Deep Residual Error and Bag-of-Tricks Learning for Gravitational Wave Surrogate Modeling

Styliani-Christina Fragkouli[1] Paraskevi Nousi[1] Nikolaos Passalis[1] Panagiotis Iosif[2] Nikolaos Stergioulas[2] Anastasios Tefas[1]

[1]Department of Informatics
Aristotle University of Thessaloniki

[2]Department of Physics
Aristotle University of Thessaloniki

Table of Contents

Creating a Surrogate Model

From the work of (Field et al. 2014) the next steps were followed in order to build a surrogate model:

1. A training of N waveforms was created, using `SEOBNRv4` (Non-Precessing, Spinning Black Hole Binary with aligned spins) with PyCBC (Nitz et al. 2021),
   - $\{h_i(t;\boldsymbol{\lambda}_i)\}_{i=1}^{N}$ where $\boldsymbol{\lambda} = (q, x_1, x_2)$, $q = \frac{m1}{m2}$ is the mass ratio $1 \leq q \leq 8$, $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$ are the spins.
2. Using routines from ROMpy (Galley 2020) :
   - Greedy algorithm selects $m < N$ waveforms (and their $\boldsymbol{\lambda}$ values), which create the reduced basis $\{e_i\}_{i=1}^{m}$ for a given tolerance.
   - the Empirical Interpolation (EIM) algorithm finds informative time points (empirical nodes $\{T_i\}_{i=1}^{m}$) that can be used to reconstruct the whole waveform for arbitrary $\boldsymbol{\lambda}$.

Creating a Surrogate Model

3. A training set of 200k samples and validation and test set (30k samples each) are generated in the same $\boldsymbol{\lambda}$ interval and the corresponding coefficients are extracted.
4. To deal with the interpolation in 3 dimensions, Artificial Neural Networks (ANNs) are implemented to map the 3D input $\boldsymbol{\lambda}$ to the coefficients from the empirical nodes T_j (Khan and Green 2021).

Implementing Neural Networks

Following (Khan and Green 2021) ,two separate networks are used; one for the amplitude and one for the phase of the waveforms

$$h(t, \lambda) = A(t, \lambda)e^{-i\phi(t,\boldsymbol{\lambda})}$$

Ground Truth

Greedy Tolerance	n (amplitude)	n (phase)	mismatch $\mathcal{M}$ (max)	mismatch $\mathcal{M}$ (median)	mismatch $\mathcal{M}$ (95^{th} percentile)
10^{-6}	8	4	8.44×10^{-3}	5.47×10^{-4}	1.81×10^{-3}
10^{-8}	13	4	8.44×10^{-3}	5.45×10^{-4}	1.80×10^{-3}
10^{-10}	18	8	4.95×10^{-4}	1.30×10^{-5}	8.22×10^{-5}
10^{-12}	41	12	2.07×10^{-6}	7.45×10^{-8}	2.83×10^{-7}
10^{-14}	84	32	1.34×10^{-8}	5.64×10^{-10}	3.95×10^{-9}
10^{-16}	93	48	6.60×10^{-9}	4.59×10^{-10}	3.02×10^{-9}

Table 1: Number of bases (n) for the amplitude and phase and mismatch $\mathcal{M}$ (shown as maximum, median and 95^{th} percentile values) between waveforms reconstructed via EIM and original waveforms for the validation set. In blue color are the chosen bases to compare our work (Ground Truth)

Baseline Network MSE and Mismatch

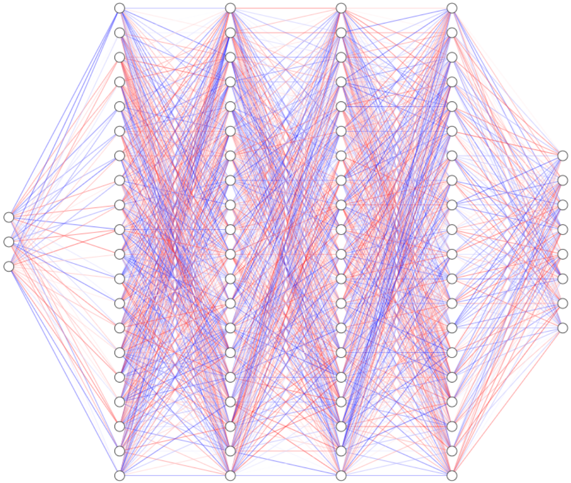
	MSE for predictions of training (average of 5 runs)	MSE for predictions of validation (average of 5 runs)
Amplitude	$1.79 \times 10^{-7} \pm 3.52 \times 10^{-10}$	$1.84 \times 10^{-7} \pm 3.29 \times 10^{-10}$
Phase	$1.05 \times 10^{-8} \pm 2.18 \times 10^{-10}$	$1.06 \times 10^{-8} \pm 2.11 \times 10^{-10}$

Table 2: Training baseline network MSE between predictions and ground truth.

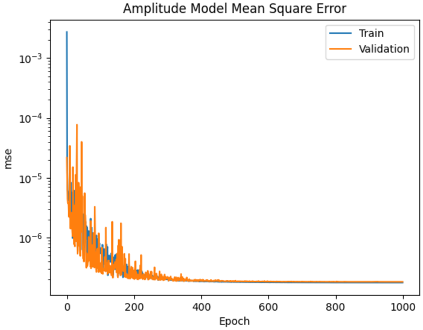
	Mismatch $\mathcal{M}$
Min	$2.70 \times 10^{-6} \pm 3.43 \times 10^{-7}$
Max	$7.73 \times 10^{-3} \pm 5.38 \times 10^{-4}$
95^{th}	$2.95 \times 10^{-4} \pm 6.61 \times 10^{-6}$
Median	$8.39 \times 10^{-5} \pm 1.91 \times 10^{-6}$

Table 3: Waveforms Mismatch for validation set.

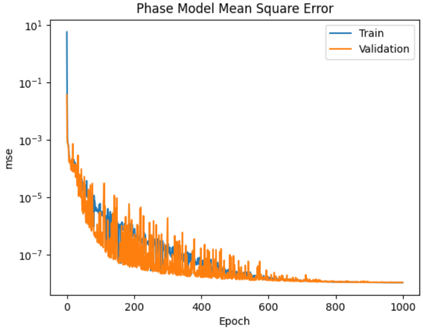
Model Architecture



ANNs Learning Curves

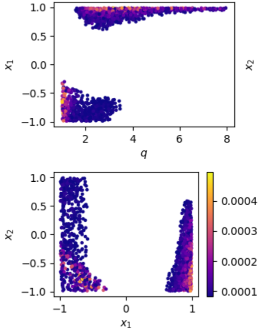


(a) MSE curve for train and validation set of the Amplitude network.

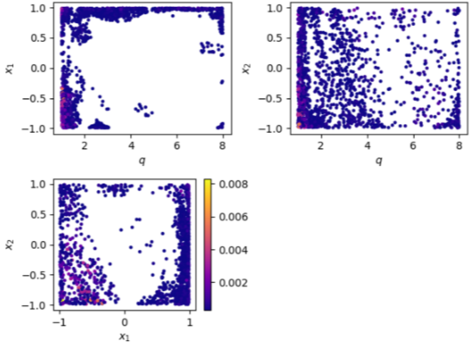


(b) MSE curve for train and validation set of the Phase network.

Mismatch



(a) $\mathcal{M}$ above 95^{th} percentile between waveforms reconstructed via EIM and original waveforms (Ground Truth).

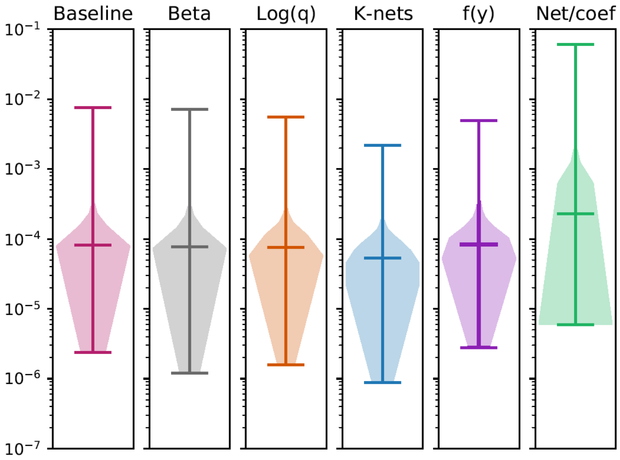


(b) $\mathcal{M}$ above 95^{th} percentile between waveforms reconstructed via ANNs predictions and original waveforms.

List of Tricks

1. Input with 4 parameters $\boldsymbol{x} = (q, x_1, x_2, \beta)$
2. Input Augmentation with $log(q)$ and -$log(q)$
3. Output Augmentation with $f(y)$
4. K-networks (input feature-based dissection)
5. Network per coefficient (output-based dissection)

Tricks Violin Mismatch Plots



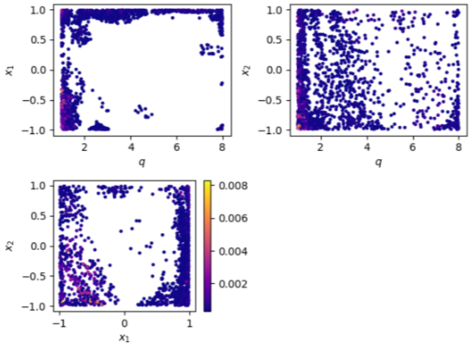
Residual Errors Network

A second residual neural network for the training network errors

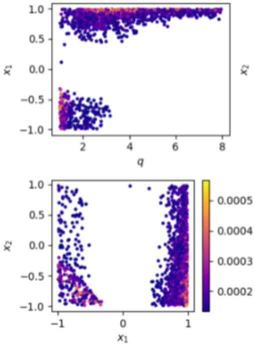
$$\tilde{\boldsymbol{y}} = h(x) + f(x)$$

- $\tilde{\boldsymbol{y}}$: Final ANNs predictions
- $h(x) = \hat{\boldsymbol{y}}$: Predictions from baseline ANN
- $f(x) = \hat{\boldsymbol{e}}$: Predictions from residual error ANN

Baseline Network Mismatch

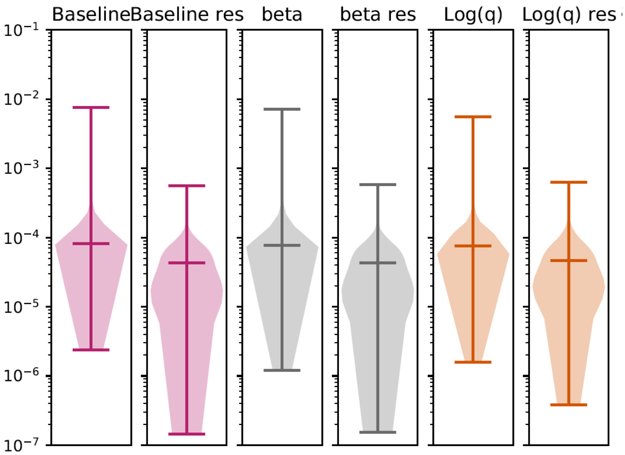


(a) $\mathcal{M}$ above 95^{th} percentile between waveforms reconstructed via ANNs predictions and original waveforms without residual network.

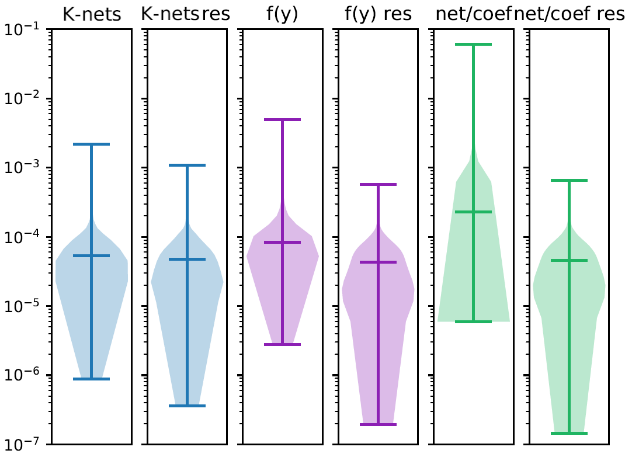


(b) $\mathcal{M}$ above 95^{th} percentile between waveforms reconstructed via ANNs predictions and original waveforms with residual network.

Violin Mismatch Plots



Violin Mismatch Plots



Conclusions

1. Physics and learning induced ideas can improve final results
2. When including a second ANN trained on residual errors, improves final mismatch by more than an order of magnitude.

Thank you for your time!

References I

Cutler, Curt and Eanna E. Flanagan (Mar. 1994). “Gravitational waves from merging compact binaries: How accurately can one extract the binary’s parameters from the inspiral waveform?” In: *Physical Review D* 49.6, pp. 2658–2697. ISSN: 0556-2821. DOI: 10.1103/physrevd.49.2658. URL: http://dx.doi.org/10.1103/PhysRevD.49.2658.

Field, Scott E et al. (2014). “Fast prediction and evaluation of gravitational waveforms using surrogate models”. In: *Physical Review X* 4.3, p. 031006.

Galley, Chad R. (2020). *RomPy package*. https://bitbucket.org/chadgalley/rompy/.

Khan, Sebastian and Rhys Green (2021). “Gravitational-wave surrogate models powered by artificial neural networks”. In: *Physical Review D* 103.6, p. 064015.

Nitz, Alex et al. (May 2021). *gwastro/pycbc: PyCBC Release 1.18.1*. Version v1.18.1. DOI: 10.5281/zenodo.4849433. URL: https://doi.org/10.5281/zenodo.4849433.

β as extra input

- Includes the dependence of the phase inspiral waveforms on spin combinations of q, x_1 *and* x_2 with same β lead to same waveforms (Cutler and Flanagan 1994)

$$\beta = \left(\frac{113}{12} + \frac{25}{4q}\right) \frac{q^2}{(1+q)^2} x_1 + \left(\frac{113}{12} + \frac{25q}{4}\right) \frac{1}{(1+q)^2} x_2$$

- The new input $\boldsymbol{x} = (q, x_1, x_2, \beta)$ has 4 parameters
- number of training samples and output nodes are similar to the baseline model

Implementation of $log(q)$ and -$log(q)$

- inserting a new branch leading to 400k training samples
- repeated output: 36 output nodes for the amplitude
- repeated output: 18 output node for the phase
- calculation of the mean of the prediction from the 2 branches

Implementation of K-networks

- input was divided into K=2 (for balanced training samples) according to the value of q
- best results from overlapping range of q (1, 4.2) and (3.8, 8)
- two separate networks were trained in both ranges
- single residual network

Implementation of $f(\boldsymbol{y})$ output

- insert a new branch corresponding to a function $f(\boldsymbol{y})$
- 200k training samples
- amplitude 36 output nodes: 18 from $\boldsymbol{y}$ space and 18 from $f(\boldsymbol{y})$ space
- phase 16 output nodes: 8 from $\boldsymbol{y}$ space and 8 from $f(\boldsymbol{y})$ space
- calculation of mean from the two branches