DarkMachines

High dimensional sampling project

Martin White, Joaquin Vanschoren

Aim of the challenge

- Find awesome techniques for sampling high dimensional parameter spaces that we have never used in particle astrophysics (with help from our machine learning experts)

- Develop realistic physics case studies, and evaluate each technique (with the help of our physics experts)

- Prepare a detailed summary that compares the techniques and presents our results

What to discuss in this meeting

- First we need to describe the typical physics problems that we want to solve to our machine learning experts
- Then we need to identify which techniques we want to try, and which case studies to apply them to
- Then we need to agree on a software framework for running fair comparisons between the techniques

A typical particle astrophysics problem

We have a bunch of data from different experiments that might be sensitive to dark matter:

- colliders (LHC + previous)
- measurements of the magnetic moment of the muon
- electroweak precision tests
- dark matter direct detection experiments
- searches for antimatter in cosmic rays, nuclear cosmic ray ratios
- radio astronomy data
- effects of dark matter on reionisation, recombination and helioseismology
- relic density (CMB + other data)
- neutrino masses and mixings
- Indirect DM searches (e.g. FERMI-LAT, HESS, CTA, IceCube, etc)

A typical particle astrophysics problem

- We may have a particular theory of dark matter (e.g. a particular Lagrangian in particle physics)
- Which values of the parameters of that theory are preferred given the data?
- How probable or likely is the model relative to other models of dark matter?
- The likelihood of the model can be expressed as a composite likelihood assuming each set of measurements is independent:

$$\mathcal{L} = \mathcal{L}_{\rm collider}\mathcal{L}_{\rm DM}\mathcal{L}_{\rm flavor}\mathcal{L}_{\rm EWPO}\cdots$$

A typical particle astrophysics problem

$$\mathcal{L} = \mathcal{L}_{\text{collider}} \mathcal{L}_{\text{DM}} \mathcal{L}_{\text{flavor}} \mathcal{L}_{\text{EWPO}} \cdots$$

- We either want to map the shape of the multi-dimensional likelihood surface and use it to define confidence intervals (Frequentist), or use a prior and our likelihood to define a posterior, and map that (Bayesian)

- The likelihood is not known analytically, but can be mapped by sampling the function: for each parameter point, we can simulate the various experiments and compare the theoretical predictions to data to obtain a likelihood

Other problems

- These sorts of problems are ubiquitous in physics, e.g.: fitting parton density functions to experimental data to obtain the structure of the proton, extracting the neutrino sector parameters from accelerator and atmospheric data, extracting flavour physics parameters, …

- In each case, the parameters of the particle physics model are usually poorly constrained *a priori*, but there are additional nuisance parameters that are better-constrained (e.g. experimental systematics, mass measurements of SM particles, velocity of dark matter in the frame of the Earth, etc)

Slow likelihood calculations

- A particular feature of interesting problems is that the calculation of each likelihood might be very slow
- For the GAMBIT Large Hadron Collider observables, we managed to get our simulations to run in 5s, but this takes massive parallelisation
- PDF fits require a calculation that takes ~20s, and there are over 100 nuisance parameters to scan over
- Cosmological calculations that require simulating the recombination history of the universe might need minutes per point

Not all problems are equally challenging

- The posterior is usually unimodal in cosmological applications (thus Markov Chain Monte Carlo techniques have remained popular), or in PDF fits

- The posterior is multimodal in, e.g., global fits of supersymmetric models, with very thin regions of interest in some cases (due to special conditions being needed to reproduce the correct dark matter relic density)

- Composite Higgs theories represent the biggest challenges I have yet seen (horrible thin sheets in the parameter space, which require delicate cancellations between sectors of the theory to get the right SM Higgs mass and quark masses)

GAMBIT sampling

Eur. Phys. J. C manuscript No.
(will be inserted by the editor)

Comparison of statistical sampling methods with ScannerBit, the GAMBIT scanning module

The GAMBIT Scanner Workgroup: Gregory D. Martinez[1,a], James McKay[2,b], Ben Farmer[3,4,c], Pat Scott[2,d], Elinore Roebber[5], Antje Putze[6], Jan Conrad[3,4]

[1]Physics and Astronomy Department, University of California, Los Angeles, CA 90095, USA
[2]Department of Physics, Imperial College London, Blackett Laboratory, Prince Consort Road, London SW7 2AZ, UK
[3]Oskar Klein Centre for Cosmoparticle Physics, AlbaNova University Centre, SE-10691 Stockholm, Sweden
[4]Department of Physics, Stockholm University, SE-10691 Stockholm, Sweden
[5]Department of Physics, McGill University, 3600 rue University, Montréal, Québec H3A 2T8, Canada
[6]LAPTh, Université de Savoie, CNRS, 9 chemin de Bellevue B.P.110, F-74941 Annecy-le-Vieux, France

Received: date / Accepted: date

Abstract We introduce ScannerBit, the statistics and sampling module of the public, open-source global fitting framework GAMBIT. ScannerBit provides a standardised interface to different sampling algorithms, enabling the use and comparison of multiple computational methods for inferring profile likelihoods, Bayesian posteriors, and other statistical quantities. The current version offers random, grid, raster, nested sampling, differential evolu- ten or more dimensions, Diver substantially outperforms the other three samplers on all metrics.

Contents

ScannerBit algorithms

- ScannerBit contains custom code or interfaces for the following methods:

  Random

  Grid

  Markov Chain Monte Carlo (MCMC)

  Ensemble Monte Carlo

  Nested Sampling

  Differential evolution

Discussion (1)

1) What other physics problems do we want to solve that have been missed?
2) Which techniques are most scalable to O(100) of parameters? (or O(1000)?)
3) What techniques can be used to mitigate the effects of slow likelihood calculations (e.g. they require less samples, or they can approximate the likelihood)
4) What changes if we have lots of better-constrained parameters rather than lots of poorly constrained parameters?
5) Are there any techniques that can intelligently change the basis of the parameters we are scanning to make the problem simpler?
6) Anything else?

Discussion (2)

- Possible case studies could include:

1) Toy, analytic functions that increase in their awkwardness (e.g. *n*-D Gaussian, lots of *n*-D Gaussians, functions that have thin sheet-like behaviour in *n*-D, functions with very small regions of interest in addition to large regions of interest)

2) A realistic physics likelihood, with actual experimental likelihoods (e.g. Higgs portal likelihood in GAMBIT scanning paper)

3) Anything else?

Discussion (3)

- Possible software frameworks are:

1) We write our own interface to existing scanning implementations and physics likelihoods (plus code up any new scanners that we need)

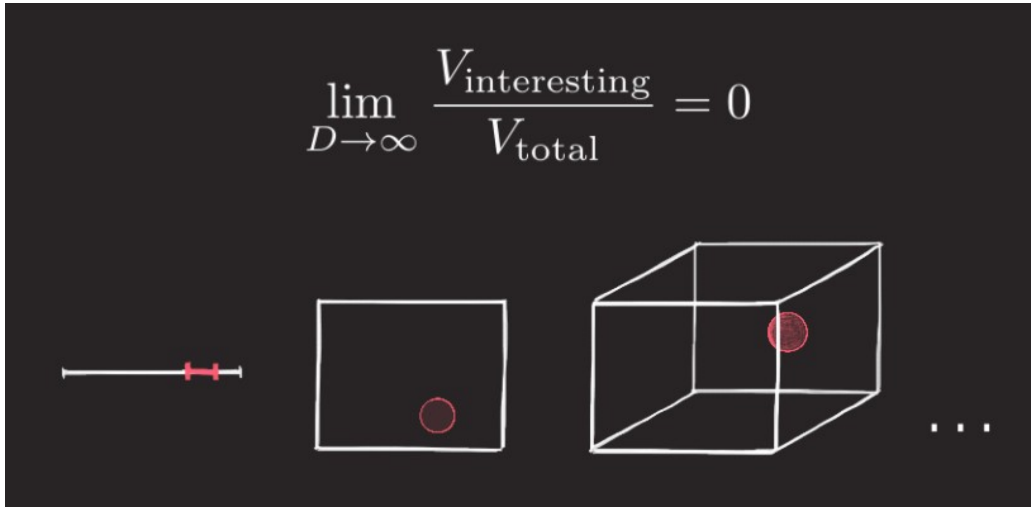
2) We use GAMBIT which gives us most of that for free (at the cost of some overhead in learning how to add samplers to it)

3) Anything else?

Backup

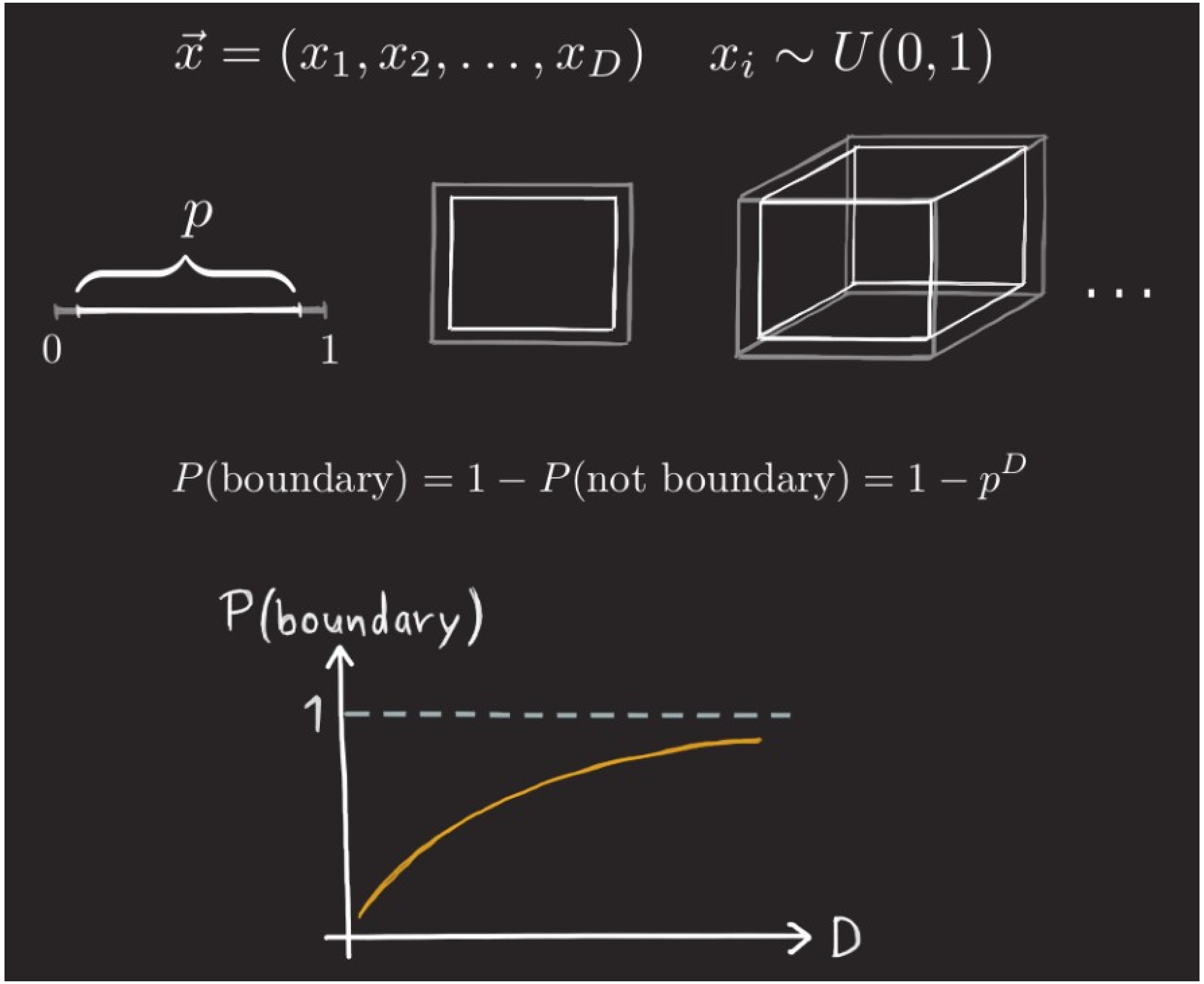
Random and Grid scanners

- Random: just sample points randomly from the space within some box specified as a prior range on each parameter
- Grid: Scan along each axis (within some prior range)
- Not useful for serious applications: random sampling leads to biased inferences when applied to almost all problems
- Random and grid scanning both scale terribly with the number of dimensions in a problem



Source: Anders Kvellestad

Random and Grid scanners



Source: Anders Kvellestad

Markov Chain Monte Carlo (MCMC) methods

- MCMC methods have been used for decades in cosmology and particle physics problems
- A popular approach is the *Metropolis-Hastings algorithm*:

1) Start at a randomly drawn initial point $\boldsymbol{\theta}_i$

2) Select another point $\boldsymbol{\theta}_{\text{trial}}$ at random using a *proposal* function $q(\boldsymbol{\theta}_{\text{trial}} | \boldsymbol{\theta}_i)$

3) The candidate point is accepted with the probability (p($\boldsymbol{\theta}$) = likelihood for flat prior on $\boldsymbol{\theta}$)

$$a(\boldsymbol{\theta}_{\text{trial}}|\boldsymbol{\theta}_i) = \min\left(1, \frac{p(\boldsymbol{\theta}_{\text{trial}})}{p(\boldsymbol{\theta}_i)} \frac{q(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{\text{trial}})}{q(\boldsymbol{\theta}_{\text{trial}}|\boldsymbol{\theta}_i)}\right)$$

4) Set $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{\text{trial}}$ if $\boldsymbol{\theta}_{\text{trial}}$ is accepted, else retain $\boldsymbol{\theta}_i$, then repeat procedure

Markov Chain Monte Carlo (MCMC) methods

- These points form a *Markov chain*, which spends time in the parameter space in proportion to the target posterior PDF of the parameters (given the supplied likelihood)
- For sufficiently long chains, one obtains independent samples from the target distribution p($\boldsymbol{\theta}$)
- To optimise efficiency, the proposal distribution *q* should match the (*a priori* unknown) true distribution
- GAMBIT includes an interface to the GreAT MCMC scanner that uses a multivariate Gaussian for *q*
- GreAT runs multiple chains, covariance matrix for chains is obtained from previous terminated chains (after thinning and removal of “burn-in” points)

Ensemble MCMC

- Standard MCMC is bad for high dimensional problems and/or multi-modal target functions
- Ensemble MCMC: run concurrent chains, each chain is individually advanced by contructing q from set of all points sampled by all chains
- GAMBIT includes the T-Walk ensemble MCMC method
- See ScannerBit paper for full details of how chains are advanced (depends on whether you are running the serial or parallelised version)

Nested sampling

- Nested sampling has been very popular in recent years, with many applications in particle physics, astronomy and cosmology
- It is much better at handling multimodal target functions than MCMC methods
- An efficient implementation is available in the public Multinest package, which GAMBIT makes use of

Nested sampling: a quick review of Bayesian inference

- Given a set of parameters **Θ** in a model *H*, plus some data **D**, Bayes' theorem gives:

likelihood

Posterior probability distribution

$$\Pr(\mathbf{\Theta}|\mathbf{D}, H) = \frac{\Pr(\mathbf{D}|\,\mathbf{\Theta}, H)\Pr(\mathbf{\Theta}|H)}{\Pr(\mathbf{D}|H)}$$

prior

- Denominator is a normalisation factor called the “Bayesian evidence”

$$\mathcal{Z} = \int \mathcal{L}(\mathbf{\Theta})\pi(\mathbf{\Theta})d^D\mathbf{\Theta},$$

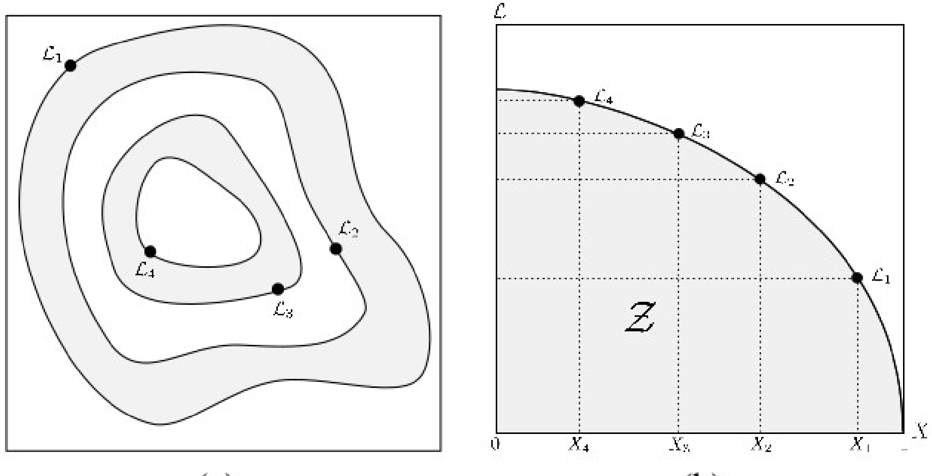
- MCMC algorithms ignore *Z* (they give samples from the unnormalised posterior)

Nested sampling

- Nested sampling instead calculates *Z* directly by Monte Carlo integration
- Clever trick: define prior volume $dX = \pi(\Theta)d^D\Theta$ $\quad X(\lambda) = \int_{\mathcal{L}(\Theta)>\lambda} \pi(\Theta)d^D\Theta$
- Can then write evidence integral as:

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

- Monotonically decreasing function of X
  - Draw *N* "live points" from prior, at each iteration replace the lowest likelihood samples with higher likelihood samples, repeat until prior volume has been traversed



(a) (b)

Differential evolution: Diver

- Optimisation algorithm, good for multimodal posteriors in high dimensional spaces
- A simple explanation is as follows:

1) Start with a random selection of points in the parameter space (called “vectors”)

2) *Mutate* vectors by e.g. picking three random vectors and making: (**V** = “donor vector”)

$$\mathbf{V}_i = \mathbf{X}_{r1} + F(\mathbf{X}_{r2} - \mathbf{X}_{r3}).$$

3) *Crossover* the donor vectors and original vectors by making *trial vectors* **U** that have a random selection of components from the original vectors and the donor vectors

4) *Select* the vectors by computing the likelihood for the original vectors and their associated trial vectors, and choosing the highest likelihood vector for the next generation

- See ScannerBit paper for full details of GAMBIT implementation

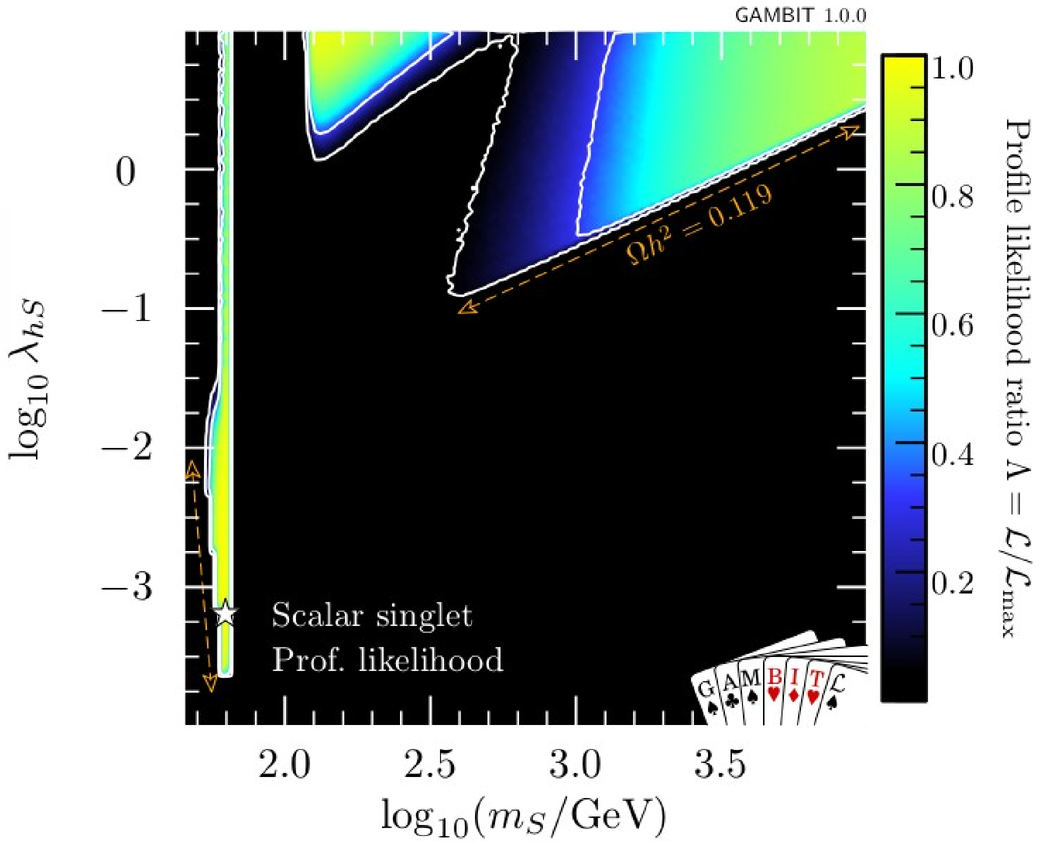
Scanner comparisons

- GAMBIT allows the scanner to be swapped by changing one line in a yaml file
- This offers a unique test bed for comparison of scanning algorithms
- Have compared algorithms on a non-trivial physics example: scalar singlet DM

$$\mathcal{L} = \frac{1}{2}\mu_S^2 S^2 + \frac{1}{2}\lambda_{hS} S^2 |H|^2 + \frac{1}{4}\lambda_S S^4 + \frac{1}{2}\partial_\mu S \partial^\mu S$$

- Constraints from direct and indirect DM detection experiments, LHC Higgs invisible width searches, relic density upper bound plus theoretical upper bound on the Higgs-singlet coupling

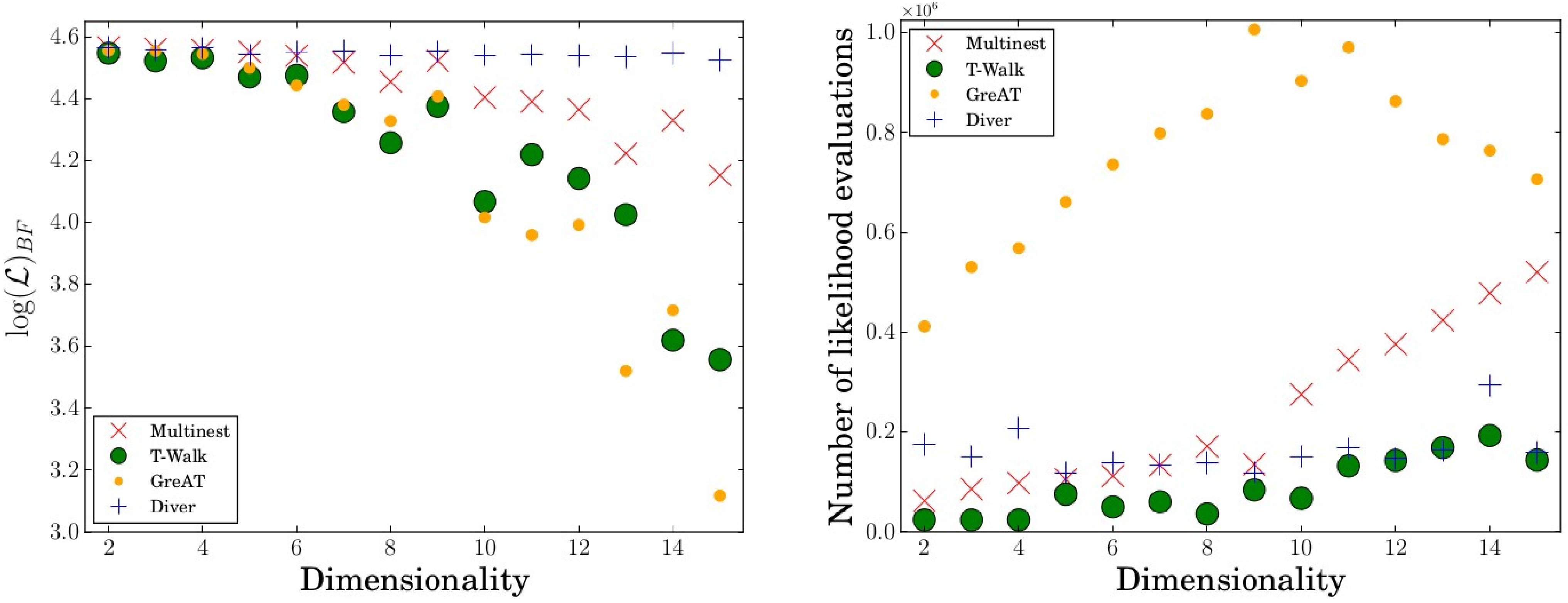
Singlet DM



$$\mathcal{L} = \frac{1}{2}\mu_S^2 S^2 + \frac{1}{2}\lambda_{hS} S^2|H|^2 + \frac{1}{4}\lambda_S S^4 + \frac{1}{2}\partial_\mu S \partial^\mu S$$

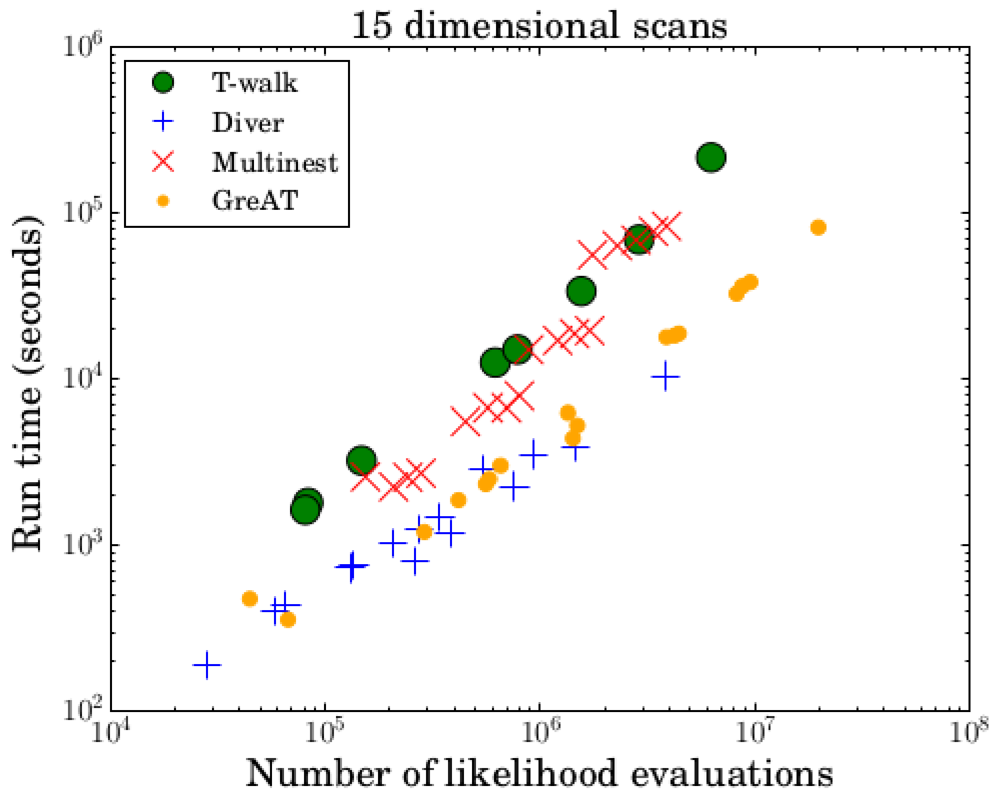
Parameter		Values
Scalar pole mass	m_S	$[45, 10^4]$ GeV
Higgs portal coupling	λ_{hS}	$[10^{-4}, 10]$
Varied in 7 and 15-dimensional scans		
Electromagnetic coupling	$1/\alpha^{\overline{MS}}(m_Z)$	127.940(42)
Strong coupling	$\alpha_s^{\overline{MS}}(m_Z)$	0.1185(18)
Top pole mass	m_t	173.34(2.28) GeV
Higgs pole mass	m_h	125.7(1.6) GeV
Local dark matter density	ρ_0	$0.4^{+0.4}_{-0.2}$ GeV cm^{-3}
Varied in 15-dimensional scans		
Nuclear matrix el. (strange)	σ_s	43(24) MeV
Nuclear matrix el. (up + down)	σ_l	58(27) MeV
Fermi coupling × 10^5	$G_{F,5}$	1.1663787(18)
Down quark mass	$m_d^{\overline{MS}}(2\,\mathrm{GeV})$	4.80(96) MeV
Up quark mass	$m_u^{\overline{MS}}(2\,\mathrm{GeV})$	2.30(46) MeV
Strange quark mass	$m_s^{\overline{MS}}(2\,\mathrm{GeV})$	95(15) MeV
Charm quark mass	$m_c^{\overline{MS}}(m_c)$	1.275(75) GeV
Bottom quark mass	$m_b^{\overline{MS}}(m_b)$	4.18(9) GeV

Scanner performance vs number of dimensions



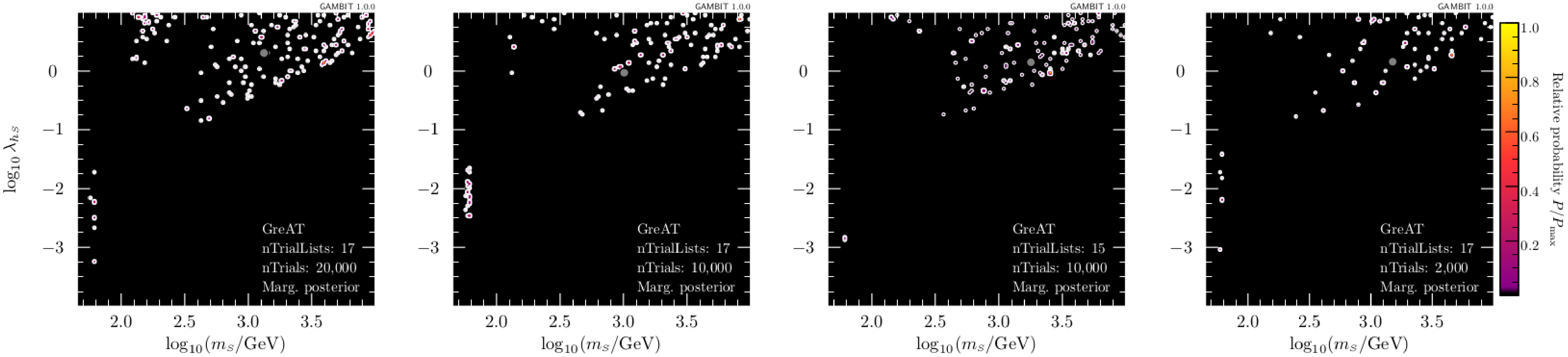
Diver: NP $= 20\,000$, convthresh $= 10^{-3}$
MultiNest: nlive $= 20\,000$, tol $= 10^{-3}$
T-Walk: chain_number $=$ number of MPI processes $+$ $N_{\rm dim} + 1$, tol $=$ sqrtR $- 1 = 0.05$
GreAT: nTrials $= 2000$, nTrialsList $= N_{\rm dim} + 1$

Real time vs number of likelihood evaluations



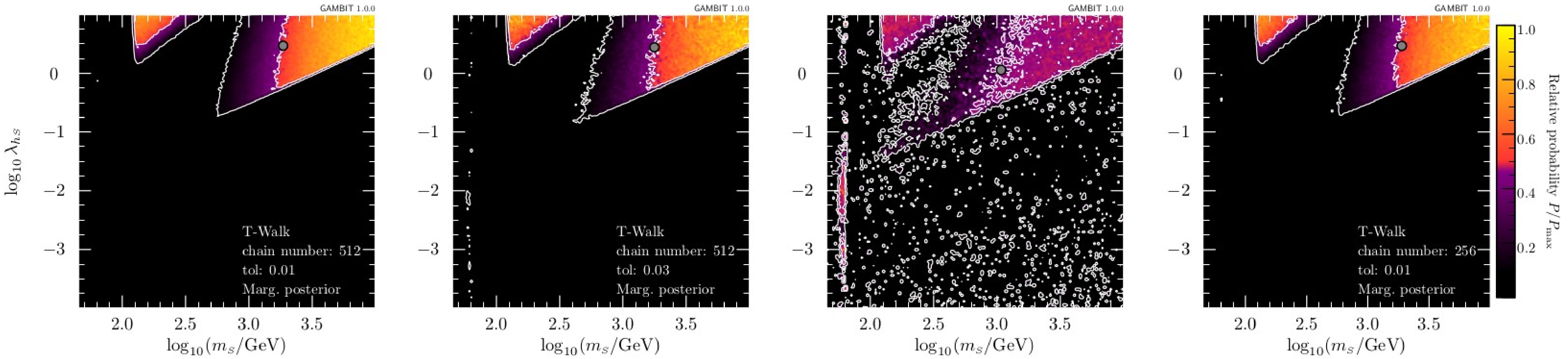
- T-Walk and Multinest are less efficient (per likelihood evaluation) than GreAT and Diver for large dimensional problems
- There are several reasons (e.g. ellipsoidal decomposition in Multinest, chain advancement calculations in T-Walk, MPI bottlenecks, etc)

Posterior mapping: 15D scan using GreAT



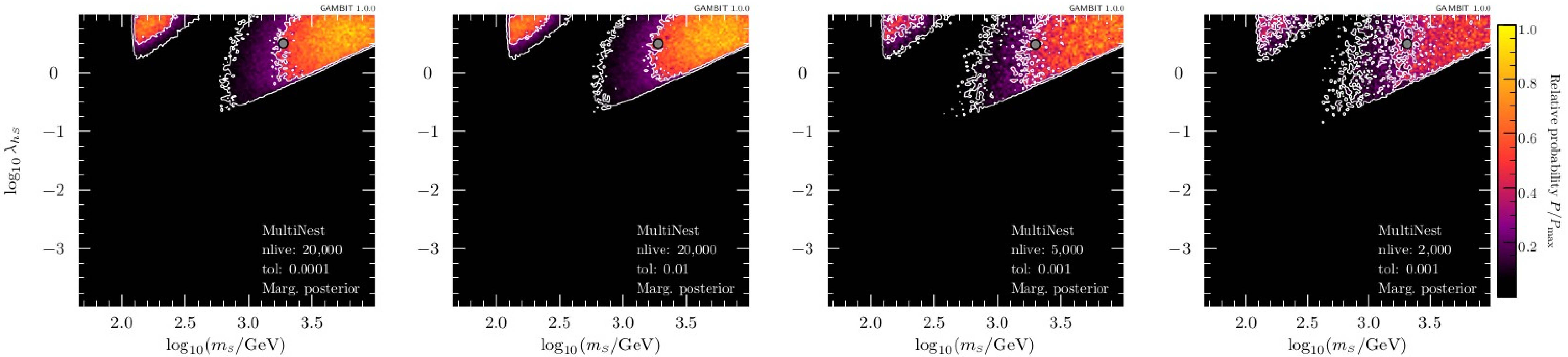
- Yikes! Validates assertion that MCMC algorithms do not cope well with multimodal posteriors?

Posterior mapping: 15D scan using T-Walk



- The best scan here was the best posterior obtained, taking 9h in total
- Poorly converged scans find all modes, but don't get relative weight correct (and don't map the posterior smoothly)

Posterior mapping: 15D scan using Multinest



- Scans with too few live points or too high a tolerance do not find all modes
- The best scan here took > 21h, and is not as smooth as the T-Walk results
- Multinest also erroneously smooths sharp features due to its ellipsoidal sampling method