Universitetet i Stavanger

Fast Bayesian inference with Gaussian Processes

DSU Sydney 2022

arXiv:2211.02045

https://github.com/jonaselgammal/GPry

JONAS EL GAMMAL (UNIVERSITY OF STAVANGER)

WITH J. TORRADO, N. SCHÖNEBERG, C. FIDLER

1. Idea

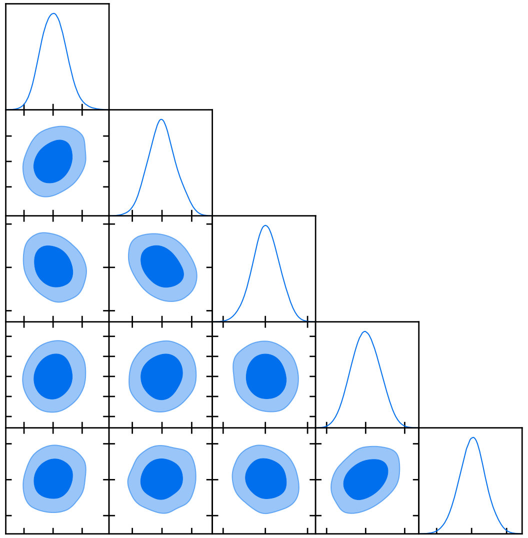
Parameters
- H_0
- Ω_b
- m_{DM}
- …

Theory code(s)

Data

(Log) Likelihood(s)

Monte Carlo



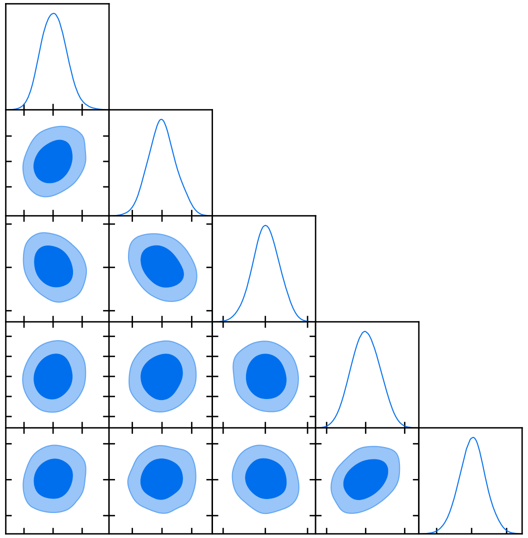
1. Idea

Parameters
- H_0
- Ω_{b}
- m_{DM}
- …

Theory code(s)

Data

(Log) Likelihood(s) → Monte Carlo →



Example: 8d $\sim 10^5$ samples for MCMC

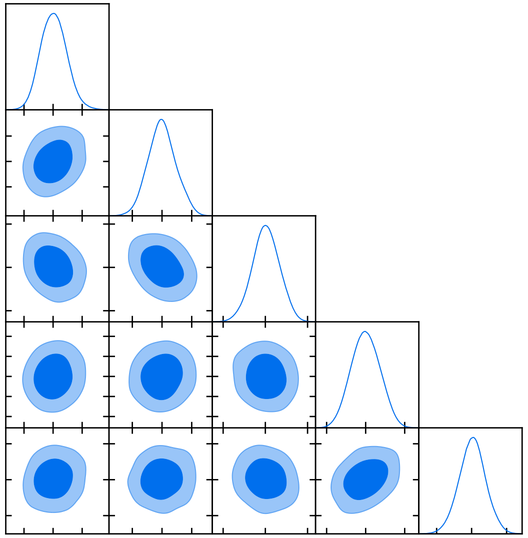
1. Idea

Parameters
- H_0
- Ω_b
- m_{DM}
- …

Theory code(s)

Data

(Log) Likelihood(s) → Monte Carlo →



Example: 8d $\sim 10^5$ samples for MCMC

Likelihood eval. time	Total time for inference
1 s	~ 1 day
1 min	
10 min	

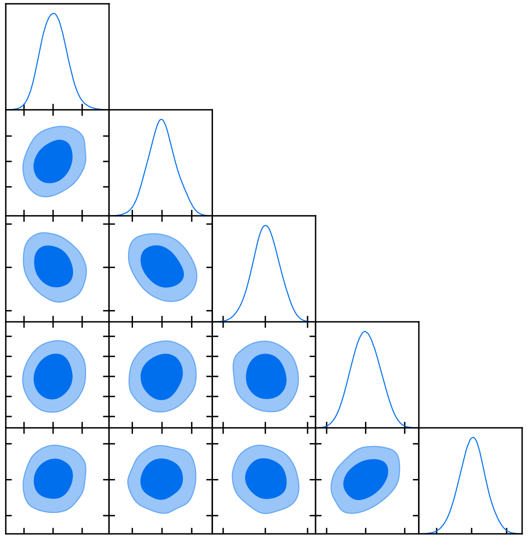
1. Idea

Parameters
- H_0
- Ω_b
- m_{DM}
- …

Theory code(s)

Data

(Log) Likelihood(s) → Monte Carlo →



Example: 8d $\sim 10^5$ samples for MCMC

Likelihood eval. time	Total time for inference
$1\ \mathrm{s}$	$\sim 1\ \mathrm{day}$
$1\ \mathrm{min}$	$\sim 1\ \mathrm{month}$
$10\ \mathrm{min}$	

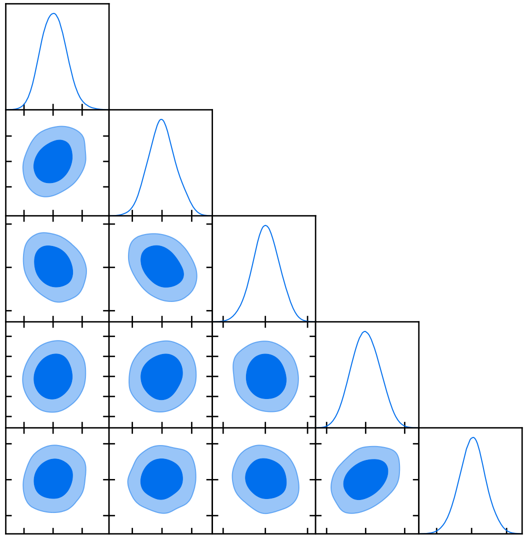
1. Idea

Parameters
- H_0
- Ω_b
- m_{DM}
- …

→ Theory code(s) →

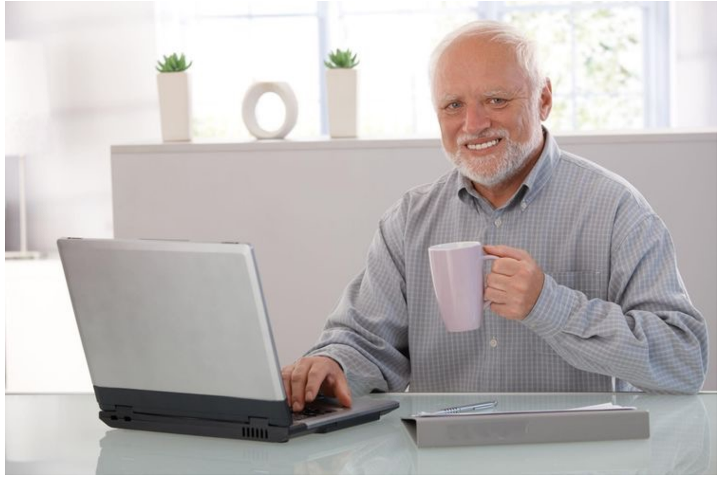
Data →

(Log) Likelihood(s) → Monte Carlo →

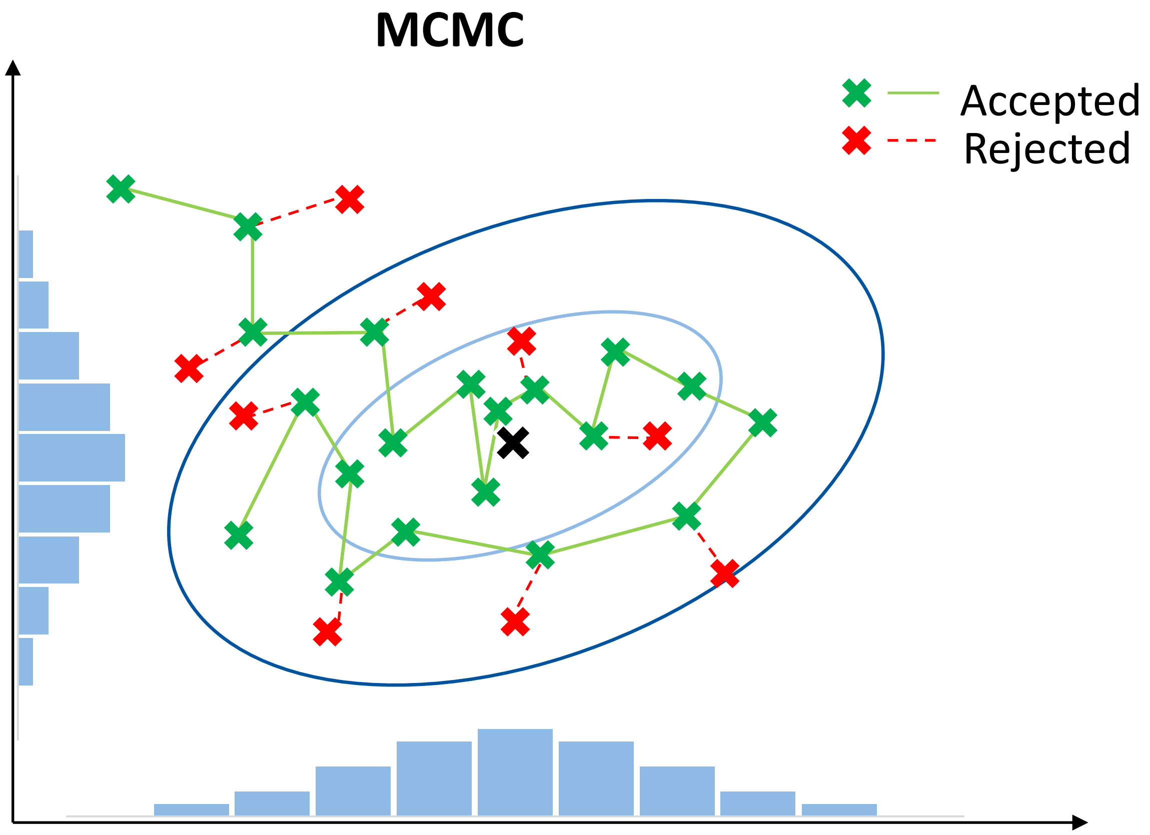


Example: 8d $\sim 10^5$ samples for MCMC

Likelihood eval. time	Total time for inference
1 s	$\sim$ 1 day
1 min	$\sim$ 1 month
10 min	$\sim$ 1 year

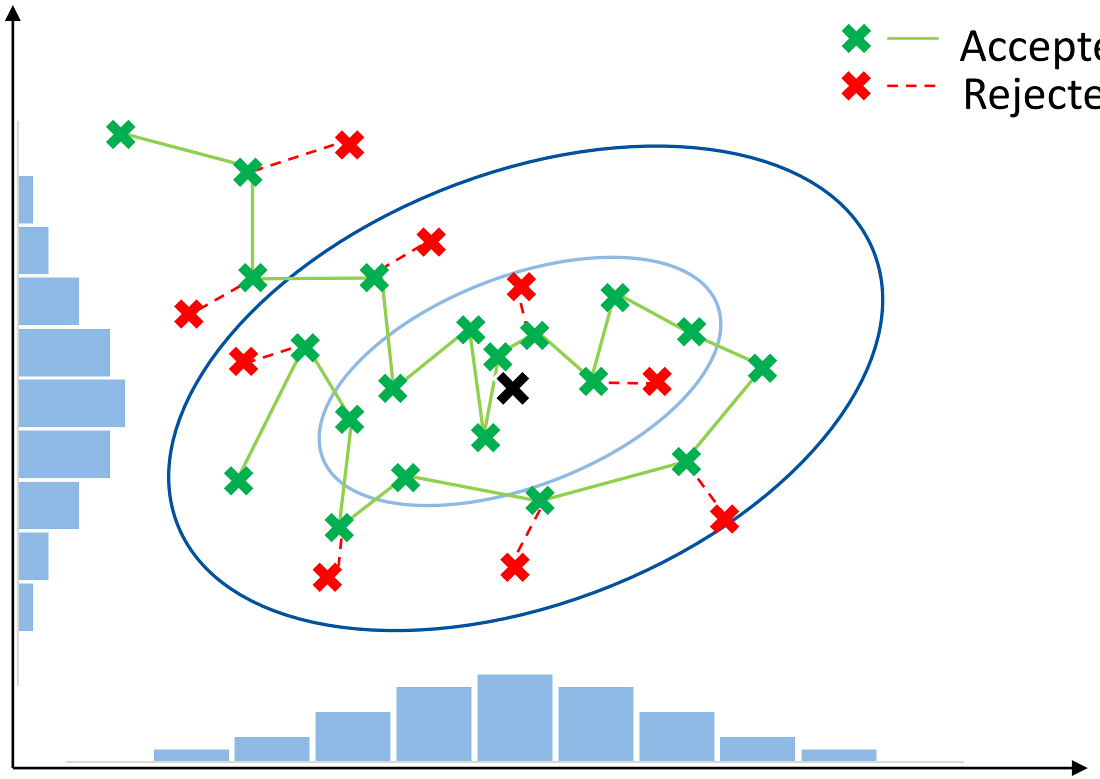


1. Idea

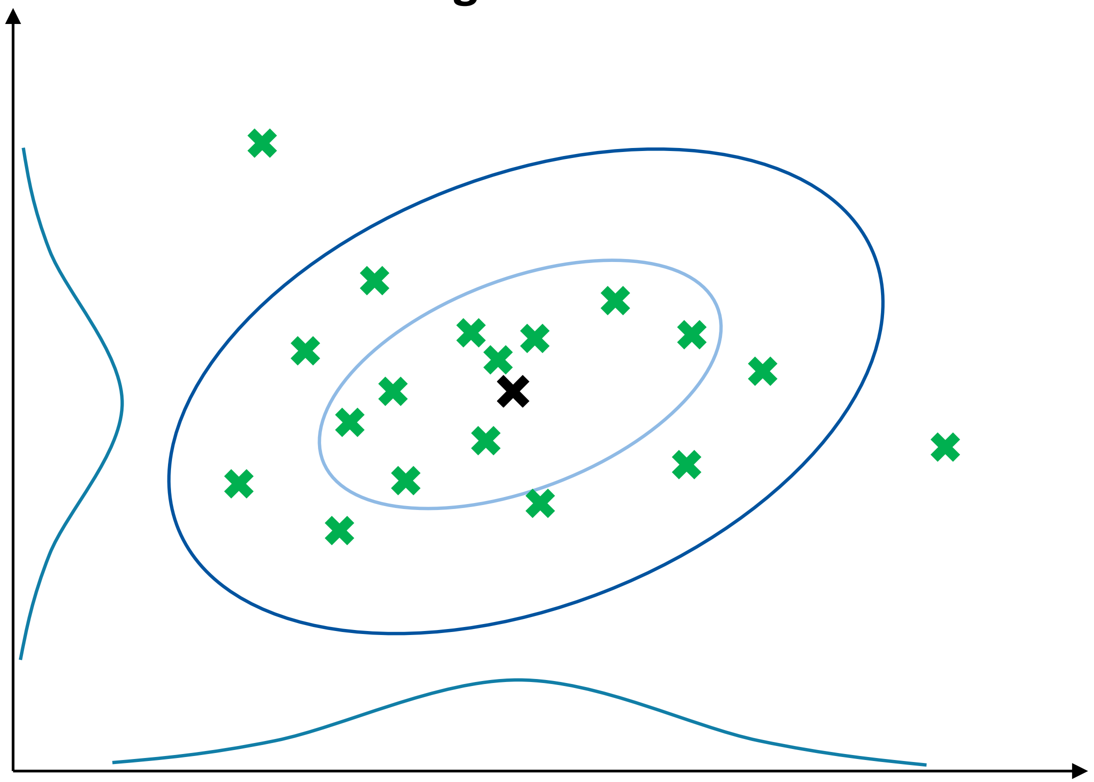


1. Idea

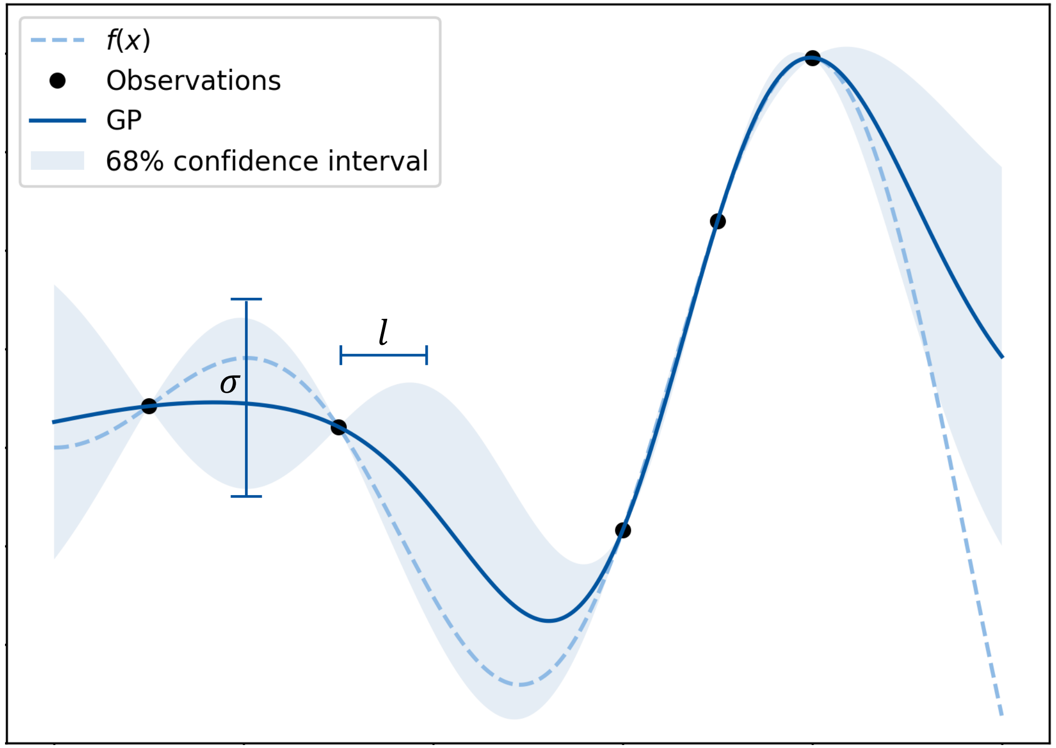
MCMC



Surrogate model



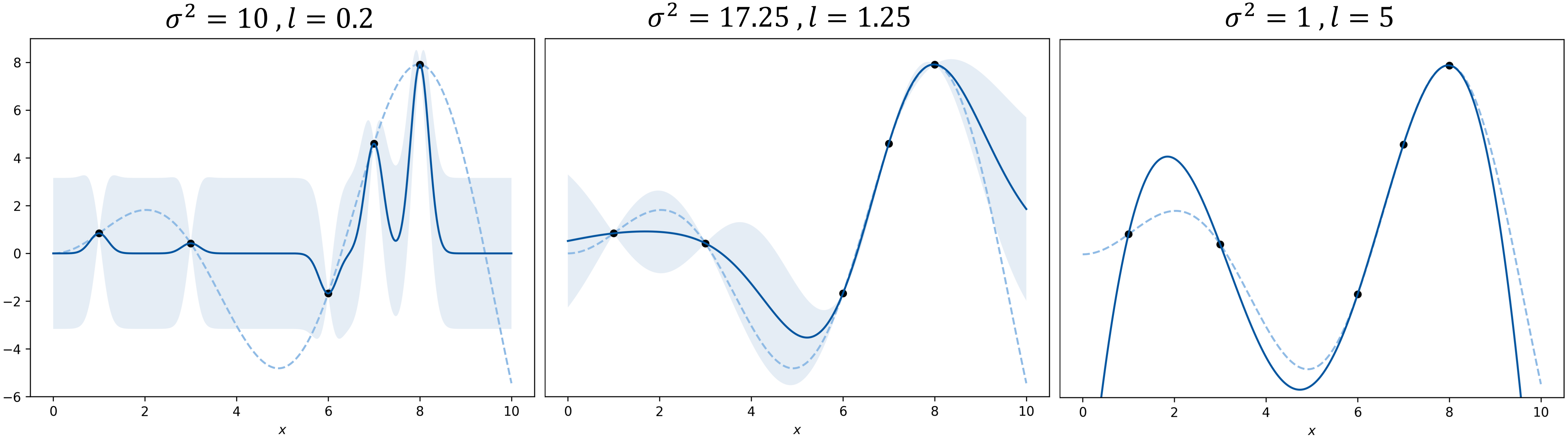
2. Gaussian Process Surrogate



$$k(x, x') = \sigma^2 \cdot \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

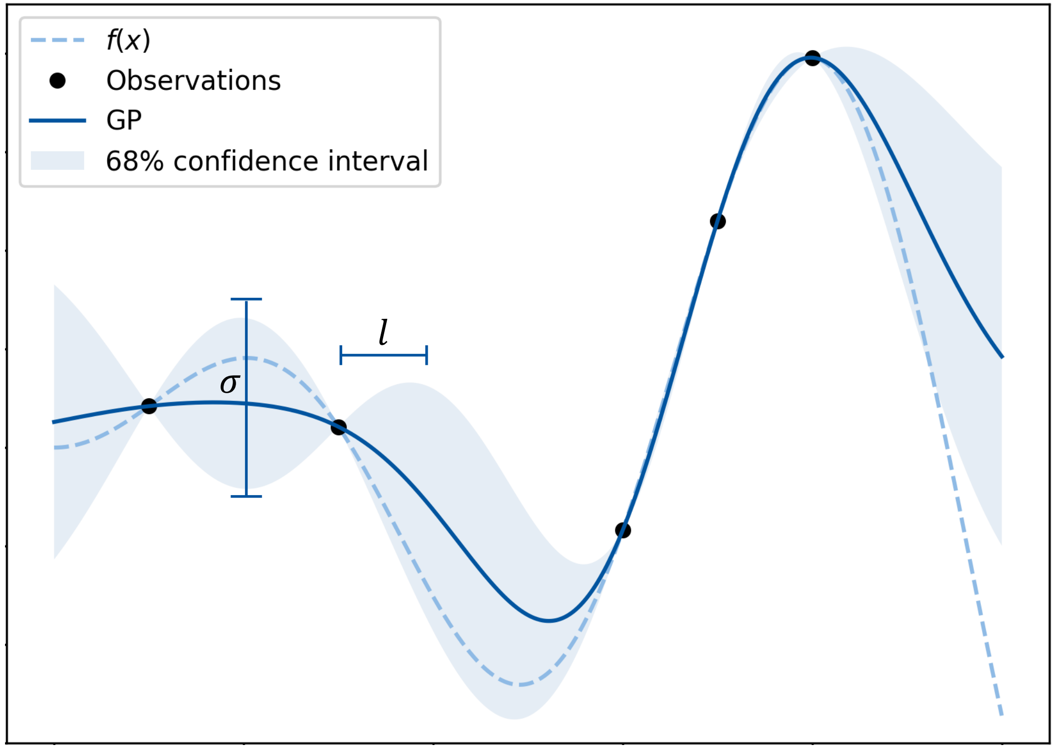
2. Gaussian Process Surrogate

$$k(x, x') = \sigma^2 \cdot \exp\left(-\frac{|x - x'|^2}{2l^2}\right)$$

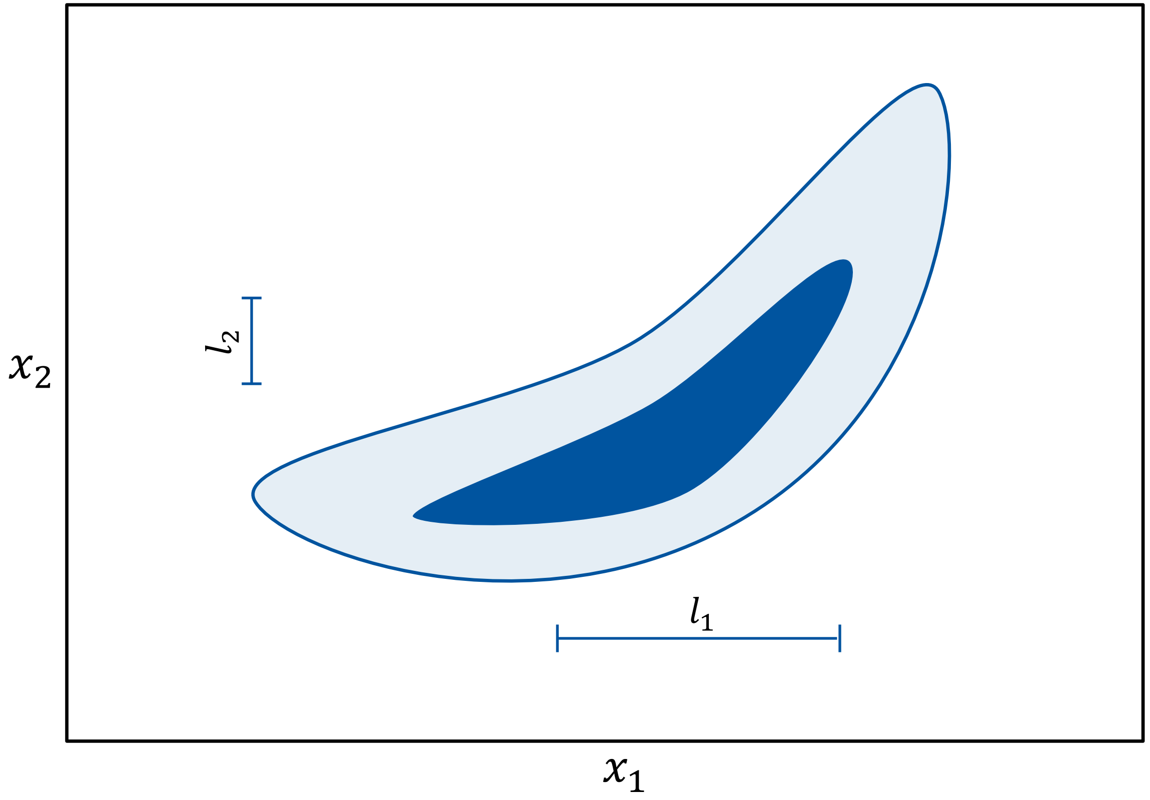


Can use MAP to get the best estimate

2. Gaussian Process Surrogate

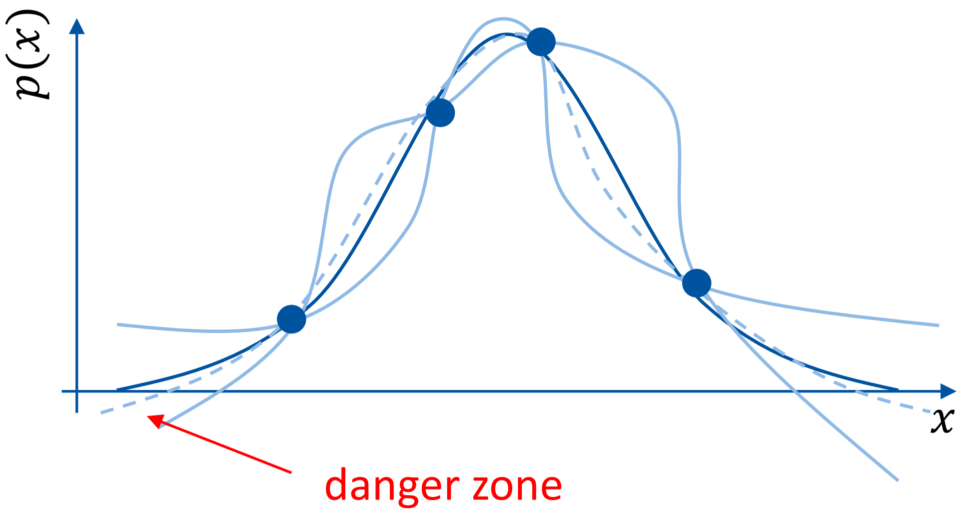


$$k(x, x') = \sigma^2 \cdot \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

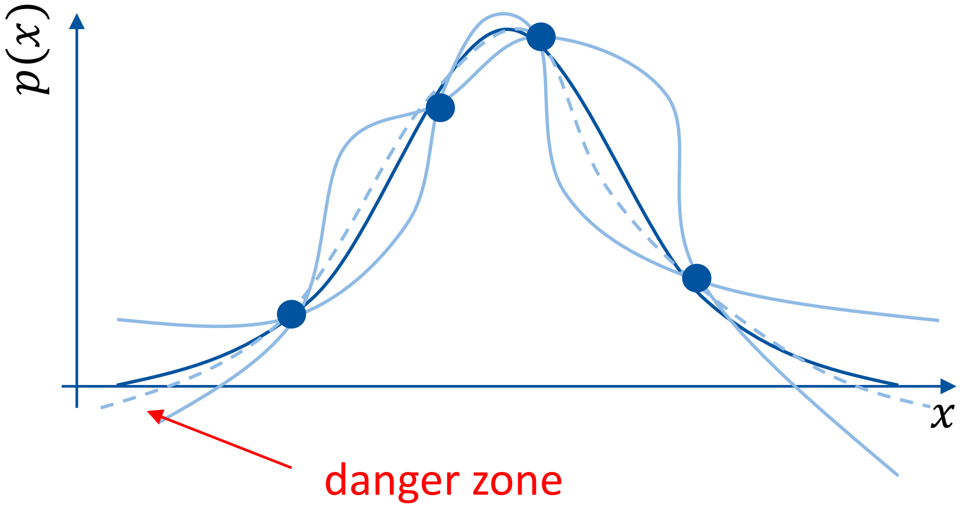


$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \cdot \exp\left(-\sum \frac{(x_i - x_i')^2}{2l_i^2}\right)$$

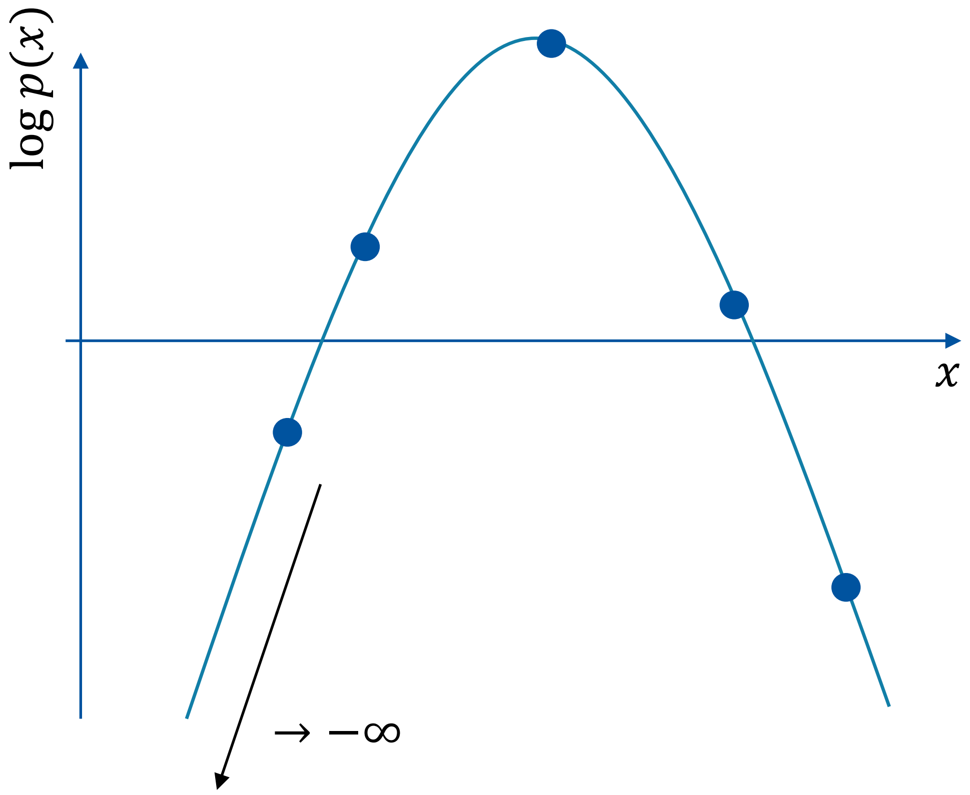
3. Region of interest



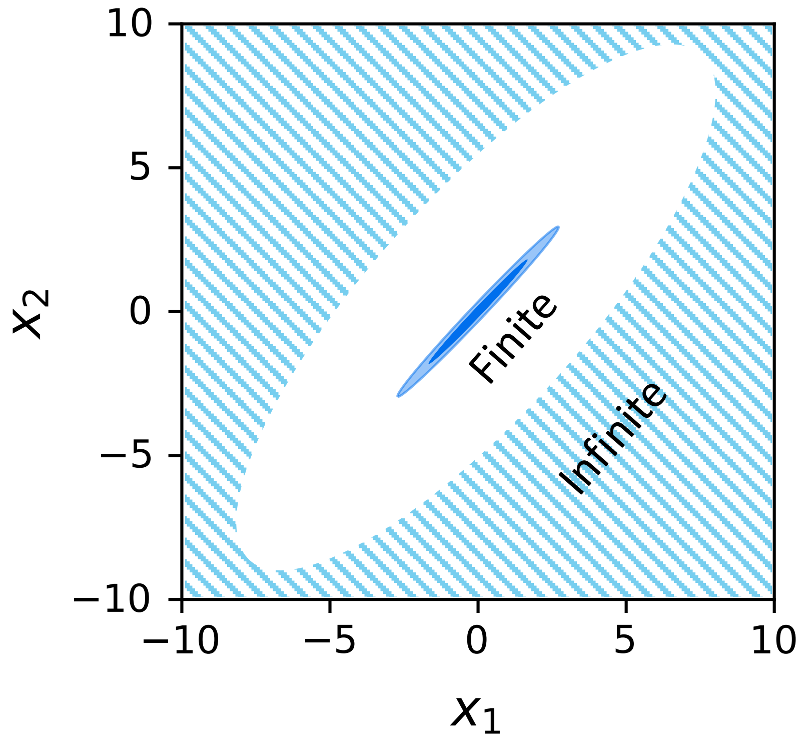
3. Region of interest



⇒Interpolate **log-posterior** to enforce positivity

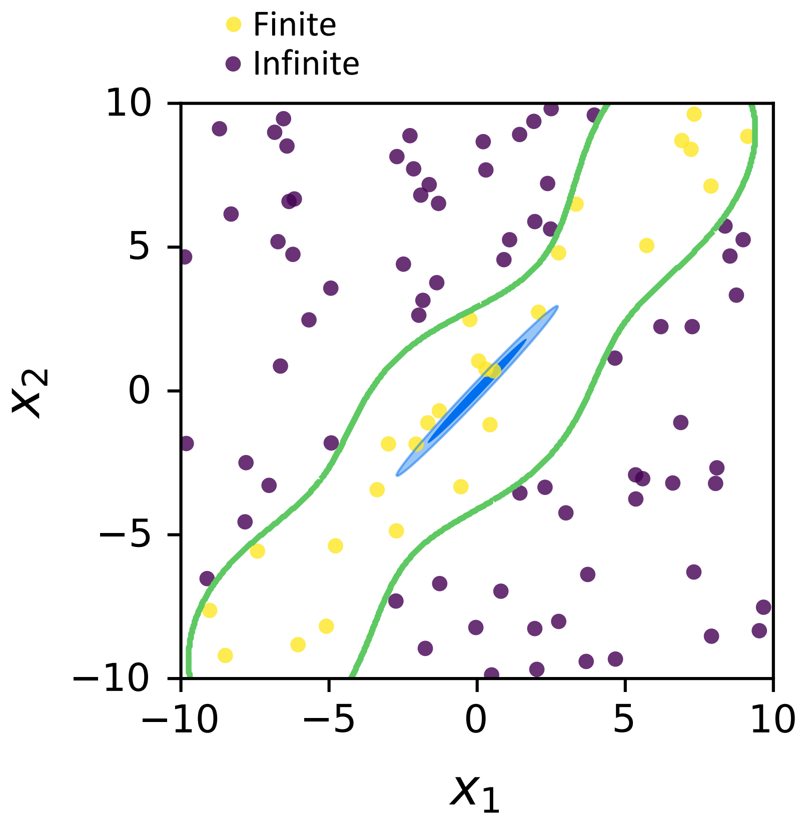


3. Region of interest



Solution: SVM Classifier

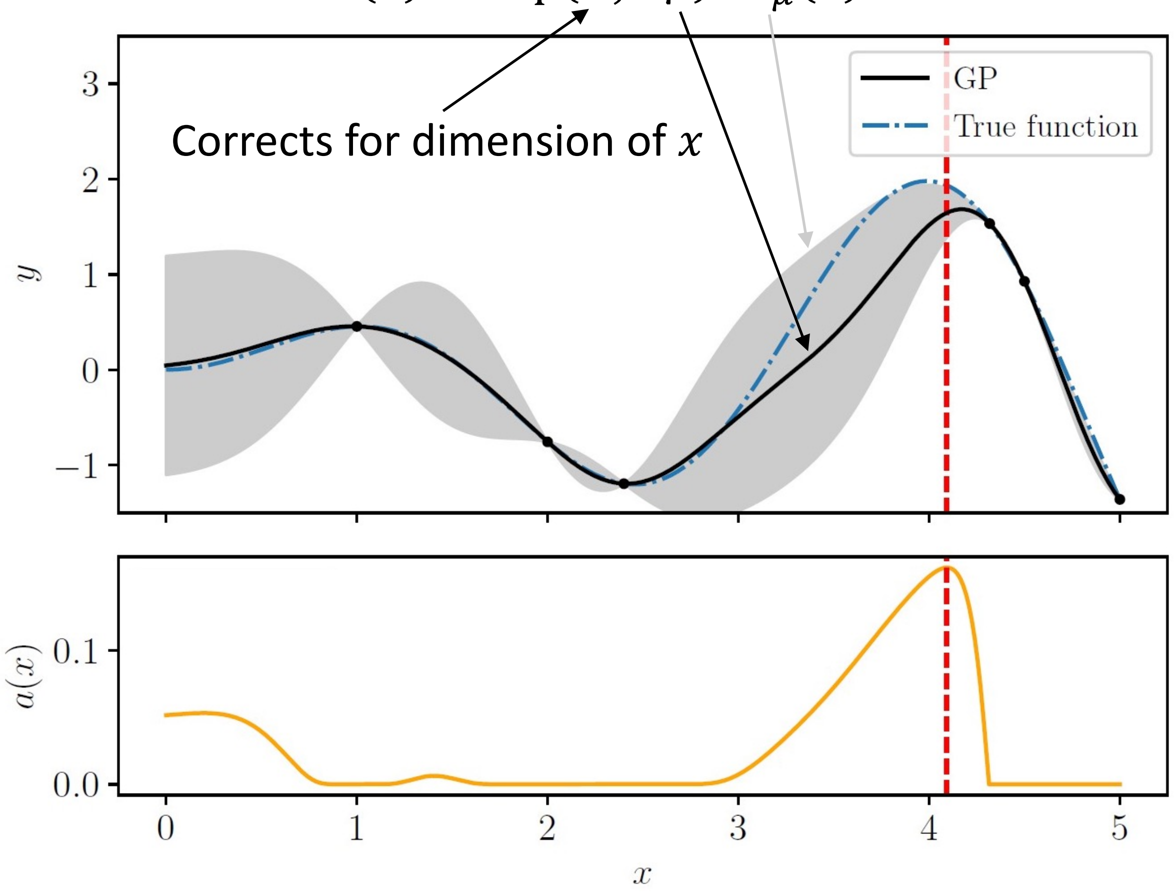
Multiply μ with $-\infty$ where SVM classifies as “infinite”



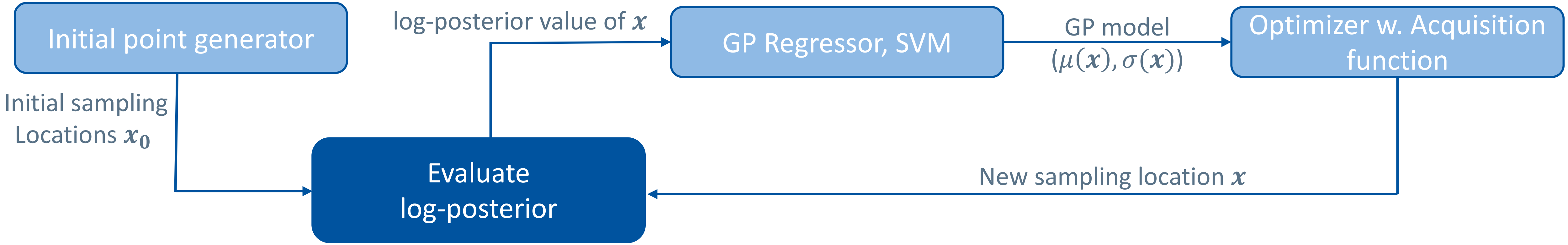
4. Active sampling

Propose samples by maximizing an **acquisition function**

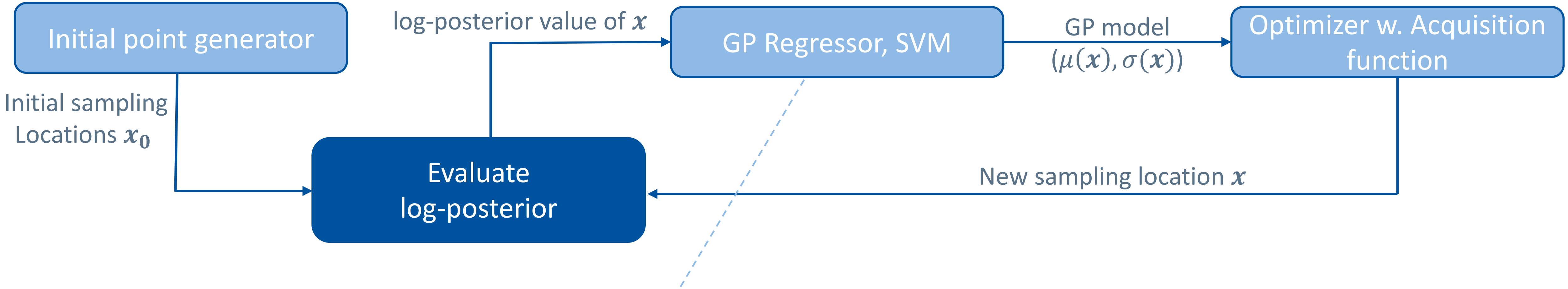
$$a(x) = \exp(2\zeta \cdot \bar{\mu}) \cdot \sigma_{\bar{\mu}}(x)$$

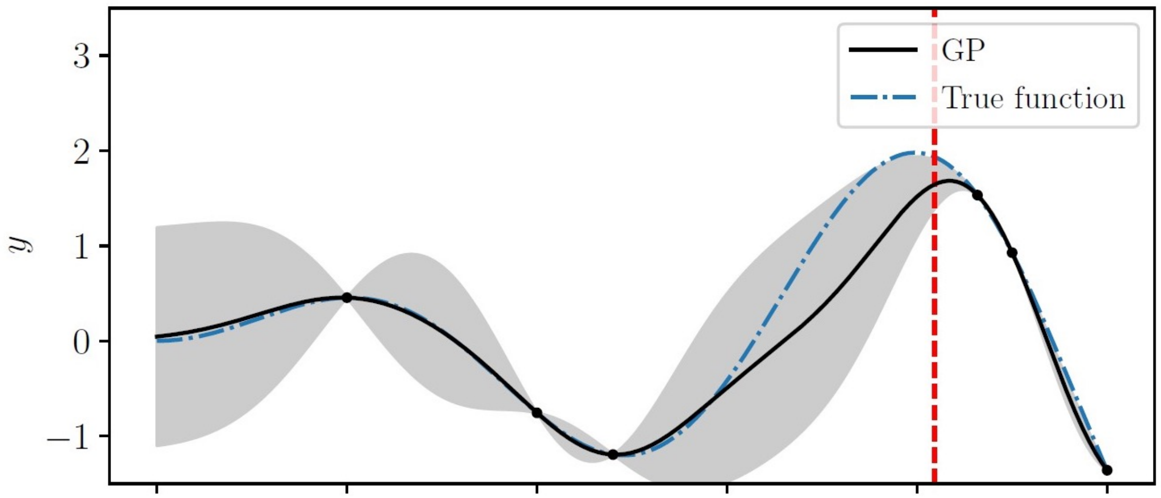


5. The Algorithm

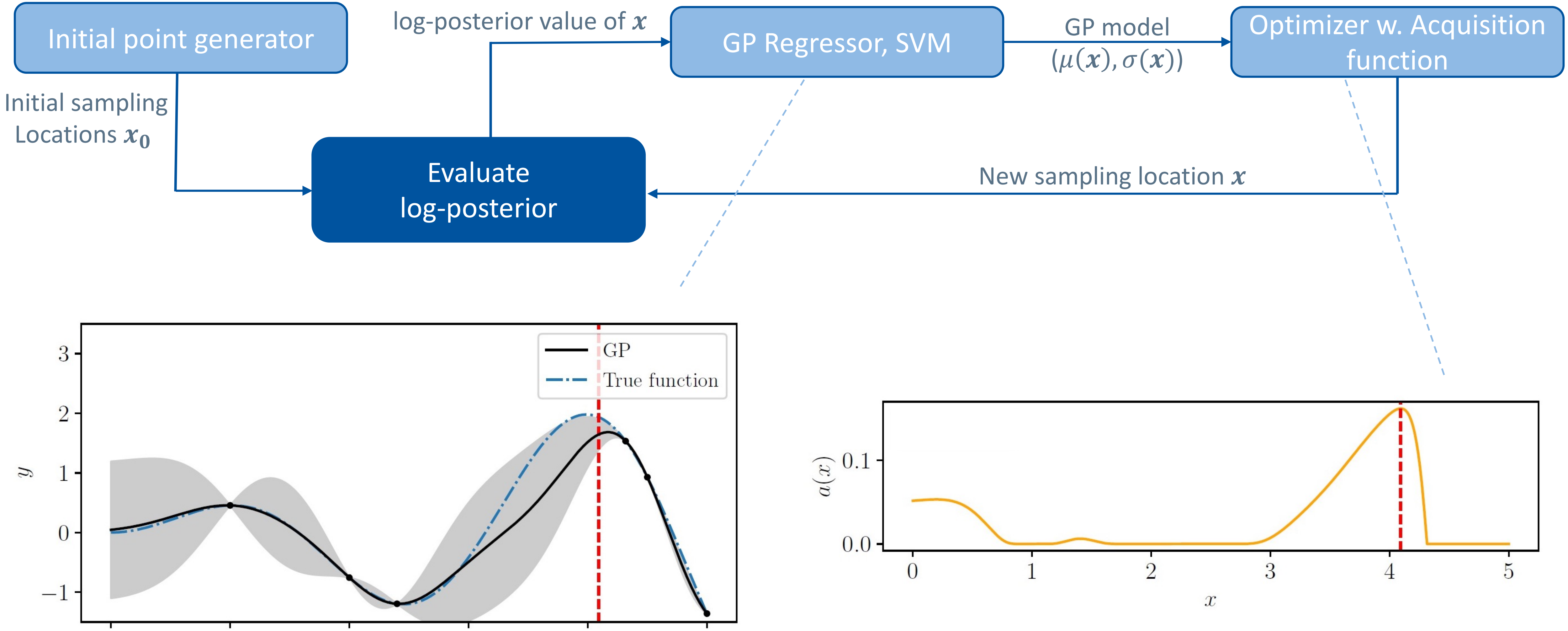


5. The Algorithm

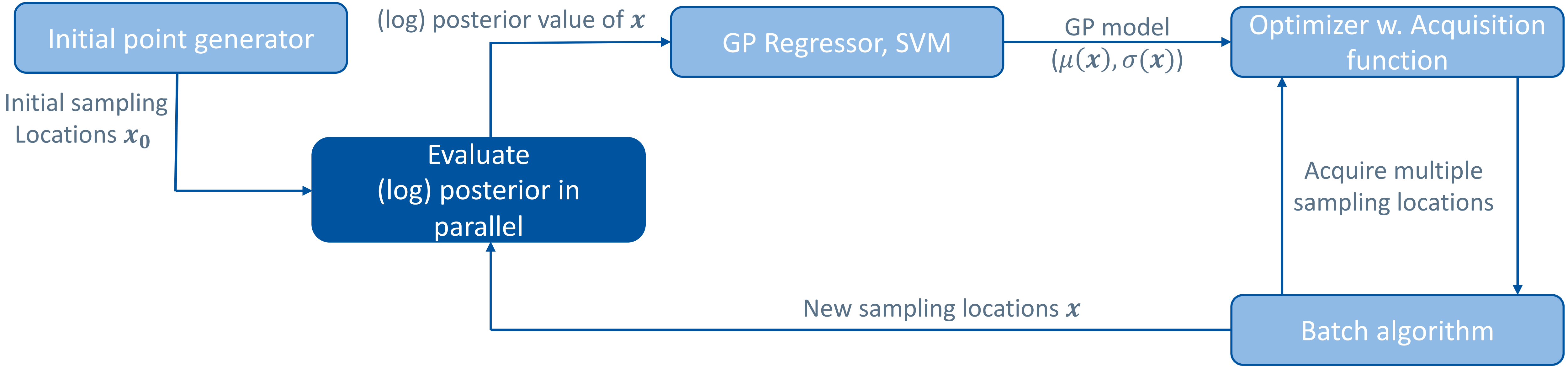




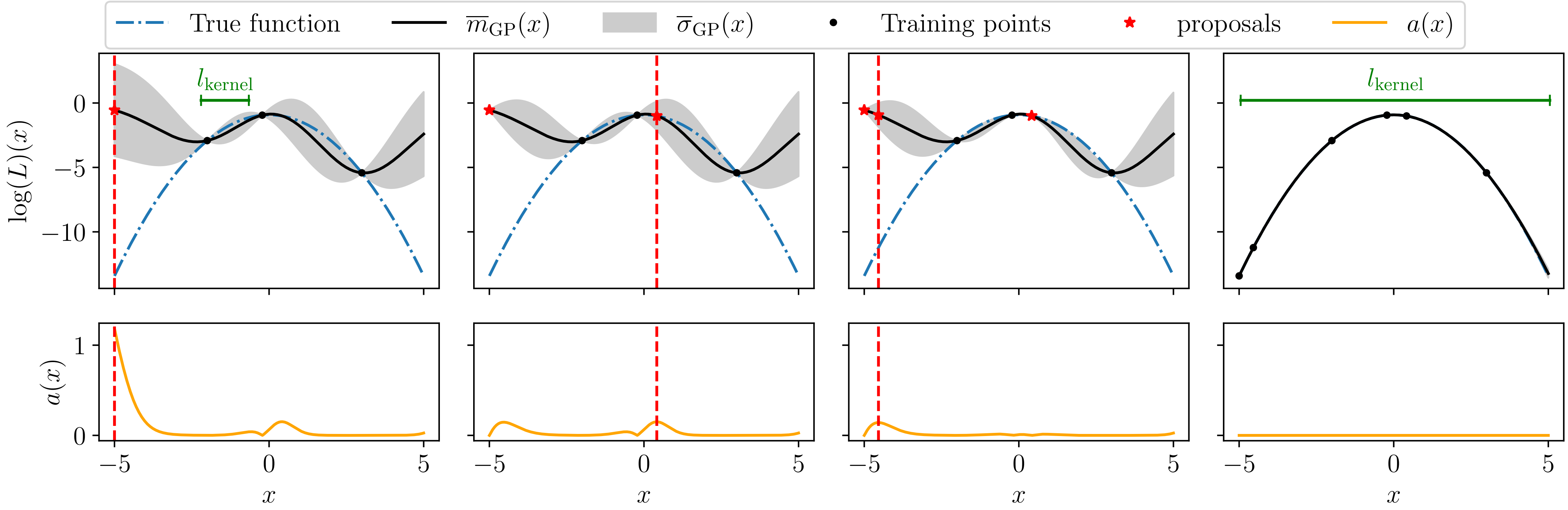
5. The Algorithm



5. The Algorithm



5. The Algorithm



5. The Algorithm

Initial point generator

Initial sampling Locations $\boldsymbol{x}_0$

Evaluate (log) posterior in parallel

(log) posterior value of $\boldsymbol{x}$

GP Regressor, SVM

GP model $(\mu(\boldsymbol{x}), \sigma(\boldsymbol{x}))$

Optimizer w. Acquisition function

Acquire multiple sampling locations

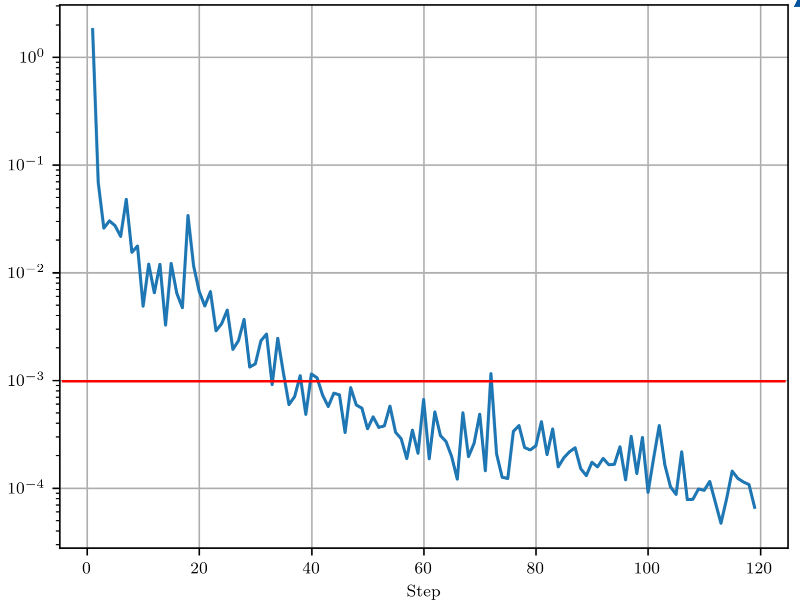
Batch algorithm

New sampling locations $\boldsymbol{x}$

Break loop

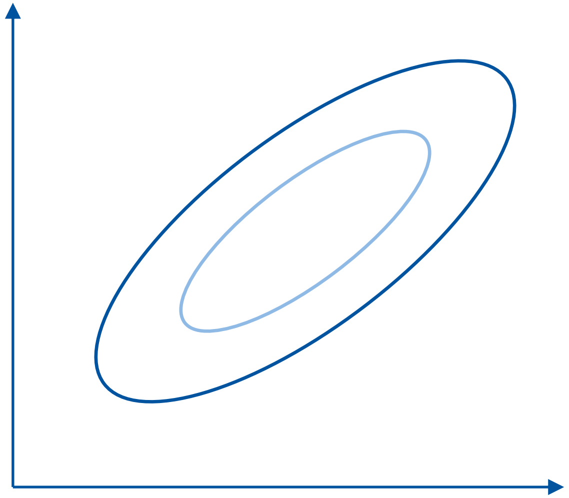
Break if

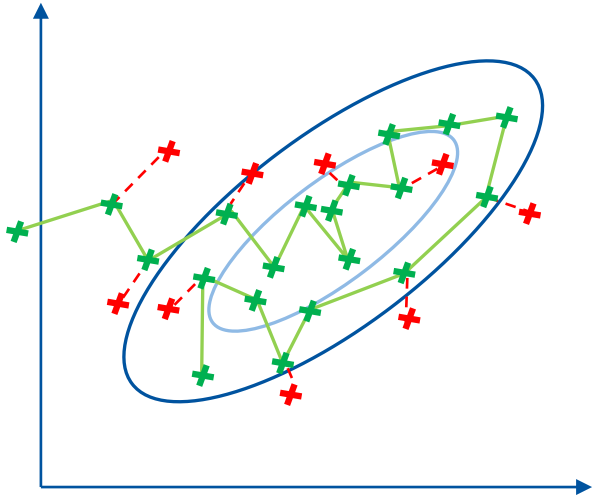
$$|\log p_{\text{true}} - \log p_{\text{predict}}| < \alpha + \beta * |\log p_{\max} - \log p_{\text{true}}|$$

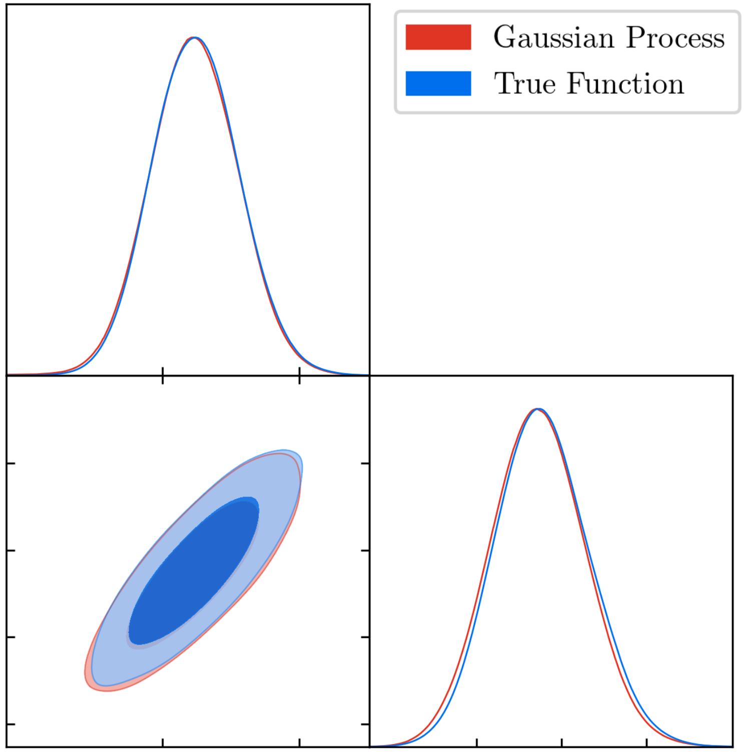


6. Marginalised quantities

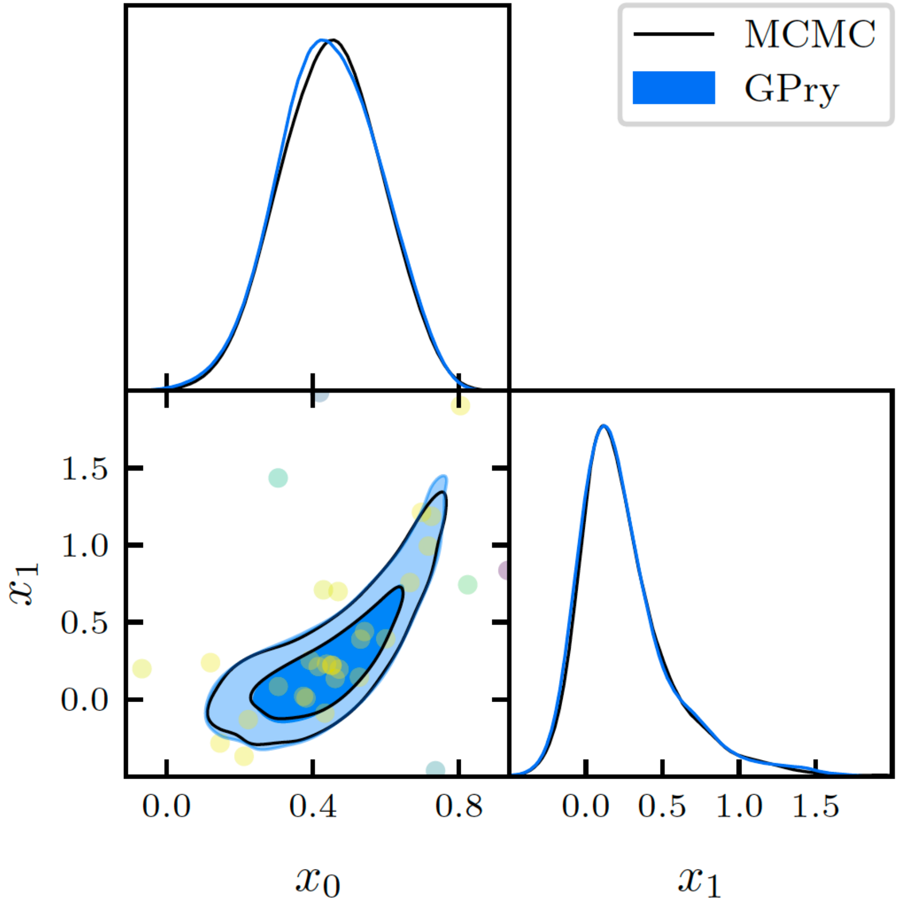
GP model $(\mu(x), \sigma(x))$ → MC sampler → Plotting etc.



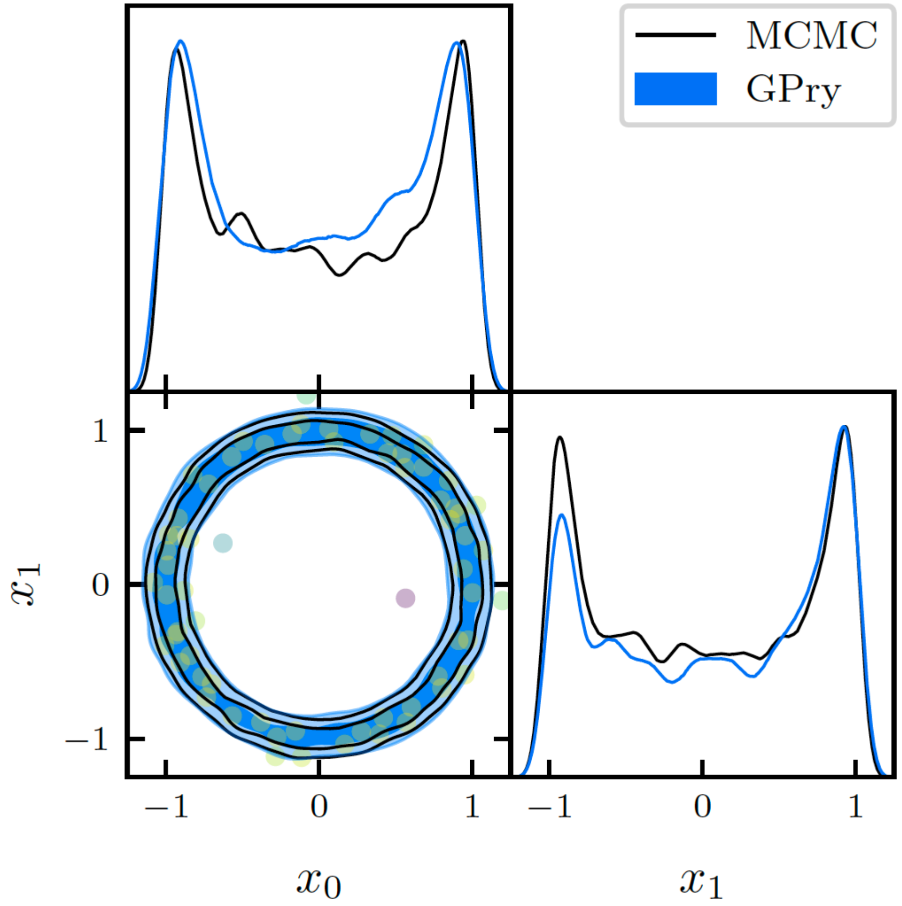




7. Experiments

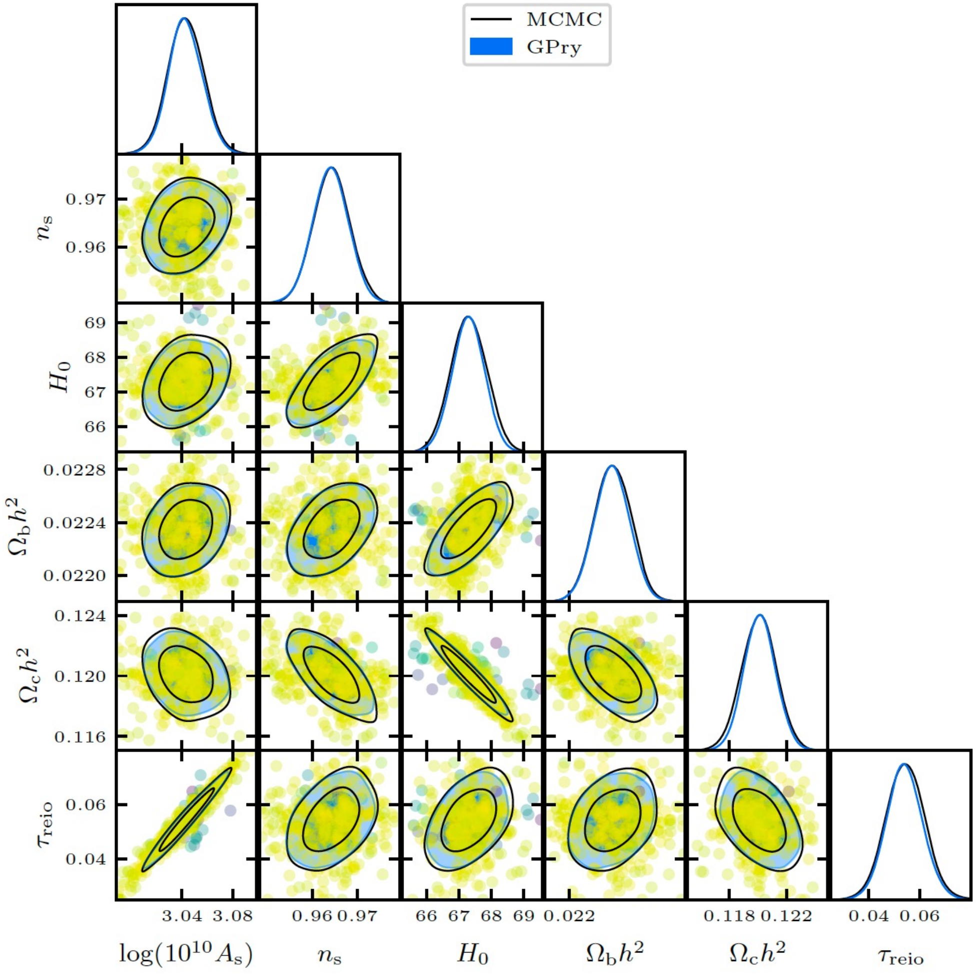


40 posterior evaluations

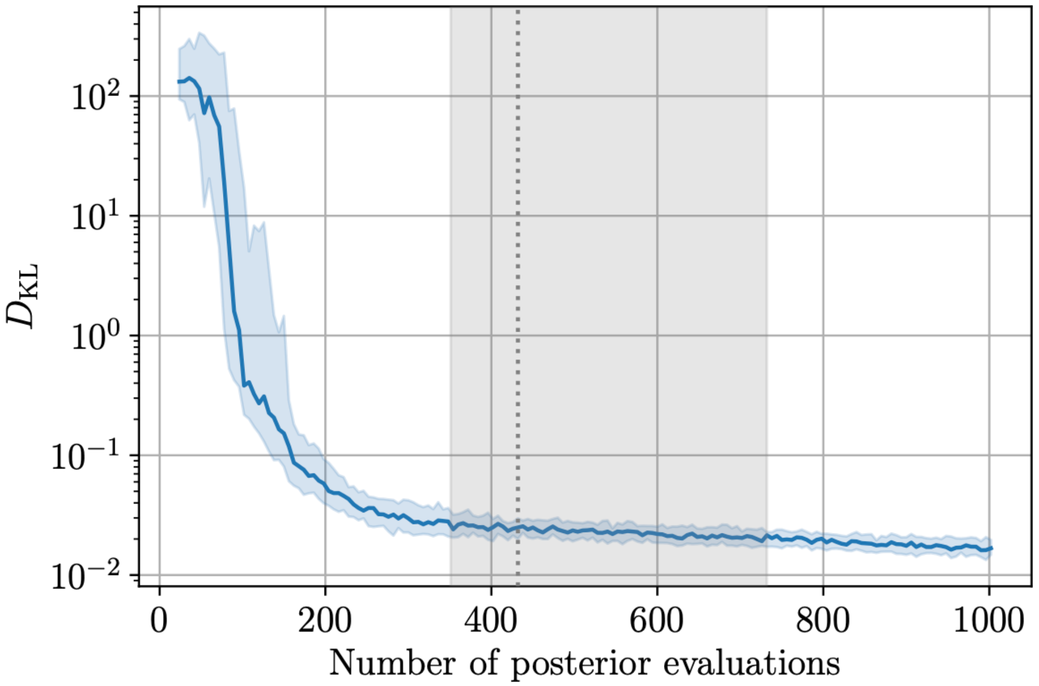


68 posterior evaluations

7. Experiments

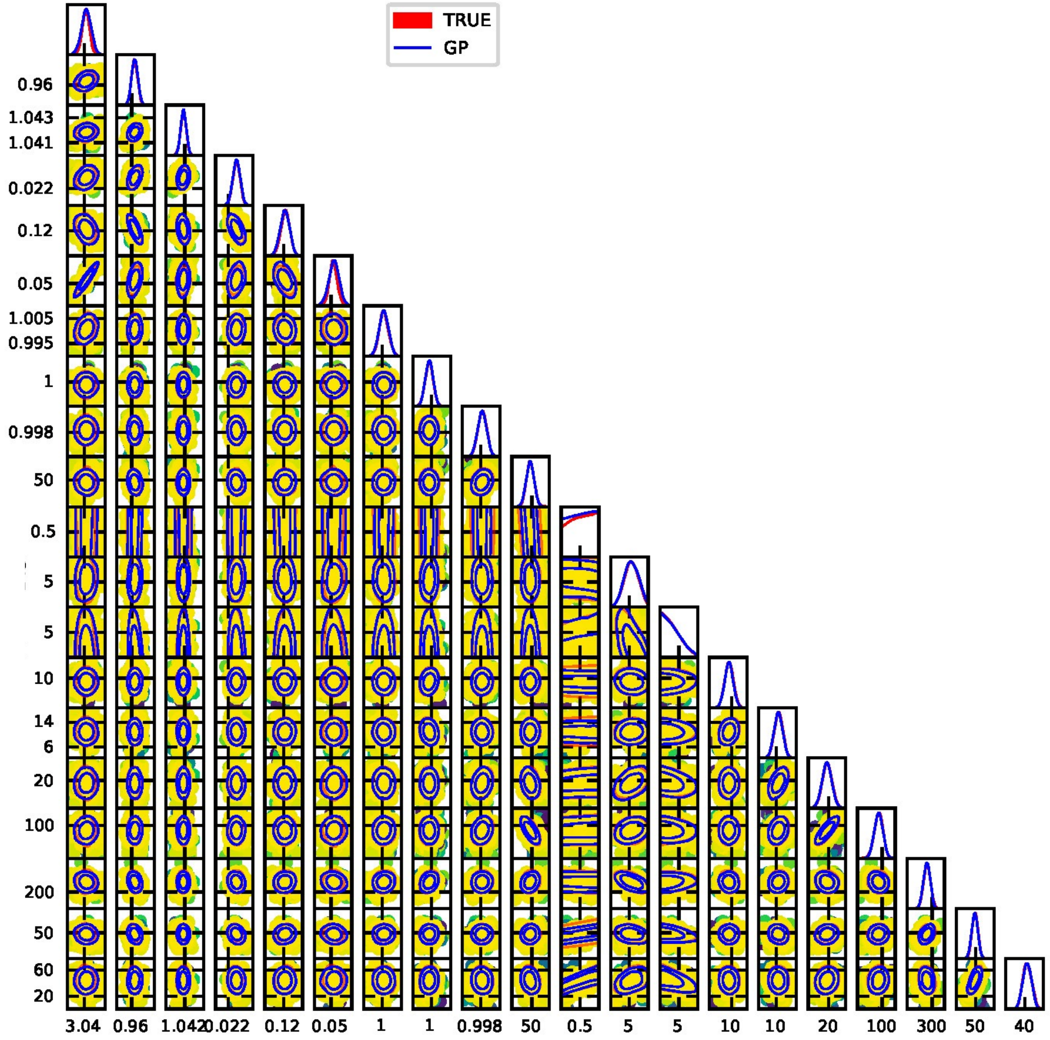


420 posterior evaluations

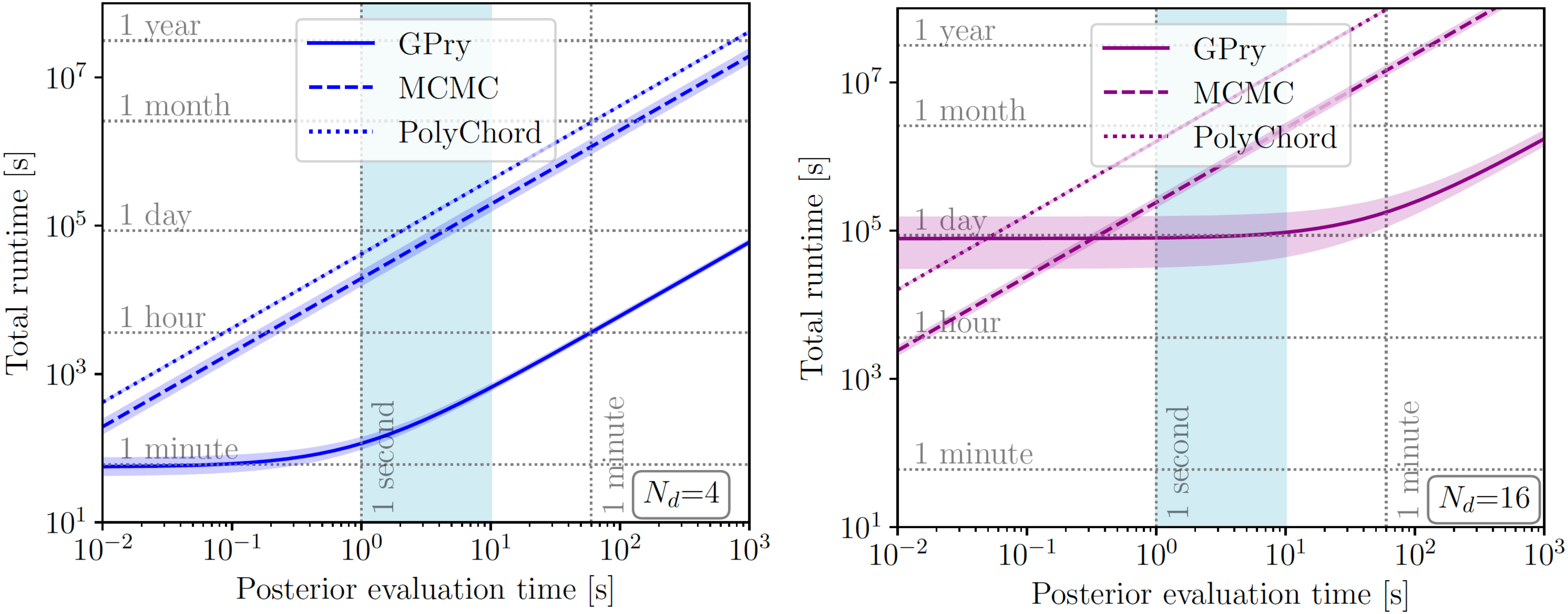


Universitetet i Stavanger

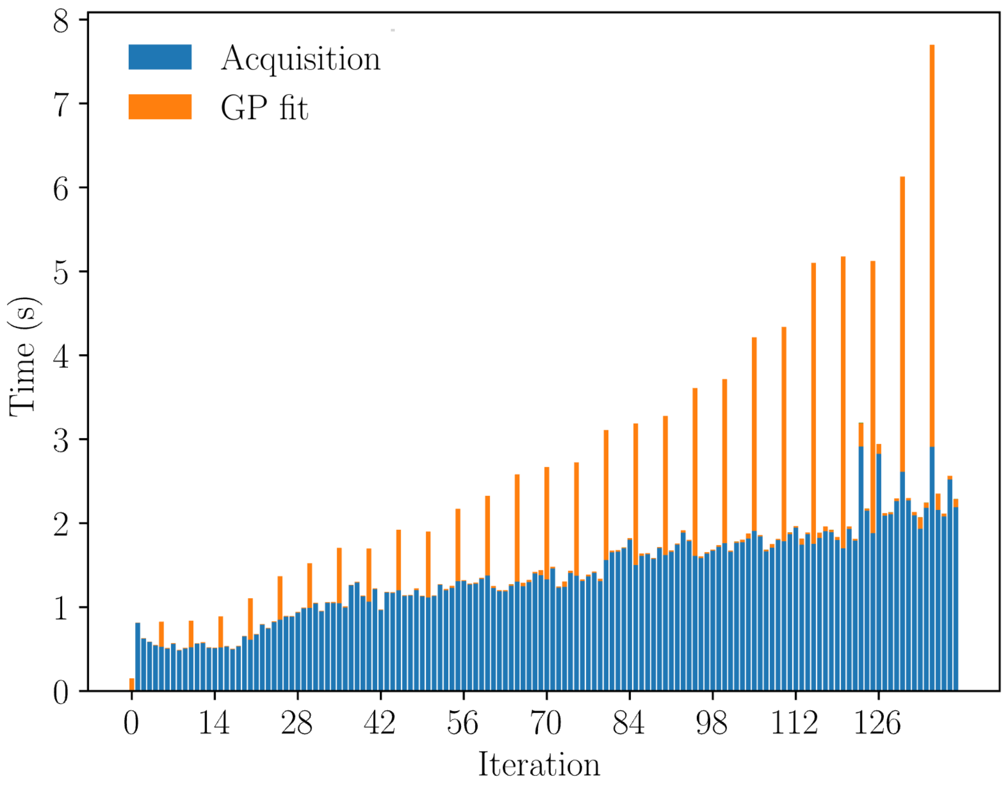
7. Experiments



8. Performance

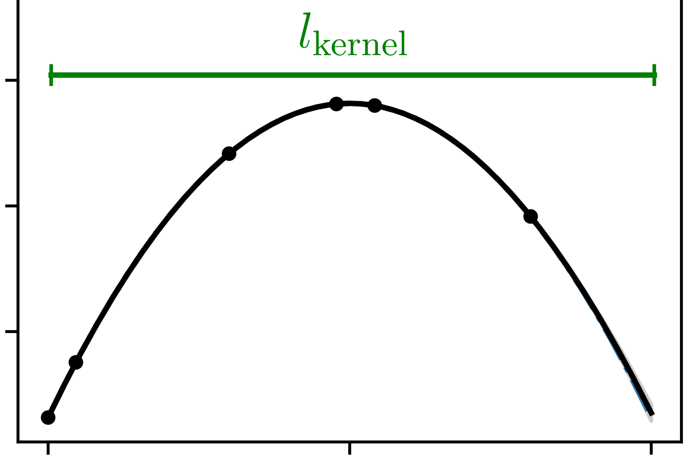


9. Limitations



Overhead

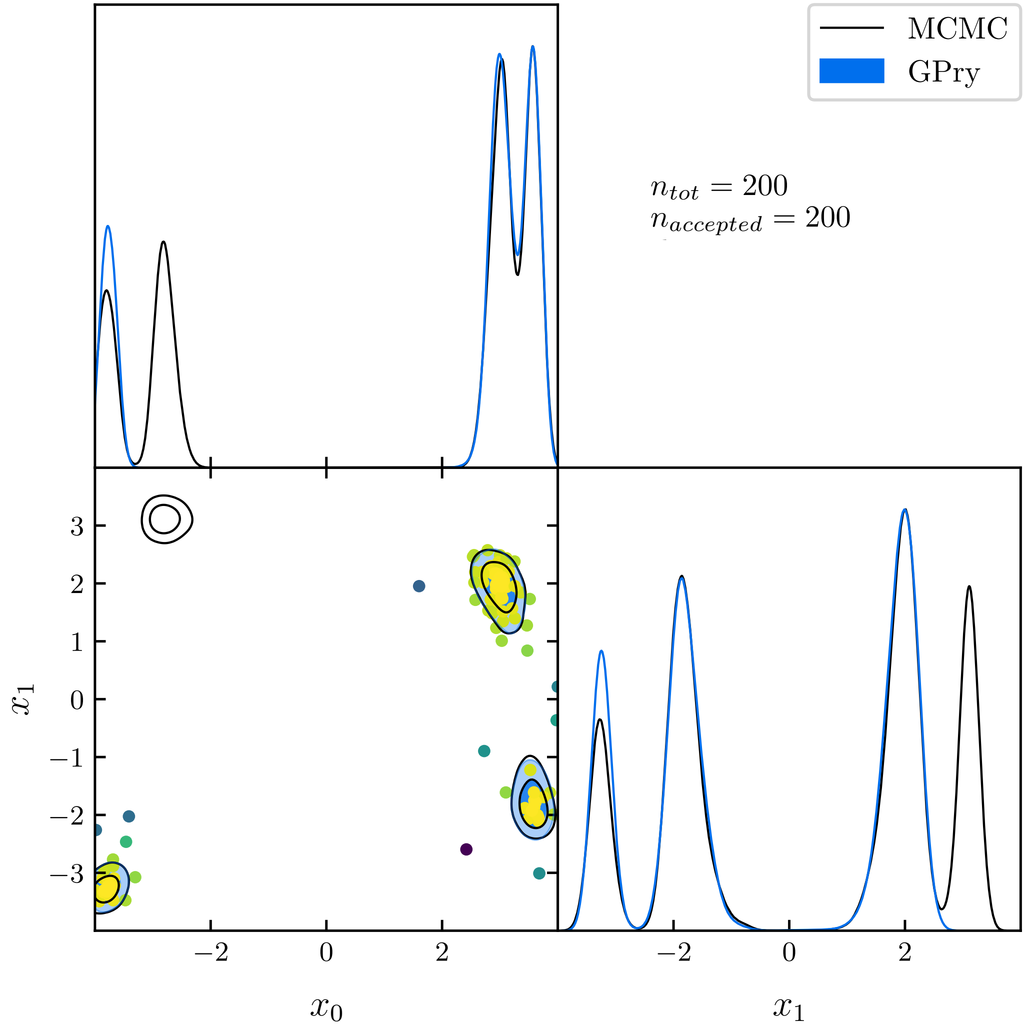
9. Limitations



Overhead

"Overfitting"

9. Limitations



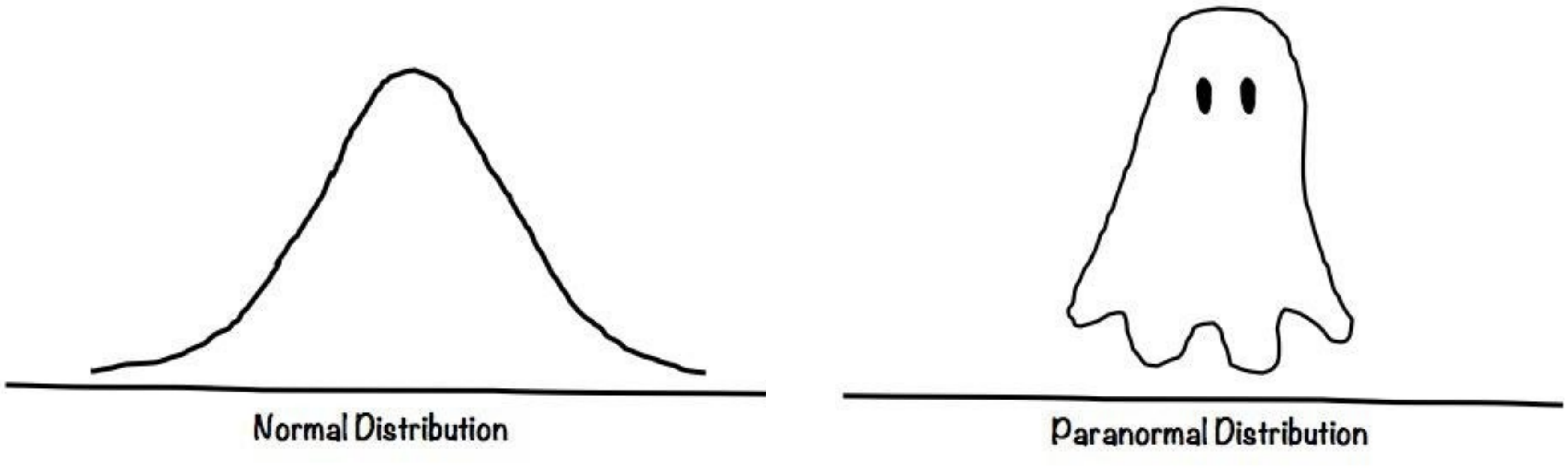
Overhead

“Overfitting”

Multimodality

We work on solving those problems…

Thank you!



https://www.memedroid.com/memes/detail/3518248/Normal-vs-paranormal-distribution

Backup

Gaussian Process Regression

Assume that

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}),\ k(\boldsymbol{x}, \boldsymbol{x}'))$$

(usually $m(\boldsymbol{x}) = 0 \ \forall \ \boldsymbol{x}$)

Then $f_* = f(\boldsymbol{x}_*)$ drawn from the GP given the measurements $\boldsymbol{y} = \boldsymbol{f}(X)$ are distributed as

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} k(X,X) & k(X,X_*) \\ k(X_*,X) & k(X_*,X_*) \end{pmatrix}\right)$$

This means, that test functions are distributed according to

$$\boldsymbol{f}_* | X, \boldsymbol{y}, X_* \sim \mathcal{N}(\bar{\boldsymbol{f}}_*, \mathrm{cov}(\boldsymbol{f}_*))$$

with

$$\bar{f}_* = E[f_* | X, y, X_*] = K(X_*, X) \cdot K^{-1}(X,X) y$$

$$\mathrm{cov}(f_*) = K(X_*, X_*) \cdot K^{-1}(X,X) K(X, X_*)$$

Computational complexity scales with n^3

Gaussian Process Regression

Marginalize: $p(\boldsymbol{y}|X) = \int p(\boldsymbol{y}|\boldsymbol{f}, X)\, p(\boldsymbol{f}|X)\, \mathrm{d}\boldsymbol{f}$

$= \mathcal{N}(\boldsymbol{f}, 0) = \boldsymbol{f}$

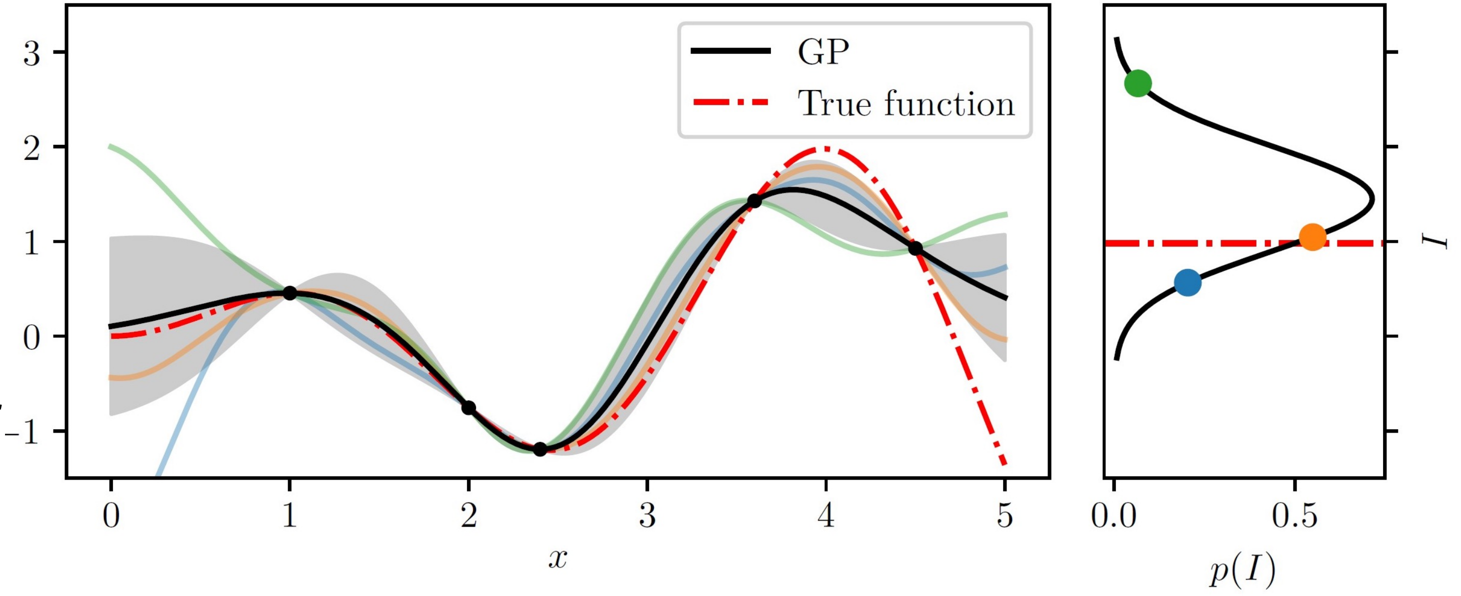
$= \mathcal{N}(0, K), \qquad K_{ij} = k(x_i, x_j)$

$$\Rightarrow \log\big(p(y|X)\big) = -\frac{1}{2} y^T K^{-1} y - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi)$$

Can use MLE to get the best estimate

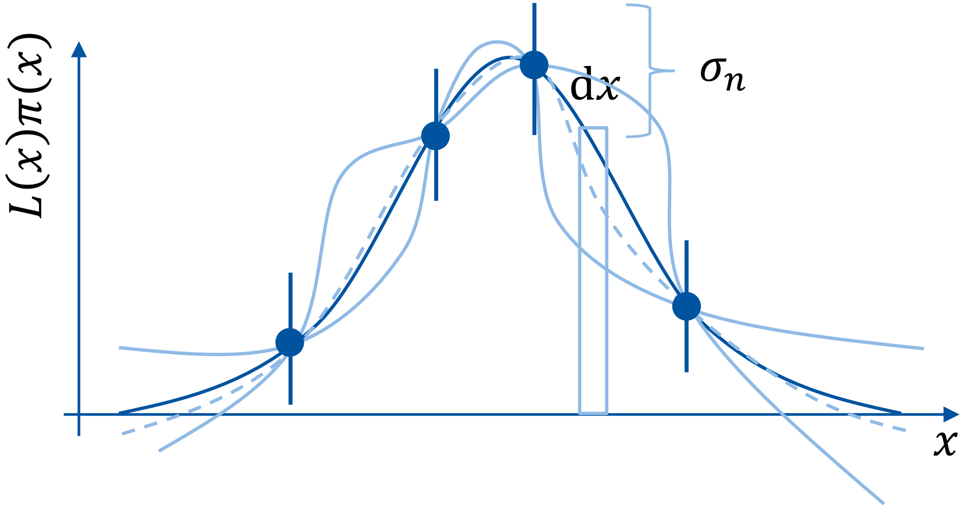
Active sampling

- To get marginalised quantities we want to integrate
  $$\int L(x)\pi(x)\,\mathrm{d}x$$
- With a GP we can get a model for
  $$L(x)\pi(x) \sim \mathcal{GP}(0, k(x, x'))$$
- We can integrate that model by integrating $\int \mu(x)\,\mathrm{d}x = \int \bar{f}(x)\,\mathrm{d}x$
- We can use $\mu(x)$ and $\sigma(x) = \sqrt{\mathrm{cov}(f_*(x,x))}$ to find the next most informative point to sample



Active sampling

$\Rightarrow$ At each step maximize an **acquisition function**



$L(x)\pi(x)$ is always positive

$$\Rightarrow a(x) = \mu(x) \cdot \sigma(x)$$

$L(x)\pi(x)$ has high dynamic range

$\Rightarrow$ Sample log-posterior:

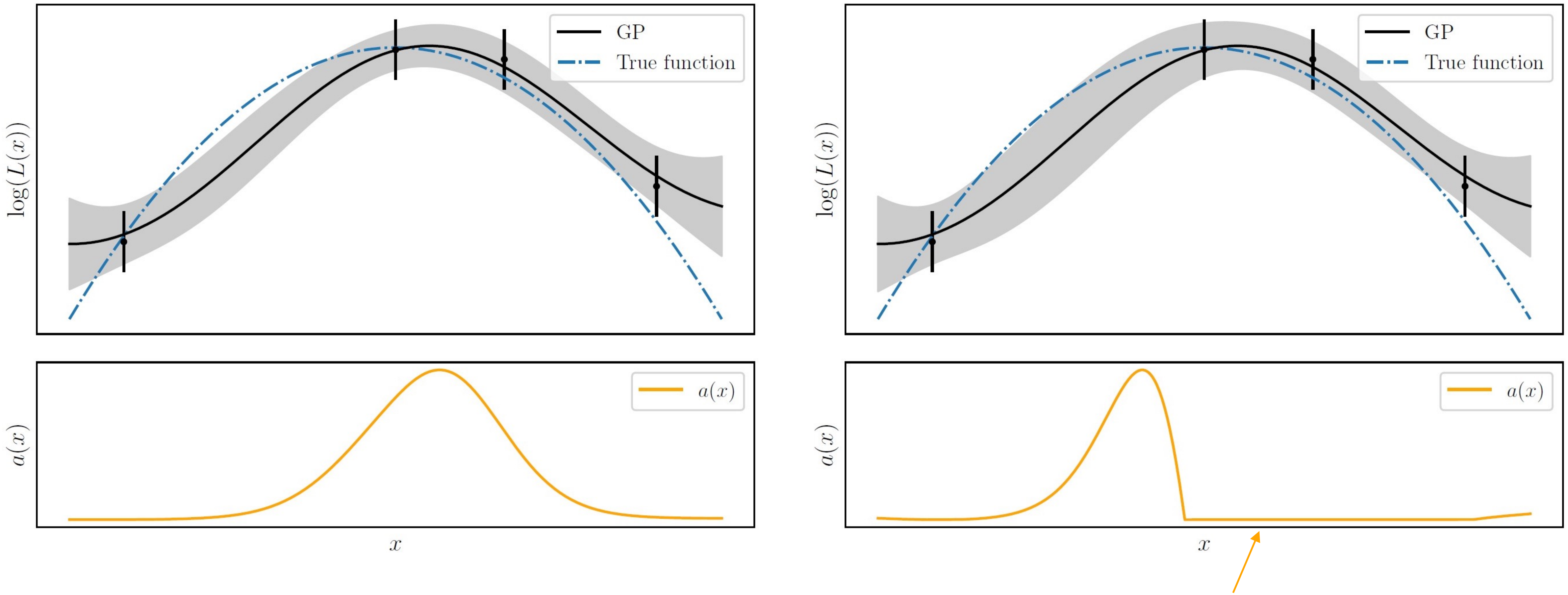
$$a(x) = \exp(2 \cdot \bar{\mu}) \cdot \sigma_{\bar{\mu}}(x)$$

$\bar{\mu}$ = Mean of GP fit to log-posterior

Correction factor ζ and statistical noise σ_n

$$a(x) = \exp(2\zeta \cdot \bar{\mu}) \cdot (\sigma_{\bar{\mu}}(x) - \sigma_n)$$

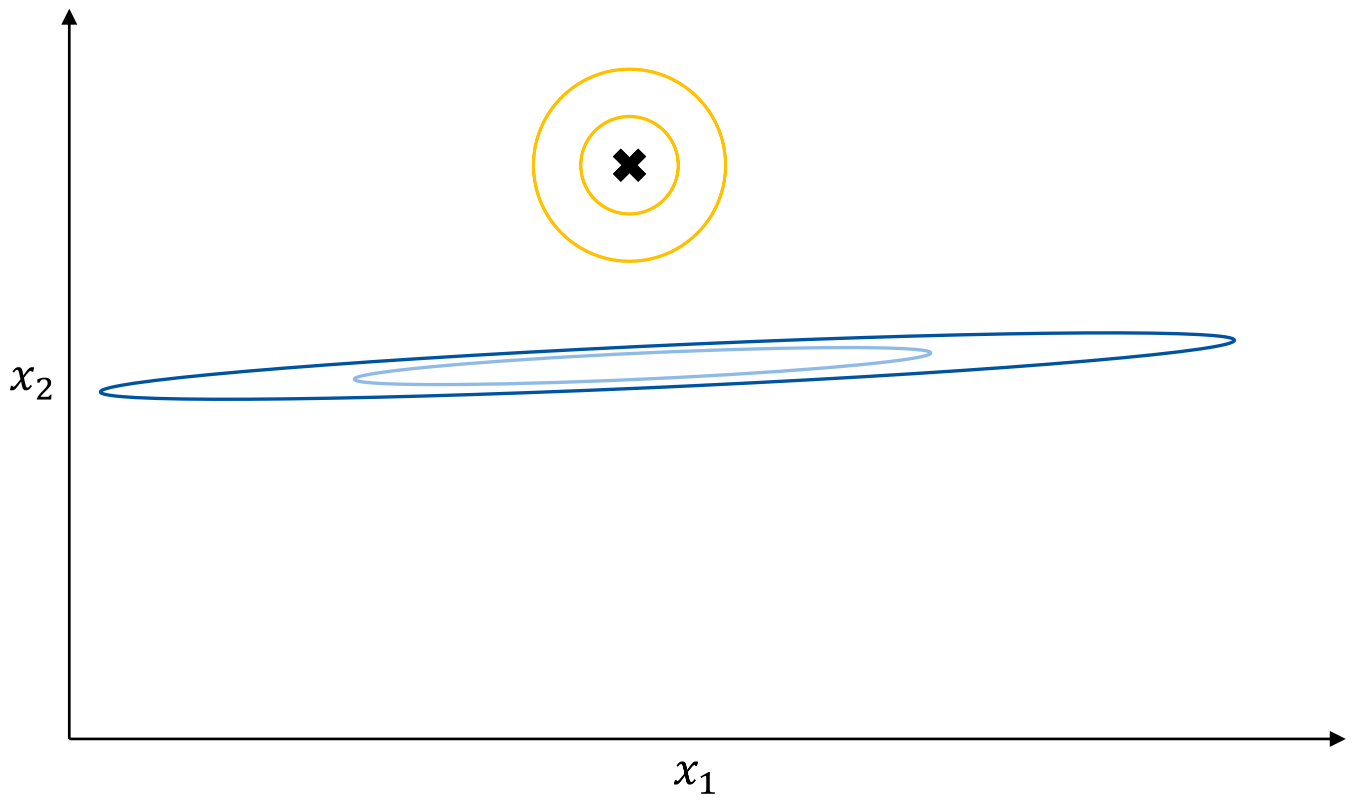
The acquisition function



Flat over large areas ⇒ We take the $\log$ of the acquisition function when actually optimizing it

Preprocessing

Problem 1: Different scales



Do two things:

1. Scale the priors such that they occupy the unit hypercube (every parameter is in $[0,1]$)

2. Make kernel asymmetric

$$k(x, x') = \sigma^2 \cdot \prod_{i=1}^{d} \exp\left(-\frac{(x_i - x_i')^2}{2l^2}\right)$$

More hyperparameters to fit ($d + 1$) but robust!

Preprocessing

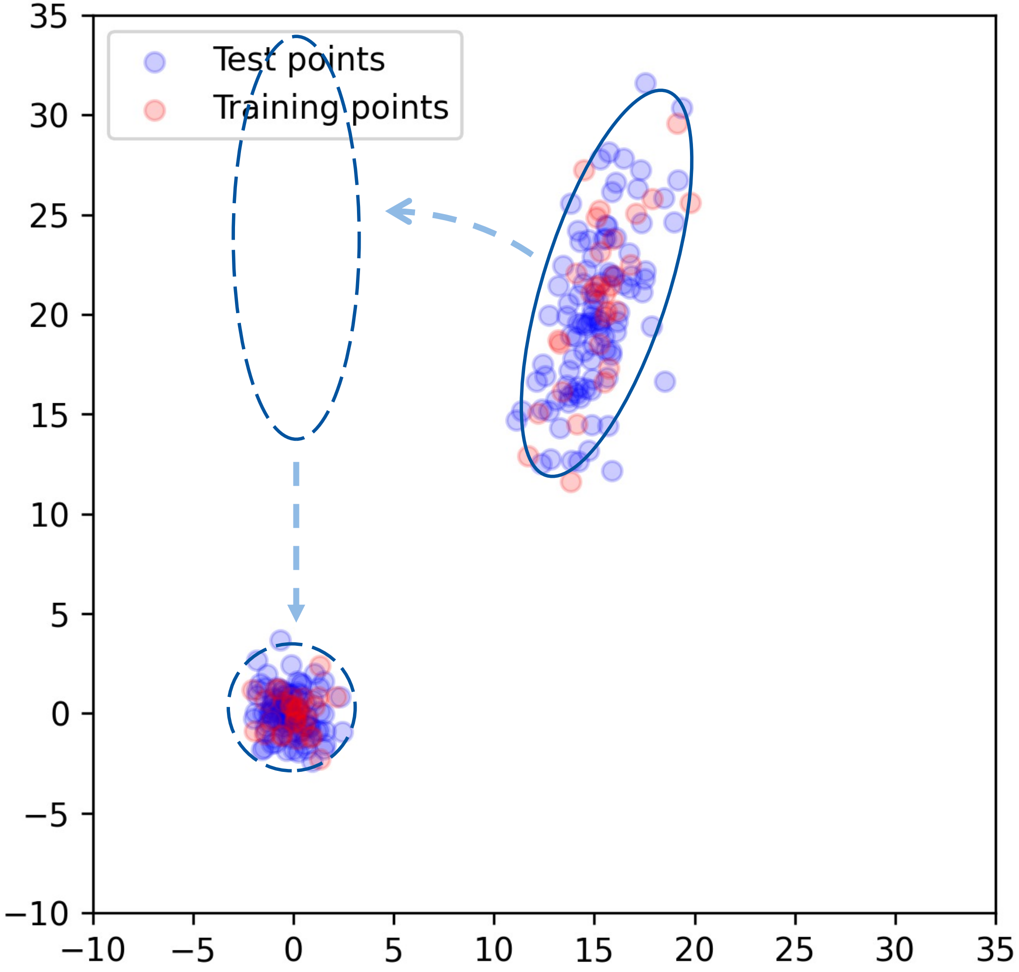
Alternative: Whitening

$$x_i \to x_i' = \frac{R_{ij}(x-\hat{\mu})_j}{\hat{\Sigma}_{ii}}$$

with

- $\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$
- $\hat{\Sigma}_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}-\hat{\mu}_j)\,(x_{ik}-\hat{\mu}_k)$
  (empirical mean and covariance along each dimension)
- $\hat{\Sigma} = R\Lambda R^{\mathrm{T}}$ with Λ diagonal

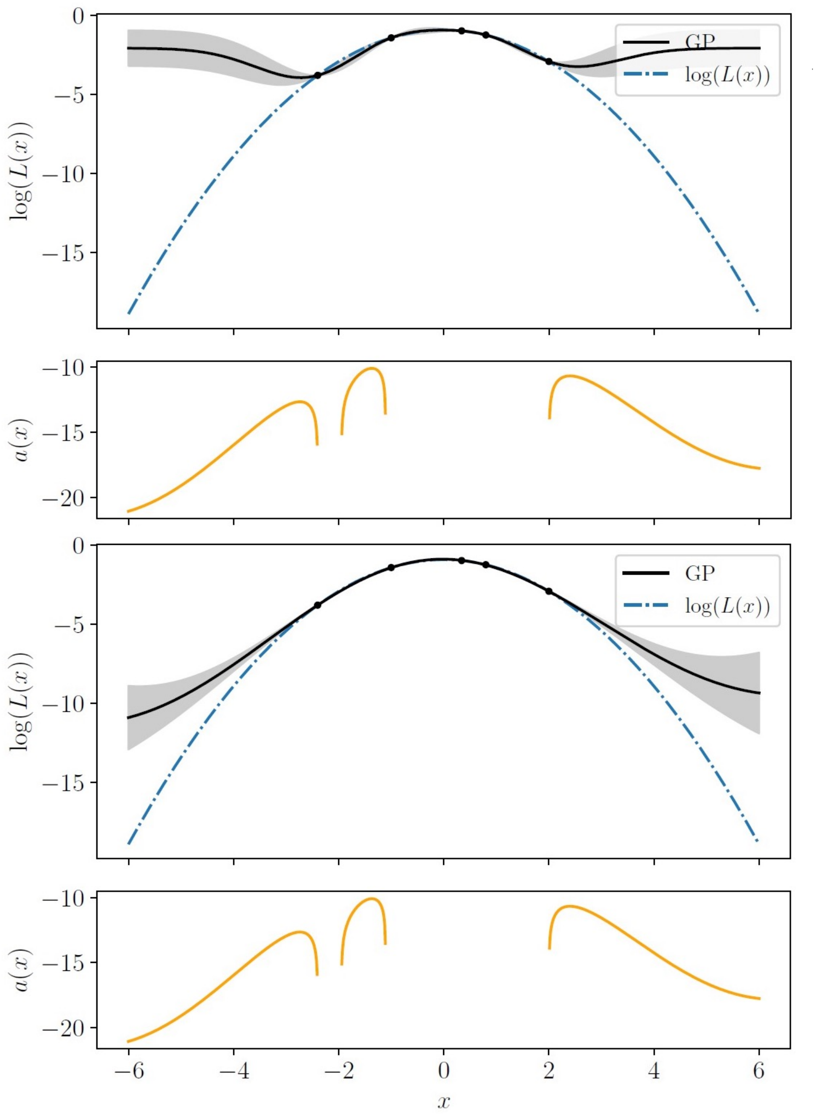
⇒ No need to make kernel asymmetric but less robust



Preprocessing

What about log-posterior values?

- Transform such that they have zero mean and unit variance
- Encourages exploration when lots of high values of the log-posterior
- Encourages exploitation when lots of low values of the log-posterior



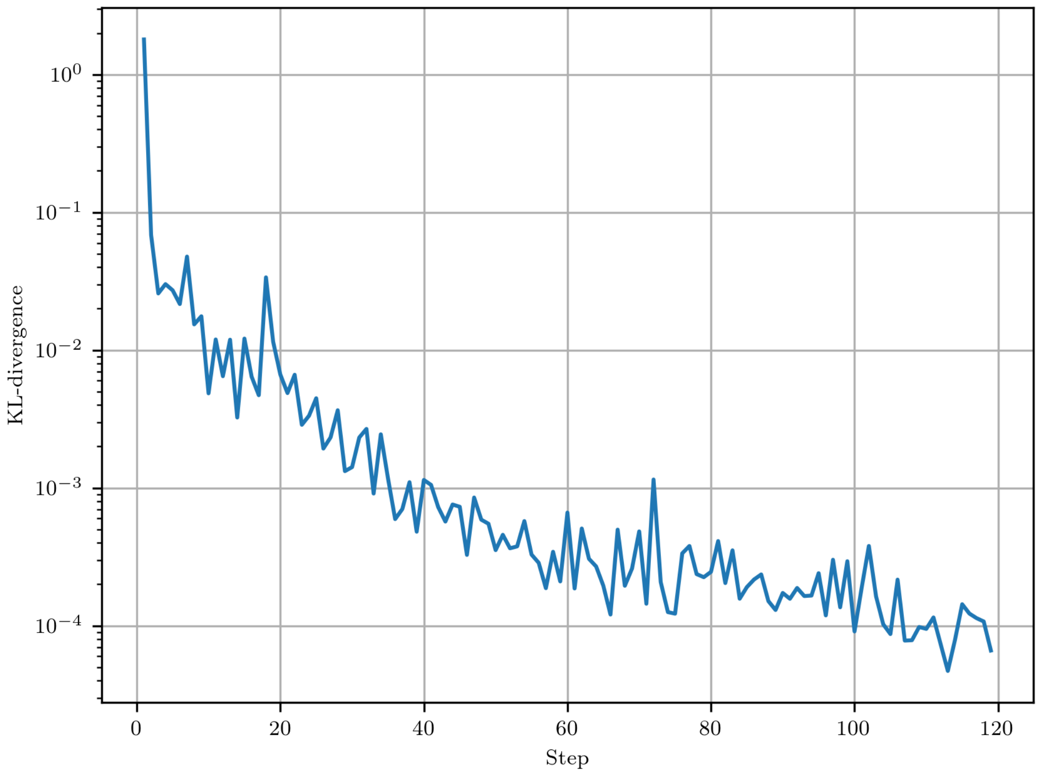
Kullback-Leibler divergence

Kullback-Leibler (KL) divergence:

$$D_{\mathrm{KL}}(P_{n+1}||P_n) = \sum_{x\in\chi} P_{n+1}(x)\log\left(\frac{P_{n+1}(x)}{P_n(x)}\right)$$

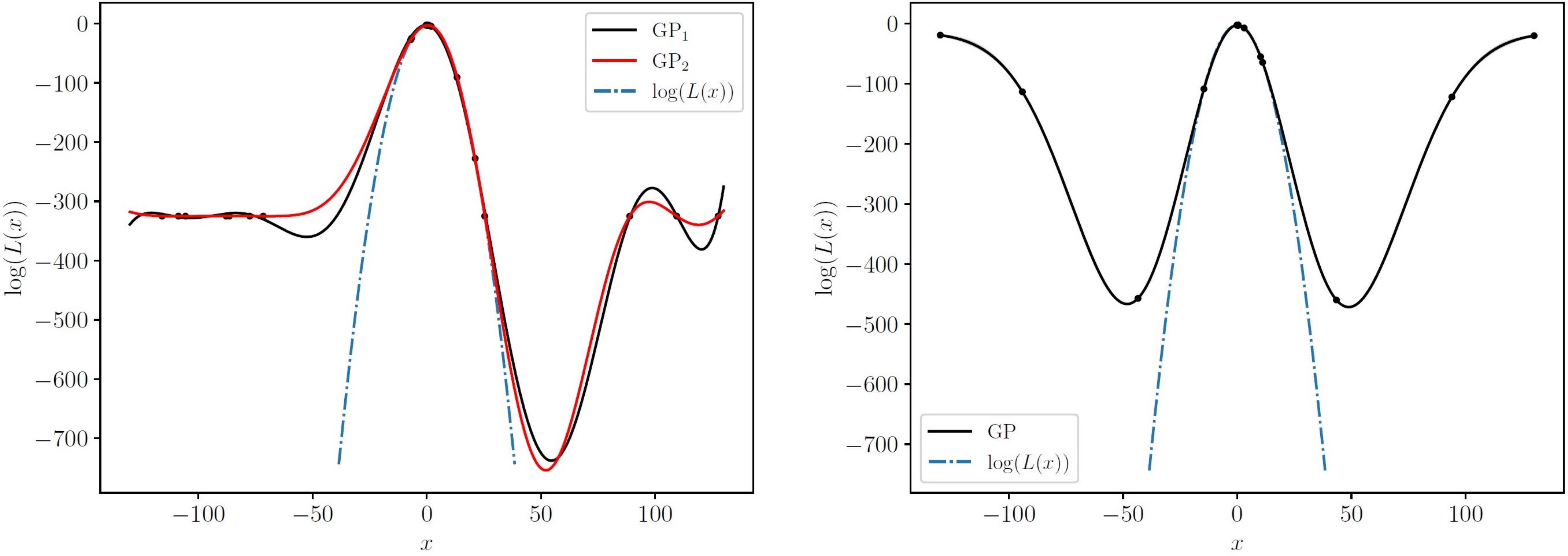
In case of a multivariate Gaussian this is just

$$D_{\mathrm{KL}}(P||Q) = \frac{1}{2}\left[\log\frac{|\Sigma_q|}{|\Sigma_p|} - d + \mathrm{tr}\left(\Sigma_q^{-1}\Sigma_p\right) + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^{\mathrm{T}}\Sigma_q^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)\right]$$



For now: Take empirical mean and covariance of the **training points**

The problem with infinity

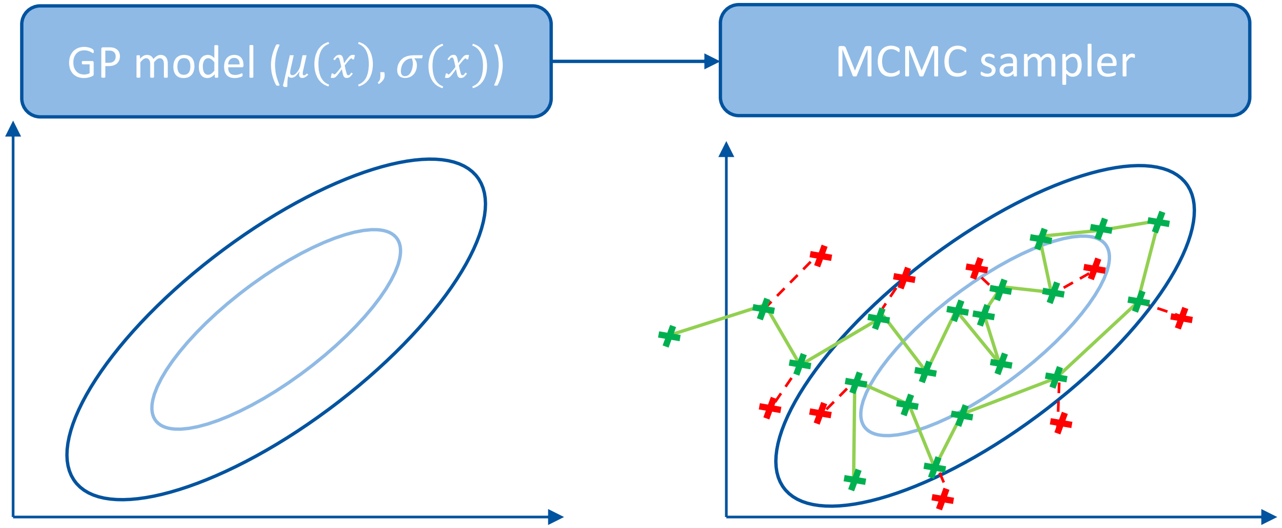


Are we preserving Bayesianity?

We are violating Bayesianity at two points:

$$\Rightarrow \log\big(p(y|X)\big) = -\frac{1}{2}y^T K^{-1} y - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi)$$

We are maximizing this with MLII. Correct Bayesian way:
Sampling the posterior distribution but very expensive!



→ Ignoring $\sigma_{\mathrm{GP}}(x)$

Correct:
Get var(posterior) by solving

$$\iint k_{f,\mathrm{D}}(x, x')\,\mathrm{d}x\,\mathrm{d}x'$$

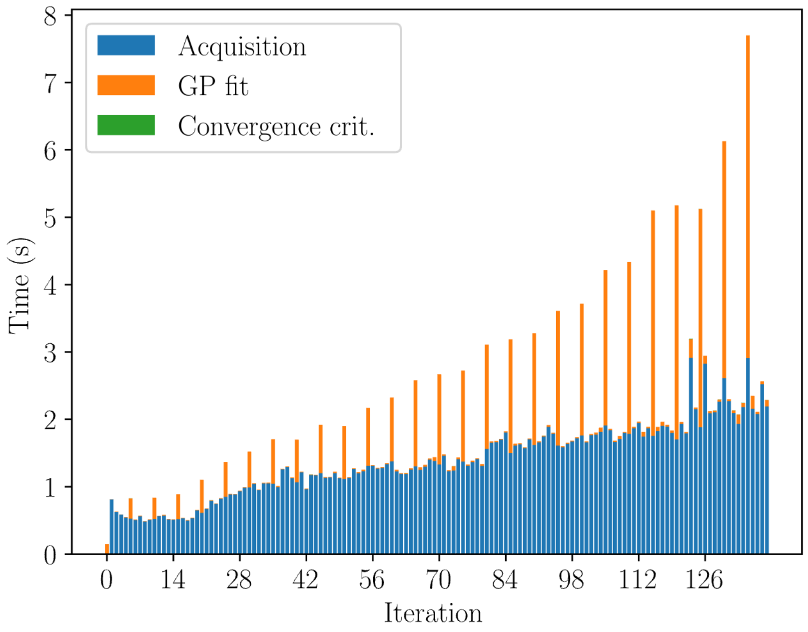
But super expensive!

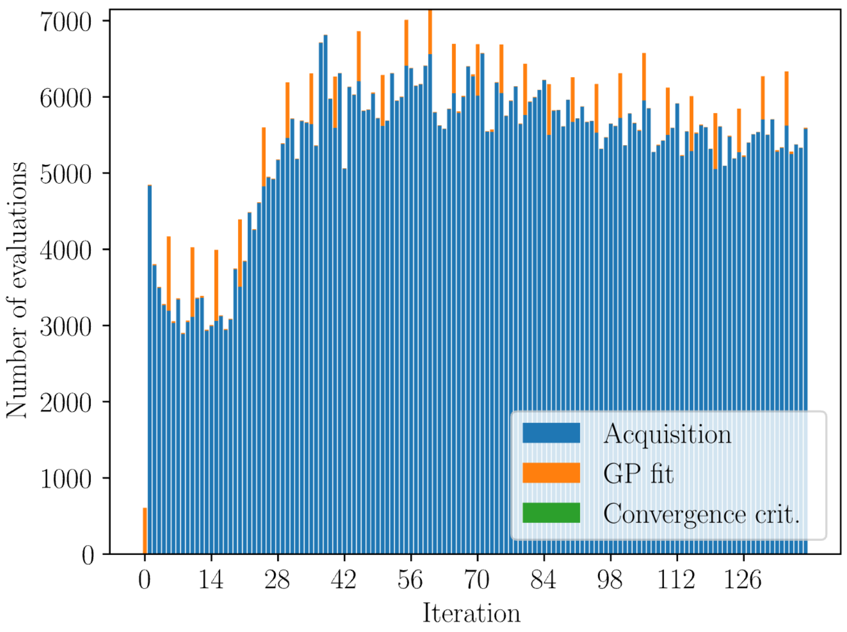
Overhead

8 dimensions

2 Kriging believer steps/iteration

In total 300 accepted samples





Refitting GP hyperparameters requires many inversions of the kernel matrix, scales $\mathcal{O}(N^3_{\text{samples}})$