Detection of anomalies amongst LIGO's glitch populations with autoencoders

Paloma Laguarta[1*], Robin van der Laag[2*], Melissa Lopez[2,3], Tom Dooney[4], Andrew L. Miller[2,3], Stefano Schmidt[2,3], Marco Cavaglia[5], Sarah Caudill[6,7], Kurt Driessens[1], Joël Karel[1], Roy Lenders[8], and Chris Van Den Broeck[2,3]

[1] *Department of Advanced Computing Sciences, Maastricht University, Maastricht, the Netherlands.*
[2] *Institute for Gravitational and Subatomic Physics (GRASP) Utrecht University, Princetonplein 1, 3584 CC, Utrecht, the Netherlands.*
[3] *Nikhef Science Park 105, 1098 XG, Amsterdam, the Netherlands.*
[4] *Centre of Actionable Research Open University, Heerlen, the Netherlands*
[5] *Physics Department Missouri University of Science and Technology 1315 N. Pine St., Rolla, MO 65409, USA*
[6] *Department of Physics, University of Massachusetts, Dartmouth, MA 02747, USA*
[7] *Center for Scientific Computing and Data Science Research, University of Massachusetts, Dartmouth, MA 02747, USA and*
[8] *Boosting Alpha B.V., Villafloraweg 1, 5928 SZ Venlo, the Netherlands*

(Dated: October 6, 2023)

Gravitational-wave (GW) interferometers are able to detect a change in distance of $\sim 1/10{,}000$th the size of a proton. Such sensitivity leads to large appearance rates of non-Gaussian transient noise bursts in the main detector strain, also known as glitches. These glitches come in a wide range of frequency-amplitude-time morphologies and are caused by environmental or instrumental processes, hindering searches for all sources of gravitational waves. Current approaches for their identification use supervised models to learn their morphology in the main strain, but do not consider relevant information provided by auxiliary channels that monitor the state of the interferometers nor provide a flexible framework for novel glitch morphologies. In this work, we present an unsupervised algorithm to find anomalous glitches. We encode a subset of auxiliary channels from LIGO Livingston in the fractal dimension, a measure for the complexity of the data, and learn the underlying distribution of the data using an auto-encoder with periodic convolutions. In this way, we uncover unknown glitch morphologies, and overlaps in time between different glitches and misclassifications. This led to the discovery of anomalies in 6.6% of the input data. The results of this investigation stress the learnable structure of auxiliary channels encoded in fractal dimension and provide a flexible framework to improve the state-of-the-art of glitch identification algorithms.

I. INTRODUCTION

The first detection of a gravitational wave (GW) signal from a binary black hole (BBH) event [1] by the Laser Interferometer Gravitational-Wave Observatory (LIGO) and Virgo collaborations established the field of GW astronomy [2, 3]. Since then, over 90 confident astronomical events have been detected in the past three observation runs by LIGO-Virgo collaboration [4–6] and other research groups [7–12]. In 2017, after an improvement of the detector configuration, the joint observation of Advanced LIGO and Advanced Virgo led to the first detection of a binary neutron star (BNS) inspiral, labelled as GW170817 [13]. The initial announcement of the detection by the Fermi Gamma-ray Burst (GRB) Monitor of GRB170817A [14, 15], and the precise sky location of GW170817 by GW detectors, enabled a rapid electromagnetic follow-up which led to the detection of the associated kilonova, later called AT2017gfo [16].

The detection of GW170817 posed the added challenge of mitigating the effect of a transient non-astrophysical burst of non-Gaussian noise from the data, also known as a glitch, for its subsequent analysis [17, 18]. Glitches may be caused by the environment (e.g., earthquakes, wind, anthropogenic noise) or instruments (e.g., control systems, electronic components [19]), though in many cases their causes remain unknown [20]. They come in a large variety of time-frequency morphologies, have a typical duration of between sub-seconds and seconds, and have a high rate of occurrence (~ 1 per minute during the first half of the third observing run, O3a [5]). They can reduce the amount of analyzable data increasing the noise floor, produce false positives in GW data, affect the estimation of the detector power spectral density and reduce candidate significance in searches for short- and long-lived GW signals [21–25].

Glitches can also bias astrophysical parameter estimation, making it difficult to determine which part of the signal corresponds to a glitch and which part to the actual GW event [26–28]. Additionally, glitches can impact line-cleaning procedures in GW searches, which rely on replacing disturbed frequency bins with artificially generated data, consistent with their neighbours [25, 29, 30]. If the surrounding data contains elevated noise floors, the efficacy of mitigation methods will be reduced.

Glitch identification and characterization is a crucial first step towards their mitigation [31, 32]. Most of the current approaches to glitch characterization with ML utilize supervised classification algorithms, where models learn to identify glitches through labelled time-frequency

[*] These authors contributed equally to this work.

representations of GW strain data $h(t)$ [33–39]. However, this procedure presents several limitations. Supervised learning needs fixed class definitions that are not exhaustive nor representative of all glitch morphologies, as there could be many possible sub-classes to discover [35]. Furthermore, as GW detectors are improved, novel glitch morphologies could arise [40]. Moreover, generating these labels is an expensive task, since ML methods need a lot of examples for training, and experts must vet the labelling procedure.

In this context, unsupervised methods to identify glitches based on ML algorithms could help overcome such limitations. In this paper, we propose a novel ML algorithm that combines auxiliary channel information with an unsupervised anomaly detection algorithm. We encode the information from auxiliary channels from LIGO Livingston in the fractal dimension, a measure of the complexity of the time series. This representation of the data is input to a data-driven algorithm, which consists of a convolutional autoencoder with periodic convolutions that learns the underlying representation of the data, clustering glitches according to their similarity in a compressed representation. By exploiting this compressed representation for anomaly detection, we can identify glitches that strongly deviate from the general distribution of the input data, improving the understanding of glitch populations. We test the method's performance by identifying anomalies on three classes of known glitches in LIGO data.

This paper is structured as follows. In section II we introduce the current state-of-the-art glitch characterization and explain the fractal dimension encoding. In section III we provide details about data acquisition and its pre-processing. In section IV we describe the ML method employed in this investigation. In section V we present the main results of this research, showing different anomalies found with our methodology, and in section VI we conclude , proposing avenues for future research.

II. IDENTIFICATION OF DETECTOR TRANSIENT NOISE

A. Characterization via auxiliary channels

The status of GW detectors is continuously monitored through a large set of data streams at various sampling rates, outputting $\sim 10^6$ time-series from instrumental and environmental sensors. These auxiliary channels can be divided into safe (insensitive to GW) and unsafe (sensitive to GW). Depending on their origins, glitches present varied morphologies in different sets of auxiliary channels. Some subset of these channels may serve as "witnesses" of glitches and are used to create data quality flags before performing GW searches [41–43].

Despite the huge amount of auxiliary channels in a single detector, many of them do not provide useful information for noise transient investigations as they remain constant or vary with a consistent pattern , constituting a data set containing redundant and/or non-informative characteristics [18, 44, 45]. Therefore, LVK researchers have compiled a "reduced" standard list of $\sim 10^3$ auxiliary channels that are used in data quality investigations. In this work, we limit our investigation to safe auxiliary channels with sampling rates $> 512\,\mathrm{Hz}$, yielding a set of 347 channels.

B. Fractal dimension

The first step towards characterizing glitches through safe auxiliary channels requires identifying anomalous data stretches within them [43, 45, 46]. In [47], the author proposes the measurement of fractal dimension (FD) as an additional effective tool for characterizing the instrument output in low latency. FD is an index that characterizes the self-similarity of a set and provides a measure of the complexity of the signal in the context of signal processing [48]. There are several definitions of this magnitude [49–51], implying that the FD measure for a physical process can differ depending on the chosen definition. Nonetheless, in this work, we focus on the *FD variation over time* as an indicator of the evolution of the signal's complexity. As the presence of a glitch in the data affects the noise power spectrum, which in turn varies the value of FD, we are only interested in the relative change which is definition independent.

To illustrate this, Fig. 1 presents the variation of FD for two minutes of data from the `L1:LSC-PRCL_OUT_DQ` auxiliary channel, which measures the Power Recycling Cavity Length (PRCL) from the Length Sensing Contol (LSC) of the LIGO Livingston (L1) interferometer. The computation was performed with a time window $\mathcal{W}(t) = 1\,\mathrm{s}$, i.e. every FD value is the result of encoding 1 s of the input data. Points greater than one standard deviation σ from the mean FD correlate to the presence of `Whistle` glitches in the detector. As we can observe from Fig. 1, FD can be an effective tool to further understand the coupling between glitches and auxiliary channels. To extend this analysis to a larger set of safe auxiliary channels and glitch classes, we first need to speed up the FD calculation to near-real time.

Following [47] we numerically estimate the measured FD with the variation (VAR) method (see [47] for details). For a discretely-sampled set of data with N measurements $\mathcal{C} \in \mathbb{R}^N$, we can define a sliding window to compute the variation of the data with centre l and scale k,

$$\mathcal{F}_{k,l} = \left|\max\left[\mathcal{C}_{l-k},\ldots,\mathcal{C}_{l+k}\right] - \min\left[\mathcal{C}_{l-k},\ldots,\mathcal{C}_{l+k}\right]\right|. \quad (1)$$

Thus, the VAR estimator for a given scale k is,

$$\mathcal{A}^{\mathbf{VAR}}(k) = \frac{1}{N-2k}\sum_{l=k}^{N-1-k}\mathcal{F}_{k,l}, \quad (2)$$

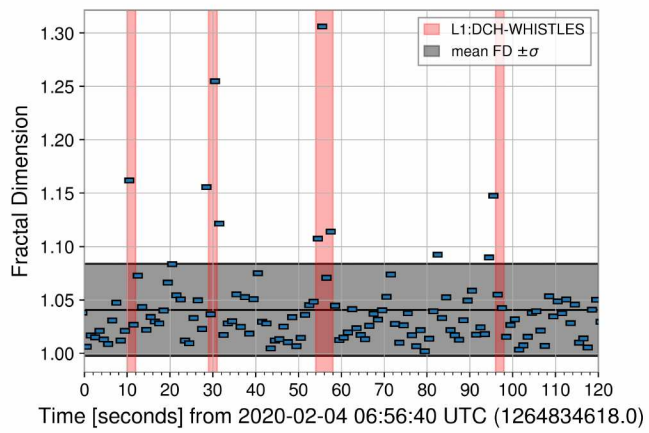


FIG. 1: Fractal dimension over a two minute period of L1 data for the `L1:LSC-PRCL_OUT_DQ` auxiliary channel. Each point represents the fractal dimension for one second of data and the red regions indicate the time period containing `Whistle` glitches.

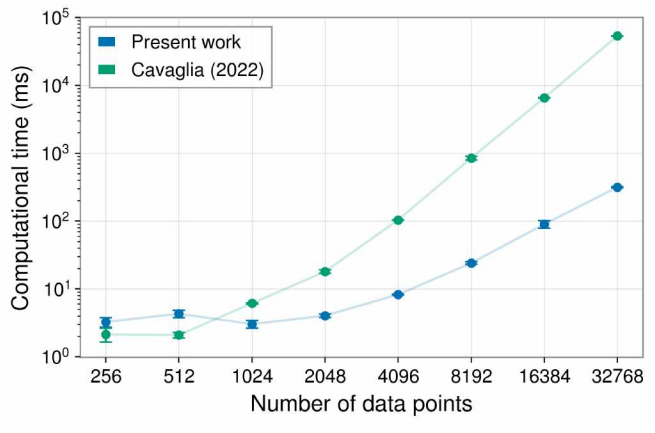


FIG. 2: Comparison of the fractal dimension computing algorithms for varying number of data points. Benchmarks done on a Intel® Xeon® Processor E5-2630 v4 CPU @ 2.20GHz.

As we can see in Algorithm 1, the implementation in [47] computes the maximum and minimum over a range of values at each iteration k, l (line 5 and 6). The runtime of this implementation is $\mathcal{O}(N^3)$.

A significant speed-up can be achieved using Algorithm 2 based on [52]. It uses the fact that at iteration $k+1$ we can compute the maximum as

$$\begin{aligned}\max[\mathcal{C}_{l-(k+1)},\ldots,\mathcal{C}_{l+(k+1)}] = \\ \max\{\max\left[\mathcal{C}_{(l-1)-k},\ldots,\mathcal{C}_{(l-1)+k}\right], \\ \max\left[\mathcal{C}_{(l+1)-k},\ldots,\mathcal{C}_{(l+1)+k}\right]\},\end{aligned} \tag{3}$$

where the components of the right-hand side have already been computed at iteration k. This step is done on line 10 in Algorithm 2, and likewise in line 11 for the minimum. Now, the computational complexity of the FD calculation is $\mathcal{O}(N^2\log(N))$ and the practical speed-up can be seen in Fig. 2, where we compute FD with both methods over data increasing in length. While this speed-up is not apparent for short stretches of data at low sampling rates, it becomes significant at sampling rates ≥ 4096 Hz.

Algorithm 1 Implementation of the VAR method from [47].

Input: f vector of size N.
Output: A vector of size $N/2$.

1: $A[1\ldots N/2] = 0$
2: **for** $k = 1$ **to** $N/2$ **do**
3: $\quad F = \emptyset$
4: $\quad$ **for** $l = k$ **to** $N-k$ **do**
5: $\quad\quad F = F \cup \{\max\{f[l-k],\ldots,f[l+k]\}$
6: $\quad\quad\quad\quad -\min\{f[l-k],\ldots,f[l+k]\}\}$
7: $\quad A[k] = \text{mean}(F)$
8: **return** A

Algorithm 2 Improved algorithm for the VAR method.

Input: f vector of size N.
Output: A vector of size $N/2$.

1: $A[1\ldots N/2] = 0$
2: $u[1\ldots N-2] = 0$
3: $b[1\ldots N-2] = 0$
4: **for** $i = 1$ **to** $N-2$ **do**
5: $\quad u[i] = \max\{f[i],\ldots,f[i+2]\}$
6: $\quad b[i] = \min\{f[i],\ldots,f[i+2]\}$
7: $A[1] = \text{mean}(u[1\ldots N-2] - b[1\ldots N-2])$
8: **for** $i = 2$ **to** $N/2$ **do**
9: $\quad$ **for** $j = 1$ **to** $N-2i$ **do**
10: $\quad\quad u[j] = \max\{u[j], u[j+2]\}$
11: $\quad\quad b[j] = \min\{b[j], b[j+2]\}$
12: $\quad A[i] = \text{mean}(u[1\ldots N-2i] - b[1\ldots N-2i])$
13: **return** A

In practice, with Algorithm 1 with computational complexity $\mathcal{O}(N^3)$, we were able to FD-encode 1 h of data in 1 h, but now with an efficient implementation with `numba` [53] of Algorithm 2 based on [52], with computational complexity $\mathcal{O}(N^2\log(N))$, we can now process 1 h in 11 s. With further parallelization in a cluster, the FD computation could characterize glitches in low latency. Now that we have a fast computation of FD-value, we can construct a data set for our application.

III. DATA SET AND PRE-PROCESSING

The aim of this work is to understand the underlying glitch population using solely information from safe auxiliary channels. Due to the overwhelming amount of information, we encode the data of the safe auxiliary channels in FD. Afterwards, we use an unsupervised ML method to learn the underlying distribution of the data, finding anomalies that strongly deviate from the general

trend of the FD-encoded data. While unsupervised ML algorithms in the context of anomaly detection are agnostic, as they do not make prior assumptions regarding the data distribution, it is challenging to interpret their results. To understand the results of our algorithm and assess its performance we can compare the output of our algorithm with the findings of supervised glitch classifiers, employing them as a benchmark. In the following subsections, we describe the benchmark used in this work, the selection of glitch populations and the FD-encoding of the data.

A. Glitch classification

In the present work, we employ *Gravity Spy* as a benchmark, finding anomalies from its high-confidence classifications. *Gravity Spy* is an algorithm that combines supervised ML and citizen science to characterize glitches present in LIGO data according to their morphologies in GW strain data $h(t)$, represented in time-frequency [33] . The trained algorithm assigns glitches a pre-defined class and gives a confidence score that it belongs to this class.

In practice, alerts are generated by Omicron, which is an algorithm designed to search for power excess in time series data using the Q-transform, a modification of the standard short-time Fourier transform parameterized by a quality factor Q [46, 54]. Gravity Spy assigns a class and a confidence value to Omicron's alert if it exceeds 7.5 signal-to-noise ratio (SNR), a magnitud related to the tranform coefficient of the Q-transform. Currently, Gravity Spy considers 22 glitch classes, which have been previously identified [18, 55, 56].

B. Glitches

In this proof-of-concept work, we select GPS times t that contain in $h(t)$ no apparent excess of power, and t of three distinct glitch morphologies in LIGO Livingston with Gravity Spy confidence $> 90\%$ [36]. One must note that for the glitches t represents the peak time of the Omicron alert. The three morphologies are chosen to have short and long-duration glitches that are abundant in LIGO Livingston data (> 800 samples per class), and that impact GW searches due to their wide frequency contribution. We detail each class below:

- `No_Glitch`: in this class, no significant excess power is visible in the Gravity Spy spectrograms (see Fig. 3a). In the context of this work, this class represents a stable behaviour of the GW detector, which is reflected by non-deviant FD values.

- `Whistle`: these glitches have a characteristic V, U or W shape at higher frequencies ($\gtrsim$ 128 Hz) with typical durations ~ 0.25 s. They are caused when radio-frequency signals beat with the voltage controlled oscillators [57]. In Fig. 3b we present a `Whistle` glitch with a frequency content > 512 Hz.

- `Tomte`: these glitches are also short-duration (~ 0.25 s) with a characteristic triangular morphology. In Fig. 3c we show a `Tomte` glitch from LIGO Livingston, where these morphologies are quite abundant. Since there is no clear correlation to the auxiliary channels, they cannot be removed from astrophysical searches.

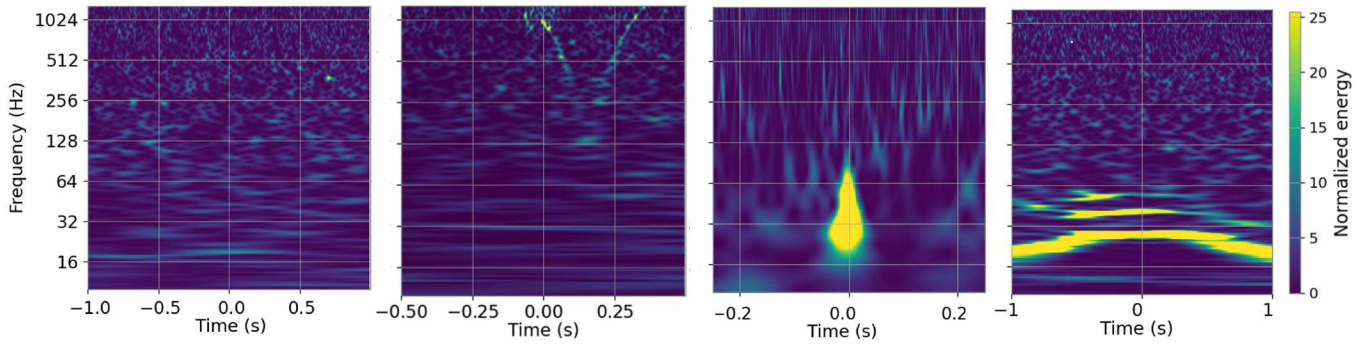
- `Scattered_Light`: also known as Slow Scattering, these glitches have longer duration harmonics ($\sim$ 2.0–4.0 s) that in time-frequency domain they appear as arches being often stacked on top of each other (see Fig. 3d). These glitches are quite problematic since their frequency content lies in the band of interest of GW astrophysical events. In O3, they were found to be coupled with the relative motion between the optical suspension system's end test-mass chain and the reaction-mass chain [58].

In this work we focus on LIGO Livingston, as the author in [47], but this investigation could be extended to LIGO Hanford and Virgo. The details on how this data was pre-processed for its posterior usage in our model, can be found in the next section.

C. Auxiliary channels encoded in fractal dimension

Given a time t of interest, we select an array of GPS time with duration $\Delta t = 8$s, where t is in the center. For each array of time, we retrieve 347 safe auxiliary channels with sampling rates > 512 Hz, excluding the GW strain $h(t)$, that is then whitened and encoded in FD with time windows $\mathcal{W}(t) \in \{0.25, 0.5, 1, 2\}$ s. For each $\mathcal{W}(t)$ we have $\Delta t/\mathcal{W}(t) - 1$ time bins to ensure that t is in the center of the FD-encoded data, yielding a total of 56 time-bins for sample. Since the duration of `Scattered_Light` is $\sim$ 2-$-$ 4 s and the duration of `Tomte` is ~ 0.25 s, the length of these varying time windows ensure that any glitch morphology will be contained at least within $\mathcal{W}(t) = 2$s [33]. Note that the sampling rate of each independent auxiliary channel varies, but we only encode safe channels with a sampling rate > 512 Hz, to have enough data points to perform a calculation of the FD, as demonstrated by the experiments performed by [47].

Limited by the number of `Whistle` present in LIGO Livingston, for the initial data set we select 896 GPS times for each class defined in Section III B and presented in Fig. 3, yielding a balanced data set. Since each auxiliary channel monitors distinct physical processes, their average FD measurements can differ, giving priority to certain channels over others. To improve the stability of our model, we normalize in the range $[0, 1]$ the data of each individual auxiliary channel, as we are only interested in their relative variation. Normalizing collectively



(a) `No_Glitch` class (b) `Whistle` class (c) `Tomte` class (d) `Scattered_Light` class

FIG. 3: From left to right: Q-transform of `No_Glitch`, `Whistle`, `Tomte` and `Scattered_Light` retrieved from Gravity Spy [33].

would give more importance to the channels with a higher FD and dismiss the channels with a lower FD.

For this work we reduce the dimensionality of the normalized data set with dimensions 347 channels × 56 time bins using a data-driven approach. Our aim is to maintain the channels that capture the most relevant features of the glitches with respect to the `No_Glitch` class, so we follow the procedure below:

1. Defining D_{NG} as the set of `No_Glitch` FD-encoded, we compute the average of all elements in D_{NG}, $\mu(D_{\mathrm{NG}})$, to minimize extreme deviations of FD. This will be the common background when `No_Glitch` is present.

2. For a single glitch d_{C} encoded in from a certain class C, we subtract the background as $d_{\mathrm{C}} - \mu(D_{\mathrm{NG}})$. This subtraction highlights the deviations produced by the presence of a glitch in the data.

3. We identify auxiliary channels A that present a low FD deviation with respect to the background, since their contribution is similar to the absence of glitch. Thus, given a glitch d_{C} and a channel A, if $d_{\mathrm{C,A}} - \mu(D_{\mathrm{NG}})_{\mathrm{A}} \lesssim 10^{-2} \quad \forall$ glitches d_{C}, the auxiliary channel A is removed. This threshold represents a balance between data compactness and expressiveness. Too many channels can introduce irrelevant information, while too few may overlook the overall data trends.

This pre-processing reduced the dimensionality to a shape of 50 channels × 56 time bins . In Fig. 4 we show an example of 8s of FD-encoded data. To train the ML model presented in the next section we will use the three glitch morphologies, with a total of 2688 samples, which contain the structure that we wish to unravel. One must note that while supervised approaches must use a subset of the data to assess the generalization ability of the model, in the present unsupervised approach we are interested in learning the details of the data at hand, such that all data instances are employed.

IV. METHODOLOGY

In recent times, ML algorithms have sparked the interest of scientists due to their success in solving various tasks in different domains and their transversal applications. In particular, they have emerged as a novel tool in the field of GW for different tasks: pattern recognition, such as identification of BBH [59, 60], BNS [61–63], transient burst GW [64–72], and continuous wave signals [73]; GW signal and glitch generation [74–79]; as well as anomaly detection [80], among others (see [81] for a comprehensive review).

The novelty of this work lies in combining auxiliary channel information with ML in the context of anomaly detection. The complexity of the dataset presented in Section II implies two main challenges:

- *Lack of ground-truth:* While glitch morphologies have been widely studied, there is no guarantee that all glitches belonging to a certain class present the same standard behaviour. In this work, the labels assigned by Gravity Spy are not considered ground truth and are used only for analysis and comparison purposes.

- *Lack of absolute ordering:* The ordering in the channels is arbitrary, as they measure different physical magnitudes. Consequently, it is not expected to find local patterns in the vertical axis with any particular channel ordering. Thus, our model needs to learn patterns beyond local correlations in an order-independent way.

In the following subsections we provide the details of our implementation and how these issues were addressed.

A. Convolutional Autoencoder

To address the concern of lack of ground truth, we employ an autoencoder in the context of anomaly detec-

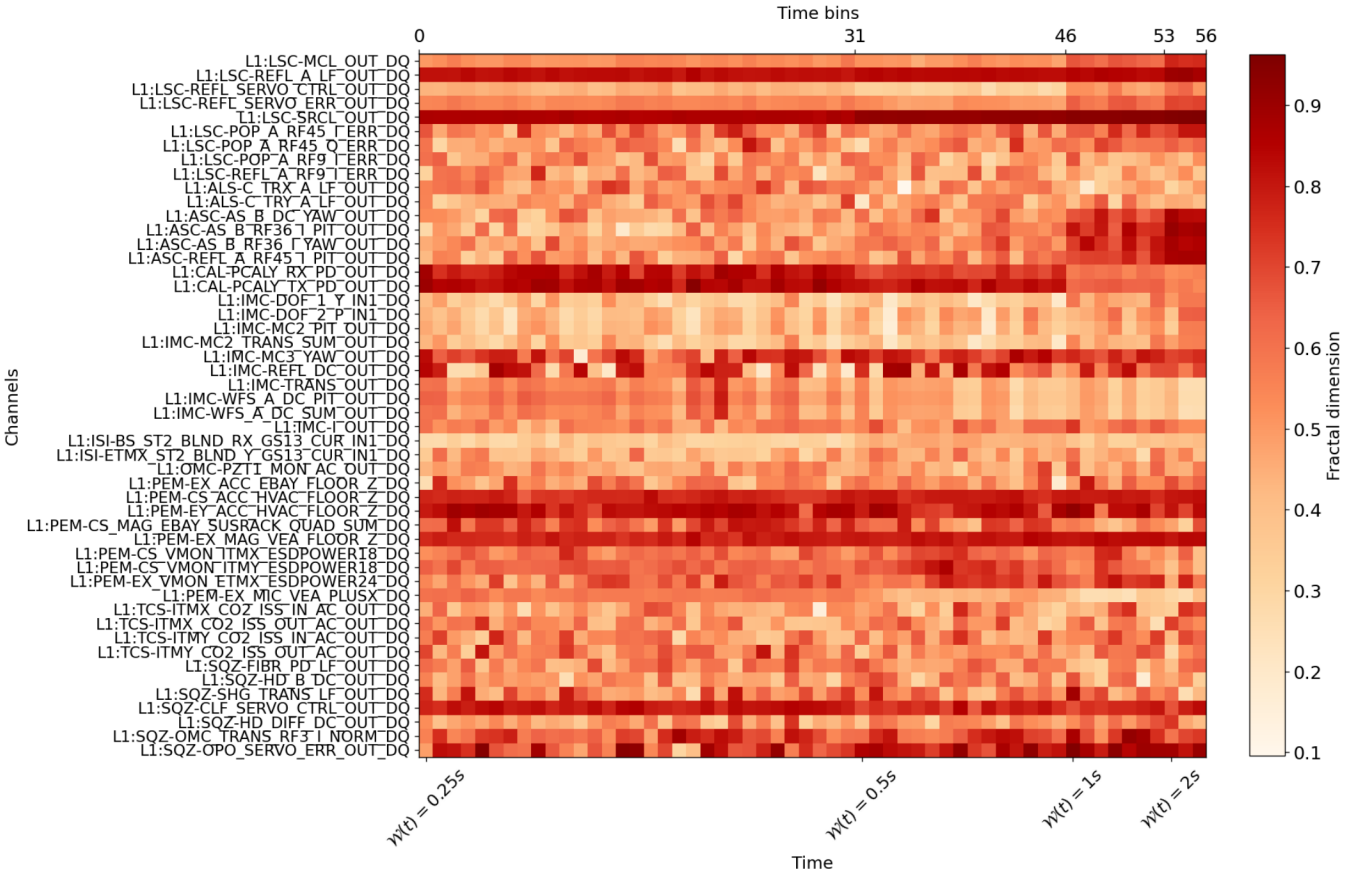


FIG. 4: A sample encoded in fractal dimension with 347 safe auxiliary channels and 56 time bins. Notably, the same event is replicated four times with a different time window: time bins $\in [0, 31)$ have a time window $\mathcal{W}(t) = 0.25\,\mathrm{s}$, time bins $\in [31, 46)$ have $\mathcal{W}(t) = 0.5\,\mathrm{s}$, time bins $\in [46, 53)$ have $\mathcal{W}(t) = 1\,\mathrm{s}$, time bins $\in [53, 56]$ have $\mathcal{W}(t) = 2\,\mathrm{s}$.

tion. Autoencoders are a type of deep-learning algorithm known for their ability to uncover essential structures and patterns within unlabeled datasets, as well as their effectiveness in anomaly detection [82, 83]. They achieve this by compressing the data into a lower-dimensional or sparse format, known as an embedded space, maintaining the most relevant information from the dataset (encoding), and subsequently reconstructing it (decoding). The encoding is expected to employ important sub-structures that can be difficult to notice in the original representation space due to its higher dimensionality and feature redundancy [83]. As a consequence, the learned embedding serves as a reference for detecting irregularities in the glitch data, since data points that deviate significantly from their embedded representations are likely anomalous. Still, the embedded space can be hard to interpret itself, since it forms a high-dimensional space in which the data is densely packed.

When dealing with data with a natural order between the features, convolutional filters can be used to detect (hierarchically combined) local structures and patterns that allow a complex encoding model without an exponential increase of its parameters [84–86]. While our data is two-dimensional (time bins × auxiliary channels), given that we want to preserve the detail in the time dimension, we allow our model only to convolve along the channel dimension using 1-dimensional convolutions.

B. Periodic Convolutions

To address the concern of lack of absolute ordering and the possible lack of local patterns to exploit with limited range convolutional filters, we employ a periodic convolution with filters sized to cover all channels instead. We take inspiration from circular convolutions, which are used in the field of signal processing and consider the input signals as circular, or periodic, rather than finite, i.e. the end of the signal wraps around to the beginning, creating a cyclic or periodic nature [87]. In the context of this work, we use convolutional filters with a size equal to the number of channels, so that the filter has the opportunity to ignore the data's arbitrarily chosen channel ordering. The model applies these filters periodicly, hence the name, so that learned filters can still be used by the model to encode structures and patterns

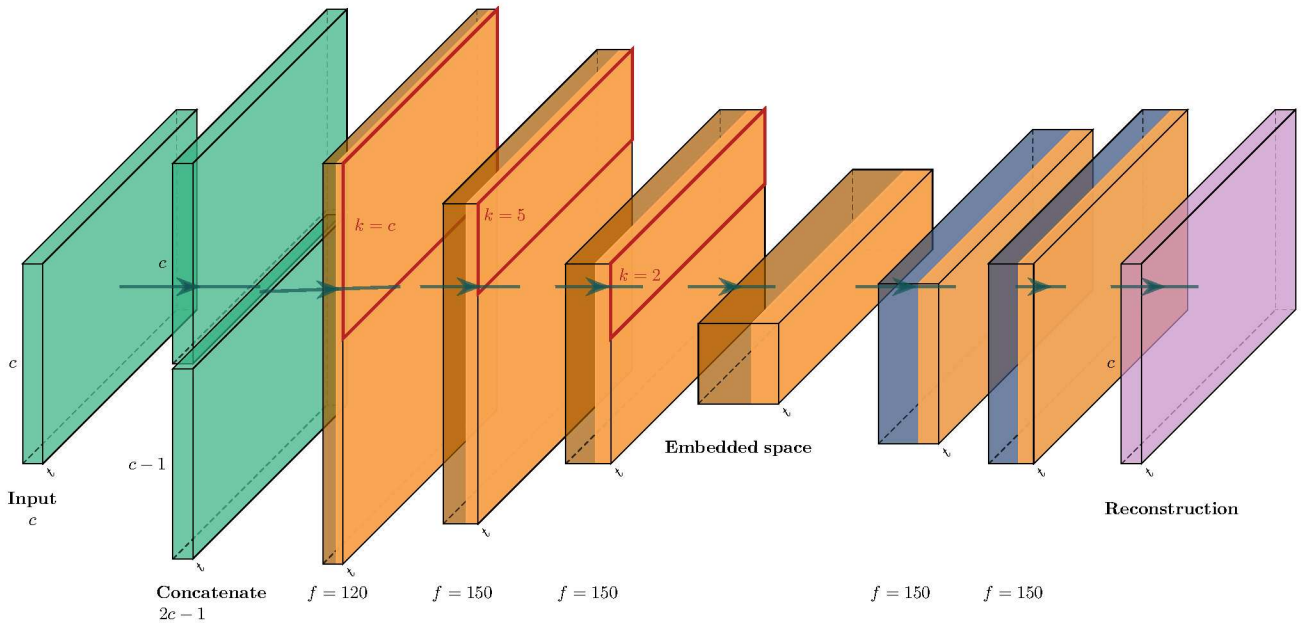


FIG. 5: Outline of the autoencoder with periodic convolutions implementation. The input data which is concatenated yields a shape $(t, 2c-1)$ is presented in green. Downsampling convolutions are coloured in dark orange while upsampling convolutions are coloured in dark blue. Each convolution is followed by a ReLU activation function coloured in orange. This architecture yields a reconstruction of shape (t, c).

found on different channel-combinations.

The outline of the autoencoder's structure is presented in Fig. 5. Periodic convolutions were implemented with `tensorflow` [88] and `keras` [89] using a custom approach: given input with c auxiliary channels and t time bins, the input gets duplicated and concatenated along the channel axis, removing the last channel, as it is represented in Fig 6. In this way, each cyclic permutation of channels is seen only once. Thus, the dimensionality of the input fed into the model is $(c, t) + (c-1, t) = (2c-1, t)$, (see second block of Fig. 5 where the concatenation happens). To maintain the time dimension, we convolve along the channel dimension with a kernel size $k = c \times 1$, stride $s = 1$, filters $f = 120$ and no padding. This custom approach ensures that there is no specific spacial ordering in the vertical (channel) dimension and that the model can capture correlations between all the channels.

Once the first convolutional layer has captured the data patterns between arbitrary channels, the rest of the encoder architecture has the goal of constraining the data into an embedded space. Thus, it consists of two conventional downsampling convolutional layers with kernel sizes $k = 5 \times 1$ and $k = 2 \times 1$ with $f = 150$, respectively, represented in Fig. 5 in dark orange. The decoder structure is a mirror of the encoder, but with up-sampling convolutional layers instead, coloured in blue in Fig. 5. The resulting embedded space has a dimensionality of 5×56 with $f = 150$, which we will use to detect anomalies.

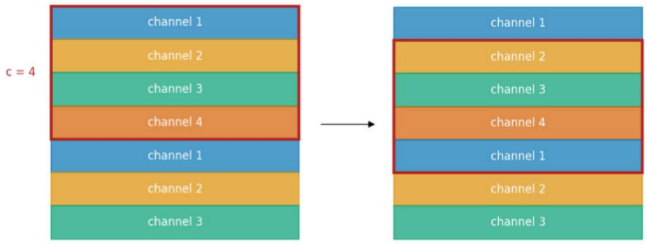


FIG. 6: Example of periodic convolution where input of shape $(2c-1, t)$ is convolved with a kernel $k = c = 4$. All cyclic channel permutations are convolved, so there is no absolute ordering.

After each convolutional layer a ReLU activation function is employed to introduce non-linearity in the model (see Fig. 5 in orange). The ReLU activation function avoids vanishing gradients [90]. The model was trained for 500 epochs and a batch size of 168, using the Adam optimizer [91], with a learning rate $l_r = 10^{-3}$. The loss function employed is the Mean Squared Error (MSE) loss, which represents the cumulative squared error between the input and its reconstruction [92]. To assess the performance of the model, we use the reconstruction error ϵ_R which is defined as follows:

$$\epsilon_R = \sum_{k=0}^{c-1}\sum_{l=0}^{t-1}(D_{i,kl} - D_{r,kl})^2, \quad (4)$$

where D_i is the input to the model and D_r its reconstruction [93]. We expect that lower reconstruction errors translate to more accurate anomaly identification.

C. t-Distributed Stochastic Neighbour Embedding

As discussed in the previous section, the embedded space is still high dimensional and difficult to interpret. Hence, to further lower the data dimensionality and make the data distribution in the embedded space easier to interpret, we use the t-distributed stochastic neighbour embedding (t-SNE) method[2], which projects high-dimensional data in a low-dimensional space, preserving local relationships between data points and underlying structure of the data, but releasing global relationships between data points [95]. While the t-SNE variables do not have an interpretable physical meaning, they are a linear correlation of physical variables and they can be traced back to assess their contribution.

After obtaining the 2-dimensional projection of the embedded space with t-SNE, it is straightforward to visualize the distribution of the different data points, which correspond to the glitch instances, as we show in the next section (see Fig. 8). The 2-dimensional plot is expected to reveal different clusters and interesting structures in the data since data points that are distant from the main clusters of their predicted class are anomalies. By labelling each glitch with its corresponding Gravity Spy label with confidence $> 90\%$, outliers and new glitch morphologies are expected to be identified.

V. RESULTS

To assess the performance of the autoencoder presented in the previous section, we show the achieved reconstruction errors $\epsilon_R \in [0.001, 0.014]$, as seen in Fig. 7. The three glitch classes present similar distributions, with the `Scattered_Light` class reaching the highest reconstruction error $\epsilon_R = 0.014$. The reconstructed input differs from its original on 1.75% and 17.5% in the best and worst reconstructed pixels, respectively. Note that the model reconstructs 98.8 % of glitches with $\epsilon_R < 0.002$, which is a low error given the range $[0.0, 1.0]$ of the input data. In general, it appears that pixels from the same auxiliary channel have similar reconstruction errors, which translates to the preservation of FD-encoded data structure within the given auxiliary channel.

[2] `TSNE` function from `scikit-learn` library [94] was employed.

In Fig. 8 we present joint and marginal distributions of the t-SNE projections with three different representations of the data: the original dataset with 347 auxiliary channels, the reduced dataset with 50 auxiliary channels (see Section III C), and the embedded space with shape 5×56 with $f = 150$. These t-SNE representations cluster the input data in different regions of the space, such that the samples present in the out-skirts will be considered anomalies. We use the labels of Gravity Spy to track in which regions of t-SNE space the different classes fall.

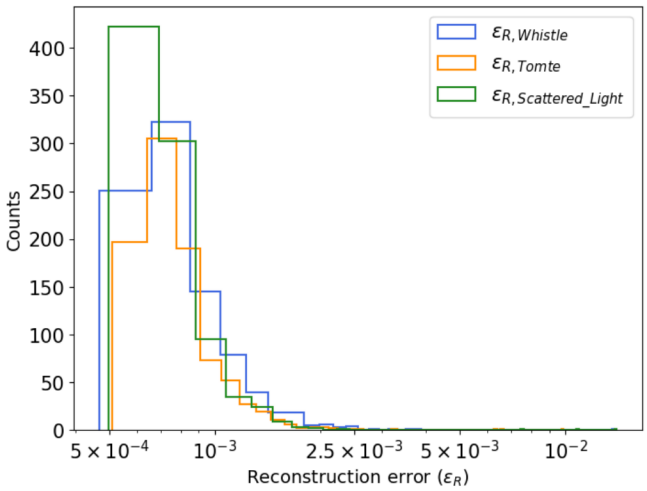
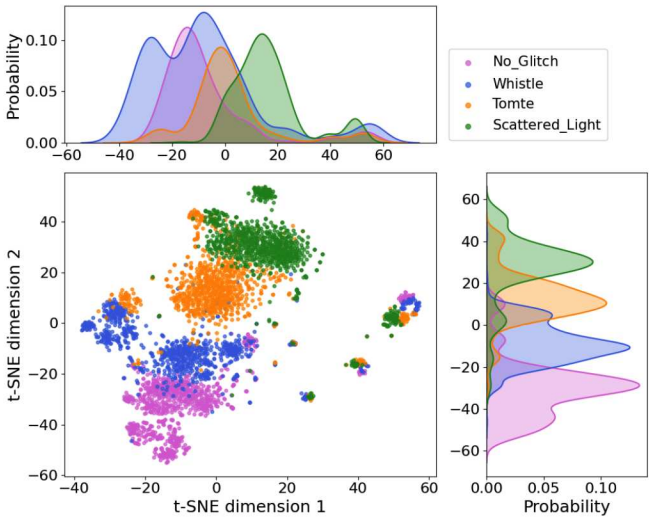
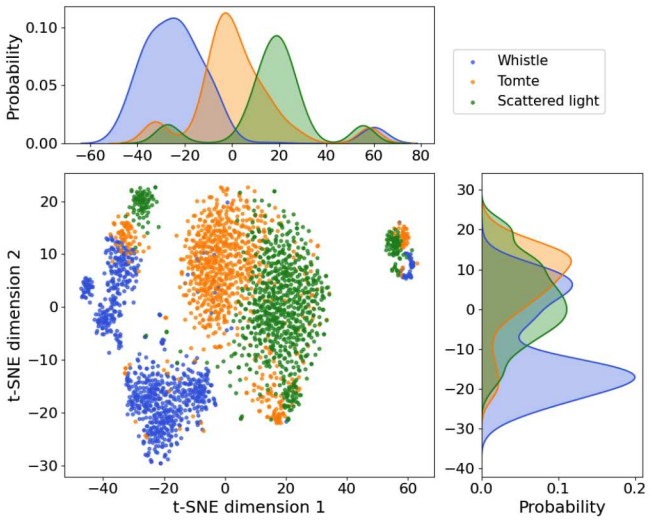


FIG. 7: Histogram with the reconstruction errors ϵ_R for each glitch class in logarithmic scale.

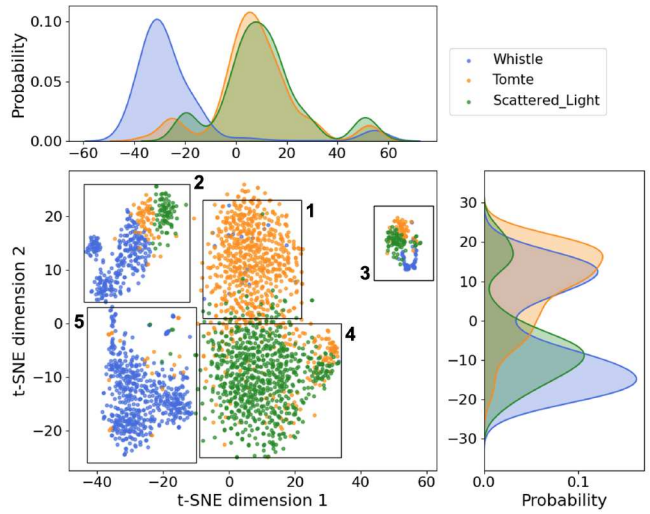
In Fig 8a, the t-SNE projection of the original FD-encoded data with 347 safe auxiliary channels clusters the glitch by similarity, which is consistent with Gravity Spy's classification revealing some overlap between classes, especially between `Tomte` and Scattered_Light, as well as between `Whistle` and `No_Glitch`. A dataset with instances of dimensions 347×56 introduces an enormous complexity. Therefore, we reduce the dimensionality of the data, using the safe auxiliary channels that show the most variance in the FD-encoding, which is related to the presence of glitches (see Section III C). Such reduction in dimensionality yields a more compressed representation of the input data and less overlap between the glitch classes, as can be seen in Fig. 8b since there are fewer sub-clusters of each class. After training the autoencoder, which yields small reconstruction errors ϵ_R, we can project its embedded space in 2 dimensions with t-SNE, as we can see in Fig. 8c. We can observe that it looks similar to the reduced t-SNE in Fig. 8b, but its marginal distributions seem similar to Gaussian distributions since the model is learning the general trend of local and global correlations of the FD-encoded data. While at each compression we are discarding some characteristics of the data, we are maintaining the general trend of the data which, as stated before, is consistent with what is observed by Gravity Spy in $h(t)$, the main strain of the data.



(a) *Original t-SNE:* projection of FD-encoded data with 347 auxiliary channels.



(b) *Reduced t-SNE:* projection of FD-encoded data with 50 auxiliary channels



(c) *Embedded t-SNE*: projection of the embedded space of the autoencoder showing five regions of interest.

FIG. 8: Joint and marginal distributions of t-SNE projections of the data.

To explore the embedded space, in Fig. 8c, we have manually outlined the distinct clusters. In the following, we present the time-frequency of the strain $h(t)$ of some anomalous examples found in the outskirts of these clusters solely employing safe auxiliary channels. The reader will encounter glitch classes that have not been mentioned before in the present work, but their description can be found in [35].

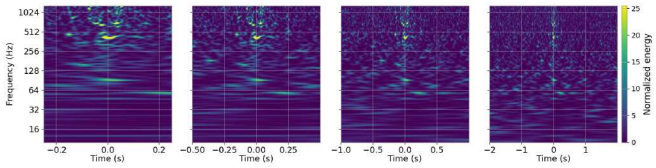
- **Region 1** corresponds to the main `Tomte` cluster. However, some `Whistle` and `Scattered_Light` labels are also present. From this region, in Fig. 9a we show a glitch classified as `Whistle` but with an anomalous morphology, and in Fig. 9c we show a misclassified Wandering Line glitch classified as a Whistle. In Fig. 9b we present a glitch classified as `Scattered_Light` but its morphology is similar to Scratchy, and in Fig. 9d we can see a `Scattered_Light` with an anomalous morphology.

- **Region 2** is a sub-cluster of Whistle, with `Tomte` and `Scattered_Light` overlaps. In Fig. 10a, we can see a `Tomte` glitch that is revealed to be overlapping with a Scratchy glitch in a longer time window. In Fig. 10c we see a `Tomte` glitch overlapping with a smaller `Tomte` glitch. Fig. 10b shows a Fast Scattering glitch mislabelled as `Scattered_Light` and Fig. 10d presents a `Scattered_Light` overlapping with an unknown morphology.

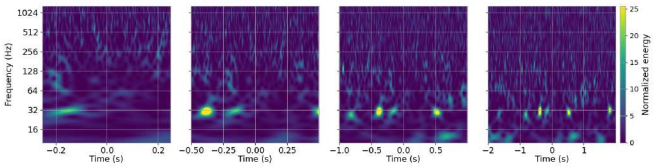
- **Region 3** is a cluster with anomalous glitches from the 3 different glitch classes. Examples of anomalies are presented in Figs. 11a and 11d, which are labelled as `Whistle` but have distinct morphologies that differ from any of the 22 Gravity Spy classes. Another anomalous glitch from this region is the misclassified glitch shown in Fig. 11b, which was labelled as `Tomte` but has a morphology consistent with Fast Scattering. Another example of an anomalous glitch from this region presents as an overlap as shown in Fig. 11c. The glitch was labelled as `Scattered_Light` but seems to be a `Tomte` overlapping with an unknown morphology.

- **Region 4** corresponds to the main `Scattered_Light` cluster. In this region, there is a high presence of `Tomte` labels, which could indicate that both physical processes are related. In Fig. 12a, we present a glitch from this region that was labelled as `Tomte` but is consistent with the `Koi_Fish` class, while 12b was also labelled as a `Tomte` but seems to be an overlap between `Tomte` and Scratchy.

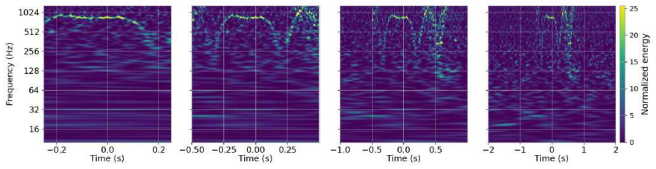
- **Region 5** corresponds to the main `Whistle` cluster, where we find some `Scattered_Light` labels as well as a few `Tomte` labels. In Figs. 13a and 13c we



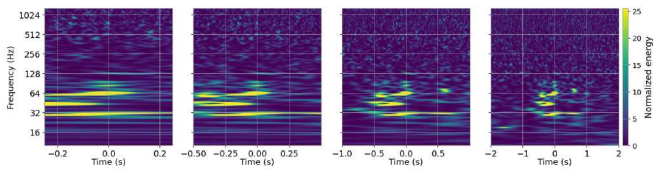
(a) Glitch labeled as `Whistle`, but appears to be an anomalous morphology.



(b) Glitch labeled as `Scattered_Light`, but its shape is consistent with the `Scratchy` class, being a misclassification.

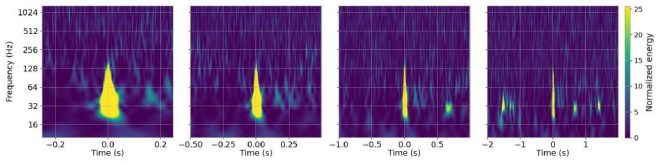


(c) Glitch labeled as `Whistle`, but its shape is more consistent with `Wandering_Line` class, constituting a misclassification.

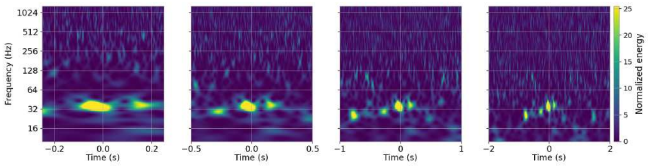


(d) Glitch labeled as `Scattered_Light`, but appears to be an anomalous morphology.

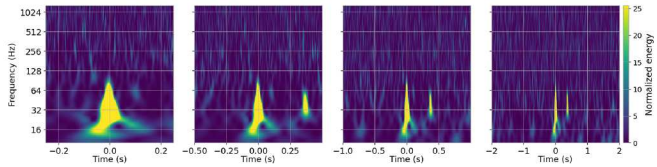
FIG. 9: **Region 1:** Example of anomalous glitches in the embedded space.



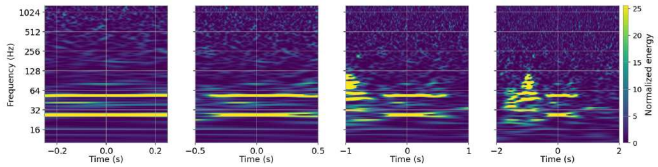
(a) Glitch labeled as `Tomte`. On the spectrogram with the longer time window we can see an overlap between a `Tomte` and `Scratchy` glitch.



(b) Glitch labeled as `Scattered_Light`. It is consistent with the `Fast_Scattering` morphology, being a misclassification.



(c) Glitch labelled as `Tomte`, overlapping with a smaller `Tomte`.



(d) Glitch labeled as `Scattered_Light`. On the left plot, an unknown morphology is overlapping at the beginning of the `Scattered_Light`.

FIG. 10: **Region 2:** Example of anomalous glitches in the embedded space.

see Tomte glitches that appear to be overlapping with Scratchy glitches, in Fig. 13b we see a glitch labelled as Scattered_Light but could be a novel morphology, and in Fig. 13d we see a Scratchy glitch misclassified as Scattered_Light.

The outliers found at the outskirts of their clusters are visually and manually selected from the t-SNE representation in Fig. 8c, with the aim of automating the procedure in future works. After outliers have been selected, their spectrograms in $h(t)$ are visually inspected by comparing them to standard Gravity Spy morphologies. With this procedure, a total of 177 anomalies were found out of 2688 samples, which implies 6.6% of the data. In particular, for each class, we found:

- Whistle: 49 anomalies were found, 45% are unknown morphologies, 28% are Gravity Spy misclassifications, and 27% are glitch overlaps.
- Tomte: 57 anomalies were found, 32% are unknown morphologies, 21% are misclassifications, and 47% are glitch overlaps.
- Scattered_Light: 71 anomalies were found, 28% are unknown morphologies, 72% are misclassifications, and only one case of overlap is found.

After a visual inspection, we found that for Whistle most outliers constitute unknown morphologies, while for Tomte most anomalies are due to overlaps, where it is common that two Tomte happen simultaneously. For Scattered_Light, most outliers correspond to misclassifications, since the other seven glitch classes happen at similar frequency intervals and duration periods, namely Low_Frequency_Burst, Low_Frequency_Lines, Power_Line, Scratchy, Air_Compressor, Paired_Doves and Fast_Scattering.

While the misclassification of glitches could be countered with the improvement of training strategies, data set construction or class definitions, the identification of anomalies arising from overlaps and novel morphologies would still be hampered by the strict class definitions from supervised methods. Therefore, unsupervised approaches, such as the one presented in this work, will improve the understanding of glitch populations for their subsequent mitigation.

VI. CONCLUSIONS

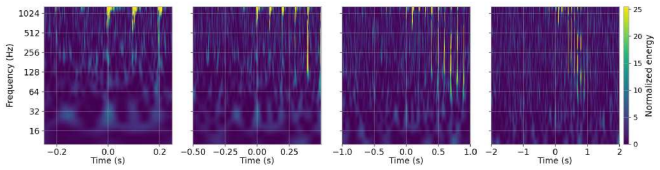
In this paper we have performed an exploratory analysis of a reduced set of safe auxiliary channels from LIGO Livingston with FD-encoding in the context of anomaly detection. The focus of this work is, on one hand, to explore the potential of this data representation in the context of glitch characterization, and on the other hand, to build a data-driven model to cluster glitches in an unsupervised way with direct information from the detector, finding anomalies that deviate from the general distribution of the data.

For this aim, we first speeded up the FD calculation from a computational complexity of $\mathcal{O}(N^3)$ in [47] to $\mathcal{O}(N^2 \log N)$, constructing the FD-encoded safe auxiliary channel data set. Afterwards, we implemented a periodic convolutional autoencoder to learn the local and global structure of the data, compressed in a lower-dimensional space, known as embedded space. The reconstruction errors of the output of the autoencoder were $\sim 98.8\%$ of glitches < 0.002, implying that the autoencoder was able to learn the general trend of the data.

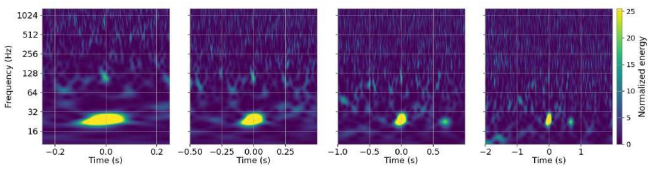
We can also observe the reliable compression of the autoencoder, using solely safe auxiliary channels, when we project the embedded space in a two-dimensional t-SNE. This t-SNE representation clusters the different classes in separate regions which are consistent with Gravity Spy's observation in the main detector strain, $h(t)$. Samples that deviate significantly from their closest cluster are considered outliers. Representing these outliers in $h(t)$, we observed novel morphologies that strongly deviated from the standard definitions of Gravity Spy.

This methodology has shown that the safe auxiliary channel in the FD-encoding acts as a complementary representation to the visualization of $h(t)$, used to characterize the noise of the detector and to identify glitches for their subsequent mitigation. Furthermore, our algorithm is flexible and completely data-driven, capable of uncovering misclassifications, glitch overlaps and novel glitch morphologies. While our method is independent of supervised classification algorithms, we used Gravity Spy as a benchmark to quantify its performance: in our FD-encoded auxiliary channel data, constituted by 2688 times where glitches were present in $h(t)$, we found a 6.6% of anomalies caused by unknown morphologies labelled as their closest glitch class, similar morphologies assigned the incorrect class or glitch overlaps being overlooked.

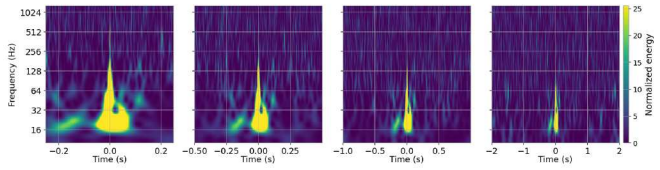
Data-driven approaches, such as the one demonstrated in this work, can unveil anomalies present in the data and reveal relations between glitch classes, allowing us to further understand the glitch population. In future work, this approach will be extended to the general population of LIGO-Livingston and other interferometers to enhance the identification of glitches. Moreover, we will provide an anomaly score to assess the significance of the outliers found by our algorithm, and explore a data fusion representation containing both the FD-encoded auxiliary channel data and the strain $h(t)$ in time-frequency representation, providing not only information about the physical process within the detector but also their impact on $h(t)$. Last but not least, we will investigate the correlation between safe auxiliary channels highlighted by our model and glitches appearing in $h(t)$, in the search of witness auxiliary channels with the goal of improving glitch mitigation in GW searches.



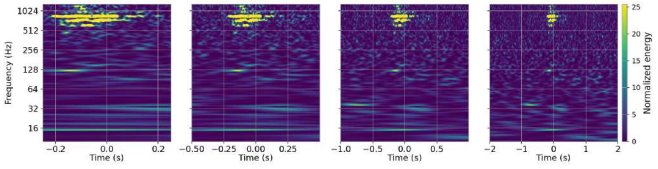
(a) Glitch labeled as `Whistle`, but it was not observed on other Gravity Spy's classes, so it could be a novel morphology.



(b) Glitch labeled as `Tomte`, but its shape is consistent with `Fast_Scattering`, being a misclassification.

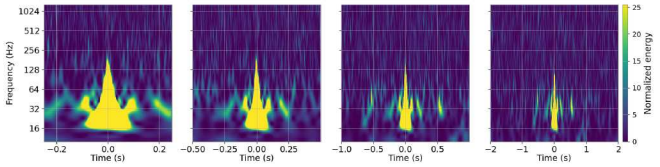


(c) Glitch labeled as `Scattered_Light`, but its shape is consistent an overlap between a `Tomte` and an unknown morphology.

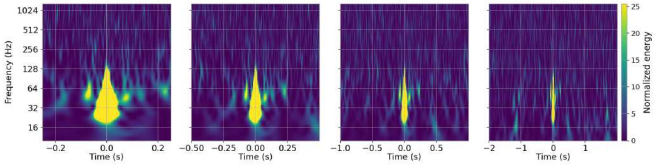


(d) Glitch labeled as `Whistle`. The observed shape is inconsistent with any Gravity Spy classes, constituting an anomalous glitch morphology.

FIG. 11: **Region 3:** Example of anomalous glitches in the embedded space.

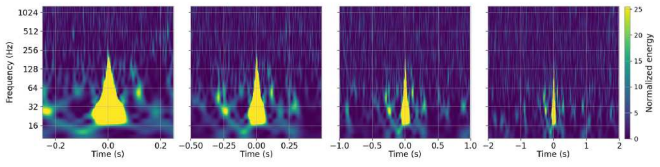


(a) Glitch labeled as `Tomte`, but consistent with `Koi_Fish` class, being a misclassification.

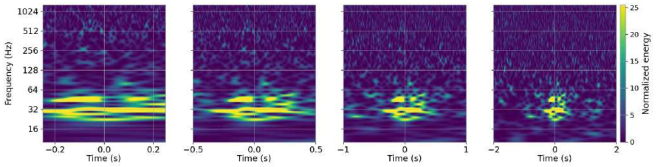


(b) Glitch labeled as `Tomte`, but seems to be an overlap between a `Tomte` and a `Scratchy`, being an overlap.

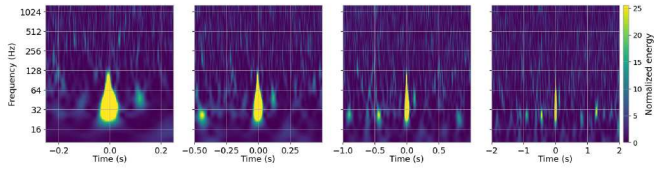
FIG. 12: **Region 4:** Example of anomalous glitches in the embedded space.



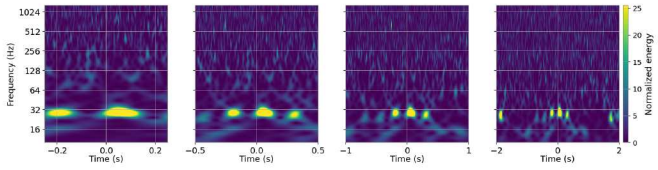
(a) Glitch labeled as `Tomte`, but it could be an overlap between `Scratchy` and `Tomte`.



(b) Glitch labeled as `Scattered_Light`, but it was not observed on other Gravity Spy's classes, so it could be a novel morphology.



(c) Glitch labeled as `Tomte`, but it could be an overlap between `Scratchy` and `Tomte`.



(d) Glitch labeled as `Scattered_Light`, but its morphology is consistent with `Scratchy`, being a misclassification.

FIG. 13: **Region 5:** Example of anomalous glitches in the embedded space.

VII. ACKNOWLEDGEMENTS

M.L. is supported by the research program of the Netherlands Organisation for Scientific Research (NWO). S.C is supported by the National Science Foundation under Grant No. PHY-2309332. M.C. is supported by the National Science Foundation Grants NSF PHY-2011334 and NSF PHY-2308693. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by the National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation.

[1] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. Lett. **116**, 061102 (2016).
[2] J. Aasi *et al.* (LIGO Scientific), Class. Quant. Grav. **32**, 074001 (2015), arXiv:1411.4547 [gr-qc].
[3] F. Acernese *et al.* (VIRGO), Class. Quant. Grav. **32**, 024001 (2015), arXiv:1408.3978 [gr-qc].
[4] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. X **9**, 031040 (2019), arXiv:1811.12907 [astro-ph.HE].
[5] R. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. X **11**, 021053 (2021), arXiv:2010.14527 [gr-qc].
[6] R. Abbott *et al.* (LIGO Scientific, VIRGO, KAGRA), arXiv e-prints (2021), arXiv:2111.03606 [gr-qc].
[7] A. H. Nitz *et al.*, Astrophys. J. **872**, 195 (2019), arXiv:1811.01921 [gr-qc].
[8] A. H. Nitz *et al.*, Astrophys. J. **891**, 123 (2020), arXiv:1910.05331 [astro-ph.HE].
[9] A. H. Nitz *et al.*, Astrophys. J. **922**, 76 (2021), arXiv:2105.09151 [astro-ph.HE].
[10] A. H. Nitz *et al.*, Astrophys. J. **946**, 59 (2023), arXiv:2112.06878 [astro-ph.HE].
[11] B. Zackay *et al.*, Phys. Rev. D **104**, 063030 (2021), arXiv:1910.09528 [astro-ph.HE].
[12] S. Olsen *et al.*, Phys. Rev. D **106**, 043009 (2022), arXiv:2201.02252 [astro-ph.HE].
[13] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. Lett. **119**, 161101 (2017), arXiv:1710.05832 [gr-qc].
[14] C. Meegan *et al.*, The Astrophysical Journal **702**, 791 (2009).
[15] A. Goldstein *et al.*, The Astrophysical Journal Letters **848**, L14 (2017).
[16] A. Perego *et al.*, Astrophys. J. Lett. **850**, L37 (2017), arXiv:1711.03982 [astro-ph.HE].
[17] L. Blackburn *et al.*, Class. Quant. Grav. **25**, 184004 (2008), arXiv:0804.0800 [gr-qc].
[18] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Class. Quant. Grav. **33**, 134001 (2016), arXiv:1602.03844 [gr-qc].
[19] S. Soni *et al.* (LIGO), Class. Quant. Grav. **38**, 025016 (2020), arXiv:2007.14876 [astro-ph.IM].
[20] M. Cabero *et al.*, Class. Quant. Grav. **36**, 15 (2019), arXiv:1901.05093 [physics.ins-det].
[21] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Class. Quant. Grav. **35**, 065010 (2018), arXiv:1710.02185 [gr-qc].
[22] R. Abbott *et al.* (KAGRA, LIGO Scientific, VIRGO), Phys. Rev. D **106**, 102008 (2022), arXiv:2201.00697 [gr-qc].
[23] B. Steltner *et al.*, arXiv e-prints (2023), arXiv:2303.04109 [gr-qc].
[24] R. Abbott *et al.* (KAGRA, Virgo, LIGO Scientific), Phys. Rev. D **104**, 022004 (2021), arXiv:2101.12130 [gr-qc].
[25] B. Steltner, M. A. Papa, and H. B. Eggenstein, Phys. Rev. D **105**, 022005 (2022), arXiv:2105.09933 [gr-qc].
[26] C. Pankow *et al.*, Physical Review D **98**, 084016 (2018).
[27] D. Davis *et al.*, Classical and Quantum Gravity **36**, 055011 (2019).
[28] J. C. Driggers *et al.*, Physical Review D **99**, 042001 (2019).
[29] J. Powell, Class. Quant. Grav. **35**, 155017 (2018), arXiv:1803.11346 [astro-ph.IM].
[30] B. Covas *et al.* (LSC), Phys. Rev. D **97**, 082002 (2018), arXiv:1801.07204 [astro-ph.IM].
[31] D. Davis, L. V. White, and P. R. Saulson, Classical and Quantum Gravity **37**, 145001 (2020).
[32] D. Davis *et al.* (LIGO), Class. Quant. Grav. **38**, 135014 (2021), arXiv:2101.11673 [astro-ph.IM].
[33] M. Zevin *et al.*, Class. Quant. Grav. **34**, 064003 (2017), arXiv:1611.04596 [gr-qc].
[34] D. George, H. Shen, and E. A. Huerta, arXiv e-prints (2017), arXiv:1706.07446 [gr-qc].
[35] S. Bahaadini *et al.*, Information Sciences **444**, 172 (2018).
[36] J. Glanzer *et al.*, Class. Quant. Grav. **40**, 065004 (2023), arXiv:2208.12849 [gr-qc].
[37] T. A. Ferreira and C. A. Costa, Class. Quant. Grav. **39**, 165013 (2022).
[38] M. Razzano *et al.*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **1048**, 167959 (2023).
[39] S. Alvarez-Lopez *et al.*, arXiv preprint arXiv:2304.09977 (2023).
[40] S. Soni *et al.*, Class. Quant. Grav. **38**, 195016 (2021), arXiv:2103.12104 [gr-qc].
[41] B. Abbott *et al.*, Classical and Quantum Gravity **37**, 055002 (2020).
[42] G. M. Harry *et al.*, Classical and Quantum Gravity **27**, 084006 (2010).
[43] J. R. Smith *et al.*, Class. Quant. Grav. **28**, 235005 (2011), arXiv:1107.2948 [gr-qc].
[44] R. E. Colgan *et al.*, Phys. Rev. D **101**, 102003 (2020), arXiv:1911.11831 [astro-ph.IM].
[45] R. E. et al., Machine Learning: Science and Technology **2**, 015004 (2020).
[46] F. Robinet *et al.*, SoftwareX **12**, 100620 (2020), arXiv:2007.11374 [astro-ph.IM].
[47] M. Cavaglia, Class. Quant. Grav. **39**, 135012 (2022), arXiv:2201.09984 [gr-qc].
[48] J. Theiler, Journal of the Optical Society of America A **7**, 1055 (1990).

[49] T. Gneiting, H. Ševčíková, and D. B. Percival, Statistical Science , 247 (2012).
[50] C. Tricot, in *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 91 (Cambridge University Press, 1982) pp. 57–74.
[51] M. Fernández-Martínez and M. Sánchez-Granero, Topology and its Applications **163**, 93 (2014).
[52] B. Dubuc and S. Dubuc, SIAM Journal on Numerical Analysis **33**, 602 (1996).
[53] S. K. Lam, A. Pitrou, and S. Seibert, in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (2015) pp. 1–6.
[54] J. C. Brown, The Journal of the Acoustical Society of America **89**, 425 (1991).
[55] J. Aasi *et al.* (LIGO Scientific, VIRGO), Class. Quant. Grav. **32**, 115012 (2015), arXiv:1410.7764 [gr-qc].
[56] L. Nuttall *et al.*, Class. Quant. Grav. **32**, 245005 (2015), arXiv:1508.07316 [gr-qc].
[57] L. Nuttall *et al.*, Class. Quant. Grav. **32**, 245005 (2015), arXiv:1508.07316 [gr-qc].
[58] S. Soni, others, and LIGO Scientific Collaboration, Classical and Quantum Gravity **38**, 025016 (2021), arXiv:2007.14876 [astro-ph.IM].
[59] D. George and E. A. Huerta, Phys. Rev. D **97**, 044039 (2018), arXiv:1701.00008 [astro-ph.IM].
[60] H. Gabbard *et al.*, Phys. Rev. Lett. **120**, 141103 (2018), arXiv:1712.06041 [astro-ph.IM].
[61] A. Menéndez-Vázquez *et al.*, Phys. Rev. D **103**, 062004 (2021), arXiv:2012.10702 [gr-qc].
[62] G. Baltus *et al.*, Phys. Rev. D **103**, 102003 (2021), arXiv:2104.00594 [gr-qc].
[63] G. Baltus *et al.*, Phys. Rev. D **106**, 042002 (2022), arXiv:2205.04750 [gr-qc].
[64] V. Skliris, M. R. K. Norman, and P. J. Sutton, arXiv e-prints (2020), arXiv:2009.14611 [astro-ph.IM].
[65] M. López *et al.*, in *International Conference on Content-Based Multimedia Indexing* (2021).
[66] M. L. Portilla *et al.*, Phys. Rev. D **103**, 063011 (2021), arXiv:2011.13733 [astro-ph.IM].
[67] V. Boudart, Phys. Rev. D **107**, 024007 (2023), arXiv:2210.04588 [gr-qc].
[68] V. Boudart and M. Fays, Phys. Rev. D **105**, 083007 (2022), arXiv:2201.08727 [gr-qc].
[69] S. Bini *et al.*, Class. Quant. Grav. **40**, 135008 (2023), arXiv:2303.05986 [gr-qc].
[70] Q. Meijer *et al.*, arXiv e-prints (2023), arXiv:2308.12323 [astro-ph.IM].
[71] M. .Cavaglia *et al.*, arXiv e-prints , arXiv:2002.04591 (2020), arXiv:2002.04591 [astro-ph.IM].
[72] J. M. Antelis *et al.*, Phys. Rev. D **105**, 084054 (2022), arXiv:2111.07219 [gr-qc].
[73] L. M. Modafferi, R. Tenorio, and D. Keitel, Phys. Rev. D **108**, 023005 (2023), arXiv:2303.16720 [astro-ph.HE].
[74] J. M. et al., Class. Quant. Grav. **38**, 155005 (2021), arXiv:2103.01641 [astro-ph.IM].
[75] M. Lopez *et al.*, Phys. Rev. D **106**, 023027 (2022), arXiv:2203.06494 [astro-ph.IM].
[76] T. Dooney, S. Bromuri, and L. Curier, arXiv e-prints (2022), arXiv:2209.13592 [astro-ph.IM].
[77] J. Yan *et al.*, Mon. Not. Roy. Astron. Soc. **515**, 4606 (2022), arXiv:2207.04001 [astro-ph.HE].
[78] J. Powell *et al.*, Class. Quant. Grav. **40**, 035006 (2023), arXiv:2207.00207 [astro-ph.IM].
[79] M. Lopez *et al.*, arXiv e-prints (2022), arXiv:2205.09204 [astro-ph.IM].
[80] F. Morawski *et al.*, Mach. Learn. Sci. Tech. **2**, 045014 (2021), arXiv:2103.07688 [astro-ph.IM].
[81] E. Cuoco *et al.*, Mach. Learn. Sci. Tech. **2**, 011002 (2021), arXiv:2005.03745 [astro-ph.HE].
[82] D. Bank, N. Koenigstein, and R. Giryes, arXiv preprint arXiv:2003.05991 (2020).
[83] K. O'Shea and R. Nash, arXiv e-prints , arXiv:1511.08458 (2015), arXiv:1511.08458 [cs.NE].
[84] X. Guo *et al.*, in *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24* (Springer, 2017) pp. 373–382.
[85] K. O'Shea and R. Nash, arXiv preprint arXiv:1511.08458 (2015).
[86] Y. Zhang, in *ICONIP17-DCEC. Available online: http://users.cecs.anu.edu.au/Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf* (2018).
[87] R. Priemer, *Introductory signal processing*, Vol. 6 (World scientific, 1991) pp. 212–215.
[88] M. Abadi *et al.*, arXiv preprint arXiv:1603.04467 (2016).
[89] F. Chollet *et al.*, "Keras," https://keras.io (2015).
[90] R. Arora *et al.*, arXiv preprint arXiv:1611.01491 (2016).
[91] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).
[92] T. Kim *et al.*, arXiv preprint arXiv:2105.08919 (2021).
[93] T. Chai and R. R. Draxler, Geoscientific model development discussions **7**, 1525 (2014).
[94] F. Pedregosa *et al.*, the Journal of machine Learning research **12**, 2825 (2011).
[95] L. V. der Maaten and G. Hinton, Journal of machine learning research **9** (2008).