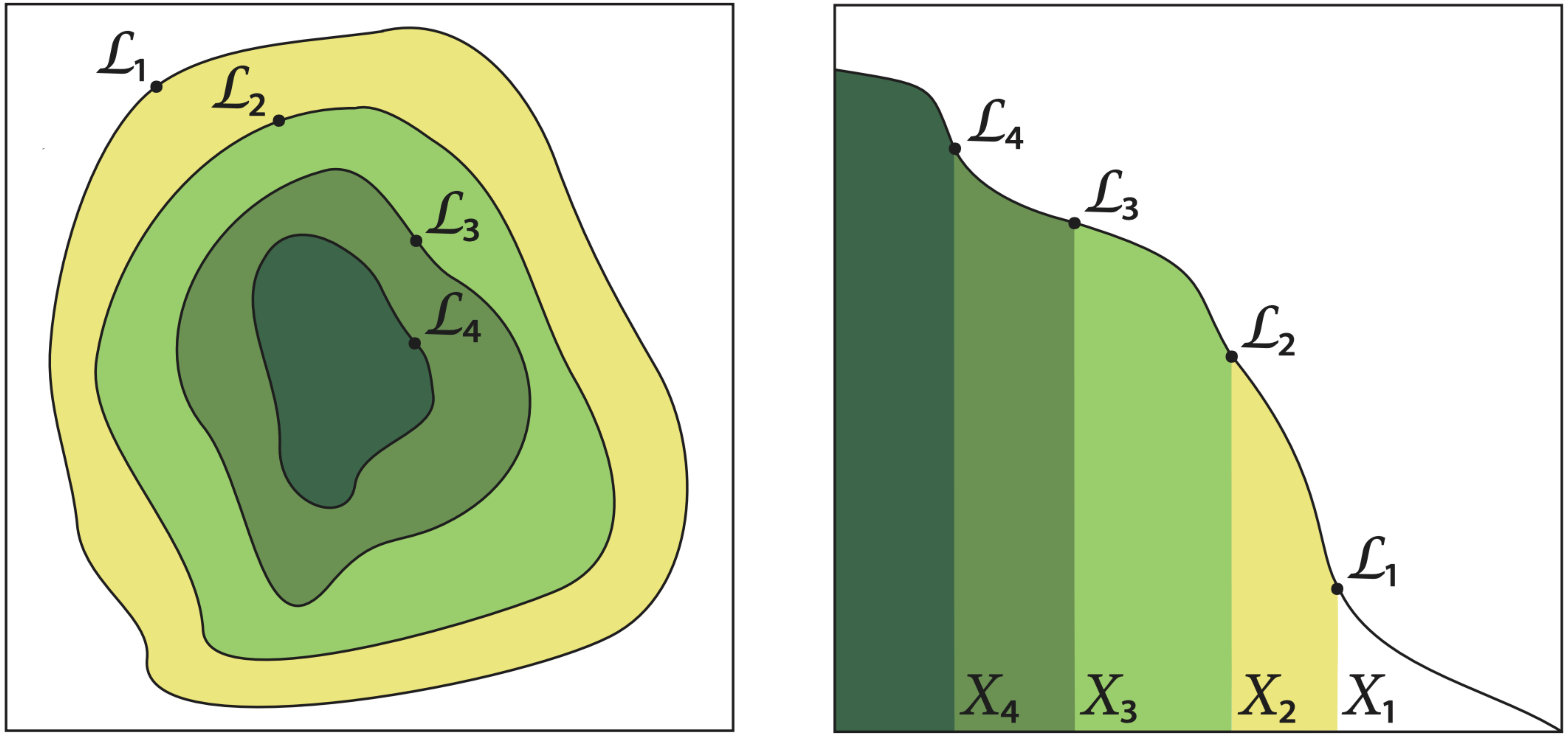
NESTED SAMPLING
A review of theory and implementations

Rencontre du group de travail “méthodes d’analyse des données”
du GdR Ondes Gravitationnelles @ IP2I Lyon - 15/11/22



Credits: Feroz et al., (2013)

Danny Laghi
CNES Postdoctoral Fellow @ L2IT

(EXTRA-) QUICK PREAMBLE ON MCMC

BAYES' THEOREM

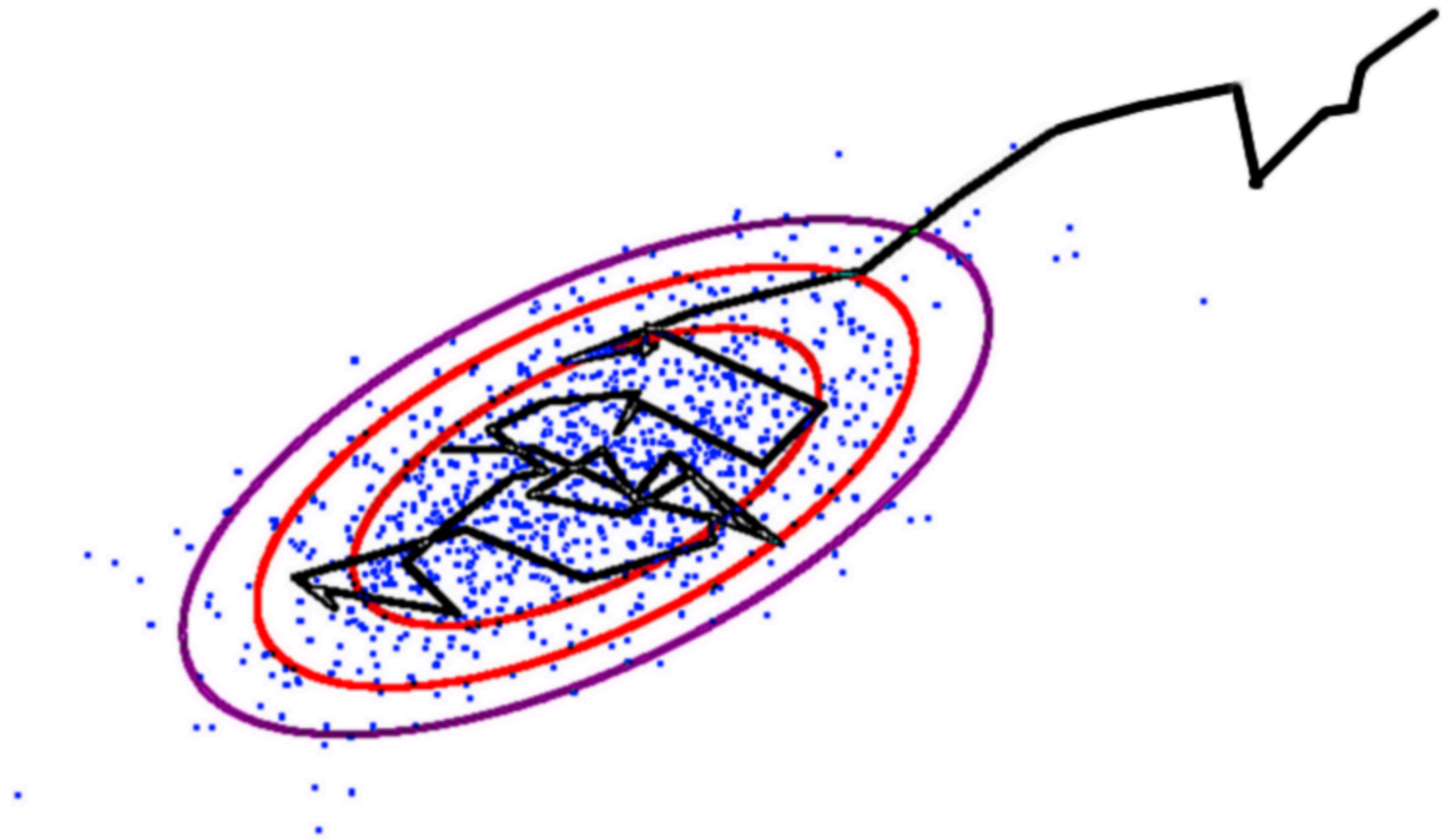
$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$$

Posterior Likelihood Prior

$$P(\Theta \mid D, H, I) \propto P(D \mid \Theta, H, I)\, P(\Theta \mid H, I)$$

MCMC allows to draw samples from the posterior distribution

MARKOV CHAIN MONTE CARLO



- Estimate the posterior by **stochastically wandering** through the parameter space

- Distribute **samples** $\propto$ density of **target** posterior distribution

- E.g. **Metropolis-Hastings** algorithm:
    - Given a starting point Θ, use a **proposal density function** $Q(\Theta' | \Theta)$ to draw a **new** sample Θ' which can only depend on the **current** sample Θ

    - New proposal accepted with **probability** $r_s = \min\left(1, \frac{p(\Theta' | D, H)\, Q(\Theta | \Theta')}{p(\Theta | D, H)\, Q(\Theta' | \Theta)}\right)$

      Hastings

      detailed balance

    - If accepted, add Θ' to the chain, otherwise Θ is repeated

MCMC LIMITATIONS & OPTIMISATIONS

- Start chains from **random location** in parameter space
    - **Discard** initial samples (**burn-in** period) in order to lose dependence of initial location

- If we want **statistically independent** samples, remove **correlation** between adjacent samples in the chain:
    - **Thin** each chains by its integrated **autocorrelation time** (ACT)

- Samples left after burn-in and ACT thinning are the **effective samples**

- Run **parallel chains** to increase the **number** of effective samples

MCMC LIMITATIONS & OPTIMISATIONS

- **Efficiency** of Metropolis-Hastings strongly depends on the **choice of proposal density**, e.g. Gaussian centred on Θ (the choice of σ affects the acceptance rate)

- For complicated multi-modal target distributions:
    - **Parallel tempering MCMC**

$$P_T(\Theta \mid D) \;\propto\; P(D \mid \Theta, H, I)^{\frac{1}{T}} \; P(\Theta \mid H, I)$$

- Increasing T **"flattens"** the posterior and **broadens** peaks: easier to sample
- As $T \to \infty$, the posterior becomes the prior
- Construct ensemble of **tempered chains** from $T \in [1, T_{\max}]$
- **High-T chains** sample a distribution closer to the prior: **easier to explore** the parameter space and move between modes
- **Pass information** about regions of high posterior support found from the high-T chains to increase sampling efficiency of $T = 1$ chain by periodically proposing **swaps** in the locations of adjacent chains.

MCMC vs NS

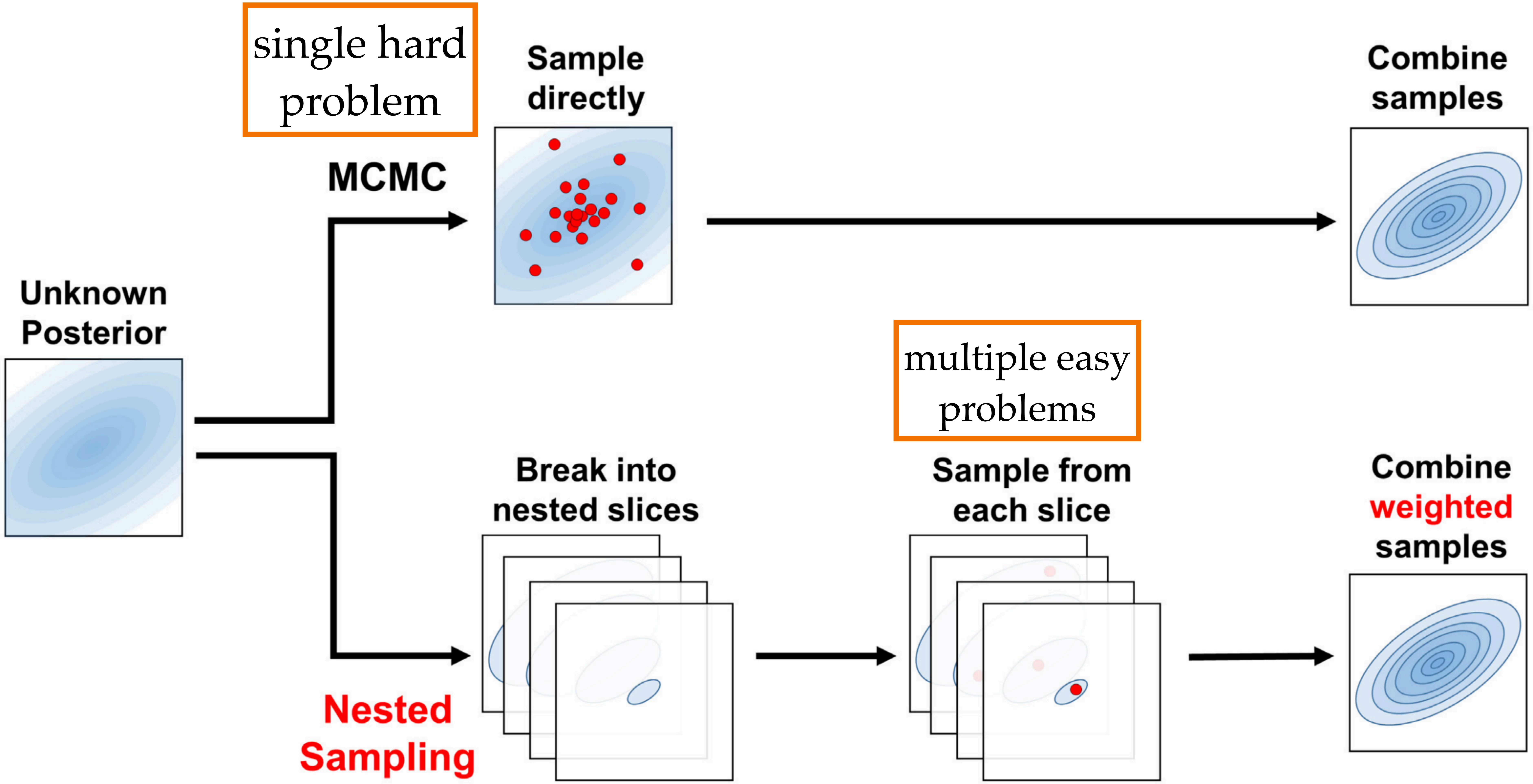


Figure 1. A schematic representation of the different approaches MCMC methods and nested sampling methods take to sample from the posterior. While MCMC methods attempt to generate samples directly from the posterior, nested sampling instead breaks up the posterior into many nested 'slices', generates samples from each of them, and then recombines the samples to reconstruct the original distribution using the appropriate weights.

Speagle, *MNRAS* (2020)

NESTED SAMPLING

OVERVIEW

- **Nested sampling (NS) in a nutshell**

- Main challenges and limitations

- Implementations & distributions: what's out there

MAIN REFERENCES

- **J. Skilling**, “Nested sampling for general Bayesian computation”
  Bayesian Anal. 1(4): 833-859 (2006)

Bayesian Analysis (2006) **1**, Number 4, pp. 833–860

Nested Sampling for General Bayesian Computation

John Skilling*

- **G. Ashton et al.**, “Nested sampling for physical scientists”
  Nature Rev. Meth. Prim. 2, 39 (2022), arXiv:2205.15570

[and references therein]

Primer | Published: 26 May 2022

Nested sampling for physical scientists

Greg Ashton, Noam Bernstein, Johannes Buchner, Xi Chen, Gábor Csányi, Andrew Fowlie, Farhan Feroz, Matthew Griffiths, Will Handley, Michael Habeck, Edward Higson, Michael Hobson, Anthony Lasenby, David Parkinson, Livia B. Pártay, Matthew Pitkin, Doris Schneider, Joshua S. Speagle, Leah South, John Veitch, Philipp Wacker, David J. Wales & David Yallup

Nature Reviews Methods Primers **2**, Article number: 39 (2022) | Cite this article

QUICK FACTS ABOUT NS

- **NS** is primarily an algorithm to integrate challenging high-dimensional integrals

- In Bayesian inference, the difficult integral we want to compute is the “evidence”

- As a by-product of this computation, we also obtain posterior samples

BAYES' THEOREM

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$$

Likelihood, Prior, Posterior, Evidence

$$\underbrace{P(\Theta \mid D, H, I)}_{\text{Posterior}} = \frac{\overbrace{P(D \mid \Theta, H, I)}^{\text{Likelihood}} \; \overbrace{P(\Theta \mid H, I)}^{\text{Prior}}}{\underbrace{P(D \mid H, I)}_{\text{Evidence}}}$$

BAYES' THEOREM

$$\underset{\text{Posterior}}{P(\Theta \mid D)} = \frac{\overset{\text{Likelihood}}{\mathscr{L}(\Theta)} \; \overset{\text{Prior}}{\pi(\Theta)}}{\underset{\text{Evidence}}{Z}}$$

BAYES' THEOREM

Posterior $\quad$ Likelihood $\quad$ Prior

$$P(\Theta \mid D) = \frac{\mathcal{L}(\Theta)\,\pi(\Theta)}{Z}$$

$$\text{Evidence} = \int_{\Omega_\Theta} \mathcal{L}(\Theta)\,\pi(\Theta)\,d\Theta$$

N-dim integral over an N-D parameter space

BAYES' THEOREM

$$\underset{\text{Posterior}}{P(\Theta \,|\, D)} = \frac{\overset{\text{Likelihood}}{\mathscr{L}(\Theta)} \quad \overset{\text{Prior}}{\pi(\Theta)}}{\underset{\text{Evidence}}{Z}}$$

$$\text{Evidence} = \int_0^1 \tilde{L}(X)\, dX$$

$\tilde{L} : [0,1] \rightarrow \mathbb{R}^+$

NS transforms it into a 1-D integral

NS: STEP 1/3

Introduce the **prior volume**:

$$X(\lambda) = \int_{\Omega_\Theta : \mathscr{L}(\Theta) \geq \lambda} \pi(\Theta)\, d\Theta$$

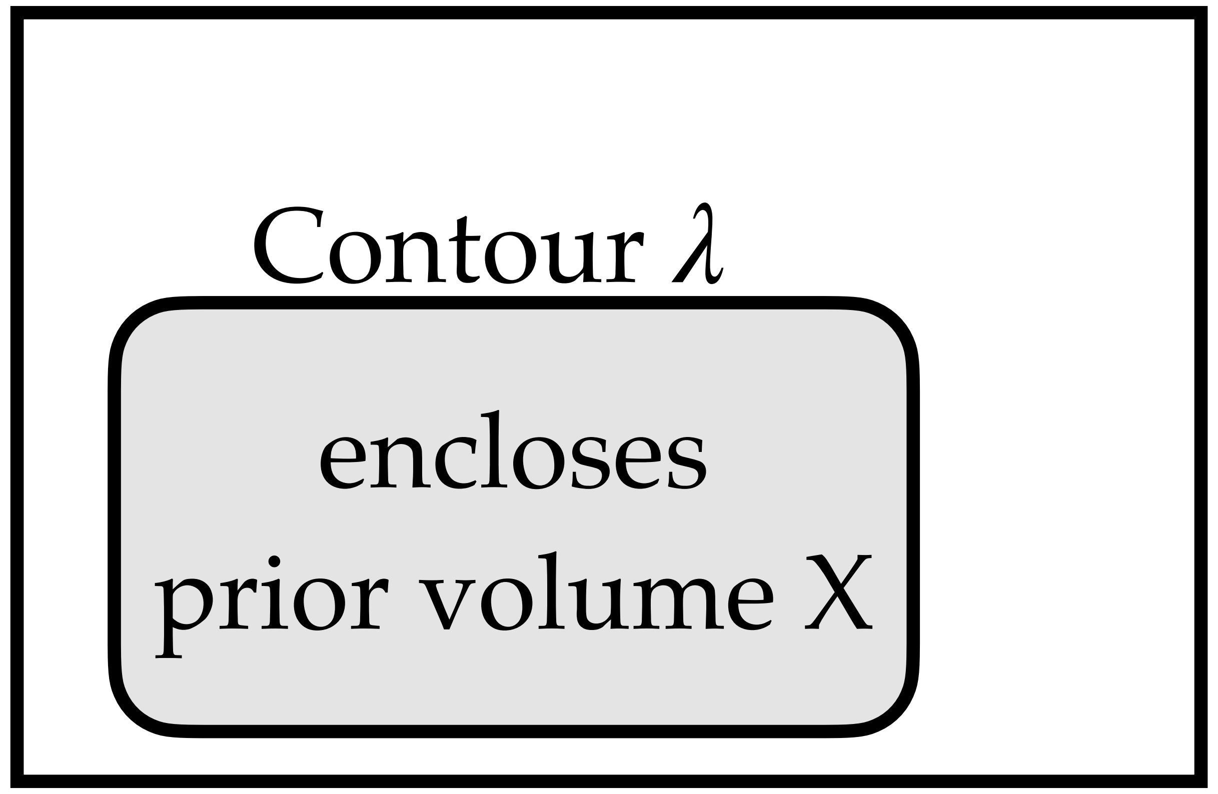
$\mathscr{L}(\Theta) \geq 0$

$\lambda \in [0,\infty)$

$X \in (0,1]$

$X(\lambda)$ = amount of prior probability with likelihood greater than λ

= tot. prob. vol. contained within a iso-likelihood contour def. by λ



$X(0) = 1$

$X(\infty) = 0 \quad \text{if } \exists!\, \mathscr{L}_{max}$

NS: STEP 2/3

$$X(\lambda) = \int_{\Omega_\Theta : \mathscr{L}(\Theta) \geq \lambda} \pi(\Theta)\, d\Theta \equiv \int_{\Omega_\Theta} \pi_\lambda(\Theta)\, d\Theta$$

Constrained prior: $\pi_\lambda(\Theta) = \begin{cases} \pi(\Theta)/X(\lambda) & \text{if } \mathscr{L}(\Theta) \geq \lambda \\ 0 & \text{otherwise} \end{cases}$

$$Z = \int_{\Omega_\Theta} \mathscr{L}(\Theta)\pi(\Theta)\, d\Theta = \int_0^\infty X(\lambda)\, d\lambda$$

NS: STEP 3/3

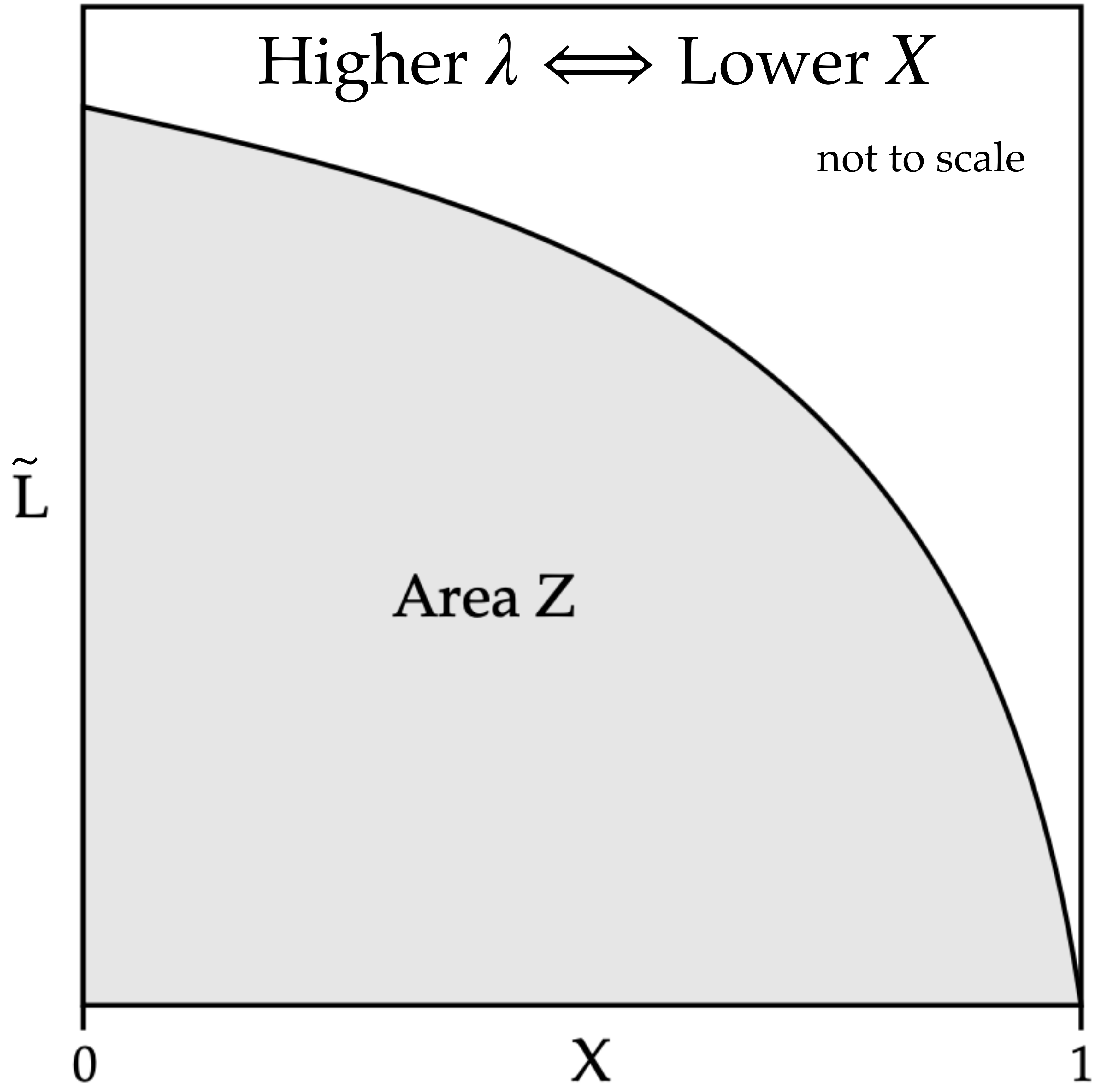
Define $\tilde{L}(X)$ as the **inverse** of the prior volume $X(\mathcal{L}(\Theta) = \lambda)$: $\tilde{L}(X(\lambda)) = \lambda$

$\tilde{L}(X)$ is a **monotonically decreasing** function of X

$$Z = \int_{\Omega_\Theta} \mathcal{L}(\Theta)\pi(\Theta)\, d\Theta$$

$$= \int_0^\infty X(\lambda)\, d\lambda$$

$$= \int_0^1 \tilde{L}(X)\, dX$$



Skilling (2006)

NS: STEP 3/3

Define $\tilde{L}(X)$ as the **inverse** of the prior volume $X(\mathscr{L}(\Theta) = \lambda)$: $\tilde{L}(X(\lambda)) = \lambda$

$\tilde{L}(X)$ is a **monotonically decreasing** function of X

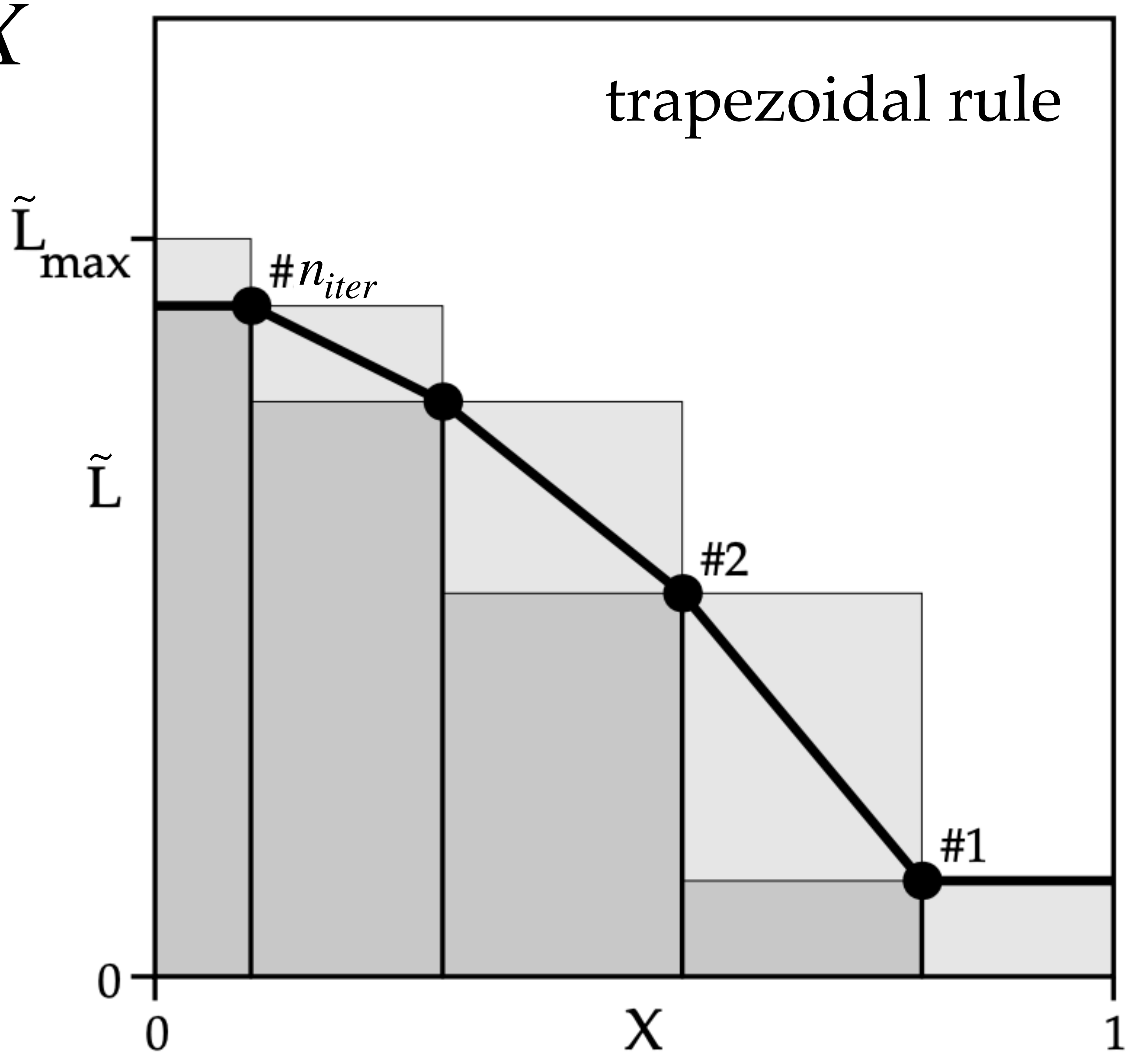
$$Z = \int_{\Omega_\Theta} \mathscr{L}(\Theta)\pi(\Theta)\, d\Theta$$

$$= \int_0^\infty X(\lambda)\, d\lambda$$

$$\simeq \sum_{i=1}^{n_{iter}} \frac{\tilde{L}_i}{2}(X_{i-1} - X_{i+1})$$

$\tilde{L}_i$ computable

X_i uncertain!



Skilling (2006)

NS ALGORITHM

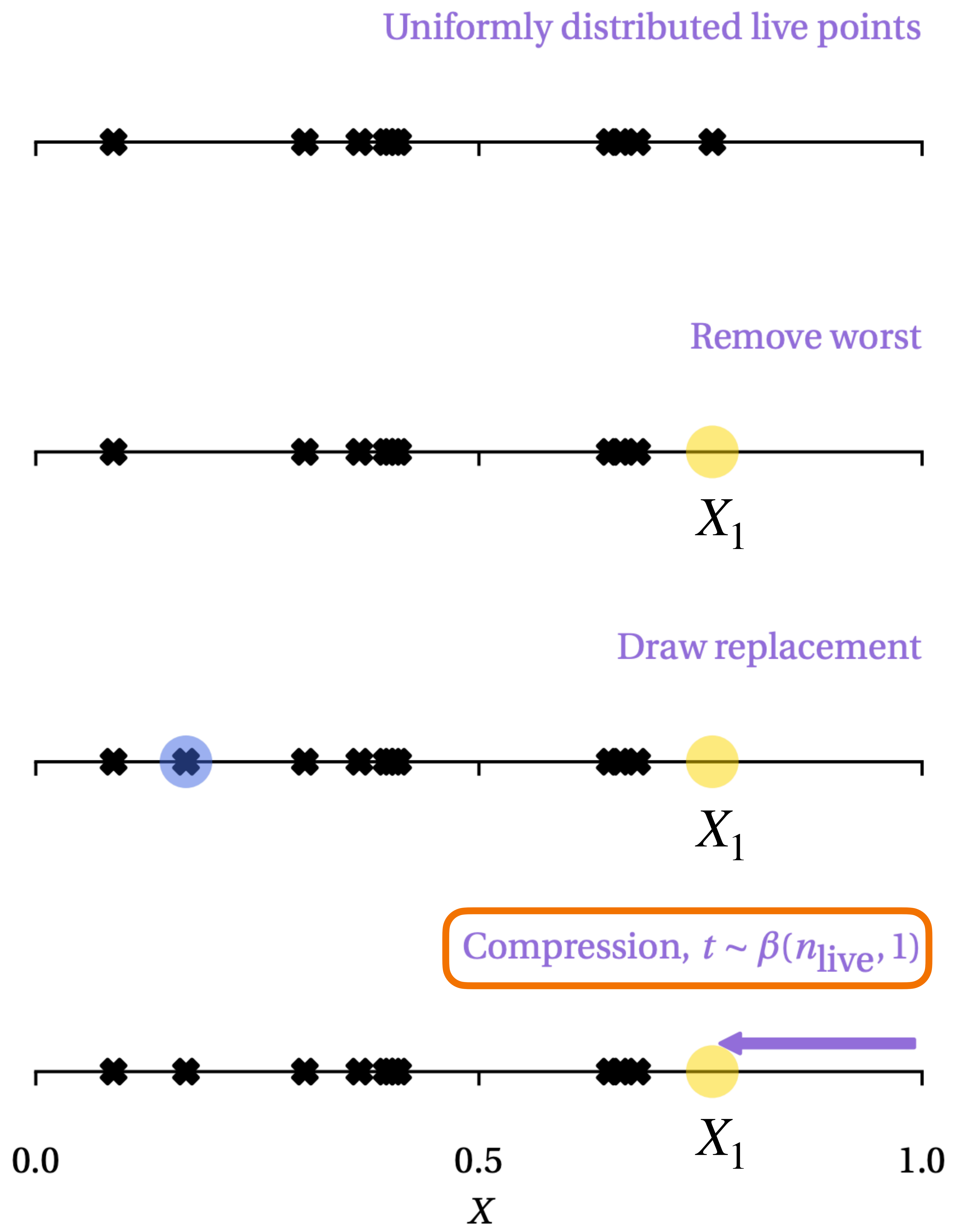
1. Sample a set of initial n_{live} "**live points**" $\{\Theta_1, \ldots, \Theta_{n_{\text{live}}}\}$ from the entire prior distribution $\pi(\Theta)$ and **sort them** by their likelihood values

2. Remove the point with the lowest likelihood λ_1

3. Replace the "dead point" by a new sample with higher likelihood drawn from $\pi_{\lambda_1}(\Theta)$

Repeat n_{iter} times

until a stopping condition is reached



Ashton et al., (2022)

SCHEMATIC OF THE NS ALGORITHM

1. Choose an estimate of the **compression factor**, e.g., $t = e^{-1/n_{\text{live}}}$
2. Initialise volume, $X_0 = 1$ and evidence, $Z = 0$
3. Sample a set of initial n_{live} "**live points**" from the entire prior distribution $\pi(\Theta)$

   REPEAT
    1. Let λ_{min} be the minimum $\tilde{L}$ of the live points
    2. Replace live point associated to λ_{min} by one drawn from the constrained prior $\pi_{\lambda_{\text{min}}}(\Theta)$
    3. Increment the estimate of the evidence, $Z = Z + \lambda_{\text{min}}\Delta X$, with e.g., $\Delta X = (1 - t)X$
    4. Contract volume, $X = tX$

   UNTIL stopping condition is satisfied
4. Add estimate of remaining evidence, e.g., $Z = Z + \bar{L}X$, where $\bar{L}$ is the average likelihood among the live points
5. **Return** the estimate of the integral Z

NS ALGORITHM

i) Divide the unit prior volume into a monotonic decreasing sequence of prior volumes X_i
ii) Sort them by likelihood

$$\tilde{L}_i = \tilde{L}(X_i) = \lambda_i$$

$$\tilde{L}_{n_{iter}} > \cdots > \tilde{L}_3 > \tilde{L}_2 > \tilde{L}_1 > 0$$

$$0 < X_{n_{iter}} < \cdots < X_3 < X_2 < X_1 < X_0 = 1$$

$$\Theta_{n_{iter}} \quad \cdots \quad \Theta_3 \quad \Theta_2 \quad \Theta_1$$

"Nested" $\tilde{L}$ contours

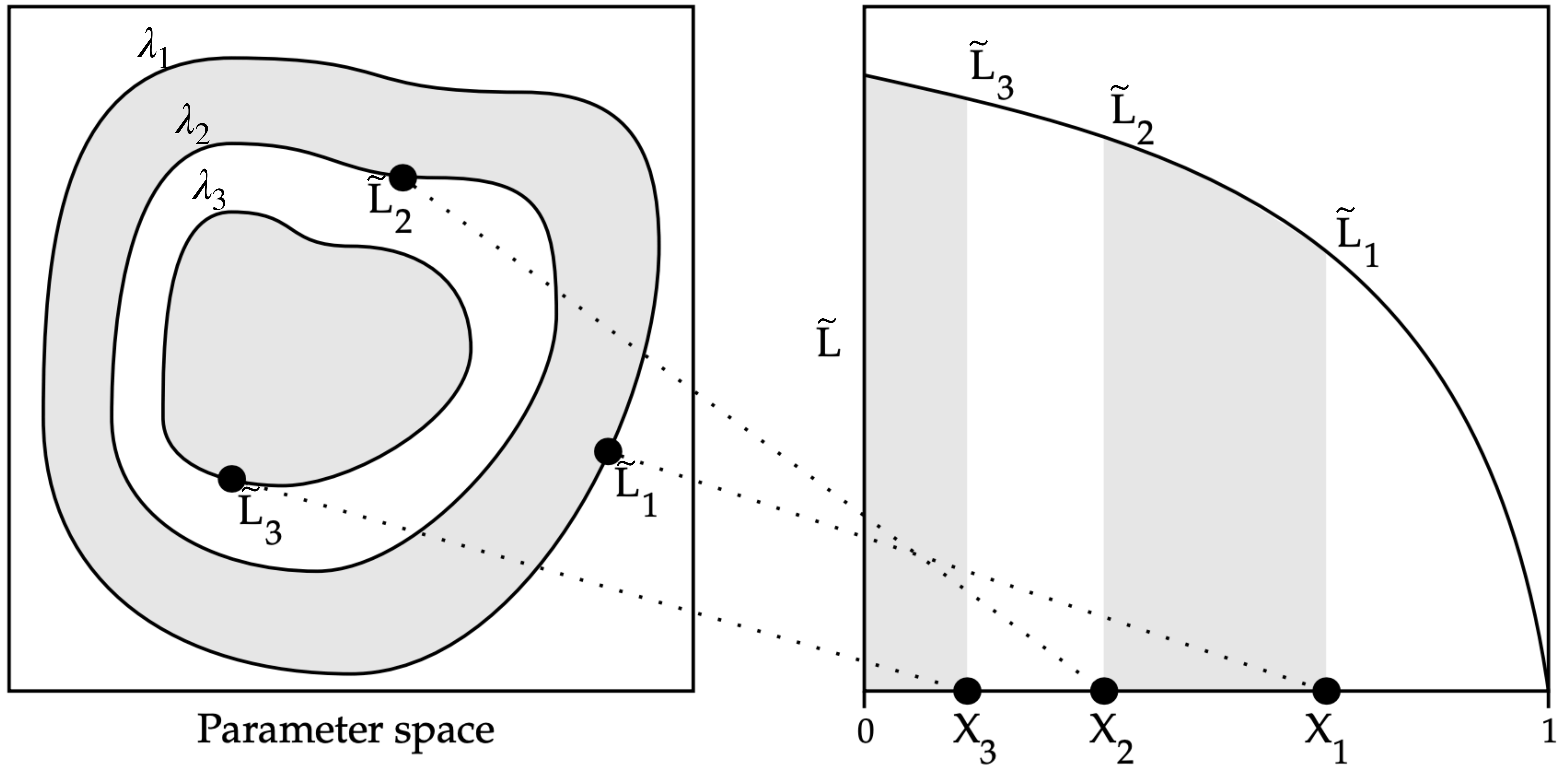


Figure 3: Nested likelihood contours are sorted to enclosed prior mass X.

Skilling (2006)

NS ALGORITHM

i) Divide the unit prior volume into a monotonic decreasing sequence of prior volumes X_i
ii) Sort them by likelihood

$$\tilde{L}_i = \tilde{L}(X_i) = \lambda_i$$

$$\tilde{L}_{n_{iter}} > \cdots > \tilde{L}_3 > \tilde{L}_2 > \tilde{L}_1 > 0$$

$$0 < X_{n_{iter}} < \cdots < X_3 < X_2 < X_1 < X_0 = 1$$

$$\Theta_{n_{iter}} \quad \cdots \quad \Theta_3 \quad \Theta_2 \quad \Theta_1$$

$$\sim \pi(\Theta)$$

"Nested" $\tilde{L}$ contours

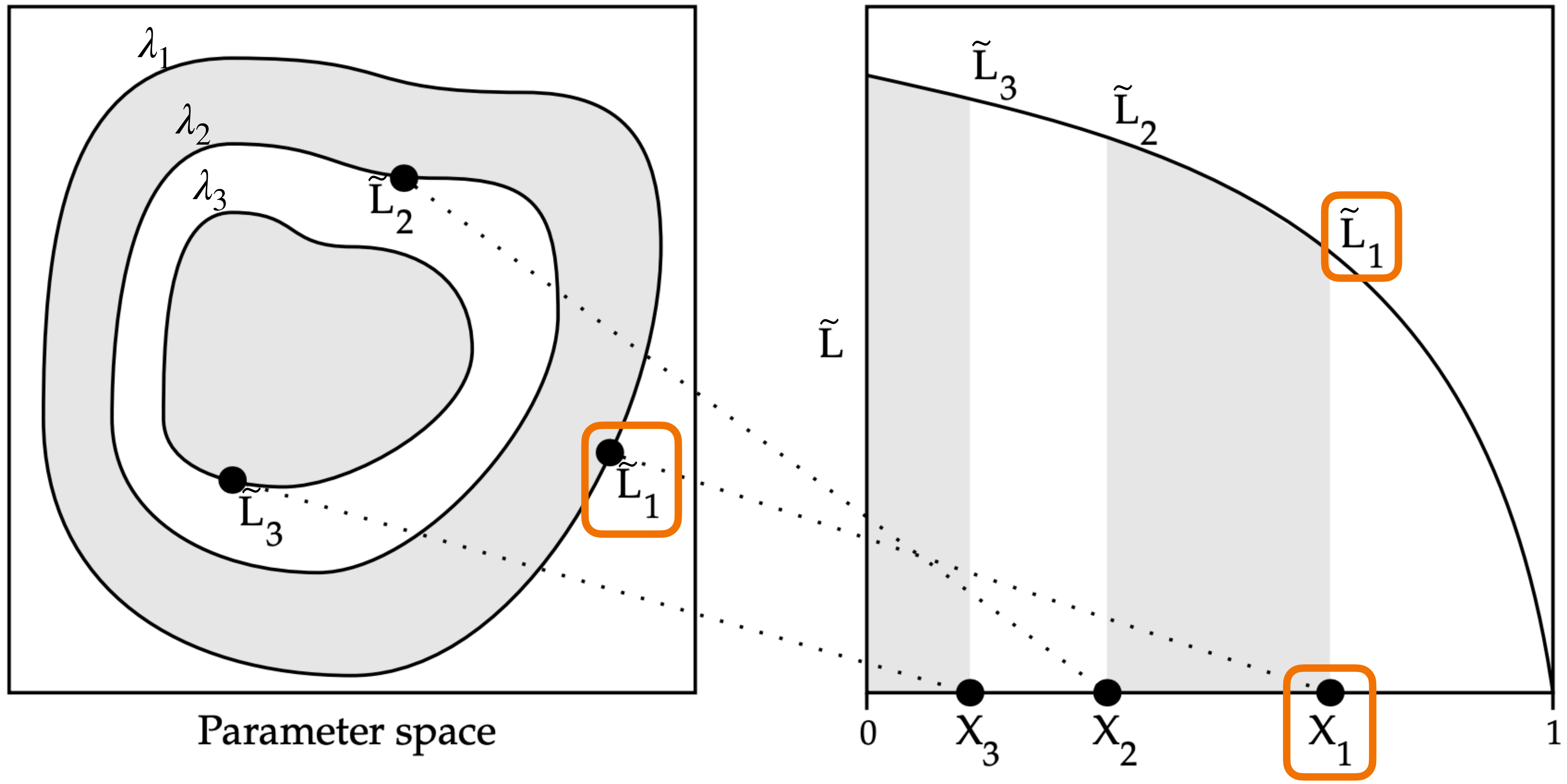


Figure 3: Nested likelihood contours are sorted to enclosed prior mass X.

Skilling (2006)

NS ALGORITHM

i) Divide the unit prior volume into a monotonic decreasing sequence of prior volumes X_i
ii) Sort them by likelihood

$$\tilde{L}_i = \tilde{L}(X_i) = \lambda_i$$

$$\tilde{L}_{n_{iter}} > \cdots > \tilde{L}_3 > \tilde{L}_2 > \tilde{L}_1 > 0$$

$$0 < X_{n_{iter}} < \cdots < X_3 < X_2 < X_1 < X_0 = 1$$

$$\Theta_{n_{iter}} \quad \cdots \quad \Theta_3 \quad \Theta_2 \quad \Theta_1$$

$$\sim \pi_{\lambda_1}(\Theta)$$

"Nested" $\tilde{L}$ contours

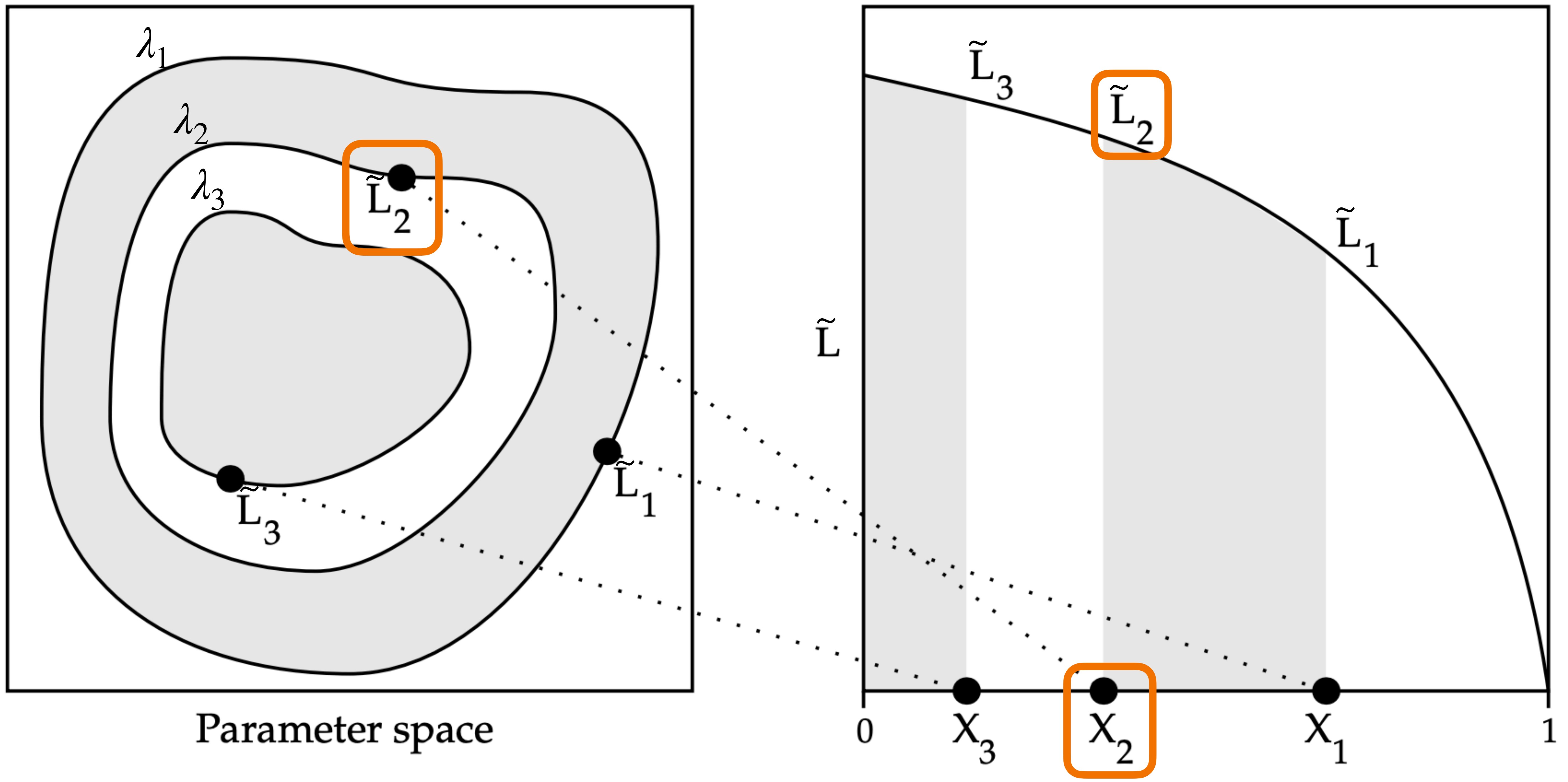


Figure 3: Nested likelihood contours are sorted to enclosed prior mass X.

Skilling (2006)

NS ALGORITHM

i) Divide the unit prior volume into a monotonic decreasing sequence of prior volumes X_i
ii) Sort them by likelihood

$$\tilde{L}_i = \tilde{L}(X_i) = \lambda_i$$

$$\tilde{L}_{n_{iter}} > \cdots > \tilde{L}_3 > \tilde{L}_2 > \tilde{L}_1 > 0$$

$$0 < X_{n_{iter}} < \cdots < X_3 < X_2 < X_1 < X_0 = 1$$

$$\Theta_{n_{iter}} \quad \cdots \quad \Theta_3 \quad \Theta_2 \quad \Theta_1$$

$$\sim \pi_{\lambda_2}(\Theta)$$

"Nested" $\tilde{L}$ contours

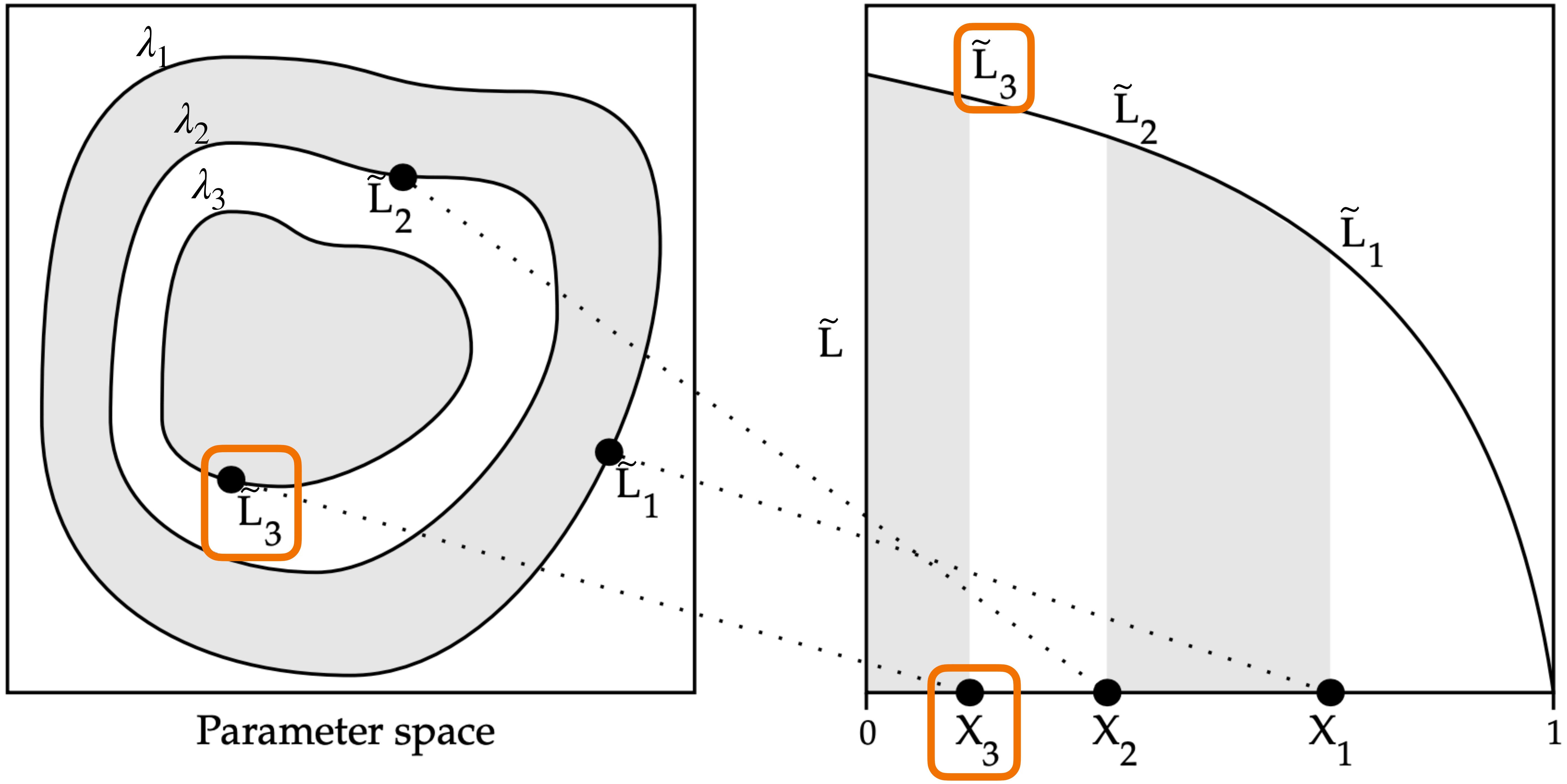


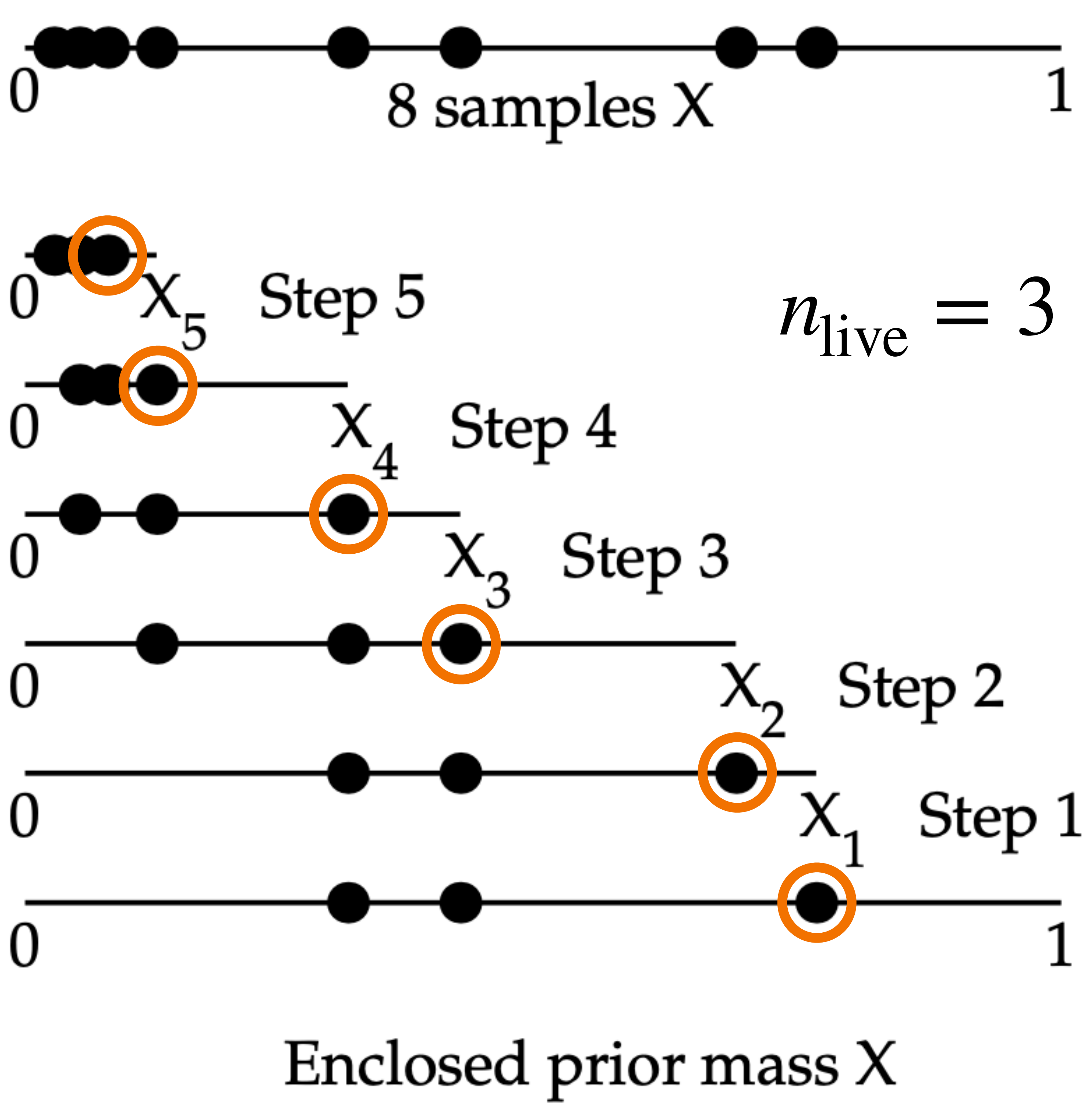
Figure 3: Nested likelihood contours are sorted to enclosed prior mass X.

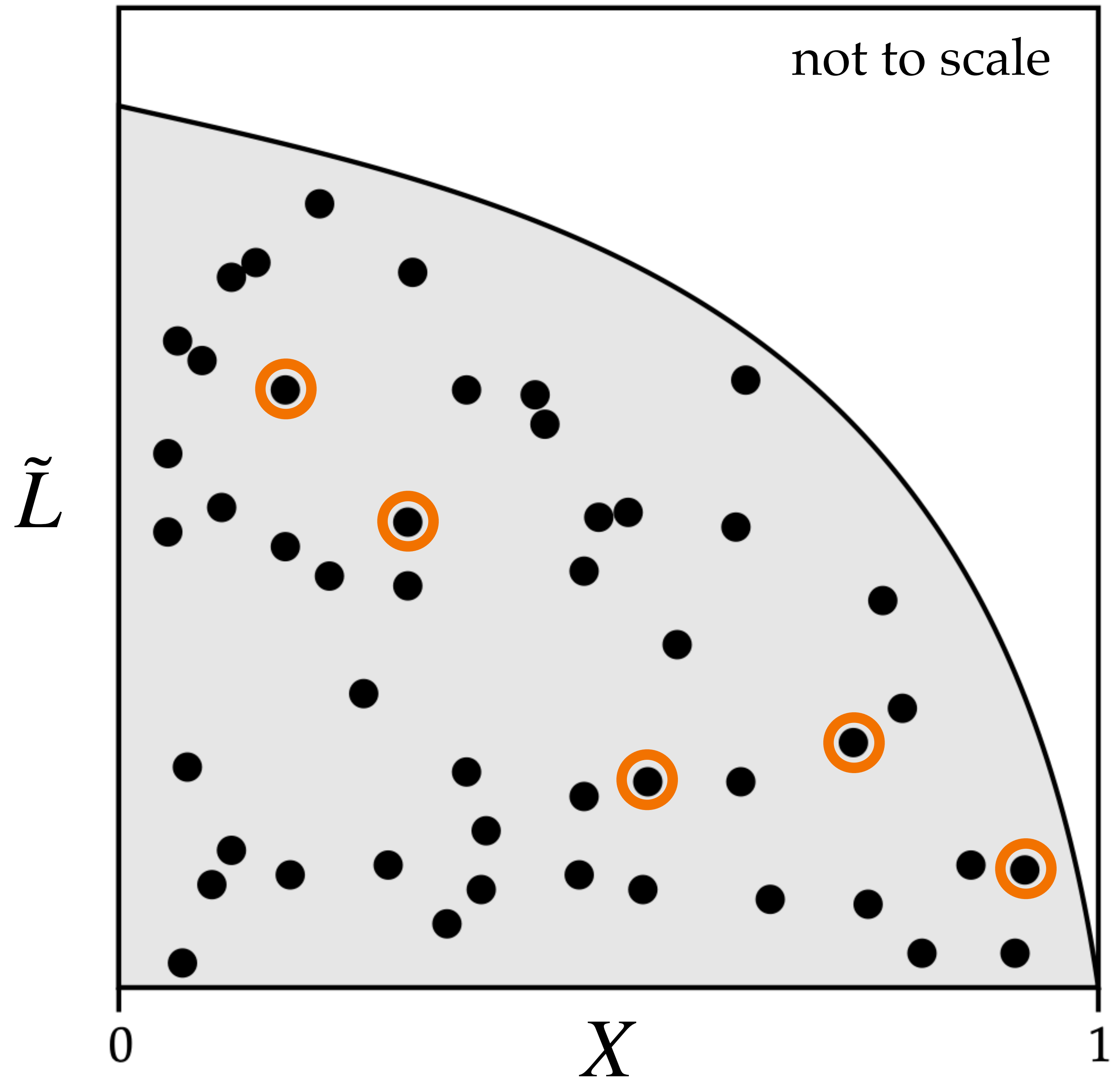
Skilling (2006)

POSTERIOR SAMPLES "FOR FREE"

"Recycle" full sequence of discarded, low-likelihood live points + final live points, to which an **importance weight** is assigned:

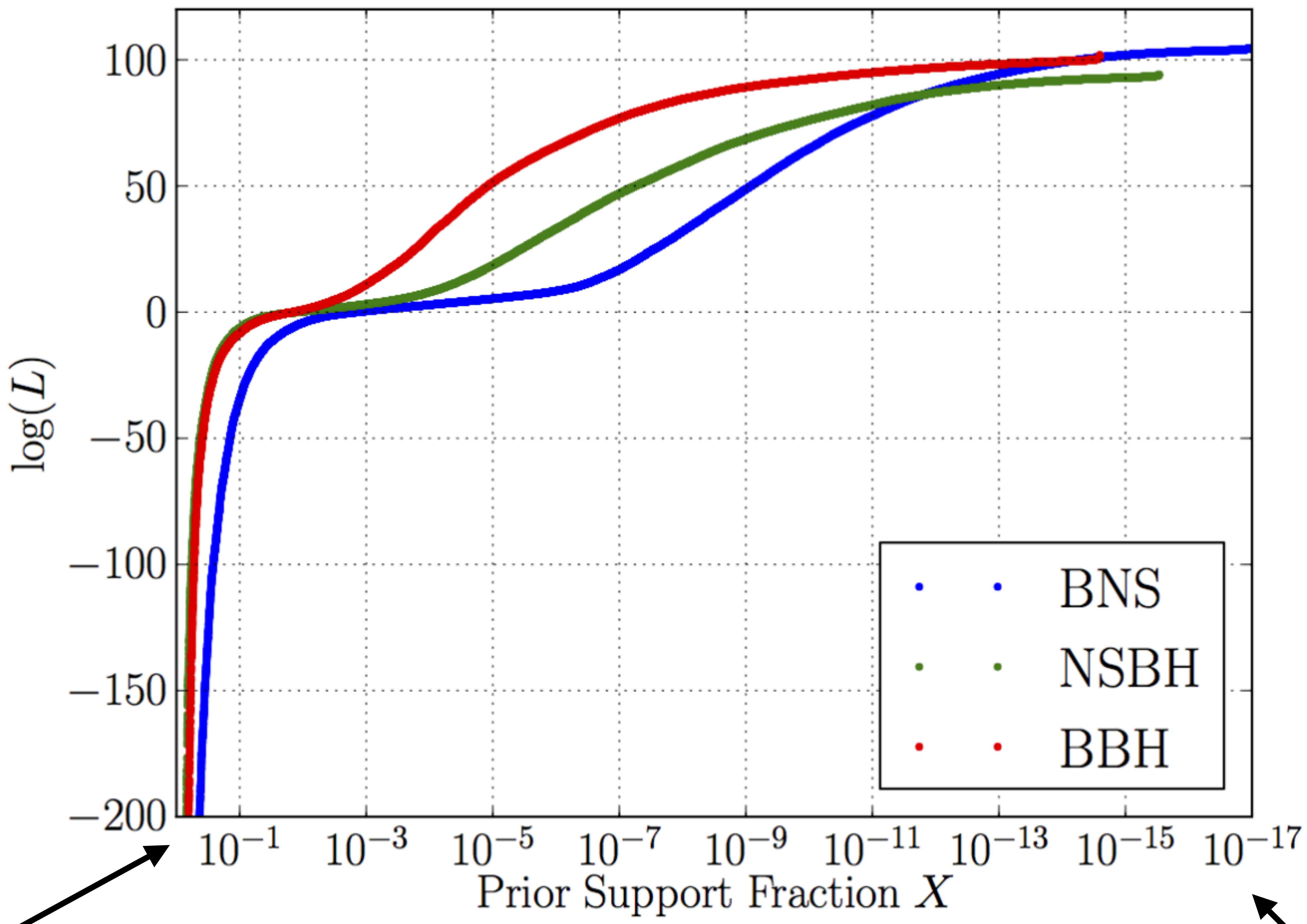
$$\left\{ \Theta_i, \quad \hat{p}(\Theta_i) = \frac{\tilde{L}_i (X_{i-1} - X_{i+1})/2}{Z} \right\}$$





Skilling (2006)

EXAMPLE: CBCs



Veitch et al., *PRD* (2015) [LALInferenceNest]

sampling
entire prior

sampling tiny restricted
part of prior

NS proceeds from L to R

OVERVIEW

- Nested sampling (**NS**) in a nutshell

- **Main challenges and limitations**

- Implementations & distributions: what's out there

#1: NS UNCERTAINTIES

- **Statistical** uncertainties (due to unknown X_i): $\sigma[\log Z] \sim \frac{1}{\sqrt{n_{\text{live}}}}$

- **Sampling** uncertainties (# samples, discrete point estimates for contours, particle path dependencies)

Provided NS is appropriately configured, the statistical uncertainty usually dominates

#2: STOPPING CONDITIONS

$$Z \simeq \sum_{i=1}^{n_{iter}} \frac{\tilde{L}_i}{2} (X_{i-1} - X_{i+1})$$

We want the truncation error to be small

E.g. use an estimate of the remaining evidence $\Delta Z / Z <$ tol:

Check whether the evidence estimate would change by more than a factor of ~0.1 if all the remaining prior support were at $\tilde{L}_{\max}$

$$\tilde{L}_{\max,i} X_i / Z_i > e^{0.1}$$

NB: if the summation is terminated too early, we could miss a spike of enormous likelihood lurking inward.

#3: HOW TO CHOOSE n_{live}?

- Trade-off between run-time and uncertainty
    - n_{live} controls the rate of compression as $\Delta \log X \simeq 1/n_{\text{live}}$ per iteration
    - Run-time scales as $\mathcal{O}(n_{\text{live}})$
    - However $\Delta \log Z \simeq \mathcal{O}(1/\sqrt{n_{\text{live}}})$

- n_{live} should exceed the dimensionality of the parameter space

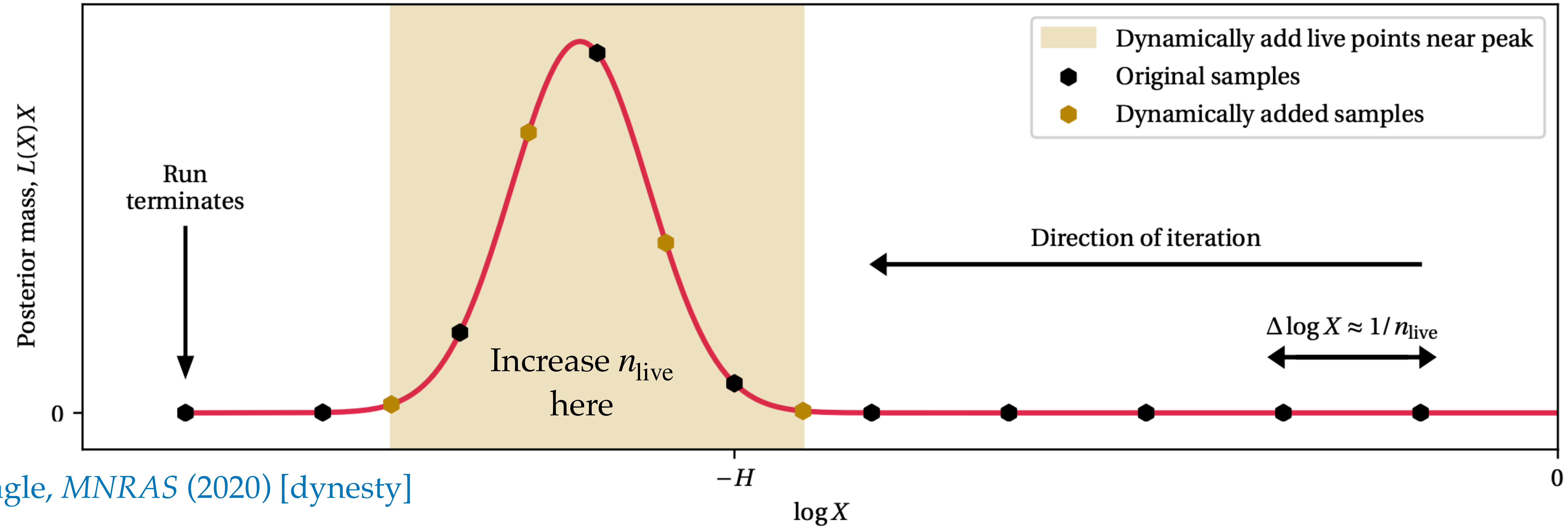
- **NB** In multi-modal problems, choose n_{live} large enough that at any time $\pi_\lambda(\Theta)$ splits into disjoint modes (at least one live point inside the footprint of each mode)

#4: STATIC vs DYNAMIC?

Fixed n_{live} during the run (**static** NS) **vs** **Varying** n_{live} during the run (**dynamic** NS)

- n_{live} can be dynamically adjusted to **maximise** calculation **accuracy** and improve computational **efficiency**
- The user can decide if to have less uncertainty on Z or on the posterior

- Variant of dynamic: diffusive NS (n_{live} can change at a given λ)



Speagle, *MNRAS* (2020) [dynesty]

a | **Schematic representation of an NS run.** The curve $L(X)X$ shows the relative posterior mass, the bulk of which lies in a tiny fraction e^{-H} of the volume. Most of the original samples lie in regions with negligible posterior mass. In dynamic NS, we add samples near the peak.

Information a.k.a. **Kullback-Liebler divergence**

$$H = \int p(\Theta|D) \log\left(\frac{p(\Theta|D)}{\pi(\Theta)}\right) d\Theta \simeq \sum_i \frac{\tilde{L}_i(X_{i+1} - X_{i-1})}{Z} \log\left[\frac{\tilde{L}_i}{Z}\right] \simeq \log\left(\frac{\text{volume of prior}}{\text{volume of posterior}}\right)$$

#5: HOW TO DRAW FROM THE CONSTRAINED PRIOR?

$$\pi_\lambda(\Theta) \propto \begin{cases} \pi(\Theta) & \text{if } \mathscr{L}(\Theta) > \lambda \\ 0 & \text{otherwise} \end{cases}$$

- **Very difficult**, especially in multi-modal problems

- **NS is self-tuning**: use the live points to build proposal structures and apply clustering algorithms

- Two main classes of sampling: **region** sampling, **step** sampling

- **NB** In multi-modal problems, if no live points lie inside a mode, that region of $\pi_\lambda(\Theta)$ almost certainly won't be sampled

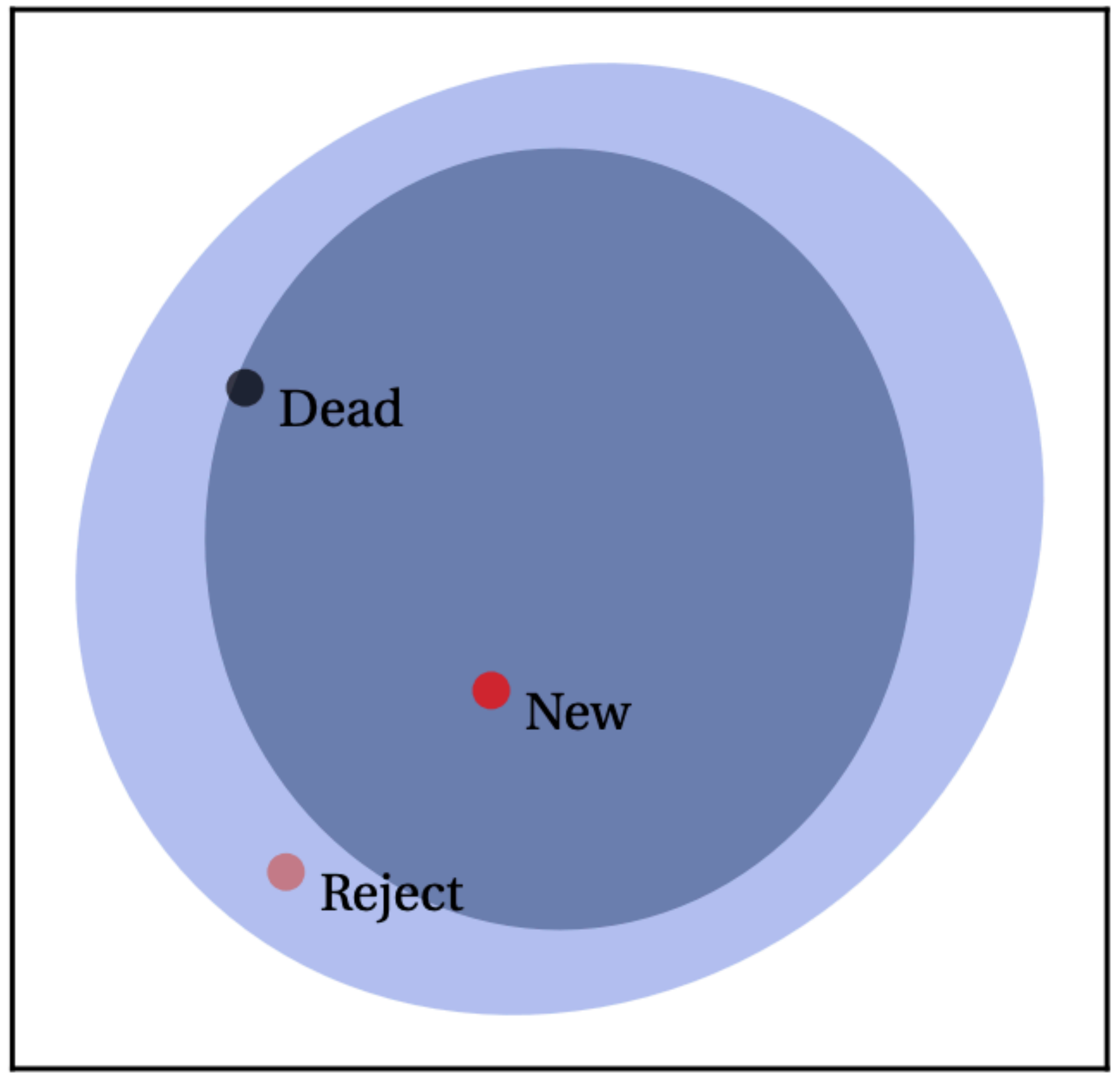
It's easier to work in the hypercube, a parametrisation in which the prior is uniform over a unit hypercube

#5: HOW TO DRAW FROM THE CONSTRAINED PRIOR?

We must sample from the true iso-likelihood contour (grey ellipse)

region samplers step samplers

Region sampling



Ellipsoid sampling

Ashton et al., (2022)

- Attempt to bound the existing live points (blue ellipse)
- Draw a new sample from within that bound
- Some proposals may be rejected

Major limitations:

- accuracy of bounds strongly depends on n_{live}
- accuracy and efficiency scale exponentially with D

Efficient and practical only for moderate-to-low dimensionalities (D≤20)

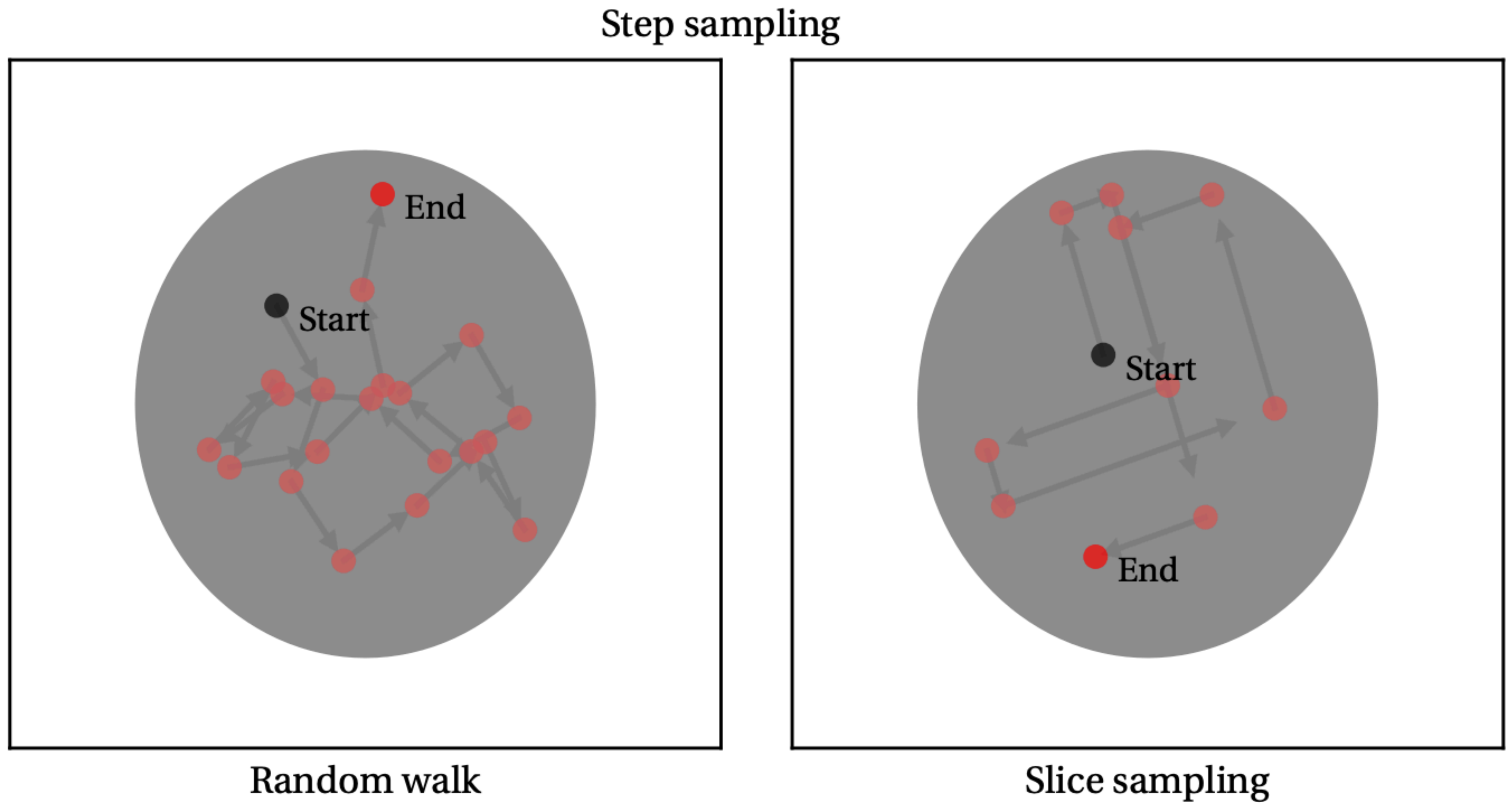
#5: HOW TO DRAW FROM THE CONSTRAINED PRIOR?

We must sample from the true iso-likelihood contour (grey ellipse)

region samplers **step samplers**

- Computational cost:
  polynomial scaling with D

- Select a live point
- Evolve that point within inside the contour to obtain an independent draw from $\pi_\lambda(\Theta)$, e.g.
    - random-walk Metropolis
    - slice sampling



Ashton et al., (2022)

Efficient when
$\mathscr{L}(\theta_1, \theta_2, \theta_3) = \text{slow}(\theta_1) \times \text{fast}(\theta_2, \theta_3)$

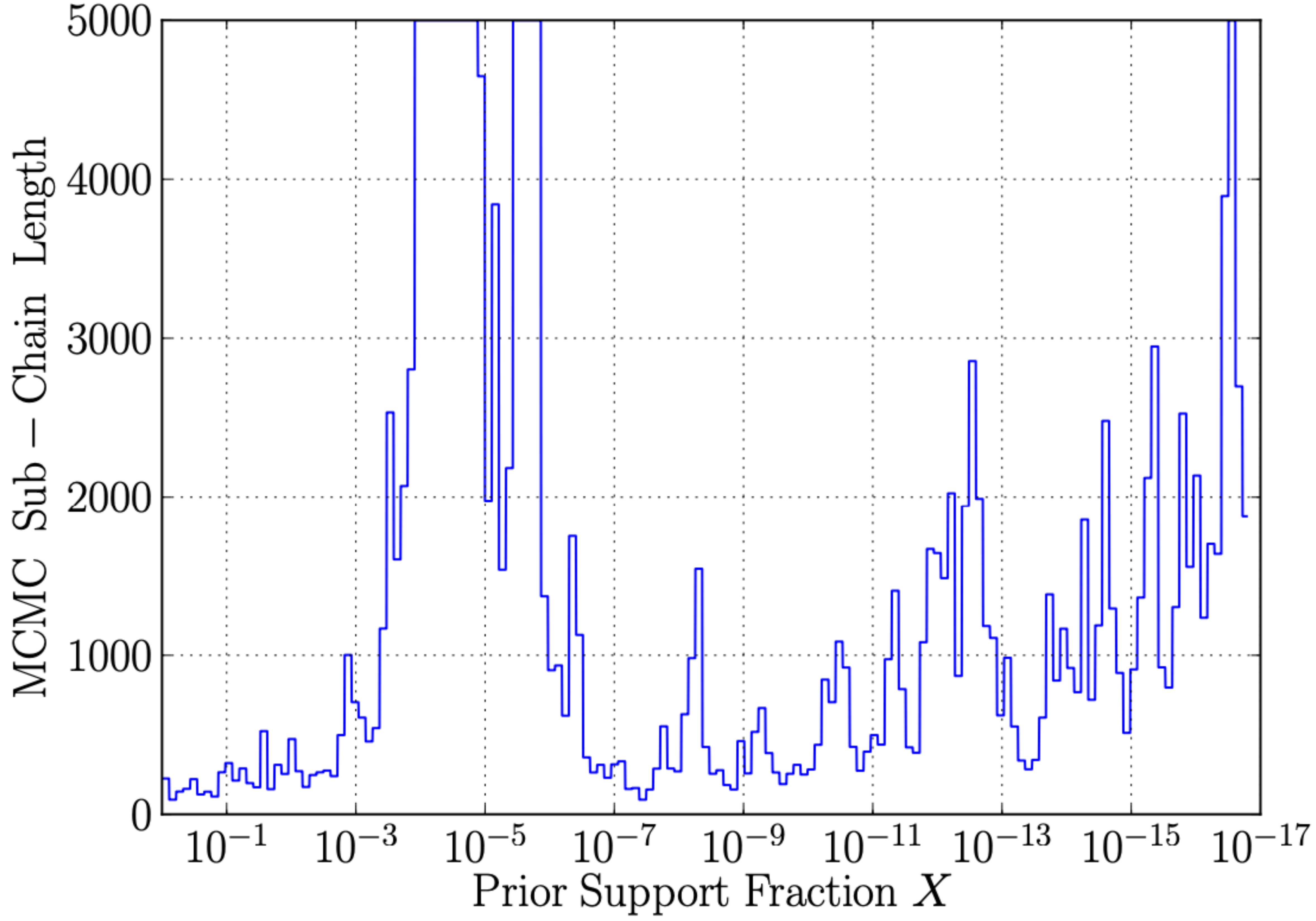
#5: HOW TO DRAW FROM THE CONSTRAINED PRIOR?

We must sample from the true iso-likelihood contour (grey ellipse)

region samplers **step samplers**

- Computational cost: polynomial scaling with D

- Select a live point
- Evolve that point within inside the contour to obtain an independent draw from $\pi_\lambda(\Theta)$, e.g.
    - random-walk Metropolis
    - slice sampling



Veitch et al., *PRD* (2015) [LALInferenceNest]

NB $\sigma[\log Z]$ depends also on **length** of MCMC subchains

Variable MCMC chain length and **GW-specific jump proposals** can be used to exploit correlations between parameters and efficiently sample between isolated modes

#6: PARALLELISATION?

Although NS is a sequential method, parallelisation can be used to increase #posterior samples

1 NS with M "live points" = M NS with 1 live point

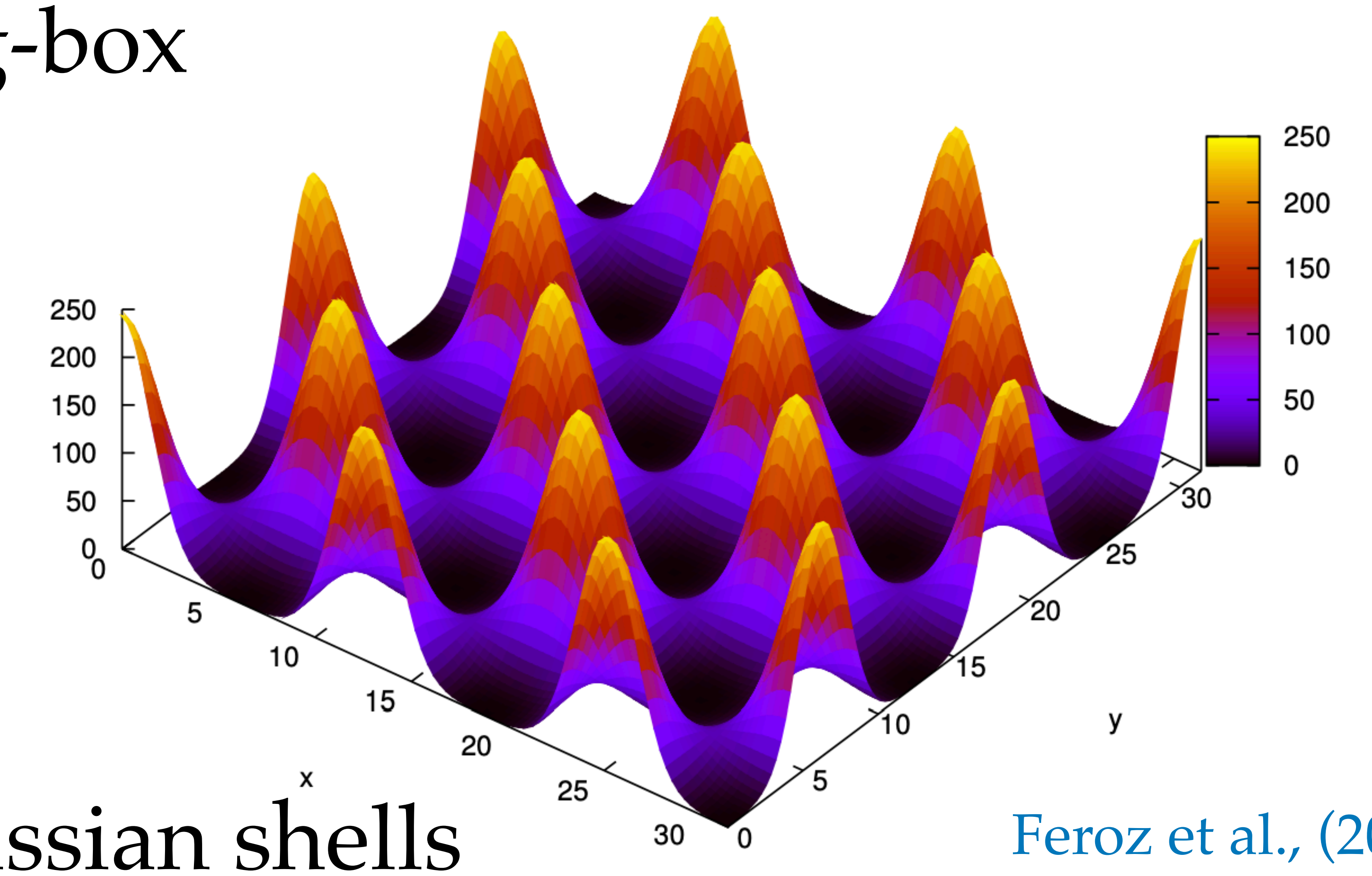
- Run independent NSs on different CPU cores, then combine the results weighted by their respective evidence
- More chains producing samples
- Each chain is weighted by its respective evidence

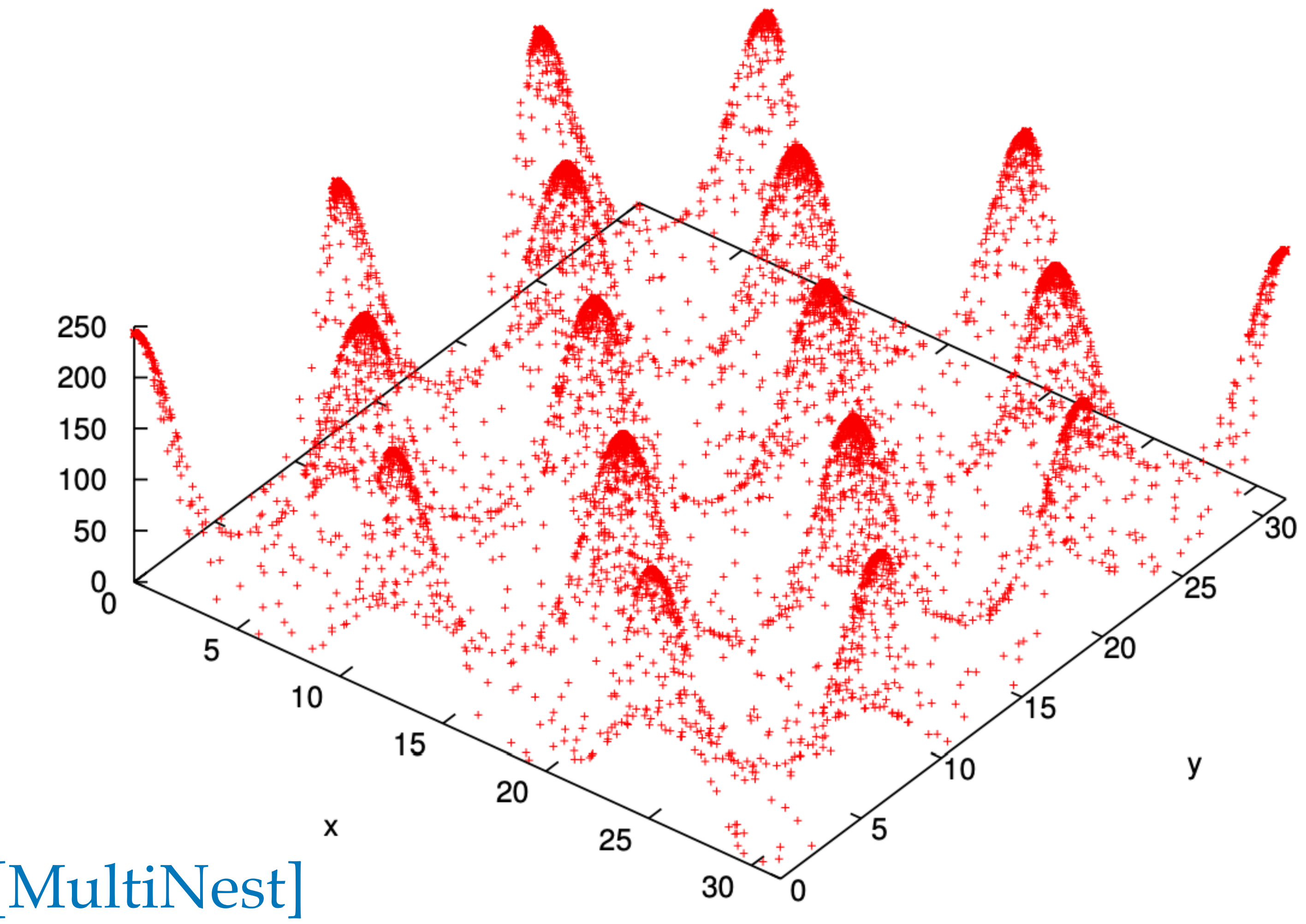
Also the number of replacements per iteration may be varied

HOW TO CHECK RESULTS?

★Compute the evidence integral for problems with known analytic solutions, e.g.:

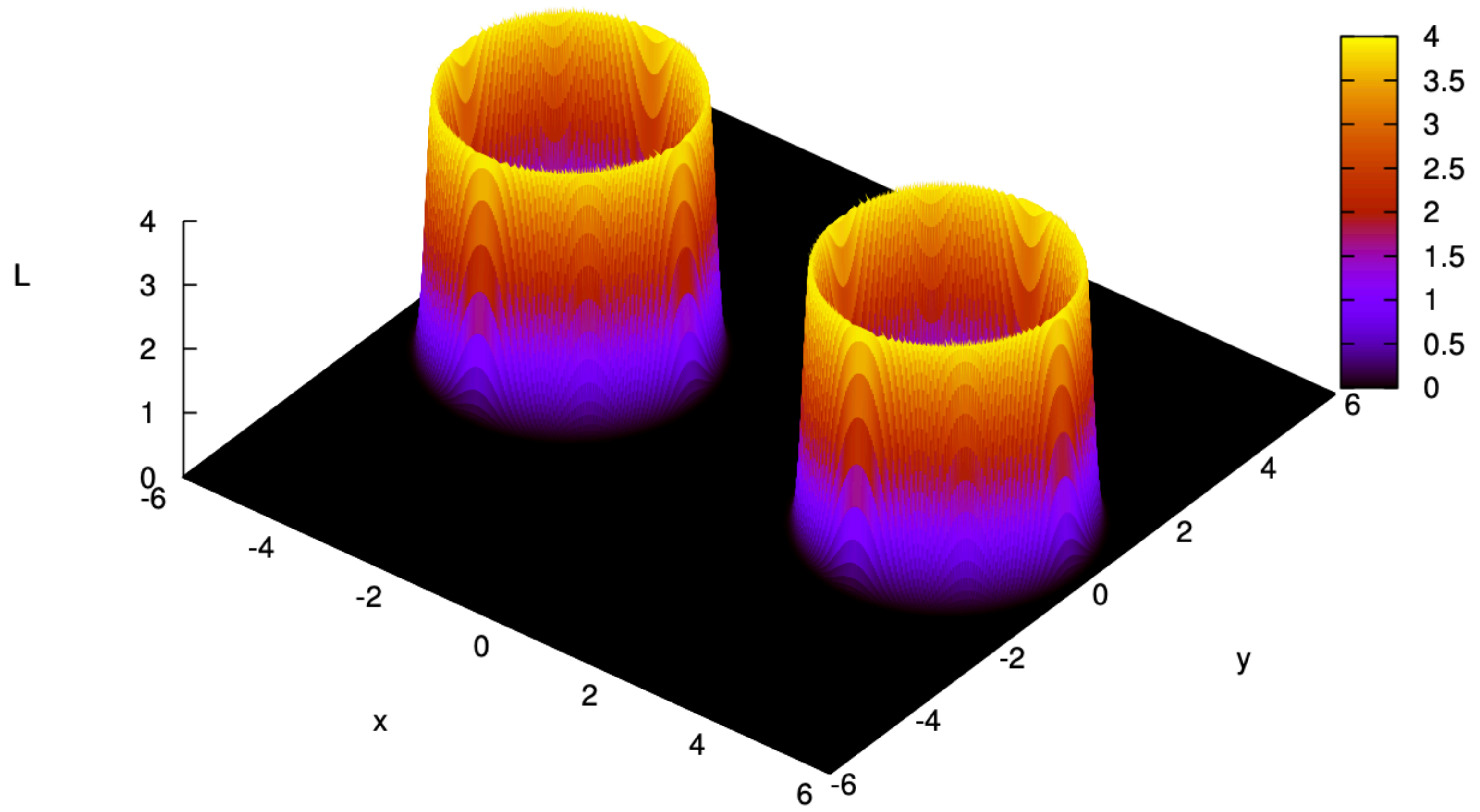
- multi-dimensional Gaussian likelihood (D=200)
- Egg-box

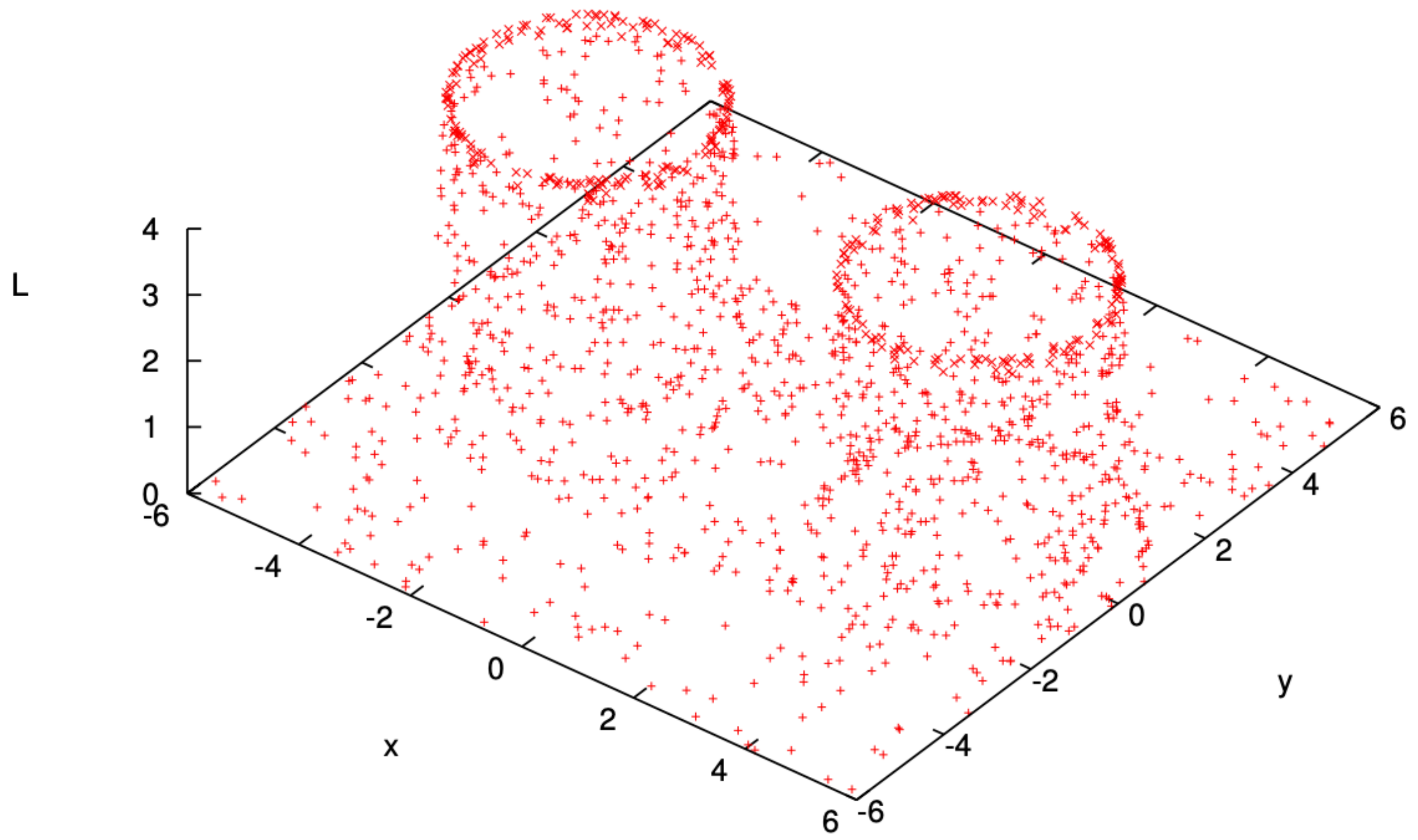




- Gaussian shells

Feroz et al., (2013) [MultiNest]





If modes are missed, increase n_{live}

★Check that live points are independently drawn from $\pi_\lambda(\Theta)$

★Compare posterior samples between different NS implementations and/or MCMC

OVERVIEW

- Nested sampling (**NS**) in a nutshell
- Main challenges and limitations
- **Implementations & distributions: what's out there**

NS IMPLEMENTATIONS

Group	Code	Methods	Dynamic	Languages	Field	Pub. Year
region samplers	CosmoNest [60, 61]	ellipsoid	fixed	Fortran	Cosmology	2006
	MultiNest [48, 84]	multi-ellipsoid	fixed	Fortran, C/C++, Python	Cosmology	2008
	DIAMONDS [249]	multi-ellipsoid	fixed	C++	Astrophysics	2015
	nestle [250]	ellipsoid, multi-ellipsoid	fixed	Python	Astrophysics	2015
	nessai [90, 91]	normalising flow ellipsoid	fixed	Python	Gravitational waves	2021
step samplers	(dy)PolyChord [53, 65]	slice	dynamic	Fortran, C/C++, Python	Cosmology	2015
	LALInferenceNest [180]	random walk, ensemble, differential evolution	fixed	C	Gravitational waves	2015
	Nested_fit [104, 257, 258]	random walk	fixed	Fortran	Atomic physics	2016
	cpnest [259]	slice, differential evolution, Gauss, Hamiltonian, ensemble	fixed	Python	Gravitational waves	2017
	pymatnest [44]	random walk, Galilean, symplectic Hamiltonian	fixed	Python	Materials	2017
	NNest [261]	normalising flow random walk	fixed	Python	Cosmology	2019
	DNest5 [55]	user-defined, random walk	diffusive	C++	Astrophysics	2020
	BayesicFitting [263]	random walk, slice, Galilean, Gibbs	fixed	Python	Astronomy	2021
region / step samplers	dynesty [52]	ellipsoid, multi-ellipsoid, MLFriends & Gauss, slice, Hamiltonian	dynamic	Python	Astrophysics	2020
	UltraNest [92]	MLFriends + ellipsoid & Gauss, hit-and-run, slice	reactive	Python, Julia, R, C/C++, Fortran	Astrophysics	2020
	jaxns [266]	multi-ellipsoid & slice	fixed	jax	Astronomy	2021

Table 2 | **Comparison of NS codes.** The first two groups are region samplers and step samplers, respectively, whereas the third group offers both. Dynamic implementations allow the number of live points to be changed during a run. We show the language in which the NS code was written followed by any additional languages for which interfaces exist, and the field from which the code originated (though most are general purpose codes).

Ashton et al., (2022)

BENEFITS OF NS

Ashton et al., (2022)

- It simultaneously returns results for **parameter inference** and **model comparison**
- It is successful in **multi-modal** problems
- It is naturally **self-tuning**

DRAWBACKS AND CHALLENGES OF NS

Ashton et al., (2022)

- Draw independent samples from the **constrained prior**
    - Due to the point above, NS may **miss modes**
- **Inefficiently sample** from the constrained prior
- Hard with particularly **awkward** likelihood (e.g. with **plateaus**)

Thank you for listening

EXTRA SLIDES