Demonstration of Machine Learning-assisted real-time noise regression in gravitational wave detectors

Muhammed Saleem,[1, *] Alec Gunny,[2, 3] Chia-Jui Chou,[4] Li-Cheng Yang,[4] Shu-Wei Yeh,[5]
Andy H. Y. Chen,[6] Ryan Magee,[7] William Benoit,[1] Tri Nguyen,[2, 3] Pinchen Fan,[2, 3]
Deep Chatterjee,[2, 3] Ethan Marx,[2, 3] Eric Moreno,[2, 3] Rafia Omer,[1] Ryan Raikman,[2, 3] Dylan
Rankin,[2, 3] Ritwik Sharma,[8] Michael Coughlin,[1] Philip Harris,[2, 3] and Erik Katsavounidis[2, 3]

[1]*School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA*
[2]*Department of Physics, MIT, Cambridge, MA 02139, USA*
[3]*LIGO Laboratory, 185 Albany St, MIT, Cambridge, MA 02139, USA*
[4]*Department of Electrophysics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan*
[5]*Department of Physics, National Tsing Hua University, Hsinchu, Taiwan*
[6]*Institute of Physics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan*
[7]*LIGO Laboratory, California Institute of Technology, Pasadena, CA 91125, USA*
[8]*Department of Physics, Deshbandhu College, University of Delhi, New Delhi, India*
(Dated: June 21, 2023)

Real-time noise regression algorithms are crucial for maximizing the science outcomes of the LIGO, Virgo, and KAGRA gravitational-wave detectors. This includes improvements in the detectability, source localization and pre-merger detectability of signals thereby enabling rapid multi-messenger follow-up. In this paper, we demonstrate the effectiveness of *DeepClean*, a convolutional neural network architecture that uses witness sensors to estimate and subtract non-linear and non-stationary noise from gravitational-wave strain data. Our study uses LIGO data from the third observing run with injected compact binary signals. As a demonstration, we use *DeepClean* to subtract the noise at 60 Hz due to the power mains and their sidebands arising from non-linear coupling with other instrumental noise sources. Our parameter estimation study on the injected signals shows that *DeepClean* does not do any harm to the underlying astrophysical signals in the data while it can enhances the signal-to-noise ratio of potential signals. We show that *DeepClean* can be used for low-latency noise regression to produce cleaned output data at latencies $\sim 1-2$ s. We also discuss various considerations that may be made while training *DeepClean* for low latency applications.

I. INTRODUCTION

The current network of ground-based laser interferometers, consisting of advanced LIGO [1, 2] and advanced Virgo [3] have facilitated the detection of approximately one hundred gravitational wave (GW) events from coalescing compact binaries consisting of neutron stars and/or black holes [4–6]. In the third observing run (referred to as O3), LIGO Livingston (L1), LIGO Hanford (H1) and Virgo (V1) had a sensitive median range for detecting binary neutron stars (BNS) of approximately 133 Mpc, 115 Mpc and 51 Mpc, respectively [4]. The fourth observing run, referred to as O4, has been officially started in May 2023, with recent technological upgrades and the addition of a fourth detector, KAGRA (K1) [7, 8]. The range of the anticipated detection of binary neutron stars in O4 is expected to be 160-190 Mpc for Advanced LIGO, 90-120 Mpc for Advanced Virgo, and 1-10 Mpc for KAGRA [9].

Upgrades in technology have improved the sensitivity of interferometers by reducing fundamental noise sources such as thermal and quantum fluctuations [10, 11]. However, environmental and instrumental processes also contribute to the noise in the interferometer strain. The presence of such noise can reduce the sensitivity of the detectors to astrophysical transient signals [12, 13], in particular, sources without well-known theoretical models (e.g., supernovae) [14]. Noise regression methods are used to remove these contaminants, typically by identifying their origin [15]. Gravitational-wave interferometers are equipped with auxiliary *witness* sensors or channels to independently monitor these processes in addition to the strain channel [16]. Identifying the couplings that exist between these witness channels is the key in estimating their contribution to the strain and removing them. However, there are thousands of witness channels tracking different noise sources, and non-linear couplings between them may result in noise that is challenging to identify using standard filtering techniques such as Wiener filtering [17–20].

The developments of machine learning neural networks have significantly enhanced our capability of noise regression in the interferometer strain data. This includes the recent deep learning algorithms that are developed to subtract non-linear and non-stationary couplings originating from instrumental and environmental sources [21–26]. These algorithms have successfully removed noise couplings such as the 60 Hz power-line noise and their sidebands, which arise from the non-linear coupling of the strain with instrumental control systems. However, these deep learning noise regression algorithms have thus far been demonstrated primarily in high latency, or *of-*

*Electronic address: mcholayi@umn.edu

fline, analysis scenarios, where time-series data of several hours are analyzed long after they were originally recorded.

Multi-messenger astronomy, where gravitational-wave sources are followed up for their counterparts in the electromagnetic spectrum and neutrinos, is one of the most promising aspects of gravitational-wave observations [27, 28]. Detecting electromagnetic counterparts that fade quickly after the gravitational-wave detection, such as γ-ray bursts and x-rays from binary neutron star mergers, requires sending out low-latency alerts to trigger follow-up observations across electromagnetic frequencies [29–34]. Ground-based detectors are still below their designed sensitivities at lower frequency ranges (below 60 Hz) [35, 36], indicating the potential for substantial improvements in the capability of sending pre-merger (or early-warning) alerts by performing low-latency noise regression at low frequencies. Even incremental improvements in the sensitive distances can lead to significant improvements in the number of detections, which scale as the cube of the distance (proportional to the volume). These improvements could result in the detection of compact binary mergers that would not have otherwise been identified at low latency.

Performing low-latency (*a.k.a* online) noise regression poses significant computational challenges compared to offline regression. To not become the dominant source of latency in the release of alerts, a low-latency noise regression should produce cleaned strain with the overall delay not more than a couple of seconds. Ref. [37] discussed in detail how to meet such computational demands in low latency, by employing the *as-a-service* computing paradigm [38, 39] into the context of gravitational-wave data analysis, in order to leverage hardware accelerators (such as GPUs) and other heterogeneous computing resources.

In this paper, we demonstrate and validate the application of *DeepClean* [25] infrastructure on gravitational wave strain data from LIGO Hanford and LIGO Livingston. *DeepClean* is a deep learning convolutional neural network algorithm for noise regression in gravitational-wave strain. *DeepClean* targets those noise that are environmental or technical[1] in origin and can be tracked independently with witness sensors. We perform a mock data challenge (MDC) to demonstrate the effectiveness of *DeepClean* as a production pipeline for low-latency and high-latency noise regression applications.

This paper is organized as follows: Section II provides a concise overview of the DeepClean architecture and the end-to-end infrastructure. Section III presents the details of our mock data challenge. In section IV, we delve into the application of *DeepClean* on our mock

---

[1] Technical noise, *a.k.a* control noise, usually refers to the noise generated by the apparatus that control the optics in the interferometer

data and present the corresponding performance metrics. Section V demonstrates the validation tests performed using astrophysically motivated metrics, including detection and parameter estimation. Section VI focuses on the feasibility of utilizing *DeepClean* for low-latency noise regression. Finally, section VII concludes the study and discusses future prospects.

II. THE *DEEPCLEAN* INFRASTRUCTURE

The *DeepClean* architecture has been described in detail in [25]. In this section, we provide a brief overview of the algorithm. The strain readout from an interferometer, $h(t)$, can be represented as the sum of a possible astrophysical signal $s(t)$ and the detector noise $n(t)$, such that $h(t) = s(t) + n(t)$.

The goal of *DeepClean* is to minimize the noise $n(t)$ to enable the detection of the astrophysical signal at the highest possible signal-to-noise ratio (SNR). While some noise sources are fundamental and cannot be eliminated, others can be removed with the help of witness sensors [40]. We can classify the noise into two categories: *witnessed* and *non-witnessed* noise. The environmental and instrumental processes that contribute to the witnessed noise $n_w(t)$ are monitored by a set of channels denoted as $w_i(t)$, as discussed in [25]. Mathematically, the noise contributed by these channels can be expressed as an output of some activation function $\mathcal{F}$, *i.e.*, $n_w(t) = \mathcal{F}(w_i(t))$.

In general, the activation function $\mathcal{F}$ involves nonlinear and non-stationary couplings, particularly in gravitational-wave interferometers. *DeepClean* is a convolutional neural network that encodes this activation function using trainable weights $\vec{\theta}$. Thus, we can express the neural network as

$$n_w(t) = \mathcal{F}(w_i(t); \vec{\theta}). \qquad (2.1)$$

The *DeepClean* architecture was designed to be a symmetric auto-encoder that has four downsampling layers (convolution) and four upsampling layers (transpose convolution). The input layer has a flexible dimensionality to match the sampling frequency and number of witness channels in the input data. the first downsampling layer is designed to have 8 channels (features) with the same sampling frequency as in the input data. Each sucessive layer downsample the data by a factor 2 and increases the number of features by a factor 2, meaning that the latent vector has 64 features. The four up-sampling layers halves the number of features and doubles the sampling frequency at each layer, thereby regaining the same dimensionality as the input data. An output convolutional layer is then applied to map the data into a one-dimensional time series of noise prediction. At each layer, convolution or transpose convolution is followed by batch normalization and a *tanh* activation function to improve the model's generalization ability. A schematic diagram
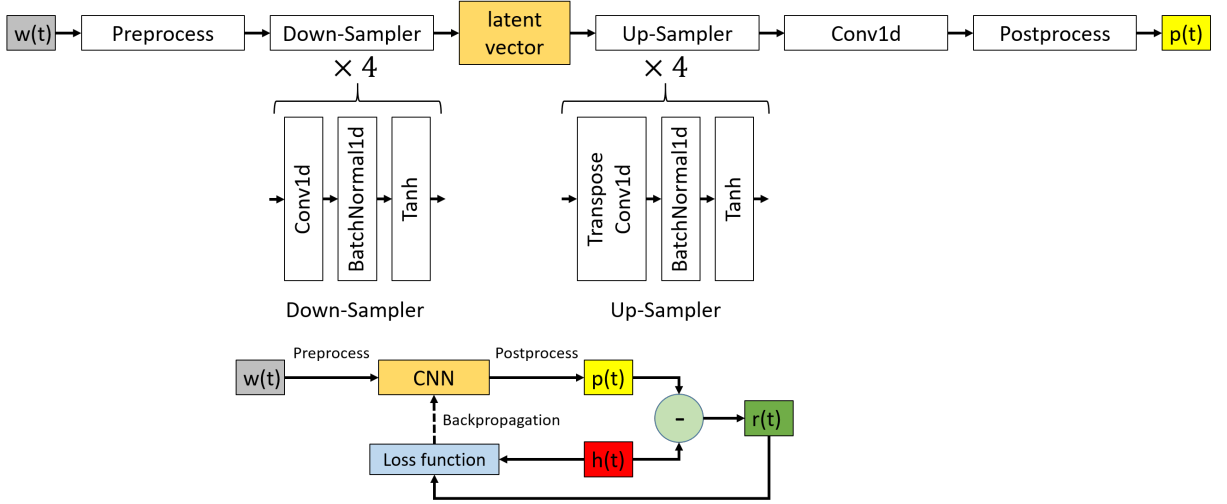
FIG. 1: The top diagram illustrates the *DeepClean* architecture and the workflow. *DeepClean* takes timeseries data from multiple witness channels as input and runs it through a fully convolutional autoencoder. The autoencoder has four convolution layers for downsampling and four transpose-convolution layers for upsampling. After each layer, batch normalization and a tanh activation function are applied. Finally, an output convolutional layer generates the one-dimensional noise prediction. The flowchart at the bottom depicts a typical training workflow for *DeepClean*. The ADAM optimizer is employed to minimize the loss function by navigating through the gradient space.

of the *DeepClean* architecture, along with a flowchart of a typical *DeepClean* workflow, is presented in Fig. 1.

The weights $\vec{\theta}$ are trained using the gradient descent algorithm [41] by minimizing an appropriate loss function. In the case of *DeepClean*, the loss function is defined as the ratio of the noise spectrum of the cleaned strain to the original strain, summed over all frequency bins within the analysis bandwidth $[f_{min}, f_{max}]$:

$$J = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{S_r^{(i)}}{S_h^{(i)}}} \qquad (2.2)$$

Here, $S_r^{(i)}$ is the power spectral density (PSD) of the residual strain at i^{th} frequency bin after subtracting $n_w(t)$. Likewise, $S_h^{(i)}$ is the PSD of the original strain at i^{th} bin before subtracting $n_w(t)$.

Prior to processing with *DeepClean*, both the strain and witness time-series are pre-processed by normalizing the time-series to ensure they have zero mean and unit variance. The strain data is further bandpass filtered to the frequency range of interest $[f_{min}, f_{max}]$. The pre-processed data is then input into the trained *DeepClean* to predict the noise contamination. To prevent boundary artifacts, predictions are made on 8 s segments with 4 s overlaps. These overlapping noise predictions are then combined after applying *Hann* windows to improve the prediction quality.

Subsequently, the predicted noise is band-pass filtered to $[f_{min}, f_{max}]$ to exclude any frequencies outside this range. After reversing the normalization steps, the predicted noise is subtracted from the original strain, yielding the cleaned strain.

In the following sections, we will use a mock data challenge to evaluate performance of *DeepClean* and to conduct validation tests.

III. A MOCK DATA CHALLENGE

To evaluate the effectiveness of *DeepClean*, we performed an end-to-end analysis of mock data through a Mock Data challenge (MDC) introduced by the LVK to benchmark and prepare the low latency analysis pipelines. The mock data is generated by injecting compact binary signals into the O3 strain data from LIGO Hanford and LIGO Livingston. We selected the low-latency O3 data (labeled as `GDS-CALIB_STRAIN`) from the 20-day period between September 1, 00:00:00 UTC and September 20, 00:00:00 UTC.This period exhibits high coherence between the strain and intended witness channels in both H1 and L1, making it well-suited for testing the performance of *DeepClean*.

The injected compact binary signals comprise binary black hole mergers, binary neutron star mergers, and neutron star-black hole mergers. The parameters of the injections such as masses, spins, luminosity distance, and other extrinsic parameters, are drawn from simulated distributions that are consistent with the O3-inferred population models [42] The coalescence times of the 25,000 injections are uniformly distributed over the 20–day period, such that there are no overlapping signals. Additionally, all the signals are generated by using 10 Hz as the lower cut-off frequency.

IV. APPLYING *DEEPCLEAN* ON THE MOCK DATA

The noise regression analysis is performed in two steps; training and cleaning (also referred to as the *inference*). Below we describe the operational parameters of training and cleaning considered in this study.

A. Strategy for Training and Cleaning the MDC data

To train and clean the 20 days of MDC data, we adopted a strategy that involves selecting only science-quality data labeled as `DMT-ANALYSIS_READY:1`. In total, we identified 47 science sub-segments (*a.k.a* active segments)[2] in H1 and 72 in L1 over the 20-day period. We used *DeepClean* to clean each science segment, with training performed once using the first 2000 s of the sub-segment, regardless of the length. This approach is supported by a detailed study outlined in section VI C. A visualization of this strategy can be seen in Fig. 2.

B. Target noise: power-line at 60 Hz and the side-bands

To illustrate non-linear and non-stationary couplings, we consider the 60 Hz line of the power mains, which is modulated by low-frequency noise from LIGO's alignment sensing and control (ASC) system [15]. This coupling produces sidebands around the central frequency, and we use a set of witness channels that were previously used to subtract these sidebands during the third observing run (O3) [24].

C. Data Pre-processing

The original strain data, recorded at a rate of 16384 Hz, is downsampled to 4096 Hz for this analysis. This is because the instrumental and environmental noise we want to subtract mostly occurs at frequencies below 100 Hz, and 4096 Hz is sufficient for downstream analyses like detection and parameter estimation of compact binary mergers. Although *DeepClean* can handle any sampling rate, a reduced rate makes training easier due to the reduced data size.

Most witness channels that are coupled to 60 Hz power-line noise have lower sampling rates than the strain data, often below 100 Hz or even below 10 Hz (known as fast

and slow channels). However, *DeepClean* requires all input channels to have the same sampling rate. Therefore, we upsample these channels to match the strain data rate. As discussed in Sec. II, each channel's data is normalized independently to have zero mean and unit standard deviation. The strain data is bandpass filtered to limit the relevant frequency range for the target noise. Specifically, *DeepClean* uses an 8th-order Butterworth filter to bandpass filter the data to the 55-65 Hz range. The frequency range considered here is wide enough to contain all the sidebands around 60 Hz.

D. Training

During the training process, the pre-processed data is divided into overlapping segments *a.k.a* kernels[3]. These kernels are then grouped into batches, with each batch consisting of a fixed number (`batch_size`) of kernels. For this analysis, we used a `batch_size` of 32 with kernels that are 8 s long and with 7 s overlap between two kernels. This results in 25 s of data in each batch. The entire training data of 2000 s is then composed of 1993 overlapping kernels, which translates to a total of 80 batches. When the entire training data is passed through *Deep-Clean* once (known as one epoch), the algorithm takes 80 iterations, with one batch taken at each iteration. At every iteration, the loss function is calculated, and back-propagated to compute the gradients, and subsequently updates the weights. In the process of weight optimization, *DeepClean* uses *ADAM* optimizer[43] to navigate through the gradient space and minimize the loss function.

Our analysis indicates that the loss function converges in approximately 20-25 epochs during typical 60 Hz noise subtraction. This translates to roughly 1600-2000 iterations using the settings described in this example.

E. Cleaning and post-processing

Unlike in training, we do not necessarily have to perform cleaning on overlapping kernels because, given a set of trained weights, the predictions for a certain data segment are always the same and do not benefit from averaging over overlapping kernels. However, when performing bulk offline analysis of thousands of seconds of data, it is inevitable to clean the long data segment after splitting into batches of shorter segments, due to memory constraints.

In addition, CNN architecture underweights the edges of the segments by design, which can lead to artifacts at the kernel edges. These artifacts get enhanced during

---

[2] For the time between two science segments, the detector is either not collecting data or the collected data does not pass data quality tests

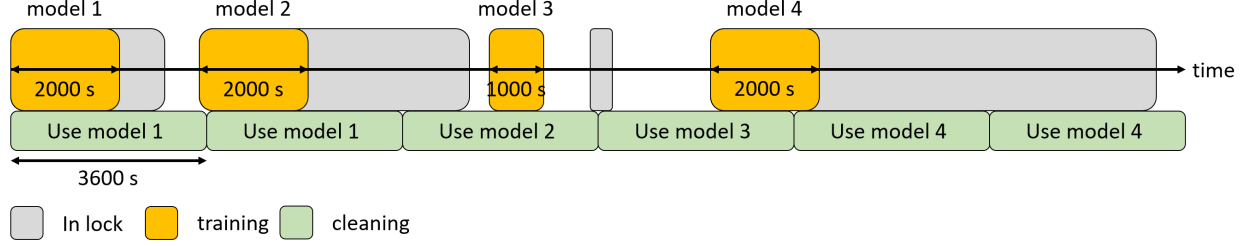[3] Not to be confused with the filter kernel used by the convolution operator in the CNN architecture

FIG. 2: This schematic shows the training strategy used for analyzing the mock data. The grey shaded segments represent science-quality data, and the yellow indicates that a model training is performed at the beginning of each science segment. The green segments represents one-hour long inference periods where the trained model is used. That means, once a model is trained at the beginning of a science segment, all the subsequent data until the start of next science segment is cleaned using that model.
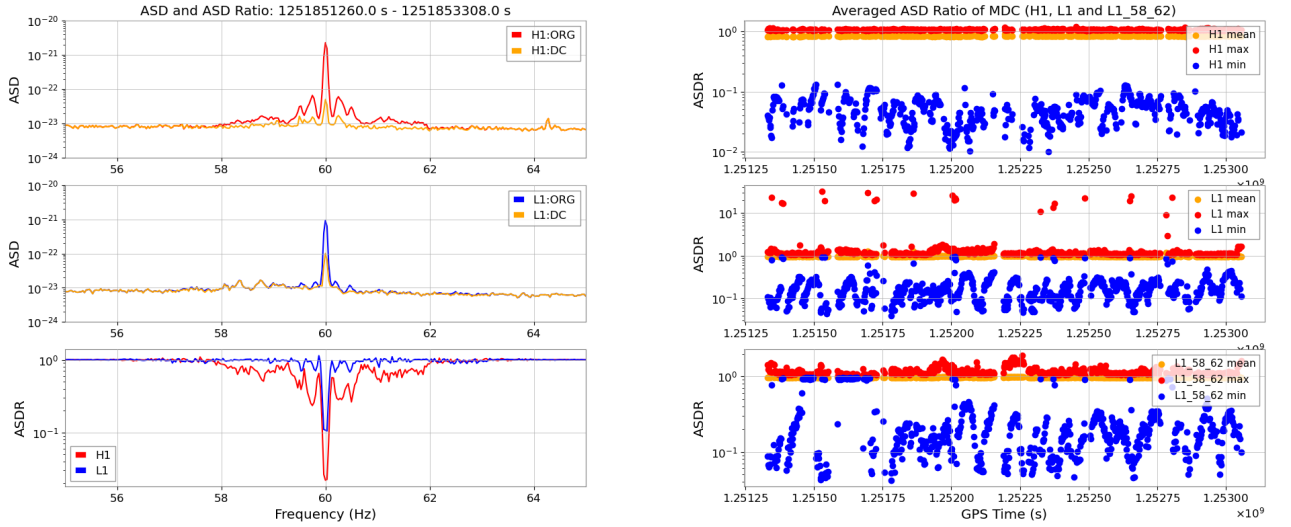

FIG. 3: [Left] Amplitude spectral densities (ASD) of the original and cleaned data are shown in the top (H1) and middle (L1) panels. The bottom panel shows the cleaned-to-original ASD ratio for both H1 (red) and L1 (blue). The plots are made using randomly picked 2048 seconds of MDC data from both H1 and L1. [Right] ASD ratios computed over the 20-day period of MDC data for H1 (upper) and L1 (middle and lower). Each point on the x-axis represents 256 s of data starting from there on and the y-axis shows the minimum (blue), maximum (red) and the mean (orange) of the ASD ratio from the [55, 65] Hz band. For L1, due to quality issues (see the descussion in the text), the analysis is repeated with a narrower frequency band;z [58, 62] Hz which is shown in the bottom panel while the middle panel shows the results obtained with [55, 65] Hz.

the bandpassing step in the post-processing, and they can also spread to samples farther from the edges. To prevent these edge effects, we apply Hann windows to the predictions from overlapping kernels and employ an averaging procedure before they are band-passed.

For the offline analysis, we perform inference on 3600 s-long (1 hour) chunks of data, using a model that is trained at the beginning of the science segment, as previously described.

F. Performance: Improvements in the noise spectral density

To assess the quality of noise regression, we compared the ASD of the cleaned strain to the original strain in the frequency band of 55-65 Hz using ASD ratio as a metric. The ASD ratio was computed on 2048 s of data from both H1 and L1 data, and the resulting plot (Fig. 3, left panel) showed a well-subtracted peak at 60 Hz and its sidebands. The right panel in Fig. 3 shows the ASD ratio computed over the 20 days of MDC data. Each point on the x-axis (in units of seconds) represents the 256 s data starting from that second onwards. For each x value, there are three y values, which are the minimum,

maximum, and the mean of the ASD ratio of that particular 256 s. For example, at each point, there will be an ASD ratio curve similar to the bottom panel on the left. The *mininum* will represent the subtraction achieved at the 60 Hz peak. The *maximum* is plotted with the intention of capturing ASD ratio that goes above 1, *i.e.* any noise that is contributed by *DeepClean*. The *mean* is meant to showcase the overall subtraction in the band including the sidebands.

In the top right panel, we have the ASD ratios from H1 noise subtraction. the maximum stays around 1, the *mean* and *minimum* below 1 consistently over the 20 days. This indicates a quality subtraction. On the other hand, for L1 subtraction in the middle right panel, we notice that the *maximum* of ASD ratios are well above one for many segments. These peaks in the ASD ratios are understood to be happening at 56 Hz and 64 Hz while the exact reason is not well understood. It can be a data quality issue either in the strain or the witness channels, leading to poorly converged models of the neural networks. We repeated the analysis by narrowing down the frequency band to $58 - 62$ Hz such that the frequencies of noisy peaks (56 Hz and 64 Hz) are excluded from the band. The results are shown in the bottom right panel. It shows that the unwanted features are filtered out by appropriately narrowing down the frequency band of the analysis.

V. VALIDATION TESTS WITH ASTROPHYSICAL METRICS

In the preceding section, we explored the use of *DeepClean* on mock data and demonstrated improvements in the ASD ratios. This section concentrates on astrophysically-motivated validation tests to ensure the effectiveness and safety of applying *DeepClean* to data containing astrophysical signals. We examine two specific areas: the impact on the sensitivity of compact binary searches, demonstrating effectiveness, and the assurance of signal integrity in source parameter estimation.

A. Compact binary search sensitivity

The GstLAL-based inspiral pipeline (referred to as GstLAL) is a matched-filtering based pipeline used to detect compact binary mergers [44–48]. GstLAL has played an instrumental role in low-latency detections of gravitational-waves [28, 49], and directly enabled the observation of electromagnetic counterparts associated with a BNS merger [50, 51].

We perform two GstLAL analyses on ~ 20 days of O3 data to assess the performance of *DeepClean*. The first analysis acts as a control and uses the final strain frames cleaned and published by Advanced LIGO and Advanced Virgo [52]. The second analysis ingests the frames processed by *DeepClean*. In each analysis, we
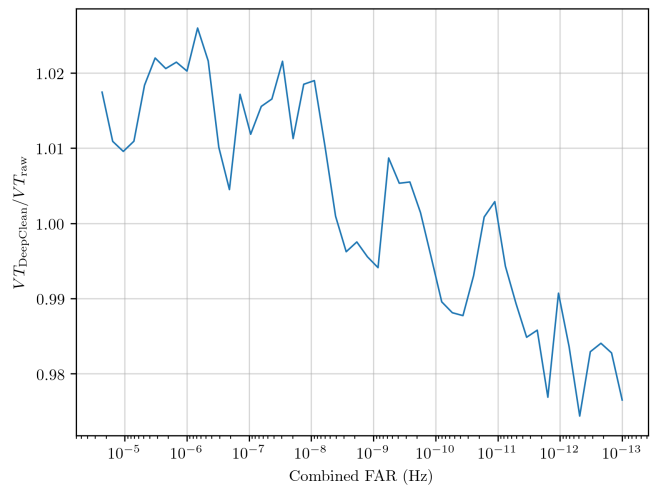

FIG. 4: The fractional improvement in sensitive volume (VT) measured by GstLAL (after *DeepClean* to before *DeepClean*) shown as a function of the estimated false-alarm-rate. *DeepClean* improves the sensitive volume of the search for false-alarm-rates of approximately 2 per day to 1 per 100 years, but there is a slight loss in sensitivity to very high significance events.

search for a set of astrophysically distributed simulated gravitational-wave signals, or *injections*. The injections span $m_i \in [5M\odot, 50M\odot]$ in component mass and $s_{iz} \in [0, 0.99]$ in component spin aligned with the orbital angular momentum. We evenly space the resulting 54000 injections 32 seconds apart, and separately place them into each data stream.

We find that we recover more injections in data cleaned by *DeepClean* for false-alarm-rates between 2 per day and 1 per 100 years, as shown in Fig. 4. For highly significant simulations (false-alarm-rates less than 1 per 100 years), here is a slight loss in sensitivity. We do not expect that this loss significantly impacts the chance of detection for these loud events. We hypothesize that this behavior is a result of *DeepClean* focusing on removing quiet noise artifacts while leaving loud noise transients from other sources in the data, causing the slope of the extrapolated background to lessen. In the limit of more data, we expect the VT ratio at high significance to asymptote to 1; we leave confirmation of this behavior to future work.

B. Parameter estimation of coalescing binaries

After applying a denoising pipeline, it is critical to perform parameter estimation (PE) of the underlying astrophysical signals as a validation test. This serves two purposes: firstly, to ensure that the regression analysis has not affected the original signals and, secondly, to assess any improvement in the credible intervals of the estimated parameters resulting from noise-subtraction. In this study, we focus only on the first purpose since the noise subtraction of 60 Hz alone may not yield any no-

table improvement in the credible interval.

To perform this test, we selected injections from our MDC dataset based on their coalescence frequencies. The coalescence frequency, also known as the frequency of the last stable circular orbit of a binary evolution, is given by $f_{lso} = (6^{3/2}\,\pi\,m)^{-1}$ Hz, where $m = m_1 + m_2$ is the total mass of the binary in the observer frame. We pick only those injections whose f_{lso} lies between 55 Hz and 70 Hz. We choose this frequency range because our target noise is around 60 Hz and signals with a peak frequency around 60 Hz would demonstrate the most significant scientific benefits. We found 258 BBH injections that satisfy this source criterion.

The literature contains well-described methods for estimating parameters from gravitational-wave signals, and there are standard analysis pipelines available that use stochastic samplers [53, 54]. For our analysis, we utilized tools from the *Bilby* [54] Bayesian library. We ran the *Dynesty* sampler [55] to sample from a 15-dimensional parameter space that included the luminosity distance, two mass parameters, six spin parameters, the time and phase of the binary coalescence, and four angles defining the binary's sky-location and orientation relative to the line of sight.

Out of the 258 injections, only 84 events met the minimum signal-to-noise ratio criterion of 4 at both detectors, confirming their detection and indicating the potential for reliable parameter estimation. Additionally, we did encounter sporadic instances where the cleaned data was noisier than the original data. We subsequently excluded these affected segments from our analysis and we were then left with 78 injections for our PE study. In order to address this issue for practical online setups, we need to incorporate validation tests to ensure that the outputs of the *DeepClean* algorithm are not noisier than the original. If the *DeepClean* output is found to be noisier, one needs to replace them with the original data as a baseline solution. More involved approaches to resolve this issue would include increasing the cadence of training.

We conducted parameter estimation (PE) on both the denoised and original strain data, and compared the results. In Fig. 5, we present 3D posteriors of the luminosity distance and two mass parameters obtained from one of the 258 injections we analyzed. The posteriors from the cleaned data (orange) are consistent with those of the original data (blue). This indicates that the noise regression analysis did not introduce any unwanted noise components or remove any spectral features from the signal itself. The same is true even if *DeepClean* is trained on data that has injections, as shown in the green curve.

Fig. 6 displays p-p plots for the fifteen parameters from the 78 events, showing excellent agreement between the p-p plots before *DeepClean* (left) and after *DeepClean* (right). This observation is essential as it validates the safety of the underlying astrophysical signals when the *DeepClean* algorithm is applied. This result demonstrates that the algorithm does not harm the underlying astrophysical signals and hence supports the reliability of
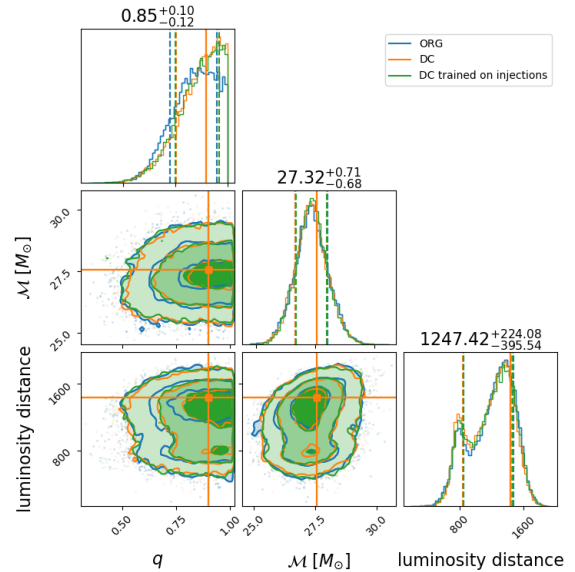

FIG. 5: Corner plot showing the posteriors of the mass parameters and the luminosity distance before and after the subtraction of the 60 Hz power-line and their side-bands using *DeepClean*

the analysis.

VI. FEASIBILITY STUDY FOR LOW-LATENCY DEPLOYMENT OF *DEEPCLEAN*

To perform noise subtraction in real-time, we must employ a different approach from the offline analysis outlined in the previous section. The offline analysis involved dividing the data into chunks of 3600 s, predicting noise on overlapping kernels of 8 s, and then combining them after applying window functions (see section IV for more details). This approach necessitates having a substantial amount of data available at once, enabling the creation of overlapping 8 s kernels. In contrast, the online analysis aims to clean the data as soon as it becomes available and make it accessible to low-latency search pipelines downstream of *DeepClean*. Therefore, we need a different workflow and strategy for the online version of *DeepClean*. A fully functional model of online *DeepClean* complemented with the *Inference-as-a-service* model will be presented in a future publication. Here, we discuss the key differences that separate it from the offline model, the issues it raises, and some preliminary figures of merit.

A. Edge effects

The need to develop a new strategy arises from *edge effects*, which occur when the noise prediction quality deteriorates towards the edges of a kernel. Fig. 7 shows that the first and last approximately 0.5 s of a 4 s kernel are
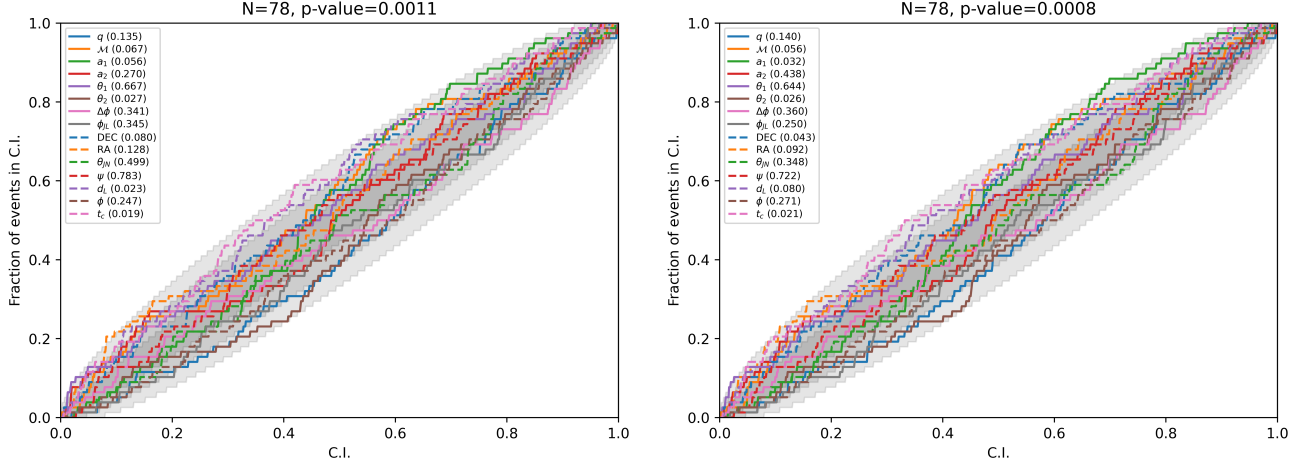
FIG. 6: P-P plots generated from the parameter estimation studies of 78 binary black hole injections, comparing the results before and after the application of the *DeepClean* algorithm for offline noise subtraction. The x-axis represents the credible interval, while the y-axis shows the fraction of injections recovered within that interval. These P-P plots are used to validate whether the injected parameters, after noise regression, can be recovered within the statistical uncertainty limits. As seen in the figure, the parameter recovery after *DeepClean* is at least as good as, if not better than, that achieved prior to applying the algorithm. The P-values included in the plots are derived from the Kolmogorov-Smirnov test, which quantifies the degree to which the credible interval distributions differ from the expected distributions.

susceptible to noisy spectral features. The width of each 4 s segment along the vertical axis shows the difference between the online and offline predictions. The offline prediction is made on a 128 second longer segment in which the 4 seconds shown are far from the edges. To mitigate these effects, *DeepClean* uses overlapping kernels and Hann windows to give more weight to the center of each kernel. This approach has been found to work well for offline cleaning. For online cleaning, our aim is always to clean the 1 s segment that is recorded most recently. Even if we divide the 1 s data into shorter overlapping kernels, there are no data available to overlap with the very last kernel and hence the edges can not be fully suppressed.

The reduction in quality at the edges of the kernel can be attributed to the natural tendency of CNN architectures to discard information at the edges of input data. This is particularly relevant in *DeepClean*, where we employ filter kernels of size 7 and strides of 2, resulting in the features at the edges being captured at a lower level compared to those at the middle of the kernel during convolution. For example, a sample from the middle is captured four times by the sliding filter, while a sample at the edge is only captured once. While *DeepClean* attempts to alleviate this issue by using zero-padding of size 3 at the edges, the edges are still captured only three times with this padding size. Increasing the padding size could be a potential solution, but it could lead to array size problems since the padding size is also constrained by the input-output sample size matching.

B. A working model for $\sim 1s$ latency

While ongoing work aims to comprehensively address edge effects, a simple modification to the workflow can mitigate the issue in the meantime. This resolution comes at the cost of a latency of approximately 1 second. To ensure that the quality of our online analysis matches that of our offline analysis, we employ a 4-second kernel that includes 2 seconds of data before and 1 second after the 1-second target segment. The additional data ensures that the target segment is located in the center of the kernel. The DeepClean is then applied on the 4-second kernel, and the 1-second target segment is extracted for analysis. A cartoon depicting this is shown in Fig. 7.

As the affected edges are not exactly 1 s in length, we can select a part of the output that is closer to the edge where we aggregated the future data. This edge, where future data is aggregated, is referred to as the aggregation latency. Fig. 8 displays a scatter plot with the achieved ASD ratio on the x-axis and the overall latency[4] on the y-axis, for different aggregation latencies. It is evident that the subtraction quality gets better by allowing higher aggregation latencies as shown by the reduced ASD ratios in the graph.

---

[4] The *overall latency* is the term referred to as the latency in producing the cleaned frames *w.r.t* the time when the original frame became available. It is comprised of the time taken to load the data, perform subtraction, and write the output frame to disc, in addition to the aggregation latency.
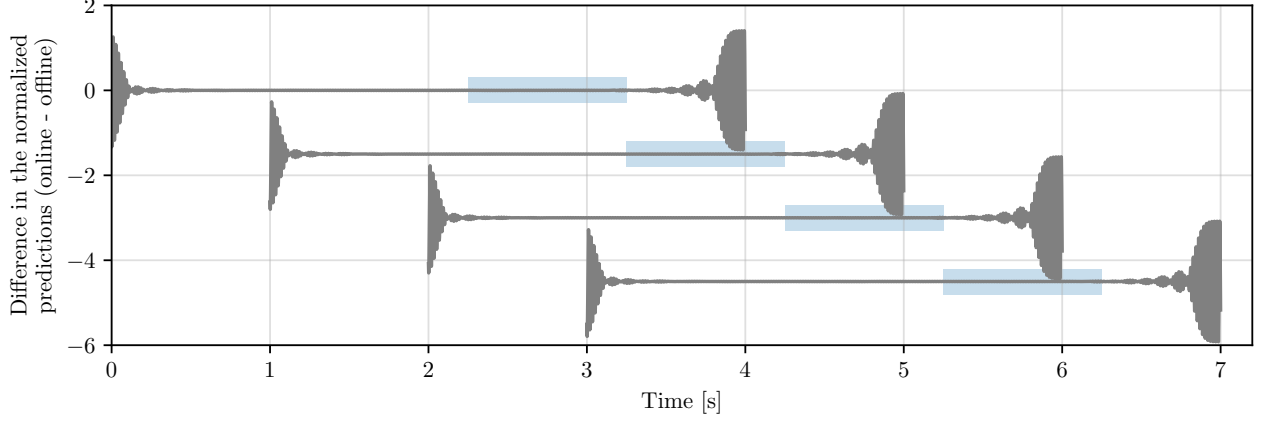
FIG. 7: The diagram shows how *DeepClean* will be used for low-latency (online) denoising. The figure compares the online and offline predictions, both normalized, for 4-second duration segments. The offline prediction for the same segment is extracted from the middle of a 128-second-long prediction. The difference bars indicate that the predictions differ only at the edges, which are less than a second long. To avoid edge effects, *DeepClean* is applied to 8 seconds of data, consisting of 6 seconds from the past and 1 second from the future, in addition to the 1 second of target data. This is done for every 1 second frame, and the wait for a future frame causes an additional 1 second of latency. After the prediction, everything except the 1 second target data is discarded from the 8-second segment and written to disk as cleaned frames, ready for downstream analyses.
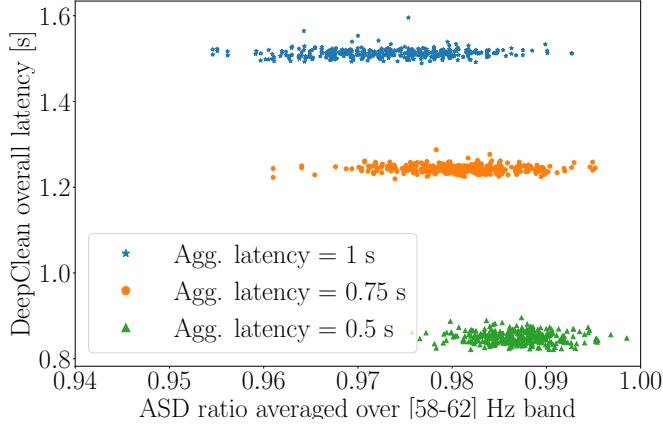

FIG. 8: Scatter plot showing the latency vs quality trade-off. The overall latency referred to here is the time taken by DeepClean to produce the output strain after the raw strain is made available. The ASD (amplitude spectral density) ratio on the x-axis is computed from every 32 s of the data. The three different colors shows the different aggregation latencies. Notably, the quality of the ASD ratio improves with higher aggregation latency, at the cost of increased overall latency.

C. A case study for training DeepClean at low latency

To enable real-time gravitational wave data cleaning, it is necessary to train and validate the machine learning model in short timescales. Unlike offline analysis where there is sufficient time to optimize and fine-tune the
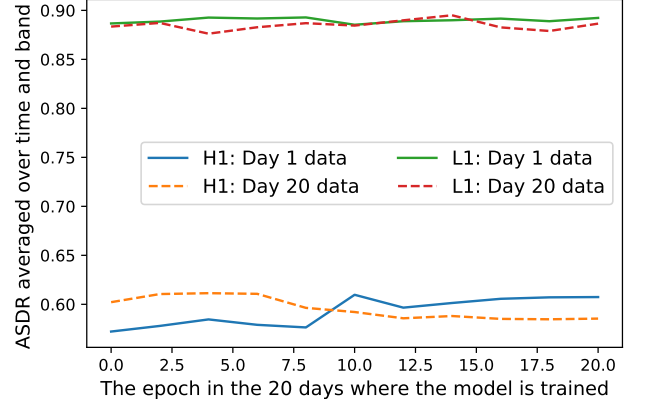

FIG. 9: The figure underscores the significance of periodically retraining the neural network, *deepClean*. It presents two traces for each detector: one representing the cleaned strain data from day 1 (solid line) and the other displaying the cleaned strain from day 20 (dashed line), derived from our 20-day MDC dataset. The horizontal axis depicts the GPS time of the training data, with an initial time conveniently set to zero. Meanwhile, the vertical axis represents the Amplitude Spectral Density Ratio (ASDR). Notably, there is an observed increase in ASDR when the training data is selected from a segment further away from the cleaning data, particularly noticeable in H1. In contrast, the disparity in ASDR between training and cleaning data is less prominent in L1 over this 20-day period. Overall, the figure emphasizes the importance of regular retraining to ensure optimal performance and accurate data cleaning.

trained model, online cleaning requires that the model be trained quickly and validated frequently. This is because the noise features in the data are generally non-stationary, and the noise coupling that *DeepClean* once learned could change after a certain time, making it necessary to have new models periodically trained on the most recent data.

To explore this in detail, we conducted a case study using the 20 days of MDC data. We trained the model once for each science segment, resulting in a total of 47 trained models for H1 and 72 for L1 over the 20 days. We then took two examples of inference data, one from day 1 and the other from day 20, and cleaned them with all the models available. Fig. 9 shows the ASD ratio results on the y-axis and the time (from the 20 days) where the model is trained on the x-axis. The solid line represents the data from day 1, and the dashed line represents the data from day 20, for both the detectors.

It is observed that the ASD ratio changes as we move away from the time of the training dataset, particularly in H1 data. For instance, the day 1 data has an ASD ratio below 0.6 when trained with data from day 1, 2, or 3, but the ASD ratio goes above 0.6 when the training dataset is from day 20. The same observation is true for cleaning the data from day 20, which is best cleaned with the model trained on day 20.

Although this study indicates that trained models become sub-optimal over time, we noticed that it does not happen over the timescale of minutes or hours, but rather changes over the timescale of days. In L1, we do not notice any significant change in the ASD ratio over time, which may be an indication that the coupling features in L1 are rather stationary. Overall, our observations emphasize that there is enough time for training and validating a model that is trained at low latency.

It should be noted that these observations are based on our cleaning of 60 Hz powerline and sidebands. The timescale over which the coupling features change will also depend on the coupling itself. For example, a different coupling in a different frequency range, can be highly non-stationary and would require model retraining at a higher cadence. Fortunately, our production deployment of *DeepClean* is highly compute-efficient due to leveraging the GPU resources and hence capable of training once every 30 minutes or less. The details of that is differed for a later publication.

VII. SUMMARY AND OUTLOOK

We conducted noise regression using the *DeepClean* algorithm on bulk offline O3 data with high latency. For offline analyses, we focused on optimizing the network configuration, without being constrained by latency, which is crucial for online analyses. The cleaned data obtained from offline analyses are validated using downstream applications, such as detection and parameter estimation of detected compact binaries.

In a separate analysis, we demonstrated the applicability of *DeepClean* for low-latency noise subtraction, where unclean data is fed as 1-second-long frames. However, we observed some discrepancies at the edges of the cleaned segments due to the inherent nature of CNN-like architectures, which require further efforts to mitigate. To overcome this issue, we proposed a workaround by waiting for a future 1-second data, such that there are no edges in the current segment. By adopting this strategy, we were able to replicate the results obtained from the high-latency analysis, showing the effectiveness of the low-latency *DeepClean* application.

To evaluate the efficacy of *DeepClean* over time, we investigated how frequently the trained models need to be updated. Our analysis of 20 days of MDC data revealed that retraining the model every 1 or 2 days is sufficient for subtracting the 60 Hz noise. However, we acknowledge that this interval may vary depending on the nature of the coupling. The *DeepClean* deployment described in Ref. [56] allows frequent model training, as often as every 30 min, which we anticipate would be sufficient for most of the couplings we encounter in the future.

Our ongoing work includes extending *DeepClean* to different frequency ranges especially aiming the broadband noise in LIGO detectors below 30 Hz. Efforts are also going on to apply *DeepClean* on Virgo and KAGRA data. Further, as mentioned before, an end-to-end model of *online-DeepClean* is being built, deployed, and tested with different validation methods. This comes as part of preparing for production application of *DeepClean* in O4. The details pertaining to all the ongoing efforts will be detailed in future publications.

This paper has been assigned the internal LIGO preprint number P2300153.

---

[1] J. Aasi et al. (LIGO Scientific), Class. Quant. Grav. **32**, 074001 (2015), 1411.4547.

[2] G. M. Harry et al., Classical and Quantum Gravity **27**, 084006 (2010), URL https://doi.org/10.1088/0264-9381/27/8/084006.

[3] F. Acernese et al. (VIRGO), Class. Quant. Grav. **32**, 024001 (2015), 1408.3978.

[4] R. Abbott et al. (LIGO Scientific, VIRGO, KAGRA) (2021), 2111.03606.

[5] A. H. Nitz, S. Kumar, Y.-F. Wang, S. Kastha, S. Wu, M. Schäfer, R. Dhurkunde, and C. D. Capano, Astrophys. J. **946**, 59 (2023), 2112.06878.

[6] S. Olsen, T. Venumadhav, J. Mushkin, J. Roulet, B. Zackay, and M. Zaldarriaga, Phys. Rev. D **106**, 043009 (2022), 2201.02252.

[7] Y. Aso, Y. Michimura, K. Somiya, M. Ando, O. Miyakawa, T. Sekiguchi, D. Tatsumi, and H. Yamamoto (The KAGRA Collaboration), Phys. Rev. D **88**, 043007 (2013), URL https://link.aps.org/doi/10.1103/PhysRevD.88.043007.

[8] K. Somiya, Classical and Quantum Gravity **29**, 124007 (2012).

[9] B. P. Abbott et al. (KAGRA, LIGO Scientific, Virgo, VIRGO), Living Rev. Rel. **21**, 3 (2018), 1304.0670.

[10] C. Cahillane and G. Mansell, Galaxies **10**, 36 (2022), 2202.00847.

[11] Davis et al., Classical and Quantum Gravity **38**, 135014 (2021), URL https://doi.org/10.1088/1361-6382/abfd85.

[12] D. Davis et al. (LIGO), Class. Quant. Grav. **38**, 135014 (2021), 2101.11673.

[13] S. Soni, C. Austin, A. Effler, R. M. S. Schofield, G. González, V. V. Frolov, J. C. Driggers, A. Pele, A. L. Urban, G. Valdes, et al., Classical and Quantum Gravity **38**, 025016 (2021), 2007.14876.

[14] K. Riles, Prog. Part. Nucl. Phys. **68**, 1 (2013), 1209.0667.

[15] V. Tiwari et al., Class. Quant. Grav. **32**, 165014 (2015), 1503.07476.

[16] P. Nguyen, R. M. S. Schofield, A. Effler, C. Austin, V. Adya, M. Ball, S. Banagiri, K. Banowetz, C. Billman, C. D. Blair, et al., Classical and Quantum Gravity **38**, 145001 (2021), 2101.09935.

[17] S. V. Vaseghi, *Wiener Filters* (John Wiley & Sons, Ltd, 2001), pp. 178–204, ISBN 9780470841624.

[18] A. H. Sayed, *Fundamentals of Adaptive Filtering* (John Wiley & Sons, 2003).

[19] D. Davis, T. J. Massinger, A. P. Lundgren, J. C. Driggers, A. L. Urban, and L. K. Nuttall, Class. Quant. Grav. **36**, 055011 (2019), 1809.05348.

[20] The LIGO Collaboration and the Virgo Collaboration, Classical and Quantum Gravity **37**, 055002 (2020), URL https://doi.org/10.1088%2F1361-6382%2Fab685e.

[21] R. Biswas, L. Blackburn, J. Cao, R. Essick, K. A. Hodge, E. Katsavounidis, K. Kim, Y.-M. Kim, E.-O. Le Bigot, C.-H. Lee, et al., Phys. Rev. D **88**, 062003 (2013), URL https://link.aps.org/doi/10.1103/PhysRevD.88.062003.

[22] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggelos, S. L. Larson, et al., Classical and Quantum Gravity **34**, 064003 (2017), URL http://stacks.iop.org/0264-9381/34/i=6/a=064003.

[23] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip, Phys. Rev. D **95**, 104059 (2017), URL https://link.aps.org/doi/10.1103/PhysRevD.95.104059.

[24] G. Vajente, Y. Huang, M. Isi, J. C. Driggers, J. S. Kissel, M. J. Szczepanczyk, and S. Vitale, Physical Review D **101** (2020), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.101.042003.

[25] R. Ormiston, T. Nguyen, M. Coughlin, R. X. Adhikari, and E. Katsavounidis, Phys. Rev. Res. **2**, 033066 (2020), 2005.06534.

[26] H. Yu and R. X. Adhikari, Front. Artif. Intell. **5**, 811563 (2022), 2111.03295.

[27] B. P. Abbott et al. (Virgo, LIGO Scientific), Phys. Rev. Lett. **116**, 061102 (2016), 1602.03837.

[28] B. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. Lett. **119**, 161101 (2017), URL https://link.aps.org/doi/10.1103/PhysRevLett.119.161101.

[29] B. D. Metzger, Living Rev. Rel. **20**, 3 (2017), 1610.09381.

[30] K. Cannon et al., Astrophys. J. **748**, 136 (2012), 1107.2665.

[31] B. P. Abbott et al. (LIGO Scientific, Virgo), Astrophys. J. **875**, 161 (2019), 1901.03310.

[32] S. Sachdev et al., Astrophys. J. Lett. **905**, L25 (2020), 2008.04288.

[33] Q. Chu et al., Phys. Rev. D **105**, 024023 (2022), 2011.06787.

[34] H. Yu, R. X. Adhikari, R. Magee, S. Sachdev, and Y. Chen, Phys. Rev. D **104**, 062004 (2021), 2104.09438.

[35] B. P. Abbott et al., Phys. Rev. D **93**, 112004 (2016), [Addendum: Phys.Rev.D 97, 059901 (2018)], 1604.00439.

[36] A. Buikema et al. (aLIGO), Phys. Rev. D **102**, 062003 (2020), 2008.01301.

[37] A. Gunny, D. Rankin, J. Krupa, M. Saleem, T. Nguyen, M. Coughlin, P. Harris, E. Katsavounidis, S. Timm, and B. Holzman (2021), 2108.12430.

[38] J. Krupa et al., Mach. Learn. Sci. Tech. **2**, 035005 (2021), 2007.10359.

[39] M. Wang, T. Yang, M. Acosta Flechas, P. Harris, B. Hawks, B. Holzman, K. Knoepfel, J. Krupa, K. Pedro, and N. Tran (2020), 2009.04509.

[40] A. Buikema, C. Cahillane, G. L. Mansell, C. D. Blair, R. Abbott, C. Adams, R. X. Adhikari, A. Ananyeva, S. Appert, K. Arai, et al., Phys. Rev. D **102**, 062003 (2020), 2008.01301.

[41] S. Ruder, arXiv e-prints arXiv:1609.04747 (2016), 1609.04747.

[42] R. Abbott et al. (KAGRA, VIRGO, LIGO Scientific), Phys. Rev. X **13**, 011048 (2023), 2111.03634.

[43] D. P. Kingma and J. Ba, arXiv e-prints arXiv:1412.6980 (2014), 1412.6980.

[44] S. Sachdev et al. (2019), 1901.08580.

[45] C. Hanna et al., Phys. Rev. D **101**, 022003 (2020),

1901.02227.

[46] C. Messick et al., Phys. Rev. D **95**, 042001 (2017), URL https://link.aps.org/doi/10.1103/PhysRevD.95.042001.

[47] L. Tsukada et al. (2023), 2305.06286.

[48] K. Cannon, S. Caudill, C. Chan, B. Cousins, J. D. E. Creighton, B. Ewing, H. Fong, P. Godwin, C. Hanna, S. Hooper, et al., SoftwareX **14**, 100680 (2021), 2010.05082.

[49] B. P. Abbott et al. (LIGO Scientific, Virgo), Phys. Rev. Lett. **116**, 241103 (2016), 1606.04855.

[50] B. P. Abbott et al. (LIGO Scientific, Virgo, Fermi GBM, INTEGRAL, IceCube, AstroSat Cadmium Zinc Telluride Imager Team, IPN, Insight-Hxmt, ANTARES, Swift, AGILE Team, 1M2H Team, Dark Energy Camera GW-EM, DES, DLT40, GRAWITA, Fermi-LAT, ATCA, ASKAP, Las Cumbres Observatory Group, OzGrav, DWF (Deeper Wider Faster Program), AST3, CAAS-TRO, VINROUGE, MASTER, J-GEM, GROWTH, JAGWAR, CaltechNRAO, TTU-NRAO, NuSTAR, Pan-STARRS, MAXI Team, TZAC Consortium, KU, Nordic Optical Telescope, ePESSTO, GROND, Texas Tech University, SALT Group, TOROS, BOOTES, MWA, CALET, IKI-GW Follow-up, H.E.S.S., LOFAR, LWA, HAWC, Pierre Auger, ALMA, Euro VLBI Team, Pi of Sky, Chandra Team at McGill University, DFN, AT-LAS Telescopes, High Time Resolution Universe Survey, RIMAS, RATIR, SKA South Africa/MeerKAT), Astrophys. J. Lett. **848**, L12 (2017), 1710.05833.

[51] V. C. LIGO Scientific Collaboration, GCN **G298048** (2017), URL https://gcn.gsfc.nasa.gov/other/G298048.gcn3.

[52] R. Abbott et al. (LIGO Scientific, VIRGO, KAGRA) (2023), 2302.03676.

[53] J. Veitch et al., Phys. Rev. D **91**, 042003 (2015), 1409.7215.

[54] G. Ashton et al., Astrophys. J. Suppl. **241**, 27 (2019), 1811.02042.

[55] E. Higson, W. Handley, M. Hobson, and A. Lasenby, Statistics and Computing **29**, 891 (2019), 1704.03459.

[56] A. Gunny, D. Rankin, P. Harris, E. Katsavounidis, E. Marx, M. Saleem, M. Coughlin, and W. Benoit, in *Proceedings of the 12th Workshop on AI and Scientific Computing at Scale Using Flexible Computing Infrastructures* (Association for Computing Machinery, New York, NY, USA, 2022), FlexScience '22, p. 9–17, ISBN 9781450393096, URL https://doi.org/10.1145/3526058.3535454.

[57] S. van der Walt, S. C. Colbert, and G. Varoquaux, Comput. Sci. Eng. **13**, 22 (2011), 1102.1523.

[58] P. Virtanen et al., Nature Meth. (2020), 1907.10121.

[59] J. D. Hunter, Computing in Science & Engineering **9**, 90 (2007).

[60] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, et al., in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides and B. Scmidt (IOS Press, Netherlands, 2016), pp. 87–90, URL https://eprints.soton.ac.uk/403913/.

[61] D. Foreman-Mackey, The Journal of Open Source Software **1**, 24 (2016), URL https://doi.org/10.21105/joss.00024.