Draft version April 29, 2025
Typeset using LaTeX **twocolumn** style in AASTeX7

ASTR8070 Capstone Project
Clustering & Classification of Gravitational Wave Glitches

Suyash Deshmukh[1]

[1] *Vanderbilt University*

ABSTRACT

The analysis of gravitational wave data from LIGO and Virgo is complicated by the presence of short-duration non-Gaussian noise transients known as glitches, which can obscure real GW signals or mimic astrophysical events. In this project, I explore the unsupervised clustering and supervised classification of glitches using data from Gravity Spy, a citizen-science project that crowd-sources glitch labeling. I investigate different time-frequency representations - Log-Mel spectrograms and Q-transforms at various Q-values - and assess their impact on the effectiveness of KMeans clustering with t-SNE projections and Random Forest classifiers. This work establishes a baseline for future developments of ensemble approaches to improve glitch classification and clustering.

1. INTRODUCTION

Gravitational wave (GW) observatories such as LIGO and Virgo operate at extreme sensitivities, making them vulnerable to short-duration noise artifacts known as glitches. These glitches, arising from environmental and instrumental disturbances, pose significant challenges for data analysis pipelines. They can obscure real GW signals or even mimic genuine astrophysical events. For instance, a glitch in the LIGO Livingston detector partially overlapped with the landmark GW170817 binary neutron star detection, delaying its confirmation and necessitating careful signal cleaning (B. Abbott et al. 2017). Around 20% of gravitational-wave candidate events in the third observing run of LIGO-Virgo required some form of glitch mitigation (D. Davis et al. 2022). As improvements to GW detectors increase their sensitivity, the rate of detected signals will rise, further elevating the risk of real events overlapping with glitches. Consequently, robust glitch characterization and mitigation are critical for future GW astronomy. Significant efforts have been devoted to modeling and mitigating glitches (e.g., J. Powell (2018); K. Chatziioannou et al. (2021); S. Ghonge et al. (2023)).

While many glitches have been successfully categorized—such as blips, helix, and whistles—and some, like air compressor glitches, have been mitigated, challenges remain. The populations underlying many glitch classes are not yet fully understood, many glitches remain uncategorized, and new glitch types emerge with each detector upgrade. Improving glitch characterization is vital for better noise modeling, subtraction, and ultimately for enabling more confident GW discoveries.

In this work, I explore the use of traditional machine learning techniques on data from Gravity Spy, a citizen-science GW glitch labeling project (M. Zevin et al. 2017), to perform glitch clustering and classification. Time-frequency representations of glitches are generated using both Log-Mel spectrograms and Q-transforms at varying Q-values. I apply t-distributed Stochastic Neighbor Embedding (t-SNE) followed by KMeans clustering to investigate the separability of glitch classes, and I train Random Forest classifiers to evaluate supervised learning performance across the different representations of the same data. This enables an assessment of how different pre-processing choices affect clustering quality and classification accuracy. This study is a stepping stone towards future development of ensemble methods to enhance glitch classification and clustering.

2. METHODOLOGY

2.1. *Data Generation*

The dataset used in this study comes from the Gravity Spy project, a citizen-science initiative that crowd-sources the labeling of LIGO and Virgo time-series data. I used the first dataset release, which contains 7,966 pre-labeled events covering 21 classes: one for true gravitational-wave chirps, one for ”no glitch” events, 18 for characterized glitch types, and one for uncharacterized glitches. The dataset is publicly available (S. Coughlin 2018). Using the provided GPS times for each event, time-series data were retrieved via GWpy’s built-in `TimeSeries.fetch_open_data()` function (D. M. Macleod et al. 2021).

For each event, time-frequency representations were generated using two techniques:

- **Q-transforms at multiple Q-values** ($Q = 2, 4, 8, 16, 32, 64, 128$): Q-transforms are a widely used time-frequency representation in GW data analysis. GWpy's `q_transform()` function converts time-series data into the frequency domain, allowing manual specification of the quality factor Q, which controls the tradeoff between time and frequency resolution. Lower Q values provide better time resolution but poorer frequency resolution, while higher Q values invert this relationship. (If not manually specified, `q_transform()` calculates and selects an optimal Q for any given input data).

- **Log-Mel spectrograms**: Log-Mel spectrograms are commonly used in audio processing, applying a mel-frequency scaling to emphasize frequency bands important for human perception. LibROSA has a handy `melspectrogram()` function which I used to get the Log-Mel spectrograms (B. McFee et al. 2015). While not originally developed for GW data, they provide an alternative time-frequency representation that highlights different features compared to Q-transforms.

Each event was thus associated with eight images, corresponding to different pre-processing choices. These images served as inputs for both the clustering and classification experiments. (Example images are shown in Fig. 1.)

2.2. *Data Analysis Techniques*

For **unsupervised clustering**:

For each set of glitch images (defined by a specific Q-value or by the Log-Mel spectrogram), a two-dimensional embedding was generated using t-distributed Stochastic Neighbor Embedding (t-SNE). t-SNE is a dimensionality reduction technique like principal component analysis (PCA) that maps high-dimensional data to a low-dimensional (usually 2D) representation where similar points remain close together. The main difference between PCA and t-SNE is that PCA is restricted to linear reductions while t-SNE is non-linear, making it particularly useful for visualizing clustering tendencies in datasets like this. The resulting t-SNE embeddings are shown in Fig. 2, with points labeled by their true classes.

Following dimensionality reduction, clustering was performed using Scikit-Learn's KMeans (F. Pedregosa et al. 2011), a widely used centroid-based clustering algorithm. KMeans partitions data into K clusters by minimizing the distance between points and cluster centroids. The predicted cluster assignments were compared to the known labels to estimate clustering accuracy.

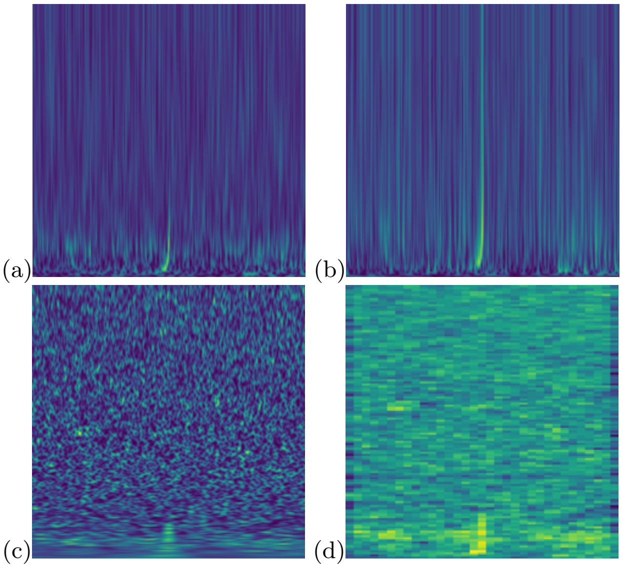

Figure 1. Examples showcasing the image inputs fed into clustering and classification algorithms. All images correspond to a chirp event detected by the LIGO Livingston detector at GPS time 1128672502. The x-axis represents time and the y-axis frequency; however, axis values were removed before model training to avoid confusion during learning. (a) Q-transform of the event without a manually specified Q-value (the `q_transform()` function selected it automatically). (b) Q-transform with Q=2, resulting in high time resolution but low frequency resolution, producing vertically streaked features. (c) Q-transform with Q=128, resulting in high frequency resolution but low time resolution, producing horizontally stretched features, especially visible at low frequencies. (d) Log-Mel spectrogram of the event, providing a lower resolution view structured along mel-frequency bands rather than linear frequency scaling.

For **supervised classification**:

Random Forest classifiers from Scikit-Learn were trained separately on each set of glitch images without prior dimensionality reduction (i.e., trained directly on the raw images shown in Fig. 1). Random Forests are based on many decision trees (hence a forest), where each tree is trained on a random subset of the data and features. The final prediction is made via majority voting across trees, resulting in a much stronger performance than any of the trees alone. Model performance was evaluated using classification accuracy measured on held-out test data.

3. RESULTS AND DISCUSSION

3.1. *Unsupervised Clustering*

The t-SNE dimensionality reductions reveal notable differences across input types (Fig. 2). Certain glitch classes, such as `Extremely_Loud` and `Koi_Fish`, form distinct clusters across nearly all representations. More

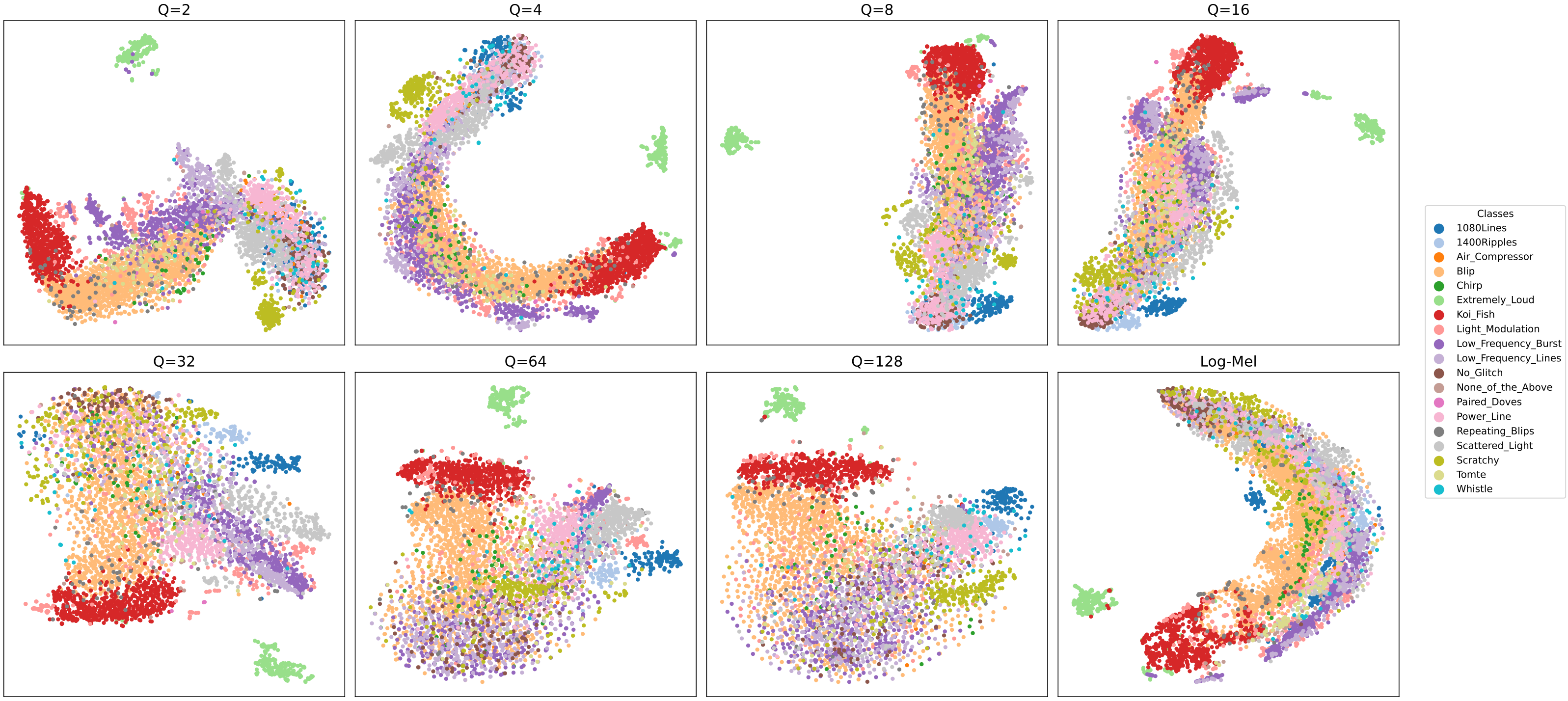

Figure 2. Two-dimensional t-SNE embeddings of glitch images. Each panel corresponds to a different input representation (seven Q-values and one Log-Mel spectrogram). Points are colored according to their true labels, highlighting the separability (or lack thereof) of different glitch types.

interestingly, other glitch types cluster clearly only at specific Q-values.

For example, `1080Lines` glitches are mixed with other classes at low Q-values but progressively separate into distinct clusters at higher Q-values. When using Log-Mel spectrograms, the `1080Lines` glitches split into three groups: one well-separated from the data bulk, and two embedded within denser regions - suggesting possible substructure within the class.

Conversely, `Scratchy` glitches cluster better at low Q-values but appear more mixed at mid-to-high Q-values and in Log-Mel representations. At low Q-values, two nearby but distinct `Scratchy` subclusters emerge, distinct from other glitches.

Overall, KMeans clustering struggled with the mixed structure of the data, with maximum clustering accuracy reaching only 34.31% (at Q=2). A full list of clustering accuracies is given in Table 1. Fig 3 shows the confusion matrix generated from KMeans clustering on the t-SNE embeddings. While there is a lot of confusion between various events, confusion between glitches differs for each embedding. For example, the Q=2 model can confidently say that `Wandering_Line` and `Whistle` glitches are not `Blips`, but gets a lot of `Tomte` and `Violin_Mode` glitches confused for `Blips`. This is different from the Q=4 and Q=16 models which don't misclassify `Violin_Mode` and the Q128 model which misclassifies `Tomte` glitches to a much lower level. This suggests that certain time-frequency representations are better suited for distinguishing between some glitch types, and worse at others.

Input	Accuracy
Q=2	0.3431
Q=4	0.3047
Q=8	0.2652
Q=16	0.3078
Q=32	0.3284
Q=64	0.3333
Q=128	0.3173
Log-Mel	0.2326

Table 1. Clustering accuracies for KMeans applied to t-SNE embeddings of glitch images generated using different Q–transform values and log-mel spectrogram inputs.

3.2. *Supervised Classification*

Supervised Random Forest classifiers achieved significantly higher accuracies than the unsupervised KMeans clustering models. The highest performance reached was 90.59% accuracy using the log-mel spectrogram representations (see Table 2 for full list of accuracies). This is a notable result, as log-mel spectrograms are not tra-

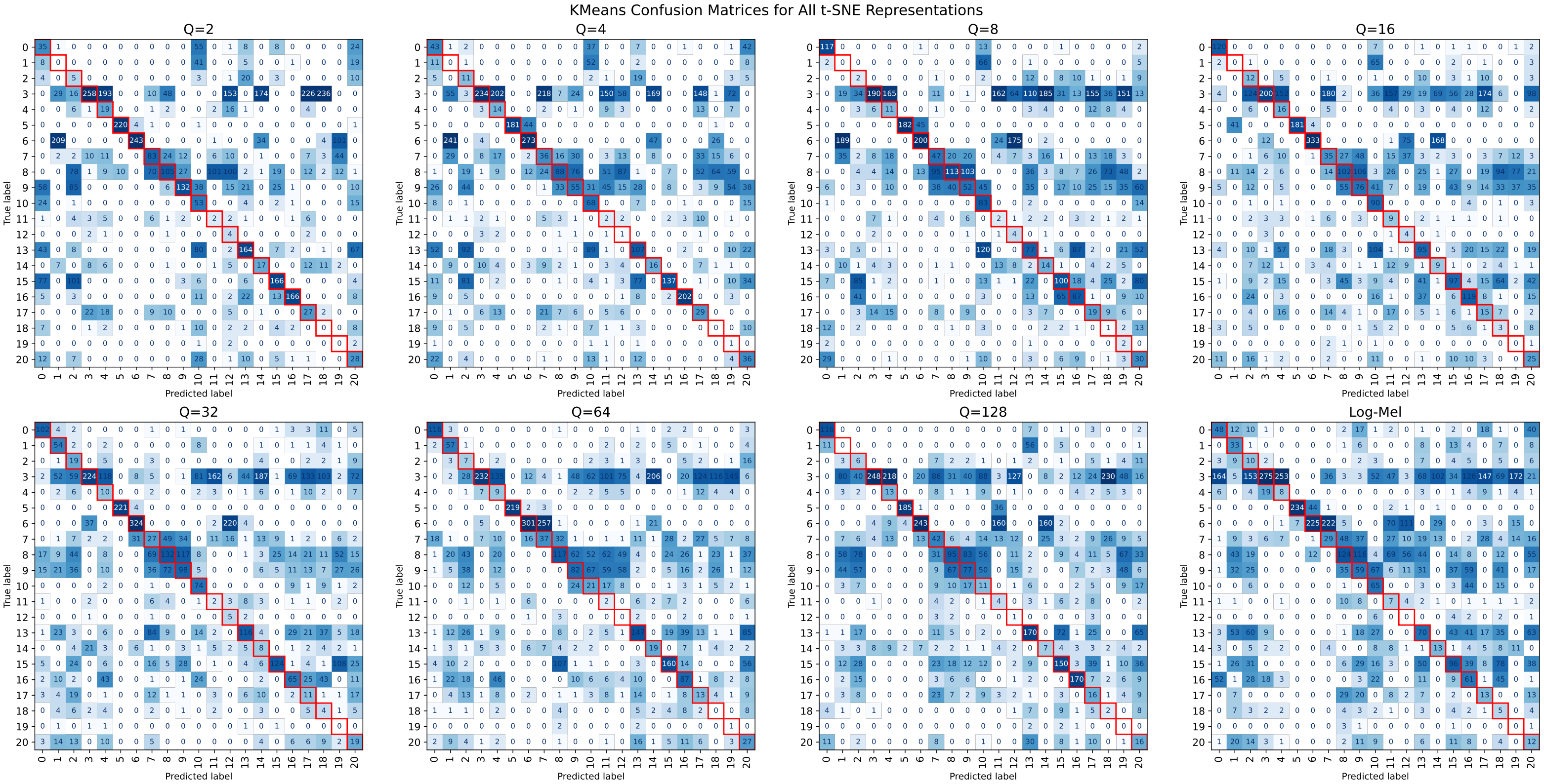

Figure 3. Confusion matrices for KMeans clustering on t-SNE embeddings of all eight input types (seven Q-transform values and one log-mel spectrogram). To conserve space, class labels are replaced with numeric indices: 0 - 1080Lines, 1 - 1400Ripples, 2 - Air_Compressor, 3 - Blip, 4 - Chirp, 5 - Extremely_Loud, 6 - Koi_Fish, 7 - Light_Modulation, 8 - Low_Frequency_Burst, 9 - Low_Frequency_Lines, 10 - No_Glitch, 11 - None_of_the_Above, 12 - Paired_Doves, 13 - Power_Line, 14 - Repeating_Blips, 15 - Scattered_Light, 16 - Scratchy, 17 - Tomte, 18 - Violin_Mode, 19 - Wandering_Line, 20 - Whistle.

ditionally used in gravitational wave analysis. Their strong performance here suggests that alternative time-frequency representations should be studied further as they may offer valuable insights for glitch classification.

Among the Q-transforms, mid-range Q-values (Q=16 and Q=32) yielded the best classification performance, outperforming both very low and very high Q-values. This supports the idea that supervised classifiers benefit from a balanced trade-off between time and frequency resolution when distinguishing glitch types.

Dimensionality reduction techniques like t-SNE were not used in the Random Forest pipeline here, and so further work is needed to assess whether reduced representations could improve or degrade supervised classification accuracy.

4. SUMMARY AND FUTURE WORK

This study demonstrates that different time-frequency image representations of the same data can have a considerable effect on clustering and classification of gravitational wave glitches. Unsupervised KMeans clustering on t-SNE embeddings were only able to reach low accuracies (up to 34.31% accuracy), but clearly showed that different time-frequency representations can better isolate specific glitch types, suggesting room for targeted ensemble modeling. They also showed that glitch classes may in fact have some sub-clusters present, which must be looked into further.

Input	Accuracy
Q=2	0.8108
Q=4	0.8494
Q=8	0.8645
Q=16	0.8744
Q=32	0.8762
Q=64	0.8177
Q=128	0.7466
Log-Mel	0.9059

Table 2. Classification accuracies for Random Forest models trained on each time-frequency representation: Q-transforms at various Q-values and the log-mel spectrogram.

Supervised Random Forest classifiers attained quite high accuracies, with a peak of 90.6% using Log-Mel spectrograms, showing that Log-mel spectrograms, de-

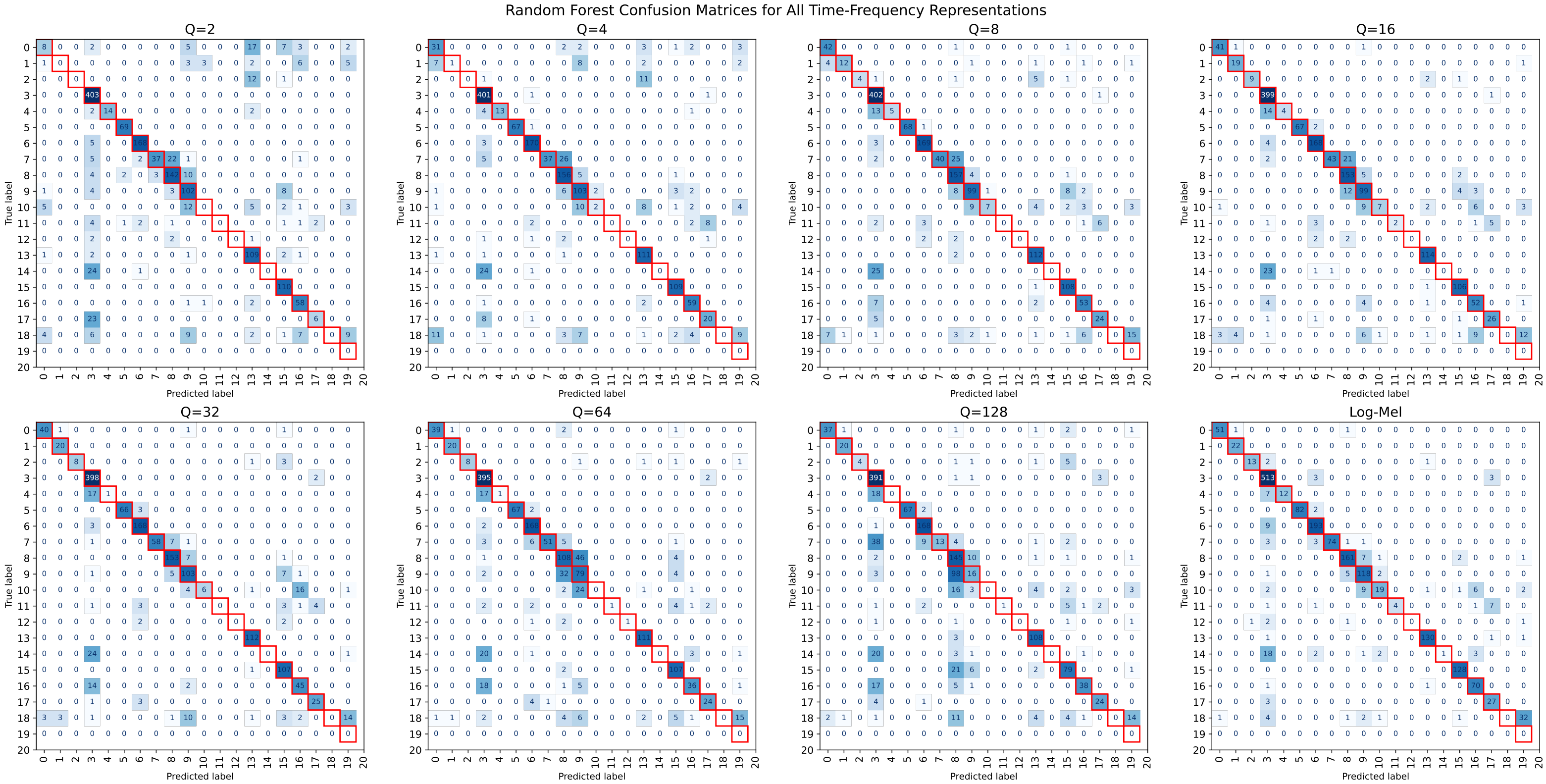

Figure 4. Confusion matrices for Random Forest classification across all eight time-frequency representations. Class labels are represented numerically to save space: 0 - 1080Lines, 1 - 1400Ripples, 2 - Air_Compressor, 3 - Blip, 4 - Chirp, 5 - Extremely_Loud, 6 - Koi_Fish, 7 - Light_Modulation, 8 - Low_Frequency_Burst, 9 - Low_Frequency_Lines, 10 - No_Glitch, 11 - None_of_the_Above, 12 - Paired_Doves, 13 - Power_Line, 14 - Repeating_Blips, 15 - Scattered_Light, 16 - Scratchy, 17 - Tomte, 18 - Violin_Mode, 19 - Wandering_Line, 20 - Whistle.

spite not being tailored for gravitational wave data, have strong potential as inputs for glitch classification.

Moving forward, several directions are promising:

- Investigating ensemble methods that combine multiple representations to take advantage of the varying confusions in each embedding.

- Evaluating dimensionality reduction techniques like t-SNE for supervised classification.

- Extending this analysis to bigger glitch datasets to improve model performances, especially one which is not so heavily skewed towards one glitch type (`Blips` were the most common type of glitch in this dataset by a huge margin).

ACKNOWLEDGMENTS

I would like to thank OpenAI's ChatGPT for help in making a tqdm progress bar work with my parallelized code. I initially just wanted it because it is cool, but without it I would have never realized that my initial multi-threading implementation was not working and would take 88 hours to run. Fixing it let me generate my data in 2 hours and 40 minutes instead. All of these multi-threaded computations were performed using resources provided by the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University.

5. APPENDIX

All code used for data generation, clustering, classification, and the creation of figures and tables in this study is publicly available at: https://github.com/Suyash-Deshmukh/astr_8070_s25/tree/main/coursework/student_folders/suyash_deshmukh/final_project

REFERENCES

Abbott, B., Abbott, R., Abbott, T., et al. 2017, Physical Review Letters, 119, doi: 10.1103/physrevlett.119.161101

Chatziioannou, K., Cornish, N., Wijngaarden, M., & Littenberg, T. B. 2021, Physical Review D, 103, 044013, doi: 10.1103/PhysRevD.103.044013

Coughlin, S. 2018, Zenodo, doi: 10.5281/ZENODO.1476550

Davis, D., Littenberg, T. B., Romero-Shaw, I. M., et al. 2022, Classical and Quantum Gravity, 39, 245013, doi: 10.1088/1361-6382/aca238

Ghonge, S., Brandt, J., Sullivan, J. M., et al. 2023, arXiv, doi: 10.48550/ARXIV.2311.09159

Macleod, D. M., Areeda, J. S., Coughlin, S. B., Massinger, T. J., & Urban, A. L. 2021, SoftwareX, 13, 100657, doi: 10.1016/j.softx.2021.100657

McFee, B., Raffel, C., Liang, D., et al. 2015, in Proceedings of the 14th python in science conference, Vol. 8

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825

Powell, J. 2018, Classical and Quantum Gravity, 35, 155017, doi: 10.1088/1361-6382/aacf18

Zevin, M., Coughlin, S., Bahaadini, S., et al. 2017, Classical and Quantum Gravity, 34, 064003, doi: 10.1088/1361-6382/aa5cea