

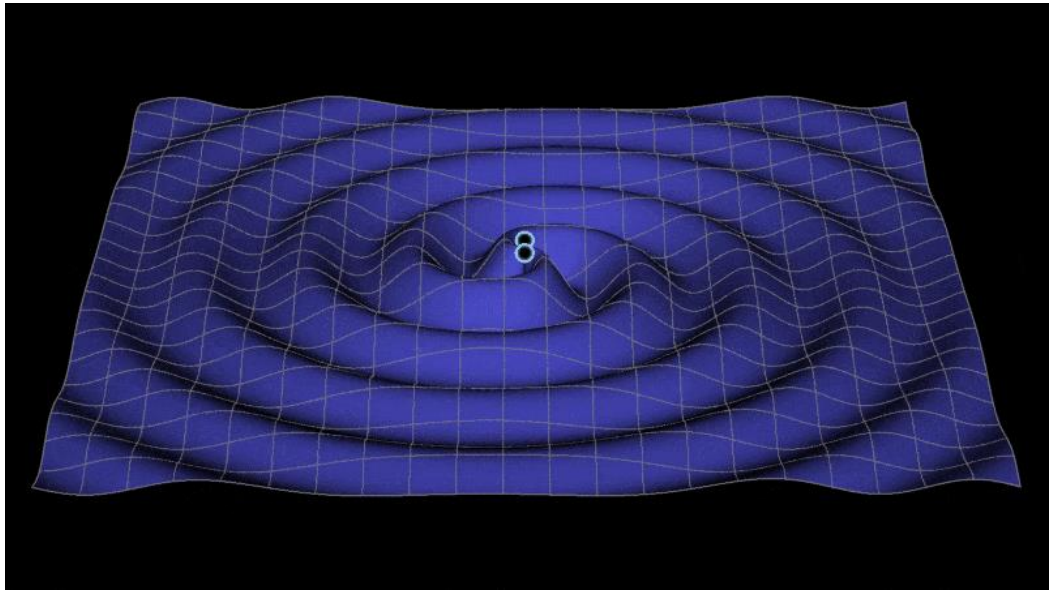
DeepExtractor

Model-Agnostic Signal and Glitch Reconstruction for Gravitational-Wave Astronomy

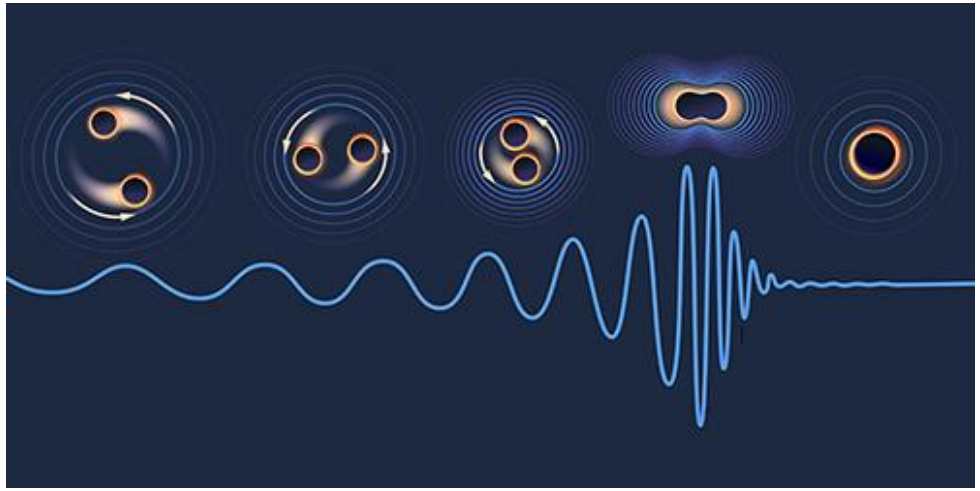
Tom Dooney · SISSA · 16 October 2026



What are Gravitational Waves (GWs)?

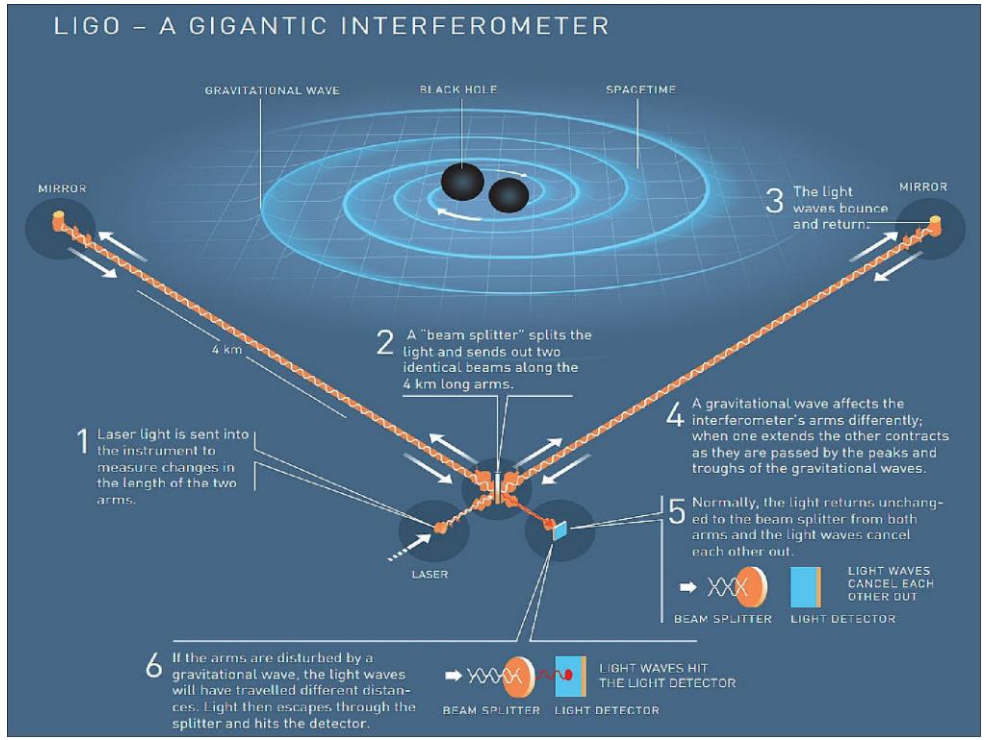


2D simulation of binary black holes emitting gravitational waves.
Wolfram Community



1D time-domain representation of a binary black hole merger.
Abbott et al. (2016), PRL

How detectors ‘hear’ Gravitational Waves



A schematic for LIGO
ESA Earth Online Portal

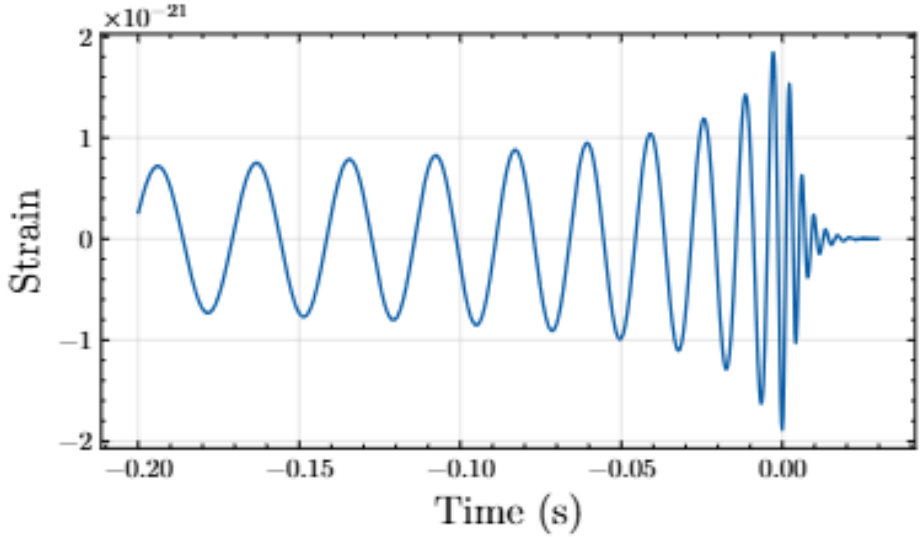


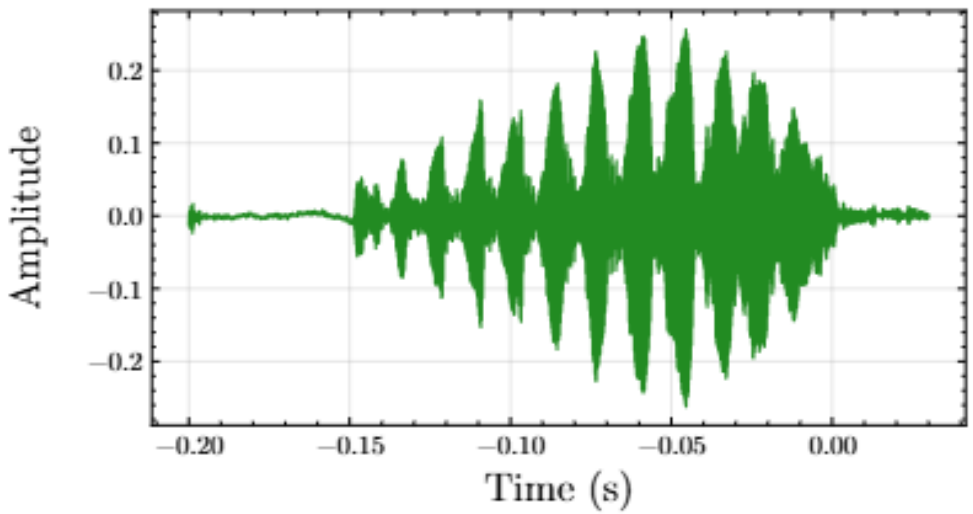
LIGO Livingston, Louisiana
LIGO Caltech

What we learn from Gravitational Waves

- **Rates and populations:** population of binary systems, rates of mergers – stellar evolution
- **Tests of General Relativity:** Was Einstein completely correct?
- **Extreme matter and multi-messenger astronomy:** Binary neutron star mergers, most extreme matter, not well understood currently, GWs + light (explosion)
- **Cosmology:** can measure distance using GWs, can measure ‘red-shift’ – can constrain the expansion of the universe

GW Data vs Digital Sound Data

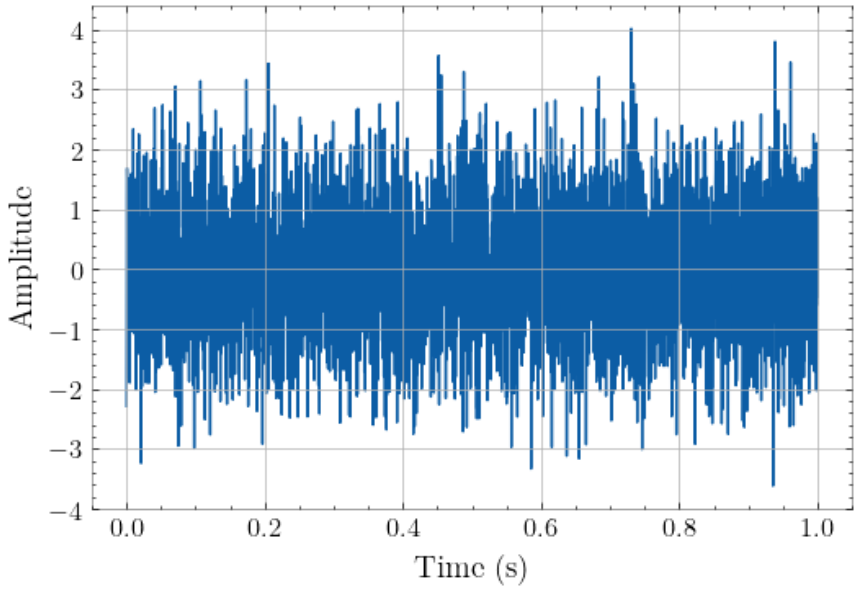


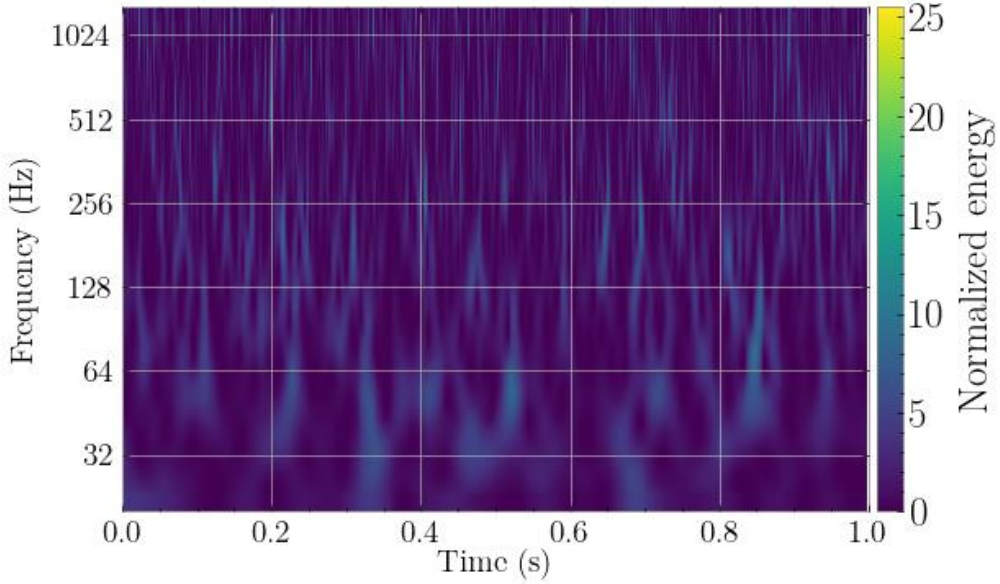


Gravitational Wave Chirp

A Bird Chirp

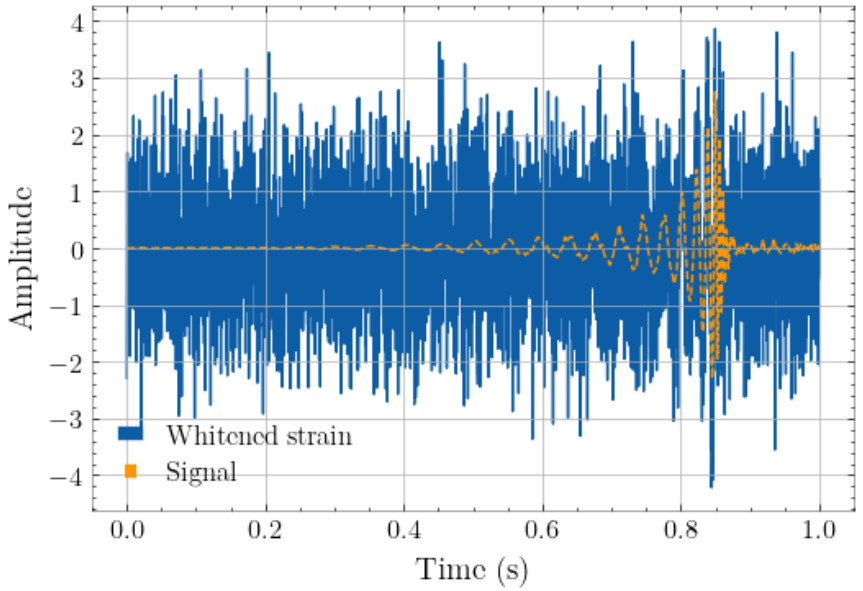
GW Data is noisy

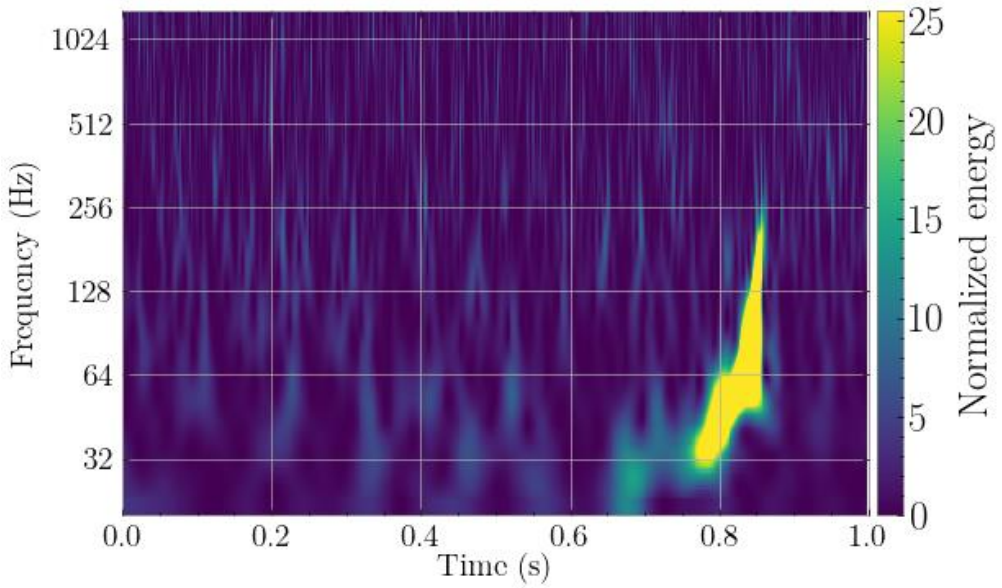




GW detector data
(Noise only)

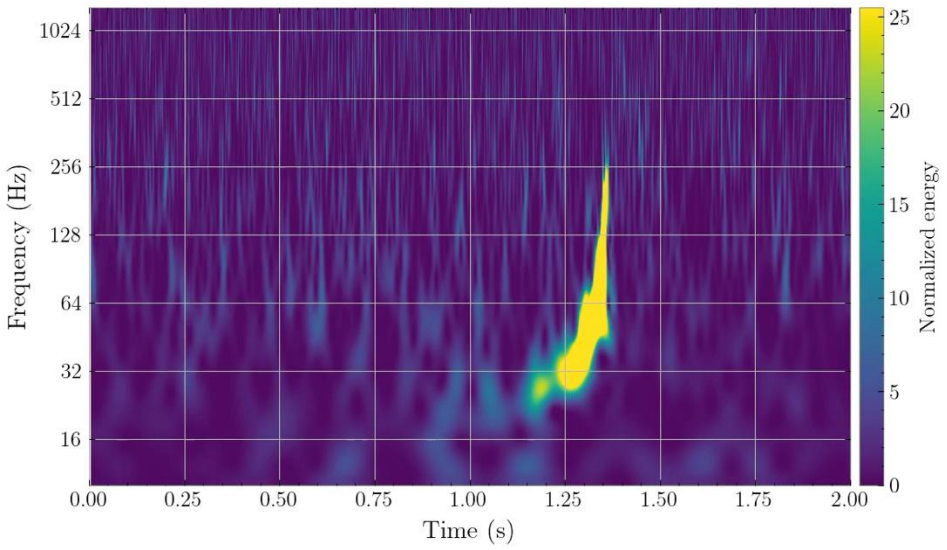
Gravitational Wave in Time-Frequency



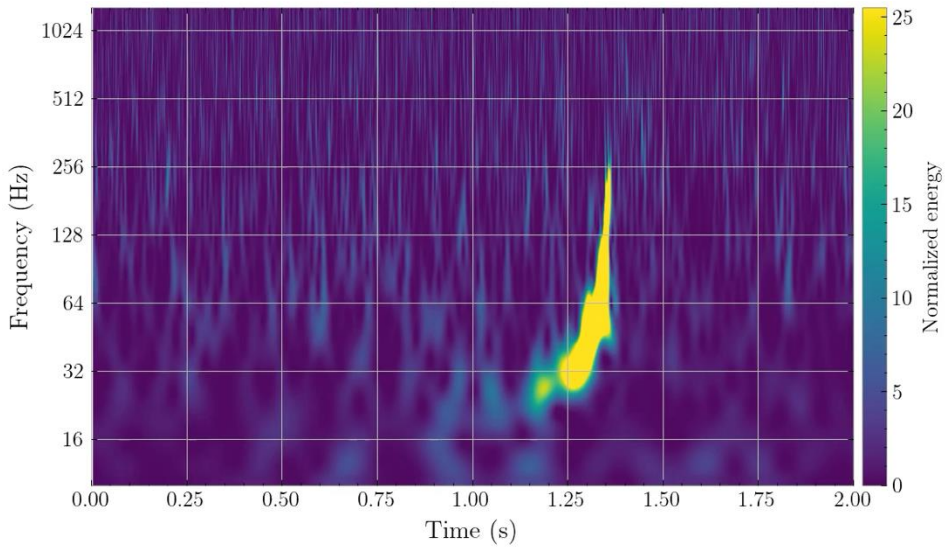


GW detector data
(Noise+Signal)

The Sound of GW Music

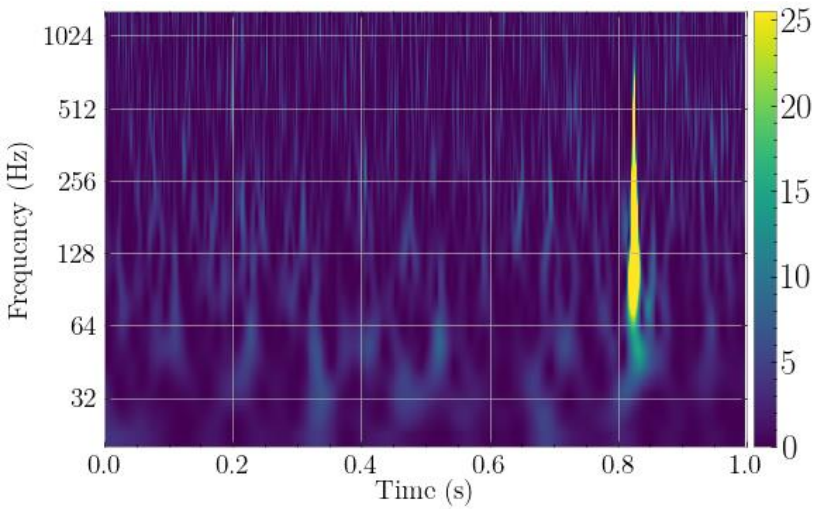


Noise + Signal

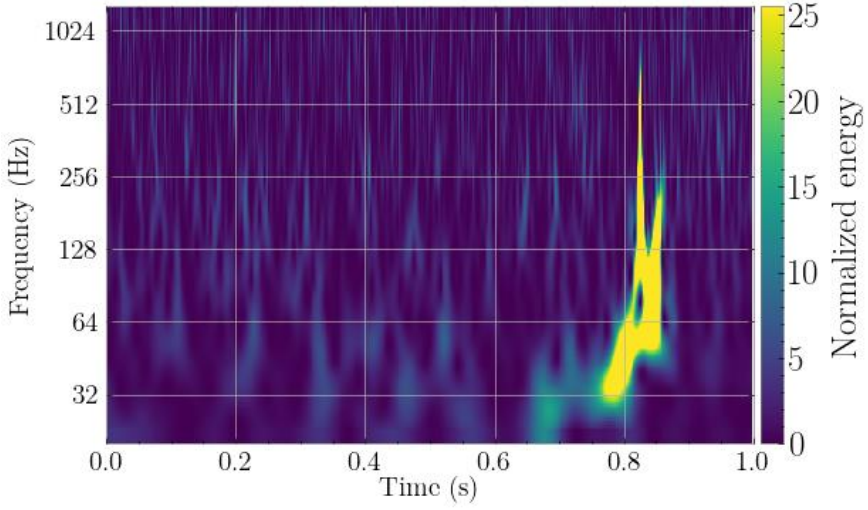


Isolated Signal

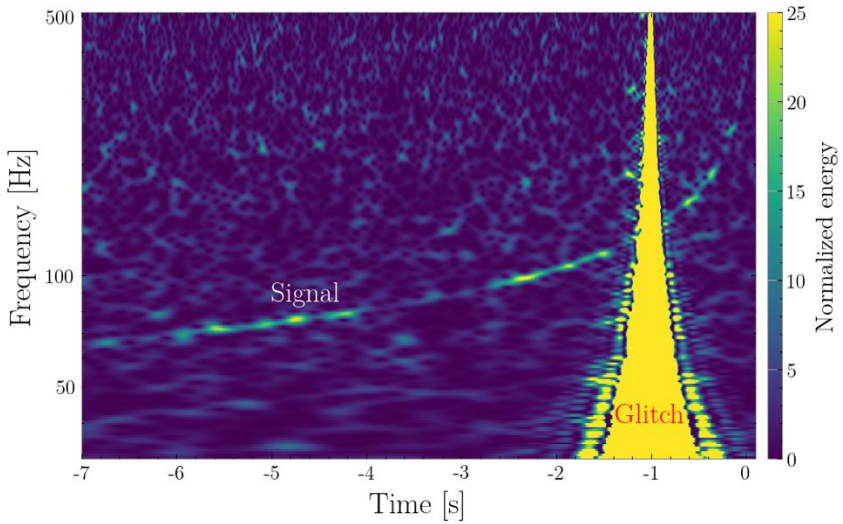
Glitches in the Matrix



Noise + Glitch

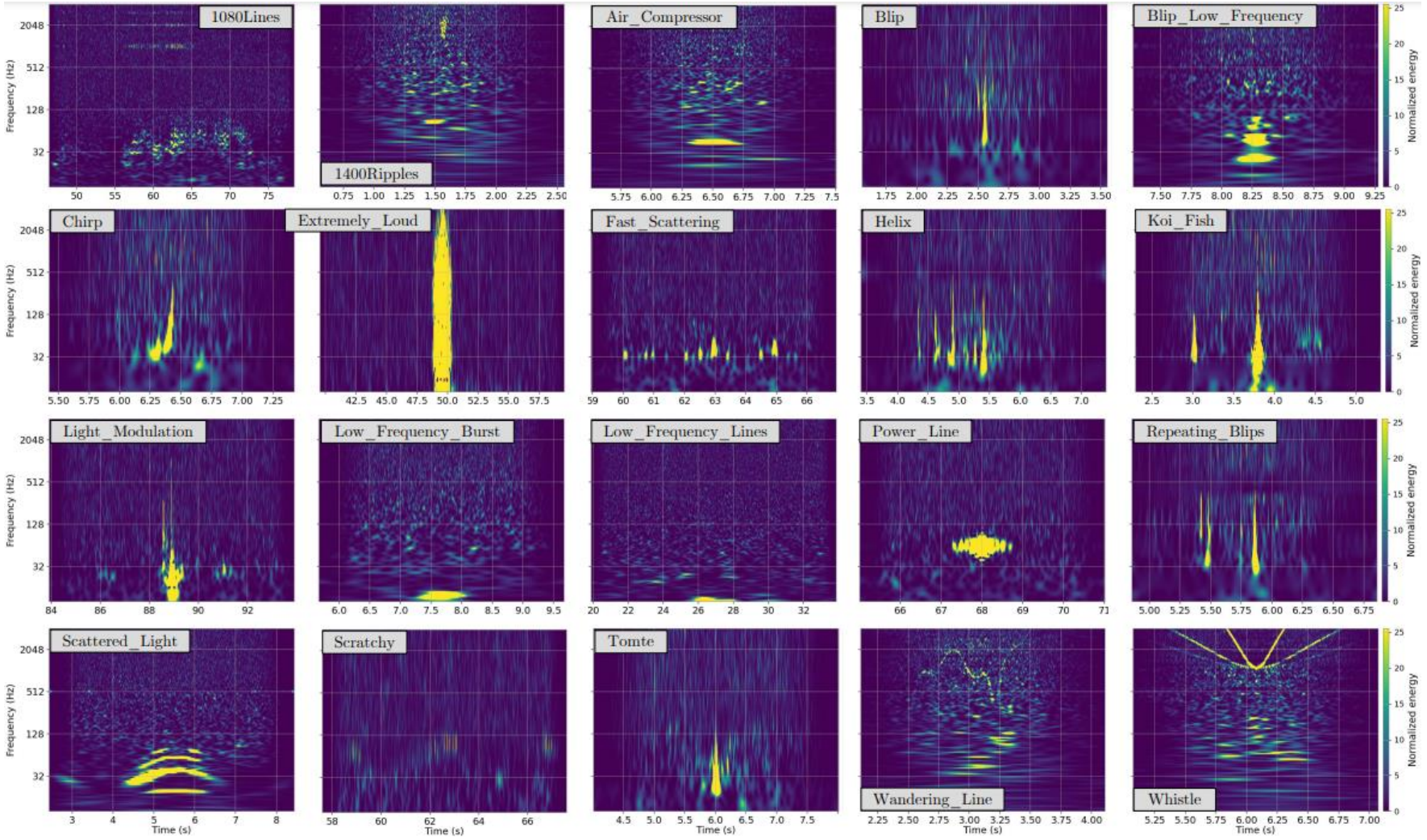


Noise + Signal+ Glitch

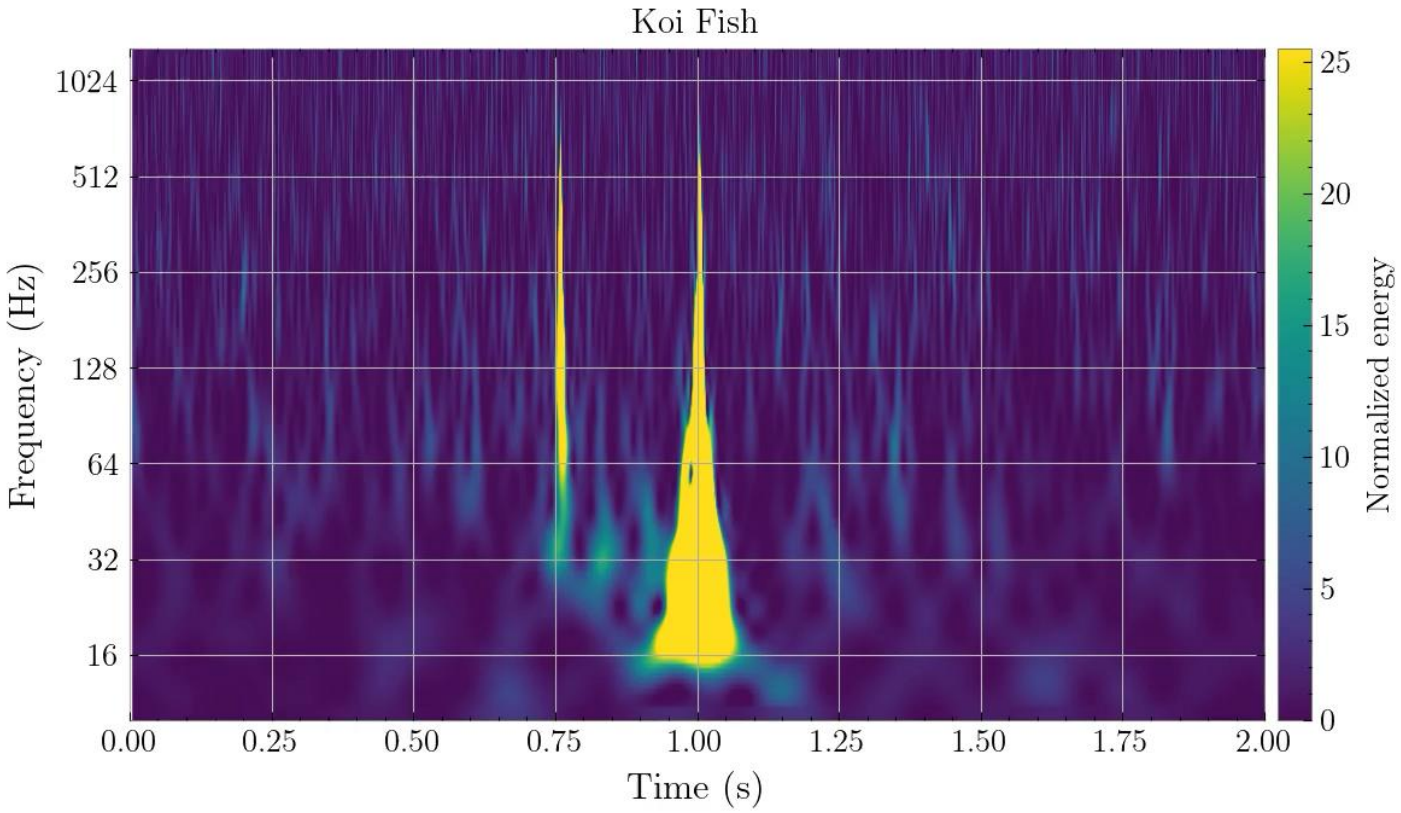


GW170817

LIGO-Virgo Glitches (Gravity Spy)

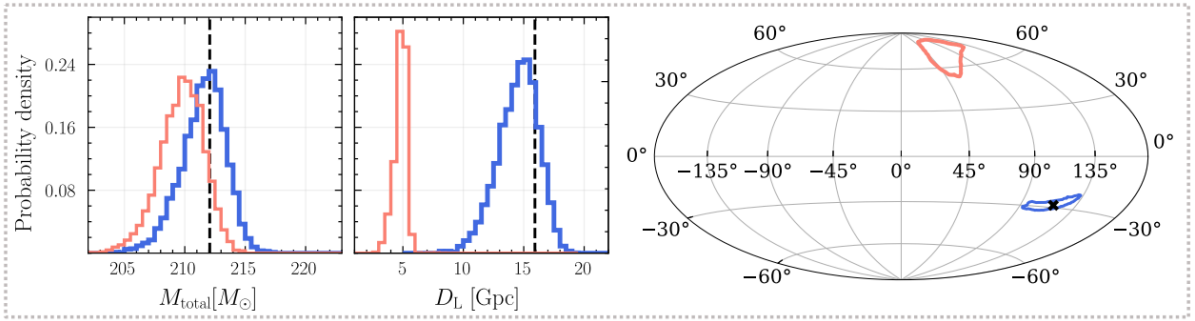


The Sound of a Glitch



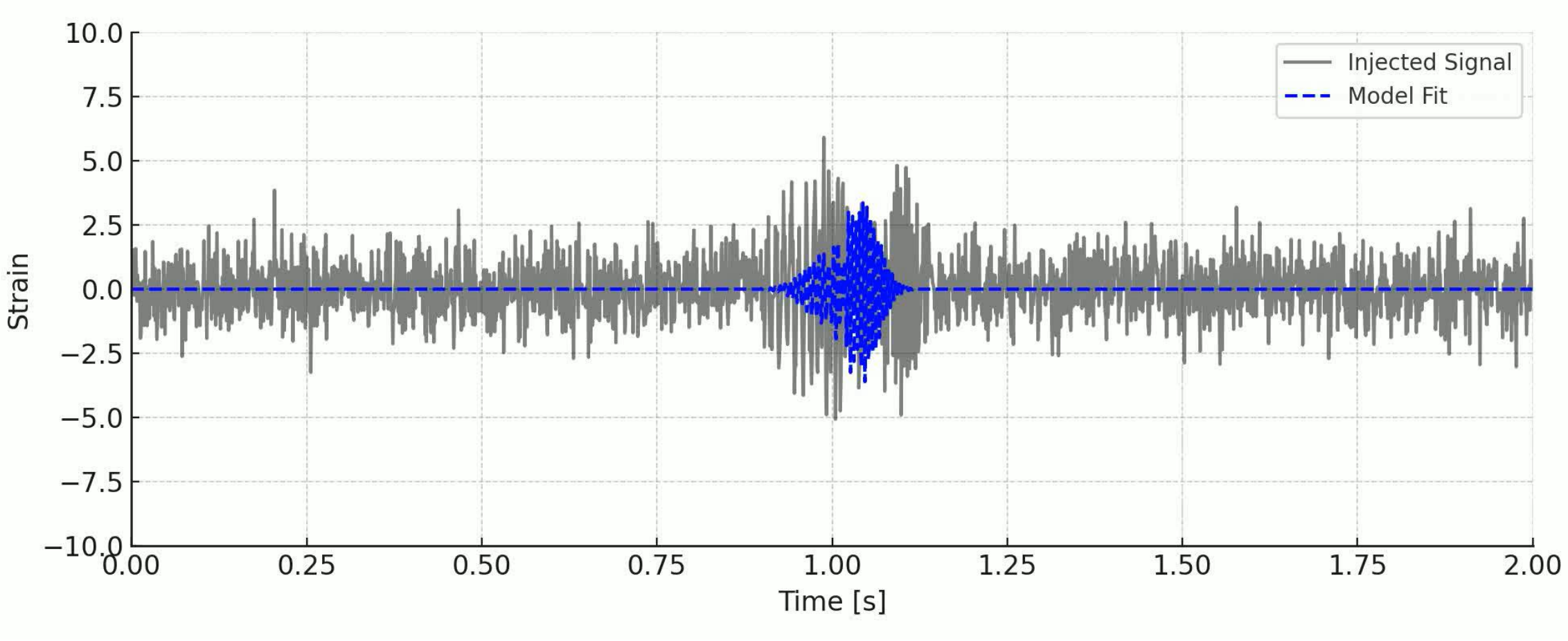
So What?

- Glitches hinder GW data analysis
  - Increase false positives
  - Bias Parameter Estimation
- BayesWave can mitigate them
  - Effective for some glitch classes but costly
  - Approximately 1 CPU hour to model one glitch
  - Not scalable towards next-gen (Einstein Telescope)
- We aim to build a fast glitch reconstruction framework
  - Mock Data Challenges
  - Pipeline validation
- We will also discuss glitch mitigation and searches



Glitch bias in parameter estimation (10.1103/l6tp-ykxp)

Toy example of BayesWave

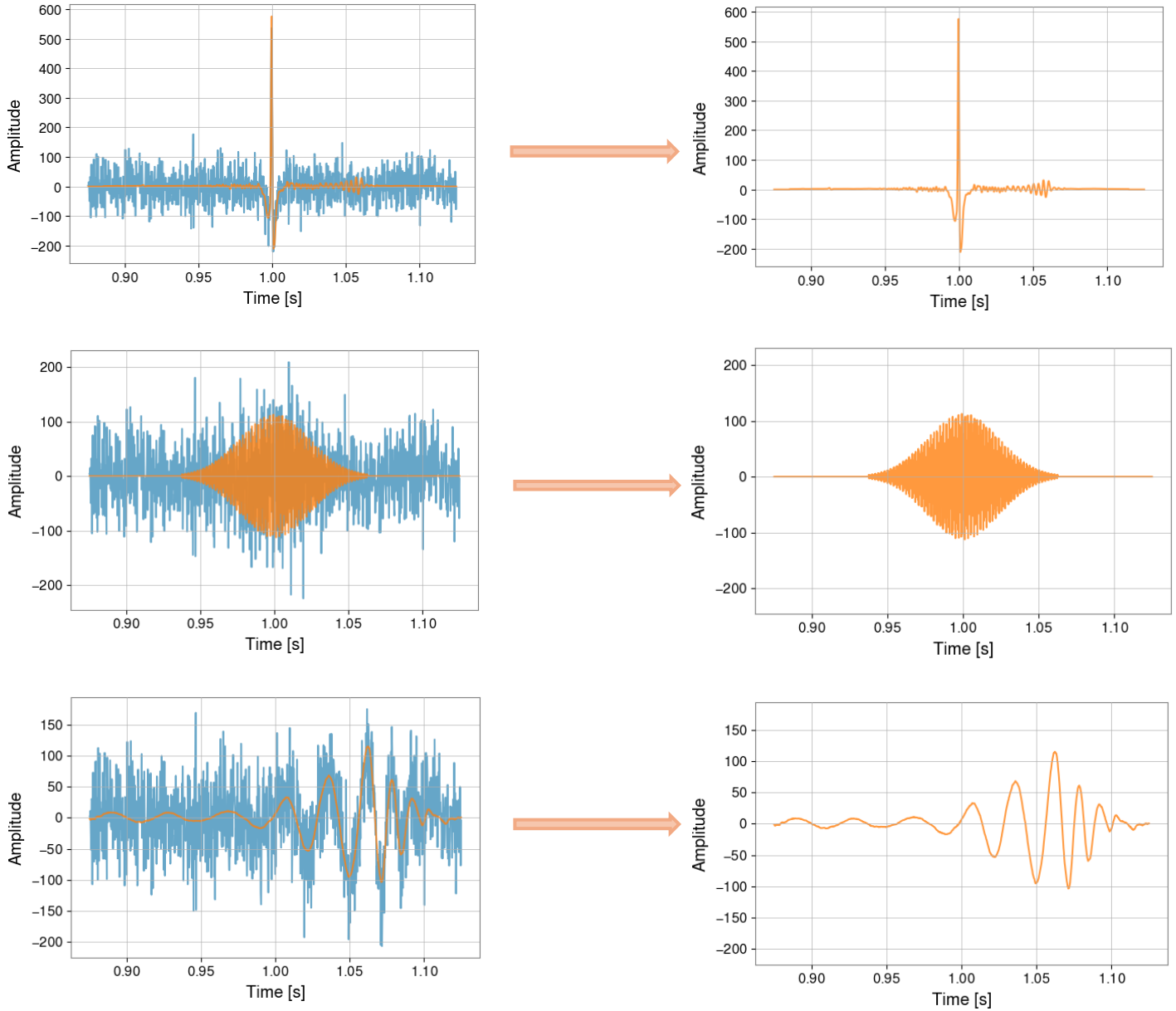


BayesWave uses **Reversible Jump MCMC** to:

1. find the **number** of these **wavelets**
2. their **parameters**

Goal

- Fast and accurate time-domain reconstruction
- Reconstruct any power excess above Gaussian background
- Agnostic of source and morphology
- Compete with BayesWave accuracy and reduce inference time

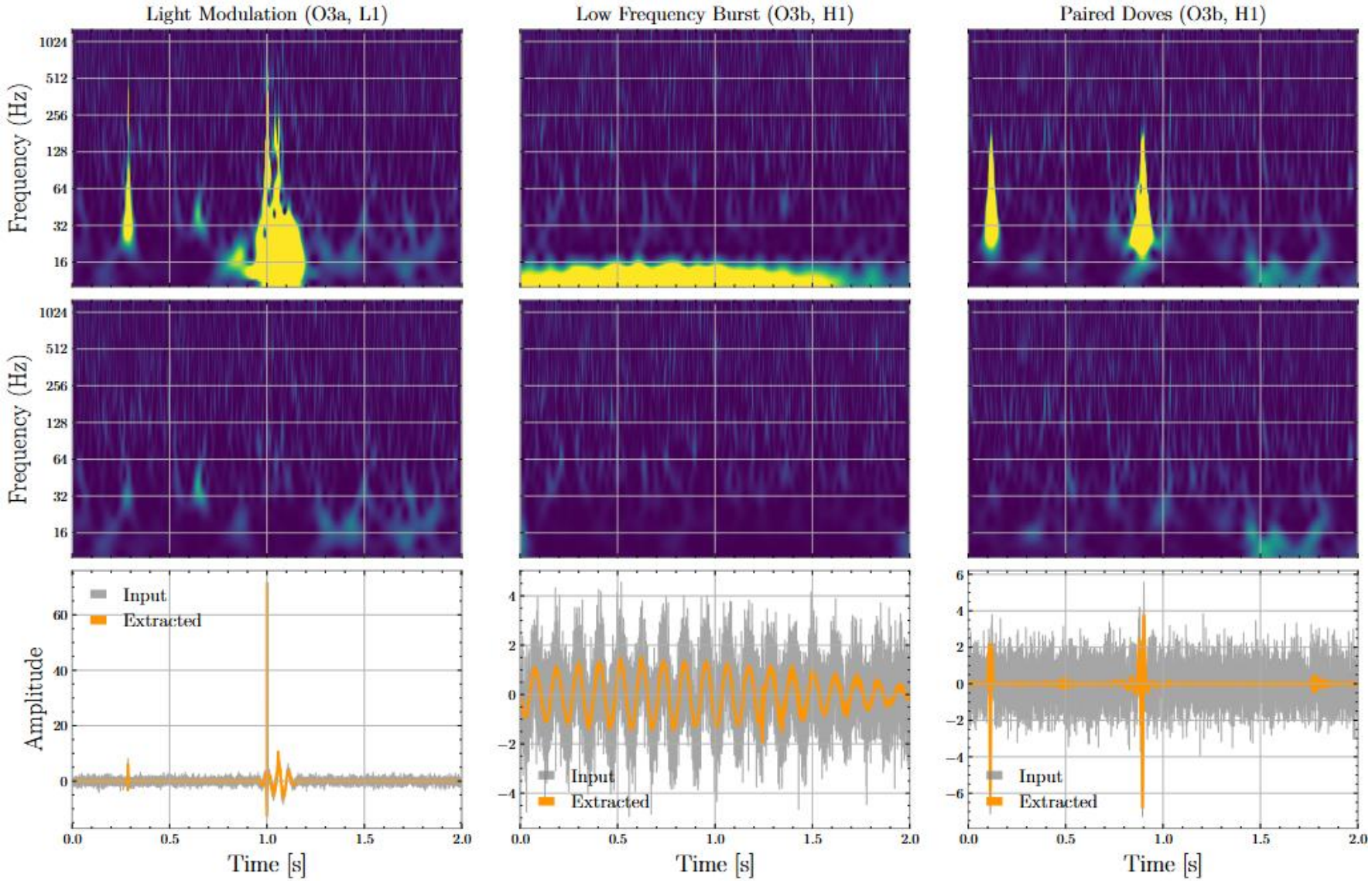


LIGO glitch reconstructions with DeepExtractor

Input

Subtracted

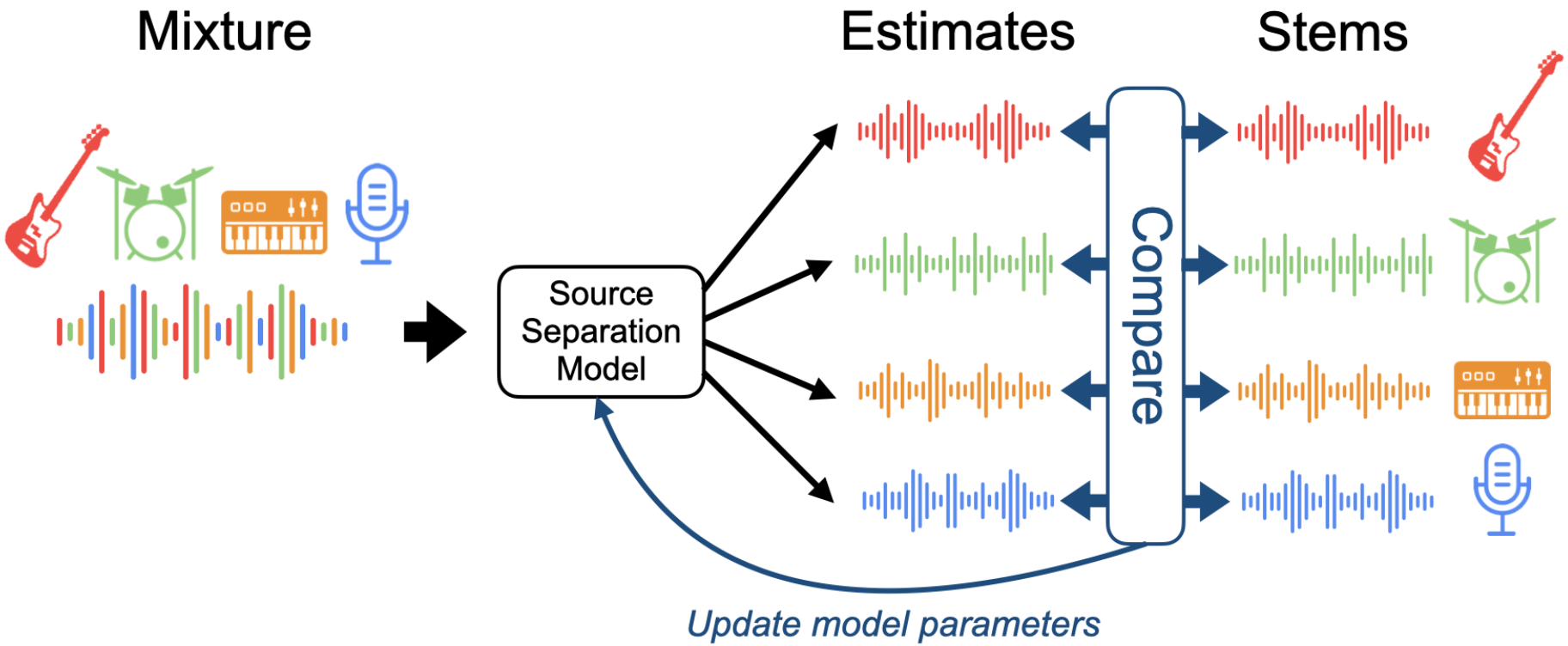
Reconstruction



DeepExtractor Applications

1. Model-Agnostic Signal/Glitch Reconstruction
2. Signal + Glitch Separation (Glitch Mitigation)
3. Template Free Searches

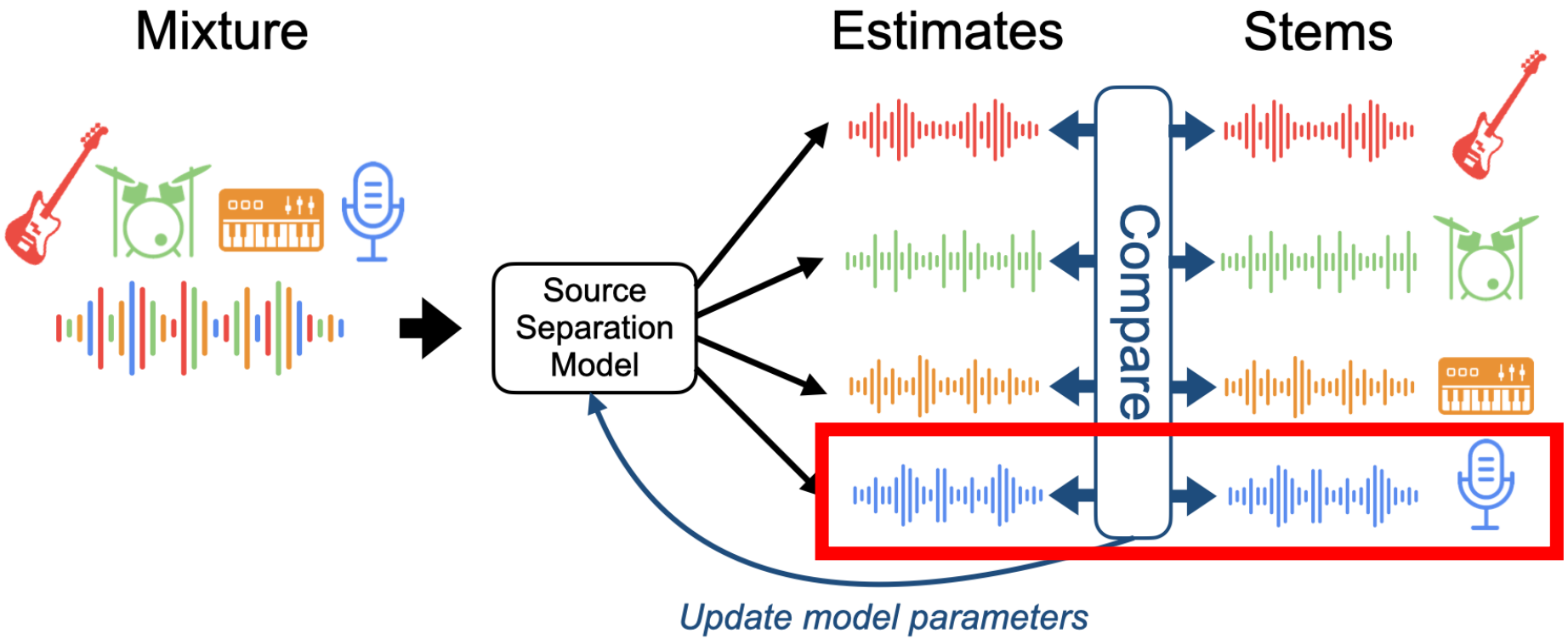
Borrowing from the music industry



Music source separation model

Source: [Source Separation Tutorial]

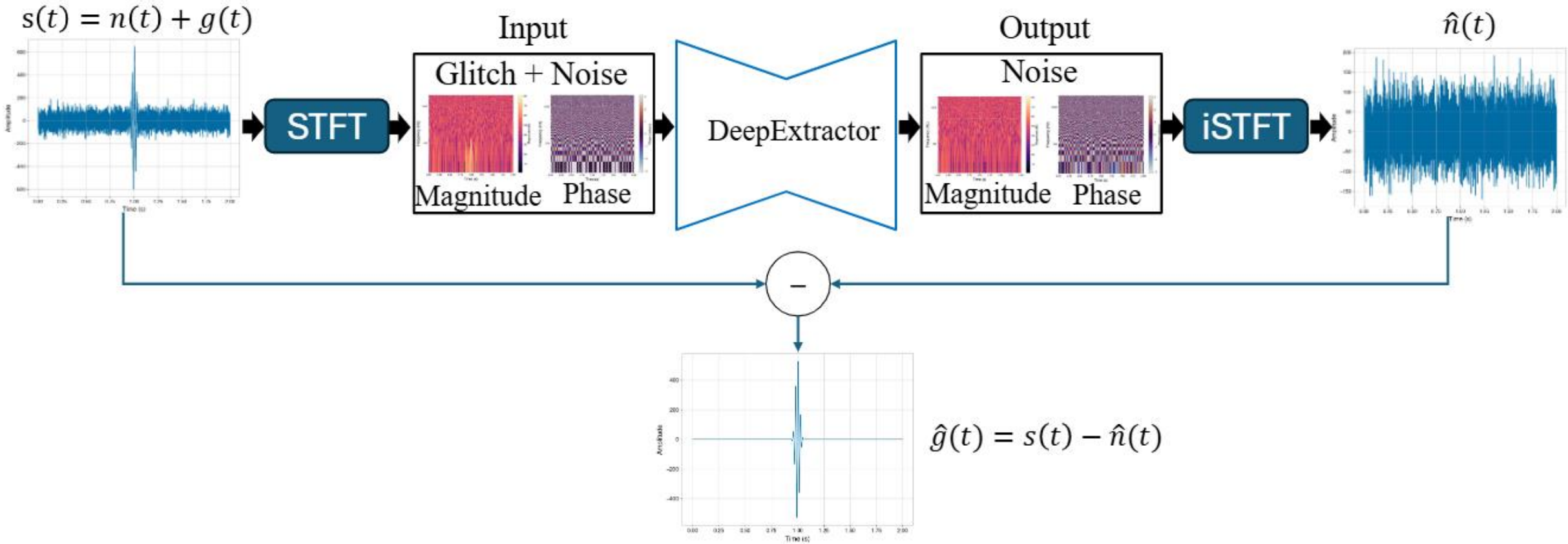
Borrowing from the music industry



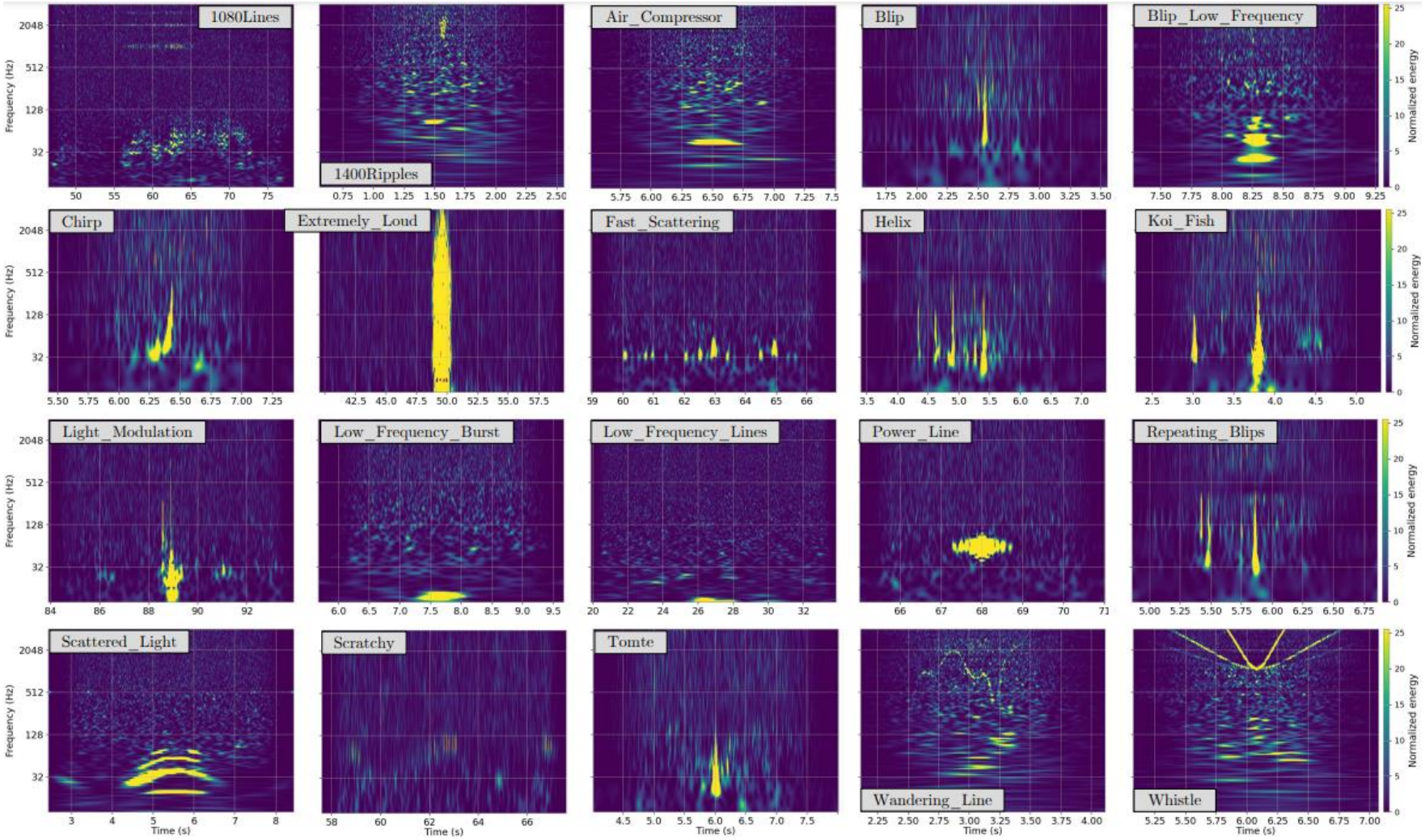
Music source separation model

Source: [Source Separation Tutorial]

DeepExtractor Framework



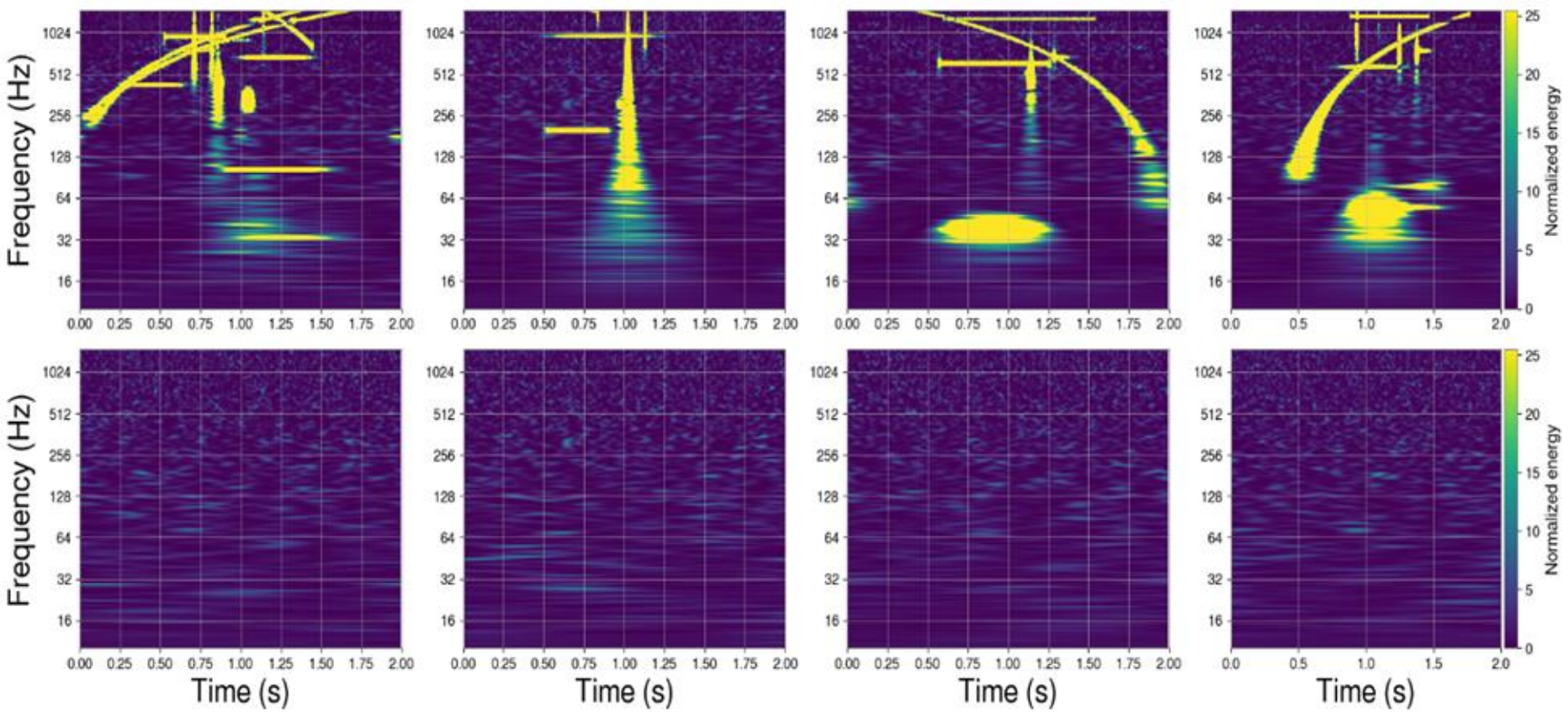
How do we capture all Gravity Spy glitches?



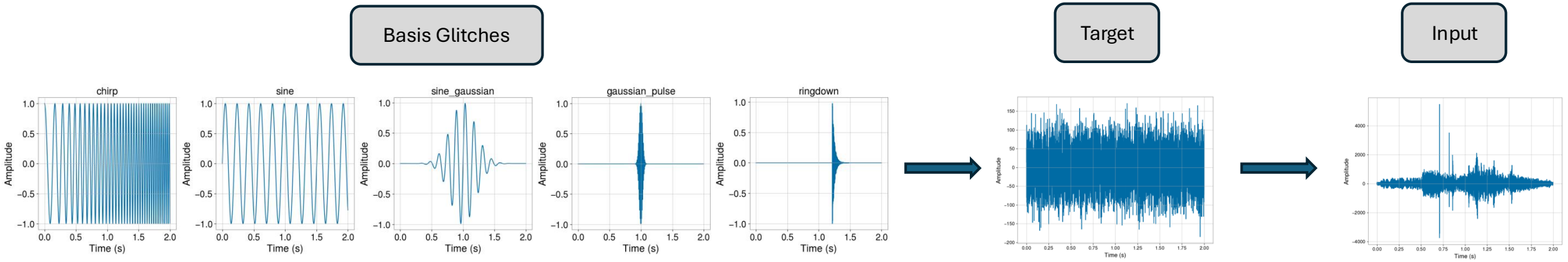
We use proxy glitches…

Input

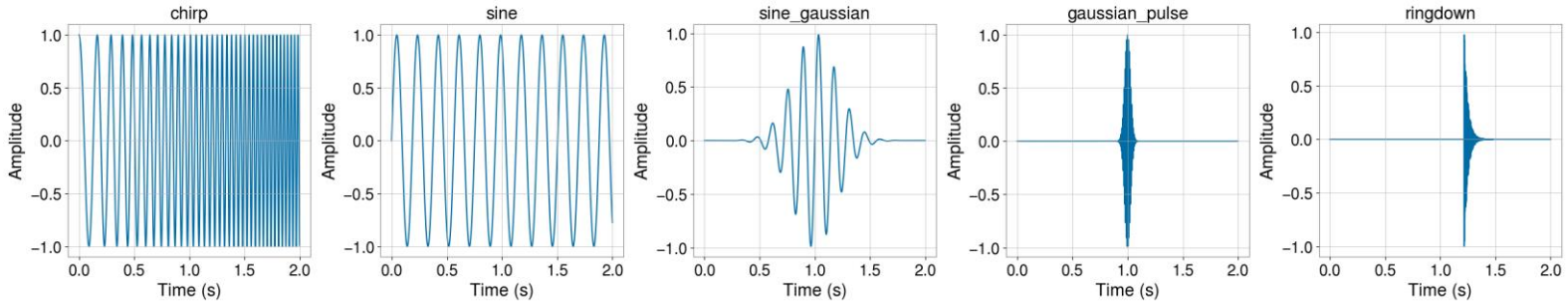
Target



Creating and Injecting Proxy Glitches

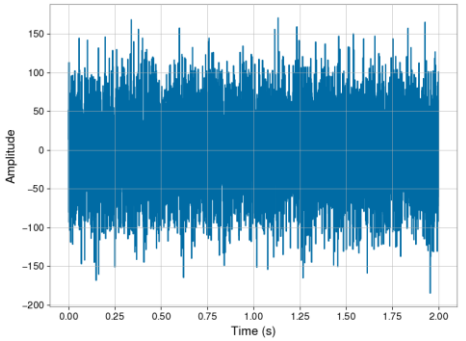


Training Data Generation

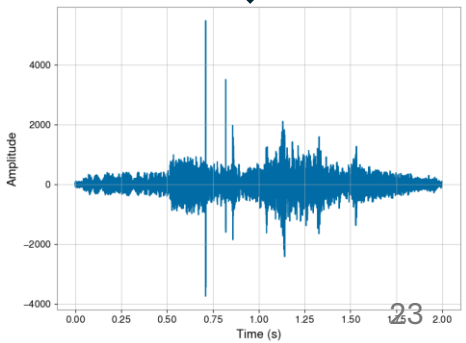


1. We create a diverse dataset using basis waveforms - proxy signals and glitches
2. Accumulate 2-second background samples outside of Omicron triggers
3. Inject (multiple) proxy glitches into each background sample randomly, do this 250,000 times
4. Learning DeepExtractor to always predict the background noise in the presence of these proxy glitches, it can generalize to real glitches

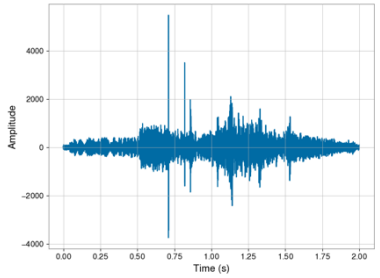
Target



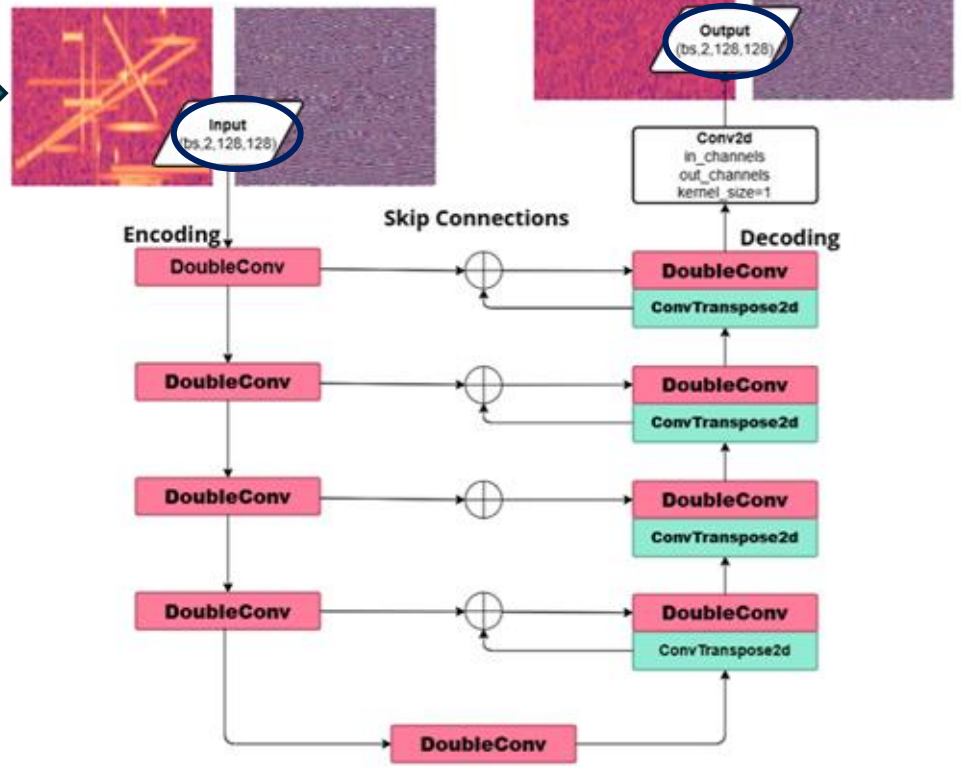
Input



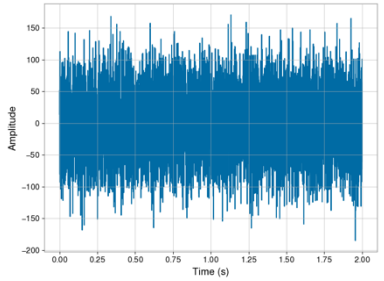
U-Net Architecture

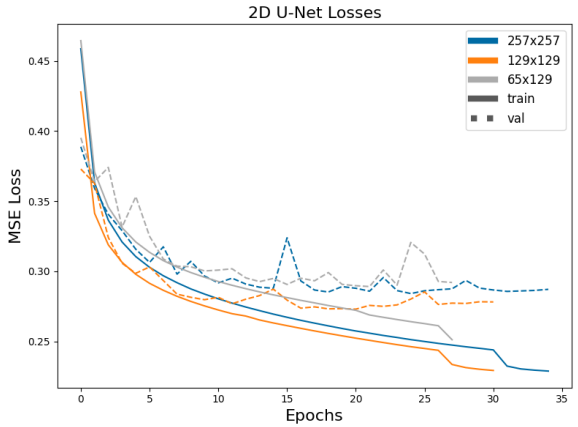


STFT

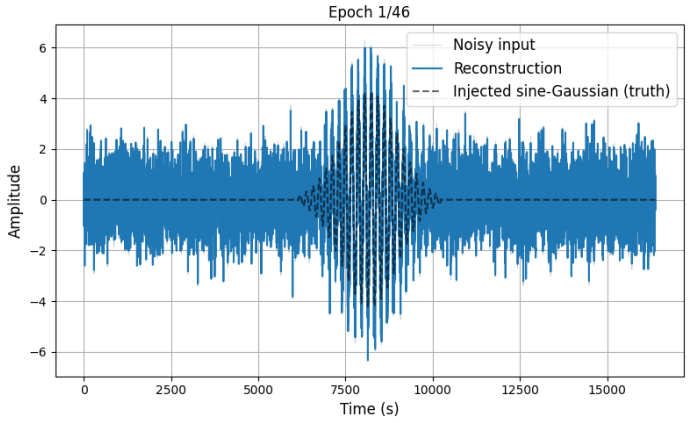


iSTFT



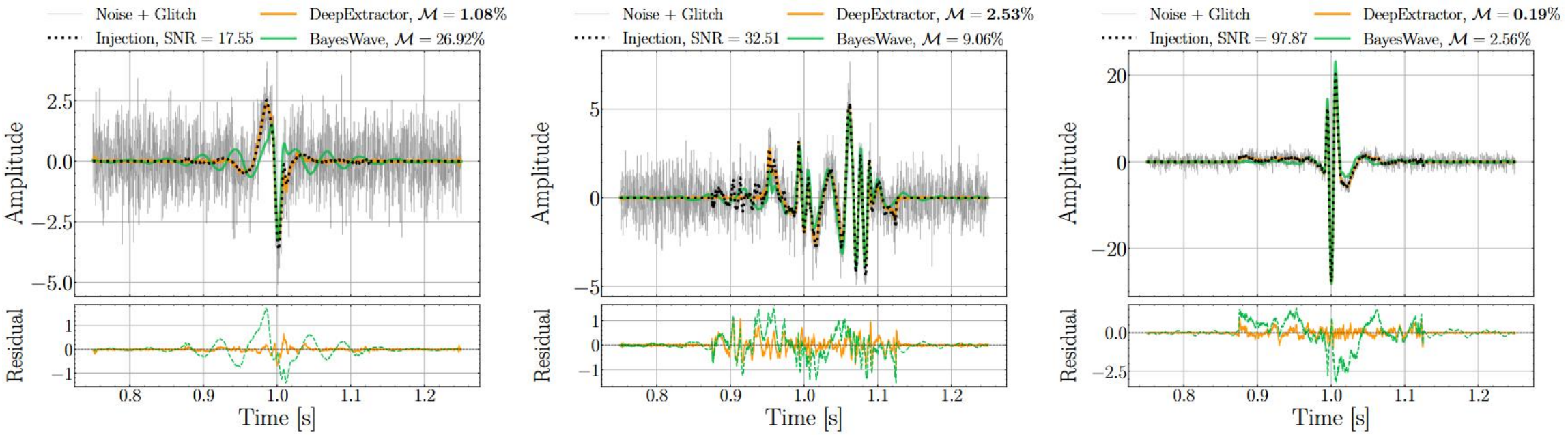


MSE train and validation loss.



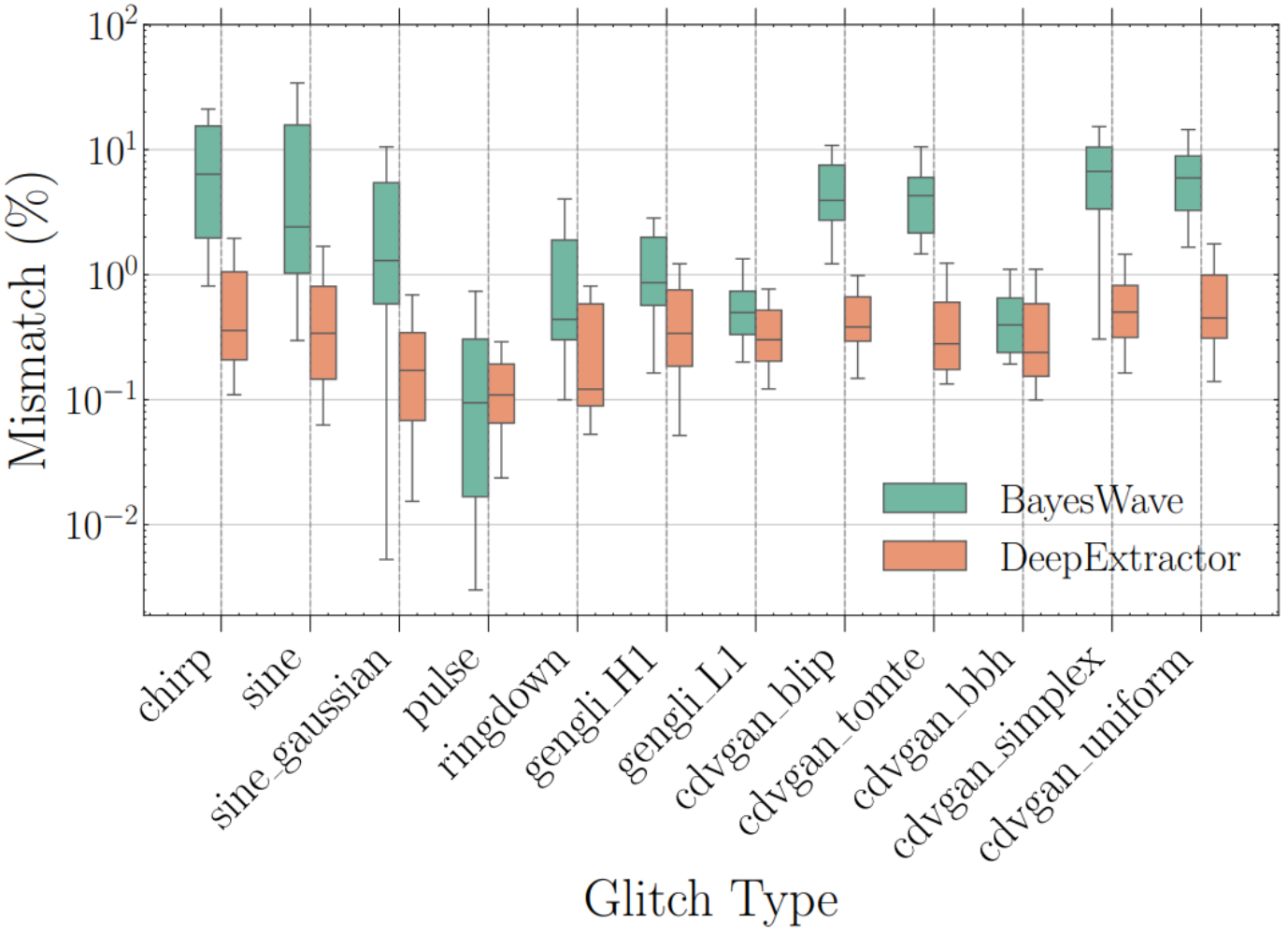
Glitch reconstruction over training epochs

Comparison with BayesWave – Mismatch performance



Example reconstructions with BayesWave and DeepExtractor

Comparison with BayesWave – Mismatch performance

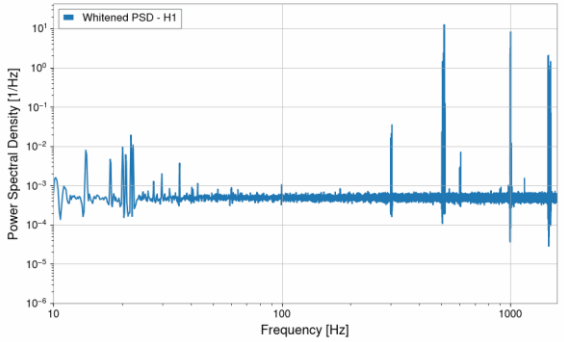


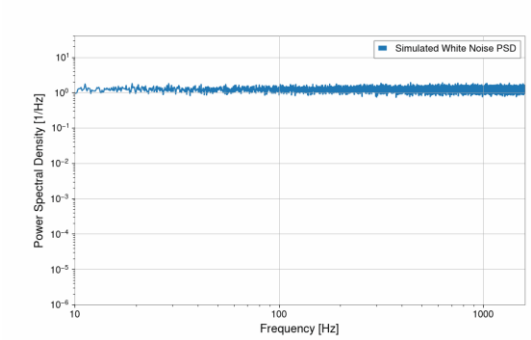
DeepExtractor reconstructs a 2s sample in 0.1s (CPU)

Mismatch between injected and reconstructed samples (30 samples per class)

Transfer learning with real backgrounds

- We accumulate real LIGO backgrounds (H1, L1, O3a, O3b)
  - At least 2s outside of Omicron/signal triggers
  - 45,000 samples of 2s
- Analytical glitch injections (as before)
- Transfer learn our pretrained (simulated) model
  - Further training phase with real backgrounds
  - Additional knowledge about real background





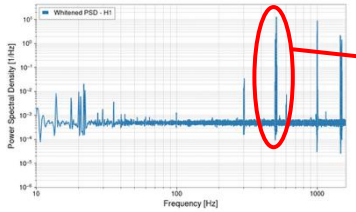
Pre-Trained DeepExtractor

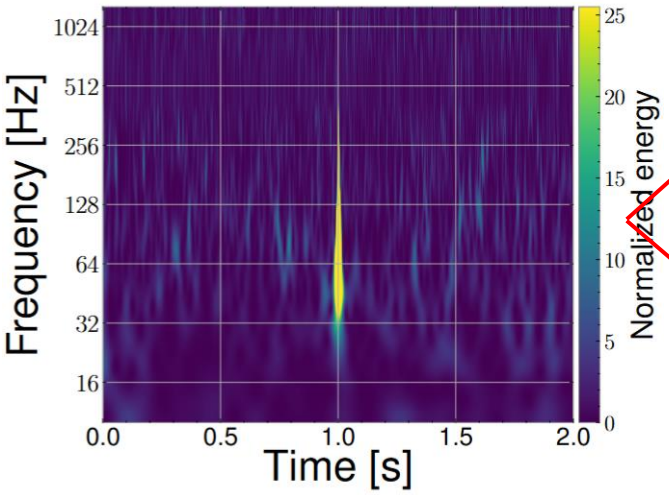
Additional Knowledge (PSD)

Transfer learning

Fine-tuned DeepExtractor

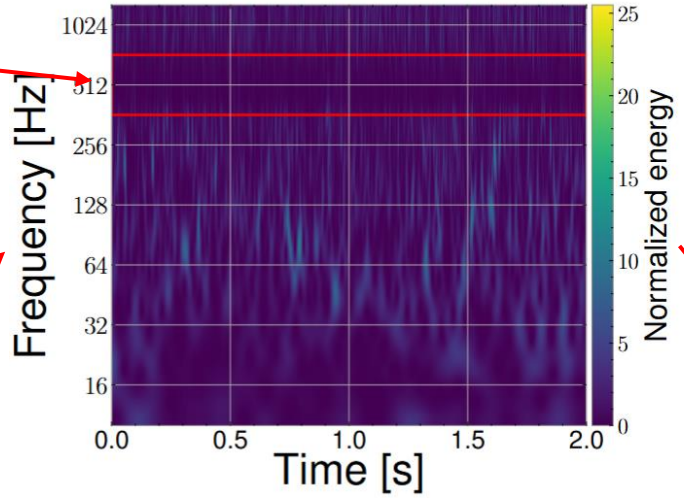
Improvements from Transfer Learning

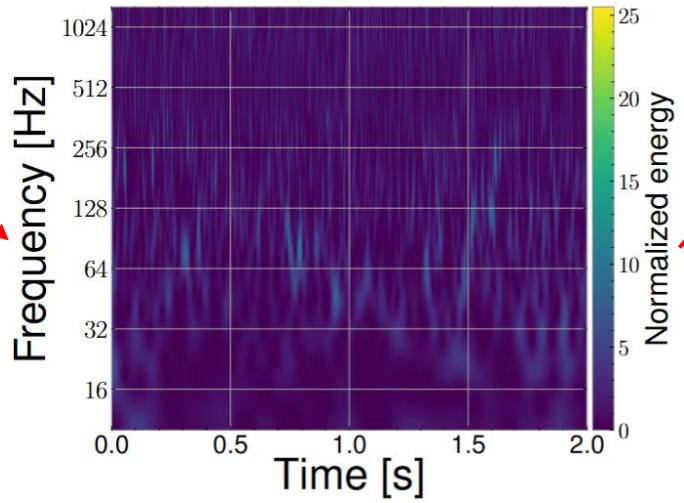


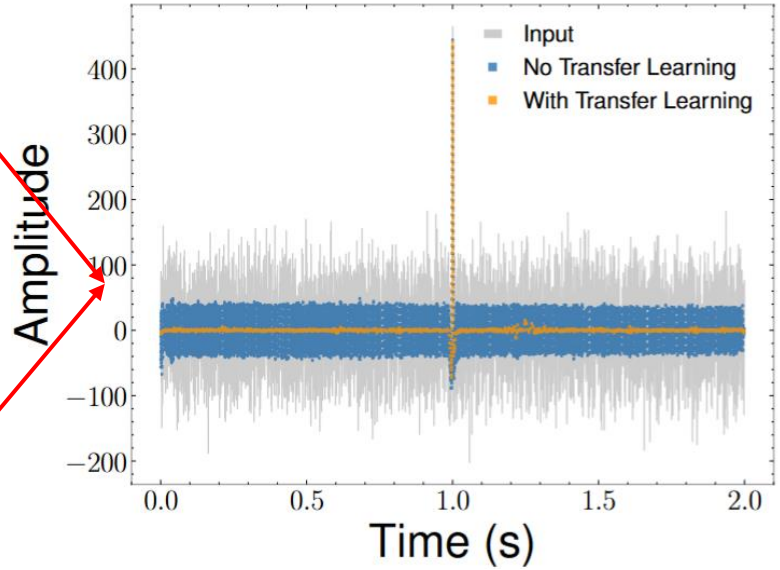


No Transfer Learning

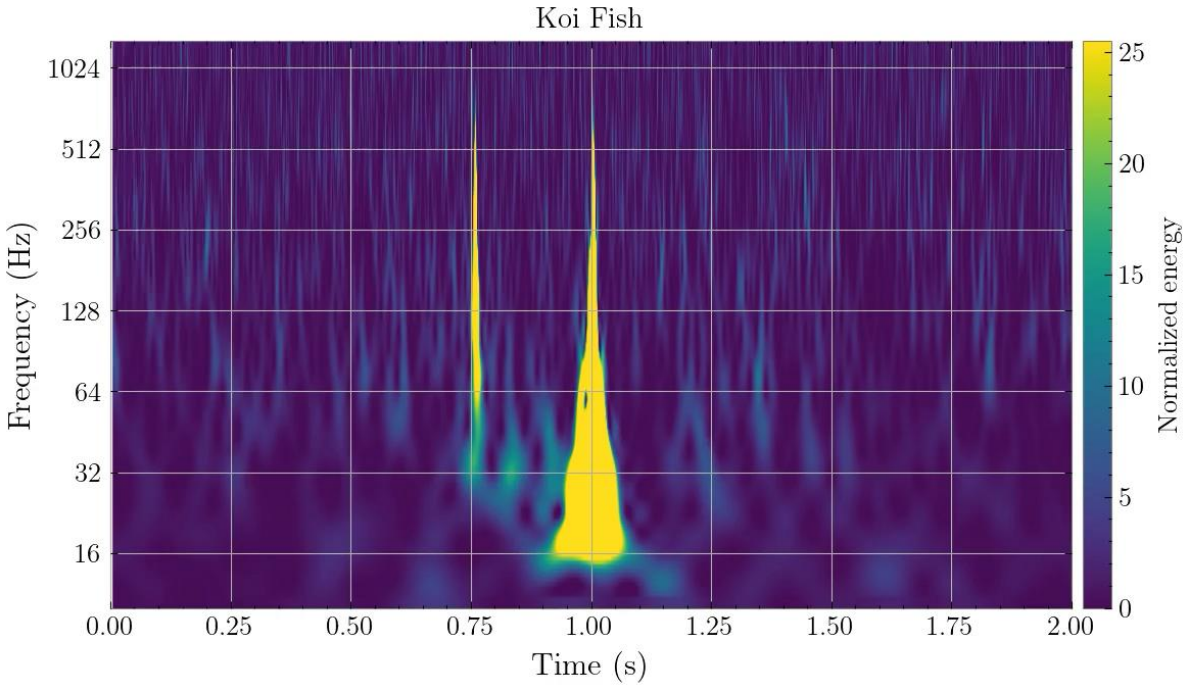
With Transfer Learning

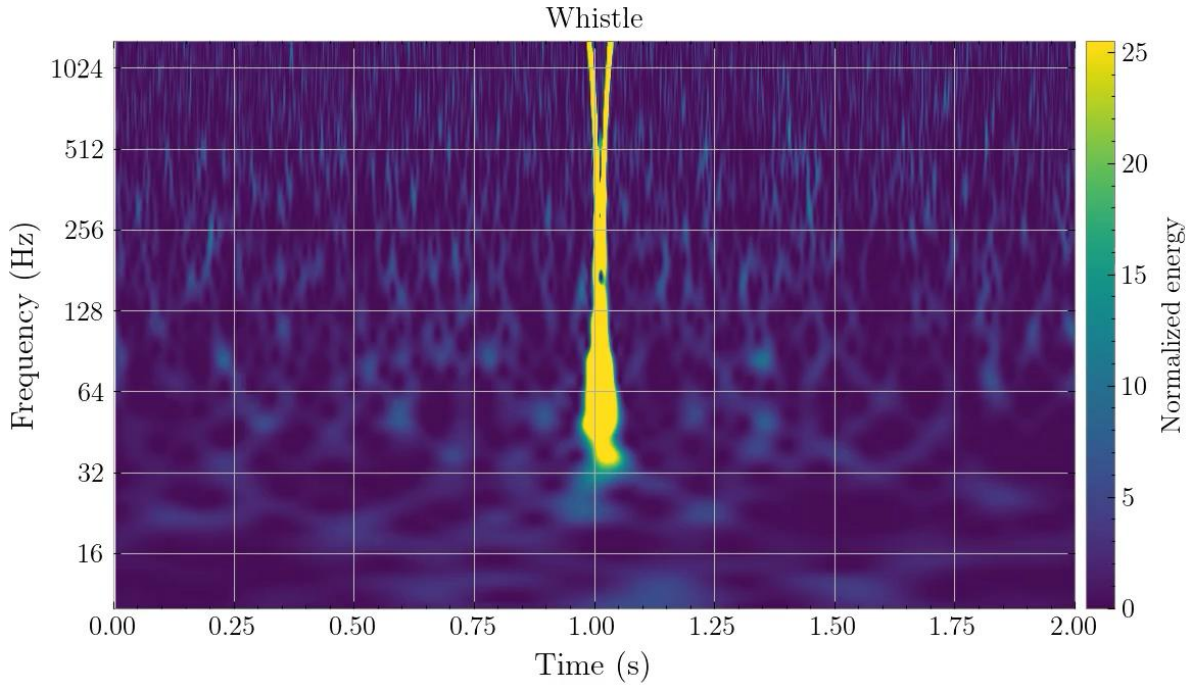






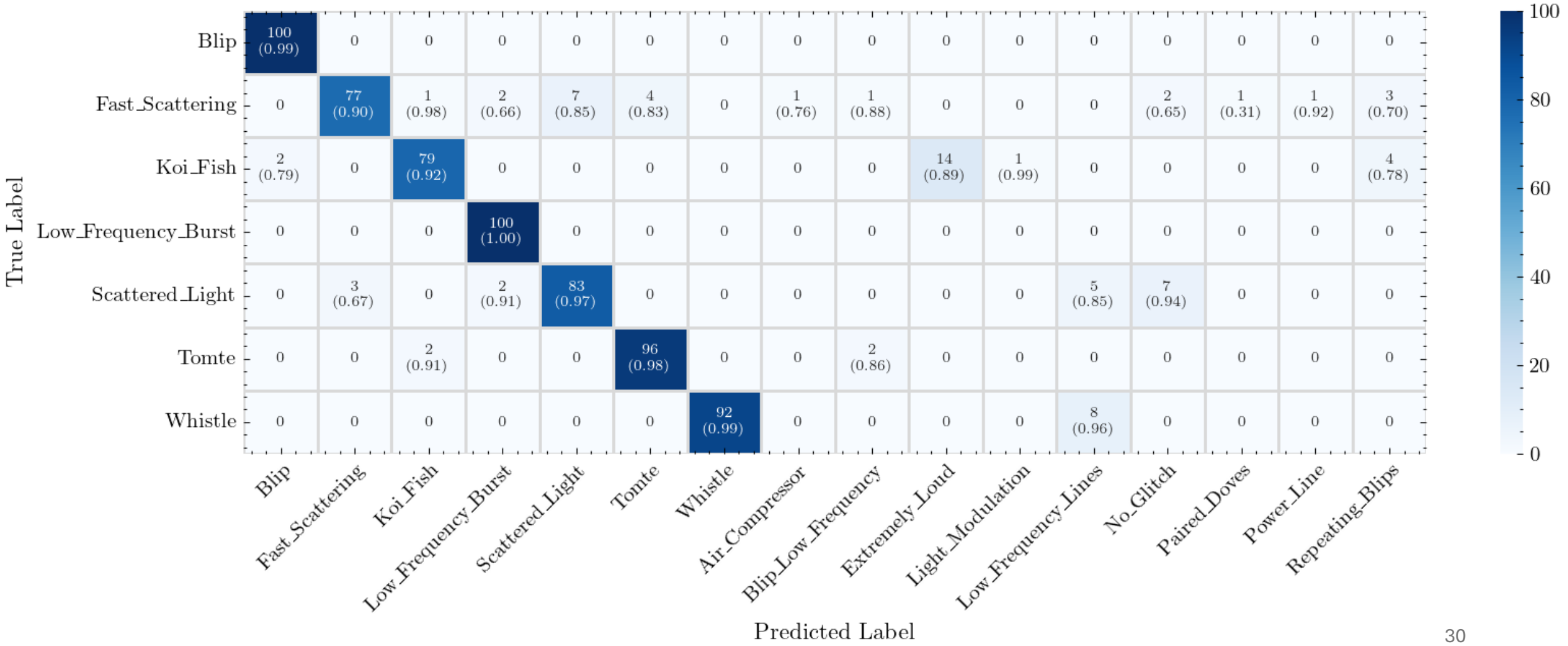
Listening to Isolated Glitches



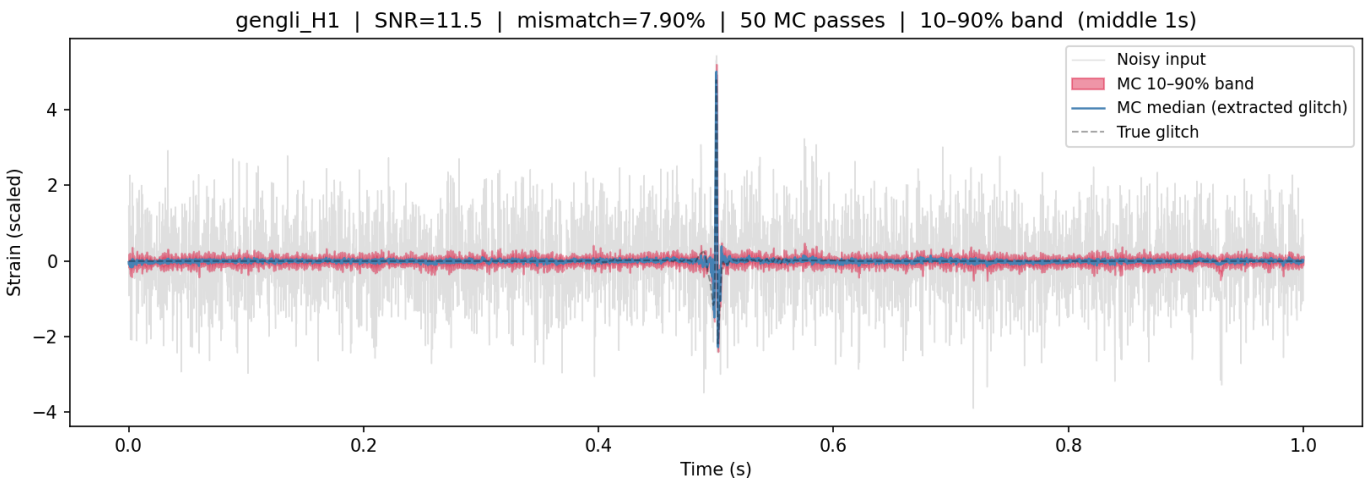


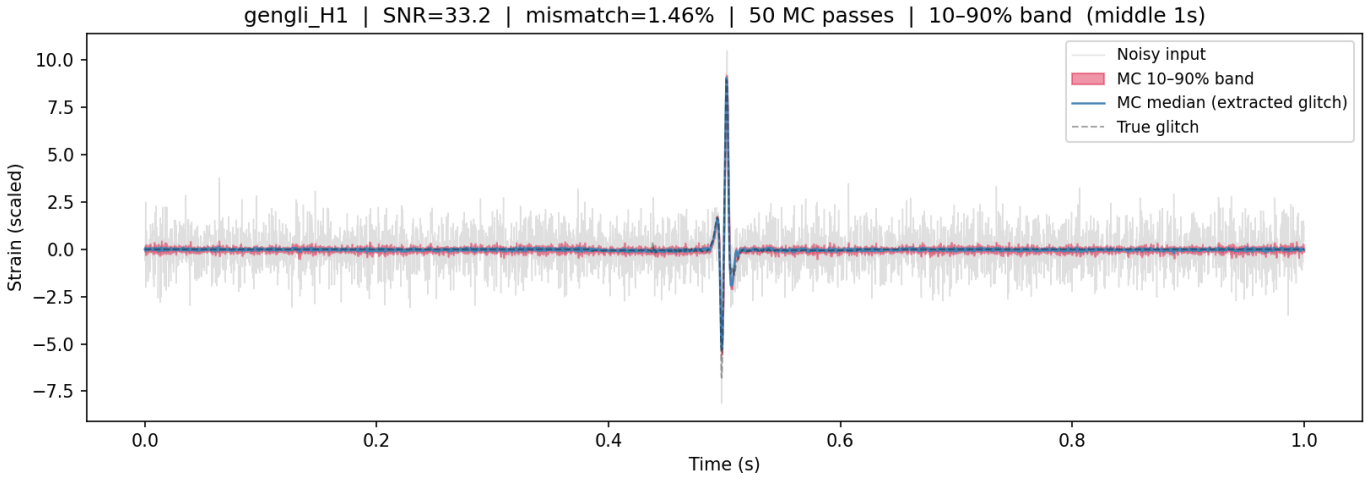
Gravity Spy Classification

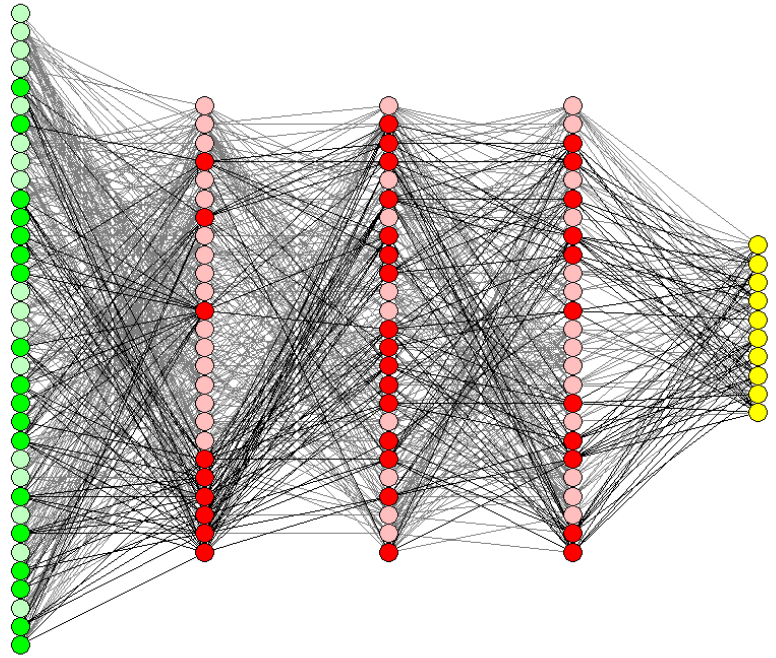
90% of reconstructions are classified with correct label



Uncertainty Quantification with Dropout

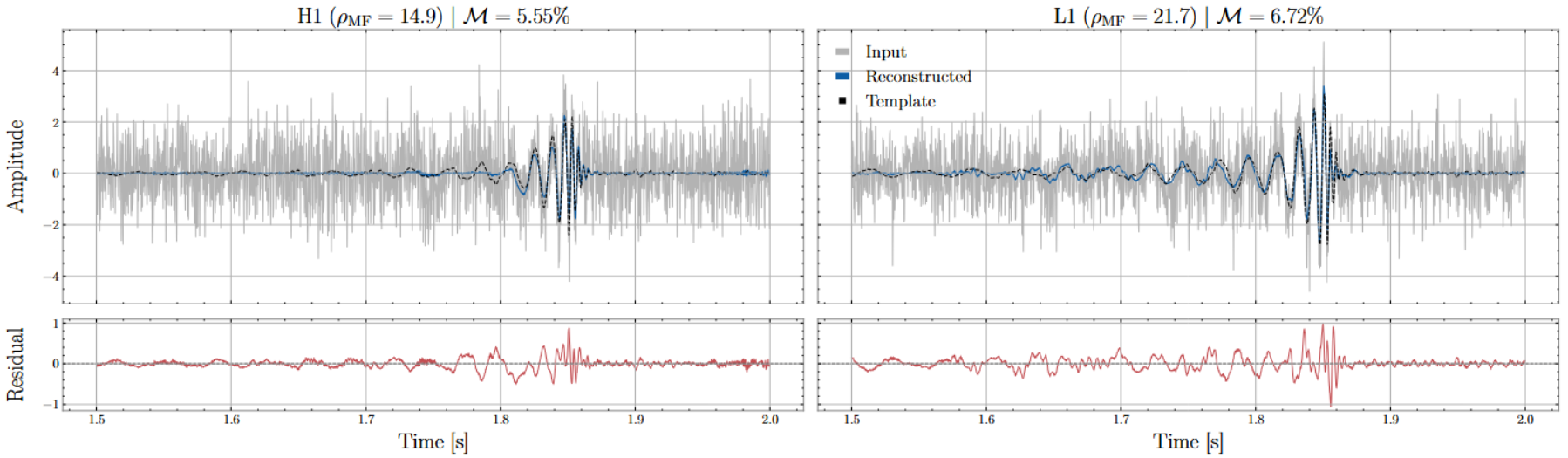




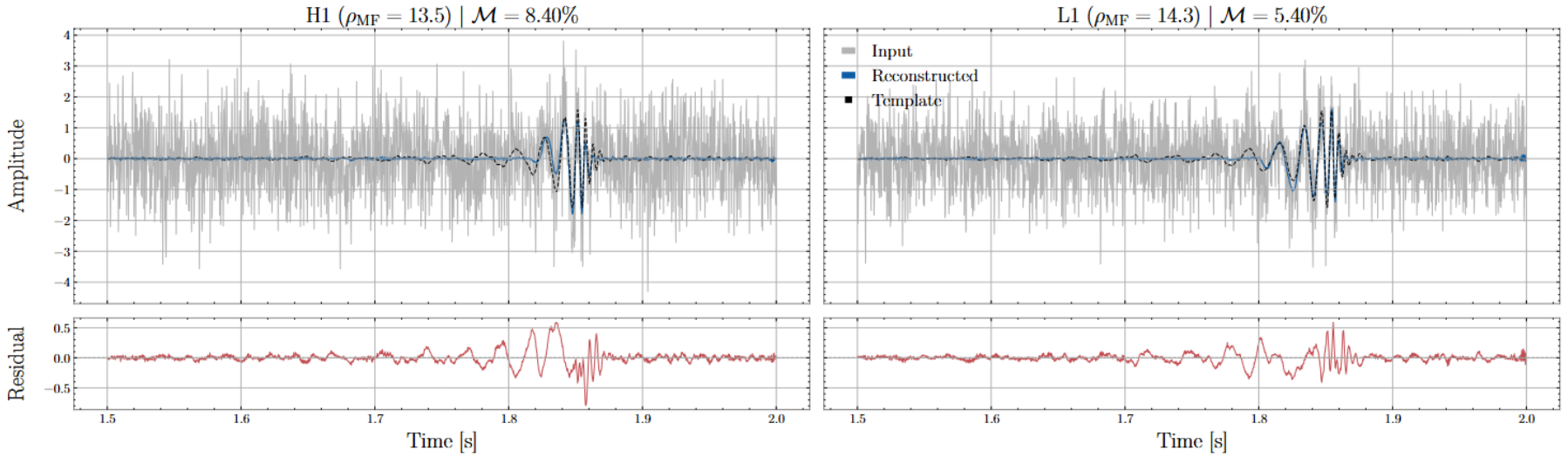


Dropout in action.
Image credit:
https://medium.com/@lipeng2/dropout-is-so-important-e517bbe3ffcc

Reconstructing Signals during LIGO's O3 run

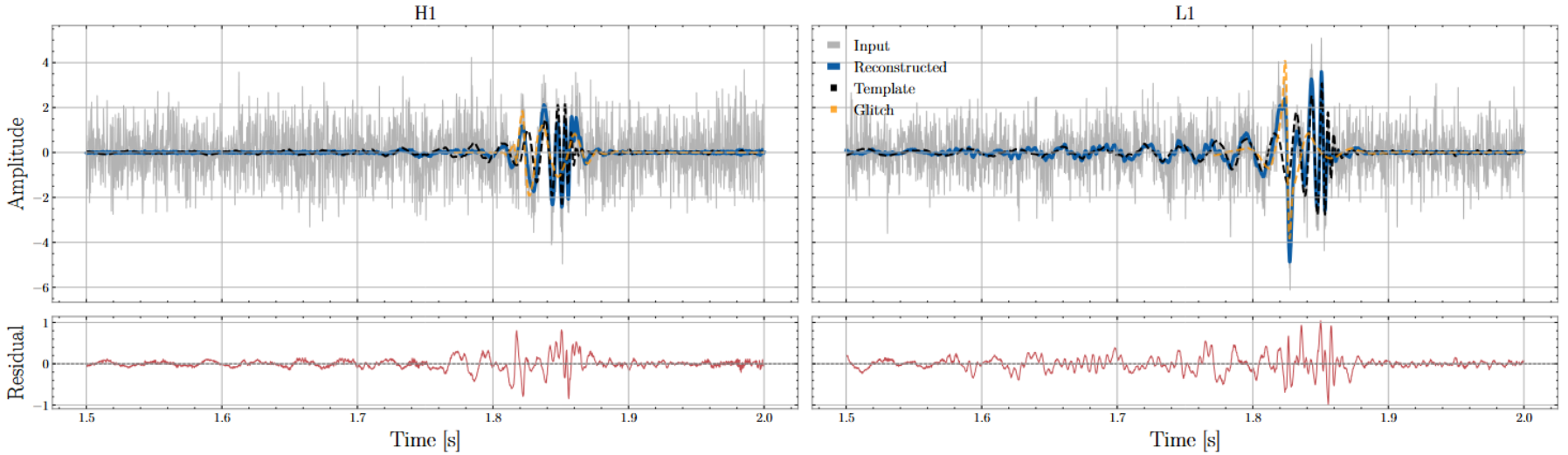


(b) *GW200129_065458*

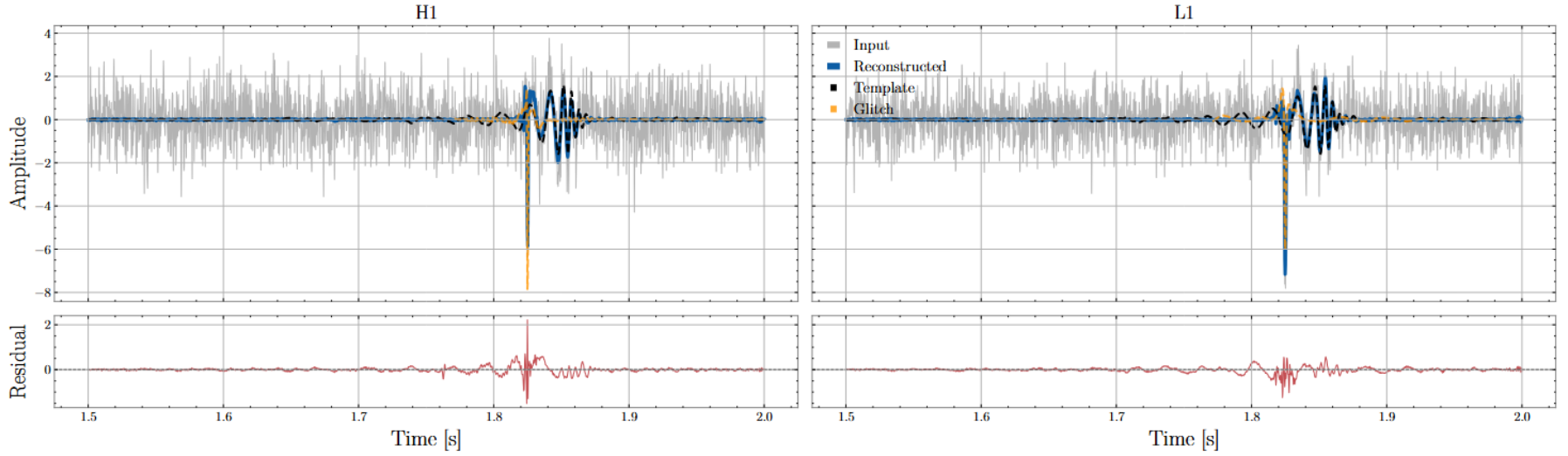


(c) *GW200224_222234*

Separating signals and glitches

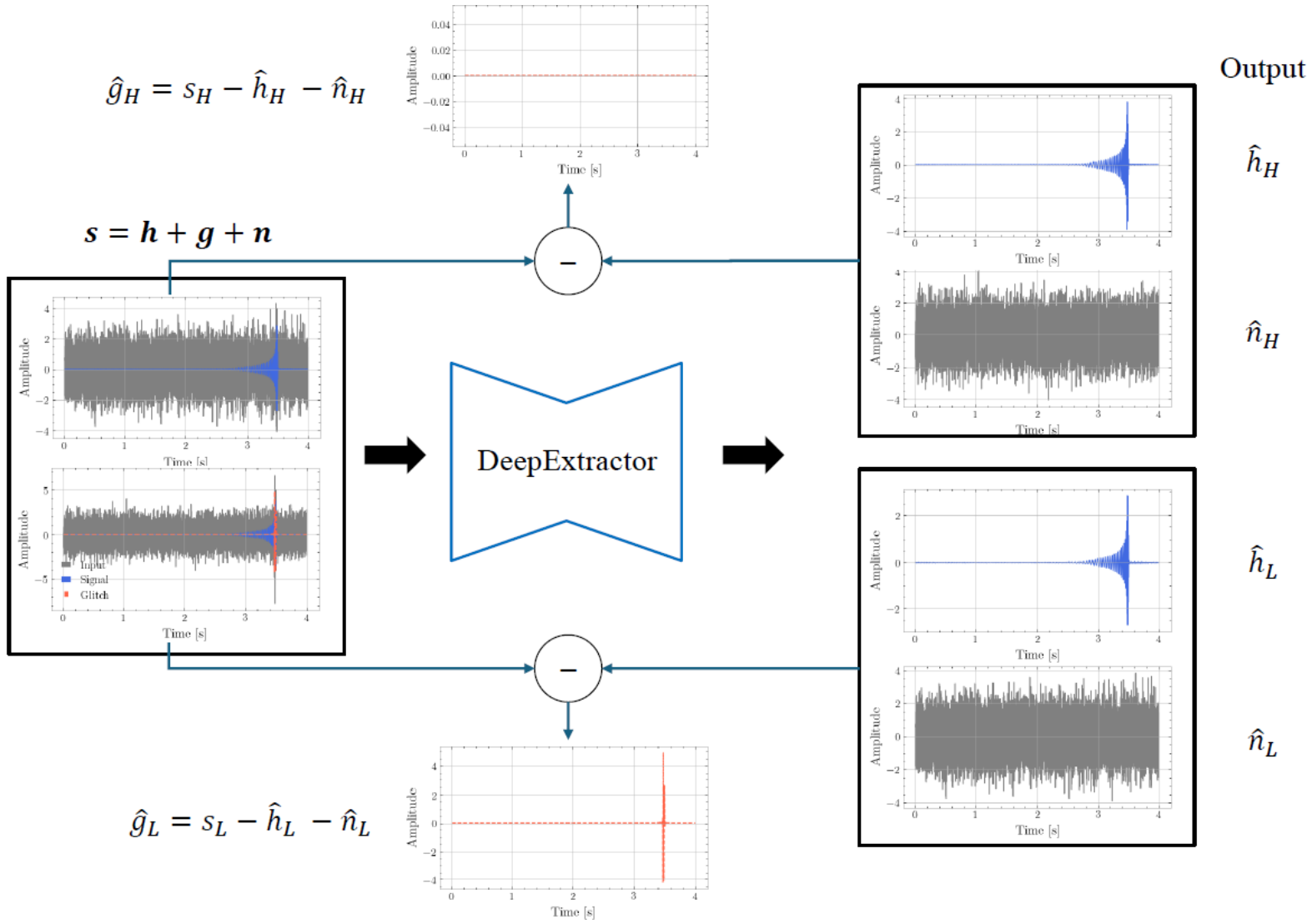


(b) *GW200129_065458*

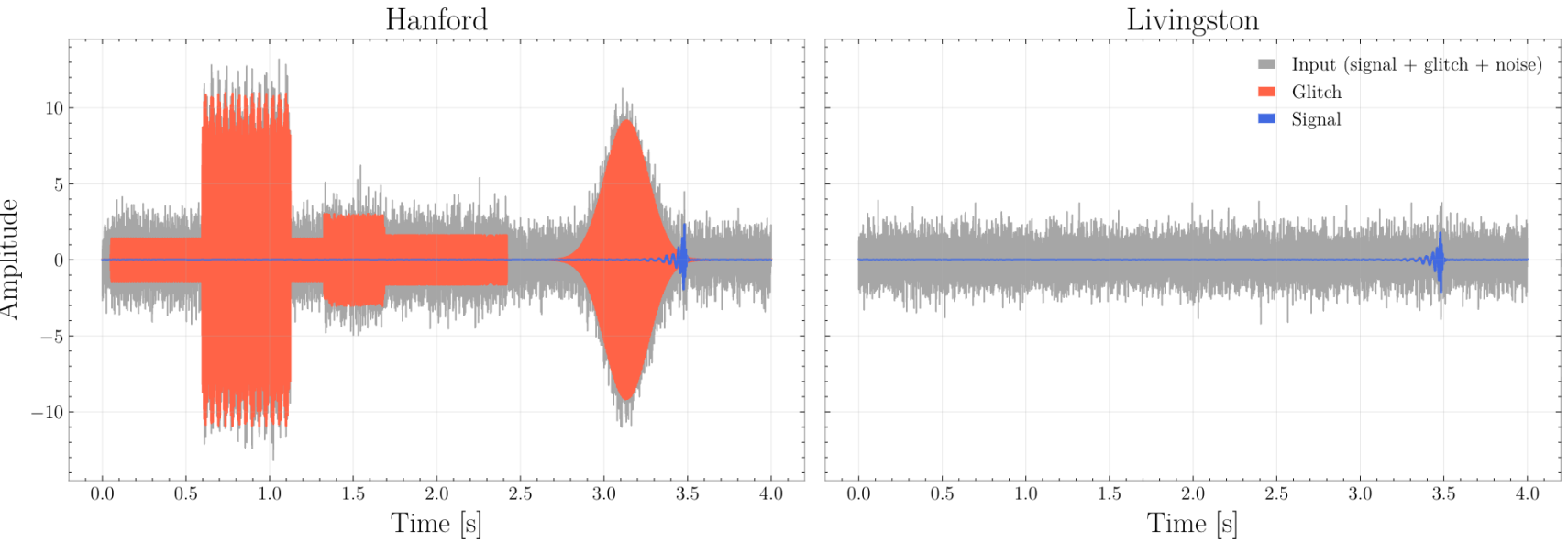


(c) *GW200224_222234*

Signal and glitch separation – Increasing the number of instruments



Signal and glitch separation – training data



A training sample for DeepExtractor signal+glitch separation. We accumulate 1.2M samples.

Parameter	Sampling range / distribution
Primary mass m_1	Uniform [5, 200] $M_\odot$
Secondary mass m_2	Uniform [5, 200] $M_\odot$ (swapped to enforce $m_1 \geq m_2$)
Spin magnitudes (a_1, a_2)	Uniform [0, 0.99]
Spin tilt angles $(\text{tilt}_1, \text{tilt}_2)$	Uniform [0, π]
Spin azimuthal angles (ϕ_{12}, ϕ_{jl})	Uniform [0, 2π]
Luminosity distance D_L	Uniform in comoving volume, [0, 3000] Mpc
Right ascension RA	Uniform [0, 2π]
Declination Dec	Isotropic: $\sin(\text{Dec}) \sim U[-1, 1]$
Inclination angle θ_{JN}	Isotropic: $\cos(\theta_{\text{JN}}) \sim U[-1, 1]$
Polarization angle ψ	Uniform [0, π]
Phase at coalescence ϕ_c	Uniform [0, 2π]
Geocentric time	Fixed at 3.5 s within 4 s window

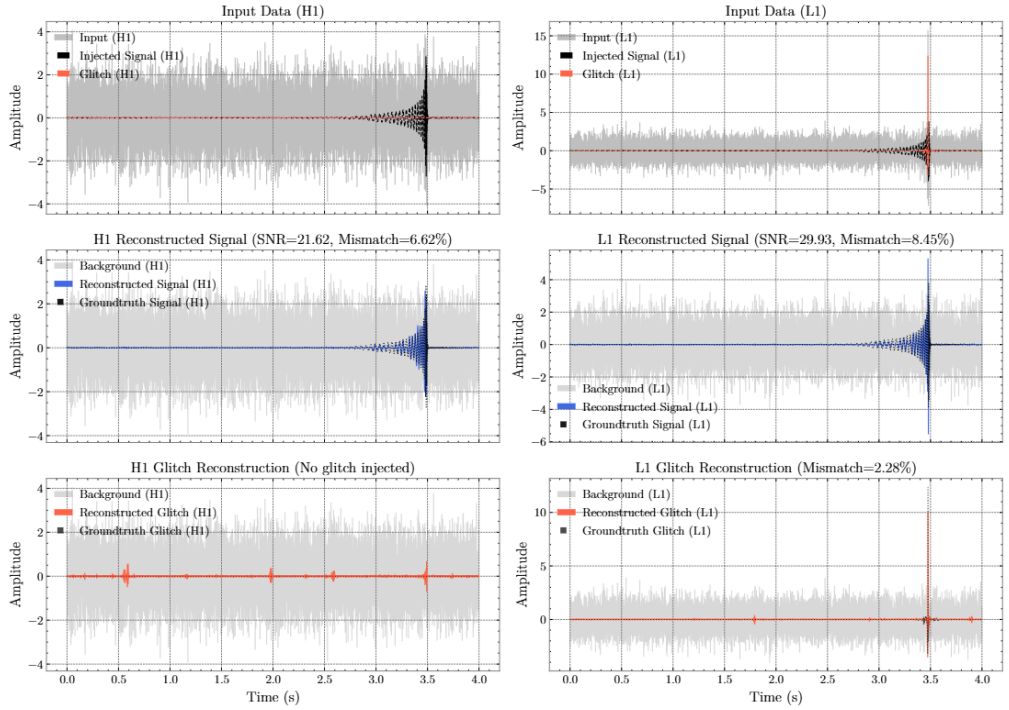
Signal parameter space.

Signal and glitch separation

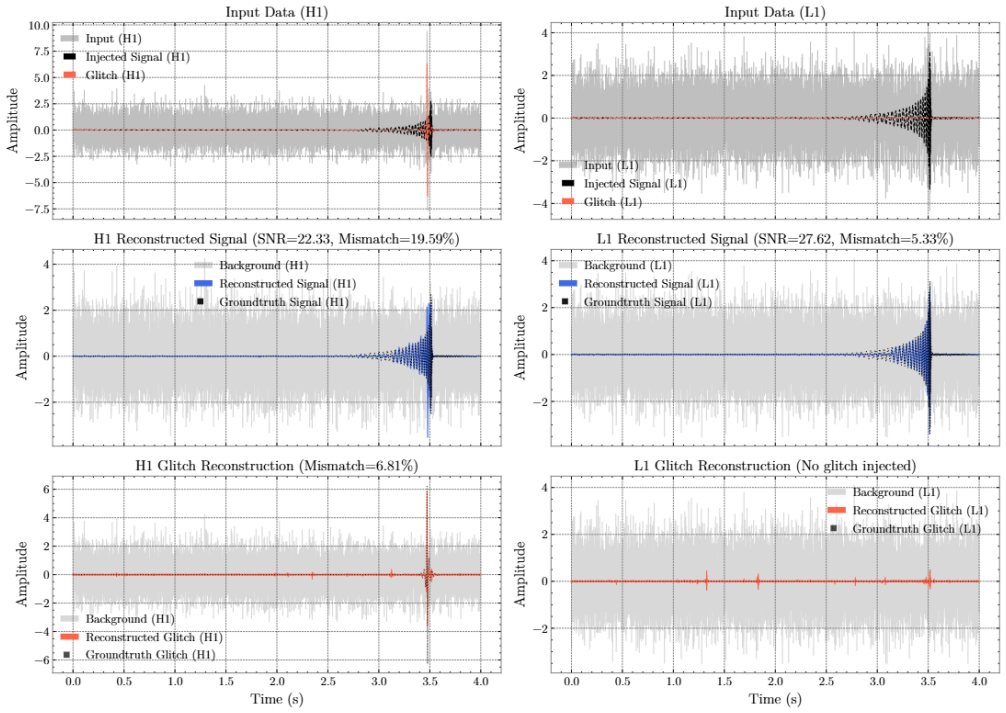
Input

Signal Reconstruction

Glitch Reconstruction



GW150914–like event



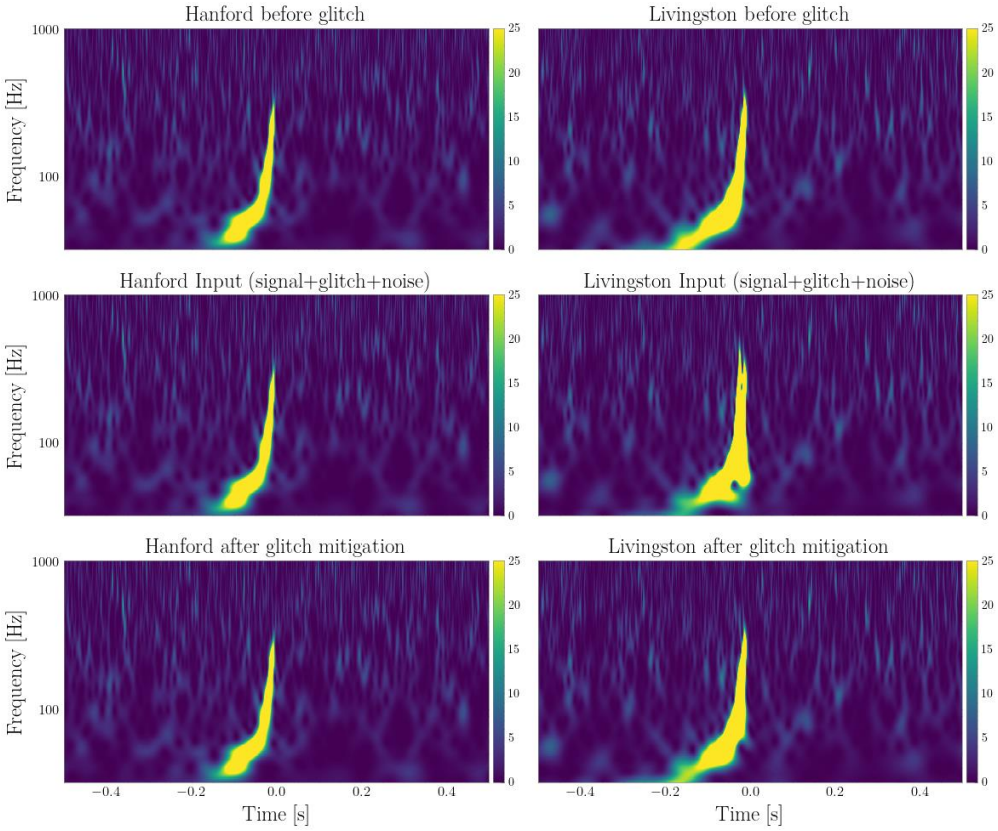
GW200129–like event

Signal and glitch separation

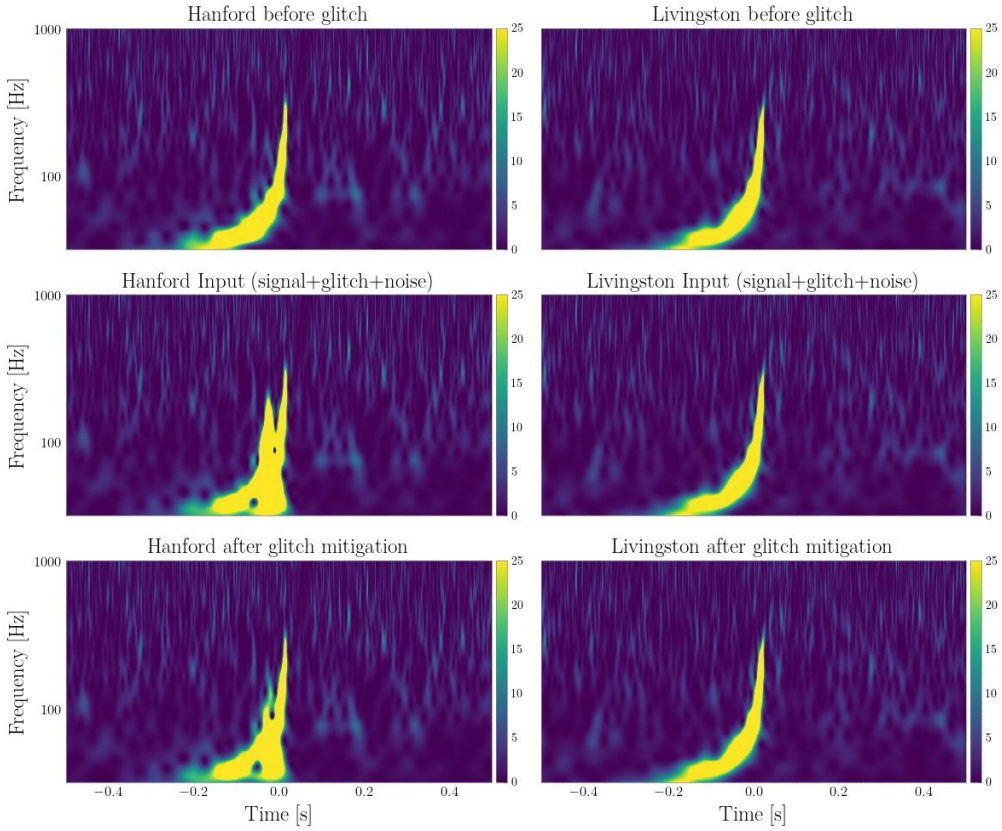
Before Glitch Injection

After Glitch Injection

After Glitch Removal

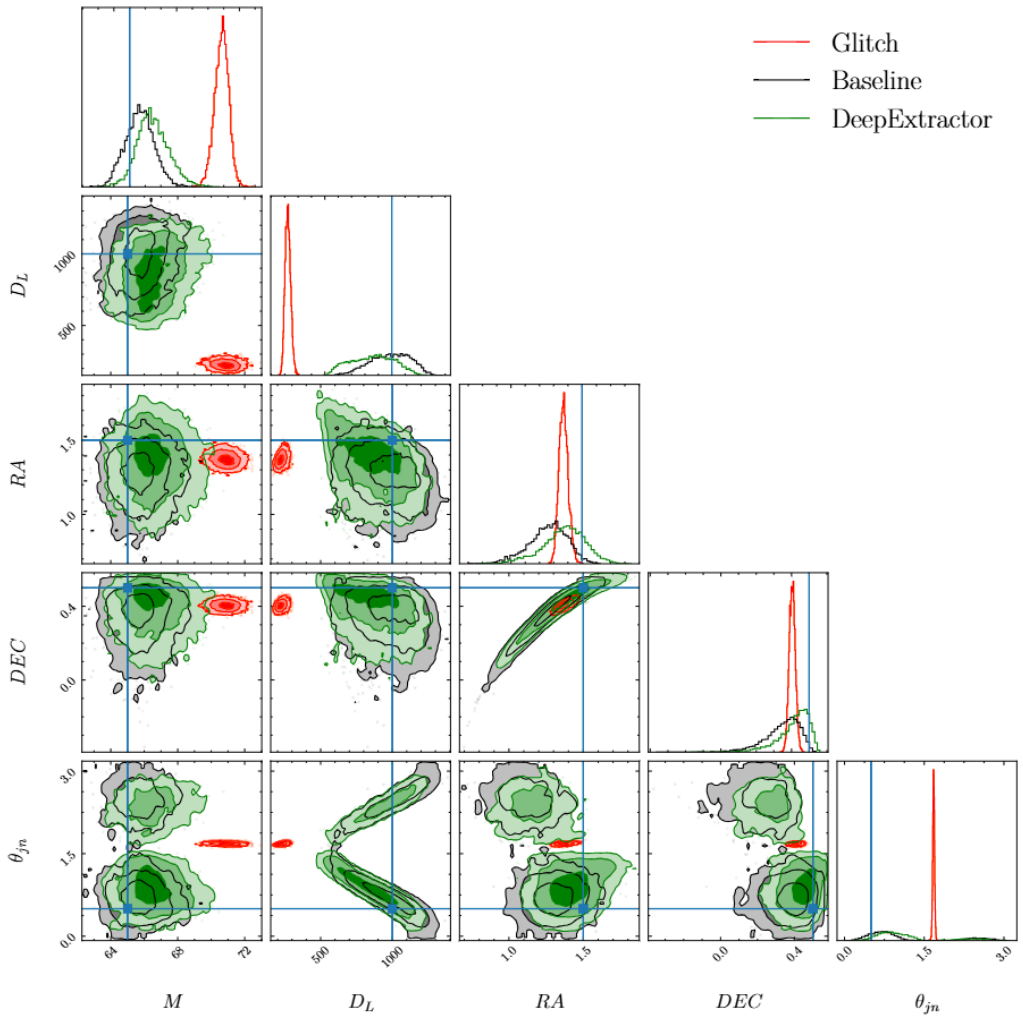


GW150914–like event

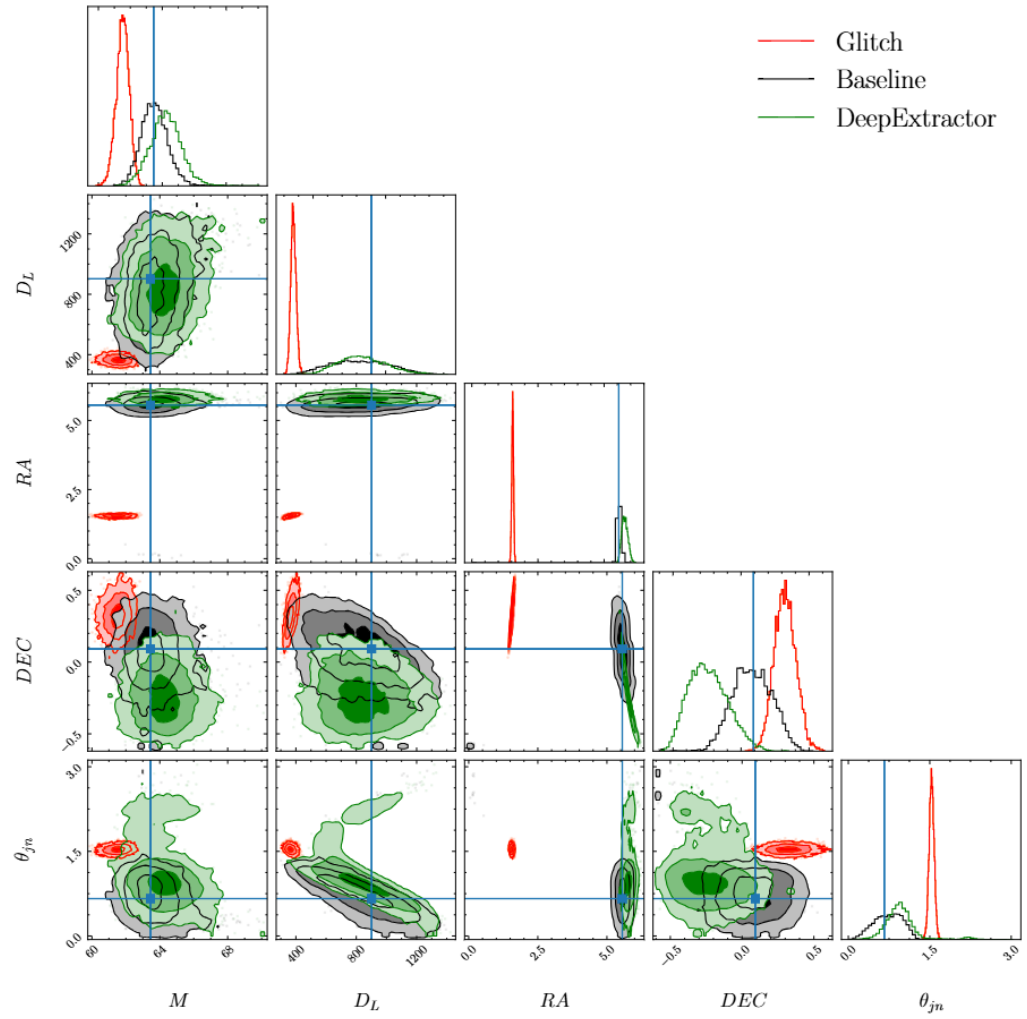


GW200129–like event

Reducing glitch bias in parameter estimation



GW150914–like event



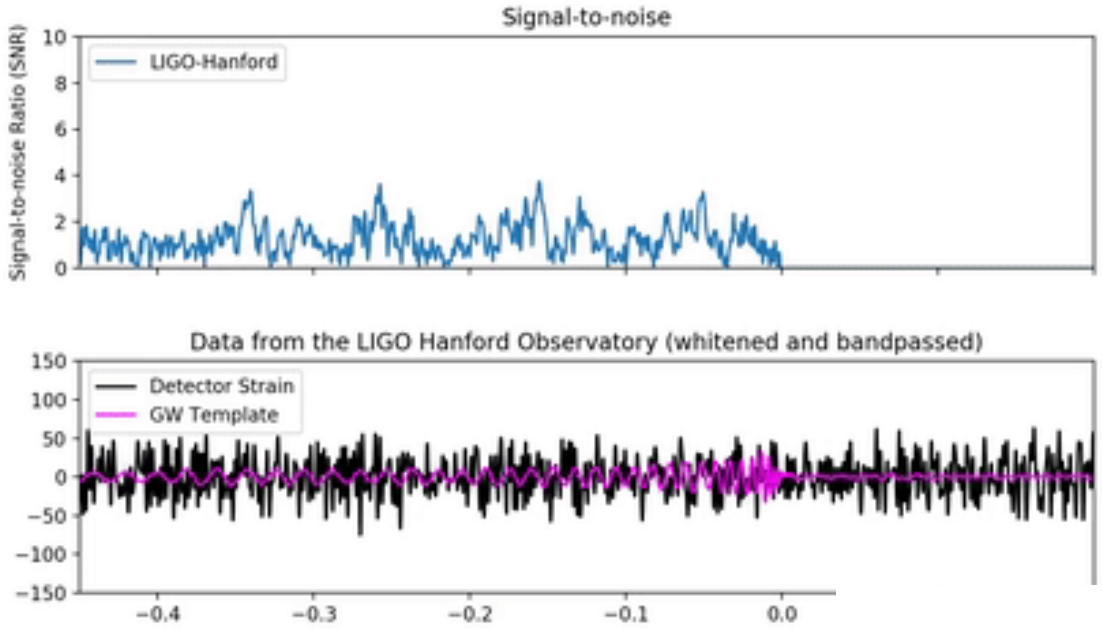
GW200129–like event

Matched-filter searches

$$\log \Lambda = \langle d|\hat{s}\rangle_t - \frac{1}{2}\langle \hat{s}|\hat{s}\rangle_t$$

$$\log \Lambda_{network} = \sum_{i \in \{\mathrm{L1},\mathrm{H1}\}} \log \Lambda_i$$

$$\hat{\rho} = \sqrt{2 \log \Lambda_{network}}$$



Matched filtering example.
Image credit: https://github.com/gwastro/pycbc

DeepExtractor for a template-free search statistic

$$\log \Lambda = \langle d|\hat{s}\rangle_t - \frac{1}{2}\langle \hat{s}|\hat{s}\rangle_t$$

$$\log \Lambda_{network} = \sum_{i \in \{L1,H1\}} \log \Lambda_i$$

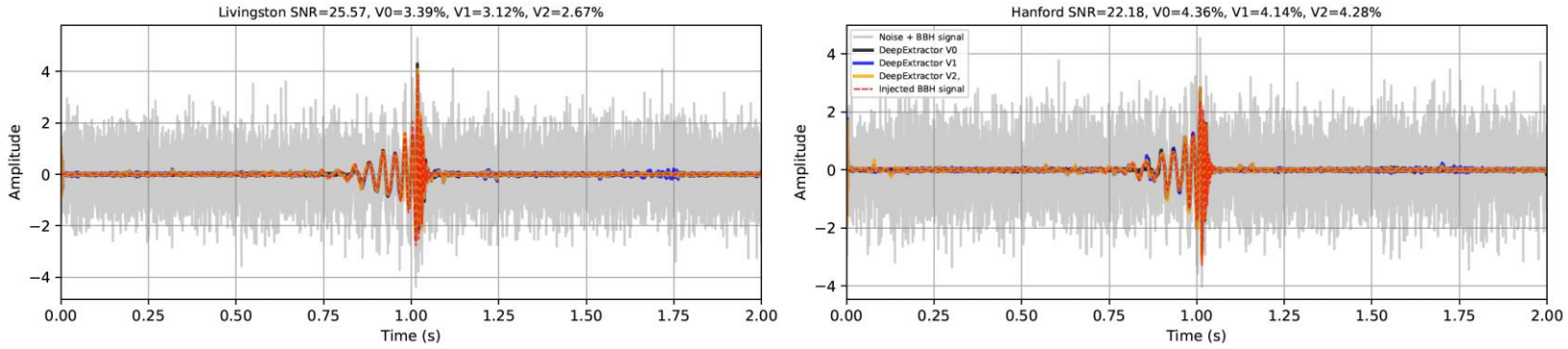
$$\hat{\rho} = \sqrt{2 \log \Lambda_{network}}$$

Our statistic

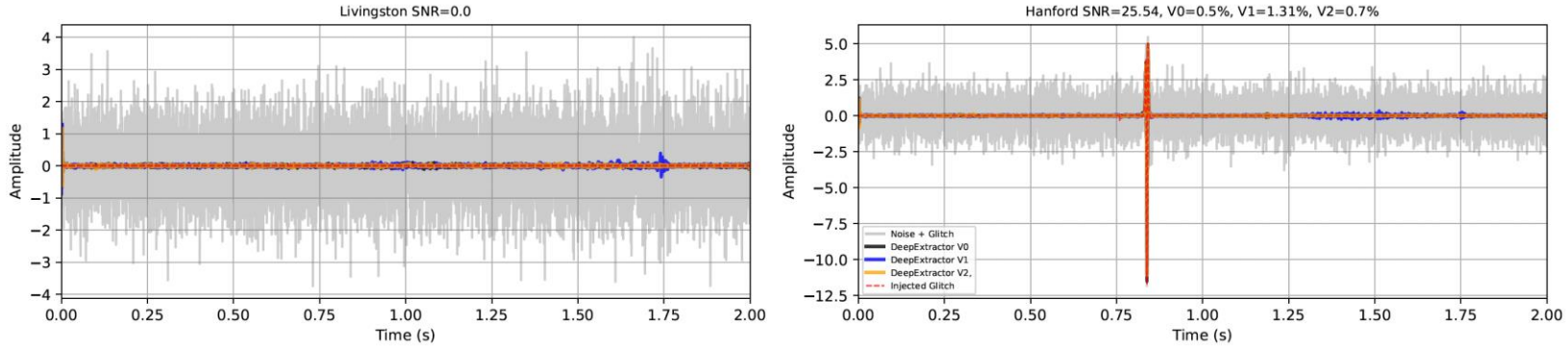
$$\omega^t_{L,H} = \max_{|\tau| \leq \tau_{max}} \left| \frac{1}{\mathcal{N}(\tau)} \sum_i \hat{s}_L(t_i)\hat{s}_H(t_i+\tau) \right|$$

$$\hat{\rho} = \omega_{L,H} \times \hat{\rho}_{network}.$$

Signals and Glitches in the Search

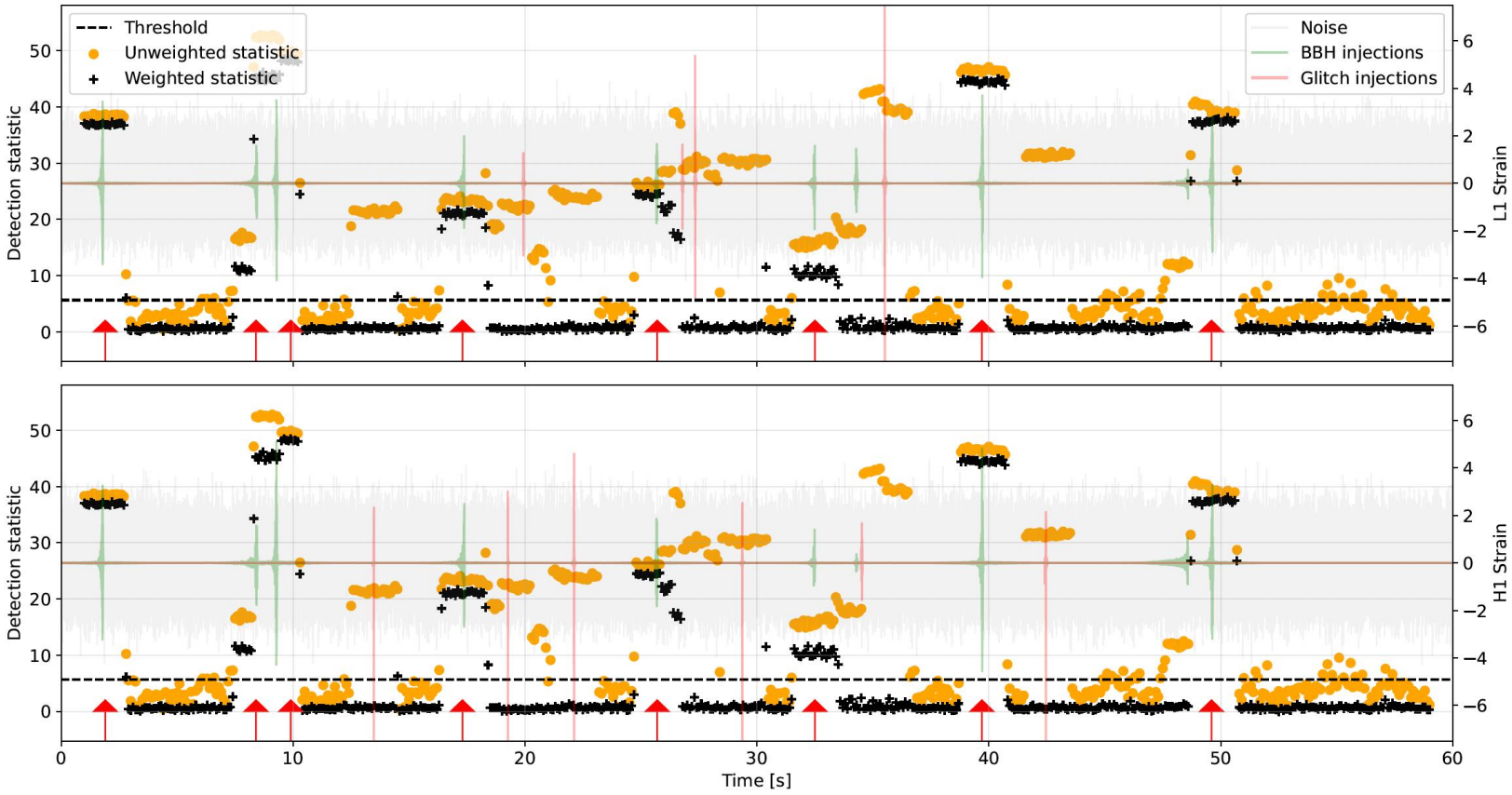


DeepExtractor signal reconstruction.



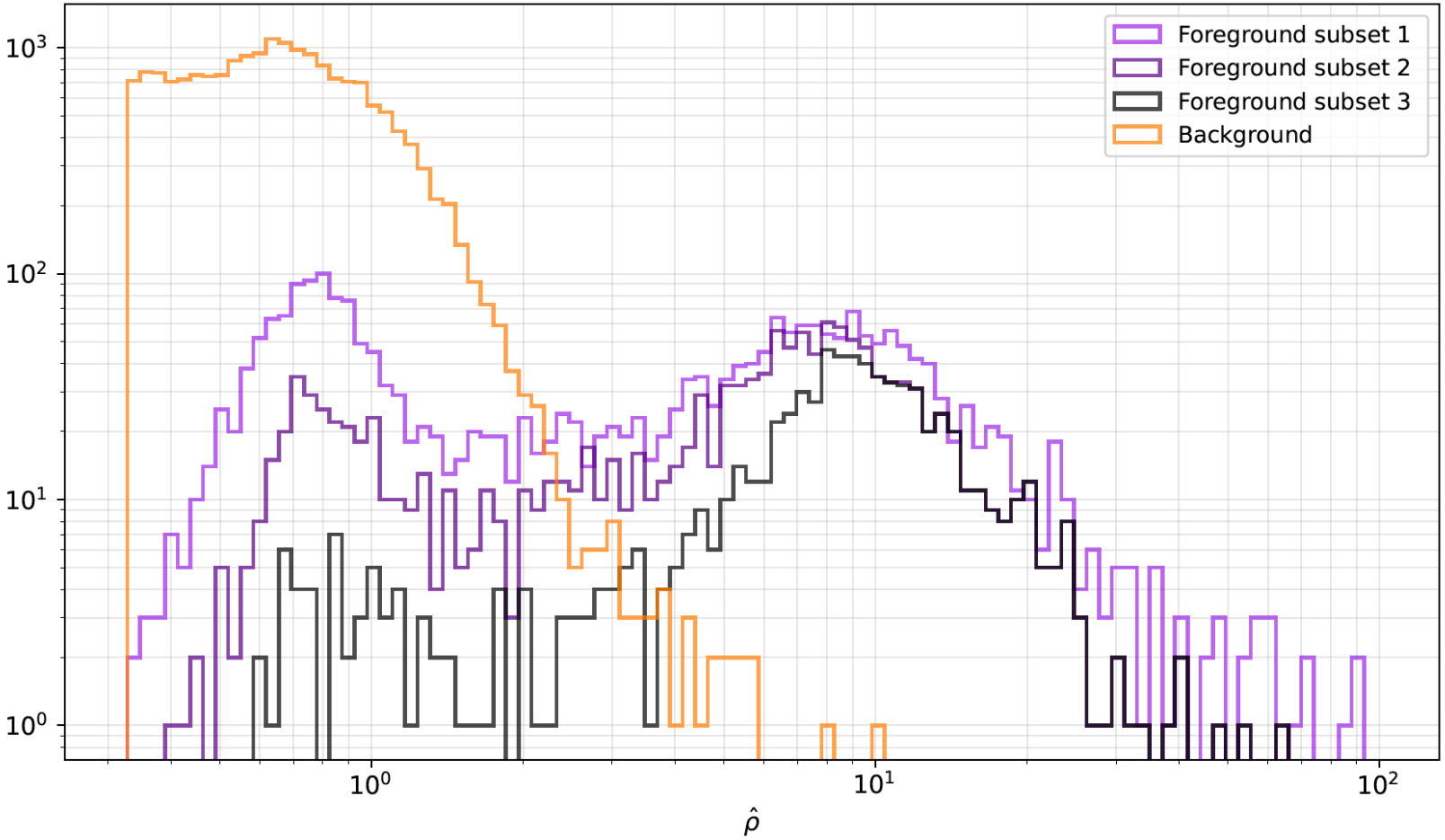
DeepExtractor glitch reconstruction.

Simulated Example of the Search



Simulated example of 60s of data.

A More Realistic Search



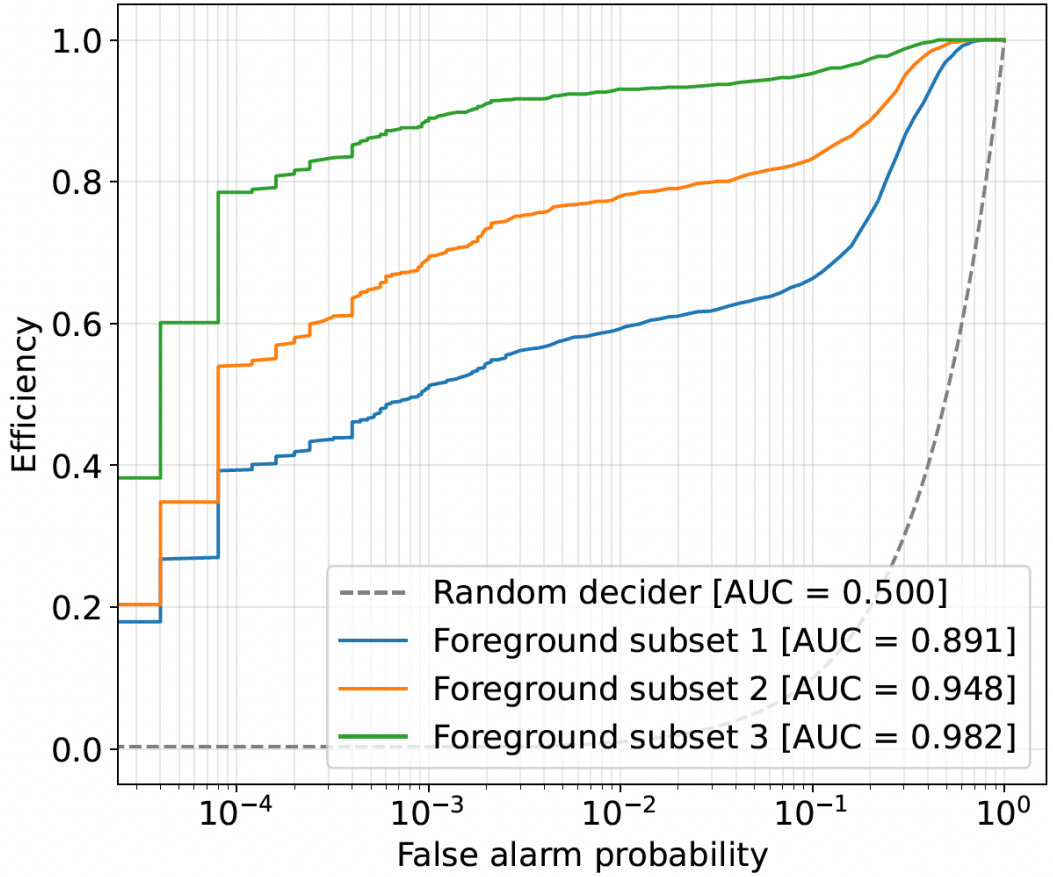
Three population subsets:

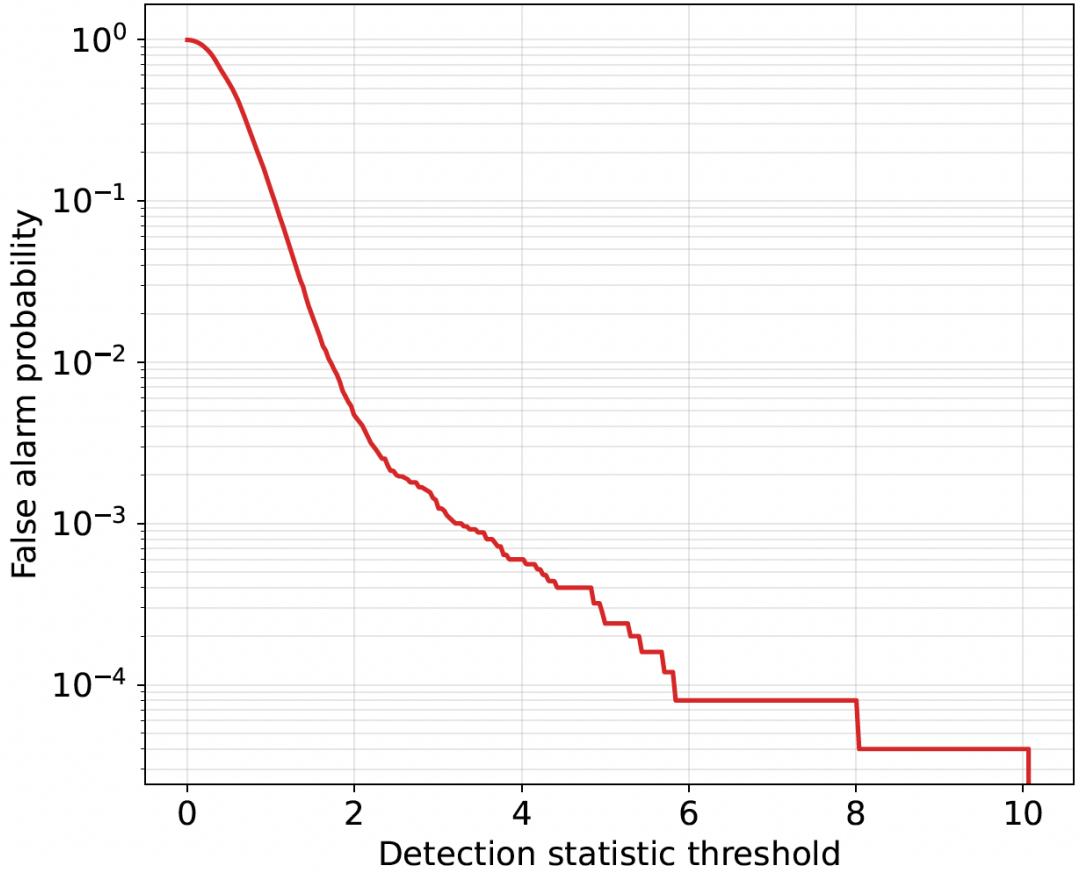
	Redshift	Signal length	Matched filter SNR	Number of samples
Subset 1	$z < 2$	$t < 10$	No limitation	N = 2502
Subset 2	$z < 2$	$t < 2$	No limitation	N = 1384
Subset 3	$z < 2$	$t < 2$	$\rho > 6$	N = 735

ROC Curve

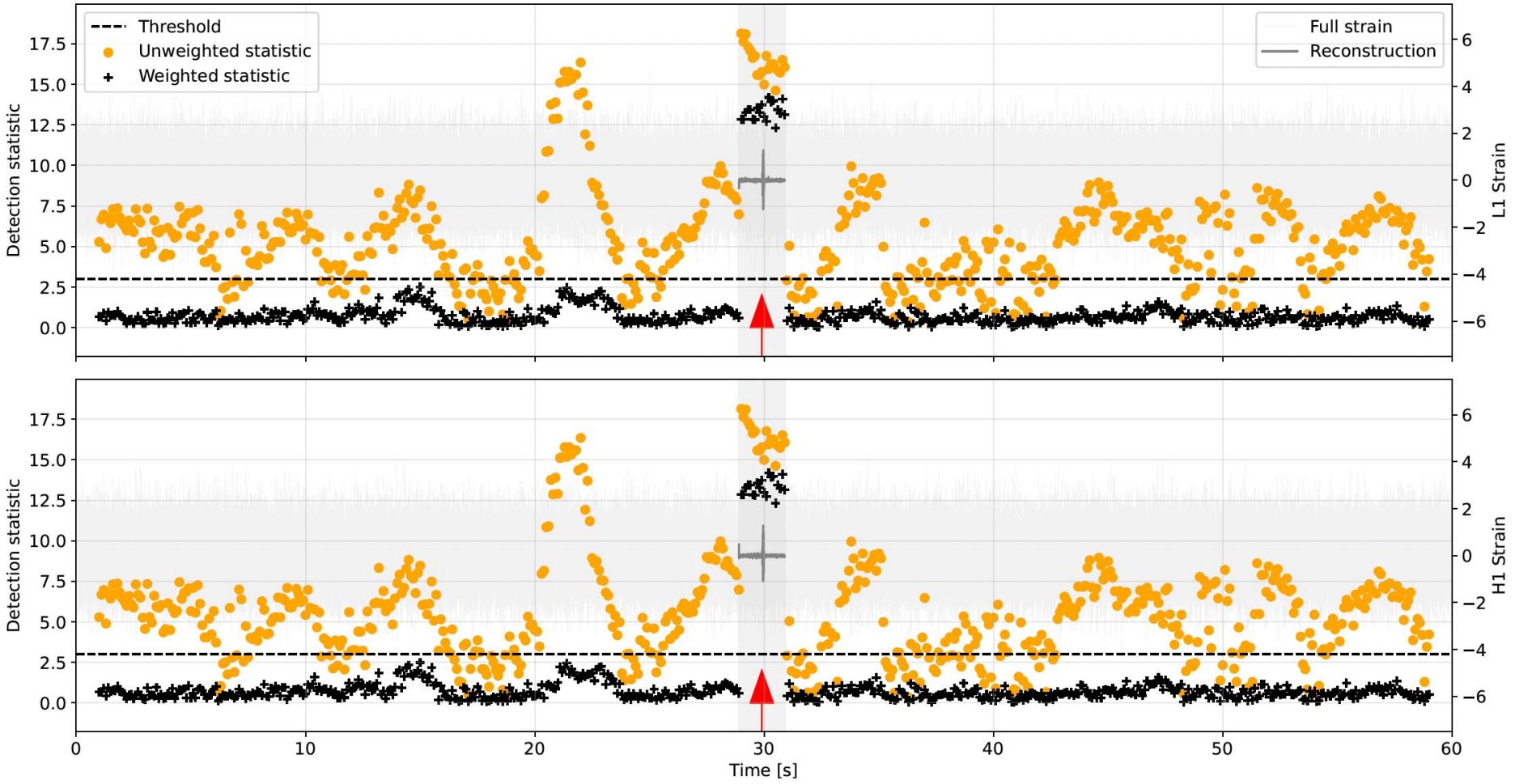
	Redshift	Signal length	Matched filter SNR	Number of samples
Subset 1	$z < 2$	$t < 10$	No limitation	N = 2502
Subset 2	$z < 2$	$t < 2$	No limitation	N = 1384
Subset 3	$z < 2$	$t < 2$	$\rho > 6$	N = 735

Three subsets of a realistic population.

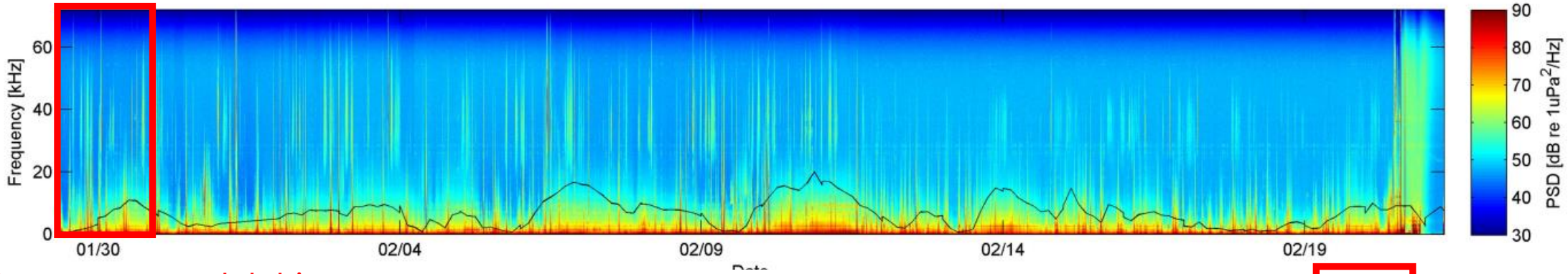




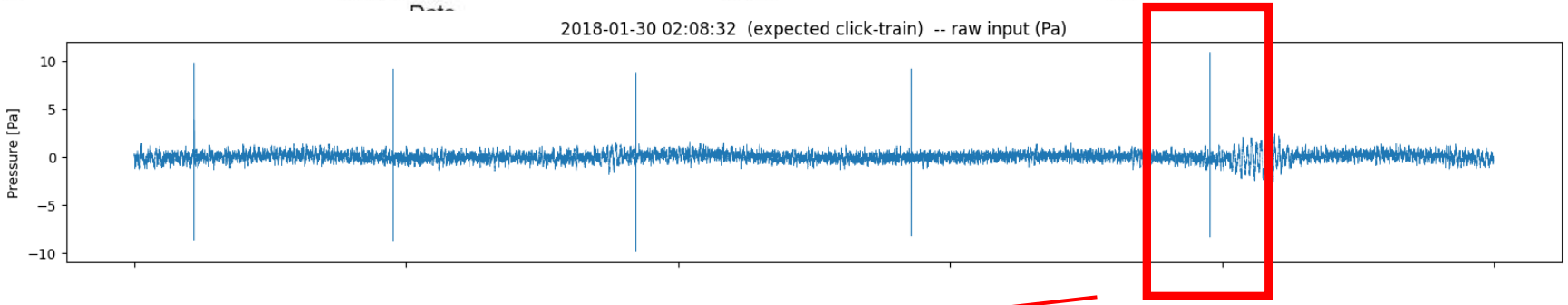
Detecting GW231123

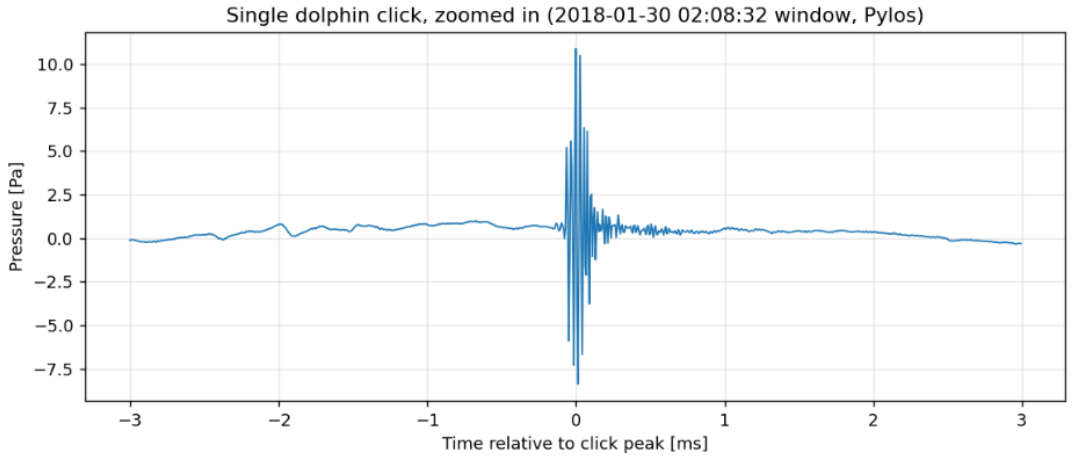


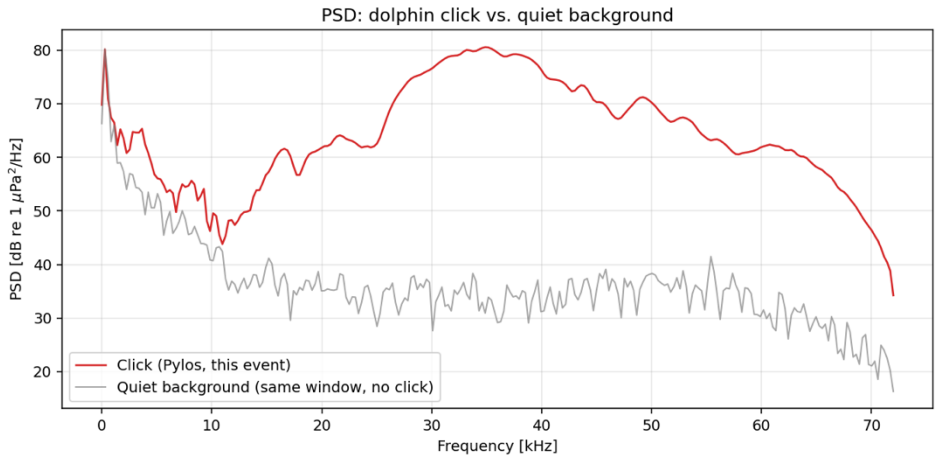
DeepExtrctor for KM3Net Glitches?



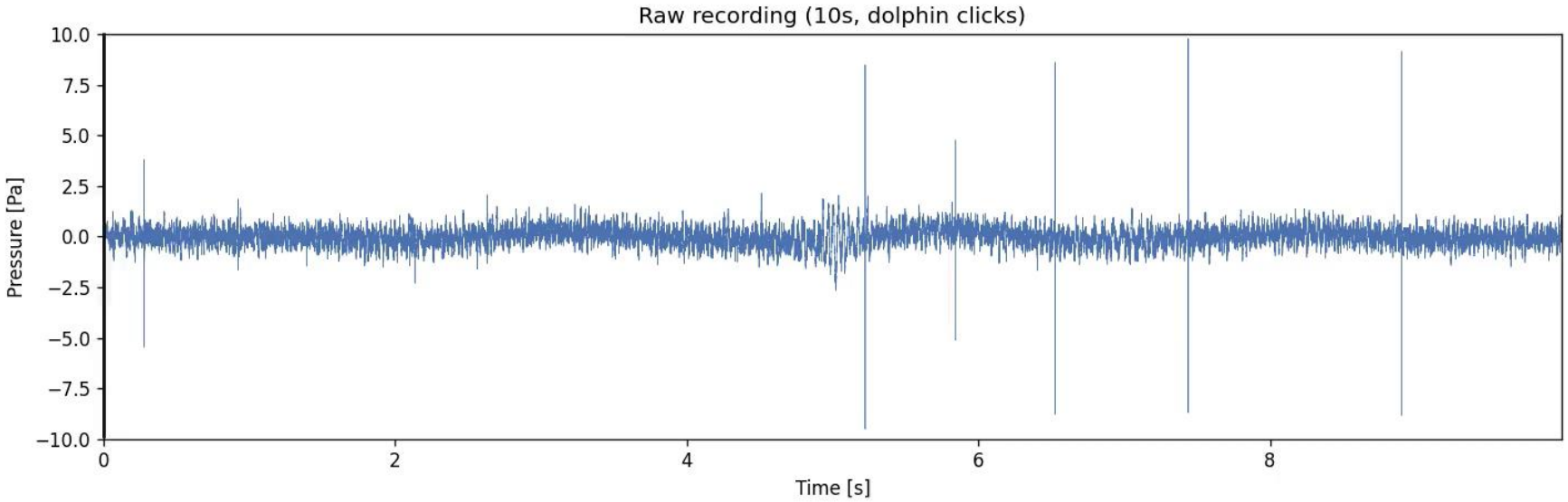
Can we extract dolphin clicks?

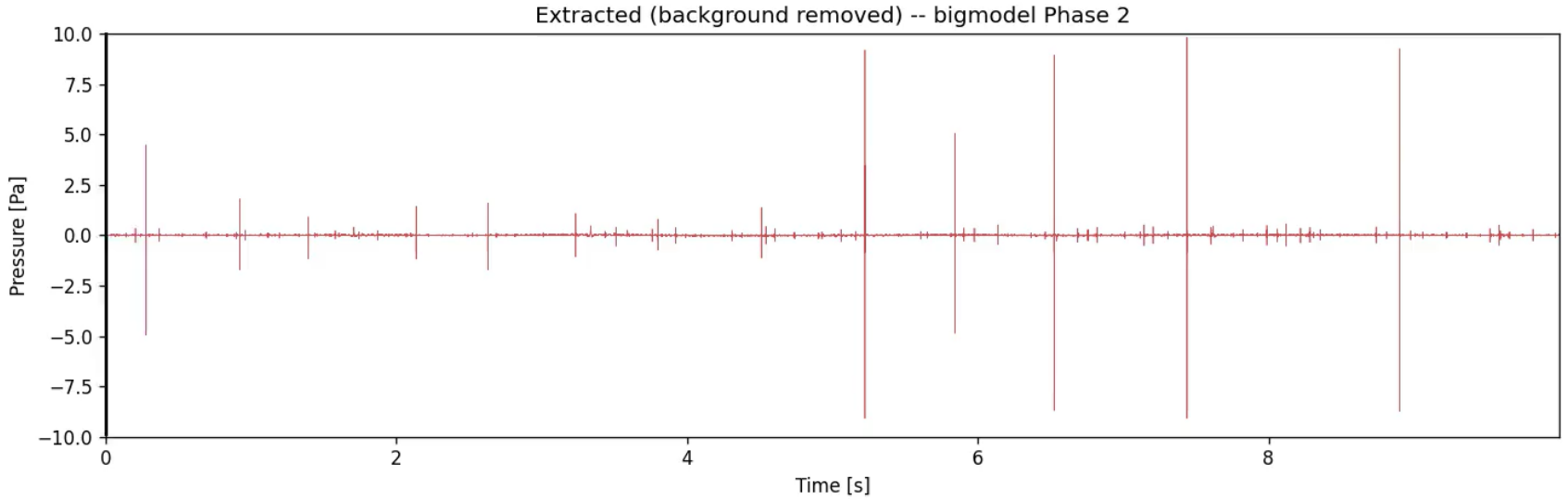






Dolphin click (glitch) extraction





Summary & Questions

- Time-domain **reconstruction** of arbitrary GW signals and glitches
- **Interpolatation** to unseen classes
  - Trained with diverse **combinations** of **proxy glitch** classes
  - **Residual** approach: **predict** the **noise** and **subtract** to give glitch
- Surpass BayesWave glitch recovery performance
  - Reconstruct 2s of data in 0.1s
  - **x10,000 speedup** vs BayesWave
- **Reduce bias** in **Parameter** Estimation with signal+glitch separation
- **Template-free and unmodelled searches** with DeepExtractor **reconstructions**
- **Package** on **PyPI**
- Generic **background** estimation?

Read the docs!

Appendix

Results – Simulated experiments

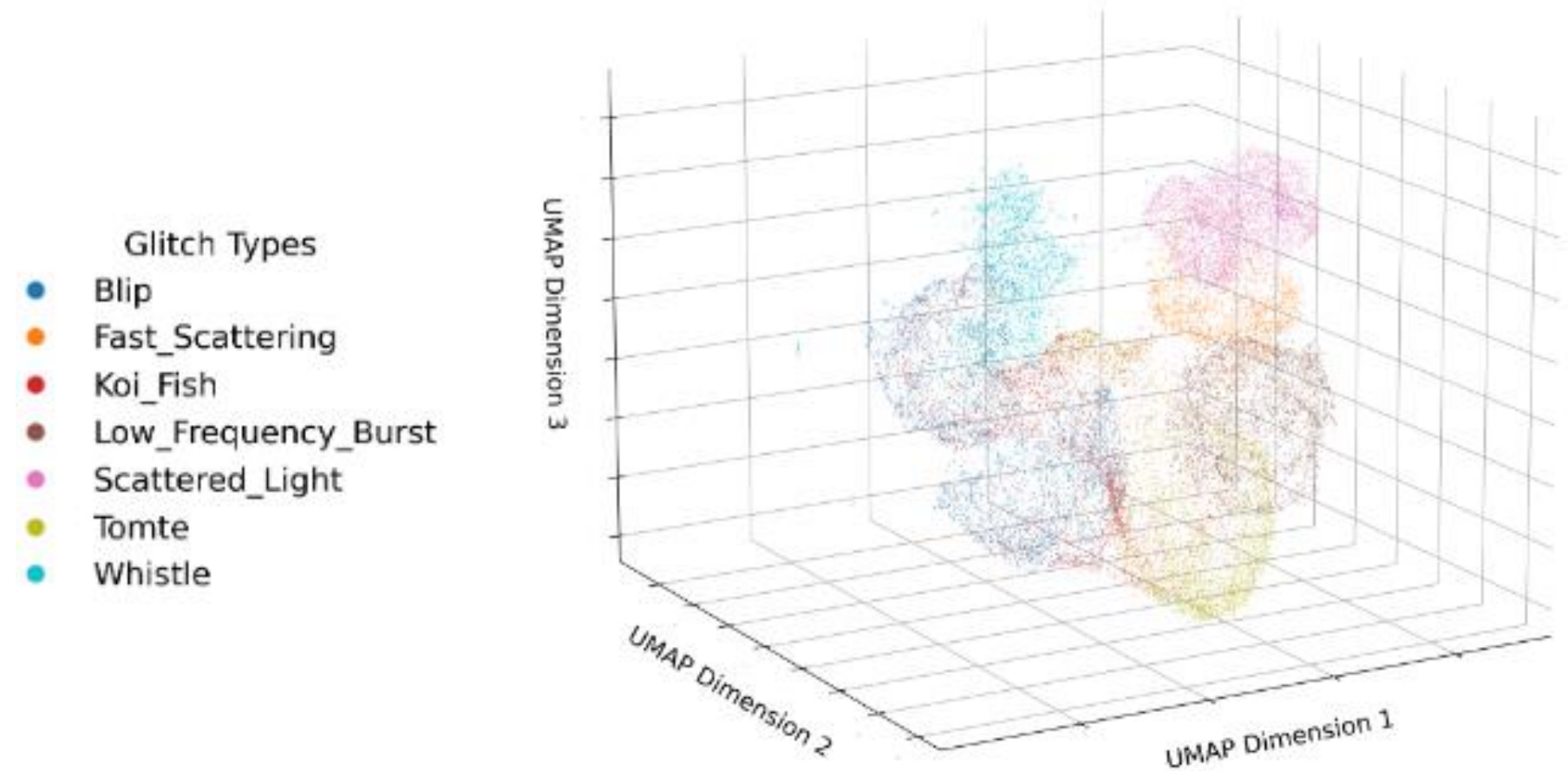
Glitch Type	UNET1D	DnCNN1D	Autoencoder1D	Autoencoder2D	DeepEx. (65x129)	DeepEx. (129^2)	DeepEx. (257^2)
chirp	$2.0^{+2.1}_{-0.8}$	$26.1^{+20.8}_{-13.9}$	$2.3^{+2.4}_{-0.9}$	$41.2^{+20.1}_{-9.3}$	$1.9^{+1.7}_{-0.6}$	$1.1^{+0.9}_{-0.4}$	$\mathbf{0.8}^{+0.7}_{-0.3}$
sine	$1.7^{+2.8}_{-0.6}$	$23.5^{+27.6}_{-11.5}$	$1.9^{+3.7}_{-0.7}$	$41.1^{+20.2}_{-9.7}$	$1.7^{+1.8}_{-0.6}$	$0.8^{+1.0}_{-0.2}$	$\mathbf{0.5}^{+0.7}_{-0.1}$
sine_gaussian	$1.4^{+1.8}_{-0.6}$	$8.9^{+13.5}_{-3.3}$	$1.6^{+2.0}_{-0.7}$	$40.2^{+16.3}_{-11.3}$	$1.7^{+1.8}_{-0.8}$	$0.6^{+0.7}_{-0.2}$	$\mathbf{0.4}^{+0.6}_{-0.1}$
gaussian_pulse	$\mathbf{0.4}^{+0.4}_{-0.2}$	$3.8^{+3.8}_{-1.7}$	$0.6^{+0.7}_{-0.3}$	$37.7^{+13.9}_{-11.5}$	$1.8^{+2.1}_{-0.8}$	$0.6^{+0.7}_{-0.2}$	$\mathbf{0.4}^{+0.4}_{-0.2}$
ringdown	$\mathbf{0.6}^{+0.8}_{-0.2}$	$5.2^{+4.3}_{-2.1}$	$0.9^{+1.0}_{-0.4}$	$39.3^{+15.0}_{-9.5}$	$2.2^{+2.4}_{-1.0}$	$0.9^{+0.8}_{-0.4}$	$0.7^{+0.6}_{-0.3}$
gengli_H1	$1.0^{+1.0}_{-0.3}$	$3.8^{+3.3}_{-1.4}$	$1.2^{+1.1}_{-0.4}$	$36.6^{+13.0}_{-9.0}$	$2.7^{+2.0}_{-1.1}$	$1.2^{+0.8}_{-0.4}$	$\mathbf{0.9}^{+0.7}_{-0.3}$
gengli_L1	$1.2^{+0.7}_{-0.4}$	$4.0^{+2.8}_{-1.1}$	$1.3^{+0.9}_{-0.4}$	$36.1^{+10.9}_{-7.6}$	$2.8^{+2.1}_{-1.1}$	$1.2^{+0.9}_{-0.3}$	$\mathbf{1.1}^{+0.7}_{-0.4}$
cdvgan_blip	$1.2^{+1.0}_{-0.4}$	$4.5^{+3.9}_{-1.7}$	$1.4^{+1.1}_{-0.5}$	$36.2^{+15.6}_{-9.1}$	$2.9^{+2.6}_{-1.0}$	$1.2^{+0.9}_{-0.4}$	$\mathbf{0.9}^{+0.8}_{-0.2}$
cdvgan_tomte	$1.0^{+0.7}_{-0.3}$	$4.5^{+3.4}_{-1.7}$	$1.2^{+0.8}_{-0.4}$	$38.0^{+12.4}_{-10.7}$	$2.6^{+2.5}_{-1.0}$	$1.0^{+0.7}_{-0.3}$	$\mathbf{0.9}^{+0.6}_{-0.3}$
cdvgan_bbh	$1.0^{+1.1}_{-0.3}$	$4.6^{+5.2}_{-1.5}$	$1.1^{+1.2}_{-0.3}$	$34.6^{+17.7}_{-6.7}$	$2.7^{+2.4}_{-0.9}$	$1.2^{+1.0}_{-0.5}$	$\mathbf{0.8}^{+0.9}_{-0.2}$
cdvgan_simplex	$1.8^{+1.4}_{-0.6}$	$5.2^{+5.4}_{-1.6}$	$2.0^{+1.6}_{-0.7}$	$36.9^{+16.1}_{-7.6}$	$3.5^{+3.5}_{-1.1}$	$1.8^{+1.4}_{-0.5}$	$\mathbf{1.6}^{+1.2}_{-0.5}$
cdvgan_uniform	$1.6^{+1.1}_{-0.5}$	$5.3^{+3.9}_{-1.9}$	$1.7^{+1.3}_{-0.5}$	$38.1^{+13.3}_{-10.0}$	$3.1^{+3.3}_{-1.1}$	$1.6^{+1.0}_{-0.5}$	$\mathbf{1.4}^{+0.9}_{-0.5}$
Total	$1.2^{+1.1}_{-0.4}$	$5.9^{+6.9}_{-2.4}$	$1.4^{+1.3}_{-0.5}$	$37.9^{+15.2}_{-9.2}$	$2.5^{+2.3}_{-0.9}$	$1.1^{+0.9}_{-0.4}$	$\mathbf{0.9}^{+0.7}_{-0.3}$

Median mismatch (%) between injected and extracted glitches over 512 samples from each class. Bounds represent 1σ confidence interval.

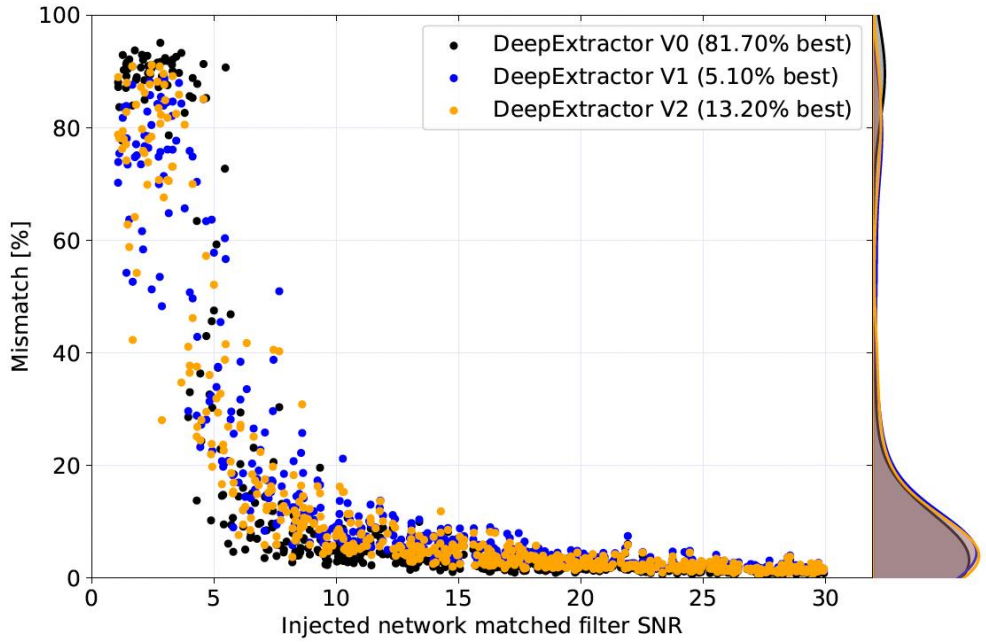
Main Findings

1. 2D model on high resolution STFT is best
2. Reconstruct a sample ≈ 0.1s (CPU)

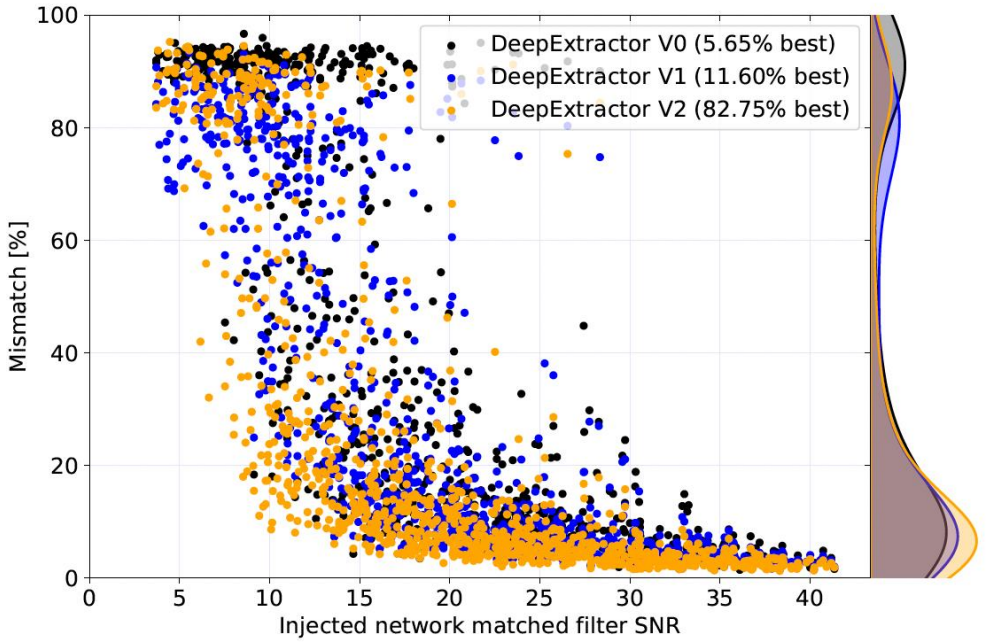
Clustering Glitch Reconstructions



Mismatch Performance vs SNR

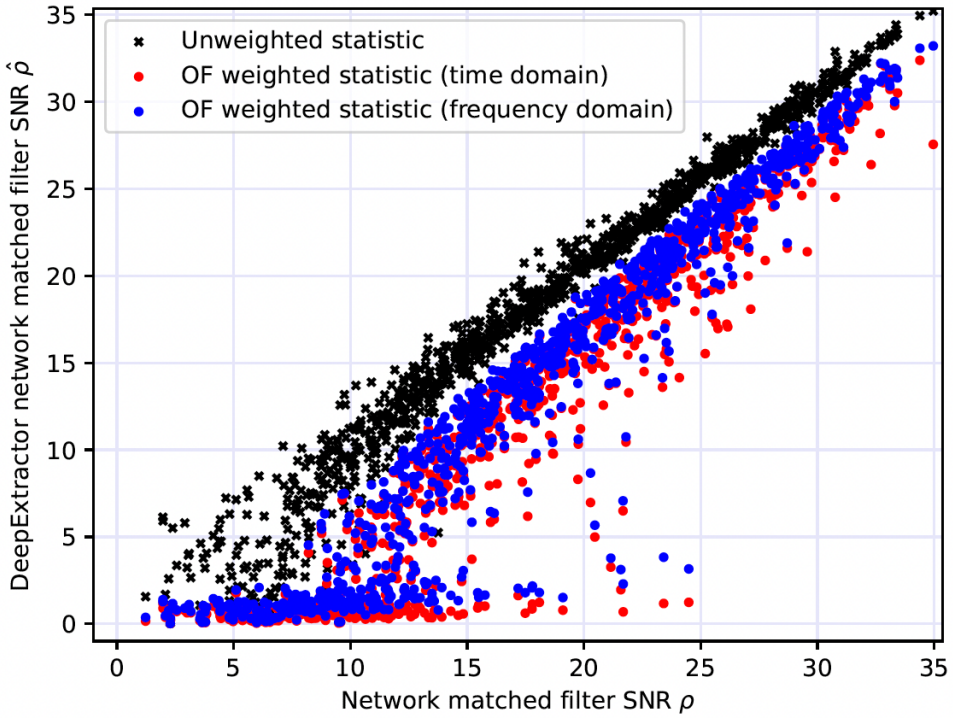


Signals

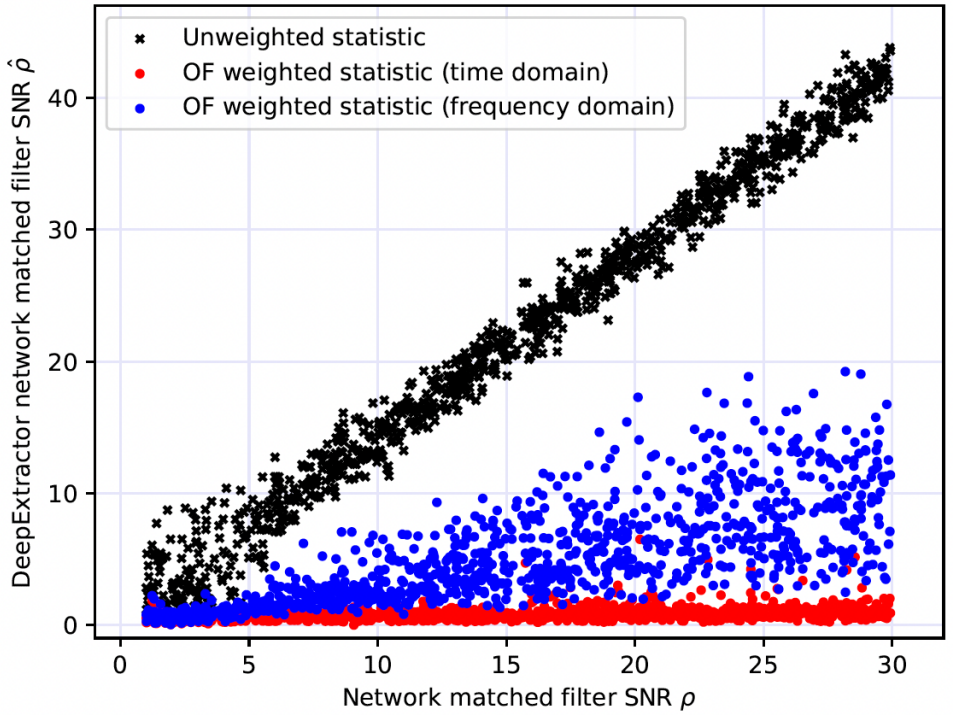


Glitches

Weighted Detection Statistic



Signals



Glitches