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

Technical Note **LIGO-T2200207–v1** 2022/07/30

Interim Report 2: 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/

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 Objectives

In this work, we aim to utilize the proposed method by [2] to explore various models (e.g., factorization techniques, clustering methods) to compare with existing results using established metrics such as precision and recall. More specifically, the goals are to find sets of auxiliary channels which may serve as veto generators and to use the techniques to cluster 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 will use the time-dependent data recorded through the main channel and the auxiliary channels. Loud triggers defined to be recorded signals that have SNR ≥ 7.5, which may include glitches, are converted into a data matrix $Z \in \mathbb{R}^{|G| \times |A|}$ and a 3-mode tensor $\chi \in \mathbb{R}^{|G| \times |A| \times |F|}$ through an established pipeline, 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 this information, we can use the pipeline established by [2] to test different clustering and factorization models. For this work, we will approach the problem in different ways. First, we will test Boolean Matrix Factorization (BMF) as a factorization model. BMF presents an advantage in which the presence of glitches is treated in a binary fashion within the data matrix. As a result, the issue of setting thresholds during the selection of witness channels post-factorization is mitigated.

Second, we will also use Coupled Matrix-Tensor Factorization (CMTF) to discover morphologically coherent glitches. This will allow us to investigate whether the introduction of glitch morphology information (e.g., labels) can lead to the discovery of patterns within glitch classes that are linked to similarities in morphology and behaviors within the auxiliary channel space.

2.1 Boolean Matrix Factorization (BMF)

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 our 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 [6].

2.2 Glitch Morphology Analysis

We aim to use Coupled Matrix-Tensor Factorization (CMTF) to approach a glitch morphology-based analysis. To do this, we 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.

Using CMTF, 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. By doing so, 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.

3 Progress

3.1 Data Collection

Using the pipeline established by [2], we are currently collecting data using the Omicron trigger catalog from the end of O3a and the beginning of O3b runs of the Livingston detector. More specifically, we will use data collected between September 25-28, 2019 for the O3a *training* interval and September 29-30, 2019 for the *validation* interval. For the O3b runs, we will collect data between November 1-4, 2019 for the *training* interval and November 5-6, 2019 for the *validation* interval. The goal of collecting data from the two different periods is to compare glitches before and after LIGO was able to mitigate noise derived from scattering [7].

3.2 Synthetic Data Generation

To approach our problem, we first established a synthetic data generator. The generator aims to mimic the data matrix obtained from LIGO through the pipeline established by [2]. There are two reasons for creating a generator. One is the time-intensive nature of data collection. Hence, creating a generator will allow us to "crash-test" the methods and algorithms we propose to explore. More importantly, a synthetic data generator can be used 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.

The first step of the synthetic data generation is generating a glitch catalog that emulates the Gravity Spy Catalog. To do this, we generate 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 can 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 orthogonal. We explore these different methods within this work.

Once the catalog is generated, it can be used 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, a Poisson-distributed "background" noise described by λ_b is randomly generated. Following this, 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. More explicitly, we can select a "strong" glitch class, which is generated at N times the rate, in comparison to other glitch classes.

Each of the channels associated with a glitch class can be mapped directly to the columns of the data matrix. Using a Poisson-distributed signal described by λ_g, a glitch is generated at the column. Note that 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.

3.3 Testing BMF

We utilized the BMF implementation from `https://github.com/mravanba/BooleanFactorization`, which utilizes posterior inference via message passing to perform the factorization [8]. To allow for convergence, we set the maximum number of iterations to 1000.

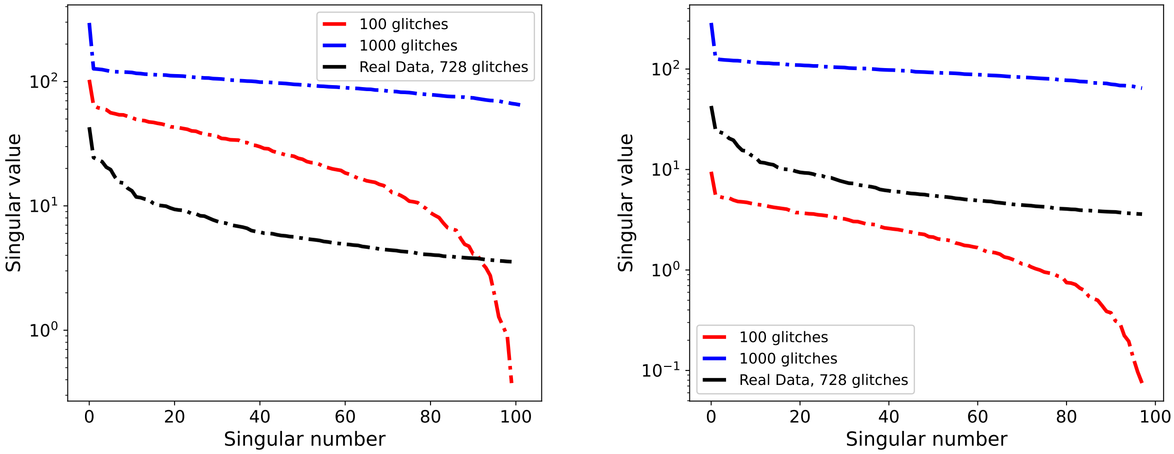

Figure 1: Singular values plot across four different Boolean data matrices. (*Upper*): Singular values plot of the data matrix generated with 100 and 1000 glitches sourced from a glitch catalog that had 50% of glitch classes share one channel. The *upper* plot represents a scenario where different glitch 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.

3.3.1 Simulated Data

As a first pass, we test the factorization model on simulated data. We generated a glitch catalog in which 50% of the glitch classes have three different sets of channels in which they can arise from. The rest of the glitch classes are independent. Moreover, the whole glitch catalog is described by orthogonal channel sets, which means that there is no overlap among the channels associated with a given glitch class. In later steps, we plan to test the factorization on datasets with non-orthogonal channel sets. More specifically, this corresponds to cases where different glitch classes can have channel overlaps. 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 under the assumption that a subset of channels are not involved in glitch generation.

To perform an exploration on the co-clustering capabilities of BMF, we generated data matrices with 2000 glitches and 200 channels, which provides a comparable size to real data matrices collected over a 3-day period. We factored the generated data matrix Z to $\approx XY$, where X is the matrix associated with glitches and Y is the matrix associated with the channels. We performed a grid-search across different factorization ranks $K \in [1, 30]$. Note that each row of X and Y represents a K-length latent space representation of glitches and channels, respectively. By sampling across different ranks, we can find the most optimal rank using our established metrics. Furthermore, the most optimal rank will reveal a "hidden" structure within the data set. We then calculate two parameters that provide insights as to how well BMF co-clusters glitches to channels: channel coverage and channel precision. 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 (108 in our case) involved with a glitch class are simulated. Hence, coverage is defined as $N_{c,rec}/108$ for this particular set of simulations. Figure 2 plots channel coverage for a single simulated data matrix factorized across different ranks $K \in [1, 30]$. Overall, channel coverage using the BMF model greatly improves as the rank of the decomposition increases which is indicative that more of the "hidden" structure is revealed at higher ranks. Comparatively, using the NMF model also demonstrates an improvement in channel coverage at higher ranks. Comparing the two models, it appears the NMF provides better channel coverage at lower ranks $K < 16$, while, in the high-rank regime, BMF overtakes the channel coverage of NMF.

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 average across all K-columns within a decomposition. Figure 2 plots channel precision for a simulated data matrix factorized across different ranks $K \in [1, 21]$. To a first, approximation, we do not find an increasing trend in precision for a data matrix factorized using BMF across different ranks, which suggests that there are no opportunities to gain better precision by changing the factorization rank of BMF. On the other NMF, demonstrates an overall increase in precision at higher ranks which suggests that rank is a tuneable parameter in improving channel precision.

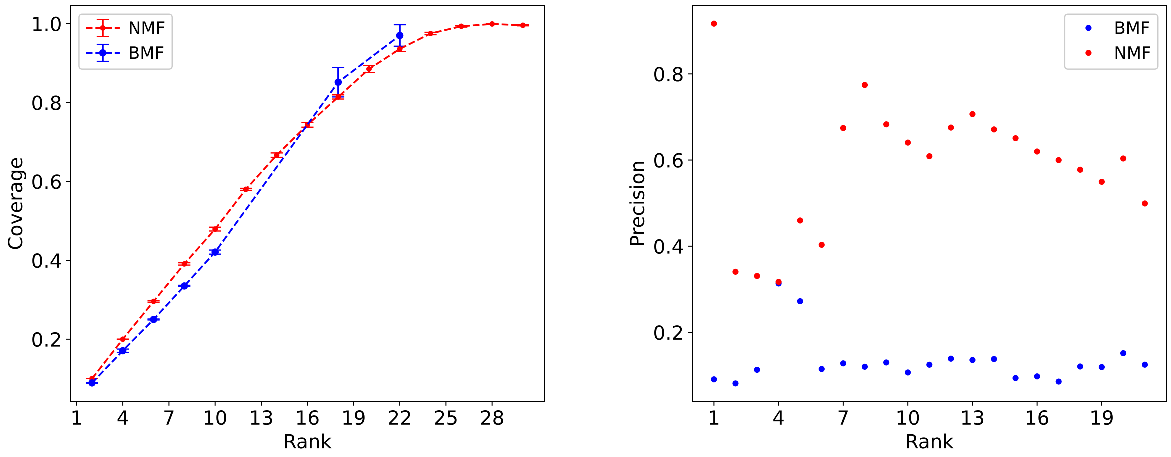

Figure 2: *Left*: Plot of the average channel coverage with the 1σ error across different factorization ranks K using NMF (red) and BMF (blue). Each point is the average across 10 simulations. The errorbar denotes represents the standard error of the mean across the 10 simulations. *Right*: Plot of the channel precision for a single realization across different factorization ranks K using NMF (red) and BMF (blue). The results indicate that for both NMF and BMF, coverage can be improved by increasing the factorization rank. On the other hand, the results demonstrate that precision remains fairly constant across different ranks for BMF, while channel precision improves at higher ranks for NMF. This suggests that, in terms of precision, rank is only a tuneable hyperparameter for NMF-based factorization models.

3.3.2 Real Data

For the real data, however, we do not have the ground truth for which gravity spy classes is associated with each channel. Furthermore, cross-talks between different channels could lead to unidentified couplings. Hence, calculating channel coverage and precision is not possible. Hence, we calculate homogeneity instead which 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). For NMF, this is defined to be the top 90% of the norm of a factor in the channel matrix factor. Note that highly homogeneous auxiliary channel clusters that correspond to gravity classes signifies a homogeneity value close to 1. As a first run, we performed NMF and BMF on the real data collected from O3b. We performed both factorizations at rank 16. We choose this rank based on the approximation that coverage for both models as seen in the simulated data is roughly equal within that rank. We find that the homogeneity for this dataset using NMF and BMF is 0.991 and 0.47, respectively.

4 Current Status and Future Plans

The current status of the project requires the calculation of multiple metrics: glitch class coverage, glitch class precision, channel coverage, and channel precision for simulated data with multiple realizations such that we can quantify the variance at each rank K. This will allow to better constrain the ideal selection of rank that will yield the "best" factorization. Furthermore, we plan to try different models such as deflation, in which we examine the different linearly independent columns of the factor matrices. Then, we choose a column or eigenvector to eliminate by setting the elements to zero. By doing so, our aim to see if we can disentangle the influence of different factors in the glitch and channel matrices to see if we could decouple sets of channels and glitches that present as correlated. Moreover, in the real data, our goal is to examine homogeneity to explore the consistency of the structure of the Gravity Spy classes across the auxiliary channel space at varying ranks to better understand whether features as maintained in NMF is better at characterizing the Gravity Spy Classes, in comparison to BMF which treats the existence of loud triggers in a binary fashion.

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] S Soni, C Austin, A Effler, R M S Schofield, G Gonzalez, V V Frolov, J C Driggers, A Pele, A L Urban, G Valdes, R. Abbott, C. Adams, R. X. Adhikari, A. Ananyeva, S. Appert, K. Arai, J. S. Areeda, Y. Asali, S. M. Aston, A. M. Baer, M. Ball, S. W. Ballmer, S. Banagiri, D. Barker, L. Barsotti, J. Bartlett, B. K. Berger, J. Betzwieser, D. Bhattacharjee, G. Billingsley, S. Biscans, C. D. Blair, R. M. Blair, N. Bode, P. Booker, R. Bork, A. Bramley, A. F. Brooks, D. D. Brown, A. Buikema, C. Cahillane, K. C. Cannon, X. Chen, A. A. Ciobanu, F. Clara, S. J. Cooper, K. R. Corley, S. T. Countryman, P. B. Covas, D. C.

Coyne, L. E. H. Datrier, D. Davis, C. Di Fronzo, K. L. Dooley, P. Dupej, S. E. Dwyer, T. Etzel, M. Evans, T. M. Evans, J. Feicht, A. Fernandez-Galiana, P. Fritschel, P. Fulda, M. Fyffe, J. A. Giaime, K. D. Giardina, P. Godwin, E. Goetz, S. Gras, C. Gray, R. Gray, A. C. Green, E. K. Gustafson, R. Gustafson, J. Hanks, J. Hanson, T. Hardwick, R. K. Hasskew, M. C. Heintze, A. F. Helmling-Cornell, N. A. Holland, J. D. Jones, S. Kandhasamy, S. Karki, M. Kasprzack, K. Kawabe, N. Kijbunchoo, P. J. King, J. S. Kissel, Rahul Kumar, M. Landry, B. B. Lane, B. Lantz, M. Laxen, Y. K. Lecoeuche, J. Leviton, J. Liu, M. Lormand, A. P. Lundgren, R. Macas, M. MacInnis, D. M. Macleod, G. L. Mansell, S. Márka, Z. Márka, D. V. Martynov, K. Mason, T. J. Massinger, F. Matichard, N. Mavalvala, R. McCarthy, D. E. McClelland, S. McCormick, L. McCuller, J. McIver, T. McRae, G. Mendell, K. Merfeld, E. L. Merilh, F. Meylahn, T. Mistry, R. Mittleman, G. Moreno, C. M. Mow-Lowry, S. Mozzon, A. Mullavey, T. J. N. Nelson, P. Nguyen, L. K. Nuttall, J. Oberling, Richard J. Oram, C. Osthelder, D. J. Ottaway, H. Overmier, J. R. Palamos, W. Parker, E. Payne, R. Penhorwood, C. J. Perez, M. Pirello, H. Radkins, K. E. Ramirez, J. W. Richardson, K. Riles, N. A. Robertson, J. G. Rollins, C. L. Romel, J. H. Romie, M. P. Ross, K. Ryan, T. Sadecki, E. J. Sanchez, L. E. Sanchez, T. R. Saravanan, R. L. Savage, D. Schaetzl, R. Schnabel, E. Schwartz, D. Sellers, T. Shaffer, D. Sigg, B. J. J. Slagmolen, J. R. Smith, B. Sorazu, A. P. Spencer, K. A. Strain, L. Sun, M. J. Szczepańczyk, M. Thomas, P. Thomas, K. A. Thorne, K. Toland, C. I. Torrie, G. Traylor, M. Tse, G. Vajente, D. C. Vander-Hyde, P. J. Veitch, K. Venkateswara, G. Venugopalan, A. D. Viets, T. Vo, C. Vorvick, M. Wade, R. L. Ward, J. Warner, B. Weaver, R. Weiss, C. Whittle, B. Willke, C. C. Wipf, L. Xiao, H. Yamamoto, Hang Yu, Haocun Yu, L. Zhang, M. E. Zucker, and J. Zweizig. Reducing Scattered Light in LIGO's Third Observing Run. *arXiv e-prints*, page arXiv:2007.14876, July 2020.

[8] 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.