# Accelerated nested sampling with $\beta$-flows

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.**

Nested sampling

Accelerating NS

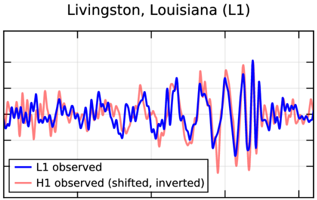
$\beta$-flows

Nested sampling

Accelerating NS

$\beta$-flows

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

```
{
  "label": "lowres",
  "outdir": "/rds/user/myp23/hpc-work/acceleratedns/old_terminate/lowres_run0",
  "sampler": "pypolychord",
  "log_evidence": -3969.4096946457507,
  "log_evidence_err": 0.351653537564021,
  "log_noise_evidence": -4051.78662163637,
  "log_bayes_factor": 82.3769269906191,
  "priors": {
    "mass_1": {
      "__prior__": true,
      "__module__": "bilby.core.prior.base",
      "__name__": "Constraint",
      "kwargs": {
        "minimum": 5,
        "maximum": 100,
        "name": "mass_1",
        "latex_label": "$m_1$",
        "unit": null
      }
    },
    "mass_2": {
      "__prior__": true,
      "__module__": "bilby.core.prior.base",
      "__name__": "Constraint",
      "kwargs": {
        "minimum": 5,
        "maximum": 100,
        "name": "mass_2",
        "latex_label": "$m_2$",
```


2006.00714

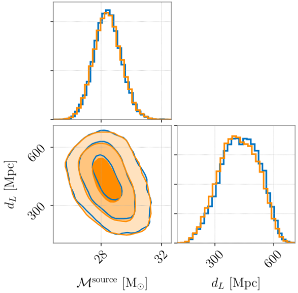
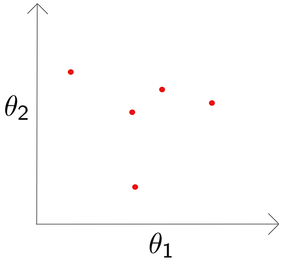



**Figure 9.** Posterior probability distributions for source-frame chirp mass $\mathcal{M}^{\rm source}$ and luminosity distance $d_{\rm L}$ for GW150914. We display posteriors obtained using BILBY in orange, and LALINFERENCE posteriors in blue. We reweight the LALINFERENCE posteriors to the BILBY default priors using the procedure outlined in Appendix C. The one-dimensional JS divergence on chirp mass $\mathcal{M}$ and luminosity distance $d_{\rm L}$ for this event are $\mathrm{JS}_{\mathcal{M}} = 0.0017\,\mathrm{nat}$ and $\mathrm{JS}_{d_{\rm L}} = 0.0015\,\mathrm{nat}$.

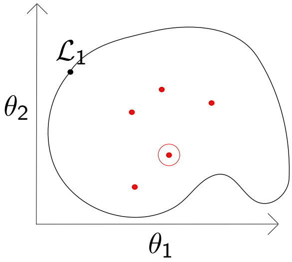
Nested sampling first and foremost calculates evidence, $\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$.




- Prior is populated with set of ‘live points’.

## Nested sampling (NS)

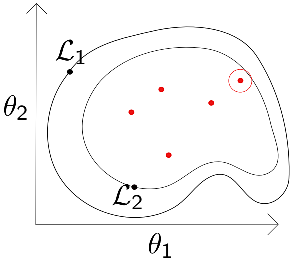
Nested sampling first and foremost calculates evidence, $\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$.




- 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.

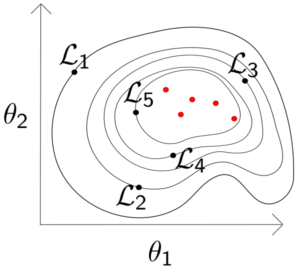
# Nested sampling (NS)

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



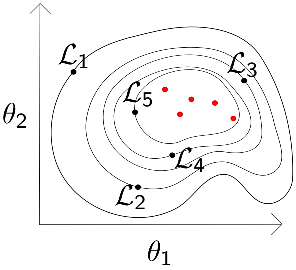

- Prior is populated with set of 'live points'.
- At each iteration $i$, point is lowest likelihood is deleted and new live point is drawn, which must have a likelihood higher than that of the deleted point.

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.

Nested sampling

Accelerating NS

$\beta$-flows

Time of convergence of NS:

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

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

Time of convergence of NS:

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

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

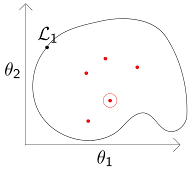
Time of convergence of NS:

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

single likelihood evaluation time

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

Time of convergence of NS:




drawing new live point subject to hard likelihood constraint

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

single likelihood evaluation time

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

Time of convergence of NS:

resolution

drawing new live point subject to hard likelihood constraint

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

single likelihood evaluation time

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

Time of convergence of NS:

resolution (baked in)

drawing new live point subject to hard likelihood constraint

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

single likelihood evaluation time

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

Time of convergence of NS:

resolution (baked in)

drawing new live point subject to hard likelihood constraint

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

faster waveform models

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

Time of convergence of NS:

resolution (baked in)

better samplers

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

faster waveform models

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

## Time of convergence of NS

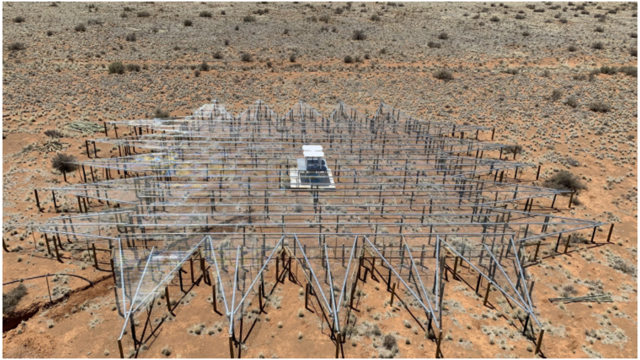
$$T \propto T_{\mathcal{L}} \times f_{\text{sampler}} \times D_{\text{KL}} \times n_{\text{live}} \tag{3}$$

## Uncertainty in $\log\mathcal{Z}$

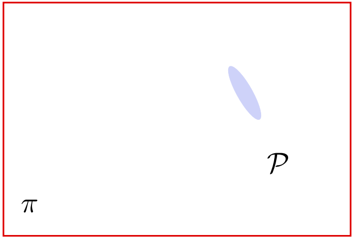
$$\sigma \propto \sqrt{D_{\text{KL}}/n_{\text{live}}} \tag{4}$$

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

One way to do this (REACH):

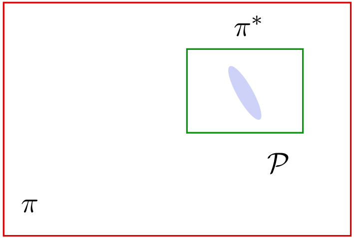


One way to do this (REACH):



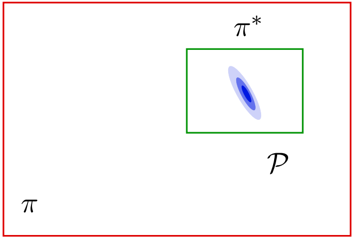

- Perform low resolution (low live points) run first to roughly identify where posterior lies.

One way to do this (REACH):



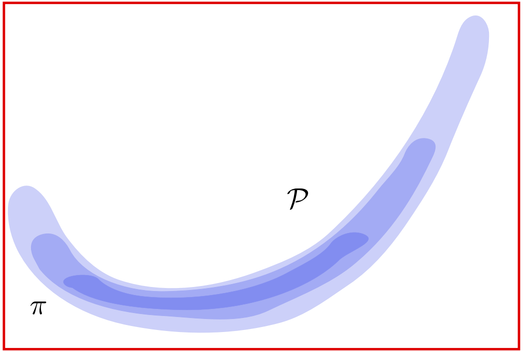

- Perform low resolution (low live points) run first to roughly identify where posterior lies.
- Then set off second, high resolution, run with **narrower** box prior (much quicker).

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 $\frac{V_{\pi^*}}{V_\pi}$)

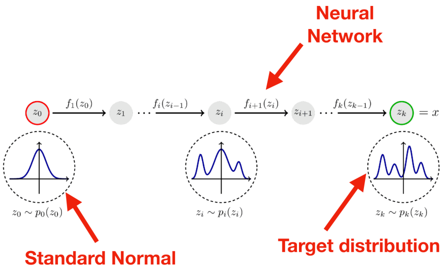
# When does this break down?




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

- Can iterate on this by using **normalizing flows** (NF) to learn the rough posterior.

- Can iterate on this by using **normalizing flows** (NF) to learn the rough posterior.
- NFs perform density estimation, by learning a series of invertible mappings from the standard normal distribution to the target (posterior).



- Use **normalizing flows** (NF) to learn the rough posterior, and use this as our new prior, $\pi^*$.
- 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!

- Use **normalizing flows** (NF) to learn the rough posterior, and use this as our new prior, $\pi^*$.
- 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 - correct likelihood accordingly to get **correct** evidence out. (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

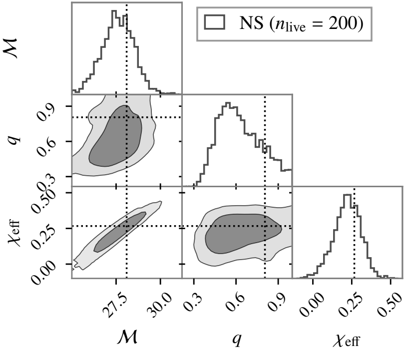
*Article*

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

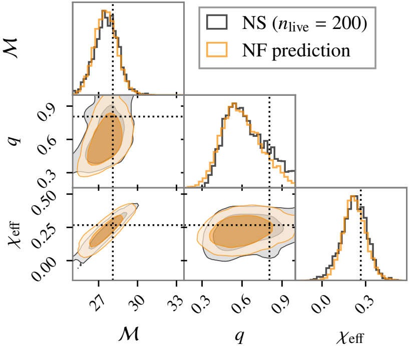
Aleksandr Petrosyan[1,2,3*] & Will Handley[1,2,4†]

# Demo on simulated example

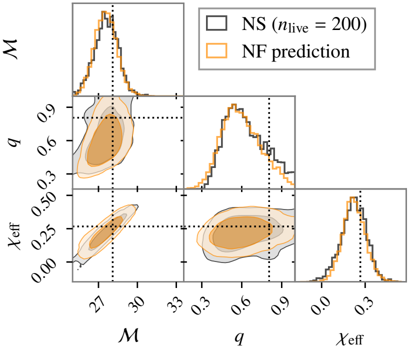
- Perform low resolution run on simulated data.



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

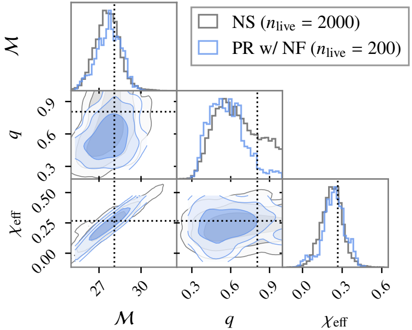


- Perform low resolution run on simulated data.
- Train NF on the weighted samples.
- Use this as ‘repartitioned prior’ for new high resolution run (using PR to also update likelihood accordingly to same evidences and posteriors out).



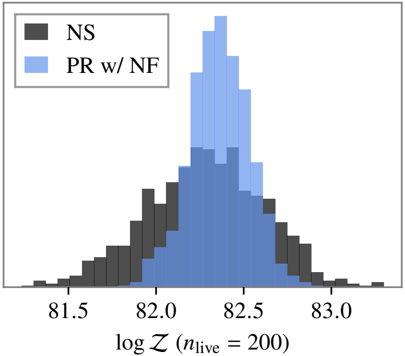
# Demo on simulated example

- 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).



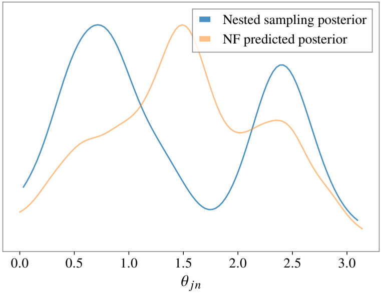
# Demo on simulated example

- 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).

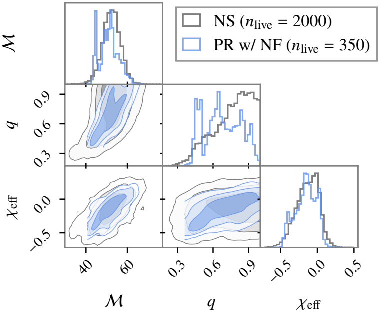


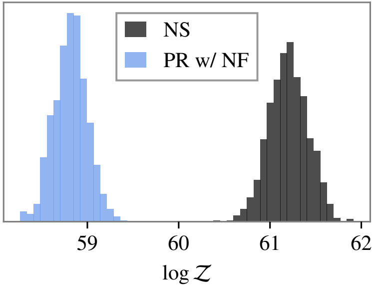

Same answer as doing a full resolution pass of NS, but 7x faster (precision-normalized).

- Still have an issue with multi-modality - if NF only learns one mode, the others can be 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):



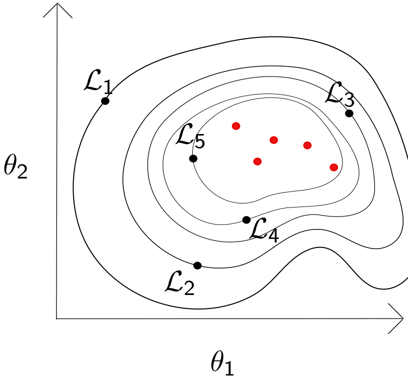
In order to improve the robustness of the method:

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

Nested sampling

Accelerating NS

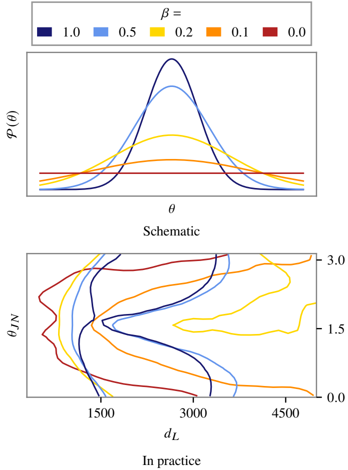
$\beta$-flows




- 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** that can better learn these deep tail events.
- $\beta$-flows are new and only used in this work so far, though broadly applicable.

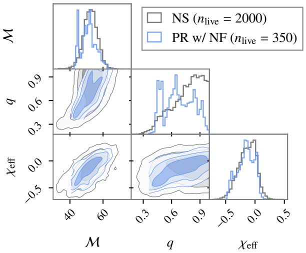
- $\beta$-flow can predict the posterior better in the tails.
- Can **widen themselves adaptively** at runtime.
- Changing a parameter $\beta$, can set repartitioned prior anywhere between posterior ($\beta = 1$) and the original prior ($\beta = 0$).

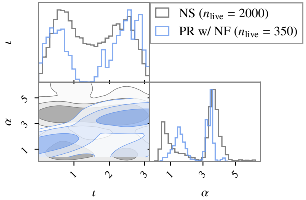
$$p(\beta) \propto \mathcal{L}^{\beta}\pi, \qquad \beta \in [0, 1] \tag{5}$$

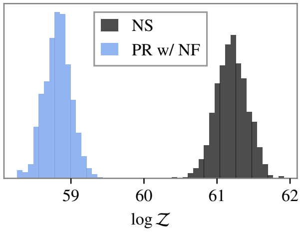


- Using $\beta$-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over $\beta$ now fixes the problem.
- Although $\beta$-flow also doesn't manage to learn the full multi-modality of posterior, we can preferentially sample from lower $\beta$s (i.e. wider distributions that include missed modes).

- Using $\beta$-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over $\beta$ now fixes the problem.
- Although $\beta$-flow also doesn't manage to learn the full multi-modality of posterior, we can preferentially sample from lower $\beta$s (i.e. wider distributions that include missed modes).

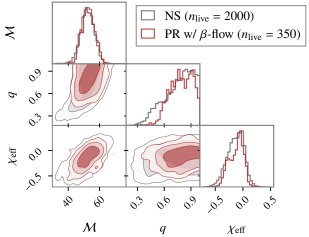


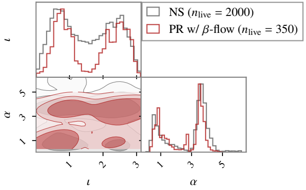



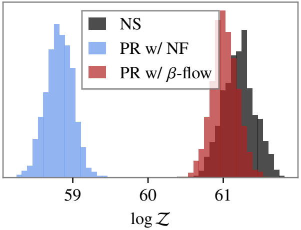



# GW191222 (again)

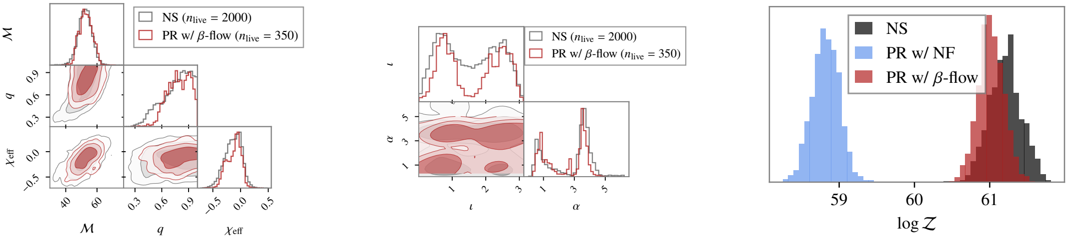
- Using $\beta$-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over $\beta$ now fixes the problem.
- Although $\beta$-flow also doesn't manage to learn the full multi-modality of posterior, we can preferentially sample from lower $\beta$s (i.e. wider distributions that include missed modes).



- Using $\beta$-flows to set the repartitioned prior for the high resolution run, instead of a typical NF, and sampling over $\beta$ now fixes the problem.
- Although $\beta$-flow also doesn't manage to learn the full multi-modality of posterior, we can preferentially sample from lower $\beta$s (i.e. wider distributions that include missed modes).




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

- 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 this 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 this updated prior.
  - Use posterior repartitioning to get correct evidences out, despite changed prior.
  - Can achieve order of magnitude speedups on realistic GW examples.

- Introduced $\beta$-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!

# Posterior repartitioning (PR)

Unlike other sampling algorithms, such as Metropolis-Hastings or Hamiltonian Monte Carlo, NS distinguishes between $\mathcal{L}$ and $\pi$ by 'sampling from the prior, subject to the hard likelihood constraint, $\mathcal{L}$'.

But evidence and posteriors only depend on product of $\mathcal{L}$ and $\pi$:

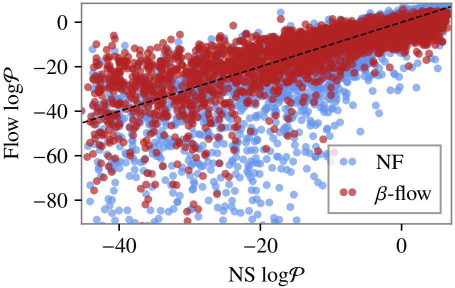
$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta \qquad (6)$$

$$\mathcal{P}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}} \qquad (7)$$

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

$$\tilde{\mathcal{Z}} = \int \tilde{\mathcal{L}}(\theta)\tilde{\pi}(\theta)d\theta = \int \mathcal{L}(\theta)\pi(\theta)d\theta = \mathcal{Z} \qquad (8)$$

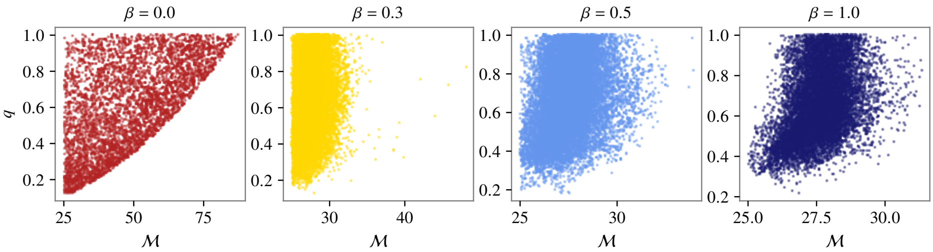
# Better at deep tail probabilities




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

# What is $\beta$?

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



- $\beta$-flow can predict the ($\beta = 1$) posterior, similarly to NF (but better in tails).
- Can predict **any intermediate stage** of NS too.
- We **sample over** $\beta$ at runtime, so proposal can now widen itself adaptively if modes have been missed!