm to determine the `relaxation_time` (see [170] for a description of failure modes and a description of the RMS algorithm).

The metadata for each simulation provides another field called `reference_time`. This is the time at which we measure quantities that parameterize the simulation, such as the individual black-hole masses and spins, and orbital parameters such as coordinate separations and orbital eccentricity. The values of these quantities at `reference_time` are different from

their values at $t = 0$ both because of junk radiation and because some of these quantities (e.g. spin directions and eccentricity) have non-negligible time-dependence even early in the simulation. Typically we choose `reference_time` equal to `relaxation_time`. However, for some simulations (typically very short ones such as nearly head-on collisions) we choose `reference_time` by hand. The catalog metadata for each simulation now includes information as to which algorithm was used to determine the relaxation and reference times, in the metadata field `t_relaxed_algorithm`.

3.6. Memory correction

An important prediction of Einstein's theory of relativity is that whenever a system emits gravitational radiation, that radiation will permanently change the spacetime that it propagates through. This permanent change is called the *GW memory* and predominantly manifests in a GW as a net change in the strain between early and late times. Waveforms that are extracted at future null infinity from numerical relativity simulations using extrapolation, however, fail to naturally capture memory [245]. Fortunately, [118] showed that these waveforms can be corrected to include memory by computing and adding certain contributions to the strain that seem to be missing by using the BMS balance laws. With this correction, extrapolated waveforms respect the Bianchi identities to a much higher degree and agree much more with waveforms extracted using Cauchy-characteristic evolution (CCE) [111, 120, 246–249] that naturally contain the memory. Consequently, we have updated the waveforms in our catalog to include memory, using the technique outlined in [118].

The procedure consists of calculating the null memory contribution to the strain via

$$h^{\text{memory}} = \frac{1}{2}\bar{\eth}^2\mathfrak{D}^{-1}\left[\frac{1}{4}\int_{-\infty}^{u}|\dot{h}|^2\mathrm{d}u\right], \tag{11}$$

where

$$\eth f(\theta,\phi) \equiv -(\sin\theta)^{+s}\left(\partial_\theta + i\csc\theta\partial_\phi\right)\left[(\sin\theta)^{-s}f(\theta,\phi)\right] \tag{12}$$

is the Geroch–Held–Penrose spin-weight operator (here represented when acting on a spin-weight s function f in spherical coordinates) [250] and

$$\mathfrak{D} \equiv \frac{1}{8}\left(\bar{\eth}\eth\right)\left(\bar{\eth}\eth + 2\right). \tag{13}$$

For each of our simulations, we correct each waveform by computing h^{memory} via equation (11) and adding it to the strain. Note that we take the lower limit of integration to be the relaxation time of the simulation. We also correct Ψ_4 by adding $-\frac{1}{2}\ddot{h}^{\text{memory}}$.

4. Data archive, versioning for reproducibility, and user interface

As the number of simulations in the SXS catalog continues to grow, the management of the data becomes increasingly complex and challenging. To support new analyses and to ensure reproducibility of existing results, the data must be made available to the scientific community in ways that are easy to search and access, while scaling to accommodate the growing number of simulations in the catalog, yet remaining accessible over the long term. To balance these sometimes conflicting objectives, we have developed a user interface that provides consistency while transparently handling a variety of data formats and storage locations.

4.1. Archiving and versioning data

We assign a unique 'SXS ID' to each simulation in the catalog, of the form `SXS:BBH:1234`. The numerical portion is—as yet—always a four-digit number, zero-padded if necessary. The numbers are not always consecutive, and may not correspond to when the simulations were performed. Each simulation is deposited as a separate record in a long-term open-access digital repository—originally Zenodo [251], and more recently CaltechDATA [252]. Corresponding to the ID, each simulation is also given a digital object identifier (DOI) that is simply the SXS ID combined with the SXS Collaboration's DOI prefix, `10.26138`. Thus, for example, SXS:BBH:1234 will always be accessible at https://doi.org/10.26138/SXS:BBH:1234, which redirects to whichever digital repository holds that simulation.

However, the data for a simulation may be updated over time, as improvements are made to the post-processing methods, as conventions change, or as bugs are fixed. We have made such updates a number of times in the past. One of the more significant changes was an overall sign change to the definition of the strain, introduced just before the 2019 catalog [93] and applied retroactively to simulations run before 2019. That update to the catalog also introduced center-of-mass corrections [159]. The update of the catalog associated with the current paper introduces memory corrections to the waveforms, as discussed in section 3.6. We have applied memory corrections and center-of-mass corrections to all simulations, not only newer ones. There have been numerous less significant changes, such as small corrections to metadata.

Any time a file changes in any way, it could potentially alter the results of some analysis. To ensure true reproducibility, we now use version numbers for each simulation, which are incremented whenever any file is changed[20]. As of this release of the catalog, all simulations have version number 3.0; prior releases have also been specified with lower version numbers. These numbers can be appended to the SXS ID to specify the version. In particular, each version of each simulation is also given a DOI, such as `10.26138/SXS:BBH:1234v3.0`. While the unversioned DOI will always be updated to point to the most recent version, the versioned DOIs will always point to the specific data associated with that version. For reproducibility, it is best to specify the version number whenever possible.

Referring individually to versioned SXS IDs can be cumbersome. Therefore, we also provide an overall version number for the collection of the most recent versions of all simulations at any time. As of the release of this paper, the catalog version is 3.0.0, and is archived on Zenodo [158]. This version number will be incremented whenever any simulation is updated. By referring to the catalog version, the version of any particular simulation—unless otherwise stated—is implied by what is stored in that version of the catalog. We recommend that users specify the catalog version when discussing analyses that use simulations from a single version the catalog, or specific simulation versions otherwise. These versions should also be cited, for which we provide a convenient function `sxs.cite` which can provide the citation for a particular catalog version, as with [158], or citations for specific simulations and the papers that introduced them. We also provide a user interface that directly uses these SXS IDs, with or without versions, and the versioned catalog.

4.2. User interface

In constructing the user interface, we begin by considering the perspective of a new user approaching the catalog. First, the user needs to know what data are available, and be able

[20] The number of versions varies between simulations, ranging from 8 for some older simulations to just 1 for the newest simulations.

to easily sort through the simulations to find those that are relevant to a particular analysis. While this paper provides a high-level aggregate survey, the user will need detailed information about each simulation individually, and means of selecting them both interactively and algorithmically. Having selected the simulations to analyze, the user will then need to know where and how to obtain the data, as well as how to load and use the data. Ideally, the procedure should be as simple and uniform as possible, remaining constant over time so that analyses can be reliably reproduced—or expanded and reused—in the future.

To meet these needs we provide the `sxs` Python package[21], which can automatically obtain and load the metadata for the entire catalog, as well as data for each simulation individually. The metadata can be loaded into a single table, which can be filtered and sorted using standard pandas queries to find simulations of interest. We provide an interactive interface to the metadata at https://data.black-holes.org[22]. The metadata can be loaded, for example, with `sxs.load("simulations", tag = "3.0.0")`, where the optional `tag` argument specifies the version of the catalog to load. Because the metadata are heterogeneous and not necessarily ideal for searching or sorting, it is also possible to load a `"dataframe"` with homogeneous data types for each field. In either case, the chosen version of the catalog will then provide the default version of simulations throughout the Python session.

Any particular simulation can be loaded by SXS ID—for example as `sxs.load("SXS:BBH:1234")` or `sxs.load("SXS:BBH:1234v3.0")`. Data available for each simulation include horizon information as well as strain and Ψ_4 waveforms, and can be 'lazily' obtained and loaded only when needed, simply by accessing attributes of the simulation object. This frees the user from having to download and manage the data manually—or even know where or in what format the data are stored. By default, the highest available resolution is used. In the case of extrapolated waveforms, a default extrapolation order of $N = 2$ is chosen. However, different choices can easily be specified by the user. The package also caches data locally, speeding up subsequent accesses, though this behavior can also be overridden by the user. See listing 1 for an example of using the `sxs` package.

Partly because of the number of versions of each simulation, and partly just because of the increasing number of simulations, the sheer size of the catalog has become difficult to sustain. The total size of all simulations prior to this data release was 12 TiB. For the previous version of the catalog, a user who wanted the highest-resolution default-extrapolated strain waveform for all simulations needed to download and/or store over 180 GiB of data which includes all 2018 simulations.

We use several techniques to deal with this problem. First, we are limiting the types of data we publish to Zenodo or CaltechDATA—though we retain all data locally. Previously, we included finite-radius waveforms from the simulations, alongside the extrapolated waveforms. We have always advised against using finite-radius waveforms, as we expect the extrapolated waveforms to be more physically relevant. Moreover, there are roughly 7 times more finite-radius waveforms than extrapolated, making the catalog far larger than it needs to be. Therefore, we no longer publish finite-radius waveforms as part of the catalog. Second, we have introduced a new waveform format that compresses each waveform by an average factor of 7 compared to the old (compressed but wasteful) format. As described in appendix C, the new format applies a number of complicated non-standard transformations to the data.

[21] The `sxs` package can be installed either using `pip/uv` or `conda/mamba/pixi`. The source code and documentation are available at https://github.com/sxs-collaboration/sxs. Each release is archived at https://doi.org/10.5281/zenodo.13891077.

[22] This interface is based on a `marimo` notebook [253], which uses `Pyodide` [254] to run a simplified version of the `sxs` package in a Python session directly in the browser.

```
import sxs
import matplotlib.pyplot as plt

# Load a pandas.DataFrame of metadata for all simulations
df = sxs.load("dataframe", tag="3.0.0")

# Find SXS ID of BBH simulation with highest effective spin
highest_chi_eff = df.BBH["reference_chi_eff"].idxmax()

# Load that simulation
sim = sxs.load(highest_chi_eff)

# Print a summary of the simulation's parameters
print(sim)

# Print all metadata of the simulation
print(sim.metadata)

# Obtain and load the strain data
h = sim.h

# Plot the real parts of the (2,2) and (2,0) modes
plt.plot(h.t, h.real[:, h.index(2,2)], label=r"(2,2)")
plt.plot(h.t, h.real[:, h.index(2,0)], label=r"(2,0)")
plt.xlabel("Time ($M$)")
plt.ylabel(r"$\Re\{h^{\ell, m}\}$")
plt.title(sim.sxs_id)
plt.legend()
```

Listing 1. An example of using the `sxs` Python package to load data. Note that the user does not have to download any data; the package will automatically manage downloading and caching data

Together, these changes have reduced the size of this most recent version of the catalog to just 410 GiB in total. A user wanting the highest-resolution default-extrapolated strain waveform for all simulations would need to download and/or store just 14 GiB which includes all 3756 simulations—23 times less space compared to the previous version of the catalog. As before, a user who wants only a single waveform from a single simulation can download only that waveform, and not the entire catalog. Because of the admittedly cryptic waveform format, it is unrealistic to expect the user to load the data directly from the supplied files. Instead, the `sxs` package is designed to insulate the user from this challenge and other inconveniences such as changes to file names and locations, or even future changes to formats.

5. Conclusion

In this paper, we have presented an update to the SXSs BBH catalog. Using the highly efficient spectral methods implemented in the `SpEC`, we have increased the total number of configurations from 2018 to 3756. The catalog now more densely covers the parameter space with precessing simulations up to mass ratio $q = 8$ with dimensionless spins up to $|\chi| \leqslant 0.8$ with near-zero eccentricity. The catalog includes some simulations at higher mass ratios with moderate spin and more than 250 eccentric simulations. We have also deprecated and rerun older simulations and ones with anomalously large errors in the waveform. The median waveform difference between resolutions over all 3756 simulations is 4×10^{-4}, with a median of 22 orbits, while the longest simulation is 148 orbits. All the waveforms in the catalog are center-of-mass and GW-memory corrected [118, 159]. We provide a python package, `sxs` [92], to simplify accessing the catalog. Because of a new compression algorithm and deprecation of lower-quality simulations, the `sxs` package is by far the best method for users to access the catalog. We estimate the total CPU cost of all the simulations in the catalog to be only 480 000 000 core-hours. Using spectral methods for long, precessing BBH inspiral-merger-ringdown simulations is over 1000 times more efficient than using finite-difference methods for a few orbits of non-spinning BBHs at comparable accuracies; see, e.g. [41]. To date GPU-based finite-difference codes have yet to prove competitive with CPU-based codes using spectral methods. The full catalog is publicly available at https://data.black-holes.org .

We assess the readiness of our catalog for use in current and next-generation detectors in section 2.4. We find that for simulations long enough to span the entire signal in the detector band, most simulations are accurate enough for current detectors, but significant improvements need to be made, both in terms of accuracy and length, for next-generation ground-based and space-based observatories. Significant improvements will also need to be made for next-generation detectors in hybridizing post-Newtonian and numerical relativity waveforms [59, 72, 85–91] in order to have waveform models that span the entire signal that will be in band.

In the future, we will expand the catalog towards higher mass ratios and focus on filling out the eccentric parameter space. More challenging will be increasing the accuracy to meet the requirements of next-generation detectors, especially since the required length of numerical-relativity waveforms depends on how late one can hybridize them with post-Newtonian waveforms. A major challenge for increasing accuracy is significantly reducing the spurious 'junk' gravitational radiation generated by imperfect initial data (see, e.g. [208] for some progress in this direction). Finally, in the near term we will release a catalog of all of the simulations presented here, but using CCE [120, 246–249] to extract the GWs. This has several advantages, such as naturally producing the correct GW memory and independently producing all five complex Weyl components, the news, and the strain at future null infinity without any need to differentiate or integrate quantities.

Data availability statement

The data that support the findings of this study are openly available via a python package at the following URL/DOI: https://zenodo.org/doi/10.5281/zenodo.15412737 [92].

Acknowledgments

We thank Zachariah Etienne and Sebastian Khan for pointing out problems with some of the now-deprecated waveforms. We thank Tom Morrell at the Caltech Library for his advice on

hosting the data archive with CaltechDATA, and we thank the Caltech Library for providing this excellent service. This material is based upon work supported by the National Science Foundation under Grants Nos. PHY-2309211, PHY-2309231 and OAC-2209656 at Caltech; by Nos. PHY-2407742, PHY-2207342 and OAC-2209655 at Cornell; by CAREER Award No. PHY-2047382 at the University of Mississippi; by Nos. PHY-2309301, DMS-2309609, PHY-2110496 and AST-2407454 at the University of Massachusetts, Dartmouth; and by Nos. PHY-2208014 and AST-2219109 at Cal State Fullerton. V V and S F acknowledge support from University of Massachusetts, Dartmouth's Marine and Undersea Technology (MUST) Research Program funded by the Office of Naval Research (ONR) under Grant No. N00014-23-1-2141. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work was supported by the Sherman Fairchild Foundation at Caltech and Cornell; by a Sloan Foundation Research Fellowship at the University of Mississippi; and by the Dan Black Family Trust, and Nicholas and Lee Begovich at Cal State Fullerton. Support for this work was provided by NASA through the NASA Hubble Fellowship Grant Number HST-HF2-51562.001-A awarded by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Incorporated, under NASA contract NAS5-26555. H R R acknowledges financial support provided under the European Union's H2020 ERC Advanced Grant 'Black holes: gravitational engines of discovery' Grant Agreement No. Gravitas–101052587. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. A R B is supported by the Veni research programme which is financed by the Dutch Research Council (NWO) under the Grant VI.Veni.222.396; acknowledges support from the Spanish Agencia Estatal de Investigación Grant PID2022-138626NB-I00 funded by MICIU/AEI/10.13039/501100011033 and the ERDF/EU; is supported by the Spanish Ministerio de Ciencia, Innovación y Universidades (Beatriz Galindo, BG23/00056) and cofinanced by UIB. K. Z. C. acknowledges support by the Hungarian Scientific Research Fund NKFIH Grant No. K-142423.

Computations were performed on the Wheeler cluster at Caltech, which is supported by the Sherman Fairchild Foundation and by Caltech. The computations presented here were conducted in the Resnick High Performance Computing Center, a facility supported by Resnick Sustainability Institute at the California Institute of Technology. This work used Anvil at Purdue University [255], Expanse at San Diego Supercomputer Center [256], Stampede 2 at Texas Advanced Computing Center, through allocation PHY990002 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation Grants #2138259, #2138286, #2138307, #2137603 and #2138296 [257]. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing computational resources that have contributed to the research results reported within this paper. URL: http://www.tacc.utexas.edu. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation Grant Number ACI-1548562. Specifically, it used the Bridges-2 system, which is supported by NSF Award Number ACI-1928147, at the Pittsburgh Supercomputing Center (PSC), and the Comet system, which is supported by NSF Award Number 1341698, at the San Diego Supercomputer Center [258]. This work is also part of the NCSA Blue Waters PRAC allocation support by the National Science Foundation (Award Number OCI-0725070) [259, 260]. Computations were performed on the high-performance computer system Minerva at the Max Planck Institute for Gravitational Physics in Potsdam, as well as on the Raven and Urania clusters at the Max Planck Computing

and Data Facility. Computations were performed on the Ocean high-performance computing cluster at Cal State Fullerton. Computations were performed using the Dutch national e-infrastructure with the support of the SURF Cooperative using Grant No. EINF-7366 and NWO-2024.002.

Appendix A. Metadata fields

Using the `sxs` package, metadata fields can be obtained as a python dictionary via `sim.metadata` for an individual simulation `sim`, or as entries in a pandas DataFrame for all simulations via `sxs.load(`dataframe')`, as illustrated in listing 1. The 'introduced' version marks the first `metadata_format_revision` in which a metadata field is available. Metadata fields that describe spatial vector quantities, such as positions, spins, and linear and angular momenta, are of type `list[float]` and are expressed in Cartesian components (x, y, z) in the inertial frame.

Metadata field group: Identification

`simulation_name` (type: `str`; introduced: v0)
A non-unique SXS-assigned identifier chosen before the simulation has been run. Useful only for SXS members building and debugging the catalog.

`alternative_names` (type: `list[str]`; introduced: v0)
Comma-separated array of alternative names, longer, more descriptive, and/or indicating the specific series of simulations this configuration belongs to. One of these alternative names is the 'SXS:BBH:dddd' id-number, which is guaranteed to be unique.

`keywords` (type: `list[str]`; introduced: v0)
List of free-form keywords. Presence of the keyword 'deprecated' means that this simulation has been deprecated.

`job_archiver_email` (type: `str`; introduced: v2)
Email of person who archived this simulation into the catalog; usually the person who ran the simulation. Useful only for SXS members building and debugging the catalog.

Metadata field group: References

`citation_dois` (type: `list[str]`; introduced: v2)
DOIs to cite when using this simulation.

Metadata field group: Input parameters for initial data

`object1` (type: `str`; introduced: v0)
Keyword description to identify the object 1 type. One of {`bh`, `ns`}.

`object2` (type: `str`; introduced: v0)
Keyword description to identify the object 2 type. One of {`bh`, `ns`}.

`initial_data_type` (type: `str`; introduced: v0)
Type of initial data. One of `BBH_CFMS`—conformally flat, maximal slice; `BBH_SKS`—superposed Kerr–Schild; `BBH_SHK`—superposed harmonic Kerr–Schild [208]; `BBH_SSphKS`—superposed spherical Kerr–Schild [207]; `BHNS`; `NSNS`.

`initial_separation` (type: `float`; introduced: v0)
Coordinate separation D_0 between centers of compact objects, as passed to the initial data solver [212, 214, 261] (code units).

`initial_orbital_frequency` (type: `float`; introduced: v0)
Initial orbital frequency Ω_0 passed to the initial-data solver [212, 214] (code units).
`initial_adot` (type: `float`; introduced: v0)
Radial velocity parameter $\dot{a}_0$ passed to the initial data solver [213, 214].

Metadata field group: Measurements of initial data

`initial_ADM_energy` (type: `float`; introduced: v0)
ADM energy of the initial data (code units).
`initial_ADM_linear_momentum` (type: `list[float]`; introduced: v0)
ADM linear momentum of the initial data (code units).
`initial_ADM_angular_momentum` (type: `list[float]`; introduced: v0)
ADM angular momentum of the initial data (code units).
`initial_mass1` (type: `float`; introduced: v0)
Christodoulou mass of apparent horizon 1 at initial data (code units).
`initial_mass2` (type: `float`; introduced: v0)
Christodoulou mass of apparent horizon 2 at initial data (code units).
`initial_dimensionless_spin1` (type: `list[float]`; introduced: v0)
Dimensionless spin of object 1 in the initial data.
`initial_dimensionless_spin2` (type: `list[float]`; introduced: v0)
Dimensionless spin of object 2 in the initial data.
`initial_position1` (type: `list[float]`; introduced: v0)
Initial coordinates of the center of body 1.
`initial_position2` (type: `list[float]`; introduced: v0)
Initial coordinates of the center of body 2.

Metadata field group: Reference quantities

`relaxation_time` (type: `float`; introduced: v0)
Time at which we deem junk radiation to have sufficiently decayed (code units).
`reference_time` (type: `float`; introduced: v0)
Time at which reference quantities are extracted from the evolution (code units).
`reference_mass1` (type: `float`; introduced: v0)
Christodoulou mass of black hole 1 at reference time (code units).
`reference_mass2` (type: `float`; introduced: v0)
Christodoulou mass of black hole 2 at reference time (code units).
`reference_dimensionless_spin1` (type: `list[float]`; introduced: v0)
Dimensionless spin of object 1 at reference time.
`reference_dimensionless_spin2` (type: `list[float]`; introduced: v0)
Dimensionless spin of object 2 at reference time.
`reference_position1` (type: `list[float]`; introduced: v0)
Coordinates of the center of body 1 at reference time.
`reference_position2` (type: `list[float]`; introduced: v0)
Coordinates of the center of body 2 at reference time.
`reference_orbital_frequency` (type: `list[float]`; introduced: v0)
Orbital angular frequency vector at reference time (code units).
`reference_mean_anomaly` (type: `float`; introduced: v0)
Mean anomaly at reference time.
`reference_eccentricity` (type: `float`; introduced: v0)
Orbital eccentricity at reference time [262].

Metadata field group: Properties of merger/final quantities

`number_of_orbits_from_reference_time` (type: `float`; introduced: v2)
Number of orbits from reference time until formation of a common apparent horizon.

`number_of_orbits_from_start` (type: `float`; introduced: v2)
Number of orbits from start of simulation until formation of a common apparent horizon.

`common_horizon_time` (type: `float`; introduced: v0)
Evolution time at which common horizon is first detected.

`remnant_mass` (type: `float`; introduced: v0)
Final Christodoulou mass of the remnant black hole after merger.

`remnant_dimensionless_spin` (type: `list[float]`; introduced: v0)
Dimensionless spin of the remnant black hole after merger.

`remnant_velocity` (type: `list[float]`; introduced: v0)
Linear velocity of the remnant black hole after merger.

Metadata field group: Code information

`spec_revisions` (type: `list[str]`; introduced: v0)
Array of git revisions of the evolution code.

`spells_revision` (type: `str`; introduced: v0)
Git revision of initial data solver.

`date_link_earliest` (type: `str`; introduced: v2)
Earliest link time of code used to perform this simulation.

`internal_changelog` (type: `dict`; introduced: v2)
Text describing changes made in different `internal_minor_versions` of this local simulation. Always starts empty for new simulations.

`internal_minor_version` (type: `int`; introduced: v2)
Incremented when anything changes in this local simulation that is not tracked by the fields `metadata_format_revision`, `metadata_content_revision`, or `postprocess_revision`. No relation to DOI revision numbers. Always starts at 0 for new simulations.

`metadata_content_revision` (type: `int`; introduced: v2)
Incremented when values in the metadata change (which should seldom happen). Updated for all (non-deprecated) simulations at once. No relation to DOI revision numbers. At the time of this catalog release, all non-deprecated simulations carried `metadata_content_revision = 1`.

`metadata_format_revision` (type: `int`; introduced: v2)
Incremented when keys in the metadata change (which should seldom happen). Updated for all (non-deprecated) simulations at once. No relation to DOI revision numbers. At the time of this catalog release, all non-deprecated simulations carried `metadata_format_revision = 2`.

`postprocess_revision` (type: `int`; introduced: v2)
Incremented when anything changes that is not a raw SpEC output, such as re-computation of extrapolation, center-of-mass-correction, or memory-correction using newer algorithms or different parameters. Should not occur often. Updated for all (non-deprecated) simulations at once. No relation to DOI revision numbers. At the time of this catalog release, all non-deprecated simulations carried `postprocess_revision = 1`.

`t_relaxed_algorithm` (type: `dict`; introduced: v2)
`t_relaxed_algorithm` is a dict. It contains fields:

- `algorithm`: either 'HHT' or 'RMS'.
- `reason`: present only for 'RMS' method; text explaining why HHT method failed and we fell back to RMS.
- `reference_time_method`: usually absent, but '`set_by_hand`' for certain simulations (almost head-on) where a reference time was explicitly set by hand and is not correlated with `relaxation_time`.

`pbj_info` (type: `dict`; introduced: v2)
This is a dict that contains {`base_lev : str, transition_time : float, base_lev_bitwise_identical : str` }

- `base_lev` is the Lev that is shared between all PBandJ Levs. It is a string like 'Lev3'.
- `transition_time` is the time at which PBandJ happens. That is, before `transition_time`, all the Levs should be identical. If there is no PBandJ, then `transition_time` is 0.0 and `base_lev` is the same as the Lev that the metadata.json file is in.
- `base_lev_bitwise_identical` is either the string 'true' or the string 'false'. If 'true', this means that the current Lev (the one the metadata.json file is in) and its `base_lev` actually are bitwise identical up to (approx) `transition_time`. The 'false' case occurs when someone ran an additional Lev at a later time, but `base_lev` had been deleted by the sysadmins, so the user reran `base_lev` and then did PBandJ to start the new lev. But the rerun of `base_lev` is not always bitwise identical to the original `base_lev` because something changed (timing-based stuff in SpEC, compiler version, libraries on the cluster, etc).

Metadata field group: Time stamps information

`date_postprocessing` (type: `str`; introduced: v2)
Timestamp of the most recent postprocessing of the raw simulation data to compute extrapolated, COM-corrected, memory-corrected waveforms.
`date_run_earliest` (type: `str`; introduced: v2)
Timestamp of when this simulation was started.
`date_run_latest` (type: `str`; introduced: v2)
Timestamp of when the last segment of this simulation started.

A.1. Deprecated metadata fields

For completeness, below are the metadata fields that are deprecated and no longer available. The 'deprecated' version marks the first `metadata_format_revision` in which a metadata field is no longer available.

Metadata field group: Identification

`point_of_contact_email`(type: `str`; introduced: v0; deprecated: v2)
Contact information for questions.
`authors_emails`(type: `list[str]`; introduced: v0; deprecated: v2)
List of authors' emails.

Metadata field group: References

`simulation_bibtex_keys`(type: `list[str]`; introduced: v0; deprecated: v2)
References which should be cited if this simulation is used.

`code_bibtex_keys`(type: `list[str]`; introduced: v0; deprecated: v2)
List of bibtex keys which are references about the evolution code.
`initial_data_bibtex_keys`(type: `list[str]`; introduced: v0; deprecated: v2)
List of bibtex keys which are references about the initial data code.
`quasicircular_bibtex_keys`(type: `list[str]`; introduced: v0; deprecated: v2)
List of bibtex keys which are references about creating quasicircular initial data.

Metadata field group: Properties of merger/final quantities

`number_of_orbits`(type: `float`; introduced: v0; deprecated: v2)
Number of orbits until formation of a common apparent horizon. Replaced by `number_of_orbits_from_reference_time` and `number_of_orbits_from_start`.

Metadata field group: Code information

`metadata_version`(type: `int`; introduced: v0; deprecated: v2)
This field has been replaced by the fields `metadata_format_revision` and `metadata_content_revision`. The 2013 catalog [171] implicitly carried metadata version 0. The 2019 catalog [93] carried metadata version 1.

Appendix B. Algorithm for superseding simulations

Since we are now deprecating simulations that we believe are not trustworthy, some automated way of finding new, similar simulations must be available. The `sxs` package provides the following algorithm as the default for finding a new simulation that is similar to the requested, deprecated simulation—although a different algorithm can be passed as an option. Our approach is very simplistic: we measure the distance between simulations in terms of their metadata parameters at their respective reference times. Since the SXS catalog also contains some simulations with neutron stars, systems of different types of compact objects are considered infinitely far apart in parameter space. We call the two systems we are comparing s and s'. We compare their mass ratios and dimensionless spin vectors, as well as their eccentricities. Because of the varying thresholds involved in eccentricity reduction, we set a pair of thresholds on the eccentricity magnitudes, $\epsilon = 10^{-2}$ and $\epsilon' = 10^{-3}$, as well as a single length threshold $N' = 20$. We ignore the eccentricities (essentially, setting $e = e' = 0$) if

(i) the reference eccentricity of system s is below ϵ,
(ii) the reference eccentricity of system s' is below ϵ', and
(ii) the length of system s' is greater than N' orbits.

We choose two different thresholds because our eccentricity-reduction methods have improved over the years [213–216], and the first system will generally be an older simulation in which eccentricities as high as 10^{-2} would have been considered non-eccentric, whereas newer simulations will need to have eccentricity below 10^{-3} to be considered non-eccentric, and thus a good match for the first system.

Finally, we compute the distance by summing the differences of all parameters in quadrature and taking the square-root:

$$\delta m = \sqrt{(q-q')^2 + |\vec{\chi}_1 - \vec{\chi}_1'|^2 + |\vec{\chi}_2 - \vec{\chi}_2'|^2 + |e\exp[i\ell] - e'\exp[i\ell']|^2}, \tag{B.1}$$

where q and q' are the reference mass ratios of the two systems, $\vec{\chi}_1$, $\vec{\chi}_1'$, $\vec{\chi}_2$, and $\vec{\chi}_2'$ are the reference dimensionless spins of the two black holes in the two systems, e and e' are

the reference eccentricities, and ℓ and ℓ' are the reference mean anomalies. Users may pass `auto_supersede = True` to the `sxs.load` function to load the 'closest' simulation by this measure, or choose a threshold such as `auto_supersede = 0.01` to raise an exception if no system with $\delta m \leqslant 0.01$ can be found. It is also possible to define a custom `MetadataMetric` and pass that as an argument to `sxs.load`.

Appendix C. Waveform format

Data published by the SXS collaboration prior to this publication occupies more than 12 TiB of storage on the open-access Zenodo repository—which currently constitutes over 1% of the total data storage on Zenodo [251]. This is a problematic amount of data for a single project, already inducing resistance from the maintainers of Zenodo. This paper more than doubles the number of simulations to be hosted, making the situation untenable. Limiting the types of data we publish will reduce the size of the catalog significantly, but still not enough.

The SXS collaboration has developed a new waveform format that compresses the data by an average factor of 7 compared to the old format, which used standard compression. This new format applies a number of complicated non-standard transformations to the data. The `sxs` package provides an interface that insulates the user from the details of the waveform format, and even allows for seamless compatibility between different versions of the catalog that use different formats. Nonetheless, details of the new waveform format and comparison to older formats may be of interest.

The fundamental idea is to manipulate the data in ways that increase the number of runs of constant bytes, which can be efficiently compressed with a standard compression algorithm. There are several lossless transformations for which the original data can be precisely reconstructed, as well as one lossy step with a selectable tolerance. The final step is compression with BZIP2 [263], followed by storage in a standard HDF5 [264] file. The full sequence of transformations is described section C.2. The results are shown in figure C1, demonstrating that the L^2 norm of the difference between the original and compressed data is less than 10^{-10} times the norm of the original data at each instant of time. The resulting files are—on average—roughly 7x smaller than they would be if the raw data were stored without compression, and over 5x smaller than they would be using standard compression techniques of HDF5.

C.1. Existing waveform formats

Because they represent the precious output of relatively expensive and long-running numerical-relativity simulations, waveform data have long been stored in formats designed to ensure that there is no information loss. The state of the art for lossless compression, used by most numerical relativity groups, is to store the data in HDF5 files with standard shuffle and gzip filters to enable compression [264][23]. However, this limits the amount of compression that can be achieved. As more and more waveforms appear in catalogs, storing and distributing the data in lossless formats becomes increasingly difficult. Lossy formats have also been used, where a tradeoff is made between the accuracy of the data and the size of the file [31, 265–269]. As numerical relativity catalogs become more unwieldy, this becomes a more attractive option.

[23] It is important to note that 'chunking' must also be done properly to ensure that the shuffle and gzip filters can perform adequately. In short, the data should be arranged so that successive timesteps of a single mode are contiguous in memory, so that data values change slowly as memory is traversed.

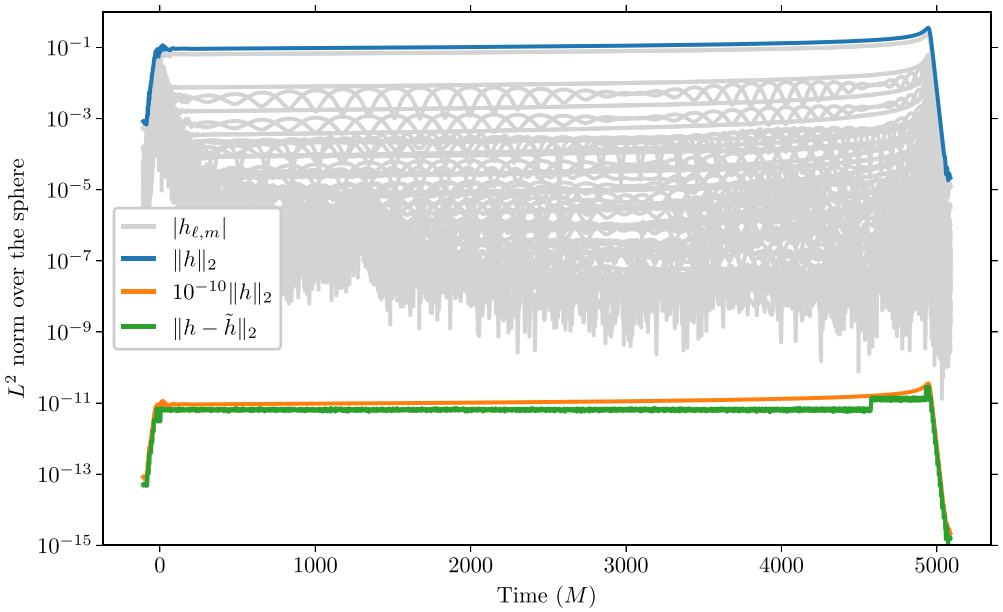

Figure C1. Error in the strain data after compression for SXS:BBH:2019—a mildly precessing system with mass ratio $q=4$. The light gray curves illustrate the amplitudes of the individual modes of the original strain. Their individual values are unimportant; we simply wish to indicate the range of values. The upper blue curve shows the L^2 norm of the strain over the sphere at each instant of time. The smooth orange curve towards the bottom shows 10^{-10} times that norm; the L^2 norm of the error in the compressed waveform is guaranteed to be less than this value. Finally, the green curve at the very bottom shows the actual L^2 norm of the difference between the original and compressed waveforms. We see that this difference does satisfy the error bound at all times. It changes in steps of roughly a factor of 2 as additional bits in the output data can be truncated as the norm increases or must be included as the norm decreases, as described in section C.2.3. The ragged band along the lower extent of the gray region suggests that the true error in each mode is *at least* that large. The conservative error tolerance in the compressed waveform is chosen to be significantly smaller out of concern for the archival integrity of the data.

The NINJA2 project [31, 265] created one of the earliest instances of a collection of waveforms from various numerical relativity groups, including 'hybridized' data extended with post-Newtonian waveforms. One need that was identified was for a common format that could drastically reduce the size of the data, so that waveforms could be distributed and analyzed more easily. Specifically, because of the varied requirements of the many groups involved, the format needed to be based on plain text files, eliminating the possibility of applying standard compression algorithms. Instead, the `GridRefinement` code [266] was introduced to reduce the number of time steps stored, while still allowing the original data to be reconstructed to within a specified tolerance. The essential idea was to decompose the waveform modes into amplitude and 'unwrapped' phase (with branch-cut discontinuities removed through the addition of multiples of 2π), then reduce the number of time steps to store such that the original data could be reconstructed to within a specified tolerance via *linear* interpolation. Linear interpolation was chosen only for simplicity; restrictions of the project required that the code be self-contained and simple enough that it would not require review. By default, the tolerances were 10^{-5} for the (instantaneous) relative amplitude error and the absolute phase error. The

algorithm broke up each time series into one or more time intervals. First, the entire time series was chosen as a single interval. If the data could not be reconstructed to within the specified tolerance, the midpoint between the first and last time steps was included, and the process repeated on each of the two new intervals, applied recursively until the data could be adequately reconstructed. The amplitude-phase decomposition was chosen because that early work only considered non-precessing systems, which—because of their symmetry—exhibit smooth variations in amplitude and phase. This would no longer be as useful for the precessing systems in more modern catalogs. Nonetheless, for the waveforms produced for the NINJA-2 project, this approach reduced the size of waveform files by 'anywhere from a few percent for short numerical data to 99% for very long hybrid waveforms.' [267]

The LVCNR format [268] extends the NINJA-2 format by using spline interpolation instead of linear. While it retains the amplitude-phase decomposition, it includes an option to store real and imaginary parts if the result would be smaller. It also uses a different tolerance, error measure, and algorithm [269] to determine the time steps to store. This format is still used by much of the LVK Collaboration for exchanging waveforms. For comparison, when loosening the tolerance of our new RPDMB format (described below in section C.2) to achieve the same accuracy as the default LVCNR values (which is achieved using $\tau = 10^{-5}$), files created by LVCNR—even after eliminating all but the essential waveform data—are roughly 15 times larger than those created by RPDMB[24].

A notable feature of these lossy formats is that some data points are simply dropped from the data, while those that remain are stored in unaltered form. This stands in contrast to the RPDMB format, which stores all data points, but modifies them for more effective compression.

C.2. Compressing in RPDMB format

Here, we outline in more detail the steps involved in converting data to the new RPDMB format. We considered and took inspiration from a number of sources dealing with compression of data [270–276]. Note that the complete process is implemented as the `sxs.rpdmb.save` function, while `sxs.rpdmb.load` will load the data back into the original form.

RPDMB stands for `rotating_paired_diff_multishuffle_bzip2`, which describes each element of the format *per se*. However, we include two additional steps in the process of converting waveforms to this format: truncation and adding zero—the first of which is probably the single most important operation for actually reducing the size of the file. The following are all the steps of the conversion process, in order.

C.2.1. Corotating frame. The older waveform formats mentioned in section C.1 decomposed the waveform into amplitude and phase to take advantage of the fact that these quantities vary

[24] This is true for LVCNR files without the metadata—such as spins and orbital elements as functions of time—and with the `slim = True` option passed to eliminate the extraneous 'error' data, so that *only* the minimal set of waveform data are included in the LVCNR file. Also note that one limitation of the reference implementation of the LVCNR format is that it scales poorly—typically as the cube of the number of time steps in the waveform. Because the waveforms presented here are relatively long compared to those the LVCNR format was designed for, converting just the public waveforms presented in this catalog would take an estimated 20 000 CPU-hours. Using an algorithm closer to the one used by the `GridRefinement` code, we can achieve equivalent results about 4000 times faster. This improvement has been implemented in `sxs.utilities.lvcnr`, but is not used for these comparisons to ensure fidelity to the reference implementation, though the results would be essentially identical.

on a secular timescale for non-precessing systems, unlike the real and imaginary parts which vary on the orbital timescale. For more general systems, the phase in particular can vary almost discontinuously in time, making this a poor choice. However, even for precessing systems we can factor out the sinusoidal dependence of the real and imaginary parts of the waveform modes by transforming to a corotating frame. This is defined as a rotating frame in which the time dependence of the waveform modes is minimized [183]. The angular velocity of this frame can be easily computed from the modes themselves, and that velocity can then be integrated to provide the frame as a function of time [277]. The waveform is then transformed into that frame. The entire procedure is implemented as the `to_corotating_frame` method to be applied to `sxs.WaveformModes` objects.

This has the effect of making the real and imaginary parts of the waveform modes vary on a slower timescale; for non-precessing systems, they vary at the same rate as the amplitude. However, we have also introduced another piece of data that we need to store for each waveform: the frame itself. The calculations use a quaternion representation, in which the frame is represented by four numbers at each instant of time. However, these numbers are not independent; the sum of their squares must be 1. Instead, we can use a more compact representation: the logarithm of the quaternion. This is a three-vector at each instant of time, representing the *generator* of the rotation (roughly its axis-angle form), rather than the rotation itself. The original rotation can be reconstructed exactly simply by exponentiating the generator, as implemented in the `quaternionic` package [278], and the waveform rotated back to the inertial frame, as implemented in the `spherical` package [279].

C.2.2. Paired modes. In the corotating frame, the modes of a non-precessing system will vary on a secular timescale, but the modes of a precessing system will still vary on the orbital timescale. This is because of spin-orbit coupling causing an asymmetry in the emission of GWs across the orbital plane. We can further factor out the asymmetric waves into symmetric and antisymmetric parts, each of which again varies on a secular timescale [280]. This is done simply by combining mode (ℓ, m) with the conjugate of mode $(\ell, -m)$, with an appropriate sign. For simplicity, we define the sum and difference:

$$s^{\ell,m} = \frac{h^{\ell,m} + \bar{h}^{\ell,-m}}{\sqrt{2}}, \qquad d^{\ell,m} = \frac{h^{\ell,m} - \bar{h}^{\ell,-m}}{\sqrt{2.}} \tag{C.1}$$

Which of these is symmetric and which is antisymmetric depends on the value of ℓ, but is irrelevant for our purposes; we only care that both are slowly varying. We then define the collective quantity

$$f^{\ell,m} = \begin{cases} d^{\ell,-m} & m < 0, \\ h^{\ell,0} & m = 0, \\ s^{\ell,m} & m > 0. \end{cases} \tag{C.2}$$

Note that the transformation from h to f is reversible, up to machine precision, via

$$h^{\ell,m} = \begin{cases} \frac{\bar{f}^{\ell,-m} - \bar{f}^{\ell,m}}{\sqrt{2}} & m < 0, \\ f^{\ell,0} & m = 0, \\ \frac{f^{\ell,m} + f^{\ell,-m}}{\sqrt{2}} & m > 0. \end{cases} \tag{C.3}$$

It is implemented as the `convert_to_conjugate_pairs` method in the `sxs` package, and the resulting $f^{\ell,m}$ is passed on to the next step.

C.2.3. Truncation. As is standard in computation, SpEC relies primarily on 64-bit floating-point numbers to represent the data—specifically, the IEEE 754 `binary64` format—which is accurate to a relative precision of roughly 10^{-16}. However, due to the nature of numerical evolutions, the results of SpEC simulations are generally accurate to significantly fewer bits. Thus, many of the lowest-significance bits in the waveform are effectively random. This means that they are essentially incompressible, and yet contribute no useful information. By discarding this randomness in some way, we can improve the overall compression of the waveforms without any real loss to the information content of the data.

The most obvious approach would be to simply use a numerical representation with lower precision—for example by using `binary32` instead of `binary64`. However, this would be a very coarse approach, with no flexibility to adapt to the data, and uniform *relative* precision (around 10^{-7} for `binary32`) for all modes at all times. For example, in an equal-mass non-precessing system, the largest modes will generally be $(\ell, m) = (2, \pm 2)$, while the $(8, \pm 1)$ modes are typically 9 orders of magnitude smaller. There would seem to be little reason to store the $(8, \pm 1)$ modes to a precision of 10^{-7}, when even its most significant digits are smaller than the least significant digits of the $(2, \pm 2)$ modes. Moreover, the *relative* magnitude of various modes will generally change significantly over the course of a simulation. These points suggest two important criteria for truncation:

(i) different modes should have different (relative) precision, and
(ii) the precision should depend on time.

Neither of these are satisfied by a simple change of floating-point format.

The HDF5 specification [281] includes 'N-Bit' and 'scale-offset' filters to provide more precise control over the precision of the data. The N-Bit filter allows the user to specify the number of bits of the data to be stored, which in principle would allow us to simply ignore bits below a certain significance threshold. The scale-offset filter allows the user to specify an *absolute* precision, below which the data will be rounded to zero. However, both of these features must be specified on a per-dataset basis, leaving no possibility to adapt to time dependence in the data.

We can easily implement a generalization of these filters that allows us to specify a time-dependent precision. Using the fact that our floating-point numbers are specified in binary form, we can multiply by an appropriate power of 2, then round the result to the nearest integer, and then divide by the same power of 2. This is equivalent to setting bits in the binary representation of the number to zero below a certain significance[25]. We can compute the L^2 norm n_i of the field at each time t_i. Then, with a tolerance δ *relative to that norm*, we find the value of the least-significant bit greater than or equal to δn_i, and construct p_i, the smallest power of 2 such that p_i times that bit's value will be greater than or equal to 1. We then multiply each mode by p_i and round the result to the nearest integer, then divide again by p_i to get the truncated number:

$$n_i = \sqrt{\int_{S^2} |h(t_i, \theta, \phi)|^2 \, \mathrm{d}\Omega} = \sqrt{\sum_{\ell,m} |h^{\ell,m}(t_i)|^2} = \sqrt{\sum_{\ell,m} |f^{\ell,m}(t_i)|^2}, \tag{C.4}$$

$$p_i = 2^{\lfloor -\log_2(\delta n_i) \rfloor}, \tag{C.5}$$

[25] We assume that the data are sufficiently well behaved that we can ignore any subtleties with subnormal numbers or infinities.

$$\mathring{f}^{\ell,m}(t_i) = \frac{\text{round}\left(f^{\ell,m}(t_i)\, p_i\right)}{p_i}. \tag{C.6}$$

Because we have used an exact power of 2, the division in the last line will be exact in `binary64`, with the result that the binary representation of $\mathring{f}^{\ell,m}(t_i)$ will have all bits below the significance threshold set to zero. Those zeros can be compressed very effectively, especially after application of the 'multishuffle' step described below. This is implemented as the `truncate` method in the `sxs` package.

Here, δ describes the worst-case error in each component (real or imaginary) of each mode of the $f^{\ell,m}$ data at each instant of time. Denoting the number of modes as N_{modes}, the worst-case error in the *total waveform* at each instant will obey

$$\sqrt{\sum_{\ell,m} \left| h^{\ell,m}(t_i) - \mathring{h}^{\ell,m}(t_i) \right|^2} \leqslant \delta \sqrt{N_{\text{modes}}} \tag{C.7}$$

Thus, if we have in mind some total error tolerance τ, we need to set the per-component tolerance δ as

$$\delta = \frac{\tau}{\sqrt{N_{\text{modes}}}} \tag{C.8}$$

For the waveforms published as part of this catalog, we chose $\tau = 10^{-10}$, limiting the individual error at each instant of time in each component of each of the 77 modes to roughly 1.14×10^{-11} times the norm of the waveform at that instant. This choice was made simply by plotting the amplitude of all modes and judging the level at which numerical errors become visually obvious, exhibiting discontinuities and noise. We expect that this choice is several orders of magnitude more conservative than necessary, because of the numerous and cumulative errors inherent in evolving numerical-relativity data. Nonetheless, to ensure the archival quality of the data, we chose to err on the side of caution.

The particular choice of τ has a significant effect on the compression ratio. For example, we can examine the effect of changing τ on the compression ratio for a particular waveform. Figure C2 shows the compression ratio as a function of τ for the strain from SXS:BBH:2265, which was the last simulation in the previous SXS catalog. A compression ratio of 1 would be equivalent to storing the exact bytes of the data. The horizontal dashed line shows the compression ratio, 1.3x, when storing in HDF5 using the best available standard options, but without otherwise altering the data in any way. The solid blue curve shows the compression ratio achieved by RPDMB using various levels of truncation. Note that even with $\tau = 10^{-16}$, *smaller* than machine precision, RPDMB can still achieve better compression than HDF5 because there are modes with significant bits smaller than machine precision relative to the norm of the waveform; these are truncated, resulting in a reduction in compressed file size.

As the tolerance is increased, the compression ratio increases. For this particular waveform, a compression ratio of 8.6x is achieved with the default value of $\tau = 10^{-10}$ used in the public data. However, with less stringent tolerances, the compression ratio grows drastically. In particular, this growth is much faster than one would expect simply from counting the number of nonzero bits to be stored—ranging from a factor of 2 improvement at small τ to an order of magnitude at large τ. Presumably, this is because the bits being zeroed out are effectively random and therefore cannot be compressed, whereas the remaining bits are increasingly continuous and compress relatively well.

The red 'x' shows the compression factor, 17x, achieved by the LVCNR format with its default settings, including only the time and waveform modes, and excluding the various spin

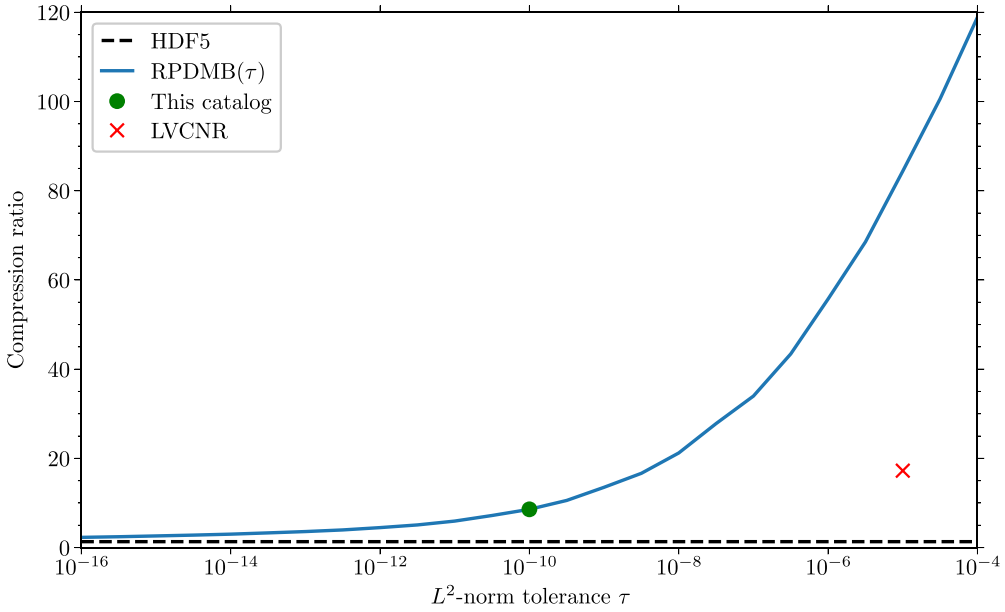

Figure C2. Compression ratio relative to the size of the raw data (waveform modes and time) as a function of tolerance τ for the strain from SXS:BBH:2265, which was the last simulation in the previous SXS catalog. The horizontal dashed line shows the compression ratio 1.3x when using standard HDF5 storage, storing one chunk per real time series, applying the shuffle and gzip filters. The blue curve shows the compression ratio achieved by RPDMB using various levels of truncation τ, with the green dot emphasizing the tolerance $\tau = 10^{-10}$ used for the data described by this paper—the compression ratio for this particular waveform being 8.6x. The red 'x' shows the compression ratio 17x achieved by the LVCNR format, excluding metadata and other non-waveform information. Note that the LVCNR tolerance is specified as 10^{-6}, but uses a different error measure; we compare it here to $\tau = 10^{-5}$, which is sufficient for RPDMB to reproduce the waveform to higher accuracy by either error measure with a compression ratio of 84x.

and orbital time series and other metadata. By default, the LVCNR format specifies a tolerance of 10^{-6}, where the error is given as the maximum of the absolute value of the difference between the original data and the data reconstructed by a degree-5 interpolating spline [269], applied separately to the amplitude and phase of each mode or the real and imaginary parts of each mode—whichever one results in a smaller set of points. This differs from the error measure used in this Paper, as we compute the error at each instant of time rather than taking the maximum, and measure it relative to the norm of the waveform. If the norm at time t_i is n_i, then we can achieve a comparable accuracy to the LVCNR format with a tolerance of $\tau \approx 10^{-6}/n_i$. By choosing $\tau = 10^{-6}/n_{\max}$, where $n_{\max}$ is the maximum norm of the waveform, we can ensure that the error in the RPDMB waveform is always less than the error in the LVCNR waveform by *both* error measures. For the waveform shown in figure C2, this corresponds to $\tau = 10^{-5}$. The actual compression ratio achieved by LVCNR with its default tolerance is shown at this value of τ as a red 'x' mark, with a value of 17x. For comparison, RPDMB achieves a compression ratio of 84x at this tolerance, while achieving smaller errors by either measure.

C.2.4. Adding zero. At this stage, many of the paired modes will be precisely zero. Specifically, for non-precessing simulations, the amplitude of half of the mode pairs should zero by symmetry; they will generally not be exactly zero due to numerical error, but after truncation they likely will be. However, the standard floating-point representation of zero is not unique; floating-point zeros can have either sign. Because those zeros were generated by—effectively—random noise, their signs will also be effectively random. This adds a great deal of entropy to the data, making it difficult to compress.

We can improve the compression by setting all zero-valued data points to a unique representation of zero. Fortunately, the IEEE floating-point standard [282] specifies that adding two zeros of either sign will result in a positive zero (by default). Thus, we can simply add zero to all the data to eliminate the sign ambiguity. Including this step reduces the total size of the waveform data by an average of 0.5%. Though this is a tiny improvement, it is achieved by a trivial operation, which can be combined with the truncation step above at essentially no cost. It should be noted that this is almost surely only beneficial because of the truncation step and the presence of very small modes. As more modes are included, there will be more very small modes, making them more likely to be truncated to zero, and thus making this step more beneficial—though likely always quite small.

C.2.5. Differencing sequential data points. For reasonably continuous data, as in a time series, we can improve compression by storing the first data point as is, but thereafter only the differences between sequential data points. Specifically, we store $h^{\ell,m}(t_0)$ as is, and then store $h^{\ell,m}(t_j) - h^{\ell,m}(t_{j-1})$ for all $j > 0$. Note that subtraction here is based on floating-point numbers; we could also reinterpret the numbers as 64-bit integers and perform integer subtraction, which would lead to different patterns in the resulting bits. A closely related procedure reinterprets the numbers as *unsigned* 64-bit integers, then uses `XOR` in place of subtraction. Note that the `XOR` operation is precisely invertible, whereas floating-point subtraction can accrue roundoff-level errors. For data that have already been subject to truncation those are negligible, whereas `XOR` is required for data that have not been truncated[26].

We have tested the effectiveness of both forms of differences and of `XOR` when applied to the waveform data. The results are similar but—at least in combination with all other steps described here—floating-point differencing achieves 37% better compression than integer differencing and 60% better compression than `XOR` when storing the real and imaginary parts of the waveform modes and the components of the logarithm of the rotation quaternion. However, to retain lossless compression for the time data, we use `XOR` for that part of the data.

C.2.6. Multishuffle. The HDF5 library includes a 'shuffle' filter [264] that reorders the bytes of the data to improve compression for continuous data. Conceptually, we can imagine storing a series of numbers

$$12345, 12346, 12347, 12348.$$

If each numeral is stored in its obvious order—the order in which it appears above—the pattern-finding techniques underlying many compression algorithms will not be able to take

[26] These are both well known techniques, used in many compression algorithms [270–276]. Sometimes referred to as 'delta encoding', both can be considered simple cases of the more general technique of 'predictive encoding', which uses other data points to predict a given value, and then stores only the difference between the prediction and the true value.

advantage of the fact that the numbers are very similar. However, if the numbers are stored as

$$11112222333344445678,$$

those longer runs of repeated digits can be encoded more efficiently. This is the basic idea behind the shuffle filter. When applied to much longer sequences of 64-bit numbers, the effect can be quite dramatic. Specifically, the 'shuffle' filter takes the most-significant byte (8 bits) of each number and stores them consecutively, followed by the second-most-significant byte, and so on. Another way to think of this is to consider a series of N 64-bit numbers as a matrix of N rows and 8 columns, where each column represents a byte of a number. The obvious storage scheme would be to go across each row, then down to the next row, and so on. The shuffle filter is essentially a transpose—it goes down all rows in the first column, then the second column, and so on.

It might be helpful to describe this more specifically as a 'byteshuffle' filter, because this approach can be generalized to put it on a spectrum of similar filters. Reference [275] introduced the 'bitshuffle' filter, which takes the most-significant bit of each number and stores them consecutively, followed by the second-most-significant bit, and so on. Again, we could think of this as a matrix transpose, except that the columns now are individual bits. We can easily imagine using different divisions also. The 'nibbleshuffle' would use groups of 4 bits (nibbles), and the 'morselshuffle' would use groups of 2 bits. (We will see below that groups of sizes that are not powers of 2 are strongly disfavored.)

Each of these options has its advantages and disadvantages, depending on the scale across which the bits can be expected to vary coherently. We would expect byteshuffle to be the best choice when consecutive bytes correlate well with each other; bitshuffle would be best when that correlation fails at the byte level but holds at the bit level. However, because of the nature of the `binary64` format, where different bits have different significance, we would expect the optimal choice to vary along the length of the 64 bits. This suggests that the best choice could be to vary the width of the groups of bits that we shuffle as we progress along the number. We call this 'multishuffle'; it is implemented via the function of the same name in the `sxs` package.

A multishuffle filter must be specified by the widths of the groups of bits that are shuffled. For example, if we group the first byte of each number, followed by the next 4 bits, followed by the next 2 bits, and then individual bits after that, we would specify the filter as $(8,4,2,1,1,\ldots)$. The sum of all those numbers must be 64. In this representation, no shuffling would be (64), the standard byteshuffle is $(8,\ldots)$, nibbleshuffle is $(4,\ldots)$, morselshuffle is $(2,\ldots)$, and bitshuffle is $(1,\ldots)$.

In general, there are $2^{63} \approx 10^{19}$ possible multishuffle filters for 64-bit data, so the search is not trivial. Because of the discrete nature of the problem, and the enormous size of the space of possibilities, a genetic algorithm seems like a natural choice for finding the best filter. We used the `Evolutionary.jl` package [283] to search for the shuffle widths that delivered the best compression for a random subset of 20 simulations, using each waveform from the various resolutions and extrapolation orders, for a total of 472 waveforms. Any form of shuffling was always better than not shuffling at all, by at least 30%. However, among all other choices tested, the bitshuffle algorithm was the worst, followed by morselshuffle, then the standard byteshuffle, and nibbleshuffle. Nonetheless, it was possible to gain significant improvements by varying the shuffle widths—by 18% to 3% over the various fixed-width options, and in particular about 6.5% over the standard byteshuffle.

At no point throughout the optimization did the genetic algorithm find better results for a multishuffle involving a width that was not a power of 2. This is presumably because the compression algorithm (BZIP2, as described below) is still based on bytes, so combining groups of bits with sizes that are not powers of 2 will not produce alignment along those bytes that can be easily compressed. The algorithm very quickly determined that the first groups should have sizes 16, 4, 2, 2, and 2. Beyond that, the results clearly favored small groups of either 1 or 2 bits, but did not depend very strongly on the exact choice of widths. The widths we have used for the waveform data in this catalog are

$$(16,4,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,4,4,4,4) \quad \text{(C.9)}$$

For `binary64`, the first 16 bits represent the sign, the exponent, and the first 4 bits of the significand. The next 4 bits represent the next 4 bits of the significand, and so on, with all remaining bits representing decreasingly significant bits.

Recalling that the previous step converted the data to successive differences, we are really encoding the *rate of change* of the data between timesteps. In particular, the first block of 16 bits will essentially encode the difference between successive timesteps, rounded to just the first few digits. The rate of change in the modes—at least to this level of accuracy—will typically be quite small, so we might expect that many of these values will be repeated, meaning that run-length encoding will efficiently compress the data. This turns out to be true for the first 8 bits, but not for all 16 bits. Instead, it appears that the benefit of grouping the first 16 bits together is that there are relatively few distinct patterns that appear in the data. Of the $2^{16} = 65\,536$ possible values that could be encoded by the first 16 bits, only 1000 or so actually appear in typical waveforms, with the most common values occurring *far* more often than most. This is ideal for the Huffman-coding [284] stage of compression to represent these values more compactly than the 16 bits would[27]. These features are decreasingly likely to occur as we proceed to bits with lower significance, which is why it makes sense that the shuffle widths decrease as we proceed through the number. On the other hand, there is an increasing likelihood that the bits will switch to being a series of 0 s, which can be compressed very effectively with run-length encoding. We want those runs to occur as soon as possible, which is why it makes sense that widths become 1 as that condition becomes more likely. Finally, the last 4 groups of 4 bits represent the last 16 bits of the significand, which will always be less significant than the 10^{-10} tolerance. Therefore, we can expect that these are just long runs of 0 s for every mode. The genetic algorithm found essentially no difference in any choice of widths for these bits, so we group them together to reduce the number of times the data must be traversed.

It is important to note that the optimal choice of shuffle widths may vary for different datasets; the results we report here are true for our data, after the particular processing steps described above. They may not extend to different types of datasets, or even to the same waveforms when preceded by different processing steps. As such, the precise set of shuffle widths is essentially adjustable, and therefore should be stored as metadata alongside the compressed data. As mentioned below, we choose the HDF5 file format to organize the data, and preserve the shuffle widths as an HDF5 attribute of each dataset. When loading the data, this attribute is read to ensure that the data are decompressed correctly.

One caveat to note about all types of shuffle filters is that they require multiple passes over the data. The number of passes required is set by the length of the width specification: 8 for byteshuffle, 64 for bitshuffle, and 31 for the widths chosen for this catalog. In the naive

[27] Extending beyond 16 bits, this particular advantage disappears; the number of observed values grows with the number of possible values. In that case, it is evidently better to take advantage of other features of the compression.

application, this can lead to memory bottlenecks with numerous cache misses. When this is a problem, the data can be divided into 'chunks' that fit into cache, and the shuffle filter applied to each chunk separately. In particular, the reference implementation of the bitshuffle filter [275] uses this technique. This will reduce the number of cache misses, but will also reduce the effectiveness of the compression stage. Because we find the cost to be minimal compared to the increased burden of disk access due to less-effective compression, we choose to treat each real or imaginary part of each waveform mode as a single chunk to be shuffled as a unit.

C.2.7. BZIP2. None of the preceding stages actually reduce the number of bytes that must be stored. In fact, the conversion to a rotating frame just adds another set of data—the logarithm of the quaternions—to be stored. Instead, each stage has been designed to reduce the entropy of the information in the waveform. The final stage is to use that reduced entropy to compress the data. We arrange the data as a single sequence of bytes, beginning with the shuffled time data, followed by the shuffled modes, and finally the shuffled logarithm of the rotation quaternions.

We have tested a variety of standard compression algorithms to do so. The best results were obtained using BZIP2 [263], which passes the data through a number of stages, including run-length encoding, the Burrows-Wheeler transform [285], the move-to-front transform [286, 287], and Huffman coding [284]. Files created using XZ/LZMA [288] were about 1% larger; about 3% for Brotli [289]; 12% for GZIP [290]; and 13% for Zstd.

The speed of compression varied widely—from an order of magnitude faster for Zstd, to an order of magnitude slower for Brotli. However, the time spent compressing the data is generally a small fraction of the time spent transforming and writing the data, so the speed of compression is not a significant concern. The speed of decompression was less varied, but also dominated entirely by the time spent reading the data from disk—not to mention the time spent transferring the data over the internet. Thus, the dominant factor in choosing a compression algorithm was the size of the compressed file, leaving BZIP2 as the clearly preferred choice.

C.2.8. HDF5 storage. The output of the BZIP2 compression stage is a single byte stream, which could be written to disk directly. However, there are also various pieces of metadata that are important for being able to reliably decompress the data. For example, the shuffle widths used for the multishuffle filter must be stored, as must the number of modes and/or the number of time steps. Perhaps most importantly, to ensure that the files remain useful in the future and that the interface can be easily extended, it is important to store the name of the format used to compress the data. It can also be helpful to organize multiple waveforms into a single file.

The HDF5 file format is well suited to this task, and is widely used in the scientific community for storing large datasets along with metadata [264]. Specifically, we store the data stream as a single 'opaque' dataset named 'data', with no filters of any kind. We then attach 'attributes' to that dataset:

- `sxs_format`
- `shuffle_widths`
- `ell_min`
- `ell_max`
- `n_times`

The first attribute value is `rotating_paired_diff_multishuffle_bzip2`, which allows the interface to automatically detect the correct decompression steps to apply. For all waveforms in this catalog, the shuffle widths are as given in equation (C.9), and the minimum and

maximum values of ℓ are 2 and 8, respectively. The number of time steps is also stored. While this could be inferred from the length of the data after BZIP2 decompression, it is convenient to store it directly, acting as a simple check on the integrity of the data. This dataset and its attributes can be stored in any group of an HDF5 file, whether the root group as for `Strain_N2.h5` files, or in descriptively named subgroups as in the `ExtraWaveforms.h5` files.

C.3. Future work

By including the name of the format within the file itself, we have left open the possibility of changing the format in the future. There may be better choices for the multishuffle widths, or for the compression algorithm used in the final stage. There is almost surely some improvement that could be made in the predictive step. For example, the various modes could be normalized in some way, or make use of post-Newtonian approximations. However, it is not clear that such a change would be worth the effort. Surely the most impactful alteration would be to simply increase the tolerance τ used in the truncation step. Regardless, all of these considerations will remain almost invisible to the user, because the `sxs` interface will be able to detect the format and apply the correct decompression steps automatically.

Appendix D. Simulations with large differences in figure 8

In figure 8 there are long tails at large waveform difference that are caused by a handful of simulations, which we list here for completeness.

For the plots comparing extrapolation orders, the problematic simulations are head-on, nearly-head-on, or scattering cases. These simulations are particularly short and hence a large fraction of the waveform still contains a considerable amount of initial transient junk radiation. The differences between extrapolation orders are always relatively large during these initial transients, but due to the shortness of these particular runs their relative contribution to the waveform difference is larger. Some of these runs also have waveform-extraction radii that are too close together, which causes unusually large extrapolation errors, especially for high extrapolation order. These simulations are:

SXS:BBH:1110	SXS:BBH:3873	SXS:BBH:3876	SXS:BBH:3881
SXS:BBH:1363	SXS:BBH:3874	SXS:BBH:3877	SXS:BBH:3882
SXS:BBH:1544	SXS:BBH:3875	SXS:BBH:3879	SXS:BBH:3883
SXS:BBH:3884	SXS:BBH:3889	SXS:BBH:3996	SXS:BBH:4000
SXS:BBH:3885	SXS:BBH:3890	SXS:BBH:3997	
SXS:BBH:3887	SXS:BBH:3995	SXS:BBH:3999	SXS:BBH:4292

ORCID iDs

Mark A Scheel 0000-0001-6656-9134
Michael Boyle 0000-0002-5075-5116
Keefe Mitman 0000-0003-0276-3856
Nils Deppe 0000-0003-4557-4115
Leo C Stein 0000-0001-7559-9597
Cristóbal Armaza 0000-0002-1791-0743
Marceline S Bonilla 0000-0003-4502-528X
Luisa T Buchman 0000-0003-3428-6003
Andrea Ceja 0000-0002-1681-7299

Himanshu Chaudhary 0000-0002-4101-0534
Yitian Chen 0000-0002-8664-9702
Maxence Corman 0000-0003-2855-1149
Károly Zoltán Csukás 0000-0002-2408-1103
C Melize Ferrus 0000-0002-2842-2067
Scott E Field 0000-0002-6037-3277
Matthew Giesler 0000-0003-2300-893X
Sarah Habib 0000-0002-4725-4978
François Hébert 0000-0001-9009-6955
Dante A B Iozzo 0000-0002-7244-1900
Tousif Islam 0000-0002-3434-0084
Ken Z Jones 0009-0003-1034-0498
Aniket Khairnar 0000-0001-5138-572X
Lawrence E Kidder 0000-0001-5392-7342
Taylor Knapp 0000-0001-8474-4143
Prayush Kumar 0000-0001-5523-4603
Guillermo Lara 0000-0001-9461-6292
Oliver Long 0000-0002-3897-9272
Geoffrey Lovelace 0000-0002-7084-1070
Sizheng Ma 0000-0002-4645-453X
Denyz Melchor 0000-0002-7854-1953
Marlo Morales 0000-0002-0593-4318
Jordan Moxon 0000-0001-9891-8677
Peter James Nee 0000-0002-2362-5420
Kyle C Nelli 0000-0003-2426-8768
Serguei Ossokine 0000-0002-2579-1246
Robert Owen 0000-0002-1511-4532
Harald P Pfeiffer 0000-0001-9288-519X
Isabella G Pretto 0009-0001-7552-551X
Teresita Ramirez-Aguilar 0000-0003-0994-115X
Antoni Ramos-Buades 0000-0002-6874-7421
Adhrit Ravichandran 0000-0002-9589-3168
Abhishek Ravishankar 0009-0006-6519-8996
Samuel Rodriguez 0000-0002-1879-8810
Hannes R Rüter 0000-0002-3442-5360
Jennifer Sanchez 0000-0002-5335-4924
Md Arif Shaikh 0000-0003-0826-6164
Dongze Sun 0000-0003-0167-4392
Daniel Tellez 0009-0008-7784-2528
Saul A Teukolsky 0000-0001-9765-4526
Sierra Thomas 0000-0003-3574-2090
William Throwe 0000-0001-5059-4378
Vijay Varma 0000-0002-9994-1761
Nils L Vu 0000-0002-5767-3949
Marissa Walker 0000-0002-7176-6914
Nikolas A Wittek 0000-0001-8575-5450
Jooheon Yoo 0000-0002-3251-0924

References

[1] Abbott B P *et al* (LIGO Scientific and Virgo) 2016 *Phys. Rev. Lett.* **116** 061102
[2] Abbott B P *et al* (LIGO Scientific and Virgo) 2016 *Phys. Rev. D* **93** 122003
[3] Abbott B P *et al* (LIGO Scientific and Virgo) 2016 *Phys. Rev. Lett.* **116** 131103
[4] Abbott B P *et al* (LIGO Scientific and Virgo) 2016 *Phys. Rev. Lett.* **116** 241102
[5] Aasi J *et al* (LIGO Scientific) 2015 *Class. Quantum Grav.* **32** 074001
[6] Acernese F *et al* (Virgo) 2015 *Class. Quantum Grav.* **32** 024001
[7] Abbott B P *et al* (LIGO Scientific and Virgo) 2019 *Phys. Rev. X* **9** 031040
[8] Abbott R *et al* (LIGO Scientific and Virgo) 2021 *Phys. Rev. X* **11** 021053
[9] Abbott R *et al* (KAGRA, VIRGO and LIGO Scientific) 2023 *Phys. Rev. X* **13** 011048
[10] Abbott R *et al* (KAGRA, VIRGO and LIGO Scientific) 2023 *Phys. Rev. X* **13** 041039
[11] Nitz A H, Kumar S, Wang Y F, Kastha S, Wu S, Schäfer M, Dhurkunde R and Capano C D 2023 *Astrophys. J.* **946** 59
[12] Lorentz H A and and Droste J 1917 *Versl. K. Akad. Wet. Amsterdam* **26** 392–403, 649–60
[13] Einstein A, Infeld L and Hoffmann B 1938 *Ann. Math.* **39** 65–100
[14] Damour T and Deruelle N 1985 *Ann. Inst. Henri Poincaré* **43** 107
[15] Damour T and Deruelle N 1986 *Ann. Inst. Henri Poincaré* **44** 263
[16] Blanchet L 2014 *Living Rev. Relativ.* **17** 2
[17] Rothstein I Z 2014 *Gen. Relativ. Gravit.* **46** 1726
[18] Porto R A 2016 *Phys. Rep.* **633** 1–104
[19] Schäfer G and Jaranowski P 2018 *Living Rev. Relativ.* **21** 7
[20] Levi M 2020 *Rep. Prog. Phys.* **83** 075901
[21] Futamase T and Itoh Y 2007 *Living Rev. Relativ.* **10** 2
[22] Blanchet L 2011 *Fundam. Theor. Phys.* **162** 125–66
[23] Schaefer G 2011 *Fundam. Theor. Phys.* **162** 167–210
[24] Foffa S and Sturani R 2014 *Class. Quantum Grav.* **31** 043001
[25] Westpfahl K and Goller M 1979 *Lett. Nuovo Cim.* **26** 573–6
[26] Westpfahl K and Hoyler H 1980 *Lett. Nuovo Cim.* **27** 581–5
[27] Pretorius F 2005 *Phys. Rev. Lett.* **95** 121101
[28] Campanelli M, Lousto C O, Marronetti P and Zlochower Y 2006 *Phys. Rev. Lett.* **96** 111101
[29] Baker J G, Centrella J, Choi D-I, Koppitz M and van Meter J 2006 *Phys. Rev. Lett.* **96** 111102
[30] Aylott B *et al* 2009 *Class. Quantum Grav.* **26** 114008
[31] Ajith P *et al* 2012 *Class. Quantum Grav.* **29** 124001
Ajith P *et al* 2013 *Class. Quantum Grav.* **30** 199401 (erratum)
[32] Hinder I *et al* 2014 *Class. Quantum Grav.* **31** 025012
[33] Jani K, Healy J, Clark J A, London L, Laguna P and Shoemaker D 2016 *Class. Quantum Grav.* **33** 204001
[34] Ferguson D *et al* 2025 *Phys. Rev. D* **112** 044043
[35] Healy J, Lousto C O, Zlochower Y and Campanelli M 2017 *Class. Quantum Grav.* **34** 224001
[36] Healy J, Lousto C O, Lange J, O'Shaughnessy R, Zlochower Y and Campanelli M 2019 *Phys. Rev. D* **100** 024021
[37] Healy J and Lousto C O 2020 *Phys. Rev. D* **102** 104018
[38] Healy J and Lousto C O 2022 *Phys. Rev. D* **105** 124010
[39] Huerta E A *et al* 2019 *Phys. Rev. D* **100** 064003
[40] Hamilton E *et al* 2024 *Phys. Rev. D* **109** 044032
[41] Rashti A, Gamba R, Chandra K, Radice D, Daszuta B, Cook W and Bernuzzi S 2025 *Phys Rev. D* **111** 104078
[42] Lange J *et al* 2017 *Phys. Rev. D* **96** 104041
[43] Buonanno A and Damour T 1999 *Phys. Rev. D* **59** 084006
[44] Buonanno A and Damour T 2000 *Phys. Rev. D* **62** 064015
[45] Buonanno A, Chen Y and Damour T 2006 *Phys. Rev. D* **74** 104005
[46] Damour T, Jaranowski P and Schaefer G 2000 *Phys. Rev. D* **62** 084011
[47] Damour T 2001 *Phys. Rev. D* **64** 124013
[48] Buonanno A and Sathyaprakash B S 2014 *Sources of Gravitational Waves: Theory and Observations*
[49] Damour T 2008 *Int. J. Mod. Phys. A* **23** 1130–48

[50] Ramos-Buades A, Buonanno A, Estellés H, Khalil M, Mihaylov D P, Ossokine S, Pompili L and Shiferaw M 2023 *Phys. Rev. D* **108** 124037
[51] Khan S, Husa S, Hannam M, Ohme F, Pürrer M, Jiménez Forteza X and Bohé A 2016 *Phys. Rev. D* **93** 044007
[52] Husa S, Khan S, Hannam M, Pürrer M, Ohme F, Jiménez Forteza X and Bohé A 2016 *Phys. Rev. D* **93** 044006
[53] Hannam M, Schmidt P, Bohé A, Haegel L, Husa S, Ohme F, Pratten G and Pürrer M 2014 *Phys. Rev. Lett.* **113** 151101
[54] Pratten G *et al* 2021 *Phys. Rev. D* **103** 104056
[55] Pratten G, Husa S, Garcia-Quiros C, Colleoni M, Ramos-Buades A, Estelles H and Jaume R 2020 *Phys. Rev. D* **102** 064001
[56] Ajith P *et al* 2007 *Class. Quantum Grav.* **24** S689–700
[57] Ajith P *et al* 2008 *Phys. Rev. D* **77** 104017
Ajith P *et al* 2009 *Phys. Rev. D* **79** 129901 (erratum)
[58] Ajith P *et al* 2011 *Phys. Rev. Lett.* **106** 241101
[59] Santamaria L *et al* 2010 *Phys. Rev. D* **82** 064016
[60] London L, Khan S, Fauchon-Jones E, García C, Hannam M, Husa S, Jiménez-Forteza X, Kalaghatgi C, Ohme F and Pannarale F 2018 *Phys. Rev. Lett.* **120** 161102
[61] Khan S, Chatziioannou K, Hannam M and Ohme F 2019 *Phys. Rev. D* **100** 024059
[62] Khan S, Ohme F, Chatziioannou K and Hannam M 2020 *Phys. Rev. D* **101** 024056
[63] Dietrich T *et al* 2019 *Phys. Rev. D* **99** 024029
[64] Dietrich T, Samajdar A, Khan S, Johnson-McDaniel N K, Dudi R and Tichy W 2019 *Phys. Rev. D* **100** 044003
[65] Thompson J E, Fauchon-Jones E, Khan S, Nitoglia E, Pannarale F, Dietrich T and Hannam M 2020 *Phys. Rev. D* **101** 124059
[66] García-Quirós C, Colleoni M, Husa S, Estellés H, Pratten G, Ramos-Buades A, Mateu-Lucena M and Jaume R 2020 *Phys. Rev. D* **102** 064002
[67] García-Quirós C, Husa S, Mateu-Lucena M and Borchers A 2021 *Class. Quantum Grav.* **38** 015006
[68] Blackman J, Field S E, Scheel M A, Galley C R, Hemberger D A, Schmidt P and Smith R 2017 *Phys. Rev. D* **95** 104023
[69] Blackman J, Field S E, Scheel M A, Galley C R, Ott C D, Boyle M, Kidder L E, Pfeiffer H P and Szilágyi B 2017 *Phys. Rev. D* **96** 024058
[70] Varma V, Field S E, Scheel M A, Blackman J, Gerosa D, Stein L C, Kidder L E and Pfeiffer H P 2019 *Phys. Rev. Res.* **1** 033015
[71] Blackman J, Field S E, Galley C R, Szilágyi B, Scheel M A, Tiglio M and Hemberger D A 2015 *Phys. Rev. Lett.* **115** 121102
[72] Varma V, Field S E, Scheel M A, Blackman J, Kidder L E and Pfeiffer H P 2019 *Phys. Rev. D* **99** 064045
[73] Reitze D *et al* 2019 *Bull. Am. Astron. Soc.* **51** 035
[74] Evans M *et al* 2023 arXiv:2306.13745
[75] Punturo M *et al* 2010 *Class. Quantum Grav.* **27** 194002
[76] Maggiore M *et al* (ET) 2020 *J. Cosmol. Astropart. Phys.* JCAP03(2020)050
[77] Abac A *et al* 2025 arXiv:2503.12263
[78] Amaro-Seoane P *et al* 2017 arXiv:1702.00786
[79] Luo J *et al* (TianQin) 2016 *Class. Quantum Grav.* **33** 035010
[80] Taiji Scientific Collaboration *et al* 2021 *Commun. Phys.* **4** 34
[81] Kawamura S *et al* 2006 *Class. Quantum Grav.* **23** S125–32
[82] Ajith P *et al* 2025 *J. Cosmol. Astropart. Phys.* JCAP01(2025)108
[83] Pürrer M and Haster C-J 2020 *Phys. Rev. Res.* **2** 023151
[84] Afshordi N *et al* (LISA Consortium Waveform Working Group) 2023 arXiv:2311.01300
[85] MacDonald I, Nissanke S and Pfeiffer H P 2011 *Class. Quantum Grav.* **28** 134002
[86] Boyle M 2011 *Phys. Rev. D* **84** 064013
[87] MacDonald I, Mroue A H, Pfeiffer H P, Boyle M, Kidder L E, Scheel M A, Szilagyi B and Taylor N W 2013 *Phys. Rev. D* **87** 024009
[88] Sadiq J, Zlochower Y, O'Shaughnessy R and Lange J 2020 *Phys. Rev. D* **102** 024012
[89] Mitman K *et al* 2021 *Phys. Rev. D* **104** 024051
[90] Mitman K *et al* 2022 *Phys. Rev. D* **106** 084029

[91] Sun D, Boyle M, Mitman K, Scheel M A, Stein L C, Teukolsky S A and Varma V 2024 *Phys. Rev. D* **110** 104076
[92] Boyle M, Mitman K, Scheel M and Stein L 2025 The sxs package (available at: https://zenodo.org/doi/10.5281/zenodo.15412737)
[93] Boyle M *et al* 2019 *Class. Quantum Grav.* **36** 195006
[94] Driesse M, Jakobsen G U, Mogull G, Plefka J, Sauer B and Usovitsch J 2024 *Phys. Rev. Lett.* **132** 241402
[95] Cheung M H Y, Berti E, Baibhav V and Cotesta R 2024 *Phys. Rev. D* **109** 044069
Cheung M H Y, Berti E, Baibhav V and Cotesta R 2024 *Phys. Rev. D* **110** 049902 (erratum)
[96] Jakobsen G U, Mogull G, Plefka J and Sauer B 2023 *Phys. Rev. Lett.* **131** 241402
[97] Siegel H, Isi M and Farr W M 2023 *Phys. Rev. D* **108** 064008
[98] van de Meent M, Buonanno A, Mihaylov D P, Ossokine S, Pompili L, Warburton N, Pound A, Wardell B, Durkan L and Miller J 2023 *Phys. Rev. D* **108** 124038
[99] Khalil M, Buonanno A, Estelles H, Mihaylov D P, Ossokine S, Pompili L and Ramos-Buades A 2023 *Phys. Rev. D* **108** 124036
[100] Shaikh M A, Varma V, Pfeiffer H P, Ramos-Buades A and van de Meent M 2023 *Phys. Rev. D* **108** 104007
[101] Damour T and Rettegno P 2023 *Phys. Rev. D* **107** 064051
[102] Jakobsen G U and Mogull G 2023 *Phys. Rev. D* **107** 044033
[103] Ma S, Mitman K, Sun L, Deppe N, Hébert F, Kidder L E, Moxon J, Throwe W, Vu N L and Chen Y 2022 *Phys. Rev. D* **106** 084036
[104] Payne E, Hourihane S, Golomb J, Udall R, Udall R, Davis D and Chatziioannou K 2022 *Phys. Rev. D* **106** 104017
[105] Calderon Bustillo J, Sanchis-Gual N, Leong S H W, Chandra K, Torres-Forne A, Font J A, Herdeiro C, Radu E, Wong I C F and Li T G F 2023 *Phys. Rev. D* **108** 123020
[106] Khalil M, Buonanno A, Steinhoff J and Vines J 2022 *Phys. Rev. D* **106** 024042
[107] Islam T, Field S E, Hughes S A, Khanna G, Varma V, Giesler M, Scheel M A, Kidder L E and Pfeiffer H P 2022 *Phys. Rev. D* **106** 104025
[108] Varma V, Biscoveanu S, Islam T, Shaik F H, Haster C-J, Isi M, Farr W M, Field S E and Vitale S 2022 *Phys. Rev. Lett.* **128** 191102
[109] Radia M, Sperhake U, Drew A, Clough K, Figueras P, Lim E A, Ripley J L, Aurrekoetxea J C, França T and Helfer T 2022 *Class. Quantum Grav.* **39** 135006
[110] Magaña Zertuche L *et al* 2022 *Phys. Rev. D* **105** 104015
[111] Moxon J, Scheel M A, Teukolsky S A, Deppe N, Fischer N, Hébert F, Kidder L E and Throwe W 2023 *Phys. Rev. D* **107** 064013
[112] Oshita N 2021 *Phys. Rev. D* **104** 124032
[113] Hamilton E, London L, Thompson J E, Fauchon-Jones E, Hannam M, Kalaghatgi C, Khan S, Pannarale F and Vano-Vinuales A 2021 *Phys. Rev. D* **104** 124027
[114] Khalil M, Buonanno A, Steinhoff J and Vines J 2021 *Phys. Rev. D* **104** 024046
[115] Steinhoff J, Hinderer T, Dietrich T and Foucart F 2021 *Phys. Rev. Res.* **3** 033129
[116] Liu X, Cao Z and Zhu Z-H 2022 *Class. Quantum Grav.* **39** 035009
[117] Estellés H, Husa S, Colleoni M, Keitel D, Mateu-Lucena M, García-Quirós C, Ramos-Buades A and Borchers A 2022 *Phys. Rev. D* **105** 084039
[118] Mitman K *et al* 2021 *Phys. Rev. D* **103** 024031
[119] Mitman K, Moxon J, Scheel M A, Teukolsky S A, Boyle M, Deppe N, Kidder L E and Throwe W 2020 *Phys. Rev. D* **102** 104007
[120] Moxon J, Scheel M A and Teukolsky S A 2020 *Phys. Rev. D* **102** 044052
[121] van de Meent M and Pfeiffer H P 2020 *Phys. Rev. Lett.* **125** 181101
[122] Akcay S, Gamba R and Bernuzzi S 2021 *Phys. Rev. D* **103** 024014
[123] Matas A *et al* 2020 *Phys. Rev. D* **102** 043023
[124] Estellés H, Ramos-Buades A, Husa S, García-Quirós C, Colleoni M, Haegel L and Jaume R 2021 *Phys. Rev. D* **103** 124060
[125] Cook G B 2020 *Phys. Rev. D* **102** 024027
[126] Boersma O M, Nichols D A and Schmidt P 2020 *Phys. Rev. D* **101** 083026
[127] Shao L 2020 *Phys. Rev. D* **101** 104019
[128] Varma V, Isi M and Biscoveanu S 2020 *Phys. Rev. Lett.* **124** 101104
[129] Shaik F H, Lange J, Field S E, O'Shaughnessy R, Varma V, Kidder L E, Pfeiffer H P and Wysocki D 2020 *Phys. Rev. D* **101** 124054

[130] Ota I and Chirenti C 2020 *Phys. Rev. D* **101** 104005
[131] Rifat N E M, Field S E, Khanna G and Varma V 2020 *Phys. Rev. D* **101** 081502
[132] Kalaghatgi C, Hannam M and Raymond V 2020 *Phys. Rev. D* **101** 103004
[133] Siemonsen N and Vines J 2020 *Phys. Rev. D* **101** 064066
[134] Antonelli A, van de Meent M, Buonanno A, Steinhoff J and Vines J 2020 *Phys. Rev. D* **101** 024024
[135] Kiuchi K, Kawaguchi K, Kyutoku K, Sekiguchi Y and Shibata M 2020 *Phys. Rev. D* **101** 084006
[136] Pompili L *et al* 2023 *Phys. Rev. D* **108** 124035
[137] Baibhav V, Cheung M H Y, Berti E, Cardoso V, Carullo G, Cotesta R, Del Pozzo W and Duque F 2023 *Phys. Rev. D* **108** 104020
[138] Cheung M H Y *et al* 2023 *Phys. Rev. Lett.* **130** 081401
[139] Mitman K *et al* 2023 *Phys. Rev. Lett.* **130** 081402
[140] Cotesta R, Carullo G, Berti E and Cardoso V 2022 *Phys. Rev. Lett.* **129** 111102
[141] Ramos-Buades A, Buonanno A, Khalil M and Ossokine S 2022 *Phys. Rev. D* **105** 044035
[142] Cristofoli A, Gonzo R, Kosower D A and O'Connell D 2022 *Phys. Rev. D* **106** 056007
[143] Warburton N, Pound A, Wardell B, Miller J and Durkan L 2021 *Phys. Rev. Lett.* **127** 151102
[144] Estellés H, Colleoni M, García-Quirós C, Husa S, Keitel D, Mateu-Lucena M, Planas M d L and Ramos-Buades A 2022 *Phys. Rev. D* **105** 084040
[145] Islam T, Varma V, Lodman J, Field S E, Khanna G, Scheel M A, Pfeiffer H P, Gerosa D and Kidder L E 2021 *Phys. Rev. D* **103** 064022
[146] Nagar A, Bonino A and Rettegno P 2021 *Phys. Rev. D* **103** 104021
[147] Calderón Bustillo J, Sanchis-Gual N, Torres-Forné A and Font J A 2021 *Phys. Rev. Lett.* **126** 201101
[148] Jiménez Forteza X, Bhagwat S, Pani P and Ferrari V 2020 *Phys. Rev. D* **102** 044053
[149] Cotesta R, Marsat S and Pürrer M 2020 *Phys. Rev. D* **101** 124040
[150] Chiaramello D and Nagar A 2020 *Phys. Rev. D* **101** 101501
[151] Nagar A, Riemenschneider G, Pratten G, Rettegno P and Messina F 2020 *Phys. Rev. D* **102** 024077
[152] Okounkova M 2020 *Phys. Rev. D* **102** 084046
[153] Compère G, Oliveri R and Seraj A 2020 *J. High Energy Phys.* JHEP10(2020)116
Compère G, Oliveri R and Seraj A 2024 *J. High Energy Phys.* JHEP06(2024)045 (erratum)
[154] Bhagwat S, Forteza X J, Pani P and Ferrari V 2020 *Phys. Rev. D* **101** 044033
[155] Abbott R *et al* (LIGO Scientific and Virgo) 2020 *Astrophys. J. Lett.* **900** L13
[156] Ossokine S *et al* 2020 *Phys. Rev. D* **102** 044055
[157] Abbott R *et al* (LIGO Scientific and Virgo) 2020 *Phys. Rev. Lett.* **125** 101102
[158] SXS Collaboration 2025 The SXS catalog of simulations v3.0.0 (available at: https://zenodo.org/doi/10.5281/zenodo.15415231)
[159] Woodford C J, Boyle M and Pfeiffer H P 2019 *Phys. Rev. D* **100** 124010
[160] Lehner L 2001 *Class. Quantum Grav.* **18** R25–R86
[161] Gourgoulhon E 2007 arXiv:gr-qc/0703035
[162] Baumgarte T W and Shapiro S L 2010 *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge University Press)
[163] Duez M D and Zlochower Y 2019 *Rep. Prog. Phys.* **82** 016902
[164] Sarbach O and Tiglio M 2001 *Phys. Rev. D* **64** 084016
[165] Regge T and Wheeler J A 1957 *Phys. Rev.* **108** 1063–9
[166] Zerilli F J 1970 *Phys. Rev. Lett.* **24** 737–8
[167] Rinne O, Buchman L T, Scheel M A and Pfeiffer H P 2009 *Class. Quantum Grav.* **26** 075009
[168] Boyle M, Iozzo D, Stein L, Khairnar A, Rüter H, Scheel M, Varma V and Mitman K 2025 scri (https://doi.org/10.5281/zenodo.15237110)
[169] Boyle M 2016 *Phys. Rev. D* **93** 084031
[170] Pretto I G, Scheel M A and Teukolsky S A 2024 arXiv:2407.20470
[171] Mroue A H *et al* 2013 *Phys. Rev. Lett.* **111** 241104
[172] Pereira T and Sturani R 2024 *Gen. Relativ. Gravit.* **56** 24
[173] Khera N, Ashtekar A and Krishnan B 2021 *Phys. Rev. D* **104** 124071
[174] Wang Z, Zhao J and Cao Z 2024 *Commun. Theor. Phys.* **76** 015403
[175] Nobili F, Bhagwat S, Pacilio C and Gerosa D 2025 *Phys Rev. D* **112** 044058
[176] Szilágyi B, Blackman J, Buonanno A, Taracchini A, Pfeiffer H P, Scheel M A, Chu T, Kidder L E and Pan Y 2015 *Phys. Rev. Lett.* **115** 031102
[177] Mitman K, Stein L C, Boyle M, Deppe N, Kidder L E, Pfeiffer H P and Scheel M A 2025 *Class.Quantum Grav.* **42** 117001

[178] Boschini M, Loutrel N, Gerosa D and Fumagalli G 2025 *Phys. Rev. D* **111** 024008
[179] Islam T and Venumadhav T 2025 arXiv:2502.02739
[180] Ramos-Buades A, van de Meent M, Pfeiffer H P, Rüter H R, Scheel M A, Boyle M and Kidder L E 2022 *Phys. Rev. D* **106** 124040
[181] Szilágyi B 2014 *Int. J. Mod. Phys.* **D23** 1430014
[182] Mitman K *et al* 2024 *Class. Quantum Grav.* **41** 223001
[183] Boyle M 2013 *Phys. Rev. D* **87** 104006
[184] SXS Collaboration 2025 Waveform differences and mismatches for v3.0.0 of the SXS collaboration's catalog of binary black hole simulations (https://doi.org/10.5281/zenodo.15465745)
[185] Flanagan E E and Hughes S A 1998 *Phys. Rev. D* **57** 4566–87
[186] Lindblom L, Owen B J and Brown D A 2008 *Phys. Rev. D* **78** 124020
[187] McWilliams S T, Kelly B J and Baker J G 2010 *Phys. Rev. D* **82** 024014
[188] Baird E, Fairhurst S, Hannam M and Murphy P 2013 *Phys. Rev. D* **87** 024035
[189] Chatziioannou K, Klein A, Yunes N and Cornish N 2017 *Phys. Rev. D* **95** 104004
[190] Toubiana A and Gair J R 2024 arXiv:2401.06845
[191] LIGO power spectral density (available at: https://dcc.ligo.org/public/0149/T1800044/005/T1800044-v5.pdf)
[192] LIGO power spectral density (available at: https://dcc.ligo.org/public/0149/T1800044/005/aLIGODesign.txt)
[193] Evans M *et al* 2021 arXiv:2109.09882
[194] Srivastava V, Davis D, Kuns K, Landry P, Ballmer S, Evans M, Hall E D, Read J and Sathyaprakash B S 2022 *Astrophys. J.* **931** 22
[195] CE power spectral density (available at: https://dcc.cosmicexplorer.org/CE-T2000017/public)
[196] ET power spectral density (available at: https://apps.et-gw.eu/tds/?r = 14065)
[197] Babak S, Petiteau A and Hewitson M 2021 arXiv:2108.01167
[198] LISA power spectral density (available at: https://gitlab.in2p3.fr/LISA/lisa_sensitivity_snr/-/blob/master/Data/LISAToolBox/IMS15pm_3yr_Sens_XAE_SemiNum.txt?ref_type = heads)
[199] Colpi M *et al* (LISA) 2024 arXiv:2402.07571
[200] (Available at: www.black-holes.org/for-researchers/spec)
[201] York J W 1999 *Phys. Rev. Lett.* **82** 1350–3
[202] Pfeiffer H P and York J W 2003 *Phys. Rev. D* **67** 044022
[203] Lovelace G, Owen R, Pfeiffer H P and Chu T 2008 *Phys. Rev. D* **78** 084017
[204] Cook G B and Scheel M A 1997 *Phys. Rev. D* **56** 4775–81
[205] Ma S, Giesler M, Scheel M A and Varma V 2021 *Phys. Rev. D* **103** 084029
[206] Varma V and Scheel M A 2018 *Phys. Rev. D* **98** 084032
[207] Chen Y, Deppe N, Kidder L E and Teukolsky S A 2021 *Phys. Rev. D* **104** 084046
[208] Varma V, Scheel M A and Pfeiffer H P 2018 *Phys. Rev. D* **98** 104011
[209] Caudill M, Cook G B, Grigsby J D and Pfeiffer H P 2006 *Phys. Rev. D* **74** 064011
[210] Pfeiffer H P, Kidder L E, Scheel M A and Teukolsky S A 2003 *Comput. Phys. Commun.* **152** 253–73
[211] Buchman L T, Pfeiffer H P, Scheel M A and Szilagyi B 2012 *Phys. Rev. D* **86** 084033
[212] Ossokine S, Foucart F, Pfeiffer H P, Boyle M and Szilágyi B 2015 *Class. Quantum Grav.* **32** 245010
[213] Pfeiffer H P, Brown D A, Kidder L E, Lindblom L, Lovelace G and Scheel M A 2007 *Class. Quantum Grav.* **24** S59–82
[214] Buonanno A, Kidder L E, Mroue A H, Pfeiffer H P and Taracchini A 2011 *Phys. Rev. D* **83** 104034
[215] Mroue A H and Pfeiffer H P 2012 arXiv:1210.2958
[216] Habib S, Scheel M A and Teukolsky S A 2025 *Phys. Rev. D* **111** 084059
[217] Knapp T, Chatziioannou K, Pfeiffer H, Scheel M A and Kidder L E 2025 *Phys. Rev. D* **111** 024003
[218] Nee P J *et al* 2025 *Class. Quantum Grav.* **42** 135011
[219] Friedrich H 1985 *Commun. Math. Phys.* **100** 525–43
[220] Garfinkle D 2002 *Phys. Rev. D* **65** 044029
[221] Pretorius F 2005 *Class. Quantum Grav.* **22** 425–52
[222] Lindblom L, Scheel M A, Kidder L E, Owen R and Rinne O 2006 *Class. Quantum Grav.* **23** S447–62
[223] Gundlach C, Martin-Garcia J M, Calabrese G and Hinder I 2005 *Class. Quantum Grav.* **22** 3767–74
[224] Lindblom L and Szilagyi B 2009 *Phys. Rev. D* **80** 084019

[225] Choptuik M W and Pretorius F 2010 *Phys. Rev. Lett.* **104** 111101
[226] Szilagyi B, Lindblom L and Scheel M A 2009 *Phys. Rev. D* **80** 124010
[227] Kidder L E and Finn L S 2000 *Phys. Rev. D* **62** 084026
[228] Scheel M A, Boyle M, Chu T, Kidder L E, Matthews K D and Pfeiffer H P 2009 *Phys. Rev. D* **79** 024003
[229] Hemberger D A, Scheel M A, Kidder L E, Szilágyi B, Lovelace G, Taylor N W and Teukolsky S A 2013 *Class. Quantum Grav.* **30** 115001
[230] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes: The Art of Scientific Computing* 3rd edn (Cambridge University Press)
[231] Ossokine S, Kidder L E and Pfeiffer H P 2013 *Phys. Rev. D* **88** 084031
[232] Rinne O 2006 *Class. Quantum Grav.* **23** 6275–300
[233] Rinne O, Lindblom L and Scheel M A 2007 *Class. Quantum Grav.* **24** 4053–78
[234] Hesthaven J 2000 *Appl. Numer. Math.* **33** 23–41
[235] Bjørhus M 1995 *SIAM J. Sci. Comput.* **16** 542–57
[236] Lovelace G, Scheel M and Szilagyi B 2011 *Phys. Rev. D* **83** 024010
[237] Deppe N *et al* 2025 SpECTRE v2025.04.21 (available at: https://spectre-code.org)
[238] Owen R 2009 *Phys. Rev. D* **80** 084012
[239] Owen R, Fox A S, Freiberg J A and Jacques T P 2019 *Phys. Rev. D* **99** 084031
[240] Scheel M A, Pfeiffer H P, Lindblom L, Kidder L E, Rinne O and Teukolsky S A 2006 *Phys. Rev. D* **74** 104006
[241] Scheel M A, Giesler M, Hemberger D A, Lovelace G, Kuper K, Boyle M, Szilágyi B and Kidder L E 2015 *Class. Quantum Grav.* **32** 105009
[242] Iozzo D A B, Boyle M, Deppe N, Moxon J, Scheel M A, Kidder L E, Pfeiffer H P and Teukolsky S A 2021 *Phys. Rev. D* **103** 024039
[243] Long O, Pfeiffer H P, Buonanno A, Jakobsen G U, Mogull G, Ramos-Buades A, Rüter H R, Kidder L E and Scheel M A 2025 arXiv:2507.08071
[244] Huang N E, Shen Z, Long S R, Wu M C, Shih H H, Zheng Q, Yen N C, Tung C C and Liu H H 1998 *Proc. R. Soc. A* **454** 903–95
[245] Favata M 2009 *Phys. Rev. D* **80** 024002
[246] Bishop N T, Gómez R, Lehner L and Winicour J 1996 *Phys. Rev. D* **54** 6153–65
[247] Bishop N T and Rezzolla L 2016 *Living Rev. Relativ.* **19** 2
[248] Handmer C J, Szilágyi B and Winicour J 2016 *Class. Quantum Grav.* **33** 225007
[249] Barkett K, Moxon J, Scheel M A and Szilágyi B 2020 *Phys. Rev. D* **102** 024004
[250] Geroch R P, Held A and Penrose R 1973 *J. Math. Phys.* **14** 874–81
[251] European Organization For Nuclear Research and OpenAIRE 2013 Zenodo (available at: www.zenodo.org/)
[252] Caltech Library 2024 CaltechDATA (available at: https://data.caltech.edu/)
[253] Agrawal A and Scolnick M 2023 marimo - an open-source reactive notebook for Python (available at: https://github.com/marimo-team/marimo)
[254] Chatham H *et al* 2025 Pyodide (https://doi.org/10.5281/zenodo.15046691)
[255] Song X C *et al* 2022 Anvil - system architecture and experiences from deployment and early user operations *Practice and Experience in Advanced Research Computing 2022: Revolutionary: Computing, Connections, You PEARC '22 (New York, NY, USA)* (Association for Computing Machinery)
[256] Strande S *et al* 2021 Expanse: Computing without boundaries: Architecture, deployment and early operations experiences of a supercomputer designed for the rapid evolution in science and engineering *Practice and Experience in Advanced Research Computing 2021: Evolution Across All Dimensions PEARC '21 (New York, NY, USA)* (Association for Computing Machinery)
[257] Boerner T J, Deems S, Furlani T R, Knuth S L and Towns J 2023 Access: advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good PEARC '23 (New York, NY, USA)* (Association for Computing Machinery) pp 173–6
[258] Towns J *et al* 2014 *Comput. Sci. Eng.* **16** 62–74
[259] Bode B, Butler M, Dunning T, Hoefler T, Kramer W, Gropp W and Hwu W m 2013 The Blue Waters super-system for super-science *Contemporary High Performance Computing* (Chapman & Hall/CRC Computational Science) pp 339–66

[260] Kramer W, Butler M, Bauer G, Chadalavada K and Mendes C 2015 Blue Waters Parallel I/O Storage Sub-system *High Performance Parallel I/O*, ed Prabhat and Q Koziol (CRC Publications, Taylor and Francis Group) pp 17–32
[261] Cook G B and Pfeiffer H P 2004 *Phys. Rev. D* **70** 104016
[262] Mroue A H, Pfeiffer H P, Kidder L E and Teukolsky S A 2010 *Phys. Rev. D* **82** 124016
[263] Seward J, Quintero F M and Snyder M 2023 BZIP2 [Computer Software] (available at: https://gitlab.com/bzip2/bzip2/)
[264] The HDF Group and Koziol Q 2020 HDF5-Version 1.12.0 [Computer Software] (https://doi.org/10.11578/dc.20180330.1)
[265] Aasi J *et al* (LIGO Scientific, Virgo and NINJA-2) 2014 *Class. Quantum Grav.* **31** 115004
[266] Boyle M 2010 GridRefinement (https://doi.org/10.5281/zenodo.13993495)
[267] Ajith P *et al* 2011 Data formats for numerical relativity (arXiv:0709.0093v3)
[268] Schmidt P, Harry I W and Pfeiffer H P 2017 arXiv:1703.01076
[269] Galley C R and Schmidt P 2016 Fast and efficient evaluation of gravitational waveforms via reduced-order spline interpolation (arXiv:1611.07529)
[270] Engelson V, Fritzson D and Fritzson P 2000 Lossless compression of high-volume numerical data from simulations *Proc. DCC 2000. Data Compression Conf.*
[271] Klimenko S, Mours B, Shawhan P and Sazonov A 2001 Data compression study with the E2 data *Technical Report* LIGO-T010033-x0 (available at: https://dcc.ligo.org/LIGO-T010033/public)
[272] Lindstrom P and Isenburg M 2006 *IEEE Trans. Vis. Comput. Graphics* **12** 1245–50
[273] Ratanaworabhan P, Ke J and Burtscher M 2006 Fast lossless compression of scientific floating-point data *Data Compression Conf. (DCC'06)* pp 133–42
[274] Hübbe N and Kunkel J 2013 *Comput. Sci. Res. Dev.* **28** 231–9
[275] Masui K *et al* 2015 *Astron. Comput.* **12** 181–90
[276] Claggett S, Azimi S and Burtscher M 2018 SPDP: an automatically synthesized lossless compression algorithm for floating-point data *2018 Data Compression Conf.* pp 335–44
[277] Boyle M 2017 *Adv. Appl. Clifford Algebras* **27** 2345–74
[278] Boyle M 2024 The quaternionic package (https://doi.org/10.5281/zenodo.13350880)
[279] Boyle M 2023 The spherical package (https://doi.org/10.5281/zenodo.10214833)
[280] Boyle M, Kidder L E, Ossokine S and Pfeiffer H P 2014 arXiv:1409.4431
[281] The HDF Group 2024 HDF5 API Reference (available at: https://support.hdfgroup.org/documentation/hdf5/latest/group___d_c_p_l.html) (Accessed 15 October 2024)
[282] IEEE 2019 IEEE Std 754-2019 (Revision of IEEE 754-2008) pp 1–84
[283] Art *et al* 2022 wildart/Evolutionary.jl: v0.11.1 (https://doi.org/10.5281/zenodo.5851574)
[284] Huffman D A 1952 *Proc. IRE* **40** 1098–101
[285] Burrows M and Wheeler D 1994 A block-sorting loss less data compression algorithm (available at: https://api.semanticscholar.org/CorpusID:2167441)
[286] Ryabko B Y 1980 *Probl. Inf. Transm.* **16** 265–9
[287] Bentley J L, Sleator D D, Tarjan R E and Wei V K 1986 *Commun. ACM* **29** 320–30
[288] Collin L 2009 XZ Utils: Free general-purpose data compression software with high compression ratio (available at: https://tukaani.org/xz/) (Accessed 1 October 2023)
[289] Alakuijala J and Szabadka Z 2016 Brotli Compressed Data Format RFC 7932 (available at: www.rfc-editor.org/info/rfc7932)
[290] Deutsch L P 1996 GZIP file format specification version 4.3 RFC 1952 (available at: www.rfc-editor.org/info/rfc1952)