OPEN ACCESS
IOP Publishing
Classical and Quantum Gravity
Class. Quantum Grav. **41** (2024) 195024 (20pp) https://doi.org/10.1088/1361-6382/ad708a

Demonstration of machine learning-assisted low-latency 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], Ritwik Sharma[8], Michael Coughlin[1], Philip Harris[2] and Erik Katsavounidis[2,3]

[1] School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, United States of America
[2] Department of Physics, MIT, Cambridge, MA 02139, United States of America
[3] LIGO Laboratory, MIT, 185 Albany St, Cambridge, MA 02139, United States of America
[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, United States of America
[8] Department of Physics, Deshbandhu College, University of Delhi, New Delhi, India

E-mail: mcholayi@umn.edu and chiajuichou@nycu.edu.tw

Received 11 March 2024; revised 5 August 2024
Accepted for publication 16 August 2024
Published 10 September 2024

Abstract

Low-latency 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

* Author to whom any correspondence should be addressed.

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 enhance 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 ~1–2 s. We also discuss various considerations that may be made while training *DeepClean* for low latency applications.

Keywords: gravitational waves, Mschine learning, noise regression

1. 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 (BNSs) 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 leading to substantial improvements in the sensitivities[9], and with the addition of a fourth detector, KAGRA (K1) [7, 8].

Upgrades in technology have improved the sensitivity of interferometers by reducing fundamental noise sources such as thermal and quantum fluctuations [9, 10]. 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 [10, 11], in particular, sources without well-known theoretical models (e.g. supernovae) [12]. Noise regression methods are used to remove these contaminants, typically by identifying their origin [13]. Gravitational-wave interferometers are equipped with auxiliary *witness* sensors or channels to independently monitor these processes in addition to the strain channel [14]. 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 [15–18].

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 [19–25]. 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.

[9] The exact sensitivity may vary over time during the course of observing run, which can be tracked at this url.

However, these deep learning noise regression algorithms have thus far been demonstrated primarily in high latency, or *offline*, 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 [26, 27]. Detecting electromagnetic counterparts that fade quickly after the gravitational-wave detection, such as γ-ray bursts and x-rays from BNS mergers, requires sending out low-latency alerts to trigger follow-up observations across electromagnetic frequencies [28–33]. Ground-based detectors are still below their designed sensitivities at lower frequency ranges (below 60 Hz) [34, 35], 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. Gunny *et al* [36] discussed in detail how to meet such computational demands in low latency, by employing the *as-a-service* computing paradigm [37, 38] 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* [23] infrastructure on GW 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[10] 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 2 provides a concise overview of the DeepClean architecture and the end-to-end infrastructure. Section 3 presents the details of our MDC. In section 4, we delve into the application of *DeepClean* on our mock data and present the corresponding performance metrics. Section 5 demonstrates the validation tests performed using astrophysically motivated metrics, including detection and parameter estimation (PE). Section 6 focuses on the feasibility of utilizing *DeepClean* for low-latency noise regression. Finally, section 7 concludes the study and discusses future prospects.

2. The *DeepClean* infrastructure

The *DeepClean* architecture has been described in detail in [23]. 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 $h(t) = s(t) + n(t)$.

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

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 [35]. 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 [23]. 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 non-linear 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

$$n_w(t) = \mathcal{F}\left(w_i(t); \vec{\theta}\right). \tag{1}$$

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

The weights $\vec{\theta}$ are trained using the gradient descent algorithm [39] 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 $[f_{\min}, f_{\max}]$:

$$J = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{S_r^{(i)}}{S_h^{(i)}}} \tag{2}$$

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

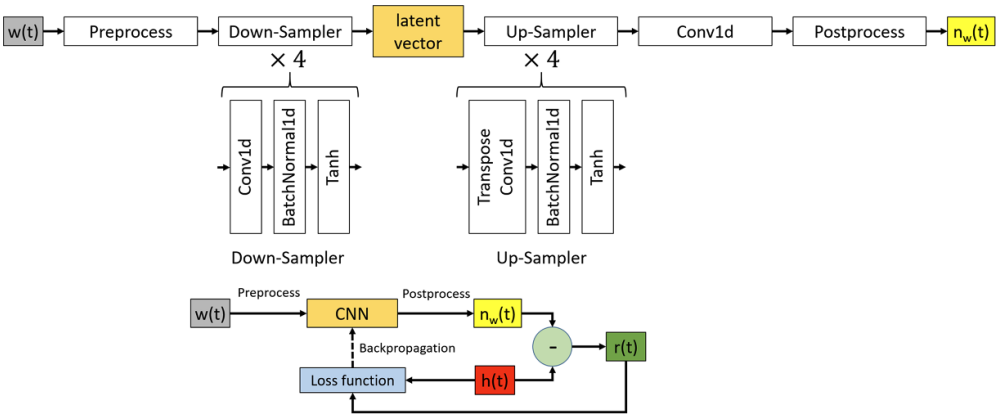

Figure 1. The top diagram illustrates the *DeepClean* architecture and the workflow. *DeepClean* takes timeseries data from multiple witness channels as input and runs it through a fully convolutional autoencoder. The autoencoder has four convolution layers for downsampling and four transpose-convolution 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.

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

3. A MDC

To evaluate the effectiveness of *DeepClean*, we performed an end-to-end analysis of mock data through a 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 01 September 2019, 00:00:00 UTC and 20 September 2019, 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*. Though this study is considering data from two LIGO detectors, the framework is applicable to Virgo and KAGRA. Virgo and KAGRA have different implementations of their auxiliary channels which vary in their names as well as functionality from those at LIGO. Therefore, the nature of the noise coupling, and the channels involved, will also be different.

The injected compact binary signals comprise binary black hole mergers, BNS 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 [40] The coalescence times of the 250 00 injections are uniformly distributed over the 20 day period in such a way that there are no overlapping signals. Additionally, all the signals are generated by using 10 Hz as the lower cut-off frequency.

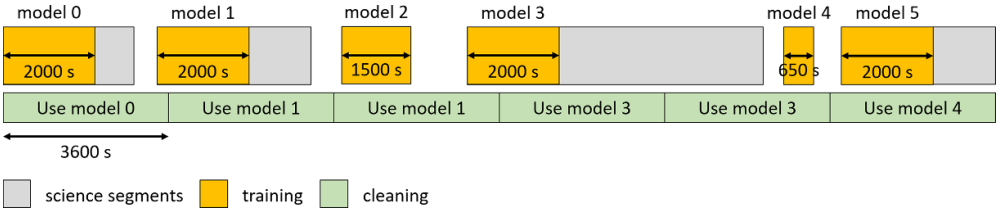

Figure 2. This schematic shows the training strategy used for analyzing the mock data. The grey shaded segments represent science-quality data, and the yellow indicates that a model training is performed at the beginning of each science-quality segment. The green segments represents one-hour long inference 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.

4. 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.

4.1. 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` [41]. In total, we identified 47 science sub-segments (a.k.a active segments)[11] 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 6.3. A visualization of this strategy can be seen in figure 2. There are exceptions to this strategy when a science segment is overall less than 2000 s. A realistic strategy going forward is to clean such segments using the most recent trained model, i.e. the very previous model and perform training on a segment only if at least 2000 s data is available.

Furthermore, it is not a good ML practice to use models trained on some data to clean the same data because our model may have been over-fitted to the training data and hence may bias the analysis results. For that reason, we exclude the injections on the very first 2000 s data on each segment as they are the training data.

4.2. 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 system [13]. 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) [22].

[11] For the time between two science segments, the data is either not collected or does not meet the quality standards.

4.3. Data pre-processing

The original strain data, recorded at a rate of 163 84 Hz, is down-sampled to 4096 Hz during pre-processing. This implies that the eventual cleaned strain will also be at this sampling rate. This choice aligns with the current needs of the downstream analyses in this study, such as the detection and PE of binary black hole mergers. Higher sampling rates may be preferred for cases like the PE of neutron star mergers. If such use cases arise, the framework is flexible enough to run at any sampling rate. The primary constraint is that the data must be sampled at a rate higher than the frequency of the noise coupling we aim to subtract. In this study, the targeted coupling is at 60 Hz, and most other instrumental and environmental noise that significantly impacts sensitivity occurs below 100 Hz. Therefore, a sampling rate of 4096 Hz is a reasonable choice, providing storage and training speed advantages over 163 84 Hz.

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 section 2, 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.

4.4. Training

During the training process, the pre-processed data is divided into overlapping segments a.k.a kernels[12]. These kernels are then grouped into batches, with each batch consisting of a fixed number (`batch_size`) of kernels. For this analysis, we used kernels of 8 s duration with 7.75 s overlap between two kernels, leading to a total of 7969 overlapping kernels. The training data is then passed to DeepClean in batches, with the size (`batch_size`) of 32 kernels resulting in a total of 249 batches. When the entire training data is passed through *DeepClean* once (known as one epoch), the algorithm takes 249 iterations, with one batch taken at each iteration. At every iteration, the loss function is calculated, and backpropagated to compute the gradients, and subsequently updates the weights. In the process of weight optimization, *DeepClean* uses *ADAM* optimizer [42] 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 4980–6225 iterations using the settings described in this example.

4.5. Cleaning and post-processing

As mentioned before, we only clean the science-quality segments and they are typically several hours to days of length. For offline cleaning of such long segments, it is convenient to split them into shorter chunks, to avoid memory issues. In this study, we use 1 h chunks. The model trained on the 2000s at the beginning of the same science segment is used to clean all the 1 h chunks, as previously described.

Every 1 h chunks of inference data are pre-processed to make them overlapping kernels of 8 s with 4 s overlap between them. The overlapping kernels are not necessary in inference

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

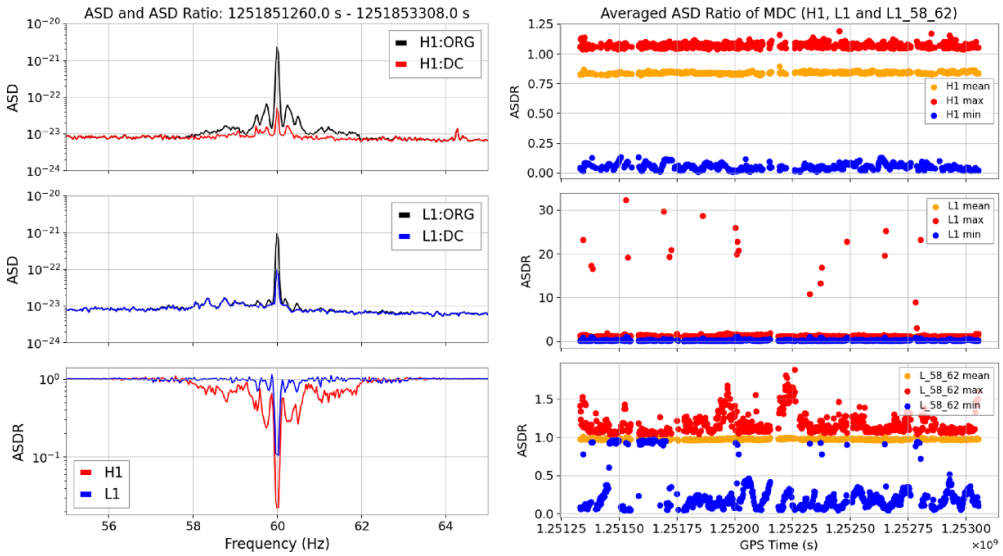

Figure 3. (Left) Amplitude spectral densities (ASD) of the original and cleaned data are shown in the top (H1) and middle (L1) panels where ORG represents the original strain and DC represents the Deepcleaned strain. The bottom panel shows the cleaned-to-original ASD ratio for both H1 (red) and L1 (blue). The plots are made using randomly picked 2048 s 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.

in ideal situations, 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, CNN architecture underweights the edges of the segments by design, which can lead to artifacts at the kernel edges. These artifacts get enhanced during the bandpassing step in the post-processing, and they can also spread to samples farther from the edges. To prevent this, we apply Hann windows to the predictions from overlapping kernels and employ a weighted-averaging procedure before they are being band-passed. Therefore, the overlapping kernels are utilized to make sure the kernel edges are under-weighted compared to the mid-region in terms of their contribution to the final prediction of the noise[13].

4.6. 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 (figure 3, left panel) showed a well-subtracted peak at 60 Hz and its sidebands. The right panel in figure 3

[13] To avoid confusion, it is important to note that overlapping kernels are used only internally within the pipeline. At the top level, the pipeline processes non-overlapping 1 h (or any desired length) chunks of input data from the witness channels and outputs noise predictions for non-overlapping 1 h chunks.

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.

5. 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 PE.

5.1. 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 [43–47]. GstLAL has played an instrumental role in low-latency detections of gravitational-waves [27, 48], and directly enabled the observation of electromagnetic counterparts associated with a BNS merger [49, 50].

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 [51]. The second analysis ingests the frames processed by *DeepClean*. In each analysis, we search for a set of astrophysically distributed simulated gravitational-wave signals, or *injections*. The injection masses are chosen 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. The injections span $m_i \in [5M_\odot, 50M_\odot]$, distributed uniformly in component masses, and with mass ratio and chirp mass restricted respectively to the ranges $[0.25M_\odot, 1M_\odot]$ and $[21M_\odot, 35M_\odot]$. This ensures that the f_{lso} lies between 55 Hz and 70 Hz, for all injections. This frequency range is preferred because our target noise is around 60 Hz and signals with a peak frequency around 60 Hz would demonstrate the most significant scientific benefits. The spins are uniformly distributed

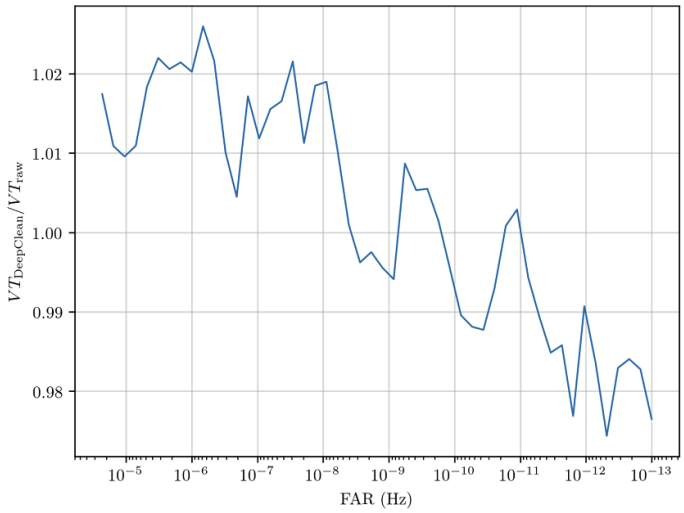

Figure 4. The fractional improvement in sensitive volume (*VT*) measured by GstLAL (after *DeepClean* to before *DeepClean*) shown as a function of the estimated false-alarm-rate (FAR) combined across two detectors. *DeepClean* improves the sensitive volume of the search for false-alarm-rates of approximately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

and $s_{iz} \in [0, 0.99]$ in component spin aligned with the orbital angular momentum. The luminosity distance is distributed as uniform in comoving volume up to redshift $\sim$0.5. We evenly space the resulting 54000 injections 32 s apart, using `SEOBNRv4_ROM` waveform model, and separately place them into each data stream.

Figure 4 shows the results where the ratio of population-averaged sensitive volume *VT* [52] is plotted against the false-alarm-rates (FARs) [53]. We find that we recover more injections in data cleaned by *DeepClean* for FAR between 2 per day and 1 per 100 years, as shown in figure 4. For highly significant simulations (FAR 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.

5.2. PE of coalescing binaries

After applying a denoising pipeline, it is critical to perform 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 notable improvement in the credible interval.

To perform this test, once again, we selected injections from our MDC dataset based on their coalescence frequencies—that the f_{lso} lies between 55 Hz and 70 Hz, as described in

section 5.1. We found 258 BBH injections that satisfy this source criterion in the science-quality segments.

The literature contains well-described methods for estimating parameters from gravitational-wave signals, and there are standard analysis pipelines available that use stochastic samplers [54, 55]. For our analysis, we utilized tools from the *Bilby* [55] Bayesian library. We ran the *Dynesty* sampler [56] with `IMRPhenomPv2` waveform model [57], 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 SNR criterion of 4 at both detectors, confirming their detection and indicating the potential for reliable PE. Similar detection criteria have been used in the literature [58, 59], rather than relying solely on the network SNR as the threshold. This approach is preferred because having a minimum SNR across multiple detectors results in higher detection significance and a lower FAR compared to an equivalent network SNR primarily coming from a single detector. Additionally, it provides better PE results due to improved localization of the source.

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. We also removed 13 injections that sit in the training segments since those segments are cleaned by the DeepClean models trained from those segments. Finally we were left with 65 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 PE on both the denoised and original strain data, and compared the results. In figure 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.

Figure 6 displays p–p plots for the fifteen parameters from the 65 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 the analysis.

6. Feasibility study for low-latency deployment of *DeepClean*

To perform noise subtraction in low-latency, 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 4 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,

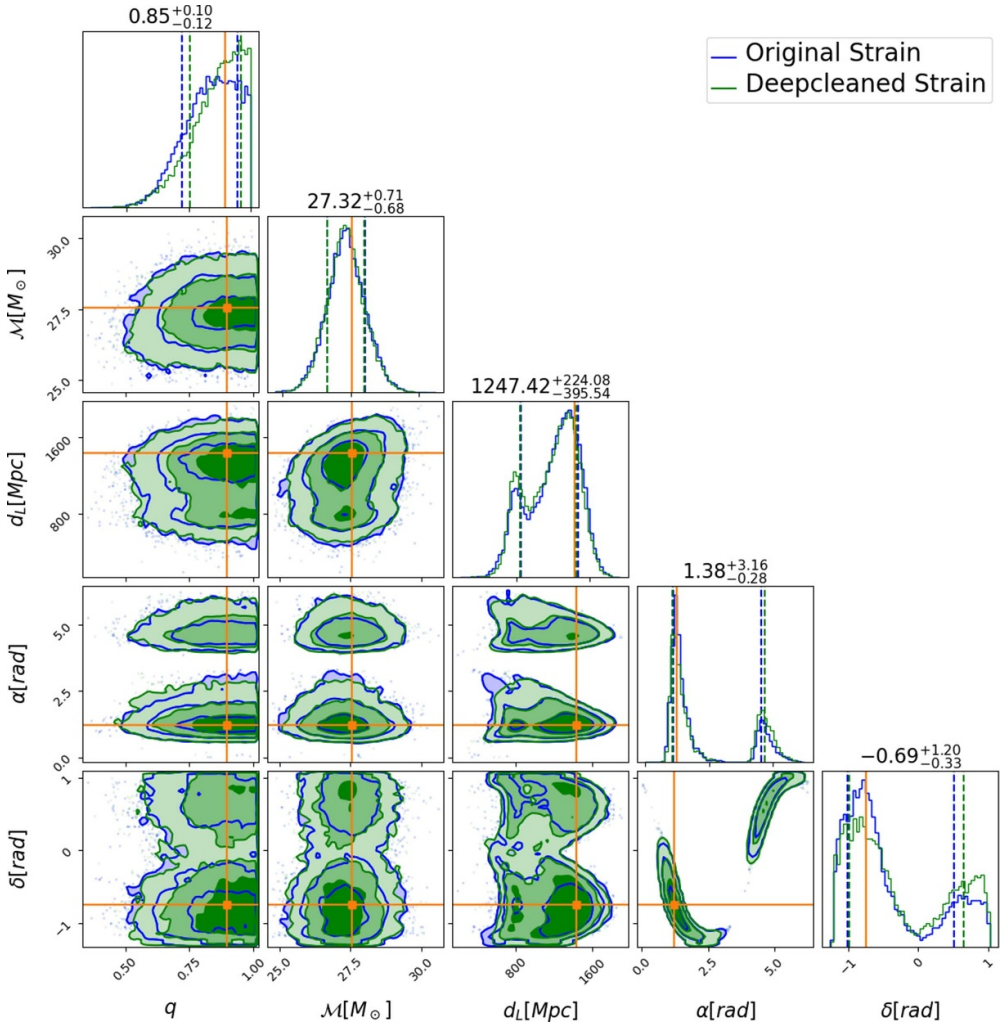

Figure 5. Corner plot showing the posteriors of the mass parameters and the luminosity distance before and after the subtraction of the 60 Hz power-line and their side-bands using *DeepClean*.

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.

6.1. 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. Figure 7 shows that the first and last approximately 0.5 s of a 4 s kernel are 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 segments that are much longer than 4 s so that

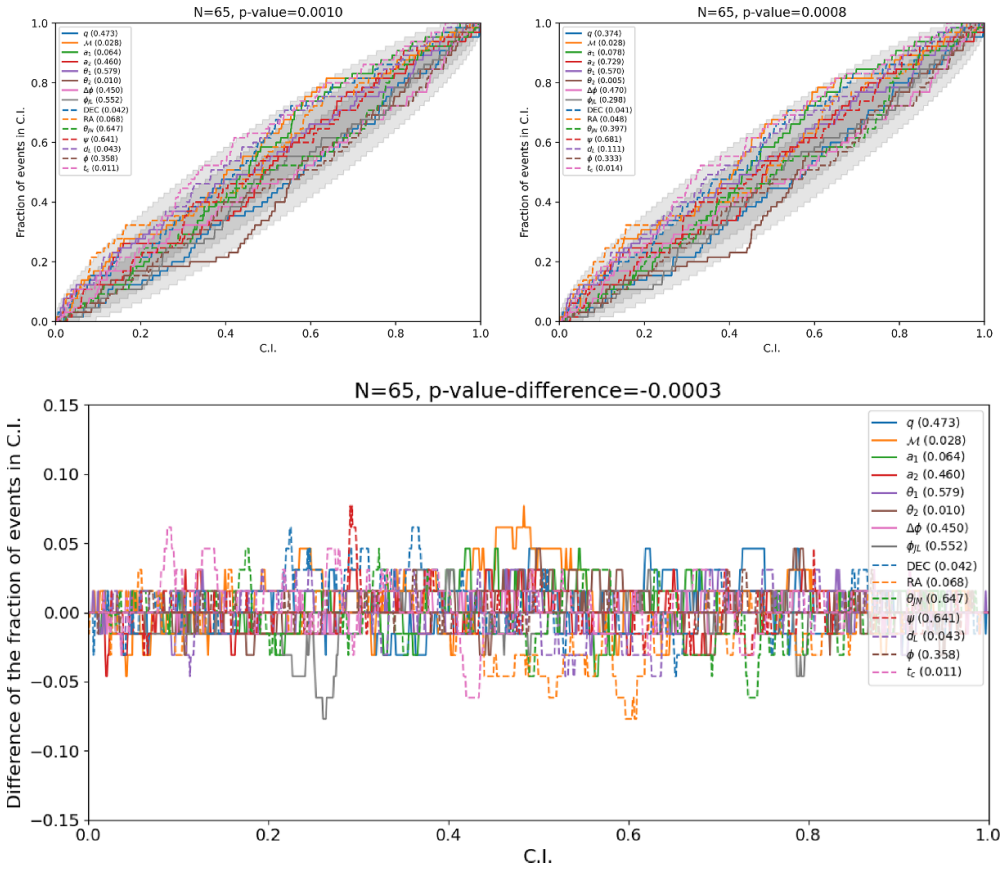

Figure 6. P–P plots generated from the parameter estimation studies of 65 binary black hole injections, comparing the results before (upper left) and after (upper right) the application of the *DeepClean* algorithm for offline noise subtraction. The *x*-axis represents the credible interval, while the *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. The figure below shows the difference of the curves in the P–P plots above (DeepClean—Original). We see that the difference fluctuate around 0 showing the similar distribution of the posteriors in both cases.

the prediction for the target 4 s 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

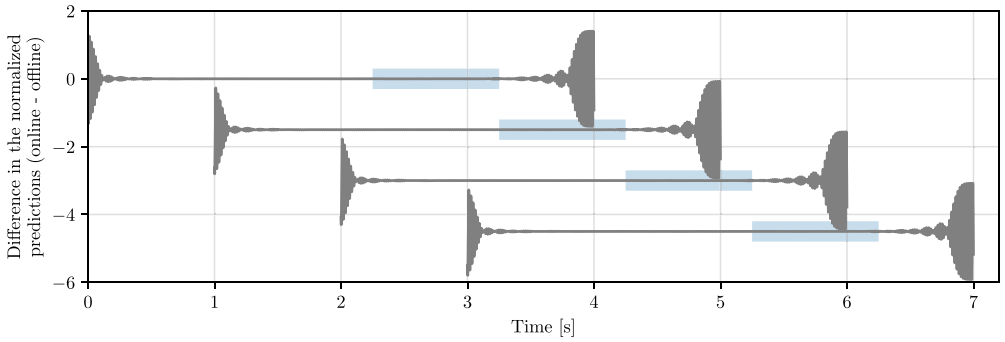

Figure 7. The diagram shows how *DeepClean* will be used for low-latency (online) denoising. The figure compares the online and offline predictions, both normalized, for 4 s duration segments. The offline prediction for the same segment is extracted from the middle of a very long segment of prediction. 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 s of data, consisting of 6 s from the past and 1 s from the future, in addition to the 1 s of target data. This is done for every 1 s frame, and the wait for a future frame causes an additional 1 s of latency. After the prediction, everything except the 1 s target data are discarded from the 8 s segment and written to disk as cleaned frames, ready for downstream analyses.

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.

6.2. A working model for ∼1 s 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 s. To ensure that the quality of our online analysis matches that of our offline analysis, we employ a 4 s kernel that includes data from before as well as after after the 1 s target segment. The additional data ensures that the target segment is located away from the edges of the kernel. The choice of 4 s kernel comes as a standard choice in the *DeepClean* configuration. Ideally, one may place the target segment exactly at the middle of the 4 s segment, however, we push it forward as much as possible, in the interest of reducing the latency. The DeepClean is then applied on the 4 s kernel, and the 1 s target segment is extracted for analysis. A cartoon depicting this is shown in figure 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. Figure 8 displays a scatter plot with the achieved

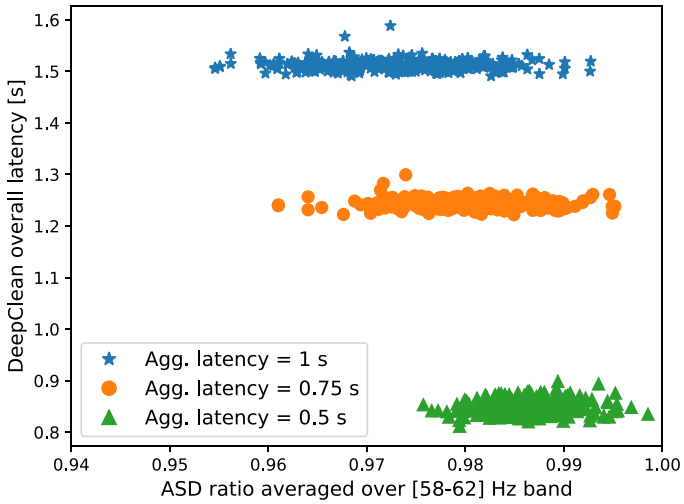

Figure 8. Scatter plot showing the latency vs quality trade-off, produced using O3 Hanford data. The overall latency referred to here is the time taken by DeepClean to produce the output strain after the raw strain is made available. The ASD (amplitude spectral density) ratio on the x-axis is computed 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.

ASD ratio on the x-axis and the overall latency[14] 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.

6.3. A case study for training DeepClean at low latency

To enable low-latency GW 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 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. Figure 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.

[14] 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.

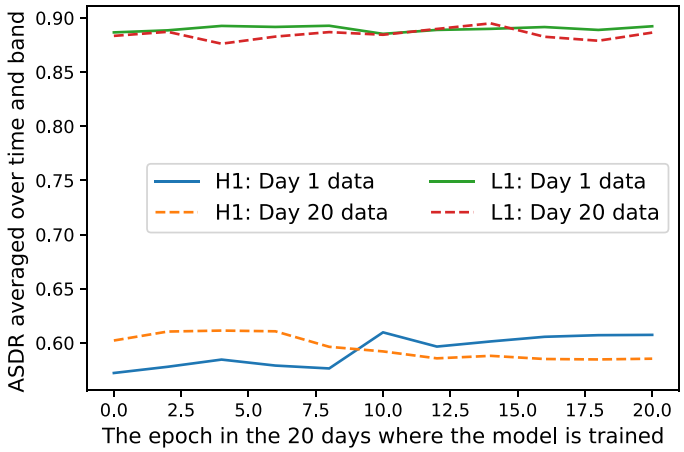

Figure 9. The figure underscores the significance of periodically retraining the neural network, *deepClean*. It presents two traces for each detector: one representing the cleaned strain data from day 1 (solid line) and the other displaying the cleaned strain from day 20 (dashed line), derived from 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) averaged over 58–62 Hz band (which differs from figure 3 for H1). 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.

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 training can be performed in less than $\sim$30 min, making it feasible to update the model once every 30 min or less. The details of that is differed for a later publication.

7. 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 exploring optimizing various factors concerning the neural network architecture. The cleaned data are validated using downstream applications, such as detection and PE 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 s-long segments.

We observed that the specific architecture we used is effective in noise subtraction, however can create some edge effects, causing quality issues for a fraction of a second at the segment edges. With a workaround of additional 1 s latency, 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 though this interval may vary slightly depending on the nature of the coupling. The *DeepClean* deployment described in [60] 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.

Data availability statement

The data that support the findings of this study will be openly available following an embargo at the following URL/DOI: https://git.ligo.org/muhammed.saleem/deepclean-mdc-data.

Acknowledgments

M S, W B, and M C acknowledge the support from the National Science Foundation with Grant Numbers PHY-2010970 and OAC-2117997. D C acknowledges support from NSF Grants Nos. OAC-2117997 and PHY-1764464.

Thanks are due to computational support provided by LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058, PHY-0823459. This material is based upon work supported by the NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation. This research has made use of data obtained from the GW Open Science Center (https://www.gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. Virgo is funded by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale della Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by Polish and Hungarian institutes. This work makes use of NumPy [61], SciPy [62], Matplotlib [63], jupyter [64], corner

[65] software packages. We thank Siddharth Soni for his useful comments on the manuscript. This paper has been assigned the internal LIGO Preprint Number P2300153.

ORCID iDs

Muhammed Saleem https://orcid.org/0000-0002-3836-7751
Tri Nguyen https://orcid.org/0000-0001-6189-8457
Deep Chatterjee https://orcid.org/0000-0003-0038-5468
Ryan Raikman https://orcid.org/0000-0003-4083-6390
Michael Coughlin https://orcid.org/0000-0002-8262-2924
Philip Harris https://orcid.org/0000-0001-8189-3741

References

[1] Aasi J *et al* (The LIGO Scientific Collaboration) 2015 *Class. Quantum Grav.* **32** 074001
[2] Harry G M (for The LIGO Scientific Collaboration) 2010 *Class. Quantum Grav.* **27** 084006
[3] Acernese F *et al* 2015 *Class. Quantum Grav.* **32** 024001
[4] Abbott R *et al* (The LIGO Scientific Collaboration, the Virgo Collaboration, the KAGRA Collaboration) 2021 arXiv:2111.03606
[5] Nitz A H, Kumar S, Wang Y-F, Kastha S, Wu S, Schäfer M, Dhurkunde R and Capano C D 2023 *Astrophys. J.* **946** 59
[6] Olsen S, Venumadhav T, Mushkin J, Roulet J, Zackay B and Zaldarriaga M 2022 *Phys. Rev.* D **106** 043009
[7] Aso Y, Michimura Y, Somiya K, Ando M, Miyakawa O, Sekiguchi T, Tatsumi D and Yamamoto H (The KAGRA Collaboration) 2013 *Phys. Rev.* D **88** 043007
[8] Somiya K (for the KAGRA Collaboration) 2012 *Class. Quantum Grav.* **29** 124007
[9] Cahillane C and Mansell G 2022 *Galaxies* **10** 36
[10] Davis D *et al* 2021 *Class. Quantum Grav.* **38** 135014
[11] Soni S *et al* (The LIGO Scientific Collaboration) 2021 *Class. Quantum Grav.* **38** 025016
[12] Riles K 2013 *Prog. Part. Nucl. Phys.* **68** 1–54
[13] Tiwari V *et al* 2015 *Class. Quantum Grav.* **32** 165014
[14] Nguyen P *et al* 2021 *Class. Quantum Grav.* **38** 145001
[15] Vaseghi S V 2001 Wiener filters *Advanced Digital Signal Processing and Noise Reduction* (Wiley) pp 178–204
[16] Sayed A H 2003 *Fundamentals of Adaptive Filtering* (Wiley)
[17] Davis D, Massinger T J, Lundgren A P, Driggers J C, Urban A L and Nuttall L K 2019 *Class. Quantum Grav.* **36** 055011
[18] Abbott B P *et al* (The LIGO Scientific Collaboration and the Virgo Collaboration) 2020 *Class. Quantum Grav.* **37** 055002
[19] Biswas R *et al* 2013 *Phys. Rev.* D **88** 062003
[20] Zevin M *et al* 2017 *Class. Quantum Grav.* **34** 064003
[21] Mukund N, Abraham S, Kandhasamy S, Mitra S and Philip N S 2017 *Phys. Rev.* D **95** 104059
[22] Vajente G, Huang Y, Isi M, Driggers J C, Kissel J S, Szczepańczyk M J and Vitale S 2020 *Phys. Rev.* D **101**
[23] Ormiston R, Nguyen T, Coughlin M, Adhikari R X and Katsavounidis E 2020 *Phys. Rev. Res.* **2** 033066
[24] Yu H and Adhikari R X 2022 *Front. Artif. Intell.* **5** 811563
[25] Macas R, Lundgren A and Ashton G 2024 *Phys. Rev.* D **109** 062006
[26] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2016 *Phys. Rev. Lett.* **116** 061102
[27] Abbott B *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2017 *Phys. Rev. Lett.* **119** 161101
[28] Metzger B D 2017 *Living Rev. Relativ.* **20** 3
[29] Cannon K *et al* 2012 *Astrophys. J.* **748** 136

[30] Abbott B P *et al* 2019 *Astrophys. J.* **875** 161
[31] Sachdev S *et al* 2020 *Astrophys. J. Lett.* **905** L25
[32] Chu Q *et al* 2022 *Phys. Rev.* D **105** 024023
[33] Yu H, Adhikari R X, Magee R, Sachdev S and Chen Y 2021 *Phys. Rev.* D **104** 062004
[34] Martynov D V *et al* 2016 *Phys. Rev.* D **93** 112004 Martynov D V 2018 *Phys. Rev.* D **97** 059901 (addendum)
[35] Buikema A *et al* 2020 *Phys. Rev.* D **102** 062003
[36] Gunny A, Rankin D, Krupa J, Saleem M, Nguyen T, Coughlin M, Harris P, Katsavounidis E, Timm S and Holzman B 2021 arXiv:2108.12430
[37] Krupa J *et al* 2021 *Mach. Learn.: Sci. Technol.* **2** 035005
[38] Wang M, Yang T, Acosta Flechas M, Harris P, Hawks B, Holzman B, Knoepfel K, Krupa J, Pedro K and Tran N 2020 *Front. Big Data* **3** 604083
[39] Ruder S 2016 arXiv:1609.04747
[40] Abbott R *et al* (LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration) 2023 *Phys. Rev.* X **13** 011048
[41] Abbott R *et al* (The LIGO Scientific Collaboration, The Virgo Collaboration, and The KAGRA Collaboration) 2023 *Astrophys. J. Suppl.* **267** 29
[42] Kingma D P and Ba J 2014 arXiv:1412.6980
[43] Sachdev S *et al* 2019 arXiv:1901.08580
[44] Hanna C *et al* 2020 *Phys. Rev.* D **101** 022003
[45] Messick C *et al* 2017 *Phys. Rev.* D **95** 042001
[46] Tsukada L *et al* 2023 arXiv:2305.06286
[47] Cannon K *et al* 2021 *SoftwareX* **14** 100680
[48] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2016 *Phys. Rev. Lett.* **116** 241103
[49] Abbott B P *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, CAASTRO, 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, ATLAS Telescopes, High Time Resolution Universe Survey, RIMAS, RATIR, SKA South Africa/MeerKAT) 2017 *Astrophys. J. Lett.* **848** L12
[50] LIGO Scientific Collaboration, Virgo Collaboration 2017 *GCN* G298048 (available at: https://gcn.gsfc.nasa.gov/other/G298048.gcn3)
[51] Abbott R *et al* (The LIGO Scientific Collaboration, The Virgo Collaboration, The KAGRA Collaboration) 2023 arXiv:2302.03676
[52] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2016 *Astrophys. J. Suppl. Ser.* **227** 14
[53] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2019 *Phys. Rev.* X **9** 031040
[54] Veitch J *et al* 2015 *Phys. Rev.* D **91** 042003
[55] Ashton G *et al* 2019 *Astrophys. J. Suppl. Ser.* **241** 27
[56] Higson E, Handley W, Hobson M and Lasenby A 2019 *Stat. Comput.* **29** 891–913
[57] Khan S, Husa S, Hannam M, Ohme F, Pürrer M, Jiménez Forteza X and Bohé A 2016 *Phys. Rev.* D **93** 044007
[58] Abbott B P *et al* (KAGRA Collaboration, LIGO Scientific Collaboration and Virgo Collaboration) 2018 *Living Rev. Relativ.* **23** 3
[59] Saleem M *et al* 2022 *Class. Quantum Grav.* **39** 025004
[60] Gunny A, Rankin D, Harris P, Katsavounidis E, Marx E, Saleem M, Coughlin M and Benoit W 2022 A software ecosystem for deploying deep learning in gravitational wave physics *Proc. 12th Workshop on AI and Scientific Computing at Scale Using Flexible Computing Infrastructures (FlexScience'22)* (Association for Computing Machinery) pp 9–17

[61] van der Walt S, Colbert S C and Varoquaux G 2011 *Comput. Sci. Eng.* **13** 22–30
[62] Virtanen P *et al* (SciPy 1.0 Contributors) 2020 *Nat. Methods* **17** 261–72
[63] Hunter J D 2007 *Comput. Sci. Eng.* **9** 90–95
[64] Kluyver T *et al* (Jupyter Development Team) 2016 Jupyter notebooks—a publishing format for reproducible computational workflows *Positioning and Power in Academic Publishing: Players, Agents and Agendas* ed F Loizides and B Scmidt (IOS Press) pp 87–90
[65] Foreman-Mackey D 2016 *J. Open Source Softw.* **1** 24