# Frontiers of Nested Sampling

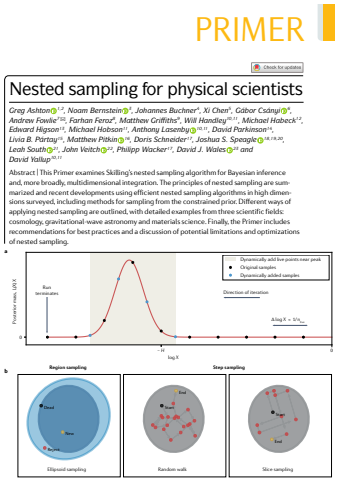
Will Handley
<wh260@cam.ac.uk>

Royal Society University Research Fellow & Turing Fellow
Astrophysics Group, Cavendish Laboratory, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
`github.com/williamjameshandley/talks`

20th July 2022


# Highlight: state-of-the-art Nature review

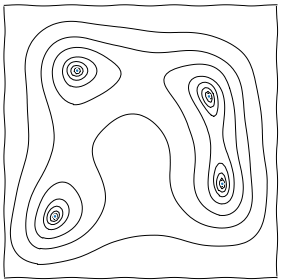
- Invented by John Skilling in 2004
- Recent Nature review primer on nested sampling led by Andrew Fowlie and assembled by the community
- Showcases the current set of tools, and applications from chemistry to cosmology
- In this talk
  - Reminder on theory of nested sampling
  - Updates to the meta algorithm since 2004
  - Updates to the set of tools surrounding nested sampling
  - Future research projects

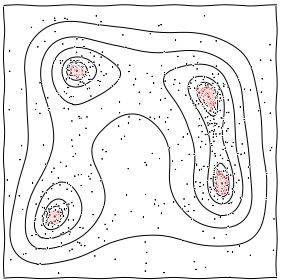


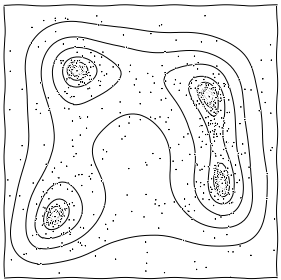
# What is Nested Sampling?

- Nested sampling is a multi-purpose numerical mathematical tool.
- Given a (scalar) function $f$ with a vector of parameters $\theta$, it can be used for:

| Optimisation | Sampling | Integration |
|---|---|---|
| $\theta_{\max} = \max_{\theta} f(\theta)$ | draw $\theta \sim f$ | $\int f(\theta) dV$ |







# MCMC sampling

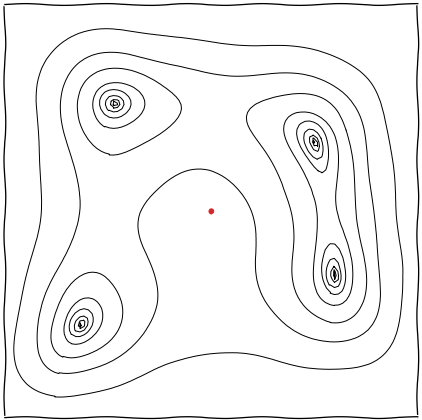
- Markov chain based methods generate samples from posterior distribution by a stepping procedure
- This can get stuck in local peaks
- Cannot compute normalisation $\mathcal{Z}$ of Bayes theorem:

$$P(\theta|D,M) = \frac{P(D|\theta,M)P(\theta|M)}{P(D|M)}$$

$$\mathcal{P} = \frac{\mathcal{L} \times \pi}{\mathcal{Z}} \qquad \text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

- We generally want the evidence $\mathcal{Z} = P(D|M)$ for the second stage of inference: model comparison

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \qquad \text{Science}(M) = \frac{\mathcal{Z}_M \Pi_M}{\sum_m \mathcal{Z}_m \Pi_m}$$

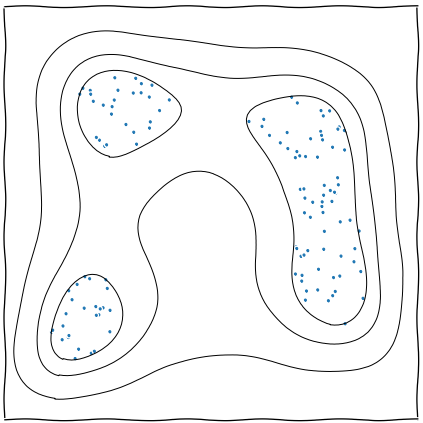


# Nested sampling

- Nested sampling: completely different way to sample
- Ensemble sampling to compress prior to posterior.
- Sequentially update a set $S$ of $n$ samples:
  - $S_0$: Generate $n$ samples uniformly over the space (from the prior $\pi$).
  - $S_{i+1}$: Delete the lowest likelihood sample in $S_i$, and replace it with a new uniform sample with higher likelihood
- Requires one to be able to sample uniformly within a region, subject to a *hard likelihood constraint*:

$$\{\theta \sim \pi : \mathcal{L}(\theta) > \mathcal{L}_*\}$$

- This procedure optimises (multimodally), and can calculate the evidence & posterior weights.



# Mathematics of Nested Sampling

## A probabilistic Lebesgque integrator

- At each iteration, the likelihood contour will shrink in volume by $\approx 1/n$.
- Nested sampling zooms in to the peak of the function $\mathcal{L}$ *exponentially*.

$$\mathcal{Z} \approx \sum_i \Delta\mathcal{L}_i X_i, \quad X_{i+1} \approx \frac{n}{n+1} X_i, \quad X_0 = 1$$

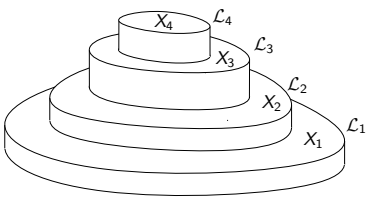
- Although this is only approximate, we can quantify the error

$$P(X_i|X_{i-1}) = \frac{X_i^{n-1}}{nX_{i-1}^n} \times [0 < X_i < X_{i-1}]$$

- Integral can be expressed in one of two ways

$$\mathcal{Z} \approx \sum_i \Delta\mathcal{L}_i X_i = \sum_i \mathcal{L}_i \Delta X_i$$

$$\mathcal{Z} \approx \sum_i X_i \Delta\mathcal{L}_i$$

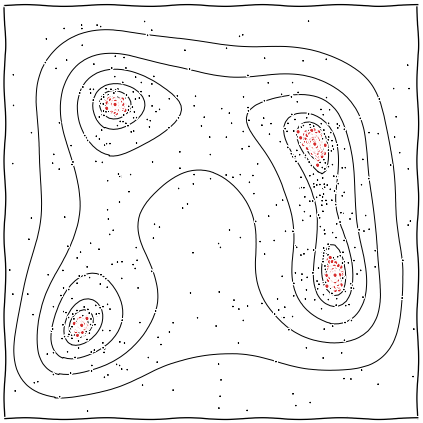


# Dead points: posteriors & evidences

- At the end, one is left with a set of discarded points
- These may be weighted to form weighted posterior samples using $w_i = \mathcal{L}_i \Delta X_i$
- They can also be used to calculate the normalisation $\mathcal{Z} = \sum \mathcal{L}_i \Delta X_i$, or more generally $\sum_i f(\mathcal{L}_i) \Delta X_i$.
  - Nested sampling probabilistically estimates the volume of the parameter space

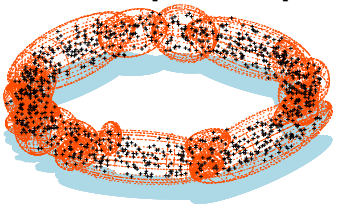
$$X_i \approx \left(\frac{n}{n+1}\right) X_{i-1} \quad \Rightarrow \quad X_i \approx \left(\frac{n}{n+1}\right)^i \approx e^{-i/n}$$

  - only statistical estimates, but we know the error bar
  - Nested sampling thus estimates the density of states
  - it is therefore a partition function calculator
- The evolving ensemble of live points allows algorithms to perform self-tuning and mode clustering.

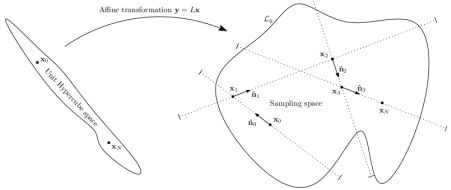


# Implementations of Nested Sampling

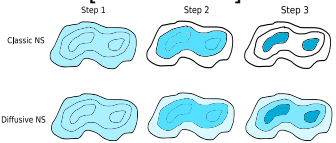
`MultiNest` [0809.3437]



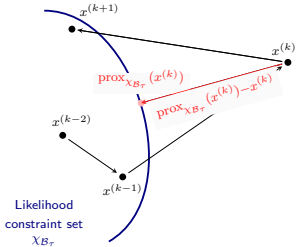
`PolyChord` [1506.00171]



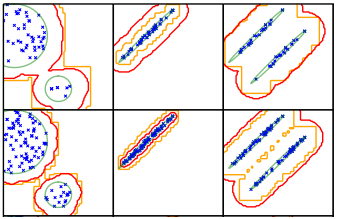

`DNest` [1606.03757]



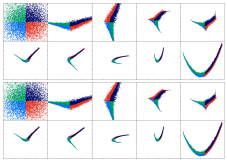

`ProxNest` [2106.03646]

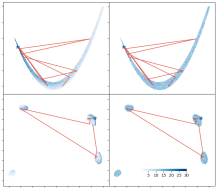



`UltraNest` [2101.09604]



`NeuralNest` [1903.10860]





`dynesty` [1904.02180]

# Types of nested sampler

- Broadly, most nested samplers can be split into how they create new live points
- i.e. how they sample from the hard likelihood constraint $\{\theta \sim \pi : \mathcal{L}(\theta) > \mathcal{L}_*\}$

## Rejection samplers

- e.g. `MultiNest`, `UltraNest`
- Constructs bounding region and draws many invalid points until one is found within $L_*$.
- Efficient in low dimensions, exponentially inefficient $\sim \mathcal{O}(e^{d/d_0})$ in high $d > d_0 \sim 10$

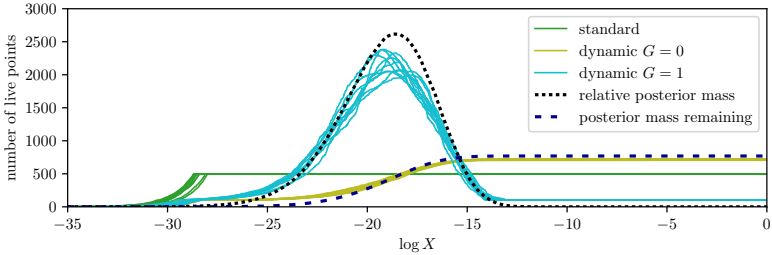
## Chain-based samplers

- e.g. `PolyChord`, `NeuralNest`, `ProxNest`
- Run Markov chain starting at a live point, generating many valid (correlated) points.
- Linear $\sim \mathcal{O}(d)$ penalty in decorrelating new live point from the original seed point

- Nested samplers usually come with
  - *resolution* parameter $n_{\text{live}}$ (which improve results as $\sim \mathcal{O}(n_{\text{live}}^{-1/2})$
  - set of *reliability* parameters [2101.04525], which don't improve results if set arbitrarily high, but introduce systematic errors if set too low.
  - e.g. `Multinest` efficiency `eff` or `PolyChord` chain length `num_repeats`

# Dynamic nested sampling

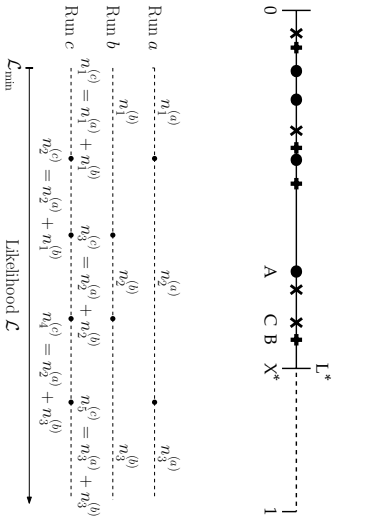
- Small change to meta-algorithm:
- Allow the number of live points to vary at run time [1704.03459]




- Separate creation and deletion:
  - $S_0$: Generate $n$ samples uniformly over the space (from the prior $\pi$).
  - $S_{i+1}$:
    - Delete the lowest likelihood sample in $S_i$ with criterion $D_i$
    - Create a new uniform sample with higher likelihood with criterion $C_i$
- Extremely straightforward to implement (just let number of live points $n_i$ vary with $i$)
- Need to specify creation and deletion criteria
- This can prove useful, e.g.
  - Killing off all remaining live points is equivalent to usual correction term
  - Oversampling the prior by `nprior`
- However, it is "exactly the right level of complexity" to get the uninitiated excited, and does not result in dramatic speedups.

# Nested sampling post-processing: Weaving and unweaving runs

- There is now a substantial literature on what you can do with a nested sampling run
- John Skilling originally noted that two nested sampling runs can be "merged":
  - Take two nested sampling runs on likelihood $\mathcal{L}$ with $n$ live points and $m$ live points each
  - Concatenate the dead points, and re-sort on likelihood
  - The resulting set of points are dead points from an $n+m$-live point run
- To generalise to dynamic nested sampling one needs to record more information

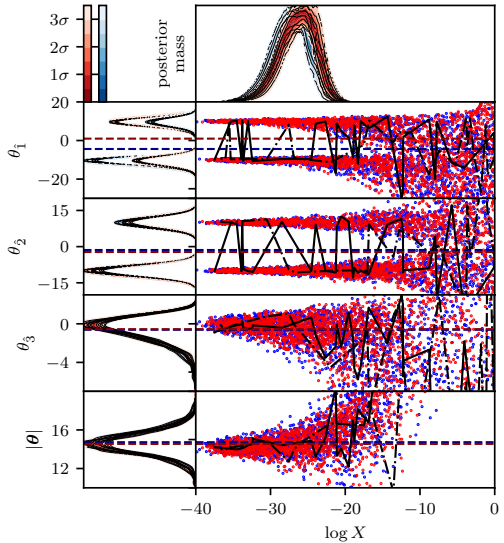


# Birth $\mathcal{L}_*$ and death $\mathcal{L}$ contours

- A nested sampling run can be losselessly reconstructed with an extra column.
- Namely the set of *birth contours*, i.e. the $\mathcal{L}_*$ at which each point was born at
- With the lossless compression of columns $\mathcal{L}$ and $\mathcal{L}_*$, one can
  - Compute the dynamic number of live points
  - Reconstruct the nested sampling run history
  - Decompose an $n$-live point run into $n$ single live point runs.
- Single live point runs form the indivisible unit of nested sampling
- This post processing suite is encapsulated in the continuously integrated python packages of `anesthetic` [1905.04768] and `nestcheck` [1804.06406]

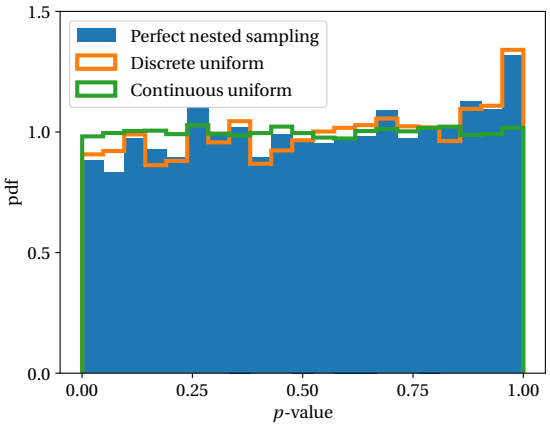
# Parameter cross-checks

- "Indivisible unit" of nested sampling identified as the *single live point run* (or *thread*)
- When a nested sampling run is unwoven, it can be recombined into smaller runs.
- Cross-checks such as bootstrap resampling can be applied to determine if these are consistent.
- This can be used to quantify the residual uncertainty in parameter estimation weights $w_i \approx L_i \Delta X_i$ [1704.03459].
  - In addition to Poisson uncertainty on $X_i$, there is also uncertainty associated with picking $\theta_i$ as representative of the entire $L_i$ contour [0801.3887].



# Insertion indices and order statistics

- At each iteration of nested sampling we generate a new live point and insert it into the list of live points sorted by loglikelihood
- IF we have done things correctly, this should obey *order statistics*
- if it doesn't, our nested sampler is not drawing live points corectly;
- We demonstrate this with KS $p$-values [2006.03371], to test reliability parameters
- should be extended to be more Bayesian
- Needs extending to dynamic $n_{\text{live}}$ case
- This can be used at run-time to tune reliability parameters



# Frontier: Insertion indices

- For the purposes of estimating volume in a statistical way, we discard the likelihood information, focussing on the ordering of the contours.
- Traditional nested sampling uses the fact that

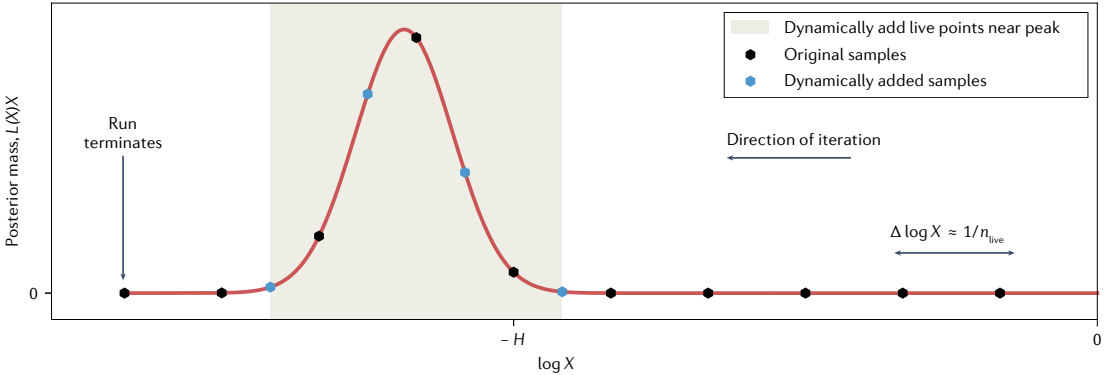
$$P(X_j|X_{j-1}, n_{\rm live}) = \frac{n_j}{X_{j-1}}\left(\frac{X_j}{X_{j-1}}\right)^{n_j-1}[0 < X_j < X_{j-1}].$$

- marginalised out dependency on everything other than $X_{j-1}$ & compressed into a vector encoding the number of live points at each iteration $n_i$.
- **Frontier**: is "Skilling compression" $(\mathcal{L}, \mathcal{L}_*) \to n$ lossless or lossy for the purposes of volume estimation?
- The results presented in [2006.03371] are suggestive that it is losing some useful information, as insertion indexes do provide further information in the context of a cross check (and are in fact a lossless compression of the birth and death contours).
- One possibility is that the Skilling compression is lossless in the context of perfect nested sampling, but if a run is biased then you may be able to use insertion indexes to partially correct a biased run.

# Reversible nested sampling

- One of the issues preventing nested sampling scaling to millions of dimensions is the need to compress from prior to posterior in all parameters
  - c.f. Skilling's argument that the Entropy/KL divergence is much greater than the width of the typical set/posterior bulk: $\mathcal{D} \sim d \gg \sqrt{d}$
- One could in principle reverse the direction of travel, and move outward from a peak.
- If one could guarantee that all peak looks gaussian close enough in (c.f. Laplace approximation), then one can estimate the final volume $X_N$, and reverse the usual argument.
- This could in principle be used to dramatically reduce the poisson error if one could estimate the volume in the final set of live points geometrically.

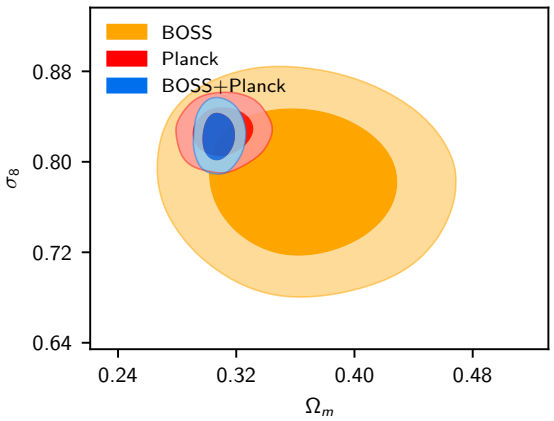
# AEONS: Approximating the end of nested sampling




- A seasoned user knows when nested sampling is approaching the end of an HPC run
- One knows that as $\Delta \log \mathcal{Z}$ approaches unity, and there are only "a few" nested sampling iterations remaining before posterior is crossed
- **Frontier:** Can we quantify this using Gaussian approximations to make a rough progress bar (with uncertainty quantification) – summer student working on this.

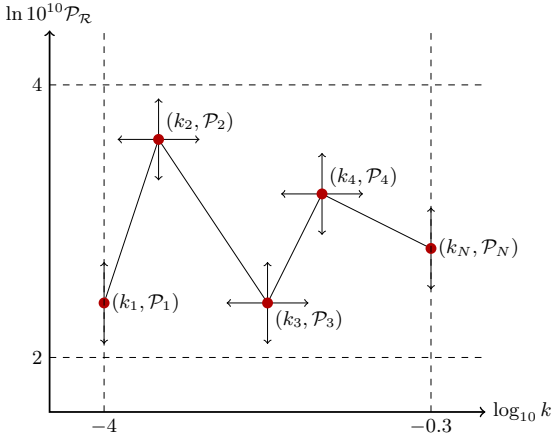
# Importance nested sampling

- Importance sampling is the procedure:
  - given a set of (weighted) samples $\{(w_i, \theta_i)\}$ drawn from posterior distribution $\mathcal{P}(\theta)$
  - If the likelihood $\mathcal{L}_0(\theta) = P(D_0|\theta)$ is updated with some additional data $D_1$, such that $\mathcal{L}_0 \to \mathcal{L}_0 \times \mathcal{L}_1$ [1902.04029]
  - Can we re-weight $w_i$ without re-running a Markov Chain?
- well-established for MCMC $w_i \to w_i \times L_1(\theta_i)$
- For nested sampling, evidences are transformable $\mathcal{Z}_0 \to \mathcal{Z}_0 \times \langle L_1 \rangle_{\mathcal{P}_0}$
- Can we do better?
- **Frontier:** Is there a way to re-weight/thin a nested sampling run to recover the equivalent nested sampling run with a new likelihood?



# Transdimensional nested sampling

- In some applications it is useful to consider parameter spaces where the number of active parameters vary
  - Object detection [0809.3437]
  - Free-form reconstruction using `FlexKnots` [1908.00906]
- At the moment, reasonable performance can be achieved by letting $N$ be a parameter (up to $N_{\max}$) and then ignoring unused parameters [1506.09024].
- Brendon Brewer [1411.3921] built some examples of RJMCMC diffusive nested sampling, but very problem specific.
- **Frontier:** Is there an ensemble-based methodology for transdimensional NS?

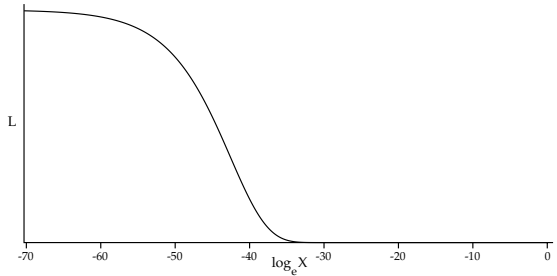


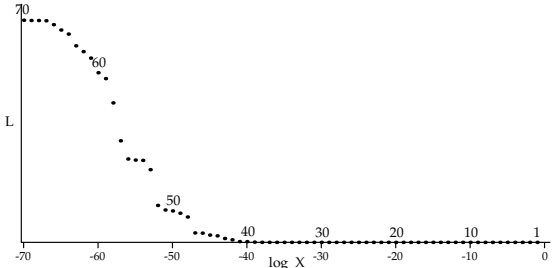
# Frontier: Multi-objective nested sampling

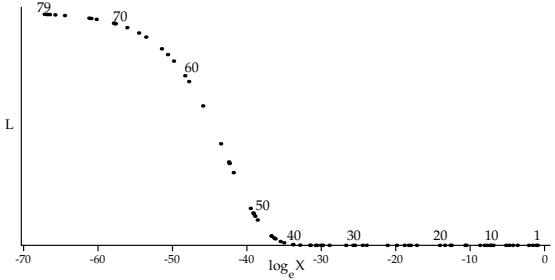
- Nested sampling is useful as a pure optimiser, particularly for industrial applications where they lack likelihoods but have objective functions.
- It still provides a unique statistical interpretation for an arbitrary function $f(\theta)$ by providing a "parameter compression" $X_i \approx e^{i/n_{\text{live}}} \leftrightarrow e^{\sum_j^i 1/n_j}$ for each value of $f_i$
- Also useful for providing a collection of solutions fairly distributed in parameter space once the desired optimum has been met
- This collection of solutions is useful for rudimentary uncertainty quantification, but also for multi-objective optimisation
- **Frontier**: Is there a more in-built multi-objective optimiser. Can nested sampling optimise two objective functions simultaneously up to a given compression?

# Likelihood values

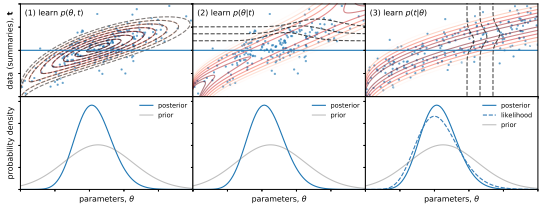
- One of the virtues of nested sampling is that it only uses the ordering of likelihood values
- **Frontier:** if one made a smoothness assumption, e.g. $\frac{d \log \mathcal{L}}{d \log X} \approx \text{constant}$, is is possible to do better?
- This could dramatically reduce the poisson error, runtime and efficiency of NS



# Nested Sampling with Likelihood Free Inference




Alsing *et al.* [1903.00007]

- In density estimation likelihood free inference, the output is to learn one/all of:

  Likelihood $P(D|\theta)$
  Posterior $P(\theta|D)$
  Joint $P(D, \theta)$

- In the first instance, nested sampling can be used to scan these learnt functions

- Data are compressed, so joint space $(D, \theta)$ is navigable by off-the-shelf codes.
  - Sanity checking the solution
  - Computing evidences/Kullback Liebler divergences from likelihoods
- Its self-tuning capacity and ability to handle multi-modal distributions can be very useful for diagnosing incompletely learnt functions
- Emulated likelihoods (e.g. normalising flows) are generally fast, so can deploy more likelihood hungry techniques like NS.
- In principle can use it to train emulators by marginalisation rather than maximisation.

# Nested Sampling for Approximate Bayesian Computation/SBI

- Assume one has a generative model capable of turning parameters into mock data $D(\theta)$
- Given infinite computing power, ABC works by selecting $\{\theta : D(\theta) = D_{\text{observed}}\}$
- These are samples from the posterior, without using a likelihood.
- In practice $D = D_{\text{obs}}$ becomes $D \approx D_{\text{obs}}$
- i.e. $|D - D_{\text{obs}}| < \varepsilon$, or more generally $\boxed{\rho(D, D_{\text{obs}}) < \varepsilon}$, where $\rho$ is some suitably chosen objective function
- Main challenges are
  1. Choice of $\rho$/summary stats
  2. Choice of $\varepsilon$ schedule
  3. Rejection sampling

- Nested sampling fits this well: In principle, can just change the usual hard likelihood constraints $\{\theta \sim \pi : \mathcal{L}(\theta) > \mathcal{L}_*\}$ to

$$\{\theta \sim \pi : \rho(D(\theta), D_{\text{obs}}) < \varepsilon\}$$

(Brewer & Foreman-Mackey [1606.03757])

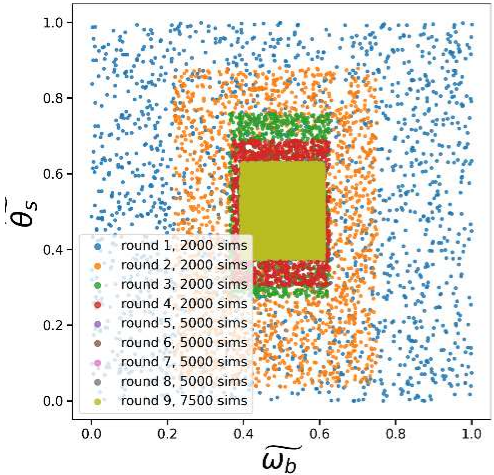
- Ongoing work with Andrew Fowlie & Sebastian Hoof
  - How to deal with nondeterminism
  - How to interpret $\rho$ as a "likelihood"
  - How to interpret the evidence $\mathcal{Z}$

# Nested sampling for truncated methods

- Will hear more on this tomorrow from Christoph
- Many Likelihood implicit approaches at the moment have some element of sampling direct from the prior
- Inefficient if number of parameters $> \mathcal{O}$(a few)
- Can get round this by truncating to region:

$$\Gamma\{\theta \in \mathrm{supp}p(\theta) \mid p(\theta|x_0) > \tilde{\varepsilon}\}$$

- At the moment regions defined by nested boxes
- **Frontier:** This seems ripe for replacement by NS – PhD student currently working on this (Kilian Scheutwinkel)




Cole *et al.* [2111.08030]

# Conclusions

- Nested sampling is a truly unique tool

Watch out for:

Johannes Buchner (next) Analysing chain based samplers

Aleksandr Petrosyan (before coffee) Accelerating nested sampling

Livia Partay (after coffee) Nested sampling in materials scienc

Harry Bevins (Friday before lunch) Nested sampling and normalising flows

# How does Nested Sampling compare to other approaches?

- In all cases:
  - \+ NS can handle multimodal functions
  - \+ NS computes evidences, partition functions and integrals
  - \+ NS is self-tuning/black-box

Modern Nested Sampling algorithms can do this in $\sim \mathcal{O}(100s)$ dimensions

## Optimisation

- Gradient descent
  - \+ NS does not require gradients
- Genetic algorithms
  - \+ NS discarded points have statistical meaning

## Sampling

- Metropolis-Hastings?
  - – Very little beats a well-tuned, customised MH
  - \+ NS is self tuning
- Hamiltonian Monte Carlo?
  - – In millions of dimensions, HMC is king
  - \+ NS does not require gradients

## Integration

- Thermodynamic integration
  - \+ protective against phase trasitions
  - \+ No annealing schedule tuning
- Sequential Monte Carlo
  - – Some people (SMC experts) classify NS as a kind of SMC
  - \+ NS is athermal

# Nested Sampling: a user's guide

1. Nested sampling is a likelihood scanner, rather than posterior explorer.
   - This means typically most of its time is spent on burn-in rather than posterior sampling
   - Changing the stopping criterion from $10^{-3}$ to 0.5 does little to speed up the run, but can make results very unreliable
2. The number of live points $n_{\mathrm{live}}$ is a resolution parameter.
   - Run time is linear in $n_{\mathrm{live}}$, posterior and evidence accuracy goes as $\frac{1}{\sqrt{n_{\mathrm{live}}}}$.
   - Set low for exploratory runs $\sim \mathcal{O}(10)$ and increased to $\sim \mathcal{O}(1000)$ for production standard.
3. Most algorithms come with additional reliability parameter(s).
   - e.g. `MultiNest`: eff, `PolyChord`: $n_{\mathrm{repeats}}$
   - These are parameters which have no gain if set too conservatively, but increase the reliability
   - Check that results do not degrade if you reduce them from defaults, otherwise increase.