Mitigating the effects of instrumental artifacts on source localizations

Maggie C. Huber
University of Michigan, Ann Arbor

Mentor: Derek Davis
LIGO, California Institute of Technology, Pasadena

Final Report
LIGO Caltech SURF Program 2021

Instrumental artifacts which materialize as glitches in strain data can overlap with gravitational wave detections and significantly impair the accuracy of sky localizations of compact binary coalescence (CBC) signals. We present our Python package, PySLIDE (Python-based Skymap Localization with Inpainted Data Editor), which takes gravitational wave (GW) signals, removes a segment of the data, and corrects for the removal. To make this correction, we employ a method that applies a reweighting formula to the signal-to-noise ratio (SNR) of the signal. From tests on ≈500 simulated GW signals, we determined that reweighting the SNR timeseries is able to improve the accuracy over simply removing the bad data. When we repeated this process for raw data with a simulated glitch, the reweighting formula likewise improves upon removing the data alone. In this report we discuss the method we used to reweight the SNR, features of PySLIDE, and the results of our tests on simulated GW signals.

I. INTRODUCTION

Detection of gravitational waves requires extreme sensitivity to changes in length on the order of 10^{-18} m [1]. The level of strain sensitivity renders LIGO detectors susceptible to noise transients (also called glitches), which are bursts of excess power in the detector [2]. Often, what causes these glitches is difficult to determine. They can be the result of either external environmental or internal instrumental interactions that alter the actual strain. Glitches are more likely to overlap with gravitational wave (GW) events that occur for a longer period, such as binary neutron star (BNS) events. As detection of GW events from BNS mergers become increasingly frequent [3], we expect to see more instances of noise transients overlapping with GW signals as seen in the case of BNS merger GW170817 [4].

Glitches are problematic for many reasons, and some types impact rapid sky localization and by extension all parameter estimation of GW signals. Instrumental artifacts can even trigger a false positive for an event, and strategies to avoid this are in active development [5, 6]. In the case of astrophysical signals, glitches can create false or biased skymaps. This is significant for BNS mergers that produce electromagnetic (EM) radiation requiring rapid and accurate followup observation. For GW170817 and throughout the second LIGO observing run (O2), glitch mitigation techniques were crucial to the success of EM followup campaigns [7]. These observations are important to growing the wider field of multi-messenger astronomy [8] and maximizing the types of signals obseved from a single event. EM followup, although important, is difficult to optimize [9, 10] and costs valuable telescope time. When observers are given an incorrect source location it magnifies these issues. In order to gain useful and accurate astrophysical information from a GW event, it is important that glitches are handled in a way that minimizes bias in localization measurements.

There are multiple approaches one can use to address a glitch which overlaps a GW signal. For GW170817, the effects of the noise transient were mitigated by applying a window function to remove it [11]. Additionally, the glitch waveform was reconstructed with an analytic model extracted through BAYESWAVE [12] that could be subtracted from the data [4], as shown in Figure 1. This method is ad hoc in nature, as we do not always have an exact mathematical model for a glitch. Different approaches are necessary to find a generalized solution that works instantaneously for various types of glitches. There have already been efforts to address this issue, notably the NNETFIX [13] algorithm that uses neural networks to model strain for gated portions of GW signals.

Window functions such as the one used for GW170817 gradually remove bad data to avoid discontinuity. However, they can introduce excess power leakage from the spectral lines in the power spectral density (PSD) of the detector. An alternative to window functions is inpainting [14], where the effects of discontinuities are calculated and subtracted. The end result is a gate that only masks bad seconds of data and has no affect on the data surrounding the inpainted hole.

When we inpaint a hole in GW data, we lose information about the amplitude and phase of a signal, which biases the sky localization. This effect is less noticeable when the fraction of data removed is less than $\lesssim 5\%$ of the total signal duration. For larger inpainting widths, this can add a significant bias to the sky localization. To ensure localization is accurate, it is necessary to correct for the effect of gating a portion of the signal.

To correct our skymaps for inpainting bias, we developed an algorithm to reweight the signal-to-noise ratio (SNR) timeseries. Our work combines techniques to

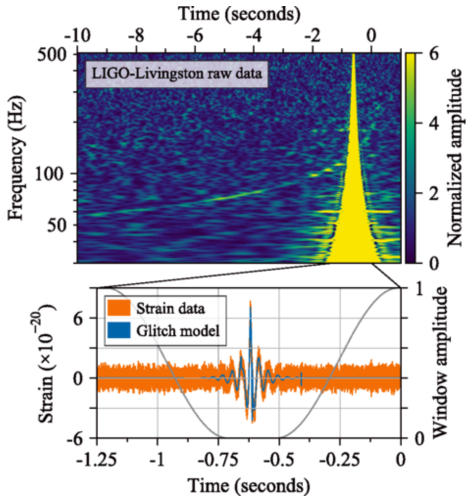

FIG. 1. *Top panel*: Time-frequency plot of LIGO Livingston data for GW170817 with the glitch present. *Bottom panel*: Strain data of the glitch, with a grey window function used to zero it out. The reproduced model of the glitch is shown with the blue curve. Replicated from [4].

manage glitches into a simple and computationally efficient package, and rigorously tests how our methods perform. The goal is to recover the correct error and accuracy so that the skymap includes the signal and guides EM telescopes in the right direction.

We used the BAYESTAR [15] algorithm in the PyCBC search pipeline [16] to create our sky localizations. BAYESTAR localizes GW sources using Bayesian inference instead of Markov chain Monte Carlo (MCMC) methods. It takes a likelihood function and a well-defined parameter space to rapidly infer the location of GW signals.

In this report we explain the functionality of our method, the setup of our Python package PySLIDE (Python-based Skymap Localizationwith Inpainted Data Editor), and the metrics we used to show how well it performs. We determine that PySLIDE was able to recover a more correct skymap both in the case of simulated signals with and without a glitch. Additionally, the package is computationally efficient and has a deterministic underlying formula with flexible input parameters.

II. PYSLIDE

A. Methods

For the initial set up of the underlying code, we first use GWpy [17] to generate a PSD and get a matched filter from PyCBC [16] to run the waveform template through the noisy data and calculate the SNR. The matched filter function for SNR timeseries $\rho(t)$ is given by a weighted inner product of the detector data s and template $h(t)$ where [16]

$$\begin{aligned}\rho^2(t) &= \frac{(s|h_{cos})^2}{(h_{cos}|h_{cos})} + \frac{(s|h_{sin})^2}{(h_{sin}|h_{sin})} \\ &= \frac{(s|h_{cos})^2 + (s|h_{sin})^2}{(h_{cos}|h_{cos})} \ .\end{aligned} \tag{1}$$

The inner product $(s|h)$ is given by

$$(s|h)(t) = 4\mathrm{Re}\int_{f_{low}}^{f_{high}} \frac{\tilde{s}(f)\tilde{h^*}(f)}{S_n(f)} \mathrm{e}^{2\pi i f t}\mathrm{d}f \tag{2}$$

with $S_n(f)$ as the PSD. This function technically gives a complex SNR, with the real part corresponding to a template that is lined up along the data and the imaginary part corresponding to a template that is 90 degrees out of phase.

The SNR timeseries given by the matched filtering process is a key input into a sky localization for a GW signal. The other two parameters needed for a skymap are the time delay in each detector and the phase of the signal. We do not expect the presence of a glitch to bias the time delay measurements or the phase, so PySLIDE focuses on correcting the amplitude (SNR).

Once the localization parameters are input into BAYESTAR, there are other important quantities we extract from the skymap. The credible regions (we use 50 percent and 90 percent credible regions in the algorithm) represent the cumulative sum of pixels in a given region. For example, the 90 percent credible region will contain 90 percent of pixels in the skymap. The area of this region is analogous to measuring the precision of the skymap. There is also the total searched area, which is the smallest credible region containing the location of the source and can be thought of as a measure of the accuracy.

B. Reweighting

As mentioned in the last section, parameters needed to localize a GW signal are the time delay, phase and the SNR. The presence of an inpainted hole in the data causes the SNR timeseries to deviate. We start by assuming we have a known signal template and input the relevant parameters. The SNR remaining after inpainting is given by [14]

$$\lambda_{hole}(t_0, h) \approx \frac{(|h_w|^2 \circledast \mathbb{1}_{valid})(t_0)}{\sum_t |h_w(t)|^2} \ , \tag{3}$$

where t_0 is the merger time, h_w is the whitened waveform, and $\mathbb{1}_{valid}$ returns zero for a data point in the inpainted hole and one otherwise. The equation convolves h_w with $\mathbb{1}_{valid}$, which we compute with a fast Fourier transform (FFT) as allowed by the convolution theorem.

After we inpaint and apply Equation 3, we multiply a normalization factor to the PSD and SNR timeseries.

When a portion of a signal is inpainted, we effectively decrease the sensitivity of our measurement. Renormalizing the PSD corrects the error for our BAYESTAR localization. For the results shown in this report, we used a factor corresponding to the maximum SNR value of the timeseries calculated in Equation 3. We are effectively reweighting the PSD and keeping the SNR the same to ensure that BAYESTAR gets the correct error measurement. If we inpaint a hole, we are losing information and the error should increase to account for that if we want the true source location to be included in the credible region of the skymap. When we refer to reweighting, we are discussing the process of normalizing the PSD to account for the correction rather than directly applying the formula to the SNR timeseries.

There are multiple advantages of using the reweighting method to renormalize the PSD. The algorithm is independent of how much time is removed with inpainting and which waveform template we use, and it allows the user to configure these settings as inputs. Reweighting is also deterministic - the calculation is the same for any variation of the input parameters. It is also instantaneous to compute, typically taking less than a second. These benefits render this method conducive to rapid and accurate sky localization of GW events in real time, even in the presence of glitches.

C. PySLIDE Workflow

PySLIDE can apply and test the reweighting method to any number of signal injections, though there are diminishing returns when you get to ≈500 injection runs. It creates a PyCondor [18] workflow that separates each task into jobs, which are then collected into a Dagman [18] object to submit to a computing cluster.

There are several components of the package that create results and plot them. The first script that reweights the SNR timeseries using injected signal parameters and a PSD. It creates a timeseries with the raw data, inpaints a hole for a specified segment, applies the reweighting formula to the SNR and PSD, and corrects for the SNR remaining. The script outputs the SNR timeseries as an XML file to be fed into the BAYESTAR algorithm.

BAYESTAR returns a localization as a FITS file. Using this file, we run scripts to calculate the credible region of the true source location, the total searched area, the area of the 90 percent credible region, and the overlap of the reweighted and inpainted skymaps with the raw skymap [19]. PySLIDE combines the results from all injection runs into one file and then that file is used for four different plotting scripts to visualize the final results.

The final component of the workflow is a script that creates an HTML page showing the summary plots front page with the performance metrics collected from BAYESTAR for an overall view of the workflow. There are pages showing the skymap plots and all parameters for the individual injection runs. On the bottom of the page there is a link to information on the command used in terminal to run the workflow, the CONFIG file with the initial variables, and the components of the environment used to run the workflow.

D. Testing simulated signals

To assess the performance of our reweighting algorithm, we simulated 500 compact binary coalescence (CBC) signals for testing. We used a gate width of 64 ms starting 64 ms from coincident time, set both masses to 10 $M_{\odot}$, and used a distance range of 10-500 Mpc. For the waveform template, we selected SEOBNRv4 [20] and filtered the signal template list to include what we would expect to detect by applying an SNR threshold of 8. We chose these signal parameters corresponding to what we expected to be the most biased by this method. If a test runs successfully, we verify that it is likely to work with most other cases.

To run our tests, we inject the simulated signals into the background PSD from the LIGO Hanford (H1) and Livingston (L1) detectors and get the raw SNR timeseries. We then get the SNR timeseries from using the inpainting function alone, then both inpainting and reweighting. We create an XML file which is put into BAYESTAR to localize all three cases. To see how the method performs, we obtain the credible region, searched area, the area of the 90 percent credible region, and the overlap.

After testing cases for raw data without a glitch, we wanted to see if we could create a glitch that biased the skymap and recover the source location by reweighting. We injected a sine-gaussian wavelet with a frequency of 80 Hz and strain of 2.5×10^{-21} m. We then created skymaps using the same method as the data without a glitch and obtained the same metrics from BAYESTAR to see if we corrected the glitch and inpainting bias.

III. RESULTS

A. Raw data without a glitch

Various metrics from BAYESTAR allow us to determine how reweighting compares to inpainting alone. We primarily use probability-probability (P-P) plots showing the credible region of the true source location vs. the fraction of total simulated signals (Figure 2). Ideally, the distribution on a P-P plot is linear with a slope of one. Due to an internal factor in BAYESTAR to normalize the plot in GstLAL, the raw data without the glitch lies above the diagonal. This distribution above the diagonal is therefore ideal for our plots made using the PyCBC pipeline.

One implication from the plot we created is that inpainting a hole in the data will bias the skymap and report an incorrect error. When we reweight the SNR

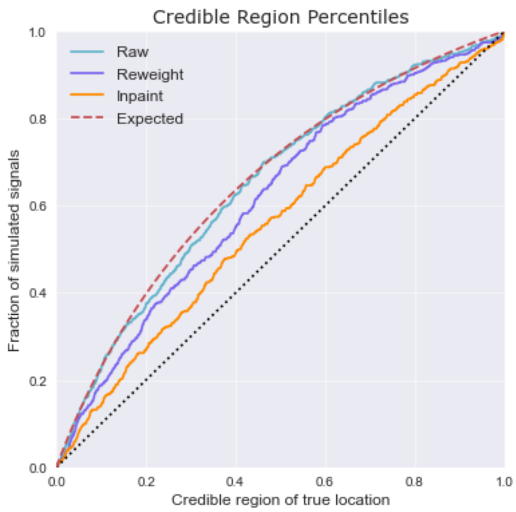

FIG. 2. P-P plot for data without a glitch showing the credible region returned by BAYESTAR of the true source location. Due to the normalization factor in BAYESTAR meant for the GstLAL pipeline, we can see the raw and reweighted lines are overestimating the error. This causes the lines to inflate above the diagonal. From this plot we can determine that inpainting a hole in the data causes the error to be underestimated, and reweighting recovers some of the error.

timeseries, the error is recovered and the skymap is more likely to return a localization that contains the source.

When we find the correct normalization to apply to the PSD, the reweighted P-P plot will line up with the raw data without a glitch and we will be accounting for the increase in error after inpainting.

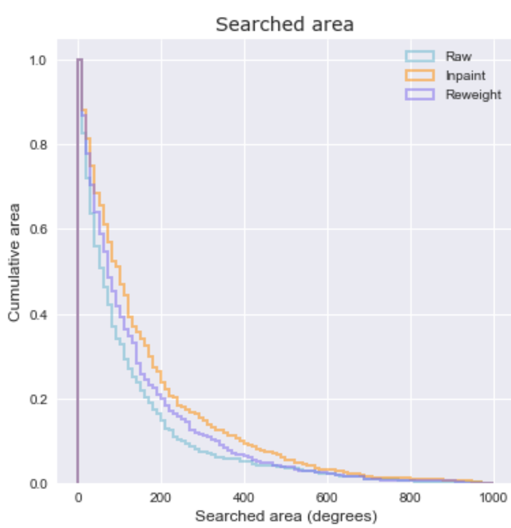

FIG. 3. Histogram showing the total searched area without a glitch by BAYESTAR in degrees vs. the cumulative sum of signals. The raw data in blue is the closest to the ideal distribution. Inpainting a hole causes the searched area to deviate and reweighting corrects for this effect by getting closer to the expected curve.

To check if the skymap shows an accurate credible region, we create a histogram of the total searched area in degrees (Figure 3). The searched area we refer to is the area of the credible region housing the true source location. Ideally, the cumulative area drops off faster as the searched area increases. This demonstrates that the resulting skymap predicts the source location to be in the lower credible region.

Similar to the results of the P-P plots, the searched area histograms show reweighting the data gets the distribution closer to the original data.

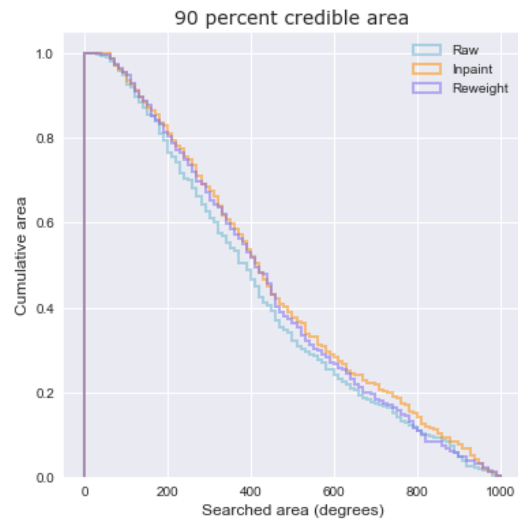

FIG. 4. Histogram showing the area of the 90 percent credible region on the skymap, which we use to roughly estimate the precision of our measurement. Because we are not actually changing the SNR timeseries and reweighting the PSD to make sure we account for the loss of sensitivity, the curves on this graph should line up with each other which appears to be the case.

For the area of the 90 percent credible region as seen in Figure 4, we expect the distributions for all three data sets to be roughly similar. The area is a way of measuring the precision of our skymaps, which is calculated using the SNR. Since the reweighting method used is only applied to the PSD to account for loss of sensitivity, we do not expect the precision to change. Figure 4 shows a distribution that agrees with our expectations.

The overlap in Figure 5 is only a meaningful metric when the raw data has no glitch present. For the ideal distribution, 90 percent of signals should have an overlap of 90 percent or greater with the raw skymap. The inpainted histogram does not follow this distribution, while the reweighted histogram does.

B. Raw data with a glitch

For the data with a simulated glitch close to the time of merger, we created the same figures to determine if

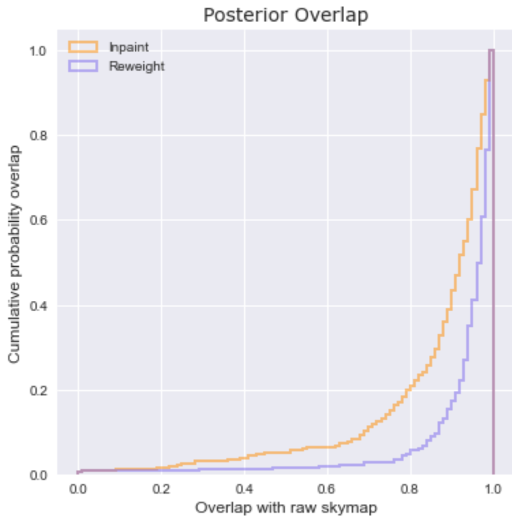

FIG. 5. Histogram showing the overlap of the inpainted and reweighted skymaps with the raw skymap. This plot is only an important performance metric when the raw data has no glitch and we can assume that it represents the correct skymap. The ideal curve has 90 percent of signals to the right of the line marking 90 percent on the x axis. If a fraction of signals fall to the left of that line, then we can assume that we will miss detections of those signals. We see this happening more for the inpainted curve, showing that we effectively miss more detections when the reweighting formula is not applied.

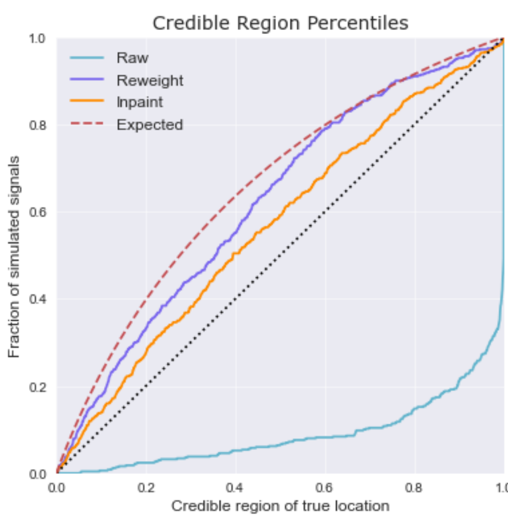

FIG. 6. P-P plot with a glitch showing the credible region returned by BAYESTAR of the true source location. Due to the normalization factor in BAYESTAR meant for the GstLAL pipeline, we can see the raw and reweighted lines are overestimating the error. This causes the reweighted line to inflate above the diagonal. From this plot we can determine that a glitch in the data underestimates the error, which we are able to improve with inpainting and even more by reweighting the signal-to-noise ratio (SNR).

reweighting recovers a more accurate skymap than inpainting alone. For the P-P plot in Figure 6, the glitch biases the error estimate in BAYESTAR. Inpainting corrects for some of the error, and reweighting gets a more accurate estimate than inpainting.

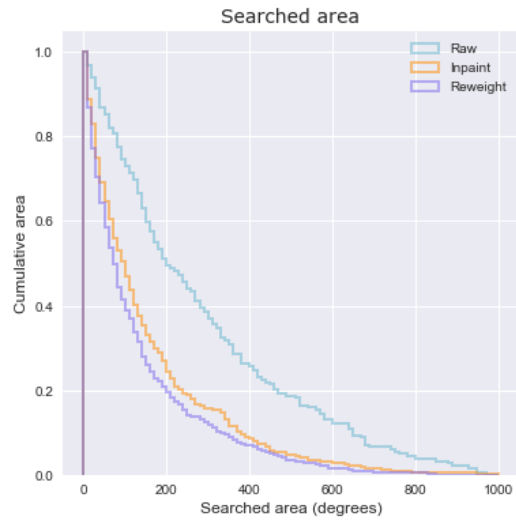

FIG. 7. Histogram showing the total searched area with a glitch by BAYESTAR in degrees vs. the cumulative sum of signals. The raw data in blue shows the glitch impairs skymap accuracy. Inpainting a hole brings the searched area closer to the ideal distribution and and reweighting does slightly better.

The searched area plot in Figure 7 displays a similar behavior. We see that a glitch biases the accuracy of the skymap in the raw data, and it is recovered best by reweighting.

The area of the 90 percent credible region shown in Figure 8 shows the inpainted and reweighted histograms agree with each other. We can determine from this that the precision of our measurements stay the same after the reweighting method is applied. For the raw data with a glitch however, the 90 percent credible area is significantly biased. Not only does a simulated glitch impair accuracy, but the precision deviates as well.

IV. DISCUSSION

One important takeaway is that we created a glitch that was able to bias the sky localization of a simulated source. The sine-gaussian that we injected into the signal data is not an unlikely model for real glitches observed in the LIGO detectors. We can conclude from these results that some form of mitigation during low-latency is necessary for accurate sky localization of real sources.

In addition to showing a glitch was able to bias a skymap, data without a glitch reveals the inpainting process contributes its own bias. This was a noticeable effect even with a gate of 64 ms (less than a tenth of a second) that we used on the tests shown previously. Other

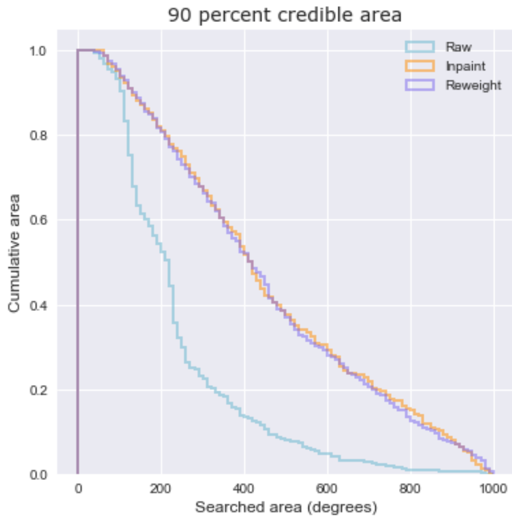

FIG. 8. Histogram showing the area of the 90 percent credible region on the skymap, which we use to roughly estimate the precision of our measurement. Because we are not actually changing the SNR timeseries and reweighting the PSD to make sure we account for the loss of sensitivity, the curves on this graph should line up with each other. The inpainted and reweighted areas line up, but the glitch we created induces a bias on this measurement and affects the overall precision.

tests we conduced show the same bias for larger gate widths that could be necessary for certain types of observed glitches such as slow-scattering. In addition to the localization bias, we see from Figure 8 that inpainting alone results in a larger fraction of real GW signals becoming undetectable than if we use reweighting. If the raw data has a glitch, inpainting does recover some accuracy. However, PySLIDE improves upon it significantly with a relatively simple calculation.

In further observing runs we expect the sensitivity of the detectors to increase and to detect more events. This means it is more likely there will be instances of slow-scattering and other long-lasting glitch types overlapping signals from BNS mergers. The method presented here is one of the few in active development that can handle these types of noise transients, and could be useful when long gates are necessary.

V. CONCLUSION

For upcoming LIGO observing runs, it is imperative that we have a way to to mitigate instrumental artifacts in the detector instantaneously. Quick and reliable sky localization of gravitational wave signals allows us to expand the field of multi-messenger astrophysics.

We demonstrated that glitches and removing segments of a GW signal are sources of bias in sky localization and parameter estimation of a source. From our results we can see that reweighting the SNR timeseries was able to correct for this bias in both cases to return a localization that is more accurate than inpainting alone.

We are working on developing more features for PySLIDE, including expanding options for the noise gate and BAYESTAR plotting. Mainly we will test the method further by exploring a wider parameter space and seeing how our package handles various types of injected signals and glitches. We anticipate that some version of PySLIDE with these expanded features will be made available publicly before the next observing run.

VI. ACKNOWLEDGEMENTS

The authors thank the National Science Foundation for sponsoring this research as part of a LIGO Caltech SURF Program. The authors are grateful for computational resources provided by the LIGO Laboratory and for support by National Science Foundation Grants PHY-0757058 and PHY-0823459. We would also like to thank all of the students and mentors in the Caltech LIGO SURF Program that made this research possible.

[1] B. P. Abbott et al. GW150914: The Advanced LIGO Detectors in the Era of First Discoveries. *Physical Review Letters*, 116(13):131103, April 2016.

[2] Derek Davis et al. LIGO detector characterization in the second and third observing runs. *Class. Quant. Grav.*, 38(13):135014, 2021.

[3] B. P. Abbott et al. Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA. *Living Reviews in Relativity*, 21(1):3, April 2018.

[4] B. P. Abbott et al. Gw170817: Observation of gravitational waves from a binary neutron star inspiral. *Phys. Rev. Lett.*, 119:161101, Oct 2017.

[5] Miriam Cabero, Ashish Mahabal, and Jess McIver. GWSkyNet: a real-time classifier for public gravitational-wave candidates. *Astrophys. J. Lett.*, 904(1):L9, 2020.

[6] Salvatore Vitale, Reed Essick, Erik Katsavounidis, Sergey Klimenko, and Gabriele Vedovato. On similarity of binary black hole gravitational-wave skymaps: to observe or to wait? *Mon. Not. Roy. Astron. Soc.*, 466:L78, 2017.

[7] B. P. Abbott et al. Low-latency Gravitational-wave Alerts for Multimessenger Astronomy during the Second Advanced LIGO and Virgo Observing Run. *Astrophys. J.*, 875(2):161, 2019.

[8] B. P. Abbott et al. Multi-messenger Observations of a Binary Neutron Star Merger. *Astrophys. J. Lett.*, 848(2):L12, 2017.

[9] Michael W. Coughlin et al. Optimizing searches for electromagnetic counterparts of gravitational wave triggers. *Mon. Not. Roy. Astron. Soc.*, 478(1):692–702, 2018.

[10] Michael W. Coughlin et al. Optimizing multitelescope observations of gravitational-wave counterparts. *Mon. Not. Roy. Astron. Soc.*, 489(4):5775–5783, 2019.

[11] Chris Pankow et al. Mitigation of the instrumental noise transient in gravitational-wave data surrounding GW170817. *Phys. Rev. D*, 98(8):084016, 2018.

[12] Neil J. Cornish, Tyson B. Littenberg, Bence Bécsy, Katerina Chatziioannou, James A. Clark, Sudarshan Ghonge, and Margaret Millhouse. BayesWave analysis pipeline in the era of gravitational wave observations. *Phys. Rev. D*, 103(4):044006, 2021.

[13] Kentaro Mogushi, Ryan Quitzow-James, Marco Cavaglià, Sumeet Kulkarni, and Fergus Hayes. NNETFIX: an artificial neural network-based denoising engine for gravitational-wave signals. *Mach. Learn. Sci. Tech.*, 2(3):035018, 2021.

[14] Barak Zackay, Tejaswi Venumadhav, Javier Roulet, Liang Dai, and Matias Zaldarriaga. Detecting Gravitational Waves in Data with Non-Gaussian Noise. *arXiv e-prints*, page arXiv:1908.05644, August 2019.

[15] Leo P. Singer and Larry R. Price. Rapid bayesian position reconstruction for gravitational-wave transients. *Physical Review D*, 93(2), Jan 2016.

[16] Samantha A. Usman et al. The PyCBC search for gravitational waves from compact binary coalescence. *Class. Quant. Grav.*, 33(21):215004, 2016.

[17] Duncan et al. Macleod. Gwpy. `https://github.com/gwpy/gwpy`, 2019.

[18] James Borbeau. Pycondor. `https://github.com/jrbourbeau/pycondor`, 2018.

[19] Rico K. L. Lo. Skymap overlap. `https://git.ligo.org/ka-lok.lo/skymap-overlap/`, 2021.

[20] Alejandro Bohé et al. Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors. *Phys. Rev. D*, 95(4):044028, 2017.