A Hierarchical Expected Improvement method for effective Bayesian optimization

C. F. Jeff Wu

School of Data Science
Chinese University of Hong Kong, Shenzhen

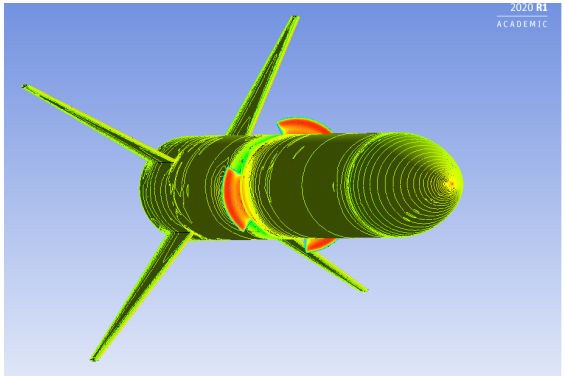
Joint work with **Zhehui Chen** (Google) and **Simon Mak** (Duke University)

[1]The paper appeared in JASA 2024.

Outline

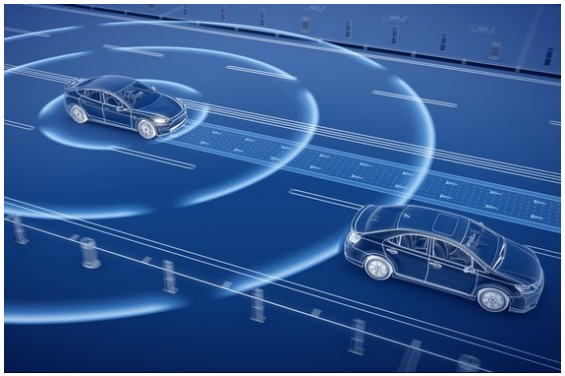
- Why **Bayesian optimization**?
- **Kriging** and **Expected Improvement (EI)**
- **Hierarchical Kriging** and **Hierarchical EI (HEI)**
- **Hyperprior specification** for HEI
- Global **convergence rates**
- Two **applications**: process optimization for **semiconductor manufacturing**, hyperparameter tuning for **deep learning**

Motivation

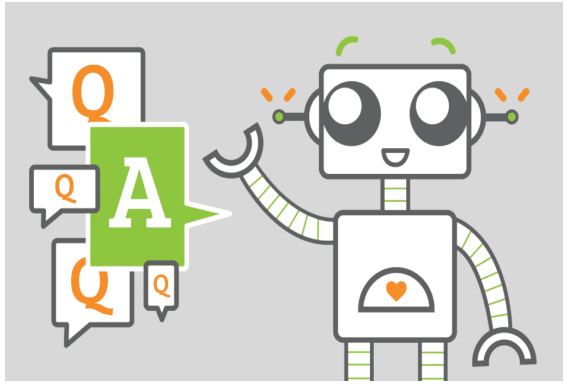


Engineering design

Portfolio management



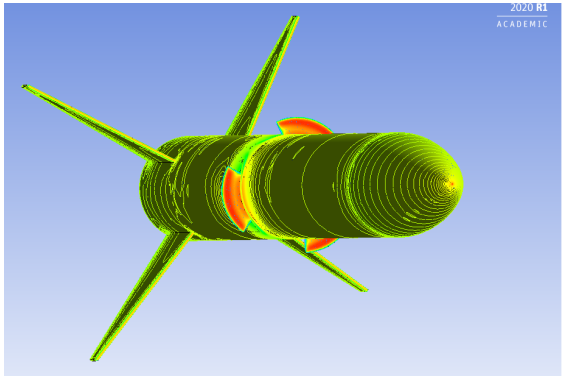
Autonomous vehicles



Q&A bot

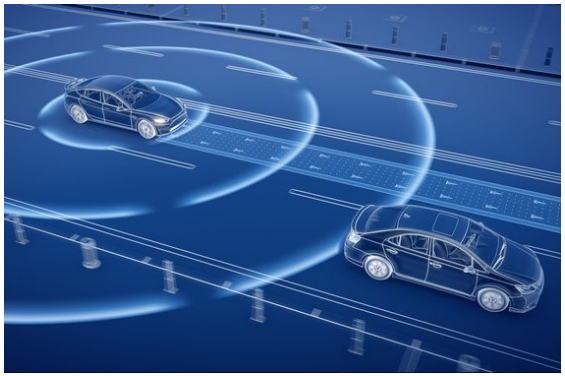
Want to find the **best** setting(s) under a **black-box** objective f'n

Motivation

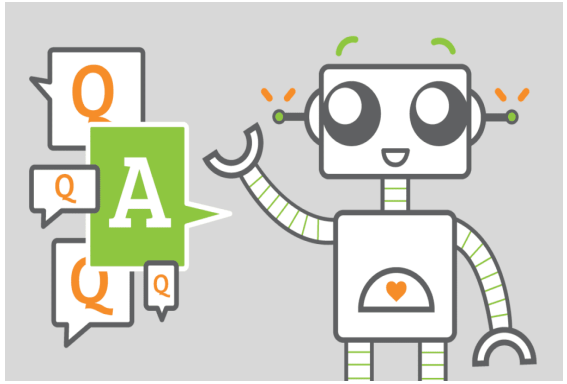


Engineering design

Portfolio management



Autonomous vehicles

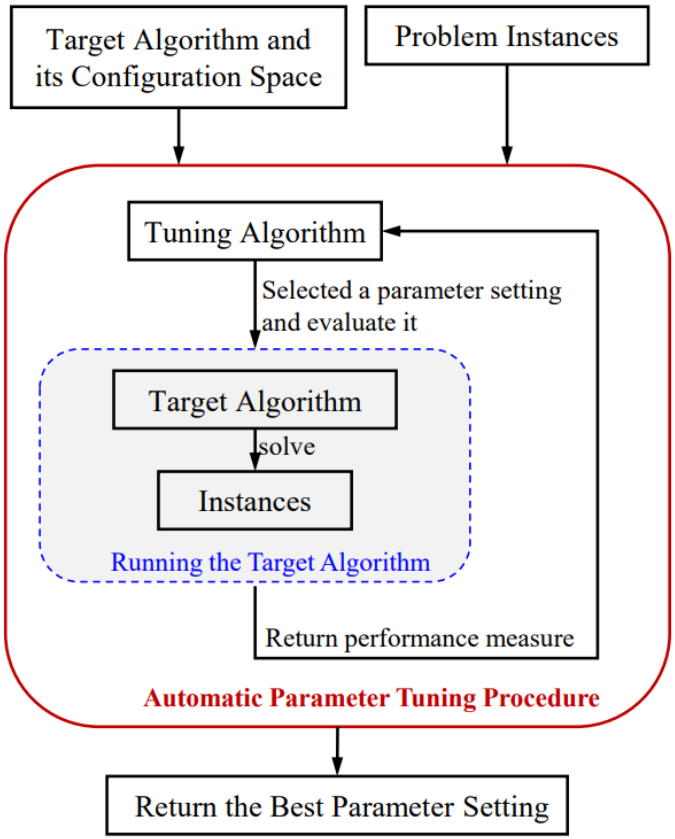


Q&A bot

Want to find the **best** setting(s) under a **black-box** objective f'n

Application in AI

Black-box optimization can be used to tune **hyper-parameters** automatically.

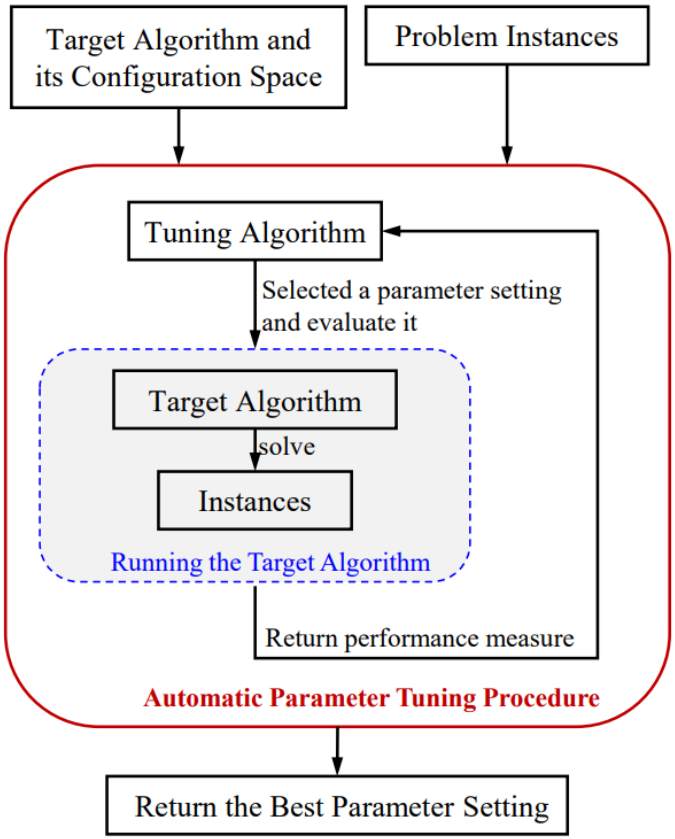


Examples of hyper-parameters in neural networks

- learning rate: $\log U(0.001, 10)$;
- n.layers: 1 to 3;
- batch size: 20 or 100;
- n.hidden units: $\log U(128, 4096)$;
- ...

Application in AI

Black-box optimization can be used to tune **hyper-parameters** automatically.



Examples of hyper-parameters in neural networks

- learning rate: $\log U(0.001, 10)$;
- n.layers: 1 to 3;
- batch size: 20 or 100;
- n.hidden units: $\log U(128, 4096)$;
  ...

Black-box optimization

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x})$$

- Ω: **feasible** domain for parameters $\boldsymbol{x}$
- f: an unknown **black-box objective** function
  - Functional form **unknown** prior to **data**
  - **Queries** on f require **experimentation** and/or **simulations**, can be **expensive**

Goal: find a **good** solution to the **optimization problem** using as **few queries** on f as possible

Black-box optimization

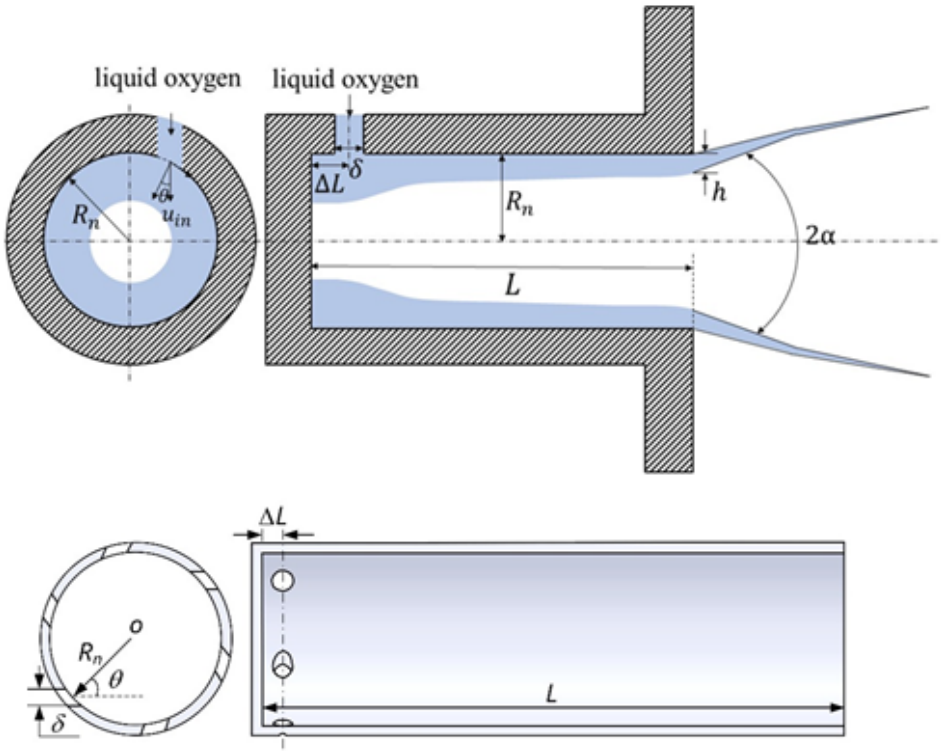
$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x})$$

- Ω: **feasible** domain for parameters $\boldsymbol{x}$
- f: an unknown **black-box objective** function
  - Functional form **unknown** prior to **data**
  - **Queries** on f require **experimentation** and/or **simulations**, can be **expensive**

Goal: find a **good** solution to the **optimization problem** using as **few queries** on f as possible

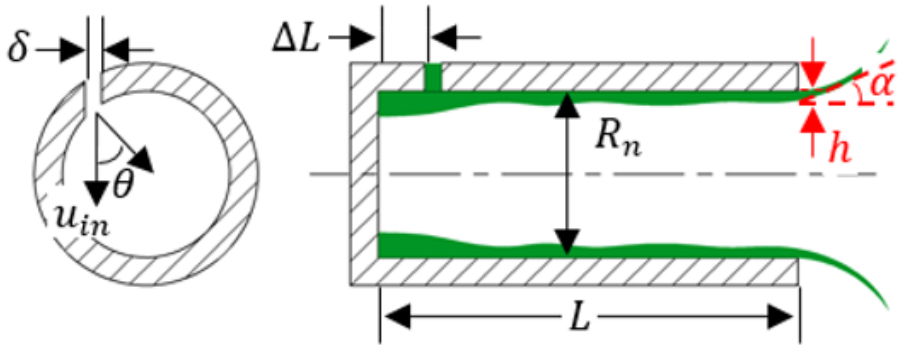
Rocket design

Collaboration project with GT Aerospace (Air Force / SpaceX)

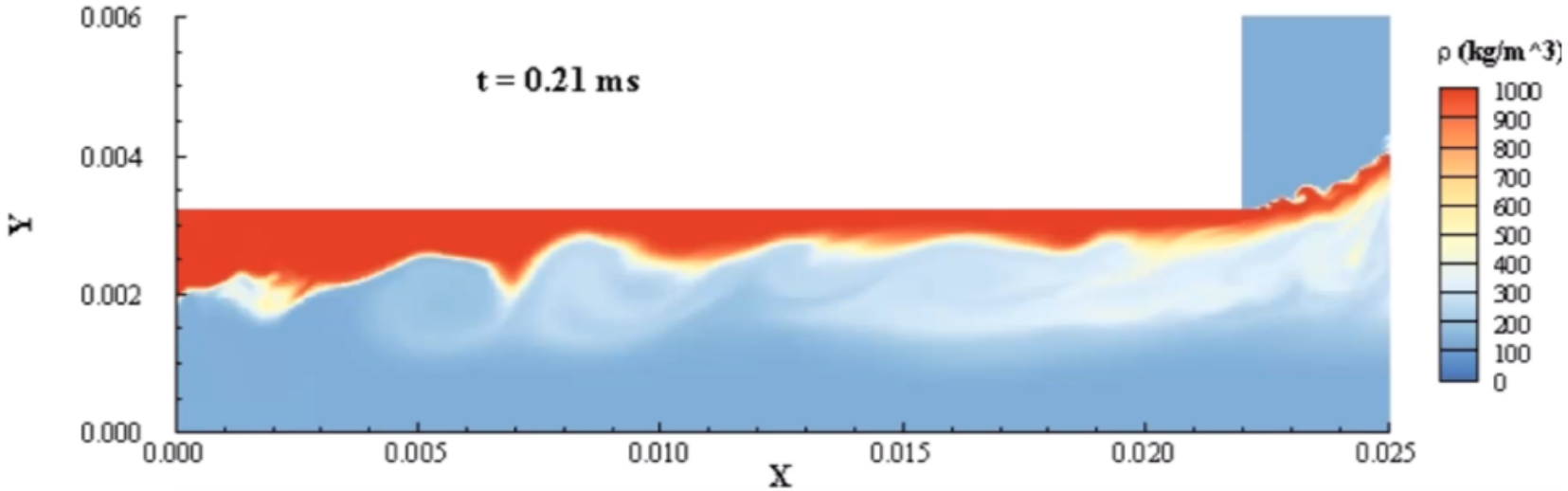


Rocket design

Optimizing **rocket injector** designs over a geometric **design space** (Mak et al. 2018; Yeh et al. 2018; Chang et al. 2019)



L	20-100 mm
R_n	2-5 mm
θ	45°-75°
δ	0.5-2 mm
ΔL	1-4 mm



Rocket design

Simulations are very **expensive**: **14 days** per **injector design** (each evaluation of f), **parallelized** over millions of cores

- **Exploring** the 5D domain Ω by simulations **alone** will take 1000's of simulations $\Rightarrow$ **years** of **computation**

One **solution**: build a **surrogate model** on f

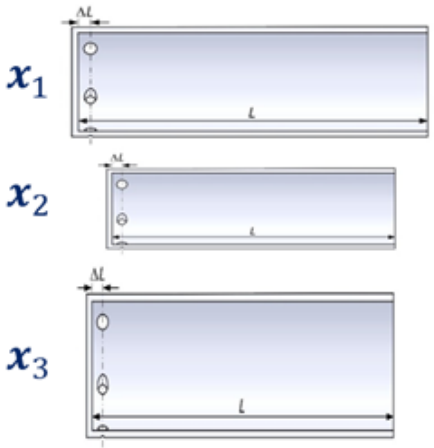
This is **Bayesian optimization**!

Rocket design

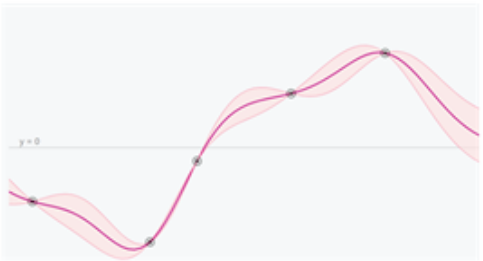
Simulations are very **expensive**: **14 days** per **injector design** (each evaluation of f), **parallelized** over millions of cores

- **Exploring** the 5D domain Ω by simulations **alone** will take 1000's of simulations $\Rightarrow$ **years** of **computation**

One **solution**: build a **surrogate model** on f



Run a few experiments at different injector geometries



Train a predictive model using simulation data

Use this model to **choose** a new geometry for simulation

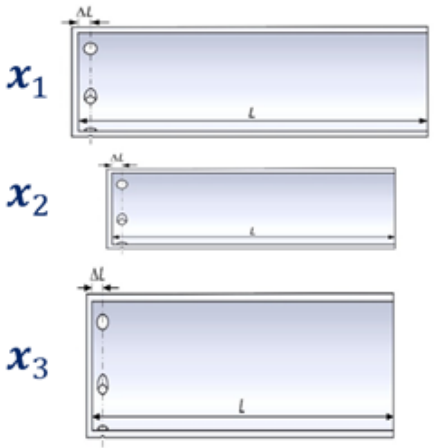
This is **Bayesian optimization**!

Rocket design

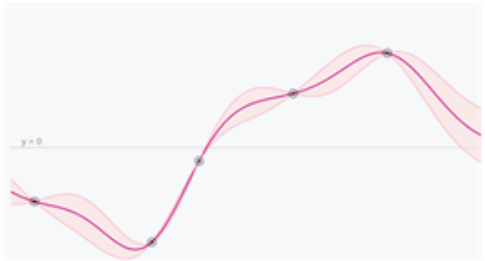
Simulations are very **expensive**: **14 days** per **injector design** (each evaluation of f), **parallelized** over millions of cores

- **Exploring** the 5D domain Ω by simulations **alone** will take 1000's of simulations $\Rightarrow$ **years** of **computation**

One **solution**: build a **surrogate model** on f



Run a few experiments at different injector geometries



Train a predictive model using simulation data

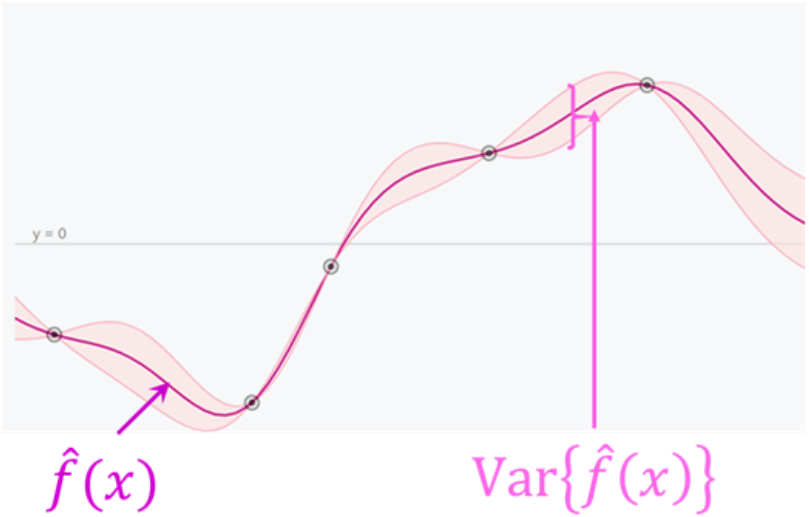
Use this model to **choose** a new geometry for simulation

This is **Bayesian optimization**!

Bayesian optimization

Two **ingredients** for **Bayesian optimization**:

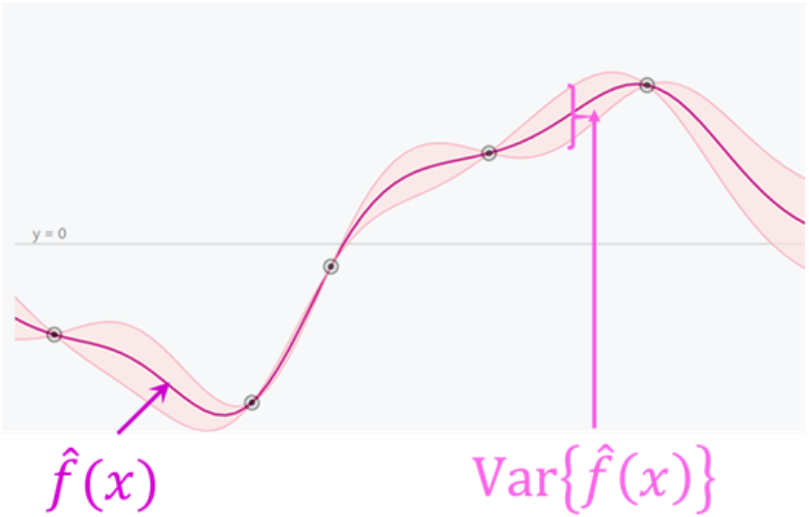
- **Bayesian** (probabilistic) model on **objective** f'n f:
  - Provide **predictor** of f: $\widehat{f} = \mathbb{E}\{f|\text{data}\}$
  - Quantify **uncertainty** on f: $\text{Var}\{f|\text{data}\}$
- **Acquisition function** for choosing **next** point x_{new}
  - **Exploitation**: want x's with low **predictions** $\widehat{f}$
  - **Exploration**: want x's with high **uncertainty** $\text{Var}\{\widehat{f}\}$

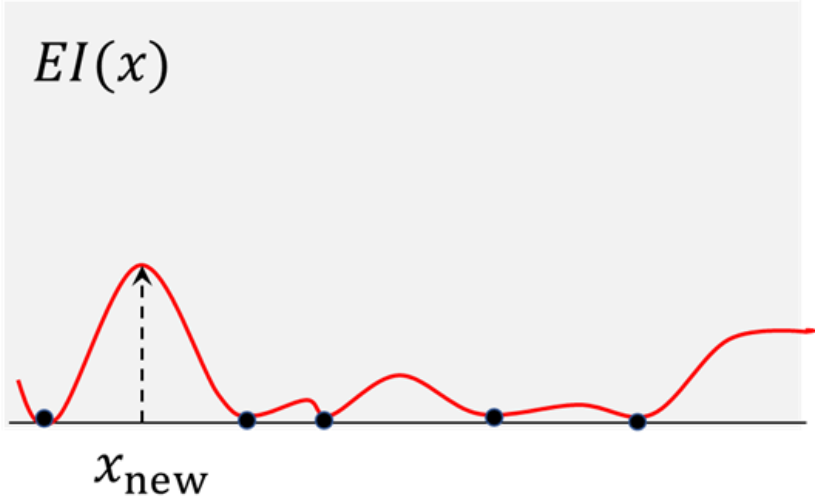


Bayesian optimization

Two **ingredients** for **Bayesian optimization**:

- **Bayesian** (probabilistic) model on **objective** f'n f:
  - Provide **predictor** of f: $\widehat{f} = \mathbb{E}\{f|\text{data}\}$
  - Quantify **uncertainty** on f: $\text{Var}\{f|\text{data}\}$
- **Acquisition function** for choosing **next** point $\boldsymbol{x}_{\text{new}}$
  - **Exploitation**: want $\boldsymbol{x}$'s with low **predictions** $\widehat{f}$
  - **Exploration**: want $\boldsymbol{x}$'s with high **uncertainty** $\text{Var}\{\widehat{f}\}$





Bayesian optimization

A **widely-used** setting in **Bayesian optimization**:

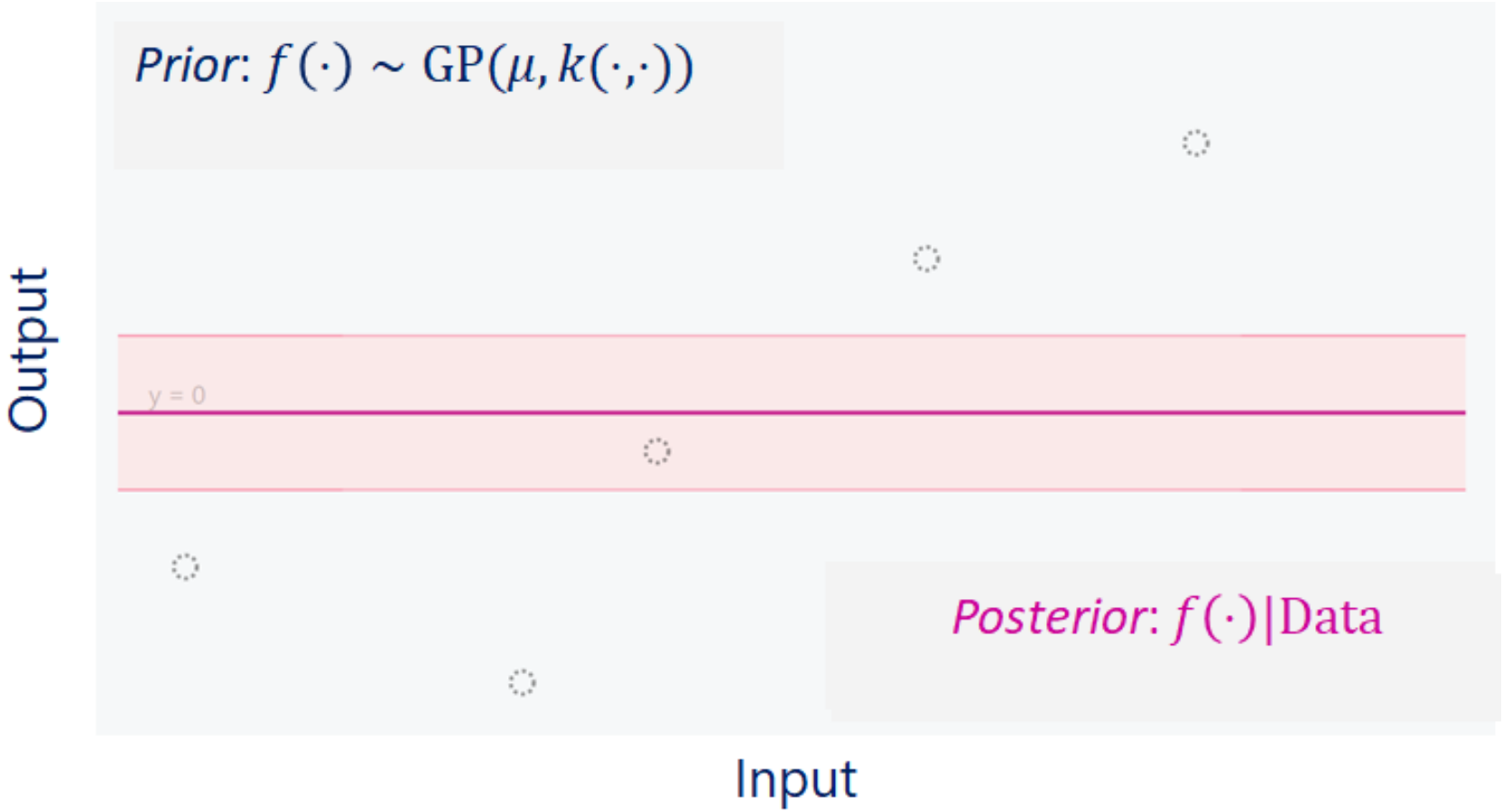
- **Bayesian** model on f: **Gaussian process** (kriging)
  - **Nonparametric** form $\Rightarrow$ **powerful approximation**
  - **Probabilistic** model $\Rightarrow$ **uncertainty quantification** (UQ)
  - **Closed-form** prediction & UQ $\Rightarrow$ **efficient modeling**
- **Acquisition function**: **Expected Improvement** (Jones et al. 1997)
  - **Closed-form** expression for $\mathrm{EI}(x)$ $\Rightarrow$ **efficient queries**
  - **Interpretable**

Bayesian optimization

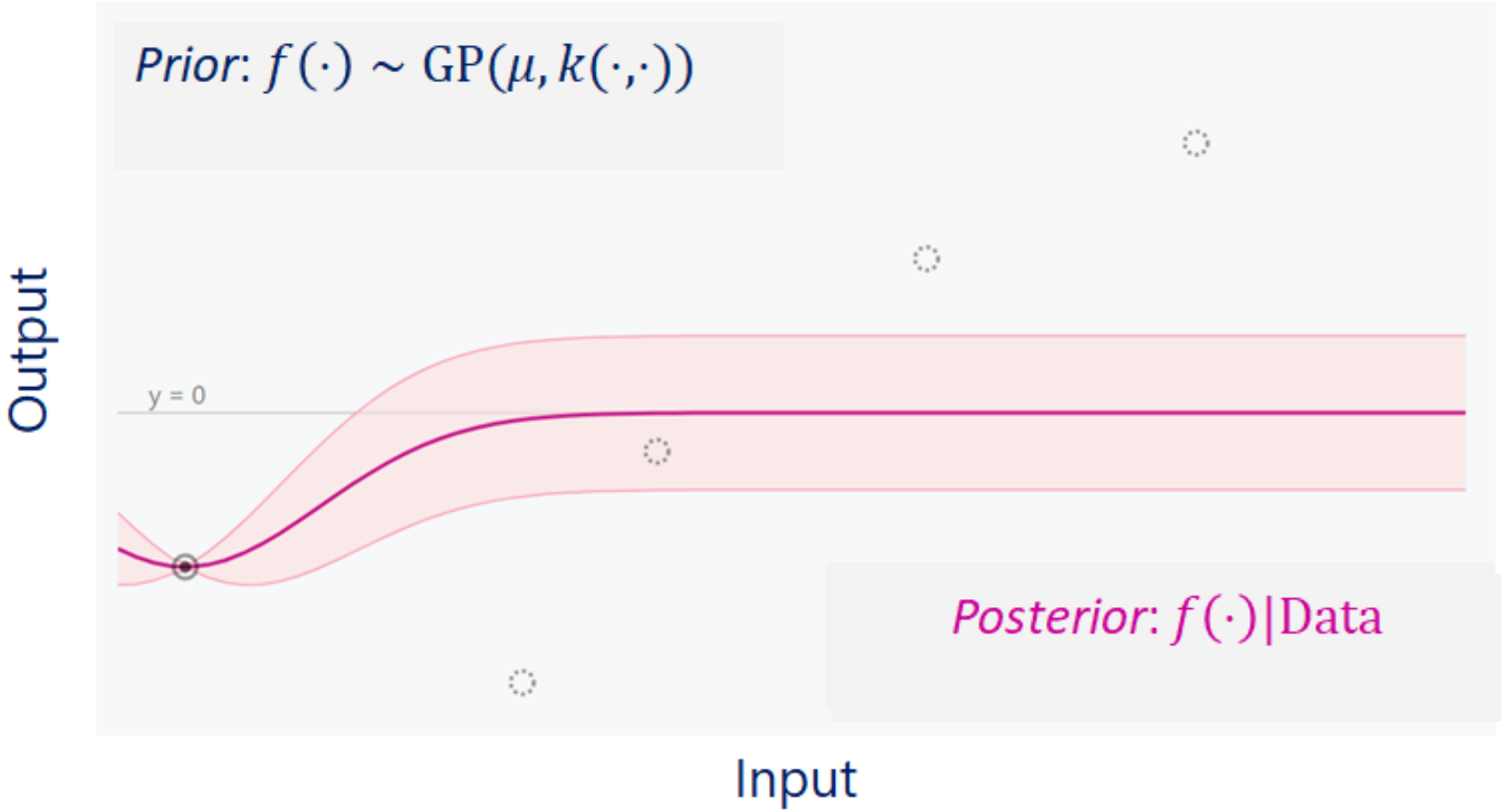
A **widely-used** setting in **Bayesian optimization**:

- **Bayesian** model on f: **Gaussian process** (kriging)
  - **Nonparametric** form $\Rightarrow$ **powerful approximation**
  - **Probabilistic** model $\Rightarrow$ **uncertainty quantification** (UQ)
  - **Closed-form** prediction & UQ $\Rightarrow$ **efficient modeling**
- **Acquisition function**: **Expected Improvement** (Jones et al. 1997)
  - **Closed-form** expression for $\mathrm{EI}(\boldsymbol{x})$ $\Rightarrow$ **efficient queries**
  - **Interpretable**

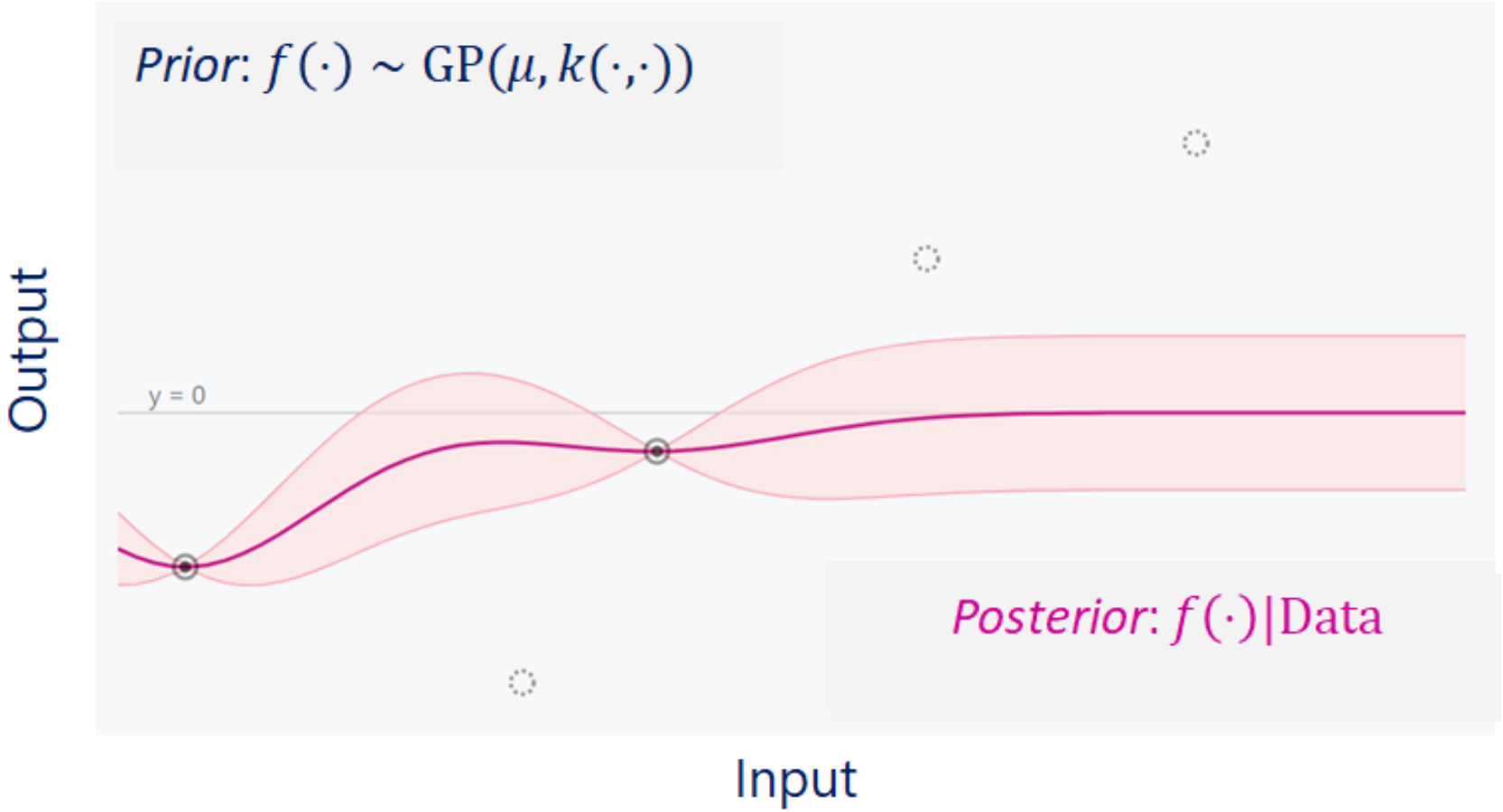
Gaussian process (kriging) model



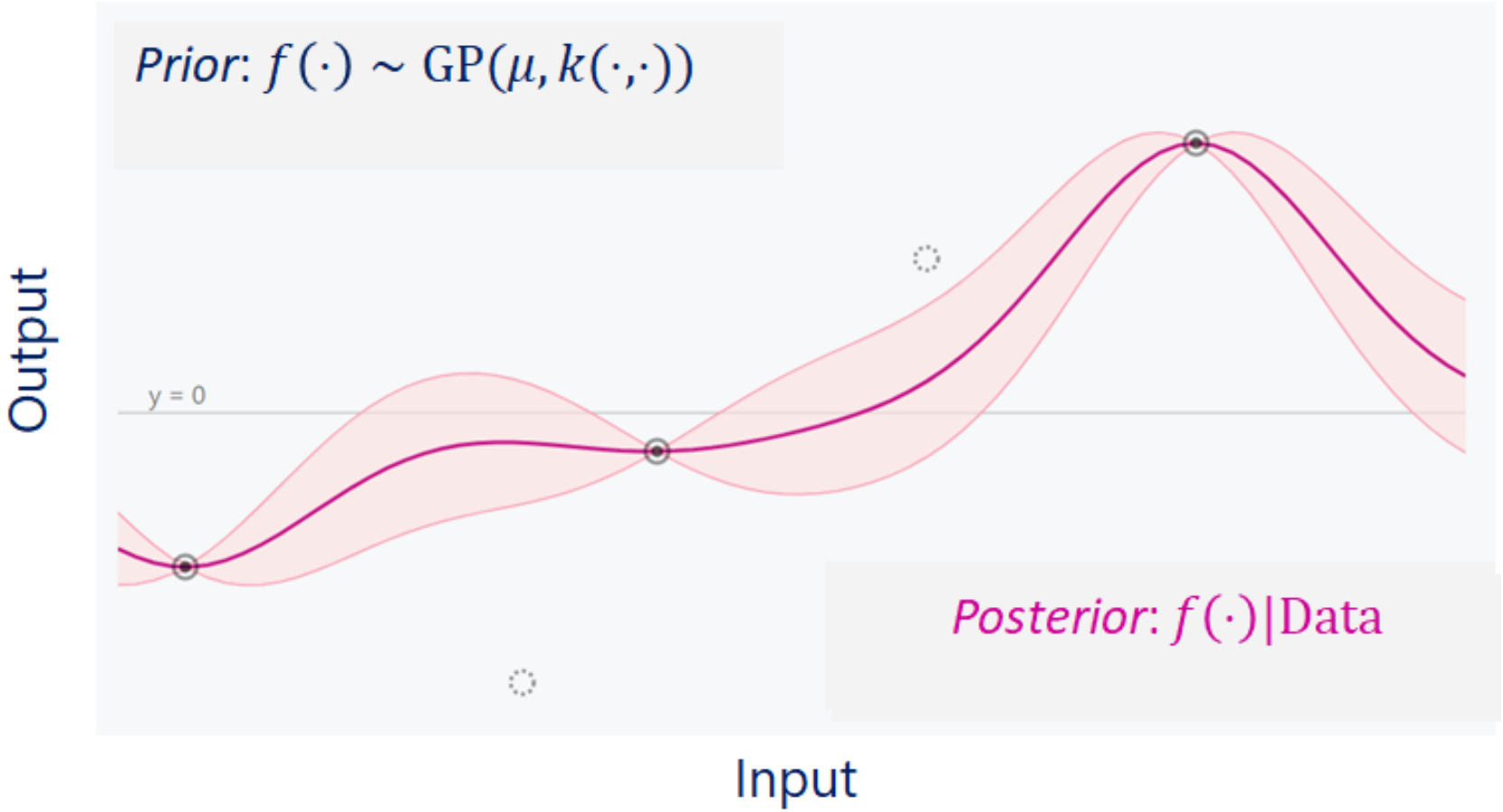
Gaussian process (kriging) model



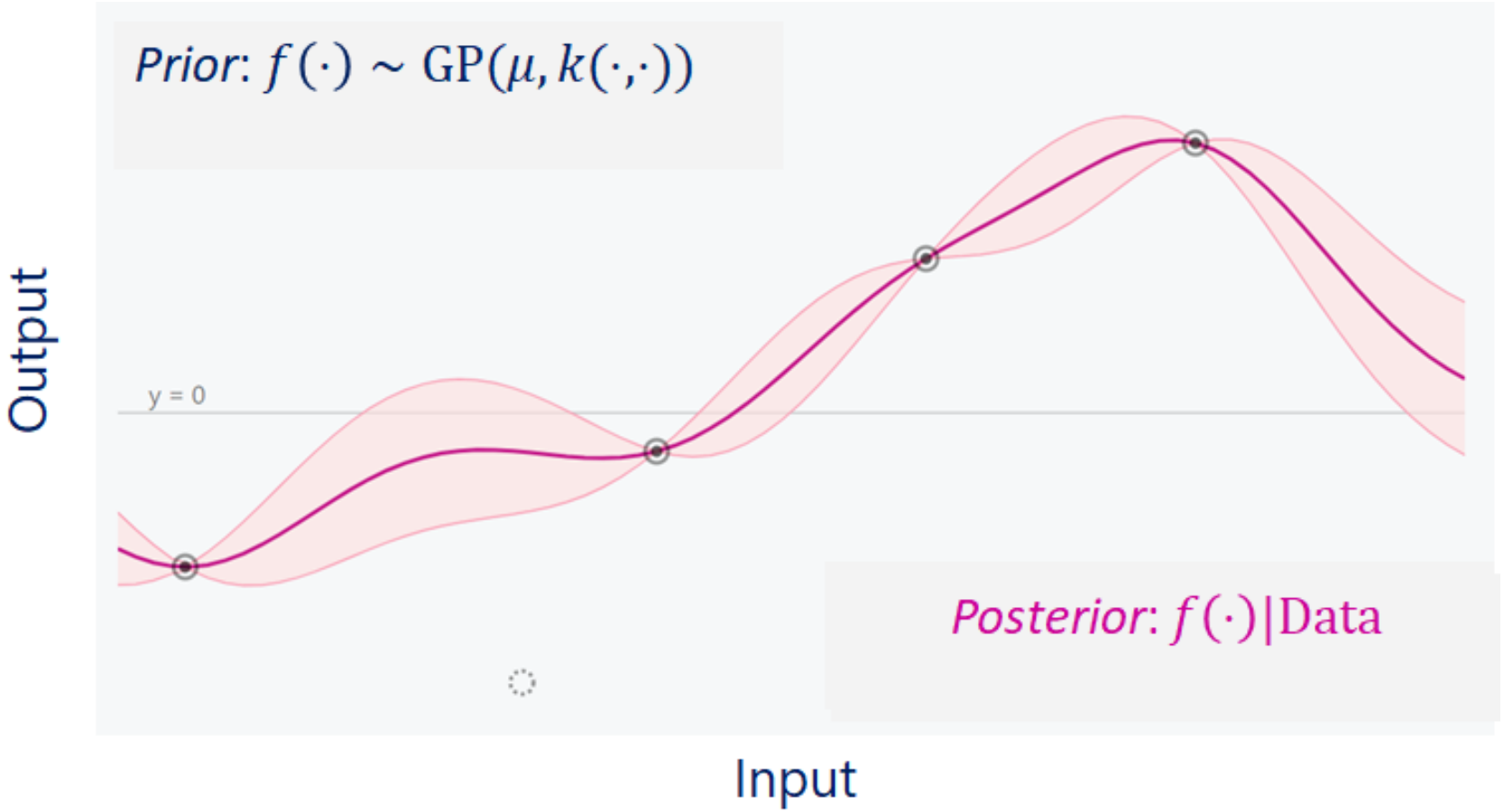
Gaussian process (kriging) model



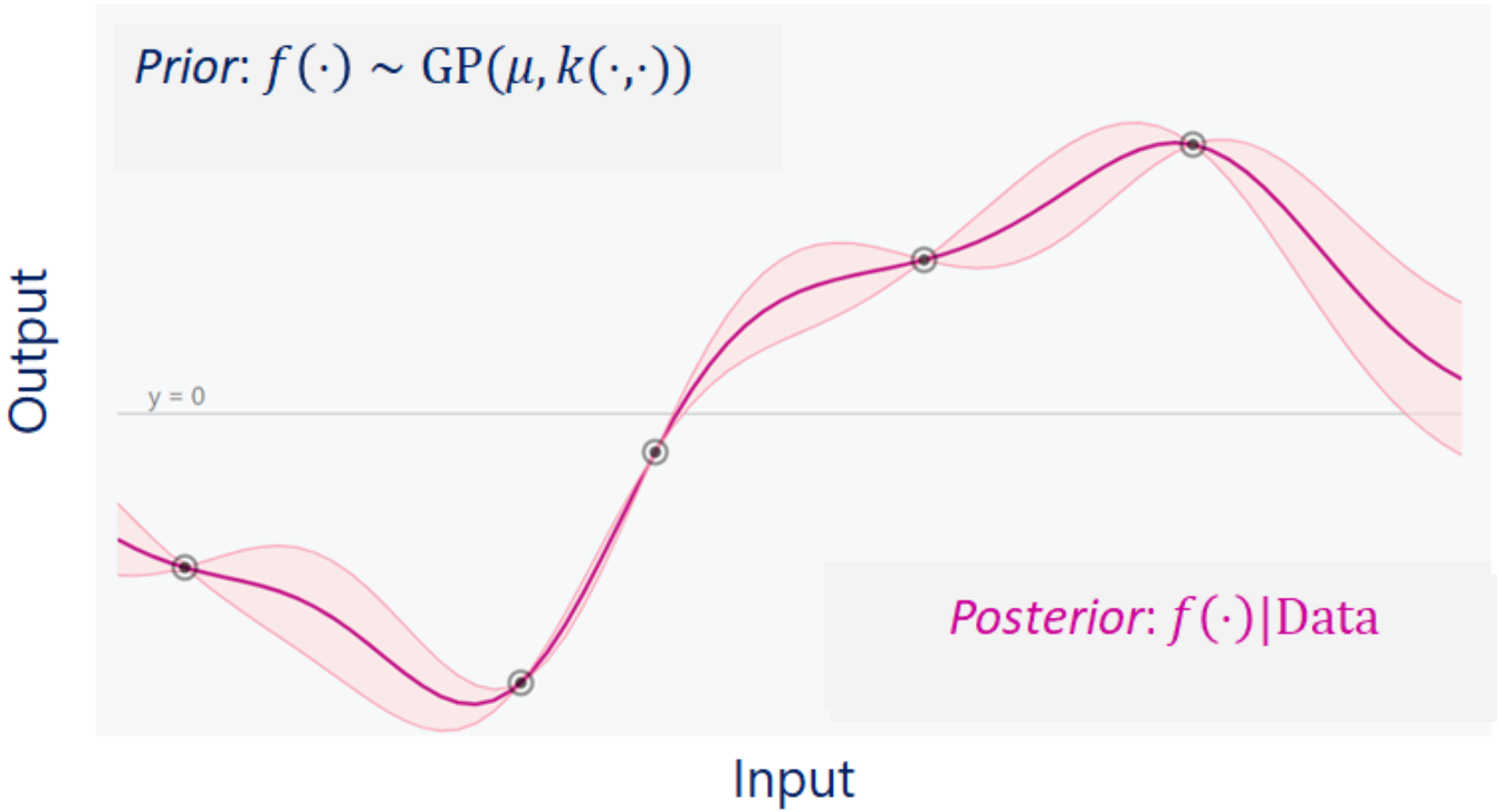
Gaussian process (kriging) model



Gaussian process (kriging) model



Gaussian process (kriging) model



Gaussian process (kriging) model

Universal kriging model (see, e.g., Santner et al. 2013):

$$f(\cdot) \sim GP\{\mu(\cdot), \sigma^2 r(\cdot,\cdot)\}, \quad \boldsymbol{x} \in \mathbb{R}^d.$$

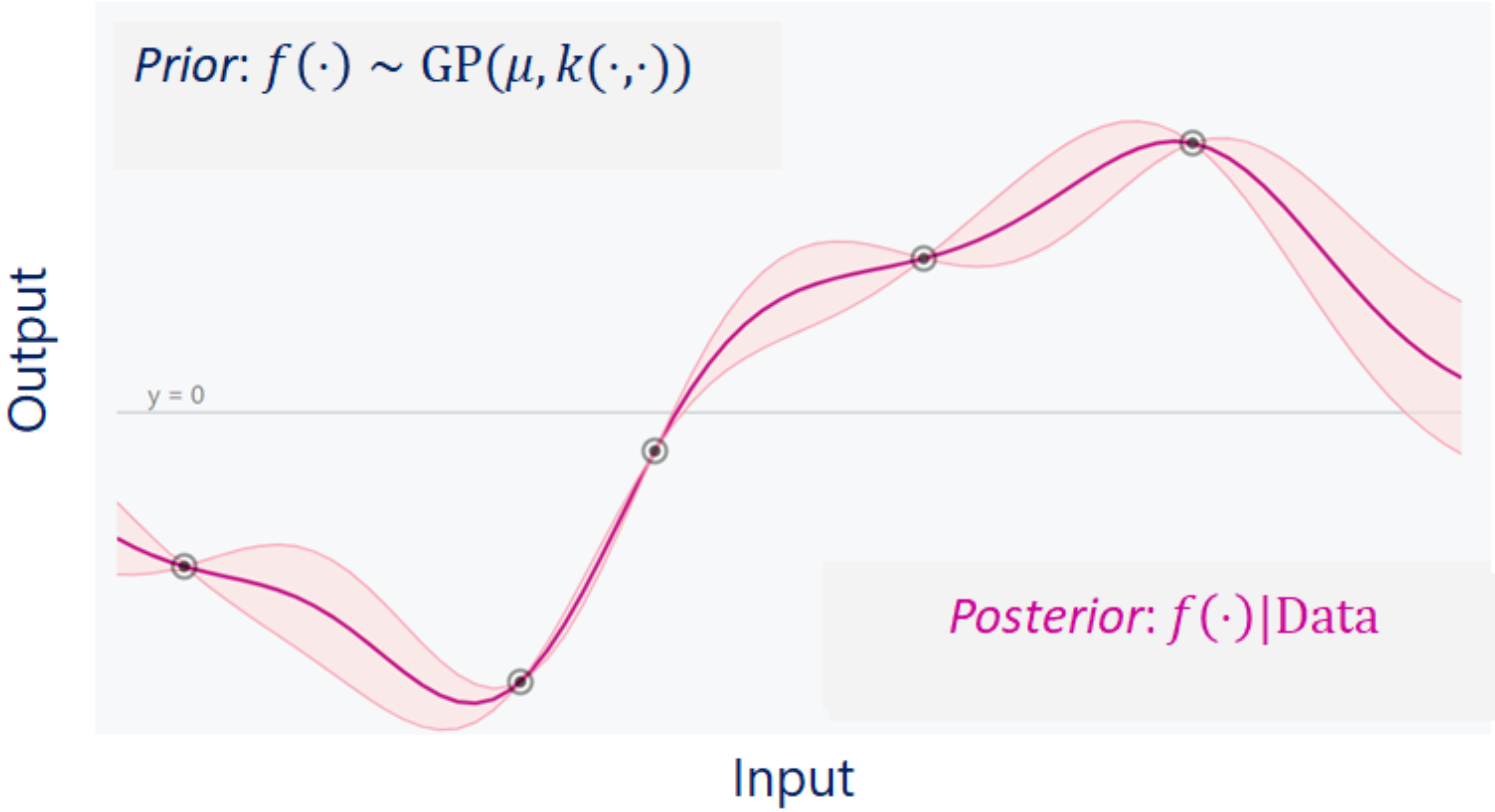
- **Mean** function: $\mu(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{\beta}$
- **Basis** functions: $\boldsymbol{p}(\boldsymbol{x}) = [p_1(\boldsymbol{x}), \cdots, p_q(\boldsymbol{x})]^\top$
- Process **variance**: $\sigma^2 = \mathrm{Var}\{f(\boldsymbol{x})\}$
- **Correlation** function: $r(\boldsymbol{x}, \boldsymbol{x}') = r(\boldsymbol{x} - \boldsymbol{x}')$

When **no** basis functions are used (i.e., $p(\boldsymbol{x}) = 1$), UK reduces to **ordinary kriging** (OK)

Gaussian process (kriging) model

Given **data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the posterior of $f(\boldsymbol{x})$ is **Gaussian**:

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim \mathcal{N}\left(\widehat{f}_n(\boldsymbol{x}), \sigma^2 s^2(\boldsymbol{x})\right)$$



Gaussian process (kriging) model

Given **data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the posterior of $f(\boldsymbol{x})$ is **Gaussian**:

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim \mathcal{N}\left(\widehat{f}_n(\boldsymbol{x}), \sigma^2 s^2(\boldsymbol{x})\right)$$

- Posterior **mean**: $\widehat{f}_n(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}\widehat{\boldsymbol{\beta}})$
  - $\widehat{\boldsymbol{\beta}} = (\boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{y}$
  - $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$, $\boldsymbol{P} = [\boldsymbol{p}(\boldsymbol{x}_1), \cdots, \boldsymbol{p}(\boldsymbol{x}_n)]^\top$
  - $\boldsymbol{R} = (r(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n\times n}$, $\boldsymbol{r}(\boldsymbol{x}) = [r(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, r(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$
- Posterior **variance**: $\sigma^2 s^2(\boldsymbol{x})$
  - σ^2: **prior** variance
  - $s^2(\boldsymbol{x}) = r(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{G}^{-1}\boldsymbol{p}(\boldsymbol{x})$, where $\boldsymbol{G} = \boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{P}$

Gaussian process (kriging) model

Given **data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the posterior of $f(\boldsymbol{x})$ is **Gaussian**:

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim \mathcal{N}\left(\widehat{f}_n(\boldsymbol{x}), \sigma^2 s^2(\boldsymbol{x})\right)$$

- Posterior **mean**: $\widehat{f}_n(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}\widehat{\boldsymbol{\beta}})$
  - $\widehat{\boldsymbol{\beta}} = (\boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{y}$
  - $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$, $\boldsymbol{P} = [\boldsymbol{p}(\boldsymbol{x}_1), \cdots, \boldsymbol{p}(\boldsymbol{x}_n)]^\top$
  - $\boldsymbol{R} = (r(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n\times n}$, $\boldsymbol{r}(\boldsymbol{x}) = [r(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, r(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$
- Posterior **variance**: $\sigma^2 s^2(\boldsymbol{x})$
  - σ^2: **prior** variance
  - $s^2(\boldsymbol{x}) = r(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{G}^{-1}\boldsymbol{p}(\boldsymbol{x})$, where $\boldsymbol{G} = \boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{P}$

Gaussian process (kriging) model

Given **data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the posterior of $f(\boldsymbol{x})$ is **Gaussian**:

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim \mathcal{N}\left(\widehat{f}_n(\boldsymbol{x}), \sigma^2 s^2(\boldsymbol{x})\right)$$

- Posterior **mean**: $\widehat{f}_n(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}\widehat{\boldsymbol{\beta}})$
  - $\widehat{\boldsymbol{\beta}} = (\boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{y}$
  - $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$, $\boldsymbol{P} = [\boldsymbol{p}(\boldsymbol{x}_1), \cdots, \boldsymbol{p}(\boldsymbol{x}_n)]^\top$
  - $\boldsymbol{R} = (r(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n\times n}$, $\boldsymbol{r}(\boldsymbol{x}) = [r(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, r(\boldsymbol{x}, \boldsymbol{x}_n))]^\top$
- Posterior **variance**: $\sigma^2 s^2(\boldsymbol{x})$
  - σ^2: **prior** variance
  - $s^2(\boldsymbol{x}) = r(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{G}^{-1}\boldsymbol{p}(\boldsymbol{x})$, where $\boldsymbol{G} = \boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{P}$

Gaussian process (kriging) model

Given **data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the posterior of $f(\boldsymbol{x})$ is **Gaussian**:

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim \mathcal{N}\left(\widehat{f}_n(\boldsymbol{x}), \sigma^2 s^2(\boldsymbol{x})\right)$$

- Posterior **mean**: $\widehat{f}_n(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}\widehat{\boldsymbol{\beta}})$
  - $\widehat{\boldsymbol{\beta}} = (\boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{y}$
  - $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$, $\boldsymbol{P} = [\boldsymbol{p}(\boldsymbol{x}_1), \cdots, \boldsymbol{p}(\boldsymbol{x}_n)]^\top$
  - $\boldsymbol{R} = (r(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n\times n}$, $\boldsymbol{r}(\boldsymbol{x}) = [r(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, r(\boldsymbol{x}, \boldsymbol{x}_n))]^\top$
- Posterior **variance**: $\sigma^2 s^2(\boldsymbol{x})$
  - σ^2: **prior** variance
  - $s^2(\boldsymbol{x}) = r(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{G}^{-1}\boldsymbol{p}(\boldsymbol{x})$, where $\boldsymbol{G} = \boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{P}$

Gaussian process (kriging) model

Given **data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the posterior of $f(\boldsymbol{x})$ is **Gaussian**:

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim \mathcal{N}\left(\widehat{f}_n(\boldsymbol{x}), \sigma^2 s^2(\boldsymbol{x})\right)$$

- Posterior **mean**: $\widehat{f}_n(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}\widehat{\boldsymbol{\beta}})$
  - $\widehat{\boldsymbol{\beta}} = (\boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{y}$
  - $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$, $\boldsymbol{P} = [\boldsymbol{p}(\boldsymbol{x}_1), \cdots, \boldsymbol{p}(\boldsymbol{x}_n)]^\top$
  - $\boldsymbol{R} = (r(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n\times n}$, $\boldsymbol{r}(\boldsymbol{x}) = [r(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, r(\boldsymbol{x}, \boldsymbol{x}_n))]^\top$
- Posterior **variance**: $\sigma^2 s^2(x)$
  - σ^2: **prior** variance
  - $s^2(\boldsymbol{x}) = r(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{G}^{-1}\boldsymbol{p}(\boldsymbol{x})$, where $\boldsymbol{G} = \boldsymbol{P}^\top \boldsymbol{R}^{-1}\boldsymbol{P}$

Gaussian process (kriging) model

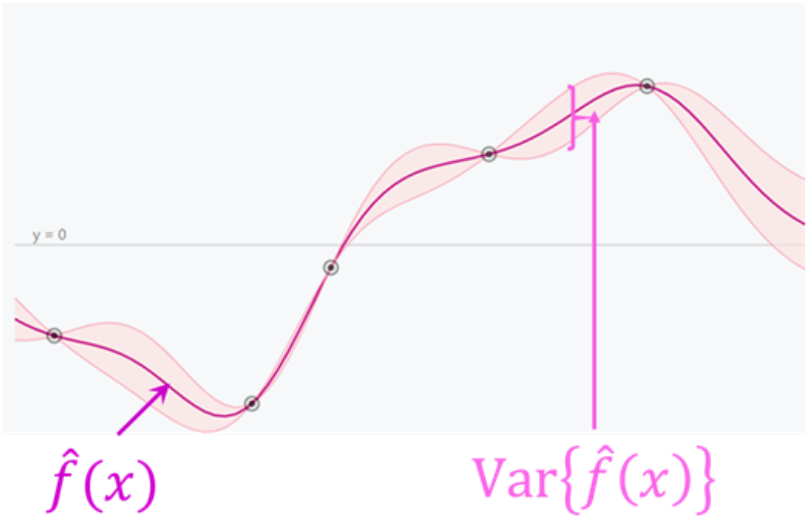
Given **data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the posterior of $f(\boldsymbol{x})$ is **Gaussian**:

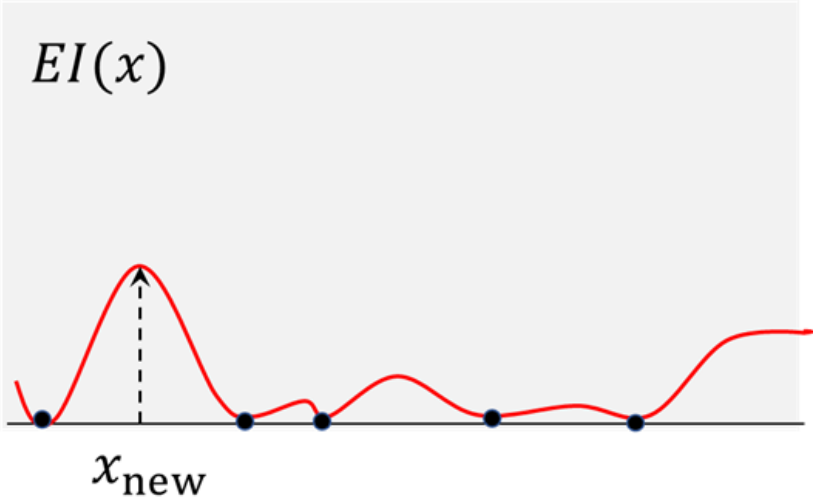
$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim \mathcal{N}\left(\widehat{f}_n(\boldsymbol{x}), \sigma^2 s^2(\boldsymbol{x})\right)$$

- Posterior **mean**: $\widehat{f}_n(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}\widehat{\boldsymbol{\beta}})$
  - $\widehat{\boldsymbol{\beta}} = (\boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{y}$
  - $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$, $\boldsymbol{P} = [\boldsymbol{p}(\boldsymbol{x}_1), \cdots, \boldsymbol{p}(\boldsymbol{x}_n)]^\top$
  - $\boldsymbol{R} = (r(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n\times n}$, $\boldsymbol{r}(\boldsymbol{x}) = [r(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, r(\boldsymbol{x}, \boldsymbol{x}_n))]^\top$
- Posterior **variance**: $\sigma^2 s^2(\boldsymbol{x})$
  - σ^2: **prior** variance
  - $s^2(\boldsymbol{x}) = r(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}^\top(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{G}^{-1}\boldsymbol{p}(\boldsymbol{x})$, where $\boldsymbol{G} = \boldsymbol{P}^\top\boldsymbol{R}^{-1}\boldsymbol{P}$

Expected Improvement

How to incorporate this **probabilistic** predictor within the **acquisition function**?



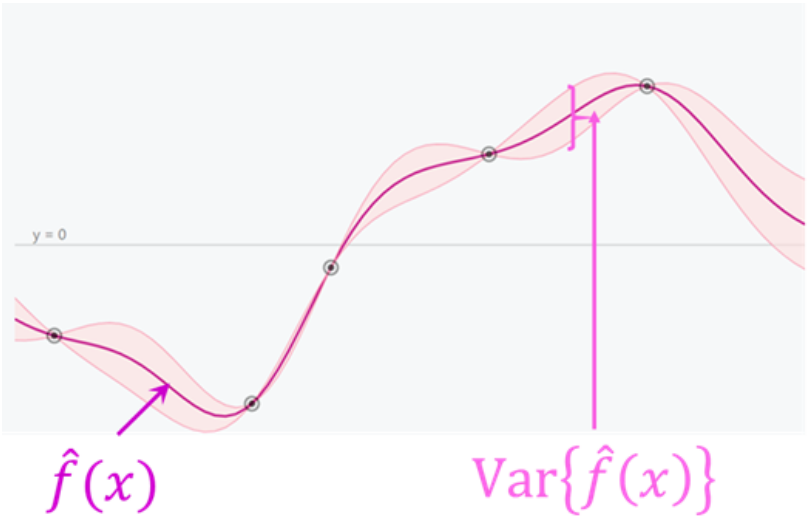


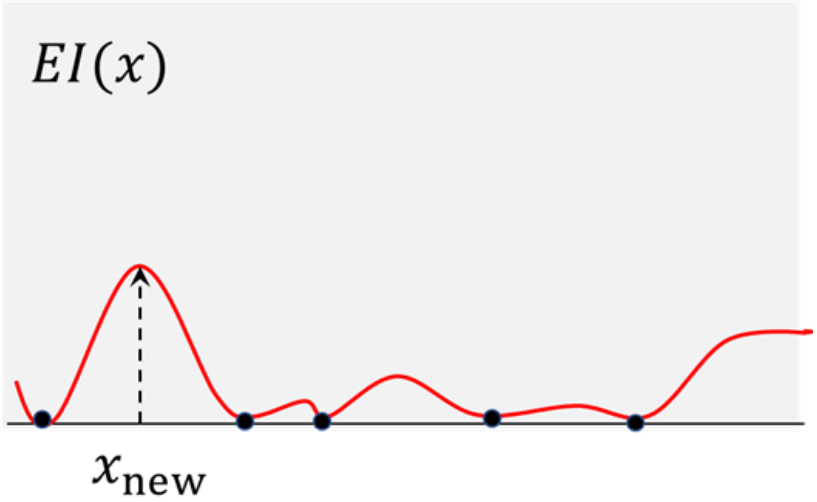
Idea: want to choose a new point x which yields greatest **improvement** over the current **minimum** $y^* = \min_{i=1}^{n} y_i$:

$$\max(y^* - f(x), 0)$$

Expected Improvement

How to incorporate this **probabilistic** predictor within the **acquisition function**?





Idea: want to choose a new point $\boldsymbol{x}$ which yields greatest **improvement** over the current **minimum** $y^* = \min_{i=1}^{n} y_i$:

$$\max(y^* - f(\boldsymbol{x}), 0)$$

Expected Improvement

Expected improvement (Jones et al. 1997)

$$\arg\max_{\boldsymbol{x}\in\Omega} \mathsf{EI}(\boldsymbol{x}) := \mathbb{E}[\max(y^* - f(\boldsymbol{x}), 0)|\mathcal{D}]$$

$$= I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right) + \sigma s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right).$$

- $y^* = \min_{i=1}^n y_i$: current **best** value
- $I(\boldsymbol{x}) = y^* - \widehat{f}_n(\boldsymbol{x})$: **difference** between y^* and predictor $\widehat{f}_n(\boldsymbol{x})$
- ϕ, Φ: p.d.f. and c.d.f. of a standard **normal** distribution
- $\sigma s_n(\boldsymbol{x})$: **posterior standard deviation** of $f(\boldsymbol{x})$

Expected Improvement

Expected improvement (Jones et al. 1997)

$$\arg\max_{\boldsymbol{x}\in\Omega} \mathsf{EI}(\boldsymbol{x}) := \mathbb{E}[\max(y^* - f(\boldsymbol{x}), 0)|\mathcal{D}]$$

$$= I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right) + \sigma s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right).$$

- $y^* = \min_{i=1}^n y_i$: current **best** value
- $I(\boldsymbol{x}) = y^* - \widehat{f}_n(\boldsymbol{x})$: **difference** between y^* and predictor $\widehat{f}_n(\boldsymbol{x})$
- ϕ, Φ: p.d.f. and c.d.f. of a standard **normal** distribution
- $\sigma s_n(\boldsymbol{x})$: **posterior standard deviation** of $f(\boldsymbol{x})$

Expected Improvement

Expected improvement (Jones et al. 1997)

$$\arg\max_{\boldsymbol{x}\in\Omega} \mathsf{EI}(\boldsymbol{x}) := \mathbb{E}[\max(y^* - f(\boldsymbol{x}), 0)|\mathcal{D}]$$

$$= I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right) + \sigma s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right).$$

- $y^* = \min_{i=1}^n y_i$: current **best** value
- $I(\boldsymbol{x}) = y^* - \widehat{f}_n(\boldsymbol{x})$: **difference** between y^* and predictor $\widehat{f}_n(\boldsymbol{x})$
- ϕ, Φ: p.d.f. and c.d.f. of a standard **normal** distribution
- $\sigma s_n(\boldsymbol{x})$: **posterior standard deviation** of $f(\boldsymbol{x})$

Expected Improvement

EI has an intuitive **interpretation**:

$$\underset{\boldsymbol{x}\in\Omega}{\arg\max}\ \mathsf{EI}(\boldsymbol{x}) := \underbrace{I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploitation}} + \underbrace{\sigma s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploration}}.$$

- $I(\boldsymbol{x}) = y^* - \widehat{f}_n(\boldsymbol{x})$: **difference** between y^* and predictor $\widehat{f}_n(\boldsymbol{x})$
- $\sigma s_n(\boldsymbol{x})$: **posterior standard deviation** of $f(\boldsymbol{x})$
- **Exploitation**: favor new points x with **small** predicted objective $\widehat{f}_n(x)$ – "**exploit**" the fitted model to find **better** solutions
- **Exploration**: favor new points x with **large** variance – "**explore**" uncertain regions to find **better** solutions
- EI balances the **exploitation-exploration trade-off** (Kearns & Singh 2002)

Expected Improvement

EI has an intuitive **interpretation**:

$$\underset{\boldsymbol{x}\in\Omega}{\arg\max}\ \mathsf{EI}(\boldsymbol{x}) := \underbrace{I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploitation}} + \underbrace{\sigma s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploration}}.$$

- $I(\boldsymbol{x}) = y^* - \widehat{f}_n(\boldsymbol{x})$: **difference** between y^* and predictor $\widehat{f}_n(\boldsymbol{x})$
- $\sigma s_n(\boldsymbol{x})$: **posterior standard deviation** of $f(\boldsymbol{x})$
- **Exploitation**: favor new points $\boldsymbol{x}$ with **small** predicted objective $\widehat{f}_n(\boldsymbol{x})$ – "**exploit**" the fitted model to find **better** solutions
- **Exploration**: favor new points $\boldsymbol{x}$ with **large** variance – "**explore**" uncertain regions to find **better** solutions
- EI balances the **exploitation-exploration trade-off** (Kearns & Singh 2002)

Expected Improvement

EI has an intuitive **interpretation**:

$$\underset{\boldsymbol{x}\in\Omega}{\arg\max}\ \mathsf{EI}(\boldsymbol{x}) := \underbrace{I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploitation}} + \underbrace{\sigma s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploration}}.$$

- $I(\boldsymbol{x}) = y^* - \widehat{f}_n(\boldsymbol{x})$: **difference** between y^* and predictor $\widehat{f}_n(\boldsymbol{x})$
- $\sigma s_n(\boldsymbol{x})$: **posterior standard deviation** of $f(\boldsymbol{x})$
- **Exploitation**: favor new points $\boldsymbol{x}$ with **small** predicted objective $\widehat{f}_n(\boldsymbol{x})$ – "**exploit**" the fitted model to find **better** solutions
- **Exploration**: favor new points $\boldsymbol{x}$ with **large** variance – "**explore**" uncertain regions to find **better** solutions
- EI balances the **exploitation-exploration trade-off** (Kearns & Singh 2002)

Expected Improvement

EI has an intuitive **interpretation**:

$$\underset{\boldsymbol{x}\in\Omega}{\arg\max}\ \mathsf{EI}(\boldsymbol{x}) := \underbrace{I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploitation}} + \underbrace{\sigma s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\sigma s_n(\boldsymbol{x})}\right)}_{\textbf{Exploration}}.$$

- $I(\boldsymbol{x}) = y^* - \widehat{f}_n(\boldsymbol{x})$: **difference** between y^* and predictor $\widehat{f}_n(\boldsymbol{x})$
- $\sigma s_n(\boldsymbol{x})$: **posterior standard deviation** of $f(\boldsymbol{x})$
- **Exploitation**: favor new points $\boldsymbol{x}$ with **small** predicted objective $\widehat{f}_n(\boldsymbol{x})$ – "**exploit**" the fitted model to find **better** solutions
- **Exploration**: favor new points $\boldsymbol{x}$ with **large** variance – "**explore**" uncertain regions to find **better** solutions
- EI balances the **exploitation-exploration trade-off** (Kearns & Singh 2002)

Plug-in estimation

In practice, the GP **variance** parameter σ^2 is **unknown** and needs to be **estimated**:

- **Estimate** σ^2 via MLE

$$\widehat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{P}^\top \widehat{\boldsymbol{\beta}})^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}^\top \widehat{\boldsymbol{\beta}})$$

- **Plug-in** $\widehat{\sigma}^2$ into acquisition function $\mathrm{EI}(\boldsymbol{x})$
- **Optimize** $\mathrm{EI}(\boldsymbol{x})$ for new query point $\boldsymbol{x}_{\mathrm{new}}$

Sounds reasonable... what can go wrong?

Plug-in estimation

In practice, the GP **variance** parameter σ^2 is **unknown** and needs to be **estimated**:

- **Estimate** σ^2 via MLE

$$\widehat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{P}^\top \widehat{\boldsymbol{\beta}})^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}^\top \widehat{\boldsymbol{\beta}})$$

- **Plug-in** $\widehat{\sigma}^2$ into acquisition function $\mathrm{EI}(\boldsymbol{x})$
- **Optimize** $\mathrm{EI}(\boldsymbol{x})$ for new query point $\boldsymbol{x}_{\mathrm{new}}$

Sounds **reasonable**... what can go **wrong**?

Plug-in estimation

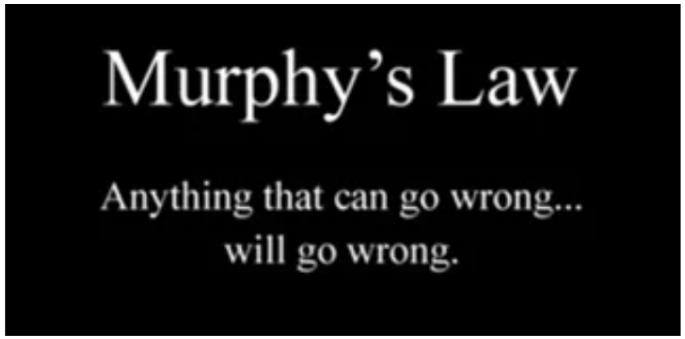
In practice, the GP **variance** parameter σ^2 is **unknown** and needs to be **estimated**:

- **Estimate** σ^2 via MLE

$$\widehat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{P}^\top \widehat{\boldsymbol{\beta}})^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{P}^\top \widehat{\boldsymbol{\beta}})$$

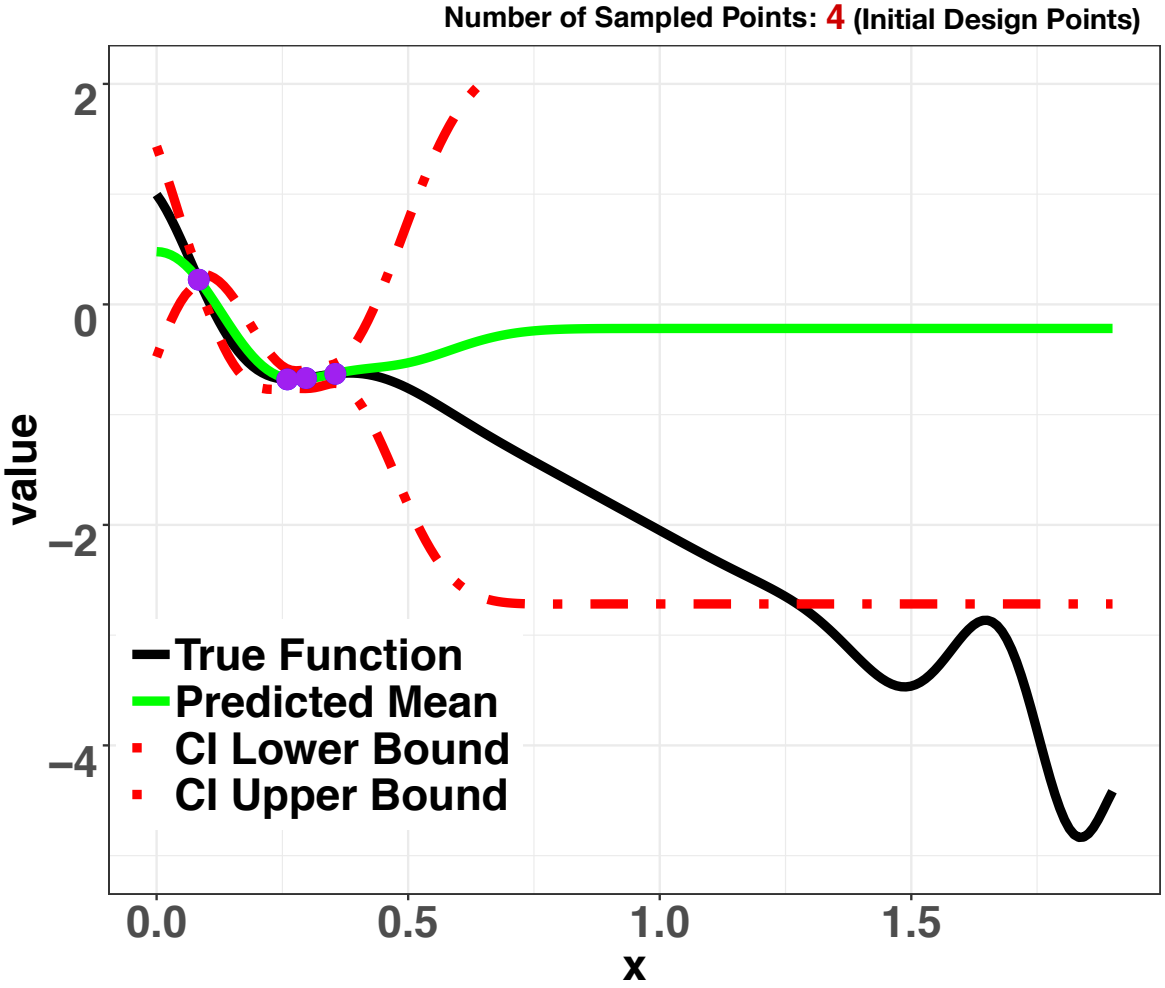
- **Plug-in** $\widehat{\sigma}^2$ into acquisition function $\mathrm{EI}(\boldsymbol{x})$
- **Optimize** $\mathrm{EI}(\boldsymbol{x})$ for new query point $\boldsymbol{x}_{\mathrm{new}}$

Sounds **reasonable**... what can go **wrong**?



Over-exploitation

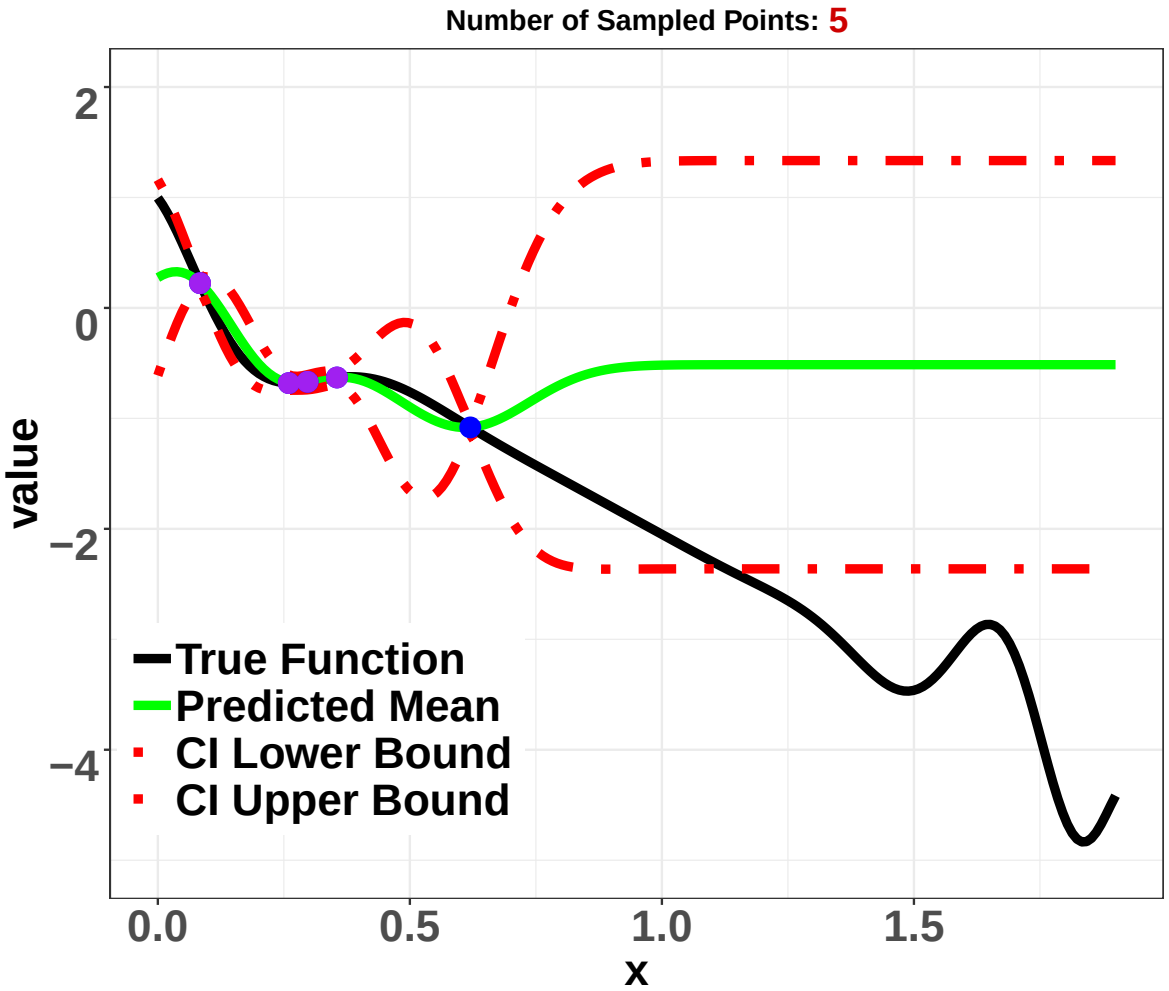
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

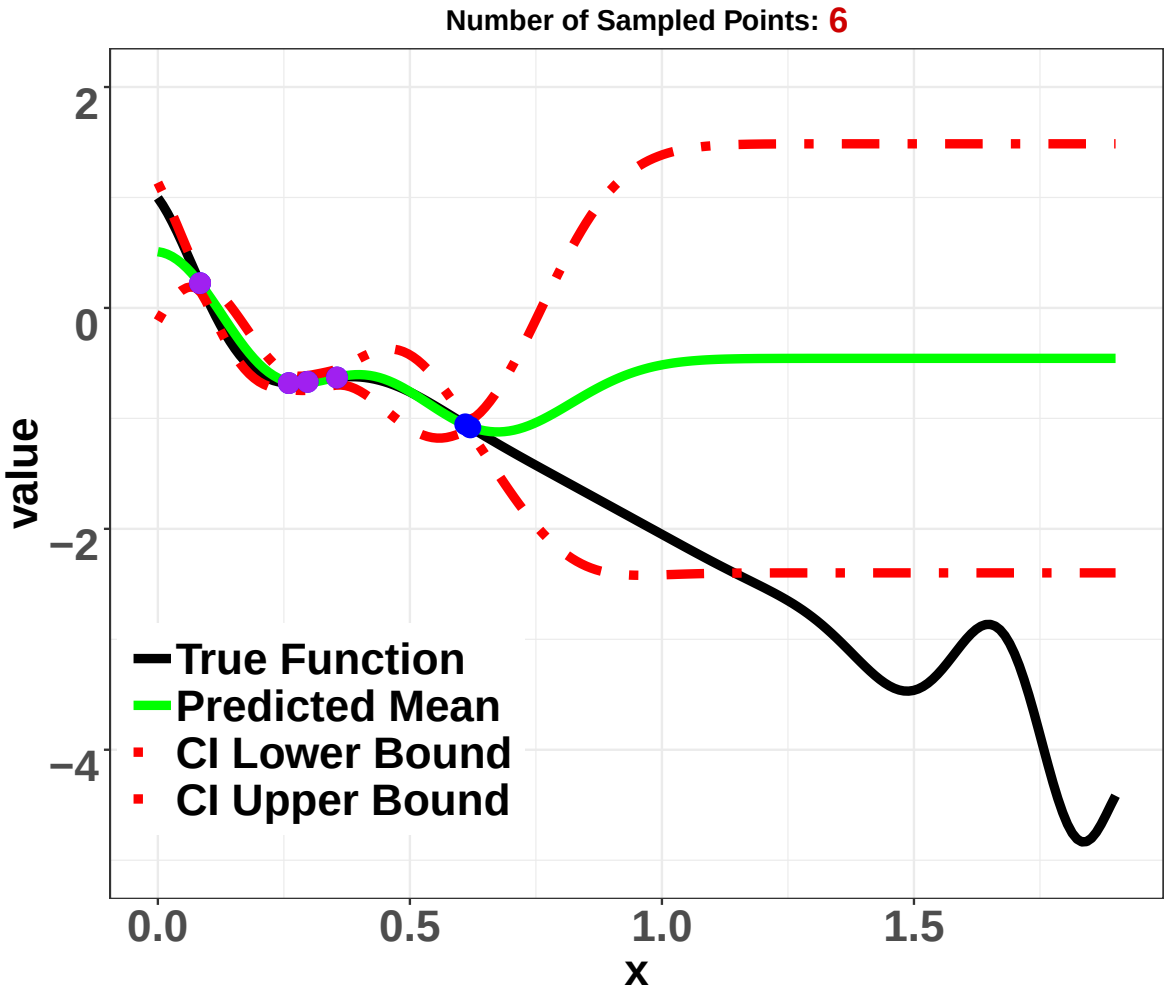
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

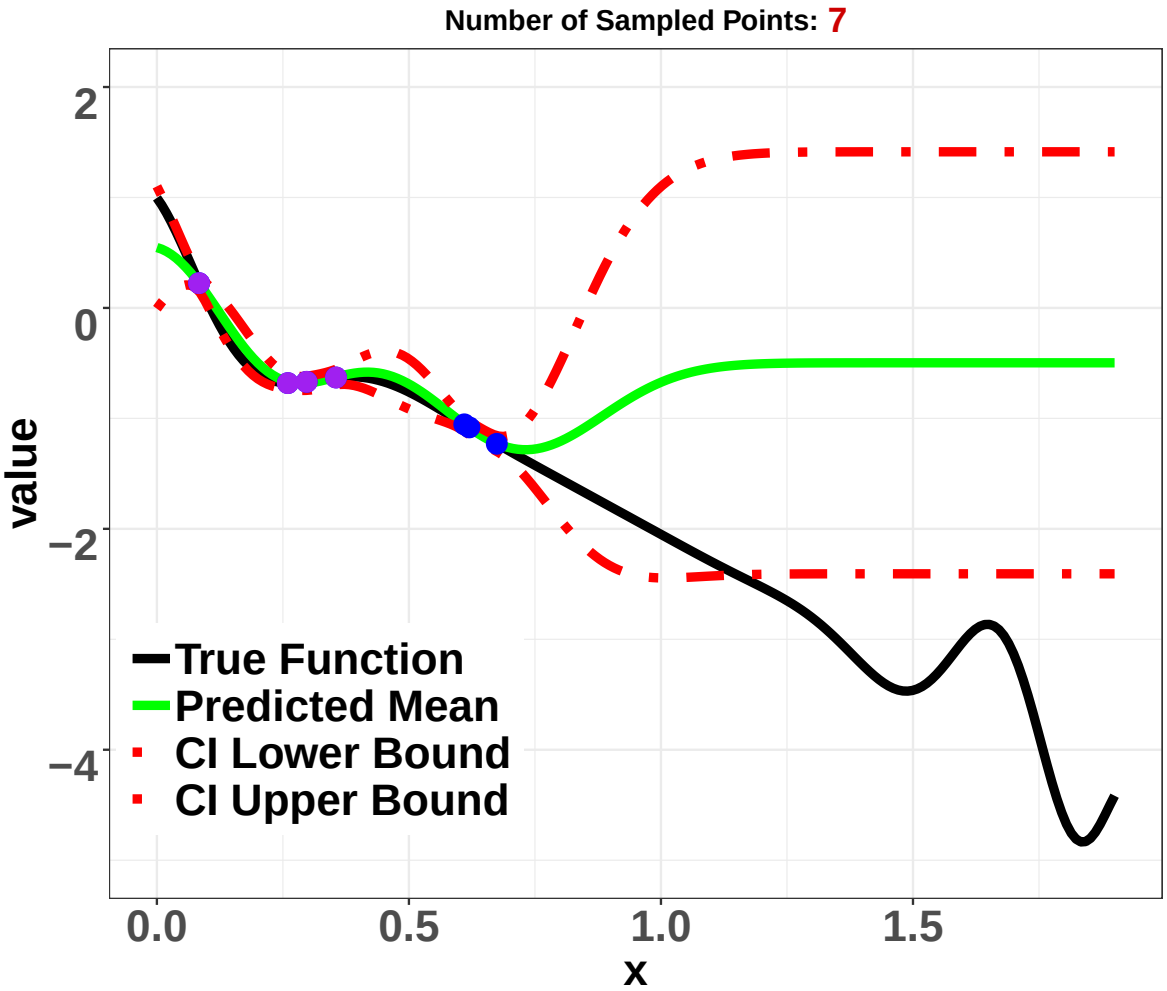
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

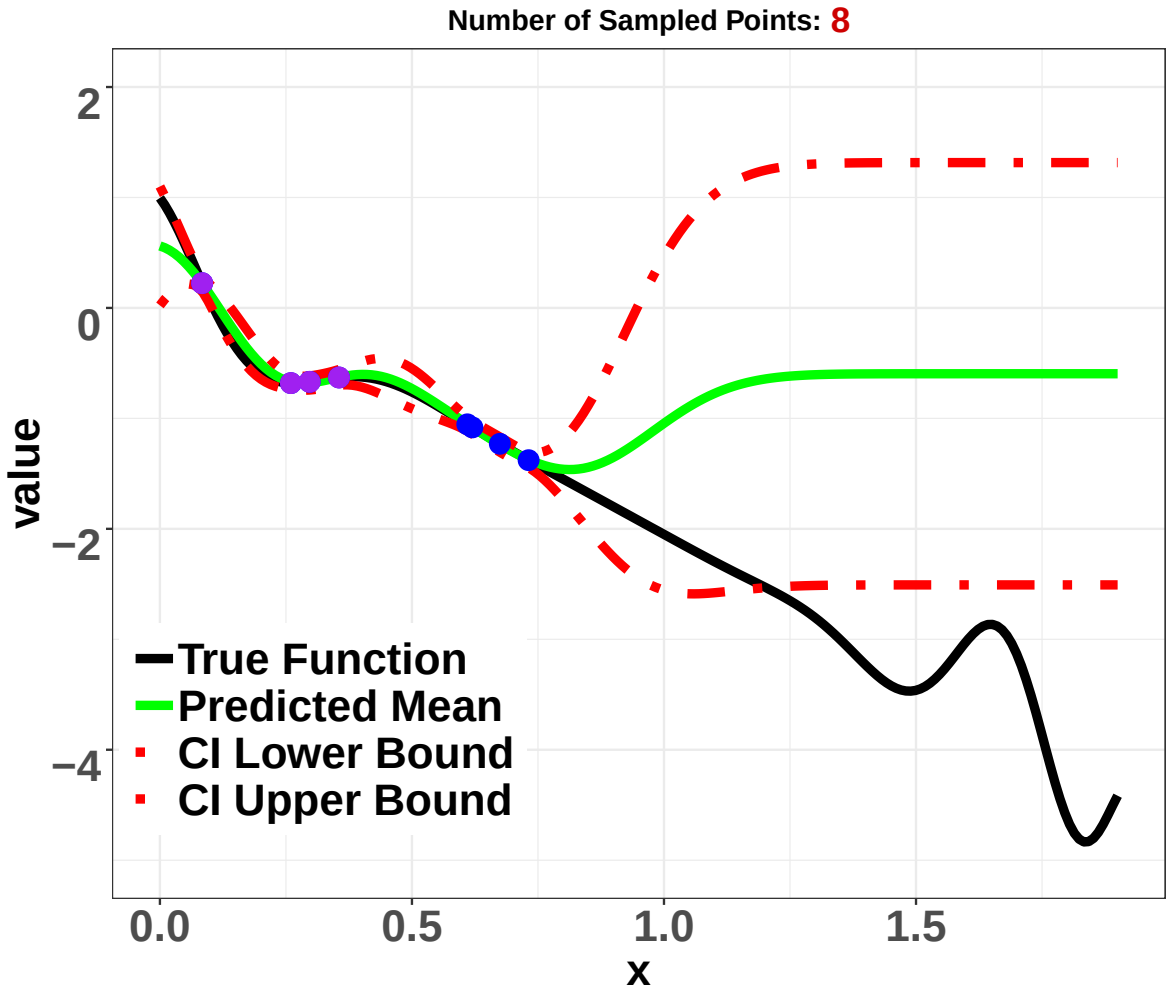
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

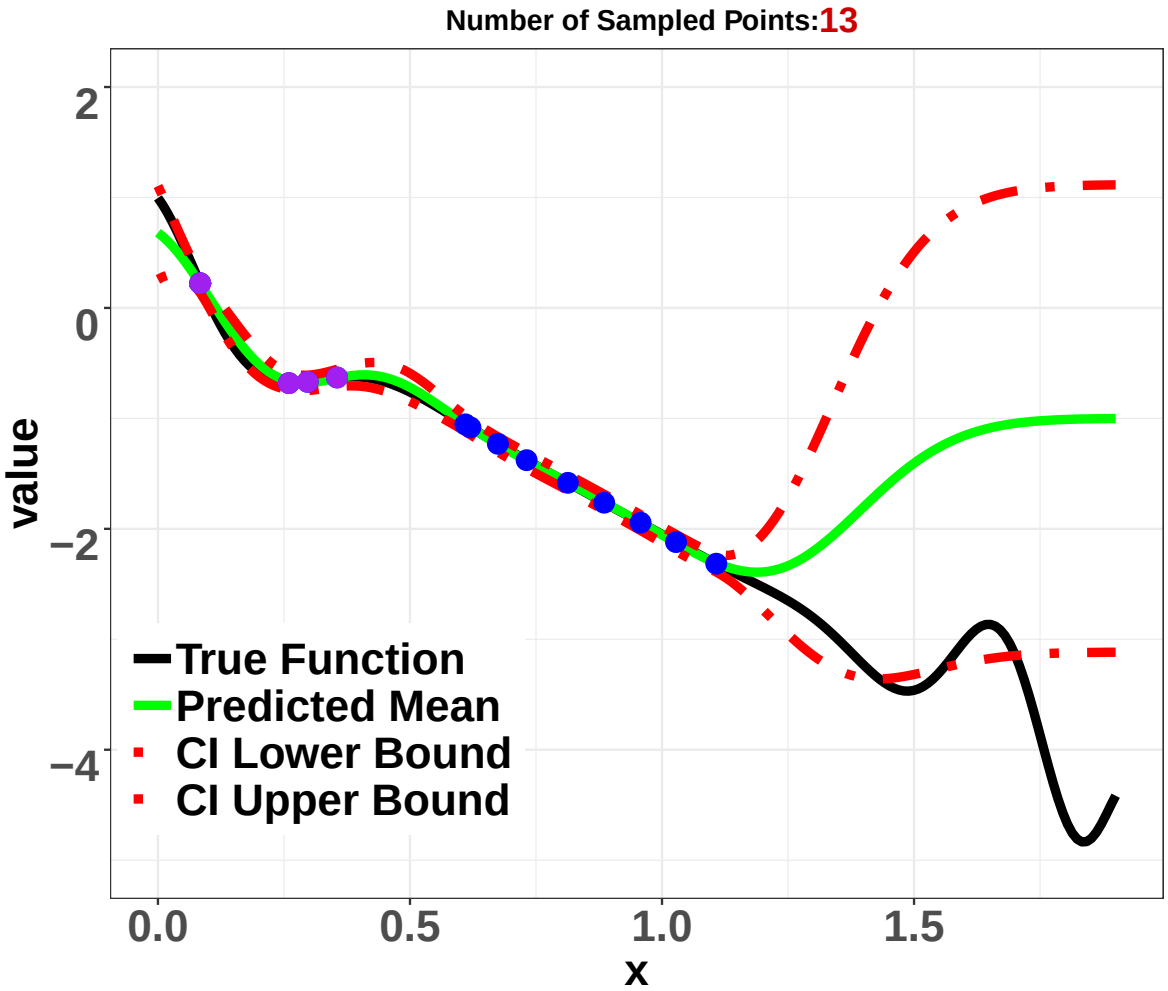
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

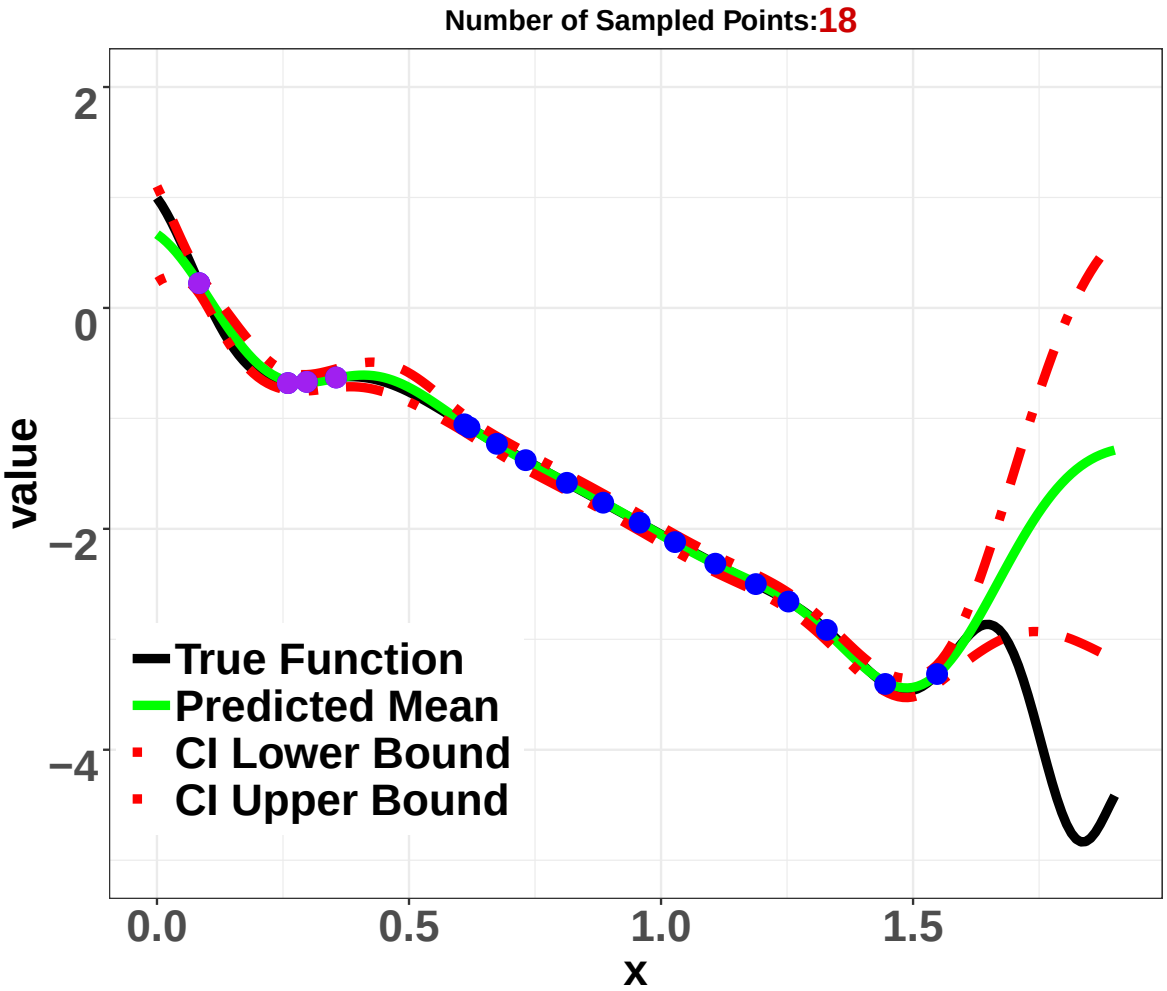
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

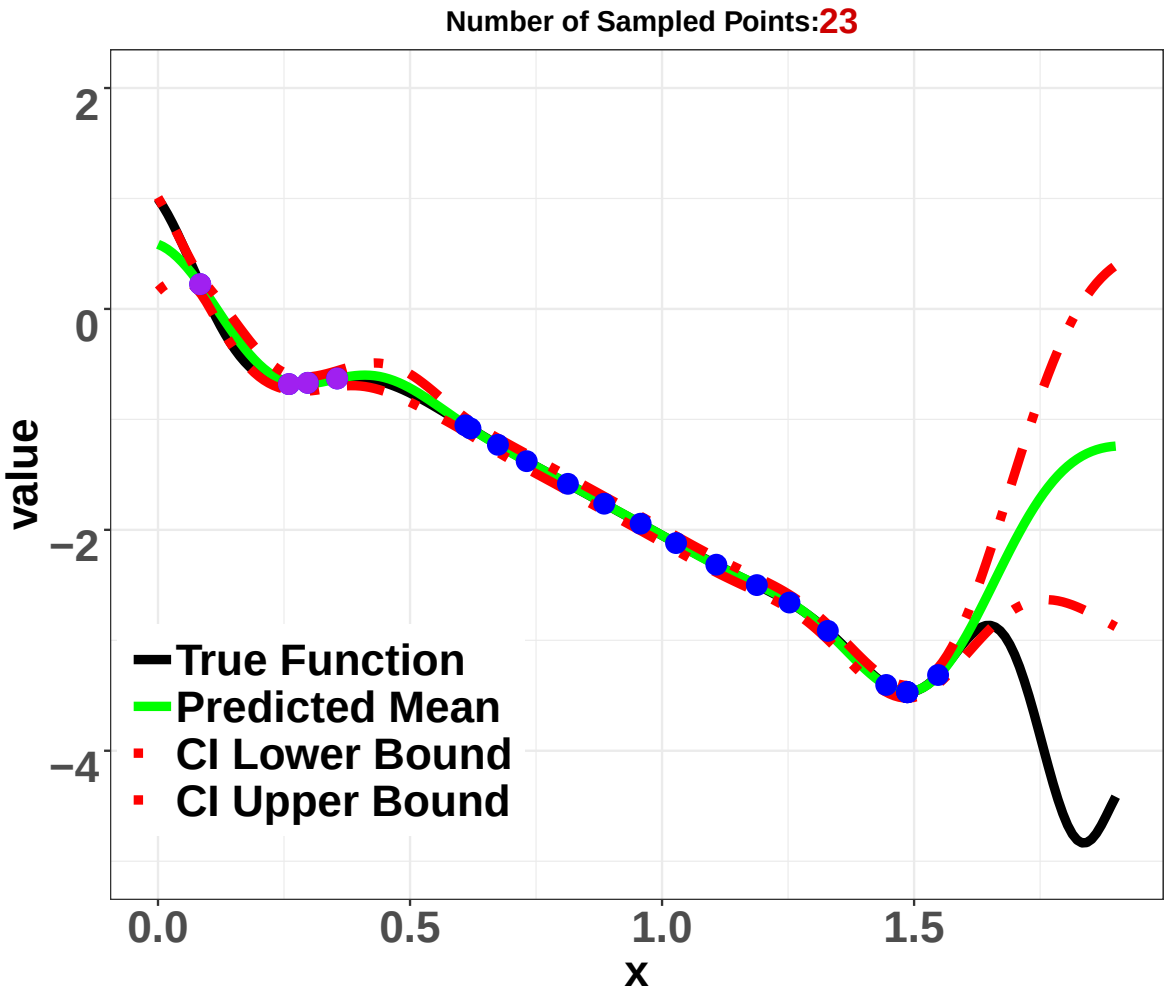
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

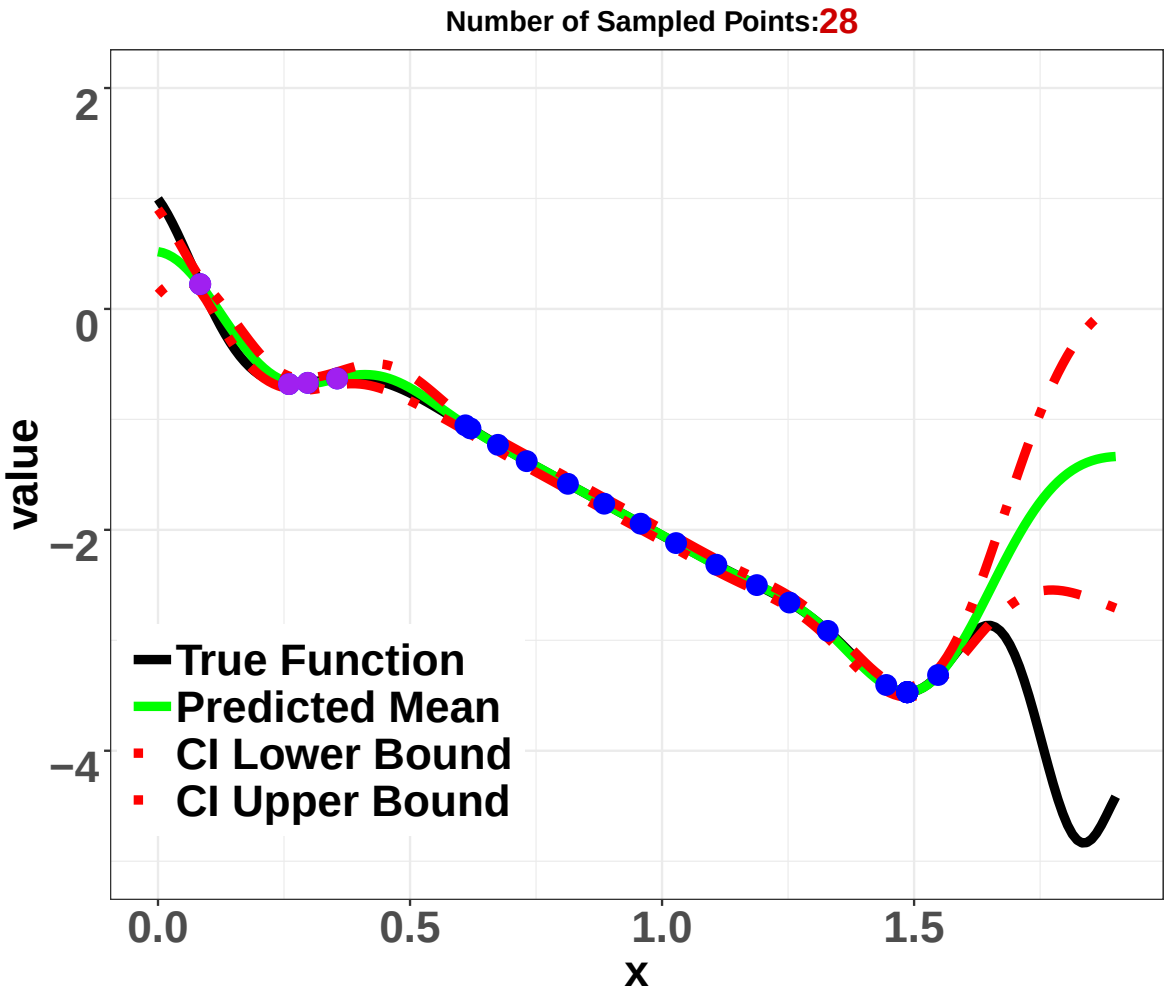
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

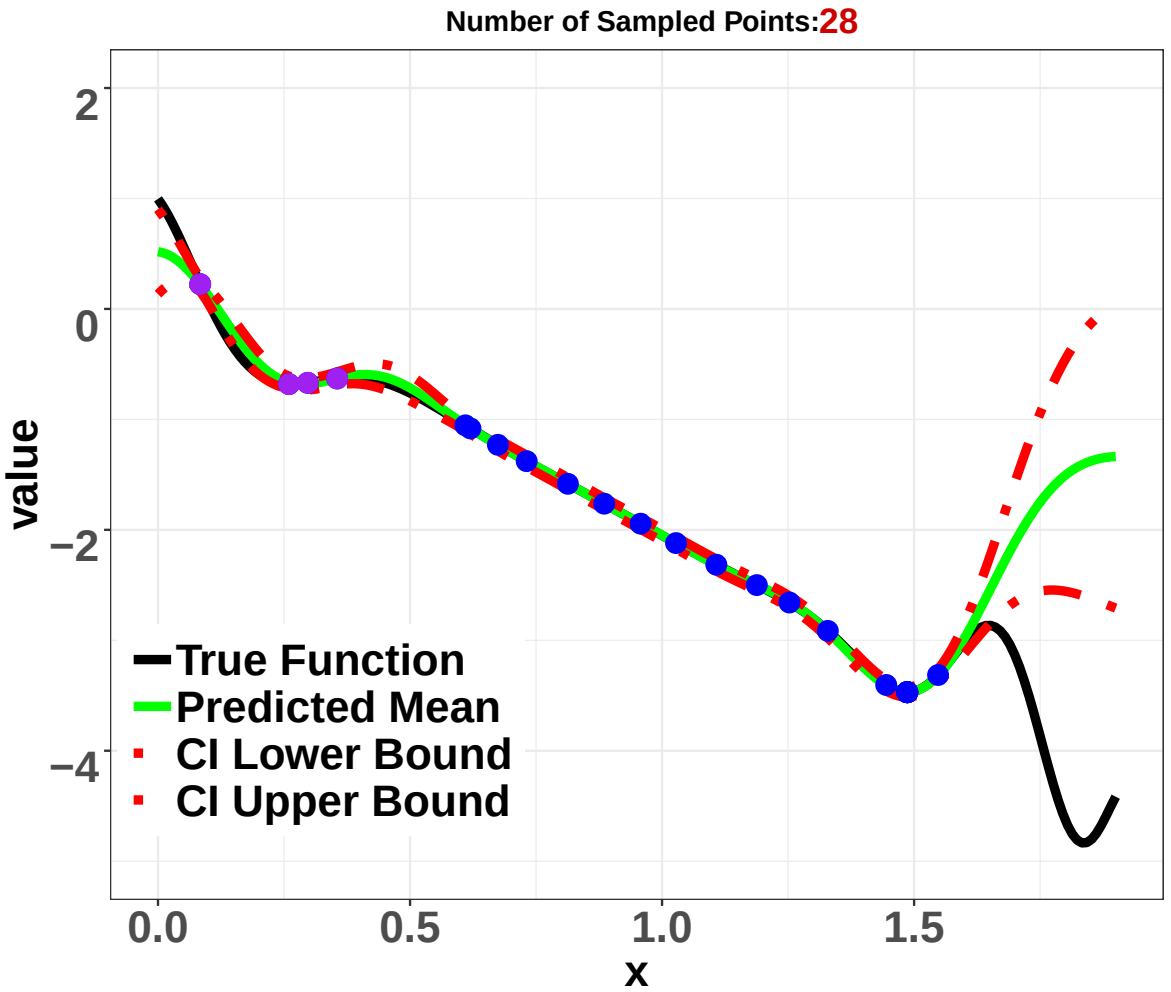
Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

Let's try this **plug-in EI** approach:



... plug-in EI gets **stuck** in local minima!

Over-exploitation

This **over-exploitation** of the **fitted GP model** for (plug-in) EI has been noted in recent literature:

- Qin et al. (2017): EI is too **greedy**, too much **exploitation** near the current best point
- Bull (2011): EI can **fail** to find the global optima!

Theorem (Bull, 2011)

Given any $\epsilon > 0$ and any random initial design F, there exist some $\delta > 0$ and some f in a certain space,

$$\mathbb{P}_F(\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) > \delta) \geq 1 - \epsilon,$$

where $(\boldsymbol{x}_n)_{n=1}^{\infty}$ is a solution sequence maximizing EI with plug-in MLEs $(\widehat{\beta}, \widehat{\sigma}^2)$.

Why? We're not factoring in **uncertainty** on the **plug-in MLEs**!

Over-exploitation

This **over-exploitation** of the **fitted GP model** for (plug-in) EI has been noted in recent literature:

- Qin et al. (2017): EI is too **greedy**, too much **exploitation** near the current best point
- Bull (2011): EI can **fail** to find the global optima!

Theorem (Bull, 2011)

Given any $\epsilon > 0$ and any random initial design F, there exist some $\delta > 0$ and some f in a certain space,

$$\mathbb{P}_F(\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) > \delta) \geq 1 - \epsilon,$$

where $(\boldsymbol{x}_n)_{n=1}^{\infty}$ is a solution sequence maximizing EI with plug-in MLEs $(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}^2)$.

Why? We're not factoring in **uncertainty** on the **plug-in MLEs**!

Solutions?

- Bull (2011): EI with **enlarged variance** and ϵ-**greedy** EI (ϵ-EI):
  - **Enlarged variance**: use $n\widehat{\sigma}_n^2$ instead of MLE $\widehat{\sigma}_n^2$
  - ϵ-**greedy**: with probability ϵ, **uniformly** select next point; with probability $1-\epsilon$, use **EI** with enlarged variance
  - Corrects **over-exploitation** by forcing EI to **explore** more
- Srinivas et al. (2010): Upper confidence bound (UCB) on GPs

$$\boldsymbol{x}_{\text{new}} = \arg\max_{\boldsymbol{x}} \left\{ \widehat{f}_n(\boldsymbol{x}) + \rho\widehat{\sigma} s_n(\boldsymbol{x}) \right\}, \quad \rho\text{: exploration parameter}$$

These methods encourage **more exploration**, but in *ad hoc* ways! Can we provide a **principled** solution to **over-exploitation** via a **hierarchical Bayesian** view on EI?

Solutions?

- Bull (2011): EI with **enlarged variance** and ϵ-**greedy** EI (ϵ-EI):
  - **Enlarged variance**: use $n\widehat{\sigma}_n^2$ instead of MLE $\widehat{\sigma}_n^2$
  - ϵ-**greedy**: with probability ϵ, **uniformly** select next point; with probability $1-\epsilon$, use **EI** with enlarged variance
  - Corrects **over-exploitation** by forcing EI to **explore** more
- Srinivas et al. (2010): Upper confidence bound (UCB) on GPs

$$\boldsymbol{x}_{\text{new}} = \arg\max_{\boldsymbol{x}} \left\{ \widehat{f}_n(\boldsymbol{x}) + \rho\widehat{\sigma} s_n(\boldsymbol{x}) \right\}, \quad \rho\text{: exploration parameter}$$

These methods encourage **more exploration**, but in *ad hoc* ways! Can we provide a **principled** solution to **over-exploitation** via a **hierarchical Bayesian** view on EI?

Hierarchical GP Model

Hierarchical GP (kriging) model (Handcock & Stein 1993):

$$\textbf{UK}: \quad f(\cdot) = \mu(\cdot) + Z(\cdot), \qquad Z(\cdot) \sim GP\{0, \sigma^2 r(\cdot,\cdot)\}$$

$$\textbf{Mean}: \quad \mu(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{\beta}, \qquad \textbf{Prior:} \quad [\boldsymbol{\beta}] \propto \mathbf{1}$$

$$\textbf{Covariance}: \quad \sigma^2 r(\cdot,\cdot), \qquad \textbf{Prior:} \quad [\sigma^2] \sim \text{Inv-Gamma}(a, b)$$

Still a **closed-form posterior** distribution (Santner et al., 2018):

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim T(2\widetilde{a}, \widehat{f}_n(\boldsymbol{x}), \widetilde{\sigma}^2 s^2(\boldsymbol{x})).$$

- $T(2\widetilde{a}, \cdot, \cdot)$: **Student** t-distribution with d.f. $2\widetilde{a}$
- $\widetilde{a} = a + (n-q)/2$, $\widetilde{b} = b + n\widehat{\sigma}^2/2$, $\widetilde{\sigma}^2 = \widetilde{b}/\widetilde{a}$

Hierarchical GP Model

Hierarchical GP (kriging) model (Handcock & Stein 1993):

$$\textbf{UK}: \quad f(\cdot) = \mu(\cdot) + Z(\cdot), \qquad Z(\cdot) \sim GP\{0, \sigma^2 r(\cdot,\cdot)\}$$

$$\textbf{Mean}: \quad \mu(\boldsymbol{x}) = \boldsymbol{p}^\top(\boldsymbol{x})\boldsymbol{\beta}, \qquad \textbf{Prior:} \quad [\boldsymbol{\beta}] \propto \mathbf{1}$$

$$\textbf{Covariance}: \quad \sigma^2 r(\cdot,\cdot), \qquad \textbf{Prior:} \quad [\sigma^2] \sim \text{Inv-Gamma}(a, b)$$

Still a **closed-form posterior** distribution (Santner et al., 2018):

$$\left[f(\boldsymbol{x})\middle|\mathcal{D}\right] \sim T(2\widetilde{a}, \widehat{f}_n(\boldsymbol{x}), \widetilde{\sigma}^2 s^2(\boldsymbol{x})).$$

- $T(2\widetilde{a}, \cdot, \cdot)$: **Student** t-distribution with d.f. $2\widetilde{a}$
- $\widetilde{a} = a + (n-q)/2$, $\widetilde{b} = b + n\widehat{\sigma}^2/2$, $\widetilde{\sigma}^2 = \widetilde{b}/\widetilde{a}$

Hierarchical Expected Improvement

Do we get a **closed-form** expression for **expected improvement** under this **hierarchical** GP model?

$$\begin{aligned}\mathrm{HEI}(\boldsymbol{x}) &= \mathbb{E}[\max(y^* - f(\boldsymbol{x}), 0)|\mathcal{D}] \\ &= I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}\, s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}\, s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}\, s_n(\boldsymbol{x})}\right)\end{aligned}$$

- ϕ_ν, Φ_ν: p.d.f. and c.d.f. of a t-**distribution** with d.f. ν

Why is this **important**?

- Function **queries** are obtained via **maximizing** $\mathrm{HEI}(\boldsymbol{x})$

$$\boldsymbol{x}_{\mathrm{new}} = \arg\max_{\boldsymbol{x}\in\Omega} \mathrm{HEI}(\boldsymbol{x})$$

- **Closed-form** $\mathrm{HEI}(\boldsymbol{x})$ $\Rightarrow$ **efficient** queries, **improved** optimization

Hierarchical Expected Improvement

Do we get a **closed-form** expression for **expected improvement** under this **hierarchical** GP model?

$$\begin{aligned}\mathrm{HEI}(\boldsymbol{x}) &= \mathbb{E}[\max(y^* - f(\boldsymbol{x}), 0)|\mathcal{D}] \\ &= I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right)\end{aligned}$$

- ϕ_ν, Φ_ν: p.d.f. and c.d.f. of a t-**distribution** with d.f. ν

Why is this **important**?

- Function **queries** are obtained via **maximizing** $\mathrm{HEI}(x)$

$$x_{\mathrm{new}} = \arg\max_{x\in\Omega} \mathrm{HEI}(x)$$

- **Closed-form** $\mathrm{HEI}(x) \Rightarrow$ **efficient** queries, **improved** optimization

Hierarchical Expected Improvement

Do we get a **closed-form** expression for **expected improvement** under this **hierarchical** GP model?

$$\begin{aligned}\mathrm{HEI}(\boldsymbol{x}) &= \mathbb{E}[\max(y^* - f(\boldsymbol{x}), 0)|\mathcal{D}] \\ &= I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right)\end{aligned}$$

- ϕ_ν, Φ_ν: p.d.f. and c.d.f. of a t-**distribution** with d.f. ν

Why is this **important**?

- Function **queries** are obtained via **maximizing** $\mathrm{HEI}(\boldsymbol{x})$

$$\boldsymbol{x}_{\mathrm{new}} = \arg\max_{\boldsymbol{x}\in\Omega} \mathrm{HEI}(\boldsymbol{x})$$

- **Closed-form** $\mathrm{HEI}(\boldsymbol{x}) \Rightarrow$ **efficient** queries, **improved** optimization

HEI vs. EI

How is HEI **different** from EI?

$$\mathsf{HEI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right),$$

vs.

$$\mathsf{EI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right) + \widehat{\sigma}s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right).$$

HEI encourages **more exploration**:

- **Finite-sample** correction $\kappa = \sqrt{\frac{\widetilde{a}}{\widetilde{a}-1}} > 1$
- $\widetilde{\sigma} > \widehat{\sigma}$ for weak hyperpriors on σ^2
- t distribution has **heavier** tails

HEI vs. EI

How is HEI **different** from EI?

$$\mathrm{HEI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right),$$

vs.

$$\mathrm{EI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right) + \widehat{\sigma}s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right).$$

HEI encourages **more exploration**:

- **Finite-sample** correction $\kappa = \sqrt{\frac{\widetilde{a}}{\widetilde{a}-1}} > 1$
- $\widetilde{\sigma} > \widehat{\sigma}$ for weak hyperpriors on σ^2
- t distribution has **heavier** tails

HEI vs. EI

How is HEI **different** from EI?

$$\mathrm{HEI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right),$$

vs.

$$\mathrm{EI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right) + \widehat{\sigma}s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right).$$

HEI encourages **more exploration**:

- **Finite-sample** correction $\kappa = \sqrt{\frac{\widetilde{a}}{\widetilde{a}-1}} > 1$
- $\widetilde{\sigma} > \widehat{\sigma}$ for weak hyperpriors on σ^2
- t distribution has **heavier** tails

HEI vs. EI

How is HEI **different** from EI?

$$\mathrm{HEI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right),$$

vs.

$$\mathrm{EI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right) + \widehat{\sigma}s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right).$$

HEI encourages **more exploration**:

- **Finite-sample** correction $\kappa = \sqrt{\frac{\widetilde{a}}{\widetilde{a}-1}} > 1$
- $\widetilde{\sigma} > \widehat{\sigma}$ for weak hyperpriors on σ^2
- t distribution has **heavier** tails

HEI vs. EI

How is HEI **different** from EI?

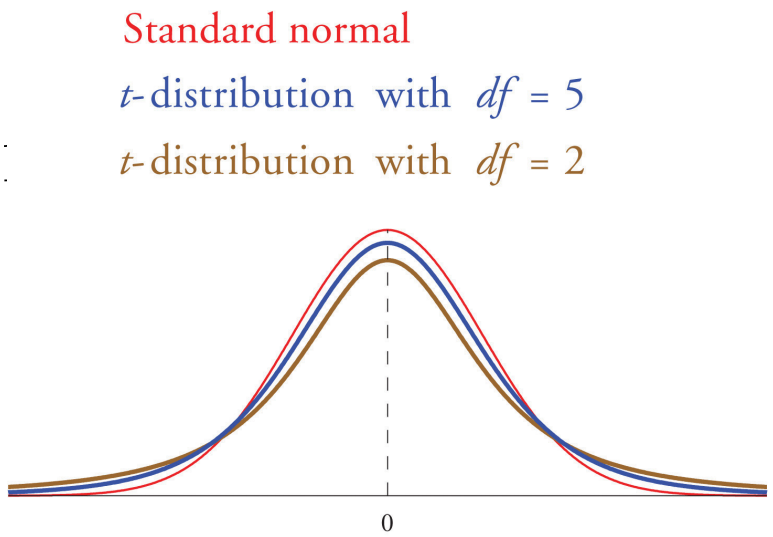
$$\mathrm{HEI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right),$$

vs.

$$\mathrm{EI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right) + \widehat{\sigma}s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right).$$

HEI encourages **more exploration**:

- **Finite-sample** correction $\kappa = \sqrt{\frac{\widetilde{a}}{\widetilde{a}-1}} >$
- $\widetilde{\sigma} > \widehat{\sigma}$ for weak hyperpriors on σ^2
- t distribution has **heavier** tails



HEI vs. EI

How is HEI **different** from EI?

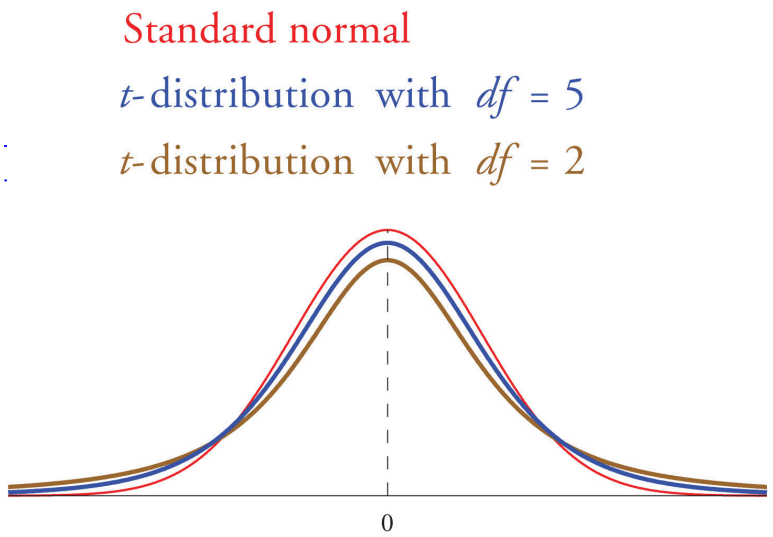
$$\mathrm{HEI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi_{2\widetilde{a}}\left(\frac{I(\boldsymbol{x})}{\widetilde{\sigma}s_n(\boldsymbol{x})}\right) + \kappa\widetilde{\sigma}s_n(\boldsymbol{x})\phi_{2\widetilde{a}-2}\left(\frac{I(\boldsymbol{x})}{\kappa\widetilde{\sigma}s_n(\boldsymbol{x})}\right),$$

vs.

$$\mathrm{EI}(\boldsymbol{x}) = I(\boldsymbol{x})\Phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right) + \widehat{\sigma}s_n(\boldsymbol{x})\phi\left(\frac{I(\boldsymbol{x})}{\widehat{\sigma}s_n(\boldsymbol{x})}\right).$$

HEI encourages **more exploration**:

- **Finite-sample** correction $\kappa = \sqrt{\frac{\widetilde{a}}{\widetilde{a}-1}} >$
- $\widetilde{\sigma} > \widehat{\sigma}$ for weak hyperpriors on σ^2
- t distribution has **heavier** tails



Hyperparameter specification

How to set **hyperparameters** a and b in $[\sigma^2] \sim$ Inv-Gamma(a, b)?

- **HEI-Weak: Weakly informative** (Gelman et al 2004)

  $a = b$ small, *e.g.*, 0.1

- **HEI-MMAP: Empirical Bayes** (EB, Robbins 1956)

  Estimate hyperprior parameters via marginal likelihood

- **HEI-DSD: Data-size dependent**

  From **global convergence theorem** (introduced later), set a constant and $b = \Theta(n) = cn$, where a and c are estimated via EB

Hyperparameter specification

How to set **hyperparameters** a and b in $[\sigma^2] \sim$ Inv-Gamma(a, b)?

- **HEI-Weak: Weakly informative** (Gelman et al 2004)
  $a = b$ small, *e.g.*, 0.1
- **HEI-MMAP: Empirical Bayes** (EB, Robbins 1956)
  Estimate hyperprior parameters via marginal likelihood
- **HEI-DSD: Data-size dependent**
  From **global convergence theorem** (introduced later), set a constant and $b = \Theta(n) = cn$, where a and c are estimated via EB

Hyperparameter specification

How to set **hyperparameters** a and b in $[\sigma^2] \sim$ Inv-Gamma(a, b)?

- **HEI-Weak: Weakly informative** (Gelman et al 2004)
  $a = b$ small, *e.g.*, 0.1
- **HEI-MMAP: Empirical Bayes** (EB, Robbins 1956)
  Estimate hyperprior parameters via marginal likelihood
- **HEI-DSD: Data-size dependent**
  From **global convergence theorem** (introduced later), set a constant and $b = \Theta(n) = cn$, where a and c are estimated via EB

Hyperparameter specification

How to set **hyperparameters** a and b in $[\sigma^2] \sim$ Inv-Gamma(a, b)?

- **HEI-Weak: Weakly informative** (Gelman et al 2004)

  $a = b$ small, *e.g.*, 0.1

- **HEI-MMAP: Empirical Bayes** (EB, Robbins 1956)

  Estimate hyperprior parameters via marginal likelihood

- **HEI-DSD: Data-size dependent**

  From **global convergence theorem** (introduced later), set a constant and $b = \Theta(n) = cn$, where a and c are estimated via EB

Empirical Bayes

- EB approximates a **fully Bayesian hierarchical model** by estimating hyperparameters via **marginal likelihood** maximization (Carlin & Louis 2000)

- **Here**, find (a^*, b^*) which **maximizes**:

$$(a^*, b^*) = \arg\max_{a,b} \int p(y|\beta, \sigma^2)\pi(\beta)\pi(\sigma^2|a, b)\, d\beta d\sigma^2$$

- ... unfortunately this problem is **unbounded!**

Empirical Bayes

- EB approximates a **fully Bayesian hierarchical model** by estimating hyperparameters via **marginal likelihood** maximization (Carlin & Louis 2000)

- **Here**, find (a^*, b^*) which **maximizes**:

$$(a^*, b^*) = \arg\max_{a,b} \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta})\pi(\sigma^2|a, b)\, d\boldsymbol{\beta} d\sigma^2$$

... unfortunately this problem is **unbounded**!

Empirical Bayes

- EB approximates a **fully Bayesian hierarchical model** by estimating hyperparameters via **marginal likelihood** maximization (Carlin & Louis 2000)

- **Here**, find (a^*, b^*) which **maximizes**:

$$(a^*, b^*) = \arg\max_{a,b} \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta})\pi(\sigma^2|a, b)\, d\boldsymbol{\beta} d\sigma^2$$

... unfortunately this problem is **unbounded**!

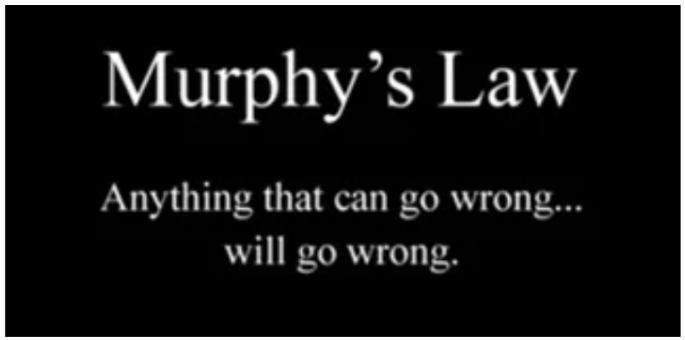
Empirical Bayes

- EB approximates a **fully Bayesian hierarchical model** by estimating hyperparameters via **marginal likelihood** maximization (Carlin & Louis 2000)

- **Here**, find (a^*, b^*) which **maximizes**:

$$(a^*, b^*) = \arg\max_{a,b} \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta})\pi(\sigma^2|a, b)\, d\boldsymbol{\beta} d\sigma^2$$

... unfortunately this problem is **unbounded**!



Empirical Bayes

- **Marginal maximum a posteriori** (MMAP, Doucet et al. 2002): add **another** layer of hyperpriors on (a, b) for **regularization**

Proposition

Assume the following independent hyperpriors:

$$[a] \sim \text{Gamma}(\zeta, \iota), \quad [b] \propto \mathbf{1}.$$

Then the MMAP estimator

$$(a^*, b^*) = \arg\max_{a,b} \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta})\pi(\sigma^2|a, b)\, d\boldsymbol{\beta} d\sigma^2 \pi(a, b)$$

exists and is **finite**.

- We can then use the estimated prior $\sigma^2 \sim IG(a^*, b^*)$ with MMAP estimates (a^*, b^*) within HEI
- **Key**: HEI with EB approximates a **fully Bayesian** EI procedure, without need for **expensive** MCMC (all **closed**-form!)

Empirical Bayes

- **Marginal maximum a posteriori** (MMAP, Doucet et al. 2002): add **another** layer of hyperpriors on (a, b) for **regularization**

Proposition

Assume the following independent hyperpriors:

$$[a] \sim \text{Gamma}(\zeta, \iota), \quad [b] \propto \mathbf{1}.$$

Then the MMAP estimator

$$(a^*, b^*) = \arg\max_{a,b} \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta})\pi(\sigma^2|a, b)\, d\boldsymbol{\beta} d\sigma^2 \pi(a, b)$$

exists and is **finite**.

- We can then use the estimated prior $\sigma^2 \sim IG(a^*, b^*)$ with MMAP estimates (a^*, b^*) within HEI
- **Key**: HEI with EB approximates a **fully Bayesian** EI procedure, without need for **expensive** MCMC (all **closed**-form!)

Empirical Bayes

- **Marginal maximum a posteriori** (MMAP, Doucet et al. 2002): add **another** layer of hyperpriors on (a, b) for **regularization**

Proposition

Assume the following independent hyperpriors:

$$[a] \sim \text{Gamma}(\zeta, \iota), \quad [b] \propto \mathbf{1}.$$

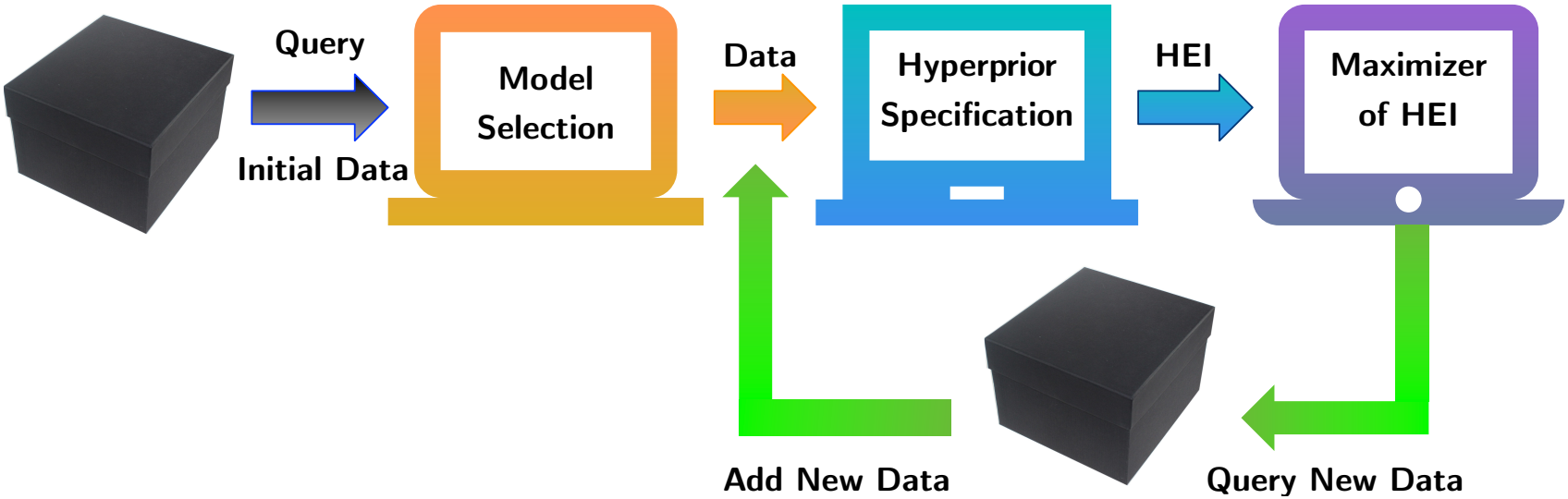
Then the MMAP estimator

$$(a^*, b^*) = \arg\max_{a,b} \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta})\pi(\sigma^2|a, b)\, d\boldsymbol{\beta} d\sigma^2 \pi(a, b)$$

exists and is **finite**.

- We can then use the estimated prior $\sigma^2 \sim IG(a^*, b^*)$ with MMAP estimates (a^*, b^*) within HEI
- **Key**: HEI with EB approximates a **fully Bayesian** EI procedure, without need for **expensive** MCMC (all **closed**-form!)

Algorithm


Query
Initial Data
Model Selection
Data
Hyperprior Specification
HEI
Maximizer of HEI
Add New Data
Query New Data

Algorithm

Algorithm 1 Hierarchical Expected Improvement for Bayesian Optimization

Initialization

- Generate n_{ini} space-filling design points $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n_{\text{ini}}}\}$ on Ω.
- Evaluate function points $y_i = f(\mathbf{x}_i)$, yielding the initial dataset $\mathcal{D}_{n_{\text{ini}}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_{\text{ini}}}$.

Model selection

- Select model order via BIC using (21).
- Estimate hyperparameters (a, b) via MMAP using (18).

Optimization

for $n \leftarrow n_{\text{ini}}$ *to* $n_{\text{tot}} - 1$ **do**

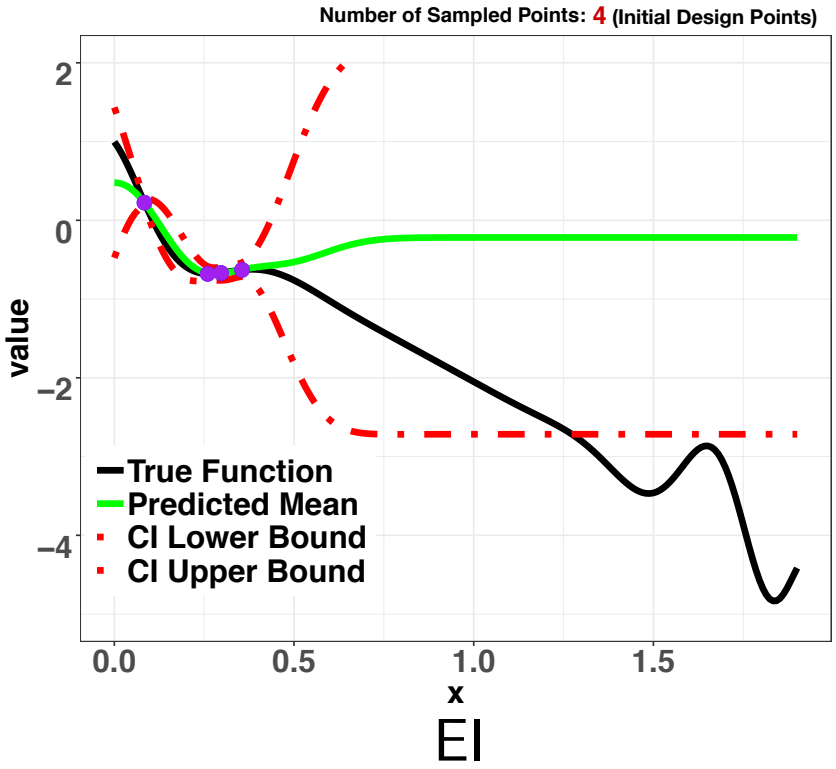
- Given $\mathcal{D}_n$, estimate length-scale parameters $\boldsymbol{\theta}$ via MAP and compute $\text{HEI}_n(\mathbf{x})$.
- Obtain the next evaluation point $\mathbf{x}_{n+1}$ by maximizing $\text{HEI}_n(\mathbf{x})$:

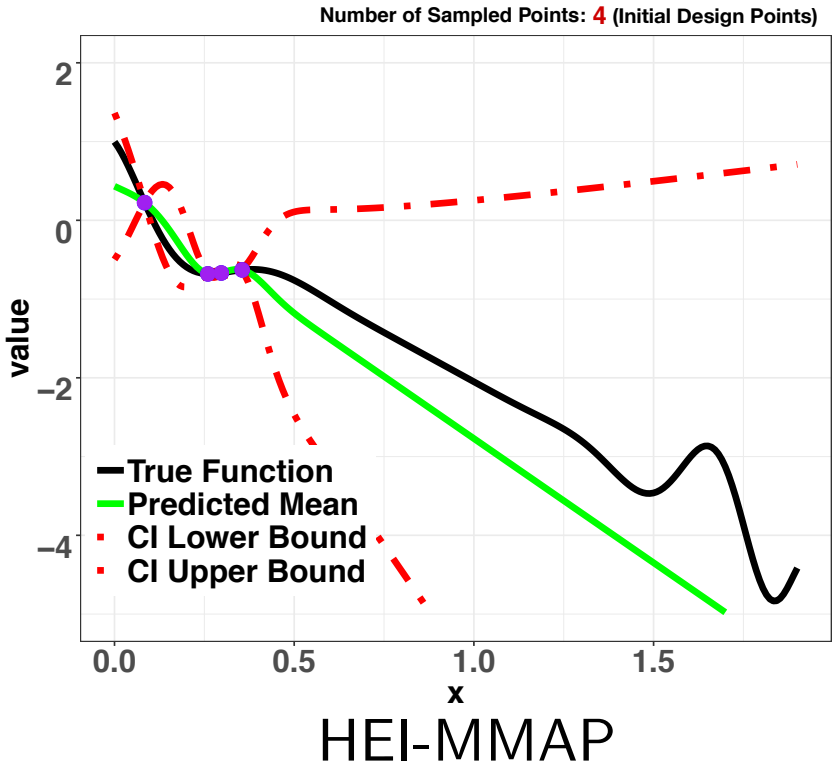
$$\mathbf{x}_{n+1} \leftarrow \underset{\mathbf{x} \in \Omega}{\text{argmax}}\, \text{HEI}_n(\mathbf{x}).$$

- Evaluate $y_{n+1} = f(\mathbf{x}_{n+1})$, and update data $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$.

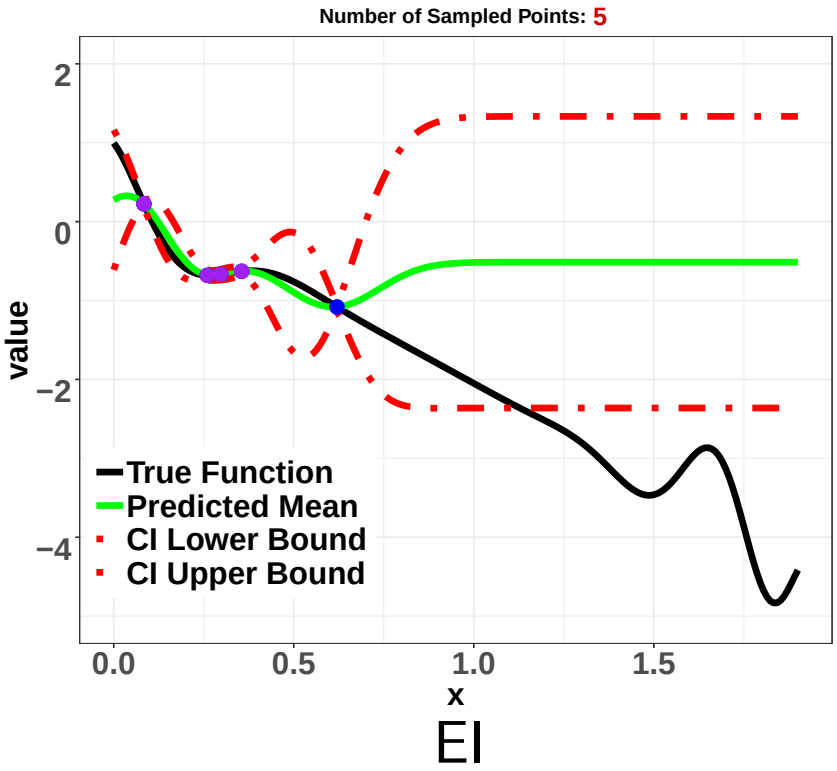
Return: The best observed solution $\mathbf{x}_{i^*}$, where $i^* = \text{argmin}_{i=1}^{n_{\text{tot}}} f(\mathbf{x}_i)$.

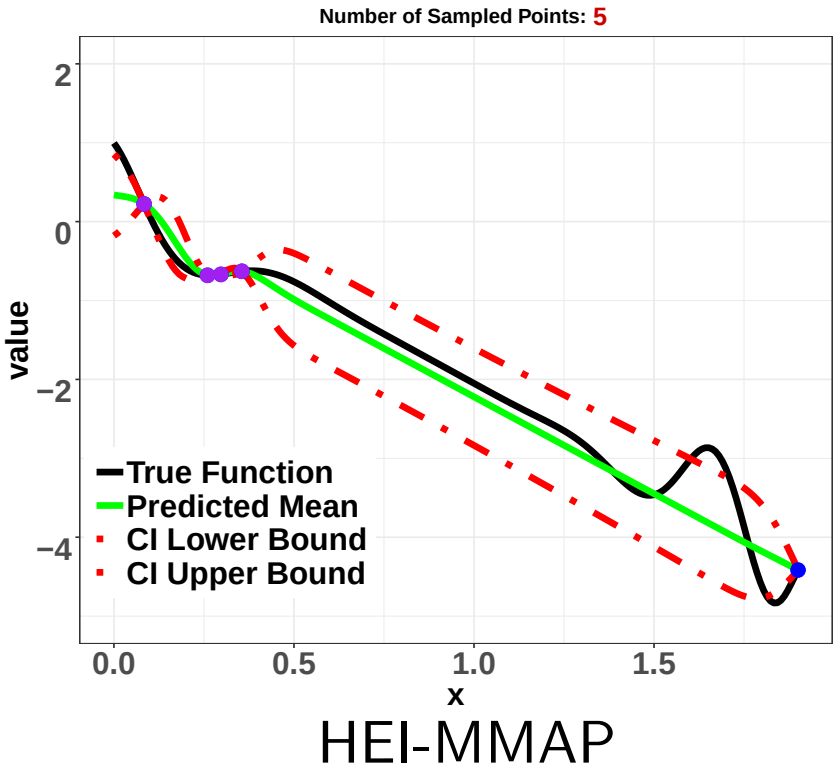
Back to example



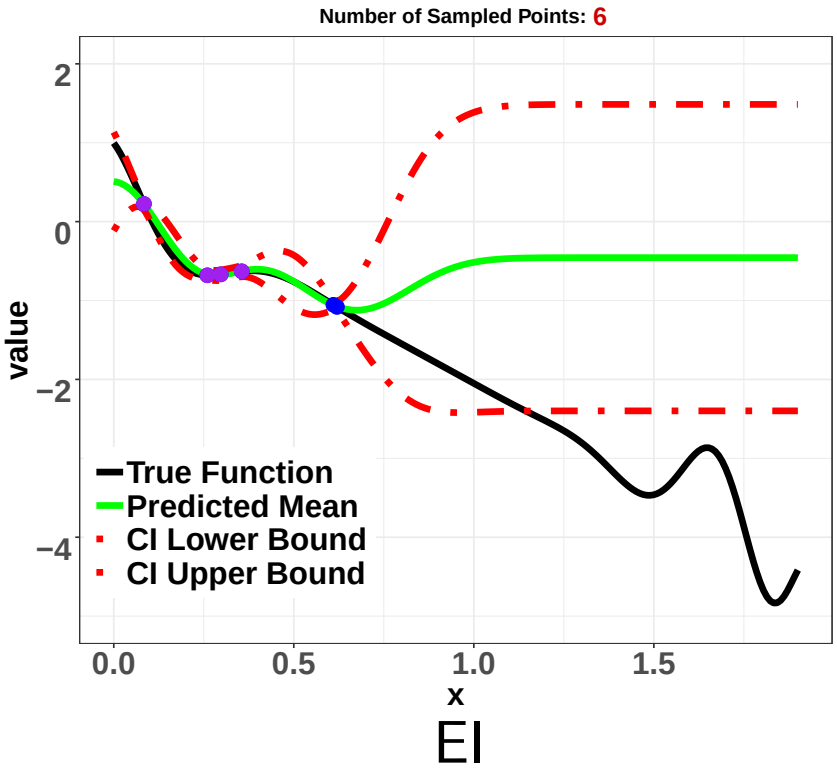


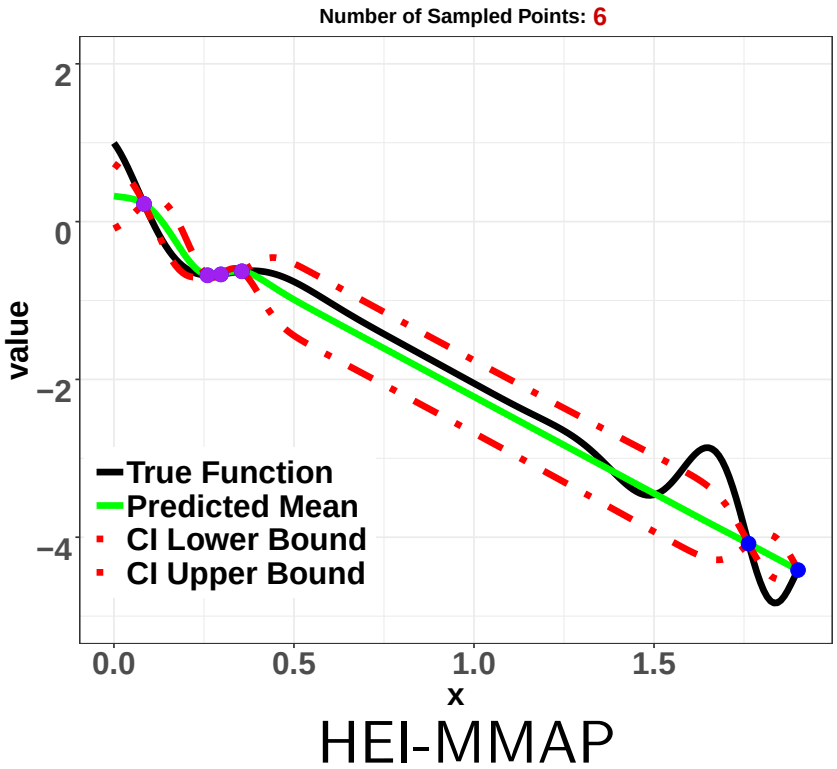
Back to example



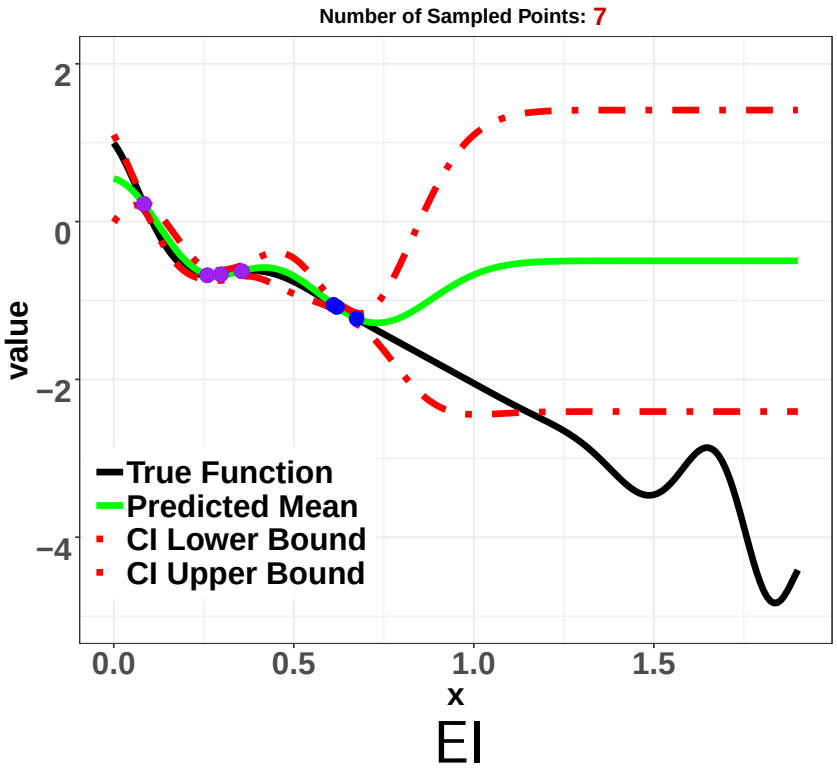


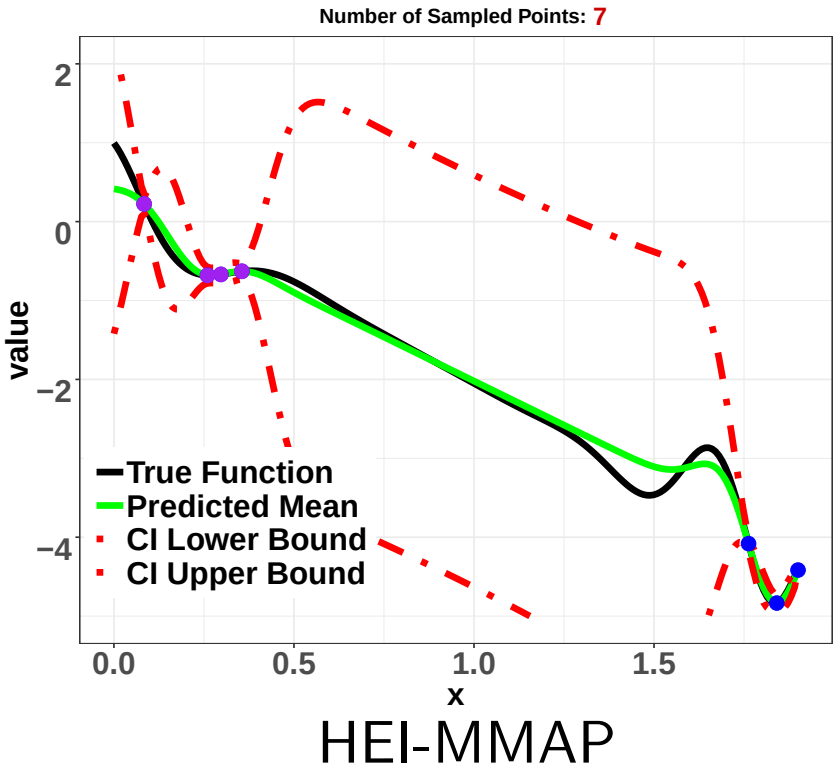
Back to example



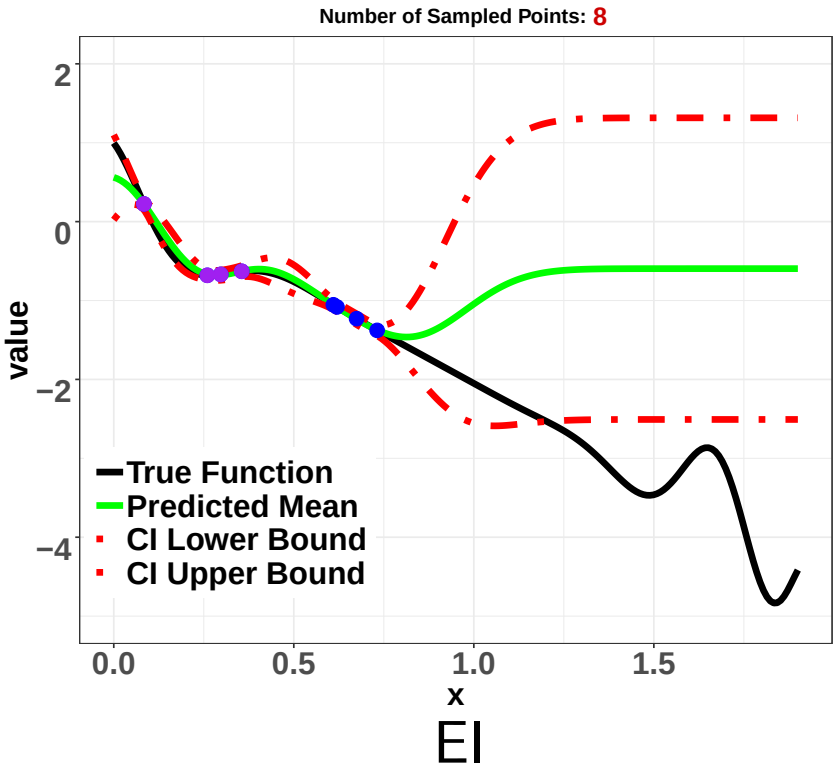


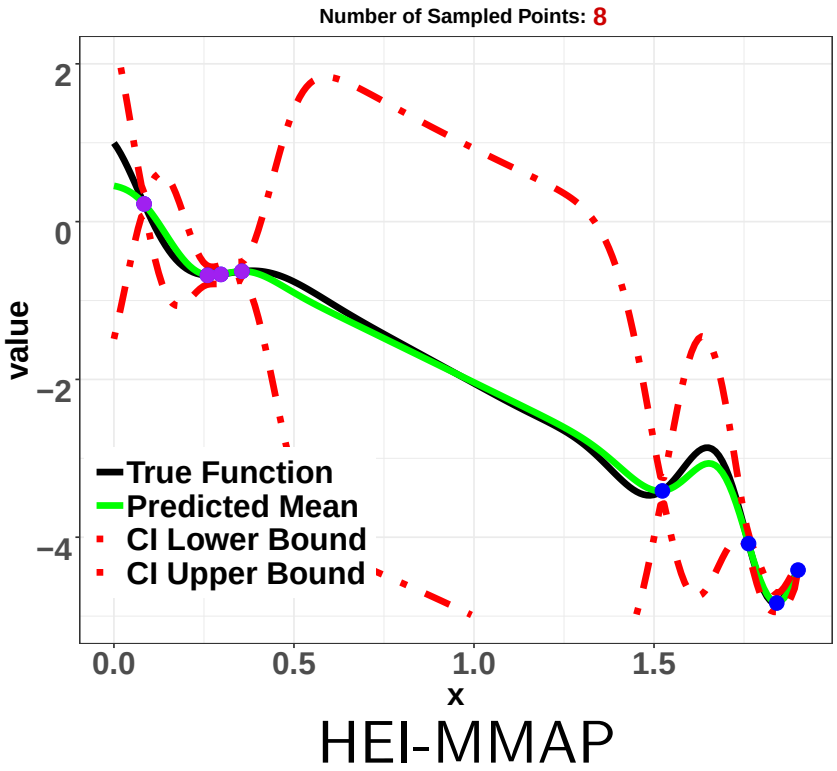
Back to example



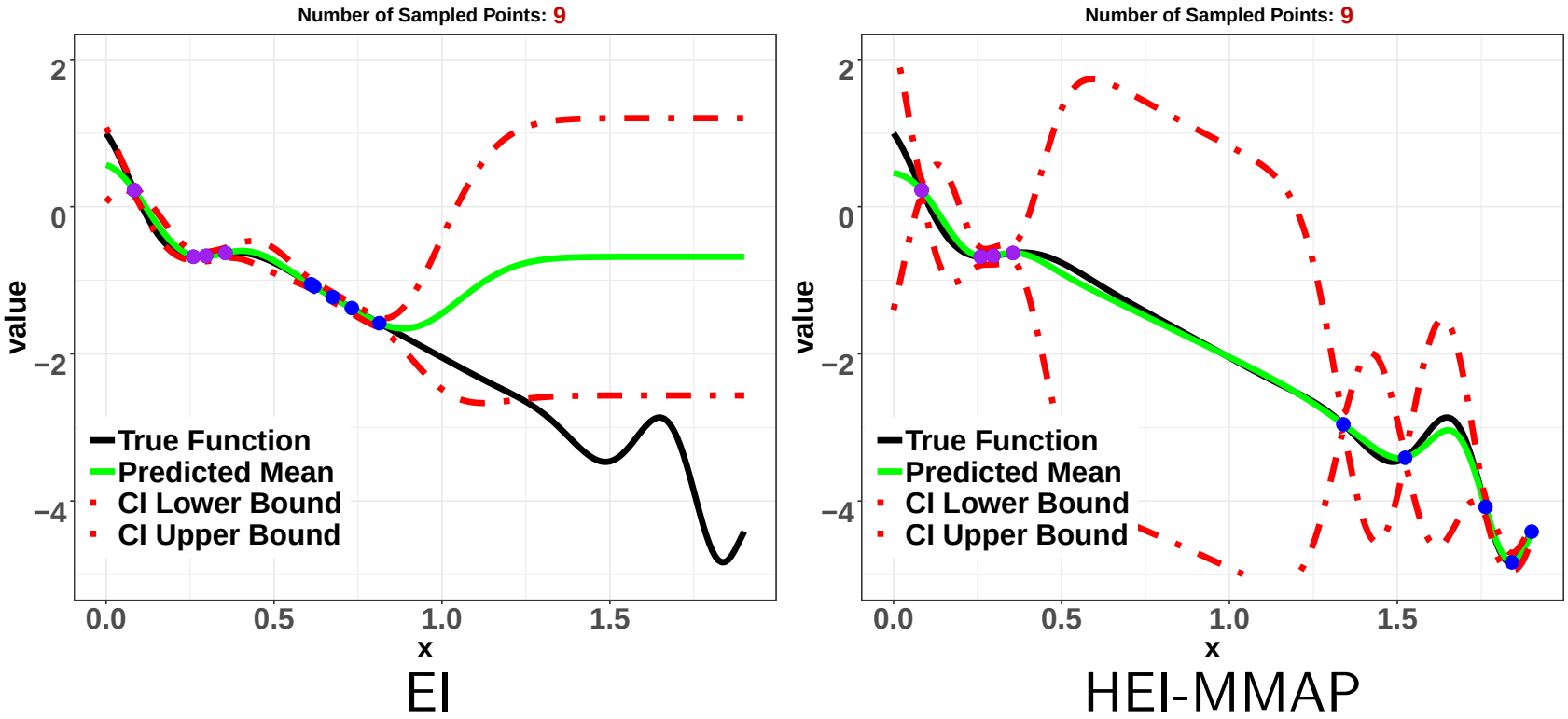


Back to example

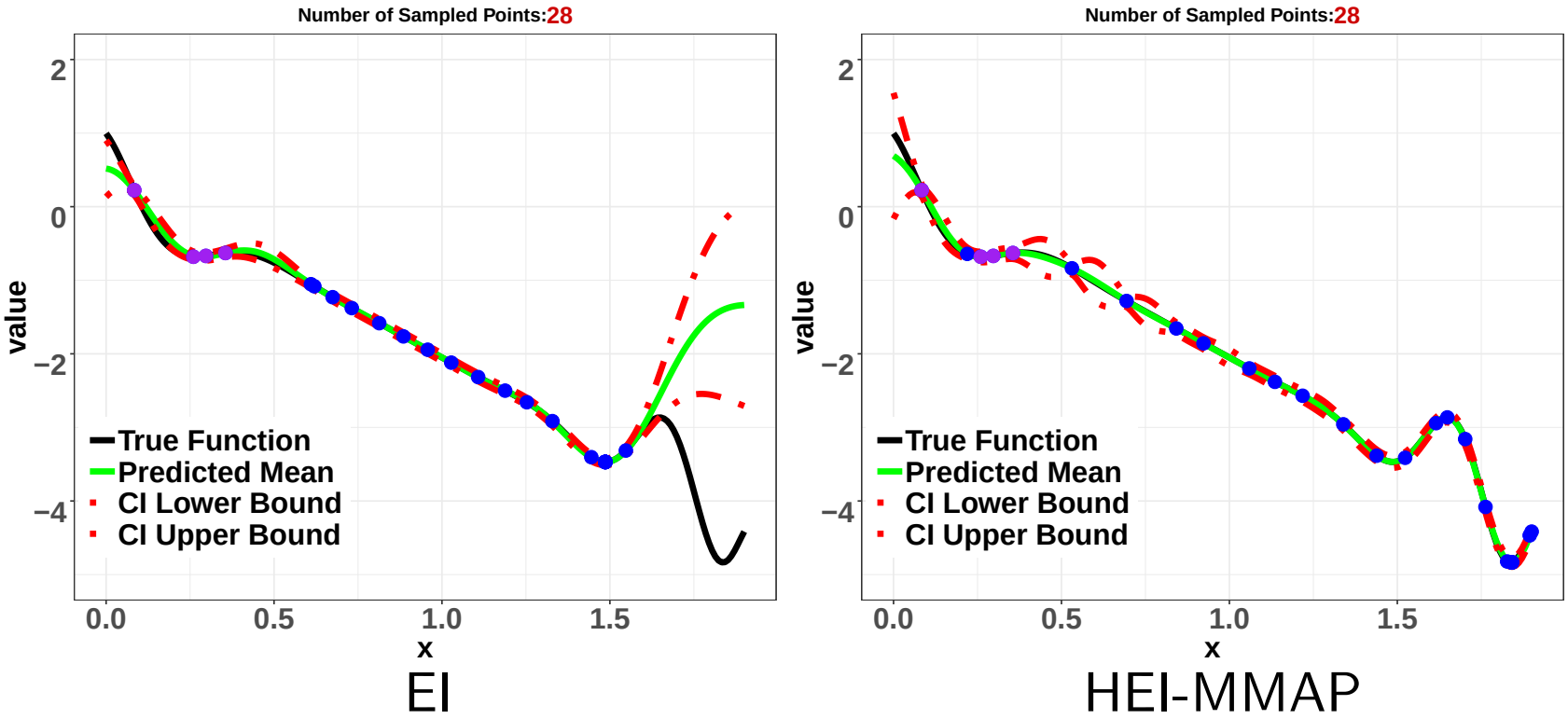




Back to example



Back to example



- EI gets **stuck** at **local optimum**, HEI **finds** the **global optimum**

Global convergence

Theorem (Chen, Mak, Wu 2023)

Suppose a is a constant, $b = \Theta(n)$, and correlation $r(\boldsymbol{x})$ satisfies

$$|r(\boldsymbol{x}) - Q_{\lfloor 2\nu \rfloor}(\boldsymbol{x})| = \mathcal{O}\big(\|\boldsymbol{x}\|^{2\nu}(-\log\|\boldsymbol{x}\|)^{2\alpha}\big), \quad \text{as } \|\boldsymbol{x}\| \to 0,$$

where $Q_{\lfloor 2\nu \rfloor}(\boldsymbol{x})$ is a polynomial of degree $\lfloor 2\nu \rfloor$. Then for any f in the RKHS $\mathcal{H}_r(\Omega)$, we have:

$$\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) = \begin{cases} \mathcal{O}(n^{-\nu/d}(\log n)^{\alpha}), & \nu \le 1 \\ \mathcal{O}(n^{-1/d}), & \nu > 1 \end{cases},$$

where $(\boldsymbol{x}_n)_{n=1}^{\infty}$ is the solution sequence by maximizing HEI.

- As sample size $n \to \infty$, $\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) \to 0$
- This shows that, unlike **plug-in EI**, HEI is **guaranteed** to converge to a **global minima**

Global convergence

Theorem (Chen, Mak, Wu 2023)

Suppose a is a constant, $b = \Theta(n)$, and correlation $r(\boldsymbol{x})$ satisfies

$$|r(\boldsymbol{x}) - Q_{\lfloor 2\nu \rfloor}(\boldsymbol{x})| = \mathcal{O}\big(\|\boldsymbol{x}\|^{2\nu}(-\log\|\boldsymbol{x}\|)^{2\alpha}\big), \quad \text{as } \|\boldsymbol{x}\| \to 0,$$

where $Q_{\lfloor 2\nu \rfloor}(\boldsymbol{x})$ is a polynomial of degree $\lfloor 2\nu \rfloor$. Then for any f in the RKHS $\mathcal{H}_r(\Omega)$, we have:

$$\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) = \begin{cases} \mathcal{O}(n^{-\nu/d}(\log n)^{\alpha}), & \nu \le 1 \\ \mathcal{O}(n^{-1/d}), & \nu > 1 \end{cases},$$

where $(\boldsymbol{x}_n)_{n=1}^{\infty}$ is the solution sequence by maximizing HEI.

- As sample size $n \to \infty$, $\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) \to 0$
- This shows that, unlike **plug-in EI**, HEI is **guaranteed** to converge to a **global minima**

Global convergence

Theorem (Chen, Mak, Wu 2023)

Suppose a is a constant, $b = \Theta(n)$, and correlation $r(\boldsymbol{x})$ satisfies

$$|r(\boldsymbol{x}) - Q_{\lfloor 2\nu \rfloor}(\boldsymbol{x})| = \mathcal{O}\big(\|\boldsymbol{x}\|^{2\nu}(-\log\|\boldsymbol{x}\|)^{2\alpha}\big), \quad \text{as } \|\boldsymbol{x}\| \to 0,$$

where $Q_{\lfloor 2\nu \rfloor}(\boldsymbol{x})$ is a polynomial of degree $\lfloor 2\nu \rfloor$. Then for any f in the RKHS $\mathcal{H}_r(\Omega)$, we have:

$$\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) = \begin{cases} \mathcal{O}(n^{-\nu/d}(\log n)^{\alpha}), & \nu \le 1 \\ \mathcal{O}(n^{-1/d}), & \nu > 1 \end{cases},$$

where $(\boldsymbol{x}_n)_{n=1}^{\infty}$ is the solution sequence by maximizing HEI.

- Theory requires a **data size dependent** hyperprior on $\sigma^2 \sim IG(a, b)$, namely $b = \Theta(n)$
- **Matérn correlation** (Stein 2012) satisfies the conditions on $r(\boldsymbol{x})$

Minimax rate

γ-**stability** condition (Wynne et al. 2020):

$$s_n(\boldsymbol{x}_{n+1}) \geq \gamma \|s_n(\boldsymbol{x})\|_\infty \quad \text{for all } n = 1, 2, \cdots, \tag{1}$$

- $s_n(\boldsymbol{x})$: **posterior variance** term from GP

With this, we can then show the following optimization rate:

Theorem (Chen, Mak, Wu 2023)

Assume the earlier conditions hold. Then for any $f \in \mathcal{H}_r(\Omega)$:

$$\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) = O(n^{-\nu/d}),$$

where $(\boldsymbol{x}_n)_{n=1}^\infty$ is the solution sequence maximizing HEI with the γ-stability condition, with $\gamma \in (0, 1)$.

This is precisely the **minimax** rate for $f \in \mathcal{H}_r(\Omega)$ (Bull 2011)!

Minimax rate

γ-**stability** condition (Wynne et al. 2020):

$$s_n(\boldsymbol{x}_{n+1}) \geq \gamma \|s_n(\boldsymbol{x})\|_\infty \quad \text{for all } n = 1, 2, \cdots, \tag{1}$$

- $s_n(\boldsymbol{x})$: **posterior variance** term from GP

With this, we can then show the following optimization rate:

Theorem (Chen, Mak, Wu 2023)

Assume the earlier conditions hold. Then for any $f \in \mathcal{H}_r(\Omega)$:

$$\min_n f(\boldsymbol{x}_n) - \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) = O(n^{-\nu/d}),$$

where $(\boldsymbol{x}_n)_{n=1}^\infty$ is the solution sequence maximizing HEI with the γ-stability condition, with $\gamma \in (0, 1)$.

This is precisely the **minimax** rate for $f \in \mathcal{H}_r(\Omega)$ (Bull 2011)!

Minimax rate

Algorithm 1 Hierarchical Expected Improvement for Bayesian Optimization

Initialization

- Generate n_{ini} space-filling design points $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n_{\text{ini}}}\}$ on Ω.
- Evaluate function points $y_i = f(\mathbf{x}_i)$, yielding the initial dataset $\mathcal{D}_{n_{\text{ini}}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_{\text{ini}}}$.

Model selection

- Select model order via BIC using (21).
- Estimate hyperparameters (a, b) via MMAP using (18).

Optimization

for $n \leftarrow n_{\text{ini}}$ *to* $n_{\text{tot}} - 1$ **do**

- Given $\mathcal{D}_n$, estimate length-scale parameters $\boldsymbol{\theta}$ via MAP and compute $\text{HEI}_n(\mathbf{x})$.
- Obtain the next evaluation point $\mathbf{x}_{n+1}$ by maximizing $\text{HEI}_n(\mathbf{x})$:

Add in stability condition as constraint

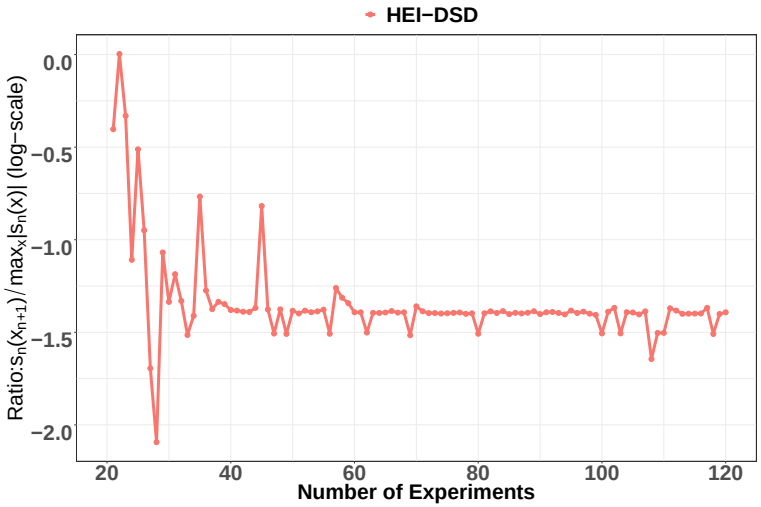
$$\mathbf{x}_{n+1} \leftarrow \underset{\mathbf{x} \in \Omega}{\text{argmax}}\, \text{HEI}_n(\mathbf{x}).$$

- Evaluate $y_{n+1} = f(\mathbf{x}_{n+1})$, and update data $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$.

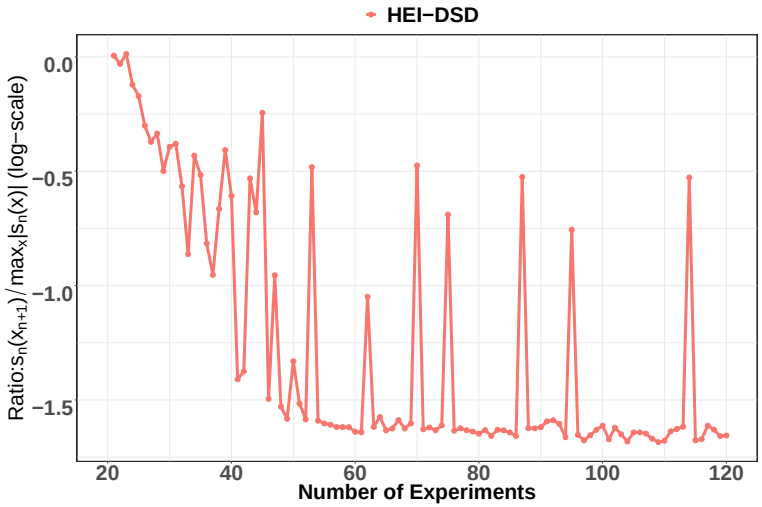
Return: The best observed solution $\mathbf{x}_{i^*}$, where $i^* = \text{argmin}_{i=1}^{n_{\text{tot}}} f(\mathbf{x}_i)$.

Minimax rate

... but in practice, this γ-**stability condition** seems to **hold** without an **explicit** constraint on the HEI



Branin



Camel Three-Hump

Minimax rate

Why is our rate $O(n^{-\nu/d})$ **minimax**?

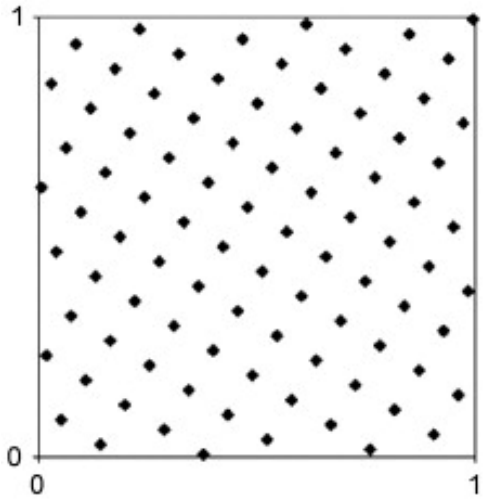
- Bull (2011): the minimax rate is $O(n^{-\nu/d})$ for $f \in \mathcal{H}_r(\Omega)$
- This rate, however, can be achieved by a **non-adaptive quasi-uniform** strategy, which performs **terribly** in practice!

Rates don't tell the whole story!

Minimax rate

Why is our rate $O(n^{-\nu/d})$ **minimax**?

- Bull (2011): the minimax rate is $O(n^{-\nu/d})$ for $f \in \mathcal{H}_r(\Omega)$
- This rate, however, can be achieved by a **non-adaptive quasi-uniform** strategy, which performs **terribly** in practice!

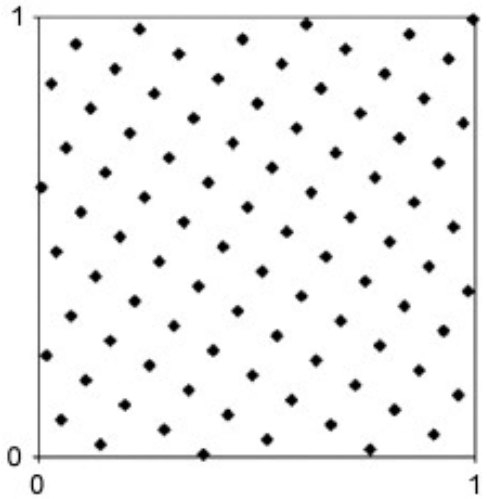


Rates don't tell the whole story!

Minimax rate

Why is our rate $O(n^{-\nu/d})$ **minimax**?

- Bull (2011): the minimax rate is $O(n^{-\nu/d})$ for $f \in \mathcal{H}_r(\Omega)$
- This rate, however, can be achieved by a **non-adaptive quasi-uniform** strategy, which performs **terribly** in practice!



Rates don't tell the whole story!

Test functions

HEI methods:

- HEI-Weak, HEI-MMAP, HEI-DSD

Plug-in EI methods (Jones et al 1997):

- EI-OK, EI-UK

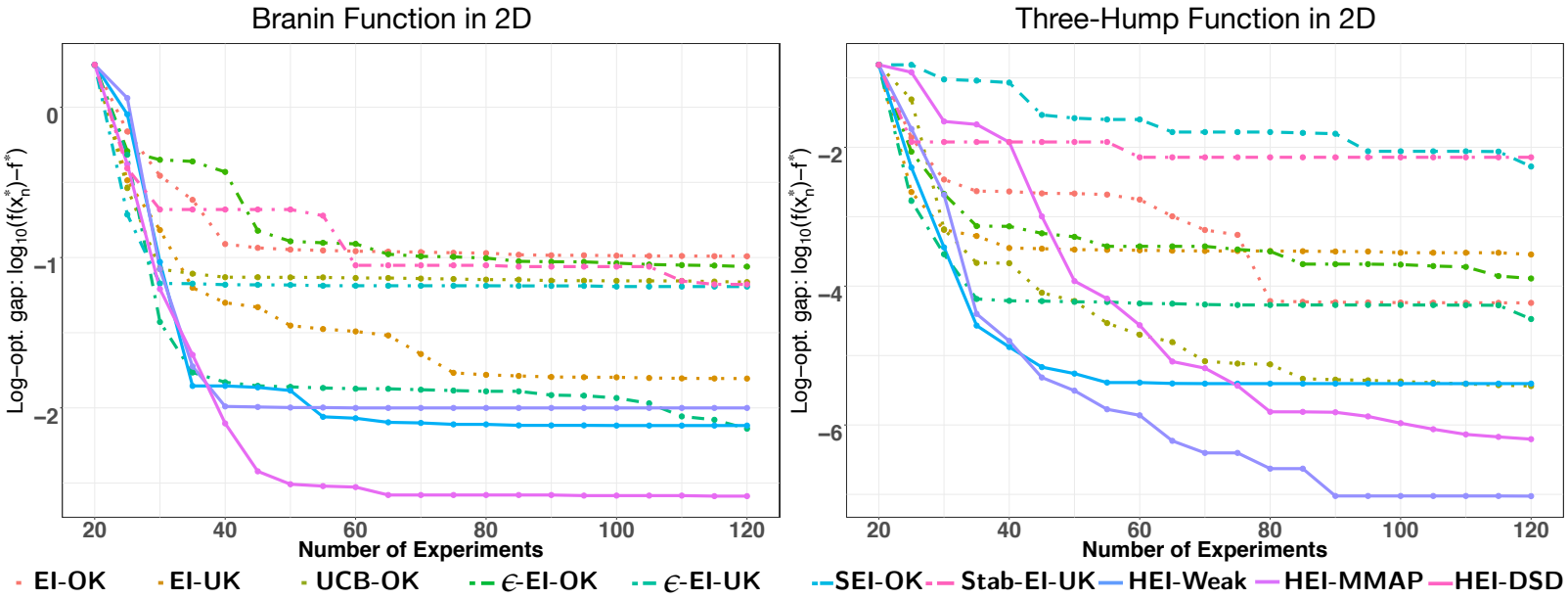
ϵ-**greedy EI** methods (Bull 2011):

- ϵ-EI-OK, ϵ-EI-UK, with $\epsilon = 0.1$ (Sutton & Barto 2018)

UCB-EI methods (Srinivas et al 2010):

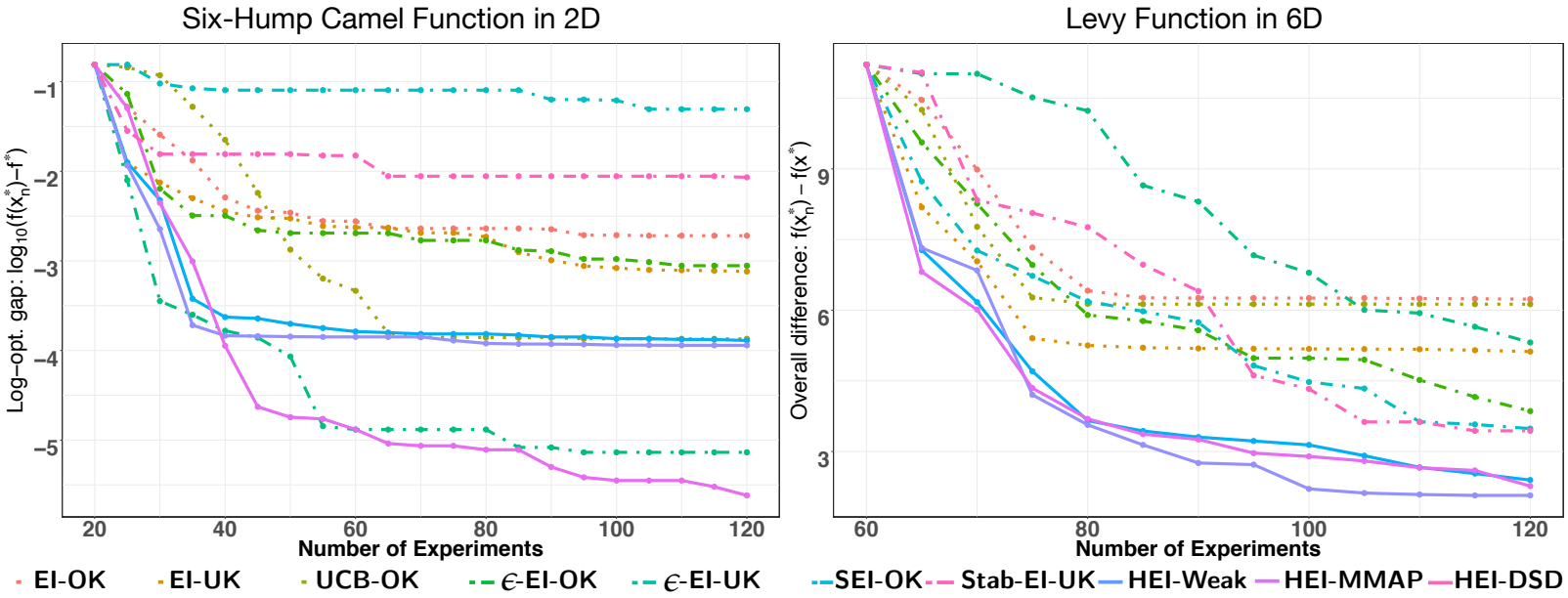
- Exploration parameter $\rho = 2.96$

Test functions



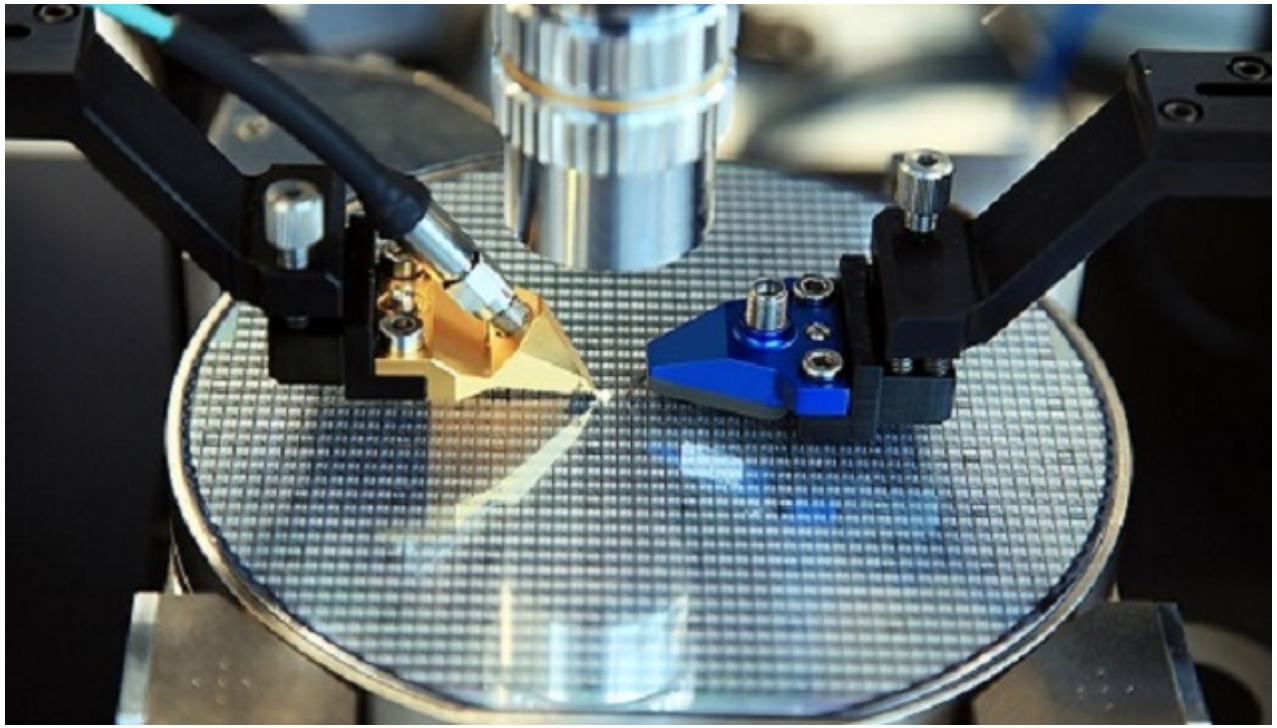
- HEI-MMAP & HEI-DSD perform **best**, EI **stuck** in **local optima**
- ϵ-greedy approaches **not bad**, but not as **effective** as HEI (**ad-hoc** fix to **over-exploitation**)
- HEI provides a **principled** correction via **hierarchical modeling**

Test functions



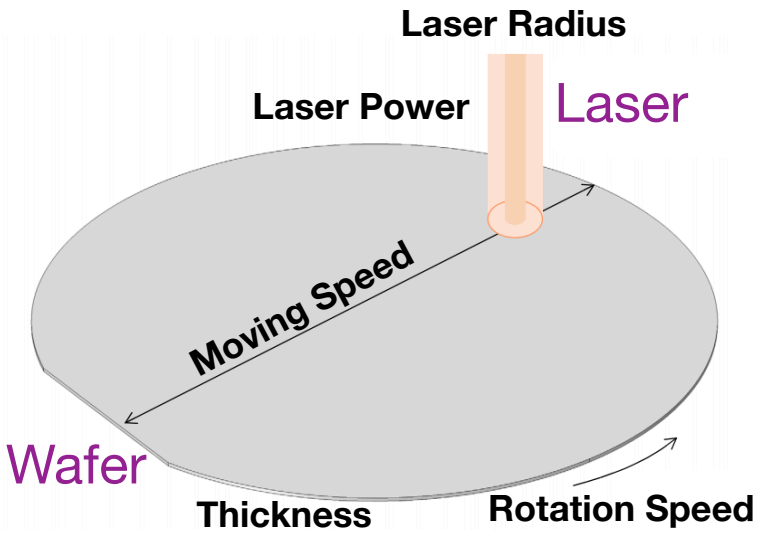
- HEI-MMAP & HEI-DSD perform **best**, EI **stuck** in **local optima**
- ϵ-greedy approaches **not bad**, but not as **effective** as HEI (**ad-hoc** fix to **over-exploitation**)
- HEI provides a **principled** correction via **hierarchical modeling**

Semiconductor manufacturing optimization



- **Semiconductor manufacturing**: process for manufacturing **integrated circuit chips** used in **electrical devices**
- **Thermal processing** is a crucial stage, allowing for necessary **chemical reactions** and **surface oxidation** (Singh et al 2000)

Semiconductor manufacturing optimization



- $d = 5$ **control variables** for **optimization**
- **Goal**: heat **silicon wafer** to a **target** temperature T $= 600$F **quickly** and **smoothly**

Semiconductor manufacturing optimization

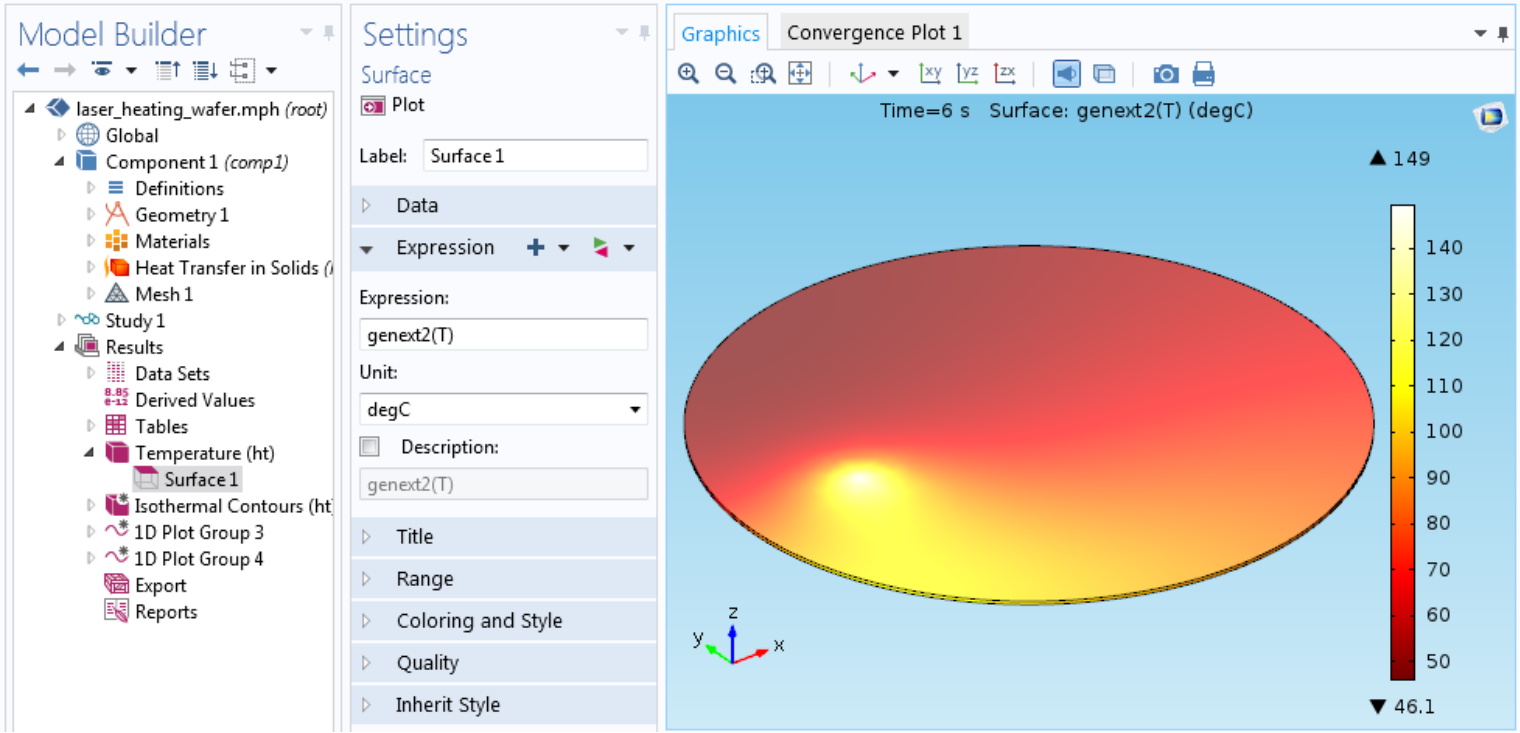
Objective function $f(\boldsymbol{x})$ to minimize:

$$f(\boldsymbol{x}) := \sum_{t=1}^{60} \max_{\mathbf{s}\in\mathcal{S}} \underbrace{\left|\mathcal{T}_t(\mathbf{s};\boldsymbol{x}) - 600\right|}_{\textbf{Gap to Target}}$$

- $\mathbf{s}$: **location** on wafer $\mathcal{S}$
- $\mathcal{T}_t(\mathbf{s};\boldsymbol{x})$: **temperature** at time t and location $\mathbf{s}$

Settings $\boldsymbol{x}$ with **small** $f(\boldsymbol{x})$ ensure that thermal processing is performed **quickly** and **smoothly**

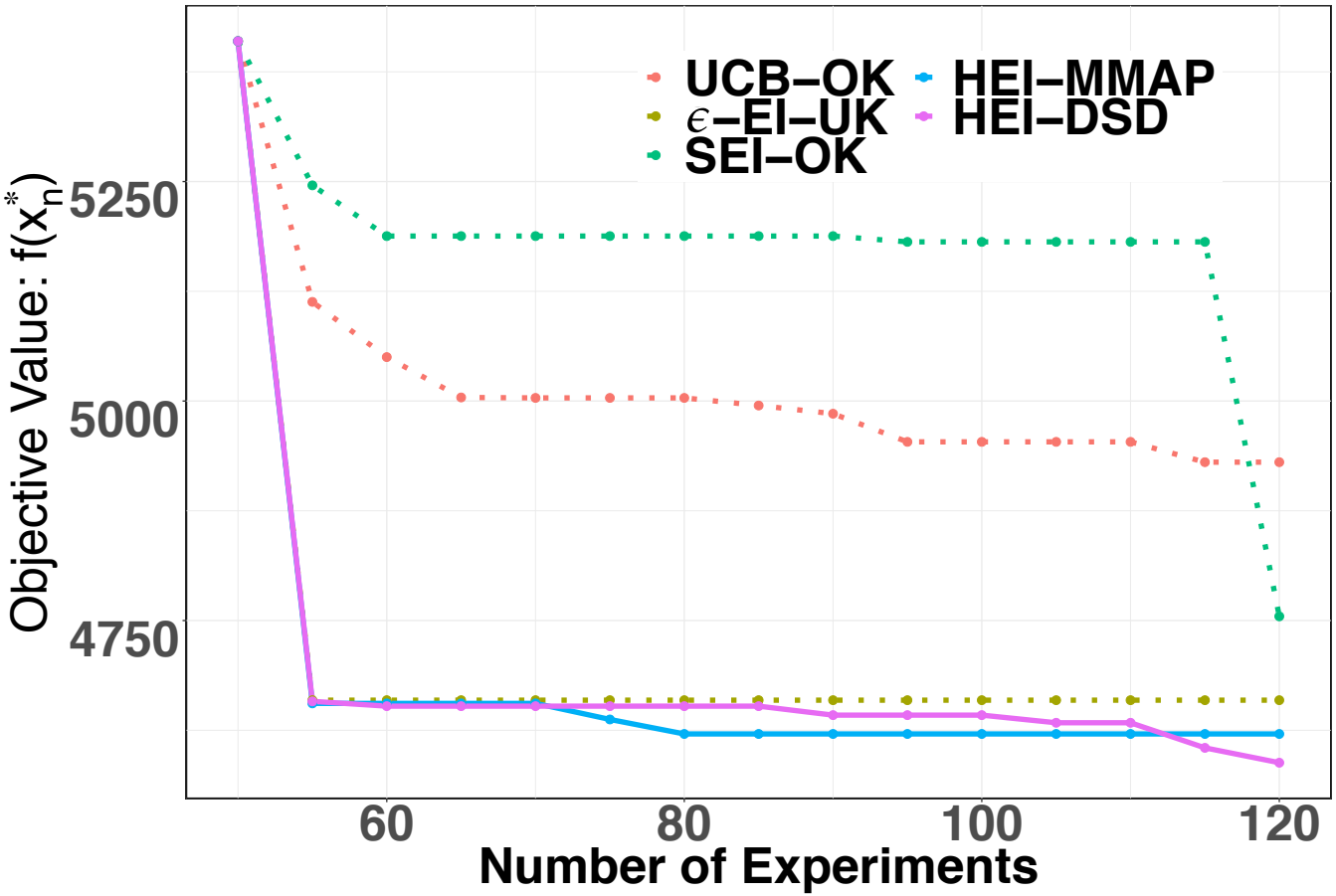
Semiconductor manufacturing optimization



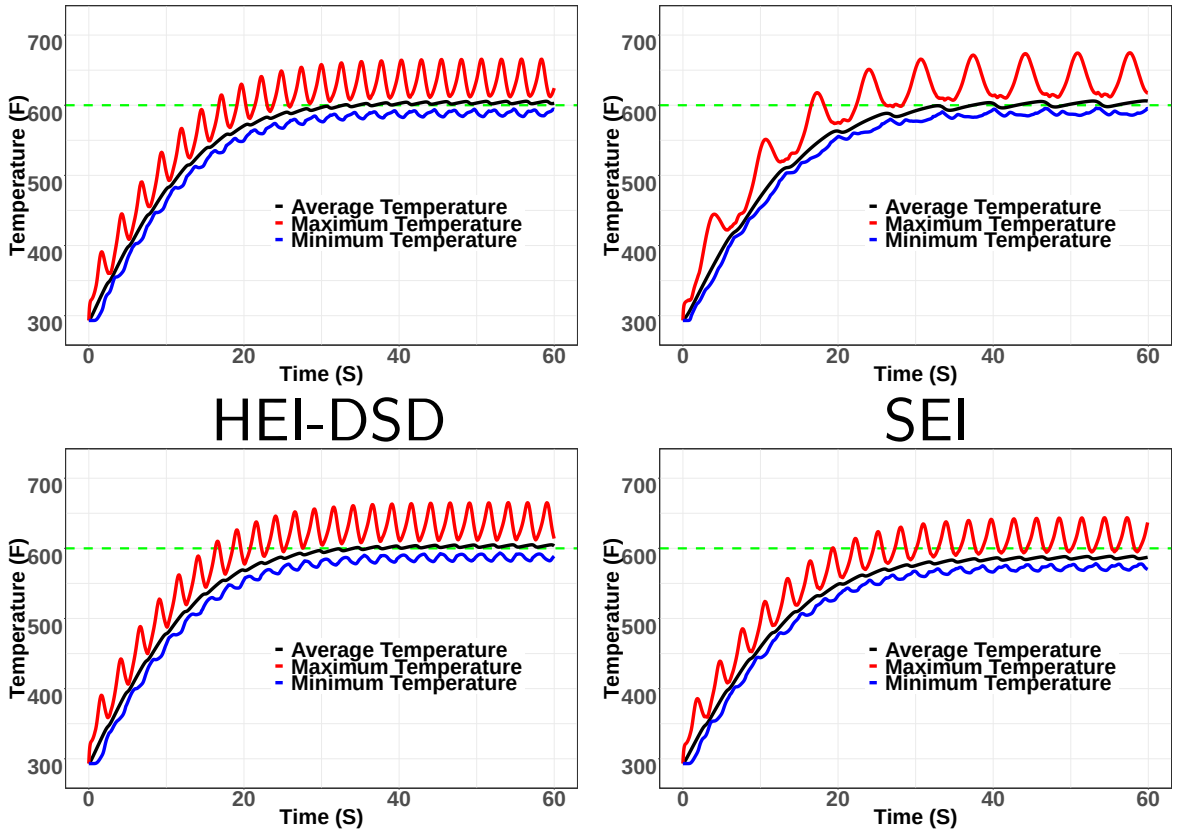
f is **expensive** and **black-box**:

- **Experiments** are **costly** and require **prototyping**
- **Simulations** can be performed on **COMSOL**, but require **30 minutes** for **each** setting

Semiconductor manufacturing optimization

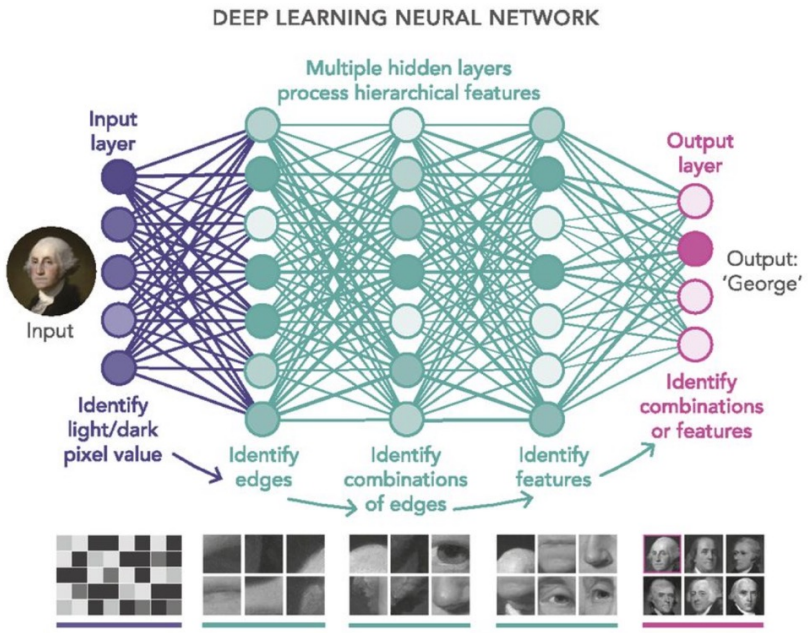


Semiconductor manufacturing optimization



- Visually, HEI-DSD and ϵ-EI-UK give the **best** settings: **average** temperature reaches **target** T **quickly** and **smoothly** (variation over wafer is **small**)

Hyperparameter tuning in deep learning

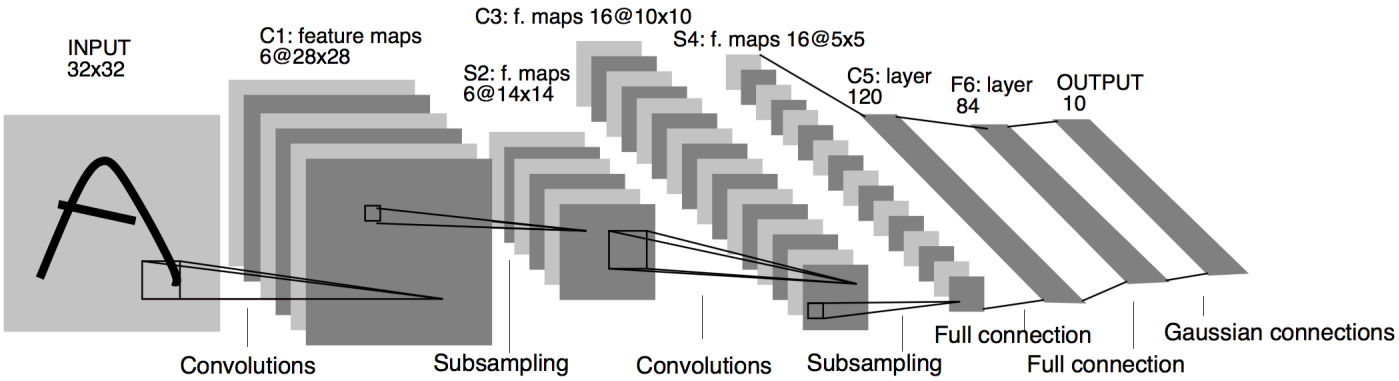


Neural networks (NNs) are widely used in **ML** problems:

- ... but in order to work **well**, many **hyperparameters** need to be carefully **tuned**
- NN training is **expensive** even for a **single** hyperparameter setting (several **hours** for real image databases, such as CIFAR)

Hyperparameter tuning in deep learning

- **Architecture**: LeNet-5 (LeCun et al. 1998)



- **Formulation**: **NN training** solves the following problem

$$\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell(\mathcal{F}(\boldsymbol{x};\boldsymbol{w}), y)$$

  - ℓ: **loss** function (**softmax** used here)
  - $\mathcal{F}(\cdot;\boldsymbol{w})$: **neural network** with weight matrix $\boldsymbol{w}$
  - $(\boldsymbol{x}, y) \sim \mathcal{D}$: **image** and corresponding **label**

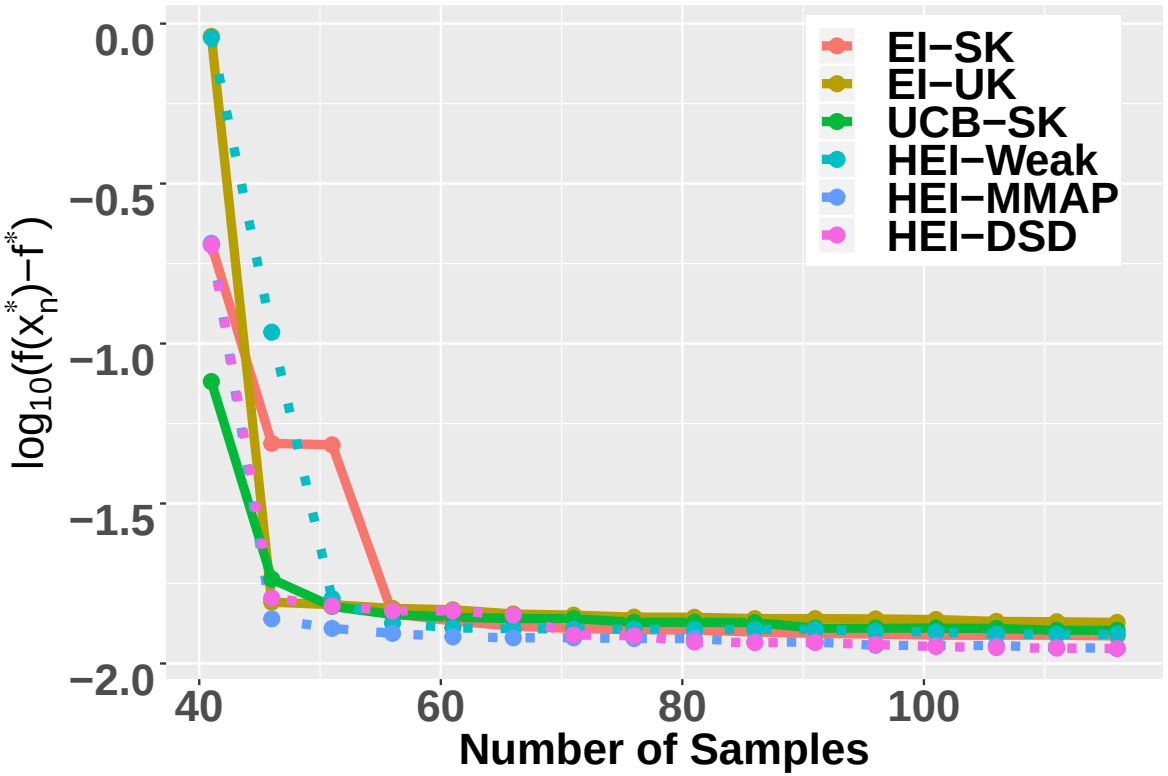
Hyperparameter tuning in deep learning

- **Training Method**:
  Momentum stochastic gradient descent with weight decay:

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \underbrace{\eta \cdot \nabla_{\boldsymbol{w}} \ell(\mathcal{F}(\boldsymbol{x}_{i_t}, \boldsymbol{w}^{(t)}), y_{i_t})}_{\text{Gradient Descent Term}} + \underbrace{\mu \cdot (\boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t-1)})}_{\text{Momentum Term}} - \underbrace{\lambda \cdot \boldsymbol{w}^{(t)}}_{\text{Weight Decay Term}}$$

- **Hyperparameters**:
  - η: learning rate
  - μ: momentum
  - τ: weight matrix initialization variance
  - λ: weight decay

Hyperparameter tuning in deep learning



- y-**axis**: **log-validation error** of trained NN
- HEI-DSD & HEI-MMAP perform **best**: **lowest** errors
- Existing tuning method (**gradient-based tuning**, Bengio 2000) gives a **higher** **log-error** of -1.854

Conclusions

- **Exploration** is important for **learning** and **optimizing** expensive **black-box** functions
- A **closed-form acquisition function** allows for **efficient optimization**
- A **hierarchical** Bayesian model which considers parameter **uncertainties** can encourage more **exploration** and **correct** the **greediness** (**overexploitation**) of EI

Questions?