Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Bayesian inference for the nHz SGWB in PTA data analysis with Machine Learning

in collaboration with Golam Shaifullah, Andrea Possenti, Carmelita Carbone and FastParam team@Koexai, Massimo Meneghetti

Spoke 3 III Technical Workshop, Perugia May 26 -29, 2025

Scientific Rationale

- **Pulsar Timing Arrays collaborations reported evidence for the presence of a stochastic Gravitational Wave Background (SGWB) at nHz frequencies**
  - **- constrain properties of the astrophysical sources**
  - **- probe potential cosmological sources**
- **Current EPTA dataset: 25 pulsars**
- **Future IPTA dataset: 100+ pulsars**

SPEEDING UP BAYESIAN INFERENCE!

FOCUS ON NESTED SAMPLER WITH ML: NESSAI

NESSAI in a nutshell

$$p(\vartheta, d) = \frac{\mathcal{L}(d,\vartheta)\pi(\vartheta)}{\int \mathcal{L}(d,\vartheta)\, \pi(\vartheta) d\vartheta} \qquad Z = \int \mathcal{L}(d,\vartheta)\, \pi(\vartheta) d\vartheta = \int \frac{\mathcal{L}(d,\vartheta)\pi(\vartheta)}{Q(\vartheta)} Q(\vartheta) d\vartheta$$

WHERE (GENERATIVE) ML COMES IN: NORMALIZING FLOWS

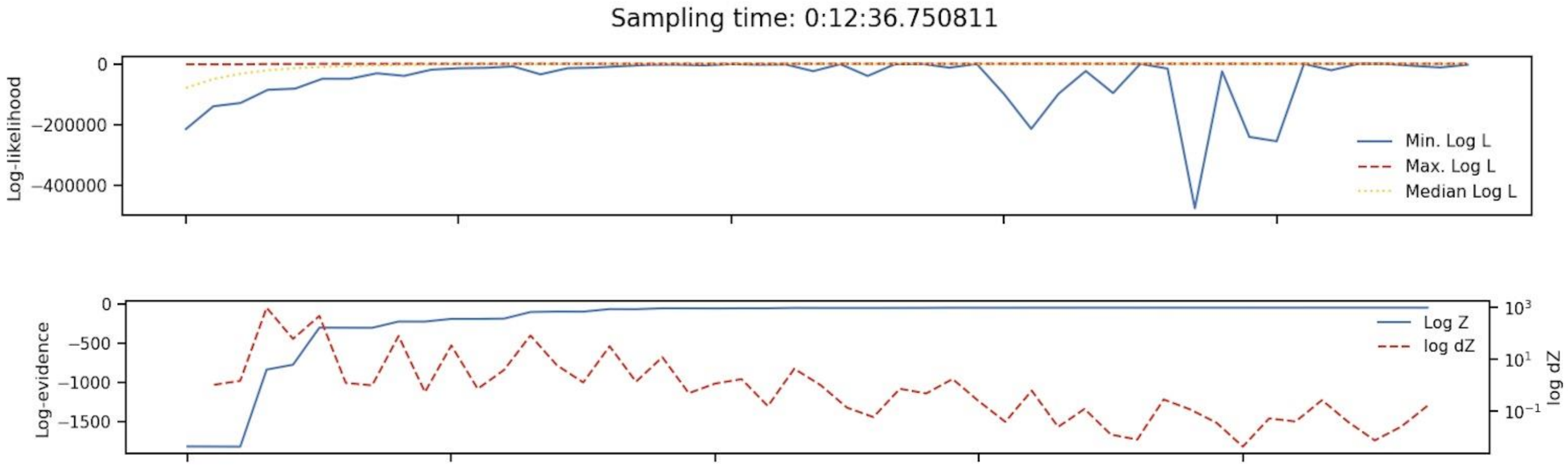
The normalizing flow f learns the distribution of a set of live points via mapping from an auxiliary known simple distribution q

$$Q(\vartheta) = q\big(f(\vartheta)\big)|det\, J|$$

- **Explicit expression of the learnt distribution → normalized**
- **Convenient when training cost << sampling computational cost**

Previously on PTA with NESSAI…

Testing the 12-dim Rosenbrock likelihood, coupling layers RNVP NF, PyTorch parallelization



😊 **It works!**

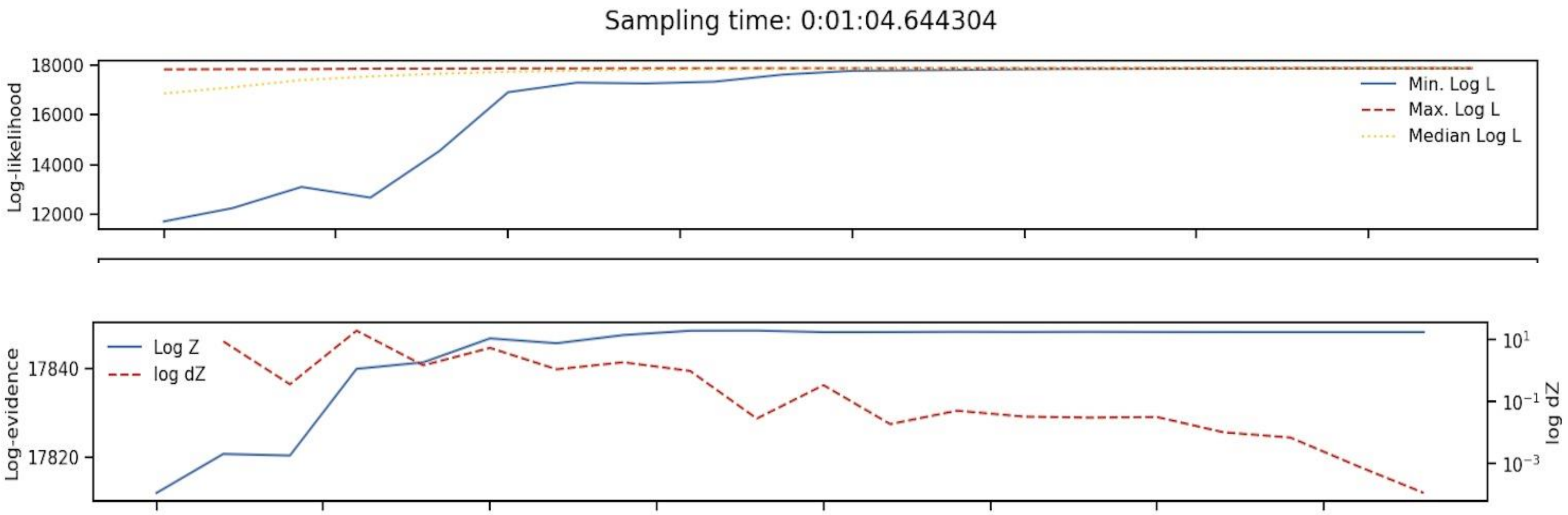
😕 **Fluctuations in the likelihood**

- ➢ **Configure user-definend parallelization**
- ➢ **Play with the NF neural network**

HARDWARE:
ASUS ZenBook
32 GB RAM
CPU Intel Core i9 (20-core) 8 GB RAM

Speeding up Bayesian inference for the PTA data analysis: ENTERPRISE + NESSAI

1 pulsar with 5 parameters

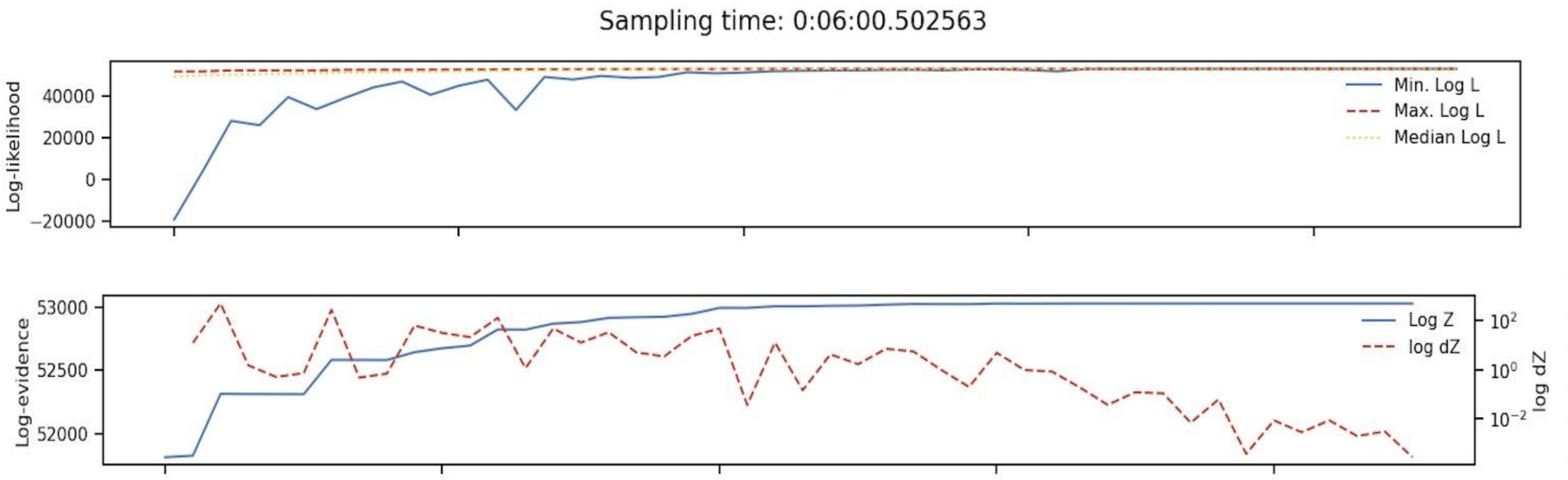


😊 **1min4s with Nessai vs ~30min with PTMCMC**

HARDWARE:
ASUS ZenBook
32 GB RAM
CPU Intel Core i9 (20-core) 8 GB RAM

Speeding up Bayesian inference for the PTA data analysis: ENTERPRISE + NESSAI

3 pulsars → 15 parameters in total

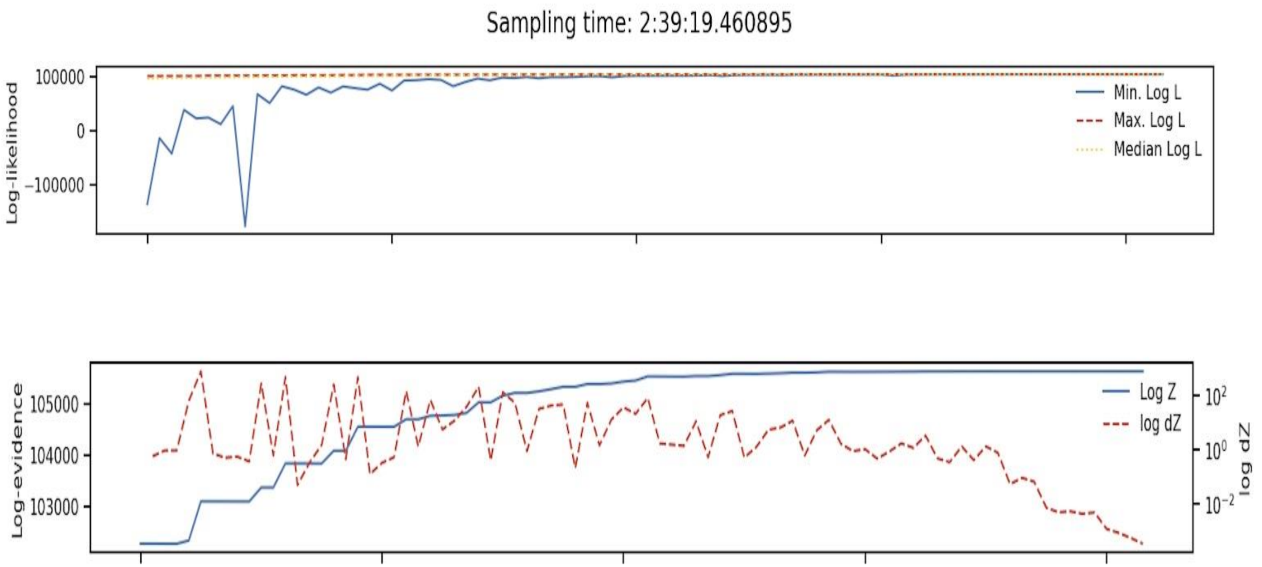


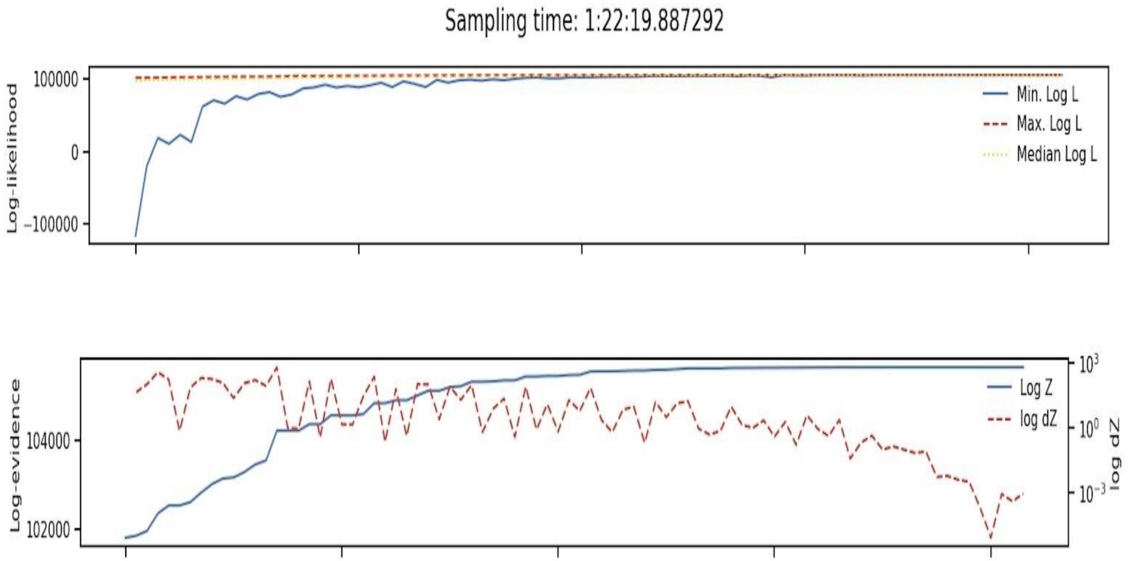
😊 **6min with Nessai vs ~120min with PTMCMC**

HARDWARE:
ASUS ZenBook
32 GB RAM
CPU Intel Core i9 (20-core) 8 GB RAM

Speeding up Bayesian inference for the PTA data analysis: ENTERPRISE + NESSAI

6 pulsars → 30 parameters in total





Nessai vs Nessai: ~ a factor of 2 gained in time by optimizing PyTorch configuration

HARDWARE:
ASUS ZenBook
32 GB RAM
CPU Intel Core i9 (20-core) 8 GB RAM

Currently on PTA with NESSAI…

- The speeding up is fully driven by the Pytorch and neural network configuration
  - ➢ boost their efficiency

- The gain in time by using Nessai is remarkable
  - ➢ address some shortcomings of the PTA data analysis

Currently on PTA with NESSAI part 1: push the speed up

- Integration of the automatic hyperparameter optimization framework OPTUNA
  - ➢ State-of-the-art and open source framework, particularly designed for machine learning
- Enable tuning of the 5 hyperparameters of Nessai

✓ Code ENTERPRISE + NESSAI + OPTUNA on GitHub

✓ Verified consistent performance across multiple pulsars

✓ Benchmarked stability over repeated runs

Currently on PTA with NESSAI part 2: addressing circular analysis

Circular analysis consists in using single-pulsar noise analysis results from model selection in the subsequent analysis of the whole PTA of the same dataset

PROBLEM: some noise components may not be detectable in a single-pulsar analysis due to low SNR

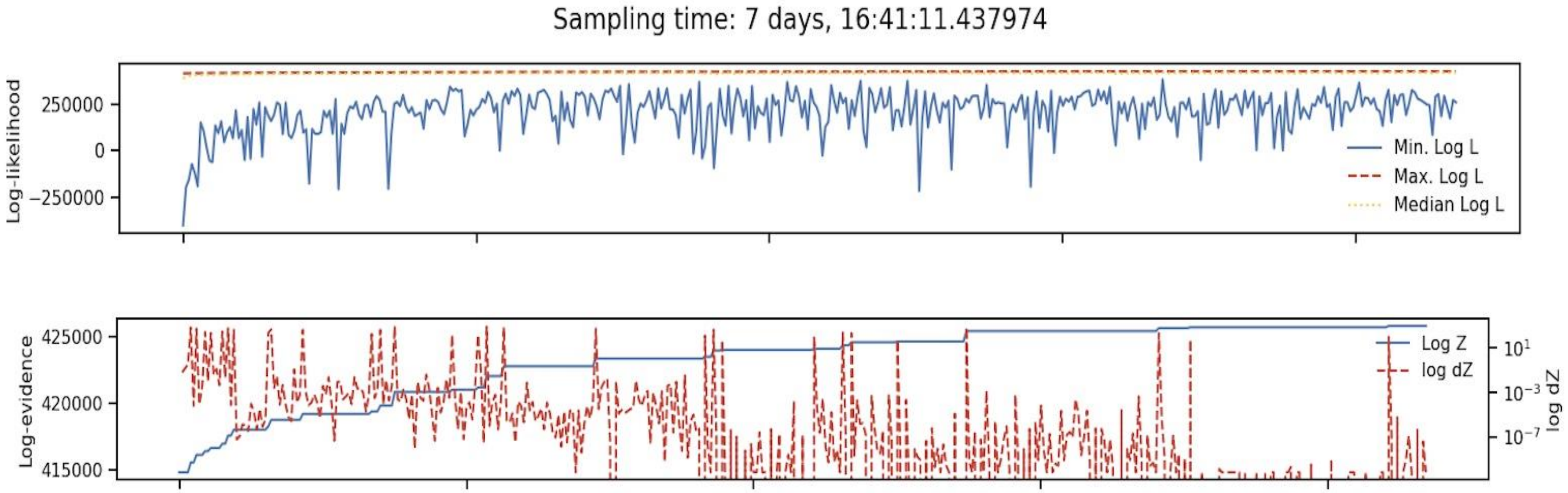
BUT.. this does not mean that they are not present! This is crucial e.g. for the intrinsic red noise, i.e. a low-frequency noise process modeled with a power law, just as the SGWB..

SO, if detected in the whole PTA but was set to zero by hand a priori, it is misinterpreted as SGWB!!

SOLUTION: include all relevant noise components and the SGWB consistently in the joint PTA analysis without relying on prior single pulsar model selection

Currently on PTA with NESSAI part 2: addressing circular analysis

EPTADR2: 25 pulsars with 5 noise parameters each → 100 parameters in total



😊 **Feasible with Nessai**

- ➢ **Can be further optimized (tests are running this week)**
- ➢ **Next step: add the common SGWB (tests starting next week)**

HARDWARE:
Cluster @unimib
376 GB RAM
CPU: 2x Intel Xeon Silver 4214 @2.20GHz
24 physical cores x 2 threads = 48 cores

Currently on PTA with NESSAI part 3: addressing noise prior dependence with FastParam team@Koexai

CURRENT PRACTICE: each pulsars red noise parameters (amplitude and spectral index) are modeled independently using uninformative uniform prior

BUT.. modeling the red noise this way may fail to capture the true underlying distributions of noise parameters across the PTA, of which each pulsar is a realization

AND we know that red noise is highly correlated with the SGWB..

SO the misspecification of red noise prior may introduce systematics and bias the inference of the SGWB

SOLUTION: account for how parameters governing pulsar noise are distributed across the PTA, i.e. parametrize the prior distributions. This is known as HIERARCHICAL BAYESIAN INFERENCE

Currently on PTA with NESSAI part 3: addressing prior dependence with FastParam team@Koexai

THE HYPERPRIOR PARAMETERS REPRESENT HOW RED NOISE IS DISTRIBUTED ACROSS PULSARS AND THEY CORRELATE WITH THE SGWB. HOW TO DEAL WITH THEM?

1. **Set hyperparameters to some fixed value based on some a priori knowledge**
   - ➢ **This is circular analysis**

2. **Marginalize over hyperparameters**
   - ➢ **Results for the SGWB account for the uncertainties in the red noise distributions but are sensitive to the shape you choose for the hyperprior**

arXiv: 2409.03661 and arXiv: 2409.03627

3. **Reduce hyperprior sensitivity by decorrelating the hyperparameters from SGWB via a ML-assisted transformation in the parameter space**

Currently on PTA with NESSAI part 3: addressing prior dependence with FastParam team@Koexai

PROS

- Leads to robust and reliable inference
- Should be faster than marginalization

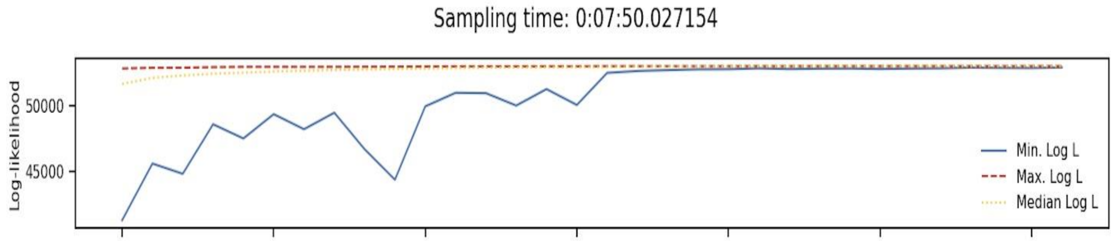
CONS

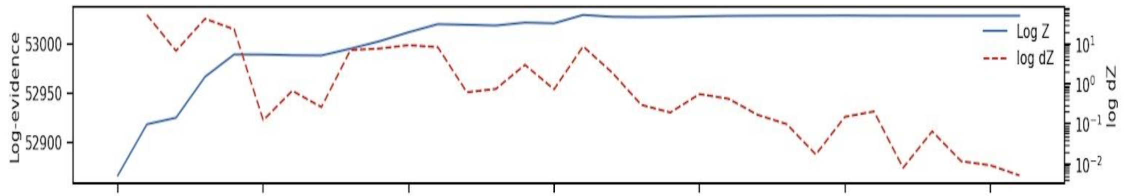
- Requires careful implementation of the transformation
- Reduction of hyperprior sensitivity depends on the specific case study

Currently on PTA with NESSAI part 3: addressing prior dependence with FastParam team@Koexai

WORK DONE:

- ✓ Reproduce results after the transformation for a simple case study from ArXiv: 2412.03503
- ✓ Add hyperprior on the red noise and DM variation noise models in ENTERPRISE + NESSAI
- ✓ Add the SGWB signal
- ✓ Test runs for two shapes for the hyperprior





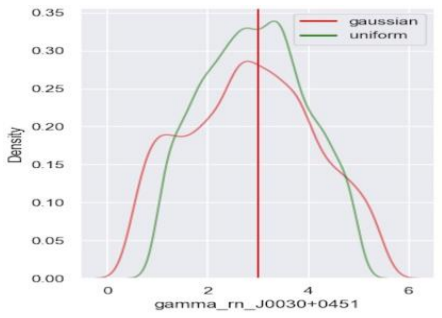
3 pulsars, each with 2 parameters x 2 noises + SGWB = 14 parameters
Gaussian hyperprior

HARDWARE:
16 GB RAM
CPU Intel Core i7 (14-core)

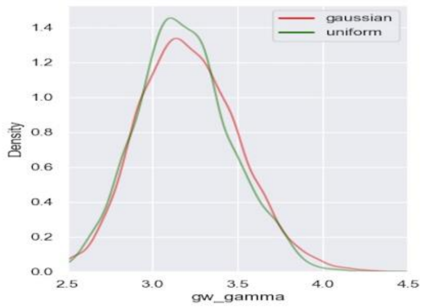
Currently on PTA with NESSAI part 3: addressing prior dependence with FastParam team@Koexai

WORK DONE:

- ✓ Check that the posterior of the two noise models is sensitive to the hyperprior shape
- ✓ Check that the posterior of the SGWB is sensitive to the hyperprior shape



Spectral index of the red noise of J0030+0451



Spectral index of the SGWB

NEXT STEPS:

- Benchmark stability of the test runs results
- Implement the transformation in the PTA parameters space

HARDWARE:
16 GB RAM
CPU Intel Core i7 (14-core)

Targets and KPI in M10

M10 INTERMEDIATE – 31/03/2025 (percentage of achievement=100%)

- code with ENTERPRISE + NESSAI + OPTUNA on GitHub

✓ ACHIEVED - KPI 3.2

M10 FINAL – 31/12/2025 (percentage of achievement=25%)

- use ENTERPRISE + NESSAI + OPTUNA to infer noise and SGWB parameters simultaneously

➢ in progress: runs for the 25 pulsars of the EPTADR2 starting next week

THANK YOU!

Back-up slides

Targets and KPI in M9: 100% in M9, 25% of the project

❖ April 24 – June 24: definition of the science case

- Incorporation of NESSAI in ENTERPRISE

✓ ACHIEVED - KPI 3.2 (i)

- Hands-on training on Neural Networks with Pytorch and nflows

✓ ACHIEVED (and in progress): 10+ hands-on courses completed - KPI 3.2 (ii)

- Test the configuration of the Normalizing Flows part of the algorithm

✓ ACHIEVED (and in progress) – KPI 3.2 (iii)

NESSAI in a nutshell

NESTED SAMPLING IDEA

$$p(\vartheta, d) = \frac{\mathcal{L}(d,\vartheta)\pi(\vartheta)}{\int \mathcal{L}(d,\vartheta)\,\pi(\vartheta)d\vartheta} \quad \rightarrow Z = \int \mathcal{L}(d,\vartheta)\,\pi(\vartheta)d\vartheta = \int_0^1 \mathcal{L}(X)dX \qquad X = \int_{\mathcal{L}>\mathcal{L}^*} \pi(\vartheta)\,d\vartheta$$

1. Standard nested sampling: sequential sampling from the likelihood-constrained prior $\pi(\vartheta)$
https://arxiv.org/pdf/2102.11056

2. Importance nested sampling: $Z = \int \frac{\mathcal{L}(d,\vartheta)\pi(\vartheta)}{Q(\vartheta)} Q(\vartheta)d\vartheta$ independent of the $\mathcal{L} > \mathcal{L}^*$ constraint
https://arxiv.org/pdf/2302.08526

NESTED SAMPLING IDEA

$$p(\vartheta, d) = \frac{\mathcal{L}(d,\vartheta)\pi(\vartheta)}{\int \mathcal{L}(d,\vartheta)\,\pi(\vartheta)d\vartheta} \quad \rightarrow Z = \int \mathcal{L}(d,\vartheta)\,\pi(\vartheta)d\vartheta = \int_0^1 \mathcal{L}(X)dX \qquad X = \int_{\mathcal{L}>\mathcal{L}^*} \pi(\vartheta)\,d\vartheta$$

Standard nested sampling: sequential sampling from the prior $\pi(\vartheta)$

1. Draw N live points ~ $\pi(\vartheta)$, calculate the $\mathcal{L}$ and choose the lowest value $\mathcal{L}^* = \mathcal{L}(\vartheta^*)$
2. Calculate the Z integral
3. Draw new points until $\mathcal{L} > \mathcal{L}^*$ and choose the lowest
4. Update the Z integral
5. Repeat until a stopping criterion is met, e.g. $\Delta\, ln\, Z < 0.1$

NESTED SAMPLING IDEA

Importance nested sampling: $Z = \int \frac{\mathcal{L}(d,\vartheta)\pi(\vartheta)}{Q(\vartheta)} Q(\vartheta) d\vartheta$

1. Draw N live points ~ $\pi(\vartheta)$ and calculate the $\mathcal{L}$
2. Use NF to get $Q(\vartheta)$
3. Calculate the Z integral
4. Repeat until a stopping criterion is met