ENM 5310: Data-driven Modeling and Probabilistic Scientific Computing

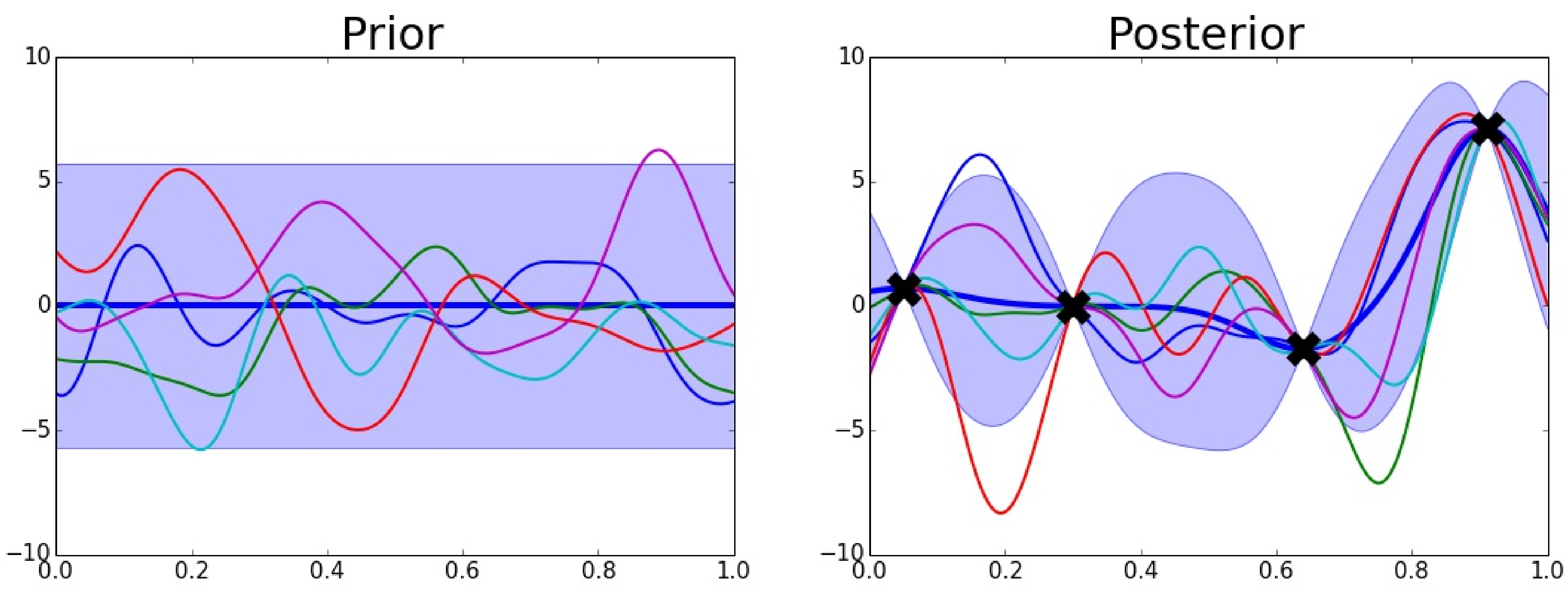
Lecture #20: Sequential decision making

Paris Perdikaris
April 13, 2023

Penn
University *of* Pennsylvania

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

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

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

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

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

Linear constraints

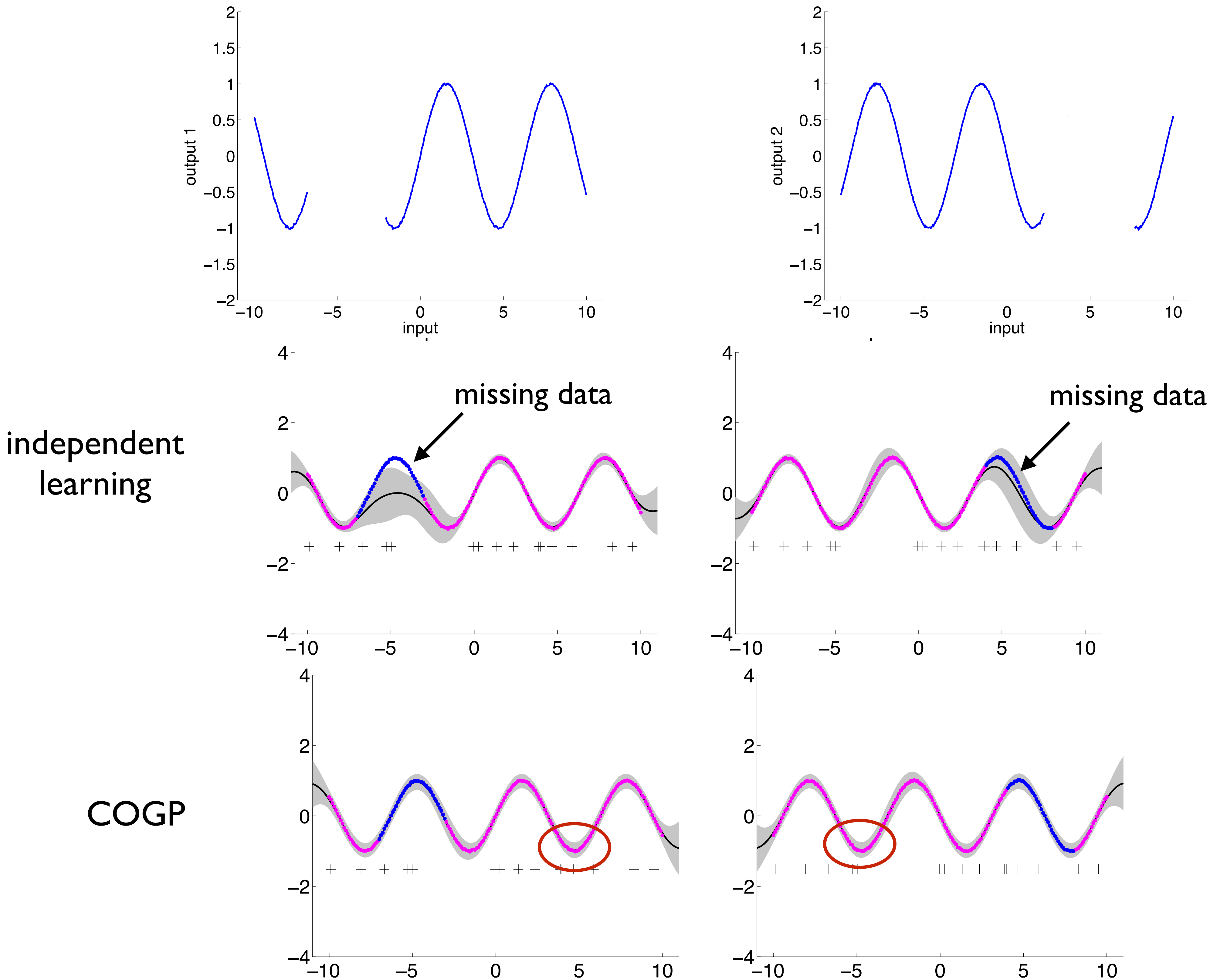
$$f \sim \mathcal{GP}\left(0, k(x, x'; \theta)\right)$$

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

Multi-output Gaussian process regression

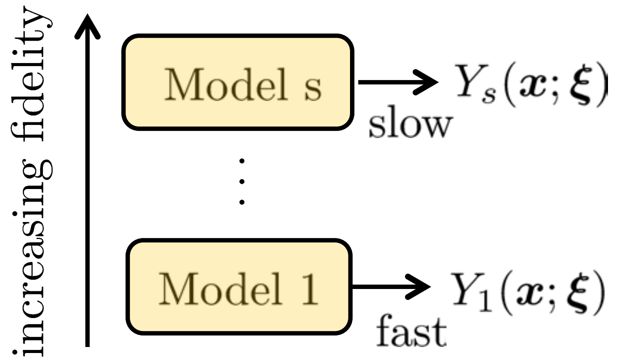
Learn two correlated tasks (outputs) with lots of data + missing data

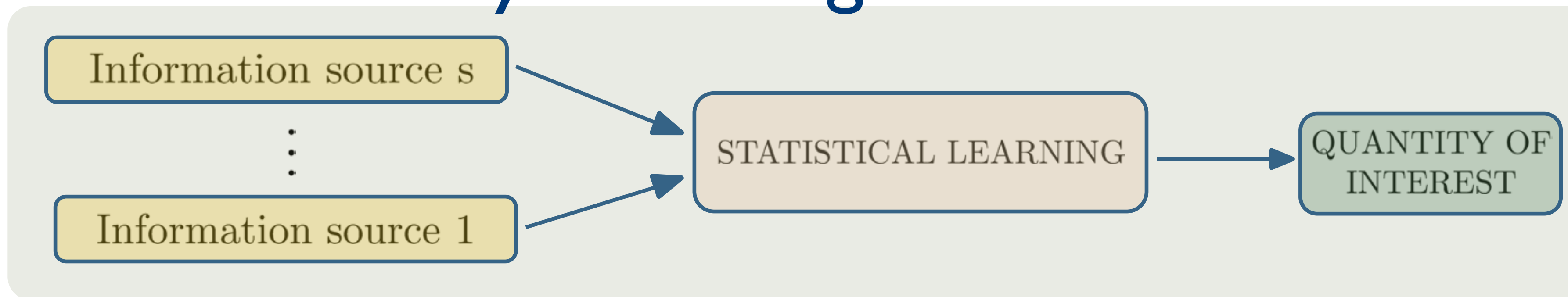
output 2 = - output 1



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

Multi-fidelity modeling

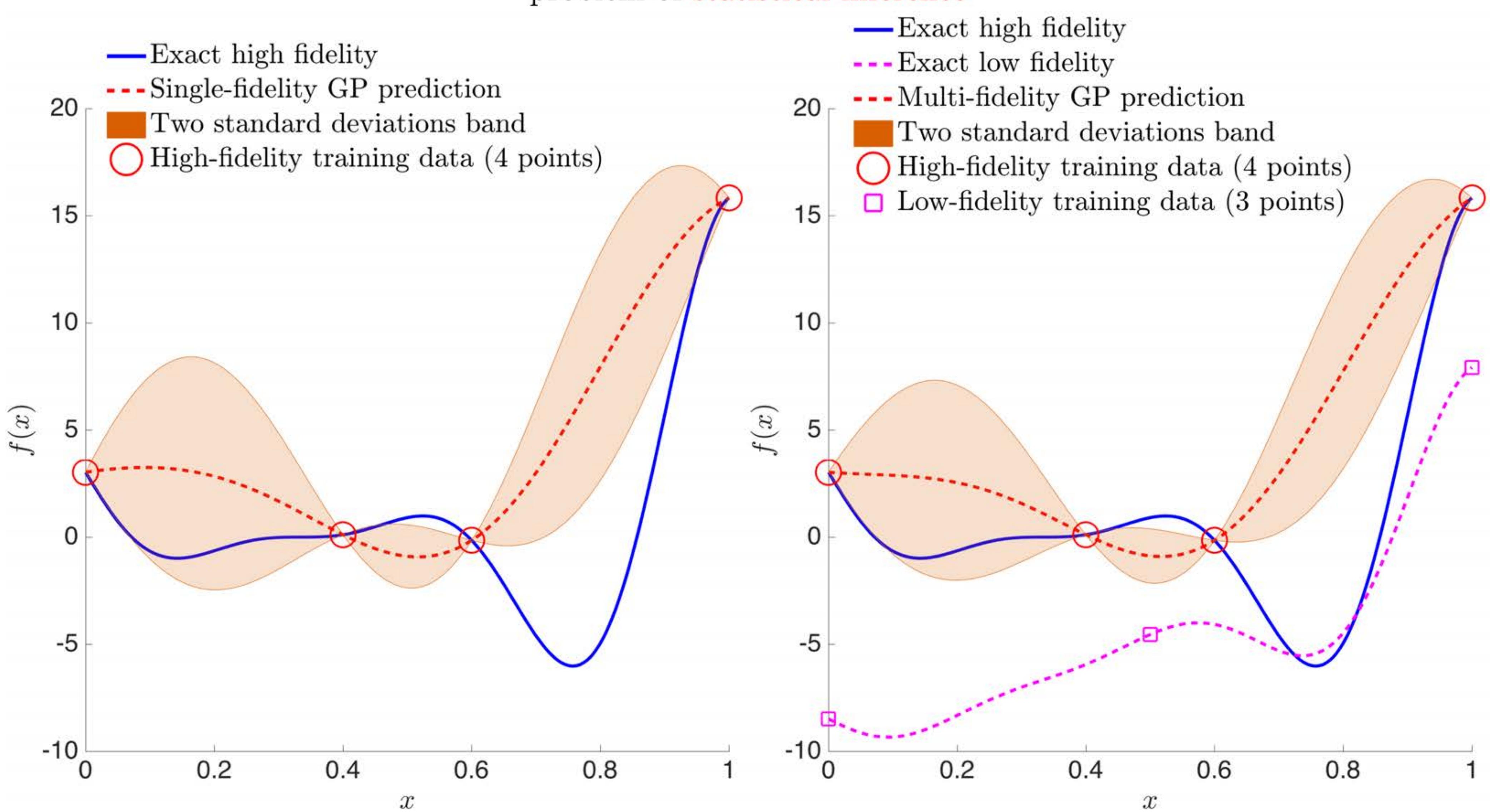




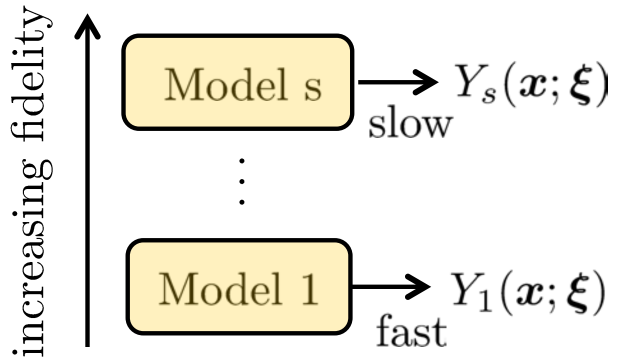
Number of runs is limited by time and computational resources

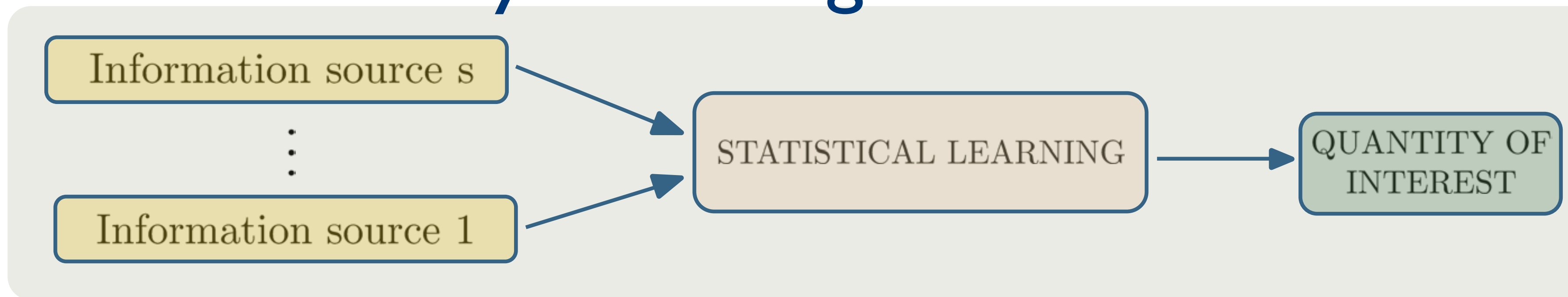
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

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



Multi-fidelity modeling

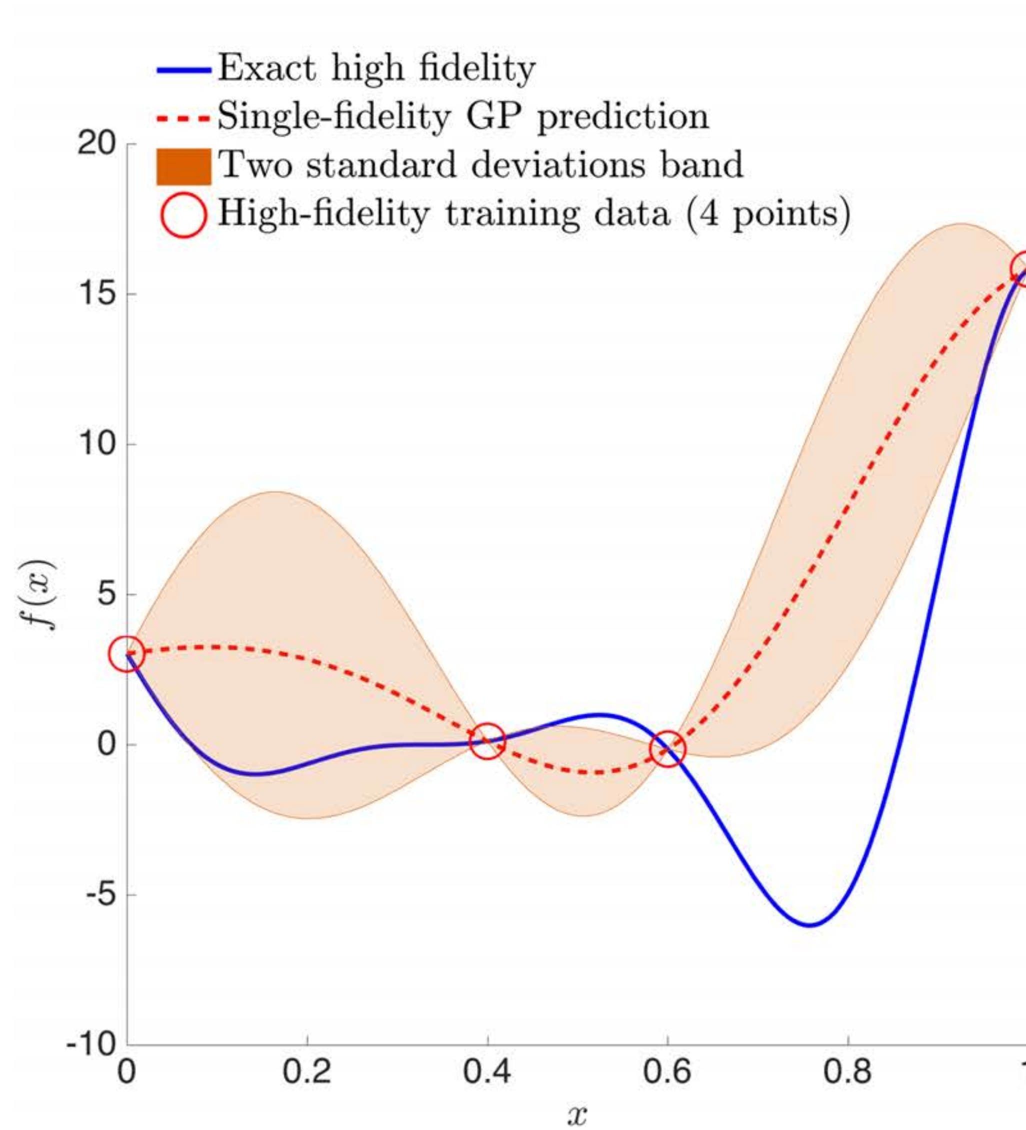


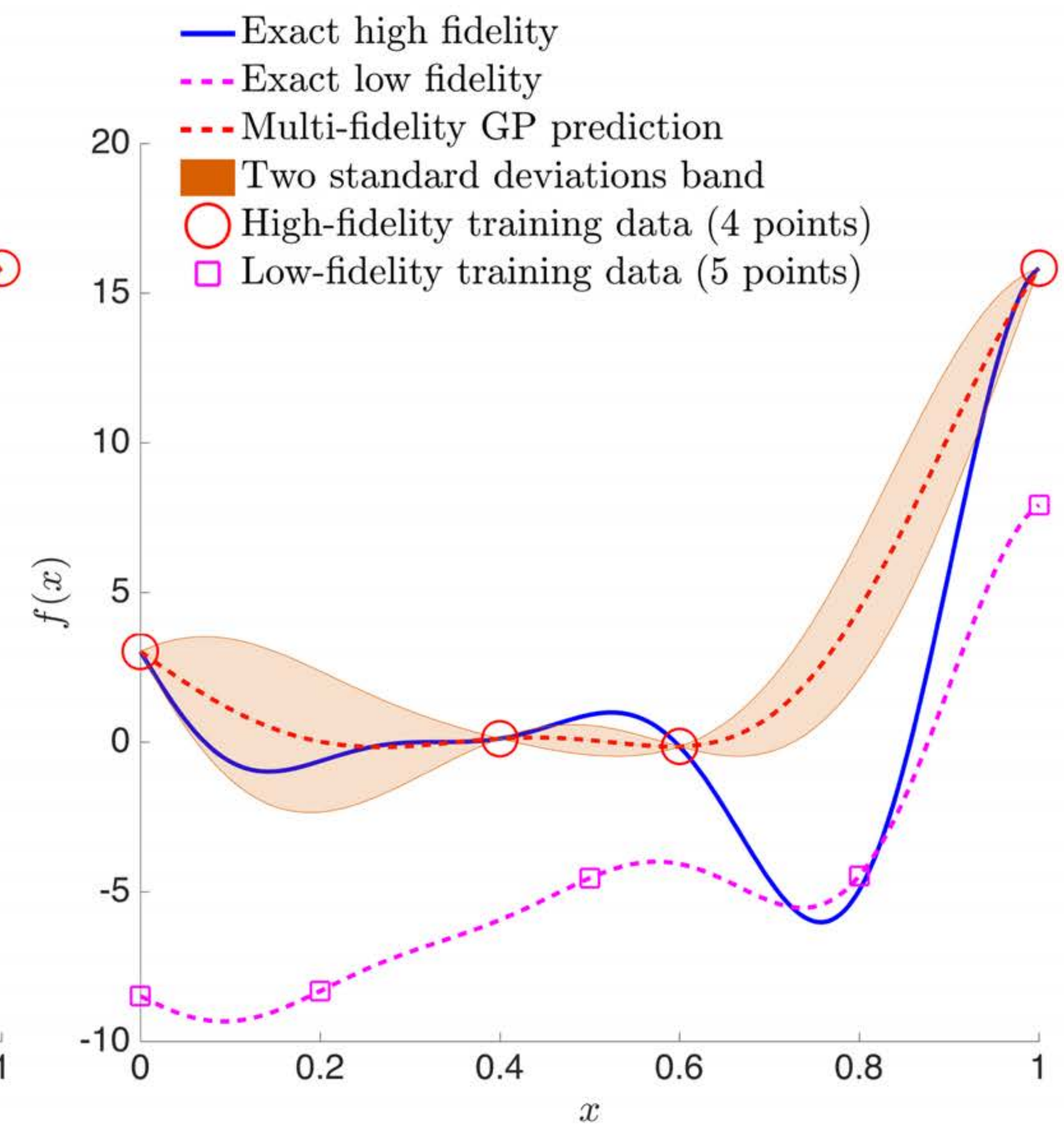


Number of runs is limited by time and computational resources

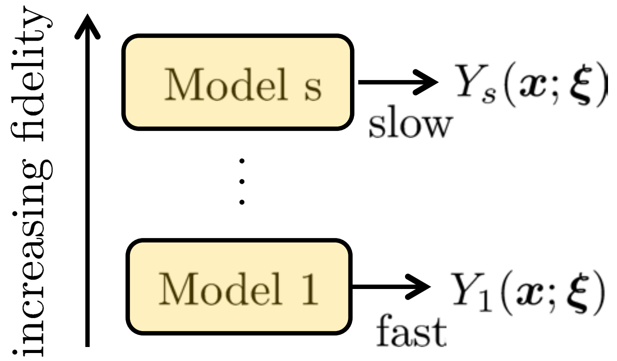
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

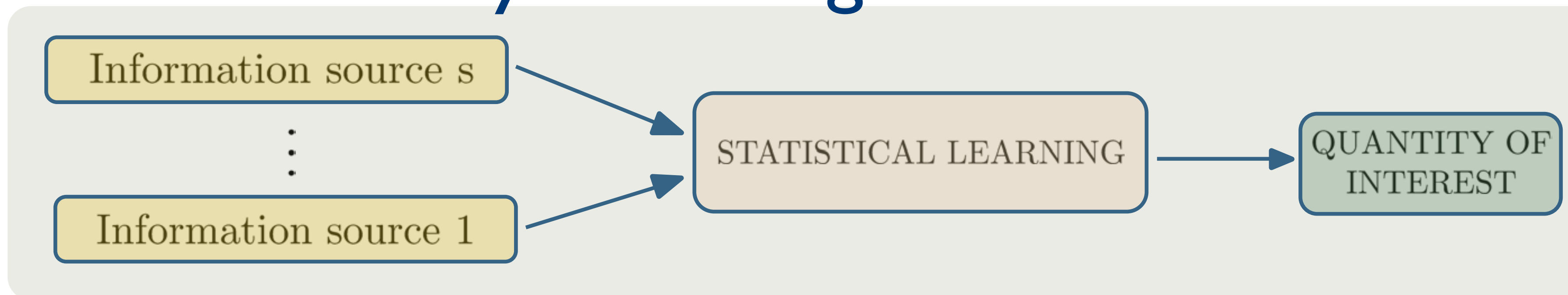
Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference





Multi-fidelity modeling

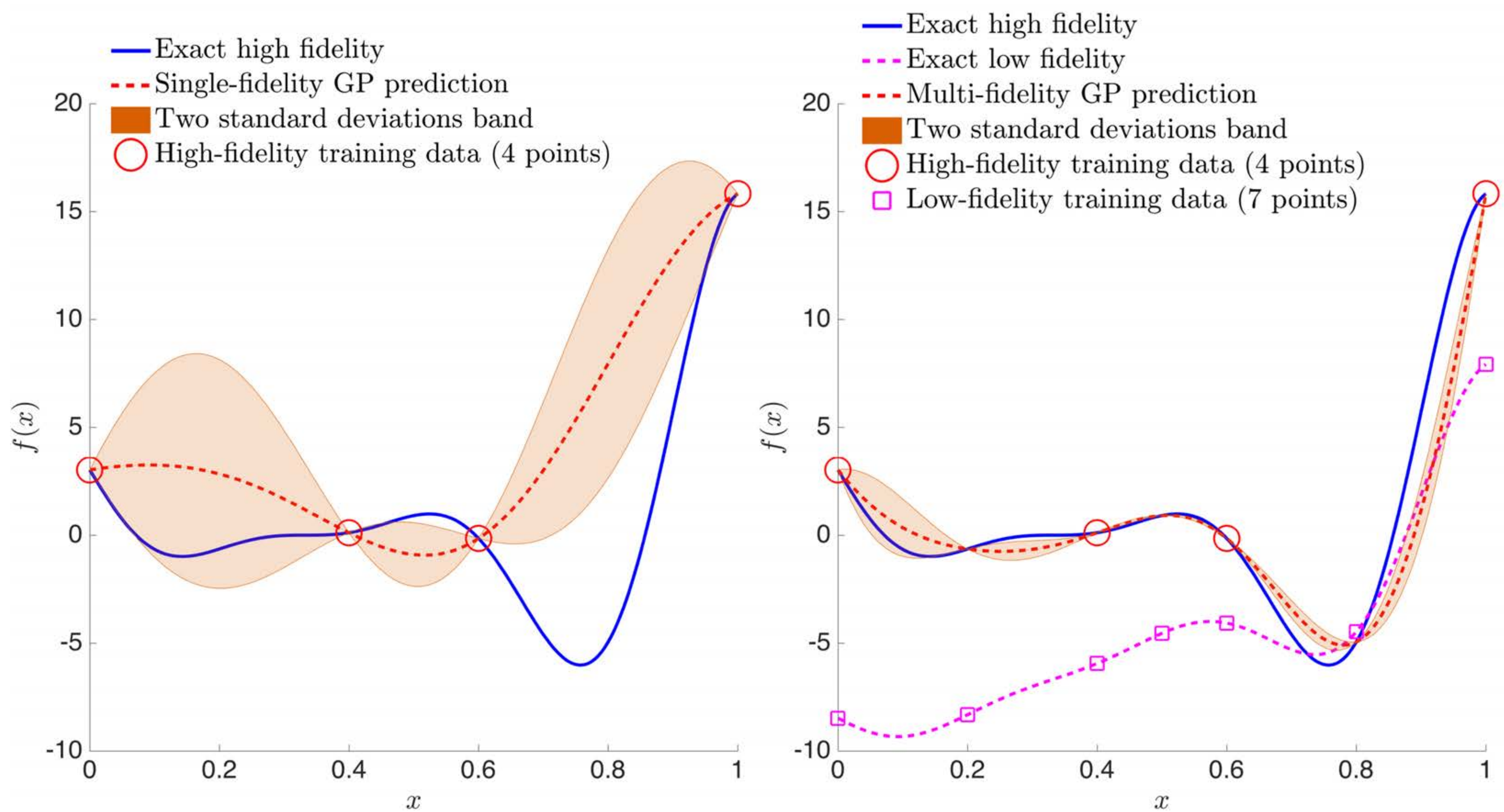




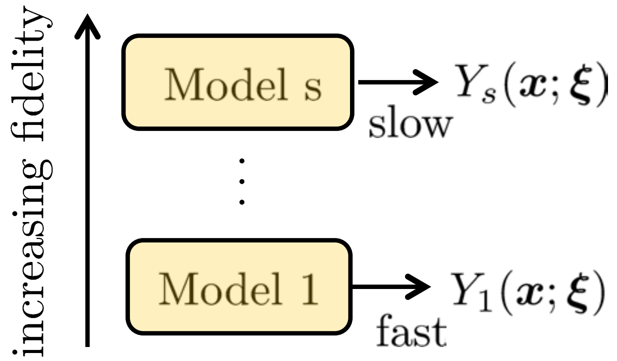
Number of runs is limited by time and computational resources

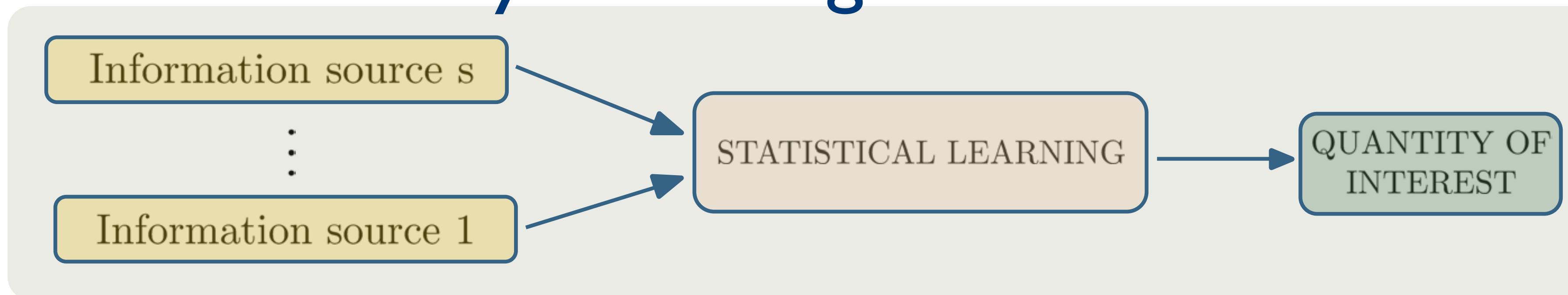
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

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



Multi-fidelity modeling

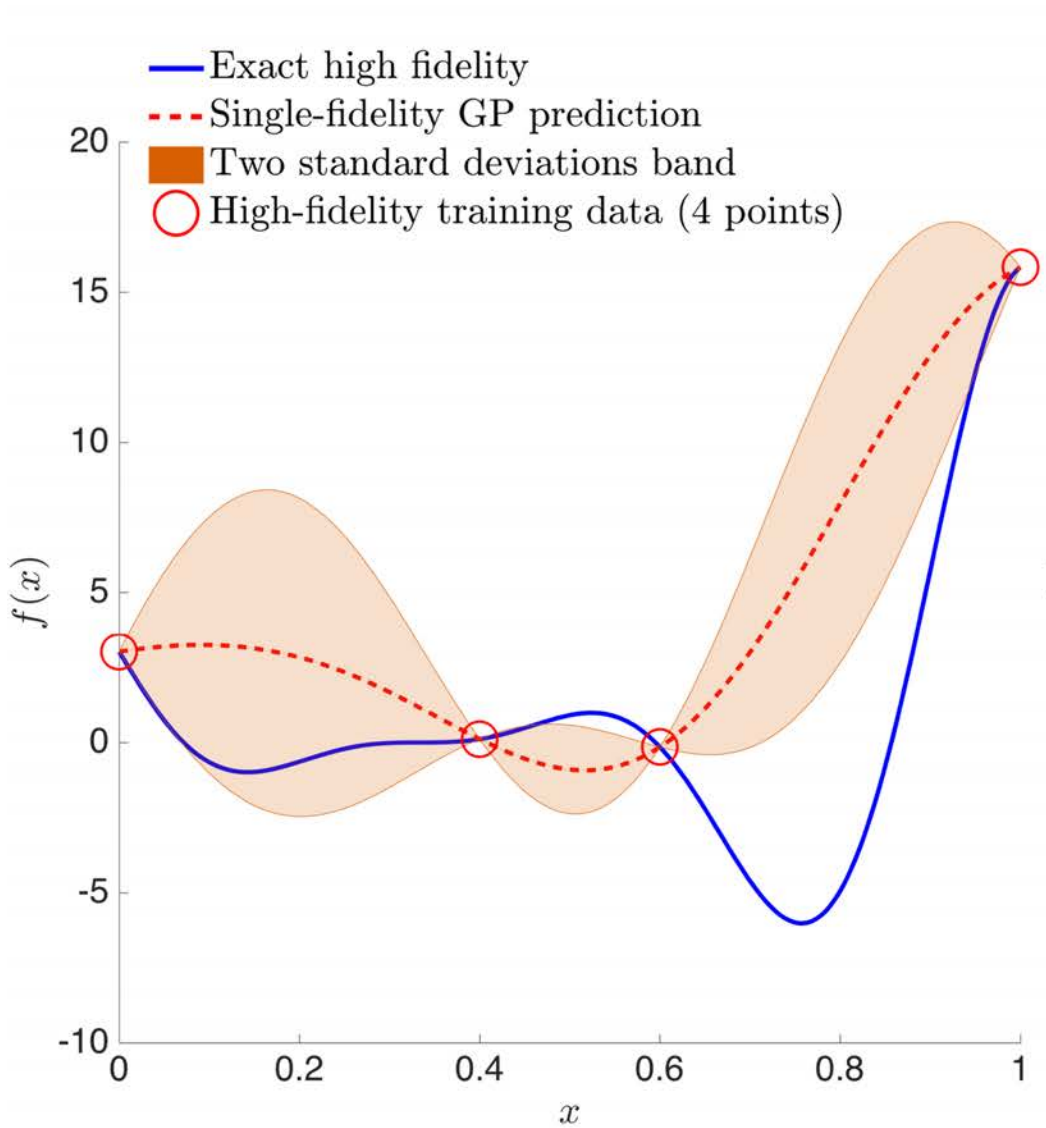


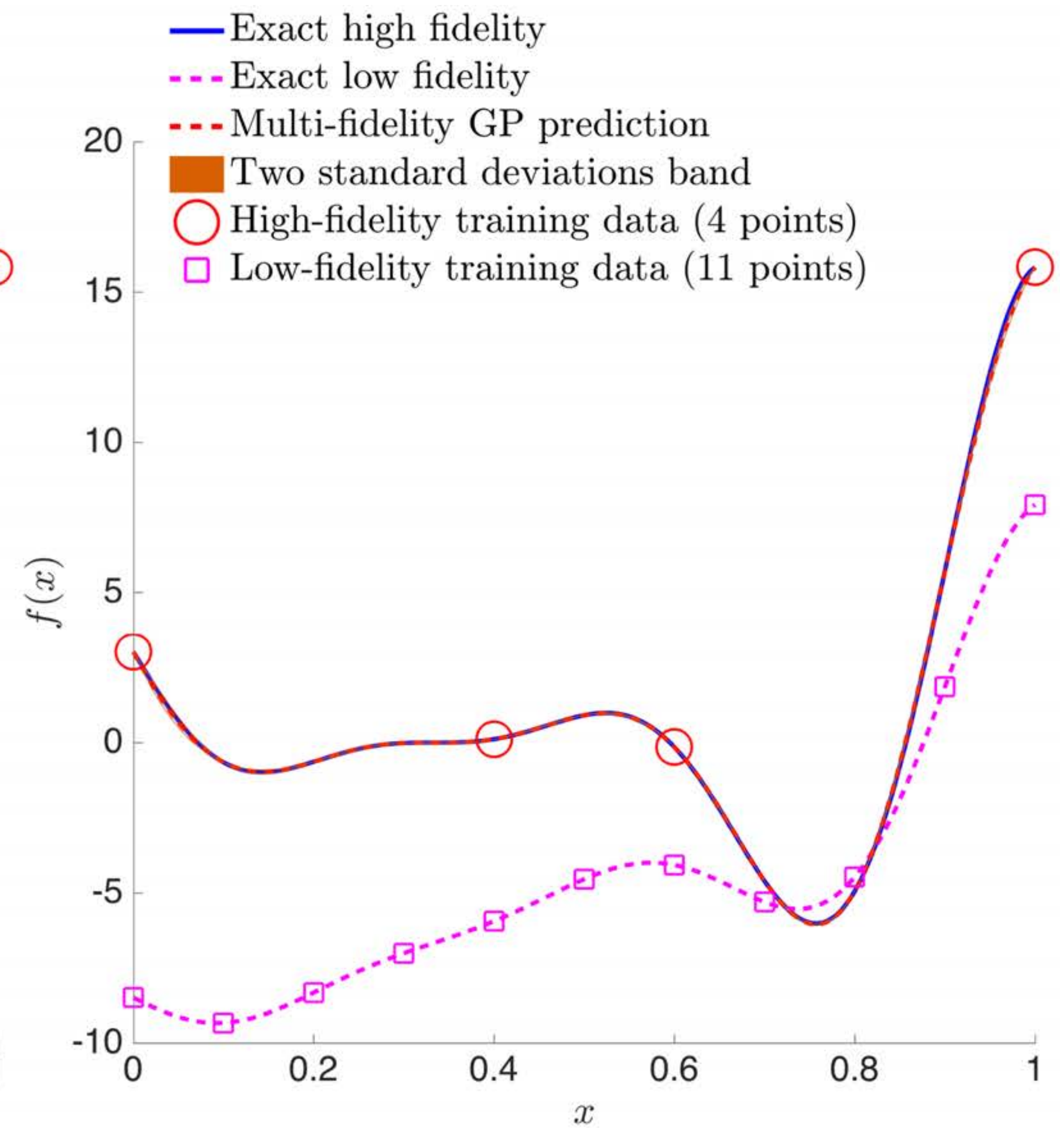


Number of runs is limited by time
and computational resources

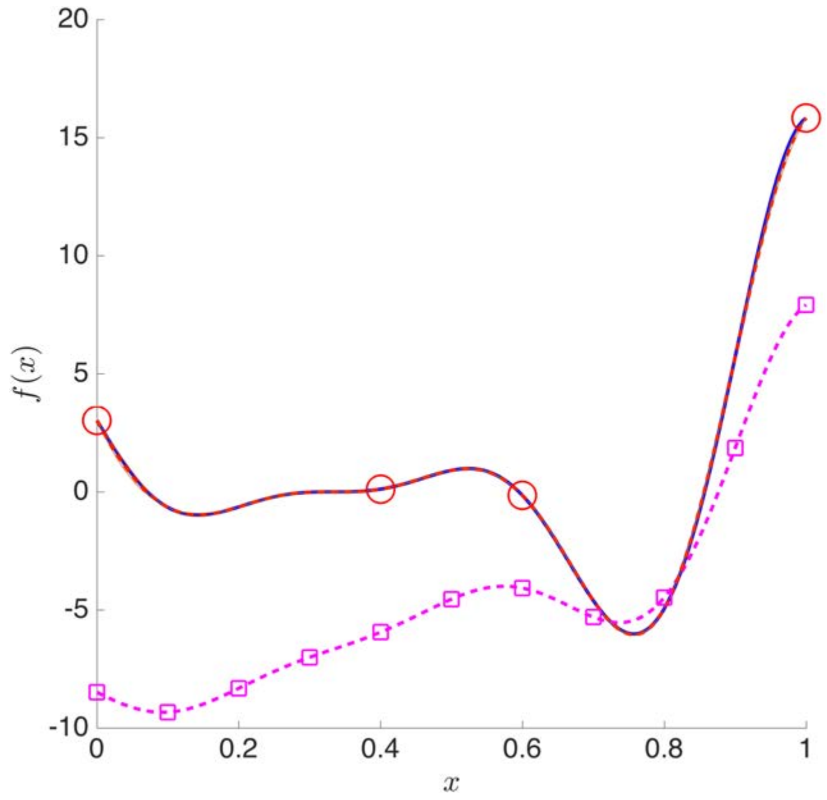
We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

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:

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

Training:

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

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

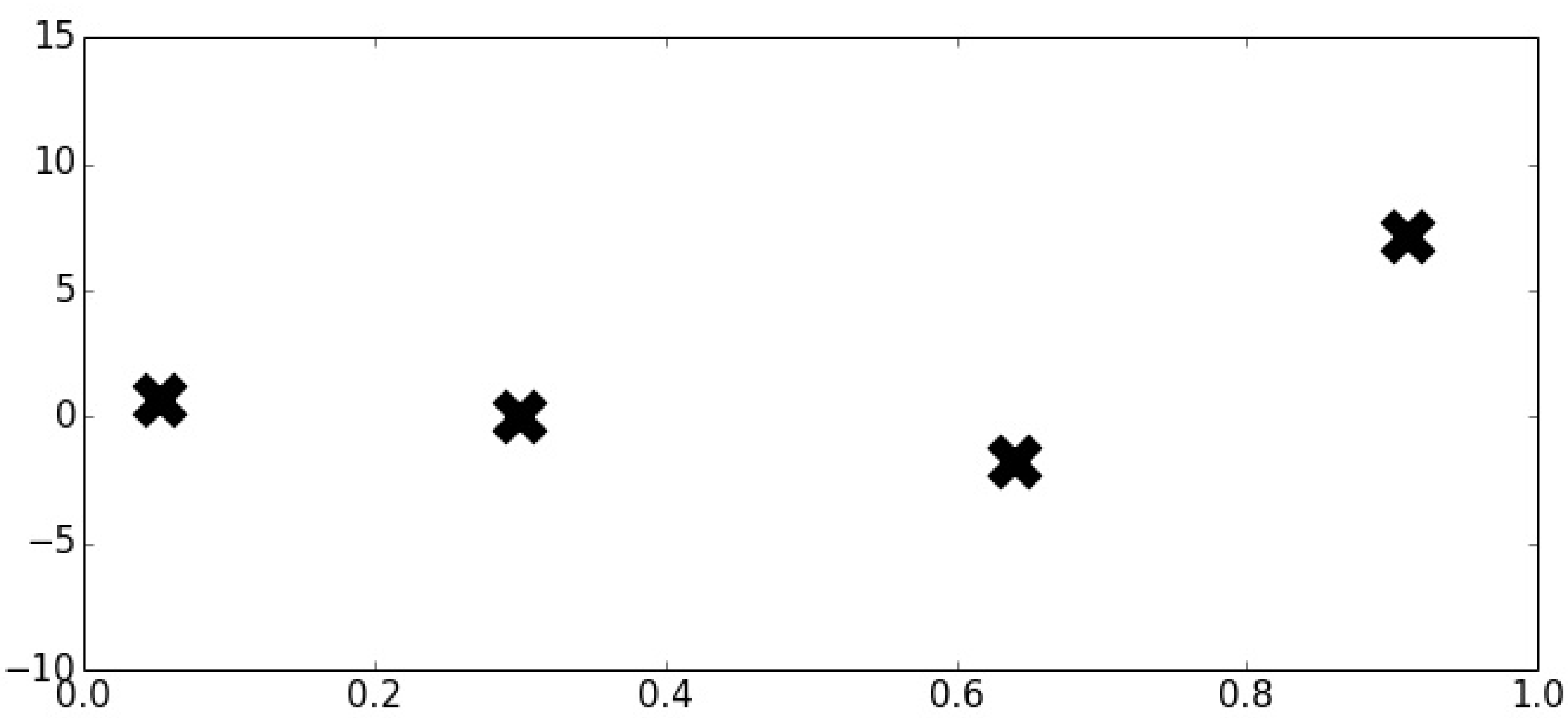
Prediction:

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

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

Bayesian optimization

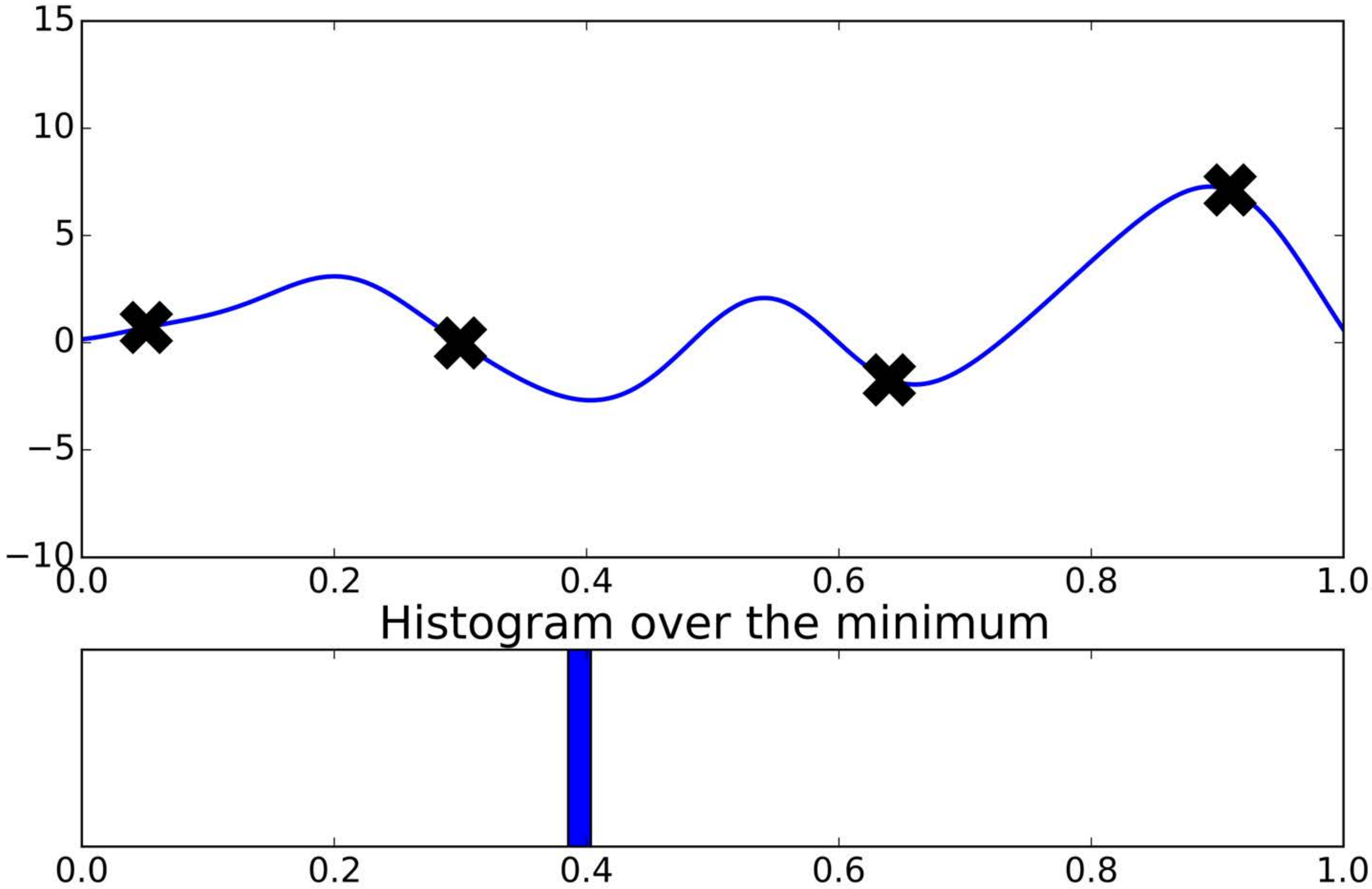


Where is the minimum of f?

Where should the take the next evaluation?

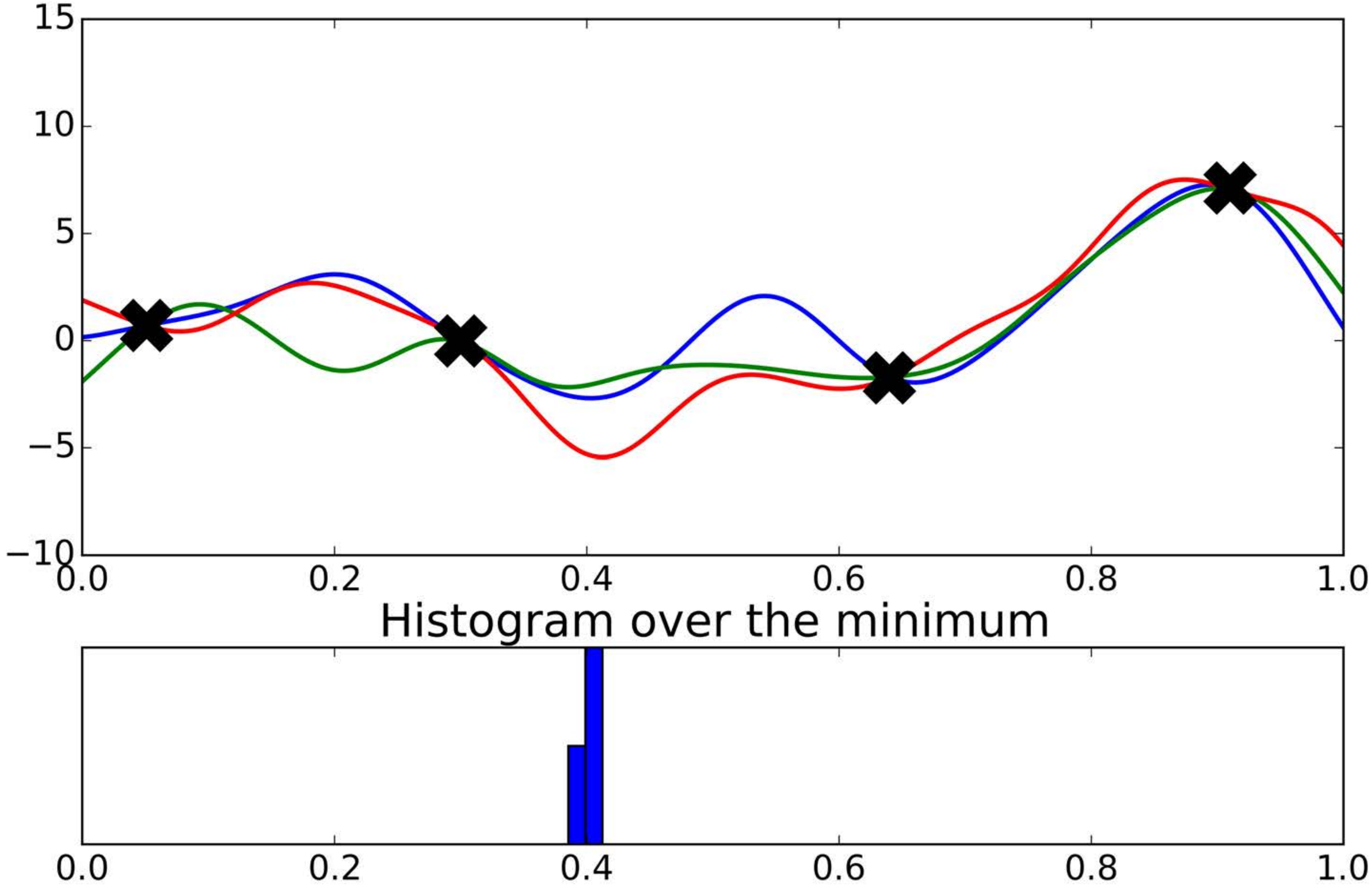
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



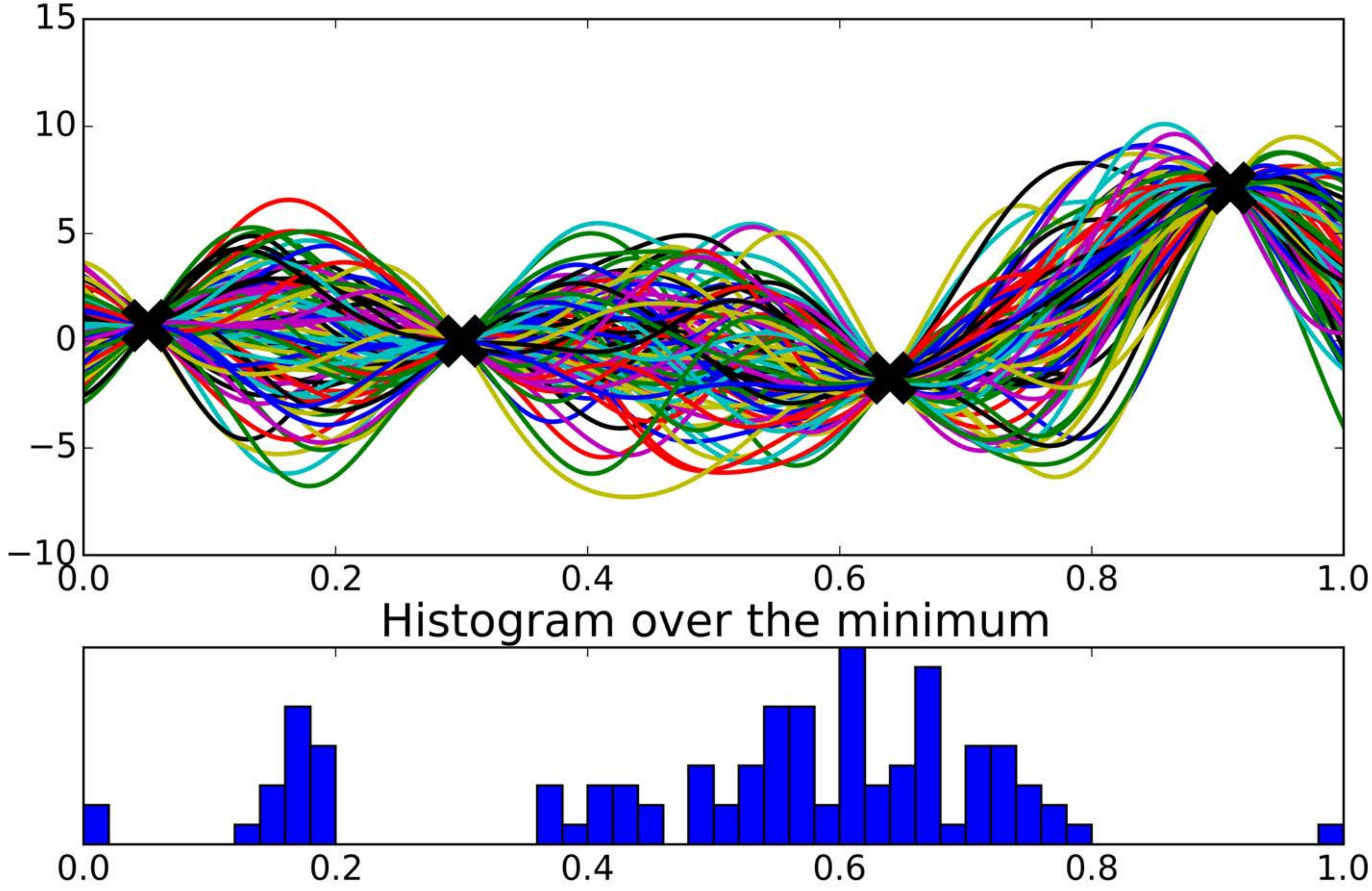
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



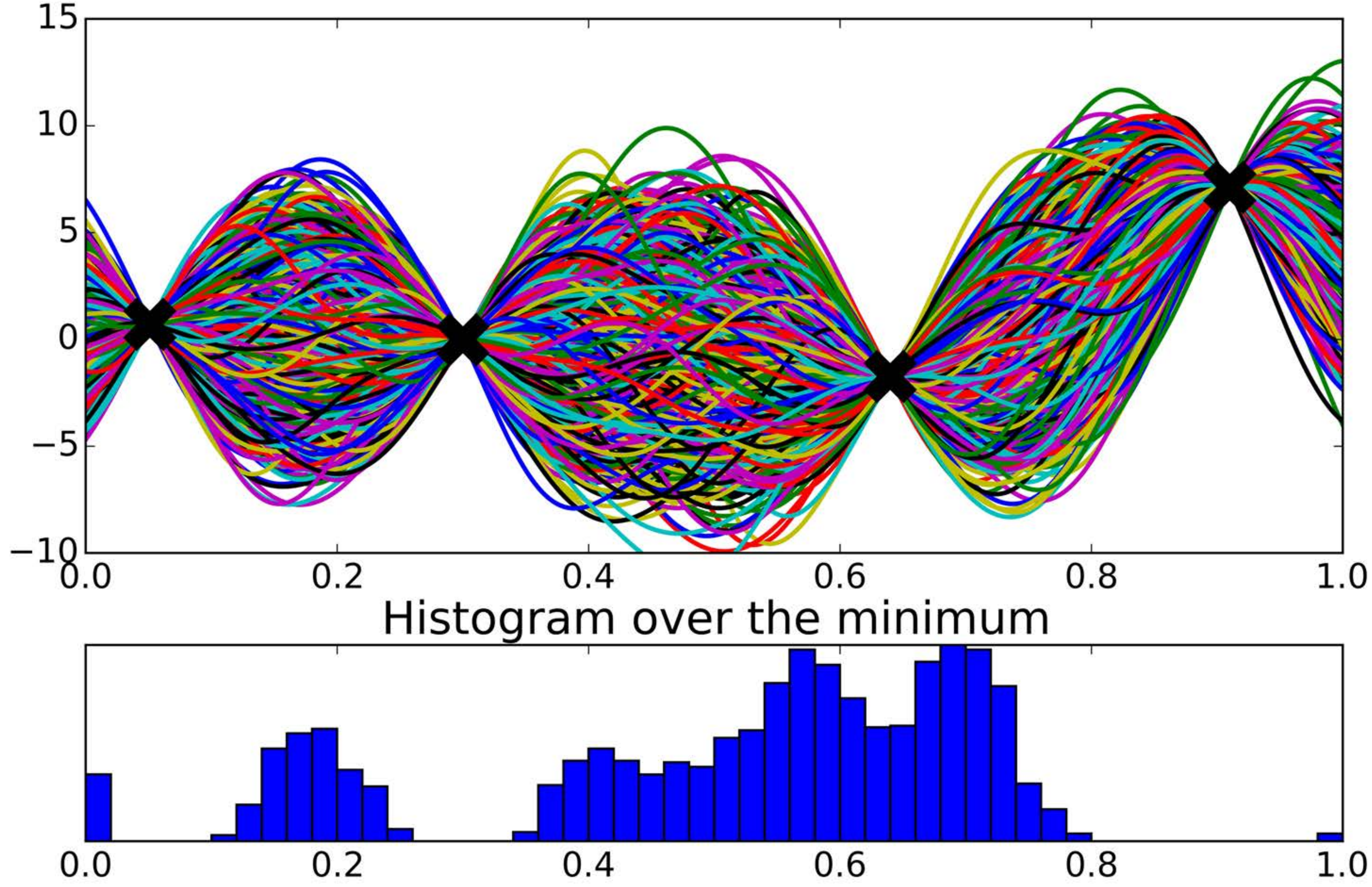
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



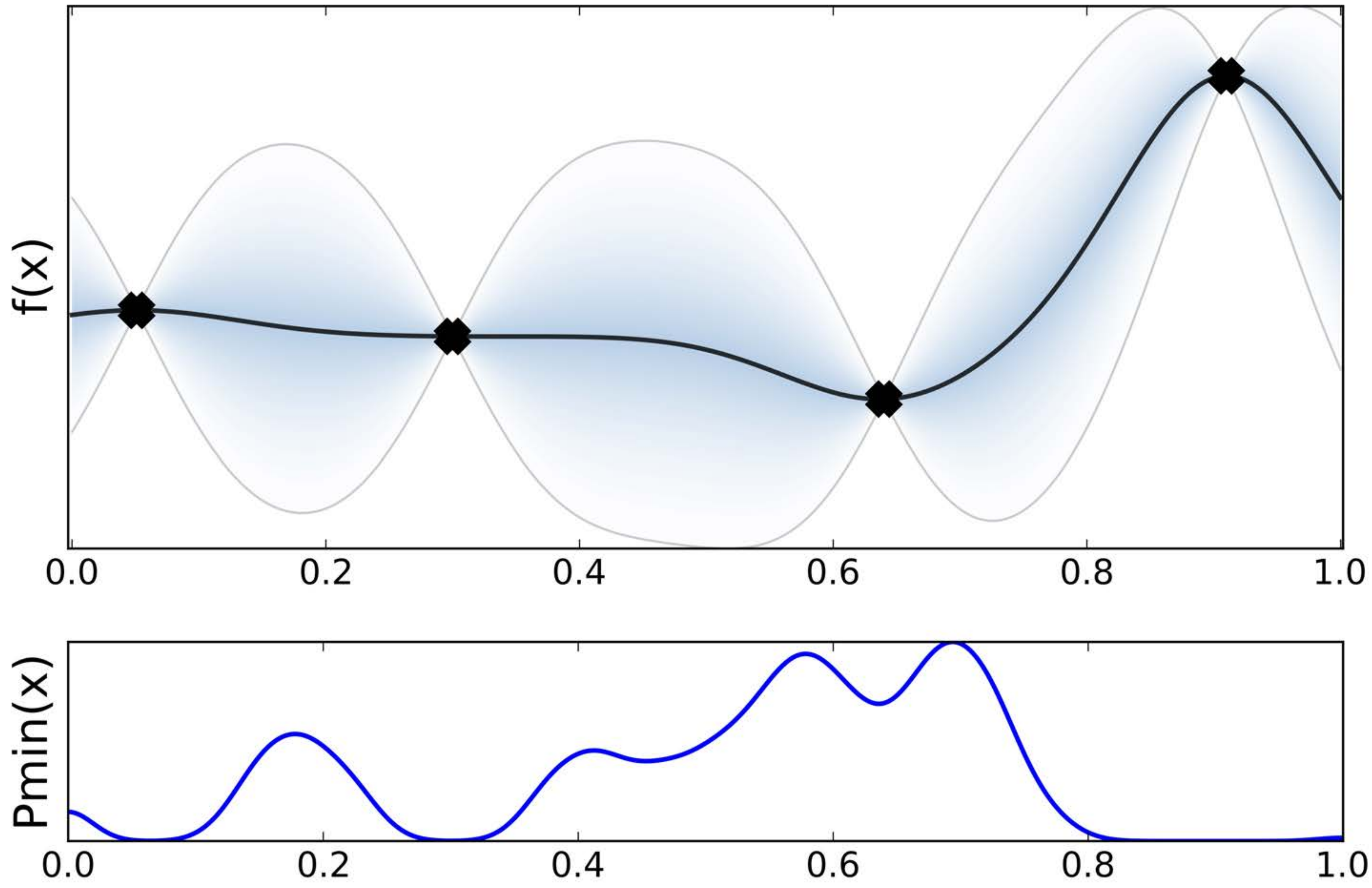
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



$x^* = \arg\min f(x)$ ..but $p(x^*|\mathcal{D})$ is not tractable.

http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

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(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

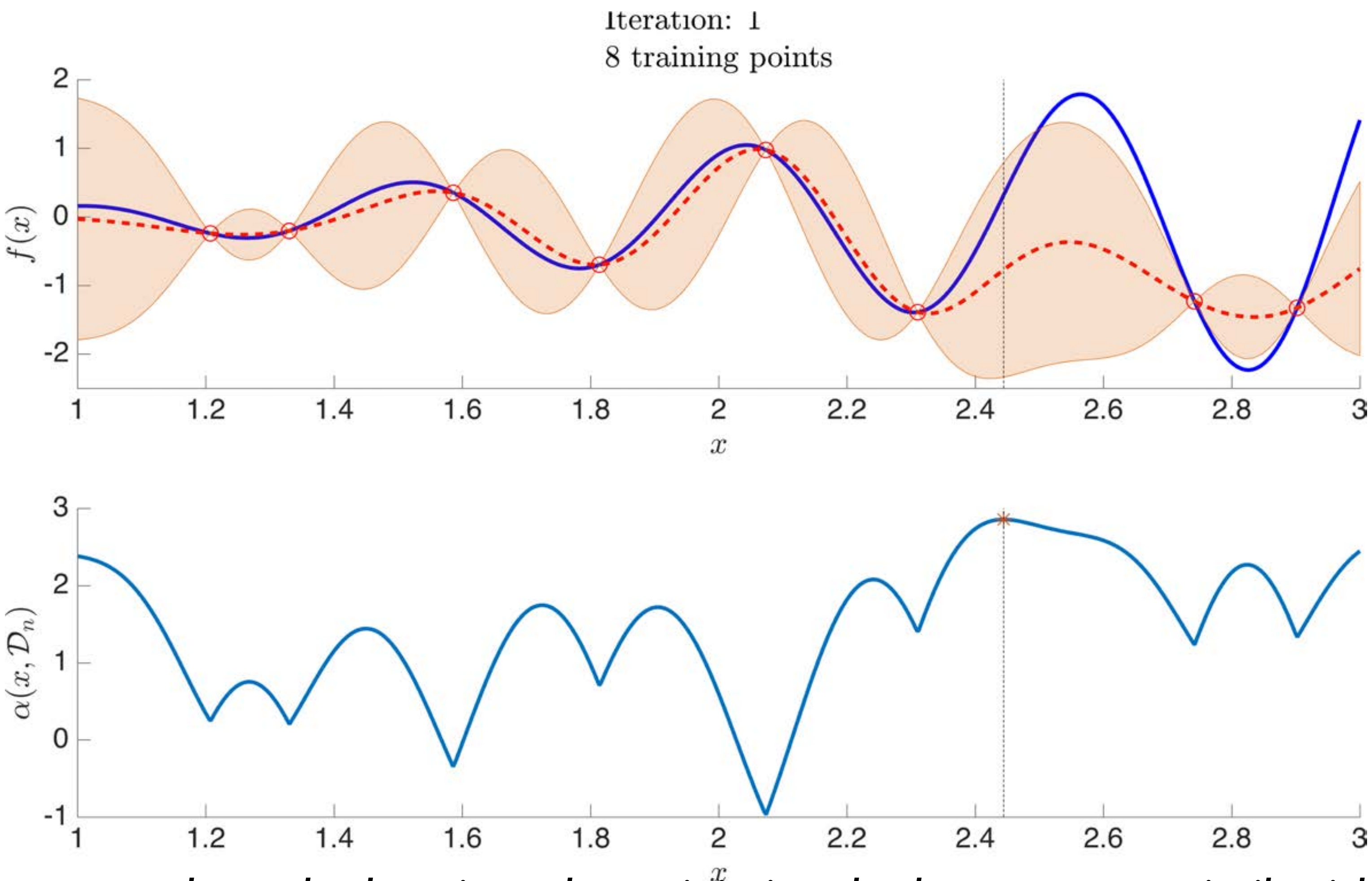
Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

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

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

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 minimize the lower super-quintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

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(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

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

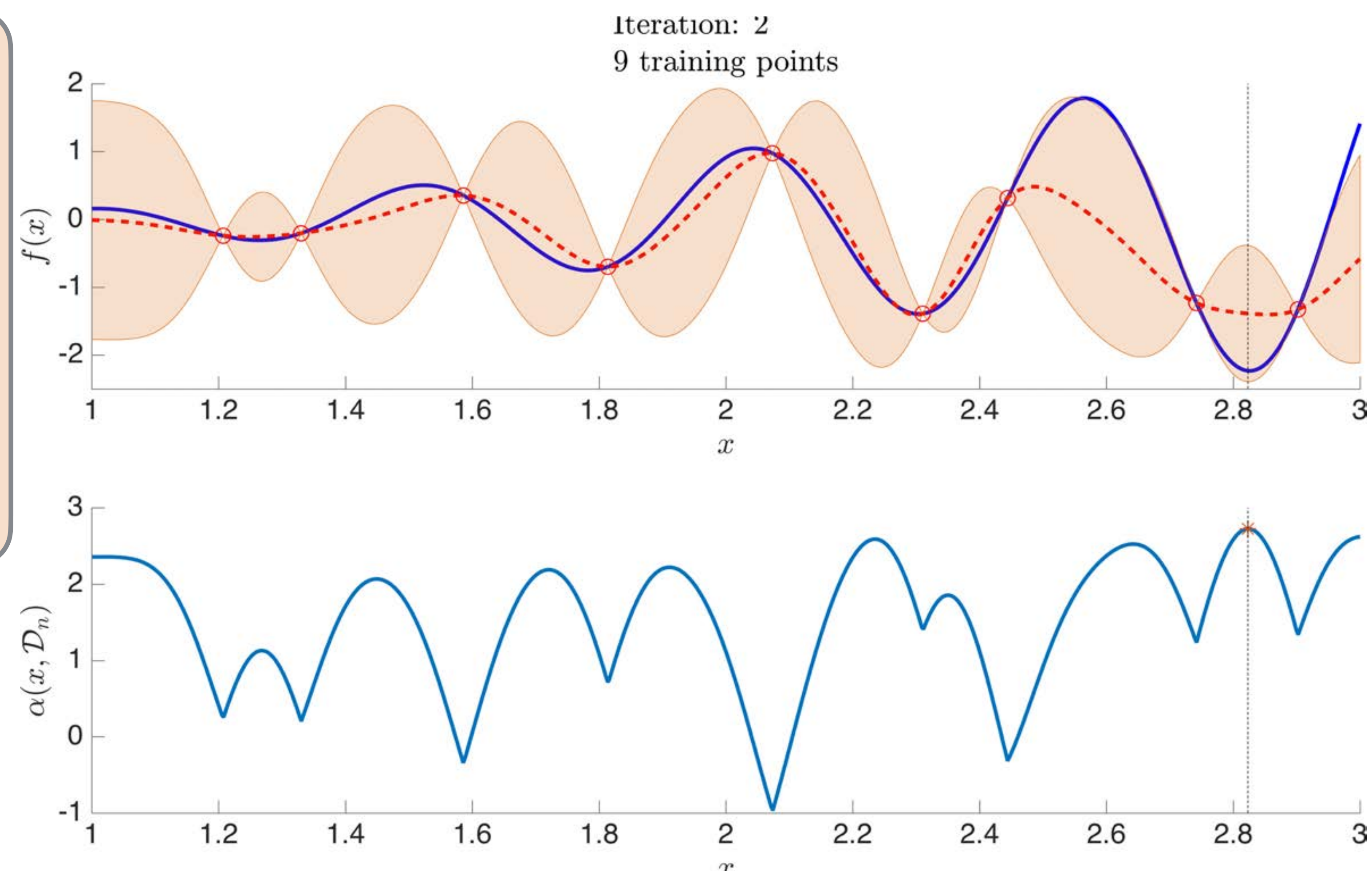
Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

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

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 minimize the lower super-quintile risk confidence bound*

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

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(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

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

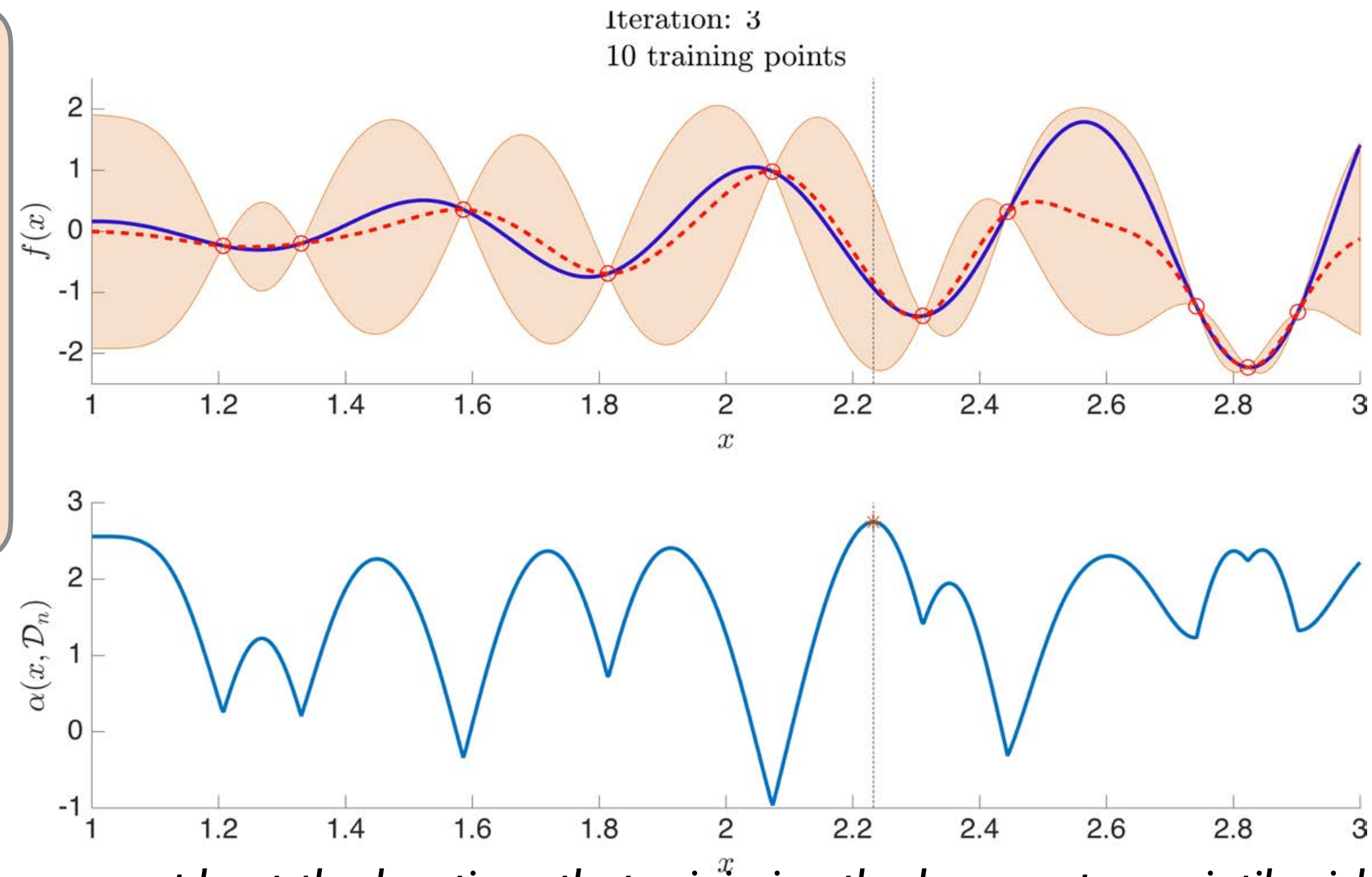
Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

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

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 minimize the lower super-quintile risk confidence bound*

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

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(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

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

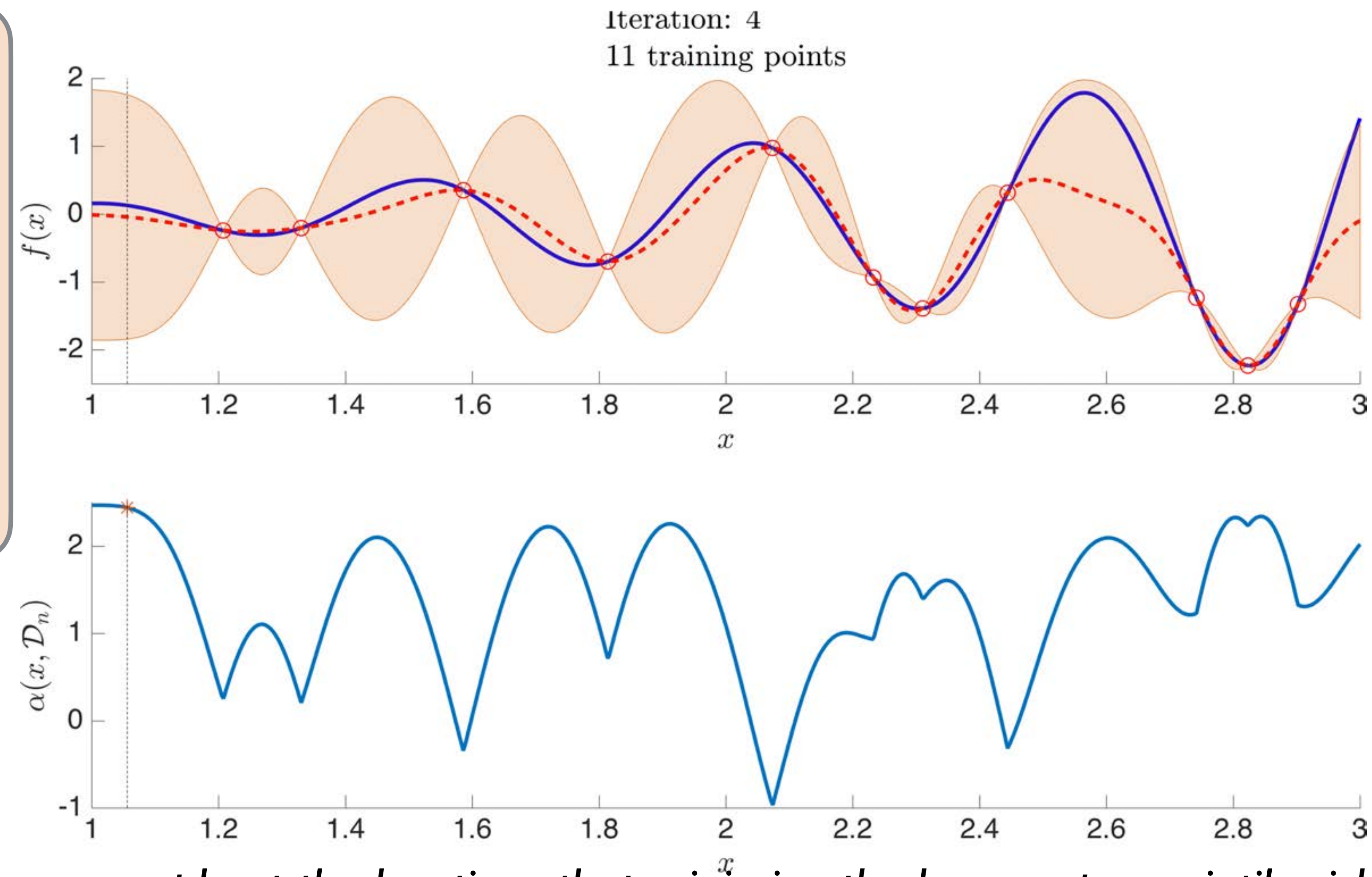
Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

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

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 minimize the lower super-quintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

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(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

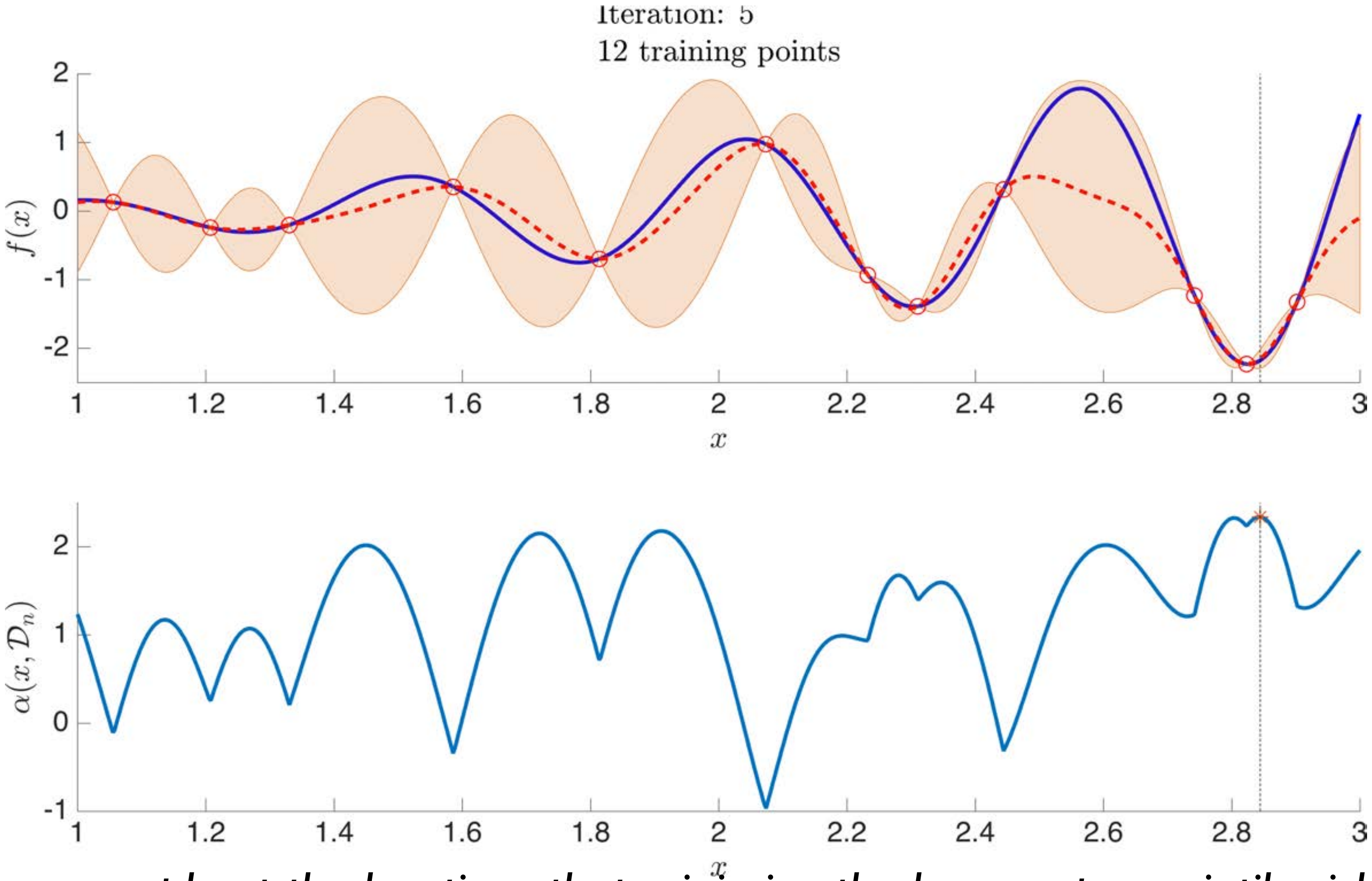
Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

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

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

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 minimize the lower super-quintile risk confidence bound*

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

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(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

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

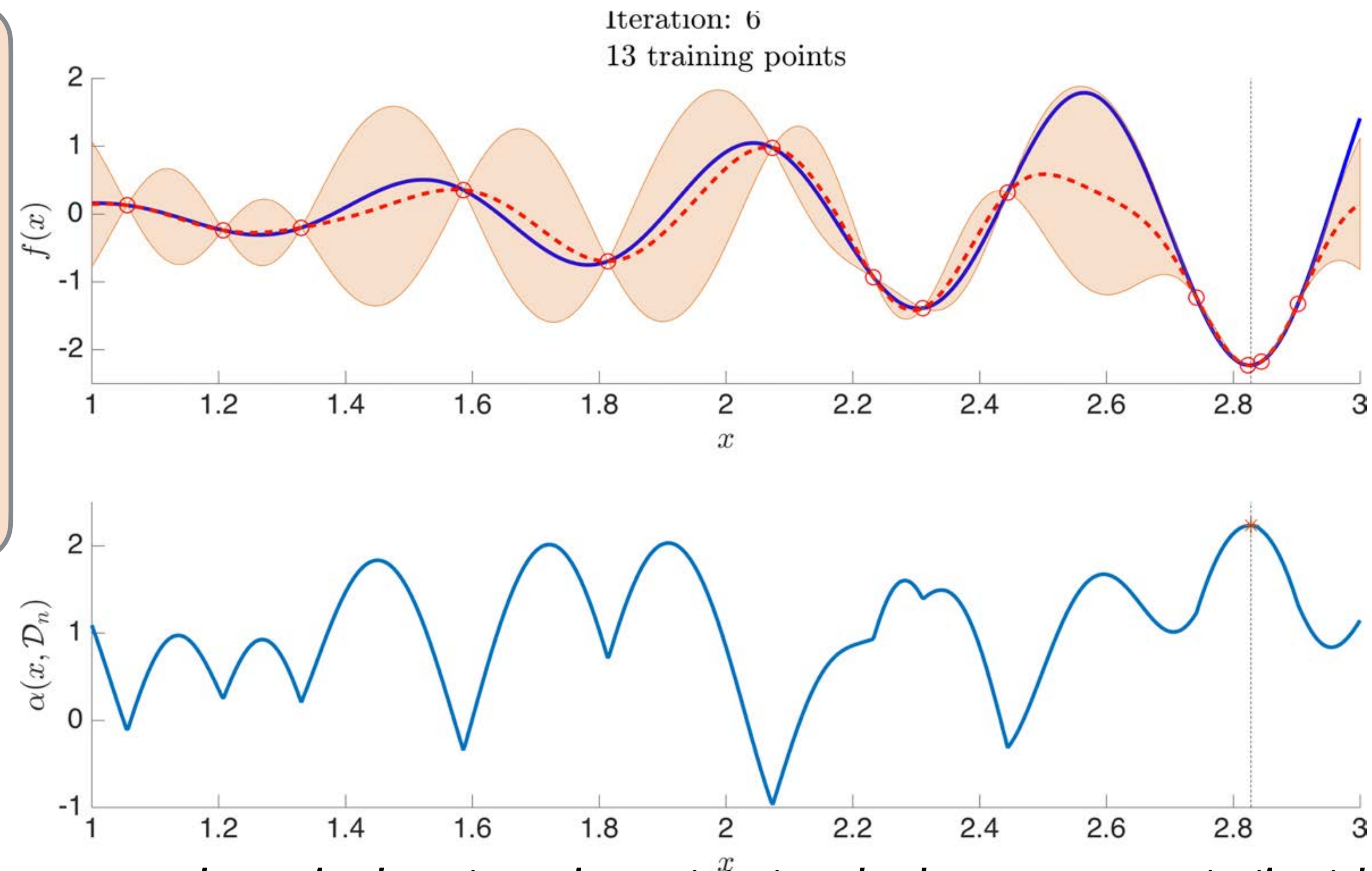
Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

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

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 minimize the lower super-quintile risk confidence bound*

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

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(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

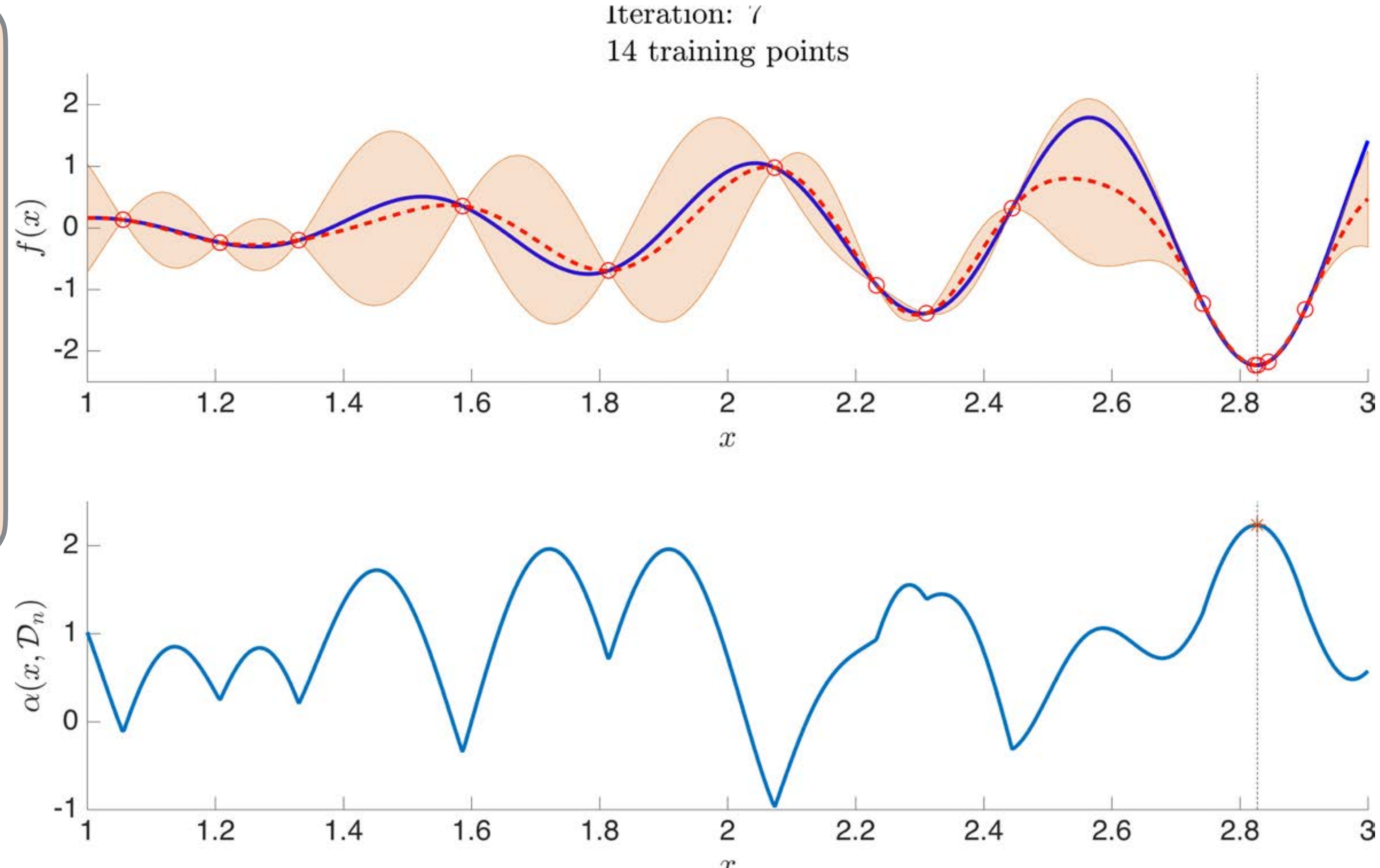
Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

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 minimize the lower super-quintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

Learning level sets

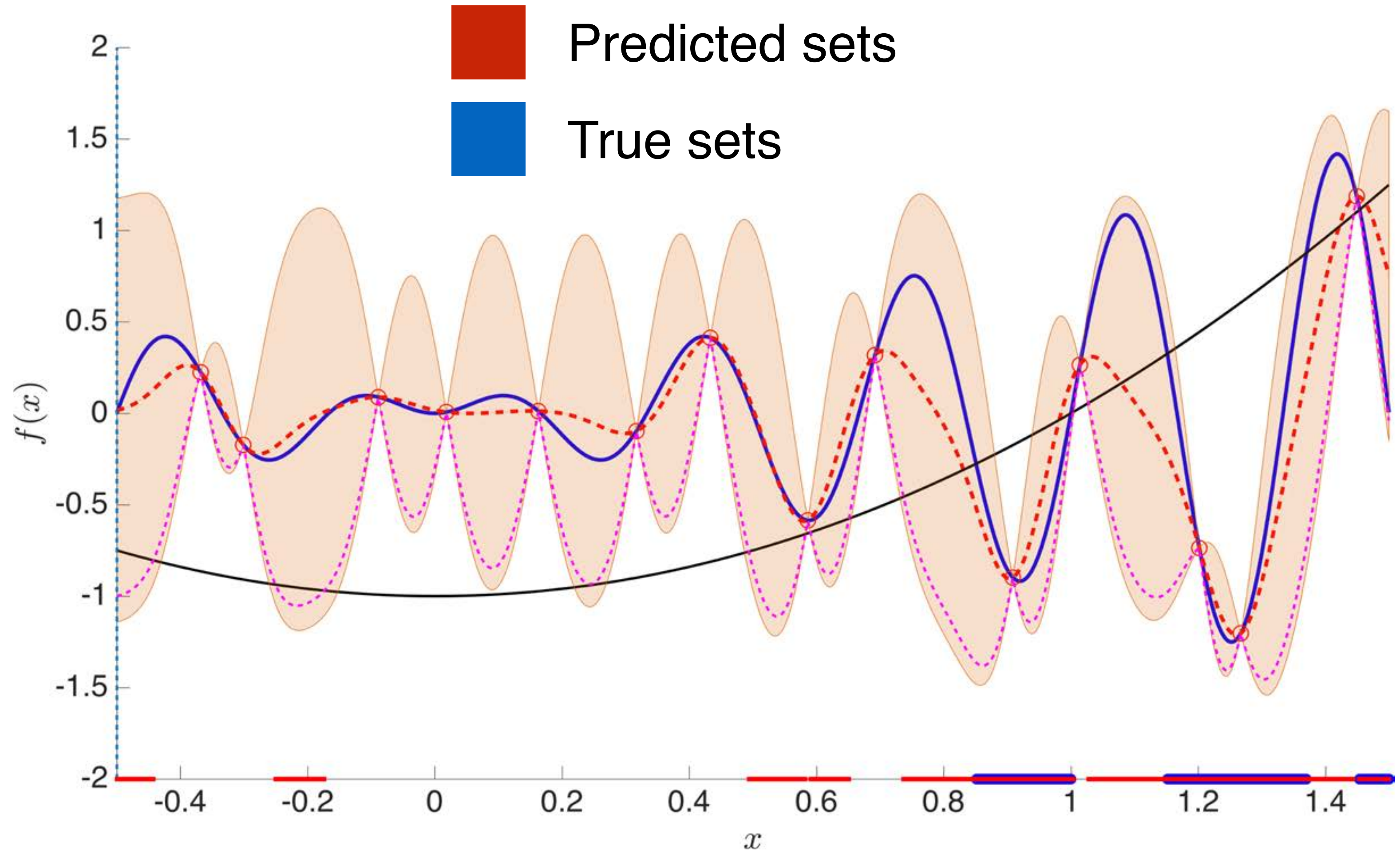
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \le t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

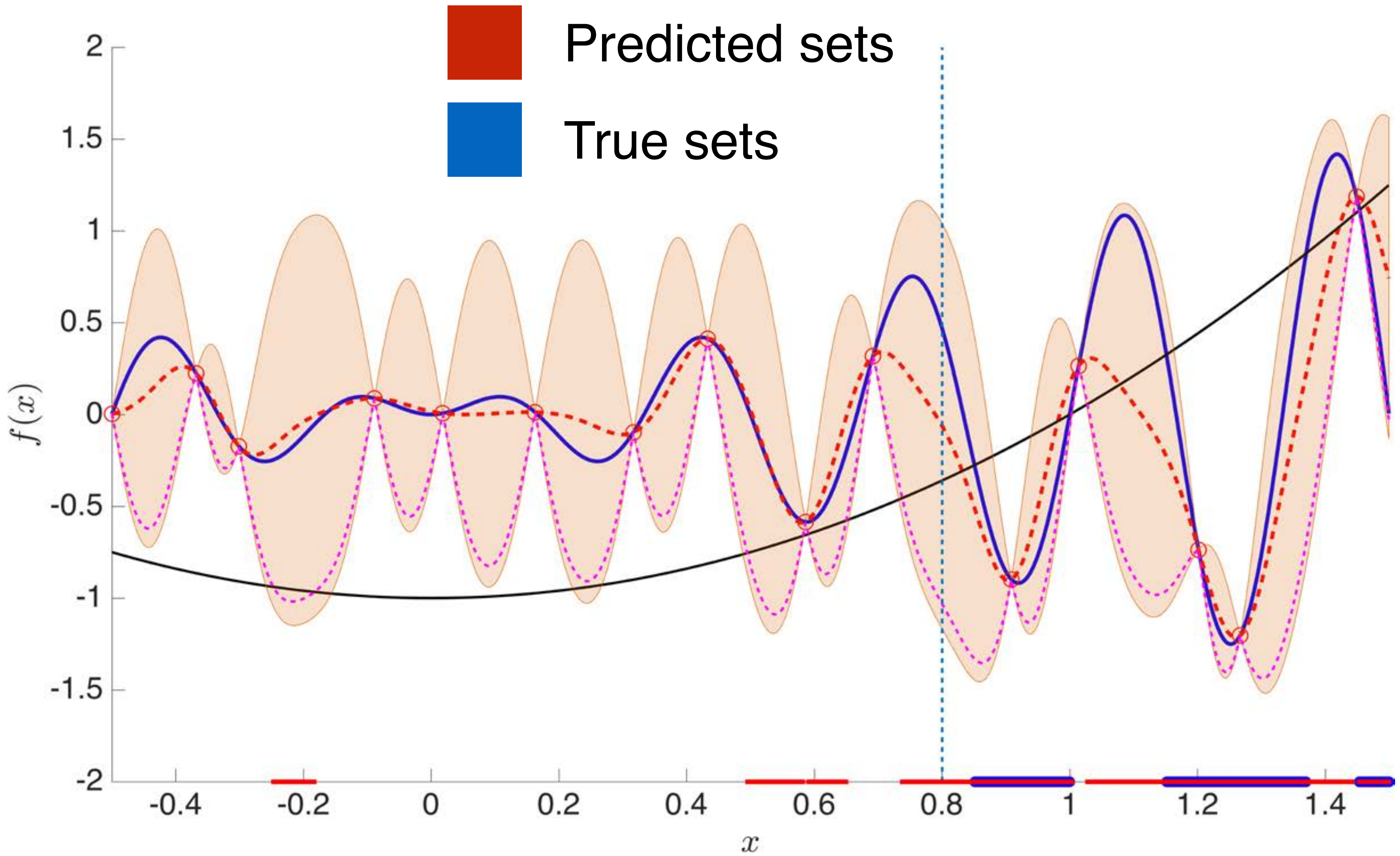
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

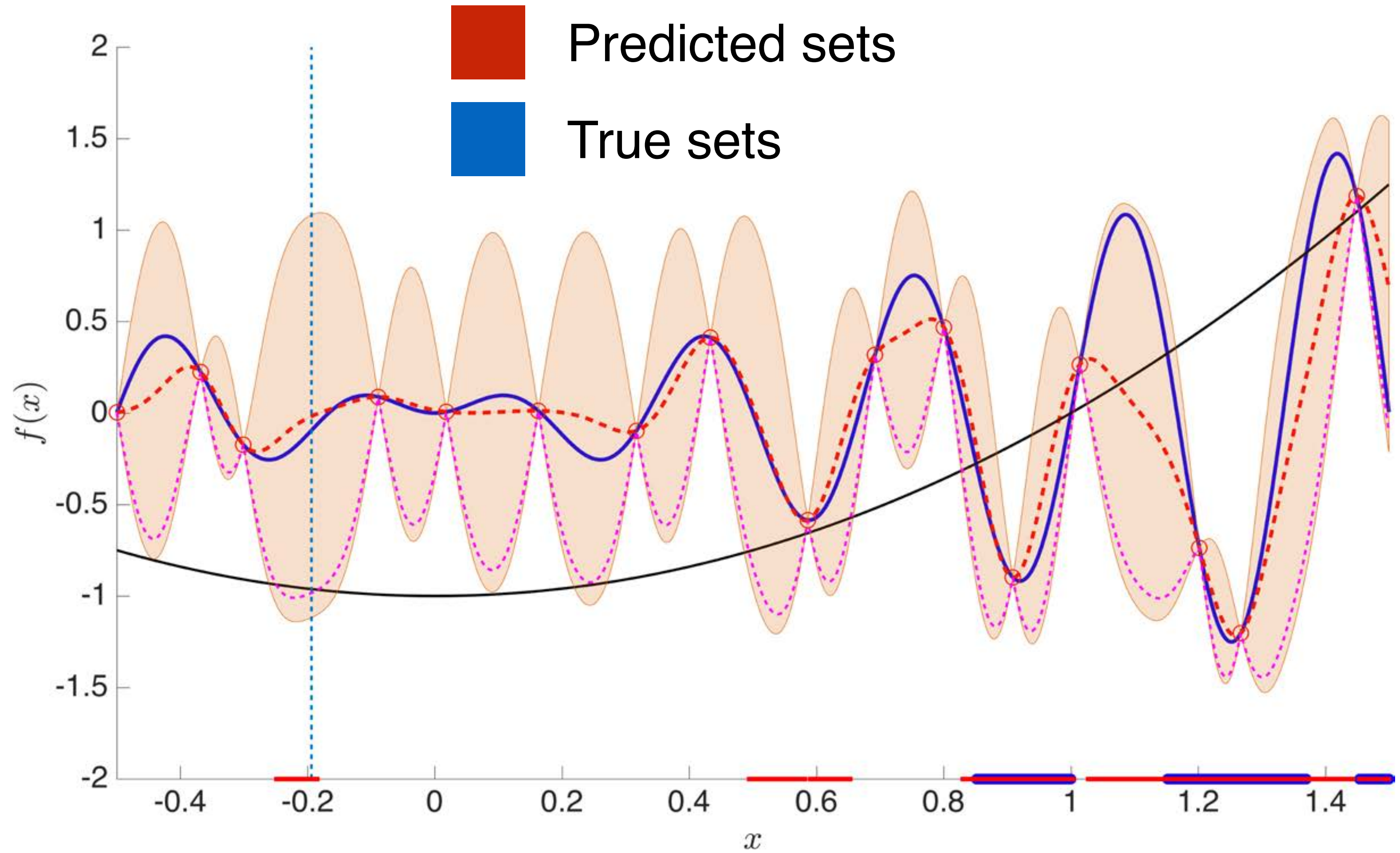
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

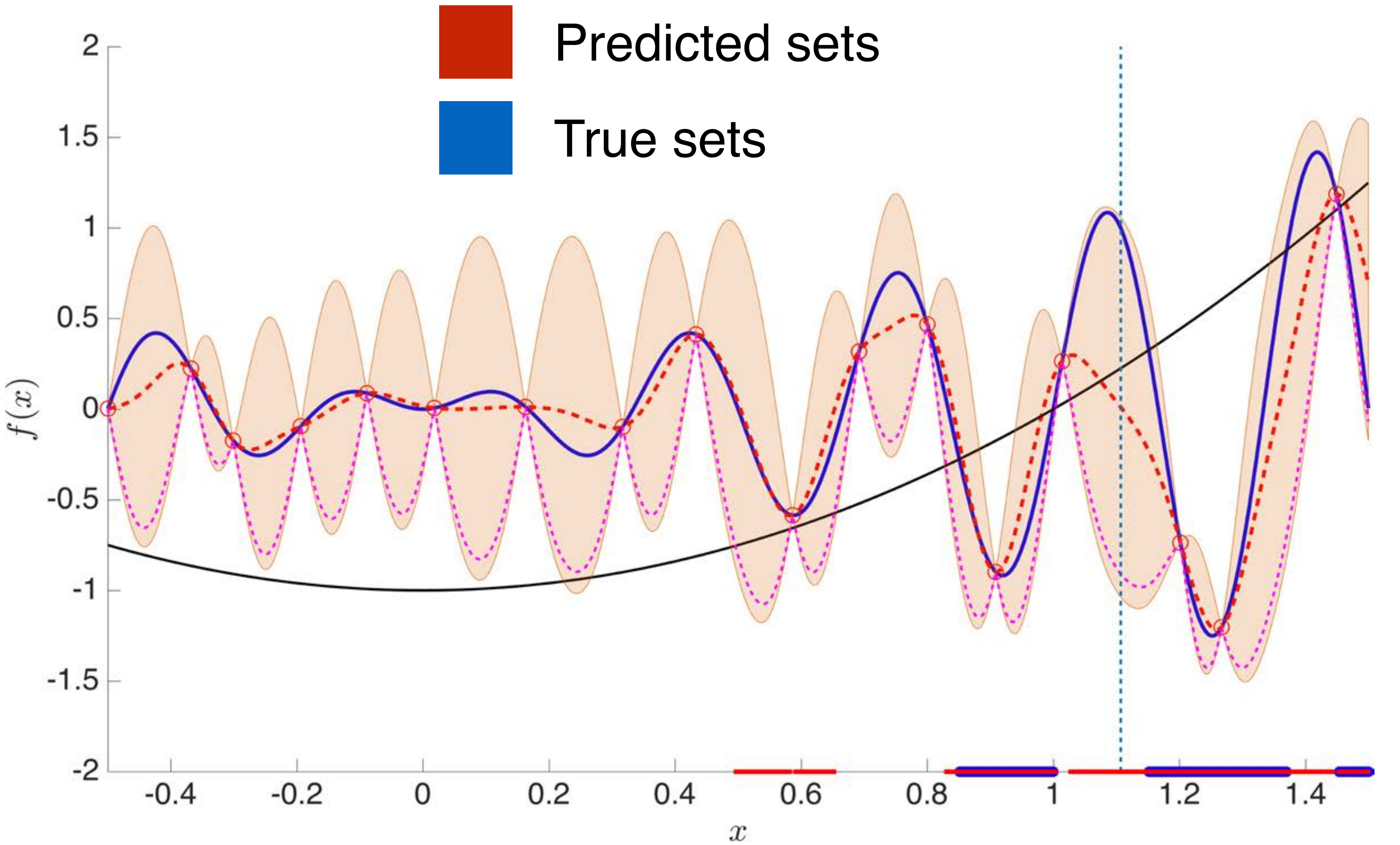
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

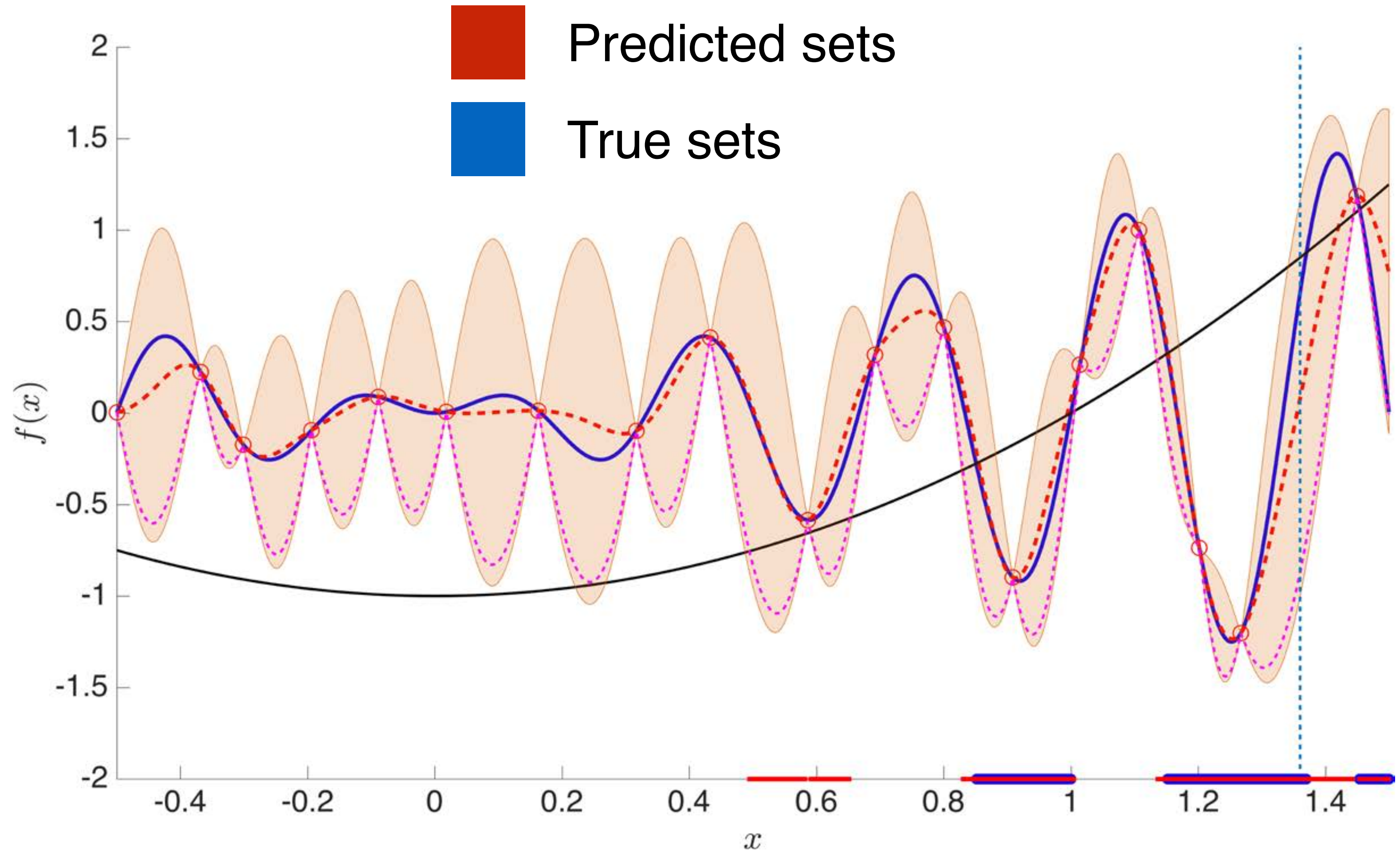
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

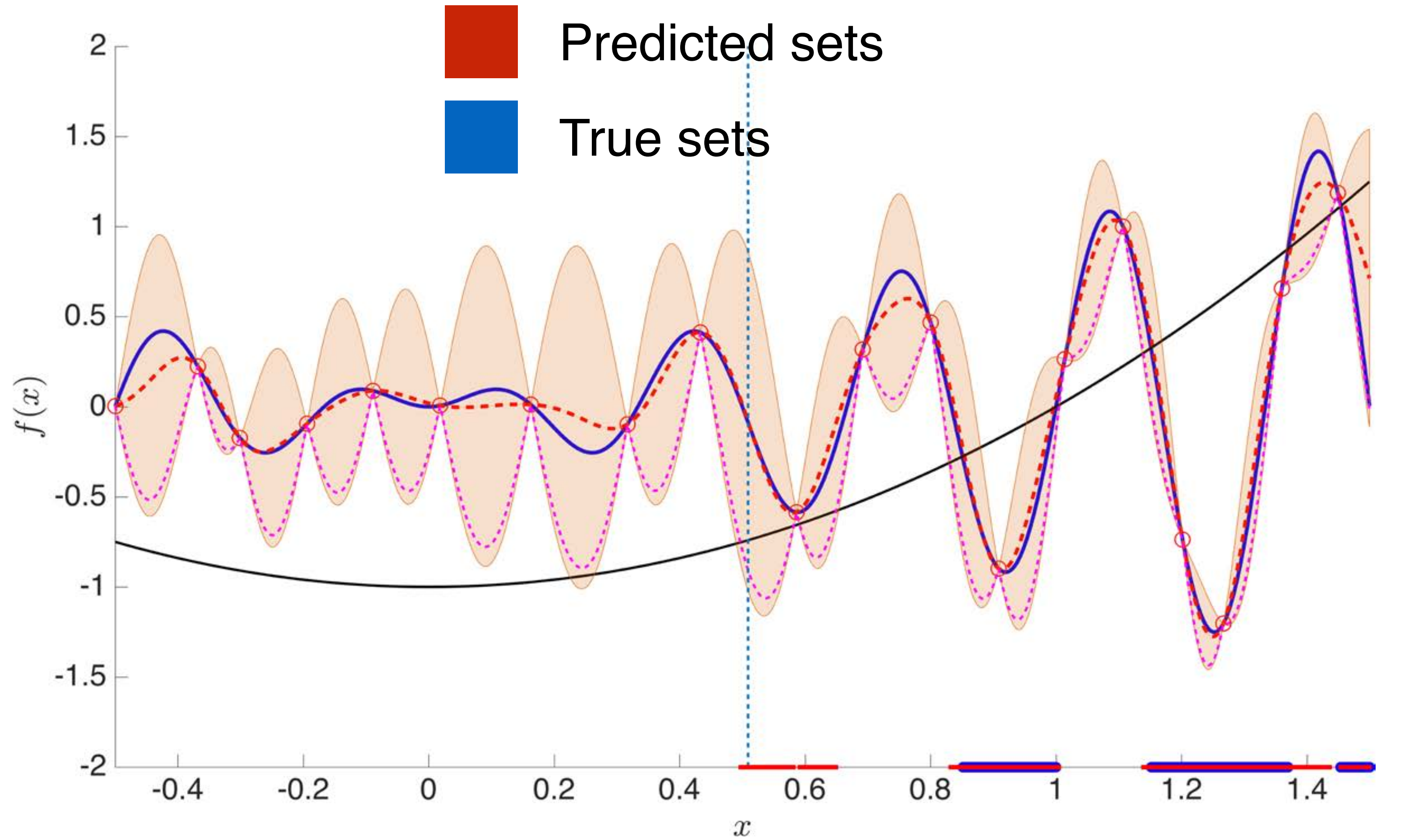
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

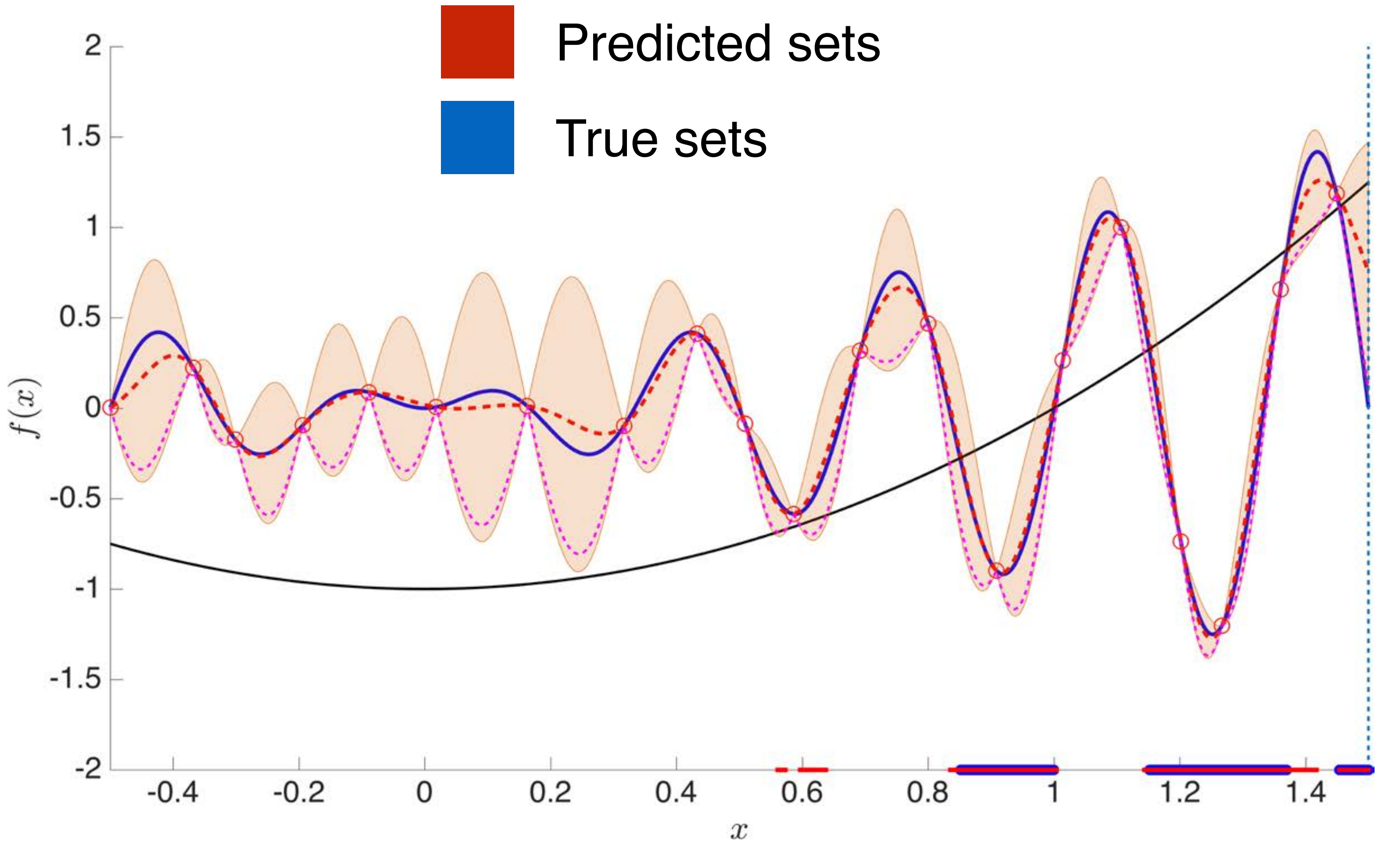
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

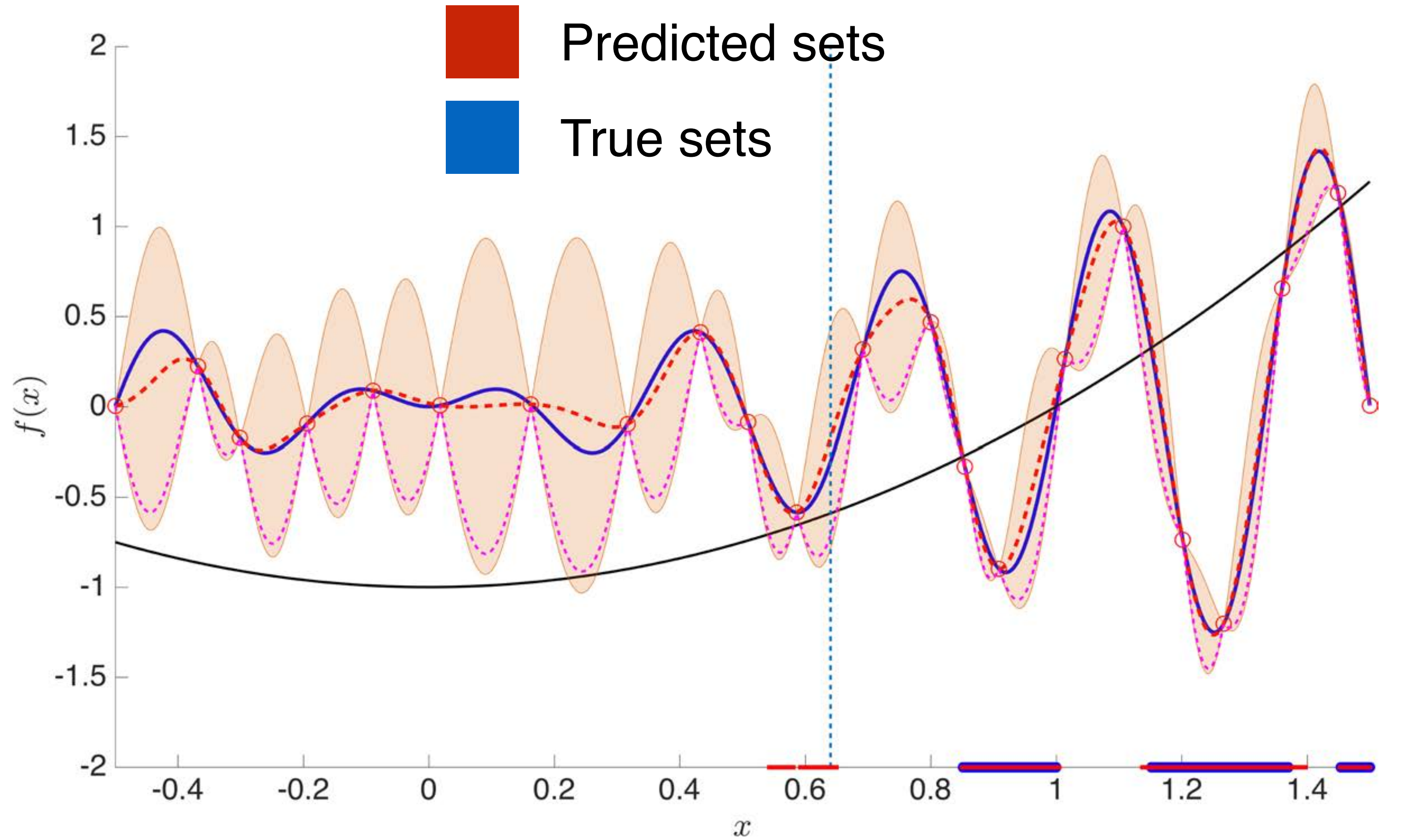
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Learning level sets

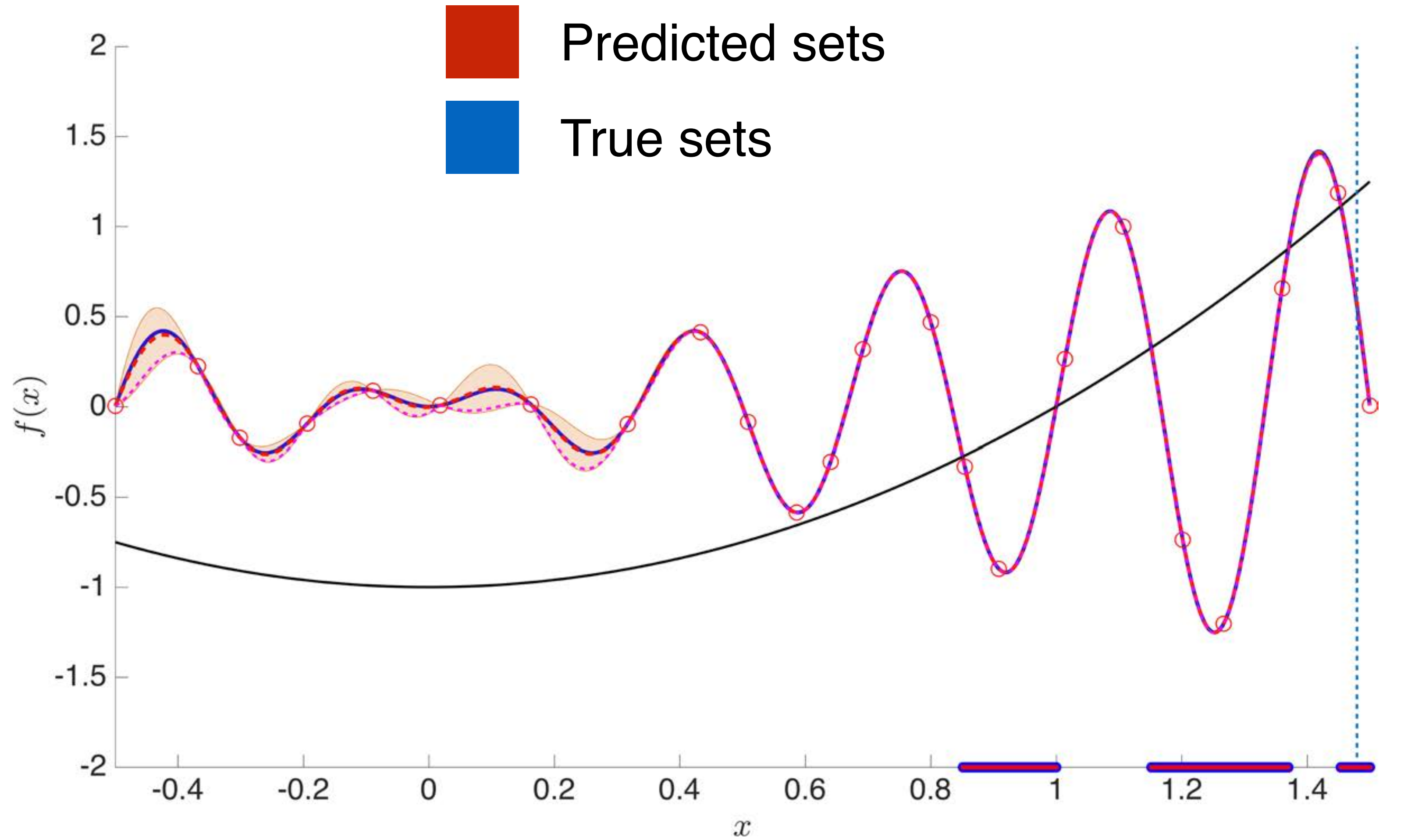
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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

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

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted sets are guaranteed to be a subset of the true sets.

Applications to control

PILCO: A Model-Based and Data-Efficient Approach to Policy Search

Marc Peter Deisenroth MARC@CS.WASHINGTON.EDU
Department of Computer Science & Engineering, University of Washington, USA

Carl Edward Rasmussen CER54@CAM.AC.UK
Department of Engineering, University of Cambridge, UK

Gaussian Processes for Data-Efficient Learning in Robotics and Control

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen