ENM 5310: Data-driven Modeling and Probabilistic Scientific Computing

Lecture #19: Gaussian process regression

Paris Perdikaris
April 11, 2023

UNIVERSITY *of* PENNSYLVANIA

From linear regression to GPs:

- Linear regression with inputs x_i and outputs y_i: $$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$
- Linear regression with M basis functions: $$y_i = \sum_{m=1}^{M} \beta_m \, \phi_m(x_i) + \epsilon_i$$
- Bayesian linear regression with basis functions:

$$\beta_m \sim \mathsf{N}(\cdot|0, \lambda_m) \quad \text{(independent of } \beta_\ell, \forall \ell \neq m\text{)}, \qquad \epsilon_i \sim \mathsf{N}(\cdot|0, \sigma^2)$$

- Integrating out the coefficients, β_j, we find:

$$E[y_i] = 0, \qquad Cov(y_i, y_j) = K_{ij} \stackrel{\text{def}}{=} \sum_{m=1}^{M} \lambda_m \, \phi_m(x_i) \, \phi_m(x_j) + \delta_{ij}\sigma^2$$

This is a Gaussian process with covariance function $K(x_i, x_j) = K_{ij}$.

This GP has a finite number (M) of basis functions. Many useful GP kernels correspond to infinitely many basis functions (i.e. infinite-dim feature spaces).

A multilayer perceptron (neural network) with infinitely many hidden units and Gaussian priors on the weights $\rightarrow$ a GP (Neal, 1996)

http://mlss2011.comp.nus.edu.sg/uploads/Site/lect1gp.pdf

Gaussian process covariance functions (kernels)

$p(f)$ is a Gaussian process if for *any* finite subset $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, the marginal distribution over that finite subset $p(\mathbf{f})$ has a multivariate Gaussian distribution.

Gaussian processes (GPs) are parameterized by a mean function, $\mu(x)$, and a covariance function, or kernel, $K(x, x')$.

$$p(f(x), f(x')) = \mathsf{N}(\mu, \Sigma)$$

where

$$\mu = \begin{bmatrix} \mu(x) \\ \mu(x') \end{bmatrix} \quad \Sigma = \begin{bmatrix} K(x,x) & K(x,x') \\ K(x',x) & K(x',x') \end{bmatrix}$$

and similarly for $p(f(x_1), \ldots, f(x_n))$ where now μ is an $n \times 1$ vector and Σ is an $n \times n$ matrix.

http://mlss2011.comp.nus.edu.sg/uploads/Site/lect1gp.pdf

Using Gaussian processes for nonlinear regression

Imagine observing a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^n\} = (\mathbf{X}, \mathbf{y})$.

Model:

$$
\begin{aligned}
y_i &= f(\mathbf{x}_i) + \epsilon_i \\
f &\sim \mathrm{GP}(\cdot|0, K) \\
\epsilon_i &\sim \mathrm{N}(\cdot|0, \sigma^2)
\end{aligned}
$$

Prior on f is a GP, likelihood is Gaussian, therefore posterior on f is also a GP.

We can use this to make predictions

$$p(y_*|\mathbf{x}_*, \mathcal{D}) = \int p(y_*|\mathbf{x}_*, f, \mathcal{D})\, p(f|\mathcal{D})\, df$$

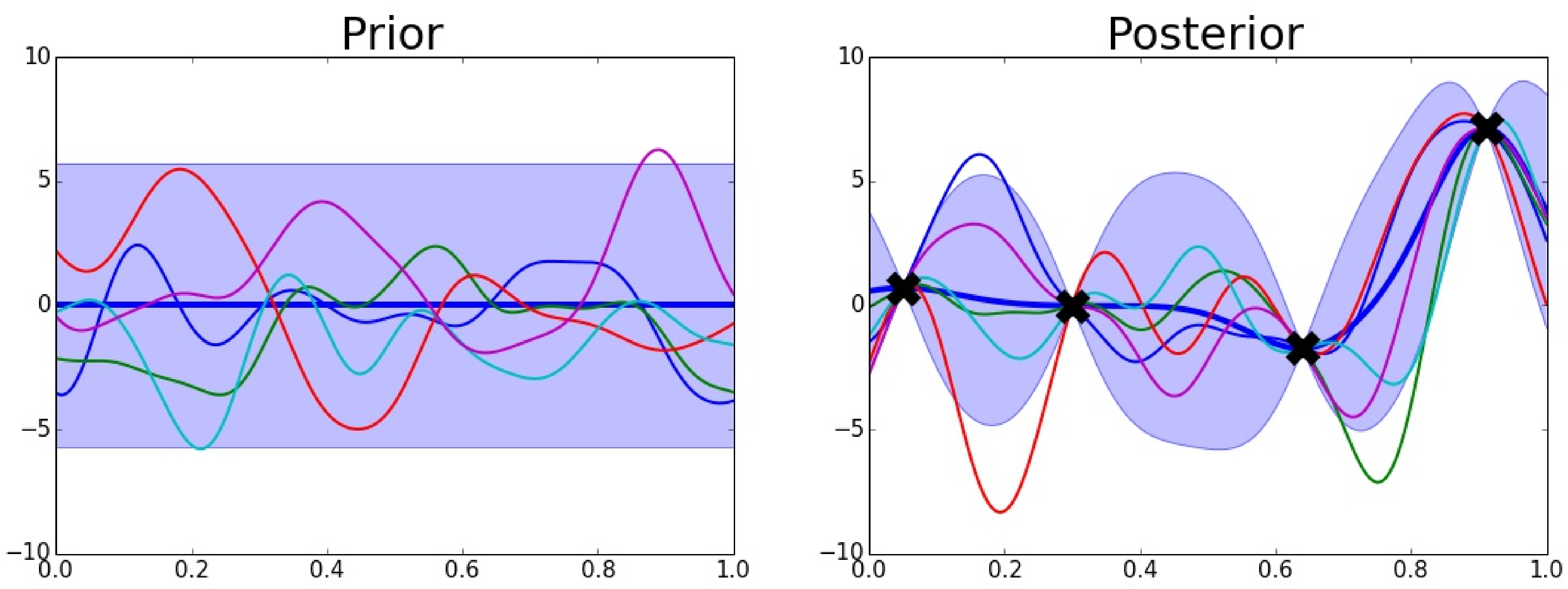
We can also compute the marginal likelihood (evidence) and use this to compare or tune covariance functions

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|f, \mathbf{X})\, p(f)\, df$$

http://mlss2011.comp.nus.edu.sg/uploads/Site/lect1gp.pdf

Data-driven modeling with Gaussian processes

$$y = f(\boldsymbol{x}) + \epsilon \qquad f \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}))$$



Training via maximizing the marginal likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I}| - \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi$$

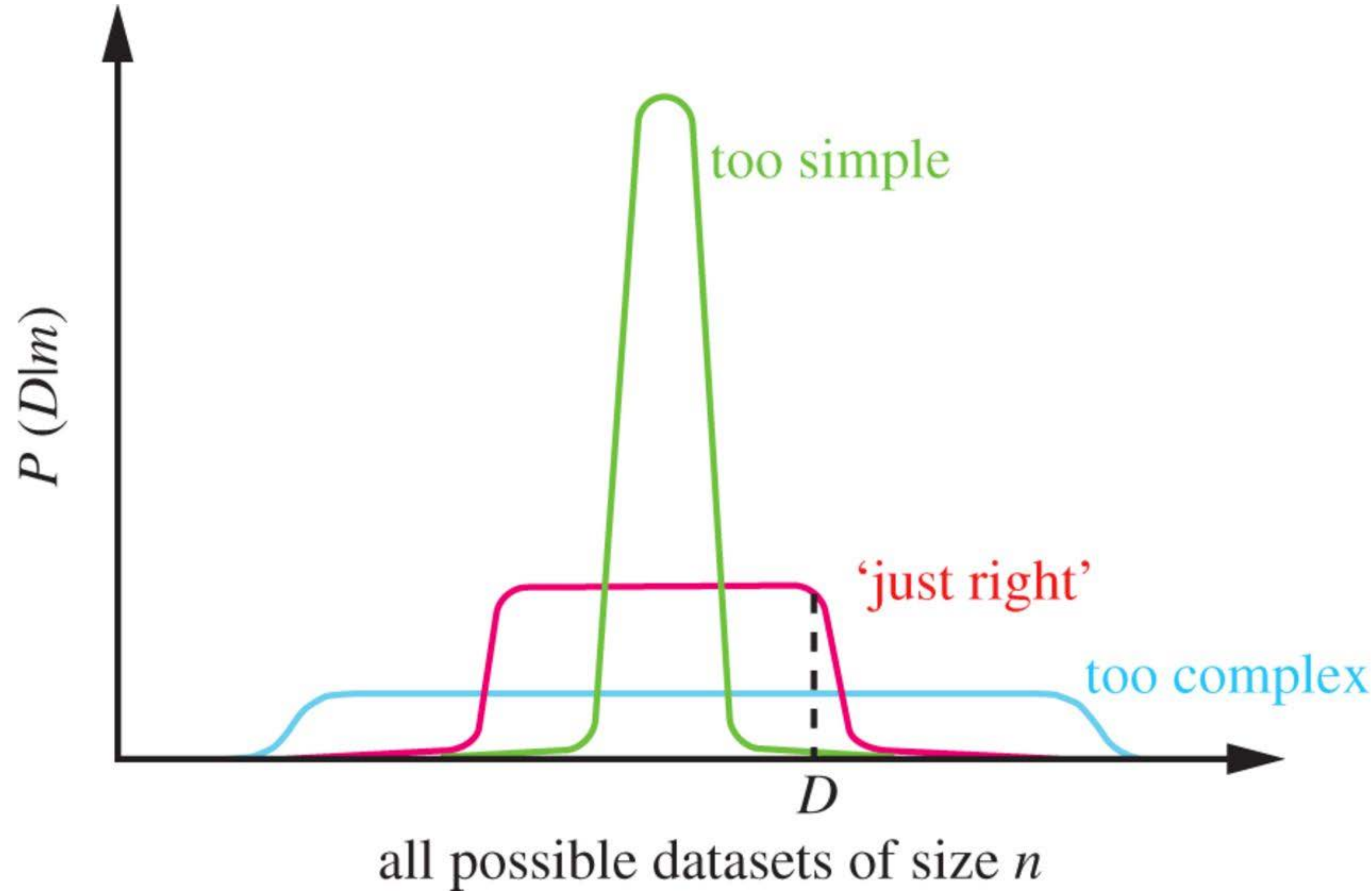
Prediction via conditioning on available data

$$\begin{aligned} p(f_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) &= \mathcal{N}(f_*|\mu_*, \sigma_*^2), \\ \mu_*(\boldsymbol{x}_*) &= \boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \\ \sigma_*^2(\boldsymbol{x}_*) &= \boldsymbol{k}_{**} - \boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{k}_{N*}, \end{aligned}$$

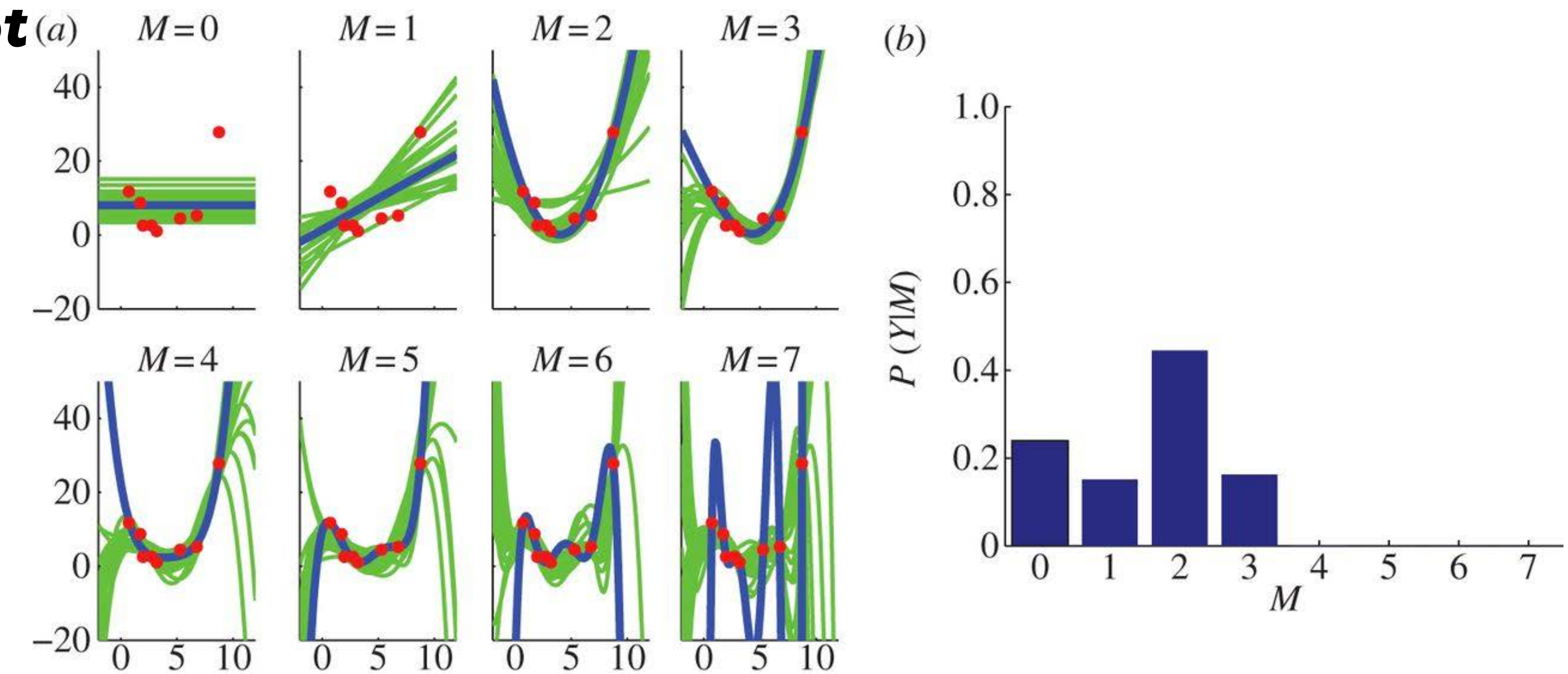
Demo code: https://github.com/PredictiveIntelligenceLab/GPTutorial

Occam's razor

William of Ockham (~1285-1347 A.D)

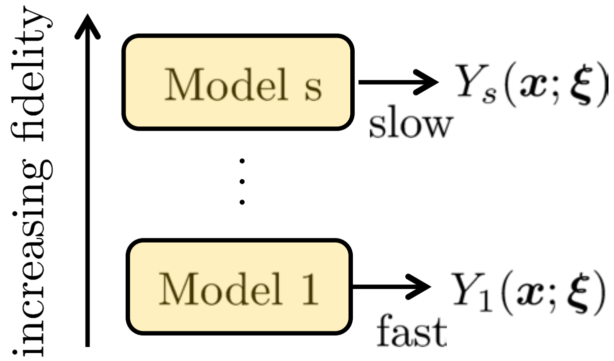


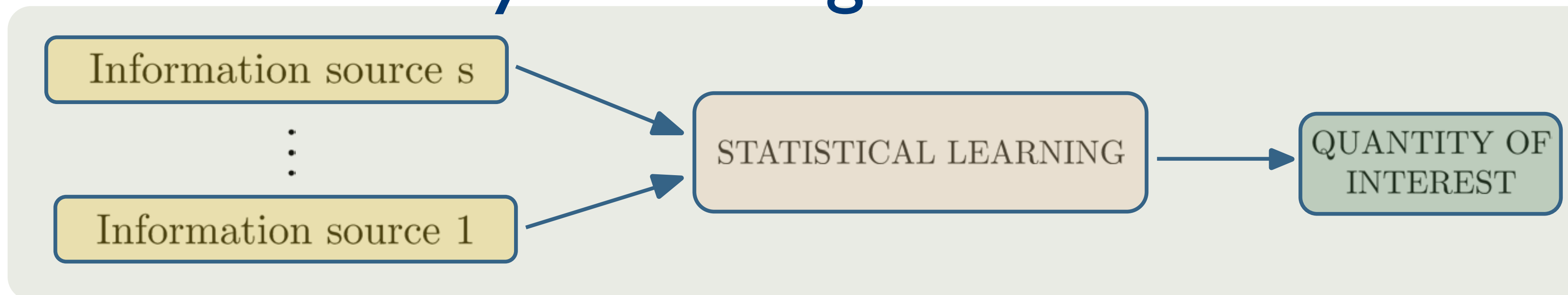
"plurality should not be posited without necessity."



Ghahramani, Z. (2013). Bayesian non-parametrics and the probabilistic approach to modelling. Phil. Trans. R. Soc. A, 371(1984), 20110553.

Multi-fidelity modeling

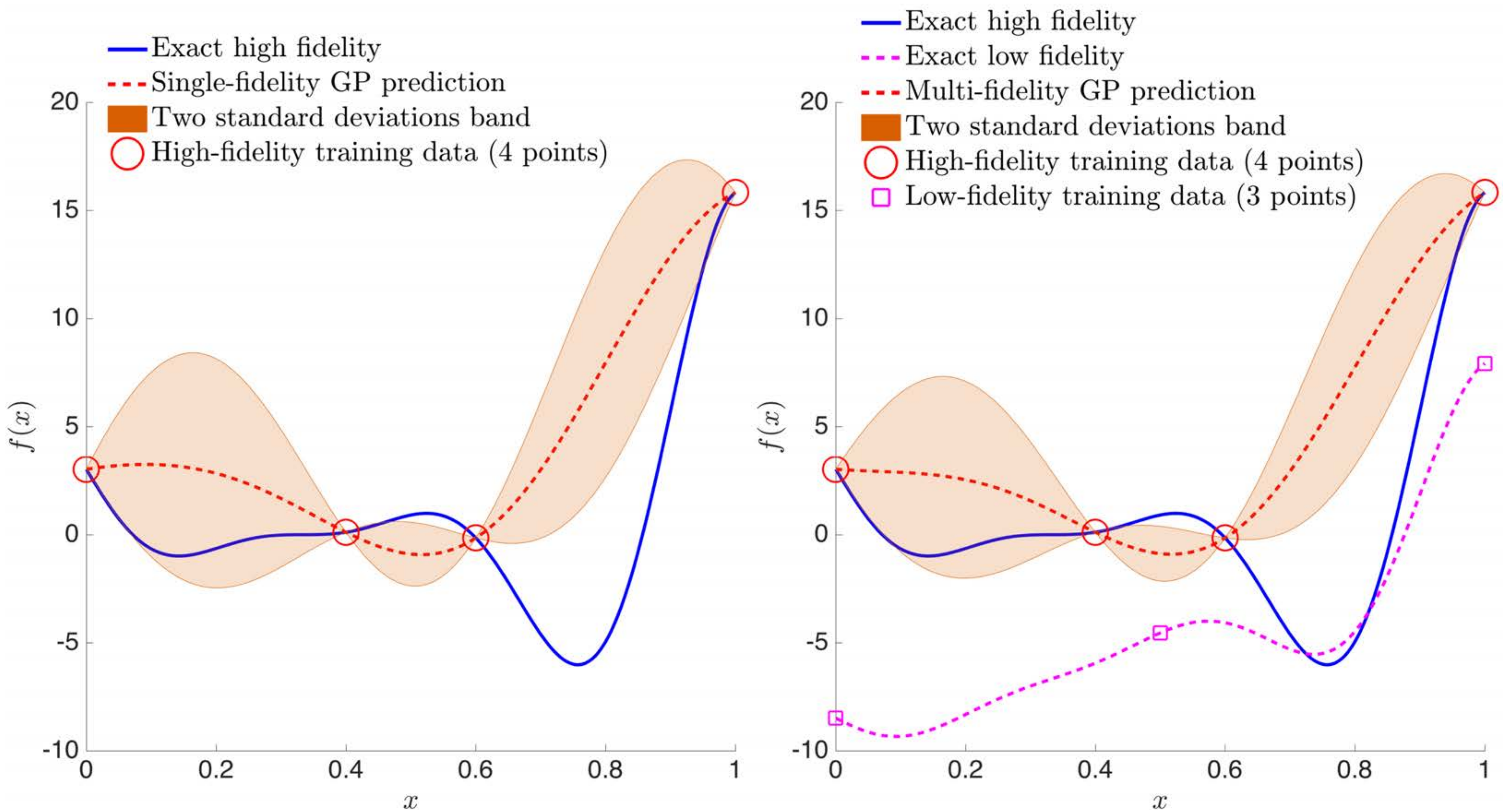




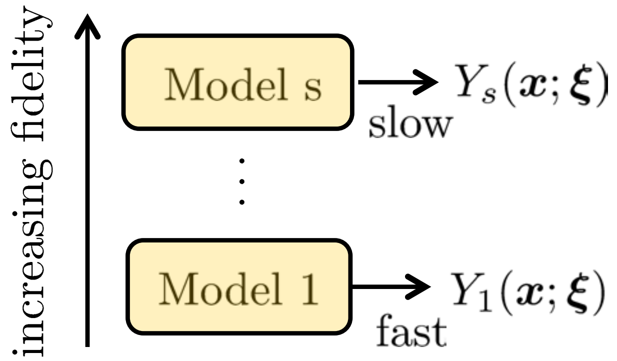
Number of runs is limited by time and computational resources

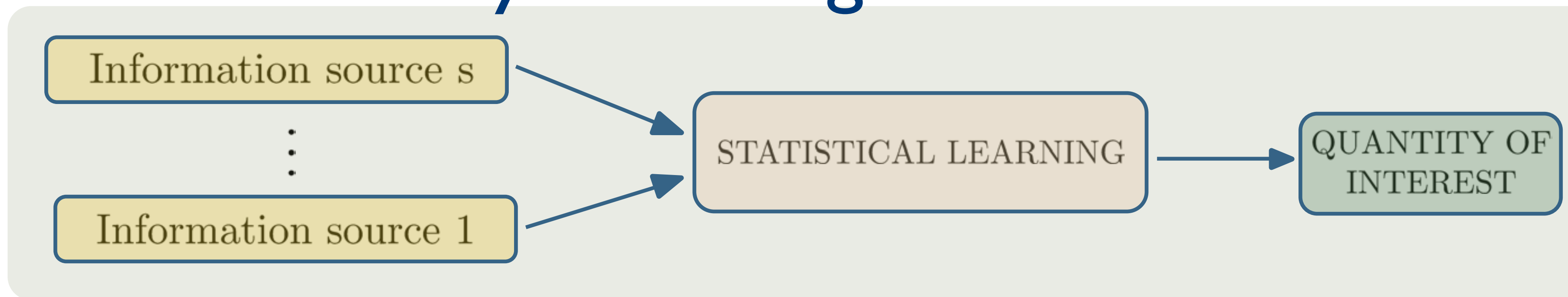
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference



Multi-fidelity modeling

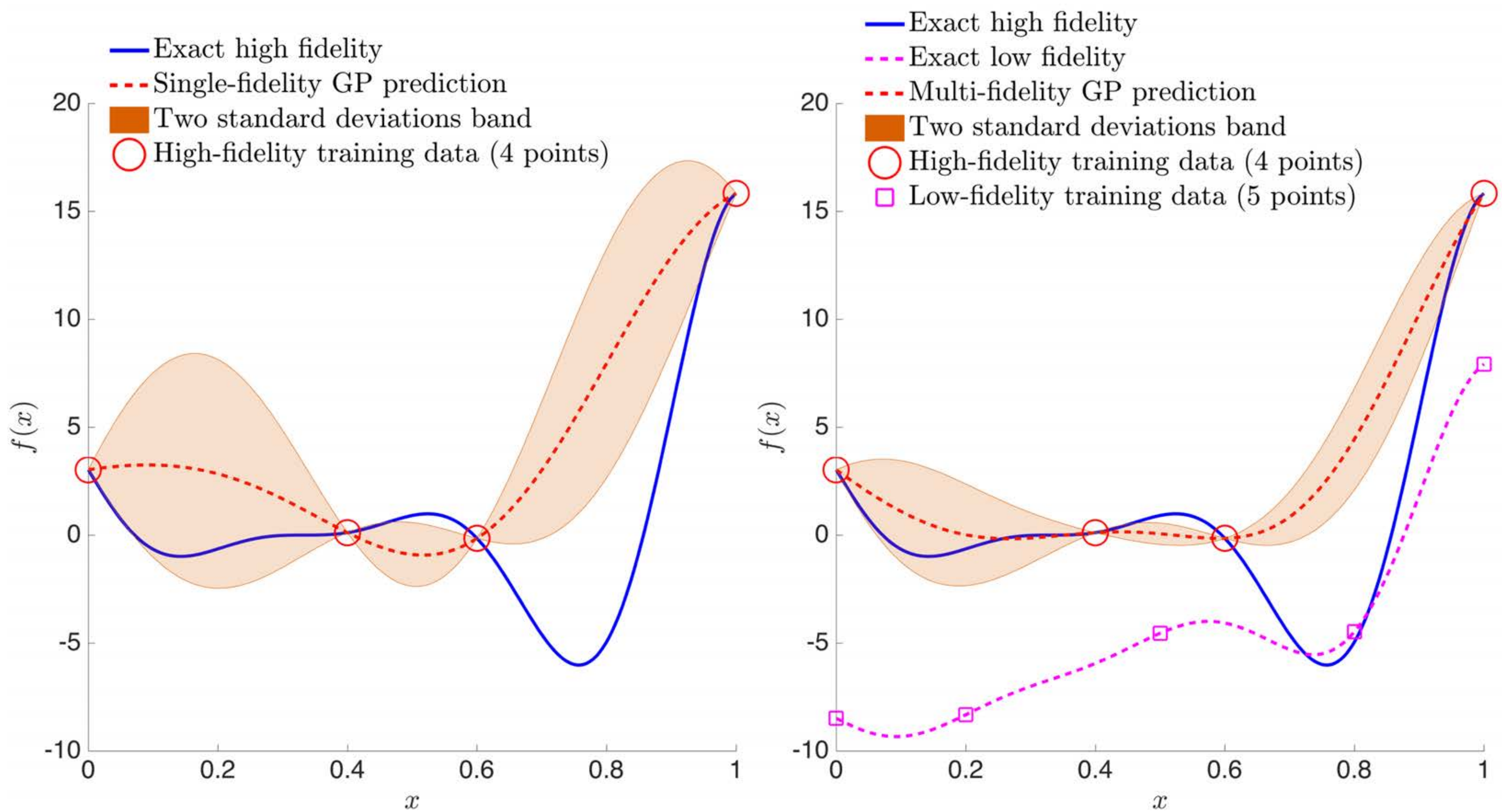




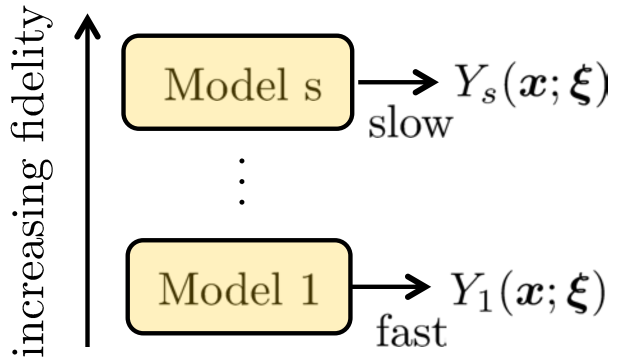
Number of runs is limited by time
and computational resources

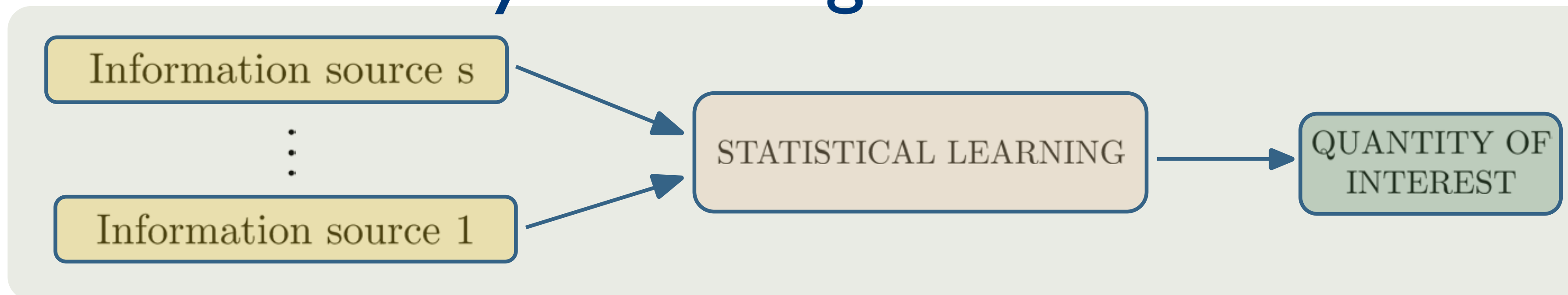
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a
problem of statistical inference



Multi-fidelity modeling

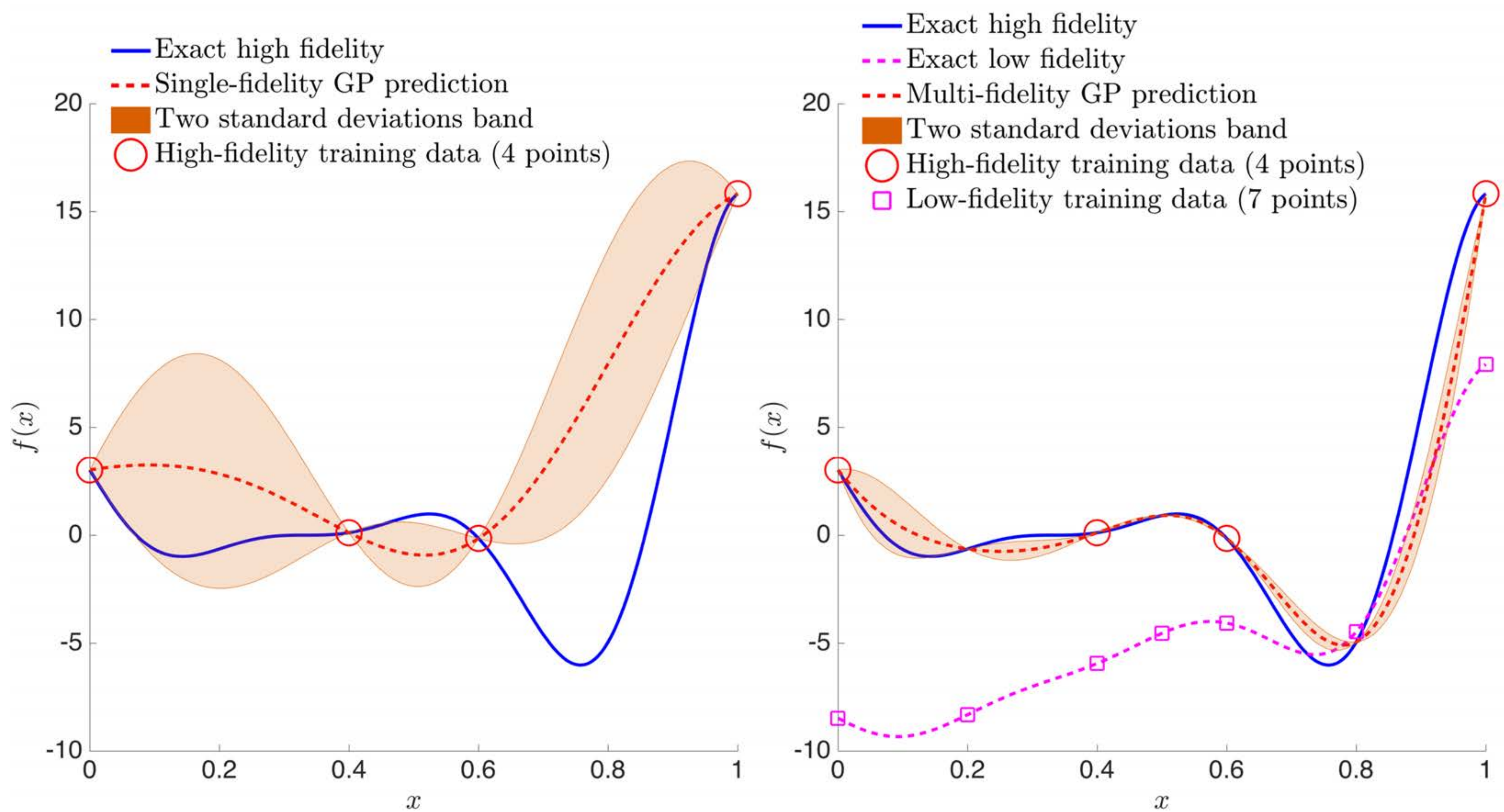




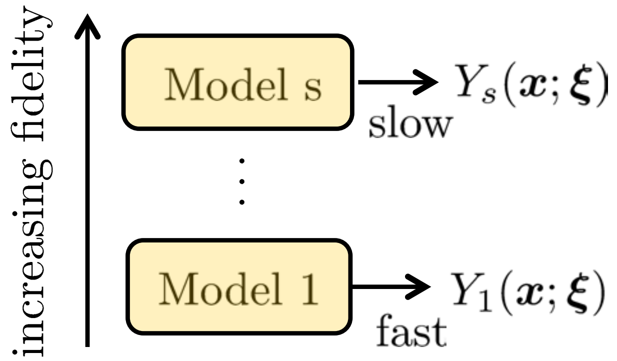
Number of runs is limited by time
and computational resources

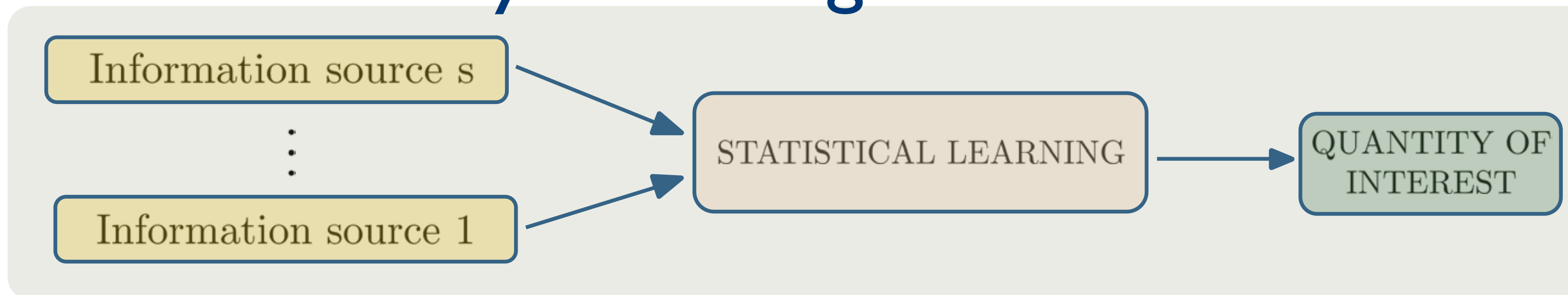
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a
problem of statistical inference



Multi-fidelity modeling

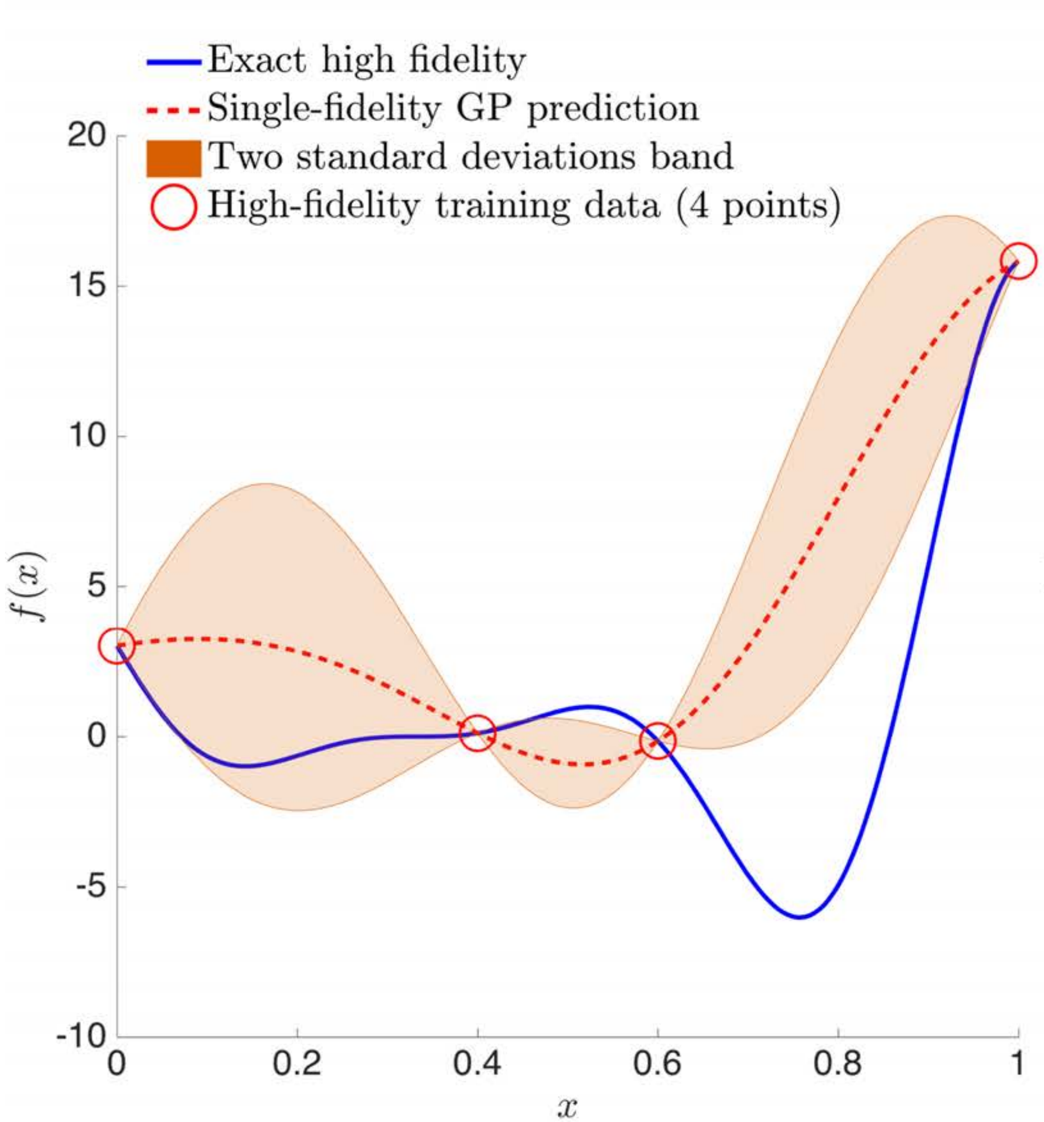


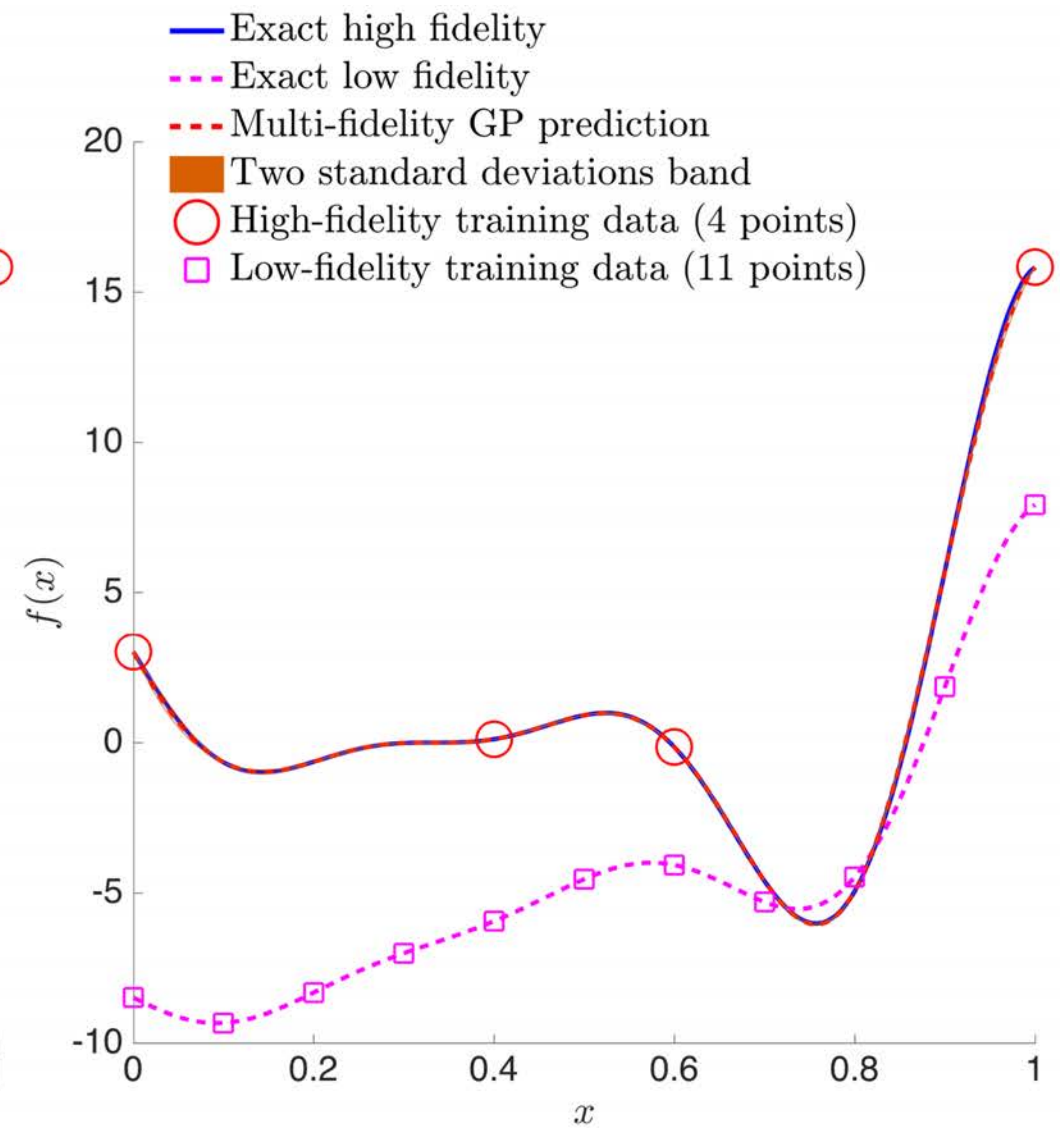


Number of runs is limited by time and computational resources

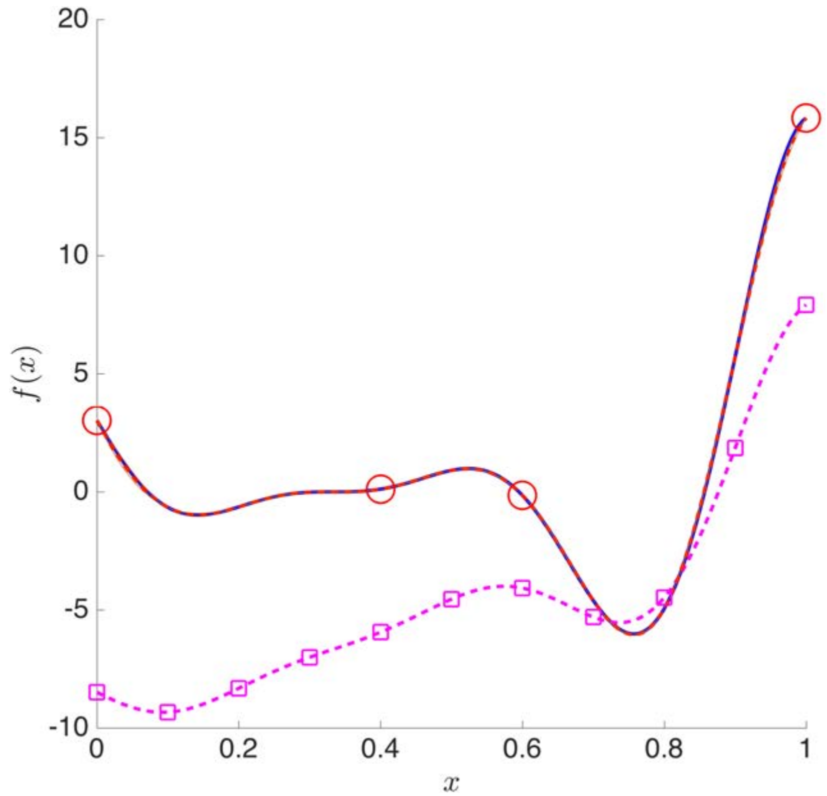
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference





Multi-fidelity modeling



Multi-fidelity observations:

$$\boldsymbol{y}_L = f_L(\boldsymbol{x}_L) + \boldsymbol{\epsilon}_L$$
$$\boldsymbol{y}_H = f_H(\boldsymbol{x}_H) + \boldsymbol{\epsilon}_H$$

Probabilistic model:

$$f_H(\boldsymbol{x}) = \rho f_L(\boldsymbol{x}) + \delta(\boldsymbol{x})$$
$$f_L(\boldsymbol{x}) \sim \mathcal{GP}(0, k_L(\boldsymbol{x}, \boldsymbol{x}'; \theta_L))$$
$$\delta(\boldsymbol{x}) \sim \mathcal{GP}(0, k_H(\boldsymbol{x}, \boldsymbol{x}'; \theta_H))$$
$$\epsilon_L \sim \mathcal{N}(0, \sigma^2_{\epsilon_L} \boldsymbol{I})$$
$$\epsilon_H \sim \mathcal{N}(0, \sigma^2_{\epsilon_H} \boldsymbol{I})$$

Training:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_L \\ \boldsymbol{y}_H \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} k_L(\boldsymbol{x}_L, \boldsymbol{x}'_L; \theta_L) + \sigma^2_{\epsilon_L} \boldsymbol{I} & \rho k_L(\boldsymbol{x}_L, \boldsymbol{x}'_H; \theta_L) \\ \rho k_L(\boldsymbol{x}_H, \boldsymbol{x}'_L; \theta_L) & \rho^2 k_L(\boldsymbol{x}_H, \boldsymbol{x}'_H; \theta_L) + k_H(\boldsymbol{x}_H, \boldsymbol{x}'_H; \theta_H) + \sigma^2_{\epsilon_H} \boldsymbol{I} \end{bmatrix} \right)$$

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_L \\ \boldsymbol{x}_H \end{bmatrix} \qquad -\log p(\boldsymbol{y} | \boldsymbol{X}, \theta_L, \theta_H, \rho, \sigma^2_{\epsilon_L}, \sigma^2_{\epsilon_H}) = \frac{1}{2} \log |\boldsymbol{K}| + \frac{1}{2} \boldsymbol{y}^T \boldsymbol{K}^{-1} \boldsymbol{y} - \frac{N_L + N_H}{2} \log 2\pi$$

Prediction:

$$p(f(\boldsymbol{x}^*) | \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}^*) \sim \mathcal{N}(f(\boldsymbol{x}^*) | \mu(\boldsymbol{x}^*), \sigma^2(\boldsymbol{x}^*))$$
$$\mu(\boldsymbol{x}^*) = \boldsymbol{k}(\boldsymbol{x}^*, \boldsymbol{X}) \boldsymbol{K}^{-1} \boldsymbol{y}$$
$$\sigma(\boldsymbol{x}^*) = \boldsymbol{k}(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}(\boldsymbol{x}^*, \boldsymbol{X}) \boldsymbol{K}^{-1} \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}^*)$$

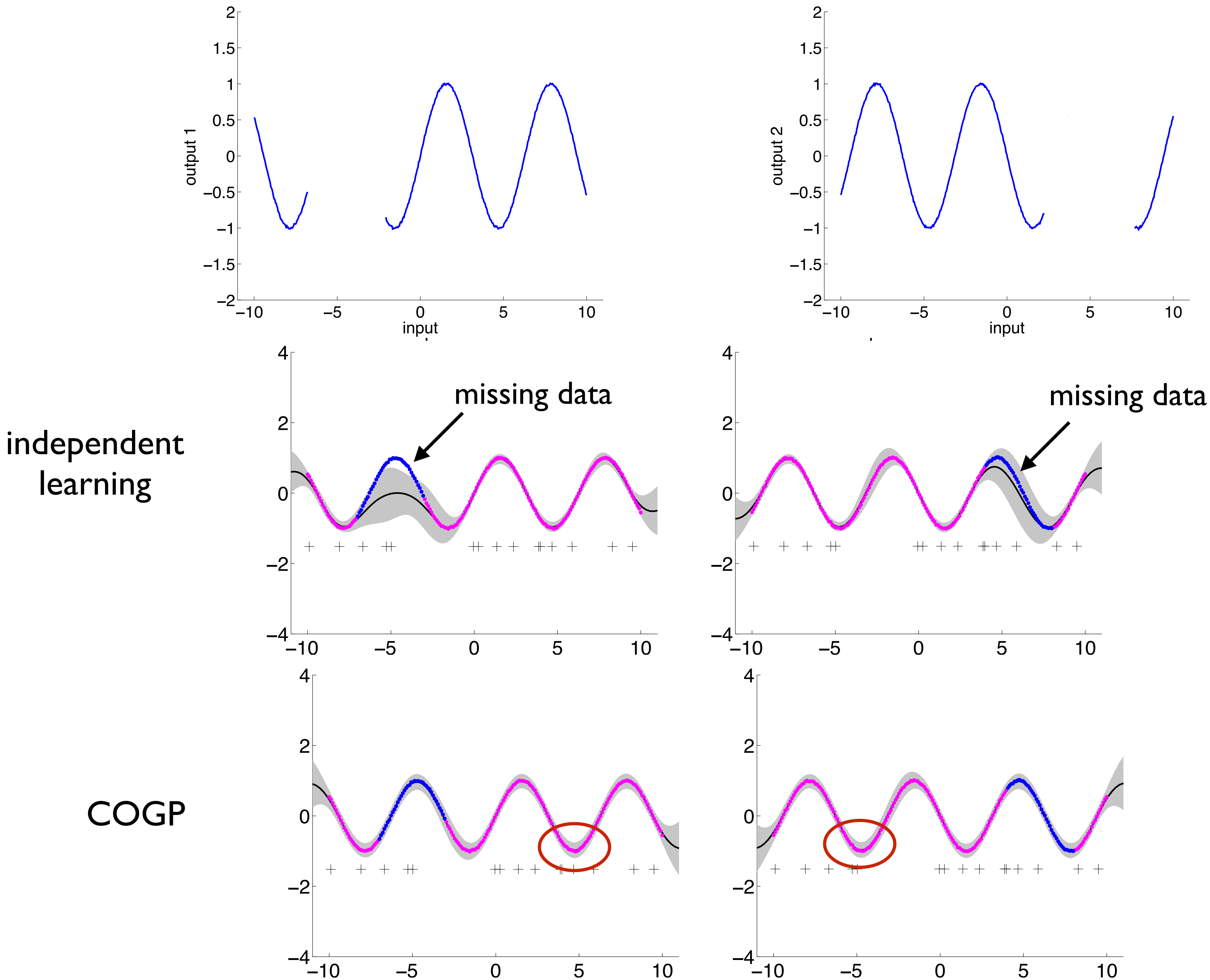
M.C Kennedy, and A. O'Hagan. *Predicting the output from a complex computer code when fast approximations are available*, 2000.

Demo code: https://github.com/PredictiveIntelligenceLab/GPTutorial

Multi-output Gaussian process regression

Learn two correlated tasks (outputs) with lots of data + missing data

output 2 = - output 1



Nguyen, T.V., & Bonilla, E.V. (2014, July). Collaborative Multi-output Gaussian Processes. In UAI (pp. 643-652).

Linear constraints

$$f \sim \mathcal{GP}\left(0, k(x, x'; \theta)\right)$$

$$\Downarrow$$

$$\mathcal{L}f \sim \mathcal{GP}\left(0, \mathcal{L}_x \mathcal{L}_{x'} k(x, x'; \theta)\right)$$