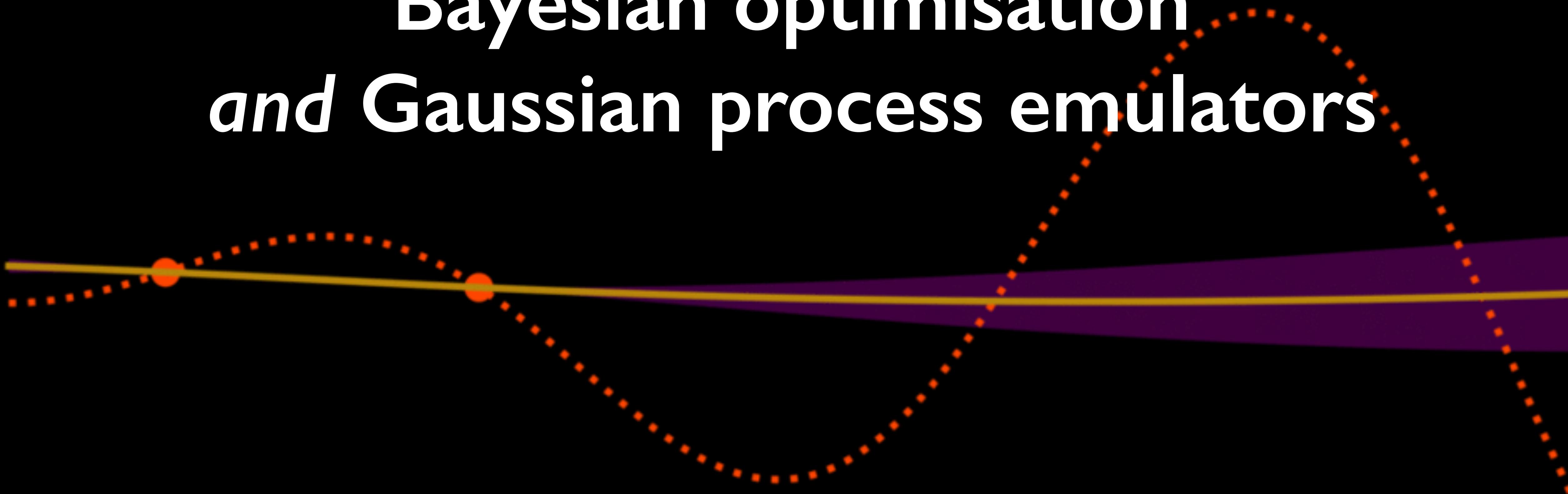
Optimised interpolation of costly simulations: Bayesian optimisation *and* Gaussian process emulators

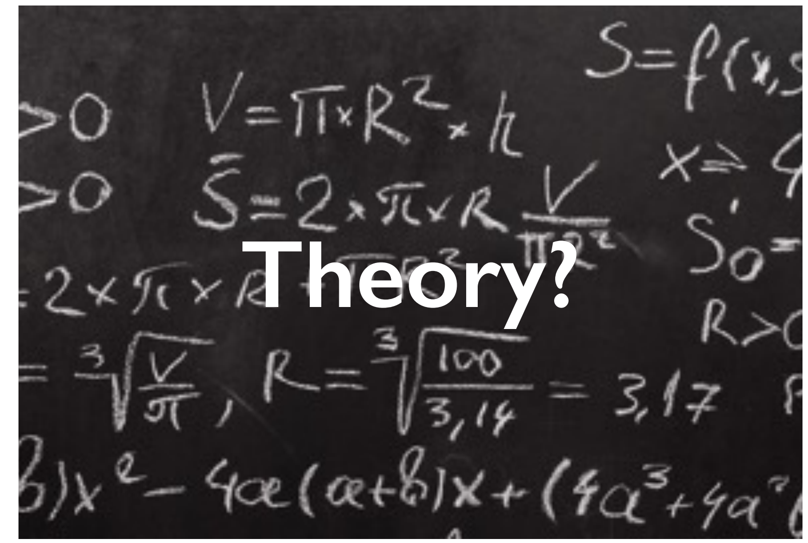


Keir K. Rogers

OKC Fellow, Oskar Klein Centre for Cosmoparticle Physics,
Department of Physics, Stockholm University

How should we compare

vs

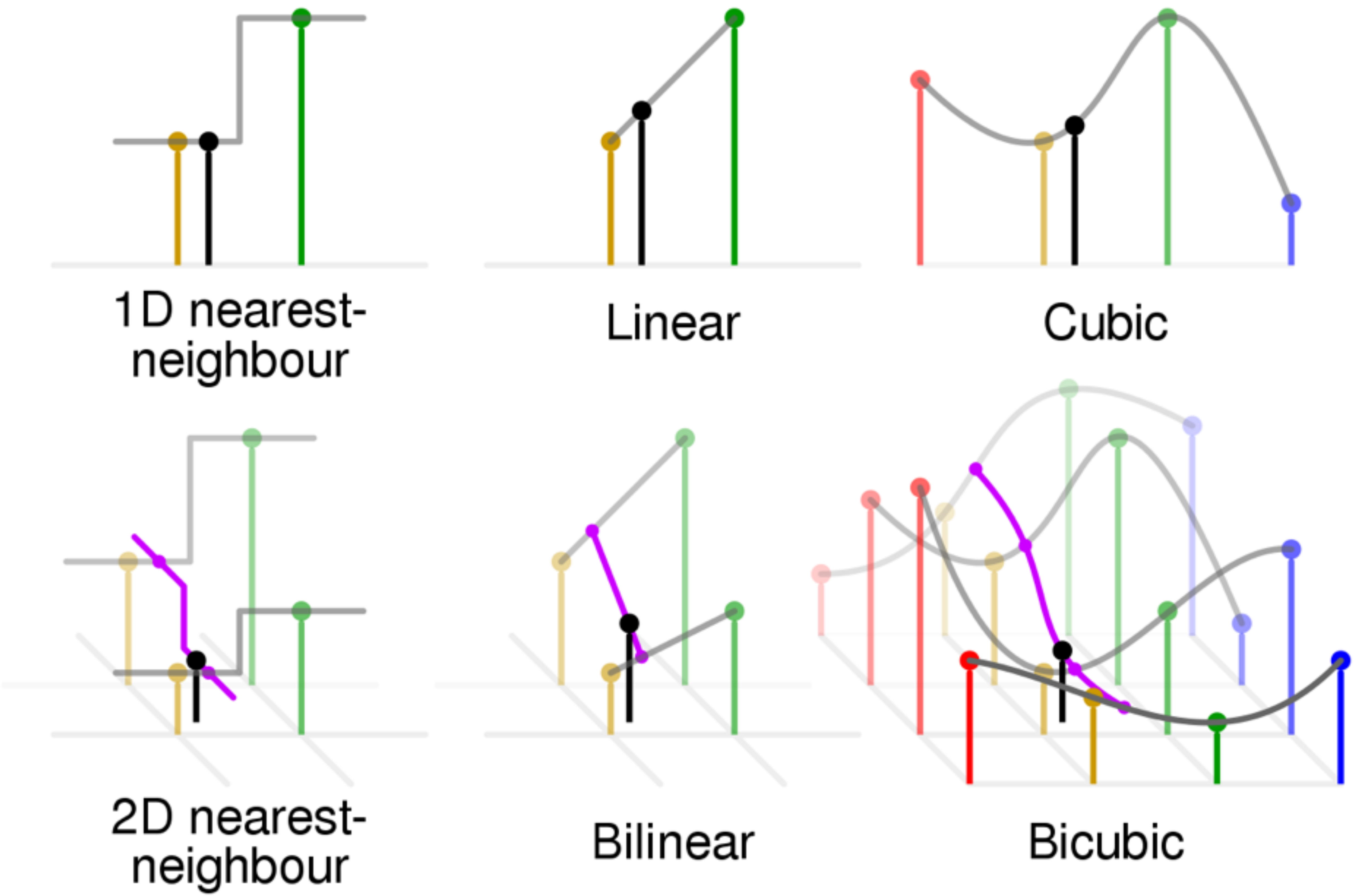


- Often require > **millions of simulations** of mock data — often unfeasible
- Generic solution is **Gaussian process emulation** (Bayesian interpolation)
- Emulator made accurate by **Bayesian optimisation** (optimal training set)

Some examples of "costly simulations" across the sciences

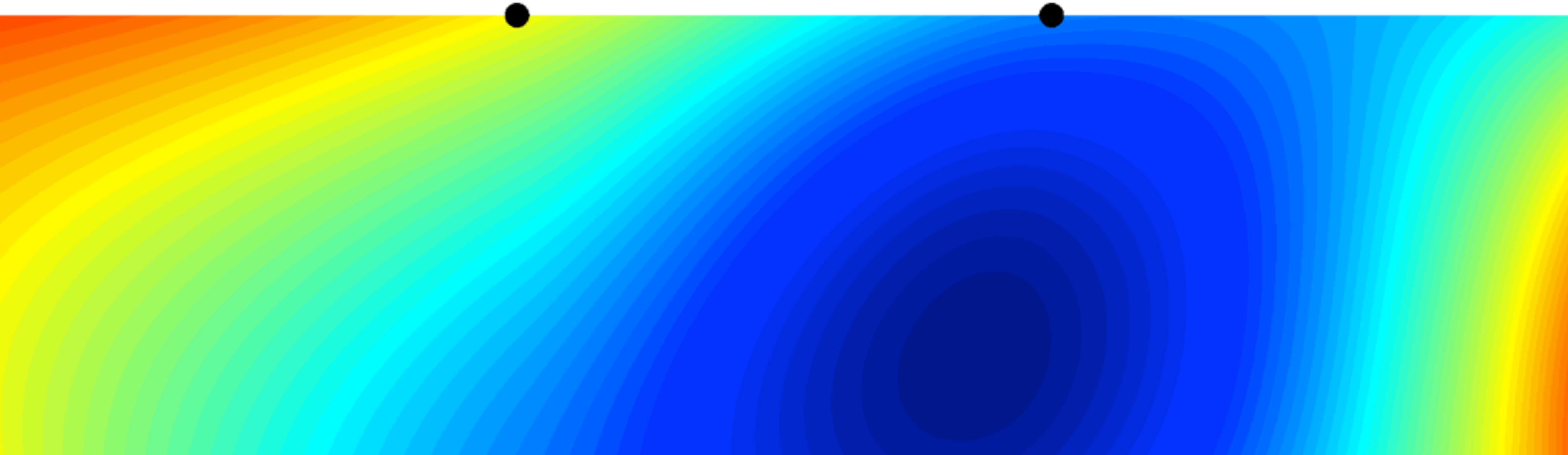
- **Cosmology** — to infer from galaxy/quasar surveys — need to simulate (non-linear) cosmological evolution of billions of dark matter/gas particles
- **Engineering** — to optimise aircraft design — need to evaluate complex, non-linear models, *e.g.*, of air flow over aircraft wing (*surrogate modelling*)
- **Climatology** — to predict when we're all doomed — need to evaluate complex, non-linear models of atmosphere/oceans
- **Biology** — to optimise wastewater treatment — need to simulate microbial community with 10^{18} (!) particles

Simple interpolation



Simple interpolation can be a bit too simple

- Have to **choose functional form** for interpolation — risk of "overfitting"
- **Inefficient use of "training set"** — often restricted to regular simulation grid
- **No reliable theoretical error estimate** (have to empirically estimate)



A Gaussian process

- **Stochastic process** = (infinite) “collection” of random variables

- **Gaussian process** = any finite sub-set is multivariate Gaussian distribution

- **Model simulation outputs** as Gaussian process

$$f(\mathbf{x}) \sim \mathcal{N}(0, K(\mathbf{x}, \mathbf{x}'; \theta))$$

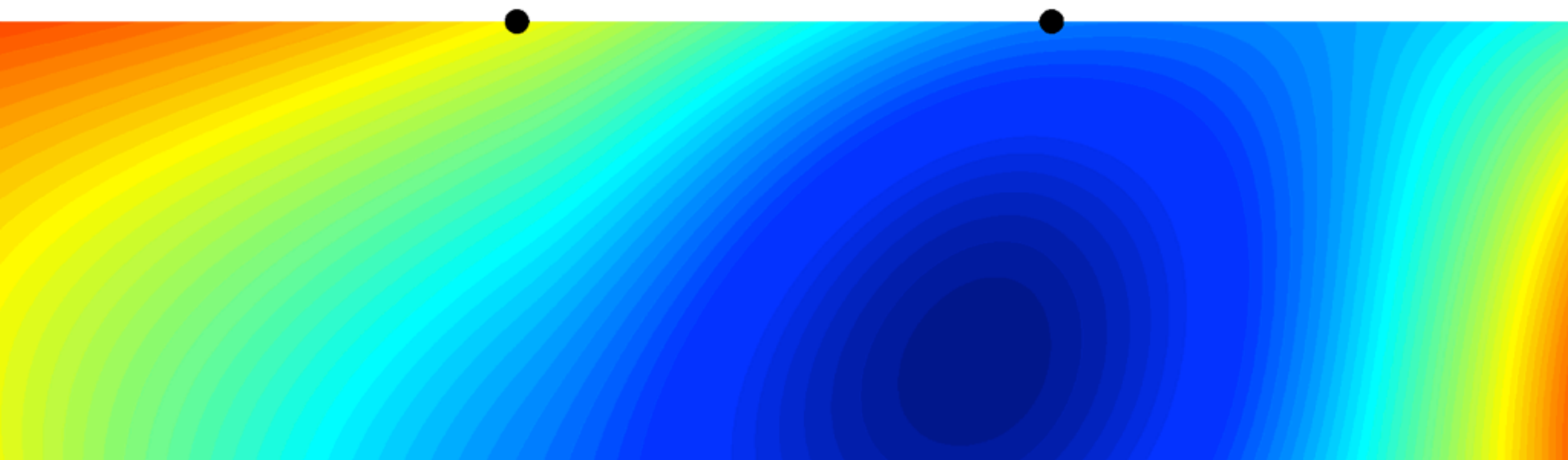
Simulation output

Kernel hyperparameters

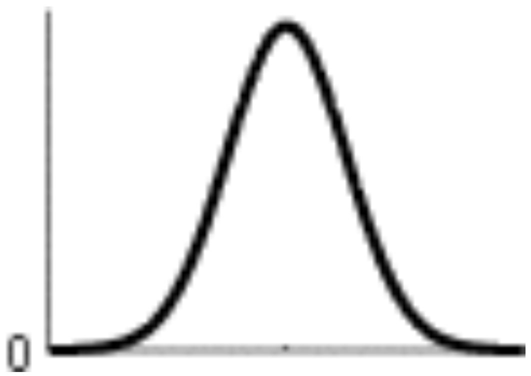
Simulation parameters

Gaussian process model is very general

- Spans **wide range of function space**
- **Full use of training set** — model correlations between all simulation outputs
- **Probabilistic model** inherently gives uncertainty



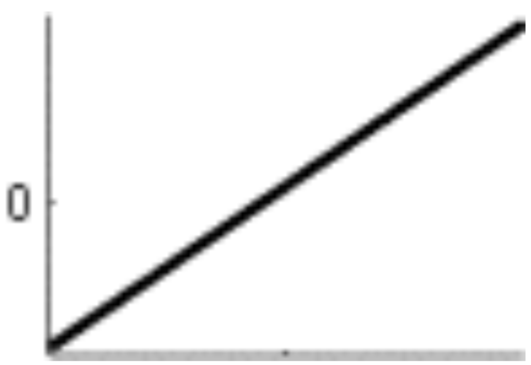
Some examples of covariance kernels



$$\sigma^2 e^{-\frac{(\mathbf{x}-\mathbf{x}')^2}{2l^2}}$$

Squared exponential

Stationary
Isotropic
Smooth



$$\sigma^2 \mathbf{x}.\mathbf{x}'$$

Linear

Non-stationary
Non-isotropic

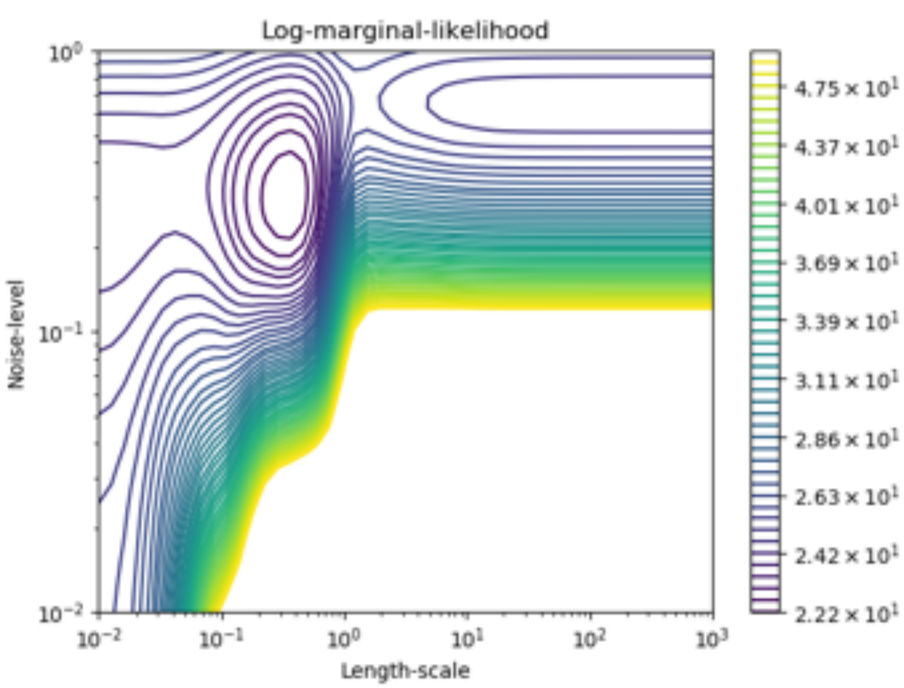
$$\sigma^2 e^{-\frac{2\sin^2\left(\frac{\pi|\mathbf{x}-\mathbf{x}'|}{p}\right)}{l^2}}$$

Periodic

Hyperparameter θ optimisation

Maximise log (Gaussian) marginal likelihood of training set w.r.t. θ:

$$\log \mathcal{L}(f(\mathbf{x})|\theta;\mathbf{x}) = -\frac{1}{2} f(\mathbf{x})^{\mathrm{T}} K(\mathbf{x},\mathbf{x};\theta)^{-1} f(\mathbf{x}) - \frac{1}{2}\log\det K(\mathbf{x},\mathbf{x};\theta) - \frac{|\mathbf{x}|}{2}\log 2\pi$$



- **L-BFGS-B** "gradient descent" method popular
- Should **marginalise over θ** (MCMC)

Interpolation as posterior predictive distribution

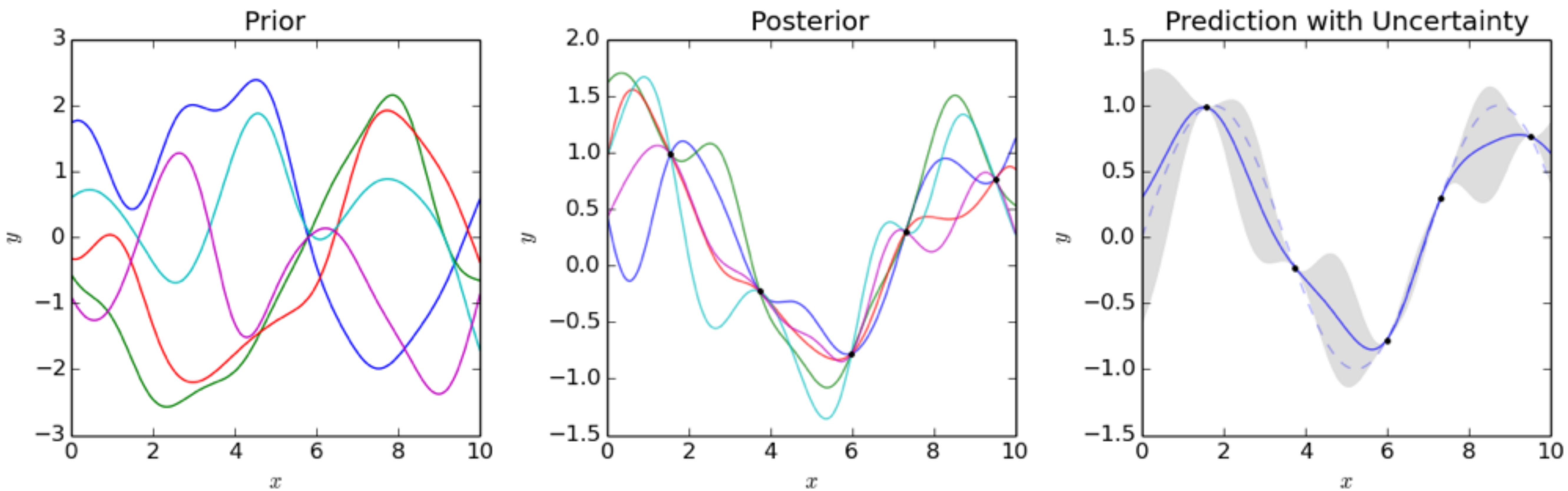
Posterior distribution of new simulation output *f*(x*) conditional on training set *f*(x)

$$p(f(\mathbf{x}^*)|f(\mathbf{x}), \mathbf{x}, \mathbf{x}^*) \sim \mathcal{N}(K_* K^{-1} f(\mathbf{x}), K_{**} - K_* K^{-1} K_*^{\mathrm{T}})$$

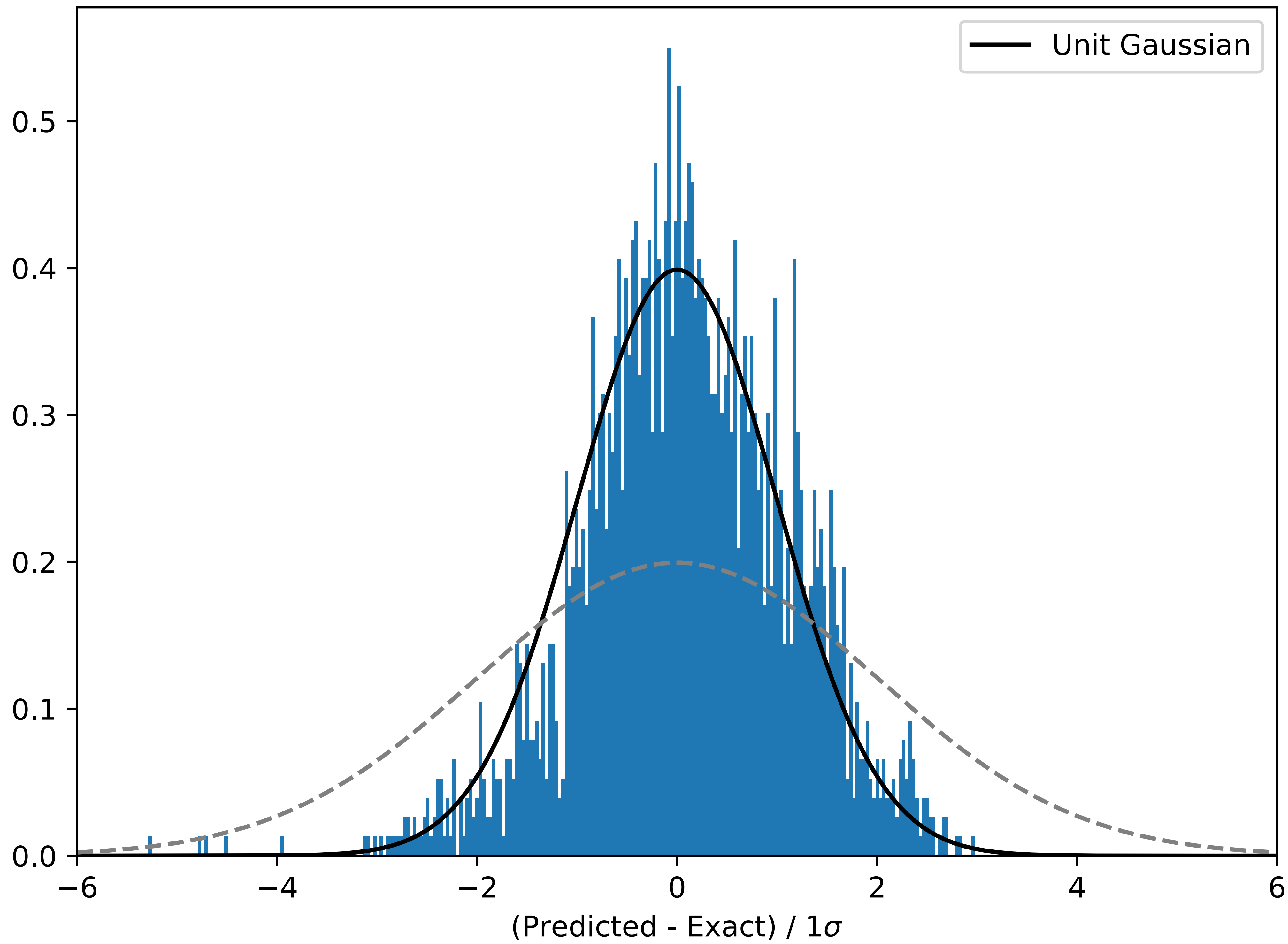
$$K_* = K(\mathbf{x}^*, \mathbf{x}; \theta) \qquad\qquad K_{**} = K(\mathbf{x}^*, \mathbf{x}^*; \theta)$$

- Posterior **analytically determined**
- Variance (uncertainty) **independent of simulation output**

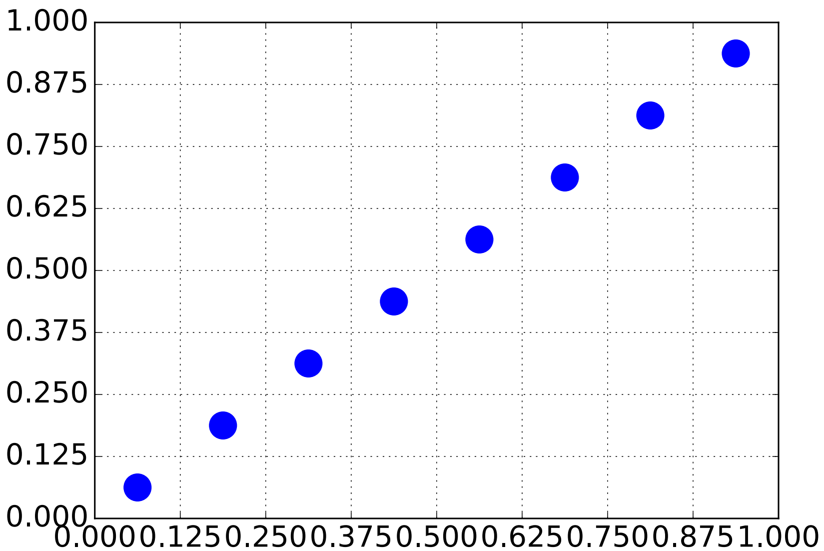
Probabilistic interpolation

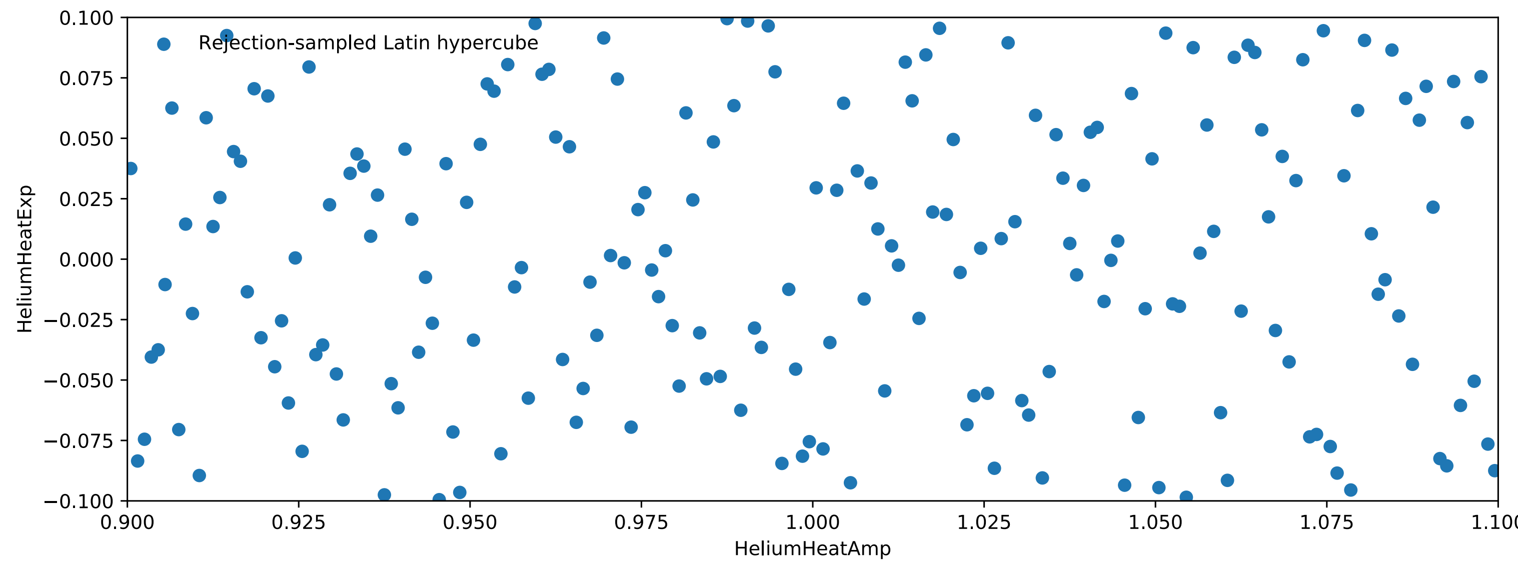


Cross-validation

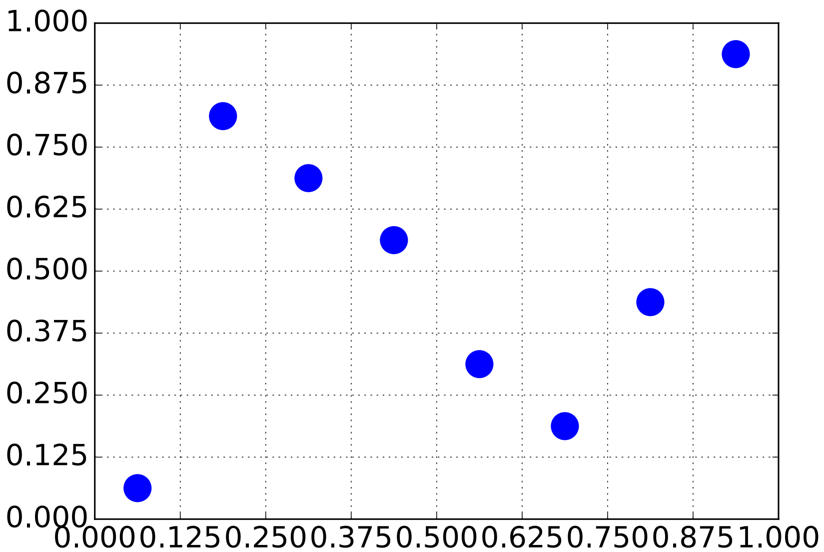


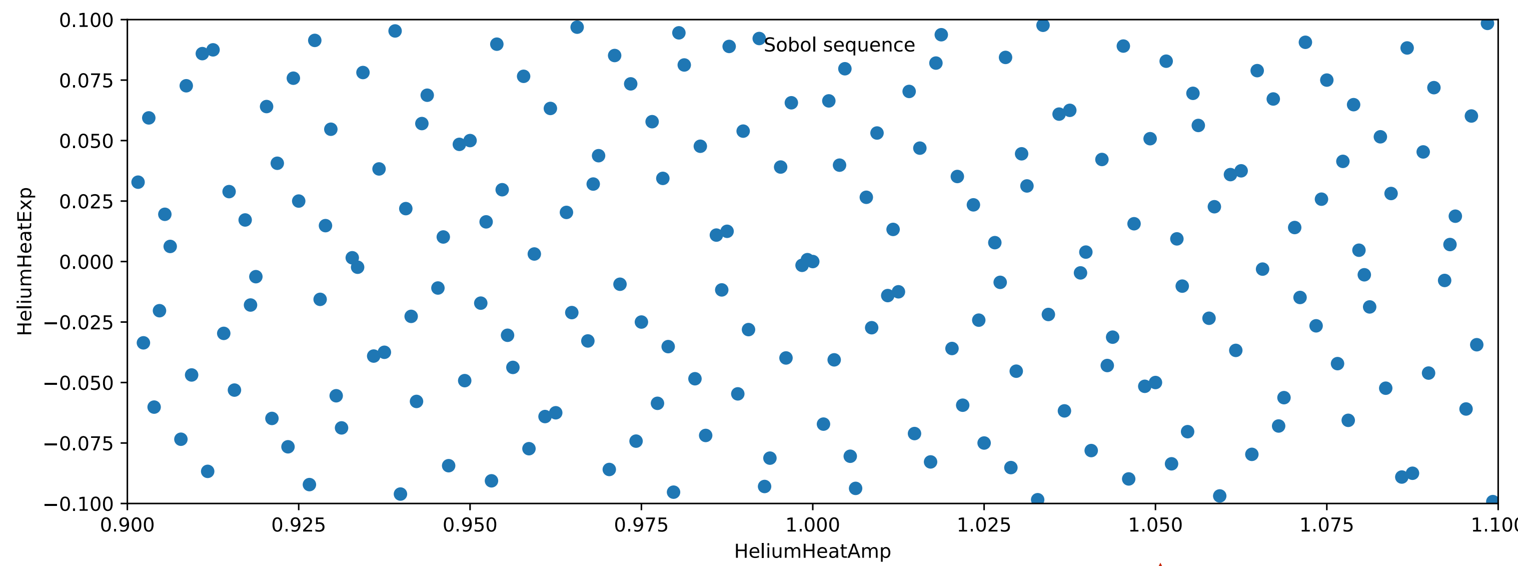
The training set is key





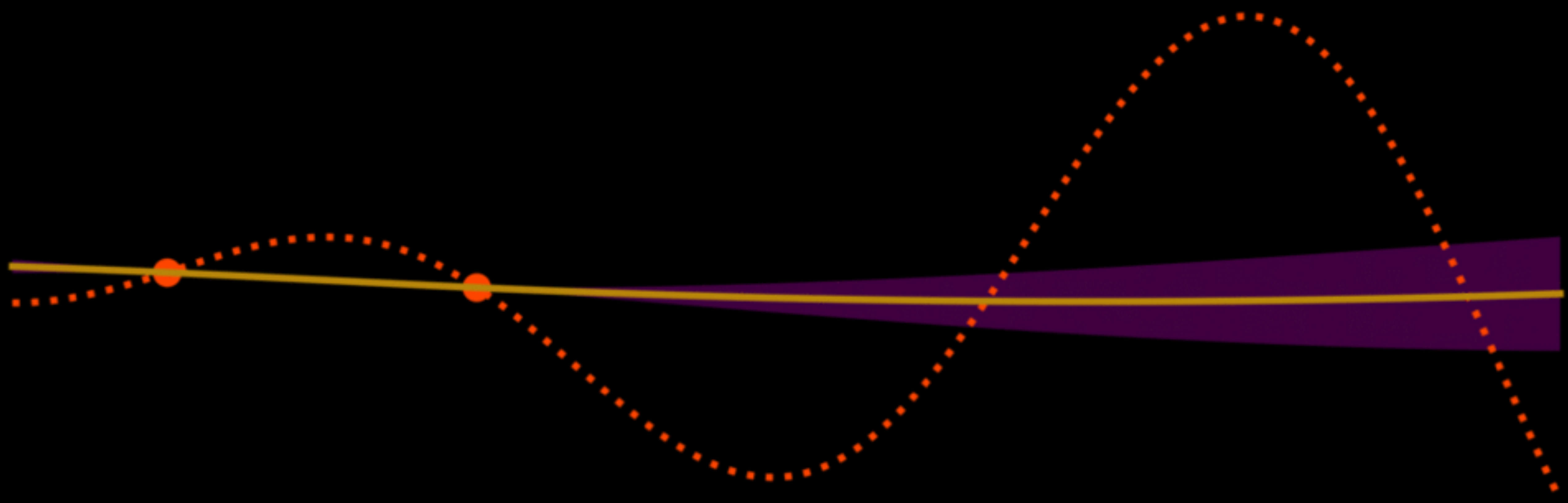
Latin hypercube





Sobol sequence

Can we actively construct the training set?



Bayesian optimisation

We need a balance between

vs

Exploitation

where Gaussian process variance is large

where objective function is large

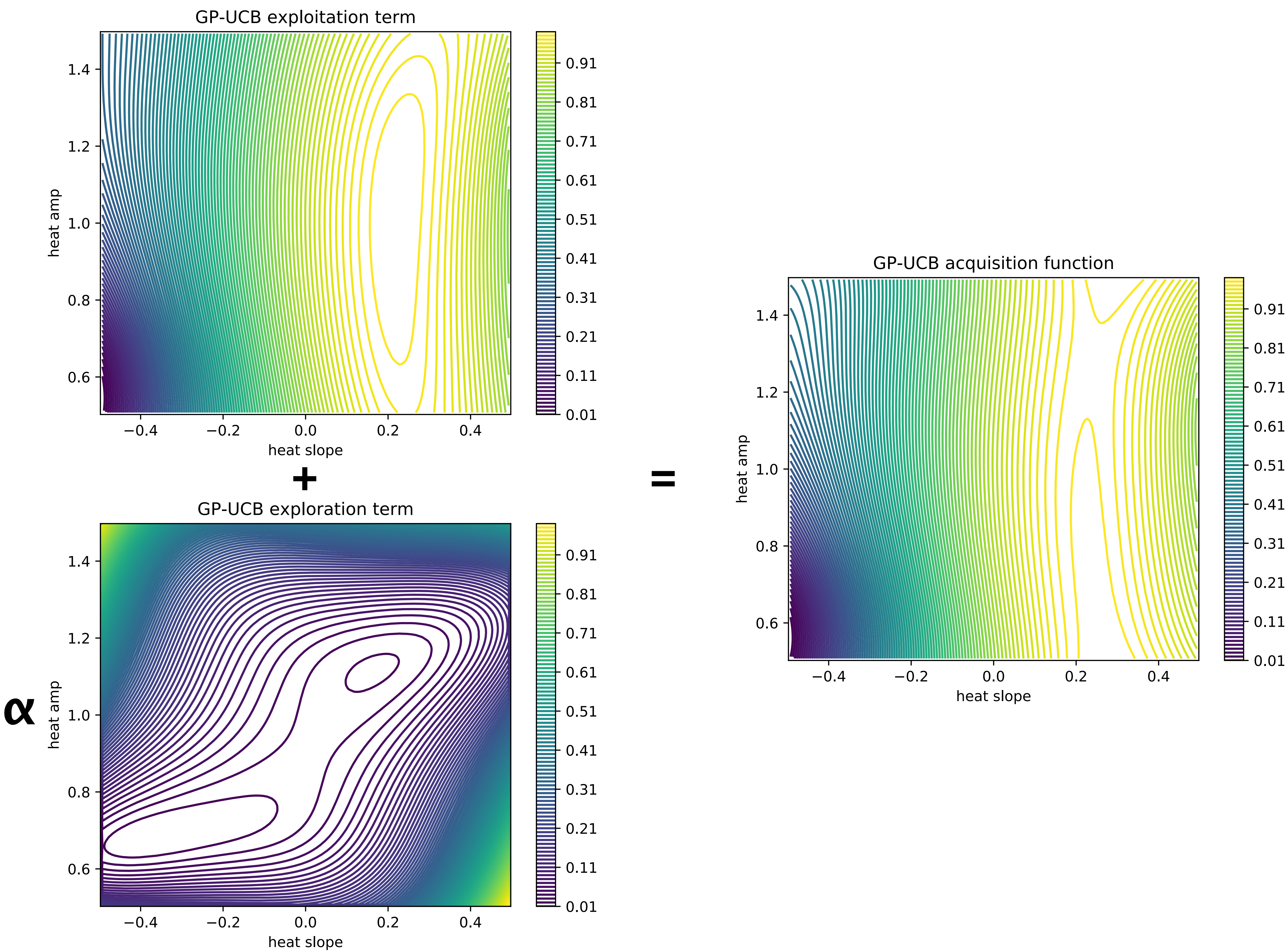
GP-UCB acquisition function =

α

+

Exploitation (μ)

- Run new training simulation at **maximum of acquisition function**
- α effectively determines **how many sigma** confident we are
- α can be optimised for **“minimum regret”** (quasi-convergence)


GP-UCB exploitation term
heat amp
heat slope
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0.01
0.11
0.21
0.31
0.41
0.51
0.61
0.71
0.81
0.91
+
GP-UCB exploration term
α
=
GP-UCB acquisition function

Some more sophisticated acquisition functions

Expected improvement

$$\mathrm{EI}(\mathbf{x}^*) \equiv \mathrm{E}\{\max[f(\mathbf{x}^*) - \max[f(\mathbf{x})], 0]\}$$

$$\equiv \sigma(\mathbf{x}^*)\left[z\Phi(z) + \phi(z)\right]$$

Exploration

$$z = \frac{\mu(\mathbf{x}^*) - \max[f(\mathbf{x})]}{\sigma(\mathbf{x}^*)}$$

Exploitation

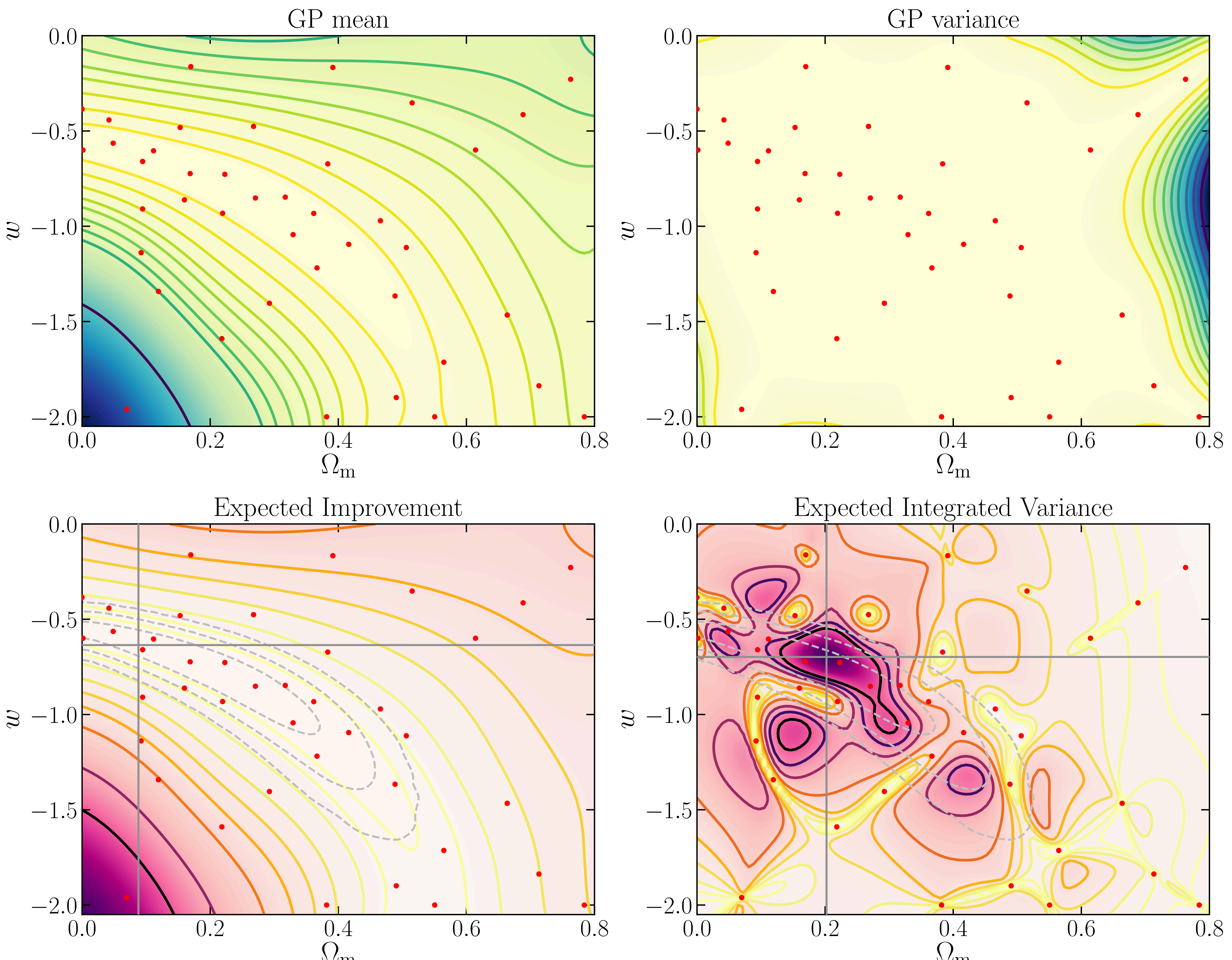
Some more sophisticated acquisition functions

Expected integrated variance

$$\mathrm{EIV}(\mathbf{x}^*) \equiv \mathrm{E}\left\{ \int \mathrm{d}\mathbf{x}\, \mathrm{Var}[p(\mathbf{x}|\mathbf{d})] \right\}$$

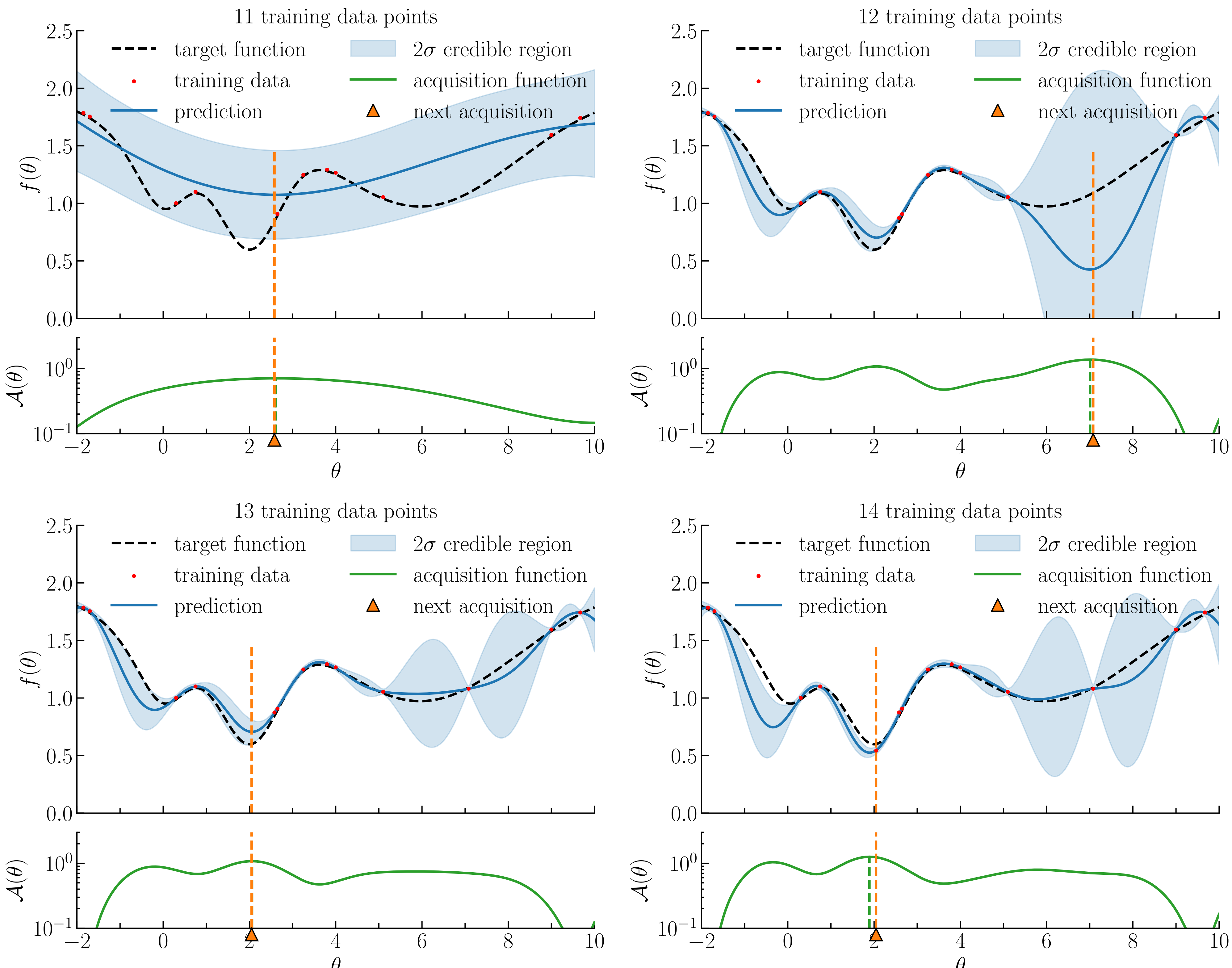
$$\equiv \int \mathrm{d}\mathbf{x}\, \frac{p(\mathbf{x})^2}{4} \mathrm{e}^{-\mu(\mathbf{x})} \left[\sigma^2(\mathbf{x}) - \frac{\mathrm{Cov}^2(\mathbf{x},\mathbf{x}^*)}{\sigma^2(\mathbf{x})} \right]$$

Exploitation **Exploration**



Leclercq (2018)

Serial Bayesian optimisation



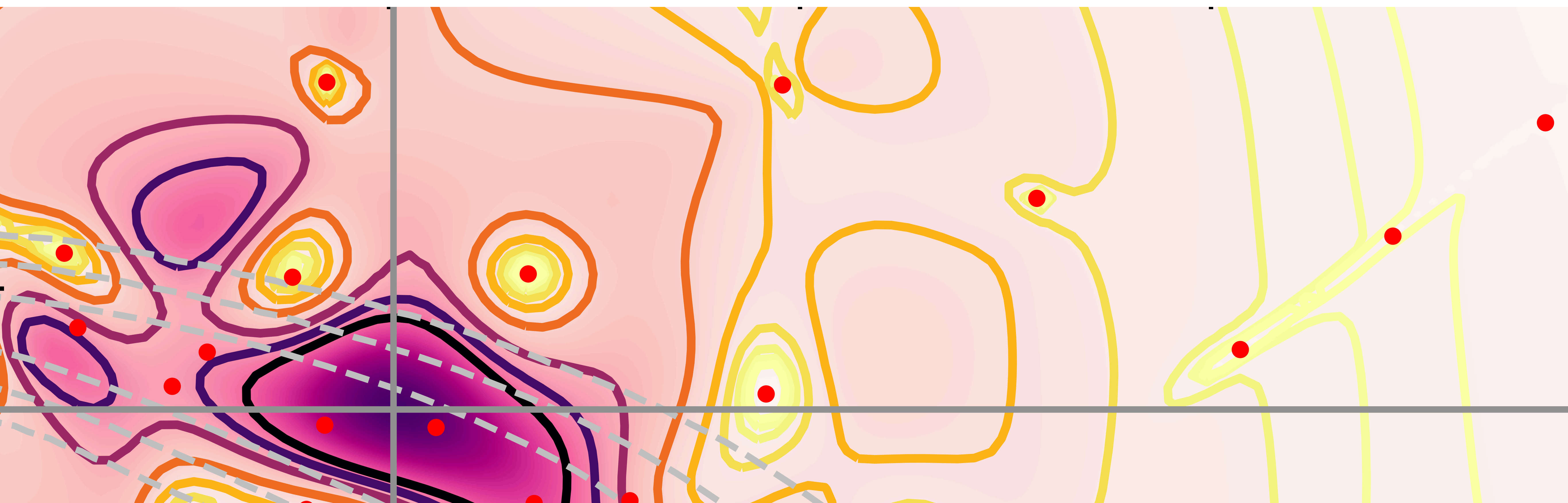
Leclercq (2018)

Batch Bayesian optimisation

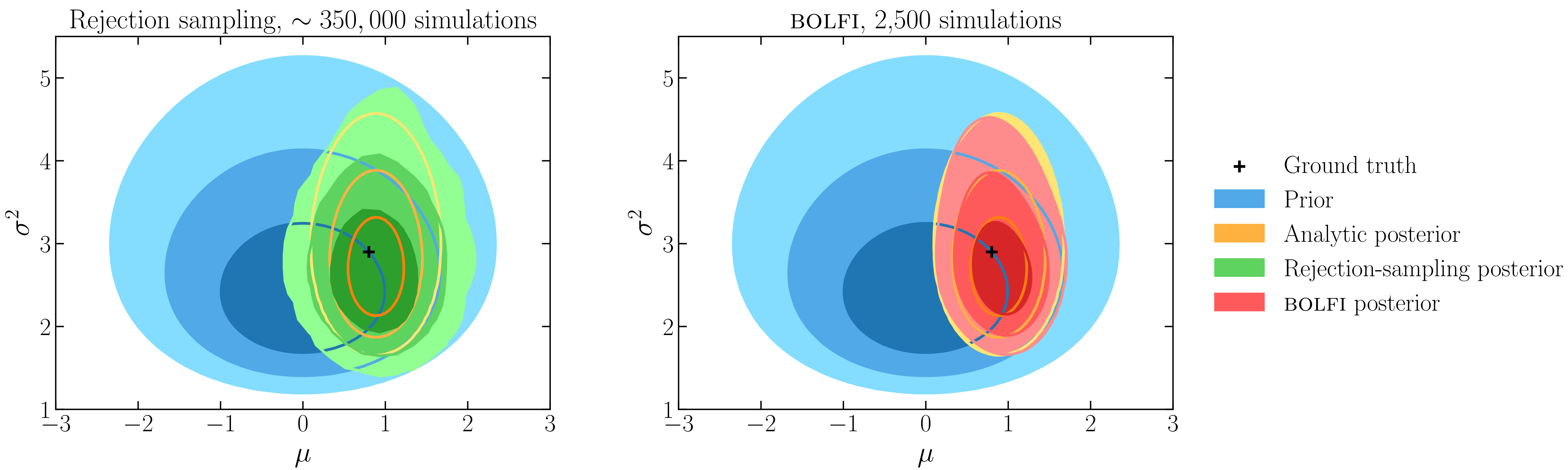
- Many simulations **too costly to run in serial**

- Must choose **batch of simulations simultaneously** from acquisition function

- **Can update uncertainty** as Gaussian process variance independent of output

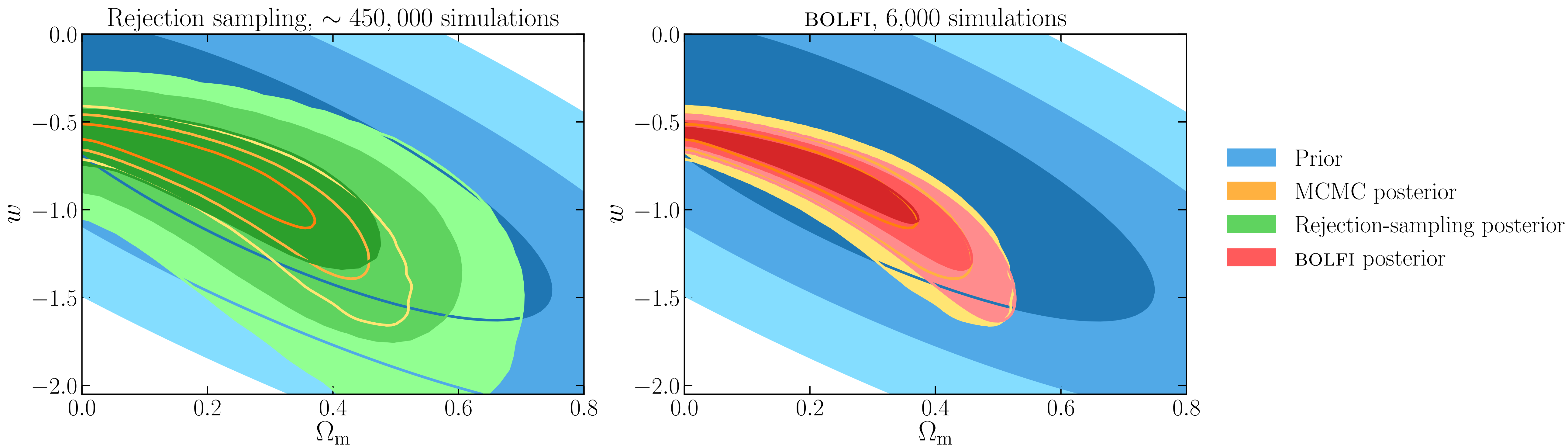


Bayesian optimisation is quicker



Leclercq (2018)

Bayesian optimisation is more accurate



Leclercq (2018)

Summary

- Gaussian process emulator gives **very general, probabilistic interpolation**
- Bayesian optimisation makes emulator **more efficient & more accurate**
- Solves problem of how to **compare data to costly simulations**