Accelerated nested sampling with applications to cosmology and gravitational waves

Metha Prathaban
myp23@cam.ac.uk

About Me

- 3rd year PhD student
- Work on Bayesian numerical method development in context of GWs

Current work is in collaboration with Will Handley and Harry Bevins.


Table of Contents

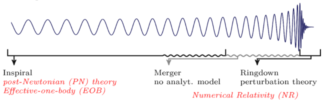
Inverse problems in GW physics

Forward problem

model parameters
$\theta = \mathcal{M}, q, d_L, t_c \dots$

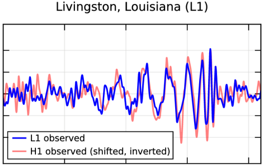
waveform model

template for strain waveform



Inverse problem

observed signal strain



inference

model parameters, inferred
θ_{est}

Given some model $\mathcal{M}$ and observed signal $\mathcal{D}$, Bayes' theorem enables us to relate the posterior probability of the set of parameters θ which generated the signal to the likelihood of the $\mathcal{D}$ given θ and the prior probability of θ given $\mathcal{M}$:

$$\mathcal{P}(\theta|D,\mathcal{M}) = \frac{P(D|\theta,\mathcal{M})P(\theta|\mathcal{M})}{P(D|\mathcal{M})} = \frac{\mathcal{L}(D|\theta)\pi(\theta)}{\mathcal{Z}} \tag{1}$$

The evidence, $\mathcal{Z}$, plays a key role in model comparison.

Samplers

Have to explore parameter space efficiently, to do inference in feasible timescales.

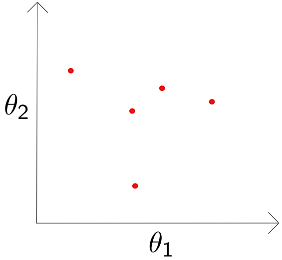
For cosmology, implemented in CosmoMC, Cobaya, CosmoSIS, Monte Python etc. For GWs, Bilby.

Posterior samplers:

- Metropolis-Hastings
- Hamiltonian Monte-Carlo (stan, blackjax)
- Ensemble samplers (emcee)

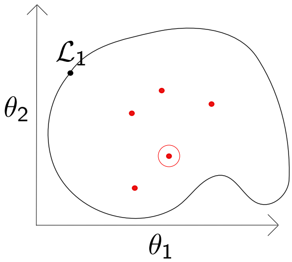
Don't calculate the evidence, $\mathcal{Z}$ (directly) - crucial for Bayesian model comparison!

Nested sampling first and foremost calculates evidence, $\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$.



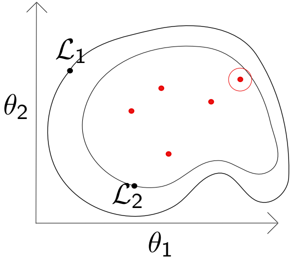
- Prior is populated with set of 'live points'.

Nested sampling first and foremost calculates evidence, $\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$.



- Prior is populated with set of 'live points'.
- At each iteration i, point is lowest likelihood is deleted and new live point is drawn, which must have a likelihood higher than that of the deleted point.

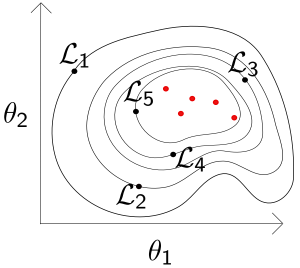
Nested sampling first and foremost calculates evidence, $\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$.



- Prior is populated with set of 'live points'.
- At each iteration i, point is lowest likelihood is deleted and new live point is drawn, which must have a likelihood higher than that of the deleted point.

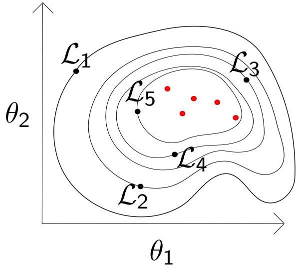
Nested sampling (NS)

Nested sampling first and foremost calculates evidence, $\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$.



- Prior is populated with set of 'live points'.
- At each iteration i, point is lowest likelihood is deleted and new live point is drawn, which must have a likelihood higher than that of the deleted point.
- Live points compress exponentially towards peak of likelihood.

Nested sampling first and foremost calculates evidence, $\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$.



- Prior is populated with set of 'live points'.
- At each iteration i, point is lowest likelihood is deleted and new live point is drawn, which must have a likelihood higher than that of the deleted point.
- Live points compress exponentially towards peak of likelihood.
- Evidence is calculated as weighted sum over deleted ('dead') points.

Table of Contents

Time of convergence of NS:

$$T \propto T_{\mathcal{L}} \times f_{\rm sampler} \times D_{\rm KL} \times n_{\rm live} \tag{2}$$

Runtime of NS

Time of convergence of NS:

$$T \propto T_{\mathcal{L}} \times f_{\text{sampler}} \times D_{\text{KL}} \times n_{\text{live}} \tag{2}$$

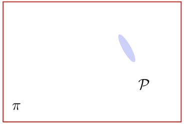
focus of this talk

Runtime of NS

Time of convergence of NS:

$$T \propto T_{\mathcal{L}} \times f_{\text{sampler}} \times D_{\text{KL}} \times n_{\text{live}} \tag{2}$$

compression from prior to posterior $\left(\approx \ln \frac{V_\pi}{V_\mathcal{P}}\right)$



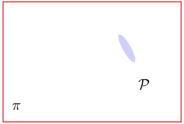
Runtime of NS

Time of convergence of NS:

likelihood evaluation time

$$T \propto T_{\mathcal{L}} \times f_{\mathrm{sampler}} \times D_{\mathrm{KL}} \times n_{\mathrm{live}} \tag{2}$$

compression from prior to posterior $\left(\approx \ln \frac{V_\pi}{V_{\mathcal{P}}}\right)$

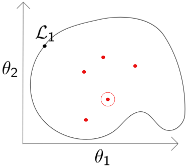


Time of convergence of NS:

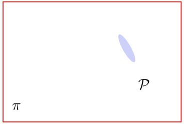
likelihood evaluation time

$$T \propto T_{\mathcal{L}} \times f_{\mathrm{sampler}} \times D_{\mathrm{KL}} \times n_{\mathrm{live}} \quad (2)$$

compression from prior to posterior $\left(\approx \ln \frac{V_\pi}{V_{\mathcal{P}}}\right)$



drawing new live point



Runtime of NS

Time of convergence of NS:

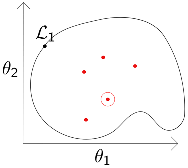
likelihood evaluation time

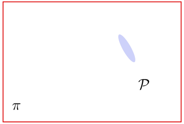
resolution

$$T \propto T_{\mathcal{L}} \times f_{\text{sampler}} \times D_{\text{KL}} \times n_{\text{live}} \tag{2}$$

compression from prior to posterior $(\approx \ln \frac{V_\pi}{V_{\mathcal{P}}})$

drawing new live point





Time of convergence of NS:

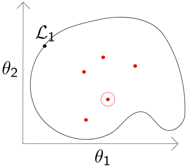
likelihood evaluation time

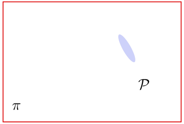
resolution (baked in)

$$T \propto T_{\mathcal{L}} \times f_{\text{sampler}} \times D_{\text{KL}} \times n_{\text{live}} \tag{2}$$

compression from prior to posterior $(\approx \ln \frac{V_\pi}{V_\mathcal{P}})$

drawing new live point





Runtime of NS

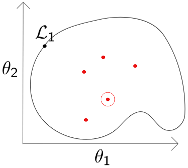
Time of convergence of NS:

faster waveform models, CosmoPower

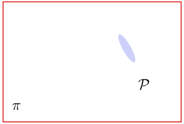
resolution (baked in)

$$T \propto T_{\mathcal{L}} \times f_{\text{sampler}} \times D_{\text{KL}} \times n_{\text{live}} \qquad (2)$$

compression from prior to posterior ($\approx \ln \frac{V_\pi}{V_{\mathcal{P}}}$)



drawing new live point



Runtime of NS

Time of convergence of NS:

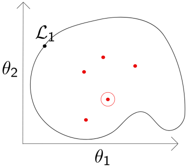
faster waveform models, CosmoPower

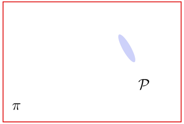
resolution (baked in)

$$T \propto T_{\mathcal{L}} \times f_{\mathrm{sampler}} \times D_{\mathrm{KL}} \times n_{\mathrm{live}} \tag{2}$$

compression from prior to posterior $\left(\approx \ln \frac{V_\pi}{V_\mathcal{P}}\right)$

better samplers





Time of convergence of NS

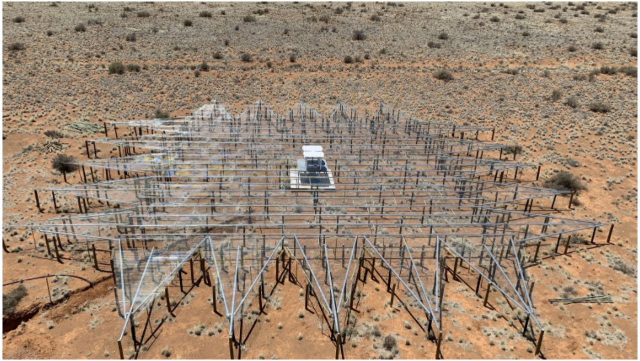
$$T \propto T_{\mathcal{L}} \times f_{\mathrm{sampler}} \times D_{\mathrm{KL}} \times n_{\mathrm{live}} \tag{3}$$

Uncertainty in $\log \mathcal{Z}$

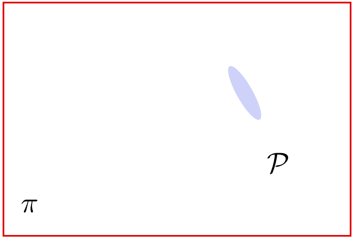
$$\sigma \propto \sqrt{D_{\mathrm{KL}}/n_{\mathrm{live}}} \tag{4}$$

Precision-normalized runtime has quadratic dependence on KL divergence. 2212.01760

One way to do this (REACH):

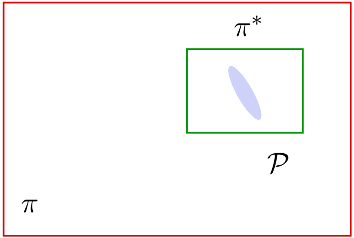


One way to do this (REACH):



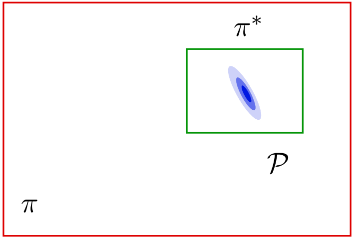
- Perform low resolution (low live points) run first to roughly identify where posterior lies.

One way to do this (REACH):



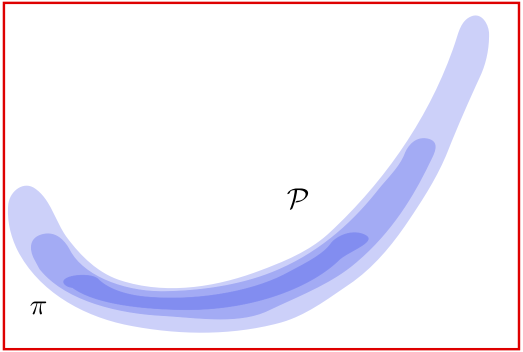
- Perform low resolution (low live points) run first to roughly identify where posterior lies.
- Then set off second, high resolution, run with **narrower** box prior (much quicker).

One way to do this (REACH):



- Perform low resolution (low live points) run first to roughly identify where posterior lies.
- Then set off second, high resolution, run with **narrower** box prior (much quicker).
- Evidence has **changed** (since different prior), but easy to correct (multiply new evidence by $\frac{V_{\pi^*}}{V_\pi}$)

When does this break down?



- Banana distributions, multi-modality etc.
- Precludes its use in most realistic GW cases...

- Use **normalizing flows** (NF) to learn the rough posterior, and use this as our updated prior, π^*.
- In this case, can't do our trick of correcting the second evidence by volume ratio, $\frac{V_{\pi^*}}{V_\pi}$!
- Must rely on another technique to get around this!

NF + PR

- Use **normalizing flows** (NF) to learn the rough posterior, and use this as our updated prior, π^*.
- In this case, can't do our trick of correcting the second evidence by volume ratio, $\frac{V_{\pi^*}}{V_\pi}$!
- Must rely on another technique to get around this!

Posterior repartitioning (PR) can help us with this! (see e.g. 2212.01760)

Bayesian Analysis (0000) **00**, Number 0, pp. 1

Bayesian posterior repartitioning for nested sampling

Xi Chen[*,†], Farhan Feroz[†] and Michael Hobson[†]

Improving the efficiency and robustness of nested sampling using posterior repartitioning

Xi Chen · Michael Hobson · Saptarshi Das · Paul Gelderblom

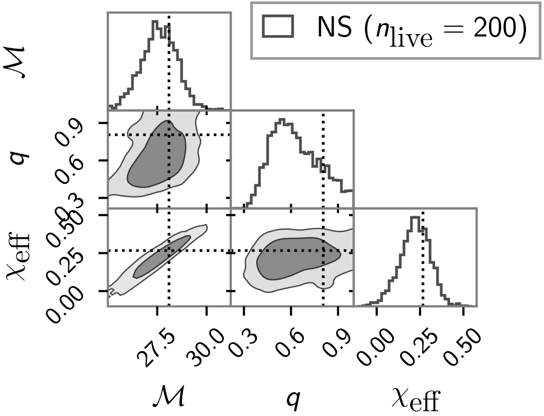
MDPI

Article

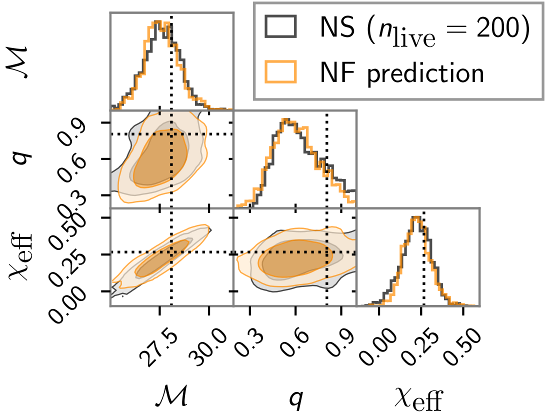
SuperNest: accelerated nested sampling applied to astrophysics and cosmology[†]

Aleksandr Petrosyan[1,2,3*] & Will Handley[1,2,4†]

- Perform low resolution run on simulated data.

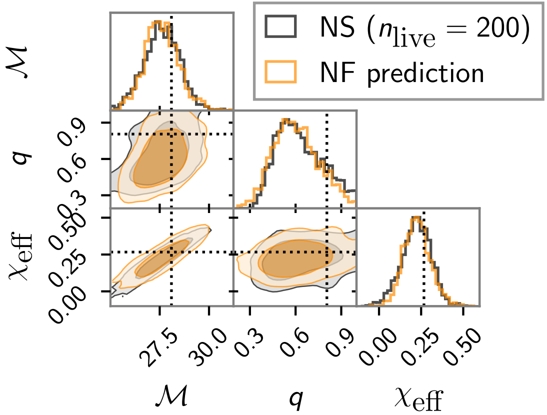


- Perform low resolution run on simulated data.
- Train NF on the weighted samples.



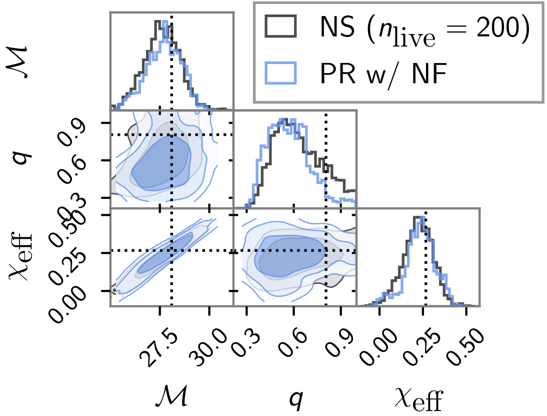
Demo on simulated example

- Perform low resolution run on simulated data.
- Train NF on the weighted samples.
- Use this as 'repartitioned prior' for new high resolution run (using PR to also update likelihood accordingly to same evidences and posteriors out).



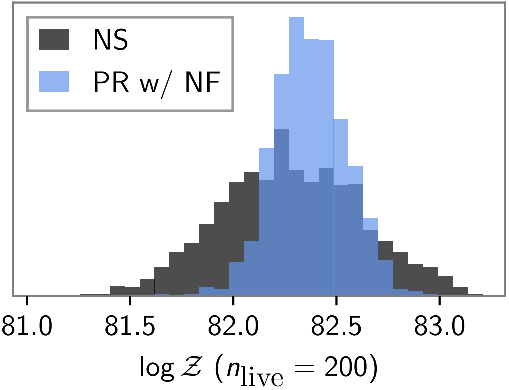
Demo on simulated example

- Perform low resolution run on simulated data.
- Train NF on the weighted samples.
- Use this as ‘repartitioned prior’ for new high resolution run (using PR to also update likelihood accordingly to same evidences and posteriors out).



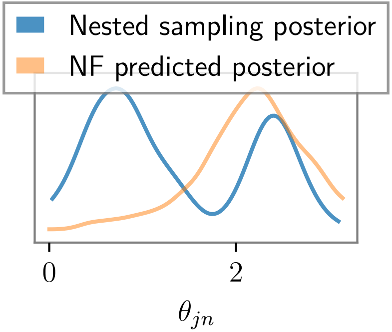
Demo on simulated example

- Perform low resolution run on simulated data.
- Train NF on the weighted samples.
- Use this as 'repartitioned prior' for new high resolution run (using PR to also update likelihood accordingly to same evidences and posteriors out).

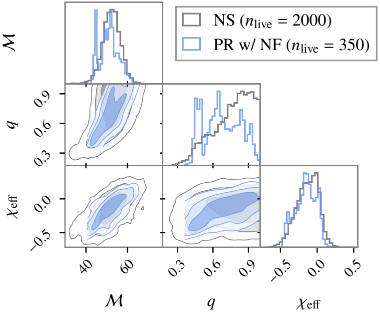


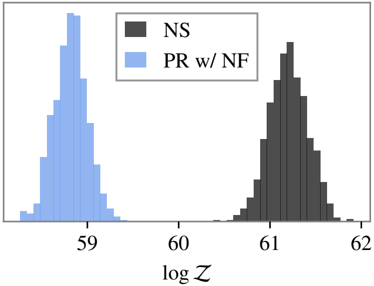
Same answer as doing a full resolution pass of NS, but 7x faster (precision-normalized).

- Still have an issue with multi-modality (if NF only learns one mode, the others are cut off at the prior level in the high resolution run)!



When the NF has been unable to properly learn the multi-modality, we can get biased posteriors and evidences (GW191222):





Improving the method

In order to improve the robustness of the method:

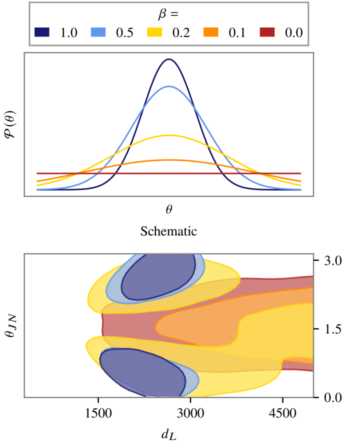
- Repartitioned prior should ideally be able to **widen itself adaptively at runtime** to mitigate missed modes and badly learned posteriors.

Table of Contents

Temperature to adapt prior

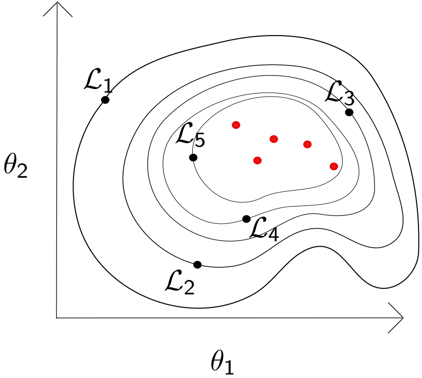
- **Temperature** can be used to broaden or contract distributions - β is inverse temperature.
- Set the repartitioned prior to be anywhere between the posterior ($\beta = 1$) and the original prior ($\beta = 0$).
- Want to exploit this temperature property.

$$p(\beta) \propto \mathcal{L}^{\beta}\pi, \qquad \beta \in [0, 1] \qquad (5)$$



Schematic

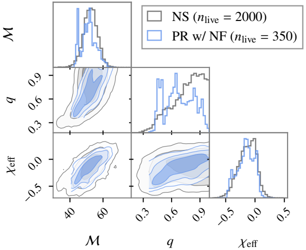
In practice

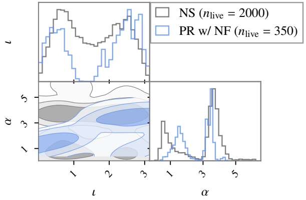


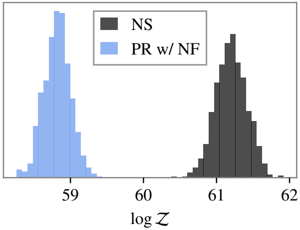
- Nested sampling sees tip to tail of the posterior in a systematic way, and NS has deep tails.
- NS can be used to train a specialized form of **conditional NFs**.
- β-flows are new and only used in this work so far, though broadly applicable, as they can better learn these deep tail events.
- Can now estimate density **at any temperature** - sample over β at runtime.

- Using β-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over β now fixes the problem.

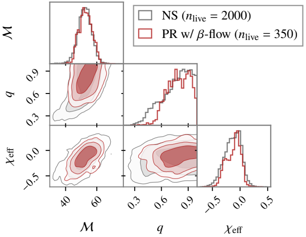
- Using β-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over β now fixes the problem.

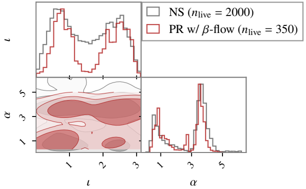


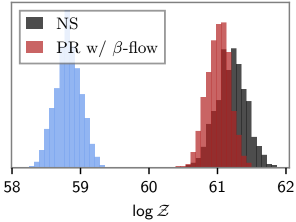




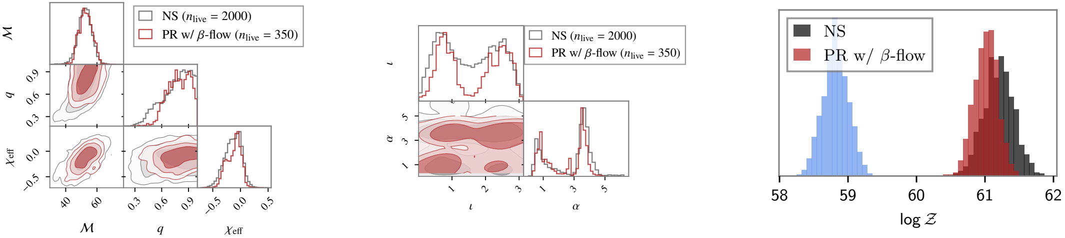
- Using β-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over β now fixes the problem.







- Using β-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over β now fixes the problem.



Only 2x (precision-normalized) as fast as normal NS for this real example, but robust.

Summary

- Several ways to reduce NS runtime, including reducing amount of compression from prior to posterior.
  - Can perform low resolution run to identify rough posterior, learn distribution with a flow, and perform high resolution run with updated prior.
  - Use posterior repartitioning to get correct evidences out, despite changed prior.
  - Can achieve order of magnitude speedups on realistic GW examples.

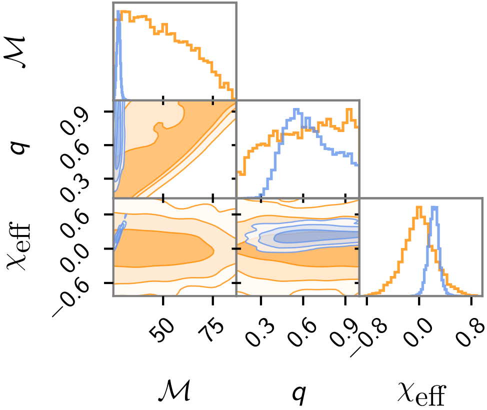
Summary

- Several ways to reduce NS runtime, including reducing amount of compression from prior to posterior.
  - Can perform low resolution run to identify rough posterior, learn distribution with a flow, and perform high resolution run with updated prior.
  - Use posterior repartitioning to get correct evidences out, despite changed prior.
  - Can achieve order of magnitude speedups on realistic GW examples.

- Introduced β-flows:
  - Conditional normalizing flow, trained with whole NS run
  - Better at deep tail events
  - First application is in this paper, but their use is much broader!

Thank you for listening!

- Define prior, **sample** (unnormalized) posterior ($\mathcal{L}(D|\theta) \times \pi(\theta)$).



Challenges:

- High-dimensional parameter spaces $\Rightarrow$ posterior occupies vanishingly small region of prior.
- Complex likelihoods with high costs

Nested sampling first and foremost calculates this evidence. The evidence is the integral of likelihood × prior over the entire parameter space,

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta, \tag{6}$$

which, in general, is a many dimensional integral.

NS turn this into a 1D problem, performing this integral by summing over nested likelihood contours in the parameter space.

Posterior repartitioning (PR)

- Evidence and posterior only depend on product of $\mathcal{L}$ and π:

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta \qquad (7)$$

$$\mathcal{P}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}} \qquad (8)$$

We are free to redefine the likelihood and prior however we like - as long as the product is the same! arXiv:1908.04655

$$\tilde{\mathcal{Z}} = \int \tilde{\mathcal{L}}(\theta)\tilde{\pi}(\theta)d\theta = \int \mathcal{L}(\theta)\pi(\theta)d\theta = \mathcal{Z} \qquad (9)$$

- Many sampling algorithms do not distinguish between $\mathcal{L}$ and π at the algorithmic level.
- e.g. Metropolis-Hastings acceptance ratio only depends on the **joint distribution**, $\mathcal{L}(\theta)\pi(\theta)$.
- Nested sampling does distinguish between prior and likelihood at the algorithmic level, by 'sampling from the prior π, subject to the hard likelihood constraint, $\mathcal{L}$'.

- $\mathcal{Z}$ and $\mathcal{P}$ will not change if we repartition $\mathcal{L}$ and π, **but $\mathcal{D}_{\mathrm{KL}}$ will**.

$$\pi(\theta) \longrightarrow \mathsf{NF}(\theta)$$

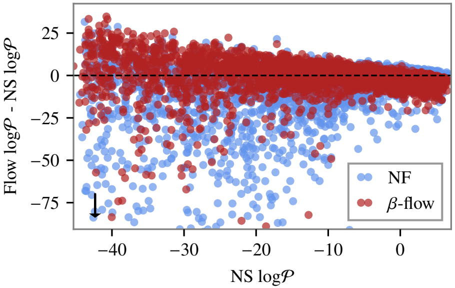
$$\pi(\theta) \longrightarrow \mathsf{NF}(\theta)$$

$$\mathcal{L}(\theta) \longrightarrow \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathsf{NF}(\theta)}$$

$$\pi(\theta) \longrightarrow \mathsf{NF}(\theta)$$

$$\mathcal{L}(\theta) \longrightarrow \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathsf{NF}(\theta)}$$

$$\mathcal{D}_{\mathrm{KL}} \approx \log \frac{V_{\mathsf{NF}}}{V_{\mathcal{P}}}$$



- For simulated example shown before, the β-flow is able to better predict the NS posterior probabilities.
- β-flows exhibit less scatter in the tails (low posterior probabilities) than the NFs.

- NS compresses step by step from prior to posterior.
- We can label these stages by a parameter β (akin to inverse temperature β in e.g. materials science).
- Sliding scale from $\beta = 0$ as the prior and $\beta = 1$ as the posterior.