GENERAL RELATIVISTIC GRAVITATIONAL WAVES FROM GR EFFECTIVE POTENTIAL

by

Argyn Yergaliyev

A Thesis Submitted to the Faculty of the

DEPARTMENT OF PHYSICS

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the School of Sciences and Humanities

NAZARBAYEV UNIVERSITY

2025

NAZARBAYEV UNIVERSITY, SCHOOL OF SCIENCES AND HUMANITIES

As members of the thesis committee, we certify that we have read the thesis prepared by Argyn Yergaliyev entitled

GENERAL RELATIVISTIC GRAVITATIONAL WAVES FROM GR EFFECTIVE POTENTIAL

and recommend that it be accepted as fulfilling the thesis requirement for the degree of Master of Science.

_____ Date: April 25, 2025

_____ *Date:April*25,2025

I hereby certify that I have read this thesis prepared under my direction and recommend that it be accepted as fulfilling the thesis requirement.

_____ Date: April 25, 2025
Thesis Director: Professor Ernazar Abdikamalov

TABLE OF CONTENTS

ABSTRACT

Core-collapse supernovae (CCSNe) are among the most powerful cosmic explosions that produce gravitational waves (GWs). The information about the explosion mechanism and stellar interior is present within the GWs. However, complete general relativity (GR) simulations of CCSNe are computationally demanding, so approximate techniques like the General Relativistic Effective Potential (GREP) approximation are required. In this thesis, machine learning and regression are formulated to act as a mapping function of efficient GREP simulations and accurate GR waveforms. We compare ridge regression and neural networks for projecting GREP-produced waveforms to their GR equivalents over four equations of state (SFHo, LS220, HSDD2, GShenFSU2.1). Our findings are that the most effective neural networks are symmetric in architecture (e.g. 80-40-40-80), achieving a waveform accuracy of RRMSE < 0.07 compared to full GR simulations in efficient computation. The trained models preserve key features like bounce timing and ringdown frequencies, enabling efficient generation of accurate waveforms for gravitational wave astronomy.

INTRODUCTION

Core-collapse supernovae

There are many ways to collect astronomical data with modern instruments. This includes the emerging field of gravitational wave astronomy. The detection of gravitational waves (GWs) by the LIGO-Virgo collaboration in 2015 opened a new window into the most violent phenomena in the universe. This metric perturbations, first predicted by Einstein's theory of general relativity (GR) in 1916 [1], carry useful information about the phenomena that generated them. These phenomena include merging black holes, merging of neutron stars, and the weaker waves are generated by catastrophic deaths of massive stars in core-collapse supernovae (CCSNe). In this thesis, I focus on the last source.

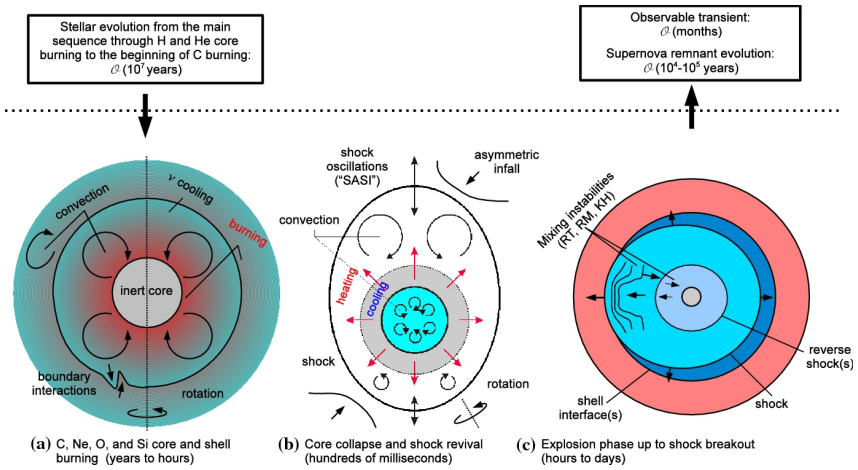


Figure 1: Schematics for CCSNs [2]

When massive stars with initial masses greater than about 8 solar masses exhaust their nuclear fuel, their iron cores collapse under gravity at a considerable fraction of light's speed. This collapse halts abruptly when nuclear densities are reached, producing a shock wave that propagates outward while the inner core settles into a proto-neutron star (PNS) [3]. This phenomenon is particularly interesting because CCSNe are primary nucleosynthesis factories, producing and dispersing most of the universe's heavy elements after iron into the interstellar medium. Also, CCSNe are natural laboratories for studying matter under extreme conditions like high densities, only behind the stellar remnants. The dynamics of this process, like core bounce, hydrodynamical inhomogenities, stellar oscillations, and rotational instabilities, produce a GW signal that encodes crucial information about the explosion mechanism [4]. The observations can help us to put further constraints on our models of GR and equations of state for high-density matter. In turn,

with better modeling of the processes, we can more efficiently process and interpret the obtained data.

Gravitational waves: derivation

Metric Decomposition

The equation for weak-field gravitational waves can be obtained by perturbation of the metric [5]. The spacetime metric $g_{\mu\nu}$ is perturbed around the Minkowski metric:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad |h_{\mu\nu}| \ll 1 \tag{1}$$

where:

- $\eta_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$ (flat spacetime)
- $h_{\mu\nu}$ = small perturbation

Einstein's Equations in Linearized Form

The Ricci tensor to first order in $h_{\mu\nu}$:

$$R_{\mu\nu} \approx \frac{1}{2}\left(\partial_\sigma\partial_\nu h^\sigma_\mu + \partial_\sigma\partial_\mu h^\sigma_\nu - \partial_\mu\partial_\nu h - \Box h_{\mu\nu}\right) \tag{2}$$

where:

- $h = \eta^{\mu\nu}h_{\mu\nu}$ (trace)
- $\Box = -\frac{1}{c^2}\partial_t^2 + \nabla^2$

The linearized Einstein tensor:

$$G_{\mu\nu} \approx \frac{1}{2}\left(\partial_\sigma\partial_\nu h^\sigma_\mu + \partial_\sigma\partial_\mu h^\sigma_\nu - \partial_\mu\partial_\nu h - \Box h_{\mu\nu} - \eta_{\mu\nu}\partial_\alpha\partial_\beta h^{\alpha\beta} + \eta_{\mu\nu}\Box h\right) \tag{3}$$

Lorenz Gauge Condition

Impose the Lorenz gauge:

$$\partial^\mu \bar{h}_{\mu\nu} = 0, \quad \text{where} \quad \bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h \tag{4}$$

This simplifies the field equations to:

$$\Box \bar{h}_{\mu\nu} = -\frac{16\pi G}{c^4}T_{\mu\nu} \tag{5}$$

Gravitational wave astronomy

The field of gravitational wave astronomy is one of the most amazing feats in recent astrophysics, having evolved from theoretical anticipation to observational verification within a century. However, the incredibly low energy of these waves made them almost undetectable for decades, confining them to mathematical curiosity alone, a phenomenon that can be observed.

The initial serious attempts at gravitational wave detection were made in the 1960s through Joseph Weber's experiments with resonant bars [6] [7]. Although his detection claims were later proven false, they initiated the key theoretical work of Kip Thorne and others in establishing a framework for astrophysical source possibilities. Around this time, core-collapse supernovae had been recognized as a promising astrophysical candidate for gravitational wave emission because their cataclysmic, asymmetric stellar death processes would produce strong enough gravitational wave radiation [8].

Later, with advances in computers and software, researchers began simulating gravitational wave emission from various astrophysical processes, including the merging of compact objects and supernovae[9]. This era also witnessed the concept and initial construction of LIGO (Laser Interferometer Gravitational-Wave Observatory), which would eventually revolutionize the field [**?**].

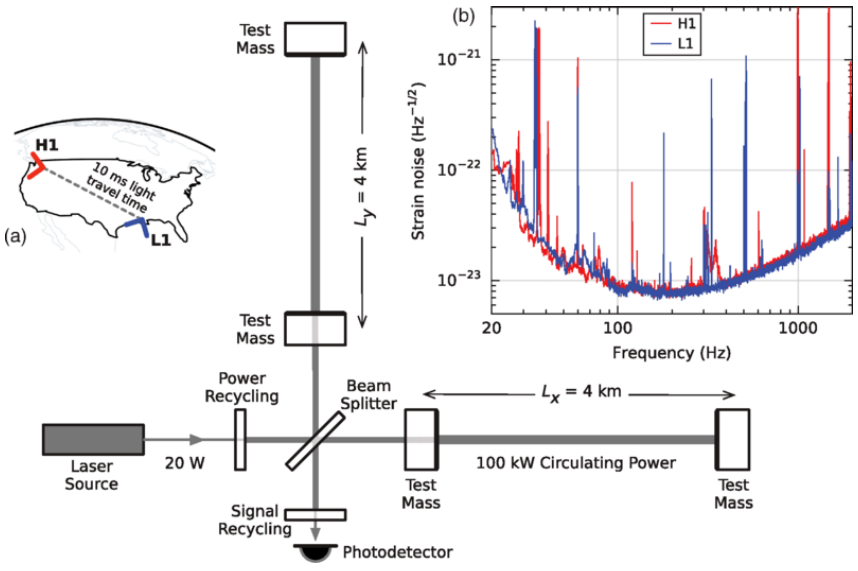


Figure 2: Simplified diagram of LIGO [10]

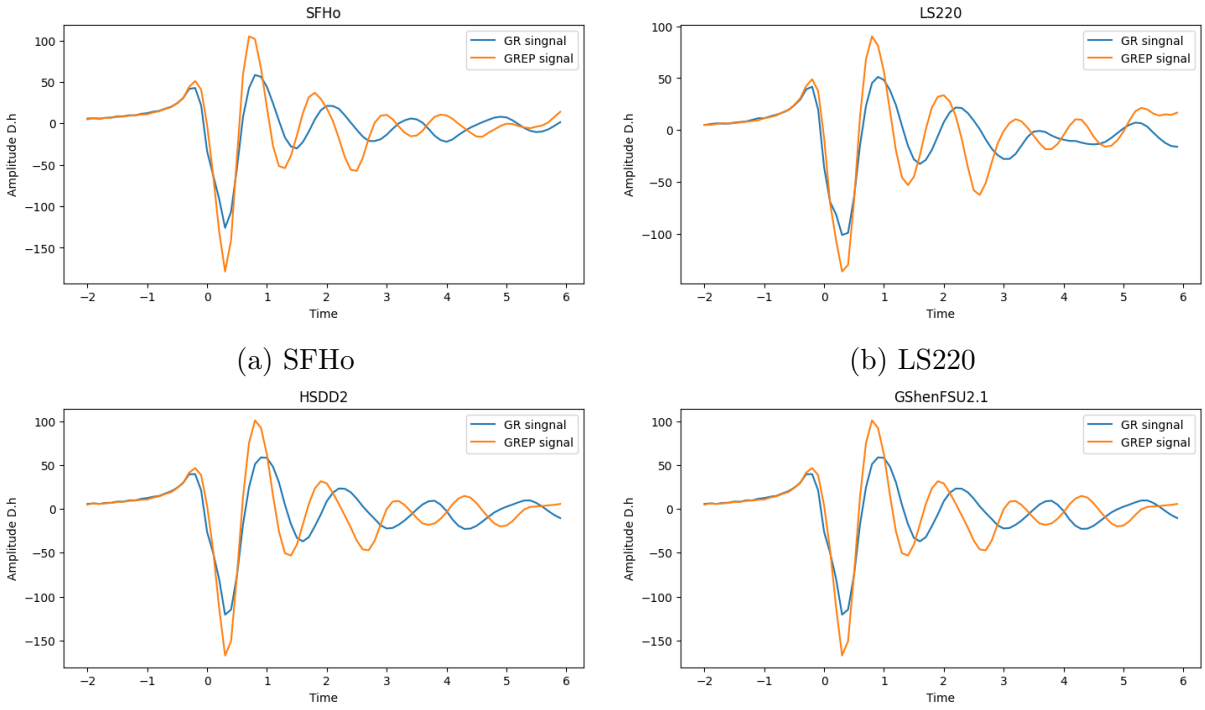
Early LIGO runs between 2002 and 2010 were unprecedentedly sensitive but did not achieve a single detection [11]. The Advanced LIGO, upgraded and running from 2015 to the present, eventually attained the sensitivity required for breakthrough findings. The first gravitational wave detection was a binary black hole merger (GW150914) [10] in

September 2015. This was followed in 2017 by the historic observation of GW170817, a binary neutron star merger [12].

Due to their relative weakness, GWs from CCSNs have not yet been observed. The next-generation observatories like the Einstein Telescope [13] should increase the sensitivity of our observations and help to finally detect GWs from CCSNs.

Limitations of GR simulations

Modeling these GW signals requires solving the full set of Einstein's field equations coupled to the equations of magnetohydrodynamics and neutrino transport - a significant computational challenge that constitutes the field of numerical relativity. Full 3D numerical simulations of CCSNe are currently still too costly, which motivates researchers to create approximate schemes that retain essential physics while dramatically reducing computational costs. Current condition in relativistic hydrodynamics, and complex neutrino transport calculations that frequently dictate the run time. Such constraints make the use of full GR simulations (like [14]) undesirable in applications requiring fast production of waveforms. One of the approaches to deal with it is the General Relativistic Effective Potential (GREP) [15] framework, which approximates the GR effects in the Newtonian framework. This drastically reduces the computational time; however, in doing so, this model loses a lot of features of GR, like time dilation and length contraction. From Fig. 3, we can see that the most obvious difference is the frequency of the signals, their amplitudes, and ringdown.



(a) SFHo (b) LS220

(c) HSDD2 (d) GShenFSU2.1

Figure 3: GR and GREP waveforms

Different equations of state

Table 1: Core Equations of State (EOS) in Supernova Simulations

EOS Name	**Short Description**
SFHo	Relativistic mean field model constrained by neutron star observations, predicts more compact neutron stars and high-frequency gravitational wave signals.
LS220	Benchmark nuclear equation of state with intermediate stiffness (incompressibility K=220 MeV), widely used for supernova simulations.
HSDD2	Hybrid model featuring a phase transition from hadronic to quark matter at high densities).
GShenFSU2.1	Stiff relativistic mean field variant predicting larger neutron stars with lower-frequency gravitational wave emission.

As discussed above CCSN models are dependent on the equation of state (EOS) of dense nuclear matter. Current theoretical models of the EOS span a range of possibilities, from relatively soft models like SFHo [16] that predict smaller neutron star radii to stiffer

variants like GShenFSU2.1 [17] that allow for larger compact stars. The SFHo EOS, put forward by Steiner, Hempel, and Fischer in 2013, is constrained by neutron star astrophysical observations as well as by nuclear physics experiments, namely neutron star measured radii and masses. In comparison to such models, the LS220 [18] EOS is a more traditional one founded on nuclear matter theory with incompressibility 220 MeV, whereas hybrid models like HSDD2 [19] incorporate phase transitions to exotic phases like quark matter in the early post-bounce phase. These EOS differences produce significant enough effects that can be observed in the GW signal.

Aims of the work

This thesis examines a machine learning approach for overcoming the computational expense gap of schemes like GREP and that of full numerical relativity codes. We employ traditional ridge regression tools and deep neural networks in order to learn the systematic difference between GREP waveforms and general relativistic waveforms. The ridge regression technique provides a simple linear mapping that is straightforward to interpret and can successfully remove phase and amplitude errors, and the neural networks will identify more complex, nonlinear patterns in the waveform data. We try various neural network architectures, including symmetric medium-width ones like the 80-40-40-80 architecture and more complex hourglass networks, to determine the optimal balance between accuracy and computational expense for this task. Our method is to train these models on a carefully collected set of paired GREP and GR waveforms from different equations of state and progenitor models and then challenge them robustly on holdout data that hadn't been used in training.

Possible applications

A successful implementation of such machine learning-based and regression-based waveform mapping methods could help to bypass the full GR simulations. With sufficient data, these modes should be able to generalize them. With high-quality training, these models can be used in place of only the GREP waveforms. This could lead to significant improvements in the accuracy of other models that require a lot of data. One example of this is providing of iron core through GWs [20] [20] [21], where they classify EoS using ML. However, training on GREP waveforms and testing on GR waveforms resulted in low accuracy. They employed time-normalization to match the peak frequencies of GR and GREP signals. Although their method increased the accuracy, the final result was still below 70% [21]. Therefore, other mapping techniques like those in this thesis should be investigated for a possible improvement.

METHODS

Environment

The coding was done on Google Colab, a cloud-based notebook designed to run Python code and write LaTeX-style text. All runs were done on CPU, no GPU acceleration was used. Libraries necessary for the tasks are: Pandas, NumPy, PyTorch, and scikit-learn. Pandas is a necessary library to work with datasets, specifically to extract desired data from .csv files. NumPy is a library for Python that adds arrays and functions on arrays. PyTorch is a library for machine learning that adds support for neural networks. Lastly, scikit-learn is another library that adds functions for regression and other operations similar to NumPy.

Waveforms

Waveforms are obtained through numerical simulations using the code CoCoNuT [14][22]. As described earlier, there are two models for gravitational waves considered, the first one is GR, and the second is Newtonian hydrodynamics with the GR effective potential. Each GW has its own set of parameters. For the task at hand, we focus on three of them. T/W - quotient of rotational kinetic energy to potential binding energy, A - measure of differential rotation, and Ω cylindrical radius [23] [24].

$$\Omega(\varpi) = \Omega_0 \left[1 + \left(\frac{\varpi}{A}\right)^2\right]^{-1}$$

First, we limit ourselves to values of $T/W < 0.05$. The reason is shown in the figure below. When we try to match GWs by the closest T/W parameter, we can see that the variance of their peak frequencies is very large. This leads to large errors in prediction by our models, indeed, larger values of T/W diverge into several directions. This is possibly related to parameter A. However, values of $T/W < 0.05$ are concentrated in the lower left corner of the figure and have low variance, which is suitable for training.

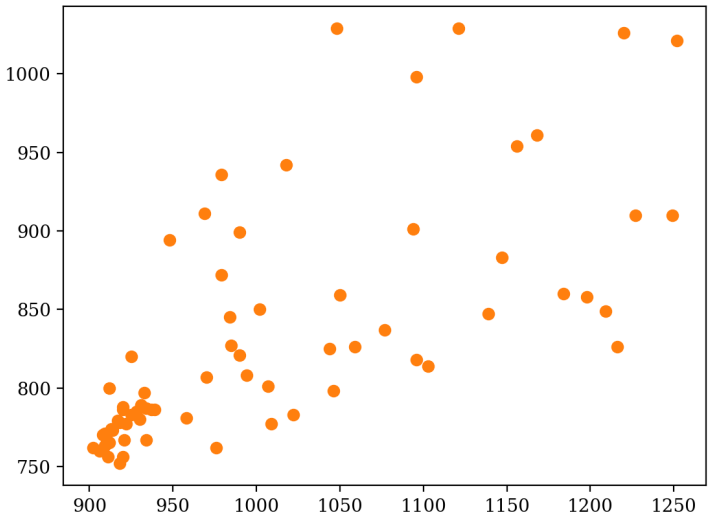


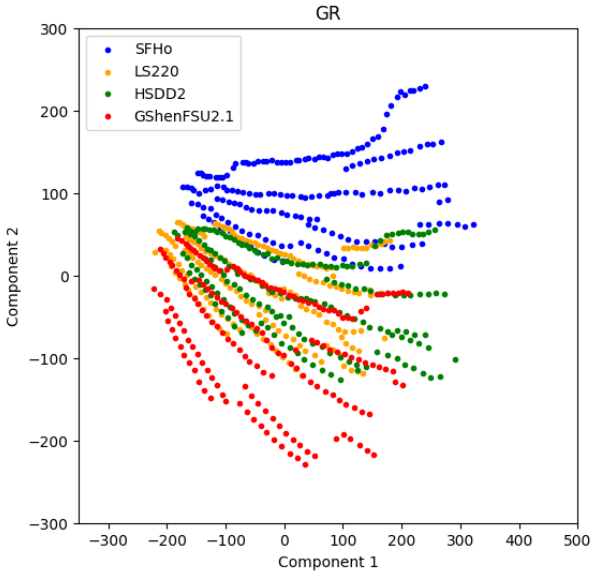
Figure 4: Peak frequencies of matched GR (y-axis) and GREP (x-axis) signals

Data separation and sorting

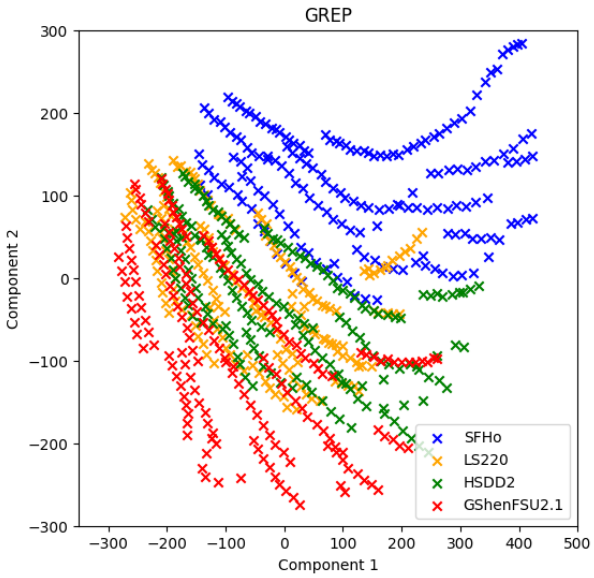
With Pandas, we can easily separate the data as we need. The first step is to divide waveforms by the equation of state and the potential. So, we end up with 8 arrays of 185 waveforms each (except for GShenFSU2.1, which has 177 waveforms). On top of that, we further need to divide the arrays by the A and Ω parameters. A ranges from 1 to 5, and Ω from $2.0 - 6.0$. Each array is separated into 5 based on parameter A. To map waveforms, we need to create appropriate datapoints, i.e,. pairs of GREP and GR waveforms. Pairs consist of waveforms with the same A and Ω.

PCA

Principal Component Analysis reduces the dimensions of our data and helps to identify patterns in it to increase the accuracy of our models. The figures below (Fig. 5) are scatter diagrams of the first two principal components.



(a) GR



(b) GREP

Figure 5: PCA GR and GREP

As we can see, the data points are separated into five curves, each corresponding to one of the values of parameter A. PCA now shows clearly that the variance of the peak frequencies discussed above can also be attributed to parameter A. $A = 2$ and $A = 4$ overlap significantly with other values, which results in higher error in models. Therefore, they were excluded from the first group of regression models.

Models

Regression

The first way to map one data point to another and to find the relationship between them is regression, including linear, quadratic, and polynomial. After several tests, only linear regression was considered. For our 80-dimensional "vectors", linear regression means finding the best-fit hyperplane, i.e. set of coefficients such that

$$\mathbf{GR}_{m\times 1} = W_{m\times m}\,\mathbf{GREP}_{m\times 1}$$

Predicts the output (GR) from the input variables (GREP). For multiple-dimensional variables, it is done through the matrix equation

$$\mathbf{Y}_{N\times 80} = W_{N\times 80+1}\,\mathbf{X}_{80+1\times N}$$

, where W is a matrix of coefficients, Y is a matrix of independent variables, and X is a matrix of dependent variables. X, Y have the shape of (N, 80), so they are a matrix of N 80-dimensional vectors. The bias (intercept) terms are included in the W matrix, with the addition of a row of ones to the X matrix.

However, during the analysis of the differences between GR and mapped GREP, it was clear that linear regression overfitted the data. The RRMSE (Relative Root-Mean-Square Error) for a training set was much smaller than the RRMSE for the test set. This happens because the matrix of coefficients of shape (80, 80) contains more parameters than the available data points. Even with the increase in the number of waveforms, the discrepancy between RRMSE for training sets and test sets remained. Therefore, it was necessary to perform regularization, specifically L2 regularization or Ridge regression.

Ridge regression

Ridge regression or Tikhonov regularization is a modification of the standard linear regression via ordinary least squares that addresses the multicollinearity and overfitting in regression models. It adds a penalty term in order to constrain the coefficients from blowing up [25].

$$J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

$$J_{\text{ridge}}(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}$$

$$\frac{\partial J_{\text{ridge}}}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = 0$$

$$(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$$

$$\Rightarrow \boxed{\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}}$$

, where $\hat{\beta}$ is a vector of coefficients that would make up a matrix. Here, we are free to choose the parameter λ. This parameter determines the strength of the regularization, with $\lambda = 0$ corresponding to ordinary least squares and larger values shrinking the coefficients. The $\lambda = 10$ was chosen for these tests. Too large values of λ could result in poorer generalization, as coefficients increase very slowly. The Ridge regression is handled with functions from the scikit-learn library.

Neural networks

Artificial neural networks are conceptually digital versions of the human brain's structure, i.e., they consist of interconnected nodes (neurons). Nodes are organized into layers, where data from previous layers passes into, transforms, and passes to the next layer. In simpler terms, it is mapping. They consist of an input layer, hidden layers, and an output layer. In the simplest forms, each node receives a weighted sum of outputs from the nodes of the previous layer, and inputs it into a function (activation function), then passes it on to other neurons in the next layer.

Feedforward neural networks are the simplest types of neural networks. The name "Feedforward" comes from the direction of how information flows. It flows in one direction, from its input layer to its output layer, without loops or other movement in its hidden layers. So, the description earlier was for FNNs.

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(\boldsymbol{x}) \tag{6}$$

where:

- $\boldsymbol{x} \in \mathbb{R}^{n_0}$ is the input vector
- $\boldsymbol{\theta} = \{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\}_{l=1}^{L}$ are the learnable parameters
- $\sigma^{(l)}$ is an activation function
- Each layer $f^{(l)}$ computes:

$$f^{(l)}(\boldsymbol{z}) = \sigma^{(l)}\left(\boldsymbol{W}^{(l)}\boldsymbol{z} + \boldsymbol{b}^{(l)}\right) \tag{7}$$

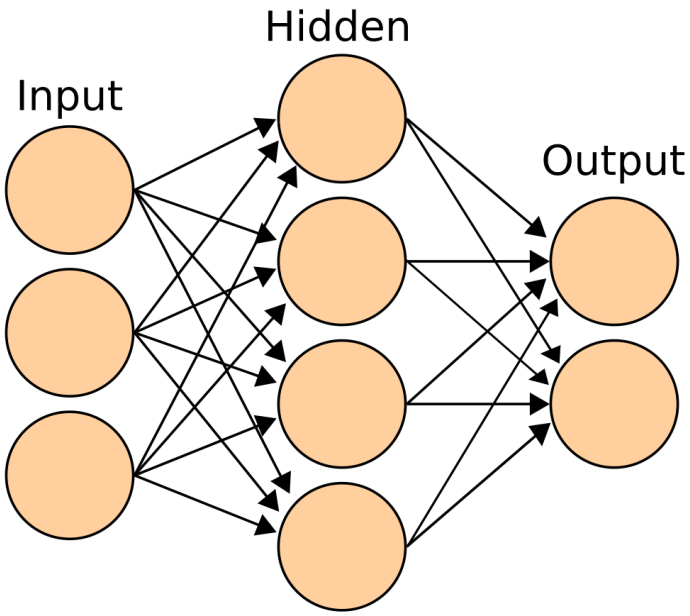


Figure 6: Structure of FNNs

Training

Data selection

For each training, the data is separated into a training set and a test set in a given ratio. Selection was done via random sampling. Pairs of waveforms have the same indices in arrays, therefore, sampling is done on the array of the indices and used on the arrays.

```
def select(x, y, per):
  num, dim = x.shape
  num_elements = int(num * per)
  indices = np.random.choice(num, num_elements, replace=False)

  x_sel = x[indices, :]
  y_sel = y[indices, :]

  rev_ind = []
  for i in range(x[:, 0].size):
    check = i in indices

    if check == False:
      rev_ind.append(i)
    else:
      continue

  return x_sel, y_sel, np.sort(indices), np.array(rev_ind)
```

Figure 7: Function for random sampling

All indices are preserved for later calculations of averages and standard deviations of

RRMSEs.

Approaches

First, we perform ridge regression for each EoS. As discussed above, for better results, we removed waveforms corresponding to $A = 2$, and 4. This will remove some overlapping waveforms.

Second, we train on waveforms that correspond to the four out of five values of parameter A. And test them on the waveforms of the fifth value. This is necessary to see if Ridge regression models can extrapolate independently of the value of A. For this part, all of the waveforms corresponding to the remaining value of A were also used as a test set. This is necessary for later comparison.

NN architectures

To evaluate the mapping capabilities of FNNs, we designed and tested multiple architectures with varying depths and widths. Five different architectures were considered for the task. The choice for the number of layers and the number of nodes were made based on the amount of data. Each data point contains 80 features ("coordinates"), and there are 185 (177 for GShenFSU2.1) waveforms for each EoS. The number of parameters had to be smaller or at least around the number of variables.

```
class DynamicFeedForwardNN(nn.Module):
    def __init__(self, layers, negative_slope=0.01):
        super(DynamicFeedForwardNN, self).__init__()
        self.model = self._build_model(layers, negative_slope)

    def _build_model(self, layers, negative_slope):
        modules = []
        for i in range(len(layers) - 1):
            modules.append(nn.Linear(layers[i], layers[i+1]))
            if i < len(layers) - 2:
                modules.append(nn.LeakyReLU(negative_slope=negative_slope))
        return nn.Sequential(*modules)

    def forward(self, x):
        return self.model(x)
```

Figure 8: Simple NN with variable number of nodes

NNs that were used were deep neural networks, i.e., with multiple hidden layers. For the tests, 2, 3, and 4 hidden layers with various numbers of nodes were used. For the activation function leaky rectified linear unit was used. Mean Squared Error (MSE) was used as a loss function, which is the usual choice for regression tasks. The optimizer of choice was Adam, which is more adaptive than the standard gradient descent. To prevent overfitting of the NNs, early stopping was implemented, i.e., if there is no improvement for a number of epochs (patience), then the NN will stop its training.

Table 2: Neural Network Architectures

Name	Structure	Parameters	Ratio of number of nodes
Deep-Medium	80-20-20-20-80	6,760	4:1
Deep-Narrow	80-10-10-10-80	3,370	8:1
Shallow-Wide	80-40-40-80	9,680	2:1
Shallow-Narrow	80-10-10-80	2,020	8:1
Hourglass	80-40-20-20-40-80	11,400	4:1 $\rightarrow$ 2:1

Again the data set was divided into training and testing set. Among the training set, 80 percent was used for training, the other 20 percent was used for validation. It helps to assess how well the model generalizes to unseen data and to prevent overfitting.

Error analysis

To measure the accuracy and precision of our models, we use the relative root mean square error (RRMSE). RRMSE is the dimensionless measure of the error, where RMSE is divided by the square root of the sum of squares of observed values [26].

$$\text{RRMSE} = \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i)^2}} = \frac{\text{RMSE}}{\sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{8}$$

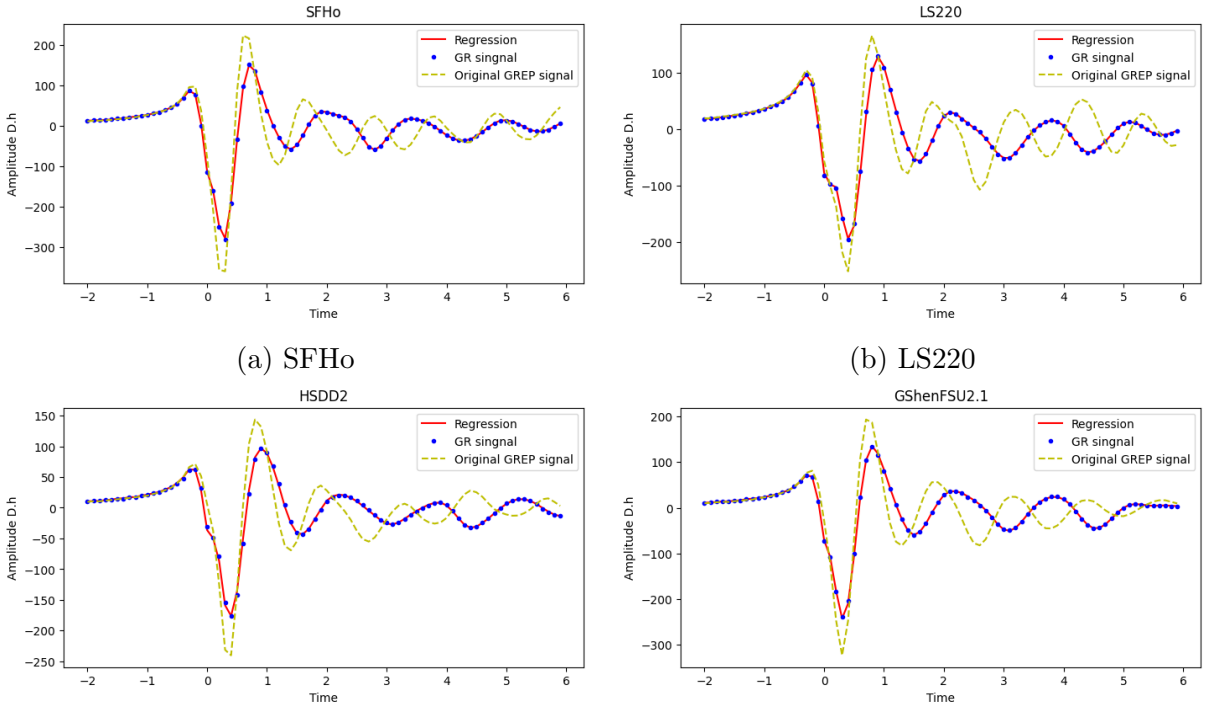
While RMSE gives error in units of the variables, it is difficult to compare directly, due to the changing number of data points.

For each modeling, 100 runs were performed, where different ratios of the training and testing sets were used. The ratio of the training set ranges from 5% to 95%. The RRMSE between GR and the modeled signal was calculated for the test set, as well as the total set and the training set. The latter two showed identical results across the EoS.

RESULTS

Tests on each EoS

First, to qualitatively assess the performance of Ridge regression, we compare the time-domain representations of original GREP waveforms, ridge-regression-mapped waveforms, and the GR waveforms for representative cases. As described in the previous section, only $A = 1, 3$, and 5 were used.



(a) SFHo

(b) LS220

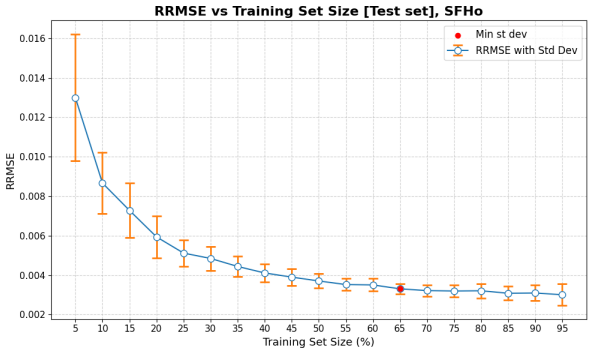
(c) HSDD2

(d) GShenFSU2.1

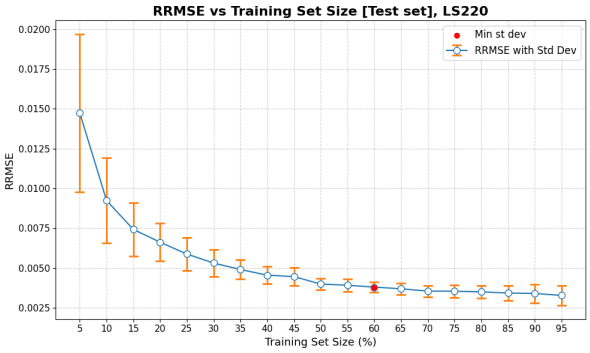
Figure 9: GR, GREP, and regression model waveforms

These are examples of an 80% training set ratio. It can be seen from Fig. 9. that the regression model performs well in mapping waveforms. The amplitude and the peak frequencies match the GR waveform. Qualitative evaluation shows that ridge regression successfully reproduces time-critical features like bounce timing and ringdown of GR waveforms.

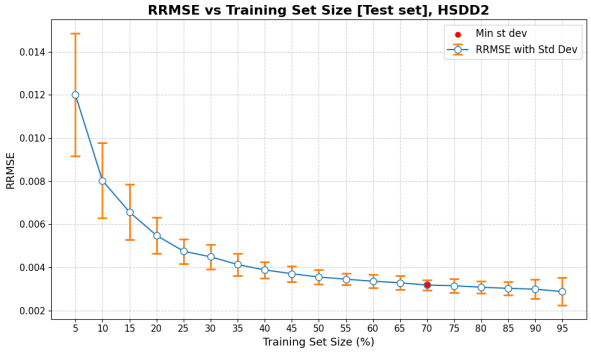
We evaluated the performance of Ridge regression in mapping GREP-generated waveforms to their GR counterparts across four equations of state (SFHo, LS220, HSDD2, GShenFSU2.1). The analysis quantified the RRMSE and its standard deviation on held-out test sets for varying fractions of the training data (from 5% to 95%, with a 5% increment).



(a) SFHo



(b) LS220



(c) HSDD2

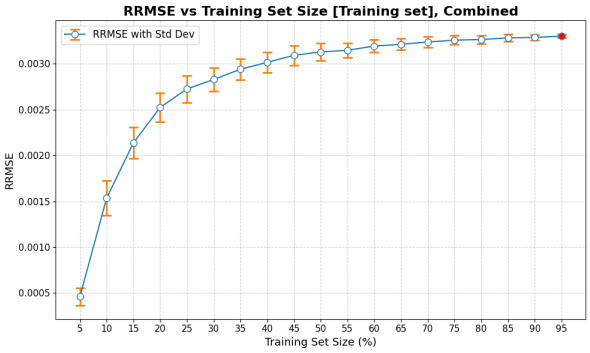
(d) GShenFSU2.1

Figure 10: RRMSE of testing sets

As expected, it can be seen from the figures above that the RRMSE decreases monotonically as the training set fraction increases. The consistency of the RRMSE values is highest for training sets of around $60 - 70\%$ of the total set. Gains in accuracy slowed beyond that ratio of training data, suggesting a trade-off between computational cost and marginal improvement.

The figures below show the waveform and RRMSE obtained by inputting all EoS, i.e. 80% of waveforms of all EoS were used. However, despite the quadruple increase in the number of datapoints, the RRMSE values showed the worst performance out of all. One of the sources of higher error is probably the overlapping features of waveforms across different EoS with the same set of parameters, which leads to the Ridge regression to average them out, dropping features of individual EoS.

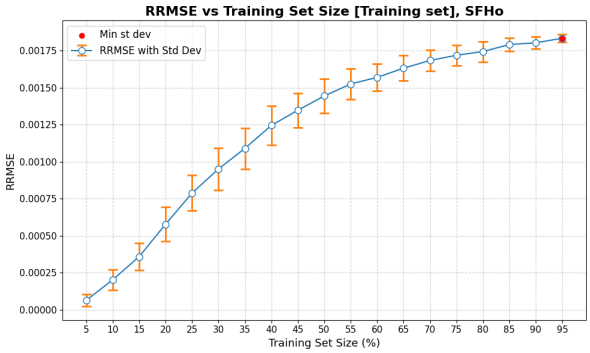
(a) Testing set



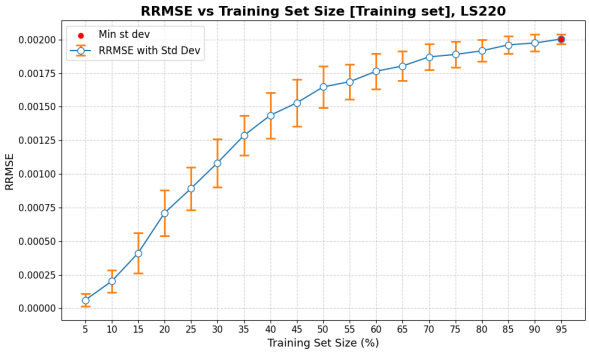
(b) Training set

Figure 11: RRMSE of combined waveforms

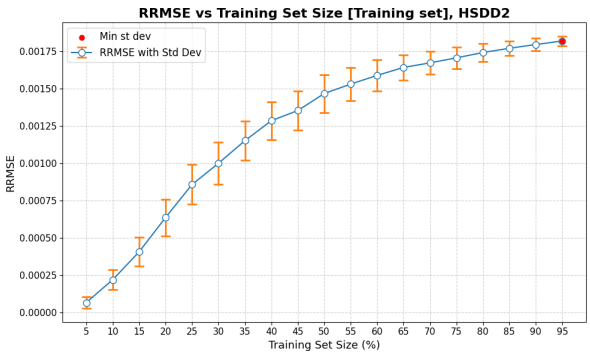
Now, let's look at the other graphs. The figure below shows the RRMSE of training sets. As we can see, it increases monotonically until a certain upper limit. The limit being the RRMSE of the training set. We can observe how the Model transitions from overfitting to generalization. The small values of RRMSE at the lower ratio of the training set show the overfitting of the model, where the number of variables is substantially smaller than the number of parameters of the regression model.



(a) SFHo



(b) LS220



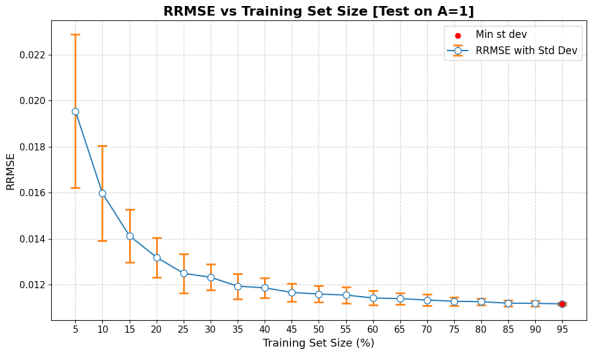
(c) HSDD2

(d) GShenFSU2.1

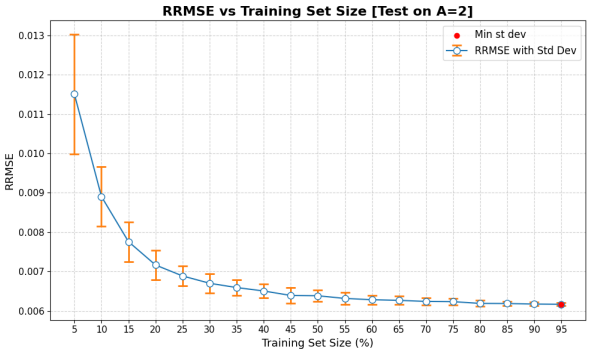
Figure 12: RRMSE of training sets

However, even with the approaching RRMSEs of training and test sets, there is still a difference, which is indicative of unresolved overfitting issues. The closer values in the higher ranges of the training set are the result of the number of variables increasing, therefore, the issue can be mitigated with a higher number of waveforms available.

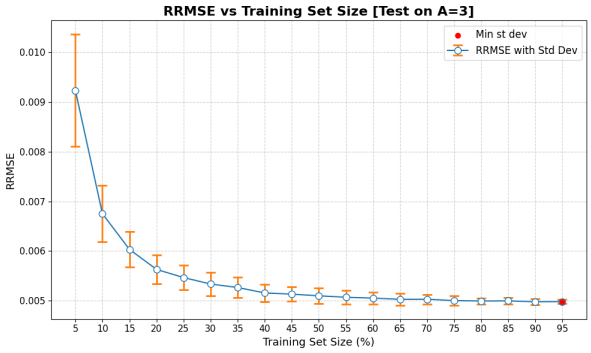
Tests on the excluded parameter



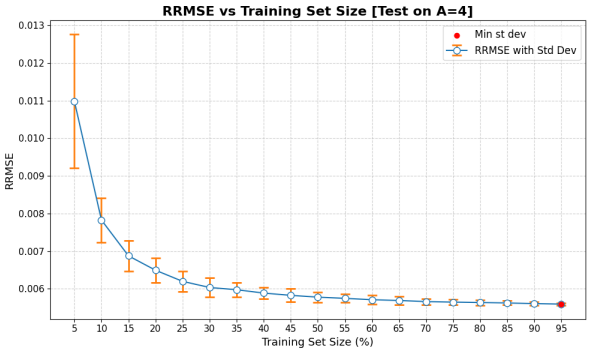
(a) Exclude A=1



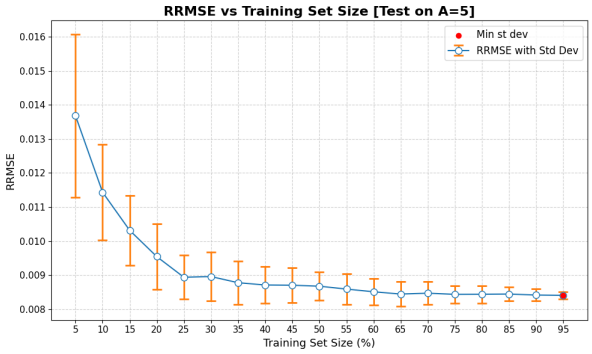
(b) Exclude A=2



(c) Exclude A=3



(d) Exclude A=4

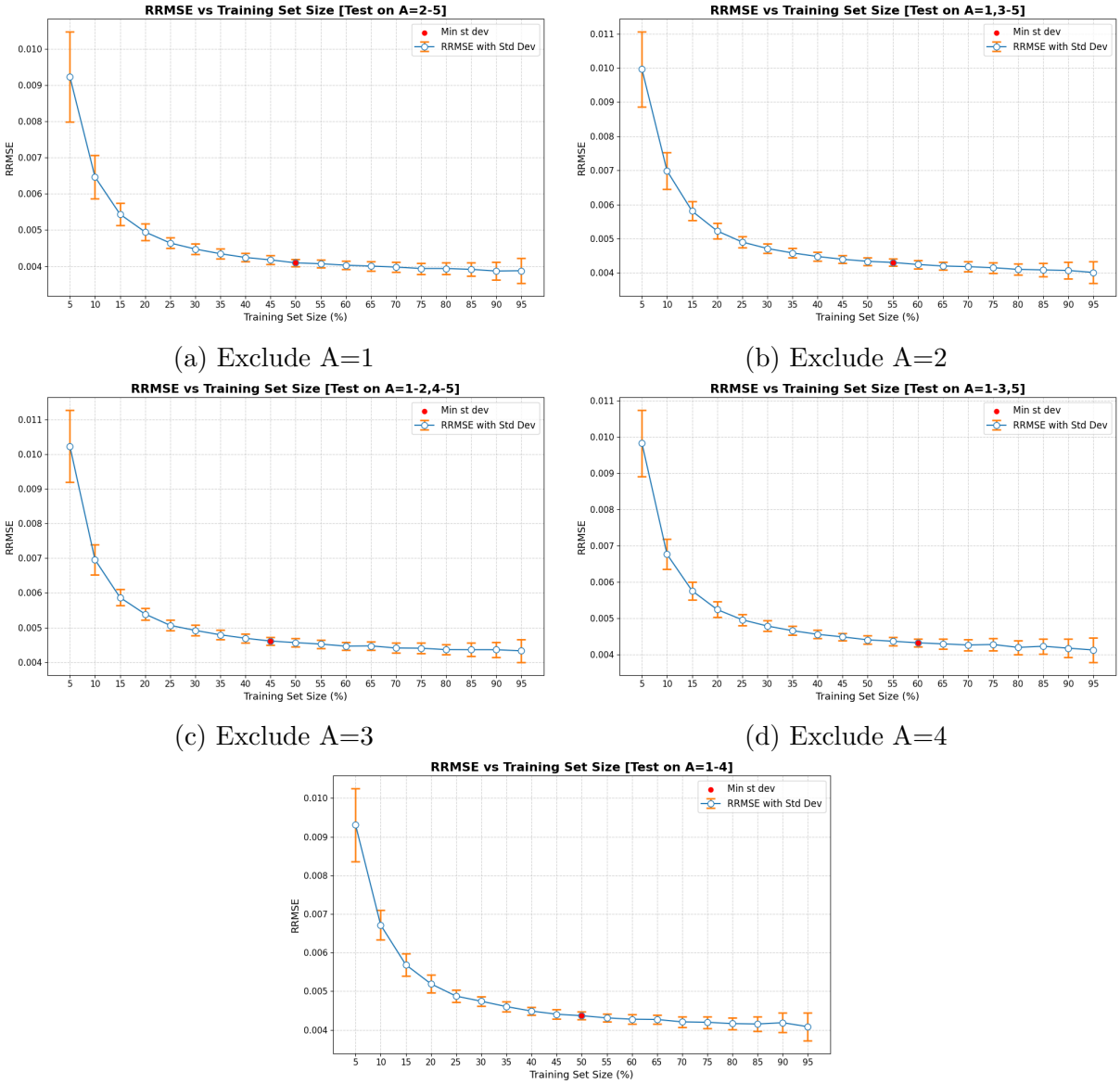


(e) Exclude A=5

Figure 13: RRMSE on testing sets, out-of-parameter

Now, let us look at the second set of tests performed. To rigorously evaluate the generalization capability of our ridge regression model, we implemented validation tests,

where one value of parameter A (e.g., progenitor mass, rotation rate, or EOS stiffness class) is removed from the training set, while testing is done on that left out value. The procedure consisted of training five different regression models, each excluding waveforms from one of the five values of parameter A. The Training set ratio in the figures below is the ratio of the number of waveforms belonging to four other values of A.



(a) Exclude A=1

(b) Exclude A=2

(c) Exclude A=3

(d) Exclude A=4

(e) Exclude A=5

Figure 14: RRMSE on testing sets, in-parameter

Fig. 13 above shows the out-of-parameter testing results. The RRMSE decreases monotonically, with the lowest standard deviation being the 95% ratio of the training set. Without the parameter-specific features, the generalization is more difficult. In contrast

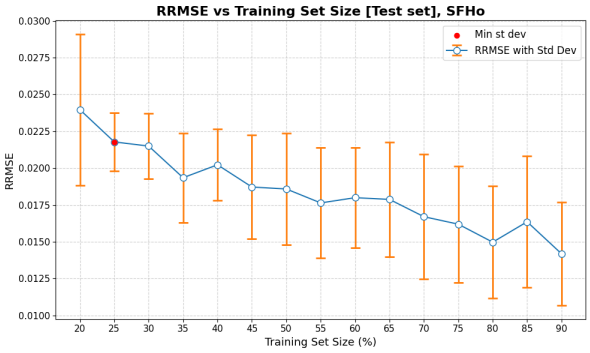
with these results, the in-parameter test (Fig. 14) showed the same results as in the previous section, with a notable difference being higher RRMSE due to the combination of all EoS.

Two significant observations can be made from the above figures. First, In-parameter testing consistently achieves the lowest RRMSE, demonstrating robust interpolation within the training domain. Second, out-of-parameter testing shows higher errors, with the border values of $A = 1$, and 5 exhibiting the highest RRMSE. The likely answer again lies within the scatter plots of principal components. While the waveforms' components of intermediate values for parameter A overlap with each other, the values of $A = 1$ for SFHo and $A = 5$ for GShenFSU2.1 are isolated. Which means, Ridge regression struggles with extrapolation outside of the training domain.

Tests on NNs

The evaluation of five neural network architectures for mapping revealed considerable variations in error across different network configurations. Five configurations represent different depths and widths of NNs.

(a) Deep-Medium



(b) Deep-Thin

(c) Shallow-Wide

(d) Shallow-Thin

(e) Hourglass

Figure 15: RRMSE of testing sets, NNs

The shallow medium-width architecture showed the highest accuracy, achieving the lowest test set RRMSE of less than 0.007 and a low standard deviation, representing a better outcome over other configurations. Being the widest and with the second highest number of parameters among configurations, it retained the majority of the features of a waveform. The deep-narrow and shallow-narrow architectures showed notably poorer performance, with RRMSE values of higher than 0.01. They also showed the highest variance in their runs, which means a strong dependence on the sampling for the training set. This shows that narrower configurations can poorly extract the features and generalize the data.

The configuration with the highest number of parameters is the hourglass configuration, and yet it did not result in the best accuracy, with RRMSE being just above 0.007. The reason probably lies within the narrowest point of the NN, where a lot of features are compressed and could be lost. Therefore, an increase in the number of parameters alone does not equate to an increase in accuracy.

DISCUSSION

Following the objective of this thesis, I explored the use of ridge regression and machine learning techniques to utilize computationally efficient GREP waveforms to generate full general relativistic simulations signals of core-collapse supernovae. This comparison of both techniques has highlighted some significant insights about them.

The results of ridge regression reflected a clear trend between model performance and training set size for different equations of state. We noticed that the relative root mean square error decreases monotonically with increasing training data, eventually reaching a level determined by the inherent limitations of the linear model. This asymptotic behavior provides a good reference point, i.e., a point of comparison, for testing the more sophisticated methods that could potentially be able to extract nonlinear features of the signals.

Interestingly, the deterioration of precision for parameters with low similarity components highlights the difficulty of extrapolation out of the training space, and it is suggested that any forthcoming works that may utilize this method should try to incorporate such waveforms. On the other hand, the waveforms achieved when the parameter A is of intermediate values are generated with much higher accuracy. Cross-parameter tests provided us with valuable feedback on the generalization capability of our models. This emphasizes the importance of representative training sets that comprehensively sample the related parameter space.

Our architecture study of neural networks showed that the shallow-wide configuration (80-40-40-80) achieved the optimal trade-off among accuracy and computational efficiency for SFHo waveforms. The enhanced performance of this structure over deeper (Hourglass) or thinner (shallow-thin and deep-thin) structures suggests that the preservation of information flow throughout the network is more significant than the number of layers in the network.

The successful demonstration of machine-learning-corrected waveforms suggests that approximate methods like GREP, if properly calibrated, can indeed produce waveforms of sufficient accuracy for detection and basic parameter estimation with detectors of today's generation.

Conclusions

The ability to rapidly generate accurate waveforms across a range of EOS models will be crucial for the analysis of potential coincident detections of GWs and neutrinos from Galactic supernovae. Building hybrid models that couple physical constraints to data-driven treatments can further refine extrapolation accuracy. Finally, generalizing these

techniques to other approximate schemes beyond GREP can put them to more use.

In conclusion, this paper has demonstrated that machine learning techniques can effectively address the deficiencies of approximate waveform generation software for core-collapse supernovae. With the efficiency of GREP simulations and the accuracy of numerical relativity with data-driven corrections, we have built a valuable toolkit capable of accelerating gravitational wave astronomy in the era of next-generation interferometers. These results not only have immediate application to data analysis, but provide a path for the future improvement of computational resources in theoretical astrophysics.

REFERENCES

[1] A. Einstein, "Über Gravitationswellen," Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften , 154 (1918).

[2] B. Müller, "Hydrodynamics of core-collapse supernovae and their progenitors," Living Reviews in Computational Astrophysics (2020), 10.1007/s41115-020-0008-5.

[3] H.-T. Janka, T. Melson, and A. Summa, "Physics of Core-Collapse Supernovae in Three Dimensions: A Sneak Preview," Annual Review of Nuclear and Particle Science (2015), 10.1146/annurev-nucl-102115-044747.

[4] K. Kotake, "Multiple physical elements to determine the gravitational-wave signatures of core-collapse supernovae," Comptes Rendus Physique (2013), 10.1016/j.crhy.2013.01.008.

[5] B. Schutz, *A First Course in General Relativity* (Cambridge University Press, 2009).

[6] J. Weber, "Gravitational Radiation," Physical Review Letters (1967), 10.1103/PhysRevLett.18.498.

[7] J. Weber, "Gravitational-Wave-Detector Events," Physical Review Letters (1968), 10.1103/PhysRevLett.20.1307.

[8] K. S. Thorne, "Gravitational Radiation – A New Window Onto the Universe," (1997), arXiv:gr-qc/9704042 [gr-qc] .

[9] R. F. Stark and T. Piran, "Gravitational-Wave Emission from Rotating Gravitational Collapse," Phys. Rev. Lett. **55**, 891 (1985).

[10] B. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), "Observation of Gravitational Waves from a Binary Black Hole Merger," Phys. Rev. Lett. **116**, 061102 (2016).

[11] B. Abbott *et al.*, "Detector description and performance for the first coincidence observations between LIGO and GEO," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **517**, 154 (2004).

[12] B. Abbott *et al.*, "GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral," The Astrophysical Journal Letters (2017), 10.3847/2041-8213/aa91c9.

[13] S. Hild, S. Chelkowski, and A. Freise, "Pushing towards the ET sensitivity using 'conventional' technology," (2008), arXiv:0810.0604 [gr-qc] .

[14] Dimmelmeier, H., Font, J. A., and Müller, E., "Relativistic simulations of rotational core collapse I. Methods, initial models, and code tests," AA **388**, 917 (2002).

[15] Marek, A., Dimmelmeier, H., Janka, H.-Th., Müller, E., and Buras, R., "Exploring the relativistic regime with Newtonian hydrodynamics: an improved effective gravitational potential for supernova simulations," AA **445**, 273 (2006).

[16] A. W. Steiner, M. Hempel, and T. Fischer, "CORE-COLLAPSE SUPERNOVA EQUATIONS OF STATE BASED ON NEUTRON STAR OBSERVATIONS," The Astrophysical Journal **774**, 17 (2013).

[17] G. Shen, C. J. Horowitz, and E. O'Connor, "Second relativistic mean field and virial equation of state for astrophysical simulations," Phys. Rev. C **83**, 065808 (2011).

[18] J. M. Lattimer and F. Douglas Swesty, "A generalized equation of state for hot, dense matter," Nuclear Physics A **535**, 331 (1991).

[19] M. Hempel and J. Schaffner-Bielich, "A statistical model for a complete supernova equation of state," Nuclear Physics A **837**, 210 (2010).

[20] A. Mitra, D. Orel, Y. S. Abylkairov, B. Shukirgaliyev, and E. Abdikamalov, "Probing nuclear physics with supernova gravitational waves and machine learning," Monthly Notices of the Royal Astronomical Society **529**, 3582 (2024), https://academic.oup.com/mnras/article-pdf/529/4/3582/57104269/stae714.pdf .

[21] Y. S. Abylkairov, M. C. Edwards, D. Orel, A. Mitra, B. Shukirgaliyev, and E. Abdikamalov, "Evaluating machine learning models for supernova gravitational wave signal classification," Machine Learning: Science and Technology **5**, 045077 (2025).

[22] H. Dimmelmeier, J. Novak, J. A. Font, J. M. Ibáñez, and E. Müller, "Combining spectral and shock-capturing methods: A new numerical approach for 3D relativistic core collapse simulations," Phys. Rev. D **71**, 064023 (2005).

[23] E. Abdikamalov, S. Gossan, A. M. DeMaio, and C. D. Ott, "Measuring the angular momentum distribution in core-collapse supernova progenitors with gravitational waves," Phys. Rev. D **90**, 044001 (2014).

[24] C. Pastor-Marcos, P. Cerdá-Durán, D. Walker, A. Torres-Forné, E. Abdikamalov, S. Richers, and J. A. Font, "Bayesian inference from gravitational waves in fast-rotating, core-collapse supernovae," Phys. Rev. D **109**, 063028 (2024).

[25] A. N. Tikhonov and V. Y. Arsenin, "Solutions of Ill-Posed Problems," (1977), ridge regression.

[26] A. Jadon, A. Patil, and S. Jadon, "A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting," (2022), arXiv:2211.02989 [cs.LG] .