LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note **LIGO-T2200207–v1** 2022/09/20

Identifying Witnesses to LIGO Glitches Using Auxiliary Channels

Luis Gabriel C. Bariuan

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW22-295
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

http://www.ligo.caltech.edu/

Abstract

The exquisite sensitivity of the Laser Interferometer Gravitational-Wave Observatory (LIGO) makes it extremely vulnerable to glitches, which are short-lived noise transients that surpass the Gaussian-distributed background noise. Glitches occur in the main channel at a high rate, in comparison to real signals. As a result, these triggers can mask or mimic real gravitational wave signals derived from astrophysical sources resulting in high false positive rate in the search pipelines. Along with the main channel, LIGO maintains a large set of auxiliary channels. These auxiliary channels monitor the state of the detector and can witness these glitches. In this work, we present an exploration of Boolean Matrix Factorization (BMF) as a possible method for clustering loud triggers in a set of auxiliary channels that coincide with glitches and selecting certain channels as glitch witnesses. We test the factorization against the real data and simulated data, to determine how BMF performs in comparison to other factorization models such as the Non-Negative Matrix Factorization (NMF). By doing so, our goal is to examine if some Gravity Spy glitch classes have consistently correlated loud triggers in certain sets of channels which may allow domain experts to localize and fix glitches at their source.

1 Introduction

The Laser Interferometer Gravitational Wave Observatory (LIGO) is a ground-based gravitational wave (GW) detector that operates using the principles of a modified Michelson interferometer. Highly energetic astrophysical events such as the mergers of compact objects (e.g., black holes, neutron stars) produce ripples in spacetime that travel at the speed of light. The ripples induce changes in the spacetime metric which become manifest when LIGO detects changes in the length of the interferometer arms. Hence, the resulting time-dependent data, $h(t)$, encodes the physical properties of the origin of the GW phenomena [1].

The GW signals detected are extremely faint; the change in the length of the interferometer arm is smaller than a proton's diameter. The small amplitude of the signal requires the LIGO detectors to have exquisite sensitivity. As a result, the detectors become vulnerable to terrestrial noise, which may include environmental and instrumental noise. Despite efforts at noise shielding, short-lived, noise transient artifacts often called "glitches" plague the main channel. Glitches may mimic or mask real GW signals which may result in false alarms [2].

Alongside the main channel, LIGO maintains a large set of auxiliary channels, upper bounded at $O(10^5)$. These channels may bear witness to these glitches. As a result, previous work has been dedicated to associating glitches within the main channel to loud triggers found within auxiliary channels. By doing so, the aim is to find witness auxiliary channels that may serve as veto generators in order remove time segments within the main channel data contaminated with glitches.

Given the large number of degrees of freedom, algorithmic and machine learning methods have been employed to tackle this problem. Algorithms such as hveto and UPV utilized iterative processes to determine the most efficient witness channels [3, 4]. On the other hand, pipelines such as iDQ, employed supervised machine learning techniques to determine in near real-time the probability of glitches within a time segment given the data recorded within auxiliary channels. The iDQ pipeline was combined with algorithms such as the Ordered Veto List (OVL) to determine suitable veto generators within a time segment [5].

Recent work, on the other hand, by [2], employed unsupervised machine learning techniques such as Non-Negative Matrix Factorization (NMF) and CP/PARAFAC Tensor Decomposition to cluster triggers within auxiliary channels as a method of finding valid veto generators. This method is unique in that it searches for witness channels in an "all-at-once" fashion, in contrast to previous methods which relied on iteration. In addition to determining veto generators, this method is able to discover groups of auxiliary channels associated with glitch morphologies which may help domain experts discover glitch generation mechanisms to potentially fix the glitches at their source.

2 Background and Motivation

We utilize the proposed pipeline of vetoing glitches established by [2] to explore various models (e.g., factorization techniques, clustering methods) to compare with existing results using various metrics such as homogeneity and coverage. The goals are to find sets of auxiliary channels which may serve as veto generators and to use the techniques to co-cluster information about glitch morphology classes to channels based on data obtained from the Gravity Spy catalog. This information can be relayed to domain experts who may be able to localize and fix the source of the glitches at their source.

To approach this problem, we utilized the time-dependent data recorded through the main channel and the auxiliary channels. Loud triggers defined to be recorded signals that have signal-to-noise ratio SNR ≥ 7.5, which may include glitches. The trigger and channel information are converted into a data matrix $Z \in \mathbb{R}^{|G|\times|A|}$ and a 3-mode tensor $\chi \in \mathbb{R}^{|G|\times|A|\times|F|}$, where $|G|$ is the number of glitches, $|A|$ is the number of auxiliary channels, and $|F|$ is the number of features in the tertiary tensor mode.

Given the data matrix, we aimed to decouple glitch and channel information using Boolean Matrix Factorization (BMF), which simplifies the analysis by binarization. Instead of using the SNR to encode the information about the glitches as in the case of the two models explored in [2], BMF only encodes information about the presence or absence of glitches.

Boolean Matrix Factorization (BMF) is a method developed to factorize a data matrix Z to two separate factor matrices, such that $Z \approx XY$. Although this method works using the same principle as traditional factorization methods, the matrix decomposition is performed over the Boolean semi-ring $\mathbb{B}$. More explicitly, if we apply this factorization method to the data matrix, we can decompose $Z \in \{1, 0\}^{|G|\times|A|}$ such that in index notation:

$$Z_{ij} \approx (X \circ Y)_{ij} = \bigvee_{l=1}^{K} B_{il} C_{lj}, \tag{1}$$

where we define $X \in \{0, 1\}^{|G|\times K}$ and $Y \in \{0, 1\}^{K\times|A|}$, and $K << \min\{|G|, |A|\}$, the rank of the data matrix [6].

To do this, BMF finds X and Y such that the "distance", defined to be the Frobenius norm $||.||_F$ of $Z - X \circ Y$ is minimized. Explicitly, BMF performs the minimization as:

$$||Z - X \circ Y||_F^2 = \sum_{i,j} Z_{ij} \oplus (X \circ Y)_{ij}. \tag{2}$$

Once the matrices are factored, X and Y become highly interpretable due to the binary nature of the data. In particular, we can use X and Y to learn more information about the associated glitches and auxiliary channels using co-clustering [6, 2].

3 Methods

We test the efficacy of the BMF model and compare it to NMF in two different ways: simulated data and real data. The simulated data allows us to verify the ability of BMF to recover known structures encoded with simulated glitch classes and their couplings to different channels as they are injected to a data matrix. The real data, which does not have the ground-truth about the association of glitch classes to channels then tests out the ability of BMF in capturing latent patterns within the channel space.

For this work, we utilized the BMF implementation `BooleanFactorization`.[1]. This particular implentation utilizes posterior inference via message passing to perform the factorization [7]. We set the number of maximum message passing (iterations) to 500.

3.1 Simulated Data

3.1.1 Synthetic Data Generator

We established a synthetic data generator that aims to mimic the data matrix obtained from LIGO through the pipeline established by [2]. By doing so, we can use the generated data matrices as a controllable testbed that helps us determine how our proposed methods behave in the presence of different kinds of patterns, before we attempt to identify them "in the wild" given the open-ended nature of the problem.

First, we generate a glitch catalog that emulates the Gravity Spy Catalog. To do this, we created a dictionary which associates a glitch class to a set of channels. In this case, we simply label as "gs_i", where i denotes the i-th glitch class in the catalog. Each glitch class is associated randomly with a set of channels ranging between $n^c_{min} = 3$ to $n^c_{max} = 6$, determined randomly. Note that for cases which require a tertiary tensor mode, the generator also appends a feature value to each class that allow for the generation of the third tensor mode.

Using the generated channels, we can cluster glitch classes in multiple ways. One is to have a proportion of the glitch classes share a set of channels, which represents multiple glitches being triggered by the same source. Alternatively, we cluster glitch classes such that a given morphology can be triggered by different sets of channels. Finally, we can also generate a catalog in which the channels associated between different glitch classes are considered to be

[1]https://github.com/mravanba/BooleanFactorization

orthogonal. In this work, we explore the ability of BMF to recover completely orthogonal glitch classes.

We use the information encoded in the glitch class catalog to generate data matrices and tensors, where entries represent the SNR value associated to a glitch, channel, and possibly tensor features. We will discuss, in explicit detail, the construction of the synthetic data matrix. First, an empty matrix is constructed. At each entry, an exponentially-distributed background noise described by the parameter λ_b is randomly generated. Then, we iterate overall all the rows to inject the glitch. At each row, a glitch class is randomly selected among the catalog of glitches along with the associated channels. The selection of the glitch class is described by a probability distribution, which we can modify and skew to bias certain classes. This is aimed to simulate particularly dominant glitch classes within the real data.

Each of the channels associated with a glitch class can be mapped directly to the columns of the data matrix. Using an exponential distribution described by the parameter λ_g, a glitch is generated at the associated columns to the glitch class. Note that for this work, the relationship between λ_b and λ_g is an order of magnitude (factor of 10). We then apply the "loud" trigger threshold and zero out any entry less than the threshold of SNR ≥ 7.5. Note that the selection of this threshold is arbitrary for the synthetic data, the relative differences between the SNR values are the most relevant for our analysis. For the application to BMF, we simply replace all non-zero matrix elements that meet the threshold criterion with "1". Note that the tensor generation works similarly, with the exception of the third mode, where instead the signal is generated with reference to the i-th glitch and j-th auxiliary channel, and k-th feature bin. Overall, the resulting matrices and tensors are sparse, which to a first approximation matches the behavior of data collected. Note that the feature bin may represent other features of the data (e.g., frequency with peak SNR as in [2]).

In order to gain insight to the structure of the data, we performed Singular Value Decomposition (SVD) and plot the singular values of the constructed data matrices. Figure 1 plots the singular values for different data matrices sampled from different glitch class catalogs generated using the synthetic data generator. For comparison, we plot the singular values of a data matrix obtained from the Livingston detector between September 29-30, 2019. The resulting singular values across the simulated and real datasets indicate a fairly low cutoff for the rapid decline in the magnitude of singular values which suggests a low-rank structure across the data.

3.1.2 Simulated Data

For the simulated data, we examined an orthogonal glitch catalog. An orthogonal glitch catalog is defined such that the associated channels in a given glitch class does has no overlap with the associated channels of another glitch class. This mimics the notion that each of the glitch classes have a well-determined origin. Within the glitch catalog, we also designate that 50% of the glitch classes have three different sets of associated channels. This represents a more non-trivial origin for a given glitch class, where a single instance may be derived from multiple sources. The other half of glitch classes are independent and orthogonal. We note that within our simulations, the number of channels associated with glitch classes is not necessarily the same as the number of channels injected within a data matrix since we work

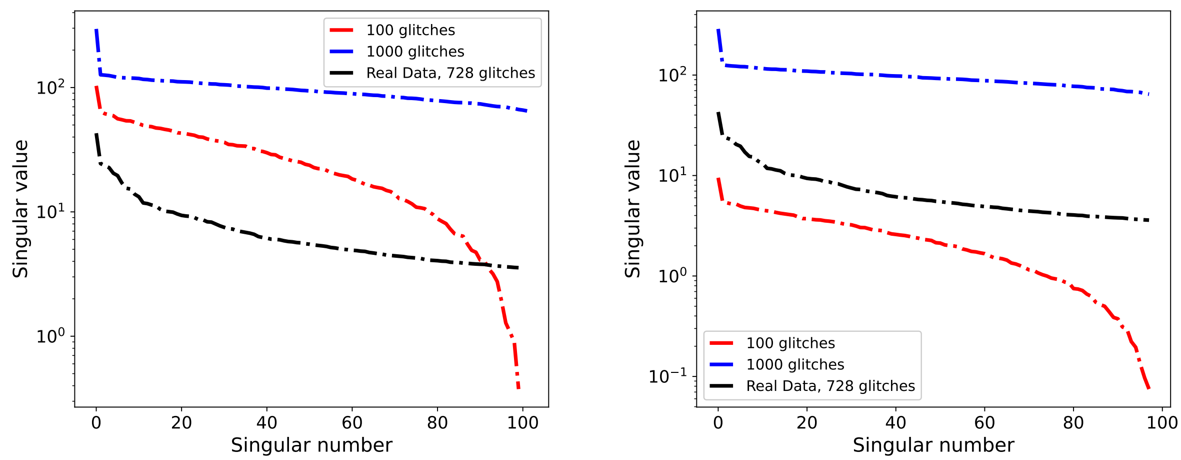

Figure 1: Singular values plot across four different Boolean data matrices. (*Upper*): Singular values plot of the data matrix generated with 100 and 1000 glitches sourced from a glitch catalog that had 50% of glitch classes share one channel. The *upper* plot represents a scenario where different glitch classes that may occur from similar sources. (*Lower*): Singular values plot of the data matrix generated with 100 and 1000 glitches sourced from a glitch catalog where a given glitch class can have multiple sets of auxiliary channels associated. The *lower* plot simulates a scenario glitch classes that could be sourced from different sets of auxiliary channels. For comparison to a real dataset, we plot the singular values of a data matrix with 728 glitches and 803 channels obtained from the Livingston detector between September 29-30, 2019 (blue). Overall, the plots indicate a fairly steep cutoff within the low singular index regime, despite different sample sizes, which suggests that a low-rank approximation should work well within BMF and other low-rank factorization models.

under the assumption that a subset of channels are not involved in glitch generation.

We generated data matrices with 2000 glitches and 200 channels, which provides a comparable size to real data matrices collected over a 4-day period. We factored the generated data matrix Z to $\approx XY$ at rank K, where X is the matrix associated with glitches and Y is the matrix associated with the channels. We treat rank as a hyper-parameter and performed a grid-search across different values of $K \in [5, 50]$. We then calculate channel coverage and channel precision, within the context of the simulated data to verify how well BMF recovers an injected structure. We then compare our results to a data matrix factorized using Non-Negative Matrix Factorization (NMF), which factorizes the data matrix over the positive real semi-ring $\mathbb{R}_{>0}$ [2].

Channel coverage is defined as the number of channels recovered versus the number of channels injected (we expect to recover). To calculate this quantity, we examine the columns of the Y^T matrix. We then take the associated row indices of the entries in each column that are non-zero (in this case "1"). These indices represent the channels that are recovered after applying the factorization. We then tabulate the number of "1"s and that represents the number of channels recovered $N_{c,rec}$. We assume using the Law of Large Numbers, that given the large number of glitches simulated, all the channels N_{total} involved with a glitch class are simulated. Hence, channel coverage is defined as $N_{c,rec}/N_{total}$ for this particular set of simulations. Figure 2 plots the average channel coverage and the 1σ error for the completely orthogonal and partially orthogonal dataset for 10 simulations factored across different ranks $K \in [5, 50]$.[2] Overall, channel coverage using the BMF model greatly improves and approaches unity faster than NMF as the rank of the decomposition increases which is indicative that the model is capable of recovering injected structures, such as the association of glitch classes to channels. Comparatively, the NMF model performs well at high ranks. These results indicate that both factorization models are valid ways of finding associations between glitches and channels.

Channel precision, on the other hand, is defined as how well the channels recovered are associated with the correct glitches. To calculate this quantity, we examine both columns of the X (glitch) and Y^T channel matrix. First, we determine, which glitches are associated with a column in X. This is done by finding the row indices of entries that have a "1" entry. We can then associate these indices with the ordered list of generated glitches and the associated glitch classes, which is stored during data generation. We can then generate the list of channels associated with the columns of X by examining the glitch class catalog. This creates a K-length list of channel sets which we denote as C_X. We can then cross-reference C_X with the channel indices obtained from examining the respective columns of Y^T. We denote this K-length list of channel sets obtained from Y as C_{Y^T}. Using these two list of sets C_X and C_Y, we can define channel precision P_C as:

$$P_C = \frac{1}{K} \sum_{i=1}^{K} \frac{|C_{X,i} \cap C_{Y^T,i}|}{\max\left(|C_{X,i}|, |C_{Y^T,i}|\right)}, \tag{3}$$

where the quantity is calculated per-column and averaged across all K-columns within a decomposition. Figure 2 plots channel precision for a simulated data matrix factorized across different ranks $K \in [5, 50]$. To a first, approximation, we find a decreasing trend in precision for a data matrix factorized using BMF and NMF across different ranks, which suggests that utilizing high-ranks within the model likely generates degeneracies and instabilities across columns. Furthermore, NMF demonstrates a much higher value for precision than BMF, which suggests that although BMF is able to cover more channels at lower ranks, a tradeoff is the lower precision due to the "over-return" of injected channels. Hence, a more intermediate rank is likely a more suitable regime to utilize the factorization models.

We can also perform a set of calculation much closer to the scenario within the real data where we do not have the ground truth for which glitch event is associated with subsets of channel. Instead, we only possess information about the glitch events and their classification within the Gravity Spy Catalog. Hence, calculating channel coverage and precision is not possible. Instead, we calculate glitch class coverage and homogeneity.

Glitch class coverage is defined similarly to channel coverage. However, the ratio is the number of glitch classes that are captured versus the total number of glitch classes that are supposed to be recovered. We perform the same assumptions of using the Law of Large Numbers as in the case of channel coverage.

[2]For the rest of this work, we define the 1σ error to be the standard error of the mean.

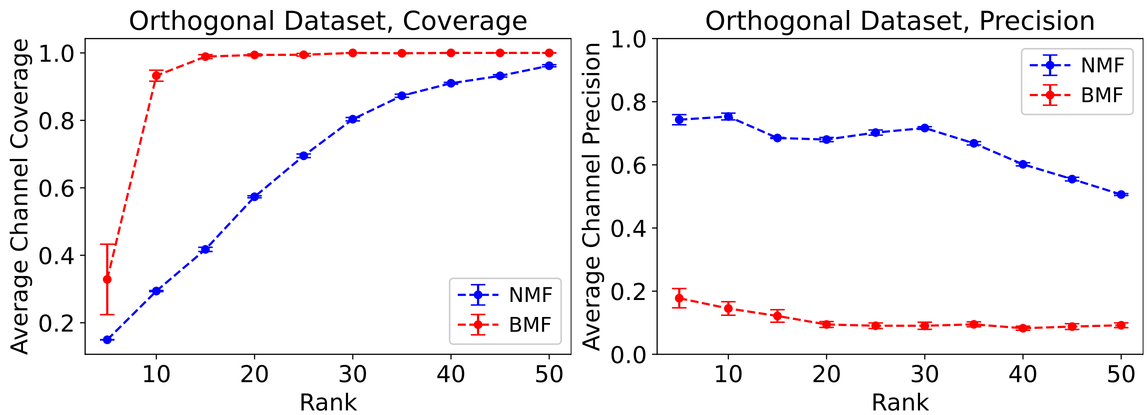

Figure 2: *Left*: Plot of the average channel coverage for the orthogonal dataset with the 1σ error across different factorization ranks K using NMF (red) and BMF (blue). Each point is the average across 10 simulations. *Right*: Plot of the average channel precision for the orthogonal dataset with the 1σ error across different factorization ranks K using NMF (red) and BMF (blue). Each point is the average across 10 simulations. The results indicate that for both NMF and BMF coverage can improve at higher ranks. On the other hand, NMF consistently has better channel precision. The low precision of BMF is primarily driven by errors in the recovered channel, whereby more channels than injected are captured by the model which is also consitent with the high channel coverage of BMF.

Homogeneity, on the other hand, is defined to be:

$$H = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{N_i}, \tag{4}$$

where N_i is the number of unique labels of the i-th factor (columns of Y^T). Note that highly homogeneous auxiliary channel clusters that correspond to gravity classes signifies a homogeneity value close to 1. Figure 3 plots the glitch class coverage and homogeneity for the simulated dataset using both NMF and BMF factorization models. As expected, both NMF and BMF recover high coverage values at higher ranks, as expected, based on the channel coverage behavior. For homogeneity, both models return fairly high values (> 0.60), which suggests that the factor columns are considered to be more "pure". Furthermore, NMF returns far higher homogeneity values which suggests that the binarization of the dataset that contributes to the recovery of the spurious channels leads to less pure columns within the glitch factor matrix.

3.2 Real Data

We divide this sub-section into three parts. The first part provides a short discussion on the data collection of this pipeline. We then proceed to discussing the use of the pipeline to find veto generators. Finally, we explore co-clustering and a brief discussion to demonstrate a proof-of-concept that aims to use use the pipeline to find class-specific veto generators.

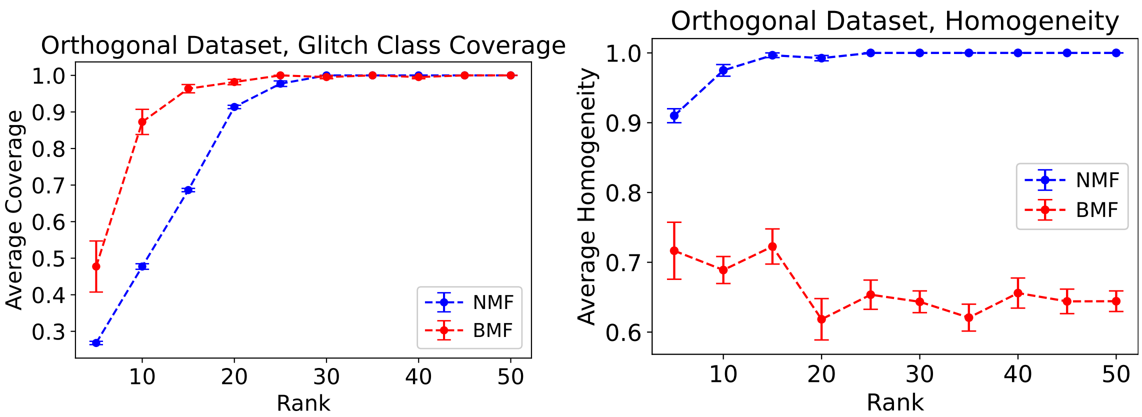

Figure 3: *Left*: Plot of the average glitch coverage for the orthogonal dataset with the 1σ error across different factorization ranks K using NMF (red) and BMF (blue). Each point is the average across 10 simulations. *Right*: Plot of the average homogeneity for the orthogonal dataset with the 1σ error across different factorization ranks K using NMF (red) and BMF (blue). Each point is the average across 10 simulations. The results indicate that for both NMF and BMF glitch coverage can improves at higher ranks. The relatively high homogeneity values for both NMF and BMF indicate that the underlying structure of the dataset in terms of the associated channels to glitch classes is orthogonal. The higher homogeneity values of NMF indicate "purer" factors which matches the channel precision behavior, where BMF tends to recover more spurious channels.

3.2.1 Data Collection

We collected data from the first week of O3b runs of the Livingston detector using the Omicron trigger catalog. More specifically, we gathered data between November 1-4, 2019 for the *training* interval and November 5-6, 2019 for the *validation* interval. We collected 2402 glitch events and we utilized 803 pre-selected safe auxiliary channels (channels that do not witness real GW signals) for our analysis.

3.2.2 Veto Generation

Using the pipeline by [2], we performed both BMF and NMF factorization at $K = 20$ on the training dataset. We find 20 candidate channels from the channel factor matrix. More specifically, we selected the channels that corresponded to the non-zero values in the column matrix for BMF or the values that comprise 90% of the norm of the column for NMF. To determine whether these candidate channels are "good" witnesses, we plot the True Positive/False Positive Curve versus SNR threshold. A "good" witness channel is defined when the True Positive Rate (TPR) becomes greater than the False Positive Rate (FPR) at a given SNR threshold value. By doing so, we are able to determine that the channel is able to witness the triggers that is coincident with the glitches events. Using this criteria, we find a single good witness channel using both NMF and BMF, which indicates that despite binarization, BMF is comparable to NMF in finding veto generators. Figure 4 plots the TPR/FPR versus SNR threshold for the selected veto generator.

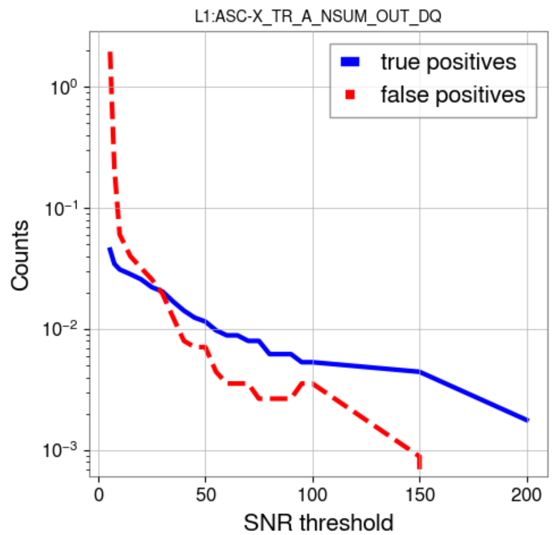

Figure 4: True Positive/False Positive versus SNR threshold for the selected good veto generator (L1:ASC-X_TR_A_NSUM_OUT_DQ) applied on the *evaluation* dataset. The TPR surpasses the FPR approximately around an SNR value of 25. This suggests that this channel is a good witness.

3.2.3 Co-clustering

For the real-data, we do not have information about the direct association between glitch events. To obtain information about the performance of both NMF and BMF, we calculate the glitch class coverage for the *training* dataset. We then calculate homogeneity to determine whether the glitch classes are considered "true" glitch classes. We want to examine whether a given glitch class is witnessed by the same channels or by multiple sets of channels that differ at each instance. This would allow us to examine whether or not the glitch classes have different sources despite the similarities in morphological structure presented. Figure 5 plots the coverage and homogeneity for the *training* datsaset across 10 runs of the factorization at different ranks $K \in [5, 50]$. We find that both models have increasing coverage as K increases, which matches the observed behavior in the simulated data. On the other hand, the homogeneity values are generally lower for both models across different K values we sampled. Compared to the highly homogeneous sample for the simulated dataset, the real dataset presents as heterogeneous, which indicates that despite visual similarities in their manifestation within the main channel, the different Gravity Spy classes, on average, do not have a consistent representation within the auxiliary channel space.

Using co-clustering, the method we present in this work can also determine subsets of channel witnesses to subsets of glitches of a given glitch classification. As a proof-of-concept, we present our analysis on the Tomte glitch which is the dominant glitch classification for events within our dataset. We examined columns in the glitch matrices generated by both NMF and BMF and selected columns where the majority label is Tomte ($> 50\%$). We examined the corresponding columns in the channel matrices to select for the channels that witnessed the glitch events captured by the factors of the glitch matrix. We performed this analysis across 10 runs on the simulated data. We obtained a list of channels across the 10 runs. This list of channels, according to both NMF and BMF, witness the glitch events captured within

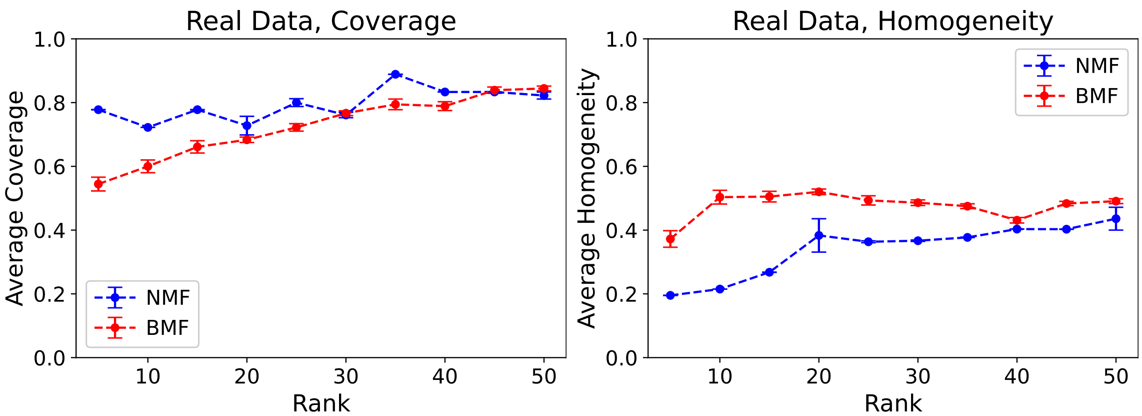

Figure 5: *Left*: Plot of the average coverage for the real dataset with the 1σ error across different factorization ranks K using NMF (red) and BMF (blue). Each point is the average across 10 runs on the *training* dataset. We find good coverage for both NMF and BMF as rank K increases. *Right*: Plot of the average homogeneity for the real dataset with the 1σ error across different factorization ranks K using NMF (red) and BMF (blue). Each point is the average across 10 runs. The results indicate that the glitch classes within the real dataset are heterogeneous. Furthermore, NMF recovers lower homogeneity values which suggests that a feature-based model versus a binary model results in more heterogeneous clusters.

the Tomte-dominated columns. We take the intersection of the channel lists generated by the two models. We find three channels selected by both NMF and BMF across the 10 runs on the *training* dataset. The sets of channels are {L1:HPI-HAM3_BLND_L4C_RY_IN1_DQ, L1:HPI-HAM3_BLND_L4C_VP_IN1_DQ, L1:HPI-HAM4_BLND_L4C_RZ_IN1_DQ}. Using the criteria for a "good" veto generator, we do not find these three channels to be good witnesses. More analysis on other glitch classes and using other sets of auxiliary channels is reserved for future work.

4 Conclusion

In this work, we explored BMF as a model for classifying and clustering glitches. We find that despite binarization, BMF remains comparable to NMF in finding veto generators. But the pipeline we explored, could also go beyond finding witnesses. More specifically, we can use the pipeline to perform co-clustering to determine associations between glitch events and subsets of auxiliary channels. Using this technique, we explored simulated and real data. We find that applied to simulated data, where the glitch classes are orthogonal, which corresponds to non-overlapping channel sets assigned to glitch classes, we find high glitch class coverage and homogeneity for both models. This suggests that the classes within the simulated data are considered to be "pure" classes. On the other hand, the real data presents lower homogeneity which indicates that the glitch classes are not as pure. This implies that despite visual similarities manifested within the main channel as determined by the Gravity Spy Catalog, the glitch classes do not have a simple representation within the auxiliary channel space.

The pipeline we explored is robust in its ability to extract information about glitch events and their witness channels. This information can be relayed to domain experts who can localize and fix the problem at their source. Furthermore, the pipeline is flexible for more future explorations such as:

- *Determining whether a Gravity Spy glitch class is a true class.* Despite different glitch events being classified under the same Gravity Spy class, our results indicate that these classes are not "true" classes, in the sense that despite visual similarity within the main channel, the glitch events are derived from different sources. Hence, an avenue of exploration is to examine each individual glitch class to determine whether the specific class is witness by a single set of channels or by multiple sets of channels.

- *Exploring other factorization models.* One such example is using Coupled Matrix-Tensor factorization to approach a glitch morphology-based analysis. To do this, we can utilize the 1-1 correspondence between the data matrix or tensor and the matrix that contains information about the trigger, in this case, the gravity spy label matrix. As a result, we can express the "coupled" data as a factorization, where the coupled mode shares the same latent factor variable. This would enable joint clustering of the two-datasets and the structure of the labeled dataset will (ideally) influence the extracted factors in such a way that triggers that cluster within the same latent factor will be morphologically similar, on the basis of the information that the second matrix is providing. By doing so, our aim is to discover morphologically coherent glitches and identify potential Gravity spy glitches which consistently exhibit the same behavior in the auxiliary channel space. Hence, we can potentially point to a consistent mechanism which generates glitches which can help domain experts localize and possibly fix the problem at the source.

- *Exploring using more channels for the analysis.* The auxiliary channel space contains a large amount of channels $O(10^5)$. For this work, we only explored 803 safe channels. An avenue of exploration is using other safe channels to determine whether they are able to better witness glitch events.

- *Exploring using other features of the triggers.* Previous work by [2] utilized frequency during the peak SNR as a feature mode. The pipeline allows for explorations of other features such as bandwith, amplitude, duration, and more.

Acknowledgements

L.B. was supported by the LIGO SURF program. The LIGO SURF program is funded by the National Science Foundation. L.B. would also like to thank Prof. Vagelis Papalexakis, Rutuja Gurav, Prof. Jon Richardson, mentors, LIGO SURF program, Caltech, UCR, and fellow students for their support in this program.

L.B. utilized the LDG and LIGO Compute Resources for this project.

References

[1] B. P. Abbott et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.*, 116(6):061102, 2016.

[2] R. Gurav, B. Barish, G. Vajente, and E. Papalexakis. Unsupervised matrix and tensor factorization for LIGO glitch identification using auxiliary channels. *Association for the Advancement of Artificial Intelligence*, 2020.

[3] Joshua R. Smith, Thomas Abbott, Eiichi Hirose, Nicolas Leroy, Duncan MacLeod, Jessica McIver, Peter Saulson, and Peter Shawhan. A hierarchical method for vetoing noise transients in gravitational-wave detectors. *Classical and Quantum Gravity*, 28(23):235005, December 2011.

[4] Tomoki Isogai. Used percentage veto for LIGO and virgo binary inspiral searches. *J. Phys. Conf. Ser.*, 243:012005, 2010.

[5] Reed Essick, Patrick Godwin, Chad Hanna, Lindy Blackburn, and Erik Katsavounidis. iDQ: Statistical Inference of Non-Gaussian Noise with Auxiliary Degrees of Freedom in Gravitational-Wave Detectors. *arXiv e-prints*, page arXiv:2005.12761, May 2020.

[6] Pauli Miettinen and Stefan Neumann. Recent Developments in Boolean Matrix Factorization. *arXiv e-prints*, page arXiv:2012.03127, December 2020.

[7] Siamak Ravanbakhsh, Barnabas Poczos, and Russell Greiner. Boolean Matrix Factorization and Noisy Completion via Message Passing. *arXiv e-prints*, page arXiv:1509.08535, September 2015.