

# Deep learning for clustering of continuous gravitational wave candidates

B. Beheshtipour<sup>1,2, a</sup> and M.A. Papa<sup>1,2,3, b</sup>

<sup>1</sup>*Max Planck Institute for Gravitational Physics (Albert Einstein Institute), Callinstrasse 38, 30167 Hannover, Germany*

<sup>2</sup>*Leibniz Universität Hannover, D-30167 Hannover, Germany*

<sup>3</sup>*University of Wisconsin Milwaukee, 3135 N Maryland Ave, Milwaukee, WI 53211, USA*

In searching for continuous gravitational waves over very many ( $\approx 10^{17}$ ) templates, clustering is a powerful tool which increases the search sensitivity by identifying and bundling together candidates that are due to the same root cause. We implement a deep learning network that identifies clusters of signal candidates in the output of continuous gravitational wave searches and assess its performance.

## I. INTRODUCTION

Gravitational waves from binary mergers have ushered us in the era of gravitational wave astronomy [1–3]. More signals are expected than the ones identified so far, namely unmodeled gravitational wave bursts from catastrophic events [4], stochastic backgrounds [5] and continuous gravitational waves [6–10]. This paper is about an important step in the analysis of results from broad surveys for continuous gravitational waves.

Continuous gravitational waves are expected from a rotating compact object when its shape deviates from perfect axisymmetry (“mountains”) or due to a non-axisymmetric motion (as an Ekman flow), to the excitation of long-lived r-modes [11], as well as more exotic scenarios [12–15]. In spite of the diversity of the emission scenarios the signals can all be described as ever-lasting nearly monochromatic waves.

Continuous wave searches can be relatively straightforward or rather complex. In the first category we find searches for emission from pulsars whose spin evolution is known from electromagnetic observations. In this case the expected gravitational waveform can be predicted based on the observed spin frequency, and standard techniques can be used [7, 16, 17]. The computational cost of these searches is negligible and optimal sensitivity can be achieved. The landscape is completely different when one searches for continuous gravitational waves from unseen objects.

The rationale for searches for continuous gravitational wave signals from unseen objects is solid: whereas we see a few thousand pulsars, based on stellar evolution models, we expect there to be over  $10^8$  compact objects in our Galaxy, 10% of which might be spinning between 10 Hz and 1000 Hz, making them candidates for emission in the high-sensitivity band of LIGO/Virgo.

The challenge with this type of search is that the number of combinations of frequency, frequency-derivatives and sky positions that one would need to search for, with a fully coherent search over months of data (this would be the optimal strategy), is too high to be doable. So one

has to resort to semi-coherent schemes that attain the highest possible sensitivity within the computing budget constraints. There exist a number of semi-coherent approaches, making different trade-offs between sensitivity, breadth, computational efficiency and robustness with respect to deviations of the signal waveform from the assumed shape [18].

Whatever the specific search may be, the detection statistic may be triggered to rise significantly above the noise by both a gravitational wave or a disturbance. Since the template grids are rather fine, whenever the data resembles a template, it is likely that it also resembles nearby templates. This means that for every signal and disturbance we typically have a plethora of templates with detection statistic values above threshold – “candidates” – to consider. Here “to consider” means to follow-up with a more sensitive search, to either confirm as interesting or to discard.

The number of candidates above a given threshold increases as the threshold decreases. The search sensitivity increases as the detection threshold decreases but the total number of candidates that we can follow-up is limited by available computing power. So on the one side we’d like to decrease the threshold to achieve higher sensitivities, on the other we’d like to increase it to keep the computing cost within bounds.

Clustering helps tip the balance towards higher sensitivities at the same computing cost, by identifying nearby-candidates due to the same root-cause and by bundling them together as one. The follow-up is then performed only based on the most significant of the set, and this reduces the computational cost.

The clustering algorithms utilised in the highest template-count searches – the Einstein@Home searches – have evolved over time [19–21]. The latest ones measure local over-densities in the detection statistic distribution in parameter space and categorize the resulting clusters as generated by noise or signal, based on their morphology. The morphologies of signals and noise disturbances vary depending on the data and the specific search setup, so this method requires lengthy preparatory studies and ad-hoc tunings for every search.

Because so much of the tuning is based on a sort of “craftmanship” in interpreting the output of key clustering quantities on fake-signal results and on noise, and based on this, in deducing effective tuning-parameter val-

<sup>a</sup> b.beheshtipour@aei.mpg.de

<sup>b</sup> maria.alessandra.papa@aei.mpg.de

ues, we decide to investigate whether a deep learning network could be trained to acquire such craftsmanship and do this just as well or better. In this paper we examine the performance of a clustering method based on a deep learning network.

Deep learning or deep neural networks are a subfield of machine learning which is inspired by the way brain works. A neural network consists of a series of neurones (connected processors) that endeavours to recognize relations between a question (input) and an answer (output). This field has greatly developed in the past decade and has proven successful in applications such as image classification, object detection [22, 23], speech recognition and text to speech [24, 25].

In physics, and specifically in the field of gravitational waves, convolutional neural networks have been used for data preparation, for instance to classify noise transient in the data [26, 27] and for de-noising the data of non-gaussian and non-stationary noise [28, 29]. Neural networks are also being pursued to detect binary inspiral signals and to estimate the associated system parameters [30–34] and recent work has investigated their use for the detection of the elusive continuous gravitational waves [35].

Armed with this background, we investigate how well a neural network could identify parameter space regions (clusters) in the results of very broad continuous gravitational wave searches, which will be further searched. We associate the input result data to an image and address the clustering method by using instance segmentation models.

The paper is organized as follows. Sec. II describes the network input data and the network architecture. We report on the result of the network in Sec. III and discuss them in Sec. IV.

## II. A NEURAL NETWORK FOR IDENTIFYING AND CLUSTERING CONTINUOUS GRAVITATIONAL WAVE CANDIDATES

### A. Basics of neural networks

Artificial neural networks are computing systems which can be trained to perform certain tasks by example rather than by following an algorithm. The field has a long history [36] but with increased accessibility to substantial amounts of computing power the last decade has seen renewed interest in artificial neural networks, the blooming of numerous techniques and a slew of applications.

In this section we introduce some of the key quantities that describe a network and provide an *intuitive* description of their role. We have no pretence of providing a thorough review of neural networks and refer the reader to the many existing excellent text books for this (see for example [37]); this section is for the reader who does not know much about neural networks but would like to

understand how we have used them for the problem at hand.

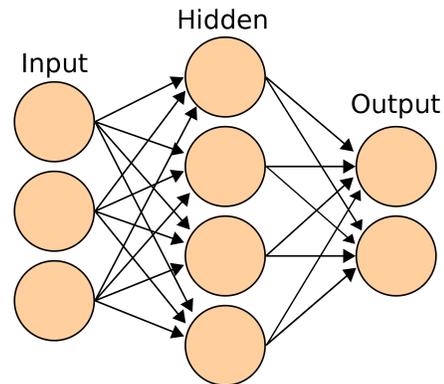


FIG. 1. Simplest neural network, consisting of a single hidden layer, from [38]. The nodes (neurones) are represented by the circles.

A neural network recognizes relationships/features in the input data and can be used to tell whether some data presents such relationships/features. The network is trained to recognize the desired features based on input data that presents such features. Broadly put, a network labels the data based on various examples of correct labelling of similar data, which are referred to as “ground truth”.

A network is composed of layers of nodes/neurones. Each node combines the data linearly with a set of coefficients and biases, the so-called model-weights.

The result of this combination is fully or partially transmitted to the next layer based on its value, through an activation function.

The training consists in determining the sets of coefficients and biases for every layer that result in the last layer returning the most accurate labelling of the data.

A simple network architecture includes an input layer, an output layer, and a middle layer which is called hidden layer, as schematically shown in Fig. 1. More complex structures are possible with multiple hidden layers or internal level networks, merged in a larger network. A neural network is called deep when it has more than two hidden layers [39].

The training is performed starting from a random set of model weights. A few data sets are run through this initial network, the outputs are produced and compared with the corresponding ground truths. Based on this comparison the coefficients and biases are adjusted, and the network is updated. The process is repeated again with another batch of input data and corresponding ground truths. A new network with new model weights is produced. The network is trained through a series of successive trial-and-error adjustments, made by iterating this procedure a large number of times, to achieve an output that is closer and closer to the ground truth. The “learning rate” describes by how much the model weights

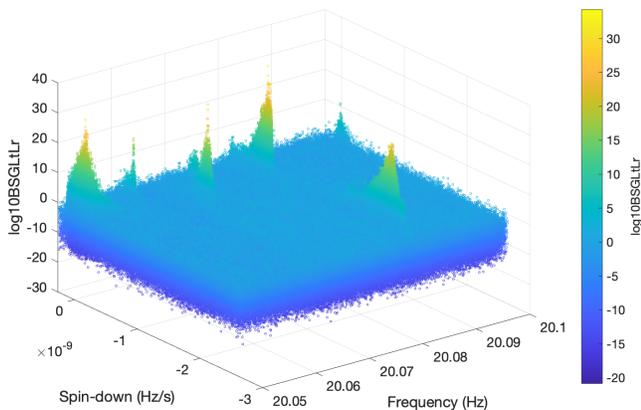


FIG. 2. Results in the 20.05-20.1 Hz band : the detection statistic as a function of  $(f, \dot{f})$ . Several fake signals were added to the data before the search and appear in this result-set as parameter space regions with enhanced values of the detection statistic.

are modified at each training iteration.

The comparison between the output of a network and the corresponding ground truth is measured by the distance between the two, i.e. by the error made by the network. The training aims at minimizing such error by measuring the way in which the error changes when the model weights are changed and by adjusting the latter accordingly. This is the optimisation problem at the heart of the functioning of neural networks. “Loss” is also used to quantify error, and “accuracy” is the complementary term to loss.

Constructing a successful network is a combination of different factors: casting the labelling problem in an effective manner, preparing the input data to reflect most clearly the relationships that one wants to test, having a broad enough training data-set and determining the best network architecture (the structure of the layers, the optimisation algorithm, the activation function) and the best training parameters (e.g. the size of each training batch, the number of batches and the learning rate).

## B. Gravitational wave search results

We use as our test bed for the new clustering implementation, the results from the Einstein@Home search [40]. This will make it possible to easily compare our results with the current clustering algorithm, used in [40].

In an Einstein@Home search [41] the computational workload is split in millions of separate tasks, called “work units” (WUs) that are shared among the volunteer computers. Each task searches of order  $10^9$  templates but only returns information about the most promising  $10^4$  results. The information includes the value of the detection statistics and the associated template parameter values:  $f, \dot{f}, \alpha, \delta$  which indicate frequency, frequency derivative, and the two sky-position coordinates right as-

cension and declination of the gravitational wave source. Typically each WU searches 50 mHz of signal frequencies, the entire  $\dot{f}$  range and a subset of the sky. The results for the same 50 mHz frequency band from all regions of the sky are combined and constitute the result-set for each 50 mHz band. Fig. 2 shows the results in the 20.05-20.1 Hz band as an example.

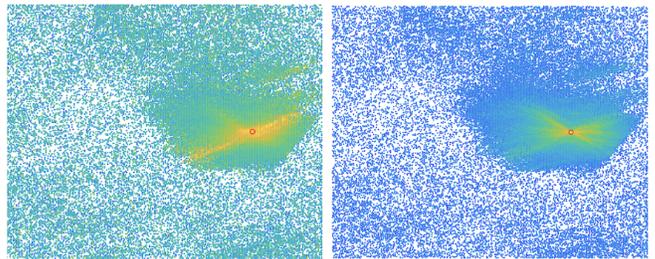


FIG. 3. A section of the input image corresponding to the frequency band between 20.45-20.5 Hz. The right panel shows the original high resolution image; the left panel shows the low resolution ( $256 \times 256$  pixels) image. The red square labels the fake signal parameter values.

## C. Preparation of the input data for the network

For each 50 mHz band of results-data we first produce a high-resolution image. The image has  $\approx 60000 \times 22000$  ( $f \times \dot{f}$ ) pixels, and the color associated with each pixel is determined by the value of the detection statistic. There are many empty pixels corresponding to templates whose detection statistic values were too low to be included in the set of results returned by the volunteer computer. There also exist  $(f, \dot{f})$  pixels where we find more than a candidate, corresponding to different sky positions. When this happens, we pick the one with the highest detection statistic value. Next we produce a lower resolution image by averaging over  $40 \times 40$  pixels. This reduces the size of the image by factor  $> 1500$  and reduces the number of empty pixels by a factor  $\gtrsim 10$ . Finally we divide this image into  $256 \times 256$  pixels sub-images which can be handled by our high-end 32 GB GPU. This is the input to the network. Figure 3 shows the high resolution (original) image and the corresponding final sub-image for data that contain a fake signal.

## D. Network ground truth and training-set

For each lower-resolution image the ground truth is a set of matrixes  $\mathcal{T}_{ij}^\alpha$ , with  $(i, j)$  labelling the  $(f, \dot{f})$  pixels of the sub-image and  $\alpha = 1 \dots N_{cl}$  the total number of clusters found in that image.  $\mathcal{T}_{ij}^\alpha = 1$  if that pixel is part of a cluster, and zero otherwise. To generate the ground truth, we use a series of result-sets created by running the search on data that contains fake signals.

For every 50 mHz band we create the lower-resolution images. In each we identify by eye the visible clusters and mark them using an image editing tool (Pixelmator) on a tablet computer equipped with a touchscreen and a stylus (Fig. 4). Each region is saved with the editing tool and then converted to  $\mathcal{T}_{ij}^\alpha$ . It takes about one minute per signal-cluster to mark and save the associated region.

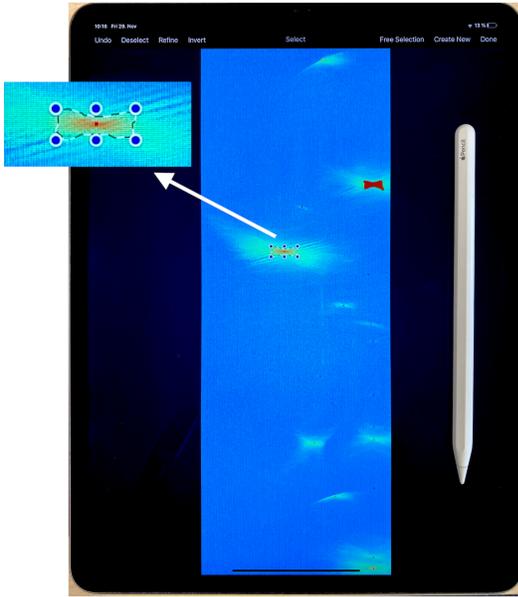


FIG. 4. A lower-resolution 50 mHz image on the tablet computer used to mark the ground-truth signal clusters. In this image 9 clusters are marked. The blue dotted-dashed-line (also the zoomed-in insert) shows the cluster region being identified and the brown area indicates a cluster which was previously identified and is already saved.

In order to train the network more thoroughly, we artificially augment the training-set by using the same sub-images again but having applied one or more of the following transformations on it: horizontal or vertical flip, rescaling all pixel values with the same random value (sampled uniformly from the interval 0.1-1.5) and dropping out a small fraction (20%) of the pixels. The corresponding ground-truth masks are consistently transformed.

### E. Network Architecture

Instance segmentation architectures have greatly developed in the past few years and are widely used in visual-recognition problems [23, 42, 43]. Instance segmentation detects objects, i.e. classifies individual objects and localizes each with a bounding box, and also labels each pixel in a fixed set of categories. For complex images containing different objects, R-CNN (Region-based Convolutional Neural Networks, [44]) are becoming a crucial pre-requisite to instance segmentation, because

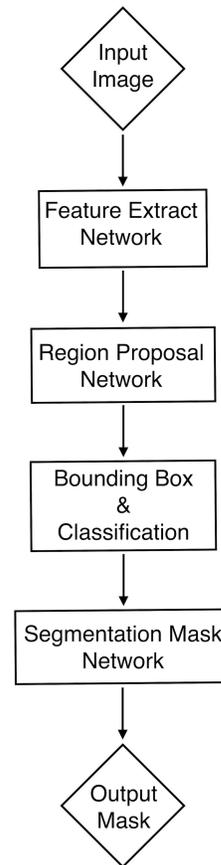


FIG. 5. Schematic diagram showing the Mask RCNN architecture.

they can successfully identify and discriminate *different* objects (*semantic* segmentation). Instance segmentation can then be applied on each identified category.

We use a Mask R-CNN architecture [22] and follow its implementation in [45] with the python deep learning library Keras. The Mask R-CNN network consists of two main stages. The first stage scans the image and finds the regions that most likely include the objects (feature extraction + region proposal); the second stage classifies the regions and generates definitive boundaries (bounding box and classification + segmentation mask). The output is a pixel mask that determines the boundaries and a score that each identified region is a cluster. We set a threshold on this score at 0.5 that determines when we accept the region as a cluster. So for our specific problem the classification is rather simple – signal cluster or not so. The boundaries and classification are provided by the output  $\mathcal{T}_{ij}^\alpha$ , which is our output mask. Figure 5 shows this network framework.

For the feature extraction step we make standard choices and use the Resnet101 [46] and the FPN (Feature Pyramid Network) [47] deep convolutional networks. As Optimizer, we use Statistical Gradient Descent.

### III. RESULTS

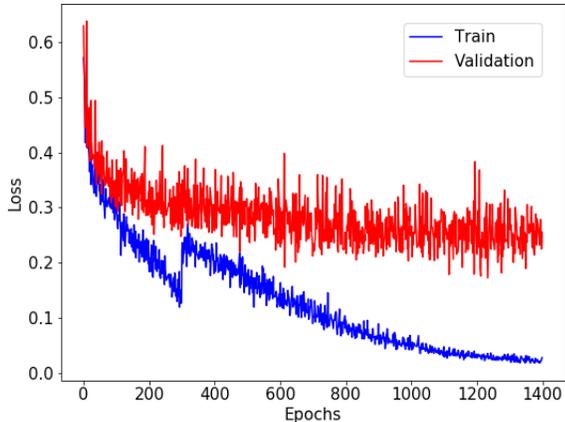


FIG. 6. The network loss as a function of training epochs. The jump in the training happens at stage 3 when new untrained layers are added to the network.

#### F. Network training

The training is performed on a GPU Quadro GV100 with 32 GB memory. To enable the training on a larger batch size, our network framework allows a further reduction in the resolution of the input ground truth mask. In our case the input ground truth mask is internally always reduced to  $28 \times 28$  pixels, represented by floats rather than zeros and ones, in order to hold more details.

We use pre-trained model weights as a starting point for the network because preparing a large training-set ab initio is demanding. Although the pre-trained network has not been trained for our specific problem, using it significantly helps convergence of the training process while using a smaller training set.

The training is performed in 3 steps: 1) only the first layer of the three last levels of the network (i.e. the three last networks of Fig. 5) are trained, for 100 epochs. 2) using the weights from the previous step, the first three levels are trained for 300 epochs 3) the complete network is trained with the weights from the second step for 1000 epochs. For the first two steps we use a learning rate=0.001. The learning rate in the third step is reduced to 0.0001. The best performance of the network is achieved with these hyper-parameters: batch size=15, weight decay=0.00001, learning momentum=0.9.

Fig 6 shows the evaluation of the network as a function of training epoch. The sudden increase in the loss shown in Fig. 6 happens at the beginning of stage 3, when new layers with random weights are added to the training and so the network is in an unoptimized condition. The loss is determined by comparing the number of clusters identified by the network and the morphology of each cluster with the ground truth.

We train the network on 700 sub-images and test it on 670 sub-images. We refer to these data as the training-set and the test-set, respectively. The training-set only contains signal-clusters that can be identified by eye in the results images, i.e. loud signals with respect to the average level of the noise, as can be seen from Fig. 2. There are 1245 such clusters in the training-set and the distribution of the detection statistics for the loudest candidate of each cluster is shown in Fig. 7. For the interested reader, we show both the  $\hat{\beta}_{S/GLtLr}$  and  $2\overline{\mathcal{F}}_r$  detection statistics because  $\hat{\beta}_{S/GLtLr}$  is the statistic that we use to rank the candidates, but the scaling of  $\hat{\beta}_{S/GLtLr}$  with the signal amplitude is hard to tell whereas the scaling of  $2\overline{\mathcal{F}}_r$  is straightforward. It took about 28 hours to complete the training process for this set of data.

The trained network is applied to the test-set. The ground-truth for the test-set is established as done for the training set, i.e. with the procedure described in Section IID. From each cluster we take as representative the candidate with the highest value of the detection statistic. The distribution of candidate detection statistics is shown in Fig. 7, from which one can verify that the training-set and the test-set are comparable.

The test-set comprises 1171 signal clusters. The network returns 1345 clusters of which 1137 are correctly identified signal clusters. This corresponds to a detection efficiency of  $\approx 97\%$ . The network detection efficiency as a function of the candidate detection statistic is shown in Fig. 8. All clusters with candidate  $\hat{\beta}_{S/GLtLr}$  above 35 or  $2\overline{\mathcal{F}}_r$  above 40 are detected and the network has detection efficiency above 95% at  $\hat{\beta}_{S/GLtLr}$  above 17 or  $2\overline{\mathcal{F}}_r$  higher than 18. As a point of reference, for a population of signals at  $\approx 90$  Hz, with  $h_0 \approx 1.75 \times 10^{-25}$ , irrespective of whether they give rise to clusters which are like the ones that the network was trained for, the overall detection efficiency is  $\sim 67\%$ .

Fig. 9 shows examples of clusters identified by the network on the test-set and the corresponding ground truth. It is occasionally hard for the network to identify clusters which are close to larger clusters. In such situation either the two clusters are detected as one or are partially detected, or only the larger one is detected. The loss of detection efficiency for high-detection statistic value clusters stems only from this type of occurrence which is however not realistic for astrophysical signals because those are expected to be few and very sparse. At lower detection statistic values the small fraction of missed signal clusters is due to the signal features being too faint to be reliably identified.

Most of the 174 false alarms are clusters that are very close to correctly-identified signal clusters. Fig. 10 shows an example of this: a loud signal produces enhanced values of the detection statistic at parameter space points relatively far from the actual signal values. The network, quite rightly at this stage, identifies this as a signal cluster.

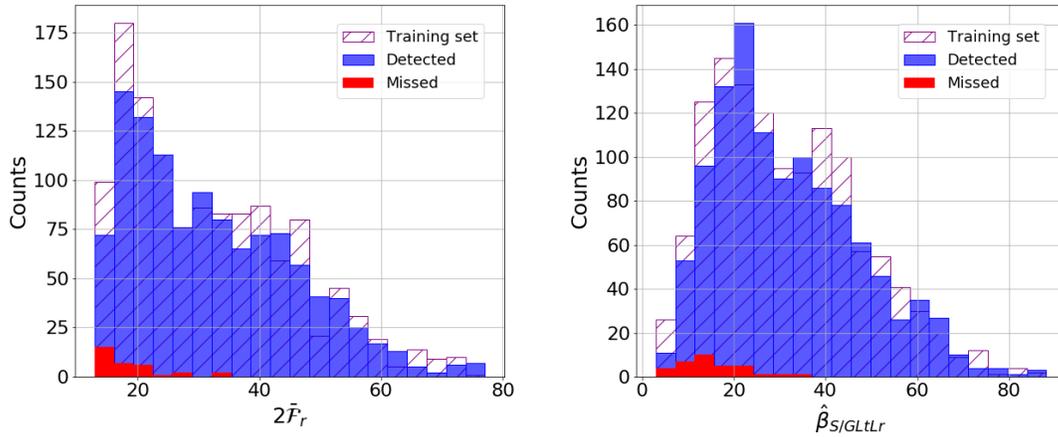


FIG. 7. Distribution of the detection statistics of the test-set (blue) and training-set (dashed) signals. For each signal we take as the representative candidate the one with the highest  $\hat{\beta}_{S/GLtLr}$  value. The distribution of these values for all the signals of the training-set is shown in right panel. Left panel shows the distribution of the  $2\bar{F}_r$  values for each candidate. Some of the weaker test-set signals are not identified by the network; they are shown in red.

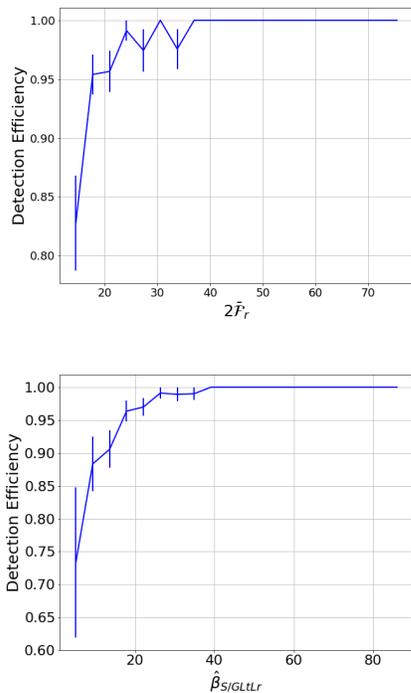


FIG. 8. Detection efficiency of the network. The bars show the statistical error.

ter candidate and hence produces a “spurious cluster”. In an actual search we certainly do not expect many very loud signals, so this type of occurrence is not going to significantly contribute to the actual false alarm rate. If we exclude these, the false alarm is 5%.

The loudest candidate associated with each signal cluster is considered and the distance of the template parameters from the actual signal parameters is determined, and shown in Fig. 11. We find that the uncertainty in the signal parameters of this procedure is comparable to the uncertainty of the clustering algorithm [40]: 90% of the cluster-maxima are within  $\approx 10^{-4}$  Hz and  $\approx 10^{-11}$  Hz/s of the true signal parameter values.

#### IV. DISCUSSION AND SUMMARY

We develop a Mask R-CNN to identify parameter space points that cluster with the morphology that we expect from a signal in the results of broad surveys for continuous gravitational waves. We train and test the network on the results from the Einstein@Home search [40]. We concentrate on loud signals, that are visible by eye in the search results.

Our network shows an excellent performance, detecting the large majority of the target-set signals even for detection statistic values  $\hat{\beta}_{S/GLtLr} = 10$ . The cluster-boundaries that the network defines are by and large consistent with the training-set and the distance of the cluster maximum from the nominal signal parameters consistent with the intrinsic uncertainties of the underlying search procedure.

A typical Einstein@Home search produces of order  $10^4$  50-mHz result-sets, which typically are visually inspected by scientists to check for obvious signals. Albeit one learns a lot about the data and the results in this way,  $10^4$  is a large number, human attention degrades over time and so the reliability of this sort of procedure is hard to gauge in absolute terms. It would be preferable if this

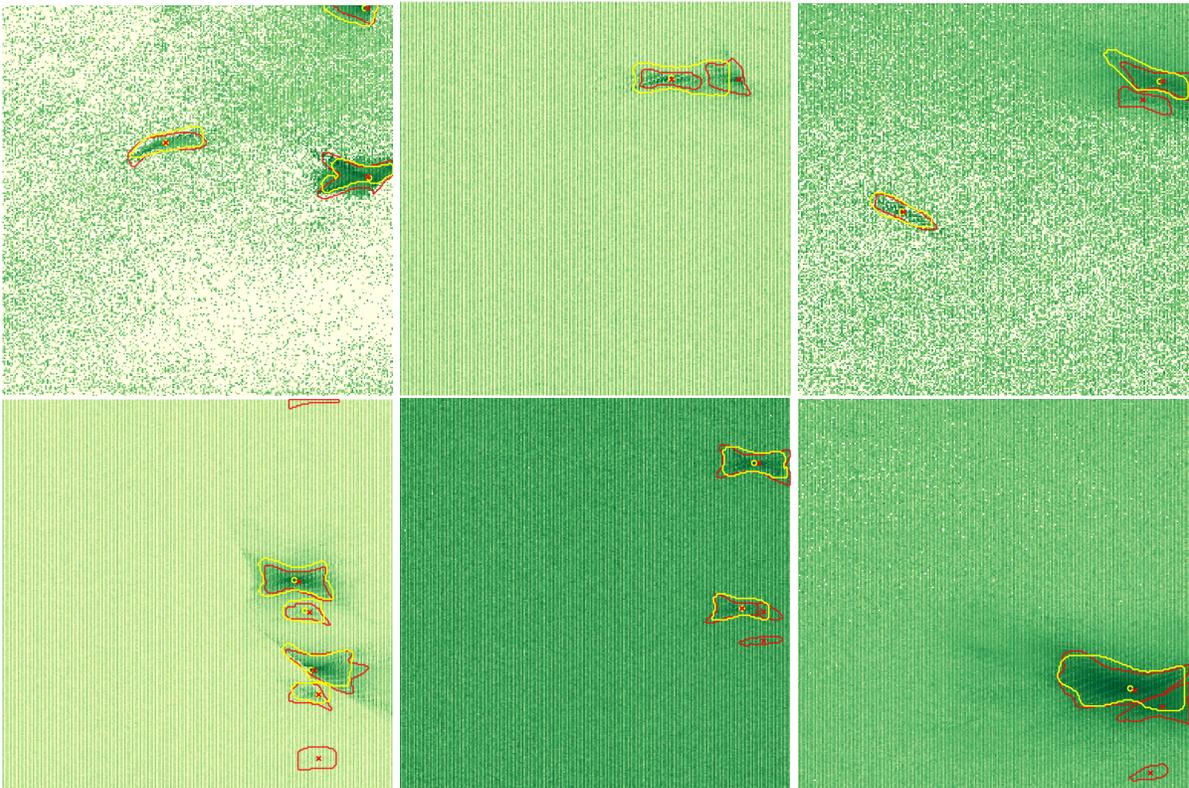


FIG. 9. Sample images of the ground-truth clusters (red) and of the clusters identified by the network on the same data (yellow). The red crosses mark the signal parameter values and the yellow circles mark the location in parameter space of the highest detection statistic candidate identified by the network.

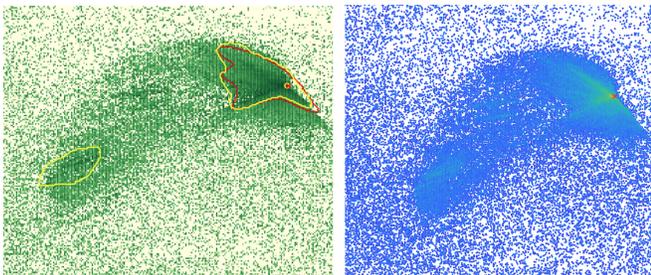


FIG. 10. A section of the input image corresponding to the frequency band between 23.15-23.20 Hz. The right panel shows the original high resolution image; the left panel shows the low resolution ( $256 \times 256$  pixels) image. The red cross labels the fake signal parameter values. The red line shows the ground truth cluster and the yellow line shows the cluster identified by the network and the circle is its loudest candidate. A spurious cluster is visible in the left hand-side panel which is a by-product of the loud signal as explained in the text.

first cursory look at the results could be crossed-check with the results from a different type of system. Our neural network provides this and we will surely utilise it in the post-processing of the results from future Einstein@Home runs.

The question naturally arises of how the network per-

forms on weaker signal. To answer this question we train the same network (i.e. without any optimization and without changing any of its hyperparameters) on much fainter signals. We find that the detection efficiency for these signals is much lower: for  $\hat{\beta}_{S/GLtLr} \sim 20$  the detection rate is  $\sim 17\%$  and when  $0 \leq \hat{\beta}_{S/GLtLr} \leq 15$  it is  $\approx 13\%$ . The reason is that the network works on images whose resolution is over 80 times lower than the resolution of the original result-set. This means that the network works on “maxi-pixels” each containing the average over nearly one hundred pixels of the original image. If the signal is weak, fewer pixels present elevated detection statistic values and the contrast that distinguishes the signal cluster is much degraded. With GPUs with a larger RAM it might be possible to use higher resolution data and improve the network performance for weaker signals.

Our clustering problem is intrinsically a multi-dimensional one. In the case of all-sky searches like [40] it is 4D : frequency, frequency derivative and two sky coordinates. We have reduced it to a 2D problem by projecting all signal parameters in the frequency, frequency-derivative plane, as described in Section II C. By doing so, we can cast our problem as an image processing problem, for which there is a vast literature and standard tools. A possible next step is to design a fully multi-

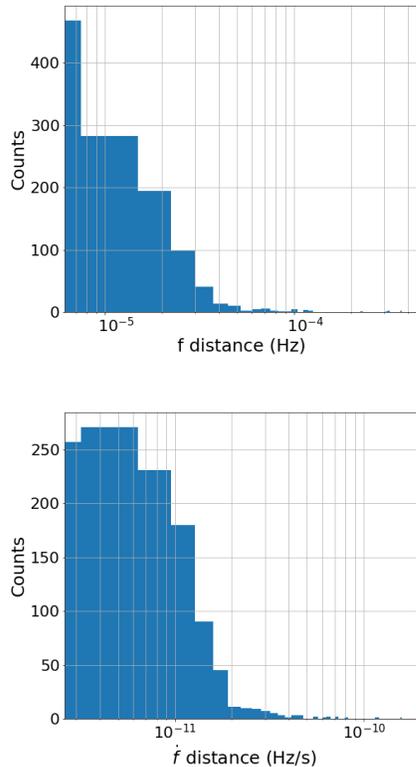


FIG. 11. Distribution of the distance between the signal parameters and the parameters of the most significant cluster candidate (the candidate with the highest  $\hat{\beta}_{S/GL+Lr}$  value) recovered by the network. The top plot shows the distance in frequency; the bottom plot the distance in spin-down (right).

dimensional cluster-recognition network.

We have also tested the ability of this network to identify large disturbances. Its performance turns out to be rather limited. This is not surprising because the network has been trained on signal-like morphologies, but interestingly it indicates that the network *distinguishes* loud signals from loud noise. We will leverage this and enhance the capabilities of the network to include the identification and classification of signals *and* disturbances. This will be extremely useful because it will allow to excise from the result-sets portions of disturbed parameter space while keeping the undisturbed ones for further analysis. Right now an automated procedure recognises 50 mHz bands that contain areas affected by disturbances and discards the entire 50 mHz as “disturbed”. The extension of this network to disturbances would allow to recuperate such regions.

## V. ACKNOWLEDGMENTS

The computing work for this project was carried out on the GPUs of the Atlas cluster of the Observational Relativity and Cosmology division of the MPI for Gravitational Physics, Hannover [48]. We thank Bruce Allen for supporting this project by giving us access to those systems.

- 
- [1] B. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addresso, R. Adhikari, and et al., *Physical Review Letters* **116** (2016), 10.1103/physrevlett.116.061102.
  - [2] A. H. Nitz, C. Capano, A. B. Nielsen, S. Reyes, R. White, D. A. Brown, and B. Krishnan, *The Astrophysical Journal* **872**, 195 (2019).
  - [3] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt, and et al., *Physical Review X* **9** (2019), 10.1103/physrevx.9.031040.
  - [4] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt, and et al., *Physical Review D* **100** (2019), 10.1103/physrevd.100.024017.
  - [5] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. Adya, C. Affeldt, M. Agathos, and et al., *Physical Review D* **100** (2019), 10.1103/physrevd.100.061101.
  - [6] P. D. Lasky, *Publications of the Astronomical Society of Australia* **32** (2015), 10.1017/pasa.2015.35.
  - [7] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, and et al., *The Astrophysical Journal* **879**, 10 (2019).
  - [8] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt, and et al., *Physical Review D* **100** (2019), 10.1103/physrevd.100.024004.
  - [9] V. Dergachev and M. A. Papa, “Results from an extended falcon all-sky survey for continuous gravitational waves,” (2019), arXiv:1909.09619 [gr-qc].
  - [10] V. Dergachev and M. A. Papa, *Physical Review Letters* **123** (2019), 10.1103/physrevlett.123.101101.
  - [11] B. Haskell, *International Journal of Modern Physics E* **24**, 1541007 (2015).
  - [12] A. Pierce, K. Riles, and Y. Zhao, *Physical Review Letters* **121** (2018), 10.1103/physrevlett.121.061102.
  - [13] H.-K. Guo, K. Riles, F.-W. Yang, and Y. Zhao, “Searching for dark photon dark matter in ligo o1 data,” (2019), arXiv:1905.04316 [hep-ph].
  - [14] S. Horowitz, M. Papa, and S. Reddy, *Physics Letters B* **800**, 135072 (2020).
  - [15] C. Palomba, S. D’Antonio, P. Astone, S. Frasca, G. Intini, I. La Rosa, P. Leaci, S. Mastrogiovanni, A. Miller, F. Muciaccia, and et al., *Physical Review Letters* **123**

- (2019), 10.1103/physrevlett.123.171101.
- [16] L. Nieder, C. J. Clark, C. G. Bassa, J. Wu, A. Singh, J. Y. Donner, B. Allen, R. P. Breton, V. S. Dhillon, H.-B. Eggenstein, and et al., *The Astrophysical Journal* **883**, 42 (2019).
- [17] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt, and et al., *Physical Review D* **99** (2019), 10.1103/physrevd.99.122002.
- [18] S. Walsh, M. Pitkin, M. Oliver, S. D’Antonio, V. Dergachev, A. Królak, P. Astone, M. Bejger, M. Di Giovanni, O. Dorosh, and et al., *Physical Review D* **94** (2016), 10.1103/physrevd.94.124010.
- [19] B. Behnke, M. A. Papa, and R. Prix, *Physical Review D* **91** (2015), 10.1103/physrevd.91.064007.
- [20] M. A. Papa, H.-B. Eggenstein, S. Walsh, I. Di Palma, B. Allen, P. Astone, O. Bock, T. D. Creighton, D. Keitel, B. Machenschalk, and et al., *Physical Review D* **94** (2016), 10.1103/physrevd.94.122006.
- [21] A. Singh, M. A. Papa, H.-B. Eggenstein, and S. Walsh, *Physical Review D* **96** (2017), 10.1103/physrevd.96.082003.
- [22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” (2017), arXiv:1703.06870 [cs.CV].
- [23] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” (2015), arXiv:1505.04597 [cs.CV].
- [24] A. Huang and R. Wu, “Deep learning for music,” (2016), arXiv:1606.04930 [cs.LG].
- [25] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, “Achieving human parity in conversational speech recognition,” (2016), arXiv:1610.05256 [cs.CL].
- [26] M. Razzano and E. Cuoco, *Classical and Quantum Gravity* **35**, 095016 (2018).
- [27] D. George, H. Shen, and E. Huerta, *Physical Review D* **97** (2018), 10.1103/physrevd.97.101501.
- [28] H. Shen, D. George, E. A. Huerta, and Z. Zhao, “Denoising gravitational waves using deep learning with recurrent denoising autoencoders,” (2017), arXiv:1711.09919 [gr-qc].
- [29] W. Wei and E. Huerta, *Physics Letters B* **800**, 135081 (2020).
- [30] X. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, *Science China Physics, Mechanics & Astronomy* **62** (2019), 10.1007/s11433-018-9321-7.
- [31] H. Shen, E. A. Huerta, Z. Zhao, E. Jennings, and H. Sharma, “Deterministic and bayesian neural networks for low-latency gravitational wave parameter estimation of binary black hole mergers,” (2019), arXiv:1903.01998 [gr-qc].
- [32] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, *Physical Review D* **100** (2019), 10.1103/physrevd.100.063015.
- [33] D. George and E. Huerta, *Physical Review D* **97** (2018), 10.1103/physrevd.97.044039.
- [34] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, *Physical Review Letters* **120** (2018), 10.1103/physrevlett.120.141103.
- [35] C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix, *Physical Review D* **100** (2019), 10.1103/physrevd.100.044009.
- [36] S. Kleene, *Annals of Mathematics Studies* **34**, 3 (1956), princeton University Press.
- [37] Y. B. Ian Goodfellow and A. Courville, *Deep learning* (The MIT Press, 2015).
- [38] C. M. Burnett, Creative Commons Attribution-Share Alike 3.0 Unported license.
- [39] J. Schmidhuber, *Neural Networks* **61**, 85–117 (2015).
- [40] B. Abbott, R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, V. Adya, and et al., *Physical Review D* **96** (2017), 10.1103/physrevd.96.122004.
- [41] “Einstein@Home,” <https://www.einsteinathome.org/>.
- [42] P. O. Pinheiro, R. Collobert, and P. Dollár, “Learning to segment object candidates,” (2015), arXiv:1506.06204 [cs.CV].
- [43] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, “Instance-sensitive fully convolutional networks,” (2016), arXiv:1603.08678 [cs.CV].
- [44] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” (2013), arXiv:1311.2524 [cs.CV].
- [45] W. Abdulla, “Mask r-cnn for object detection and instance segmentation on keras and tensorflow,” [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN) (2017).
- [46] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” (2015), arXiv:1512.03385 [cs.CV].
- [47] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” (2016), arXiv:1612.03144 [cs.CV].
- [48] “Observational Relativity and Cosmology division of the MPI for Gravitational Physics Hannover,” <https://www.aei.mpg.de/obs-rel-cos>.