

Using machine learning to autotune chi-squared tests for gravitational wave searches

Connor McIsaac^{1,*} and Ian Harry²

¹*DISCnet Centre for Doctoral Training, University of Portsmouth, Institute of Cosmology and Gravitation, Portsmouth PO1 3FX, United Kingdom*

²*University of Portsmouth, Institute of Cosmology and Gravitation, Portsmouth PO1 3FX, United Kingdom*



(Received 8 March 2022; accepted 2 May 2022; published 25 May 2022)

The sensitivity of gravitational wave searches is reduced by the presence of non-Gaussian noise in the detector data. These non-Gaussianities often match well with the template waveforms used in matched filter searches, and require signal-consistency tests to distinguish them from astrophysical signals. However, empirically tuning these tests for maximum efficacy is time consuming and limits the complexity of these tests. In this work we demonstrate a framework to use machine-learning techniques to automatically tune signal-consistency tests. We implement a new χ^2 signal-consistency test targeting the large population of noise found in searches for intermediate mass black hole binaries, training the new test using the framework set out in this paper. We find that this method effectively trains a complex model to downweight the noise, while leaving the signal population relatively unaffected. This improves the sensitivity of the search by $\sim 11\%$ for signals with masses $> 300 M_{\odot}$. In the future this framework could be used to implement new tests in any of the commonly used matched-filter search algorithms, further improving the sensitivity of our searches.

DOI: [10.1103/PhysRevD.105.104056](https://doi.org/10.1103/PhysRevD.105.104056)

I. INTRODUCTION

Since the first detection of gravitational waves in 2015 [1], the LIGO detectors have been upgraded multiple times, and the network of detectors now includes Advanced LIGO [2], Advanced Virgo [3] and Kagra [4]. The third observing run of the LIGO-Virgo-Kagra network [5] has been completed and a whole suite of new compact binary coalescence (CBC) signals have been observed [6–11]. Along with improvements in the detectors, search algorithms used to search for CBC signals have also been improved by using information from the full network and introducing new methods for removing noise [12–14].

A set of modeled [11–23] and unmodeled [24] search algorithms have been used to observe CBCs in the past. In this paper we will focus on modeled searches. Modeled searches construct a large bank of simulated signals (templates) with a variety of masses and spins in order to cover the targeted parameter space. These templates are then each used to perform a matched filter over the data for each detector. The matched filter signal-to-noise ratio (SNR) is then compared across the detector network in order to check for consistency across detectors.

The matched filter is the optimal solution when searching for known signals assuming that only Gaussian noise is

present [15]. However, in the presence of non-Gaussian noise transients, large SNRs can be produced where no signal is present [16]. Gravitational-wave detectors contain many such transients [25,26], commonly referred to as “glitches.” Glitches can produce huge SNR values while having little resemblance to the CBC signals being searched for. It is therefore necessary to employ signal-consistency tests to remove as many of these glitches as possible in order to reduce the rate of false alarms. In this paper we will explore how to effectively develop and tune such a signal-consistency test to separate the signal and noise populations. We will apply this methodology to the PyCBC search [12] as a demonstration of its use. However, we strongly emphasize that this method will be applicable to any modeled search.

In the PyCBC search, multiple signal-consistency tests are employed. First, the matched filter SNR is modified using two χ^2 tests that compare the morphology of the signal in the data with that of the template [27,28], penalizing any signals found to be inconsistent. Peaks in the reweighted SNR time series are then compared across the detector network, checking for consistency in the template parameters, as well as the relative time of arrival, amplitude and phase of the signals [29]. Each potential candidate event is then given a detection statistic in order to rank the likelihood that it is a real signal.

After these tests are performed, the remaining signals are shifted in time relative to each other to empirically measure

*connor.mcisaac@port.ac.uk

a nonastrophysical background. The detection statistic values of the background signals can then be compared to the observed coincident signals in order to produce a false alarm rate for each observed signal [17].

The existing signal consistency tests remove a large number of glitches from the single detector data, reducing the rate of background coincidences, and therefore reducing the false alarm rate of observed signals. However, a large number of glitches continue to be detected, particularly when searching with high mass templates [30], $M_{\text{total}} \gtrsim 100 M_{\odot}$, where the signal may only be in the detector frequency band for a fraction of a second.

In this work we propose a method of creating and tuning new signal-consistency tests in order to separate signal and noise populations. We show that we can use machine-learning techniques such as stochastic gradient descent to optimize these tests efficiently, and thus improve the sensitivity of our searches to gravitational wave signals.

The use of machine-learning in gravitational wave searches is an area where much work is being done. Several works have explored the use of neural networks to replace the matched filter statistic [31–35], using convolution neural networks to predict the probability of a signal being present.

One advantage of convolution neural networks compared to a matched filter is the computational cost involved. This is particularly important for multimessenger astronomy, where prompt detection of gravitational wave signals could enable a followup with electromagnetic observations. It has been shown that convolution neural networks could be an effective method for enabling such observations [36–40].

These works have shown that machine learning can, in some cases, compete with the sensitivity of a matched filter search when applied to a single detector. However, such methods have not yet been demonstrated to be effective in large-scale searches for CBCs covering a wide range of parameters. Current methods also do not produce additional information such as the amplitude and phase of the signal, used in the matched filter search to test triggers across detectors, they therefore lose sensitivity when compared to a matched filter search using a network of detectors [41].

In this work we choose to introduce a machine learning model within the current matched filter framework in order to utilize the statistical tests already available to us. We implement a new χ^2 test using stochastic gradient descent to train a set of tunable parameters within the model, optimizing the test using noise triggers from a previous search along with a set of simulated signals. By implementing the model as a χ^2 test it should remain rigorous in the case of unseen data, such as a new population of glitches.

We start by describing the general use of χ^2 tests within gravitational wave searches and the tests currently used within the PyCBC search in Sec. II. In Sec. III we describe

our proposed framework for training new χ^2 tests using machine-learning methods. We then utilize this framework in Sec. IV to train a new χ^2 test for use in a search for intermediate mass black hole (IMBHs). In Sec. V we present the effect of this trained model on the IMBH search showing that it improves the sensitivity of the search, particularly at high masses, where the effect of non-Gaussian noise is most prominent.

II. A REVIEW OF χ^2 TESTS IN GW SEARCHES

We will begin by reviewing existing χ^2 signal-consistency tests used in gravitational wave searches.

In order to search for signals in strain data, a matched filter is used. Assuming the strain data takes the form of $s(t) = n(t) + h(t)$, where $n(t)$ is stationary Gaussian noise and $h(t)$ is a known signal, matched filtering is the optimal method for detecting the signal $h(t)$. The calculated SNR is analytically maximized over the amplitude and phase of the signal. The SNR is calculated as

$$\rho^2 = \frac{|(s|h)|^2}{(h|h)}, \quad (1)$$

where the inner product is

$$(a|b) = 4 \int_{f_{\text{low}}}^{f_{\text{high}}} \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n(f)} df, \quad (2)$$

and $S_n(f)$ is the one-sided power spectral density (PSD) of the noise. However, in the case of non-Gaussian noise, large peaks in the SNR time series can be produced. Short bursts of non-Gaussian noise are often referred to as “glitches,” these can produce extremely large values of SNRs with little similarity to the signal being searched for. In order to remove such triggers signal-consistency tests may be introduced to test the morphology of the trigger compared to that of the search template.

A χ^2 test is one such test. A χ^2 test is constructed by performing a matched filter with additional templates, \hat{h}_{\perp} , that are orthogonal to the search template such that $(h|\hat{h}_{\perp}) = 0$. Given well-behaved noise, and a signal which is an exact match to the search template, the matched filter SNR of the orthogonal template will follow a reduced χ^2 distribution with 2 degrees of freedom [15]. However, when there is non-Gaussian noise present, such as a glitch, the SNR will deviate from this distribution, taking a larger value. By examining triggers on the SNR- χ^2 plane the signal and noise populations may then be separated.

After choosing a suitable template, \hat{h} , to be used for the χ^2 test one first normalizes it so that $(\hat{h}|\hat{h}) = 1$. The part of the signal orthogonal to the search template is then selected:

$$\hat{h}_\perp = \frac{\hat{h} - (\hat{h}|h)h}{\sqrt{1 - (\hat{h}|h)^2}}. \quad (3)$$

N such templates are created in this way and their SNRs are combined to produce the χ^2 statistic.

$$\chi_r^2 = \frac{1}{2N} \sum_i^N \rho_i^2. \quad (4)$$

In the case that the templates are orthogonal to one another this will produce a reduced χ^2 distribution with $2N$ degrees of freedom. However, in general, orthogonality between the χ^2 templates is not always enforced, in which case the statistic will follow a generalized χ^2 distribution with increased variance.

The χ^2 test also assumes that the signal in the data matches the search template. However, due to the discrete placement of templates within the parameter space of the search there will be some mismatch between these two signals. This mismatch means that the χ^2 template will no longer be orthogonal to the signal and some of the signal's SNR will be included in the χ^2 statistic, increasing the mean of the distribution, creating a noncentral χ^2 distribution. A similar effect will be caused if the PSD is miscalculated, or if the noise is nonstationary.

In general any N templates can be chosen to construct a χ^2 test. However, this test will be most effective when templates are chosen that have some overlap with known non-Gaussian noise in the data, in particular, aiming to target noise which produces high SNR triggers for the targeted parameter space.

To separate the signal and noise populations a reweighted SNR is then calculated that penalizes triggers where the χ_r^2 is larger than expected. This reweighting takes the general form

$$\hat{\rho} = f(\rho, \chi_r^2). \quad (5)$$

This reweighted SNR is then used to rank potential candidate events.

A. Existing χ^2 tests in the PyCBC search

There are currently two χ^2 signal-consistency tests employed within the PyCBC search. The first of these is the Allen χ^2 test [27]. This test divides the template into p independent frequency bins, splitting the template such that each bin will contribute equally to the SNR. The SNR contribution for each of these subtemplates is calculated and compared to the expected value, calculating the χ^2 statistic as

$$\chi_r^2 = \frac{p}{2p-2} \sum_{i=1}^p \left(\frac{\rho}{p} - \rho_{\text{bin},i} \right)^2. \quad (6)$$

This will take large values when a glitch is present in the data that does not share the same morphology as the search template. Specifically this test checks the distribution of power along the track of the CBC signal. Although this test does not follow the exact form described in the previous section, it follows the same principle detecting excess power along the track of the signal. The reweighted SNR [42] is then calculated as

$$\tilde{\rho} = \begin{cases} \rho, & \text{if } \chi_r^2 \leq 1 \\ \rho[(1 + (\chi_r^2)^3)/2]^{-\frac{1}{6}}, & \text{if } \chi_r^2 > 1 \end{cases}. \quad (7)$$

By ranking the candidates based on this reweighted SNR, a large number of noise triggers can be down-weighted. This test is particularly powerful for lower mass systems where the signals span a wide range of frequencies within the band of the detectors, allowing for a larger number of frequency bins to be used effectively. The number of frequency bins to be used is varied as a function of the search templates parameters [43]. The number of frequency bins to use and the form of the SNR reweighting are currently tuned empirically and have evolved over time [42–44].

The second χ^2 test is the sine-Gaussian χ^2 test [28]. This works by performing a matched filter with n sine-Gaussian signals, each being placed at frequencies higher than those expected from the search template. As these sine-Gaussian signals do not overlap with the search template, we can construct a χ^2 test as the sum of their SNRs

$$\chi_{r,sg}^2 = \frac{1}{2n} \sum_i^n \rho_{sg,i}^2. \quad (8)$$

This statistic tests if excess power is present above the final frequency of the search template. When excess power is present large values of $\chi_{r,sg}^2$ will be produced and the SNR is reweighted again

$$\tilde{\rho}_{sg} = \begin{cases} \tilde{\rho}, & \text{if } \chi_r^2 \leq 4 \\ \tilde{\rho}(\chi_{r,sg}^2/4)^{-\frac{1}{2}}, & \text{if } \chi_r^2 > 4 \end{cases}. \quad (9)$$

The addition of this second test further reduces the rate of noise triggers due to glitches. This test has a significant impact for higher mass templates where there is a population of short duration glitches known as ‘‘blips’’ [45]. A subset of these blips have power extending to high frequencies allowing this test to remove them successfully [28]. However a large number of these glitches are not removed by this test [30].

Both of these tests have been tuned empirically by hand, choosing the number of frequency bins to be used and the

placement of the sine-Gaussian signals, as well as the exact form of the reweighting of the SNR. In the next section we propose an approach that allows us to create and tune new χ^2 tests using a data-driven approach.

III. AUTOTUNING OF A χ^2 SIGNAL-CONSISTENCY TEST

We propose a framework in which we create new χ^2 tests and empirically tune them based on a set of training data. To achieve this we take a set of noise triggers from a previous search along with a set of simulated signals and use stochastic gradient descent to tune the parameters of our model.

In order to optimize the parameters of our chosen model we first must define a loss function: this is the quantity that we aim to minimize during the training process. The loss function is a function of the triggers SNR, ρ , and its SNR reweighted by the new χ^2 test described in Sec. IV, $\hat{\rho}$. For this work we choose to define a separate loss function for noise triggers and simulated signals. The loss functions used in the case of noise triggers is

$$L_n(\hat{\rho}, \rho) = \begin{cases} \hat{\rho} - 4, & \text{if } \hat{\rho} > 4 \\ 0, & \text{if } \hat{\rho} \leq 4 \end{cases}, \quad (10)$$

which penalizes any cases where the reweighted SNR is greater than 4. Below this threshold the PyCBC search currently discards all triggers, so it is unnecessary to reduce it any further.

The loss function used for simulated signals is

$$L_{\text{inj}}(\hat{\rho}, \rho) = \rho - \hat{\rho}. \quad (11)$$

This penalizes the case where the χ_r^2 value is large and the SNR is reduced. This contribution to the loss will also allow us to train a function that reweights the SNR as in Eq. (5) in order to create a greater separation between the signal and noise populations in the SNR- χ^2 plane.

To update the parameters of the model we must then calculate the gradients with respect to the loss function. This is done using backpropagation after calculating the loss function using a set of training data. The χ^2 model and matched filter are implemented in TensorFlow [46,47], allowing the gradients to be tracked through the calculation and the parameters updated.

Stochastic gradient descent has been used widely in the field of deep learning to optimize extremely complex models [48,49], this framework therefore allows us to produce highly complex transformations while allowing us to effectively tune them to the detector data at a reasonable computational cost.

In the next section we will describe one such model and the data used to train the model.

IV. A χ^2 TEST FOR INTERMEDIATE MASS BLACK HOLE SEARCHES

To demonstrate the training scheme described in the previous section we will attempt to train a χ^2 test that improves the separation between glitches and signals when searching for IMBHs, which we consider as signals with $M_{\text{tot}} > 100 M_{\odot}$. In this mass range the Allen χ^2 test described in the Sec. II A has a limited effect due to the systems merging at low frequencies and covering a relatively small frequency range within the detector bandwidth. The sine-Gaussian χ^2 test is successful in removing a population of glitches that affect templates within this mass range, however, many glitches remain that do not have significant high frequency power [30].

In this section we will define a transformation using the framework outlined in the previous sections using data from a previous IMBH search in order to train the χ^2 test and improve the sensitivity of the search.

A. Creating χ^2 templates

Existing χ^2 tests effectively test the distribution of power of a candidate event along the track of the signal in the time-frequency plane and test for excess power at high frequencies. We aim to test for excess power in a frequency band similar to those of the search templates, aiming to cover areas of the time-frequency plane not currently covered by existing tests.

To achieve this we transform the search template itself, shifting the template in time and frequency. The optimal values for these time and frequency shifts will depend on the template being used and the noise present in the data. Time shifted templates have previously been used in the PyGRB search [50,51] to create a χ^2 signal-consistency test using fixed values for the time shifts. We propose a model that allows different time and frequency shifts for each template, tuning these values based on the current data.

The time and frequency shifts are selected using a dense neural network. The template is first turned into an input by sampling it between 12 and 256 Hz with a sample width of 0.1 Hz, we then take the absolute value of the template and normalize it so that the mean of the input is one. This input is then passed to a dense neural network with two output values between -1 and 1 . The first of these values is used as the time shift after being scaled by the maximum allowed time shift Δt_{max} , similarly, the second value is used as the frequency shift after being scaled by the maximum allowed frequency shift Δf_{max} .

The dense neural network consists of 11 dense layers, using the rectified linear unit (RELU) activation functions for hidden layers and the hyperbolic tangent function for the output layer. In order to produce multiple time/frequency shift pairs we train multiple networks with this configuration. However, to speed up the training of this model the first six dense layers are shared between each network. The sizes of the dense layers are listed in Table I.

TABLE I. This table details the architecture of the neural network used to calculate time and frequency shifts. The input is the absolute value of the search template sampled at 0.1 Hz between 12 and 256 Hz. Dense layers transform their input using a matrix multiplication followed by the addition of a bias vector, the output is then passed to the activation function listed. The hyperbolic tangent activation function of the final layer ensures the output is in the range $[-1, 1]$. The two outputs are multiplied by Δt_{\max} and Δf_{\max} , respectively, to generate the time and frequency shifts. Four such networks are generated, the layers marked with an asterisk (*) share their weights between the four networks.

Layer	Output size
Input	2440
Dense + RELU *	128
Dense + RELU *	128
Dense + RELU *	64
Dense + RELU *	64
Dense + RELU *	32
Dense + RELU *	32
Dense + RELU	16
Dense + RELU	16
Dense + RELU	8
Dense + RELU	8
Dense + tanh	2

After calculating these shifts they are applied to the template before using Eq. (3) to generate our χ^2 templates.

In addition to this model we must define a function to reweight the SNR with the χ_r^2 value. We create a parametrized model that can reproduce the reweighting in Eqs. (7) and (9).

$$\hat{\rho} = \begin{cases} \rho, & \text{if } \chi_r^2 \leq \sigma \\ \rho((\delta + (\chi_r^2/\sigma)^\beta)/(\delta + 1))^{-\frac{1}{\alpha}}, & \text{if } \chi_r^2 > \sigma \end{cases} \quad (12)$$

Here σ , α , β , and δ are parameters that can also be tuned to increase the effectiveness of the test. This reweighting leaves any signals with χ_r^2 less than the threshold, σ , unchanged. At large χ_r^2 values α and β determine how quickly the reweighted SNR decreases with the χ_r^2 value, while β and δ affect the transition between these two regimes.

B. Data

In order to most effectively train the model we target glitches that are missed by previous signal-consistency tests. We achieve this by performing a search using the setup described in [30] covering 45 days of data from the first half of the third observing run. The data used in this search is available from GWOSC [52,53]. From this search we then select a set of noise triggers with $\tilde{\rho}_{sg} > 6$ and $6 \leq \rho \leq 64$. These noise triggers may have been downweighted by existing signal-consistency tests, but have not

been downweighted enough to remove them from the analysis. In order to avoid including real signals we remove any triggers that are coincident across multiple detectors. This will also remove a number of noise triggers that have formed coincident triggers, however, enough triggers remain to create a substantial training set. These triggers are then clustered over a window of 1 second and the triggers with the largest $\tilde{\rho}_{sg}$ in that window are kept. We record the times of the triggers and the parameters of the template that produced them.

We also select a set of simulated signals to include during training. From the list of simulated signals that were recovered by the search, with false-alarm rates smaller than 1 per year, we select a set using the same constraints as the noise triggers. For these triggers we record the parameters of the simulated signal, as well as the template that recovered the signal. By using the template that recovered the signal in the search we are including the effect of template mismatch within the training scheme, this allows the SNR reweighting step in Eq. (12) to be tuned to account for this contribution. The simulated signals used in this analysis include effects from precession and higher-order modes that are not present in the template bank, by including these effects in the analysis we can train the model to avoid identifying these effects as noise, maintaining our sensitivity to these signals.

For each sample, the strain is loaded at 2048 Hz, for samples containing simulated signals the signal is then added. The strain data is high-pass filtered at 12 Hz and PSDs are calculated using 512 seconds of data around the time of the sample, following the same procedure as the PyCBC search. 64 seconds of data around the trigger is then selected, ensuring that the trigger is not within the first 32 seconds or the last 16 seconds to allow time for the inspiral and ringdown of the search templates.

The search templates are generated using the SEOBNRv4_ROM [54,55] waveform model, and are generated at the same length and sample frequency as the strain data. The simulated signals are generated using the NRSUR7DQ4 [56], SEOBNRv4 [54,55] and SEOBNRv4HM [57,58] waveform models.

In order to ensure the noise and signal samples have similar importance during training we select an equal number of each. Additionally, to ensure that the parameters in Eq. (12) are trained to separate the noise and signal populations across a range of SNRs we bin our samples by SNR and select an equal number of samples from the noise and signal models for each bin. The boundaries of these SNR bins are 6, 8, 10, 14, 18, 24, and 64. In each of these SNR bins we draw 1200 noise samples and signal samples from those remaining after filtering has been applied.

We set aside 10% of all samples to be used as a validation set to monitor the performance of the model on unseen data, all other samples are used to train the model. This gives us a total of 12960 samples for training and 1440 samples for validation.

C. Training

The training of the model is performed in batches, each batch contains 32 samples from the training set described in the previous section. For each sample the peak SNR within 0.1 seconds of the trigger time is calculated. The search template is then generated and passed to the transformation described in Sec. IV A. The resulting templates are then normalized and the orthogonal templates calculated using Eq. (3). These templates are then used to calculate χ_r^2 at the time of the peak SNR and the reweighted SNR is calculated. This value can then be passed to the loss function in order to calculate the loss values for the batch.

Once the losses have been calculated, backpropagation is used to obtain the gradients of the loss function with respect to the trainable weights of the network described in Sec. IV A, as well as the trainable parameters in Eq. (12). Stochastic gradient descent is then used to apply small changes to the variables based on the calculated gradients. In order to speed up training and improve performance for sparse gradients we use Nesterov momentum [59] when calculating the parameter updates.

Before training the full model we perform a pretraining phase where only the parameters of the SNR reweighting in Eq. (12) are trained. This step is faster than the training of

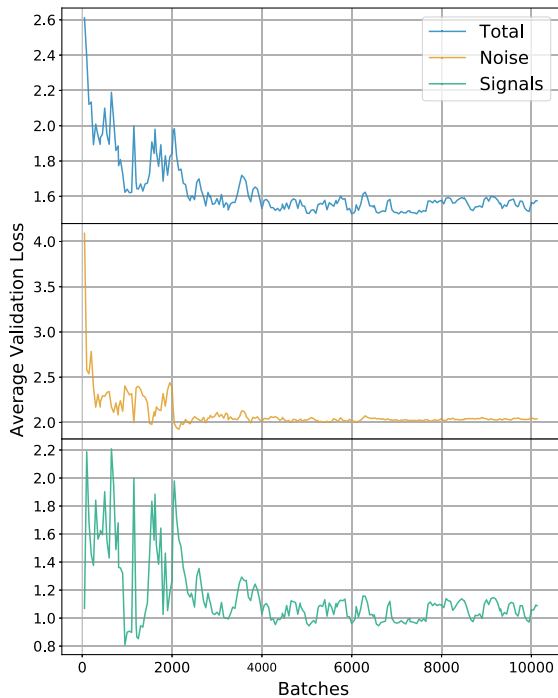


FIG. 1. The average loss calculated using the unseen test data as it changes with the number of training batches used. The average losses contributed by noise samples and signal samples are also plotted. This shows a large improvement in the loss contributed by noise samples, while the loss from signals remains reasonably steady.

the full model and by performing this step first the training of the χ^2 model is more effective early in the training. After this step we proceed with the main training step, training the parameters of the model described in Sec. IV A and the SNR reweighting in Eq. (12) at the same time.

The main training step is repeated until all samples in the training set have been analyzed 25 times, taking a total of ~ 24 hours using 8 CPU cores. During training the learning rate determines how quickly parameters are changed in response to the calculated gradients. In order to improve convergence late in the training stage we employ learning rate decay. After each full cycle of the training set the learning rate is multiplied by a factor of 0.9, allowing the model to make smaller adjustments late in training.

D. Trained model

As shown in Fig. 1 we see that the loss calculated using the test set decreases as training continues. This change is mainly driven by the model downweighting noise triggers, while the contribution from signals makes a smaller change over the course of the training. The effect of this training can also be seen in Fig. 2, and we can see that as training continues a larger fraction of the SNR is removed as noise triggers are targeted more effectively by the model, while signals are left relatively unaffected.

In Fig. 3 we can see that the noise and signal populations are well separated in the SNR- χ^2 plane, particularly at high SNRs. The parameters in Eq. (12) have been trained such that the majority of signal samples are below the threshold, σ , while noise samples above the threshold are heavily downweighted.

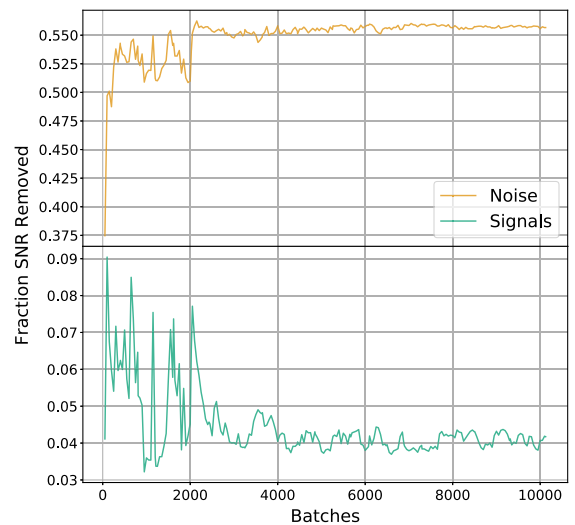


FIG. 2. The average fraction of the SNR removed calculated using the unseen test data as it changes with the number of training batches used, split by noise and signal samples.

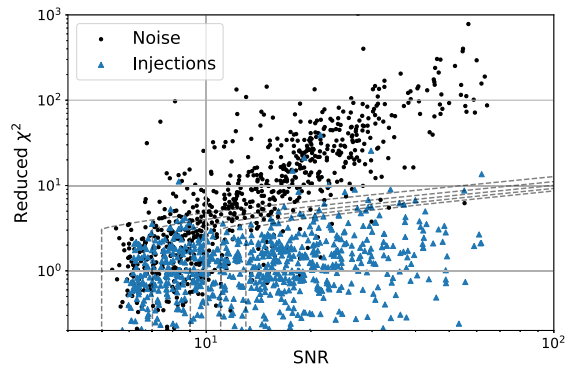


FIG. 3. The test samples plotted in the SNR- χ^2 plane after training is complete. Lines of constant reweighted SNR are plotted. We can see that at high SNR values there is good separation between the noise samples (black dots) and signal samples (blue triangles), allowing the model to downweight noise triggers heavily.

V. EFFECT ON AN INTERMEDIATE MASS BLACK HOLE SEARCH

In this section we will show the effect of introducing this model to a search for IMBHs. For this test we carry out a search following the configuration set out in [30], covering eight days of data from the first half of the third observing run. We run the search twice, with the only change being the introduction of the model trained in the previous section. To ensure that the model generalizes to new data we run this search on a stretch of data completely separate to that used during training.

By introducing this model, noise triggers can be effectively downweighted. Figure 4 shows the change in the number of triggers found when using the new ranking statistic. This reduction in triggers will reduce the number

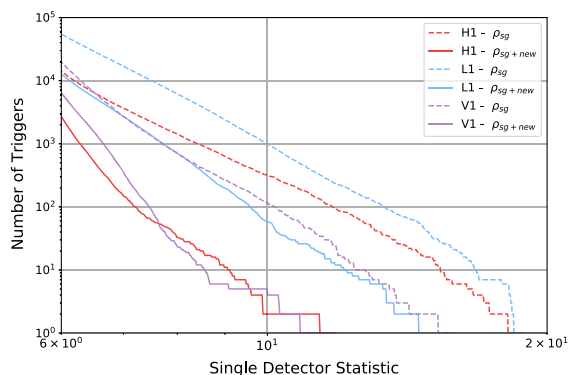


FIG. 4. The cumulative number of single-detector triggers with ranking statistics below a given value. We can see a reduction in the number of single-detector triggers for all three detectors when changing from the previous ranking statistic (dashed line) to the new ranking statistic (solid line) including the new tuned model.

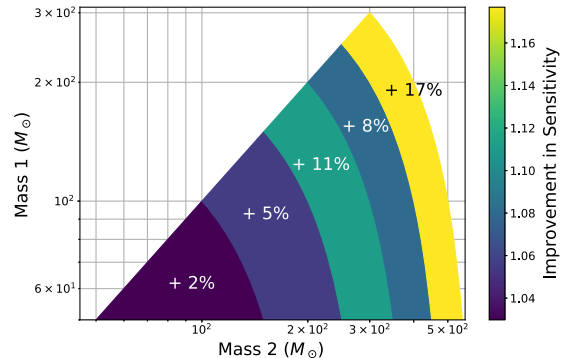


FIG. 5. The ratio of the sensitive volume time for the search including the trained model to the search without, calculated using simulated signals added to the data with a detection threshold on false-alarm rate of 1 per year.

of coincident noise triggers in the foreground and the empirically measured background. It is this decrease in the rate of background triggers that produces an increase in the significance of remaining foreground triggers, thereby improving the sensitivity of the search.

We evaluate the sensitivity of the search using a number of simulated signals added to the data and analyzed in the same way as the main search. These simulated signals follow the same distribution as those in [30]. The SEOBNRv4 and SEOBNRv4HM waveform models are used to generate aligned spin signals, with total masses in the range $[100, 600] M_{\odot}$ and mass ratios in the range $[1, 10]$. Precessing signals are generated using the NRSUR7DQ4 waveform model with total masses in the range $[100, 600] M_{\odot}$, mass ratios in the range $[1, 4]$ and component spins isotropically distributed. For all simulated signals the distance is drawn uniformly in the range $[0.5, 11]$ Gpc, with isotropic sky positions and binary orientations. The sensitive volume of the search is then calculated by applying a threshold to the calculated false-alarm rate of 1 per year and measuring the detection efficiency in a number of distance bins. The detection efficiencies are then multiplied by the volume enclosed in the distance bins and the volumes summed to find the total sensitive volume of the search.

In Fig. 5 we see that the sensitivity of the search has been increased by the addition of the new χ^2 test, with an increase in sensitivity of $\sim 4\%$ for signals with total masses in the range $[100, 300] M_{\odot}$, increasing to $\sim 11\%$ for signals with total masses in the range $[300, 600] M_{\odot}$. This is due to the higher rate of glitches matching high mass templates, any decrease in the glitch population will therefore have a larger effect for these templates.

VI. CONCLUSION AND OUTLOOK

In this work we have demonstrated a new framework to automatically train complex new χ^2 signal-consistency tests

within modeled searches for gravitational wave signals. We have applied this to the example of a search for IMBH signals, where glitches have a strong effect on the sensitivity of the search. Our framework is able to train a new χ^2 model, which provides an improved separation of the signal and noise populations allowing the noise triggers to be downweighted. Using this new χ^2 test improves the sensitivity of the search to real signals by $\sim 4\%$ for signals with total masses in the range $[100, 300] M_\odot$ and $\sim 11\%$ for signals with total masses in the range $[300, 600] M_\odot$.

The introduction of new χ^2 tests is difficult and usually requires empirical tuning by hand to be effective, and often requires retuning for different target parameter spaces or noise populations. As signal-consistency tests become more complex this can become unfeasible. However, by utilizing machine-learning techniques we have shown that we can tune these automatically, removing the burden in improving and optimally tuning these tests. The method we demonstrate here could be applied to any of the commonly used matched-filter search pipelines targeting compact binary mergers.

The population of glitches within the interferometer data continues to be one of the largest challenges facing gravitational wave searches. By continuing to develop signal-consistency tests that specifically target such noise we can continue to improve the sensitivity of searches and increase the chance of observing new events in areas of the parameter space most affected by glitches.

The trained parameters as described in Sec. IV A are available as a supplementary data file at [60].

ACKNOWLEDGMENTS

C. M. was supported by the Science and Technology Facilities Council through the DISCnet Centre for Doctoral Training. I. H. was supported by STFC Grants No. ST/T000333/1 and No. ST/V005715/1. This research has made use of data or software obtained from the Gravitational Wave Open Science Center [61], a service of LIGO Laboratory, the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal and Spain. The construction and operation of KAGRA are funded by Ministry of Education, Culture, Sports, Science and Technology (MEXT), and Japan Society for the Promotion of Science (JSPS), National Research Foundation (NRF) and Ministry of Science and ICT (MSIT) in Korea, Academia Sinica (AS) and the Ministry of Science and Technology (MoST) in Taiwan. For the purpose of open access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising.

-
- [1] B. P. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), Observation of Gravitational Waves from a Binary Black Hole Merger, *Phys. Rev. Lett.* **116**, 061102 (2016).
 - [2] J. Aasi *et al.* (LIGO Scientific Collaboration), Advanced LIGO, *Classical Quantum Gravity* **32**, 074001 (2015).
 - [3] F. Acernese *et al.* (VIRGO Collaboration), Advanced Virgo: A second-generation interferometric gravitational wave detector, *Classical Quantum Gravity* **32**, 024001 (2015).
 - [4] T. Akutsu *et al.* (KAGRA Collaboration), Overview of KAGRA: Detector design and construction history, *Prog. Theor. Exp. Phys.* **2021**, 05A101 (2021).
 - [5] B. P. Abbott *et al.* (KAGRA, LIGO Scientific, Virgo, VIRGO Collaborations), Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA, *Living Rev. Relativity* **21**, 3 (2018).
 - [6] R. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run, *Phys. Rev. X* **11**, 021053 (2021).
 - [7] R. Abbott *et al.* (LIGO Scientific, VIRGO Collaborations), GWTC-2.1: Deep extended catalog of compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run, [arXiv:2108.01045](https://arxiv.org/abs/2108.01045).
 - [8] R. Abbott *et al.* (LIGO Scientific, VIRGO, KAGRA Collaborations), GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, [arXiv:2111.03606](https://arxiv.org/abs/2111.03606).
 - [9] A. H. Nitz, C. D. Capano, S. Kumar, Y.-F. Wang, S. Kastha, M. Schäfer, R. Dhurkunde, and M. Cabero, 3-OGC: Catalog of gravitational waves from compact-binary mergers, *Astrophys. J.* **922**, 76 (2021).
 - [10] A. H. Nitz, S. Kumar, Y.-F. Wang, S. Kastha, S. Wu, M. Schäfer, R. Dhurkunde, and C. D. Capano, 4-OGC: Catalog of gravitational waves from compact-binary mergers (2021), [10.48550/arXiv.2112.06878](https://arxiv.org/abs/10.48550/arXiv.2112.06878).
 - [11] S. Olsen, T. Venumadhav, J. Mushkin, J. Roulet, B. Zackay, and M. Zaldarriaga, New binary black hole mergers in the LIGO–Virgo O3a data, [arXiv:2201.02252](https://arxiv.org/abs/2201.02252).

- [12] G. S. Davies, T. Dent, M. Tápai, I. Harry, C. McIsaac, and A. H. Nitz, Extending the PyCBC search for gravitational waves from compact binary mergers to a global network, *Phys. Rev. D* **102**, 022004 (2020).
- [13] S. Sachdev *et al.*, The GstLAL search analysis methods for compact binary mergers in Advanced LIGO's second and Advanced Virgo's first observing runs, [arXiv:1901.08580](https://arxiv.org/abs/1901.08580).
- [14] T. Adams, D. Buskulic, V. Germain, G. M. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era, *Classical Quantum Gravity* **33**, 175012 (2016).
- [15] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries, *Phys. Rev. D* **85**, 122006 (2012).
- [16] S. Babak *et al.*, Searching for gravitational waves from binary coalescence, *Phys. Rev. D* **87**, 024033 (2013).
- [17] S. A. Usman *et al.*, The PyCBC search for gravitational waves from compact binary coalescence, *Classical Quantum Gravity* **33**, 215004 (2016).
- [18] K. Cannon *et al.*, Toward early-warning detection of gravitational waves from compact binary coalescence, *Astrophys. J.* **748**, 136 (2012).
- [19] C. Messick *et al.*, Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data, *Phys. Rev. D* **95**, 042001 (2017).
- [20] F. Aubin *et al.*, The MBTA pipeline for detecting compact binary coalescences in the third LIGO–Virgo observing run, *Classical Quantum Gravity* **38**, 095004 (2021).
- [21] Q. Chu, Low-latency detection and localization of gravitational waves from compact binary coalescences, Ph.D. thesis, Western Australia University, 2017.10.4225/23/5987feb0a789c
- [22] T. Venumadhav, B. Zackay, J. Roulet, L. Dai, and M. Zaldarriaga, New search pipeline for compact binary mergers: Results for binary black holes in the first observing run of Advanced LIGO, *Phys. Rev. D* **100**, 023011 (2019).
- [23] T. Venumadhav, B. Zackay, J. Roulet, L. Dai, and M. Zaldarriaga, New binary black hole mergers in the second observing run of Advanced LIGO and Advanced Virgo, *Phys. Rev. D* **101**, 083030 (2020).
- [24] S. Klimenko, G. Vedovato, M. Drago, F. Salemi, V. Tiwari, G. A. Prodi, C. Lazzaro, K. Ackley, S. Tiwari, C. F. Da Silva, and G. Mitselmakher, Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors, *Phys. Rev. D* **93**, 042004 (2016).
- [25] L. K. Nuttall, Characterizing transient noise in the LIGO detectors, *Phil. Trans. R. Soc. A* **376**, 20170286 (2018).
- [26] D. Davis *et al.* (LIGO Collaboration), LIGO detector characterization in the second and third observing runs, *Classical Quantum Gravity* **38**, 135014 (2021).
- [27] B. Allen, χ^2 time-frequency discriminator for gravitational wave detection, *Phys. Rev. D* **71**, 062001 (2005).
- [28] A. H. Nitz, Distinguishing short duration noise transients in LIGO data to improve the PyCBC search for gravitational waves from high mass binary black hole mergers, *Classical Quantum Gravity* **35**, 035016 (2018).
- [29] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, Detecting binary compact-object mergers with gravitational waves: Understanding and Improving the sensitivity of the PyCBC search, *Astrophys. J.* **849**, 118 (2017).
- [30] K. Chandra, V. Villa-Ortega, T. Dent, C. McIsaac, A. Pai, I. W. Harry, G. S. Cabourn Davies, and K. Soni, An optimized PyCBC search for gravitational waves from intermediate-mass black hole mergers, *Phys. Rev. D* **104**, 042004 (2021).
- [31] D. George and E. A. Huerta, Deep neural networks to enable real-time multimessenger astrophysics, *Phys. Rev. D* **97**, 044039 (2018).
- [32] D. George and E. A. Huerta, Deep learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data, *Phys. Lett. B* **778**, 64 (2018).
- [33] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Matching Matched Filtering with Deep Networks for Gravitational-Wave Astronomy, *Phys. Rev. Lett.* **120**, 141103 (2018).
- [34] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Convolutional neural networks: a magic bullet for gravitational-wave detection?, *Phys. Rev. D* **100**, 063015 (2019).
- [35] J. Yan, M. Avagyan, R. E. Colgan, D. Veske, I. Bartos, J. Wright, Z. Márka, and S. Márka, Generalized approach to matched filtering using neural networks, *Phys. Rev. D* **105**, 043006 (2022).
- [36] W. Wei and E. A. Huerta, Deep learning for gravitational wave forecasting of neutron star mergers, *Phys. Lett. B* **816**, 136185 (2021).
- [37] W. Wei, E. A. Huerta, M. Yun, N. Loutrel, M. A. Shaikh, P. Kumar, R. Haas, and V. Kindratenko, Deep learning with quantized neural networks for gravitational-wave forecasting of eccentric compact binary coalescence, *Astrophys. J.* **919**, 82 (2021).
- [38] W. Wei, A. Khan, E. A. Huerta, X. Huang, and M. Tian, Deep learning ensemble for real-time gravitational wave detection of spinning binary black hole mergers, *Phys. Lett. B* **812**, 136029 (2021).
- [39] M. B. Schäfer, F. Ohme, and A. H. Nitz, Detection of gravitational-wave signals from binary neutron star mergers using machine learning, *Phys. Rev. D* **102**, 063015 (2020).
- [40] P. G. Krastev, Real-time detection of gravitational waves from binary neutron stars using artificial neural networks, *Phys. Lett. B* **803**, 135330 (2020).
- [41] M. B. Schäfer and A. H. Nitz, From one to many: A deep learning coincident gravitational-wave search, *Phys. Rev. D* **105**, 043003 (2022).
- [42] J. Abadie *et al.* (LIGO Scientific, VIRGO Collaborations), Search for gravitational waves from low mass compact binary coalescence in LIGO's sixth science run and Virgo's science runs 2 and 3, *Phys. Rev. D* **85**, 082002 (2012).
- [43] B. P. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), GW150914: First results from the search for binary black hole coalescence with Advanced LIGO, *Phys. Rev. D* **93**, 122003 (2016).
- [44] B. P. Abbott *et al.* (LIGO Scientific Collaboration), Search for gravitational waves from low mass binary coalescences in the first year of LIGO's S5 data, *Phys. Rev. D* **79**, 122001 (2009).
- [45] M. Cabero *et al.*, Blip glitches in Advanced LIGO data, *Classical Quantum Gravity* **36**, 155010 (2019).

- [46] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from [tensorflow.org](https://www.tensorflow.org).
- [47] T. Developers, TensorFlow (2021), specific TensorFlow versions can be found in the “Versions” list on the right side of this page. See the full list of authors on GitHub, [10.5281/zenodo.5177374](https://zenodo.org/record/5177374).
- [48] S. Ruder, An overview of gradient descent optimization algorithms, [arXiv:1609.04747](https://arxiv.org/abs/1609.04747).
- [49] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature (London)* **521**, 436 (2015).
- [50] I. W. Harry and S. Fairhurst, A targeted coherent search for gravitational waves from compact binary coalescences, *Phys. Rev. D* **83**, 084002 (2011).
- [51] A. R. Williamson, C. Biwer, S. Fairhurst, I. W. Harry, E. Macdonald, D. Macleod, and V. Predoi, Improved methods for detecting gravitational waves associated with short gamma-ray bursts, *Phys. Rev. D* **90**, 122004 (2014).
- [52] M. Vallisneri, J. Kanner, R. Williams, A. Weinstein, and B. Stephens, The LIGO open science center, *J. Phys. Conf. Ser.* **610**, 012021 (2015).
- [53] R. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo, *SoftwareX* **13**, 100658 (2021).
- [54] A. Taracchini, A. Buonanno, Y. Pan, T. Hinderer, M. Boyle, D. A. Hemberger, L. E. Kidder, G. Lovelace, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, B. Szilágyi, N. W. Taylor, and A. Zenginoglu, Effective-one-body model for black-hole binaries with generic mass ratios and spins, *Phys. Rev. D* **89**, 061502(R) (2014).
- [55] A. Bohé *et al.*, Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors, *Phys. Rev. D* **95**, 044028 (2017).
- [56] V. Varma, S. E. Field, M. A. Scheel, J. Blackman, D. Gerosa, L. C. Stein, L. E. Kidder, and H. P. Pfeiffer, Surrogate models for precessing binary black hole simulations with unequal masses, *Phys. Rev. Research* **1**, 033015 (2019).
- [57] R. Cotesta, A. Buonanno, A. Bohé, A. Taracchini, I. Hinder, and S. Ossokine, Enriching the symphony of gravitational waves from binary black holes by tuning higher harmonics, *Phys. Rev. D* **98**, 084028 (2018).
- [58] R. Cotesta, S. Marsat, and M. Pürrer, Frequency domain reduced order model of aligned-spin effective-one-body waveforms with higher-order modes, *Phys. Rev. D* **101**, 124040 (2020).
- [59] Y. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Proc. USSR Acad. Sci.* **269**, 543 (1983).
- [60] C. McIsaac and I. Harry (2022), <https://icg-gravwaves.github.io/chisqnet/>.
- [61] gw-openscience.org.