12th Iberian Gravitational Waves Meeting

Hierarchical approach to matched filtering using a reduced basis

Rahul Dhurkunde, Henning Fehrmann and Alexander H. Nitz

Phys. Rev. D 105, 103001

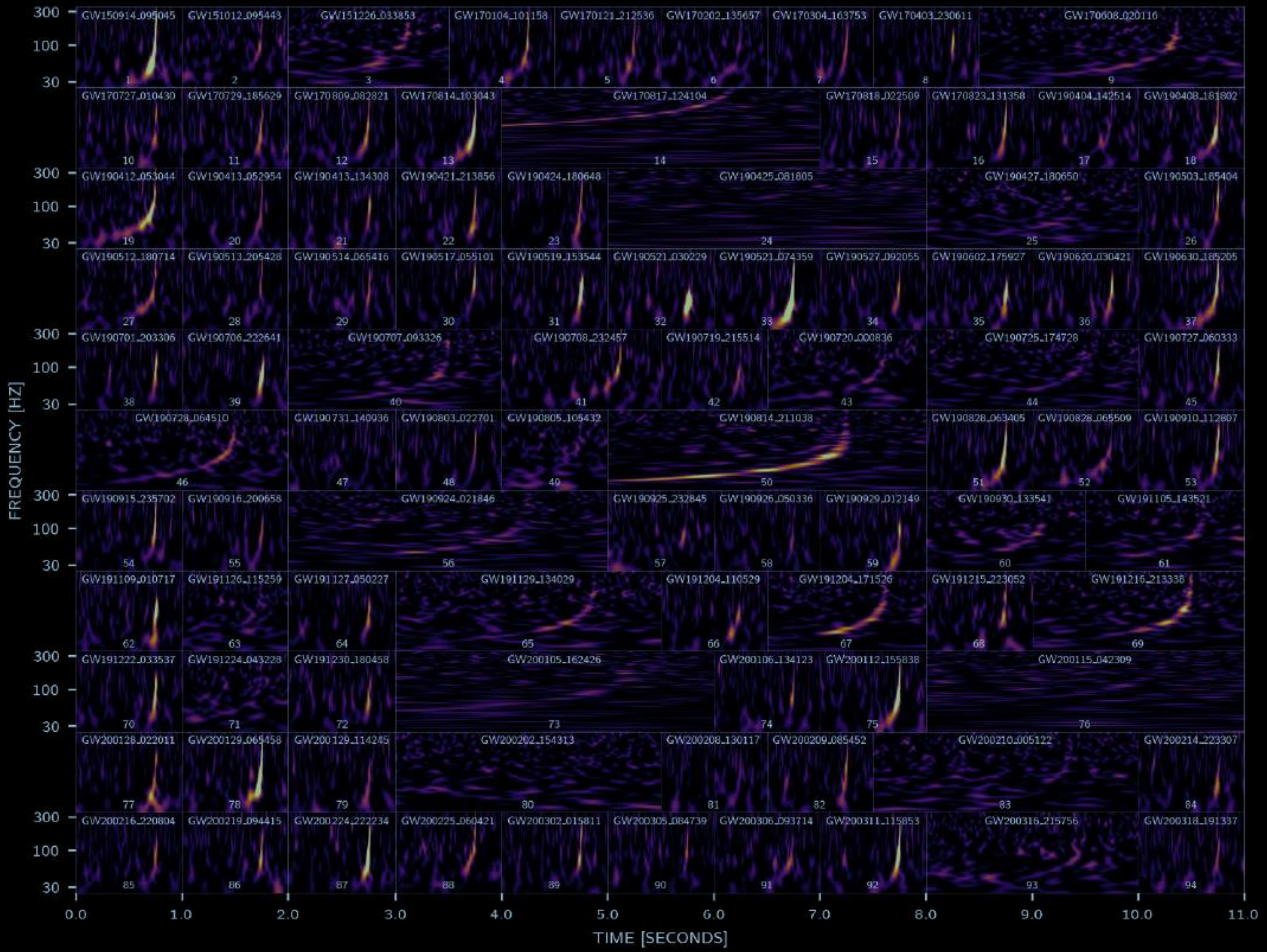
Max-Planck-Institut für Gravitationsphysik ALBERT-EINSTEIN-INSTITUT

MAX-PLANCK-GESELLSCHAFT

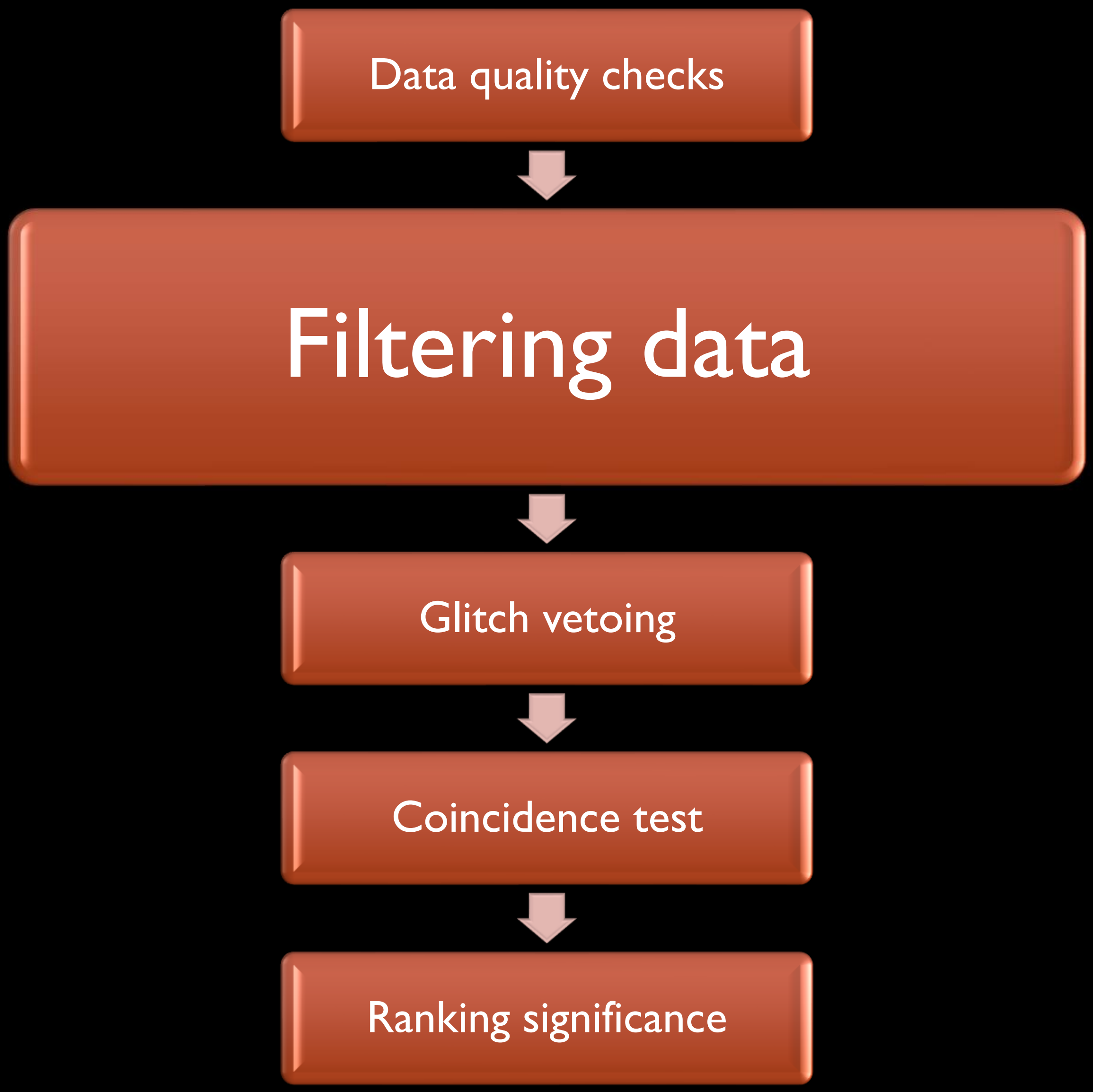
Leibniz Universität Hannover

4-OGC: Open Gravitational-wave Catalog 2015-2020

Alexander H. Nitz, Sumit Kumar, Yi-Fan Wang, Shilpa Kastha, Shichao Wu, Marlin Schäfer, Rahul Dhurkunde, Collin D. Capano



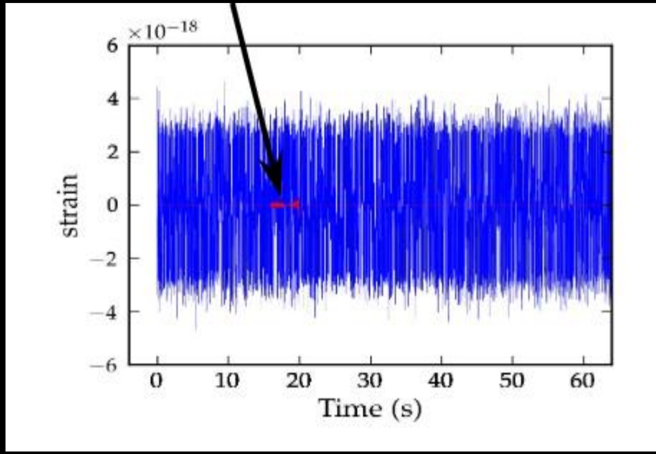
Modeled search



Matched filtering to extract the signal

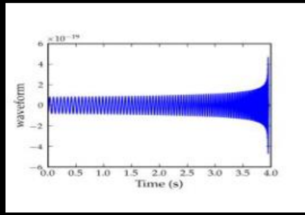
Optimal detection statistic → Likelihood of data containing a signal

Signal embedded in strain data **s(t)**

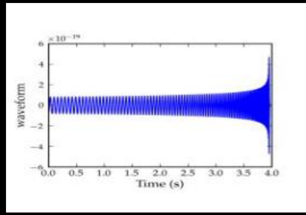


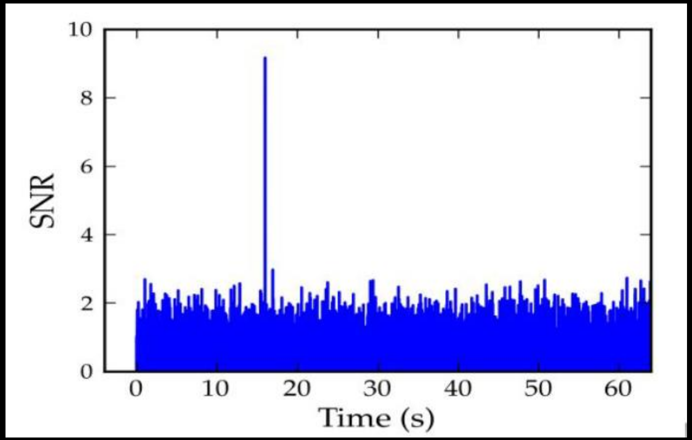
$\otimes$

Modeled waveform filters $h_\zeta(t)$



. . . .





$$\rho(\zeta) = \frac{(\tilde{s}|\tilde{h}_\zeta)}{\sqrt{(h|h)}}$$

$$(a|b) = 4\int_{-\infty}^{\infty} \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n} df$$

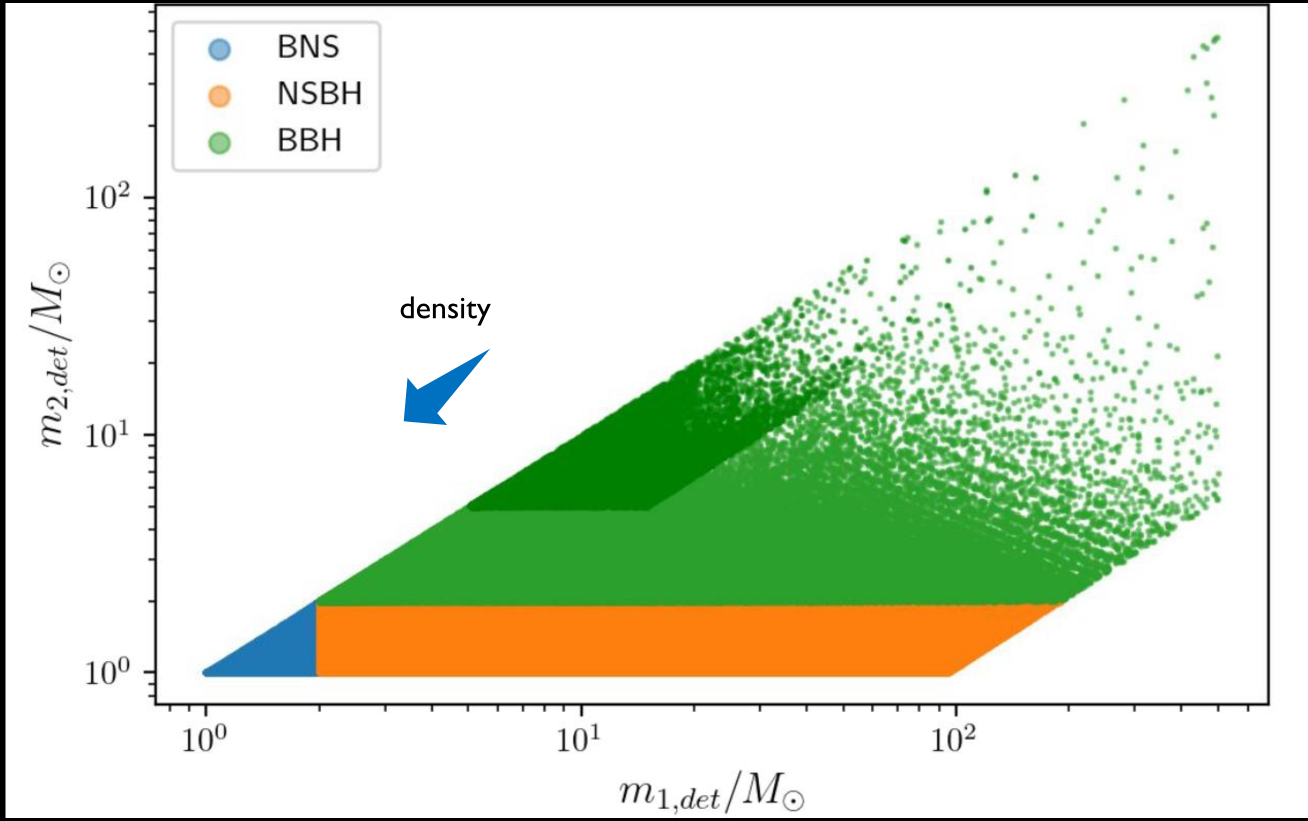
Search assumptions

Approximating the search statistic

Aligned spins | Quasi circular orbits | Dominant mode of GW emission

15 parameters (Likelihood maximization)

- Masses m_1, m_2
- Spins $\boldsymbol{S}_{1z}, \boldsymbol{S}_{2z}$
- Extrinsic parameters $A\,,\ \phi_0$ analytically
- Signal arrival time t_c FFTs

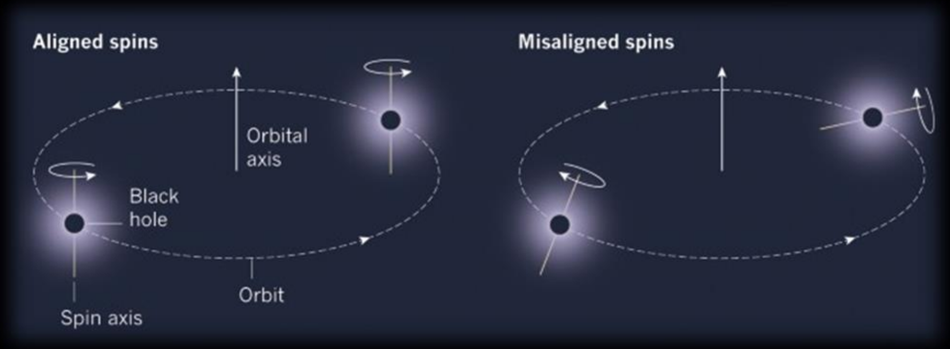


Aligned-spin template bank

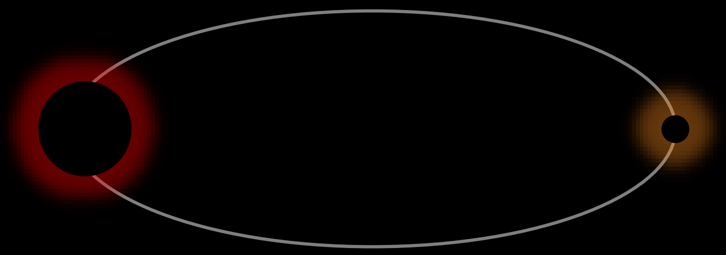
Computational costs $\propto$ No. of templates Signal duration

Current template banks are missing

Precessing systems



Eccentric binaries

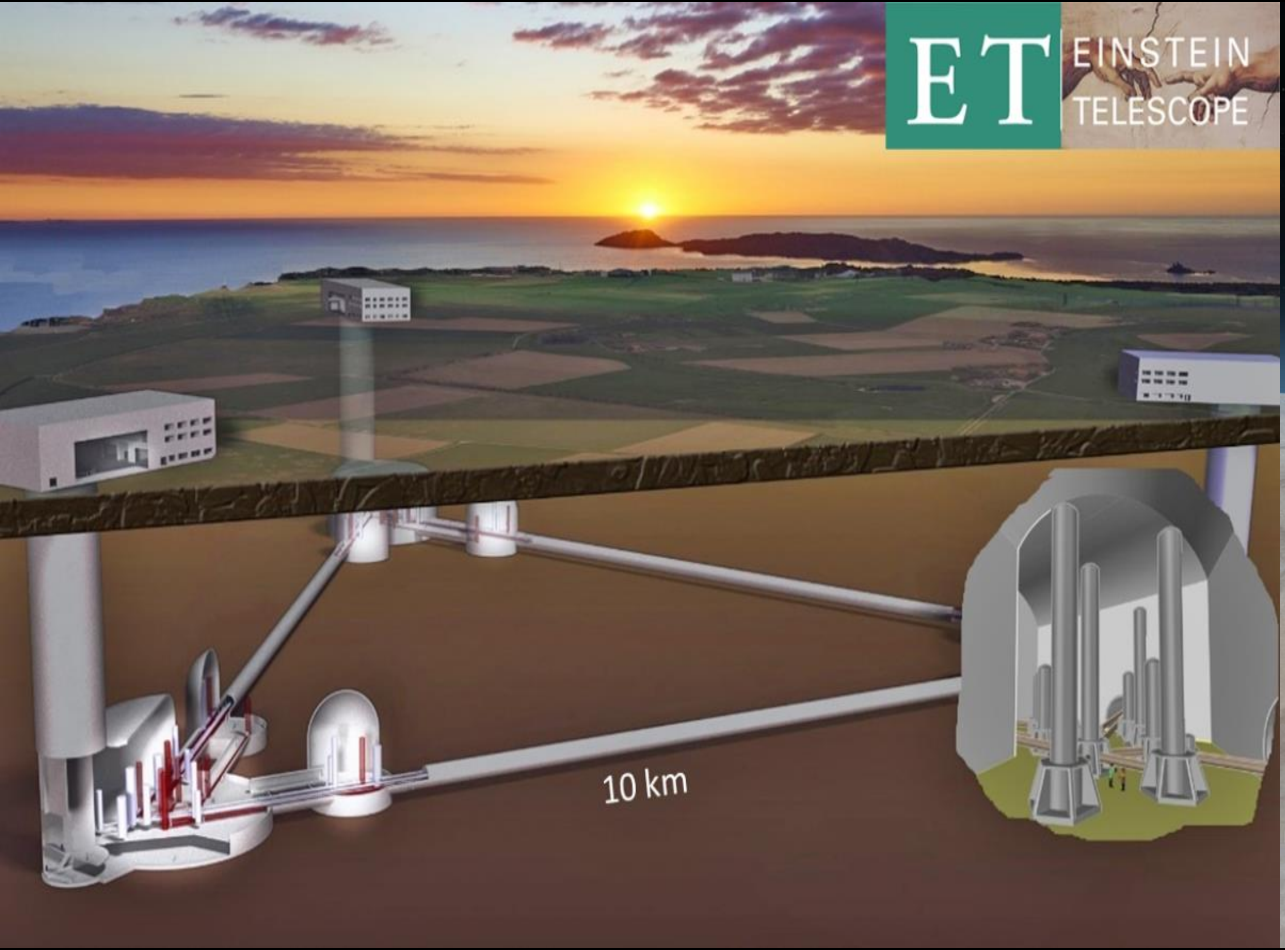


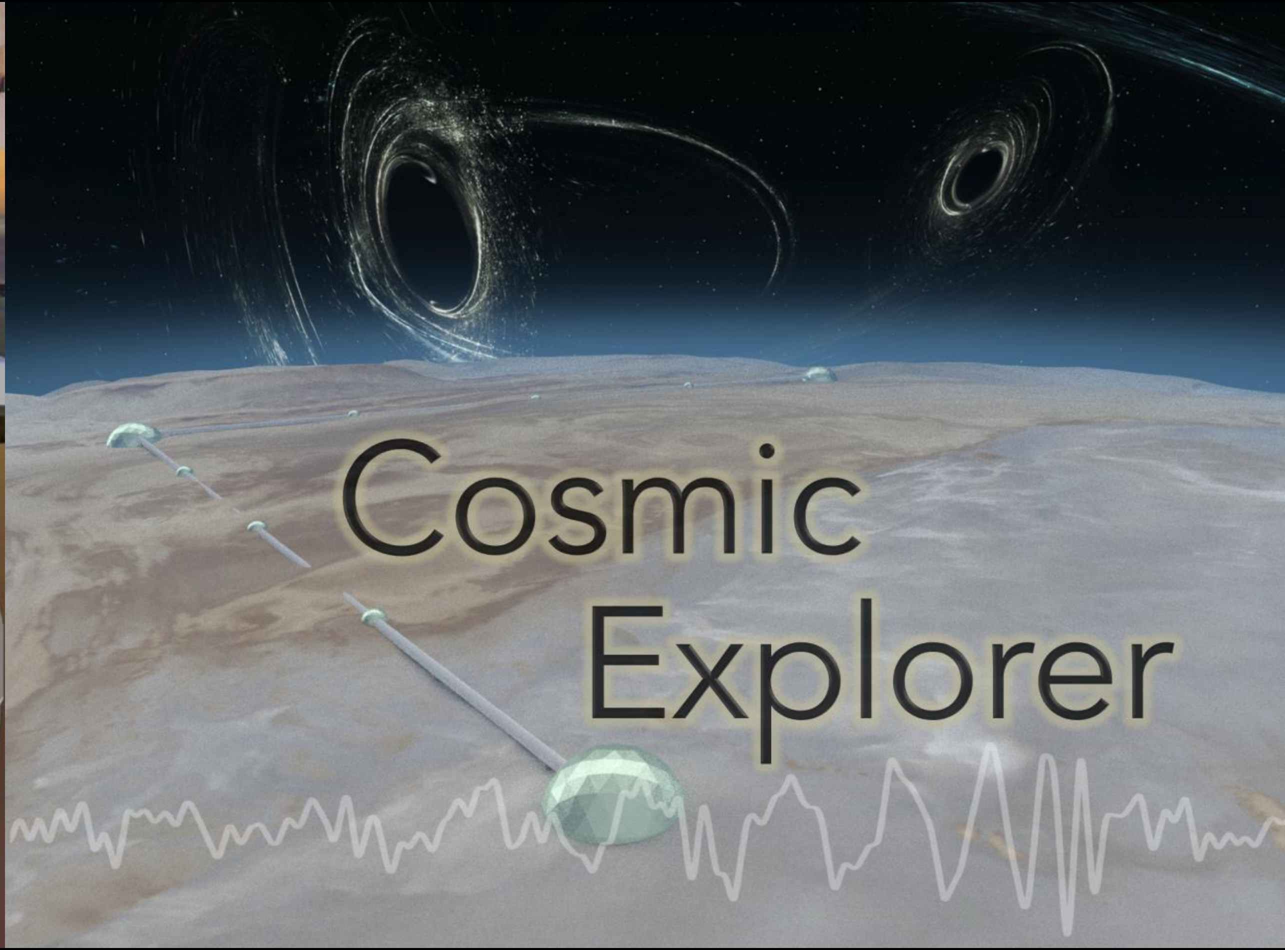
~100x bigger template banks

Computationally limited

Huge astrophysical implications

Third generation detectors





Better sensitivity at low frequencies → Longer templates → Larger costs

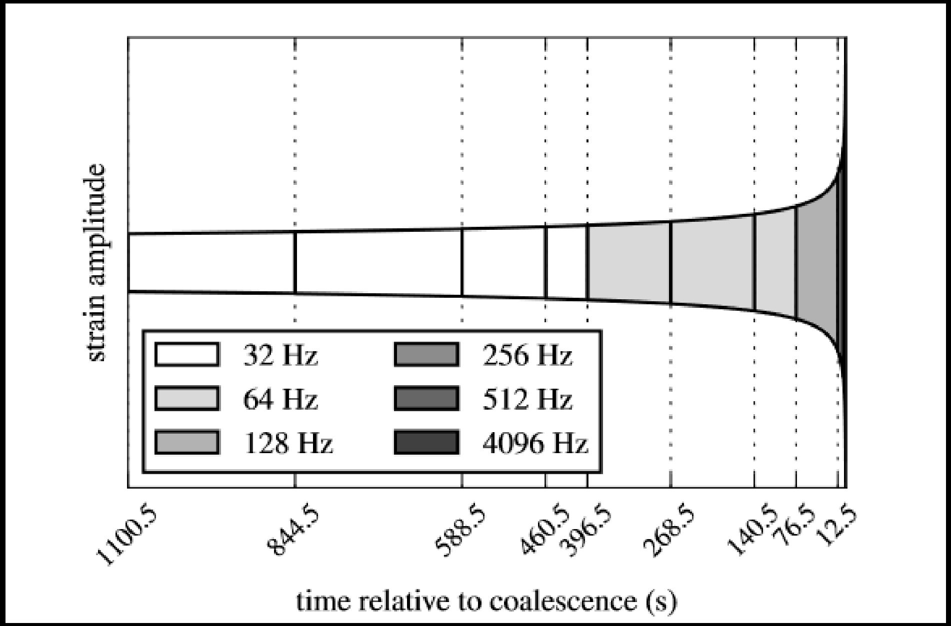
Improving the search performance

Multirate sampling with reduced basis

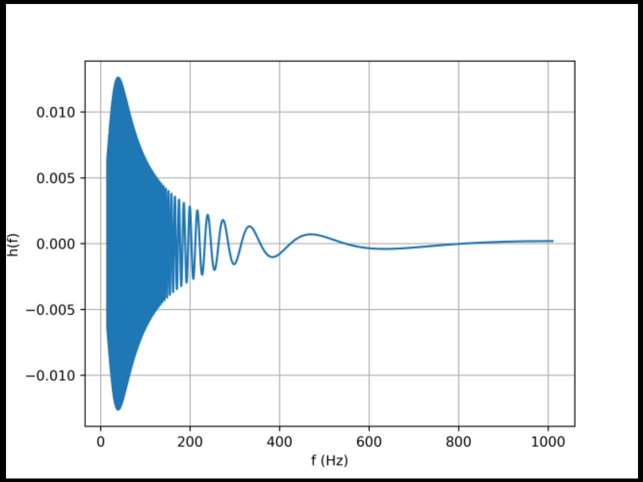
Nyquist's criterion

$$N = \frac{1}{(df\ dt)}$$ number of samples

[3]



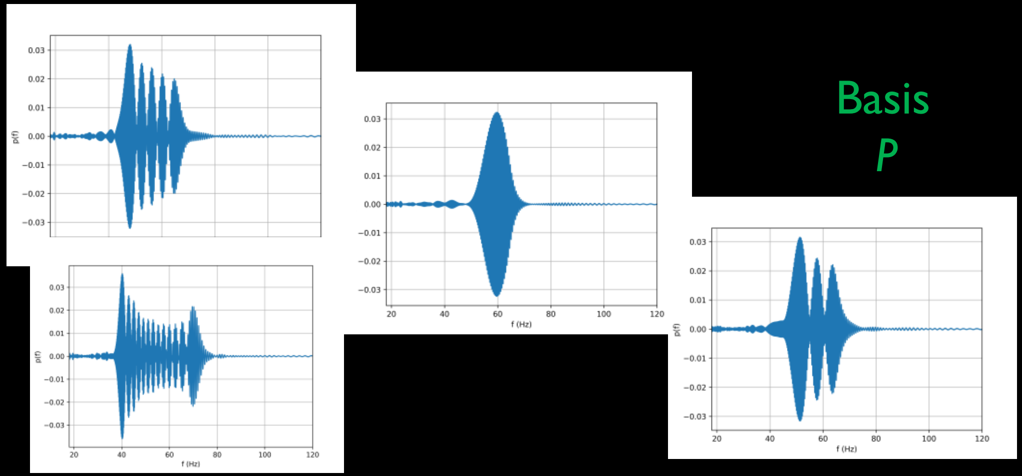
- Greatly reduces subsequent computations
- Used by GSTLaL, MBTA and SPIIR pipelines



Templates
T

$P < T$

$$\tilde{h}_\zeta(f) = \sum_{k=0}^{p_t - 1} c_{k,\zeta}\tilde{p}_k$$

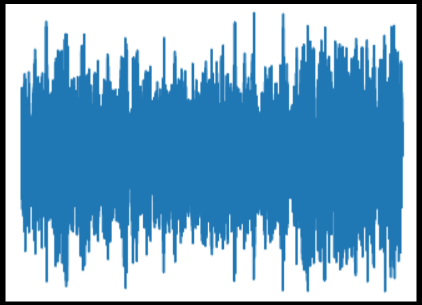


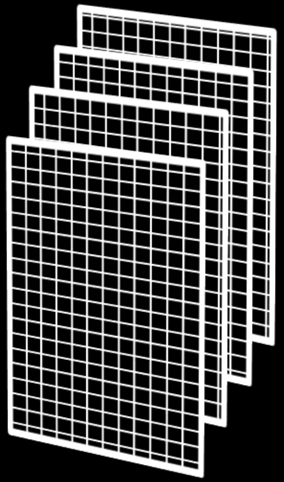
Basis
P

(Continued) Reduced basis matched filtering

$$\beta_k(t) = \mathrm{IFFT}(\langle \tilde{s} | \tilde{p}_k \rangle)$$

Filter output from the basis





Basis

β_0

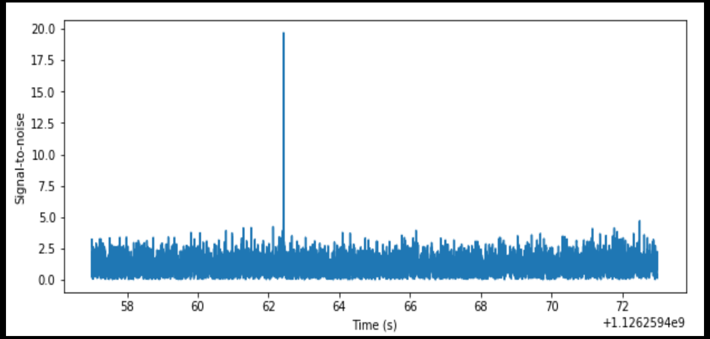
β_1

β_2

β_{p-1}

$$\rho_r(t) = \sum_{k=0}^{p-1} c^*_{k,\zeta} \beta_k$$

Reconstruction



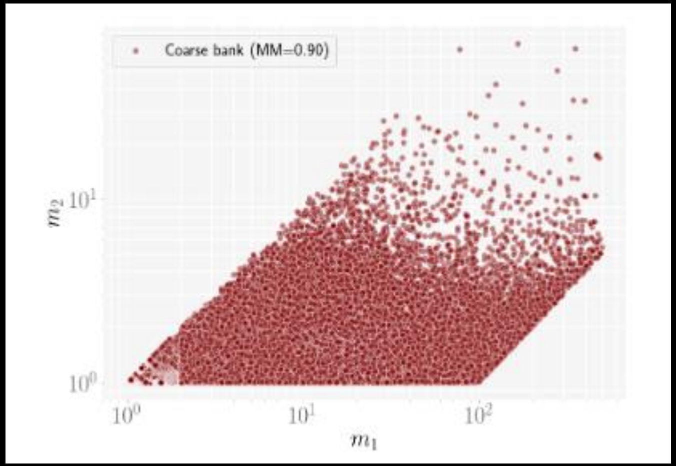
Final output

$\mathcal{O}(NT\log N)$ < $\mathcal{O}(NpT)$!! Expensive !!

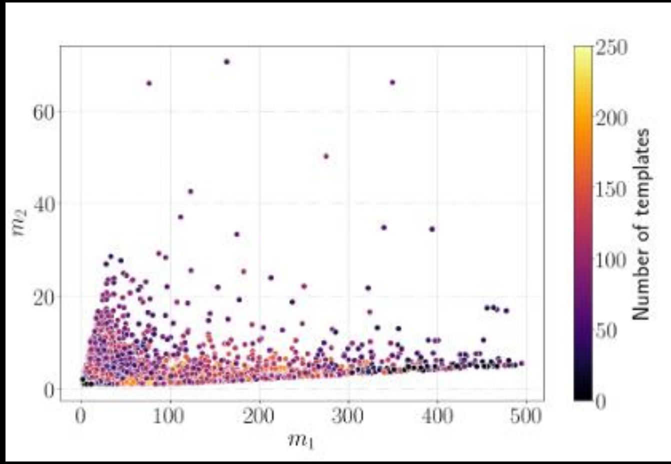
Hierarchical methods

- Two stage filtering using coarse and fine template banks
- Only foreground triggers are followed from 1st stage to 2nd
- This results in poor background estimation which can result in incorrect significance of an event
- Computational gain at the expense of sensitivity

Coarse bank



Fine bank



Phys. Rev. D 105, 064005

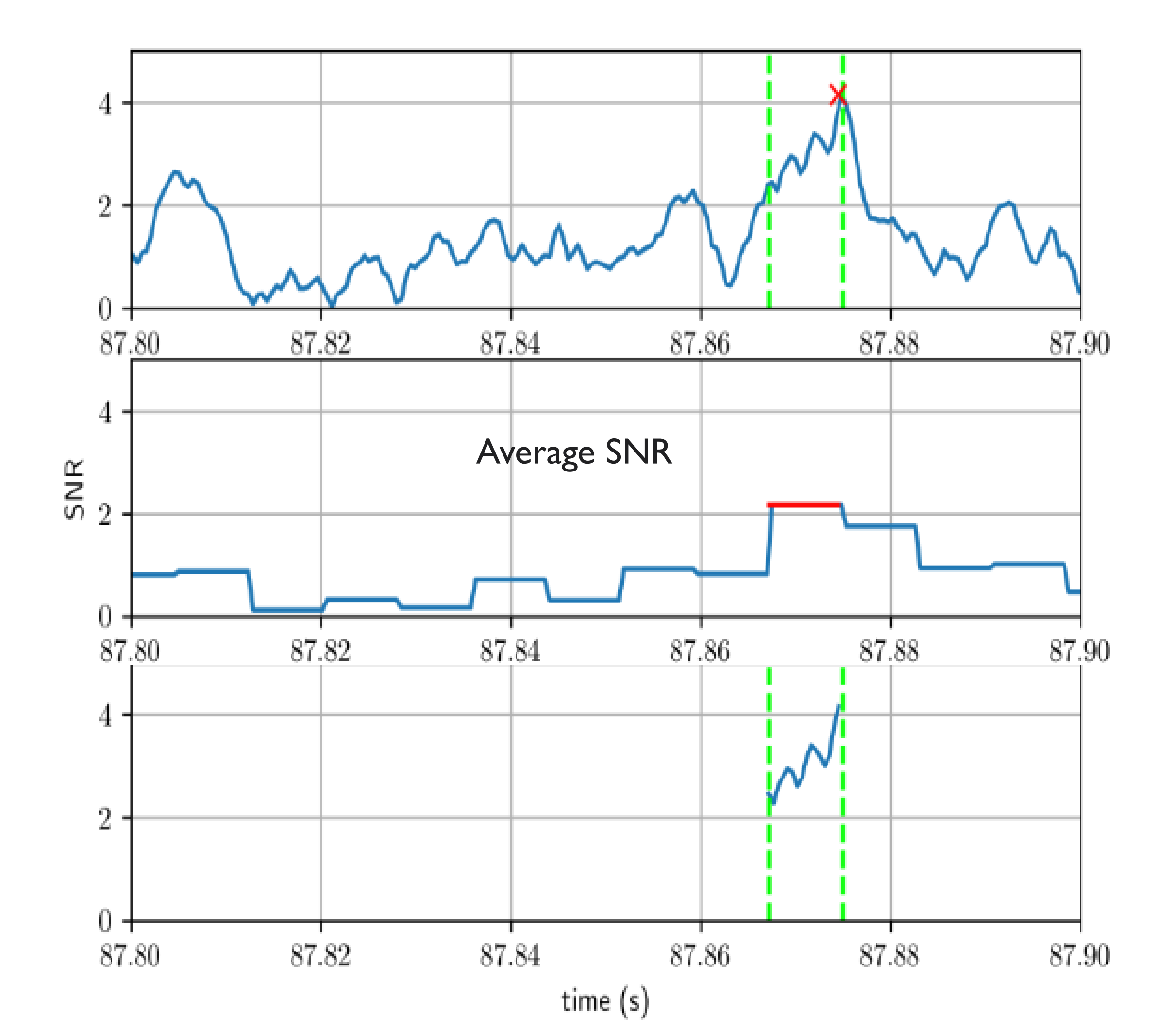
Reduced basis hierarchical method

Reconstructing SNR series (hierarchically)

Fully sampled
SNR time-series

First stage

Second stage



Down-sampling SNR series

Recovered using basis

Estimating the average SNR

Averaging of SNR series in time-domain can be performed in the Fourier-domain

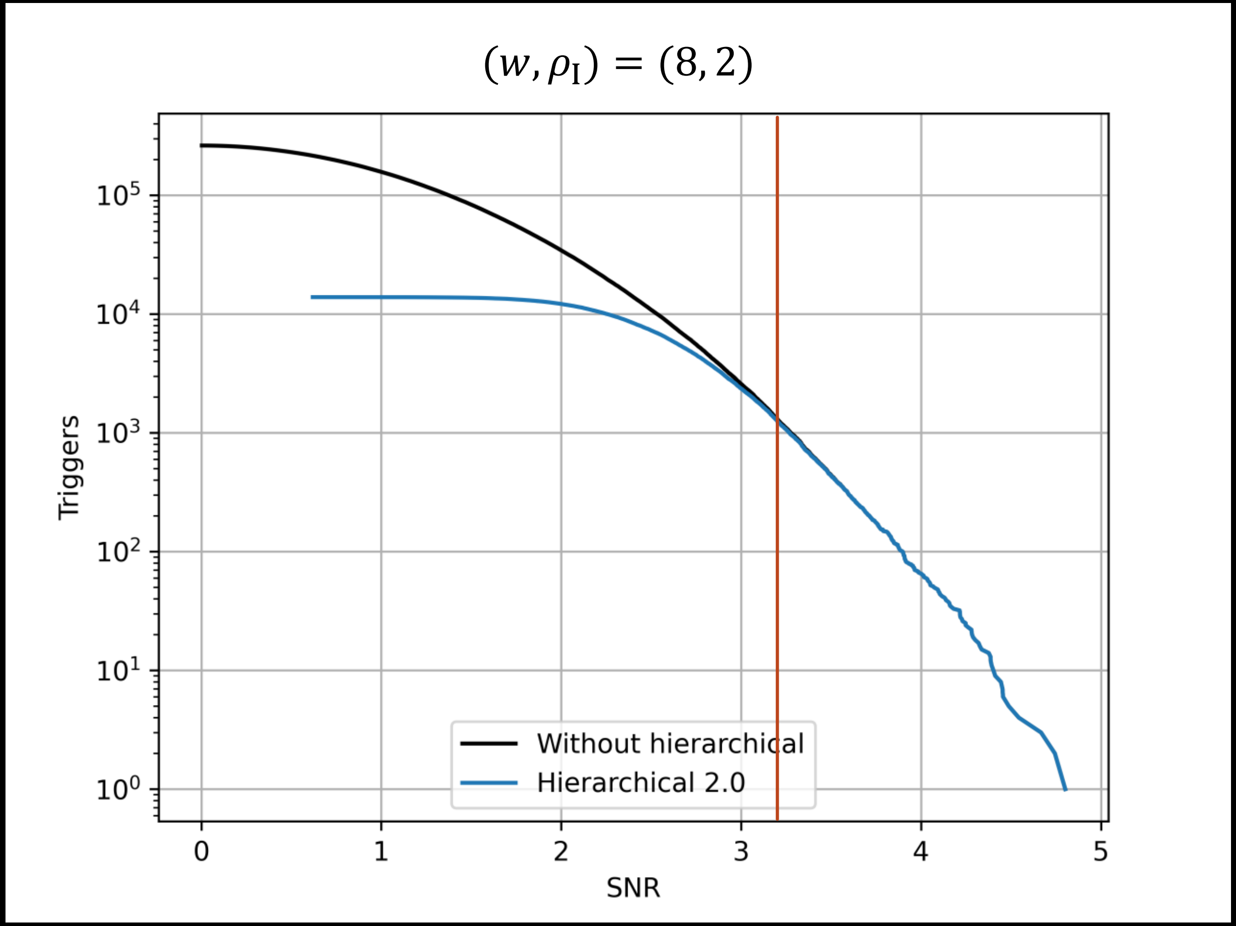
$$\langle \rho_\zeta(t) \rangle_b = \frac{1}{Nw} \sum_{r=0}^{w-1} 4\Delta f \sum_{f=0}^{N-1} \frac{\tilde{s}[f]\tilde{h}_\zeta^*[f]}{S_n[f]} e^{2\pi i f (wb+r)/N}$$

Bin containing w samples

$$\begin{aligned}\langle \rho_\zeta(t) \rangle_b &= \frac{4w\Delta f}{N} \sum_{f'=0}^{N/w-1} e^{2\pi i f' \frac{w}{N} b} \\ &\times \underbrace{\frac{1}{w^2} \sum_{l=0}^{w-1} \frac{\tilde{s}[l\frac{N}{w}+f']\tilde{h}_\zeta^*[l\frac{N}{w}+f']}{S_n[l\frac{N}{w}+f']} \sum_{r=0}^{w-1} e^{2\pi i (l\frac{N}{w}+f')\frac{r}{N}}}_{=\Omega(f')} \\ &= \frac{4w\Delta f}{N} \sum_{f'=0}^{N/w-1} e^{2\pi i f' \frac{w}{N} b}\Omega(f').\end{aligned}$$

- Thus, we need to perform IFFT of a frequency series of reduced length N/w only
- First stage costs are reduced by a factor of w ⟶ $\mathcal{O}\left(\frac{\mathrm{N}}{\mathrm{w}}\log\left(\frac{N}{w}\right)\right)$

Triggering criteria



No. of triggers

$n_{flat}(\rho_{\mathrm{II}})$ flat scheme

$n_{final}(\rho_{\mathrm{II}})$ hierarchical scheme

Two free parameters

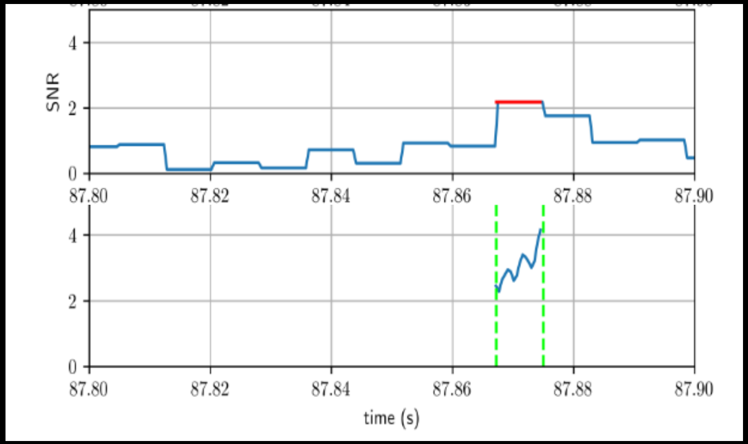
w, ρ_{I}

No loss in sensitivity

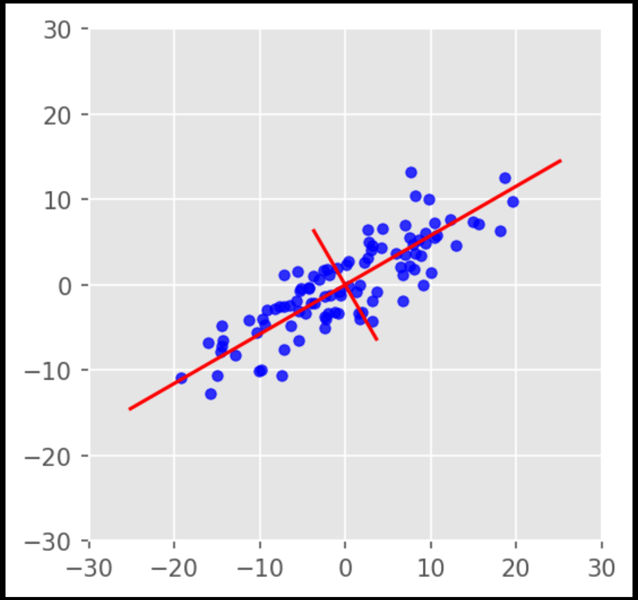
$$\rho_{target}(w, \rho_{\mathrm{I}}) = \left(\min(\rho_{\mathrm{II}}) \middle| n_{final}(\rho_{\mathrm{II}}) \geq 0.99 n_{flat}(\rho_{\mathrm{II}}) \right)$$

Second stage and cost estimation

- Follow up triggering bins
- Reconstruct sample points using the basis



Principal component analysis (PCA)



$$z_{flat} = NT(5\log N + 6)$$

Baseline costs (FLOP)

$$z_{total}(w,\rho_I) = NT\left(\frac{5}{w}\log\left(\frac{N}{w}\right) + 6 + \frac{2}{w}\right) + 4pwf(w,\rho_I) + Np(\log N + 6)$$

Hierarchical costs (FLOP)

Implementation

Codebase in C language and operations on the GPU are performed using CUDA by Nvidia

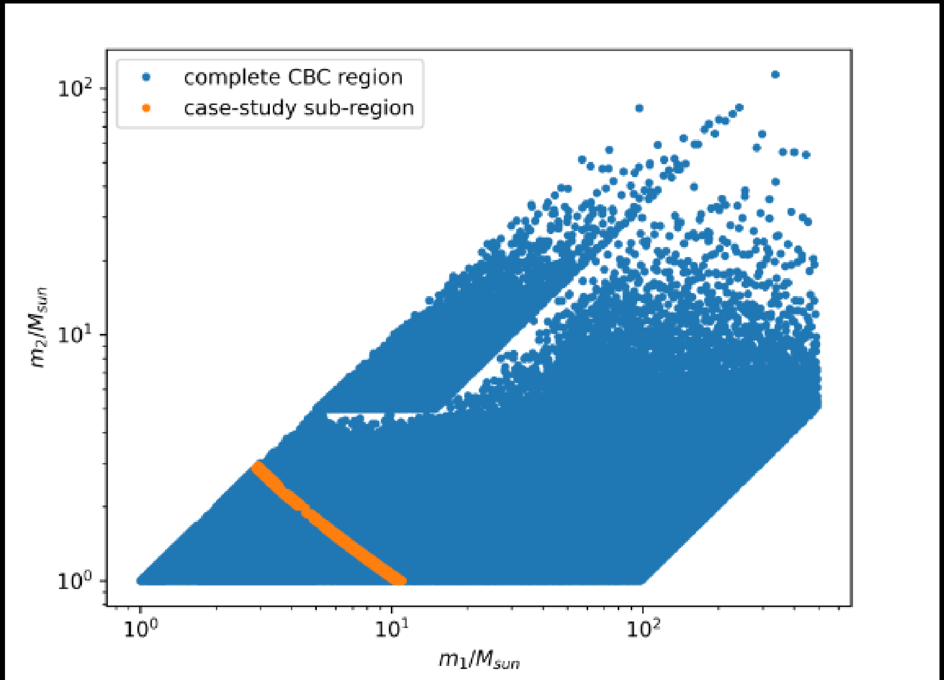
- Pre-compute the basis and store them on hard-drives
- Data is divided into smaller segments of 128s and sampled at 2048 Hz
- Highly parallelizable operations – Matrix multiplications, FFTs
- Optimized libraries -- cuBLAS, cuFFT

ATLAS computing center at AEI

Nvidia Tesla V100 and RTX 2070

Results

Case-study



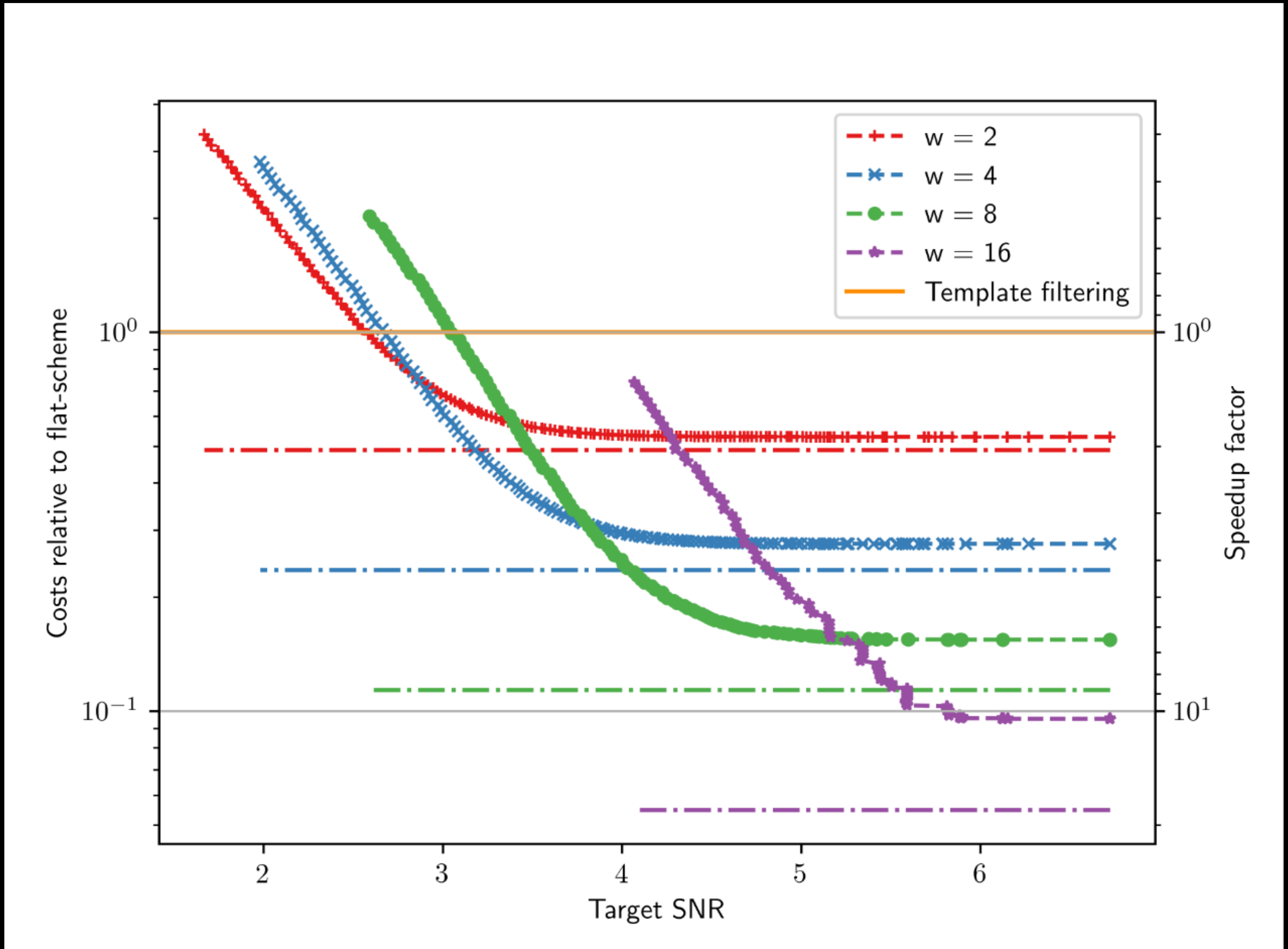
$M \in [5.72, 12.05]$
$q \in [1.0, 11.05]$

6250 templates

- Case-study performed on a sub-region
- Conservative reduction in costs $p = 254 \sim \langle p \rangle$
- We target for SNR 5 and above

- Primarily tested on simulated data containing only Gaussian noise
- Also tested on a small population of BBH signals
- Data generated using PyCBC

Total Costs



Improvement of

10x

for SNR = 6

5x

for SNR = 5

Observed performance

- Improvement using GPUs
- Evaluate performance using throughput of any method

$$\text{Throughput} = \left(\frac{\text{secs of data filtered}}{\text{time taken for filtering}}\right)(\text{No. of templates filtered}) = NT/t$$

Method	Throughput	Throughput/ Euro	Throughput/ W
cuFFT(in-situ)	4000 x 10^3	400	14 x 10^3
Hierarchical scheme (expected)	3300 x 10^4	3300	116 x 10^3
PyCBC live	6300	17	31
PyCBC offline	12,000	32	60

in real-time

Conclusions and future prospects

- Demonstrated our new hierarchical scheme using simulated data.
- Achieved an improvement of 10x and 5x respectively for SNR = 6 and 5 respectively (without losing sensitivity)
- Cost and energy efficient way of performing matched filtering using GPUs.

What's next

- Implement the scheme in PyCBC before O4.
- Use it to perform precessing/eccentric or sub-solar searches

Thank you for your attention

References

1. https://www.nature.com/articles/548397a
2. https://towardsdatascience.com/visualizing-principal-component-analysis-with-matrix-transforms-d17dabc8230e
3. Kipp Cannon, et. al "TOWARD EARLY-WARNING DETECTION OF GRAVITATIONAL WAVES FROM COMPACT BINARY COALESCENCE," The Astrophysical Journal 748, 136 (2012)

Backup slides

Reduced basis matched filtering

- Consider T templates $\tilde{h}_\zeta$ and a basis $\tilde{p}_k$

- Any template can be represented as a linear combination of the basis

$$\tilde{h}_\zeta(f) = \sum_{k=0}^{p_t-1} c_{k,\zeta}\tilde{p}_k$$

complete representation

$$p_t = T$$

- Truncating the basis for an approximate representation

$$\left\langle \frac{\delta\rho}{\rho} \right\rangle = 1 - \frac{\left| \sum_{k=0}^{p-1} \sigma_k^2 \right|}{\left| \sum_{k=0}^{p_t-1} \sigma_k^2 \right|}$$

$$p < T$$

- Matched filter in terms of reduced basis

$$\rho_r(t) = \sum_{k=0}^{p-1} 4 \int_0^\infty \frac{\tilde{s}(f) c_{k,\zeta}^* \tilde{p}_k^*(f)}{S_n(f)} e^{2\pi i f t} df$$

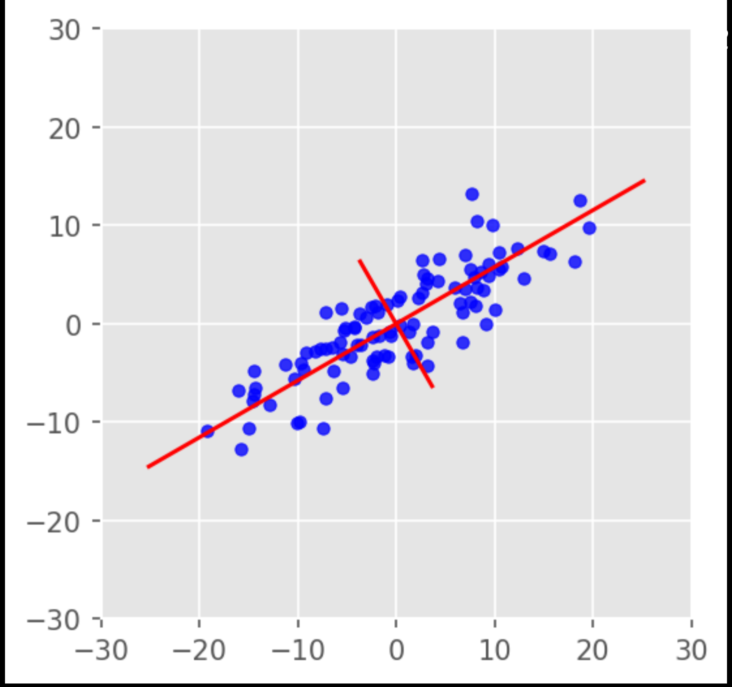
PCA (in a nutshell)

- Obtain the basis → Perform principal component analysis (PCA)

- How to perform PCA ?

1. Consider a set of n data points denoted as vectors $\mathbf{v}$
2. Center the vectors and then normalize them $\mathbf{v}_s = \mathbf{v} - \mathbf{b}$
3. Create a covariance matrix $\mathbf{C} = \hat{\mathbf{v}}_s^{\intercal}\, \hat{\mathbf{v}}_s$
4. Perform an EVD of $\mathbf{C}$ to get the orthonormal basis vectors $\boldsymbol{p}$
5. Decomposition coefficients $\mathbf{D} = \mathbf{p}\hat{\mathbf{v}}_s$
6. Approximated vectors $\hat{\mathbf{v}}_s^{\text{approx}} = \mathbf{D}^{\intercal}\mathbf{p}$

$$\mathbf{b} = {}^{1}/_{n} \sum_{i=0}^{n-1} v_i$$



PCA on a template bank

- Sample waveforms using a non-uniform frequency list (saves up lot of space and computation time)

- Templates are whitened and normalized according to aLIGO ZDHP PSD.

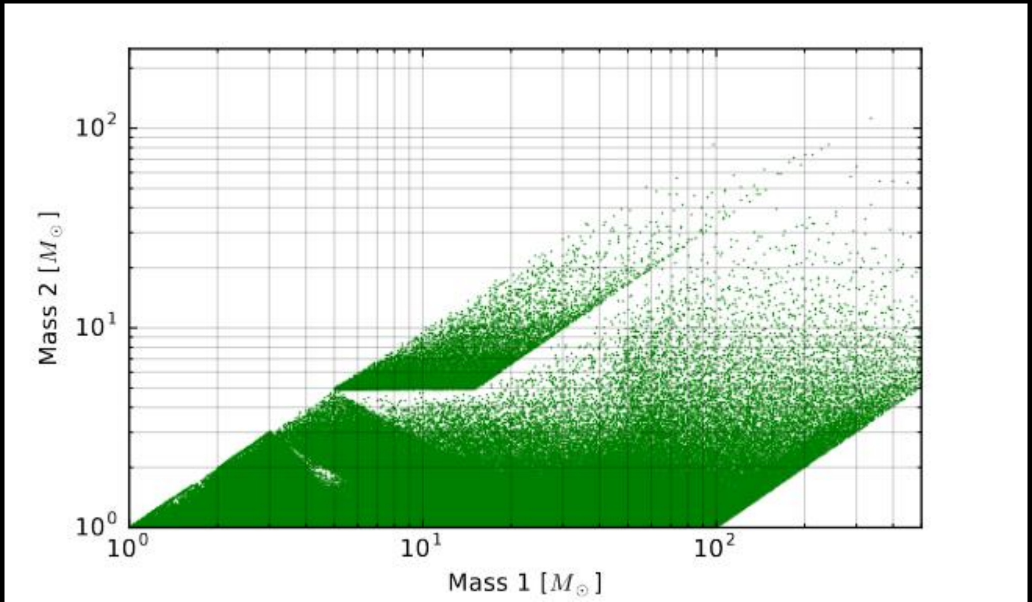
$$\begin{pmatrix} \tilde{h}_0(f_0) & \tilde{h}_0(f_1) & \cdots & \tilde{h}_0(f_{N_t-1}) \\ \tilde{h}_1(f_0) & \tilde{h}_1(f_1) & \cdots & \tilde{h}_1(f_{N_t-1}) \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{h}_{T/64-1}(f_0) & \tilde{h}_{T/64-1}(f_1) & \cdots & \tilde{h}_{T/64-1}(f_{N_t-1}) \end{pmatrix}$$

- Covariance matrix for each sub-bank $\mathbf{C}^m = \mathbf{T}^m \times (\mathbf{T}^m)^\intercal$

- Diagonalisation using *Lanczos algorithm*

- Obtain decomposition matrix $\mathbf{D}^m$ and basis matrix $\mathbf{P}^m$

Aligned spin



~ 400,000 templates

MM = 0.97

sub-bank index	parameter τ_0	p
1	[0.1, 5.1]	64
.	.	.
.	.	.
34(case study)	[98.0, 103.4]	254
.	.	.
.	.	.
64	[442.5, 595.7]	200

6250 templates in each sub-bank

Hierarchical Matched filtering

$$\rho_r(t) = \sum_{k=0}^{p-1} 4 \int_0^{\infty} \frac{\tilde{s}(f) c^*_{k,\zeta} \tilde{p}^*_k(f)}{S_n(f)} e^{2\pi i f t} df$$

Primary idea – Split the reconstruction stage

- Compute forward FFT of $s(t)$ at a uniform sampling rate $1/dt$
- Linearly interpolate $\tilde{p}(f)$ at the uniform frequencies
- Filter data with every basis vector $\tilde{p}_k$ to obtain β_k
- Average β_k in fixed bins of w samples to get β_k^{avg}
- Perform first stage reconstruction to obtain averaged SNR time-series
- Perform second stage reconstruction around the triggers from the first stage.

$$\beta_k^{avg}[t_i] = \sum_{j=0}^{w-1} \beta_k[t_{i\times w+j}]/w$$

Second stage filtering

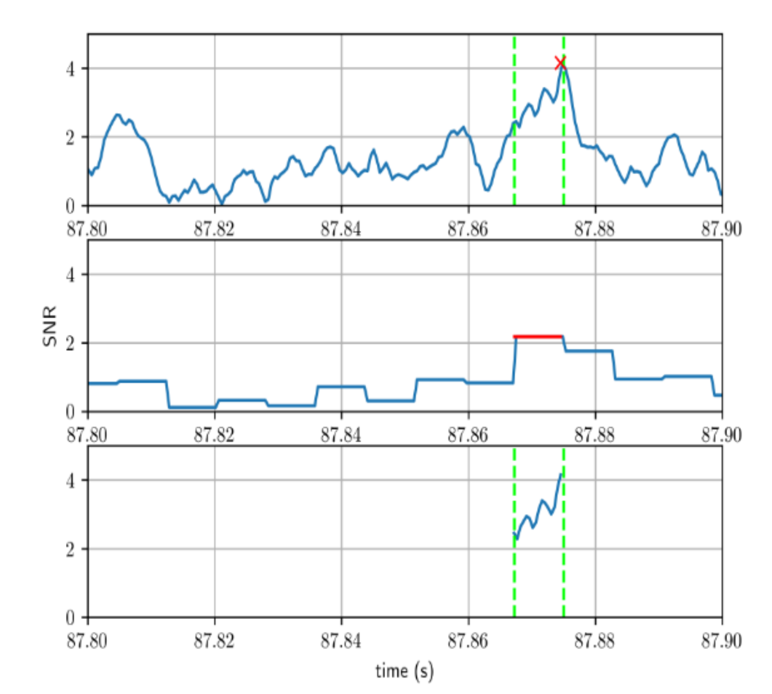
- Requires the basis vectors
- Collect the first stage triggers using a threshold to then perform a finer reconstruction of the triggering bins

$$\rho_r(t) = \sum_{k=0}^{p-1} c^*_{k,\zeta}\beta_k$$

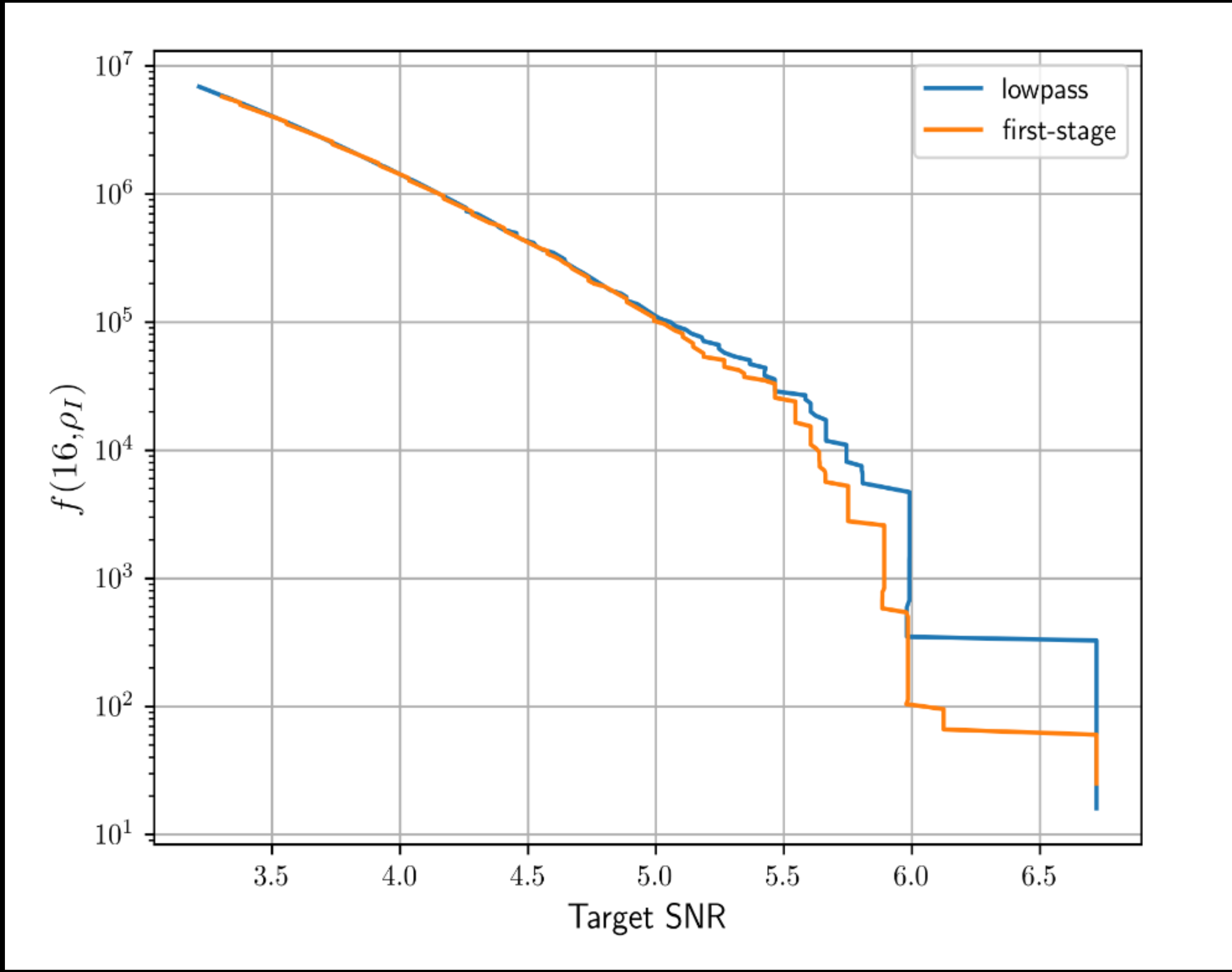
Partial reconstruction

- Triggering criteria for the first stage ?

Two parameters -- w and ρ_{I}



Low-pass filter as first stage



Cost Estimation

- We compute the theoretical FLOP for filtering a data segment with N samples
- Consider 6 operations for multiplication and 2 operations for addition
- For FFTs we consider a split-radix method

Baseline comparison – Template method

1. Forward real-to-half-complex FFT of data → $3/2\ N\log N$
2. Integrand for T templates → $6NT$
3. Inverse complex-to-complex FFT to obtain the SNR time-series → $5NT\ \log N$

Since $T \gg 1$

$$z_{basic} = NT(5\ \log N\ + 6)$$

(Continued) Cost Estimation

Fast first stage

1. Forward FFT and integrand computation — $3/2\, N \log N + 6NT$
2. Binned averaging — $\frac{2NT}{w}$
3. IFFT to get averaged SNR time-series — $\frac{5NT}{w}\log(\mathrm{N/w})$

Second stage

Assuming number of triggers $f(w, \rho_{\mathrm{I}})$ do not vary with template

1. Compute the β's — $N\mathrm{p}(\log N\; + 6)$
2. Second stage reconstruction — $4pwf(w, \rho_{\mathrm{I}})$

$$z_{total} = NT\left(\frac{5}{w}\log\left(\frac{\mathrm{N}}{\mathrm{w}}\right) + 6 + \frac{2}{w}\right) + 4pwf(w, \rho_{\mathrm{I}}) + N\mathrm{p}(\log N\; + 6)$$

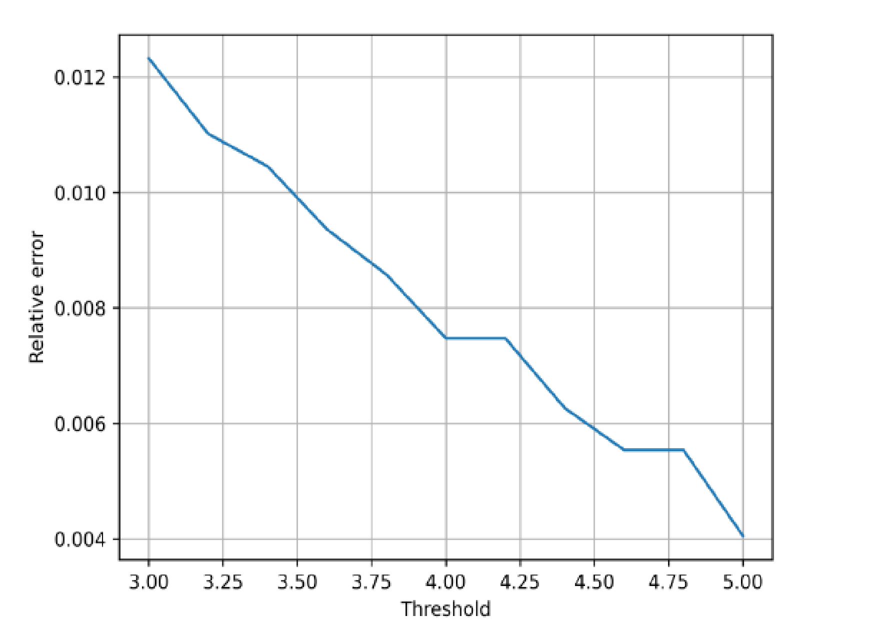
Fine-tune the parameters to minimize the total costs

Accuracy of the SNR

Two primary contributions to the loss in SNR

Truncation of the number of eigenvalues

Interpolation of the basis/templates

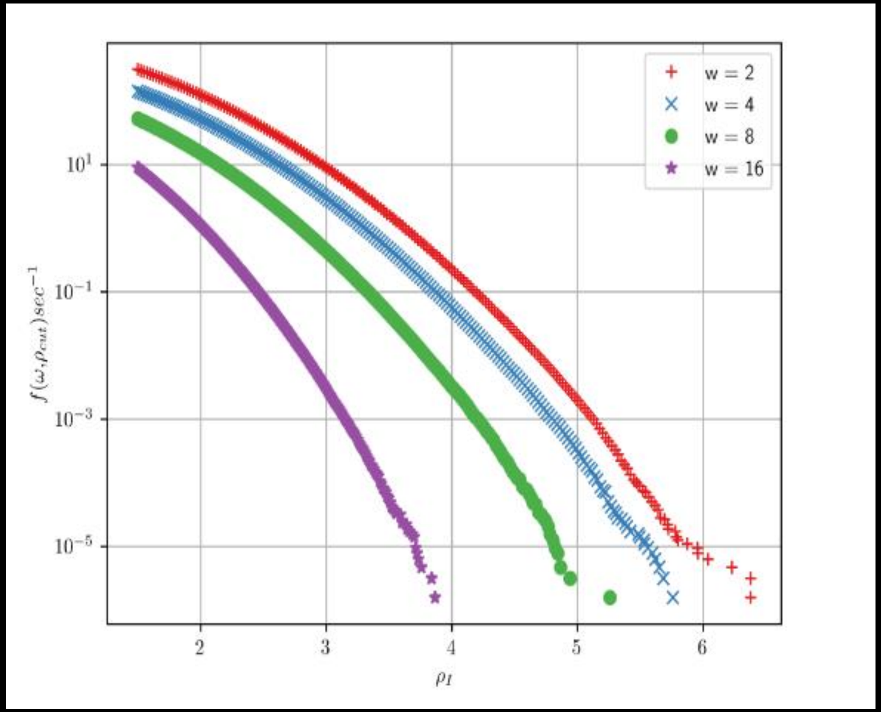


Comparing performance

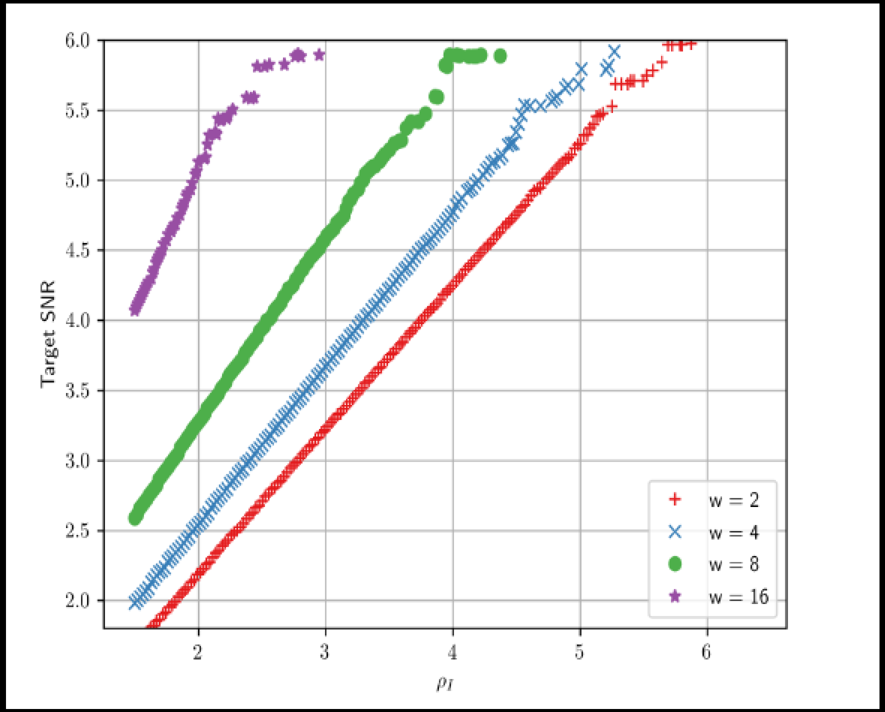
Number of required operations

$$z_{total} = NT\left(\frac{5}{w}\log\left(\frac{\mathrm{N}}{\mathrm{w}}\right) + 6 + \frac{2}{w}\right) + 4pwf(w,\rho_{\mathrm{I}}) + N\mathrm{p}(\log N + 6)$$

- First obtain the $f(w,\rho_{\mathrm{I}})$



- Get the costs in terms of the target SNR



Implementation

Preparation stage
PCA

- Partially implemented on CPUs
- Matrix multiplications – Covariance computation, Decomposition coefficients using cuBLAS
- Diagonalization on CPUs using Lanczos algorithm
- Results are compressed and stored on hard-drive

Matched filtering

- Completely implemented on GPUs
- Data is divided into smaller segments of 128s and sampled at 2048 Hz
- FFTs are performed using cuFFT in parallel batches
- First stage is implemented using basis with help of cuBLAS
- Second stage is using customized kernel

Fast First Stage Filtering using Templates

- Average the matched filter before computing the IFFT

- Consider a single bin b of size w samples and we compute the averaged SNR for that bin

$$\langle \rho_\zeta(t) \rangle_b = \frac{1}{Nw} \sum_{r=0}^{w-1} 4\Delta f \sum_{f=0}^{N-1} \frac{\tilde{s}[f]\tilde{h}^*_\zeta[f]}{S_n[f]} e^{2\pi i f(wb+r)/N}$$

$$t = wb + r$$
$$r \in [0, w-1]$$

- Split the summation into a double sum

$$\sum_{f=0}^{N-1} \tilde{g}[f] = \sum_{f'=0}^{N/w-1} \sum_{l=0}^{w-1} \tilde{g}[l\frac{n}{w} + f'],$$