Identifying Witnesses to Noise Transients in Ground-based Gravitational-wave Observations using Auxiliary Channels with Matrix and Tensor Factorization Techniques

Rutuja Gurav[1*] **Evangelos E. Papalexakis**[1]
Barry C. Barish[2] **Jonathan Richardson**[2] **Gabriele Vajente**[3]
[1]UC Riverside (Computer Science and Engineering)
[2]UC Riverside (Physics and Astronomy)
[3]LIGO, Caltech
rutuja.gurav@email.ucr.edu
{epapalex, barry.barish, jonathan.richardson}@ucr.edu
gvajente@ligo.caltech.edu

Abstract

Ground-based gravitational-wave (GW) detectors are a frontier large-scale experiment in experimental astrophysics. Given the elusive nature of GWs, the ground-based detectors have complex interacting systems made up of exquisitely sensitive instruments which makes them susceptible to terrestrial noise sources. As these noise transients - termed as *glitches* - appear in the detector's main data channel, they can mask or mimic real GW signals resulting in false alarms in the detection pipelines. Given their high rate of occurrence compared to astrophysical signals, it is vital to examine these glitches and probe their origin in the detector's environment and instruments in order to possibly eliminate them from the science data. In this paper we present a tensor factorization-based data mining approach, based on irregular tensor mining, to finding witness events to these glitches in the network of heterogeneous sensors that monitor the detectors and build a catalog which can aid human operators in diagnosing the sources of these noise transients.

1 Introduction

Noise coupling into a large-scale, complex instrument's main channel(s) is a typical issue in big scientific experiments that push the limits of technology towards ground-breaking discoveries. To tackle this issue, along with constant engineering improvements, these complex frontier experiments deploy a large array of monitoring systems for day-to-day diagnostics and calibration, producing vast quantities of raw data-streams about the overall state of the detectors. This raw data, if strategically mined for latent patterns of interest, can potentially point to origins of noise couplings and aid the operators in their diagnostics. The raw data is often heterogeneous temporal data from which we need to engineer datasets suitable for data mining and the authors of this paper wish to explore the end-to-end matrix and tensor factorization based data mining pipeline in the context of complex scientific instruments like the ground-based gravitational wave detectors at LIGO.

The ground-based gravitational-wave (GW) detectors like advanced LIGO Aasi et al. [2015] are one such frontier complex instrument that have successfully reached the state-of-the-art sensitivity needed to detect astrophysical signals. Given their exquisite sensitivity, these detectors are plagued by various sources of transient terrestrial noise which affects the searches for GWs. Glitches are non-Gaussian noise transients appearing in the **main channel** of the detector which measures the

NeurIPS 2022 AI for Science Workshop.

amount of strain[1], $h(t)$, produced by a passing gravitational wave. Their origins are environmental and instrumental in nature. These noise transients stand out from the expected stationary Gaussian noise as transient power excesses which trigger GW search pipelines leading to false alarms and, along with the stationary noise, limit the detector's sensitivity. Hence, glitch characterization is a crucial problem and has been studied at LIGO in two important ways - 1. the time-frequency morphology of the glitches in the main channel, 2. presence of triggers in auxiliary diagnostic channels that are coincident with the glitches.

Glitch's morphology in the main channel Glitches have complex morphologies and vary in duration and frequency. Therefore, LIGO scientists have been studying the glitches based on the morphological differences evident in their time-frequency spectrograms. Several different classes of glitches have been identified and given their diversity and abundance, a morphology-based glitch catalog, called the Gravity Spy catalog Zevin et al. [2017] Bahaadini et al. [2018] Glanzer et al. [2022], has been built using citizen scientists to label glitch images and machine-learning based image classification techniques. Although this catalog of glitches based on their morphological characteristics has been meticulously built, it is important to find origins of these noise transients in the system and build a richer, more actionable catalog which can equip the operators to locate sources of these noise transients, understand coupling mechanisms and possibly eliminate these glitches with instrumental upgrades.

Auxiliary channels Apart from the main channel which measures the strain $h(t)$, LIGO maintains $O(10^5)$ auxiliary channels to record the state of the instrument and its environment using a wide variety of sensors, some of which are used in calibration of the detector. Some of these channels may witness noise sources induce a glitch. Finding correlations between glitches in the main channel and excess power events in the auxiliary channels is therefore useful in determining the origins of these glitches. Our contribution is a data mining approach for unsupervised analysis of glitches in the main channel and the triggers in the auxiliary diagnostic channels using *matrix and tensor factorization*.

We provide a brief background on LIGO operations in section 2 followed by some existing related work and motivate the prevalence of matrix and tensor factorization techniques for a variety of real-world applications. We describe the end-to-end data mining pipeline in Section 3 beginning with how we performed data collection in Section 3.1, how we did feature engineering in Section 3.2 to create datasets as described in Section 3.3. Section 4 describes the analysis results and Section 5 concludes the paper with some future directions for this line of work.

2 Background

Gravitational-wave Astronomy with LIGO LIGO detectors are state-of-the-art laser interferometers, which, in absence of a gravitational wave are designed to register no signal at the output photo-diode. When a gravitational wave passes through the earth, a relative change in arm lengths is induced which is measured in terms of strain $h(t)$, which is the main data product of LIGO. Unfortunately, however, transient terrestrial disturbances originating in the environment or the instrument can also cause a brief power excess and result in a signal being registered at the output photo-diode. This coupling of noise into the main channel, $h(t)$, is, in essence, a glitch. LIGO data analysts use the auxiliary channels as *veto generators* to remove time segments contaminated with glitches and generate clean segments of $h(t)$ that are then searched for GW signals. Thus, as described in Essick et al. [2013] it is useful to determine important auxiliary channels which can act as *veto generators*. Given the large number of auxiliary channels, only some of which may witness a glitch, it is conducive to use machine learning-based techniques to find correlation between the main and auxiliary channel power excesses in an attempt to determine what type of mechanical couplings produces a certain type of glitch.

Previous Efforts at LIGO There have been several efforts at LIGO that use these auxiliary channels for glitch analysis. Notably, iDQ Essick et al. [2020] is a machine learning-based low-latency glitch prediction pipeline which uses auxiliary channels information to train a binary classifier to compute

[1]Strain is the fractional change in the distance between two measurement points due to the deformation of space-time by a passing gravitational wave. (`https://www.ligo.org/science/Publication-DataAnalysisGuide/`)

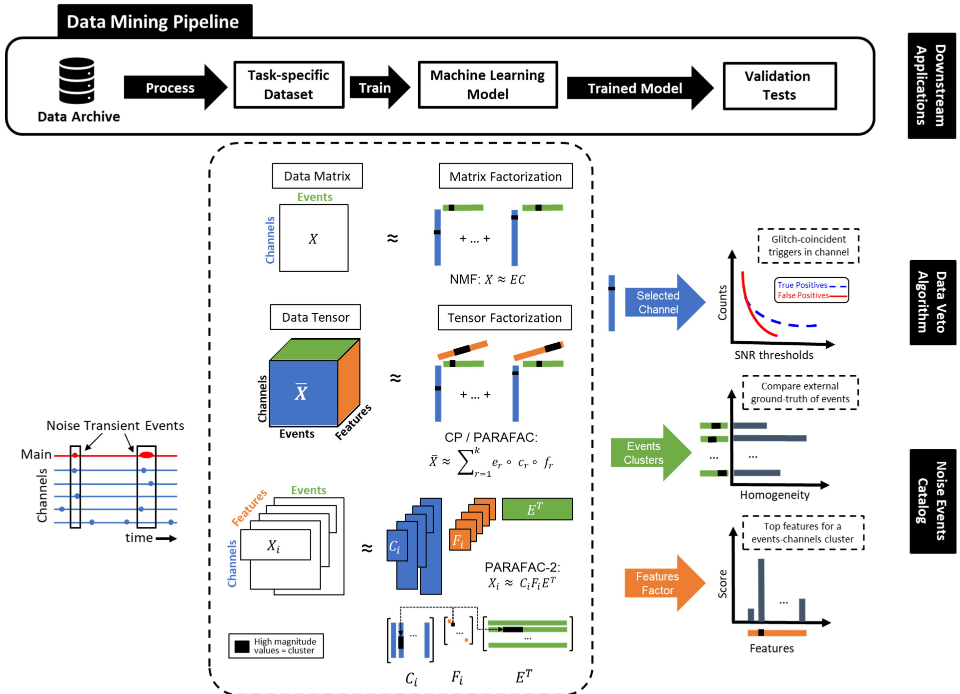

Figure 1: Proposed Pipeline

the probability of presence of glitches in the main channel. Similarly, in Cavaglia et al. [2018] the authors use auxiliary channel information and binary classification as a tool to study 2 sets of well-understood glitches from the first two observing runs of LIGO.

Data Mining using Matrix and Tensor Factorization Techniques There is a wealth of works that leverage matrix and tensor factorization especially for exploratory unsupervised data mining of multi-modal data in a wide and seemingly disparate number of real-world applications. Examples include brain network discovery Davidson et al. [2013] and Electronic Health Record mining and extraction of disease phenotypes Ho et al. [2014]. The motivation behind the choice of using matrix and tensor factorization to do these types of analysis is the computational efficiency and the simple and interpretable nature of these techniques.

3 Proposed Data Mining Pipeline

The task at hand is to find *co-clusters* of glitch events and their witness channels given a set of glitch events (G) occurring in the main channel ($h(t)$) with respect to the set of auxiliary channels (A) selected for the analysis. In this paper, we formulate this task as a soft co-clustering problem which is discussed in brief in Section 3.3. Figure 1 shows a diagrammatic representation of the complete data mining pipeline. We begin by choosing an analysis time interval in which we collect a set of glitch events that occurred in the main channel (G) and corresponding loud triggers that occurred in a set of auxiliary channels (A) around the time of the glitch events. We represent this information first as a 2-mode tensor, i.e a matrix, where each element represents the occurrence of a trigger in an auxiliary channel for each glitch event. We can later expand the matrix into a tensor to *featurize* the matrix elements with a set of features (F) related to the channels or the triggers in the channels. The constructed tensor is then factorized and the latent patterns resulting from the factors are examined using various validation tests that are specific to certain downstream applications.

3.1 Data Collection

A **channel** is typically a raw data-readout over time from a monitoring sensor (or some useful derived quantity). Each channel can be tagged with certain features (properties) like - 1. which subsystem in the complex system it belongs to (for example, a temperature sensor belonging to the physical environment monitoring subsystem or an inertial sensor measuring ground motion belonging to the internal seismic isolation subsystem), 2. where it is located in the observatory's infrastructure.[2]

A **trigger** refers to a transient power excess event in a channel - $h(t)$ or any $a(t)$ - detected by an event trigger generator (ETG) like Omicron Robinet [2016] which runs continually in real-time at LIGO during its operational runs. Thus, glitches are simply loud triggers in $h(t)$ that are not astrophysical in origin. Omicron infers certain features (properties) of the generated triggers viz. signal-to-noise ratio (SNR), peak frequency, bandwidth, amplitude, duration, phase, etc. [3]

We can do a systematic trigger collection inspired by Cavaglia et al. [2018] as follows:

1. Obtain a set, G, of loud triggers, i.e. trigger with peak SNR $>$ 7.5, that occurred in $h(t)$ between t_{start} and t_{end}.
2. For all $t_g \in G$, find triggers from a set, A, of safe auxiliary channels in a fixed window $[t_g - \gamma, t_g + \gamma]$ around each t_g where, $t_g \in G$ is the peak SNR time of the glitch trigger. γ is window parameter to center the glitch trigger around its peak SNR time. In this analysis γ ranges from 0.1 to 1 second.

If no trigger is present in the window, a default null trigger is stored for that channel corresponding to that t_g. In case of multiple loud triggers in the window, we pick the loudest trigger.

3.2 Feature Engineering and Dataset Construction

Matrix Construction: The matrix $\boldsymbol{X} \in \mathbb{R}^{|G| \times |A|}$ encodes the presence or absence of a loud trigger in each auxiliary channel for every glitch in G weighted by peak SNR. Each $|A|$-length row of $\boldsymbol{X}$ corresponds to a glitch and each element $\boldsymbol{X}(i,j)$ is the peak SNR of i_{th} trigger in j_{th} auxiliary channel if trigger present, zero otherwise.

Tensor Construction: The matrix described in the previous section can be extended into a tensor[4] to additionally encode the features (properties) of the 2 entities in the matrix i.e. the events and/or the channels. Structurally, we can create 2 types of the tensor as follows -

1. "Regular" 3-mode cuboid tensor: $\mathcal{X} \in \mathbb{R}^{|G| \times |A| \times |P|}$ where, $|P|$ = number of features. Thus, each element $\mathcal{X}(i,j,k)$ is the peak SNR of i^{th} trigger in j^{th} auxiliary channel the has the k^{th} feature (property) if trigger present, zero otherwise.
2. "Irregular" tensor / Collection of matrices: $\boldsymbol{X}_i \in \mathbb{R}^{|G| \times |A_i|}$ where, i is the i^{th} feature (property), $i = 1, ..., |P|$. Thus, each element $\boldsymbol{X}_i(m,n)$ is the peak SNR of m^{th} trigger in n^{th} auxiliary channel that has the i^{th} feature (property) if trigger present, zero otherwise.

3.3 Co-clustering glitches, channels and their features

Consider the traditional clustering problem where, a data matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ has m data-points (or observations) each of which has n features. Clustering methods, like the popular *k-means* algorithm, partition the data-points to discover k subsets called *clusters* such that data-points in a cluster are *similar* to each other and *distinct* from data-points in other clusters. Co-clustering refers to simultaneous clustering of multiple modes of the data. In case of the 2-mode data (i.e. a data matrix) $\mathbf{D}$ described above, co-clustering partitions along the n rows as well as the m columns to find subsets. Co-clustering can be formulated as a multi-linear factorization as described in Papalexakis and Sidiropoulos [2011] such that $\mathbf{D}^{m \times n} \approx [\mathbf{R}^{m \times k} \mathbf{C}^{k \times n}]$ where k is referred to as the *rank* of the factorization. In this paper, to co-cluster the 2-mode data matrix, we use Non-negative Matrix

[2]We use a list of safe auxiliary channels maintained by LIGO's Detector Characterization group for the analyses present in this paper.

[3]In this analysis, the threshold for loud triggers in $h(t)$ is set at peak SNR $\geq$ 7.5 following the example of Gravity Spy Zevin et al. [2017] which only registers triggers with peak SNR $\geq$ 7.5 as glitches in their catalog. Since we want to use the Gravity Spy glitch catalog in this work to validate our findings, we will set the same threshold for the analyses presented in this paper.

[4]A tensor in this context is a multidimensional array.

Factorization (NMF). The choice of NMF is dictated by the non-negative values (the SNRs of the triggers) in $\mathbf{X}$. Imposing the non-negativity constraint can potentially yield more interpretable factors that prove useful for further analysis. To co-cluster the 3-mode data tensors which add trigger or channel related features to the event-channel pairs, we will use variants of PARAFAC tensor factorization. We describe these factorization methods formally in the following sections.

3.3.1 Non-negative Matrix Factorization (NMF)

The formulation of NMF is stated as follows. Given $\boldsymbol{X} \in \mathbb{R}_+^{|G| \times |A|}$ and desired number of components (rank) $k << min(|G|, |A|)$, find $\boldsymbol{E} \in \mathbb{R}_+^{|G| \times k}$ and $\boldsymbol{C} \in \mathbb{R}_+^{k \times |A|}$ such that $\boldsymbol{M} \approx \boldsymbol{EC}$. In our case, each row of $\boldsymbol{E}$ and $\boldsymbol{C}^T$ is a k-length latent space representation of triggers and channels respectively. For computation of NMF, we used the implementation from Scikit-Learn library available for Python.

3.3.2 Non-negative CP/PARAFAC Tensor Factorization

In order to incorporate features of the trigger events, we can factorize $\mathcal{X} \in \mathbb{R}^{|G| \times |A| \times |P|}$ using the CP/PARAFAC tensor factorization. As depicted in figure 1, the CP/PARAFAC tensor factorization of $\mathcal{X}$ is expressed as a sum of outer products of k rank-1 tensors called components or factors. These components can be arranged as 3 factor matrices viz. $\boldsymbol{E} \in \mathbb{R}_+^{|G| \times k}$, $\boldsymbol{C} \in \mathbb{R}_+^{|A| \times k}$, and $\boldsymbol{F} \in \mathbb{R}_+^{|P| \times k}$ corresponding to each mode of $\mathcal{X}$, each with r columns. Formally stated as, $\mathcal{X} \approx \sum_{r=1}^{k} \mathbf{e}_r \circ \mathbf{c}_r \circ \mathbf{f}_r$, where $\mathbf{e}_r$, $\mathbf{c}_r$ and $\mathbf{f}_r$ are the r^{th} column in factor matrices $\boldsymbol{E}$, $\boldsymbol{C}$ and $\boldsymbol{F}$ respectively. The true rank k of the tensor is defined as the minimum number of rank-1 components required to exactly reconstruct the original tensor. However, a low-rank approximation of the tensor is of interest to our analysis since it can capture latent patterns across the modes of the tensor. In our case, the rows of factor matrices $\boldsymbol{E}$, $\boldsymbol{C}$ and $\boldsymbol{F}$ hold the k-length latent space representations of the glitch events, channels and trigger features respectively. For the computation of the PARAFAC decomposition we used the implementation from Tensorly Kossaifi et al. [2019] which uses hierarchical alternating least squares method. Additionally, we impose non-negativity constraint on the factors.

3.3.3 PARAFAC-2 Tensor Factorization

In order to incorporate features of channels, we can jointly factorize the collection of matrices $\boldsymbol{X}_i$ using the PARAFAC-2 factorization. The choice of irregular tensor mining technique like PARAFAC-2 is made instead of CP/PARAFAC for incorporating features of channels to avoid obtaining trivially orthogonal slices (since each channel belongs to exactly one subsystem and location) in the third mode of the typical 3-mode cuboid tensor. As depicted in figure 1, the PARAFAC-2 factorization of a collection of matrices where a matrix $\boldsymbol{X}_i \in \mathbb{R}^{|G| \times |A_i|}$ is a frontal slice in an *irregular* 3-mode tensor. A_i is the set of auxiliary channels in the i^{th} subsystem or at the i^{th} location. Formally stated as $\boldsymbol{X}_i \approx \boldsymbol{C}_i \boldsymbol{F}_i \boldsymbol{E}^T$ where k is the rank of the decomposition, $\boldsymbol{C}_i \in \mathbb{R}^{|A_i| \times k}$ is factor matrix corresponding to the channels belonging to the i^{th} subsystem or at the i^{th} location, $\boldsymbol{E} \in \mathbb{R}^{|G| \times k}$ is the factor matrix corresponding to the glitch events that is *shared* across all i factor matrices corresponding to channels and finally $\boldsymbol{F}_i \in \mathbb{R}^{k \times k}$ is a diagonal matrix indicating which clusters of channels in a specific subsystem or at a specific location are co-clustered with a cluster of glitches.

4 Results

For this analysis, we chose a 10-day time interval in the first half of the third observing run of LIGO (referred to as O3a by the LIGO community) Tse et al. [2019]. There were approximately 3300 loud glitches at the LIGO Hanford observatory. We used approximately 550 safe auxiliary channels obtained from a list maintained by the Detector Characterization group at LIGO. We construct the data matrix $\mathbf{X}$ described in 3.3.1, the data tensor $\mathcal{X}^{frequency}$ as described in 3.3.2 using a feature related to the trigger events i.e. their peak frequencies and finally two variants for the irregular data tensor described in, 3.3.3 using 2 sets of features related to the auxiliary channels viz. 1. a set of discrete subsystems that the channels belong to: $\mathcal{X}^{subsystems}$, 2. a set of discrete locations where the sensors corresponding to the channels are located: $\mathcal{X}^{locations}$.

Clustering glitch events The $|G|$-length vectors in $\boldsymbol{E}$ are indicators of clusters of glitchs. Given some external ground-truth labeling for the glitch events, like the Gravity Spy catalog, we can examine whether we find homogeneous clusters for the different Gravity Spy classes w.r.t. the auxiliary channels obtained via co-clustering. More specifically, to quantify ***homogeneity*** of the clusters found in the factors w.r.t. the Gravity Spy classes, we count the number of occurrences of glitches belonging to each Gravity Spy class in the top n values of each $|G|$-length vector in $\boldsymbol{E}$ and aggregate this quantity across all factors. For example, homogeneity of a co-clustering instance with rank k is defined as $\frac{1}{k}\sum_{i=1}^{k}\frac{1}{N_i}$ where N_i is the number of unique Gravity Spy classes associated with top n values of ith factor (column) in $\boldsymbol{E}$. Here, n is adaptively chosen to be 90% of the norm of the factor. Thus if we find highly homogeneous clusters for the Gravity Spy classes this quantity is close to 1. To quantify how many Gravity Spy classes a factorization instance covers, we define ***coverage*** as the fraction of unique Gravity Spy classes represented across all factors (columns) of $\boldsymbol{E}$ over the total number of classes in the dataset. We expect coverage to approach 1 as we increase the factorization rank.

Selecting veto-generator channels The $|A|$-length vectors in $\mathbf{C}$ are indicators of clusters of channels. More specifically, each vector encodes the *contribution* of a subset of channels to the corresponding subset of glitches. We used these vectors to obtain a set of candidate channels $S \subset A$, to test as *witnesses* of occurrences of glitches. We pick the channel corresponding to the maximum magnitude value in the vector, i.e. the highest contribution, and add it to S. For each channel $s_i \in S$, if there is a glitch in $h(t)$ and a corresponding trigger in s_i in a window around the glitch, we say s_i ***witnessed*** the glitch. For each channel $s_i \in S$, we count the number of glitches in $h(t)$ that it *witnessed* to test whether s_i can be used as a veto generator and remove segments of $h(t)$ before searching for real GW signals and reduce the number of false alarms.

Channels-based glitch catalog Using factor matrices $\mathbf{E}$ and $\mathbf{C}$ we can *label* each glitch event with the channels that witness it. Since each row i in $\mathbf{E}$ is an embedding of i^{th} glitch event (e_i) in the $\mathbb{R}^k$ latent space discovered by the rank-k factorization, we can choose factor j from the k available factors which has the highest magnitude value for e_i and associate the channels in the j^{th} factor of $\mathbf{C}$ with e_i.

4.1 Analysis

Figure 2 shows factors discovered by a rank-5 Non-negative CP/PARAFAC instance for the data tensor $\mathfrak{X}^{frequency}$. We can observe that the two factors (out of 5) shown in the figure have high homogeneity and find clusters of glitch events belonging to class #5 and we can find the diagnostic channels that strongly witnesses this cluster is channel #281 in the first factor and channels #367 and #369 in the second factor. We can also observe that the second factor finds higher frequency witness triggers compared to the first factor.

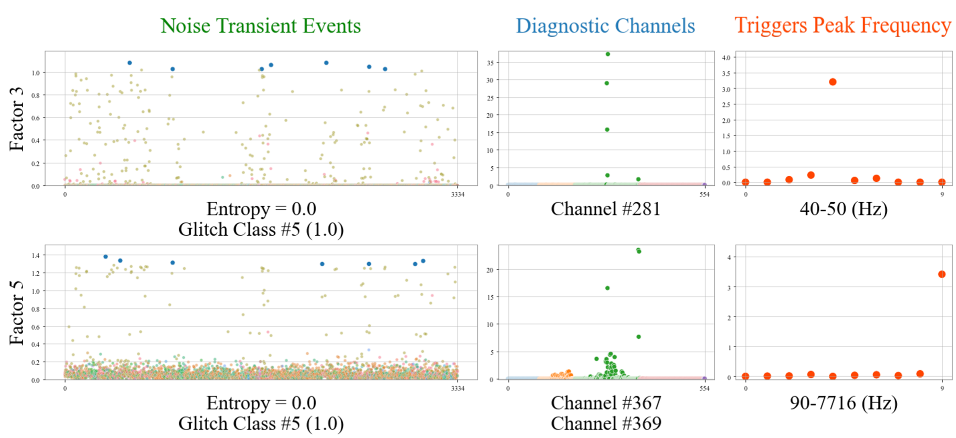

Figure 2: Factors #3, #5 of a Non-negative CP/PARAFAC tensor factorization ($rank = 5, \gamma = 0.5$) showing co-clusters of glitch events, witness channels and peak frequencies of witness triggers.

Testing homogeneity of glitch clusters Although we can examine individual factors to discover such patterns, as shown in figure 3, the overall homogeneity of factors discovered by NMF across various choices of factorization rank (k) and coincidence window sizes (γ) for the glitch events in our dataset with respect to the external ground-truth (Gravity Spy) labels is low. This suggests that morphological similarity in the time-frequency spectrograms of glitches as they appear in the main channel does not necessarily translate to having a common set of witness channels for the wide variety of Gravity Spy classes. This finding matches the one in Gurav et al. [2020] made about a set of glitch events in a different analysis period and w.r.t. a superset of auxiliary channels than ours.

Testing channels for veto generation A veto generator is a channel used by LIGO data analysts to iteratively remove segments of $h(t)$ that are contaminated with glitches before searching for astrophysical signals in it. If a channel has coincident triggers with $h(t)$ glitches, the data segment around the trigger time is removed from $h(t)$. Thus an ideal veto generator will only have triggers that coincide with $h(t)$ glitch events (true positives) to avoid removing valuable science data segments that don't have a coincident glitch (false positives). Figure 3 shows 2 candidate witness channels selected by an NMF instance. Channel #281 can be a good veto generator as the true positive rate exceeds the false positive rate above an SNR threshold of 25. This is not true for channel #300 which is a noisy channel with high rate of spurious loud triggers.

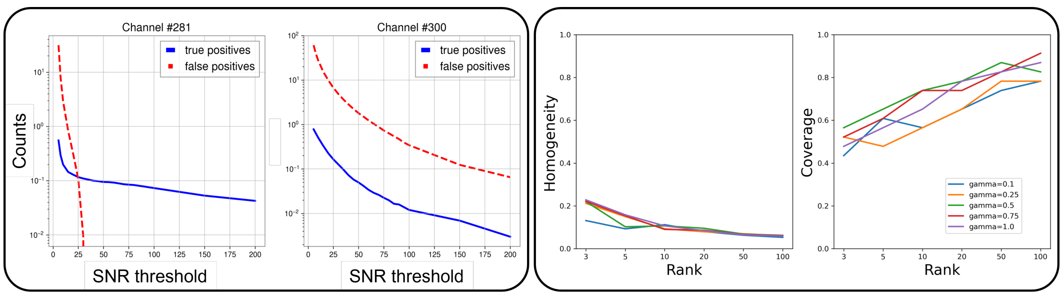

Figure 3: **(Left Panel)** True and false positive rates of 2 different witness channels. *(Left)* Channel #281 a good veto generator candidate as it has a higher true positive rate above SNR 25. *(Right)* Channel #300 is a poor veto generator as it is a highly noisy channel. **(Right Panel)** Overall homogeneity and coverage across glitch events factors discovered by NMF with respect to the external ground-truth labels available for glitches i.e. the Gravity Spy catalog. The homogeneity remains low at increasing factorization ranks as well as wider coincidence window sizes. The coverage increases with increasing factorization ranks as expected.

Building a channels-based glitch catalog Figure 4 show how we can build a glitch catalog based on witness channels. A rank 5 NMF instance selected the same 2 witness channels (out of approx. 550 available channels) for 2 different glitch events occurring in $h(t)$. We can see these two channels have loud triggers in them around the time of these glitches but the morphology of the loud triggers differ for each glitch event as evident in the spectrograms of the trigger signals. The data analysis pipeline described in this paper enables us to systematically label each glitch event with a small subset of channels that witness it and further examine the morphology of the triggers in the witness channels. Thus far the morphology of only the glitches in the main channel ($h(t)$) have been exhaustively studied and cataloged by Gravity Spy since doing so for hundreds of auxiliary diagnostic channels is not feasible. The proposed data analysis pipeline addresses this feasibility issue by systematically clustering the glitch events w.r.t. their witness channels instead of resorting to a brute force search.

5 Conclusions and Future Directions

In this paper, we present an end-to-end data mining pipeline using unsupervised matrix and tensor factorization techniques to build a more comprehensive, data-driven catalog of noise transient events occurring in large-scale, complex scientific instruments like the ground-based gravitational wave detectors at LIGO. Such catalogs can aid operators to study sources of noise events, investigate mechanisms through which noise gets coupled into the detector's highly sensitive main channel and ultimately push the detectors towards their design sensitivity to achieve the scientific goal of observing

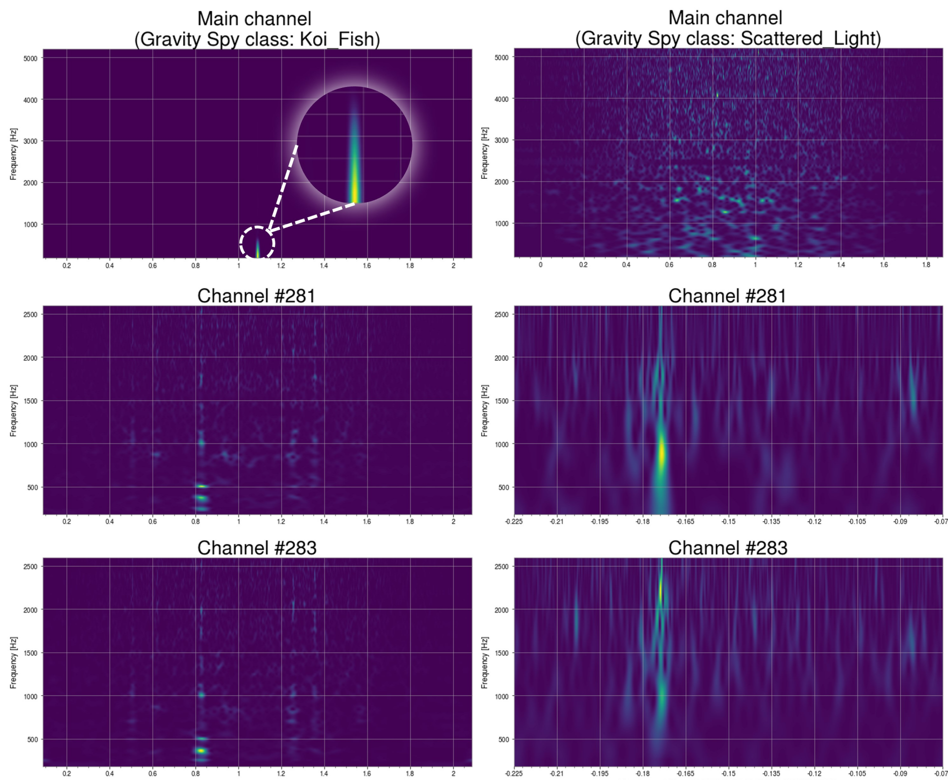

Figure 4: *(column 1, row 1)* A glitch occurrence in $h(t)$; *(column 1, rows 2,3)* the two channels selected by an NMF instance clearly show triggers in them around the time of the glitch. *(column 2, row 1)* Another glitch occurrence in $h(t)$ that was also witnessed by the same two channels *(column 2, rows 2,3)* but the triggers in the witness channels have different time-frequency morphologies.

larger volumes of spacetime to detect gravitational waves emanating from merging black holes and neutron stars. The procedures described to build useful datasets using hundreds of raw data-streams from a complex instrument's infrastructure and the unsupervised machine learning techniques used to systematically mine this data are fairly generic and can be generalized to any similarly large-scale complex scientific instruments that are becoming prevalent in modern experimental physics. The authors of this paper will test the proposed methods to build comprehensive catalog of noise events occurring in LIGO data at scale and extend the utility of this catalog for discovering and modelling the mechanisms of noise couplings in order to subtract them from the science data.

6 Acknowledgements

The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459. Research at UCR was supported by the National Science Foundation under EAGER grant no. PHY 2141072 and CAREER grant no. IIS 2046086, and by a UCR Regents Faculty Development Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding party.

References

Junaid Aasi, BP Abbott, Richard Abbott, Thomas Abbott, MR Abernathy, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, et al. Advanced ligo. *Classical and quantum gravity*, 32(7):074001, 2015.

Sara Bahaadini, Vahid Noroozi, Neda Rohani, Scott Coughlin, Michael Zevin, JR Smith, Vicky Kalogera, and A Katsaggelos. Machine learning for gravity spy: Glitch classification and dataset. *Information Sciences*, 444:172–186, 2018. URL `https://zenodo.org/record/1476156#.XNr-3Y5KhPY`.

Marco Cavaglia, Kai Staats, and Teerth Gill. Finding the origin of noise transients in ligo data with machine learning. *arXiv preprint arXiv:1812.05225*, 2018.

Ian Davidson, Sean Gilpin, Owen Carmichael, and Peter Walker. Network discovery via constrained tensor analysis of fmri data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 194–202, 2013.

Reed Essick, Lindy Blackburn, and Erik Katsavounidis. Optimizing vetoes for gravitational-wave transient searches. *Classical and Quantum Gravity*, 30(15):155010, 2013.

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 preprint arXiv:2005.12761*, 2020.

J Glanzer, S Banagiri, SB Coughlin, S Soni, M Zevin, CPL 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. *arXiv preprint arXiv:2208.12849*, 2022.

Rutuja Gurav, Barry C. Barish, Gabriele Vajente, and Evangelos E. Papalexakis. Unsupervised matrix and tensor factorization for ligo glitch identification using auxiliary channels. 2020.

Joyce C Ho, Joydeep Ghosh, and Jimeng Sun. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 115–124, 2014.

Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly: Tensor learning in python. *The Journal of Machine Learning Research*, 20(1):925–930, 2019.

Evangelos E Papalexakis and Nicholas D Sidiropoulos. Co-clustering as multilinear decomposition with sparse latent factors. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2064–2067. IEEE, 2011.

Florent Robinet. Omicron: an algorithm to detect and characterize transient events in gravitational-wave detectors, 2016.

M Tse, Haocun Yu, N Kijbunchoo, A Fernandez-Galiana, P Dupej, L Barsotti, CD Blair, DD Brown, SE Dwyer, A Effler, et al. Quantum-enhanced advanced ligo detectors in the era of gravitational-wave astronomy. *Physical Review Letters*, 123(23):231107, 2019.

Michael Zevin, Scott Coughlin, Sara Bahaadini, Emre Besler, Neda Rohani, Sarah Allen, Miriam Cabero, Kevin Crowston, Aggelos K Katsaggelos, Shane L Larson, et al. Gravity spy: integrating advanced ligo detector characterization, machine learning, and citizen science. *Classical and Quantum Gravity*, 34(6):064003, 2017.