Gaussian Processes for HEP

Adam Bozson

IoP APP/HEPP Conference 2018

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Definition

A collection of random variables, any finite number of which have a joint Gaussian distribution

Gaussian Processes for Machine Learning, Rasmussen & Williams

A GP is completely defined by mean and covariance functions

$$m(\boldsymbol{x}) = E[f(\boldsymbol{x})]$$
$$k(\boldsymbol{x}, \boldsymbol{x}') = E[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))]$$

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

Sampling from the GP

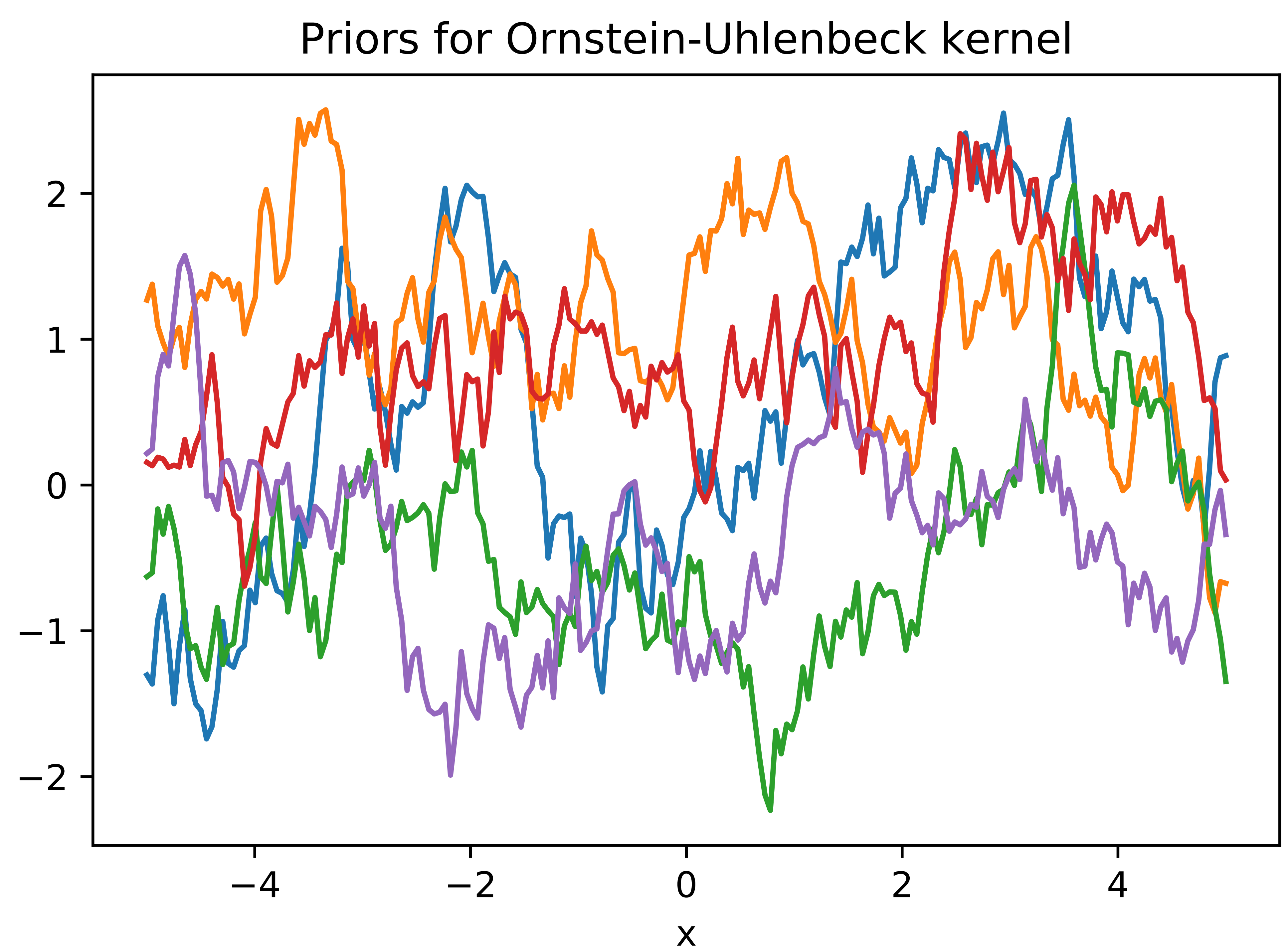
$$\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, I)$$
$$\boldsymbol{f} = \boldsymbol{m} + L\,\boldsymbol{u}$$

where $L\,L^{\mathsf{T}} = K$ (Cholesky)

Examples of Kernels

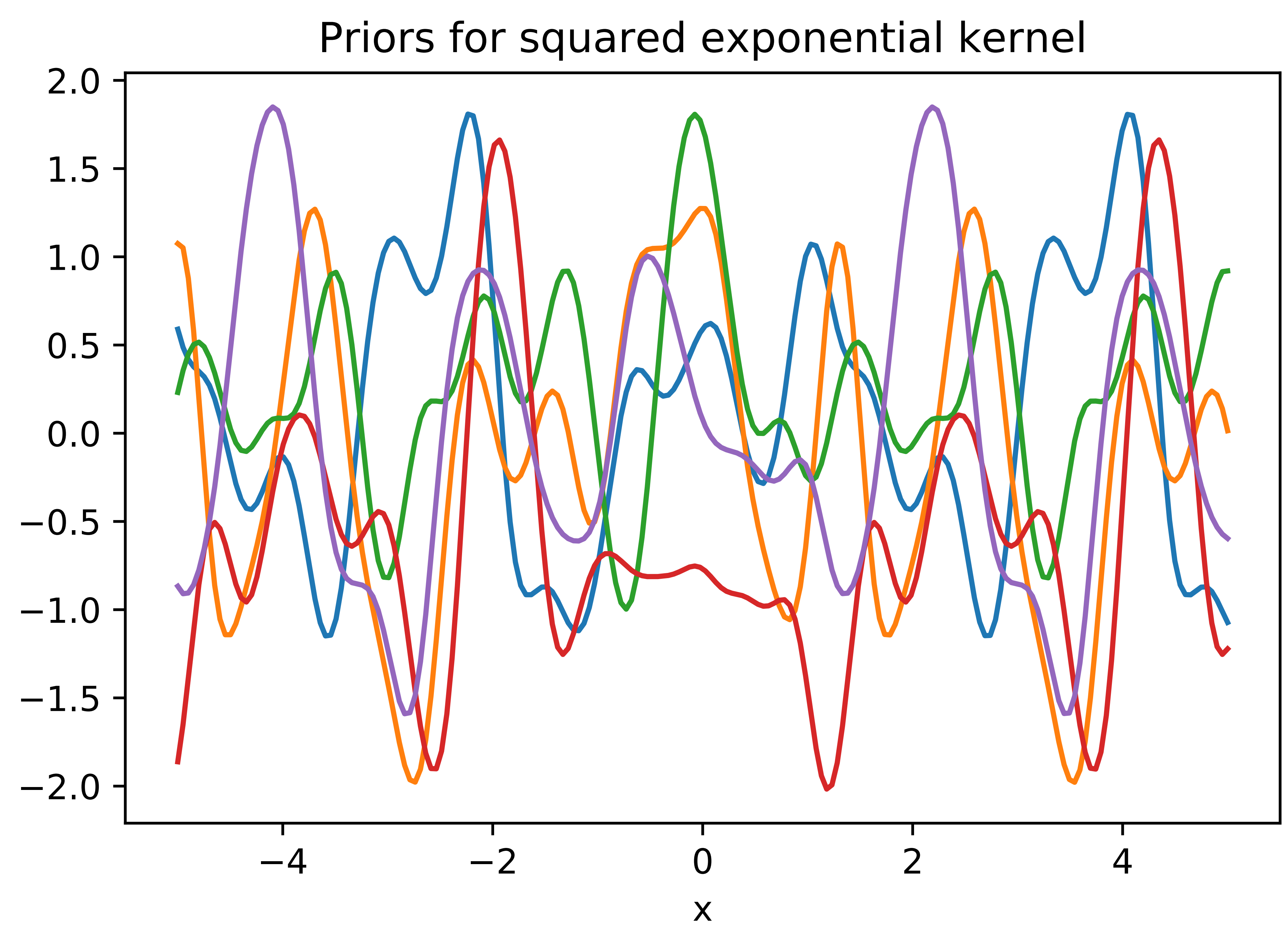
Ornstein-Uhlenbeck kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||}{l}\right)$$



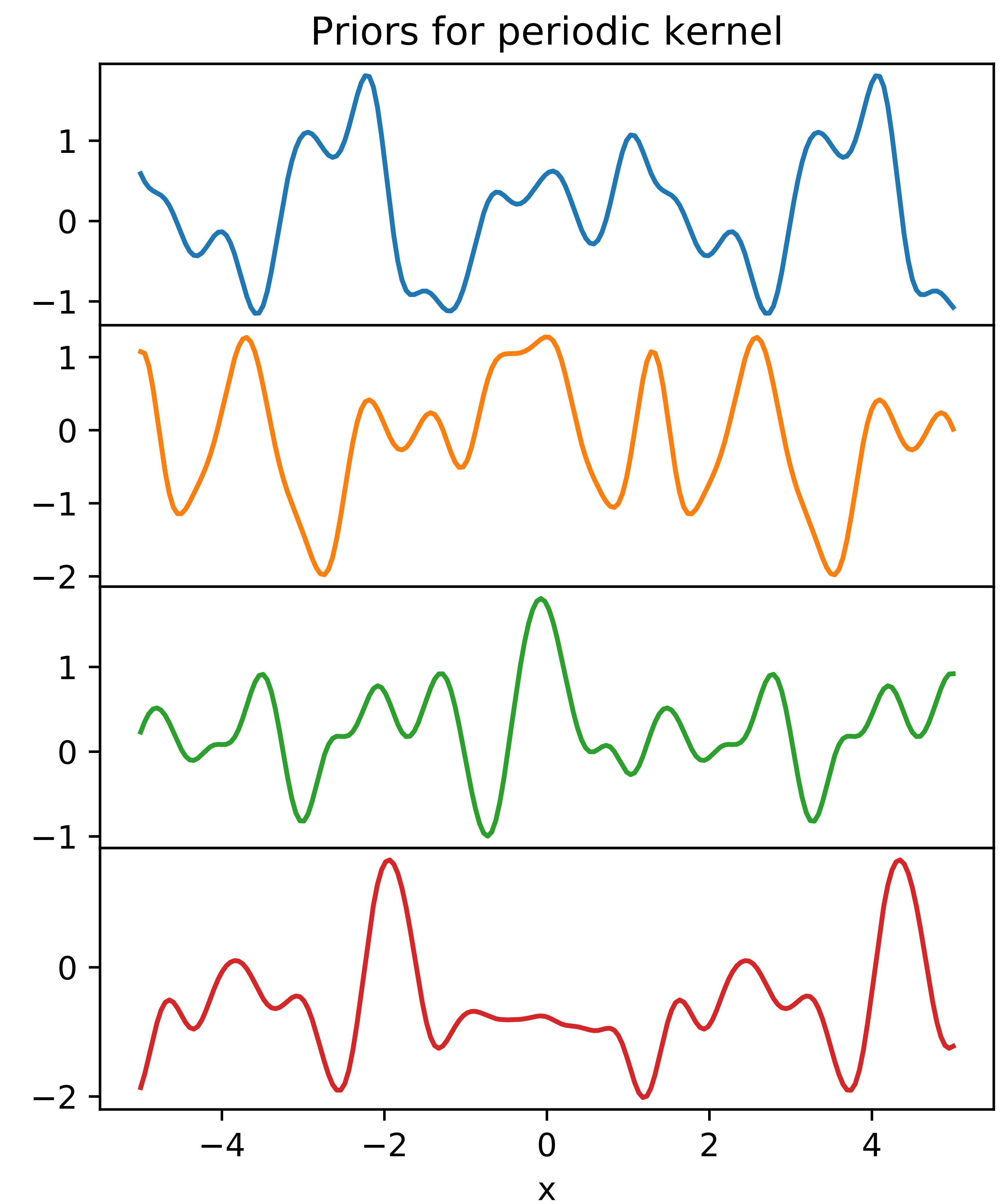
Squared exponential kernel

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



Periodic kernel

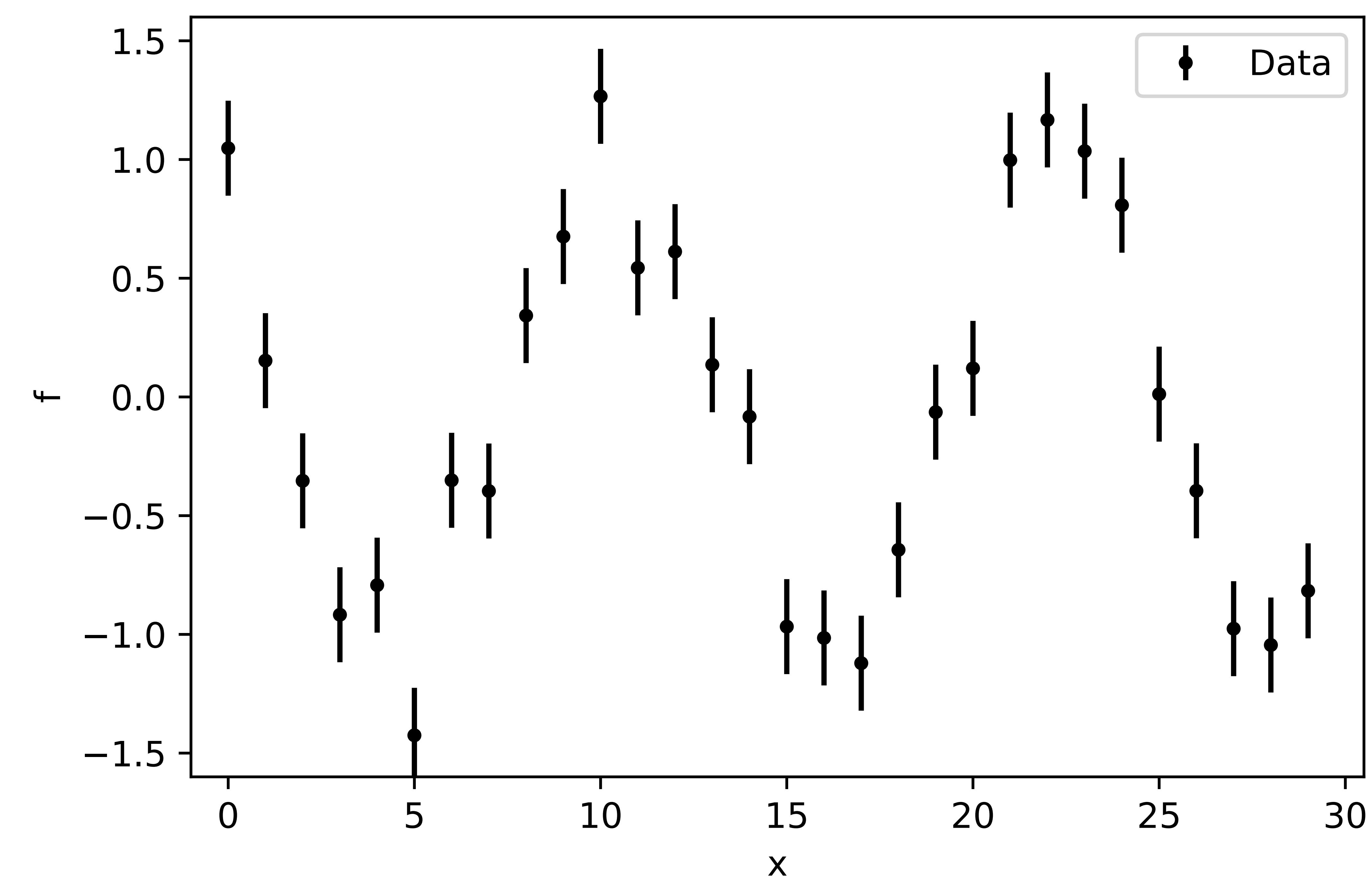
$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{2\sin^2\left(\frac{|\boldsymbol{x}-\boldsymbol{x}'|}{2}\right)}{l^2}\right)$$



Application

Regression

Aim: Given data f at X, predict f_* at X_*



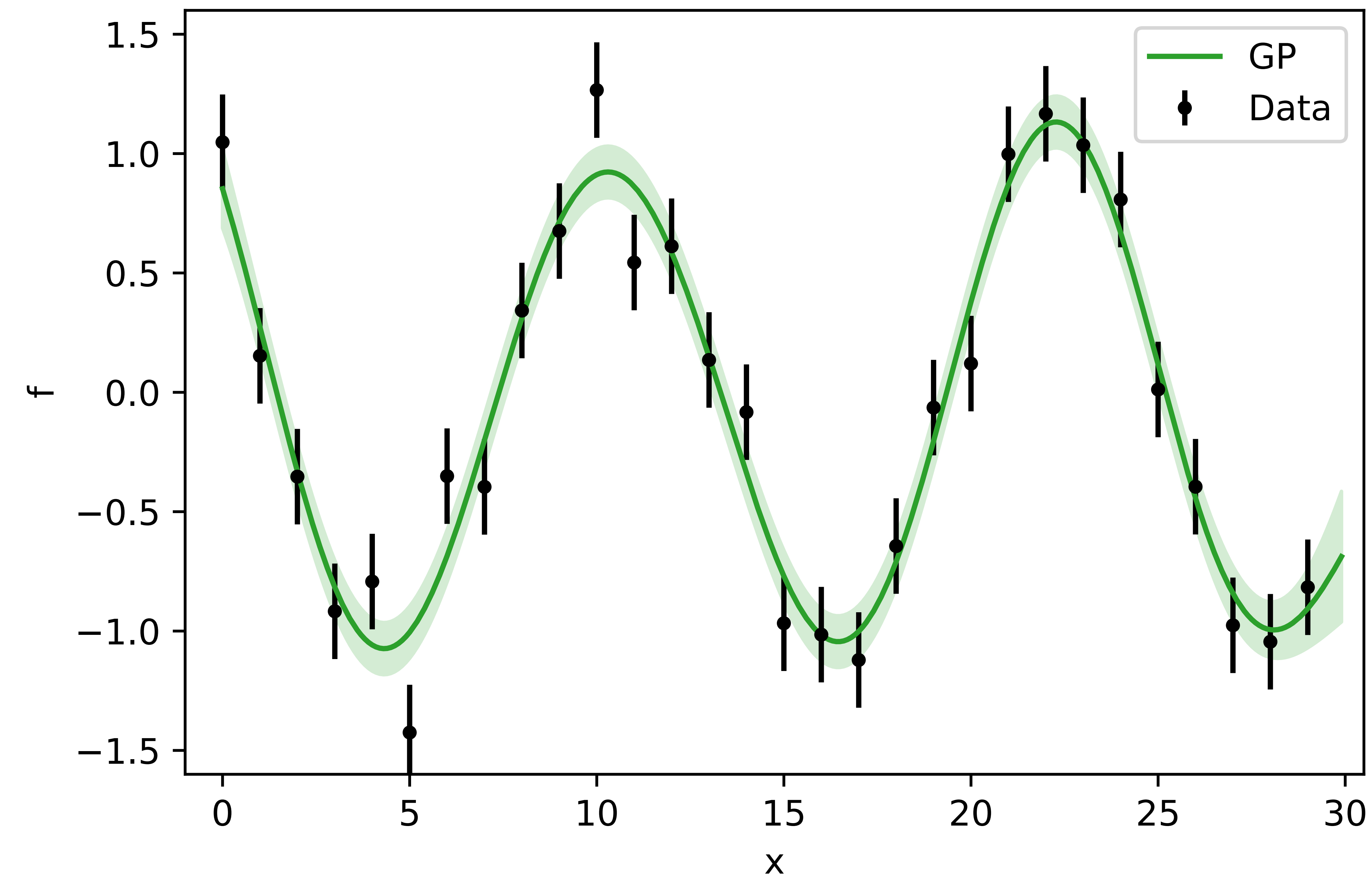
Aim: Given data $\boldsymbol{f}$ at X, predict $\boldsymbol{f}_*$ at X_*

Condition the Gaussian prior on the data to give posterior distribution

$$\boldsymbol{f}_*|X_*, X, \boldsymbol{f} \sim \mathcal{N}\big(K(X_*, X)K(X, X)^{-1}\boldsymbol{f}, \\ K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)\big)$$

Include data uncertainties with

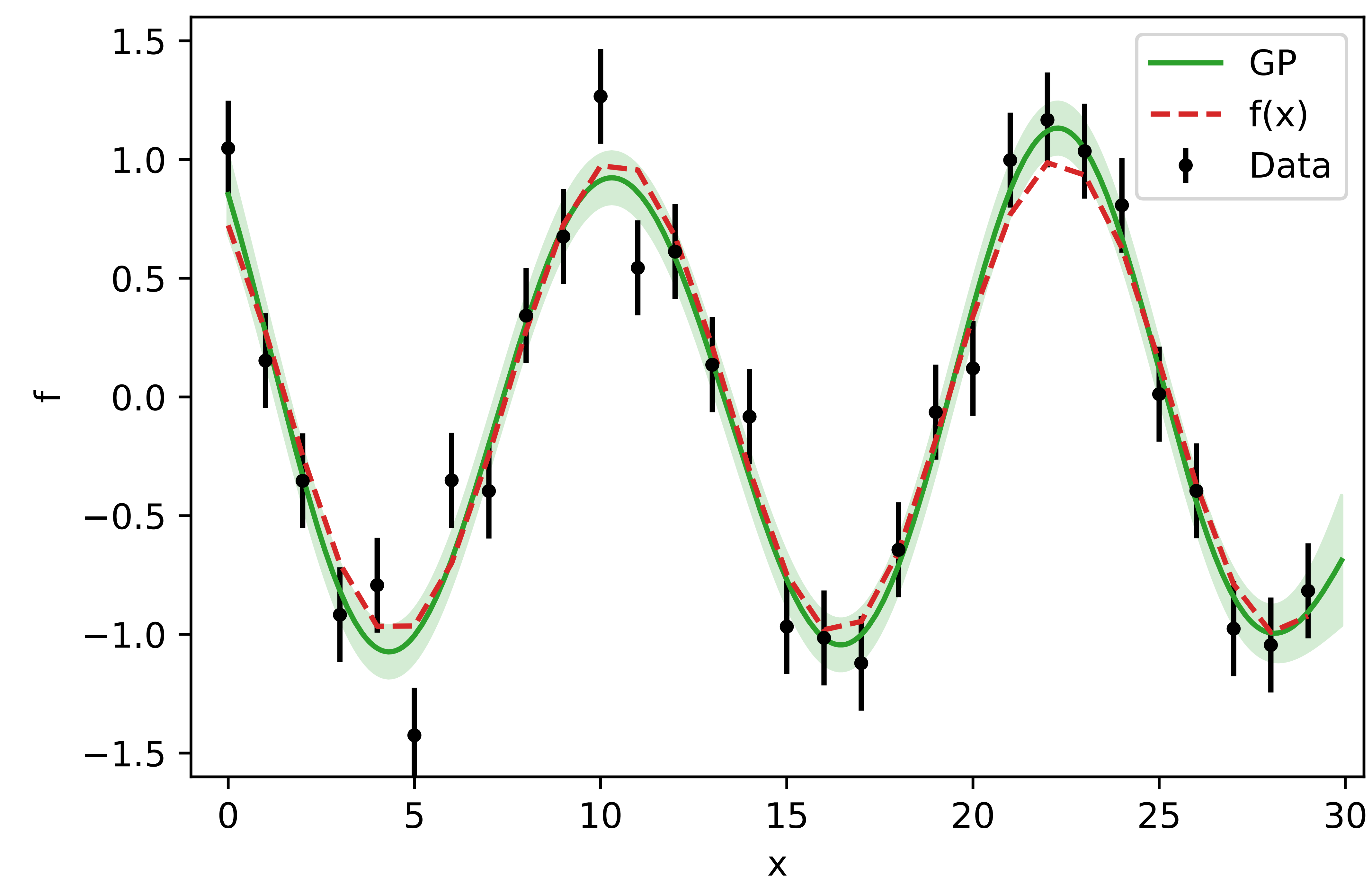
$$K(X, X)_{ij} \to K(X, X)_{ij} + \delta_{ij}\sigma_i^2$$



Aim: Given data $\boldsymbol{f}$ at X, predict $\boldsymbol{f}_*$ at X_*

Condition the Gaussian prior on the data to give posterior distribution

$$\boldsymbol{f}_*|X_*, X, \boldsymbol{f} \sim \mathcal{N}\big(K(X_*, X)K(X, X)^{-1}\boldsymbol{f}, \\ K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)\big)$$



Include data uncertainties with

$$K(X, X)_{ij} \to K(X, X)_{ij} + \delta_{ij}\sigma_i^2$$

Maximise likelihood to find optimal kernel parameters

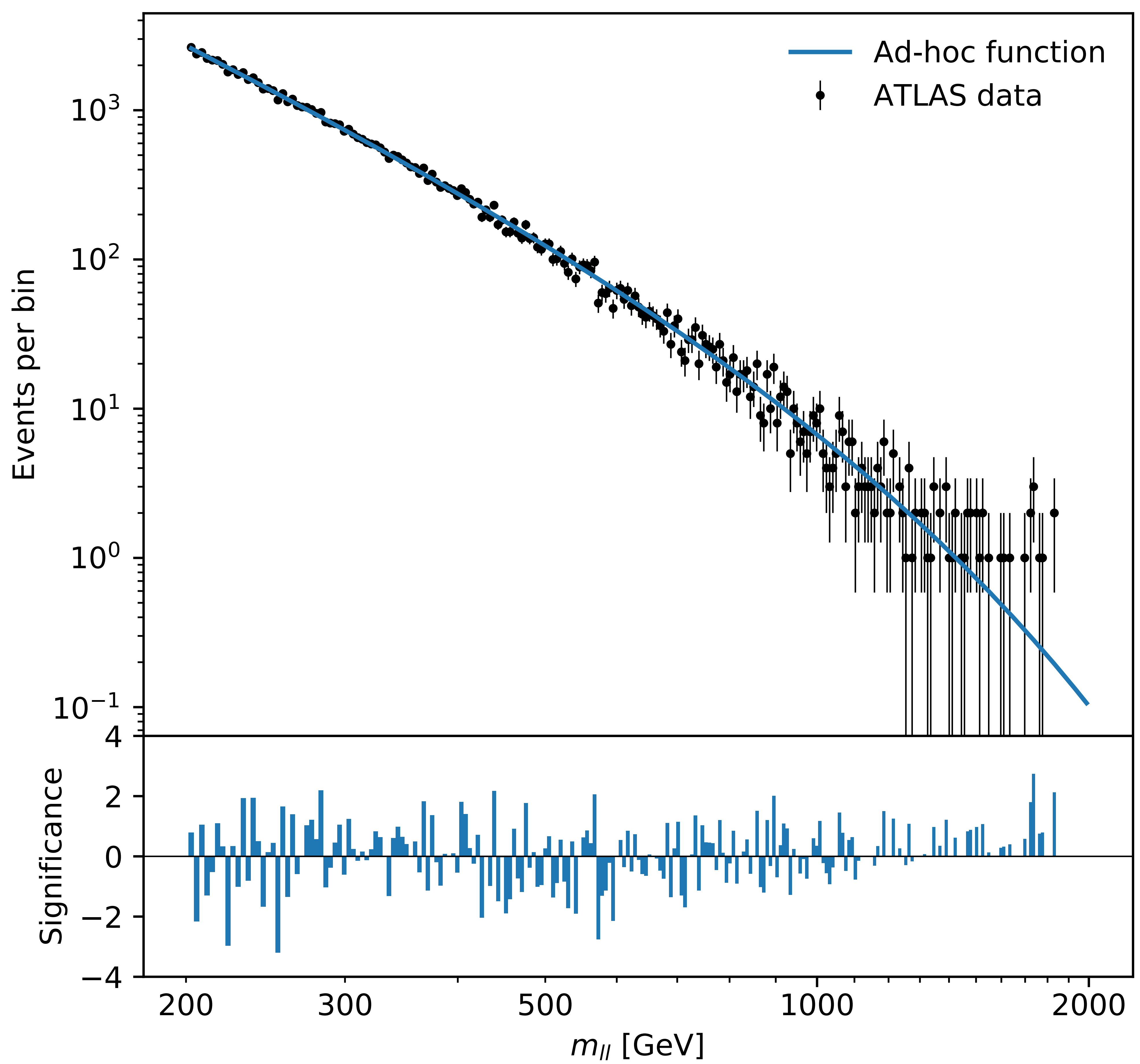
$$\log L(\boldsymbol{f}|X, \theta) = -\frac{1}{2}\boldsymbol{f}^\mathsf{T} K_\theta(X, X)^{-1}\boldsymbol{f} - \frac{1}{2}\log|K_\theta(X, X)| - \frac{n}{2}\log 2\pi$$

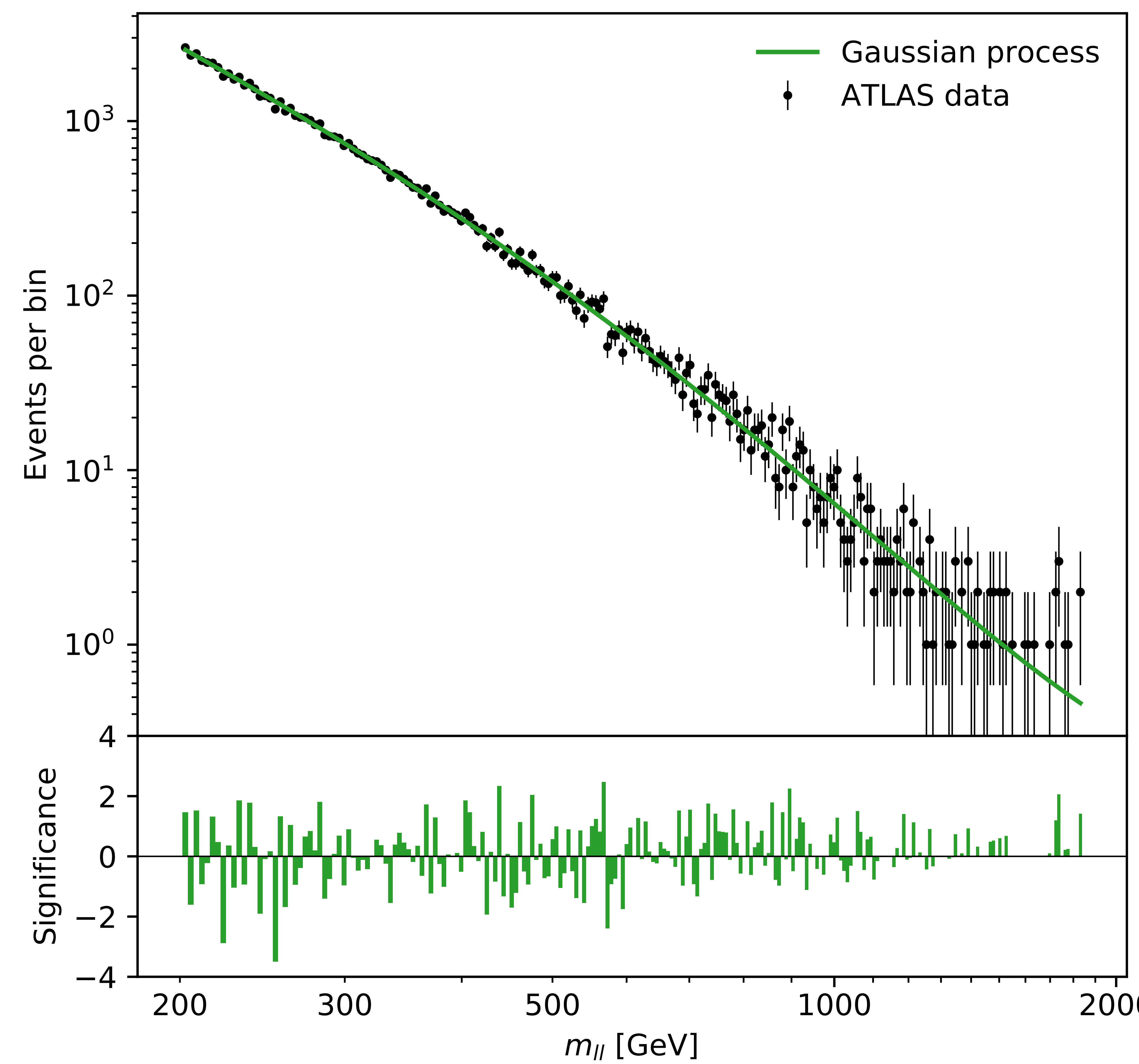
Application

Background Modelling

Dijet: arXiv:1709.05681 *Modeling Smooth Backgrounds and Generic Localized Signals with Gaussian Processes*, K Cranmer et al.

ATLAS **ee** dataset: HEPData (Table 2)



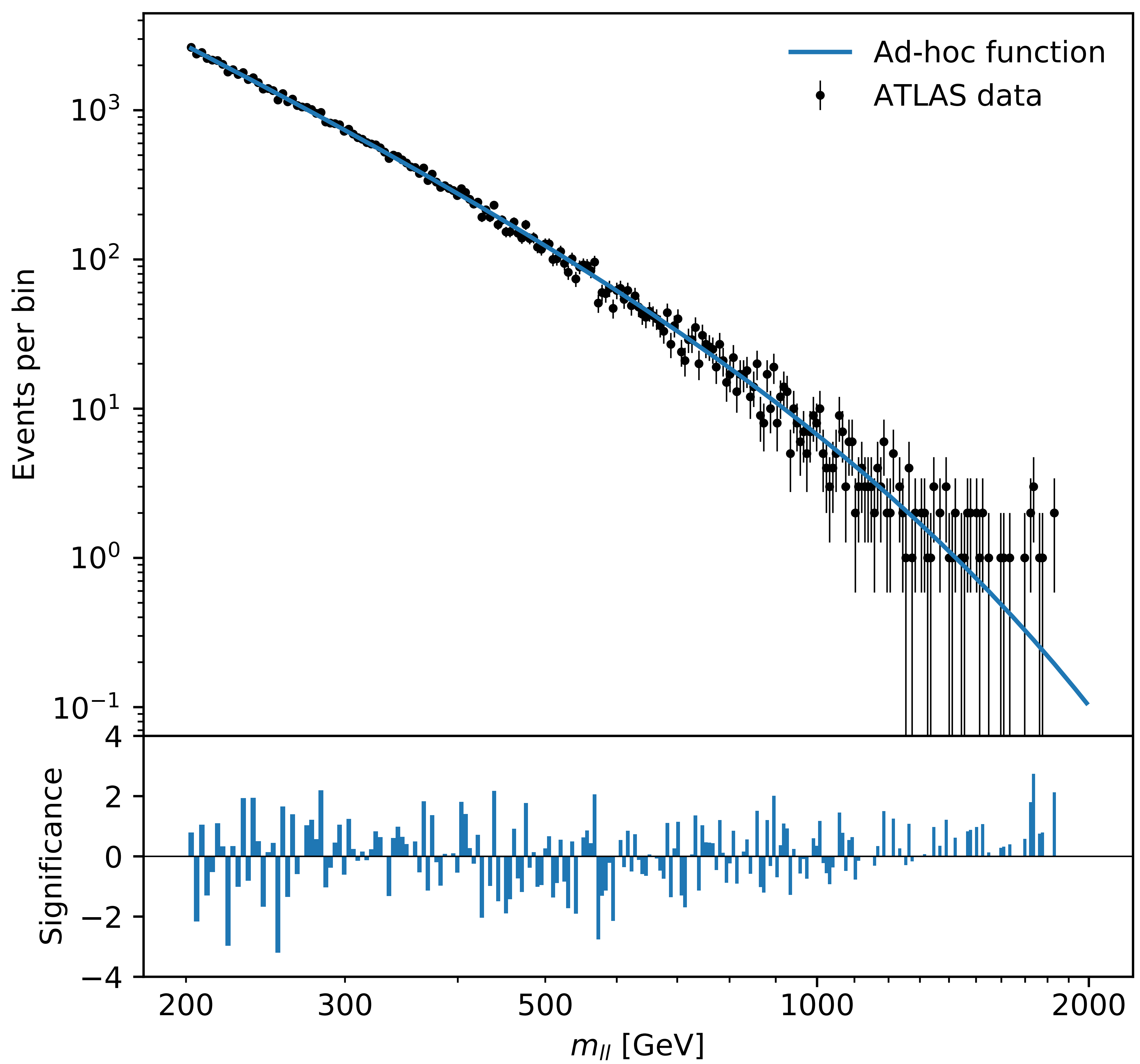


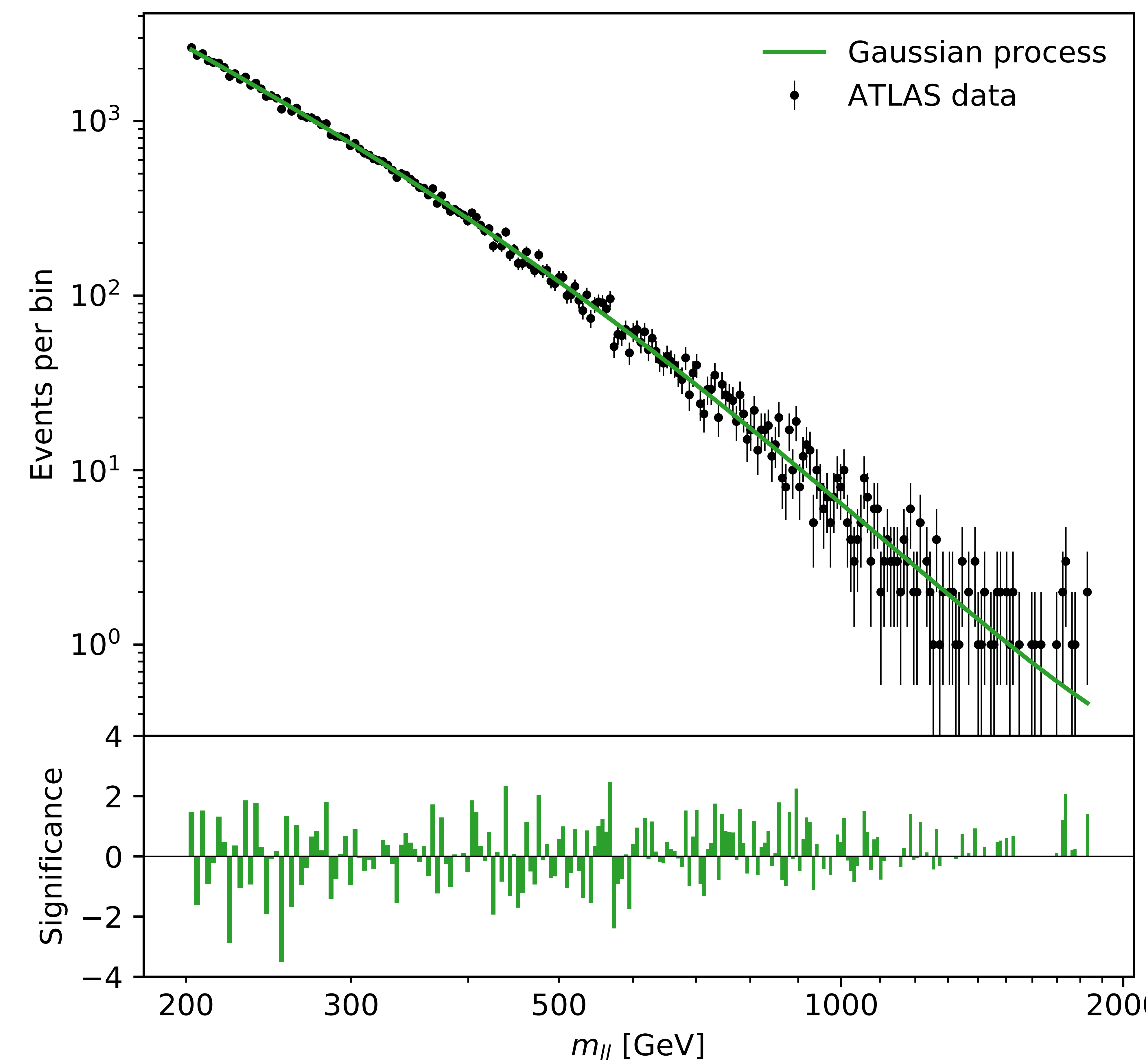
$$y = \theta_0 \, (1 - x)^{\theta_1} \, x^{\theta_2}$$

$$x = m_{ll}/\sqrt{s}$$

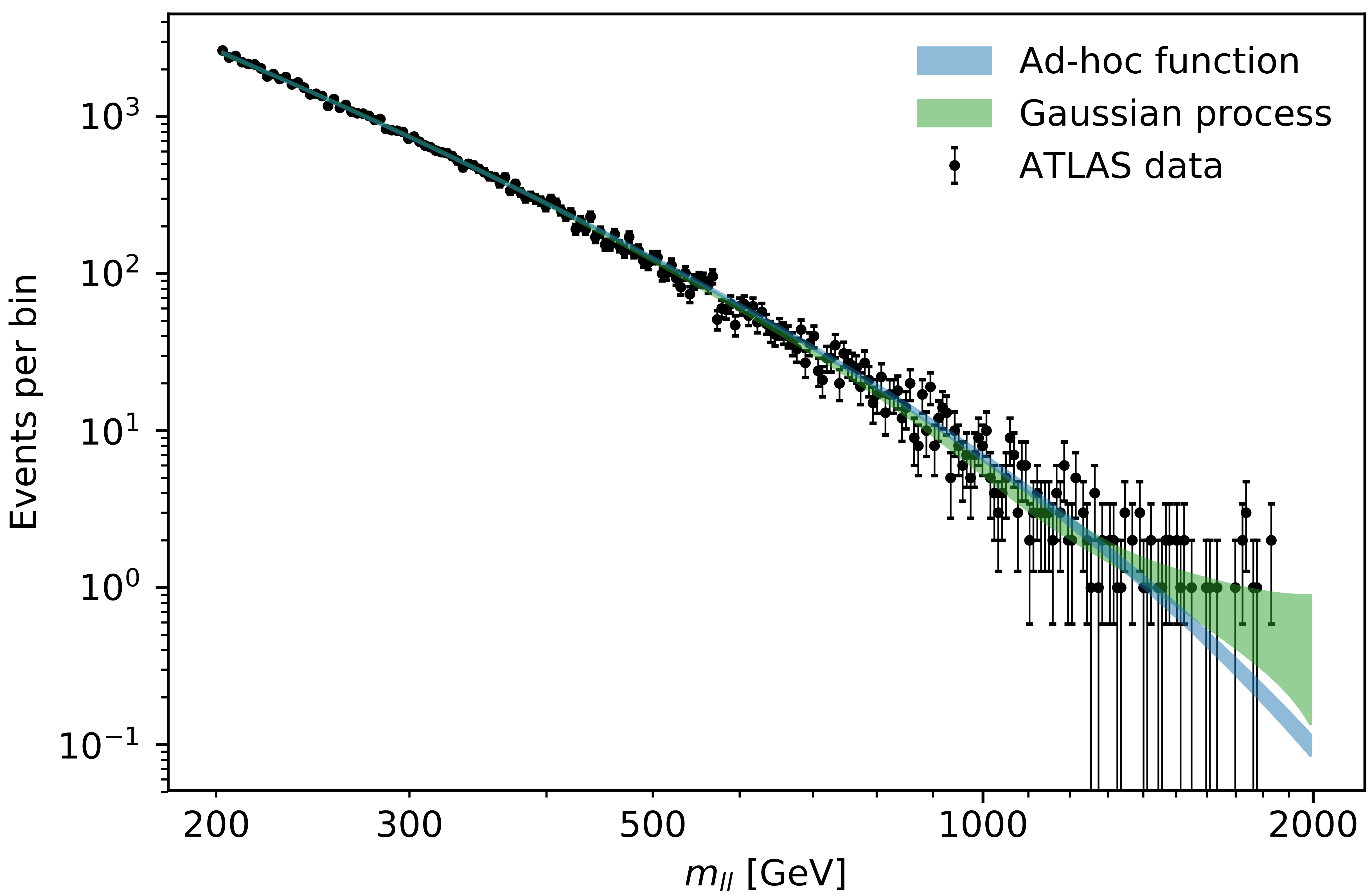
$$k(x, x') = A e^{\frac{d-(x+x')}{2a}} \sqrt{\frac{2l(x)l(x')}{l(x)^2 + l(x')^2}} e^{\frac{-(x-x')^2}{l(x)^2+l(x')^2}}$$

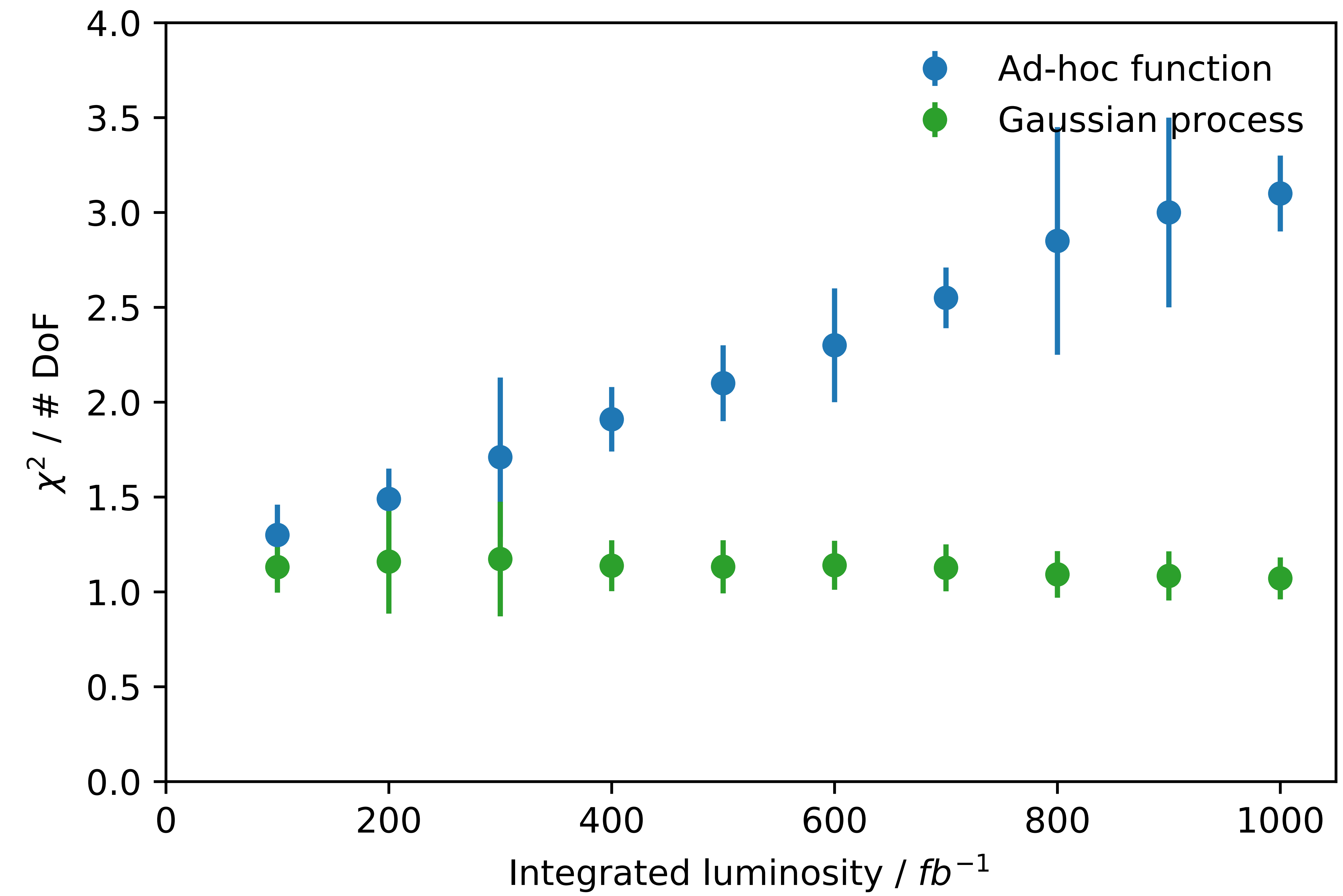
$$l(x) = b\,x + c$$





Fitting to toys





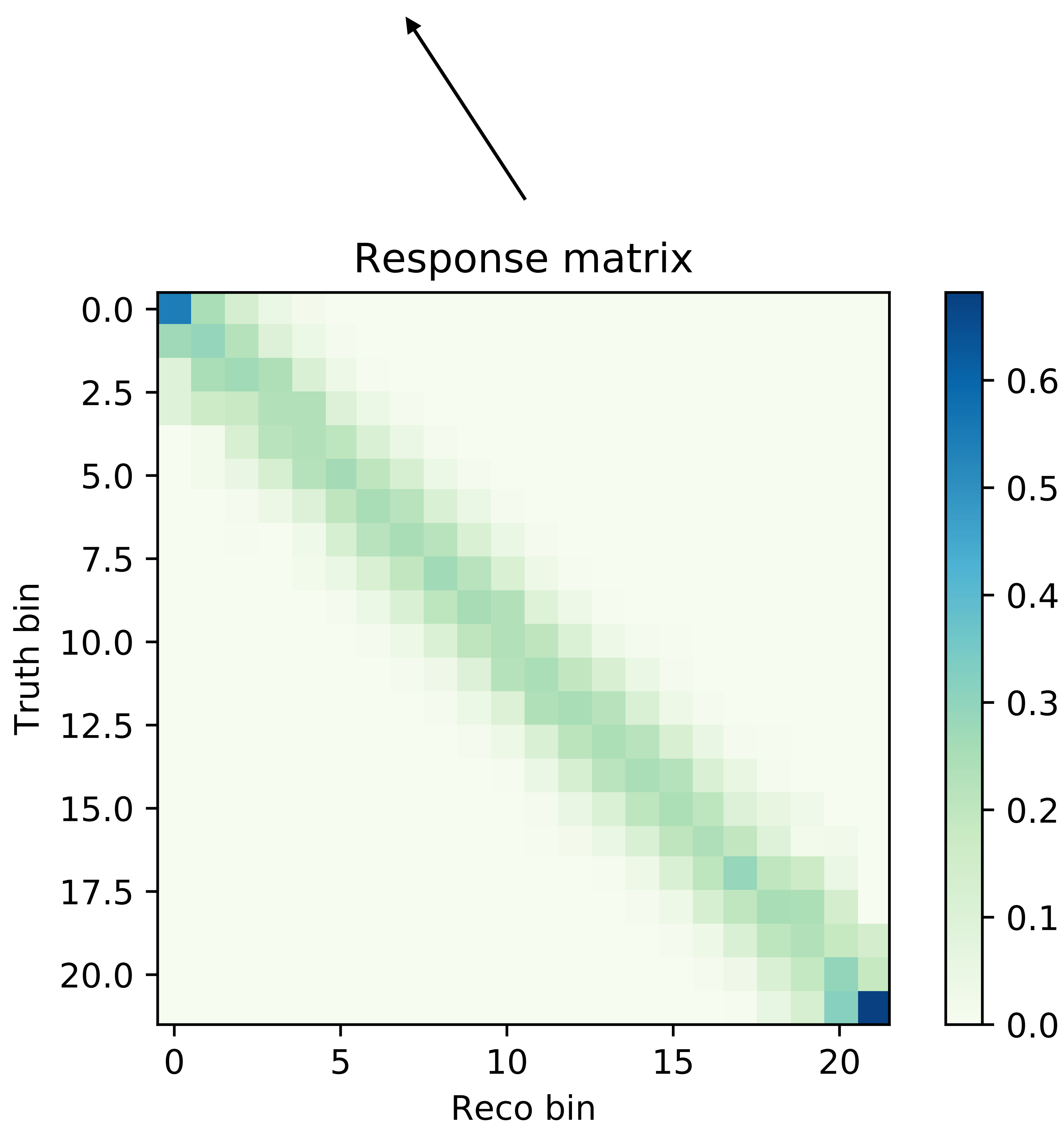
Application

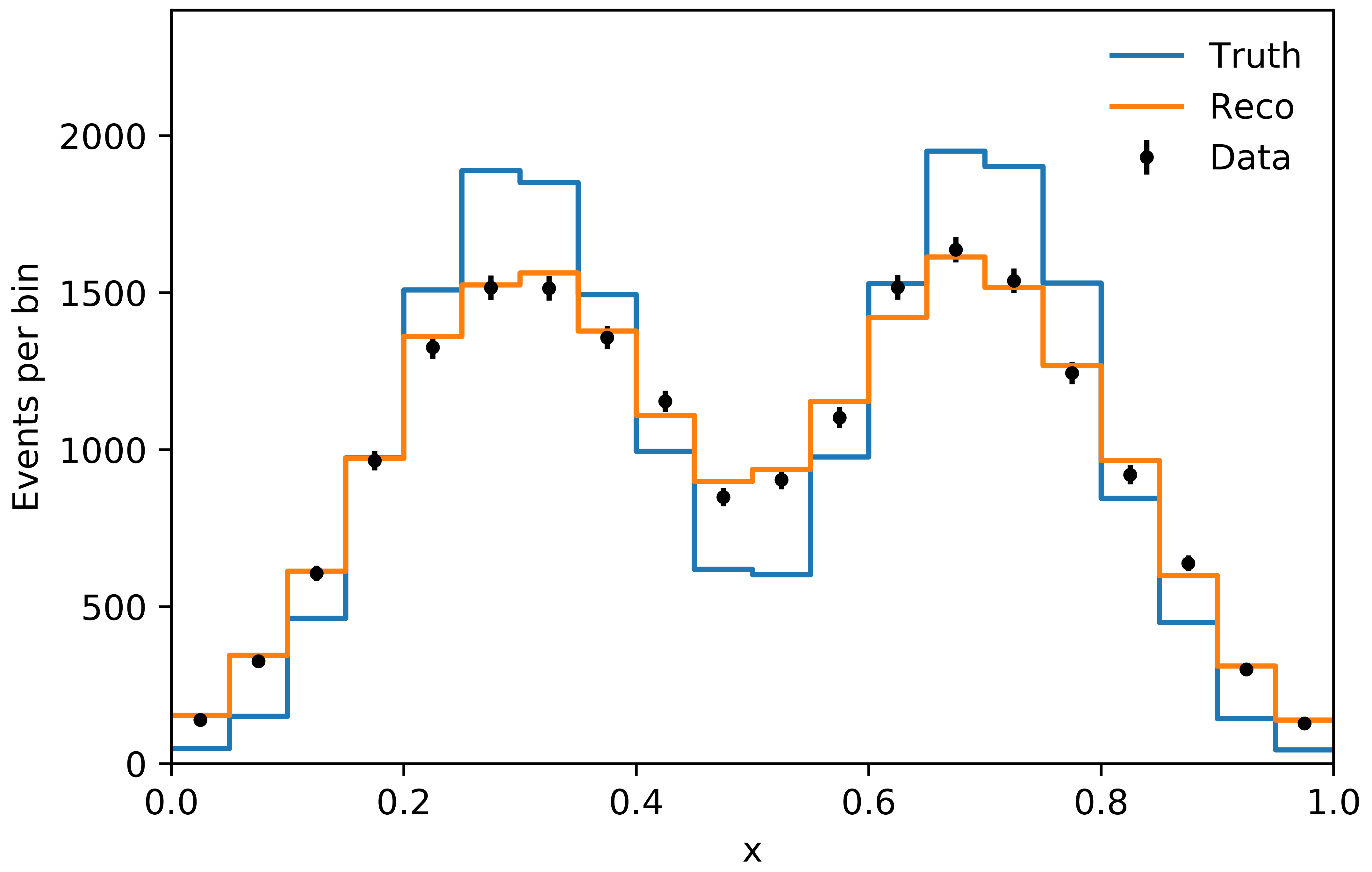
Unfolding

Aim: Solve $\boldsymbol{\nu} = R\boldsymbol{\mu} + \boldsymbol{\beta}$ for $\boldsymbol{\mu}$ given data

Regression task, but with posterior mean

$$\bar{\boldsymbol{f}}_* = K(X_*, X)\, R\, K(X,X)^{-1} \boldsymbol{f}$$



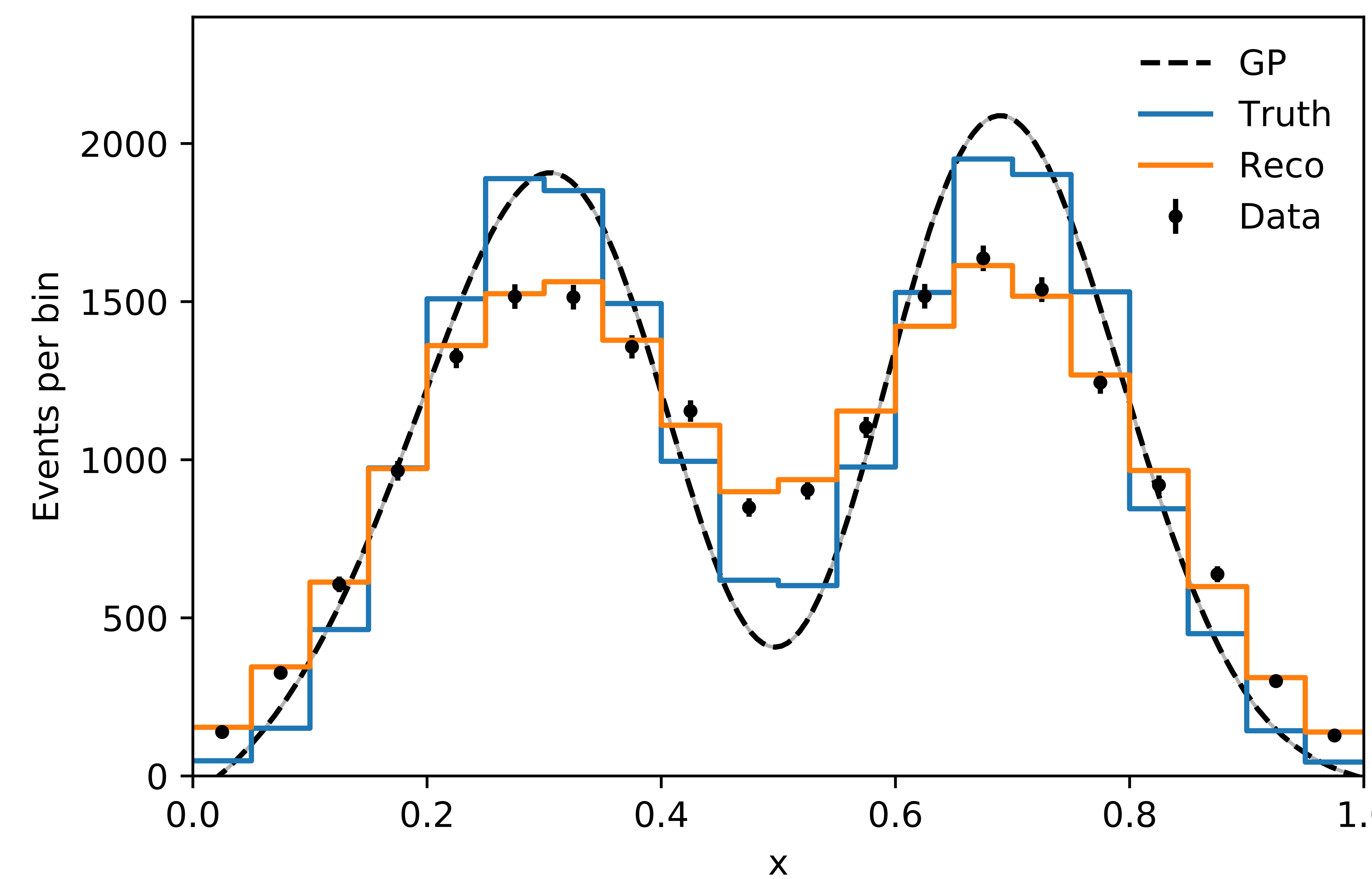


Aim: Solve $\boldsymbol{\nu} = R\boldsymbol{\mu} + \boldsymbol{\beta}$ for $\boldsymbol{\mu}$ given data

Regression task, but with posterior mean

$$\bar{\boldsymbol{f}}_* = K(X_*, X)\, R\, K(X, X)^{-1} \boldsymbol{f}$$

The kernel imposes smoothness in the posterior — *regularisation* (See Kernel Reproducing Hilbert Space)



Aim: Solve $\boldsymbol{\nu} = R\boldsymbol{\mu} + \boldsymbol{\beta}$ for $\boldsymbol{\mu}$ given data

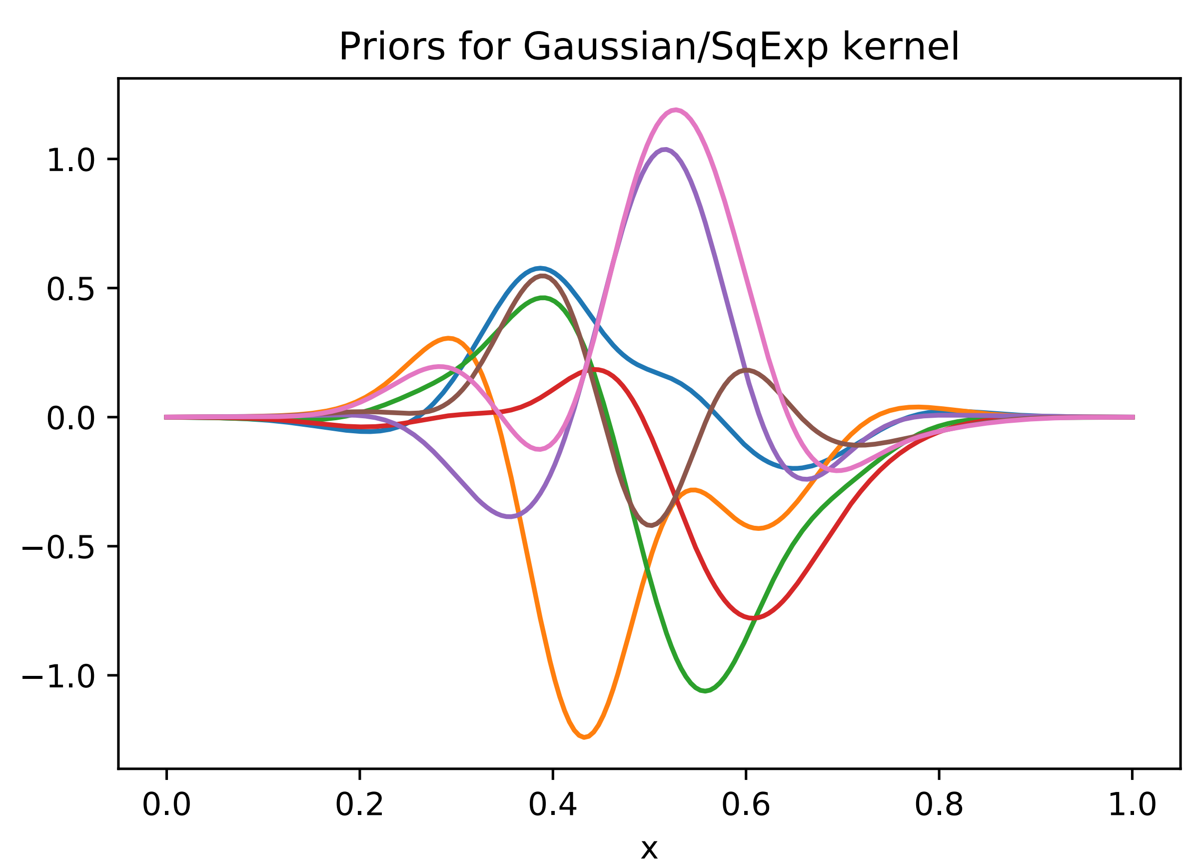
Regression task, but with posterior mean

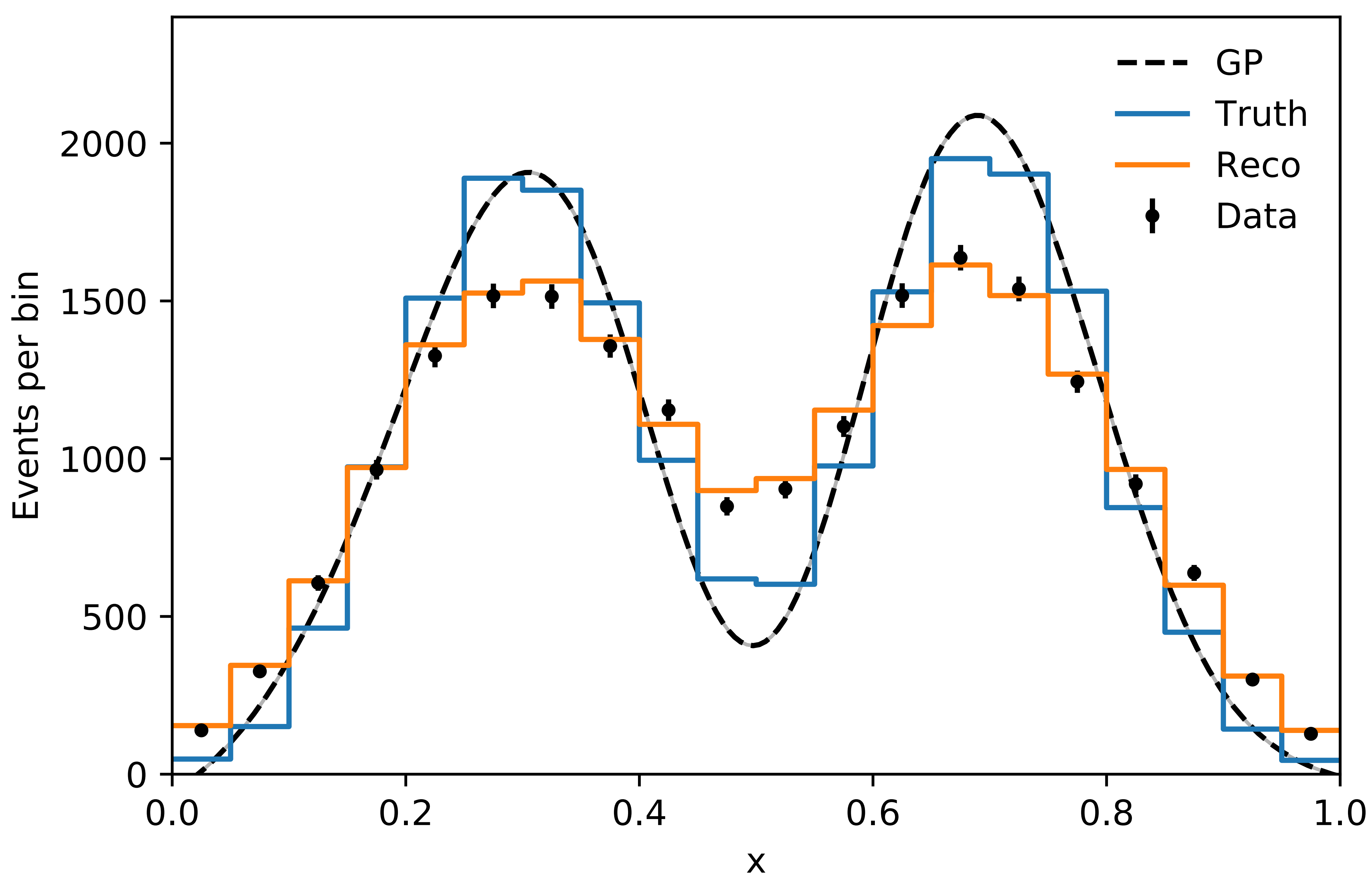
$$\bar{\boldsymbol{f}}_* = K(X_*, X)\, R\, K(X, X)^{-1} \boldsymbol{f}$$

The kernel imposes smoothness in the posterior — *regularisation* (See Kernel Reproducing Hilbert Space)

e.g. Gaussian noise ⇒ Squared exponential kernel

Can encode physics in the kernel





Conclusion

Conclusion

Gaussian Processes are really useful

github.com/adambozson/gp-talk

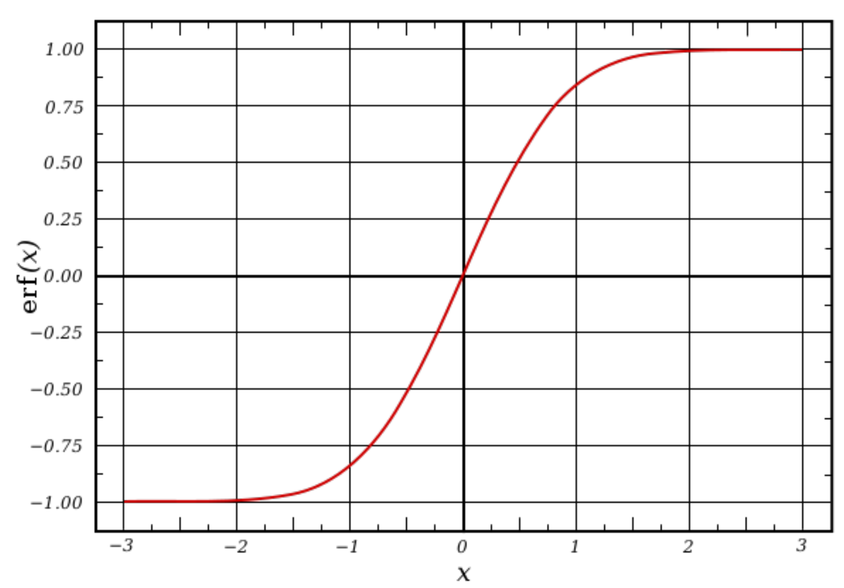
Backup

Link to Machine Learning
Neural network with 1 hidden layer

$$f(\boldsymbol{x}) = b + \sum_{i=1}^{N_H} v_i \, h(\boldsymbol{x}; \boldsymbol{u}_i)$$

$$E[f(\boldsymbol{x}) f(\boldsymbol{x}')] = \sigma_b^2 + \sum_i \sigma_v^2 \, E[h(\boldsymbol{x}; \boldsymbol{u}_i) \, h(\boldsymbol{x}'; \boldsymbol{u}_i)]$$

$$= \sigma_b^2 + N_H \, \sigma_v^2 \, E[h(\boldsymbol{x}; \boldsymbol{u}) \, h(\boldsymbol{x}'; \boldsymbol{u})]$$

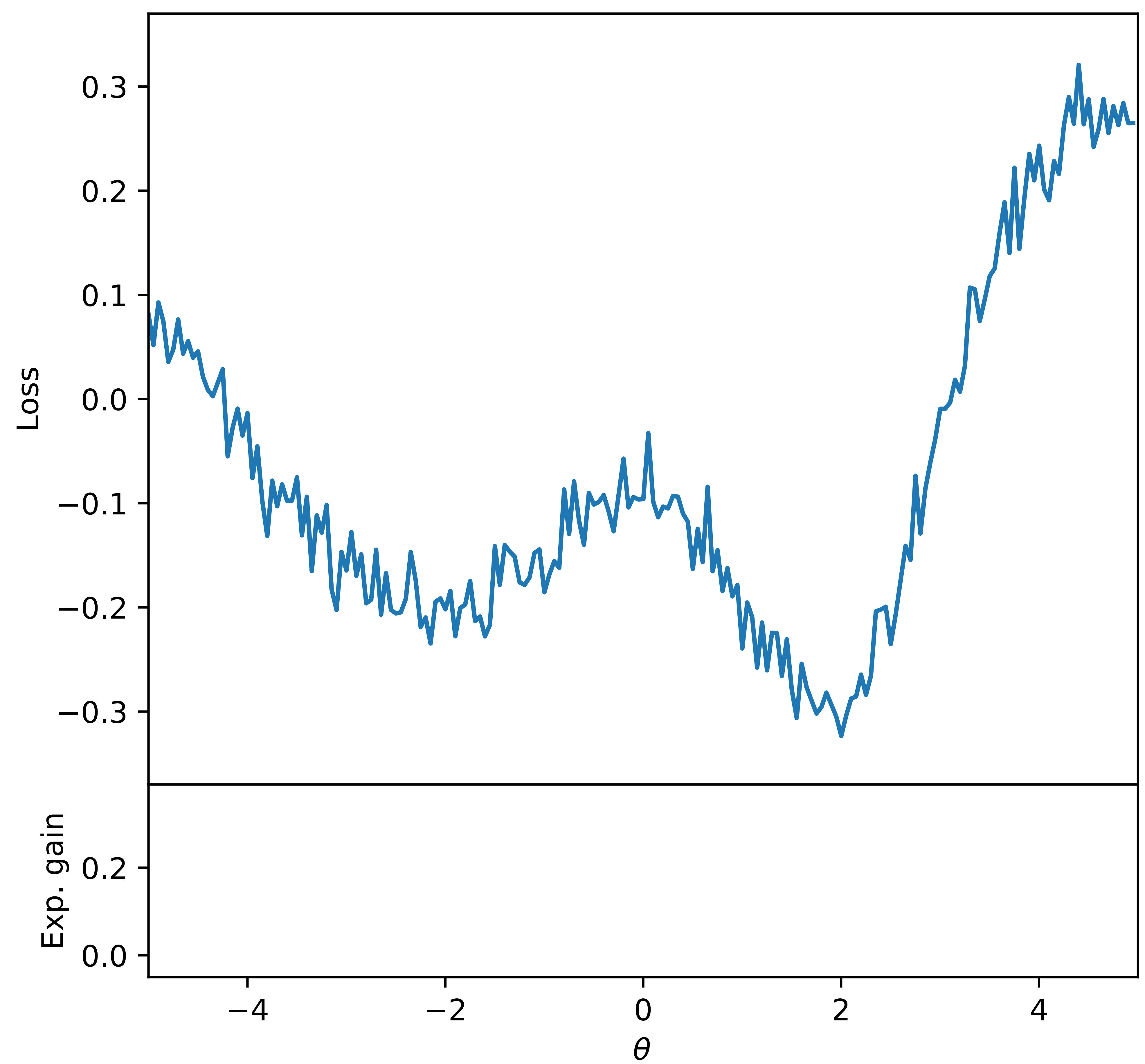


Central limit theorem: converges to a GP as $N_H \to \infty$

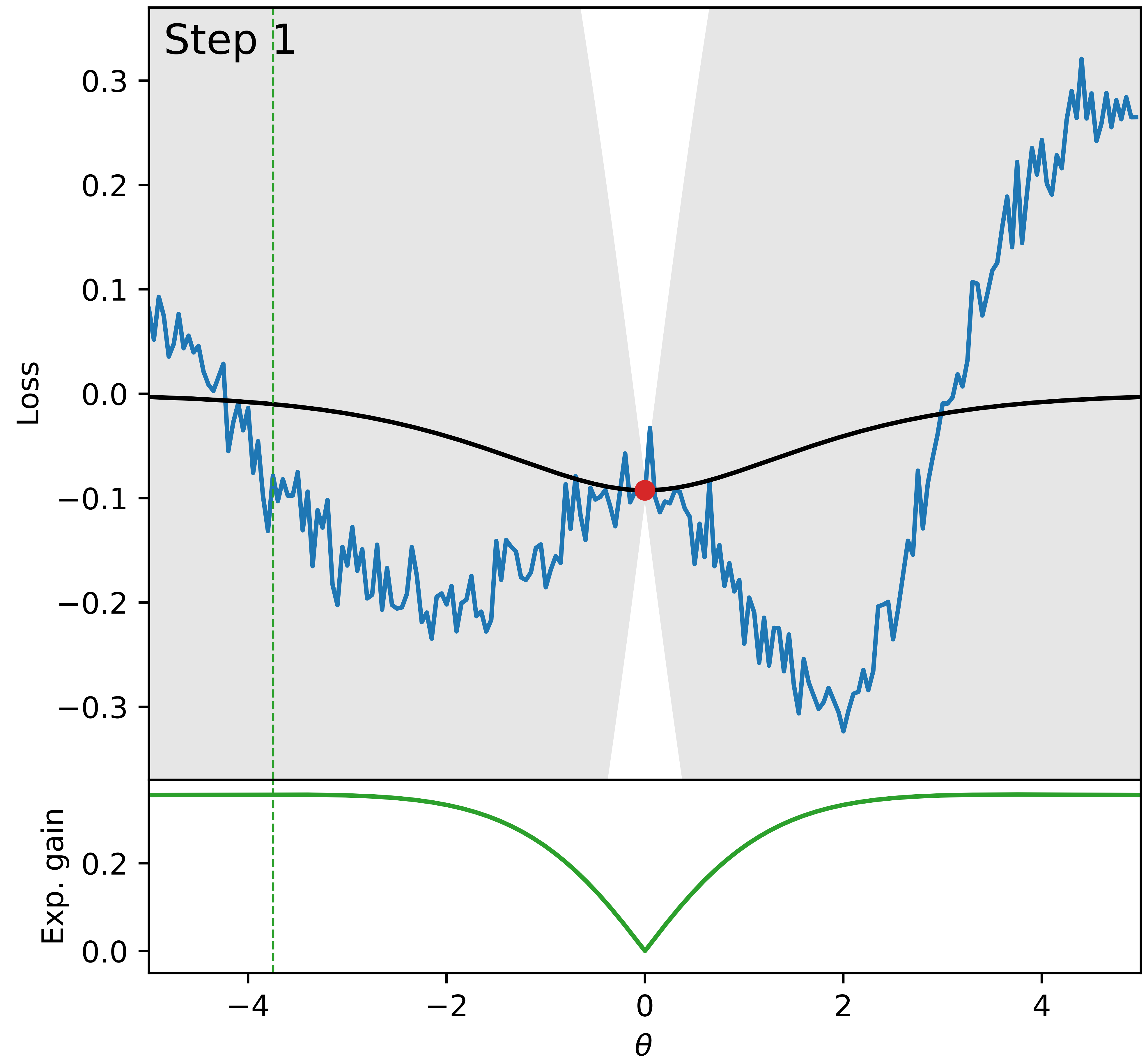
$$h(\boldsymbol{x}; \boldsymbol{u}) = \mathrm{erf}(\boldsymbol{x} \cdot \boldsymbol{u}) \quad \Rightarrow \quad k(\boldsymbol{x}, \boldsymbol{x}') = \frac{2}{\pi} \sin^{-1} \left(\frac{2\boldsymbol{x}^\top \Sigma \boldsymbol{x}'}{\sqrt{(1 + 2\boldsymbol{x}^\top \Sigma \boldsymbol{x})(1 + 2\boldsymbol{x}'^\top \Sigma \boldsymbol{x}')}} \right)$$

Application

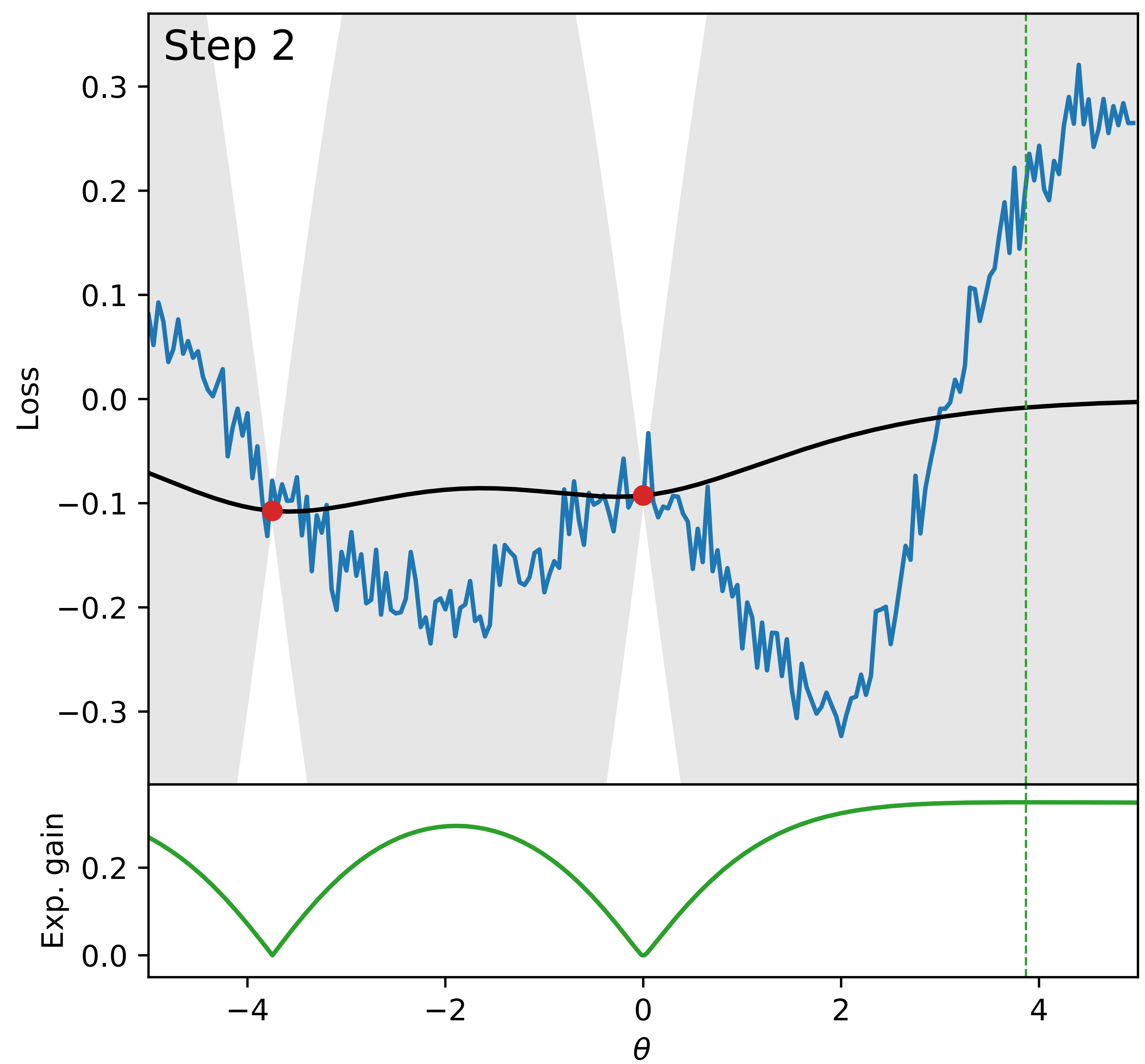
Hyperparameter Optimisation



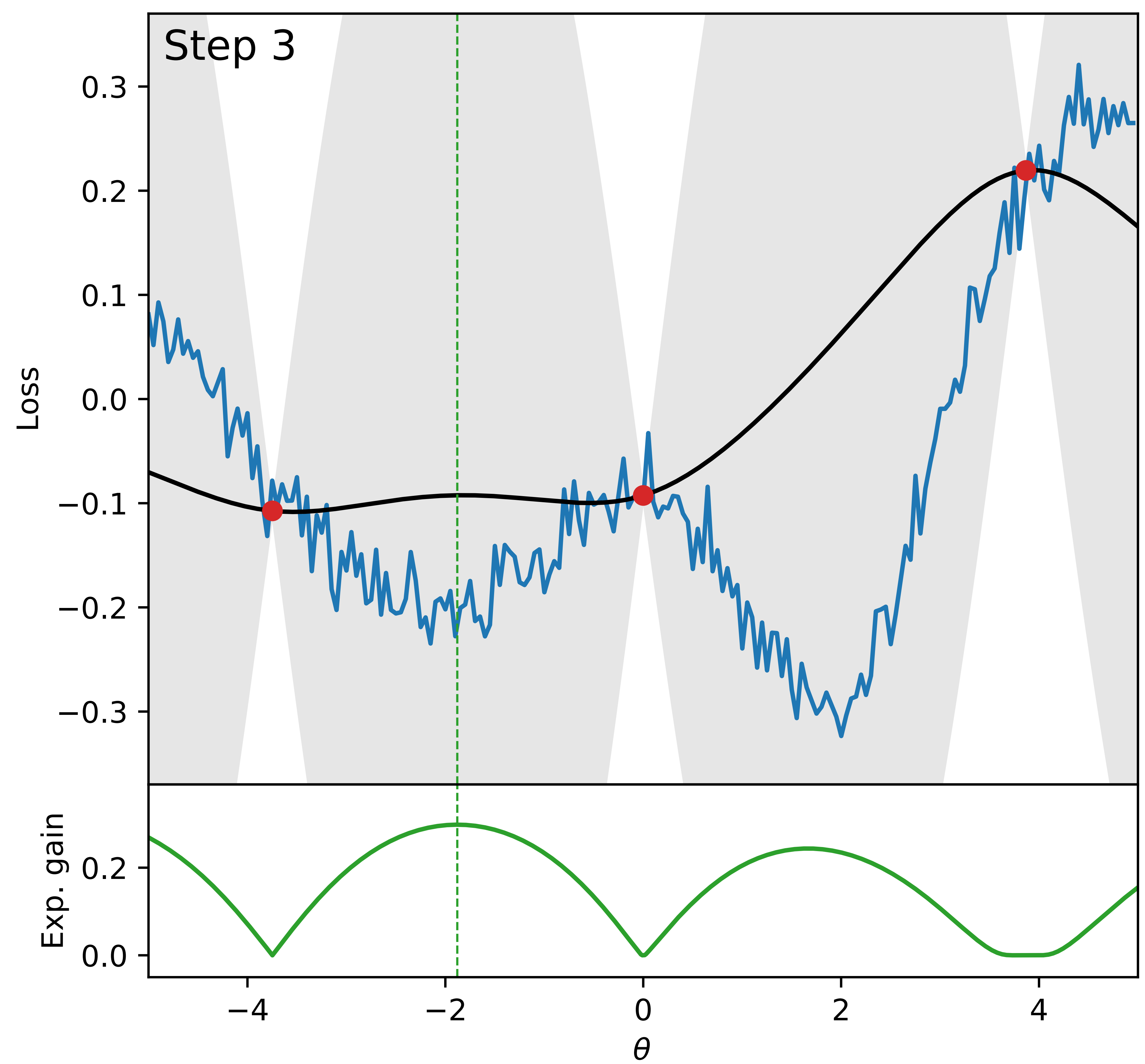
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Loss
Exp. gain
0.2
0.0
−4
−2
0
2
4
θ



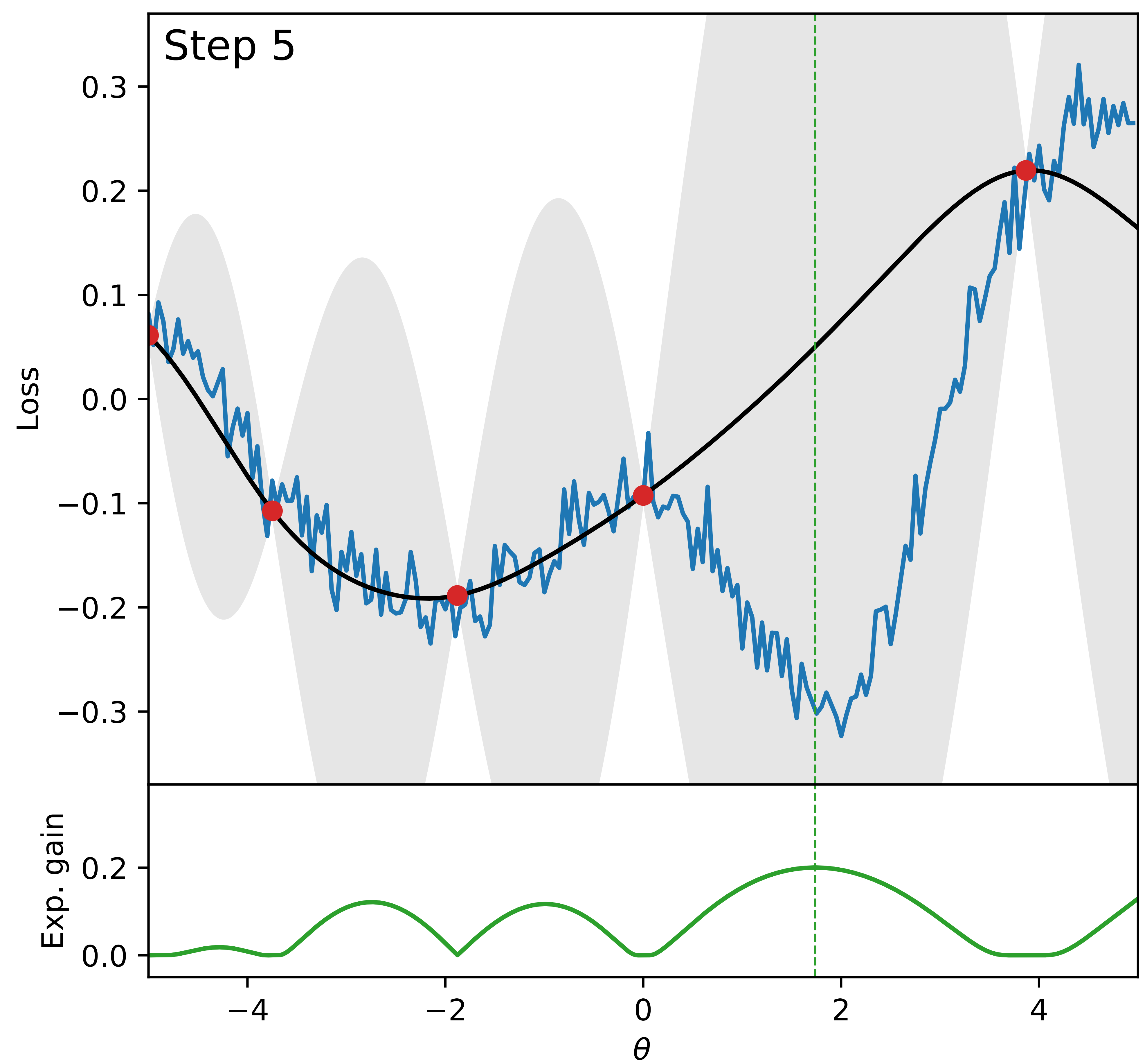
Acquisition function
See Jones et al. (1998)



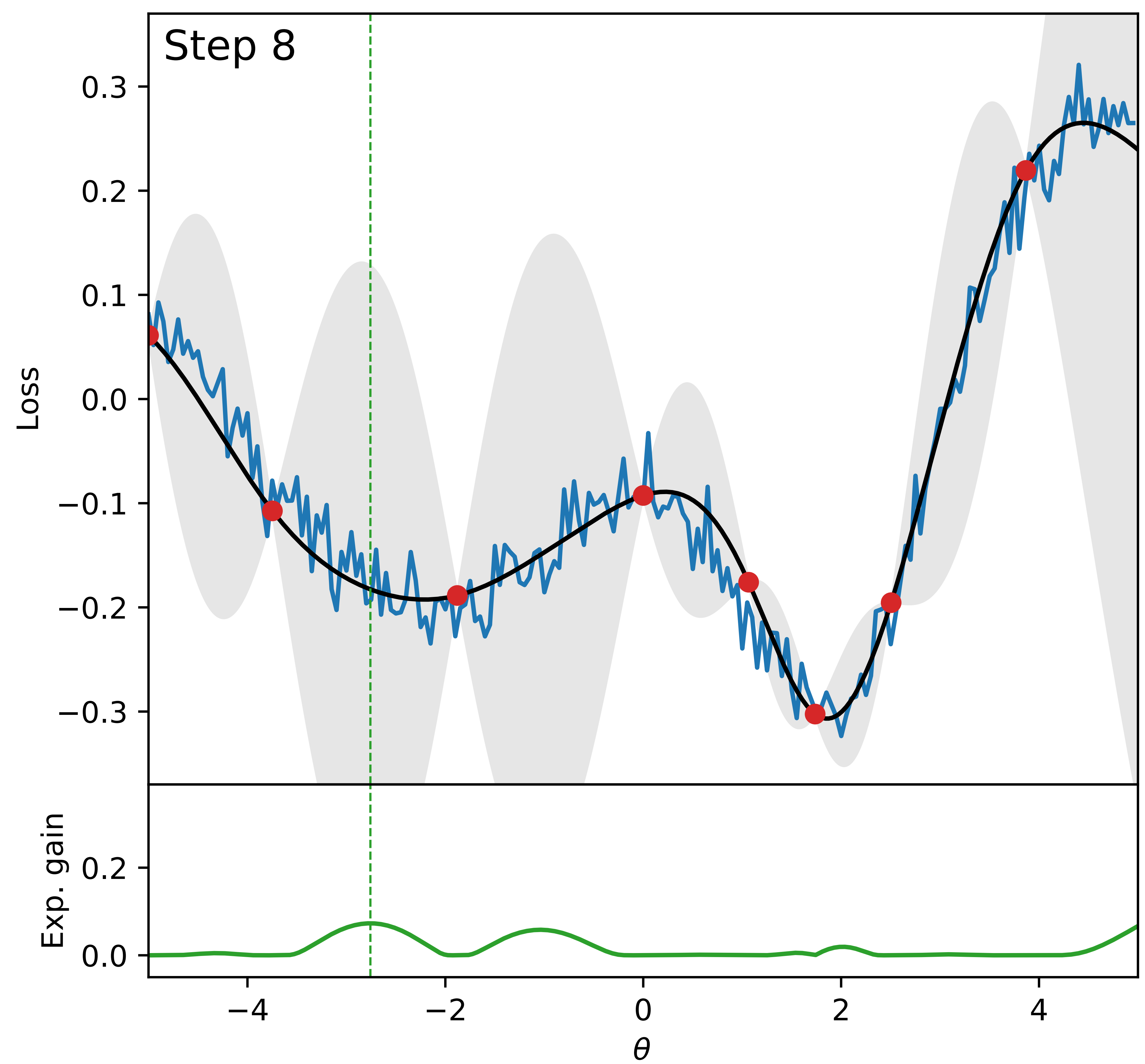
Step 2
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Loss
0.2
0.0
Exp. gain
−4
−2
0
2
4
θ



Step 3
Loss
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Exp. gain
0.2
0.0
−4
−2
0
2
4
θ



Step 5
Loss
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Exp. gain
0.2
0.0
−4
−2
0
2
4
θ



Spearmint, scikit-optimize