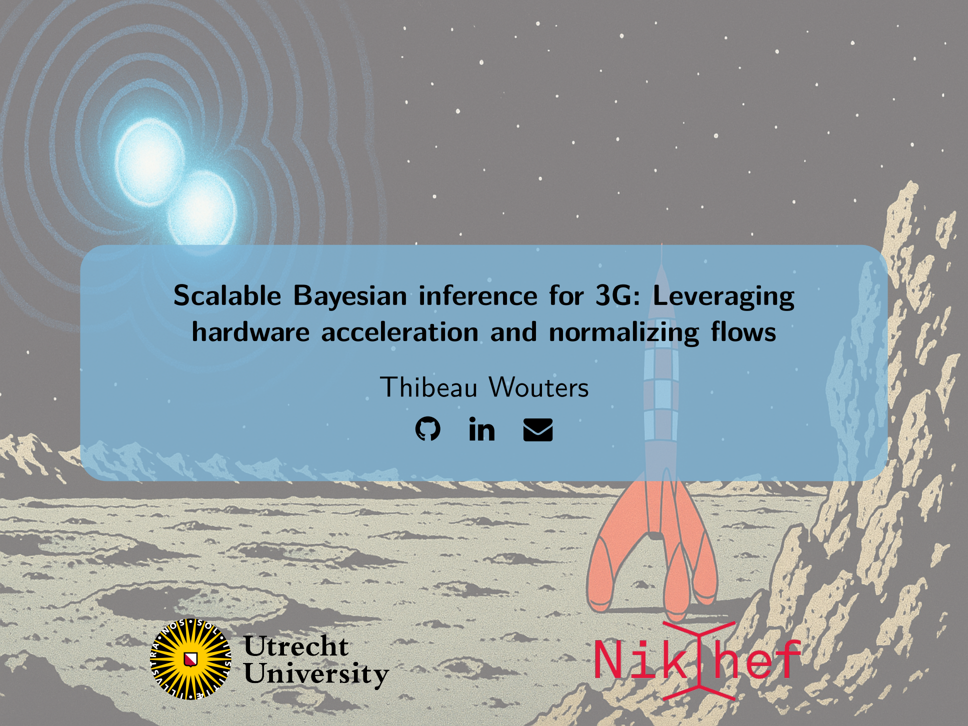


Scalable Bayesian inference for 3G: Leveraging hardware acceleration and normalizing flows

Thibeau Wouters

Utrecht University

Nikhef

Contents

Parameter estimation in 3G

Parameter estimation is done with Bayesian inference:

$$\text{posterior} \propto \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

- Sample the posterior: MCMC or nested sampling
- $\mathcal{O}(10^6) - \mathcal{O}(10^8)$ likelihood evaluations per inference

Parameter estimation in 3G

Parameter estimation is done with Bayesian inference:

$$\text{posterior} \propto \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

- Sample the posterior: MCMC or nested sampling
- $\mathcal{O}(10^6) - \mathcal{O}(10^8)$ likelihood evaluations per inference

What about 3G detectors?

- ET will observe $\mathcal{O}(10^5)$ events per year
- Signals will be longer and have higher SNRs

Parameter estimation in 3G

Parameter estimation is done with Bayesian inference:

$$\text{posterior} \propto \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

- Sample the posterior: MCMC or nested sampling
- $\mathcal{O}(10^6) - \mathcal{O}(10^8)$ likelihood evaluations per inference

What about 3G detectors?

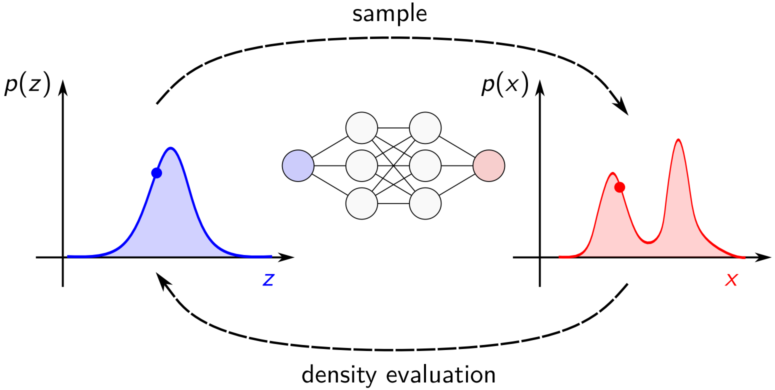
- ET will observe $\mathcal{O}(10^5)$ events per year
- Signals will be longer and have higher SNRs

Premise: Current software will not scale to 3G [1]

Contents

Normalizing flows (NFs)

- Trainable bijection between latent and data spaces
- Sample and evaluate complicated densities
- Used as proposal distribution, trained from MCMC chains



JAX & FLOWMC

Accelerate Python with JAX:

- Use GPU accelerators
- Automatic differentiation → gradient-based samplers
- Just-in-time (JIT) compilation

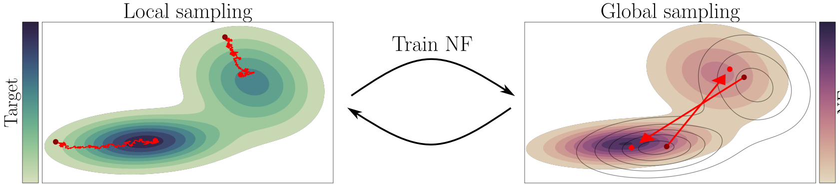
JAX & flowMC

Accelerate Python with JAX:

- Use GPU accelerators
- Automatic differentiation → gradient-based samplers
- Just-in-time (JIT) compilation

flowMC [2, 3]: MCMC + normalizing flows in JAX

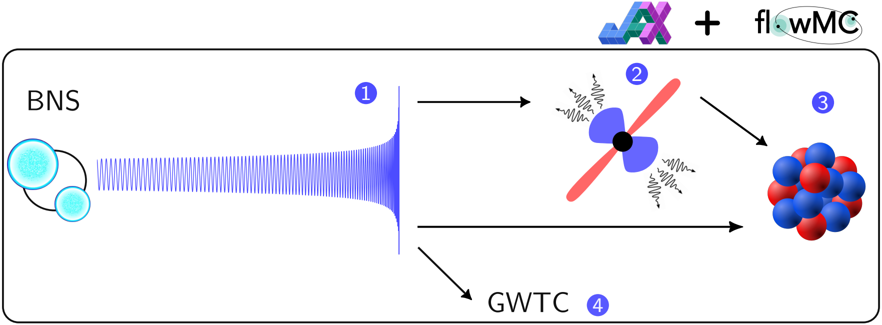
- MCMC chains as training data → no pre-training
- Also see NESSAI [4, 5]



Contents

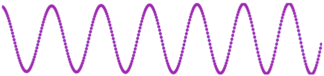
Analyzing a multi-messenger **binary neutron star** (BNS) signal:

1. **Gravitational waves**
2. Electromagnetic counterparts
3. Nuclear equation of state
4. Gravitational wave transient catalogue



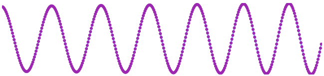
Gravitational waves

- Waveforms on GPU: $\mathcal{O}(10^3)$ faster
- From LALSUITE to JAX: RIPPLE [6]



Gravitational waves

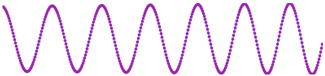
- Waveforms on GPU: $\mathcal{O}(10^3)$ faster
- From LALSUITE to JAX: RIPPLE [6]



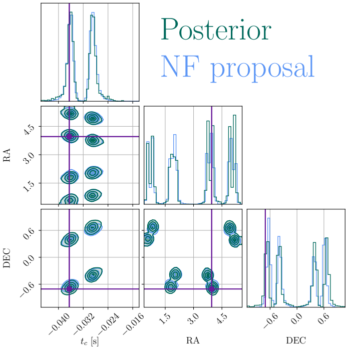
- Parameter estimation: JIM [7, 8]
- BNS in LVK analyzed in ~ 15 min

Gravitational waves

- Waveforms on GPU: $\mathcal{O}(10^3)$ faster
- From LALSUITE to JAX: RIPPLE [6]



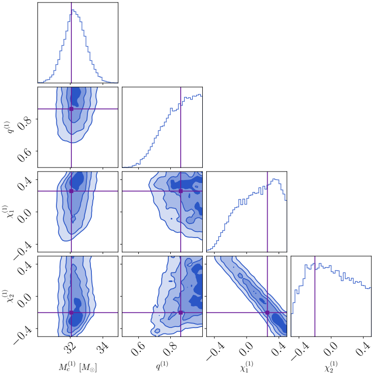
- Parameter estimation: JIM [7, 8]
- BNS in LVK analyzed in ~ 15 min
- Ongoing work for ET – example:
  - BNS, $f_{\min} = 20$ Hz, SNR $= 21$
  - ET-Δ, `IMRPhenomD_NRTidalv2`
  - 30 mins on H100 GPU
- Evidence? HARMONIC [9]

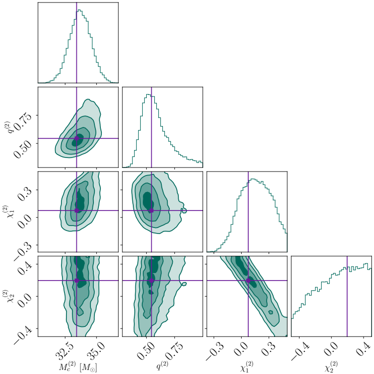


Overlapping signals (Luca Negri, Justin Janquart, James Alvey, Uddipta Bhardwaj)

- Assess scaling of Jim: BBH+BBH in O5 with LVK
  - `IMRPhenomD` $\rightarrow$ 22 parameters (joint parameter estimation)
  - $M_c^{(1)} = 32 M_\odot$, $M_c^{(2)} = 33 M_\odot$, $\Delta t = 70$ ms
  - $\mathrm{SNR}^{(1)} = 25.76$, $\mathrm{SNR}^{(2)} = 25.24$

- Assess scaling of JIM: BBH+BBH in O5 with LVK
  - IMRPhenomD $\rightarrow$ 22 parameters (joint parameter estimation)
  - $M_c^{(1)} = 32 M_\odot$, $M_c^{(2)} = 33 M_\odot$, $\Delta t = 70$ ms
  - $\mathrm{SNR}^{(1)} = 25.76$, $\mathrm{SNR}^{(2)} = 25.24$
  - 1h28m on H100 (vs 23 days on 16 CPUs [10])

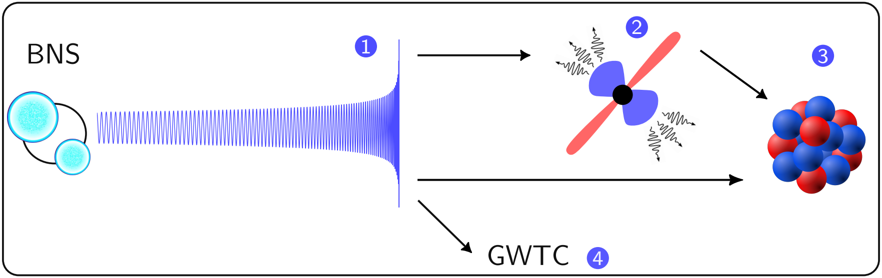




Overview

Analyzing a multi-messenger **binary neutron star** (BNS) signal:

1. Gravitational waves
2. **Electromagnetic counterparts**
3. Nuclear equation of state
4. Gravitational wave transient catalogue

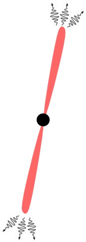


Electromagnetic counterparts (Hauke Koehn, Tim Dietrich)

- BNS mergers lead to kilonovae, gamma-ray bursts (afterglows)
- Numerical models are too expensive (e.g. AFTERGLOWPY [11])

Electromagnetic counterparts (Hauke Koehn, Tim Dietrich)

- BNS mergers lead to kilonovae, gamma-ray bursts (afterglows)
- Numerical models are too expensive (e.g. AFTERGLOWPY [11])
- Neural network surrogates for inference: FIESTA

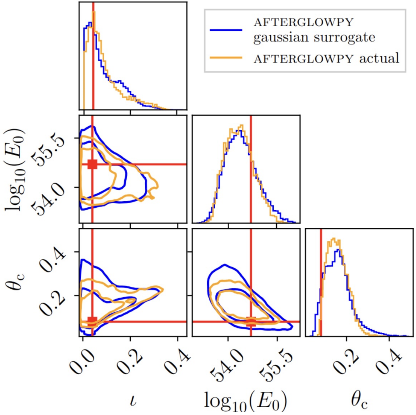


FIESTA

- 1m36s
- 1 H100 GPU

AFTERGLOWPY

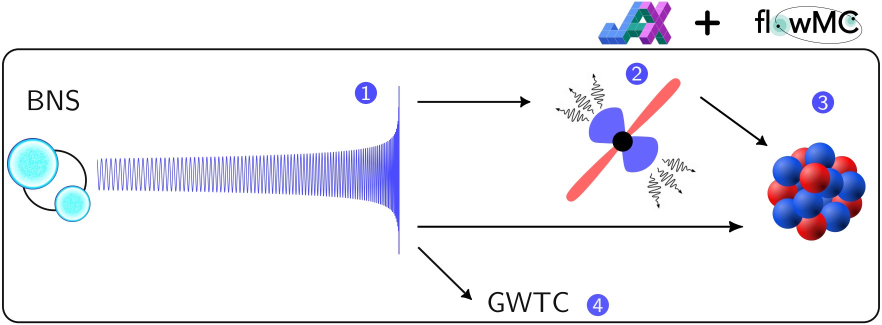
- 4 hours
- 30 CPUs



Overview

Analyzing a multi-messenger **binary neutron star** (BNS) signal:

1. Gravitational waves
2. Electromagnetic counterparts
3. **Nuclear equation of state**
4. Gravitational wave transient catalogue

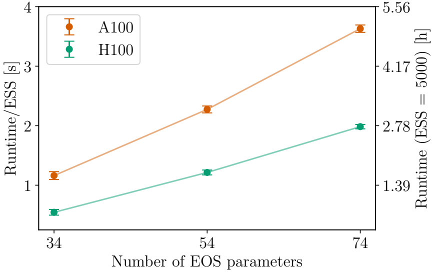


Equation of state inference (Peter T.H. Pang)

- Goal: infer nuclear equation of state (EOS) of neutron stars [12]
- Computational bottleneck: solve TOV equations

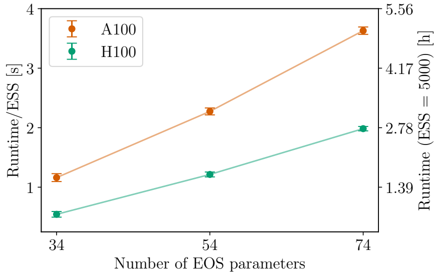
Equation of state inference (Peter T.H. Pang)

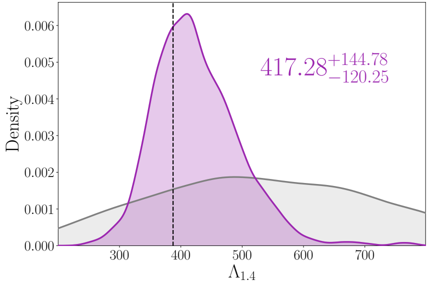
- Goal: infer nuclear equation of state (EOS) of neutron stars [12]
- Computational bottleneck: solve TOV equations
- JESTER [13]: JAX-based TOV solver
  - Full inference in ∼hours
  - No need for machine learning surrogates



Equation of state inference (Peter T.H. Pang)

- Goal: infer nuclear equation of state (EOS) of neutron stars [12]
- Computational bottleneck: solve TOV equations
- JESTER [13]: JAX-based TOV solver
  - Full inference in ∼hours
  - No need for machine learning surrogates
- End-to-end analysis: constrain EOS from 20 BNS in O5



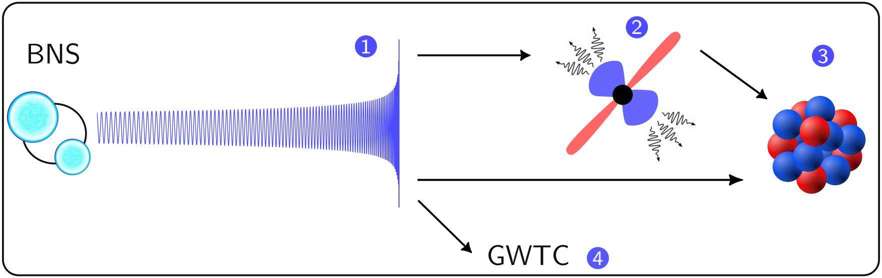


Contents

Overview

Analyzing a multi-messenger **binary neutron star** (BNS) signal:

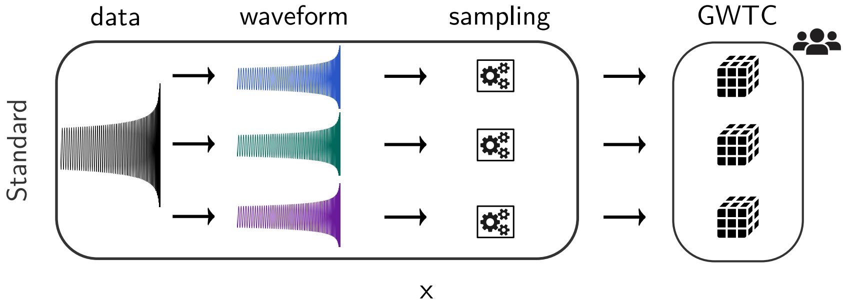
1. Gravitational waves
2. Electromagnetic counterparts
3. Nuclear equation of state
4. **Gravitational wave transient catalogue**



Constructing GWTCs (Thomas Ng, Kaze Wong)

GWTCs do not scale well in memory:

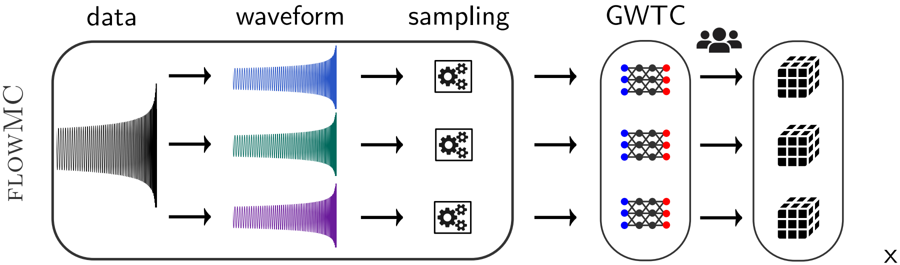
- GWTC stores several samples (different waveforms)
- Standard: fixed sample size, ~ 100 MB



Constructing GWTCs (Thomas Ng, Kaze Wong)

GWTCs do not scale well in memory:

- GWTC stores several samples (different waveforms)
- Standard: fixed sample size, $\sim$ 100 MB
- FLOWMC: generate samples from normalizing flows, $\sim$ 10 MB



Conclusion

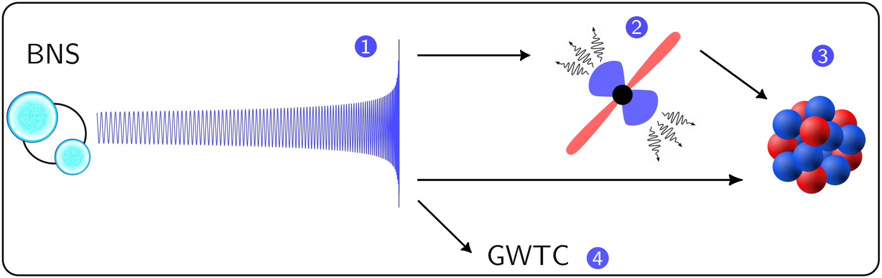
- Progress on scalable Bayesian inference software for 3G, with minimal amount of pre-training
- Hybrid acceleration: GPUs + normalizing flow proposals
  - JAX/GPU: likelihoods faster
  - FLOWMC: sampling converges faster
- Goal: joint multimessenger analyses in ~hours (NMMA [14] in JAX)
- To do
  - GW injection studies for ET
  - More waveform models in JAX
  - Equation of state study with ET data

Let's talk!

Thank you for your attention!

Software written in JAX:

- FLOWMC [2, 3]
- JIM [7, 8] 1 2 4
- FIESTA 2
- JESTER [13] 3
- HARMONIC [9, 15, 16]



References I

[1] Qian Hu and John Veitch. "Costs of Bayesian Parameter Estimation in Third-Generation Gravitational Wave Detectors: a Review of Acceleration Methods". In: (Dec. 2024). arXiv: 2412.02651 [gr-qc].

[2] Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. "Adaptive Monte Carlo augmented with normalizing flows". In: *Proc. Nat. Acad. Sci.* 119.10 (2022), e2109420119. DOI: 10.1073/pnas.2109420119. arXiv: 2105.12603 [physics.data-an].

[3] Kaze W. k. Wong, Marylou Gabrié, and Daniel Foreman-Mackey. "flowMC: Normalizing flow enhanced sampling package for probabilistic inference in JAX". In: *J. Open Source Softw.* 8.83 (2023), p. 5021. DOI: 10.21105/joss.05021. arXiv: 2211.06397 [astro-ph.IM].

[4] Michael J. Williams, John Veitch, and Chris Messenger. "Nested sampling with normalizing flows for gravitational-wave inference". In: *Phys. Rev. D* 103.10 (2021), p. 103006. DOI: 10.1103/PhysRevD.103.103006. arXiv: 2102.11056 [gr-qc].

[5] Michael J. Williams, John Veitch, and Chris Messenger. "Importance nested sampling with normalising flows". In: *Mach. Learn. Sci. Tech.* 4.3 (2023), p. 035011. DOI: 10.1088/2632-2153/acd5aa. arXiv: 2302.08526 [astro-ph.IM].

[6] Thomas D. P. Edwards et al. "Differentiable and hardware-accelerated waveforms for gravitational wave data analysis". In: *Phys. Rev. D* 110.6 (2024), p. 064028. DOI: 10.1103/PhysRevD.110.064028. arXiv: 2302.05329 [astro-ph.IM].

References II

[7] Kaze W. K. Wong, Maximiliano Isi, and Thomas D. P. Edwards. "Fast Gravitational-wave Parameter Estimation without Compromises". In: *Astrophys. J.* 958.2 (2023), p. 129. DOI: 10.3847/1538-4357/acf5cd. arXiv: 2302.05333 [astro-ph.IM].

[8] Thibeau Wouters et al. "Robust parameter estimation within minutes on gravitational wave signals from binary neutron star inspirals". In: *Phys. Rev. D* 110.8 (2024), p. 083033. DOI: 10.1103/PhysRevD.110.083033. arXiv: 2404.11397 [astro-ph.IM].

[9] Alicja Polanska et al. "Accelerated Bayesian parameter estimation and model selection for gravitational waves with normalizing flows". In: *38th conference on Neural Information Processing Systems*. Oct. 2024. arXiv: 2410.21076 [astro-ph.IM].

[10] Justin Janquart et al. "Analyses of overlapping gravitational wave signals using hierarchical subtraction and joint parameter estimation". In: *Mon. Not. Roy. Astron. Soc.* 523.2 (2023), pp. 1699–1710. DOI: 10.1093/mnras/stad1542. arXiv: 2211.01304 [gr-qc].

[11] Geoffrey Ryan et al. "Gamma-Ray Burst Afterglows in the Multimessenger Era: Numerical Models and Closure Relations". In: *Astrophys. J.* 896.2 (2020), p. 166. DOI: 10.3847/1538-4357/ab93cf. arXiv: 1909.11691 [astro-ph.HE].

[12] Adrian Abac et al. "The Science of the Einstein Telescope". In: (Mar. 2025). arXiv: 2503.12263 [gr-qc].

References III

[13] Thibeau Wouters et al. "Leveraging differentiable programming in the inverse problem of neutron stars". In: (Apr. 2025). arXiv: 2504.15893 [astro-ph.HE].

[14] Peter T. H. Pang et al. "An updated nuclear-physics and multi-messenger astrophysics framework for binary neutron star mergers". In: *Nature Commun.* 14.1 (2023). Available at https://github.com/nuclear-multimessenger-astronomy/nmma, p. 8352. DOI: 10.1038/s41467-023-43932-6. arXiv: 2205.08513 [astro-ph.HE].

[15] Jason D. McEwen et al. *Machine learning assisted Bayesian model comparison: learnt harmonic mean estimator*. 2023. arXiv: 2111.12720 [stat.ME]. URL: https://arxiv.org/abs/2111.12720.

[16] Alicja Polanska et al. *Learned harmonic mean estimation of the marginal likelihood with normalizing flows*. 2024. arXiv: 2307.00048 [stat.ME]. URL: https://arxiv.org/abs/2307.00048.

[17] Kurzgesagt. *Figures taken from "Neutron Stars - The Most Extreme Things that are not Black Holes"*. Accessed on May 14, 2025. 2019. URL: https://www.youtube.com/watch?v=udFxKZRyQt4.

[18] Hergé. *Cover figure created with ChatGPT using this input figure from the comic Destination Moon*. Accessed on May 14, 2025. 2019. URL: https://www.youtube.com/watch?v=udFxKZRyQt4.

Evidence calculation: HARMONIC I

Evidence Z can be computed from posterior samples with HARMONIC [15] with the harmonic mean estimator

$$\rho \equiv \mathbb{E}_{P(\theta|d)}\left[\frac{1}{L(\theta)}\right] = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} P(\theta|d) = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{Z} = \frac{1}{Z}$$

Therefore, estimate ρ with posterior samples:

$$\hat{\rho} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\mathcal{L}(\theta)}, \quad \theta_i \sim P(\theta|d)$$

Evidence calculation: HARMONIC II

Can be interpreted as importance sampling

$$\rho = \int \mathrm{d}\theta \frac{1}{Z} \frac{\pi(\theta)}{P(\theta|d)} P(\theta|d) \, ,$$

but with target = prior and sampling density = posterior. Therefore, importance sampling is inefficient – how to solve?
New proposal:

$$\begin{aligned}
\rho &= \mathbb{E}_{P(\theta|d)} \left[\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \right] \\
&= \int \mathrm{d}\theta \, \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \, P(\theta|d) \\
&= \int \mathrm{d}\theta \, \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \, \frac{\mathcal{L}(\theta)\pi(\theta)}{Z} = \frac{1}{Z}
\end{aligned}$$

Evidence calculation: HARMONIC III

Use the following estimator:

$$\hat{\rho} = \frac{1}{N}\sum_{i=1}^{N}\frac{\varphi(\theta_i)}{\mathcal{L}(\theta_i)\pi(\theta_i)}, \quad \theta_i \sim P(\theta|d)$$

Replace the target distribution π with φ: only requirement is that it is normalized

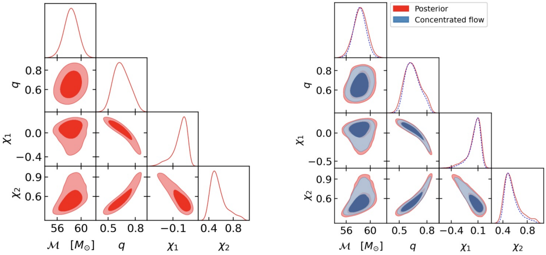
In practice, this can be achieved with a normalizing flow [16].

This has been verified to give accurate evidences (similar values as nested sampling) when GW posteriors are used [9].

HARMONIC with JIM

Table 1: Total wall times to compute the evidence estimates for the examples discussed in the main text. We run BILBY on 16 CPU cores and JIM + `harmonic` on 1 GPU.

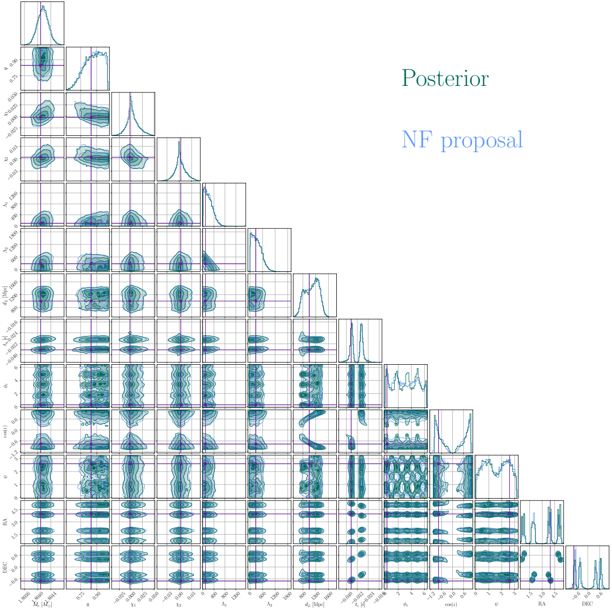
Example	Method	$\log(z)$	Sampling time	Evidence estimation time
4D	BILBY	390.33 ± 0.11	31.3 min	–
	JIM + `harmonic`	$390.360^{+0.006}_{-0.006}$	3.4 min	1.9 min
11D	BILBY	378.29 ± 0.15	3.5 h	–
	JIM + `harmonic`	$378.420^{+0.09}_{-0.08}$	11.8 min	2.4 min



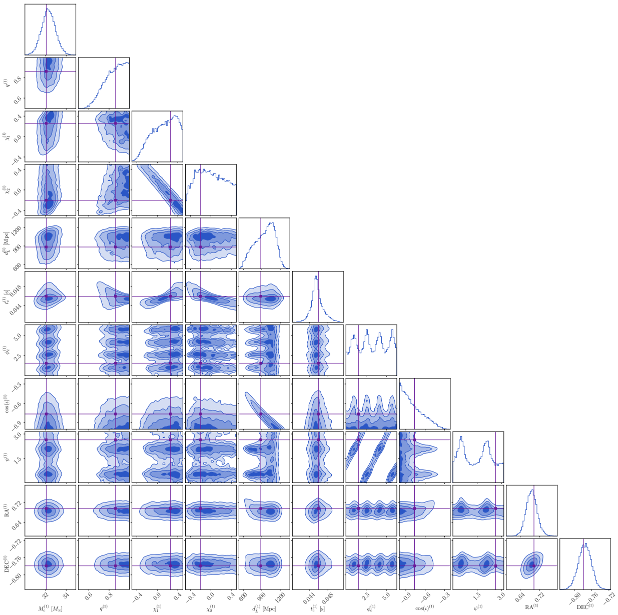
(a) BILBY (b) JIM + `harmonic`

Figure 1: Corner plots for the 4-dimensional posterior samples from (a) BILBY and (b) JIM used for inference (solid red) alongside the concentrated flow at $T = 0.8$ used in the learned harmonic mean (dashed blue).

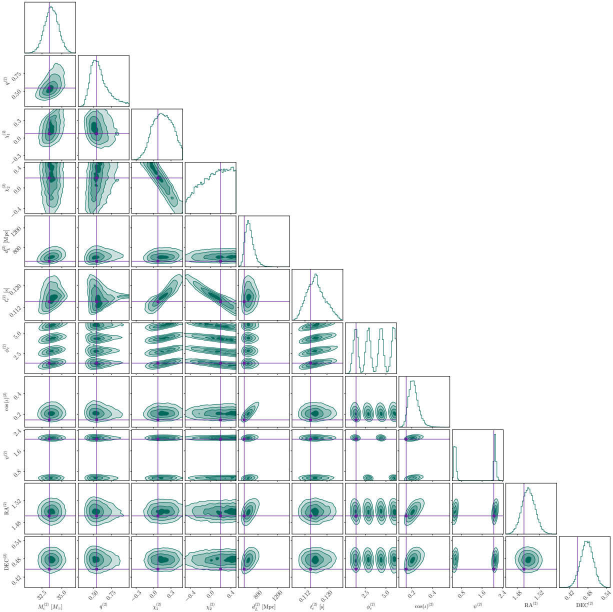
BNS in ET-Δ example: all parameters



Overlapping signals: all parameters signal A

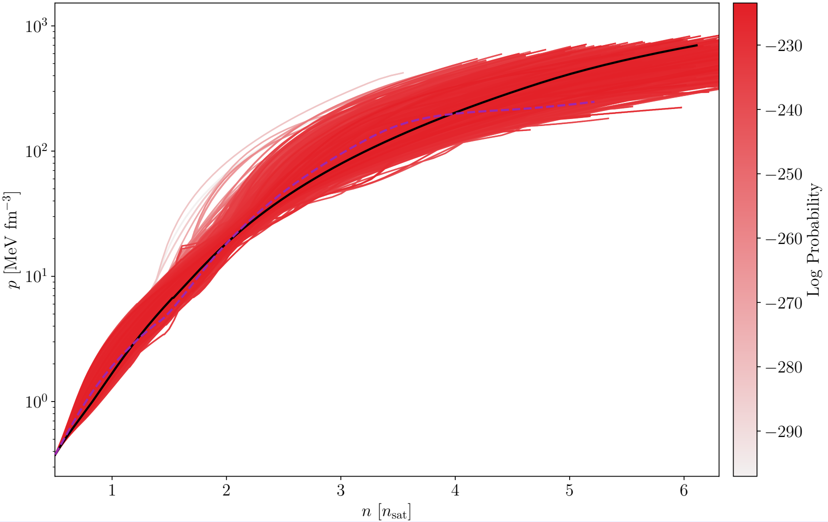


Overlapping signals: all parameters signal B



Equation of state O5 projection with 20 BNS: EOS

- **Purple**: target
- **Red**: posterior EOS samples (**black**: maximum log posterior)



Equation of state O5 projection with 20 BNS: NS