Accelerating gravitational wave parameter estimation with normalizing flows

Thibeau Wouters

December 5, 2023

Utrecht University

Table of Contents

Table of Contents

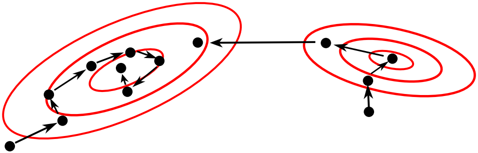
Parameter estimation

- Parameter estimation (PE): get posterior of GW parameters θ

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

- Sampling via Markov Chain Monte Carlo (MCMC) [1]
- For binary neutron stars (BNS): computationally expensive

How to sample from high-dimensional, multi-modal posteriors?



We extend `jim` [2], based on `jax` [3], with building blocks:

1. Normalizing flow-enhanced, gradient-based MCMC (`flowMC` [4, 5])
2. Automatically-differentiable (AD) GW (`ripple` [6])
3. Relative binning likelihood [7]

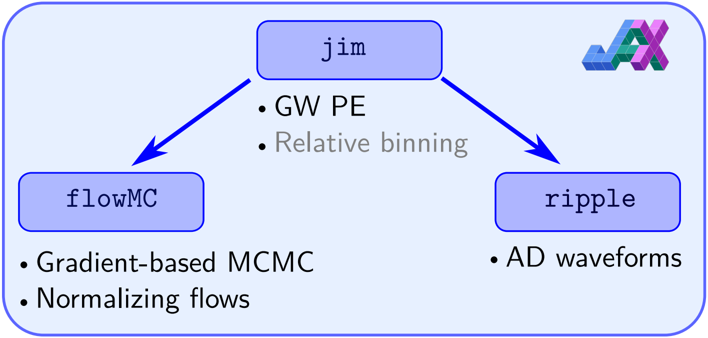


Table of Contents

Why `jax`?

What are the benefits of `jax` for MCMC?

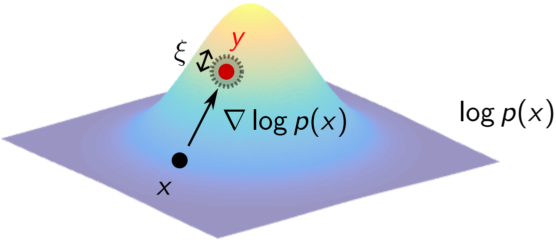
1. Automatic differentiation (AD)
2. Just-in-time (JIT) compilation
3. GPU acceleration
4. Parallelization
5. Interoperability with `numpy`

Table of Contents

1. **Local sampling**: MALA (Metropolis-adjusted Langevin algorithm)
   - Proposal y: Langevin diffusion

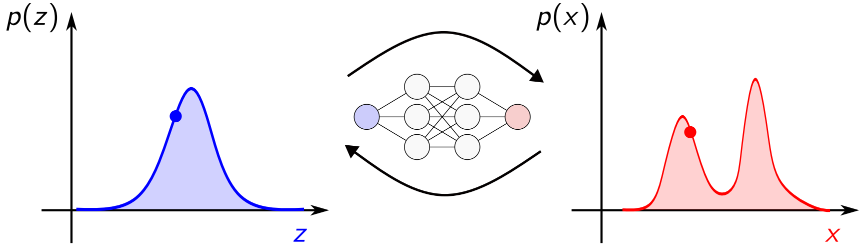
$$y = x + \frac{\epsilon^2}{2} \nabla \log p(x) + \epsilon \xi$$

   - Metropolis-Hastings acceptance step



Normalizing flows (NF):

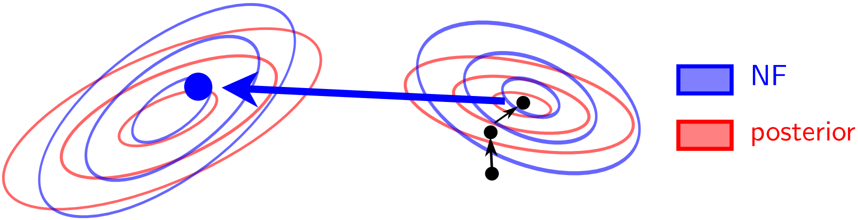
- Latent space: easy to sample (e.g. Gaussian)
- Data space: distribution learned from samples
- Enable approximate sampling from complicated distributions



flowMC – global sampling

2. **Global sampling**

- Global proposal by sampling from NF
- Metropolis-Hastings acceptance step



flowMC – complete algorithm

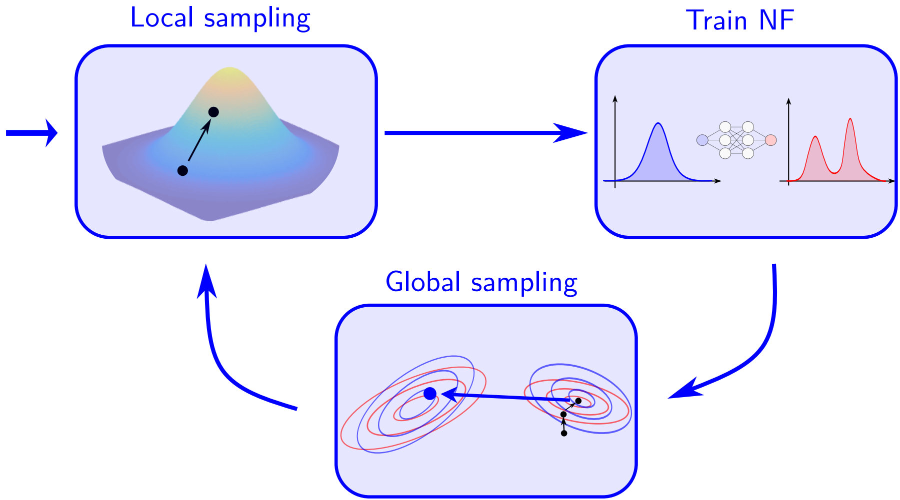


Table of Contents

Results

- TaylorF2 in `ripple`
- IMRPhenomD_NRTidalv2 in `ripple` (ongoing)
- Reproduced PE for GW170817 & GW190425 with TaylorF2
- $\sim$ 30 mins training, $\sim$ 1 min sampling

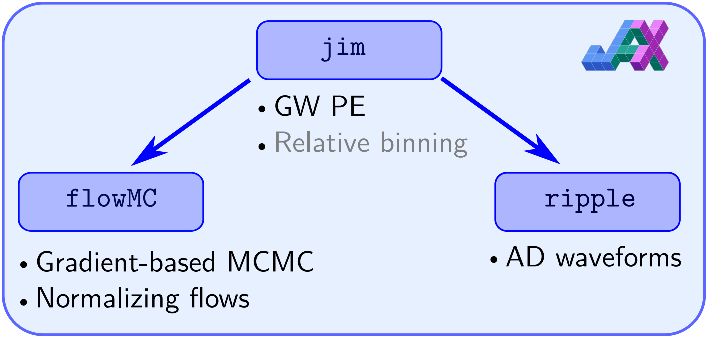


Table of Contents

Future work

- Finish IMRPhenomD_NRTidalv2 in `ripple`
- Injection studies and pp-plot
- Investigate synergy with simulation-based inference

Conclusion

- `flowMC`: NF-enhanced, gradient-based MCMC
- `ripple`: automatically differentiable GW
- `jim` = `jax` + `flowMC` + `ripple`
- `jim` is a promising tool for fast and accurate parameter estimation
- `jim` can be used for parameter estimation of BNS and BBH
- `jim` can enhance and benefit from simulation-based inference

References

[1] Steve Brooks et al. *Handbook of markov chain monte carlo*. CRC press, 2011.

[2] Kaze WK Wong, Maximiliano Isi, and Thomas DP Edwards. "Fast gravitational wave parameter estimation without compromises". In: *arXiv preprint arXiv:2302.05333* (2023).

[3] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: http://github.com/google/jax.

[4] Marylou Gabrié, Grant M Rotskoff, and Eric Vanden-Eijnden. "Efficient bayesian sampling using normalizing flows to assist markov chain monte carlo methods". In: *arXiv preprint arXiv:2107.08001* (2021).

[5] Kaze WK Wong, Marylou Gabrié, and Daniel Foreman-Mackey. "flowMC: Normalizing-flow enhanced sampling package for probabilistic inference in Jax". In: *arXiv preprint arXiv:2211.06397* (2022).

[6] Thomas DP Edwards et al. "ripple: Differentiable and Hardware-Accelerated Waveforms for Gravitational Wave Data Analysis". In: *arXiv preprint arXiv:2302.05329* (2023).

[7] Barak Zackay, Liang Dai, and Tejaswi Venumadhav. "Relative binning and fast likelihood evaluation for gravitational wave parameter estimation". In: *arXiv preprint arXiv:1806.08792* (2018).

BACK-UP SLIDES

flowMC: production loop

- After training, NF is “frozen”, and the Markov property is restored.
- We run a few “production loops” to sample from the posterior for the final PE.