DUNE long baseline physics meeting

July 15, 2024

Testing algorithms for neutrino oscillation parameter estimation

Leïla Haegel, Romain Faure, Mathis Roinsard / *IP2I Lyon*

Status

- **Neutrino parameter estimation in DUNE:**
  - MaCh3 is the current "official" parameter estimation software for DUNE
  - Samples the posterior distribution of neutrino oscillation parameters
  - Based on Markov-Chain Monte-Carlo with Metropolis-Hastings algorithm (documentation about the method here and here)

- **MaCh3:**
  - Created for T2K, modified to be experiment-independent when ported over DUNE
  - Perform event-by-event reweighing to estimate the parameters
  - Requires O(weeks) for oscillation parameter estimation (beam neutrinos)

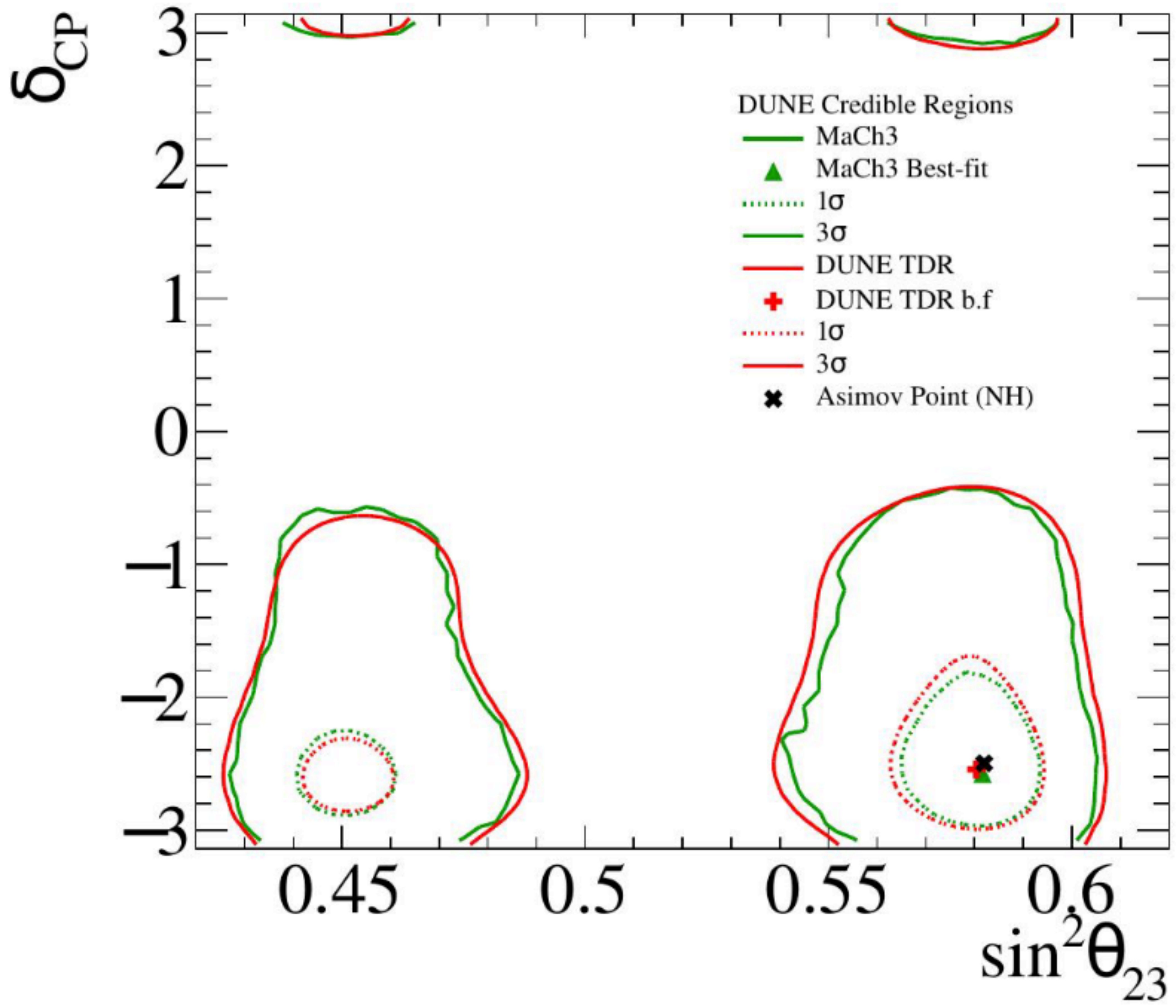
Open questions and potential answers

- **Open questions in DUNE long-baseline (LBL) group:**
  - How many points are needed to extract a 5σ contour?
  - Can we have a faster sampler for sensitivity analyses?
  - Do we want another parameter estimator for cross-checks & validation?

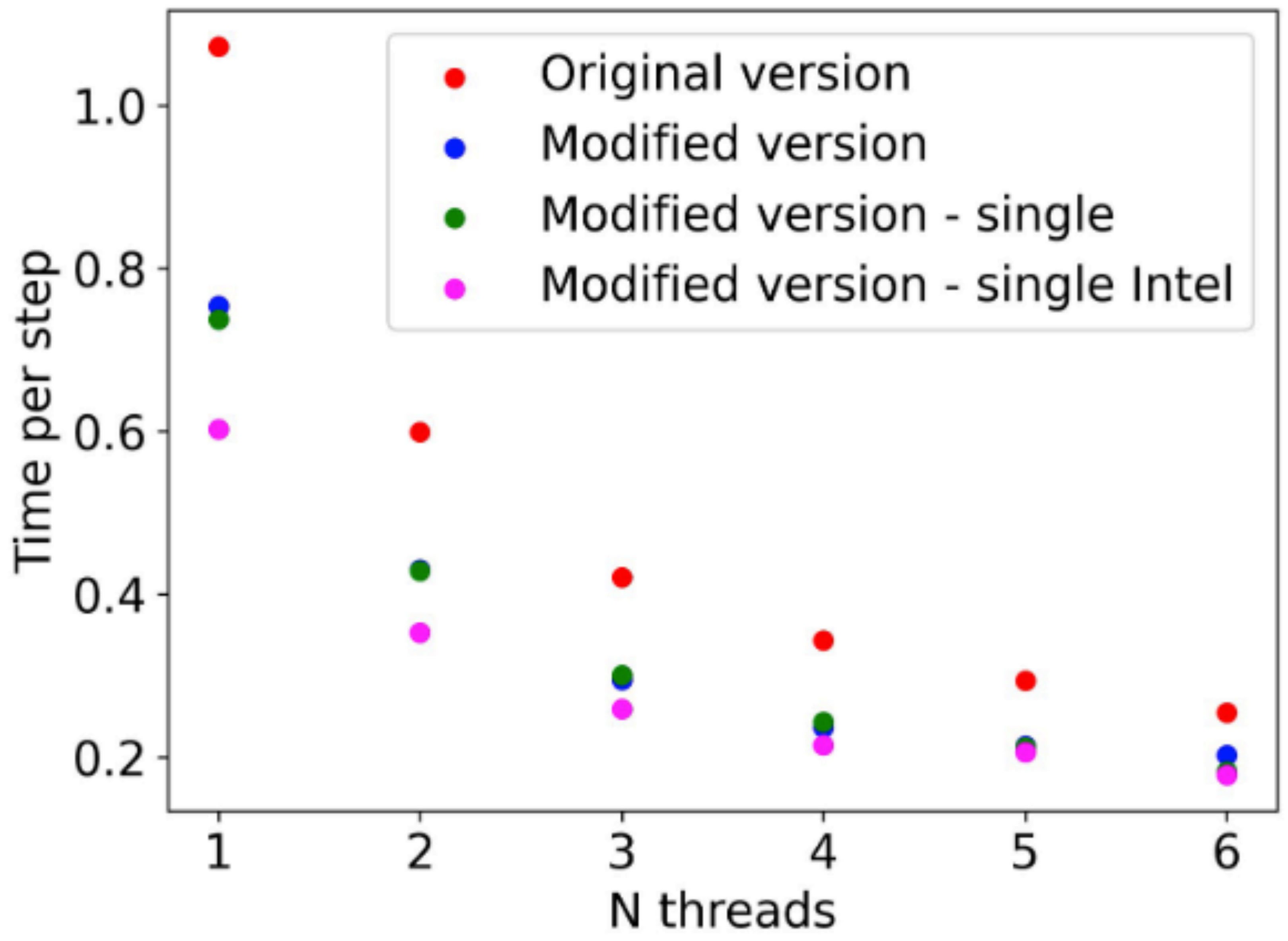
- **This study:**
  - Demonstration of other sampling algorithms for parameter estimation
  - Ensemble sampling → w/ Mathis Roinsard
  - Nested sampling → w/ Romain Faure

MaCh3 performances

- **January 2024 CM:**
  - MaCh3 with 1.8 x 10^8 steps

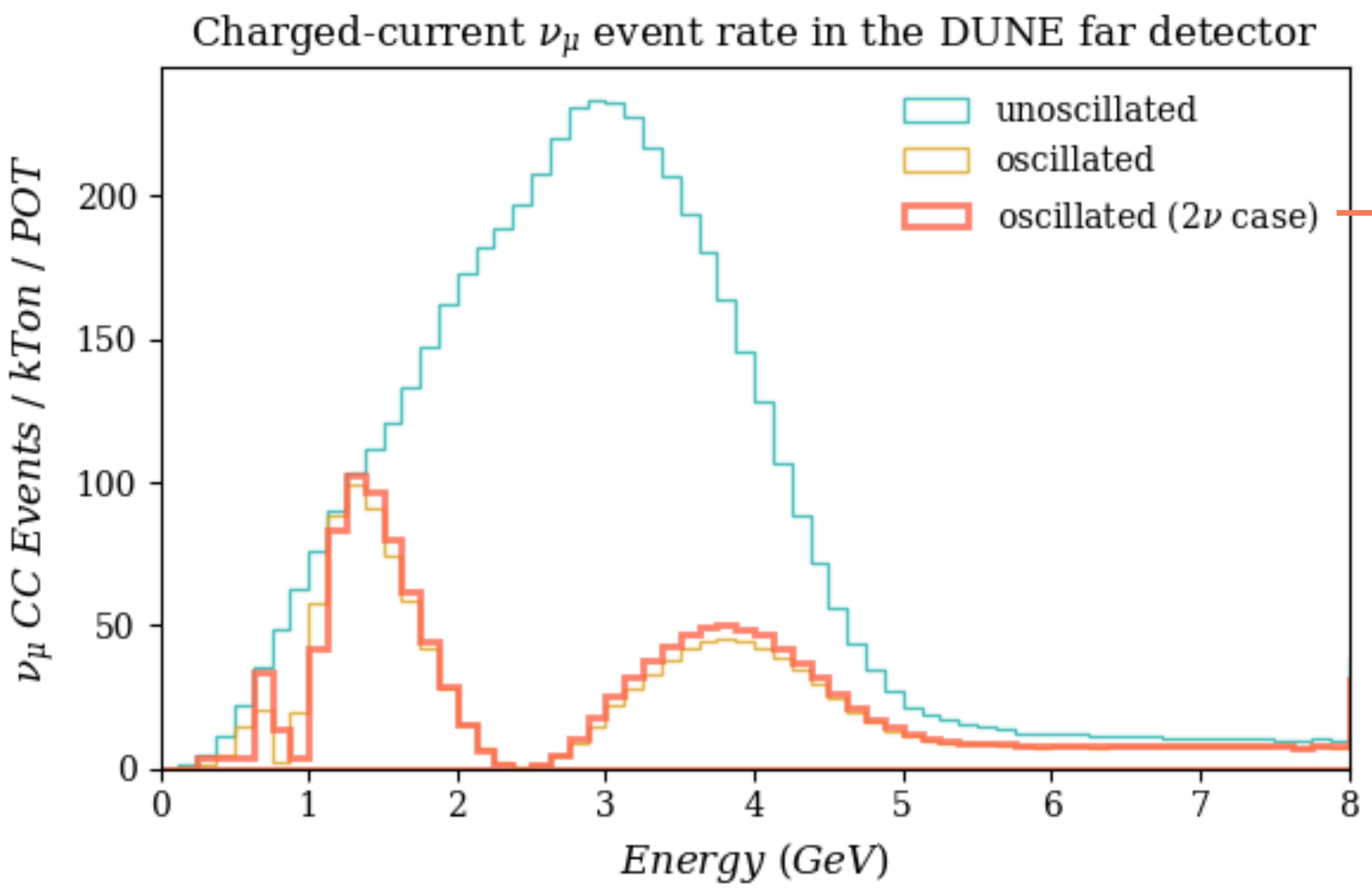


- **Since:**
  - Pierre Granger worked on accelerating the software
  - Implementation of faster features from T2K presented at May CM



DUNE oscillation toy model

- **Muon neutrino disappearance**
  - Uses event rate from DUNE Technical Design Report: arXiv:2103.04797
  - Normalised for 1 year of beam data
  - Approximate the oscillation probability with the 2-flavour equation
  - Notebooks available here

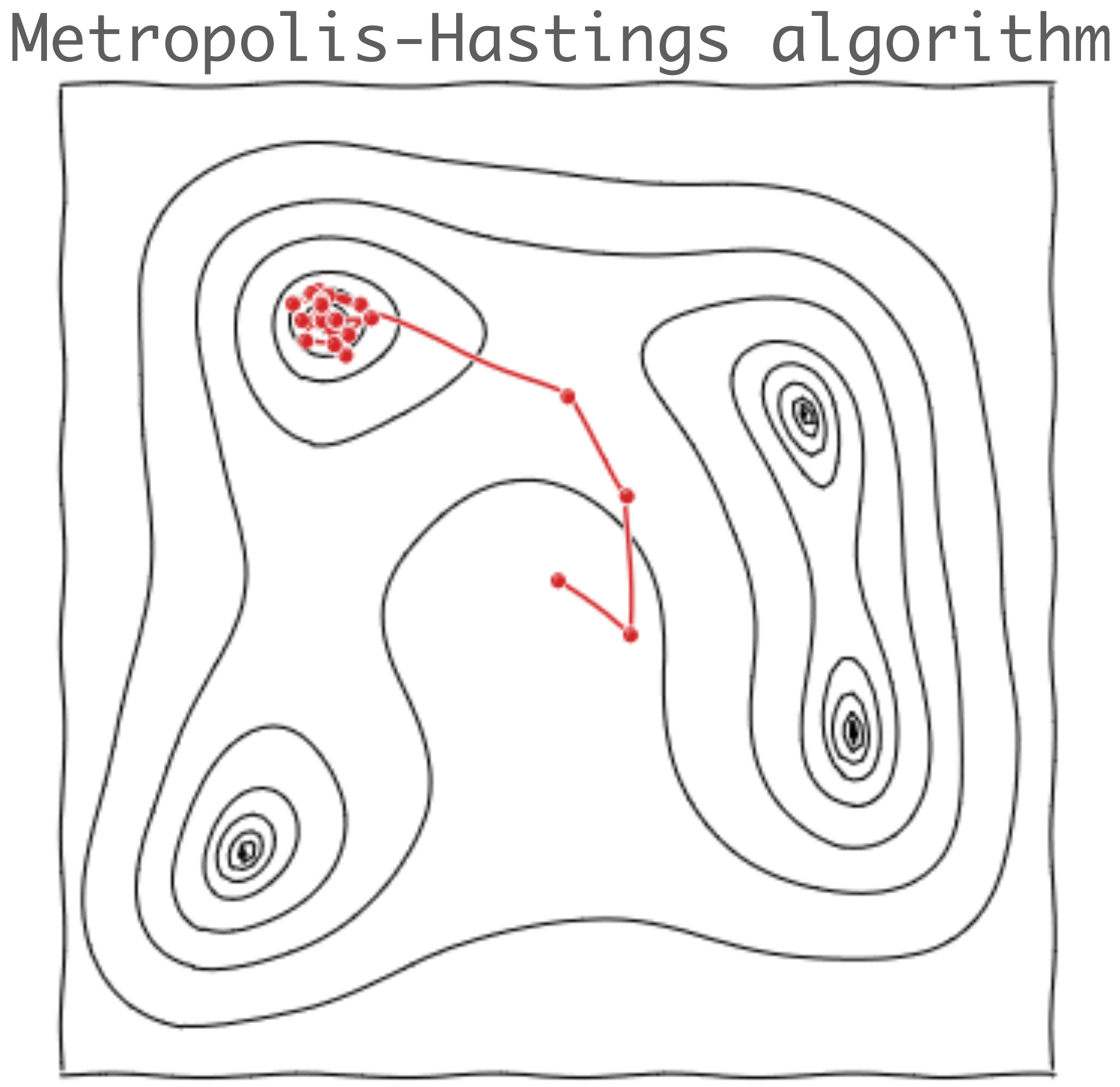


$$P_{\nu_e \to \nu_\mu}(l, E) = \sin^2(\theta)\sin^2\left(\frac{\Delta m_{ij}^2 l}{4E}\right)$$

with $\begin{cases} \theta = \pi/4 \\ \Delta m^2 = 2.2\ \mathrm{eV}^2 \end{cases}$

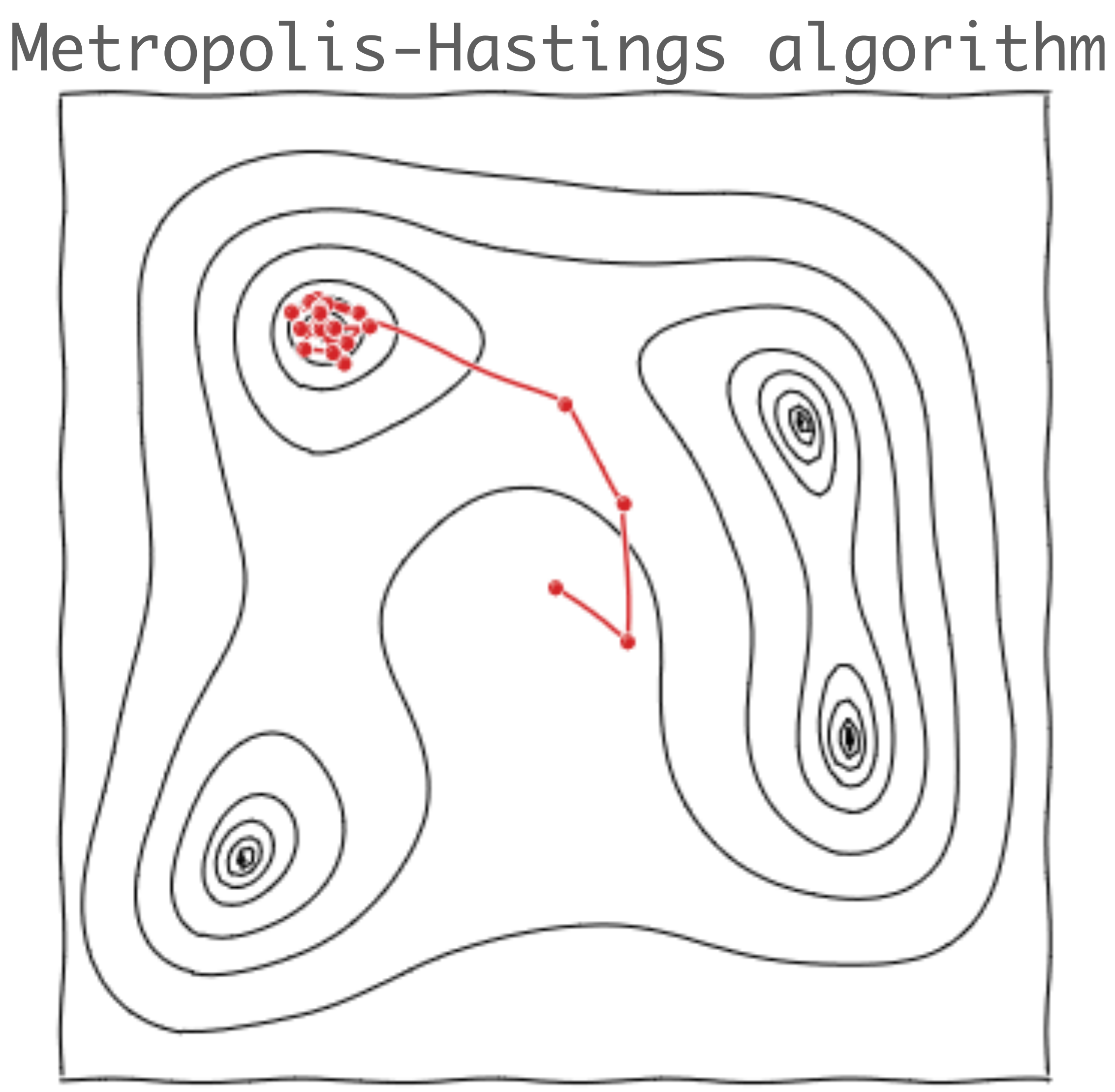
MCMC with Metropolis-Hasting

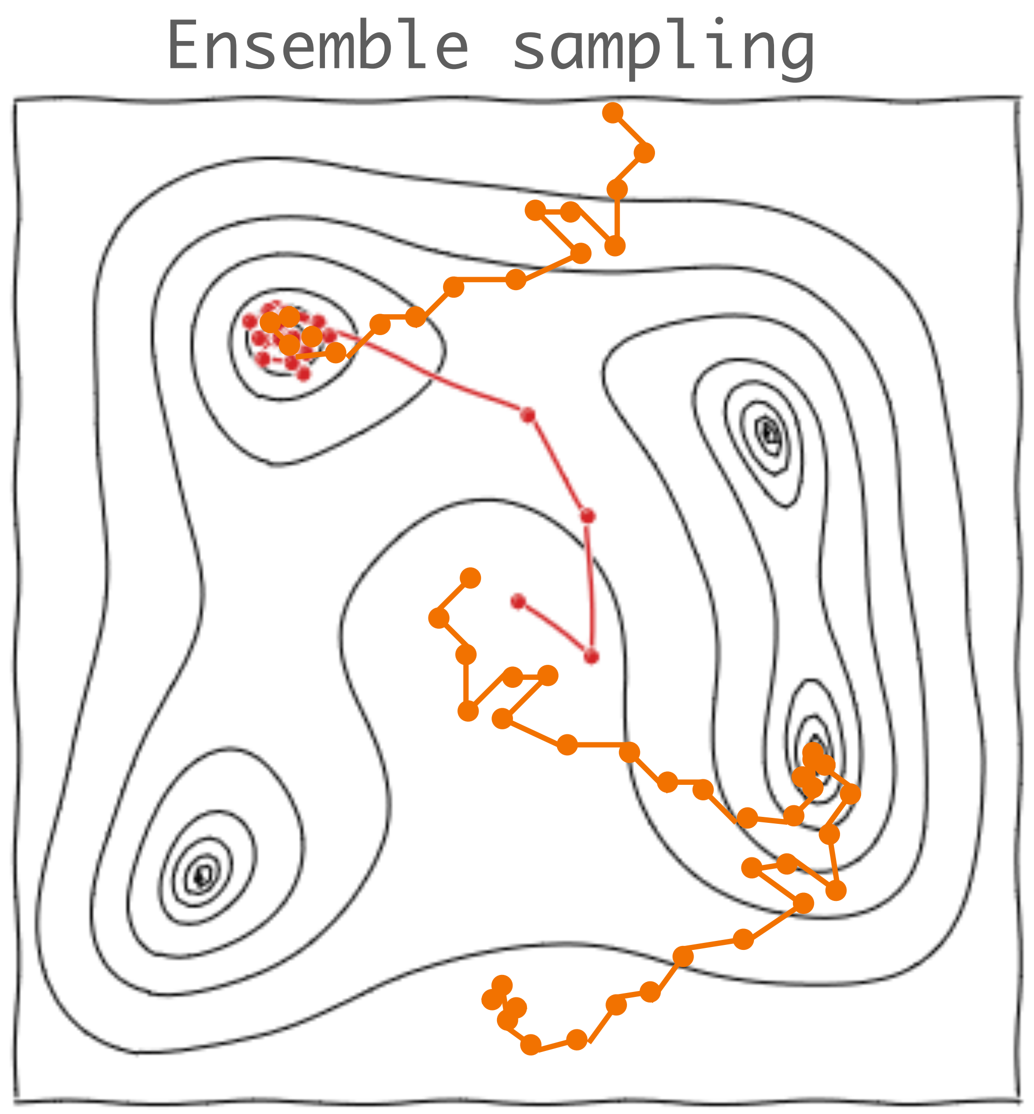
- **Markov-Chain Monte-Carlo (MCMC) with Metropolis-Hastings (M-H) algorithm:**
  - 1 point at the time probes the parameter space step by step
  - At each step, proposes a new point isotropically, based on the current point
  - Points are proportional to the target distribution (posterior probability of oscillation parameters)
  - Several chains can be ran in parallel and combined (after reaching convergence)



Ensemble sampling

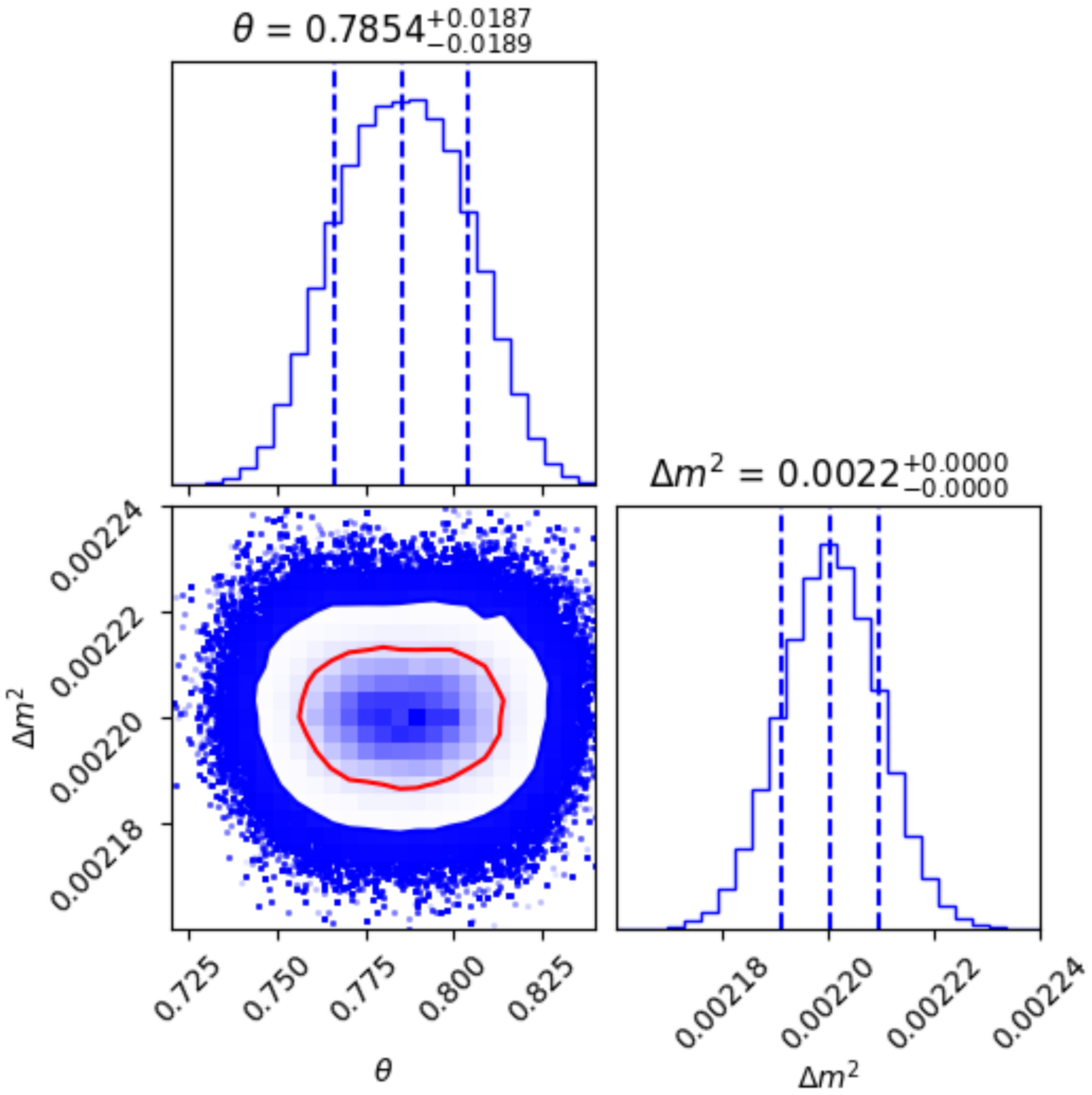
- **Ensemble sampling**
  - Similar to MCMC with M-H, but always contains several chains (« walkers »)
  - Chains exchange about their state: if one has reach convergence, the other propose points in the direction of convergence
  - Tested with the `emcee` package, includes several options with internal tuning (widely used in astro/cosmo)



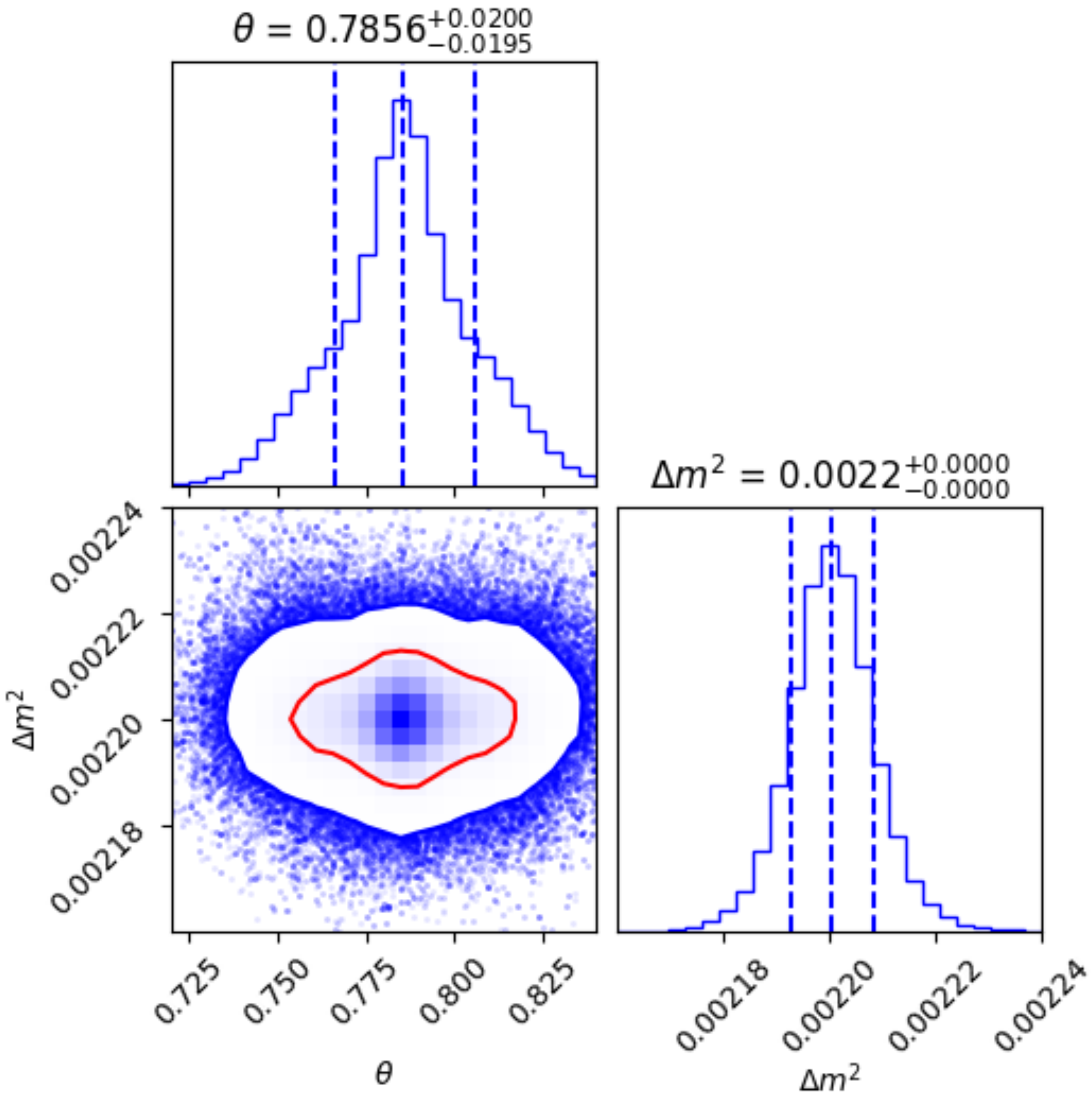


M-H vs emcee results

MCMC with M-H

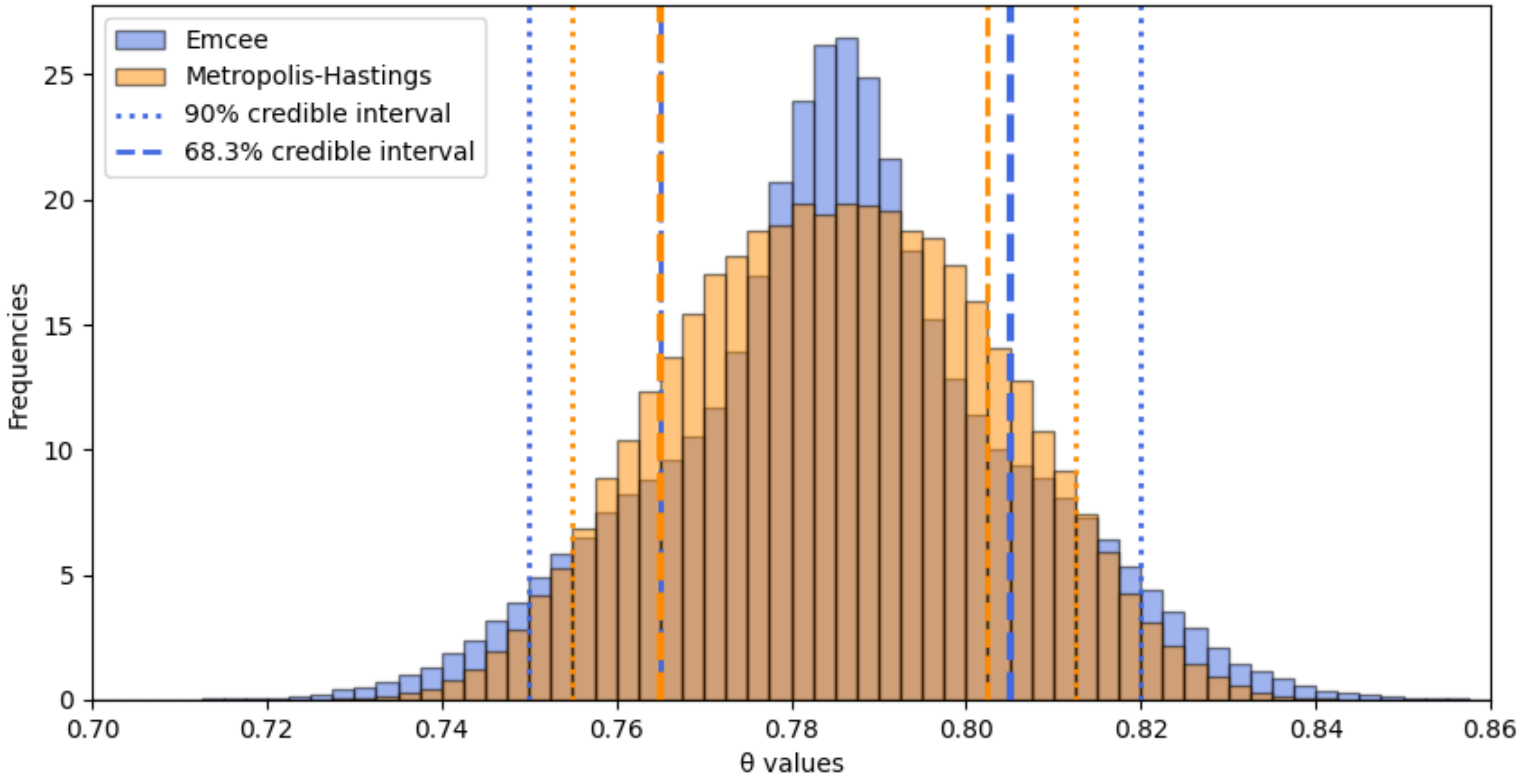


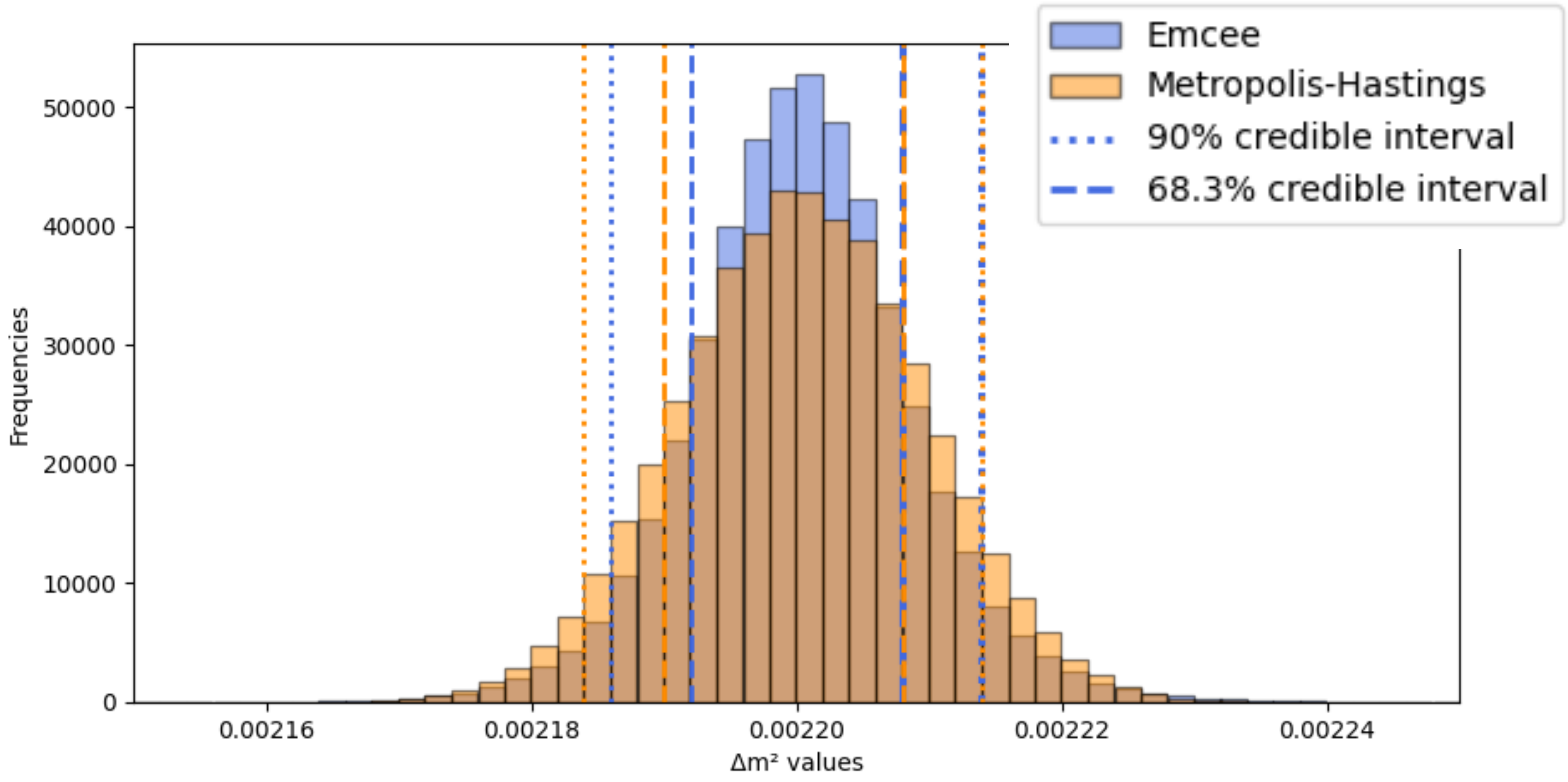
emcee



M-H vs emcee results

- **Posterior probability distribution comparisons**
  - Both samplers give similar results, centred on the correct oscillation parameter value
  - emcee distribution is slightly more peaked than M-H: leads to tighter credible intervals
  - Output differences to be investigated, but proof of principle is encouraging

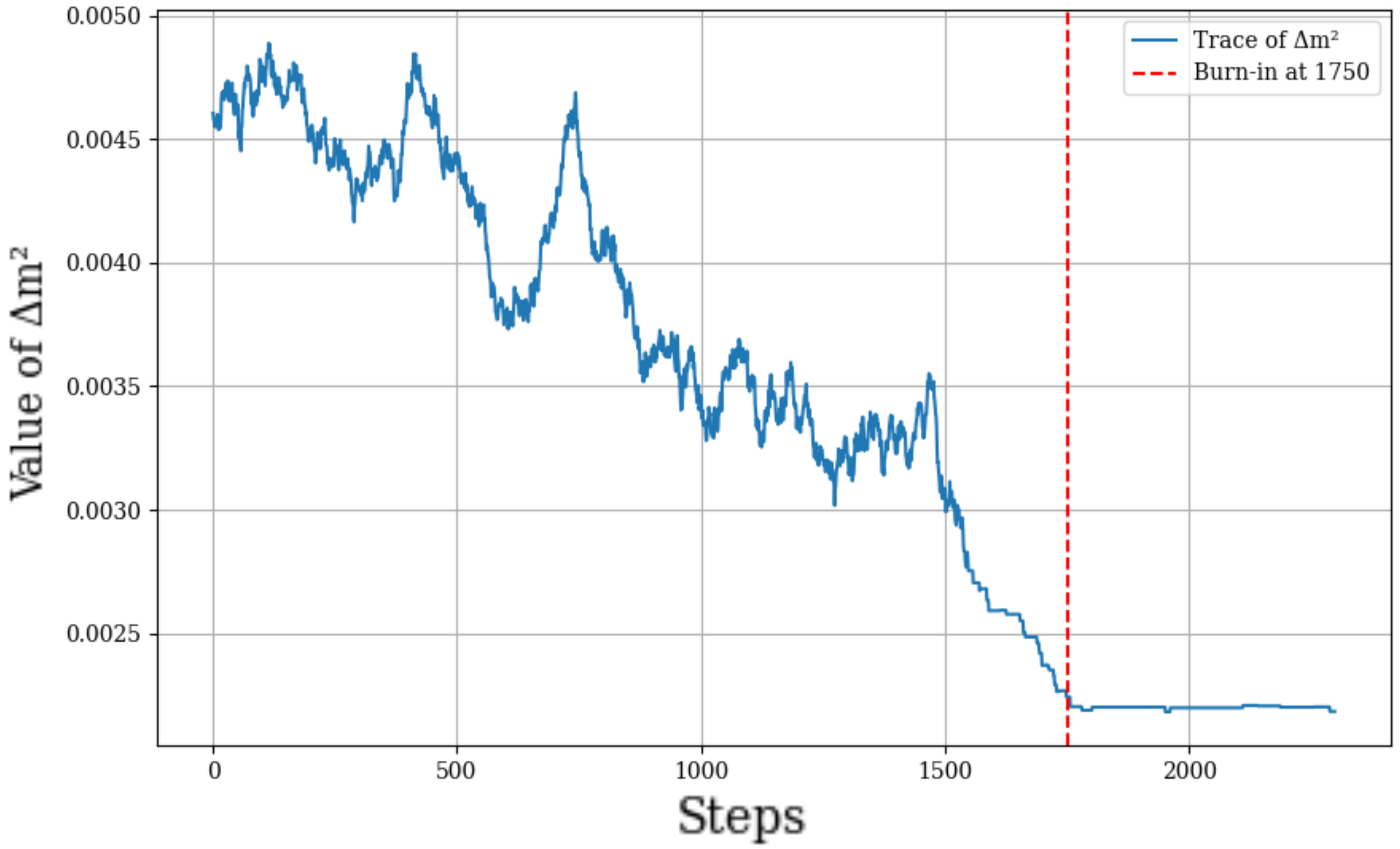




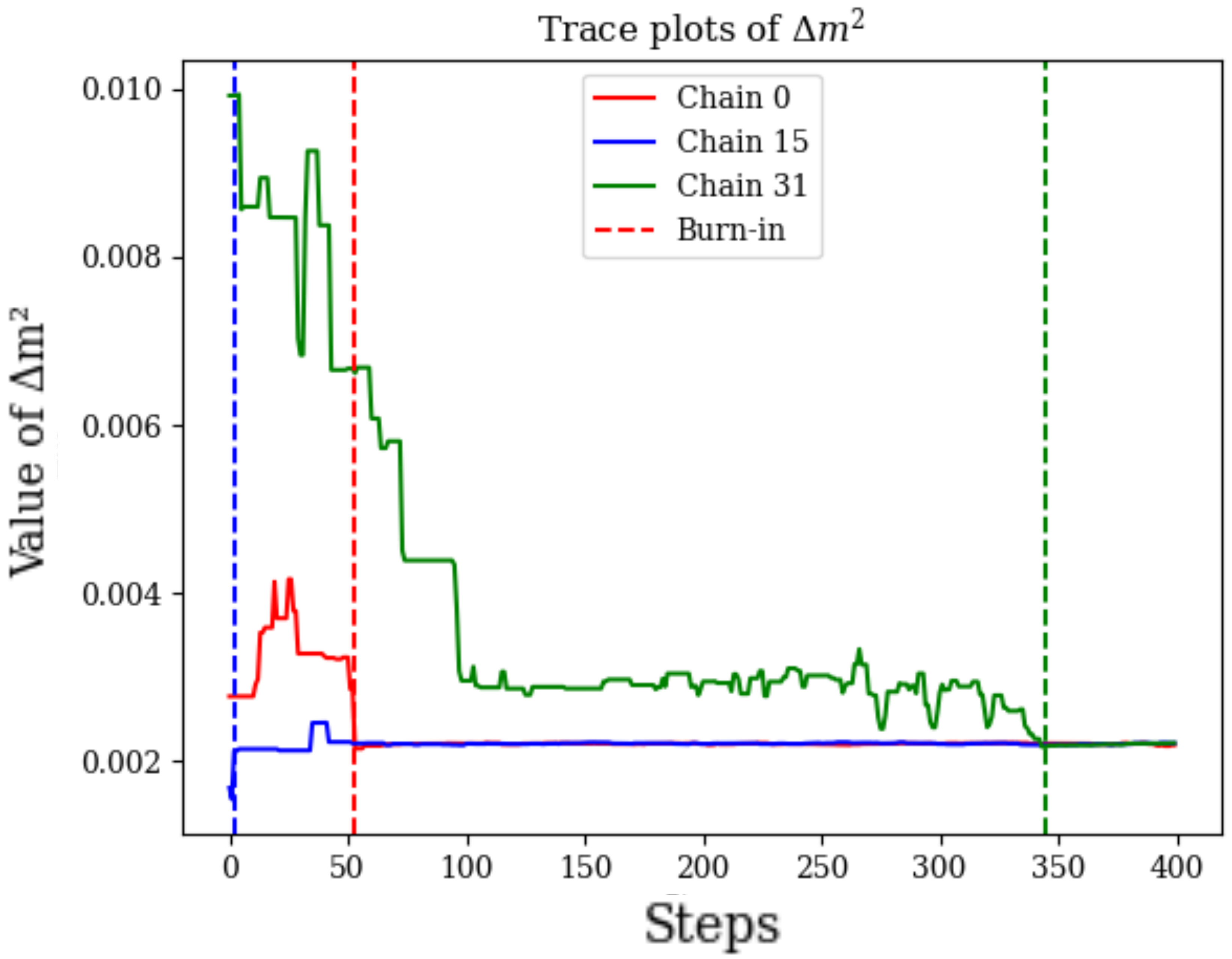
M-H vs emcee results

- **emcee was faster in this example**
  - MCMC with M-H: 10^7 steps, 6 hours on CPU (no parallelisation, tuning possible)
  - emcee: 32 walkers, 1.5 x10^4 steps each, 10 min on CPU

MCMC with M-H

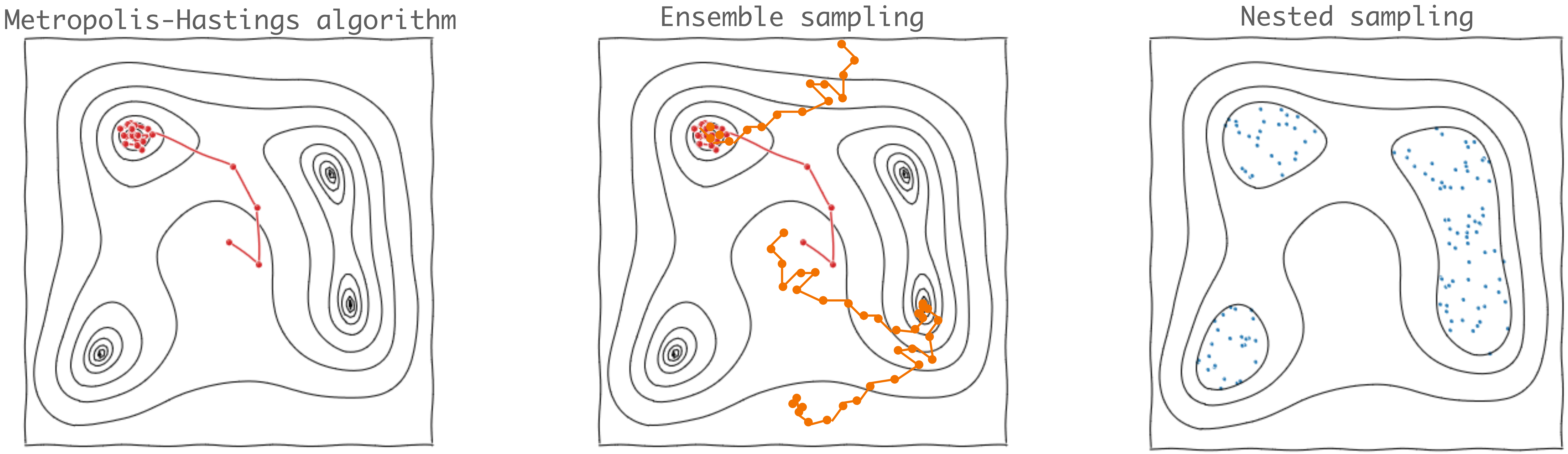


emcee



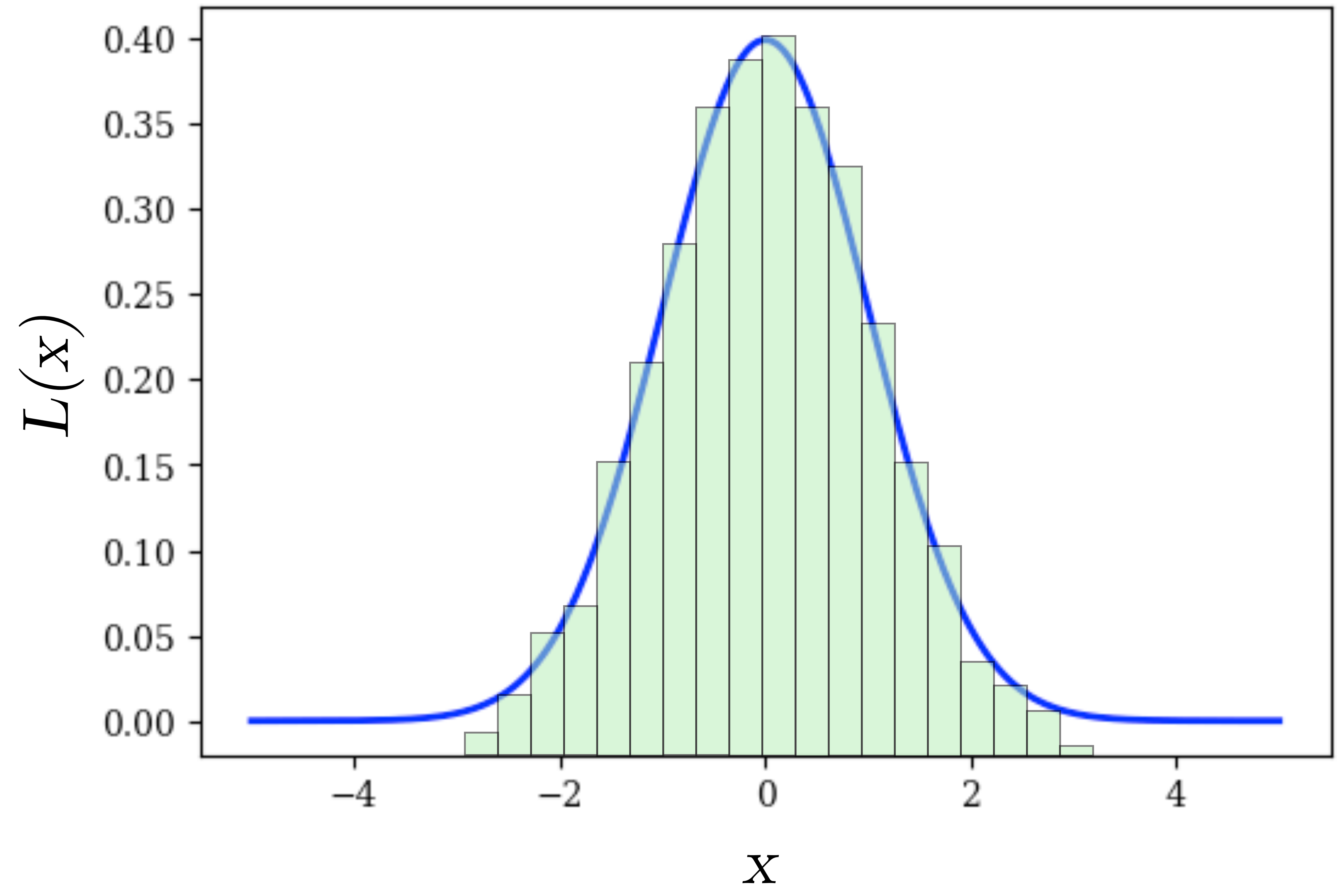
Nested sampling

- **Nested sampling:**
  - Set of live points distributed in the prior space
  - At each step, kill the point of lowest likelihood and propose a point of higher likelihood
  - Consist in scanning the likelihood from lower to higher values
  - Also good at escaping local minima and performing model comparison

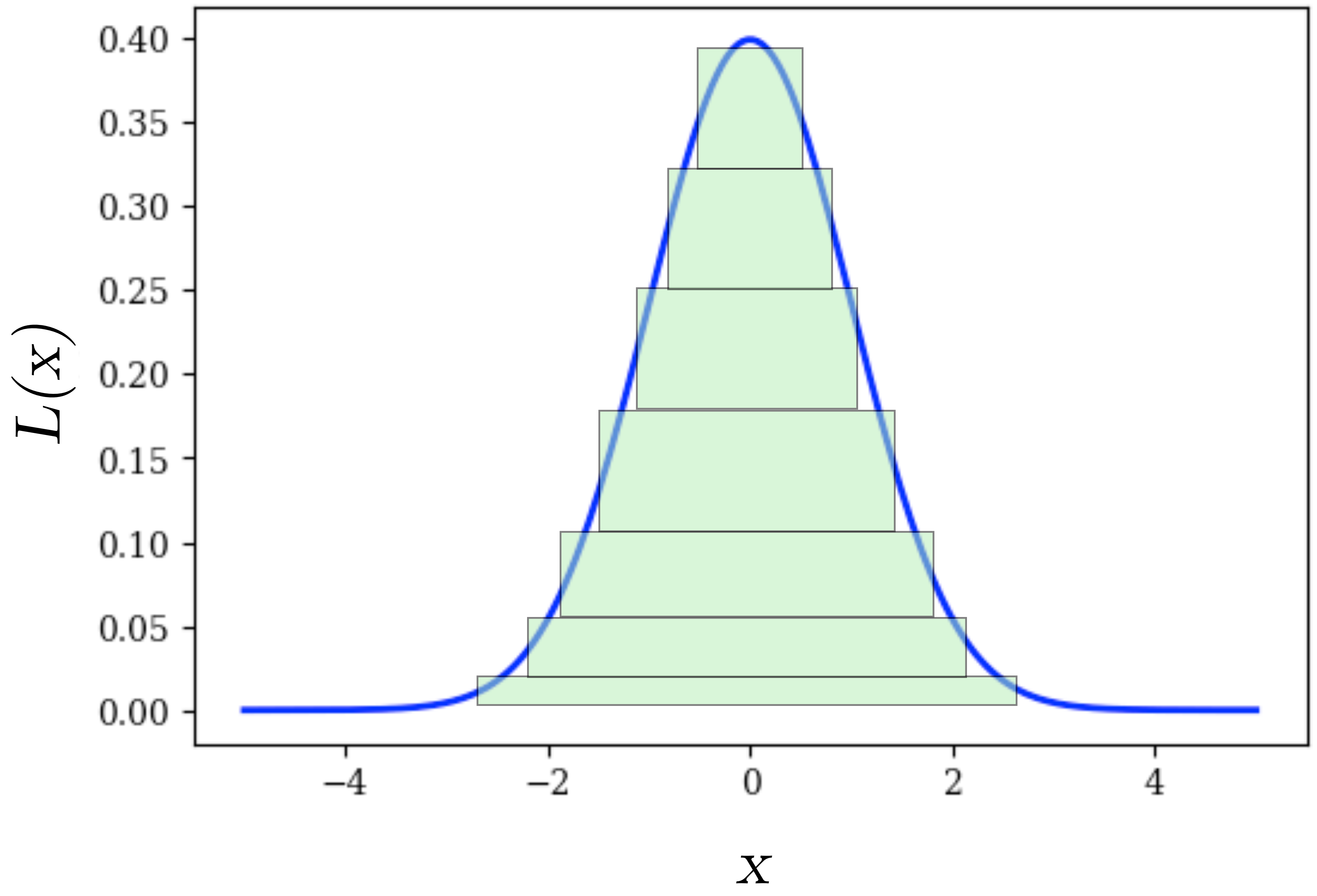


Nested sampling vs MCMC

- **Metropolis-Hastings algorithm**
  - designed to estimate the posterior probability
  - populate the samples distributions by filling vertical sections (the steps)

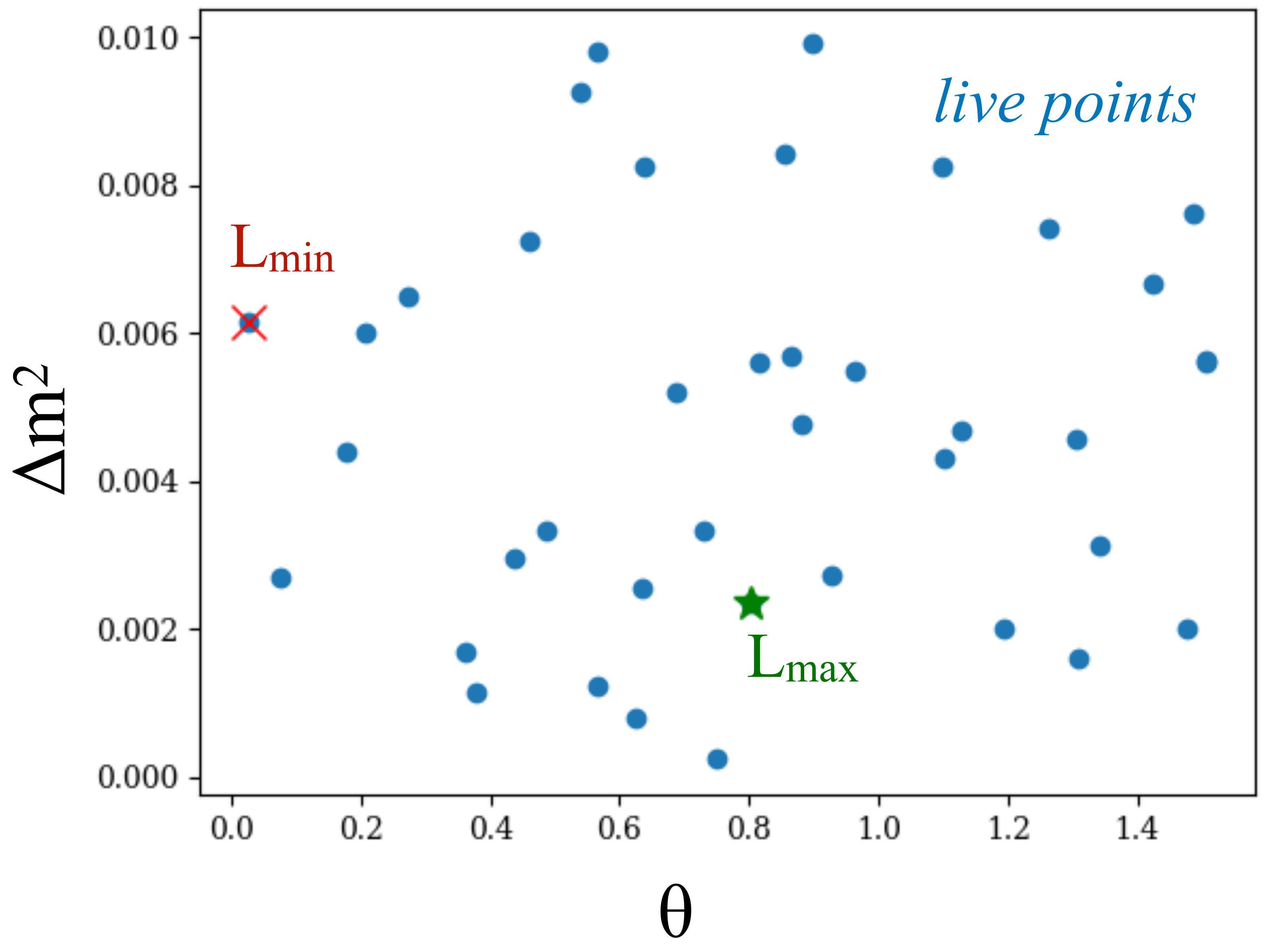


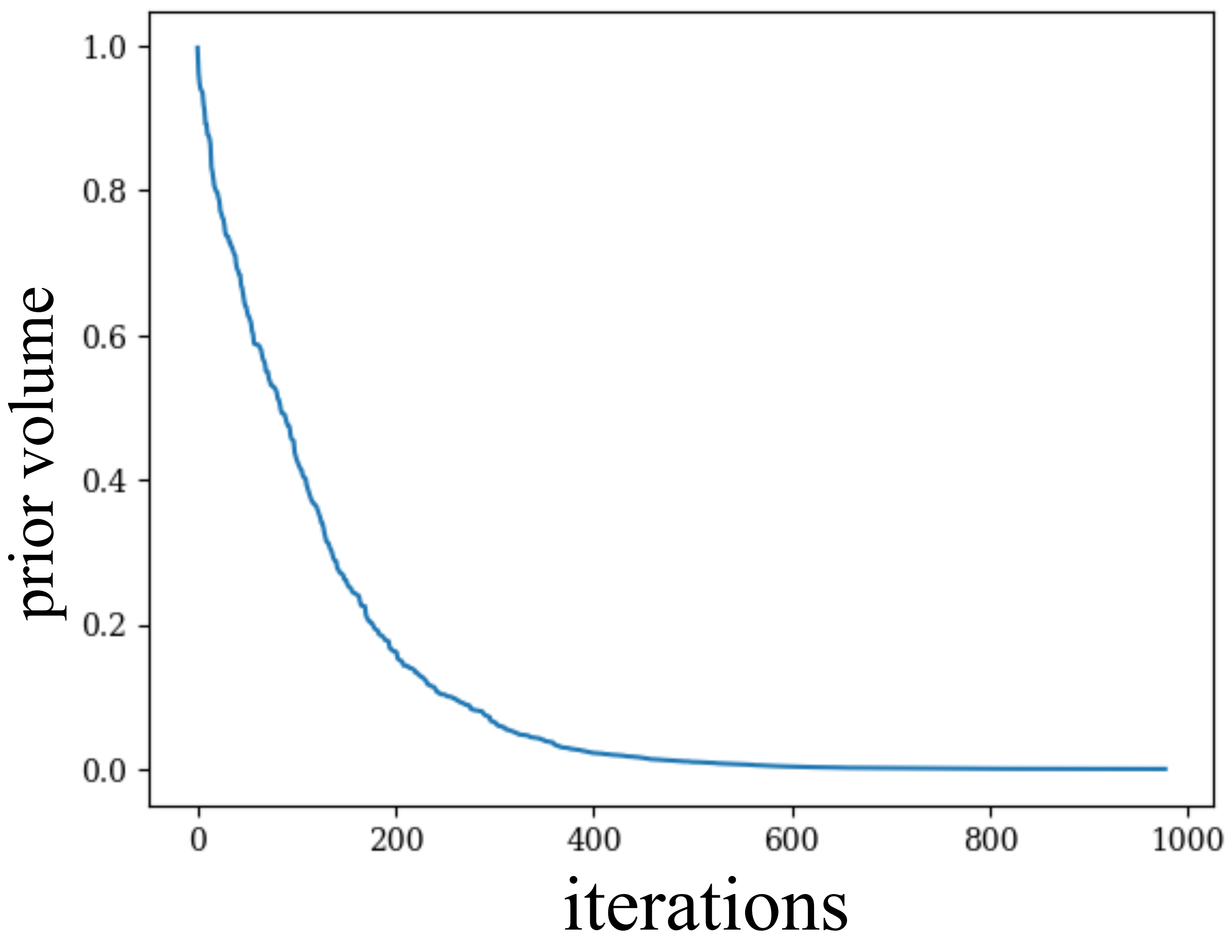
- **Nested sampling algorithm**
  - designed to estimate the evidence
  - populate the samples distributions by spanning horizontal sections of the likelihood

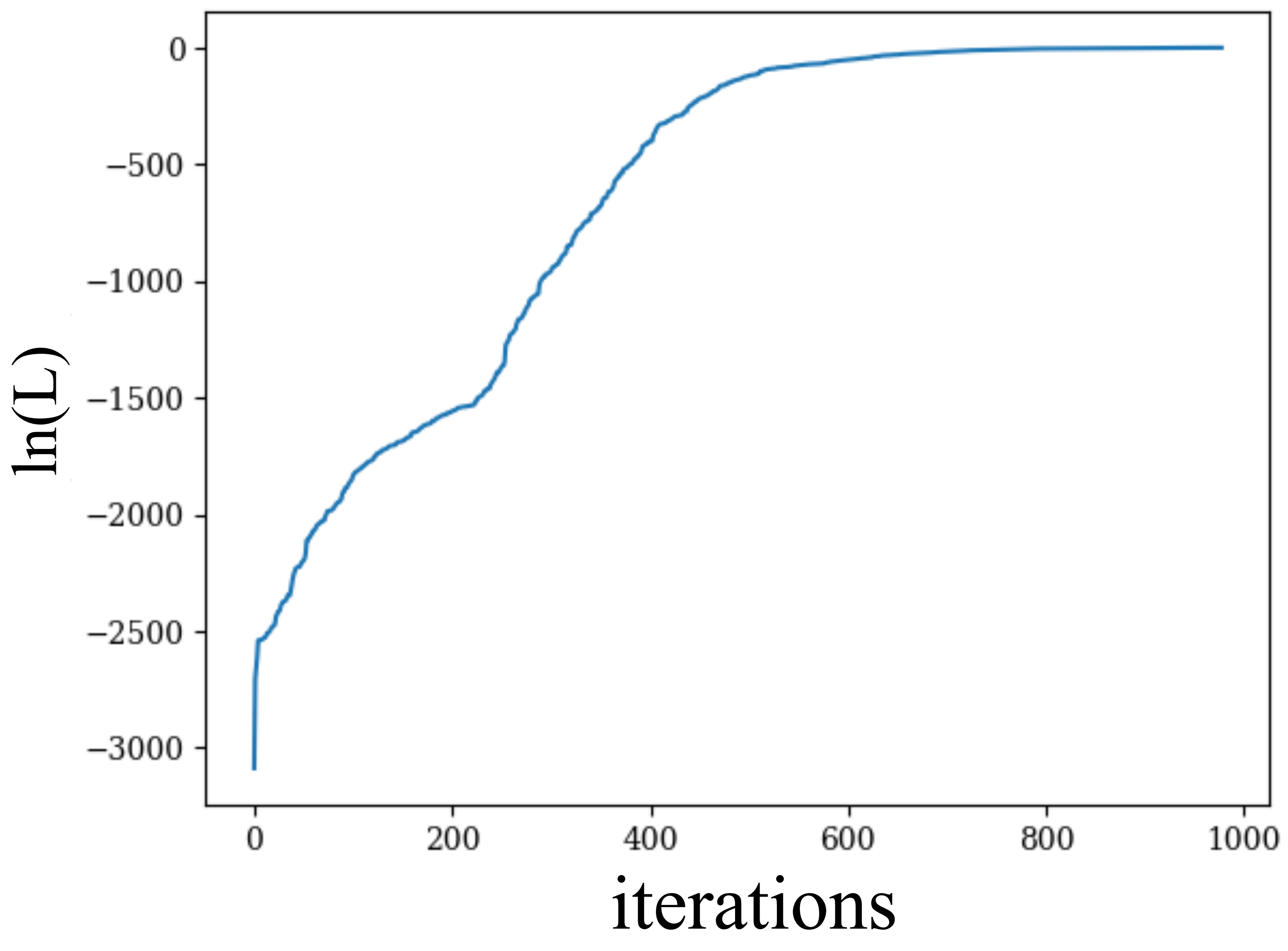


Nested sampling procedure

- **Hand-made implementation:**
  - Throw N=40 live points in the prior volume
  - Select point of lower likelihood and keep it apart
  - Propose new points with higher likelihood than thrown point
  - Repeat to scan likelihood profile from the bottom

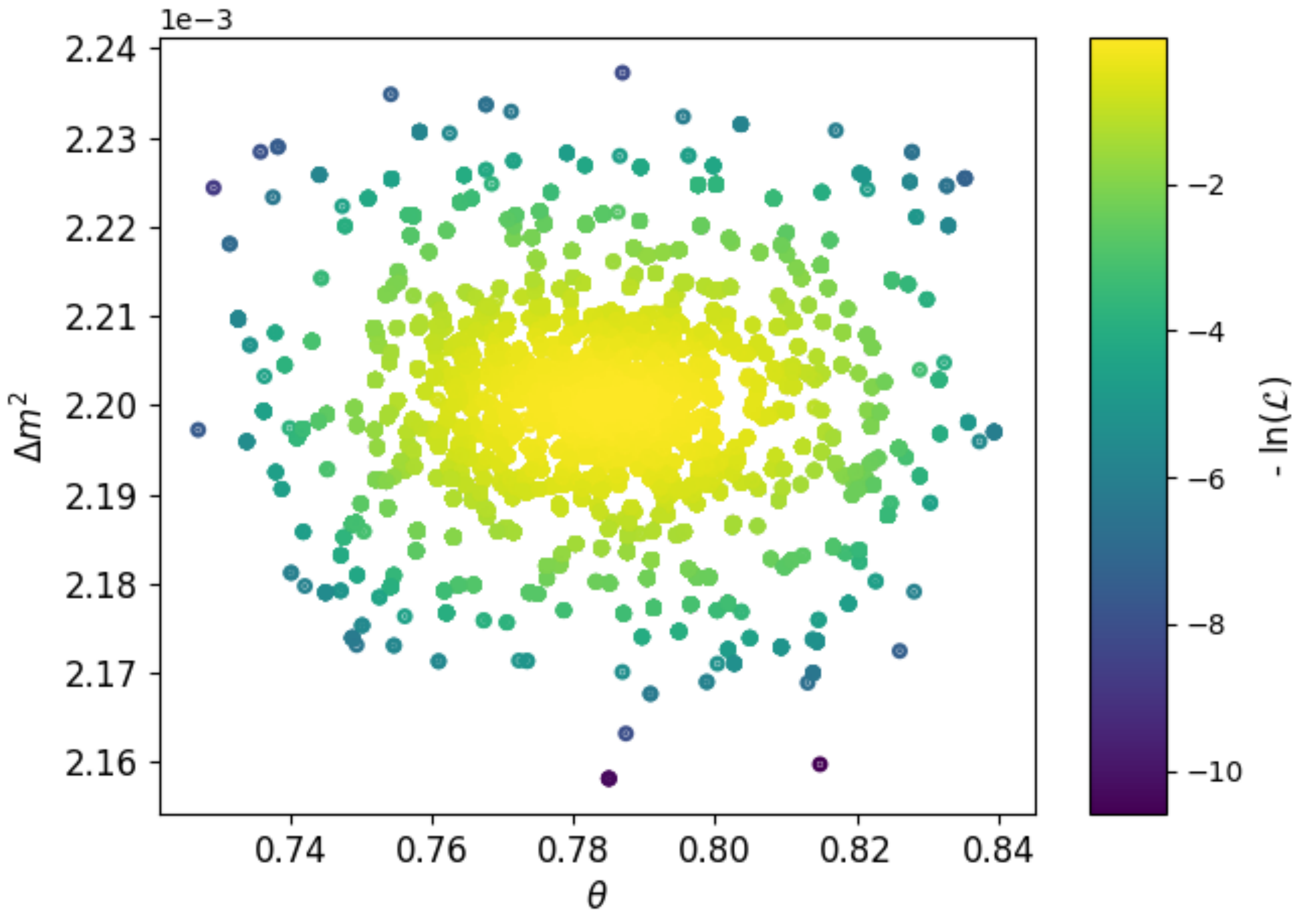


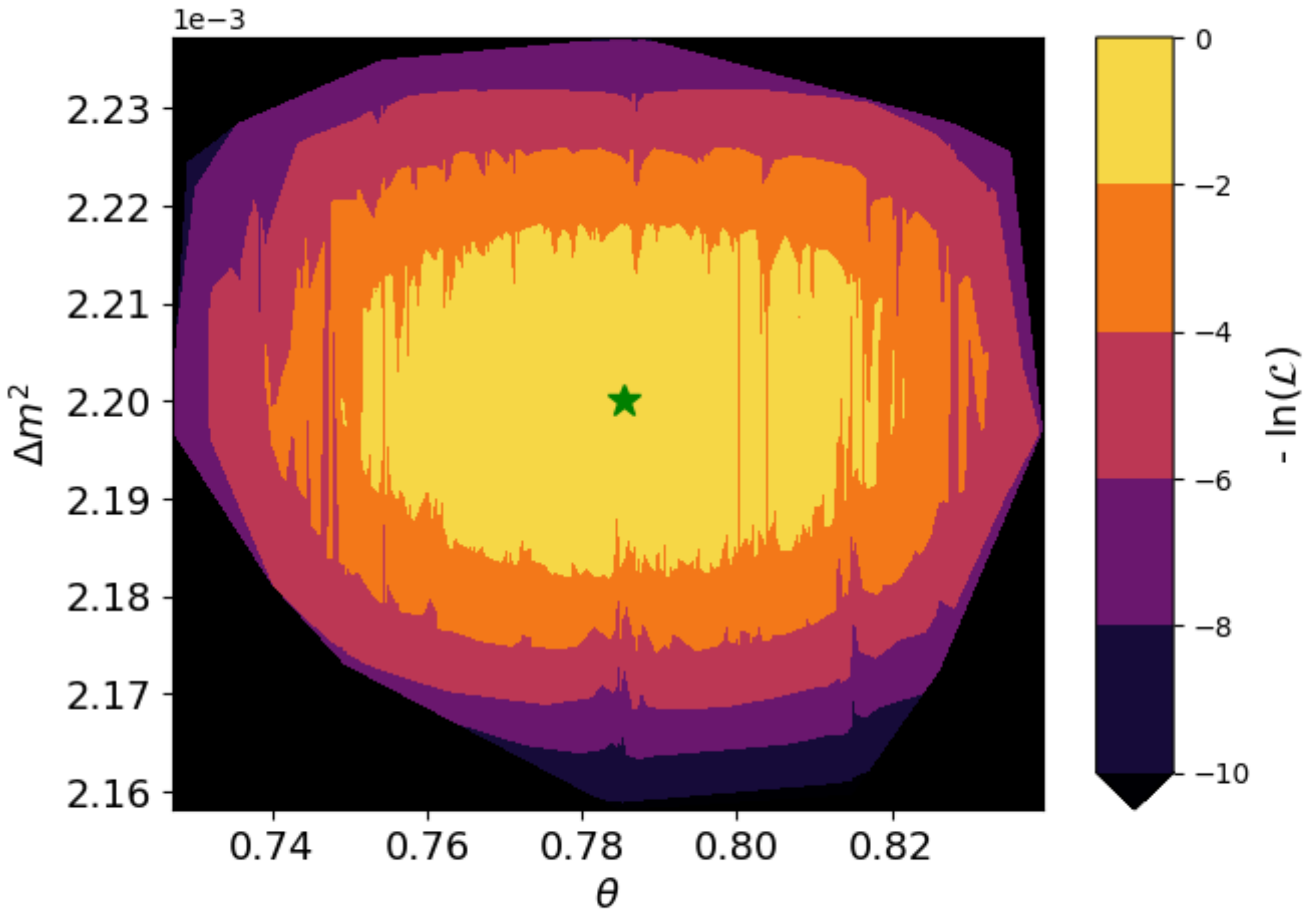




Nested sampling results

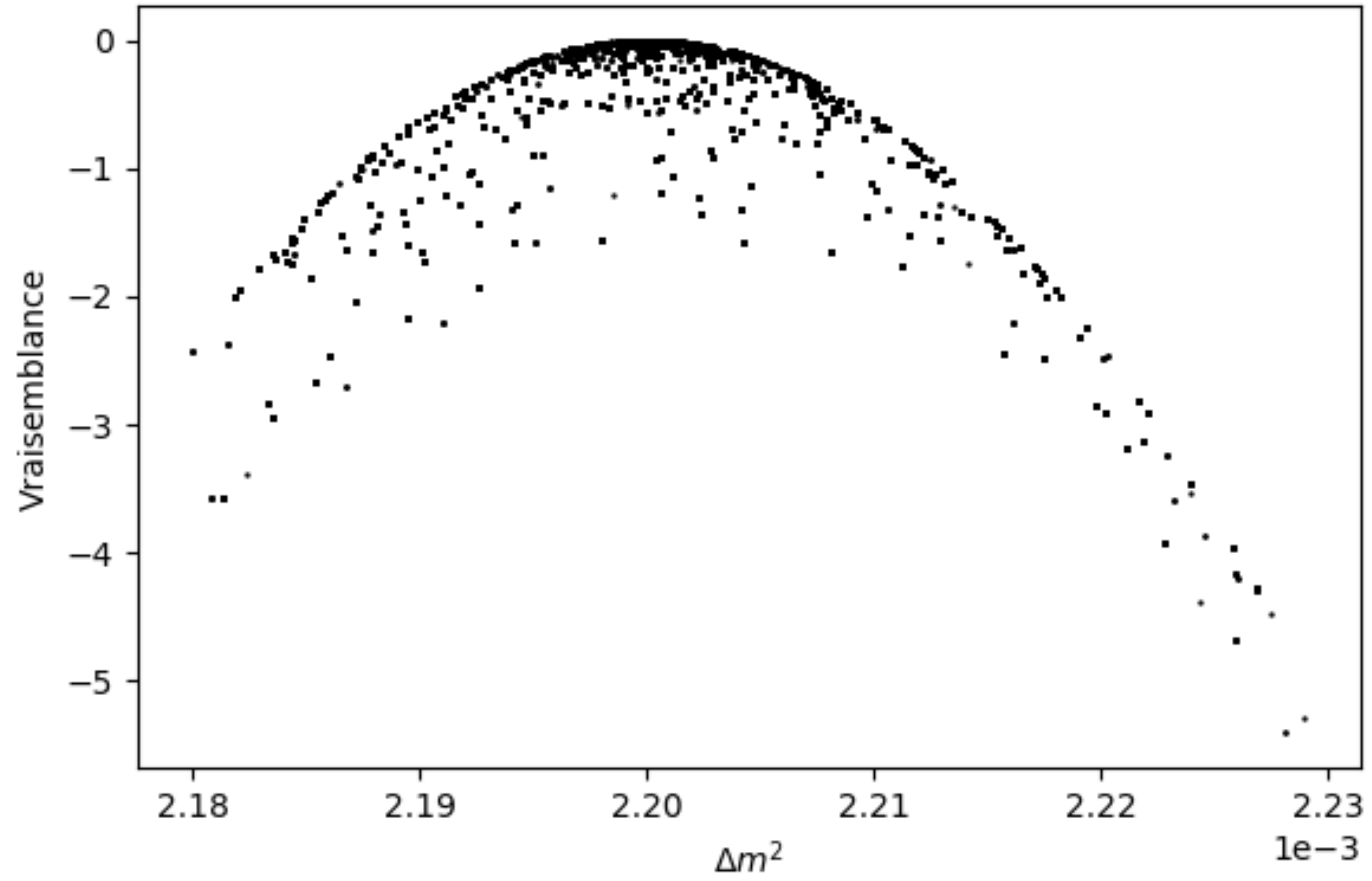
- **Output:**
  - Converge to correct value of oscillation parameters
  - Execution time: 30 min for 200 points, 20’000 iterations (CPU)
  - Note: python packages exist that would probably make this faster

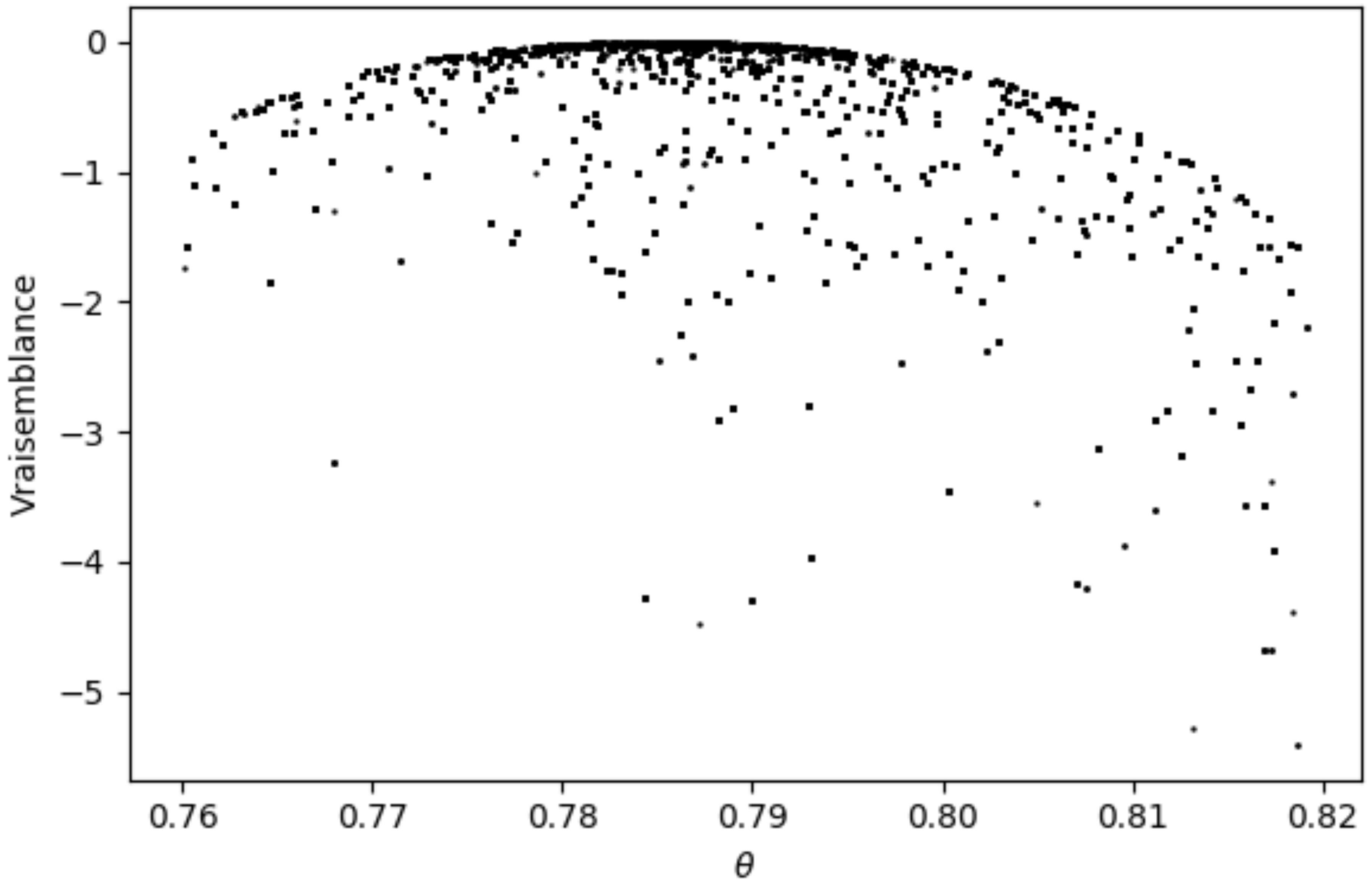




Nested sampling results

- **Interesting feature**
  - The procedure keeps more points at high likelihood
  - They must be interpolated to extract credible intervals
  - The high density can be exploited to extract accurate 5σ intervals





Conclusion

- **Accelerating neutrino oscillation parameter estimation**
  - Using other algorithms may ensure faster results
  - Existing packages have been optimised for speed and convergence
  - Demonstration on toy model, to be extended to more realistic cases (3 neutrino oscillation, systematical uncertainties)

- **This is just the sampling part**
  - Proposing new parameters is also time-consuming (oscillation probability computation, spline reading, event / bin reweighing)
  - The sampling can be plugged on NuSystematics (DUNE current systematics reweighing)
  - Can be also be tested with GUNDAM (experiment independent systematics propagation package)