

ENM531: Data-driven modeling and probabilistic scientific computing

Lecture #20: Gaussian Processes

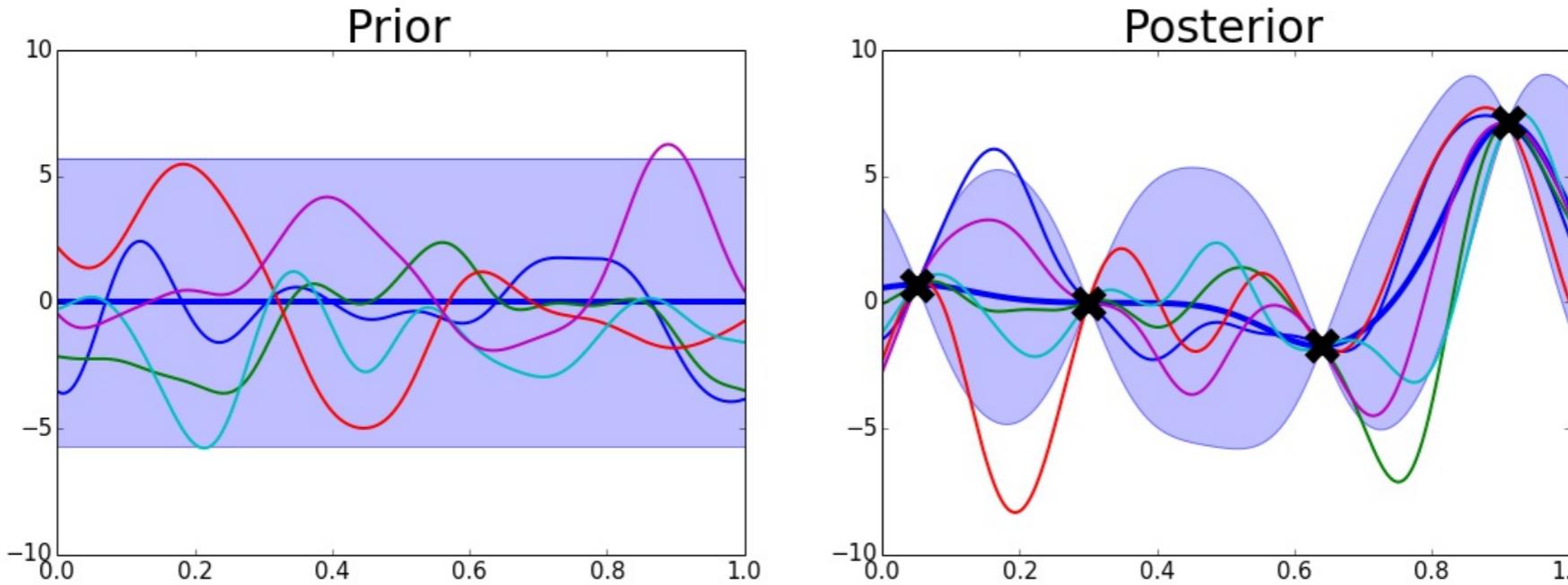
Paris Perdikaris
April 16, 2019



Data-driven modeling with Gaussian processes

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

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



Training via maximizing the marginal likelihood

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

Prediction via conditioning on available data

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

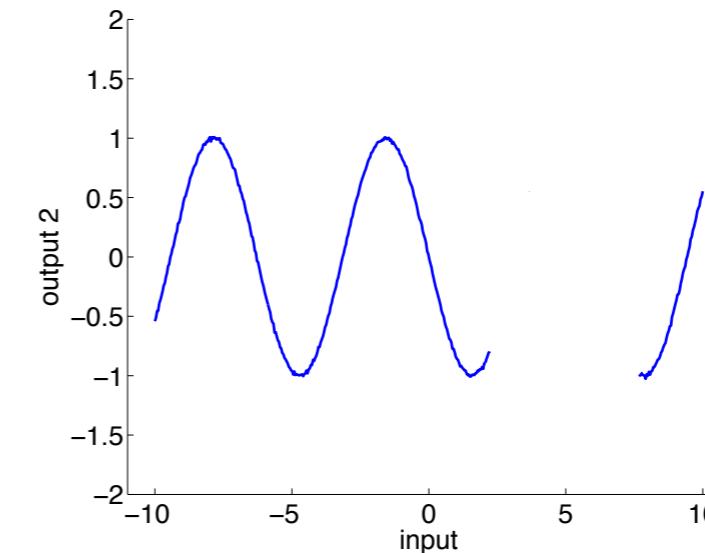
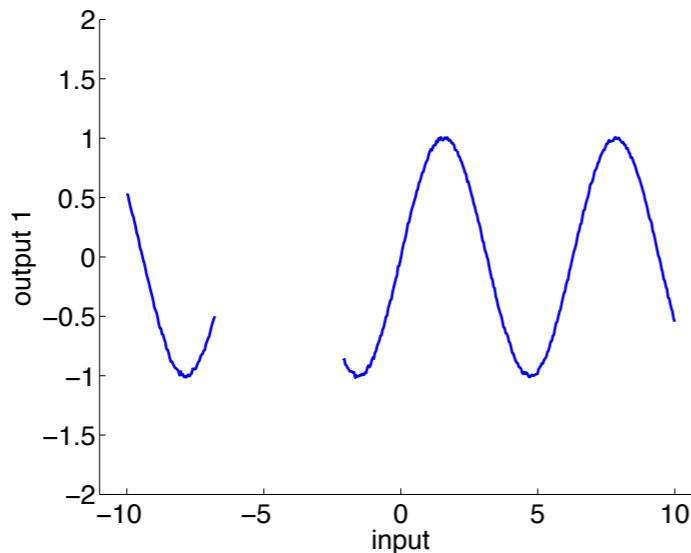
$$\mu_*(\mathbf{x}_*) = \mathbf{k}_{*N} (\mathbf{K} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y},$$

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

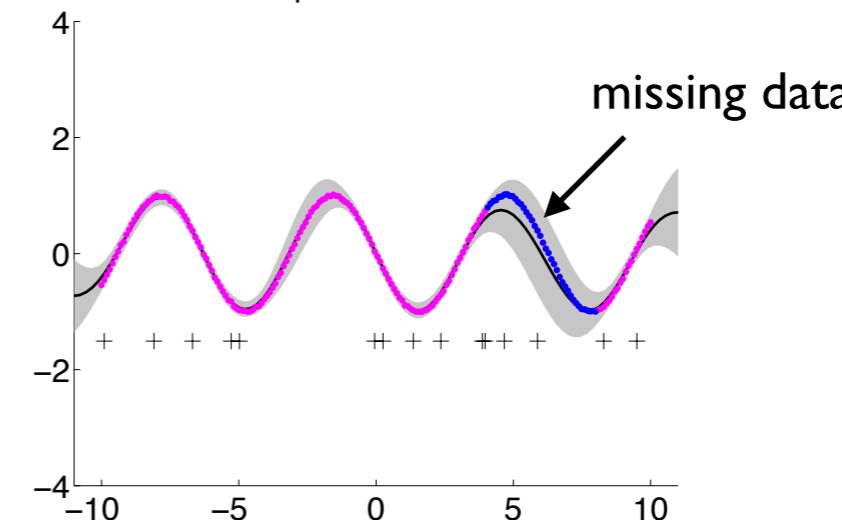
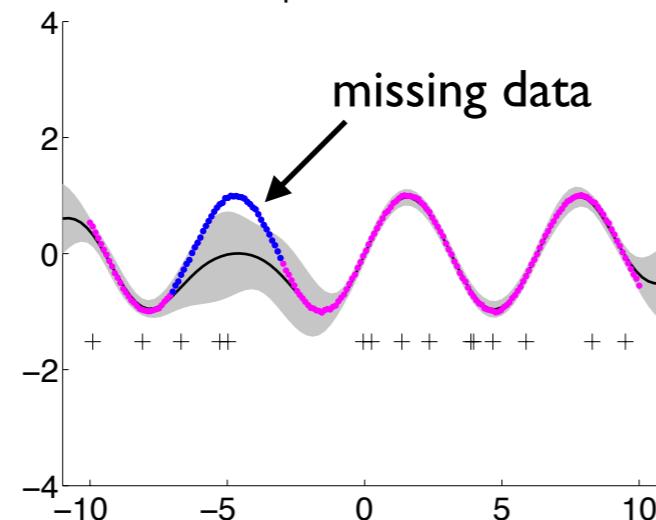
Multi-output Gaussian process regression

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

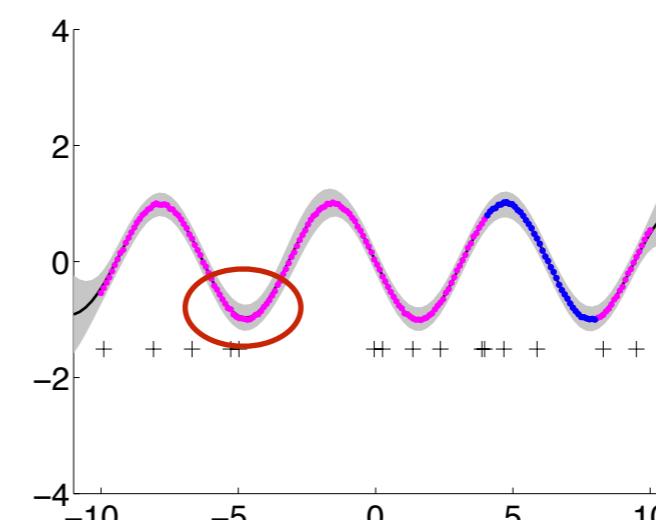
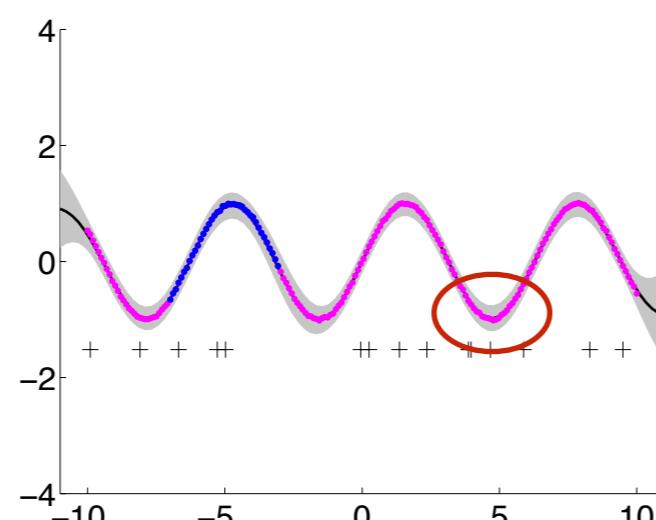
$$\text{output 2} = -\text{output 1}$$



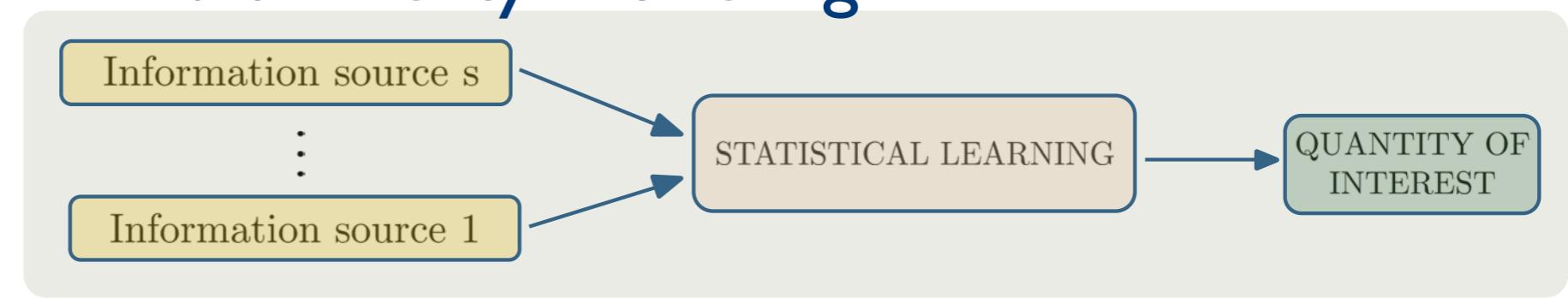
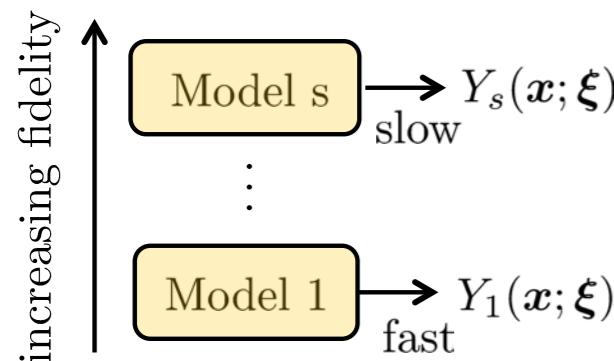
independent learning



COGP



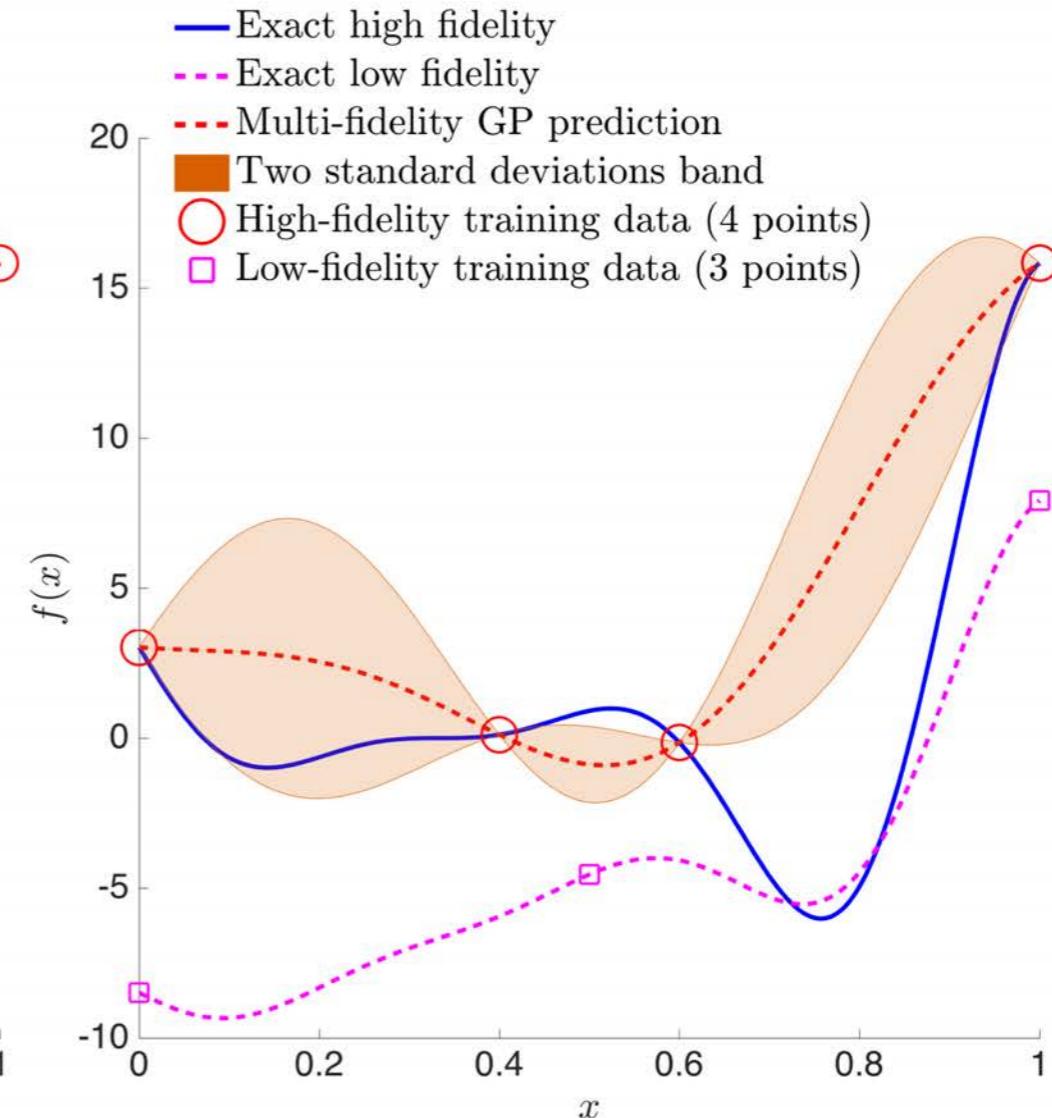
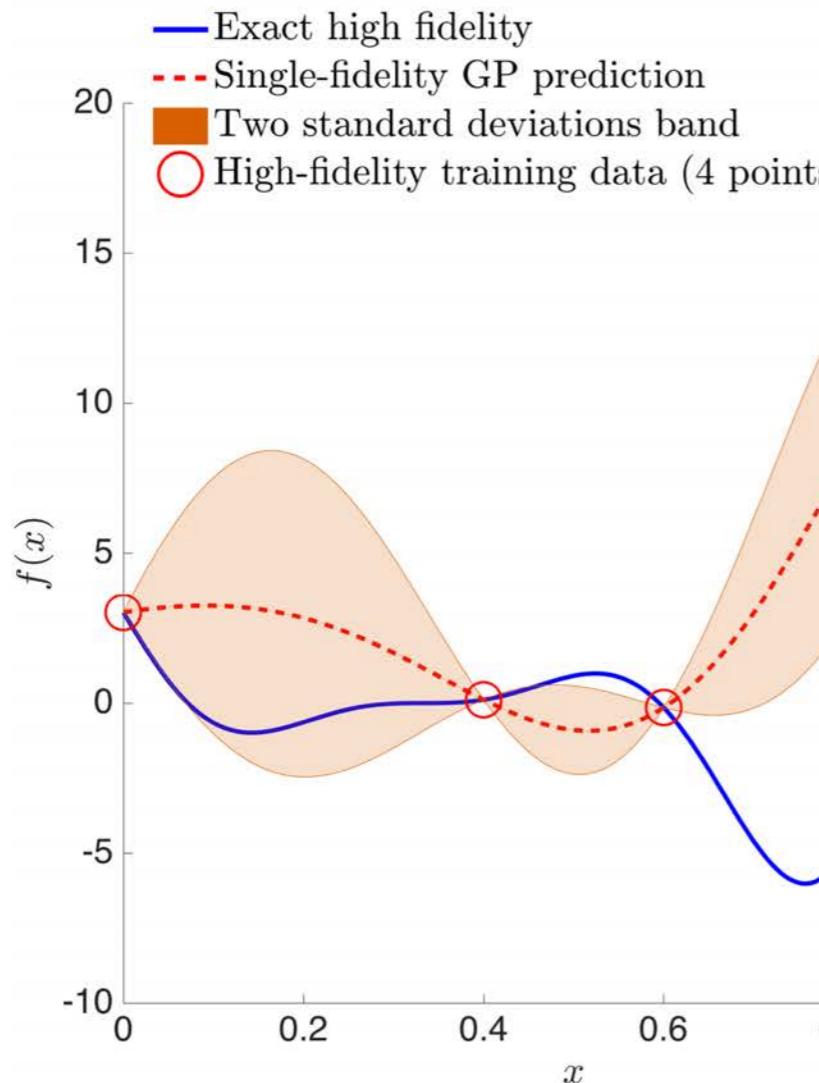
Multi-fidelity modeling



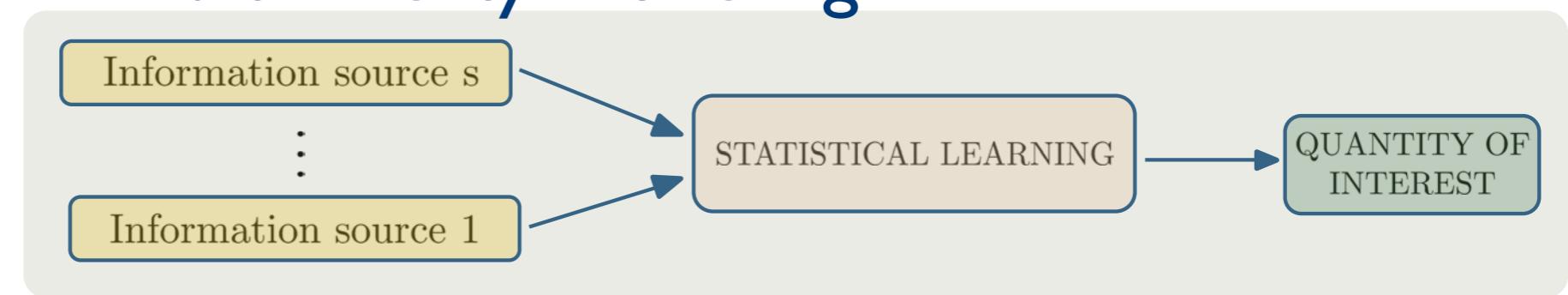
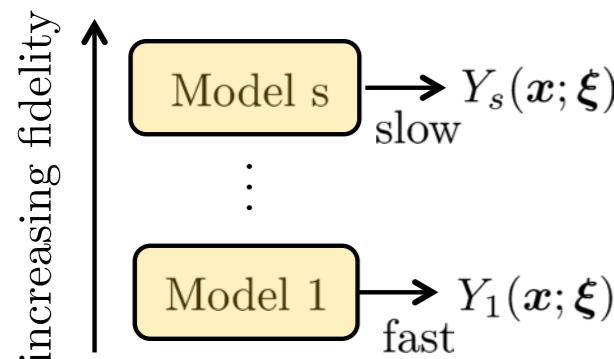
Number of runs is limited by time
and computational resources

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**



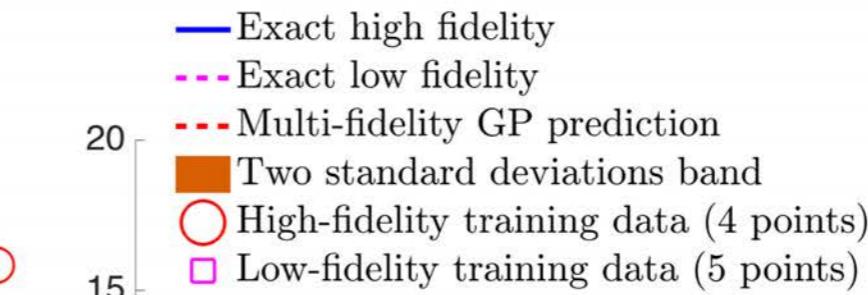
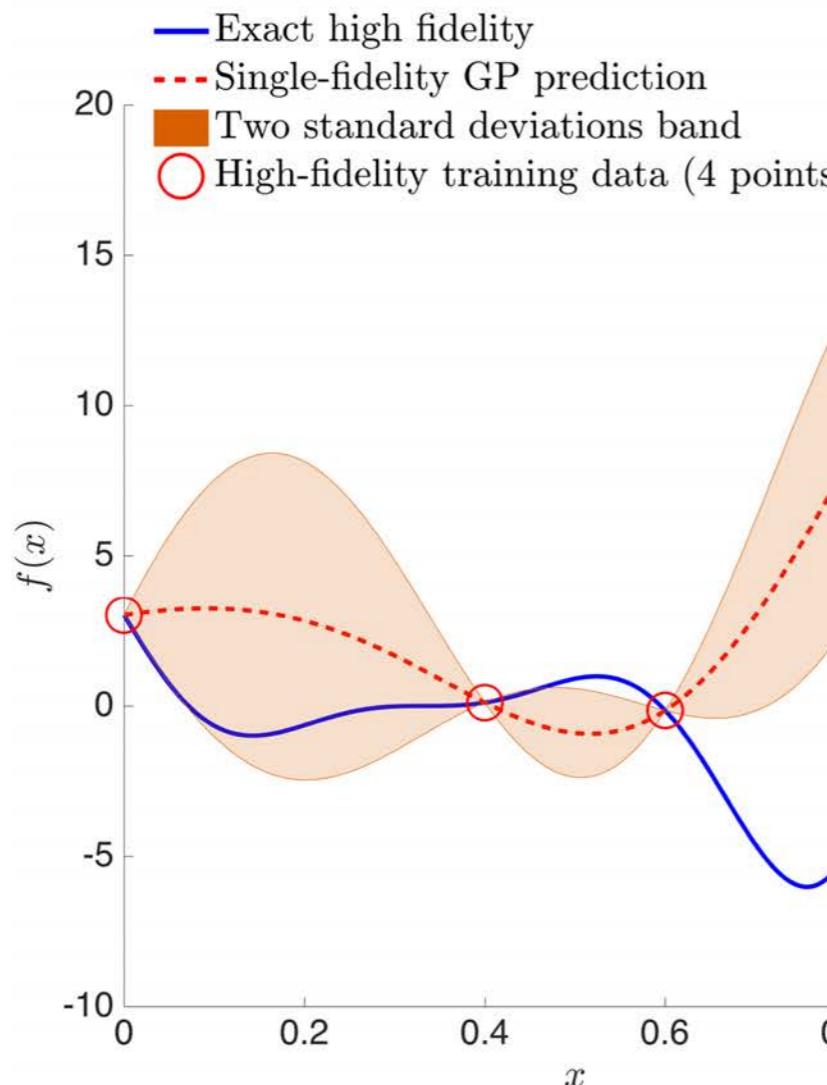
Multi-fidelity modeling



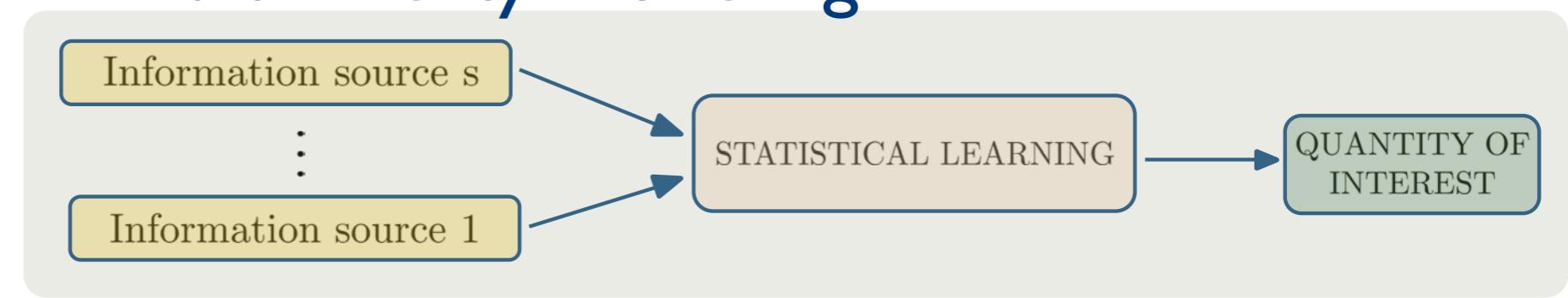
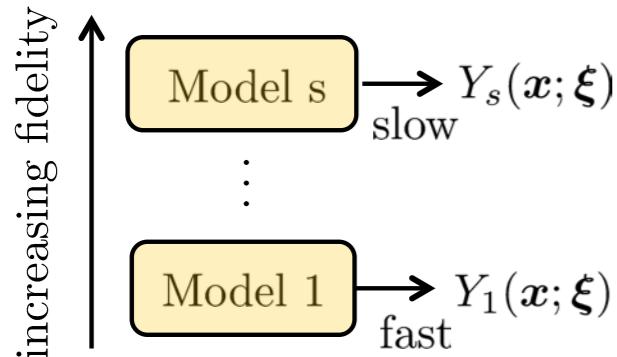
Number of runs is limited by time
and computational resources

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

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



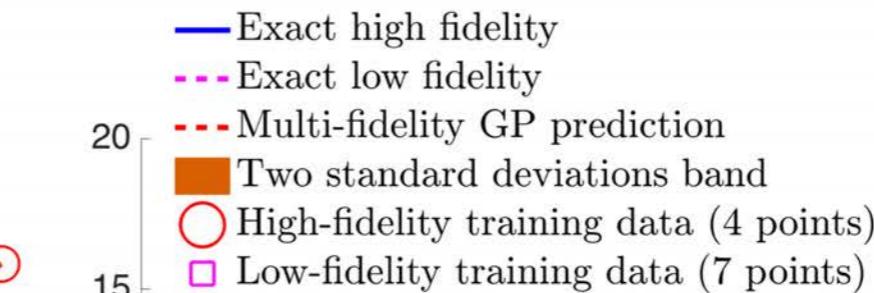
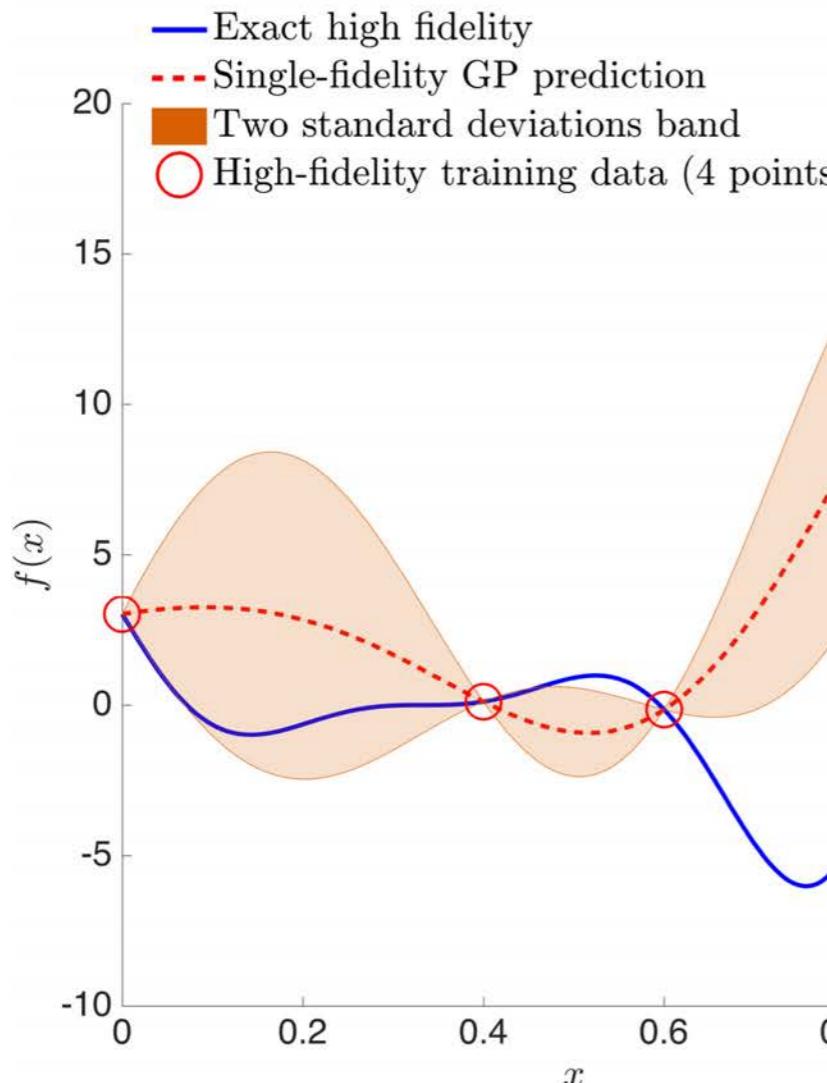
Multi-fidelity modeling



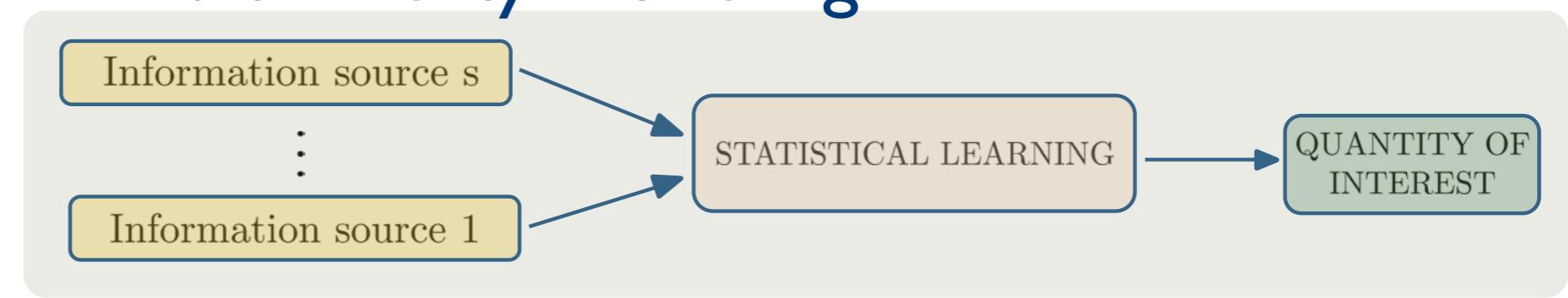
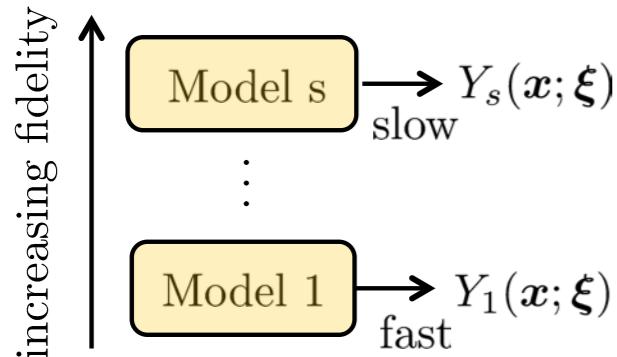
Number of runs is limited by time
and computational resources

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**



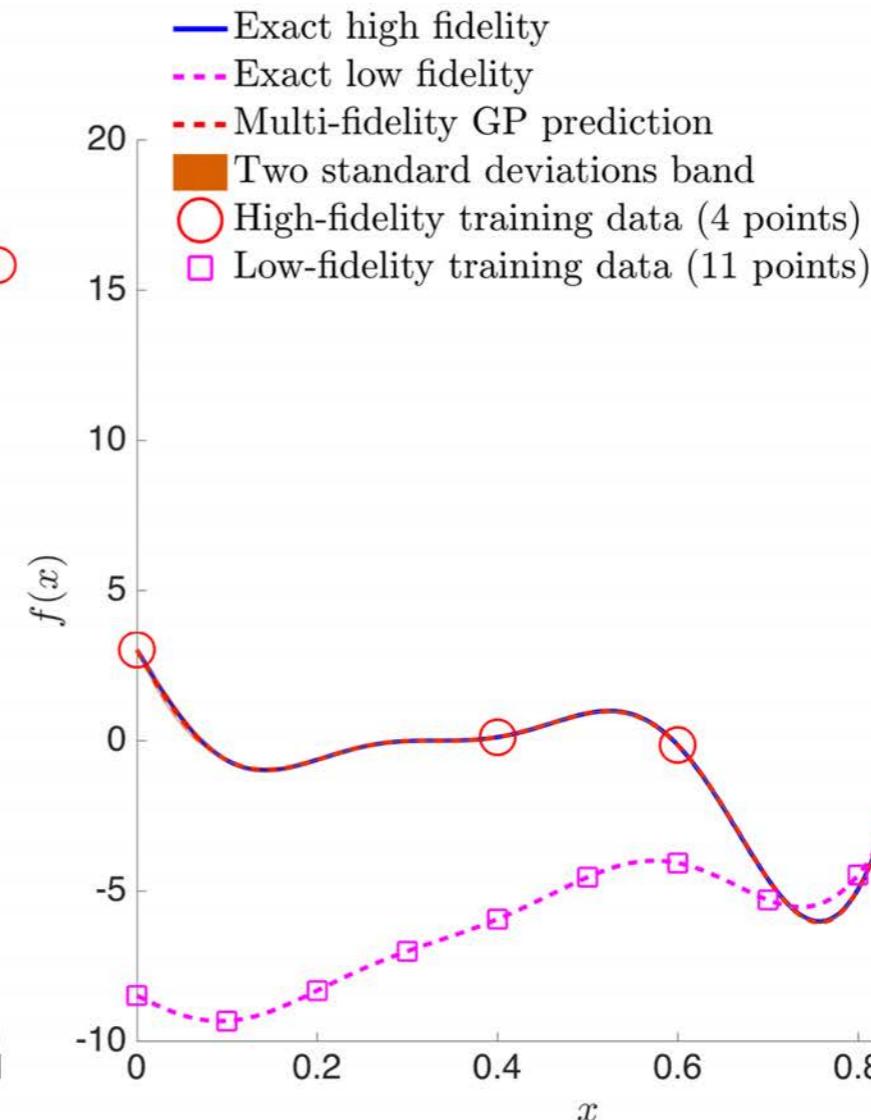
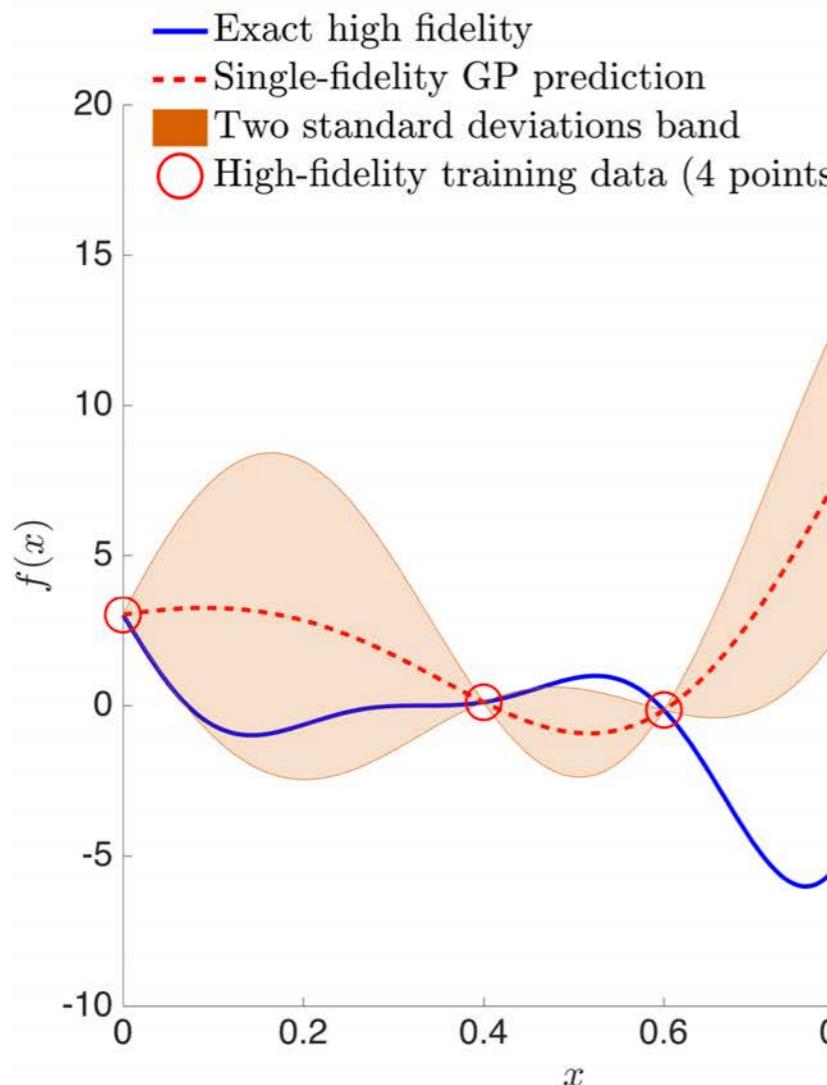
Multi-fidelity modeling



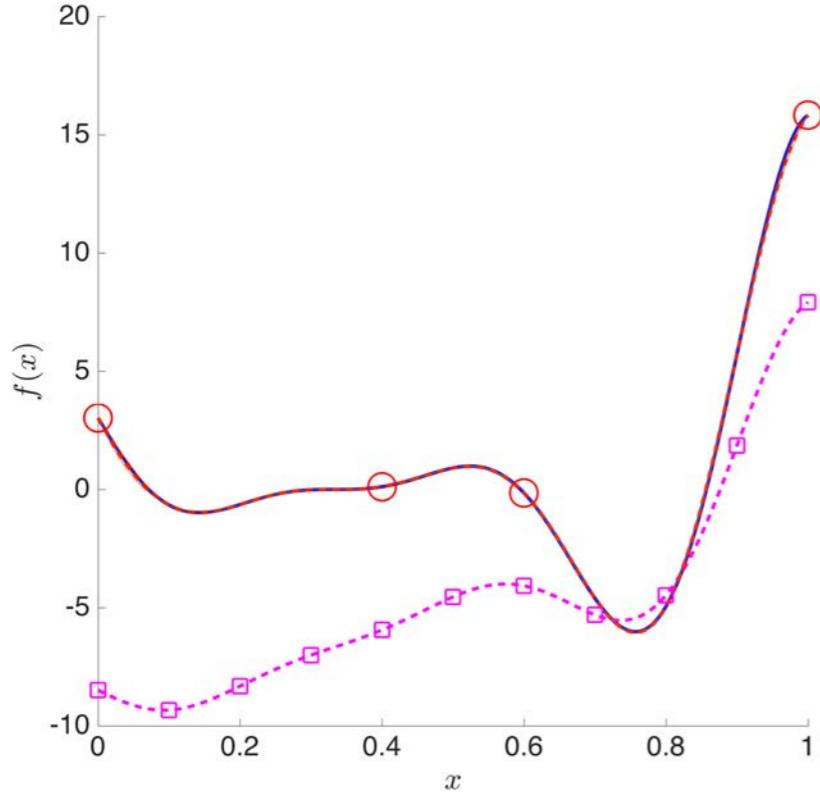
Number of runs is limited by time
and computational resources

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



Multi-fidelity observations:

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

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

Probabilistic model:

$$f_H(\mathbf{x}) = \rho f_L(\mathbf{x}) + \delta(\mathbf{x})$$

$$f_L(\mathbf{x}) \sim \mathcal{GP}(0, k_L(\mathbf{x}, \mathbf{x}'; \theta_L))$$

$$\delta(\mathbf{x}) \sim \mathcal{GP}(0, k_H(\mathbf{x}, \mathbf{x}'; \theta_H))$$

$$\boldsymbol{\epsilon}_L \sim \mathcal{N}(0, \sigma_{\epsilon_L}^2 \mathbf{I})$$

$$\boldsymbol{\epsilon}_H \sim \mathcal{N}(0, \sigma_{\epsilon_H}^2 \mathbf{I})$$

Training:

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

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

Prediction:

$$p(f(\mathbf{x}^*) | \mathbf{y}, \mathbf{X}, \mathbf{x}^*) \sim \mathcal{N}(f(\mathbf{x}^*) | \mu(\mathbf{x}^*), \sigma^2(\mathbf{x}^*))$$

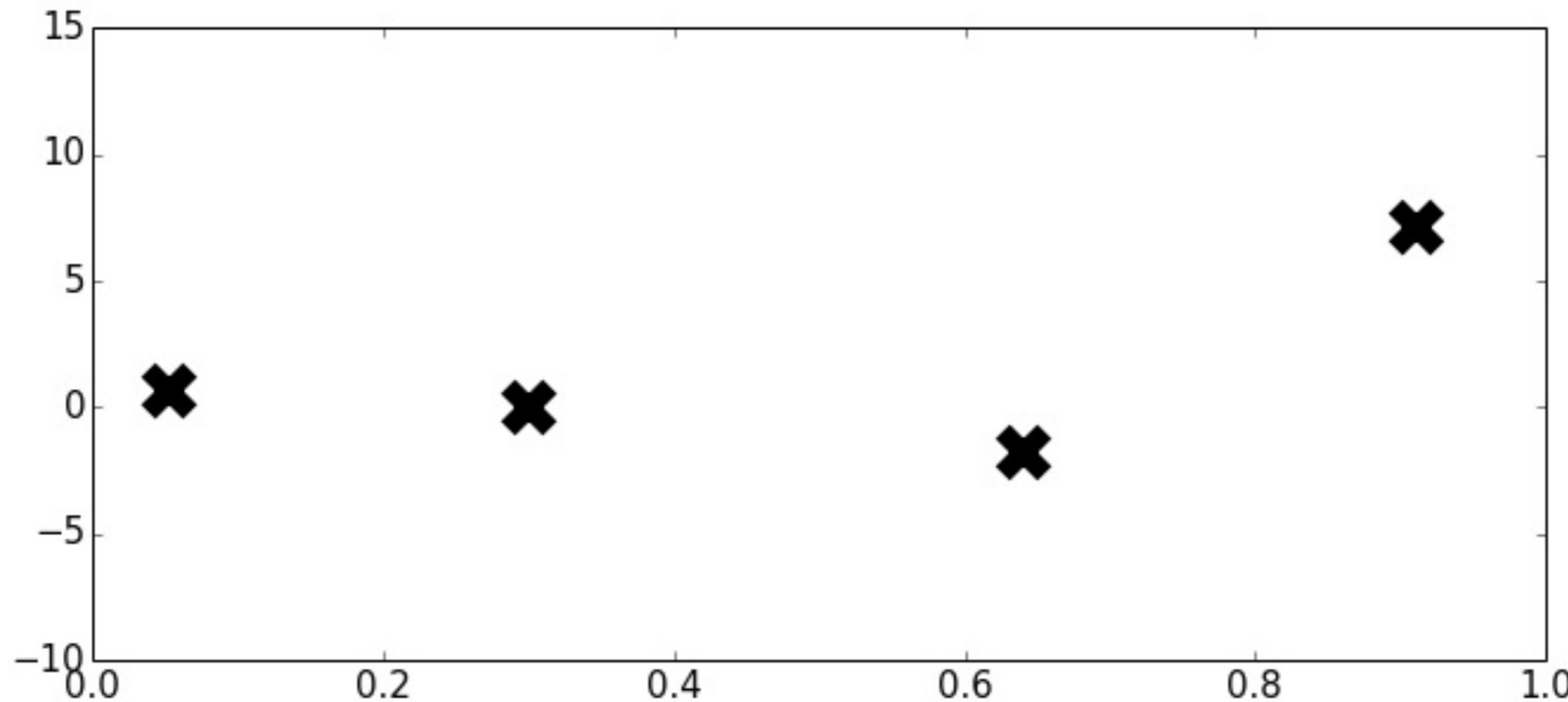
$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*, \mathbf{X}) \mathbf{K}^{-1} \mathbf{y}$$

$$\sigma(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, \mathbf{X}) \mathbf{K}^{-1} \mathbf{k}(\mathbf{X}, \mathbf{x}^*)$$

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

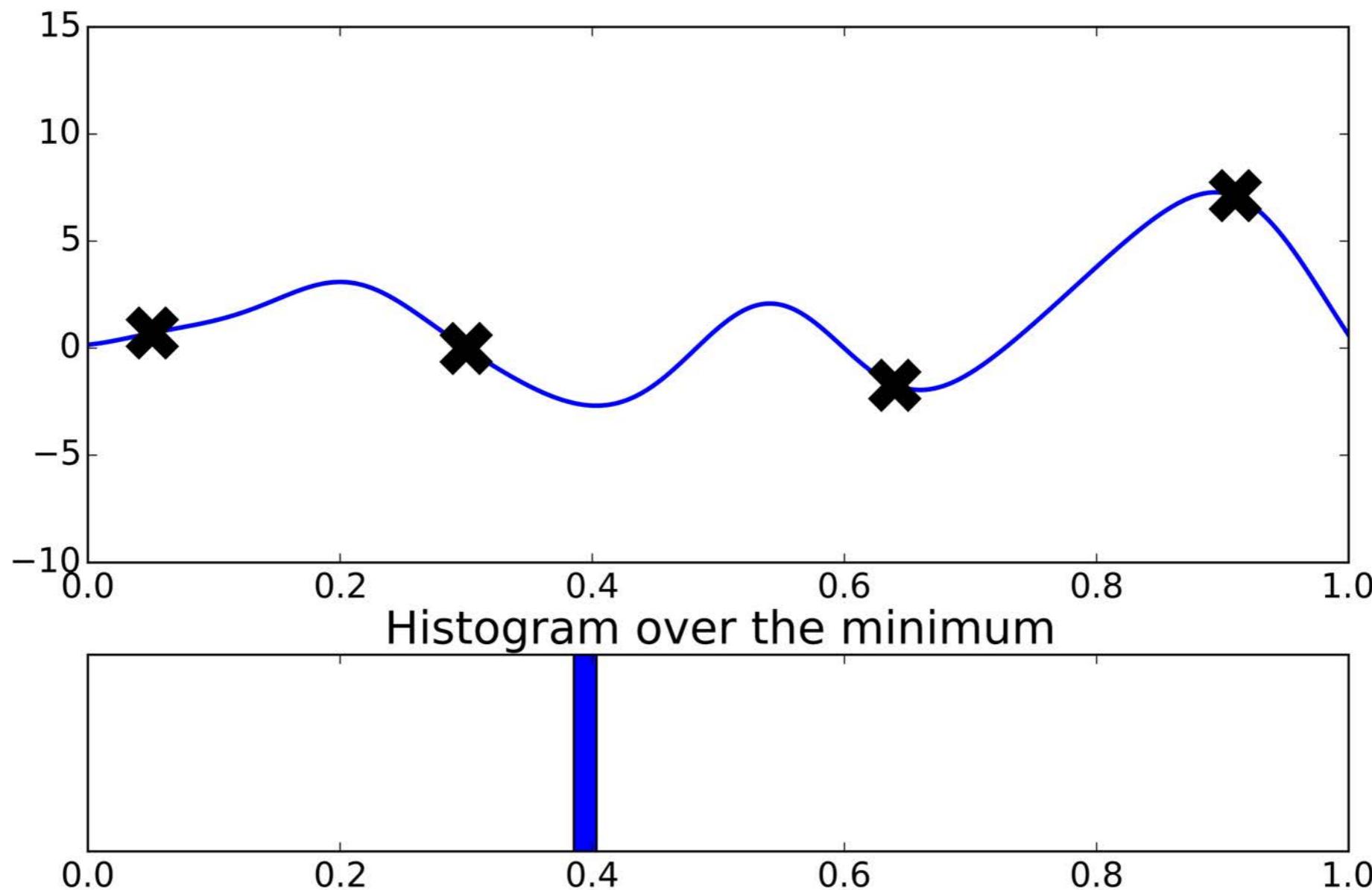
Demo code: <https://github.com/PredictiveIntelligenceLab/GPTutorial>

Bayesian optimization

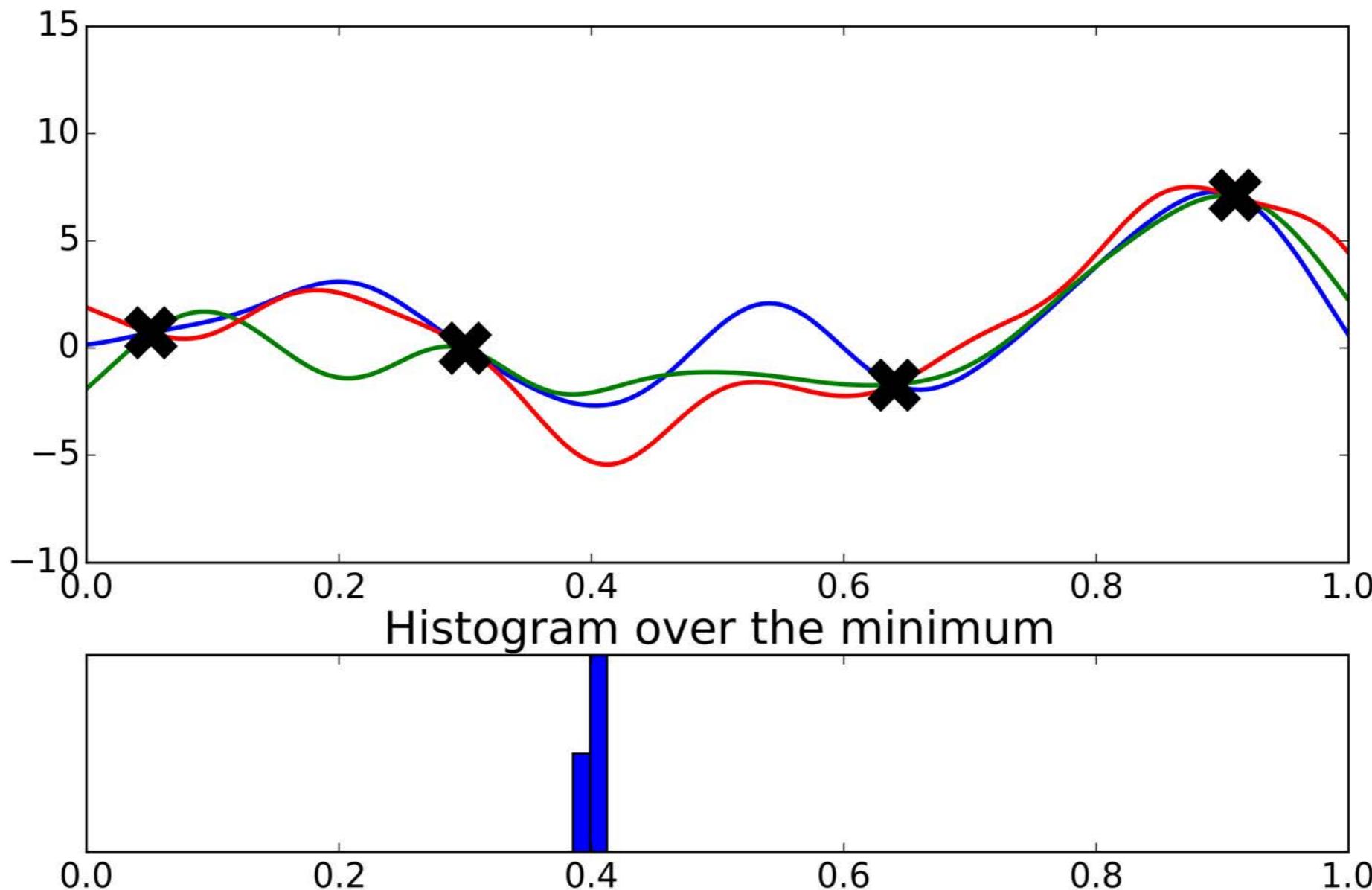


Where is the minimum of f ?
Where should we take the next evaluation?

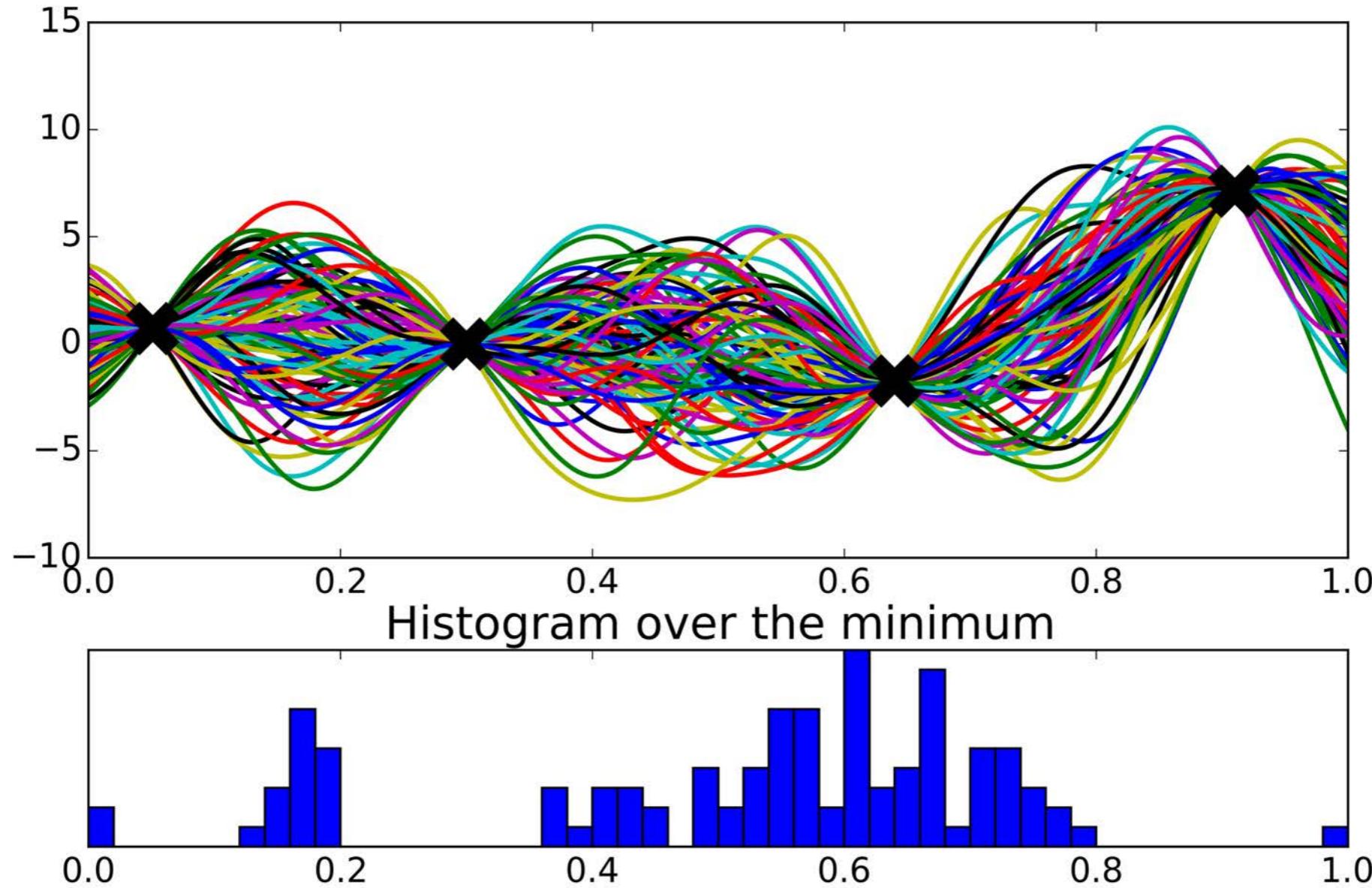
Bayesian optimization



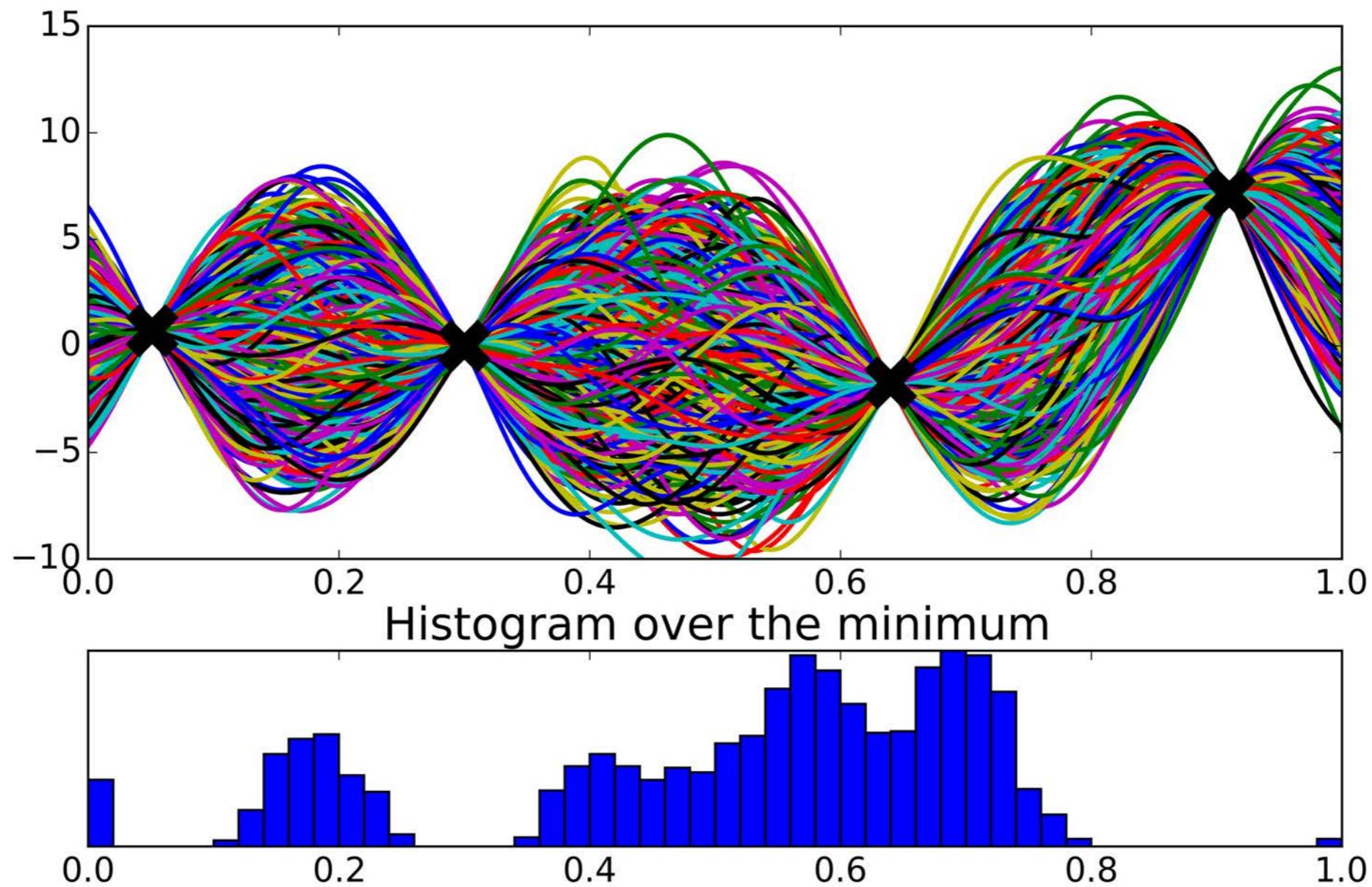
Bayesian optimization



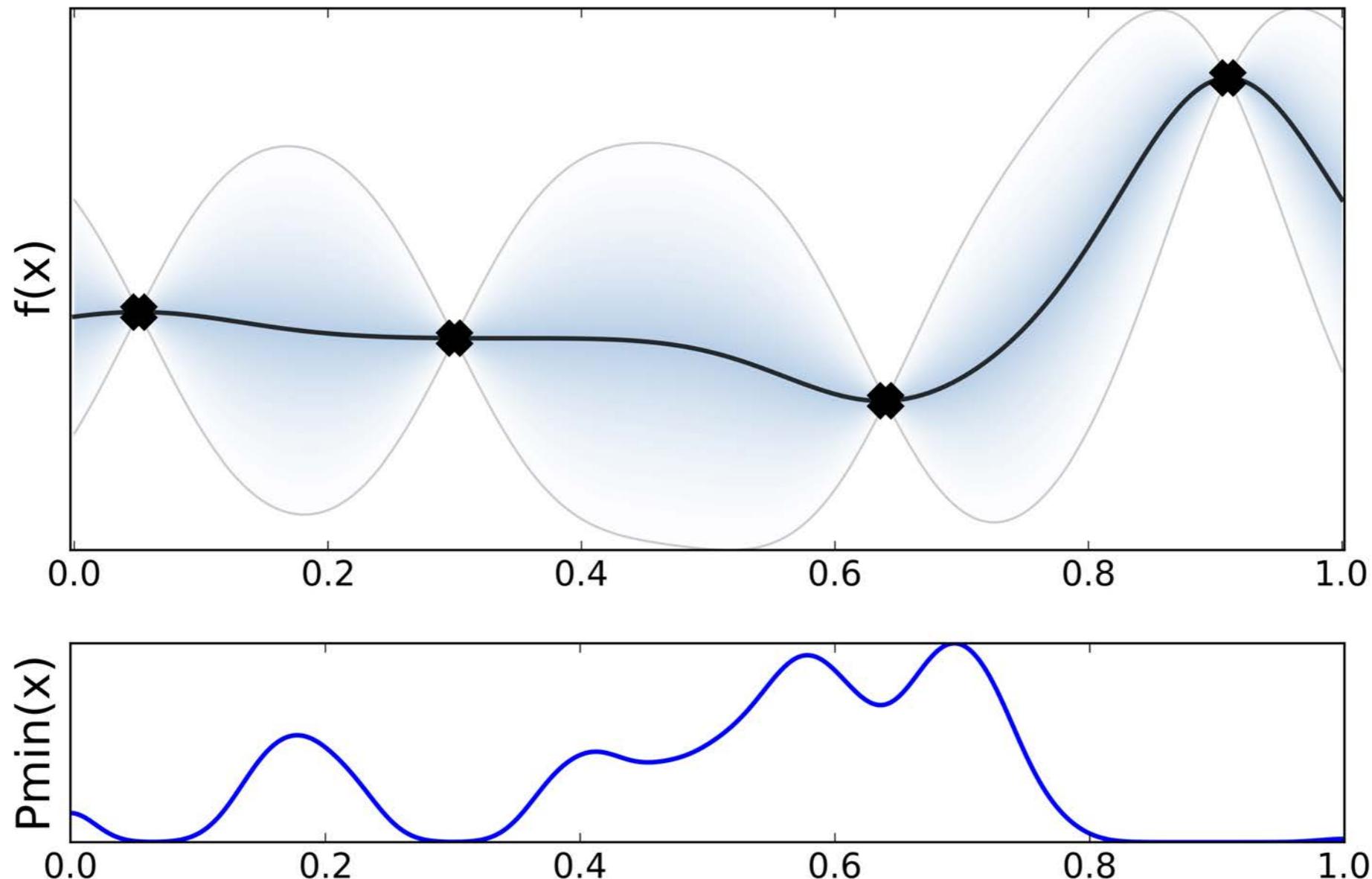
Bayesian optimization



Bayesian optimization



Bayesian optimization



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

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

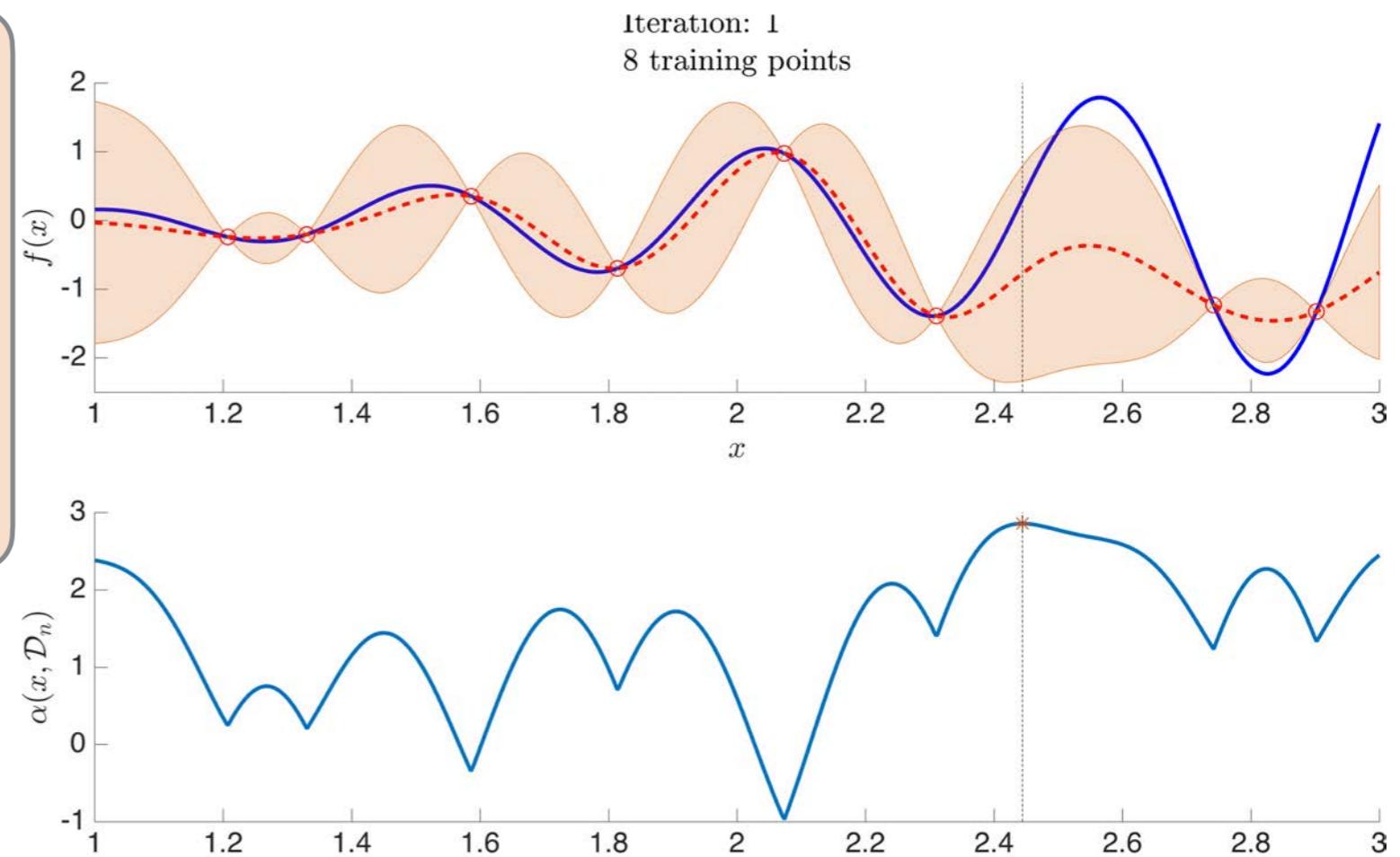
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v | \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

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

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

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: $g(\mathbf{x})$ 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

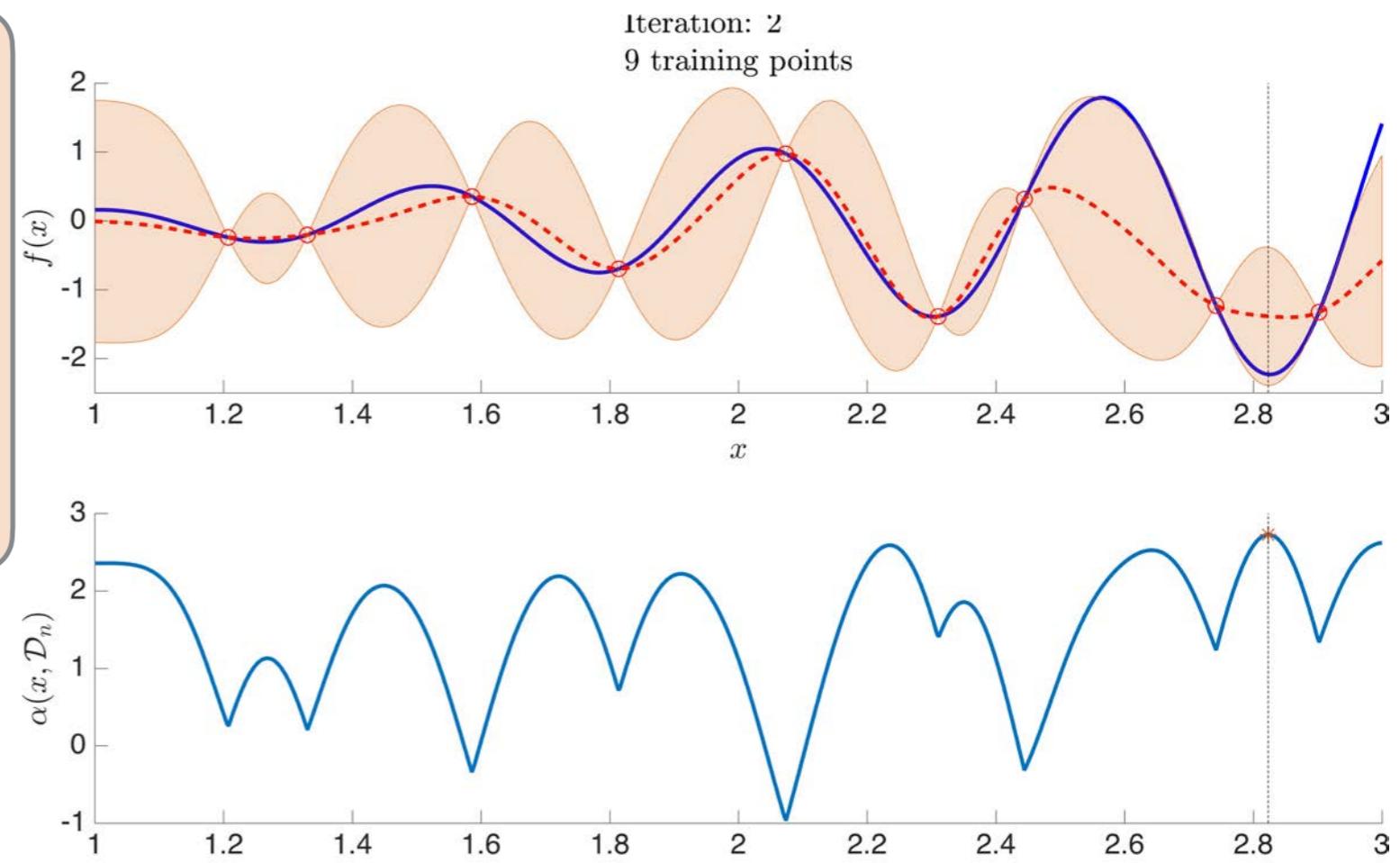
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v | \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

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

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

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: $g(\mathbf{x})$ 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

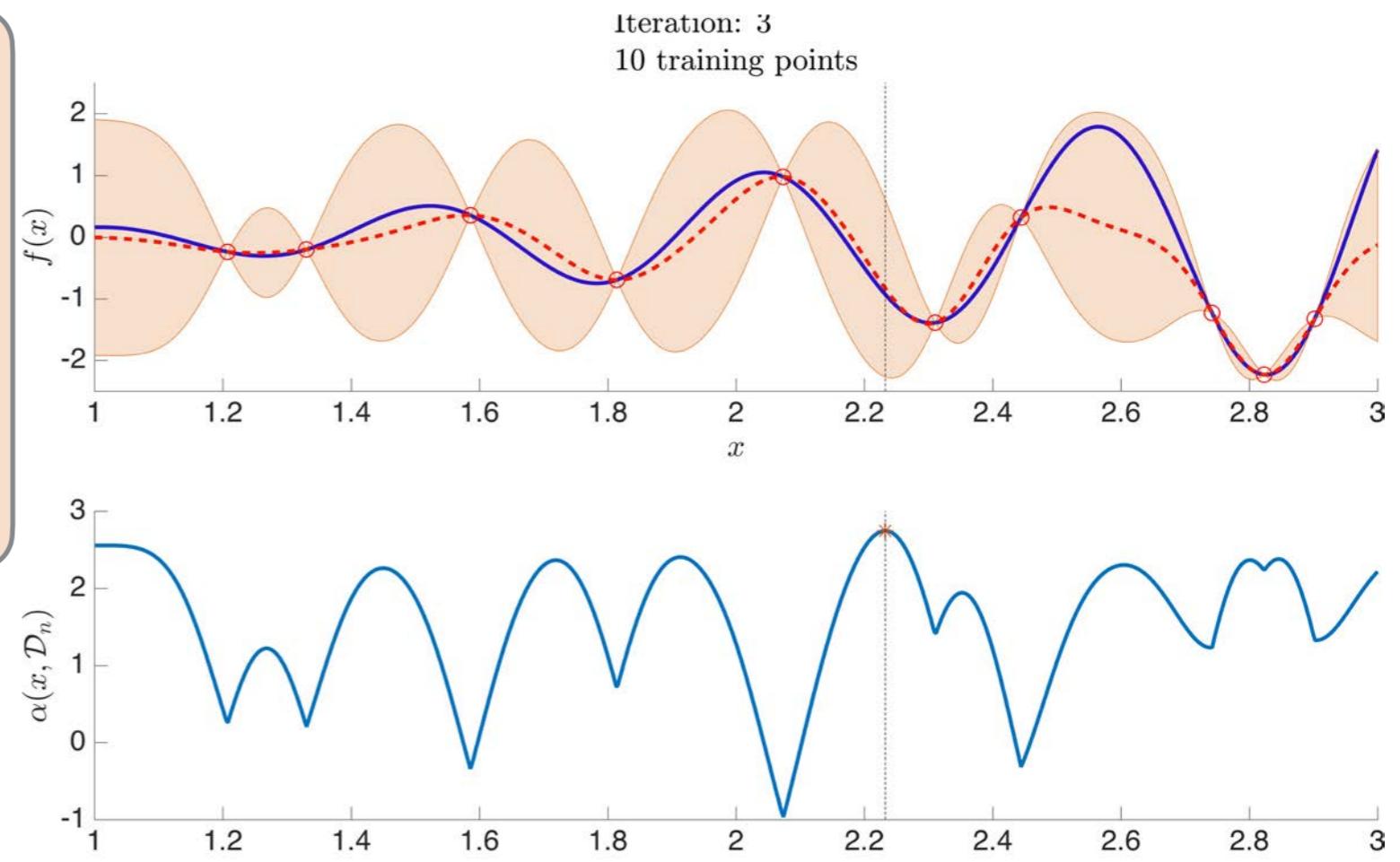
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v | \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

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

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

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: $g(\mathbf{x})$ 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

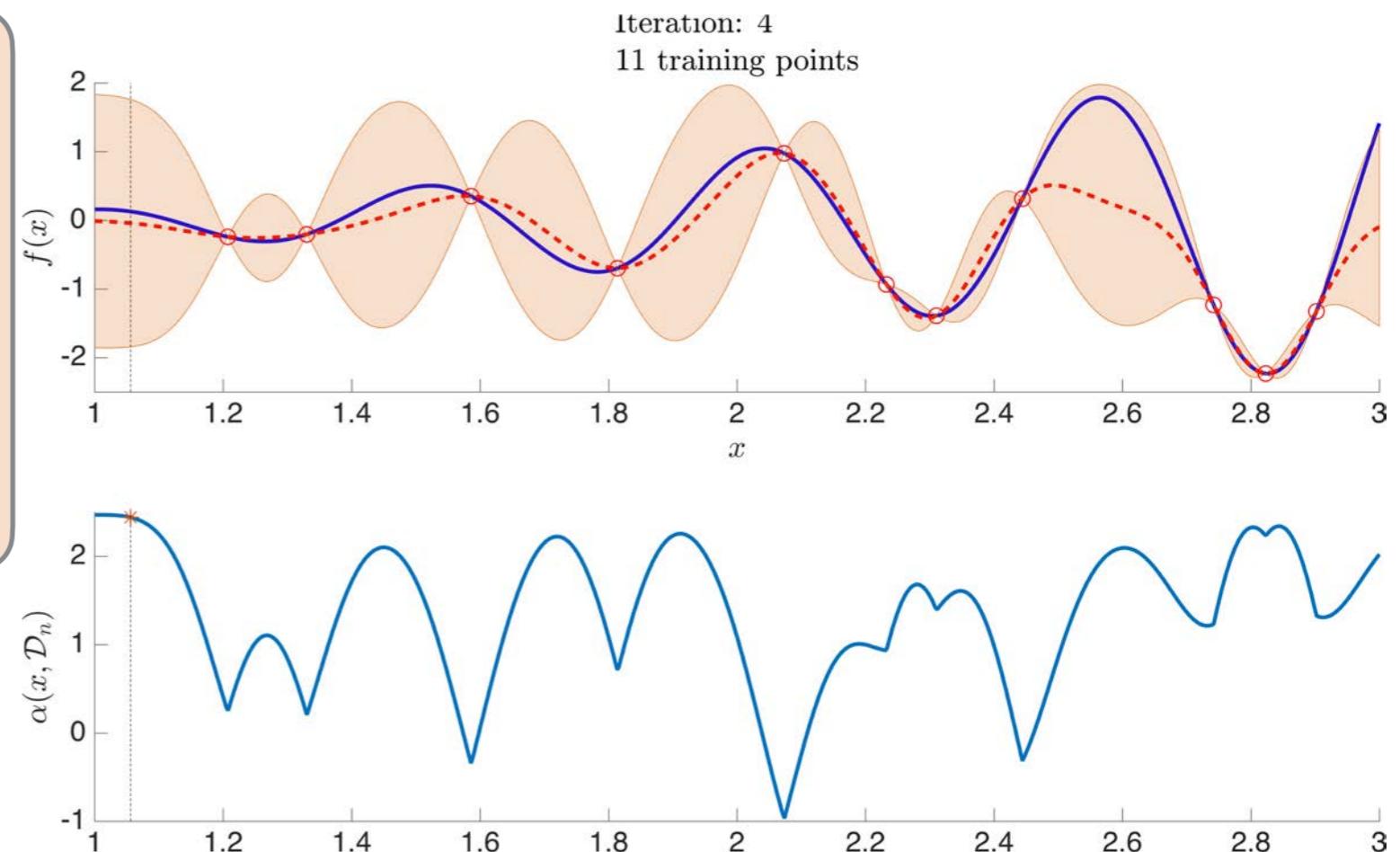
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v | \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

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

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

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: $g(\mathbf{x})$ 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

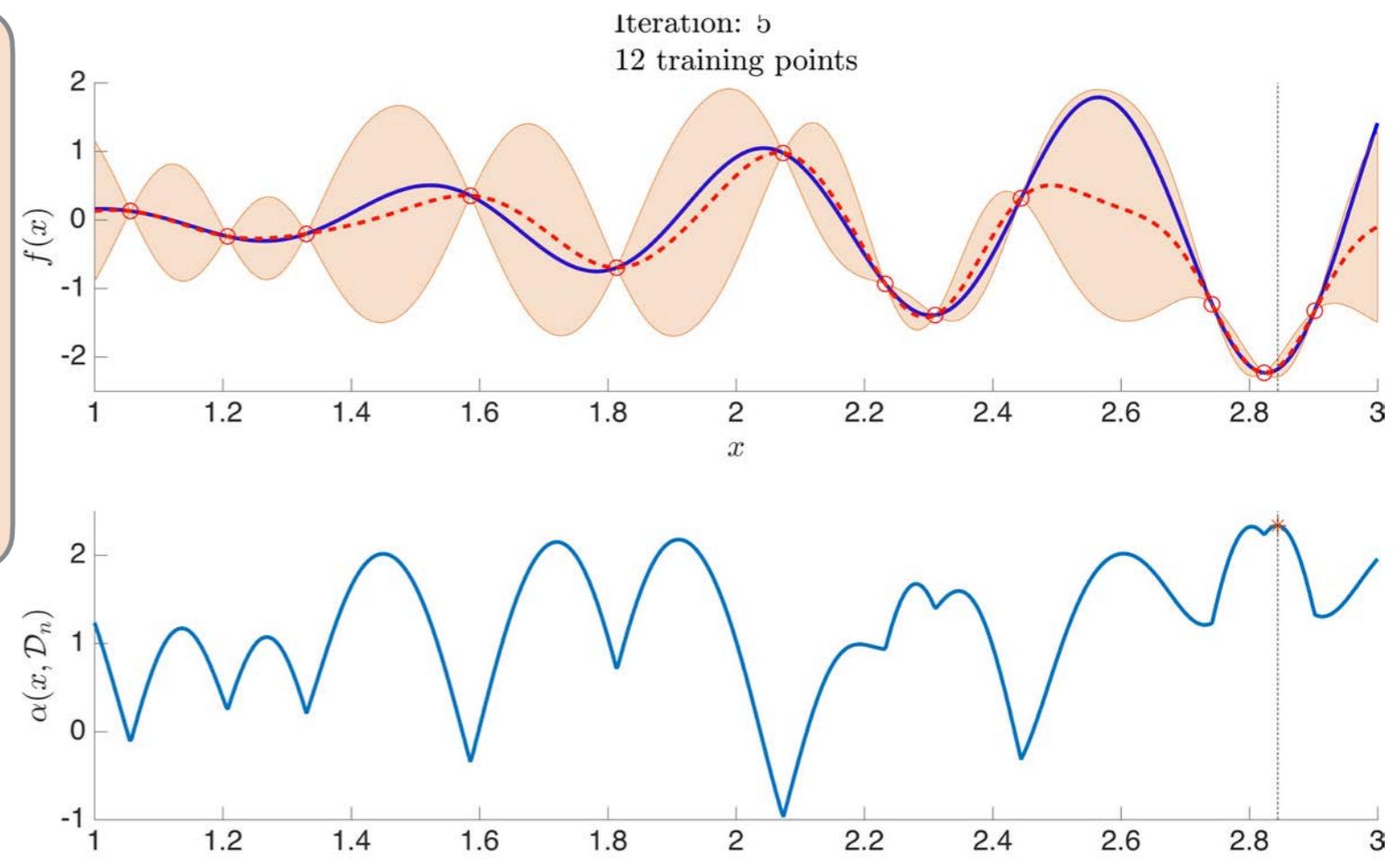
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v | \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

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

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

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: $g(\mathbf{x})$ 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

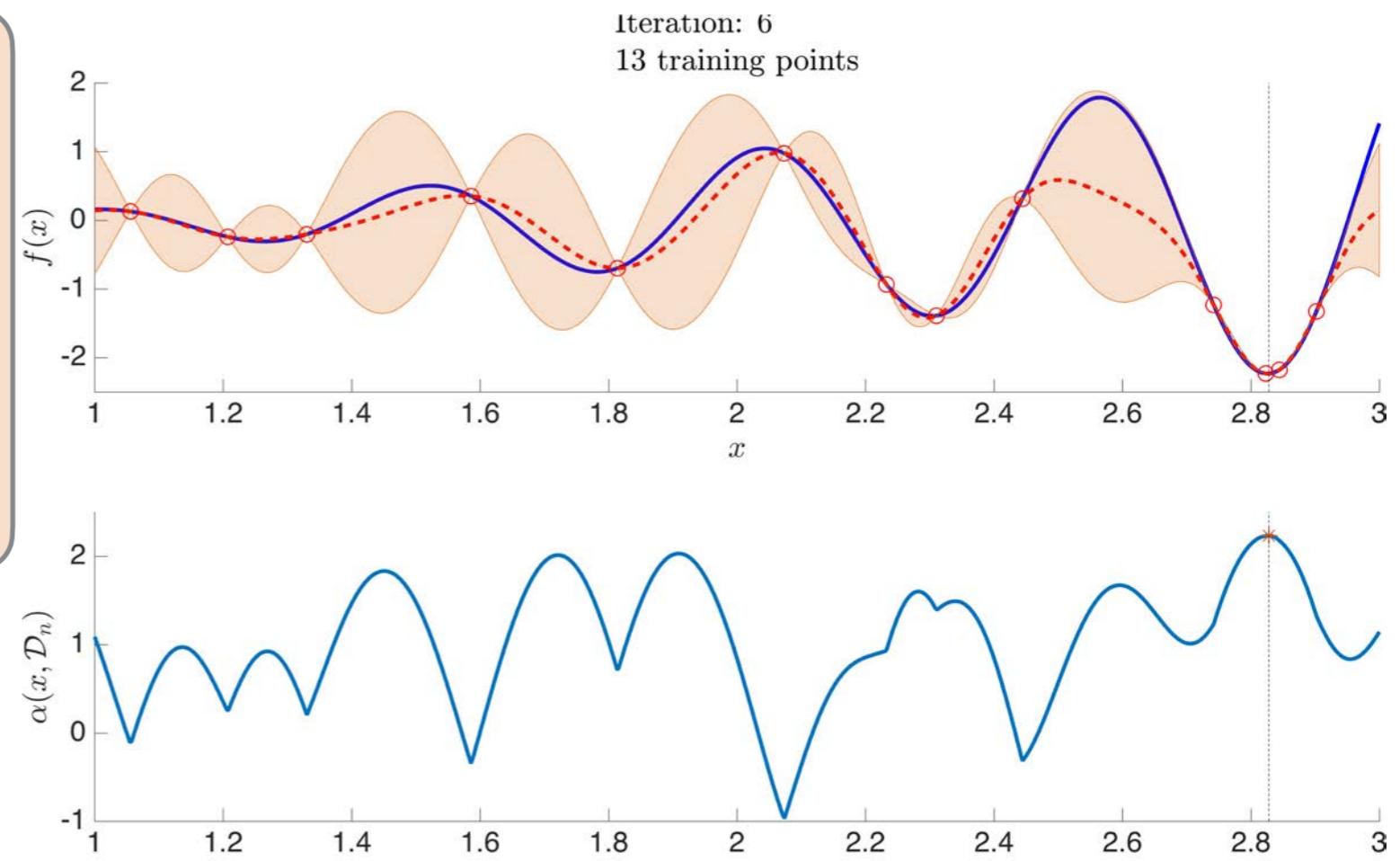
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v | \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

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

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

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: $g(\mathbf{x})$ 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

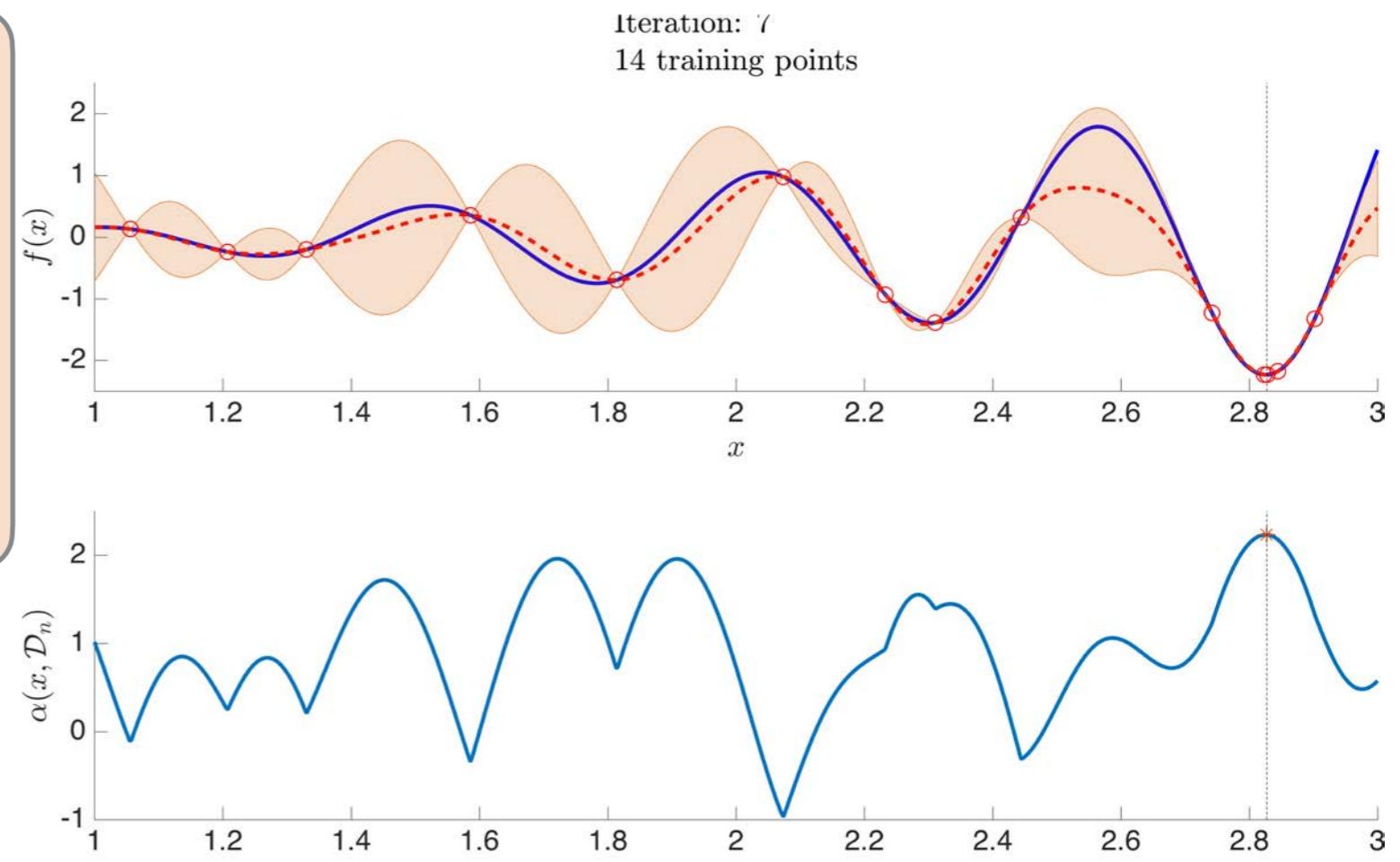
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{v | \mathbf{x}, \theta} [U(\mathbf{x}, v, \theta)]$$

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

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

Learning level sets

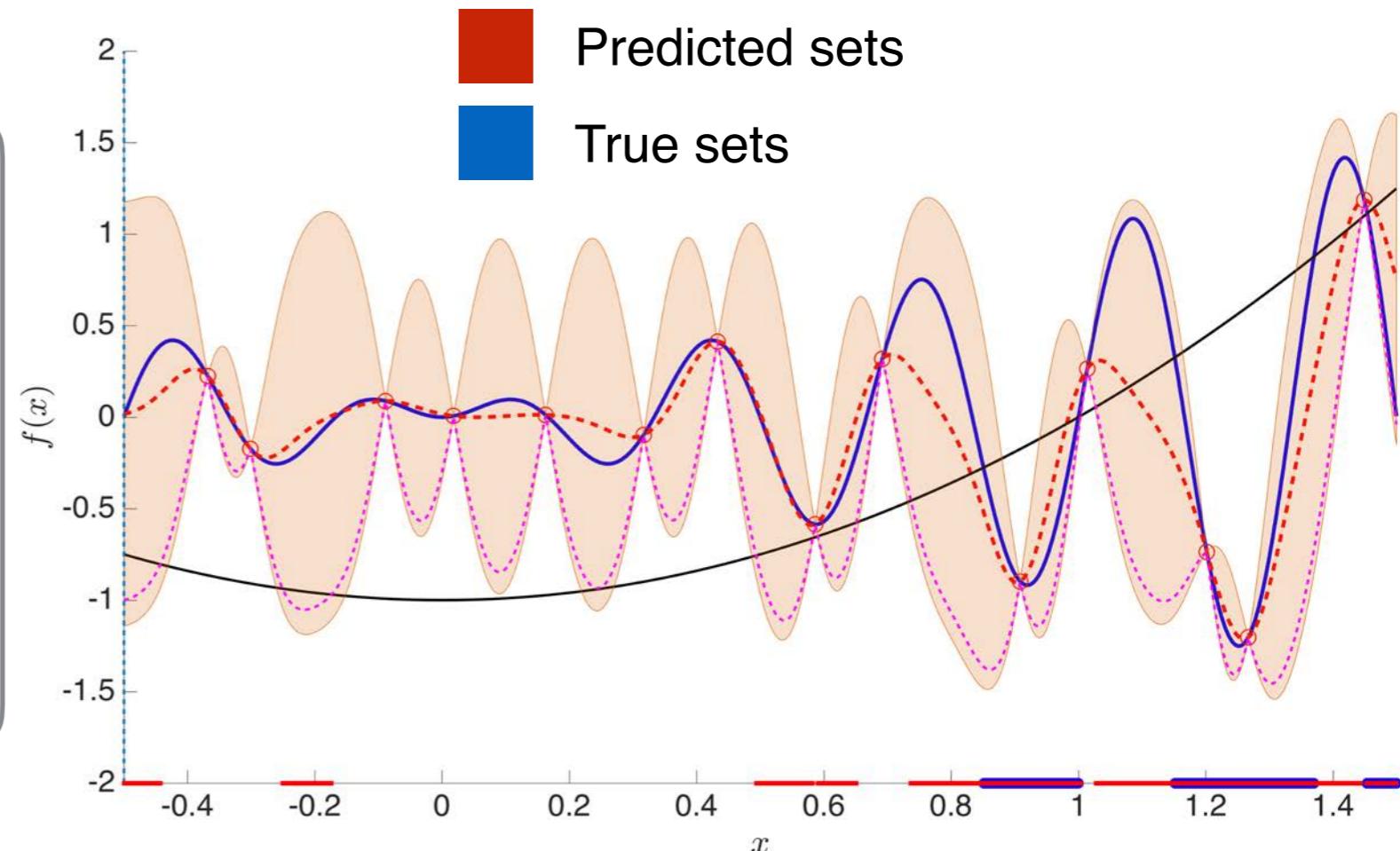
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

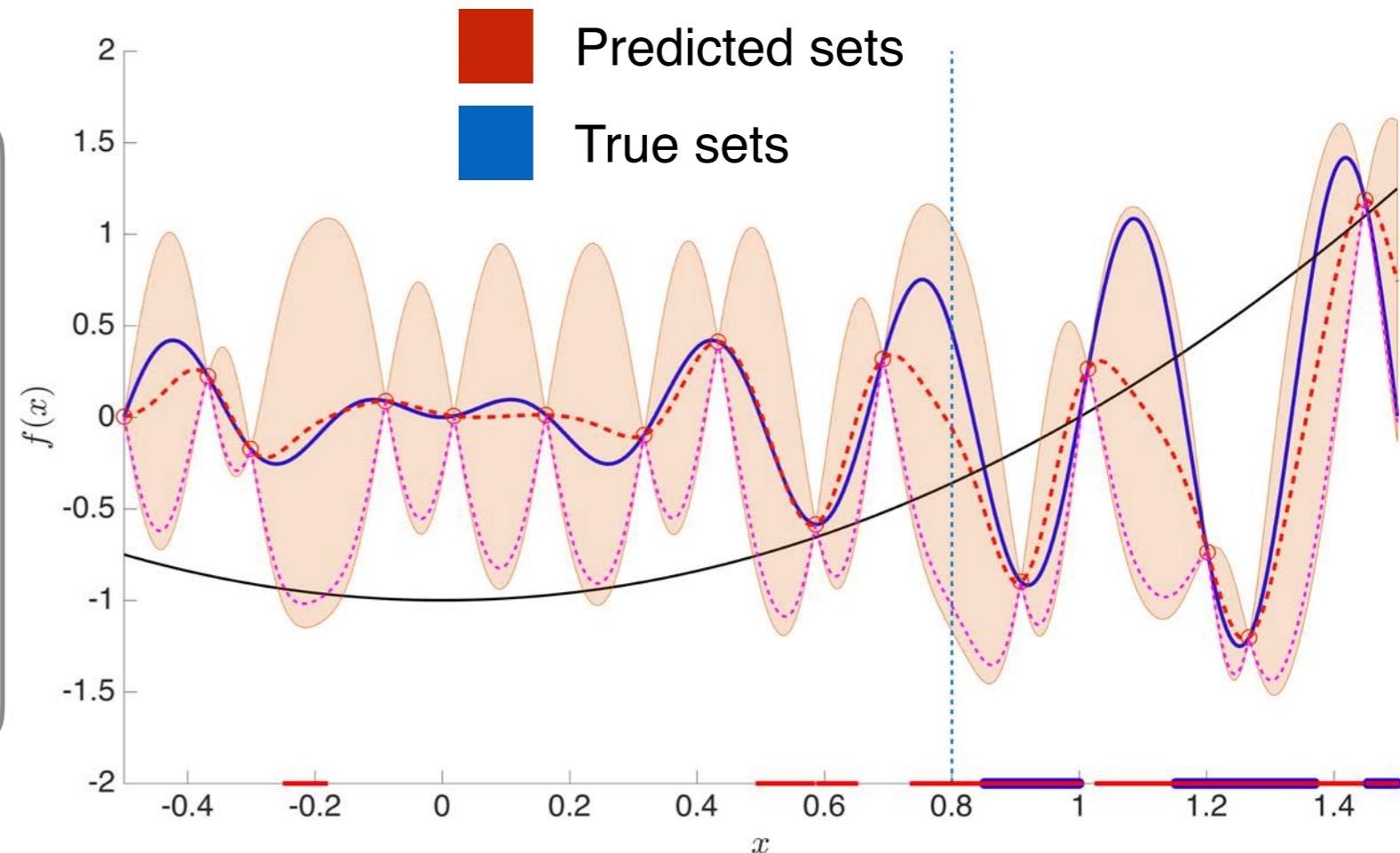
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

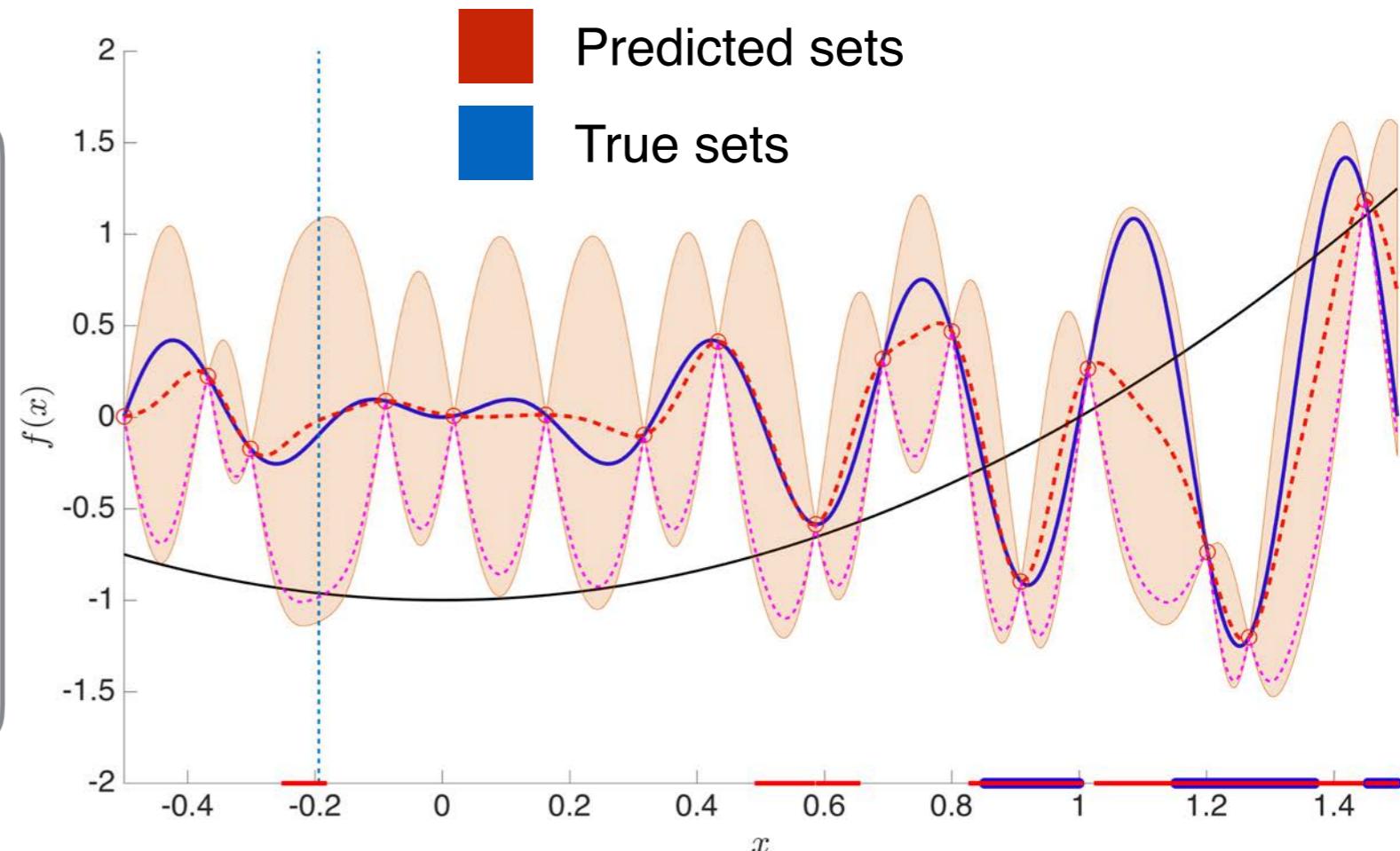
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

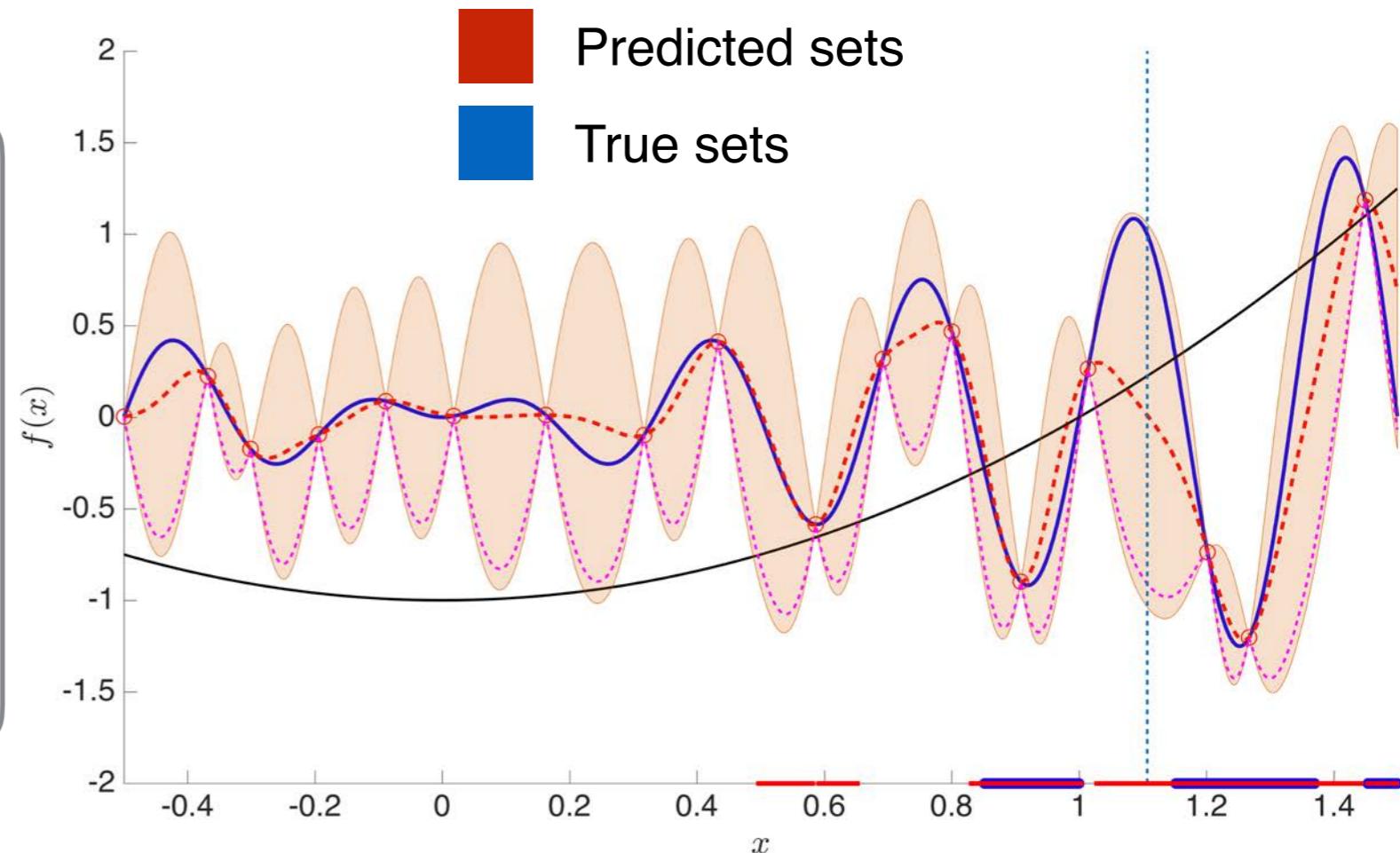
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

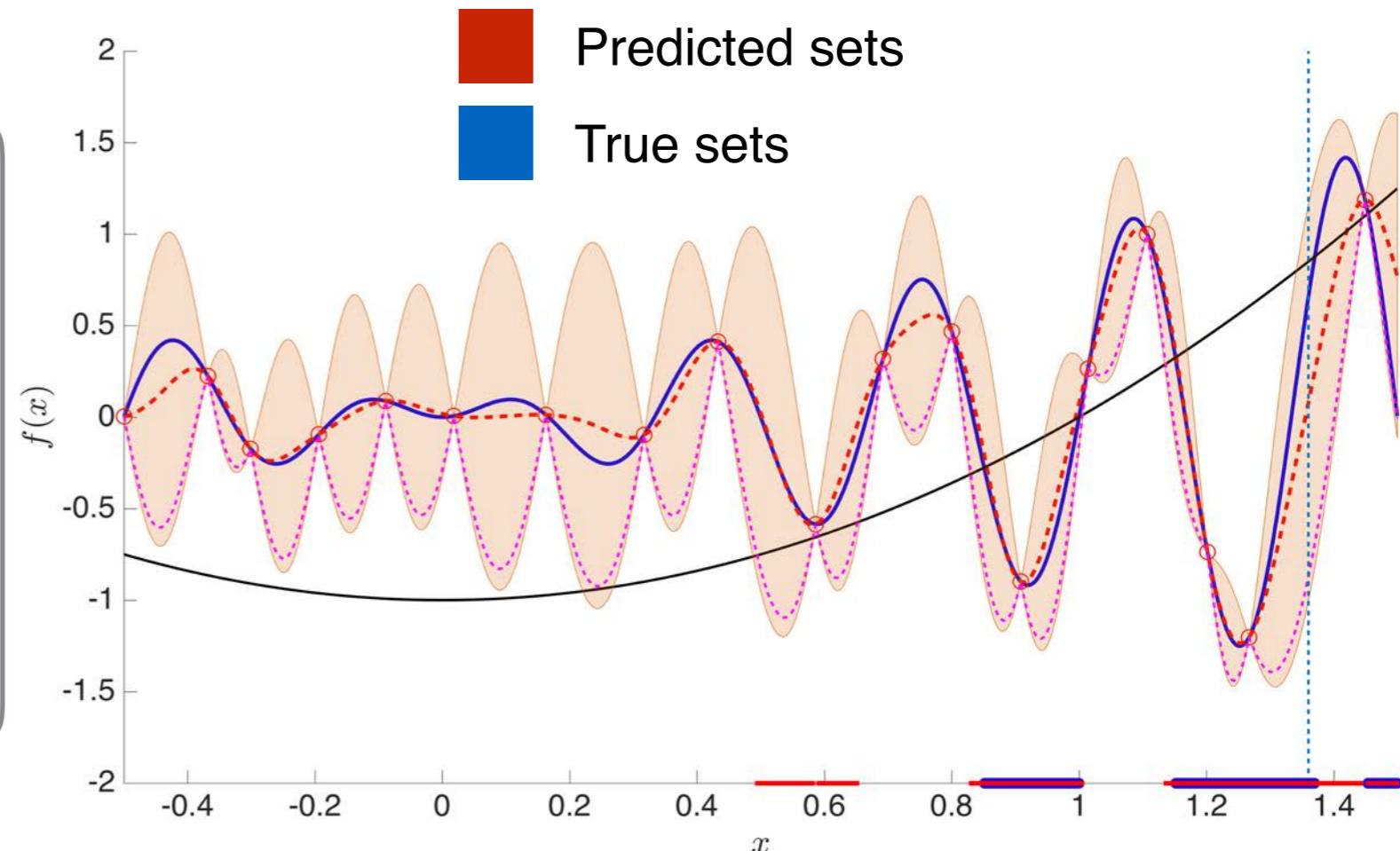
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

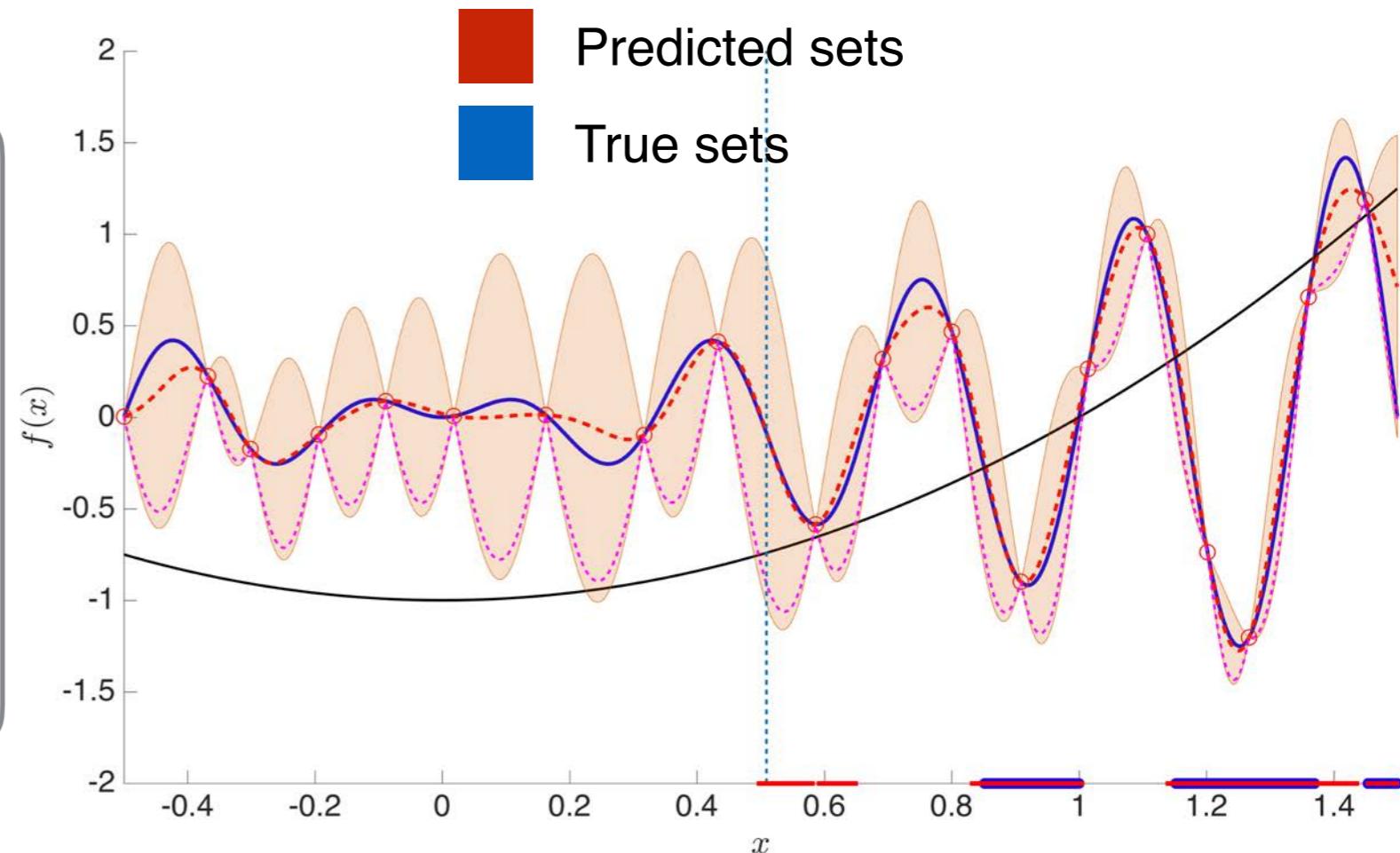
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

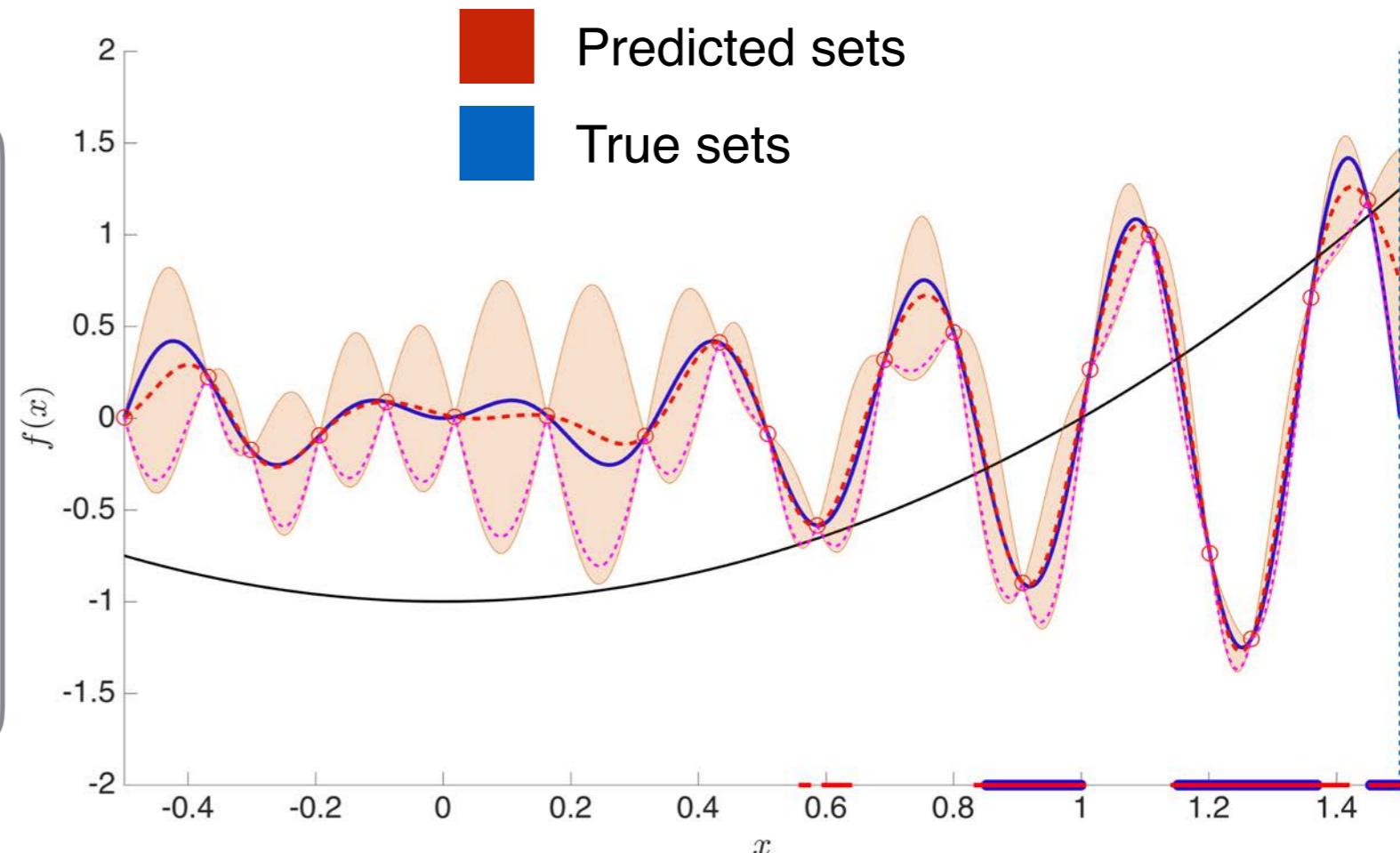
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

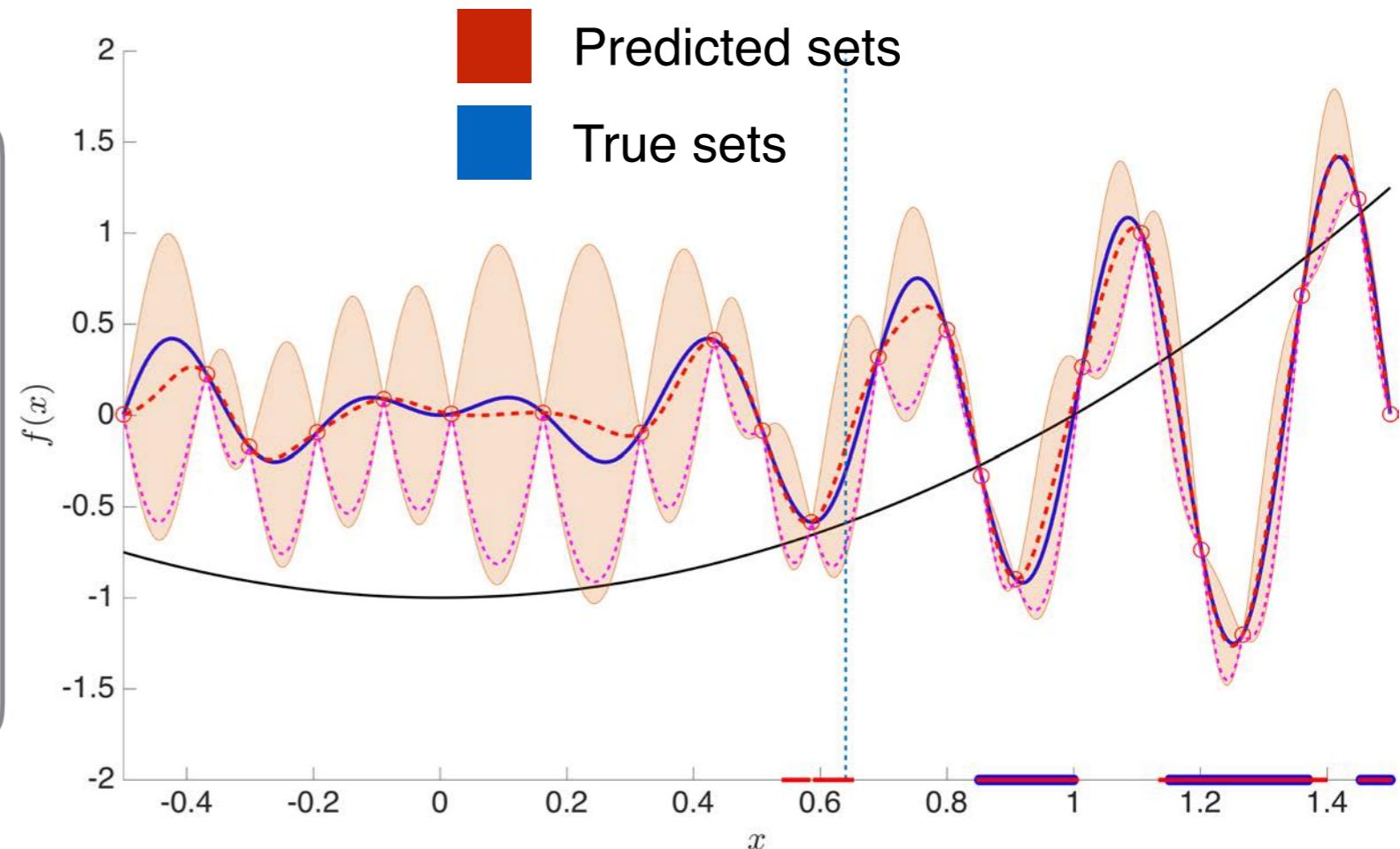
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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

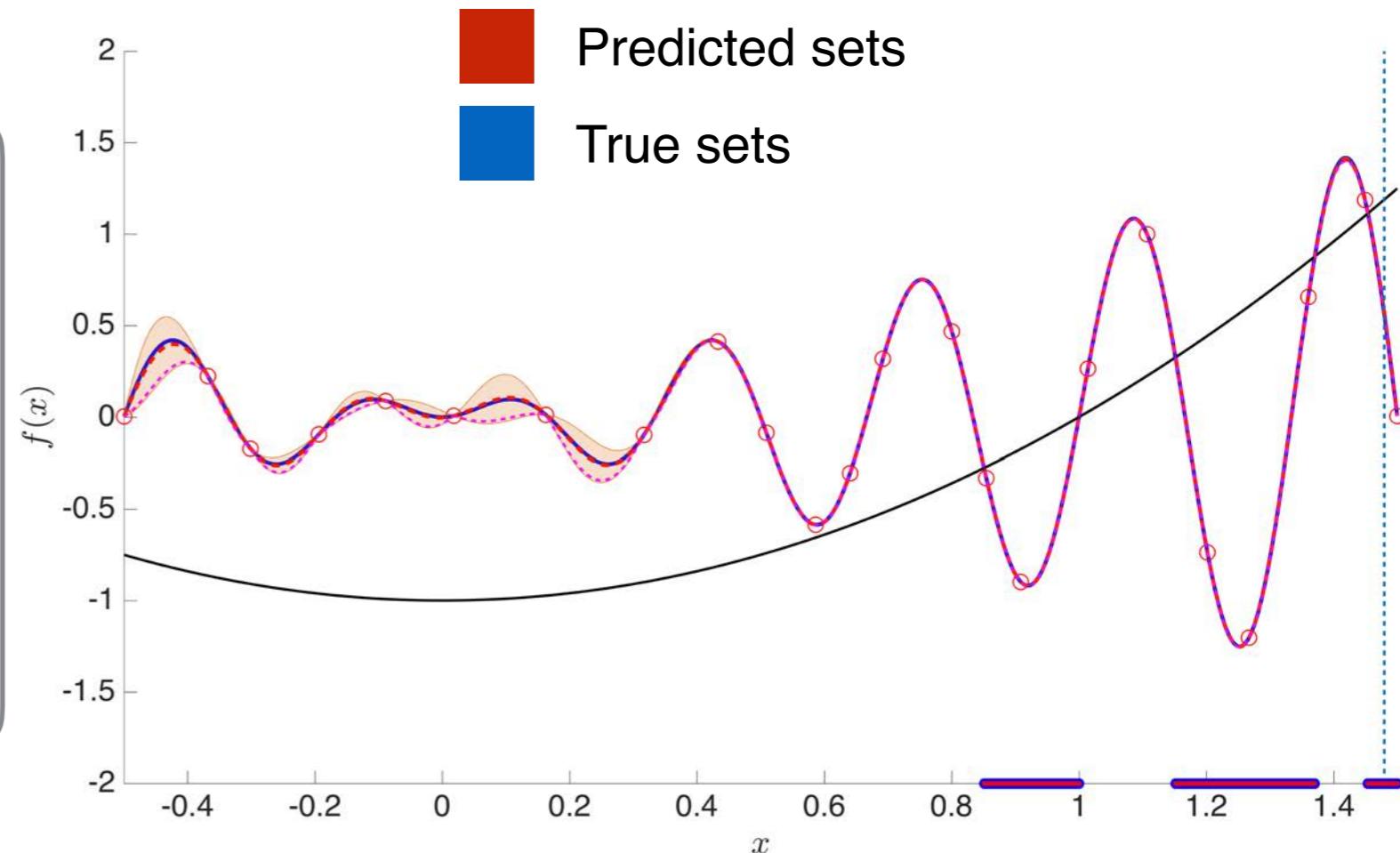
Goal: Identify the sets $L_\alpha(t) = \{\mathbf{x} : \mathcal{R}_\alpha(f(\mathbf{x})) \leq t(\mathbf{x})\}$

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 $L(t)$

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



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

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

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



Applications to control

Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics

Felix Berkenkamp*, Andreas Krause*, and Angela P. Schoellig†

*Learning & Adaptive Systems Group, Department of Computer Science, ETH Zurich, Switzerland

†Dynamic Systems Lab, Institute for Aerospace Studies, University of Toronto, Canada

Email: *{befelix, krausea}@ethz.ch, †schoellig@utias.utoronto.ca

Safe Controller Optimization for Quadrotors with Gaussian Processes

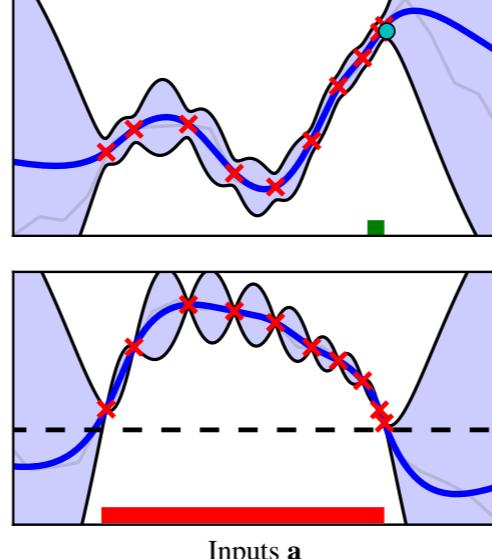
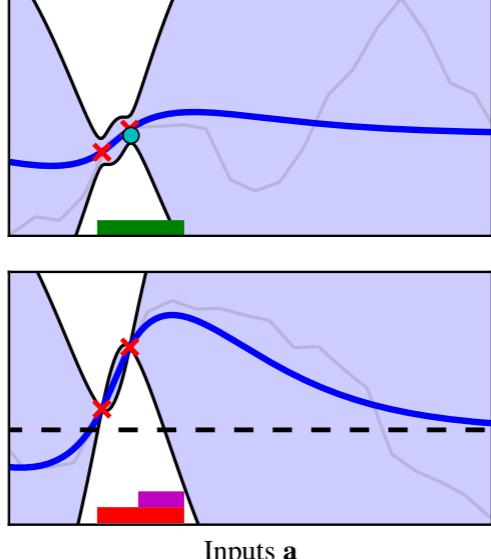
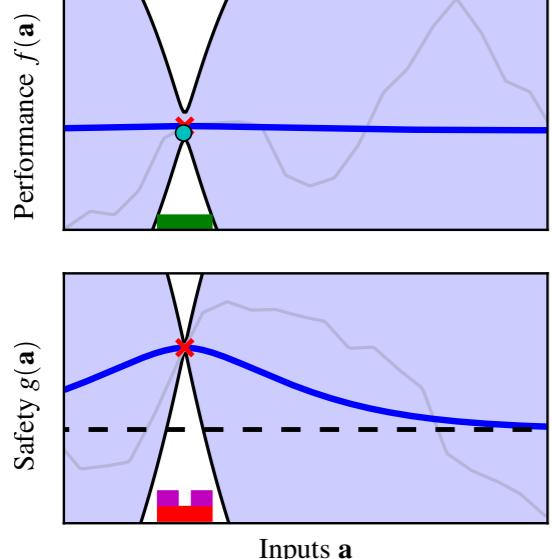
Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause

SAFE OPT-MC



Algorithm with Parameters \mathbf{a}

Evaluate on Real System
Performance: $f(\mathbf{a}_n)$
Safety Constraints: $g_i(\mathbf{a}_n)$



(d) Initial, safe parameters.

(e) Safe exploration.

(f) After 10 evaluations: safe maximum found.

