Draft version April 3, 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

[TE: General points:

- Overall the text needs to be refined
- Plot styles should be made more uniform

]

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 (Abbott et al. 2020). 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 (Pürrer & Haster 2020). 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) (Taracchini et al. 2014), Numerical Relativity (NR) surrogate (Varma et al. 2019) and phenomenological (Phenom) models (Husa et al. 2016; Khan et al. 2016). 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` (Paszke et al. 2019), `tensorflow` (Abadi et al. 2015), etc. utilizes AD to train machine learning models. AD's algorithm is intuitive in nature. Functions defined are

kelvin33550336@gmail.com

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 step and the desired derivative can then be obtained by composing back the original function according to the original structure. `ripple` (Edwards et al. 2023) 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. 2.1, we explain how IMRPhenomD is constructed and have a detailed explanation of how waveform coefficients are introduced to the model. We discuss the mismatch in Sec. 2.2, where we explained how we calculate it in conjunction with NR waveforms. Then, we describe the optimization scheme used for recalibration in Sec. 2.3. 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 (Khan et al. 2016). We also explain how reduced spin parameterization affects the accuracy of the model.

2. METHOD

2.1. *Waveform Model*

IMRPhenomD is a frequency-domain phenomenological model for modeling spin-aligned BBH coalescence events. It was developed as an overhaul of previous Phenom models, such as IMRPhenomB/C (Santamaria et al. 2010; Ajith et al. 2011), to give higher accuracy for GW data analyses. IMRPhenomD is constructed in separate segments, i.e. inspiral, intermediate, and merger-ringdown.

$$h(f,\theta) = h_{\rm ins}(f,\theta) + h_{\rm int}(f,\theta) + h_{\rm me-rd}(f,\theta) \tag{1}$$

In each segment, the amplitude and phase are made using simple functions of frequency such as polynomials or lorentzians. These simple analytic functions consists of parameters Λ^i, which are defined as follows.

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

Λ^i depends on waveform coefficients λ and intrinsic parameters including symmetric mass ratio η and post-newtonian spin parameter $\chi_{\rm PN}$,

$$\chi_{\rm PN} = \frac{m_1\chi_1 + m_2\chi_2}{m_1 + m_2} - \frac{38\eta}{113}(\chi_1 + \chi_2). \tag{3}$$

Here, $m_{1,2}$ and $\chi_{1,2}$ are the mass and spin of larger and smaller black hole respectively. Finally, the individual segments are connected in a way that the final waveform is continuous in its first derivative.

From Eq. 2, we can see that the values of λ are fundamental to waveform generation, which can significantly affect the shape of final waveforms. Thus, having a set of accurate waveform coefficients is important. Generally, waveform coefficients are obtained by calibrating with NR waveforms, which are waveforms computed using NR simulations. In the case of (Khan et al. 2016), λ were obtained by fitting model-generated waveforms to NR waveforms. For each NR simulation, they can obtain one set of Λ's through optimization. With many sets of Λ, λ are subsequently found by fitting against Eq. 2. However, this piece-wise fitting deployed by (Khan et al. 2016) causes calibrated coefficients to have systematic biases. Since coefficients were not fitted jointly, correlations between segments are omitted. Hence, the model introduces systematic biases to λ and will be accumulated and passed along to the data analysis pipeline. Also, their calibration procedure does not guarantee to have the optimal set of λ, since the process of connecting segments alters the previously fitted waveform. The model-generated waveforms do not take this effect into account, and thus introduces 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, piece-wise 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. *Mismatch*

To quantify the difference between waveforms, we use the standard *mismatch* $\mathcal{M}$ as the metric (Husa et al. 2016). It is defined as

$$\mathcal{M}(h_1, h_2) = 1 - \max_{t_0,\phi_0} \langle \hat{h}_1, \hat{h}_2 \rangle, \tag{4}$$

where $h_{1,2}$ represents GW strain in frequency domain, t_0 and ϕ_0 are time shift and phase shift respectively. The inner product is defined as

$$\langle h_1, h_2 \rangle = 4\,\mathrm{Re} \int_{f_{\min}}^{f_{\max}} \frac{h_1(f) h_2^*(f)}{S_n(f)}\, df, \tag{5}$$

where $\hat{h} = h/\sqrt{\langle h, h \rangle}$ is the normalized GW strain, $S_n(f)$ is the noise spectrum, $f_{\max}$ and $f_{\min}$ are the maximum and minimum frequencies in GW frequency spectra.

In the original calibration (Khan et al. 2016; Husa et al. 2016), training waveforms are hybrid waveforms of NR simulations 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 most NR waveforms used have long enough time series data, i.e. > 15 orbits (Boyle et al. 2019), 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 $f_{\min} = 0.1 f_{\mathrm{RD}}$ and $f_{\max} = 1.2 f_{\mathrm{RD}}$, where f_{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. [TE: What is the frequency spacing?] [TE: Why do we not just go to the cutoff of 0.2, or do the NR waveforms end earlier? Think these choices need more motivation.]

[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.] We choose to use two different noise spectra $S_n(f)$, namely the zero-detuned high-power (`zdethp`) noise spectrum and a constant spectrum. Since in any data analysis tasks, model-generated waveforms will ultimately be used to compare with GW observations, they are subjected to the sensitivity of detectors. Hence, it is sensible to have waveforms calibrated with `zdethp` spectrum rather than a non-weighted spectrum. However, with sensitivity curve involved, we have to choose a corresponding mass scale. Here, we simply choose $M = 50 M_\odot$ for all waveforms, since this mass scale is common in LIGO observations. [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.] On the other hand, we also recalibrated coefficients with the constant spectrum for a more general purpose.

In our work, mismatches are used as a measure of discrepancies between IMRPhenomD waveforms and NR waveforms. To compute such mismatches, NR waveforms from the SXS catalog are taken (Boyle et al. 2019). These NR waveforms are calculated using the `SpEC` code. Initially, NR waveforms are in the form of time-series data. Since time-series data is oscillatory, performing optimization in the time-domain is not ideal. Hence, we had chosen to 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. Then, mismatches can be calculated through Eq. 4 and 5.

2.3. *Optimization Scheme*

Using `ripple`, we can make use of `jax` to evaluate gradients accurately. Thus, we can utilize simple gradient descent to recalibrate the model.

To recalibrate the model, we need to define a loss function that quantifies the difference between the model-generated waveforms and NR waveforms. We use mismatch over the set of all training waveforms as the loss function. While in (Khan et al. 2016), they chose to use the mean square error as the loss function, we choose mismatch as the loss function because it is closely related to downstream objectives such as the likelihood in

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

parameter estimation. The mismatch is averaged over the training dataset in two ways to cover the parameter space of interest. One of which takes a simple average and the other takes a normalized average based on initial mismatch.

$$\mathcal{L}_{\rm mean} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{M}_i \tag{6}$$

$$\mathcal{L}_{\rm fl} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathcal{M}_i}{\mathcal{M}_{i,\rm ini}}, \tag{7}$$

$\mathcal{M}_i$ represents the mismatch of an individual training waveform, $\mathcal{M}_{i,\rm ini}$ represents the initial mismatch of the individual training waveform, and N is the total number of individual training waveforms. For the first choice, it takes the mean of the mismatches, which serves as a basic loss function. However, it 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, the mean of normalized mismatches, eliminates the aforementioned issue. Nevertheless, it excludes the information on initial mismatches. $\mathcal{L}_{\rm fl}$ 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, $\mathcal{L}_{\rm mean}$ allows the loss function to automatically adjust and final individual mismatches would be in a similar order of magnitude. In this paper, we display the results of using both loss functions and examine the difference between them.

We have chosen 11-16 NR waveforms from the set of waveforms used in the original calibration process. Originally, 19 waveforms are taken from NR simulations for calibrating IMRPhenomD (Khan et al. 2016; Husa et al. 2016), which they 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. They are listed in Tab. 1 and 2. Notice the maximum mass ratio in our training set is lower than that in (Khan et al. 2016). Instead of having $q \leq 18$, the maximum mass ratio is set as 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

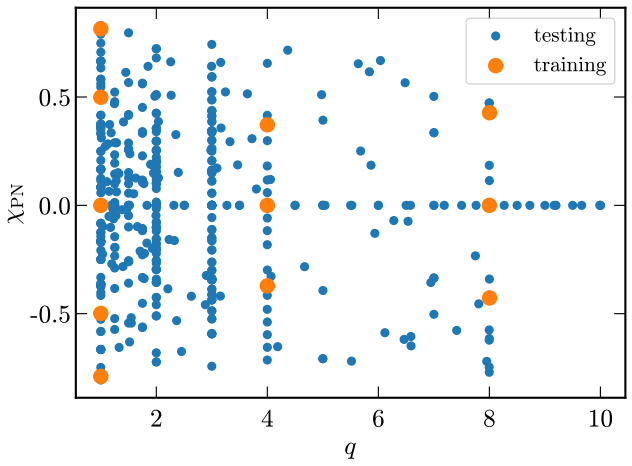

Figure 1. Parameter space with mass ratio against normalized reduced spin. Orange: Training waveforms; Blue: Testing waveforms

observed have $q \leq 8$. Hence, we set the maximum mass ratio to 8.

[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.]

With both mismatches, we apply gradient descent to optimize the tunable coefficients as follows: Here, λ_i are

Algorithm 1: Gradient descent pseudocode

Input: initial coefficients λ_i
Parameters: number of iterations N, learning rate α
Variables: current coefficients λ, mismatch gradient $\nabla\mathcal{L}$
Result: output coefficients λ

$\lambda \leftarrow \lambda_i$
`/* 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 λ

the original coefficients given in (Khan et al. 2016). We take them as the initial waveform coefficients because they lie in the neighborhood of the minimum that we wish to find. With such an optimization procedure, we obtain the optimized coefficients.

3. RESULT AND COMPARISON WITH ORIGINAL MODEL

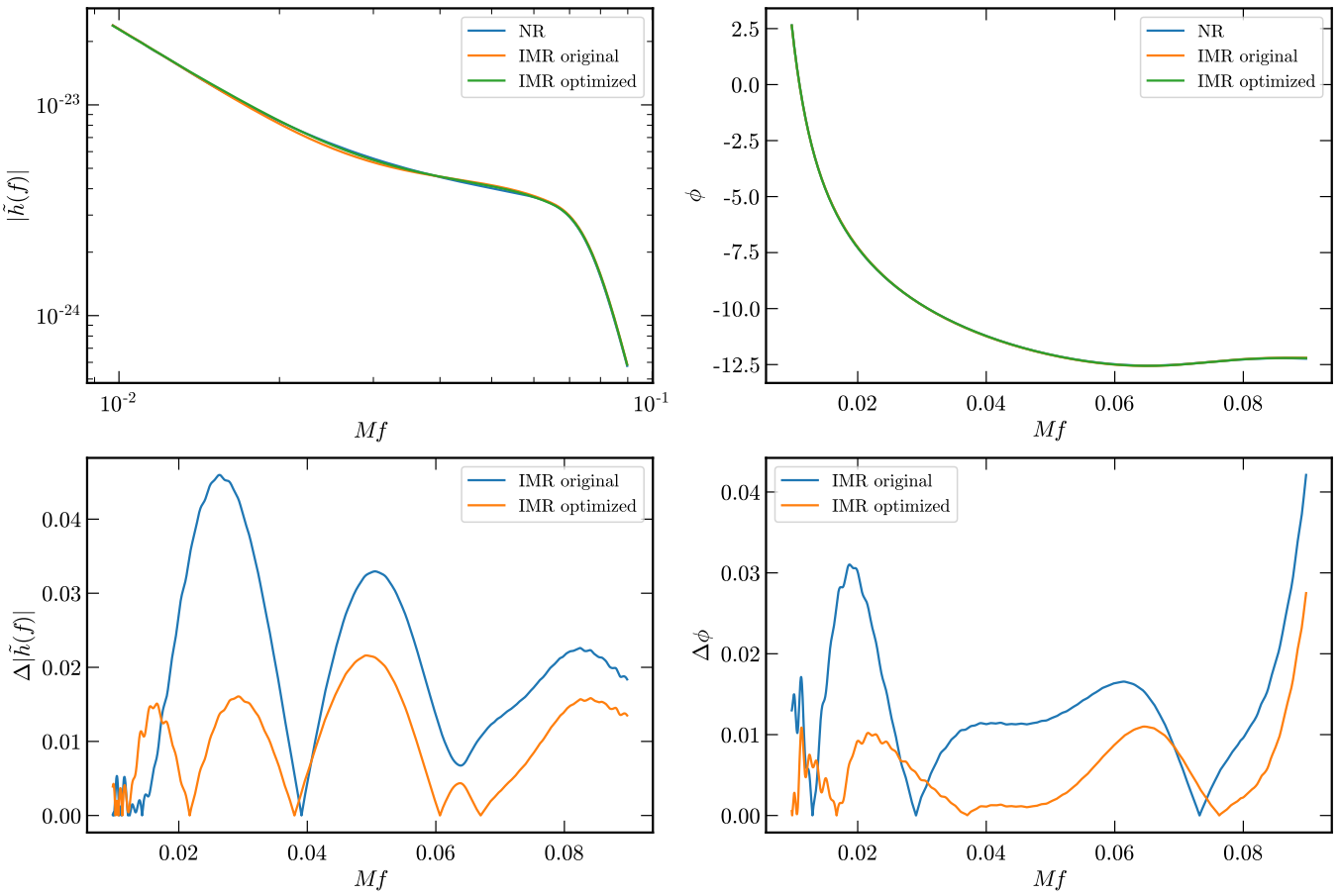

Figure 2. Comparison between original and optimized IMRPhenomD waveforms. Here shows the SXS:BBH:0154 NR waveform, which has mass ratio $q = 1$ and $\chi_1 = \chi_2 = -0.8$. The original mismatch is around 2.8×10^{-4} and the optimized mismatch is around 5.3×10^{-5}. Top: It shows the amplitude and phase 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 [TE: I realize that this is also an error in the ripple paper, but the bottom panels should have abs round the whole y axis, otherwise why are they strictly positive? I actually think just plotting the error itself i.e. allowing for positive and negative values would potentially look better. Its also useful to add the vertical lines corresponding the different waveform regions. Otherwise one might think that the oscillations will line up with this and its some weird artifact of the splitting. The top and bottom panels should probably share x axes, or at least have the same scales so that they can be compared easily.]

First, to judge the performance of the optimization scheme, we take $\sim$ 530 NR waveforms from the SXS catalog. We choose testing waveforms with 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 (Khan et al. 2016) for a fair comparison.

Using a constant noise spectrum with $\mathcal{L}_{\rm mean}$, 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 $\mathcal{L}_{\rm fl}$, 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. Nevertheless, both distributions show improvement and

Code	q	χ_1	χ_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. (Khan et al. 2016) The two remaining waveforms were from BAM simulation, to which we do not have access.

Code	q	χ_1	χ_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.

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 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.

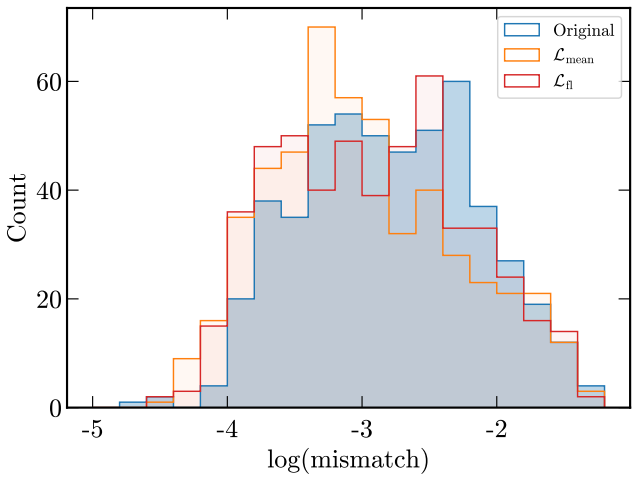

Figure 3. Distributions of waveform mismatches calculated using both $\mathcal{L}_{\rm mean}$ and $\mathcal{L}_{\rm fl}$ in recalibration. We use training waveforms listed in Tab. 1 and mismatches are weighted with the constant noise spectrum. For the $\mathcal{L}_{\rm mean}$ distributions, the median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

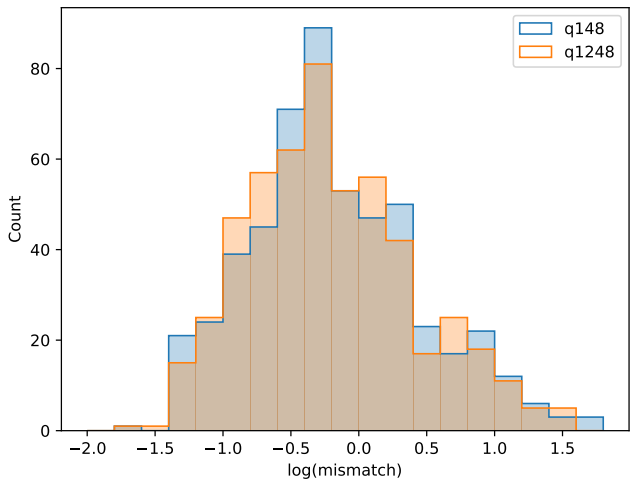

Figure 4. Distributions of $\log_{10}$ difference in mismatch. The distribution labeled `q148` uses training waveforms listed in Tab. 1 while the `q1248` distribution uses waveforms listed in Tab. 1 and 2. Mismatches are calculated using the constant noise spectrum with the loss function $\mathcal{L}_{\rm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

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 intrinsic 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 χ_1 vs. χ_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. [TE: I think this kind of behaviour needs to be matched with the discussion about our stopping criteria. If it gets worse, why aren't we stopping earlier?]

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.

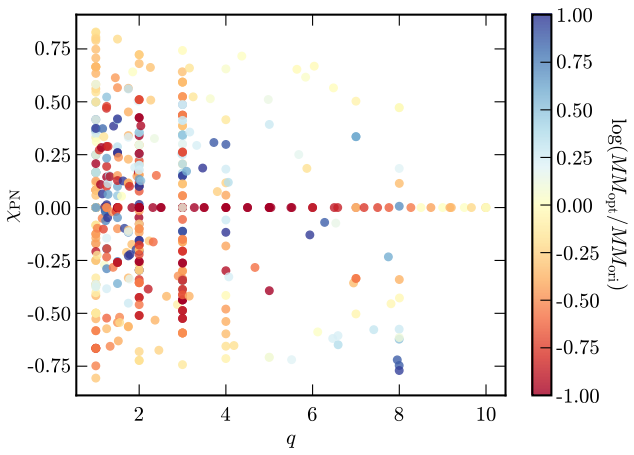

Figure 5. Parameter space of testing waveforms of q vs. $\chi_{\rm PN}$. We use the recalibrated result from $\mathcal{L}_{\rm mean}$ with the constant noise spectrum and training waveforms in Tab. 1. Here, the colorbar represents the $\log_{10}$ difference between optimized and original unweighted mismatches. [TE: I find this plot a little odd. The bounding of between -1 and 1 gives the impression that the colourbar has strict limits between these regions. Are we truncating the colors?And if so, why?]

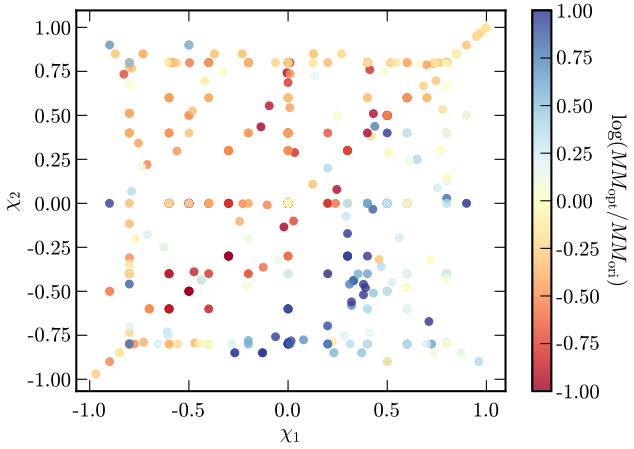

Figure 6. Parameter space of testing waveforms of χ_1 vs. χ_2. We use the recalibrated result from $\mathcal{L}_{\rm mean}$ with the constant noise spectrum and training waveforms in Tab. 1.

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

Code	q	χ_1	χ_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.

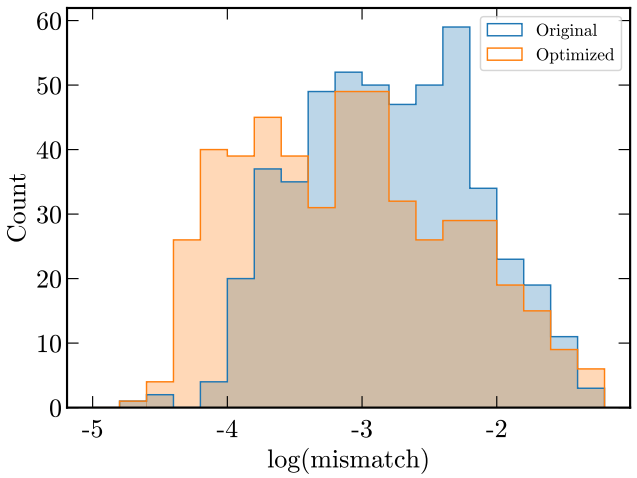

Figure 7. Distributions of mismatches after optimizing in separate quadrants. We use a constant noise spectrum to calculate mismatch and $\mathcal{L}_{\rm mean}$ for the loss function. Generally, most waveforms with mismatch $> 10^{-2}$ lies in the fourth quadrant. [TE: It would be nice to have four histograms for the four quadrants here. Ideally for both the original and optimized waveform but this might get a little cluttered. Maybe just plot the best and worst quadrant?][TE: This figure also doesn't appear to be referenced in the text?]

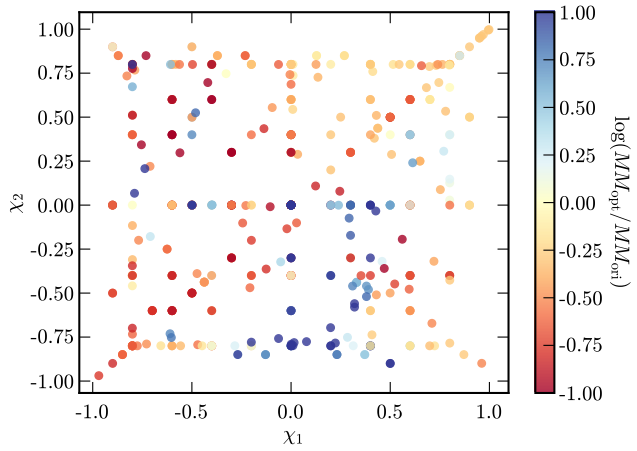

Figure 8. Parameter space of testing waveforms. Each quadrant is fitted independently. Colorbar represents log difference of mismatches before and after optimization.

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.

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

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 (Pratten et al. 2020) or IMRPhenomP (Hannam et al. 2014) 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 (Pratten et al. 2020). 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 (Varma et al. 2019) or EOB models (Taracchini et al. 2014). 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 χ_1 and χ_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

REFERENCES

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/

Abbott, B. P., Abbott, R., Abbott, T., et al. 2020, Living reviews in relativity, 23, 1

Ajith, P., Hannam, M., Husa, S., et al. 2011, Physical Review Letters, 106, 241101

Boyle, M., Hemberger, D., Iozzo, D. A., et al. 2019, Classical and Quantum Gravity, 36, 195006

Edwards, T. D. P., Wong, K. W. K., Lam, K. K. H., et al. 2023, RIPPLE: Differentiable and Hardware-Accelerated Waveforms for Gravitational Wave Data Analysis. https://github.com/tedwards2412/ripple

Hannam, M., Schmidt, P., Bohé, A., et al. 2014, Physical review letters, 113, 151101

Husa, S., Khan, S., Hannam, M., et al. 2016, Physical Review D, 93, 044006

Khan, S., Husa, S., Hannam, M., et al. 2016, Physical Review D, 93, 044007

Paszke, A., Gross, S., Massa, F., et al. 2019, in Advances in Neural Information Processing Systems 32, ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Curran Associates, Inc.), 8024–8035. http://arxiv.org/abs/1912.01703

Pratten, G., Husa, S., Garcia-Quiros, C., et al. 2020, Physical Review D, 102, 064001

Pürrer, M., & Haster, C.-J. 2020, Physical Review Research, 2, 023151

Santamaria, L., Ohme, F., Ajith, P., et al. 2010, Physical Review D, 82, 064016

Taracchini, A., Buonanno, A., Pan, Y., et al. 2014, Physical Review D, 89, 061502

Varma, V., Field, S. E., Scheel, M. A., et al. 2019, Physical Review D, 99, 064045