PROBING NON-GENERAL RELATIVITY THEORIES USING DEEP LEARNING MODELS

Siddarth Ajith (author), Kent Yagi (advisor)
University of Virginia

Abstract: Gravitational wave parameter estimation is used to extract physical observables such as mass from gravitational wave signals. However, the conventional method is extremely time and resource intensive. This process can take up to a week to run for a single event, which will be computationally prohibitive as detection capabilities improve and scale. Recent work has put deep learning to use on this problem; neural networks can be trained in a fraction of that time, and they can be used to analyze the data virtually instantly. These models work for general relativity, and it is crucial to extend them to estimate beyond-general-relativity parameters in order to test a larger space of theories. Such theories can explain modern problems like dark energy and quantum gravity, and this neural network can be used to test gravitational theories efficiently. We found that our original models have some features that restrict the generality and accuracy of the resulting estimations. Following recent work, we are implementing autoregressive network flows which will improve and extend the results to be more general. We are currently tuning the model to improve the loss and accuracy, which is critical in making it a useful tool in analyzing future detections.

Introduction

Einstein's theory of general relativity (GR) is the most successful theory of gravity to date, having replaced Newton's theory of gravitation due to GR appropriately explaining the bending of light around the sun and the orbit of Mercury. Since these initial tests of GR, our ability to test gravitational theories has expanded, most importantly through the advent of gravitational wave GW astronomy[6]. Gravitational waves (GWs) are ripples in spacetime sourced by the mergers of extreme compact objects such as neutron stars and black holes. The spacetime around these mergers constitutes an important test bed for gravity; this is a region where the gravitational field is extremely strong and dynamical (fluctuating strongly with time). In the past eight years, the LIGO/VIRGO collaboration has detected nearly 100 gravitational waves, and GR has passed all tests put to it with flying colors. More interesting tests lay in the horizon with the NASA and ESA collaboration on the LISA mission. Next-generation detectors will open an even wider range of GW detection capabilities. Still, there are reasons to believe our understanding of gravitation is not final.

Modern physics mysteries such as the expansion of the universe, measurement of galactic rotation curves, and unexpected gravitational lensing requires the introduction of new matter and energy sources, the so-called dark energy and dark matter. Thus, the way that gravity works at the largest scales is rich with the possibility of new physics. Additionally, we are still learning new things in the strong-gravity regime, such as the spacetime around a black hole. We know GR breaks down as we approach the spacetime singularity, so there is yet another case where beyond-GR theories may prove to be useful. Finally, cosmological solutions predict a singularity which could indicate the need for a more advanced theory of gravity.

Many frameworks and procedures exist to test gravitational theories, but a particularly

powerful one is the parametrized-post-Eisnteinian (ppE) formalism[5]. This formalism allows for a generic mapping from beyond-GR theory to a set of parameters in the phase of the gravitational wave, each of which indicate the deviations of beyond-GR theories from general relativity. The parameters can be mapped to specific gravitational theories, and if one can measure these parameters in the signal of the GW, one can test *en masse* many theories of gravity. In principle, however, this can be computationally expensive or even prohibitive since GW parameter analysis is already such difficult problem.

To extract values of observables from the merging binary system which sourced a GW, one must do parameter estimation on the signal. The conventional method which estimates the Bayesian prior distribution using Markov Chain Monte Carlo sampling works very well, but it is incredibly time and resource intensive[2]. For double neutron star mergers, this analysis can take on the order of a week to analyze for a single merger event. As we scale up to detecting more than one GW event per day, this can be an extreme bottleneck in the process. Additionally, faster detection can inform us where to look for electromagnetic counterpart signals, which, if measured, would give us multi-messenger signals from which we can extract new physics. Multi-messenger signals are when we have gravitational and electromagnetic signal data from a given type of merger event. Thus, improving the efficiency of parameter estimation is crucial to improving GW astronomy.

Recently, machine learning has been put to use in order to improve this process[1,3,4]. A type of deep learning network called a conditional variational autoencoder network has been used to mimic the calculation of the prior distribution that the conventional methods find. This type of network has seen much use in image analysis, and the fruits of such work have become quite popular with AI generated art. CVAEs do parameter estimation by training the network to minimize the difference between the its output and the true Bayesian prior that encodes the physical parameters to be extracted from the signal. By training the network on simulated gravitational wave signals, the machine learning algorithm has been shown to give similar results to MCMC sampling[3]. However, the networks take on the order of days to train, and the networks run almost instantly. Thus, we have orders of magnitude speedup in computational time per event. There is much work to be done still. Relying on machine learning should come only after we know the results are accurate and reliable. Furthermore, the networks can be improved and extended to include ppE parameters, which can allow for efficient tests of GR.

The rest of the paper is organized as follows. We outline the ppE formalism, discuss the conditional variational auto encoder network, briefly discuss masked autoregressive flows, and finally give a description of the current status of the project. Then we summarize in a conclusion.

Parameterized-Post-Einsteinian Formalism

When doing the parameter analysis to match a GW signal to specific observable values, the theory that is being assumed will change the results. To create a more theory-agnostic framework, ppE formalism was developed[5]. To start, note that the signal of the gravitational wave can be split into its amplitude and phase, denoted by A and Ψ, respectively. The waveform of the GW, denoted as h, can be expanded as

$$h = A(t)e^{i\Psi} = [A_{\mathrm{GR}}(t) + \delta A(t)]e^{i[\Psi_{\mathrm{GR}}+\delta\Psi]}, \quad (1)$$

Where δA and $\delta\Psi$ encapsulate the deviations from general relativity. In general, differences

in the phase contribute more, so we shall focus on this term mainly.

The phase of the gravitational wave can be further parametrized by splitting the deviations in a sensible manner. A sensible splitting turns out to be a series expansion in the merging binary's orbital velocity (denoted u), giving a splitting that looks like

$\delta\Psi = \Sigma_j \beta_j u^j$.

In principle, there are infinite terms in this series, but for a feature being tested, only a few parameters may be of interest. Previous analysis has been done where one parameter at a time is tested, but ideally we want a method to test as many parameters so that all analysis is done free of theory-bias. These β_j parameters are precisely the terms we are looking to include in the network.

Deep Learning Methods

Conditional Variational Autoencoder Neural Networks

Briefly, we shall lay out what the computational challenge is that we set out to solve. The problem of extracting observable values from the data begins with having a model vs. the data. From this model and data, one should have a list of extracted parameters, which are observables, usually the masses of the black hole, the distance to the GW event from earth, and the time the black holes collide. In reality there are more parameters such as the spin of black holes, but we consider these initially to start our model. This naturally turns into a Baye's theorem problem, where we denote θ to be our observable values and s to be the signal data[1]. The signal is comprised of the waveform model h and noise n. The model can be put into what is called a latent space with variables denoted with z. This latent space essentially encodes aspects of the model. When z is present, this is where an explicit model is at play. The parameters given a signal is then the same as the parameters given the latent space and signal and the latent space formed given a signal. This is precisely the integral

$$p(\theta \mid s) = \int \mathrm{d}z\, p(\theta \mid z, s) p(z \mid s),$$

where in the integral there are two Gaussian distributions that get mixed into the final distribution that is more general in structure[3]. This is equivalent to

$p(\theta \mid s) = p(\theta \mid z, s) p(z \mid s) / p(z \mid \theta, s)$,

which more explicitly looks like a Baye's theorem problem. $p(z \mid \theta, s)$ turns out to be a computationally intensive step, so this is a good place to try to approximate using deep learning. The goal here is to construct $p(\theta \mid s)$ using a deep learning model. This is accomplished by making a network to represent all three of the expressions on the RHS of Eq. (). Since we have reduced the problem to a system of networks, let us explicitly define the networks to be given by

$$p(\theta \mid s) \approx \frac{p_{\mathrm{D}}(\theta \mid z, s) p_{\mathrm{E}}(z \mid s)}{q(z \mid \theta, s)} = p_{\mathrm{NN}}(\theta \mid s),$$

where the $\{p_E, q\}$ parts are known as encodes and p_D are the decoder[3]. This is where the neural network gets its name, conditional variational autoencoder (CVAE). The network is trained using two measures, the loss (denoted L) and the Kullback-Leiber (KL) divergence, . The loss measures how well the decoder gives the distribution of parameters, controlled by $p_D(\theta \mid z, s)$. The KL divergence measures how closely both of the encoders' outputs are. The idea is that at first $p_E(z \mid s)$ may not account for the true parameters, θ, very well, but as the network trains, the KL divergence will make sure this encoder outputs a latent space that will accurately capture features correlated to good guesses for the parameters given a random signal. $q(z \mid \theta, s)$ is a more "biased" encoder that accounts for both the parameters and the

signal. Finally, the decoder $p_D(\theta \mid z, s)$ is training to take in a latent space and signal and give a parameter estimation. How accurate this guess is determines the loss, which in turn tunes up the decoder. The neural network thus gives an approximation of the distribution we wanted, $p_{\mathrm{NN}}(\theta \mid s)$, can be tested for how well it replicates the true posterior distribution using cross-entropy

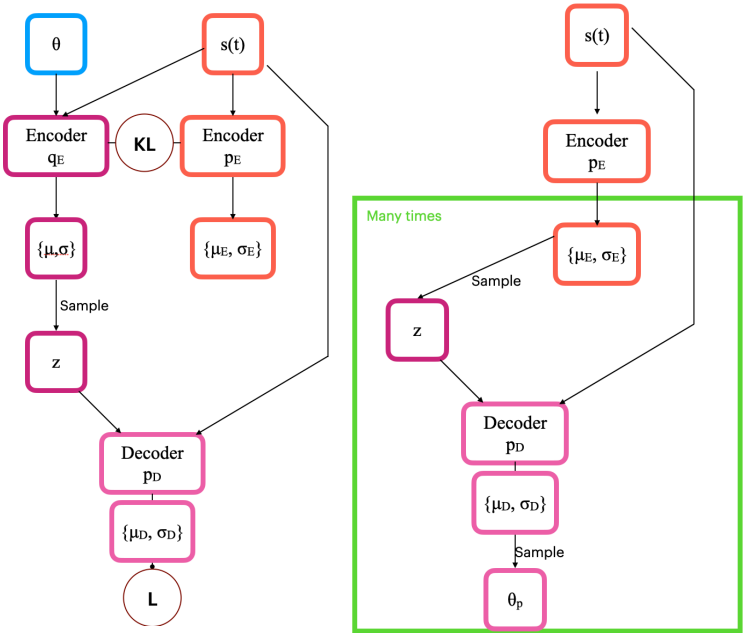
$$H = \int \mathrm{d}x p(\theta \mid s) \log p_{\mathrm{NN}}(\theta \mid s).$$

It can be shown that this is equivalent to an expression explicitly in terms of the loss and KL divergence, given by

$$H \lesssim \frac{1}{N} \Sigma_j \left\{ L_j(p_D) + KL_j(q, p_E) \right\},$$

where the N denotes the number of times the network is run in batches to train and j indexes a sum over all of these runs[3].

To make this set up less nebulous, let us see how the network is put together and interacts. The figures below show a schematic drawing of the CVAE[3]. The network architecture varies whether we test or train it.

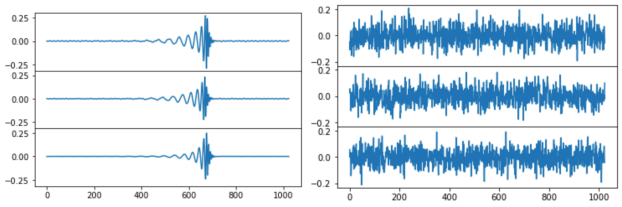


On the left, we have the training architecture while on the right, we have the test. When training, the "biased" encoder on the left helps the right encoder train by minimizing the KL divergence. This will train the right encoder how to handle signals without input as to what the correct value θ is. The decoder is trained by being given input from a sample of the latent space denoted z and the signal s. This allows the network to encode the modeling aspect of parameter estimation into the latent space created by a multivariate Gaussian of means μ and standard deviations σ. Note these are a vector of means and deviations, and they have a dimension equal to whatever the creator of the network deems fit. Often the power of these networks is the ability to create a latent space smaller than the number of parameters the network is trying to estimate. This means it can condense information into a small profile based on features the network finds to be important, and from the latent space and a signal, the decoder can make predictions. If the network is trained properly, the right encoder (p_{E}) will get better at making a latent space that best matches signals to true parameters without ever "seeing" what the true values were. The decoder is trained to take a sample of such latent space and create accurate parameter estimates. Thus the testing procedure is done using just $\{p_{\mathrm{E}}, p_{\mathrm{D}}\}$; p_{E} takes in a signal and encodes the signal into a latent space of Gaussians, parametrized by means μ_D and standard deviations σ_D. The decoder then takes in the signal and the latent space and guesses the parameters, outputting guess θ_p with distributions of means μ_D and standard deviations σ_D[3].

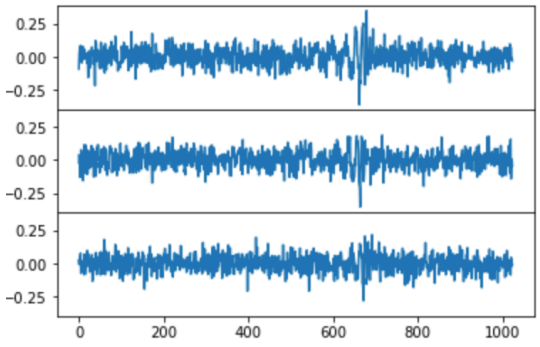
Note there are a few subtleties to watch out for. One common example is overfitting which in this case can lead to "posterior collapse" where the encoders are too similar in

guesses. This will likely lead to inaccurate guesses since the encoders should hold some generality and should make guesses different from the exact examples it has already seen[4]. This is why it is important that the encoders are separate; part of the power of this methodology comes from the second encoder being somewhat blind to the true values, allowing its predictive power to be more generic.

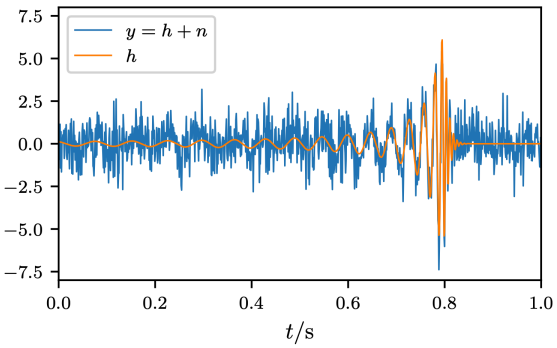
Previous work has put this kind of CVAE network to use in GW parameter



estimation[3]. We aimed to replicate this result and then extend the network. To do so, we made our own training data, which requires generating simulated noise and simulated waveforms. These are shown below:

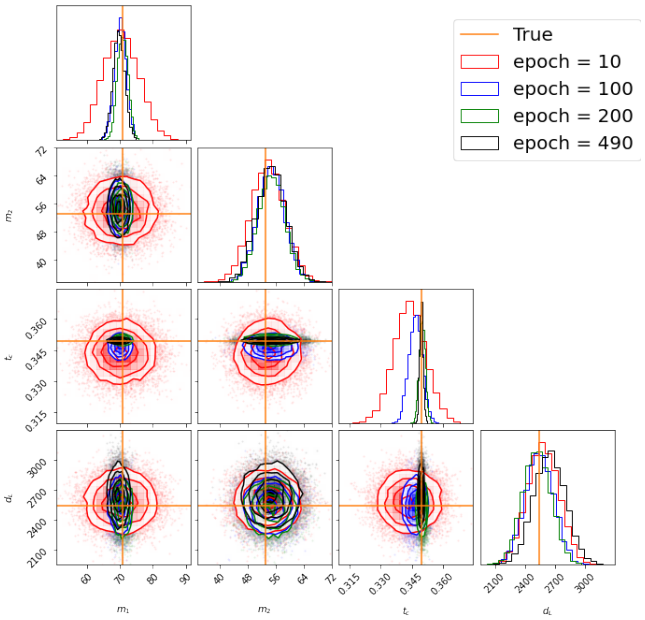


These are then combined to get the full simulated signal (waveform+noise=signal).



The above figure with orange is from Green, et al. and illustrates a fit waveform in a signal[4]. To improve the model, one could use more realistic noise realizations like real LIGO/ VIRGO noise values.

Below I show an output of the neural network.



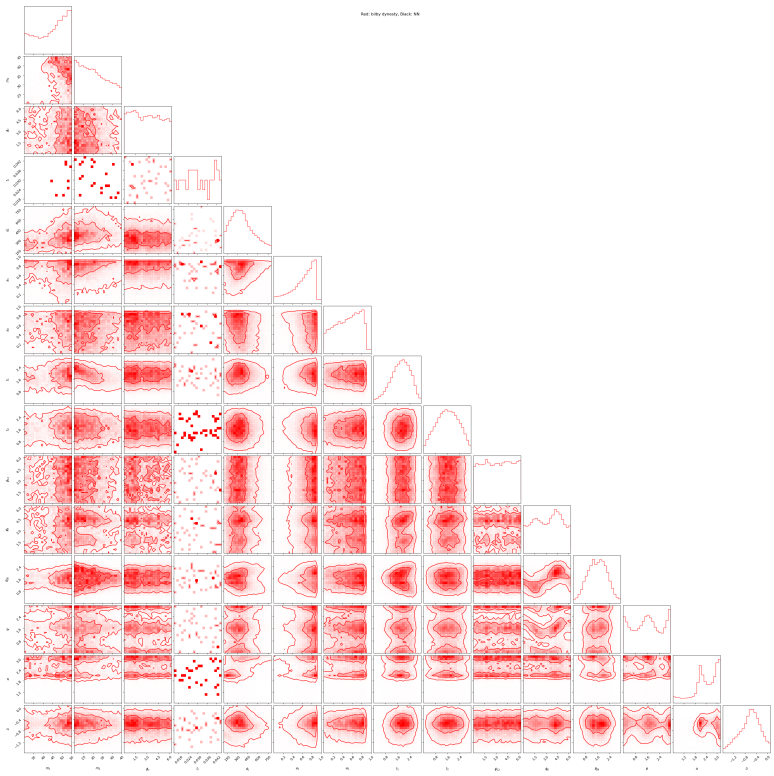
The parameter estimates are denoted by epoch, where contours denote confidence intervals and higher epochs narrows the uncertainty range of the estimates. We can see by almost 500 epochs we get fairly accurate results, and the network takes about 1-2 days to train. One thing to notice however, is that the predictions are very Gaussian in shape.

To get a more general shapes (less Gaussian), more advanced techniques may be applied. The mentioned previous work has gotten CVAE alone to get quite amazing results, but a straightforward way to improve the generic features that can be captured is by combining CVAE with other techniques like masked autoregressive flow, a type of normalizing flow[3,4].

Masked Autoregressive Flow and Current Status

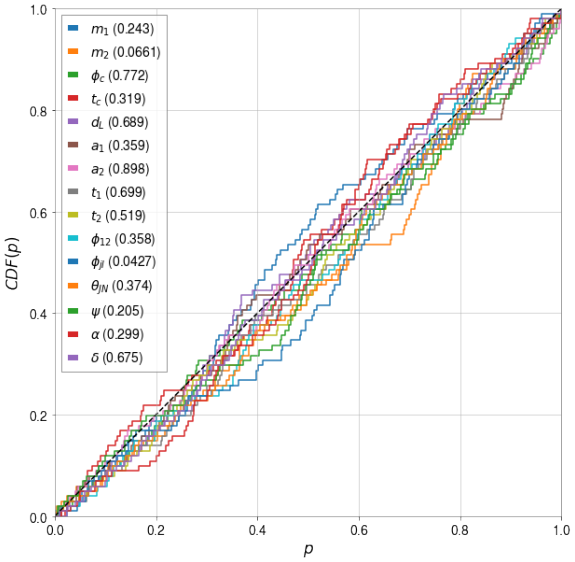
Broadly speaking, masked autoregressive flow (MAF) networks were made to make encoder networks more flexible/ versatile[4]. The math behind this method is a bit slick, where the chain rule and properties of Gaussian distrbutions are used in conjunction, but we paint a more qualitative picture of how the technique is used. This addition of masked layers to make MAF is based on the work of Green et al. To build a masked layer, one starts with a single masked autoencoder layer (a fully connected network with specific layers then disconnected). This is a network with very specific geometry, but a single layer will be somewhat Gaussian. As you add layers, you get much more complex distribution geometry, and as you build the MAF network, the non-Gaussian nature emerges from stacking more and more layers[4]. However, too many of these layers may require you to add a normalization layer to make the distribution easier to sample. Essentially, the latent space of the above CVAE network can have MAF layers added to it before the decoder. The exact configuration is largely up to the programmer making the network, but some combination of MAF and normalization layers will mix up the latent space and make the CVAE network more versatile. These non-Gaussian filters allow more generic features to be captured by the network.

Currently, the model we have has some MAF layers incorporated after the latent layer, and an example output parameter estimation is included in the next page. We can thus see that more parameters can be estimated here, and the distributions as a whole are more flexible in their parameter estimations rather than always clumping into normal distributions. The literature uses p-p plots to compare the accuracy of these networks. The idea of this



kind of analysis is to compare two cumulative probability distributions, and the closer the distributions are, the closer the lines are to being at 45 degrees, along the central line[3,4]. The p-p plot from the above parameter estimation is given on the following page. We note that the accuracy is not sufficient to compete with the conventional methods yet, but it is a good starting step. To improve this network output, we need to tune many aspects of the model. This includes layer size, MAF layers and normalization layers, and the hyper parameters (values controlling how the network is interconnected). This process can be tricky, and it takes a lot of trial and error to see what improves the model.

We have thus far excluded ppE parameters from this iteration until we can improve the accuracy of this revision. Once this gets smoothed over, we will go back to extending the parameter space so that our network will help with tests of GR. As



upgrades to the network are made, it is often necessary to reduce the parameter space until a working model is constructed. The tuning will likely be guided by what changes will make the loss function more closely match previous work, and afterwards we need to study what will tune the accuracy to be higher (i.e. make the p-p plot lines converge towards the central line).

Conclusion

Deep learning has proven to be a useful and powerful asset to GW astronomy. New uses are constantly being found, and GW data analysis is an especially fruitful use case for deep learning[1,3,4]. CVAE networks in conjunction with other recent network architectures are quickly becoming popular topics of investigation, and the promised speedup makes the endeavor worthwhile.

Through our investigations, we started with CVAE networks to do our parameter estimation, but when testing the network, we found that new methods could greatly help the flexibility of our network. The largest area of improvement when it comes to the flexibility of our network is being able to capture features that deviate from non-Gaussian distributions. In general, GW parameter estimation needs to produce fairly versatile posteriors. Adding MAF layers to the latent layer of our encoder has helped this issue, but we must improve the accuracy of this new iteration. Refining these networks will require further tuning, but once this is done we can keep adding parameters for the network to estimate. The goal is to incorporate as many features as possible, but, due to the stochastic nature of this work, any changes can lead to unpredictable complications. A complete network that includes ppE parameters will be powerful in efficiently testing gravitational theories, and such tools are crucial to fully realizing the possibilities that future detectors like LISA will afford us.

Acknowledgements

S. Ajith acknowledges the support of the Virginia Space Grant Consortium for this work. S. Ajith also recognizes that this work is done with graduate students Nan Jiang and Sheng Zhang at University of Virginia.

References

[1] Alvin J.K. Chua et al., *Phys.Rev.Lett.* 124 (2020) 4, 041102.
[2] D. Foreman-Market et al., *Publ.Astron.Soc.Pac.* 125 (2013) 306-312
[3] H. Gabbard et al., *Nature Phys*. 18, 112 (2022), arXiv:1909.06296 [astro-ph.IM].
[4] S. Green et al., *Phys.Rev.D* 102 , 104057 (2020), arXiv:2002.07656 [astro-ph.IM].
[5] N. Yunes and F. Pretorius, *Phys. Rev. D* 80, 122003 (2009), arXiv:0909.3328 [gr-qc].
[6] C. M. Will, *Living Rev. Rel.* 17, 4 (2014), arXiv:1403.7377 [gr-qc].