Published in final edited form as:
Class Quantum Gravity. 2017 ; 34(No 6): . doi:10.1088/1361-6382/aa5cea.

Gravity Spy: integrating advanced LIGO detector characterization, machine learning, and citizen science

M Zevin[1], **S Coughlin**[1], **S Bahaadini**[2], **E Besler**[2], **N Rohani**[2], **S Allen**[3], **M Cabero**[4], **K Crowston**[5], **A K Katsaggelos**[2], **S L Larson**[1,3], **T K Lee**[6], **C Lintott**[7], **T B Littenberg**[8], **A Lundgren**[4], **C Østerlund**[5], **J R Smith**[9], **L Trouille**[1,3], and **V Kalogera**[1]

[1]Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA) and Deptartment of Physics and Astronomy, Northwestern University, 2145 Sheridan Rd, Evanston, IL 60208, United States of America

[2]Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60201, United States of America

[3]Adler Planetarium, Chicago, IL 60605, United States of America

[4]Max-Planck-Institut für Gravitationsphysik, Callinstrasse 38, D-30167 Hannover, Germany

[5]School of Information Studies, Syracuse University, Syracuse, NY 13210, United States of America

[6]Department of Communication, University of Utah, Salt Lake City, UT 84112, United States of America

[7]Department of Physics, University of Oxford, Oxford, United Kingdom

[8]NASA/Marshall Space Flight Center, Huntsville, AL 35812, United States of America

[9]Department of Physics, California State University Fullerton, Fullerton, CA 92831, United States of America

Abstract

With the first direct detection of gravitational waves, the advanced laser interferometer gravitational-wave observatory (LIGO) has initiated a new field of astronomy by providing an alternative means of sensing the universe. The extreme sensitivity required to make such detections is achieved through exquisite isolation of all sensitive components of LIGO from non-gravitational-wave disturbances. Nonetheless, LIGO is still susceptible to a variety of instrumental and environmental sources of noise that contaminate the data. Of particular concern are noise features known as *glitches*, which are transient and non-Gaussian in their nature, and occur at a high enough rate so that accidental coincidence between the two LIGO detectors is non-negligible. Glitches come in a wide range of time-frequency-amplitude morphologies, with new morphologies appearing as the detector evolves. Since they can obscure or mimic true gravitational-wave signals, a robust characterization of glitches is paramount in the effort to achieve the gravitational-wave detection rates that are predicted by the design sensitivity of LIGO. This proves a daunting task for members of the LIGO Scientific Collaboration alone due to the sheer amount of data. In this paper

we describe an innovative project that combines crowdsourcing with machine learning to aid in the challenging task of categorizing all of the glitches recorded by the LIGO detectors. Through the Zooniverse platform, we engage and recruit volunteers from the public to categorize images of time-frequency representations of glitches into pre-identified morphological classes and to discover new classes that appear as the detectors evolve. In addition, machine learning algorithms are used to categorize images after being trained on human-classified examples of the morphological classes. Leveraging the strengths of both classification methods, we create a combined method with the aim of improving the efficiency and accuracy of each individual classifier. The resulting classification and characterization should help LIGO scientists to identify causes of glitches and subsequently eliminate them from the data or the detector entirely, thereby improving the rate and accuracy of gravitational-wave observations. We demonstrate these methods using a small subset of data from LIGO's first observing run.

Keywords

gravitational waves; LIGO; detector characterization; citizen science; machine learning

1. Introduction

Following a major upgrade, advanced LIGO completed its first observing run (O1), which spanned from September 12, 2015 through January 19, 2016 [1]. During this run, the LIGO detectors made the first direct detection of gravitational waves and the first observations of binary black hole coalescences [2–4]. With these detections, LIGO has initiated a new field of astronomy by providing an alternative means of sensing the universe. Over the coming years, the increased sensitivity of the LIGO detectors and additional interferometers joining the network of gravitational-wave observatories [5–7] will further increase sensitivity to the gravitational universe.

In order to detect gravitational waves, LIGO requires sensitivity to length fluctuations a thousandth the diameter of a proton in the 4 km detector arms. In future observing runs this sensitivity will further increase; at design sensitivity LIGO aims to have the ability of detecting neutron star–neutron star mergers up to a distance of 200 Mpc [8]. This high sensitivity is achieved through exquisite isolation of the lasers, mirrors, and all sensitive components of LIGO from non-gravitational-wave disturbances [9, 10]. However, LIGO detectors are still susceptible to non-cosmic disturbances that cause noticeable signals in the detectors. The effort to identify, characterize, and separate sources of noise from cosmic signals is paramount in achieving LIGO sensitivity goals [11].

Of particular concern are transient, non-Gaussian noise features known as *glitches*. Glitches are instrumental or environmental in nature (caused by e.g. small ground motions, ringing of the test-mass suspension system at resonant frequencies, or fluctuations in the laser) and come in a wide variety of time-frequency-amplitude morphologies. These artifacts can produce false-positive results in gravitational-wave searches, reduce the significance of candidate gravitational-wave signals, corrupt data, bias astrophysical parameter estimation, and reduce the amount of analyzable data. The sensitivity of searches for unmodeled

gravitational waves are especially limited by the high rate of glitches in LIGO [11, 12]. In the 51.5 d of O1 alone, approximately 10^6 glitches over a minimum signal-to-noise ratio (SNR) threshold of 6 were recorded. To maximize the gravitational-wave detection rate, the causes of glitches must be identified and fixed within the detectors (in the best case) or glitches must be removed from the data set. Identifying how many different glitches have a similar morphology is an important first step to this, allowing prioritizing by number and characteristics. Therefore, it is necessary to develop robust methods to identify and characterize glitches.

Teaching computers to identify and morphologically classify glitches in detector data is a challenge. Only a small number of glitch classes have been understood to the level where they could be removed from the data with confidence. Attempts to use machine learning algorithms have shown promise in glitch classification endeavors [13–17], however these techniques do not yet capture the full range of glitch morphologies present in LIGO data. Though human ability to recognize patterns is a proven tool for such diverse classification endeavors, though the high volume of data that LIGO streams would easily overwhelm any small group of scientists.

To address this challenge, we have developed *Gravity Spy*—an interdisciplinary project that will leverage the strengths of both humans and computers to create a superior classifier of glitches in LIGO data. Gravity Spy addresses this task through the convergence of four science areas: gravitational physics, human-centered computing, machine learning, and citizen science. Specifically, the goal of the project is to leverage the advantages of citizen science along with those of machine- and human-learning techniques to design a socio-computational system with which to analyze and characterize LIGO glitches and improve the effectiveness of gravitational-wave searches. Gravity Spy also complements current glitch classification techniques, as it scales with an increasing number of unique glitch classes and continually bolsters labeled sets of pre-existing classes. Furthermore, the human classification aspect of the project acts to readily identify new categories of glitches that arise as the detectors evolve.

The Gravity Spy project couples human classification with machine learning models in a symbiotic relationship: volunteers provide large, labeled sets of known glitches to train machine learning algorithms and identify new glitch categories, while machine learning algorithms ‘learn’ from the volunteer classifications, rapidly classify the entire dataset of glitches, and guides how information is provided back to participants. The Gravity Spy project includes research on the human-centered computing aspects of this socio-computational system, as empirical testing of the human-computer interface leads to better project design and an enhanced performance of citizen science volunteers. Gravity Spy is implemented through Zooniverse.org, the leading online platform for citizen science, which has fielded a workable crowdsourcing model. Currently, over 1.5 million ‘citizen scientists’ work to provide analyses of scientific data on more than 40 projects [18]. A beta version of Gravity Spy has already resulted in the identification of new glitch morphological classes, and shows promise for helping to improve LIGO data quality during upcoming observing runs.

In this paper, we summarize the impact of glitches on LIGO data analysis and current efforts to mitigate their effects (section 2). We then discuss the Gravity Spy project in full (section 3), highlighting in particular data preparation for the project (3.1), the citizen science interface (3.2), machine learning algorithms used for image classification and crowdsourcing classifiers (3.3), and social science experiments for the socio-computational system (3.4). Next we discuss preliminary results of the project using data from the first LIGO observing run (section 4). Lastly, we comment on future prospects for the Gravity Spy project and its role in LIGO detector characterization (section 5).

2. Characterization of transient noise in LIGO

2.1. Impact of glitches on gravitational-wave data analysis

Searches for transient gravitational-wave signals, especially those that are short duration, in LIGO's sensitive frequency band, and/or poorly modeled [11], are highly susceptible to glitches in the data. One method for mitigating the impact of glitches is the requirement of coincidence between the LIGO observatories, which are located in Hanford, Washington and Livingston, Louisiana. Gravitational waves would appear in both detectors separated in time by less than or equal to the light travel time between the observatories. If a signal appears in only one observatory during this time window, it is rejected. Furthermore, most searches for generic transient events require some kind of signal consistency (e.g. coherence) to limit the impact of glitches on search pipelines. Despite these requirements, glitches occur at a high enough rate and with common enough morphology that accidental coincidence and coherence between the two detectors is non-negligible.

Glitches impact LIGO data analysis efforts in three critical ways. First, they increase the loudness of the background in gravitational-wave searches, which reduces the significance of candidate events. Even searches that utilize signal models to create discriminating signal statistics (e.g. compact binary coalescence searches [19, 20]) are afflicted by glitch occurrences. Second, glitches impact the recovery of astrophysical parameters from a gravitational wave source [4, 21, 22], since glitches that occur near the same time as a gravitational-wave signal reduce the SNR of the event and lead to broader uncertainties in parameter estimation. Finally, glitches reduce the amount of usable data. While data 'vetoes' can be constructed for times when glitches are known to occur, they eliminate the data available to be searched for astrophysical signals. Therefore, identifying the cause of glitches and eliminating the source of the glitch is much preferred to constructing such vetoes. The negative effects of glitches on data analysis make the identification and mitigation of glitches an essential part of the LIGO science effort.

2.2. Identifying glitches

Several categories of glitches have been identified by the LIGO Scientific Collaboration (LSC), grouped by common origin and/or similar morphological characteristics [15–17]. Some of these categories have known causes, while others have causes yet to be identified. For example, two common morphological classes of glitches are shown in figure 1. Blip glitches figure 1(a) are caused by unknown processes, whereas whistle glitches figure 1(b)

are caused by radio signals at megahertz frequencies that beat with voltage controlled oscillators in the interferometer control system [23].

Techniques have been developed to identify and categorize some categories of glitches automatically. Identification algorithms search for excess power in the time-frequency space of LIGO strain data and in hundreds of auxiliary channels, which are insensitive to gravitational waves and monitor the many instrumental and environmental factors potentially affecting the detectors. In addition to identifying a glitch, these algorithms parameterize glitches according to their time, frequency, SNR, and duration, among other parameters [24, 25]. Current approaches also search for statistical correlation between glitches in the gravitational-wave strain data channel and triggers in auxiliary channels [26–29]. However, due to the sheer volume of data, the LSC has not yet been able to filter through the millions of glitches to create a comprehensive categorization.

2.3. Mitigating glitches

Having identified a glitch, the goal is to eliminate it from the detector. If the root cause of a glitch cannot be determined or its source cannot be fixed, information from glitch identification algorithms can be used to create data vetoes. Such vetoes improve gravitational-wave searches by removing times strongly affected by noise transients.

Even these efforts, however, suffer from problems stemming from the very large number of glitches and their variety of morphologies. First, automated glitch classification algorithms have been unable to capture the varied morphological characteristics of all unique classes of glitches. In addition, certain types of glitches come and go over the course of an observing run, making their discovery challenging even for members of the LIGO science team. Finally, the software which implements data quality vetoes would benefit from being fed information from specific categories of glitches instead of entire batches of glitches. This specificity would improve the ability to identify potential auxiliary channels that correlate with certain glitch morphologies, which in turn would contribute to identifying their source.

3. Gravity Spy project

The data challenges faced by LIGO are not unique. The increasingly large datasets that permeate every realm of modern science require new and innovative techniques for analysis [30]. In astronomy, individual researchers have traditionally analyzed images of astronomical objects themselves; however, the digital surveys of today image hundreds of millions of objects, making the previous paradigm impractical. The acceleration in data acquisition has not been matched by an increase in human capacity to turn data into knowledge.

Crowdsourcing data to volunteer citizen scientists offers one solution to this problem. Early efforts, such as NASA's Clickworkers, demonstrated the utility of crowdsourcing data to volunteers and the innate desire that the public has to contribute to scientific research [31]. Another early astronomical project, StardustHome [32], led to the development of a general set of tools for citizen science projects known as BOSSA (now pyBOSSA[10]). The highly successful Galaxy Zoo (e.g. [33, 34]) and Zooniverse projects (e.g. [35–38]) have

demonstrated that it is possible to recruit hundreds of thousands of volunteers to make an authentic contribution to data analysis. To date, Zooniverse users have contributed to more than 100 peer-reviewed publications across a broad range of scientific disciplines.

Glitch classification and characterization in LIGO currently utilizes human inspection, and therefore fits naturally into a citizen science framework. However, as scientific endeavors such as LIGO and future astronomical sky surveys become more data intensive, new methodologies must be explored for utilizing citizen scientists in data analysis. The large synoptic survey telescope (LSST), for example, will image tens of billions of galaxies [39], which is orders of magnitude more data than even the most successful citizen science projects can analyze. Supervised machine learning has proven to be a useful tool in projects which require a systematic analysis of substantial datasets such as these. However, these algorithms require a large, labeled dataset for training and struggle to identify new morphological categories as they appear.

The data challenges faced in astronomy and other sciences today require a new generation of intelligent citizen science projects that are smarter about allocating tasks and more sophisticated in combining human and machine classification. This provides a two-way path to developing better machine learning algorithms and, for the first time with Gravity Spy, better human classifiers as well. Gravity Spy facilitates a symbiotic relationship between humans and computers, leveraging human pattern recognition skills as a tool for image recognition and machine learning as a tool for systematic analysis of large datasets. Citizen scientists analyze glitches from the LIGO data stream via human classification interfaces known as *workflows*, providing labeled morphological classes as training data for machine learning algorithms. Trained machine learning algorithms classify the LIGO glitches data in full, determining confidence scores in each classification and feeding the most questionable glitches back to the citizen scientists for further analysis.

A further innovation is that machine-analyzed glitches will guide training of new volunteers. As part of the Gravity Spy system, images whose morphology is agreed upon by experts, known as ‘gold standard’ images, are integrated into the user workflows. Individual user performance is analyzed by comparing that user’s classifications with such gold standard images. This form of user analysis expedites the retirement of glitches and the growth of machine learning training sets (see section 3.3.2 for more details). Figure 2 shows the interconnected components of the Gravity Spy project, and the movement of glitches through the project.

Developing the next generation of citizen science projects requires significant advances in our understanding of human-centered computing. Studies of such projects have begun to answer important human-centered computing design questions [40], such as what kinds of tasks can non-experts perform reliably? What factors motivate participants? How do participants learn to perform the task or learn about the underlying science? Gravity Spy provides a platform to explore these questions more systematically, asking participants not only to apply existing scientific knowledge, but also to generate new knowledge (in this

[10] www.pybossa.com

case, new categories of glitches). This setting allows the exploration of additional questions, such as how to support not just individual citizen scientists but teams working together, and what organizational structures are most appropriate?

Finally, Gravity Spy addresses the pressing need to understand the development of socio-computational systems that merge the distinctive strengths of computers (i.e. the ability to process large amounts of data systematically) and the humans (i.e. the ability to see patterns and spot discrepancies) [41–44]. Knowledge of how to use human-coded data to improve machine learning (e.g. by applying an active learning approach) is fairly well developed, though there are still opportunities to study the human-interface aspects of the process. In contrast, we still know little about how to use machine-analyzed data to improve human performance and thus how we best leverage human learning and machine learning in a joint effort.

3.1. Data preparation

Data preparation for the Gravity Spy project (i.e. the link from LIGO to the rest of the project in figure 2) presented three critical challenges:

1. Given that during O1 alone there were more than 10^6 glitch triggers identified by the Omicron transient search algorithm [24, 25], it is crucial to determine which glitches were best fit for volunteer classification and most useful for LIGO detector characterization and data analysis
2. Deciding the proper presentation of the morphologically-diverse zoo of glitches to both volunteers and machine learning algorithms
3. Since there is no complete catalog of glitch categories that appeared during O1, the preparation of a training set needed to develop organically from various sources associated with the project

3.1.1. Data selection—In order to tackle the first challenge, we only use glitches that satisfied the following criteria. First, the glitch occurs while the detector is in *lock* and in observing mode, meaning the state of the detector was adequate enough to be searching for gravitational waves and ready for data analysis. For O1 glitches, we also neglected times that were flagged for poor data quality, though depending on the latency at which such flags are raised in future observing runs this cut may not be applied when feeding data into the system. The data quality flags implemented remove data that should not be analyzed due to a critical issue with a key detector component not operating in its nominal configuration (category 1) and times when a noise source with known physical coupling to the main gravitational-wave channel is occurring (category 2) [11]. Second, we neglect glitches where the SNR reported by the Omicron search pipeline is below 7.5, as glitches below this threshold prove to be exceedingly difficult to classify by eye. Third, the peak frequency of the glitch falls between 10 Hz and 2048 Hz. These choices are motivated by our goal to analyze and understand glitches that have the largest impact on the gravitational-wave searches: low-SNR glitches are less detrimental to searches, and this frequency range aligns well with LIGO's most sensitive frequency band and the frequency range expected for compact binary coalescence gravitational-wave events. In addition, as the Gravity Spy

pipeline was first run after the conclusion of O1, we had the benefit of being able to apply the same data quality vetoes [11, 26] to the data as were applied during astrophysical searches. Again, this was in order to analyze the glitches that have the largest impact on gravitational-wave searches.

The gravitational-wave events GW150914 [2] and GW151226 [3] and gravitational-wave trigger LVT151012 [4] are not included in the Gravity Spy dataset. Hardware injections [45] are included, and constitute most of the subjects in the 'Chirp' glitch class. However, in future observing runs potential gravitational-wave signals will not necessarily be redacted, as new images will be added to the project before the results of gravitational-wave searches are available. To ensure astrophysical claims cannot be made by non-LSC users, GPS times are replaced by a random, unique ID for each image in the Gravity Spy system. Therefore, potential astrophysical signals will be indistinguishable from hardware injections in the detectors, and users will have no knowledge of when a particular trigger was recorded.

3.1.2. Omega scans—We met the second challenge by representing these glitches with Omega Scans [46]. Omega scans originated as a pipeline for the detection of gravitational wave transients, and are similar to spectrograms in that they represent glitches in time-frequency-energy space. They are also excellent at visualizing glitches that may cause problems in gravitational-wave searches. Omega scans represent a generic signal as a combination of sine-Gaussians. The main utility of Omega Scans is an unmodeled SNR calculation with the template for a signal defined by its 'Q' value, where Q is the quality factor of a sine-Gaussian waveform. In practice, this template signal consists of a time-frequency tiling. Like all template searches, an omega scan searches over a range of Q templates (i.e. time-frequency tilings) and identifies the template that gives the loudest SNR value. After identifying the Q template that provides the loudest value, the most significant tile for that Q template is identified and a spectrogram is generated. The color scale of the image is the normalized energy, which is directly related to the SNR of a tile and defined as the square of a given tile's Q transform magnitude divided by the mean squared magnitude in the presence of stationary white noise:

$$Z = \frac{|X|^2}{\langle |X|^2 \rangle} \quad (1)$$

where Z is the normalized energy and $|X|$ is the Q transform magnitude of a tile [46].

As shown in figure 1, each image has the glitch fixed at the center of the omega scan, and each glitch is visualized using four different time windows (±0.25, 0.5, 1.0, and 2.0 s) to accommodate the varied durations that different glitch morphologies persist. Human volunteers and machine learning algorithms are presented all four time durations of each glitch for classification purposes.

3.1.3. Training set—The final challenge was the construction of a large and accurately-labeled set of LIGO glitches. The generation of such *training sets* is one of the most difficult components of supervised machine learning, and necessary to properly train classification

algorithms. The past attempts to compile glitches into morphological classes using computer algorithms (e.g. [13–17]) often rely solely on raw data or metadata from search pipelines rather than by-eye classification. In addition, new glitch morphologies that appeared during the first observing run of LIGO were not analyzed nor categorized to the level of pre-existing glitches.

A training set of glitches from O1 was generated for the Gravity Spy project by observing large quantities of Omega Scans and categorizing the images by morphology, with the aid of simple machine learning algorithms. First, consultation with LIGO detector characterization experts helped identify a few prominent and documented classes of glitches. Omega scans of all LIGO Omicron triggers within the frequency and SNR cuts specified above were generated, which reduced the dataset to about 10^5 glitches for the entirety of O1. We proceeded by classifying glitches from this set into preexisting categories based on the morphology of the glitch in its omega scan, and new categories of glitches were identified and accumulated in the process. Due to the similar morphological characteristics of many glitch classes, this process took multiple iterations to assure reliability in the class differentiation. Nonetheless, this tactic only accumulated ~100 glitches per class.

This small set of human-identified glitches was used to train preliminary machine learning algorithms to classify the remainder of the glitch dataset. Though such algorithms only achieved classification accuracy of ~80% – 90%, they were useful in differentiating the unlabeled dataset into morphologically-similar classes, thus fostering an easier by-eye classification process. As will be described in section 4.2, during the beta-testing of the project, two new classes were also identified and characterized by Gravity Spy volunteers. Additional training data for these new classes was identified using the same methods described above.

In total, a labeled training set of 7718 glitches was built from both the Livingston and Hanford detectors for the preliminary machine learning analyses presented in this paper. These glitches are grouped into 20 classes, with exact proportions shown in Table 1. Given that each glitch is imaged at a maximum duration of 4 s, this amounts to 8.58 h (0.7%) of O1 data [4].

3.2. Citizen science

Once the Omega Scans of glitches are in the system, they are classified by Gravity Spy volunteers, who populate the human-classification unit of the system.

3.2.1. User interface—The user interface for Gravity Spy was created using the Zooniverse DIY project builder[11], which enables anyone to build their own Zooniverse citizen science project for free through a set of easy-to-use, browser-based tools. The Gravity Spy classification interface containing the currently-known 20 glitch classes as options is shown in figure 3. Example images of each glitch morphology can be found on the Gravity Spy website[12]. Through this interface, volunteers are shown individual Omega

[11] www.Zooniverse.org/lab
[12] www.gravityspy.org

Scans of glitches to classify into one of the categories. Volunteers have the ability to cycle through multiple renderings of a given glitch over differing time durations, enabling a volunteer to visualize both long-duration and short-duration glitches. After classifying a glitch, the volunteer has the option of moving on to further classifications, or posting the glitch to 'Talk', which is the Zooniverse discussion forum that provides a basis for interaction between Zooniverse volunteers and Gravity Spy project scientists.

Clicking on any glitch morphology option provides basic written information about that class to the volunteer along with multiple example images belonging to that glitch class. In addition, this dialog contains images of glitch morphologies that are often confused with that class, providing a simple means of changing classification choice to similar glitches if the volunteer misidentified the image initially. Alternatively, users can narrow down glitch options by filtering based on how long the glitch persists (*duration*), the characteristic frequency of the glitch (*frequency*), and whether the glitch is evolving in time (*evolving*). Further information regarding each glitch class can be found in the *field guide* (visible on the right side of figure 3).

If a glitch does not fit into any of the predefined categories, a user can classify it as 'None of the Above'. In doing so, a volunteer is asked follow-up questions describing the morphology of the glitch (i.e. information about its duration, frequency, and time evolution). By this process and through user activity on Talk, new classes of glitches can be identified and integrated into the Gravity Spy project. This allows the Gravity Spy glitches classes to evolve and follow changes in the glitch types that occur in the LIGO detectors.

3.2.2. Volunteer training—A key question in citizen science is how reliably volunteers perform the classification task, known as *results quality*. Zooniverse approach to citizen science directly addresses this question and has led to an established track record of producing quality data for use by the wider scientific community and publications across the disciplines. By embedding training within the interface and creating consensus results based on numerous classifications for each image, Zooniverse projects help to make a disparate crowd of volunteers produce reliable results [18].

As with other Zooniverse projects, Gravity Spy begins with a brief tutorial, explaining the project's goals, how to interpret the spectograms, and how to use the classification interface. The field guide and additional content pages describe properties of each glitch class and the LIGO project in more detail.

Research on learning suggests that an effective way to train humans to perform image classification tasks is to provide them with exemplary images from which to learn [47, 48]. Accordingly, as in other citizen science projects, the Gravity Spy classification interface shows the volunteers example images of all the glitch classes to guide the choice.

An advance over the current state of the art citizen science project is that Gravity Spy uses machine learning results to train the human volunteers more systematically. Specifically, the system moves new volunteers through a sequence of levels in which they are presented with an increasing number glitches classes and sophistication of features within the classification

interface, intended to improve their ability to classify glitches [49]. Essentially, the system is 'tutoring' volunteers, but rather than simply taking images from a predefined set of training materials, it identifies novel images in need of classification that should still help beginners to learn.

Upon joining a project, a volunteer is presented glitches that have been classified by the machine learning models as likely belonging to only one of two very distinctive classes. For each glitch, volunteers are asked to annotate it as being an instance of one of the two classes or 'None of the Above' (a reduced version of the interface shown in figure 4). These exemplary images help the volunteer to learn how to identify this subset of glitch classes. Once volunteers are reliably classifying these two initial classes, additional classes are introduced.

In the current implementation, volunteers also classify gold standard images, which in practice are a subset of the full machine learning training set. After classifying a gold standard image, the volunteer immediately receives feedback as to whether their classification agrees with the expert classification. Initially, 40% of images presented to beginning volunteers are gold standard, and this frequency dynamically decreases as a volunteer classifies gold standard images correctly.

As volunteers progress through the training regimen, they are presented with more classes that the machine learning model has classified with high confidence. The classifications during this training period contribute to the project by verifying the high-confident, yet imperfect, machine learning results. In addition to training the volunteers in recognizing members of more glitch classes, the levels are expected to motivate users by appealing to their sense of accomplishment.

Once the user has completed multiple rounds of training on a subset of glitch classes with high machine learning confidence scores, they are considered fully qualified and will be given glitches to classify at varying levels of machine learning confidence in all known classes or even glitches for which the machine learning has no good classification, thus further contributing to the identification of new glitch categories. Since the system tracks each volunteer reliability, it can also assign tasks based on the capabilities of each volunteer.

3.2.3. Workflows—Glitches are first sent through the machine learning classifier, which is trained on a set of images pre-classified by experts (see section 3.1.3) and images retired from the project. Based on the machine learning confidence of the classification of each image, it is routed either to beginning, intermediate, or advanced workflows, as illustrated in figure 5.

Similarly, based on their expertise and reliability level as determined by their performance in classifying (described in section 3.3.2), volunteers are divided into three levels that correspond to the beginning, intermediate, and advanced workflows. Through the Gravity Spy interface, LIGO detector characterization experts will be fed glitches for which the most advanced users cannot reach a consensus. Each volunteer starts at the simplest level and can be promoted to higher levels based on their performance.

As images are classified, the models of both the image and the volunteer are updated. The destination of a glitch (whether it stays in its current workflow, moves to a more difficult workflow, or is retired) is determined by a combination of machine learning and user confidence posteriors. If an image achieves high enough confidence in its classifications, it will be retired and added to the training set to further improve the performance of the machine learning classifer.

The system is built to optimize the retirement of images. Most citizen science projects rely solely on number of classifications as a gauge for retirement (e.g. any image that has 20 classifications is retired from the project). However, this methodology presents multiple problems. Images can be retired even when there is strong disagreement on the correct class. Furthermore, many classifications are essentially wasted on easy images, which may only require a few identical classifications for accurate retirement, whereas difficult images that require deeper analysis may not receive enough classifications. By relying on the combination of machine learning and user classification, and weighting user classifications differently based on their prior performance, the Gravity Spy project aims to ameliorate such issues.

3.3. Machine learning

The following section describes the application of machine learning to the problem of classifying images in the Gravity Spy system and how the classifications contributed by volunteers are used to update models of both machine learning image classification and volunteer capabilities.

3.3.1. Image classifier—Deep learning is a branch of machine learning which utilizes algorithms that attempt to model high level abstractions in data by using multiple processing layers, composed of multiple linear and non-linear transformations. The Gravity Spy system uses a deep model with convolutional neural network (CNN) layers, which has shown great performance and is considered the state-of-the-art in image classification [50].

Another reason for exploiting deep learning is its scalability; compared to traditional machine learning methods such as support vector machines (SVMs), deep learning can handle and take advantage of copious amounts of data. Figure 6 illustrates the machine learning process used.

Many studies (e.g. [51, 52]) have shown that using multiple sources of information can improve the overall performance of classification. In this project, the multiple glitch durations that are also shown to Zooniverse volunteers are utilized. These durations are merged into a square form so that kernels can slide over all different durations and learn the glitch patterns. Two convolution layers are utilized first. The kernels slide over the input matrix, multiplying their corresponding weights to the input matrix and outputting a new matrix. The output of each kernel is known as a *feature map*.

Feature maps are usually subsampled using a max (or mean) operation. Here, max-pooling is used for down sampling—a square matrix slides over the feature map and gives the maximum value among the elements inside it. A layer of activation functions is used to

determine the output of a given neuron. The Gravity Spy model uses a popular activation function known as rectified linear unit (ReLU) which is defined as max(0, x). Then, a fully connected layer is applied. Each node in the fully connected layer is connected to all nodes of the previous layer. The final layer is a softmax layer with 20 outputs. Softmax is a fully connected layer with the same number of nodes as the number of classes, and is widely used as the final layer in multi-class classification tasks. The output of the softmax layer, when image 'i' is given as the input to the classifier, is defined as

$$o_i^c = \frac{e^{w_c^T x}}{\sum_{c=1}^{C} e^{w_c^T x}} \qquad \text{for } c = 1, \cdots, C \quad (2)$$

where o_i^c is the output score of class c for glitch i, x is the output of the layer before softmax when image 'i' has been given as input to the model, and w_c is the vector of weights connecting the output of the previous layer to cth node in softmax layer. C represents the total number of classes, in our current case 20. The output score of the softmax layer, o_i^c, is used as the probability distribution found by the image classifier. The score vector obtained from machine learning for image i is defined as follows:

$$\mathbf{p}_i^{\mathrm{ML}} = [o_i^1, \cdots, o_i^c, \cdots, o_i^C] \quad (3)$$

The next step is to train the model. The model optimizes a loss function defined on the training data, using cross-entropy:

$$\text{loss} = -\sum_{j=1}^{N}\sum_{c=1}^{C} y_j^c \log o_j^c \quad (4)$$

where o_j^c is the model's output for class c when the jth training sample is given to the network, y_j^c is equal to unity if the jth sample is from class c, otherwise it is zero, and N and C are the total numbers of the training samples and classes, respectively. To optimize the objective function, the Adadelta [53] optimizer is used. This optimizer monotonically decreases the learning rate and shows good performance in our experiments. More details about the proposed machine learning image classifier and experiments can be found in [54].

3.3.2. Crowdsource classifier—As noted above, the system will maintain a model of each volunteer ability to classify glitches of each class and will update the models after each classification (e.g. increasing its estimate of the volunteer's ability when they agree with an assessment and decreasing it if they disagree). When the volunteer model shows that a volunteer abilities is above a certain threshold, the volunteer will be advanced to the next workflow level, in which they will be presented with new classes of glitches and/or glitches

with lower machine learning confidence scores. In addition, the movement of images through the project is determined by these volunteer performance models, as well machine learning and volunteer classification. As a collective, these algorithms are referred to as the *crowdsourcing classifier*. Further details regarding the crowdsourcing classifier will be presented in an upcoming publication [55].

A confusion matrix is assigned to each volunteer to record their labeling performance. It is defined as $\mathcal{M}^k \in \mathbb{N}^{C\times C}$ for the kth volunteer, where C denotes the total number of classes. An entry of this matrix, m_{pq}^k gives the number of samples belonging to class p labeled as belonging to class q by the kth volunteer. All entries will be initiated as 0 and updated when an image from the golden set is labeled by the volunteer. It will also be retrospectively updated with the labels of testing images that are retired.

Using a volunteer's confusion matrix $\mathcal{M}^k$, a reliability measure is defined for volunteer k as the vector $\mathbf{a}^k = [\alpha_1^k, \cdots, \alpha_c^k, \cdots, \alpha_C^k] \in \mathbb{R}^{C\times 1}$, where α_c^k quantifies the reliability of volunteer k in classifying samples of class c. It is defined as:

$$\alpha_c^k = \frac{m_{cc}^k}{\sum_{j=1}^{C} m_{cj}^k} = p(\hat{y}^k = c \,|\, y = c) \qquad \text{for } c \in \{1, \cdots, C\} \quad (5)$$

where α_c^k is also equal to the probability that the kth volunteer provides a label $\hat{y}^k$ for an image, as belonging to class c, given the true label y is indeed equal to c.

After modeling the volunteers' reliability, the classification of a test sample of images using multiple annotations is determined. A test image is initially provided to the machine learning classifier which outputs a probability vector $\mathbf{p}_i^{\mathrm{ML}}$. The developed algorithm uses the machine learning probabilities and volunteer classification labels to predict the true label [55].

With the assigned labels from R_i volunteers for a given image i, the goal is to fuse these labels and find the posterior probabilities $p(y_i^{cr} = j \,|\, \hat{y}_i^1, \cdots, \hat{y}_i^{R_i})$ for $j \in \{1, \cdots, C\}$, where y_i^{cr} is the predicted label from crowdsourcing information. The final predicted label $\tilde{y}_i$ is calculated as:

$$\tilde{y}_i = \mathrm{argmax}_j \frac{p(y_i^{cr} = j \,|\, \hat{y}_i^1, \cdots, \hat{y}_i^{R_i}) + \mathbf{p}_i^{\mathrm{ML}}(j)}{\sum_{j=1}^{C} p(y_i^{cr} = j \,|\, \hat{y}_i^1, \cdots, \hat{y}_i^{R_i}) + \mathbf{p}_i^{\mathrm{ML}}(j)} \quad (6)$$

where $\mathbf{p}_i^{\mathrm{ML}}(j)$ denotes the jth component of $\mathbf{p}_i^{\mathrm{ML}}$.

As classifications are made, the initial priors provided by machine learning are replaced by the posterior probability of each class, which contains both machine learning and volunteer

classification information. The posterior probabilities continually update until an image is retired or the image receives a predefined maximum number of volunteer classifications and is moved to a higher workflow to be investigated by more advanced volunteers. To decide on the retirement of the test image, a threshold t_j is defined per class based on the difficulty of classifying glitches in that class. The threshold vector can be thus defined as $\mathbf{t} = [t_1, t_2, \cdots, t_C]^T$.

Having the posterior probabilities of all the classes from equation (6) and putting them in a vector $\mathbf{y}^i = [y_1^i, \cdots, y_C^i] \in \mathbb{R}^C$, the posterior probability vector $\mathbf{y}^i$ can be compared with threshold vector $\mathbf{t}$. If the entry of $\mathbf{y}^i$ that carries the highest posterior probability is greater than the corresponding entry of $\mathbf{t}$, the image is retired with label j for which $y_j^i \geq t_j$. Then this retired image is sent to training set with label j as its true label. If no entry of $\mathbf{y}^i$ is greater than the corresponding entry of $\mathbf{t}$, further action is needed. Based on the number of volunteers who have labeled the image, either more volunteers at the same level must label the image or the image is moved to a more advanced workflow.

As for volunteer promotion, when a volunteer labels images from the golden set, their confusion matrices are updated. Also, as test images are retired, the golden set is updated and the confusion matrices are updated retrospectively by comparing their labels with the label of the retired image. With equation (5), the $\mathbf{a}^k$ vector is calculated from the confusion matrix $\mathcal{M}^k$. Reliability threshold values are defined for each class: ($\mathbf{T}_j = [T_1, \cdots, T_C]$). If all the values of the vector $\mathbf{a}^k$ exceed the threshold values in $\mathbf{T}_j$, the volunteer is promoted to the next level. If not, they will need to do more correct classifications to be promoted.

3.4. Socio-computational research support

Finally, the socio-computational research component will allow for systematic measurement and experimentation with the performance of project components. Our first planned experiment is to compare the performance of volunteers who have gone through the training process described above to the performance of those who start right away with the full set of classes for classification (i.e. the typical approach for citizen science projects). By doing so, one can test if users who go through the training regimen contribute more and show better performance on the classification tasks.

Second, the training system described above has a large number of parameters (e.g. how many and which classes to introduce at each level and the class-specific machine learning certainty cutoffs for images to be placed in each level). Experimentation will be useful to determine the optimal settings. For example, one can test the benefits and tradeoffs of advancing volunteers to higher levels more rapidly: quicker advancement might be good for motivation but negative for performance (and vice versa).

Finally, the system will enable us to experiment with other factors that affect volunteer performance, such as the kinds of motivational messages provided or information on the novelty of glitches. A particularly interesting set of questions gauge the effects of feedback that can be provided to volunteers based on machine learning classification confidence. Again, it is possible that there are tradeoffs involved: letting a volunteer know the machine

learning confidence score of an image might be useful feedback to improve performance but also potentially demotivating if the machine learning and the volunteer disagree, or if it leads to volunteers feeling that their contributions are unnecessary.

There are many unanswered questions about how volunteers will learn in this setting that go beyond the specifics of glitch classification. In particular is the concern of how much the volunteers will need to know about gravitational-wave astrophysics and the workings of the detectors that produce the glitches. Included as part of the workflows is a mini-course on gravitational-wave astrophysics and LIGO detector characterization that presents the next slide of the course after a given number of classifications. Additionally, there are background information pages on the site that describe the detector in more detail. Though the background pages are optional and one can opt-out of the mini-course, one can track which volunteers visit these pages to examine the impact on performance. Further details on the socio-computational research related to Gravity Spy can be found in [56].

4. Preliminary results

The full public launch of the Gravity Spy project was on October 12 2016, about a month before the planned commencement of LIGO's second observing run (O2). Through the initial renditions of machine learning models and beta-testing of the human interface, the preceding phases of this project have already shown promise in achieving high-level, multi-class glitch classification using true (rather than synthesized) LIGO detector data and the ability of the public to distinguish new categories of glitches.

4.1. Initial machine learning performance

As discussed in section 3.1.3, the initial machine learning training set consists of 7718 total glitches from 20 classes, using 75%, 12.5%, and, 12.5% of the full set as training, validation, and test sets, respectively. The number of iterations and the batch size were set to 130 and 30, respectively. The classification of testing data achieved an average accuracy of 97.1%.

As can be seen in the training set breakdown (table 1), the distribution of samples over classes is highly imbalanced. Therefore, it is better to study precision and recall values of each class to analyze the performance of the glitch classifier. *Precision* is defined as the number of glitches that are correctly labeled as a particular class divided by the total number of glitches that are predicted as that particular class, gauging how often a classifier is correct when it predicts a glitch is in a given class. *Recall*, also known as sensitivity, is the number of glitches predicted correctly as a particular class divided by the actual number of glitches in that particular class, in essence a measure of how often a classifier predicts a glitch in a particular class when it is actually in that class. These values are presented in figure 7.

As one can observe from figure 7, the precision and recall values are near unity for most classes. Certain classes, particularly classes that suffered from a low number of training samples (e.g. 'Wandering Line' and 'Paired Doves') or a high variability in morphological characteristics (e.g. 'None of the Above' and 'No Glitch'), achieved lower precision and recall values. 'None of the Above' and 'No Glitch' are not defined by specific morphological traits. 'None of the Above' is the category which harbors all glitches that do

not fit in the other 19 classes. Therefore, this class does not have a specific morphological distribution over sample space. The 'No Glitch' category has a similar property, as this class consists of all glitches which do not have intense energy in the image, and the low-level noise does not have a consistent morphology through the training set. Though not morphologically defined compared to the other classes, the inclusion of these two *catch-all* classes allows for the full classification of the dataset, and provides a medium for determining new classes of glitches as the project progresses. The challenge of the classification of 'Paired Doves' and 'Wandering Line' groups is likely due to a lack of samples, as these two classes have the lowest number of samples with 30 and 44, respectively.

4.2. Gravity Spy system beta testing results

The Gravity Spy project launched three Beta versions to test the user interface and user promotion in April, June, and September 2016, each of which lasted approximately one week. During this time, a version of the project was made public and promoted to a small subset (~2000) of Zooniverse volunteers. The main goal of the beta testing was to check the functionality of the site and to receive feedback on the interface design. However, the activity on the site also proved the basic premise of the project: volunteers can reliably classify glitches and identify new morphological classes. Beta testing of the website engaged over 1400 users and delivered over 45 000 glitch classifications. This activity in turn led to hundreds of conversation threads on the website talk forum and fostered excitement and intrigue for the nascent field of gravitational-wave astrophysics. The work culminated in the discovery of multiple new and substantial glitch categories from LIGO first observing run, including glitches which would later receive the names 'Paired Doves' [57] and 'Helix' [58]. Example images of these glitch morphologies are shown in figure 8. In particular, the discovery of the 'Paired Doves' class proved significant in LIGO detector characterization endeavors, as this glitch resembles signals from compact binary inspirals and is therefore detrimental to the search for such astrophysical signals in LIGO data. The project activity during the Beta versions is testament to the ability of citizen science projects to engage and involve the public in scientific advancement. A deeper analysis of these morphologies with regard to LIGO detector characterization and further techniques to optimize the integration of citizen science output to large-scale data analysis will be presented in future publications.

5. Conclusions and future prospects

As LIGO searches for gravitational waves, the Gravity Spy project will endeavor to improve the understanding of the LIGO detectors and reduce the impact of harmful noise, all while engaging the general public in gravitational-wave physics. Gravity Spy also plans to incorporate data from the multiple interferometers joining the advanced network in upcoming years (e.g. [5, 6]) to further assist in noise characterization. The full launch of the Gravity Spy project on October 12 2016 incorporated the machine learning analysis and crowdsource classifier into the system, providing each user with a tailored progression through the multiple workflows and pairing machine learning confidence scores with user classifications to optimize the retirement of images and classification accuracy. The project

shows clear utility in aiding gravitational wave detector characterization and creates an avenue to analyze the socio-computational interaction.

Each day during LIGO's upcoming observing runs, the Gravity Spy system will generate Omega Scans of triggers that have passed low-latency data quality cuts and fit within the SNR and frequency thresholds defined in section 3.1. These newly-acquired images will be analyzed using the most current renditions of the machine learning classifier, and integrated into the testing sets available for human classification. As images are retired from the test set, they are added to the machine learning training sets, which re-trains whenever 100 new images are retired and appended. Daily pages summarizing the results are available to all LSC members.

When new classes appear in the detector and trends in the 'None of the Above' class emerge (via clustering of descriptive features from the follow-up questions and collections on the Gravity Spy Talk forum), new categories are added to the interface at the discretion of the Gravity Spy team. By doing so, the project maintains the ability to evolve with the detectors. In addition, the data synthesis for this project can adapt to the activity of the users; adjusting the SNR threshold of triggers will greatly affect the number of glitches that are generated from the LIGO data stream, and lowering this threshold will provide many more difficult images for users to analyze.

As the project progresses, continual engagement of volunteers will be cultivated by providing complementary data and new tools to aid in the classification (e.g. the ability to view spectrograms from auxiliary channels of data, deeper classifications that included sub-classes of morphologies, and tools to support the discovery of new glitch classes and collaboration among volunteers). This, along with continued interaction between project scientists and volunteers on the Talk forum, will foster sustained engagement in the project. Gravity Spy also presents a test bed for socio-computational interaction. Some of the many possible empirical tests that will be implemented include presenting a different interface to subsets of users to examine its impact on user activity (e.g. retracting the training regimen, changing the wording of the project pitch) and analyzing the classification output to investigate how users learn (e.g. examining if the use of filters diminishes for a user over time, inspecting the performance of a user over time). Furthermore, as the human and machine learning components of the project utilize the exact same data for their classification endeavors, it will provide an interesting comparison of each classifier on a level playing field.

Though crowdsourcing models have proven effective in data analysis endeavors across multiple scientific disciplines, the exponential growth of data acquisition necessitates a smarter way to perform citizen science. The sheer amount of data that modern projects produce will soon outstrip human volunteer time, and simple crowdsourcing methods will no longer suffice as a means to scrutinize such sets. The coupling of citizen science to machine learning algorithms that resourcefully choose the optimal data for human classification is essential to preserve crowdsourcing as a powerful means of data analysis. The integration of human and computer classification schemes will maintain citizen science

as a prolific scientific tool and allow it to scale with the ever-increasing datasets of the future.

Acknowledgments

The Gravity Spy team would like to acknowledge and thank the many Zooniverse volunteers who provided invaluable feedback during Gravity Spy beta tests, and delivered initial glitch classifications for which to test the methods presented in this paper. In addition, the team would like to thank the detector characterization working group of the LSC for useful comments and suggestions, in particular Jess McIver for input during the planning phases, Chris Pankow for useful discussions, and Duncan Macleod for technical support and thorough comments on this manuscript. We would like to thank our undergraduate research team: Luke Calian, Jessie Duncan, Ethan Marx, Isa Patane, Leah Perri, and Ben Sandeen, for contributing to multiple components of the project, including the building and curation of the initial training set of glitches. Lastly, we would like to thank the anonymous referees for useful comments on this manuscript. Gravity Spy is partly supported by the National Science Foundation, award INSPIRE 15-47880. This paper has been assigned LIGO document number ligo-P1600303.

References

1. Abbott BP, et al. Phys. Rev. Lett. 2016; 116:131103. [PubMed: 27081966]
2. Abbott BP, et al. Phys. Rev. Lett. 2016; 116:061102. [PubMed: 26918975]
3. Abbott BP, et al. Phys. Rev. Lett. 2016; 116:241103. [PubMed: 27367379]
4. Abbott BP, et al. Phys. Rev. X. 2016; 6:041015.
5. Acernese F, et al. Virgo Internal Report: VIR 0089A-08. 2008
6. Somiya K. KAGRA. Class. Quantum Grav. 2012; 29:124007.
7. Abbott BP, et al. Living Rev Relativ. 2016; 19:1. [PubMed: 28179853]
8. Aasi J, et al. Class. Quantum Grav. 2015; 32:074001.
9. Abbott R, et al. Class. Quantum Grav. 2002; 19:1591.
10. Matichard F, et al. Class. Quantum Grav. 2015; 32:185003.
11. Abbott BP, et al. Class. Quantum Grav. 2016; 33:134001.
12. Aasi J, et al. Class. Quantum Grav. 2015; 32:115012.
13. Mukherjee S, Obaid R, Matkarimov B. J. Phys.: Conf. Ser. 2010; 243:012006.
14. Rampone S, Pierro V, Troiano L, Pinto IM. Int. J. Mod. Phys. C. 2013; 24:1350084.
15. Powell J, Trifirò D, Cuoco E, Heng IS, Cavaglià M. Class. Quantum Grav. 2015; 32:215012.
16. Powell J, Torres-Forné A, Lynch R, Trifirò D, Cuoco E, Cavaglià M, Heng IS, Font JA. Class. Quantum Grav. 2016; 34:034002.
17. Mukund N, Abraham S, Kandhasamy S, Mitra SS, Philip NS. arXiv. 2016:1609.07259.
18. Borne KD, Fortson L, Gay P, Lintott C, Raddick MJ, Wallin J. AGU Fall Meeting Abstracts. 2009
19. Allen B. Phys. Rev. D. 2005; 71:062001.
20. Allen B, Anderson WG, Brady PR, Brown DA, Creighton JDE. Phys. Rev. D. 2012; 85:122006.
21. Aasi J, et al. The LIGO Scientific Collaboration, The Virgo Collaboration and The NINJA-2 Collaboration. Class. Quantum Grav. 2014; 31:115004.
22. Abbott BP, et al. LIGO Scientific Collaboration and Virgo Collaboration. Phys. Rev. Lett. 2016; 116:241102. [PubMed: 27367378]
23. Nuttall LK, et al. Class. Quantum Grav. 2015; 32:245005.
24. Robinet F. Virgo technical document VIR-0545A-14. 2014
25. Lynch R, Vitale S, Essick R, Katsavounidis E, Robinet F. arXiv. 2015:1511.05955.
26. Smith JR, Abbott T, Hirose E, Leroy N, MacLeod D, McIver J, Saulson P, Shawhan P. Class. Quantum Grav. 2011; 28:235005.
27. Essick R, Blackburn L, Katsavounidis E. Class. Quantum Grav. 2013; 30:155010.
28. Ajith P, Isogai T, Christensen N, Adhikari RX, Pearlman AB, Wein A, Weinstein AJ, Yuan B. Phys. Rev. D. 2014; 89:122001.

29. Isogai T. The Ligo Scientific Collaboration and The Virgo Collaboration. J. Phys.: Conf. Ser. 2010; 243:012005.
30. Hey T, Tansley S, Tolle K. Proc. IEEE. 2011; 99:1334–7.
31. Kanefsky, B., Barlow, G., Gulick, VC. Can distributed volunteers accomplish massive data analysis tasks?; Lunar and Planetary Science Conf. (Lunar and Planetary Science Conf. vol 32); 2001.
32. Méndez, BJH. SpaceScience@Home: authentic research projects that use citizen scientists. In: Garmany, C., Gibbs Moody, JW., editors. EPO and a Changing World: Creating Linkages and Expanding Partnerships (Astronomical Society of the Pacific Conf. Series vol 389); 2008. p. 219
33. Lintott CJ, et al. Mon. Not. R. Astron. Soc. 2008; 389:1179–89.
34. Galloway MA, Willett KW, Fortson LF, Cardamone CN, Schawinski K, Cheung E, Lintott CJ, Masters KL, Melvin T, Simmons BD. Mon. Not. R. Astron. Soc. 2015; 448:3442–54.
35. Smith A, Lintott C, Bamford S, Fortson L. AGU Fall Meeting Abstracts. 2011
36. Kendrew S, Simpson R, Bressert E, Povich MS, Sherman R, Lintott CJ, Robitaille TP, Schawinski K, Wolf-Chase G. Astrophys. J. 2012; 755:71.
37. Hennon CC, et al. Bull. Am. Meteorol. Soc. 2015; 96:591–607.
38. Geach JE, et al. Mon. Not. R. Astron. Soc. 2015; 452:502–10.
39. Abell PA, et al. arXiv. 2009:0912.0201.
40. Sears A, Lazar J, Ozok A, Meiselwitz G. Int. J. Hum. Comput. Interact. 2008; 24:2–16.
41. Introne J, Laubacher R, Olson G, Malone T. Künstl Intell. 2013; 27:45.
42. Prestopnik, N., Crowston, K. Proc. of the 17th ACM Int. Conf. on Supporting Group Work GROUP. New York: ACM; 2012. Purposeful gaming and socio-computational systems: a citizen science design case; p. 75-84.
43. Kittur, A., Kraut, RE. Proc. of the 2008 ACM Conf. on Computer Supported Cooperative Work. New York: ACM; 2008. Harnessing the wisdom of crowds in wikipedia: quality through coordination; p. 37-46.
44. Malone, TW., Bernstein, MS. Handbook of Collective Intelligence. Cambridge, MA: MIT Press; 2015.
45. Biwer C, et al. arXiv. 2016:1612.07864.
46. Chatterji S, Blackburn L, Martin G, Katsavounidis E. Class. Quantum Grav. 2004; 21:S1809.
47. Kim S, Murphy GL. J. Exp. Psychol. 2011; 37:1092.
48. Kulatunga-Moruzi C, Brooks LR, Norman GR. J. Exp. Psychol.: Appl. 2011; 17:195. [PubMed: 21942311]
49. Roads, BD., Mozer, MC. Cogn. Sci. Multidiscip. J. 2016. (https://doiorg/101111/cogs12400)
50. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 2012:1097–105.
51. Katsaggelos AK, Bahaadini S, Molina R. Proc. IEEE. 2015; 103:1635–53.
52. Rohani, N., Ruiz, P., Besler, E., Molina, R., Katsaggelos, AK. Variational gaussian process for sensor fusion; Signal Processing Conf. (EUSIPCO), 2015 23rd European; 2015. p. 170-4.
53. Zeiler MD. arXiv. 2012:1212.5701.
54. Bahaadini, S., Rohani, N., Coughlin, S., Zevin, M., Vicky, K., Katsaggelos, A. Joint deep multi-view models for glitch classification; The 42nd IEEE Int. Conf. on Acoustics, Speech and Signal Processing; 2017.
55. Besler E, Katsaggelos A. Aiding classification tasks by combining machine learning and crowdsourcing data. To be submitted to EUSIPCO 2017. 2017
56. Crowston, K., Østerlund, C., Lee, TK. Blending machine and human learning processes; Proc. of Hawai'i Int. Conf. on System Sciences HICSS (CSCW '08); 2017.
57. Lundgren A. New glitch class: paired doves. advanced LIGO electronic log 27138. 2016
58. Smith J. Glitches from misbehaving pcal-y on October 9. advanced LIGO electronic log 21463. 2015

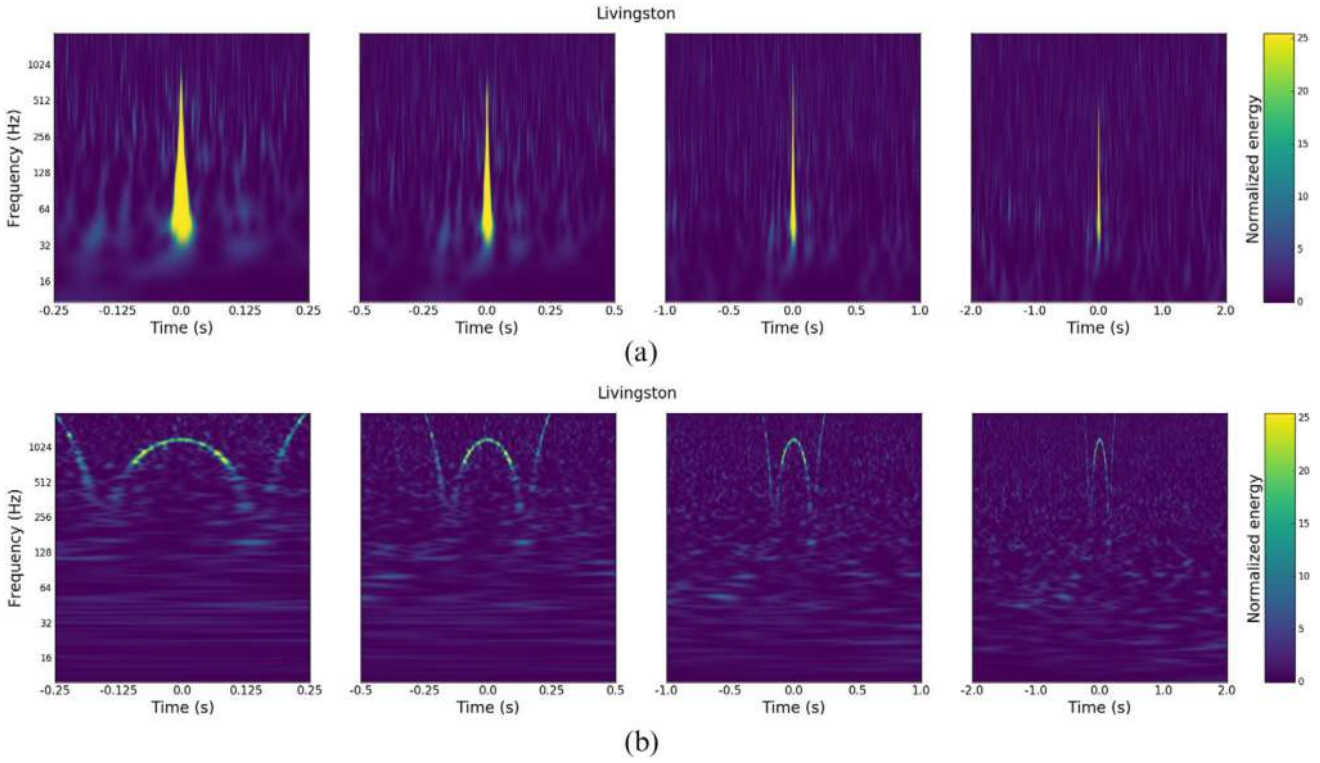

Figure 1.

Spectrogram representation of two example glitches, with color representing the 'loudness' of the signal. Blips (a) are short glitches that usually appear in LIGO's gravitational-wave channel with a symmetric 'teardrop' shape in time-frequency. Blips are the single most important class of glitches in LIGO [11], as they appear in both Hanford and Livingston detectors and are the most stringent limit on LIGO's ability to detect binary black hole merger signals [4]. No clear correlation to any auxiliary channel has yet been identified. Whistles (b), also known as radio frequency beat notes, usually appear in time-frequency plots with a characteristic 'W' or 'V' shape. Whistles are caused by radio signals at megahertz frequencies that beat with the LIGO voltage controlled oscillators [23]. These types of images are what volunteers in the Gravity Spy project classify, and what the associated machine learning algorithms use for training.

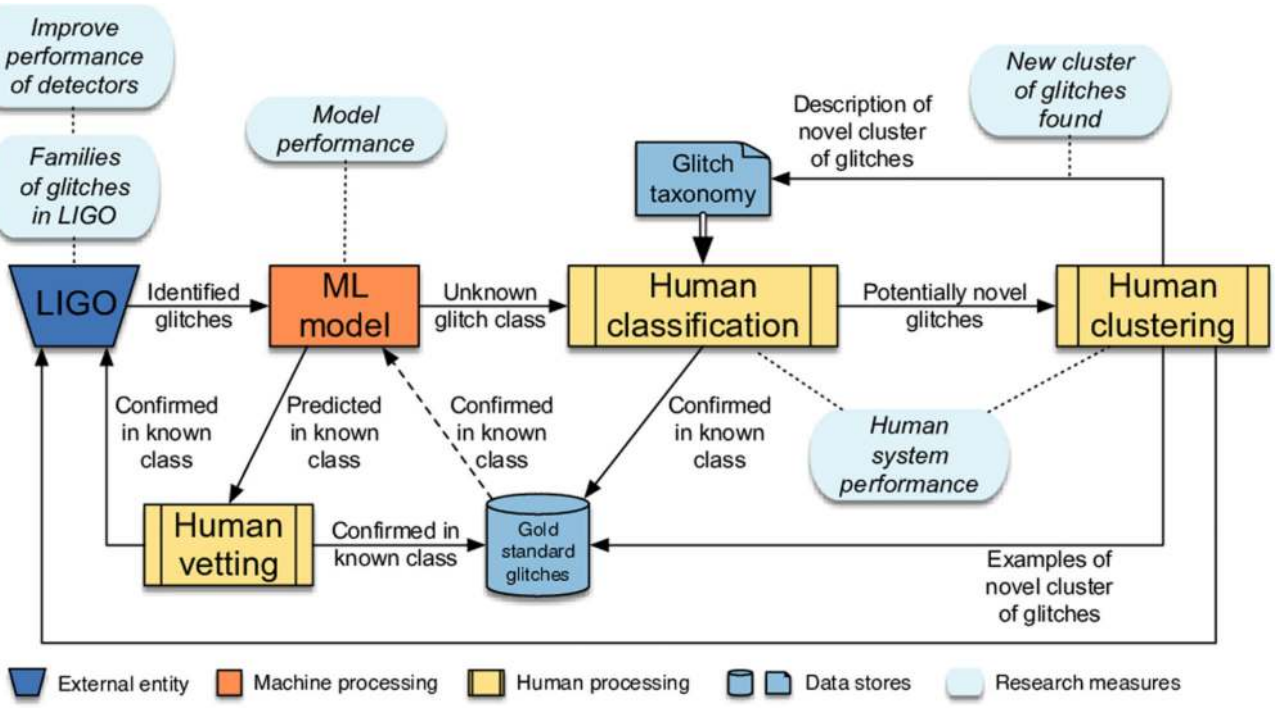

Figure 2.

Gravity Spy system architecture, and overall data flow through the interconnected, interdisciplinary components of the project.

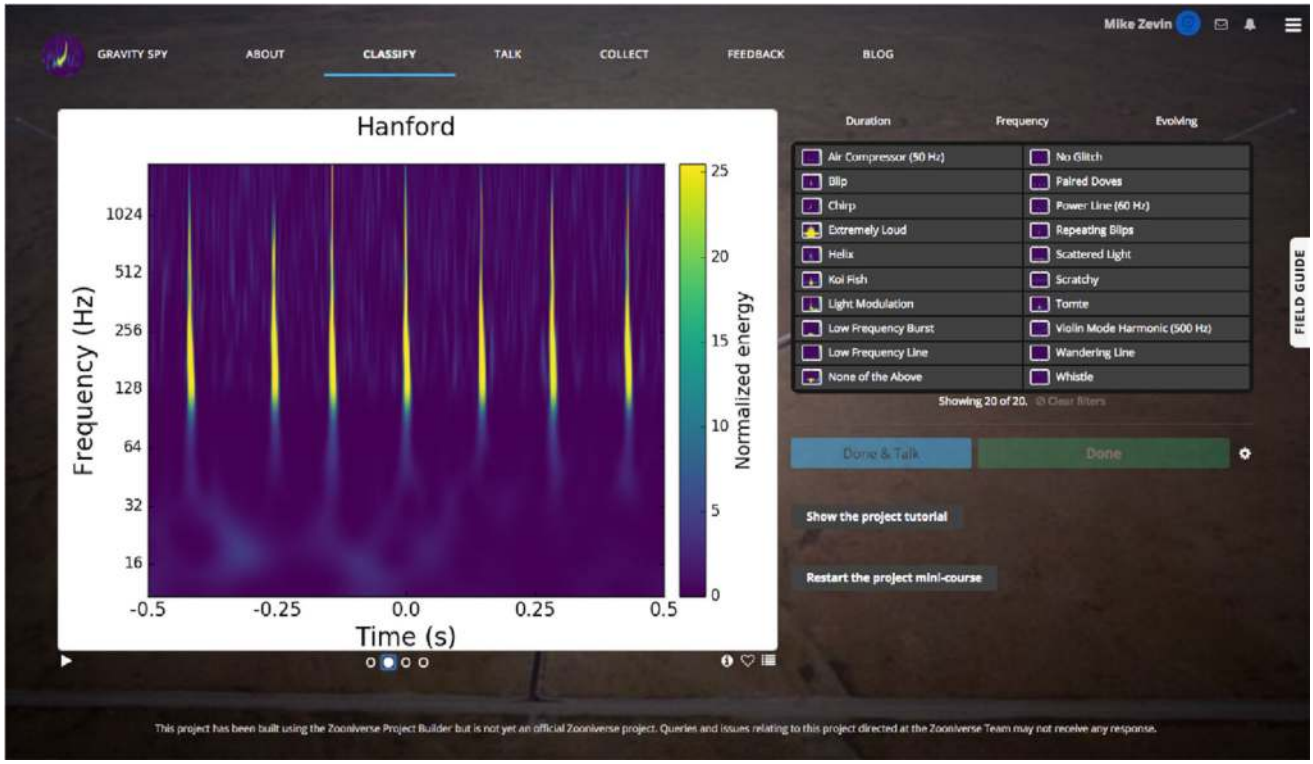

Figure 3.

Gravity Spy user interface. This image shows the *black hole merger* workflow (see section 3.2.3), with all 20 currently designated categories as options.

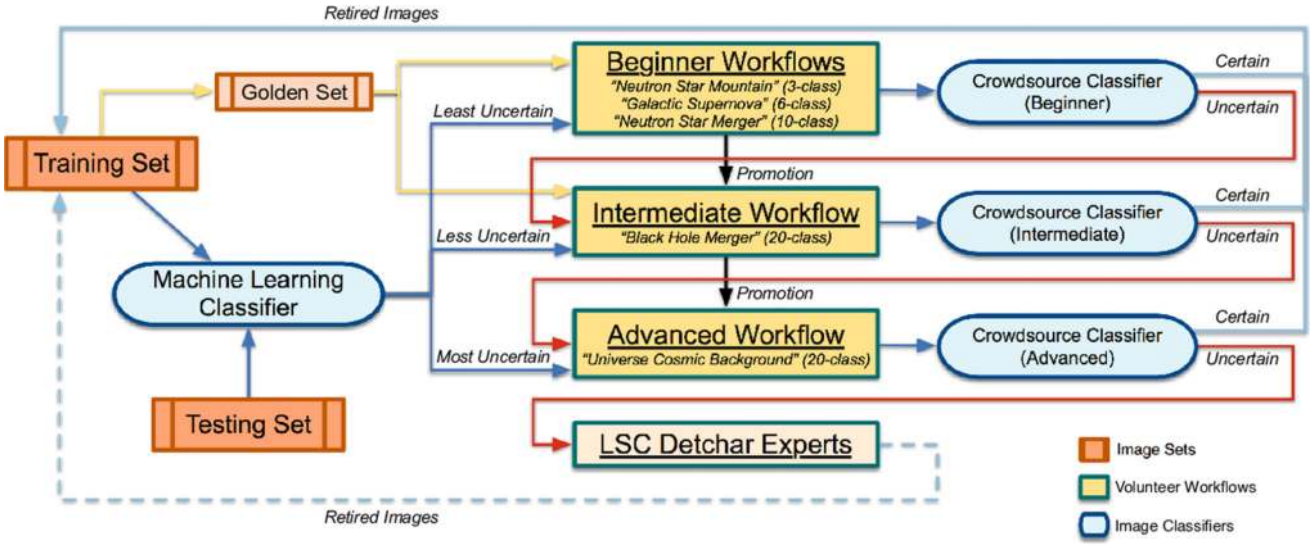

Figure 4.

Movement of images and volunteers through the Gravity Spy project. Green boxes represent the multiple workflows within the project (including the images which are forwarded to experts within the LSC), blue boxes represent the machine learning and crowdsourcing image classifiers, and orange boxes represent the full sets of images, which are designated either as training or testing images (the 'golden set' is the subset of the training set which is used to train volunteers). Note that there are multiple beginner workflows with an increasing number of glitch classes which volunteers progress through as they proceed through the training regimen.

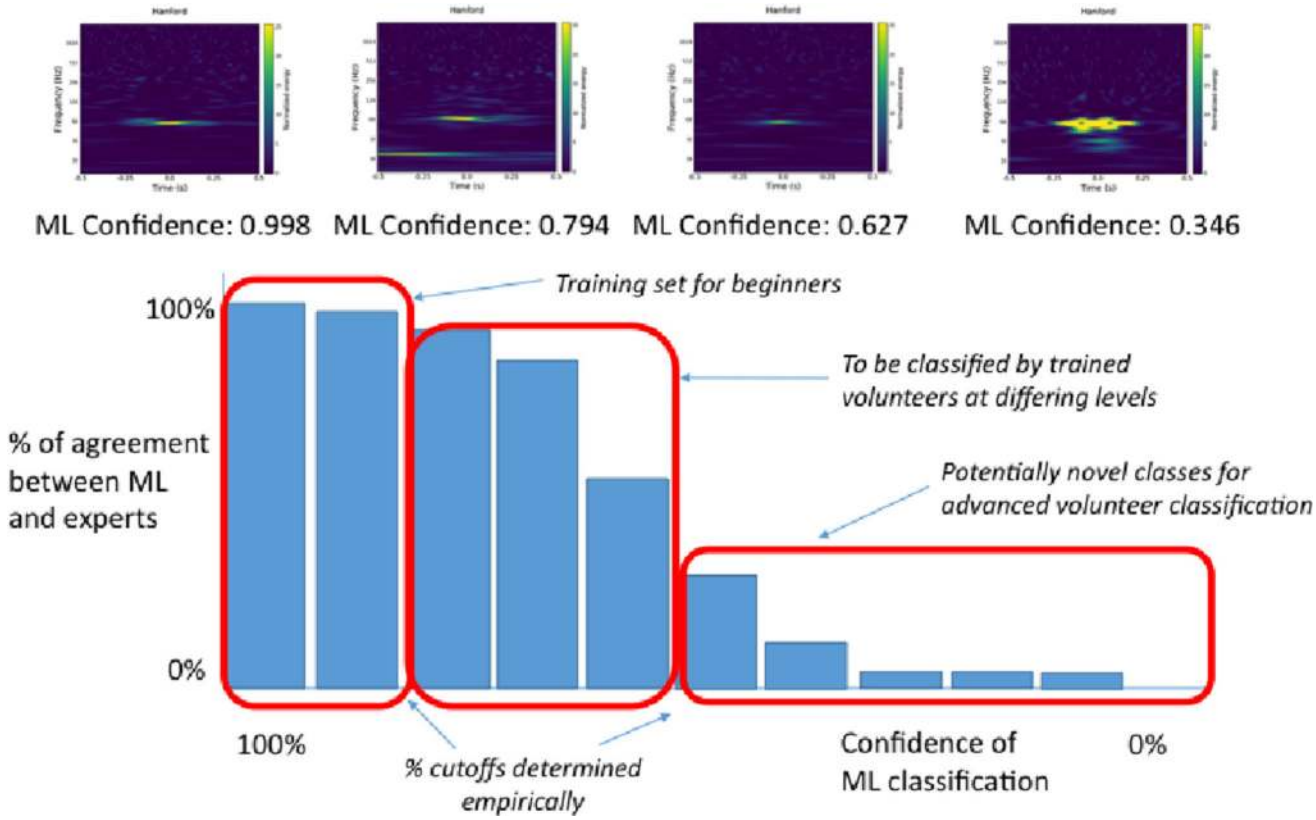

Figure 5.

Relationship between machine learning confidence in glitch classification (*x*-axis) and proportion of images from that class assessed by human volunteers at different skill levels. Example glitches classified as a single class ('power line' glitches) with differing machine learning confidence scores are shown above.

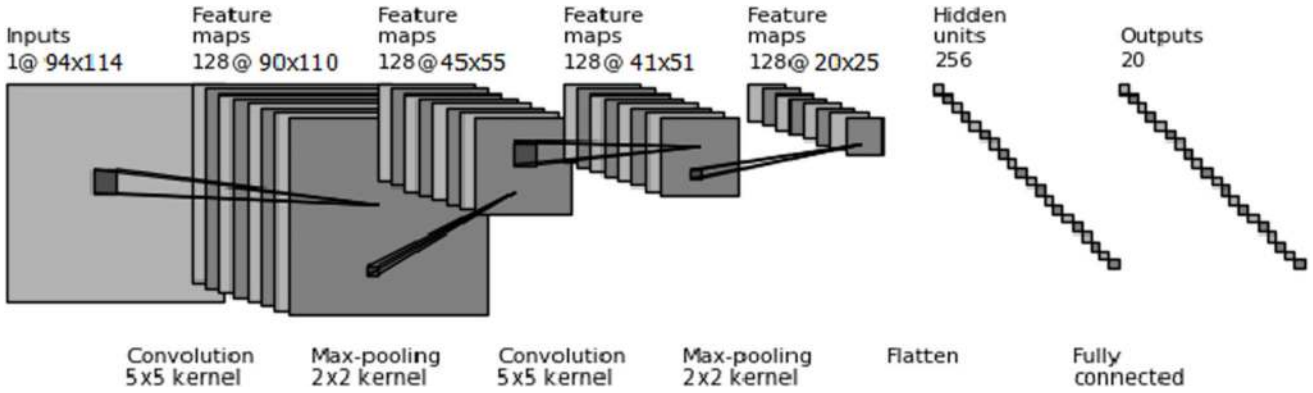

Figure 6.

Deep CNN used for glitch image classification. The network has been introduced on top of the four merged glitch durations. Dimensions of the kernels and feature maps are in units of pixels.

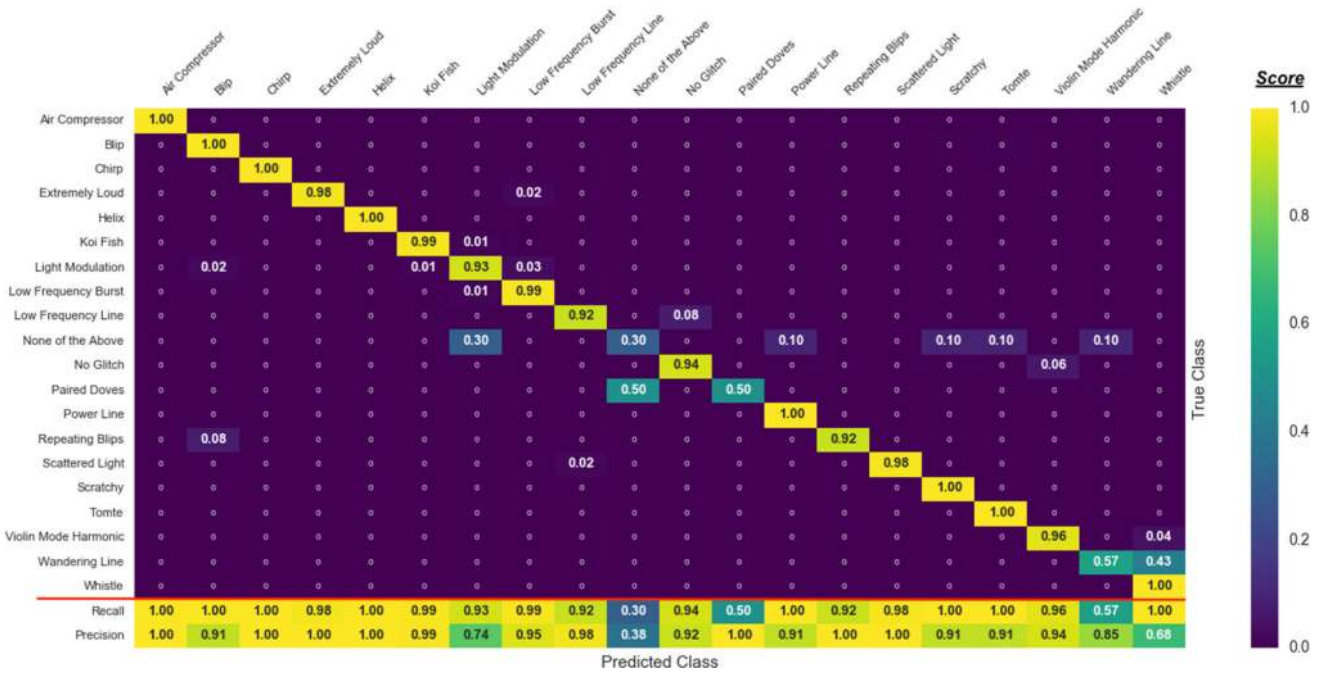

Figure 7.

Confusion matrix for the 20 glitch classes in the testing set classified using CNNs, with recall and precision values appended below for reference. The x and y axes represent the predicted and true classes, respectively, and the confusion matrix is normalized by the total number of glitches in each class in the training set. Due to the normalization chosen, the diagonal elements are identical to the recall values for each class. Closer to unity in precision and recall values corresponds to a more accurate classification for a particular class.

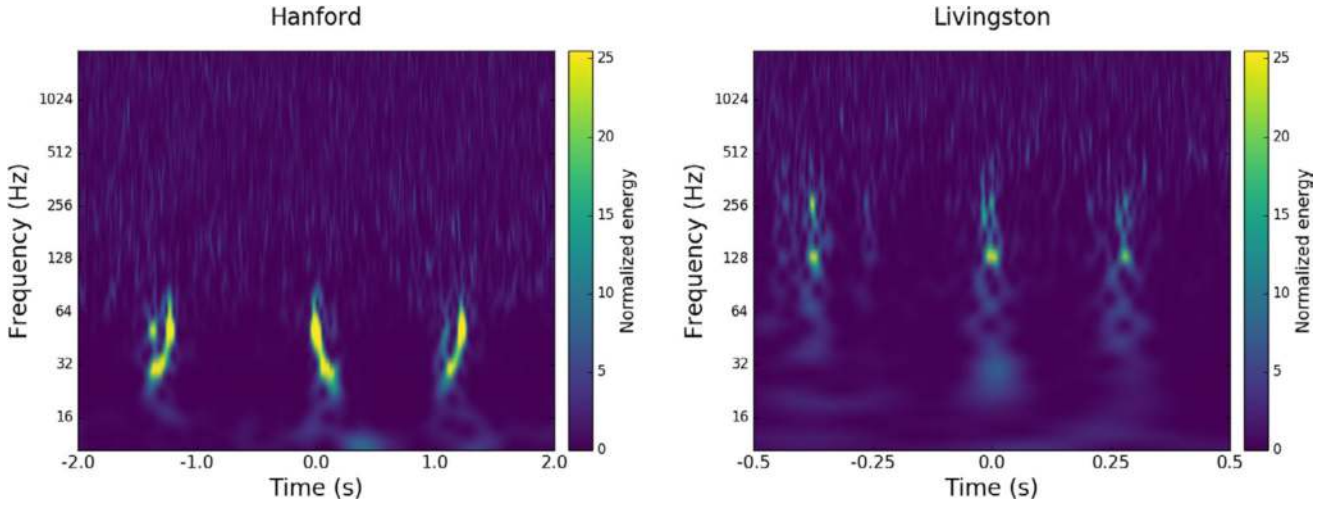

Figure 8.

Two new O1 glitch classes uncovered during Gravity Spy beta testing: ‘Paired Doves’ (left) and ‘Helix’ (right). ‘Paired Doves’ [57] resemble chirps, but alternate between increasing frequency and decreasing frequency. These glitches are related to the ringing of a 0.4 Hz resonance of the pendulum mode in the Hanford beamsplitter, and couple to auxiliary channels monitoring the beamsplitter suspension system.‘Helix’ [58] are possibly related to glitches in the auxiliary lasers (called photon calibrators) that are used to push the LIGO mirrors and calibrate the detectors.

Table 1

Breakdown of morphological categories in the Gravity Spy training set, indicating the component of each class that comes from Livingston detector data and Hanford detector data.

Class	Total		Livingston		Hanford	
Air compressor	54	(0.7%)	0	(0.0%)	54	(1.1%)
Blip	1869	(24.2%)	374	(12.7%)	1495	(31.4%)
Chirp	65	(0.8%)	32	(1.1%)	33	(0.7%)
Extremely loud	453	(5.9%)	187	(6.3%)	266	(5.6%)
Helix	279	(3.6%)	276	(9.4%)	3	(0.1%)
Koi fish	829	(10.7%)	250	(8.5%)	579	(12.1%)
Light modulation	573	(7.4%)	5	(0.2%)	568	(11.9%)
Low frequency burst	652	(8.4%)	473	(16.0%)	179	(3.8%)
Low frequency line	452	(5.9%)	371	(12.6%)	81	(1.7%)
None of the above	189	(2.4%)	36	(1.2%)	153	(3.2%)
No glitch	84	(1.1%)	64	(2.2%)	20	(0.4%)
Paired doves	30	(0.4%)	0	(0.0%)	30	(0.6%)
Power line	454	(5.9%)	180	(12.6%)	274	(1.7%)
Repeating blips	285	(3.7%)	36	(1.2%)	249	(5.2%)
Scattered light	453	(5.9%)	59	(2.0%)	394	(8.3%)
Scratchy	354	(4.6%)	259	(8.8%)	95	(2.0%)
Tomte	116	(1.5%)	46	(1.6%)	70	(1.5%)
Violin mode harmonic	178	(2.3%)	0	(0.0 %)	178	(3.7%)
Wandering line	44	(0.6%)	0	(0.0%)	44	(0.9%)
Whistle	305	(4.0%)	303	(10.3%)	2	(0.0%)