Trans-dimensional sampling methods for LISA Data Analysis: Current status and future prospects

Nikolaos Karnesis
Aristotle University of Thessaloniki
karnesis@auth.gr

LIDa Workshop
32/10/2022

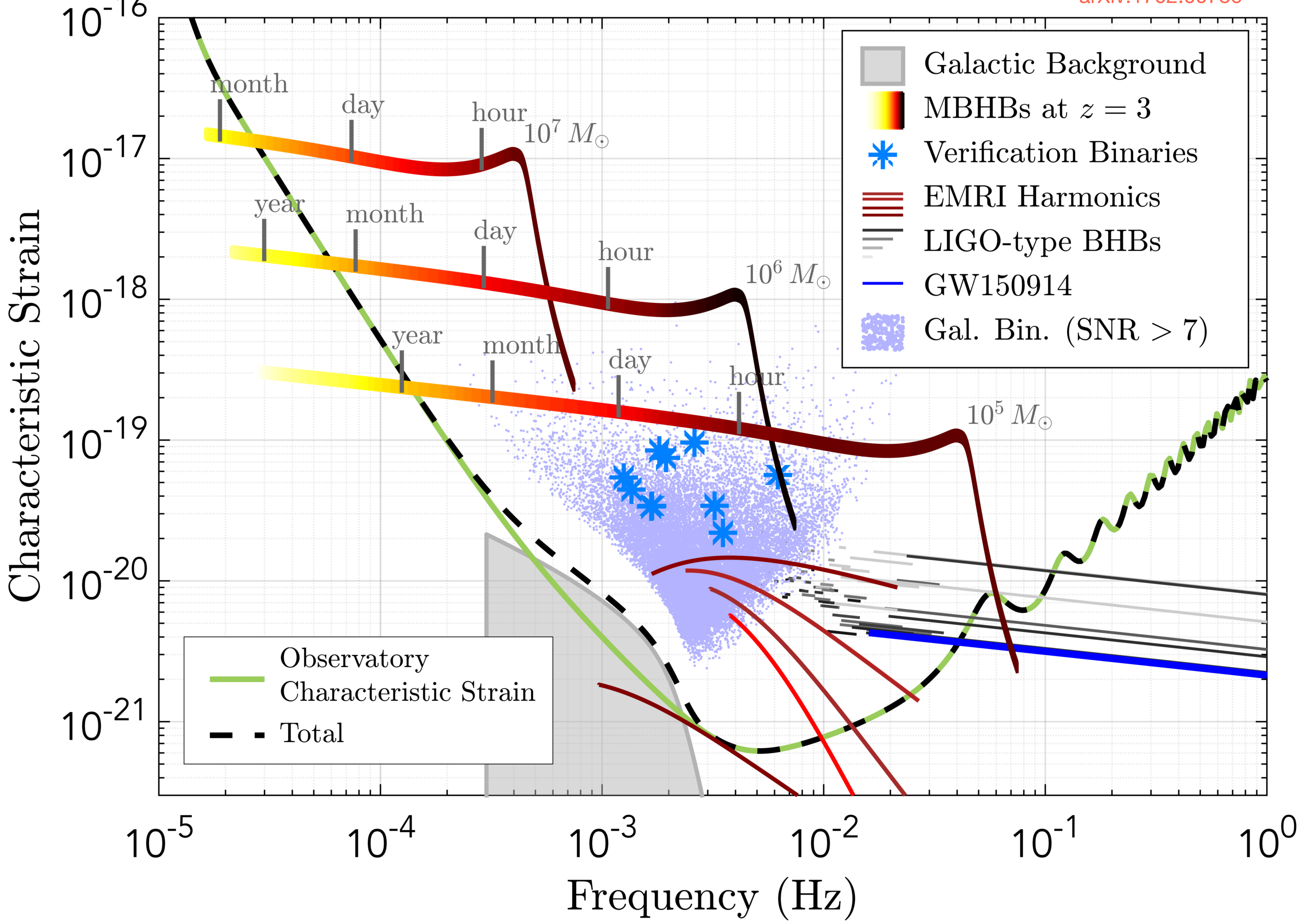
LISA CONSORTIUM

ESA | NASA
lisa

Outline

- Why do we choose to use sampling methods?
- Trans-dimensional sampling: What, why, how, limits?
- Future usage for LISA Data Analysis?

What do we expect for LISA?


Galactic Background
MBHBs at $z = 3$
Verification Binaries
EMRI Harmonics
LIGO-type BHBs
GW150914
Gal. Bin. (SNR > 7)
Observatory Characteristic Strain
Total
month
day
hour
$10^7\ M_\odot$
year
month
day
hour
$10^6\ M_\odot$
year
month
day
hour
$10^5\ M_\odot$
Characteristic Strain
10^{-16}
10^{-17}
10^{-18}
10^{-19}
10^{-20}
10^{-21}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
Frequency (Hz)

So, what is in the future?

Way too many events to be measured!

- LISA: ~10^6 DWD (~10^4 resolvable), (~10^1) SMBHB/year,
  ? EMRIs/year, ~10^6 SOBBHs (~10^1 resolvable)
  ? Stochastic GW backgrounds.

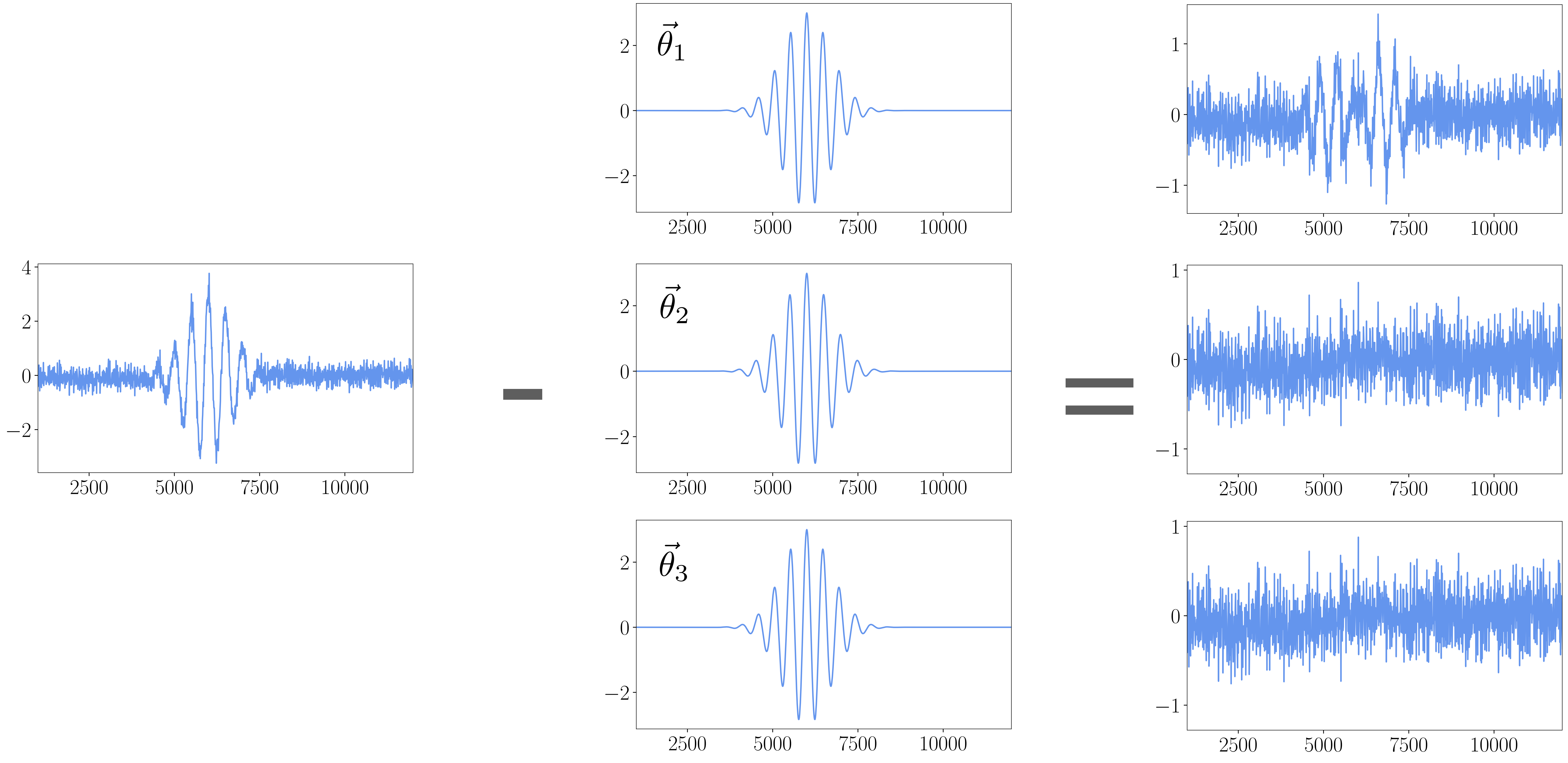
- In some cases, we'll have so many sources, that they will generate a stochastic GW signal above the detector noise!

arXiv:1702.00786
arXiv:2109.09882

Why sampling methods & Global fit for LISA DA?

Matched filtering

… this is what we normally do, first assume $y = h(\vec{\theta}) + n$



In practice, we define a likelihood function,

form a posterior, which we then need to investigate.

$$\pi(y|\vec{\theta}) = C \times e^{-\frac{1}{2}\left(y - h(\vec{\theta})\middle|y - h(\vec{\theta})\right)} = C \times e^{-\chi^2/2}$$

$$\pi(\vec{\theta}|y) \propto \pi(y|\vec{\theta})p(\vec{\theta})$$

$$(a|b) = 2\int_0^\infty \mathrm{d}f \left[\tilde{a}^*(f)\tilde{b}(f) + \tilde{a}(f)\tilde{b}^*(f)\right] / \tilde{S}_n(f)$$

Defining the parameter space

Way too many events to be measured!

- Focusing on LISA, we get to measure thousands of *overlapping* signals of different types.

$$N_{\vec{\theta}} = N_{\rm ucbs} + N_{\rm smbhb} + N_{\rm sobhbs} + N_{\rm emris} + N_{\rm stoch} + N_{\rm noise} \approx \mathcal{O}(10^5)$$

Defining the parameter space

Way too many events to be measured!

- Focusing on LISA, we get to measure thousands of *overlapping* signals of different types.

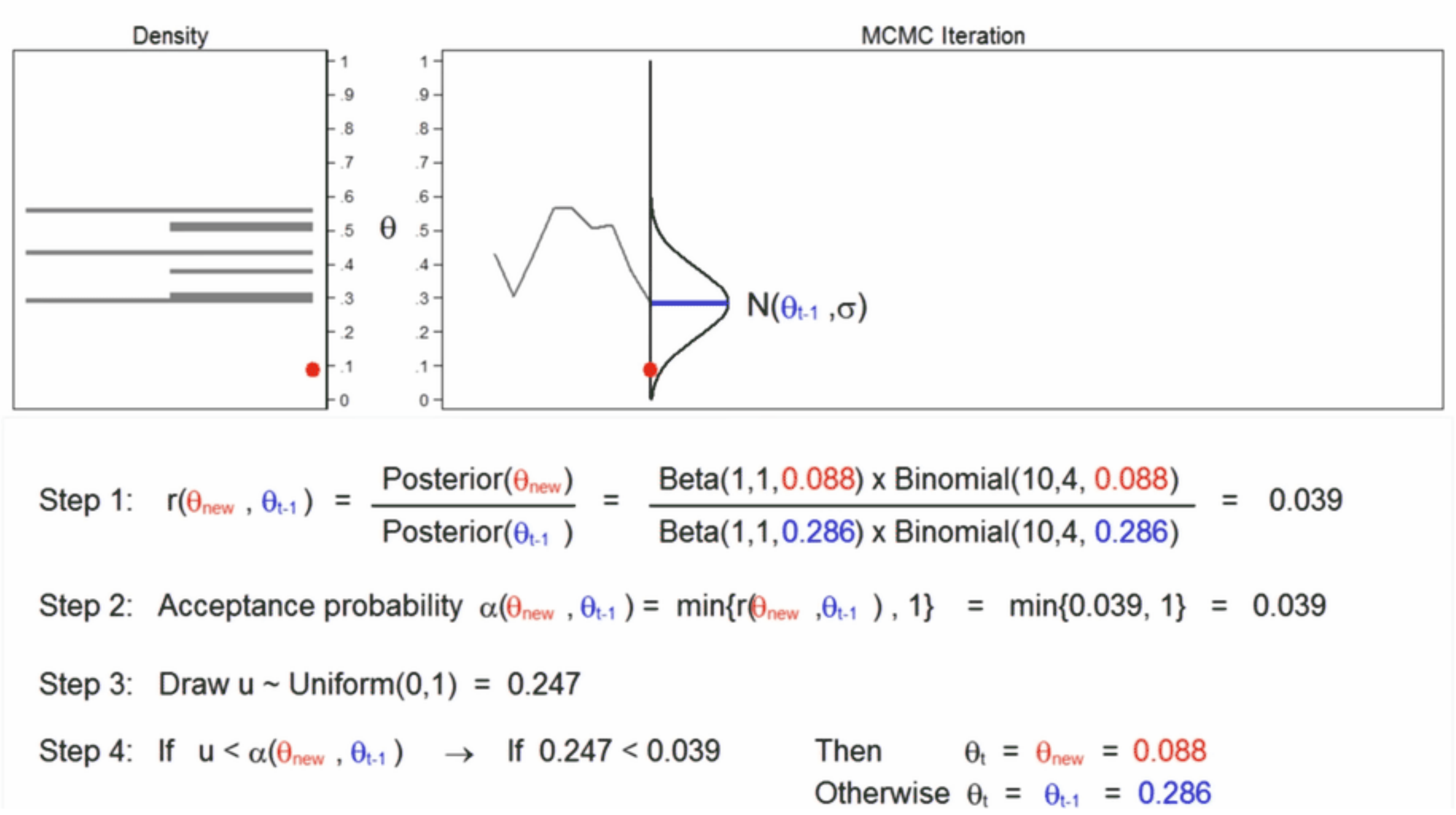
$$N_{\vec{\theta}} = N_{\mathrm{ucbs}} + N_{\mathrm{smbhb}} + N_{\mathrm{sobhbs}} + N_{\mathrm{emris}} + N_{\mathrm{stoch}} + N_{\mathrm{noise}} \approx \mathcal{O}(10^5)$$

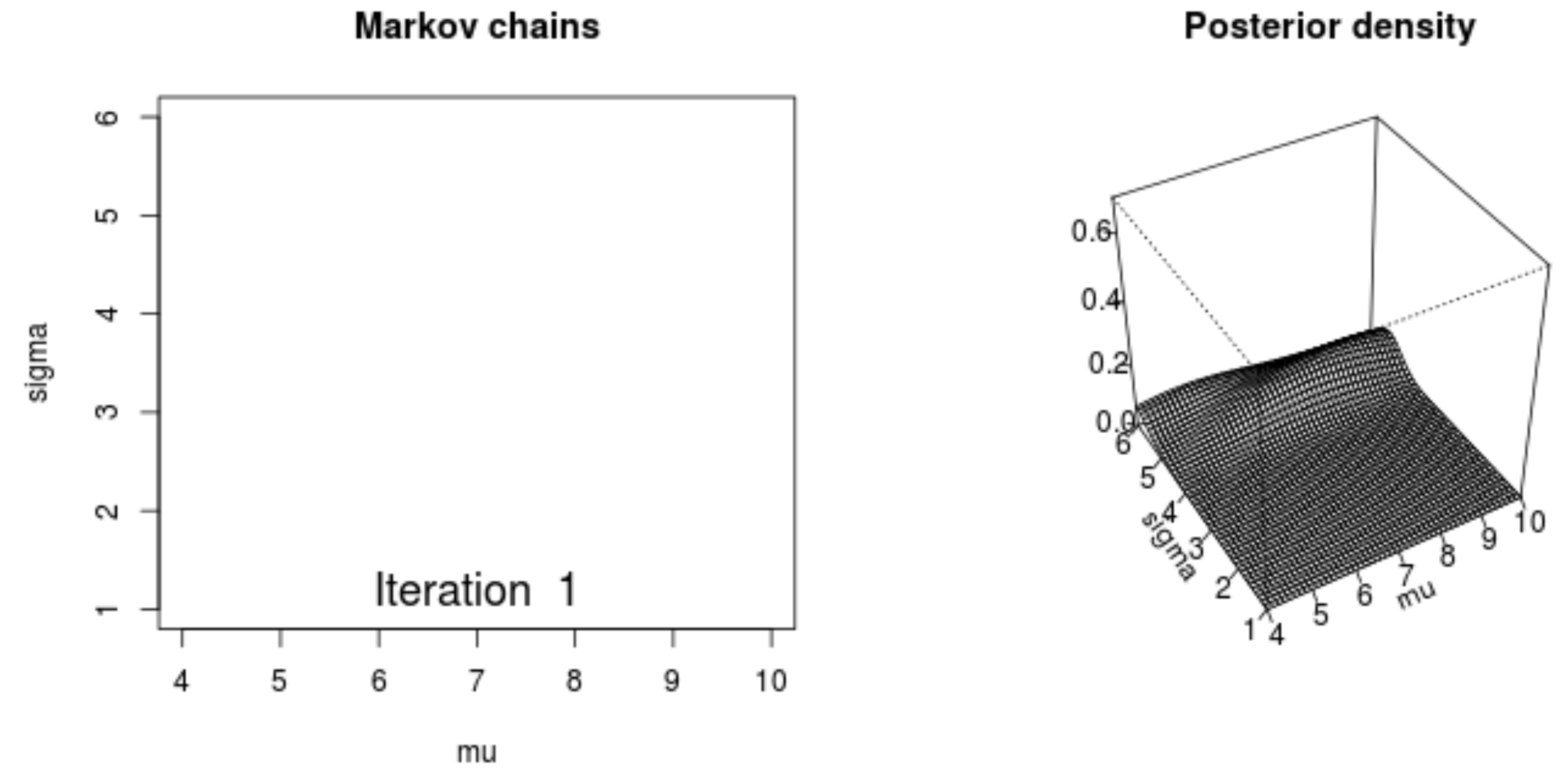
LISA Global Fit

- Computational reasons: sequential fits are inefficient
- Grid searches are almost impossible
- Correlations between sources become important for that many signals
- Imperfect source subtraction yields imperfect residuals
- Uncertainties propagation
- Not fixed dimensions!

Trans-dimensional MCMC

Start with fixed dimensionality





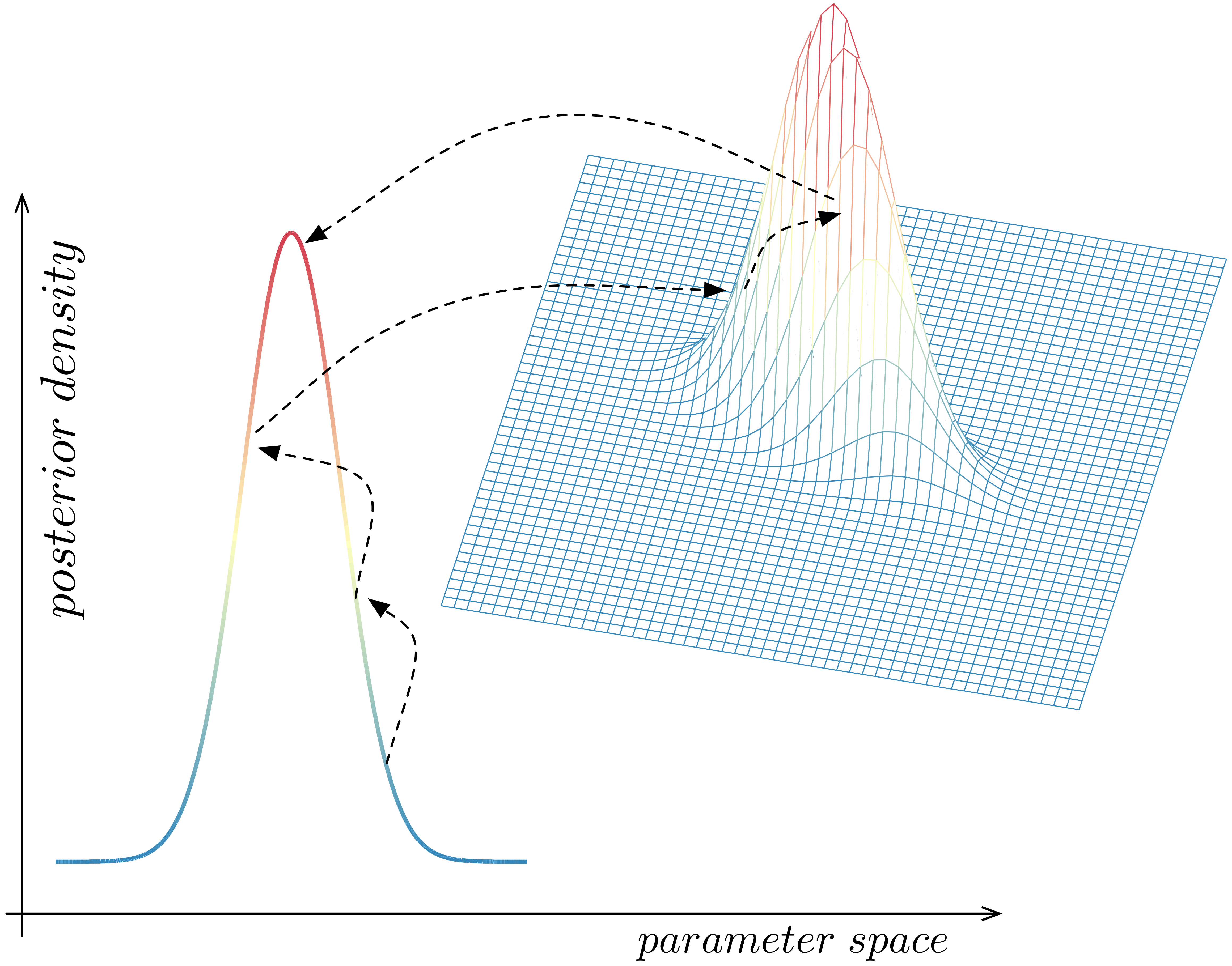
https://blog.stata.com/
https://blog.revolutionanalytics.com/

- Start from theta_0
- Propose a new point from proposal distribution q
- Accept, or reject with a probability

$$\alpha = \min\left[1, \frac{p(\vec{\theta}_1|y)q(\vec{\theta}_0, \vec{\theta}_1)}{p(\vec{\theta}_0|y)q(\vec{\theta}_1, \vec{\theta}_0)}\right]$$

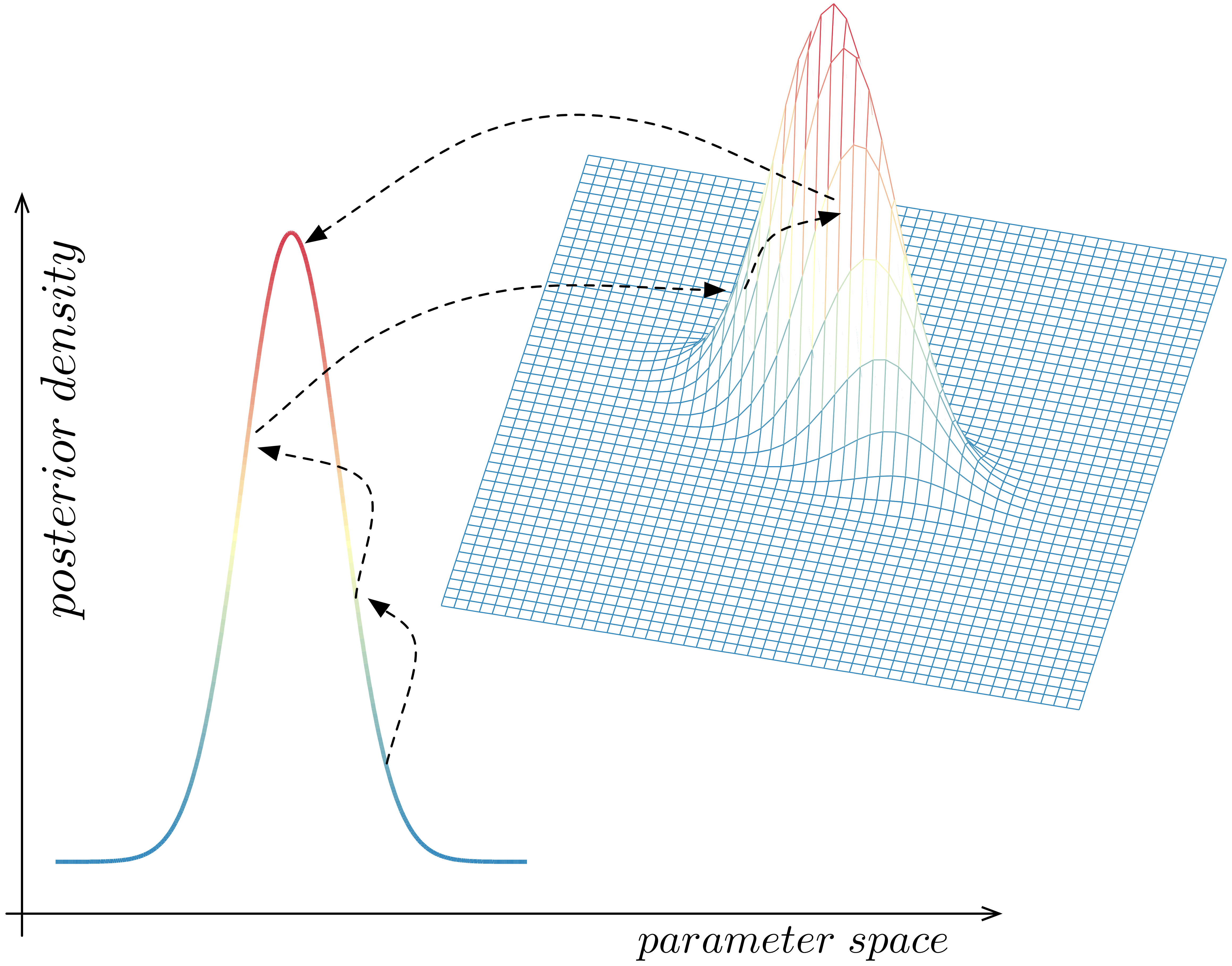
Continue with *non*-fixed dimensionality

Assume a model with a changing dimensionality…



Continue with *non*-fixed dimensionality

Assume a model with a changing dimensionality…



- Same procedure, now generalized for ***k***-order of model. It is organized in two steps.

- Before all, we begin with $\boldsymbol{\theta}_k$ for model ***k***.

1. In-Model Step: The usual MH step, for model ***k***.

2. Outer-Model Step:

  - Propose new $\boldsymbol{\theta}_m$ for model ***m*** from a given proposal distribution *q*.

  - Essentially propose the “birth” or “death” of dimensions at each iteration.

  - Accept, or reject with a probability:

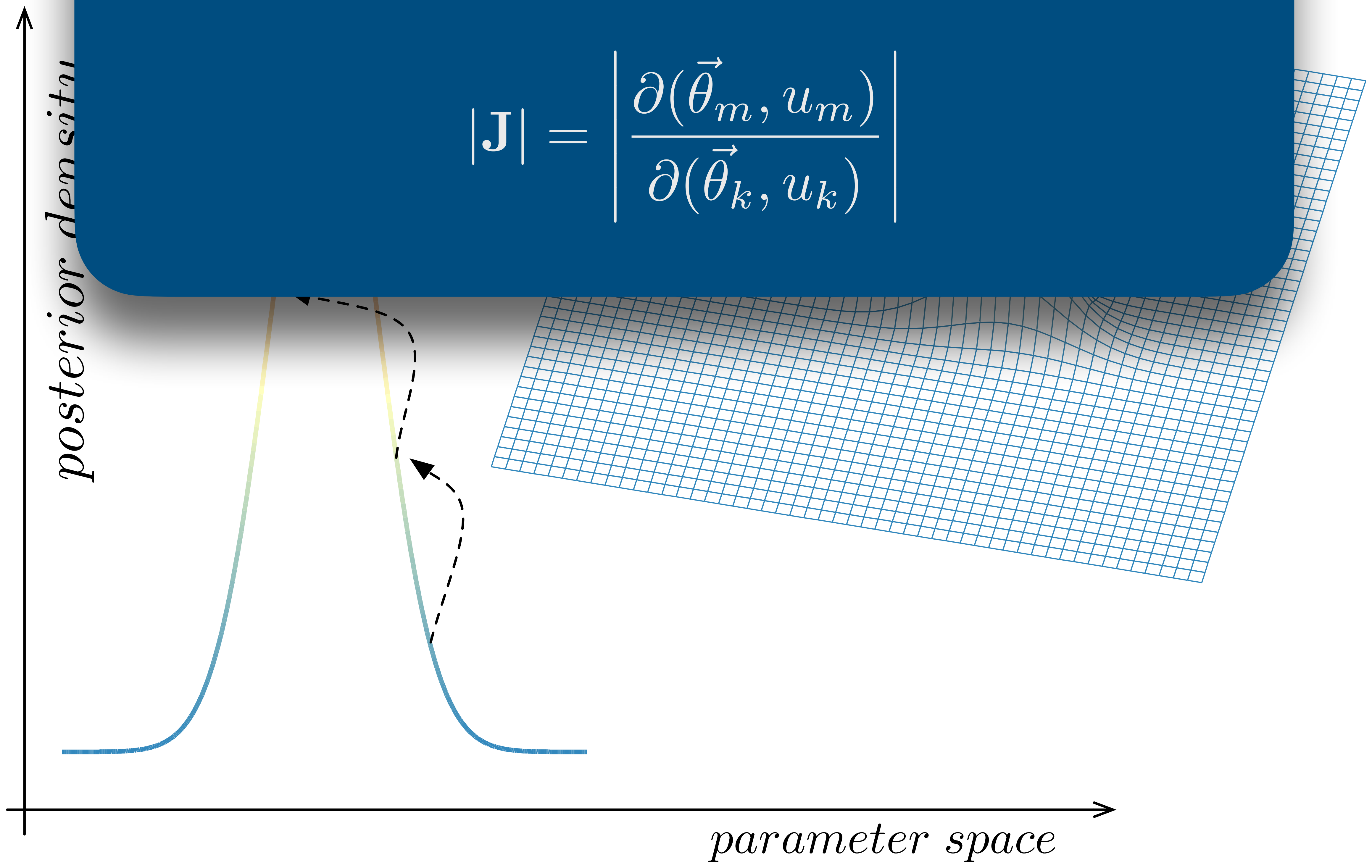
$$\alpha = \min\left[1, \frac{p(y|\vec{\theta}_k)p(\vec{\theta}_k)q(\{k,\vec{\theta}_k\},\{m,\theta_m\})}{p(y,\vec{\theta}_m)p(\vec{\theta}_m)q(\{m,\vec{\theta}_m\},\{k,\theta_k\})}\right]$$

Continue with *non*-fixed dimensionality

Assume a model with a changing dimensionality…

- Usually, this means that we have to compute a Jacobian term at each iteration, which is given by:

$$|\mathbf{J}| = \left| \frac{\partial(\vec{\theta}_m, u_m)}{\partial(\vec{\theta}_k, u_k)} \right|$$



- Same procedure, now generalized for ***k***-order of model. It is organized in two steps.
- Before all, we begin with $\boldsymbol{\theta_k}$ for model ***k***.

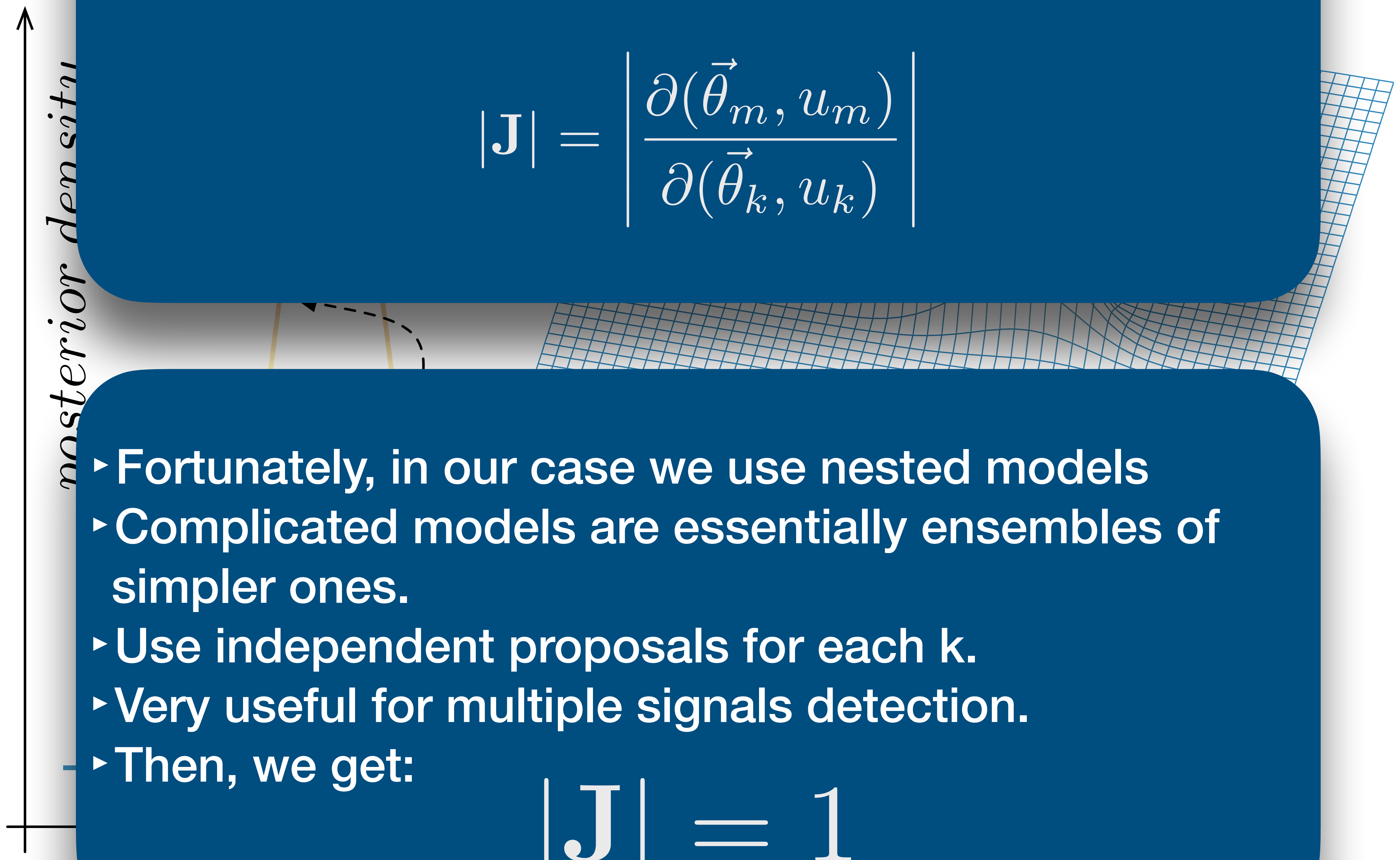
1. In-Model Step: The usual MH step, for model ***k***.
2. Outer-Model Step:

- Propose new $\boldsymbol{\theta_m}$ for model ***m*** from a given proposal distribution *q*.
- Essentially propose the “birth” or “death” of dimensions at each iteration.
- Accept, or reject with a probability:

$$\alpha = \min\left[1, \frac{p(y|\vec{\theta}_k)p(\vec{\theta}_k)q(\{k,\vec{\theta}_k\},\{m,\theta_m\})}{p(y,\vec{\theta}_m)p(\vec{\theta}_m)q(\{m,\vec{\theta}_m\},\{k,\theta_k\})}\right]$$

Continue with *non*-fixed dimensionality

Assume a model with a changing dimensionality…



- Usually, this means that we have to compute a Jacobian term at each iteration, which is given by:

$$|\mathbf{J}| = \left| \frac{\partial(\vec{\theta}_m, u_m)}{\partial(\vec{\theta}_k, u_k)} \right|$$

- Fortunately, in our case we use nested models
- Complicated models are essentially ensembles of simpler ones.
- Use independent proposals for each k.
- Very useful for multiple signals detection.
- Then, we get:

$$|\mathbf{J}| = 1$$

- Same procedure, now generalized for ***k***-order of model. It is organized in two steps.
- Before all, we begin with $\boldsymbol{\theta}_k$ for model ***k***.

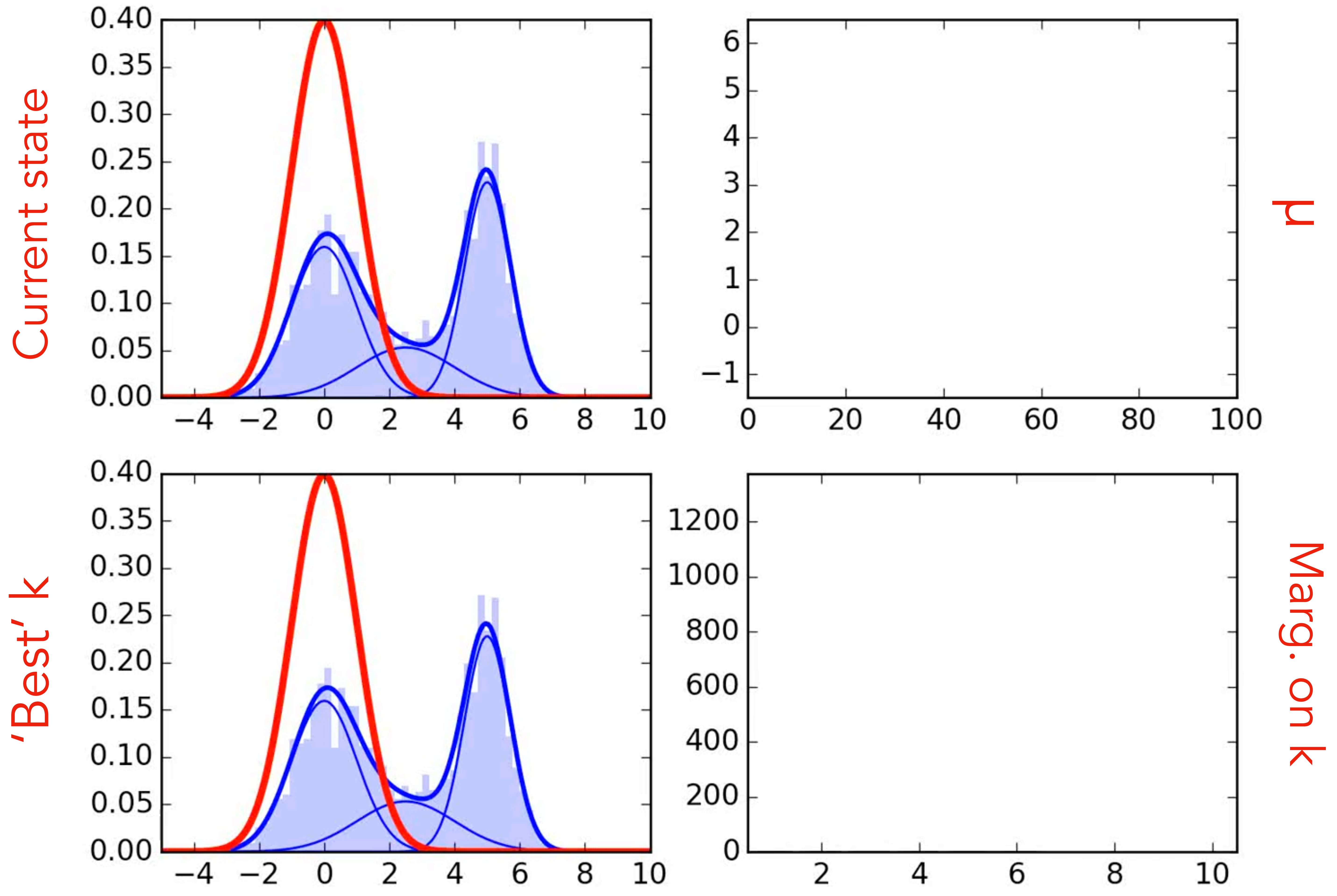
1. In-Model Step: The usual MH step, for model ***k***.
2. Outer-Model Step:

- Propose new $\boldsymbol{\theta}_m$ for model ***m*** from a given proposal distribution *q*.
- Essentially propose the “birth” or “death” of dimensions at each iteration.
- Accept, or reject with a probability:

$$\alpha = \min\left[1, \frac{p(y|\vec{\theta}_k)p(\vec{\theta}_k)q(\{k,\vec{\theta}_k\},\{m,\theta_m\})}{p(y,\vec{\theta}_m)p(\vec{\theta}_m)q(\{m,\vec{\theta}_m\},\{k,\theta_k\})}\right]$$

A simple example.

Searching for Gaussian pulses.



Video source: https://www.youtube.com/watch?v=wBTGoA_dIlo

What about the LISA Data Analysis?

It really sounds quite painful to achieve convergence…

Two ways to improve

One focusing on the sampler, the second on the waveforms/likelihoods.

- Ensemble Walkers.
- Delayed Rejection.
- Multiple Try.
- Parallel Tempering.

- CPU parallelization.
- GPU accelerated Waveforms.
- Advanced Search techniques.
- Efficient proposals.

On the sampler side

- Ensemble Walkers.
- Delayed Rejection.
- Multiple Try.
- Parallel Tempering.

On the sampler side

- **Ensemble Walkers.**
- Delayed Rejection.
- Multiple Try.
- Parallel Tempering.

D. Foreman-Mackey +, 2013

- Run multiple walkers in parallel.
- Sample a transform of the parameters:

$$\vec{\zeta} = A\vec{\theta} + b$$

$$p_{A,\,b}(\vec{\zeta}|y) = p_{A,\,b}(A\vec{\theta} + b|y) \propto p(\vec{\theta}|y)$$

- Less sensitive to covariance “features”.
- Use walkers to draw candidates (stretch proposal).

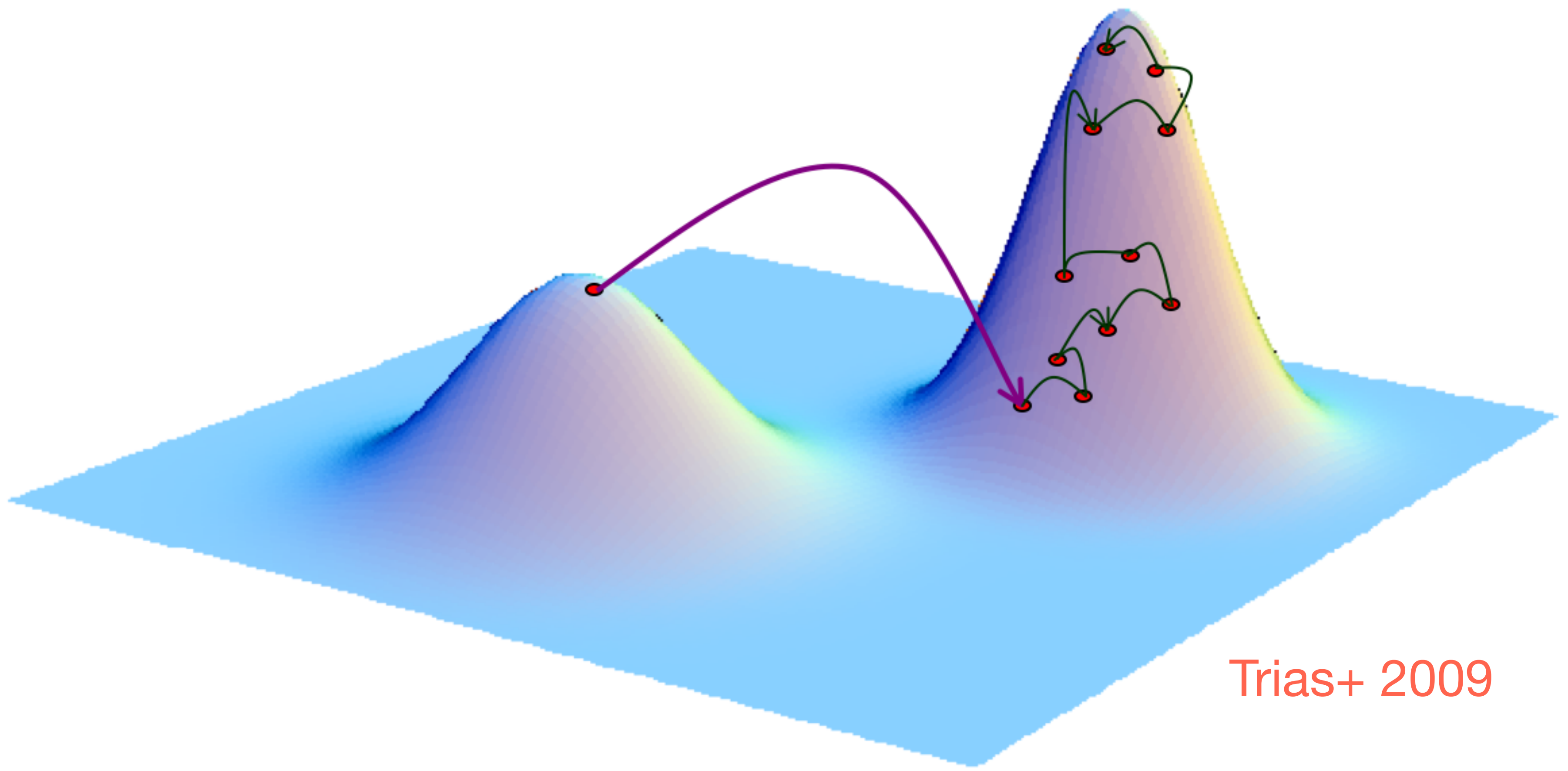
▲ Allows for locating secondary maxima.

▲ Healthier chains, good mixing.

▲ Parallelizable.

On the sampler side

- Ensemble Walkers.
- **Delayed Rejection.**
- Multiple Try.
- Parallel Tempering.



Trias+ 2009

$$1 \wedge \left\{ \frac{p(\vec{\theta}_2|y)q(\vec{\theta}_1,\vec{\theta}_0)q(\vec{\theta}_2,\vec{\theta}_1,\vec{\theta}_0)\left[1-\alpha_1\left(\vec{\theta}_2,\vec{\theta}_1\right)\right]}{p(\vec{\theta}_0|y)q(\vec{\theta}_0,\vec{\theta}_1)q(\vec{\theta}_0,\vec{\theta}_1,\vec{\theta}_2)\left[1-\alpha_1\left(\vec{\theta}_0,\vec{\theta}_1\right)\right]} \right\}$$

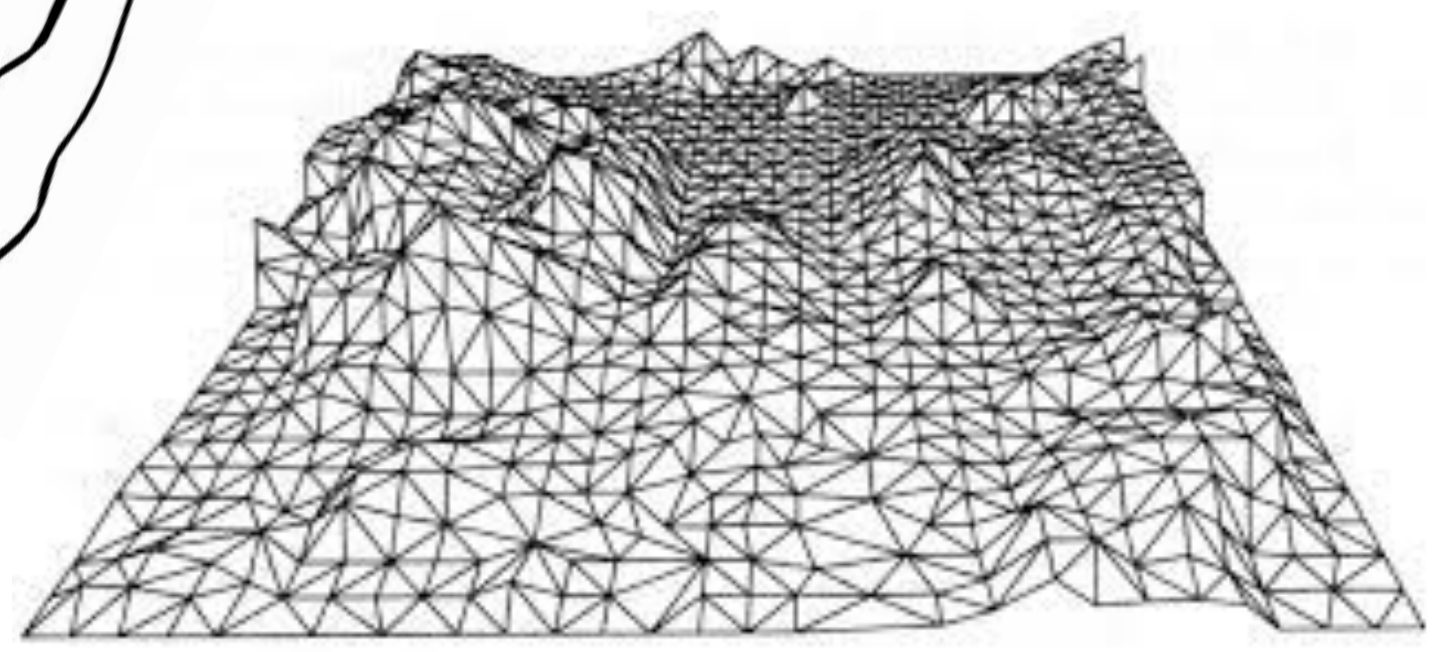
▲ Allows for locating secondary maxima.

▲ Healthier chains, good mixing.

▼ Serial calculations

On the sampler side

- Ensemble Walkers.
- Delayed Rejection.
- **Multiple Try.**
- Parallel Tempering.



$$\alpha(\vec{\theta}_{t-1}, \vec{\theta}_t) = 1 \wedge \left\{ \frac{w(\vec{\theta}_t^j) + \sum_{n, n\neq j}^{N} w(\vec{\theta}_t^n)}{w(\vec{\theta}_{t-1}) + \sum_{n, n\neq j}^{N} w(\vec{\theta}_t^n)} \right\}$$

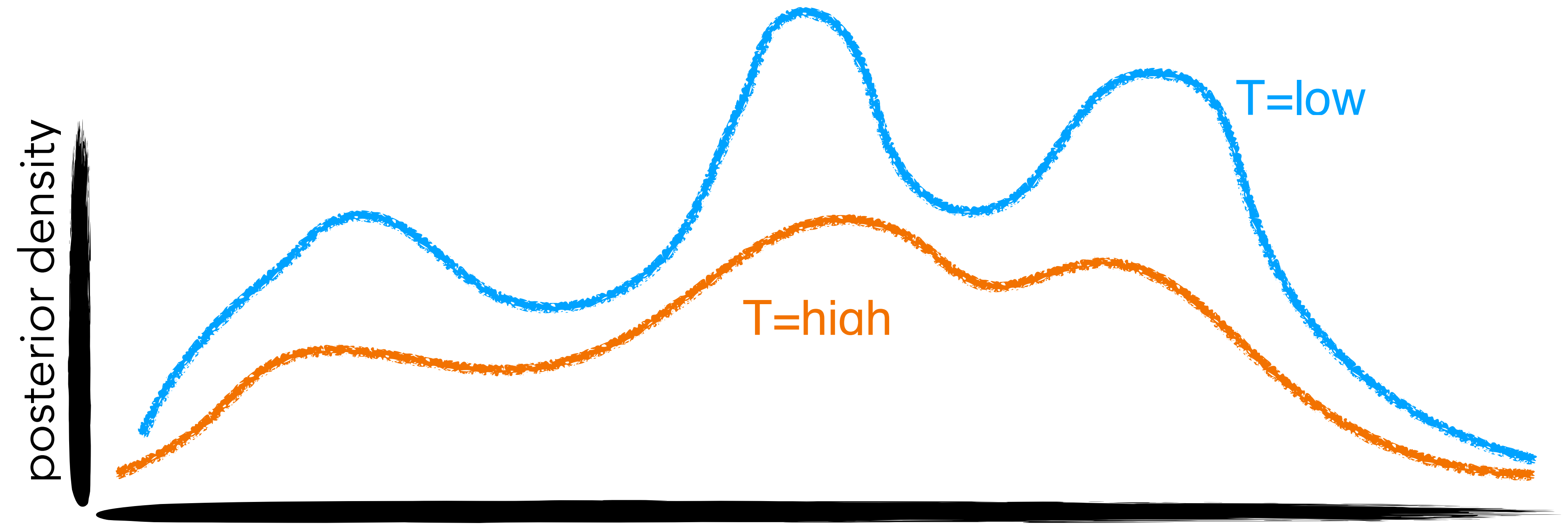
▲ Allows for mapping the posterior surface.

▲ Healthier chains, good mixing.

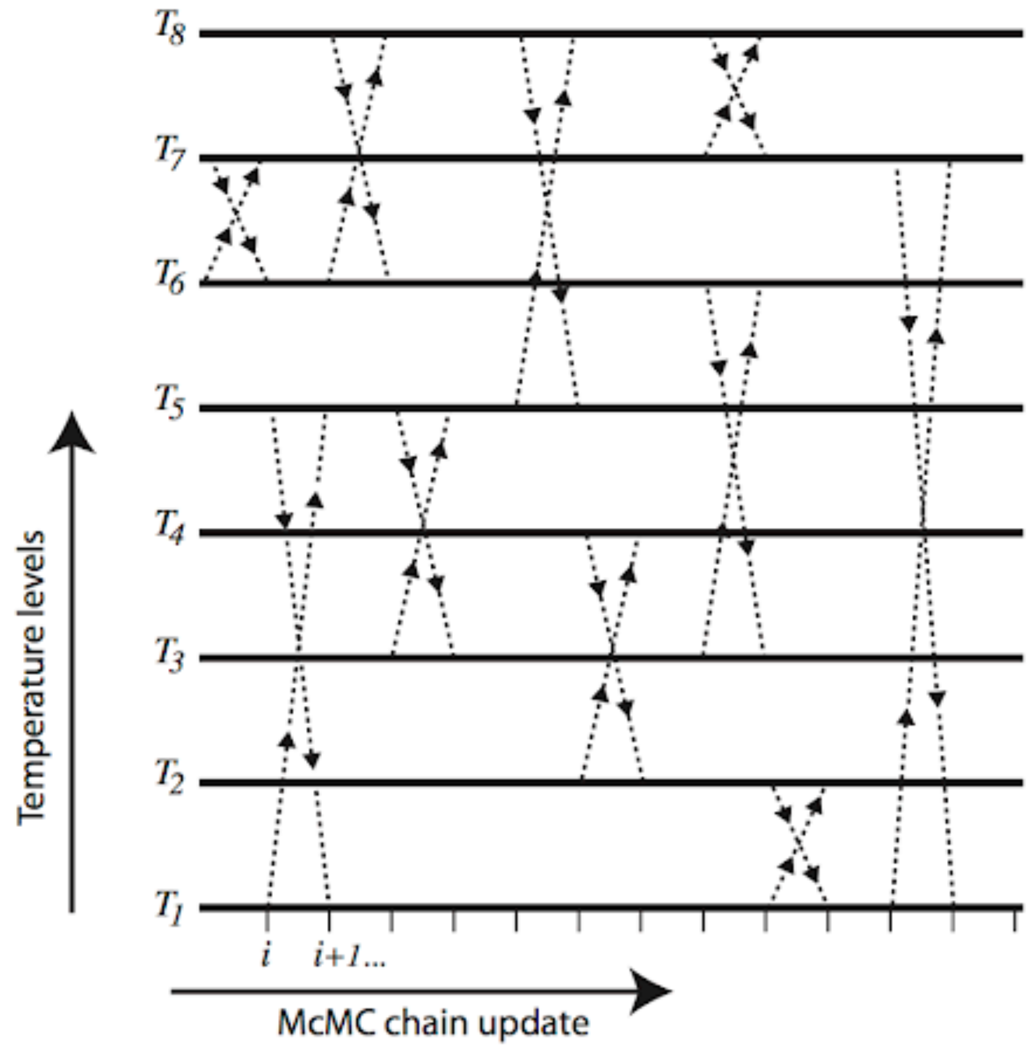
▼ Many likelihood evaluations.

On the sampler side

- Ensemble Walkers.
- Delayed Rejection.
- Multiple Try.
- **Parallel Tempering.**



$$p_T(\vec{\theta}|y) \propto p(y|\vec{\theta})^{1/T} p(\vec{\theta})$$



$$\alpha_{i,j} = 1 \wedge \left\{ \left(\frac{p(y|\vec{\theta}_i)}{p(y|\vec{\theta}_j)} \right)^{\beta_j - \beta_i} \right\}$$

$$Z(\beta) \equiv \int L(\vec{\theta})^{\beta} p(\vec{\theta})\, \mathrm{d}\vec{\theta},$$

$$\Delta \log Z \equiv \log Z(1) - \log Z(0) = \int_0^1 \mathrm{E}[\log L]_\beta \, \mathrm{d}\beta$$

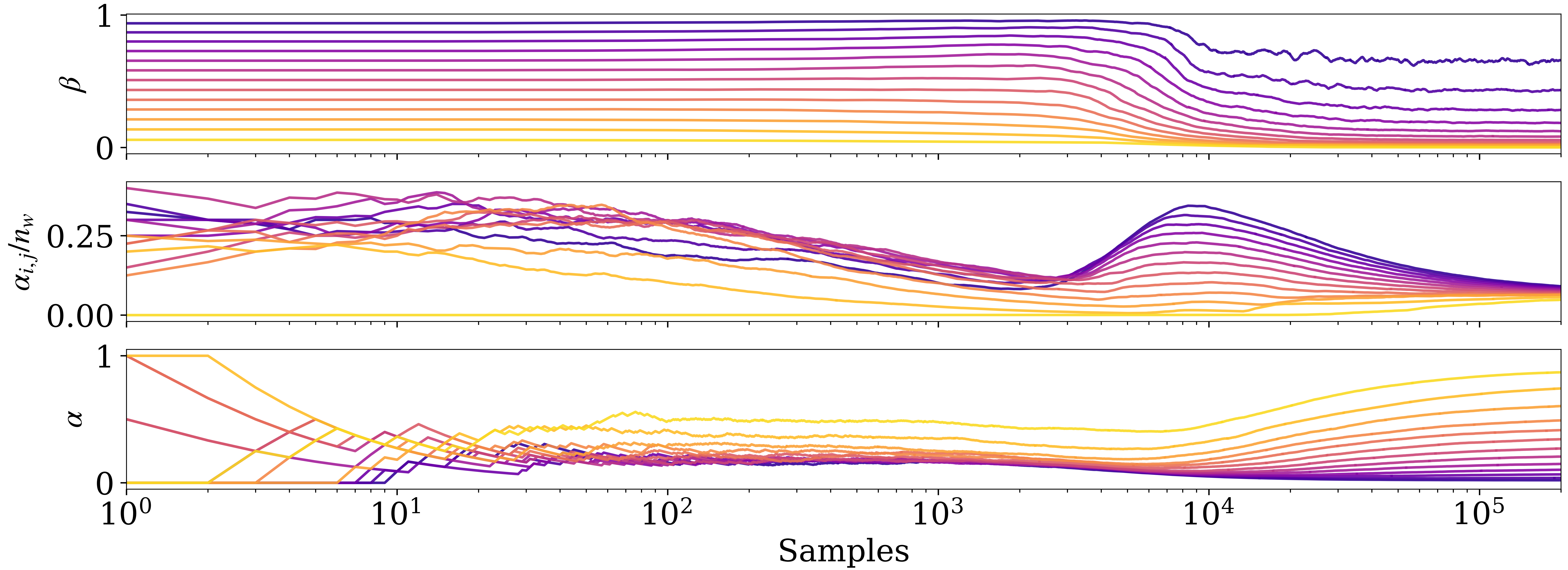
▲ Allows for mapping the posterior surface.

▲ Healthier chains, good mixing.

▼ Many likelihood evaluations.

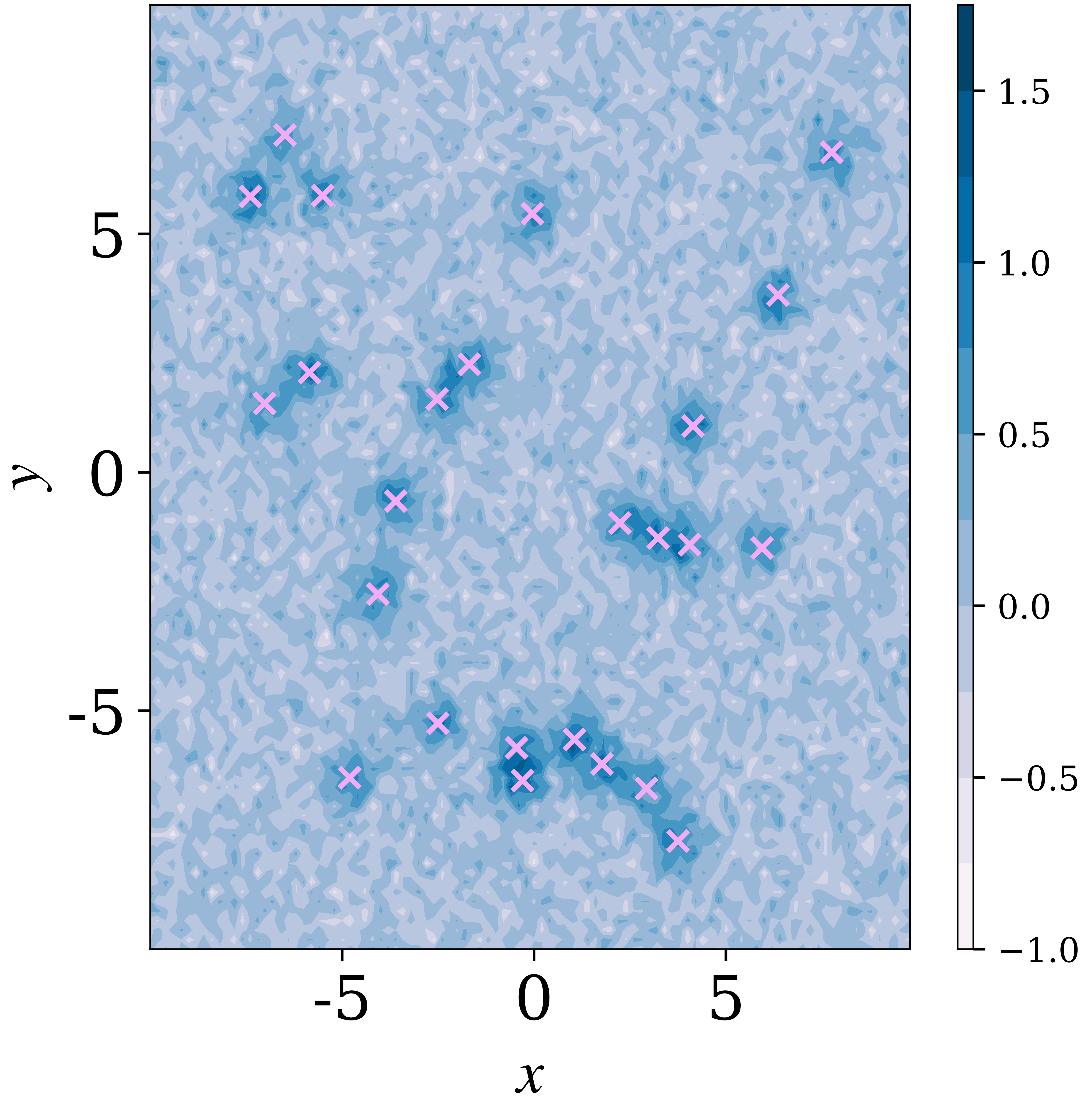
Adaptive Parallel Tempering in action

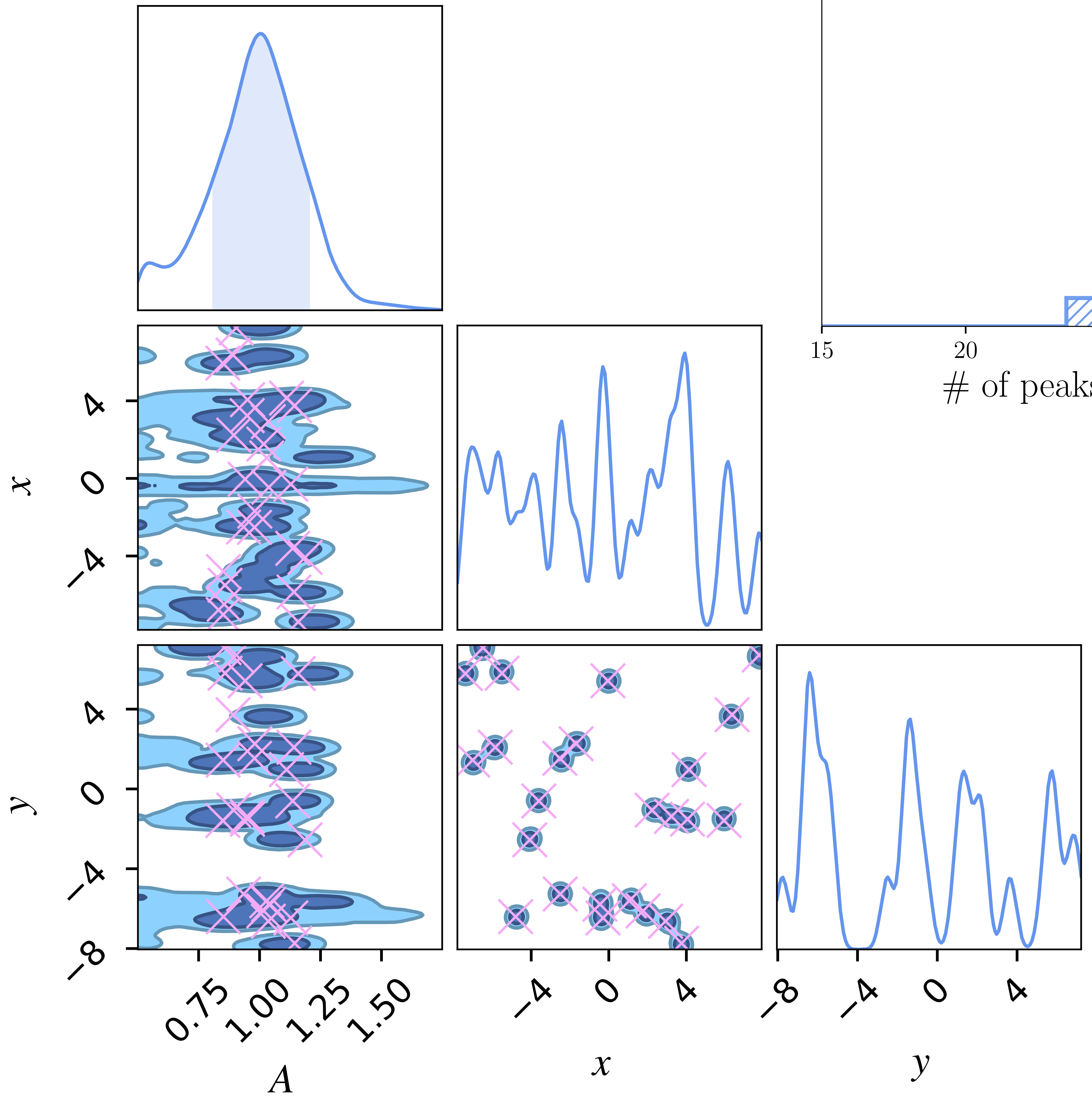
As presented in Vousden et al 2016

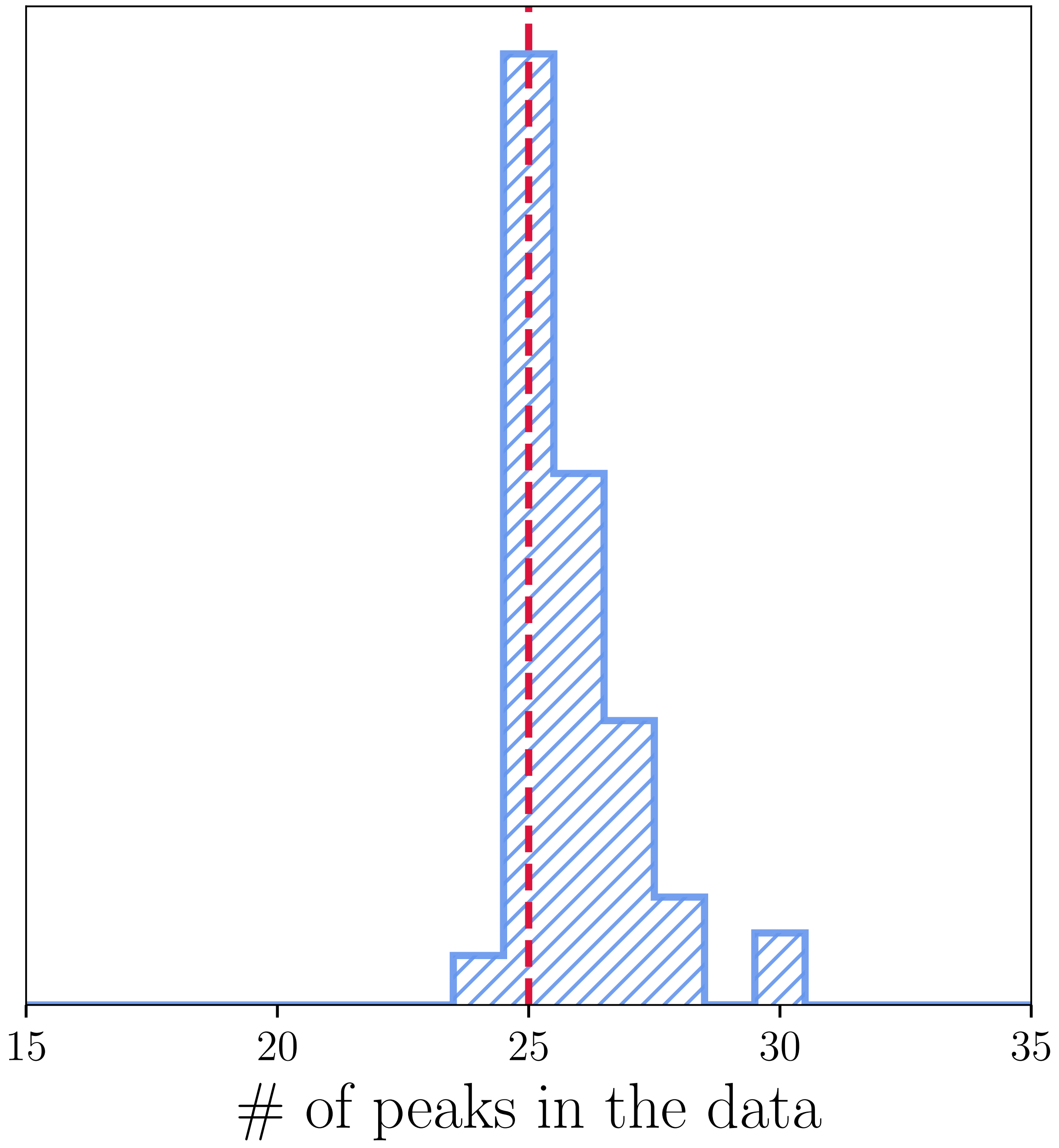


Simple applications

Usually encountered in data analysis

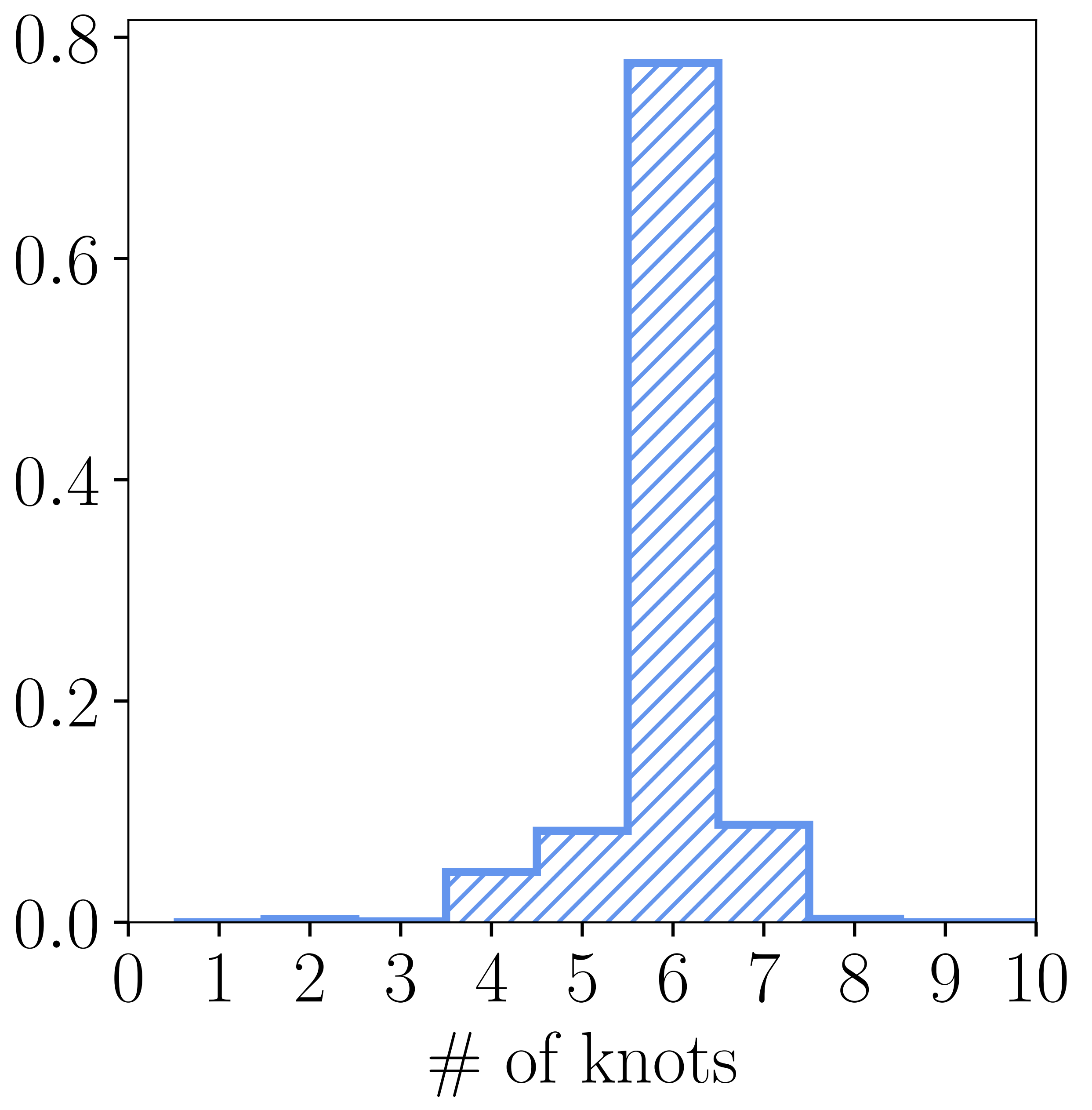


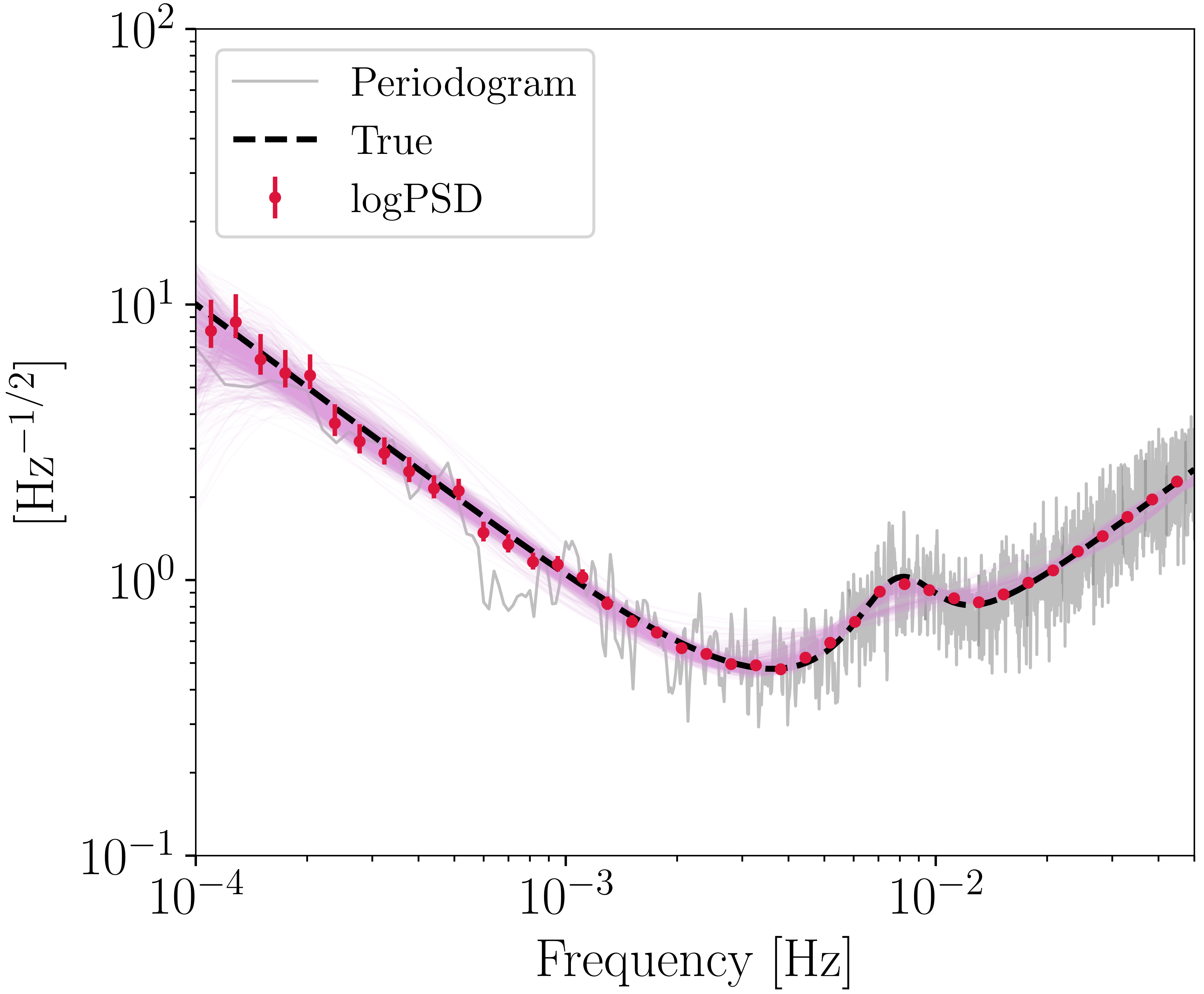


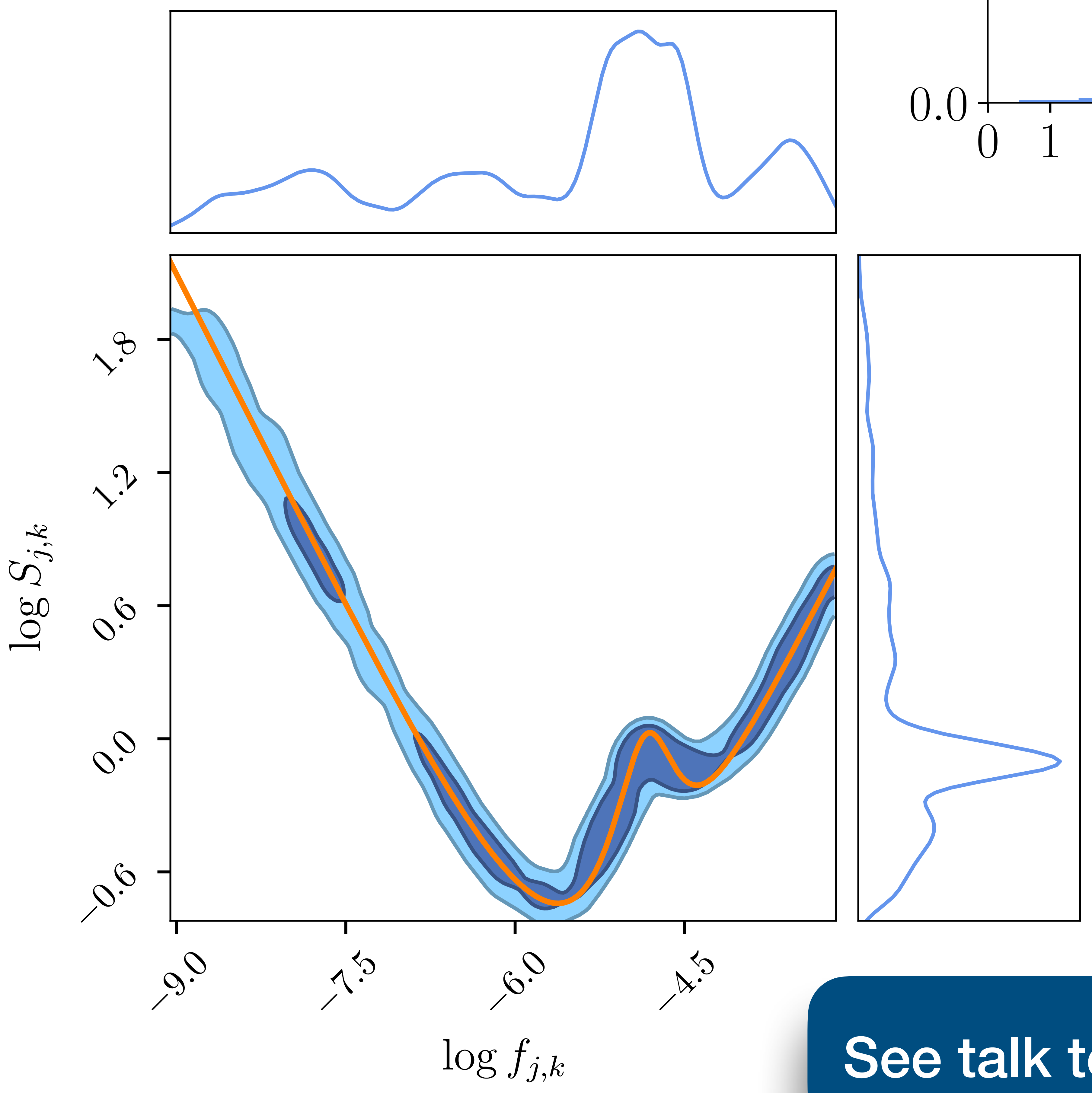


Simple applications

Usually encountered in data analysis







See talk tomorrow by Q. Baghi

A realization

All of the above methods for improving, have one requirement:

▼ Many likelihood evaluations.

- Running this machinery with limited resources requires a long convergence time.
- Parallelizing really helps!
  - Analyze the data in segments.
  - Parallelize MCMC processes.
- GPU waveforms/likelihoods change the game!

Erebor!

A proposal for a pipeline

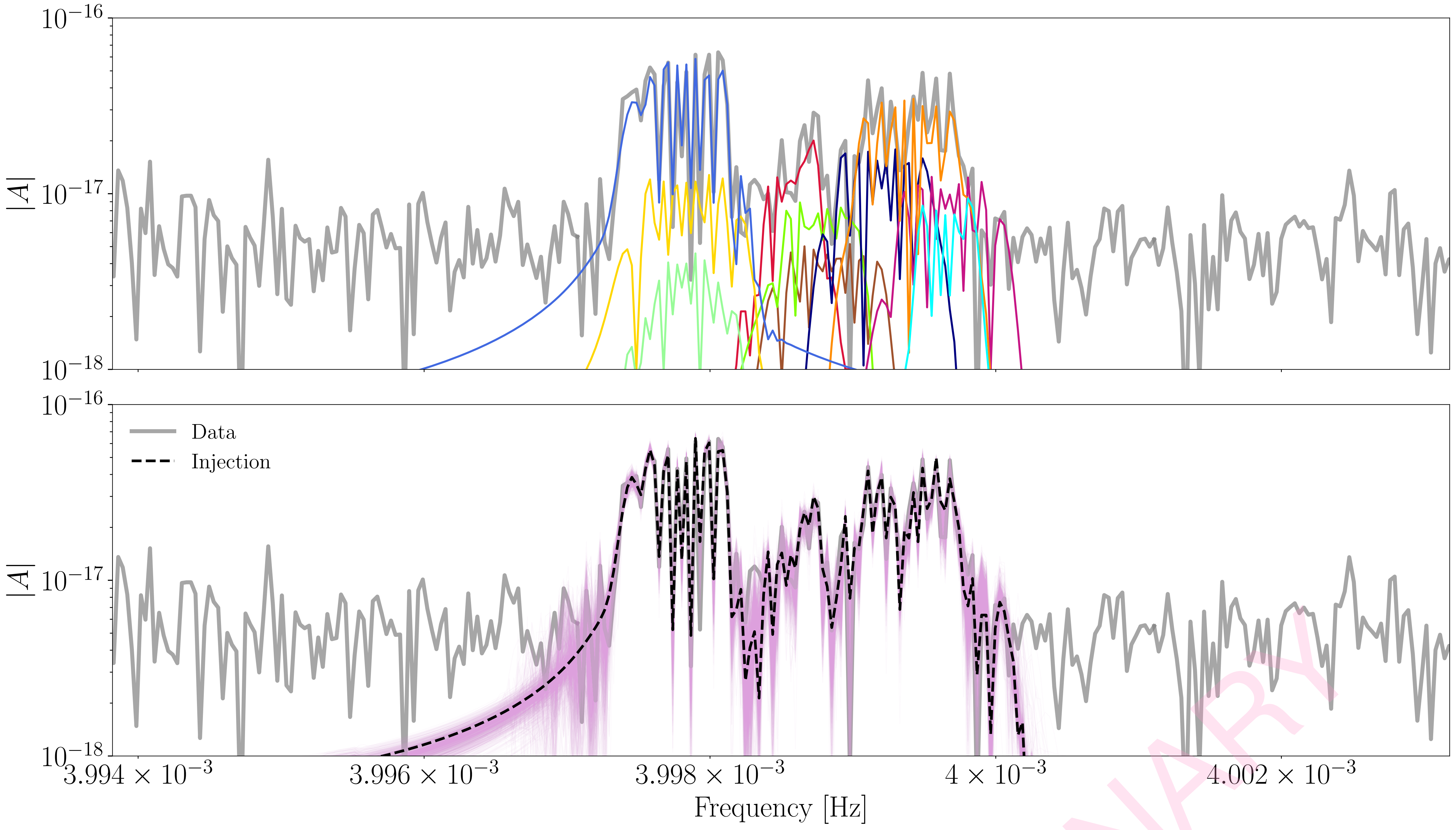
- M. Katz, J Gair (AEI), & N. Korsakova (APC Paris), N Stergioulas, NK (AUTh).

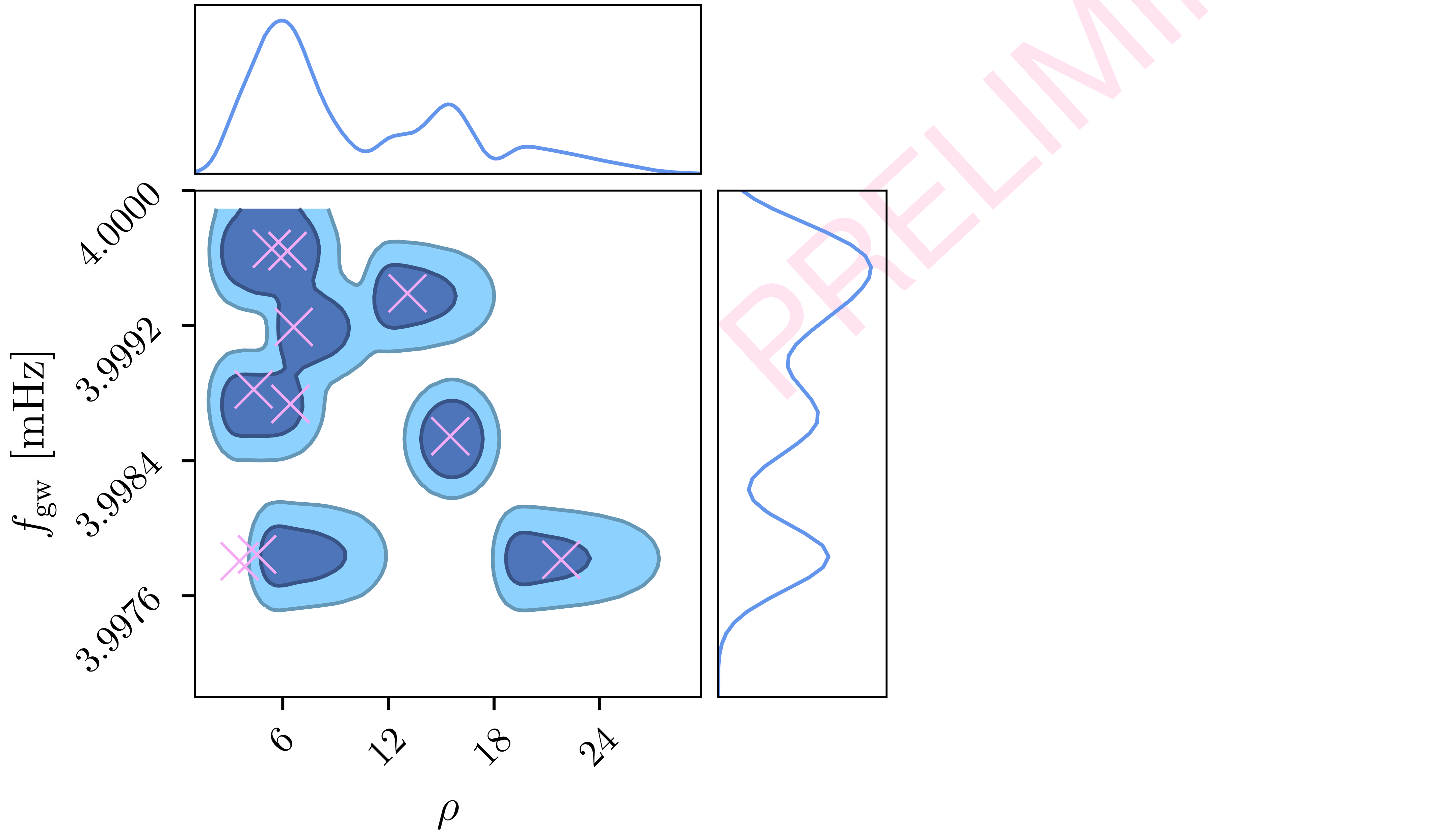
➡https://github.com/mikekatz04/Eryn

- Using Eryn plus:

1. GPU accelerated Waveforms.
2. Advanced Search techniques.
3. Efficient proposals.
4. Ability to search for multiple models.

See next talk of K. Lackeos!





And yet…

Tuning this type of algorithms is hard!

- The scale of the problem of LISA DA is huge!
- The algorithm needs to be exactly fine-tuned to the specific problem, also depending on the frequency band, also accounting for other types of sources, also …
- Convergence greatly depends on efficiently sampling:
    - Need to improve acceptance rate.
- This is where different improvements/enhancements can enter.

Proposal distributions are crucial

An example application, part of Erebor, led by N. Korsakova

- First run a search phase on the data.
- Subtract “loudest” sources.
- Get an estimate of residuals,
- and then run a set of RJ MCMC on those residuals, looking for the “harder-to-get” lower SNR signals.
- Use those samples we to construct efficient proposals!

Proposal distributions are crucial

An example application, part of Erebor, led by N. Korsakova

- First run a search phase on the data.
- Subtract “loudest” sources.
- Get an estimate of residuals,
- and then run a set of RJ MCMC on those residuals, looking for the “harder-to-get” lower SNR signals.
- Use those samples we to construct efficient proposals!

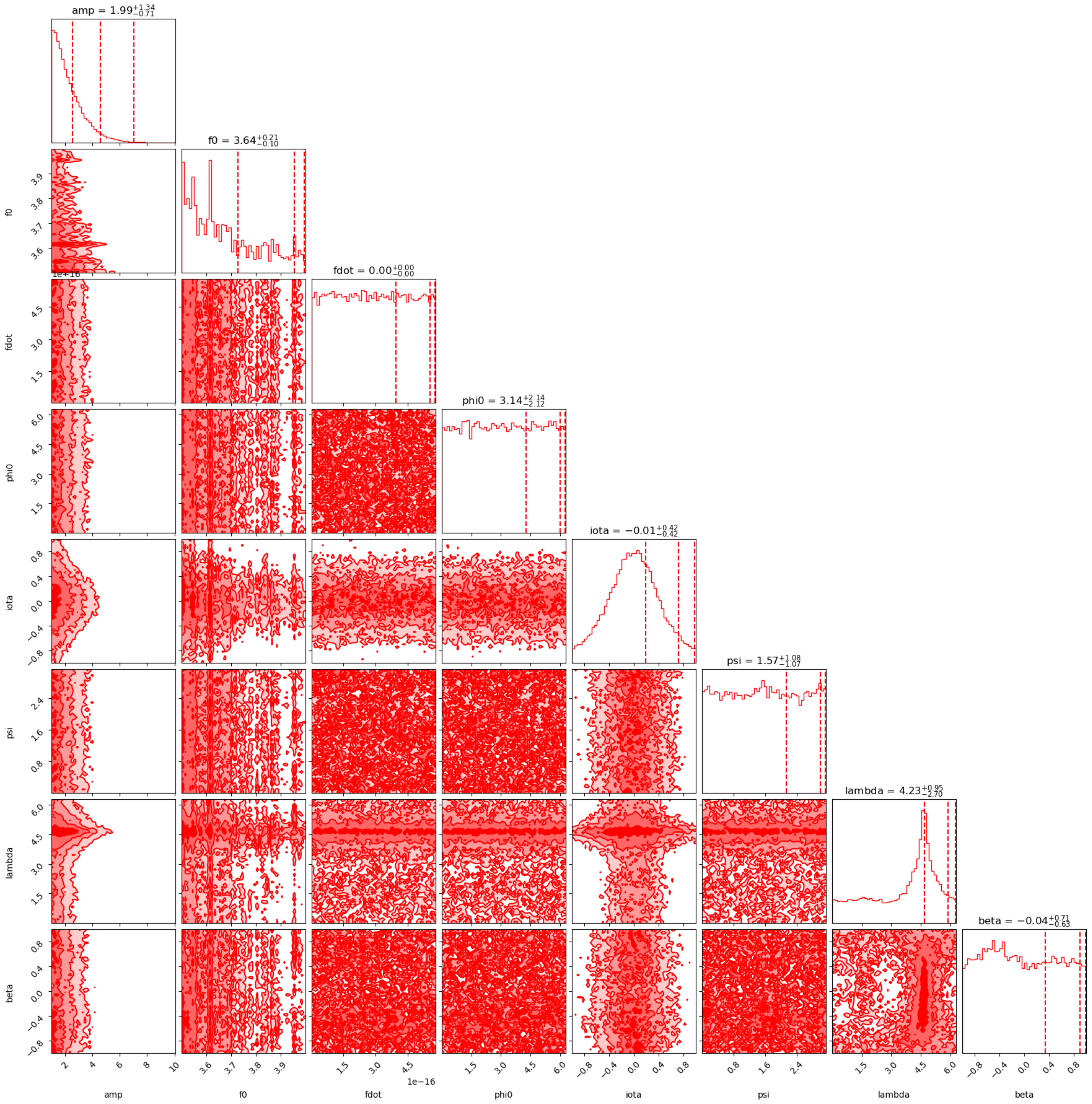
- Fit probability distribution function from the samples.
- Use Normalising Flows as a density estimator.
- Train network by optimising Kullback–Leibler divergence between samples and transformed base distribution.

$$KL(p||q) = \sum_x p(x) \log \left[\frac{p(x)}{q(x)} \right]$$

- Use estimated distribution for proposals.

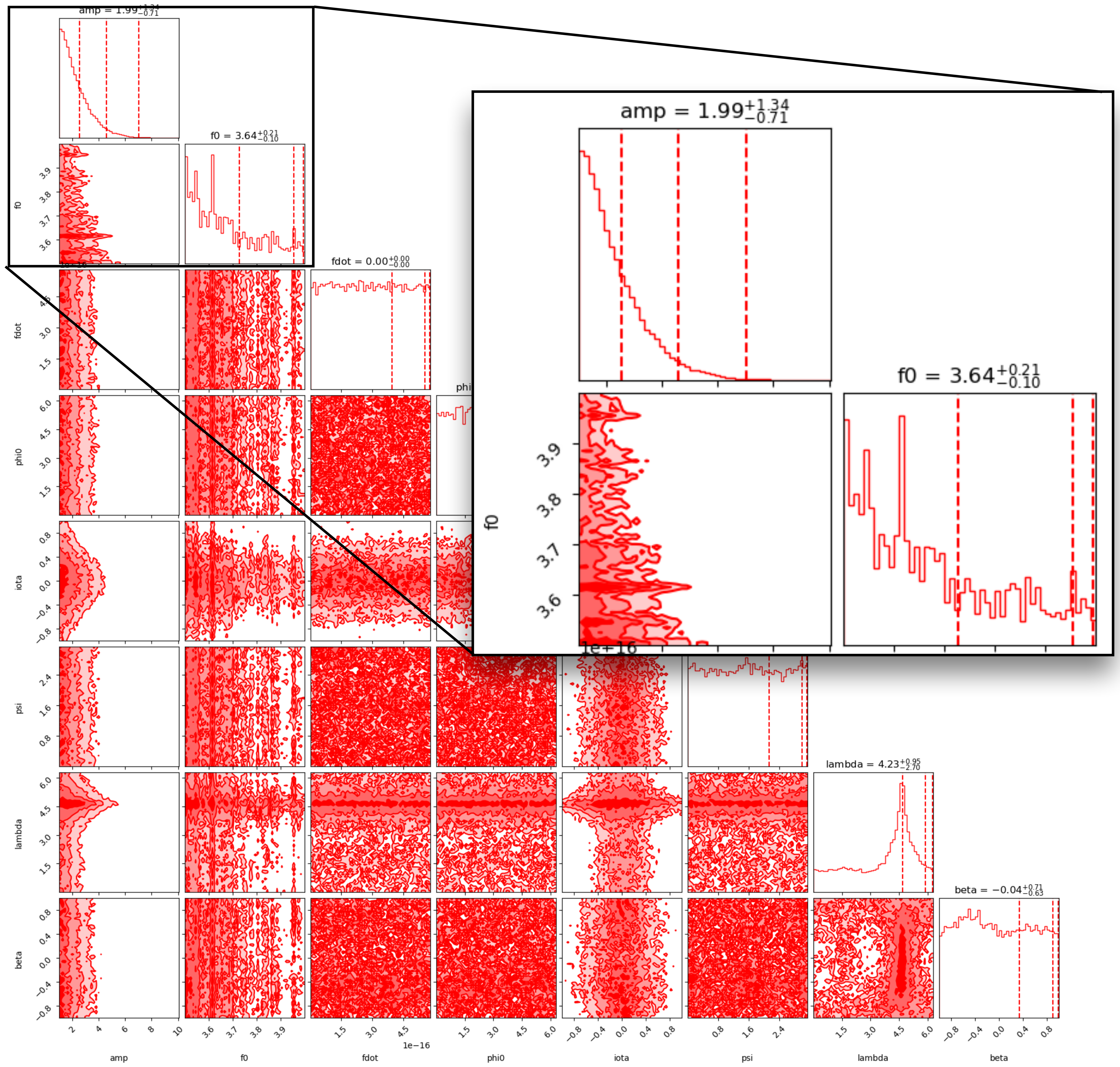
Proposal distributions are crucial

An example application, part of Erebor, led by N. Korsakova



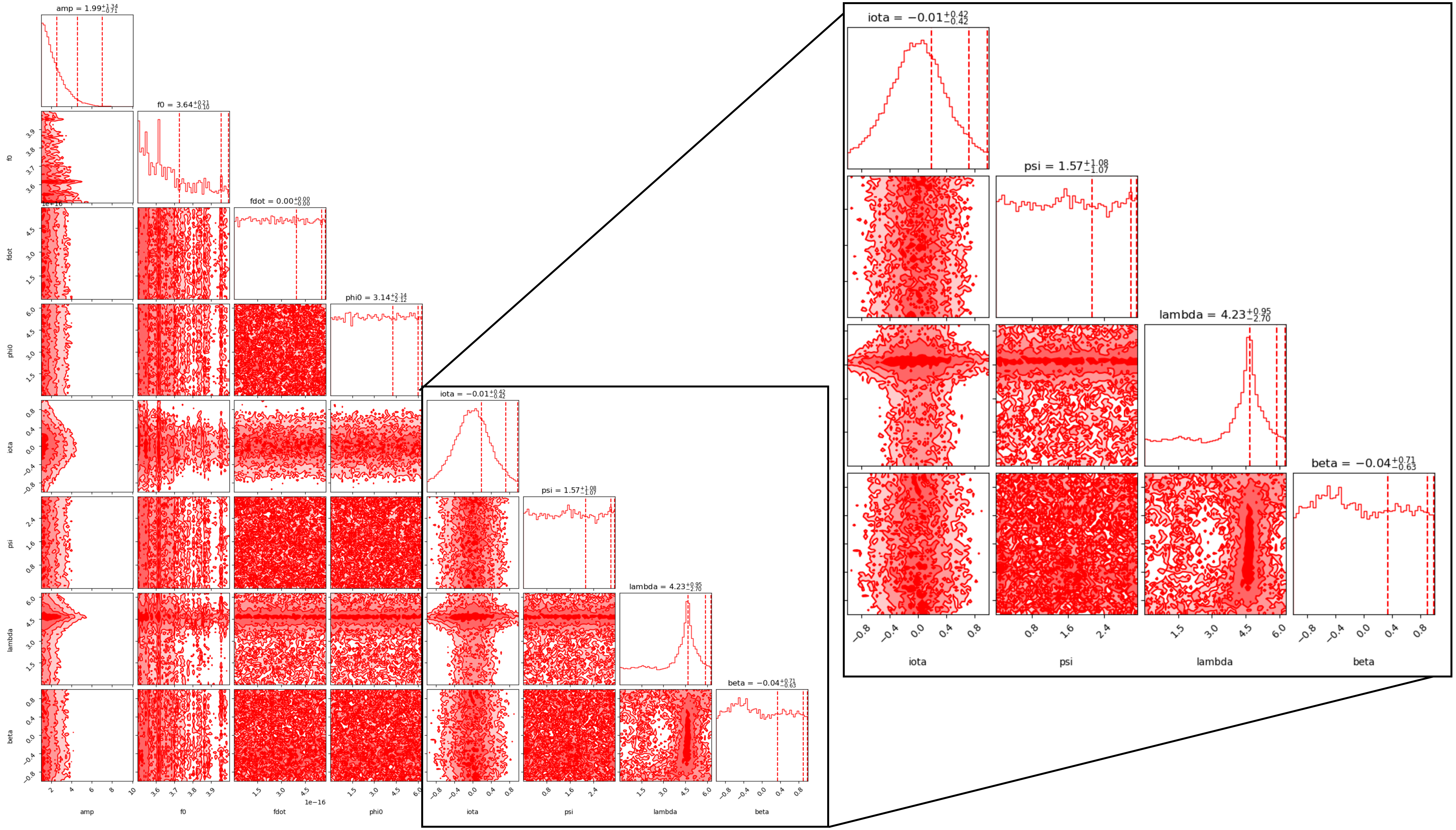
Proposal distributions are crucial

An example application, part of Erebor, led by N. Korsakova



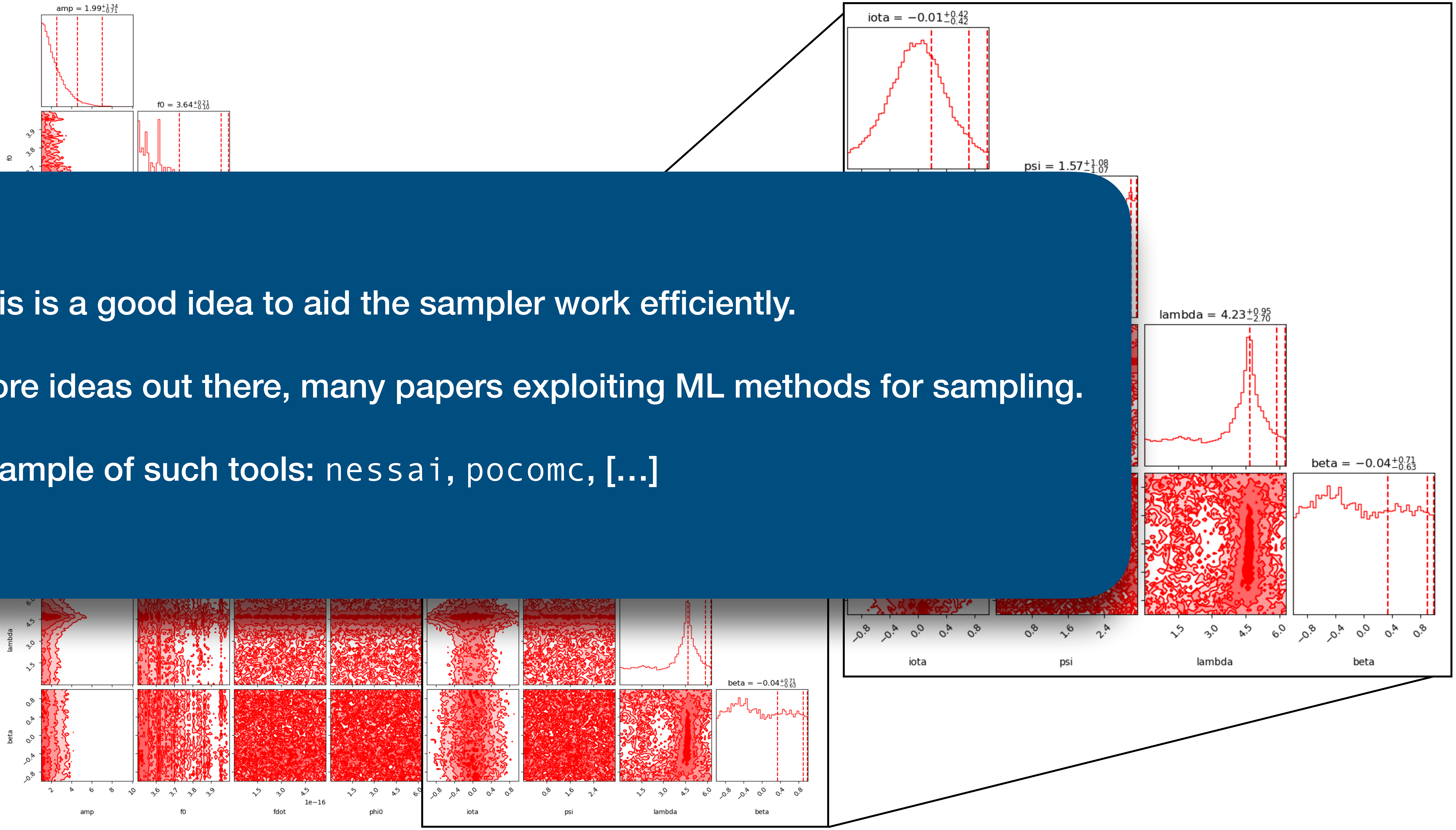
Proposal distributions are crucial

An example application, part of Erebor, led by N. Korsakova



Proposal distributions are crucial

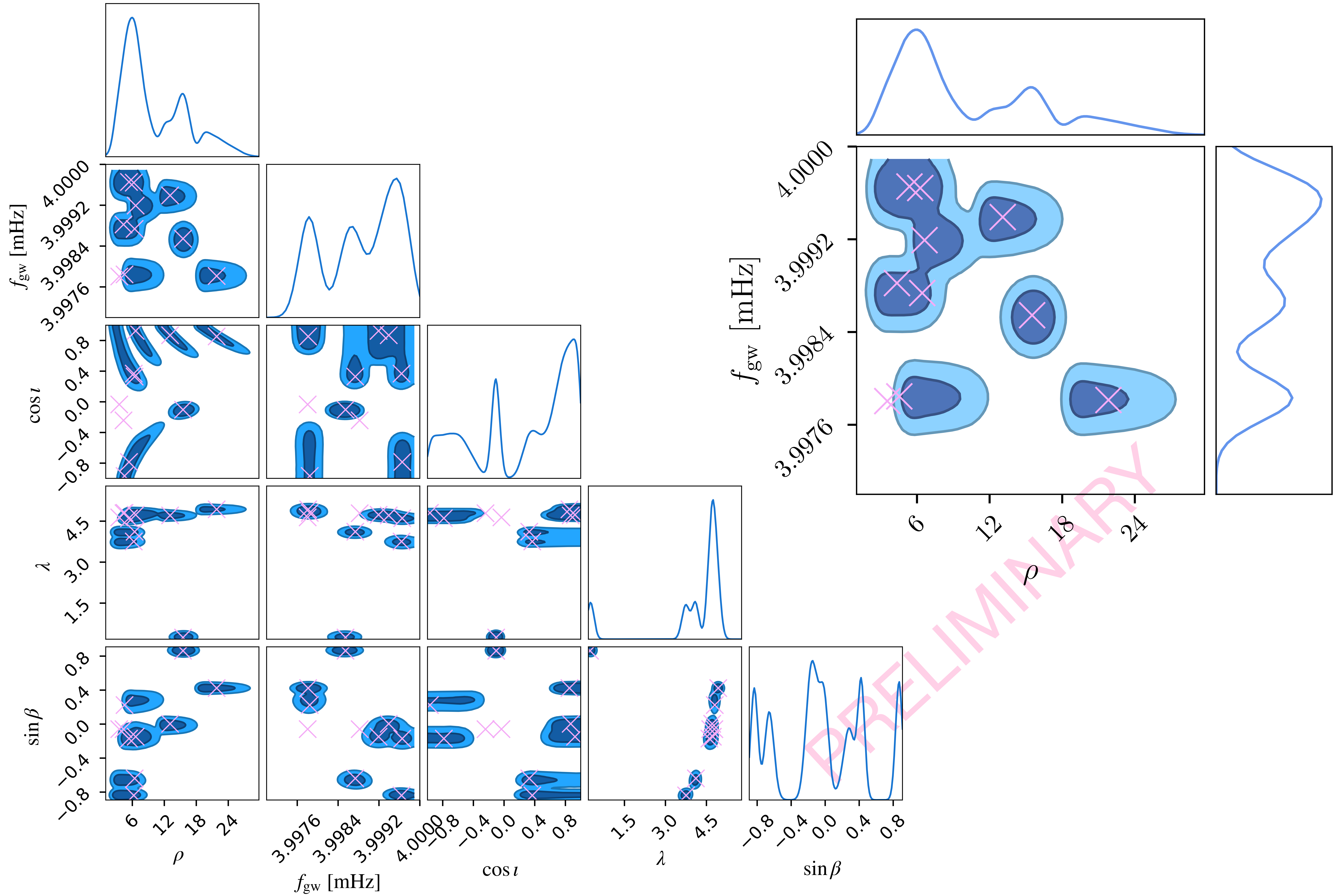
An example application, part of Erebor, led by N. Korsakova



- This is a good idea to aid the sampler work efficiently.
- More ideas out there, many papers exploiting ML methods for sampling.
- Example of such tools: `nessai`, `pocomc`, […]

- Trans-dimensional methods for LISA global fit are extremely useful.
- But also very hard to tune and scale them to the problem.
- We are in a good state though! [See next talk by K. Lackeos]
- Novel methods can help ease the burden of stochastic methods, or in some cases replace them entirely.
- The scale of the problem is so large, that any improvement counts!

Έξτρα Ματέριαλ



ρ
f_gw [mHz]
cos ι
λ
sin β
4.0000
3.9992
3.9984
3.9976
6
12
18
24
PRELIMINARY

Posterior: $$p(k|y) = \frac{p(y|k)p(k)}{p(y)} = \frac{\int_{\Theta_k} p(y|\theta_k, k)p(\theta_k|k)d\theta_k \, p(k)}{p(y)}$$

Acceptance ratio: $$\alpha = \min\left(1, \frac{p(k', \theta_{k'}|y) \, q_1(k; k')q_2(u)}{p(k, \theta_k|y) \, q_1(k'; k)q_2(u')} \left|\frac{\partial(\theta_{k'}, u)}{\partial(\theta_k, u')}\right|\right)$$

Proposal distributions for dimension matching parameters: $q_2(u)$ and $q_2(u')$

Map functions for different k: $\theta_{k'} = g(\theta_k, u') \qquad \theta_k = g(\theta_{k'}, u)$

Jacobian is Unity when we use independent proposals: $\theta_k = g(\theta_{k'}, u) = u$ and $\theta_{k'} = g(\theta_k, u') = u'$

Godsill, 2001