

# Gravitational-wave searches for cosmic string cusps in Einstein Telescope data using deep learning

Quirijn Meijer<sup>1,2,\*</sup>, Melissa Lopez<sup>1,2</sup>, Daichi Tsuna<sup>3,4</sup> and Sarah Caudill<sup>5,6</sup>

<sup>1</sup>*Institute for Gravitational and Subatomic Physics (GRASP), Department of Physics, Utrecht University, Princetonplein 1, 3584CC Utrecht, The Netherlands*


<sup>2</sup>*Nikhef, Science Park 105, 1098XG Amsterdam, The Netherlands*

<sup>3</sup>*TAPIR, Mailcode 350-17, California Institute of Technology, Pasadena, California 91125, USA*

<sup>4</sup>*Research Center for the Early Universe (RESCEU), School of Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*

<sup>5</sup>*Department of Physics, University of Massachusetts, Dartmouth, Massachusetts 02747, USA*

<sup>6</sup>*Center for Scientific Computing and Data Science Research, University of Massachusetts, Dartmouth, Massachusetts 02747, USA*

 (Received 25 August 2023; revised 29 November 2023; accepted 8 December 2023; published 18 January 2024)

Gravitational-wave searches for cosmic strings are currently hindered by the presence of detector glitches, some classes of which strongly resemble cosmic string signals. This confusion greatly reduces the efficiency of searches. A deep-learning model is proposed for the task of distinguishing between gravitational-wave signals from cosmic string cusps and simulated blip glitches in design sensitivity data from the future Einstein Telescope. The model is an ensemble consisting of three convolutional neural networks, achieving an accuracy of 79%, a true positive rate of 76%, and a false positive rate of 18%. This marks the first time convolutional neural networks have been trained on a realistic population of Einstein Telescope glitches. On a dataset consisting of signals and glitches, the model is shown to outperform matched filtering, specifically being better at rejecting glitches. The behaviour of the model is interpreted through the application of several methods, including a novel technique called waveform surgery, used to quantify the importance of waveform sections to a classification model. In addition, a method to visualize convolutional neural network activations for one-dimensional time series is proposed and used. These analyses help further the understanding of the morphological differences between cosmic string cusp signals and blip glitches. Because of its classification speed in the order of magnitude of milliseconds, the deep-learning model is suitable for future use as part of a real-time detection pipeline. The deep-learning model is transverse and can therefore potentially be applied to other transient searches.

DOI: [10.1103/PhysRevD.109.022006](https://doi.org/10.1103/PhysRevD.109.022006)

## I. INTRODUCTION

Since the first confirmed detection of the gravitational-wave signal GW150914 in 2015 [1], over 90 gravitational waves have been confirmed by the LIGO, Virgo, and KAGRA detectors [2–5]. These observatories are currently in their second generation [6,7]. A third generation of detectors including Cosmic Explorer [8], the Laser Interferometer Space Antenna (LISA) [9] and the Einstein Telescope [10] are already in development. The Einstein Telescope will have a greatly increased sensitivity compared to the current generation and is expected to detect many more signals, possibly from new sources. Gravitational waves observed thus far have been the product of compact binary coalescences, which are pairs of coalescing stellar- or intermediate-mass black holes and

neutron stars [2–5]. Searches, however, are not limited to such systems. One class of unary sources is that of cosmic strings.

Cosmic strings are objects that are conjectured by several theories to have formed in the early Universe, and if present, have evolved as the Universe expanded [11,12]. They should present themselves as strings at cosmological scales. Cosmic strings interact with gravity through gravitational lensing on background light sources due to their angular deficit [11], but also through gravitational waves. The focus of this paper is the detection of cusps on cosmic strings [13–15]. Cusps can be understood as points on the cosmic string that instantaneously accelerate to the speed of light, and in doing so generate gravitational waves.

Current searches for cosmic string signatures, of which cusp signals are an example, rely on matched filtering [16–21]. Matched filtering is a process where modelled waveforms (called templates) are convolved with detector

\*Corresponding author: [r.h.a.j.meijer@uu.nl](mailto:r.h.a.j.meijer@uu.nl)

strain data in order to check for the presence of a signal matching the template. Although these searches have not resulted in observational evidence for the existence of cosmic strings, their results have been used to constrain the model parameters of cosmic strings [16,19,21]. Matched filter searches for cosmic strings are hindered by the presence of detector glitches [17,21], bursts of non-Gaussian noise that may look very similar to modelled cusp signals. Although it is uncertain what glitches will look like in the Einstein Telescope, short-duration glitches that mimic cusp signals are likely to appear. In this paper, machine learning is employed to demonstrate it is possible to differentiate cosmic string cusps from a common class of transient glitches known as blip glitches in LIGO and Virgo data [22], assuming similar glitches in Einstein Telescope data.

This paper details the training of convolutional neural networks for the task of distinguishing modeled cosmic string cusp signals from artificial blip glitches in simulated Einstein Telescope data. The goal is to both prepare for the arrival of the third generation of detectors, as well as to utilize the higher sensitivity of these detectors to learn about the morphological differences between the two types when obfuscated by detector noise. Having this information may aid in current searches in second-generation data as well, as it can be incorporated to design better searches and confirmation tests for observed gravitational-wave candidates.

This paper is organized as follows. Section II reviews cosmic strings before drawing the comparison to glitches through their waveform similarity. Section III reviews matched filtering, the current method for cosmic string searches. Section IV details the methodology of this paper, from the creation of the dataset to the analysis of the model. Section V reports on the results of the applied methods. Conclusions are collected in Sec. VI.

## II. COSMIC STRINGS AND GLITCHES

Cosmic strings are found in field theories, where they appear as one-dimensional topological defects [11,12]. Such defects may arise as the result of a process called spontaneous symmetry breaking, where the internal symmetry group of the vacuum manifold  $M$  is lowered to a strict subgroup [23]. Although both global and local symmetries can be broken the restriction to local symmetry breaking is made, due to the possible relation with unification [24]. It is for this reason that cosmic strings originating from symmetry breaking in local symmetry groups, or gauge groups [25], are studied in this paper. Assuming the presence of a Lie group structure leads to the gauge group being a manifold, and in particular to the gauge group admitting a topology. It is through the homotopy groups [26] of the gauge group that topological defects can be detected and classified. In particular, the

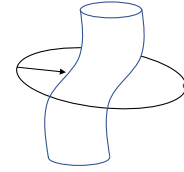


FIG. 1. A contraction of the circle  $S^1$  to the basepoint on the right, caught on a stringlike deficiency. This deficiency, or defect, does not allow the contraction to complete.

fundamental group  $\pi_1(M)$  being nontrivial leads to the conclusion that effectively one-dimensional (or stringlike) topological defects must be present in the theory, as the contractions of the  $S^1$  embeddings get caught on such presences (illustrated in Fig. 1). As the circle shrinks, the defect prohibits the circle from collapsing onto the basepoint. Different defects are then signaled by the classes in the fundamental group. More generally, nontriviality of the  $k$ th homotopy group demonstrates the presence of topological defects of dimension  $k$ . Although cosmic strings remain hypothetical as of yet, the detection of topological defects in other dimensions for other systems gives reason to assume they may exist. Domain walls, two-dimensional topological defects, appear when a ferromagnetic material undergoes a phase transition as its temperature passes the Curie point [27].

Alternatively, a class of cosmic strings arises from string theory. In string theory, strings are small elemental objects that vibrate in dimensions beyond the four spacetime dimensions postulated by general relativity [28]. As these additional dimensions are compactified (for instance through the Kaluza-Klein mechanism [29]), this takes place at unobservably small scales, meaning it is extremely difficult to obtain observational evidence. However, it is possible for these strings to grow to a cosmological scale, forming so-called cosmic superstrings that exhibit behavior similar to cosmic strings [24,30].

Spontaneous symmetry breaking [25], and therefore the appearance of cosmic strings, may be caused by phase transitions such as the ones associated with grand unification or lower-energy scales. Cosmic strings are therefore of interest to the scientific community as their study can unveil information about both the early Universe and a string-theoretical description of the Universe [24].

As physical phenomena, cosmic strings appear at cosmological scale as extremely thin strings with massive densities. As such, their large-scale dynamics are governed by the zero-thickness limit by the Nambu-Goto action [24]. Cosmic strings can either be open strings or closed loops and moreover may interact if two cosmic strings meet. Networks of multiple interacting cosmic strings have been simulated [31–35].

In order to detect cosmic strings, observational signatures are needed. Cosmic strings are massive dynamic objects, producing gravitational waves through a variety

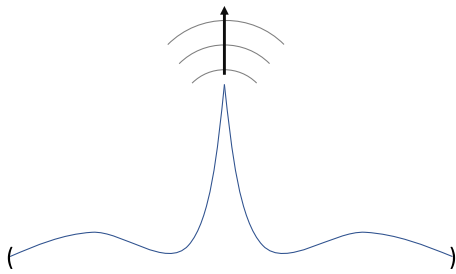


FIG. 2. A visualization of a burst gravitational wave produced at a cosmic string cusp. As the string snaps into a cusp, a directed gravitational wave is emitted in the direction of acceleration.

of mechanisms. Examples are the formation of cusps and kinks [16]. This work focuses on cusps in closed cosmic strings. A cusp is a singular point on a curve where the tangent vector vanishes, or in other words, a singularity where a point traveling along the curve would have to reverse its direction. When this happens, the physical string snaps into a cusp shape and is at that point instantaneously accelerated to the speed of light. A burst gravitational wave is then emitted in the direction of acceleration [13–15]. This is visualized in Fig. 2. The waveform  $h$  of such a signal in the Nambu-Goto limit for loop length  $l$  at redshift  $z$  and tension  $G\mu$ , in natural units where the speed of light  $c$  is taken to be unity, has been computed as a function of frequency  $f$  as [16]

$$h_{l,z,G\mu}(f) = \left[ (2/3)^{2/3} 8/\Gamma^2(1/3) \frac{l^{2/3} G\mu}{(1+z)^{1/3} r(z)} \right] f^{-4/3} \approx \left[ 0.85 \frac{l^{2/3} G\mu}{(1+z)^{1/3} r(z)} \right] f^{-4/3}. \quad (1)$$

In this formula  $r(z)$  is the comoving distance to the loop, or the distance of the observer to the loop, and  $\Gamma$  is the Gamma function. The extrinsic parameters for detection are distance and the sky location. The shape of this waveform in the time domain, and a spectrogram of a strain of noise including this waveform, are shown in Fig. 3.

State-of-the-art methods employed in cosmic string searches such as matched filtering (reviewed in Sec. III) are hindered by the similarity of cosmic string cusp signals to short-duration transient glitches like blip glitches. Blip glitches are defined as transient bursts with a duration of around 25 ms with frequency concentrated between 30 Hz and 250 Hz [22]. Depending on the viewing angle and assumptions on loop length [21], a cosmic string cusp signal may occupy this same frequency range. This paper is focused on the development of methods with respect to blip glitches. However, the methods treated could be extended to any class of short-duration glitches affecting cosmic string cusp searches. Although the morphology of such glitches can differ strongly from cusp signals, in the worst-case scenario they may look near-identical, especially

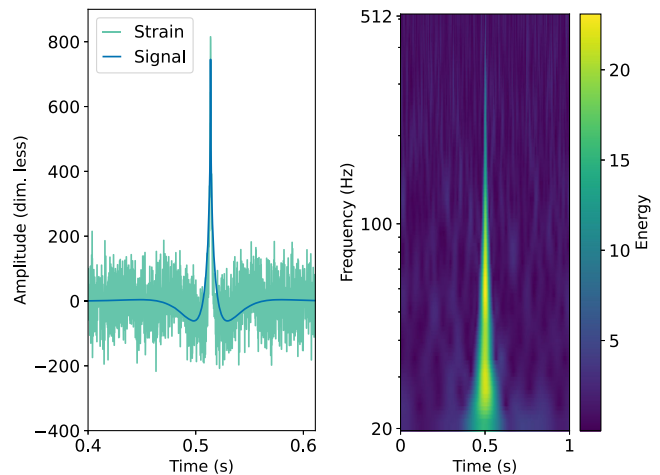


FIG. 3. A cusp signal with an amplitude of approximately  $9.85 \times 10^{-22} \text{ Hz}^{1/3}$  prior to injection, overlaid onto the noise it was injected into on the left, and the spectrogram of this strain on the right. Note that the amplitude absorbs the parameters  $l$ ,  $z$ , and  $G\mu$  per Eq. (4).

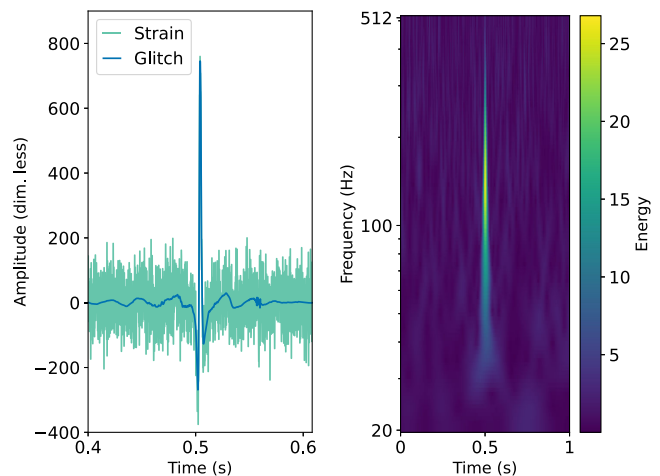


FIG. 4. A glitch generated by GENGLI, overlaid onto the noise it was injected into on the left, and the spectrogram of this strain on the right. The glitches were scaled to follow the SNR distribution of the cusp signals. This procedure is explained in Sec. IV A.

when accounting for the diffusion caused by background noise. This worst-case likeness is demonstrated in Figs. 3 and 4.

### III. CURRENT METHODS

Current state-of-the-art methods rely on matched filtering, which is optimal for finding known signals in the presence of stationary and Gaussian noise [36]. Matched filtering convolves a known signal (or filter) with a data segment in order to obtain a signal-to-noise ratio (SNR) value that indicates the presence of the signal in the data. If the value so obtained exceeds a preset threshold, it is said the filter was matched to the data, and the GPS time of the

trigger is stored. In gravitational-wave pipelines, this trigger is the starting point for a series of statistical tests to confirm a gravitational-wave candidate [16,37].

Given a linear space of complex functions into which the waveforms can be embedded, the SNR is dependent on the following Hermitian inner product for  $u$  and  $v$  taken from this space,

$$\langle u, v \rangle = 4\text{Re} \left[ \int_0^\infty \frac{u(f)\overline{v(f)}}{S_n(f)} df \right], \quad (2)$$

where  $S_n$  is the power spectral density (PSD) characterizing the detector noise and the bar denotes the complex conjugate. Any template can now be normalized with respect to this inner product. For a template  $x$ , let the normalising factor  $\langle x, x \rangle$  be labeled  $c_x$ . Taking the detector strain as being  $s(t) = n(t) + h(t)$  where a signal  $h$  is expected, with  $n$  being the noise, the SNR  $\rho_s(x)$  of the normalized template  $x$  in the strain  $s$  is defined as

$$\rho_s(x) := \langle s, x \rangle. \quad (3)$$

In the presence of a signal, meaning  $h$  is not identical to zero, the measured SNR (signified by a tilde)  $\tilde{\rho}_x(t)$  for a signal of amplitude  $A$  is a random variable normally distributed as  $\mathcal{N}(c_x A, 1)$  [18,38,39]. Using this observation, the data can be match filtered against a set of waveform templates called a template bank. The template that best matches the data will produce the highest SNR.

Matched filtering has two major drawbacks. The first is the need for a template bank that sufficiently covers the parameter space which in general can be of high dimension, showcasing issues with scalability. The second is that, specifically for cosmic string searches, matched filtering is not robust to glitches, confusing the two classes due to their similar morphology. These points argue the case that it is worthwhile to explore alternatives to matched filtering for candidate detection in search pipelines. One natural choice is that of neural networks which in theory can address both drawbacks. From a theoretical point of view, it is interesting to note that work is being done towards the replication of matched filtering as neural networks [40]. One could then make a case that neural networks can strictly improve on matched filtering.

#### IV. METHODOLOGY

For the task of training convolutional neural networks on both the as-of-yet undetected cosmic string cusp signals and the per definition unpredictable glitches, a dataset incorporating advanced domain knowledge needs to be constructed. Once this data format is established, the network architecture is treated, along with the design decisions involved. Finally, the methodologies for a comparison to the state-of-the-art and making interpretations of the deep-learning model are described.

#### A. Construction of the dataset

The Einstein Telescope will consist of three detectors in a triangular configuration [41]. As such, three detector strain data streams will simultaneously be collected. In this work, these streams will be labeled stream 0 through 2. Einstein Telescope data was simulated by first producing colored Gaussian noise and then injecting cusp signals and blip glitches into the streams.

For each of the three streams of the Einstein Telescope, a Gaussian noise time series of length twelve seconds was generated, that was subsequently colored by the PSD representing the Einstein Telescope design sensitivity [42] using PyCBC [43]. The design sensitivity of the Einstein Telescope along with the sensitivities of current (second) generation detectors [44] are shown in Fig. 5. The noise realizations were then injected with cusp signals to form the positive class, and artificial glitches to form the negative class.

The cusp waveforms in the time domain were generated through the use of the LALSimulation package [45]. The function for the generation of the plus-polarized cusp strain components requires three inputs; an amplitude  $A$  in  $\text{Hz}^{1/3}$  [normalizing Eq. (1)], a high-frequency cutoff  $f_{\text{high}}$  in Hz past which the waveform will drop exponentially, and a sample period  $\Delta_t$  in Hz. Here,  $A$  represents the amplitudal prefactor in Eq. (1),

$$A := 0.85 \frac{l^{2/3} G \mu}{(1+z)^{1/3} r(z)}, \quad (4)$$

so that the waveform in the time domain is given by the inverse Fourier transform of

$$h(f) = A f^{-4/3} (f_{\text{high}} - f)^+, \quad (5)$$

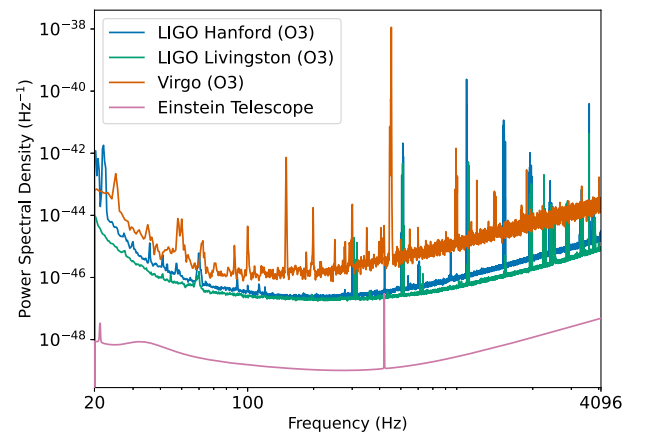


FIG. 5. The Einstein Telescope design sensitivity compared to the realized sensitivities of the current generation of detectors in the third observational run (O3).

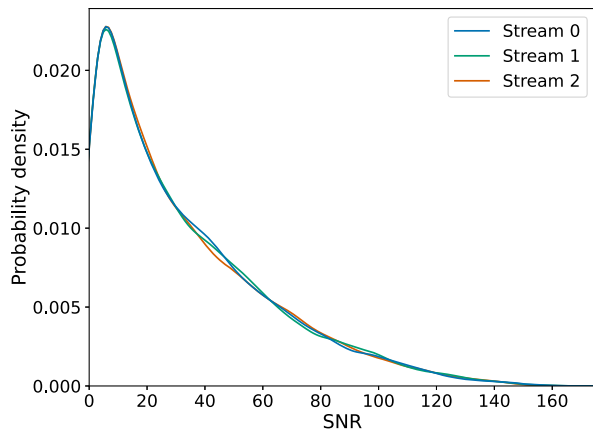


FIG. 6. The SNR distribution of injected cusp waveforms modelled as a probability density. The streams refer to the streams of the detectors making up the Einstein Telescope.

where the plus signifies the taking of the positive part, and the output time series is of this transformed function at sample period  $\Delta_s$ . In order to randomly generate waveforms, the amplitudes  $A$  and cutoff frequencies  $f_{\text{high}}$  were uniformly sampled from  $[10^{-23}, 10^{-21}]$  and  $\{20, 21, \dots, 4000\}$ , respectively. These generated cusp signals, assuming an isotropic distribution, were projected according to the Einstein Telescope antenna pattern and injected into Gaussian-colored noise sampled at 8192 Hz, before the strain was whitened and cropped to a length of eight seconds. The resulting SNR distributions of this positive class are shown in Fig. 6.

The glitches were artificially generated using the GENGLI package [46], which has learned to model blip glitches in the time domain by harnessing generative adversarial networks [47]. Currently, GENGLI approximates the real distribution of glitches in O2 data, specifically that of LIGO Hanford and Livingston. This data includes anomalies, and it is, therefore, possible anomalies showing a different morphology than blip glitches are generated. Using the GENGLI similarity metrics, an accepted region is defined that excludes roughly one in ten glitches that are deemed too dissimilar from blip glitches. These outliers are discarded. This procedure is described in [48].

The true morphology and intensity of Einstein Telescope glitches are currently unknown. It is however reasonable to assume that short-duration glitches similar to blips will be present in the recorded data, and as they are in fact a worst-case scenario in terms of similarity to the cusp signals, they form the best possible preparation. In order to further ensure the robustness of the models to be trained on this dataset, the generated glitches are scaled in amplitude to follow the SNR distribution of the injected cusps shown in Fig. 6. This ensures that the models do not learn a difference in SNR distribution. The injection procedure itself differs from that of cusp signals, since GENGLI generates whitened glitches. These glitches are summed as a time series to eight seconds

of whitened noise at randomly drawn offsets. The offsets per stream are uncorrelated and the glitches are chosen randomly, meaning there is no detector coincidence for the glitches. Examples of both injected glitches and cusp signals are shown in Figs. 3 and 4.

For both classes, no further preprocessing has taken place. In order to both preserve the original information and retain computational efficiency, time series are used instead of alternative representations like spectrograms.

The resulting dataset consists of 30,000 examples (or data points), split into training, validation and test sets of sizes 16,000, 4000 and 10,000 respectively. Each subset is balanced, meaning it is made up of equal parts positive examples (signals) and negative examples (glitches).

## B. The WaveNet architecture

The convolutional neural networks [49] discussed in this section are implemented in PyTorch [50] and were run on the LIGO Data Grid. The specific machine used has the following specifications; Intel E5-2670 CPU, NVIDIA Tesla V100 16 GB GPU, and 128 GB of memory.

WaveNet [51] is an expressive convolutional neural network designed for the generation of high-fidelity speech audio. The architecture is capable of handling long-range temporal dependencies at high sampling rates, achieved by creating a large receptive field through the use of dilated convolutions, or dilations. Dilations allow the network this reception by skipping over a preset number of neurons in each layer, dilating the layers. By appending dilated layers, an exponential increase in the receptive field is gained at the cost of a linearly increasing number of layers, as is illustrated in Fig. 7.

The major building blocks of WaveNet are residual block modules as presented in Fig. 8. The figure shows that input to the module is passed through a convolutional layer, after which it is simultaneously passed through both tanh and sigmoid gates. The activations [49] are recombined in elementwise multiplication, where the sigmoid activations modulate the throughput, determining how much of the tanh output activation is passed [52]. The output is convolved with  $1 \times 1$  filters to reduce the number of parameters before being fed into the residual connection [53,54]. Note that at this point a copy of the throughput is sent to a skip connection [54].

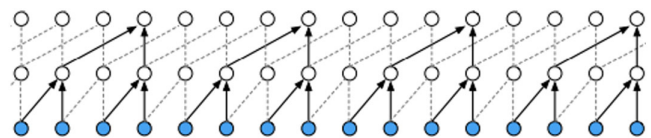


FIG. 7. Dilation between the layers of the neural network (shown horizontally), retrieved from [51]. As data is passed upwards through the layers, an increasing number of neurons is passed over, creating a larger diagonal reach for the neurons in the top layer.

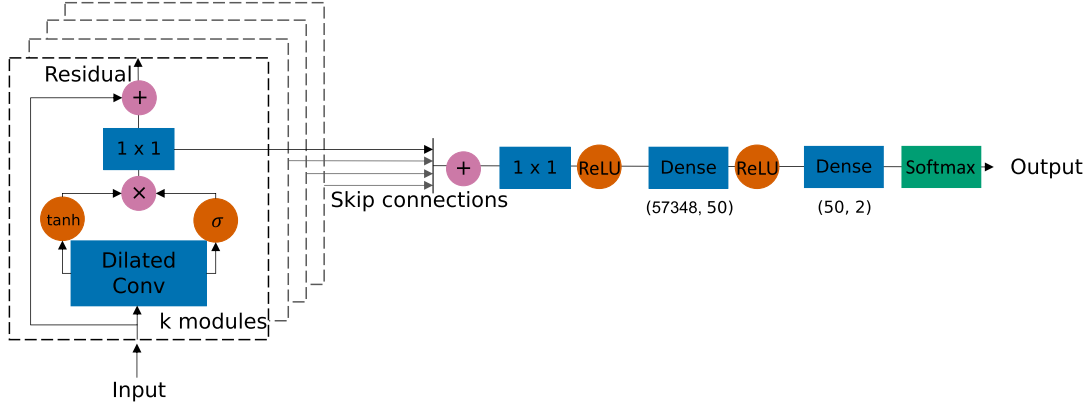


FIG. 8. Overview of the modified wavenet architecture for a single data stream, adapted from [51]. The hidden layers are colored blue, the internal activations are shown in orange, and the normalizing softmax layer is shown in green.

Inspired by the methodology proposed in [55], where the full WaveNet architecture was modified for the discovery of binary black hole systems, modifications have been made for this project as well. The major changes are listed below:

- (i) Instead of encoding the time series amplitudes in a range of 256 possible values (see [51] for details), no such limit is imposed in our implementation;
- (ii) The causal structure intended for the dependencies in human speech was removed so as to provide the most possible information to the model;
- (iii) The dilated convolutions have a kernel size of 3 to capture fine details, and the dilation within the  $k$ th block module (of 11) is set to  $2^k$ ;
- (iv) The steps preceding the softmax activations were removed in favor of dense layers. In order to produce a probability, the activations need to be collapsed onto a scalar value in the unit interval. This too is shown in Fig. 8.

Together, these changes tailor the architecture to the needs of binary classification instead of the originally intended generation.

### C. Design and parameter choices

The first major design choice is the use of an ensemble. Instead of training a single network on the three streams, one network was trained for each, and the three final networks were combined into an ensemble. This has several advantages. The first is the handling of different glitches being injected into the streams, therefore not allowing the ensemble to resort to using coincidence for its classification and forcing it to consider morphologies. Second, the independent networks can learn different characteristics during their training phase, averaging out to a more well-informed final decision by the ensemble. This average is taken literally, as the probability  $\mathbb{P}(x)$  output by the ensemble for an example  $x = (x_0, x_1, x_2)$  of strains is the average of the components networks  $\mathbb{P}_i$  for  $i \in \{0, 1, 2\}$ ,

$$\mathbb{P}(x) = \frac{\mathbb{P}_0(x_0) + \mathbb{P}_1(x_1) + \mathbb{P}_2(x_2)}{3}. \quad (6)$$

When the time does arrive that coincidence is needed to confirm a candidate detection in joint analysis, these probabilities can be transferred to a central machine instead of the data containing the candidate. This greatly reduces latency, as a single probability is less costly to transmit than a time series.

The weights of each network were determined using stochastic gradient descent, specifically using the AdamW optimizer [56] with learning rate  $10^{-4}$  and a weight decay of  $10^{-3}$ . These values were further varied, yielding no significant improvement at this small scale. The batch size was set to 13. Due to the complexity of the model, increases in the batch size resulted in a direct gain in performance, and this trend is likely to continue. For this model, the batch size was limited by memory.

In the training phase, each separate network was trained independently for 20 epochs, resulting in the training and validation cross-entropy losses shown in Fig. 9. This phase was repeated multiple times to ensure the optimizer did not get stuck in an avoidable local minimum. It can be read from the validation losses that because of the small batch size, overfitting started between the second and fourth epochs, marked by vertical lines.

From here on, an ensemble is defined by the three ordered epochs at which the training of the networks was halted, denoting the ensemble so created as an  $[i, j, k]$ -ensemble for  $i, j, k$  between the values of 0 and 19. Choosing weights according to the times where overfitting started, the  $[2, 4, 2]$ -ensemble was established as the initial candidate. Classifiers defined by nearby stopping times in the  $i, j, k$  lattice were checked by brute force iteration but gave no improvement over the  $[2, 4, 2]$ -ensemble. This ensemble was therefore chosen.

Both the individual network thresholds, the ensemble threshold, and combinations of the two were fine-tuned on the validation set. The most important measures used in the

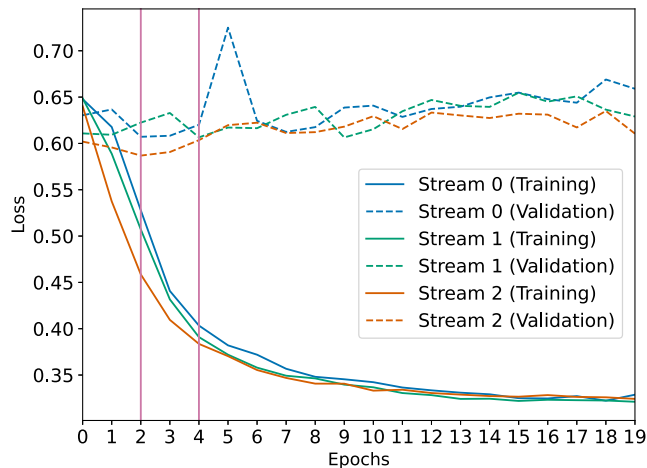


FIG. 9. Training and validation losses for the three networks. The stopping times are represented by vertical lines, marking the weights used for the networks.

fine-tuning were the accuracy, true positive rate and false positive rate. As can be seen from Fig. 10, the ensemble probabilities  $\mathbb{P}$  on the validation set are highly concentrated in the neighbourhoods of 0 and 1, giving no cause to deviate far from an ensemble threshold of 0.5. The viable range for thresholds to test was set to the uniform set spanning from 0.4 to 0.6 with step size 0.01, with none leading to a significant improvement over the default value of 0.5. A similar line of reasoning has led to thresholds of 0.5 for the component networks.

#### D. Comparison to matched filtering

A direct comparison between the deep-learning model, a binary classifier, and matched filtering, which is not a binary classifier, is not straightforward. In order to benchmark the deep-learning model against matched filtering, a

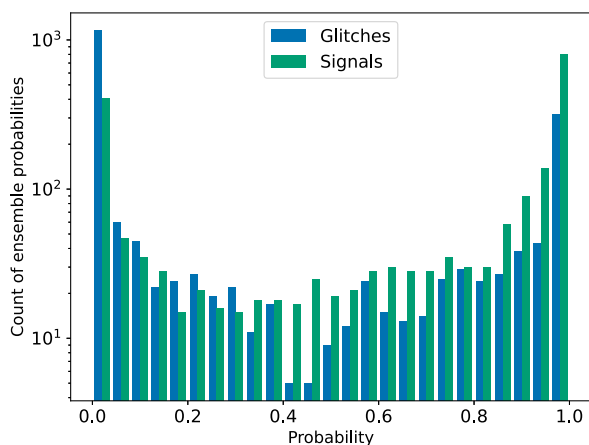


FIG. 10. Counts of the ensemble probabilities for both classes, with the y-axis in the log scale.

new balanced dataset of total size 400 was constructed, again following the SNR distribution shown in Fig. 6 for both signals and glitches injected into Gaussian noise colored by the Einstein Telescope design sensitivity. Recall that these were labeled the positive and negative examples, respectively.

For a given positive example, each of the three streams was match filtered against the exact injected waveform. This waveform is per definition the optimum filter, and the trigger value is defined as the global maximum of the three SNR time series.

For the negative examples, the optimum filter will not exist as a cusp waveform template, as no true signal was injected. The choice of templates is therefore arbitrary. In order to simulate realistic circumstances, a template bank was created by randomly sampling 20 of the 200 signals generated during the creation of the comparison dataset. Including more templates would be detrimental to the performance of matched filtering, as these additional templates would only allow for the measured SNR to be increased, where it is known no cusp signal is present. Hence, the results from this comparison can be considered conservative. The performance of matched filtering could only be improved by constructing a template bank of cusp waveforms where no template can be matched to blip glitches, which defeats the purpose of the comparison. The remainder of the procedure is identical to that for the positive examples so that the method is internally consistent.

#### E. Model interpretability

Neural networks are notoriously hard to interpret because of their large dimensionality and opaque optimization procedures. Ideally, however, the discriminative properties the networks have learned would be extracted, in order to better understand the morphological differences between the injected signals and injected glitches. So as to learn what the neural networks have learned, a variety of methods is proposed to interpret the behaviour of our deep-learning model.

##### 1. Surgeries

The first method employed to better the understanding of our deep-learning model is what will be referred to as glitch surgeries. Surgeries are limited to the class of glitches which are not subject to detector antenna patterns, meaning impact can more directly and accurately be measured for glitches. Moreover, they are more readily split into different regions on which surgeries can be performed. The predetermined parts of a selected glitch are excised before reclassifying the modified example and quantifying the change in the ensemble prediction with respect to the original input. In doing so, features salient to the deep-learning model can be identified.

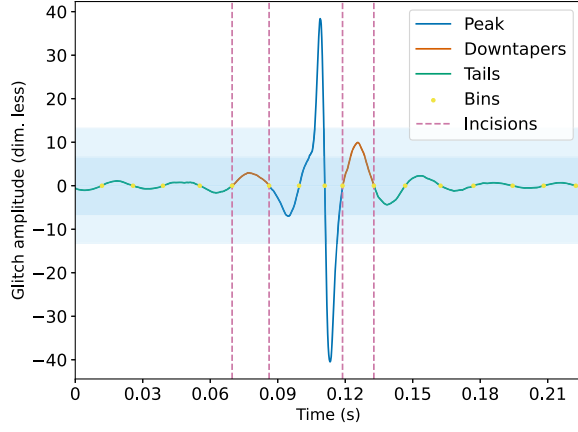


FIG. 11. The sections of a glitch within a visualization of the surgery procedure. The shaded areas represent the standard deviation of the amplitude from zero.

The observation underpinning the procedure is that a glitch  $g(t)$  can generally be divided into five regions based on the maximum amplitude within these regions and that these regions together form the sections shown in Fig. 11. These regions can be automatically detected by partitioning a glitch into bins delimited by the zero crossings and comparing the absolute maximum of each bin with a function of the standard deviation  $\sigma$  of the glitch amplitude. The edges of the bins are constrained to correspond with zero crossings to ensure continuity, as an excision amounts to setting the value of the glitch waveform amplitude to zero within the bins that are excised. Whereas continuity is required so the neural networks do not pick up on the transitions, it is not necessary to extend the waveform to be smooth at the bin edges, as this transition is lost within the noise after injection.

First, the area of peak activity, which one should note may contain more than one peak, is identified as being the bin containing the absolute maximum amplitude  $|\max(g)|$  of the glitch in the time domain. Moving outwards left and right over the bins, starting from the identified bin, the peak area is extended to include adjacent bins if the absolute maximum within these bins exceeds  $2\sigma$ . The down taper of the glitch starts at the first bin where the absolute maximum within the bin is below  $2\sigma$ , and the tail of the glitch starts at the first bin where the absolute maximum is below  $\sigma$ . Note that these definitions may imply the absence of named sections in a glitch waveform, as for instance, a section corresponding to the down taper might not exist. This can be the case if the maximum amplitude of the waveform is extremely high compared to the average amplitude. Such glitches can safely be included in the surgery procedure. The excision of a non-existent section amounts to nothing changing at all, and the results from the reclassification will reiterate that the nonexistent section did not contribute to the classification.

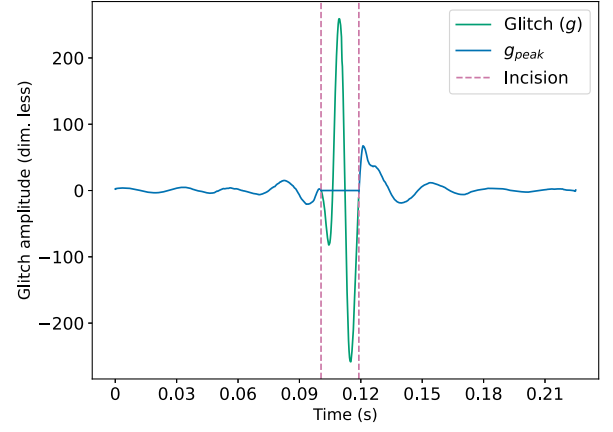


FIG. 12. A glitch before and after surgery. During the surgery the peak was excised from  $g$ , thus obtaining  $g_{\text{peak}}$ . The corresponding statistic is  $\Delta_{\text{peak}}(g) \approx -0.47$ .

For a representative sample taken from the dataset, the procedure is then as follows:

- (1) Choose a glitch example  $g$  from the set, and retrieve the ensemble probability  $\mathbb{P}(g)$ ;
- (2) For any of the three sections, set  $g$  identical to zero within the corresponding bins to obtain  $g_{\text{section}}$  (performing the surgery);
- (3) Reinject and classify  $g_{\text{section}}$  before retrieving the ensemble probability  $\mathbb{P}(g_{\text{section}})$ .

The statistic of interest is then

$$\Delta_{\text{section}}(g) := \mathbb{P}(g) - \mathbb{P}(g_{\text{section}}). \quad (7)$$

Note that this statistic takes values in  $[-1, 1]$ . The natural interpretation is that a value close to  $-1$  means that the classification has significantly changed, with  $g$  being classified as a glitch previously and as a signal following the surgery. A value close to 1 would imply the reverse. A stream from an example with a value of  $\Delta_{\text{peak}} \approx -0.47$  is shown in Fig. 12. This behavior can be further explored by considering the changes for the individual component networks within the ensemble.

## 2. Activations

Another way of investigating the behaviour of the ensemble is the extraction and visualization of the activations in the hidden layers as a testing example is passed through. This information can then be used to tie certain convolutional filters to specific confusion matrix classes (Fig. 13) in the dataset. Note that these are filters according to the terminology of neural networks, not those of matched filtering. Inspiration was drawn from saliency maps [57] from computer vision, meant to highlight the most salient and therefore recognisable regions of images. Although projects like CAPTUM [58] offer similar ways of interpreting convolutional neural networks, they differ from the method



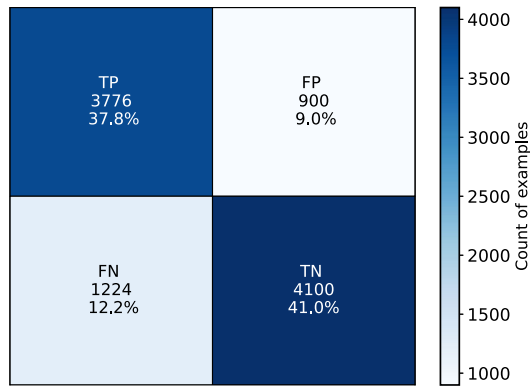


FIG. 13. The confusion matrix showing the true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) for the  $[2, 4, 2]$ -ensemble on the test set, visualized as a heatmap.

described here, designed specifically for the analysis of time series data.

A straightforward way of obtaining the activations (that also works for general networks) is to deconstruct a given network into an ordered set of individual layers, applying these layers one by one, and saving the outputs before feeding the output forward. Once these values are recovered, the challenge of interpreting the activations is reduced to trying to connect the activation of specific filters to fundamental characteristics of the example that was passed through. This is akin to detailing a collection of neurons that fire when a specific example is seen. As this is an extremely difficult task with high dimensionality, only isolated observations can be made.

In order to understand how the activations can be best visualized, it is useful to review the process of a filter being applied. As a filter is convolved with the one-dimensional time series, a new time series containing a large number of activation values is obtained and fed out. Due to the number of values, a direct plot of the activations would be unreadable. Instead, the values are binned and smoothed with a kernel density estimate. The resulting curve is an indicative visualization of the activation for the specific filter used. The reader is invited to look ahead at Fig. 18, which shows these curves for the examples that will be interpreted in the next section.

### 3. Principal component analysis

Whereas the extraction of the activations serves mostly to delve into the hidden representations of the data as it is passed through the modules, principal component analysis (or PCA) [59] can be used to analyze the representation in the linear layers. PCA is a dimensionality reduction method that linearly maps vector data into a lower-dimensional space with an ordered basis consisting of what are called the principal components. These components are determined as being the basis vectors carrying the most amount of information measured by variance, and their ordering is

based on these same amounts. This means that for instance, the first principal component contributes the most to the overall variance of the dataset. PCA is applied to the second to last dense layer shown in Fig. 8, where an input of size 57,348 is collapsed to an output of size 50 before the latter values are further reduced to a single probability. Based on these numbers one can argue that in this layer the most amount of information is condensed, making it a valuable object of study. In Sec. VC3 the first two principal components obtained from the dense layer are studied.

## V. RESULTS

In this section, the numerical results of the chosen deep-learning model are reported and discussed before treating the information extracted from the model by applying the interpretability methods presented in Sec. IV.

### A. Numerical results

On the test set, the  $[2, 4, 2]$ -ensemble yields the confusion matrix visualized in Fig. 13, from which the metrics presented in Table I were computed. In terms of cosmic string searches, the accuracy refers to the model's capability of recovering the injected signals and glitches. Likewise, the true positive rate (TPR) quantifies how well the model can recognize a signal, given that a signal was injected. Finally, the false positive rate (FPR) measures to what degree the model mistakes glitches for signals, given that no signal was injected. This means that a value of 0 is ideal for the FPR, and 1 is ideal for the accuracy and TPR.

Manually investigating the most extreme false positive examples, spurred by the relatively high FPR, most skew high on this metric due to the three streams having been injected with glitches sharing a similar morphology between them, shown in Fig. 14. It appears the extremely high maximum amplitude (as compared to the average amplitude) dominates the morphology, leaving the deep-learning model little other distinguishing features to base its classification on. For these specific examples, the model erroneously resorted to a positive classification.

There are a few noteworthy exceptions, each appearing only in one of the three streams that make up an example, showing oscillations in amplitude over a larger period of

TABLE I. A selection of performance metrics for the  $[2, 4, 2]$ -ensemble on the test set, along with their formulas. These values were computed from the confusion matrix in Fig. 13 using the true positives (TP), false positives (FP), false negatives (FN), true negatives (TN), with  $P := TP + FP$  (positives) and  $N := TN + FN$  (negatives).

Metric	Formula	Value
Accuracy	$(TP + TN)/(P + N)$	0.7876
True positive rate	$TP/(TP + FN)$	0.7552
False positive rate	$FP/(FP + TN)$	0.1800

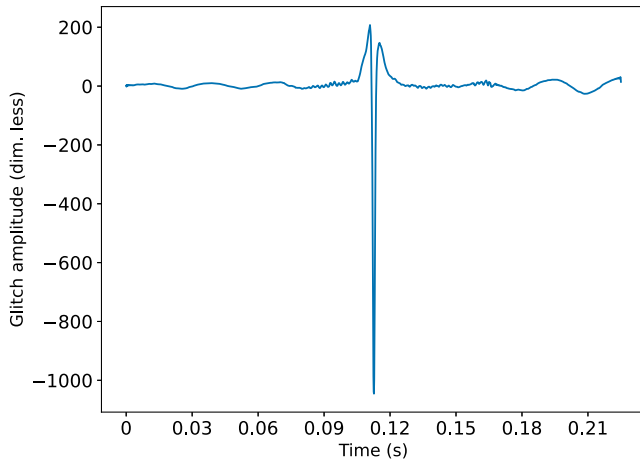


FIG. 14. An example of a glitch that was falsely classified as a positive with high probability. It is therefore an extreme example of a false positive.

time. Such a glitch is shown in Fig. 15. One possible explanation for these instances is that the component network is given more opportunity to detect the presence of a signal signature and that one such signature is sufficient for the example to be classified as a signal. This underlines the importance of the signal morphology to the deep-learning model.

For the archetypal examples shown in Figs. 3 and 4, the component networks for the streams these examples were taken from assigned the glitch a probability of 0.0008 of being a signal, and the signal a probability of 0.7005. This means the networks assign these examples to the right classes with considerable confidence.

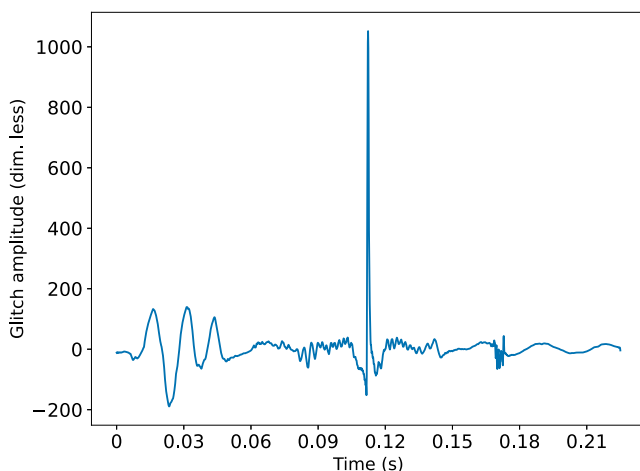


FIG. 15. A different extreme example of a false positive. One possible explanation for this misclassification is that the component network corresponding to this stream is given more opportunity to identify a signature that is believed to be that of a signal.

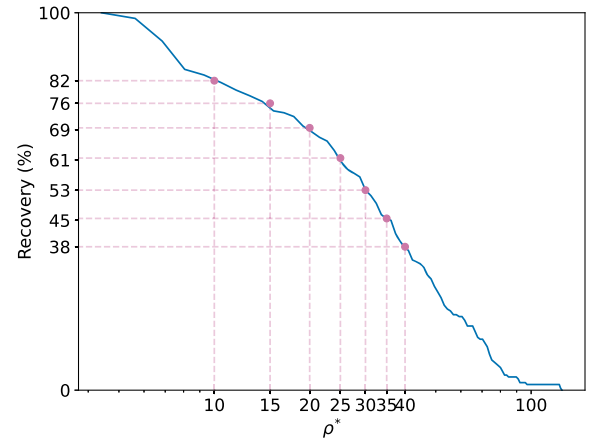


FIG. 16. The recovery percentage of matched filtering for different values of the SNR threshold  $\rho^*$  in log scale. This latter value represents the cutoff from which point onwards an SNR is considered high enough for the corresponding example to be labeled as containing a signal.

Lastly, on the machine used (described in Sec. IV B), the classification speed of one example (consisting of three data streams of 8 seconds) was computed to be 10 milliseconds on average.

## B. Comparison to matched filtering

Conventionally, the performance of matched filtering is measured with positive examples being signals added to noise, and negative examples being drawn from a colored Gaussian noise background without a signal present. Recall however that the current consideration is the distinguishing power of the deep-learning model and matched filtering for a dataset where the positive examples are injected signals, and the negative examples are injected glitches.

The true positives recovered by matched filtering at different SNR thresholds  $\rho^*$  is shown in Fig. 16. This choice for the representation of the results was made in order to remain agnostic towards the chosen threshold, which may differ per analysis. A direct comparison between the deep-learning model and matched filtering is shown through the receiver operating characteristic (ROC) curves in Fig. 17. The diagonal represents a random binary classifier, meaning that on this dataset, matched filtering is weighed down by false positives to the point where its performance is worse than random classification. In contrast, the deep-learning model is very effective on the same dataset. The conclusion is that the model is better at rejecting glitches than a simple matched-filter search with cosmic string cusp templates by a large margin. A more complete comparison, including for instance the additional mechanisms that would be present in a full gravitational-wave search pipeline and would work to ameliorate false alarms, is deferred to future work.

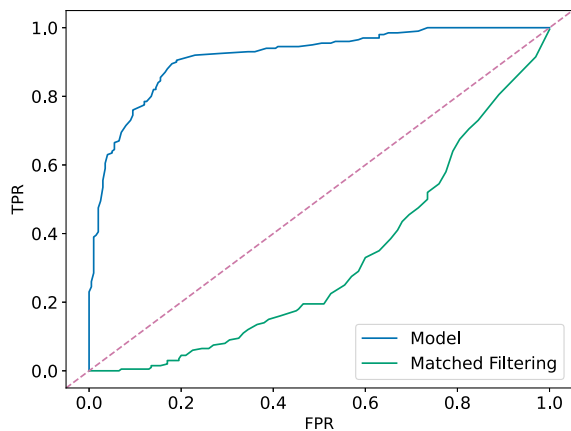


FIG. 17. The receiver operating characteristic curves for both the deep-learning model and matched filtering on a dataset consisting of injected signals and glitches. The performance of a random binary classifier is represented by the diagonal.

### C. Interpretability

In this section the results of the various interpretability methods described in Sec. IV E for the interpretation of the deep-learning model are presented.

#### 1. Surgeries

By performing the surgeries described in Sec. IV E 1, the statistics given in Table II were obtained. These statistics show that with a division of a glitch into three sections, the peaks will by far be the most informative, with the downtapers and tails contributing relatively little. The low average and maximum values for the latter section statistics bound the values on the whole set of glitches, indicating that for no negative example the removal of either the down tapers or tails has made a significant impact on the reclassification.

Investigating the outliers near the maximum of 0.205 for  $\Delta_{\text{tails}}$ , it was found that these values stem from glitches with high fluctuations in tails that were removed. For the values on the low end of  $\Delta_{\text{peak}}$ , a similar observation is made. The removal of the peaks for these glitches left behind fluctuations in amplitude in the down tapers or tails, on which the deep-learning model will presumably base its classification instead. Most of the examples with a high value of  $\Delta_{\text{peak}}$  have very large amplitudes in the peak

TABLE II. The average and maximum values of  $\Delta_{\text{section}}$  statistics, separated per section. These values offer a summary of the output from the surgery process, where  $\Delta_{\text{section}}$  measures the impact of a glitch section removal on the model classification.

Section	Average( $\Delta_{\text{section}}$ )	Maximum( $\Delta_{\text{section}}$ )
Peak	0.161	0.832
Down tapers	0.001	0.055
Tails	0.004	0.205

section, the removal of which confuses the model. An interesting note is that for these examples the network trained on stream 0 seems to be less impacted by the excision of peaks than the other two networks in the ensemble, which is possible evidence that the three networks have learned to identify different glitch signatures. Further manual inspection of the small number of examples with  $\Delta_{\text{peak}}$  near  $-1$ , meaning the classification has changed from a glitch to a signal following the surgeries, shows that all have remaining fluctuations in their waveforms. One theory is that the model considers these remnants as the new peak sections, viewing at least one as evidence of a present signal. This observation might suggest that without detecting a clear glitch signature, the model defaults to a signal classification. This would complement the discussion on the false positives in Sec. VA.

Relating to the preceding discussion, if the model indeed resorts to analyzing amplitude spikes within the sections that remain after a surgery has been performed, this suggests the model considered these sections as secondary to the peak region before. In turn, this suggests that the model does not simply detect rapid changes in amplitude, but has learned to differentiate morphologies.

#### 2. Activations

Based on the confidence of the deep-learning model, one example was chosen for each of the classes in the confusion matrix, and their activation values were extracted. This means, for instance, that in the case of the true negative, an example with an output probability very close to 0 was selected. The activations of these four examples for a single module are visualized in Fig. 18 and each example will be discussed individually.

For the true negative example in Fig. 18(a), a number of filters show activation, meaning the mean of the density curve is closer to 1 than it is to 0. However, this is with a high spread in the curve, indicating uncertainty in the activation values for this filter. Some filters, such as 27 in green or 37 in blue, show higher certainty. This is however not enough to mislead the model into making a false positive classification.

The filters in the false negative in Fig. 18(b) see barely any activation taking place at all. For this example, there was nothing giving the model the impression there could be a signal present. The only filters showing a semblance of activation do so with little certainty, with output probabilities not high enough to cross the classification thresholds at which point the model would classify the example as positive.

The false positive example shown in Fig. 18(c) seems to invoke response from the filters, with activity within multiple filters. However, as was the case for the false negative, there is not much certainty. In this case, however, the probabilities did cross the classification thresholds.

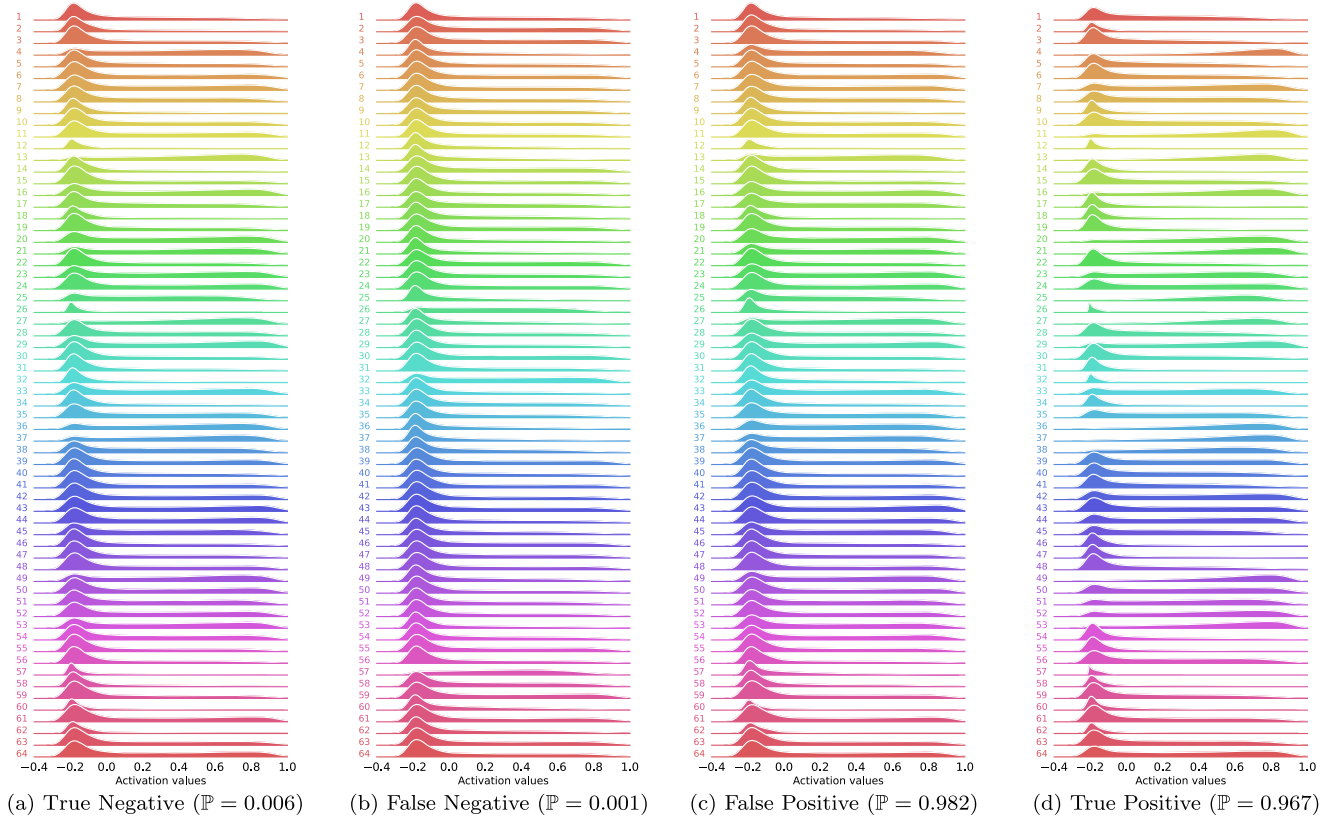


FIG. 18. Residual activations output by the 11th module in Fig. 8. Every horizontal axis represents a filter with the y-axis limited to  $[0, 0.1]$ , showing the distribution of activation values as a density curve. The mode of such a curve being close to 1 indicates high activation values for the filter.

Finally, the centroids of the true positive example in Fig. 18(d) skew strongly towards the right, meaning high values of the activations are achieved. A wide array of filters show strong activation, with certainty higher than the previous examples. The individual filters, and the network as a whole, are certain this example contains a signal.

The above observations were made on single examples, and are therefore not guaranteed to generalize. They do however show a clear difference in response to examples from the four classes and are therefore a proof of concept for further investigation. Individual filters can for instance be mapped back to certain sections of the input streams and examined further. This is outside the scope of this work.

As a final remark for this subsection, there are some filters that show little to no activation for any of the four examples, with 9 in orange and 60,62 in red being such filters. While this is possibly due to the choice of examples or a lack of need for these filters, it is also possible this is a result of the low number of training epochs, meaning the weights for these filters have not been properly adjusted. If this is the case, one might conclude there is room to improve the model further. One of the ways this could be done is by reducing memory usage during the training phase, leading to better training that may in turn recruit the now dormant filters.

### 3. Principal component analysis

For the first stream in the test set, the activations of the dense layer were projected onto the subspace spanned by the first two principal components. The results from this projection are shown in Figs. 19 and 20, colored by the probability  $\mathbb{P}_0$  output by the first network in the ensemble and the SNR, respectively. These figures are shown in log scale to improve the visual separation between the two classes. Both figures show a portion of the signal population being located in the top left of the plot, whereas a portion of the glitch population is located in the top right. Both classes overlap in the center and are therefore plotted separately to improve visibility. It is relevant to note that in the principal component space, glitches show a larger spread than the signals. This follows from their more varied possible morphologies.

It can be observed from Fig. 19 that the probability is related to the first principal component on the x-axis. Compared to Fig. 20, the extremes of this same principal component show high values for the SNR. From this, it can be inferred that at least within this representation, the signals and glitches exhibiting the most separability are the ones that are loudest and therefore most obvious to the model. The first principal component can thus be interpreted as a

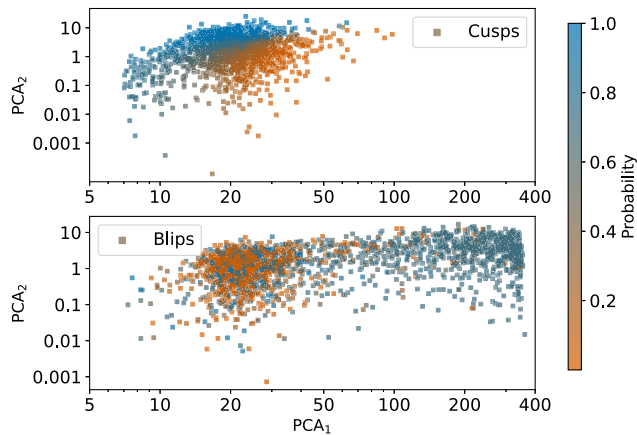


FIG. 19. A representation of a selection of activations from the dense layer in the space spanned by the first two principal components  $PCA_1$  and  $PCA_2$ , in log scale. The plots of the two classes were split to improve visibility that may otherwise be hindered by the overlap of the classes. The points are colored by the probability  $\mathbb{P}_0$  output by the first network in the ensemble.

measure of the example class. For the second principal component, there is no such apparent meaning.

## VI. CONCLUSIONS

A deep-learning model that can distinguish between cosmic string cusp signals and blip glitches with significant accuracy was designed and analyzed. Given that matched-filter searches for short-duration gravitational-wave signals are heavily hindered by short transient glitches such as blip glitches, the exploration of this task is important for both current and future searches. In this work, both populations were scaled to follow the same SNR distribution, meaning loudness was removed from the equation. With remarkable

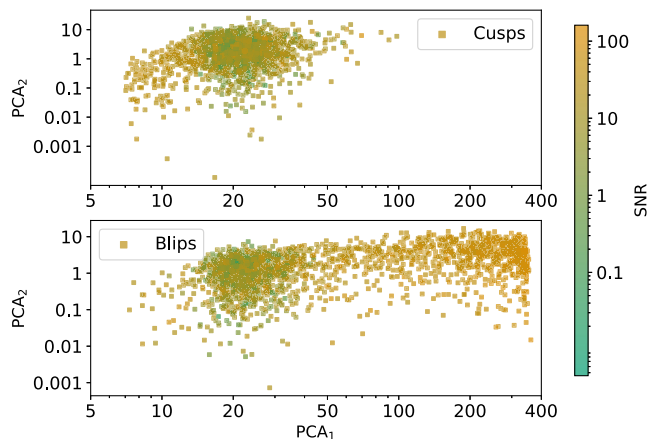


FIG. 20. A representation of a selection of activations from the dense layer in the space spanned by the first two principal components  $PCA_1$  and  $PCA_2$ , in log scale. The plots of the two classes were split to improve visibility that may otherwise be hindered by the overlap of the classes. The points are colored by SNR.

results for the accuracy (79%) and true positive rate (76%) in particular, it has been shown that deep learning is a viable candidate for use in cosmic string searches. Moreover, due to the classification speed of 10 milliseconds per three data streams of 8 seconds, the deep-learning model is fast enough to run as part of a real-time detection pipeline.

On a dataset consisting of injected signals and injected glitches, the deep-learning model was shown to outperform matched filtering at the task of distinguishing strains including signals from strains including glitches, winning mostly on the volume of false positives (as can be seen from the model's slow increase in true positive rate in Fig. 17). This demonstrates that the deep-learning model is significantly better at rejecting glitches.

The behaviour of neural networks is notoriously difficult to understand, earning them the name of black-box models. As evidenced by their proven effectiveness, however, these black boxes hide valuable information. The hidden representations within the deep-learning model were interpreted through the application of three interpretability methods. The first of these is the method of waveform surgery, introduced in this paper, where parts of a waveform are removed to study the effects on the classification of such a procedure. The second method is a routine developed in this paper for the visualization of convolutional filter activations for one-dimensional time series. The third is principal component analysis. These interpretations have resulted in several observations that may prove useful in future work. The glitch surgery procedures demonstrate the possibility of dividing waveforms into sections and show that models can be sensitive to changes within these sections. Surgery procedures can therefore be used to study the importance of distinct sections of waveforms to their classification. By considering a comparison between waveforms based on their sections, the complexity of signal discrimination can be reduced, therefore potentially reducing the difficulty of the task. Through the study of the convolutional filter activations, these latter values can be connected to the classes in the confusion matrix. Such studies may aid in making informed choices for convolutional filters, or more generally in neural network design. Lastly, principal component analysis applied to the throughput of the second to last dense layer of the network has enabled the study of class separability and the hidden procedure reducing the internal representation of the deep-learning model to output probabilities.

In the process of interpreting the deep-learning model, the morphological differences between cosmic string cusp signals and blip glitches were considered from the point of view of the model. Because the model was not allowed to rely on coincidence, it was fully dependent on these morphologies, yielding unique insight. At the same time, this serves as a proof of concept for high classification performance before coincidence is introduced to further improve a pipeline.

There are several directions open for continued work, such as the inclusion of other gravitational-wave generating mechanisms on cosmic strings. These mechanisms are comprised by kinks and kink-kink collisions, signals with different spectral indices from cusps that are now starting to be considered in cosmic string searches [16,21]. Furthermore, there are indications the proposed deep-learning model offers additional room for improvement, for instance through an extended training phase. It is expected this adjustment will also serve to lower the false positive rate. In terms of analysis, the surgery process can be further refined, for instance by working at a resolution higher than three sections. This can be achieved by redefining the function of the standard deviation that marks the incisions.

Altogether, it is expected that the Einstein Telescope will bring a variety of new opportunities for the detection of cosmic strings and that deep learning will play a vital role in their analysis.

## ACKNOWLEDGMENTS

The authors thank Adrian Helmling-Cornell and the anonymous referee for their helpful comments. With thanks to Artim Bassant for being open to string-theoretical discussion. Q. M and M. L. are supported by the research program of the Netherlands Organisation for Scientific Research (NWO). D. T. is supported by the Sherman Fairchild Postdoctoral Fellowship at Caltech. S. C. is supported by the National Science Foundation under Grant No. PHY-2309332. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by the National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation.

- 
- [1] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Observation of gravitational waves from a binary black hole merger, *Phys. Rev. Lett.* **116**, 061102 (2016).
  - [2] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs, *Phys. Rev. X* **9**, 031040 (2019).
  - [3] R. Abbott *et al.* (LIGO Scientific and 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).
  - [4] R. Abbott *et al.*, GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, *Phys. Rev. X* **13**, 041039 (2023).
  - [5] The LIGO Scientific and the 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, *Phys. Rev. D* **109**, 022001 (2024).
  - [6] J. Aasi *et al.*, Advanced LIGO, *Classical Quantum Gravity* **32**, 074001 (2015).
  - [7] F. Acernese *et al.*, Advanced Virgo: A second-generation interferometric gravitational wave detector, *Classical Quantum Gravity* **32**, 024001 (2014).
  - [8] M. Evans *et al.*, A horizon study for cosmic explorer: Science, observatories, and community, [arXiv:2109.09882](https://arxiv.org/abs/2109.09882).
  - [9] J. Baker *et al.*, The laser interferometer space antenna: Unveiling the millihertz gravitational wave sky, [arXiv:1907.06482](https://arxiv.org/abs/1907.06482).
  - [10] M. Maggiore *et al.*, Science case for the Einstein telescope, *J. Cosmol. Astropart. Phys.* **03** (2020) 050.
  - [11] A. Vilenkin and E. P. S. Shellard, *Cosmic Strings and Other Topological Defects*, Cambridge Monographs on Mathematical Physics (Cambridge University Press, Cambridge, England, 1994).
  - [12] T. W. B. Kibble, Topology of cosmic domains and strings, *J. Phys. A* **9**, 1387 (1976).
  - [13] T. Damour and A. Vilenkin, Gravitational radiation from cosmic (super)strings: Bursts, stochastic background, and observational windows, *Phys. Rev. D* **71**, 063510 (2005).
  - [14] T. Damour and A. Vilenkin, Gravitational wave bursts from cosmic strings, *Phys. Rev. Lett.* **85**, 3761 (2000).
  - [15] T. Damour and A. Vilenkin, Gravitational wave bursts from cusps and kinks on cosmic strings, *Phys. Rev. D* **64**, 064008 (2001).
  - [16] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), Constraints on cosmic strings using data from the third Advanced LIGO—Virgo observing run, *Phys. Rev. Lett.* **126**, 241102 (2021).
  - [17] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), All-sky search for short gravitational-wave bursts in the second Advanced LIGO and Advanced Virgo run, *Phys. Rev. D* **100**, 024017 (2019).
  - [18] X. Siemens, J. Creighton, I. Maor, S. R. Majumder, K. Cannon, and J. Read, Gravitational wave bursts from cosmic (super)strings: Quantitative analysis and constraints, *Phys. Rev. D* **73**, 105001 (2006).
  - [19] J. Aasi *et al.*, Constraints on cosmic strings from the LIGO-Virgo gravitational-wave detectors, *Phys. Rev. Lett.* **112**, 131101 (2014).
  - [20] B. P. Abbott *et al.*, First LIGO search for gravitational wave bursts from cosmic (super)strings, *Phys. Rev. D* **80**, 062002 (2009).
  - [21] B. P. Abbott *et al.*, Constraints on cosmic strings using data from the first Advanced LIGO observing run, *Phys. Rev. D* **97**, 102002 (2018).

- [22] M. Cabero *et al.*, Blip glitches in Advanced LIGO data, *Classical Quantum Gravity* **36**, 155010 (2019).
- [23] G. B. Folland, *Quantum Field Theory: A Tourist Guide for Mathematicians*, Mathematical Surveys and Monographs (American Mathematical Society, Providence, 2008).
- [24] M. Sakellariadou, Cosmic strings and cosmic superstrings, *Nucl. Phys. B, Proc. Suppl.* **192–193**, 68 (2009), Theory and Particle Physics: The LHC Perspective and Beyond.
- [25] M. E. Peskin and D. V. Schroeder, *An Introduction To Quantum Field Theory*, Frontiers in Physics (Avalon Publishing, Reading, Pennsylvania, 1995).
- [26] G. E. Bredon, *Topology and Geometry*, Graduate Texts in Mathematics (Springer, New York, 1993).
- [27] A. S. Schwarz, Topologically stable defects, in *Quantum Field Theory and Topology* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1993), pp. 43–55.
- [28] J. Polchinski, *String Theory*, Cambridge Monographs on Mathematical Physics (Cambridge University Press, Cambridge, England, 1998), Vol. 1.
- [29] M. J. Duff, B. E. W. Nilsson, and C. N. Pope, Kaluza-Klein supergravity, *Phys. Rep.* **130**, 1 (1986).
- [30] E. J. Copeland and T. W. B. Kibble, Cosmic strings and superstrings, *Proc. R. Soc. A* **466**, 623 (2010),
- [31] J. J. Blanco-Pillado, K. D. Olum, and B. Shlaer, Number of cosmic string loops, *Phys. Rev. D* **89**, 023512 (2014).
- [32] L. Lorenz, C. Ringeval, and M. Sakellariadou, Cosmic string loop distribution on all length scales and at any redshift, *J. Cosmol. Astropart. Phys.* **10** (2010) 003.
- [33] Andreas Albrecht and Neil Turok, Evolution of cosmic string networks, *Phys. Rev. D* **40**, 973 (1989).
- [34] David P. Bennett and Francois R. Bouchet, High-resolution simulations of cosmic-string evolution. I. Network evolution, *Phys. Rev. D* **41**, 2408 (1990).
- [35] B. Allen and E. P. S. Shellard, Cosmic-string evolution: A numerical simulation, *Phys. Rev. Lett.* **64**, 119 (1990).
- [36] C. W. Helstrom, *Statistical Theory of Signal Detection*, International Series of Monographs on Electronics and Instrumentation (Pergamon Press, New York, 1960).
- [37] Kipp Cannon *et al.*, *GSLAL*: A software framework for gravitational wave discovery, *SoftwareX* **14**, 100680 (2021),
- [38] 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).
- [39] C. M. Biwer, C. D. Capano, S. De, M. Cabero, D. A. Brown, A. H. Nitz, and V. Raymond, *PyCBC* inference: A PYTHON-based parameter estimation toolkit for compact binary coalescence signals, *Publ. Astron. Soc. Pac.* **131**, 024503 (2019),
- [40] J. Yan, M. Avagyan, R. E. Colgan, D. Veske, I. Bartos, J. Wright, Z. Marka, and S. Marka, Generalized approach to matched filtering using neural networks, *Phys. Rev. D* **105**, 043006 (2022).
- [41] ET Steering Committee Editorial Team, Design Report Update 2020 for the Einstein Telescope, Technical Report, 2020.
- [42] Unofficial sensitivity curves (ASD) for a LIGO, KAGRA, Virgo, Voyager, Cosmic Explorer, and Einstein Telescope, <https://dcc.ligo.org/LIGO-T1500293/public> (2020).
- [43] A. Nitz *et al.*, *GWASTRO/PyCBC*: v2.0.6 release of *PyCBC* (2023).
- [44] Noise curves used for simulations in the update of the observing scenarios paper, <https://dcc.ligo.org/LIGO-T2000012/public> (2022).
- [45] LIGO Scientific Collaboration, *LALSuite*: LIGO Scientific Collaboration Algorithm Library Suite, Astrophysics Source Code Library, record ascl:2012.021 (2020).
- [46] M. Lopez and S. Schmidt, Documentation of the *GENGLI* Package, <https://melissa.lopez.docs.ligo.org/gengli/index.html> (2022).
- [47] M. Lopez, V. Boudart, K. Buijsman, A. Reza, and S. Caudill, Simulating transient noise bursts in LIGO with generative adversarial networks, *Phys. Rev. D* **106**, 023027 (2022),
- [48] M. Lopez *et al.*, Simulating transient noise bursts in LIGO with *GENGLI*, [arXiv:2205.09204](https://arxiv.org/abs/2205.09204).
- [49] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).
- [50] A. Paszke *et al.*, *PyTorch*: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., Red Hook, New York, 2019), pp. 8024–8035.
- [51] A. van den Oord *et al.*, *WaveNet*: A generative model for raw audio, [arXiv:1609.03499](https://arxiv.org/abs/1609.03499).
- [52] A. van den Oord *et al.*, Conditional image generation with PixelCNN decoders, [arXiv:1606.05328](https://arxiv.org/abs/1606.05328).
- [53] C. Szegedy *et al.*, Going deeper with convolutions, [arXiv:1409.4842](https://arxiv.org/abs/1409.4842).
- [54] K. He *et al.*, Deep residual learning for image recognition, [arXiv:1512.03385](https://arxiv.org/abs/1512.03385).
- [55] 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),
- [56] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, [arXiv:1711.05101](https://arxiv.org/abs/1711.05101).
- [57] T. Kadir and M. Brady, Saliency, scale and image description, *Int. J. Comput. Vis.* **45**, 83 (2001).
- [58] N. Kokhlikyan *et al.*, *CAPTUM*: A unified and generic model interpretability library for *PyTorch*, [arXiv:2009.07896](https://arxiv.org/abs/2009.07896).
- [59] I. Jolliffe and J. Cadima, Principal component analysis: A review and recent developments, *Phil. Trans. R. Soc. A* **374**, 20150202 (2016).