

Determining remnant parameters from black-hole binary systems

Final Report - LIGO Document T1700339

Nicholas Meyer¹ and Mark Scheel¹

¹*Theoretical Astrophysics 350-17, California Institute of Technology, Pasadena, CA 91125, USA*
(Dated: September 22, 2017)

Computing the remnant mass, spin, and recoil of a black hole binary in principle requires numerical relativity (NR) simulations. Unfortunately, NR simulations cannot be performed quickly enough for some waveform models and LIGO data analysis routines that require remnant parameters. We develop phenomenological formulae for the remnant mass, spin, and recoil of binary systems given arbitrary initial spins and mass ratios. We do this by constructing fits to NR simulations in the SXS catalog. In particular, we explore the use of gaussian process regression. We use the SXS catalog to compare the accuracy of our remnant mass and spin fits with that of the remnant mass and spin formulae in the Effective One Body (EOB) waveform model in the LSC Algorithm Library (LALSuite).

MOTIVATION FOR PHENOMENOLOGICAL FITS AND METHODS OF FOCUS

The detection of gravitational waves by the advanced LIGO interferometers [1] represents a confirmation of a substantial prediction of Einstein’s theory of general relativity. This discovery has additionally established gravitational waves as a new source of information about the observable universe.

Of particular relevance is the use of gravitational-wave data to estimate the parameters (masses and spins) of mergers of black-hole binary systems. For this purpose, it is often beneficial to predict the post-merger mass, spin, and recoil velocity as a function of the initial mass ratio and spins of the binary. For example, the SEOBNRv3 waveform model [2] used for LIGO data analysis uses such a prediction of the post-merger mass and spin to compute the ringdown portion of the waveform. Predicting the parameters of the remnant black hole requires numerical relativity simulations, and exhausting the input parameter space through direct simulation is not tractable. Approximate formulas for these final parameters are useful because they provide a procedure for obtaining final parameter values at a continuum of initial parameter inputs at a lower computational cost.

To address this need, phenomenological fits of final parameters as functions of initial parameters have been developed by several efforts for the case of aligned spins [3, 4] and some cases of generic and precessing spins [5, 6]. Our goal is to expand upon this work using a new set of over 1000 simulations of binary mergers from the SXS public and “incoming” catalogs [7], with an emphasis on the use of general machine learning methods to obtain fits. In particular, we evaluate the suitability of Gaussian Process Regression for remnant parameter fits.

In the appendix, some code is provided for the cross validation routine used in this analysis. This is provided in order to assist in replication of the results and use of these fits in further work.

REGRESSION ON NUMERICAL RELATIVITY SIMULATIONS

Our ultimate goal is to fit the remnant mass, spin, and recoil as a function of all seven input parameters in the case of generic spins. As a first step, we fit the remnant mass and spin for the case where the initial black holes have zero spin and we found good agreement between the data and the model (see Figures 1, 2, 3). Next, we treat the case where the black holes have spins aligned or anti-aligned with the orbital angular momentum and compare this with [8]; residual plots are given in Figures 4 and 5. The remnant mass for the full dimensional case (generic spins) is then fit with results summarized in Figure 6 and Table I. Fits for the remnant spin magnitude in this case are also summarized in Figure 7 and Table II. Work on recoil fits is ongoing.

Gaussian Process Regression

A Gaussian Process can be viewed as a means of applying a nonlinear transformation to training data, and then linearly combining the results to predict the value of a function at a new input point [9]. This nonlinear transformation is given by a “kernel” function defined between two input points, which is often a squared-exponential or “radial basis function” (RBF) [9]. Parameters of this function are referred to as “hyperparameters”. In contrast to a parametric fit, which optimizes parameters of a given function according to training data, Gaussian Process Regression optimizes kernel hyperparameters, so that the trained model uses the training data itself to predict function values at new points [9].

The Gaussian Process Regression class provided by `sklearn` is used to perform the fits. This package provides a procedure to train a Gaussian Process model in which kernel hyperparameters are determined by maximizing log-marginal-likelihood [10]. Unless otherwise

noted, the default RBF kernel is used.

Extremely Randomized Trees Regression

Tree regression methods construct a decision tree from a training data set. When predicting function values at new data points, the values of input features are used to traverse the tree until a leaf node is reached, at which point a final regression model is applied.

“Extremely Randomized Tree” regression uses several decision trees, each trained on a random subset of the data using random [11]. When predicting the function value at a new point, an average of the predictions of the trees is taken.

In `sklearn`, this is implemented in `ExtraTreesRegressor` in the `ensemble` subpackage [11].

Training / Validation Partitioning and Cross Validation Routine

In order to accurately estimate fitting error, a validation set is separated from remaining data and excluded from the fit; a model’s ability to predict correct values for the validation set is used as a proxy for its error in general. The histograms in Figures 2, 3, 4, 5, 6, 7 report residuals for both the training and validation subsets of data.

In addition, a 10-fold cross validation routine was developed using the `KFold` class in `sklearn.model_selection`. In this procedure, the data is shuffled and then partitioned into ten equal blocks. For each of these blocks, a fit is performed on the other nine blocks and then evaluated on the block not used in fitting. Table I contains norms produced by the cross validation procedure for the Gaussian Process Regression fit to the remnant mass of the generic spin case, when setting the noise parameter “alpha” [10] to 10^{-4} .

Fitting residuals

Zero initial spin

Gaussian Process Regression was used to fit the final remnant mass and spin magnitude in the case of initial spins with a squared magnitude of less than 10^{-10} . Forty-eight points meeting this criterion were used in the fits.

From the residuals in Figure 2 it can be seen that the error in the fit is within 0.1% for all simulations. In the top plot in Figure 1 the best value fit appears smooth and appears to interpolate well for input mass ratios between 1 and 10. Plots for the remnant spin and residuals in the

Spinless plots

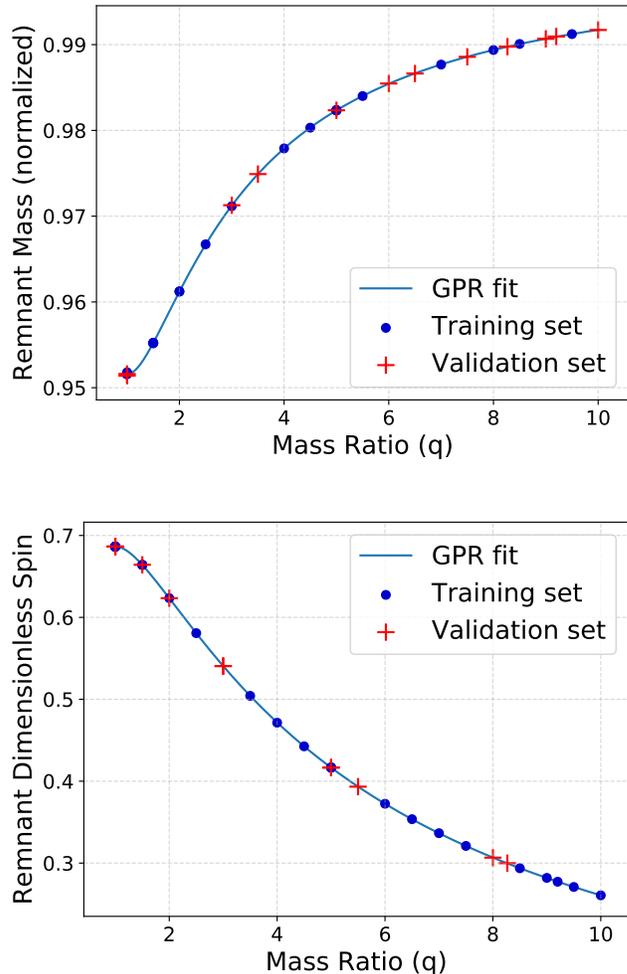


FIG. 1. Plot of predicted remnant mass (solid curve, top) and predicted remnant spin (solid curve, bottom) for spinless case from Gaussian Process Regression fit, each with training data set (solid dots) and validation set (plus signs) overlaid. The training set was used to fit the Gaussian Process model, while the validation set was used only in assessing residuals. Fitting was performed on the relaxed masses of each input black hole (two input features). Note that the mass ratio can be similar between different simulations, and so some points at certain mass ratios represent more than one training or validation simulation.

initially spinless case appear in Figures 1 (bottom) and 3.

Aligned / anti-aligned initial spin

The validation residuals obtained from the Gaussian Process Regression fit for remnant mass and spin magnitude on the SXS catalog aligned spin data improved on residuals produced by the remnant mass formula in Ref.

Spinless - Mass Residuals

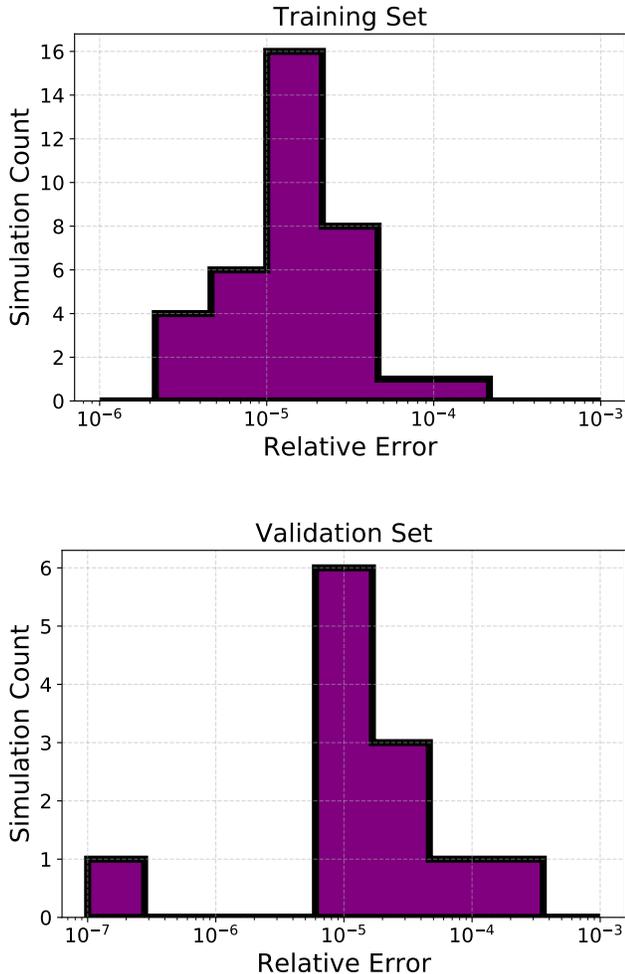


FIG. 2. Training set residuals (top) and validation set residuals (bottom) for a Gaussian Process Regression fit of the remnant mass for the spinless data. Here 36 randomly selected training points were used for the fit out of the 48 spinless simulations, and the remaining 12 formed the validation set. The histogram bins are divided into ranges of relative errors, and the height of each bar corresponds to the number of simulations for which the error lies within the bin boundaries.

[8] using the published coefficients (see Figure 4 - the Gaussian Process Regression fit errors are clustered closer to zero in both the training and validation sets). Additionally, a least-squares fit of the remnant mass formula in Ref. [8] was performed on the SXS catalog aligned spin data, and the Gaussian Process fit performed better (Figure 4).

Spinless - Spin Residuals

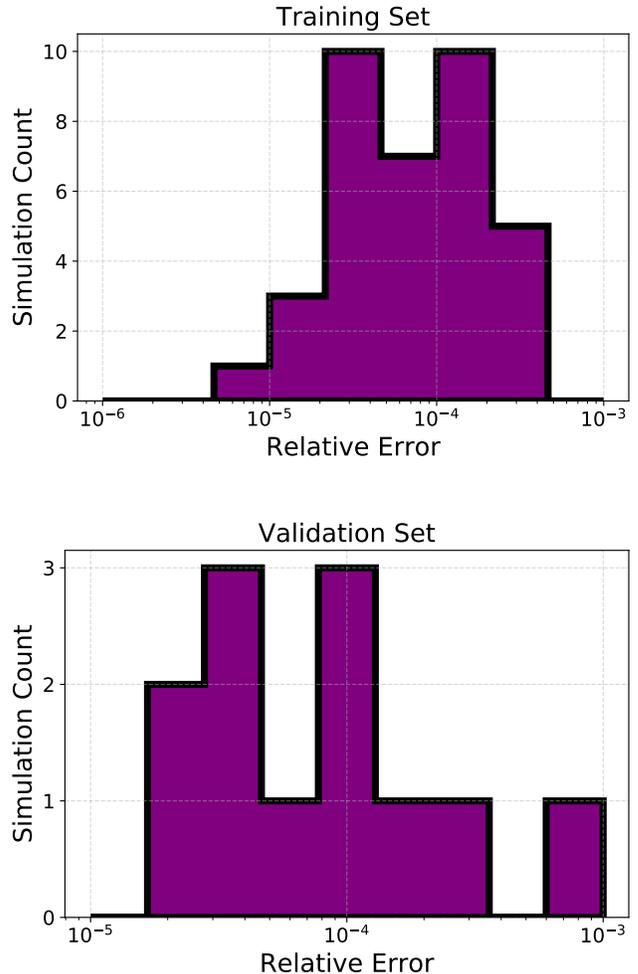


FIG. 3. Training set residuals (top) and validation set residuals (bottom) for a Gaussian Process Regression fit of the remnant spin magnitude for the spinless data.

Arbitrary initial spin

A Gaussian Process Regression fit was performed for the remnant mass and spin magnitude in the generic spin case.

In Figure 6 and Table I, it can be seen that typical errors for remnant mass are well under 1% and the maximum absolute error is estimated to be of order 1%. Table I summarizes normed quantities from a cross validation analysis of the generic input spin fit. For remnant mass, a fit using Extremely Randomized Tree regression was performed as well, and residual norms are present in I for comparison.

The remnant spin magnitude Gaussian Process fit was performed by fitting each spin component (x , y , z) individually and then taking the magnitude of the result.

Aligned / Anti-aligned - Mass

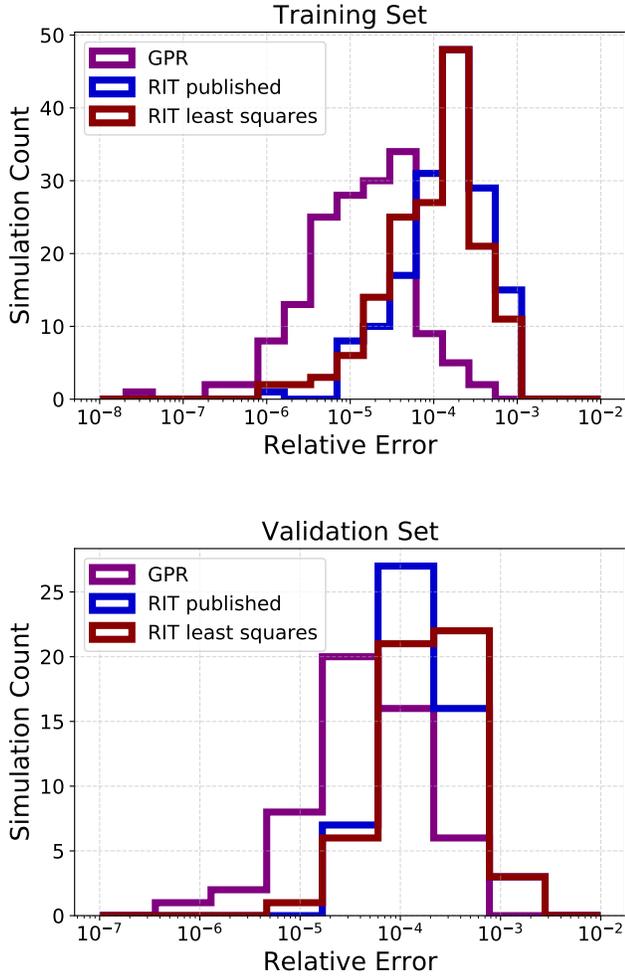


FIG. 4. Training set residuals (top) and validation set residuals (bottom) for Gaussian Process Regression and Ref. [8] fits of the remnant mass in the aligned / anti-aligned input spin case (3 dimensional input parameter space). A total of 212 aligned spin data points were chosen from the public and incoming SXS catalogs by selecting all BBH simulations for which both initial spins retained at least 99.9 percent of their magnitude when the inner product was taken with the initial angular momentum direction; these points were partitioned into a training set and a smaller validation set. “RIT published” refers to residuals obtained using the remnant mass model in Ref. [8] with the published coefficients. “RIT least squares” refers to the same model using coefficients determined by a least squares fit against the training set. Fitting was performed on the relaxed input parameters (parameters measured after the dissipation of junk radiation). The histogram bins are divided into ranges of relative errors, and the height of each bar corresponds to the number of simulations for which the error lies within the bin boundaries.

Aligned / Anti-aligned - Spin

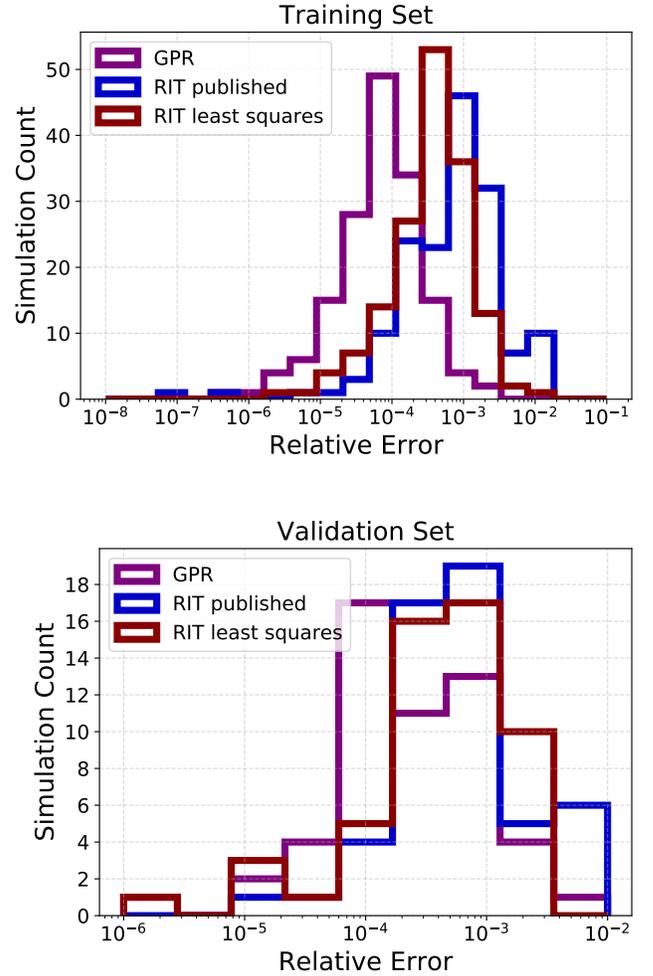


FIG. 5. Training set residuals (top) and validation set residuals (bottom) for a Gaussian Process fit of the remnant spin in the aligned / anti-aligned input spin case.

Full Dimensional - Mass

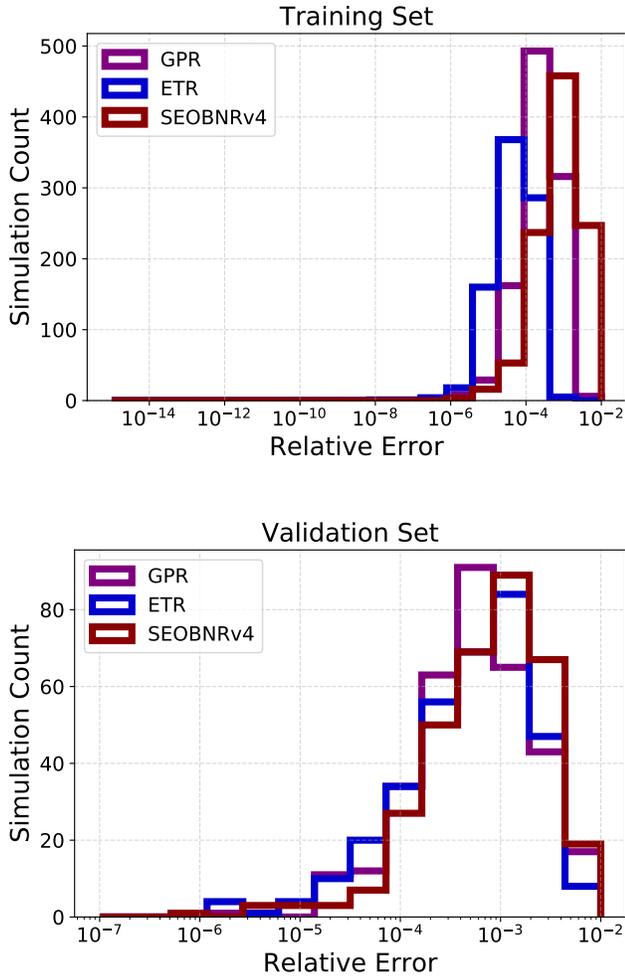


FIG. 6. Training set residuals (top) and validation set residuals (bottom) for a Gaussian Process and Extremely Randomized Trees fit of the remnant spin in the arbitrary spin case (7 dimensional input space). The training and validation sets were partitioned from 1352 black hole binary simulations with arbitrary initial spins. The residuals of the remnant mass fit used in the SEOBNRv4 waveform model as implemented in LALSimulation are also overlayed on each plot for comparison.

Full Dimensional - Spin

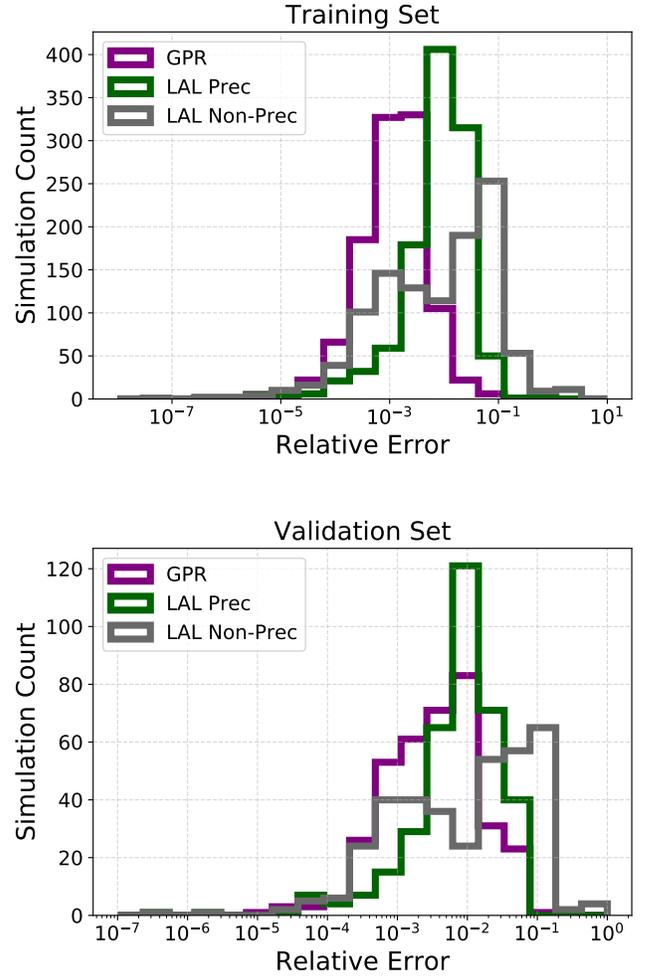


FIG. 7. Training set residuals (top) and validation set residuals (bottom) for a Gaussian Process fit of the remnant spin magnitude in the case of arbitrary initial spins. The 'LAL' labeled line refers to the SEOBNRv4 model fit, while the 'LALPrec' labeled line refers to the SEOBNRv3 model (which accounts for non-aligned spins, unlike SEOBNRv4).

TABLE I. Relative error norms for the cross validation of the Gaussian Process Regression fit of remnant mass for the arbitrary spin case. Units are in percent or 10^{-2} . The norms are computed for the residual set X as $L_1 = \frac{1}{|X|} \sum_{x \in X} |x|$, $L_2 = \frac{1}{|X|} \sqrt{\sum_{x \in X} x^2}$, $L_{\text{inf}} = \max_{x \in X} |x|$. Columns 1-10 represent the error evaluated on each subset when withheld from fitting during cross validation; the mean of each norm across these 10 values is given in the final column.

| Validation Subset GPR | | | | | | | | | | | Mean |
|---|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| L_1 | 0.12 | 0.12 | 0.12 | 0.14 | 0.13 | 0.1 | 0.11 | 0.14 | 0.13 | 0.12 | 0.12 |
| L_2 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| L_{inf} | 1.0 | 0.73 | 1.06 | 0.96 | 1.34 | 0.84 | 1.3 | 1.28 | 0.96 | 1.2 | 1.07 |
| Validation Subset Extra Trees Regressor | | | | | | | | | | | Mean |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| L_1 | 0.09 | 0.08 | 0.08 | 0.08 | 0.1 | 0.09 | 0.08 | 0.1 | 0.08 | 0.09 | 0.09 |
| L_2 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| L_{inf} | 1.03 | 0.48 | 0.53 | 0.38 | 0.97 | 0.75 | 0.86 | 0.57 | 0.63 | 0.53 | 0.67 |

TABLE II. Relative error norms for the cross validation of the Gaussian Process Regression fit of remnant spin for the arbitrary spin case. Units are in percent or 10^{-2} . The norms are computed for the residual set X as $L_1 = \frac{1}{|X|} \sum_{x \in X} |x|$, $L_2 = \frac{1}{|X|} \sqrt{\sum_{x \in X} x^2}$, $L_{\text{inf}} = \max_{x \in X} |x|$. Columns 1-10 represent the error evaluated on each subset when withheld from fitting during cross validation; the mean of each norm across these 10 values is given in the final column.

| Validation Subset GPR | | | | | | | | | | | Mean |
|---|------|-------|------|------|-------|------|------|------|-------|-------|------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| L_1 | 0.61 | 0.7 | 0.59 | 0.55 | 0.55 | 0.62 | 0.56 | 0.51 | 0.51 | 0.64 | 0.58 |
| L_2 | 0.08 | 0.17 | 0.08 | 0.07 | 0.08 | 0.08 | 0.08 | 0.06 | 0.07 | 0.13 | 0.09 |
| L_{inf} | 3.95 | 20.91 | 4.1 | 4.16 | 4.15 | 4.09 | 3.8 | 2.68 | 3.71 | 13.58 | 6.51 |
| Validation Subset Extra Trees Regressor | | | | | | | | | | | Mean |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| L_1 | 1.01 | 1.13 | 0.88 | 1.1 | 1.0 | 1.12 | 0.99 | 0.86 | 1.2 | 1.19 | 1.05 |
| L_2 | 0.13 | 0.15 | 0.11 | 0.14 | 0.16 | 0.15 | 0.13 | 0.11 | 0.29 | 0.16 | 0.15 |
| L_{inf} | 4.83 | 7.29 | 3.73 | 7.12 | 12.46 | 7.35 | 4.82 | 5.41 | 33.93 | 8.65 | 9.56 |

DISCUSSION AND FUTURE WORK

Evaluation of Gaussian Process Regression

Gaussian Process Regression proved robust in predicting remnant mass and spin, and matched or improved upon other implementations available. Residuals are small in the spinless case, and the interpolation between data points is smooth (Figures 1, 2, 3). Residual plots in the aligned initial spin case show lower error obtained for a Gaussian Process Regression fit on the SXS data than both the “RIT” fit (Ref [8]) as published and when trained on SXS data by least squares regression (Figures 4, 5). In the arbitrary initial spin case, Gaussian Process Regression matches the best EOB formula for remnant mass and improves upon the best EOB formula for remnant spin (Figures 6, 7).

Extremely Randomized Tree Regression for Remnant Mass

In the full dimensional input case, the `ExtraTreesRegressor` implementation provided by Scikit-learn performed better than the Gaussian Process Regression package provided by the same package; this can be seen in Table I where the mean L_1 improves from 0.12 to 0.09 percent, and mean L_{inf} improves from 1.07 to 0.67 percent. This advantage did not hold for spin magnitude predictions though, where Extremely Randomized Tree regression increased all error norms (Table II).

Comparison to EOB fits in LALSsimulation

The implementation of the SEOBNRv4 model in `src/LALSImBlackHoleRingdown.c` was used to predict remnant parameters for the SXS catalog data. The source code was minimally modified and incorporated into fitting python scripts using `ctypes`.

Table III summarizes the normed mass residuals for the models currently implemented in LALSsimulation when applied to the SXS catalog data. In Table I, the mean maximum absolute relative error is 1.08% for the Gaussian Process Regression mass fit, which is slightly higher than the L_{inf} of 0.93% for SEOBNRv2 and SEOBNRv4. Conversely, the average absolute relative error is improved in the Gaussian Process Regression fit from 0.14% to 0.12%. A fit to the remnant mass using Extremely Randomized Tree Regression (ETR) yielded more definitive improvements over the EOB model, as is summarized in the previous section.

A graphical comparison between GPR, ETR, and the SEOBNRv4 model for remnant mass is shown in Figure 6.

TABLE III. Relative error norms of remnant mass predictions on the SXS catalog data from models currently used in LALSsimulation. Units are in percent or 10^{-2} . The L_1 , L_2 , and L_{inf} norms are defined as in Table I

| | L_1 | L_2 | L_{inf} |
|-----------|-------|-------|------------------|
| EOBNR | 0.85 | 0.03 | 7.76 |
| EOBNRv2 | 0.78 | 0.03 | 7.44 |
| EOBNRv2HM | 0.78 | 0.03 | 7.44 |
| SEOBNRv1 | 0.45 | 0.02 | 4.48 |
| SEOBNRv2 | 0.14 | 0.01 | 0.93 |
| SEOBNRv4 | 0.14 | 0.01 | 0.93 |

In Figure 7, a comparison for the spin magnitude residuals is shown between an EOB model which assumes aligned spins (“LAL Non-Prec” / SEOBNRv4), an EOB model fit to the precessing spins case (“LAL Prec” / SEOBNRv3), and a Gaussian Process Regression fit. The precessing model improves on the non-precessing model, and the Gaussian Process Regression fit has smaller error when evaluated on the validation set than both EOB models. When the SEOBNRv3 model was evaluated for remnant mass, the residuals were similar to those given by the SEOBNRv4 model.

Implication for Gravitational Wave Searches

The Effective One Body (EOB) waveform models implemented in the LSC Algorithm Library (LAL) expand the waveform in quasi normal modes of the remnant black hole; these quasi normal modes depend on the final mass and spin of the remnant [12]. An improvement to these remnant parameter predictions could improve the accuracy of EOB waveforms, which could in turn improve searches.

Future Analysis of Recoil

Although the output of the code includes the final coordinate velocity of the remnant, this is only a coordinate velocity and not physically meaningful. Much better is to compute recoils from the gravitational waveform; in our case we use the spherical harmonic decomposition of the Weyl scalar Ψ_4 [13]. This is currently implemented in a script in SpEC.

Upon inspection, the magnitude of the radiated linear momentum calculated by this procedure has varied by up to order 10% or higher between resolution levels in the same simulation, even for cases with high recoil. For comparison, initial and remnant mass and spin typically vary by less than 0.1% between resolution levels for the same data set. The problem was initially worse due to contributions from “junk radiation” (gravitational

radiation produced by artifacts of the simulation itself) at early times. Although integrating starting at a time after the dissipation of the junk radiation has improved results, the aforementioned discrepancies still exist.

This uncertainty is likely because the recoil has not converged in several simulations, and re-running these at higher resolutions may be necessary [14].

Eccentricity

The eccentricity of the binary system orbit is a feature present in the simulation data. One line of future work will be to include an eccentricity cutoff, above which simulations would be removed from the training data for fits. It is possible that this will reduce the uncertainty in the input recoils, allowing for meaningful fits [14]. In addition, the remnant mass and spin fits performed did not exclude high eccentricity binary simulations; an eccentricity cutoff could improve these fits as well.

ACKNOWLEDGMENTS

Fits and analysis were performed on the wheeler machine. The SXS simulation catalog provided data for fitting. The NSF, the LIGO lab, and SFP at Caltech provided funding and resources for the project. Fitting procedures and analysis used the `sklearn` package. Code from LALSsimulation in LALSuite was incorporated into analysis routines for comparison. The Scikit-learn package provided various machine learning and fitting routines. Plots were made using Matplotlib [15]. Vijay Varma provided advice and assistance on the use of Scikit-learn. Mark Scheel provided guidance and mentorship throughout the project.

REFERENCES

-
- [1] B. P. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration), *Phys. Rev. Lett.* **116**, 061102 (2016), [arXiv:1602.03837 \[gr-qc\]](#).
- [2] Y. Pan, A. Buonanno, A. Taracchini, L. E. Kidder, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, and B. Szilágyi, *Phys. Rev. D* **89**, 084006 (2013), [arXiv:1307.6232 \[gr-qc\]](#).
- [3] J. Healy, C. O. Lousto, and Y. Zlochower, *Phys. Rev. D* **89**, 104052 (2014), [arXiv:1406.7295 \[gr-qc\]](#).
- [4] X. Jiménez-Forteza, D. Keitel, S. Husa, M. Hannam, S. Khan, and M. Pürrer, *Phys. Rev. D* **95**, 064024 (2017), [arXiv:1611.00332 \[gr-qc\]](#).
- [5] C. O. Lousto, M. Campanelli, and Y. Zlochower, *Class. Quantum Grav.* **27**, 114006 (2010), [arXiv:0904.3541 \[gr-qc\]](#).
- [6] Y. Zlochower and C. O. Lousto, *Phys. Rev. D* **92**, 024022 (2015).
- [7] <http://www.black-holes.org/waveforms>.
- [8] J. Healy and C. O. Lousto, *Phys. Rev. D* **95**, 024037 (2017), [arXiv:1610.09713 \[gr-qc\]](#).
- [9] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (The MIT Press, 2005).
- [10] http://scikit-learn.org/stable/modules/gaussian_process.html ().
- [11] <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html> ().
- [12] A. Taracchini, A. Buonanno, Y. Pan, T. Hinderer, M. Boyle, D. A. Hemberger, L. E. Kidder, G. Lovelace, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, B. Szilágyi, N. W. Taylor, and A. Zenginoglu, *Phys. Rev. D* **89** (R), 061502 (2014), [arXiv:1311.2544 \[gr-qc\]](#).
- [13] M. Ruiz, R. Takahashi, M. Alcubierre, and D. Núñez, *Gen. Relativ. Gravit.* **40**, 1705 (2008), [arXiv:0707.4654 \[gr-qc\]](#).
- [14] M. Scheel, Private Communication (2016).
- [15] J. D. Hunter, *Comput. Sci. Eng.* **9**, 90 (2007).

APPENDIX

Importance of Careful Parameter Choices

Although there is no information difference between a given input feature and a reversible mathematical transformation of this feature, these choices may be significant in fitting; for example, in the spinless case it was observed that directly using the mass ratio q or not normalizing masses by the total initial mass in fitting yielded errors which were higher by several orders of magnitude. The two suboptimal choices of parameters reflect two separate issues: fitting from the mass ratio q did not work well because Gaussian Process Regression fits seem to perform better when the input parameters are normalized; fitting using non-normalized masses did not work well because a relatively small number of early simulations did not normalize all masses by the initial mass [14]. Because Gaussian Process Regression uses locality to predict function values at new points, error can be higher when predicting a function at points far away from the training set points. Because remnant mass scales linearly with initial mass, failing to normalize masses by the initial mass causes artificially large distances between some data points.

Using the larger normalized mass¹ as the input pa-

rameter and using the normalized remnant mass as the output parameter yielded the results shown in Figures 1, 2, and 3.

Package Management

Package management of python modules on the `wheeler` machine used by SXS has presented some challenges. The class `sklearn.gaussian_process.GaussianProcessRegressor`, which is necessary for the Gaussian Process Regression code, is present only in `sklearn` version 0.18.1 and above; `wheeler` has version 0.17.1 installed system-wide using the default python environment.

To obtain an updated version of `sklearn`, a custom local root was created using the conda package manager using the following commands:

```
conda create -n name_of_local_root \
--clone=/anaconda/install/directory
source activate name_of_local_root
conda remove conda-env
conda update anaconda
```

¹ Here, the larger normalized mass refers to the quantity $q/(q+1)$,

where q is the ratio of the larger mass to the smaller mass.

Code Sample

```

from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.ensemble import ExtraTreesRegressor

def cross_validate(X, Y, folds = 10, regressor = GaussianProcessRegressor,
                  **kwargs):
    """
    Returns a dictionary of norms on the residuals from a fit
    using the specified regressor (provide class for regressor and
    provide class arguments in kwargs)

    X: function input
    Y: function output
    regressor: Regression model to evaluate
               Must implement fit and predict functions
    folds: Number of cross validation partitions
    kwargs: Arguments to be passed to regressor
    """
    assert(len(X) == len(Y))
    kf = KFold(n_splits = folds, shuffle = True)
    norms = {"L1": [], "L1_train": [], "L2": [], "L2_train": [],
            "Linf": [], "Linf_train": []}
    for train, test in kf.split(X):
        (X_train, X_test,
         Y_train, Y_test) = (X[train], X[test], Y[train], Y[test])
        gp = regressor(**kwargs)
        gp.fit(X_train, Y_train)
        residuals_train = (Y_train - gp.predict(X_train)) / Y_train
        residuals_test = (Y_test - gp.predict(X_test)) / Y_test
        # quick sanity check
        assert(len(residuals_test) < len(residuals_train))
        norms["L1"].append(L1(residuals_test))
        norms["L2"].append(L2(residuals_test))
        norms["Linf"].append(Linf(residuals_test))
        norms["L1_train"].append(L1(residuals_train))
        norms["L2_train"].append(L2(residuals_train))
        norms["Linf_train"].append(Linf(residuals_train))
    return norms

def L1(residuals):
    return sum(abs(residuals)) / len(residuals)

def L2(residuals):
    return (sum(residuals**2))**0.5 / len(residuals)

def Linf(residuals):
    return max(abs(residuals))

```