Gravitational Wave Searches for Compact Binary Mergers

En-Tzu Lin

2023 Gravitational Wave Data Workshop @ National Tsing Hua University

國立清華大學
NATIONAL TSING HUA UNIVERSITY

Adapted from lectures provided by Gravitational Wave Open Science Center

Focus Questions

- Introduction
- What are we looking for?
- What does the data look like?
- How do we find the signal?
- Implementation/ Current detection tools in use by LVK


GRAVITATIONAL WAVE SPECTRUM
esa
Observatories & experiments
Ground-based experiment
Space-based observatory
Pulsar timing array
Cosmic microwave background polarisation
Timescales
milliseconds
seconds
hours
years
billions of years
Frequency (Hz)
100
1
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-16}
Cosmic fluctuations in the early Universe
Cosmic sources
Supernova
Pulsar
Compact object falling onto a supermassive black hole
Merging supermassive black holes
Merging neutron stars in other galaxies
Merging stellar-mass black holes in other galaxies
Merging white dwarfs in our Galaxy
#lisa
ESA | NASA
lisa

GRAVITATIONAL WAVE SPECTRUM

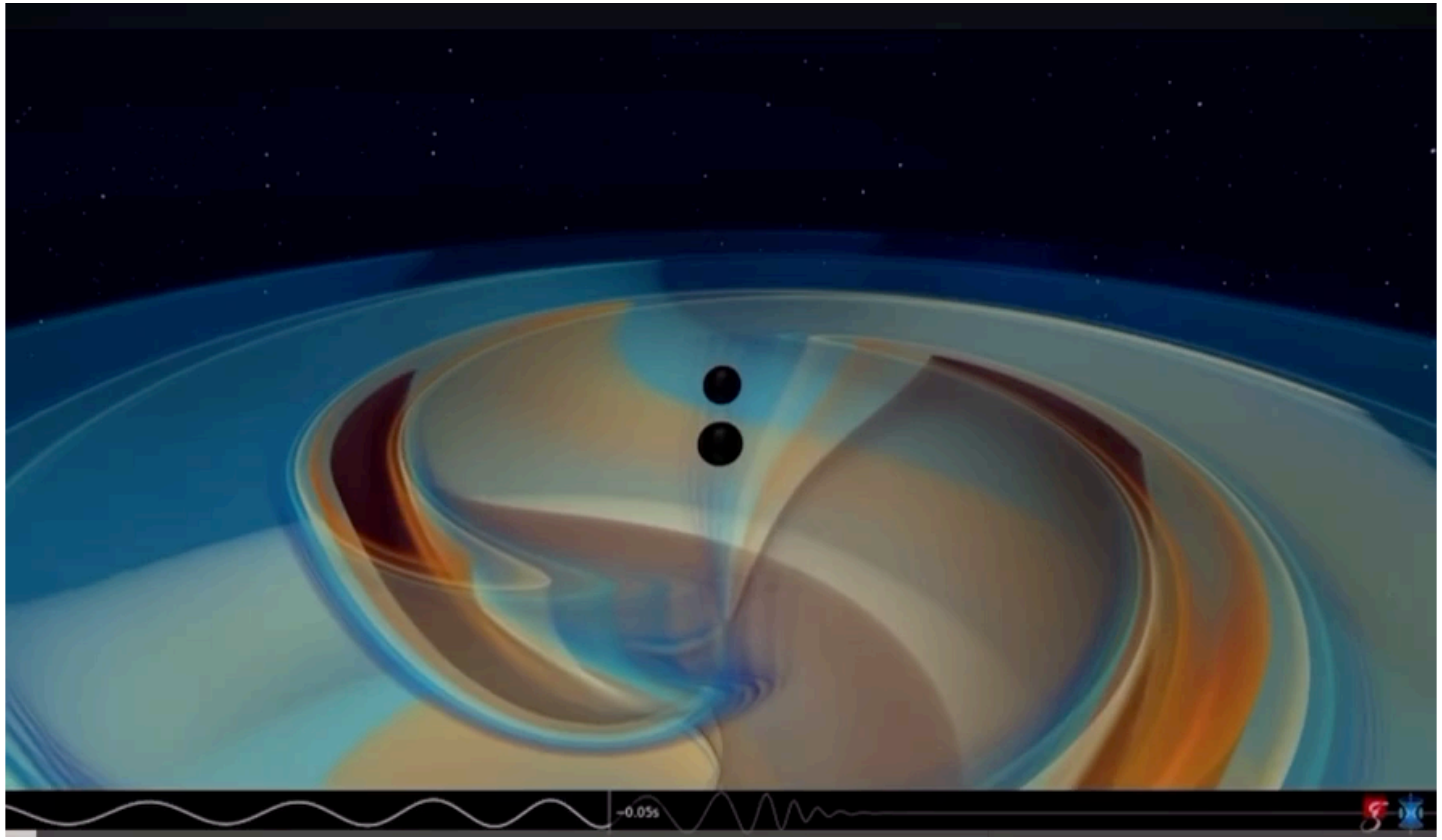
- Gravitational waves are predictions from general relativity
  - Ripples in space-time
  - Any time-varying non-axisymmetric mass distribution can produce gravitational waves
  - Compact binary coalescence (CBCs), Supernova explosion, Pulsars … etc.
- Current ground-based detectors observe gravitational waves at ~10 - a few 1000 Hz

Focus Questions

- Introduction
- **What are we looking for?**
- What does the data look like?
- How do we find the signal?
- Implementation/ Current detection tools in use by LVK

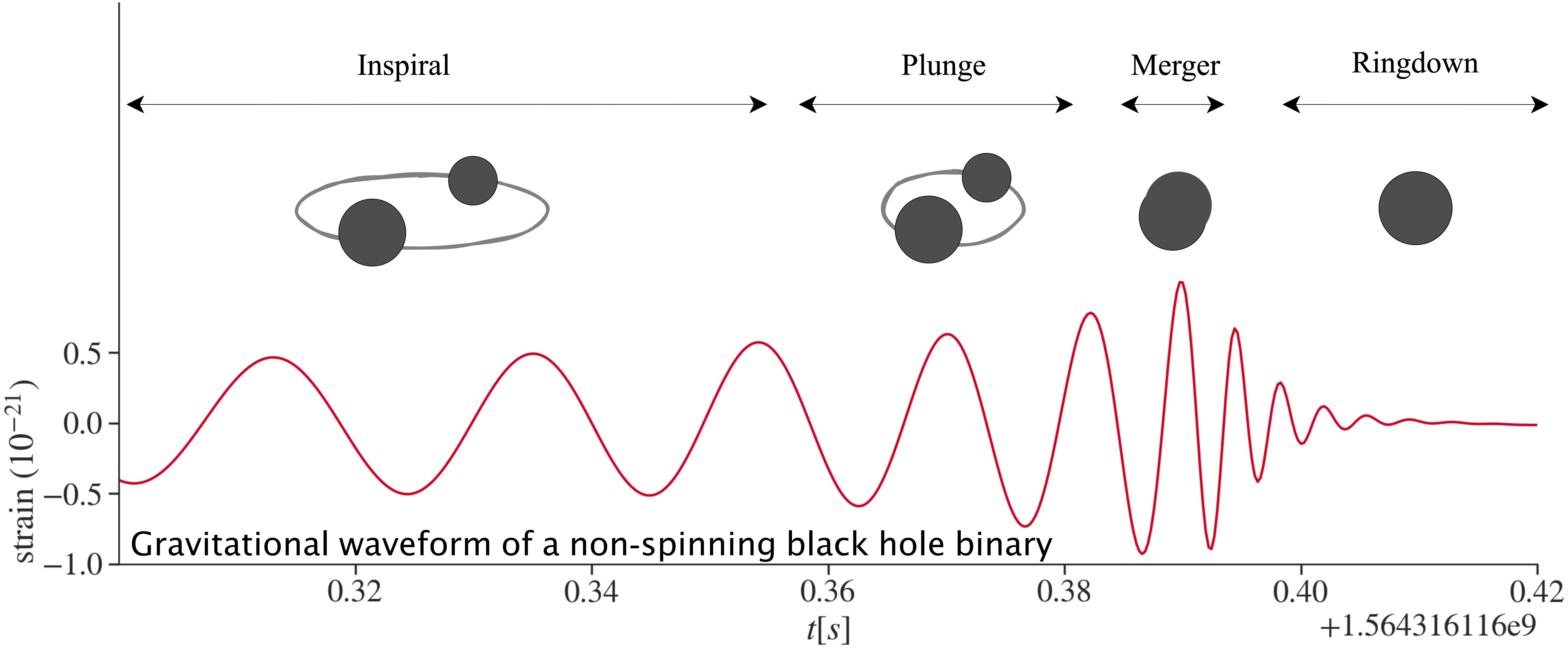
Compact Binary Coalescence

- A pair of compact objects (white dwarfs, neutron stars, black holes)
    - So far we detect binary neutron star (BNS), binary black hole (BBH) and neutron star-black hole (NSBH) mergers
- As objects orbit, they lose energy to GWs
    - The orbit shrinks and speeds up, releasing more energy to GWs
    - Frequency and amplitude increase monotonically
    - Creates a runaway process leading to inspiral and merger



Link to video

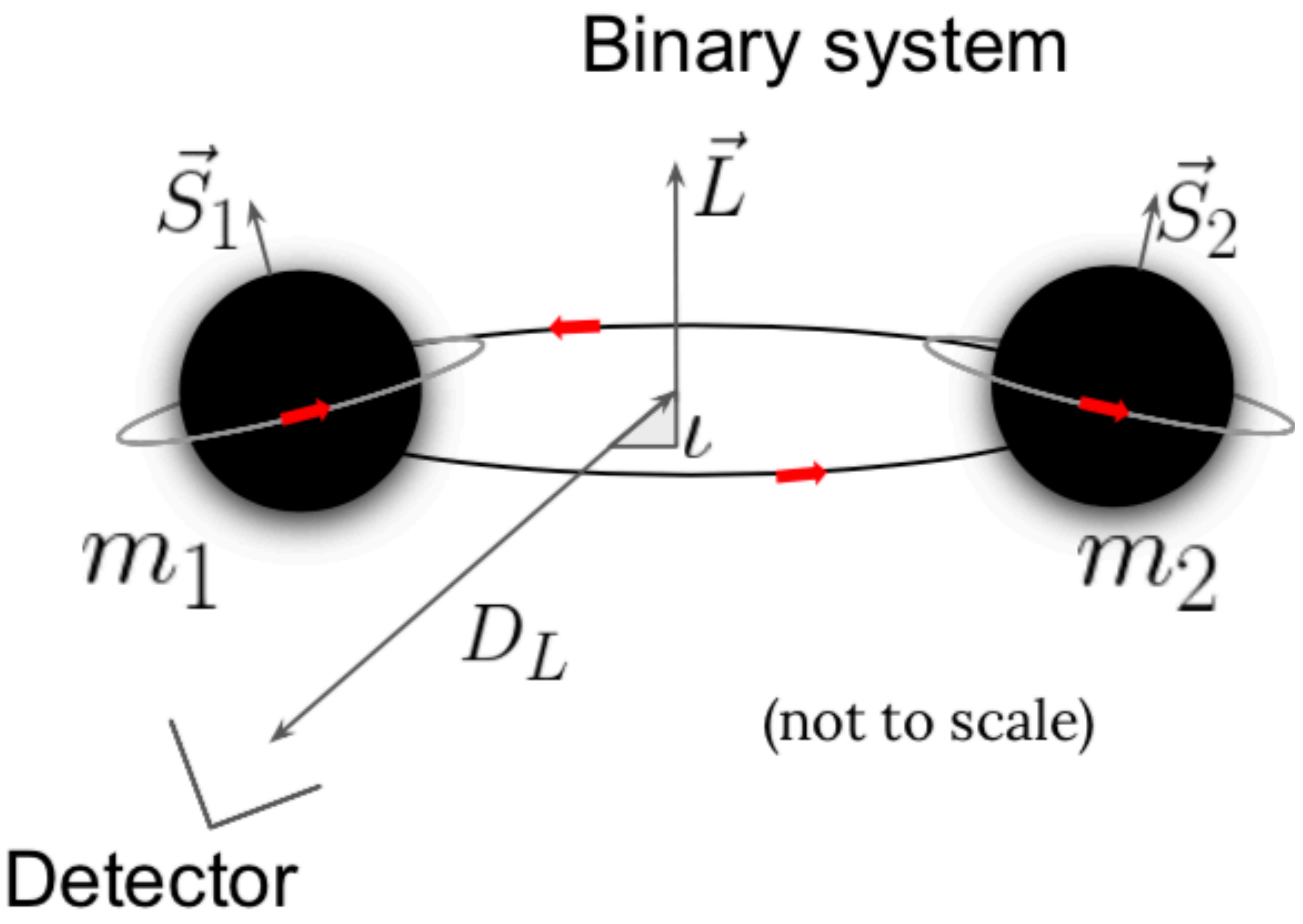
Waveforms model the **inspiral**, **merger**, and **ringdown** of the binary



$$\text{Strain} = s = \frac{\Delta L}{L} \sim 10^{-21} \rightarrow \Delta L \sim 10^{-18}\text{m, given } L \sim \mathcal{O}(1\ \text{km})$$

Modelling Compact Binary Mergers

The signal from a binary system made up of black holes is described by 15 parameters



Intrinsic parameters

- Two component masses: m_1, m_2
- Six spin Components: χ_1, χ_2

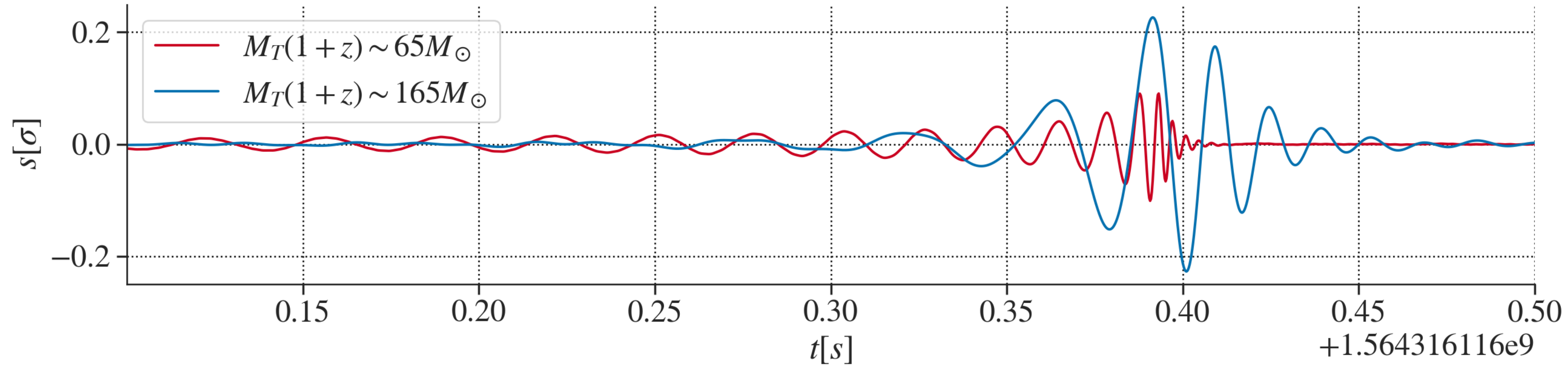
Extrinsic parameters

- Sky Location: (α, δ)
- Luminosity distance: D_L (or equivalently the redshift z)
- Binary orientation parameters: (ι, φ)
- Polarisation angle: ψ
- Merger time: t_c

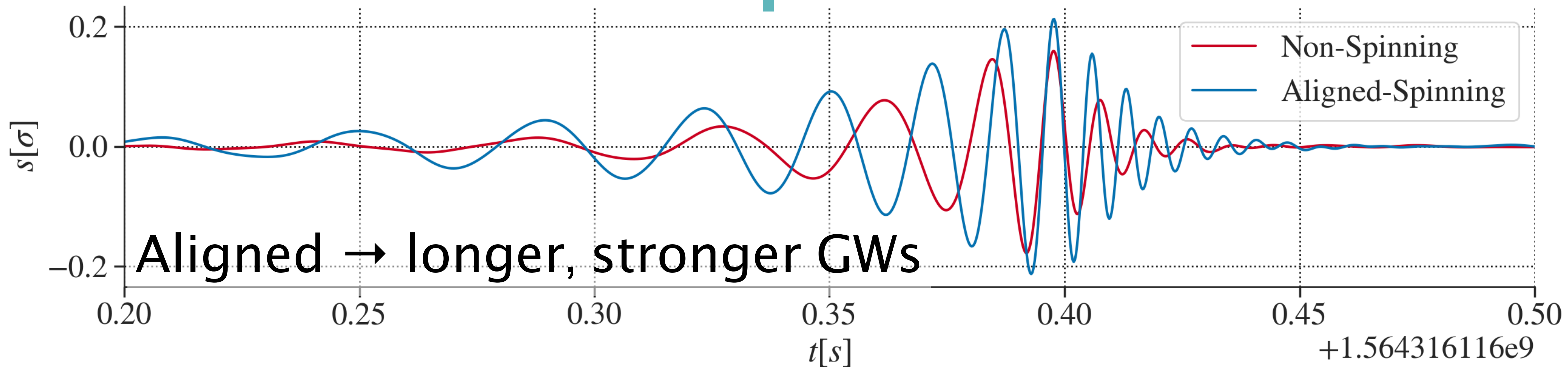
More parameters required if matter or new physics is included

Phenomenology of black hole binaries

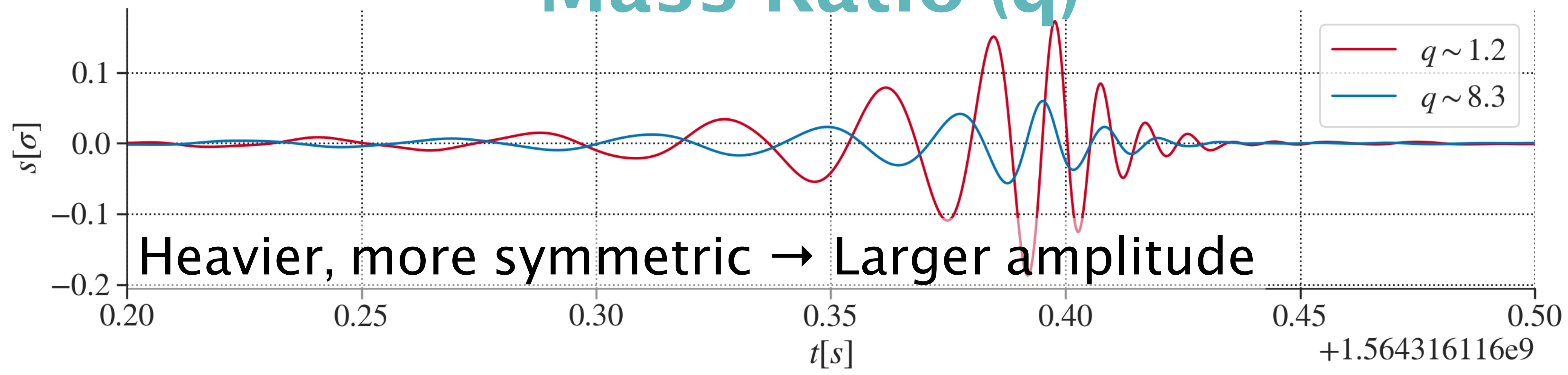
Mass

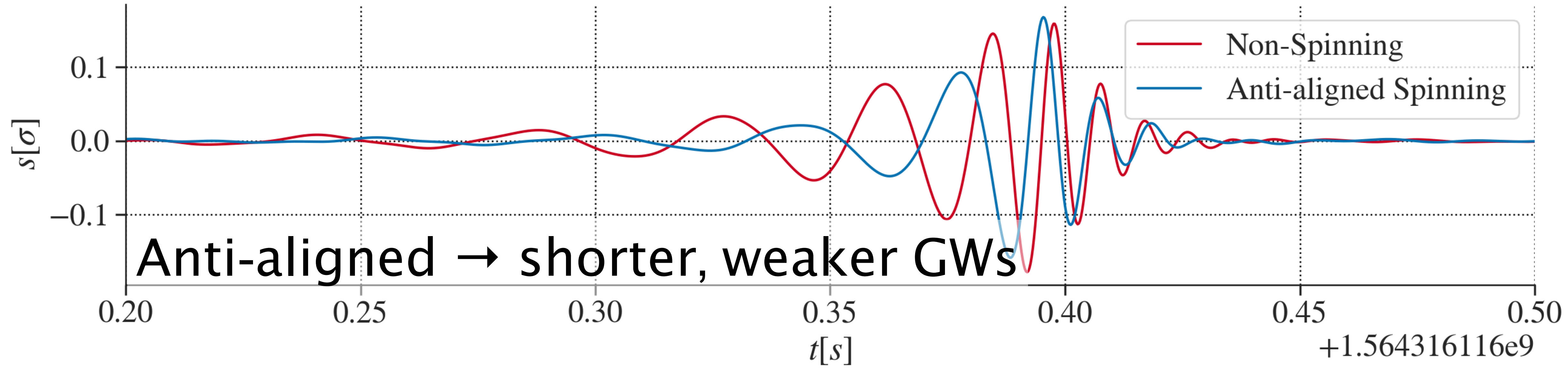


Spins



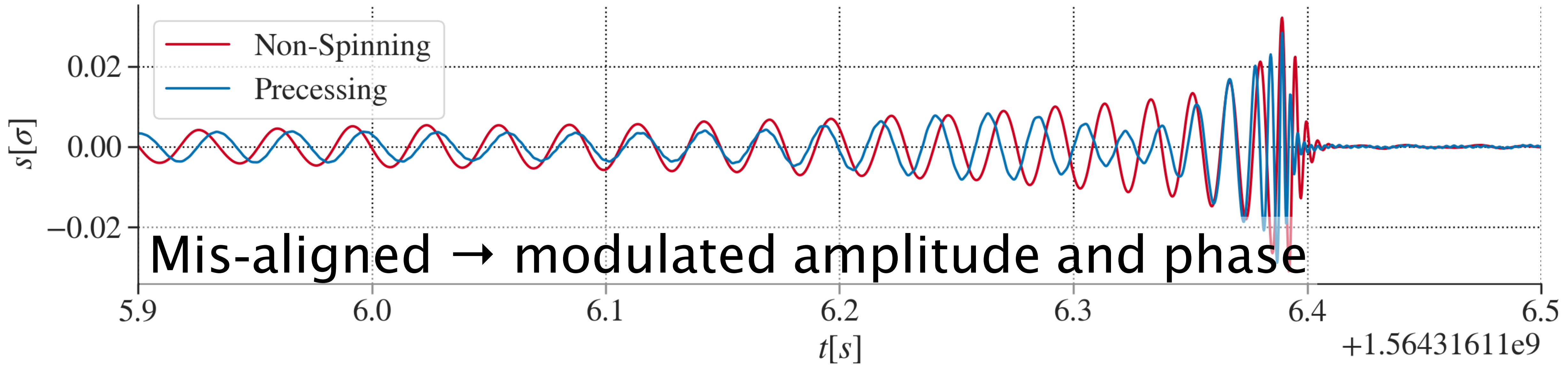
Mass Ratio (q)





$$s(t\mid\boldsymbol{\theta}) \propto \left(M_T(1+z)\right)^{5/6}\sqrt{\frac{q}{1+q^2}} \quad , q=\frac{m_1}{m_2}=\frac{\text{Heavier black hole}}{\text{Lighter black hole}}\geq 1$$

Leading order



Focus Questions

- Introduction
- What are we looking for?
- **What does the data look like?**
- How do we find the signal?
- Implementation/ Current detection tools in use by LVK

Gravitational Wave Detector Data

GW interferometers record data as a **discretely sampled time series**

$d = \{d(t_1), \ldots, d(t_n)\}$ at sampling frequency $f_s = 16\ \mathrm{kHz}$

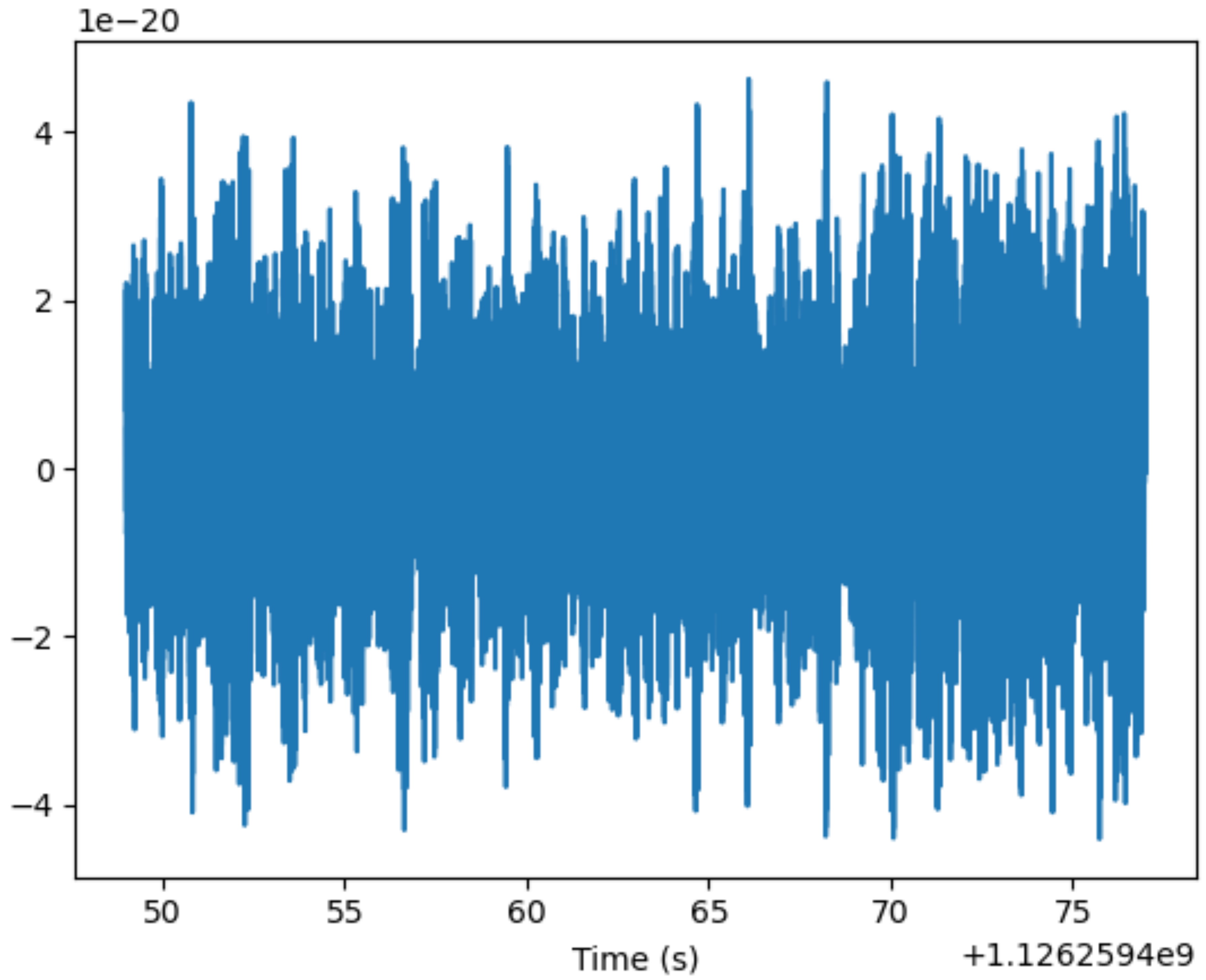
$\rightarrow N_{\mathrm{samples}} = T \times f_s$, where T = data segment duration

$$\boldsymbol{d} = \boldsymbol{s}\left(\boldsymbol{\theta}\right) + \boldsymbol{n}$$

Strain data

Signal
Deterministic for CBC

Noise
Stochastic



Gravitational Wave Detector Data

GW interferometers record data as a **discretely sampled time series**

$d = \{d(t_1), \ldots, d(t_n)\}$ at sampling frequency $f_s = 16$ kHz

$\rightarrow N_{\text{samples}} = T \times f_s$, where T = data segment duration

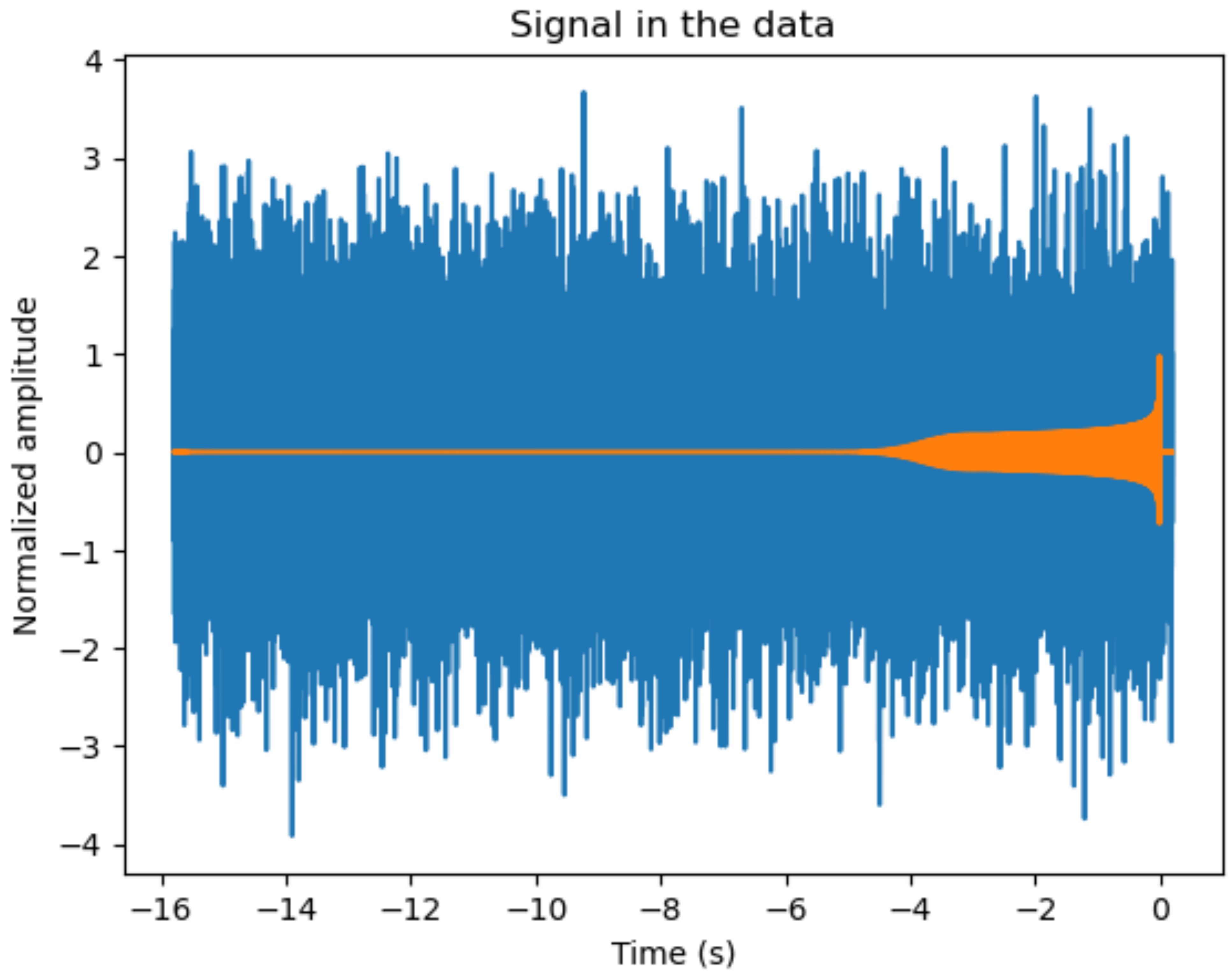
$$\boldsymbol{d} = \boldsymbol{s}\left(\boldsymbol{\theta}\right) + \boldsymbol{n}$$

Strain data

Signal
Deterministic for CBC

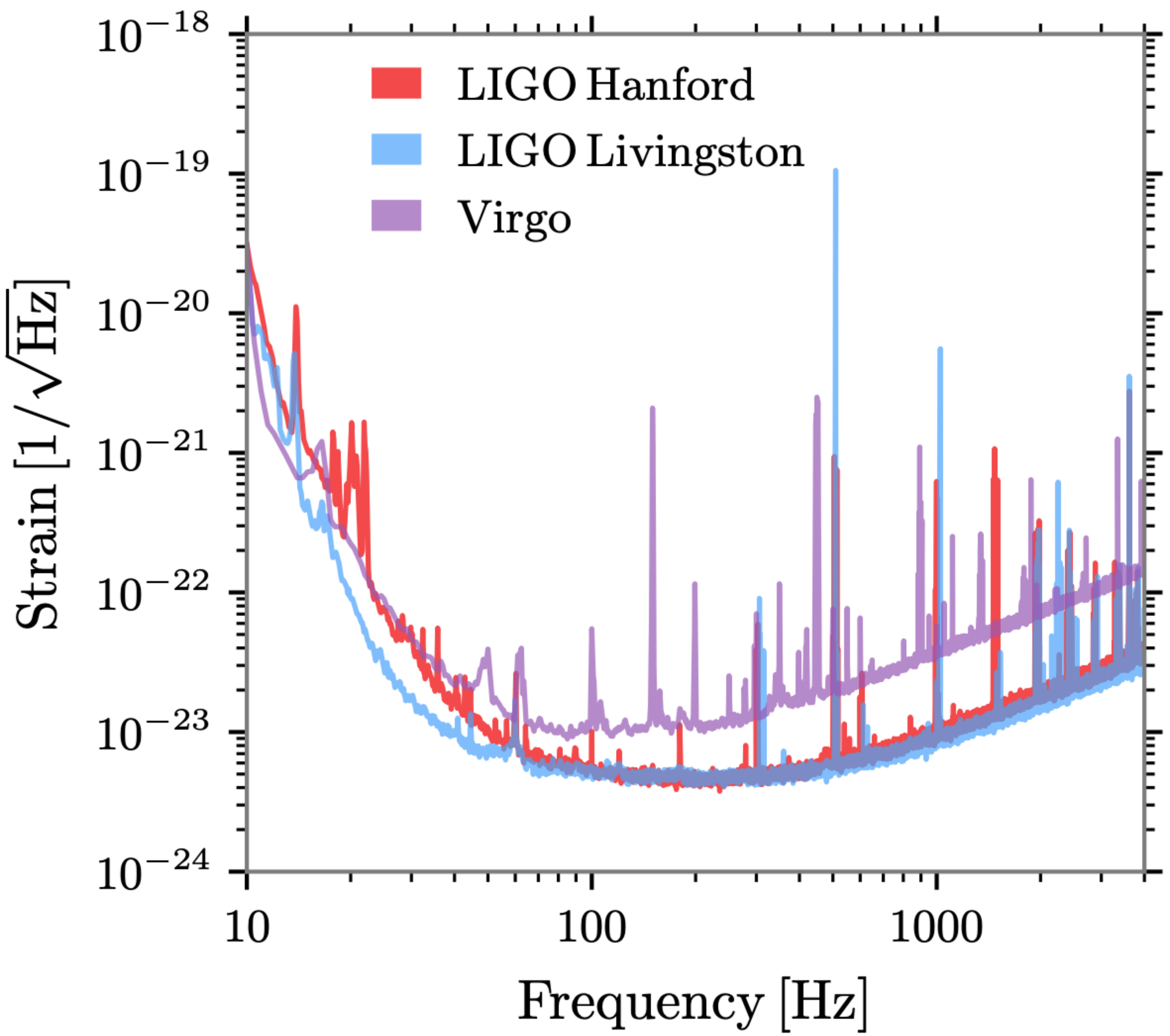
Noise
Stochastic

$|\boldsymbol{n}| \gg \left|\boldsymbol{s}\left(\boldsymbol{\theta}\right)\right| \rightarrow$ Needle in a haystack



Complicated Noise Curve

Detector noice is not **white** – not distributed evenly across frequencies.



LIGO and Virgo PSD for O3.

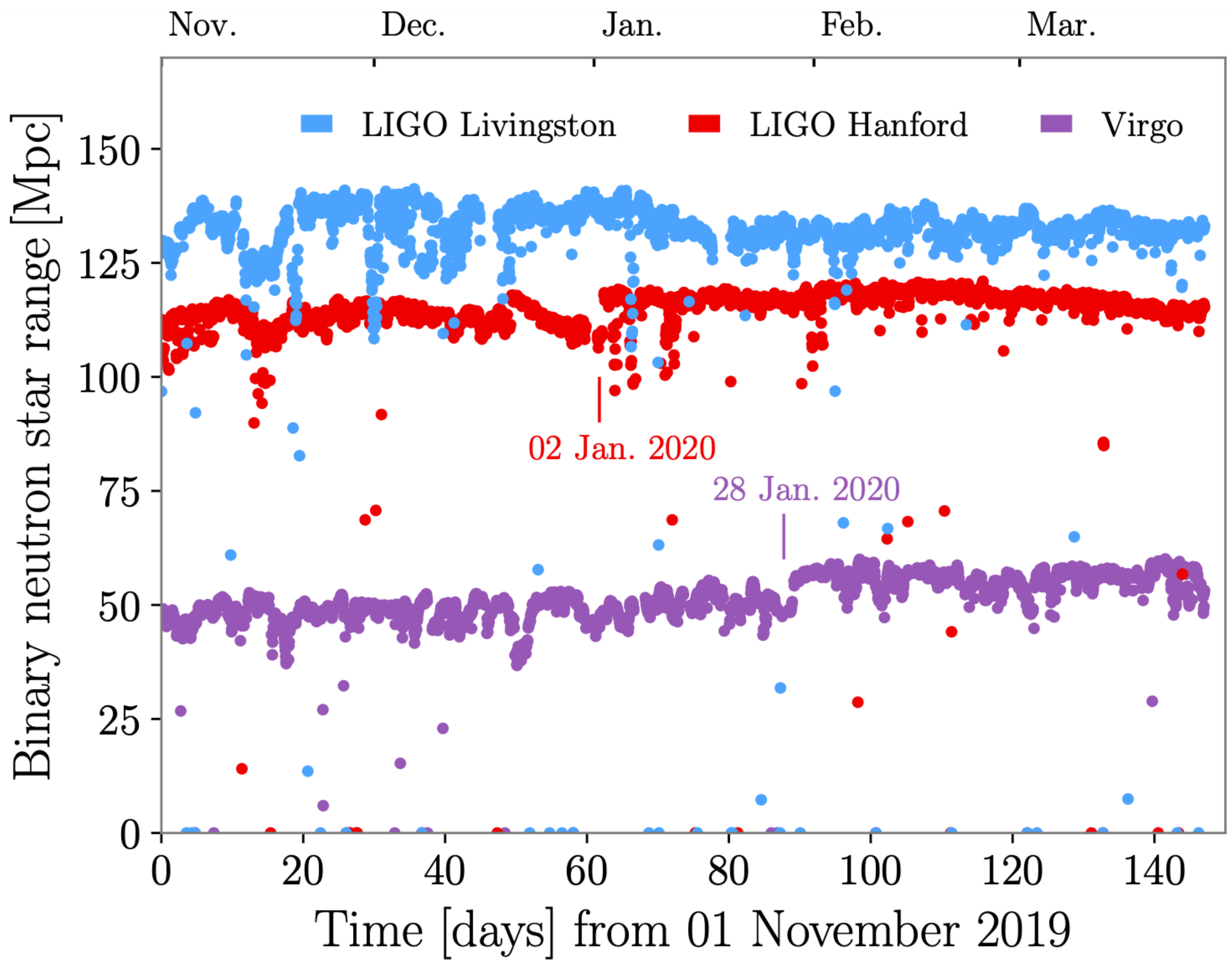
The PSD varies across frequency bins, so the data needs to be whitened before filtering, essentially scaled by the PSD in frequency space.

GWTC-3 paper: arXiv:2111.03606

Non-stationarity

Noise properties can vary over **long timescales**.

We must continuously track the noise properties and update our estimate of the PSD.



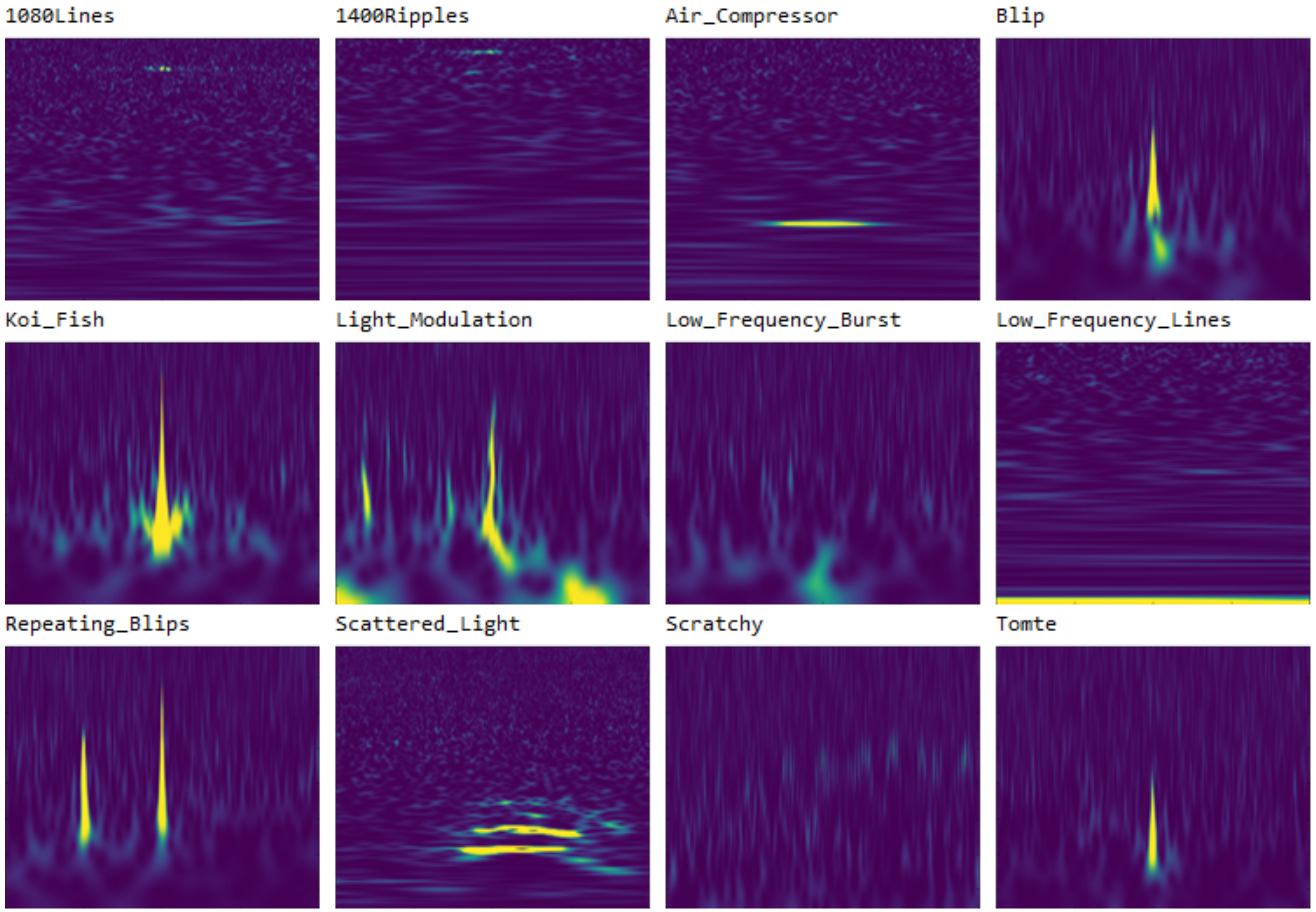
GWTC-3 paper: arXiv:2111.03606

Non-Gaussian

Noise properties can also vary over short timescales.

Short duration non-Gaussian artifacts in the data are **glitches**.

Glitches can be a major problem for matched filtering searches!



Detector Noice Summary

- The noise curves are complex, with many lines
- Sensitivity is highly non-stationary
- Non-Gaussian artifacts regularly appear in the data

The 4km long arms of the LIGO experiment at Hanford. LIGO lab: www.ligo.caltech.edu

Focus Questions

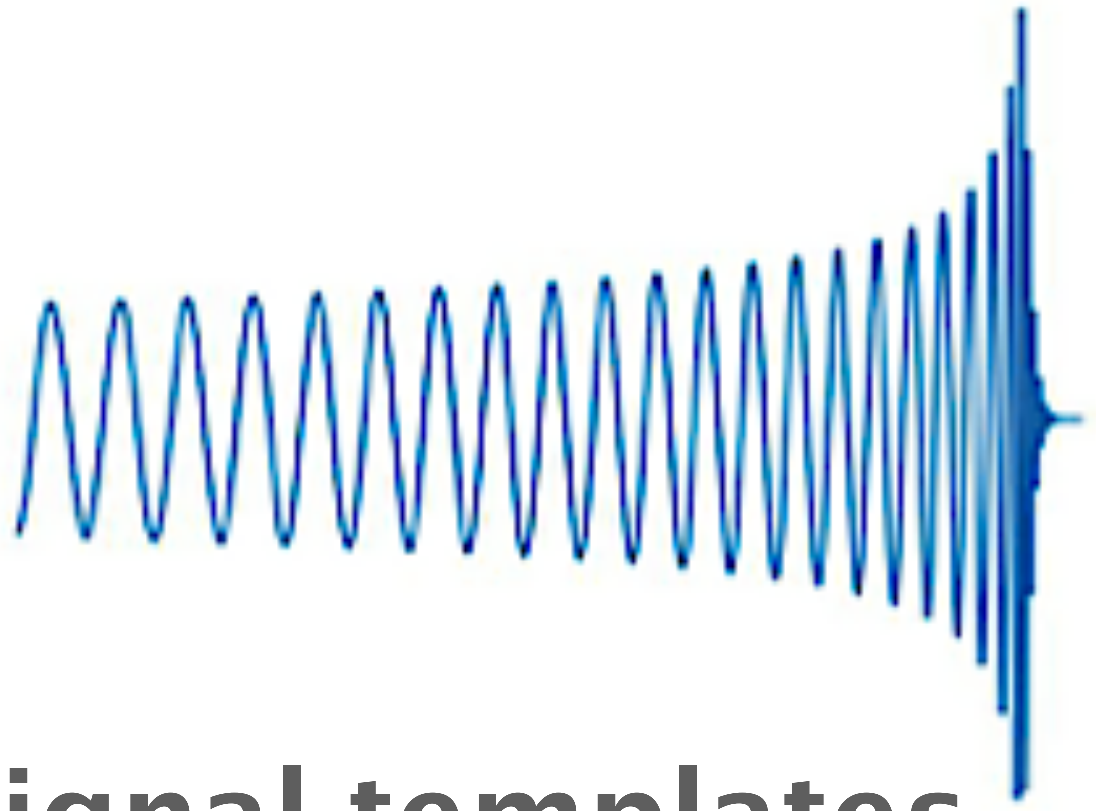
- Introduction
- What are we looking for?
- What does the data look like?
- **How do we find the signal?**
- Implementation/ Current detection tools in use by LVK

Matched Filtering

If we know what the signal looks like, we can use matched filtering to find signals in the data.

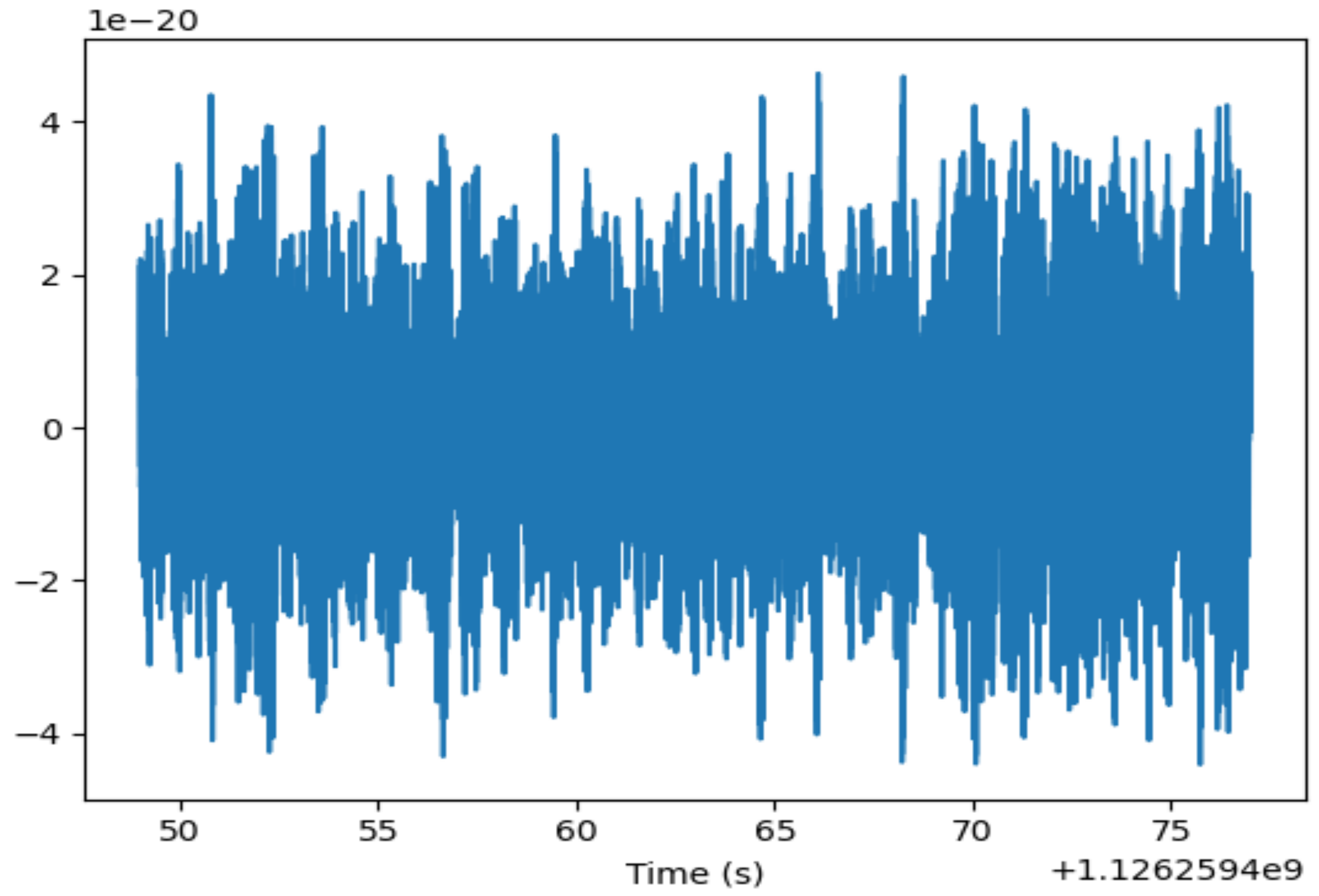
Matched filtering is a **correlation of a template waveform with the data**.

$$(s\,|\,h) = 4\mathbb{R}\int_0^\infty \frac{\tilde{s}(f)\tilde{h}^*(f)}{S_h(f)}df$$

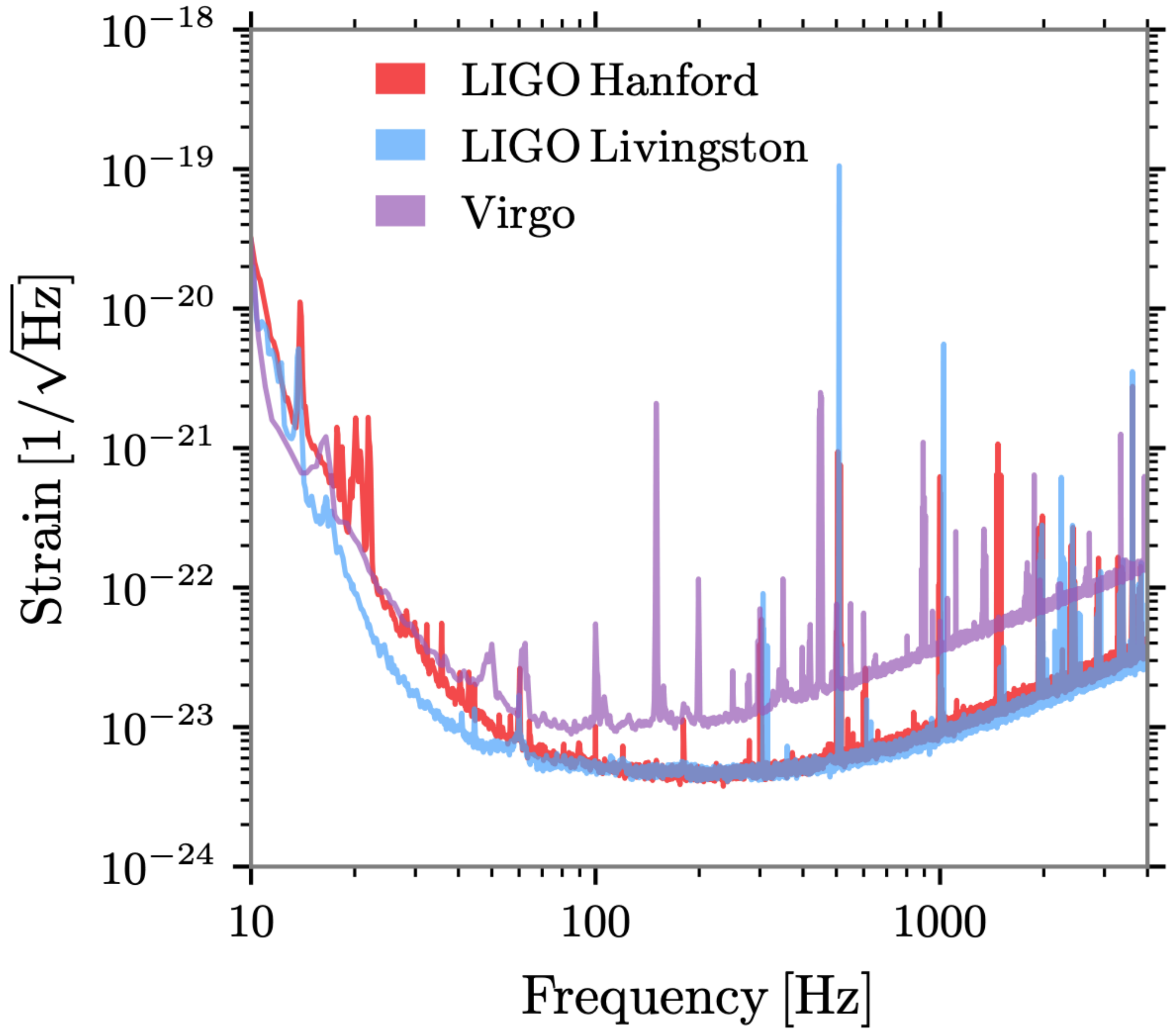


Signal templates

Strain data



PSD



Matched Filtering

Step 1: Whiten the detector data: $\boldsymbol{d} \rightarrow \dfrac{\tilde{d}(f_i)}{\sqrt{S_n(f_i)}}$

Whitening normalises the power at all frequencies so that any excess power at any frequency becomes obvious.

Step 2: Whiten the template: $\boldsymbol{h} \rightarrow \dfrac{\tilde{h}(f_i \mid \boldsymbol{\theta}')}{\sqrt{S_n(f_i)}}$

Adjust the template's amplitude at each frequency to account for the detector's noise level

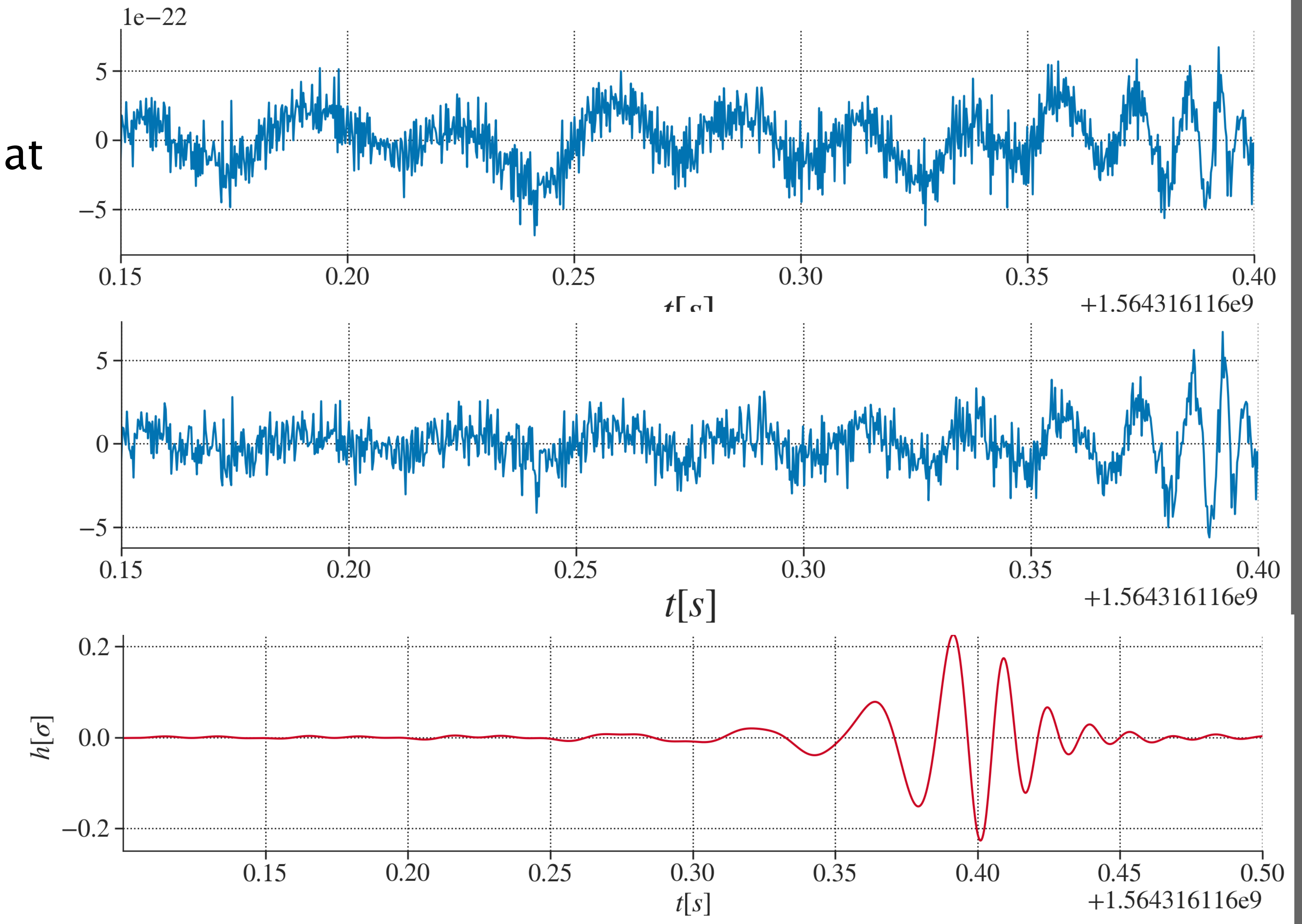
Step 3: Calculate the optimal signal-to-noise ratio (SNR) of the template

$$\rho_{\text{opt}}^2 = \left(h \mid h\right) = 4\Re \sum_{f_i} \frac{\tilde{h}^*(f_i \mid \boldsymbol{\theta})\tilde{h}\left(f_i \mid \boldsymbol{\theta}'\right)}{S_n(f_i)} \Delta f,$$

Δf = frequency resolution

Step 4: Cross correlate the whitened data and whitened normalised template

$$\rho = \frac{\left(d \mid h\right)}{\sqrt{\left(h \mid h\right)}} \rightarrow \text{matched-filter SNR}$$



Sathyaprakash + Dhurandhar, PRD 44, 3819 (1991)

Matched Filtering

The output of matched filtering is the **SNR time-series**.

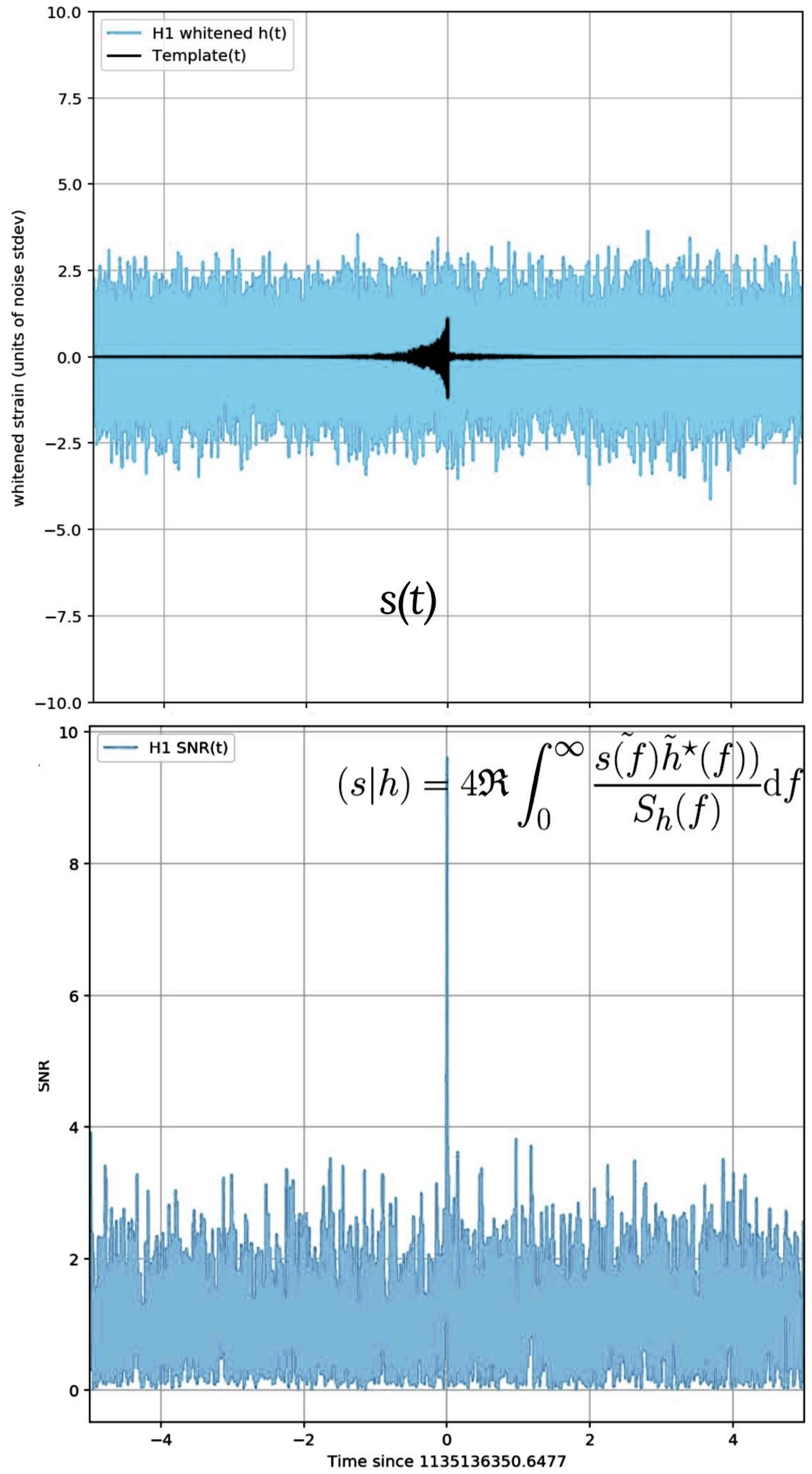
Peaks in the SNR time-series are used to identify signals in the data.

A coincident SNR peak in the data from multiple detectors increases the significance.

Optimal for signals:

- in stationary Gaussian noise
- with known PSD

(Wainstein and Zubakov, 1962)



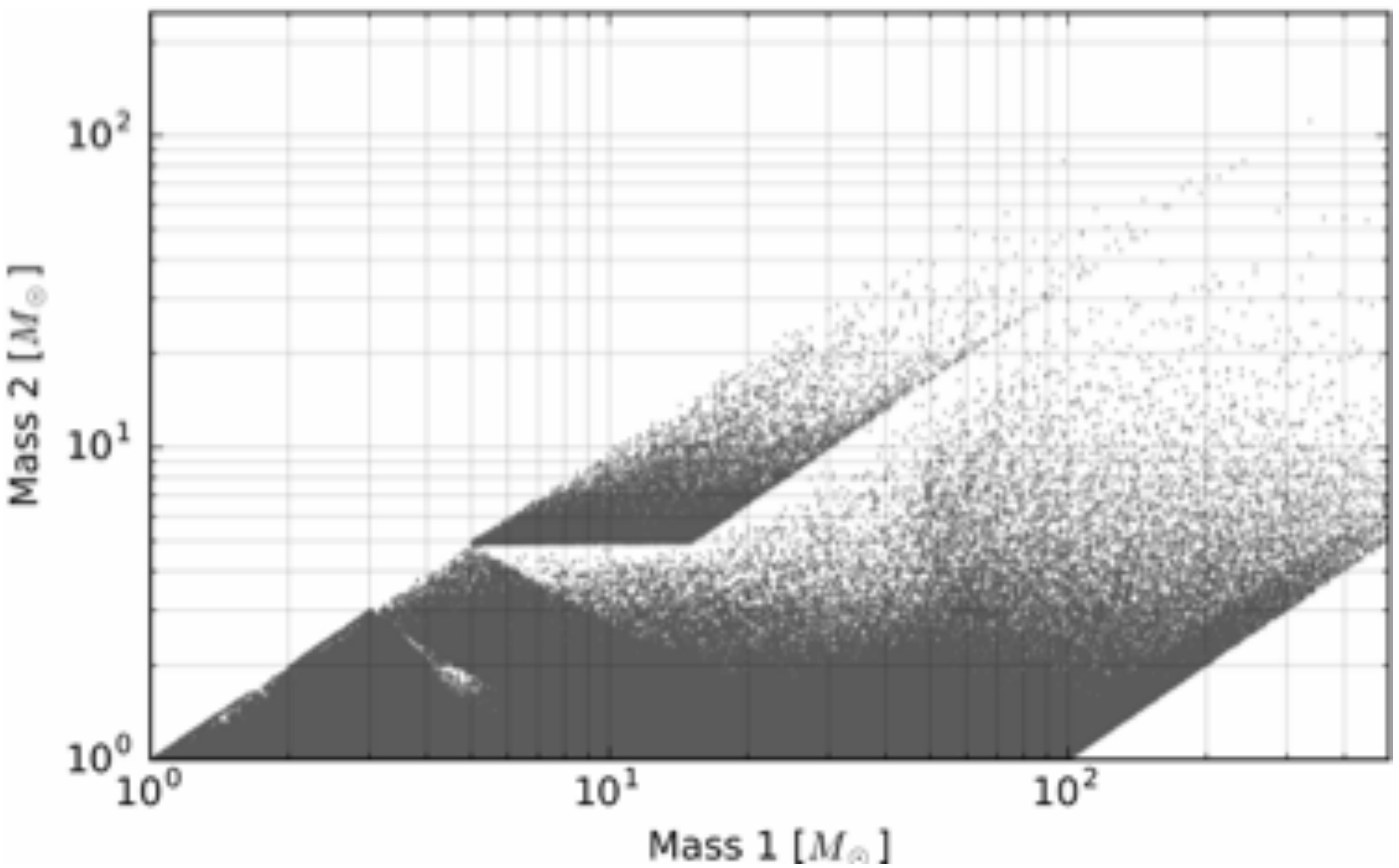
Template Banks

Matched filtering relies on **knowing the shape of the signal**.

We construct template banks of waveforms that vary over the intrinsic parameters.

4D template banks - $\{m_1, m_2, s_{1z}, s_{2z}\}$

$$N_{\text{templates}} \sim \mathcal{O}(10^5 - 10^6)$$



Template bank used for the PyCBC-Broad search for many recent publications, and contains ~400k templates

Image credit: Dal Canton and Harry (2017)

Template Banks

If the signal perfectly matches the template we will have an optimal SNR.

Any mismatch causes an SNR loss.

The template bank is chosen such that even for signals lying between the templates, we lose no more than 3% of the optimal matched-filter SNR.

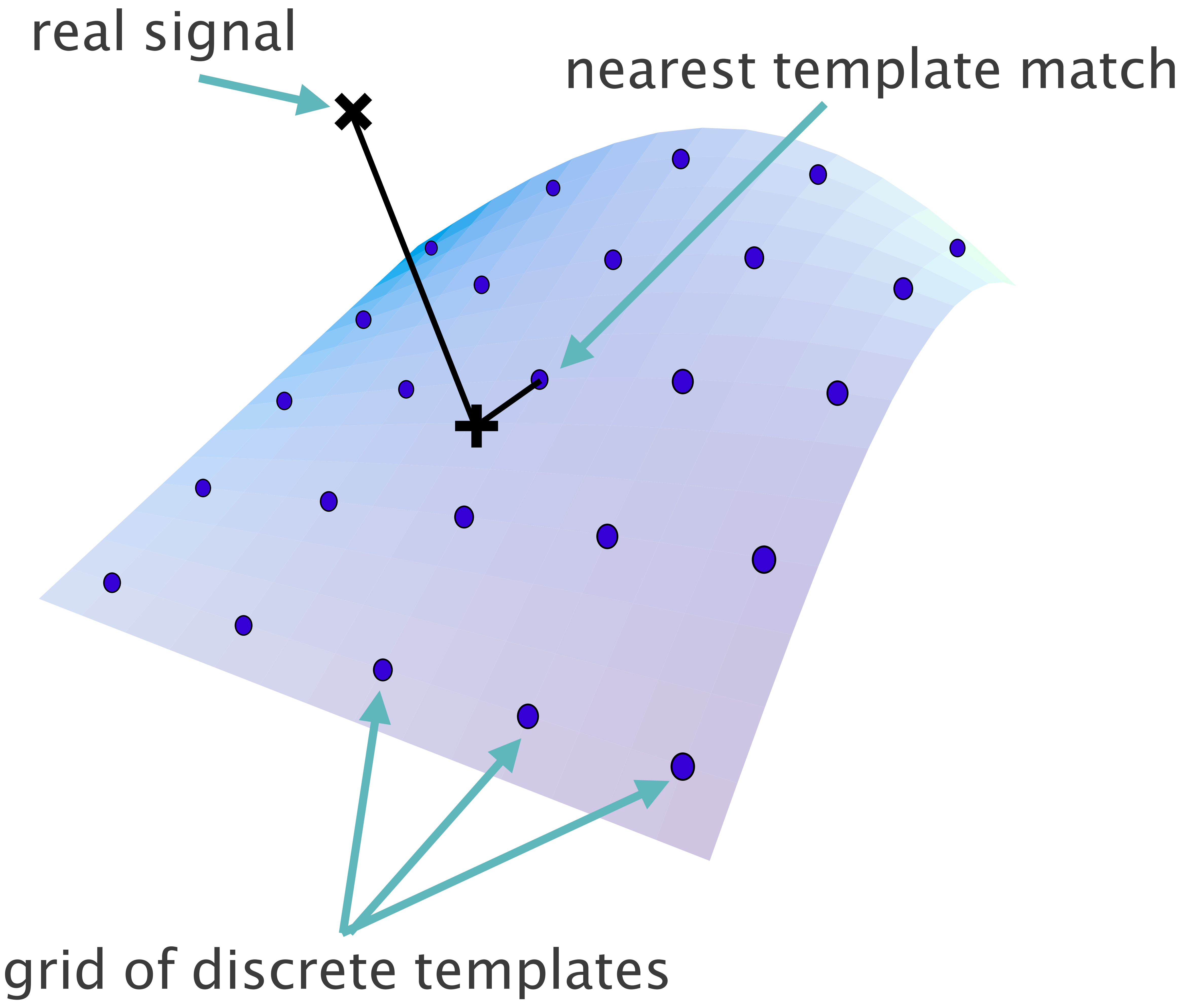
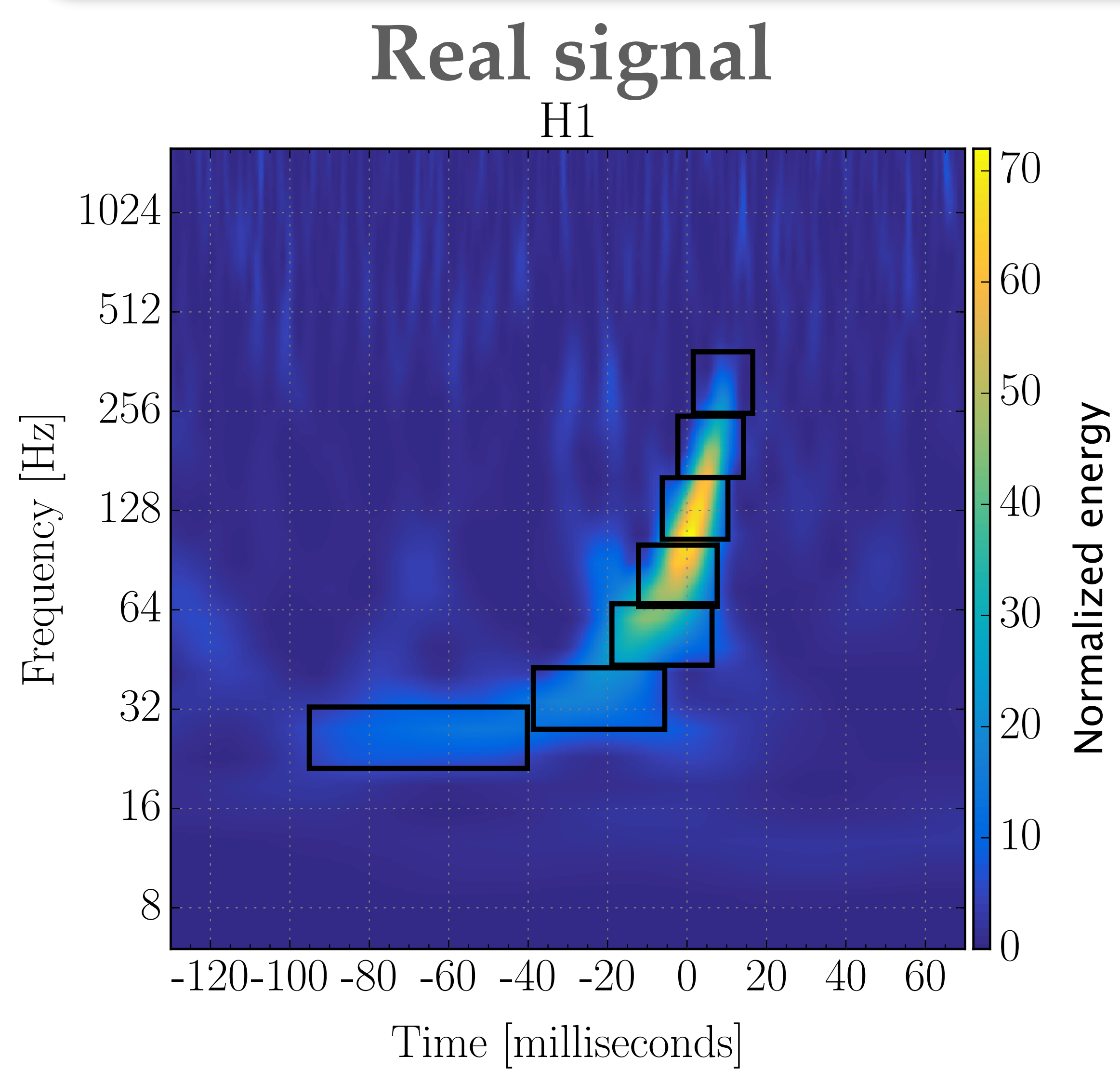
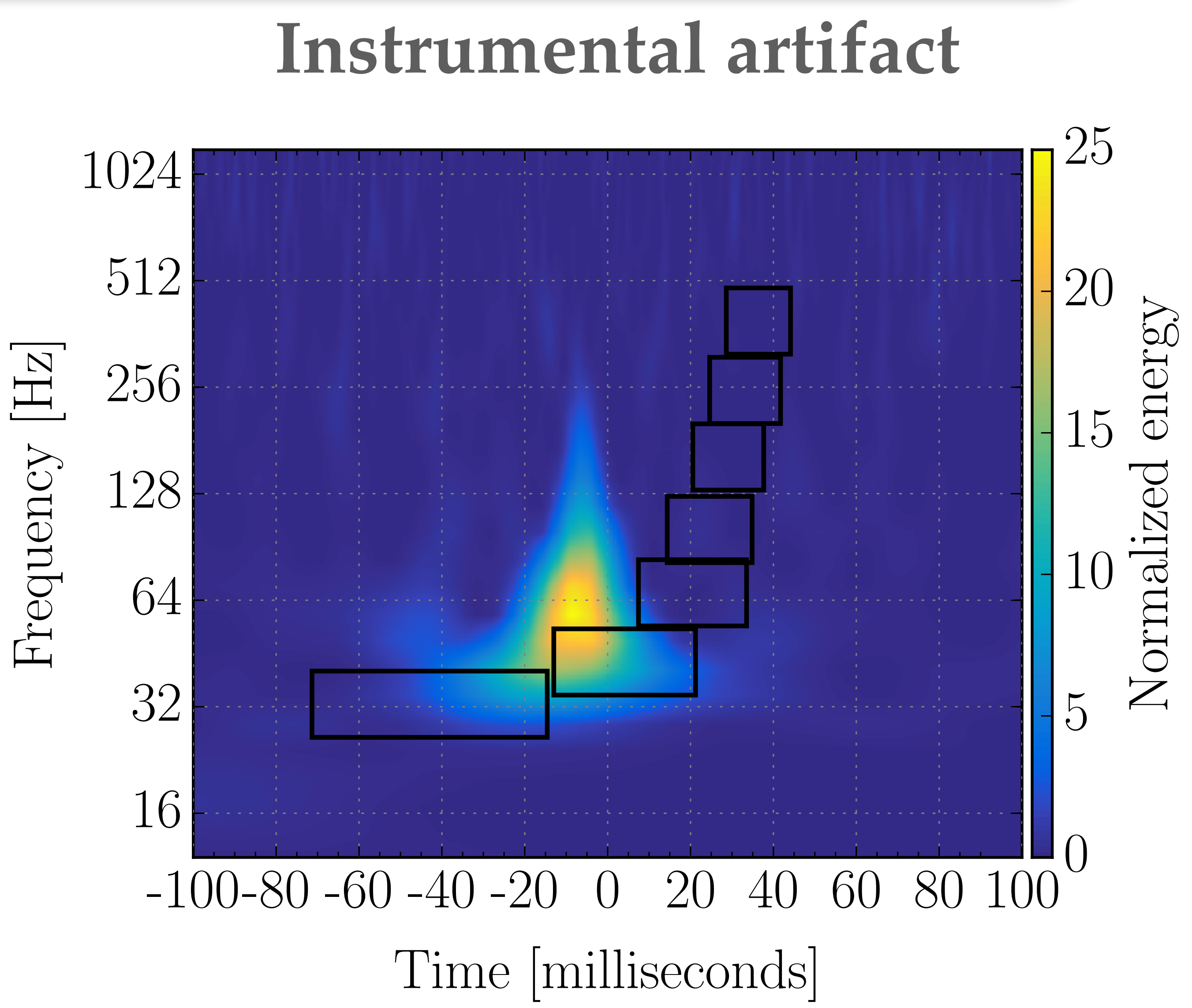


image credit: arXiv:gr-qc/9511032

Signal Consistency Tests

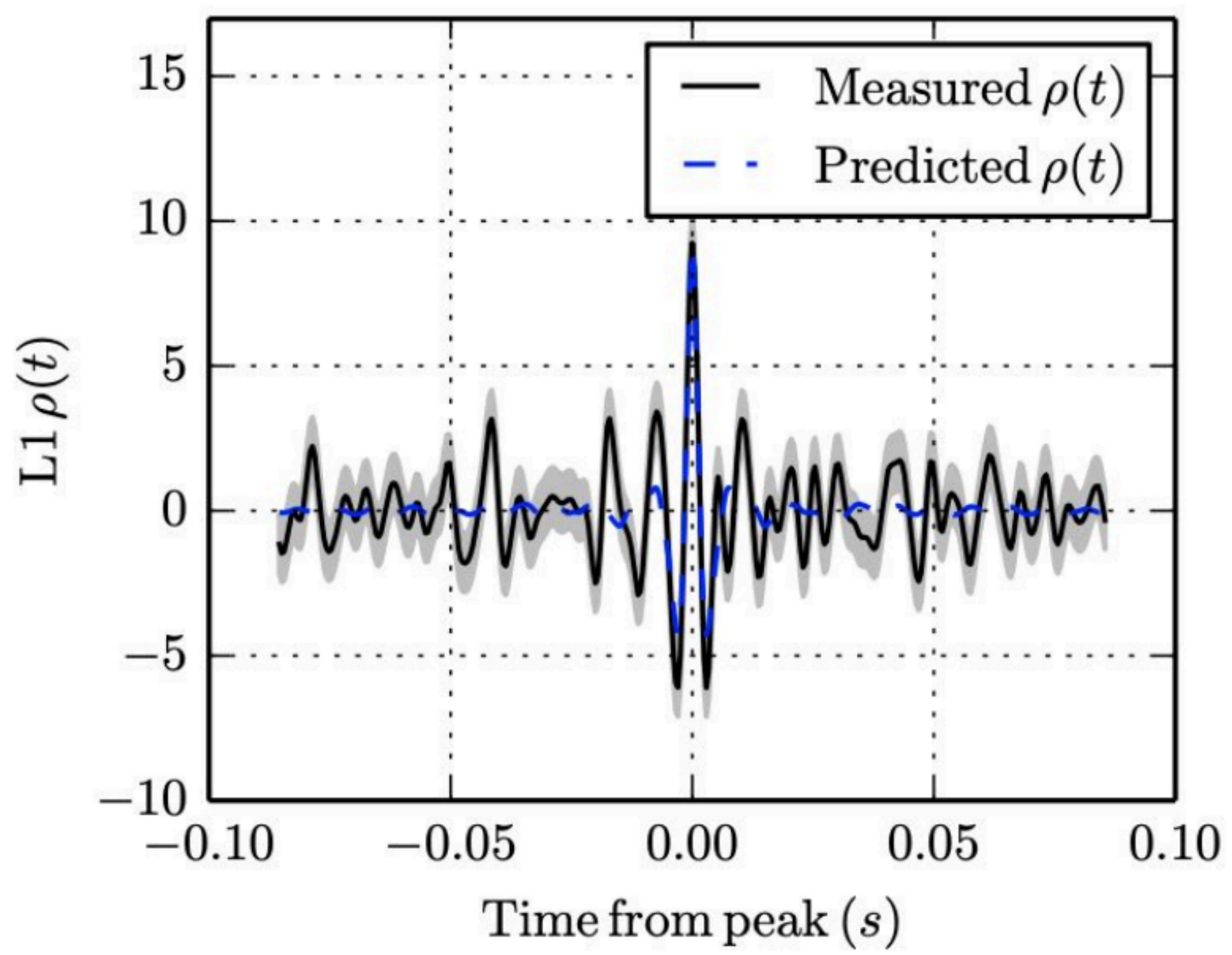
χ_r^2- test





- Step-1: Divide the template into p frequency bands of equal expected power.
- Step-2: Calculate $\chi_r^2 = \frac{p}{2p-2}\sum_{l=1}^{p}\left(\rho_l^2 - \frac{\rho^2}{p}\right)$
- Trigger consistent with template $\chi_r^2 \to 1$.
- Use χ_r^2-output to calculate $\varrho = f(\rho, \chi_r^2) \to$ amended ρ

Auto-correlation test



- Matched filtering doesn't produce just an SNR peak, but a time-series of SNR data.
- Compare the SNR time-series shape to the predicted shape for a template waveform.

Allen PRD 71 (2005) 062001 SB, …, IH, SP et al. PRD 87 (2013) 024003

Focus Questions

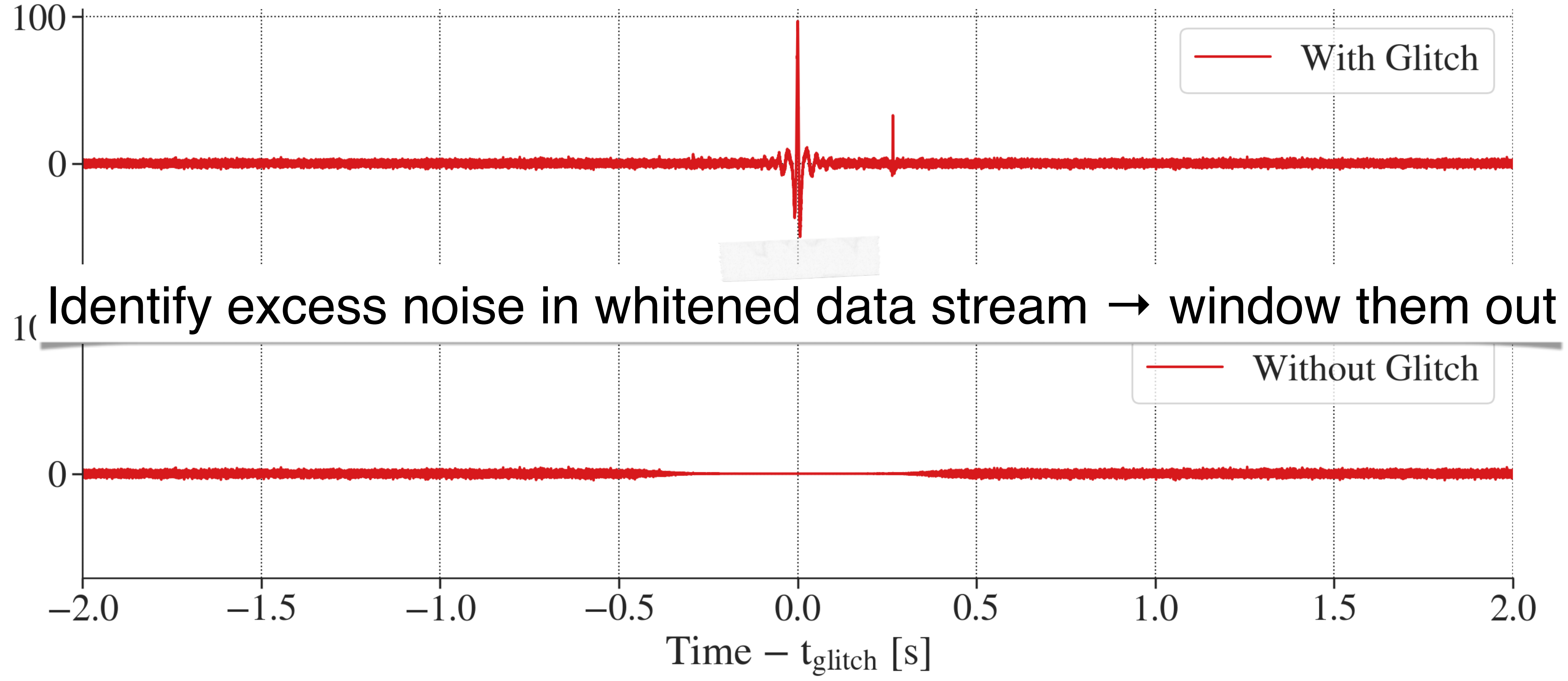
- Introduction
- What are we looking for?
- What does the data look like?
- How do we find the signal?
- **Implementation/ Current detection tools in use by LVK**

Current CBC search pipelines in use

- **Templated Searches**
  - PyCBC – https:/pycbc.org
  - GstLAL– https:/lscsoft.docs.ligo.org/gstlal/
  - MBTA – T. Adams et al., Class. Quant. Grav., 33 175012 (2016)
  - SPIIR – Q. Chu, Ph.D. Thesis, The University of Western Australia. (2017)
- **Non-templated Search**
  - cWB – gwburst.gitlab.io

    Identifying excess in energy coherently across the detector network (in time-frequency plane) to look for transient GW events.

Gating



Identify excess noise in whitened data stream → window them out

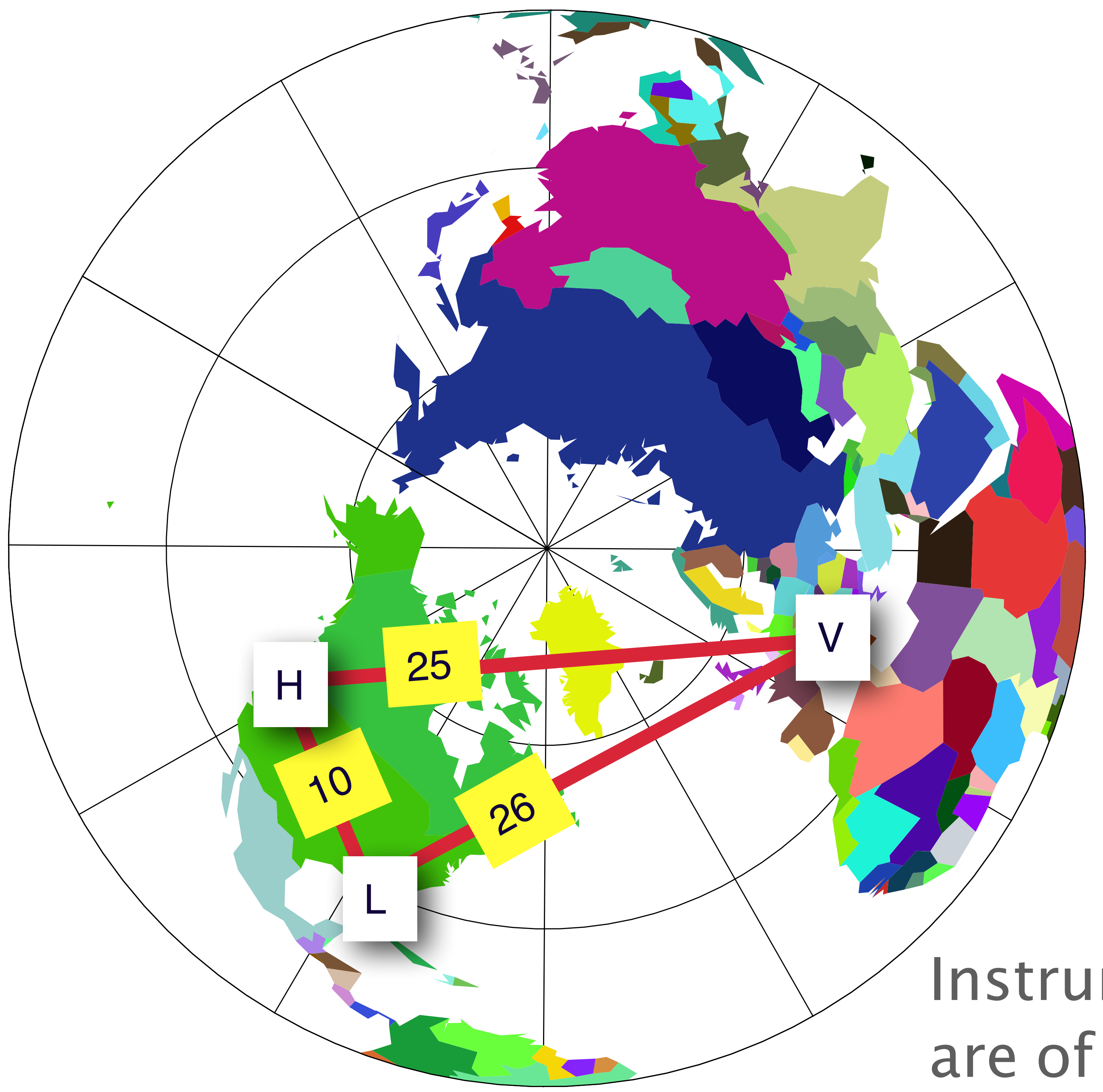
integrated Data Quality (iDQ)

- Use machine learning and data from auxiliary channels to identify glitches.
- Clean data → improves statistical significance

Coincidence test

If the trigger is of astrophysical origin:

- must be observed within physically allowed time-delays across the detector network.
- must share the same best-matched template



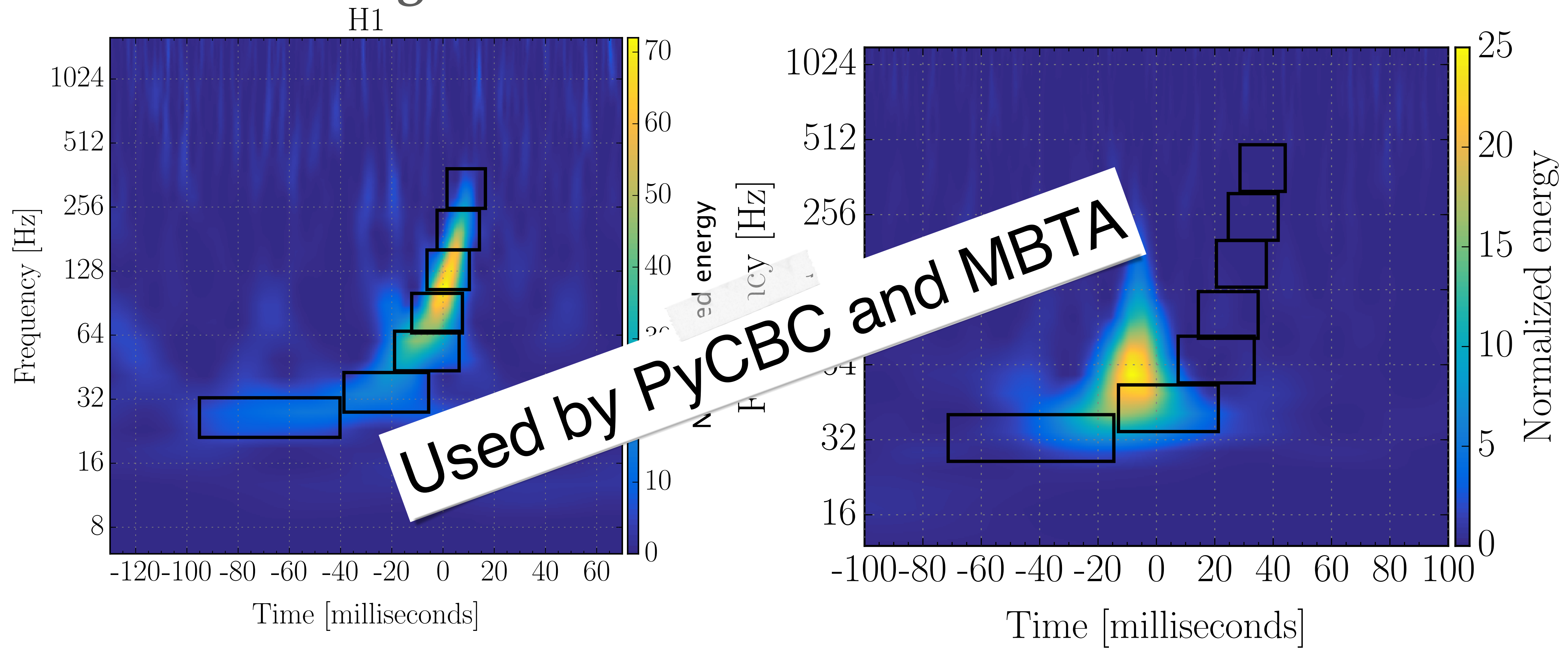
Instrumental noises are of local origin

Courtesy: Sathya

Signal Consistency Tests

χ_r^2- test

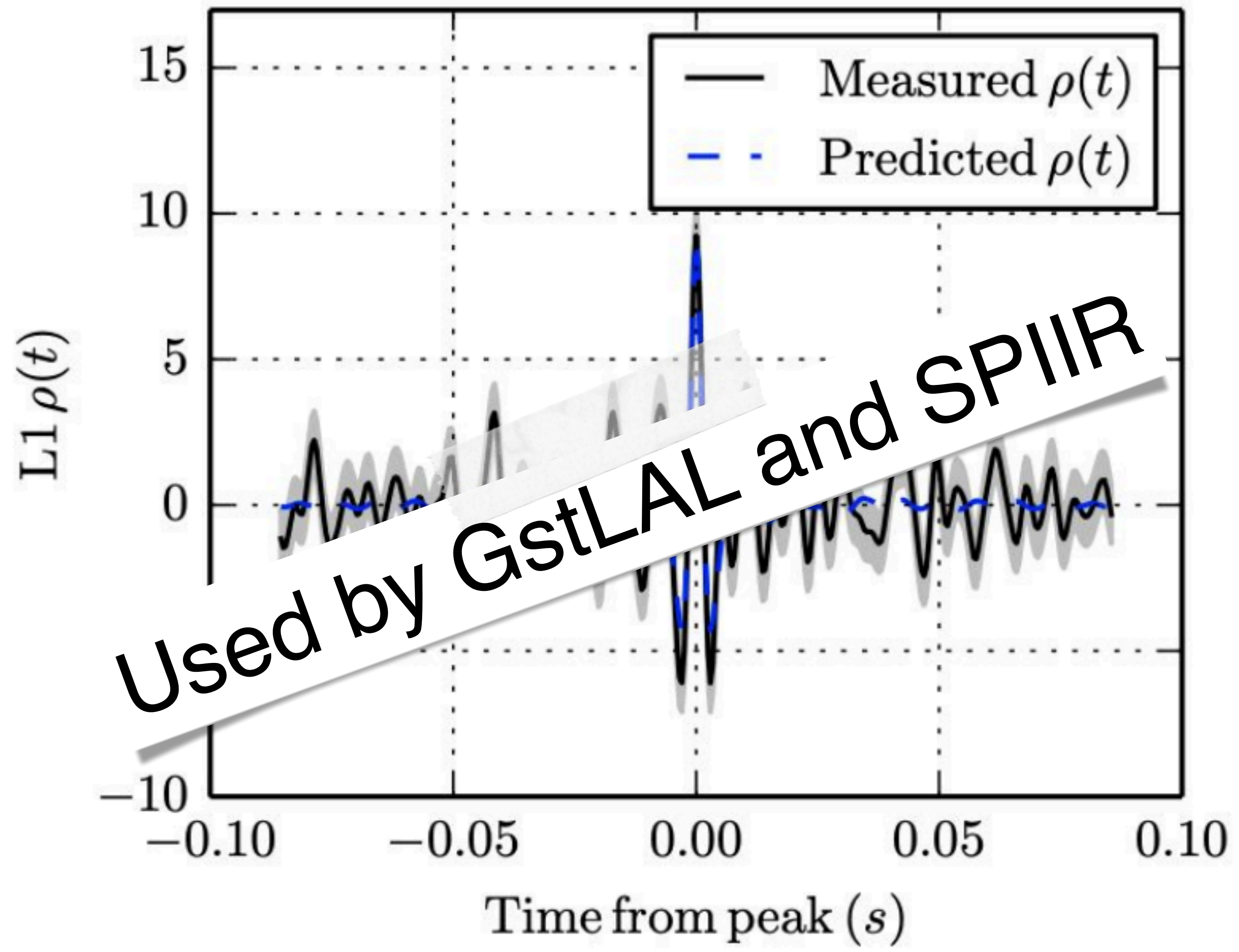
Real signal Instrumental artifact



Used by PyCBC and MBTA

- Step-1: Divide the template into p frequency bands of equal expected power.
- Step-2: Calculate $\chi_r^2 = \dfrac{p}{2p-2} \sum_{l=1}^{p} \left(\rho_l^2 - \dfrac{\rho^2}{p} \right)$
- Trigger consistent with template $\chi_r^2 \to 1$.
- Use χ_r^2-output to calculate $\varrho = f(\rho, \chi_r^2) \to$ amended ρ

Auto-correlation test



Used by GstLAL and SPIIR

- Matched filtering doesn't produce just an SNR peak, but a time-series of SNR data.
- Compare the SNR time-series shape to the predicted shape for a template waveform.

Allen PRD 71 (2005) 062001 SB, …, IH, SP et al. PRD 87 (2013) 024003

MBTA and PyCBC both use **gating** and signal consistency tests similar to GstLAL.

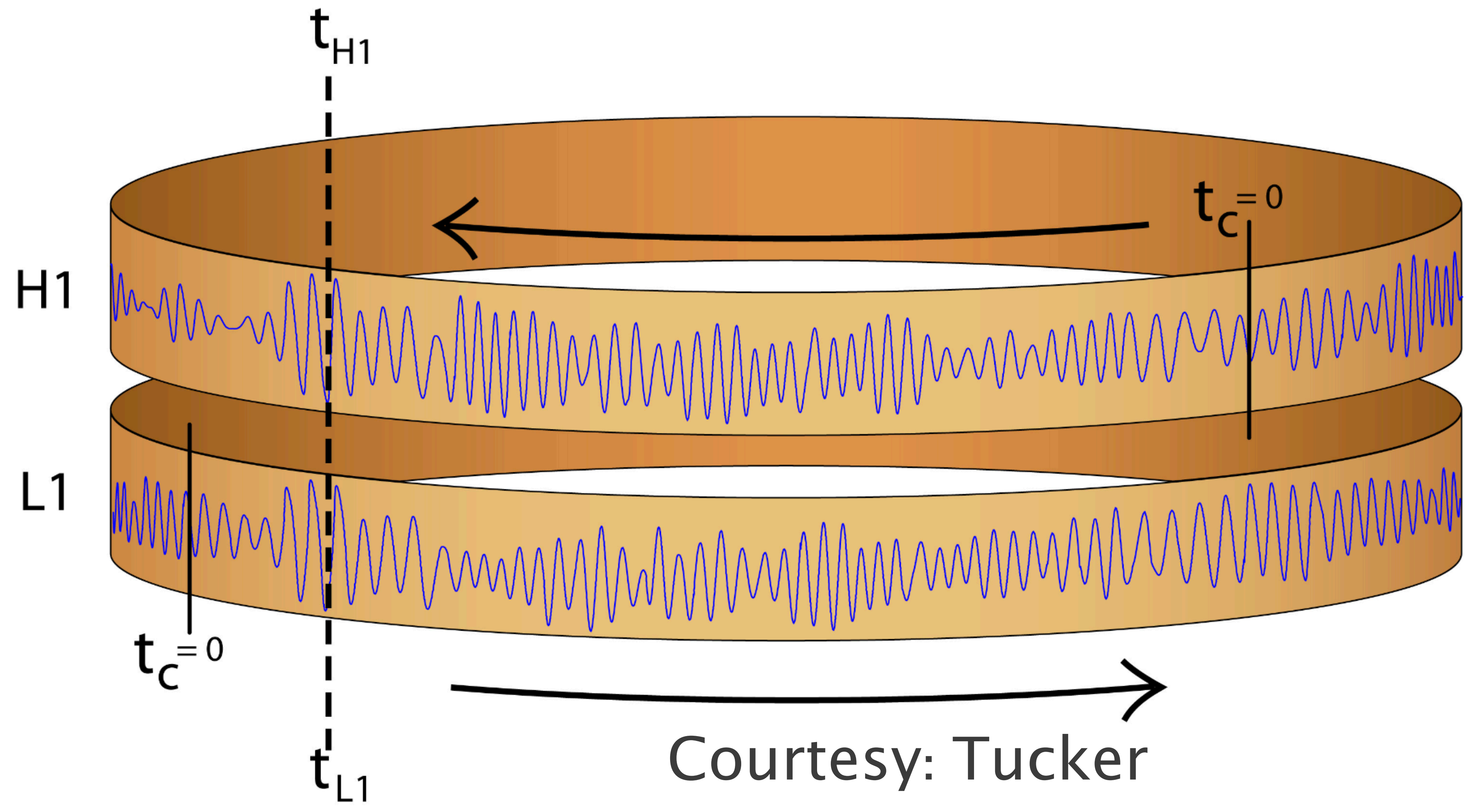
MBTA and **PyCBC do not use iDQ**, but they do use **vetoes** to reject triggers from times identified to have poor data quality.

MBTA: re-filter high mass templates

- Heavy BBH systems can be mistaken for glitches due to their short duration.
- MBTA re-filters templates with duration < 3 sec. without gating first
- This helps to avoid accidentally gating real signals

Statistical Significance

Step-1: Rank coincident candidates → PyCBC calculates $R = \sqrt{\sum_{k=1}^{N_{\rm ifo}} \varrho_I^2}$, where $N_{\rm ifo}$= # of detectors in the network. Other pipeline do this differently

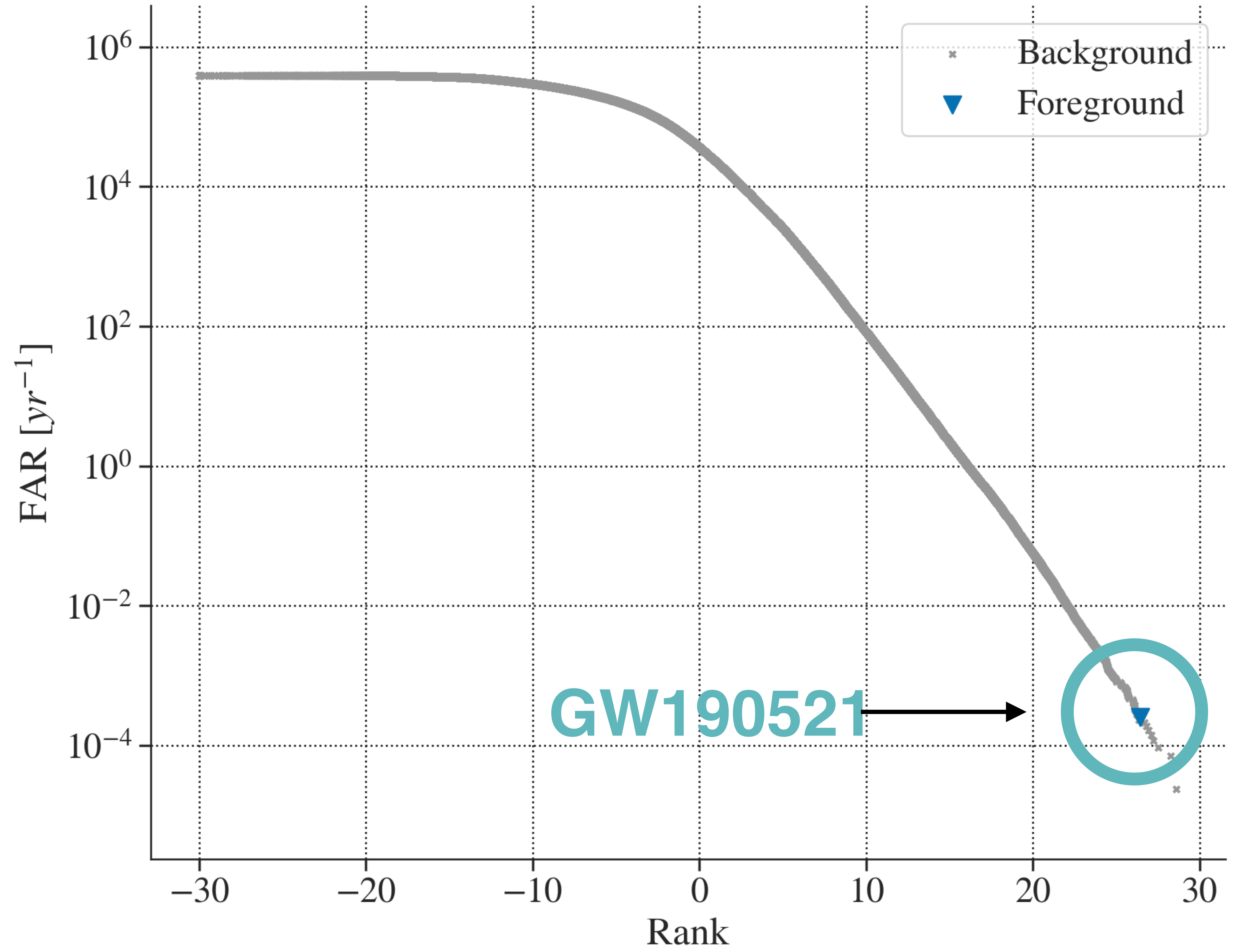


Step-2: Generate background triggers by **time-slides method** → shift one detector's data with respect to other(s) and look for accidental coincidences. (GstLAL doesn't use this method.)

Step-3: Calculate false alarm rate or $\mathrm{FAR} := \frac{1 + n_b(R_b > R)}{T_b}$,

n_b = # of background triggers with rank $R_b > R$ in time T_b

Related to false alarm probability $p = 1 - e^{-\mathrm{FAR} \times \mathrm{T}}$



Was et al. CQG 27 (2010) 015005

Summary

- GW signals from compact binary mergers are in general well-modelled.
- CBC searches find gravitational waves from **merging black holes** and **neutron stars** in LIGO/Virgo/KAGRA strain data.
- Current search rely on **matched filtering** - a correlation between the data and numerical or analytical models of the signal - to identify signals in the data.
- Real detection pipelines - **PyCBC, MBTA, GstLAL** and **SPIIR** - each employs similar but distinct methods to:
    - implement matched filtering
    - improve sensitivity in non-stationary non-Gaussian data, and
    - reduce computational cost