Simulation-Based-Inference (SBI) (for GWB reconstruction)

Mauro Pieroni
Instituto de Estructura de la Materia (IEM - CSIC)
mauro.pieroni@csic.es

Seminar @ SISSA, Trieste

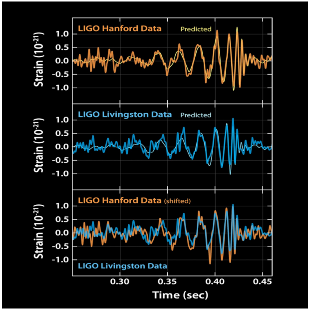
December 18th, 2026

Outline

The dawn of GW astronomy

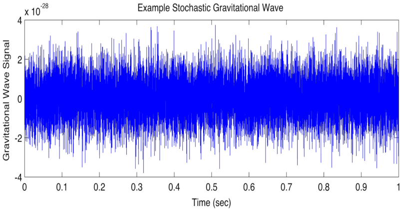
Gravitational Waves (GWs) are:

- Spacetime perturbations
- Almost free streaming
- The ultimate cosmological probe



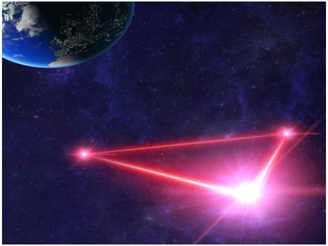
There are two classes of signals:

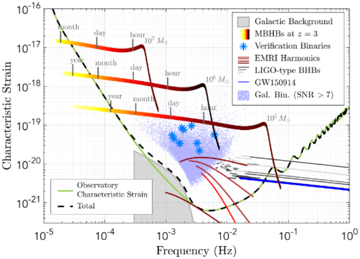
- Deterministic signals
  - Typically from binaries
  - Time coherency
  - Localized in the sky
- GW Backgrounds (GWBs)
  - Superposition of many signals
  - Early Universe processes
  - No time coherency
  - Diffuse signals



* Figures from https://www.ligo.org/detections/images/ligoGW150914signals-lg.jpg and https://www.ligo.org/science/GW-Overview/images/stochastic.jpg

Laser Interferometer Space Antenna

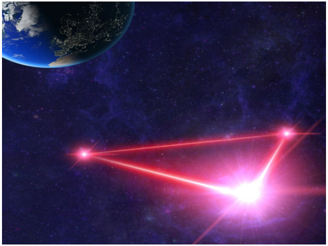


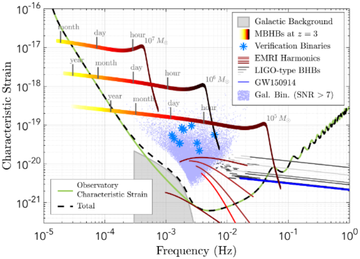


Few details on LISA:

- First GW interferometer in space
- Constellation of three satellites
- 2.5 million km arm lengths
- Peak sensitivity $10^{-2} \div 10^{-3}$Hz
- Three correlated detectors
- Expected launch in mid 2030
- Operating for 4.5 yrs (nominal)

Laser Interferometer Space Antenna





Few details on LISA:

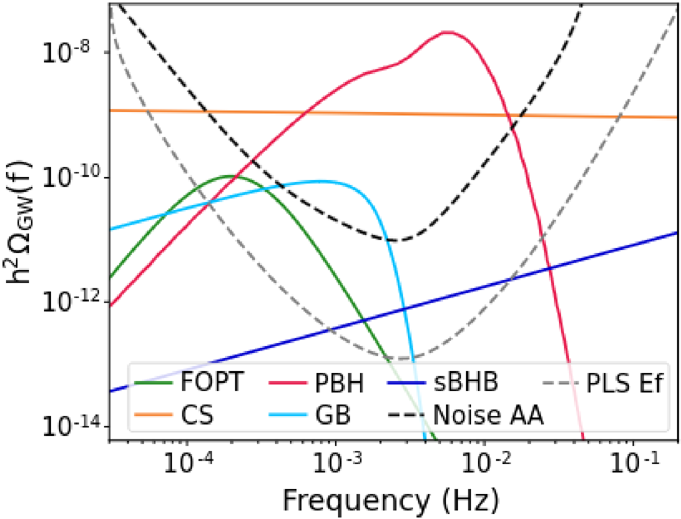
- First GW interferometer in space
- Constellation of three satellites
- 2.5 million km arm lengths
- Peak sensitivity $10^{-2} \div 10^{-3}$Hz
- Three correlated detectors
- Expected launch in mid 2030
- Operating for 4.5 yrs (nominal)

Very interesting for cosmology since we can (among others):

- Measure H_0
- Test modified gravity
- (Hopefully) detect and characterize GWBs!

* Figures from:
https://www.lisamission.org/multimedia/image/lisa-astro2020
LISA Collaboration, P. Amaro-Seoane et al., ArXiv: 1702.00786

Sources of GWBs in the LISA



* Figure from M. Colpi et al., ArXiv:2402.07571

Basics of GW data analysis

Data $\tilde{d}$ (in frequency space) $\longrightarrow$ $\tilde{d} = \tilde{s} + \tilde{n}$

- For individual sources $\langle \tilde{s} \rangle \neq 0$
- For GWBs $\langle \tilde{s} \rangle = 0$

- For noise $\langle \tilde{n} \rangle = 0$

For an isotropic GWB $\longrightarrow$ $\langle h_\lambda(\vec{k})\, h^*_{\lambda'}(\vec{k}') \rangle \propto \delta_{\lambda\lambda'} P_h^\lambda(k) \delta(\vec{k} - \vec{k}')$

Assuming $\langle \tilde{s}\tilde{n} \rangle = 0$ and Gaussian signal and noise

$$\left\langle \tilde{d}^2 \right\rangle = \left\langle \tilde{s}^2 \right\rangle + \left\langle \tilde{n}^2 \right\rangle = \sum_\lambda \mathcal{R}_\lambda\, P_h^\lambda + N \equiv \mathcal{R}\left[P_h + S_n\right]$$

where we have introduced

- The (quadratic) response function of the instrument $\mathcal{R}$
- The (intensity of the) signal power spectrum P_h (in 1/Hz)
- The noise power spectrum N (in 1/Hz)
- The (square of the) Strain sensitivity S_n (in 1/Hz)

Basics of GW data analysis

Data $\tilde{d}$ (in frequency space) $\longrightarrow$ $\tilde{d} = \tilde{s} + \tilde{n}$

- For individual sources $\langle \tilde{s} \rangle \neq 0$
- For GWBs $\langle \tilde{s} \rangle = 0$
- For noise $\langle \tilde{n} \rangle = 0$

For an isotropic GWB $\longrightarrow$ $\langle h_\lambda(\vec{k})\, h^*_{\lambda'}(\vec{k}') \rangle \propto \delta_{\lambda\lambda'} P_h^\lambda(k) \delta(\vec{k} - \vec{k}')$

Assuming $\langle \tilde{s}\tilde{n} \rangle = 0$ and Gaussian signal and noise

$$\left\langle \tilde{d}^2 \right\rangle = \left\langle \tilde{s}^2 \right\rangle + \left\langle \tilde{n}^2 \right\rangle = \sum_\lambda \mathcal{R}_\lambda\, P_h^\lambda + N \equiv \mathcal{R}\left[P_h + S_n\right]$$

where we have introduced

- The (quadratic) response function of the instrument $\mathcal{R}$
- The (intensity of the) signal power spectrum P_h (in 1/Hz)
- The noise power spectrum N (in 1/Hz)
- The (square of the) Strain sensitivity S_n (in 1/Hz)

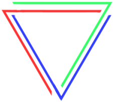
In order to compare with cosmological predictions it's customary to introduce

$$\Omega_{\rm GW} \equiv \frac{1}{3H_0^2 M_p^2} \frac{\partial \rho_{\rm GW}}{\partial \ln f} = \frac{4\pi^2}{3H_0^2} f^3 P_h \qquad \text{and} \qquad \Omega_n(f) = \frac{4\pi^2}{3H_0^2} f^3 S_n(f)\,,$$

where $H_0 \simeq h_0 \times 3.24 \times 10^{-18}\,\mathrm{Hz}$ is the Hubble parameter today.

Time Delay Interferometry

The simplest option is to build three (correlated) Michelson-like data streams

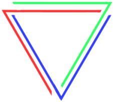


Laser frequency noise is **huge** and must be suppressed with Time Delay Interferometry (TDI)

TDI = Combine measurements at different times to achieve noise reduction!

Time Delay Interferometry

The simplest option is to build three (correlated) Michelson-like data streams

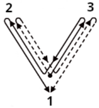


Laser frequency noise is **huge** and must be suppressed with Time Delay Interferometry (TDI)

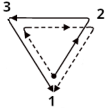
TDI = Combine measurements at different times to achieve noise reduction!

Using η_{ij} measurement in i coming from j and the delay operator D_{ij} we define:

- Michelson variables, dubbed XYZ, defined as (YZ are permutations):
  $X \equiv (1 - D_{13}D_{31})(\eta_{12} + D_{12}\eta_{21}) + (D_{12}D_{21} - 1)(\eta_{13} + D_{13}\eta_{31})$

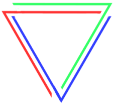


- Sagnac variables, dubbed $\alpha\beta\gamma$, defined as ($\beta\gamma$ permutations):
  $\alpha \equiv \eta_{12} + D_{12}\eta_{23} + D_{12}D_{23}\eta_{31} - (\eta_{13} + D_{13}\eta_{32} + D_{13}D_{32}\eta_{21})$



Time Delay Interferometry

The simplest option is to build three (correlated) Michelson-like data streams

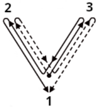


Laser frequency noise is **huge** and must be suppressed with Time Delay Interferometry (TDI)

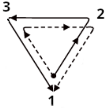
TDI = Combine measurements at different times to achieve noise reduction!

Using η_{ij} measurement in i coming from j and the delay operator D_{ij} we define:

- Michelson variables, dubbed XYZ, defined as (YZ are permutations):
  $\mathrm{X} \equiv (1 - D_{13}D_{31})(\eta_{12} + D_{12}\eta_{21}) + (D_{12}D_{21} - 1)(\eta_{13} + D_{13}\eta_{31})$



- Sagnac variables, dubbed $\alpha\beta\gamma$, defined as ($\beta\gamma$ permutations):
  $\alpha \equiv \eta_{12} + D_{12}\eta_{23} + D_{12}D_{23}\eta_{31} - (\eta_{13} + D_{13}\eta_{32} + D_{13}D_{32}\eta_{21})$



In these variables, signal and noise (in different channels) are correlated!

For equal arms, diagonalization via:
(diagonal variables dubbed AET and $\mathcal{AET}$)

$$\longrightarrow \begin{pmatrix} -1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1/\sqrt{6} & -2/\sqrt{6} & 1/\sqrt{6} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{pmatrix}$$

The LISA noise model

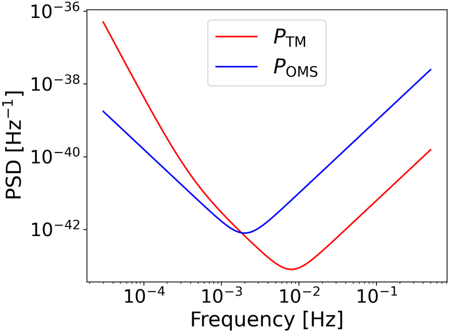
After TDI two main noise components:

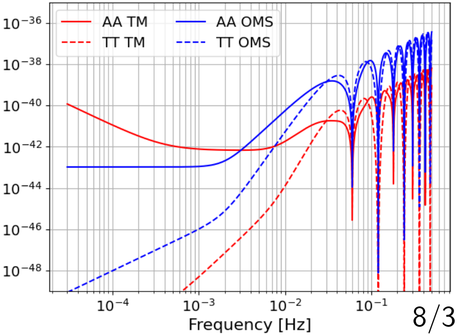
Low frequencies are dominated by Test Mass (TM) noise
large frequencies by Optical Metrology System (OMS) noise:

$$P_{TM}(f, A) = A^2 \times 10^{-30} \times F_{TM}(f) \, ,$$

$$P_{OMS}(f, P) = P^2 \times 10^{-24} \times F_{OMS}(f) \, ,$$

where $F_{TM}(f)$, $F_{OMS}(f)$ are some functions of frequency.



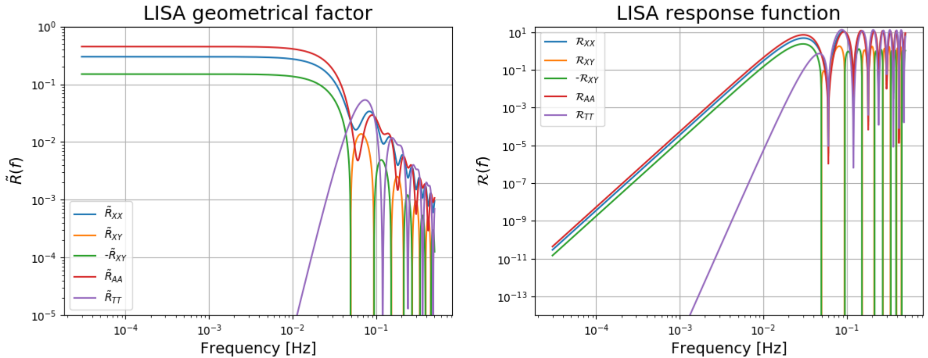


The LISA response function

For an isotropic and non-chiral spectrum we get (see 2009.11845):

$$\left\langle s_i^{TDI} s_j^{TDI} \right\rangle = \int \mathrm{d}k\, P_h(k) \mathcal{R}_{ij}^{TDI}(k)\,, \qquad \mathcal{R}_{ij}^{TDI}(k) \equiv 4\,(2\pi k L)^2 |W^{TDI}(kL)|^2 \tilde{R}_{ij}^{TDI}(k)\,.$$

where $\mathcal{R}_{ij}^{TDI}(k)$ is the LISA response and $\tilde{R}_{ij}^{TDI}(k)$ is a geometrical factor.
For XYZ/AET (AET is $\sim$ diagonal) combinations we get:

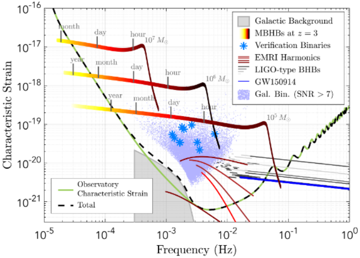


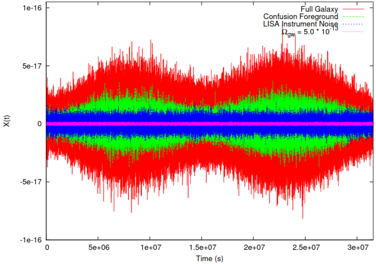
At low frequencies the TT response is suppressed by a factor f^6!
See https://github.com/Mauropieroni/GW_response

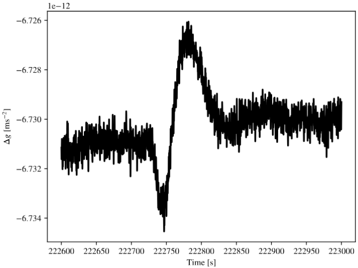
Stationarity won't hold ...

The LISA data won't be stationary:

- There will be transient signals
- Measured GWBs change with time
- There will be transient gliches



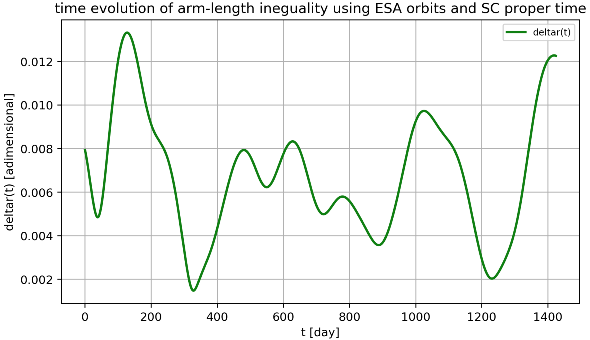




* LISA Collaboration, P. Amaro-Seoane et al., ArXiv: 1702.00786. M. R. Adams, Phys. Rev. D 89 (2014), 022001, ArXiv: 1307.4116. Q. Baghi et al., Phys. Rev. D 105 (2022), 042002, ArXiv: 2112.07490.

... LISA won't have equal arms...

Fluctuations in the arm-lengths of order up to 10^{-2} are expected!



Response functions and noise spectra will be modified $\longrightarrow$

- Orthogonality of TDI variables might be affected
- T is not signal orthogonal anymore
- ...

An accurate description is necessary to avoid biases in the analysis!

... and noise characterization is not as simple!

Let us have a closer look at the problem of noise characterization
(still stick with TM and OMS noise only with known templates)

Each spacecraft contains two test masses and two lasers $\longrightarrow$ 12 (6 Acc +6 OMS) independent noise components are expected!

$$\begin{pmatrix} A & 0 \\ 0 & P \end{pmatrix} \longrightarrow \begin{pmatrix} \begin{matrix} A_{12} & 0 & 0 & D_{12}A_{21} & 0 & 0 \\ 0 & A_{23} & 0 & 0 & D_{23}A_{32} & 0 \\ 0 & 0 & A_{31} & 0 & 0 & D_{31}A_{13} \\ D_{12}A_{12} & 0 & 0 & A_{21} & 0 & 0 \\ 0 & D_{23}A_{23} & 0 & 0 & A_{32} & 0 \\ 0 & 0 & D_{31}A_{31} & 0 & 0 & A_{13} \end{matrix} & 0 \\ 0 & \begin{matrix} P_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{31} & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{21} & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{32} & 0 \\ 0 & 0 & 0 & 0 & 0 & P_{13} \end{matrix} \end{pmatrix}$$

Several complications are added in the problem:

- Noise components propagate differently in different TDI variables
- Higher dimensionality of the parameter space
- Correlations between the noise parameters

Again, this requires care!

Simulation-Based-Inference (SBI)

Traditional methods (MCMC, nested sampling, whatever) are quite efficient and guaranteed to converge (in some cases)

but

scale poorly with number of parameters and require explicit likelihoods

Can alternative approaches perform better in some cases?

Simulation-Based-Inference (SBI)

Traditional methods (MCMC, nested sampling, whatever) are quite efficient and guaranteed to converge (in some cases)

but

scale poorly with number of parameters and require explicit likelihoods

Can alternative approaches perform better in some cases?

Normally, with Bayesian inference, we try to study the posterior probability:

$$p(\theta|d) = \frac{p(d|\theta)\,\pi(\theta)}{p(d)}$$

the idea of SBI is to use simulated (mock) data to build a (ML-based) emulator for (a part of) the posterior!

For example:

- Neural Posterior Estimation (NPE) directly targets $p(\theta|d)$
- Neural Ratio Estimation (NRE) targets $r(d,\theta)$ defined as

$$r(d,\theta) \equiv \frac{p(d|\theta)}{p(d)} = \frac{p(\theta|d)}{\pi(\theta)} = \frac{p(\theta,d)}{p(d)\,\pi(\theta)}\,,$$

- ...

Basic ingredients and benefits of SBI

Key ingredients of the approach:

- A reasonable prior range for the parameters θ
- A simulator to generate mock data x_i given the parameters θ_i
- A neural network approximant, say $f_\phi(x, \bar{\theta})$, of the (unknow) target pdf
- A loss function $\mathcal{L}_\phi$ to minimize during training
- A training strategy/algorithm to optimize the parameters ϕ of the network

Basic ingredients and benefits of SBI

Key ingredients of the approach:

- A reasonable prior range for the parameters θ
- A simulator to generate mock data x_i given the parameters θ_i
- A neural network approximant, say $f_\phi(x, \bar{\theta})$, of the (unknow) target pdf
- A loss function $\mathcal{L}_\phi$ to minimize during training
- A training strategy/algorithm to optimize the parameters ϕ of the network

Notice that:

- SBI does not require an explicit likelihood $\longrightarrow$ likelihood-free inference
- $f_\phi(x, \bar{\theta})$ is a smooth (infinitely differentiable function)
- $\bar{\theta}$ (target parameters), can be a subset of θ $\longrightarrow$ automatic marginalization
- SBI requires *independent* simulations $\longrightarrow$ fully profit of GPU scalability

There are good reasons to test its application ...

Neural Ratio Estimation (NRE)

As already mentioned:

$$r(d,\theta) \equiv \frac{p(d|\theta)}{p(d)} = \frac{p(\theta|d)}{\pi(\theta)} = \frac{p(\theta,d)}{p(d)\,\pi(\theta)}\ ,$$

i.e., $r(d,\theta)$ is the ratio between joint probability and marginal probability.

Given a pair (θ, d), $r(d,\theta)$ can be used to assess whether θ can generate d!
The network will be a classifier to say whether θ, d are joint or marginal....

Neural Ratio Estimation (NRE)

As already mentioned:

$$r(d,\theta) \equiv \frac{p(d|\theta)}{p(d)} = \frac{p(\theta|d)}{\pi(\theta)} = \frac{p(\theta,d)}{p(d)\,\pi(\theta)}\,,$$

i.e., $r(d,\theta)$ is the ratio between joint probability and marginal probability.

Given a pair (θ, d), $r(d,\theta)$ can be used to assess whether θ can generate d! The network will be a classifier to say whether θ, d are joint or marginal....

The typical loss function for NRE is

$$\mathcal{L}[f_\phi] = -\int \mathrm{d}x\,\mathrm{d}\theta\;[p(x,\theta)\,\ln(\sigma(f_\phi(x,\theta))) + p(x)\pi(\theta)\,\ln(1-\sigma(f_\phi(x,\theta)))],$$

where $\sigma(x) = (1+e^{-x})^{-1}$ is the sigmoid function.

- Integrals are computed via Monte Carlo integration
- The *optimal* classifier $f^\star_\phi(x,\theta)$ is precisely $f^\star_\phi(x,\theta) = \ln r(x,\theta)$.

Let's see how this works in practice ...

A very simple example

Let's consider:

$$d = \theta^2 + \epsilon$$

with

$$\theta \sim U[-1, 1]$$
$$\epsilon \sim U[-.2, .2]$$

A very simple example

Let's consider:

$$d = \theta^2 + \epsilon$$

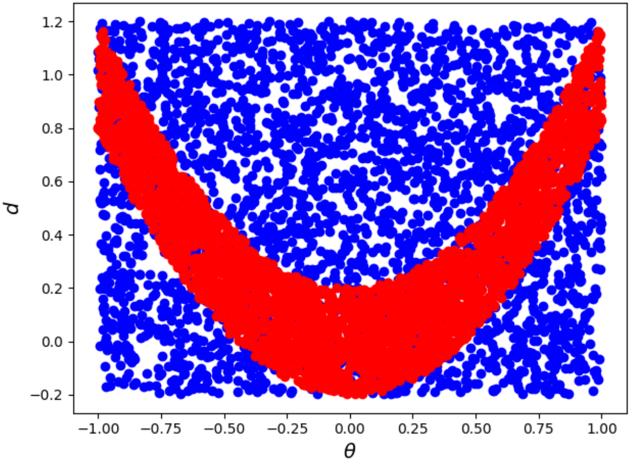
with

$\theta \sim U[-1, 1]$

$\epsilon \sim U[-.2, .2]$

Marginal samples
$d, \theta \sim p(d)p(\theta)$

Joint samples
$d, \theta \sim p(d, \theta)$



A very simple example

Let's consider:

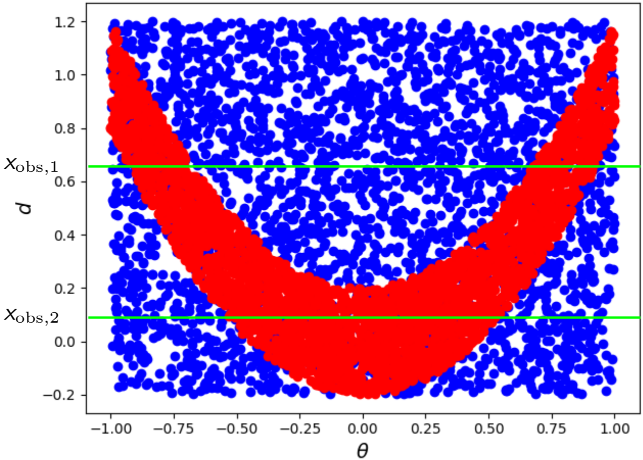
$$d = \theta^2 + \epsilon$$

with

$$\theta \sim U[-1, 1]$$
$$\epsilon \sim U[-.2, .2]$$

Marginal samples
$d, \theta \sim p(d)p(\theta)$

Joint samples
$d, \theta \sim p(d, \theta)$



In very large parameter spaces you can target a subset of the parameters!

↓

Automatically marginalize over the other parameters (here ϵ)!

Neural Ratio Estimation (NPE)

The the Kullback-Leibler (KL) divergence is defined as

$$\mathrm{KL}(P\|Q) \equiv \int P(\theta) \log \frac{P(\theta)}{Q(\theta)} \, \mathrm{d}\theta.$$

measures the difference between two pdfs P and Q.

Neural Ratio Estimation (NPE)

The the Kullback-Leibler (KL) divergence is defined as

$$\mathrm{KL}(P\|Q) \equiv \int P(\theta) \log \frac{P(\theta)}{Q(\theta)} \, \mathrm{d}\theta .$$

measures the difference between two pdfs P and Q.

The typical loss function for NPE is

$$\mathcal{L}(\phi) = -\int p(\theta, x) \log q_\phi(\theta|x) \, \mathrm{d}\theta \, \mathrm{d}x \simeq -\mathbb{E}_{p(\theta,x)}[\log q_\phi(\theta|x)],$$

- Again, the integral is computed via Monte Carlo integration
- Up to ϕ independent terms, $\mathcal{L}(\phi) = \mathrm{KL}(p(\theta|x)\|q_\phi(\theta|x))$

$$\mathcal{L}(\phi) = \left(-\int p(\theta|x)p(x) \log q_\phi(\theta|x) \, \mathrm{d}\theta \, \mathrm{d}x \right)$$

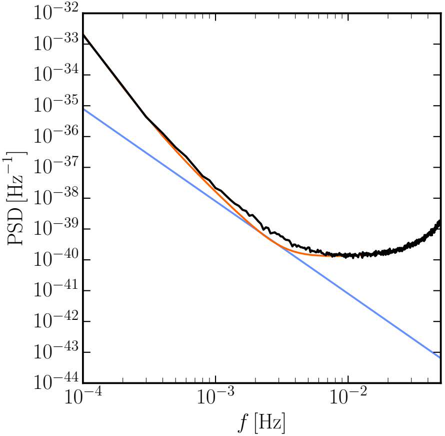
$$\simeq \int p(x) \left(\int p(\theta|x) \log \frac{p(\theta|x)}{q_\phi(\theta|x)} \, \mathrm{d}\theta \right) \mathrm{d}x \simeq \mathbb{E}_{p(x)} \left[\mathrm{KL}(p(\theta|x)\|q_\phi(\theta|x)) \right],$$

where $p(x) = \int p(x|\theta)\pi(\theta)\mathrm{d}\theta$ is the marginal data distribution.

Thus, $q_\phi(\theta|x)$ is an approximant for the full posterior $p(\theta|x)$

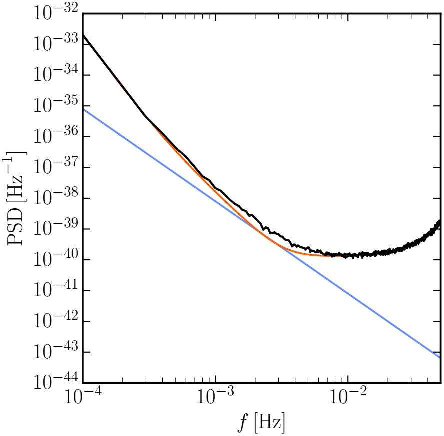
Recover previous results ...

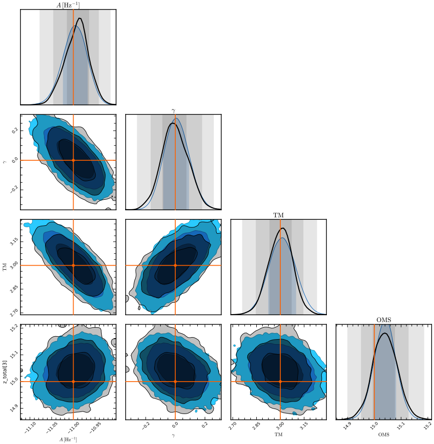
Assume we inject a power law signal:
Can we recover it with the same level of accuracy?



Recover previous results ...

Assume we inject a power law signal:
Can we recover it with the same level of accuracy?





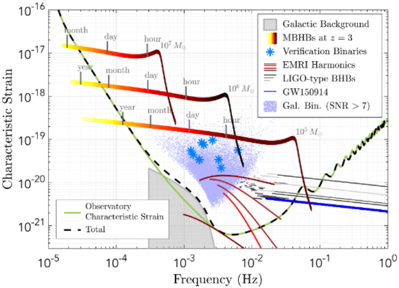
Good news!

James Alvey et al., Phys. Rev. D 109 (2024) 083008, ArXiv:2309.07954.

Code available at `https://github.com/PEREGRINE-GW/saqqara/`

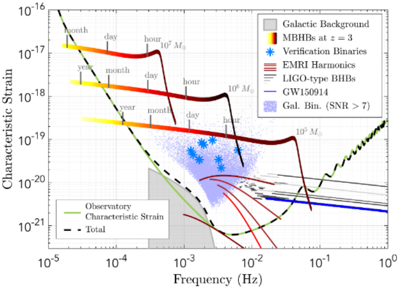
... plus something completely new!

What if there's something else beyond GWB and noise?



… plus something completely new!

What if there's something else beyond GWB and noise?

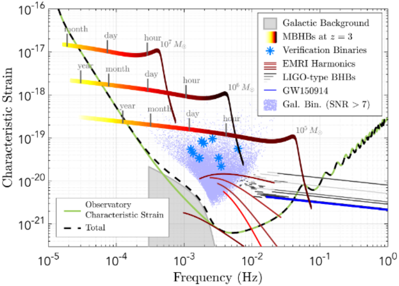


A minimal setup for this: randomly inject transients slightly below the threshold for detection

Would this still work??

… plus something completely new!

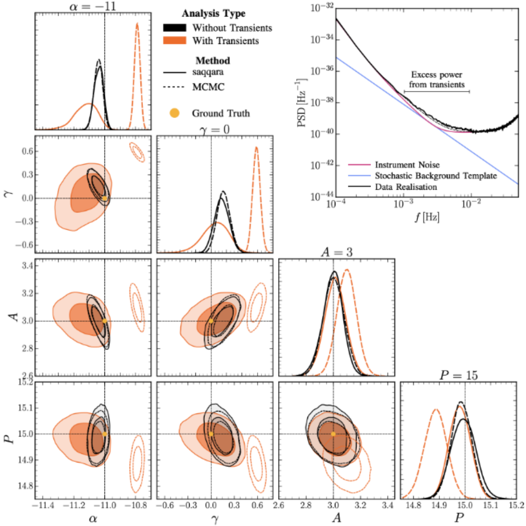
What if there's something else
beyond GWB and noise?



A minimal setup for this:
randomly inject transients slightly below the threshold for detection

Would this still work??

Yes!!



James Alvey et al., Phys. Rev. D 109 (2024) 083008, ArXiv:2309.07954.
Code available at https://github.com/PEREGRINE-GW/saqqara/

What about noise non-stationarities? (FIM)

The noise won't be stationary for the whole mission duration ...

How does this impact the signal parameters reconstruction?

What about noise non-stationarities? (FIM)

The noise won't be stationary for the whole mission duration ...

How does this impact the signal parameters reconstruction?

A strategy to answer this question:

1. Cut the data into segments (where stationarity holds)
2. Analyze segment-by-segment (FIM or training a network)
3. Combine the results (assuming each segment is an independent draw)

What about noise non-stationarities? (FIM)

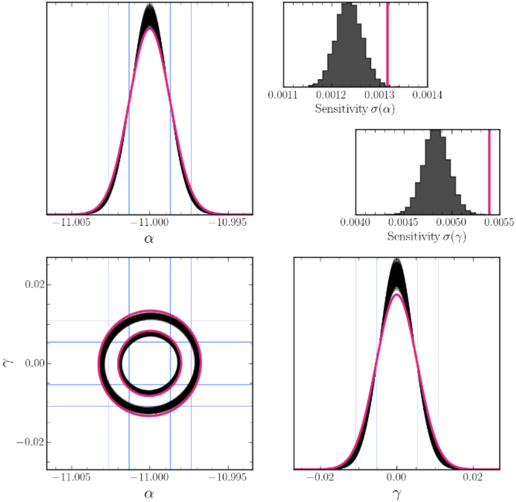
The noise won't be stationary for the whole mission duration ...

How does this impact the signal parameters reconstruction?

A strategy to answer this question:

1. Cut the data into segments (where stationarity holds)
2. Analyze segment-by-segment (FIM or training a network)
3. Combine the results (assuming each segment is an independent draw)

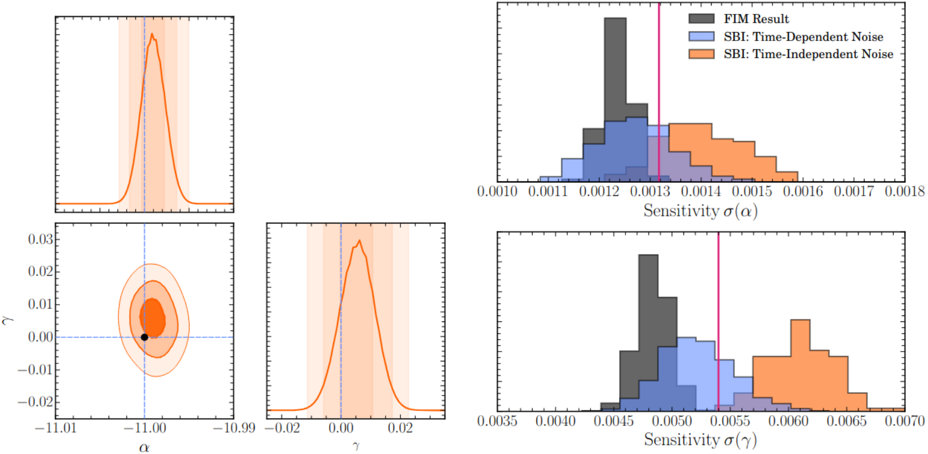
First test with FIM:
Looks like you actually do better!



James Alvey et al., ArXiv:2408.00832.
Code available at `https://github.com/PEREGRINE-GW/saqqara/`
See also `https://github.com/Mauropieroni/GW_response`

What about noise non-stationarities? (Full analysis)

Validate FIM results with a full analysis



Moreover, training segment-by-segment is 100 times faster!

James Alvey et al., ArXiv:2408.00832.
Code available at `https://github.com/PEREGRINE-GW/saqqara/`
See also `https://github.com/Mauropieroni/GW_response`

Ensamble learning for SBI

Can we assess the convergence / test model mis-specifications?

Ensamble learning for SBI

Can we assess the convergence / test model mis-specifications?

A pretty useful lesson for MCMC:

Do not run a single chain!

Runnig many chains (in parallel) and checking convergence is better

Ensamble learning for SBI

Can we assess the convergence / test model mis-specifications?

A pretty useful lesson for MCMC:

Do not run a single chain!

Runnig many chains (in parallel) and checking convergence is better

You can do the same (train several networks in parallel) with SBI

In ML context this is known as ensamble learning

Ensamble learning for SBI

Can we assess the convergence / test model mis-specifications?

A pretty useful lesson for MCMC:

***Do not* run a single chain!**

Runnig many chains (in parallel) and checking convergence is better

You can do the same (train several networks in parallel) with SBI

In ML context this is known as ensamble learning

Some observations:

- Checking the consistency between the networks is great for diagnostics
- KL is a natural choice for this ... and brings different info from the loss
- KL can be mapped into (interpreted as) σ levels
- Train until all KLs are below some value ensures consistency
- With an ensamble we can test model mis-specifications!

In collaboration with James Alvey and Carlo Contaldi, ArXiv:2507.13495.

KL to assess convergence

Let's consider 1d Gaussians

expressing $\sigma_2 = \sigma_1(1+\epsilon)$ and $\mu_2 = \mu_1 + \gamma\sigma_1$ with ϵ, $\gamma \ll 1$ we get

$$\mathrm{KL}(\mathcal{N}_1 \| \mathcal{N}_2) \simeq \frac{\gamma^2}{2} + \epsilon^2 - \gamma^2\epsilon$$

pretty clear scalings with ϵ and γ + can be generalized to *n* dimensions!

KL to assess convergence

Let's consider 1d Gaussians

expressing $\sigma_2 = \sigma_1(1+\epsilon)$ and $\mu_2 = \mu_1 + \gamma\sigma_1$ with ϵ, $\gamma \ll 1$ we get

$$\mathrm{KL}(\mathcal{N}_1 \| \mathcal{N}_2) \simeq \frac{\gamma^2}{2} + \epsilon^2 - \gamma^2\epsilon$$

pretty clear scalings with ϵ and γ + can be generalized to n dimensions!

Can we learn something more from this?

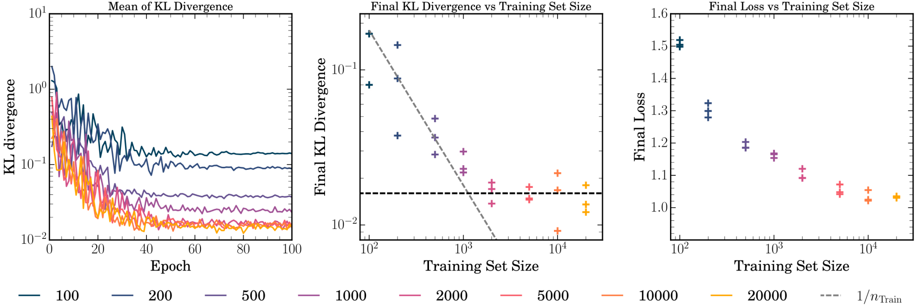
KL to assess convergence

Let's consider 1d Gaussians
expressing $\sigma_2 = \sigma_1(1+\epsilon)$ and $\mu_2 = \mu_1 + \gamma\sigma_1$ with ϵ, $\gamma \ll 1$ we get

$$\mathrm{KL}(\mathcal{N}_1 \| \mathcal{N}_2) \simeq \frac{\gamma^2}{2} + \epsilon^2 - \gamma^2\epsilon$$

pretty clear scalings with ϵ and γ + can be generalized to *n* dimensions!

Can we learn something more from this?



KL brings extra information compared to the loss function!

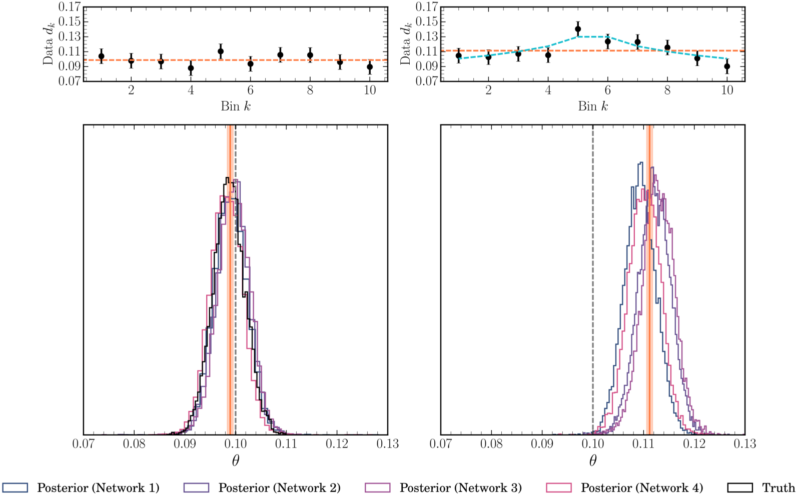
In collaboration with James Alvey and Carlo Contaldi, ArXiv:2507.134955.

Test model mis-specifications I

Consider $d_k \sim \mathcal{N}(\mu_k, \sigma_k)$, with $\sigma_k = 0.01$ and $\mu_k = \theta + a\mathcal{B}_k$
Train for $a = 0$, then apply on data with $a \neq 0$, what happens?

Test model mis-specifications I

Consider $d_k \sim \mathcal{N}(\mu_k, \sigma_k)$, with $\sigma_k = 0.01$ and $\mu_k = \theta + a\mathcal{B}_k$
Train for $a = 0$, then apply on data with $a \neq 0$, what happens?



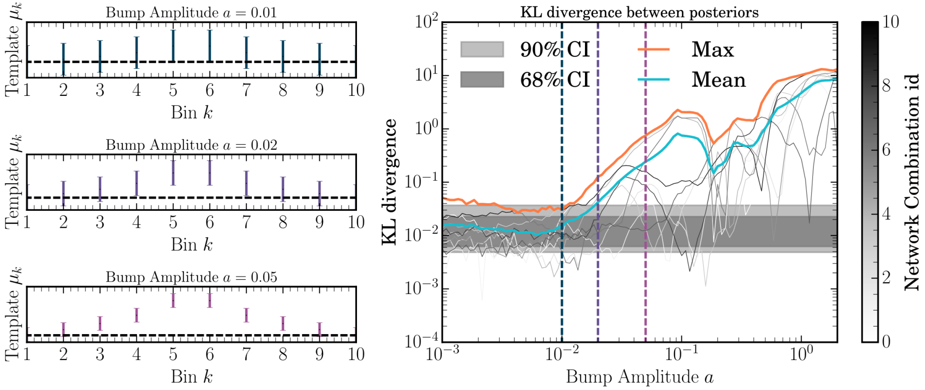
In collaboration with James Alvey and Carlo Contaldi, ArXiv:2507.13495.

Test model mis-specifications II

How does KL increase with a (i.e., increasing the bump height)?

Test model mis-specifications II

How does KL increase with *a* (i.e., increasing the bump height)?



KL differences start to be grow for a $\sim 1\sigma$ deviations!

In collaboration with James Alvey and Carlo Contaldi, ArXiv:2507.13495.

Dynamical (Round-free) Sequential SBI: the general idea

Problem: Amortization struggles for large posterior to prior ratios.

To overcome this issue, we can use Sequential SBI

Key idea: use the posterior from the round i to update the prior for round $i+1$.

Main drawback: requires some criterion to decide when to stop one round.

Dynamical (Round-free) Sequential SBI: the general idea

Problem: Amortization struggles for large posterior to prior ratios.

To overcome this issue, we can use Sequential SBI

Key idea: use the posterior from the round i to update the prior for round $i + 1$.

Main drawback: requires some criterion to decide when to stop one round.

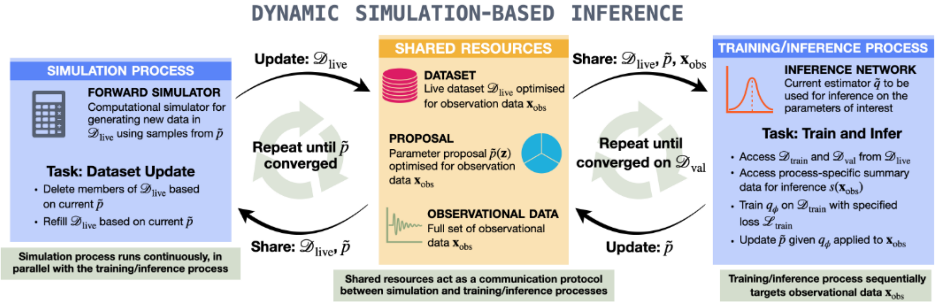
Can we do better than that?

Dynamical (Round-free) Sequential SBI: the general idea

Problem: Amortization struggles for large posterior to prior ratios.

To overcome this issue, we can use Sequential SBI
Key idea: use the posterior from the round i to update the prior for round $i+1$.
Main drawback: requires some criterion to decide when to stop one round.

Can we do better than that?



H. Lyu, et al., ArXiv: 2510.13997.
Code available at `https://github.com/dynamic-sBI/chameleon-sbi` and `github.com/dynamic-sbi/falcon-dsbi/`

Dynamical (Round-free) Sequential SBI: an example run

Does it work for some non-trivial problems?

Consider a 10d problem with complex posterior structure i.e., multi-modality
$+$
large posterior to prior ratio $10^4/10^{-2}$ per dimension, i.e. $\sim\ 10^{60}$

Can we recover the posterior?

Dynamical (Round-free) Sequential SBI: an example run

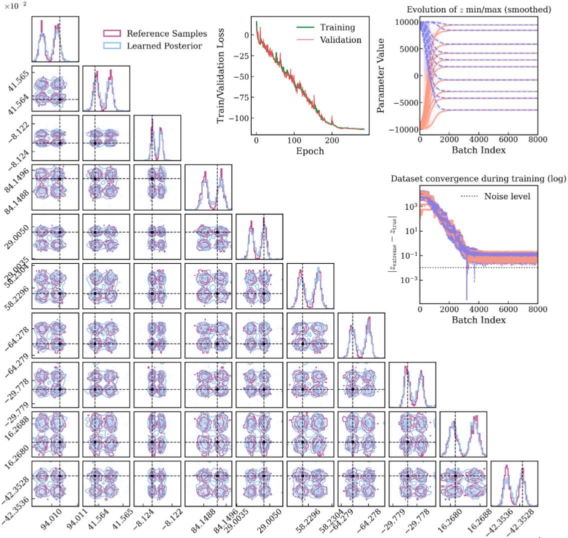
Does it work for some non-trivial problems?

Consider a 10d problem with complex posterior structure i.e., multi-modality
+
large posterior to prior ratio $10^4/10^{-2}$ per dimension, i.e. $\sim\ 10^{60}$

Can we recover the posterior?

Looks so!

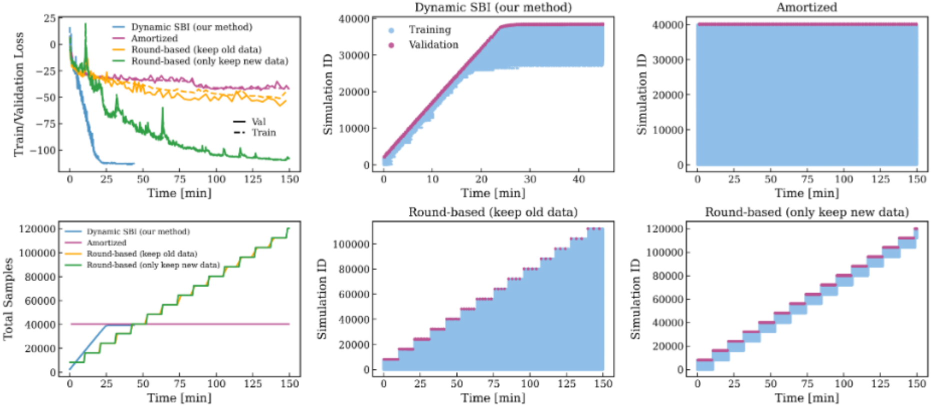
H. Lyu, et al., ArXiv: 2510.13997.
Code available at
`https://github.com/dynamic-sBI/chameleon-sbi`
and `github.com/dynamic-sbi/falcon-dsbi/`



Dynamical (Round-free) Sequential SBI: benchmark test

... great but does it work better?

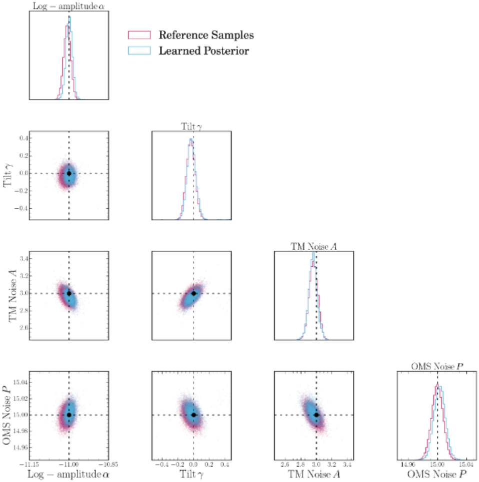
Let's compare it with other training strategies:

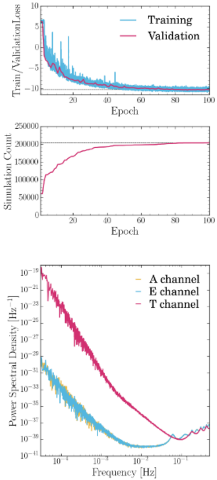


H. Lyu, et al., ArXiv: 2510.13997.
Code available at `https://github.com/dynamic-sBI/chameleon-sbi`
and `github.com/dynamic-sbi/falcon-dsbi/`

Dynamical (Round-free) Sequential SBI: SGWB with LISA

Does it work well for “real life” problems? e.g., SGWB with LISA data?





H. Lyu, et al., ArXiv: 2510.13997. Code available at `https://github.com/dynamic-sBI/chameleon-sbi` and `github.com/dynamic-sbi/falcon-dsbi/`

Conclusions and outlook

Conclusions

- Analyzing LISA data will be quite difficult
- Traditional methods work (but they might be expensive)
- SBI (and ML in general) looks quite promising ...
- ... it works with transients and non-stationarities in the noise!
- Some convergence criteria can be quite useful

Future perspectives

- Keep improving on detector modeling
- More realistic noise model
- Faster data generation procedure?
- More complexity in the simulator
- New ideas/techniques?
- ...

Last slide

The end

Thank you for your attention!