School of Physics
and Astronomy

Gravity Spy and X-Pypeline: A multidisciplinary approach to characterizing and understanding non-astrophysical gravitational wave data and its impact on searches for unmodelled signals

Scott Benjamin Coughlin

Submitted for the degree of Doctor of Philosophy
School of Physics and Astronomy
Cardiff University

October 7, 2019

Summary of thesis

With the first direct detection of gravitational waves, the Advanced Laser Interferometer Gravitational-wave Observatory (aLIGO) has initiated a new field of astronomy by providing an alternate means of sensing the Universe. The extreme sensitivity required to make such detections is achieved through exquisite isolation of all sensitive components of aLIGO from non-gravitational-wave disturbances. Nonetheless, aLIGO 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 that the possibility of accidental coincidence between the two aLIGO 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 allowed by the design sensitivity of aLIGO. For this reason, over the past few years, glitch classification techniques have been developed to help make this task easier. Specifically, I explore the effect of glitches, and their suppression, on key gravitational-wave searches such as that for a Galactic supernova. Moreover, I explore the impact of including machine learning techniques in the post-processing stage of the gravitational-wave search algorithm, "X-Pypeline". When performing a two detector network search for a gravitational wave from a Galactic supernova, this thesis finds that including information about glitch families and using machine learning techniques in the post-processing stages of the analysis can improve the sensitive range of the search by 10-15 percent over the standard post-processing method.

Declaration of authorship

- Declaration:
  This thesis is the result of my own independent work, except where otherwise stated, and the views expressed are my own. Other sources are acknowledged by explicit references. The thesis has not been edited by a third party beyond what is permitted by Cardiff University's Use of Third Party Editors by Research Degree Students Procedure.

  Signed: ..... (candidate) Date: .....

- Statement 1:
  This thesis is being submitted in partial fulfillment of the requirements for the degree of PhD.

  Signed: ..... (candidate) Date: .....

- Statement 2:
  This work has not been submitted in substance for any other degree or award at this or any other university or place of learning, nor is it being submitted concurrently for any other degree or award (outside of any formal collaboration agreement between the University and a partner organization).

  Signed: ..... (candidate) Date: .....

- Statement 3
  I hereby give consent for my thesis, if accepted, to be available in the University's Open Access repository (or, where approved, to be available in the University's library and for inter-library loan), and for the title and summary to be made available to outside organizations, subject to the expiry of a University-approved bar on access if applicable.

  Signed: ..... (candidate) Date: .....

- Word Count: 40453

Contents

List of Figures

List of Tables

I am grateful to many for this work. First of all, I thank my advisers Patrick Sutton and Vicky Kalogera, without whom, this PhD would not have been possible. Their joint support of me not only now, but through my undergraduate and master's research, laid the foundation for this PhD. I thank the Cardiff School of Physics and Astronomy and CIERA, the Center for Interdisciplinary Research and Exploration in Astrophysics, for their support. In addition, I would like to thank the NSF for funding the Gravity Spy project without which this research and joint project would not be possible. In this same vein, I would like to thank the entire Gravity Spy team for all of their efforts to make the machine that is Gravity Spy not only work, but work so well. I thank my fiancée, Jessica, for her support throughout this program. Last but not least, I thank the rest of my family for their support over this year, especially my Dad and Mom for encouraging me to take a chance and apply for this joint supervised PhD.

Chapter 1

Introduction

Einstein's general theory of relativity (GR) predicts that all accelerating objects with non-symmetric mass distributions produce gravitational waves (GW), which are oscillations in the space-time metric [1]. In the same way as light, gravitational effects do not propagate with infinite speed. Whenever the distribution of mass in a given system changes (for example, when one drops a basketball), the gravitational field adapts to this new mass distribution. The speed at which this change propagates is equal to the speed of light, and the resulting changes in the curvature of space-time are GWs. GWs expand and contract space-time orthogonal to the direction of their motion. Therefore, an instrument designed to detect the displacement of two objects relative to each other could sense a gravitational wave passing through it. The problem, however, lies in the amplitude of GWs, and, consequently how much they displace objects. Even from the most dramatic birthplaces of GWs, the moments before, during and after the merger of two large compact binary objects such as binary black holes (BBH) and binary neutron stars (BNS), the amplitude of the resulting waves at cosmological distances from the event are on the order of 10^{-18} m when sensed by a 4 kilometer interferometer. For context, this displacement of space-time is 1000 times smaller than the nucleus of an atom.

Despite these challenges, GWs are detectable through the combined use of multiple instruments called interferometers. These Michelson interferometers with Fabry-Perot cavities utilize laser light that is sent to a beam splitter which causes half the light to go down each of two orthogonal arms. At the end of each of these arms is a reflective mirror. The beams reflect off these mirrors and recombine at the beam splitter, thereby sending a portion of the light toward a photodiode, and the remaining light back towards the laser. This photodiode outputs a current proportional to the average photon flux at the detector [2]. Any differential variation in the lengths of the arms will change the power seen at the photodetector. For example, if a GW passes through an interferometer perpendicular to its arms (i.e. incident from directly below or above it), the mirror of one arm will be expanded away from the photodiode as the other mirror is contracted towards the photodiode. As a result,

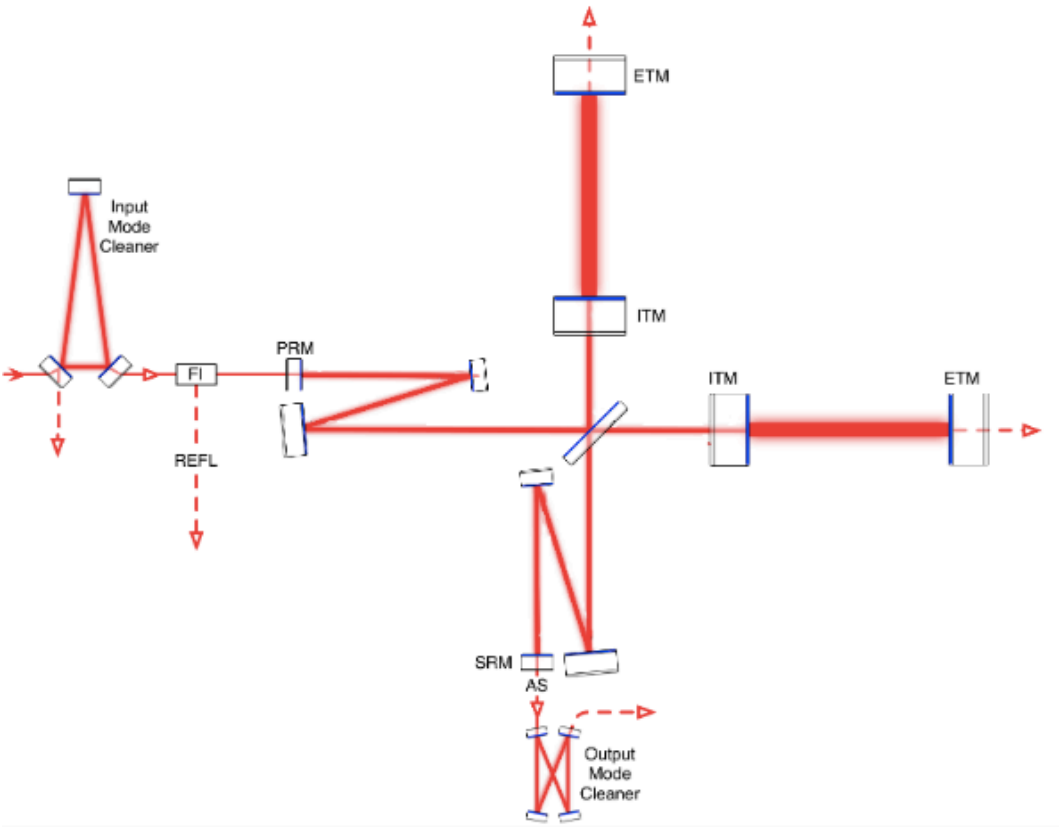


Figure 1.1: Illustration of the optical layout of Advanced Laser Interferometer Gravitational Wave Observatory [3].

the power seen at the photodiode modulates as the length of both arms change as a function of time. In this way, the length of the cavity affects the sensitivity of an interferometer as longer cavities yield larger phase delays between the light in the arms in the cavities.

Current ground-based detectors include the Advanced Laser Interferometer Gravitational Wave Observatory (aLIGO) [3], a diagram of which can be seen in 1.1, Advanced Virgo (AdVirgo) [4], and GEO600 [5]. Future ground-based detectors include KAGRA [6], which will be located in Japan, and LIGO India [7]. aLIGO consists of two 4 kilometer interferometers at Hanford, WA (H1) and Livingston, LA (L1) in the United States. Virgo consists of one 3 kilometer interferometer located in Pisa, Italy (V1). GEO consists of one 600 meter interferometer located in Hanover, Germany (G1). Despite the interferometers inherit abilities to sense the passing of a GW, many environmental (i.e. earthquakes, magnetic field, etc) and instrumental-sources of noise can move the mirrors of the interferometer or affect the light measured at the photo-diode in such a way as to either mimic or mask a GW. Therefore, great care has been taken to isolate the mirrors from as many known sources of transient non-gravitational wave noise, such as the addition of pendulums to reduce the motion of the mirrors and transducers which use seismometers to de-

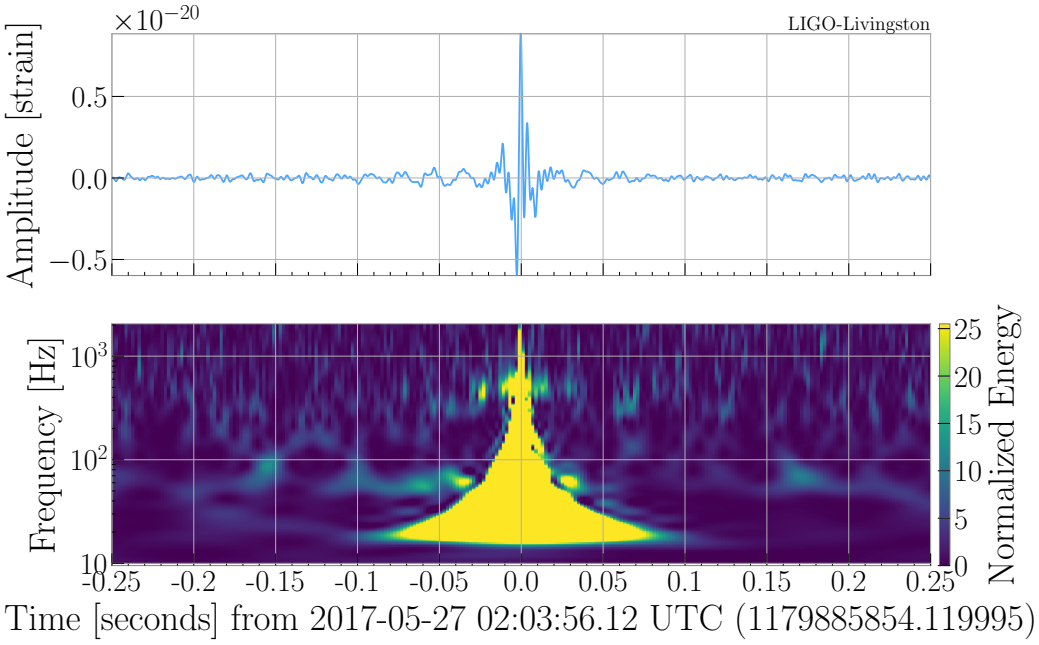


Figure 1.2: Example of an isolated Electrostatic Drive Overflow similar to that which occurred in the Livingston detector at the time of the BNS signal [13]. Top panel: Gravitational wave strain data that has been high and low pass filtered at frequencies of 50 and 290, respectively. Bottom panel: A special type of spectrogram that is made by performing a wavelet transformation on the timeseries data. The resulting image is referred to as a q-scan [14].

tect and counteract earthquakes. Starting in September 2015 and continuing to this day, the success of this work has been seen in the detection of GWs from compact binary coalescences of binary black holes [8–12] and binary neutron stars [13] by aLIGO and AdVirgo. These detections came while the aLIGO detectors collected data as part of their first observing run from September 2015 to January 2016 (O1) and second observing run from November 2016 to August 2017 (O2) and AdVirgo collected data between August 01, 2017 until August 25, 2017.

Despite these successes and efforts to isolate the interferometers from sources of transient non-gravitational wave noise, aLIGO and AdVirgo data is still contaminated with these transient artifacts. In the 51.5 days of O1 alone, approximately 10^6 glitches over a minimum signal-to-noise ratio (SNR) threshold of 6 were recorded [15]. Many of these noise features appear similar when viewed in a time-frequency space known as a spectrogram. Figure 1.2 shows the strain and spectrogram of one such excess noise occurrence at the Livingston detector that is similar to the noise feature seen in coincidence with the BNS detection [13]. Due to the sheer quantity of these excess noise features, however, the ability to comprehensively group or understand these morphological classes has been challenging for LIGO and Virgo scientists. Clean data is desirable for a number of reasons including, but not

limited to, having more "searchable" data, improving the confidence of detections by lowering the "loudness" of the background, and possibly elevating "marginal" detections to "gold-plated" detections. GW signals for which cleaner data would have the most impact are the so-called "unmodelled" signals. Unlike GWs from compact binary objects such as BNS and BBH, whose time-frequency characteristics can be meticulously modelled, gravitational waves from sources such as supernovae (SN) lack such comprehensive modeling. Therefore, without a model or template, algorithms designed to detect this type of gravitational wave are susceptible to the large quantity of excess noise transients. This effect can be seen from the search of the data surrounding and containing GW150914 by the unmodelled GW search algorithm coherent WaveBurst (cWB) [16, 17]. The paper describing the detection of GW150914 [18] discusses how the list of candidate events, or triggers, from cWB had to be divided into classes to handle the background distribution. In particular, the so-called C1 class included glitches with a low quality factor, that is the ratio of central frequency to bandwidth. The background distribution from this class was much louder than the other classes, implying that this search is less sensitive to GWs with similar morphology to specific types of glitches. Moreover, a number of data-quality algorithms can be utilized to determine the source of the excess noise [19–22] and try to mitigate its impact on a GW search, but never before have these been specially tuned to target specific families of excess noise. Therefore, a systematic and reliable grouping could improve the efficiency of the data-quality algorithms. However, as the detector undergoes changes over time, which, in turn, can change the data quality and types of excess noise in the data (as well as the sheer volume of excess noise), this classification effort presents a daunting challenge. To address this challenge, Gravity Spy [15], an innovative and interdisciplinary project that combines machine learning and citizen science was created to provide an adaptive and reliable program for classifying excess noise in aLIGO data.

This thesis explores a variety of topics related to GW detection from specific sources of unmodelled GWs such as a core-collapse supernova (CCSN) and the ability of Gravity Spy to aid in the search for these types of signals. Specifically, this thesis will introduce methods to detect unmodelled GWs that are employed by the open-source Python algorithm "X-Pypeline". The core methodology of "X-Pypeline" is based on the MATLAB algorithm "X-Pipeline" [23–26], but utilizes a number of modern open-source machine learning libraries to tune the analysis. This thesis is organized as follows. In chapter 2, I discuss the current implementation of the glitch classification system known as Gravity Spy and show how citizen science volunteers were able to identify new glitch families in LIGO data from O1. In chapter 3, I expand upon Gravity Spy's method for identifying new excess noise transients that appear as the detector changes over time, showing how the machine learning algorithm DIRECT, which stands for Deep Discriminative Embedding for Clustering of LIGO Data [27], impacts a LIGO or citizen scientist's ability to find

new glitches in LIGO data from O2. In chapter 4, I summarize the methodology that the unmodelled GW search algorithm "X-Pypeline" employs to find signals and reject glitches. I then demonstrate how Gravity Spy results and machine learning, in the form of random forests, convolutional neural networks, and Gaussian processes, can aid in the efforts to improve the sensitivity of the analysis, emphasizing the importance of using Gravity Spy to build the training sets that are used to train the machine learning models. In chapter 5, I discuss the possibility of a Galactic CCSN, the current theory surrounding CCSN, and how a neutrino trigger from the SuperNova Early Warning System (SNEWS) can guide an analysis of GW data. I focus on the importance of quality numerical relativity CCSN simulations when searching for a GW from a CCSN and how a neutrino detection of a Galactic supernova can guide a low-latency "X-Pypeline" analysis. Next, in chapter 6, I apply the methodology described in chapter 4 to a GW search of O1 data in response to a mock Galactic CCSN, finding that using machine learning techniques to tune the analysis and using Gravity Spy data to build the training sets used to train the machine learning models can improve the sensitive volume of the search by 10 to 15 percent. Finally, in chapter 7, I highlight some future work that can be done in both Gravity Spy and "X-Pypeline".

Chapter 2

Gravity Spy

2.1 The problem of non-Gaussian noise features

In order to detect gravitational waves, LIGO requires sensitivity to length fluctuations a thousandth the diameter of a proton in the 4-kilometer 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 450 Mpc [3]. This high sensitivity is achieved through exquisite isolation of the lasers, mirrors, and all sensitive components of LIGO from non-gravitational-wave disturbances. 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 [28].

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 [28, 29]. In the 51.5 days 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. The idea is that we want to prioritize those excess noise transients that are most similar to the our gravitational-wave signal model and which occur at the highest rates. 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 [30–35], 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, 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*[1] - 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 readily identifies new categories of glitches that arise as the detectors evolve, scales with an increasing number of unique glitch classes, and continually bolsters labeled sets of preexisting classes.

The Gravity Spy project couples human classification with machine learning models in a symbiotic relationship: volunteers give labels to spectrograms of known glitches which are then used 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 guide 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 [36]. A full release of Gravity Spy on October 12, 2016 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 chapter, we summarize the impact of glitches on LIGO data analysis and current efforts to mitigate their effects (Section 2.2). We then discuss the Gravity Spy project in full (Section 2.3), highlighting in particular data preparation for the project (3.1), the citizen science interface (3.2), machine learning algorithms

[1] https://github.com/Gravity-Spy/GravitySpy

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 2.4). Lastly, we comment on future prospects for the Gravity Spy project and its role in LIGO detector characterization (Section 2.5).

2.2 Characterization of transient noise in LIGO

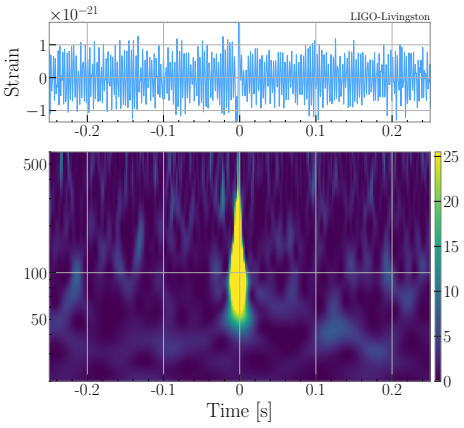
2.2.1 Impact of Glitches on Gravitational-Wave Data Analysis

Searches for transient gravitational-wave signals, especially those that are short duration, in LIGO's sensitive frequency band, and/or poorly modeled [28], 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. Despite this requirement, glitches occur at a high enough rate that accidental coincidence between the two detectors is non-negligible.

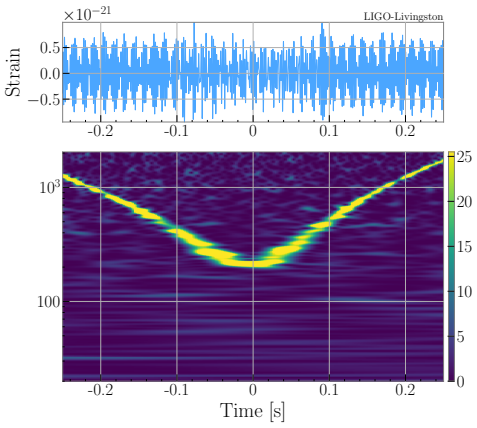
Glitches impact LIGO data analysis efforts in three critical ways. First, they increase the loudness of the background in gravitational-wave searches, which reduces the significance of candidate events. Even searches that utilize signal models to create discriminating signal statistics (e.g. compact binary coalescence searches [37, 38]) are afflicted by glitch occurrences. Second, glitches impact the recovery of astrophysical parameters from a gravitational wave source [39–41], 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 have occurred, they eliminate the data available to be searched for astrophysical signals. Therefore, identifying the cause of glitches and correcting the detector system which is 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.

2.2.2 Identifying Glitches

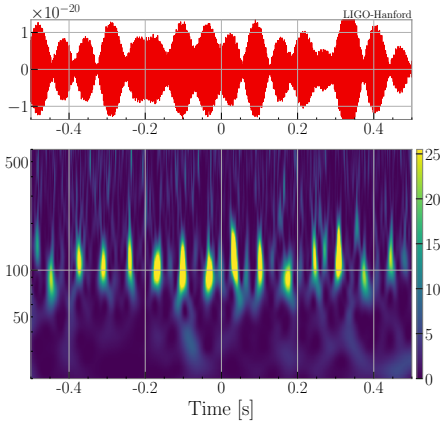
Several categories of glitches have been identified by the LIGO Scientific Collaboration (LSC), grouped by common origin and/or similar morphological characteristics [45–47]. Some of these categories have known causes, while others have causes yet to be identified. For example, three common morphological classes of glitches are shown in Figure 2.1. Blip glitches (a) are caused by unknown processes, whereas whistle glitches (b) are caused by radio signals at megahertz frequencies that beat



(a) Blip glitch



(b) Whistle glitch



(c) Scratchy glitch

Figure 2.1: Spectrogram representation of three 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 [28], 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 [39]. 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 [42]. Scratchy glitches (c) are believe to be related to the baffle in the input arm of the interferometer which has several openings used to intercept stray laser light and is referred to as the "Swiss Cheese Baffle" [43]. Prior to May 2017 most of the Hanford scratchy glitches seemed to have 12-15 nodes (bright dots) per second, but after the May 2017 work at Hanford, in which the "Swiss Cheese Baffle" was damped with rubber corks in order to limit its movement, most of the scratchy glitches began to appear with about 25 nodes per second [44]. After the dampers were added, the drop in the sensitive range of the detector related to this glitch disappeared [43]. These types of images are what volunteers in the Gravity Spy project classify, and what the associated machine learning algorithms use for training.

with Voltage Controlled Oscillators in the interferometer control system [42]. In addition, there are some glitches with probable causes, some of which have been discovered through the classification efforts of Gravity Spy. For instance, scratchy glitches (c) are believe to be related to the "swiss cheese baffle". Prior to May 2017 most of the Hanford scratchy glitches seemed to have 12-15 nodes (bright dots) per second, but after the May 2017 work at Hanford, in which the "swiss cheese baffle" was damped with rubber corks in order to limit its movement, most of the scratchy glitches began to appear with about 25 nodes per second [44].

Techniques have been developed to identify and categorize some classes 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 [48, 49]. Current approaches also search for statistical correlation between glitches in the gravitational-wave strain data channel and triggers in auxiliary channels [19–22]. 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.

2.2.3 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. An example of this would be to provide these algorithms with segments of times known to contain the Blip glitch, specifically, instead of segments of time known to contain loud excess noise, generally. This specificity would improve the ability to identify potential auxiliary channels that correlate with certain glitch morphologies, such as the Blip glitch, which in turn would contribute to identifying their source. We note here that the Blip glitch currently has know known source or correlated auxiliary channel.

2.3 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 [50]. 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 [51]. Another early astronomical project, Stardust@Home [52], led to the development of a general set of tools for citizen science projects known as BOSSA (now pyBOSSA[2]). The highly successful Galaxy Zoo (e.g. [53, 54]) and Zooniverse projects (e.g. [55–58]) 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 [59], 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 algo-

[2] pybossa.com

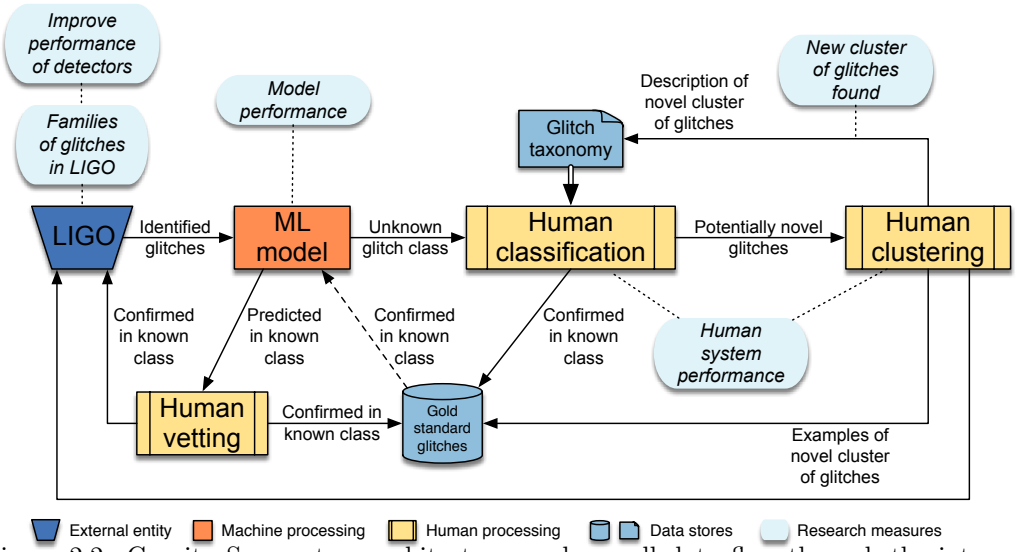


Figure 2.2: Gravity Spy system architecture, and overall data flow through the interconnected, interdisciplinary components of the project.

rithms. 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 classified 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. For instance, our user analysis provides information that can be used to treat classifications from certain users as more likely to be correct than those from other users which can lead to needing fewer users to label an image in order to retire it. We refer to an image as retired if it is decided that the image does not need to be labelled by anymore users because we are confident that it is of certain class. How the Gravity Spy system determines if an image has reached this point is discussed in detail in Section 2.3.3. Figure 2.2 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 [60], 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 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? Gravity Spy has provided the ability for collaborators to begin to answer a number of these questions and the reader is referred to [61–66] for more information.

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) [67–70]. 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.

2.3.1 Data Preparation

Data preparation for the Gravity Spy project (i.e. the link from LIGO to the rest of the project in Figure 2.2) 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 [48, 49], 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.

Data Selection

In order to tackle the first challenge, we only use glitches that satisfy 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 neglect times that were flagged for poor data quality (DQ), 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. 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

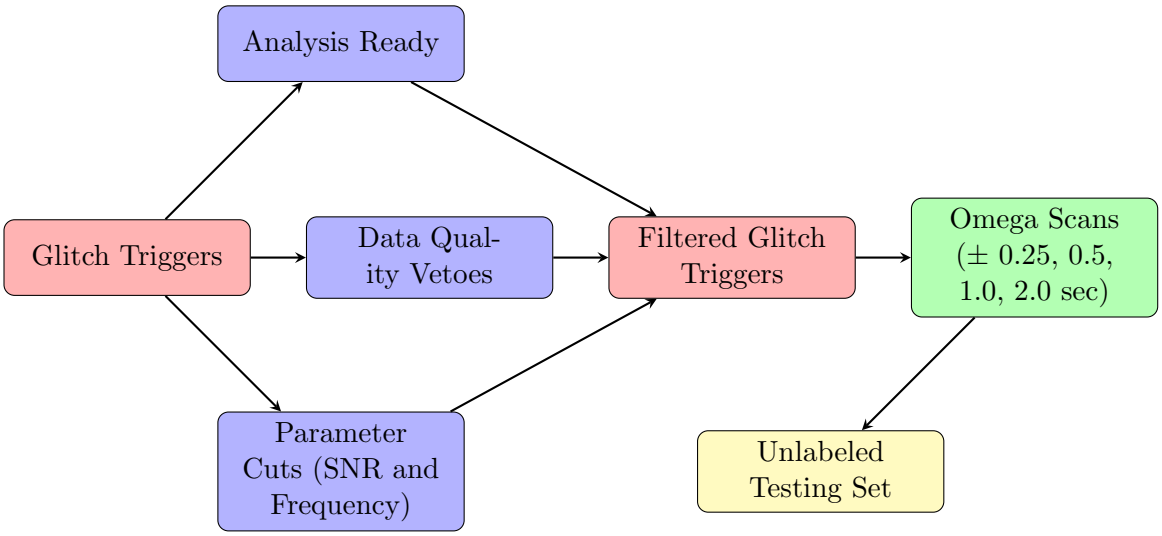


Figure 2.3: LIGO data processing and rendering workflow. The glitch triggers (red) are filtered through three separate criteria (blue). Analysis ready refers to time segments when the detector is in *lock* and in observing mode, meaning the state of the detector was adequate enough to be searching for gravitational waves and ready for data analysis. Those glitch triggers that survive are rendered into Omega Scans (green) of four durations and added to the unlabeled test set (yellow).

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 DQ vetoes [20, 28] 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.

Figure 2.3 provides a visualization of this work flow that occurs before a glitch ends up in the testing set of the Gravity Spy Project.

The gravitational-wave events GW150914 [18] and GW151226 [71] and gravitational-wave trigger LVT151012 [39] are not included in the Gravity Spy dataset. At the time these were the only confirmed gravitational wave detections. Hardware injections [72] 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.

Omega Scans

We met the second challenge by representing these glitches with Omega Scans [14]. 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:

$$Z = \frac{|X|^2}{\langle |X|^2 \rangle} \tag{2.1}$$

where Z is the normalized energy and $|X|$ is the Q transform magnitude of a tile [14].

As shown in Figure 2.1, each image has the glitch fixed at the center of the Omega Scan, and each glitch is visualized using four different time windows ($\pm$ 0.25, 0.5, 1.0, and 2.0 seconds) to accommodate both long-duration and short-duration glitches. Human volunteers and machine learning algorithms are presented all four time durations of each glitch for classification purposes.

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. [45–47, 73, 74]) 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.

Listed below are the 22 classes of glitches included in this dataset. Examples from each class are shown in Figure 2.4. The names associated with these classes and the typical morphology of the glitches belonging to each class were set by a

combination of LIGO scientists, Gravity Spy scientists, and in some cases, citizen scientists. Many of the glitch classes listed here, and in some cases their physical causes, were previously identified in work to characterize the LIGO data [28, 29, 42]. Glitch classes for which LIGO scientists have uncovered or fixed the cause are also highlighted below.

1. **1080 Lines**
   At LIGO Hanford during O2, there was a steady stream of glitches with central frequencies around 1080 Hz. In Omega Scan images these appeared as a string of yellow dots, sometimes connected to form a line, and sometimes more sparse. These glitches were greatly reduced following a configuration change that increased the gain of a control loop for the LHO output mode cleaner's length (see LHO electronic logbook entry 33104 [75]). 1080 Hz lines were most prevalent between 11 October 2016 (i.e., during the engineering run before the start of O2) and 14 January 2017.

2. **1400 Ripples**
   These are somewhat strong short-duration glitches at around 1400 Hz. They can appear isolated (one glitch per image) though sometimes multiple glitches are seen within a 4-second-long image. So far, the source of these glitches is unknown. Both the name of the class and the images used to create the training set for the class were motivated by collections of glitches and forum discussions by Gravity Spy citizen scientists.

3. **Air Compressor**
   These short-duration glitches look like a thick line centered at a frequency of 50 Hz. At LIGO Hanford, these were found to be related to air compressor motors switching on and off at the end stations. This issue was solved on September 29, 2016 by replacing the vibration isolators (rubber feet) on the air compressors (see LHO electronic logbook entries 22081 [76] and 21436 [77]).

4. **Blip**
   These short glitches usually have a duration of around 40 ms, frequencies between 30 and 500 Hz, and typically appear as a narrow, vertical and symmetric 'teardrop' shape in time-frequency domain [28]. They appear in both Hanford and Livingston detectors and their root cause is unknown.

5. **Chirp**
   A "chirp" is the characteristic time-frequency shape created by gravitational waves from inspiraling compact objects, sweeping upwards in frequency over time. The only chirps in this Gravity Spy dataset are so-called "hardware injections" [72], simulated gravitational wave signals physically added to the detectors for testing and calibration purposes. Additionally, though it was

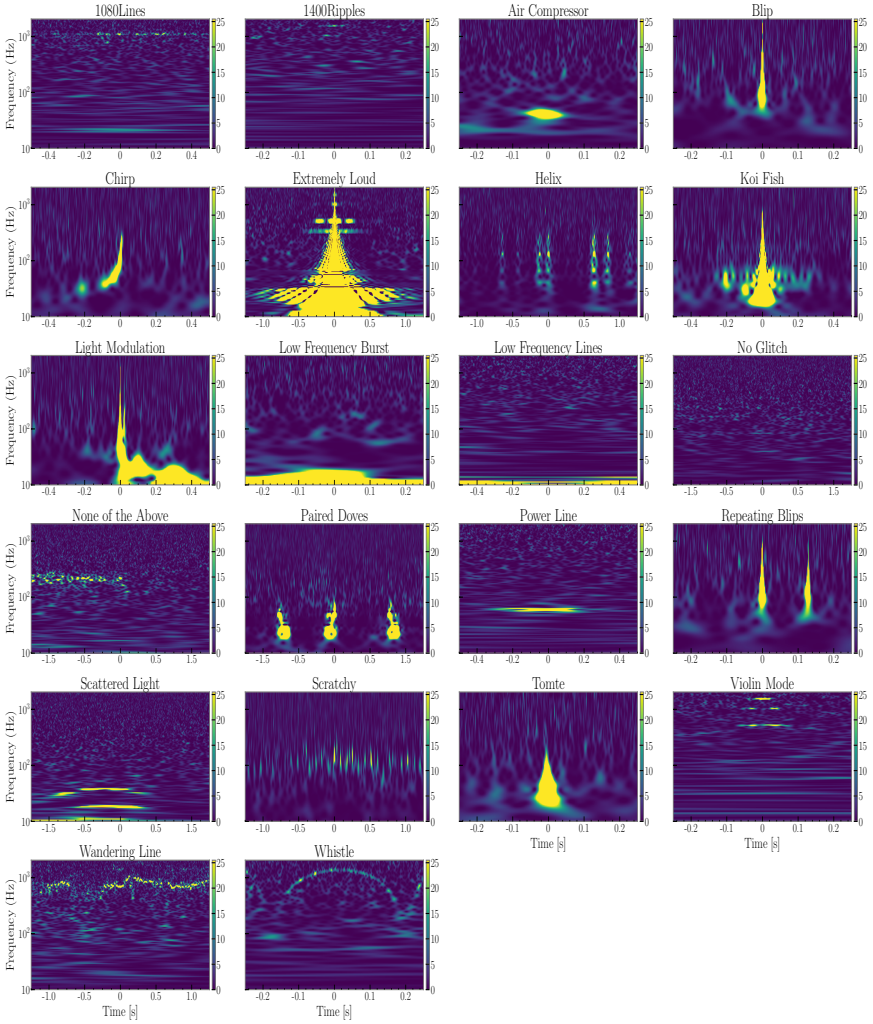


Figure 2.4: Omega Scan images for example members of each class within the Gravity Spy dataset. From top left to bottom right; row one: 1080Lines, 1400Ripples, Air Compressor, Blip, row two: Chirp, Extremely Loud, Helix, Koi Fish, row three: Light Modulation, Low Frequency Burst, Low Frequency Lines, No Glitch, row four: None of the Above, Paired Doves, Power Line, Repeating Blips, row five: Scattered Light, Scratchy, Tomte, Violin Mode, row six: Wandering Line, Whistle. Chirp is not strictly a glitch but is an important category as real gravitational waves can appear in our data stream and the example of None of the Above is only one example as this class can have various forms.

not included in this dataset, Gravity Spy users were recently presented with images of the gravitational wave signal GW170104 [10].

6. **Extremely Loud**
   This is a catch-all category for glitches that result from a major disturbance in the detectors. They usually span much of the spectrogram and have extremely high energies. High energy glitches from other categories (e.g., Koi Fish) that saturate most of the image are also placed in this category.

7. **Helix**
   This class contains glitches that resemble a vortex, occur at intermediate frequencies, and often come in groups. Their origin is unknown, but it is possible that they are related to glitches in auxiliary lasers (called photon calibrators) that are used to push the LIGO mirrors by known amounts to calibrate the detectors. Both the name and the images that were used to create the training set for this class were motivated by the collections and forum comments made by Gravity Spy citizen scientists.

8. **Koi Fish**
   Koi Fish glitches are similar to Blip glitches, but they resemble a fish with the head at the low frequency end of the plot, pectoral fins around 30 Hz, and a thin tail around 500Hz. They are possibly a sub-class of blip glitches.

9. **Light Modulation**
   The morphology of this glitch is not always the same, but it often looks like several bright spikes in close succession, often with low frequency noise at the same time. They are caused by amplitude fluctuations in the signal used to generate the 45 MHz optical sidebands (used to control the length and alignment of some of LIGO's optical cavities).

10. **Low Frequency Burst**
    This is a catch-all category for loud, short-lived, low-frequency noise. Though multiple morphologies make up this category, they often resemble small humps with a nearly triangular shape growing from low frequency to a peak and then dying back down in a second or two. This glitch class was common in LIGO Livingston's first observing run. During its first observing run, LIGO Livingston was plagued by a non-linear effect that causes bursts of noise at low frequency (up to 20-30Hz) which last up to a few seconds.

11. **Low Frequency Line**
    This category appears as horizontal lines at low frequencies. They are distinct from Low Frequency Bursts because they are more long-lived and from Scattered Light (see below) because they do not vary noticeably in frequency over long durations.

12. **No Glitch**
    The No Glitch category is designated for spectrograms that have no apparent or obvious transient noise structure visible in the Omega Scan. The lack of visibility could be due to a number of factors including issues displaying glitches whose peak frequency is high (2 kHz) and duration is short. Because Omega scans implement what amounts to a log tiling (and the images are on a log scale) these glitches could be visibly shrunk to the point that they blend into the background. This class is kept so volunteers do not make arbitrary classifications of spectrograms with minimal activity.

13. **Paired Doves**
    This category, inspired by the collections and forum comments made by Gravity Spy citizen scientists, is a collection of repeating glitches that look similar to chirps at low frequencies, and alternate between increasing and decreasing frequency. These glitches are believed to be related to periods of excess 0.4 Hz motion of the beamsplitter at LIGO Hanford (see LHO electronic logbook entry 27138 [78]).

14. **Power Line**
    Power Line glitches usually look narrow in frequency, and last for about 0.2-0.5 seconds in time, centered around 60 Hz or one of its harmonics. They are due to glitches in the United States mains power (alternating current at a frequency of 60 Hz). Because of this, quasi-periodic frequencies of 60 Hz and its harmonics (120, 180, etc.) are present in the LIGO data.

15. **Repeating Blips**
    This category is for blip glitches that repeat in the Gravity Spy images. Though this category encompasses all cadences of repetition, they are often found to repeat every 0.25 or 0.5 seconds.

16. **Scattered Light**
    Scattered light glitches come in many different morphologies, though they are often low-frequency, long duration, humpy glitches that look like one or several curved lines stacked on top of each other. They are due to light from the main LIGO laser beam path being scattered off moving objects and re-combining with the main beam but with a rapidly varying phase [79].

17. **Scratchy**
    This is a series of short-duration repeating glitches (often with 10-30 glitches per second) with intermediate frequencies, often lasting several seconds. They occur mostly at LIGO Hanford and some of them are related to scattered light from a "baffle" used to intercept stray laser light. This was determined due to the change in the characteristic of the glitch after damping was done to

this "baffle" [44]. Recognizing them with usual machine learning classification methods is non-trivial because their morphologies can be scattered through a wide range of feature space. The name Scratchy comes from how this type of glitch sounds when the signal is converted to an audio signal.

18. **Tomte**
    These glitches are similar to, or possibly a sub-class of blips. They are lower-frequency glitches that are usually triangular in shape, and look like the hat worn by a garden gnome.

19. **Violin Mode Harmonic**
    This category appears short and dot-like, and occurs at 500 Hz and multiples (harmonics) of this frequency. LIGO's main mirrors are suspended by thin glass fibers that have resonances like that of a violin string. These glitches occur at the frequencies of those resonances.

20. **Wandering Line**
    These are lines that last on the order of minutes to an hour and meander in frequency. Some example causes of wandering lines include motors (such as vacuum pumps) that do not have fixed frequencies and beats between higher frequency signals that have some frequency variation.

21. **Whistle**
    Whistles have a characteristic "W" or "V" shape that sweeps down to lower frequencies. They are caused by radio frequency signals (i.e. signals at MHz frequencies) that interfere and beat with the LIGO Voltage Controlled Oscillators. The name Whistle comes from how this type of glitch sounds when the signal is converted to audio.

22. **None of the Above**
    This category is a catch-all for glitches that do not fit into the other 21 categories, and therefore has much variability in its morphology.

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. 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 as-

Class	Hanford	Livingston
1080Lines	327	0
1400Ripples	0	81
Air Compressor	55	3
Blip	1452	369
Chirp	28	32
Extremely Loud	266	181
Helix	3	276
Koi Fish	517	189
Light Modulation	511	1
Low Frequency Burst	166	455
Low Frequency Lines	79	368
No Glitch	65	52
None of the Above	51	30
Paired Doves	27	0
Power Line	273	176
Repeating Blips	230	33
Scattered Light	385	58
Scratchy	90	247
Tomte	61	42
Violin Mode	141	271
Wandering Line	42	0
Whistle	2	297

Table 2.1: Breakdown of morphological categories in the Gravity Spy training set, indicating the number of training set samples of each class that comes from Livingston detector data and Hanford detector data. Note that some of the glitches are detector dependent.

sure 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 $\sim 80\% - 90\%$, they were useful in differentiating the unlabeled dataset into morphologically-similar classes, thus fostering an easier by-eye classification process. The idea is that this first machine learning algorithm provided rough bins within which the large amount of previously unlabelled data could be placed. Searching these bins allowed the Gravity Spy team to label the data faster and thereby provide more training set samples for each class. Moreover, as will be described in Section 2.4.2, during early stages 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 7932 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 22 classes, with exact proportions shown in Table 2.1. As can be seen from the table, not every class occurs in both detectors. Therefore, although it is not necessary nor has been done at the present moment, in the future it may be useful to have detector dependent training sets that are used to train detector dependent convolutional neural networks. Given that each glitch is imaged at a maximum duration of 4 seconds, this amounts to having spectrograms which contain 8.58 hours (0.7%) of O1 data [39]. Since every glitch does not last 4 seconds, we do not necessarily have 8.58 hours worth of glitches.

2.3.2 Citizen Science

Once we make the four spectrograms of a given LIGO glitch, it is classified by Gravity Spy volunteers. These volunteers make two different contributions to the system. First, they provide a label to the glitch which combined with other user labels and the label provided by the machine learning model determine the class of glitch (see Section 2.3.3 for more details). Second, if they cannot label the glitch as one of the known classes, they determine if it unique and prevalent enough in the Gravity Spy dataset to warrant a new class of glitch. In this way, these users make up what is considered the human-classification unit of the system (yellow boxes in Figure 2.2).

User Interface

The user interface for Gravity Spy was created using the Zooniverse DIY Project Builder[3], which enables anyone to build their own Zooniverse citizen science project

[3] Zooniverse.org/lab

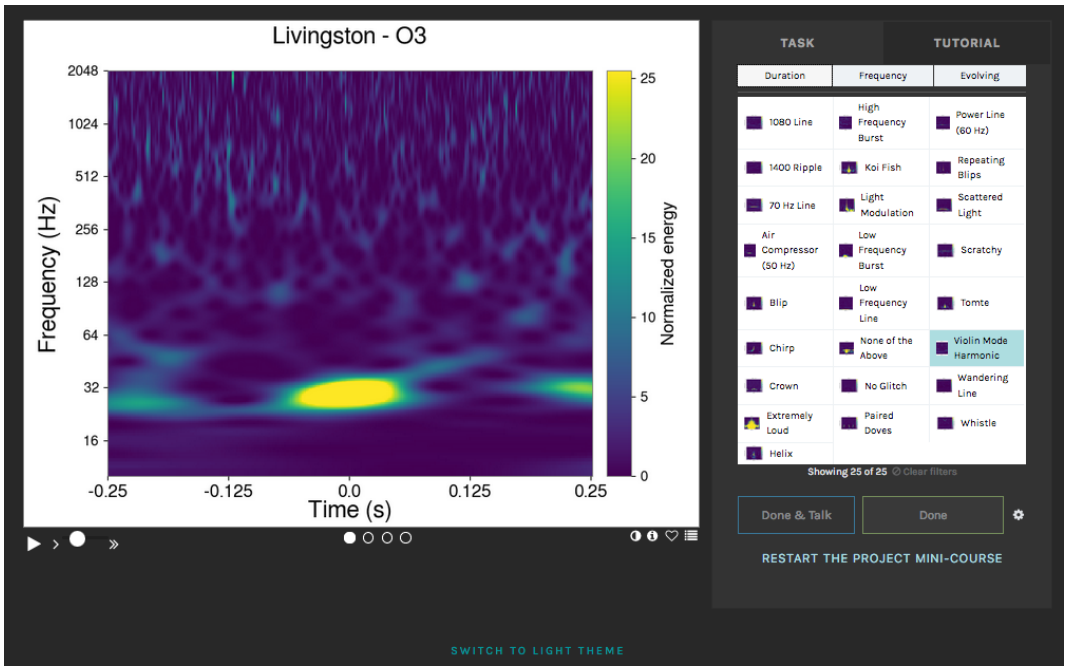


Figure 2.5: Gravity Spy user interface. This image shows the *Black Hole Merger* workflow (see Section 2.3.2), with all 22 currently designated categories as options.

for free through a set of easy-to-use, browser-based tools. The Gravity Spy classification interface containing the currently-known 22 glitch classes as options is shown in Figure 2.5. Example images of each glitch morphology can be found on the Gravity Spy website[4]. Through this interface, volunteers are shown individual Omega 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 2.5) .

[4] gravityspy.org

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. In Section 2.4.2 and 2.5, we discuss how this worked in practice for two of the current Gravity Spy classes, the Helix and Paired Doves classes, as well as how this might work generally. This allows the Gravity Spy glitches classes to evolve and follow changes in the glitch types that occur in the LIGO detectors.

Volunteer Training

A key question in citizen science is how reliably volunteers perform the classification task, known as *results quality*. Zooniverse's 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 [36].

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 [80, 81]. 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.

Gravity Spy advances the current state of the art citizen science project by using 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 of glitch classes and sophistication of features within the classification interface, intended to improve their ability to classify glitches [82]. 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. Specifically, by allowing the machine learning model to pre-classify all of the images that are provide to the volunteers, we have a sense of which images are easier or harder than others. In this way, we can show newer users the images that should be more straightforward to classify (at least from the perspective of the machine learning model), and show more experienced users images that the machine learning model struggled to classify.

For example, upon joining the 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, Blip and Whistle. 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's reliability, it can also assign tasks based on the capabilities of each volunteer.

Workflows

Glitches are first sent through the machine learning classifier, which is trained on a set of images pre-classified by experts (see section 2.3.1) 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 2.7.

Similarly, based on their expertise and reliability level as determined by their performance in classifying (described in section 2.3.3), 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.

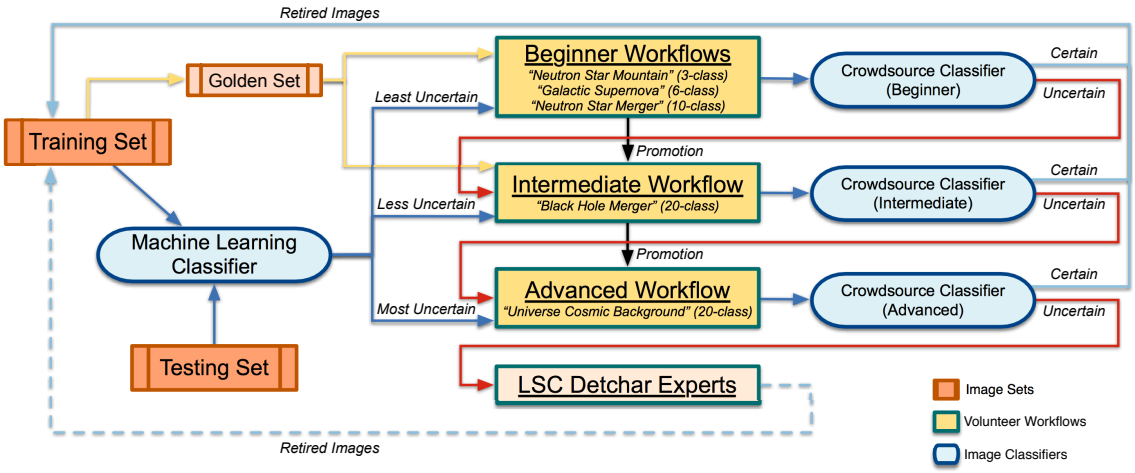


Figure 2.6: Movement of images and volunteers through the Gravity Spy project. Yellow boxes represent the multiple workflows within the project (including the images which are forwarded to experts within the LSC), blue boxes represent the machine learning and crowdsourcing image classifiers, and orange 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.

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 classifier.

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. User performance is gauged by tracking their classification of *golden images*, which are a subset of labeled images from the training set. Once a user classifies enough images correctly, they will be prompted to progress to more advanced workflows. The user weighting and its impact on retirement is discussed further in Section 2.3.3.

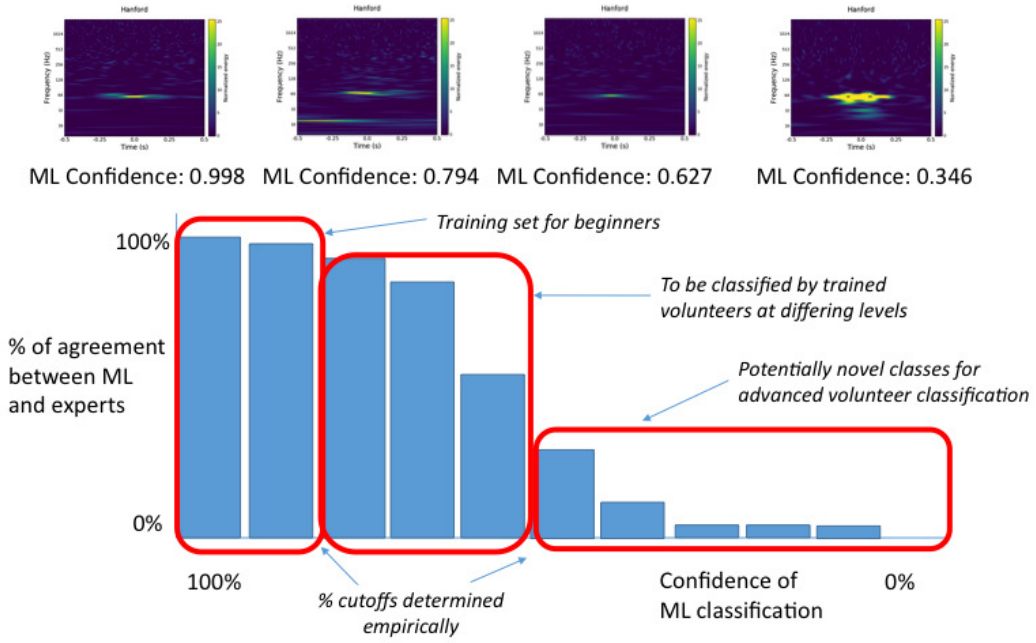


Figure 2.7: Relationship between machine learning confidence in glitch classification (x-axis) and proportion of images from that class assessed by human volunteers at different skill levels. Example glitches classified as a single class ("Power Line" glitches) with differing machine learning confidence scores are shown above.

2.3.3 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.

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 [83]. 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 2.8 illustrates the machine learning process used.

Many studies (e.g. [84, 85]) 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 window-panel like form where each panel is one

of the durations 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 22 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

$$o_i^c = \frac{e^{w_c x}}{\sum_{c=1}^{C} e^{w_c x}} \qquad \text{for } c = 1, \cdots, C \tag{2.2}$$

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 c^{th} node in softmax layer. C represents the total number of classes, in our current case 22. 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:

$$\mathbf{p}_i^{\mathrm{ML}} = [o_i^1, \cdots, o_i^c, \cdots, o_i^C] \tag{2.3}$$

Specifically, the Gravity Spy CNN uses four separate convolutional layers (see Figure 2.8). Each has 128 kernels with size 5×5, and a max-pooling kernel with size 2×2 is applied to each convolutional layer. As mentioned above, we apply the ReLu activation function to the output of each convolutional layer. Then, we flatten the output of these four convolutional layers into the a single layer to which we connect

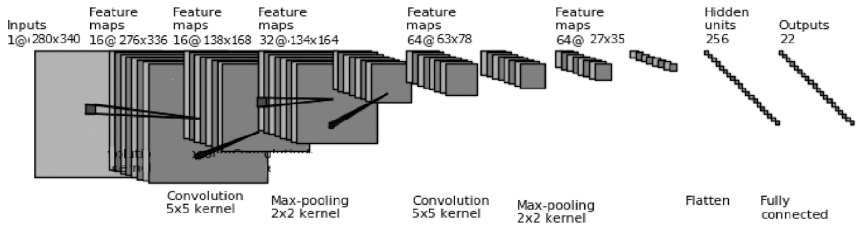


Figure 2.8: Deep CNN used for glitch image classification. The network has been introduced on top of the four merged glitch durations. Dimensions of the kernels and feature maps are in units of pixels.

Table 2.2: Model Specification

Input 280×340
5×5 Convolutional layer (16) with reg.
2×2 Maxpooling, 0.5 drop-out
5×5 Convolutional layer (32) with reg.
2×2 Maxpooling, 0.5 drop-out
5×5 Convolutional layer (64) with reg.
2×2 Maxpooling, 0.5 drop-out
5×5 Convolutional layer (64) with reg.
2×2 Maxpooling, 0.5 drop-out
Fully connected (256), 0.5 drop-out
Softmax (22)

a fully connected layer with 256 nodes and a softmax layer with 22 nodes (equal to the number of classes). All these details of the CNN used in Gravity Spy are shown in Table 2.2. The parameters of this architecture were obtained based on extensive experimentation and guidance from literature.

Training the model

In order to train any machine learning algorithm, the weights of the kernels in the convolutional layers must be learned. In order to do this, one must select a function for our model to optimize. For Gravity Spy, we optimize a loss function defined on the training data, using cross-entropy:

$$\text{loss} = -\sum_{j=1}^{N}\sum_{c=1}^{C} y_j^c \log o_j^c \tag{2.4}$$

where o_j^c is the model's output for class c when the j^{th} training sample is given to the network, y_j^c is equal to unity if the j^{th} sample is from class c, otherwise it is zero, and N and C are the total numbers of the training samples and classes, respectively. A method must be selected by which we can minimize or optimize our model with respect to the loss function above. To do so, we use gradients which can be defined in various levels of "steepness", i.e. learning rates, through which we find the minimum of the loss function. Choosing a learning rate dictates the trade off between the speed and accuracy at which you find this minimum. In first passes at optimization, it is likely that the randomly selected weights will be far off from the minimum of the loss function so it finds step gradients to take towards the minimum which is sensible, but, as you get closer to the minimum, you want to make smaller steps along these gradients as you try to converge on the minimum. Therefore, to achieve this optimisation logic, the Adadelta [86] optimizer is used. This optimizer monotonically decreases the learning rate and shows good performance in our experiments. More details about the machine learning image

classifier and experiments can be found in [87].

Crowdsource Classifier

As noted in Sectin 2.3.2, the system will maintain a model of each volunteer's 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's 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 as the 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 [88].

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 k^{th} volunteer, where C denotes the total number of classes. An entry of this matrix, m^k_{pq} gives the number of samples belonging to class p labeled as belonging to class q by the k^{th} volunteer. All entries will be initiated as 0 and updated when an image from the golden set is labeled by the volunteer. In the event a volunteer at some point labeled an image whose true label at the time was unknown, but that image is eventually retired with a true label, then the users confusion matrix will retroactively update based on the label they had given for that image.

Using a volunteer's confusion matrix $\mathcal{M}^k$, a reliability measure is defined for volunteer k as the vector $\mathbf{a}^k = [\alpha^k_1, \cdots, \alpha^k_c, \cdots, \alpha^k_C] \in \mathbb{R}^{C\times 1}$, where α^k_c quantifies the reliability of volunteer k in classifying samples of class c. It is defined as:

$$\alpha^k_c = \frac{m^k_{cc}}{\sum_{j=1}^{C} m^k_{cj}} = p(\hat{y}^k = c|y = c) \qquad \text{for} \quad c \in \{1, \cdots, C\} \tag{2.5}$$

where α^k_c is also equal to the probability that the k^{th} 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 volunteer's reliability, the classification of a test sample of images using multiple volunteer labels is determined. A test image is initially provided to the machine learning classifier which outputs a probability vector $\mathbf{p}^{\text{ML}}_i$. Depending on this machine learning confidence, the test sample is forwarded to volunteers in a given workflow, who provide classification labels to this image. In the following paragraphs, we propose a method which uses the machine learning probabilities and volunteer classification labels to predict the true label [88].

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:

$$\tilde{y}_i = \text{argmax}_j \frac{p(y_i^{\text{cr}} = j|\hat{y}_i^1, \cdots, \hat{y}_i^{R_i}) + \mathbf{p}_i^{\text{ML}}(j)}{\sum_{j=1}^{C} p(y_i^{\text{cr}} = j|\hat{y}_i^1, \cdots, \hat{y}_i^{R_i}) + \mathbf{p}_i^{\text{ML}}(j)} \tag{2.6}$$

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

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 Eq. 2.6 and putting them in a vector $\mathbf{y}^i = [y_1^i, \cdots, y_C^i] \in [0, 1]^C$, 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. In practice, we determined class dependent threshold t through trial and error. We started with a reasonable probability criterion of 90% per class and then evaluated by eye the rough false alarm rate of the final label given to the images using the above described method versus their true label as determined by the experts. A more quantitative identification of these class dependent thresholds and the trade of between the quality of the retired images and the speed at which we can retired them is in progress [89].

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 Eq. 2.5, the $\mathbf{a}^k$ vector is calculated from the confusion matrix $\mathcal{M}^k$. Defining $\mathbf{T}_\alpha = [T_1, \cdots, T_C]$, , the following decision rule is used for promotion of a volunteer:

$$\mathbf{a}^k \geq \mathbf{T}_\alpha \tag{2.7}$$

If all the values of the vector $\mathbf{a}^k$ exceed the threshold values in $\mathbf{T}_\alpha$, the volunteer is

promoted to the next level. If not, they will need to do more correct classifications to be promoted.

2.3.4 Socio-Computational Research Support

Finally, the socio-computational research component (yellow boxes in Figure 2.2) will allow for systematic measurement and experimentation with the performance of project components. Our first 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. Results of this study can be found in [66].

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). Again, this work is in progress and is described in more detail in [89].

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 [61].

2.4 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.

2.4.1 Initial Machine Learning Performance

As discussed in Section 2.3.1, the initial machine learning training set consists of 7932 total glitches from 22 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 94.1%. We define the classification accuracy of the algorithm as simply the rate of correct classifications with respect to the testing set.

As can be seen in the training set breakdown (Table 2.1), 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 2.9.

As one can observe from Figure 2.9, 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") 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 21 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

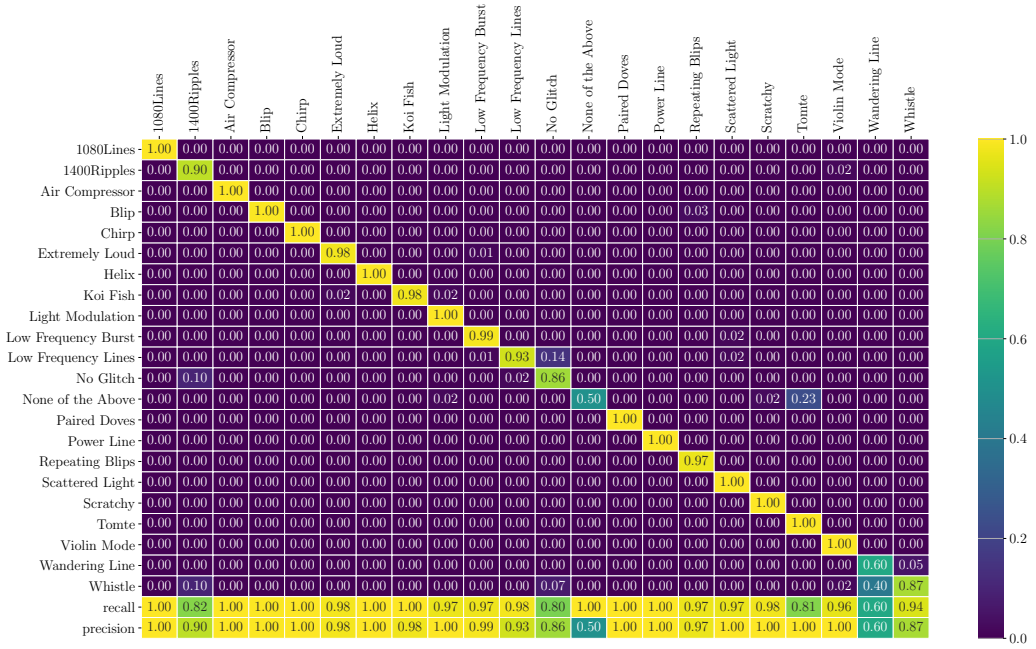


Figure 2.9: Confusion matrix for the 22 glitch classes in the testing set classified using CNNs, with recall and precision values appended below for reference. The x and y axes represent the predicted and true classes, respectively, and the confusion matrix is normalized by the total number of glitches 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.

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.

2.4.2 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's talk forum and fostered excitement and intrigue for the nascent field of gravitational-wave astrophysics. The work culminated in the discovery of multi-

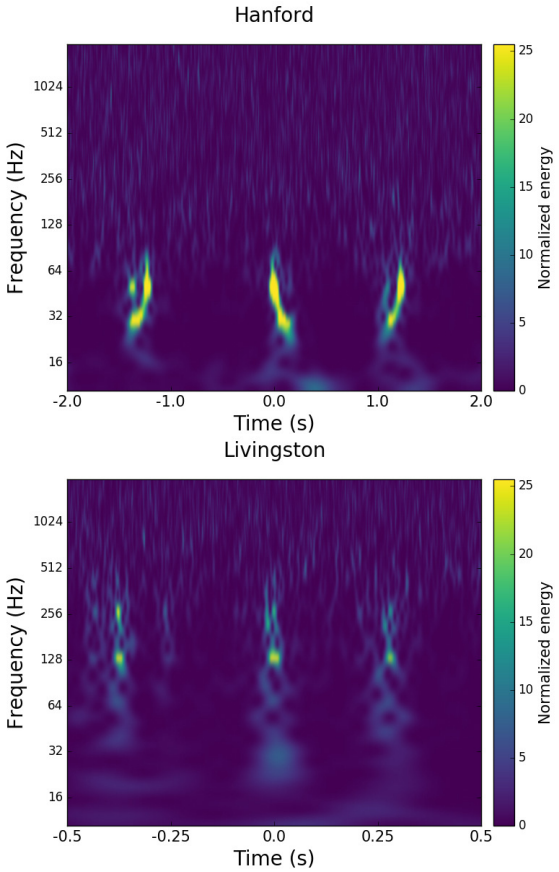


Figure 2.10: Two new O1 glitch classes uncovered during Gravity Spy beta testing: "Paired Doves" (left) and "Helix" (right). "Paired Doves" [78] resemble chirps, but alternate between increasing frequency and decreasing frequency. These glitches are potentially related to 0.4 Hz motion of the beamsplitter at the Hanford detector. "Helix" [90] are possibly related to glitches in the auxiliary lasers (called photon calibrators) that are used to push the LIGO mirrors and calibrate the detectors.

ple new and substantial glitch categories from LIGO's first observing run, including glitches which would later receive the names "Paired Doves" [78] and "Helix" [90]. 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 [91, 92].

2.5 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. 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 LIGO 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 2.3.1. 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. An example of this clustering happening in practice could be from many different images labelled as "None of the Above" by users receiving the same follow up description of "A line feature at 200 Hz". We can then use this description to identify all of the images corresponding to this new glitch and create a new class. 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.

Chapter 3

Classifying the unknown: discovering novel gravitational-wave detector glitches using similarity learning

Though the classification of glitches through ML approaches has shown promise, these approaches suffer from some shortcomings. First, the supervised ML methods, where previously known classes of transients are given as a training set to the algorithm, have no immediate way to identify other classes also present in the data. Alternatively, unsupervised ML methods, where the algorithm seeks to learn the discriminative features of the data set in order to create its own classes or clusters of similar data, have the downside of decoupling the analysis from the understanding of how known classes relate to the detector. For example, the Whistle glitch, which is described in Chapter 2, has a known source but can render itself in the gravitational wave data in morphologically diverse ways. These diverse renderings would likely be considered by an unsupervised algorithms as separate classes, when in reality, they are the "same" in the sense that they all came from the same source. Moreover, unsupervised methods inevitably suffer from the need to validate the self-identified classes, as the clusters are far from exclusive because the features the algorithm learns from the unlabeled data set are not discriminative enough. In the case of glitches from gravitational wave data their is some "correct" number of clusters or classes. This number, however, is unknown and, most likely, the number of clusters you request from the unsupervised algorithm will be too few (leading to classes full of too many different glitches) or too many (leading to clusters that split the same class unnecessarily into separate classes, see example of the Whistle glitch above). Both supervised and unsupervised ML techniques have merits, but neither is a perfect solution.

In an effort to address the glitch classification problem, we previously introduced *Gravity Spy* [15].This combines the crowd-sourcing power of citizen science with the rapid classification ability of ML [93] to support the characterization of glitches in GW data. Gravity Spy is hosted on Zooniverse, a leading online platform that has enabled over 1.5 million citizen scientists to analyze scientific data. Gravity Spy users are asked to classify time–frequency plots depicting glitches into one of a number of classes. The large number of people supporting this work provide data sets of known glitch classes, which are then used as training sets for the ML algorithms. The ML algorithms can then rapidly classify the entire data set of known glitches. These data sets are then used for the purpose of long term trend studies as well as targeted auxiliary channel follow-up, e.g comparing humidity at the detector with the rate of the blip glitch [94]. Although the classification and verification of known classes has proven effective in Gravity Spy, it remains challenging to collect sufficient numbers of novel glitches to identify new classes. We note that that unsupervised clustering has been shown to classify new types of glitches, e.g., "Reverse Chirp", shown in Figure C2 of [34].

To solve this problem we employ techniques from transfer learning. Transfer learning applies the knowledge from a labeled data set to an unlabeled data set with different features. In this case, we are interested in transferring the knowledge of what makes the known glitch classes in Gravity Spy similar and different from each other to the domain of images that do not belong to any known class. Although this method proves useful in helping the algorithm extract more discriminating features that make for cleaner clustering of the unlabeled data, they are still not discriminate enough to confidently contain a single new class of glitch. For example, in the Gravity Spy project we consider distinct the glitch classes Koi Fish, Blip and Tomte, and unsupervised clustering algorithms tend to lump all these distinctive classes together. Therefore, combining the feature space obtained through transfer learning techniques with human controlled clustering of this space may prove the most effective way to rapidly identify new glitch classes.

We introduce a new method for the rapid identification of novel transients that combines techniques within the field of transfer learning with the crowd-sourcing power of Gravity Spy. In Section 3.1, we discuss the specifics of our transfer learning algorithm. We then discuss the new proposed infrastructure for Gravity Spy in Section 3.2. In Section 3.3, we highlight the impact the proposed methodology could have had on discovering two sets of new LIGO glitches from the second observing (O2) run. In Section 3.3.1, we summarize the impact of different settings of the transfer learning algorithm on the discriminative ability of the feature space. In Section 3.4, we discuss future iterations of the Gravity Spy project and its role in GW detector characterization.

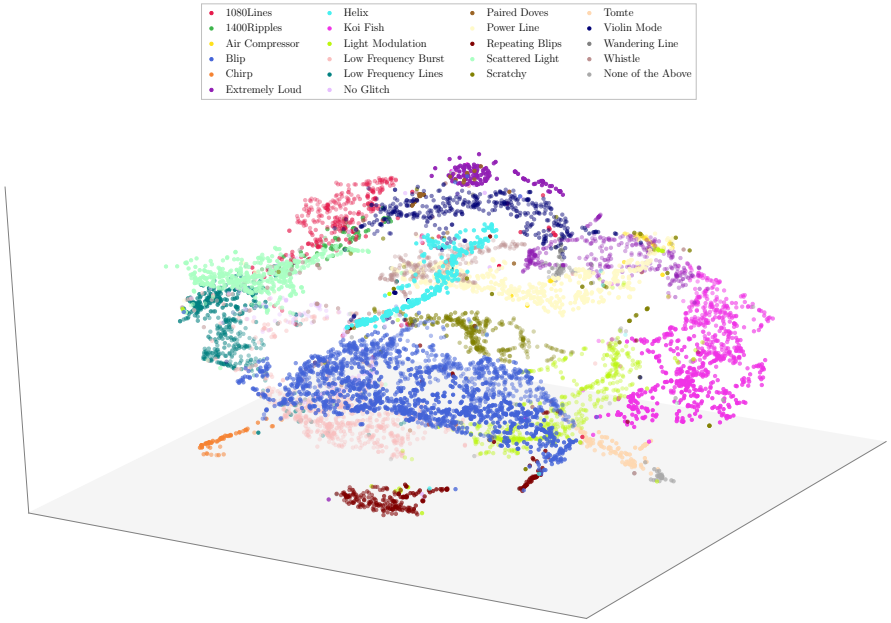


Figure 3.1: Visual representation of the training set in the DIRECT feature space using the t-distributed Stochastic Neighbor Embedding (t-SNE) statistic. This metric is purely designed to project groups of samples in the N-dimensional feature space into 3 dimensions and has no physical meaning.

3.1 Transfer Learning

Transfer learning applies knowledge obtained from a model that was trained on one data set to another data set. Specifically, for our method, we hope to transfer knowledge about what makes the spectrograms of the known Gravity Spy classes similar and different from each other to the unlabeled Gravity Spy glitches. We anticipate that this knowledge will enable a better clustering of these glitches that will lead to the discovery of new classes.

To accomplish this goal, we must first train an algorithm designed to model the similarity and differences between images on the known set of Gravity Spy glitches. For this analysis, we use the transfer-learning algorithm DIRECT [27] to quantify similarity between Gravity Spy images. In short, this algorithm solves for a nonlinear embedding function f_θ, i.e. the discriminative feature space, by using a deep neural network. Using pairs of labeled images as input, the neural network is trained by solving for the f_θ that minimizes the function

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{N} l(y^i, x_1^i, x_2^i) \\ &= \sum_{i=1}^{N} y^i \mathrm{dist}\left(f_\theta(x_1^i), f_\theta(x_2^i)\right) \\ &\quad + (1-y^i)\max\left\{0, m - \mathrm{dist}\left(f_\theta(x_1^i), f_\theta(x_2^i)\right)\right\} . \end{aligned} \tag{3.1}$$

Here N is the number of training pairs; x_1^i and x_2^i are the first and second items of the i-th pair; y^i is the binary label of the i-th pair, which is one when the two items of the pair belong to the same class and zero when they belong to different classes; dist is a distance function (such as Euclidean or cosine), and m is the margin that is used to bound the distance between the items of pairs from different classes. A convolutional neural network models the nonlinear function f_θ by adding a fully connected dense layer onto the pre-trained VGG16 network [95]. The VGG16 network consists of 13 convolutional layers and 2 fully connected layers and was pre-trained on the ImageNet [96] database of images. We use the cosine distance metric as our distance function,[1] and to train the model we use the Gravity Spy training set described in [87]. Each glitch is portrayed as four spectrograms with different temporal durations. These are generated using gwpy [97]. We take these four images to create a single 4-panel image for each glitch, identical to the input currently used for the convolutional neural network classifications in Gravity Spy [87]. By propagating through the DIRECT network described above, the pixel data of the input image is mapped to a smaller, 200 dimensional feature space. The dimension of the feature space is fixed at 200 based on Fig. 2 of [27]. In Figure 3.1 (cf. Fig. 3 of [87]) we show a visual representation of the training set in the DIRECT feature space using the t-distributed stochastic neighbor embedding statistic (t-SNE) [98]. t-SNE is a technique for dimensionality reduction that is well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. As can be seen, samples from the same glitch class are put closer to each other while samples of different classes are far from each other. Such a property is called *discriminative* feature representation. Having trained this model on the known set of glitches, we can now apply it to the unlabeled glitches so that in this new, more discriminative feature space we can cluster similar images together and find new classes.

Having established a means of clustering glitches, in the next section, we describe how we use the discriminative feature space obtained using DIRECT on the Gravity Spy data set to empower volunteers to build large data sets of unknown glitches.

3.2 Identifying Novel Glitches

In our previous work, we relied on the None-of-the-Above classification to identify glitches from previously unknown classes, and the volunteer *Talk* forum (a thread of comments on the image from other volunteers) to consolidate examples in order to develop training sets and add new classes to the supervised model. As the volume of data increases, this design will prove ineffective as a user would have to go through too many classifications before seeing multiple examples of a novel glitch. The Zooniverse system has no method for delivering specific glitch images to a given

[1] The cosine distance between two vectors $\vec{a}$ and $\vec{b}$ is $1 - \vec{a} \cdot \vec{b}/(|\vec{a}|\,|\vec{b}|)$.

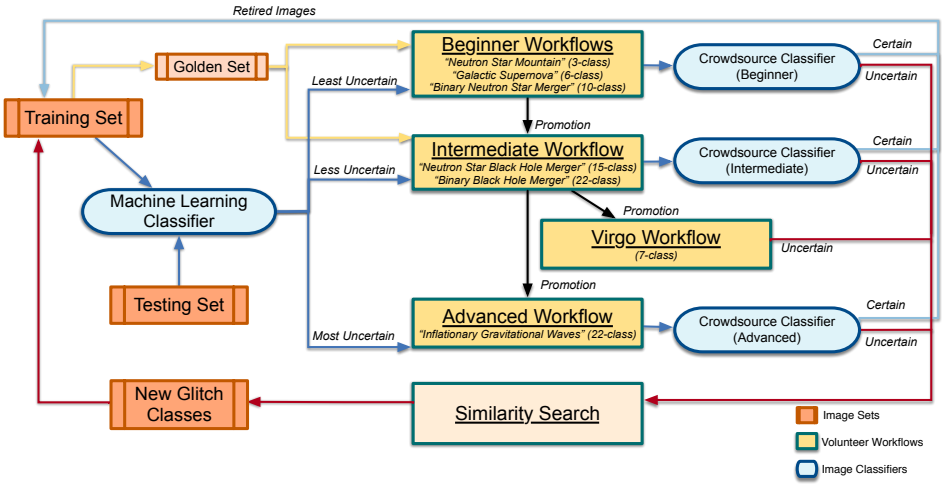


Figure 3.2: New infrastructure proposal for Gravity Spy. This design differs from that described in [15] by facilitating the direct follow-up of single examples of unknown transients through the similarity search algorithm. This is in contrast to the reliance on the None-of-the-Above classifications for filtering out novel glitches from the data set.

user and it can only randomly select a single glitch from the entire pool of glitches. Therefore, if the novel glitches comprise only 1 percent of the overall data set, it is unlikely a single user doing hundreds of classifications will see the novel glitch more than once.

To this end, we are introducing a similarity search tool to the Gravity Spy infrastructure. This tool uses the feature space output for every Gravity Spy glitch, which is obtained using the DIRECT algorithm described in Section 3.1, to enable users to take a single example of a glitch and perform a search querying for glitches in the data which are morphologically similar to the example. In order to determine similarity, we subtract the same cosine distance metric described in Section 3.1 by one. In this case, identical images have a distance metric of 1, while orthogonal images, that is the two images share no common components in DIRECT feature space, images have a distance metric close to 0. This metric has the advantage of being efficient to evaluate and not being affected by the overall scaling of the input vectors. Specifically, this tool is introduced in the form a supplementary web service gravityspytools. The new infrastructure design is highlighted in Figure 3.2.

We anticipate that this tool, which outputs results in the form of *Collections* (galleries of user-selected Gravity Spy glitches), will reduce the size of the data set from which to find examples of a new class, such that building substantial training sets for unknown glitch types is manageable by a single user. Imagine, again, that the novel glitch is only found in 1 percent of the data pool. Instead of relying on the Zooniverse system to randomly select one of the novel glitches, the user will now be able to target that 1 percent of the data set through the similarity tool

and, therefore, not have to go through the other 99 percent of the data image by image. In the next section, we demonstrate how this tool could have proved useful in the rapid identification of two glitches classes that appeared in O2 which were *not* previously included in the Gravity Spy classification.

3.3 Results

We now highlight the application of the similarity search tool on data from O2. Specifically, we assess the impact the similarity search tool could have had on the identification of two glitch classes that appeared during O2: the Water Jet [99] and the Raven Peck [100] glitches. The Water Jet glitch was caused by local seismic noise which resulted in loud bangs near the input optics, and the Raven Peck glitch was caused by ravens pecking on ice built up along vent lines transporting nitrogen outside of the detector [101]. The resulting time–frequency morphologies of these glitches as they appear in the GW data channel can be seen in the top panel of Figure 3.3. These glitches occurred in the LIGO-Hanford detector. The Raven Peck glitch is found in Gravity Spy data between 14 April 2017 and 9 August 2017, and the Water Jet glitch in data between 4 January 2017 and 28 May 2017. Over these durations, there are a total of 13513 and 26871 instances of all types of Gravity Spy glitches, respectively.

We use these two glitches because they highlight the strengths and weaknesses of the DIRECT algorithm, and they allow us to emphasize the importance of incorporating crowd-sourcing methods into the identification of new glitches classes. As DIRECT is a transfer learning algorithm, it is only able to employ concepts of similarity and difference learned from the training set to the unlabeled data. If the distinguishing characteristic of a particular glitch is not also something present or extractable from the training set, it may be more difficult for the algorithm to find other examples easily. This is demonstrated well here because the Raven Peck glitch is most prominently defined as a line feature which *is* present in many of the Gravity Spy classes used in training (such as Power Line, Low Frequency Line and 1080 Line); on the other hand, the unique aspect of the Water Jet glitch is the subtle frequency decay that occurs after the initial pulse, which *is not* obviously extractable from glitches in the training set.

We demonstrate the ability of the similarity search tool to organize testing images by their similarity to a queried glitch. As was done in the DIRECT paper [27], we also compare finding similar images with DIRECT to more straightforward approaches such as using the raw pixel data or doing a Principle Components Analysis (PCA). The raw pixel method does a pixel-by-pixel comparison (using a distance metric like Euclidean) of the energy in each pixel of both images. PCA is a technique used for the identification of a smaller number of uncorrelated variables known as principal components from a larger set of (possibly) correlated data. Therefore,

Act. Layer	Train Rnds	10,000 pr.	50,000 pr.	100,000 pr.
tanh	30	(0.93, 0.78)	(0.96, 0.92)	(0.94, 0.87)
	100	(0.90, 0.85)	(0.93, 0.82)	(0.94, 0.72)
	200	(0.93, 0.82)	(0.87, 0.80)	(0.93, 0.93)
leakyReLU	30	(0.96, 0.94)	(0.95, 0.92)	(0.95, 0.91)
	100	(0.95, 0.91)	(0.96, 0.91)	(0.96, 0.94)
	200	(0.95, 0.91)	(0.97, 0.92)	(0.96, 0.92)

Table 3.1: The fraction of the original data set with similar scores lower than the similarity score of 50.0% of other known Raven Peck (left) and Water Jet glitches (right). Columns refer to different choices in the activation layer used in the dense layer of the model and the number of training rounds where each round draws a new set of X number of similar and dissimilar pairs. In bold is the configuration(s) that yielded the best reduction versus retention rate for both glitches.

the PCA method first performs this dimensionality reduction on all the images (as DIRECT does) and then uses a distance metric to compare how close in this space an image is to all other images in the dataset. The bottom panel of Figure 3.3 shows the fraction of known samples that have a higher similarity score than a given percentage of the other data set samples. For example, while retaining 50.0% of known Raven Peck glitches, we can remove about 99.9% percent of the other data set samples, increasing the purity of the set to be examined by the user. For the same glitch, the raw pixel data approach and the PCA approach perform similarly with the Raw Pixels approach doing best at near 100.0%. For Water Jet glitches, DIRECT also gives a similar performance retaining 50.0% of known samples as it did for the Raven Peck. However, if a retention rate of 100.0% of the known samples is desired, the data set reduction rate for Raven Peck is 92.0% compared to 55.0% for the Water Jet glitch. For this glitch, the methods of raw pixel data and PCA prove ineffective. For a retention rate of 50.0% only about 30.0% percent of sample in the data set have lower similarity scores. We believe these examples represent both a challenging and less challenging task for the model, and in both cases DIRECT performs well, and the other approaches fail to be effective in the case of the Water Jet glitch. We anticipate the reduction in the size of the original data set combined with the retention rate of similar samples to be significant enough that a single user can produce large data sets of novel glitches.

3.3.1 Different Configurations

To test the best training and setting configuration for DIRECT, we tried two different activation layers, tanh and leakyReLU, for the custom fully connected layer that DIRECT adds to the VGG16 model. In addition, we varied the number of training rounds and the number of pairs of similar and dissimilar images that are drawn from the training set each time. As training this model can be expensive

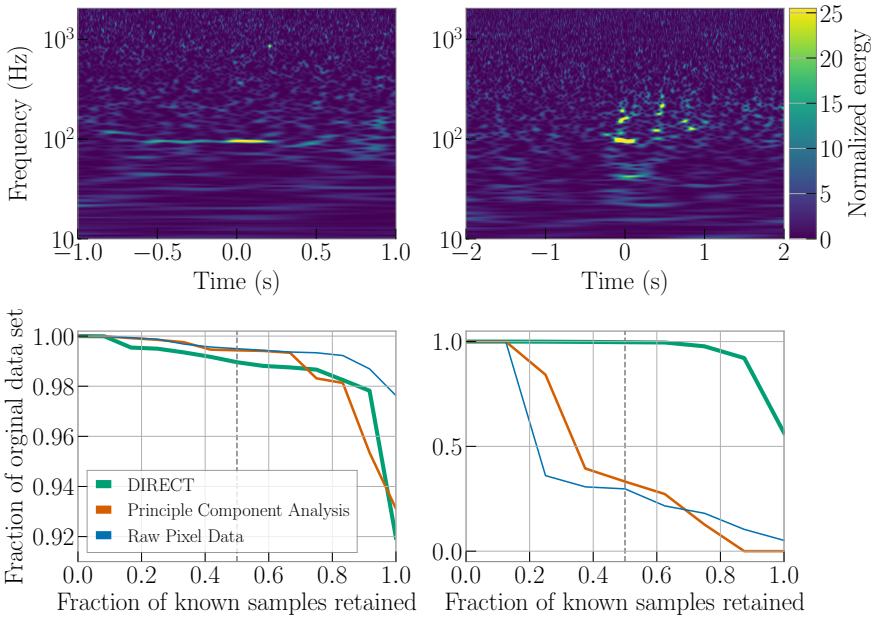


Figure 3.3: *Top*: Nominal examples of the Raven Peck (*left*) and Water Jet (*right*) glitches. *Bottom left*: The fraction of known Raven Peck samples that have a higher similarity score than a given percentage of other data set samples when calculating similarity to a single known Raven Peck glitch. For example, while retaining 50.0% of known Raven Peck glitches, we can disregard about 99.0% percent of the other data set samples, increasing the purity of the set to be examined by the user. *Bottom right*: Same for Water Jet glitches. Similarly, while retaining 50.0% of known Water Jet glitches, we can disregard about 99.0% percent of the other data set samples.

because of the possible pairs of images one can produce from the training set, it is critical to understand what the minimal expense is that still produces an effective model. To judge effectiveness, we quote the percentage reduction in the data set samples at which we still retain 50.0% of other known Raven Peck and Water Jet glitches, respectively. This value for each model is shown in Table 3.1. The models using tanh as the activation layer perform worse than that with leakyReLU. We anticipate this is due to the restricted range allowed by the tanh activation layer, $[-1, 1]$, compared to that of leakyReLU, $(-\infty, \infty)$. Specifically, the distances away from each other that similar and dissimilar images can be is restricted in the one case causing the discriminative feature space that is created to be less discerning than the other. In terms of number of training pairs and rounds of training, it appears that increasing each does not lead to significantly improved results. We anticipate this is due to the fact that most of the Gravity Spy classes are quite distinct from each other, and therefore using or drawing more pairs of images is unnecessary to produce a useful discriminate feature space representation of the data. Therefore, this method can still be effective without an extremely costly training stage.

3.4 Conclusions

We have described a novel extension of current GW data transient class identification combining the power of citizen scientists with the latest techniques in ML. In the original paper, we allowed volunteers to classify glitches as None of the Above in order to identify individual subjects which belong to an unknown class. Utilizing DIRECT, we have shown the ability to expedite the identification of new glitch classes compared with this original method, which will be important as we get new data from upcoming observing runs.

Using two noise transients from LIGO's O2 data, the Raven Peck and water Jet glitches, we demonstrated that DIRECT creates a discriminative feature space representation of the Gravity spy data set such that single examples of each glitch can efficiently lead to the discovery in the data set of other Raven Peck and Water Jet glitches. We compared DIRECT to simpler approaches such as using the raw pixel data or PCAs to find similar images and found that DIRECT produces either comparable or better results depending on the glitch.

There are a variety of plans for future related research. For example, we can explore the use of other metrics to further the inter- and intra-class separation, thereby identifying separate classes that are otherwise improperly associated. In addition, there are lessons learned that are applicable in other areas of astronomy, in line with the on-going applications of unsupervised learning to large-scale astronomical surveys [e.g., 102–105]. For example, the Large Synoptic Survey Telescope (LSST) [106], an 8 meter class telescope being constructed on Cerro Pachon near La Serena, Chile, will take millions of images over its lifetime, identifying approximately 100,000 objects each night. Although it is impossible for most of them to be analyzed by astronomers directly, citizen scientists can contribute to the monitoring, classifying, and annotating of spurious and surprising data. The expectation is that LSST, with its unprecedented field-of-view and rapid cadence, will discover a multitude of astrophysical phenomena, and can benefit from the ability to rapidly identify unique signals. Fast transients which fade over a few days timescales, such as kilonova, which have not been identified in previous surveys are likely to be found and will constitute a new class of transient for this survey.

In general, the possibilities for further science education with citizen science initiatives, which place students on the edge of the scientific frontier, lie strongly in the identification of previously unknown phenomena. Projects such as these create an environment where not all phenomena are known and understood, in contrast to textbook science lessons, and achieve a more realistic view of the wonder and challenges of science [107–109]. For this reason, these initiatives help provide the foundation for further education in any scientific field, where the goal is to be able to follow a logical account of a problem to a solution, through the creation of a hypothesis, the taking of data, and the eventual explanation to understand the phenomena.

Projects like this will have significant educational benefits and will impact the research projects both inside of LIGO and LSST and outside in numerous research groups conducting other astrophysical studies. We anticipate that the combination of more novel ML techniques with web applications will continue to help with the efficacy of this work.

Chapter 4

X-Pypeline

Although classifying existing and identifying new classes of excess noise has proven useful for characterizing detector data, it is also critical to apply lessons learned from these efforts to gravitational-wave data searches. To this end, this Chapter discusses the current method for performing a search for a GW signal from a Galactic supernova using a new Python-based implementation of the software package known as "X-Pipeline" [23–25]. Moreover, we describe a method for using both machine learning techniques, and Gravity Spy classification results to help tune the results of the algorithm. This open-source algorithm, called "X-Pypeline"[1], will offer not only a faster execution of the current methodology, but set up future efforts to integrate open source machine learning techniques.

In general, a GW search algorithm relies on the use of signal models, also known as templates, and detection statistics resulting from these models to claim a detection of a GW. As mentioned in the introduction, a number of GW signals, such as a GW from a CCSN, cannot be meticulously modelled, or have too broad a range of possible emissions to cover with modelled templates. In this case, we must think of a GW signal more generically, and create an agnostic template that can span the entire space of signal models. These searches fall under the umbrella of so-called unmodelled GW search algorithms and generally employ a coherent analysis of detector data. In a coherent analysis, data from multiple detectors is combined in both phase and amplitude before it is analyzed for a GW. In Section 4.1, we describe the forms GW searches for unmodelled signals take, and motivate how coherently analyzing GW data can help detect GW signals. In section 4.2, we give an overview of the X-Pypeline algorithm, describing how the data is characterized such that a signal template in the form of a time-frequency pixel can be made and a detection statistic constructed from this template. These detection statistics rely on an assumption that the underlying noise of the data is Gaussian which we know from Chapter 2 is untrue. Therefore, we discuss how a coherent analysis of the data can guide the creation of coherent consistency statistics which can be

[1] https://github.com/X-Pypeline/X-Pypeline

used to suppress loud excess noise transients such as those described in Chapter 2. In section 4.3, we discuss different ways to tune the results of the algorithm such that the "loud" non-Gaussian background is suppressed in order to better claim a detection of a GW. The tuning methods described in this section all rely on the creation of training and testing sets. For this reason, in section 4.4, we describe how Gravity Spy classifications of glitches can improve the selection of the training and testing sets that are used to tune the results of the analysis.

4.1 Unmodelled Gravitational Wave Searches

Gravitational Wave Bursts (GWBs) have been a long sought class of gravitational wave signals. In general, two forms of burst searches exist, all-sky, all-time searches and triggered searches. The former search scans all available data and sky directions for a GWB. The latter search relies on a counterpart event such as the detection of neutrinos and optical emissions from core-collapse supernovae (CCSN) or gamma-ray-bursts (GRB) to identify either the time and/or sky location of the astrophysical source of the GWB [25, 110]. For these astrophysical events, the "X-Pypeline" algorithm uses the sky location of the source to tune the analysis. The detectors will have different sensitivities to a plus and cross polarized GWB from different sky locations, and these responses can be mathematically modelled and various likelihood statistics calculated to extract time-frequency pixels from the data that are more likely to contain GWBs [26]. In this thesis, we will focus on the implementation of "X-Pypeline" in response to an optical and/or neutrino counterpart because we plan on using it to search for a Galactic supernova.

Along with other burst search algorithms, "X-Pypeline" employs the technique of coherent analysis [16, 17, 23, 111, 112]. A coherent analysis uses both individual and combined detector data streams in order to better reject background noise events and thereby increase sensitivity to GWBs. The motivation behind this is that GWBs will appear in some quantifiable capacity in all the data streams, but "glitches", such as those described in Chapter 2, would typically occur in only one of the data streams at a given time. More specifically, glitches would appear in the autocorrelation terms of any statistical composition of the data streams and gravitational waves in the cross-correlation terms. To this end, in the following section, we discuss how "X-Pypeline" characterizes GW data in order to motivate the use of time-frequency pixels as templates for GWBs and how these templates can be used to construct detection statistics that find GWs in the data and coherent consistency statistics that reject glitches.

4.2 Overview of X-Pypeline Data Analysis

In this section, the standard method and notation of an X-Pypeline analysis is presented, for a more complete discussion, see [23]. X-Pypeline focuses on techniques useful for detecting signals on the order of a couple of seconds, particularly when the exact waveform is unknown. As the exact form of the supernova waveform (discussed more in Chapter 5) is uncertain (due to, among other things, uncertainty in the physics, lack of 3-D numerical relativity simulations, and the stochasticity of the explosion), and the duration will be at most 1-2 seconds, this is an ideal astrophysical event for this type of algorithm. Specifically, in Section 4.2.1, we motivate the selection of a template by characterizing GW data as the linear combination of signal and noise. In Section 4.2.2, we motivate the use of the short time Fourier transform to project the GW detector data into time-frequency pixels. In Section 4.2.3, we describe how projection vectors can be applied to these time-frequency pixels in order to create GWB templates. These templates can then be used when creating detection statistics. Specifically, we formulate two such statistics, a match-filter motivated statistic, the standard likelihood, which is described in Section 4.2.4 and a Bayesian motivated statistic which is described in Section 4.2.5. Finally, in Section 4.2.6, we show how these projection vectors can also motivate the creation of coherent consistency statistics designed to reject glitches.

4.2.1 Characterizing the Data

A GWB search has three necessary components, the sky location Ω, the detector data d for a given detector α and the characterization of the GW as its plus and cross polarization components, $(h_+(t), h_x(t))$. The detector data d can be further subdivided into two component parts, the data that is background noise n and the data that is signal, which is given by the detector response to each polarization for a given sky location, $F_\alpha(\Omega)^+$ and $F_\alpha(\Omega)^\times$, multiplied by the signal. The sky location plays two roles in this analysis. First, given a sky location, the exact time-of-arrival delay, written as $\Delta t_\alpha(\Omega)$, of the GW signal between the various detectors is known. Second, the sensitivity of the detectors to the signal at that sky location is also known. Therefore, the data from a detector $\alpha \in [1 \dots D]$ where D is the number of detectors can be written as a linear combination of the GWB and the noise,

$$d_\alpha(t + \Delta t_\alpha(\Omega)) = F_\alpha(\Omega)^+ h_+(t) + F_\alpha(\Omega)^\times h_\times(t) + n_\alpha(t + \Delta t_\alpha(\Omega)). \tag{4.1}$$

After applying a time shift to each individual detector corresponding to $\Delta t_\alpha(\Omega) = \frac{1}{c}(r_0 - r_a)\cdot\Omega$, where r_0 is some reference position, say the position of detector $\alpha = 1$, and r_a is the position of detector α, d_α contains the simultaneous contributions of amplitudes due to the GWB. In addition, we will consider the sky location as being

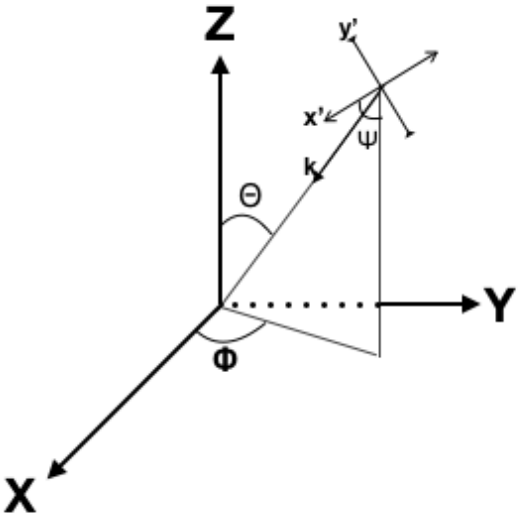


Figure 4.1: Coordinate frame in which the antenna pattern functions are described. An example sky location, given by (ϕ, θ) and reference polarization angle ψ as measured by local south are shown above. The (X, Y, Z) coordinates are such that X and Y are along the arms of the interferometer. (x', y', k) coordinates define the propagation and polarization of the gravitational wave.

angles (ϕ, θ) defined in a frame based at the center of the Earth with the x-axis pointing at the Greenwich meridian and the z-axis pointing at the north pole and having Euler angles (ϕ, θ, ψ). This coordinate frame is called *Earth fixed*, and is the frame in which all X-Pypeline data is analyzed. As such, data d from a series of detectors D can now be written as,

$$\begin{bmatrix} d_1(t) \\ \vdots \\ d_D(t) \end{bmatrix} = \begin{bmatrix} F_1(\phi,\theta,\psi)^+ h_+(t) + F_1(\phi,\theta,\psi)^\times h_\times(t) + n_1(t) \\ \vdots \\ F_D(\phi,\theta,\psi)^+ h_+(t) + F_D(\phi,\theta,\psi)^\times h_\times(t) + n_D(t) \end{bmatrix}. \tag{4.2}$$

In order to obtain the detector sensitivity in these Earth fixed coordinates, one must rotate the Earth fixed angles into *detector frame* angles. In order to accomplish this rotation, one must use the orientation of the interferometer, see the appendix of [112] for more information. This detector frame coordinate system for detector i is where the x-axis is along one of the arms of the interferometer and the y-axis is along the other and has angles $(\phi_i, \theta_i, \psi_i)$ where ψ_i is the polarization angle. In this frame, we can write the sensitivity of detector i to the plus and cross polarization of a GW coming from direction (ϕ_i, θ_i) as,

$$F^+(\phi_i, \theta_i, \psi_i) = \frac{1}{2}(1 + \cos^2\theta_i)\cos 2\phi_i \cos 2\psi_i - \cos\theta_i \sin 2\phi_i \sin 2\psi_i \tag{4.3}$$

and

$$F^{\times}(\phi_i, \theta_i, \psi_i) = \frac{1}{2}(1 + \cos^2\theta_i)\cos 2\phi_i \sin 2\psi_i + \cos\theta_i \sin 2\phi_i \cos 2\psi_i. \tag{4.4}$$

.

These functions are referred to as the *antenna response functions* of the interferometer. Figure 4.1 gives a graphical representation of the detector frame coordinate system. As this thesis focuses on an optically triggered search, expected to have a single sky location to analyze, we will drop the notation including the sky location, for the remainder of this discussion. Now that we have characterize the data streams in such a way that we have the contribution of every data stream to the overall GWB amplitude, we must project the time series data in such a way that we can combine, or add, these data contributions together. Moreover, we must select a projection of the data that is in line with our expectation of the GWB signal morphology. We discuss how we project the detector data in the following section.

4.2.2 Time-Frequency Representation of the data

As discussed at the beginning of this section, the exact supernova waveform, i.e. the shape of the signal, is unknown. However, many simulated GWBs show power that is distributed over a compact time-frequency region. Therefore, the ideal rendering of the data should be to project the data onto a set of basis functions that are limited in both time and frequency. X-Pypeline does this by performing a short-time Fourier transform (STFT) on data d. STFTs are useful for determining the changing frequency content of a timeseries as a function of time. Since a STFT is essentially a continuous Fourier Transform (FT), limited inside some window δt, applied to the timeseries, it is useful to first introduce the notation of the FT of timeseries $d(t)$ as,

$$D(\omega) = \int_{-\infty}^{\infty} d(t) e^{-i\omega t} dt, \tag{4.5}$$

where ω is the angular frequency.

Although the data has been described as continuous so far, in reality, GW detector data is sampled discretely. Thus it is also useful to note that the discrete Fourier Transform (DFT), $d[k]$, of the discrete time series $d[n]$, can be written as

$$D[k] = \sum_{n=0}^{N-1} d[n] e^{\frac{-i2\pi}{N} kn} \tag{4.6}$$

where N is the number of data points in the time series. We also note here the inverse discrete Fourier Transform which can be written as

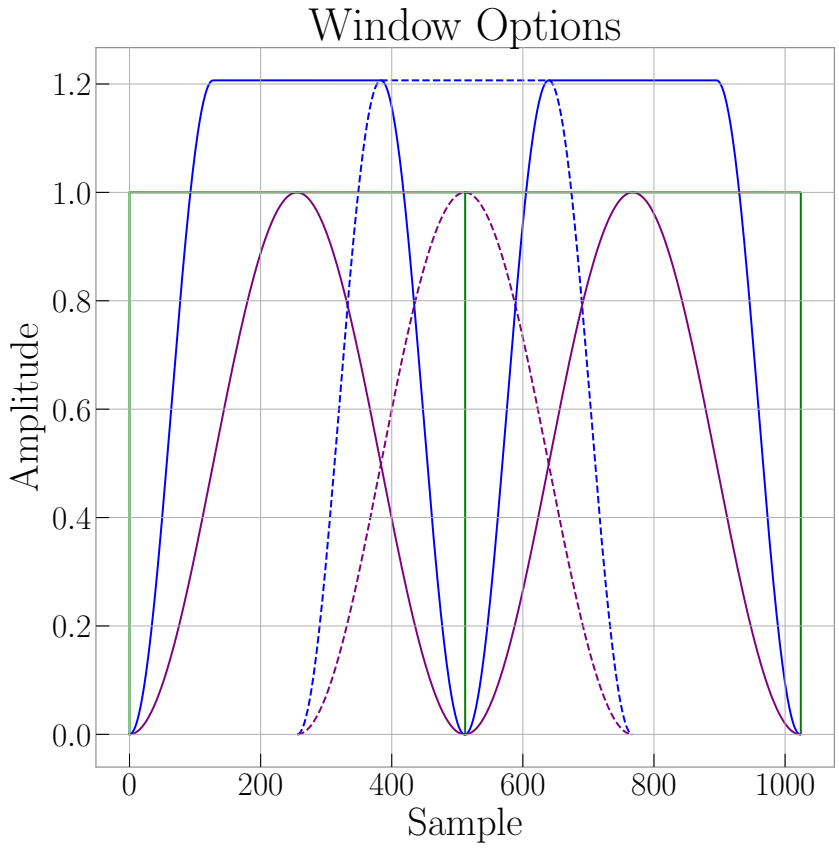


Figure 4.2: Three examples of discrete window functions over a span of two segments of 512 samples. In blue, three fifty percent overlapping Tukey windows. In green, two boxcar windows with no overlap. In purple, three Hann windows with fifty percent overlap. Each Tukey and boxcar window have an area of 1, and each Hann an area of 0.5. We use these examples to illustrate the many different ways the same number of samples could be windowed.

$$d[n] = \frac{1}{N}\sum_{k=0}^{N-1} D[k]e^{\frac{i2\pi}{N}kn}. \tag{4.7}$$

With this notation, it is easy to write both the continuous and discrete STFT of time series $d(t)$ and $d[n]$. The continuous STFT of timeseries $d(t)$ can be written as,

$$D(\tau,\omega) = \int_{-\infty}^{\infty} d(t)w(t-\tau)e^{-i\omega t}dt \tag{4.8}$$

where $w(\tau)$ is a window function satisfying,

$$\int_{-\infty}^{\infty} w(\tau)d\tau = 1, \tag{4.9}$$

examples of which can be seen in Figure 4.2.

The discrete STFT can be written as

$$D[m,k] = \sum_{n=0}^{N-1} d[m]w[n-m]e^{\frac{-i2\pi}{N}kn} \tag{4.10}$$

where $w[m]$ is a window function satisfying,

$$\sum_{m=0}^{M} w[m] = 1, m \in [1,M], \tag{4.11}$$

where M is the size of the window.

The discrete STFT of a time series amounts to a DFT being performed on m chunks of (usually overlapping) samples in d which are multiplied by some window, $w[m]$. We note here that if we compare the factor in the exponential from the inverse DFT, Equation 4.7, to $sin(2\pi ft)$ where $t = \frac{n}{fs}$ we find that $f = \frac{k}{Nfs} = \frac{k}{T}$ and so this necessitates a spectrogram with frequency spacing of $\delta f = \frac{1}{t}$. In this way, time-frequency *pixels* with the properties $\delta t \delta f = 1$ are made for each chunk of samples. Each of these pixels contains information about the phase and magnitude of the signal over that time and frequency span. Because the pixels must satisfy $\delta t \delta f = 1$, there is a time-frequency resolution trade off when using a STFT. Figure 4.3 provides a visualization of this trade off. The best time-frequency resolution for a GWB is not known before hand, and, so, X-Pypeline uses time resolutions from $\frac{1}{2}$ s to $\frac{1}{128}$ s in steps of powers of two. The default window used in "X-Pypeline" is a Tukey window [113] which is a cosine-tapered window that is constructed by taking a cosine lobe of width $\alpha(N+1)/2$ and convolving it with a rectangular window of width $(1-\frac{\alpha}{2})(N+1)$ where α is the so-called shape parameter. In essence, you are splicing a cosine taper to the end of a rectangular window which can be seen by the blue curve in 4.2. By default, we use a value of $\alpha = 0.25$ in "X-Pypeline". Finally,

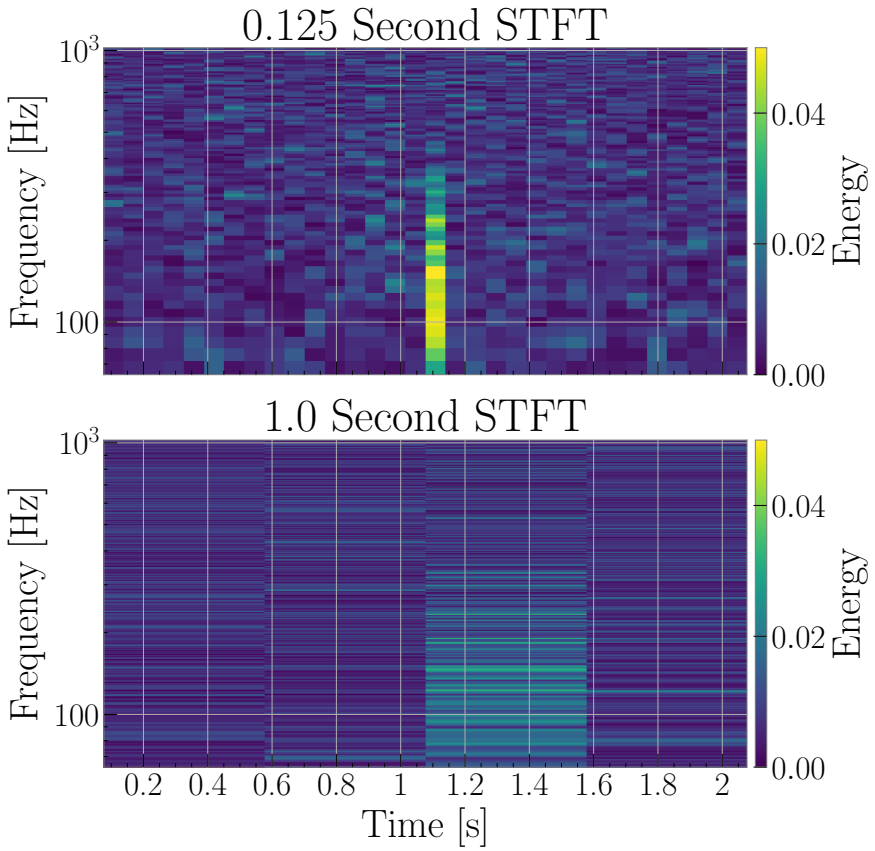


Figure 4.3: A blip glitch visualized using a STFT with two different resolutions, (top) 0.125 seconds and (bottom) 1.0 seconds. The top represents pixels which have a better time resolution and the bottom represents pixels that have better frequency resolution. For this event, it is easier to resolve the event with the shorter duration, better time resolution STFT. Note that in each case the pixels have the same area $\delta t \delta f = 1$.

we use 50 percent overlapping segments

We have now projected the detector data streams onto time-frequency pixels. Based on the characterization of the data from section 4.2.1, these pixels contain energy contributions from all of the detectors, which includes not only signal but also noise. If we can project these time-frequency pixels so that we can isolate the energy contribution that is due almost solely to the signal, then we will have a basis for a signal model and, consequently, a way to build a detection statistic. In the following section, we discuss how we can project these time-frequency pixels onto a basis that is spanned by the antenna response functions in order to isolate the contribution of the energy in the pixel that is mostly due to the signal. It is with these projections that "X-Pypeline" creates its templates.

4.2.3 Characterizing a single time-frequency pixel

"X-Pypeline" coherent consistency and detection statistics can be most easily understood with respect to a single time-frequency pixel. Therefore, we restrict our discussion of how we project time-frequency pixels in order to isolate the contribution of the energy in the pixel that is due to a signal. Moreover, it is convenient at this point to whiten, or normalize, each detector data stream by the one-sided power spectral density $S_\alpha(f)$ of the noise n_α for detector α. The whitened data is denoted with a w. In this way, Equation 4.2 for a single time-frequency pixel (t, f) becomes

$$\begin{bmatrix} d_1^w(t,f) \\ \vdots \\ d_D^w(t,f) \end{bmatrix} = \begin{bmatrix} F_1(\phi,\theta,\psi)^{+w}(f) \\ \vdots \\ F_D(\phi,\theta,\psi)^{+w}(f) \end{bmatrix} h_+(t,f) + \begin{bmatrix} F_1(\phi,\theta,\psi)^{\times w}(f) \\ \vdots \\ F_D(\phi,\theta,\psi)^{\times w}(f) \end{bmatrix} h_\times(t,f) + \begin{bmatrix} n_1^w(t,f) \\ \vdots \\ n_D^w(t,f) \end{bmatrix} \tag{4.12}$$

.

Note that in the signal term we combine the whitening factor with the antenna responses so the new vectors $F_+, F_\times$ contain all information about the sensitivity of the network for the sky position in question. Equation (4.12) presents the data of a single time-frequency pixel as a vector $\vec{d}$ in the basis that is formed by the set of single detector strains. For convenience we will drop the explicit notation including (t, f), the whitening, and the sky position, which as noted earlier will be known ahead of time. Concisely, the amplitude and phase of a single-time frequency pixel has the form

$$\mathbf{d} = \mathbf{F}^+ h_+ + \mathbf{F}^\times h_\times + \mathbf{n}. \tag{4.13}$$

This single pixel is essentially a vector $\mathbf{d}$ in a basis that is spanned by the D detector data streams. The basis created by D detectors is not the only way to characterize pixel data $\mathbf{d}$, however, and, in fact, is sub-optimal when developing detection statistics and coherent consistency checks. By whitening the data, we have made the covariance matrix between the superfluous detector noise $\mathbf{n}$ and the detectors the identity matrix, thus making the covariance matrix unchangeable with a change in basis. Therefore, it is useful to find a basis that is created by a plane that is spanned by the antenna response vectors, $\mathbf{F}^+$ and $\mathbf{F}^\times$, which in this case will simply be a two dimensional plane. Moreover, by examining (4.3) and (4.4), it is clear that under a choice of polarization angle ψ, one can find a direction in which the detector network sensitivity is maximized and an orthogonal direction in which it is minimized. Specifically, one can explicitly solve for an angle ψ such that $\mathbf{F}^+$ is along the maximum response and $\mathbf{F}^\times$ is along the minimum. This basis is referred to as the *Dominant Polarization Frame* (DPF) [17]. In this way, we are able to

project the vector $\mathbf{d}$ onto a plane containing the maximum part of the data vector that can be explained by a plus and cross polarized GWB. Moreover, the $D-2$ streams of the data that are orthogonal to this plane is the part of $\mathbf{d}$ that cannot be explained by a plus and cross polarized GWB, and therefore should be consistent with Gaussian noise.

One solves for the DPF by first looking at the sensitivity of the detector network ($\mathbf{F}^+$, $\mathbf{F}^\times$) to all linearly polarized signals (h_+, h_x). h_+ and h_x are proportional to $\cos(2\psi)$ and $\sin(2\psi)$, respectively, and so we can write the sensitivity of the detector network as

$$\begin{aligned}\mathbf{F}^+(\psi) &= \cos(2\psi)\mathbf{F}^+ + \sin(2\psi)\mathbf{F}^\times, \\ \mathbf{F}^\times(\psi) &= -\sin(2\psi)\mathbf{F}^+ + \cos(2\psi)\mathbf{F}^\times.\end{aligned} \tag{4.14}$$

To find the choice of ψ such that the network response to the plus polarization is maximized, we maximize the magnitude of the plus network response, $|\mathbf{F}^+(\psi)|^2$. The magnitude is maximized when

$$\frac{\delta|\mathbf{F}^+|^2}{\delta\psi} = 0. \tag{4.15}$$

The magnitude of Equation 4.14 is

$$\begin{aligned}|\mathbf{F}^+(\psi)|^2 &= |\mathbf{F}^+\cos(2\psi) + \mathbf{F}^\times\sin(2\psi)|^2 \\ &= |\mathbf{F}^+|^2\cos^2(2\psi) + 2\mathbf{F}^+\cdot\mathbf{F}^\times\sin(2\psi)\cos(2\psi) + |\mathbf{F}^\times|^2\sin^2(2\psi).\end{aligned} \tag{4.16}$$

If we perform the following substitutions,

$$\begin{aligned}\cos^2(2\psi) &= \frac{1}{2}\left(\cos\left(4\psi\right)+1\right), \\ \sin(2\psi)\cos(2\psi) &= \frac{1}{2}\sin(4\psi), \\ \sin^2(2\psi) &= \frac{1}{2}\left(1-\cos\left(4\psi\right)\right),\end{aligned} \tag{4.17}$$

then the magnitude of the plus network response can be written as

$$|\mathbf{F}^+(\psi)|^2 = \frac{1}{2}|\mathbf{F}^+|^2\left(\cos\left(4\psi\right)+1\right) + \mathbf{F}^+\cdot\mathbf{F}^\times\sin(4\psi) + \frac{1}{2}|\mathbf{F}^\times|^2\left(1-\cos\left(4\psi\right)\right). \tag{4.18}$$

Finally, we maximize the plus network response

$$0 = \frac{\delta|\mathbf{F}^+|^2}{\delta\psi} = -2(|\mathbf{F}^+|^2 - |\mathbf{F}^\times|^2)\sin(4\psi) + 4\mathbf{F}^+\cdot\mathbf{F}^\times\cos(4\psi), \tag{4.19}$$

and solving for ψ, we obtain

$$\psi_{\mathrm{DPF}} = \frac{1}{4}\arctan\left(\frac{2\mathbf{F}^+\cdot\mathbf{F}^\times}{|\mathbf{F}^+|^2 - |\mathbf{F}^\times|^2}\right). \tag{4.20}$$

Plugging this result back into Equation 4.14, we obtain vectors $\mathbf{f}^+$ and $\mathbf{f}^\times$,

$$\begin{aligned}\mathbf{f}^+(\psi_{\mathrm{DPF}}) &= \cos(2\psi_{\mathrm{DPF}})\mathbf{F}^+ + \sin(2\psi_{\mathrm{DPF}})\mathbf{F}^\times, \\ \mathbf{f}^\times(\psi_{\mathrm{DPF}}) &= -\sin(2\psi_{\mathrm{DPF}})\mathbf{F}^+ + \cos(2\psi_{\mathrm{DPF}})\mathbf{F}^\times,\end{aligned} \tag{4.21}$$

which are orthogonal. The magnitude of $\mathbf{f}^+$ is either at a maximum or a minimum and if it is at a minimum we rotate by an addition 45 degrees to ensure that the vectors have the property $|\mathbf{f}^+| \geq |\mathbf{f}^\times|$. We normalize these vectors producing

$$\mathbf{e}^+ = \frac{\mathbf{f}^+}{|\mathbf{f}^+|}, \mathbf{e}^\times = \frac{\mathbf{f}^\times}{|\mathbf{f}^\times|}. \tag{4.22}$$

At this point, the value $\mathbf{e}^+$ and $\mathbf{e}^\times$ represent our template for what a GWB signal looks like in the data. More specifically, we have siphoned off the signal into 2 data streams of signal plus Gaussian noise from D data streams of signal plus noise. Therefore, we have only Gaussian noise in the remaining $D-2$ data streams, and these data streams can then be used as tests for Gaussianity that can help to reject glitches. This is an example of a coherent consistency check which will be discussed in more details in Section 4.2.6. This template, based on signal amplitude, will be what is used to do a match-filter like search for a GWB in the data. The benefit of characterizing and transforming the data in this manner is twofold. First, there are two unknown values in each pixel, $(h_+(t,f), h_\times(t,f))$, and with three or more data streams, i.e. detectors, per pixel then it is possible to solve for these quantities. With only two detectors and the addition of a time constraint on the coincidence of candidate events, then it is also possible to determine $(h_+(t,f), h_\times(t,f))$. We add the caveat for solving for the two unknown quantities with two detectors because with only two detectors any output can be interpreted as $(h_+(t,f), h_\times(t,f))$, and this means noise glitches (as well as actual GWBs) will be assign high likelihood values (likelihood values are discussed in Section 4.2.4). With more than 2 detectors, the system is over constrained and we can enforce consistency tests which we will be discussed in 4.2.6. Second, and in a similar line of reasoning, this basis can be used to construct two types of statistics: detection statistics and coherent consistency statistics. A detection statistic is used to determine the probability that a given data segment contains a gravitational-wave signal and will be discussed in Section 4.2.4 and 4.2.5. Coherent consistency tests, on the other hand, are used to reject spurious noise transients which can be confused as a signal by the detection statistics if they are not ruled out as being from a GWB. These tests will be discussed in Section 4.2.6.

4.2.4 Building a Detection Statistic for Gravitational Wave Bursts

Before we discuss how to leverage the single time-frequency pixel templates $\mathbf{e}^+$ and $\mathbf{e}^\times$ to detect signals in data $\mathbf{d}$, we briefly introduce the concepts of optimal match filtering and Bayesian statistics to motivate choosing such a statistic.

The optimal matched filtering [114] statistic employs the simple concept of a likelihood ratio

$$L = \frac{P\left(d(t)|H_1\right)}{P\left(d(t)|H_0\right)} \tag{4.23}$$

where P is the probability and there is a way to map d to one of the two hypotheses,

- H_0 detector timeseries d contains only noise
- H_1 detector timeseries d contains a plus and cross polarized signal and noise.

Any function that increases when the data is better explained by the signal than by the noise, is called a detection statistic. It is clear that Equation 4.23 fits this criteria. Here we can substitute the characterization of the data from 4.2.1 and obtain the likelihood ratio

$$L = \frac{p(\mathbf{d}|h_+, h_\times, \mathbf{n})}{p(\mathbf{d}|\mathbf{n})}. \tag{4.24}$$

That is, what is the probability of this data realization, i.e. single time-frequency pixel, in the presence of a signal that is fully mapped to h^+, $h^\times$ and noise, versus the probability of this data realization with just noise. For this likelihood, the noise is assumed to be Gaussian. This assumption regarding the noise is why rejecting loud non-Gaussian noise transients is critical. Otherwise, the motivation behind the detection statistic is flawed. With this assumption, the probably of obtaining $\mathbf{d}$ under stationary coloured Gaussian noise, i.e. the denominator of the equation, can be written as,

$$p(\mathbf{d}|\mathbf{n}) \propto \exp\left(-\frac{1}{2}\int_{-\infty}^{\infty} 2|\mathbf{d}|^2 df\right). \tag{4.25}$$

where $S(f)$ is the one-sided power spectral density. We will use continuous notation for the time being, even though $\mathbf{d}$ has been characterized as discerete. In other words, we assume $\mathbf{d}$ was obtained via the *continuous* STFT. We will introduce the discrete characterization of the multivariate Gaussian distribution, shortly. The numerator takes the form

$$p(\mathbf{d}|h_+, h_\times, \mathbf{n}) \propto \exp\left(-\frac{1}{2}\int_{-\infty}^{\infty} 2|(\mathbf{d} - (F^+ h_+ + F^\times h_\times)|^2 df\right). \tag{4.26}$$

Now we can rewrite Equation 4.24 as,

$$L = \frac{p(\mathbf{d}|h_+, h_\times, \mathbf{n})}{p(\mathbf{d}|\mathbf{n})} = \frac{\exp\left(-\frac{1}{2}\int_{-\infty}^{\infty} |(\mathbf{d} - (F^+ h_+ + F^\times h_\times))|^2 df\right)}{\exp\left(-\frac{1}{2}\int_{-\infty}^{\infty} 2|\mathbf{d}|^2 df\right)}. \tag{4.27}$$

We can simplify the above equation by expressing it as a log-likelihood, obtaining

$$\begin{aligned} 2\log(L) &= -||(\mathbf{d} - (F^{+}h_{+} + F^{\times}h_{\times}))||^2 + ||\mathbf{d}||^2 \\ &= 2(\mathbf{d} \cdot (F^{+}h_{+} + F^{\times}h_{\times})) - ||(F^{+}h_{+} + F^{\times}h_{\times})||^2 \end{aligned} \tag{4.28}$$

where $||a||^2 = a \cdot a$ and

$$a \cdot b = \int_{-\infty}^{\infty} a^{*}(f)b(f) + a(f)b^{*}(f)df \tag{4.29}$$

where $*$ is the complex conjugate.

Here, we introduce the concept of the signal to noise ratio (SNR). The SNR is the expected value of the log-likelihood ratio over all noise realizations. The SNR, when the data contains a signal, is as follows

$$\begin{aligned} &E[2\log(L)|\mathbf{d} = (F^{+}h_{+} + F^{\times}h_{\times}) + n] \\ &= E[2\left(\left(F^{+}h_{+} + F^{\times}h_{\times}\right) + n\right) \cdot (F^{+}h_{+} + F^{\times}h_{\times}) - (F^{+}h_{+} + F^{\times}h_{\times})^2] \\ &= 2||\left(F^{+}h_{+} + F^{\times}h_{\times}\right)||^2 - ||\left(F^{+}h_{+} + F^{\times}h_{\times}\right)||^2 \\ &= ||\left(F^{+}h_{+} + F^{\times}h_{\times}\right)||^2. \end{aligned} \tag{4.30}$$

We note that the noise, n, vanishes in the expectation value as the noise is assumed uncorrelated with h. Therefore, the SNR represents the optimal way to formulate a detection statistic when the signal is known a priori. As mentioned earlier, this assumes that the template used is identical to the signal in the data, but we do not have a way to make such a template. Thus, the match-filter log-likelihood is inappropriate for data containing a GWB.

It is critical, therefore, to consider what signal model could be used in the case of a GWB. A GWB signal can, at best, be characterized as a member of a set that span a range of possible amplitudes, time of arrivals, shapes, etc. We will refer to this signal as $s(\theta)$ in the set of possible signal realizations. We call this new hypothesis that we will try to map the data to $H_{1_{mod}}$. At this point, the match-filter statistic fails because there is not a direct mapping of the data to $H_{1_{mod}}$. Nonetheless, one can create a detection statistic using the log-likelihood where we maximize over all signal templates

$$\max_{\theta}\left(2\log L(d|s(\theta))\right). \tag{4.31}$$

There are two critical components involved in the maximization above. First, if possible, it is useful to marginalize out some of the parameters that make up θ. Second, that the data $\mathbf{d}$ has been projected onto unit vectors that span the space of signals, because then we can perform the maximization analytically. Otherwise, it quickly becomes too computationally expensive to perform the maximization above.

In section 4.2.3, we discussed how *templates*, $\mathbf{e}^+$ and $\mathbf{e}^\times$, span the space of possible plus and cross polarized GWBs. Therefore, if we can marginalize out the uncertainty in the amplitude of the signal, we can use our templates to make a detection statistic. We must marginalize out the amplitude of both templates separately as $\mathbf{e}^+$ and $\mathbf{e}^\times$ are independent.

To do so, we solve for

$$\frac{\delta \log L}{\delta A} = 0, \tag{4.32}$$

and

$$\frac{\delta \log L}{\delta B} = 0, \tag{4.33}$$

where A and B is the amplitude of the plus and cross templates, respectively. We rewrite Equation 4.28 where $F^+h_+ + F^\times h_\times = (A\mathbf{e}_+ + B\mathbf{e}_\times)$, remembering that the templates are constructed so that $|\mathbf{e}_+|^2 = 1$ and $|\mathbf{e}_\times|^2 = 1$. We can now write the derivative of the likelihood with respect to A and B as,

$$\frac{\delta \log L}{\delta A} 2\left(\left((\mathbf{d}\cdot A\mathbf{e}_+) - ||A\mathbf{e}_+||^2\right) + \left((\mathbf{d}\cdot B\mathbf{e}_+) - ||B\mathbf{e}_\times||^2\right)\right) = 0, \tag{4.34}$$

and

$$\frac{\delta \log L}{\delta B} 2\left(\left((\mathbf{d}\cdot A\mathbf{e}_+) - ||A\mathbf{e}_+||^2\right) + \left((\mathbf{d}\cdot B\mathbf{e}_+) - ||B\mathbf{e}_\times||^2\right)\right) = 0. \tag{4.35}$$

$A = \mathbf{d}\cdot\mathbf{e}_+$ and $B = \mathbf{d}\cdot\mathbf{e}_\times$, respectively, solve for the above. Finally, after maximising out the amplitude, the appropriate matched-filter detection statistic is as follows

$$S_{MF} = \max_{\mathbf{e}_+,\mathbf{e}_\times}\left((\mathbf{d}\cdot\mathbf{e}_+)^2 + (\mathbf{d}\cdot\mathbf{e}_\times)\right)^2\Big). \tag{4.36}$$

The above is equivalent to the *standard likelihood* (SL) statistic of a single-time frequency pixel which is described in more detail in [23]. More concretely, expanding past a single time-frequency pixel, and summing over a *cluster* of pixels, Equation 4.36 takes the form

$$E_{\mathrm{SL}} = \sum_k \left(|\mathbf{d}\cdot\mathbf{e}_+|^2 + |\mathbf{d}\cdot\mathbf{e}_\times|^2\right). \tag{4.37}$$

where k is the number of pixels in the cluster.

Under the assumption of Gaussian background noise, this statistic, E_{SL}, follows a χ^2 distribution with $2N_{\mathrm{p}}D_{\mathrm{proj}}$ degrees of freedom where N_p is the number of pixels in the cluster, D_{proj} is the number of dimensions of the projection, in this case 2, and the factor of 2 comes from the fact that the data is complex.

$$2E_{SL} \sim \chi^2_{2N_{\mathrm{p}}D_{\mathrm{proj}}}(\lambda_{\mathrm{SL}}) \tag{4.38}$$

The non-centrality parameter λ_{SL} is the expected squared SNR, i.e. Equation 4.30, and has the form

$$\lambda_{SL} = \frac{4}{N} \sum_a \sum_k \left(F_a^+ h_+[k] + F_a^\times h_\times[k] \right) =: \rho^2. \tag{4.39}$$

We note here again that the data has been whitened.

The non-central χ^2 distribution 4.38 has a mean and standard deviation of $4N_p + \lambda_{SL}$ and $\sqrt{4N_p}$, respectively. Therefore, under this statistic a signal is expected to be detectable when

$$\frac{\lambda_{SL}}{2\sqrt{N_p}} \gg 1. \tag{4.40}$$

.

Essentially, every pixel adds D_{proj} Gaussian noise to the signal statistic and therefore the more pixels considered part of the GWB, than the louder the statistic is required to be to be considered significant. Concretely, we need the signal contribution to be significantly larger than the typical fluctuation of the noise. In addition to adding more noise, each additional pixel with signal is also contributing to the overall signal statistic. For a discussion of when to stop adding pixels to a cluster, see Chapter 5 of [26].

In the following section, we discuss how the generalized log-likelihood statistic introduced in Equation 4.31 and solved in the form of the standard likelihood in Equation 4.37 might not be ideal in the case of GWBs as out signal templates are not as reliable. Instead, we discuss how Bayesian statistics, by including priors, can help solve some of the problems surrounding the creation of a detection statistic in the case of GWBs.

4.2.5 Building a Bayesian Detection Statistic for Gravitational Wave Bursts

We explore the Bayesian framework for creating the detection statistic. The intuitive difference between the optimal match filter framework and the Bayesian framework is as follows. When trying to map the data to the hypothesis, $H_{1_{mod}}$, where the data contains the signal $s(\theta)$ in the space of possible signals spanned by parameters θ, the match filter statistic relies on the concept of templates as a way to produce a likelihood of such a signal existing in the data. Doing so requires not only mapping the data to the space covered by the template, but also covering a large grid of possible templates. As discussed earlier, in the case of the standard likelihood, we managed to avoid using a large grid of templates by using dimensions in the space of signals which are vector spaces which allowed us to analytically maximize Equation 4.36. Bayesian statistics, however, solves for $H_{1_{mod}}$ by introducing prior probabilities on the parameters in θ and by using conditional probabilities. More

over, as already described we have two orthogonal template vectors whose linear combinations span the entire manifold of possible signals. We will refer to them as $s_1(\theta)$ and $s_2(\theta)$, where $s_1(\theta)$ is $A\mathbf{e}^+$ and $s_2(\theta)$ is $B\mathbf{e}^\times$ with respect to the discussion in Section 4.2.4. In this framework, the likelihood ratio for hypothesis $H_{1_{mod}}$ takes the form

$$L(\mathbf{d}) = \int_\theta L(\mathbf{d}|s_1(\theta) + s_2(\theta))p(\theta)d\theta. \tag{4.41}$$

As with the previous discussion, it is important to determine what parameters are incorporated into θ and what priors can be assigned to those parameters. One unknown parameter of the GWB is the time of arrival, which can be assigned a flat prior. Even for the case of the neutrino triggered supernova search where a window of only $[-1, +2]$ around the neutrino event is searched [115], it makes sense to set a flat prior on the time of arrival within this window of 3 seconds. Another unknown parameter is the amplitude, A and B. A and B have a prior expectation associated with it because only signals that are as strong or stronger than the current sensitivity of the detectors are of interest. More precisely, we expect a GWB will have some characteristic strain is Gaussian distributed with standard deviations of σ_{h1} and σ_{h2} that [116]. There is no astrophysical motivation for this prior other than it is analytically solvable.

Therefore, we now explicitly write $s_1(\theta)$ and $s_2(\theta)$ in the form used in the previous section $A\mathbf{e}^+$ and $BA\mathbf{e}^\times$ where $p(A)$ is

$$p(A) = \frac{1}{\sqrt{2\pi\sigma_{h1}^2}} \exp\left(-\frac{1}{2}\frac{A^2}{\sigma_{h1}^2}\right), \tag{4.42}$$

respectively, $p(B)$.

Now we can rewrite 4.34, marginalizing over the amplitude, as

$$L(\mathbf{d}|\theta) = \int_A L(\mathbf{d}|A\mathbf{e}^+ + B\mathbf{e}^\times)p(A)p(B)dAdB. \tag{4.43}$$

$L(\mathbf{d}|\theta)$ has the form of Equation 4.27, where $(h_+, h_\times)$ is now $(A\mathbf{e}^+, B\mathbf{e}^\times)$. Therefore, we can write the marginalized likelihood as

$$\begin{aligned} L(\mathbf{d}|\theta) = \int_A & \frac{dA}{\sqrt{2\pi\sigma_{h1}^2}} \frac{dB}{\sqrt{2\pi\sigma_{h2}^2}} \\ & \exp\left((\mathbf{d}\cdot A\mathbf{e}^+) + (\mathbf{d}\cdot B\mathbf{e}^\times) - \frac{1}{2}(A^2+B^2) - \frac{1}{2}\left(\frac{A^2}{\sigma_{h1}^2} + \frac{B^2}{\sigma_{h2}^2}\right)\right). \end{aligned} \tag{4.44}$$

Here we pause to remember that, as mentioned above, $\mathbf{d}$ is not obtained via a continuous but a discrete STFT. Luckily, when the data is discrete and in the Fourier basis, the mathematics needed to solve for the likelihood above is simplified. The dot product from Equation 4.29 can be written as the sum

$$a \cdot b = \sum_{k=0}^{k=N-1} \frac{2a_k^* b_k}{|S_k|} = \mathbf{a}^\dagger \mathbf{R} \mathbf{b} \tag{4.45}$$

where a_k are the Fourier coefficients of vector $\mathbf{a}$, b_k are the Fourier coefficients of vector $\mathbf{b}$ and $\mathbf{R}$ is a diagonal matrix with coefficients of $\frac{2}{|S(k)|}$.

In finite dimensions, it is easy to define the multivariate Gaussian distribution. Given a matrix $\mathbf{R}$ where $\mathbf{R}^\dagger = \mathbf{R}$ and the eigenvalues of $\mathbf{R}$ are positive, a vector $\vec{\mathbf{b}}$ and a constant c, the Gauss Integral is written as

$$\int_{R^M} \exp\left(-\frac{1}{2}\mathbf{x}^\dagger \mathbf{R} \mathbf{x} + \vec{\mathbf{b}}^\dagger \mathbf{x} + c\right) d\mathbf{x}. \tag{4.46}$$

The Gauss Integral has the solution

$$\sqrt{\det 2\pi \mathbf{R}^{-1}} \exp\left(\frac{1}{2}\mathbf{b}^\dagger \mathbf{R}^{-1} \mathbf{b} + c\right). \tag{4.47}$$

Therefore, if we can reformat the likelihood in such a way that it mimics the form of the Gauss integral, we will be able to solve for it analytically. In our case, if we do the following substitutions,

$$\mathbf{R} = \begin{pmatrix} 1 + \frac{1}{s_{1h}^2} & 0 \\ 0 & 1 + \frac{1}{s_{2h}^2} \end{pmatrix} \tag{4.48}$$

and

$$\mathbf{b} = \begin{pmatrix} (\mathbf{d} \cdot A\mathbf{e}^+) \\ (\mathbf{d} \cdot B\mathbf{e}^\times) \end{pmatrix} \tag{4.49}$$

we obtain the Gauss integral.

When the marginalization is complete, we log the result and are left with a log-likelihood statistic which has the form

$$\log(L(\mathbf{d}|\theta)) = \frac{(\mathbf{d} \cdot \mathbf{e}^+)^2}{1 + \frac{1}{\sigma_{h1}^2}} + \frac{(\mathbf{d} \cdot \mathbf{e}^\times)^2}{1 + \frac{1}{\sigma_{h2}^2}} - \log\left(\sigma_{h1}^2 + 1\right) - \log\left(\sigma_{h2}^2 + 1\right). \tag{4.50}$$

At this point, all that is left to do is determine how our templates $\mathbf{e}^+$ and $\mathbf{e}^\times$ fit into the above. Again, it is computationally tractable to assume Gaussian priors with a standard deviation of σ_h on the amplitude of h_+ and $h_\times$, and, therefore, our templates would have amplitude priors of $\sigma_{h1} = \sigma_h |\mathbf{f}^+|$ and $\sigma_{h2} = \sigma_h |\mathbf{f}^\times|$, respectively. Plugging into the Bayesian log-likelihood formulation above, we obtain

$$\log(L(\mathbf{d}|\sigma_h)) = \frac{(\mathbf{d} \cdot \mathbf{e}^+)^2}{1 + \frac{1}{(\sigma_h |\mathbf{f}^+|)^2}} + \frac{(\mathbf{d} \cdot \mathbf{e}^\times)^2}{1 + \frac{1}{(\sigma_h |\mathbf{f}^\times|)^2}} - \log\left((\sigma_h |\mathbf{f}_+|)^2 + 1\right) - \log\left((\sigma_h |\mathbf{f}_\times|)^2 + 1\right). \tag{4.51}$$

The form of the Bayesian likelihood may appear similar to the standard likelihood. In fact, following [26] it can be shown that both of these statistics reduce to the same quantity when the waveform template is well known beforehand. The main difference between the likelihoods is that the Bayesian likelihood provides an additional penalty for signals whose amplitudes are not consistent with our expectation. Specifically, it down-weights pixels for which the expected SNR, σ_f^2, is small. Therefore, it is useful to quantify what our amplitude expectation for GWB signals should be.

It is not obvious what σ_h should be but we know it should simultaneously be linked to astrophysical models while also not straying too far from the current sensitivity of the detector. An easy way to get around this issue is to choose a uniformly log-distributed discrete set of σ_h and marginalize the log-likelihood above. We select strains $[10^{-23}, 10^{-21}]\text{Hz}^{-\frac{1}{2}}$ which correspond to uniformly log-spaced values of σ_h. From this, the final detection statistic looks as follows

$$L(\mathbf{d}|A) = 2\log\left(\sum_{\sigma_h \in A} \frac{1}{\sigma_h} \exp\left(\frac{1}{2}\log(L(\mathbf{d}|\sigma_h))\right)\right). \tag{4.52}$$

In either detection statistic, the standard likelihood or the Bayesian log-likelihood, we rely on the squared projection of the data onto template vectors $\mathbf{e}^+$ and $\mathbf{e}^\times$. Over and above the standard likelihood, the Bayesian log-likelihood provides a penalty when the amplitude of the pixel is inconsistent with our prior expectation on the amplitude of GWBs. Both statistics, however, are still susceptible to the impact of loud transient glitches in the data. Even though the data vector in this case should not be aligned with projection vector (since it is not a GWB), the data vector is still expected to have a large value with respect to the dot product (i.e. the numerator) and can create large values in either statistic. It is for this reason that we spend the following section discussing how to reject these glitches.

4.2.6 Coherent Consistency Checks

One critical assumption prevalent throughout the discussion of the detection statistic section was the idea that the background noise distribution was consistent with Gaussian noise. However, in reality the noise is filled with loud noise transients, also known as glitches, a number of which were discussed at length in Chapter 2. The attempt to make the background noise Gaussian motivates this section and highlights a significant advantage of the coherent analysis. In general, loud noise transients are uncorrelated among the detectors, unlike the GW signal which is correlated in a particular way across the detectors dictated by the antenna response functions, and, therefore, we can create statistics and perform tests to eliminate many of these transients.

We briefly mentioned one such statistic at the end of Section 4.2.3. For this coherent consistency statistic, we can take the part of data $\mathbf{d}$ that is orthogonal to

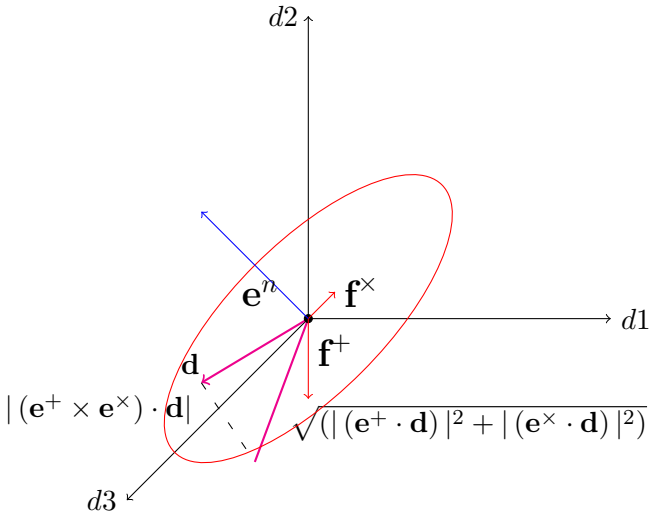


Figure 4.4: Illustration of the representation of data $\mathbf{d}$ in the space of detector strains for the three detector case. The red ellipse is the sensitivity of the detector network to linearly polarized gravitational waves. We mark the plane spanned with the unit detector response vectors $\mathbf{f}^+$ and $\mathbf{f}^\times$ and the orthogonal vector to the plane as $\mathbf{e}^n$ which forms the null space. See Equation 4.14 for more details. Here we see the projection of data $\mathbf{d}$ onto the plane spanned by $\mathbf{f}^+$ and $\mathbf{f}^\times$. This (bold) line represents the standard likelihood, the (dashed) line parallel to $\mathbf{e}^n$ represents the null energy of $\mathbf{d}$.

the plane spanned by the plus and cross unit vectors and compare the part of the data in that projection to Gaussian noise. We refer to this projection of the data as the *null space*, i.e. the space orthogonal to the unit orthonormal basis $\mathbf{e}^+$ and $\mathbf{e}^\times$, defined as $\mathbf{e}^n$. An example of this projection is illustrated in Figure 4.4. We define two terms, the coherent and incoherent null energy labelled as E_n and I_n, respectively. For a projection vector $\mathbf{e}$ and for $\alpha, \beta \in D$ where D is the number of detectors, the coherent energy, a measure of cross-correlation, has the form

$$E_n = ||\mathbf{e}^n \cdot \mathbf{d}||^2 = \sum_{\alpha,\beta} e^{n*}_\alpha e^n_\beta d^*_\alpha d_\beta. \tag{4.53}$$

The incoherent energy, a measure of auto-correlation, is the sum of the diagonal terms of the vectors,

$$I_n = \sum_\alpha e^{n*}_\alpha e^n_\alpha d^*_\alpha d_\alpha = \sum_\alpha ||e^n_\alpha||^2 ||d_\alpha||^2 \tag{4.54}$$

where $*$ represents the complex conjugate of the variable. At this point, we can sketch where we would expect a GWB and a glitch to fall on a line of equal coherent and incoherent null energy, referred to as the $E_n - I_n$ line. To do this, it is useful to revisit the discussion around the motivation behind the standard likelihood detection statistic. This statistic represents the maximum amount of energy that is consistent with the hypothesis that there is a GWB in the data. We note here that the

max likelihood is the squared magnitude of the projection into the $f_+, f\times$ plane. Conversely, the null stream is a the minimum amount of energy that is inconsistent with the hypothesis that there is a GWB in the data. That is, data vectors which contain signal should only deviate from the space spanned by the antenna response vectors by a small amount. Therefore, one can think of the null stream as simply the energy leftover after the maximum energy consistent with a GW is subtracted out. This residual energy should be consistent with Gaussian noise. To this end, we introduce the *total energy* which can be found by taking the dot product of the data with itself

$$E_{tot} = ||\mathbf{d}||^2. \tag{4.55}$$

Therefore, the null energy can be represented as

$$\begin{aligned} E_n &= E_{tot} - E_{SL} \\ &= ||\mathbf{d}||^2 - ||\left(\mathbf{e}^+ \cdot \mathbf{d}\right)||^2 - ||\left(\mathbf{e}^\times \cdot \mathbf{d}\right)||^2. \end{aligned} \tag{4.56}$$

With this description of the coherent null energy in mind, we can motivate what side of the $E_n - I_n$ line GWBs will fall when it comes to the incoherent and coherent null energy. The E_n of a GWB will be small as the cross correlation terms of the statistic cancel in the null stream leaving only the uncorrelated Gaussian noise. Conversely, in the auto-correlation terms the energy contributions from the signal will not cancel for a GWB. Therefore, we anticipate that GWBs will fall to the left of the $E_n - I_n$ line, i.e. $E_n \ll I_n$. The glitch, however, will tend to have similar coherent and incoherent null energy because the signal is not cross correlated between detectors and, therefore, the energy is almost entirely in the auto-correlation terms. Therefore, the null projection operator will not have the impact of canceling any cross correlation components and so $E_n \simeq I_n$. Another way to intuitively think of the above test is as a phase and amplitude consistency test. If the phase and amplitude is consistent across all three detectors, for instance, then the data will not lie far away from the plane spanned by the antenna response vectors. If it is not consistent, say there is a large amplitude glitch in one of the detectors, then it is likely that the vector will point far enough from the plane that the null projection is too large a value to be consistent with Gaussian noise.

The null energy is not the only available coherent consistency check that can be performed. When in the realm of only two detectors, i.e. there are not enough data streams to have a null projection of the data, one can still perform coherent checks that amount to checking for consistent phase across detectors. To confirm that this is the case, one can imagine a noiseless signal in two aligned detectors. In such a situation the coherent and incoherent energies would be

$$E = |A^2 + A^2| = 4|A|^2 \tag{4.57}$$

and

$$I = |A|^2 + |A^2| = 2|A|^2, \tag{4.58}$$

respectively, where A is the amplitude of the projected signal.

For a specific example, we can look at the coherent and incoherent plus and cross energies. We have constructed the projection such that a predominantly plus polarized signal should have a build up of coherent plus energy and end up to the right of the $E_+ - I_+$ line, i.e. $E_+ > I_+$, and, conversely, should have lack of build up of coherent cross energy and end up to the left of the $E_\times - I_\times$ line, i.e. $E_\times < I_\times$. By contrast, one would expect the opposite to be true of a predominately cross polarized signal. That is, the signal should have a build up of coherent cross energy and end up to the right of the $E_\times - I_\times$ line, i.e. $E_\times > I_\times$, and, conversely, should have lack of build up of coherent plus energy and end up to the left of the $E_+ - I_+$ line, i.e. $E_+ < I_+$.

$$\begin{matrix} E_+ > I_+ \\ E_\times < I_\times \end{matrix} \quad \text{predominantly plus polarized} \tag{4.59a}$$

$$\begin{matrix} E_+ < I_+ \\ E_\times > I_\times \end{matrix} \quad \text{predominantly cross polarized} \tag{4.59b}$$

Figure 4.5 demonstrates how we can use a threshold on the difference between the coherent and incoherent plus (respectively cross) energies outlined above to distinguish between simulated GWBs and detector noise. In this case, it is clear that the signal was predominately + polarized as the simulated GWB (□) falls to the right of the line of equal coherent and incoherent plus energy, but the background (x) falls on this line. Whereas the simulated GWBs fall to the left of the line of equal coherent and incoherent cross energy. This is an example of performing a "two-sided" cut in the space of coherent and incoherent energies. The motivation for accepting triggers that lie either to the left or right of either line is in line with the reasoning described in 4.59.

We have demonstrated that the projection of the data **d** onto the plane spanned by the antenna response vectors can motivate selecting functions that slice the coherent and incoherent likelihoods such that loud incoherent glitches are rejected. We have yet to discuss an appropriate way to determine the best exploration of this space. In the following Section, we discuss how "X-Pypeline" leverages injections of fake GWBs and the creation of training sets of clusters of pixels that are known to contain and not to contain GWs to choose the optimal slicing of this space. An example of one such slicing is displayed in Figure 4.5. In addition, we discuss how "X-Pypeline" can also leverage random forests (RF), convolutional neural networks (CNN), and Gaussian processes (GP) to explore this space.

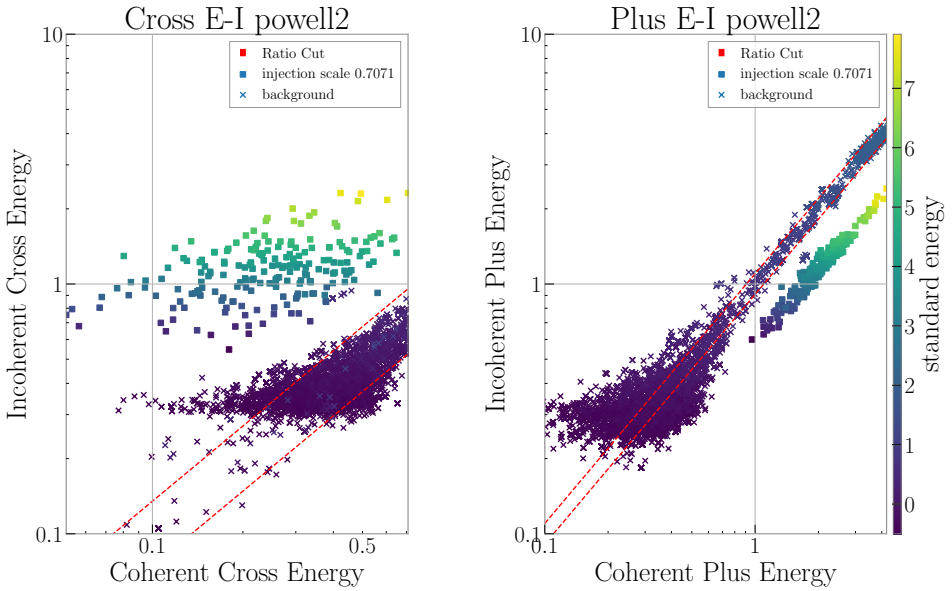


Figure 4.5: Example of E_+ versus I_+ and $E_\times$ versus $I_\times$ for clusters produced by background noise (x) and by simulated GWBs (□). The color scale is the base-10 logarithm of the detection statistic, in this case the standard likelihood, associated with each cluster. We can see that this is an example of a predominantly + polarized GW signal as the coherent and incoherent plus energies fall to the right of the $E - I$ line, and, conversely the coherent and incoherent cross energies fall to the left of the $E - I$ line.

4.3 Tuning the Analysis in X-Pypeline

To this point, we have motivated our choice of using projected time-frequency pixels as templates for searching for GWBs. They have been useful when creating detection statistics and when creating statistics to reject spurious noise transients. In the following sections, we discuss how we utilize these statistics to make detection statements with "X-Pypeline". First, in Section 4.3.1 we present the standard method for tuning the coherent consistency cuts and providing a statement of detection of a gravitational wave at a given *False Alarm Probability* (FAP). Second, in Section 4.3.2 we present a different way of tuning the coherent consistency cuts and providing a statement of detection at a given FAP through the use of Random Forests (RF). Third, in Section 4.3.3 we present a different way of tuning the coherent consistency cuts and providing a statement of detection at a given FAP through the use of Convolutional Neural Networks (CNN). Finally, in Section 4.3.4 we present a different way of tuning the coherent consistency cuts and providing a statement of detection at a given FAP through the use of Gaussian process (GP).

4.3.1 Standard Tuning

The goal of any "X-Pypeline" analysis is to be able to make a claim of a gravitational wave detection with some confidence which we refer to as the FAP. In order to obtain confidence of a detection at a given FAP, we also need to replicate the analysis laid out in Section 4.2 on data believed to *not* contain the GWB. We refer to this data as the "off source" data. Conversely, the data which is believed to contain the GWB is called the "on source" data. Repeating our analysis on the off source data through some number of independent trials will allow us to make statements about the chances that the statistical properties observed in the on source could be obtained randomly from noise alone. For a specific example, imagine we want to claim a detection at a FAP of 1%, then we would have to perform at least 100 identical analyses on off source data. For each analysis, we would obtain the loudest (using the likelihood statistic) event and then take the loudest event from that list of 100 trials. However, this is not enough as we know that the data could contain glitches which impact our detection statistics. For this reason, we must use coherent consistency checks. In order to understand where we can safely make cuts in the incoherent and coherent energy space, we must also know what types of coherent and incoherent energies we can expect from a GWB at the time of the analysis. That is, depending on the sky location, the sensitivity of each detector, the time-frequency shape of the expected waveform, and the types of glitches, the function that optimally cuts the coherent and incoherent parameter space will be different. To appropriately probe this space, we simulate GWBs of varying amplitudes and families (time-frequency shapes) as proxies for the real GWB. These fake signals are added to into nearby chunks of gravitational wave data so that a.) the simulated

signal is not injected on top of the actual signal, and b.) so that the noise added to the fake injections is as much like the noise that may be part of the real GWB. These fake injections can form a training set that can be used to tune the cuts in the coherent and incoherent energy space. However, in order to tune the appropriate cuts we must also have a training set of background clusters. Because we perform this additional tuning step (using half of the injection and background trials), we must double the number of trials we perform on injections and background data in order to achieve the same FAP.

In summary, to run "X-Pypeline" the following information must be input by either a human or automated triggering software

- a desired False Alarm Probability
- a set of waveforms families, amplitude scaling, and number of injections at each amplitude
- set of detector data streams
- interval of data to be analyzed
- sky locations
- a list of STFT lengths to use.

From these inputs, we obtain clusters of pixels and their statistical properties from independent runs of the algorithm on data not containing signals and data that does contain simulated signals. To these clusters, we perform the following post-processing steps that allow detection statements to be made by the final output of "X-Pypeline".

First, we reject clusters whose duration overlap with *data quality vetoes* which are provided as part of detector characterization efforts in the LIGO and Virgo Collaborations. Data quality vetoes consist in a list of start and stop times dictating times when one or more detectors were effected by non-GW disturbances, i.e. environmental or instrumental, that cause loud glitches. Data quality vetoes are provided separately from "X-Pypeline" by algorithms Hierarchical Veto, iDQ, and Karoo GP which are described in [20, 117, 118] and where discussed in Chapter 2. Second, we reject clusters from injection trials whose duration in time does not overlap the time of the injection within some window. This is intuitive, as we do not want to tune cuts using clusters that are implausibly connected to the injection. Third, we apply an asymmetric on-source window cut. That is, we have astrophysically motivated time windows within which a counter part such as a neutrino trigger and the GW can be seen (for instance $[-1, +2]$ so we throw out all clusters from our injection, on source, and background trials that do not fall within this window. Injections are set up so that they occur within a specified on-source time window (which does not need to be the real on-source window) so this criteria should not affect injections.

Once these steps have been completed, we can use the surviving triggers from the injection and background trials to tune coherent cuts. In order to understand what kind of cuts in this space would make sense, we take a step back and revisit the properties of the coherent and incoherent energies. The goal of this step is to remove high amplitude glitches while not removing high amplitude GWBs. As mentioned in Equation (4.59), depending on the polarization of the signal we expect a build up or cancellation of coherent energy due to the GWB existing in the cross correlation terms. A large amplitude glitch, however, will have $E \simeq I$ and so we state that a candidate cluster survives the *ratio* coherent test if it survives all of the following conditions,

$$\frac{I_{\text{null}}}{E_{\text{null}}} > c_{\text{null}}, \tag{4.60a}$$

$$\left|\log_{10}\left(\frac{I_+}{E_+}\right)\right| > \log_{10}\left(c_+\right), \tag{4.60b}$$

$$\left|\log_{10}\left(\frac{I_\times}{E_\times}\right)\right| > \log_{10}\left(c_\times\right), \tag{4.60c}$$

where c_+, $c_\times$, and c_{null} are values that the "X-Pypeline" algorithm automatically discovers through a training set of injection and background clusters.

This is not the only type of coherent consistency test available in "X-Pypeline". As denoted in equations (4.57) and (4.58), for a large amplitude GWB of amplitude A, i.e. one where the contribution to the amplitude from Gaussian noise can be ignored, the coherent and incoherent energies have a A^2 dependence. Because the signal is cross correlated between the detectors the coherent energy minus the incoherent energy will also have an A^2 dependence, i.e. the incoherent energy does not have cross correlation terms and the coherent energy does so when they are subtracted the cross correlations terms remain in the statistic. However, for a loud glitch of amplitude A in one detector the cross correlation terms are between the glitch and Gaussian noise in the other detector(s). Therefore, the coherent energy minus the incoherent energy will only be proportional to A. We leverage this different behavior between the coherent minus incoherent energy to introduce another function for rejecting glitches

$$g(I, E|\alpha) = \frac{|E - I|}{(E + I)^\alpha} > c. \tag{4.61}$$

Similarly to the ratio test, we compare the output of the above function to some threshold c. Every choice of (c, α) elicits a different way to slice the coherent and incoherent energy space. For more discussion of the impact of different α selections on rejecting glitches and keeping signals see [26]. In practice, we utilize a value of $\alpha = 0.8$ and, as with the ratio test, c is a value that the "X-Pypeline" algorithm automatically discovers through a training set of injection and background clusters.

The above test is called the *alpha ratio test*.

We have introduced two ways to slice the coherent and incoherent energy space, but have yet to present a method for which cuts C_{alpha} and C_{ratio} can be automatically selected by the algorithm. Given a discrete set of cuts $c \in C_{\text{ratio}}$, the ratio coherent test is applied to all clusters in each background trial. The surviving cluster with the maximum value in the detection statistic, for instance the standard likelihood, is selected from each background trial. Given a FAP, the algorithm then selects the $(1-p) * N_{\text{trials}}$ largest value in the detection statistic from the list of the loudest surviving cluster from each trial.

At this point, the same discrete ratio cuts are applied to clusters from the injection trials, with the additional requirement that the detection statistic from the injection cluster is higher than that of the above selected background cluster. We now have a basis for selecting the best value for the ratio cut. We have information on exactly how many clusters from injection trials are louder than the loudest background at a given FAP and are not rejected by each cut, but we need a metric to select the most "efficient" cut. The sum squared fractional excess upper limit (SSFEUL) is used to define the optimal cut value for a given FAP. Given a desired detection efficiency, typically 50% or 95%, this statistic determines the cut which allows for claiming a detection for a given waveform family at the lowest amplitude. The SSFEUL has the following form

$$\min_{c \in C} \left(\sum_{w \in W} \left(\frac{(A_w - \min(A)}{\min(A)} \right)^2 \right) \tag{4.62}$$

where the minimum amplitude is the smallest amplitude for a given waveform w across cut values tested such that the desired fraction of the injection clusters are recovered. As X-Pypeline does injections at discrete amplitudes, often interpolation between amplitudes is required to determine the amplitude for each waveform family at which the desired fraction of clusters are recovered. We then take the minimum value from the SSFEUL across cuts $c \in C$ and use these cut values when applying cuts to the clusters from the remaining injection, background, and on-source trials. The same process is used for determining the appropriate value for the alpha cut, with the additional requirement that clusters are also submitted to being rejected from the ratio cut values found through the above method.

To summarize,

- Split the injection and off source trials in half using clusters from 50 percent of the trials for tuning the coherent cuts.
- Apply data quality, asymmetric, and coincident window vetoes.
- Over a discrete set of ratio cuts, determine which value provides the best SSFEUL for recovering 95 percent of the injections.

- After using the ratio cut determined above, over a discrete set of alpha cuts, determine which value provides the best SSFEUL for recovering 50 percent of the injections.
- Take the remaining clusters from the other half of the injection and off source trials and repeat the above steps using the tuned alpha and ratio values in order to measure the background distribution and efficiency of the tuned analysis.

We have sketched how to tune coherent cuts to reject loud background clusters, in order to improve the robustness of our detection statistics. One of the key issues with the methodology laid out above is that the optimal value for each of c_+, $c_\times$, and c_{null} requires searching over a grid of cut options that totals $(N_+ * N_\times * N_{\text{null}})$ where each N is the total number of discrete values $C_{+,\times,\text{null}}$. Further computational costs are incurred if the analysis tunes cuts in other coherent and incoherent energy spaces that can be obtained from assuming a different polarization of the signal, such as circular or scalar signals. Moreover, the above methodology assumes that each remaining cluster after the cuts have been applied, is ranked by *one* of the detection statistics described earlier. It is sometimes desirable to try out multiple detection statistics for a given analysis. It is for these reasons we introduce the idea of using machine learning techniques, such as random forests and convolutional neural networks, to find functional form of the coherent cuts and make decisions on the probability of a cluster being a signal or not.

4.3.2 Using Random Forests to Tune

It is useful to think of every coherent, incoherent and detection statistic as a *feature* of the cluster. In some ways, the motivation behind the coherent consistency checks described above is that clusters from injection trials are expected to have different features than those from background trials. Thinking of these statistics as features associated with each cluster, in addition to the fact that training and testing sets are created during the tuning process described in 4.3.1, indicates that machine learning techniques may be suitable to apply to the data. The simplest way to consider coherent consistency checks and the FAP tuning described above is that we are making a decision for every cut and detection statistic combination on whether it effectively rejects background and retains signal. To this end, it seems suitable to use a decision tree as a basis for deciding whether a cluster is likely to be associated with a GWB or a background event. Simply put, a decision tree amounts to asking a series of true or false questions in order to determine a given outcome. Imagine we started off with 100 clusters that are associated with background and 100 clusters that are associated with injections. A decision tree would utilize the *features* of those clusters, such as the value of the standard likelihood and the value of the ratio between the coherent and incoherent energy, to split the 200 samples into two sub groups based on the answers to a true or false question. From those

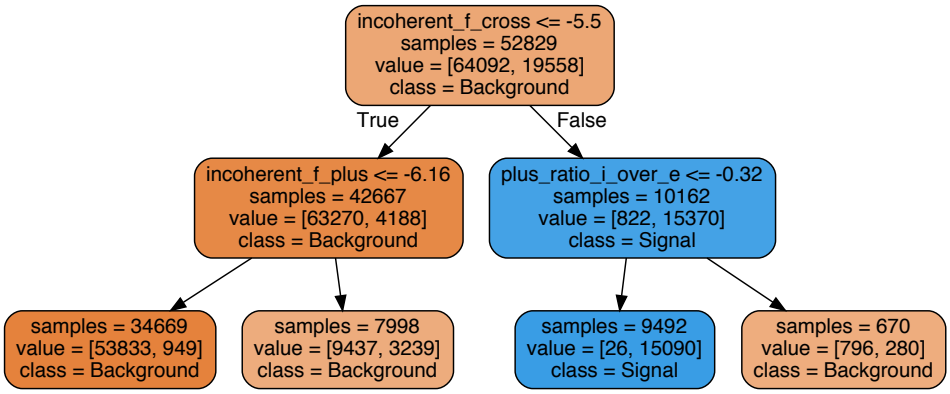


Figure 4.6: Example of decision tree utilizing the coherent, incoherent and detection statistics associated with background and injection triggers to predict whether the cluster is a signal or background. Samples are the number of events from injection trails and off source trials. Samples can be one of two classes, background or signal. Values are the number of samples with true label background or signal that fell into a given node based on the question asked in the node above. Nodes coloured orange indicate events with features consistent with background and those coloured blue indicate events with features consistent with signal. Due to the bootstrap sampling (i.e. replacement sampling), the numbers in the values do not match up with the number reported by samples. This decision tree is limited to only two layers and makes decisions based on the incoherent plus and cross energies and the ratio of incoherent to coherent plus energy. As expected, even in such a small decision tree, a threshold on the ratio between the coherent and incoherent energies proves a valuable way to distinguish signals from the noise.

two sub groups, referred to as nodes, additional true/false questions will be asked, sequentially, until the original 200 samples have exactly split into the inputted two bins of 100 injections and 100 background or further splitting does not positively add any predictive power. Figure 4.6 illustrates an example of how a decision tree may look when run on the training sets of injection and background clusters. In this example, of the limited statistics used in this decision tree, an important distinguishing characteristic between signal and background was a threshold on the ratio between the incoherent and coherent energy. In fact, based on the reasoning from Equation 4.59, this is an example of a predominantly plus polarized signal as the question asked by the node at a depth of two is

$$\log_{10}\left(\frac{I_+}{E_+}\right) <= -0.32 \tag{4.63}$$

Specifically, the node asks if the coherent energy build up was larger than the incoherent, and when the answer is false, this is predictive of clusters associated with signals, and when true, predictive of clusters associated with the background.

One can easily imagine that the decision tree as shown in Figure 4.6, which is restricted to a depth of two and uses only some of the available statistics associated with each cluster, is unlikely to overfit to the training data but may miss out on information that provides valuable predictive power. Overfitting is a modeling error which occurs when a function is too closely fit to a limited set of data points. Overfitting occurs when an overly complex model is created to explain idiosyncrasies in the data under study. In our case, this over complex model would happen, if the decision tree was very large and given access to all of the metadata. Therefore, in practice, one never utilizes a single decision tree to make a prediction. Instead, it is common to leverage an ensemble of decision trees. This eliminates fears of overfitting and improves robustness. A specific ensemble of decision trees is the basis of the method we will use in practice, random forests. The idea behind a random forest is that every decision tree will have a certain variance associated with its predictions. If we average across the predictions of hundreds of decision trees, however, we will hone in on the correct prediction. In this way, we use a "forest" of decision trees whose results are pooled or averaged to make a more robust prediction. Specifically, each cluster is assigned a score that is a weighted average of the predictions of the individual trees (0=background, 1=signal). In addition, each decision tree in the forest has access to a limited number of training set samples and features associated with those samples. This motivates the "random" component, by limiting the information available to make a decision, we guarantee that every decision tree in the forest will be a little different. This variation in information across trees will, again, lead to a more robust prediction.

It seems clear that the use of a random forest in the case of the data provided by "X-Pypeline" is motivated, and to actually implement in order to provide de-

tection statements, we follow a similar methodology as highlighted in Section 4.3.1. After applying all of the same rejection criteria to clusters as in 4.3.1, we feed all remaining clusters from the training injection and background trials into a random forest classifier. This classifier has the following properties: 100 decision trees each with a max depth of five and access to $\sqrt(N)$ where N is the total number of statistics associated with every cluster. For instance, when assuming a plus and cross polarized signal, as we have assumed so far in this thesis, one can construct about 14 statistics for every cluster. The number of features we give each tree access to can effect how generally applicable our forest is in two ways: selecting many features increases the strength of the individual trees whereas reducing the number of features leads to a lower correlation among the trees increasing the strength of the forest as a whole. We use $\sqrt(N)$ as it is recommended as a value which balances the above concerns []. Moreover, we use a max depth of 5 as it was suitable for our classifications needs and the larger the max depth the longer the training time. We note here that some of the statistics (or features) are different representations of the same values. For instance, the coherent and incoherent plus energy as well as the ratio of the two values are all separate statistics. This means every decision tree has access to some number of clusters from the training set and 4 statistics. After training the classifier, we compute the forest-averaged classifier score of signal or noise for every cluster in each trial from the background set aside for testing. From each trial we take the cluster that received the highest score (most likely to be of signal). Taking the $(1-p)*N_{\text{trials}}$ largest value from this list is the threshold that injection clusters must exceed in order to be considered as"detected". In Chapter 6, we show the results of using the random forests method when applied to an "X-Pypeline" analysis.

Decision trees are very fast to train and evaluate and an ensemble of them can be run in parallel. For this reason, "X-Pypeline" utilizes scikit-learn's implementation of the random forest classifier. Although random forests are quick to train, especially when the number of trees and the max depth of each tree is small, it is unclear if one setting, i.e. one choice of the max number of metadata features to expose to each decision tree, the max depth of each decision tree, etc., of the random forest is optimally exploring the coherent and incoherent energy space. We could potentially loop over different settings for the random forest and then pick the settings that minimize the SSFEUL, similar to how the coherent cuts are tuned via the standard method. However, doing so belies the purpose of using random forests in the first place, that is, they are much faster than the standard method, but looping over different settings will limit that impact. In addition, efforts to tune the parameters too extensively, including allowing the max-depth to be a large number can lead to instances of over training. For this reason, we also explore the efficiency of more abstract machine learning algorithms to tackle the task of separating signal clusters from noise clusters. In the following section, we discuss how one dimensional

convolutional neural networks can also be used to model the function that optimally probes the coherent and incoherent energy space. One advantage is that CNN's converge on a given loss function, therefore, after a certain number of iterations some level of confidence can be had that the space has been explored thoroughly if the change in the loss function for every iteration of training changes minimally.

4.3.3 Using Convolutional Neural Networks to Tune Coherent Cuts

As was discussed in Chapter 3, CNNs are useful when trying to model functions. In Section 4.3.1, we motivated some functions that could reasonably slice the space between the coherent and incoherent energies. Therefore, it seems reasonable that a CNN could find a function $f(\theta)$ that can ideally explore the coherent and incoherent energy space. CNN's can be tricky to use in the case of the result the "X-Pypeline" algorithm is trying to provide. Although CNN's can have strong predictive power and find abstract dependencies between the features provided in the training set, they can also be subject to occasionally make very strong but *wrong* predictions. In the case of detecting gravitational waves and making upper limit statements, it is important that a handful or even a single background event not be predicted as a signal with a very high score, or very few of the clusters associated with the injections will be able to be claimed as a detection following the methodology laid out in the previous sections. More concretely, we are striving for an algorithm that has a low FAP. Again, we do not want a background cluster falsely claimed as a GWB. In addition, in order to claim a detection, the cluster from the on source data must have a detection statistic, i.e. probability of being class signal, larger than that of the detection statistic of the $(1-p) * N_{\text{trials}}$ loudest cluster from the background trials. This means that the algorithm cannot have many, if any, outlier classifications. This fact, that we do not want many outlier classifications, i.e. background triggers that are very confidently, but wrongly, predicted as signals, motivates our choice of loss function. There are a number of loss functions well suited to penalizing outlier classifications. Mean Squared Error, or L2 loss, is an excellent example and has the functional form,

$$P = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 \tag{4.64}$$

where $y_i = 0$ for background and $y_i = 1$ for signal and f is the classifier score. We note that P has a large value when y and $f(x_i)$ disagree. Also, the loss function *binary crossentropy* is well suited for both types of errors and has the functional form

$$P = -\frac{1}{N}\sum_{i=1}^{N} -(y_i \log(f((x_i)) + (1-y_i)\log(1-f((x_i))) \tag{4.65}$$

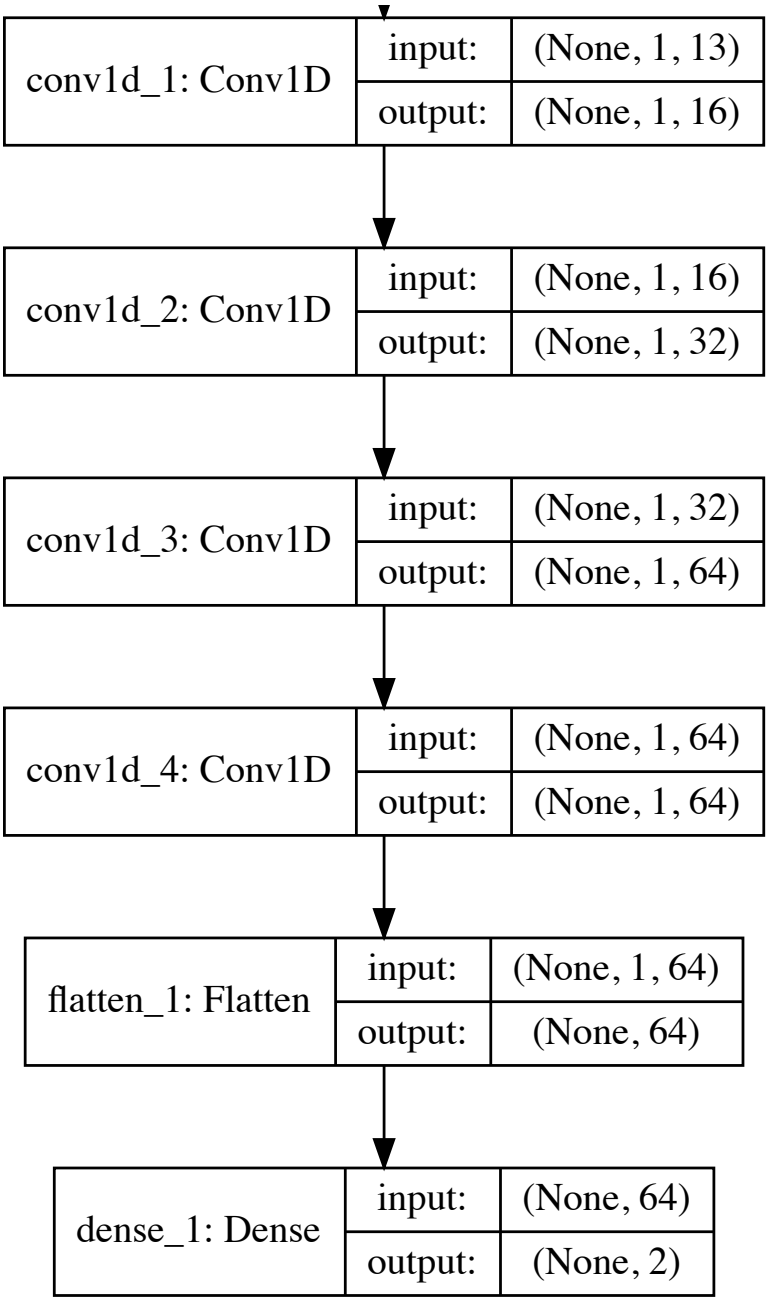


Figure 4.7: Illustration of the one dimensional convolutional neural network used to classify clusters from injection and background trials as either signal or noise. This selection is motivated by a desire to balance concerns of overfitting the data by providing too many tuneable parameters while limiting the chance the neural network cannot infer the desired relationship between the coherent and incoherent energies. Each convolutional layer utilizes the hyperbolic arc tangent activation function.

where the terms are the same as in the L2 loss and we again note that P has a large value when y and $f(x_i)$ disagree. We train models that are optimized by both types of loss functions and determine which performs better following the same methodology of SSFEUL laid out in Section 4.3.1.

Using the same statistics as features for every cluster, as we used in the case of Random Forests, each cluster only has 17 features. It makes sense to treat every feature independently, that is, to choose a kernel size of 1×1 for the first layer. How many convolutional layers are necessary for the problem is less clear. It appears that the relationship between the coherent and incoherent energy is, at best, mildly complex, but we also do not want to limit the models ability to make inferences. Therefore, we use a model with 4 one-dimensional convolutional layers each with an activation function of the hyperbolic arc tangent with filters of 16, 32, 64, and 64, respectively. Figure 4.7 visualizes this model. The selections for this model are driven by a desire to balance concerns of overfitting the data by providing too many tuneable parameters while limiting the chance the neural network cannot infer the desired relationship between the coherent and incoherent energies and the detection statistic.

On modern GPUs with standard libraries like Tensorflow [119], training a one dimensional CNN is extremely fast, almost as fast as the random forest. Moreover, the need to retrain multiple models with different combinations of number of convolutional layers and activation functions is limited by the fact that one can construct a robust enough model, i.e. has enough tuneable parameters, at the outset which can handle most data sets. Once trained, we follow the same methodology for making a detection statement as we used for random forests in Section 4.3.2. Again, in Chapter 6, we show the effects of using this CNN approach to tuning the results of an "X-Pypeline" analysis.

We have motivated how CNN's could probe the coherent and incoherent energy space and provided a way to make a detection statement with respect to GWBs. One drawback of all of the tuning methods described so far is that it would be nice to provide *uncertainties* associated with our predictions. A naive way to do this could be to redo the above methodologies selecting different training and testing sets and providing error bars on the fraction of recovered GWBs at a given amplitude. For the standard method, this would incur great computational cost, but could be plausibly executed with the random forest and convolutional neural network methods. There does exist, however, a machine learning classification tool that inherently comes with uncertainties in its predictions, the Gaussian process (GP). In the next section, we motivate the use of GP to classify our clusters as either signal or noise.

4.3.4 Using Gaussian Processes to Tune Coherent Cuts

To this point, we have looked for a function $f(\theta)$ that can map inputs $[x_1, .., x_n]$ to values of either 0 or 1, signal or noise. In our case x_i would be equivalent to an array of coherent, incoherent and detection statistics that are associated with every cluster. The sections 4.3.1, 4.3.2, and 4.3.3 presented approaches that attempted to solve this mapping. However, what if we wanted a way to quantify uncertainty in this mapping, that is, how confident are we that the $f(\theta)$ found through the above methods is the only mapping consistent with the data? This is where we can leverage the power of the GP. We will briefly explain GPs and motivate their use to solve this problem, but please see [120, 121] for more information on GPs and their derivation.

GP is non-parameteric in that it finds a distribution over a possible set of functions $f(\theta)$ that are consistent with the observed data, not simply $f(\theta)$. In this way, we must define a prior assumption about all of the possible $f(\theta)$, and then using a set of observed data, i.e. a training set, update this prior assumption and arrive at a posterior distribution. The critical question is how does one define a prior over a set of functions? It turns out it is sufficient to define a prior on the output of the function $f(\theta)$. That is, a GP assumes that the outputs of the function are jointly Gaussian. Specifically, that $p\left(f_\theta(x_i), ..., f_\theta(x_M)\right)$ is jointly Gaussian with some mean and covariance matrix, i.e. can be modeled by a multivariate Gaussian distribution. We have touched on multivariate Gaussian distributions in 4.2.5. In this way, we can model the joint distribution of points x_a and x_b as

$$p\begin{pmatrix} x_a \\ x_b \end{pmatrix} \propto N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}\right). \tag{4.66}$$

The elements of the covariance matrix are determined by a covariance kernel, K, such that $K_{ab} := K(x_a, x_b) = Cov\left(f(x_a), f(x_b)\right)$. In most cases, for data points x_a and x_b that are similar or close together, the kernel is chosen such that the there is a larger covariance for these points. In this way, GPs constrain the function to be smooth over the length scale set by the kernel. The key motivation for this selection is that if the kernel function deems x_a and x_b to be similar, then we anticipate that the output of the function $f(\theta)$ to be similar. For example, if x_a and x_b both have properties consistent with a GWB, say a large value in the detection statistic and a larger coherent plus energy than incoherent, then the mapping of the function should be similar for both points, i.e. both should map to a signal. More generally, the kernel determines the preferred functional behavior of realizations drawn from the GP. Each realization of the GP is therefore a sequence of correlated random variables, but all such functions nonetheless share some general features.

There are a number of commonly used kernels in GP, including the *squared exponential* kernel, also known as the radial basis function (RBF) kernel, and the

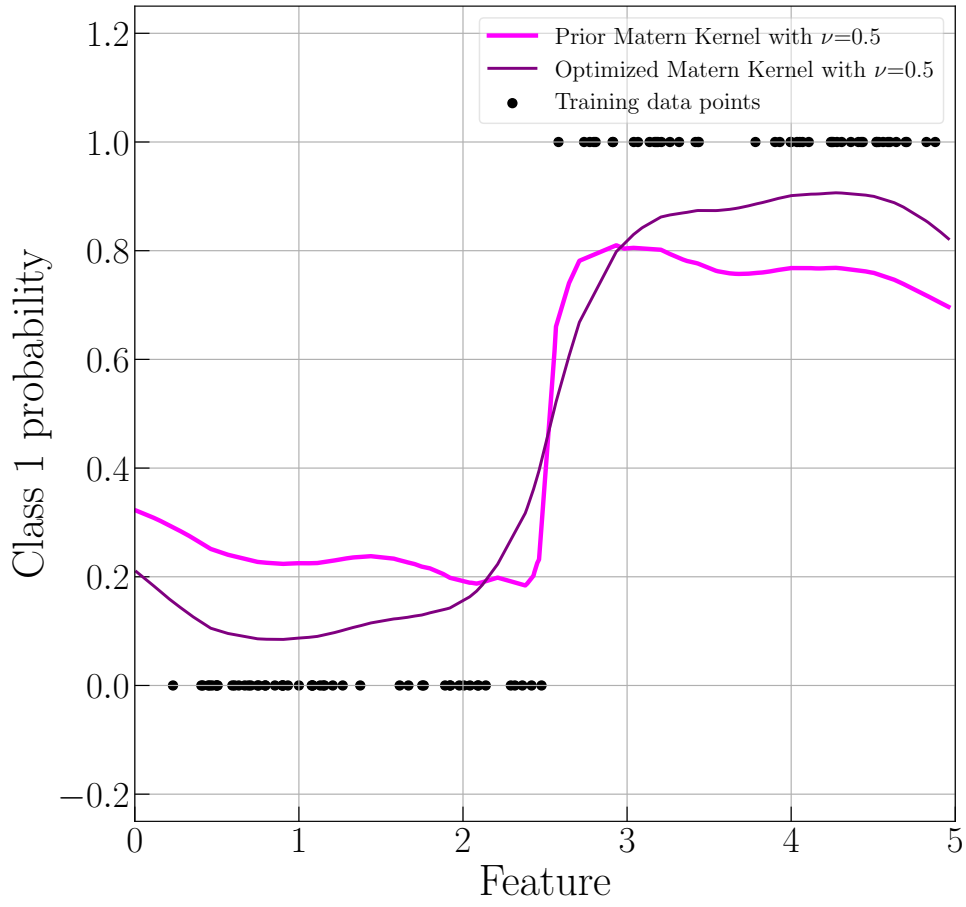


Figure 4.8: Simple example of the use of the Matern kernel with a value of $\nu = \frac{1}{2}$ applied to an example of points $[x_i, .., x_M]$ where $x_i \in [0, 5]$ and $y_i = 0$ when $x_i < 2.5$ and $y_i = 1$ when $x_i > 2.5$. Although they are similar due to the simplicity of the example, the optimized GP (purple) provides a closer prediction of the real distribution of data points than the prior (magenta). Specifically, we can see here that after fitting to the training set, the values of the hyperparameters between the prior and the optimized kernel are different. Specifically, the hyperparameters of the prior were $1^2 * K(l = 1, \nu = 0.5)$ and $39.9^2 * K(l = 6.43, \nu = 0.5)$ were the optimized hyper parameters. As we fixed $\nu = \frac{1}{2}$ it makes sense that this did not change through optimization.

Matern kernel. The Matern kernel is a generalization of the RBF with an additional parameter ν which controls the smoothness of the resulting function. For a more complete list of kernels, see Section 4.2 of [120]. We present the form of the Matern kernel with $\nu = \frac{1}{2}$, which is equivalent to the *absolute exponential* kernel as this is what we will use in practice when training the GP.

$$K_{\text{matern}}(x_a, x_b) = \sigma^2 \exp\left(-\frac{(x_a - x_b)}{l}\right) \tag{4.67}$$

where l is the length scale parameter. Figure 4.8 shows what the prior and optimized version of this kernel would be with respect to a dummy set of points $x_i \in [0, 5]$ with a mapping to 0 or 1. Specifically, by prior, we mean a kernel that starts with a naive selection of hyperparameters of $\sigma = 1$ and $l = 1$ and then letting the GP optimize these hyper parameters based on the training data. Other kernel options could be explored but this kernel is able to capture the variations in our data and the use of other kernels is left to future work.

From 4.66 it is clear that we can get the conditional probability of x_a given x_b or vice versa. In fact, for all functions f_i, we can extract the conditional probability

$$P(f_i|x_i, f_j*, x_j*) = \frac{P(f_i, f_j*|x_i, x_j*)}{P(f_j*|x_j*)} \tag{4.68}$$

where $*$ values represent all data and non-starred values our training data. In this way, we can derive the posterior distribution using observed values for $[x_i, .., x_m]$. Moreover, since the prediction is probabilistic, we can model the uncertainty associated with every prediction. This information is valuable when trying to understand what values of x_i elicit the least confident mapping to signal or noise. To state again, because the prediction from the GP is Gaussian, we can compute empirical confidence intervals and decide whether additional tuning is needed in some region of interest. For example, we may want to look closer at injection clusters that receive a classification of being a signal but with relatively high uncertainty.

To this point, we have brushed one aspect of GP under the rug, that we are using it to do a classification task. We have been discussing the use of GP, so far, in the context of Gaussian process *regression*. That is, we have been assuming that the joint distribution in the space of x can be modelled as Gaussian. In the case of classification, however, the likelihood function is clearly not Gaussian, i.e. in the case of binary classification, it is a mapping to either 0 or 1. The likelihood we are dealing with is actually a Binomial likelihood. For this reason, Gaussian process *classification* (GPC) is left to approximate the actual likelihood. As the name would suggest, GPC utilizes a Gaussian approximation of the actual likelihood through a Laplacian approximation. We will not go into the details of this approximation here but [120] provides an excellent explanation.

We have motivated the ability of GP to provide labels with uncertainty, which

can be useful when understanding how confident we are that a given cluster is signal or noise. Although Gaussian process has proved a powerful and useful tool to make predictions with uncertainties, they are challenging to scale to large data sets. This should be clear from the above as we are not just calculating the joint distribution between x_a and x_b, but x_1 through x_M. These lead to very large covariance matrices that most be approximated by the kernel function for every data point. Specifically, the cost of inference for GPs is $O(N^3)$ where N is the number of data points in the training set [122]. In the case of X-Pypeline, the training set of clusters from background and injections is routinely, at a minimum, about 100000. This means that the standard tools for doing Gaussian process are unusable and either we cannot use GPC to classify "X-Pypeline" clusters or we must find a different solution.

To help resolve the issue described above, we introduce the concept of the sparse variational Gaussian process (SVGP) [122]. Instead of learning the posterior from all of the training set points x_1 through x_M, SVGP leverages *inducing* points to serve as a sort of proxy for all of the data points. As long as the inducing points are selected such that they are representative of the whole data set, then a.) the computational cost of finding the underlying posterior becomes manageable, and b.) will be similar to the posterior of the GP on the whole data set. For many real world tasks, the selecting of these inducing points can be difficult and this is where the variational technique can be beneficial. In the case of "X-Pypeline", however, we can smartly select a sampling of points and do not need to rely on the algorithm to select these points on our behalf. Specifically, we select a sampling of points that make sure to represent the three types of clusters in our data, that is, clusters which are associated with glitches, Gaussian noise, and injections. In a standard analysis, when processing chunks of 256 seconds of data for each trial, an "X-Pypeline" analysis keeps around 62 clusters with the largest value in the detection statistic from a given trial. If we select as inducing points, the maximum and minimum value from those 62 clusters, this will likely capture clusters associated with both glitches and Gaussian noise. For our injection trials, we select the cluster with the largest value in the detection statistic, over a sampling of waveform families and amplitudes. This selection criteria will allow "X-Pypeline" to still use GPC to classify clusters as it limits the computational cost of marginalizing over the joint distribution but also utilizes a sampling of data points representative of the whole training set.

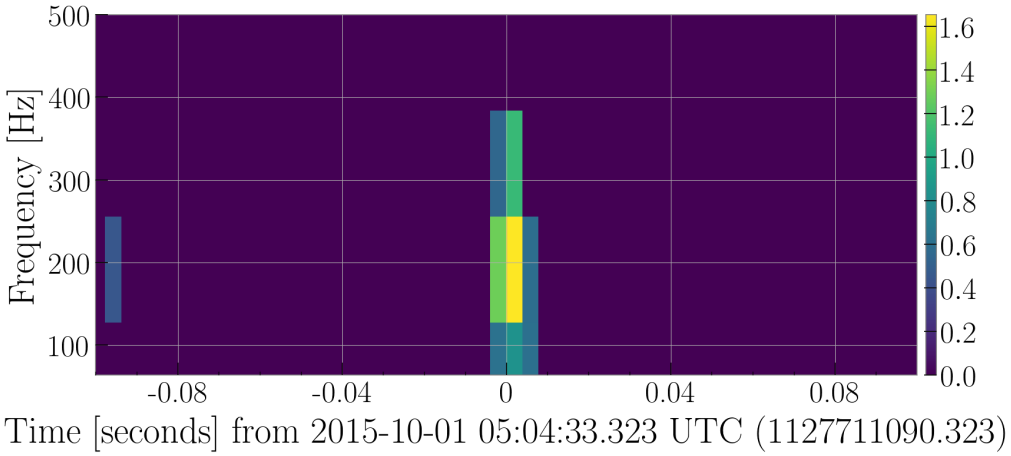
In order to implement SVGP, we utilize the open-source Python library GPflow [123]. GPflow uses Tensorflow as a backend to enable GPU support to speed up the optimization of the SVGP. For our GPC, we model whether a given cluster is a signal or noise event using the Bernoulli likelihood (a special case of the Binomial distribution where a single trial is conducted) and, as mentioned earlier, use the Matern kernel with a value of $\nu = \frac{1}{2}$ to model the covariance. Once trained on the training set of clusters, the question becomes how best to leverage the predictive power of GPC

to make statements on upper limits and detections. As a baseline, it is possible to follow a similar method to 4.3.2 and 4.3.3 and label all the clusters from every background trial with a probability of being signal and take the max probability from each trial. Then following the same implementation of FAP as before, we would select the $(1-p) * N_{\text{trials}}$ percent largest value from the list of these clusters and compare the probability of every injection cluster of being a signal to this value. In the future, under a different selection of likelihood (because the Bernoulli distribution is defined by its mean and the estimate of the variance provided by the GP gives no additional information), it may be possible to leverage the uncertainty that is associated with every prediction from the GPC to do something more. For now, and to ease the comparison between the machine learning methods, we will use this method. In Chapter 6, we show the effects of using this GP approach to tuning the results of an "X-Pypeline" analysis and show how it effective it is when compared to all the methods outlined above.

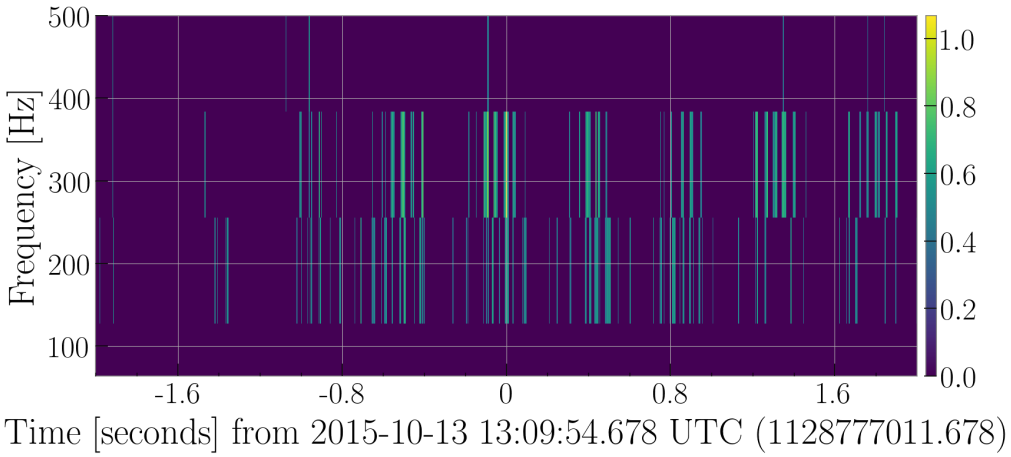
At this point, we have motivated the use of different machine learning algorithms to perform the task of probabilistically mapping clusters $[x_1, .., x_i]$ to 0 or 1, signal or noise. By selecting a range of different methods for modeling this mapping, we will be able to see which one performs the best for a given realization of clusters. In fact, for every run of "X-Pypeline", we can use every method described above and compare the results because all can be performed quickly due to either the inherent speed of the method, i.e. random forests, or the ability of modern hardware and software, i.e. GPUs and Tensorflow, to implement methods such as GPs and CNNs. A critical aspect of every machine learning technique that we have not studied closely yet, except in the case of selecting inducing points to train the SVGP, is the importance of selecting a diverse training set of clusters when training the machine learning algorithm. Moreover, as discussed at length, glitches in the detector data streams are the biggest hindrance to the algorithms sensitivity to GWBs. Therefore, it would be prudent to make sure that the training set of clusters from the background contain not only loud amplitude glitches, but those with a diverse set of time-frequency morphologies. It is in this step of post-processing, the creation of the training set of clusters from the background, that we can incorporate all of the classification efforts described in Chapters 2 and 3.

4.4 Incorporating Gravity Spy

An important aspect of using any machine learning algorithm is to make sure that a balanced training set is used when training the algorithm. As discussed at length, because gravitational wave data does not only contain stationary coloured Gaussian noise, it is likely that many of the background clusters will be associated with glitches and will contain large values in one or more of the detection statistics. We have proposed a number of methods that could use a training set of clusters from



(a) Blip glitch



(b) Scratchy glitch

Figure 4.9: The time-frequency representation of two glitches as they would appear in "X-Pypeline" using a STFT with a duration of $\frac{1}{128}s$, with color representing the total energy, see Equation 4.55, of the pixel. Similarly to Figure 2.1, we again show an example of the Blip glitch (a) and the Scratchy glitch (b). In terms of coherent and incoherent energies, it will depend on what is happening in the other detector, which should typically be Gaussian noise or, in any case, should be uncorrelated. Therefore, these glitches should average to zero correlation ($I = E$) if one sums over enough pixels. Therefore, the Scratchy glitch will typically be closer to $I = E$, while the Blip glitch might have random fluctuations such that to $I \neq E$ due to small-number statistics.

injection and background trials to probe the space of coherent and incoherent energy statistics to discern the clusters with properties consistent with GWBs and those consistent with glitches and background noise. It is conceivable that different families of glitches may lead to different optimal coherent and incoherent cuts. For instance the Scratchy glitch, identified with its repeated notches in time-frequency space, could elicit different coherent and incoherent energies than say the Blip glitch, identified by its characteristic pulse in time-frequency space. Having an even distribution of Blip glitches in the testing and training sets could be especially important in training optimal coherent cuts to remove troublesome glitches because Blip glitches have a similar time-frequency shape to GWBs. Therefore, it is not enough to make sure the training set has clusters associated with loud amplitude glitches, which could be identified through an algorithm such as Omicron [49], but also a diverse set of time-frequency morphologies. Figure 4.9 shows an example of the sparse time-frequency map of two different glitches, the Blip glitch and the Scratchy glitch. This map was made with a FFT of duration $\frac{1}{128}s$, and as is standard practice for an "X-Pypeline" analysis, only the top 1 percent loudest pixels are shown from this map. It is clear that both glitches appear in a sparse "X-Pypeline" time-frequency map, but have very different time-freqeuncy morphologies. Therefore, we can use Gravity Spy to ensure that the training and testing sets used in "X-Pypeline" not only contain glitches, but also a roughly equal distribution of families of glitches.

Moreover, in cases where a background cluster is loud in one of the "X-Pypeline" detection statistics, but the duration of the cluster does not overlap with a classified Gravity Spy glitch, tools such as the similarity search tool can be used on all the background clusters to find other similar clusters. In this way, we can also utilize the similarity search model in creating the training set of "X-Pypeline" clusters to make sure that the training and testing sets are representative of each other and the background. It is anticipated that taking the extra effort to make similar training and testing sets will limit the possibility that one of the machine learning methods described above becomes over-fitted and performs poorly on the testing set. In Chapter 6, we will use Gravity Spy to assist in the creation of the training set used for the RF post-processing method, demonstrating the impact on the efficiency of the RF method when the training set does not include information of glitch families.

4.5 Conclusion

In this section, we have presented the underlying methods of the "X-Pypeline" algorithm. This included a discussion how we use STFTs to project gravitational wave data into the form of time-frequency pixels. We then described how we can use projection vectors to create statistics associated with every pixel aimed at distinguishing GWBs from noise and glitches. Identifying nearby pixels in time-frequency space and summing over these statistical properties, we create clusters of pixels. To

determine exactly what statistical values determine if the cluster is likely a GWB or not, we repeat the same analysis on chunks of data containing simulated GWBs and data known not to contain GWBs. We presented four different methods, using discrete coherent cuts, using random forests, using convolutional neutral networks and using Gaussian processes, for exploring the space of these statistical properties and mapping these clusters to signal or noise and to make statements on upper limits. Each method involved creating training and testing sets of background and injection clusters. To this end, we conjectured how results from Gravity Spy could be useful in the creation of this data by ensuring that both sets receive even distribution of background triggers that are associated with glitches with differing time-frequency morphologies. In the Chapter 6, we compare the effectiveness of the different methods when creating upper limit statements when doing a search for a GWB associated with a Galactic supernova event. Before we can get to those results, it is important to go into more details about Galactic supernovae. This includes how we will be alerted to the event, which will effect the duration of the on-source window used when searching the gravitational wave data, what is the latest in Supernova physics, and what is the latest in three-dimensional supernova simulations, both of which will impact the types of simulated GWBs we use in our injections. These are the subject of the next Chapter.

Chapter 5

A Galactic Core-Collapse Supernova

In the introduction to this thesis, we discussed the potential of a Galactic supernova (GS) occurring. The rate of GS is uncertain and estimates can range widely, but estimates roughly average a maximum of 1 every 40 $\pm$10 years [124]. The close distance of a GS will enable the scientific community to gather information through many different messengers. Neutrinos are one such messenger and will provide critical information on the time of arrival of the gravitational wave. The community in charge of the initial alert of a GS based on the detection of neutrinos is the SuperNova Early Warning System (SNEWS) [125]. The current members of the SNEWS experiment include Super Kamiokande (Super-K) [126], the Large Volume Detector (LVD) [127], IceCube [128], Borexino [129], Daya Bay [130], KamLAND [131], and the Sudbury Neutrino Observatory (SNO) [132]. The majority of these detectors were designed for other purposes than strictly the detection of GS neutrinos, but all are capable of serving this purpose. In addition, we can anticipate an electromagnetic (EM) counterpart as the GS will be very optically bright in the sky. This means that a GS presents a rare opportunity to search gravitational wave data with an exact sky location and time stamp.

In their own right, studying the neutrino and EM data will provide insight into a GS, but neither can directly probe the inner workings of the core-collapse and subsequent explosion of the star. A gravitational wave signal from a core-collapse supernova (CCSN) would provide unprecedented information about how massive stars explode. After the initial shock wave that occurs after core bounce stalls at a radius of 150km [133] due to matter falling in on the core, it is unclear what reignites this shock wave that will eventually cause the star to explode. The prevailing theory is neutrino driven heating or convection. The neutrino driven theory proposes that after core bounce some of the neutrinos emitted from the proto-neutron star (PNS) are reabsorbed which causes rapid heating and revitalizes the shock wave. There are other proposed explosion mechanisms including the magnetohydrodynamic mecha-

nism (MHD) [134, 135] which would only occur in the case of rapidly rotating CCSN.

It is important to understand the two classes of CCSN, in order to distinguish the characteristics of the GW associated with a given CCSN. CCSN can be divided into two types, depending upon whether the core is rapidly rotating or not. In this case, we understand rapid rotation to correspond to an initial angular velocity of $\omega \gtrsim 2 - 3\frac{\text{rad}}{\text{s}}$ [136]. In this case, the GW emission is dominated by the signal from the non-axisymmetric collapse and bounce of the core, and is directly related to the speed of the initial rotation as well as the frequency of the fundamental quadruple mode of the PNS [137]. Essentially, the faster the initial rotation, the stronger the signal, and the denser the PNS, i.e. a softer nuclear equation of state (EOS), the stronger the signal and the higher the frequency of the emission. The rapidly rotating progenitors are expected in only a small fraction of all CCSN, on the order of one percent, and therefore it is important to also analyze GW emission from non-rapidly rotating CCSN. In the non-rapidly rotating core collapse scenario, the core-bounce does not elicit a strong GW emission and instead strong GW emission comes in the post core bounce stage through hydrodynamical instabilities. These instabilities include prompt convective heating of the PNS through the re-absorption of neutrinos or standing accretion shock instabilities (SASI). Each set of initial conditions, and subsequent post-core bounce events will lead to a dramatically different GW signature [136, 138, 139].

As mentioned before, it is critical to have waveforms that span the range of possible time-frequency morphologies, i.e. shapes in the time-frequency map, when performing injections in order to properly tune the "X-Pypeline" analysis and to provide meaningful upper limit statements. Previous work shows that standard sine-Gaussian waveform templates like those used in GRB searches are not sufficient for the tuning of an analysis pipeline when searching for a GW from a supernova [140]. To this end, we discuss the latest in three-dimensional supernova simulations as the ability to perform 3-dimensional simulations has dramatically improved in recent years. We note that the prominent use of recent 3-D simulations in Chapter 6 distinguishes this study of supernovae waveform detectablility from the studies done using the MATLAB algorithm. In [110], the predominant simulations available were 2-D and very few 3-D simulations were available. In addition, a number of the waveforms resulted from simulations that did not actually lead to an explosion. Therefore, we utilize gravitational waves from the latest CCSN simulations.

In this chapter, we discuss how SNEWS works, discuss how different initial conditions of the star can lead supernova to produce different gravitational waves and what information a detection of a GW from a supernova would provide about the pre-explosion inner workings of the CCSN. In Section 5.1, we describe how SNEWS works, and why neutrinos provide ideal external triggers for GWB searches. In Section 5.2, we discuss how supernovae create GWs and discuss the current state of 3D supernova simulations.

5.1 Neutrino Detection of a Galactic Supernova

In this section, we briefly introduce the method for detecting neutrinos from a GS, why neutrinos provide an excellent timestamp for the arrival time of the GW, and how the alert is distributed to the astrophysical community. For more information, please see [125].

The detection of neutrinos has two components. First, the actual detection of a sufficient number of neutrinos to assure the source is a GS, and second, any information on the source direction obtainable by the detection. Neutrinos undergo neutrino-electron scattering and the detectors can measure if the energy of the recoil electrons, T_{act}, in the detector exceeds some experimental threshold T_{exp}, which is different for each detector. For an example of how this happens in practice, we summarize the process used by one of the neutrino detectors, Super-K. Super-K analyzes data in chunks of 120 seconds. It searches for clusters of neutrinos, in time windows varying from half a second to ten seconds. If pre-defined thresholds are exceeded in any of these clustering windows, then noise reduction algorithms are applied to these clusters. After this noise reduction, the candidate clusters undergo stricter thresholds. For clusters that surpass this secondary threshold, the mean time between clusters is calculated. The motivation for this calculation is that clusters with small mean times are unlikely to be astrophysical in nature. When 100 cluster events have sufficiently large mean times and pass both thresholds, then staff members on site are alerted of a possible supernova candidate. Like GWB searches, the threshold is set at some false alarm rate (FAR) to ensure that the candidate event is not simply a "loud" background event. Also like the GWB search, it is difficult to confirm a detection with a single detector. Therein lies the motivation for SNEWS, which would confirm a detection through coincident neutrino detections among some combination of detectors. This has the opportunity to reduce the FAR to less than 1 per century, which would be critical given the rare nature and importance of a GS.

Neutrino detectors can achieve pointing information one of two ways. First, detectors can utilize the scattering angle of the neutrinos to extract the direction from which they came. Second, a group of neutrino detectors, like those in SNEWS, can employ a technique called trianglization. This takes the timing of neutrino detection from several detectors and attempts to triangulate the source. Both of these techniques, especially the latter, present a number of challenges which are discussed in [141]. Nonetheless, any sky location information would be valuable not only to optical astronomers, but also in assisting the GWB search. We discussed at length the importance of knowing the sky location for an "X-Pypeline" analysis in 4.2. Regardless, any sky location information would be unavailable with the first SNEWS alert. Therefore, it is necessary to prepare a GW search which would utilize no sky location information.

While the sky location information may be poor, the timestamp provided by SNEWS is both accurate and valuable for a GWB search. CCSN most likely emit GWBs either during the collapse or during the explosion over the next few seconds following the collapse. Recent simulations show that the time of the bounce and the maximum amplitude of the GWB from a CCSN have a strong correlation [115]. Simulations also show that the time of the bounce is correlated with the onset of neutrino luminosity [115]. The question is, within what margin of error can the time of the first neutrino detection from the CCSN bound the time of the core bounce, and consequently, the time of the largest GWB. In a study by Pagliaroli et al, [115], the timing of the first neutrino detection and the core bounce is on the order of milliseconds. This means we can comfortably place a window of plus or minus 2 minutes around the neutrino alert and know that it would contain the GWB from the core bounce. The reason for the larger than necessary window is that other processes before and after the core bounce may also create GWBs [136, 138, 139, 142–144].

To understand the impact of this narrow time window on the search, it is useful to compare with the time windows provided for more distant CCSN. Electromagnetic observations provide these time windows. These observations, however, typically only localize the time of GW signal to a 10 - 100 hour on-source region. The main causes of this large window is observation cadence. That is, when a SN is observed, the window can only be constrained to the last time that portion of the sky was observed by a telescope. There are two key advantages to this small time window. First, the background, i.e. detector noise, around the event is easier to understand. In the case of a neutrino trigger, the analysis employs the same 3 hour background estimation that the GRB search utilizes. This 3 hour background helps ensure that the detectors are operating under similar sensitivity levels for both off-source and on-source trials. More importantly, this 3 hour background means there are fewer background noise events then for 10+ hours. The fewer background events means lower amplitudes of GWs can be detected. Third, the computational cost involved in a search of 4 minutes of on source data and 3 hours of background data with the goal of a 3-sigma detection is far less than a search with 10-100 hours of on source data. Follow-up analysis, if shown necessary, will be easier to conduct than follow-up analysis to EM triggered CCSN.

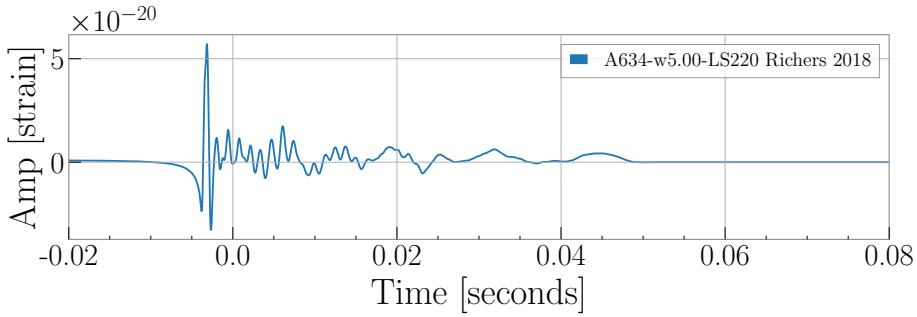
SNEWS will publicize two types of alerts, GOLD and SILVER. A SILVER alert requires manual on-site follow-up of the candidate event. On the other hand, a GOLD alert will be automatically sent to the astronomical community at large with the following information; the UTC time of the coincidence, all detectors involved in the coincidence and the type of alert. There are three criteria for a GOLD alert, and all criteria are in order to avoid a false alert. First, there must be at least 2 coincident detections within 10 seconds from two physically separated experiments. Second, at least two of the alarms from the individual detectors are tagged as GOLD. Finally, the detectors signifying a GOLD alert must have had consistent rates of GOLD

detection candidates leading up to the coincident detection. A rate of GOLD alerts that is considered consistent is on the order of 1 per week. In the instance of a GOLD alert, an online GW event catalog system called GraceDB[1] will receive and parse the information. This information will be used to set-up and automatically launch an X-Pipeline analysis. If more information such as sky location is available in a later alert, then a follow-up analysis of the event will be performed.

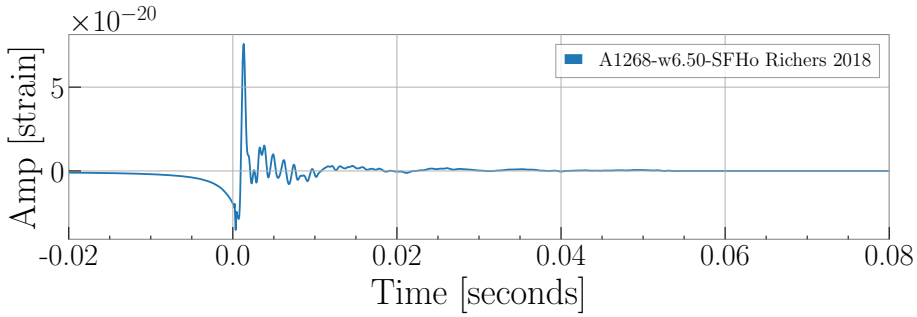
5.2 Discussion of Gravitational Waves from Core-Collapse Supernovae

In this section, we focus on the neutrino driven CCSN explosion mechanism, the two classes of CCSN, rapidly rotating and non-rapidly rotating, and the type of gravitational wave emission that is expected from each based on the most recent CCSN theory. Moreover, we discuss the state of numerical relativity (NR) simulations for this mechanism in both situations. Again, the selection of a diverse and comprehensive set of waveforms is important in order to accurately benchmark the ability of "X-Pypeline" to detect a GW from a GS. Depending on the assumed physics, such as the nuclear equation of state (EOS) of the PNS and neutrino transport schemes, the GW signal morphologies of a CCSN are broader than a simple sine-Gaussian can capture. Therefore, NR simulations of the CCSN and the resulting gravitational wave provide much better approximations of the true time-frequency morphology. Unlike compact binary coalescence signals, however, it is impossible to robustly predict the signal's detailed time series because the CCSN explosion is necessarily complex not unlike the merger and ringdown of a CBC signal. Even if the CCSN explosion mechanism and whether the star was rapidly rotating or not was known and understood beforehand, this issue would still persist on account of this stochasticity. This fact is the key to understanding why we still use an unmodelled search instead of a templated search when searching for a GW from a SN even though we have (some) waveforms from NR. Moreover, using waveforms from NR simulations will allow for meaningful upper limits to be placed on the strength of GWs from CCSN and could possibly allow us to rule out a theorized explosion mechanism in the event of a non-detection. In Section 5.2.1, we discuss what it means for the core of a star to be rapidly rotating and what features characterize the GW emission from a rapidly rotating CCSN (RRCCSN). In Section 5.2.2, we discuss the much more prevalent CCSN with a non-rapidly rotating core and what features of the post-bounce, pre-explosion phase, including SASI and convection, characterize the GW emission from this scenario. In Section 5.2.3, we describe the state of NR simulations for each and choose waveforms to use in our analysis.

[1] https://gracedb.ligo.org/



(a) Initial Rotation 5 radians per second LS220 EOS



(b) Initial Rotation 6.5 radians per second SFH_0 EOS

Figure 5.1: Two examples of GW emission from rapidly rotating CCSN. Here we present a case of "mildly" rapidly rotating progenitor with a stiffer EOS and a slightly more rapidly rotating progenitor with a softer assumed EOS [145]. As is expected, the amplitude of the GW emitted from the more rapidly rotating progenitor and softer EOS is larger than that of the less rapidly rotating progenitor. Both amplitudes are scaled to a distance of one kiloparsec.

5.2.1 Rapidly Rotating Core-Collapse Supernovae

The GW emission from a rapidly rotating CCSN is dominated by the GW signal from the pressure-dominated core bounce [145, 146]. As the iron core collapses the central density of the core becomes greater than the nuclear saturation density, ρ_{nuc}. At this point, the collapsing core "bounces" due to the the stiffening of the EOS [136]. At this moment, the non-spherically symmetric oscillation of the core causes the peak GW emission from this progenitor. The subsequent damping oscillations of the core cause smaller peaks in the GW emission. Exactly how non-spherically symmetric these oscillations are, how fast the core is rotating, and what the correct EOS of the PNS is, all dictate the amplitude of the resulting peaks in the GW emission. A study from Richers et.al [145] performed simulations of the core-bounce and resulting GW emission over a number of combinations of initial rotations and EOSs to analyze both the impact on the strength of the GW signal and whether a given combination eventually leads to an explosion. We plot what the waveform looks like from two of

these combinations in Figure 5.1. As can be seen, the higher initial rotation value and softer EOS both lead to a larger amplitude signal. However, the study found that the strength of the GW was predominantly tied to the ratio of rotational to gravitational energy and independent of EOS, but the frequency of the GW from the post-bounce oscillations was tied to the EOS. It is anticipated that the frequency of the GW emission associated with the core-bounce will be

$$f_{GW} \propto \frac{1}{T_{\rm dyn}} \simeq O(10^2 \sim 10^3)\text{Hz} \tag{5.1}$$

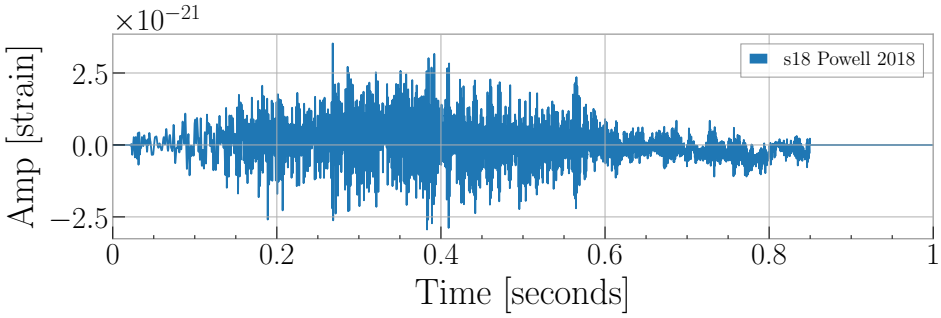
where $T_{\rm dyn}$ is the dynamical timescale at core-bounce and is directly proportional to the EOS of the PNS [136]. Again, a softer EOS means the core can achieve a higher density, and therefore a shorter $T_{\rm dyn}$ which leads to GW emission at higher frequencies.

If there are non-axisymmetric rotational instabilities associated with the PNS of a rapidly rotating progenitor, then a strong post-bounce emission can be expected [136]. In [146], it is shown that even when the ratio of the rotational to gravitational potential energy is on the order of 0.1, the PNS becomes dynamically unstable which leads to a significant GW emmission. However, 3D simulations of the post-bounce phase of the CCSN are still lacking. These simulation could probe the non-axisymmmetric properties of the PNS and how the oscillations of the PNS inter play with, for example, neutrino heating [136].

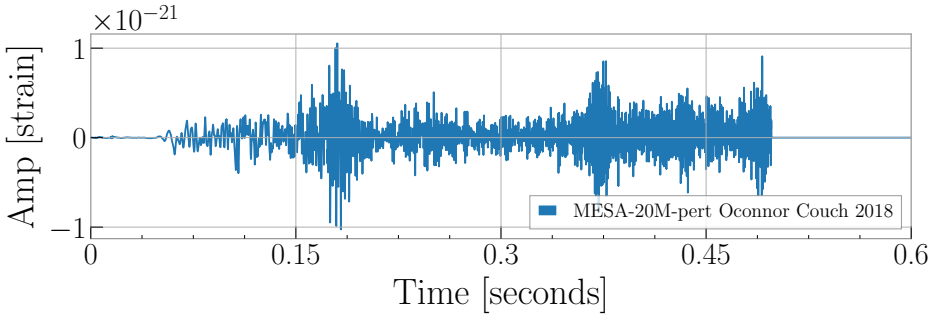
With respect to neutrinos, anisotropic neutrino radiation emitted from the PNS after core-bounce can also provide a GW emission [136, 138, 147]. The strength of the emission is directly related to the deviation of the neutrino radiation from spherical symmetry. This emission is expected to be at frequencies lower than 100 Hz and could possibly be below aLIGO and Virgo's sensitivity band. As mentioned earlier, rapidly rotating CCSN only accounts for roughly 1 percent of CCSN, but are still significant as they are expected to emit a large amplitude GW. Therefore, we still utilize NR simulations of the GW emission expected from this progenitor when performing a GW search to follow up a GS. In Chapter 6, we consider simulations of rapidly rotating CCSN from Scheidegger (3D with magnetic fields) [146], Richers (2D with different combinations of EOS and initial rotations) [145] and Dimmelmeier (2D) [148].

5.2.2 Non-Rapidly Rotating Core-Collapse Supernovae

In the case of a non-rapidly rotating CCSN, the core is either not or negligibly rotating and so the GW signature from the bounce is also negligible. Therefore, the expectation is that hydrodynamic instabilities associated with PNS in the post-bounce, pre-explosion stage of the SN are the most emitters of GWs. These instabilities include neutrino driven convection and the SASI. One type of convection that is expected to lead to GW emission from a non-rapidly rotating CCSN is re-



(a) Powell 2018



(b) Couch 2018

Figure 5.2: Two examples of GW emission from non-rapidly rotating CCSN. The three-dimensional Powell waveform is dominated by convection occurring at the surface of the PNS and does not expect a strong impact from SASI [137]. The three-dimensional Couch model, however, shows a significant contribution of the GW emission from SASI [144]. Both amplitudes are scaled to a distance of one kiloparsec.

ferred to as *prompt*. As the initial shock wave from core-bounce begins to stall a negative entropy gradient forms behind the shock wave which causes oscillations of the PNS. In general, the emission from this physical phenomena is weak, except in cases where the core is slightly rotating, as seen in model $M13 - SFH_o -$ rotating from [143]. The reason for this is in line with the results shown for the rapidly rotating CCSN. Essentially, a softer EOS combined with a relativity high rotational to gravitational potential energy would mean that oscillations of the PNS would emit stronger signals. After this so called prompt convection, further heating from the PNS and neutrinos lead to the so called "postshock convection" and is associated with a strong GW emission from g-modes of the PNS. Finally, in some situations the material in the postshock region begin to undergo violent motions which can also lead to a stronger and higher frequency GW emission. Essentially, the stalled shock, which is beginning to become re-energized through convection, starts to oscillate due to enhanced turbulent motions caused by SASI. In fact, a major difference between three-dimensional and two-dimensional simulations is the effect of SASI in the pre-explosion phase of the CCSN, and thus, its GW emmision. Non-axisymmetric treatments of CCSN tend to see a stronger GW emission from SASI, and [149] found that a softer EOS state, which can lend itself to a more compact core, also leads to stronger SASI activity.

However, not all progenitors are expected to have strong emission due to SASI. Neither model from [137] undergo strong GW emission due to SASI. This is due, in large part, to the initial conditions of the star. One is a 3.5 solar mass helium star which was evolved to an ultra-stripped star, i.e. went through a common envelope with another star, with a pre-collapse core mass of 2.6 solar masses and the other a larger 18 solar mass zero-age-main-sequence progenitor with a helium core mass of 5.3 solar masses. In the case of the former, the core mass is too low to expect strong SASI activity, in the latter, there is essentially too much pressure being applied to the shock from the outside to have large shock oscillation driven by SASI. Therefore, it is possible that the detection of a GW from a GS could provide strong evidence in favor of a stiffer or softer EOS depending on the observed GW emission from SASI as well as the properities of the progenitor star. Figure 5.2 shows examples of GW emission from non-rapidly rotating CCSN with and without strong contributions from SASI.

To summarize, in the case of the non-rapidly rotating CCSN, in the pre-explosion phase we are looking for any turbulent motions that cause non-spherically symmetric oscillations, either of the PNS or the shock. The more turbulent the hydrodynamics, the more optimistic one can be about the strength of GW emission. Outstanding work improving the speed of these hydrodynamical simulations and updating the physics, such as neutrino transport schemes, varying EOS assumptions, and initial conditions of the progenitor, has been on-going in recent years. Therefore, in the following section, we note some of this work, and select a combination GW waveforms

Waveform Type	Ref.	Model Name	$h_{\rm rss}$ [10^{-21}@1 kpc]	$f_{\rm char}$ [Hz]	Polarizations [$10^{-8} M_{\odot} c^2$]
Rotating CC	[148]	Dim-s15A2O05ls	1.623×10^{1}	461	+
Rapidly Rotating CC	[145]	A1268-w6.50-SFHo	1.257×10^{2}	395	+
Rapidly Rotating CC	[145]	A634-w5.00-LS220	9.635×10^{1}	560	+
Rapidly Rotating CC	[146]	Sch-R1E1CA-L	3.443×10^{-1}	518	+, ×
Rapidly Rotating CC	[146]	Sch-R3E1AC-L	8.202	571	+, ×
3D Convection	[137]	Powell-He3.5	2.371	144	+, ×
3D Convection	[137]	Powell-s18	5.636	175	+, ×
3D Convection	[143]	M13-SFHo-rotating	4.825	663	+
3D Convection	[143]	Morozova-M10LS220	3.0468	970	+
3D Convection	[144]	MESA-20M-pert	1.686	978	+, ×

Table 5.1: From left to right it provides the physical mechanism, literature reference, model name, the root sum square gravitational wave strain ($h_{\rm rss}$), the characteristic frequency of gravitational wave emission ($f_{\rm peak}$), and polarizations for the numerical waveform injections utilized in Chapter 6.

from 2D and 3D simulations of both CCSN scenarios.

5.2.3 Numerical Relativity Simulations

Recent work has led to a flurry of new numerical relativity simulations from both 2D (axisymmetric) and 3D (non-axisymmetric) models of rapidly and non-rapidly rotating CCSN [133, 137, 138, 144, 145, 147, 149]. In addition, a wide range of EOS, initial conditions of the progenitor, and neutrino transport schemes have been utilized in this recent work, leading to some of the most detailed GW emission from the pre and post explosion phase of a CCSN. Recent work from Powell et. al [137] has even made GW waveforms available from 3D simulations of successful explosions which model the GW emission well into the explosion phase of the CCSN. It is for this reason that we select for use in our analysis, results from these most recent simulations.

Due to the observed differences between GW emission in 2D and 3D simulations as well as GW emission in models where the presence of shock oscillations due to SASI are present or not present, we take a broad sampling of waveforms from the literature. It is expected that the methods described in Chapter 4 should be suited to capture the burst like properties of the GW emission expected from the core-bounce of rapidly rotating progenitors, and also the spikes in GW emission from non-rapidly rotating progenitors from either convection or shock oscillations from SASI. Table 5.1 summarizes useful information about the waveforms we have selected and Figure 5.3 shows what some of these waveforms look like. In Chapter 6, we present results from both 2D and 3D simulations as well as the two types of progenitors, rapidly rotating and non-rapidly rotating.

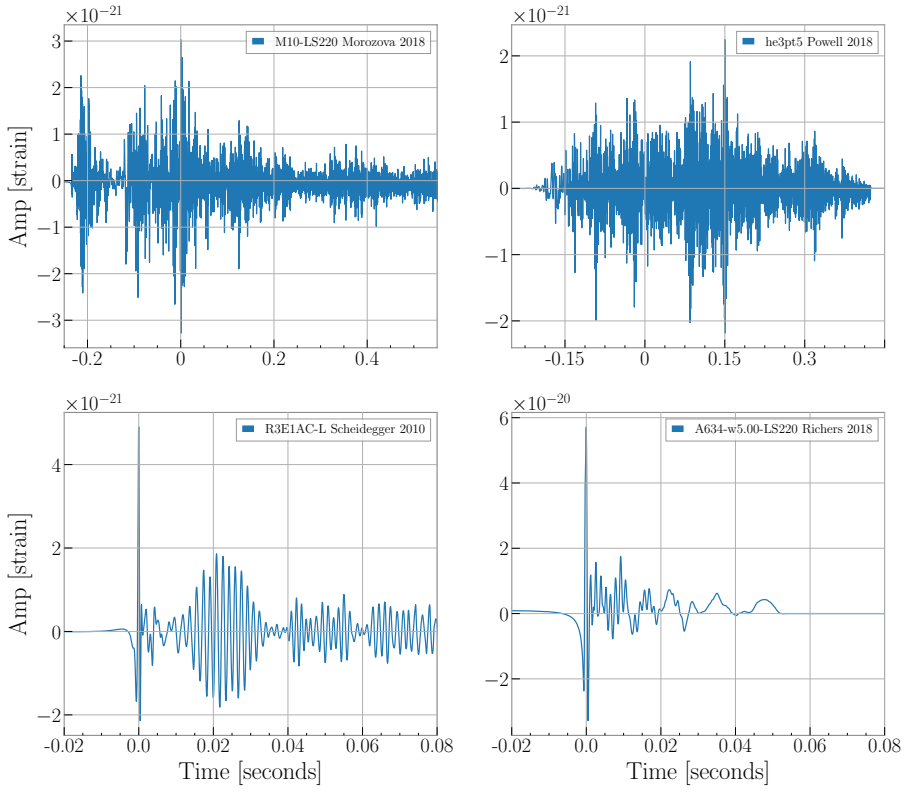


Figure 5.3: A selection of the SN waveforms used in Chapter 6. Top row: 2-D and 3-D examples of GW signals from numerical simulations of non-rotating core-collapse progenitors. Bottom row: 2-D and 3-D examples of GW signals from numerical simulations of rapidly rotating core-collapse progenitors Top-Left: Waveform from a 2D simulation of the neutrino driven convection explosion mechanism from Morozova et al; [143]. Top-Right: Waveform from a 3D simulation of neutrino driven convection explosion mechanism from Powell et al; [137]. Bottom-Left: Waveform from a 3D simulation from Scheidegger; [146]. Bottom-Right: Waveform from a 2D simulation of a rotating CCSN with an assumed EOS of LS220 and an initial rotation of 5 $\frac{\text{rad}}{\text{s}}$ from Richers et al., [145]. All amplitudes are scaled to a distance of one kiloparsec.

Chapter 6

Results

In this section, we present results of applying the "X-Pypeline" algorithm to a search for GWs associated with a GS. We utilize injections based on GW waveforms from recent 2D and 3D simulations of rapidly and non-rapidly rotating CCSN. We discuss the impact of applying the tuning methods laid out in Section 4.3 to our candidate clusters when making upper limit statements. In addition, we discuss the impact of using Gravity Spy results to enhance the selection of the clusters which are used in the training and testing stages of the tuning methods.

An important aspect of the search for a GW from a GS is how the search must first be conducted in response to an alert from a neutrino trigger. As mentioned in 5, the neutrino trigger will provide an excellent timestamp for when to search the GW data. Although triangulation efforts will take place to provide additional sky location information, nonetheless this will likely be unavailable with the first alert or contain very large error bars. Moreover, electromagnetic information, which will provide an exact sky location, will not come for another $\sim 8 - 24$ hours later. Therefore, the automated "X-Pypeline" analysis in response to the neutrino trigger must be performed with no sky location information. This means we need to search all possible time delays in order to perform the coherent analysis laid out in Chapter 4. We first present in Section 6.1 the selection criteria that will be used to select the time-delays that "X-Pypeline" will use in this situation.

Additional analyses that utilize sky location information from electromagnetic data will also be performed but in a higher latency. In this thesis, we focus on results that come from this type of analysis. In order to mimic what this search might look like in the future and best apply the methods discussed in 4, we perform a search that uses data streams from two detectors from LIGO's first observing run (O1). The reason we use this data to benchmark the metholodogy is that the Gravity Spy results are well quantified for this period. In addition, we do not want to use data that simulates future detector sensitivity because this data would not contain glitches and would be unlikely to highlight any of the methods from Section 4.3.

This chapter is laid out as follows. In Section 6.1, we present the logic behind

selecting the time delays to use when performing a search for a GW from a GS when only the timing is known from the neutrino trigger. In Section 6.2, we describe the data used to benchmark the "X-Pypeline" algorithm, including how much coincident data we use, how many background trials are performed, and how many and what types of glitches are present in the data. In addition, we describe the parameters used for the analysis including the STFT lengths, the sky location, and the number of injections. In Section 6.3, we present the upper limit or efficiency curves that result from running the "X-Pypeline" algorithm and tuning the candidate clusters with the proposed tuning methods. We show results from all methods when selecting training sets with and without Gravity Spy results in mind. Finally, in Section 6.4, we discuss some of the implications of the different tuning methods as well as future work that could be done to improve the efficiency of the analysis.

6.1 SNEWS Triggered All-Sky vs. Optically Triggered Search

In this section, the difference between a SNEWS search with only timing information, called an all-sky search, and one with timing and sky location information is described. The difference between these two searches is that a neutrino trigger gives no sky position information whereas an EM counterpart would contain the precise sky position of the SN. The motivation behind the setup for the all-sky search is the same as one where the sky location is known a priori. In the optical case, the time delay between the detectors is known from the sky location and thus the search can coherently combine the individual detector data streams without potential loss of signal. In the all-sky case, there is an infinite range of possible time delays for a given detector configuration. Therefore, in order to ensure the entire recovery of the signal, it would be necessary to search over all possible sky locations. This analysis is, however, computationally inefficient and the analysis with "X-Pypeline" would take too long. The key to constructing the all-sky search is creating a dense enough grid of sky locations that will sufficiently cover the full range of possible time delays. In [150], by J. Aasi et al. it is shown that using a set of sky locations which map to equally spaced time delays was sufficient to detect GWBs in GRB searches with sky areas of several hundred degrees. Concretely, for a 2 detector case the search space on sky is equivalent to a 1D search over time delay. In this chapter, we extend the idea to an all-sky search and evaluate its effectiveness. Specifically, the grid must be dense enough so one or more of the grid points will recover at least 90% of the signal to noise (SNR) ratio of the signal. In order to accomplish this, we chose a time delay between grid points of 0.1 milliseconds. To see why, it is best to imagine the worst case scenario true sky location for this configuration which would occur if the true sky location was halfway between two of the sky locations in the grid. To

see how 90 percent of the SNR would still be recovered, we assume the true signal is a sine-Gaussian. A sine-Gaussian waveform coming from this location would be maximally out of phase of a sine-Gaussian coming from either of the two grid points around it. The timing error of this sine-Gaussian would be 0.05 milliseconds. The SNR loss from this phase shift can be approximated as

$$\mathrm{SNR}^2 = \cos(2\pi f_{\mathrm{signal}} t_{\mathrm{err}}) \times SNR^2_{\mathrm{max}} \tag{6.1}$$

where t_{err} is the timing error in seconds and SNR^2_{max} is the SNR from a sine-Gaussian at one of the two nearby grid points. In this analysis, the search looks for signals up to 2 kHz. Therefore, if this extreme situation is considered, that is $f_{\mathrm{signal}} = 2000$ Hz and $t_{\mathrm{err}} = 0.00005$ seconds, then

$$\begin{aligned} SNR^2 &= \cos(2\pi f_{\mathrm{signal}} t_{\mathrm{err}}) \times SNR^2_{\mathrm{max}} \\ SNR^2 &= \cos(0.2\pi) \times SNR^2_{\mathrm{max}} = 0.81 \times SNR^2_{\mathrm{max}} \\ SNR &= 0.9 \times SNR_{\mathrm{max}}. \end{aligned} \tag{6.2}$$

For this reason, we propose to search over time delays offset by 0.1 milliseconds. In practice, this will lead to a slower analysis of the data, but "X-Pypeline" is configured such that it can easily parallelize processing different time delays across many processors. In this way, given enough resources "X-Pypeline" can perform this type of analysis almost as quickly as if only processing a single sky location. Although we do not present upper limits results from an analysis of the GW data by "X-Pypeline" under these conditions, it is generally shown that a search in this configuration is about 10% less sensitive than a search of a single sky location [110, 150].

6.2 Analysis Set Up

For benchmarking "X-Pypeline", we select data from LIGO's first observing run, O1. During O1, the detectors in Hanford, Washington and Livingston, Louisiana were operational and were sensitive to a GW signal from a Binary Neutron Star (BNS) merger at a distance of 80 Mpc and 60 Mpc, respectively [12]. As mentioned earlier, the motivation for using data from this period is the Gravity Spy results during this period of time are well understood. By this, we mean that the non-Gaussian artefacts in O1 have been identified and classified by the Gravity Spy system described 2.

Moreover, this data has been made available to the public which makes efforts to replicate this analysis easier. Specifically, we select data between 2015-10-09 22:32:07 and 2015-10-10 05:55:44 UTC. The amplitude spectrum of each detector during this time period is plotted in Figure 6.2. In LIGO's second observing run, the detectors were sensitive to a GW signal from a Binary Neutron Star (BNS) merger

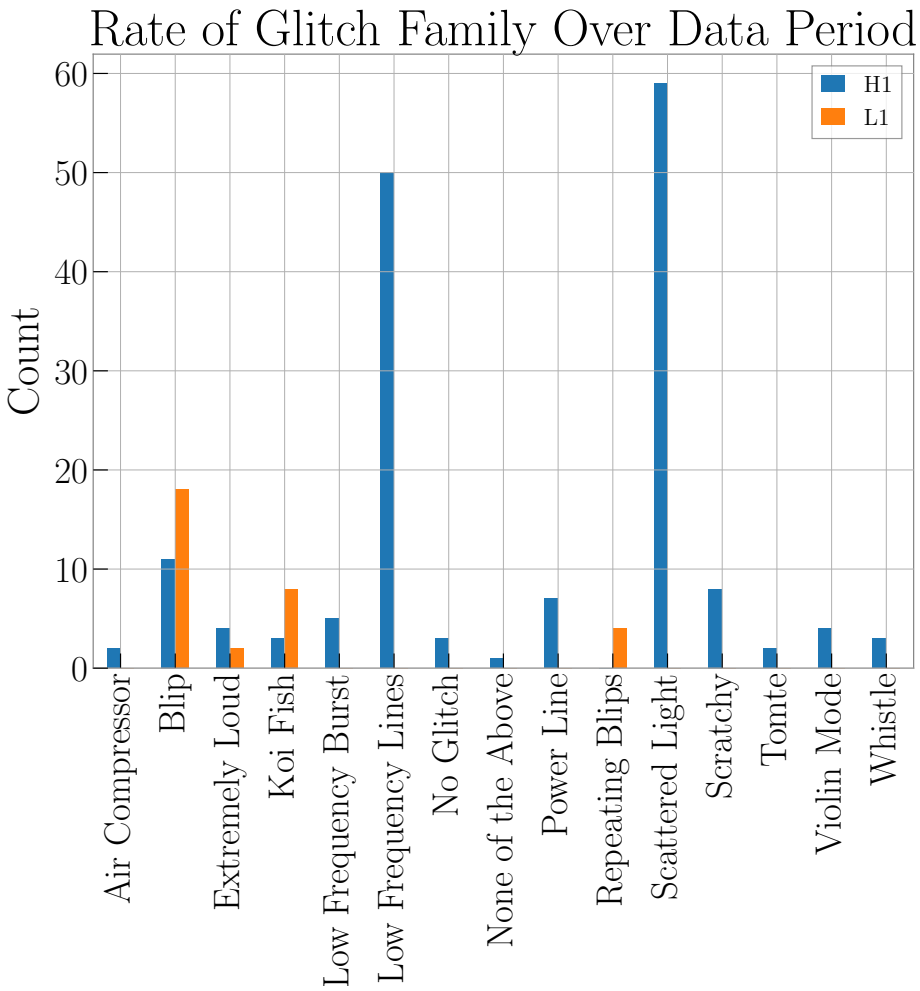


Figure 6.1: Number of glitches in Hanford (blue) and Livingston (orange) identified as being from one of the Gravity Spy classes during the data period analyzed. The Scattered Light and Low Frequency Line glitches are the most prevalent during this period and both occur exclusively at Hanford. The next most prominent glitches are the Blip and Koi fish glitches which happen at both detectors but slightly more frequently at Livingston. It was important to have a number of Blip and Koi Fish glitches during the time period analyzed because they are the most GWB signal like glitches that occur in the LIGO data streams. Therefore, any attempts to understand the impact of including Gravity Spy results in the proposed tuning methods need the background to contain these glitches.

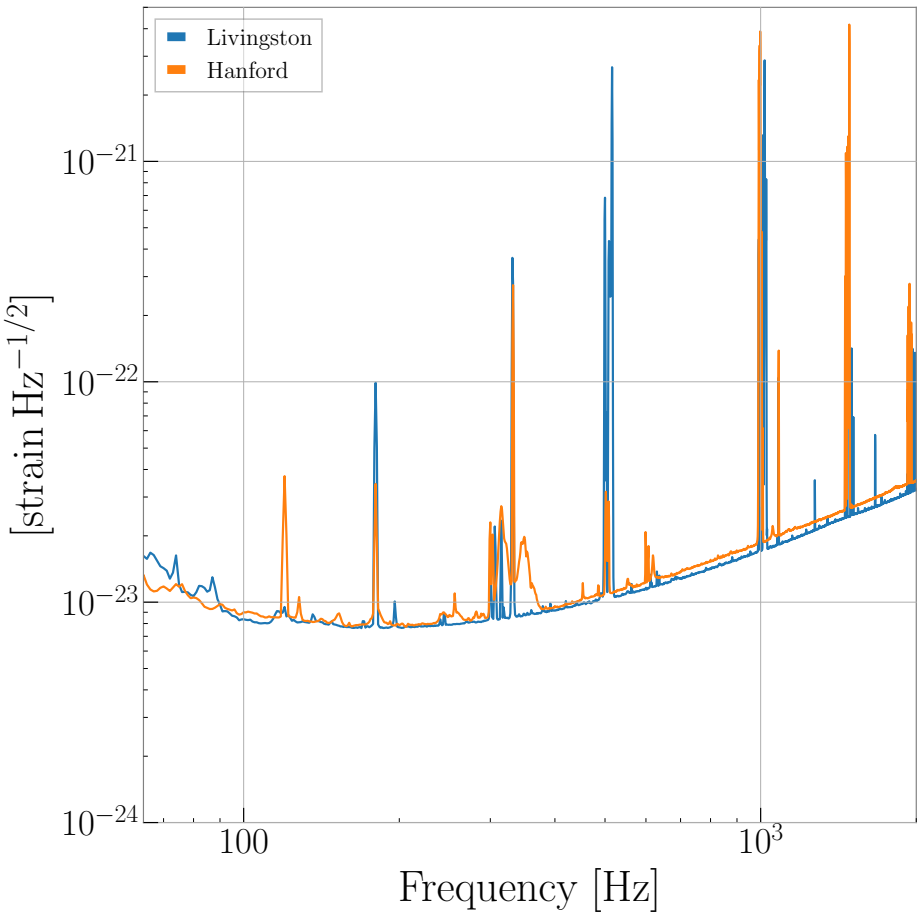


Figure 6.2: The amplitude spectrum of the Hanford (orange) and Livingston (blue) detectors during the period of time analyzed. The frequency range has been restricted to that of interest to the "X-Pypeline". The excess noise at 300 Hz is a well known feature of the Hanford data caused by the increased coupling of input "jitter" noise from the laser table into the interferometer [12]. This data was cleaned using a method laid out in [151]. In the analysis we utilize this clean data, but we wanted to highlight the sensitivity of the detector before this cleaning was applied.

at a distance of 80 Mpc and 100 Mpc, respectively [12]. In LIGO's ongoing third observing run, the Livingston detector is sensitive to this signal out to 135 Mpc and Hanford out to 110 Mpc. We can use the current sensitivity of the LIGO detectors to contextualize the results of this section. In general, the ability of "X-Pypeline" to "see" a GW from a CCSN at a given distance in O1 data can be roughly scaled proportionally to the increased sensitivity of the current detectors.

The seven hour time window used in this search is representative of the amount of GW data surrounding the neutrino trigger that would need to be searched. In order to make statements with high confidence, i.e. low FAP, we need to repeat the algorithm performed on the on source data many times on data not containing a GW. However, as mentioned earlier, the data we repeat the analysis on should be as representative as possible of the data during the on source window, otherwise the coherent consistency checks may not be optimally tuned. Therefore, it is important that we select data from the detectors for background analysis that is close in time to the on-source window. The amount of background data selected allows us to perform 20000 background trials. In order analyze 20000, 256 seconds blocks of data using only 8 hours of data, we must time slide the Hanford and Livingston data from different periods within this window to generate more time segments. We refer to this method as externally time sliding the data. Moreover, we can shift the data from one of the detectors within each of these 256 second chunks by some number of seconds (say 3 seconds) to generate even more time segments to analyze. We refer to this method as internally time sliding the data. Concretely, in this analysis, we produce the number of background trials used by the following combinations of internal and external time slides,

$$2_{\text{external}} * \left(\frac{256}{3}\right)_{\text{internal}} * \frac{8 * 3600}{256} \approx 20000_{\text{trials}}. \tag{6.3}$$

Having 20000 trials enable us to make upper limit and detection statements up to a FAP of 0.01%, i.e. this would mean selecting the loudest event (based on the detection statistic) out of the 10000 trials as our threshold. We again note that half of the background trials are used for tuning the analysis and the other half are used for statements about detection efficiency.

During this interval of time, we can also quantify the number of Gravity Spy labelled glitches that occurred. Figure 6.1 displays the number of Gravity Spy classified glitches. As can be seen from the figure, the predominant glitch during this time was Scattering and Low Frequency Lines both of which occurred exclusively at the Hanford site. Outside of these two, the most prevalent glitches were a number of Blip, Koi Fish, and Scratchy glitches. We anticipate that the properties of the clusters associated with the Blip and Koi Fish family of glitches to be the most similar to those of the clusters that are associated with the injections. Therefore, in order to test that building training sets with knowledge of Gravity Spy labels can be

Trigger Time	Right Ascension	Declination
Saturday October 10 2015, 02:00:00 UTC	307.65°	45.72°

Interferometer	Ant. Resp. × (Eq. 4.4)	Ant. Resp. + (Eq. 4.3)	Magnitude
H1	0.688322	-0.689871	0.974530
L1	-0.709405	0.642243	0.956939

Table 6.1: Information about the detector network sensitivity to the sky location used in the search at the time of the dummy neutrino trigger. The magnitude of the detector response is given by $\sqrt{(|f^+|^2 + |f^\times|^2)}$. We can see that the network is fairly sensitive to the plus and cross polarization from this sky location, and therefore, it presents a good opportunity to benchmark the algorithm and the tuning methods. At the same time, we did not choose a sky location that maximized the network sensitivity at this trigger time as this presented too optimistic a scenario.

helpful to the tuning methods, we needed to make sure the data window overlapped with a number of these glitches. This being said, we did not want to select a time interval to benchmark the analysis with an inordinately high rate of glitches as this would not be representative of an average ~ 8 hour interval of coincidence detector data.

To conclude, we use 8 hours of coincident O1 data during a time when a number of Gravity Spy labelled glitches occurred. This data allows us to produce enough background to make upper limit and detection statements up to a FAP of 0.01% and to test the impact of including Gravity Spy information in the creation of the training sets used in the tuning methods proposed in Section 4.3. In the following section, we discuss the parameters used in the search and the results of the analysis and subsequent tuning when performed on this data.

6.3 Results

In this section, we lay out the results of applying the "X-Pypline" algorithm to the data period described above and tuning the resulting clusters using the methods laid out in Section 4.3. Moreover, we explore whether information from Gravity Spy can impact these tuning methods. First, however, we discuss the exact settings of the analysis that we perform including the dummy neutrino trigger time, the sky location, the STFT durations, the number of injections and the distance scaling of the injections.

We selected a nominal neutrino trigger time of Saturday October 10, 2015 at 02:00:00 UTC and a electromagnetic counterpart from a sky location with a right ascension of 307.65° and a declination of 45.72°. The network sensitivity at this time to a GW from this sky location is summarized in Table 6.1. As can be seen from the table the network sensitivity to the plus and cross polarization from this sky location

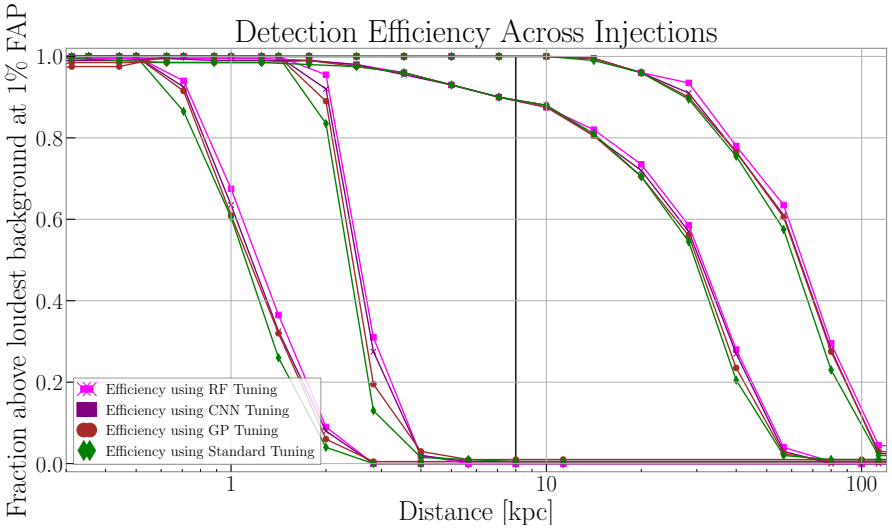


Figure 6.3: Efficiency curves of four waveform families utilizing the four proposed tuning methods. We demonstrate the results of the methods using two waveforms from non-rapidly rotating CCSN (Couch (far left set of curves) and Powell (second from the left)) and two waveforms from rapidly rotating CCSN (Richers (second from the right) and Scheiddeger (far right)), see Chapter 5 for more information on the waveforms. The upper limit statements are made at a FAP of 1%. The RF method (pink) appears performs the best compared to CNN (purple), GP (brown), and the standard tuning method (green). Since these curves were made using 200 injections, we cannot definitively rule out that all methods produce the same efficiency curves (using Poisson error bars).

is favorable. Thus we can expect that the proposed polarization consistency statistics posited in Section 4 will work well in a search for a GWB from this sky location. We did not choose a sky position to maximize the sensitivity, however, as this presented an overly optimistic scenario for benchmarking the analysis. Therefore, this time and sky location present a reasonable proxy for the network sensitivity in the case of a real GS.

The rest of the parameters of the analysis are as follows. We utilize STFT durations ranging from 1 second to $\frac{1}{128}$ seconds in steps of one over the next power of two. We perform 400 injections of every waveform, but we perform distance scalings depending on the GW progenitor. For GW signals from rapidly rotating CCSN, which are expected to be louder than their non-rapidly rotating counterparts, distance scaling ranges from 100 parsecs to 10 Mpc away. Conversely, for non-rapidly rotating CCSN, the amplitude scaling was from 10 parsecs to 1 Mpc away. These settings allow us to use 200 injections at each distance for tuning the analysis and the remaining 200 to make upper limit statements.

Figure 6.3 shows the efficiency of the "X-Pypeline" algorithm to detect a GW signal from four different CCSN simulations at a FAP of 1% using the methods described in Section 4.3. We plot the distance that "X-Pypeline" can "see" each signal, displaying two waveforms from simulations of rapidly rotating progenitors and

and two waveforms from non-rapidly rotating progenitors. As is expected, signals from rapidly rotating core collapse supernova can be detected much further away, well past the Galactic center (plotted with a vertical black line). The non-rapidly rotating progenitors, on the other hand, cannot be seen to the Galactic center, although it is likely with the current sensitivity of the LIGO detector and additional data streams that the Powell s18 waveform could be seen out to the Galactic center. As can also be seen from the figure, all tuning methods perform similarly, with the random forest method producing the best results across all waveforms. The second highest performing method is the CNN, followed by the GP method and the standard method. It is exciting to see that the faster performing machine learning methods are able to probe the incoherent and coherent energy space more efficiently than the standard method. Moreover, random forests are by far the least computationally expensive and fastest tuning method of the four. We note that the GP method fails to handle injections at unrealistically high amplitudes (i.e. unrealistically close), which we attribute to the selection of inducing points to train the model. We selected the loudest cluster from each background trial as inducing points which includes clusters which are associated with extremely loud glitches. Conversely, since we only select a handful of injections at each distance scaling, including the very loudest amplitude, as inducing points, it is likely that this part of parameter space is dominated by inducing points from the background and not injections. In the future, we propose that clusters from all injections at the highest amplitude scaling be used as inducing points to mitigate this effect. We also note that the similar performance across the tuning methods can provide support that additional exploring of the parameter space is unlikely to be needed for this analysis.

In Figure 6.4, we show the distribution of the probability of being the signal class for background clusters when using the three machine learning based tuning methods. As can be seen from the figure, the random forest method provides the "quietest" background values for a FAP of 1% (dashed green line) and 0.1% (dashed purple line). This might lead one to think that the random forest method should significantly outperform the GP or CNN based methods, but the CNN method also scores more of the injection clusters with a higher probability of being the signal class on average than the random forest. Thus, the two methods' efficiency at recovering injections remain similar despite the loudest background being "louder" for the CNN method compared to the RF. In the case of the CNN, tweaks to the model, i.e. changing the number of layers and activation functions, can elicit a quieter background distribution, but did not noticeably impact the efficiency curves at a given FAP.

With these results in mind, it seemed best to detail the impact of including Gravity Spy results when it came to constructing the training with respect to the RF method. Given the similar performance across the methods, we believe that any results from the RF can be expected to be shared by the CNN and GP methods.

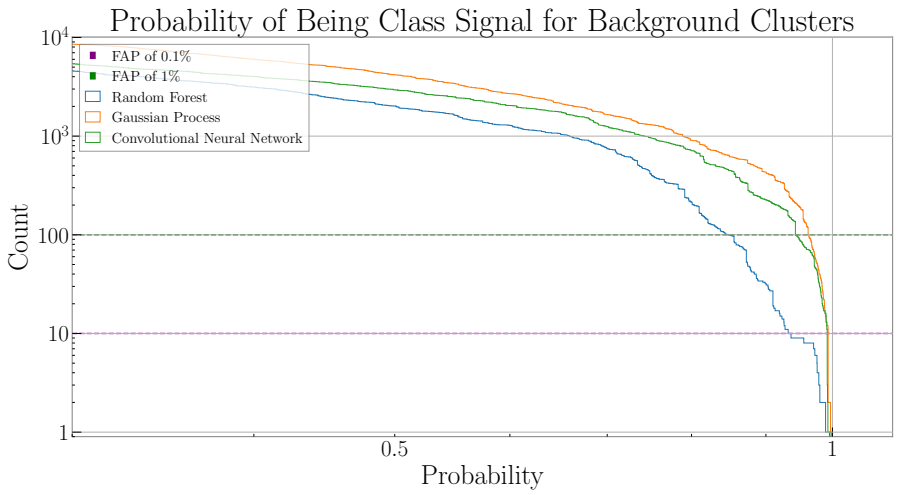


Figure 6.4: Cumulative distribution of the probability of being the signal class for testing set clusters from the background when applying the RF (blue), GP (orange) and CNN (solid green) models. That is, we show the number of background samples that received a given score between 0, class background, and 1, class signal from the given method. In addition, we indicate the score of the sample that would be used for making upper limit statements at the 1% (dashed green line) and 0.1% (dashed purple line) FAP by where the solid and dashed lines intersect. The RF has the quietest background (i.e. lowest score) at the 1% and 0.1% FAP. The CNN and GP background distribution are similar with a louder background, but based on the efficiency curves in Figure 6.3 also label the injection clusters with a very high (~ 1) probability meaning that the efficiency curves across the methods remain similar. In the case of the CNN, tweaks to the model, i.e. changing the number of layers and activation functions, can elicit a quieter background distribution, but did not noticeably impact the efficiency curves at a given FAP.

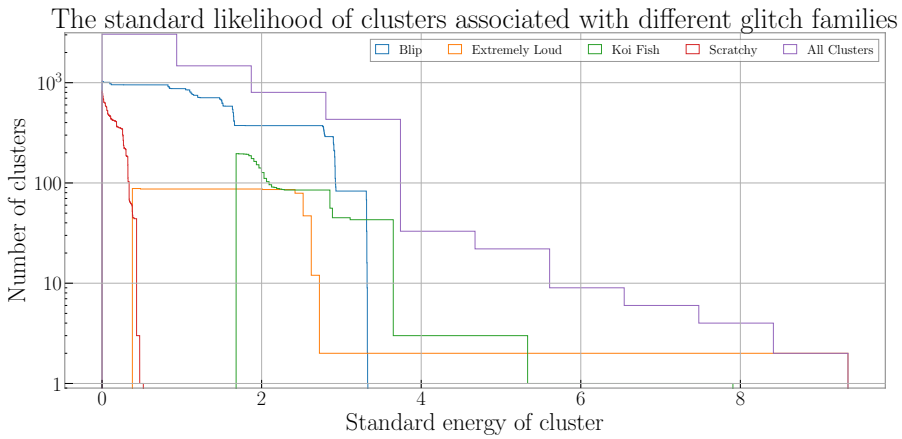


Figure 6.5: Cumulative distribution of the standard likelihood of clusters (Equation 4.37) from the background that are associated with a Blip (blue), Extremely Loud (orange), Koi Fish (green), or Scratchy (red) glitch and for all clusters (purple). Clusters from these glitches account for about 80% of all clusters with a standard likelihood value above 2, the expected value of the standard likelihood for Gaussian noise. When including clusters associated with any Gravity Spy labelled glitch, the percentage barely changes indicating that these glitches are the most signal like Gravity Spy glitches.

Figure 6.5 shows a cumulative histogram of the "X-Pypeline" standard likelihood for all background clusters and for background clusters associated with four types of Gravity Spy glitches, the Blip, Koi Fish, Extremely Loud and Scratchy glitches. The latter clusters account for 80% of clusters with a standard likelihood above 2. When considering glitches from all Gravity Spy families the percentage is still around 80% indicating that these glitches are the most signal like. It is clear from this histogram that these glitches form an important part of the background that hinders the analysis from making stronger the upper limit statements. Therefore, we perform four iterations of the RF analysis with different training set combinations to highlight the importance of using Gravity Spy results to build the training set.

First, we use the training set as constructed when training the original RF, GP and CNN models. Second, we remove all clusters from the above training set which overlap with any Gravity Spy labelled glitch. Third, we remove all clusters from the training set which are associated with the Blip, Koi Fish, Extremely Loud and Scratchy glitches. Finally, we remove all clusters from the training set associated with any non Blip, Koi Fish, Extremely Loud or Scratchy Gravity Spy labelled glitch. The effect on the background is detailed in Figure 6.6. As can be seen from this figure, a significant part of the RF method's ability to achieve a reliably "quiet" background stems from having background clusters from the Blip, Koi Fish, Extremely Loud and Scratchy glitches. The background distributions are nearly identical between having the entire training set and the training set that only retains clusters that are associated with the Blip, Koi Fish, Extremely Loud or Scratchy glitch and

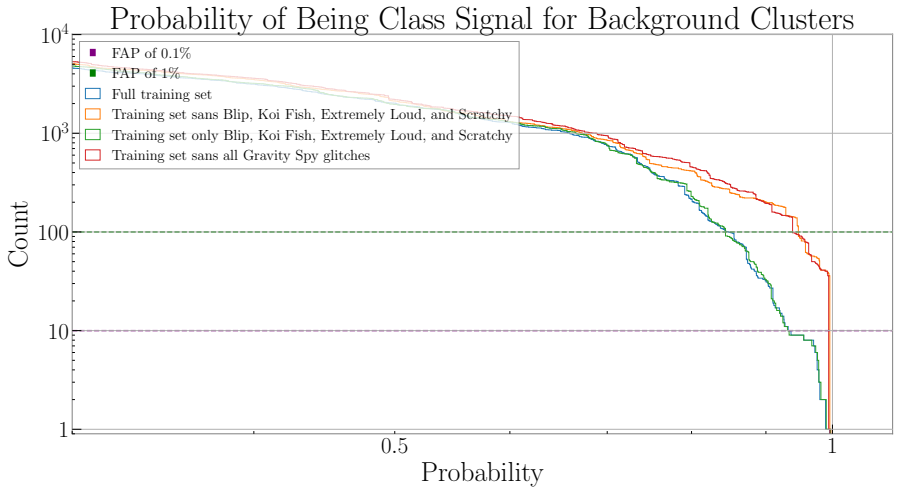


Figure 6.6: Distribution of the probability of being the signal class for testing set clusters from the background when training the random forest with four different types of training sets. Blue: All clusters from training set are used in training. Red: Training set where all clusters associated with Gravity Spy glitches are removed from the data used for training. Orange: Training set where only clusters associated with the Blip, Koi Fish, Extremely Loud, or Scratchy labelled Gravity Spy glitches are removed from the data that is used for training. Green: Training set where only clusters associated with the non Blip, Koi Fish, Extremely Loud, or Scratchy labelled Gravity Spy glitches are removed from the training set. As can be seen it is critical that the training set contain clusters associated with the Blip, Koi Fish, Extremely Loud, and Scratchy glitches or the value of the background at a given FAP will be significantly higher using the RF method.

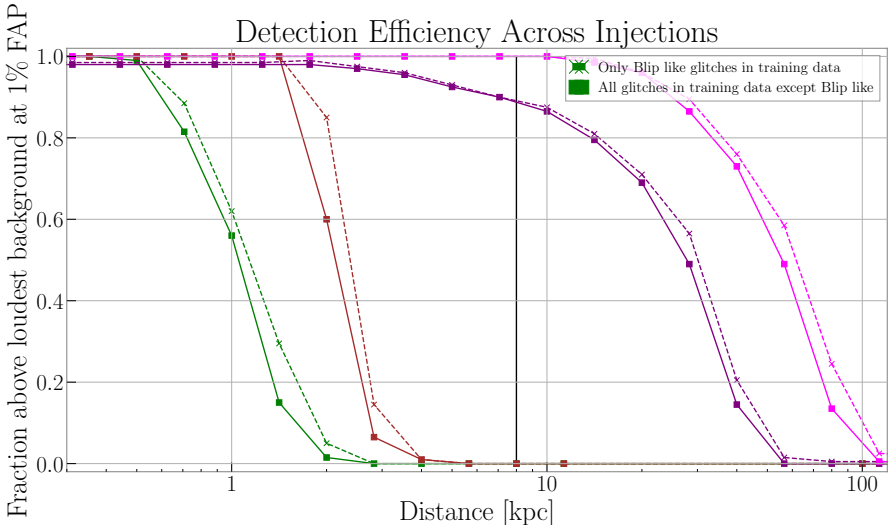


Figure 6.7: Efficiency curves of four waveform families utilizing the random forest method trained with all Gravity Spy glitches except for Blip, Koi Fish, Scratchy and Extremely Loud in the training set, and with only those families in the training set. We demonstrate the results of using the two training sets when applied to waveforms from non-rapidly rotating CCSN (Couch and Powell) and two waveforms from rapidly rotating CCSN (Richers and Scheiddeger), see Chapter 5 for more information on the waveforms. Across the different waveform families, there is a consistent improvement in the detectable volume of the search of 10 percent at a FAP of 1% when the random forest is trained with a training set containing clusters associated with the Blip family of glitches.

removes the clusters associated with the other Gravity Spy classes. Training the RF without clusters from these specific glitch families leads to a 10 percent loss in the 50% exclusion distance across waveform families, see Figure 6.7. Therefore, it is important not only to have glitches, but to have specific families of known LIGO glitches when building the training set in order to maximize the tuning method.

To conclude, we have benchmarked the use of the "X-Pypeline"algorithm to perform a coherent analysis of GW data in the event of a GS. We have used 7 hours of data from Hanford and Livingston from LIGO's first observing run. We tuned the results using four different methods in order upper limit statements, all of which performed similarly with the RF method performing the best. In this setting, rapidly rotating core-collapse signals were able to be detected well past the Galactic center whereas their non-rapidly rotating counterparts were unable to be seen to the Galactic center. However, with the current and future sensitivity of detectors it is likely that GW signals from non-rapidly rotating progenitors such as Powell s18 will be detectable to that distance. We also explored the impact of using information from Gravity Spy to build the training set used for training the RF. We show that if the Blip, Koi Fish, Extremely Loud and Scratchy glitches are not included in the training set then the method suffers a 10% reduction in distance at a given FAP. We anticipate that as the detector evolves, it is possible that new families of glitches

will also be vital to include in the training set used for tuning a given method, and that Gravity Spy will be able to provide assistance in assuring these new glitches make it into the training set as well.

6.4 Conclusions

We conclude that through the combination of speed and performance, the RF method provides the best way to tune the "X-Pypeline" analysis. Although the RF performs the best in this scenario, we still think that employing the CNN is valuable. For instance, the properties of the statistics in a three detector run may elicit a different efficacy of each method. Therefore, we anticipate future work will explore the impact of these tuning methods when performing a coherent analysis with more data streams, as well as when performing the analysis over more than a single sky location. In addition, although the GP elicits the most computational expensive and finicky of the methods, i.e. having to use the sparse variational method and having to pick good inducing points, nonetheless it also seems to provide the most possibilities. Future work with this method will include choices of inducing points and choices of likelihoods. A non-binary likelihood could allow for more information about troublesome background clusters to be gleaned by accounting for the variance in the label that the GP method provides. As it stands, a likelihood distribution, such as the Bernoulli likelihood, is entirely described by the mean of the distribution and the variance cannot be used to better understand which clusters are more challenging to label than others. Finally, we believe that the future of machine learning with respect to aiding LIGO GW data analysis includes performing analysis not only on the metadata associated with clusters of time-frequency pixels but directly on the GW time series data itself. There is extensive on-going work in trying to use machine learning in this way in both modeled searches, i.e. compact binary coalescence, and unmodelled searches [152–154].

Chapter 7

Conclusions

This thesis has explored the use of crowd-sourcing and machine learning to provide an adaptive and reliable program for classifying excess noise in aLIGO data. Moreover, it has also explored machine learning's ability to help aid in the discovery of new morphologically similar glitches that may arise over time as the LIGO instruments undergo upgrades. Specifically, we demonstrated how citizen scientists were able to discover two new glitch families previously unknown to LIGO scientists, the Helix and the Paired Doves glitches. Both of these glitches sources were discovered and resolved. Moreover, through the use of a web interface and the machine learning algorithm DIRECT, we demonstrated how users could utilize a single example of a morphologically novel glitch to rapidly discover more occurrences of the glitch in the Gravity Spy data set. This method can facilitate the rapid creation of large training sets which are needed to train the supervised algorithms used in Gravity Spy. Moreover this method can provide enough examples of the glitch so that LIGO scientists can find the cause of the glitch. This thesis also introduced the open-source Python algorithm "X-Pypeline" which implements a coherent analysis of GW data to detect unmodelled GWs. In introducing "X-Pypeline", we detailed a number of ways that machine learning algorithms and open-source libraries could help in the "tuning" stage of the analysis. These techniques included random forests, Gaussian process, and convolutional neural networks. We applied this algorithm and the proposed tuning methods to a GW search associated with a Galactic CCSN on data from LIGO's first observing run. We demonstrated the ability of all the methods to work with respect to making upper limit and detection statements but the RF performed the best. As all of the proposed tuning methods are supervised machine learning methods and rely on a quality training set to ensure optimal performance, we explored how Gravity Spy classifications could motivate the selection of the training and testing sets of candidate clusters. To this end, we demonstrate the impact on the efficiency of the random forest method when specific clusters associated with specific glitch families are not part of the training data. For a given FAP, the RF method loses 10% of its detection volume when the training does not have clus-

ters associated with the Blip, Koi fish, Extremely Loud and Scratchy glitch families. Therefore, it is important that the training set not only contain glitches, but specific families of glitches for the method to perform at its best.

Although we have explored ways that machine learning can aid in tasks associated with LIGO data analysis, most of the aid has come from applying machine learning to auxiliary GW data products. By this, we refer to data products that have undergone some sort of transformation, i.e. clusters of pixels from a time-frequency map in the case of "X-Pypeline" and PNG images of Q transforms in the case of gravity Spy. Both Gravity Spy and "X-Pypeline" could benefit from a speed improvement if it was demonstrated that whitened timeseries data was sufficient to achieve the analysis performance of both algorithms. For Gravity Spy, this means classifying morphologically similar excess noise by using only the timeseries, and for "X-Pypeline", performing an umodelled GW search using just the whitened timeseries data. Some efforts on this front have been taken on by various groups [152–154], but we look forward to the exciting prospect of utilizing machine learning to improve the speed and efficiency of both Gravity Spy and "X-Pypeline".

Bibliography

[1] Hans Ohanian and Remo Ruffini. *Gravitation and Spacetime*. W.W. Norton and Company, 2nd edition, 1976.

[2] Rana Adhikari. *Sensitivity and noise analysis of 4 km laser interferometric gravitational wave antennae*. PhD thesis, Massachusetts Institute of Technology, 2004.

[3] J. Aasi et al. Advanced LIGO. *Class. Quant. Grav.*, 32:074001, 2015. doi: 10.1088/0264-9381/32/7/074001.

[4] F. Acernese et al. Advanced Virgo: a second-generation interferometric gravitational wave detector. *Class. Quant. Grav.*, 32(2):024001, 2015. doi: 10.1088/0264-9381/32/2/024001.

[5] Grote H. for the LIGO Scientific Collaboration. The GEO 600 status. *Class. Quantum Grav.*, 27:084003, 2010.

[6] Kentaro Somiya. Detector configuration of KAGRA: The Japanese cryogenic gravitational-wave detector. *Class.Quant.Grav.*, 29:124007, 2012. doi: 10.1088/0264-9381/29/12/124007.

[7] C. S. Unnikrishnan. Indigo and ligo-india: Scope and plans for gravitational wave research and precision metrology in india. *International Journal of Modern Physics D*, 22(01):1341010, 2013. doi: 10.1142/S0218271813410101.

[8] Abbott, B. P. et al. Observation of gravitational waves from a binary black hole merger. *Phys. Rev. Lett.*, 116:061102, Feb 2016. doi: 10.1103/PhysRevLett.116.061102. URL `http://link.aps.org/doi/10.1103/PhysRevLett.116.061102`.

[9] Abbott, B. P. et al. Gw151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence. *Phys. Rev. Lett.*, 116:241103, Jun 2016. doi: 10.1103/PhysRevLett.116.241103. URL `http://link.aps.org/doi/10.1103/PhysRevLett.116.241103`.

[10] Abbott, B. P. et al. Gw170104: Observation of a 50-solar-mass binary black hole coalescence at redshift 0.2. *Phys. Rev. Lett.*, 118:221101, Jun 2017.

doi: 10.1103/PhysRevLett.118.221101. URL https://link.aps.org/doi/10.1103/PhysRevLett.118.221101.

[11] Abbott, B. P. et al. Gw170814: A three-detector observation of gravitational waves from a binary black hole coalescence. *Phys. Rev. Lett.*, 119:141101, Oct 2017. doi: 10.1103/PhysRevLett.119.141101. URL https://link.aps.org/doi/10.1103/PhysRevLett.119.141101.

[12] B. P. Abbott et al. GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs. 2018.

[13] Abbott, B. P. et al. Gw170817: Observation of gravitational waves from a binary neutron star inspiral. *Phys. Rev. Lett.*, 119:161101, Oct 2017. doi: 10.1103/PhysRevLett.119.161101. URL https://link.aps.org/doi/10.1103/PhysRevLett.119.161101.

[14] S Chatterji, L Blackburn, G Martin, and E Katsavounidis. Multiresolution techniques for the detection of gravitational-wave bursts. *Classical and Quantum Gravity*, 21(20):S1809, 2004. URL http://stacks.iop.org/0264-9381/21/i=20/a=024.

[15] Michael Zevin et al. Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science. *Class. Quant. Grav.*, 34 (6):064003, 2017. doi: 10.1088/1361-6382/aa5cea.

[16] S Klimenko, I Yakushin, A Mercer, and G Mitselmakher. Coherent method for detection of gravitational wave bursts. *Class. Quant. Grav.*, 25:114029, 2008.

[17] S. Klimenko, S. Mohanty, M. Rakhmanov, and G. Mitselmakher. Constraint likelihood analysis for a network of gravitational wave detectors. *Phys. Rev. D*, 72:122002, Dec 2005. doi: 10.1103/PhysRevD.72.122002. URL http://link.aps.org/doi/10.1103/PhysRevD.72.122002.

[18] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, and et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *Physical Review Letters*, 116(6):061102, February 2016. doi: 10.1103/PhysRevLett.116.061102.

[19] R. Essick, L. Blackburn, and E. Katsavounidis. Optimizing vetoes for gravitational-wave transient searches. *Classical and Quantum Gravity*, 30(15): 155010, August 2013. doi: 10.1088/0264-9381/30/15/155010.

[20] 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, 2011. URL `http://stacks.iop.org/0264-9381/28/i=23/a=235005`.

[21] Parameswaran Ajith, Tomoki Isogai, Nelson Christensen, Rana X. Adhikari, Aaron B. Pearlman, Alex Wein, Alan J. Weinstein, and Ben Yuan. Instrumental vetoes for transient gravitational-wave triggers using noise-coupling models: The bilinear-coupling veto. *Phys. Rev. D*, 89:122001, Jun 2014. doi: 10.1103/PhysRevD.89.122001. URL `http://link.aps.org/doi/10.1103/PhysRevD.89.122001`.

[22] Tomoki Isogai, the Ligo Scientific Collaboration, and the Virgo Collaboration. Used percentage veto for ligo and virgo binary inspiral searches. *Journal of Physics: Conference Series*, 243(1):012005, 2010. URL `http://stacks.iop.org/1742-6596/243/i=1/a=012005`.

[23] P. J. Sutton, G. Jones, S. Chatterji, P. Kalmus, I. Leonor, S. Poprocki, J. Rollins, A. Searle, L. Stein, M. Tinto, and M. Was. X-Pipeline: an analysis package for autonomous gravitational-wave burst searches. *New Journal of Physics*, 12:053034–+, 2010.

[24] Thomas S. Adams, Duncan Meacher, James Clark, Patrick J. Sutton, Gareth Jones, et al. Gravitational-Wave Detection using Multivariate Analysis. *Phys.Rev.*, D88:062006, 2013. doi: 10.1103/PhysRevD.88.062006.

[25] Michal Was, Patrick J. Sutton, Gareth Jones, and Isabel Leonor. Performance of an externally triggered gravitational-wave burst search. *Phys.Rev.*, D86: 022003, 2012. doi: 10.1103/PhysRevD.86.022003.

[26] Michal Was. Searching for gravitational waves associated with gamma-ray bursts in 2009-2010 LIGO-Virgo data. 2011.

[27] Sara Bahaadini, Vahid Noroozi, Neda Rohani, Scott Coughlin, Michael Zevin, and Aggelos K. Katsaggelos. DIRECT: Deep Discriminative Embedding for Clustering of LIGO Data. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 748–752, 2018. doi: 10.1109/ICIP.2018.8451708.

[28] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, M. Adamo, C. Adams, T. Adams, P. Addesso, and et al. Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914. *Classical and Quantum Gravity*, 33(13):134001, July 2016. doi: 10.1088/0264-9381/33/13/134001.

[29] J. Aasi, J. Abadie, B. P. Abbott, R. Abbott, T. Abbott, M. R. Abernathy, T. Accadia, F. Acernese, C. Adams, T. Adams, and et al. Characterization of the LIGO detectors during their sixth science run. *Classical and Quantum Gravity*, 32(11):115012, June 2015. doi: 10.1088/0264-9381/32/11/115012.

[30] Rahul Biswas, Lindy Blackburn, Junwei Cao, Reed Essick, Kari Alison Hodge, Erotokritos Katsavounidis, Kyungmin Kim, Young-Min Kim, Eric-Olivier Le Bigot, Chang-Hwan Lee, John J. Oh, Sang Hoon Oh, Edwin J. Son, Ye Tao, Ruslan Vaulin, and Xiaoge Wang. Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data. *Phys. Rev. D*, 88:062003, Sep 2013. doi: 10.1103/PhysRevD.88.062003. URL `https://link.aps.org/doi/10.1103/PhysRevD.88.062003`.

[31] Jade Powell, Daniele Trifirò, Elena Cuoco, Ik Siong Heng, and Marco Cavaglià. Classification methods for noise transients in advanced gravitational-wave detectors. *Class. Quant. Grav.*, 32(21):215012, 2015. doi: 10.1088/0264-9381/32/21/215012.

[32] Jade Powell, Alejandro Torres-Forné, Ryan Lynch, Daniele Trifirò, Elena Cuoco, Marco Cavaglià, Ik Siong Heng, and José A. Font. Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data. *Class. Quant. Grav.*, 34(3):034002, 2017. doi: 10.1088/1361-6382/34/3/034002.

[33] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip. Transient classification in ligo data using difference boosting neural network. *Phys. Rev. D*, 95:104059, May 2017. doi: 10.1103/PhysRevD.95.104059. URL `https://link.aps.org/doi/10.1103/PhysRevD.95.104059`.

[34] Daniel George, Hongyu Shen, and E. A. Huerta. Classification and unsupervised clustering of LIGO data with Deep Transfer Learning. *Phys. Rev. D*, 97 (10):101501, 2018. doi: 10.1103/PhysRevD.97.101501.

[35] Massimiliano Razzano and Elena Cuoco. Image-based deep learning for classification of noise transients in gravitational wave detectors. *Class. Quant. Grav.*, 35(9):095016, 2018. doi: 10.1088/1361-6382/aab793.

[36] K. D. Borne, L. Fortson, P. Gay, C. Lintott, M. J. Raddick, and J. Wallin. The Zooniverse. *AGU Fall Meeting Abstracts*, December 2009.

[37] B. Allen. χ^2 time-frequency discriminator for gravitational wave detection. *Phys. Rev. D*, 71(6):062001, March 2005. doi: 10.1103/PhysRevD.71.062001.

[38] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton. FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries. *"Phys. Rev. D"*, 85(12):122006, June 2012. doi: 10.1103/PhysRevD.85.122006.

[39] The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams,

T. Adams, and et al. Binary Black Hole Mergers in the first Advanced LIGO Observing Run. *ArXiv e-prints*, June 2016.

[40] J. Aasi et al. The NINJA-2 project: Detecting and characterizing gravitational waveforms modelled using numerical binary black hole simulations. *cqg*, 31: 115004, 2014.

[41] B. P. Abbott et al. Properties of the binary black hole merger gw150914. *Phys. Rev. Lett.*, 116:241102, Jun 2016. doi: 10.1103/PhysRevLett.116.241102. URL `http://link.aps.org/doi/10.1103/PhysRevLett.116.241102`.

[42] L. K. Nuttall, T. J. Massinger, J. Areeda, J. Betzwieser, S. Dwyer, A. Effler, R. P. Fisher, P. Fritschel, J. S. Kissel, A. P. Lundgren, D. M. Macleod, D. Martynov, J. McIver, A. Mullavey, D. Sigg, J. R. Smith, G. Vajente, A. R. Williamson, and C. C. Wipf. Improving the data quality of Advanced LIGO based on early engineering run results. *Classical and Quantum Gravity*, 32 (24):245005, December 2015. doi: 10.1088/0264-9381/32/24/245005.

[43] Beverly K. Berger. Identification and mitigation of advanced LIGO noise sources. *Journal of Physics: Conference Series*, 957:012004, feb 2018. doi: 10.1088/1742-6596/957/1/012004. URL `https://doi.org/10.1088%2F1742-6596%2F957%2F1%2F012004`.

[44] Robert Schofield. Damping reduced in-air velocity of swiss cheese baffle by more than an order of magnitude, May 2017. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=36147`. Advanced LIGO electonic log 36147.

[45] Jade Powell, Daniele Trifirò, Elena Cuoco, Ik Siong Heng, and Marco Cavaglià. Classification methods for noise transients in advanced gravitational-wave detectors. *Class. Quant. Grav.*, 32(21):215012, 2015. doi: 10.1088/0264-9381/32/21/215012.

[46] J. Powell, A. Torres-Forné, R. Lynch, D. Trifirò, E. Cuoco, M. Cavaglià, I. S. Heng, and J. A. Font. Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data. *ArXiv e-prints*, September 2016.

[47] N. Mukund, S. Abraham, S. Kandhasamy, s. S. Mitra, and N. S. Philip. Transient classification in ligo data using difference boosting neural networks. *ArXiv e-prints*, September 2016.

[48] Florent Robinet. Omicron: an algorithm to detect and characterize transient events in gravitational-wave detectors, December 2014. URL `https://tds.virgo-gw.eu/`. VIRGO technical document VIR-0545A-14.

[49] R. Lynch, S. Vitale, R. Essick, E. Katsavounidis, and F. Robinet. An information-theoretic approach to the gravitational-wave burst detection problem. *ArXiv e-prints*, November 2015.

[50] T. Hey, S. Tansley, and K. Tolle. The Fourth Paradigm: Data-Intensive Scientific Discovery. *Proceedings of the IEEE*, 99(8):1334–1337, 2011. doi: dx.doi.org/10.1109/JPROC.2011.2155130.

[51] B. Kanefsky, N. G. Barlow, and V. C. Gulick. Can Distributed Volunteers Accomplish Massive Data Analysis Tasks? In *Lunar and Planetary Science Conference*, volume 32 of *Lunar and Planetary Science Conference*, March 2001.

[52] B. J. H. Méndez. SpaceScience@Home: Authentic Research Projects that Use Citizen Scientists. In C. Garmany and J. W. Gibbs, Moody, editors, *EPO and a Changing World: Creating Linkages and Expanding Partnerships*, volume 389 of *Astronomical Society of the Pacific Conference Series*, page 219, June 2008.

[53] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *mnras*, 389:1179–1189, September 2008. doi: 10.1111/j.1365-2966.2008.13689.x.

[54] M. A. Galloway, K. W. Willett, L. F. Fortson, C. N. Cardamone, K. Schawinski, E. Cheung, C. J. Lintott, K. L. Masters, T. Melvin, and B. D. Simmons. Galaxy Zoo: the effect of bar-driven fuelling on the presence of an active galactic nucleus in disc galaxies. *mnras*, 448:3442–3454, April 2015. doi: 10.1093/mnras/stv235.

[55] A. Smith, C. Lintott, S. Bamford, and L. Fortson. Zooniverse - A Platform for Data-Driven Citizen Science. *AGU Fall Meeting Abstracts*, December 2011.

[56] S. Kendrew, R. Simpson, E. Bressert, M. S. Povich, R. Sherman, C. J. Lintott, T. P. Robitaille, K. Schawinski, and G. Wolf-Chase. The Milky Way Project: A Statistical Study of Massive Star Formation Associated with Infrared Bubbles. *ApJ*, 755:71, August 2012. doi: 10.1088/0004-637X/755/1/71.

[57] Christopher C. Hennon, Kenneth R. Knapp, Carl J. Schreck III, Scott E. Stevens, James P. Kossin, Peter W. Thorne, Paula A. Hennon, Michael C. Kruk, Jared Rennie, Jean-Maurice Gadéa, Maximilian Striegl, and Ian Carley. Cyclone center: Can citizen scientists improve tropical cyclone intensity records? *Bulletin of the American Meteorological Society*, 96(4):591–

607, 2015. doi: 10.1175/BAMS-D-13-00152.1. URL http://dx.doi.org/10.1175/BAMS-D-13-00152.1.

[58] J. E. et al. Geach. The red radio ring: a gravitationally lensed hyperluminous infrared radio galaxy at z = 2.553 discovered through the citizen science project space warps. *mnras*, 452:502–510, September 2015. doi: 10.1093/mnras/stv1243.

[59] LSST Science Collaboration, P. A. Abell, J. Allison, S. F. Anderson, J. R. Andrew, J. R. P. Angel, L. Armus, D. Arnett, S. J. Asztalos, T. S. Axelrod, and et al. LSST Science Book, Version 2.0. *ArXiv e-prints*, December 2009.

[60] Andrew Sears, Jonathan Lazar, Ant Ozok, and Gabriele Meiselwitz. Human-centered computing: Defining a research agenda. *International Journal of Human-Computer Interaction*, 24(1):2–16, 1 2008. ISSN 1044-7318. doi: 10.1080/10447310701771456.

[61] K. Crowston, C. Østerlund, and T. Kyoung Lee. Blending machine and human learning processes. In *Proceedings of Hawai'i International Conference on System Sciences HICSS*, CSCW '08, 2017.

[62] Tae Kyoung Lee, Kevin Crowston, Carsten Østerlund, and Grant Miller. Recruiting messages matter: Message strategies to attract citizen scientists. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17 Companion, pages 227–230, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4688-7. doi: 10.1145/3022198.3026335. URL http://doi.acm.org/10.1145/3022198.3026335.

[63] Corey Jackson, Kevin Crowston, Carsten Østerlund, and Mahboobeh Harandi. Folksonomies to support coordination and coordination of folksonomies. *Computer Supported Cooperative Work (CSCW)*, 27(3):647–678, Dec 2018. ISSN 1573-7551. doi: 10.1007/s10606-018-9327-z. URL https://doi.org/10.1007/s10606-018-9327-z.

[64] Corey Brian Jackson, Kevin Crowston, and Carsten Østerlund. Did they login?: Patterns of anonymous contributions in online communities. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW):77:1–77:16, November 2018. ISSN 2573-0142. doi: 10.1145/3274346. URL http://doi.acm.org/10.1145/3274346.

[65] Tae Kyoung Lee, Kevin Crowston, Mahboobeh Harandi, Carsten Østerlund, and Grant Miller. Appealing to different motivations in a message to recruit citizen scientists: results of a field experiment. *Journal of Science Communication*, 17(01), 2 2018. ISSN 1824-2049. doi: 10.22323/2.17010202.

[66] Kevin Crowston, Carsten S. Østerlund, Tae Kyoung Lee, Corey Brian Jackson, Mahboobeh Harandi, Sarah Allen, Sara Bahaadini, Scott Coughlin, Aggelos K.

Katsaggelos, Shane L. Larson, Neda Rohani, Joshua Ernest Smith, Laura Trouille, and Michael Zevin. Knowledge tracing to model learning in online citizen science projects.

[67] Joshua Introne, Robert Laubacher, Gary Olson, and Thomas Malone. Solving wicked social problems with socio-computational systems. *K?nstl Intell*, 27: 45, 12 2013. doi: 10.1007/s13218-012-0231-2.

[68] Nathan Prestopnik and Kevin Crowston. Purposeful gaming and socio-computational systems: A citizen science design case. In *Proceedings of the 17th ACM International Conference on Supporting Group Work*, GROUP '12, pages 75–84, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1486-2. doi: 10.1145/2389176.2389188.

[69] Aniket Kittur and Robert E. Kraut. Harnessing the wisdom of crowds in wikipedia: Quality through coordination. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, CSCW '08, pages 37–46, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-007-4. doi: 10.1145/1460563.1460572.

[70] Thomas W. Malone and Michael S. Bernstein. *Handbook of Collective Intelligence*. The MIT Press, 2015. ISBN 0262029812, 9780262029810.

[71] Abbott, B. P. et al. Gw151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence. *Phys. Rev. Lett.*, 116:241103, Jun 2016. doi: 10.1103/PhysRevLett.116.241103. URL `http://link.aps.org/doi/10.1103/PhysRevLett.116.241103`.

[72] C. et al. Biwer. Validating gravitational-wave detections: The advanced ligo hardware injection system. in prep., 2016.

[73] S. Mukherjee, R. Obaid, and B. Matkarimov. Classification of glitch waveforms in gravitational wave detector characterization. *Journal of Physics Conference Series*, 243(1):012006, August 2010. doi: 10.1088/1742-6596/243/1/012006.

[74] S. Rampone, V. Pierro, L. Troiano, and I. M. Pinto. Neural Network Aided Glitch-Burst Discrimination and Glitch Classification. *International Journal of Modern Physics C*, 24:1350084, November 2013. doi: 10.1142/S0129183113500848.

[75] Sheila Dwyer. Change in omc length gain helps with 1083 hz glitches, January 2017. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=33104`. Advanced LIGO electonic log 33104.

[76] Bubba Gateley. End station instrument air compressors, 2015. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=22081`. Advanced LIGO electonic log 22081.

[77] Joshua Smith. The 50hz glitches in darm: Ex mains glitches coupling into ex seismic, 2015. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=21436`. Advanced LIGO electonic log 21436.

[78] Andrew Lundgren. New glitch class: paired doves, May 2016. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=27138`. Advanced LIGO electonic log 27138.

[79] T Accadia, F Acernese, F Antonucci, P Astone, G Ballardin, F Barone, M Barsuglia, Th S Bauer, MG Beker, A Belletoile, et al. Noise from scattered light in virgo's second science run data. *Classical and Quantum Gravity*, 27(19): 194011, 2010.

[80] ShinWoo Kim and Gregory L Murphy. Ideals and category typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(5):1092, 2011. ISSN 1939-1285.

[81] Chan Kulatunga-Moruzi, Lee R Brooks, and Geoffrey R Norman. Teaching posttraining: influencing diagnostic strategy with instructions at test. *Journal of Experimental Psychology: Applied*, 17(3):195, 2011. ISSN 1433810859.

[82] Brett D Roads and Michael C Mozer. Improving human-machine cooperative classification via cognitive theories of similarity. *Cognitive Science: A Multidisciplinary Journal*, In press. URL `https://www.cs.colorado.edu/~mozer/Research/Selected\%20Publications/reprints/RoadsMozer2016.pdf`.

[83] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[84] A. K. Katsaggelos, S. Bahaadini, and R. Molina. Audiovisual fusion: Challenges and new approaches. *Proceedings of the IEEE*, 103(9):1635–1653, Sept 2015.

[85] N. Rohani, P. Ruiz, E. Besler, R. Molina, and A. K. Katsaggelos. Variational gaussian process for sensor fusion. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 170–174, Aug 2015.

[86] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[87] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J.R. Smith, V. Kalogera, and A. Katsaggelos. Machine learning for gravity spy: Glitch classification and dataset. *Information Sciences*, 444:172–186, 2018. doi: https://doi.org/10.1016/j.ins.2018.02.068.

[88] Emre Besler and Aggelos Katsaggelos. Aiding classification tasks by combining machine learning and crowdsourcing data. In *To be submitted to EUSIPCO 2017*, 2017.

[89] M. et al. Zevin. The synergy of humans and machines: Improving the efficiency of classification schemes. in prep., 2019.

[90] Josh Smith. Glitches from misbehaving pcal-y on october 9, October 2015. URL `https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=21463`. Advanced LIGO electonic log 21463.

[91] LIGO-Virgo Collaboration. Methods for classification and characterization of transient noise in the first observing run of advanced ligo. in prep., 2016.

[92] Coughlin, S. and Zevin, M. Optimization of citizen science output in the era of big data,. in prep., 2016.

[93] Sara Bahaadini, Neda Rohani, Scott Coughlin, Michael Zevin, Vicky Kalogera, and Aggelos K Katsaggelos. Deep Multi-view Models for Glitch Classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2931–2935, 2017. doi: 10.1109/ICASSP.2017.7952693.

[94] Derek Davis, Laurel White, and Miriam Cabero. Correlations between O2 blip glitches and low relative humidity, April 2018. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=41263`. Advanced LIGO electonic log 41263.

[95] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ArXiv e-prints*, 2014.

[96] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

[97] Duncan Macleod, Alex L. Urban, Scott Coughlin, Thomas Massinger, paulaltin, Joseph Areeda, Eric Quintero, and The Gitter Badger. gwpy/gwpy: 0.12.2, Oct 2018.

[98] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9:2579–2605, 2008.

[99] Thomas Dent. Severe transient scattering events in darm caused by loud 20–30 Hz disturbances ('thuds') in cs/lvea, January 2017. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=33761`. Advanced LIGO electronic log 33761.

[100] Robbert Schofield. Why the GW channel detects thirsty black ravens along with colliding black holes, July 2017. URL `https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=37630`. Advanced LIGO electronic log 37630.

[101] L. K. Nuttall. Characterizing transient noise in the LIGO detectors. *Phil. Trans. Roy. Soc. Lond.*, A376(2120):20170286, 2018. doi: 10.1098/rsta.2017.0286.

[102] H. Domínguez Sánchez et al. Transfer learning for galaxy morphology from one survey to another. *Mon. Not. Roy. Astron. Soc.*, 484(1):93–100, 2019. doi: 10.1093/mnras/sty3497.

[103] Asad Khan, E. A. Huerta, Sibo Wang, and Robert Gruendl. Unsupervised learning and data clustering for the construction of Galaxy Catalogs in the Dark Energy Survey. *ArXiv e-prints*, 2018.

[104] H. Domínguez Sánchez, M. Huertas-Company, M. Bernardi, D. Tuccillo, and J. L. Fischer. Improving galaxy morphologies for SDSS with Deep Learning. *Mon. Not. Roy. Astron. Soc.*, 476:3661–3676, May 2018. doi: 10.1093/mnras/sty338.

[105] Mahabal Ashish et al. Machine Learning for the Zwicky Transient Facility. *Publ. Astron. Soc. Pac.*, 131:038002, Mar 2019. doi: 10.1088/1538-3873/aaf3fa.

[106] Z. Ivezic et al. LSST: from science drivers to reference design and anticipated data products. *http://arxiv.org/abs/0805.2366*, 2014.

[107] Cathal Doyle, Yevgeniya Li, Markus Luczak-Roesch, Dayle Anderson, Brigitte Glasson, Matthew Boucher, Carol Brieseman, Dianne Christenson, and Melissa Coton. What is online citizen science anyway? An educational perspective. *ArXiv e-prints*, Apr 2018.

[108] Ginger Tsueng, Arun Kumar, Max Nanis, and Andrew I Su. Aligning needs: Integrating citizen science efforts into schools through service requirements. *bioRxiv e-prints*, 2018. doi: 10.1101/304766.

[109] Harsh R Shah and Luis R Martinez. Current approaches in implementing citizen science in the classroom. *Journal of Microbiology & Biology Education*, 17(1):17–22, March 2016. doi: 10.1128/jmbe.v17i1.1032.

[110] Abbott, B. P. et al. First targeted search for gravitational-wave bursts from core-collapse supernovae in data of first-generation laser interferometer detectors. *Phys. Rev. D*, 94:102001, Nov 2016. doi: 10.1103/PhysRevD.94.102001. URL `https://link.aps.org/doi/10.1103/PhysRevD.94.102001`.

[111] Eric Thrane et al. Long gravitational-wave transients and associated detection strategies for a network of terrestrial interferometers. *Physical Review D*, 83: 083004, 2011.

[112] Warren G. Anderson, Patrick R. Brady, Jolien D. E. Creighton, and Éanna É. Flanagan. Excess power statistic for detection of burst sources of gravitational radiation. *Phys. Rev. D*, 63:042003, Jan 2001. doi: 10.1103/PhysRevD.63.042003. URL `http://link.aps.org/doi/10.1103/PhysRevD.63.042003`.

[113] F. J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, Jan 1978. ISSN 0018-9219. doi: 10.1109/PROC.1978.10837.

[114] Abadie, J et al. Search for gravitational waves from low mass compact binary coalescence in ligo's sixth science run and virgo's science runs 2 and 3. *Physical Review D*, 85(8):082002, 2012.

[115] G. Pagliaroli, F. Vissani, E. Coccia, and W. Fulgione. Neutrinos from Supernovae as a Trigger for Gravitational Wave Search. *Phys. Rev. Lett.*, 103(3): 031102, July 2009.

[116] Antony C Searle, Patrick J Sutton, Massimo Tinto, and Graham Woan. Robust bayesian detection of unmodelled bursts. *Classical and Quantum Gravity*, 25(11):114038, 2008. URL `http://stacks.iop.org/0264-9381/25/i=11/a=114038`.

[117] Rahul Biswas et al. Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data. *Phys. Rev. D*, 88(6):062003, 2013. doi: 10.1103/PhysRevD.88.062003.

[118] Marco Cavaglia, Kai Staats, and Teerth Gill. Finding the origin of noise transients in LIGO data with machine learning. *Commun. Comput. Phys.*, 25 (4):963–987, 2019. doi: 10.4208/cicp.OA-2018-0092.

[119] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[120] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

[121] Philippe Landry and Reed Essick. Nonparametric inference of the neutron star equation of state from gravitational wave observations. *Phys. Rev. D*, 99: 084049, Apr 2019. doi: 10.1103/PhysRevD.99.084049. URL `https://link.aps.org/doi/10.1103/PhysRevD.99.084049`.

[122] James Hensman, Alex Matthews, and Zoubin Ghahramani. Scalable Variational Gaussian Process Classification. *arXiv e-prints*, art. arXiv:1411.2005, Nov 2014.

[123] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017. URL `http://jmlr.org/papers/v18/16-537.html`.

[124] Shunsaku Horiuchi, John F. Beacom, Matt S. Bothwell, and Todd A. Thompson. EFFECTS OF STELLAR ROTATION ON STAR FORMATION RATES AND COMPARISON TO CORE-COLLAPSE SUPERNOVA RATES. *The Astrophysical Journal*, 769(2):113, may 2013. doi: 10.1088/0004-637x/769/2/113.

[125] Pietro Antonioli, Richard Tresch Fienberg, Fabrice Fleurot, Yoshiyuki Fukuda, Walter Fulgione, et al. SNEWS: The Supernova Early Warning System. *New J.Phys.*, 6:114, 2004. doi: 10.1088/1367-2630/6/1/114.

[126] M. Ikeda et al. Search for Supernova Neutrino Bursts at Super-Kamiokande. *ApJ*, 669:519, nov 2007. doi: 10.1086/521547.

[127] C. Vigorito and LVD Collaboration. Galactic supernovae monitoring at LVD. *Nuclear Physics B Proceedings Supplements*, 221:410–410, December 2011. doi: 10.1016/j.nuclphysbps.2011.10.054.

[128] T. Gaisser and F. Halzen. Icecube. *Ann. Rev. Nuc. Part. Sc.*, 64(1):101, 2014.

[129] G Alimonti, C Arpesella, H Back, M Balata, D Bartolomei, A De Bellefon, G Bellini, J Benziger, A Bevilacqua, D Bondi, et al. The borexino detector at the laboratori nazionali del gran sasso. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 600(3):568–593, 2009.

[130] Daya Bay Collaboration. A side-by-side comparison of daya bay antineutrino detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 685:78 – 97, 2012. ISSN 0168-9002. doi: http://dx.doi.org/10.1016/j.nima.2012.05.030. URL `http://www.sciencedirect.com/science/article/pii/S016890021200530X`.

[131] KamLAND Collaboration. First results from kamland: Evidence for reactor antineutrino disappearance. *Phys. Rev. Lett.*, 90:021802, Jan 2003. doi: 10.1103/PhysRevLett.90.021802. URL `http://link.aps.org/doi/10.1103/PhysRevLett.90.021802`.

[132] SNO Colloboration. The sudbury neutrino observatory. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 449(1):172 – 207, 2000. ISSN 0168-9002. doi: https://doi.org/10.1016/S0168-9002(99)01469-2. URL `http://www.sciencedirect.com/science/article/pii/S0168900299014692`.

[133] H. Andresen, B. Müller, E. Müller, and H.-Th. Janka. Gravitational wave signals from 3D neutrino hydrodynamics simulations of core-collapse supernovae. *Monthly Notices of the Royal Astronomical Society*, 468(2):2032–2051, 03 2017. ISSN 0035-8711. doi: 10.1093/mnras/stx618. URL `https://doi.org/10.1093/mnras/stx618`.

[134] Philipp Mösta, Sherwood Richers, Christian D. Ott, Roland Haas, Anthony L. Piro, Kristen Boydstun, Ernazar Abdikamalov, Christian Reisswig, and Erik Schnetter. MAGNETOROTATIONAL CORE-COLLAPSE SUPERNOVAE IN THREE DIMENSIONS. *The Astrophysical Journal*, 785(2):L29, apr 2014. doi: 10.1088/2041-8205/785/2/l29.

[135] N. Nishimura, H. Sawai, T. Takiwaki, S. Yamada, and F.-K. Thielemann. The intermediate r-process in core-collapse supernovae driven by the magneto-rotational instability. *The Astrophysical Journal*, 836(2):L21, feb 2017. doi: 10.3847/2041-8213/aa5dee.

[136] K. Kotake and T. Kuroda. *Gravitational Waves from Core-Collapse Supernovae*, page 1671. 2017. doi: 10.1007/978-3-319-21846-5_9.

[137] Jade Powell and Bernhard Müller. Gravitational Wave Emission from 3D Explosion Models of Core-Collapse Supernovae with Low and Normal Explosion Energies. *Monthly Notices of the Royal Astronomical Society*, 05 2019. ISSN 0035-8711. doi: 10.1093/mnras/stz1304. URL `https://doi.org/10.1093/mnras/stz1304`.

[138] B. Müller, H.-T. Janka, and A. Marek. A New Multi-dimensional General Relativistic Neutrino Hydrodynamics Code of Core-collapse Supernovae. III. Gravitational Wave Signals from Supernova Explosion Models. *ApJ*, 766:43, 2013. doi: 10.1088/0004-637X/766/1/43.

[139] Jade Powell. Parameter Estimation and Model Selection of Gravitational Wave Signals Contaminated by Transient Detector Noise Glitches. *Class. Quant. Grav.*, 35(15):155017, 2018. doi: 10.1088/1361-6382/aacf18.

[140] S. E. Gossan, P. Sutton, A. Stuver, M. Zanolin, K. Gill, and C. D. Ott. Observing gravitational waves from core-collapse supernovae in the advanced detector era. *Phys. Rev. D*, 93:042002, Feb 2016. doi: 10.1103/PhysRevD.93.042002. URL `https://link.aps.org/doi/10.1103/PhysRevD.93.042002`.

[141] J. F. Beacom and P. Vogel. Can a supernova be located by its neutrinos? *Phys. Rev. D*, 60(3):033007, 1999. doi: 10.1103/PhysRevD.60.033007.

[142] C. D. Ott. TOPICAL REVIEW: The gravitational-wave signature of core-collapse supernovae. *Class. Quantum Grav.*, 26:063001, 2009. doi: 10.1088/0264-9381/26/6/063001.

[143] Viktoriya Morozova, David Radice, Adam Burrows, and David Vartanyan. The gravitational wave signal from core-collapse supernovae. *The Astrophysical Journal*, 861(1):10, jun 2018. doi: 10.3847/1538-4357/aac5f1.

[144] Evan P. O'Connor and Sean M. Couch. Exploring fundamentally three-dimensional phenomena in high-fidelity simulations of core-collapse supernovae. *The Astrophysical Journal*, 865(2):81, sep 2018. doi: 10.3847/1538-4357/aadcf7.

[145] Sherwood Richers, Christian D. Ott, Ernazar Abdikamalov, Evan O'Connor, and Chris Sullivan. Equation of state effects on gravitational waves from rotating core collapse. *Phys. Rev. D*, 95:063019, Mar 2017. doi: 10.1103/PhysRevD.95.063019. URL `https://link.aps.org/doi/10.1103/PhysRevD.95.063019`.

[146] S. Scheidegger, S. C. Whitehouse, R. Käppeli, and M. Liebendörfer. Gravitational waves from supernova matter. *Class. Quantum Grav.*, 27:114101, June 2010.

[147] K. Kotake. Multiple physical elements to determine the gravitational-wave signatures of core-collapse supernovae. *Comptes Rendus Physique*, 14:318, 2013. doi: 10.1016/j.crhy.2013.01.008.

[148] H. Dimmelmeier, C. D. Ott, A. Marek, and H.-T. Janka. Gravitational wave burst signal from core collapse of rotating stars. *Phys. Rev. D*, 78:064056, 2008.

[149] Takami Kuroda, Kei Kotake, Kazuhiro Hayama, and Tomoya Takiwaki. Correlated signatures of gravitational-wave and neutrino emission in three-dimensional general-relativistic core-collapse supernova simulations. *The Astrophysical Journal*, 851(1):62, dec 2017. doi: 10.3847/1538-4357/aa988d.

[150] Aasi, J. et al. Methods and results of a search for gravitational waves associated with gamma-ray bursts using the geo 600, ligo, and virgo detectors. *Phys.*

Rev. D, 89:122004, Jun 2014. doi: 10.1103/PhysRevD.89.122004. URL `http://link.aps.org/doi/10.1103/PhysRevD.89.122004`.

[151] Derek Davis, Thomas Massinger, Andrew Lundgren, Jennifer C Driggers, Alex L Urban, and Laura Nuttall. Improving the sensitivity of advanced LIGO using noise subtraction. *Classical and Quantum Gravity*, 36(5):055011, feb 2019. doi: 10.1088/1361-6382/ab01c5.

[152] Hunter Gabbard, Michael Williams, Fergus Hayes, and Chris Messenger. Matching matched filtering with deep networks for gravitational-wave astronomy. *Phys. Rev. Lett.*, 120:141103, Apr 2018. doi: 10.1103/PhysRevLett.120.141103. URL `https://link.aps.org/doi/10.1103/PhysRevLett.120.141103`.

[153] Timothy D. Gebhard, Niki Kilbertus, Ian Harry, and Bernhard Schölkopf. Convolutional neural networks: a magic bullet for gravitational-wave detection? 2019.

[154] Daniel George and E.A. Huerta. Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced ligo data. *Physics Letters B*, 778:64 – 70, 2018. ISSN 0370-2693. doi: https://doi.org/10.1016/j.physletb.2017.12.053. URL `http://www.sciencedirect.com/science/article/pii/S0370269317310390`.