

Wavelet scattering transform for gravitational wave analysis: An application to glitch characterization

Original

Wavelet scattering transform for gravitational wave analysis: An application to glitch characterization / Licciardi, A., Carbone, D., Rondoni, L., Nagar, A.. - In: PHYSICAL REVIEW D. - ISSN 2470-0010. - 111:8(2025), pp. 1-10. [10.1103/PhysRevD.111.084044]

Availability:

This version is available at: 11583/2999476 since: 2025-04-23T14:20:05Z

Publisher:

American Physical Society - APS

Published

DOI:10.1103/PhysRevD.111.084044

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

APS postprint/Author's Accepted Manuscript e postprint versione editoriale/Version of Record

This article appeared in PHYSICAL REVIEW D, 2025, 111, 8, and may be found at <http://dx.doi.org/10.1103/PhysRevD.111.084044>. Copyright 2025 American Physical Society

(Article begins on next page)

Wavelet Scattering Transform for Gravitational Waves Analysis. An Application to Glitch Characterization

Alessandro Licciardi

*Department of Mathematical Sciences, Politecnico di Torino, 10129 Torino, Italy and
INFN Sezione di Torino, 10125 Torino, Italy*

Davide Carbone

*Laboratoire de Physique de l'Ecole Normale Supérieure, ENS Université PSL,
CNRS, Sorbonne Université, Université de Paris, Paris, France*

Lamberto Rondoni

*Department of Mathematical Sciences, Politecnico di Torino, 10129 Torino, Italy and
INFN Sezione di Torino, Torino, Italy*

Alessandro Nagar

*INFN Sezione di Torino, 10125 Torino, Italy and
Institut des Hautes Etudes Scientifiques, 91440 Bures-sur-Yvette, France*

Gravitational waves, first predicted by Albert Einstein within the framework of general relativity, were confirmed in 2015 by the LIGO/Virgo collaboration, marking a pivotal breakthrough in astrophysics. Despite this achievement, a key challenge remains in distinguishing true gravitational wave signals from noise artifacts, or "glitches," which can distort data and affect the quality of observations. Current state-of-the-art methods, such as the Q-transform, are widely used for signal processing, but face limitations when addressing certain types of signals. In this study, we investigate the Wavelet Scattering Transform (WST), a recent signal analysis method, as a complementary approach. Theoretical motivation for WST arises from its stability under signal deformations and its equivariance properties, which make it particularly suited for the complex nature of gravitational wave data. Our experiments on the LIGO O1a dataset show that WST simplifies classification tasks and enables the use of more efficient architectures compared to traditional methods. Furthermore, we explore the potential benefits of integrating WST with the Q-transform, demonstrating that ensemble methods exploiting both techniques can capture complementary features of the signal and improve overall performance. This work contributes to advancing machine learning applications in gravitational wave analysis, introducing refined preprocessing techniques that improve signal detection and classification.

I. INTRODUCTION

Gravitational waves, predicted by Albert Einstein in the framework of general relativity [1], stand as one of the most significant discoveries of the past century. The experimental confirmation of these spacetime ripples, first achieved in 2015 by the LIGO/Virgo collaboration [2], has opened a new observational window into the universe, particularly in the context of multimessenger analysis of astrophysical data [3]. However, despite remarkable technical progress in gravitational wave detection, we are faced with a fundamental challenge: the identification of true signals in opposition to the so-called "glitches". Glitches represent interruptions or anomalies in the detected data that can distort signals of interest and compromise the accuracy of observations. These undesired phenomena may stem from a variety of sources, including environmental interference, instrumental errors, or data analysis issues. Establishing a taxonomy of such signals is fundamental to discerning between them and the true events; in fact, because of the non-gaussianity [4], standard preprocessing methods usually fail to remove glitches from the time series.

Most of the analysis of such spurious signals is based on spectrograms obtained through the Q-transform [5], and even Bayesian methods make use of wavelet-like transforms [6]. More recently, time-frequency plots were collected in datasets, as Gravity Spy [7], and used in the context of machine learning techniques used for image classification [8][9]. Let us stress that the use of Q-transform is standard in the community; see, for instance, the Python libraries GWpy [10] and PyCBC [11]. The result is that most of the statistical and machine learning analyses directly exploit images.

In other contexts, the analysis of complex signals like natural images or sounds has benefited from the use of more advanced preprocessing methods. Among all, the recently developed Wavelet Scattering Transform (WST) [12][13] has sparked particular interest within the machine learning community. Contrary to standard wavelet or Fourier methods, the WST generates representations that enjoy stability properties with respect to small rotations and deformations. The basic aim of WST is to map elements which are close in the data space (e.g. a natural image or a sound) into representations close in the target, e.g. frequency, space. This request seems

to be simple and fundamental for classification and clustering tasks, but standard methods used for temporal series, for instance Q-transform and short time Fourier transform, do not enjoy such stability properties. In this sense, WST has been shown to outperform standard preprocessing methods for 1D or 2D data, e.g. for music classification [14], ECG tracks [15], bioacoustics [16], observational cosmology [17][18] and deep learning application [19].

In the present work we aim to evaluate the performance of WST as opposed to Q-transform in the context of Virgo data, and in particular for classification and clustering tasks on real glitches data. As expected, we show that the intracluster dispersion is critically reduced by WST; consequently, classification is made simpler and possible even with simple architecture, contrarily to the CNN needed for standard spectrograms obtained from Q-transform. In Section II we review the theoretical definition of both preprocessing methods; in Section IV we present the experimental results which evidently shows a critical advantage in using WST in place of Q-transform.

RELATED WORK

The first papers on early Machine Learning techniques in the context of gravitational waves analysis trace back to the late 2010s. Initially, the focus was on unsupervised hierarchical learning [20], early classification techniques [21], or simply the advantages of using GPUs for statistical analysis [22]. A review [23] on Machine Learning for quantum measurement partially predicted the importance of ML applied to gravitational waves, and its broader application in Astrophysics was highlighted already in [24][25].

With the advancements in machine learning techniques, the number of works applying ML to discriminate gravitational waves from noise artifacts increased as seen in [26][27], until the milestone work on Gravity Spy dataset [7]. The existence of a benchmark dataset dramatically increased the scientific production in the area; see, for instance, the review [28]. The use of CNN follows state of the art in computer science [29], and in spite of a usual tradition in the ML community, even open competitions regarding gravitational-data search were proposed, e.g. [30].

Regarding preprocessing methods, the use of time-frequency spectrograms for gravitational waves traces back to [5]. Among all the proposed methods, the first work highlighting the use of wavelets [31] and Q-Transform [32] is [33]; the two proposed methods become a sort of standard. For instance, Gravity Spy project made use of spectrograms and not of bare time-series. Just recently, the use of filtering methods based on ML has been proposed in [34], even as generative tool. The coherent WaveBurst algorithm [35], widely used by the LVK collaboration, is a key wavelet-based approach for gravitational wave data analysis. More recently, the

multi-resolution Wavescan algorithm [36] has been introduced to enhance the Q-transform by integrating multiple resolutions into a unified framework.

Regarding Wavelet Scattering Transform [12]: its interpretability has been exploited in Physics Informed ML [37] and for neural network analysis [38]. The fields of application of such method are numerous; for the sake of the reader and with an eye to gravitational waves, we mention some regarding 1D signals: from music genre classification [14], to EEG [39] and ECG [15] in medicine, and in general to time-series analysis [40].

II. THEORY: WAVELET SCATTERING TRANSFORM

In this section we present the standard definitions of Q-transform and WST; since the experimental analysis will focus on 1D data, for convenience of the reader we recall just the properties of WST in such field of application, even if it can be defined for data in higher dimension (as 2D or 3D images).

A. State-of-the-art Representation: Q-Transform

The Q-transform, also known as the Constant Q-transform (CQT), is the most common technique for representing gravitational waves [5][33]. It is a time-frequency analysis technique that provides a representation of a signal in the joint time-frequency domain with a constant quality factor, denoted as Q .

Let $h(t)$ be a continuous-time signal, and consider a family of window functions $g_Q(t)$, such as Hann window functions, parameterized by a quality factor Q . The window functions have a constant bandwidth, centered at logarithmically spaced frequencies ω_k .

Given a 1D signal $h(t)$, the Q-transform coefficient $H_k(t)$ at time t , for a specific frequency f_k and factor Q is obtained by the convolution:

$$H_k(t) = \int_{-\infty}^{\infty} h(\tau) g_Q(t - \tau) e^{-if_k\tau} d\tau, \quad (1)$$

To obtain the full Q-transform of the signal, we compute the Q-transform coefficients for a range of frequencies f_k in the chosen frequency grid. The resulting Q-transform provides a time-frequency representation of the signal, where the time axis corresponds to the original signal's time domain, the frequency axis represents the logarithmically spaced f_k and the color map refers to the module of the coefficients H_k . This kind of plot is referred to as spectrogram, see Figure 1. The Q-transform is particularly useful for analyzing signals with varying frequency content, as it provides higher resolution at lower frequencies and lower resolution at higher frequencies. This logarithmic frequency scaling ensures that the Q-transform captures the details of signals with complex harmonic structures and varying frequency components [32]. It,

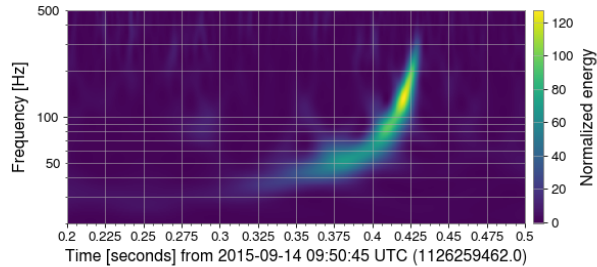


FIG. 1. Spectrogram of GW150914 taken from GWpy repository <https://gwpy.github.io>, the first gravitational wave signal ever detected [2].

or its modification, enables the analysis of signals with complex frequency content and is widely applied in various domains, including audio processing, music analysis [41][42], and signal processing tasks [43].

It is possible to delineate similarities and dissimilarities with the Short Time Fourier Transform (STFT) [44]. On one hand, the STFT divides a signal into short, overlapping segments and applies the Fourier Transform to each segment. This results in a time-frequency representation where the time axis is preserved, but the frequency resolution is constant across all frequencies. The STFT provides good frequency localization at the expense of time resolution.

On the other hand, the Q-transform uses a family of window functions with constant bandwidths and logarithmically spaced frequencies, allowing for higher frequency resolution at lower frequencies and lower resolution at higher frequencies. The Q-transform provides a time-frequency representation with a constant quality factor, denoted as Q , which captures the details of signals with complex harmonic structures and varying frequency components. Therefore, the main difference between the STFT and the Q-transform lies in their frequency resolution characteristics. The STFT offers a constant frequency resolution but sacrifices time resolution, while the Q-transform provides variable frequency resolution with a constant quality factor, allowing for a better representation of signals with varying frequency content, such as gravitational waves [5].

B. Wavelet Scattering Transform for time series

The Wavelet Scattering Transform (WST) [12] is a mathematical operator that allows to obtain for a given signal a stable and invariant representation. In particular, under suitable hypothesis [13], the representation is translation invariant, stable to additive noise, i.e. it is non-expansive, and stable to deformations; this property is mathematically expressed in its original derivation as Lipschitz-continuity under the action of C^2 -diffeomorphisms. Utilizing a representation op-

erator with these properties in a machine learning scenario could significantly reduce the computational effort required in training classification algorithms [45].

The construction of the WST on one dimensional signals relies on properly defining a family of wavelet functions, starting from a mother wavelet $\psi \in L^2(\mathbb{R}, dx)$. Let $a > 1$ be a scalar. A family of wavelet can be defined as

$$\psi_\lambda(t) = \lambda^{-1} \psi(\lambda^{-1}t) \quad \lambda \in a^{\mathbb{Z}}, \quad (2)$$

where $\lambda = a^j$. In practice, for greater values of j the wavelet has a larger support in time domain, hence a smaller bandwidth in frequency domain. In the usual definition of WST, they define $A \in \mathbb{N}$ such that $a = 2^{1/A}$; this will play a role of a hyperparameter. Furthermore, we fix a maximum *depth* $J \in \mathbb{Z}$, corresponding to a set of allowed scaling operators λ , i.e. belonging to the set

$$\Lambda_J = \{\lambda \in a^{\mathbb{Z}} : |\lambda| = a^j \leq 2^J\}. \quad (3)$$

The couple (J, A) contains the hyperparameters of the WST, analogously to Q for the Q-transform.

In order to establish the desired stability properties, the WST exploits the non-linearity introduced by the so-called *propagator operator* [12]. The propagator operator over a path $(\lambda_1, \dots, \lambda_m)$, with $\lambda_i \in \Lambda_J$, applied to a given signal $h(t)$, is defined as

$$U[p]h(t) = U[\lambda_m] \dots U[\lambda_1]h(t), \quad (4)$$

where, denoting with \star the convolution, $U[\lambda]h(t) = |\psi_\lambda \star h(t)|$. Therefore the propagator operator cascades convolutions and moduli – this attempt tries to emulate the structure of a layer of a convolutional neural network [46]. The WST of a signal $h(t) \in L^1(\mathbb{R}, dx)$ over a path $p = (\lambda_1, \dots, \lambda_m)$, with $\lambda_i \in \Lambda_J$, is defined as

$$S_J[p]h(t) = U[p]h(t) \star \phi_J(t), \quad (5)$$

where $\phi_J(t)$ is a low-pass filter, scaled according to the depth parameter $J \in \mathbb{Z}$. Namely

$$S_J[p]h(t) = |\psi_{\lambda_m} \star \dots \star \psi_{\lambda_2} \star \psi_{\lambda_1} \star h| \dots \star \phi_J(t). \quad (6)$$

For the conducted experiments we couple *Morlet wavelets* with a Gaussian low-pass filter [31].

As in [12], introducing the path set up to length m , $\Lambda_J^m = \{(\lambda_1, \dots, \lambda_m) : |\lambda_i| = a^j \leq 2^J\}$, it is possible to define the induced norm of the scattering operator over the set $\mathcal{P}_J = \bigcup \Lambda_J^m$, i.e.

$$\|S_J[\mathcal{P}_J]h\| = \sum_{p \in \mathcal{P}_J} \|S_J[p]h\| \quad (7)$$

where $\|\cdot\|$ stands for the L^2 -norm. For fixed J and A , and given the definition of Λ_J^m , the WST can be graphically visualized as an iterative tree process, see Figure 2. One could be concerned about the depth requested in practice, but in different experiment [45] it has been showed that just 2 or 3 layers of WST are sufficient to

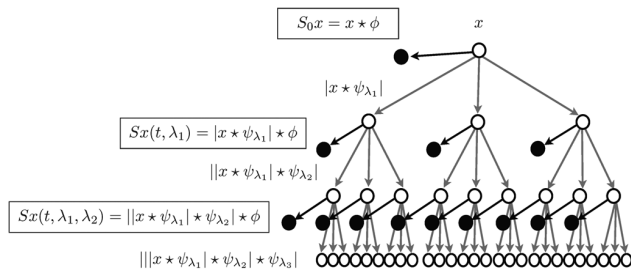


FIG. 2. Visualization of Wavelet Scattering Transform as an iterative process, from [14]. In their notation the signal is $x(t) = h(t)$ the path p at depth m is explicated in parentheses as a tuple $(\lambda_1, \dots, \lambda_m)$.

represent around 98% of the energy of the signal. Indeed the energy of each layer, i.e. $\|U[\Lambda^m]\|$, is empirically observed to rapidly converge to zero. The norm induced by the WST is *non-expansive*, therefore the representation operator is stable to additive perturbation. Hence, for any signal $h \in L^2(\mathbb{R}, dt)$ and any perturbed version $h' = h + \epsilon$, it holds

$$\|S_J[\mathcal{P}_J]h' - S_J[\mathcal{P}_J]h\| \leq \|h' - h\| = \|\epsilon\|. \quad (8)$$

Moreover through the WST operator is possible to achieve translation invariance [13], i.e. for any signal $h(t) \in L^2(\mathbb{R}, dx)$ and any translated version $h'(t + a)$ it holds

$$\lim_{J \rightarrow \infty} \|S_J[\mathcal{P}_J]h' - S_J[\mathcal{P}_J]h\| = 0. \quad (9)$$

Further theoretical results [12] show that, when hypothesis are met, the representation is stable to continuous domain stretches and displacement fields. These properties are expected to become particularly relevant when studying gravitational waves data, where the information is highly affected by noise, due to the high sensitivity of the interferometers.

We stress that centering the peak of the signal before preprocessing is a core problem is glitch analysis; since WST is less sensible to time shifts and deformations, we expect it to be helpful against this issue.

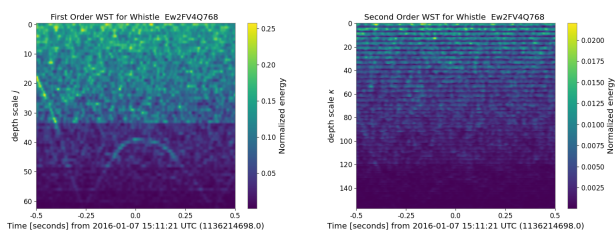


FIG. 3. First and second order WST for the glitch with ID Ew2FV4Q768. The plots are not normalized and no customized color map is applied.

C. Comparison Between Q-Transform and WST

The main takeaway is that in several scenarios, such as 1D time series classification, machine learning models can achieve state-of-the-art performance with smaller architectures when using WST [13]. These models are consequently easier to train due to a reduced number of trainable parameters. Mathematically, this advantage arises from WST's ability to map data into a feature space that enhances separability by satisfying the following properties:

- i stability to additive noise
- ii local translation invariance
- iii stability to small continuous time warping

By construction, WST satisfies these properties, theoretically ensuring better data separation. In contrast, the Q-transform and standard wavelet transforms do not generally satisfy these properties, which can result in less robust feature representations—for example, small additive noise may significantly alter the feature space representation. The key structural differences arise from WST's non-linearity and recursive nature. While the Q-transform (Eq. (1)) is a linear operator based on convolution with a specific window function, linearity alone prevents it from achieving properties (i), (ii), and (iii). In contrast, first-order WST introduces non-linearity through the modulus operation and applies an averaging step using a low-pass filter, leading to improved feature stability and separability. As far as the analysis led in our study is concerned, other differences are in the hyper-parameters, such as the choice of the wavelets, or the frequency resolution scale.

III. TRAINING AND TEST DATASETS SET-UP

In this section, our objective is to conduct a comprehensive comparison of the data analysis between the Q-transform and the WST. We reconstruct and explicitly describe the pipeline for the preprocessing adopted in [8]. Notably, the application of WST as a theoretical tool is already prevalent in diverse fields such as cosmology [18] and field theory [47]. As a widely acknowledged principle in the literature [13], WST is preferable to STFT methods when their performances are comparable, primarily due to the invariance properties that facilitate cross-signal interpretation.

A. Gravity Spy 1.1.0

In this study, we use Gravity Spy 1.1.0 [8] as a foundational dataset for our research. The database is public and available at <https://zenodo.org/records/1486046> and contains a collection of glitches, alongside

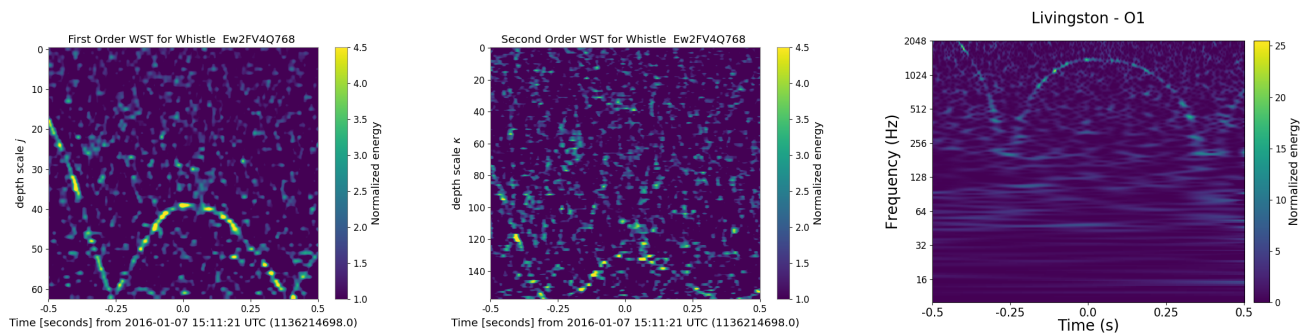


FIG. 4. *Left and Center*: First and second order WST for the glitch with ID `Ew2FV4Q768`. The plots are normalized row-wise and the original signal is not downsampled to 2048 Hz. For the sake of visualization, we used $J, A = (6, 16)$ and the customized color map (10) is applied.

with a detailed description of its content. For readers' convenience, we report the main features of the available material. From the highlighted source, it is possible to download a collection of images containing the spectrograms computed with Q-transform. Each image is identified by an ID and it is possible to associate the corresponding time series using the available csv file. Then, every time series can be downloaded using the method `fetch_open_data()` from GwPy library.

Our first objective was to recompute the presented spectrograms as sanity check using an independent pipeline. Before proceeding to the details of the preprocessing, some time series present in the csv file contains NaN entries when downloaded because gated before the public release, and so the corresponding spectrograms in the public dataset are empty; as for instance in the glitch with id `rymrZrzFCx` in the class *Extremely Loud* (see Figure 5). Some signals appear to have a sample rate of 4096 Hz; removing such time series is necessary since the `q_transform` method in GwPy requires higher sample rate to build a spectrogram with maximum frequency 2048 Hz (see GwPy documentation for detailed information).

The first passage is then to detect and remove such time series from the dataset. In the present work, we consider the dataset as presented in the original paper, composed by 22 classes; however, after the removal of NaN instances, the number of classes reduces to 21 (see Figure 6).

B. Data processing

We download the time series indexed in the file `trainingset_v1d1_metadata.csv`, with a window of $\pm 2s$ centered in the `event_time` attribute of each signal. As briefly discussed in the previous subsection, we then seek for NaN entries in the time series; hence, we remove such instances from the list of time series. The second step is whitening, performed with the corresponding `whiten()` method in GwPy library. After these introductory passages, we have raw time series with sample

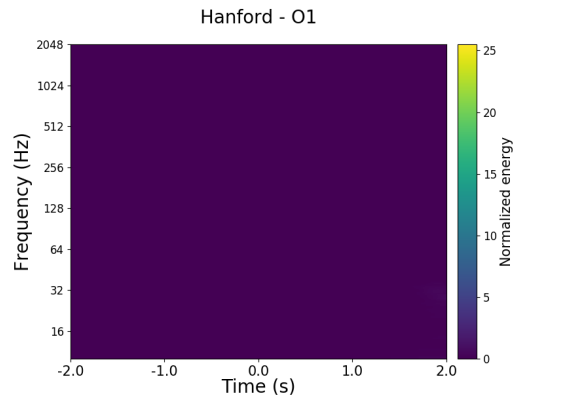


FIG. 5. Image corresponding to the file `H1_rymrZrzFCx_spectrogram_4.0.png` in the dataset of Gravity Spy 1.1.0.

rate of 16384 Hz. We now separately present the pipeline for Q-transform and spectrogram generation, as inferred from [8]. Subsequently, we present the procedure for WST.

a. Q-transform. We use the method `q_transform()` present in GwPy library to compute the spectrogram from the `TimeSeries` objects we obtained from the previous steps. As for the arguments of such methods, we followed [8] which adopts: $qrange = (4, 64)$ for the range of Q search. We recall that the obtained spectrogram is in fact the result of a tiling procedure, as described in GwPy documentation. Then, $fres = 470$ and $tres = 2 * \delta_t / 570$ in order to obtain a spectrogram of dimension 470×570 , where $\delta_t = 2s$ is the semi-length of the time window. Coherently, the `outseg` is chosen to be the segment centered in the `event_time` with with size $2\delta_t$. Subsequently, we fix $frange = (10, 2048)$ for the frequency range, $whiten = False$ since already performed and $logf = True$ to span frequency range in logarithmic scale. At last, we normalized the obtained spectrogram with respect to the mean using $norm = 'mean'$. We stress again that by construction of the GwPy Q-transform, we

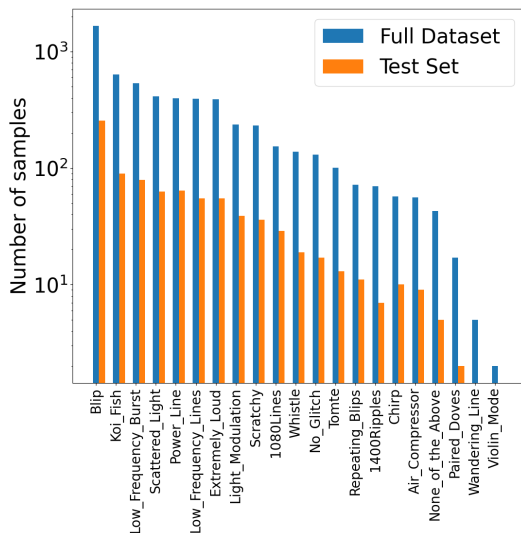


FIG. 6. Number of samples per class after eliminating the time series containing NaN values and the four observations sampled at 4096 Hz, in log-scale and sorted in decreasing order. The dataset is very imbalanced: *Blip* is the most represented glitch and contains 1668 instances, while the smallest, *Violin Mode*, has just 2 samples in the training set.

are forced to preprocess signal with high sample rate even if the frequency axis is capped at 2048 Hz.

At this point, as suggested in the original paper, the image is reshaped to the size 140×170 ; we use the method `Resize` from Pytorch library for this task. The final step is the choice of the normalization, that is of the color map present in the effective plots. We stress that every step modifying the image is fundamental since the final task will be image classification; there is no information, when performing a machine learning task, about the original time series. For instance, the classification accuracy could be dramatically different for a classifier trained on image having linear, and not logarithmic, frequency scale. The same occur for the choice of the color map, since the spectrogram can appear really different. Coherently with the attribute `cmap` present in `Matplotlib` package, we fix an upper and lower cap such that $0 < c_{min} < c_{max}$ and we remap the value of normalized energy p of each pixel using the following piecewise function:

$$f(p) = \begin{cases} c_{max} & \text{if } p \geq c_{max} \\ p & \text{if } c_{min} < p < c_{max} \\ c_{min} & \text{if } p \leq c_{min} \end{cases} \quad (10)$$

This function represent a capped linear color map. In Figure 7 we show how different values of the bounds can dramatically impact the appearance of the image. Moreover, we also present the comparison with the corresponding one present in the image folder of the downloaded dataset. As a side note, every image is plotted with a forced squared aspect ratio; such choice is maintained throughout the whole paper.

b. Wavelet Scattering Transform. For each signal, we compute the Wavelet Scattering Transform (WST) up to the second order on a time window of 4 s centered in the event time. Since the Q-transform cap the frequency at 2048 Hz, we decided to downsample the signal at that frequency before computing the WST. We choose $(J, A) = (7, 7)$ as hyperparameters for both orders. The zeroth-order WST, which provides no informative content, was excluded from the analysis. As far as the dimensions of the resultant images is concerned, for the chosen configuration the dimensions for the first and second order are 42×128 and 114×128 , respectively.

We plot in Figure 3 the unnormalized first and second order WST for a chosen signal before downsampling. Let us focus for a moment on the meaning of such plots: while the content of the time axis is evident, we recall that each depth parameter, for the first order, correspond to a precise bandwidth of the wavelet used for the convolution operation. For the second order, we use an identification rule $(j_1, j_2) \rightarrow \kappa$ to map the couples of admitted depths in a single value. Their definition was introduced in Section II.

As evident from Figure 3, a normalization appears to be necessary, even at first glance. Analogously with Q-transform, we normalize with respect to the mean; the main dissimilarity is that such normalization is performed row-wise, that is for every value of j and κ . Despite the fact that the use of the color map seems to be suggested also for WST (e.g. Figure 4), we noticed that better classification performance (see next Section) are obtained without the use of color map. We stress as this is not a limitation; on the contrary, classification results based on the choice of the color map, as for Q-transform, can be problematic since based on the human eye.

In conclusion, for the sake of a direct comparison with Q-transform, we reshape the size 140×170 . We stress as for WST we are actually performing an interpolation, since the original dimension along depth axis is lower than 140.

C. Training and Test Datasets

To ensure a rigorous validation experiment, the cleaned and preprocessed dataset was split into two distinct subsets. Specifically, following the original split proposed in the csv file, 70% of the data samples were allocated for model's training, while the remaining 30% were equally split in validation (15%) and test (15%). We adopted their same split for the sake of a precise comparison. As per standard practice, generalization capability is assessed using the test set, which is strictly excluded from the training process of the classifiers.

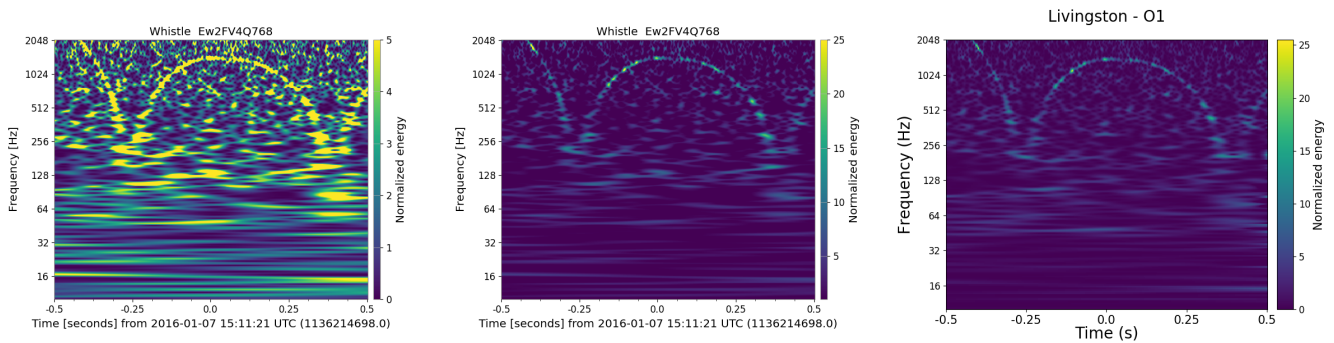


FIG. 7. Spectrograms corresponding to the same signal (id: Ew2FV4Q768) with different values of c_{min} and c_{max} – the same color map and corresponding image in Gravity Spy are used for consistency. *Left*: $c_{min} = 0$ and $c_{max} = 5$. *Center*: $c_{min} = 1$ and $c_{max} = 25$. *Right*: image taken from Gravity Spy dataset.

D. Software and Computational Resources

The Q-transform is computed utilizing the GwPy library, whereas the Wavelet Scattering Transform (WST) is performed using the `Kymatio` library [48]. The training of the classifier is executed on the high-performance computing (HPC) facilities of Department of Mathematical Sciences (DISMA) at Politecnico di Torino.

E. Classifier

For the sake of the present work, we use a simple Random Forest Classifier, as taken from CuML Python library. We use it separately on Q-transform, first order and second order of WST. Then, coherently with the proposal of [8], we apply a merge method to use the best of each preprocessing. In particular, we readopt the *hard merge*, which corresponds to build a convex combination of the predicted probability vector on the validation set, i.e. for p_1 and p_2 vectors of probability predictions

$$p_{merge} = \alpha p_1 + (1 - \alpha) p_2$$

The test set is *never used* for hyperparameter selection; then we perform a grid search on the hyperparameter $0 \leq \alpha \leq 1$ in order to maximize the accuracy. We perform such merge procedure firstly between first and second order of WST, that we name for the sake of brevity "S1+S2"; then we define a second convex combination between the probability predictions using "S1+S2" and using the Q-transform. An immediate follow up for our analysis is the use of CNNs, as proposed in [9].

IV. EXPERIMENTAL RESULTS

We present in Table I the accuracy result on the test set for the classifiers on the first order WST, on the second order WST, on Q-transform, on "S1+S2" and on the total merge of WST + Q-transform.

Evidently, for the dataset in study there is no statistical advantage in the use of Q-transform over WST. However, the possibility of having an independent preprocessing methods allows to combine them with ensemble methods, such for example an hard merge. The final result we obtain is an improvement of 3% of the classification accuracy on the test set.

In details, second order WST slightly increases the performance of the first order of the WST but a detailed grid search on (J, A) could reveal more insights about this fact. Combining the two orders, i.e. in "S1+S2", leads to an notable improvement with respect to the state-of-the-art. Moreover, as depicted in Table II, where we compare Q-transform and WST by changing the color-scale of the representation, it is possible to see that the WST consistently performs better than the Q-transform. However, when the "S1+S2" classifier is merged with the Random Forest trained on Q-transform we manage to obtain an improvement of 2% in accuracy.

Preprocessing Method	Accuracy (%)
First Order WST	88.11
Second Order WST	68.65
Total WST	88.57
Q-transform	87.41
WST + Q-transform	91.03

TABLE I. Test set accuracy for different preprocessing methods. In this case we did not applied a colormap on the WST, while the Q-transform had been rescaled to $(0, 25)$, as in [49]. With these hyper-parameters the WST improves the performance of around 1%. It is interesting to see that the best performance had been achieved by combining WST and Q-transform, improving the test accuracy to 91.03%.

A. Ablation on the Colour Scale

In this section we show that WST method is robust with respect to changes in the color scale, while, on the

other hand the Q-transform method is more sensitive to this further pre-processing step. In particular, in accordance to what reported in [49] Q-transform achieve the best results when its color map is changed to (0, 25), while WST has its best performance without imposing a color-scale. In Table II we present the results obtained with different color scales.

Computational Cost

Aside from the possible use in ensemble methods, a fundamental comparison between Q-transform and WST regards the computational cost and speed. Actually, this provides a further motivation for the adoption of WST when possible. In fact, the possibility to perform real time analysis of the signals is a fundamental problem in gravitational wave detection and processing [50][51]. The comparison between GwPy and Kymatio on CPU seems to show that the computational cost is similar. However, WST performed on GPU clearly outperforms Q-transform in term of computational time. We stress as the GPU we used are not state-of-the-art in term of performance, and so that such result can be possibly improved.

Moreover, there is another clear advantage in using WST on GPU: the latter can be computed by batch, while Q-transform has to be computed sample by sample. The benefit in terms of computational time is critical. Specifically, the computation times are 0.211 ± 0.003 seconds per sample for the Q-transform (GwPy, CPU), 0.197 ± 0.007 seconds for WST (Kymatio with NumPy, CPU), and 0.063 ± 0.002 seconds for WST (Kymatio with PyTorch, GPU), indicating that the GPU-based WST implementation is approximately $3.13 \times$ faster than the CPU-based WST and $3.35 \times$ faster than the Q-transform. The total number of considered samples is 100 and the standard error is used to quantify the uncertainty. For instance, on GPU for a batch of 100 samples the computational time is 0.063, which is basically equivalent to the time needed per one single sample. Such results represent a motivation towards the use of WST is glitch classification, or in general in time series analysis, especially when a machine learning task, and so batching, is involved.

Apart from the computational time associated with the two pre-processing methods, the cost of the two methodologies is comparable. In particular, the implementation of the WST depends on the fast scattering algorithm [14], which utilizes a highly efficient embedding for convolutions. The computational cost of the first-order WST is $\mathcal{O}(JAT \log T)$, where T represents the number of time stamps of the signals, and J, A are the WST hyperparameters. Conversely, extracting features from the second order of the WST incurs a cost of $\mathcal{O}(J^2 AT \log T)$ due to an increase in the number of scattering coefficients.

While documentation on the Q-transform implementation of GwPy is limited, it is plausible that its cost can be represented as $\mathcal{O}(QMT \log T)$, where Q denotes the number of tiles comprising the Q-tile, and M signifies the number of windows. We assessed the efficiency of the two representations by comparing the number of elementary operations and functions required to transform the same signal, utilizing the `cProfile`, which resulted in 258,761 for the Q-transform versus 14,374 for the WST. This comparison highlights the efficiency of the WST.

V. CONCLUSIONS

In this study, we conduct a thorough comparison between the Q-transform, the current leading preprocessing method for gravitational wave analysis, and the Wavelet Scattering Transform (WST), a recently developed mathematical tool for data preprocessing. Our work reveals how, for the dataset available, the two preprocessing methods leads to analogous results in term of classification accuracy. Notably, WST seems to bypass the issue of the color scaling of the spectrogram, which is potentially subtle since based on human eye. Plus, we stress as the fundamental advantage of WST stems in a notable reduction in computational time, surpassing Q-transform by more than one order of magnitude. Moreover, WST is completely compatible with `TensorFlow` and `Pytorch`, making it naturally applicable in deep learning applications. We believe that a deeper investigation of the hyperparameters of WST, coherently with a more systematic construction of the dataset, could possibly provide even better motivation for the use of WST. Moreover, being WST a different independent preprocessing method, we show how one can improve the classification accuracy with a merge method, considering the best of two worlds. Looking ahead, future directions necessitate testing WST on other collections of data of glitches and gravitational waves to further validate our results. While we anticipate no conceptual challenges, a larger dataset would enable more comprehensive testing in classification, potentially achieving state-of-the-art performance with simplified architectures as SVM or even CNN.

ACKNOWLEDGEMENTS

We are grateful to L. Asprea, F. Legger, F. Sarandrea and S. Vallero for discussions and help in getting access to Virgo data at the beginning of these project. A.L., D.C. and L.R. worked under the auspices of Italian National Group of Mathematical Physics (GNFM) of INdAM. A.L. is part of the project PNRR-NGEU which has received funding from the MUR – DM 117/2023.

c_{max}	First Order WST	Second Order WST	Total WST	Q-transform	WST + Q-transform
1	81.35 %	63.75 %	81.47%	75.17%	82.28%
25	88.11%	68.65%	88.57%	87.41%	91.03%
None	88.11%	68.65%	88.57%	87.30%	90.56%

TABLE II. The table presents the test accuracy of various pre-processing methods as correlated with differing maximum values of the color scale. The WST color scale has been adjusted using a colormap within the range $(-c_{max}, c_{max})$, as the representation includes negative features that are crucial to the glitch detection process. Conversely, the Q-transform has been adjusted to a range of $(0, c_{max})$. Notably, the WST outperforms the Q-transform, with an average improvement of 1%. However, when implementing the combined approach of WST and Q-transform, referred to as WST + Q-transform, there is an enhancement of 2 to 3% compared to the Q-transform alone.

-
- [1] A. Einstein, Die feldgleichungen der gravitation, Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin), 844 (1915).
- [2] B. P. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, *et al.*, Observation of gravitational waves from a binary black hole merger, *Physical review letters* **116**, 061102 (2016).
- [3] B. Abbott *et al.*, Multi-messenger observations of a binary neutron star merger, *The Astrophysical Journal Letters* **848** (2017).
- [4] L. Blackburn, L. Cadonati, S. Caride, S. Caudill, S. Chatterji, N. Christensen, J. Dalrymple, S. Desai, A. Di Credico, G. Ely, *et al.*, The lsc glitch group: monitoring noise transients during the fifth ligo science run, *Classical and Quantum Gravity* **25**, 184004 (2008).
- [5] W. G. Anderson and R. Balasubramanian, Time-frequency detection of gravitational waves, *Physical Review D* **60**, 102001 (1999).
- [6] N. J. Cornish and T. B. Littenberg, Bayeswave: Bayesian inference for gravitational wave bursts and instrument glitches, *Classical and Quantum Gravity* **32**, 135012 (2015).
- [7] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggeles, S. L. Larson, *et al.*, Gravity spy: integrating advanced ligo detector characterization, machine learning, and citizen science, *Classical and quantum gravity* **34**, 064003 (2017).
- [8] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. R. Smith, V. Kalogera, and A. Katsaggeles, Machine learning for gravity spy: Glitch classification and dataset, *Information Sciences* **444**, 172 (2018).
- [9] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. R. Smith, V. Kalogera, and A. Katsaggeles, Machine learning for gravity spy: Glitch classification and dataset, *Information Sciences* **444**, 172 (2018).
- [10] D. M. Macleod, J. S. Areeda, S. B. Coughlin, T. J. Massinger, and A. L. Urban, GWpy: A Python package for gravitational-wave astrophysics, *SoftwareX* **13**, 100657 (2021).
- [11] A. Nitz, I. Harry, D. Brown, C. M. Biwer, J. Willis, T. Dal Canton, C. Capano, L. Pekowsky, T. Dent, A. R. Williamson, *et al.*, gwastro/pycbc: Pycbc release v1.16.11, Zenodo (2020).
- [12] S. Mallat, Group invariant scattering, *Communications on Pure and Applied Mathematics* **65**, 1331 (2012).
- [13] J. Bruna and S. Mallat, Invariant scattering convolution networks, *IEEE transactions on pattern analysis and machine intelligence* **35**, 1872 (2013).
- [14] J. Andén and S. Mallat, Deep scattering spectrum, *IEEE Transactions on Signal Processing* **62**, 4114 (2014).
- [15] Z. Liu, G. Yao, Q. Zhang, J. Zhang, X. Zeng, *et al.*, Wavelet scattering transform for ecg beat classification, *Computational and mathematical methods in medicine* **2020** (2020).
- [16] A. Licciardi and D. Carbone, Whalenet: A novel deep learning architecture for marine mammals vocalizations on watkins marine mammal sound database, *IEEE Access* **12**, 154182 (2024).
- [17] S. Cheng, Y.-S. Ting, B. Ménard, and J. Bruna, A new approach to observational cosmology using the scattering transform, *Monthly Notices of the Royal Astronomical Society* **499**, 5902 (2020).
- [18] G. Valogiannis and C. Dvorkin, Towards an optimal estimation of cosmological parameters with the wavelet scattering transform, *Physical Review D* **105**, 103534 (2022).
- [19] E. Oyallon, E. Belilovsky, and S. Zagoruyko, Scaling the scattering transform: Deep hybrid networks, in *Proceedings of the IEEE international conference on computer vision* (2017) pp. 5618–5627.
- [20] S. Mukherjee, on behalf of the LIGO Science Collaboration, *et al.*, Multidimensional classification of kleinwelle triggers from ligo science run, *Classical and Quantum Gravity* **23**, S661 (2006).
- [21] S. Mukherjee, R. Obaid, and B. Matkarimov, Classification of glitch waveforms in gravitational wave detector characterization, in *Journal of physics: Conference series*, Vol. 243 (IOP Publishing, 2010) p. 012006.
- [22] G. Egri, *Parallel computation on graphical processor units with an eye on gravitational-wave data analysis needs*, Tech. Rep. (Tech. rep., LIGO, 2008).
- [23] A. Hentschel and B. C. Sanders, Machine learning for precise quantum measurement, *Physical review letters* **104**, 063603 (2010).
- [24] J. VanderPlas, A. J. Connolly, Ž. Ivezić, and A. Gray, Introduction to astroml: Machine learning for astrophysics, in *2012 conference on intelligent data understanding* (IEEE, 2012) pp. 47–54.

- [25] M. J. Way, J. D. Scargle, K. M. Ali, and A. N. Srivastava, *Advances in machine learning and data mining for astronomy* (CRC Press, 2012).
- [26] 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.*, Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data, *Physical Review D* **88**, 062003 (2013).
- [27] J. Powell, D. Trifiro, E. Cuoco, I. S. Heng, and M. Cavaglia, Classification methods for noise transients in advanced gravitational-wave detectors, *Classical and Quantum Gravity* **32**, 215012 (2015).
- [28] E. Cuoco, J. Powell, M. Cavaglia, K. Ackley, M. Berger, C. Chatterjee, M. Coughlin, S. Coughlin, P. Easter, R. Essick, *et al.*, Enhancing gravitational-wave science with machine learning, *Machine Learning: Science and Technology* **2**, 011002 (2020).
- [29] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Convolutional neural networks: A magic bullet for gravitational-wave detection?, *Physical Review D* **100**, 063015 (2019).
- [30] M. B. Schäfer, O. Zelenka, A. H. Nitz, H. Wang, S. Wu, Z.-K. Guo, Z. Cao, Z. Ren, P. Nousi, N. Stergioulas, *et al.*, First machine learning gravitational-wave search mock data challenge, *Physical Review D* **107**, 023021 (2023).
- [31] S. Mallat, *A wavelet tour of signal processing* (Elsevier, 1999).
- [32] J. C. Brown, Calculation of a constant q spectral transform, *The Journal of the Acoustical Society of America* **89**, 425 (1991).
- [33] S. Chatterji, L. Blackburn, G. Martin, and E. Katsavounidis, Multiresolution techniques for the detection of gravitational-wave bursts, *Classical and Quantum Gravity* **21**, S1809 (2004).
- [34] J. Yan, M. Avagyan, R. E. Colgan, D. Veske, I. Bartos, J. Wright, Z. Márka, and S. Márka, Generalized approach to matched filtering using neural networks, *Physical Review D* **105**, 043006 (2022).
- [35] S. Klimenko, I. Yakushin, A. Mercer, and G. Mitselmakher, A coherent method for detection of gravitational wave bursts, *Classical and Quantum Gravity* **25**, 114029 (2008).
- [36] S. Klimenko, Wavescan: multiresolution regression of gravitational-wave data, arXiv preprint arXiv:2201.01096 (2022).
- [37] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, *Nature Reviews Physics* **3**, 422 (2021).
- [38] Y. Zhang, P. Tiño, A. Leonardis, and K. Tang, A survey on neural network interpretability, *IEEE Transactions on Emerging Topics in Computational Intelligence* **5**, 726 (2021).
- [39] M. Z. Ahmad, A. M. Kamboh, S. Saleem, and A. A. Khan, Mallat's scattering transform based anomaly sensing for detection of seizures in scalp eeg, *IEEE Access* **5**, 16919 (2017).
- [40] M. B. Arouxet, V. E. Pastor, and V. Vampa, Using the wavelet transform for time series analysis, in *Applications of Wavelet Multiresolution Analysis* (Springer, 2021) pp. 59–74.
- [41] C. Schörkhuber and A. Klapuri, Constant-q transform toolbox for music processing, in *7th sound and music computing conference, Barcelona, Spain* (2010) pp. 3–64.
- [42] C. Schörkhuber, A. Klapuri, and A. Sontacchi, Audio pitch shifting using the constant-q transform, *Journal of the Audio Engineering Society* **61**, 562 (2013).
- [43] C. O. Sakar, G. Serbes, A. Gunduz, H. C. Tunc, H. Nizam, B. E. Sakar, M. Tutuncu, T. Aydin, M. E. Isenkul, and H. Apaydin, A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform, *Applied Soft Computing* **74**, 255 (2019).
- [44] R. A. Roberts and C. T. Mullis, *Digital signal processing* (Addison-Wesley Longman Publishing Co., Inc., 1987).
- [45] J. Bruna, *Scattering Representations for Recognition*, Theses, Ecole Polytechnique X (2013), déposée Novembre 2012.
- [46] S. Mallat, Understanding deep convolutional networks, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**, 20150203 (2016).
- [47] T. Marchand, M. Ozawa, G. Biroli, and S. Mallat, Wavelet conditional renormalization group, arXiv preprint arXiv:2207.04941 (2022).
- [48] M. Andreux, T. Angles, G. Exarchakis, R. Leonarduzzi, G. Rochette, L. Thiry, J. Zarka, S. Mallat, J. Andén, E. Belilovsky, *et al.*, Kymatio: Scattering transforms in python, *Journal of Machine Learning Research* **21**, 1 (2020).
- [49] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. R. Smith, V. Kalogera, and A. Katsagelos, Machine learning for gravity spy: Glitch classification and dataset, *Information Sciences* **444**, 172 (2018).
- [50] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Real-time gravitational wave science with neural posterior estimation, *Physical review letters* **127**, 241103 (2021).
- [51] A. Gunny, D. Rankin, J. Krupa, M. Saleem, T. Nguyen, M. Coughlin, P. Harris, E. Katsavounidis, S. Timm, and B. Holzman, Hardware-accelerated inference for real-time gravitational-wave astronomy, *Nature Astronomy* **6**, 529 (2022).