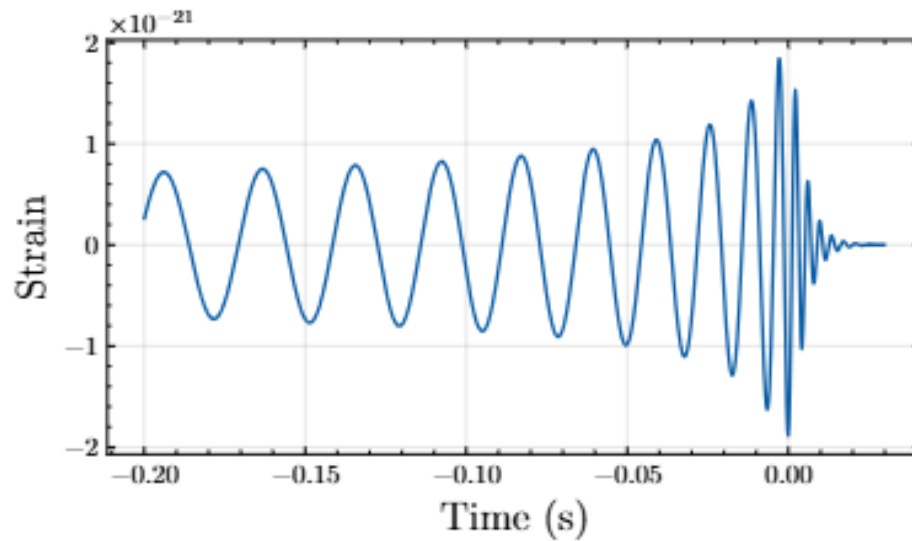


# DeepExtractor

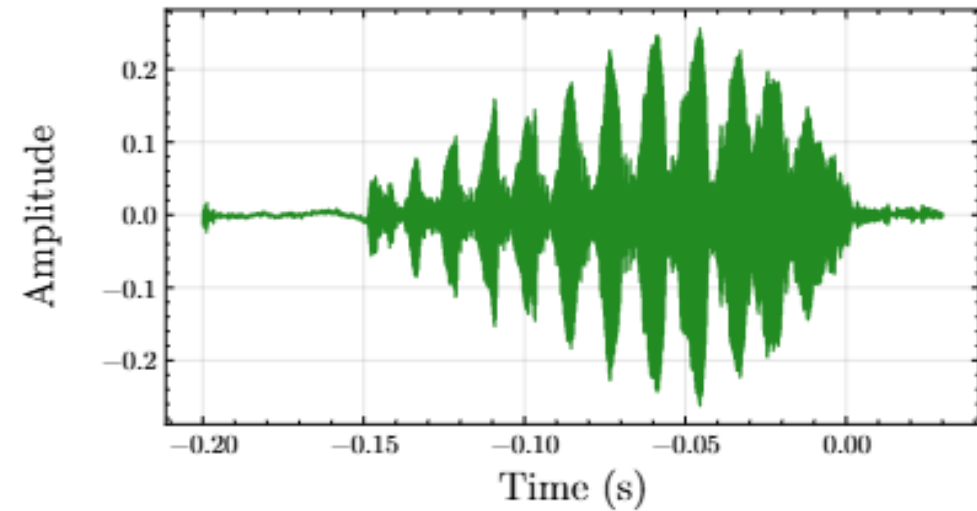
Glitch Mitigation and Template-Free Searches with  
Model-Agnostic Reconstruction using Deep Learning

Tom Dooney · EuCAIFCon · 26 August 2026

# GW Data vs Digital Sound Data

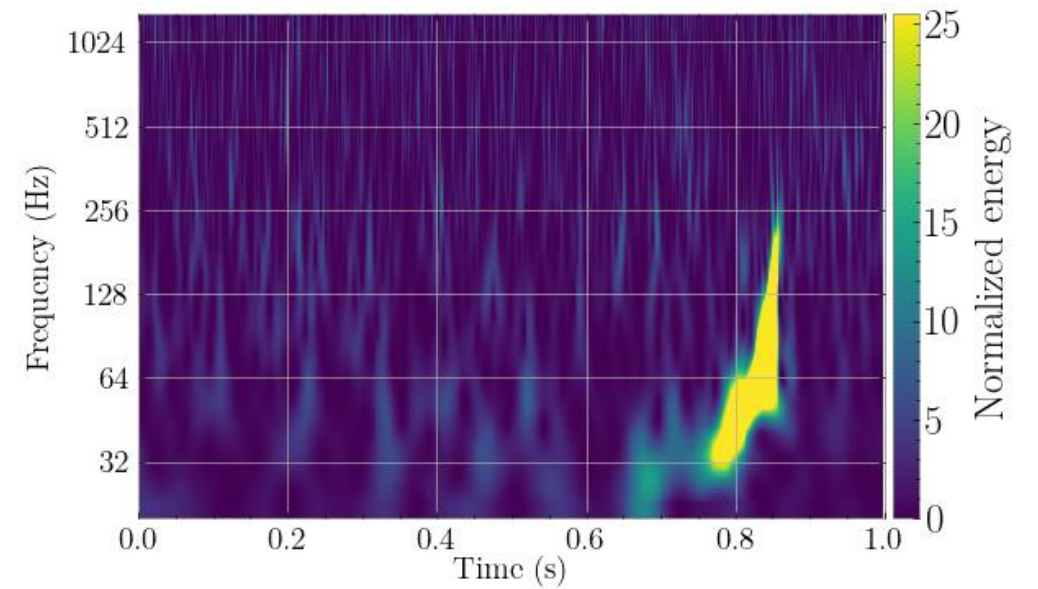
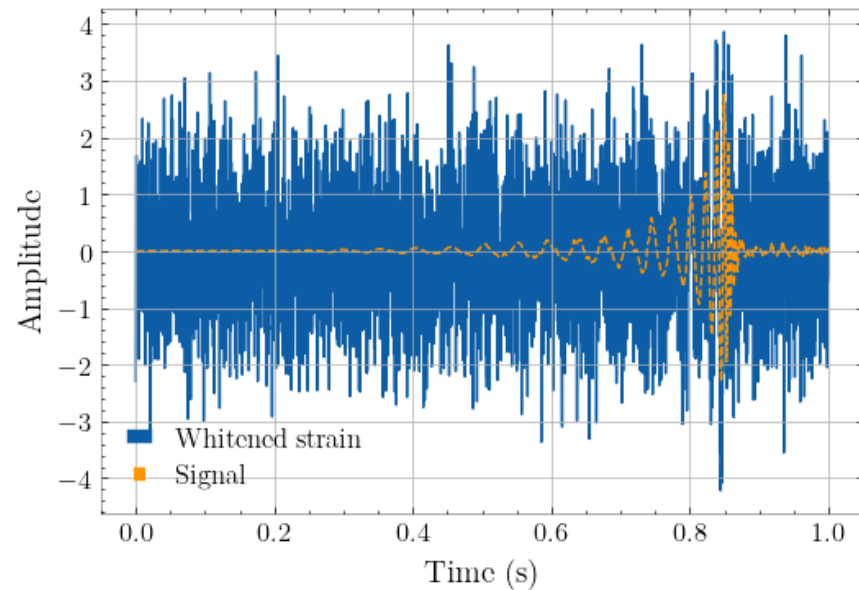


Gravitational Wave Chirp



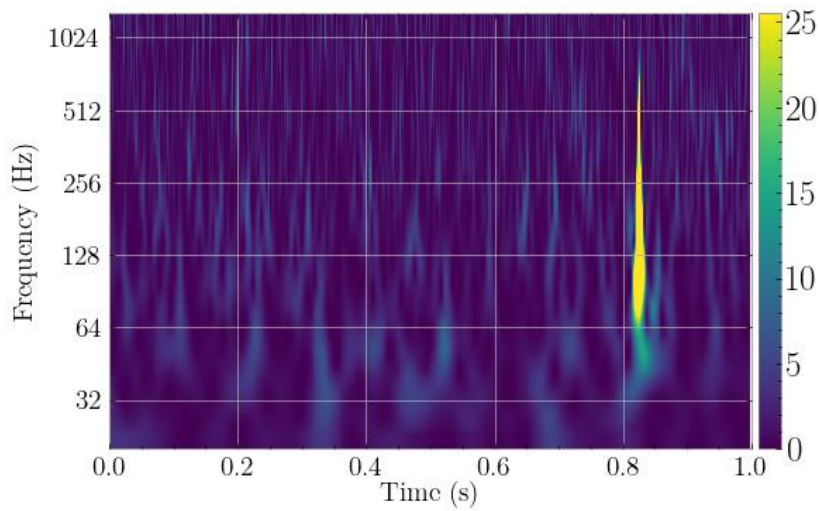
A Bird Chirp

# Gravitational Wave in Time-Frequency

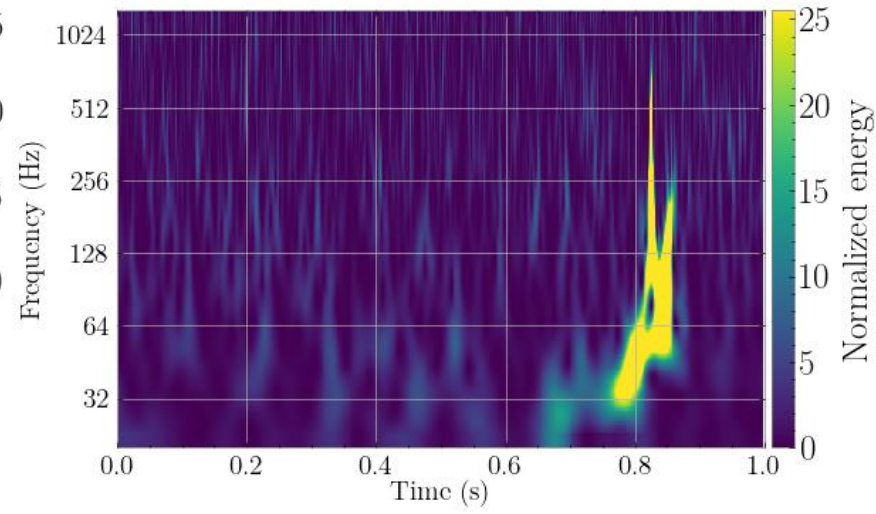


GW detector data  
(Noise+Signal)

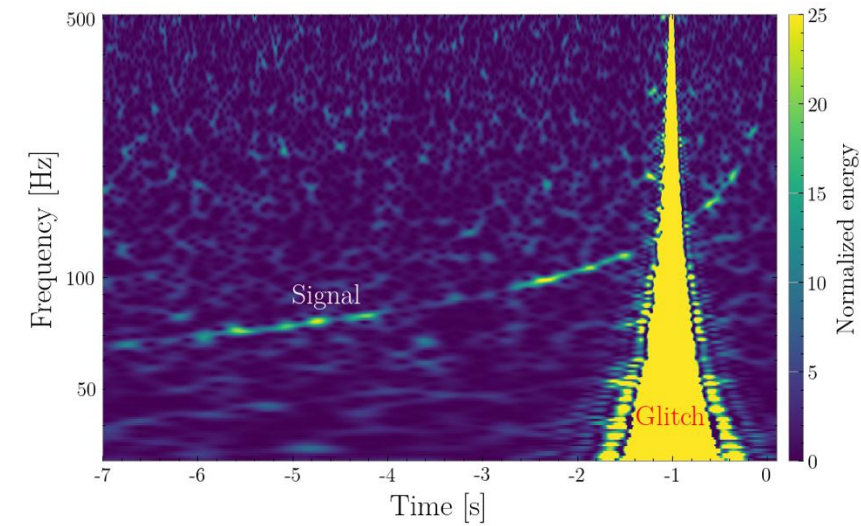
# Glitches in the Matrix



Noise + Glitch

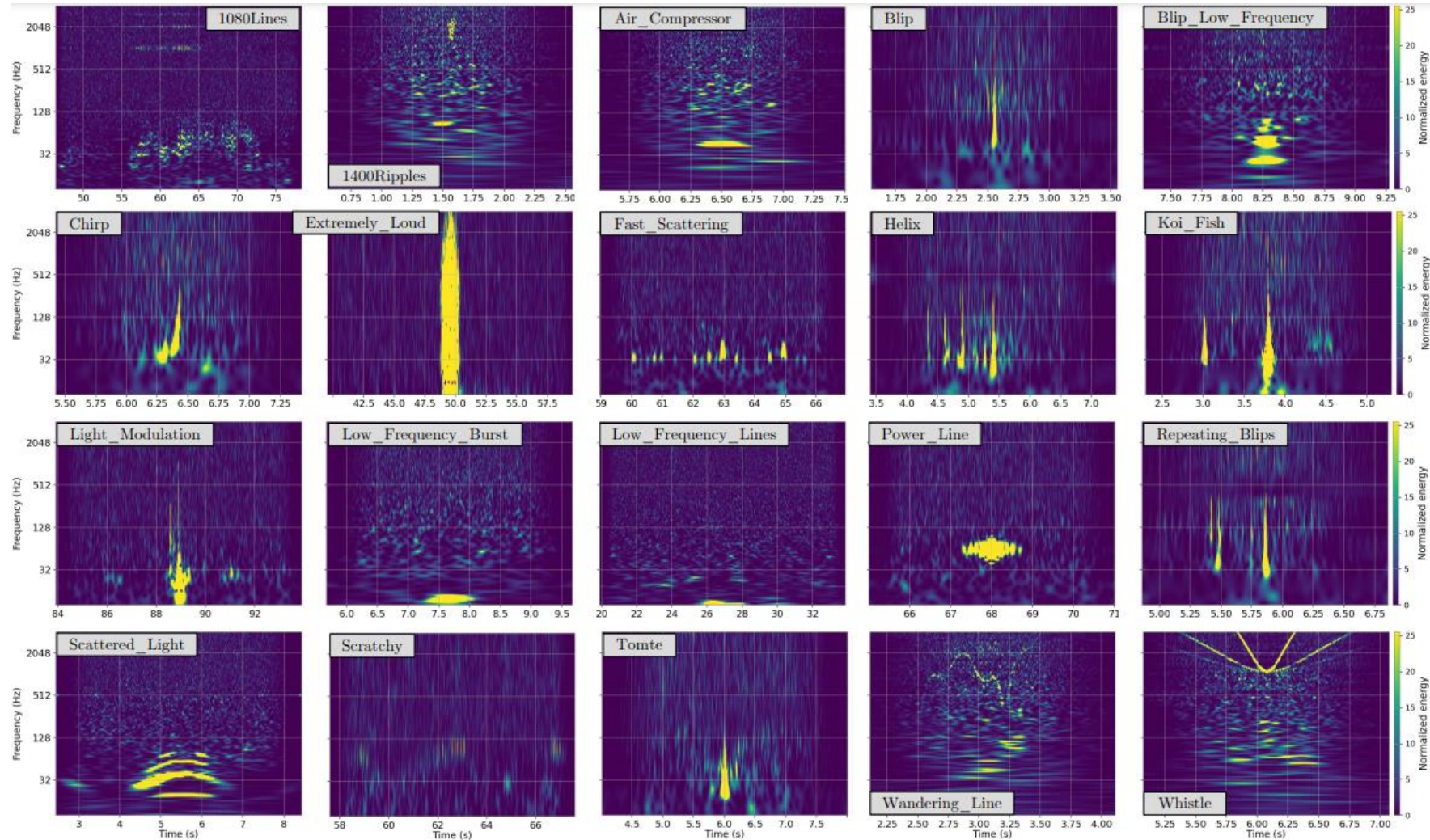


Noise + Signal+ Glitch



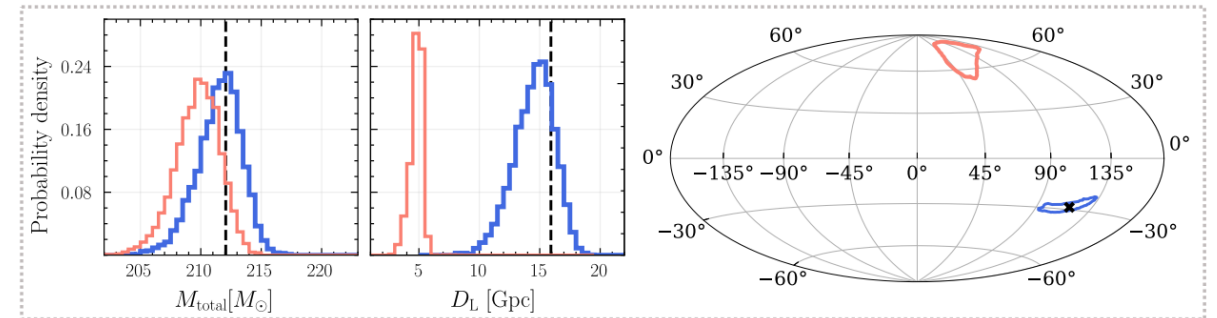
GW170817

# LIGO-Virgo Glitches (Gravity Spy)



# So What?

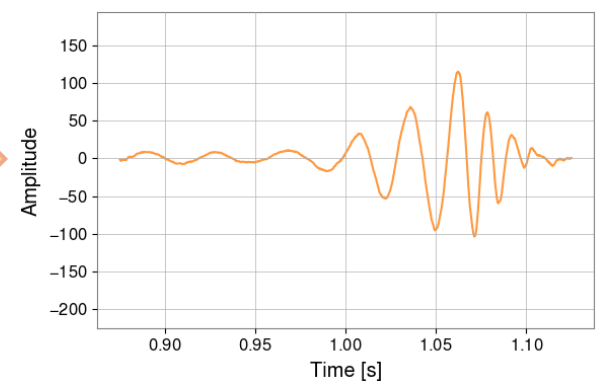
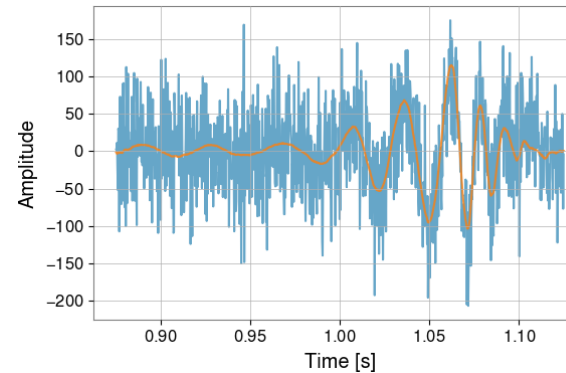
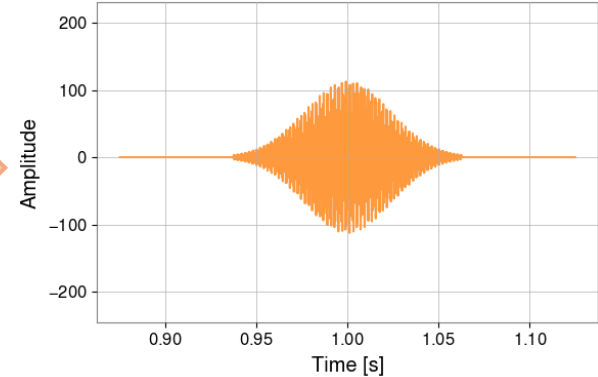
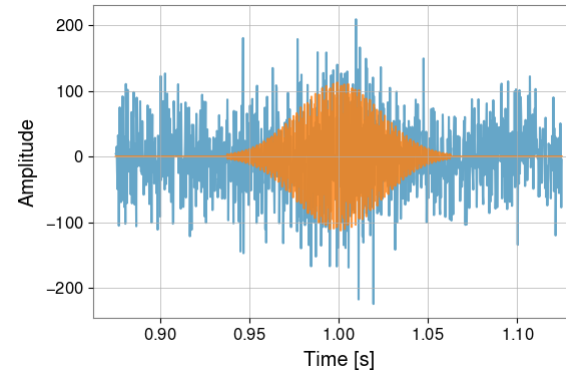
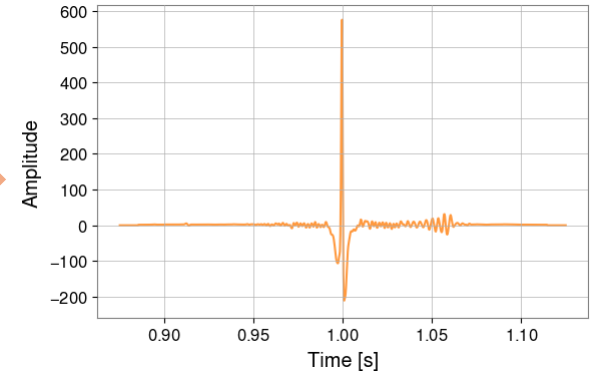
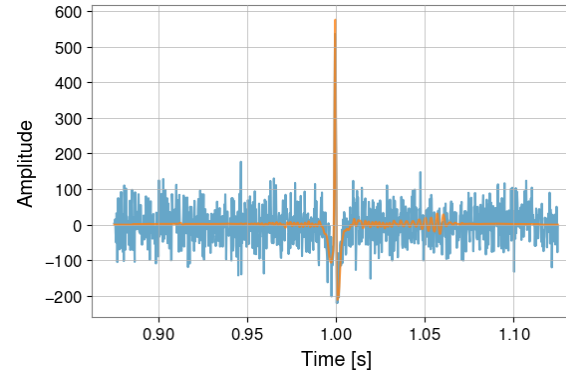
- Glitches hinder GW data analysis
  - Increase false positives
  - Bias Parameter Estimation
- BayesWave can mitigate them
  - Effective for some glitch classes but costly
  - Approximately 1 CPU hour to model one glitch
- We aim to build a fast glitch reconstruction framework
  - Mock Data Challenges
  - Pipeline validation
- We will also discuss glitch mitigation and searches



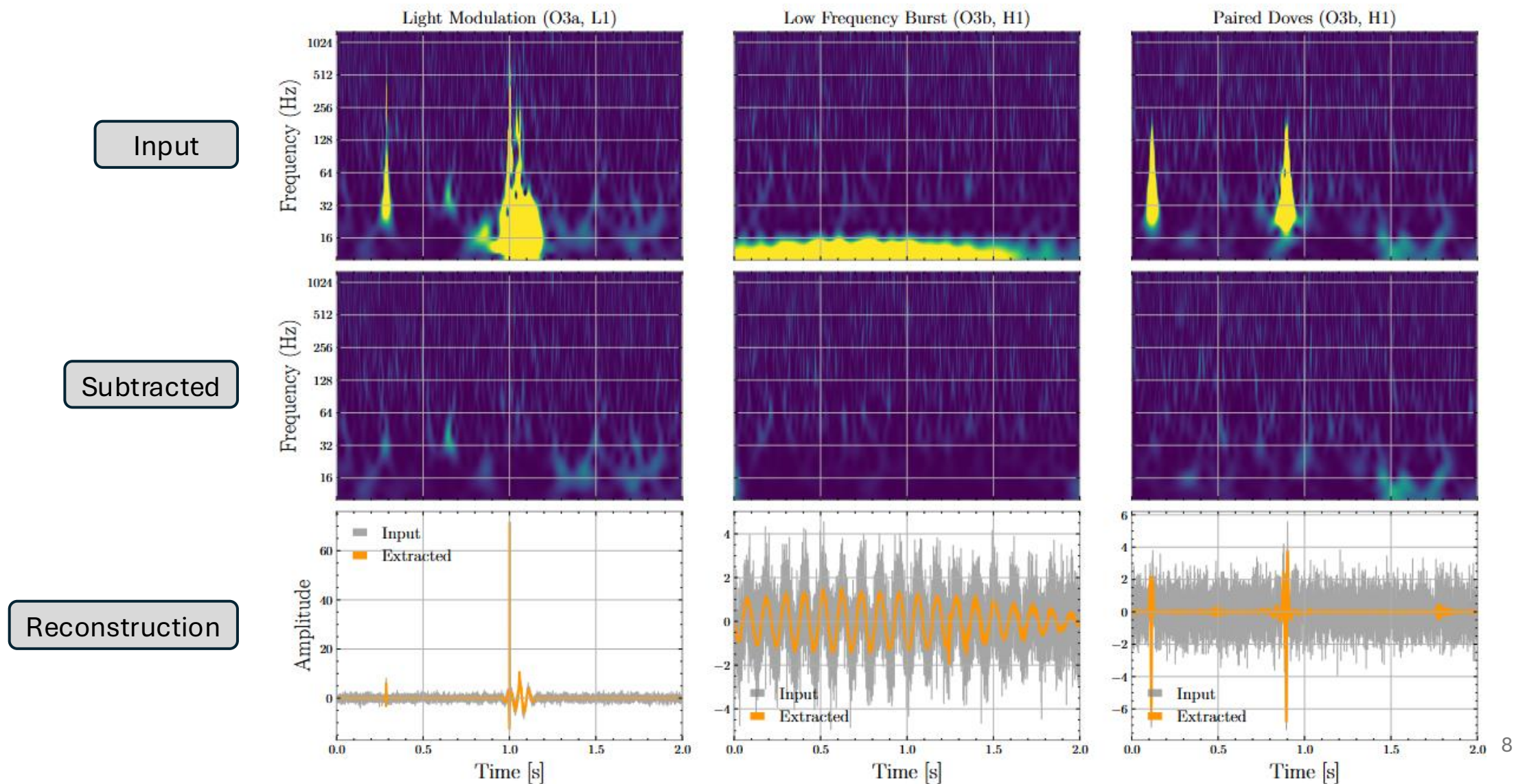
Glitch bias in parameter estimation  
([10.1103/l6tp-ykxp](https://arxiv.org/abs/10.1103/l6tp-ykxp))

# Goal

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



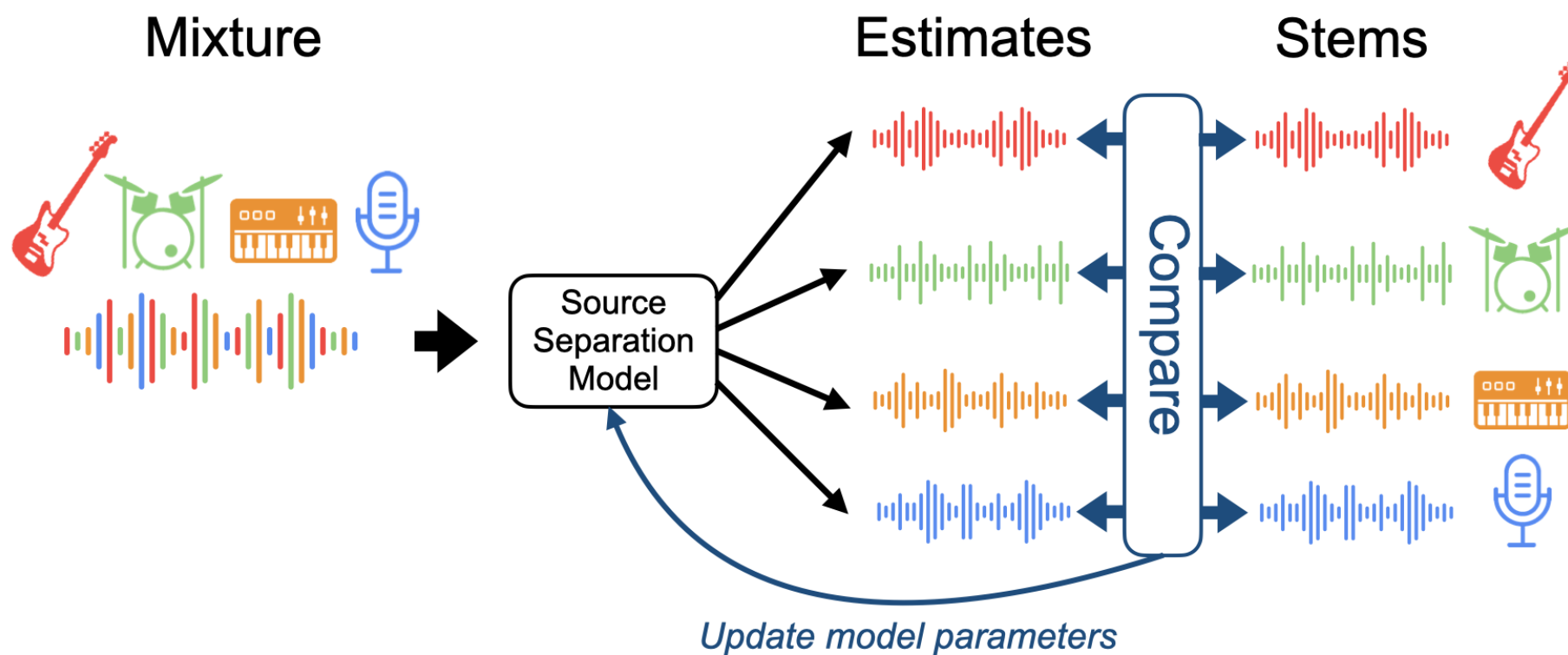
# LIGO glitch reconstructions with DeepExtractor



# DeepExtractor Applications

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

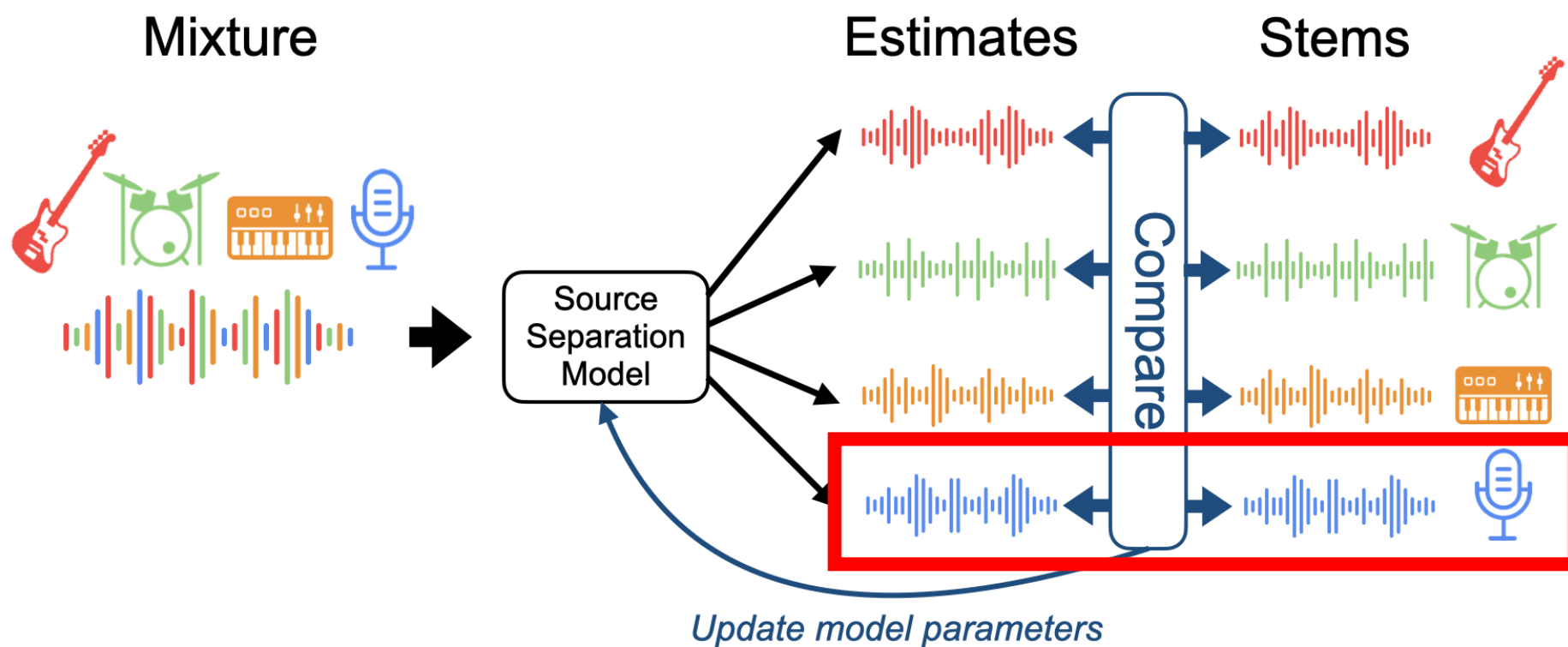
# Borrowing from the music industry



Music source separation model

Source: [Source Separation Tutorial](#)

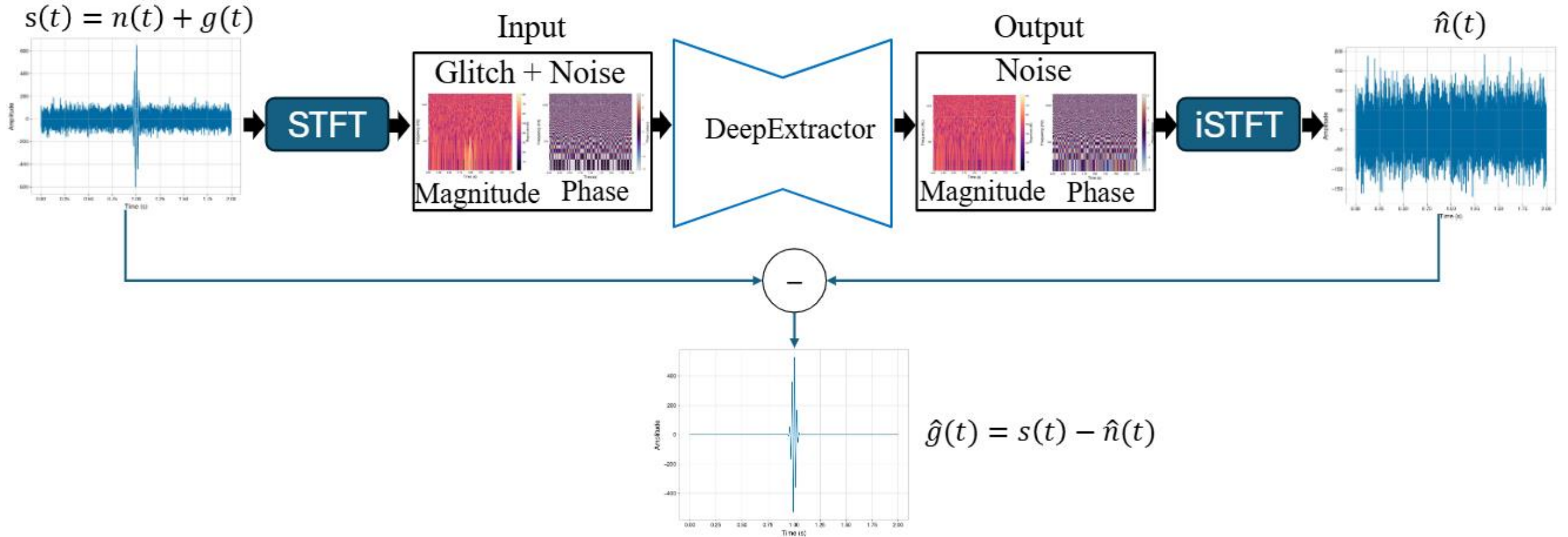
# Borrowing from the music industry



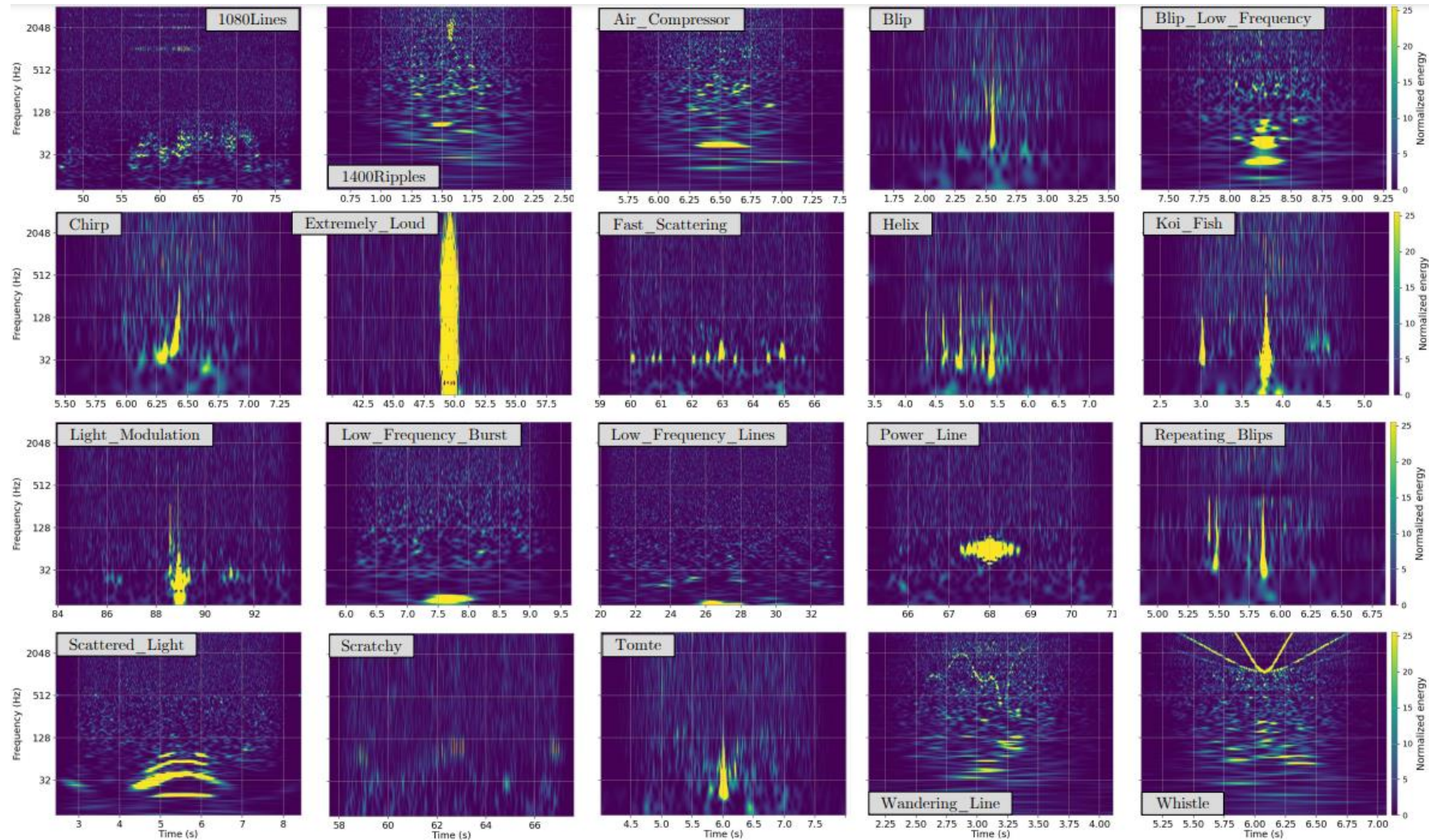
Music source separation model

Source: [Source Separation Tutorial](#)

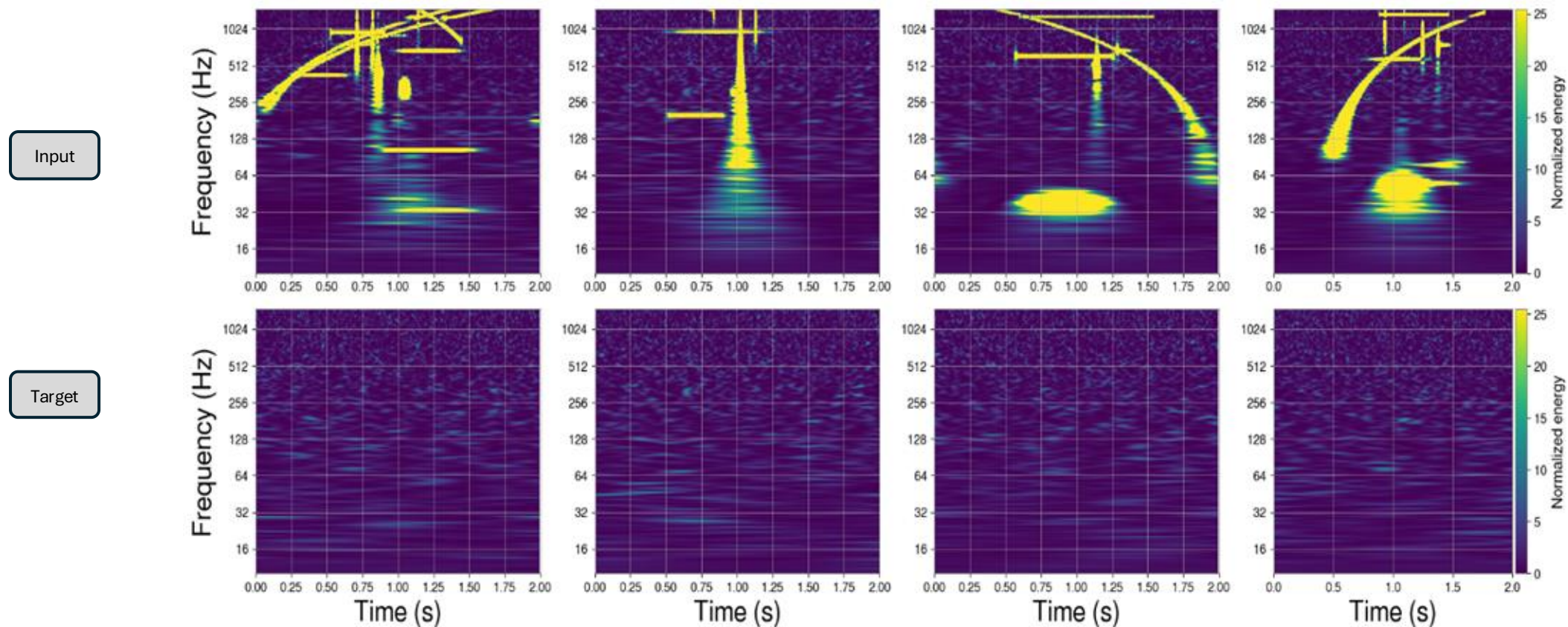
# DeepExtractor Framework



# How do we capture all Gravity Spy glitches?

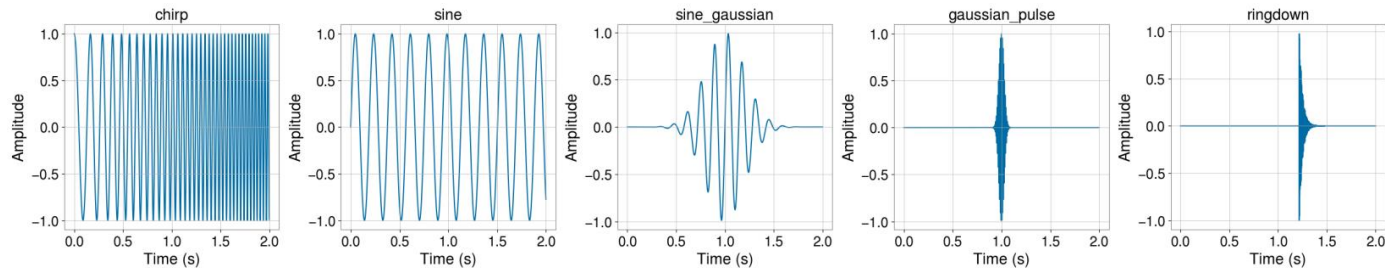


# We use proxy glitches...

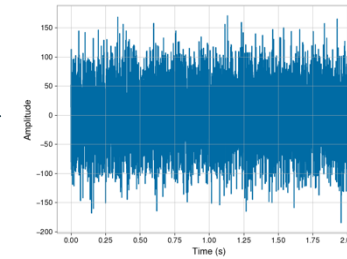


# We use proxy glitches...

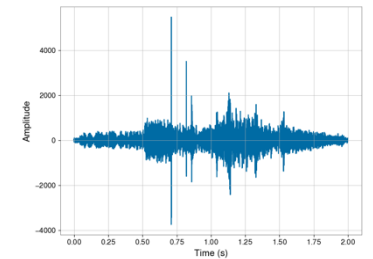
Basis Glitches



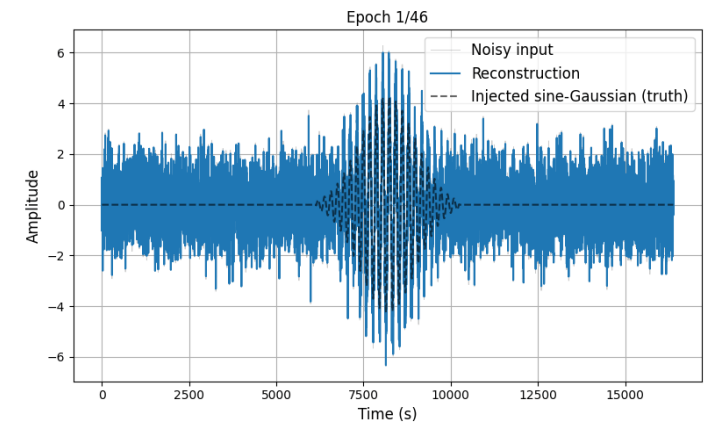
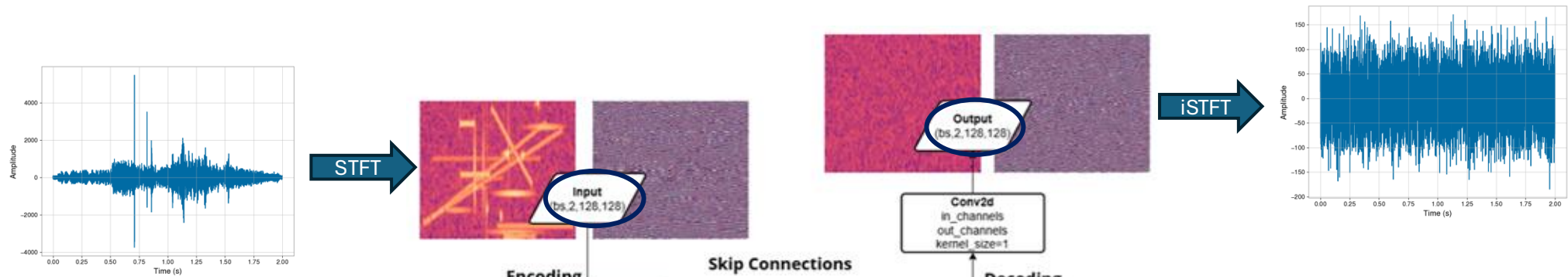
Target



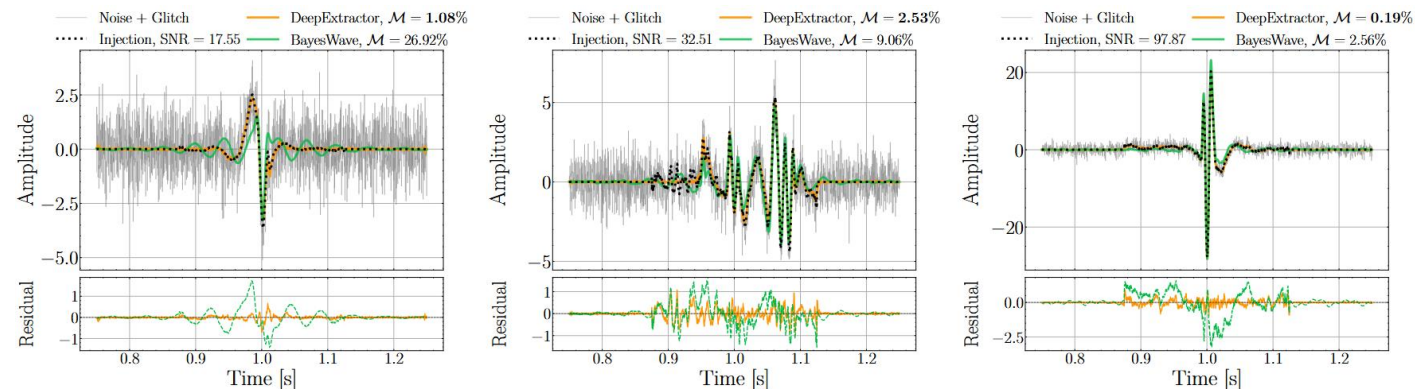
Input



# U-Net Architecture

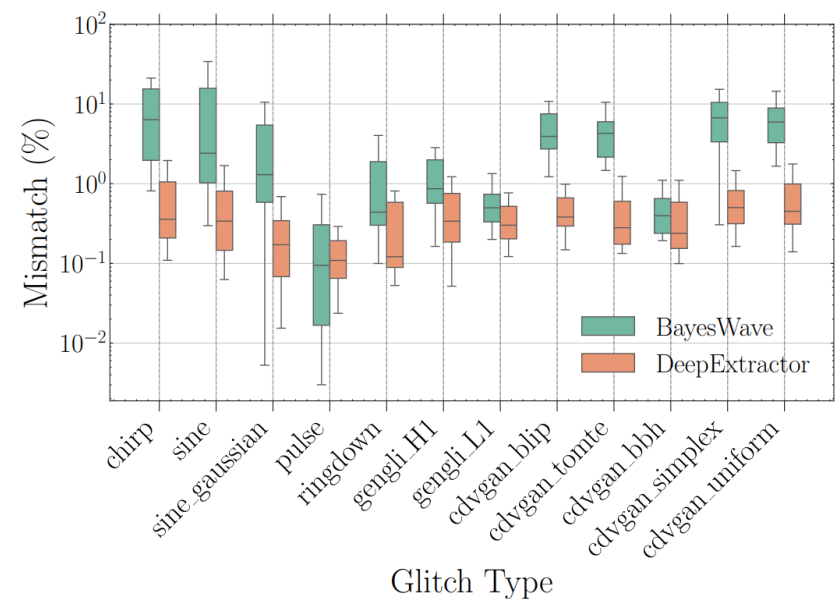


# Comparison with BayesWave – Mismatch performance



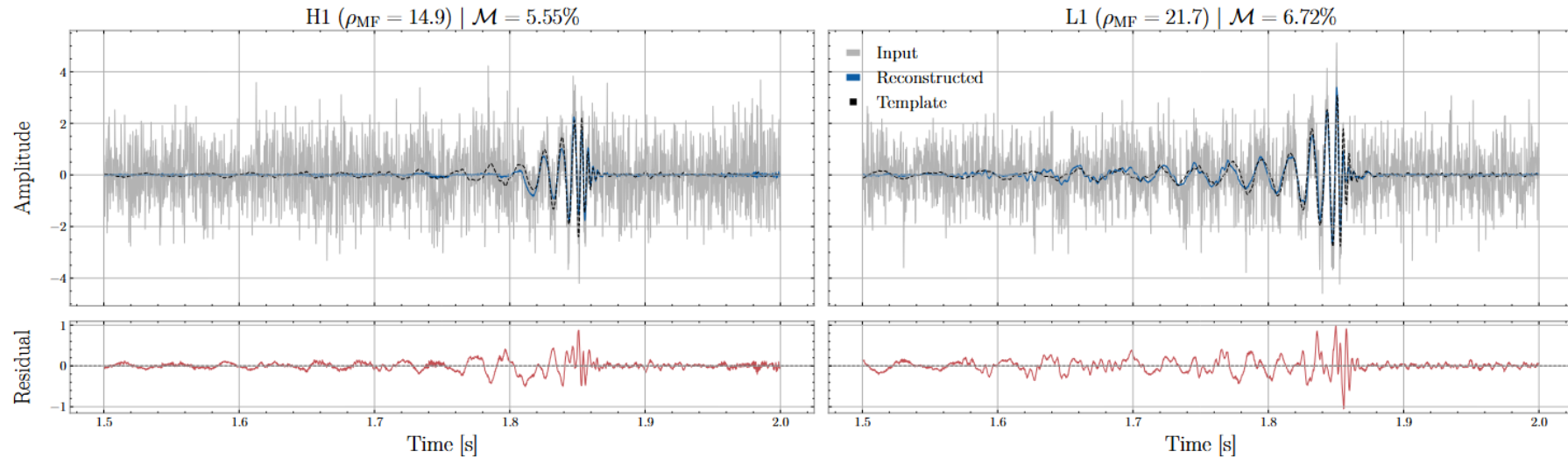
Example reconstructions with BayesWave and DeepExtractor

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

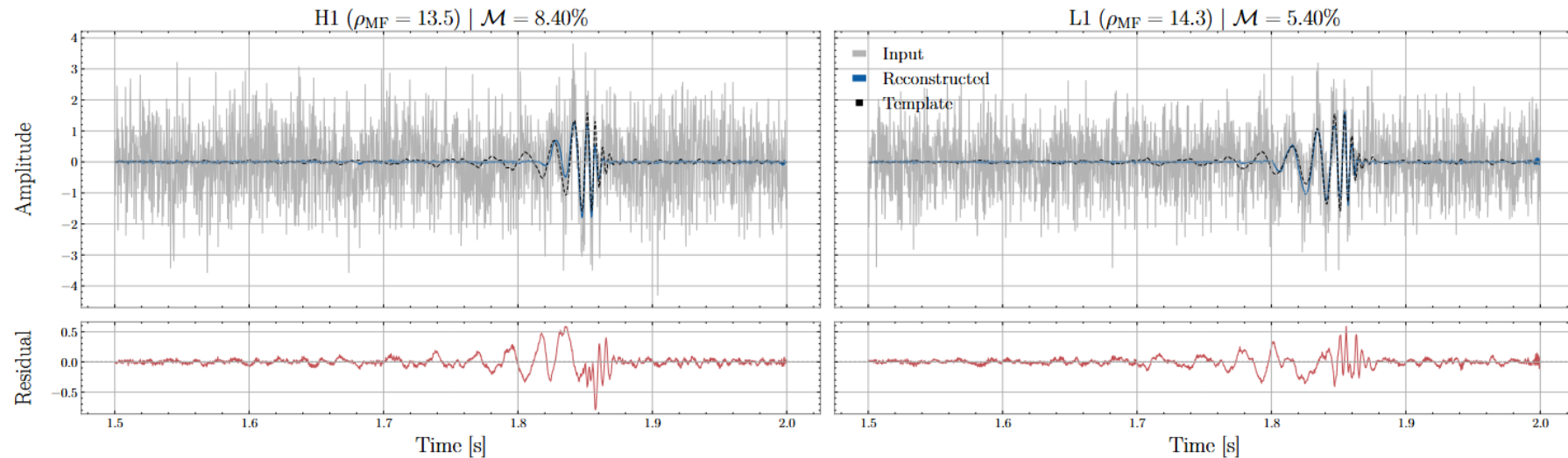


Mismatch between injected and reconstructed samples (30 samples per class)

# Reconstructing Signals during LIGO's O3 run



(b) *GW200129\_065458*

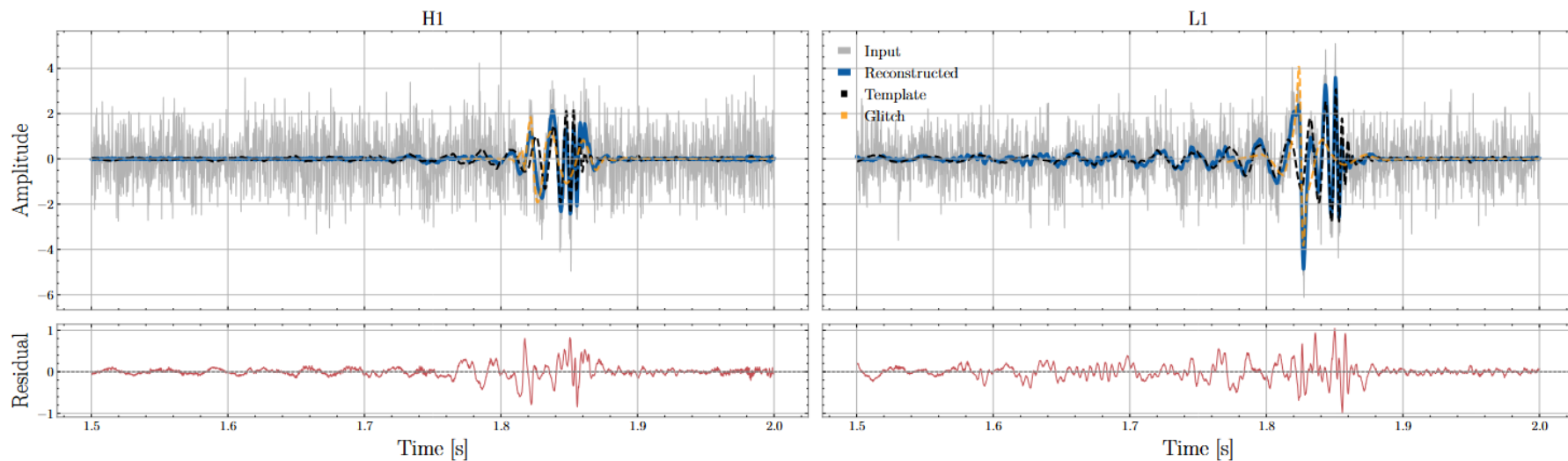


(c) *GW200224\_222234*

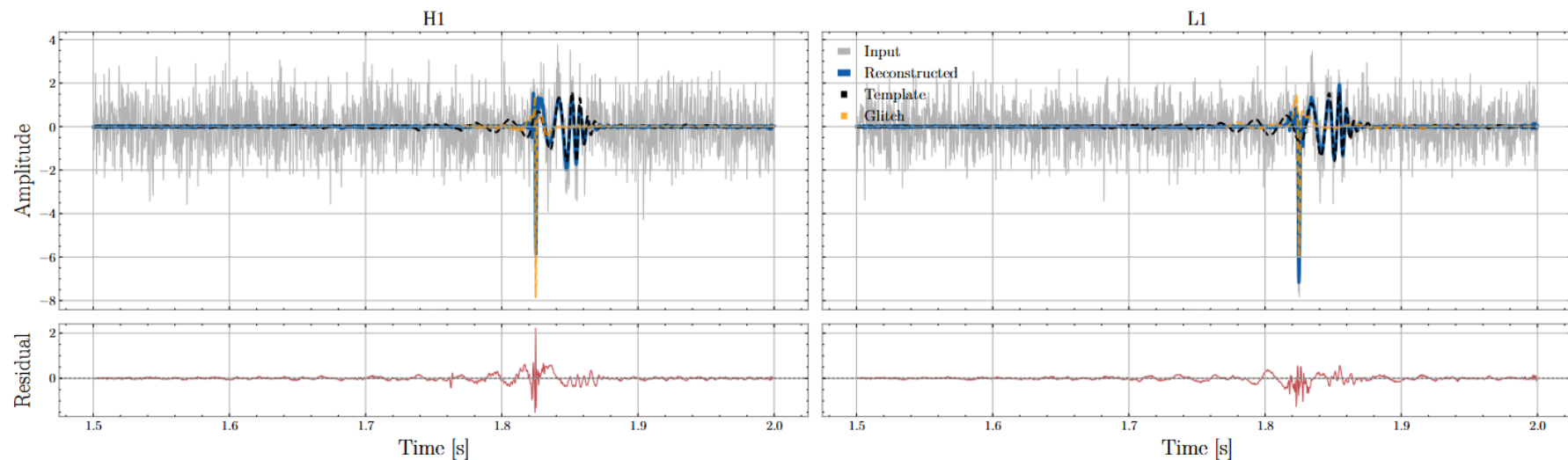
Reconstructions  
can be used for  
template-free  
searches!

Come by Poster 27  
to see how!

# Separating signals and glitches



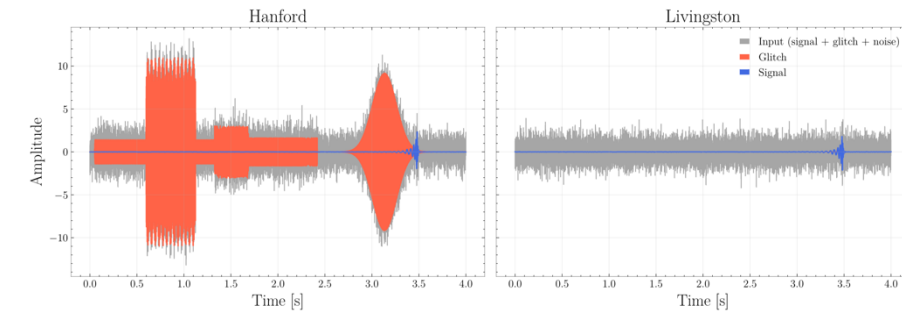
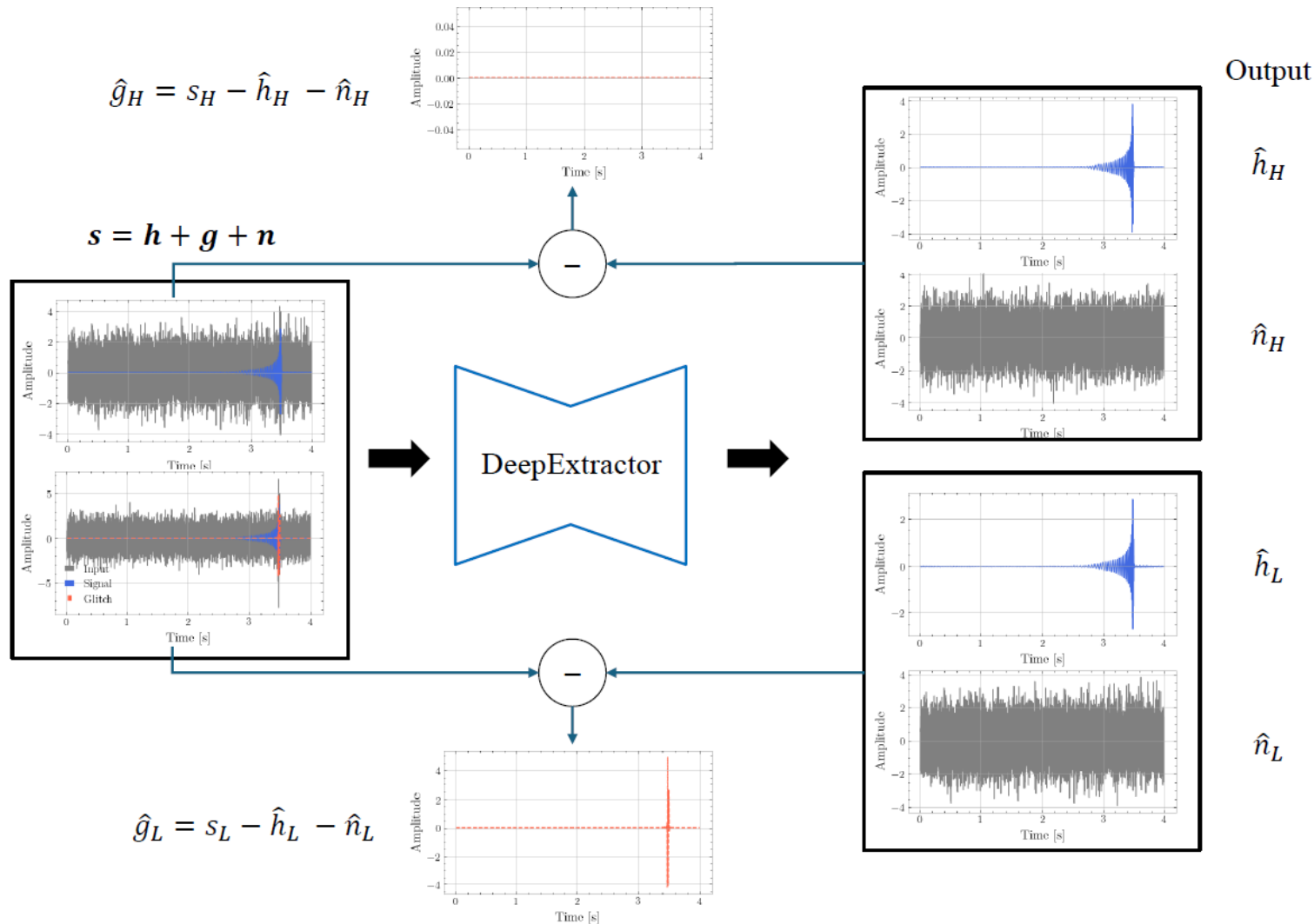
(b) *GW200129-065458*



(c) *GW200224-222234*

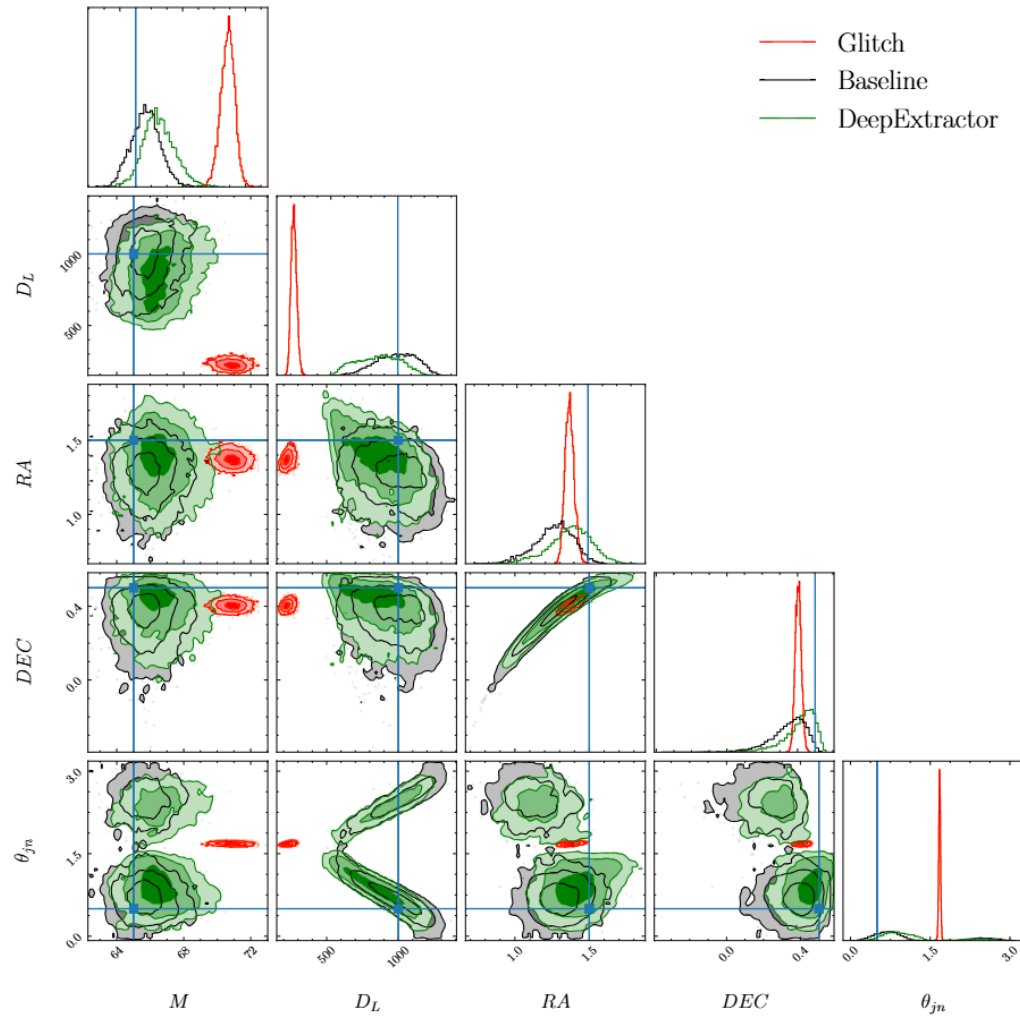
Come by Poster 27  
to see how we can  
separate signals,  
glitches and noise!

# Signal and glitch separation – Increasing the number of instruments

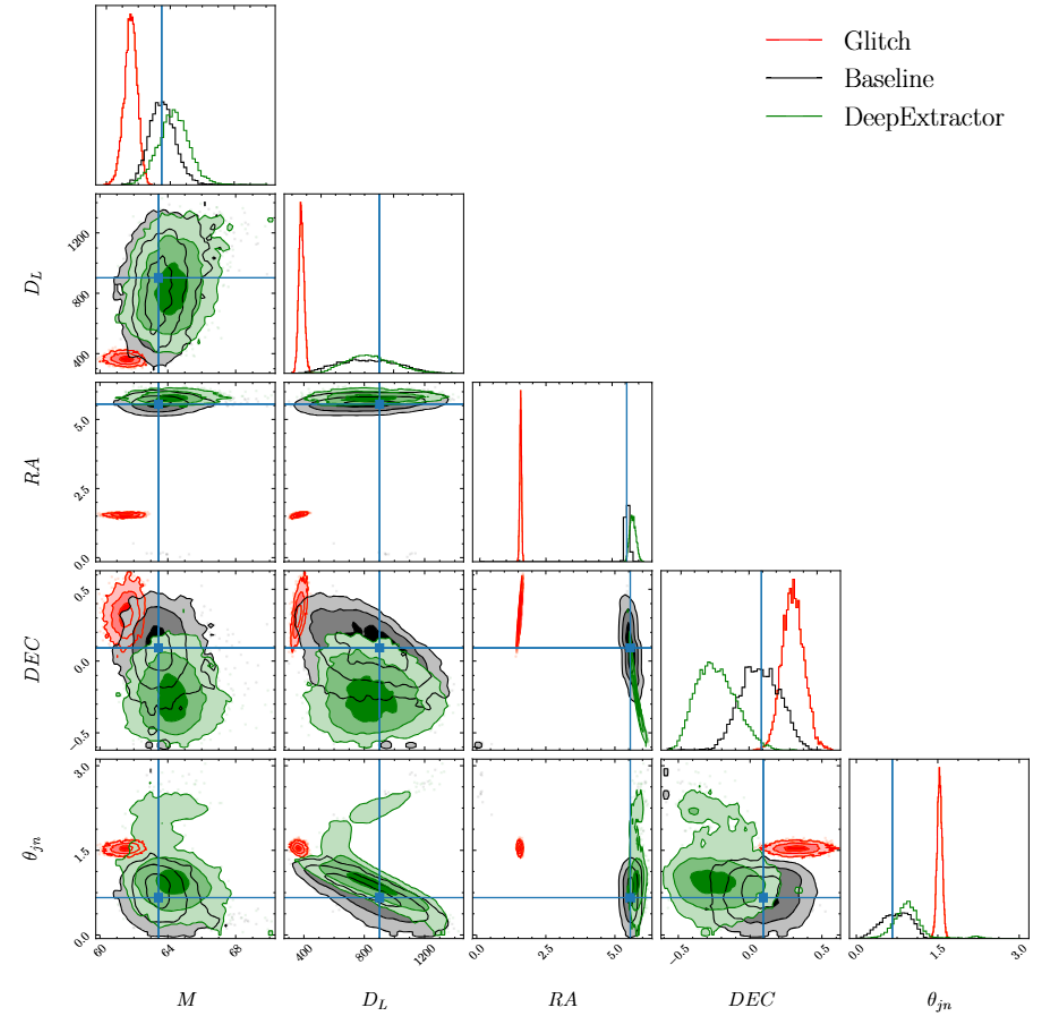


A training sample for DeepExtractor signal+glitch separation. We accumulate 1.2M samples.

# Reducing glitch bias in parameter estimation



*GW150914-like event*



*GW200129-like event*

## Poster 27!



# DeepExtractor

Glitch Mitigation and Template-Free Searches with Model-Agnostic Reconstruction using Deep Learning

Tom Dooney<sup>1,2</sup> · Bo Ribbens<sup>1</sup> · Harsh Narula<sup>1,2</sup> · Daniel Stanley Tan<sup>1</sup> · Stefano Bromuri<sup>1</sup> · Chris Van Den Broeck<sup>1,2</sup>

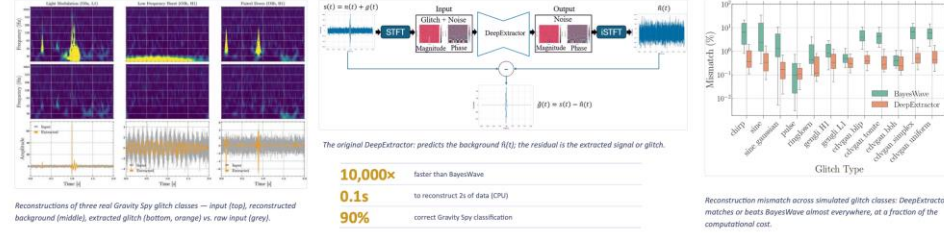
<sup>1</sup>Nikhef <sup>2</sup>Utrecht University <sup>3</sup>Open Universiteit



Read the Docs

### Model-Agnostic Glitch Reconstruction

DeepExtractor reconstructs arbitrary signals or glitches with power above the detector noise floor by learning to model and subtract the underlying detector noise rather than the signal itself.



### Online, Template-Free Searches

DeepExtractor's morphology-agnostic, template-free reconstruction offers a low-latency alternative to matched-filtering and unmodelled search pipelines.

- A detection statistic is built directly from the reconstructed residual — no waveform templates required
- Reconstructs 2s of data in 0.1s (CPU), suited to online, low-latency operation
- Separates real signals from glitch and noise backgrounds by measuring coincidence

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

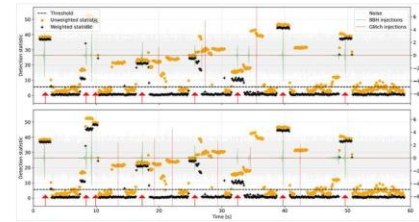
$$\log \Lambda_{\text{network}} = \sum_{i \in \{L, H\}} \log \Lambda_i$$

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

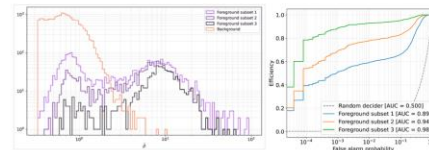
$$\hat{\rho} = \omega_{L,H} \times \hat{\rho}_{\text{network}}$$

$$\omega_{L,H} = \max_{\tau \in [5, 75] \text{ ms}} \left| \frac{1}{N(\tau)} \sum_i \hat{h}_i(t_i) \hat{h}_H(t_i + \tau) \right|$$

Network log-likelihood statistic, converted to an SNR-like  $\hat{\rho}$ , re-weighted by an inter-detector coincidence factor  $\omega$ .



Simulated 60 s of two-detector data: the weighted statistic (black) suppresses glitches (red) that inflate the unweighted statistic (orange), while BBH injections (green) remain above threshold.

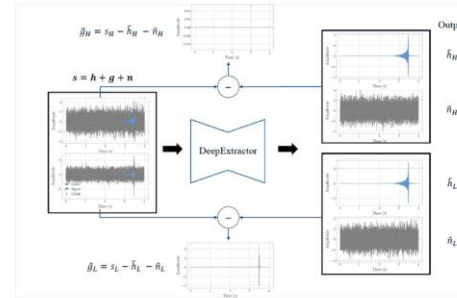


| Redshift         | Signal length | Matched filter SNR | Number of samples |
|------------------|---------------|--------------------|-------------------|
| Subset 1 $z < 2$ | $t < 10$      | No limitation      | $N = 2502$        |
| Subset 2 $z < 2$ | $t < 2$       | No limitation      | $N = 1384$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

Detection-statistic distributions for background and three population subsets ( $p_0, p_1, p_2$ ), with corresponding ROC curves showing search efficiency vs. false alarm probability for each subset. Population subsets defined by redshift, signal length, and matched-filter SNR cuts below.

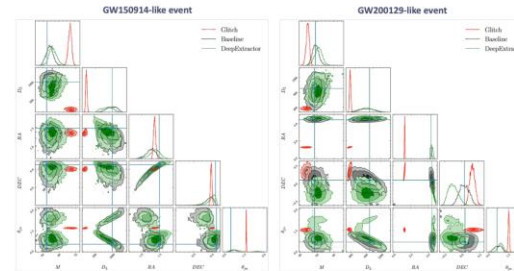
### Signal + Glitch Separation

We extend DeepExtractor to the case where a signal and a glitch overlap in time and frequency, jointly modelling signal and background components alongside so the glitch can be reconstructed via subtraction.



Extended framework: jointly reconstructs signal  $h$  and background  $B$  per detector; each residual  $B = s - h$  recovers the glitch.

- Joint model outputs the signal, glitch, and background per detector
- The separation step recovers a cleaner signal for downstream analysis
- Reduces glitch bias in parameter estimation on the recovered signal



Recovered posteriors after separation (green) track the glitch-free baseline (black) closely, away from the glitch-biased posterior (red).

# Summary & Questions

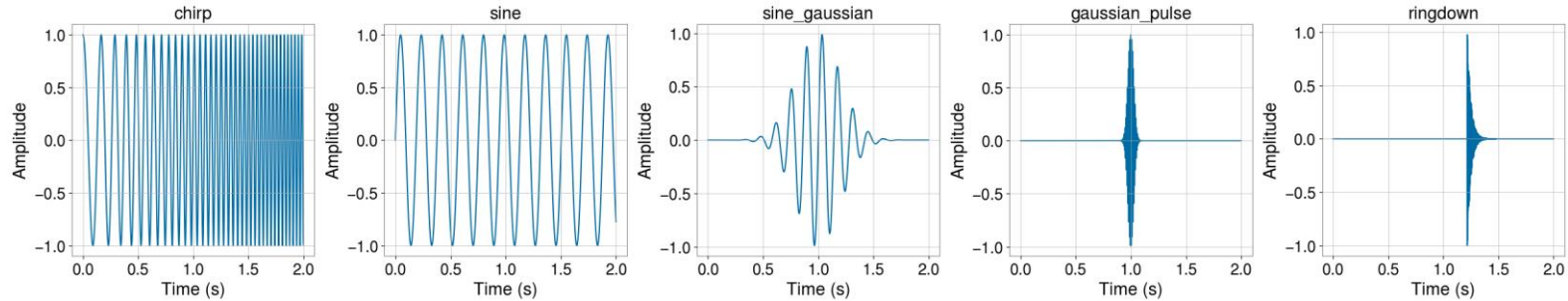
- Time-domain **reconstruction** of arbitrary GW signals and glitches
- **Interpolation** to unseen classes
  - Trained with diverse **combinations** of **proxy glitch** classes
  - **Residual** approach: **predict** the **noise** and **subtract** to give glitch
- Surpass BayesWave glitch recovery performance
  - 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**, looking for suggestions on features
- **Background** estimation for **HEP/Astro?**



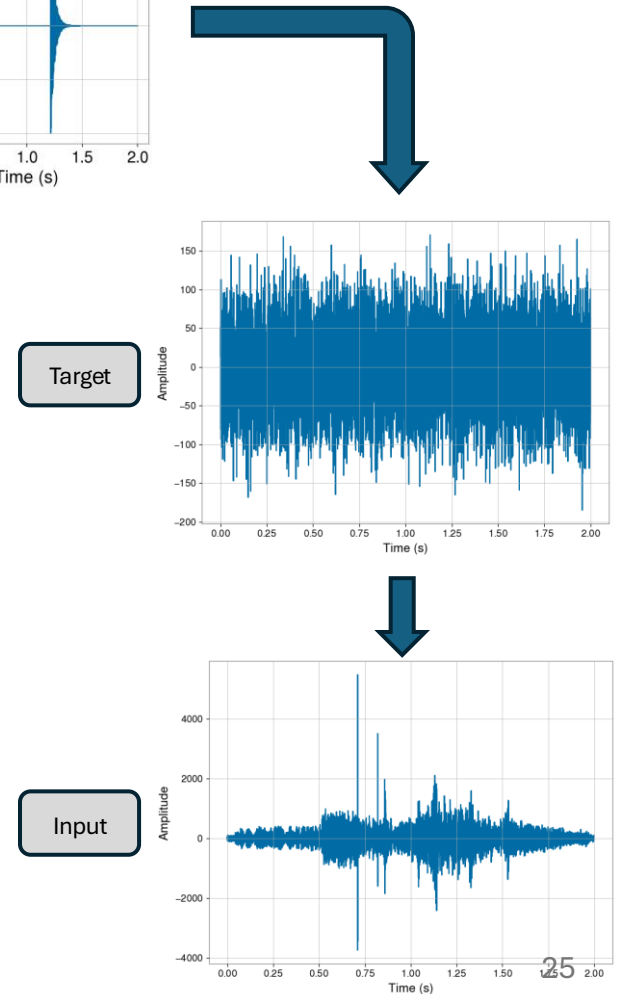
Read the docs!

# Appendix

# Synthetic 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



# Results – Simulated experiments

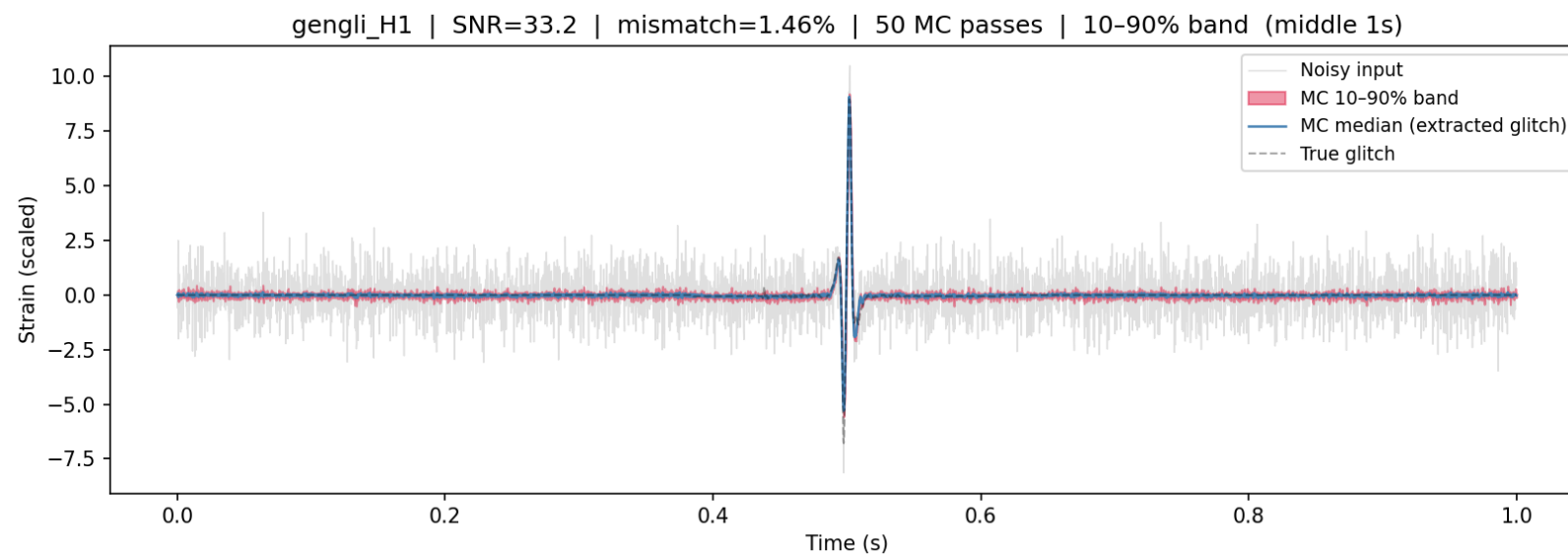
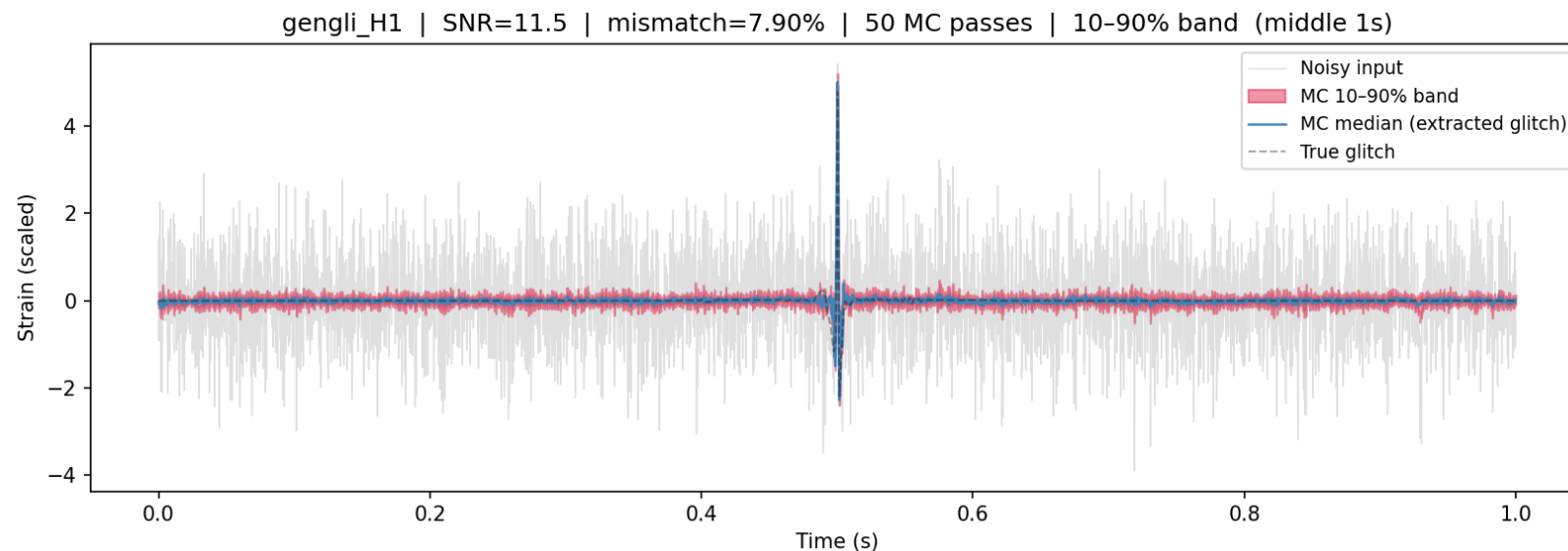
| Glitch Type    | UNET1D                              | DnCNN1D                                | Autoencoder1D                       | Autoencoder2D                          | DeepEx. (65x129)                    | DeepEx. (129 <sup>2</sup> )         | DeepEx. (257 <sup>2</sup> )         |
|----------------|-------------------------------------|-----------------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

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

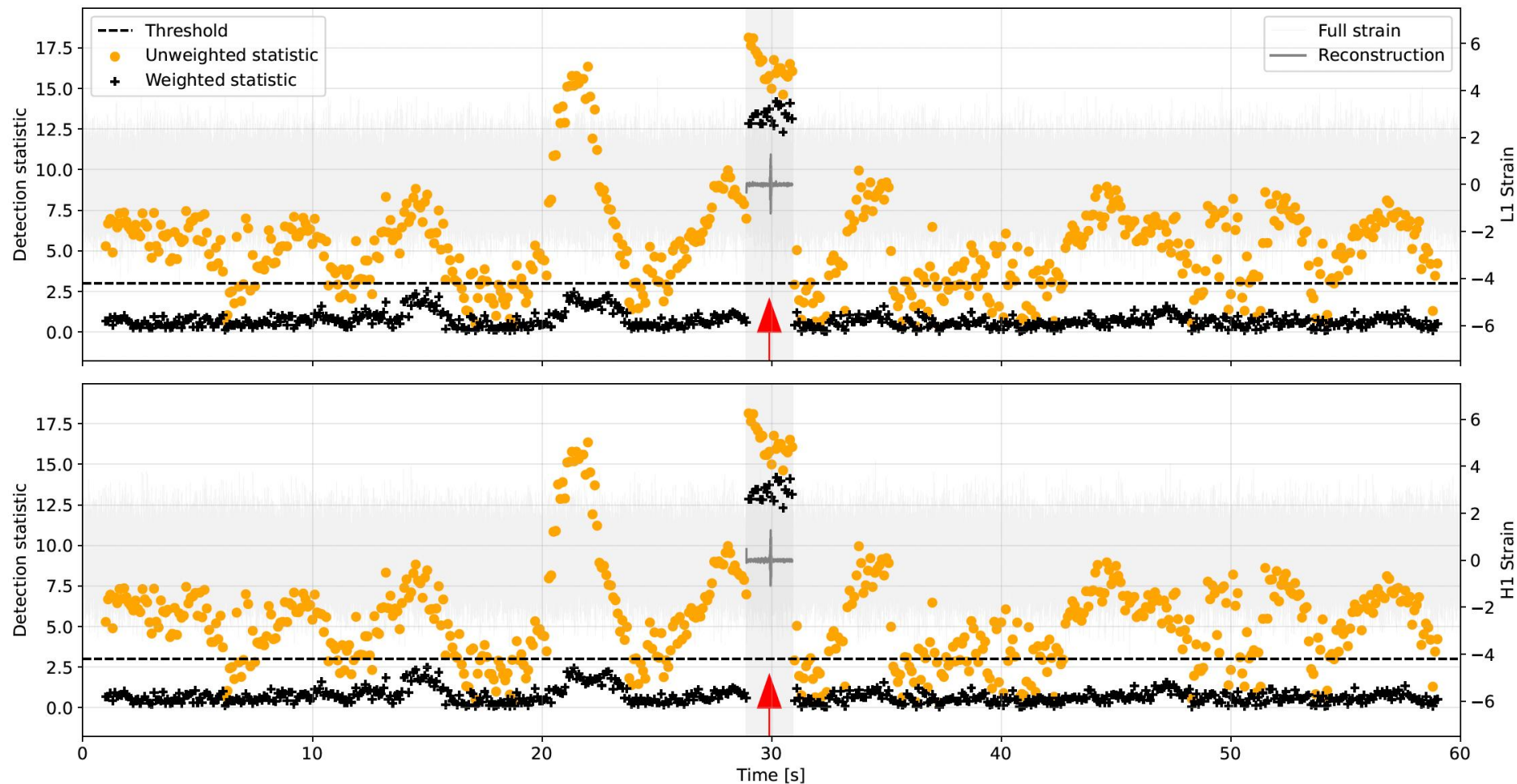
## Main Findings

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

# Uncertainty Quantification with Dropout

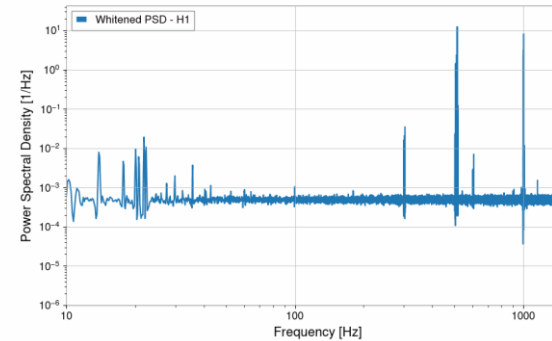
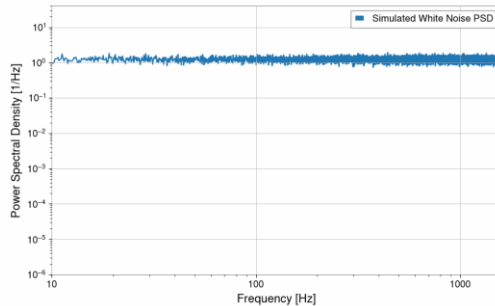


# Detecting GW231123



# 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



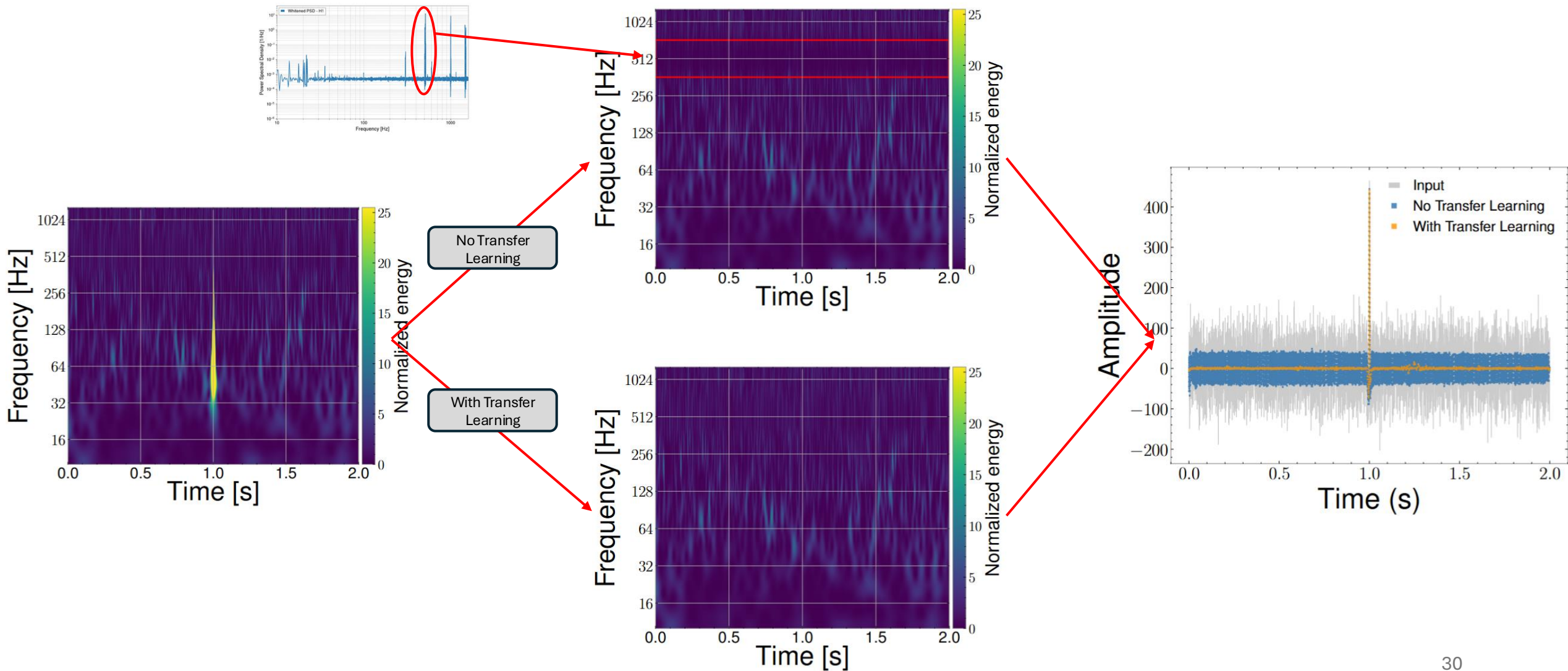
Pre-Trained  
DeepExtractor

Additional  
Knowledge (PSD)

Transfer learning

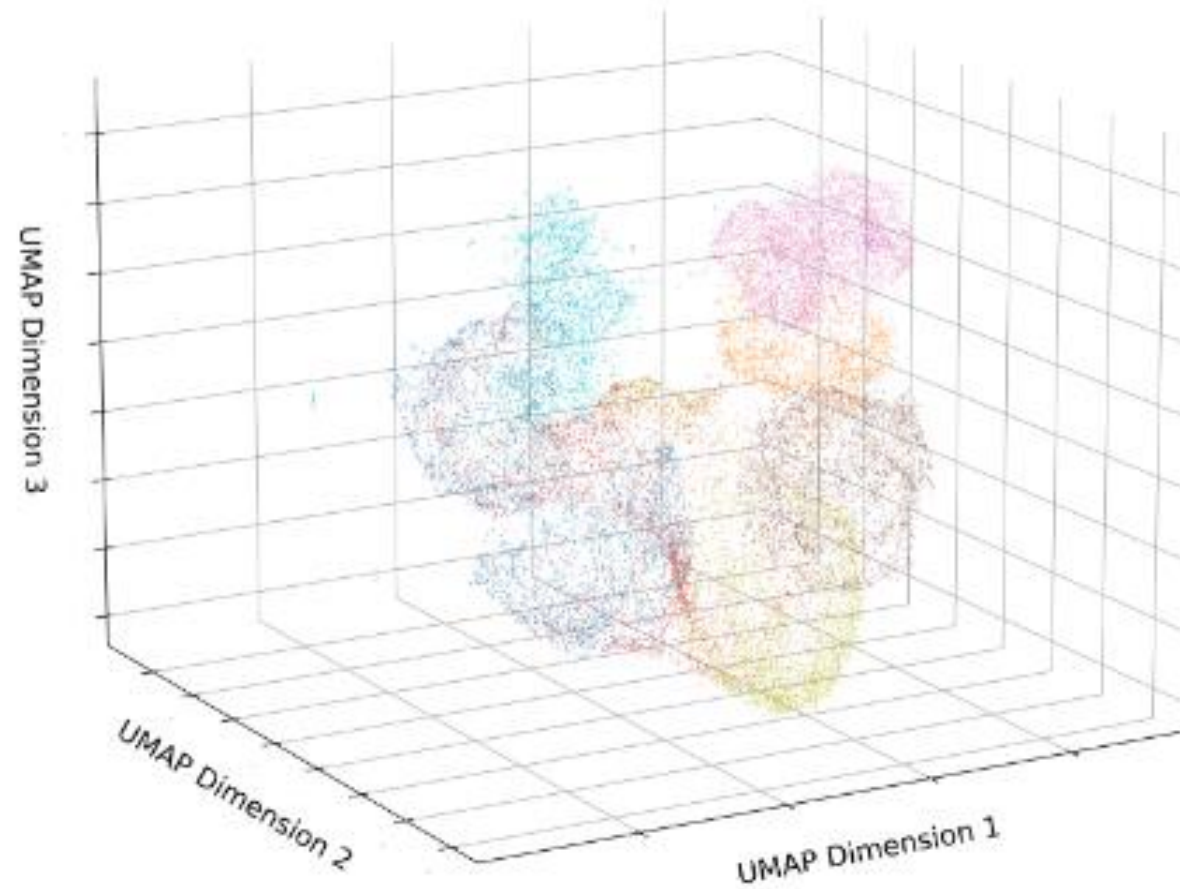
Fine-tuned  
DeepExtractor

# Improvements from Transfer Learning

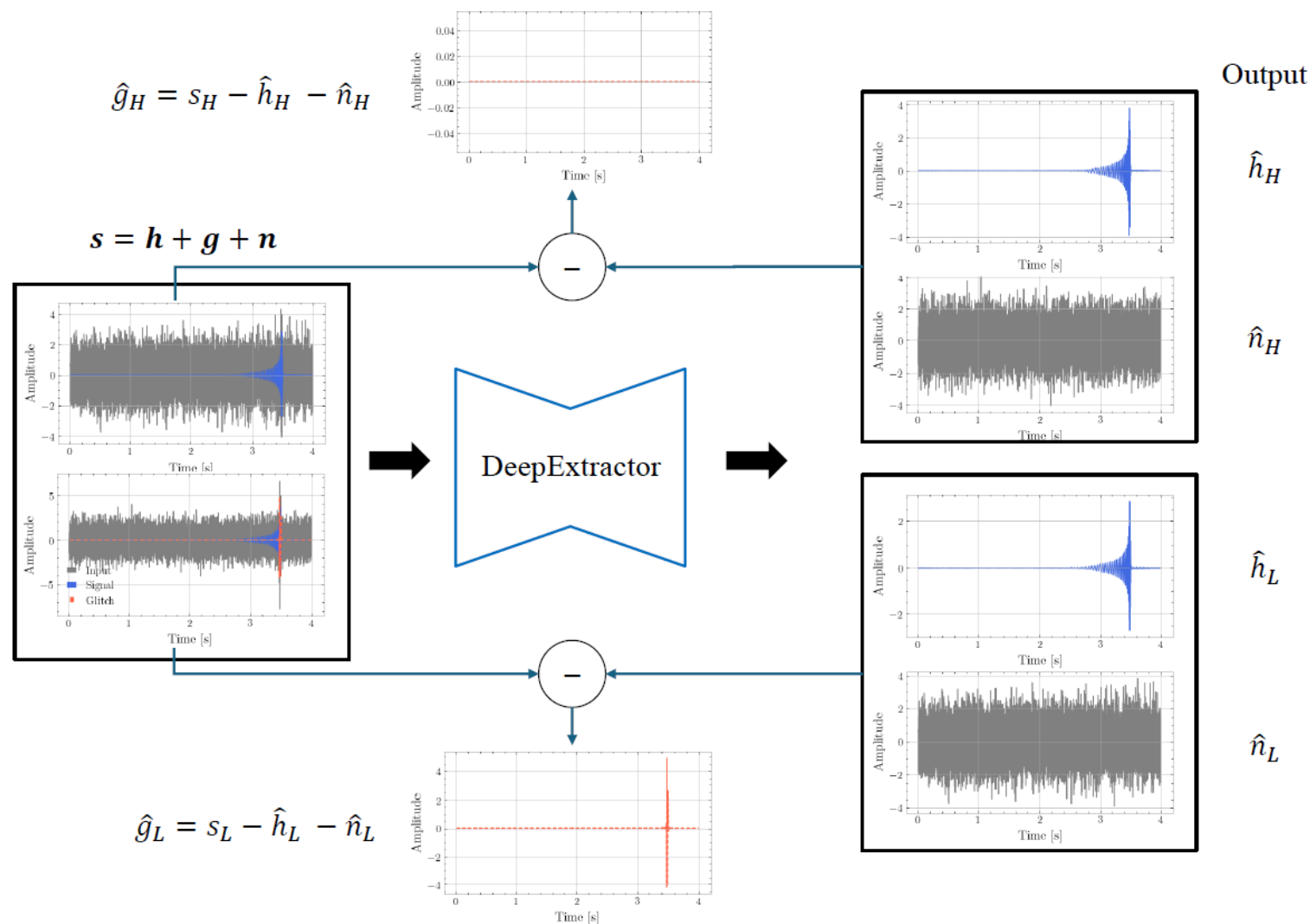


# Clustering Glitch Reconstructions

- Glitch Types
- Blip
  - Fast\_Scattering
  - Koi\_Fish
  - Low\_Frequency\_Burst
  - Scattered\_Light
  - Tomte
  - Whistle

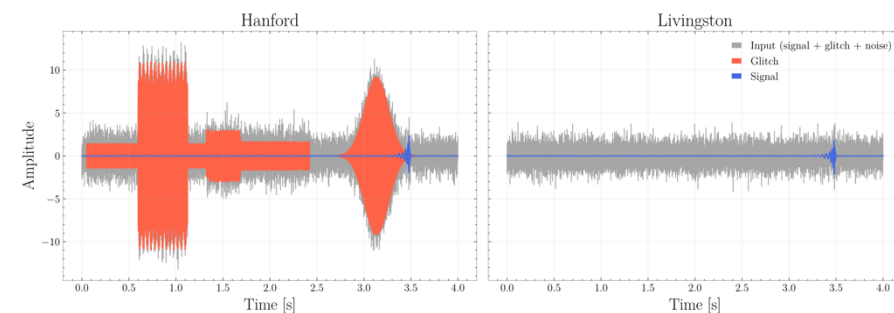


# Signal and glitch separation – Increasing the number of instruments



| 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 $[0, \pi]$                                              |
| Spin azimuthal angles $(\phi_{12}, \phi_{jl})$    | Uniform $[0, 2\pi]$                                             |
| Luminosity distance $D_L$                         | Uniform in comoving volume, $[0, 3000]$ Mpc                     |
| Right ascension RA                                | Uniform $[0, 2\pi]$                                             |
| Declination Dec                                   | Isotropic: $\sin(\text{Dec}) \sim U[-1, 1]$                     |
| Inclination angle $\theta_{\text{JN}}$            | Isotropic: $\cos(\theta_{\text{JN}}) \sim U[-1, 1]$             |
| Polarization angle $\psi$                         | Uniform $[0, \pi]$                                              |
| Phase at coalescence $\phi_c$                     | Uniform $[0, 2\pi]$                                             |
| Geocentric time                                   | Fixed at 3.5 s within 4 s window                                |

Signal parameter space.



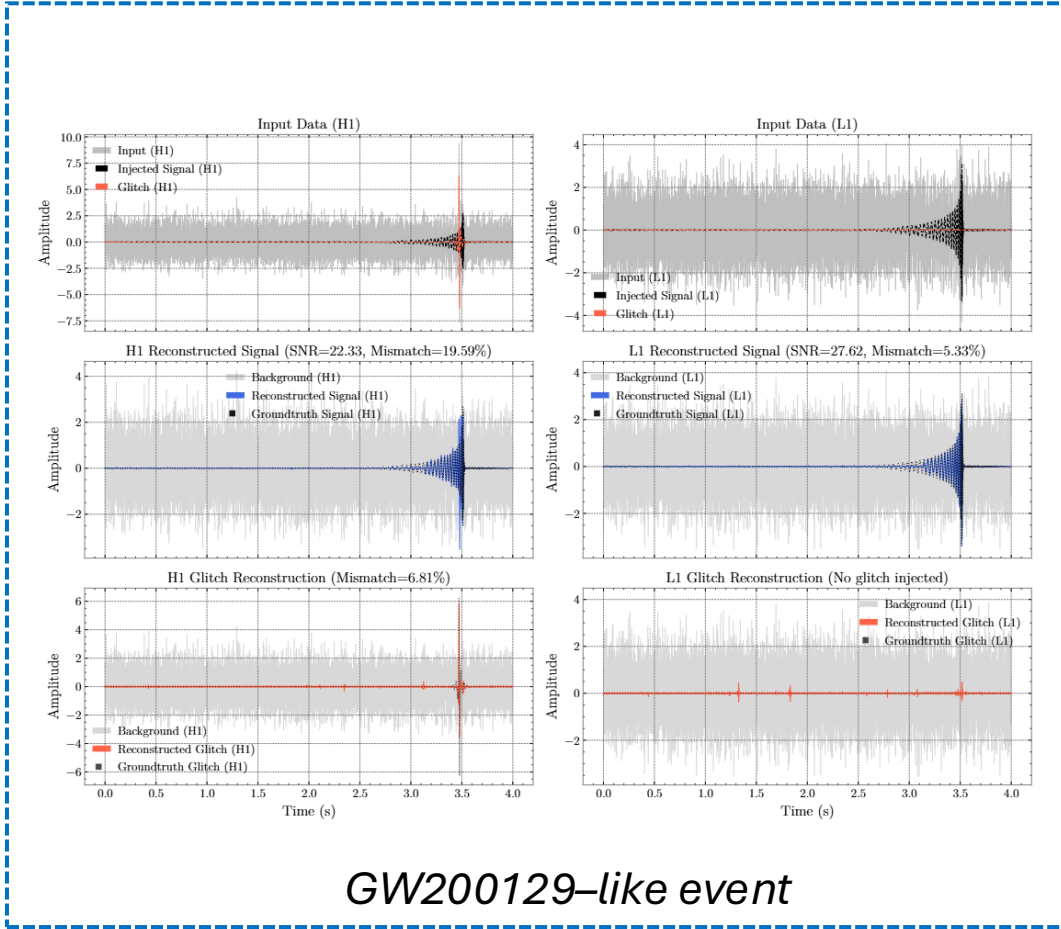
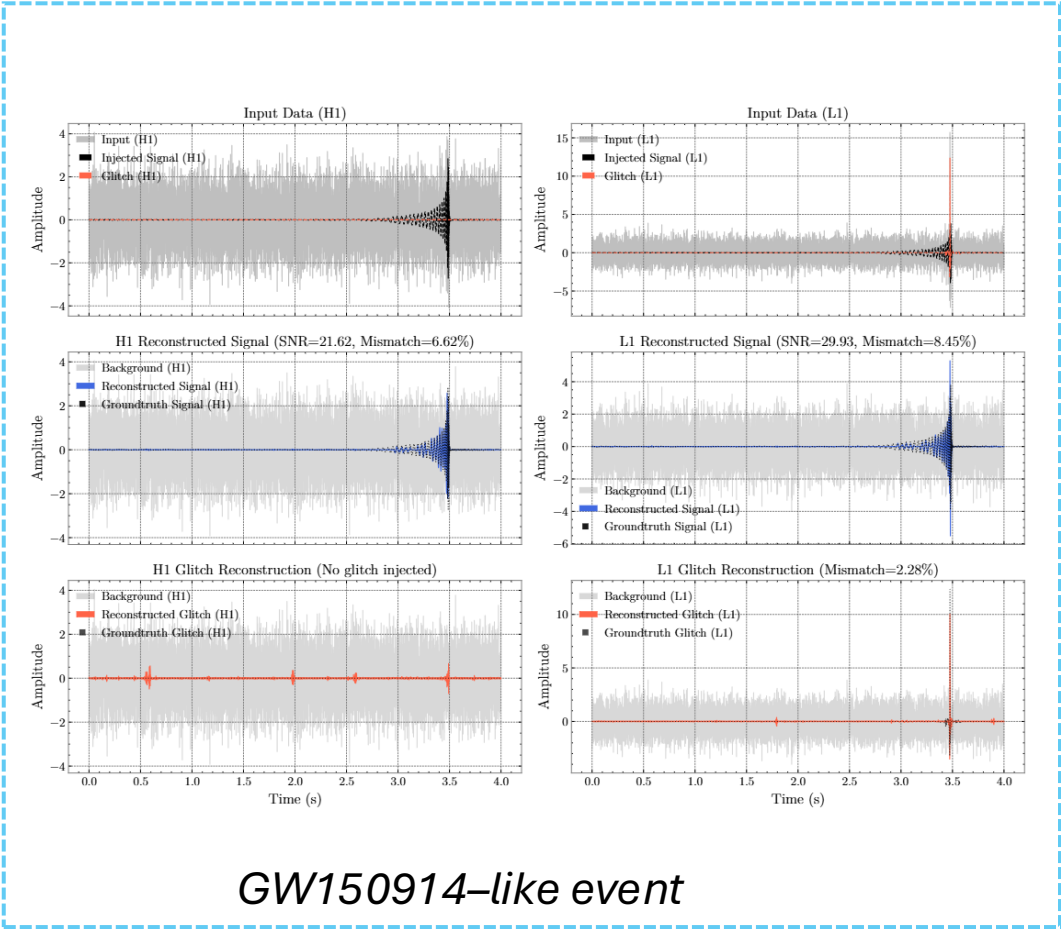
A training sample for DeepExtractor signal+glitch separation. We accumulate 1.2M samples.

# Signal and glitch separation

Input

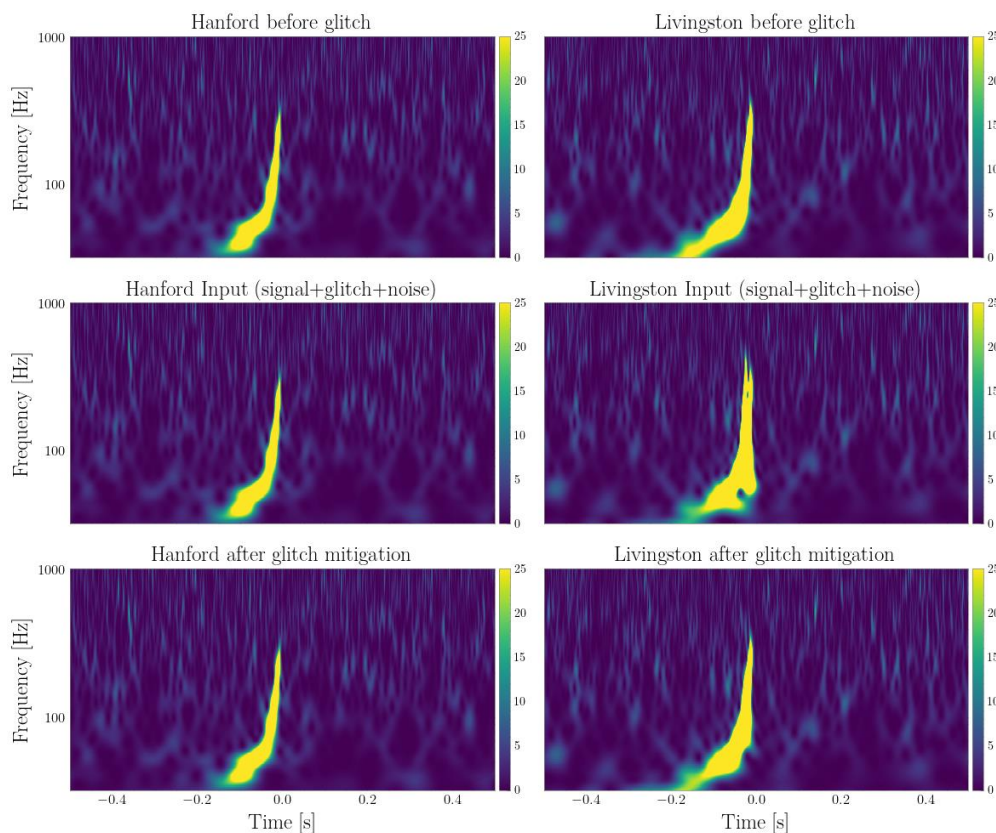
Signal Reconstruction

Glitch Reconstruction

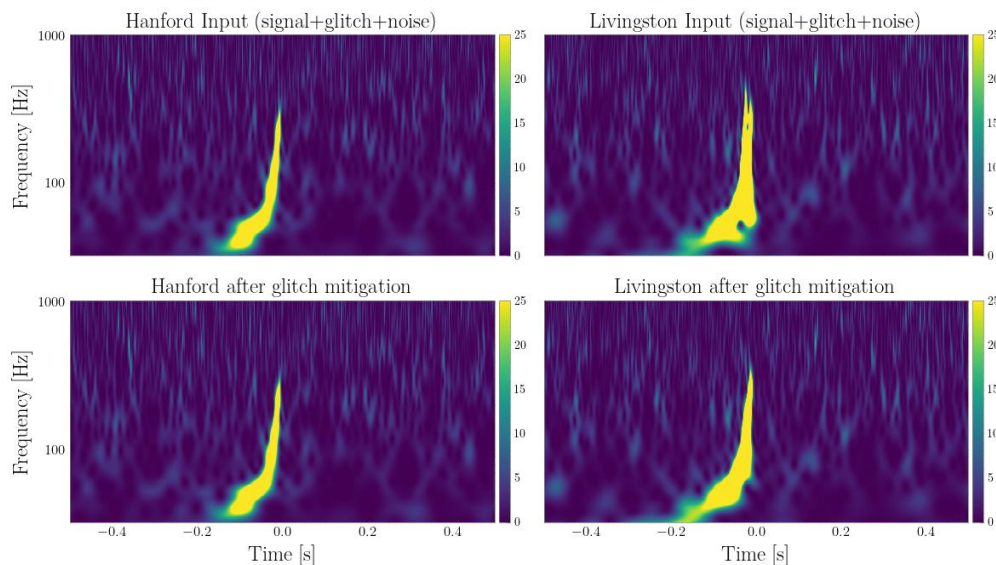


# Signal and glitch separation

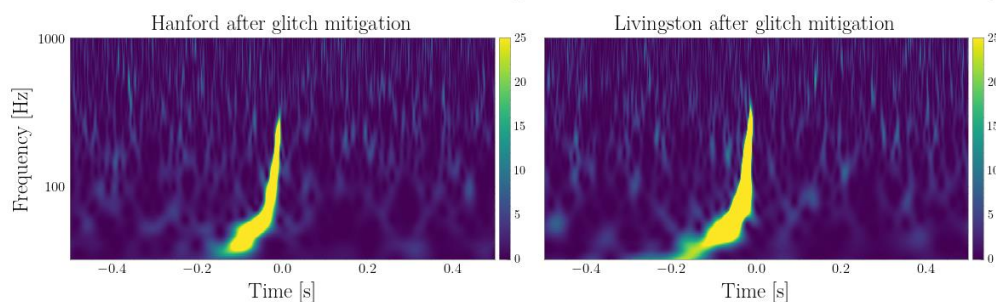
**Before Glitch Injection**



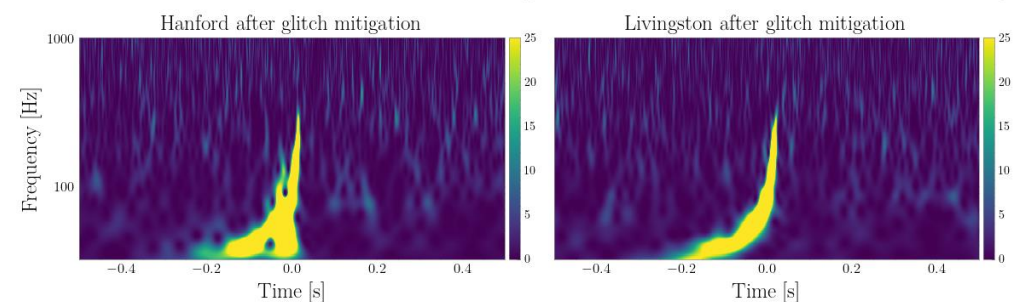
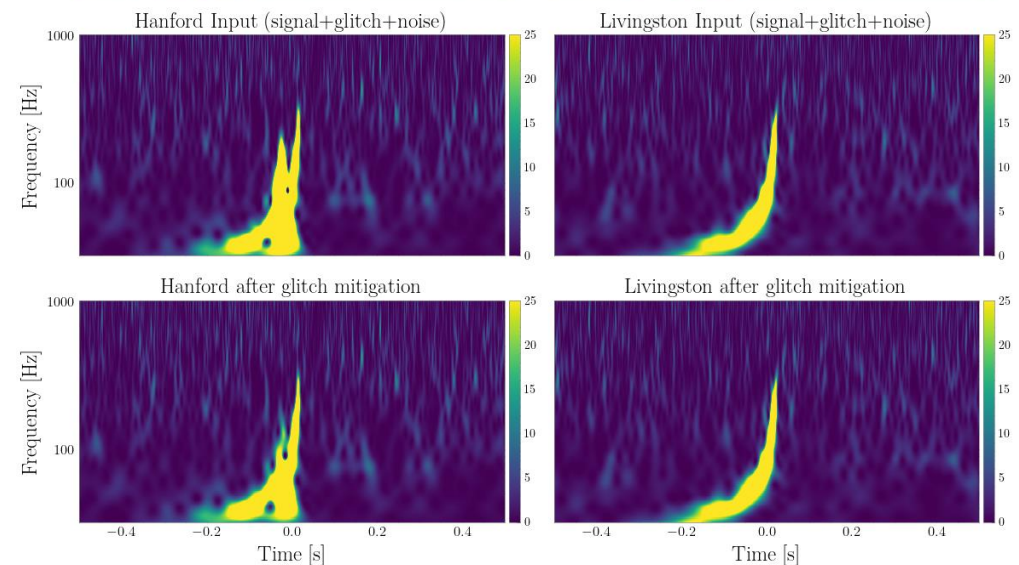
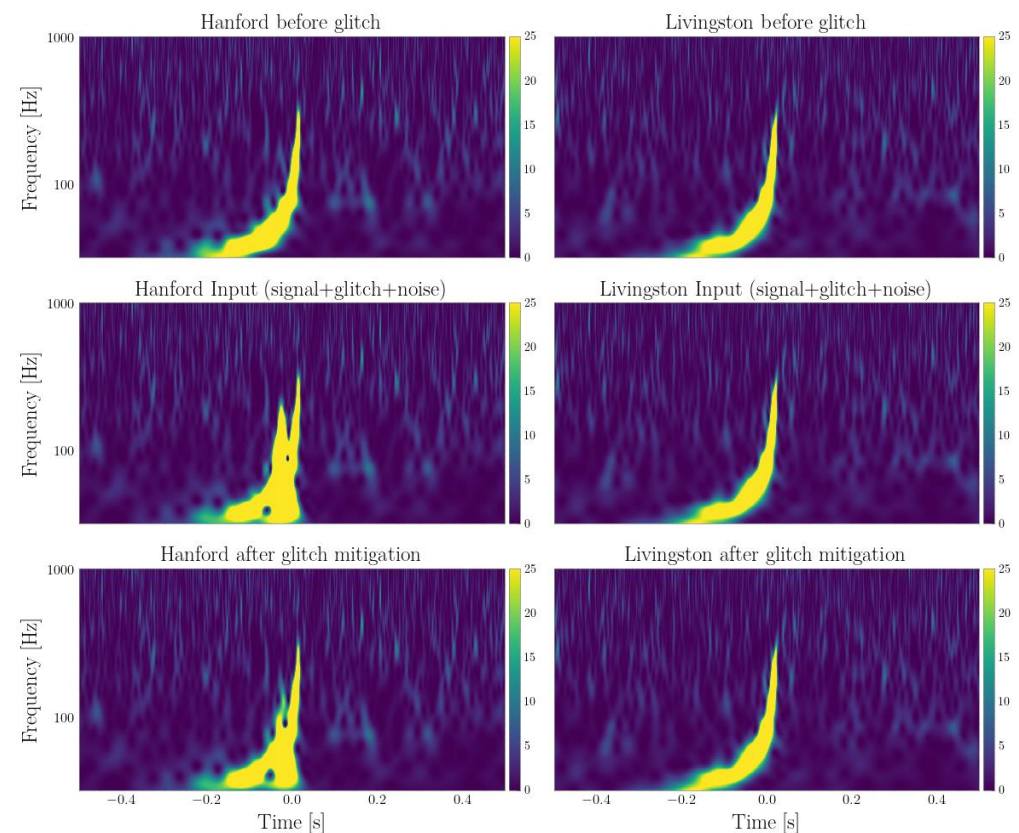
**After Glitch Injection**



**After Glitch Removal**



*GW150914-like event*



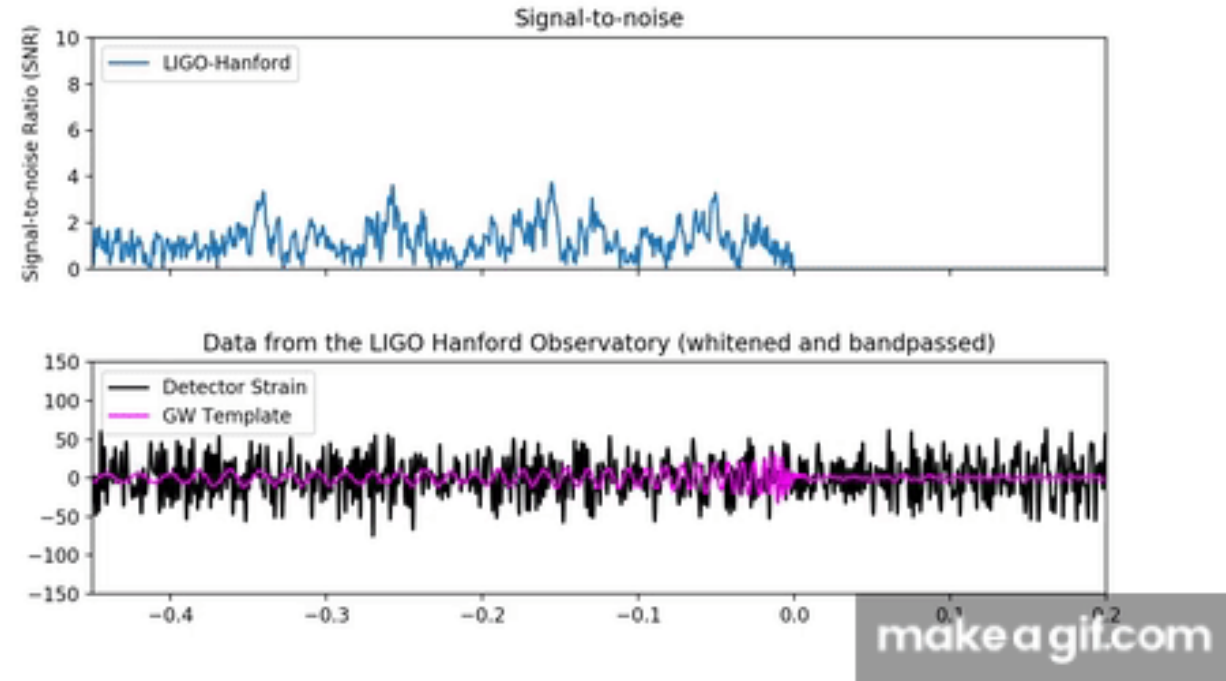
*GW200129-like event*

# DeepExtractor for a template-free search statistic

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

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

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



Matched filtering example.

Image credit: <https://github.com/gwastro/pycbc>

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

**Our statistic** →

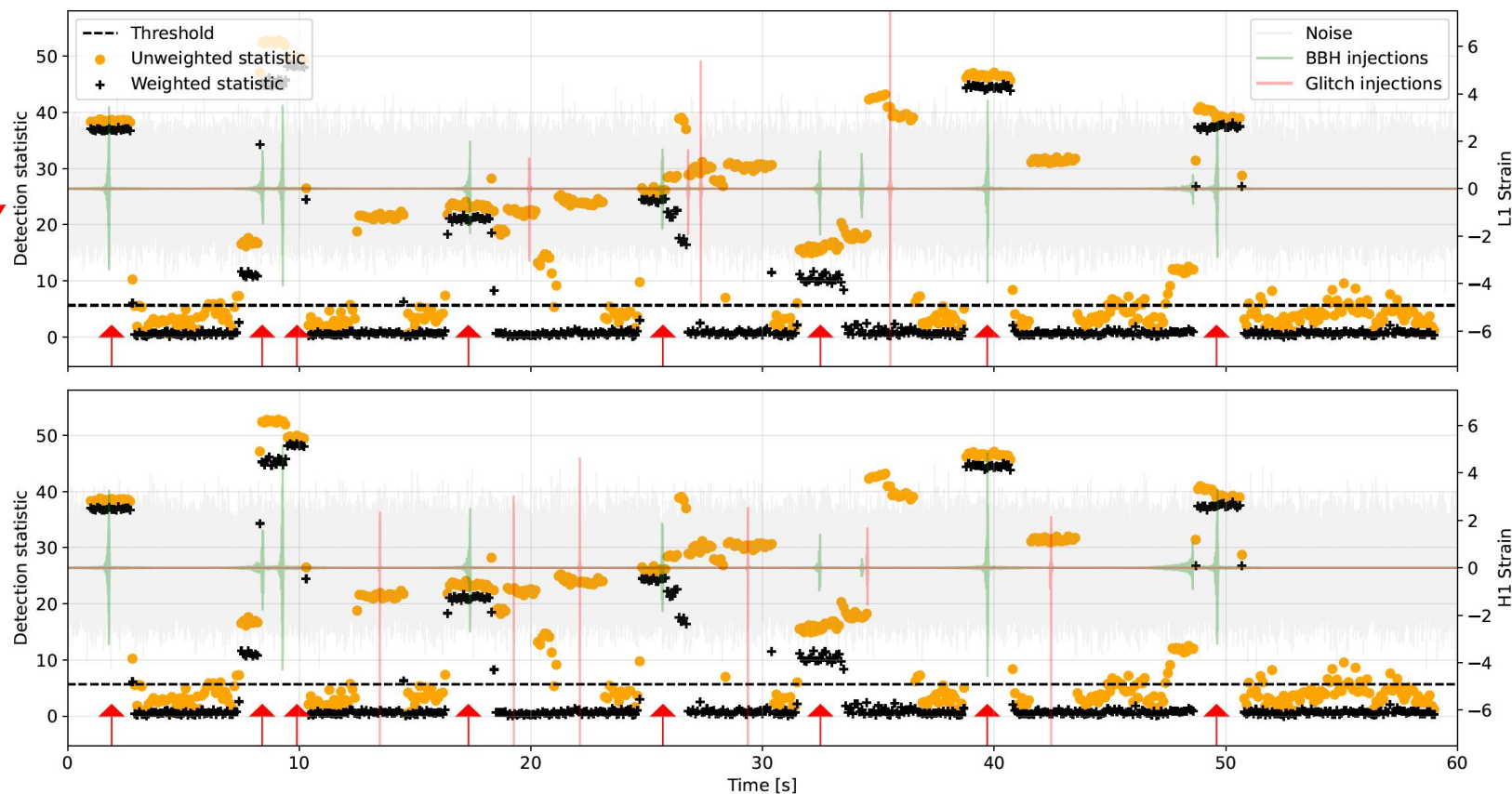
$$\hat{\rho} = \omega_{L,H} \times \hat{\rho}_{network}.$$

# Simulated Example of the Search

LIGO Hanford

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

LIGO Livingston

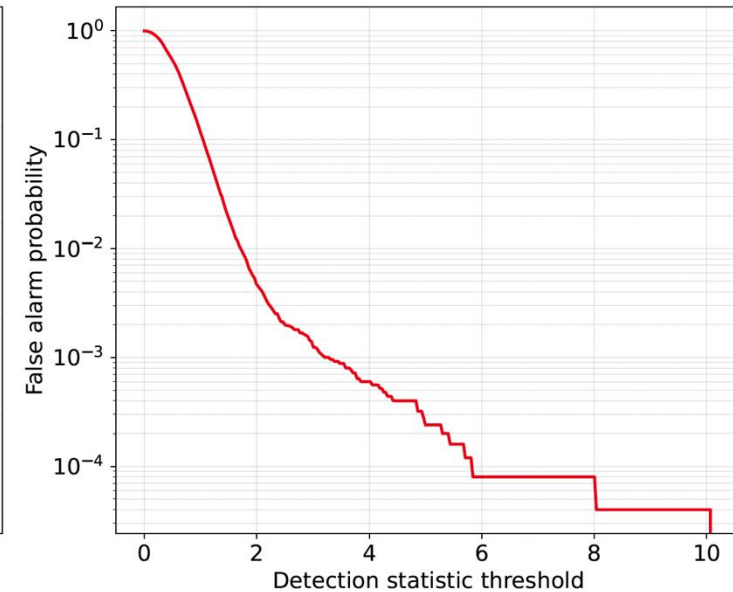
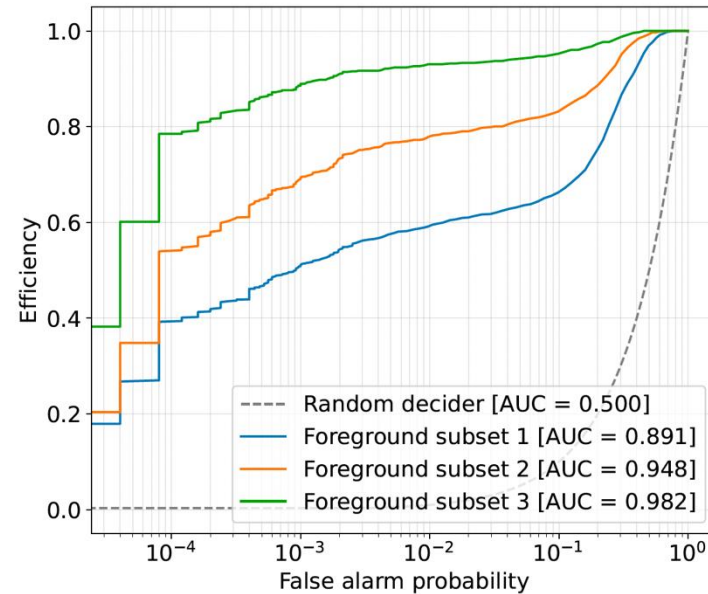
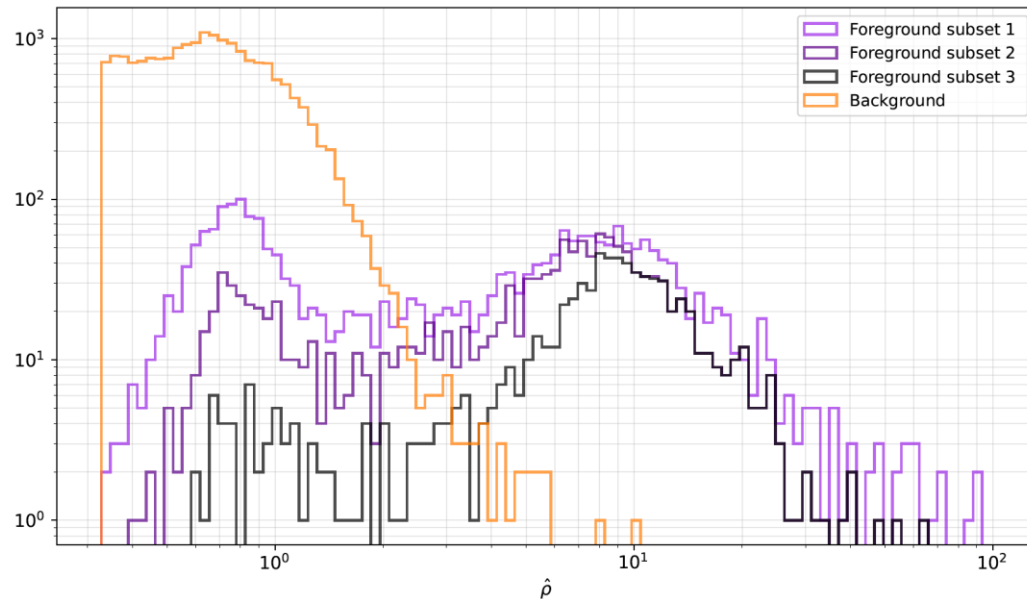


Simulated example of 60s of data.

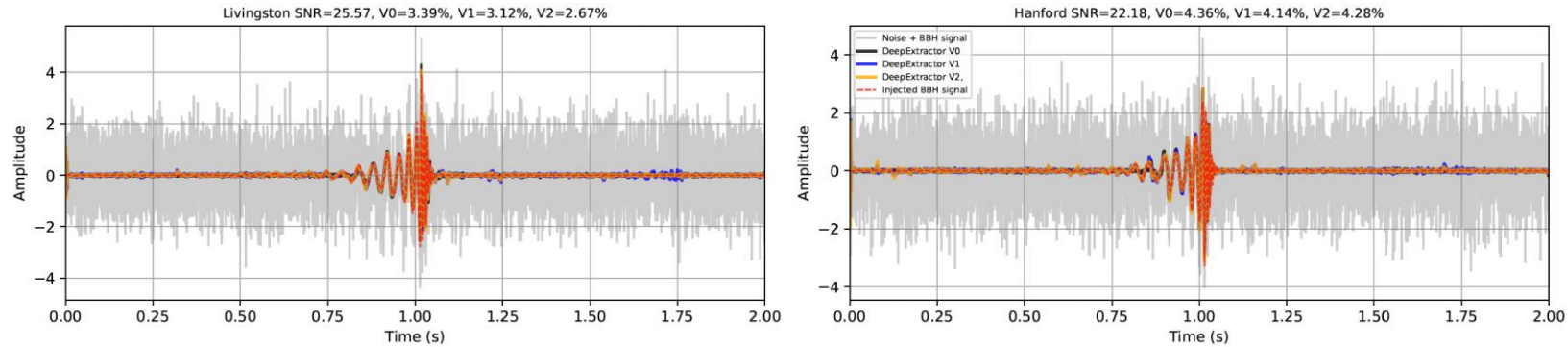
# Simulated Example of the Search

Three population subsets:

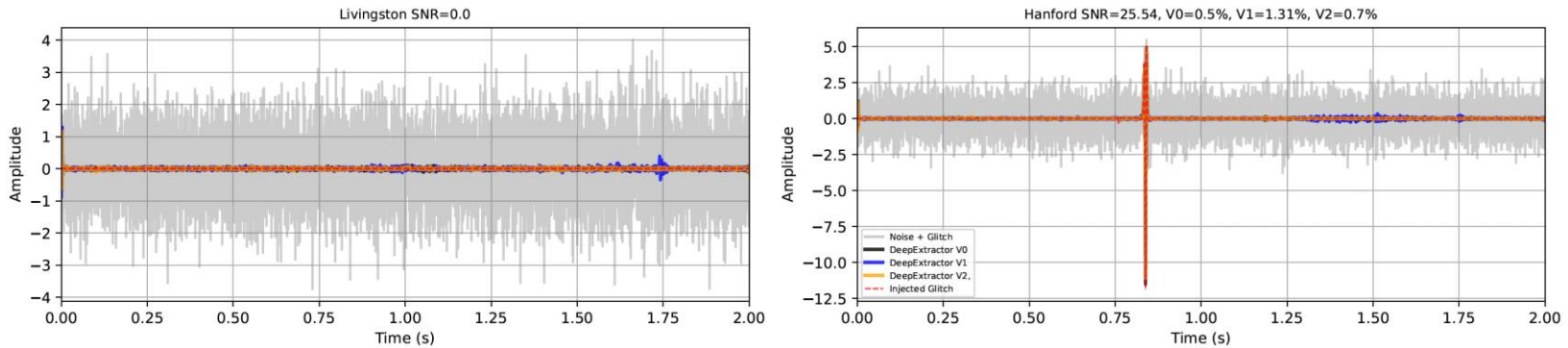
|          | Redshift | Signal length | Matched filter SNR | Number of samples |
|----------|----------|---------------|--------------------|-------------------|
| Subset 1 | $z < 2$  | $t < 10$      | No limitation      | $N = 2502$        |
| Subset 2 | $z < 2$  | $t < 2$       | No limitation    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



# Signals and Glitches in the Search

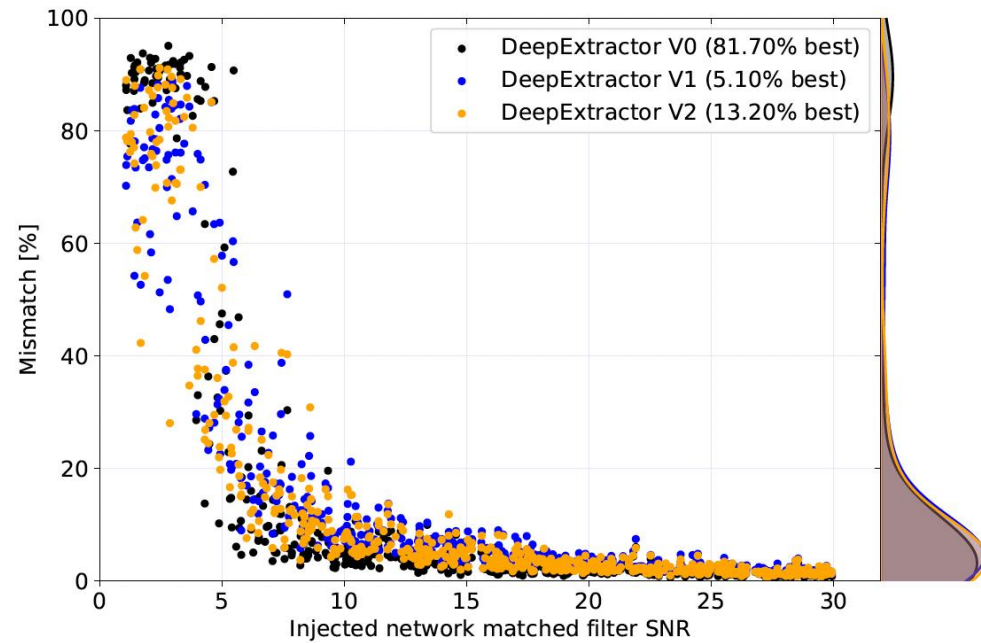


DeepExtractor signal reconstruction.

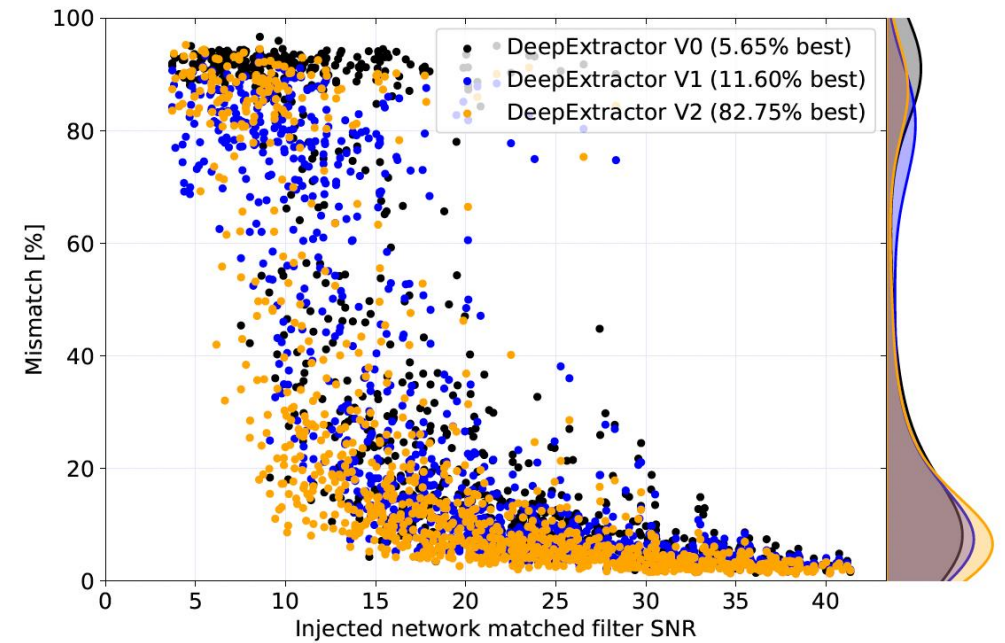


DeepExtractor glitch reconstruction.

# Mismatch Performance vs SNR

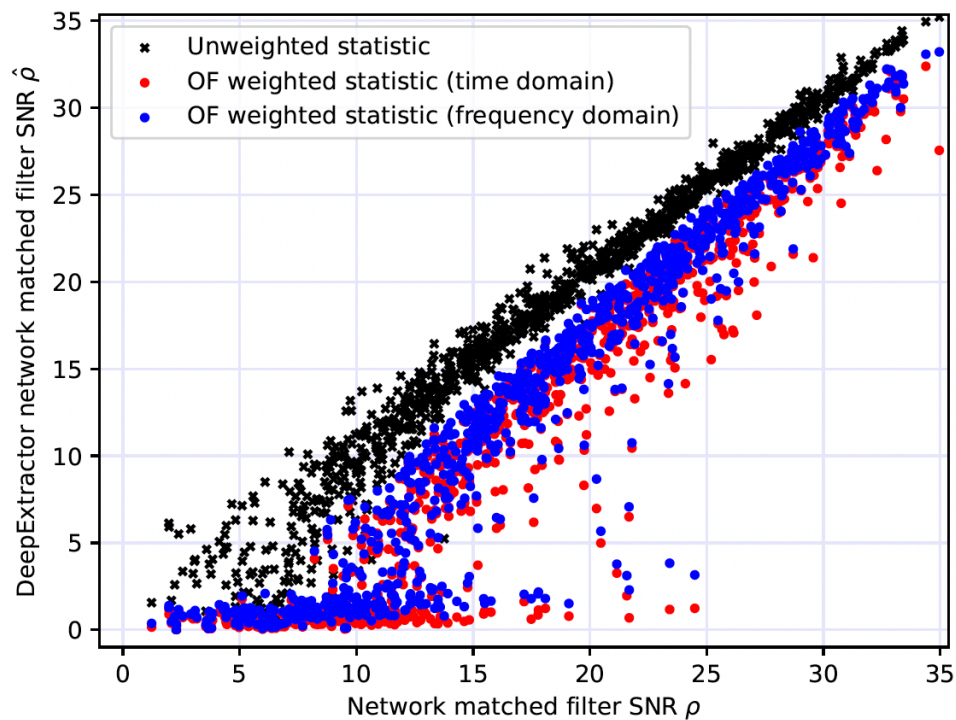


Signals

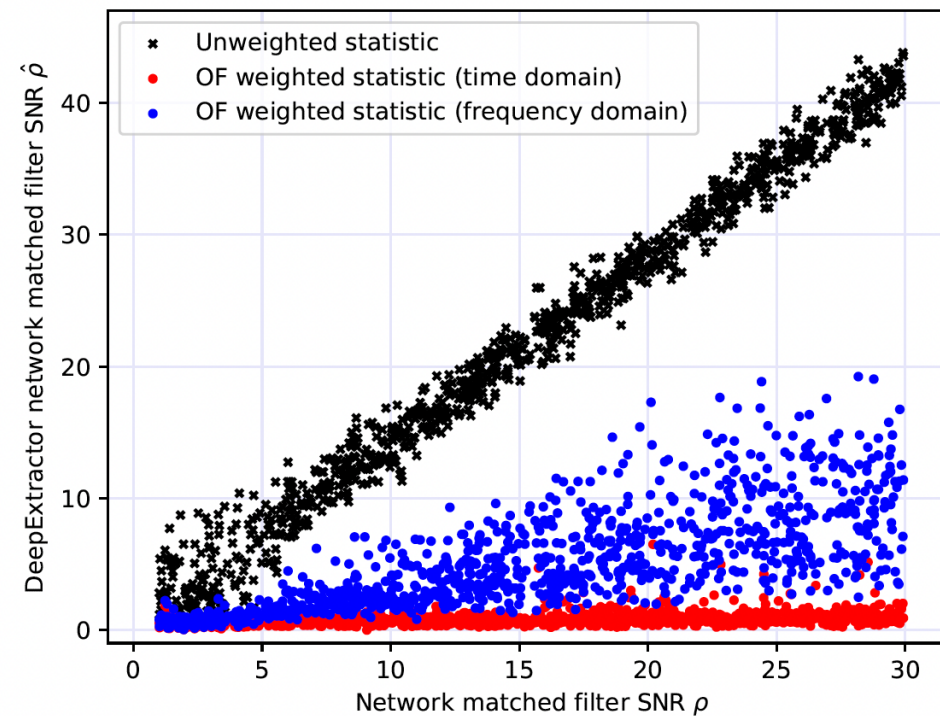


Glitches

# Weighted Detection Statistic

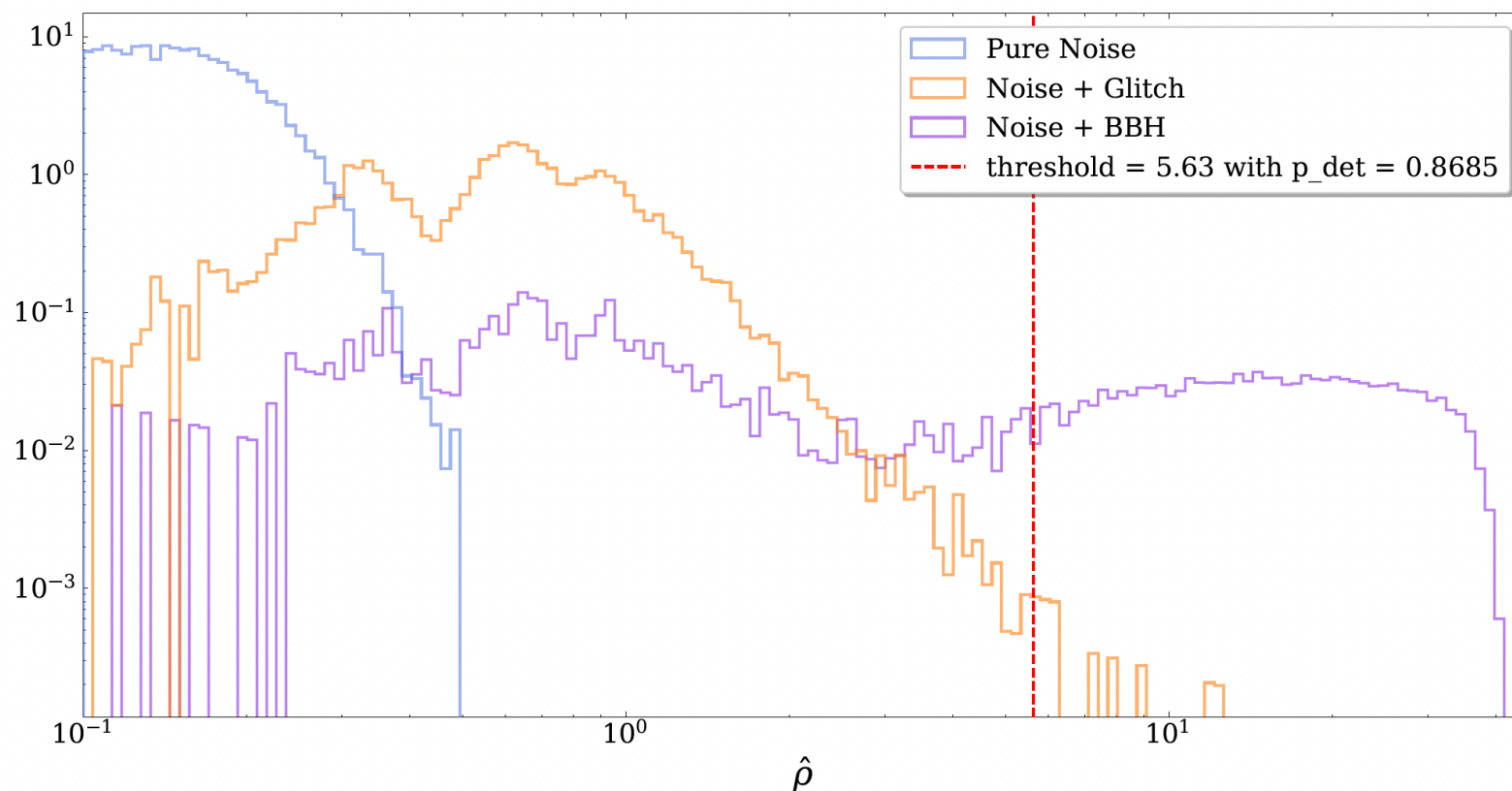


Signals



Glitches

# Weighted Detection Statistic for Simulated Example



| Parameter                 | Distribution                                          |
|---------------------------|-------------------------------------------------------|
| Mass $m_1$                | $\mathcal{U}(5, 100) M_{\odot}$                       |
| Mass $m_2$                | $\mathcal{U}(5, 100) M_{\odot}$                       |
| Spin $a_1$                | $\mathcal{U}(0, 0.99)$                                |
| Spin $a_2$                | $\mathcal{U}(0, 0.99)$                                |
| Luminosity distance $D_l$ | $\mathcal{U}(1, 1000) \text{ Mpc}$                    |
| Coalescence time $t_c$    | V1: 1.75 s<br>V2: $\mathcal{U}(0.25, 1.75) \text{ s}$ |

Signal parameters.

Statistic for 10,000 samples of each class i.e. background, blip glitches ( $\text{SNR} \in [1, 30]$ ), signal.