Draft version April 11, 2023
Typeset using LaTeX **twocolumn** style in AASTeX631

# Recalibrating Gravitational Wave Phenomenological Waveform Model

Kelvin K. H. Lam,[1] Kaze W. K. Wong,[2] and Thomas D. P. Edwards[3]

[1] *Department of Physics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong*
[2] *Center for Computational Astrophysics, Flatiron Institute, New York, NY 10010, USA*
[3] *William H. Miller III Department of Physics and Astronomy, Johns Hopkins University, Baltimore, Maryland 21218, USA*

## ABSTRACT

We present a simple and general method of recalibrating gravitational wave (GW) phenomenological waveform models jointly. By using `jax` and `ripple`, we can perform automatic differentiation to functions, which allows us to use gradient-based optimization methods to recalibrate waveform coefficients in IMRPhenomD model. This method reduces systematic bias previously introduced to the model and generally can improve waveform accuracy. With recalibrated coefficients, we found that the typical *mismatch* has a 50% decrease. Furthermore, we analyze the accuracy base on the waveform's intrinsic parameters. We found that waveform accuracy has significant dependence on black hole spin. Reduced spin approximation introduces degeneracy in spin, which prevented further improvement. We isolated regions in the parameter space that does not fit the waveform ansatz. These results allow us to understand more about how to develop newer phenomenological models.

## 1. INTRODUCTION

In the future, the Laser Interferometer Gravitational-wave Observatory (LIGO) will finish its maintenance and start observing new gravitational wave (GW) results. This new O4 run is expected to double the rate of current binary black hole (BBH) observations (**?**). Additionally, the sensitivity of interferometers will be increased to capture more details of GW. Having instruments with higher sensitivity, GW models of equal or higher accuracy than observations should be used to extract GW information. Otherwise, the extracted information would be affected more by GW models instead of interferometer sensitivity, resulting in a bottleneck in GW analyses. Although GW models are accurate enough for current analyses, the accuracy of current models will no longer suffice for future data analyses (**?**). Hence, it is necessary for us to develop and improve GW models.

Currently, three families of GW models are commonly used. They are the effective-one-body (EOB) (**???**), Numerical Relativity (NR) surrogate (**???**) and phenomenological (Phenom) models (**????**). EOB models are constructed by mapping two masses onto an effective body under an effective metric; NR surrogate models construct waveforms using combinations of NR waveforms; Phenom models are formulated using specific ansatz and inspiral approximations. While EOB and NR surrogate models give better waveform approximants, Phenom waveforms can be produced much faster, hence it is used mostly in data analysis tasks that requires many waveform generations. This advantage scales up in data analysis tasks such as matched filtering and parameter estimation, where many waveforms are required in each run. This motivates us to improve upon the current framework of Phenom models, thus can retain the advantage of fast waveform generation while improving the model's accuracy.

Automatic differentiation (AD) is a method to calculate derivatives of functions up to machine precision. In traditional numerical calculations, derivatives are usually obtained through numerical derivatives. Symbolic derivatives were available but it was less efficient. Both methods were not viable in machine learning, where back-propagation requires precise and rapid derivative calculations. In `python`, packages including `pytorch` (**?**), `tensorflow` (**?**), etc. utilizes AD to train machine learning models. AD's algorithm is intuitive in nature. Functions defined are decomposed into tree structures of primitives, such as addition or function evaluations. Since these operations are fundamental, they were saved as pairs internally. Differentiation proceeds forward following the tree structure, with the application of the chain rule in each step to evaluate its derivative. Analytic derivatives of such operations are applied in each

kelvin33550336@gmail.com

step and the desired derivative can then be obtained by composing back the original function according to the original structure. `ripple` (**?**) was a new implementation of IMRPhenomD, one of the Phenom models. It was first implemented in `lalsuite` using `C`. In order to make use of AD, it was rewritten using `jax`, another `python` package that supports AD. Using `ripple`, one can apply AD to GW models to obtain precise derivatives, thus allowing one to freely use derivative-based algorithms to perform data analyses.

In this paper, we investigate the possibility of further improving the accuracy of IMRPhenomD by jointly optimizing all the fitting coefficients given NR waveforms, and what constraints one may face when trying to further improve. We find that simply by applying gradient descent algorithm, one can obtain a better set of waveform coefficients, thus improving the accuracy of the model. Furthermore, by comparing the accuracy of optimized and original waveforms, we find that model-generated waveforms are very sensitive to their intrinsic parameters. Specifically, IMRPhenomD favors certain parts of the parameter space. This means IMRPhenomD introduces systematic bias to other GW analysis tasks. This showcases the flaws of the ansatz and allows us to have a deeper understanding of Phenom models.

The rest of the paper is structured as follows: In Sec. 2, we review the parameterization of the IMRPhenomD model and the mismatch function that is used as an objective function for the calibration, followed by outlining the specific optimization scheme used for recalibration. In Sec. 3, we give the optimization result by comparing mismatches of optimized waveforms with original waveforms. We also show how the optimization result differs with waveforms of different intrinsic parameters. In Sec. 4, we address the difference between our calibrating procedure with (**?**). We also explain how reduced spin parameterization affects the accuracy of the model.

## 2. METHOD

### 2.1. *Waveform Model*

In order to recalibrate the model, we have to understand what parameters the model has. Here we give a succinct summary of the IMRPhenomD model and the relevant parameters. For interested readers, please refer to (**?**) for more details on construction of the model.

The IMRPhenomD model is constructed by combining three individually fitted parts into one coherent waveform model, which consists of the inspiral, intermediate, and merger-ringdown part, [KW: Check if the following equation is correct. I think it is off by some smoothing factors. ] [KL: I don't remember there is any smoothing factors. They are just connected using step functions, then the entire waveform is made to have continuous first derivative by fixing some of the coefficients for the two connections. ]

$$h(f,\theta,\Lambda^i) = h_{\rm ins}(f,\theta,\Lambda^i) + h_{\rm int}(f,\theta,\Lambda^i) + h_{\rm rd}(f,\theta,\Lambda^i). \tag{1}$$

Instead of fitting the strain, which is a highly oscillatory function that is difficult to fit, the amplitude and phase are fitted since they are smoother functions. In each part, the amplitude and phase are made using simple functions of frequency such as polynomials or lorentzians. Specifically, the merger-ringdown amplitude is fitted by a lorentzian and the other parts are fitted using polynomials.

$$\begin{aligned}
A_0 &\equiv \sqrt{\frac{2\eta}{3\pi^{1/3}}} f^{-7/6} \\
A_{\rm ins}(f;\theta) &= A_{\rm PN}(f;\theta) + A_0 \sum_{i=1}^{3} \rho_i f^{(6+i)/3} \\
A_{\rm int}(f;\theta) &= A_0(\delta_0 + \delta_1 f + \delta_2 f^2 + \delta_3 f^3 + \delta_4 f^4) \\
A_{\rm rd}(f;\theta) &= A_0 \left[\gamma_1 \frac{\gamma_3 f_{\rm damp}}{(f-f_{\rm RD})^2 + (\gamma_3 f_{\rm damp})^2} e^{-\frac{\gamma_2 (f-f_{\rm RD})}{\gamma_3 f_{\rm damp}}}\right],
\end{aligned} \tag{2}$$

where $A_{\rm PN}$ is the post-newtonian expansion of the insprial amplitude up to order $A_0 f^2$, $f_{\rm damp}$ is the damping frequency, and $f_{\rm RD}$ is the frequency at ringdown. The ansatzes for phase can be found in (**?**). These simple analytic functions consists of parameters $\Lambda^i = \{\rho_i, \delta_i, \gamma_i\}$, which are defined as follows.

$$\begin{aligned}
\Lambda^i = {} & \lambda^i_{00} + \lambda^i_{10}\eta \\
& + (\chi_{\rm PN} - 1)(\lambda^i_{01} + \lambda^i_{11}\eta + \lambda^i_{21}\eta^2) \\
& + (\chi_{\rm PN} - 1)^2(\lambda^i_{02} + \lambda^i_{12}\eta + \lambda^i_{22}\eta^2) \\
& + (\chi_{\rm PN} - 1)^3(\lambda^i_{03} + \lambda^i_{13}\eta + \lambda^i_{23}\eta^2),
\end{aligned} \tag{3}$$

where $\lambda$ are fitting coefficients obtained during calibration, $\eta$ is the symmetric mass ratio, and $\chi_{\rm PN}$ is the post-Newtonian spin parameter, which is defined as

$$\chi_{\rm PN} = \frac{m_1\chi_1 + m_2\chi_2}{m_1 + m_2} - \frac{38\eta}{113}(\chi_1 + \chi_2). \tag{4}$$

Here, $m_{1,2}$ and $\chi_{1,2}$ are the primary and secondary mass and spin, respectively. Finally, the individual segments are first connected directly using step functions. Then, by fixing coefficients in the intermediate segment, one can make the final waveform is continuous in its first derivative.

Combining Eq. 1, 2 and 3, we can see that the entire waveform is non-linear in $\lambda$. A slightly inaccurate set

of $\lambda$ can significantly affect the shape of the generated waveforms. Thus, having a set of accurate waveform coefficients is important and fundamental to having accurate GW models. Generally, waveform coefficients are obtained by calibrating with NR waveforms, which are waveforms computed using NR simulations. In the case of (**?**), they first obtain a set of $\Lambda$ by fitting model generated waveforms to Eq. 2 and the phase ansatzes. Repeating with different NR waveforms, they obtain multiple sets of $\Lambda$, and $\lambda$ are subsequently found by fitting against Eq. 3. Since the fitting procedure is done in a piece-wise manner, the correlations between different segments are omitted, which could limit the accuracy of the model. Also, since fitting was performed before connecting individual segments, the final waveform does not guarantee to achieve the optimal waveform. The connecting procedure can alter the previously fitted waveform. Hence, the model generated waveforms contains additional inaccuracies.

Instead, we recalibrate coefficients jointly, which we can remove inaccuracies and biases discussed above, and can improve model accuracy. In the past, due to the complex nature of GW strains and piece-wise formalism of IMRPhenomD, non-linear fitting was difficult to be performed in optimizing coefficients. Hence, piecewise optimization was done to obtain coefficients. However, with `ripple` and AD from `jax`, gradients of IMRPhenomD can be easily obtained, thus allowing the use of gradient-based algorithms for us to recalibrate the model.

### 2.2. *Loss Function*

In order to recalibrate the model, we need to define an objective function that quantify the performance of the model. A common choice for such metric is the *mismatch* function (**?**). It is defined as

$$\mathcal{M}(h_1, h_2) = 1 - \max_{t_0,\phi_0} \langle \hat{h}_1, \hat{h}_2 \rangle, \tag{5}$$

where $h_{1,2}$ are the two GW waveform we are comparing, and $t_0$ and $\phi_0$ are time shift and phase shift respectively. The $\langle h_1, h_2 \rangle$ is commonly referred as the inner product, which is defined as

$$\langle h_1, h_2 \rangle = 4\,\mathrm{Re} \int_{f_{\min}}^{f_{\max}} \frac{h_1(f) h_2^*(f)}{S_n(f)}\, df, \tag{6}$$

where $\hat{h} = h/\sqrt{\langle h, h \rangle}$ is the normalized GW strain, $S_n(f)$ is the power spectral density(PSD) of noise from the instrument, $f_{\max}$ and $f_{\min}$ are the relevant maximum and minimum frequencies for the integration. The mismatch is a quantity that is closely related to the mean square error (MSE) between the two waveforms, as it is often considered a weighted version of the MSE.

[KL: The maximum is calculated by comparing the phase of IMRPhenomD and NR waveforms. We know that they are off by a linear relation of $f$,

$$\phi_{NR} - \phi_{IMR} = (2\pi t_0) f + \phi_0.$$

Since I have $\phi_{NR}$, $\phi_{IMR}$ and the frequency array, I did a linear regression to get $2\pi t_0$ and $\phi_0$. ]

Since we wish to optimize the model over the whole parameter space, we need to compare multiple model generated waveforms with NR waveforms. However, mismatch only quantifies the difference between IMRPhenomD and NR waveform for one particular set of intrinsic parameters. To take into account of various different waveforms in the parameter space, we pick waveforms from the different parts of the parameter space. We define the loss function as an average of training waveforms in two ways, the simple average of mismatches and the normalize average of mismatches,

$$\mathcal{L}_{\text{mean}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{M}_i \tag{7}$$

$$\mathcal{L}_{\text{fl}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathcal{M}_i}{\mathcal{M}_{i,\text{ini}}}, \tag{8}$$

where $\mathcal{M}_i$ represents the mismatch of an individual training waveform, $\mathcal{M}_{i,\text{ini}}$ represents the initial mismatch of the individual training waveform, and $N$ is the total number of individual training waveforms. Note that we choose to use two different averages, since they have different preferences in optimization base on waveform mismatches. For the first choice, simple average serves as the simplest choice of loss function, but is prone to be dominated by a single waveform with a large mismatch. Other waveforms with smaller mismatches would be insignificant comparatively, and might not be able to improve under such optimization. Alternatively, the second choice, normalized average eliminates the aforementioned issue. Nevertheless, it excludes the information on initial mismatches. $\mathcal{L}_{\text{fl}}$ restricts every training waveform to decrease at similar rates, hence it is hard to obtain optimized waveforms with mismatches in the same order of magnitude. Instead, their ratios in mismatches would remain approximately the same. Conversely, $\mathcal{L}_{\text{mean}}$ allows the loss function to automatically adjust and individual mismatches would be in a similar order of magnitude after optimization. In this paper, we showcase the results of using both loss functions and examine the differences between them.

### 2.3. *Optimization Scheme*

To compute the loss functions, we have to take NR waveforms for calculating the mismatch. We choose 11-16 NR waveforms from the set of waveforms used in

the original calibration process as training waveforms. Originally, 19 waveforms are taken from NR simulations for calibrating IMRPhenomD (**?**), which are waveforms from the SXS catalog (**?**) or BAM simulation. As BAM waveforms are not publicly available, we cannot take the identical training set as them. Instead, we take the available waveforms from the SXS catalog to construct our loss function. Training waveforms used are listed in Tab. 1 and 2. The training waveforms chosen has maximum mass ratio to be 8. This is because SXS catalog does not have NR waveforms with extremely high mass ratio. In fact, the SXS catalog only has NR waveforms with $q \leq 10$. Nevertheless, we are interested in the behavior of IMRPhenomD model with small $q$, as most BBH events observed from LIGO have $q \leq 8$. Hence, we calibrate IMRPhenomD with waveforms of $q \leq 8$.

In the SXS catalog, NR waveforms are in the form of time-series strain. Since time-series data is oscillatory, performing optimization in the time-domain is not ideal. Hence, we transform NR waveforms to frequency-domain to compare with IMRPhenomD waveforms with the same intrinsic parameters. We taper the time-series using Tukey window. [1] Then, the frequency spectra can be obtained by taking the Fourier transform of the time-series.

[TE: I think this discussion needs to be much more precise. Having a waveform model calibrated to particular noise curve could make sense, but its inherently different to just fitting the phase and the amplitude separately like they do in the original paper. So I would recommend that our default is to use a flat PSD and then in addition discuss the one with the PSD. This way, I think we demonstrate more clearly that its the high dimensional fitting that helps, not the just changing the metric used when fitting. ] [TE: This choice seems pretty arbitrary. Looking at Fig 8 of 2111.03606, they seem to cluster closer to 60-70. This choice is important and should be more informed. ]

Other than NR waveforms, one need to choose a relevant noise PSD for mismatch. We have opted to use a flat PSD for this purpose, as it provides results that are independent of the detector sensitivity and mass scale. The use of a flat PSD ensures that the improvement in accuracy is due solely to the difference in high-dimensional fitting and piece-wise fitting. Furthermore, we are interested in examining the effect of introducing a detector PSD on the optimization process. For this, we have chosen the zero-detuned high-power (`zdethp`) noise PSD (**?**). Since the total mass of the system scales with the frequency of the waveform, we must choose a corresponding mass scale to match the frequency range of our noise PSD. Therefore, we have selected a mass scale of $M = 50M_\odot$ for all waveforms, as this is a commonly observed mass scale in LIGO observations. We will then examine the differences in optimization.

We point out that our treatment to NR waveforms is different from that of (**??**). In the original calibration process, training waveforms are hybrid waveforms of NR and SpinAlignedEOB (SEOB) waveforms. The low frequency inspiral part is taken from the SEOB waveforms while the rest of the waveforms are taken from NR simulations. Instead, we solely use NR waveforms for comparison, since we are only exploring the possibility of optimizing waveform models. Thus, for simplicity sake, we ignore this procedure. On the other hand, most NR waveforms used (for both training and testing) have long enough time series data, i.e. $> 15$ orbits (**?**), in which they are long enough to contain part of the inspiral segment and all merger and ringdown frequency information. We take the frequency limits as $f_{\rm min} = 0.1 f_{\rm RD}$ and $f_{\rm max} = 1.2 f_{\rm RD}$, where $f_{\rm RD}$ is the frequency at ringdown. This range covers most of the IMRPhenomD's frequency range, except the minimum frequency is set higher than that in the original calibration due to NR length. When compared with IMRPhenomC, the frequency range is slightly extended to have a higher maximum frequency. We have the dimensionless frequency spacing $M\Delta f = 2.5\times10^{-6}$, which is sufficient to capture all features of GW strain.

With the loss function evaluated, we apply gradient descent to optimize the tunable coefficients as shown in Algorithm 1. We take $\lambda_i$ to be the original coefficients given in (**?**). We take them as the initial waveform coefficients because they lie in the neighborhood of the minimum that we wish to find. Then, by taking $\alpha = 10^{-6}$ and $N = 30000$, we can optimize the coefficients systematically. [TE: I think we need a bit more discussion about the stopping criterion here. What is N, and why did we choose it to be this number? Ideally we would actually have a plot of the loss function during training. I don't have a good intuition for if its noisy. Does it plateau? If we wanted to ensure that people don't complain about overfitting we could also plot the loss of the validation set to show its not going up. ]

## 3. RESULT AND COMPARISON WITH ORIGINAL MODEL

First, to judge the performance of the optimization scheme, we take $\sim$ 530 NR waveforms from the SXS catalog. We choose testing waveforms with

[1] Specifically, we choose $\alpha = 2t_{\rm RD}/T$, where $t_{\rm RD}$ is the duration of ringdown and $T$ is the duration of the entire GW strain.


**Algorithm 1:** Gradient descent pseudocode

**Input:** initial coefficients $\lambda_i$
**Parameters:** number of iterations $N$, learning rate $\alpha$
**Variables:** current coefficients $\lambda$, mismatch gradient $\nabla\mathcal{L}$
**Result:** output coefficients $\lambda$

$\lambda \leftarrow \lambda_i$
/* Gradient Descent */
**for** $i < N$ **do**
  $\mathcal{L} \leftarrow Mismatch(\lambda)$
  $\nabla\mathcal{L} \leftarrow AutoDiff(\mathcal{L})$
  $\lambda \leftarrow \lambda - \alpha\nabla\mathcal{L}$
**return** $\lambda$

negligible eccentricity ($e < 2 \times 10^{-3}$) and precession ($\chi_{x,y} < 5 \times 10^{-3}$) to fit with the limitation of the model. We see that in Fig. 1, testing waveforms have intrinsic parameters that are within the parameter space spanned by training waveforms. Hence, we can take these testing waveforms to compare with the original model.

We first examine the effect of joint optimization on a single waveform. In Fig. 2, we can see that the optimized waveform has better accuracy than the original waveform, especially in the inspiral region, where the amplitude has a 50% decrease in error. We have chosen one of the testing waveforms listed in (**?**) for a fair comparison.

Using a constant noise spectrum with $\mathcal{L}_{\text{mean}}$, we compute the mismatch for all testing waveforms and plot the distribution of the mismatch in Fig. 3. We find the distribution's peak has shifted to the end with a lower mismatch. Quantitatively, the peak has lowered by almost an order-of-magnitude and the median of distribution has a $\sim 50.0\%$ decrease. Using $\mathcal{L}_{\text{fl}}$, the mismatch distribution also has a similar improvement. We see that the median of distribution has a 22.9% decrease. However, the distribution does not have a clear peak as Fig. 3. This is due to the problem mentioned in section 2.3.

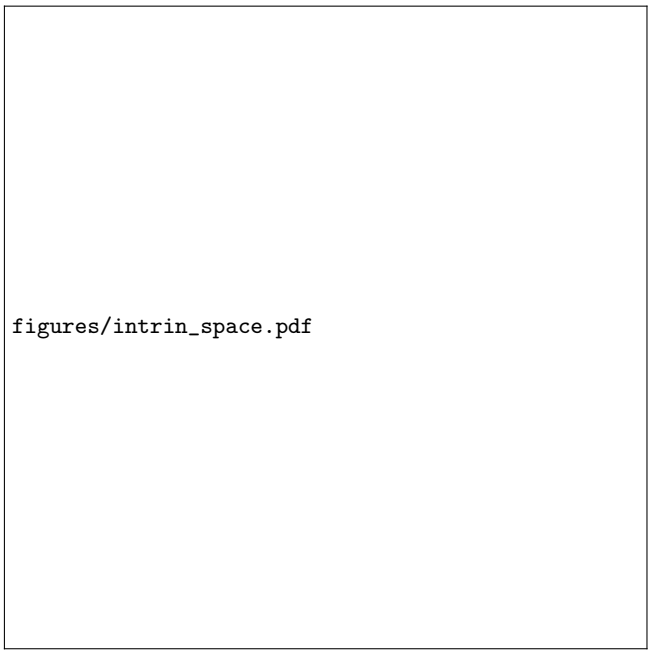


**Figure 1.** Parameter space with mass ratio against normalized reduced spin. Orange: Training waveforms; Blue: Testing waveforms

| Code | $q$ | $\chi_1$ | $\chi_2$ |
|---|---|---|---|
| SXS:BBH:0156 | 1.0 | -0.95 | -0.95 |
| SXS:BBH:0151 | 1.0 | -0.60 | -0.60 |
| SXS:BBH:0001 | 1.0 | 0.00 | 0.00 |
| SXS:BBH:0152 | 1.0 | 0.60 | 0.60 |
| SXS:BBH:0172 | 1.0 | 0.98 | 0.98 |
| SXS:BBH:1418 | 4.0 | -0.40 | -0.50 |
| SXS:BBH:0167 | 4.0 | 0.00 | 0.00 |
| SXS:BBH:1417 | 4.0 | 0.40 | 0.50 |
| SXS:BBH:0064 | 8.0 | -0.50 | -0.46 |
| SXS:BBH:0063 | 8.0 | 0.00 | 0.00 |
| SXS:BBH:0065 | 8.0 | 0.50 | 0.46 |

**Table 1.** List of waveforms used to recalibrate the model. The mass ratio $q = m_1/m_2 \geq 1$ with spins $\chi_{1,2}$. Out of the 11 waveforms listed here, 9 of them are also used in the original IMRPhenomD calibration. (**?**) The two remaining waveforms were from BAM simulation, to which we do not have access.

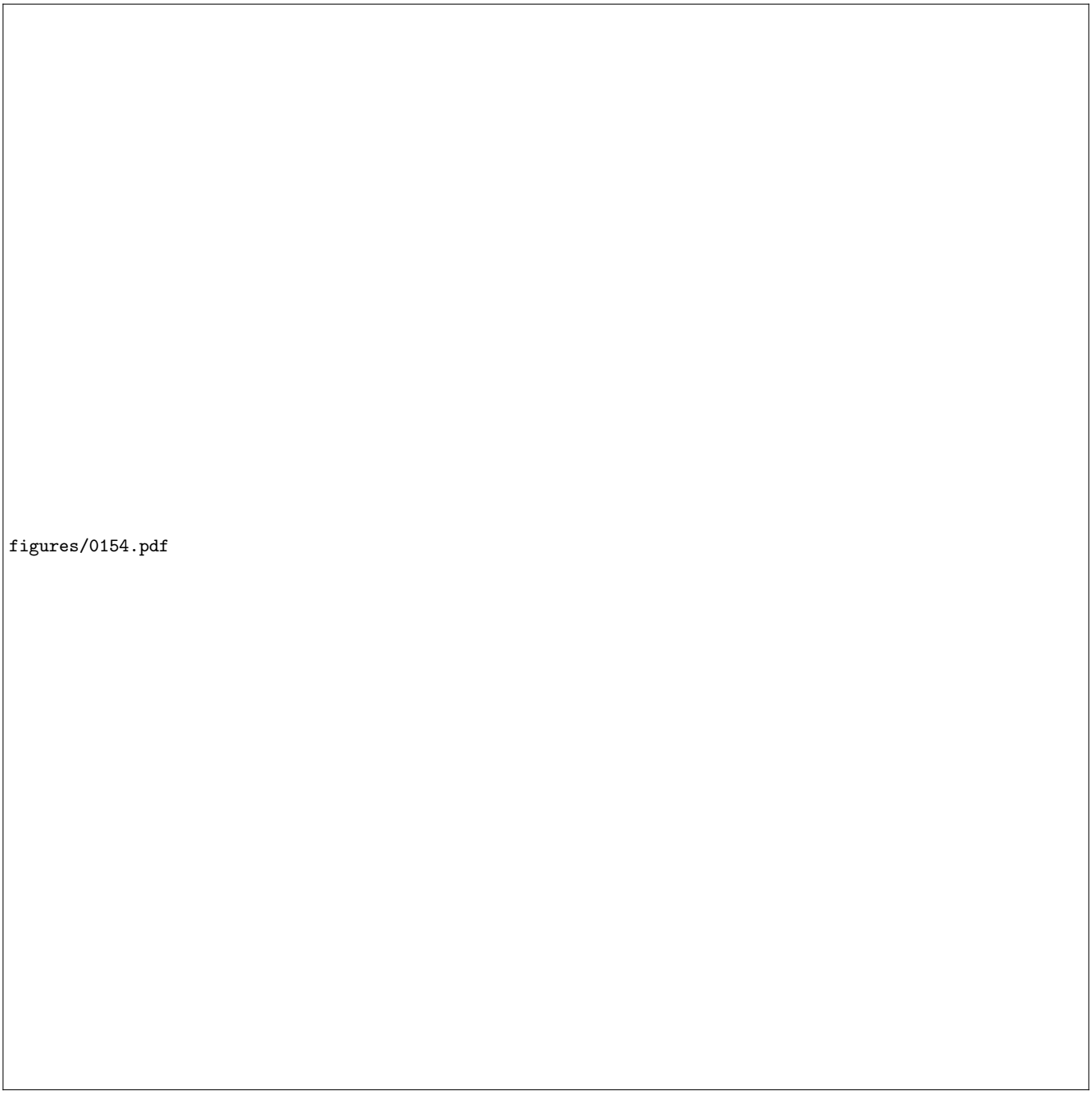


**Figure 2.** Comparison between original and optimized IMRPhenomD waveforms. Here shows the SXS:BBH:0154 NR waveform, which has mass ratio $q = 1$ and $\chi_1 = \chi_2 = -0.8$. The original mismatch is around $2.8 \times 10^{-4}$ and the optimized mismatch is around $5.3 \times 10^{-5}$. Top: It shows the amplitude and phase of NR, original IMRPhenomD and optimized IMRPhenomD waveform. Bottom: It shows the relative error of amplitudes between NR and IMRPhenomD waveforms, and the absolute error of phases between NR and IMRPhenomD waveforms

| Code | $q$ | $\chi_1$ | $\chi_2$ |
|---|---|---|---|
| SXS:BBH:0234 | 2.0 | -0.85 | -0.85 |
| SXS:BBH:0235 | 2.0 | -0.60 | -0.60 |
| SXS:BBH:0169 | 2.0 | 0.00 | 0.00 |
| SXS:BBH:0256 | 2.0 | 0.60 | 0.60 |
| SXS:BBH:0257 | 2.0 | 0.85 | 0.85 |

**Table 2.** Additional waveforms used in further recalibration.

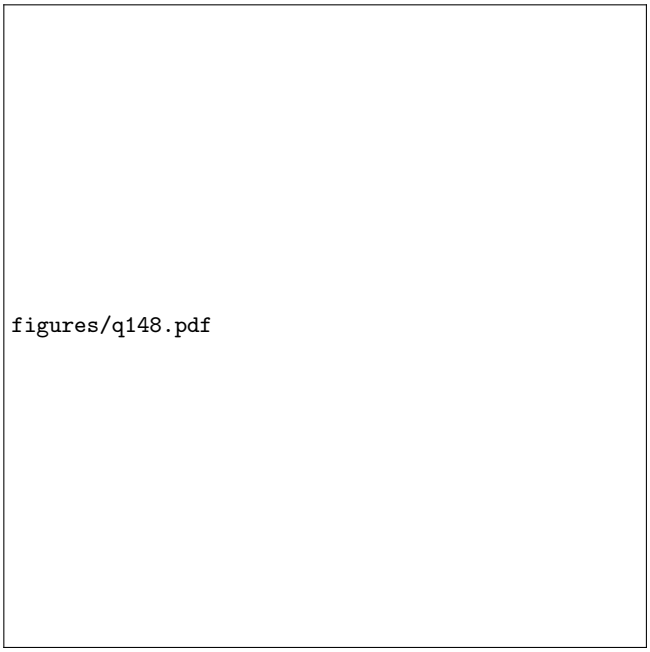


**Figure 3.** Distributions of waveform mismatches calculated using both $\mathcal{L}_{\rm mean}$ and $\mathcal{L}_{\rm fl}$ in recalibration. We use training waveforms listed in Tab. 1 and mismatches are weighted with the constant noise spectrum. For the $\mathcal{L}_{\rm mean}$ distributions, the median decreased by 50.0% while the median of $\mathcal{L}_{\rm fl}$ distribution decreased by 22.9%.

Nevertheless, both distributions show improvement and the potential problem did not affect the optimization scheme significantly.

Similarly, using the same procedures, distributions of mismatches calculated using `zdethp` noise spectrum show better improvement than the weighted mismatch. The shape of the distribution is similar to Fig. 3. This effect is expected since the IMRPhenomD model was originally constructed and fitted using the `zdethp` weighted mismatch. Thus, it should be a model that fits closely to the NR waveforms with the influence of `zdethp` noise spectrum instead of the constant spectrum.

With the success of improving waveforms, we increase the number of training waveforms for optimization. Taking additional waveforms listed in Tab. 2, we obtain a


**Figure 4.** Distributions of $\log_{10}$ difference in mismatch. The distribution labeled `q148` uses training waveforms listed in Tab. 1 while the `q1248` distribution uses waveforms listed in Tab. 1 and 2. Mismatches are calculated using the constant noise spectrum with the loss function $\mathcal{L}_{\rm mean}$.

new set of coefficients. We see in Fig. 4, new waveforms produced only have a very small improvement. The high mismatch tail of the optimized distribution remains similar in length and endpoint as the original distribution, meaning that they cannot be improved using our procedure. Likewise, using additional waveforms to optimize loss function with `zdethp` spectrum shows the same result, where the new distribution barely shows any improvement.

Given the ansatz used in the waveform model is unlikely to be fully compatible with NR, and the optimization procedure is done over a distribution of waveforms with different source parameters, it is conceivable that there are some trade-off in accuracy of the waveform models between different part of the parameter space. If this is the reason why the high mismatch tail is not improving during the joint-optimization, separating the parameter space into smaller subspace should help alleviate this issue. On the other hand, if the ansatz does not have the right parameterized form to capture the behavior of the NR waveforms as a function of the intrinsic parameters, the result should be always biased, and we should not expect any improvement even if we separate the parameter space into smaller space during training.

Since we know intrinsic parameters play an important role in the ansatz, we would like to investigate how in-

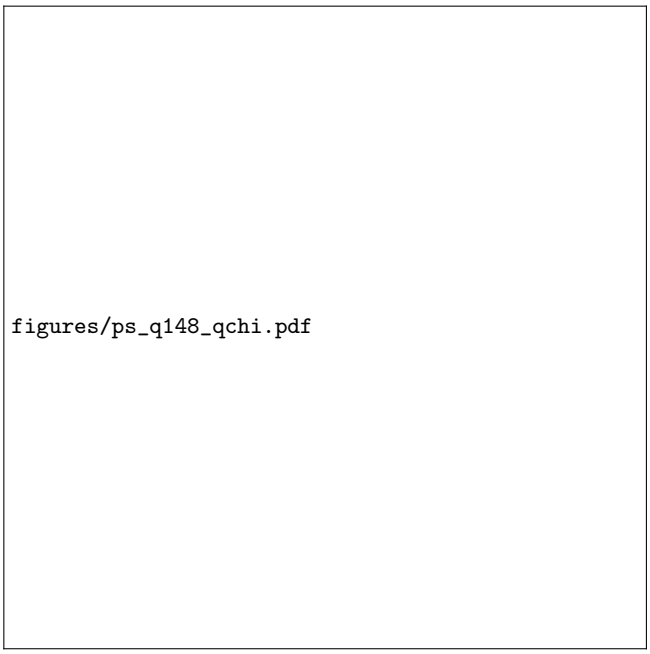


**Figure 5.** Parameter space of testing waveforms of $q$ vs. $\chi_{\rm PN}$. We use the recalibrated result from $\mathcal{L}_{\rm mean}$ with the constant noise spectrum and training waveforms in Tab. 1. Here, the colorbar represents the $\log_{10}$ difference between optimized and original unweighted mismatches.


**Figure 6.** Parameter space of testing waveforms of $\chi_1$ vs. $\chi_2$. We use the recalibrated result from $\mathcal{L}_{\rm mean}$ with the constant noise spectrum and training waveforms in Tab. 1.

trinsic parameters affect the recalibration process. First, we plot the parameter space of $q$ vs. $\chi_{\rm PN}$ in Fig. 5. In the low mass ratio region, waveforms with both positive and negative log differences are mixed up. On the other hand, along the horizontal line $\chi_{\rm PN} = 0$, waveform mismatches consistently improve under recalibration.

Furthermore, we plot the parameter space of $\chi_1$ vs. $\chi_2$ in Fig. 6. Waveforms along the diagonal axis, i.e. $\chi_1 \approx \chi_2$, show good mismatch improvements. We expect to see this feature since the original coefficients were fitted using NR waveforms with equal or similar spin, hence the model prefers waveforms with similar spin. One interesting feature is how the second and fourth quadrants respond to optimization. In the second quadrant ($\chi_1 < 0$ and $\chi_2 > 0$), waveforms generally improve with along optimization. However, mismatches in the fourth quadrant ($\chi_1 > 0$ and $\chi_2 < 0$) do not improve after optimization. Most waveforms even turned worse after optimization. These waveforms correspond to the waveforms in the high mismatch tail in Fig. 3.

As we see in Fig. 6 that the recalibration procedure is significantly different in different regions in the parameter space, we split the parameter space into 4 quadrants and perform separate fitting with training waveforms listed in Table 3. Note that in the second and fourth quadrants, there are not enough waveforms with $q > 4$, hence the result are only valid up to $q \leq 4$. From Fig. 8, we see that all waveforms, except those in the fourth quadrant, show improvement in mismatch. Many features seen in Fig. 6 can be found here again. First, we focus on waveforms with the same spin direction, i.e. $\chi_1\chi_2 > 0$. Waveforms lying in the neighborhood of the diagonal axis have noticeable improvements in mismatch due to most training waveforms being equal-spin waveforms. For the second quadrant, waveforms improved significantly with only a few defects due to some testing waveforms having $q > 4$. In the fourth quadrant, most optimized waveforms have a higher mismatch than the original waveforms, as indicated by the positive log difference of mismatches. As performing optimization in a smaller subspace does not show any improvement compared to the original result, the ansatz mostly does not fit waveforms in the fourth quadrant.

## 4. DISCUSSION

We have shown the result of recalibrating waveform coefficients. One thing to note is that our recalibration procedure is not exactly the same as the original calibration. For instance, we use a different set of NR waveforms, frequency range, etc. Nonetheless, as the decrease in mismatch is rather significant, this optimization procedure should be able to improve the accuracy of IMRPhenomD on a similar scale regardless of the differences. Here, this result serves as a demonstration of the general method used.

| Code | $q$ | $\chi_1$ | $\chi_2$ |
|---|---|---|---|
| SXS:BBH:0172 | 1.0 | 0.98 | 0.98 |
| SXS:BBH:0152 | 1.0 | 0.60 | 0.60 |
| SXS:BBH:0001 | 1.0 | 0.00 | 0.00 |
| SXS:BBH:1417 | 4.0 | 0.40 | 0.50 |
| SXS:BBH:0167 | 4.0 | 0.00 | 0.00 |
| SXS:BBH:1426 | 8.0 | 0.48 | 0.75 |
| SXS:BBH:0167 | 8.0 | 0.00 | 0.00 |
| SXS:BBH:0370 | 1.0 | -0.20 | 0.40 |
| SXS:BBH:2092 | 1.0 | -0.50 | 0.50 |
| SXS:BBH:0330 | 1.0 | -0.80 | 0.80 |
| SXS:BBH:2116 | 2.0 | -0.30 | 0.30 |
| SXS:BBH:2111 | 2.0 | -0.60 | 0.60 |
| SXS:BBH:0335 | 2.0 | -0.80 | 0.80 |
| SXS:BBH:0263 | 3.0 | -0.60 | 0.60 |
| SXS:BBH:2133 | 3.0 | -0.73 | 0.85 |
| SXS:BBH:0263 | 4.0 | -0.80 | 0.80 |
| SXS:BBH:0156 | 1.0 | -0.95 | -0.95 |
| SXS:BBH:0151 | 1.0 | -0.60 | -0.60 |
| SXS:BBH:0001 | 1.0 | 0.00 | 0.00 |
| SXS:BBH:1418 | 4.0 | -0.40 | -0.50 |
| SXS:BBH:0167 | 4.0 | 0.00 | 0.00 |
| SXS:BBH:1419 | 8.0 | -0.80 | -0.80 |
| SXS:BBH:0063 | 8.0 | 0.00 | 0.00 |
| SXS:BBH:0304 | 1.0 | 0.50 | -0.50 |
| SXS:BBH:0327 | 1.0 | 0.80 | -0.80 |
| SXS:BBH:2123 | 2.0 | 0.30 | -0.30 |
| SXS:BBH:2128 | 2.0 | 0.60 | -0.60 |
| SXS:BBH:2132 | 2.0 | 0.87 | -0.85 |
| SXS:BBH:2153 | 3.0 | 0.30 | -0.30 |
| SXS:BBH:0045 | 3.0 | 0.50 | -0.50 |
| SXS:BBH:0292 | 3.0 | 0.73 | -0.85 |

**Table 3.** List of waveforms used in recalibrating coefficients in 4 quadrants. From top to down are the first, second, third and fourth quadrants. Note that for the first and third quadrants, waveforms are chosen to have equal or similar spins, while the training waveforms for the second and fourth quadrants are chosen to have opposite spins.

We see that in Fig. 4, performing optimization with more training waveforms only has a small increase in the accuracy of the waveform model. We believe that by increasing the number of waveforms, the accuracy will not have a significant change as the waveform coefficients are already over-determined. Using more calibration NR waveforms will not further improve the model significantly. This suggests the form of the parameterized ansatz is not suitable for certain regions in the parameter space, thus mismatches of only a few waveforms decreased while other waveforms remain at the high mismatch tail with little changes. This implies that the

figures/all_quadrants.pdf

**Figure 7.** Distributions of mismatches after optimizing in separate quadrants. We use a constant noise spectrum to calculate mismatch and $\mathcal{L}_{\rm mean}$ for the loss function. Generally, most waveforms with mismatch $> 10^{-2}$ lies in the fourth quadrant.

figures/ps_q148_quadrants.pdf

**Figure 8.** Parameter space of testing waveforms. Each quadrant is fitted independently. Colorbar represents log difference of mismatches before and after optimization.

model is ultimately restricted by the flexibility of the ansatz.

One of the major problems causing inaccuracy in the ansatz is the reduced spin approximation. In IMRPhenomD, it is modeled using a single spin parameter, namely $\chi_{\rm PN}$ as outlined in Sec. 2. Parameterizing a BBH merger with one spin parameter introduces degeneracy within the parameter space. Events with distinct black hole spins could result in equal $\chi_{\rm PN}$, thus generating the same waveform. Especially with high unequal spin events, $\chi_{\rm PN}$ would identify them as the same as events with small spin, thus giving an inaccurate result. Generally, the approximation gives straight lines of degeneracy in the parameter space, with its slope (always negative) dependent on the mass ratio. From Fig. 6, we see that along a degeneracy line, the ansatz behaves better in the top left region than the bottom right. To try to accommodate this issue, we split the parameter space into 4 quadrants as described in Sec. 3. However, even with separate optimizations, we see in Fig. 8 that the fourth quadrant still shows similar mismatches as before while the second quadrant further improved. This suggests the ansatz is region-specific, with a higher preference for BBH events with $\chi_1 < 0$ and $\chi_2 > 0$.

Separating regions into 4 quadrants is done purely out of simplicity. To give a more comprehensive analysis, one should be systematic about region selection. One can use level set estimation algorithms to obtain systematic regions of interest. This general algorithm reveals further degeneracies or issues within the ansatz. Then, recalibrating such individual regions might give better results. Alternatively, one can select regions according to the lines of degeneracy. However, with limited NR waveforms, such a selection scheme is not viable for us. In the future, with more NR waveforms spanning the entire parameter space, one can perform optimization with fewer restrictions.

While our work focused mainly on the IMRPhenomD model, this simple yet general method can be utilized in other differentiable GW models. For instance, within the same family, IMRPhenomX (**?**) or IMRPhenomP (**?**) models. By jointly fitting a new set of coefficients, it is expected that both models can be improved since they are constructed in a similar way as the IMRPhenomD model. For example, they also use PN approximant as part of the ansatz in the inspiral segment. One interesting result might arise while recalibrating IMRPhenomX model (**?**). Since it is parameterized by an additional anti-symmetric spin parameter, it is expected to not show the same degeneracy as described above. Further analysis might give insights into the systematics of Phenom models. Moreover, this method can be applied to other GW model families, such as NR surrogate models (**?**) or EOB models (**?**). NR waveform calibration procedures could be made easier and are likely to improve current models.

## 5. CONCLUSION

In this paper, we have presented a systematic method to recalibrate GW models. This method utilizes `jax`'s automatic differentiation to apply derivative-based optimization to recalibrate GW models jointly. Using the new implementation of the IMRPhenomD model, `ripple`, which is written in `jax`, in conjunction with NR waveforms from the SXS catalog, we recalibrate waveform coefficients of the IMRPhenomD model. In general, the waveform accuracy can be improved by 50%. Comparing `zdethp` weighted and unweighted mismatch, weighted mismatches have a slightly better improvement. In contrast, different types of loss function result in significantly different final mismatch distributions, where the result can be seen in Fig. 3. By increasing the number of training waveforms, we see a slight improvement increase in Fig. 4.

Furthermore, we investigated how the intrinsic parameters affect the improvement. Fig. 6 shows that the optimization procedure has a certain preference for waveforms lying in the second quadrant while the fourth quadrant cannot be improved. To further test this result, we recalibrate waveforms in separate regions in parameter space. As shown in Fig. 8, this recalibration process gives further improvement to the second quadrant while the fourth quadrant shows similar result. This indicates that the model's ansatz does not fit waveforms in the fourth quadrant. This phenomenon is due to the reduced spin approximation used in parameterizing the ansatz, where degeneracies between $\chi_1$ and $\chi_2$ are introduced.

While we naively separate the optimization process into 4 quadrants, one can perform systematic region-selection. In principle, we can apply this general method to other newer and more accurate models such as IMRPhenomX or IMRPhenomP models. Then, we can perform all the above analyses to understand how to construct better GW Phenom models in the future.

## 6. ACKNOWLEDGMENTS