

Gaussian processes

A hands-on tutorial

Slides and code: <https://github.com/paraklas/GPTutorial>

Paris Perdikaris

Massachusetts Institute of Technology, Department of Mechanical Engineering

Web: <http://web.mit.edu/parisp/www/>

Email: parisp@mit.edu

ICERM
Providence, RI
June 5th, 2017



Data-driven modeling with Gaussian processes

Tuesday **June 6, 2017**

Wednesday **June 7,
2017**

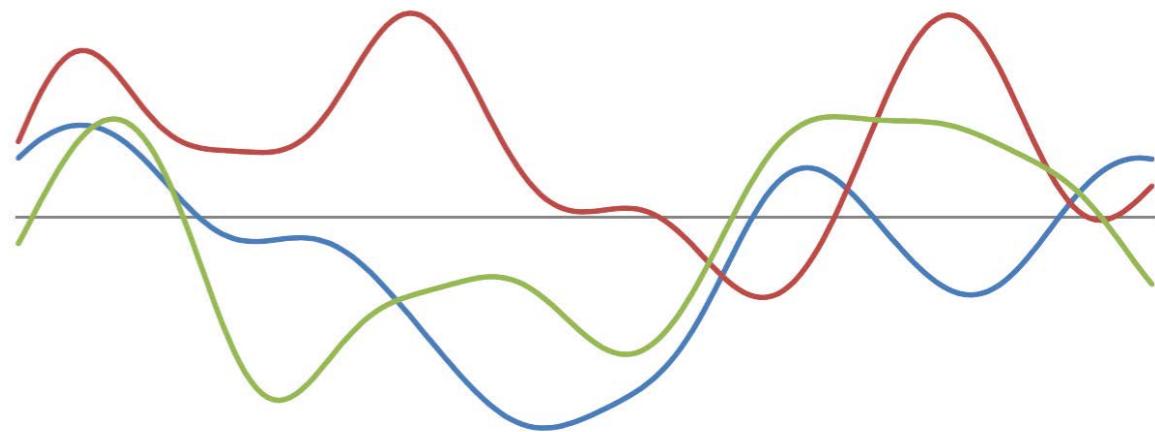
Time	Description	Speaker	Time	Description	Speaker			
9:00 – 9:45	Probabilistic Dimensionality Reduction	Neil Lawrence, University of Sheffield and Amazon Research Cambridge	9:00 – 9:45	Bayesian Calibration of Simulators with Structured Discretization Uncertainty	Oksana Chkrebtii, The Ohio State University	2:30 – 3:15	Bayesian Probabilistic Numerical Methods. (Part I)	Chris Oates, Newcastle University
10:00 – 10:30	Coffee Break		10:30 – 11:15	Numerical Gaussian Processes for Time- dependent and Non- linear Partial Differential Equations	Maziar Raissi, Brown University	3:30 – 4:00	Coffee/Tea Break	
10:30 – 11:15	Bayesian optimization for automating model selection	Roman Garnett, Washington University in St. Louis	11:30 – 12:15	Variational Reformulation of the Uncertainty Propagation Problem in Linear Partial Differential Equations	Ilias Bilionis, Purdue University	4:00 – 4:45	Bayesian Probabilistic Numerical Methods. (Part II)	Jon Cockayne, University of Warwick

*GPs will be mentioned
in ~50% of the
workshop talks!*

Data-driven modeling with Gaussian processes

"The linear algebra of computation under uncertainty"

Priors over functions: $f \sim \mathcal{GP}(\mu(x), K(\mathbf{x}, \mathbf{x}'; \theta))$



Samples from a GP prior

Marginalization:

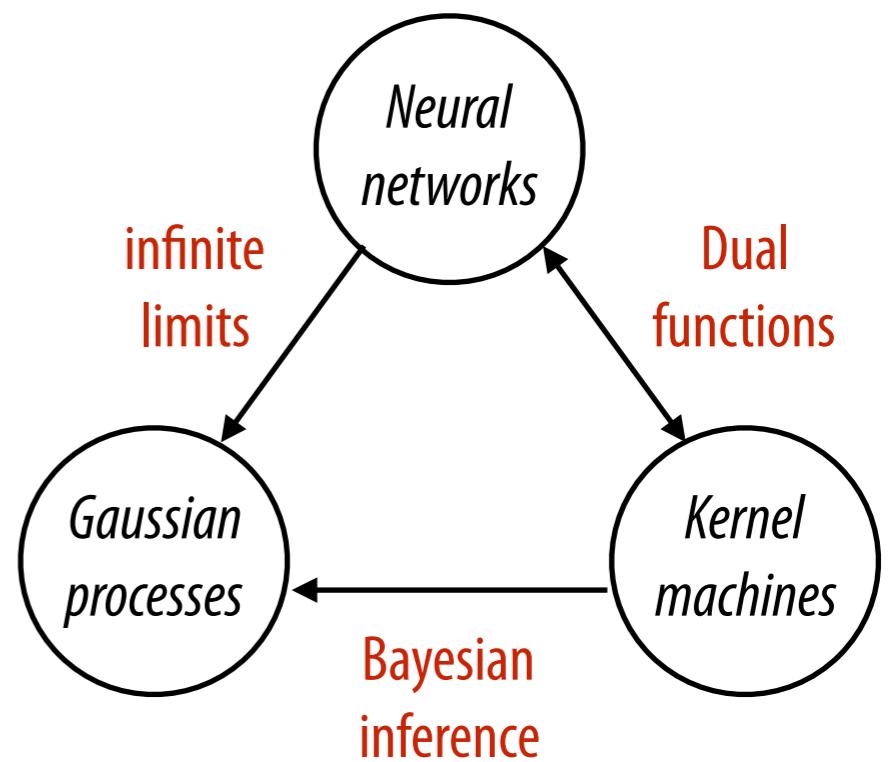
$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. Then:

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) d\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Posterior is also Gaussian:

$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. Then:

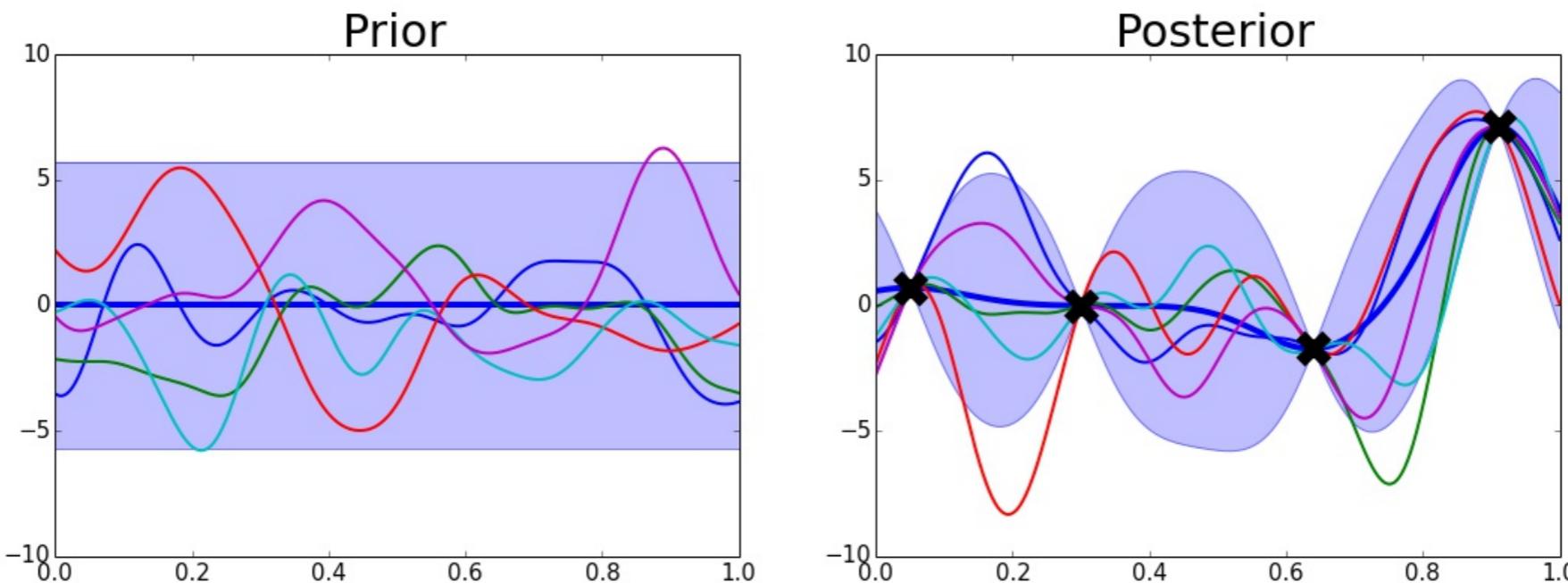
$$p(\mathbf{f}_A | \mathbf{f}_B) = \mathcal{N}(\boldsymbol{\mu}_A + \mathbf{K}_{AB} \mathbf{K}_{BB}^{-1} (\mathbf{f}_B - \boldsymbol{\mu}_B), \mathbf{K}_{AA} - \mathbf{K}_{AB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{BA})$$



Data-driven modeling with Gaussian processes

$$y = f(\mathbf{x}) + \epsilon$$

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



Training via maximizing the marginal likelihood

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

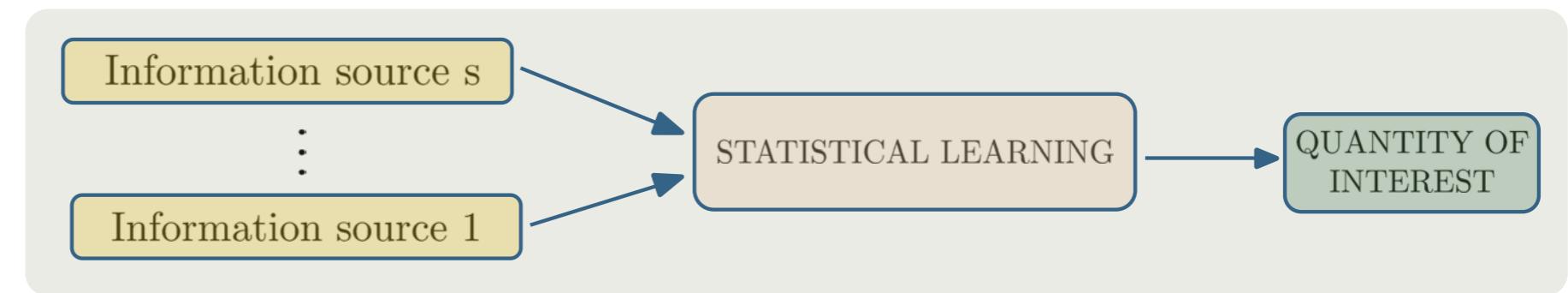
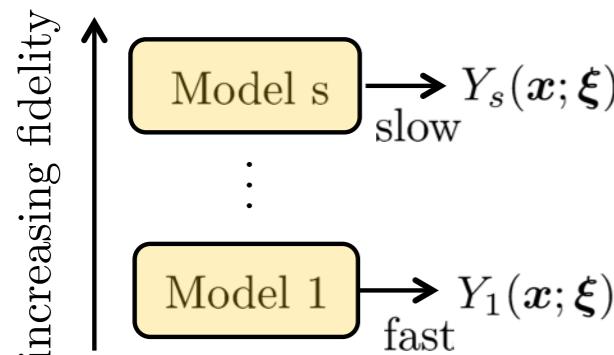
Prediction via conditioning on available data

$$p(f_* | \mathbf{y}, \mathbf{X}, \mathbf{x}_*) = \mathcal{N}(f_* | \mu_*, \sigma_*^2),$$

$$\mu_*(\mathbf{x}_*) = \mathbf{k}_{*N} (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y},$$

$$\sigma_*^2(\mathbf{x}_*) = \mathbf{k}_{**} - \mathbf{k}_{*N} (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{k}_{N*},$$

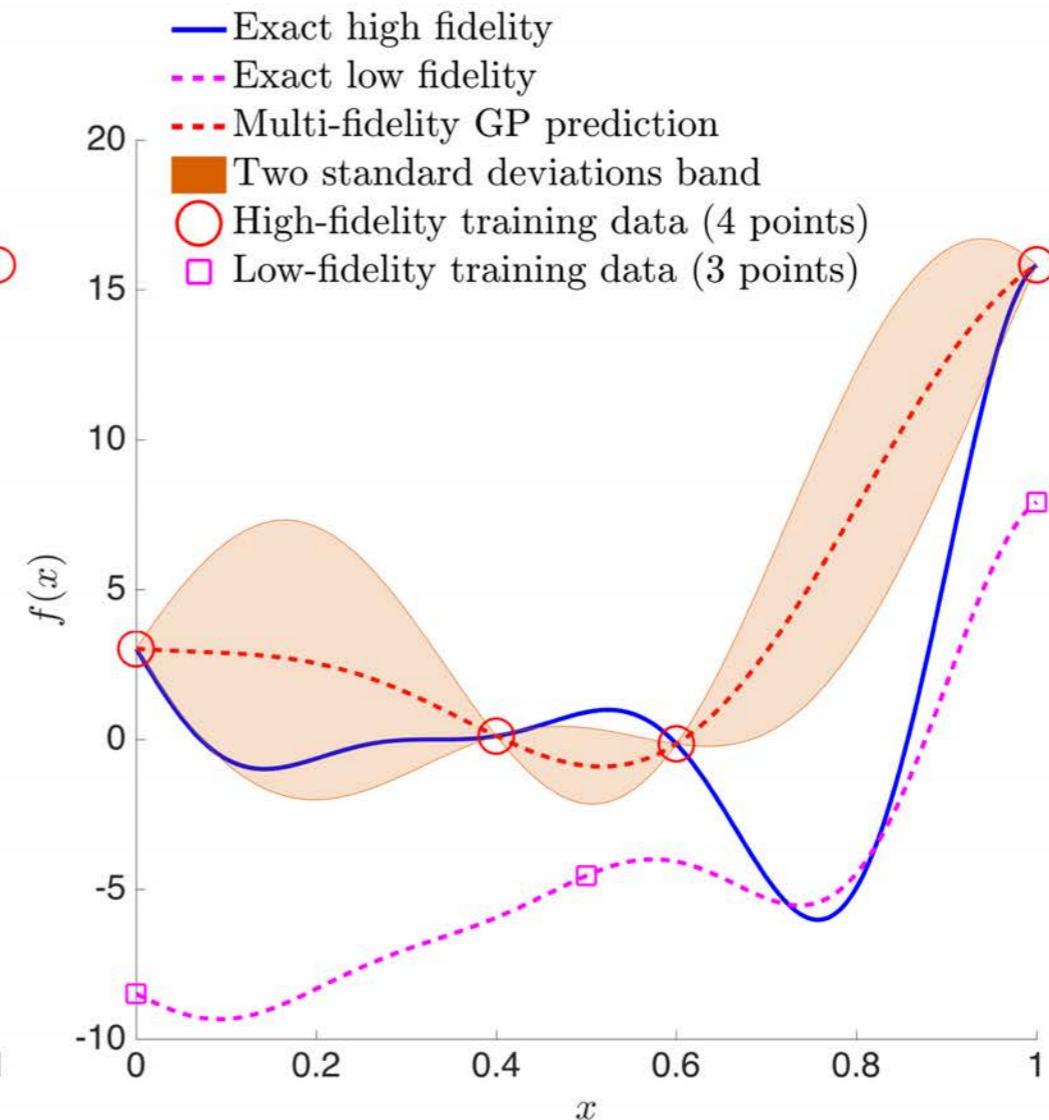
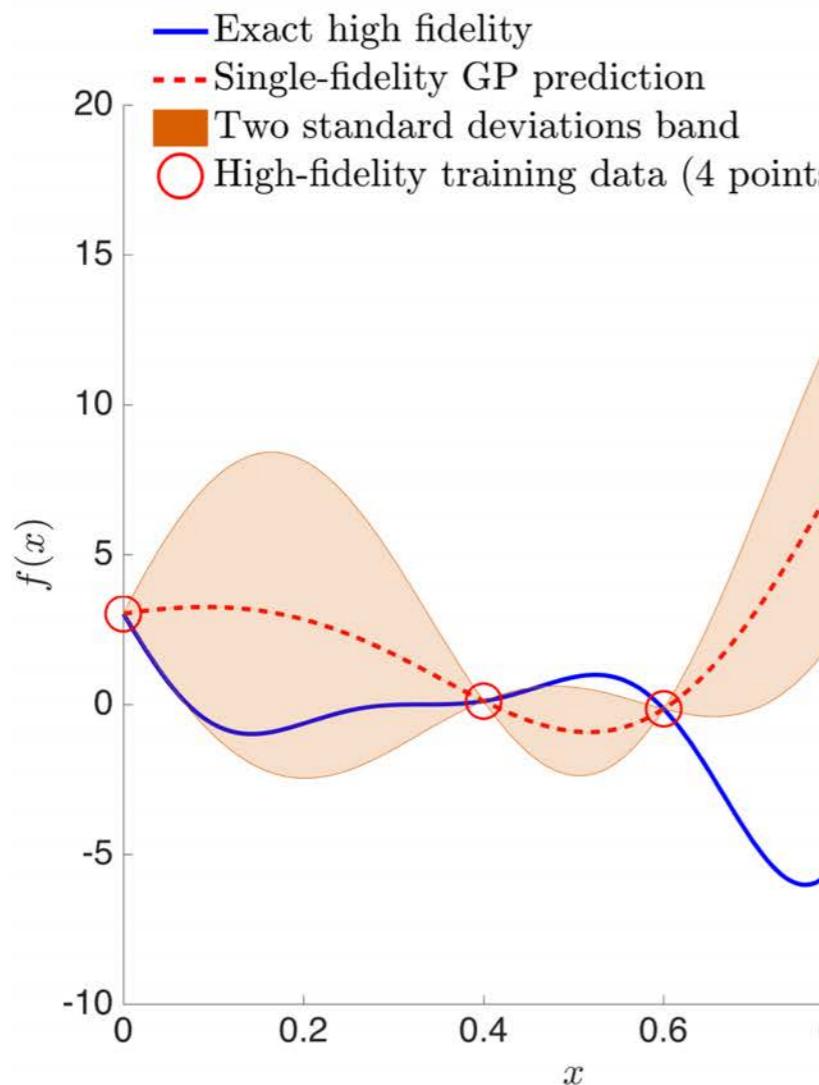
Multi-fidelity modeling



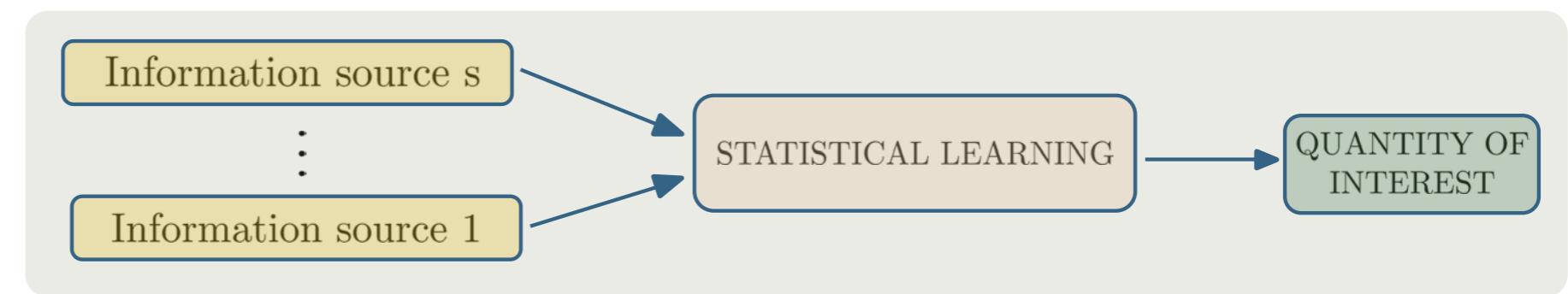
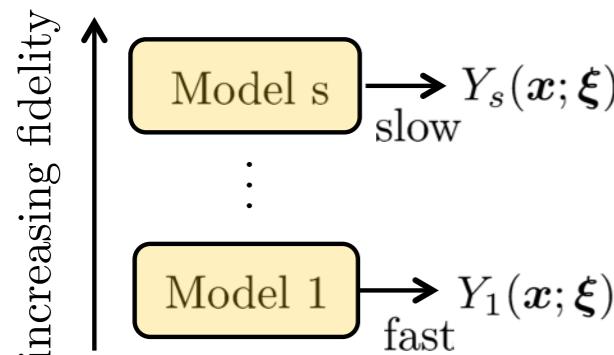
Number of runs is limited by time
and computational resources

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

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



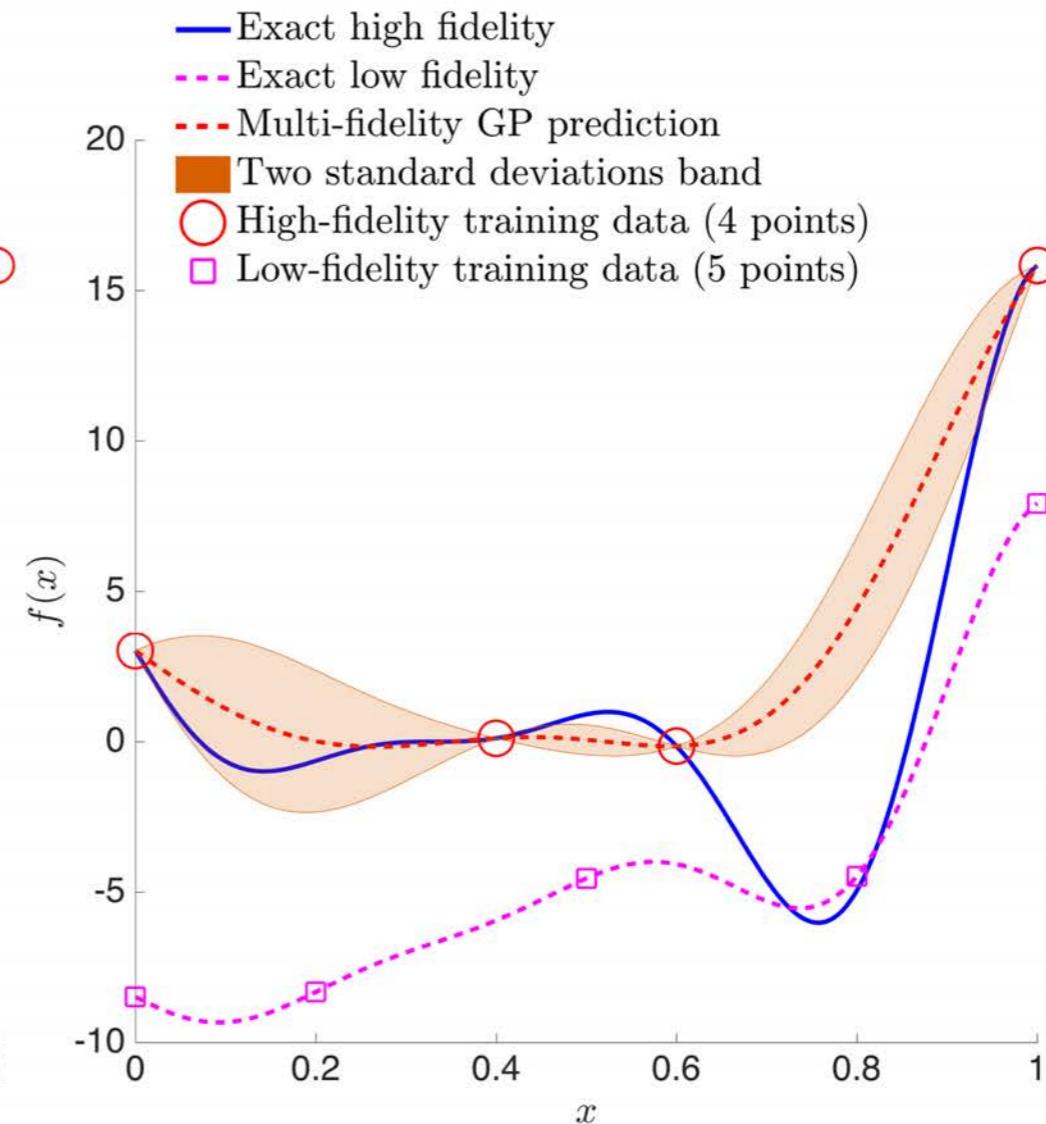
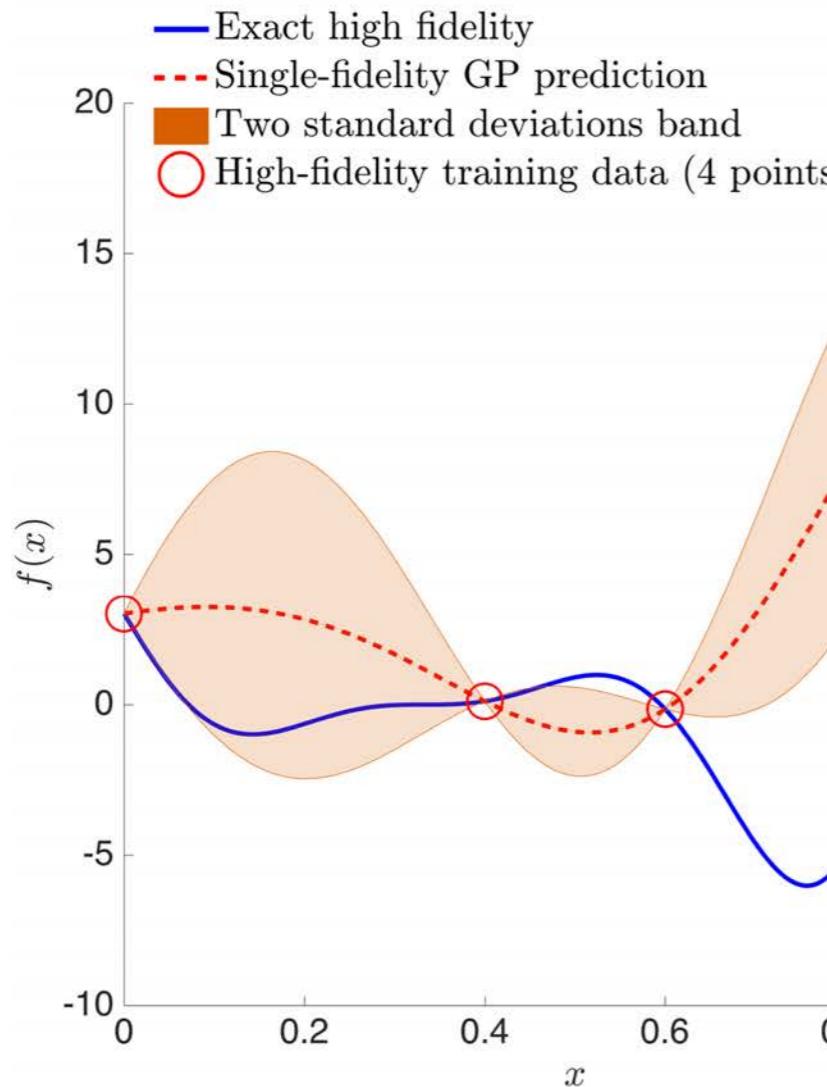
Multi-fidelity modeling



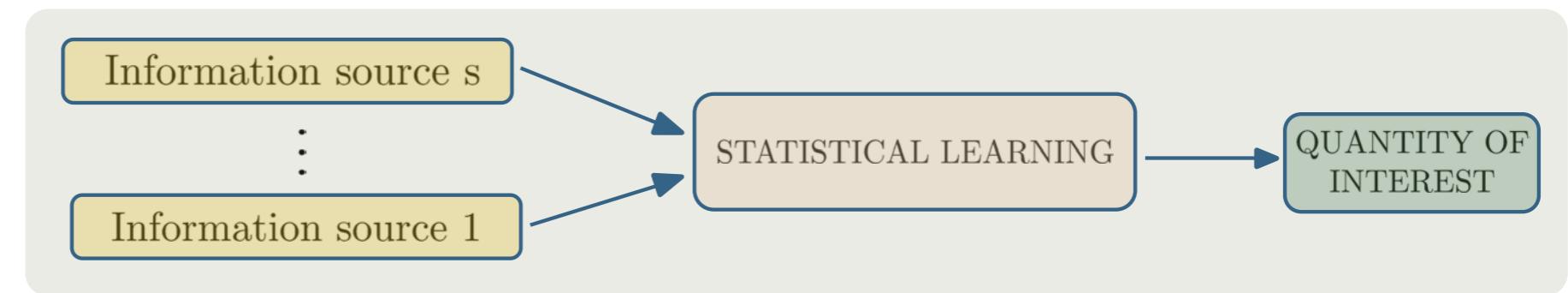
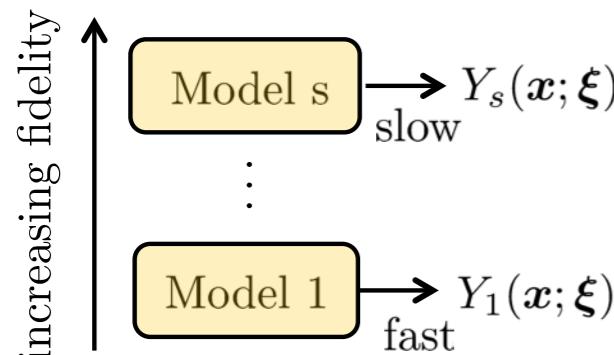
Number of runs is limited by time
and computational resources

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

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



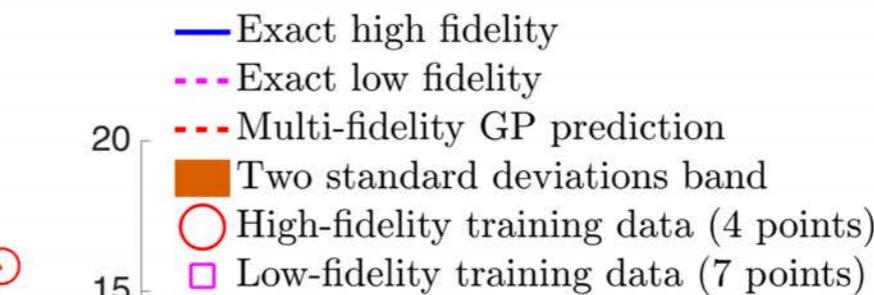
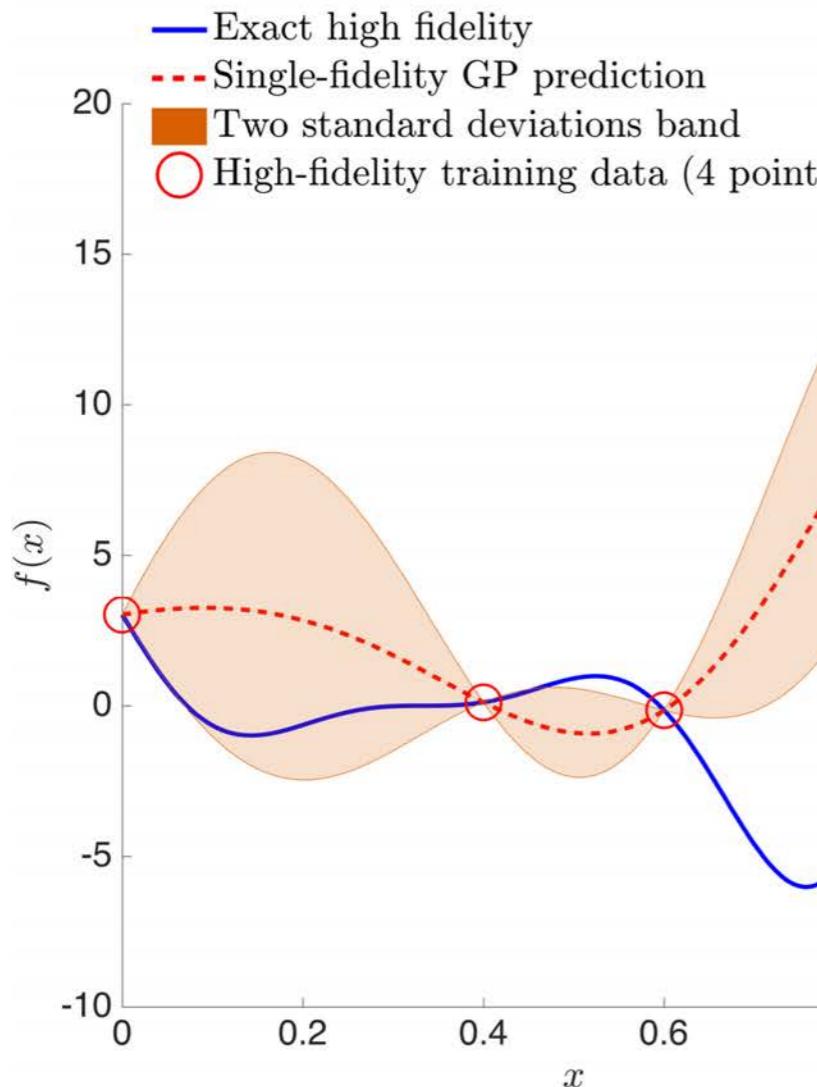
Multi-fidelity modeling



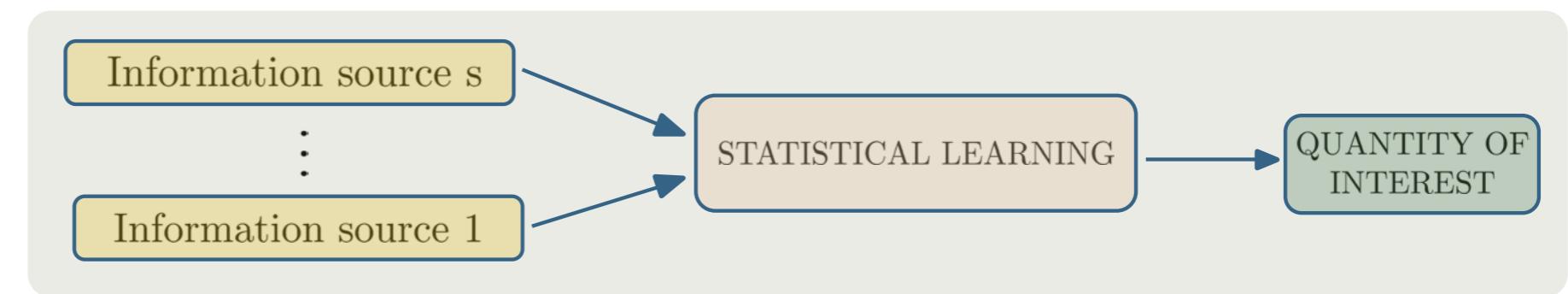
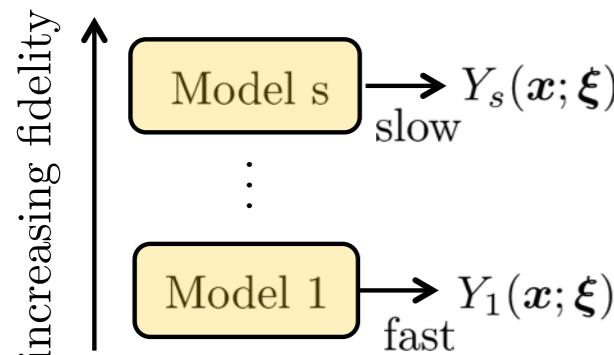
Number of runs is limited by time
and computational resources

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

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



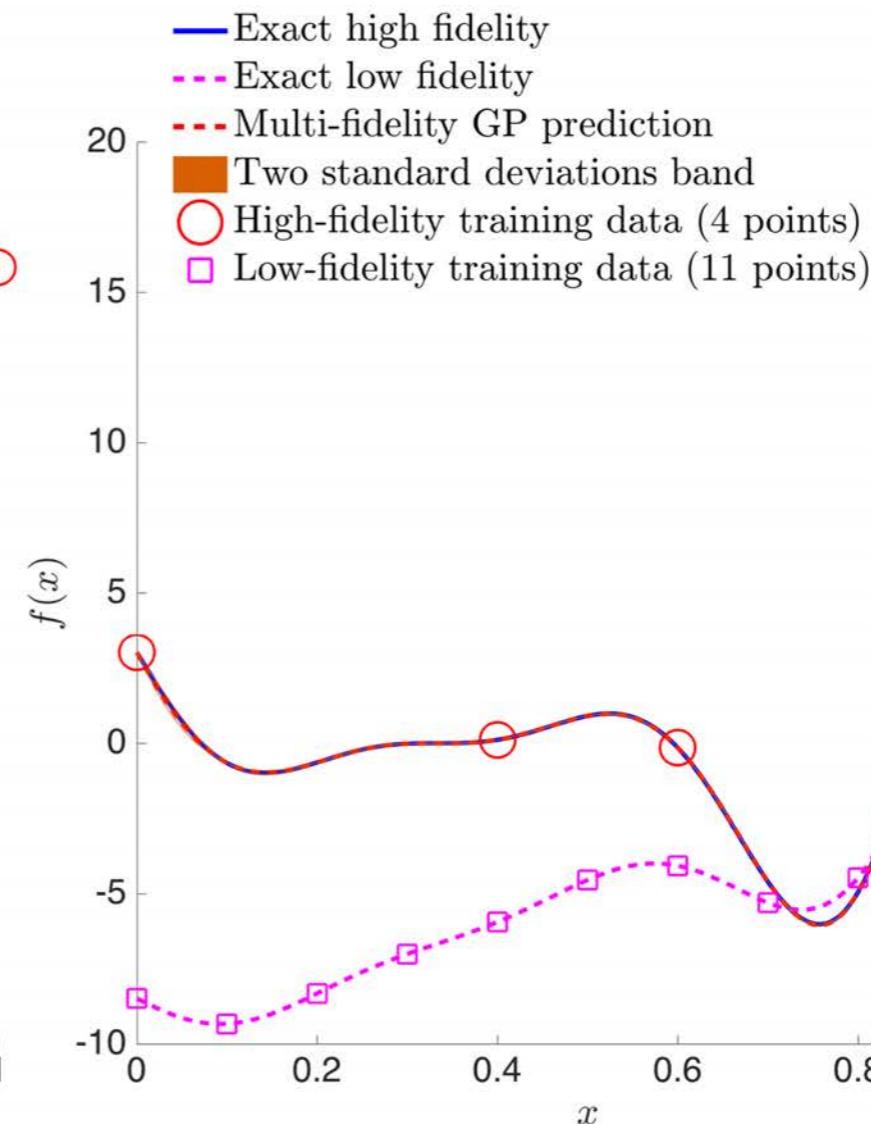
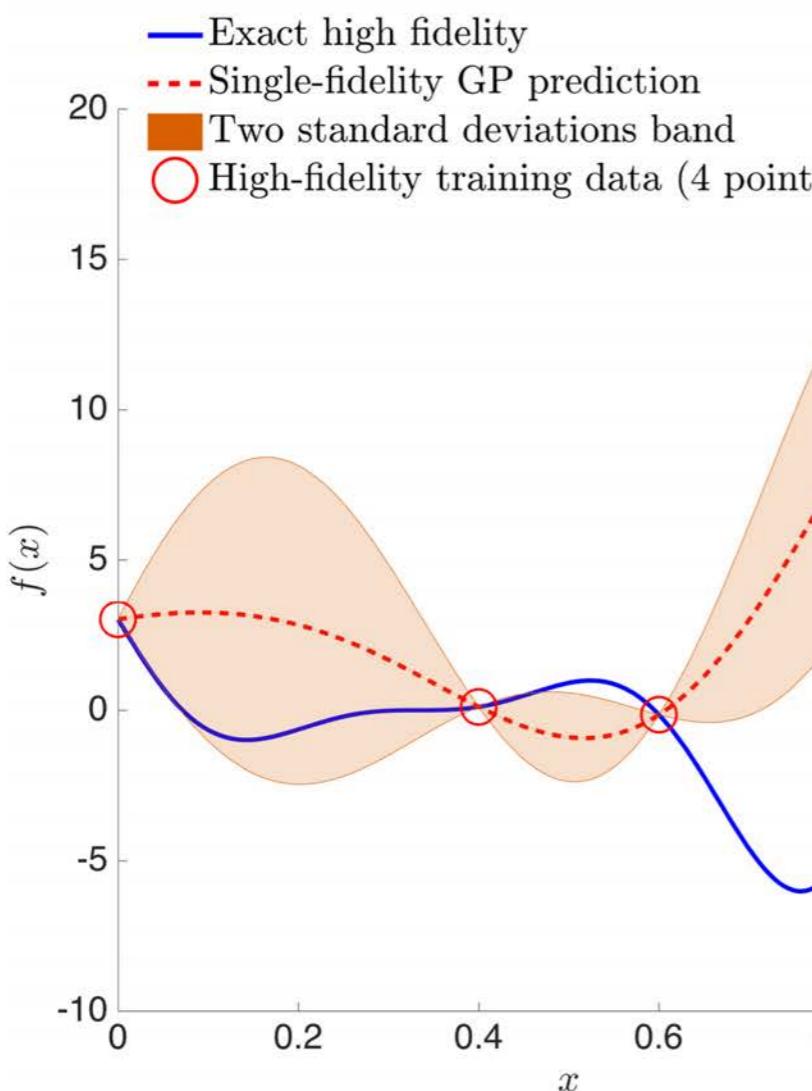
Multi-fidelity modeling



Number of runs is limited by time
and computational resources

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

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



Multi-fidelity modeling with GPs



Auto-regressive model:

AR1

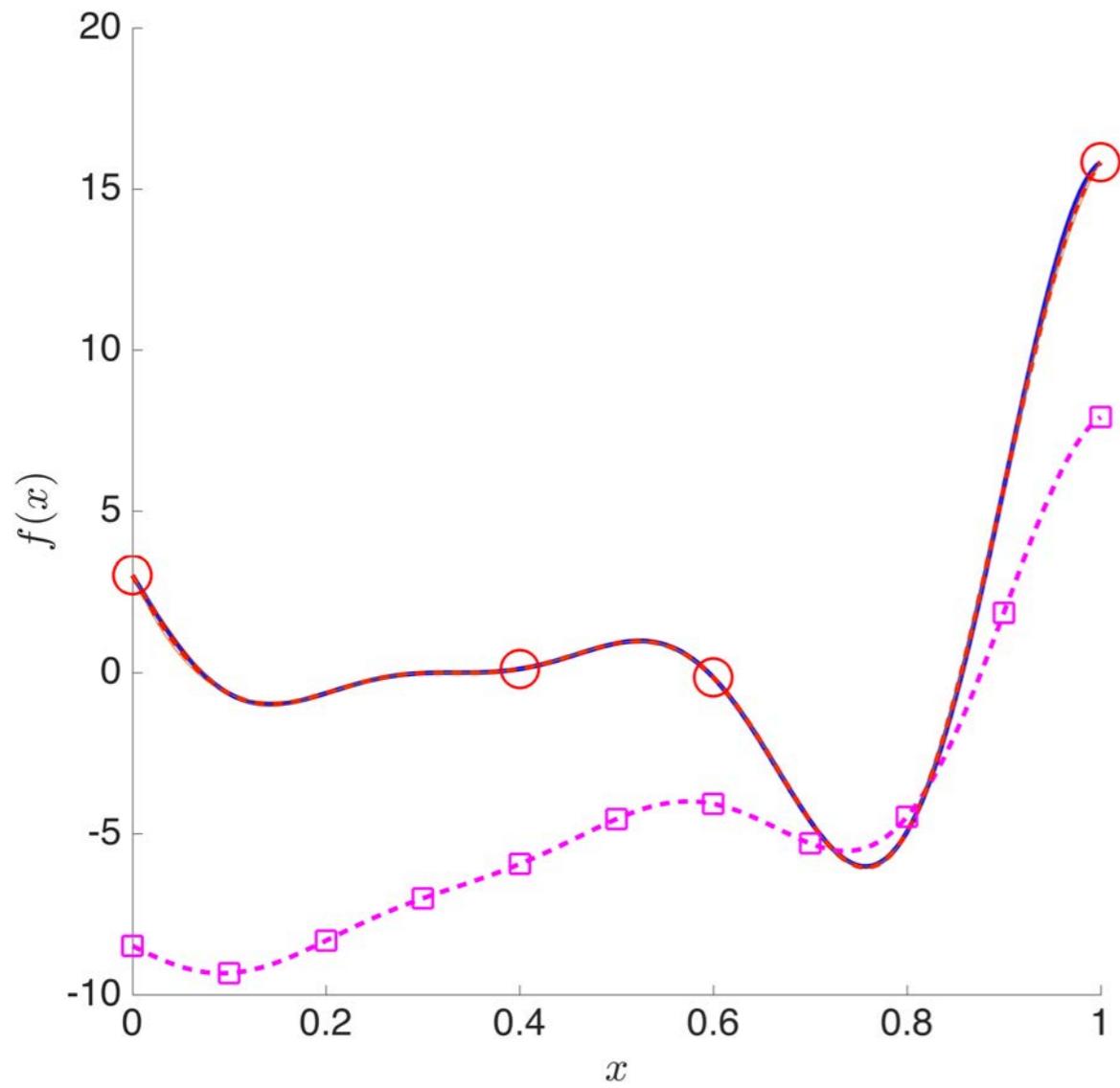
$$f_t(\mathbf{x}) = \rho_{t-1}(\mathbf{x}) f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x})$$

$$t = 1, \dots, s$$

Predicting the Output from a Complex Computer Code When Fast Approximations Are Available

M. C. Kennedy; A. O'Hagan

Biometrika, Vol. 87, No. 1. (Mar., 2000), pp. 1-13.

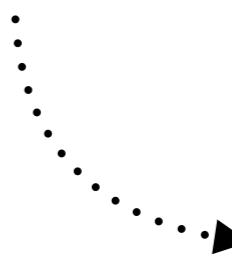


Predictive posterior

$$p(f_* | \mathbf{y}, \mathbf{X}, \mathbf{x}_*) = \mathcal{N}(f_* | \mu_*, \sigma_*^2),$$

$$\mu_*(\mathbf{x}_*) = \mathbf{k}_{*N} (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y},$$

$$\sigma_*^2(\mathbf{x}_*) = \mathbf{k}_{**} - \mathbf{k}_{*N} (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{k}_{N*},$$



$$\begin{matrix} N_1 & \left[\begin{matrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{matrix} \right] \\ N_2 & \end{matrix}$$

Block covariance matrix

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: $g(\mathbf{x})$ is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate $g(\mathbf{x})$ using a GP surrogate: $y = f(\mathbf{x}) + \epsilon$, $f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta))$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

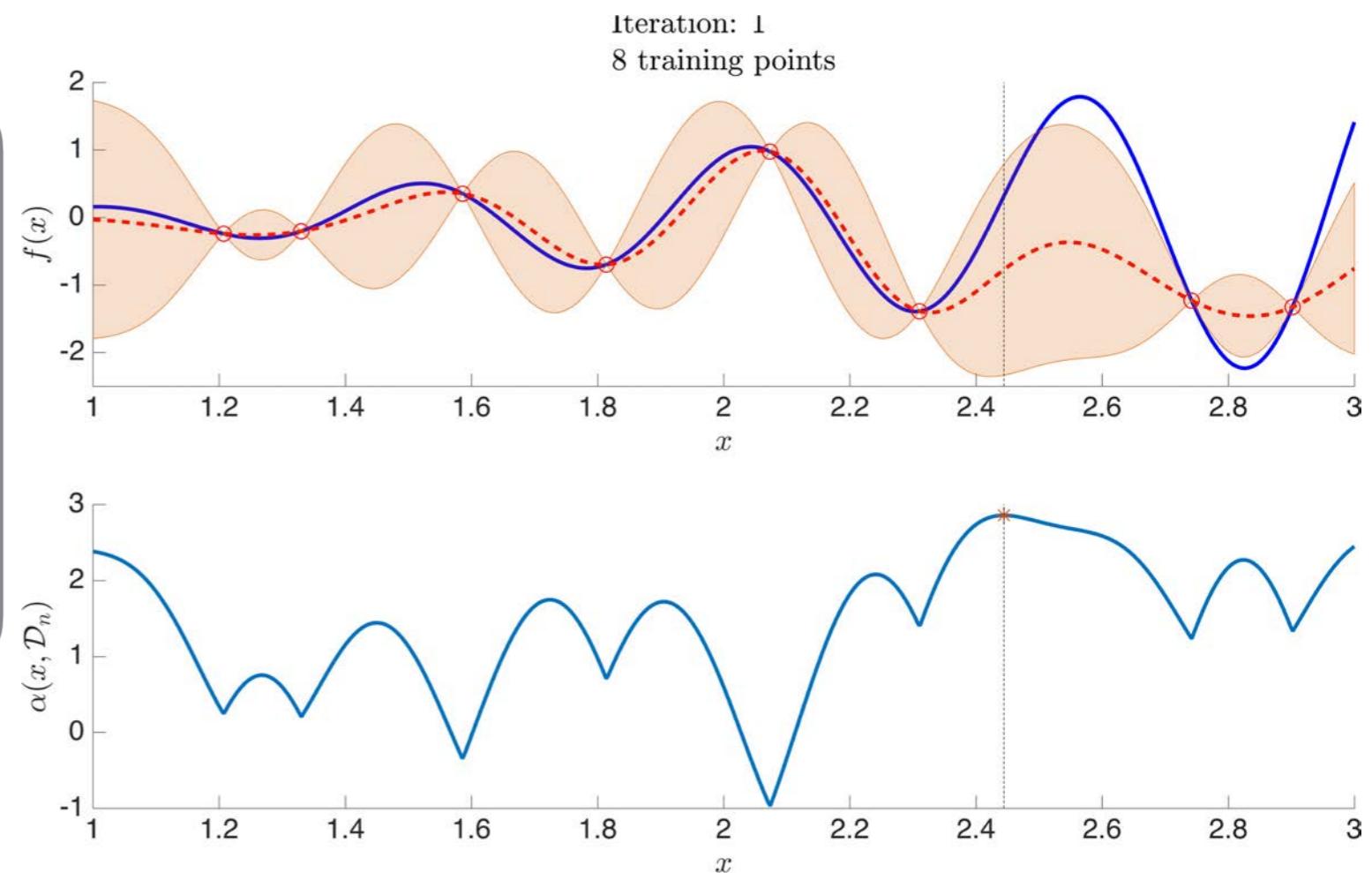
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v \mid \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that maximize the expected improvement

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: $g(\mathbf{x})$ is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate $g(\mathbf{x})$ using a GP surrogate: $y = f(\mathbf{x}) + \epsilon$, $f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta))$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

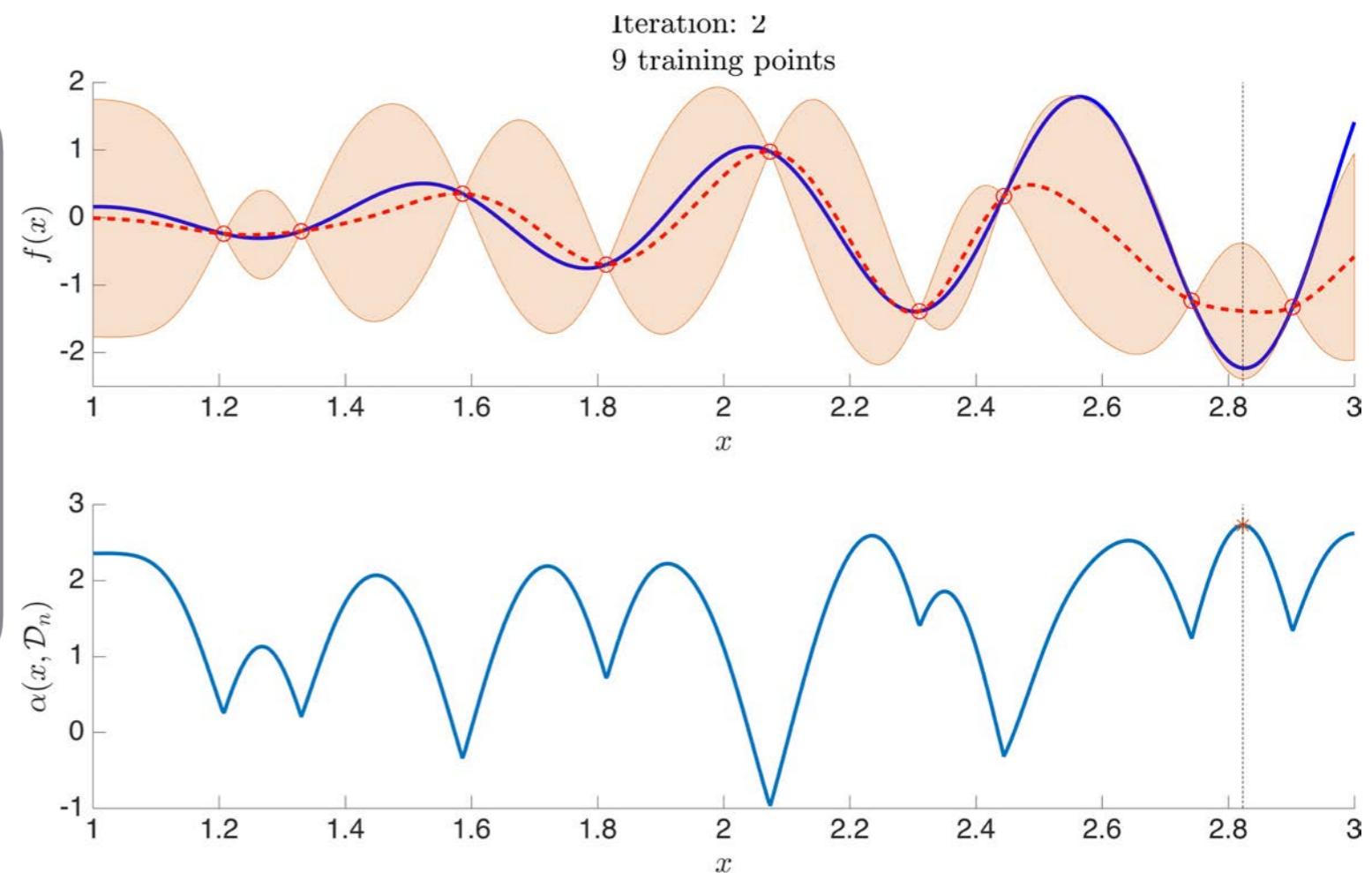
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v \mid \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that maximize the expected improvement

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: $g(\mathbf{x})$ is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate $g(\mathbf{x})$ using a GP surrogate: $y = f(\mathbf{x}) + \epsilon$, $f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta))$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

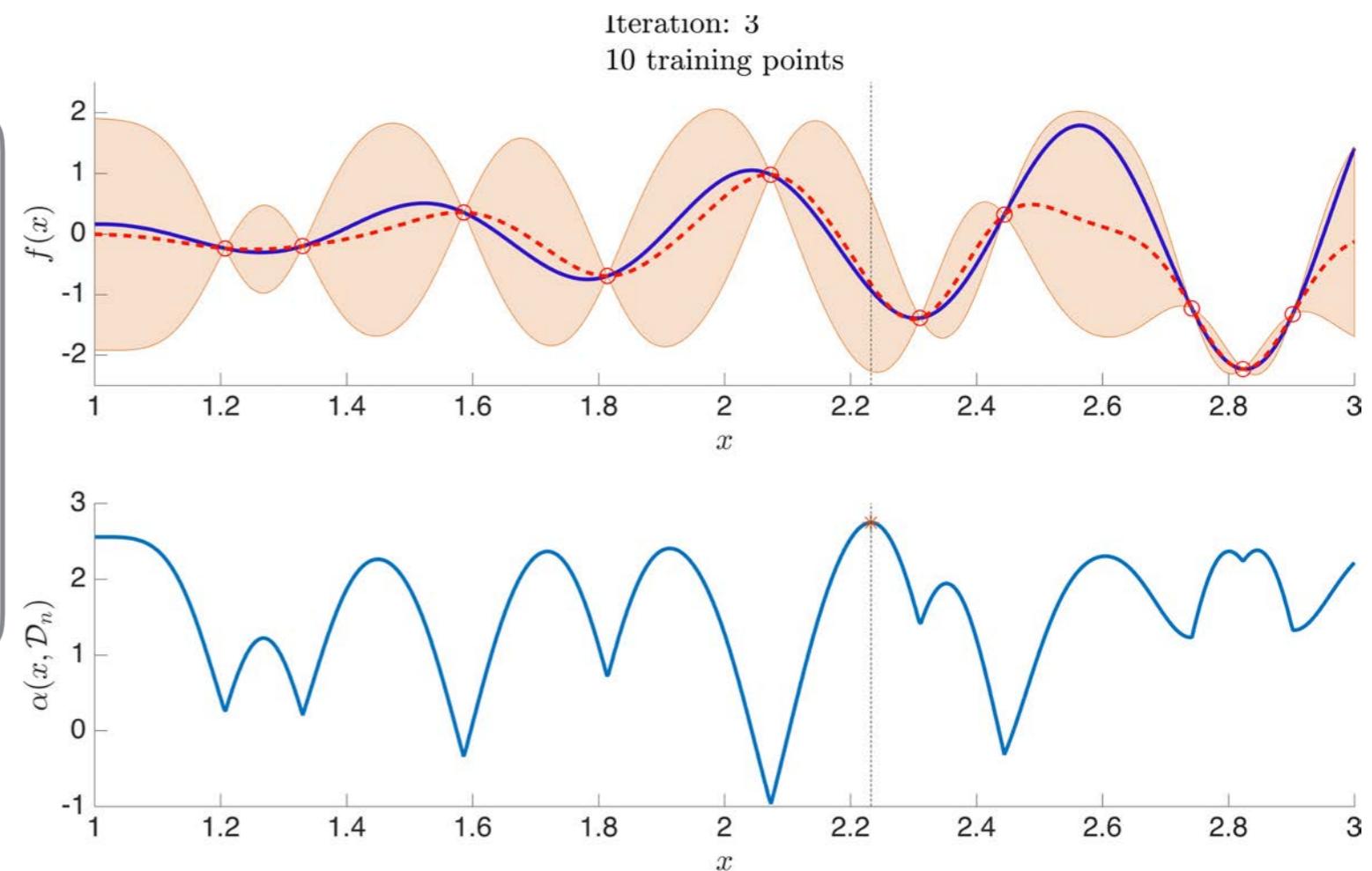
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v \mid \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that maximize the expected improvement

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: $g(\mathbf{x})$ is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate $g(\mathbf{x})$ using a GP surrogate: $y = f(\mathbf{x}) + \epsilon$, $f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta))$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

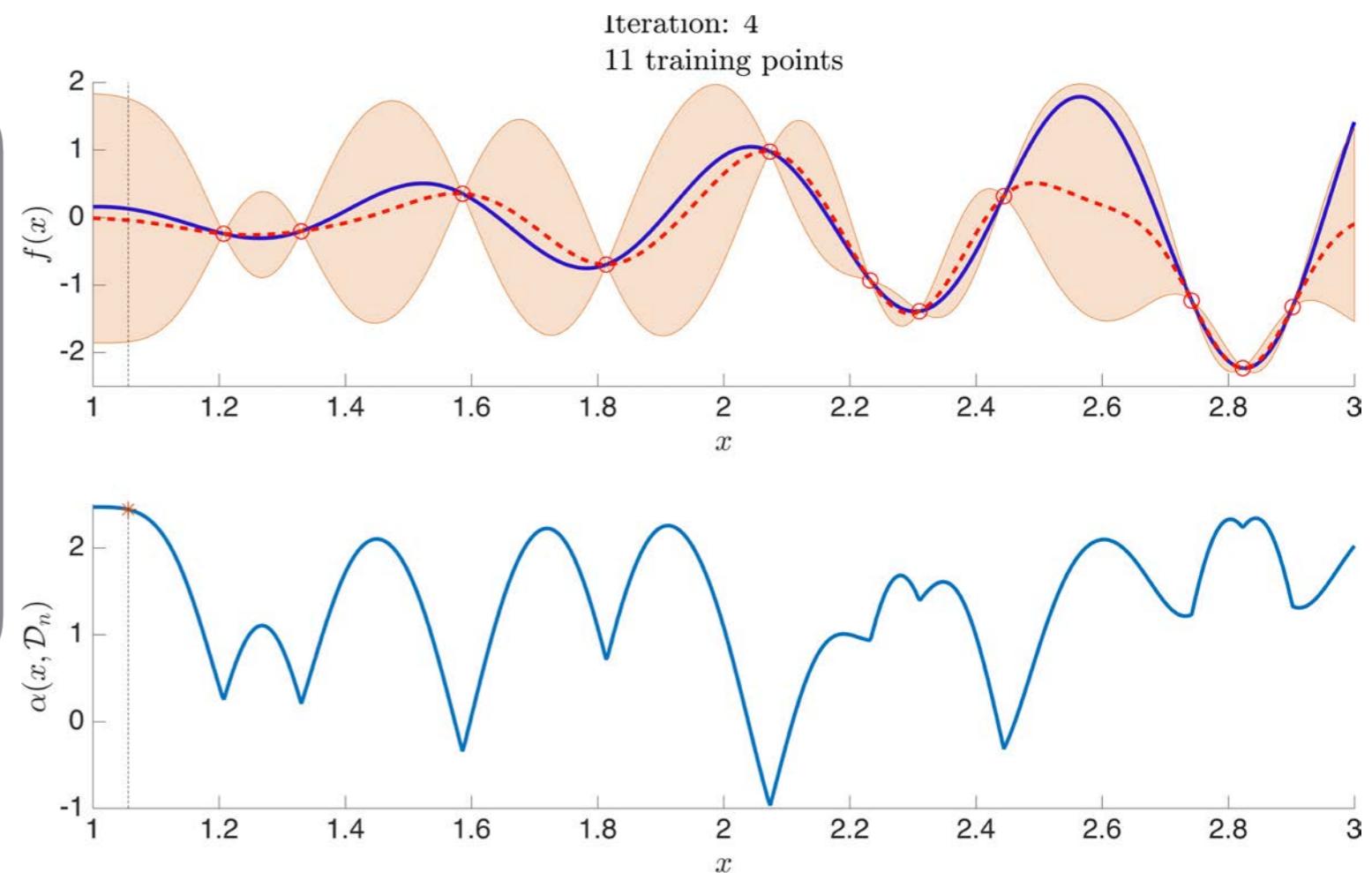
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v \mid \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that maximize the expected improvement

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: $g(\mathbf{x})$ is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate $g(\mathbf{x})$ using a GP surrogate: $y = f(\mathbf{x}) + \epsilon$, $f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta))$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

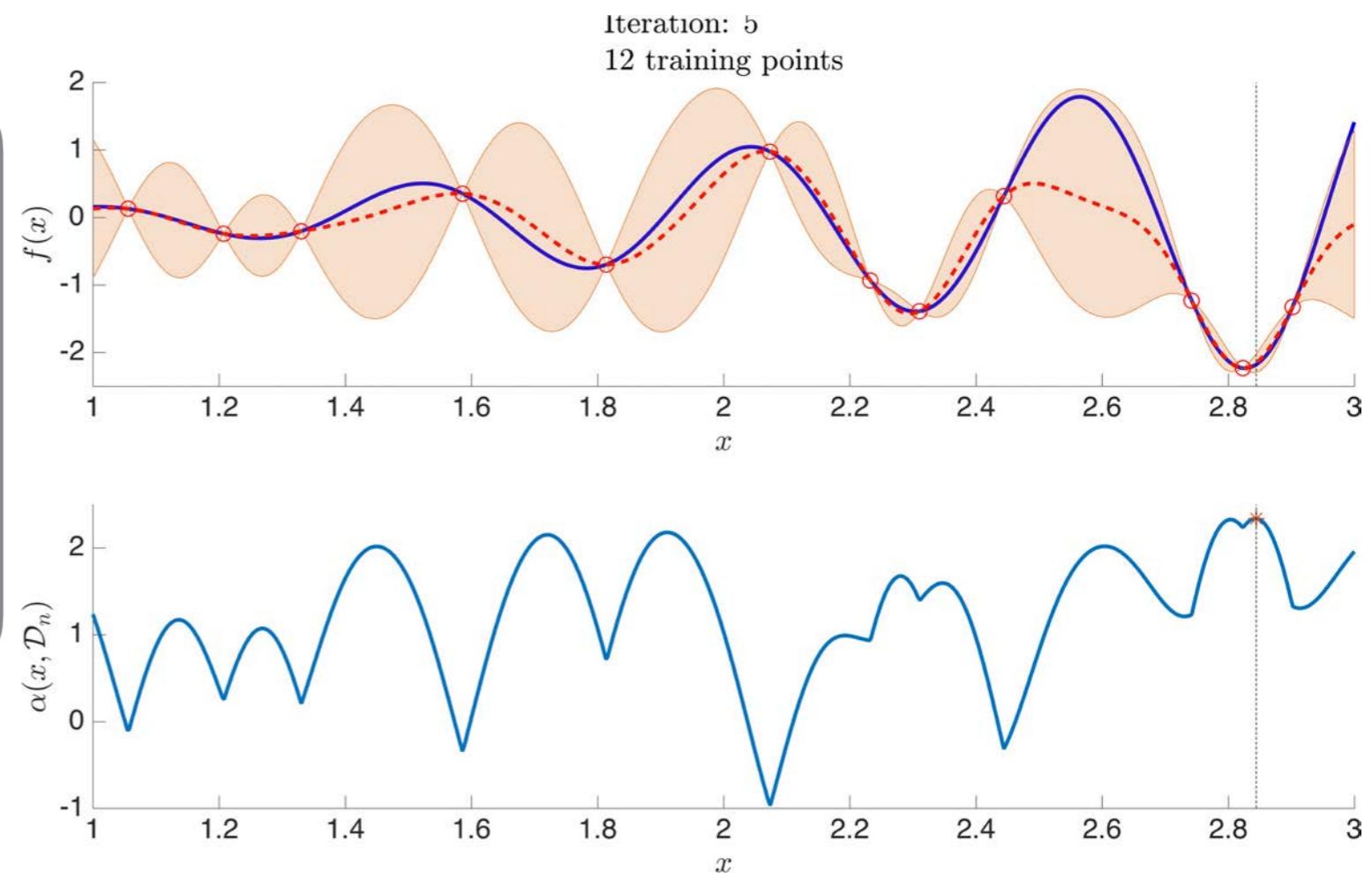
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v \mid \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that maximize the expected improvement

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: $g(\mathbf{x})$ is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate $g(\mathbf{x})$ using a GP surrogate: $y = f(\mathbf{x}) + \epsilon$, $f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta))$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

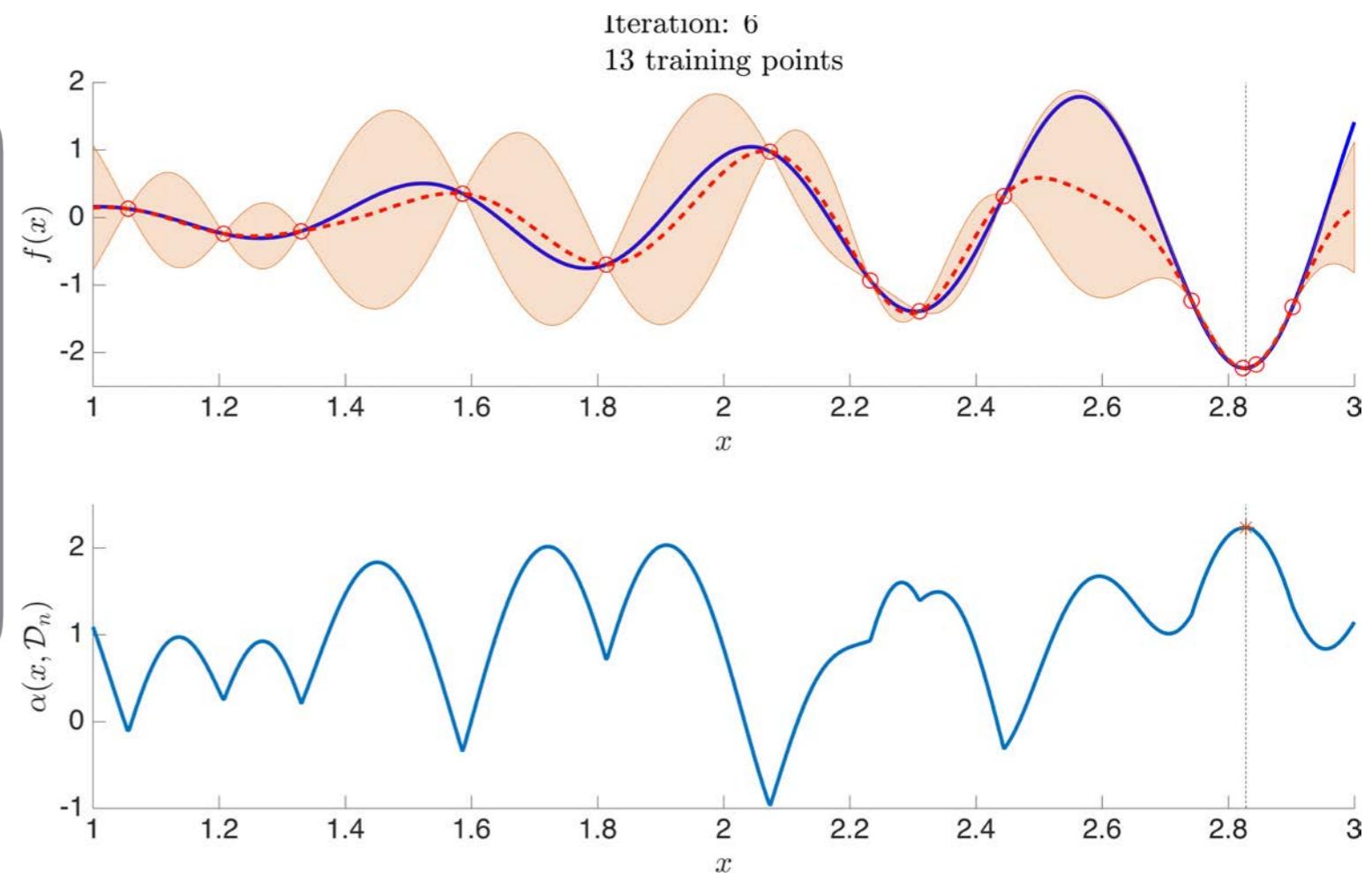
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v \mid \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that maximize the expected improvement

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: $g(\mathbf{x})$ is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate $g(\mathbf{x})$ using a GP surrogate: $y = f(\mathbf{x}) + \epsilon$, $f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta))$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

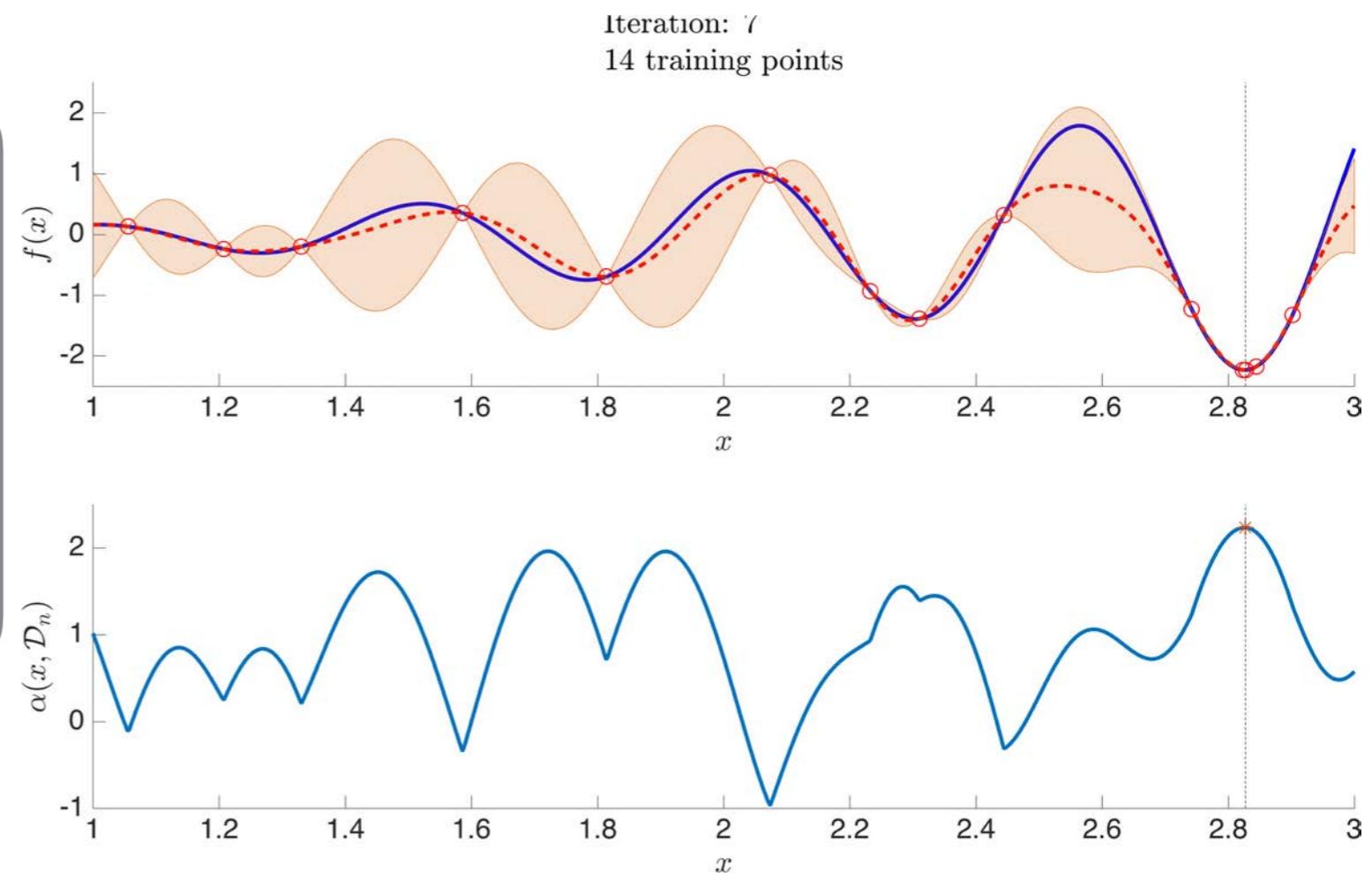
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v \mid \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that maximize the expected improvement