

Backpropagating Gravitational wave events

KAZE W. K. WONG¹

¹*Center for Computational Astrophysics, Flatiron Institute, New York, NY 10010, USA*

(Dated: May 23, 2022)

ABSTRACT

In this



1. INTRODUCTION

As the rate of detection of gravitational-wave (GW) increase with the sensitivity of the GW detector network [KW:], understanding astrophysical processes related to progenitors of GW events through analyzing a population of GW events has become more popular [KW: cite]. A very common way to study the origin of detected GW events is through population synthesis simulation [KW: cite], which evolves a population of binaries to form a population of GW events. Forward modelling of a population of GW events start with sampling a set of initial binary parameters from some distributions related to the progenitor population, such as the initial mass function or star-formation history, then evolve each individual members according to the physics of interest until some stopping criteria is met. This usually mean either a merger event or the binary system stays stable for a Hubble time. After this, one can collect all the GW emitting binaries in the synthetic population and compare to the observed population.

[KW: Shorten] While forward modelling has been the canonical pathway to simulate binary systems and compare to GW observation, it is not necessarily the most efficient way to constrain the underlying physical model. Most of the systems do not end in a merger, and the systems ending up as merger events do not necessarily agree with the observation. This means if we want to evolve a binary to match the observation by pure trial and error, a large fraction of computation will be wasted in trying initial binary configurations that are not related to the observation. To alleviate this inefficiency, the community often takes a population approach — instead of trying to match binary systems to observations event by event, one can draw systems from an initial

population, forward model each member in that population then match it to the observed population properties. However, this approach is essentially trading one inefficiency with another, since we still need to choose an initial distribution of binary parameters. For relatively well-known source properties such as initial masses this is less of a problem, but the distribution of some less well-studied properties such as common envelope efficiency is often heuristically chosen to be some simple distribution such as a delta function at some value, i.e. a fixed value for the entire population. This is a problem when the target of a study is to understand these less studied parameters, since one either need to justify their choice of distribution with other independent studies, or they should vary the family of distributions. In general, it is unclear how one can cover all possible distribution that may agree with the observation in generating a suite of simulations. The problem worsen as we increase the dimensionality of the parameter space. WF: This says what you want to say, but it's a bit wordy right now—I'll return to it, but it would be good to tighten up the language a lot.

Indeed, one of the most common skepticism population synthesis studies receives is a simulation set usually only cover a small subdomain of the entire parameter space, i.e. only vary a number of hyper-parameters while holding all others fixed to some "default" values. Even with recently development on using emulators to speed up the simulation process, training the emulators also require a decent amount of samples in the parameter space of interest. For example, say we want to emulate a model with 10 parameters, this simplest option for constructing a bank of simulations for training is to lay down a grid. Along each axis we put the barely minimum of 2 simulations, this would still require 1024 simulations. The fastest kind of population synthesis simulation takes about an hour to complete, where longer

simulations could take days. This means even training an emulator is impossible in high dimensional space.

In this paper, instead of forward modelling a population of binary star to match the observed GW events population, we propose a method to "backward" model each GW event to its progenitor state. The idea is since we can compute the GW event properties of a guess by forward modelling its progenitor, we can define the difference between the GW event properties of this guess and the observed GW event properties. By minimizing this difference, i.e. solving a root finding problem, we can hope to recover the progenitor properties.

This paper is structured as follows: We detail our method in Sec. 3. In Sec. 4, we show the results of applying our method to a number of real events. We discuss the implications of this work and future direction in Sec. 5.

2. METHOD REWRITE

[WF: Re-writing the methods section with a new viewpoint, gonna try to eventually merge the content.](#)

We define the parameters that describe the progenitor's properties such as ZAMS masses and eccentricity as progenitor parameters θ' , the parameters that control different physical prescriptions such as wind strength and common envelope efficiency as hyper-parameters λ , and random variables that affect certain stochastic process such as whether natal kick unbinds a binary as \mathbf{X} . To avoid clutter in the following derivation, we denote all the parameters related to mapping a particular progenitor system into a GW event collectively as evolutionary parameters Θ , which includes θ' , λ , and \mathbf{X} .

Once all the parameters including a particular draw of all the random variables represented by \mathbf{X} are known, the population synthesis code is a deterministic function that transforms the properties of the progenitors into the GW parameters θ :

$$\theta = f(\Theta). \quad (1)$$

The mapping f from Θ to θ is many-to-one. In order to draw inferences about Θ from gravitational wave data, we need to be able to evaluate the likelihood of that data at fixed progenitor parameters,

$$p(d | \Theta) = p(d | \theta = f(\Theta)), \quad (2)$$

because the likelihood depends only on the gravitational waveform generated by parameters θ to which the progenitor parameters map. In principle this likelihood could be computed at arbitrary values of Θ using the same machinery that is used to estimate source parameters in gravitational wave catalogs ([Veitch et al. 2015](#);

[Ashton et al. 2019](#); [Romero-Shaw et al. 2020](#); [The LIGO Scientific Collaboration et al. 2021](#)) to develop samples of progenitor parameters, hyper-parameters, and random variables in Θ from a posterior density.

However, it is very expensive to couple progenitor evolution to stochastic sampling in this way. Because the gravitational wave likelihood function ultimately depends only on the GW parameters θ , and catalogs have already produced samples over these parameters from a posterior density

$$p(\theta | d) \propto p(d | \theta) p_{\text{GW}}(\theta), \quad (3)$$

where p_{GW} is the prior density used for sampling, it is much more efficient to [WF: WIP](#).

3. METHOD

We define the parameters that describe the progenitor's properties such as ZAMS masses and eccentricity as progenitor parameters θ' , the parameters that control different physical prescriptions such as wind strength and common envelope efficiency as hyper-parameters λ , and random variables that affect certain stochastic process such as whether natal kick unbinds a binary as \mathbf{X} . To avoid clutter in the following derivation, we denote all the parameters related to mapping a particular progenitor system into a GW event collectively as evolutionary parameters Θ , which includes θ' , λ , and \mathbf{X} .

Once all the parameters including a particular draw of all the random variables are known, the population synthesis code is a deterministic function that transforms the properties of the progenitors into the GW parameters. Hence, the probability of obtaining a GW event with parameters θ given the progenitor parameters Θ is given by

$$p(\theta | \Theta) = \delta(f(\Theta) - \theta), \quad (4)$$

where f represents the evolution encoded by the population synthesis model with fixed random parameters \mathbf{X} .

While it is straightforward to write down the "forward" probability, the "backward" probability is more complicated since multiple evolutionary parameters can produce the same GW event. Using Bayes theorem, the probability of an event with evolutionary parameters given the GW parameters can be derived as following

$$\begin{aligned}
p(\Theta|\theta) &= \frac{p(\theta|\Theta)p(\Theta)}{p(\theta)}, \\
&= \frac{p(\theta|\Theta)p(\Theta)}{\int p(\theta|\Theta)p(\Theta)d\Theta}, \\
&= \frac{\delta(f(\Theta) - \theta)p(\Theta)}{\int d\theta \delta(f(\Theta) - \theta)p(\Theta)}. \tag{5}
\end{aligned}$$

Using the result in eq.5, the probability of an event having evolutionary parameters given data can be obtained by marginalizing over the GW parameters

$$p(\Theta|d) = \int p(\Theta|\theta)p(\theta|d)d\theta \tag{6}$$

$$= \frac{1}{N} \sum_{i=1}^N p(\Theta|\theta_i). \tag{7}$$

The term $p(\theta|d)$ is the inferred posterior probability for a particular event, which is usually presented by the LVK collaboration in the form of posterior samples. In eq.7, we use importance sampling to turn the integration into a sum over the posterior samples.

For each posterior sample point, we can find the corresponding evolutionary parameters by solving an optimization problem that minimize the mean square difference between the GW event parameters and evolutionary parameters, i.e.

$$\mathcal{L}(\Theta, \theta) = \|f(\Theta) - \theta\|^2 \tag{8}$$

In principle, we should only accept solutions that exactly reproduce the LVK posterior samples in the observable space. However, it is not feasible to achieve such condition in practice, therefore we relax the condition in eq.8 to a small acceptance threshold, in our case we picked 10^{-2} to be our threshold. To make sure we find a reasonably complete set of progenitor parameters that corresponds to the posterior sample point, we use 1000 different initial guesses in solving the optimization problem. As long as the solution fulfill the acceptance criteria, the root is as valid as all other roots that fulfill the same acceptance criteria, regardless to its initial guess. In other word, we have the freedom to choose the set of initial guesses however we think it might benefit the optimization. Most of the points in the evolutionary parameter space do not produce mergers, and for systems that do not merge, we set the binary masses to be 0. This means the gradient for systems that do not merge is 0, and the root finder would likely to get

stuck in such state during the first step.¹ Therefore, it is beneficial to put initial guesses in region of the evolutionary parameter space that does produce merger. To construct a list of initial guesses, we evolve a set of points that we uniformly sampled from the evolutionary parameter space, then keep the points that merge within a Hubble time.

Ideally we would like to have explicit control over all evolutionary parameters. In particular, having explicit control over random variables allows us to marginalize over their contribution, so we can focus on progenitor parameters and hyper-parameters. However, in practice most of the population synthesis simulation code (including COSMIC which we use in this study.) define their random variables implicitly, i.e. the random variables are drawn within the program, instead of being arguments of the program. This means the function we use to evolve the binary is not a deterministic function but a stochastic function instead. Hence, even if the root-finding algorithm performs perfectly, forward modelling this set of roots does not guarantee to reproduce the set of posterior samples due to randomness of the evolution function. To assure the recovered progenitor and hyper-parameters robustly corresponds to the posterior in the GW observables space, we forward model ("Reproject") the recovered evolutionary parameters to the observables space to check whether the reprojected posterior agrees with the posterior given by the LVK collaboration. We use KL divergence to measure the agreement between the two posterior distributions, which is given by

$$D_{KL}(P||Q) = \int P(x) \log(P(x)/Q(x))dx. \tag{9}$$

A small KL divergence means the reprojected COSMIC posterior is similar to the original posterior in the observable space, hence it is a viable channel for that specific event, otherwise it either means COSMIC can only explain part of the posterior or cannot explain the event at all. This is completely expected since COSMIC comes with its own assumptions and is expected to fail in reproducing a subset of the event in GWTC3, such as event with at least one component beyond the pair instability supernova massgap. On the other hand, the reprojection could subject to stochasticity in the evolution function we use. Instead of reprojecting the posterior only once, we reproject the posterior multiple times with different random seeds and check whether the KL

¹ On a plateau of constant loss, the root finder performs no better than a random guess. In a high dimensional space, this means the root finder will need a long time to get close to a solution.

divergence varies significantly. A varying KL divergence means this particular event is subjected to randomness in the evolution function, hence one should take extra caution in interpreting the result.

On top of using the raw result from the root-finding process, we also perform MCMC sampling² using the roots to initialize the MCMC chains. The two results agree well with each other. This should eliminate all concern related to the ambiguity in interpreting the roots because MCMC result should not depend on the initialization.

4. RESULT

1. Raw posterior of successful events.
2. Trace back the event's z_{ZAMS} .
3. Reproject to observable space
4. Evidence ratio?
5. KL divergence?

We apply the method described in section 3 to the first GW events GW150914. The parameters involved in the analysis and the range we allow for finding the root are tabulated in table 1. [KW: @Katie state why not spin here.] Therefore, we solve the rootfinding problem with only the two component masses in the observable space. For progenitor parameters, we characterize each guess with five parameters: the two component masses $M_{\text{ZAMS},1}$ and $M_{\text{ZAMS},2}$, the orbital period t_{orb} , the eccentricity e , the metallicity Z at ZAMS. Note that the progenitor formation redshift is not obtained through the root-finding process. This is because the formation redshift does not affect the evolution of the binary, so we do not need to include in the optimization. Instead, we compute the time it takes for the binary to evolve from formation to the merger, then we add this time back to the lookback time when this event is observed. Using the total lookback time, we can compute the redshift of the binary at formation. We adopt the cosmological model in [KW: cite Planck 18] to compute the redshift of the binary at formation. For hyper-parameters, we choose parameters that affect binary evolution in COSMIC mostly significantly, including the common envelope efficiency α , the fraction of accretion during stable mass transfer f_{acc} , the critical mass fraction $q_{\text{crit},3}$, and the accretion rate σ . We show a portion of the joint posterior of the progenitor parameters and hyper-parameters of GW150914 in figure 1.

² We use sampling package `emcee`[KW: cite] for the MCMC.

[KW: Recover ratio of the posterior sample goes here.]

The full posterior is available in appendix [KW: Or we should put a link to the dataset?]

To check the validity of the recovered posterior, it is essential to check whether it is consistent with the original posterior in the observable space. We reproject the recovered posterior to the observable space, and compare the KL divergence between the original posterior and the reprojected posterior. For GW150914, the KL divergence between the reprojected posterior and the original posterior is [KW: fill later] on average. The KL divergence stays stable for reprojecting with different random seeds.

We can see the reprojected posterior agrees very well with the original posterior in fig.2. [KW: Quote the KL divergence.]

The most important point in this paper is illustrated in figure 3. We can see the posterior samples in $M_1 - M_2$ space are spatially correlated with α , this means if we fixed the value of α instead of allowing it to jointly vary with all other parameters, we would miss the samples that do not match the chosen value of α . By folding in the hyper-parameters into the root-finding algorithm, we can capture the correlation between [KW: Dartboard comparsion] Comparing our result with [KW: Add figure showing hyper-parameters are important to recovering the progenitor parameters correctly.]

5. DISCUSSION

We present a new pathway to understand binary evolution with GW events in this paper. Instead of forward modelling an assumed distribution of initial binary to the observed population, we solve a root-finding problem to find the best initial binary parameters that reproduce the posterior distribution for each event. The most obvious advantage our method provides is it returns the joint posterior distribution of progenitor parameters and hyper-parameters for each event. This enables a data-driven way to study the distribution of hyper-parameters, i.e. once we have a catalog of events, each has their own set of posterior in the hyper-parameter space, we can employ common techniques such as hierarchical Bayesian analysis to fit a population model to the distribution of hyper-parameters, without making specific assumptions such as fixing the hyper-parameters for all types of event.

In this work we showcase the power of the proposed method with two events, GW150914 and GW191129. We show the joint posterior of the two events' progenitor parameters and hyper-parameters. In particular, to the authors knowledge, this is the first time the correlation

Parameters	Description	Optimization range
Observables		
$M_{\text{GW},1}$	Primary mass of the GW event	NA
$M_{\text{GW},2}$	Secondary mass of the GW event	NA
Progenitor parameters		
$M_{\text{ZAMS},1}$	Primary mass at ZAMS	[10, 150] M_{\odot}
$M_{\text{ZAMS},2}$	Secondary mass at ZAMS	[10, 150] M_{\odot}
t_{orb}	Orbital period at ZAMS	[5, 5000] days
e	Eccentricity at ZAMS	[0, 1]
Z	Metallicity at ZAMS	$[10^{-4}, 2 \times 10^{-2}]$
z	Redshift at ZAMS	NA
Hyper-parameters		
α	Common envelope efficiency	[0.1, 10]
f_{acc}	Fraction of accretion during stable mass transfer	[0, 1]
$q_{\text{crit},3}$	[KW: Katie fill in]	[0.1, 10]
σ	[KW: Katie fill in]	[10, 300] km/h

Table 1. A list of parameters we used in this study.

between progenitor parameters and hyper-parameters is shown. Our work is not only a more efficient way to study GW events’ progenitors, but also allows the possibility of measuring the astrophysics related the binary evolution, especially in capturing the correlation between different hyper-parameters.

Another perk of our method is the ”rate” calculation is now independent of assumed initial condition such as a particular star formation rate model, which is a dominant factor of uncertainty for rate prediction in the past. Once we have pull back the GW event posterior from the observable space to the evolutionary parameter space, we can apply the same methods we use to account for the rate in the observable space. In the observable space, the overall rate is basically a normalization that is defined in the local universe i.e. the local merger rate. [KW: cite] In the evolutionary parameter space, we can instead fit for the production rate of merger events at some redshift. [KW: Talk about prior are now in post-processing instead of pre-processing.]

While our method allows data-driven exploration of the hyper-parameters space for the first time, there are a number of improvements can be implemented in future studies. In this study we use only COSMIC as our evolutionary function, which by design cannot explain all the events in GWTC3. For example, event with either of the component mass larger than the PISN gap such as GW190521 cannot be explained by COSMIC. Alternative channels such as dynamic formation channel will be needed to explain some subset of the events in GWTC3. As pointed out in the literature [KW: Cite mike], it is

unlikely that all the observed events in GWTC3 can be explained by a single formation channel. As GW detectors sensitivity increase, we expect to see more and more events that are unusual in some way. Therefore, having a self-consistence population synthesis code that contains multiple formation channels is essential to accommodate the growing catalog of GW events. In this paper, our main focus is to illustrate the concept of ”back-propagating” GW events posterior sample in this work, highlighting the capability of our method and motivating the community to build the next generation population synthesis code that can work with our method. To avoid cluttering of focus, we discuss the physical implication of the result presented in this work under our specific assumption (i.e. using COSMIC as our evolutionary function) in a companion paper.

Due to the implicit definition of random variables in COSMIC, our evolutionary function is stochastic. This introduces significant inefficiency in our root-finding algorithm. The main stochasticity in COSMIC comes from natal kick, which significantly affect the evolution pathway of low mass events such as BNS events. The effect of natal kick is suppressed for heavier mass events due to fallback. This means events with the root-finding process for lighter masses are subject to stochasticity of the function, where the root-finding process for heavier events behaves as if the evolutionary function we use are deterministic. Indeed, we see this behavior in a couple of case studies. For heavy event such as GW150914, the reprojected posterior agrees with the data posterior. For light event such as GW170817, although during the

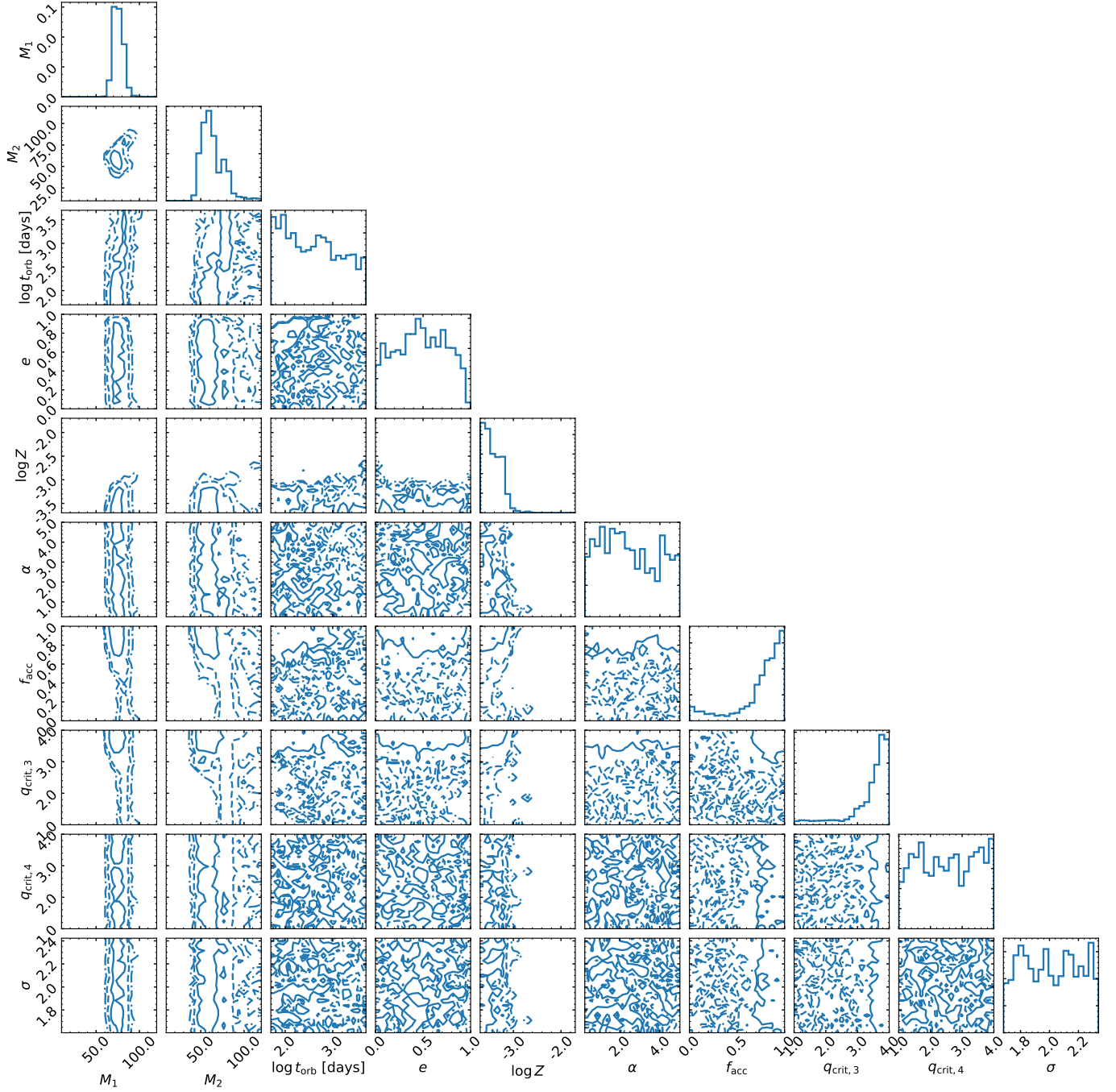


Figure 1. The posterior for GW150914 in both the progenitor parameter space and hyper-parameter space. M_1 and M_2 are the progenitors' masses. $\log Z$ is log metallicity at ZAMS. z is the redshift at ZAMS. α is the common envelope efficiency. f_{acc} is the fraction of mass accreted during stable mass transfer. $q_{\text{crit},3}$ are the critical mass during [KW: Katie fill in], Note that the redshift is not fitted during the root finding process. Once we have find the evolutionary parameters, we add the time it would take the progenitor to merge to the lookback time of the observed posterior sample, then from the total lookback time we can compute the redshift at ZAMS.

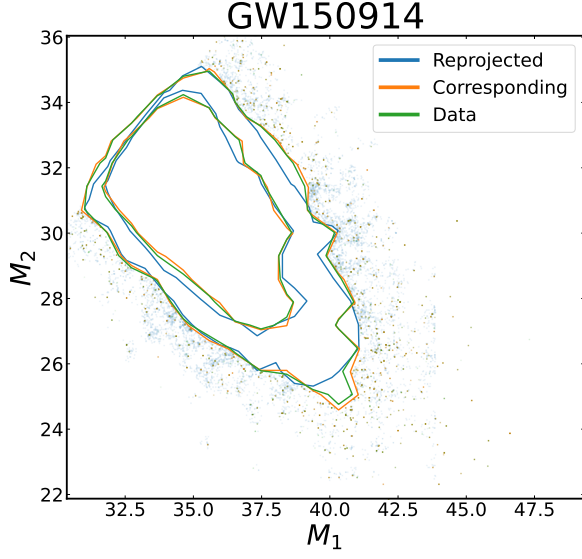


Figure 2. Reprojecting the posterior in evolutionary parameter space of GW150914 to observable space. The blue contour is the posterior reprojected using the samples in the evolutionary parameter space. The orange contour is the posterior samples from LVK that have corresponding samples in the evolutionary parameter space. The green contour is the posterior plotted using all of LVK posterior samples.

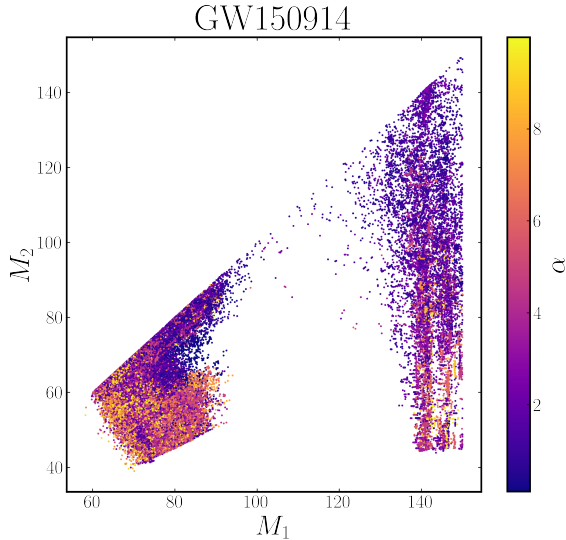


Figure 3. The posterior samples of GW150914 in the progenitor masses space, colorcoded by the common envelope efficiency α for that sample.

root-finding process the convergence criteria is met, the reprojected posterior does not agree with the data posterior. Also, due to computational limitation, we only try 1000 different initial guesses per posterior sample in the root-finding process. This means any posterior sample that has a probability of merging rarer than 1 in 1000 could be missed. [KW: comment on the low mass ratio end.] Obviously the problems which comes with the randomness can be alleviated by performing more tries per posterior sample, but this is not scalable in practice. On top of limitation in efficiency, some formation channels require explicit control of random variables by construction. For example, in a dynamical formation scenario such as binaries that form in a globular cluster, each binary has some probability of undergoing a multi-body encounter with another member in the cluster. These encounter probability distributions are either studied with direct N-body simulations or semi-analytical methods. In both cases, each member of the cluster is no longer completely independent of the other members, but coupled through the encounter probability distribution. By studying the encounter probability distribution, we can infer the properties of the environment which the binary lives in. This can only be done if we have explicit control over the random variables that characterize the encounter probability distribution.

By leveraging gradient-based optimization strategy, obtaining posterior samples in the evolutionary parameter space is much more efficient than rejection sampling. In each optimization step during the root-finding process, the current state of the guess and its gradient provide useful information about where the root might be. In contrast to sampling algorithm, which discard the sample whenever it is not accepted. This greatly benefit the efficiency of the root-finding process. On the other hand, this method is not a sampling algorithm, but a coordinate transform of the posterior samples. One should pay extra caution in interpreting the posterior samples in the evolutionary parameter space. Since posterior samples in the observable space can in principle have multiple roots in the evolutionary parameter space, the relatively weighting of these roots becomes ambiguous when the number of roots is large. For example, should we consider roots that are close to each other a single root or multiple roots? If we have two clusters of roots, the relative number of roots within each cluster would determine the relative weight of each mode. Note that since they both map to the same observable posterior sample within the acceptance threshold, all weighting assigned are equally valid from a "matching data" standpoint. This leave the interpretation of the posterior in the evolutionary parameter space ambiguous. In our

case, we found empirically each posterior sample may at most only correspond to a handful of roots (most of the posterior samples have a unique root.), therefore we choose to assign equal weight to each root.

The potential reason for the good agreement between MCMC and the root finding is due to the behavior of posterior in the evolutionary parameter space are larger determined by the posterior distribution in the observable space instead of multiple possible roots per posterior sample. We have also compared two MCMC results, one is initialized using the roots and another is initialized using samples drawn from a uniform prior. We found empirically the MCMC result using the roots as initialization converges at least an order of magnitude faster than the uniform prior initialization. This makes sense since the roots basically eliminate the long burn-in needed to go from random location in the evolutionary parameters to the most probable region. We leave a detail comparison between this method and direct sampling to future studies.

Another technical note is we use finite differencing to estimate the gradient of the objective function, which could be a significant source of error near transition points in the evolutionary parameter space. Also, finite differencing is increasing inefficiency as we increase the dimensionality of the problem. To improve the accuracy and efficiency in estimating the gradient of objective function, automatic differentiation is a promising feature that modeler should consider incorporating in their population synthesis code in the future.

[KW: Mention potential complimentary studies of direct sampling, and how all these suggestions are still useful regardless of whether we are doing root-finding or not.]

To summarize, we propose a novel method to recover the posterior samples in the evolutionary parameter space for each GW event. We point out hyper-parameters in the usual population synthesis simulation context are not actually parameters related to the population, but parameters about the evolutionary function. This means the binary evolution functions can be constrained on an individual event basis. We "back propagate" the posterior in the observable space to the evolutionary parameter space, thus allowing us to study hyper-parameters and its correlation with progenitor parameters in a data-driven manner. Our method makes less assumption than the traditional forward modelling approach, which often fix the hyper-parameters across the entire population. Since we are not limited to the fix hyper-parameters assumption, we can explore the behavior of the hyper-parameters across the population much more efficiently. While our work lay down a data analysis pathway to understand the population of GW events, no physics can be learned without a comprehensive complimentary physical model. We hope this letter would motivate the community to build the next generation population synthesis codes that have the following properties: first, they should have explicit control over the random variables so marginalizing over random variables can be done more precisely; second, they should be as automatically differentiable as much as possible so exploring the evolutionary parameter space is efficient. Combining this work and the next generation population synthesis code, we can explore the full parameter space of binary evolution models with the next-generation GW detectors network in the near future.

6. ACKNOWLEDGEMENT

REFERENCES

- Ashton, G., Hübner, M., Lasky, P. D., et al. 2019, ApJS, 241, 27, doi: [10.3847/1538-4365/ab06fc](https://doi.org/10.3847/1538-4365/ab06fc)
- Romero-Shaw, I. M., Talbot, C., Biscoveanu, S., et al. 2020, MNRAS, 499, 3295, doi: [10.1093/mnras/staa2850](https://doi.org/10.1093/mnras/staa2850)
- The LIGO Scientific Collaboration, the Virgo Collaboration, the KAGRA Collaboration, et al. 2021, arXiv e-prints, arXiv:2111.03606. <https://arxiv.org/abs/2111.03606>
- Veitch, J., Raymond, V., Farr, B., et al. 2015, PhRvD, 91, 042003, doi: [10.1103/PhysRevD.91.042003](https://doi.org/10.1103/PhysRevD.91.042003)