Denoising gravitational-wave signals from binary black holes with a dilated convolutional autoencoder

IOP Publishing Mach. Learn.: Sci. Technol. 4 (2023) 035024 https://doi.org/10.1088/2632-2153/acd90f

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
9 June 2022

REVISED
2 May 2023

ACCEPTED FOR PUBLICATION
25 May 2023

PUBLISHED
23 August 2023

PAPER

Denoising gravitational-wave signals from binary black holes with a dilated convolutional autoencoder

Philippe Bacon[1,*], **Agata Trovato**[2,3] **and Michał Bejger**[4,5]

[1] Université Paris Cité, CNRS, Astroparticule et Cosmologie, F-75013 Paris, France
[2] Dipartimento di Fisica, Università di Trieste, I-34127 Trieste, Italy
[3] INFN, Sezione di Trieste, I-34127 Trieste, Italy
[4] INFN Sezione di Ferrara, Via Saragat 1, 44122 Ferrara, Italy
[5] Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, Bartycka 18, 00-716 Warszawa, Poland
[*] Author to whom any correspondence should be addressed.

E-mail: bacon@apc.in2p3.fr

Keywords: denoising autoencoder, convolutional neural network, data analysis, gravitational waves

Abstract

The broadband frequency output of gravitational-wave (GW) detectors is a non-stationary and non-Gaussian time series data stream dominated by noise populated by local disturbances and transient artifacts, which evolve on the same timescale as the GW signals and may corrupt the astrophysical information. We study a denoising algorithm dedicated to expose the astrophysical signals by employing a convolutional neural network in the encoder-decoder configuration, i.e. apply the denoising procedure of coalescing binary black hole signals to the publicly available LIGO O1 time series strain data. The denoising convolutional autoencoder neural network is trained on a dataset of simulated astrophysical signals injected into the real detector's noise and a dataset of detector noise artifacts ('glitches'), and its fidelity is tested on real GW events from O1 and O2 LIGO-Virgo observing runs.

1. Introduction

The onset of gravitational wave (GW) astronomy began in 2015 with the first direct detection of GWs from a binary inspiral and merger of two stellar-mass black holes (BHs), an event later denoted as GW150914 [1]. Since then, the Advanced LIGO [2] and the Advanced Virgo [3] detector networks have detected many GW signals, mainly binary black holes (BBHs) but also binary neutron star (NS) inspiral events [4, 5], and BH-NS systems [6, 7]; for the list of published GW transient signal detections, see the GWTC-1 [8], GWTC-2 [9], GWTC-2.1 [10] and GWTC-3 [11] catalogs. During the latest observational campaign (LIGO-Virgo O3 run, 1st April 2019–26 March 2020), the LIGO-Virgo network was reporting GW signals at a rate of approximately once per week, with alerts on highly significant events (sky localization, type of source) released to the scientific community in the low-latency mode [12]. The detection rate will steadily increase with the increasing instrumental sensitivity, approaching one BBH detection every few days in the incoming O4 run [13]. Therefore, a rapid and reliable selection algorithm for data periods in which signals are hidden will be very useful standalone, or as part of a broader detection or data characterization framework.

Raw GW strain data are fundamentally noisy time series in which the GW signals are hidden; for the detailed description of the GW data, see [14, 15]. Classically, in order to confirm the existence of a signal in the data time-series, one has to apply a matched filtering technique (e.g. [16–18]), which is an optimal technique only when the background noise is Gaussian and stationary, and which requires substantial computational resources, and a fine grid of filter templates built from GW signal parameters, to find the match.

In this work, we study an enhancement to the established data-processing methods, and employ the machine learning (ML) approach to the GW signal processing in order to facilitate a trigger generation

process or simply be one of the first parts of a GW data-analysis pipeline. Several applications of ML in the GW astronomy low-latency data analysis exist; see e.g. [19–21] for specific applications in the context of BBH searches, as well as [22] for a recent review of ML in the GW domain. ML methods are uniquely suited to identify patterns in data, but also to perform other data processing tasks, such as *denoising*. Denoising of GW data was applied in the past using the total-variation method [23, 24], with the split Bergman regularization to obtain the total-variation regularization [25] and with dictionary learning [26]. From the deep neural network (NN) point-of-view, denoising methods were applied in [27] with the `WaveNet` implementation [28], as well as using the auto-encoder (AE) architecture [29, 30] to perform the denoising task, i.e. as a denoising auto-encoder (DAE) [31], where instead of encoding and subsequently decoding an input sequence to itself, the training consists of feeding the noisy ('corrupted') input and expecting a noiseless ('clean') output. Recent works that apply this paradigm include [32] with the long short-term memory/recurrent neural networks (LSTM/RNNs, see e.g. [33–35]) concept, and [36], using a combination of the convolutional NN (CNN, see e.g. [37, 38]) and LSTM as well as simple artificial NN [39] to denoise GW signals overlapping with instrumental glitches. An algorithm implemented in [40], based on the local polynomial approximation combined with the relative intersection of confidence intervals rule for the filter support selection is applied to denoise the GW burst signals emitted during core collapse supernovae events.

Here, we implement a purposefully simple version of the DAE, based on the one-dimensional input CNN paradigm, and apply it to the noisy ('corrupted') time series GW data containing astrophysical signals immersed in the real noise, in order to study limitations in recovering the noiseless ('clean') GW signals in this realistic setup. The CNN-DAE approach has advantages over implementations of DAE already existing in the literature, the primary being the fact that the CNN implementation is smaller and trains faster than recurrent NN. We demonstrate that a relatively small CNN DAE with a few dilated decoder layers [41] is able to train on the GW signal waveforms injected into realistic LIGO data time series and recover real GW events. We consider this type of lightweight algorithm a potentially useful trigger generator, performing the role of rapid initial classification of GW signals, and/or data characterization tasks.

This article is composed as follows. Section 2 contains a brief description of the CNN and AE methodology, description of the DAE network, as well as the training and testing data. Section 3 describes the results, obtained using both the simulated signals and real O1 and O2 events, whereas section 4 contains the conclusions.

2. Methods and training data

2.1. Convolutional AEs

A CNN is a type of NN that applies a set of convolutions (by means of the kernel filters) to the network's input [37, 38]. In the context of denoising, training a CNN consists of finding the filter weights which optimally extract the preferable features in the input data and are present in clean (noise-free) data. The filters are weighted, and their relative importance is also acquired during training. CNNs are widely used in data processing and are able to perform a wide range of tasks such as classification, pattern recognition or feature extraction. Specifically, the convolutional filter is moved across the input image to reveal important features while preserving the spatial relationship between samples, see e.g. [42, 43]. In all cases, the CNN effectively performs the role of a dimensionality reduction algorithm, with the output of the convolutional layer being a 'simplified view' of the input. In our one-dimensional context of time series data, we adopt the CNN architecture for a practical advantage of its much faster training times, as compared to more complicated LSTM/RNN architectures [38], while retaining the ability to learn temporal information encoded in the time series [44]. Specifically, we introduce dilated convolutional layers [41], with an aim to connect distant samples within the signal, and use multi-scale information contained in the time series.

2.2. Denoising AE model

The CNN layers in this work adopt the AE architecture. In general, the AE NN represents an identity function by compressing the representation of input data and then decompressing it. During training, the overall network learns its own sparse representation of the input signals. The AE is composed of two networks: an encoder g_ϕ and a decoder f_θ.

The parameters ϕ and θ denote the parameters of the encoder and the decoder, respectively. The encoder maps the original high-dimensional input into a *latent space* $\mathbf{z} = g_\phi(\mathbf{x})$, where $\mathbf{x}$ is the training or testing input. Note that we do not make an explicit use of the latent space in the present study, although in principle

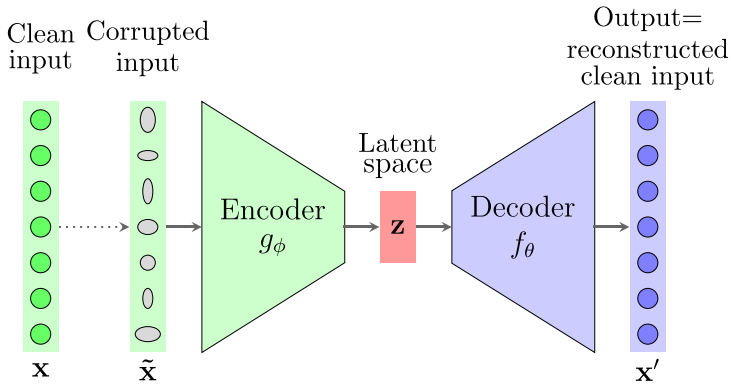

Figure 1. Data flow diagram of a DAE architecture. The clean input data **x** is corrupted (denoted by $\tilde{\mathbf{x}}$) and fed to the encoder part of the AE; for the well-trained network, the decoded output (reconstructed clean input) $\mathbf{x}'$ should be a close analogue of the clean input **x**, i.e. $\mathbf{x}' \approx \mathbf{x}$. The middle part of the DAE—the latent space—is denoted by **z**. In case of $\mathbf{x} \equiv \tilde{\mathbf{x}}$ the DAE becomes a classical AE.

it may be used for parameter estimation study, for example, as it is related to the Bayesian formulation of the problem through the variational AE (VAE) approach [45, 46] or a version of the VAE, conditioned by the actual values of signal parameters in question (conditional VAE, CVAE [47]). In contrast to the encoder, the decoder recovers the data from **z**, and successively unfolds the compressed data. The output data of the decoder (and hence the AE) is denoted $\mathbf{x}' = f_\theta(\mathbf{z})$. AE accomplishes dimensionality reduction in the same way as the principal component analysis [48] or matrix factorization algorithms [49], but the underlying process of AEs is highly non-linear and explicitly optimizes the data reconstruction.

The training of an AE consists of finding the set of parameters (θ, ϕ) which minimize the distance between **x** and $\mathbf{x}'$. This distance is properly defined by the so-called *loss function* $L_{\rm AE}(\theta, \phi)$. In the case of regression problems like the AE identity function, a popular choice is the mean square error (MSE):

$$L_{\rm AE}(\theta,\phi) = \sum_{i=1}^{N} (x_i - x_i')^2 = \sum_{i=1}^{N} (x_i - f_\theta(g_\phi(x_i)))^2 , \tag{1}$$

where N is the number of samples of input/output signals.

We adopt a one-dimensional CNN AE to denoise the generated astrophysical GW signals added to real GW detector data time series, collected during the LIGO O1 run. A DAE procedure consists in training an AE with a corrupted version of the input signal (i.e. clean signal immersed in the noisy time series), denoted by $\tilde{\mathbf{x}}$, and by demanding that the recovered output $\mathbf{x}'$ is as close to the original clean input **x** as possible. A schematic representation of a DAE is presented in figure 1. The training data set is described in detail in the subsequent section 2.3. The results of the trained DAE on simulated data in real detector's noise are described in section 3, whereas the denoising of *real* GW signals found in the O1 and O2 LIGO runs are presented in section 3.4.

The layer-by-layer structure of the NN is described in detail in table 1. The batch normalization technique [50] normalizes and scales the layer inputs in order to stabilize (prevent saturation of the non-linearities) and to speed up the training. The overall effect is also to make the network more robust to the initialization of weights. We use 32 input instances per batch. In the encoder part of the AE, the pooling layers [51] apply a non-linear down-sampling and compactify (reduce) the information of the input. In the decoder part of the AE, we introduce upsampling layers to ensure that the low dimensional information is successively unfolded. We use a rectified linear unit activation function all over the layers. As there are no strong arguments for the use of asymmetric encoder-decoder structure, we introduce three dilated convolutional layers [41] which correlate non-adjacent samples within the signal, and are able to systematically gather multi-scale information contained in the time series without losing resolution. In this respect, the final layers (CNN dilated layers) of the decoder bear resemblance to the LSTM architecture, in the sense that they register and keep track of the signal evolution in various scales. More precisely, they include a causal padding that couples signal samples that are originally far from each other. The more important the dilation rate, the wider the coupling area.

Table 1. Architecture of the DAE model. Here, $N = 2048$ denotes the number of samples in the time series signal instance provided at the input of the encoder (the number of samples corresponds to 1 s of data with a frequency sampling of 2048 Hz). The AE latent space 'bottleneck' is located between layers 8 and 9. DR stands for dilation rate. The dilated part of the decoder starts at layer 14.

Layer number	Layer type	Output shape	
1	Input	(N)	[encoder]
2	BatchNormalisation	(N)	
3	Reshape	$(N, 1)$	
4	Conv1D (128 units)	$(N, 128)$	
5	MaxPooling1D	$(N/2, 128)$	
6	Conv1D (64 units)	$(N/2, 64)$	
7	MaxPooling1D	$(N/4, 64)$	
8	Conv1D (32 units)	$(N/4, 32)$	
9	Conv1D (32 units)	$(N/4, 32)$	[decoder]
10	UpSampling1D	$(N/2, 32)$	
11	Conv1D (64 units)	$(N/2, 64)$	
12	UpSampling1D	$(N, 64)$	
13	Conv1D (128 units)	$(N, 128)$	
14	Conv1D (128 units, DR = 2)	$(N, 128)$	[dilation]
15	Conv1D (128 units, DR = 3)	$(N, 128)$	
16	Conv1D (128 units, DR = 4)	$(N, 128)$	
17	Conv1D (128 units, DR = 8)	$(N, 128)$	
18	Dense (1 unit)	$(N, 1)$	
19	Reshape	(N)	

For the DAE loss function, we chose the following MSE function:

$$L_{\mathrm{DAE}}(\theta, \phi) = \sum_{i=1}^{N} \left(x_i - f_\theta(g_\phi(\tilde{x}_i))\right)^2, \tag{2}$$

where $f_\theta(g_\phi(\tilde{\mathbf{x}})) = \mathbf{x}'$ is the DAE output, and $\mathbf{x}$ is the ground-truth (clean) signal waveform input.

2.3. Training and testing data

The input 'clean' signals used in this project are simulated astrophysical GW signals from BBHs. In general, astrophysical GW signals from close binary compact systems exhibit a characteristic increase of GW amplitude and frequency during the inspiral (the 'chirp'), followed by the merger of the binary components and the ringdown GW emission from the remnant [52]. Approximately, the GW inspiral frequency evolves as [52]

$$f_{\mathrm{GW}}^{-8/3}(t) = \frac{(8\pi)^{8/3}}{5}\left(\frac{G\mathcal{M}_{\mathrm{c}}}{c^3}\right)^{5/3}(t_{\mathrm{c}} - t) + \text{higher order corrections}, \tag{3}$$

where t_{c} is the time of coalescence, and the $\mathcal{M}_{\mathrm{c}}$ is a function of component masses M_1 and M_2, called the *chirp mass*:

$$\mathcal{M}_{\mathrm{c}} = \frac{(M_1 M_2)^{3/5}}{(M_1 + M_2)^{1/5}}. \tag{4}$$

At a given moment the GW strain amplitude h depends, approximately, on the binary system parameters as follows:

$$h(r) \propto \mathcal{M}_{\mathrm{c}}^{5/3} f_{\mathrm{GW}}^{2/3}/r. \tag{5}$$

For production runs, we assume that signal waveforms (the amplitude-frequency evolution $h(f)$ or, equivalently, $h(t)$) are well-modeled using general relativity, and for the sake of this study employ the SEOBNRv4 waveform model [53], assuming non-spinning components with masses randomly chosen from a uniform distribution in the range $M_1, M_2 \in (10,\ 30)\, M_\odot$, compatible with the current state of observational knowledge on the BBH population [8]. Sky localizations were chosen to be optimal with

respect to the antenna pattern of the detector at a given GPS time of the data segment, i.e. the source has sky coordinates directly above/below the detector, to maximize the signal-to-noise ratio (SNR). Other parameters describing the source were selected randomly: coalescence phase and polarization angle from a range of $(0, 2\pi)$, and the inclination angle from a range of $(0, \pi)$.

The 'corrupted' input is obtained by injecting these astrophysical GW signals into pre-selected time series segments from the science-quality data stream of the LIGO Livingston detector, collected during the O1 data taking campaign and released by the LIGO and Virgo Collaborations via the Gravitational Wave Open Science Center [15]. For convenience, these time series segments are packaged with a fixed length of 1 s and down-sampled from the detector's raw sampling rate of 16 384 Hz–2048 Hz (corresponding to the Nyquist frequency of 1024 Hz). They are selected to contain only 'quiet' data, i.e. rejecting periods when transient artifacts of instrumental origins (so-called 'glitches') [54, 55], as well as hardware signal injections ([56], arranged to test and calibrate the detection system) are present. The GW signals are injected into the time series segments randomly off-center (taking the signal's maximum amplitude peak as a reference), with random time shifts drawn from a uniform distribution of $(-0.1, 0.4)$ s.

GW data analysis is essentially based on matched-filtering techniques, which consist in finding the waveform template that matches the data best. Working under the assumption of the additive property of the noise (i.e. the time series d containing the GW signal h immersed in noise n is $d = n + h$), then the optimal SNR ρ, corresponding to the best matching filter (template h equals the signal) is

$$\rho = \frac{(d|h)}{\sqrt{(h|h)}}, \quad \text{where} \quad (d|h) = 4\Re \int \frac{\tilde{d}(f)\tilde{h}^*(f)}{S_n(f)} \mathrm{d}f, \tag{6}$$

with $\tilde{d}(f)$ a Fourier transform of the time series $d(t)$, and $S_n(f)$ the power spectral density (PSD) of the detector; asterisk denotes complex conjugation [18]. Detector's PSD $S_n(f)$ represents the frequency-dependent sensitivity in a broad range of frequencies and is quantified by its sensitivity curve (for details on how the PSD is computed see, e.g. [14]). From the astrophysical perspective, the SNR is a function of the waveform amplitude and, since the waveform describes the evolution of the GW amplitude, which is inversely proportional to the luminosity ('loudness') distance. While preparing the data set, we label the signal waveform with their *optimal* matched-filter SNR

$$\rho_{\mathrm{opt}} = \sqrt{(h|h)} = \sqrt{4 \int_0^\infty \frac{|\tilde{h}(f)|^2}{S_n(f)} \mathrm{d}f}, \tag{7}$$

which approximates ρ, assuming that the noise effect is negligible, $d \approx h$. The optimal matched-filter SNR ρ_{opt} is a good first-order approximation to the actual matched-filter SNR ρ in e.g. stationary Gaussian noise [57]. Subsequently, we produce a synthetic distribution of source luminosity distances such that the ρ_{opt} distribution is uniform in the range of $(5,\ 20)$ in order to consider a wide range of signals during the training. Source distances are in the range between 100 and 1000 Mpc. The adopted ρ_{opt} range is comparable to the figure-of-merit optimal SNR of 8, which correspond to a confident single-interferometer detection of an optimally-oriented compact binary inspiral [58, 59] and is also consistent with the previous LIGO-Virgo detections [8–10]. We extend the ρ_{opt} range to the lower values to study the sensitivity and robustness of our DAE implementation in the low ρ regime.

Last, but not least, to normalize the dependence of signals' strength at different frequencies we additionally perform the *whitening* of the time series data with added signals: we divide the Fourier representation of the time domain data by an estimate of the amplitude spectral density (ASD) of the noise $\sqrt{S_n(f)}$ (square root of the PSD) to ensure that the data has equal significance in each frequency bin [60]. A low-frequency cutoff at $f_{\mathrm{low}} = 30$ Hz was chosen for the simulated signals and the data, taking into account the low frequency (seismic) limit of the detectors' sensitivity. This is reflected in the example of GW strain amplitude evolution $h(t)$, immersed in the LIGO Livingston detector noise, shown in figure 2.

The network is trained during 50 epochs on 7000 data time series containing injected astrophysical signals. In addition, the training set contains 1000 time series from the O1 LIGO Livingston data when known instrumental artifacts ('glitches') are present in the data. We have used the `Gravity Spy` database [55] to obtain various common types of glitches with an estimated SNR larger than 10. All the instances in this dataset are treated with the whitening procedure and a 30 Hz high band pass filter. Mixing signals and glitches prevent any over-fitting because the network is fed with several distinct datasets which all span the

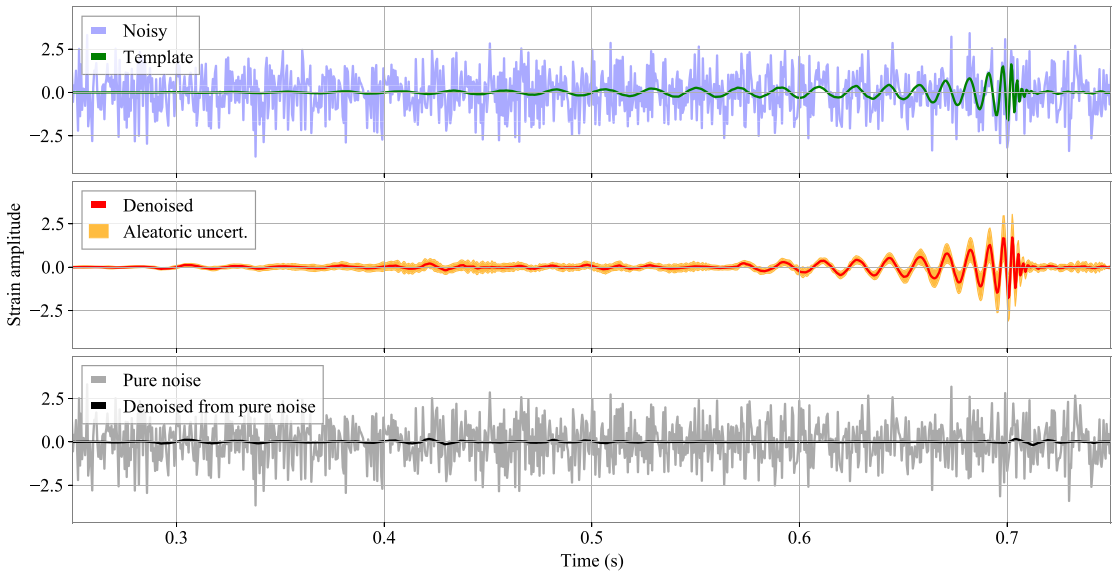

Figure 2. An example of a training/testing data time series instance, and the DAE output. Top panel: a GW signal (green) immersed in the noisy data (blue). Middle panel: denoised version of the signal (red), with the aleatoric uncertainty estimate (yellow); see section 3.3 for more details. Bottom panel: for tests, we perform a denoising procedure on the time series without a GW signal added (grey). The result is the black line, optimally with signal-to-noise equal to 0. Parameters of the GW waveform are as follows: $M_1 = 22.74\,M_\odot$, $M_2 = 17.34\,M_\odot$, distance $r = 479.27$ Mpc, corresponding to the optimal matched filter signal-to-noise ratio $\rho_{\rm opt} = 9.3$ (equation (7)). The denoised $\rho_{{\rm opt},d}$ is 10.3. Output waveform obtained from pure-noise sample (containing no GW signal) has $\rho_{{\rm opt},pn}$ of 1.5. Both the data and GW waveform are whitened by the detector's PSD, calculated using neighboring data segments; see section 3 for more details.

targeted parameter space. Additionally, we introduce randomization of the data at two levels. First, the set of training and testing datasets are split in two after randomly shuffling the initial dataset; the train-test split factor is 0.75. In addition, we randomly replace some of the training and testing astrophysical signal input data with a null signal (array of zeros) with a probability $p = 35\%$, to increase the robustness of the trained CNN to Gaussian noise fluctuations (in the case of instrumental glitches, the instances of signal are automatically set to arrays containing zero values). In training we have used the Adam optimizer [61] with a constant learning rate of 0.001 as an optimization algorithm for stochastic gradient descent calculations.

3. Results

Using the figures of merit commonly used in GW astronomy, the MSE (equation (2)), and the waveform overlap $\mathcal{O}$, which compares the original ('clean', ground truth) waveform h and the denoised waveform h^d obtained at the output of the DAE:

$$\mathcal{O} = \sqrt{\sum_{i=0}^{N} h_i h_i^d \left(\sum_{i=0}^{N} h_i h_i \right)^{-1}}, \tag{8}$$

where N is the number of points in the time series. In the evaluation, we also show the distributions of selected astrophysical parameters of the GW signals. The distribution of the SNR of denoised signals is compared to the SNR of injected signals. As a sanity check, we evaluate the DAE on time series not containing injected signals ('only noise' time series) and on several types of the `Gravity Spy` database [55] glitches.

3.1. Population of injected signals

In figure 3 we show both the estimated MSE and the recovered waveform overlap $\mathcal{O}$ as a function of the injected matched filtering SNR [18, 27], with additional information on the distance to the source (top plot) and the chirp mass (bottom panel) encoded in color, for 1000 GW signals added to detector's noise. We do not detect a correlation between the MSE and the SNR, as expected in properly executed training on the input data set with uniform distribution SNR. While we do not detect a clear correlation between the overlap and the chirp mass $\mathcal{M}_c$, a preference towards smaller SNR for large distances is seen. Signals with waveform overlap smaller than 0.75 constitute 6.6% of all the signals; 5.6% of all the signals have both $\mathcal{O} < 0.8$ and

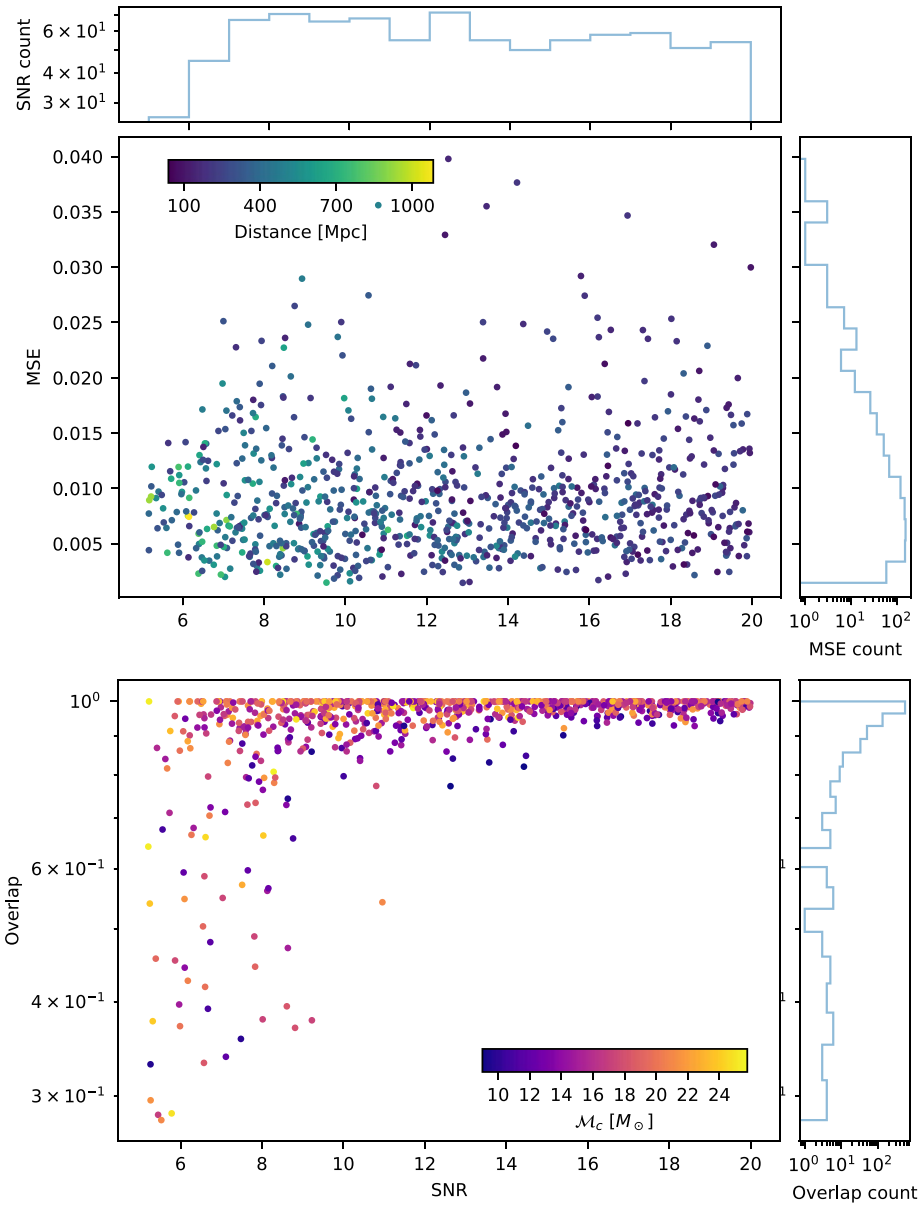

Figure 3. Upper part of the plot: mean square error (MSE, equation (2)) as a function of the signal-to-noise (SNR) resulting from for evaluating 1000 data instances with added astrophysical GW waveforms. The color of the points corresponds to the source distance (in Mpc). Side panels show histogram counts (in logarithmic scale) along both axes. Lower part of the plot: the overlap (equation (8)) between the original and the denoised GW waveforms as a function of the SNR. The color of the points corresponds to the chirp mass $\mathcal{M}_c$. The histogram count of the SNR values shows the approximately uniform distribution of the values in the dataset.

SNR < 8, so using this threshold overlap criterion we estimate that about 1% of potentially detectable signals (with SNR > 8) are incorrectly recovered by the DAE.

Figure 4 shows the comparison between the injected SNR $\rho_{\rm opt}$ and the recovered (denoised) output SNR $\rho_{{\rm out},d}$, both calculated using the optimal matched filter SNR formula (equation (7)). The color code indicates the waveforms overlap. As expected, in cases of high overlap the denoised SNR approximates quite well the injected one. The cases of lower overlap have a denoised SNR close to zero. The distribution follows the ideal $\rho_{\rm out} \equiv \rho_{{\rm out},d}$ relation with a root-mean-square of residuals of 1.9 and variance of residuals of 3.8. The denoised SNR may be used as an approximate proxy for the detection criterion : 8.2% of the output signals have $\rho_{{\rm opt},d} < 5$, whereas 1.3% of the signals have both $\rho_{\rm opt} > 8$ and $\rho_{{\rm opt},d} < 5$, i.e. are potentially strong enough to detect with the standard methods, but incorrectly recovered by the DAE. Additionally, we perform the denoising procedure on the same detector time series samples but *without* injected GW signals, to study the output signals. The distribution of output SNR in that case is depicted by the red histogram; only a few noise-only samples have $\rho_{{\rm out},d} > 5$.

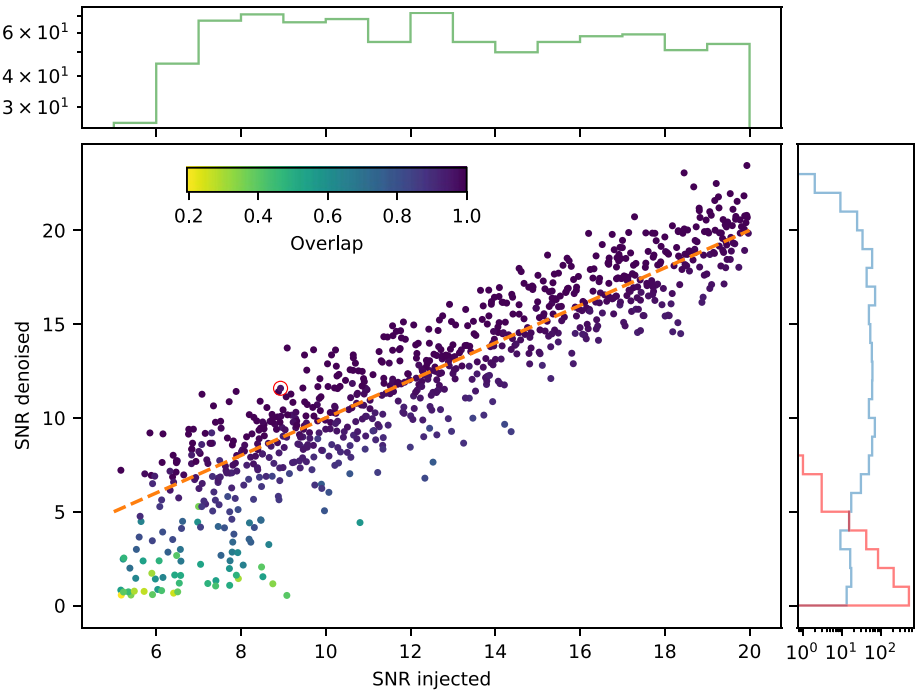

Figure 4. Denoised SNR (calculated from the DAE output analogously to equation (7), but with denoised output waveform h^d instead of the originally-injected signal h, vertical axis) as a function of the injected SNR (horizontal axis) for a testing dataset of 1000 data instances with added astrophysical GW waveforms. Points are colored by their corresponding overlap values. Orange dashed line denotes the denoised SNR equal to the injected SNR. Example waveform presented in figure 2 is denoted by a red circle. Side histograms (in logarithmic scale) show the distribution of the injected SNR (upper plot), and SNRs denoised from samples containing added GW waveforms (blue histogram), and—for comparison—not containing GW signals (i.e. pure noise, red histogram), respectively.

3.2. Evaluation on known instrumental glitches

As discussed in section 2.3, we have also used time series containing known glitches to train the network. The GPS times of these glitches have been extracted from the `Gravity Spy` database [55]. The corresponding 1 s LIGO data segments centered at these GPS times have been downloaded via the Gravitational Wave Open Science Center [15]. For the training phase, 1000 GPS times have been randomly selected among the list of glitch times available in O1 with SNR > 10 and duration < 1 s. To test the ability of the network to denoise time series that also contain glitches, we have tested the network on different glitch families, selecting some examples of recurrent glitch types, still with the conditions SNR > 10 and duration < 1 s. The nomenclature of these families comes from `Gravity Spy`. In some cases, the origin of the glitch is known, for example for the glitch type called *whistle*, which seems to be caused by signals at megahertz frequencies that beat with Voltage Controlled Oscillators in the interferometer control system [62]. The other glitch families considered for this test, *Low-Frequency Burst*, *Koi Fish* and *Blip*, have an unknown origin. A Low-Frequency Burst appears in a time-frequency spectrogram as an excess noise at low frequency. A Koi Fish is a short-duration broadband noise. A blip is a short duration noise that appears in a spectrogram as a symmetric 'teardrop' typically between 30 and 250 Hz, with the majority of the power appearing at the lowest frequencies [54].

Figure 5 contains a histogram of the recovered SNR for a group of various glitches randomly selected (blue line) and for specific glitch classes. It is visible that, even if all the selected glitches have SNR > 10, the denoised SNR is always quite small, so it can be assumed that a denoised time series with sufficiently high SNR is probably a signal of cosmic origin and not a glitch. However, this test has been done with few samples so a deeper study in this respect is needed.

3.3. Aleatoric uncertainty modelling

The results presented above are obtained with a forward pass of a noisy GW signal in our DAE model. It produces another denoised signal which has the same length as the input one. It is the result of minimizing an unweighted L2 distance between samples (equation (2)) or equivalently of maximizing a Gaussian

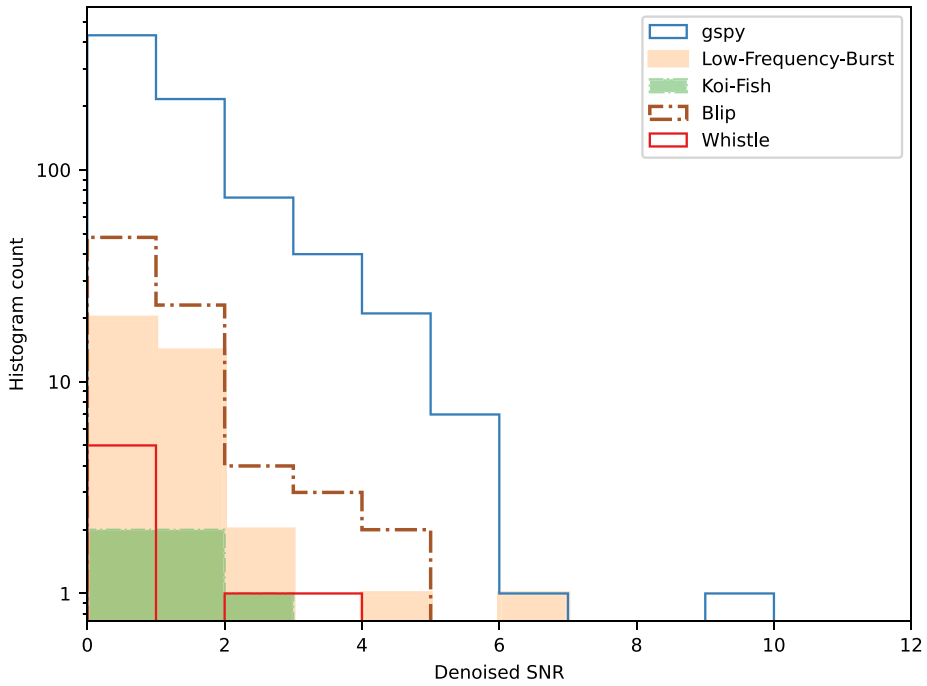

Figure 5. Evaluation of the DAE on instrumental glitches. The logarithmic vertical scale plot shows histograms of denoised output SNR for a selection of 38 Low Frequency Burst glitches, 5 Koi Fish type glitches, 80 Blips and 7 Whistle glitches. The blue line marks the evaluation of 792 assorted various types of glitches. All the glitches data are obtained from the `Gravity Spy` database [55]. The glitches have their estimated intrinsic SNR > 10.

likelihood. Obtaining a maximum likelihood estimator provides an unbiased estimator of the mean. However, this does not provide an unbiased estimator for the standard deviation.

When it comes to estimating uncertainty in a statistical model, two contributions are often distinguished. The *epistemic uncertainty* is caused by the fact that the model is not appropriate for the data. Indeed, multiple model parameters could be consistent with the observed training data. In practice, it arises in parts of the parameter space where there are fewer samples for training. The *aleatoric uncertainty* is the uncertainty arising from the stochastic nature of observed data.

A further improvement of our work is notably to enrich the predictions made by the NN model by introducing the estimation of the aleatoric uncertainty. To do so we follow the following procedure. First, the noisy signal $\tilde{x}(t)$ is fed into the DAE and it predicts a point estimate of the denoised signal $x'(t)$. We then add a random noise $n(t)$ from a standard normal distribution to $x'(t)$ in order to form a new noisy signal. Repeating the last step several time one gathers $\{\tilde{x'}(t)\}_M$, i.e. M distinct Gaussian noise realizations on top of $x'(t)$. Finally, the M noisy time series are fed to the DAE and one gets $\{\tilde{x''}\}_M$ denoised time series from which the standard deviation at each time sample is computed.

This estimation of the aleatoric uncertainty relies on two assumptions. First, the noise part of the input signal follows a Gaussian distribution. Second, the point estimate is an unbiased estimate of the mean true signal. On figures 2 and A1–B4, this mean is effectively close to the clean signal. No quantitative estimation of the closeness of the mean with the true signal has been performed.

Recent works on Bayesian deep learning like normalizing flows [63] and VAEs [46] offer a formalized statistical and probabilistic framework to properly estimate both the epistemic and aleatoric uncertainties. This implies estimating a distribution on every sample of the signal. For instance, if we assume Gaussian distributed prediction of a denoising CVAE model, this implies the inference of $\{\mu[s], \sigma[s]\}_s$ for s ranging from 0 to 2048. We leave this investigation for further work.

3.4. Real GW events

The result of the denoising of data segments containing real GW events registered during O1 and O2 are shown in appendices A and B, respectively. We study events included in the `GWTC-1-confident` catalog subset at the GWOSC [64], and in particular evaluate how well the DAE trained with the O1 LIGO Livingston data only performed on O1 and O2 data gathered by both the LIGO Livingston and LIGO Hanford instruments, see the Gravitational Wave Open Science Center for detailed information [15]. For brevity, we denote the Livingston detector by L1 and the Hanford detector by H1. The data have been whitened and high-pass filtered with a f_{low} threshold of 30 Hz, as described for the training and testing

dataset in section 2.3. The events' waveforms are taken from the data behind figure 10 of the GWTC-1 catalog paper [8], more precisely from the *lalinference* folder of [65]. The plots in appendices A and B display the aleatoric uncertainty estimate discussed in section 3.3.

3.5. Model size and training time

All key parameters of the network (batch size, learning rate, number of units per layer, activation functions) have been determined following a simple grid-based approach. In total, the final version of the DAE model consists of 1491 137 parameters (4096 non trainable parameters). The training time on 8000 data instances (7000 GW signals immersed in noise, 1000 glitch instances) is approximately 480 s on NVIDIA Tesla V100-SXM2-32GB, including reading in the training data. The evaluation of the trained DAE takes approximately 20 ms to denoise one time series instance.

4. Conclusions

We introduce a deep learning technique relying on a CNN AE network architecture to reduce Gaussian and non-Gaussian contributions from BBH GW measurements using ground-based detectors. The model is trained on a population of simulated astrophysical sources injected into real interferometric noise from O1 and O2 LIGO-Virgo observing runs (LIGO Livingston and Hanford detectors). We studied the efficiency in recovering the SNR and overall waveform from a population of injected signals, and assess the model robustness with respect to some classes of instrumental glitches. Finally, we propose and implement an aleatoric uncertainty estimation method before applying the method to real events observed during the O1 and O2 LIGO-Virgo observing runs (GW events robustly detected and confirmed by other methods).

A direct comparison with other results present in the literature is not straightforward; for example, the overlap distribution of figure 3 is similar to results of [36] in their figures 3 and 4, but their definition (their equation 4) is different than ours, in addition to the fact that the SNR range and the $\mathcal{M}_c$ ranges are also different, resulting in different training data waveforms. It is even more difficult in the case of [32] since their results in figure 2 comprise unknown GW waveforms injected into real O1 noise taken from the vicinity of the LVT151012 event, whereas we discuss a set of known O1 and O2 events.

The DAE method presented here is potentially a versatile pre-processing tool prior to detection and/or source parameter estimation pipelines used to analyze data collected by ground based instruments. An immediate step is to extend the source parameter ranges such as the individual masses or the sky localization, and apply the conditional parameter training, as in the CVAE. The approach can also be improved by (i) considering other morphologies of instrumental glitches [55] and increase the training/testing dataset size; (ii) make the loss function more complex, e.g. by adding a regularization term to penalize glitches and unwanted features [36]; (iii) include more ground based detectors in the CNN architecture and see whether it improves noise removal (note that we have not used the currently available Virgo data in the analysis because of the low SNR of recovered events); (iv) consider more sophisticated hyperparameter tuning of the network; (v) improve the uncertainties estimation by e.g. the analysis of the latent space features.

We emphasize that our aim was to obtain a denoising reconstruction method for realistic GW BBH signals with a relatively small training data set (of the order of thousands of samples), and a minimal-size NN. Vast majority of test sample GW signals are recovered well, especially the high-frequency part which is also correctly resolved in phase, making the DAE method a potentially computationally-inexpensive trigger generator working in low-latency (a pre-processing step before more expensive parameter estimation methods). Specifically, the feasibility of this application is straightforward from presented figures: figure 3, which demonstrates that the overlap (equation (8)) between the ground-truth waveform template h and denoised waveform h^d behaves properly, i.e. the algorithm recovers the sufficiently loud signals (SNR > 8), figure 4, which demonstrates that one can use the denoised waveform SNR (as defined in equation (7)) as a proxy information to decide, if the data sample contains the GW signal (in addition note that 'only noise' samples consistently return low denoised SNR), and figure 5, which shows that the same procedure evaluated on known instrumental glitches yields proper results, i.e. glitches do not trigger the algorithm as they return low denoised SNR. However, to perform a thorough performance comparison with the 'standard trigger generators' (e.g. the low-latency matched-filter searches like the `pyCBC` [66]) would require comprehensive additional studies, which are beyond the scope of current work.

The denoised output may also be utilized to provide approximate parameter estimation. In this case, one could use the denoised waveform for example to study the characteristic duration of the signal in order to infer the chirp mass of the system: it is straightforward to see from equation (3) that the time spent by the signal within the sensitive frequency range (between the low frequency cut off $f_{\rm low} \approx 30$ Hz) to the

coalescence time t_c is inversely proportional to the power of the chirp mass. Consequently, this would decrease the size of the template bank needed by the matched filter procedure, saving computing costs. The shortcomings presented on real data examples are generally related to the parameters of the GW signals being outside the training dataset parameters, and therefore possible to relieve with training tuned to specific signal parameters (taking into account known techniques to prevent catastrophic forgetting [67]), and occasional non-stationarities of the data (e.g. long duration glitches). We interpret these shortcomings as a natural outcome of the small size of the model and the training set parameter space. Improvements in the uncertainties estimation by the DAE model would additionally strengthen its role as a pre-processing step in the general parameter estimation.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The authors would like to acknowledge the European COST action G2Net (CA17137), Tom Charnock and Florian Führer for fruitful discussions, and Éric Chassande-Mottin for his contribution on the dataset preparation. This research was supported in part by the PLGrid infrastructure with the computing Grant on the ACK Cyfronet AGH `Prometheus` cluster, the Grant of the Polish Ministry of Science and Higher Education (MNiST) for the expansion of the IT infrastructure at the Nicolaus Copernicus Astronomical Center, and the Polish National Science Centre Grant Nos. 2016/22/E/ST9/00037 and 2021/43/B/ST9/01714.

Research presented here has made use of data or software obtained from the Gravitational Wave Open Science Center (gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society, and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain. The construction and operation of KAGRA are funded by Ministry of Education, Culture, Sports, Science and Technology (MEXT), and Japan Society for the Promotion of Science (JSPS), National Research Foundation (NRF) and Ministry of Science and ICT (MSIT) in Korea, Academia Sinica (AS) and the Ministry of Science and Technology (MoST) in Taiwan.

This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation.

The code and data were prepared in `python v3.8`, using `Tensorflow/Keras v2.2` with the support of GPU (`CUDA` toolkit v10.1), and the GW libraries `gwpy` v2.0.2 [68] and `pyCBC` v1.17 [66]. We acknowledge the use of `matplotlib` v3.3.3 [69].

Appendix A. Real O1 GW events

In all the plots, the L1 and H1 data are depicted with a blue line. The green line represents the whitened waveform recovered by lalinference (ML waveform) and taken from [65]. The red line is the result of the denoising and the yellow band represents the aleatoric uncertainty calculation. The caption of each plot recalls the estimated masses and the single-detector optimal SNRs from parameter-estimation analyses made by the LVC Collaboration in [8]. Below we summarize the reconstructed events, and discuss the reasons of some cases of unsatisfactory reconstructions.

For the GW150914 event in figure A1, both the amplitude and phase are very well reconstructed in L1 and this despite a clear non-Gaussian contribution at the time of the event. In H1, the phase is well reconstructed while there is a loss of amplitude in the part of the waveform just before the merger.

For the GW151012 event in figure A2, the poor reconstruction may be understood by the magnitude of the single detector optimal SNR found in the parameter-estimation analyses: $6.4^{+1.3}_{-1.3}$ for H1 and $5.8^{+1.2}_{-1.2}$ for

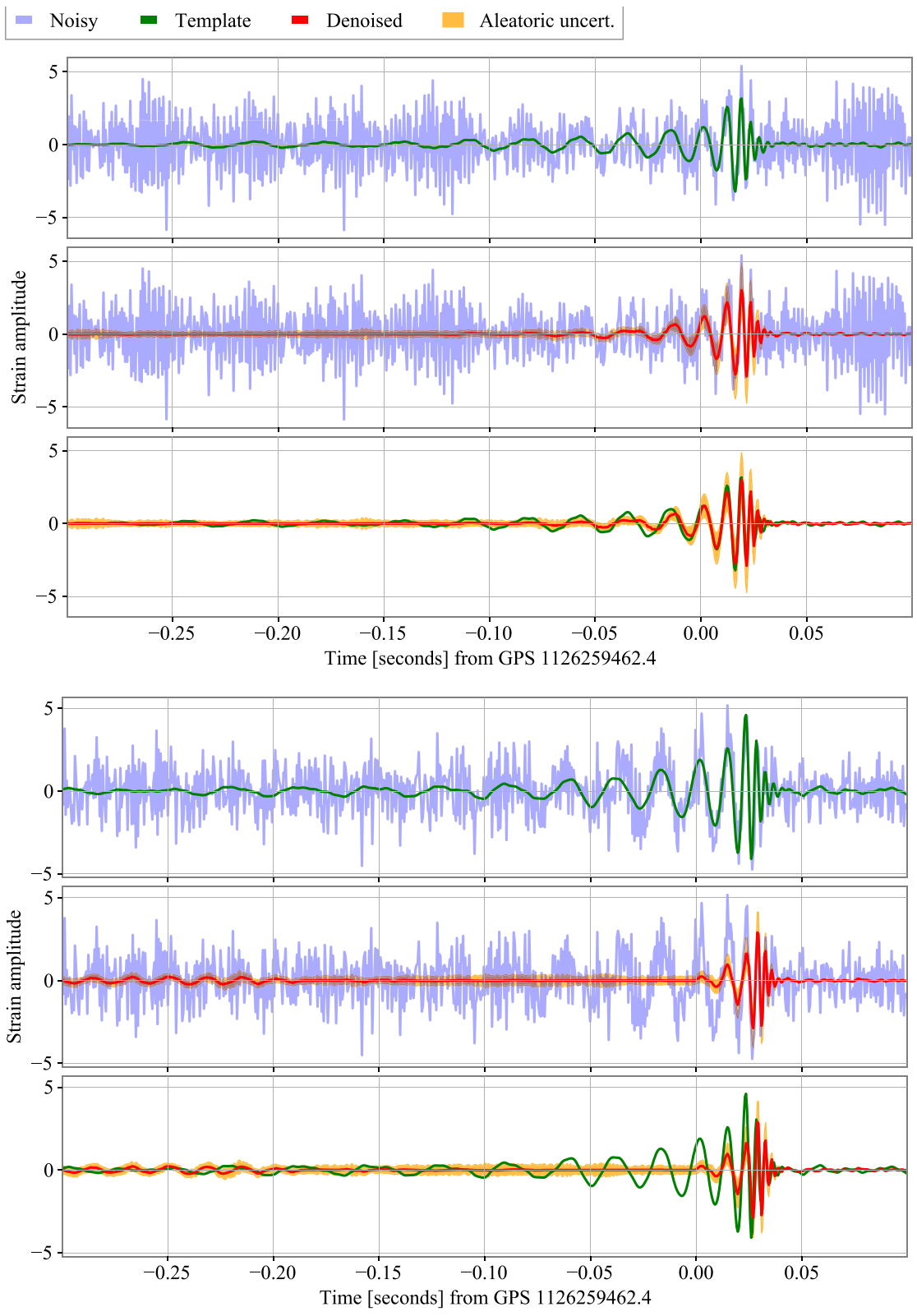

Figure A1. Denoising applied to the O1 data GW150914 event for the L1 detector (3 top panels) and the H1 detector (3 bottom panels). The component masses are $35.6^{+4.7}_{-3.1}\,M_{\odot}$ and $30.6^{+3.0}_{-4.4}\,M_{\odot}$, and the single-detector optimal SNRs are $14.2^{+1.6}_{-1.4}$ for L1 and $20.6^{+1.6}_{-1.6}$ for H1.

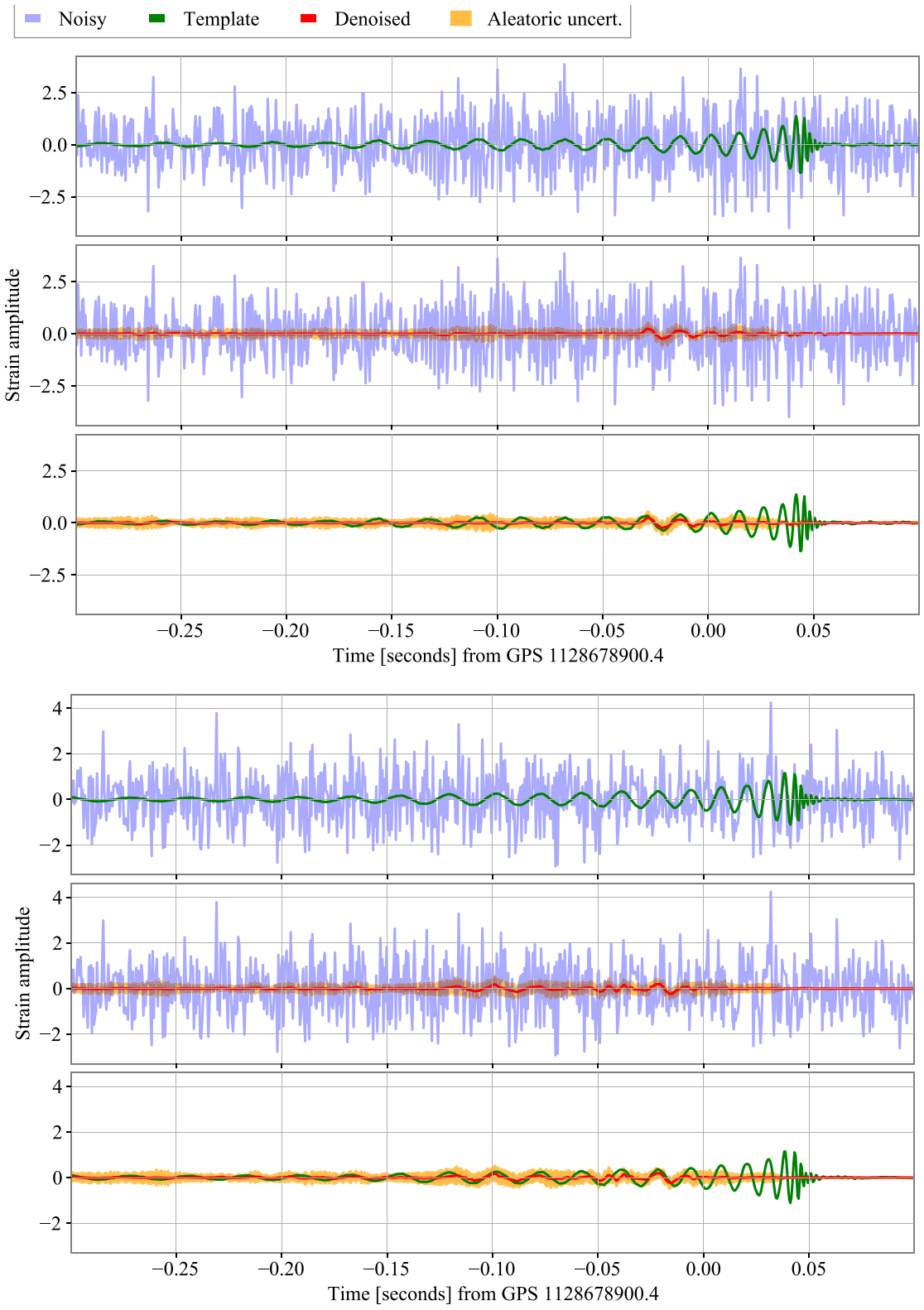

Figure A2. Denoising applied to the O1 data GW151012 event for the L1 detector (3 top panels) and the H1 detector (3 bottom panels). The component masses are $23.2^{+14.9}_{-5.5}\ M_\odot$ and $13.6^{+4.1}_{-4.8}\ M_\odot$, and the single-detector optimal SNRs are $5.8^{+1.2}_{-1.2}$ for L1 and $6.4^{+1.3}_{-1.3}$ for H1.

L1 [8]. By consulting these values with figure 3 we conclude that the event lies in an area where the overlap is usually not sufficient for satisfactory denoising.

For the GW151226 event in figure A3, the phase of the signal is well recovered in many parts of the waveform, while the amplitude is almost everywhere underestimated, except for the first and last portion of

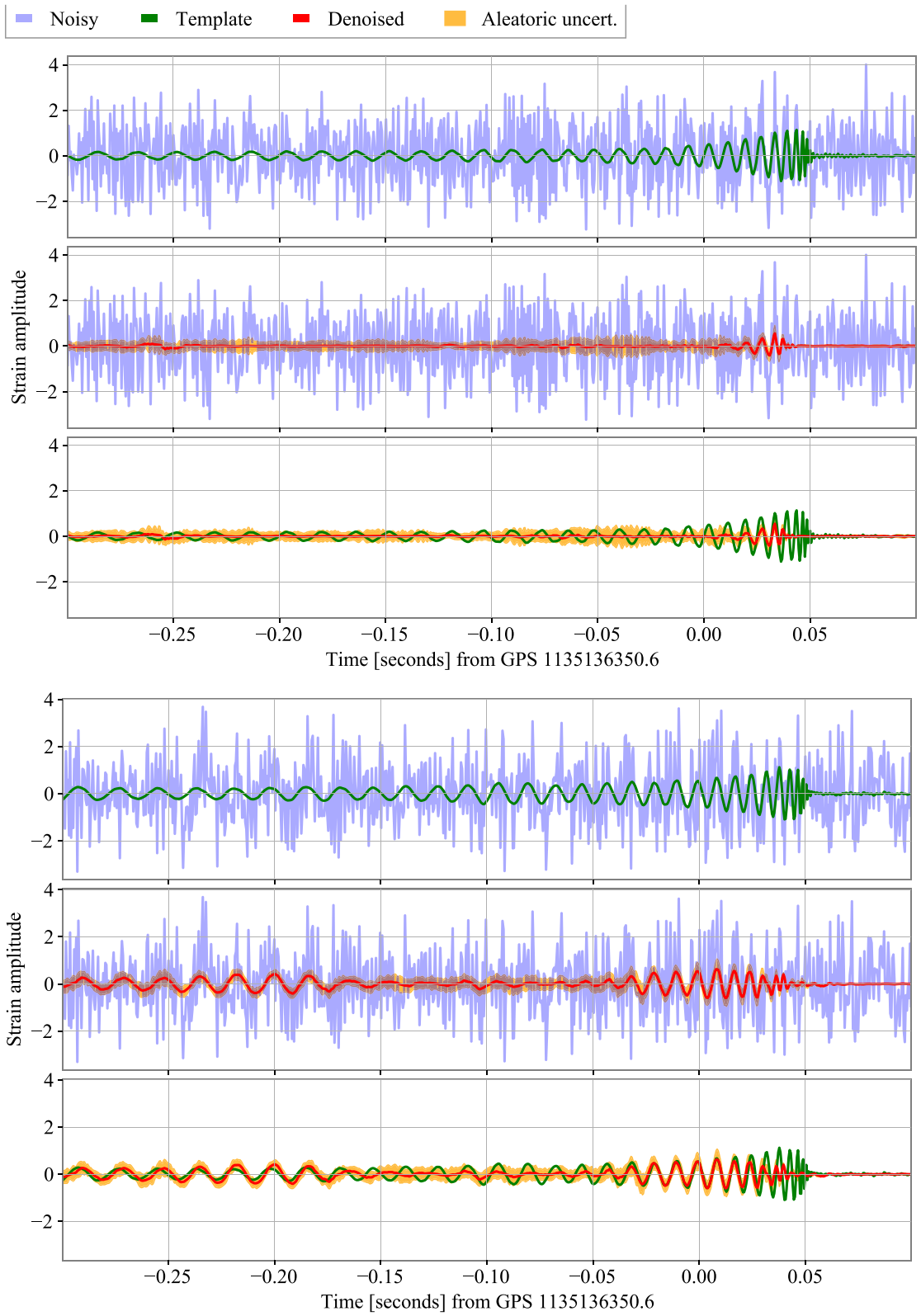

Figure A3. Denoising applied to the O1 data GW151226 event for the L1 detector (3 top panels) and the H1 detector (3 bottom panels). The component masses are $13.7^{+8.8}_{-3.2}\,M_\odot$ and $7.7^{+2.2}_{-2.5}\,M_\odot$, and the single-detector optimal SNRs are $6.9^{+1.2}_{-1.1}$ for L1 and $9.8^{+1.5}_{-1.4}$ for H1.

the event in H1, for which the reconstruction worked. We associate the relatively poor quality of this result with the lighter mass of one of the BH system components, in comparison to the values used in the training. Component mass estimates are $13.7^{+8.8}_{-3.2}$ and $7.7^{+2.2}_{-2.5}\,M_\odot$ [8], whereas our training set lower mass was $10\,M_\odot$. Lower component masses make the GW signal last longer in time in the sensitive frequency range of the

detectors; the DAE failed to generalize to a duration of the GW signal longer than exposed to during the training.

Appendix B. Real O2 GW events

The DAE model was also tested on real events from the O2 run; we show selected interesting results in figures analogous to the appendix A, which are mostly cases of non-optimal denoising. As before, we discuss reasons behind these results. Despite the DAE model was trained on the O1 L1 data only, in case of the louder O2 events whose masses are in the range used for the training, namely the GW170104, GW170809, GW180814, GW170818 events, the waveforms are reconstructed.

For the GW170104 event, figure B1 displays such an example of a clean denoising in both L1 and H1 detectors. Several ($\sim$10 cycles) high frequency cycles are well-recovered. In general, this event is a good example of the low-frequency part of the signal having too small amplitude and therefore too small SNR to be correctly denoised.

For the GW170823 event in figure B2, the phase of the original GW signal is well-recovered while an additional 'ghost' GW signal is inferred from our DAE model (in between -0.10 s and -0.05 s) in H1. Note that these spurious cycles do not belong to the original GW signal, i.e. it is not that the clean signal cycles are somewhere well and poorly recovered as it is the case in the previous figure. In this case, the low single detector SNR of this event in H1 (SNR is $6.8^{+1.4}_{-1.2}$ in Hanford—see table 5 in [8]) explains the poor reconstruction. This is coherent with results presented on figure 4. We interpret the 'ghost' signal reconstruction as false-positive reaction of the DAE model on a possible non-Gaussian transient feature in the real data. The issue of deceiving the trained model with specifically-crafted input data is beyond the scope of the current work, but it is a possible future development direction.

For the GW170608 event in figure B3, the DAE model manages to retrieve some cycles in the L1 data, although phase information is not recovered completely. A clearly visible low-frequency glitch in the H1 data prevents the GW signal to be correctly denoised. However, in figure B4 we perform an experiment with changing the low-pass filter value $f_{\rm low}$, from 30 Hz assumed at training to 50 Hz. In this case the DAE model performs relatively well, although it was trained on the $f_{\rm low} = 30$ Hz data, and despite the fact that the source mass parameters lie outside the training dataset parameter space used for training.

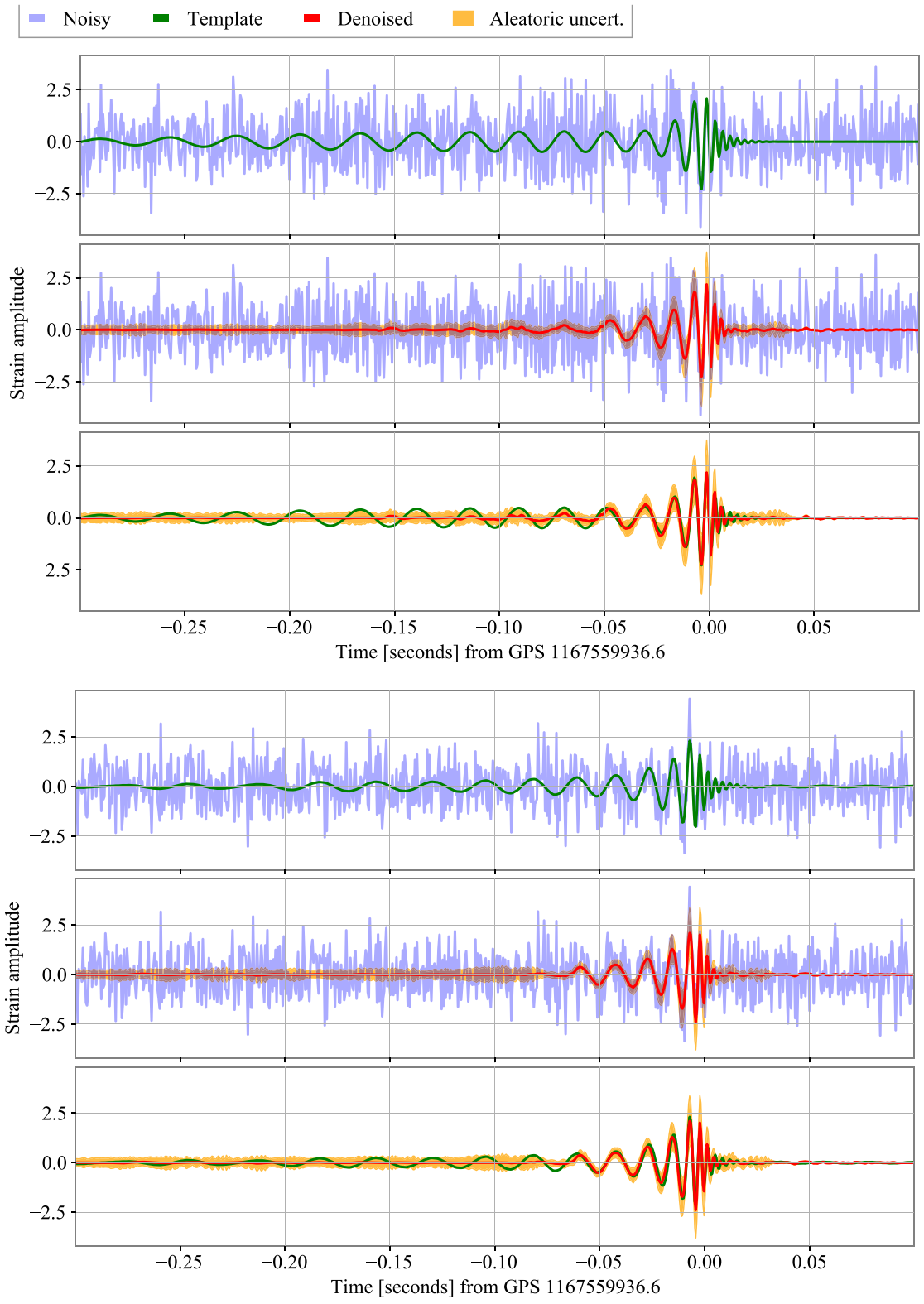

Figure B1. Denoising applied to the O2 data GW170104 event for the L1 detector (3 top panels) and the H1 detector (3 bottom panels). The component masses are $30.8^{+7.3}_{-5.6}\,M_\odot$ and $20.0^{+4.9}_{-4.6}\,M_\odot$, and the single-detector optimal SNRs are $9.9^{+1.5}_{-1.3}$ for L1 and $9.5^{+1.3}_{-1.6}$ for H1.

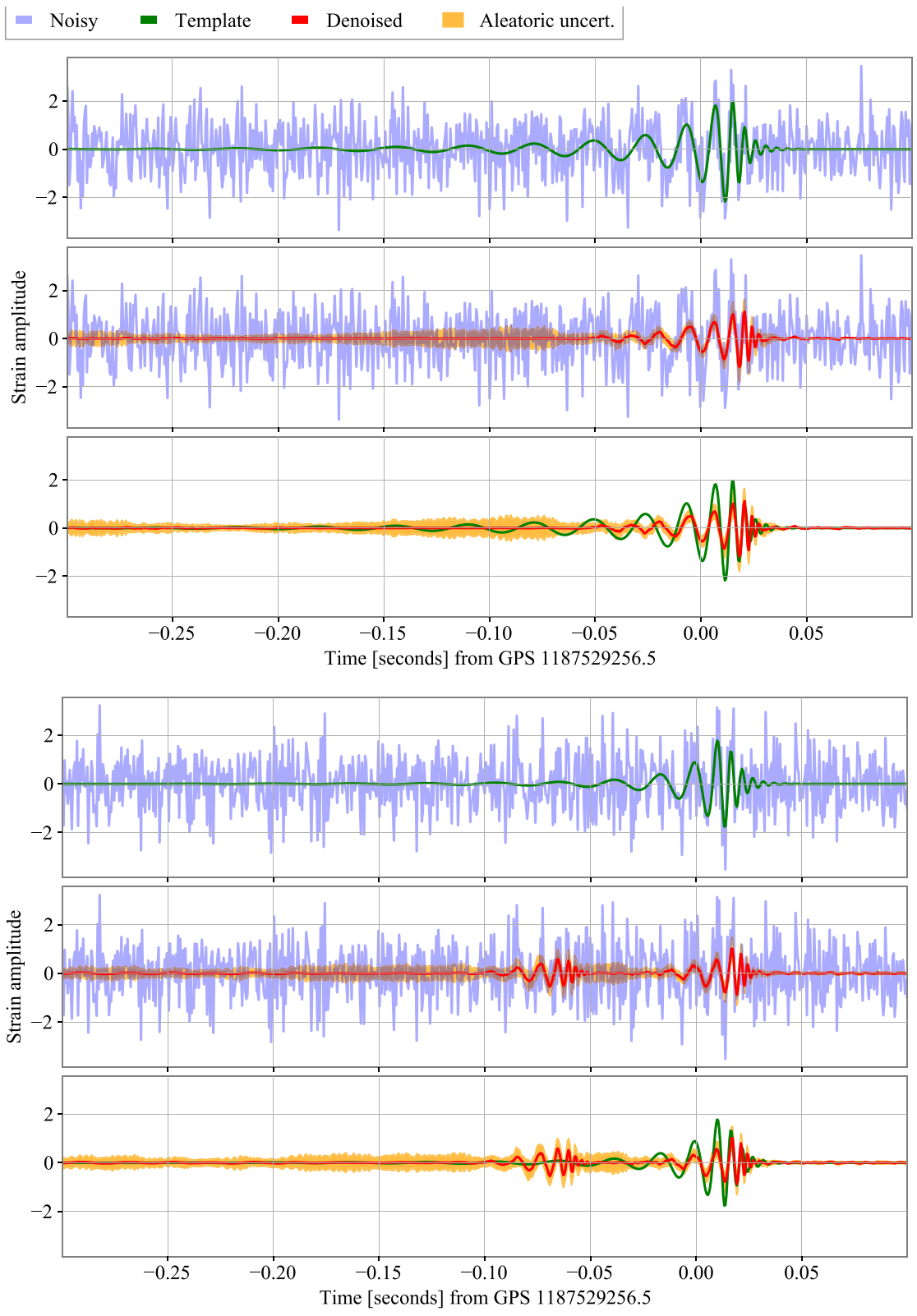

Figure B2. Denoising applied to the O2 data GW170823 event for the L1 detector (3 top panels) and the H1 detector (3 bottom panels). The component masses are $39.5^{+11.2}_{-6.7}\ M_\odot$ and $29.0^{+6.7}_{-7.8}\ M_\odot$, and the single-detector optimal SNRs are $9.2^{+1.7}_{-1.5}$ for L1 and $6.8^{+1.4}_{-1.2}$ for H1.

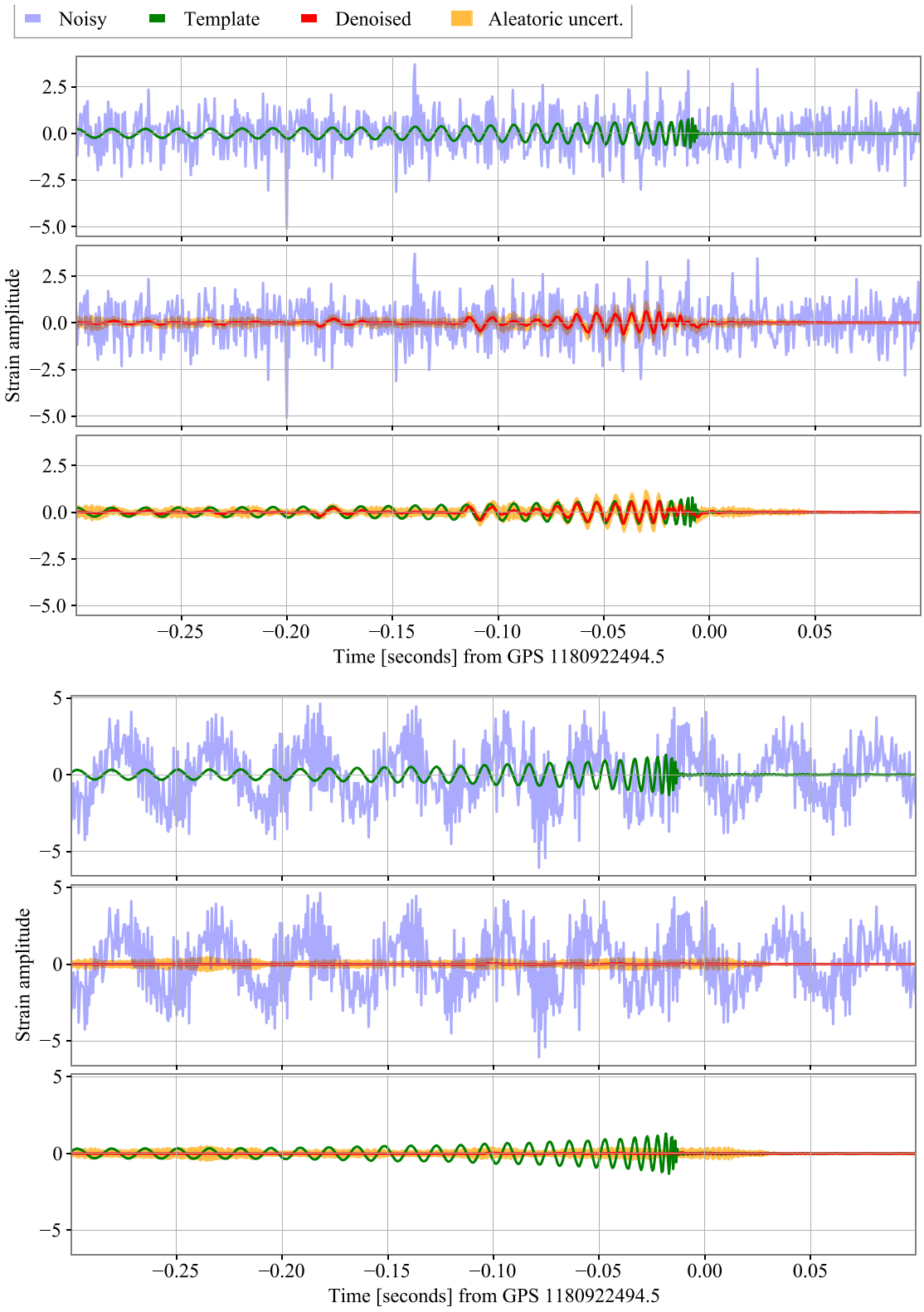

Figure B3. Denoising applied to the O2 data GW170608 event for the L1 detector (3 top panels) and the H1 detector (3 bottom panels). The component masses are $11.0^{+5.5}_{-1.7}\,M_\odot$ and $7.6^{+1.4}_{-2.2}\,M_\odot$, and the single-detector optimal SNRs are $9.2^{+1.5}_{-1.2}$ for L1 and $12.1^{+1.6}_{-1.6}$ for H1.

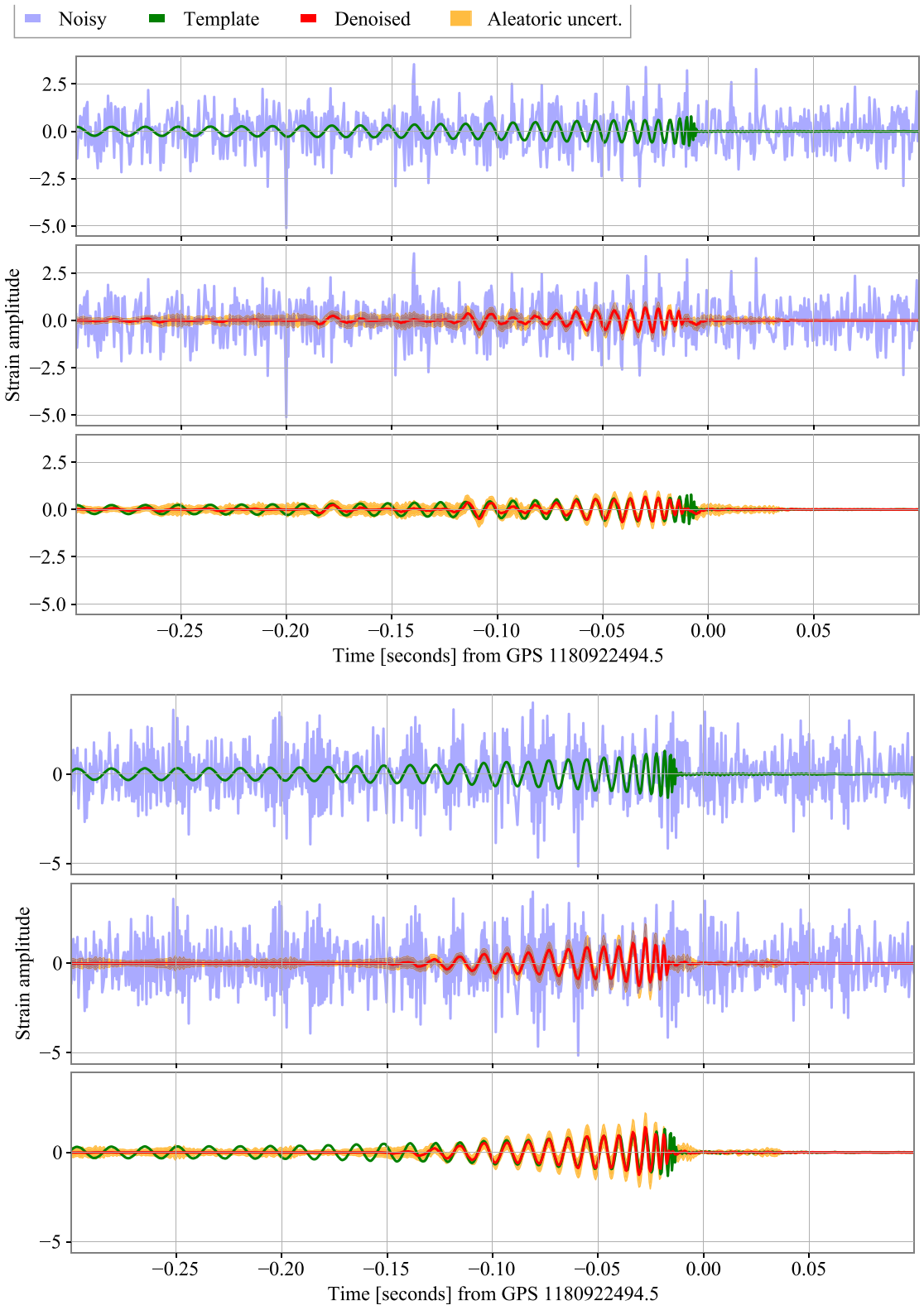

Figure B4. Denoising applied to the O2 data GW170608 event for the L1 detector (3 top panels) and the H1 detector (3 bottom panels). While for the other plots the high-pass filter was set to 30 Hz (as in the training set), in this case we apply a high pass at 50 Hz to the original data before the denoising.

ORCID iDs

Philippe Bacon https://orcid.org/0000-0003-1350-2037
Michał Bejger https://orcid.org/0000-0002-4991-8213

References

[1] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2016 *Phys. Rev. Lett.* **116** 061102
[2] Aasi J *et al* 2015 *Class. Quantum Grav.* **32** 074001
[3] Acernese F *et al* 2015 *Class. Quantum Grav.* **32** 024001
[4] Abbott B P *et al* 2017 *Phys. Rev. Lett.* **119** 161101
[5] Abbott B P *et al* 2020 *Astrophys. J.* **892** L3
[6] Abbott R *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2020 *Astrophys. J.* **896** L44
[7] Abbott R *et al* 2021 *Astrophys. J. Lett.* **915** L5
[8] Abbott B P *et al* 2019 *Phys. Rev.* X **9** 031040
[9] Abbott R *et al* 2021 *Phys. Rev.* X **11** 021053
[10] Abbott R *et al* 2021 arXiv:2108.01045
[11] Abbott R *et al* 2021 arXiv:2111.03606
[12] GraceDB – gravitational-wave candidate event database 2020 (available at: https://gracedb.ligo.org/superevents/public/O3) (Accessed 1 May 2020)
[13] Abbott B P *et al* 2018 *Living Rev. Relativ.* **21** 3
[14] Abbott B P *et al* 2020 *Class. Quantum Grav.* **37** 055002
[15] Abbott R *et al* 2021 *SoftwareX* **13** 100658
[16] Wiener N 1949 *Extrapolation, Interpolation and Smoothing of Stationary Time Series* (Wiley)
[17] Sathyaprakash B S and Dhurandhar S V 1991 *Phys. Rev.* D **44** 3819–34
[18] Owen B J and Sathyaprakash B S 1999 *Phys. Rev.* D **60** 022002
[19] Huerta E A *et al* 2019 *Nat. Rev. Phys.* **1** 600–8
[20] Gabbard H, Williams M, Hayes F and Messenger C 2018 *Phys. Rev. Lett.* **120** 141103
[21] George D and Huerta E A 2018 *Phys. Lett.* B **778** 64–70
[22] Cuoco E *et al* 2020 *Mach. Learn.: Sci. Technol.* **2** 011002
[23] Torres A, Marquina A, Font J A, Ibáñez J M and Ibáñez J M 2014 *Phys. Rev.* D **90** 084029
[24] Torres-Forné A, Cuoco E, Marquina A, Font J A and Ibáñez J M 2018 *Phys. Rev.* D **98** 084013
[25] Torres A, Marquina A, Font J A and Ibáñez J M 2015 Split Bregman method for gravitational wave denoising *Gravitational Wave Astrophysics* vol 40 (Springer) p 289
[26] Torres-Forné A, Marquina A, Font J A and Ibáñez J M 2016 *Phys. Rev.* D **94** 124040
[27] Wei W and Huerta E A 2020 *Phys. Lett.* B **800** 135081
[28] van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A and Kavukcuoglu K 2016 arXiv:1609.03499
[29] Shen H, Zhao Z, George D and Huerta E 2018 Denoising gravitational waves using deep learning with recurrent denoising autoencoders *APS April Meeting Abstracts* (*APS Meeting Abstracts* vol 2018) p S14.008
[30] Shen H, George D, Huerta E A and Zhao Z 2019 arXiv:1903.03105
[31] Hinton G E and Salakhutdinov R R 2006 *Science* **313** 504–7
[32] Shen H, George D, Huerta E A and Zhao Z 2017 arXiv:1711.09919
[33] Jain L C and Medsker L R 1999 *Recurrent Neural Networks: Design and Applications* 1st edn (CRC Press, Inc.)
[34] Pascanu R, Gulcehre C, Cho K and Bengio Y 2013 arXiv:1312.6026
[35] Hochreiter S and Schmidhuber J 1997 *Neural Comput.* **9** 1735–80
[36] Chatterjee C, Wen L, Diakogiannis F and Vinsen K 2021 *Phys. Rev.* D **104** 064046
[37] Gu J *et al* 2018 *Pattern Recognit.* **77** 354–77
[38] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (The MIT Press)
[39] Mogushi K, Quitzow-James R, Cavaglià M, Kulkarni S and Hayes F 2021 *Mach. Learn.: Sci. Technol.* **2** 035018
[40] Lopac N, Lerga J and Cuoco E 2020 *Sensors* **20** 6920
[41] Yu F and Koltun V 2016 Multi-scale context aggregation by dilated convolutions (arXiv:1511.07122)
[42] Dhillon A and Verma G 2019 *Prog. Artif. Intell.* **9** 85–112
[43] Yao G, Lei T and Zhong J 2019 *Pattern Recognit. Lett.* **118** 14–22
[44] Ismail Fawaz H, Forestier G, Weber J, Idoumghar L and Muller P A 2018 arXiv:1809.04356
[45] Kingma D P and Welling M 2013 arXiv:1312.6114
[46] Kingma D P and Welling M 2019 *Found. Trends Mach. Learn.* **12** 307–92
[47] Sohn K, Lee H and Yan X 2015 Learning structured output representation using deep conditional generative models *Advances in Neural Information Processing Systems* vol 28, ed C Cortes, N Lawrence, D Lee, M Sugiyama and R Garnett (Curran Associates, Inc.)
[48] Jolliffe I 2011 *Principal Component Analysis* (Springer) pp 1094–6
[49] Lee D D and Seung H S 2000 Algorithms for non-negative matrix factorization *Proc. 13th Int. Conf. on Neural Information Processing Systems (NIPS 2000)* (MIT Press) pp 535–41
[50] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift (arXiv:1502.03167)
[51] Scherer D, Müller A and Behnke S 2010 Evaluation of pooling operations in convolutional architectures for object recognition *Artificial Neural Networks (ICANN 2010)* pp 92–101
[52] Abbott B P *et al* 2017 *Ann. Phys., Lpz.* **529** 1600209
[53] Bohé A *et al* 2017 *Phys. Rev.* D **95** 044028
[54] Abbott B P *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2016 *Class. Quantum Grav.* **33** 134001
[55] Zevin M *et al* 2017 *Class. Quantum Grav.* **34** 064003
[56] Biwer C *et al* 2017 *Phys. Rev.* D **95** 062002
[57] Jaranowski P and Królak A 2012 *Living Rev. Relativ.* **15** 4

[58] Finn L S and Chernoff D F 1993 *Phys. Rev.* D **47** 2198–219
[59] Abadie J *et al* 2010 *Class. Quantum Grav.* **27** 173001
[60] Abbott B P *et al* 2020 *Class. Quantum Grav.* **37** 055002
[61] Kingma D P and Ba J 2014 arXiv:1412.6980
[62] Nuttall L K *et al* 2015 *Class. Quantum Grav.* **32** 245005
[63] Rezende D J and Mohamed S 2016 Variational inference with normalizing flows (arXiv:1505.05770)
[64] Gravitational wave open science center - gwtc-1 confident detection event list 2022 (available at: www.gw-openscience.org/eventapi/html/GWTC-1-confident/) (Accessed 26 January 2022)
[65] Abbott B P *et al* 2018 GWTC-1: figure 10 (available at: https://dcc.ligo.org/LIGO-P1800376)
[66] Nitz A *et al* 2020 gwastro/pycbc: PyCBC Release (1.17.0)
[67] Kirkpatrick J *et al* 2017 *Proc. Natl Acad. Sci.* **114** 3521–6
[68] Macleod D M *et al* 2021 *SoftwareX* **13** 100657
[69] Hunter J D 2007 *Comput. Sci. Eng.* **9** 90–95