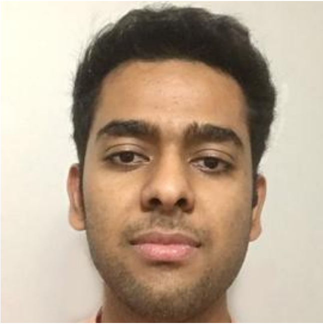
Efficient model selection with Bayesian optimisation

Jan Hamann

based on JCAP 03 (2022) 03, 036 [arXiv:2112.08571] with Julius Wons

and work in progress with Nathan Cohen and Ameek Malhotra

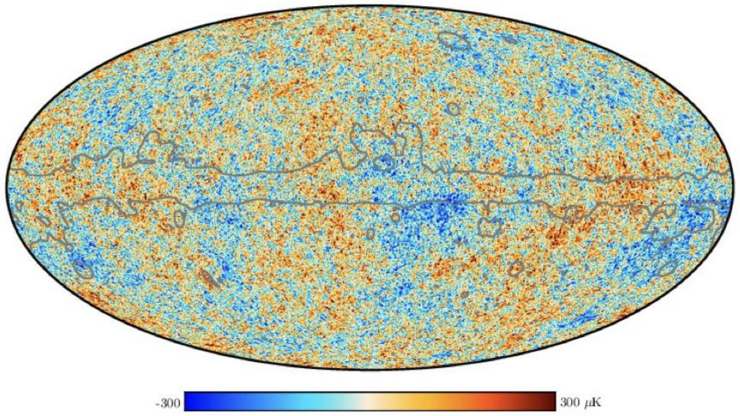


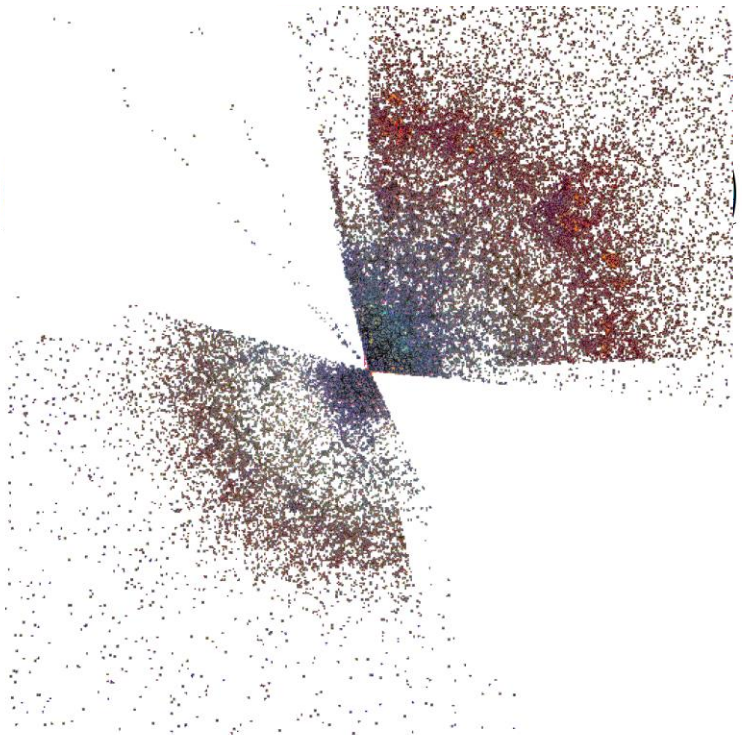


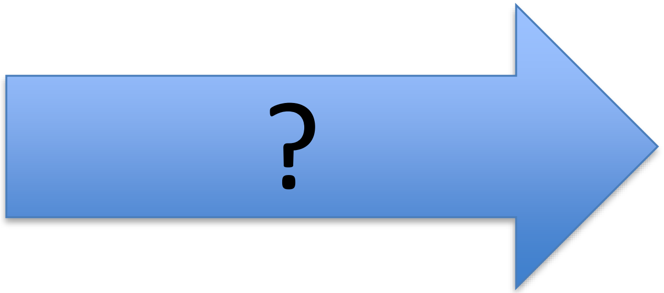
UNSW SYDNEY

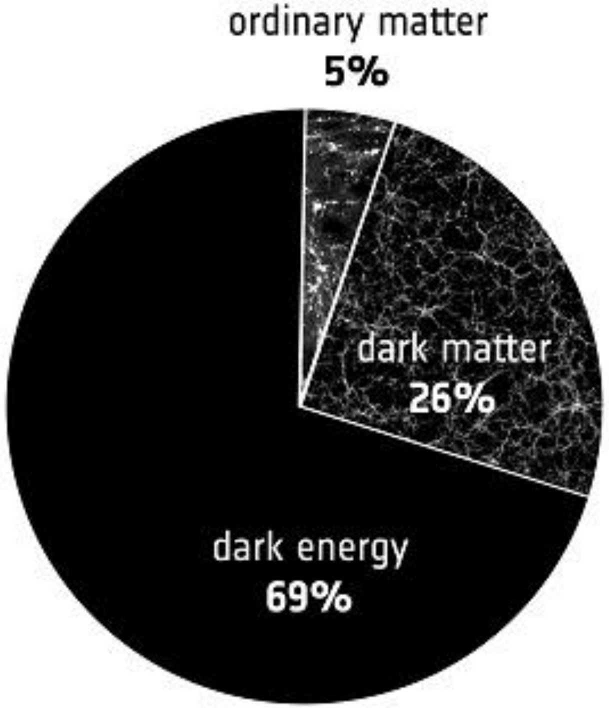
PPC 2024

14 -18 October 2024, Hyderabad, India


-300
300 μK




?


ordinary matter
5%
dark matter
26%
dark energy
69%

Parameter inference/optimisation

Cosmological model $\mathcal{M}$

Parameters $\boldsymbol{\theta}$

For instance:

Standard LCDM

$\boldsymbol{\theta} = (\omega_{\mathrm{b}}, \omega_{\mathrm{cdm}}, H_0, \tau, A_{\mathrm{s}}, n_{\mathrm{s}})$

or

LCDM + Neff

$\boldsymbol{\theta} = (\omega_{\mathrm{b}}, \omega_{\mathrm{cdm}}, H_0, \tau, A_{\mathrm{s}}, n_{\mathrm{s}}, \mathrm{N}_{\mathrm{eff}})$

etc.

Parameter inference/optimisation

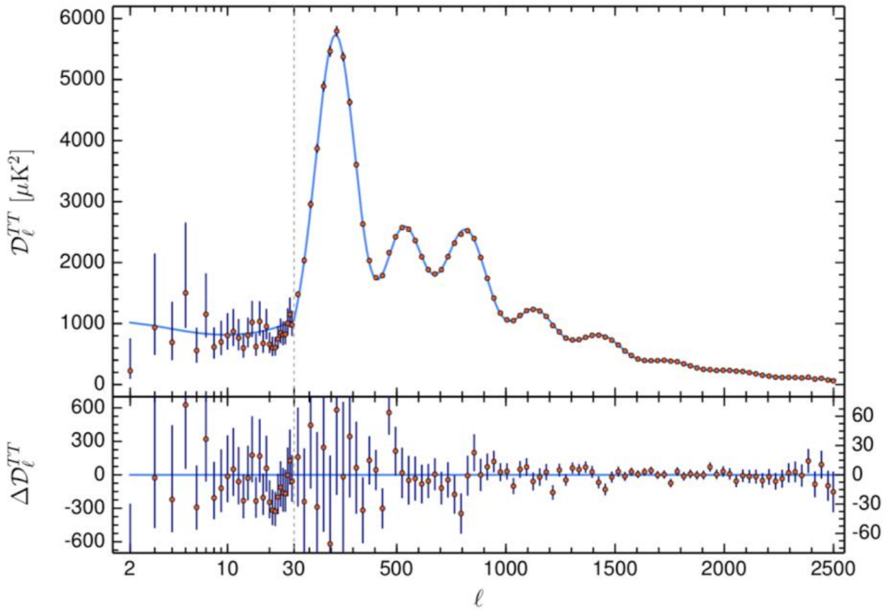
Cosmological model $\mathcal{M}$

Parameters $\boldsymbol{\theta}$

Boltzmann code
(CAMB/Class)

Theoretical prediction

(CMB angular power spectra)



Parameter inference/optimisation

Cosmological model $\mathcal{M}$

Parameters $\boldsymbol{\theta}$

→ Boltzmann code (CAMB/Class) →

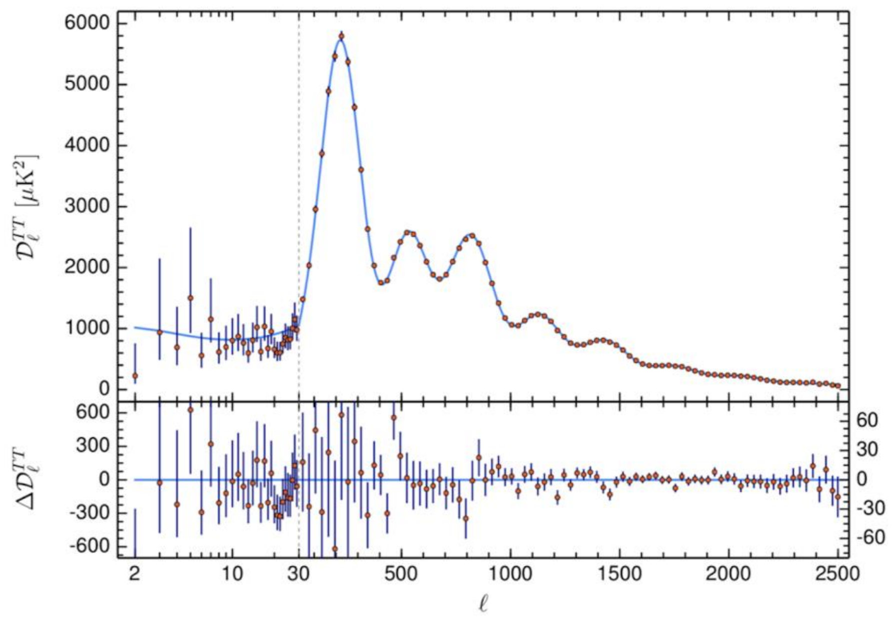
Theoretical prediction

(CMB angular power spectra)

↓ Data $\mathcal{D}$

Likelihood

$\mathcal{L}(\mathcal{D}\,|\,\boldsymbol{\theta}, \mathcal{M})$



Probability of the data given the model and specific values of the parameters

For uncorrelated Gaussian measurements:

$$-2\ln\mathcal{L} = \chi^2 = \sum_i \left(\frac{x_{\mathrm{th}}^{(i)} - x_{\mathrm{d}}^{(i)}}{\sigma^{(i)}} \right)^2$$

(theory: $x_{\mathrm{th}}^{(i)}$; data: $x_{\mathrm{d}}^{(i)}$; error: $\sigma^{(i)}$)

Parameter inference/optimisation

Cosmological model $\mathcal{M}$
Parameters $\boldsymbol{\theta}$

→ Boltzmann code (CAMB/Class) →

Theoretical prediction
(CMB angular power spectra)

↓ Data $\mathcal{D}$

Likelihood
$\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})$

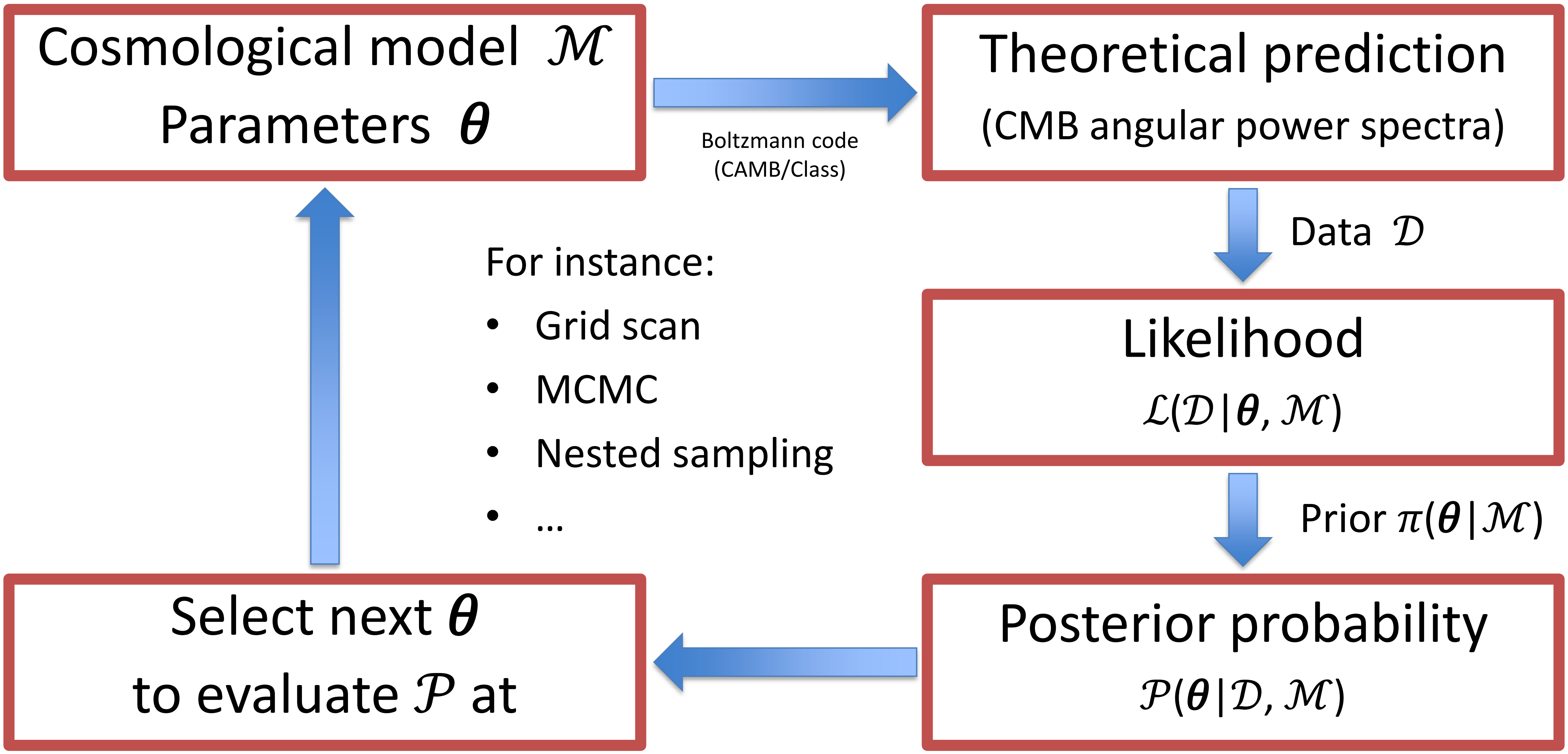
↓ Prior $\pi(\boldsymbol{\theta}|\mathcal{M})$

Posterior probability
$\mathcal{P}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$

Bayes' Theorem

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$

Parameter inference/optimisation



Parameter inference/optimisation

Cosmological model $\mathcal{M}$
Parameters $\boldsymbol{\theta}$

→ Boltzmann code (CAMB/Class)

Theoretical prediction
(CMB angular power spectra)

↓ Data $\mathcal{D}$

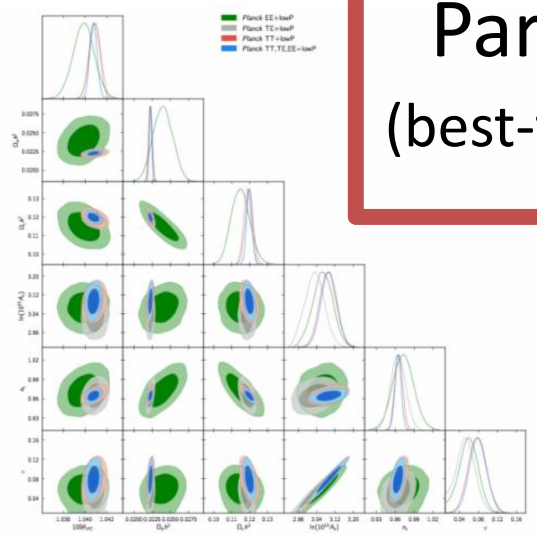
Likelihood
$\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})$

↓ Prior $\pi(\boldsymbol{\theta}|\mathcal{M})$

Posterior probability
$\mathcal{P}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$

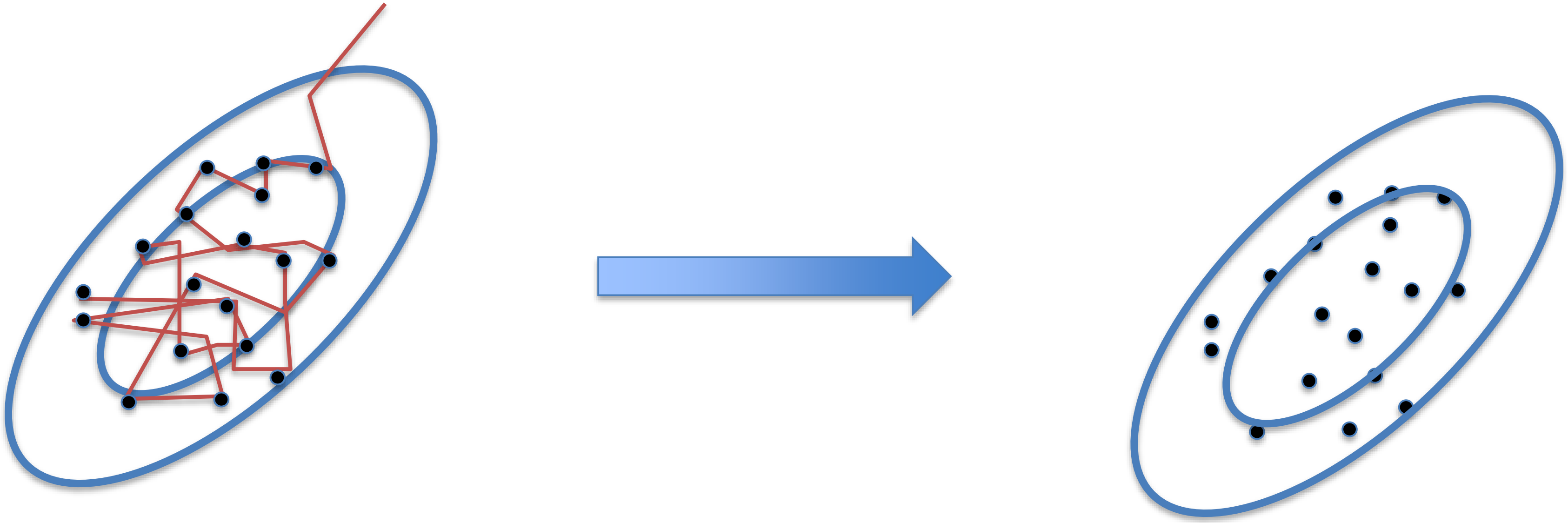
← termination criterion

Parameter constraints
(best-fits, means, variances, etc.)



The usual approach: Markov chain Monte Carlo

- Basic idea: random walk in parameter space that explores $\mathcal{P}(\boldsymbol{\theta})$



posterior
isocontours

Markov chain:

Density of samples proportional to $\mathcal{P}(\boldsymbol{\theta})$

The usual approach: Markov chain Monte Carlo

Metropolis-Hastings algorithm:

[Metropolis et al. (1953)]

1. Start at point $\boldsymbol{\theta}$ in parameter space
2. Save $\boldsymbol{\theta}$ to Markov chain
3. Propose a step to a new point $\boldsymbol{\theta}'$
4. Decide whether to accept the proposal and take the step:

   If $\mathcal{P}(\boldsymbol{\theta}') \geq \mathcal{P}(\boldsymbol{\theta})$, accept the proposal

   If $\mathcal{P}(\boldsymbol{\theta}') < \mathcal{P}(\boldsymbol{\theta})$, accept the proposal with a probability p = $\mathcal{P}(\boldsymbol{\theta}')/\mathcal{P}(\boldsymbol{\theta})$, otherwise reject
5. If step was accepted set $\boldsymbol{\theta}' = \boldsymbol{\theta}$
6. Go to 2.

Animated illustration:

http://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=standard

[Feng et al., Github]

Pros and cons of MCMC

+ easily implemented
+ easily parallelisable
+ essentially zero overhead
+ mild scaling of number of required samples with dimension N of parameter space (power law $\sim N^{\alpha}$ rather than exponential)
+ works great for near-Gaussian posteriors (most of cosmology)

o not very good at finding the maximum
o typically requires $\mathcal{O}(10^4)$ function evaluations for N = $\mathcal{O}(10)$

– struggles with complicated (multi-modal, non-Gaussian, non-linearly correlated, etc.) posteriors
– not very smart: most of the information is ignored!

Bayesian optimisation

Step 1: Regression

Guess the shape of the function based on known function values (“data”)

Step 2: Selection

Decide at which point to evaluate the next function value

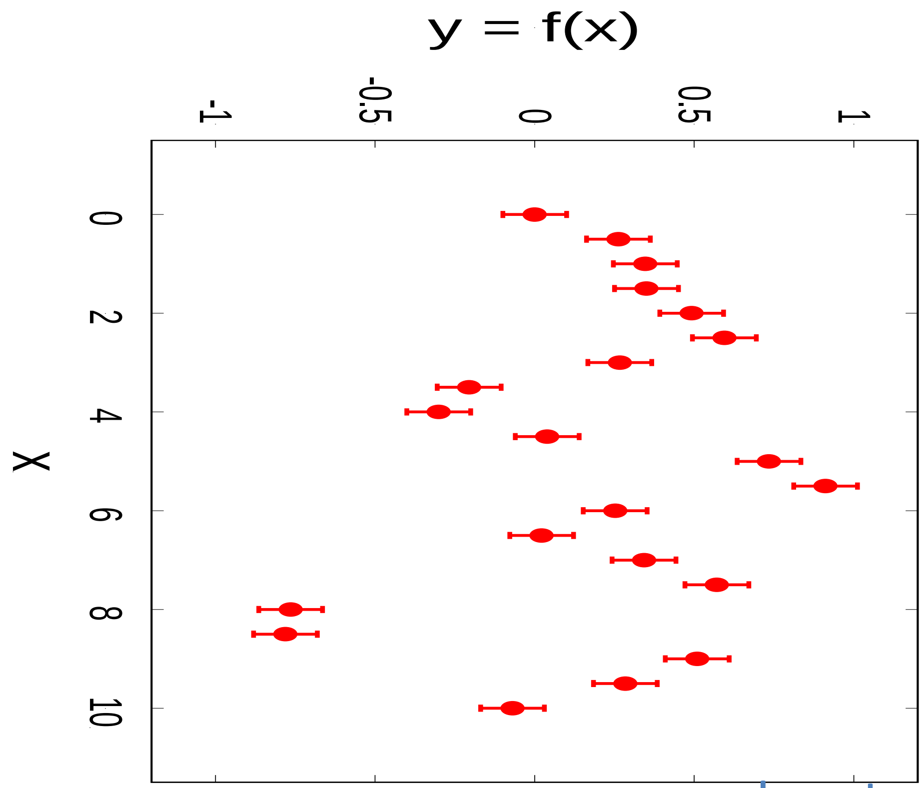
Gaussian Process Regression (GPR)

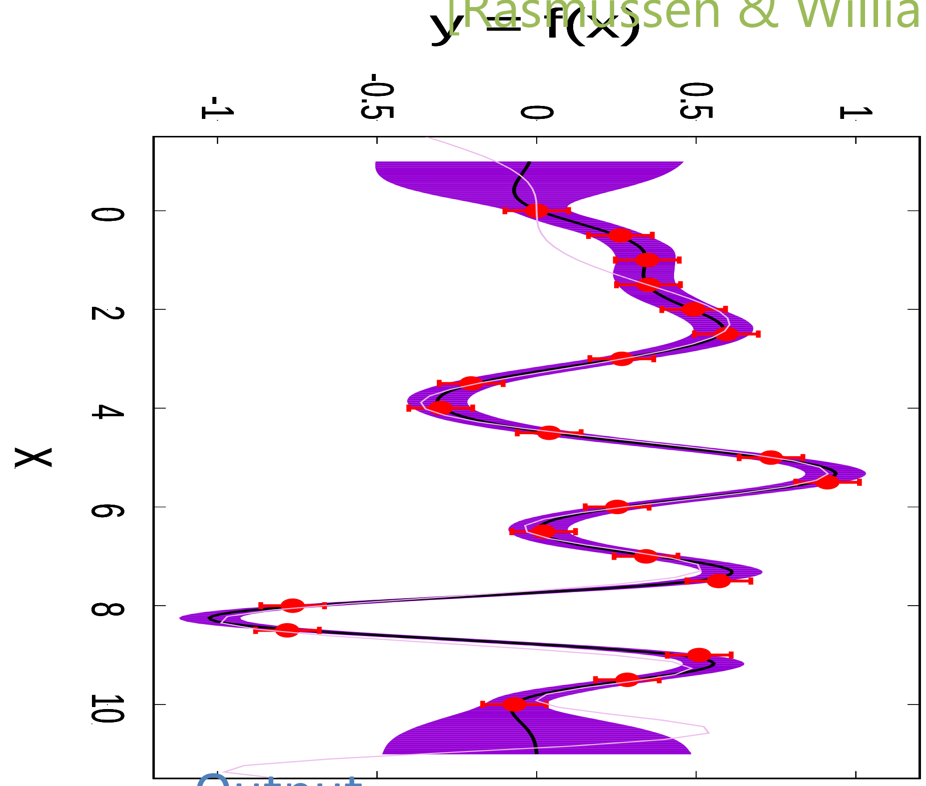
- Non-parametric probabilistic regression model

Gaussian Process Regression (GPR)…

…is a non-parametric probabilistic regression model

[Rasmussen & Williams 2006]

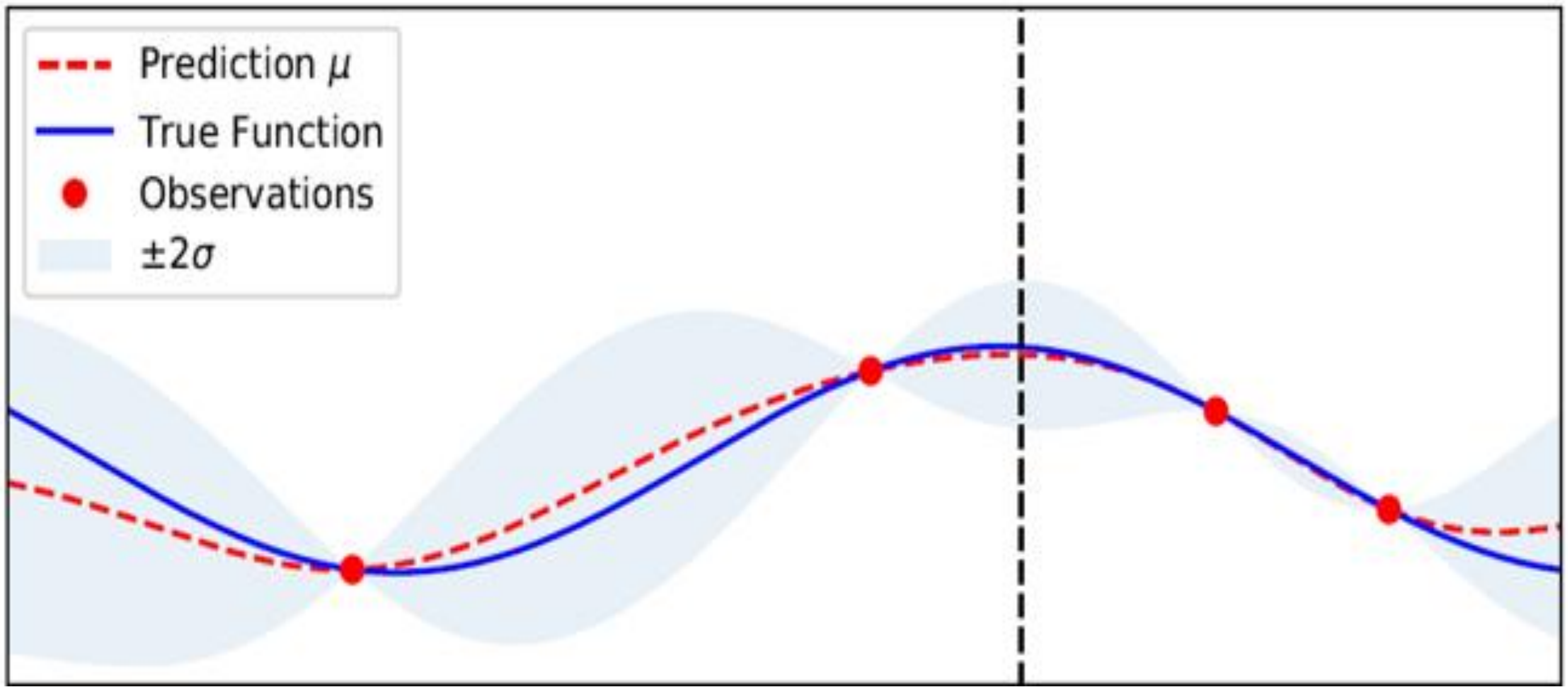




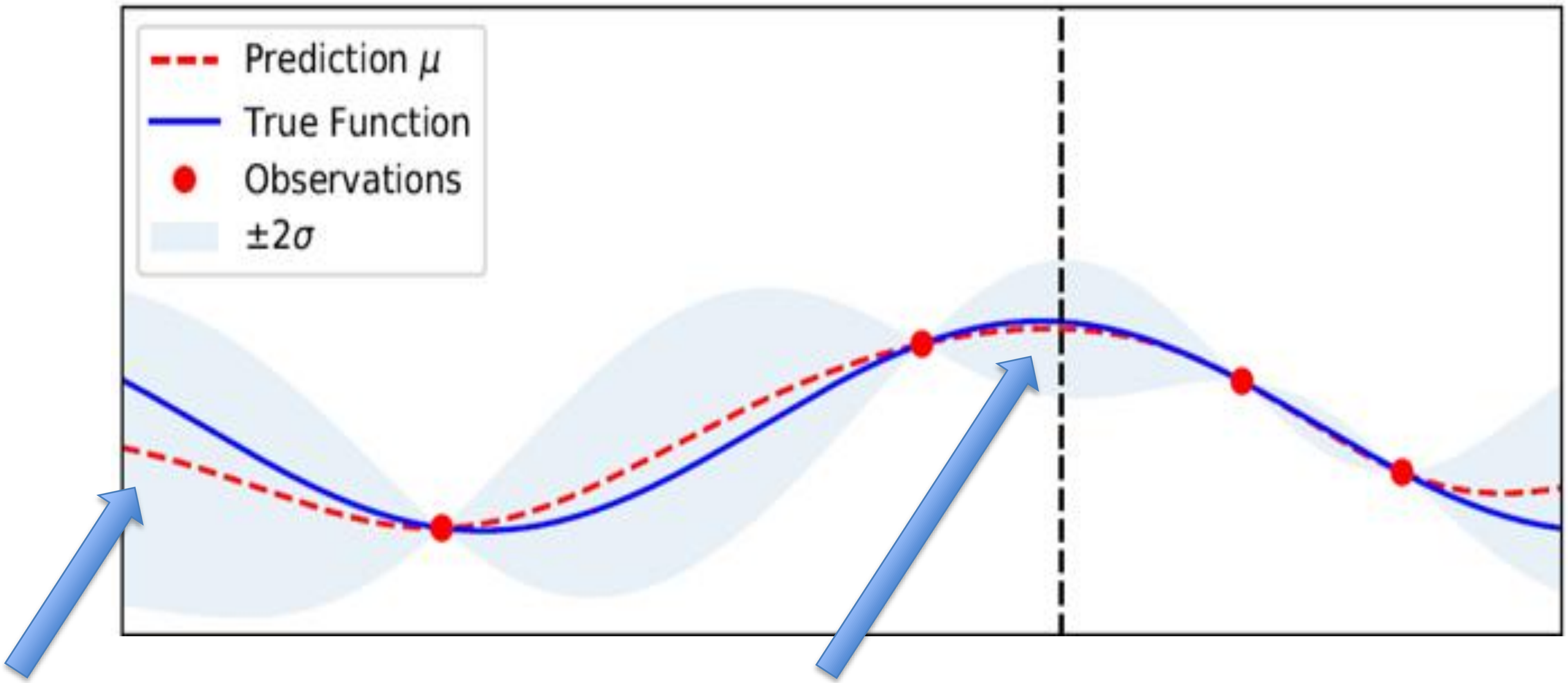
Input
- data points
- covariance of data

Output
- interpolation
- uncertainty

Gaussian Process Regression



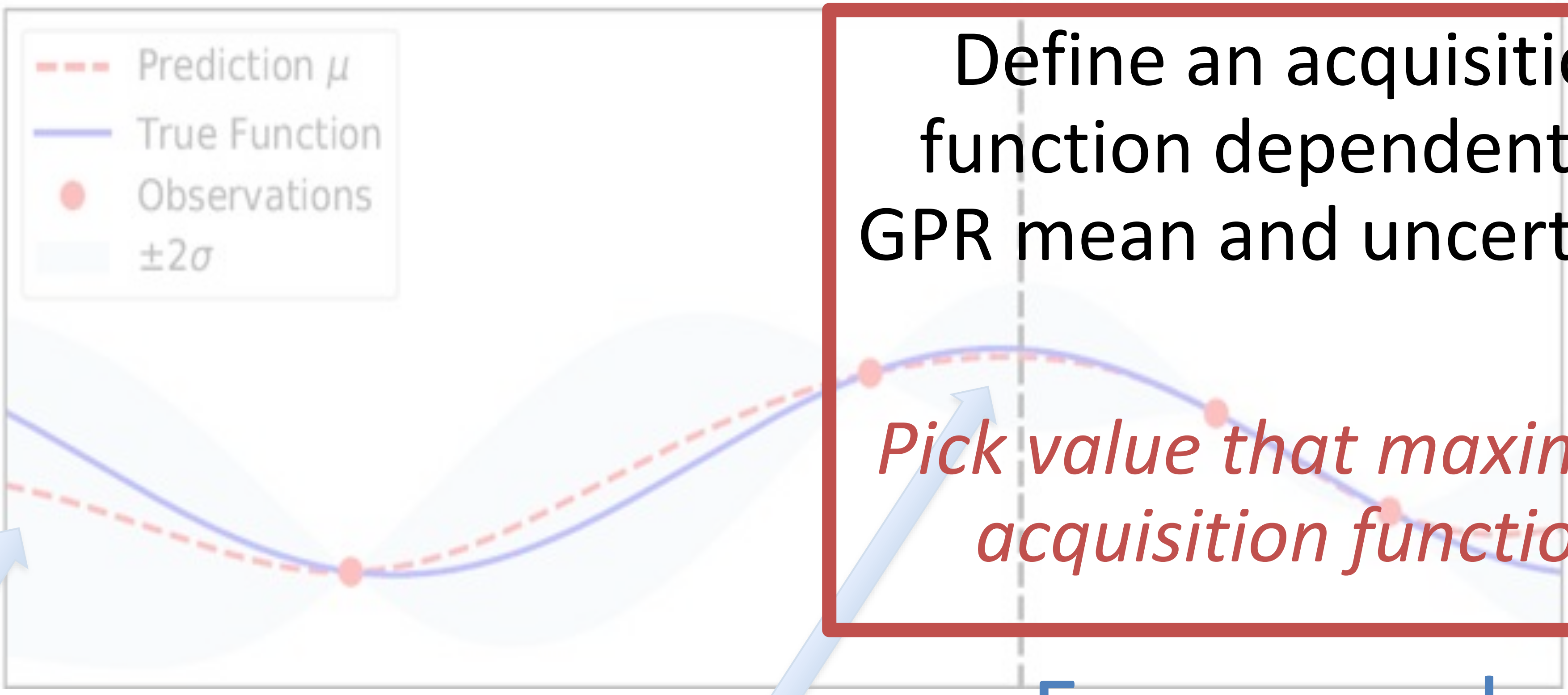
Where to draw the next sample?



Exploration?

Exploitation?

Where to draw the next sample?



Define an acquisition function dependent on GPR mean and uncertainty

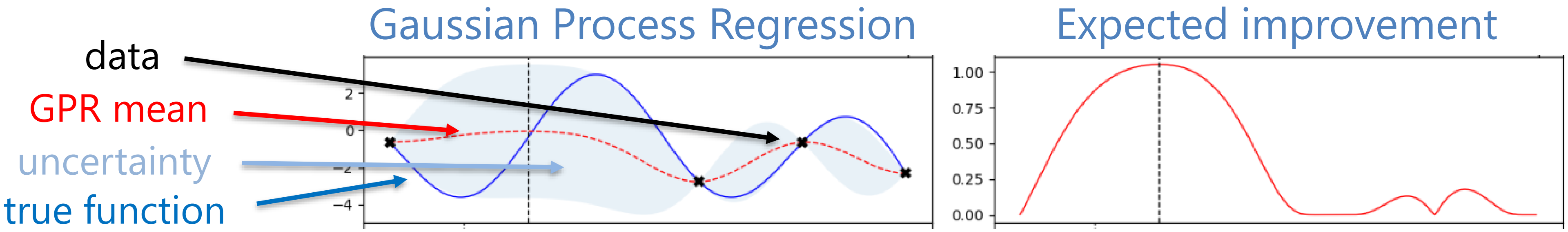
Pick value that maximises acquisition function

Exploration?

Exploitation?

For example:
Expected Improvement

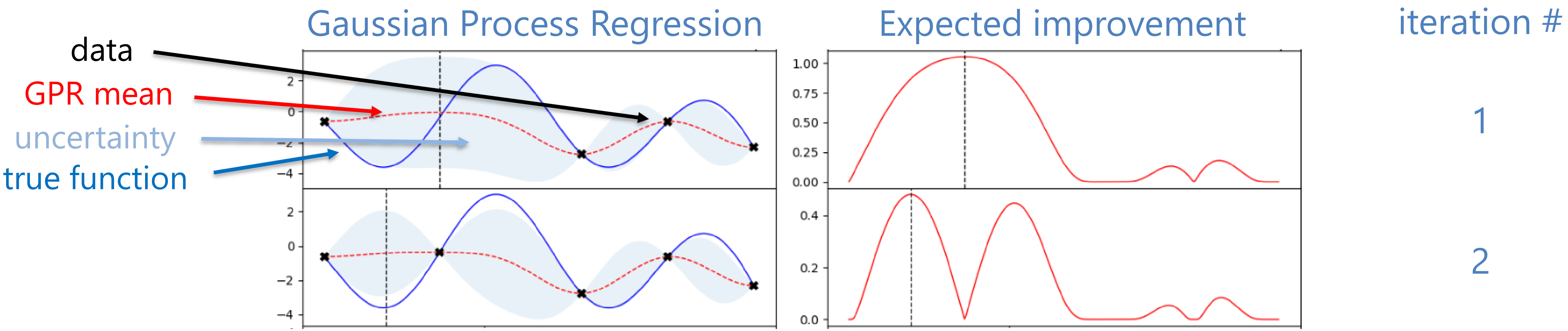
Bayesian optimisation



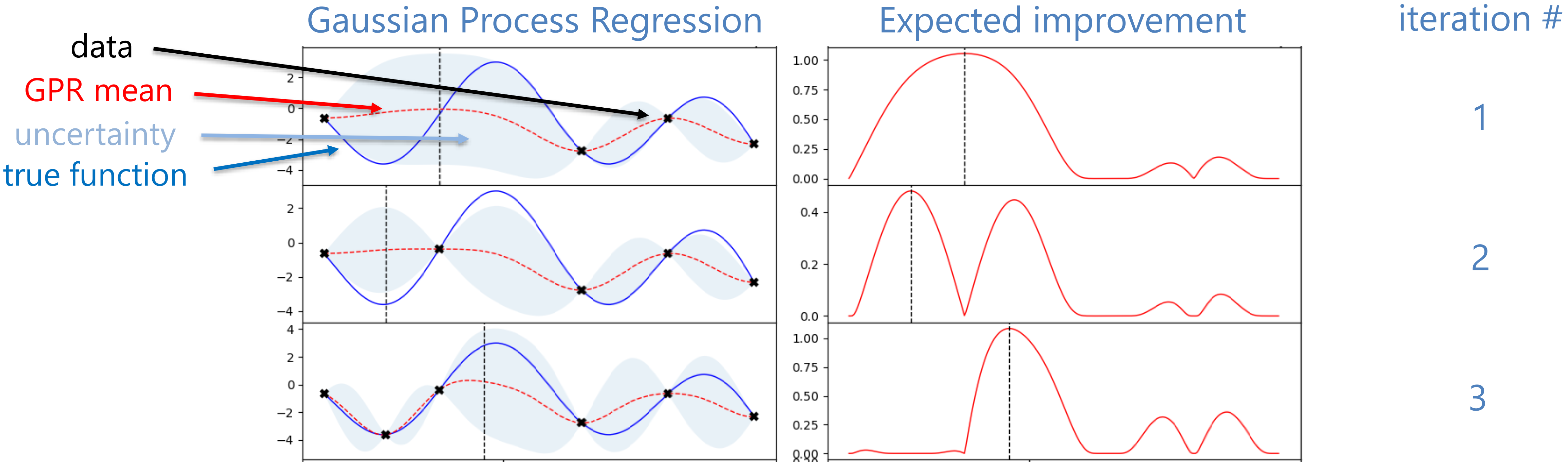
iteration #

1

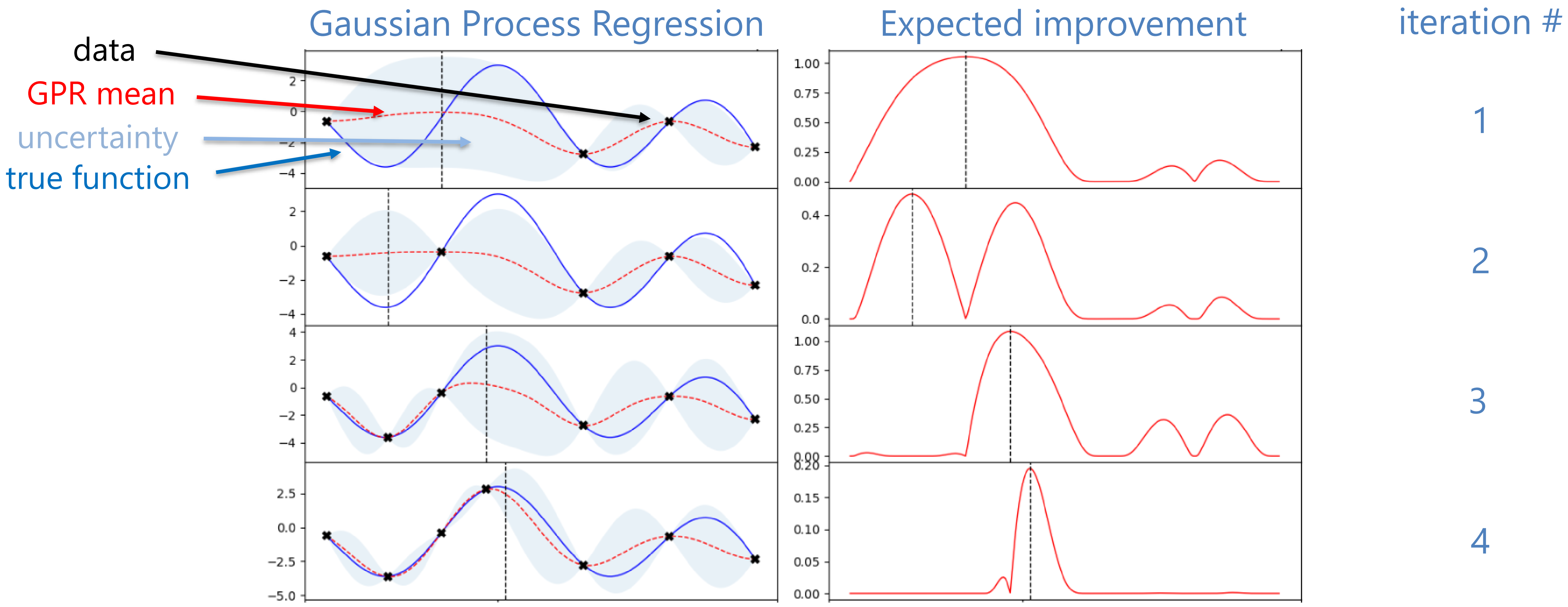
Bayesian optimisation



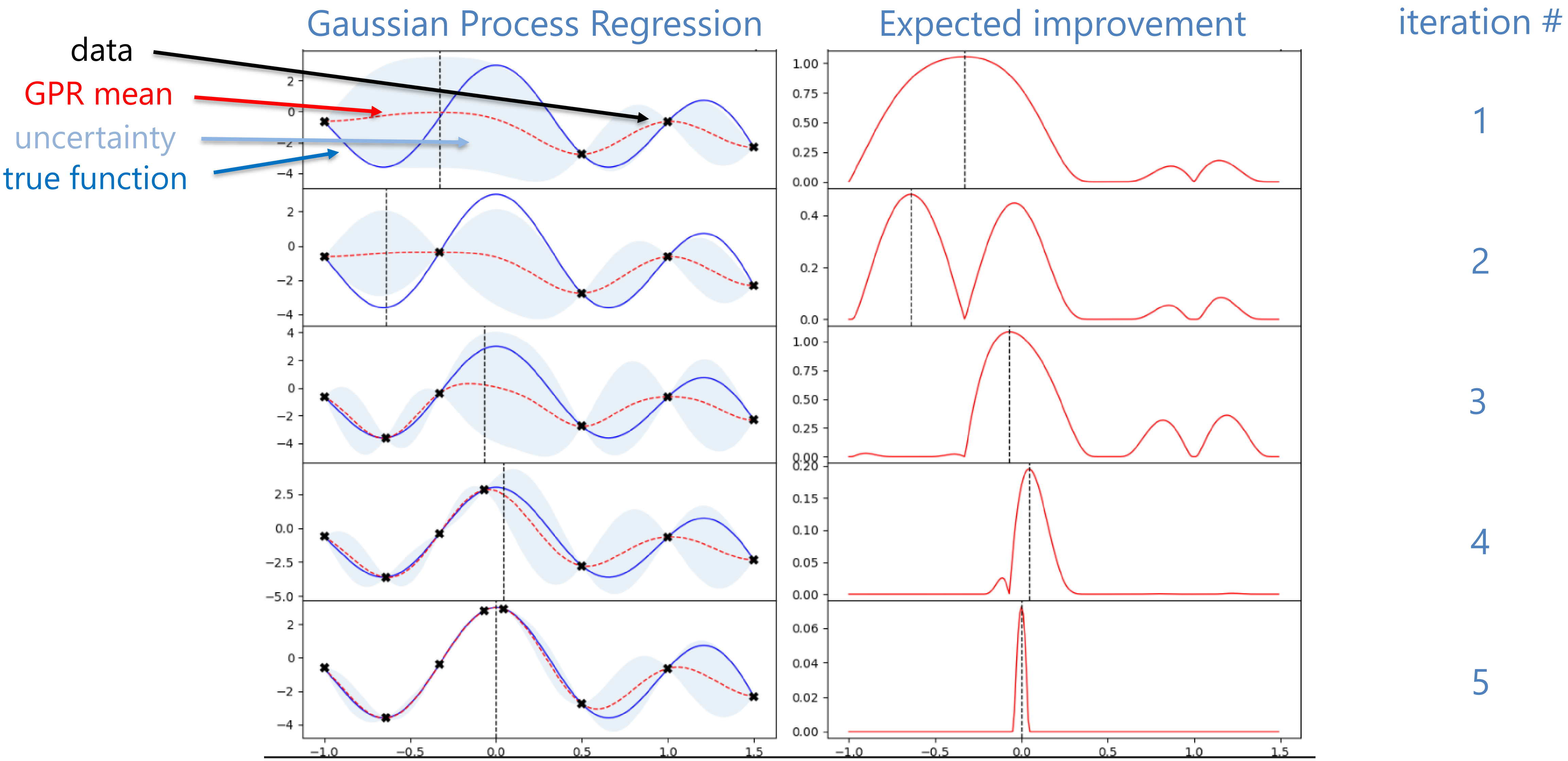
Bayesian optimisation



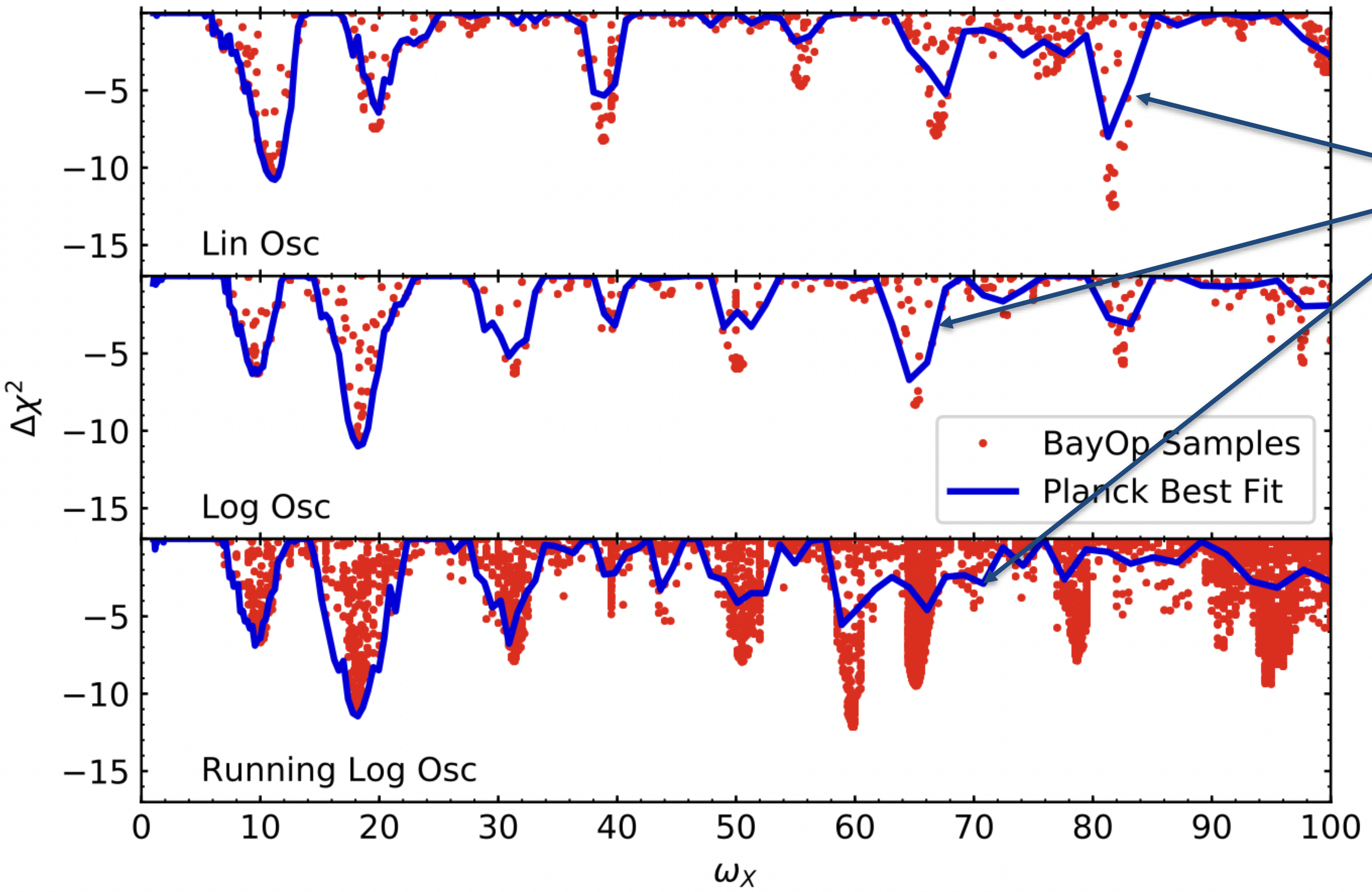
Bayesian optimisation



Bayesian optimisation



An example application: inflation models with modulated primordial power spectra



[*Planck* inflation 2018]

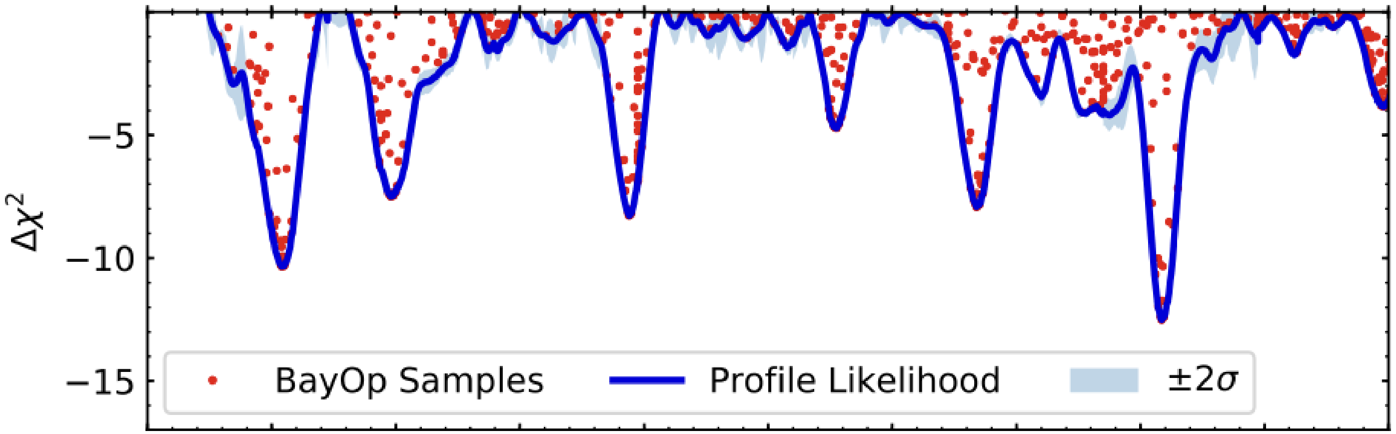
using nested sampling, O(10^5) samples

red dots: our results with BO

- Two orders of magnitude fewer function evaluations
- Much better at finding global and local extrema

[JH & Wons, 2021]

BayOp – not only good for optimisation



1700 samples
8 frequency bins

… it also learns the global shape of the function

[JH & Wons, 2021]

Pros and cons of Bayesian Optimisation

+ high efficiency
+ excellent at finding global maximum
+ very good at determining overall shape, profiles of functions
+ works even for very nasty (non-Gaussian, multimodal, etc.) functions
+ does not require user input or fine-tuning of settings to work
– may struggle with higher-dimensional problems ($D \gtrsim 10$)
– non-trivial computational overhead (CPU time, memory)

Bayesian optimisation for parameter inference

- Learn shape of posterior probability density
- Replace (potentially expensive) calculation of theoretical prediction and likelihood evaluation with (cheap!) GPR emulation
- Implemented in a Python package: GPry [El Gammal et al., 2022]

But this assumes we know the right model…

Model selection: Bayesian method

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M}) \cdot P(\mathcal{M})}{P(\mathcal{D})}$$

Probability of model $\mathcal{M}$ given the data $\mathcal{D}$

Bayesian evidence

$$P(\mathcal{D}|\mathcal{M}) = \int \mathrm{d}\theta \; \mathcal{L}(\mathcal{D}|\theta, \mathcal{M}) \, \pi(\theta|\mathcal{M})$$

Comparing two models: Bayes factor B_{12}

$$B_{12} = \frac{P(\mathcal{D}|\mathcal{M}_1)}{P(\mathcal{D}|\mathcal{M}_2)}$$

"Model $\mathcal{M}_1$ is B_{12} times more probable than $\mathcal{M}_2$"

Model selection: Bayesian method

Bayesian evidence

$$P(\mathcal{D}|\mathcal{M}) = \int \mathrm{d}\theta \; \mathcal{L}(\mathcal{D}|\theta, \mathcal{M}) \, \pi(\theta|\mathcal{M})$$

- Integral over entire parameter space
- Rewards models that make *risky* predictions and *get it right* over generic models that can *fit anything*
- Natural implementation of Occam's razor:

Numquam ponenda est pluralitas sine necessitate

Plurality must never be posited without necessity

(Don't make things unnecessarily complicated)

Bayesian model selection

- Multi-dimensional integration is a challenging task
- Standard approach: Nested sampling algorithm

[Skilling 2004, Feroz et al. 2013, Handley et al. 2015]

- typically requires $\mathcal{O}(10^5\text{-}10^6)$ function evaluations for features models

This is even harder than parameter inference

Can Bayesian Optimisation help?

Evidence calculation with Bayesian optimisation

- Goal is to select next function value to be evaluated in such a way that it maximises the expected reduction in uncertainty of the integral
- Use a different acquisition function: Integrated Mean Square Prediction Error (IMSPE)

GPR uncertainty

$$\mathrm{IMSPE}(\theta) = \int \mathrm{d}\theta' \; \sigma_{\widehat{\mathrm{GP}(\theta)}}(\theta')$$

Pretend to take a sample at $\boldsymbol{\theta}$, then do a new GPR

Very convenient:
gives estimate of the uncertainty of the evidence integral

Evidence calculation with Bayesian optimisation

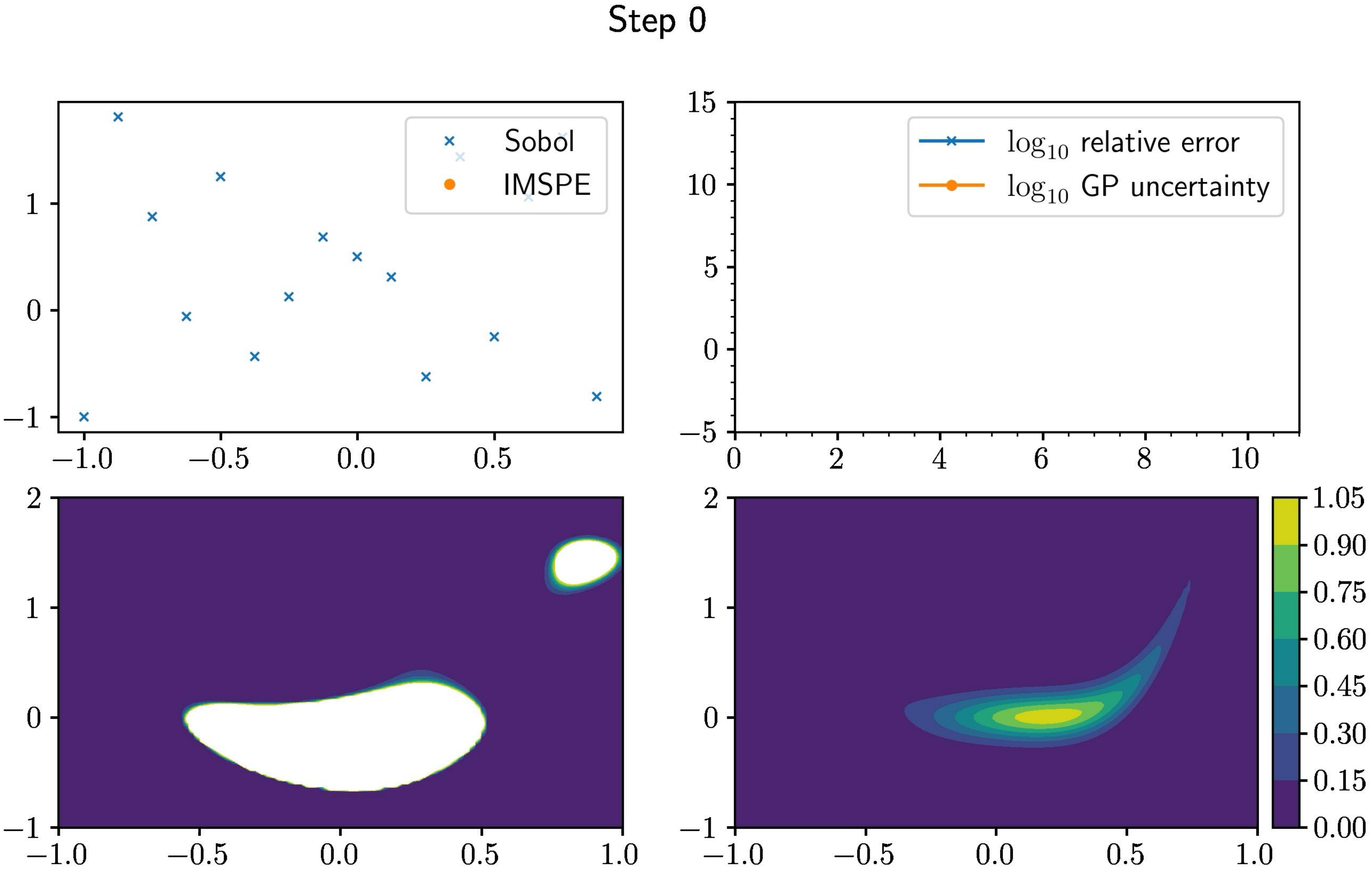
- Our code still in development…
- Code based largely on existing Python frameworks (BoTorch)

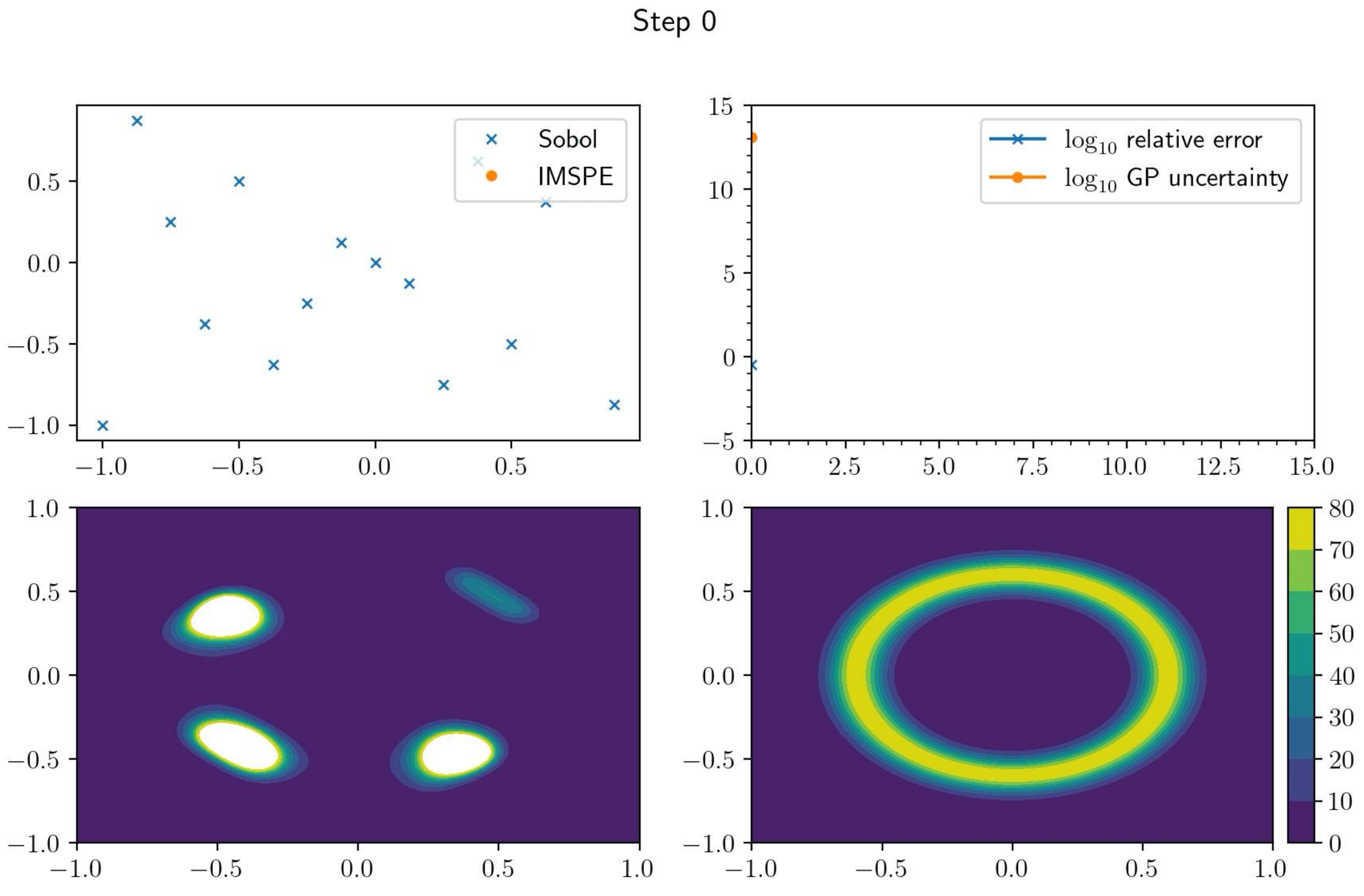
[Balandat et al. (2019)]

- Uses clever method for dealing with hyperparameters and acquisition function maximization (Sparse Axis-Aligned Subspace Bayesian Optimisation (SAASBO))

[Eriksson & Jankowiak (2021)]

- Sampling from hyperparameter space PDF instead of maximizing (overengineering? – but more Bayesian in spirit)


Step 0
Sobol
IMSPE
$\log_{10}$ relative error
$\log_{10}$ GP uncertainty
1.05
0.90
0.75
0.60
0.45
0.30
0.15
0.00



Step 0
Sobol
IMSPE
$\log_{10}$ relative error
$\log_{10}$ GP uncertainty

Conclusions

- Bayesian optimisation is a machine-learning technique for extremising unknown functions
- It can also be applied to cosmological parameter estimation and Bayesian model comparison
- Very efficient: in our examples it requires factor O(100) fewer function evaluations compared to random sampling-based methods
- Most useful for expensive-to-calculate likelihoods and complicated posterior distributions
- Paper and code for Bayesian evidence calculation out soon!