MACHINE LEARNING
Science and Technology

PAPER • OPEN ACCESS

Enhancing the rationale of convolutional neural networks for glitch classification in gravitational wave detectors: a visual explanation

To cite this article: Naoki Koyama *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 035028

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
25 March 2024

REVISED
17 June 2024

ACCEPTED FOR PUBLICATION
15 July 2024

PUBLISHED
26 July 2024

PAPER

Enhancing the rationale of convolutional neural networks for glitch classification in gravitational wave detectors: a visual explanation

Naoki Koyama[1,*], Yusuke Sakai[2], Seiya Sasaoka[3], Diego Dominguez[3], Kentaro Somiya[3], Yuto Omae[4], Yoshikazu Terada[5,6], Marco Meyer-Conde[2,7,8] and Hirotaka Takahashi[2,9,10,*]

[1] Graduate School of Science and Technology, Niigata University, 8050 Ikarashi-2-no-cho, Nishi-ku, Niigata City, Niigata 950-2181, Japan
[2] Research Center for Space Science, Advanced Research Laboratories and Department of Design and Data Science, Tokyo City University, 3-3-1 Ushikubo-Nishi, Tsuzuki-ku, Yokohama, Kanagawa 224-8551, Japan
[3] Department of Physics, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8551, Japan
[4] Artificial Intelligence Research Center, College of Industrial Technology, Nihon University, 1-2-1 Izumi-cho, Narashino, Chiba 275-8575, Japan
[5] Graduate School of Engineering Science, Osaka University,1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
[6] RIKEN Center for Advanced Intelligence Project (AIP), 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan
[7] Department of Physics, Graduate School of Science, Osaka Metropolitan University, 3-3-138 Sugimoto-cho, Sumiyoshi-ku, Osaka City, Osaka 558-8585, Japan
[8] University of Illinois at Urbana-Champaign,Department of Physics,Urbana, IL 61801-3080, United States of America
[9] Institute for Cosmic Ray Research (ICRR), The University of Tokyo, 5-1-5 Kashiwa-no-Ha, Kashiwa City, Chiba 277-8582, Japan
[10] Earthquake Research Institute, The University of Tokyo, 1-1-1 Yayoi, Bunkyo-ku, Tokyo 113-0032, Japan
[*] Authors to whom any correspondence should be addressed.

E-mail: f22l004a@mail.cc.niigata-u.ac.jp and hirotaka@tcu.ac.jp

Keywords: gravitational waves, glitch noise classification, Gravity Spy, explainable AI, saliency map, score-CAM, UMAP

Abstract

In the pursuit of detecting gravitational waves, ground-based interferometers (e.g. LIGO, Virgo, and KAGRA) face a significant challenge: achieving the extremely high sensitivity required to detect fluctuations at distances significantly smaller than the diameter of an atomic nucleus. Cutting-edge materials and innovative engineering techniques have been employed to enhance the stability and precision of the interferometer apparatus over the years. These efforts are crucial for reducing the noise that masks the subtle gravitational wave signals. Various sources of interference, such as seismic activity, thermal fluctuations, and other environmental factors, contribute to the total noise spectra characteristic of the detector. Therefore, addressing these sources is essential to enhance the interferometer apparatus's stability and precision. Recent research has emphasised the importance of classifying non-stationary and non-Gaussian glitches, employing sophisticated algorithms and machine learning methods to distinguish genuine gravitational wave signals from instrumental artefacts. The time-frequency-amplitude representation of these transient disturbances exhibits a wide range of new shapes, variability, and features, reflecting the evolution of interferometer technology. In this study, we developed a convolutional neural network model to classify glitches using spectrogram images from the Gravity Spy O1 dataset. We employed score-class activation mapping and the uniform manifold approximation and projection algorithm to visualise and understand the classification decisions made by our model. We assessed the model's validity and investigated the causes of misclassification from these results.

1. Introduction

Observation of gravitational waves is currently taking place within the international gravitational wave network [1] involving Advanced LIGO [2], Advanced Virgo [3], and KAGRA, the Large-scale Cryogenic Gravitational Wave Telescope located in Japan [4]. Since 2015, the collaborative efforts have led to the detection and subsequent data refinement of over 90 gravitational wave events and their candidate signals [5–9].

The primary challenge in gravitational wave data analysis lies in extracting faint signals from the background noise present in interferometric data. The enhanced sensitivity of interferometers makes them susceptible to various disturbances, collectively referred to as 'noise'. Non-stationary and non-Gaussian noises, known as 'transient glitch' noises, can have various origins, including ground movement, air pressure, optical suspensions, laser fluctuations, vacuum, mirrors, and other environmental and instrumental conditions. These glitches can lead to false positive detections and introduce specific biases in astrophysical parameter estimation, both of which can diminish the reliability of candidate signals and decrease the amount of data suitable for analysis. Therefore, it is essential to identify the causes of these glitches accurately. This challenge can be addressed either by implementing corrective measures within the detector or by excluding these glitches from the detector's output. Achieving the desired sensitivity requires identifying and eliminating such glitches, which display diverse signatures in terms of time, frequency, and amplitude. Notably, the evolving nature of interferometer technology leads to the emergence of glitch forms, emphasising the importance of developing new glitch mitigation techniques.

In recent years, machine learning, especially deep learning, has gained traction in gravitational wave physics, as evidenced by comprehensive reviews [10, 11]. A notable example is the Gravity Spy project [12–14], where supervised learning has played a crucial role in classifying glitch noises. Despite significant progress, identifying all glitch morphologies from interferometric data using deep learning models is still challenging today. While deep learning has enhanced automatic glitch classification, fundamental limitations remain: the lack of transparency in the decision-making process and the interpretability of the models. In response to these concerns, the field of explainable artificial intelligence (explainable AI) [15] has recently gained attention. Within this field, class activation mapping (CAM) [16] and its variants are prominent for visualising influential regions in input images that contribute to specific predictions. CAM calculates a weighted sum of the final convolutional layer's outputs using the global average pooling layer's outputs as weights, sometimes at the expense of accuracy. This limitation led to the development of gradient-weighted class activation mapping (Grad-CAM) [17], which relies on the gradients from the prediction to assign weights without requiring structural modifications to the network. Built upon these advances, score-class activation mapping (Score-CAM) [18] is a gradient-free alternative that produces more accurate saliency maps than Grad-CAM.

Additional data analysis tools are essential to enhance further understanding of the relationship between classified glitch noise and its visual representation by Score-CAM. The uniform manifold approximation and projection (UMAP) [19] algorithm works by approximating the manifold structure of data using fuzzy simplicial sets. This method translates high-dimensional data into a lower-dimensional space, revealing intricate patterns and structures within the classified glitches that may not be immediately discernible in the initial high-dimensional representation.

This study explores multi-class glitch classifiers using a convolutional neural networks (CNN) trained on the Gravity Spy O1 dataset. We describe the inference process using explainable AI methods, namely Score-CAM. In particular, we evaluate how this technique connects morphological features to classification. Additionally, we integrate UMAP with this approach to better understand the 'rationale' behind the CNN's decision-making process. An attempt at sufficient dimension reduction is also performed.

This paper is organised as follows. In section 2, we overview our datasets, the architecture of the CNN models, and the theoretical background of the methods used to visualise the rationale of the estimations. In section 3, after discussing the classification performance of the supervised learning architecture, we discuss how the model classifies glitches and the reasons for misclassification based on the results of visualising the classification rationale. Section 4 offers a summary of this study.

2. Methodology

2.1. Datasets

This study makes use of the LIGO O1 Gravity Spy dataset [12, 13] processed using the Omicron software [20] for multi-resolution time-frequency interferometric data analysis. Data were selected within peak frequencies ranging from 10 Hz to 2048 Hz, corresponding to the bandwidth sensitivity of compact binary coalescences. Data with a signal-to-noise ratio lower than 7.5 are excluded, as they have less influence on the signal search.

Spectrograms centred on the identified glitch noises were created as 2D images using Omega Scan [21]. A total of 22 labels related to the characteristics and causes of the glitch noises (e.g. 'Blip', 'Power_Line', 'Koi_Fish', etc) were assigned through collaboration with detector experts and volunteer citizen scientists. This collaborative effort [22–31] utilises the Gravity Spy sparkling cloud resources [11], where both glitch

[11] https://www.zooniverse.org/projects/zooniverse/gravity-spy.

images and labels are saved as datasets [32]. Glitch dataset images and labels used in this study were provided by [32]. This dataset consists of 8535 spectrogram images of glitch noise, as outlined in table 1. Glitch spectrograms were recorded for four different time durations: 0.5 s, 1.0 s, 2.0 s, and 4.0 s. These images of glitch noise were converted in gray-scaled images from colour-scaled images.

In the pre-processing stage, 24 pixels were trimmed from both ends of the original glitch noise spectrogram images (272px × 224px) in the time direction, resulting in a final size of 224px × 224px. These images, representing four different time scales, were then combined in a 2 × 2 grid to create a 'merged-view model' of 448px × 448px, as shown in figure 1. The labels of the x and y axes are removed in this step. This pre-processing step improves the classification accuracy by reducing discrepancies arising from varying time scales. After selection, the dataset was divided into training, validation, and test datasets with proportions of 70%, 15%, and 15%, respectively. During the training process, the initial weights of the model were set using the training dataset, and the model hyperparameters were evaluated using the validation dataset. The final model performance was then determined unbiasedly using the test dataset.

2.2. Architecture description

Our model adopts a variation of the CNN architecture as proposed by Bahaadini *et al* [22]. Table 2 introduces a comparison of the two architectures.

Our current model covers a total of 22 glitches, which is the original list of 20 glitch classes [12] but introducing two new additional classes, namely '1080Lines' and '1400Ripples' [13]. Both glitch images and labels are taken from [13, 32].

Our model architecture, with deeper convolutional layers compared to Bahaadini's model, also includes ReLU activation functions and max-pooling layers. Finally, the feature maps were transformed into a one-dimensional vector through a flattening layer and sent to a fully connected layer to produce a 22-class output. This allows additional glitch categories to be effectively distinguished and their variability to be captured.

The model was trained using a cross-entropy loss function and Adadelta [33] as the optimisation algorithm. The model underwent training for 20 epochs using a learning rate of 0.1 and a batch size of 60. In this implementation, the model relies on the PyTorch library [34], and training was performed using three GPUs: one NVIDIA GeForce RTX3060 and two NVIDIA GeForce RTX3090.

2.3. Score-CAM

Score-CAM [18] visually explains a CNN feature by showing the part of the input image on which the model's prediction is based. Score-CAM maps are generated by estimating the importance of each channel in the feature maps. These are computed as follows:

(i) The outputs of the last convolutional layer of the network are obtained as feature maps.

(ii) Each feature map is upsampled to the same input size and normalised. Let X be the input image and H^k be the upsampled and normalised feature map. The Hadamard product of the input and the feature map gives a masked image, highlighting regions of interest.

(iii) The importance for the kth feature map is computed as the difference between the scores of the masked image and the baseline image X_b, which is given by

$$C\left(A_l^k\right) = f\left(X \circ H_l^k\right) - f(X_\mathrm{b}), \tag{1}$$

where f is the output of the network and $\circ$ is the Hadamard product.

(iv) Given a class c and a layer l, the Score-CAM map is generated as the weighted sum of the feature maps:

$$L^c_{\text{Score-CAM}} = \mathrm{ReLU}\left(\sum_k \alpha_k^c A_l^k\right), \tag{2}$$

where

$$\alpha_k^c = C\left(A_l^k\right). \tag{3}$$

2.4. Uniform manifold approximation and projection

In addition to the previous exercise, a refined approach for explaining classification models using deep learning, particularly focusing on discriminative features, involves visualising the features through dimension reduction. Compressing and visualising high-dimensional features into two or three dimensions can help us understand the cluster structure of features in relation to the input data. Various methods have

Table 1. Dataset distributions for each class and model accuracies for the test dataset.

Label/Class	Training dataset	Validation dataset	Test dataset	Model accuracy
1080Lines	69.8% (229/328)	14.9% (49/328 images)	15.2% (50/328 images)	100.0%
1400Ripples	69.8% (162/232 images)	15.1% (35/232 images)	15.1% (35/232 images)	100.0%
Air_Compressor	69.0% (40/58 images)	15.5% (9/58 images)	15.5% (9/58 images)	100.0%
Blip	70.0% (1308/1869 images)	15.0% (280/1869 images)	15.0% (281/1869 images)	99.3%
Chirp	69.7% (46/66 images)	15.2% (10/66 images)	15.2% (10/66 images)	100.0%
Extremely_Loud	69.8% (317/454 images)	15.0% (68/454 images)	15.2% (69/454 images)	100.0%
Helix	69.9% (195/279 images)	15.1% (42/279 images)	15.1% (42/279 images)	100.0%
Koi_Fish	70.0% (581/830 images)	14.9% (124/830 images)	15.1% (125/830 images)	99.2%
Light_Modulation	70.0% (401/573 images)	15.0% (86/573 images)	15.0% (86/573 images)	98.8%
Low_Frequency_Burst	69.9% (459/657 images)	15.1% (99/657 images)	15.1% (99/657 images)	99.0%
Low_Frequency_Lines	70.0% (317/453 images)	15.0% (68/453 images)	15.0% (68/453 images)	97.1%
No_Glitch	69.6% (126/181 images)	14.9% (27/181 images)	15.5% (28/181 images)	96.4%
Paired_Doves	66.7% (18/27 images)	14.8% (4/27 images)	18.5% (5/27 images)	80.0%
Power_Line	70.0% (317/453 images)	15.0% (68/453 images)	15.0% (68/453 images)	100.0%
Repeating_Blips	69.8% (199/285 images)	15.1% (43/285 images)	15.1% (43/285 images)	95.4%
Scattered_Light	69.9% (321/459 images)	15.0% (69/459 images)	15.0% (69/459 images)	97.1%
Scratchy	69.8% (247/354 images)	15.0% (53/354 images)	15.3% (54/354 images)	98.2%
Tomte	69.8% (81/116 images)	14.7% (17/116 images)	15.5% (18/116 images)	100.0%
Violin_Mode	69.9% (330/472 images)	15.0% (71/472 images)	15.0% (71/472 images)	98.6%
Wandering_Line	68.2% (30/44 images)	15.9% (7/44 images)	15.9% (7/44 images)	85.7%
Whistle	69.8% (213/305 images)	15.1% (46/305 images)	15.1% (46/305 images)	91.3%
Unidentified_Glitch (None_of_the_Above)	69.3% (61/88 images)	14.8% (13/88 images)	15.9% (14/88 images)	71.4%

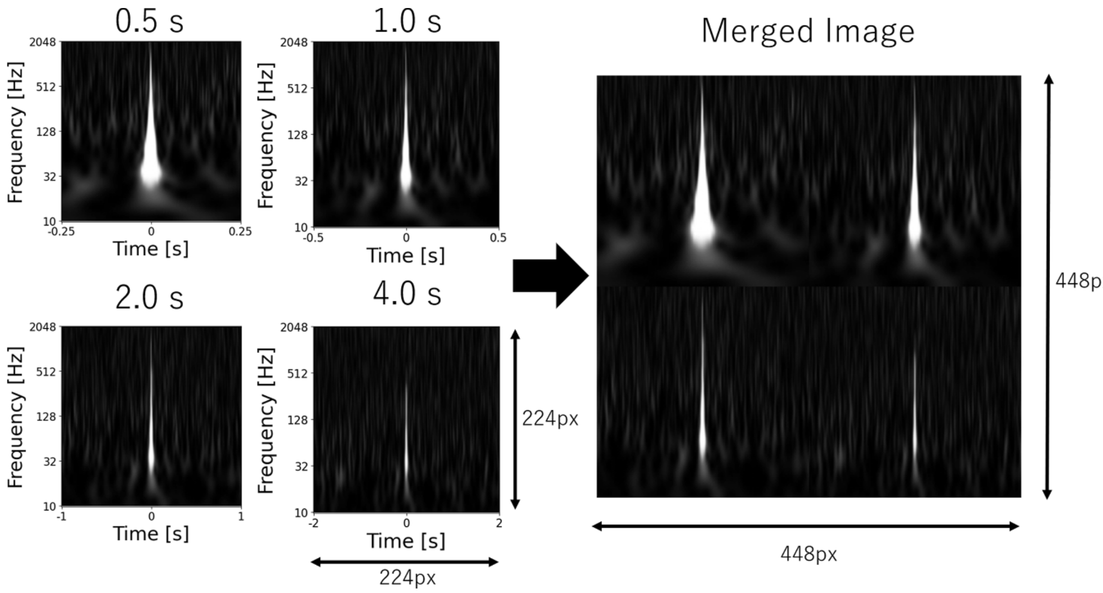


Figure 1. Pre-processing of the Gravity Spy dataset: Each of the initial glitch noise spectrogram image is taken from [32], and cast into grey-scaled images. Combining the glitch noise spectrograms of four different time durations (0.5 s, 1.0 s, 2.0 s, and 4.0 s) into a 2 × 2 layout image of 448px × 448px. The labels of the x and y axes are removed in this step.

Table 2. CNN architectures for glitch classification.

Bahaadini *et al* [22]	In this study
Input 94 × 114	Input 448 × 448
5 × 5 Convolutional layer (128)	5 × 5 Convolutional layer (32) with ReLU
2 × 2 Max-pooling layer	2 × 2 Max-pooling layer
ReLU	
5 × 5 Convolutional layer (128)	5 × 5 Convolutional layer (32) with ReLU
2 × 2 Max-pooling layer	2 × 2 Max-pooling layer
ReLU	
	5 × 5 Convolutional layer (64) with ReLU
	2 × 2 Max-pooling layer
	5 × 5 Convolutional layer (64) with ReLU
	2 × 2 Max-pooling layer
	5 × 5 Convolutional layer (128) with ReLU
Fully connected layer (256)	Fully connected layer (100352)
Softmax (20)	Softmax (22)

been proposed for dimension reduction. The principal component analysis is a representative linear analysis method that can capture the global structure of features. In contrast, the nonlinear method known as t-distributed stochastic neighbour embedding (t-SNE) [35, 36] can capture more local structures. UMAP [19] can also represent the local structures of features at a lower computational cost than t-SNE.

We used UMAP to visualise the extracted features. The UMAP algorithm comprises two phases: the graph construction phase, which constructs a neighbourhood graph from high-dimensional data, and the graph embedding phase, which embeds the graph's vertices in a lower-dimensional space.

A. Graph Construction

Let $X = \{x_1, \ldots, x_n\}$ be the input dataset (or features), with a (Euclidean) metric $d : X \times X \to \mathbb{R}_{\geqslant 0}$, where n represents the number of data points. In the context of UMAP, the construction of a graph representation of X proceeds as follows:

A.1. Nearest neighbour search

Using some nearest neighbour search algorithm, construct the unweighted directed k-nearest neighbour graph (V, E), where V is the set of vertices, and E is the set of directed edges. Each is corresponding to each data point. Let $x_i^{(j)}$ be the jth nearest neighbour of the ith data point x_i. Then, $E = \{(i,j) \mid d(x_i, x_j) \leqslant d(x_i, x_i^{(k)})\}$.

A.2. Determination of local scale parameters
A scale parameter is selected to constrain the local distance between the data points. The distance between x_i and its nearest neighbour is defined as $\rho_i = \min\{d(x_i, x_i^{(j)}) | 1 \leqslant j \leqslant k, d(x_i, x_i^{(j)}) > 0\}$. The scale parameter σ_i is determined to satisfy the following equation:

$$\log_2(k) = \sum_{j=1}^{k} \exp\left(-\frac{\max\left(0, d\left(x_i, x_i^{(j)}\right) - \rho_i\right)}{\sigma_i}\right). \tag{4}$$

The scale parameter σ_i is small in high-density regions, while it takes a large value in low-density regions. This adaptive choice can be interpreted as a local standardization of the similarity and is essential to capture the hidden local structures.

A.3. Construction of the symmetric weighted graph
Construct the symmetric weighted graph (V, W) as follows:

$$W = A + A^T - A \circ A^T, \tag{5}$$

where $W = (w_{ij})_{n\times n}$ denotes the $n \times n$ symmetric adjacency matrix, $\circ$ is the Hadamard product, A^T is the transpose of matrix A, and the asymmetric matrix $A = (a_{ij})_{n\times n}$ is defined as

$$a_{ij} = \begin{cases} \exp\left(-\frac{\max\left(0, d\left(x_i, x_j\right) - \rho_i\right)}{\sigma_i}\right) & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

B. Graph Embedding
Let $\boldsymbol{s}_i = (s_{i1}, \ldots, s_{id})^T$ be a d-dimensional representation of the ith data point, and $S = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)^T$ denotes the coordinate matrix. The UMAP proceeds the graph embedding such that the weights w_{ij} are approximated by the following weights based on the low-dimensional representation S:

$$v_{ij}(S) = \frac{1}{1 + \alpha \|\boldsymbol{s}_i - \boldsymbol{s}_j\|_2^{2\beta}}, \tag{6}$$

where $\alpha > 0$ and $\beta > 0$ are hyperparameters, and $\|\cdot\|_2$ denotes the Euclidean norm. More precisely, UMAP finds the coordinate S, which minimised the following cost function:

$$C_{\text{UMAP}}(S) = \sum_{i\neq j} \left\{ w_{ij} \log\left(\frac{w_{ij}}{v_{ij}(S)}\right) + (1 - w_{ij}) \log\left(\frac{1 - w_{ij}}{1 - v_{ij}(S)}\right) \right\}. \tag{7}$$

Here, we note that the cost function C_{UMAP} is known as 'fuzzy set cross entropy'.
In the UMAP algorithm, α and β in equation (6) are determined by nonlinear least-squares fitting against the curve $\Psi : \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$, as follows:

$$\Psi\left(\boldsymbol{s}_i, \boldsymbol{s}_j\right) = \begin{cases} 1 & \text{if } \|\boldsymbol{s}_i - \boldsymbol{s}_j\|_2 \leqslant \delta, \\ \exp\left\{-\left(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|_2 - \delta\right)\right\} & \text{otherwise,} \end{cases} \tag{8}$$

where δ is a new hyperparameter instead of α or β. The hyperparameter δ is called as 'min-dist' and represents the desired separation between close points in the embedding space. For more details on UMAP, see the original paper [19].

3. Results and discussions

3.1. Glitch classification

The learning curve in figure 2 shows a convergence of losses and accuracies up to epoch 20. The behaviour of the learning curve in figure 2 is consistent with the figures 9(a) and (c) as shown in [30]. Although [30] discusses some advanced learning strategies, early stopping was applied in our case to suppress overfitting for simplicity. The training concluded at epoch 20, leading to the model's performance evaluation using the test dataset. The accuracy for each glitch class, detailed at the right end of table 1, along with the confusion matrix in figure A1 of appendix A, show the model's efficacy. The overall classification accuracy for the test dataset is about 98.1%, which is on par with the 96.9% classification accuracy reported by [22]. Among the 22 glitch classes, 18 exhibited an accuracy greater than 95%.

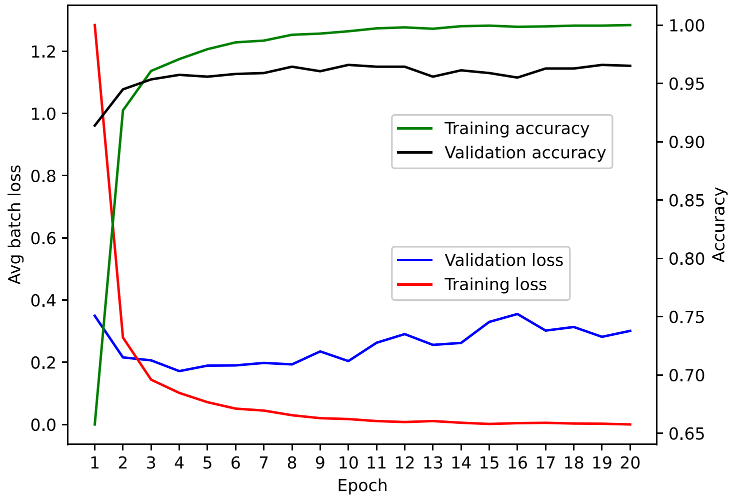


Figure 2. Learning curve during the training phase. Accuracy and average batch loss for the training and validation datasets are plotted against the number of epochs.

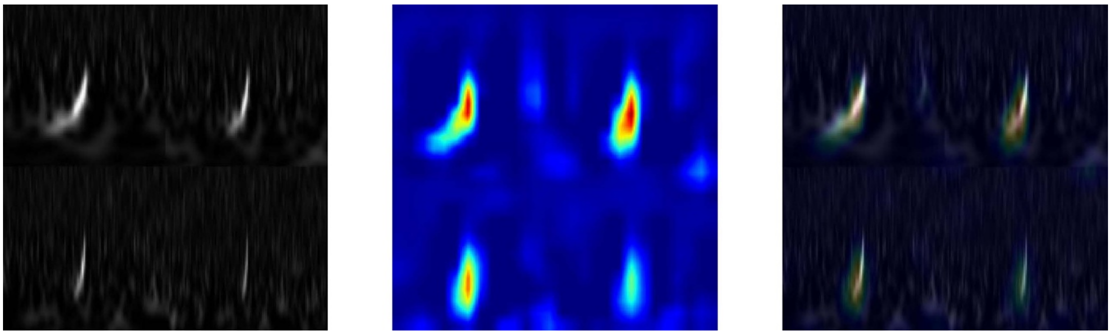


Figure 3. Estimation rationale of a correctly classified 'Chirp' sample. The figure comprises an input image (left); an estimation rationale heatmap (centre) is obtained from Score-CAM using the input image and backpropagated to the 'Chirp' Softmax output; the overlapping picture (right) highlights the coincident region between the input and the heatmap.

Figure A1 and the detailed table 1 both highlight some clear limitations of the classification accuracy for a few classes, namely 'Paired_Doves', 'Wandering_Line', 'Whistle', and 'Unidentified_Glitch'. Both 'Paired_Doves' and 'Wandering_Line' classes showed lower accuracies due to their limited proportion in the training dataset. Moreover, the model's classification of 'Whistles' was hindered by the wide variety of characteristics including shape and position. It is noticeable that an expanded training dataset is necessary to further address these challenges. Eventually, 'Unidentified_Glitch' gathers all other glitches outside the defined 21 classes, showing a wide variety of unrelated spectral mixed shapes, and resulting in a lower classification accuracy.

3.2. Saliency map

Figures 3 and 4 show examples of saliency maps generated by Score-CAM showing the estimation rationale. The 'estimation rationale' heatmaps obtained from Score-CAM highlight the areas that contributed to the classification. Based on these results, evaluating the model's accuracy becomes possible by discerning whether it reliably identifies relevant glitch features or bases its predictions on incorrect attributes.

Examples of Score-CAM visualizations illustrating the estimation rationale for each glitch class are presented in tables B1–B5 of appendix B. It is evident from these tables that the model accurately identifies and highlights the regions corresponding to each glitch type. Additionally, please note that axis labels have been removed from the spectrograms, hereafter. For instance, glitches appear in various positions with different time scales in some classes, such as 'Repeating_Blips'. This feature is successfully highlighted in similar areas by our model's estimation rationale, as shown in table B4. These observations suggest the model identifies and discriminates between glitches and minor background noise.

In the following, we focus on the shape and position of the estimation rationale.

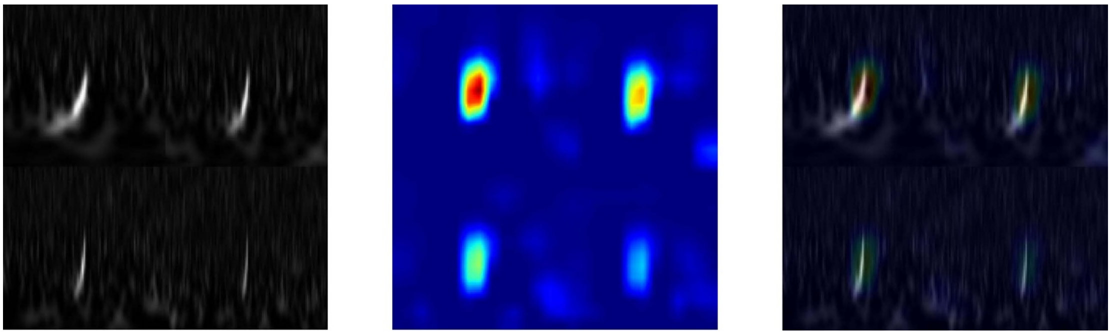


Figure 4. Estimation rationale for another high-scoring class such as 'Blip'. The figure comprises the same input image as figure 3(left); the estimation rationale heatmap is obtained from Score-CAM (centre); and the overlapping picture (right) is highlighting the region that will mainly contribute to the computation of the Softmax layer. The 'Blip' class is the second most likely candidate class determined by the model for such input. Only the best candidate is then used in the classification.

3.2.1. Shape of rationale.
The estimation rationale shapes appear to draw singular patterns in each class from tables B1 to B5.

We aim to stress that one input image can be used to generate 22 Score-CAM heatmaps highlighting the region that is most likely used by the model to classify the input map. The best candidate is then tagged accordingly and is used in the final scoring. (e.g. for the 'Chirp' class, our model is perfectly detecting the feature without confusion as highlighted by the confusion matrix figure A1). For example, figures 3 and 4 demonstrate the rationale behind the estimation of two specific classes following the input of an image of the 'Chirp' class into the model. The two classes are the 'Chirp' class and the 'Blip' class, which are the top and second highest-scoring classes, respectively. These scores were the output from Softmax layer after inputting the image into the model and passing them through fully connected layer to classify them into the 22 classes (as shown in table 2). The estimation rationale for glitched images, which are noted by their frequency swept shape and correctly classified as 'Chirp', and the second highest-scoring class, 'Blip', does not have this sweep and only shows a drop shape. By comparing the saliency map to each class in detail, it is possible to identify the parts that contribute to the identification of each class and the parts that cause confusion in the model.

Furthermore, estimation rationale heatmaps with broadly different shapes appear for 'No_Glitch', 'Unidentified_Glitch', and 'Extremely_loud'. These classes have multiple shape characteristics, suggesting that they are classified into each class despite the complexity of distinguishing among several features.

3.2.2. Position of rationale.
Based on tables B1–B5, the estimation rationale heatmap is displayed along the shape of the glitches and in all classes except for some of them, such as '1080Lines' and 'No_Glitch'. The heatmap is absent or very faint in areas without noise. In tables B1–B5, heatmaps show that the estimated rationale aligns with the shape of the glitches.

In most classes, except for '1080Lines' and 'No_Glitch', the heatmap is much weaker in areas without noise than in those with glitches. In the '1080Lines' class, the estimation rationale heatmap appears as two typical shapes at the low-to-mid-frequency ranges as line shapes and at the locations of glitches that resemble a row of dots at near 1080 Hz. This could be due to the model learning with both the glitches (near 1080 Hz) and background noise (in the low-to mid-frequency ranges) as its features, or it could be due to the superposition of the background noise in this frequency range with the row of dots, which is a characteristic shape of this class.

From this discussion, we can conclude that the model successfully identifies glitch characteristics, such as shape and position, across different classes by analysing estimation rationale heatmaps, despite the presence of misclassified samples. The next subsection details these misclassifications.

3.2.3. Misclassified samples.
Focusing on misclassified glitches, we now create Score-CAM saliency maps for both the top-scoring class in the classification and the class with the correct label.

By definition, the 'No_Glitch' class should be devoid of glitches; however there are clear instances of misclassification. This serves as an empirical benchmark to understand the overall model's decision-making process, where background noise may be mistaken for a classification rationale. Figure 5 illustrates the estimation rationale of a 'No_Glitch' sample, which the model misclassified into the 'Low_Frequency_Lines' class (top-scoring class). The 'Low_Frequency_Lines' class shows a strong rationale for the long noise section

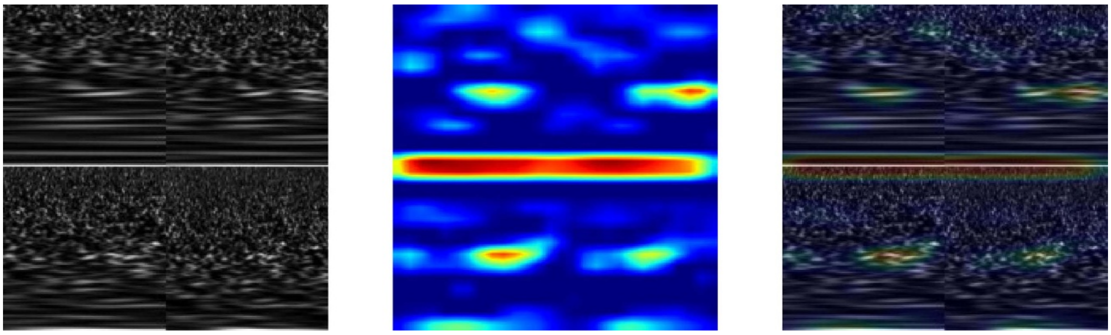


Figure 5. Input images (left), heatmap of estimation rationales (centre), and overlap (right) when 'No_Glitch' is misclassified as 'Low_Frequency_Lines'.

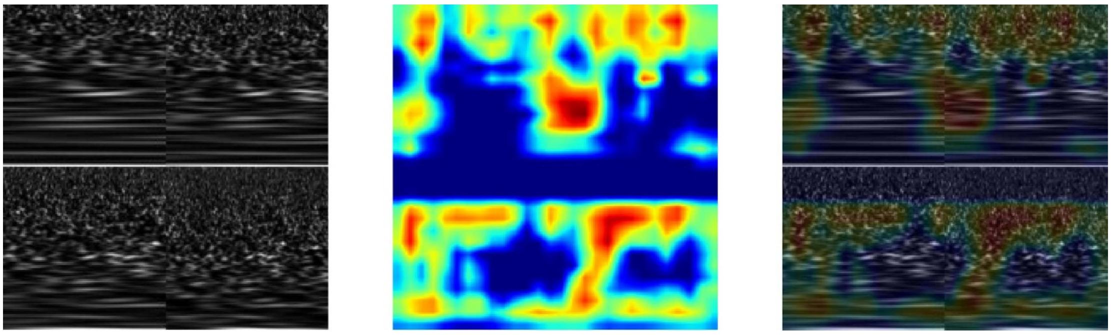


Figure 6. Input images (left), heatmap of estimation rationales (centre), and overlap (right) when 'No_Glitch' is correctly classified as 'No_Glitch'.

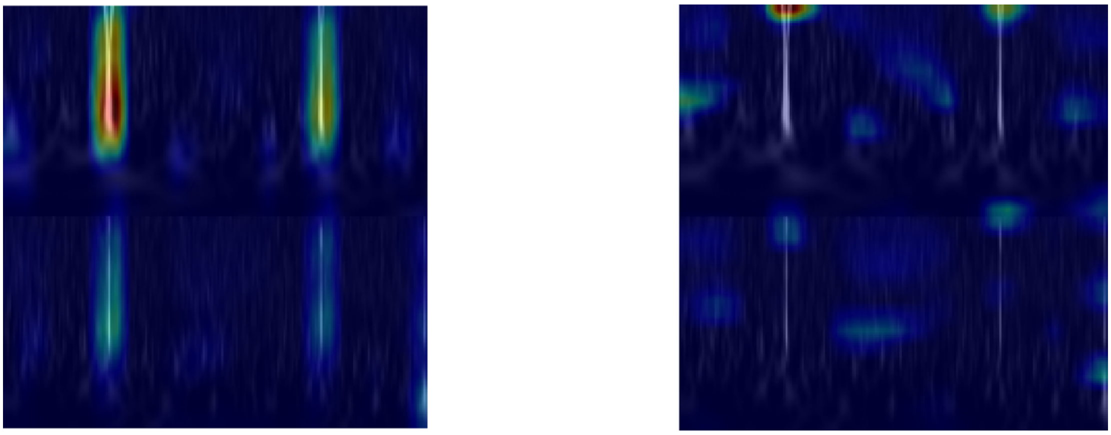


Figure 7. Left: Heatmap of the overlap image (input image and the estimation rationale) when 'Whistle' is misclassified as 'Blip'. Right: Heatmap of the overlap image (input image and the estimation rationale) when 'Whistle' is correctly classified as 'Whistle'.

in the time direction at the centre of the image, which is a typical picture for such a glitch. Figure 6 shows the rationale for the correct 'No_Glitch' class, which is widely displayed across the entire image, excluding the long noise section in the time direction. The shape of this rationale matches the typical shape of the estimation rationale when 'No_Glitch' is correctly classified as 'No_Glitch', as shown in table B3.

Furthermore, the estimation rational heatmap of 'Whistle' misclassified as 'Blip' is shown on the left side of figure 7. As the estimated rationale for classifying the 'Whistle' sample as 'Blip', a teardrop shape covering the entire vertically long glitch located in the central part of the time direction in each of the four-time scales images is displayed. This shape is typical of the 'Blip' class. In the estimation rational heatmap when 'Whistle' is correctly classified as 'Whistle' (right side of figure 7), only the bifurcated part above this vertically long glitch is displayed as the rationale, suggesting that it identifies the arch-like shape, which is a typical shape of 'Whistle', in the image as the estimation rationales.

Table 3. Duration of glitch, the difference in accuracy between the different timescales of the images used for training, and the location of the estimation rationales in the merged-view model. The results for single timescale images are from the model in [22], while the results for the merged-view are from the model in this study. Note that 'Sh' and 'L' denote short and long duration glitches, respectively.

Duration	Label/Class	Accuracy of 0.5[s] image input (%)	Accuracy of 4(s) image input (%)	Accuracy of merged-view model in this study	Position of the estimated rationale in the merged-view image
Sh	Air_Compressor	100	80.0	100	Top left and top right
Sh	Blip	98.1	97.6	99.3	Top left
Sh	Tomte	100	100	100	Topside
Sh	Power_Line	100	100	100	Topside
Sh	Helix	96.7	96.7	100	All timescales
Sh	Repeating_Blips	80.0	76.0	95.4	All timescales
L	Wandering_Line	57.1	71.4	85.7	Bottom side
L	Extremely_Loud	96.8	98.4	100.00	All timescales
L	Low_Frequency_Lines	89.8	89.8	97.1	Appears in the centre line
L	Scattered_Light	98.4	98.4	97.1	Appears in the centre line
L	Light_Modulation	83.1	86.5	98.8	Both Topside and Bottom side

3.2.4. Noise duration and estimation rationale.

Table 3 shows that the differences in classification accuracy between classes are due to the correlation between noise duration and image timescale. Models trained with a single image on a short timescale showed superior performance in classifying short-duration glitches, whereas those trained with a single image on a long timescale showed high performance in classifying long-duration glitches (L).

We compared the position of the estimation rationales in the merged-view model with the optimal timescales, as shown in [22]. The results are summarised in table 3. For glitches of short durations (Sh), such as 'Air_Compressor' and 'Wandering_Line', and long durations (L), where there is a considerable difference in accuracy due to the timescale of the input image, it is observed that in the merged-view image (figure 1), the estimation rationale appears in areas where the input images with high-accuracy timescales are positioned. Despite the merged image, the estimated rationales provide strong evidence towards the side with the shorter timescale, which becomes less prominent as the timescale increases. This observation is likely be due to the model's ability to detect glitch shapes in images with shorter timescales accurately.

The merged-view images of 'Repeating_Blips' and 'Helix' show that the estimation rationales appear across all timescale regions with no specific tendency in their positions. Given that glitch noise often manifests in cluster shapes, the model will likely classify based on these repetitive segments. This may be the case for images taken at various timescales. In the case of 'Extremely_Loud' for long durations (L), estimation rationales appeared throughout the merged-view images. This may be attributed to the characteristics of large noise covering a wide range of time and frequency, suggesting that the model does not prioritise classification based on specific timescale regions. Similarly, in 'Light_Modulation', the estimation rationales in the merged-view images were evenly distributed throughout the image, without any preference towards areas at various timescales. Two main estimation rationales were observed: a slender cone on the high-frequency side and a spreading shape towards the low-frequency side.

Additionally, when there is no significant difference in accuracy due to timescale variance, in classes such as 'Low_Frequency_Lines' and 'Scattered_Light', the estimation rationales tend to appear in regions with smaller timescales in the merged-view images.

3.2.5. Feature visualisation using UMAP.

To further enhance the explainability of the CNN model, we used UMAP to visualise the features extracted from the final convolutional layer of our classification model and display them in a 3D representation space, utilising data from the training, validation, and test sets. We developed a tool to visualise the latent variables of all 8583 glitches in a 3D space, as shown in figure 8, along with the corresponding Score-CAM images and Gravity Spy labels. We used Plotly and Dash [12], a Python web application framework to implement this tool. This framework allows plots to be displayed in a web browser, enabling immediate interaction through mouse controls, such as zooming and rotating. For instance, users can zoom in to closely examine a cluster of glitches or rotate the 3D plot to view it from different angles without having to replot when adjusting the display. This interactive functionality enhances user experience and contributes to improved analysis efficiency.

[12] https://dash.plotly.com/.

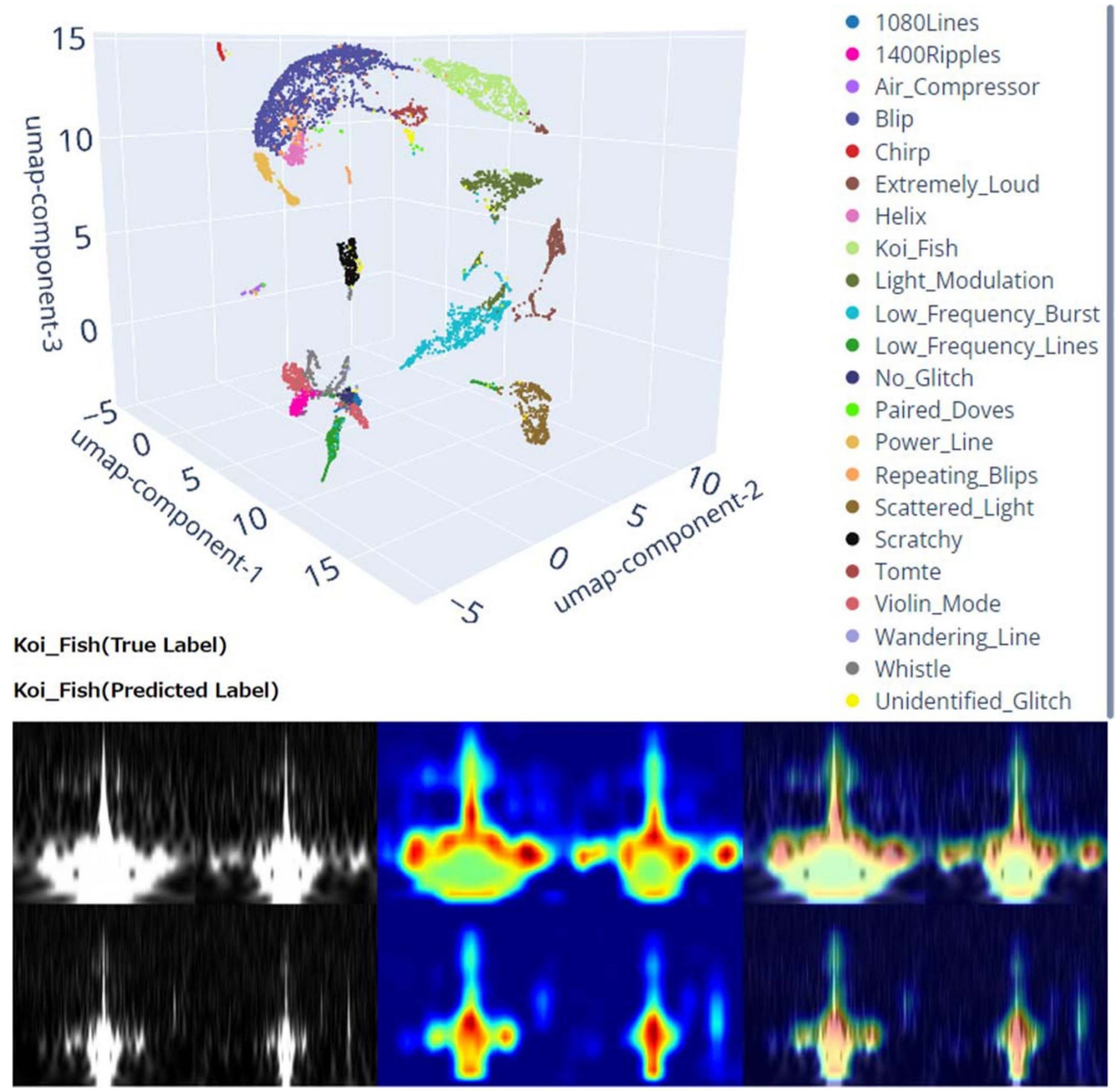


Figure 8. The features are compressed into 3D space using UMAP with the last convolution layer of the CNN model as input. This space embeds a total of 8583 glitch features. The colours indicate the 22 different labels in the Gravity Spy dataset. (Top) For the latent variable selected with the mouse, the Gravity Spy label name and its position on the UMAP are displayed in the frame. (Bottom) The input image (Bottom left), Score-CAM (Bottom centre), and overlap (Bottom right) are displayed for the mouse-hovered feature. The Gravity Spy label for the input image is displayed as True Label, and the label estimated by the CNN is displayed as Predicted Label.

Figure 8 shows that the entire set of 8583 Gravity Spy classes is separated in the feature space with sufficiently large intervals, except for a few classes. The clusters for 'Violin_Mode', '1080Lines', and 'Whistle' are closely positioned, as are the clusters for 'Violin_Mode', '1400Ripples', and 'Whistle'. Additionally, 'Wandering_Line' is distributed across the entire clusters of the three classes: 'Violin_Mode', '1080Lines', and '1400Ripples'. Furthermore, the 'Repeating_Blips' class is neighbouring to the 'Blip' and 'Helix' clusters. Excluding these groups of classes, the distribution of features in the training, validation, and test datasets was similar across each class, indicating that the model in this study was generalised well. The distributed features were extracted from the second-to-last layers of the model, and the final layer was a linear combination layer for classification into 22 classes. Therefore, the separation of the features in the latent space accurately reflects of the final classification results.

By comparing the shape of the estimation basis with its position on UMAP, we confirmed that the shape of the estimation rationale corresponds to the clustering on the UMAP. Furthermore, as shown in figure 9, classes such as 'Extremely_Loud' with multiple shapes of the estimation rationales within the same class belong to different clusters on UMAP. This indicates the possibility of further subdivision into more classes in the future. Additionally, in classes that show a wide distribution on UMAP, such as the 'Blip' class, we observed the presence of shapes that closely resemble the estimation rationales of neighbouring classes within this distribution.

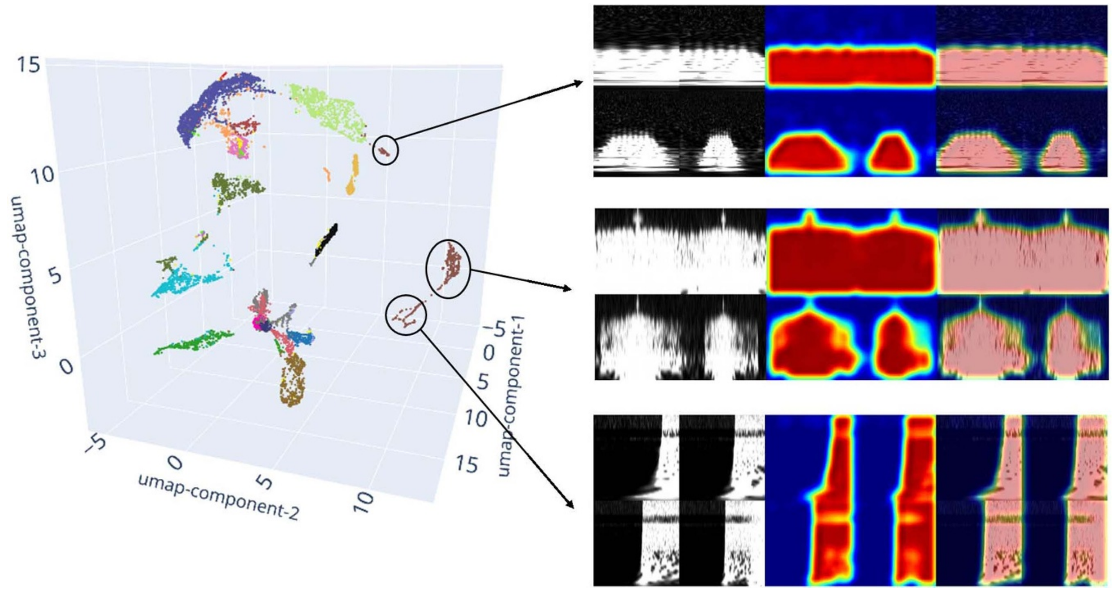


Figure 9. Distribution of the 'Extremely_Loud' class divided into multiple clusters in feature space and the shape of the representative estimation rationale for each cluster. In the feature space, similar features are extracted for each class and distributed nearby. The shape of the estimation rationale is similar for each class. In this example, both the feature space and the shape of the estimation rationale are divided into multiple clusters, suggesting the possibility of subclasses.

4. Summary

The classification of glitches in gravitational wave detector data can enhance our understanding of the origins of noise and improve detector sensitivity and performance. Our study developed a weakly supervised CNN model based on the Bahaadini's architecture [22]. After training our CNN model, we visualised the estimation rationale for glitch classification using Score-CAM. Based on the visualisation of the estimation rationale for classification, we examined whether the model was classified based on appropriate learning and why misclassifications occurred. We confirmed that similar estimation rationales exist for each cluster by displaying the features extracted by the model via UMAP algorithm.

LIGO O1 Gravity Spy dataset was used to train our classification model and visualise the rationale behind our estimations. Our future research will expand to include datasets from the O2 and O3 observation runs. We aim to use these additional datasets to visualise our estimates' rationale further. Moreover, we intend to conduct a detailed assessment of specific glitches associated with the KAGRA environment and extend the scope of our analysis.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgment

This research was supported in part by the Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (No. 22H01228 (K Somiya) and No. 20K19756 (Y Terada) and Nos. 19H01901, 23H01176, 23K25872 and 23H04520 (H Takahashi)) and Grant-in-Aid for JSPS Fellows (No. P21726 (M Meyer-Conde)). This research was supported by the Joint Research Program of the Institute for Cosmic Ray Research, University of Tokyo, and Tokyo City University Prioritised Studies.

Appendix A. Confusion matrix

The generalisation performance of the model described in section 2.2 was evaluated using the test dataset. The confusion matrix is shown in figure A1.

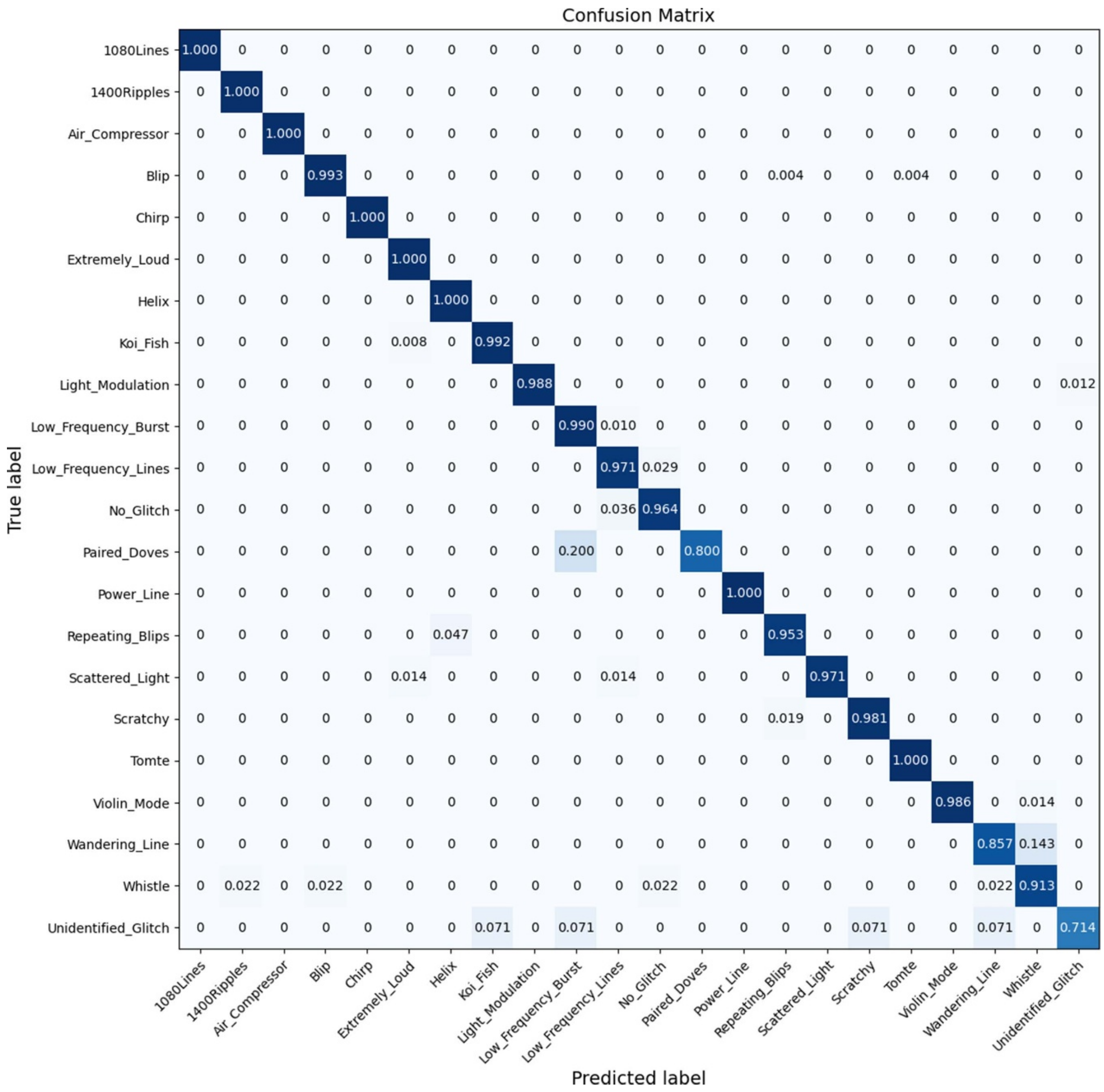


Figure A1. Confusion matrix for the 22 glitch classes. The *x* and *y* axes represent the predicted and true classes, respectively. Normalisation is by the total number of glitches in each class.

Appendix B. Saliency map

The examples of saliency maps are a Score-CAM visualisation of estimation rationale for each glitch class as shown in tables B1–B5.

Table B1. Input images for each representative class, estimation rationale, and their visual characteristics (1).

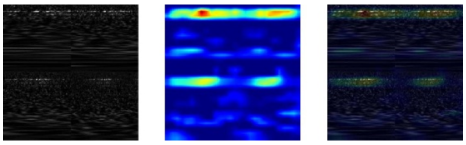
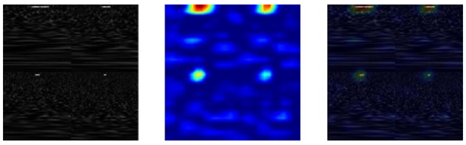
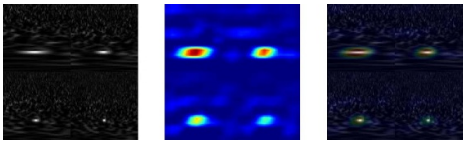
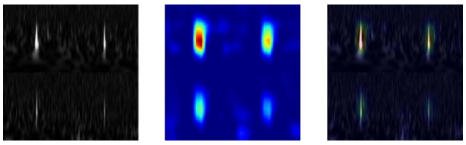
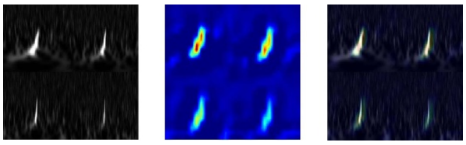
Label/Class	Input image and the image of estimation rationale	Visual features
1080Lines		As the name implies, the estimation rationale appears strongly around 1080 Hz.
1400Ripples		The estimation rationale appears for a short period of time existing at 1400 Hz.
Air_Compressor		The estimation rationale appears as a thick line around 50 Hz. Strong estimation rationale appears in the image areas with short timescales and tends to become weaker in the image areas with long timescales.
Blip		The estimation rationale appears in a narrower drop shape.
Chirp		The estimation rationale appears as the frequency sweeps with time.

Table B2. Input images for each representative class, estimation rationale and their visual characteristics (2).

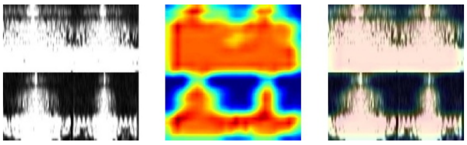
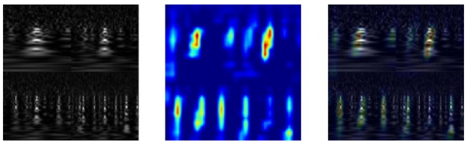
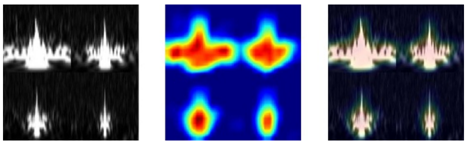
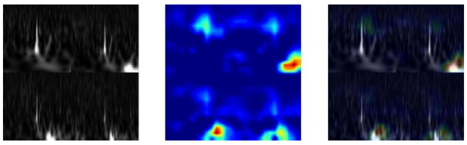
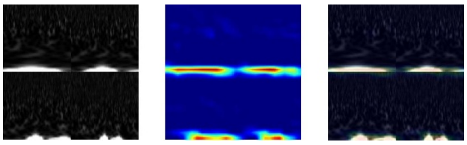
Label/Class	Input image and the image of estimation rationale	Visual features
Extremely_Loud		Estimation rationale appears in large areas that occupy the majority of the image.
Helix		Estimation rationale also appears in the noise areas generated by vortex-like shapes and groups that occur in the mid-frequency domain.
Koi_Fish		In some cases, estimation rationale occurs in a large region similar to Extremely Loud. In other cases, estimation rationale occurs only in the pectoral fin near 30 Hz and in the tail fin area near 500 Hz.
Light_Modulation		Stronger estimation rationale appears in the low-frequency domain, and weaker estimation rationale tends to appear in the sharp spike-like parts of the noise.
Low_Frequency_Burst		There is almost no estimation rationale in the high-frequency domain, and estimation rationale appears in the low-frequency domain with a small knob or their connected shapes.

Table B3. Input images for each representative class, estimation rationale, and their visual characteristics (3).

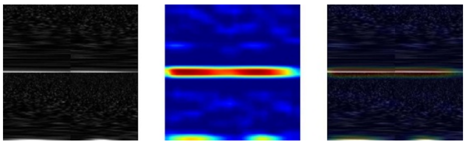
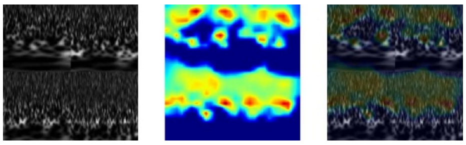
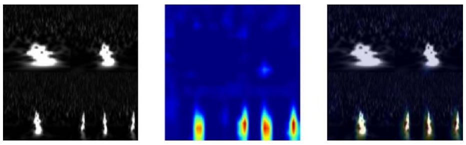
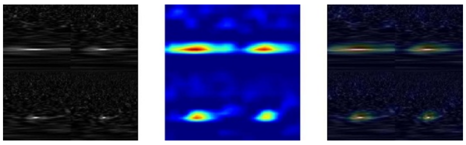
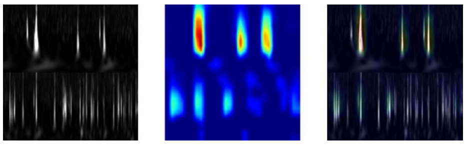
Label/Class	Input image and the image of estimation rationale	Visual features
Low_Frequency_Lines		As the name implies, a strong estimation rationale appears in a straight line on the low frequency.
No_Glitch		Weak estimation rationale tends to appear on the whole image.
Paired_Doves		Hump-like shapes are repeated as estimation rationale in images with long time scales.
Power_Line		A thin estimation rationale appears for long durations at around 60 Hz. The estimation rationale also appears weakly in the harmonics of 60 Hz.
Repeating_Blips		The estimation rationale repeatedly appears in the same narrow drop shape, with varying intensity for each sample image.

Table B4. Input images for each representative class, estimation rationale, and their visual characteristics (4).

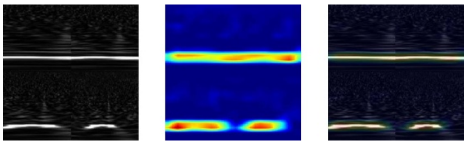
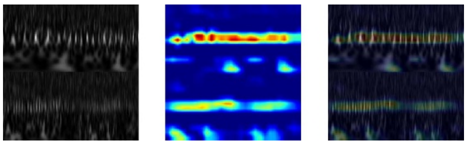
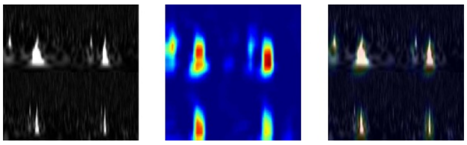
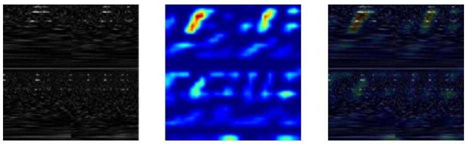
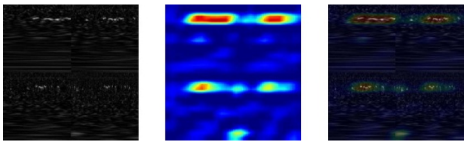
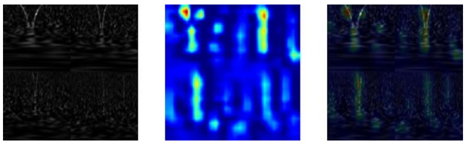
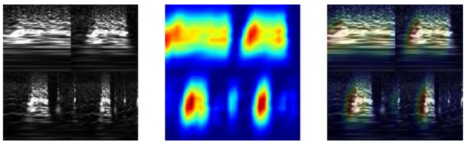
Label/Class	Input image and the image of estimation rationale	Visual features
Scattered_Light		Estimation rationale appears in the low-frequency domain and for long time scales. Not only straight lines but also curved shapes may appear.
Scratchy		Two lines of estimation rationale appear at the top and bottom of the image.
Tomte		Similar shape to the estimation rationale for 'Blip', the estimation rationale often appears weakly in image areas with a time scale of 4s.
Violin_Mode		It appears as a repetition of similarly shaped estimation rationale.
Wandering_Line		The shape along the frequency meander appears as an estimation rationale.

Table B5. Input images for each representative class, estimation rationale, and their visual characteristics (5).

Label/Class	Input image and the image of estimation rationale	Visual features
Whistle		The estimation rationale appears as the shape of the arches and valleys.
Unidentified_Glitch (None_of_the_Above)		Many estimation rationales appear for this class, which correspond to remaining unclassified classes.

ORCID iDs

Naoki Koyama https://orcid.org/0009-0008-2194-3741
Yusuke Sakai https://orcid.org/0000-0001-8810-4813
Seiya Sasaoka https://orcid.org/0000-0002-2155-8092
Diego Dominguez https://orcid.org/0009-0007-7550-1933
Kentaro Somiya https://orcid.org/0000-0003-2601-2264
Yuto Omae https://orcid.org/0000-0002-5924-6959
Yoshikazu Terada https://orcid.org/0000-0002-4509-1108
Marco Meyer-Conde https://orcid.org/0000-0003-2230-6310
Hirotaka Takahashi https://orcid.org/0000-0003-0596-4397

References

[1] Brady P, Losurdo G and Shinkai H 2020 *Ligo, Virgo and Kagra as the International Gravitational Wave Network* (Springer) pp 1–21
[2] Aasi J *et al* (LIGO Scientific Collaboration) 2015 *Class. Quantum Grav.* **32** 074001
[3] Acernese F *et al* 2014 *Class. Quantum Grav.* **32** 024001
[4] Akutsu T *et al* 2019 *Nat. Astron.* **3** 35–40
[5] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2016 *Phys. Rev. Lett.* **116** 061102
[6] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2019 *Phys. Rev.* X **9** 031040
[7] Abbott R *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2021 *Phys. Rev.* X **11** 021053
[8] Abbott R *et al* 2024 (The LIGO Scientific Collaboration and the Virgo Collaboration) *Phys. Rev.* D **109** 022001
[9] Abbott R *et al* (LIGO Scientific Collaboration, Virgo Collaboration and KAGRA Collaboration) 2023 *Phys. Rev.* X **13** 041039
[10] Cuoco E *et al* 2020 *Mach. Learn.: Sci. Technol.* **2** 011002
[11] Zhao T, Shi R, Zhou Y, Cao Z and Ren Z 2023 Dawning of a new era in gravitational wave data analysis: unveiling cosmic mysteries via artificial intelligence – a systematic review (arXiv:2311.15585)
[12] Zevin M *et al* 2017 *Class. Quantum Grav.* **34** 064003
[13] Bahaadini S, Noroozi V, Rohani N, Coughlin S, Zevin M, Smith J, Kalogera V and Katsaggelos A 2018 *Inf. Sci.* **444** 172–86
[14] Zevin M *et al* 2024 *Eur. Phys. J. Plus* **139** 100
[15] Barredo Arrieta A *et al* 2020 *Inf. Fusion* **58** 82–115
[16] Zhou B, Khosla A, Lapedriza A, Oliva A and Torralba A 2016 Learning deep features for discriminative localization *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (IEEE) pp 2921–9
[17] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D and Batra D 2017 Grad-CAM: visual explanations from deep networks via gradient-based localization *2017 IEEE Int. Conf. on Computer Vision (ICCV)* (IEEE) pp 618–26
[18] Wang H, Wang Z, Du M, Yang F, Zhang Z, Ding S, Mardziel P and Hu X 2020 Score-CAM: score-weighted visual explanations for convolutional neural networks *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops* (IEEE) pp 24–25
[19] McInnes L, Healy J and Melville J 2018 arXiv:1802.03426
[20] Robinet F, Arnaud N, Leroy N, Lundgren A, Macleod D and McIver J 2020 *SoftwareX* **12** 100620
[21] Chatterji S, Blackburn L, Martin G and Katsavounidis E 2004 *Class. Quantum Grav.* **21** S1809
[22] Bahaadini S, Rohani N, Coughlin S, Zevin M, Kalogera V and Katsaggelos A K 2017 Deep multi-view models for glitch classification *2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 2931–5
[23] George D, Shen H and Huerta E A 2018 *Phys. Rev.* D **97** 101501
[24] Coughlin S *et al* 2019 *Phys. Rev.* D **99** 082002
[25] Colgan R E, Corley K R, Lau Y, Bartos I, Wright J N, Márka Z and Márka S 2020 *Phys. Rev.* D **101** 102003
[26] Soni S *et al* 2021 *Class. Quantum Grav.* **38** 195016
[27] Sakai Y *et al* 2022 *Ann. Phys., Lpz.* 536 2200140
[28] Sakai Y *et al* 2022 *Sci. Rep.* **12** 9935
[29] Glanzer J *et al* 2023 *Class. Quantum Grav.* **40** 065004
[30] Fernandes T, Vieira S, Onofre A, Bustillo J C, Torres-Forné A and Font J A 2023 *Class. Quantum Grav.* **40** 195018

[31] Lin Y C and Kong A K H 2024 Extract non-Gaussian features in gravitational wave observation data using self-supervised learning (arXiv:2403.04350)

[32] Glanzer J *et al* 2021 Gravity Spy machine learning classifications of LIGO glitches from observing runs O1, O2, O3a, and O3b *Zenodo* https://doi.org/10.5281/zenodo.5649212

[33] Zeiler M D 2012 ADADELTA: an adaptive learning rate method (arXiv:1212.5701)

[34] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems 32* (Curran Associates) pp 8024–35

[35] Roweis S T and Saul L K 2000 *Science* **290** 2323–6

[36] van der Maaten L and Hinton G 2008 *J. Mach. Learn. Res.* **9** 2579–605