

A CLUE TO CROSS-PERIOD GLITCH MORPHOLOGY IN GRAVITATIONAL-WAVE DETECTORS VIA TRANSFER LEARNING

YUSUKE SAKAI^{1,2}, NAOKI KOYAMA³, MARCO MEYER-CONDE^{1,2}, SEIYA SASAOKA⁴
DIEGO DOMINGUEZ⁵, KENTARO SOMIYA⁵, YUTO OMAE⁶, YOSHIKAZU TERADA⁷,
AND HIROTAKA TAKAHASHI^{1,2}

¹Research Center for Space Science, Advanced Research Laboratories
Tokyo City University

1-28-1 Tamazutsumi, Setagaya-ku, Tokyo 158-8557, Japan
{ysakai; marco; hirotaka}@tcu.ac.jp

²Graduate School of Information and Data Science, and Department of Design and Data Science
Tokyo City University

3-3-1 Ushikubo-Nishi, Tsuzuki-ku, Yokohama, Kanagawa 224-8551, Japan

³Graduate School of Science and Technology, Niigata University
8050 Ikarashi-2-no-cho, Nishi-ku, Niigata City, Niigata 950-2181, Japan

⁴Department of Physics, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8551, Japan

⁵Department of Physics, School of Science, Institute of Science Tokyo
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8551, Japan
diego@gw.phys.sci.isct.ac.jp; somiya@phys.sci.isct.ac.jp

⁶Artificial Intelligence Research Center, College of Industrial Technology
Nihon University
1-2-1 Izumi-cho, Narashino, Chiba 275-8575, Japan
oomae.yuuto@nihon-u.ac.jp

⁷Graduate School of Engineering Science, The University of Osaka
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
terada.yoshikazu.es@osaka-u.ac.jp

ABSTRACT. *The international gravitational wave observation network detects gravitational waves from compact binary mergers. However, the detectors are frequently affected by nonstationary and non-Gaussian noise artifacts called “glitches.” Glitches reduce detector sensitivity and complicate the identification of true gravitational wave signals. As these detectors undergo upgrades to enhance sensitivity, the resulting glitches may exhibit new or altered morphologies. Identifying these variations may provide insight into the environmental or instrumental origin of the glitch. This study proposed glitch classification in gravitational wave interferometers during different observation periods via transfer learning. Specifically, we utilized a pretrained model with Observation 1 & Observation 2 glitches and retrained it with new Observation 3a glitches. This approach enhanced the generalization and adaptability of the model for new observation runs and reduced the training costs. To ensure interpretability, we employed feature visualization methods using Score-weighted class activation mapping to explain the classification mechanism for glitches and uniform manifold approximation and projection to visualize their distribution. We demonstrated that continuous utilization of the model during observation runs resulted in both efficient learning and interpretability, and further indicated the tracking of the evolving nature of glitch morphologies over periods. The proposed approach will help improve future frameworks for glitch analysis and long-term strategies in gravitational wave data processing.*

Keywords: gravitational wave data analysis, transfer learning, noise classification, convolutional neural networks, Score-CAM, UMAP

1 Introduction The international gravitational wave observation network [1] comprising LIGO, Virgo, and KAGRA has detected more than 200 gravitational wave events from the mergers of binary black holes or neutron stars. A simultaneous observation of gravitational and electromagnetic waves was achieved in the neutron star merger event [2]. The following observation runs were performed during specific periods: Observation 1 (O1: September 2015 – January 2016), Observation 2 (O2: November 2016 – August 2017), and Observation 3a (O3a: April 2019 – November 2019). Gravitational wave observations are expected to contribute to further advancements in astronomy and physics.

Gravitational waves were observed as a perturbation of the space-time metric, and their distortion was insignificant. State-of-the-art instrumentation was installed in the detector to detect such a weak signal. However, nonstationary and non-Gaussian noise, known as “glitch” noise, often happens during observations. These glitches possess various morphological features originating from both instrumental and environmental factors, including ground vibrations, pendulum control signals, and laser fluctuations. Glitches affect the detector sensitivity, continuous operation, and quality of gravitational wave signals.

The Gravity Spy project analyzed various glitches in LIGO detectors during O1&O2 observation runs and identified 22 types of glitches [3, 4]. Two new glitches were identified and classified using convolutional neural networks (CNNs) during O3a [5]. Subsequent studies have demonstrated that transfer learning is effective for glitch classification by employing models pretrained on ImageNet [6] or on previous observation runs [7]. Because gravitational wave detectors have been upgraded continuously to improve their sensitivity, different observation periods may generate glitches with distinct morphological features. However, for glitch types that did not change significantly in morphology during the observation periods, a transfer learning approach would suffice for classification. The reuse of network weights obtained from the training of previous observations may be effective for future glitch classification tasks, both in terms of computational cost and classification performance.

In this study, we investigated the effectiveness of transfer learning for glitch datasets with different morphological features, specifically between the O1&O2 and O3a datasets. We used CNNs, whose architectures are based on previous studies on glitch classification [3, 5], and applied a model pretrained on the O1&O2 dataset and transfer learning to the O3a dataset. Additionally, we incorporated Score-weighted class activation mapping (Score-CAM) [8] to explain the classification and uniform manifold approximation and projection (UMAP) [9] for the visualization of glitch clustering. This made the analysis more accessible and interpretable. We evaluated the classification performance on the O3a dataset using the transfer learning model, and further investigated glitch morphologies based on their UMAP clustering during the O1&O2 and O3a observation periods.

The remainder of this paper is organized as follows: In Sec. 2, the dataset and analysis method employed in this study are introduced & briefly reviewed. The dataset preprocessing method is elucidated in this section. The results and discussion are presented in Sec. 3. Finally, Sec. 4 provides a summary of the study.

2 Method In this section, we provide an overview of the glitch dataset and its preprocessing, including the analysis framework, which includes CNNs, Score-CAM, and UMAP.

2.1 Dataset The Gravity Spy project [3] provides a time–frequency spectrogram image of glitches for O1&O2. Glitch images for O3a [5] were also provided. Glitch images were recorded with four different time-windows: 0.5, 1.0, 2.0, and 4.0 s, typically focusing the frequency range from 8 to 2048 Hz. We selected O3a glitch images evaluated with a machine learning confidence of 0.9 or higher. This threshold value was equivalent to that employed in [10]. The dataset distribution of each label is listed in Table 1. Notably,

TABLE 1. Dataset distributions for each class.

Label	O1&O2 dataset amount	O3a dataset amount
1080Lines	328	20
1400Ripples	232	261
Air_Compressor	58	116
Blip	1869	3737
Chirp	66	14
Extremely_Loud	454	908
Helix	279	33
Koi_Fish	830	1660
Light_Modulation	573	111
Low_Frequency_Burst	657	1314
Low_Frequency_Lines	453	906
No_Glitch	181	362
None_of_the_Above	88	24
Paired_Doves	27	41
Power_Line	453	572
Repeating_Blips	285	478
Scattered_Light	459	1218
Scratchy	354	490
Tomte	116	832
Violin_Mode	472	256
Wandering_Line	44	32
Whistle	305	610
Blip_Low_Frequency	0	1260
Fast_Scattering	0	627
total amount	8583	15882

although the dataset exhibits an imbalanced distribution across labels, the degree of imbalance is comparable to that used in datasets from previous studies [11, 12]. The O1&O2 dataset contains 22 classes, whereas the O3a dataset contains 24 classes, including new glitches such as “Blip_low_frequency” and “fast_scattering.”

Our preprocessed dataset is based on a “merged-view” image from previous studies [13, 14], which is constructed by joining four time-window images into a 2×2 grid as illustrated in Figure 1. This preprocessing was applied to both O1&O2 and O3a datasets. Transfer learning was applied to fine-tune the pretrained model on the O3a dataset after training on the O1&O2 dataset. We investigated the generalization capability of the proposed model for glitch classification in different observational runs. The ratios of the training, validation, and test datasets were 70%, 15%, and 15%, respectively.

2.2 Architecture Figure 1 shows the proposed CNN architecture, which is typically selected for glitch classification, including [12, 13]. The proposed architecture comprises convolutional layers with rectified linear unit (ReLU) activation functions and max-pooling layers for feature learning, followed by a fully connected layer using features as inputs to classify glitches. The model was configured to classify glitches into 22 classes during training on the O1&O2 dataset. Subsequently, transfer learning was applied to adapt the pretrained model to the O3a dataset, which comprised 24 classes, by replacing its fully connected layer (initially outputting 22 classes) with a new layer that outputs 24 classes.

The training was conducted under the following conditions: The cross-entropy loss of a loss function. A learning rate of 0.1 with the Adadelata [15] optimizer, batch size of 60, and 20 epochs. The computational environment was the PyTorch library [16] with three graphics processing units, including NVIDIA GeForce RTX3060 and RTX3090.

2.3 Score-CAM Class activation mapping (CAM) visually explains the features of a CNN and highlights the regions of the input image that make the most significant

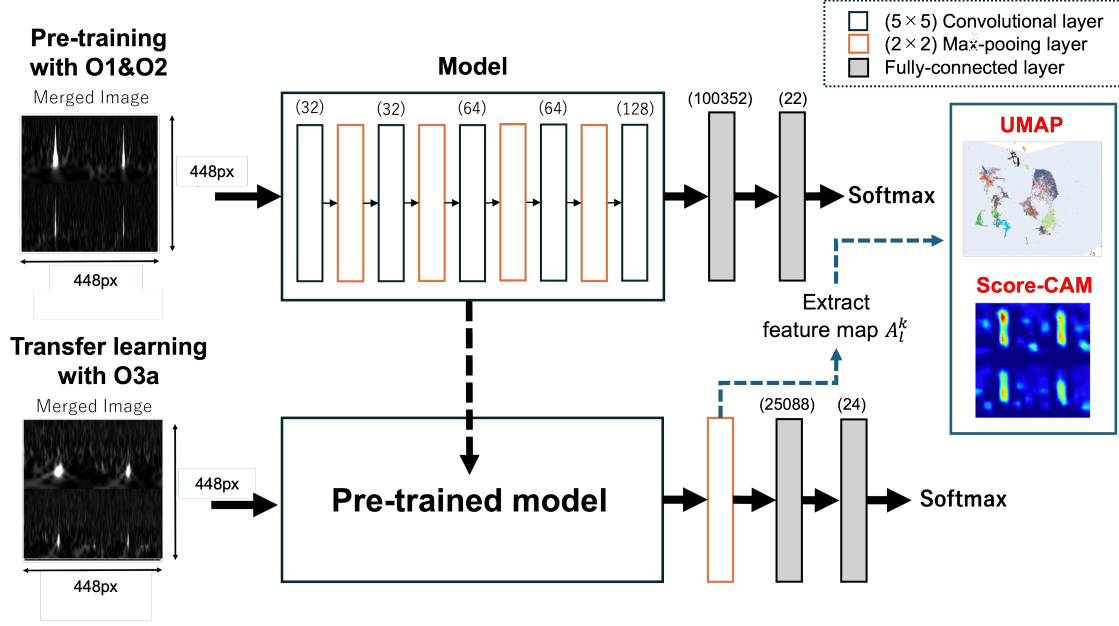


FIGURE 1. Proposed architecture overview. $(x \times x)$ is the kernel size of a convolutional layer and a max-pooling layer, and (y) denotes the output feature size. After training the O1&O2 dataset, transfer learning was applied to the pretrained model using the O3a dataset. The Score-CAM and UMAP are applied to the last max-pooling layer of the model.

contributions to the predictions of the model. Grad-CAM[17], which is a notable CAM method, may not provide an appropriate visualization when a vanishing gradient problem emerges. Score-weighted CAM (Score-CAM) [8] addresses this issue by avoiding the use of gradients for computation. The Score-CAM algorithm comprises the following steps:

1. **Feature Map Extraction** Let l denote the index of the convolutional layer, and k denote the index of a feature map within this layer. The outputs of the l -th convolutional layer of the network are obtained as a set of feature maps, represented as $\{A_k^l\}$, where A_k^l denotes the k -th feature map in the l -th layer.
2. **Upsampling and Normalization of Feature Maps** Each feature map A_k^l is upsampled to match the size of the input image and normalized to the range of 0–1. This normalized feature map is denoted as H_k^l .
3. **Calculation of Feature Map Importance** A masked image $X \circ H_k^l$ is constructed by applying the Hadamard product of X and the normalized feature map H_k^l for a given input image X . The output score $f(X \circ H_k^l)$ of the network for the masked image is computed. Additionally, a baseline image X_b , which typically represents a reference input (such as an all-zero image), is used to calculate its corresponding score $f(X_b)$. The importance of the k -th feature map is then defined as the difference between these scores:

$$C(A_k^l) = f(X \circ H_k^l) - f(X_b),$$

where $f(\cdot)$ represents the output function of the network.

4. **Generation of the Score-CAM Map** For target class c , Score-CAM map $L_{\text{Score-CAM}}^c$ is generated as a weighted sum of feature maps A_k^l , where weight α_k^c corresponds to importance score $C(A_k^l)$. The map is computed as follows:

$$L_{\text{Score-CAM}}^c = \text{ReLU} \left(\sum_k \alpha_k^c A_k^l \right),$$

where $\alpha_k^c = C(A_k^l)$ and $\text{ReLU}(\cdot)$ represents the ReLU function, ensuring non-negative values in the output.

2.4 Dimensionality Reduction and Feature Visualization using UMAP Dimension reduction methods are typically applied for feature extraction and visualization. Principal component analysis is a linear dimensionality reduction method that excels when the input data have a linear structure. UMAP [9] is a nonlinear dimensionality reduction method that is effective when the input data have a more complex structure. Compared with t -SNE [18], which is another nonlinear dimensionality reduction method, the computational cost of UMAP is less than that of t -SNE.

The UMAP algorithm computes an undirected weighted k -nearest neighbor graph based on input data, typically using the Euclidean distance metric. Subsequently, the symmetric weighted adjacency matrix $W = (w_{ij}(k))$ is computed from the graph, where the indices of i and j correspond to the i -th and j -th data points in the dataset, respectively. Let $v_{ij}(\delta)$ denote the distance in the embedded space of the i -th and j -th data points, where δ represents a tuning parameter related to cluster compactness. A larger value of δ results in an expansive cluster structure in the resulting embedding. In UMAP, when parameters k and δ are determined, the embedding is obtained by minimizing the following fuzzy cross-entropy:

$$C_{\text{UMAP}}(k, \delta) = \sum_{i \neq j} \left\{ w_{ij}(k) \log \left(\frac{w_{ij}(k)}{v_{ij}(\delta)} \right) + (1 - w_{ij}(k)) \log \left(\frac{1 - w_{ij}(k)}{1 - v_{ij}(\delta)} \right) \right\}. \quad (1)$$

For further details, see [9]. In this study, we set $k = 7$ as the number of neighbors and $\delta = 0.02$.

3 Results and Discussion We present the classification results of the O3a dataset using a pretrained model of the O1&O2 dataset. We compared the saliency maps generated by Score-CAM to investigate how the model recognized the O3a glitch. Furthermore, we examined the feature space distribution of the O1&O2 and O3a datasets using UMAP.

3.1 Glitch Classification The training curve converged at epoch 20 during the O1&O2 dataset training, and its accuracy was 98.1% using the test dataset, which is comparable to the 96.9% accuracy reported in [13] using the O1&O2 dataset. Therefore, the model at epoch 20 was selected as the pretrained model. The O3a dataset training using the pretrained model converged at approximately 8 epochs. By contrast, the model trained from scratch converged at approximately the 18th epoch with an accuracy of 96.0% using test dataset. Such an early convergence indicates the benefit of reduced training costs when employing transfer learning. A model at 8 epochs was selected for evaluation testing, and its confusion matrix obtained using the O3a test dataset is shown in Fig. 2; its accuracy was 97.6%.

Although 16 out of 24 classes achieved recalls exceeding 95%, low recall classes were observed, such as “Chirp,” “Paired_Doves,” and “Wandering_Line,” likely owing to their limited number in the training dataset. As the outcomes of Score-CAM and UMAP affect the classification results of the model, we will discuss these analyses only for classes with a recall exceeding 80%.

3.2 Saliency Map for New O3a Glitches The Score-CAM highlights the regions within an image that contribute to its classification. This provides a visualization of the features that affect the decisions of a model. We focused on the new O3a glitches to investigate the process of their classification by the model in transfer learning as follows:

- “Blip_Low_Frequency” is characterized by a lower maximum frequency and is morphologically similar to the existing “Blip” glitch. Fig. 3 (A) shows the representative

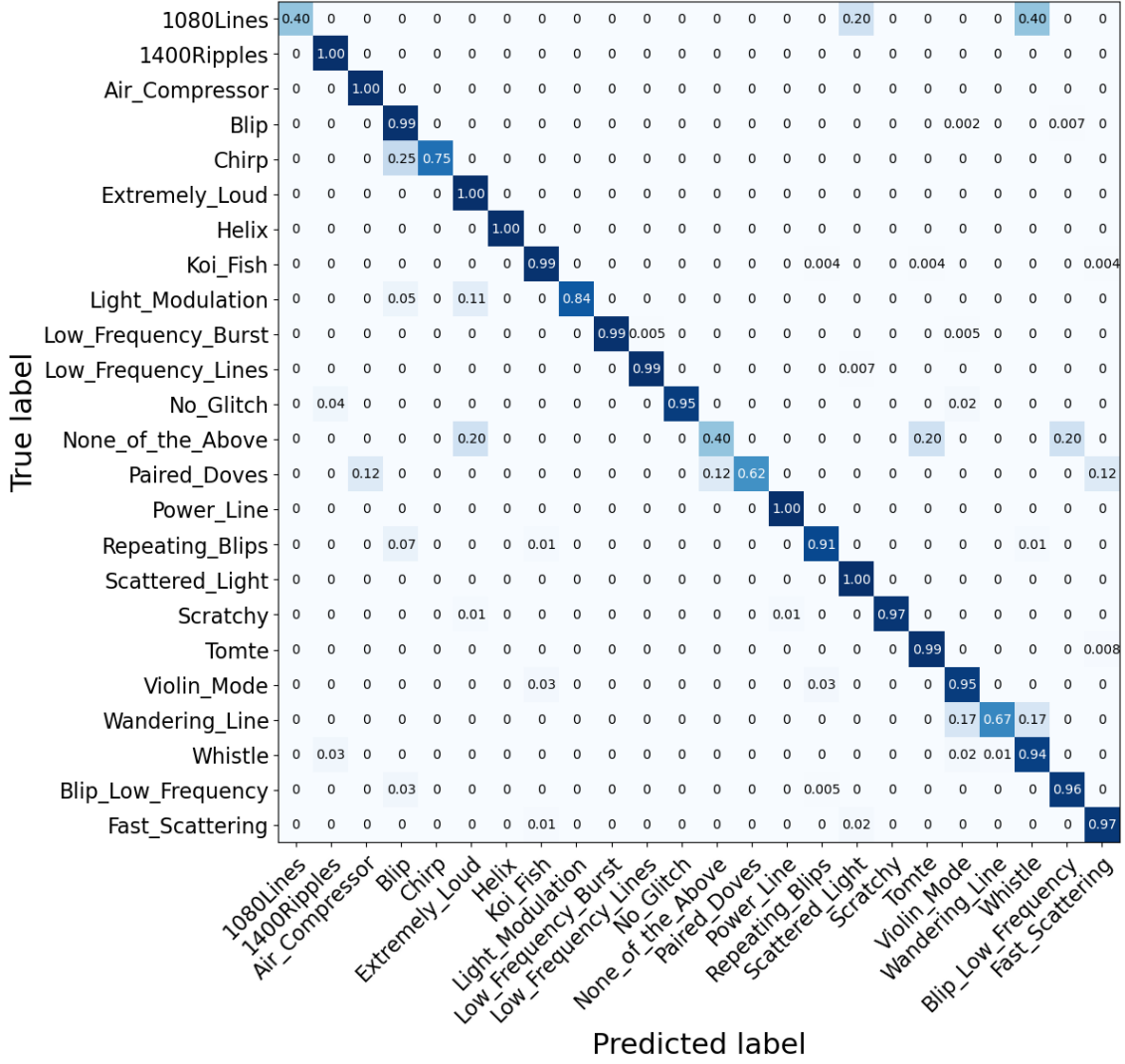


FIGURE 2. Confusion matrix using the O3a test dataset. The accuracy is 97.6%. Each row of the confusion matrix is normalized.

saliency map of “Blip_Low_Frequency,” and Fig. 3 (B) shows that of “Blip” for comparison. These maps show that “Blip_Low_Frequency” appears as a more rounded and larger structure than “Blip;” this suggests that the model accurately captures the features of “Blip_Low_Frequency.”

- “Fast_Scattering” is characterized by a shorter duration, typically from 0.25 to 0.30 s, compared with those of traditional long-duration glitches, “Scattered_Light.” In the saliency maps shown in Fig. 4 (A) and (B), “Fast_Scattering” is visualized as a band-like structure, whereas that of “Scattered_Light” appears more as a line, accurately reflecting the morphological features of this glitch.

3.3 Glitch Clustering using UMAP The final layer of the proposed transfer learning model, which was used for saliency map generation in Sec. 3.2, has a high-dimensional tensor. We can visualize the clustering of the glitches by embedding such a tensor into a three-dimensional (3D) space using UMAP.

3.3.1 Highlighting by Classification Labels Figure. 5 depicts the visualization of the O1&O2 and O3a glitches using UMAP. This plot includes the glitch clustering and its saliency map. An input to Score-CAM and UMAP is the same feature as the final layer of the proposed model. Glitch features are highlighted by their label color in 3D space, and clustering appears to be well-separated according to the labels. Such a well-separated

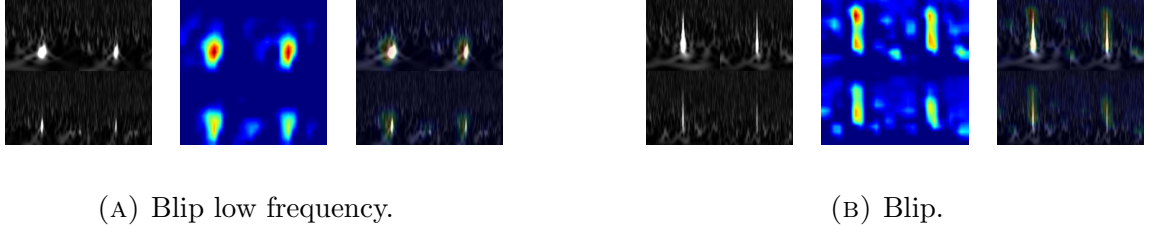


FIGURE 3. Three-column layout displays the input image (left), the saliency map (center), and the overlaid image (right). (A) and (B) correspond to “Blip_Low_Frequency” and “Blip” from the O3a test dataset, respectively.

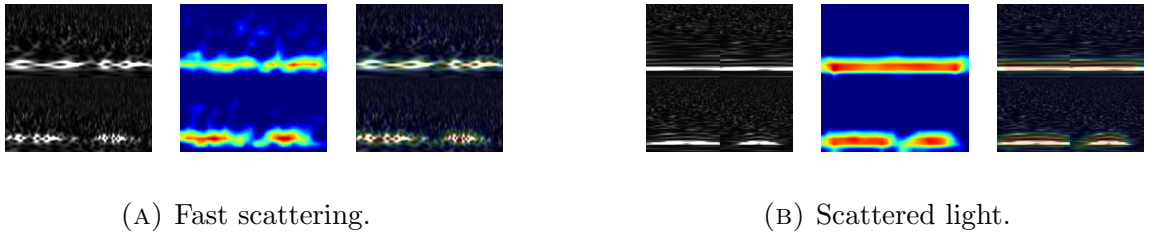


FIGURE 4. (A) and (B) plotted as the three-column layout correspond to “Fast_Scattering” and “Scattered_Light” from the O3a test dataset.

structure for each label indicates easier classification and is consistent with the accuracy of the confusion matrix shown in Fig .2.

3.3.2 Highlighting by Observation Periods The O1&O2 and O3 datasets comprise glitches collected from different periods. Because gravitational wave detectors are upgraded daily to improve their detection sensitivity, we investigated whether the characteristics of these glitches changed over time. We focused on “Blip” and “Koi_Fish” as representative glitches from the O1&O2 and O3 datasets, highlighting them based on different observation periods, as illustrated in Fig. 6. We confirmed that each glitch was closely clustered and the other glitches were not separated between the O1&O2 and O3a datasets. These morphological stabilities of the glitches may suggest that their sources — whether instrumental or environmental — did not significantly change during the detector upgrades. These UMAP results could render the analysis more comprehensive and insightful, potentially providing additional information on the physical origins of the glitches.

Conversely, “Helix” and “Light_Modulation” were clearly separated between the O1&O2 and O3a datasets, as depicted in Fig. 7. Regarding the saliency maps shown in Fig. 7 (A), the “Helix” in O3a exhibits stronger amplitudes than those in O1&O2. A similar result is observed for “Light_Modulation” in Fig. 7 (B), where long-duration signals are present in the low-frequency band in O3a.

Consequently, different observation periods may generate variations in glitch features. One possible factor that could explain why the machine learning model recognized them as separate clusters is that the sensitivity of O3a was better than that of the O1&O2 period. This improved the visibility of the glitches and changed their appearance. However, increasing the dataset size and investigating different observation periods is necessary.

In this section, we demonstrated the effectiveness of saliency maps for evaluating the rationale behind model predictions, even for transfer learning. Furthermore, we confirmed that the O1&O2 and O3a datasets had a well-separated structure using UMAP

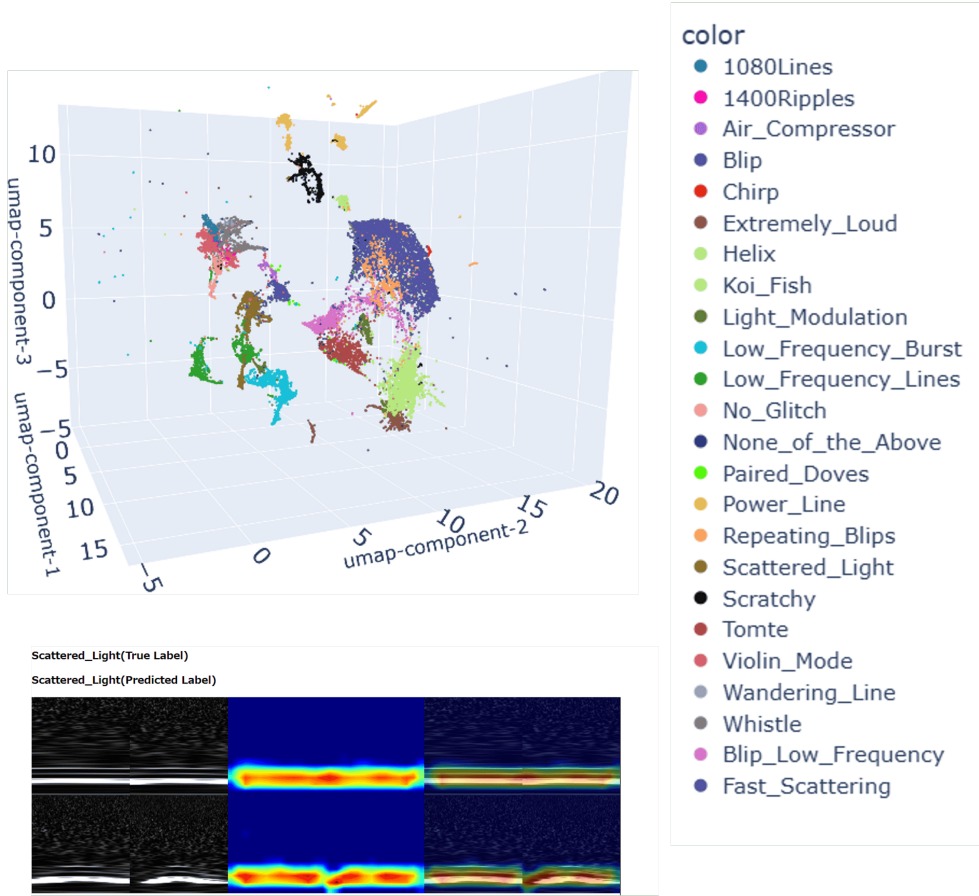


FIGURE 5. Glitch clustering in 3D space is performed by UMAP, where the input features for UMAP are extracted from the last max-pooling layer of the transfer learning model. All of the O1&O2 and O3a glitches are plotted in this space. A saliency map of a glitch on an arbitrary coordinate is also displayed.

visualization, and some of the glitches were separated owing to the different observation periods.

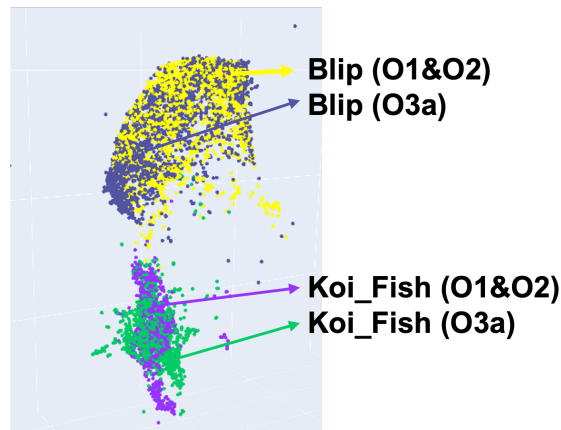


FIGURE 6. Glitch clustering on a 3D space, focusing on “Blip” and “Koi.Fish,” shows that glitches labeled O1&O2 and O3a tend to cluster in relatively close regions. Notably, plotted colors differ from Fig. 5 for comparison between the O1&O2 and O3a glitches.

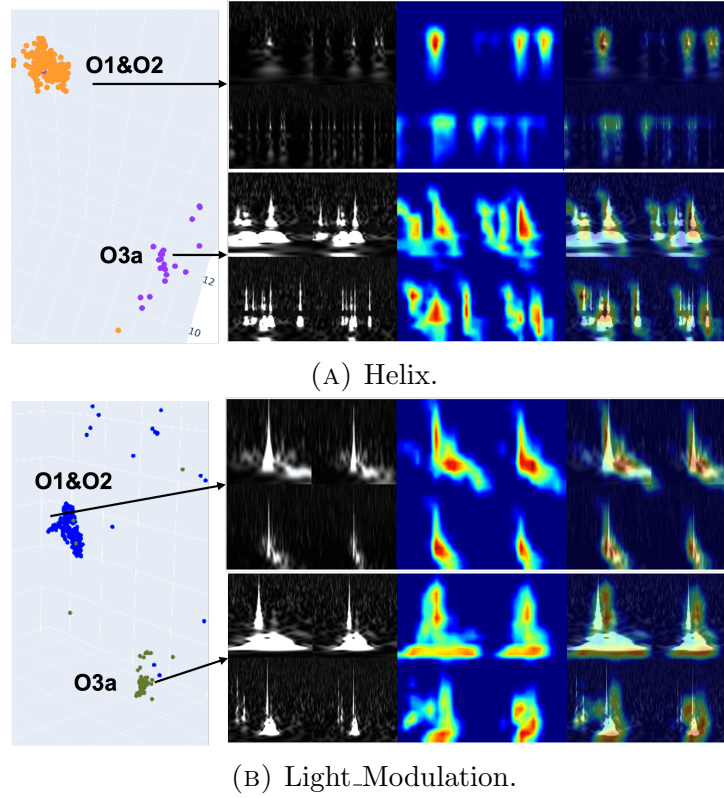


FIGURE 7. Glitch clustering on a 3D space, focusing on (A) “Helix” and (B) “Light Modulation,” shows that glitches labeled O1&O2 and O3a tend to cluster in relatively far regions. Notably, plotted colors differ from Fig. 5 for comparison between the O1&O2 and O3a glitches.

4 Summary In this study, we proposed an explainability framework for glitch noise classification during different observation periods. The pretrained model on the O1&O2 dataset achieved a high accuracy value in transfer learning with O3a. The newly detected glitch in O3a was reasonable, with interpretable results discussed using both UMAP and Score-CAM. Furthermore, our analysis suggested that glitch morphology changed over different periods.

These visualizations promote understanding of the decision-making process in deep learning models. This ensures the validity of the predictions of the model. The observed variations in the morphology of the glitch may provide information to understand their origins, whether instrumental or environmental sources.

In this study, the O1 and O2 periods were conventionally treated as one observation period. However, this approach may overlook potential morphological changes in glitches during these two periods. Future analyses should consider using shorter periods to capture more dynamic variations in glitch morphology.

Future work will involve verifying the effectiveness of the pretrained model on larger and different period datasets, such as O3b [5] and O4 [10], by comparing its performance with those of models trained from scratch on these datasets and demonstrating its applicability. Moreover, we intend to track the evolution of glitch morphology over longer durations and with shorter observation periods.

Acknowledgments This research was supported in part by JSPS Grant-in-Aid for Scientific Research [Nos. 22H01228 and 23K22499 (K. Somiya) and Nos. 23H01176, 23K25872, and 23H04520 (H.Takahashi)] and Grant-in-Aid for JSPS Fellows [No. 22KF0329 (M.

Meyer-Conde)]. This research was supported by the Joint Research Program of the Institute for Cosmic Ray Research, the University of Tokyo, and Tokyo City University Prioritized Studies.

REFERENCES

- [1] P. Brady, G. Losurdo, and H. Shinkai. *LIGO, VIRGO, and KAGRA as the International Gravitational Wave Network*, pages 1–21. Springer Singapore, Singapore, 2020.
- [2] B. P. Abbott et al. Multi-messenger observations of a binary neutron star merger. *The Astrophysical Journal Letters*, 848(2):L12, oct 2017.
- [3] S. Soni, C.PL. Berry, S.B. Coughlin, M. Harandi, C.B. Jackson, K. Crowston, C. Østerlund, O. Patane, A.K. Katsaggelos, L. Trouille, et al. Discovering features in gravitational-wave data through detector characterization, citizen science and machine learning. *Class. Quantum Grav.*, 38(19):195016, September 2021.
- [4] M. Zevin et al. Gravity spy: lessons learned and a path forward. *The European Physical Journal Plus*, 139(1), jan 2024.
- [5] J Glanzer, S Banagiri, SB Coughlin, S Soni, M Zevin, Christopher Philip Luke Berry, O Patane, S Bahaadini, N Rohani, K Crowston, et al. Data quality up to the third observing run of advanced LIGO: Gravity Spy glitch classifications. *Class. Quantum Grav.*, 40(6):065004, 2023.
- [6] D. George, H. Shen, and E. Huerta. Deep transfer learning: A new deep learning glitch classification method for advanced LIGO. *arXiv preprint arXiv:1706.07446*, 2017.
- [7] T. Fernandes, S. Vieira, A. Onofre, J. C. Bustillo, A. Torres-Forné, and J. A. Font. Convolutional neural networks for the classification of glitches in gravitational-wave data streams. *Classical and Quantum Gravity*, 40(19):195018, 2023.
- [8] W. Haofan, W. Zifan, D. Mengnan, Y. Fan, Z. Zijian, D. Sirui, M. Piotr, and H. Xia. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, Seattle, WA, USA, 2020. IEEE.
- [9] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [10] Y. Wu, M. Zevin, C.PL. Berry, K. Crowston, C. Østerlund, Z. Doctor, S. Banagiri, C.B. Jackson, V. Kalogera, and KA.K. atsaggelos. Advancing glitch classification in Gravity Spy: multi-view fusion with attention-based machine learning for Advanced LIGO’s fourth observing run. *Classical and Quantum Gravity*, 42(16):165015, aug 2025.
- [11] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A.K. Katsaggelos, S.L. Larson, T.K. Lee, C. Lintott, T.B. Littenberg, A. Lundgren, C. Østerlund, J.R. Smith, L. Trouille, and V. Kalogera. Gravity Spy: integrating advanced LIGO detector characterization, machine learning, and citizen science. *Class. Quantum Grav.*, 34(6):064003, feb 2017.
- [12] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J.R. Smith, V. Kalogera, and A. Katsaggelos. Machine learning for Gravity Spy: Glitch classification and dataset. *Information Sciences*, 444:172–186, 2018.
- [13] S. Bahaadini, N. Rohani, S. Coughlin, M. Zevin, V. Kalogera, and A.K. Katsaggelos. Deep multi-view models for glitch classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2931–2935, New Orleans, LA, USA, 2017. IEEE.
- [14] N. Koyama, Y. Sakai, S. Sasaoka, D. Dominguez, K. Somiya, Y. Omae, Y. Terada, M. Meyer-Conde, and H. Takahashi. Enhancing the rationale of convolutional neural networks for glitch classification in gravitational wave detectors: a visual explanation. *Machine Learning: Science and Technology*, 5(3):035028, 2024.
- [15] M.D. Zeiler. ADADELTA: An Adaptive Learning Rate Method, 2012.
- [16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Vancouver, Canada, 2019.
- [17] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [18] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, Dec 2000.