Towards a realistic forecast detection of Primordial Gravitational Wave Backgrounds

Valencia, 9th-13th December 2024

SBI Approach to GW Reconstruction (A case study on Backgrounds)

Bryan Zaldívar

IFIC

In collaboration with:

Andromki Dimitriou, Dani G. Figueroa & Peera Simakachorn

OUTLINE

- Simulation-based Inference (SBI)

  (Intro, Motivation, GWB context)

- Case study on SBI of arbitrary backgrounds

  ("Blind" reconstruction, incl. foregrounds; templates)

- Conclusions

STATISTICAL APPROACH

or time domain, or both

Aim: Given some $D^{obs}(f)$ data/simulation, infer the physical parameters of interest $\vec{\theta}$

STATISTICAL APPROACH

or time domain, or both

Aim: Given some $D^{obs}(f)$ data/simulation, infer the physical parameters of interest $\vec{\theta}$

simulator

$\vec{\theta} \Rightarrow$  $\Rightarrow \Omega^{sim}(f)$

STATISTICAL APPROACH

or time domain, or both

Aim: Given some $D^{obs}(f)$ data/simulation, infer the physical parameters of interest $\vec{\theta}$

simulator

$\vec{\theta} \Rightarrow$  $\Rightarrow \Omega^{sim}(f) \Longleftrightarrow D^{obs}(f) \Rightarrow \vec{\theta}$?

STATISTICAL APPROACH

Aim: Given some $D^{obs}(f)$ data/simulation, infer the physical parameters of interest $\vec{\theta}$

or time domain, or both

simulator

$\vec{\theta} \Rightarrow$  $\Rightarrow \Omega^{sim}(f) \Leftrightarrow D^{obs}(f) \Rightarrow \vec{\theta}$?

Traditional approach: Likelihood-based

$$\mathcal{L}(\vec{\theta}) = \prod_{i=1}^{N_{bins}} p\left(D^{obs}(f_i) \,\middle|\, \Omega^{sim}(f_i; \vec{\theta})\right)$$

$$\log \mathcal{L}(\vec{\theta}) = \frac{1}{3} N(\vec{\theta}) + \frac{2}{3} LN(\vec{\theta})$$

STATISTICAL APPROACH

or time domain, or both

Aim: Given some $D^{obs}(f)$ data/simulation,
infer the physical parameters of interest $\vec{\theta}$

simulator

$\vec{\theta} \Rightarrow$  $\Rightarrow \Omega^{sim}(f) \Leftrightarrow D^{obs}(f) \Rightarrow \vec{\theta}$?

Traditional approach: Likelihood-based

$$\mathcal{L}(\vec{\theta}) = \prod_{i=1}^{N_{bins}} p\left(D^{obs}(f_i) \,|\, \Omega^{sim}(f_i;\vec{\theta})\right)$$

$$\log \mathcal{L}(\vec{\theta}) = \frac{1}{3} N(\vec{\theta}) + \frac{2}{3} LN(\vec{\theta})$$

* Fisher Matrix Analysis
* MCMC

$$\text{MCMC}: \left\{ \vec{\theta}_k \right\}_{k=1}^{N_{sim}} \sim p(\vec{\theta} \mid D^{obs}) \propto \mathcal{L}(\vec{\theta}) \cdot p(\vec{\theta})$$

MCMC: $\{\vec{\theta}_k\}_{k=1}^{N_{sim}} \sim p(\vec{\theta} | D^{obs}) \propto \mathcal{L}(\vec{\theta}) \cdot p(\vec{\theta})$

Bias issue $\{$ • Assumes a Likelihood

$$\text{MCMC}: \left\{ \vec{\theta}_k \right\}_{k=1}^{N_{sim}} \sim p(\vec{\theta} \mid D^{obs}) \propto \mathcal{L}(\vec{\theta}) \cdot p(\vec{\theta})$$

Bias issue
- Assumes a Likelihood

Computational issue
- Needs to sample the full posterior
  → typically requires more samples than SBI
- A full MCMC from scratch required for every new D^{obs}

Alternative Likelihood-free approach

(a.k.a. Simulation-based Inference)

Alternative Likelihood-free approach

(a.k.a. Simulation-based Inference)

The true likelihood is intractable in most of the times

Alternative Likelihood-free approach

(a.k.a. Simulation-based Inference)

The true likelihood is intractable in most of the times

[2206.11312]

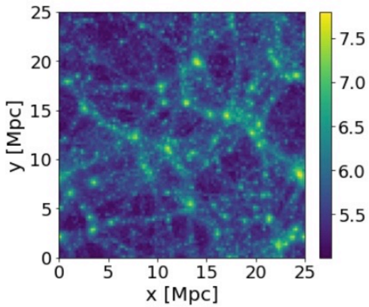


Figure 1. Logarithmic surface density of dark matter

Latent variables { e.g. positions of the DM halos in N-body sims.

Alternative Likelihood-free approach
(a.k.a. Simulation-based Inference)

The true likelihood is intractable in most of the times

[2206.11312]

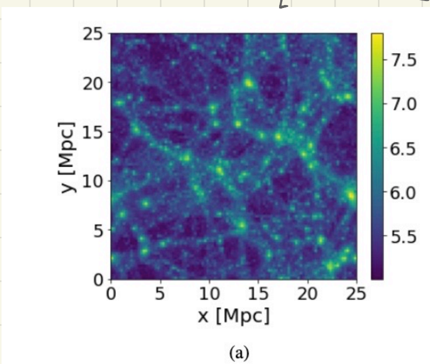
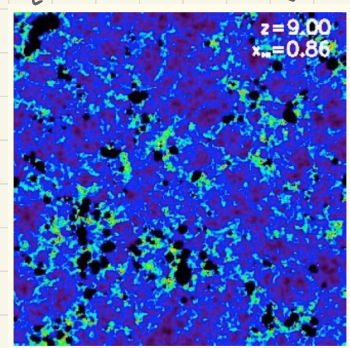


Figure 1. Logarithmic surface density of dark matter

[SKA 1210.0197]



Latent variables { e.g. positions of the DM halos in N-body sims.

Initial conditions of density and velocity fields in 21cm brightness temp. sims

Alternative Likelihood-free approach
(a.k.a. Simulation-based Inference)

The true likelihood is intractable in most of the times

[2206.11312]

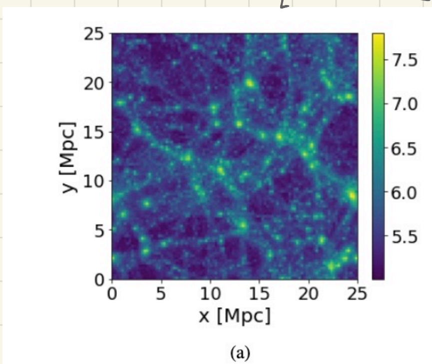
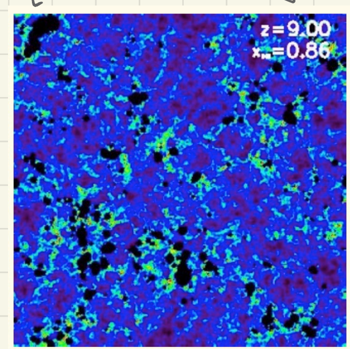
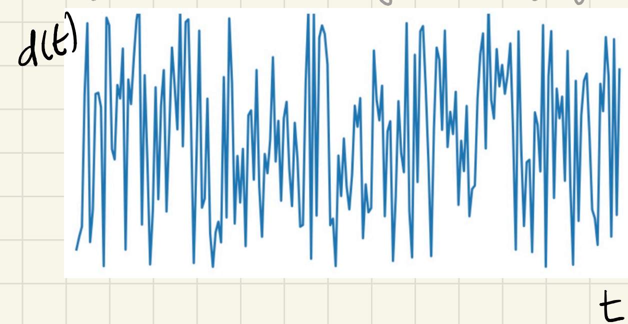


Figure 1. Logarithmic surface density of dark matter

[SKA 1210.0197]



[Taken from talk by Joe Bayley, '22]



Latent variables { e.g. positions of the DM halos in N-body sims.

Initial conditions of density and velocity fields in 21cm brightness temp. sims

Phase and amplitud of a SGWB in Fourier space, e.g. in simulations for LISA

Latent variables $\vec{\eta}$ $\leadsto$ As many as $\mathcal{O}(10^{5-6})$ commonly

Likelihood $p(D^{obs} | f, \vec{\theta}) = \int d\vec{\eta} \; p(D^{obs} | f, \vec{\eta}, \vec{\theta}) \, p(\vec{\eta} | f, \vec{\theta})$

Intractable integral in general

Latent variables $\vec{\eta}$ ⇝ As many as $\mathcal{O}(10^{5-6})$ commonly

Likelihood $p(D^{obs}|f,\vec{\theta}) = \int d\vec{\eta}\; p(D^{obs}|f,\vec{\eta},\vec{\theta})\, p(\vec{\eta}|f,\vec{\theta})$

Intractable integral in general

* Decades old idea (see "Approximate Bayesian Computation")

ABC ~ 80's

with computational disadvantages

* Meteoric revival in the last few years thanks to
  - Old good statistical theorems
  - Nowadays computational power and modern optimisation & **ML** algorithms

MODERN SBI

$p(\vec{\theta} \mid \text{OBS. DATA})$

or

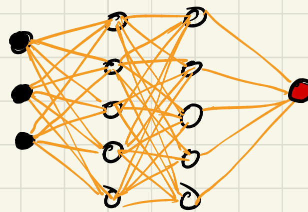
$p(\text{OBS DATA} \mid \vec{\theta})$

MODERN SBI

$p(\vec{\theta} \mid \text{OBS. DATA})$

or

$p(\text{OBS DATA} \mid \vec{\theta})$

$\approx g\Big($  $\Big)$

Inputs

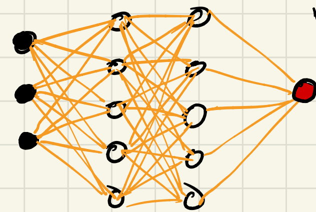
Output

depend on the SBI *flavor*

MODERN SBI

$p(\vec{\theta} \mid \text{OBS. DATA})$

or

$p(\text{OBS DATA} \mid \vec{\theta})$ $\approx g\Big($  $\Big)$

1 $g : \dfrac{p(\text{OBS DATA} \mid \vec{\theta})}{p(\text{OBS DATA})}$

NEURAL RATIO ESTIMATION

Inputs ●
Output ●

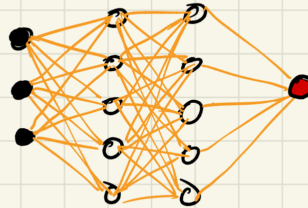
depend on the SBI flavor

MODERN SBI

$$p(\vec{\theta} \mid \text{OBS. DATA})$$

or

$$p(\text{OBS DATA} \mid \vec{\theta}) \approx g\Big(\quad \Big)$$



Inputs

Output

depend on the SBI flavor

1 $g : \dfrac{p(\text{OBS DATA} \mid \vec{\theta})}{p(\text{OBS DATA})}$

NEURAL RATIO ESTIMATION

2 $g : q(\vec{\theta} \mid \text{OBS DATA})$

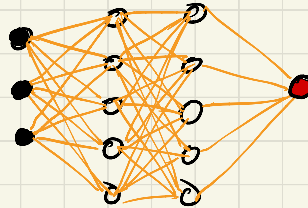
↳ variational distribution

NEURAL POSTERIOR ESTIMATION

MODERN SBI

$p(\vec{\theta} \mid \text{OBS. DATA})$

or

$p(\text{OBS DATA} \mid \vec{\theta}) \approx g\Big($  $\Big)$

Inputs
Output

depend on the SBI flavor

① $g : \dfrac{p(\text{OBS DATA} \mid \vec{\theta})}{p(\text{OBS DATA})}$

NEURAL RATIO ESTIMATION

② $g : q(\vec{\theta} \mid \text{OBS DATA})$

↳ variational distribution

NEURAL POSTERIOR ESTIMATION

③ $g : q(\text{OBS DATA} \mid \vec{\theta})$

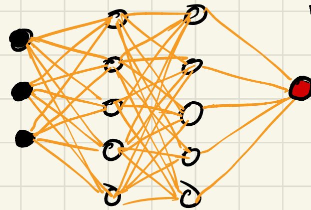
NEURAL LIKELIHOOD ESTIMATION

MODERN SBI

$p(\vec{\theta} \,|\, \text{OBS. DATA})$

or

$p(\text{OBS DATA} \,|\, \vec{\theta}) \approx g\Big($



$\Big)$

Inputs ●
Output ●

depend on the SBI flavor

① $g : \dfrac{p(\text{OBS DATA} \,|\, \vec{\theta})}{p(\text{OBS DATA})}$

NEURAL RATIO ESTIMATION

② $g : q(\vec{\theta} \,|\, \text{OBS DATA})$

↳ variational distribution

NEURAL POSTERIOR ESTIMATION

③ $g : q(\text{OBS DATA} \,|\, \vec{\theta})$

NEURAL LIKELIHOOD ESTIMATION

Crucially:

$g(\cdot)$ fitted to a large # of simulated datasets

Some $\text{DATA}_{\text{SIM}} \approx \text{DATA}_{\text{OBS}}$

Procedure

Procedure

* Input: Parameters $\vec{\theta}$, priors $p(\vec{\theta})$,
  a simulator $\mathcal{S}: \vec{\theta} \longmapsto \{D^{sim}(f)\}$

f	D^{sim}

Procedure

* Input: Parameters $\vec{\theta}$, priors $p(\vec{\theta})$,
a simulator $S: \vec{\theta} \longmapsto \{D^{sim}(f)\}$

f	D^{sim}

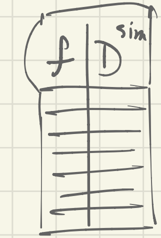
* Algorithm:

① Sample the prior N times

$$\{\vec{\theta}_i\}, \quad i = 1, .., N$$

Procedure

* Input: Parameters $\vec{\theta}$, priors $p(\vec{\theta})$,
  a simulator $S: \vec{\theta} \longmapsto \{D^{sim}(f)\}$



* Algorithm:

① Sample the prior N times

$$\{\vec{\theta}_i\}, \quad i=1,..,N$$

② Run the simulator for each $\vec{\theta}_i$

$\vec{\theta}_i \rightarrow$  $\rightarrow \{D_i^{sim}(f)\}$

$$\text{tuple}_i = \{\vec{\theta}_i \,;\, D_i^{sim}(f)\}$$

Procedure

* Input: Parameters $\vec{\theta}$, priors $p(\vec{\theta})$,
  a simulator $S: \vec{\theta} \mapsto \{D^{sim}(f)\}$

* Algorithm:

① Sample the prior N times

$$\{\vec{\theta}_i\}, \; i = 1, .., N$$

② Run the simulator for each $\vec{\theta}_i$

$\vec{\theta}_i \rightarrow$  $\rightarrow \{D_i^{sim}(f)\}$

$$\text{tuple}_i = \{\vec{\theta}_i \, ; \, D_i^{sim}(f)\}$$

③ Train/fit our posterior approx. $g(\text{tuple})$

Note: No explicit dependence on latent variables

Procedure

* Input: Parameters $\vec{\theta}$, priors $p(\vec{\theta})$,
  a simulator $\mathcal{S}: \vec{\theta} \mapsto \{D^{sim}(f)\}$

* Algorithm:

① Sample the prior N times

$\{\vec{\theta}_i\}, \; i=1,..,N$

② Run the simulator for each $\vec{\theta}_i$

$\vec{\theta}_i \rightarrow$  $\rightarrow \{D_i^{sim}(f)\}$

$\text{tuple}_i = \{\vec{\theta}_i \, ; \, D_i^{sim}(f)\}$

③ ⊛ Train/fit our posterior approx. $g(\text{tuple})$

⇓ **Note:**
No explicit dependence on latent variables

⊛ Pre-trained (a.k.a. "Amortised") strategy ⇒ Inference on a new (obs) dataset takes ≈ no time.

Neural Ratio Estimation

$$g: \frac{p(D(f)|\vec{\theta})}{p(D(f))} \approx \frac{NN(\vec{\theta}; D(f))}{1 - NN(\vec{\theta}; D(f))}$$

Consequence of the Neyman-Pearson lemma

Neural Ratio Estimation

$$g: \frac{p(D(f)|\vec{\theta})}{p(D(f))} \approx \frac{NN(\vec{\theta}; D(f))}{1 - NN(\vec{\theta}; D(f))}$$

Consequence of the Neyman-Pearson lemma

NN: neural network

Task: Binary classification

Neural Ratio Estimation

$$g: \frac{p(D(f)|\vec{\theta})}{p(D(f))} \approx \frac{NN(\vec{\theta}; D(f))}{1 - NN(\vec{\theta}; D(f))}$$

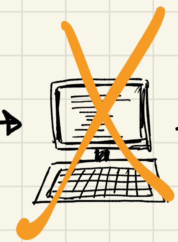
Consequence of the Neyman-Pearson lemma

NN: neural network

Task: Binary classification

$\vec{\theta}_i \rightarrow$  $\rightarrow \{D_i^{sim}(f)\}$

✓ "good"

$\vec{\theta}_i \rightarrow$  $\rightarrow \{D_i^{sim}(f)\}$

"bad"

Neural Posterior Estimation

$g: q_{\vec{\omega}}(\vec{\theta}; D) \approx p(\vec{\theta} \mid D)$

$\vec{\omega}$: optimizable parameters

Neural Posterior Estimation

$$g: \; q_{\vec{\omega}}(\vec{\theta}; D) \approx p(\vec{\theta} \,|\, D)$$

$\vec{\omega}$: optimizable parameters

e.g. $\vec{\theta} = T_{\vec{\omega}}(D, \vec{u})$, $\vec{u} \sim p_{easy}(\vec{u})$

Here using a Normalizing Flow

$\Rightarrow$ if $T_{\vec{\omega}}$ is invertible:

$$q_{\vec{\omega}}(\vec{\theta}) = p_{easy}\left(T^{-1}_{\vec{\omega}}(\vec{\theta})\right) \left|\det J_{T^{-1}}(\vec{\theta})\right|$$

Neural Posterior Estimation

$$g: q_{\vec{\omega}}(\vec{\theta}; D) \approx p(\vec{\theta} | D)$$

$\vec{\omega}$: optimizable parameters

e.g. $\vec{\theta} = T_{\vec{\omega}}(D, \vec{u})$, $\vec{u} \sim p_{easy}(\vec{u})$

Here using a Normalizing Flow

$\Rightarrow$ if $T_{\vec{\omega}}$ is invertible:

$$q_{\vec{\omega}}(\vec{\theta}) = p_{easy}\left(T_{\vec{\omega}}^{-1}(\vec{\theta})\right) \left| \det J_{T^{-1}}(\vec{\theta}) \right|$$

Optimization criterion:

$$\mathbb{E}_D \left[KL\left[p(\vec{\theta} | D) \, ; \, q_{\vec{\omega}}(\vec{\theta}; D) \right] \right] \Leftarrow \text{minimize it}$$

$$\approx -\frac{1}{N_{sims}} \sum_{s=1}^{N_{sims}} \ln q_{\vec{\omega}}(\vec{\theta}_s ; D_s) + \text{const}$$

Neural Posterior Estimation

$$g : q_{\vec{\omega}}(\vec{\theta}; D) \approx p(\vec{\theta} | D)$$

$\vec{\omega}$: optimizable parameters

e.g. $\vec{\theta} = T_{\vec{\omega}}(D, \vec{u})$, $\vec{u} \sim p_{easy}(\vec{u})$

Here using a Normalizing Flow

$\Rightarrow$ if $T_{\vec{\omega}}$ is invertible :

$$q_{\vec{\omega}}(\vec{\theta}) = p_{easy}(T_{\vec{\omega}}^{-1}(\vec{\theta})) \left| \det J_{T^{-1}}(\vec{\theta}) \right|$$

Optimization criterion :

$$\mathbb{E}_D \left[KL\left[p(\vec{\theta} | D) ; q_{\vec{\omega}}(\vec{\theta}; D) \right] \right]$$

$\Leftarrow$ minimize it

$$\approx -\frac{1}{N_{sims}} \sum_{s=1}^{N_{sims}} \ln q_{\vec{\omega}}(\vec{\theta}_s ; D_s) + \text{const}$$

We implement NPE in our analysis

Shallow Look at Bibliography

literature | t simulation-based or t likelihood-free

arXiv Category

astro-ph.CO	74
astro-ph.IM	48
cs.LG	30
hep-ph	26
stat.ML	26
astro-ph.GA	18
astro-ph.HE	16
physics.data-an	15
hep-ex	12
gr-qc	8

Show 10 more

Collaborations
- ATLAS
- DES
- COIN
- COSINUS
- LSST

- N-body simulations
- Gravitational lensing
- Gravitational Wave Stochastic Backgrounds
- Dark Matter Direct & Indirect Detection
- SMEFT at colliders

⋮

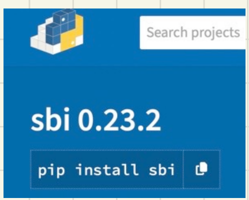
PUBLIC PACKAGES

Python

Main SBI Techniques:

- Neural Posterior Estimation
- Neural Ratio Estimation
- Neural Likelihood Estimation

Packages:



2007.09114
(Tejero-Cantero et al)

[tailored to Collider Physics]

MadMiner: ML based inference for particle physics

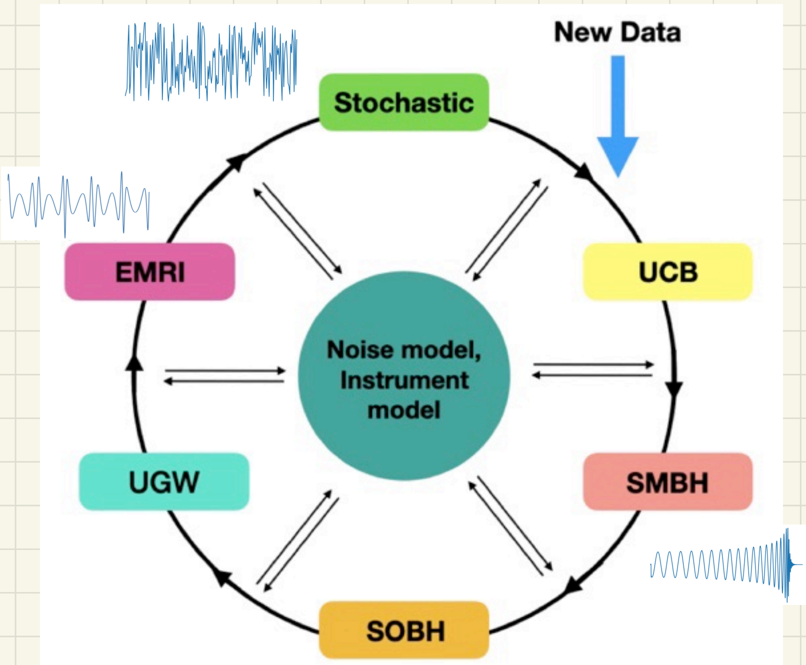
1907.10621
(Kyle Cranmer et al)

SWYFT

2011.13951
(Weniger et al)

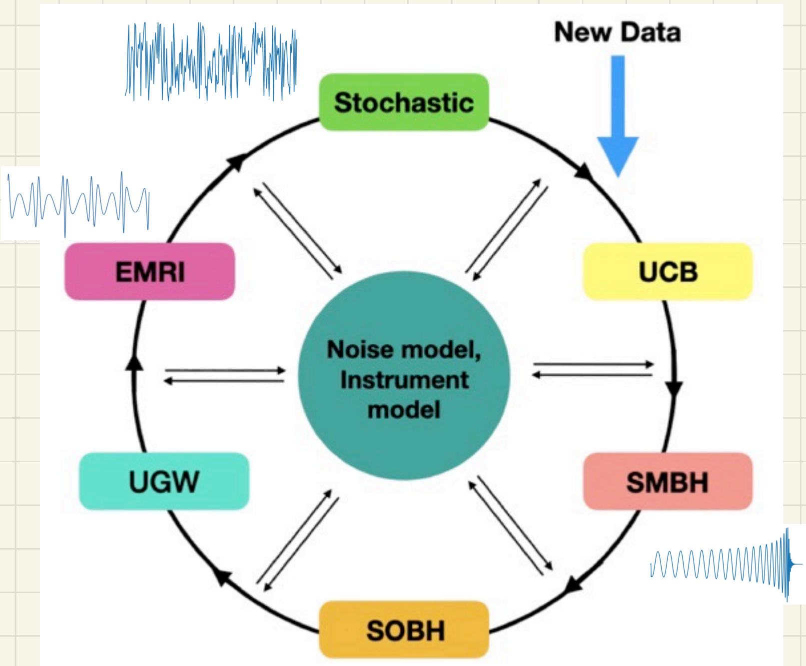
In the context of
GW inference...

The Need for a Global Fit



(Neil's talk, 2023)

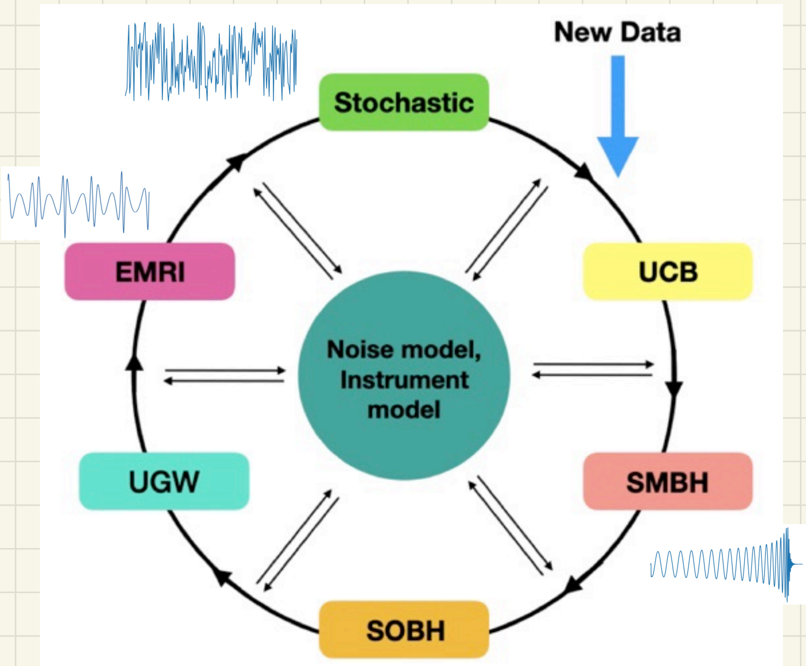
The Need for a Global Fit



(Neil's talk, 2023)

- $\mathcal{O}(10^4)$ expected detectable signals at LISA
  ↳ overlapping in both time and frequency

The Need for a Global Fit

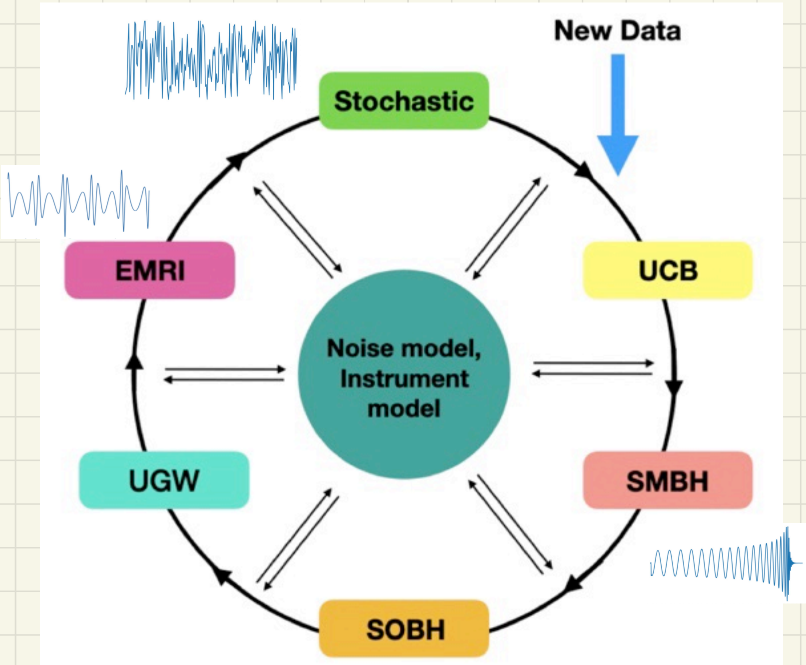


(Neil's talk, 2023)

- $\mathcal{O}(10^4)$ expected detectable signals at LISA
  ↳ overlapping in both time and frequency
- Interdependency among some sources

e.g. SMBH, UCB, EMRI ← Galaxy dynamics

The Need for a Global Fit



(Neil's talk, 2023)

- $\mathcal{O}(10^4)$ expected detectable signals at LISA
  ↳ overlapping in both time and frequency
- Interdependency among some sources
  e.g. SMBH, UCB, EMRI ← Galaxy dynamics
- Need to simultaneously update the fit as new data arrives

Existing Proposal to Global Fit

(Cornish et al, 2301.03673)

Existing Proposal to Global Fit

(Cornish et al, 2301.03673)

- Blocked (trans-dimensional) MCMC sampler

Existing Proposal to Global Fit

(Cornish et al, 2301.03673)

- Blocked (trans-dimensional) MCMC sampler

Block

UCB($\vec{\theta}_1$)

SMBH($\vec{\theta}_2$)

EMRI($\vec{\theta}_3$)

$\vdots$

SGBW($\vec{\theta}_s$)

may even share parameters!

Existing Proposal to Global Fit

(Cornish et al, 2301.03673)

- Blocked (trans-dimensional) MCMC sampler

Block	MCM Chain
UCB $(\vec{\theta}_1)$	$\vec{\theta}_1^{(1)} \rightarrow \vec{\theta}_1^{(2)} \ \ldots$
SMBH $(\vec{\theta}_2)$	$\vec{\theta}_2^{(1)} \rightarrow \vec{\theta}_2^{(2)} \ \ldots$
EMRI $(\vec{\theta}_3)$	$\vec{\theta}_3^{(1)} \rightarrow \vec{\theta}_3^{(2)} \ \ldots$
$\vdots$	
SGBW $(\vec{\theta}_s)$	$\vec{\theta}_s^{(1)} \rightarrow \vec{\theta}_s^{(2)} \ \ldots$

may even share parameters!

synchronized parameter updates

Existing Proposal to Global Fit

(Cornish et al, 2301.03673)

- Blocked (trans-dimensional) MCMC sampler

Block	MCM Chain		
UCB($\vec{\theta}_1$)	$\vec{\theta}_1^{(1)} \to \vec{\theta}_1^{(2)}$	...	
SMBH($\vec{\theta}_2$)	$\vec{\theta}_2^{(1)} \to \vec{\theta}_2^{(2)}$	...	
EMRI($\vec{\theta}_3$)	$\vec{\theta}_3^{(1)} \to \vec{\theta}_3^{(2)}$	...	
⋮			
SGBW($\vec{\theta}_s$)	$\vec{\theta}_s^{(1)} \to \vec{\theta}_s^{(2)}$	...	

may even share parameters!

synchronized parameter updates

Features

✓ • Independent "Blocked" samplers

modular

$$\vec{\theta}_j^{(i+1)} \leftarrow \vec{\theta}_{j,k}^{(i)} \quad \forall k \neq j$$

Existing Proposal to Global Fit

(Cornish et al, 2301.03673)

- Blocked (trans-dimensional) MCMC sampler

Block	MCM Chain		
UCB($\vec{\theta}_1$)	$\vec{\theta}_1^{(1)} \rightarrow$	$\vec{\theta}_1^{(2)}$	…
SMBH($\vec{\theta}_2$)	$\vec{\theta}_2^{(1)} \rightarrow$	$\vec{\theta}_2^{(2)}$	…
EMRI($\vec{\theta}_3$)	$\vec{\theta}_3^{(1)} \rightarrow$	$\vec{\theta}_3^{(2)}$	…
⋮			
SGBW($\vec{\theta}_s$)	$\vec{\theta}_s^{(1)} \rightarrow$	$\vec{\theta}_s^{(2)}$	…

may even share parameters!

synchronized parameter updates

Features

✓ • Independent "Blocked" samplers

modular

$$\vec{\theta}_j^{(i+1)} \leftarrow \vec{\theta}_{j,k}^{(i)} \quad \forall k \neq j$$

✓ • Distributed (parallel) computing of every chain

Existing Proposal to Global Fit

(Cornish et al, 2301.03673)

- Blocked (trans-dimensional) MCMC sampler

Block	MCM Chain		
UCB($\vec{\theta}_1$)	$\vec{\theta}_1^{(1)} \rightarrow \vec{\theta}_1^{(2)}$	...	
SMBH($\vec{\theta}_2$)	$\vec{\theta}_2^{(1)} \rightarrow \vec{\theta}_2^{(2)}$	...	
EMRI($\vec{\theta}_3$)	$\vec{\theta}_3^{(1)} \rightarrow \vec{\theta}_3^{(2)}$	...	
⋮			
SGBW($\vec{\theta}_s$)	$\vec{\theta}_s^{(1)} \rightarrow \vec{\theta}_s^{(2)}$	...	

may even share parameters!

synchronized parameter updates

Features

✓ • Independent "Blocked" samplers → modular

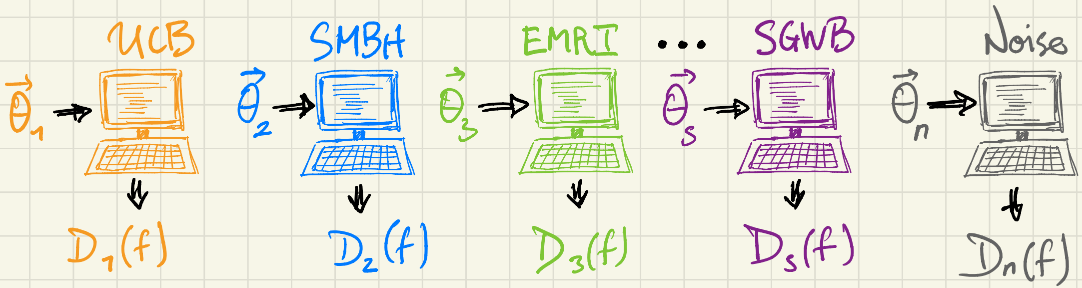
$\vec{\theta}_j^{(i+1)} \leftarrow \vec{\theta}_{j,k}^{(i)} \quad \forall k \neq j$

✓ • Distributed (parallel) computing of every chain

Challenges

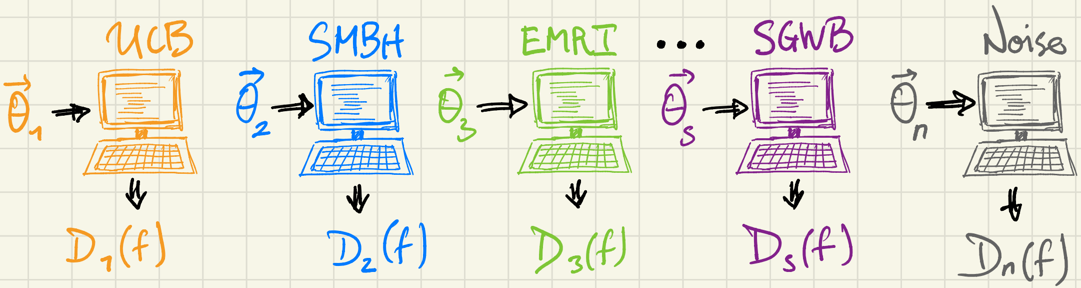
- MCMC acceptance probability (related to the proposal dist.)
- Convergence (related to inter-block dependencies)
- Synchronization overhead and scalability

SBI for GWB physics



$$D(f) = \sum_k D_k(f)$$

SBI for GWB physics



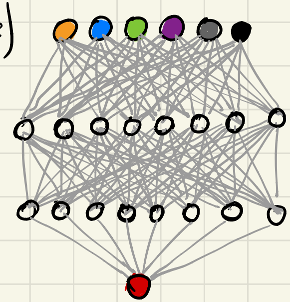
$$D(f) = \sum_k D_k(f)$$

<u>Possible inputs</u>:

a) $\{\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \dots, \vec{\theta}_s, \vec{\theta}_n ; D(f)\}$

<u>Corresponding "architecture"</u>:

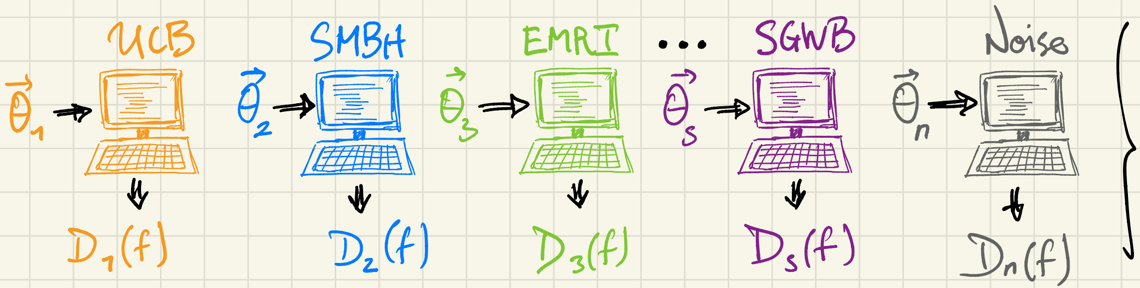
a)



full posterior

<u>Outputs</u>: $p(\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \dots, \vec{\theta}_s, \vec{\theta}_n \,|\, D(f))$

SBI for GWB Physics



$$D(f) = \sum_k D_k(f)$$

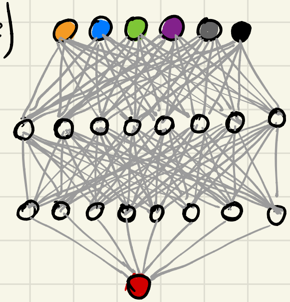
Possible inputs:

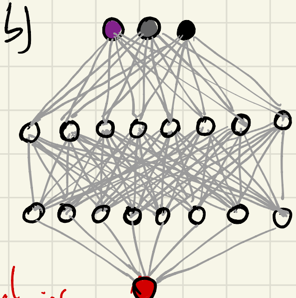
a) $\{\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \ldots, \vec{\theta}_s, \vec{\theta}_n ; D(f)\}$

b) $\{\vec{\theta}_s, \vec{\theta}_n ; D(f)\}$

⋮

Corresponding "architecture":

a) 

b) 

...

Outputs:

full posterior: $p(\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \ldots, \vec{\theta}_s, \vec{\theta}_n \mid D(f))$

marginal posterior: $p(\vec{\theta}_s, \vec{\theta}_n \mid D(f))$

SBI for GWB physics

Features

SBI for GWB physics

Features

- Parallel training for different (marginal) posteriors

SBI for GWB physics

Features

- Parallel training for different (marginal) posteriors
- Scalability with the dimension of the full param. space

  (nuisance parameters automatically marginalised in the procedure)

SBI for GWB physics

Features

- Parallel training for different (marginal) posteriors
- Scalability with the dimension of the full param. space

  (nuisance parameters automatically marginalised in the procedure)

- No synchronization overheads

SBI for GWB Physics

Features

- Parallel training for different (marginal) posteriors
- Scalability with the dimension of the full param. space (nuisance parameters automatically marginalised in the procedure)
- No synchronization overheads
- Pre-trained (a.k.a. "amortized") strategy { No need for retraining upon new data arrival!

SBI for GWB physics

Features

- Parallel training for different (marginal) posteriors
- Scalability with the dimension of the full param. space

  (nuisance parameters automatically marginalised in the procedure)
- No synchronization overheads
- Pre-trained (a.k.a. "amortized") strategy { No need for retraining upon new data arrival!

Challenges

- Architecture complexity / Training cost trade-off
- Careful validation of the inference results

A case study:

Fast likelihood-free reconstruction of gravitational wave backgrounds

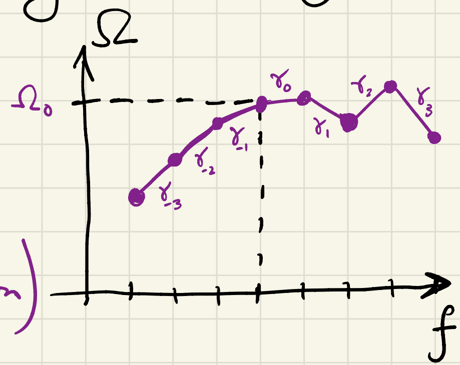
Androniki Dimitriou (Valencia U., IFIC), Daniel G. Figueroa (Valencia U., IFIC), Bryan Zaldivar (Valencia U., IFIC) (Sep 15, 2023)

Published in: *JCAP* 09 (2024) 032 • e-Print: 2309.08430 [astro-ph.IM]

- Blind (and template) reconstruction of SGWB, potentially including foregrounds

- Blind (and template) reconstruction of SGWB, potentially including foregrounds

- SGWB (agnostic parametrisation)

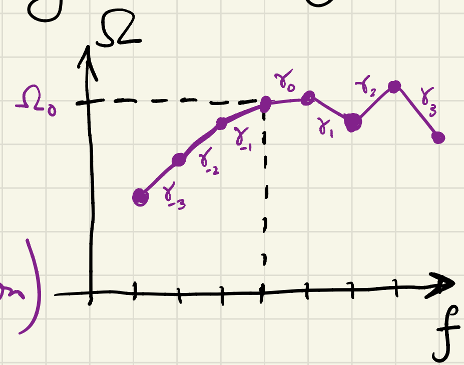


Piecewise power-law (1 amplitud + 27 slopes)

$\{\Omega_0, \gamma_0, \gamma_1, \gamma_{-1}, \gamma_2, \gamma_{-2}, \dots\}$

- Blind (and template) reconstruction of SGWB, potentially including foregrounds

- SGWB (agnostic parametrisation)



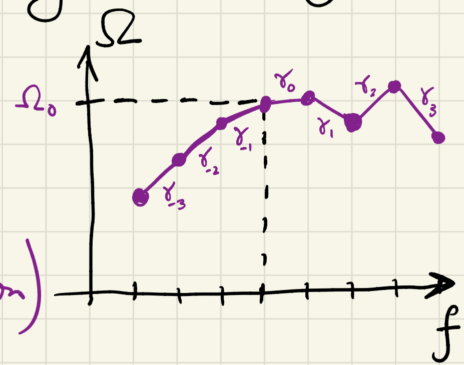
Piecewise power-law (1 amplitud + 27 slopes)

$\{\Omega_0, \gamma_0, \gamma_1, \gamma_{-1}, \gamma_2, \gamma_{-2}, \dots\}$

- Foregrounds

- Blind (and template) reconstruction of SGWB, potentially including foregrounds

- SGWB (agnostic parametrisation)



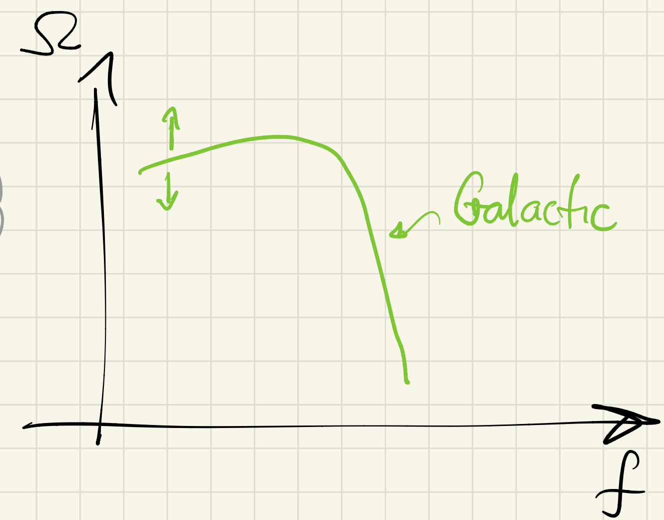
Piecewise power-law (1 amplitud + 27 slopes)

$\{\Omega_0, \gamma_0, \gamma_1, \gamma_{-1}, \gamma_2, \gamma_{-2}, \dots\}$

- Foregrounds

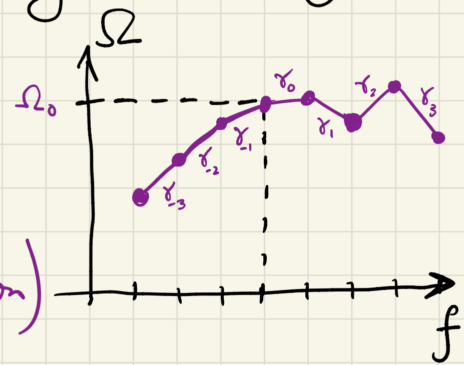
1) Galactic origin
[binary mergers]

Fixed shape, Free amplitud
average isotropic
& stationary signal



- Blind (and template) reconstruction of SGWB, potentially including foregrounds

- SGWB (agnostic parametrisation)



Piecewise power-law (1 amplitud + 27 slopes)

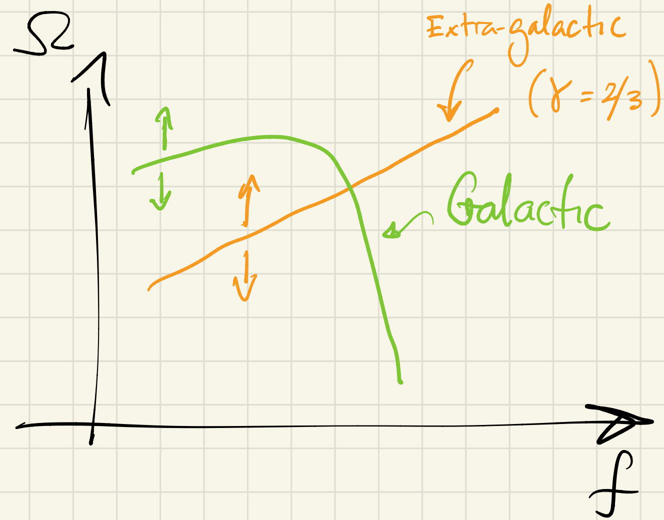
$\{\Omega_0, \gamma_0, \gamma_1, \gamma_{-1}, \gamma_2, \gamma_{-2}, \dots\}$

- Foregrounds

1) Galactic origin [binary mergers]

Fixed shape, Free amplitud
average isotropic
& stationary signal

2) Extra-galactic [BH & neutron star binaries]



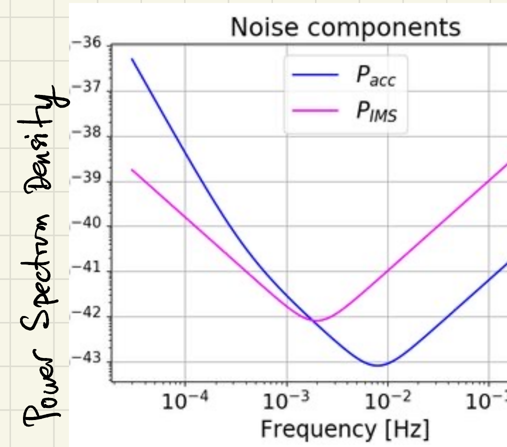
- **Noise:** TDI channels $X, Y, Z \rightarrow A, E, T$ basis

(uncorrelated data streams)

- **Noise:** TDI channels X, Y, Z $\Rightarrow$ A, E, T basis
(uncorrelated data streams)

Effective noise modelling with 2 contributions

- $P_{acc}(f; A_{acc})$
- $P_{IMS}(f; A_P)$



- **Noise:** TDI channels $X, Y, Z \rightarrow A, E, T$ basis

(uncorrelated data streams)

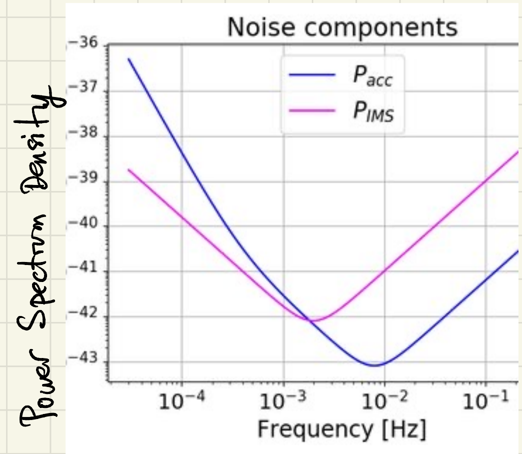
Effective noise modelling with 2 contributions

- $P_{acc}(f; A_{acc})$ • $P_{IMS}(f; A_p)$

$$\underbrace{\qquad\qquad\qquad\qquad}$$

$N_{EE} = N_{AA}(f; A_{acc}, A_p)$

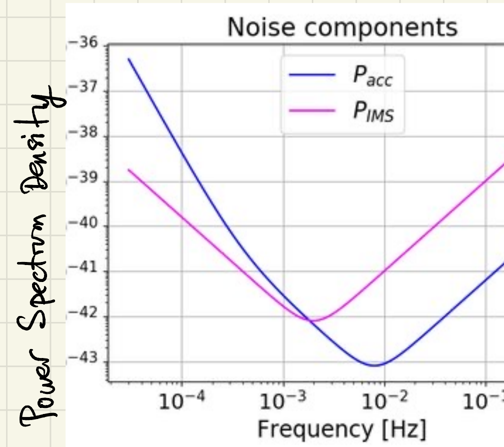
$N_{TT}(f; A_{acc}, A_p)$



- **Noise:** TDI channels $X, Y, Z \rightarrow A, E, T$ basis
(uncorrelated data streams)

Effective noise modelling with 2 contributions

- $P_{acc}(f; A_{acc})$
- $P_{IMS}(f; A_P)$



$$\underbrace{P_{acc}(f; A_{acc}) \quad P_{IMS}(f; A_P)}$$

$$N_{EE} = N_{AA}(f; A_{acc}, A_P)$$

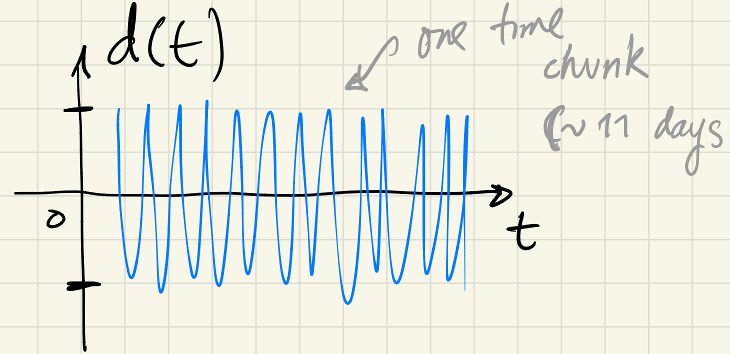
$$N_{TT}(f; A_{acc}, A_P)$$

TT channel only sensitive to noise for $f \lesssim 0.02\,\text{Hz}$

AA, EE channels: both noise and signals present

DATA SIMULATION

- Data stream



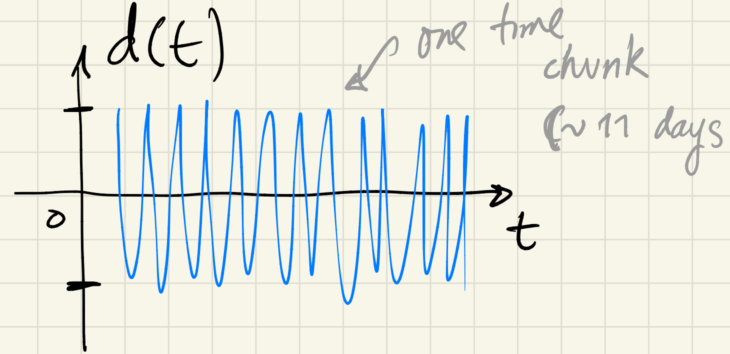
DATA SIMULATION

- Data stream

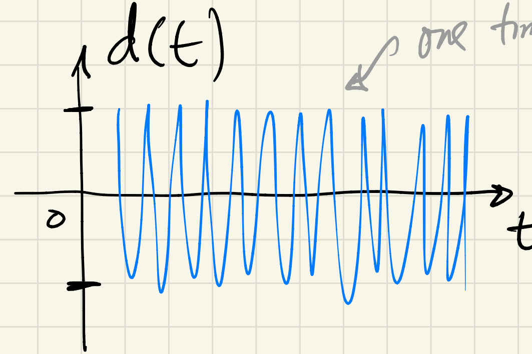
$d(t) \to d(f) \in \mathbb{C}$

$\mathrm{Re}[d(f)]$ $\mathrm{Im}[d(f)]$ indep random variables with variance $\frac{1}{2} D(f)$

↳ a realization of the power spectrum



DATA SIMULATION



one time chunk (~11 days)

- Data stream

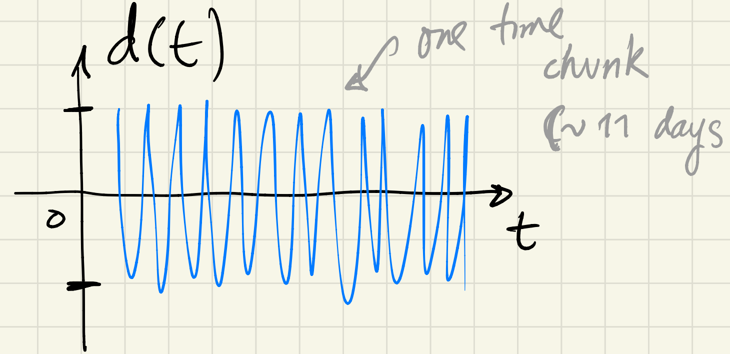
  $d(t) \to d(f) \in \mathbb{C}$

  $\mathrm{Re}[d(f)]$ $\mathrm{Im}[d(f)]$ indep random variables with variance $\frac{1}{2}D(f)$

  ↳ a realization of the power spectrum

- $D(f)$

DATA SIMULATION



- Data stream

$d(t) \to d(f) \in \mathbb{C}$

$\mathrm{Re}[d(f)]$ $\quad$ $\mathrm{Im}[d(f)]$ indep random variables with variance $\frac{1}{2} D(f)$

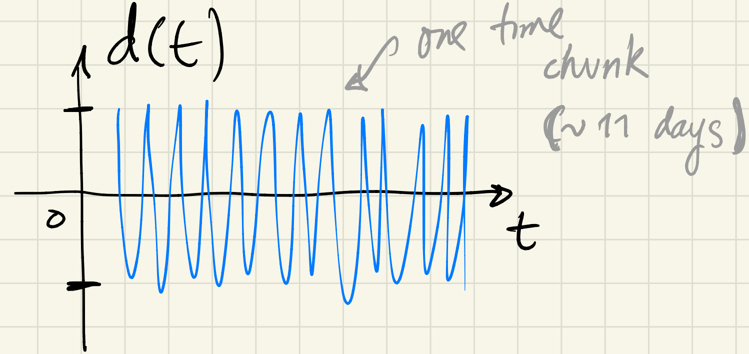
↳ a realization of the power spectrum

- $D(f) \Rightarrow D_i^{\alpha\beta}(f_j)$

channel

↳ time chunk

DATA SIMULATION



- Data stream

$d(t) \to d(f) \in \mathbb{C}$

$\mathrm{Re}[d(f)]$ $\quad$ $\mathrm{Im}[d(f)]$ indep random variables with variance $\frac{1}{2} D(f)$

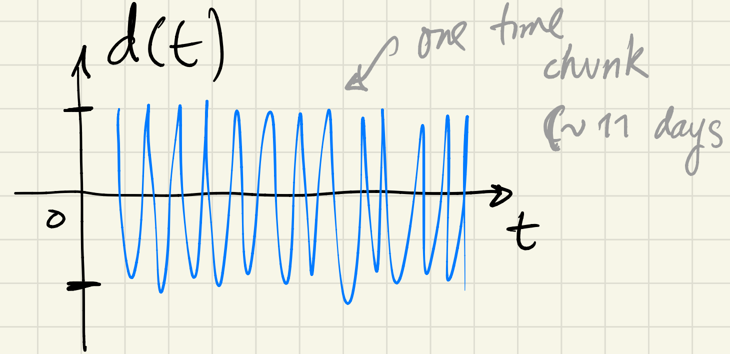
a realization of the power spectrum

- $D(f) \Rightarrow D_i^{\alpha\beta}(f_j) \Rightarrow D_{i,j}^{\alpha\beta} = S_{i,j} + N_{i,j}^{\alpha\beta}$

channel

time chunk

DATA SIMULATION



- Data stream

$d(t) \to d(f) \in \mathbb{C}$

$\mathrm{Re}[d(f)]$ $\quad$ $\mathrm{Im}[d(f)]$ $\quad$ indep random variables with variance $\frac{1}{2} D(f)$

a realization of the power spectrum

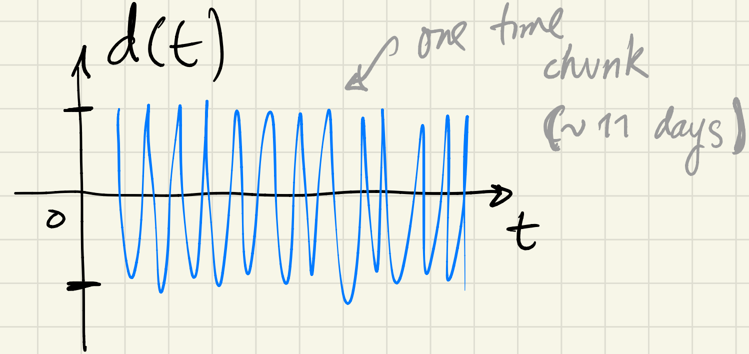
- $D(f) \to D_i^{\alpha\beta}(f_j) \to D_{i,j}^{\alpha\beta} = S_{i,j} + N_{i,j}^{\alpha\beta}$

channel ($\alpha\beta$); time chunk (i)

$$S_{i,j} = \frac{1}{2} \left| G_1(f_j) + i\, G_2(f_j) \right|^2 \qquad N_{i,j}^{\alpha\beta} = \frac{1}{2} \left| G_3(f_j) + i\, G_4(f_j) \right|^2$$

inherited from the complex-valued nature of $d(f)$

DATA SIMULATION



- Data stream

$$d(t) \to d(f) \in \mathbb{C}$$

$\mathrm{Re}[d(f)]$ $\quad$ $\mathrm{Im}[d(f)]$ $\quad$ indep random variables with variance $\frac{1}{2}D(f)$

a realization of the power spectrum

- $D(f) \Rightarrow D_i^{\alpha\beta}(f_j) \Rightarrow D_{i,j}^{\alpha\beta} = S_{i,j} + N_{i,j}^{\alpha\beta}$

channel

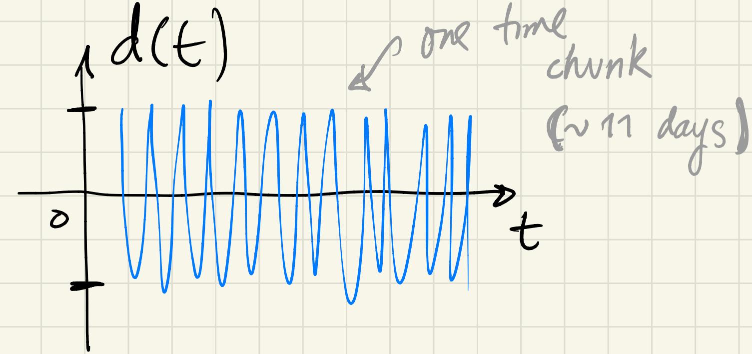
time chunk

$$S_{i,j} = \frac{1}{2}\left| G_1(f_j) + i\, G_2(f_j) \right|^2 \qquad N_{i,j}^{\alpha\beta} = \frac{1}{2}\left| G_3(f_j) + i\, G_4(f_j) \right|^2$$

inherited from the complex-valued nature of $d(f)$

Latent variables
$$\begin{cases} G_{1,2}(f_j) \sim \mathcal{N}\left(0, \Omega_{GW}(f_j)\right) \\ G_{3,4}(f_j) \sim \mathcal{N}\left(0, \Omega^{\alpha\beta}_{noise}(f_j)\right) \end{cases}$$

DATA SIMULATION



- Data stream

$d(t) \to d(f) \in \mathbb{C}$

$Re[d(f)]$ $Im[d(f)]$ indep random variables with variance $\frac{1}{2} D(f)$

↳ a realization of the power spectrum

channel

- $D(f) \Rightarrow D_i^{\alpha\beta}(f_j) \Rightarrow D_{i,j}^{\alpha\beta} = S_{i,j} + N_{i,j}^{\alpha\beta}$

↳ time chunk

$$S_{i,j} = \frac{1}{2} \left| G_1(f_j) + i\, G_2(f_j) \right|^2 \qquad N_{i,j}^{\alpha\beta} = \frac{1}{2} \left| G_3(f_j) + i\, G_4(f_j) \right|^2$$

inherited from the complex-valued nature of $d(f)$

Latent variables
$$\begin{cases} G_{1,2}(f_j) \sim \mathcal{N}(0, \Omega_{GW}(f_j)) \\ G_{3,4}(f_j) \sim \mathcal{N}(0, \Omega^{\alpha\beta}_{noise}(f_j)) \end{cases}$$

$$\bar{D}_j^{\alpha\beta} = \frac{1}{N_c} \sum_{i=1}^{N_c} D_{i,j}^{\alpha\beta}$$

D_j^{TT}

D_j^{AA}

SETUP SUMMARY

- Data generated in the TT & AA channels (frequency domain, time-averaged)

SETUP SUMMARY

- Data generated in the TT & AA channels (frequency domain, time-averaged)
- $\Omega_{GW}(f)$ characterised by piecewise power-laws (fixed binning configuration)

  → For "blind" reconstruction

SETUP SUMMARY

- Data generated in the TT & AA channels (frequency domain, time-averaged)
- $\Omega_{GW}(f)$ characterised by piecewise power-laws (fixed binning configuration)

  → For "blind" reconstruction
- Case studies with/without foregrounds

SETUP SUMMARY

- Data generated in the TT & AA channels (frequency domain, time-averaged)

→ For "blind" reconstruction

- $\Omega_{GW}(f)$ characterised by piecewise power-laws (fixed binning configuration)
- Case studies with/without foregrounds
- Case study with template parameter reconstruction

SETUP SUMMARY

- Data generated in the TT & AA channels (frequency domain, time-averaged)
- $\Omega_{GW}(f)$ characterised by piecewise power-laws (fixed binning configuration)
  → For "blind" reconstruction
- Case studies with/without foregrounds
- Case study with template parameter reconstruction
- Inference using Neural Posterior Estimation

SETUP SUMMARY

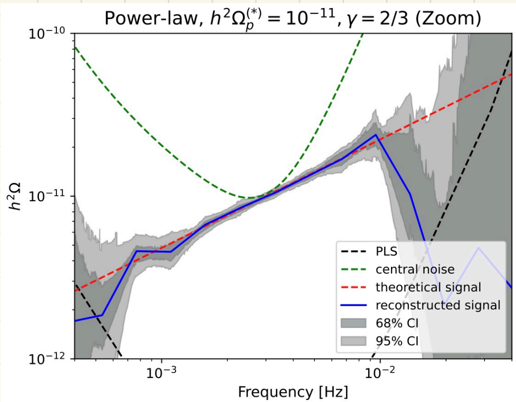
- Data generated in the TT & AA channels (frequency domain, time-averaged)
- $\Omega_{GW}(f)$ characterised by piecewise power-laws (fixed binning configuration) → for "blind" reconstruction
- Case studies with/without foregrounds
- Case study with template parameter reconstruction
- Inference using Neural Posterior Estimation
- $\mathcal{O}(10^6)$ simulations used to train the algorithms

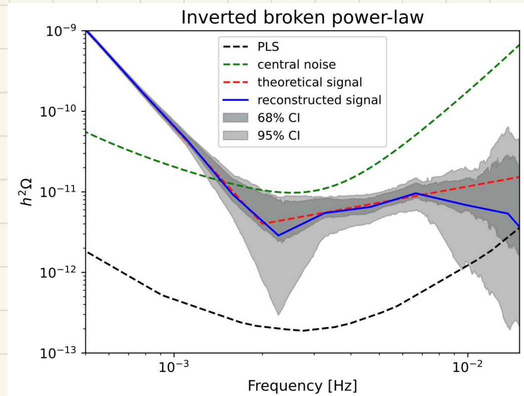
SETUP SUMMARY

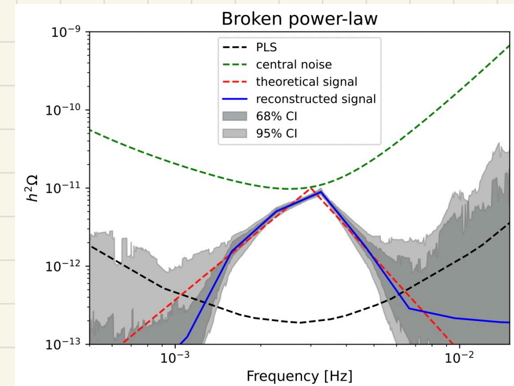
- Data generated in the TT & AA channels (frequency domain, time-averaged)
- $\Omega_{GW}(f)$ characterised by piecewise power-laws (fixed binning configuration) → for "blind" reconstruction
- Case studies with/without foregrounds
- Case study with template parameter reconstruction
- Inference using Neural Posterior Estimation
- $\mathcal{O}(10^6)$ simulations used to train the algorithms
- Public package "GWBackFinder"

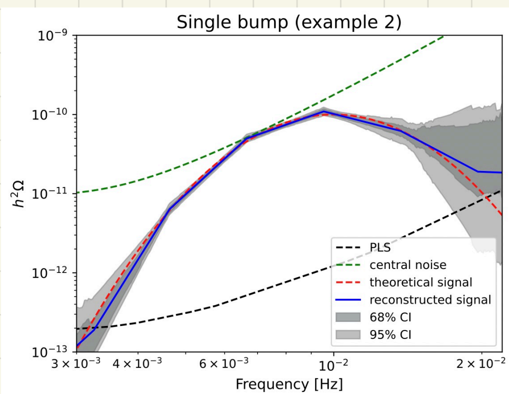
Results

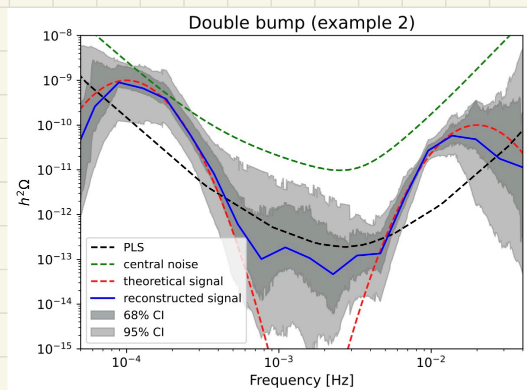
BLIND SIGNAL RECONSTRUCTION (showing only the most challenging cases)

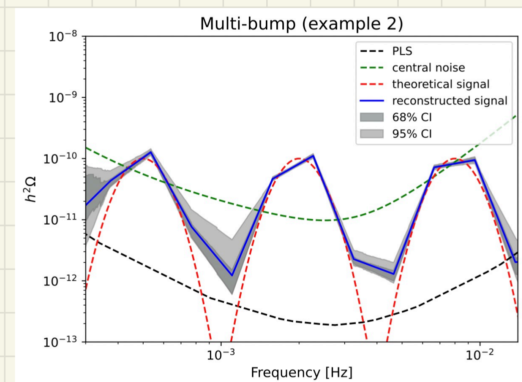




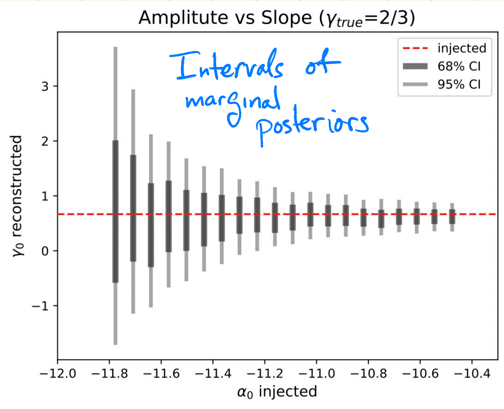


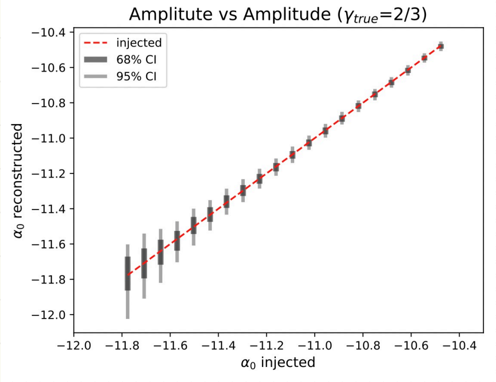




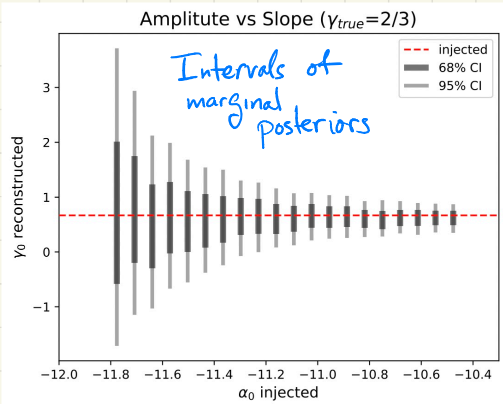


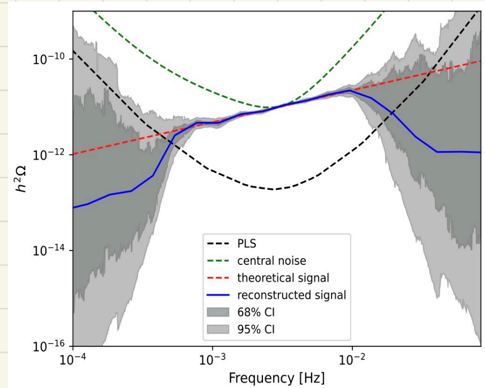
FIGS OF MERIT FOR RECONSTRUCTION QUALITY

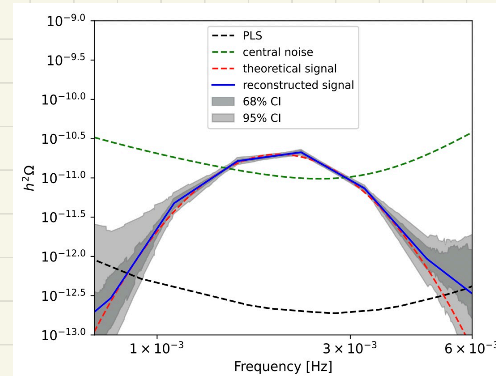


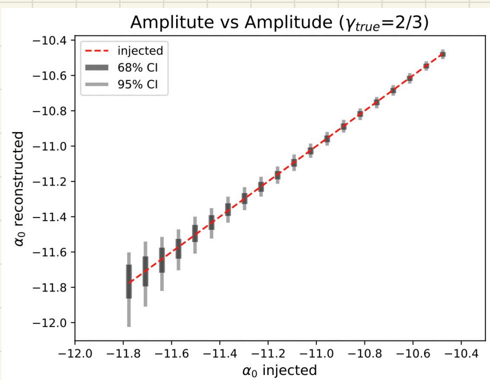


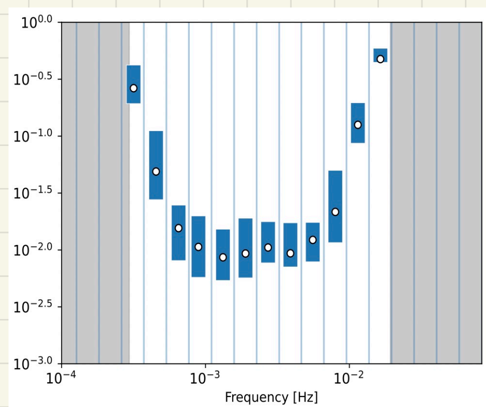
FIGS OF MERIT FOR RECONSTRUCTION QUALITY



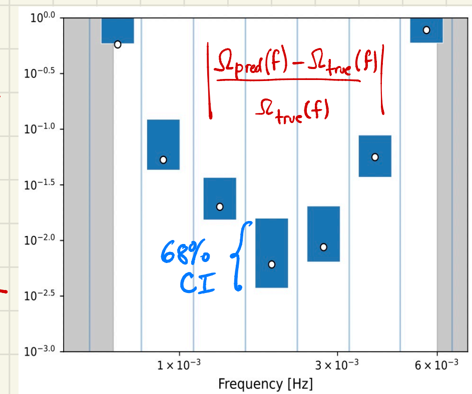






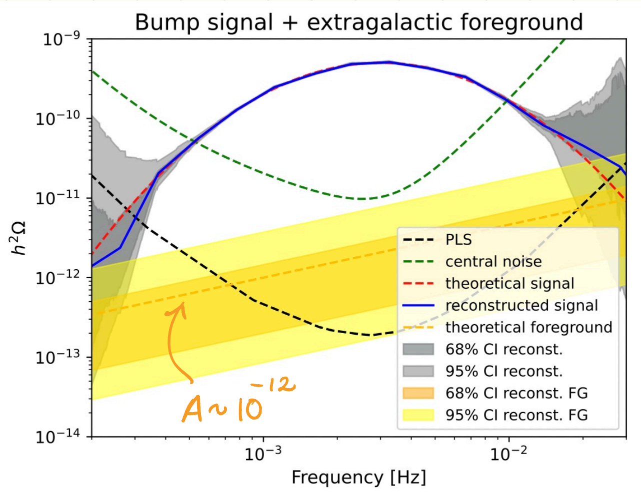


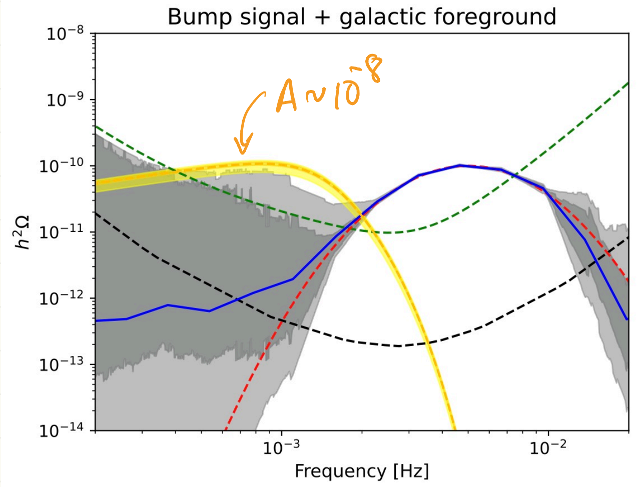
|Relative error|



RECONSTRUCTION IN PRESENCE OF FOREGROUNDS

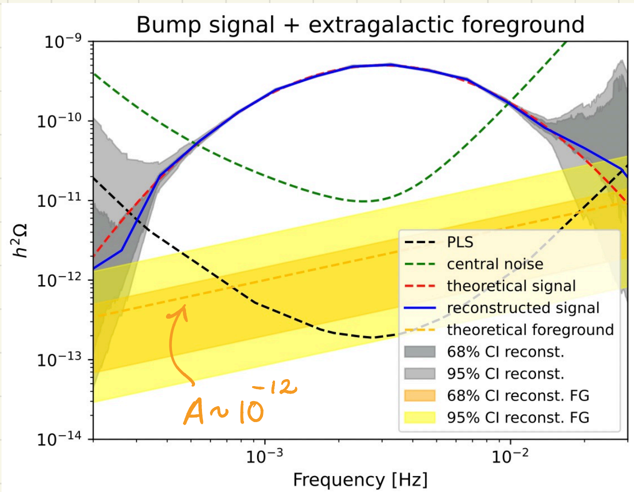
(Blind rec.)

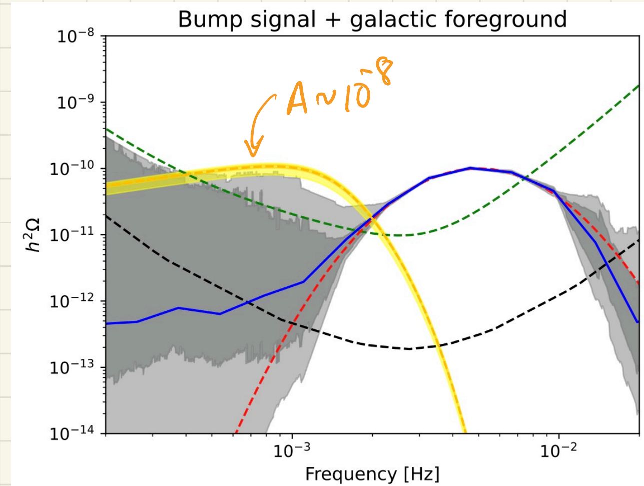




RECONSTRUCTION IN PRESENCE OF FOREGROUNDS

(Blind rec.)

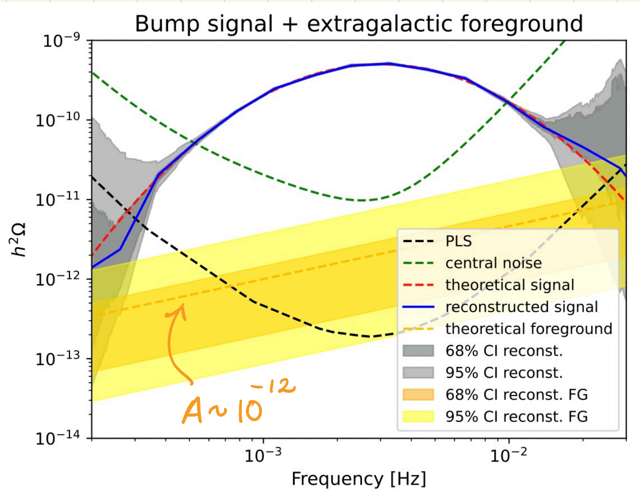


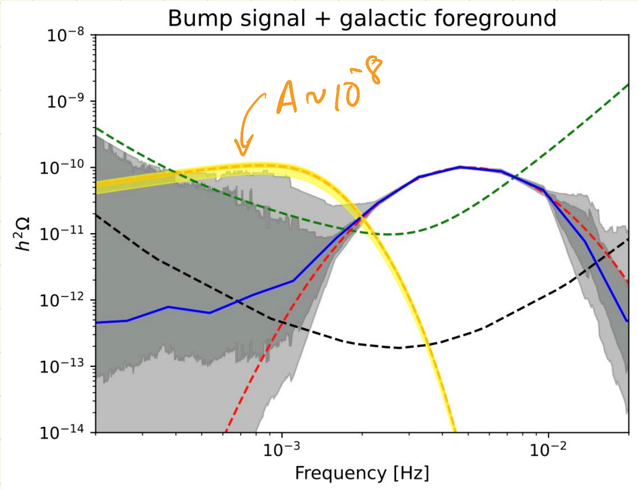


- Signal is precisely reconstructed when foreground is subdominant

RECONSTRUCTION IN PRESENCE OF FOREGROUNDS

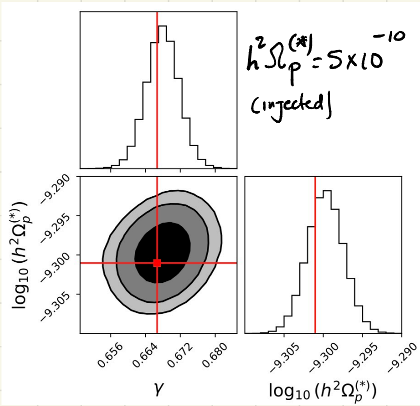
(Blind rec.)

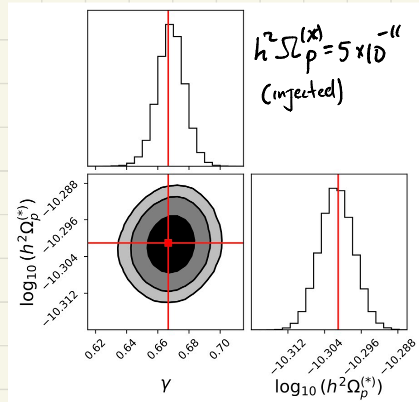


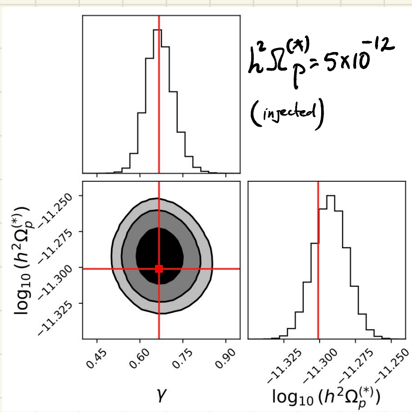


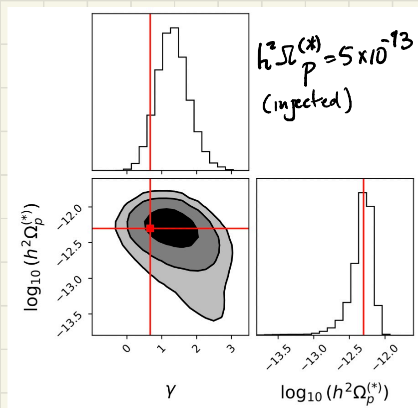
- Signal is precisely reconstructed when foreground is subdominant
- Challenging Background/foreground degeneracies
  (but better if foreground model is more realistic!)

TEMPLATE RECONSTRUCTION

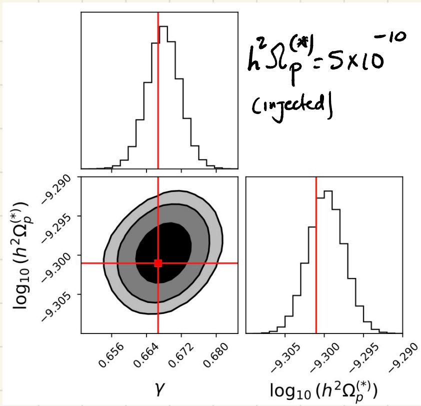


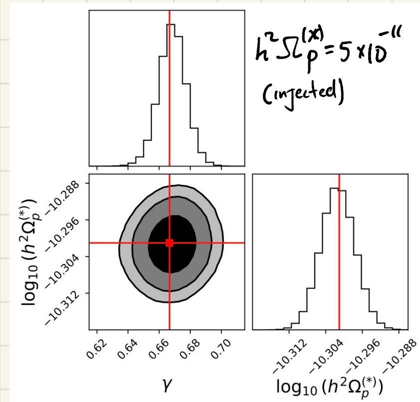


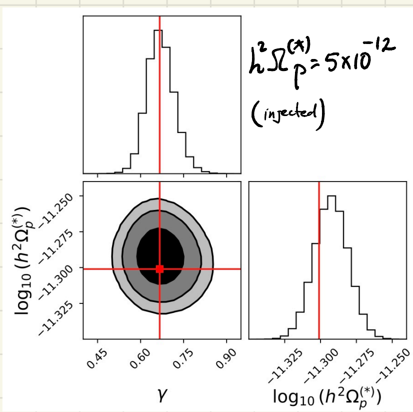


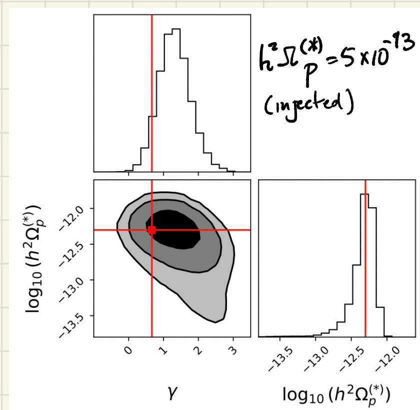


TEMPLATE RECONSTRUCTION



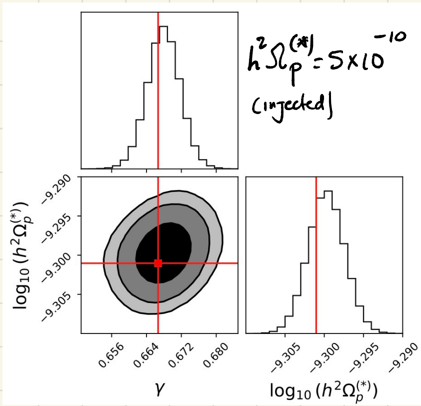


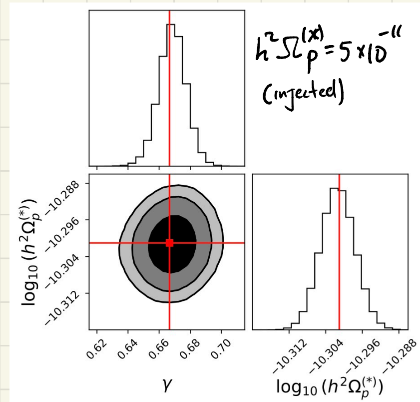


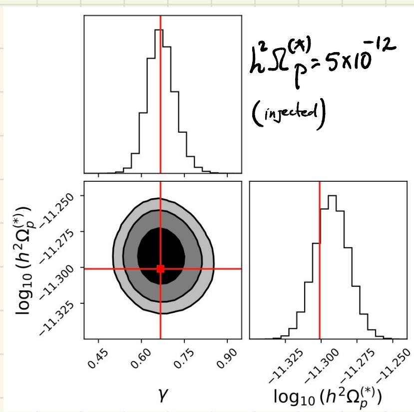


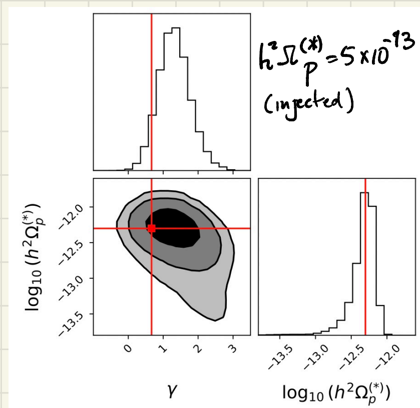
* $\mathcal{O}(10)$ times less # of simulations to train (w.r.t. blind rec.)

TEMPLATE RECONSTRUCTION





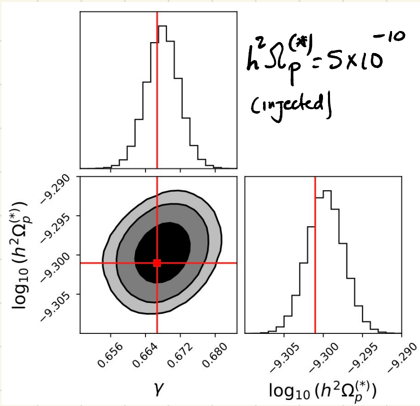


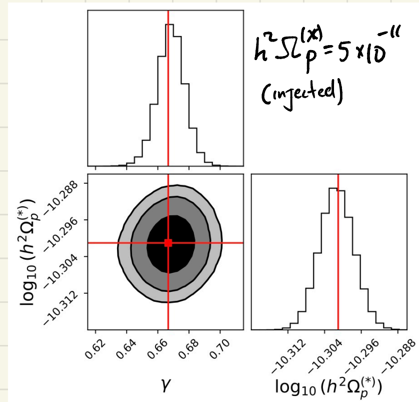


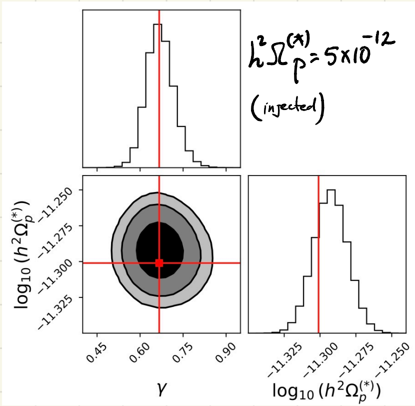
* $\mathcal{O}(10)$ times less # of simulations to train (w.r.t. blind rec.)

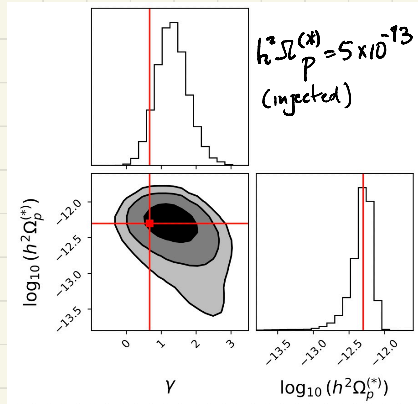
* In general better precision w.r.t. blind reconstruction

TEMPLATE RECONSTRUCTION









* $\mathcal{O}(10)$ times less # of simulations to train (w.r.t. blind rec.)

* In general better precision w.r.t. blind reconstruction

* Succesful but less precise reconstruction for $(h^2\Omega_p^{(*)})_{\text{injected}} \sim \mathcal{O}(10^{-13})$

CONCLUSIONS & OUTLOOK

- SBI approach to the Global Fit seems very promising (computational advantages w.r.t. MCMC)
- SGWB already leveraging SBI (this work, see also talk by Mauro)
- Application of our technique to specific template reconstruction (Cosmic Strings, out very soon)
- Key improvements in the procedure (binning, data representation, model selection, etc)

Thanks!