

Scalable early-warning for binary neutron-star mergers in third-generation gravitational-wave detectors

Martin Gerini (martin.gerini@uclouvain.be)
Davide Gerosa (davide.gerosa@unimib.it)
Rodrigo Tenorio (rodrigo.tenorio.marquez@gmail.com)



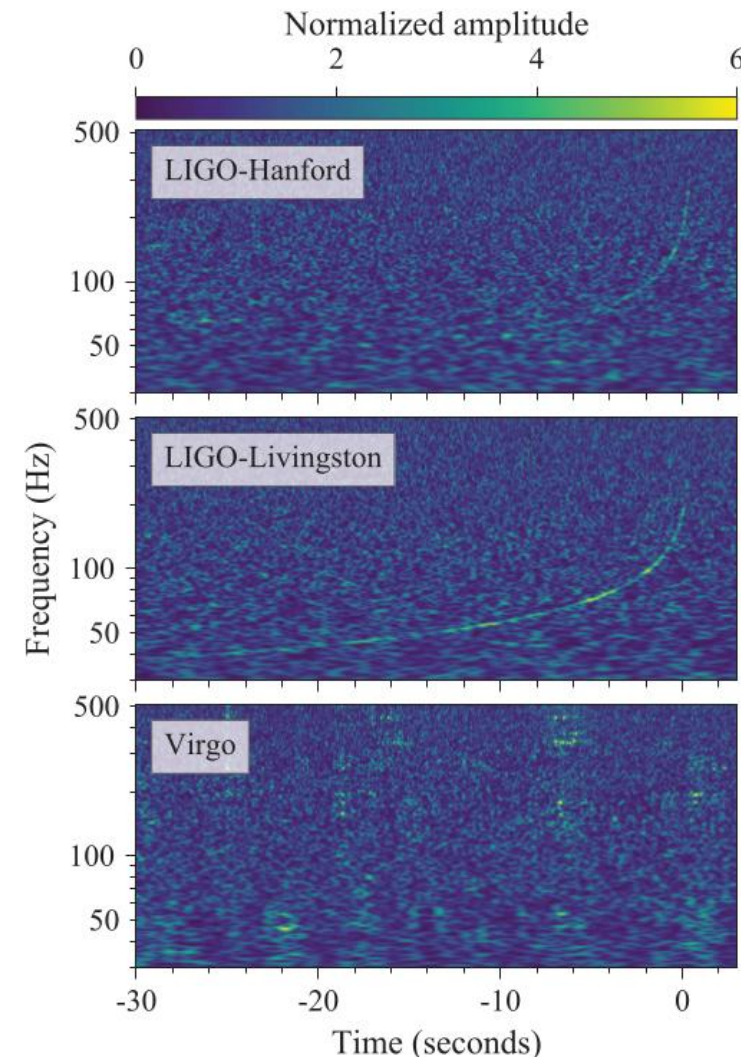
Introduction

- GWs are ripples in the fabric of spacetime produced by accelerating masses (e.g, binary neutron star coalescence **BNS**)
- Routinely detected** by ground-based LIGO-Virgo-KAGRA (LVK) interferometers network
- The observable **duration** is dictated by the interferometers' **minimum frequency limit $f_{\min}$**

Current (LVK): $f_{\min} \gtrsim 20 \text{ Hz}$ $\rightarrow T_{\text{sig}} \approx 30 - 60 \text{ s}$

Einstein Telescope (ET): $f_{\min} \approx \text{few Hz}$ $\rightarrow T_{\text{sig}} \approx \text{several hours}$
Cosmic Explorer (CE)

What configuration? Triangle? Double L-shaped?
What minimum frequency? 2, 3, 4, 5 Hz?

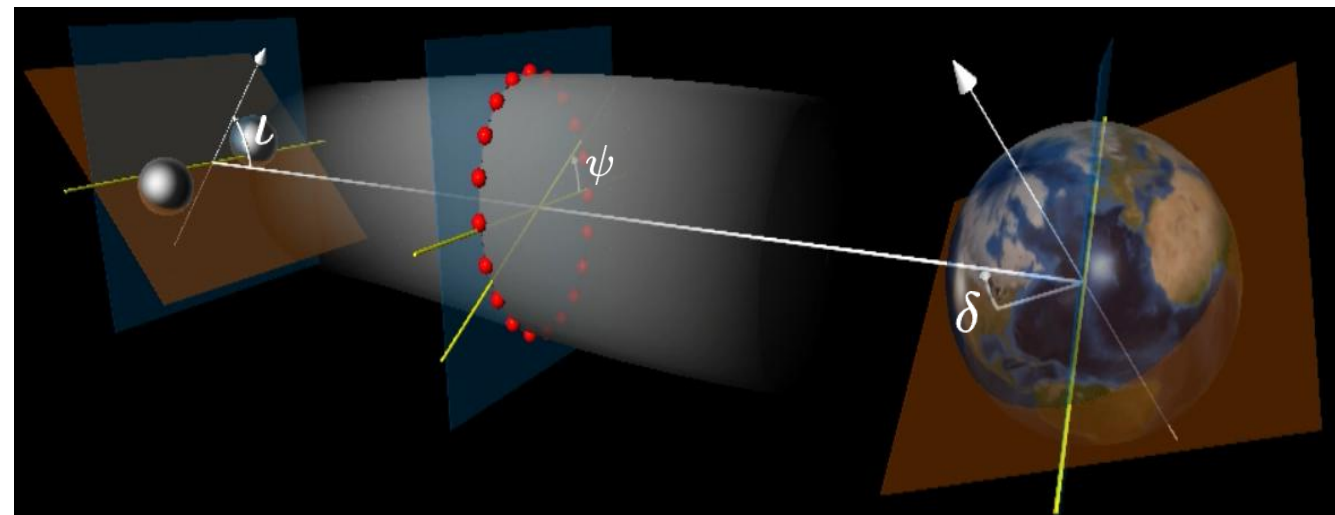


<https://doi.org/10.1103/PhysRevLett.119.161101>

Signal Model

$$\vec{\theta} = \{m_1, m_2, \chi_{1L}, \chi_{2L}, \alpha, \delta, \psi, \iota, \phi_0, t_0, d_L\}$$

Masses $\rightarrow m_1, m_2$
 Spins $\rightarrow \chi_{1L}, \chi_{2L}$
 Sky location $\rightarrow \alpha, \delta$
 Polarization $\rightarrow \psi, \iota$
 Inclination $\rightarrow \phi_0$
 t_0 and d_L are also parameters in the set.



Graham Woan - <https://www.glowscript.org/#/user/grahamwoan/folder/Public/program/gw gw>

Røemer delay + Doppler effect

$$t_{\text{SSB}}(\tau) = \tau + \frac{r_{\text{det}}(\tau) \cdot n(\alpha, \delta)}{c}$$

$$f_{\min} = 2 \text{ Hz} \quad f_{\max} = 10 \text{ Hz}$$

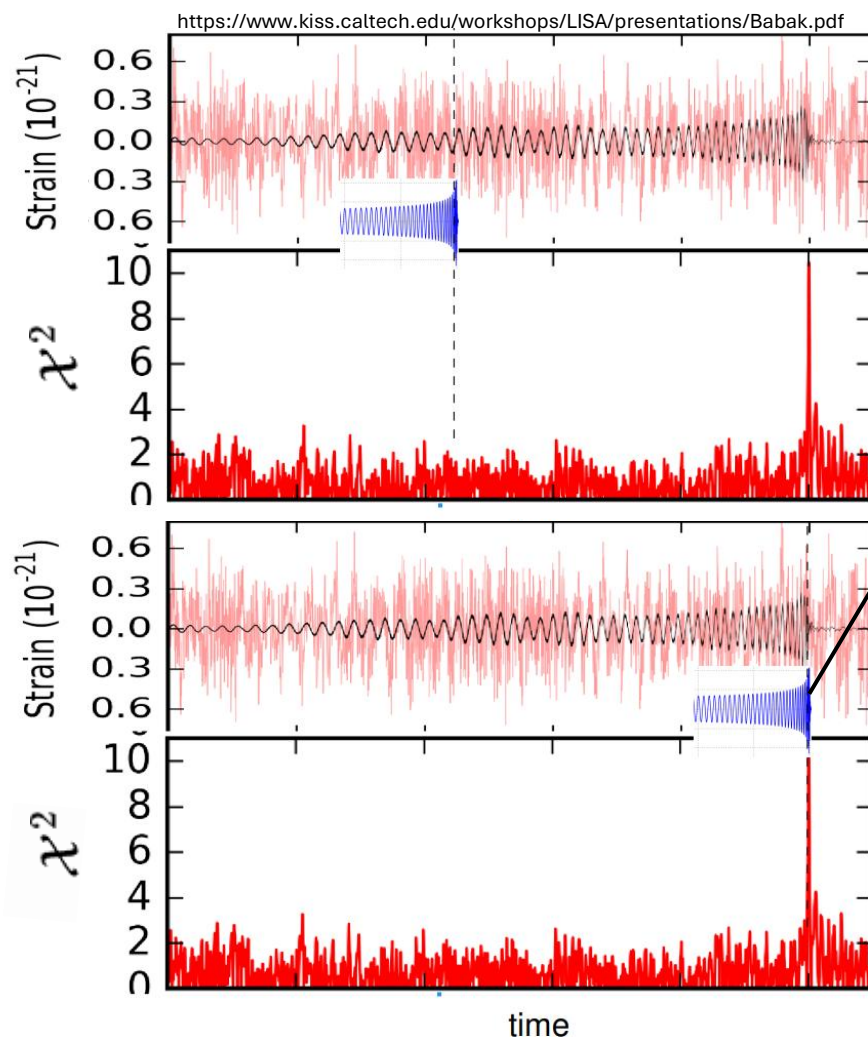
$$\approx 6 \text{ hours}$$

Antenna response

$$h(\tau) = \underbrace{F_+(\tau)}_{\text{Antenna response}} \underbrace{h_+(t_{\text{SSB}}(\tau))}_{\text{GW polarizations}} + \underbrace{F_\times(\tau)}_{\text{Antenna response}} \underbrace{h_\times(t_{\text{SSB}}(\tau))}_{\text{GW polarizations}}$$

+
 Colored Gaussian Noise (ET PSD)

How to detect? Matched Filtering



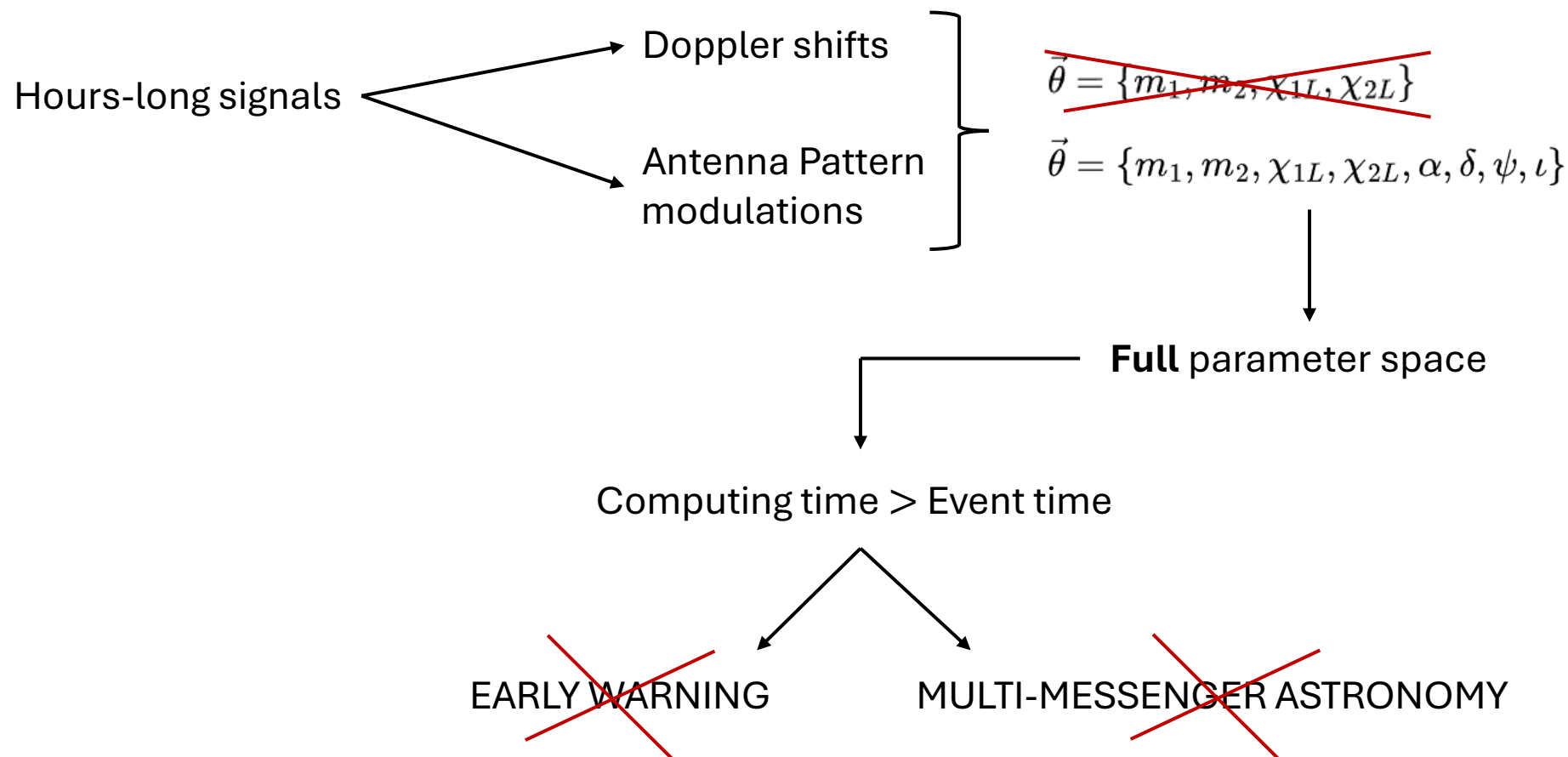
This is the **true waveform** (a **template**)
But in a real application **we don't know**
what the true parameters are!

We need a **TEMPLATE BANK**

For **short signals**,
requires sampling from
the **reduced** parameter
space $\vec{\theta} = \{m_1, m_2, \chi_{1L}, \chi_{2L}\}$

$$\hat{s} = \max_{j \in \{1, \dots, \mathcal{N}\}} \chi^2(\vec{\theta}_j)$$

The problem



SFT-based Matched Filtering

$$\chi^2 = \frac{|\langle d, h \rangle|^2}{\langle h, h \rangle} \quad \langle d, h \rangle = 4 \operatorname{Re} \int_0^\infty \frac{\tilde{d}^*(f) \tilde{h}(f)}{S_n(f)} df$$

← The Matched Filtering statistic... a bit slow

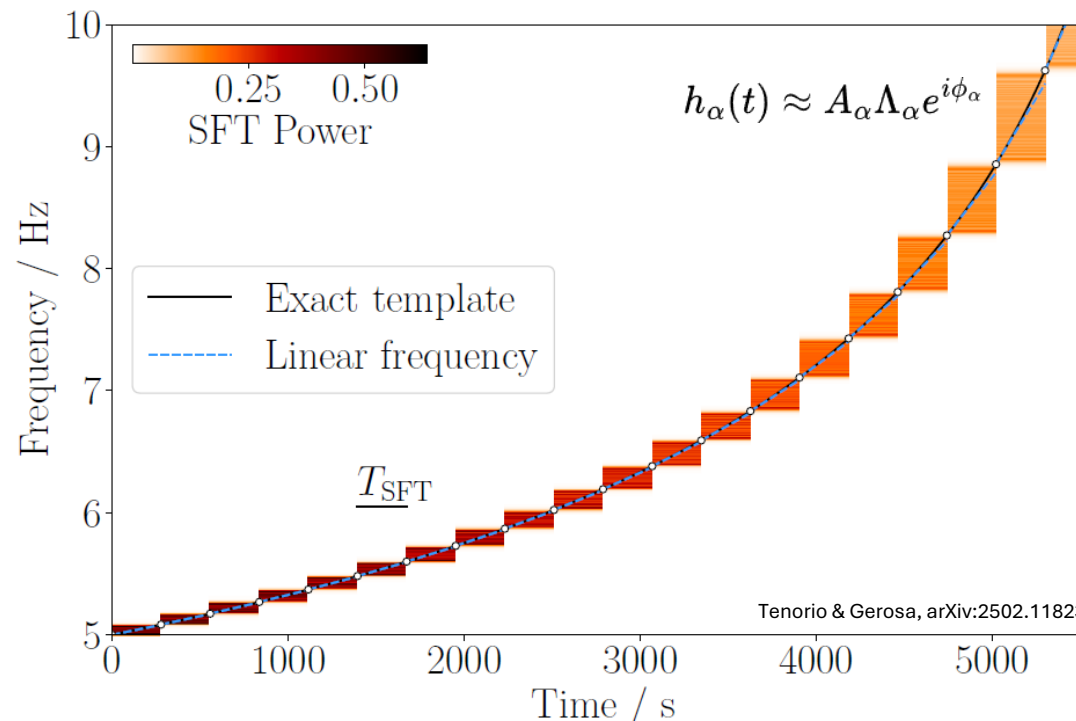
$$([d, h]) \doteq \Delta f \sum_{\alpha=0}^{N_{\text{SFT}}-1} \mathcal{W}_\alpha \cdot h_\alpha^* \cdot C(f_\alpha, \dot{f}_\alpha; \tilde{d}_\alpha)$$

$$\chi^2 = \frac{|\langle d, h \rangle|^2}{\langle h, h \rangle} \approx \frac{([d, h])^2}{([h, h])}$$

$$\hat{s}_{\text{SFT}} = \max_{j \in \{1, \dots, \mathcal{N}\}} \chi^2(\vec{\theta}_j)$$



→ GPU-acceleration



Still requires to sample from the **full parameter space**!

Amortized Inference

$$\mathcal{B}(d) = \frac{p(d|\mathcal{H}_S)}{p(d|\mathcal{H}_N)} = \int \frac{p(d|\theta, \mathcal{H}_S)}{p(d|\mathcal{H}_N)} p(\theta|\mathcal{H}_S) d\theta \quad \leftarrow \text{Computationally intractable at inference}$$

To reduce this computational burden, we propose **marginalizing** over the **extrinsic parameters** using deep learning

$$\mathcal{B}(\lambda; d) = \int \frac{p(d|\lambda, \bar{\lambda}, \mathcal{H}_S)}{p(d|\mathcal{H}_N)} p(\bar{\lambda}|\mathcal{H}_S) d\bar{\lambda} \quad \begin{aligned} \theta &= \{\lambda, \bar{\lambda}\} \\ \lambda &= \{\mathcal{M}_c\} \end{aligned}$$



Binary Cross-Entropy (BCE) loss computes this integral

$$r_\phi(d) \approx \mathcal{B}(\lambda; d)$$

$$\mathcal{B}(d) \approx \max_{\lambda} \mathcal{B}(\lambda; d)$$

CNN – training step

Sample a **batch** of parameters

Generate a **matched pair** of samples

Signal Hypothesis

Crop the SFT data around the *approximated* frequency track (0PN approximation)

$$d(t_\alpha; \mathcal{M}_c) = [\tilde{d}_\alpha[k_\alpha - P], \dots, \tilde{d}_\alpha[k_\alpha + P]] \in \mathbb{C}^{2P+1}$$

↓ Re / Im decomposition

$$\mathbf{D}_{\text{det}}(\mathcal{M}_c) = [\text{Re } d(t_\alpha; \mathcal{M}_c), \text{Im } d(t_\alpha; \mathcal{M}_c)]$$

$$\mathbf{D}(\mathcal{M}_c) = [\mathbf{D}_{\text{det}}(\mathcal{M}_c), \text{det} = 1, \dots, N_{\text{det}}]$$

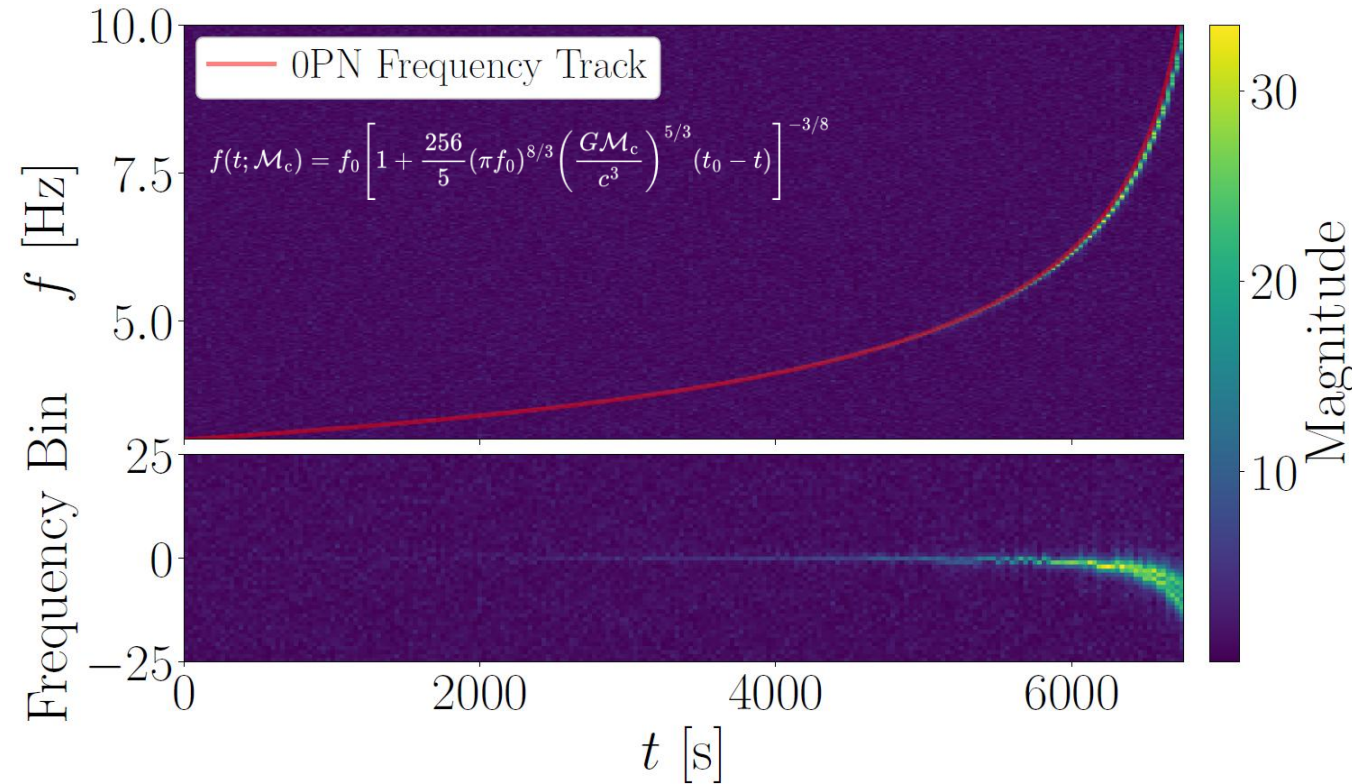
↓ Whitening

$$\mathbf{D}^w(\mathcal{M}_c) \propto \frac{\mathbf{D}(\mathcal{M}_c)}{\sqrt{(S_n)}}$$

Noise Hypothesis

Generate an independent realization of colored noise

Crop the SFT data using the **same frequency track** and **same parameters** of the signal sample

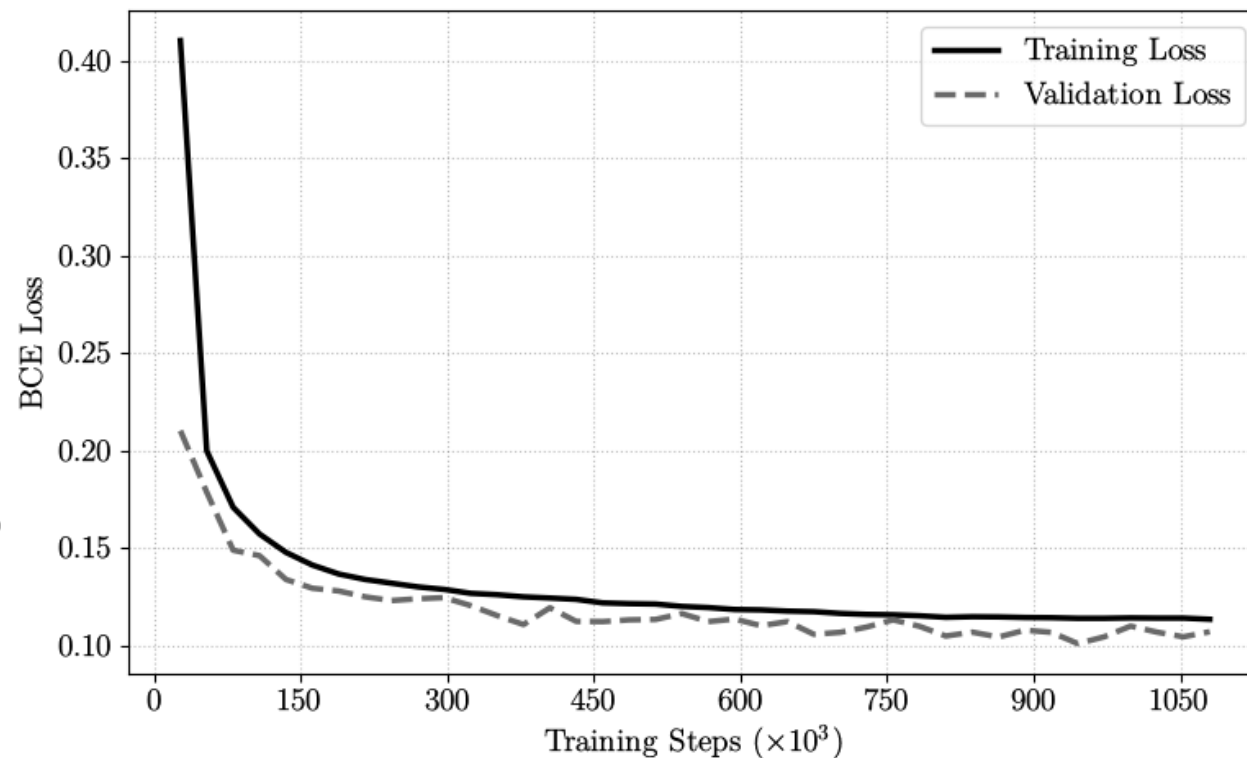


CNN – training

- *On-the-fly* generated dataset
- BCE with logits
- Batch size = 128
- $N_{training\ steps} = 1\ 080\ 000$
- After 27 000 steps $\rightarrow N_{validation\ steps} = 200$
- NVIDIA A100-SXM-64GB



138 240 000 TRAIN SAMPLES
1 024 000 VALIDATION SAMPLES
 $t_{train} \approx 24\ hrs$



Equinox



CNN – execution

→ Template bank generation $\mathcal{T}_{\text{NN}} = \{\mathcal{M}_c^{(l)}\} \quad l = 0, \dots, \mathcal{N} - 1 \quad \longleftarrow \quad \text{1D parameter space!}$

→ For each template $\mathcal{M}_c^{(l)} \in \mathcal{T}_{\text{NN}}$

1. **Track computation** $\rightarrow f(\mathcal{M}_c^{(l)})$
2. **Tensor construction** $\rightarrow \mathbf{D}^{(l)} = \mathbf{D}(f(\mathcal{M}_c^{(l)}))$
3. **Whitening**

↓

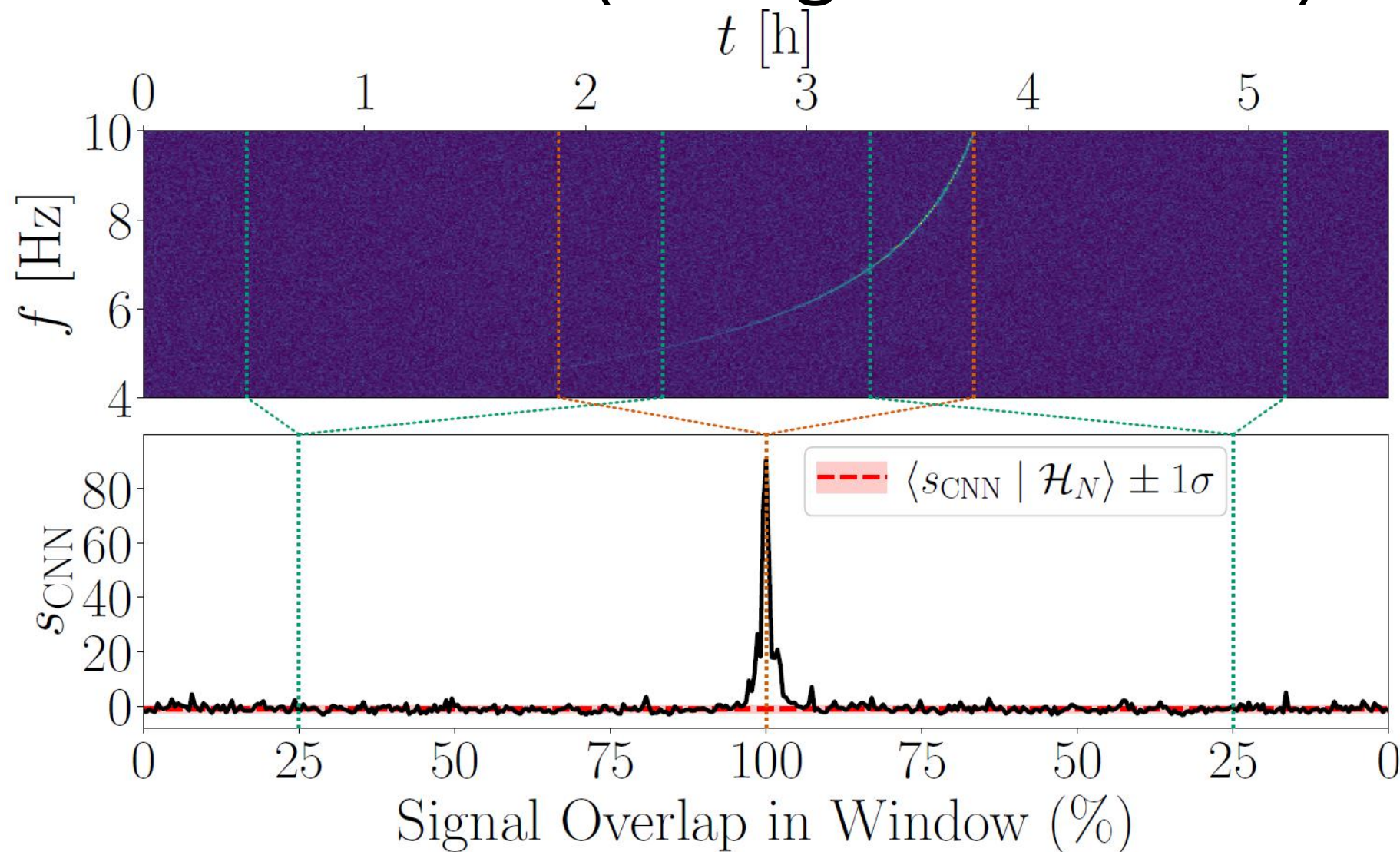
CNN

↓

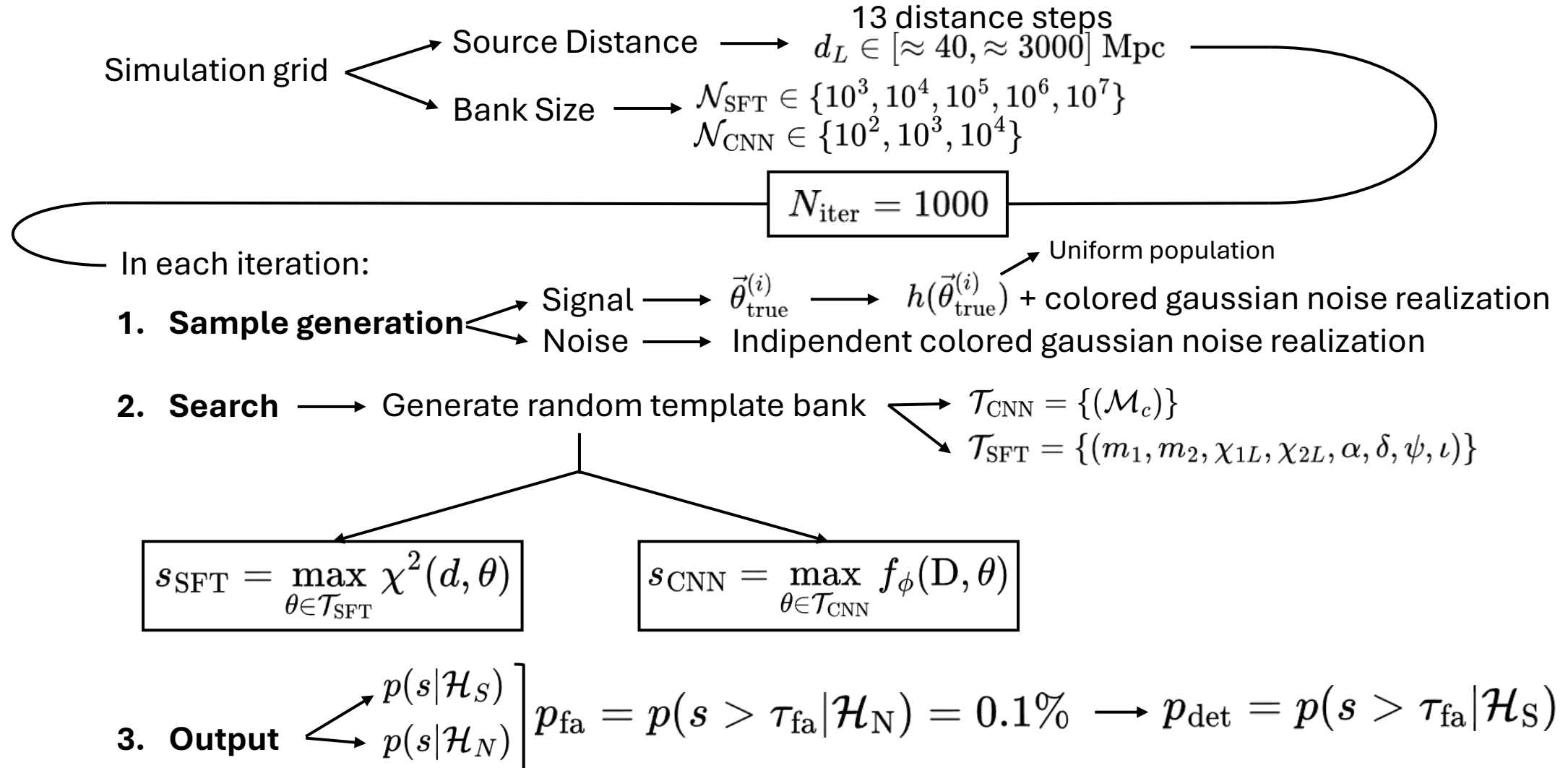
$$r_\phi^{(l)} = f_\phi(\mathbf{D}^{(l)}) \in \mathbb{R}$$

$$\boxed{s_{\text{CNN}} = \max_l r_\phi^{(l)}}$$

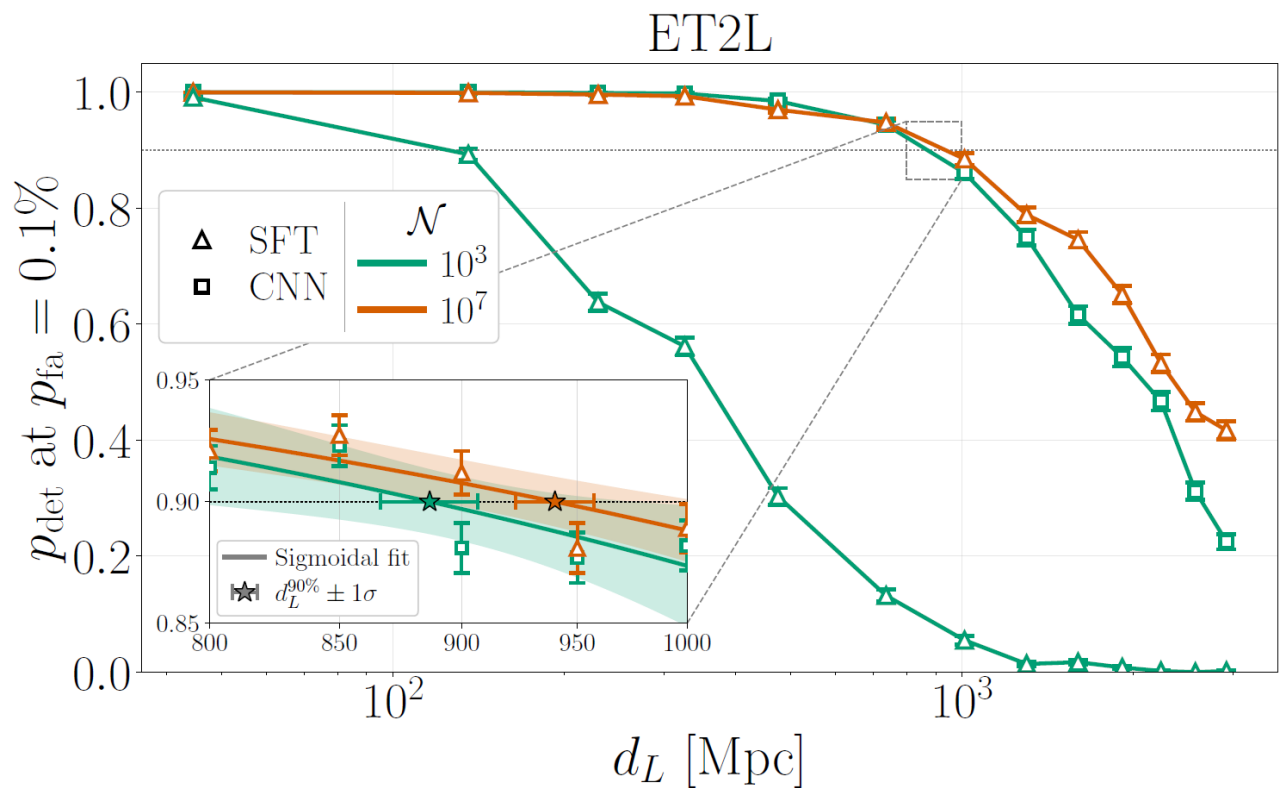
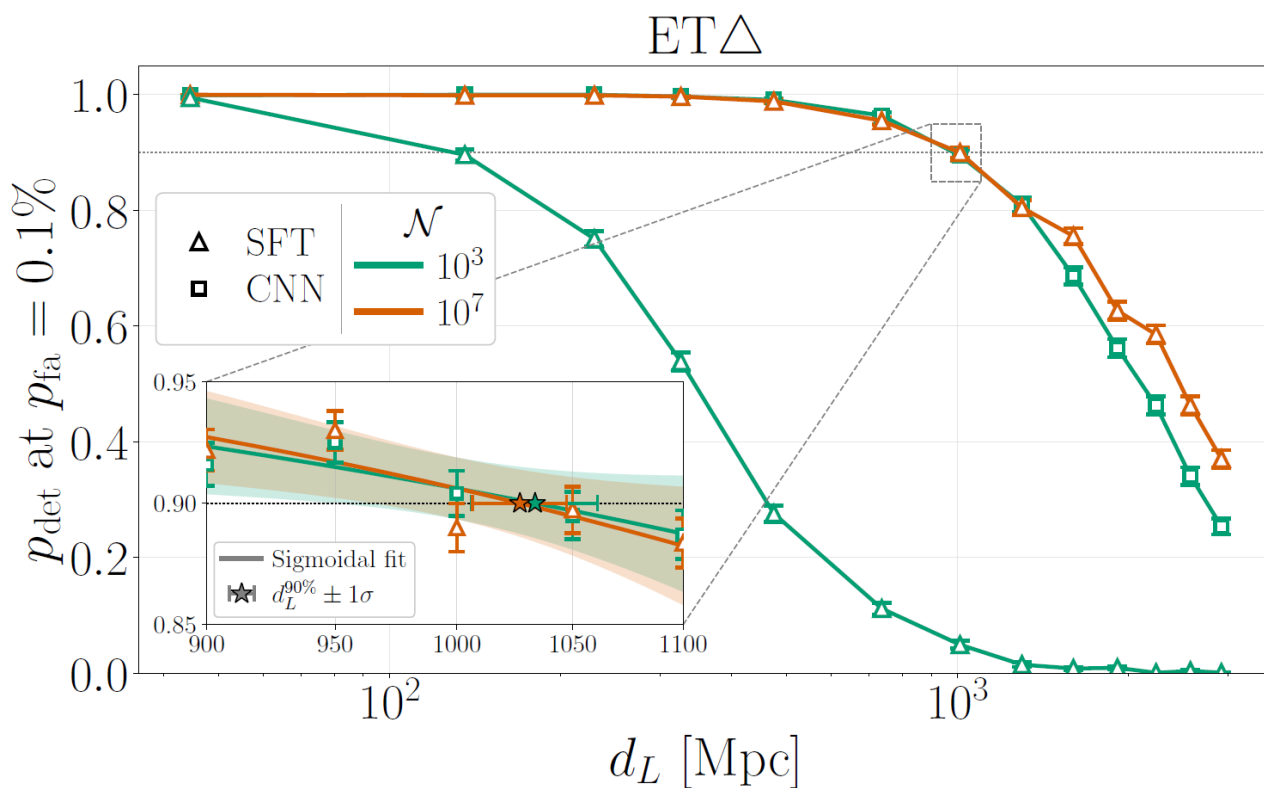
CNN – execution (rolling window test)



Simulation setup



Results ($f_{\min} = 2$ Hz)



Results – Execution time

CNN

$\approx 40\,000$ templates/s ($\triangle$)

$\approx 45\,000$ templates/s (2L)

$$T_{\text{CNN}}^{\triangle} = \frac{1\,000}{40\,000} \text{ s} \approx \mathbf{25 \text{ ms}}$$

$$T_{\text{CNN}}^{2\text{L}} = \frac{1\,000}{45\,000} \text{ s} \approx \mathbf{22 \text{ ms}}$$

SFT Matched Filtering

$\approx 10\,000$ templates/s ($\triangle$)

$\approx 14\,000$ templates/s (2L)

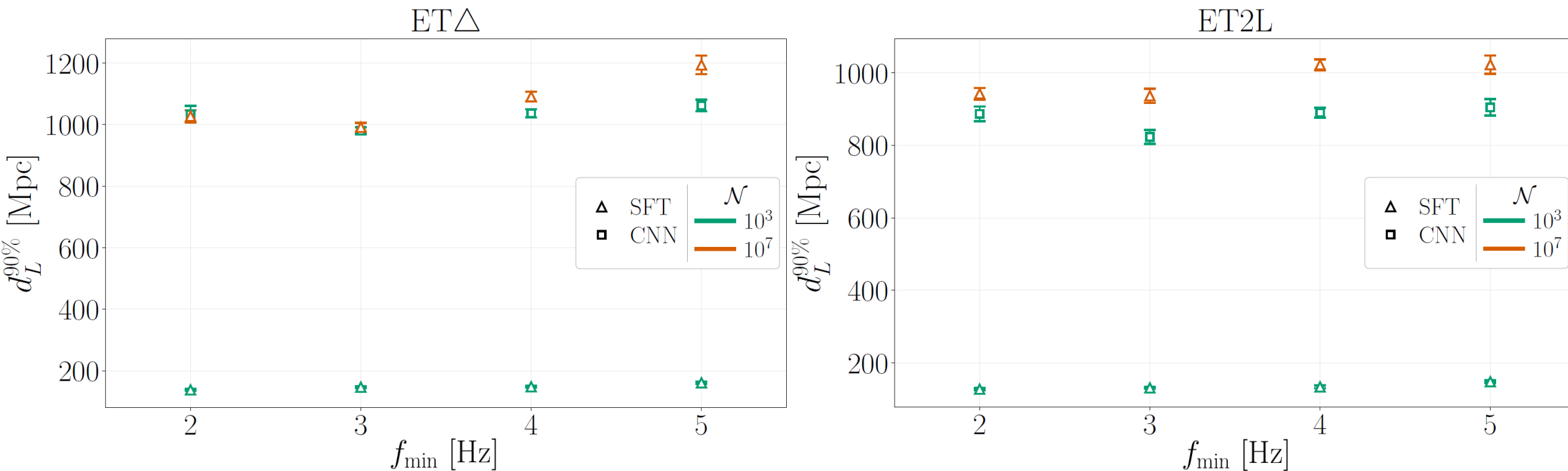
$$T_{\text{MF}}^{\triangle} = \frac{10^7}{10\,000} \text{ s} \approx \mathbf{17 \text{ min}}$$

$$T_{\text{MF}}^{2\text{L}} = \frac{10^7}{14\,000} \text{ s} \approx \mathbf{12 \text{ min}}$$

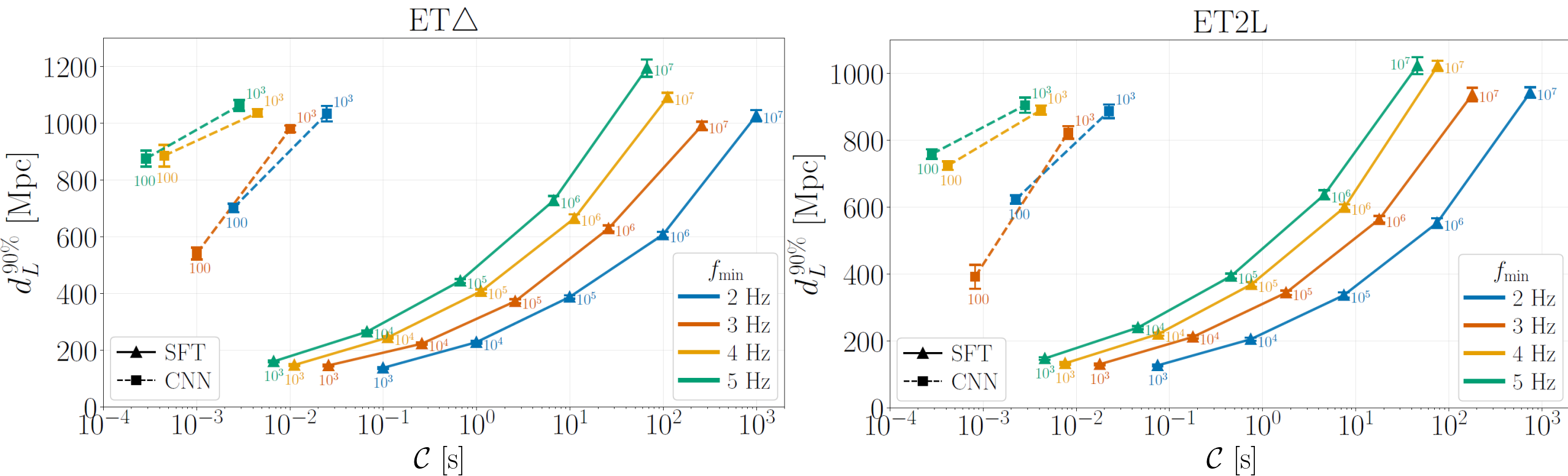
$$\frac{T_{\text{MF}}^{\triangle}}{T_{\text{CNN}}^{\triangle}} \approx 4 \times 10^4$$

$$\frac{T_{\text{MF}}^{2\text{L}}}{T_{\text{CNN}}^{2\text{L}}} \approx 3 \times 10^4$$

Results – different signal lengths



Results – different signal lengths



$$\mathcal{C} = \mathcal{N}\langle \mathbf{c}_1 \rangle$$

Conclusions

GPU-optimized SFT **Matched Filtering** → **Optimal** sensitivity
But **high** computational costs

CNN → Sensitivity **comparable** to SFT Matched Filtering
Efficiency **gain** of over four orders of magnitude

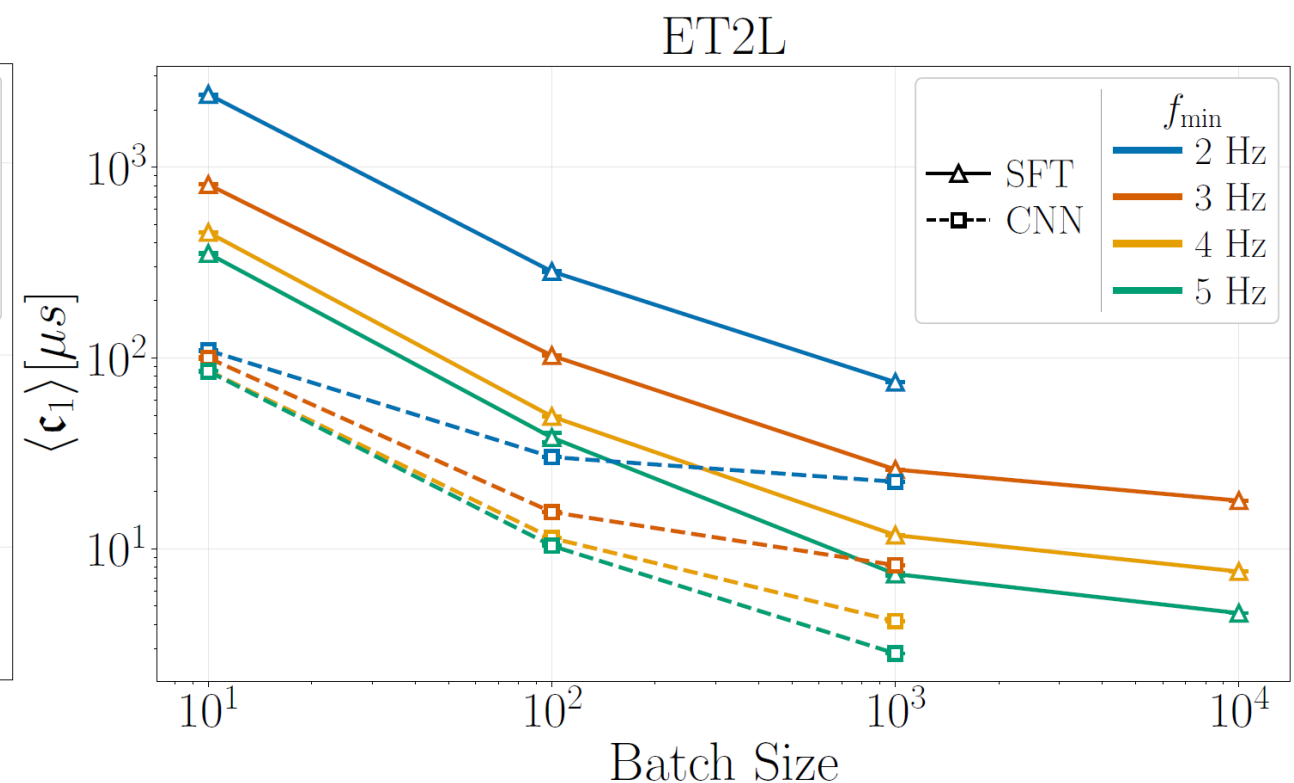
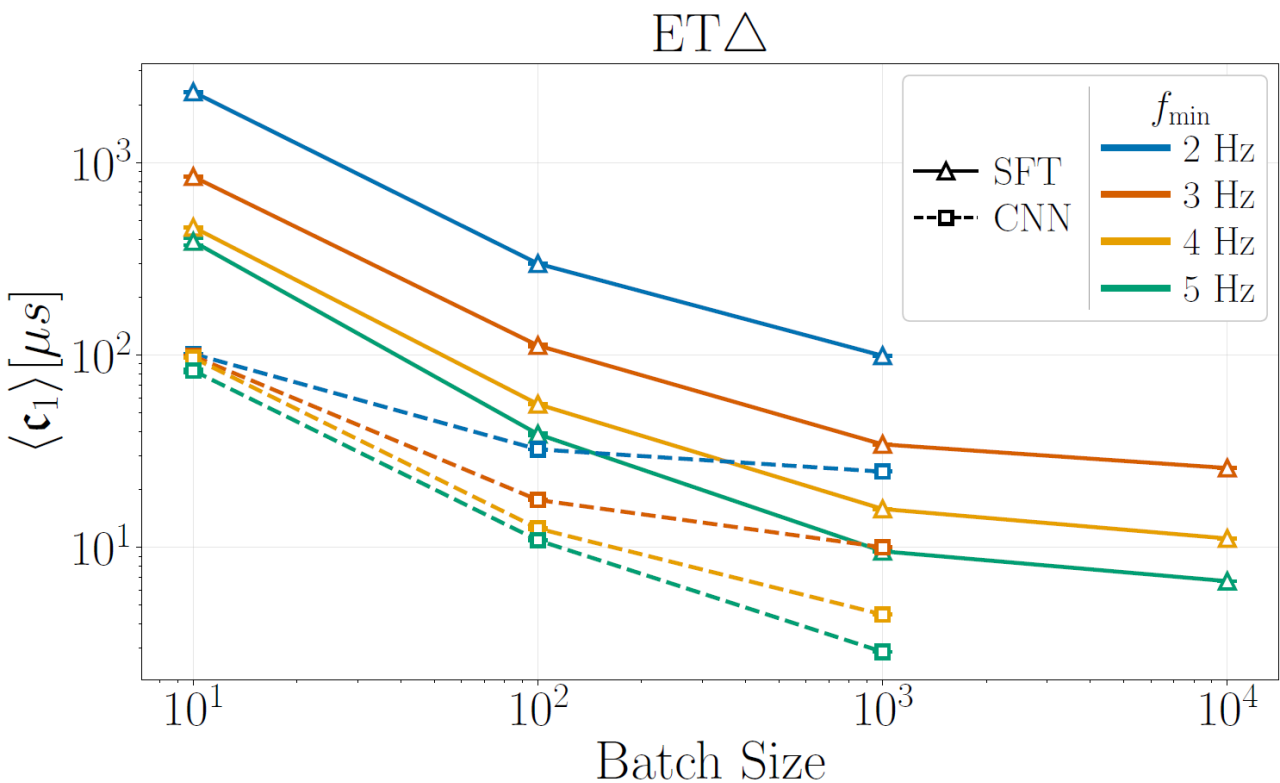
↓
Low-latency triggering of long-duration GW signals

What's next?

- Overlapping signals
- **Skylocation** with flowmatching exploiting the chirpmass estimation of this detection pipeline

Block	Layer / Operation	Kernel	Stride	Output Shape
Input	Input Tensor	-	-	(4/6, 51, 218)
Conv 1	Conv2D	(2, 2)	(1, 2)	(32, 50, 109)
	ReLU	-	-	(32, 50, 109)
Conv 2	Conv2D	(5, 5)	(3, 3)	(64, 16, 35)
	BatchNorm	-	-	(64, 16, 35)
	ReLU	-	-	(64, 16, 35)
Conv 3	Conv2D	(5, 5)	(3, 4)	(128, 4, 8)
	BatchNorm	-	-	(128, 4, 8)
	ReLU	-	-	(128, 4, 8)
Conv 4	Conv2D	(2, 2)	(2, 2)	(256, 2, 4)
	BatchNorm	-	-	(256, 2, 4)
	ReLU	-	-	(256, 2, 4)
Flatten	Ravel	-	-	(2048)
Classifier	Dropout ($p = 0.5$)	-	-	(2048)
	Linear (Dense)	-	-	(128)
	ReLU	-	-	(128)
	Dropout ($p = 0.5$)	-	-	(128)
	Linear (Output)	-	-	(1)

$$\mathcal{C} = \mathcal{N}(\mathbf{c}_1)$$



$$\mathcal{L} = \langle d, h \rangle - \frac{1}{2} \langle h, h \rangle$$

$$\langle a, b \rangle = 4\text{Re} \int_0^\infty df \frac{\tilde{a}^*(f) \tilde{b}(f)}{S_n(f)}$$

$$h(t) = \frac{h_c(t) \cos \phi_0 + h_s(t) \sin \phi_0}{d_L}$$

$$\begin{aligned} h_c(t) &= \mathcal{A}_+(t) \cos \phi(t) + \mathcal{A}_\times(t) \sin \phi(t) \\ h_s(t) &= \mathcal{A}_\times(t) \cos \phi(t) - \mathcal{A}_+(t) \sin \phi(t) \end{aligned}$$

$$\begin{aligned} \mathcal{A}_+(t) &= A(t) \frac{1 + \cos^2 \iota}{2} F_+(t), \\ \mathcal{A}_\times(t) &= A(t) \cos \iota F_\times(t). \end{aligned}$$

$$\mathcal{X} \equiv \max_{\phi_0, d_L} \mathcal{L} = \frac{\langle d, h_c \rangle^2 + \langle d, h_s \rangle^2}{2 \langle h_c, h_c \rangle}$$

$$s_{\text{SFT}} = \max_{j \in \{1, \dots, \mathcal{N}\}} \mathcal{X}(\theta_j)$$

$$([d, h_{c/s}]) = 2 \sum_{\alpha=0}^{N_{\text{SFT}}-1} \Lambda_{c/s, t_\alpha} e^{i\varphi_\alpha} C(f_\alpha, \dot{f}_\alpha; \tilde{d}_\alpha)$$

$$h_{c/s}(t) = \Lambda_{c/s, t} e^{i\phi(t)},$$

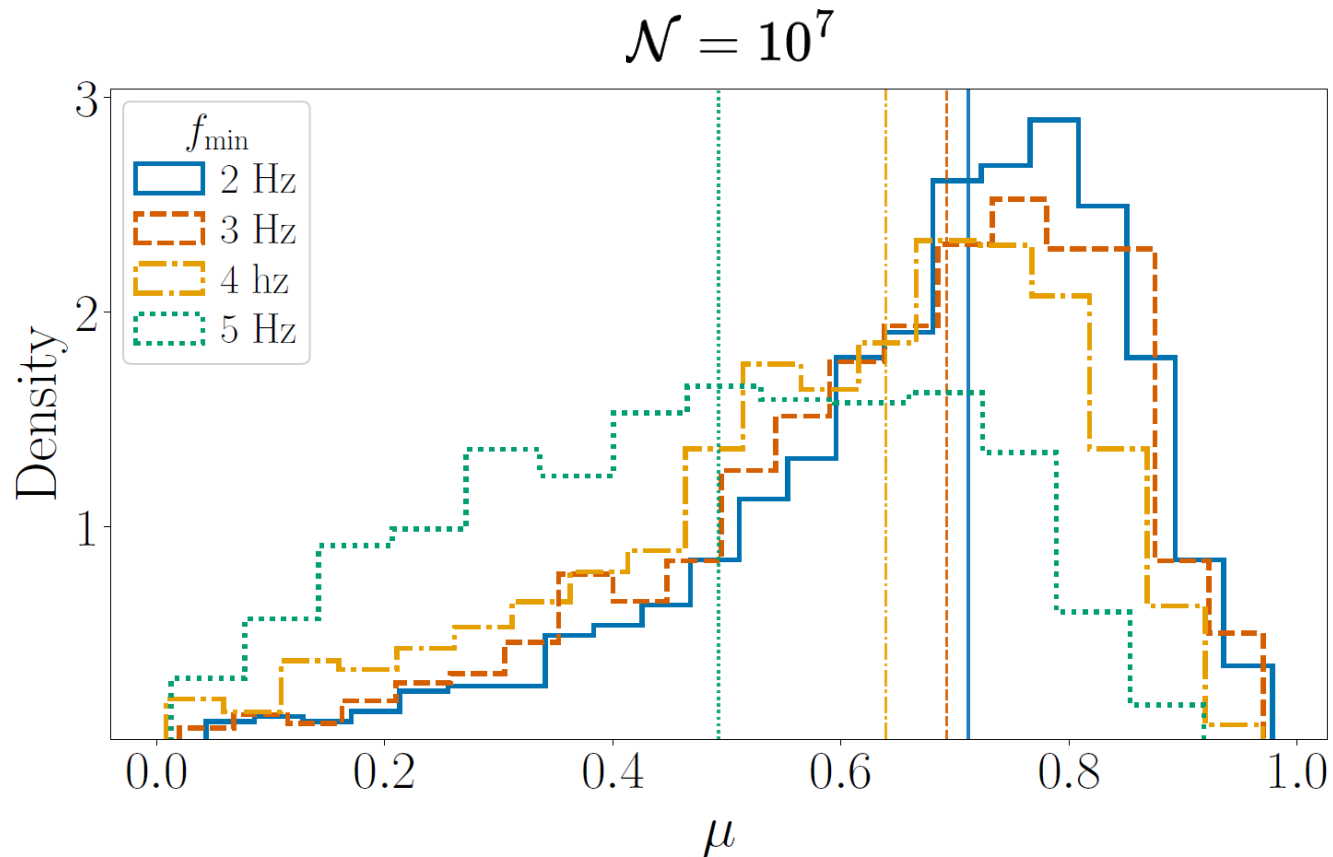
$$\Lambda_{c, t} = \mathcal{A}_+(t) \text{Re} + \mathcal{A}_\times(t) \text{Im},$$

$$\Lambda_{s, t} = \mathcal{A}_\times(t) \text{Re} - \mathcal{A}_+(t) \text{Im}.$$

$$C(f_\alpha, \dot{f}_\alpha; \tilde{d}_\alpha) = \Delta f \sum_{k=k_{\min}}^{k_{\max}} \frac{\tilde{d}_\alpha^*[k]}{S_n(f_k)} \mathfrak{F}\mathfrak{r}\mathfrak{e}(f_\alpha - f_k, \dot{f}_\alpha)$$

$$\mathfrak{F}\mathfrak{r}\mathfrak{e}(f_0, f_1) = \Delta t \sum_{j=0}^{n_{\text{SFT}}-1} e^{i(2\pi f_0 \tau_j + \pi f_1 \tau_j^2)}$$

$$([h_c, h_c]) \approx \sum_{\alpha} [\mathcal{A}_+^2(t_\alpha) + \mathcal{A}_\times^2(t_\alpha)] \left(\Delta f \sum_{k=k_{\min}}^{k_{\max}} \frac{|\mathfrak{F}\mathfrak{r}\mathfrak{e}(f_\alpha - f_k, \dot{f}_\alpha)|^2}{S_n(f_k)} \right)$$



$$\langle s \rangle \sim \rho_{\text{opt}}^2 (1 - \langle \mu \rangle)$$

$$\mu(\theta; \theta_{\text{true}}) = 1 - \frac{s(\theta)}{s(\theta_{\text{true}})}$$

$$\rho_{\text{opt}}^2 \propto \int_{f_{\min}}^{10 \text{ Hz}} df \frac{f^{-7/3}}{S_n(f)}$$

Lowering from 5 Hz to 2 Hz yields an increase in ρ_{opt}^2 of $\sim 7.14\%$

Median mismatch increases from ~ 0.49 to ~ 0.71

The expected detection statistic decreases to $1.0714 \times (1 - 0.71)/(1 - 0.49) \approx 61\%$ of its original value