Parallel predictive entropy search for multi-objective Bayesian optimization with constraints

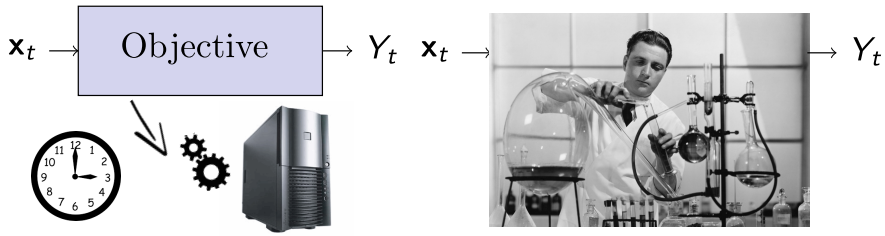
Daniel Hernández–Lobato
Computer Science Department
Universidad Autónoma de Madrid

http://dhnzl.org, daniel.hernandez@uam.es

Joint work with
Eduardo C. Garrido-Merchán and Daniel Fernández Sánchez

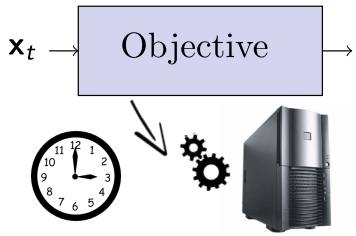
Bayesian Optimization: Common Features

- Very expensive evaluations.



Bayesian Optimization: Common Features

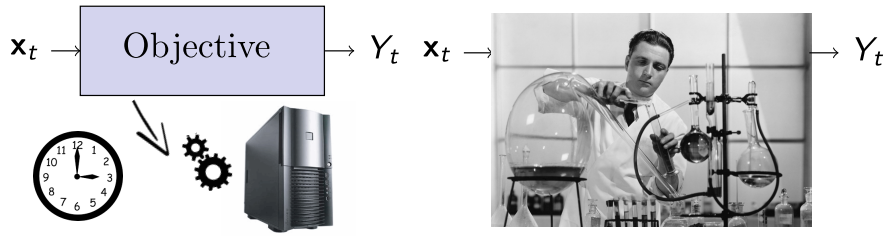
- Very expensive evaluations.



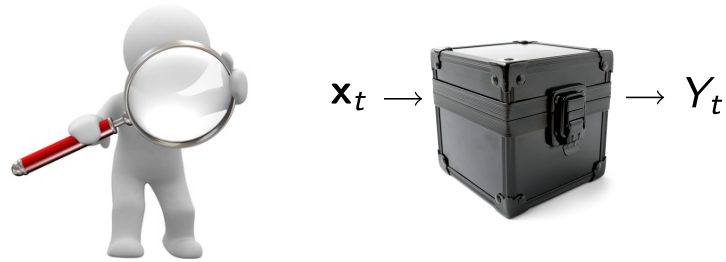
- The objective is a black-box.

Bayesian Optimization: Common Features

- Very expensive evaluations.

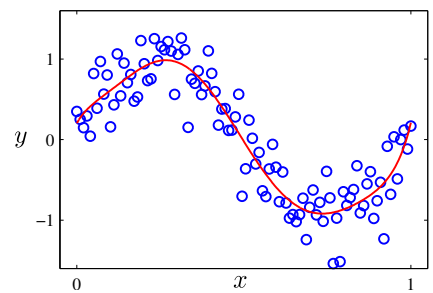


- The objective is a black-box.



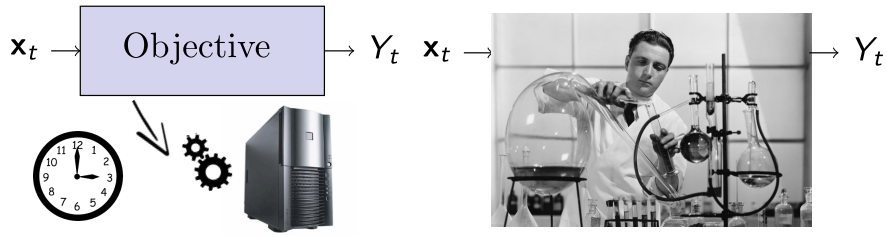
- The evaluation can be noisy.

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

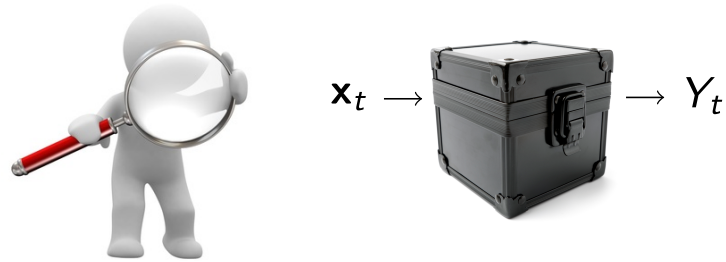


Bayesian Optimization: Common Features

- Very expensive evaluations.

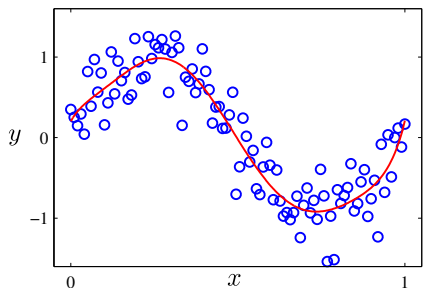


- The objective is a black-box.



- The evaluation can be noisy.

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

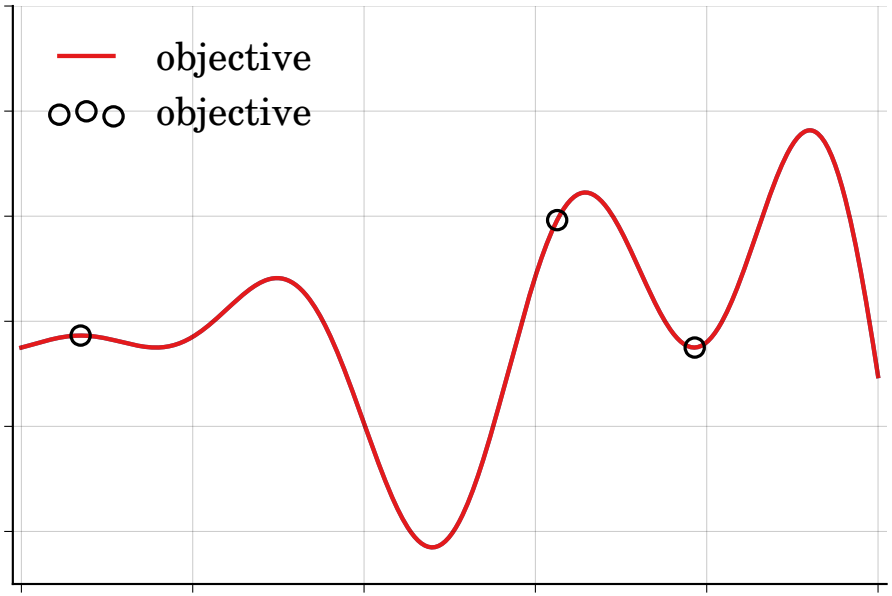


Bayesian optimization methods can be used to solve these problems!

Bayesian Optimization in Practice

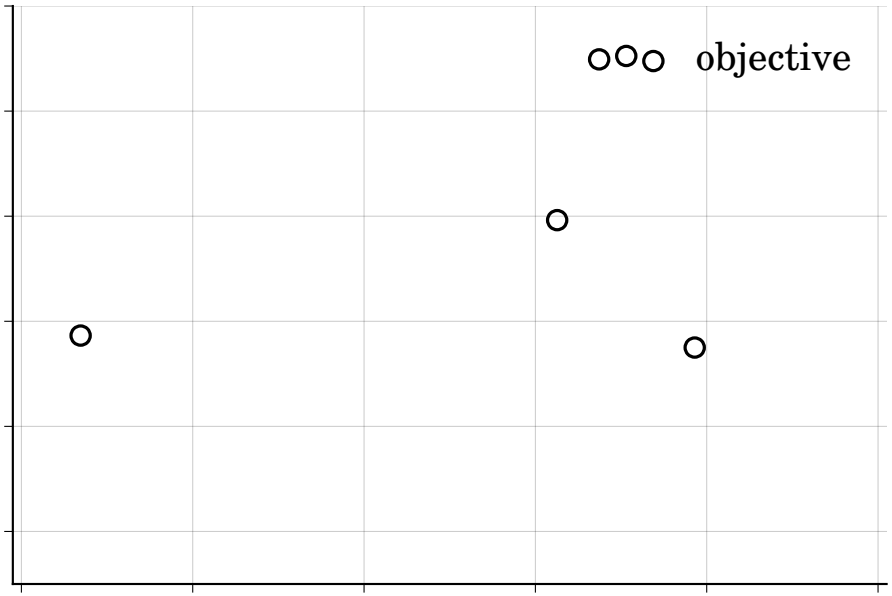
1. Get initial sample.

Bayesian Optimization in Practice



1. **Get initial sample.**

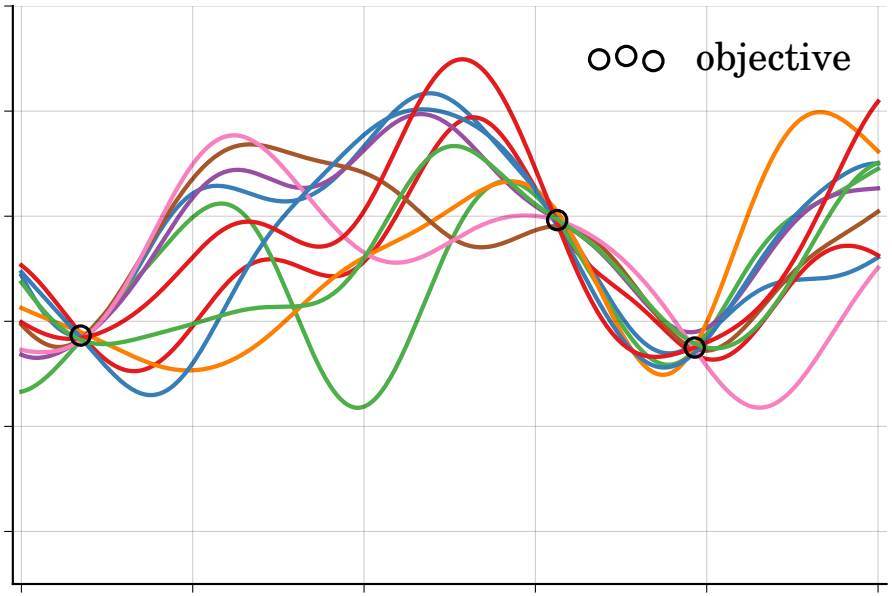
Bayesian Optimization in Practice



1. **Get initial sample.**
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

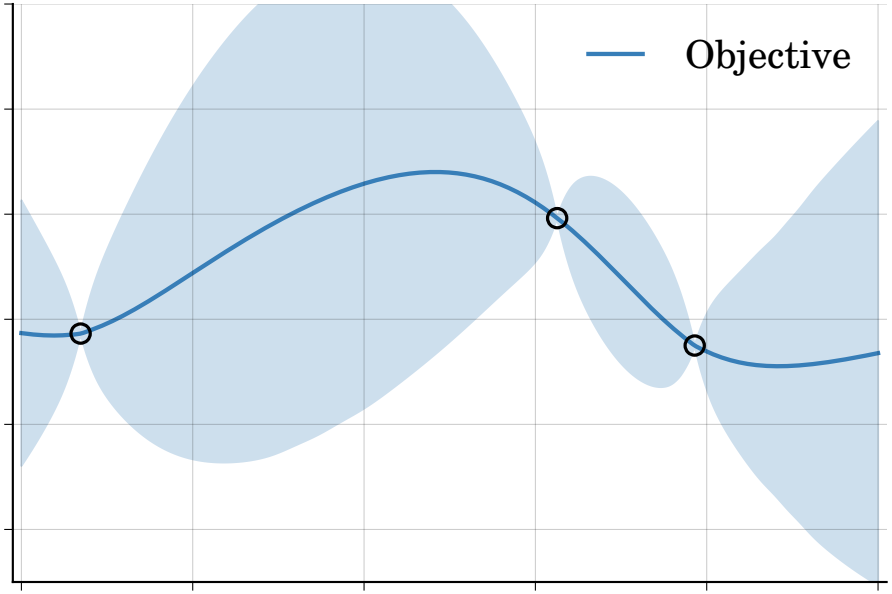
Bayesian Optimization in Practice



1. Get initial sample.
2. **Fit a model to the data:**

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

Bayesian Optimization in Practice



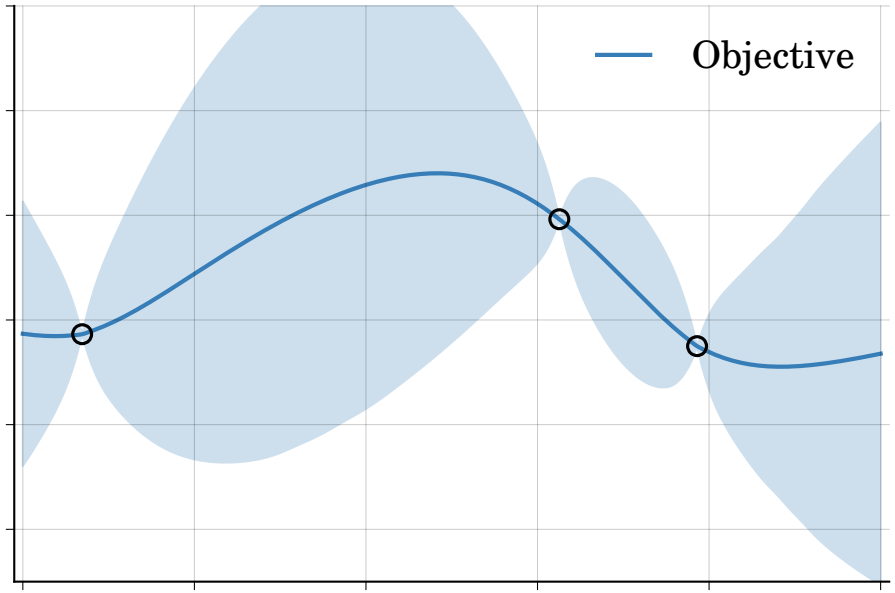
1. Get initial sample.
2. **Fit a model to the data:**

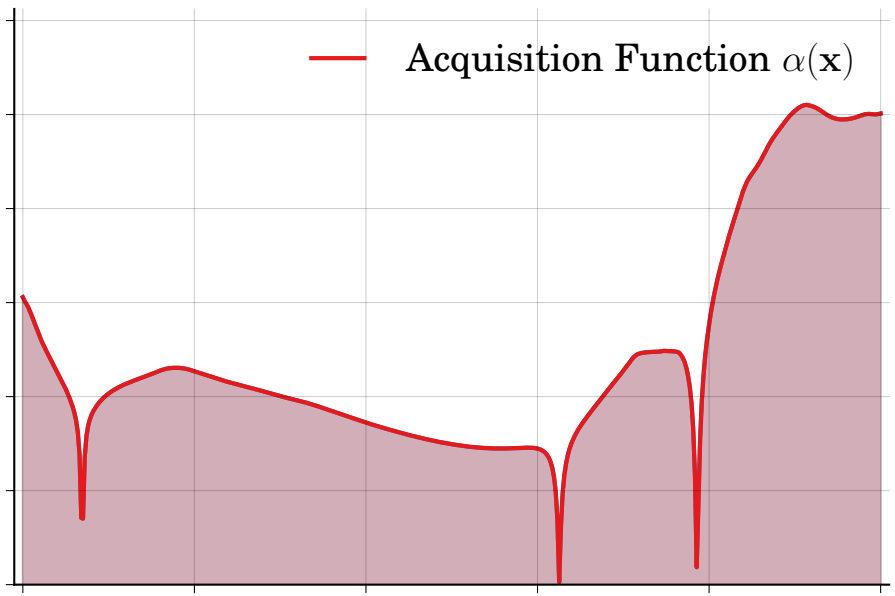
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. Select data collection strategy:

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

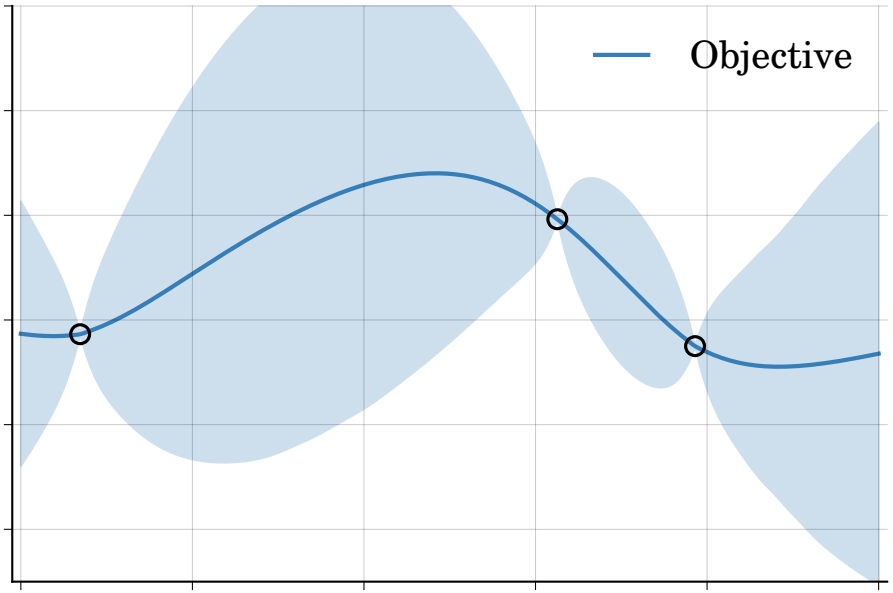
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

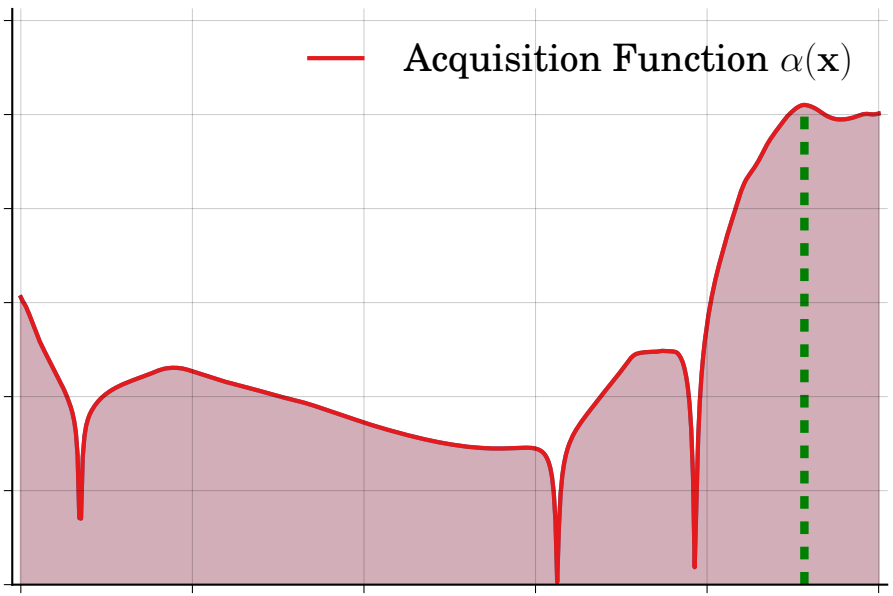
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. Optimize acquisition function $\alpha(\mathbf{x})$.

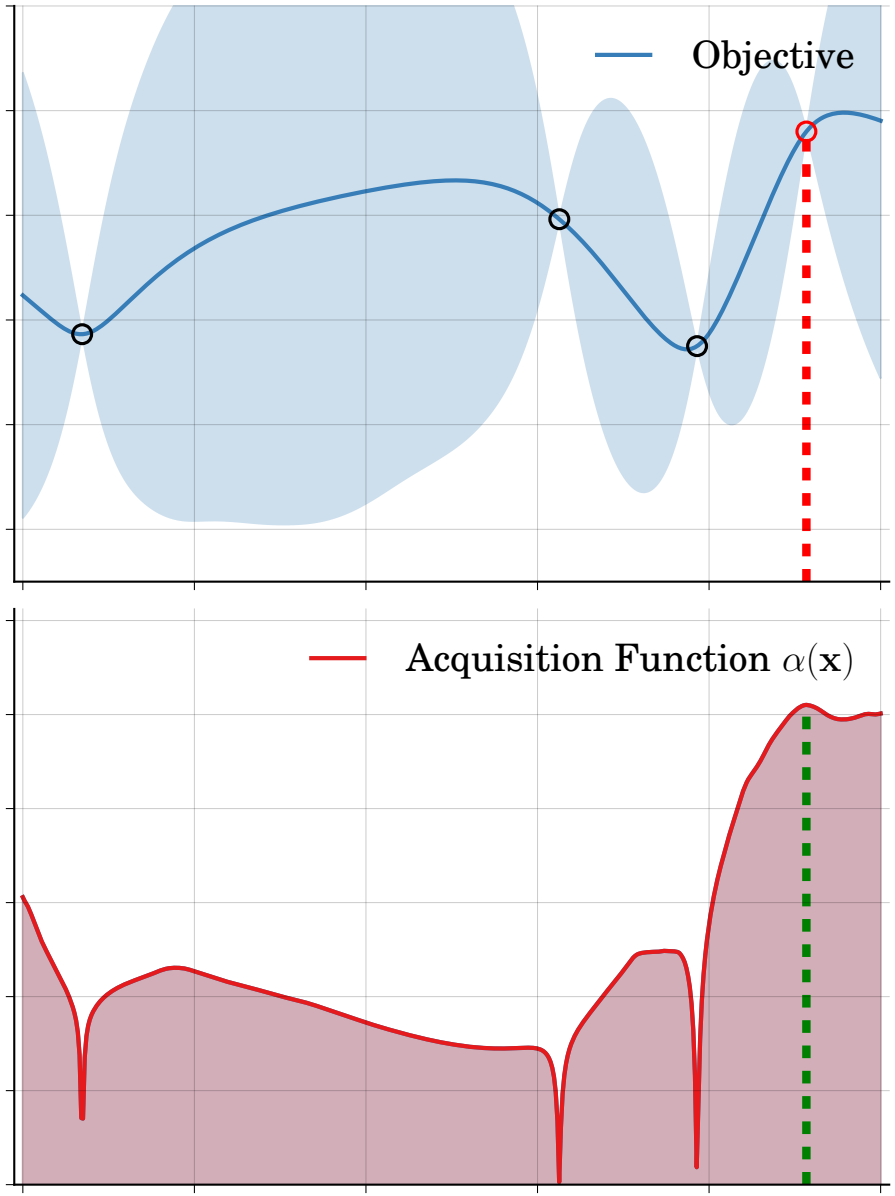
Bayesian Optimization in Practice





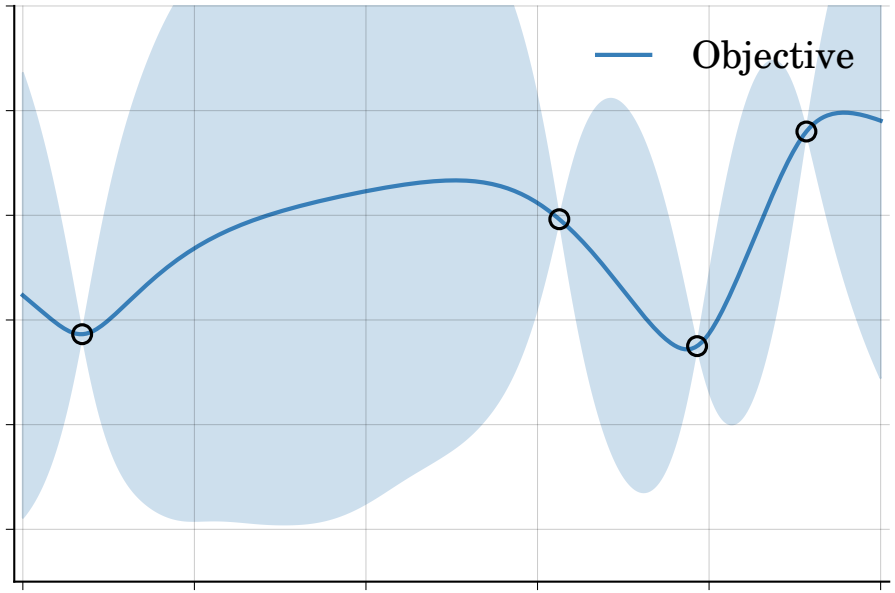
1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$
3. Select data collection strategy:
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$
4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. Collect data and update model.

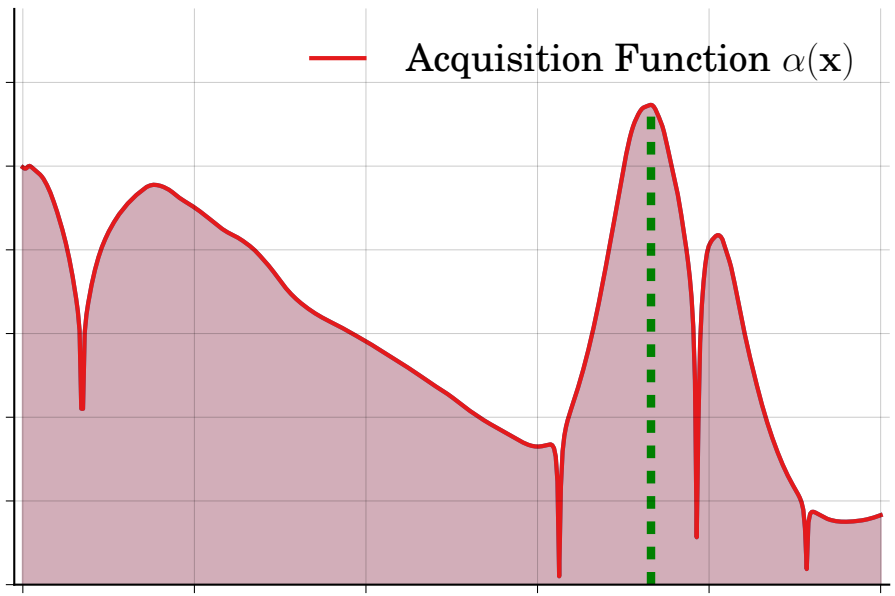
Bayesian Optimization in Practice



1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$
3. Select data collection strategy:
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$
4. Optimize acquisition function $\alpha(\mathbf{x})$.
5. **Collect data and update model.**
6. Repeat!

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

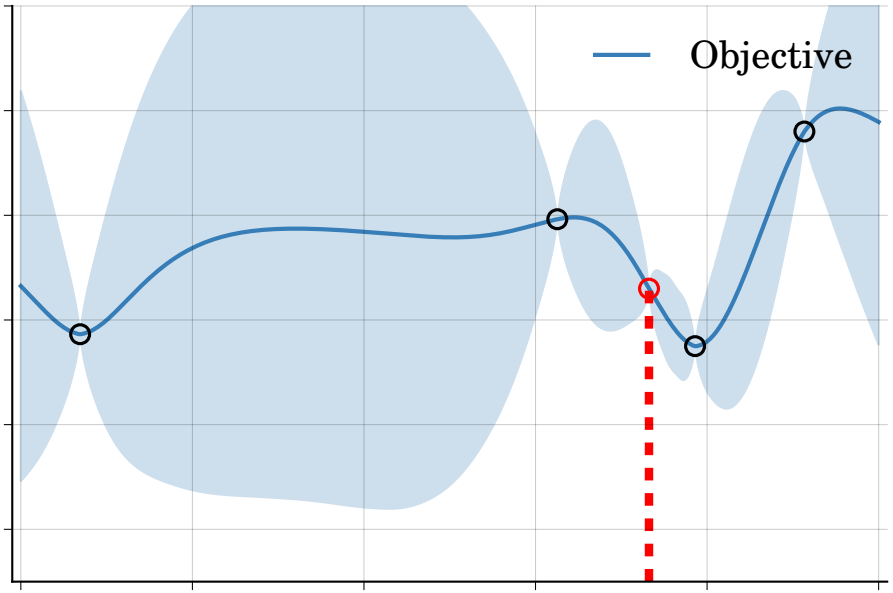
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

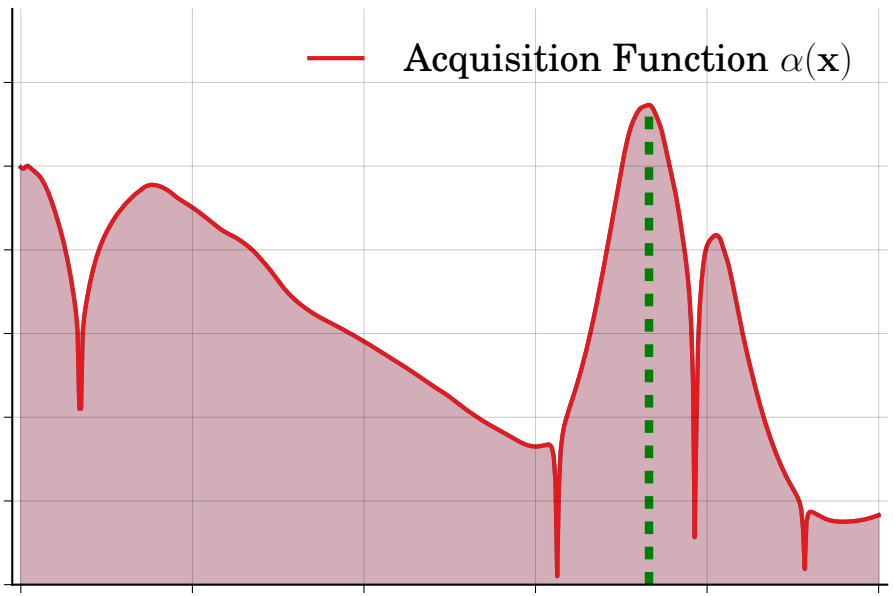
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

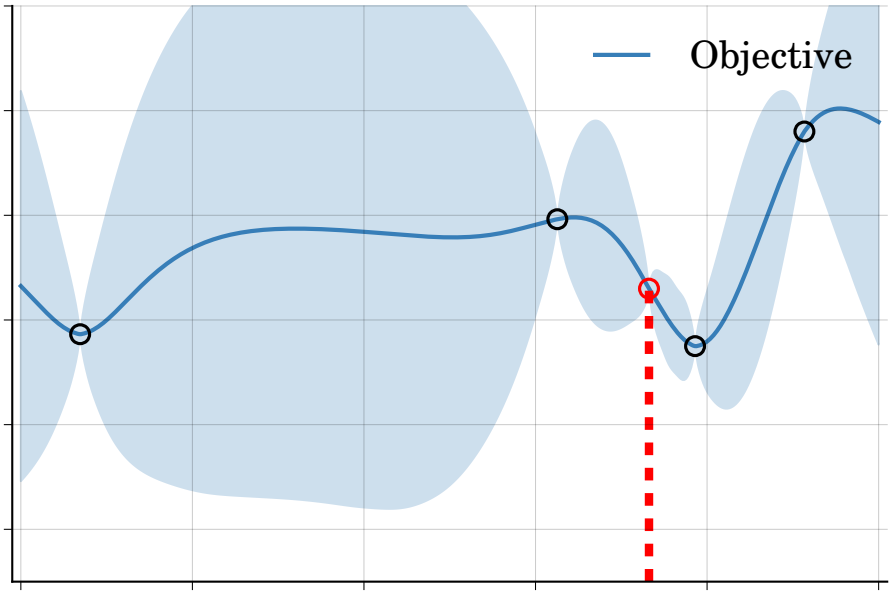
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

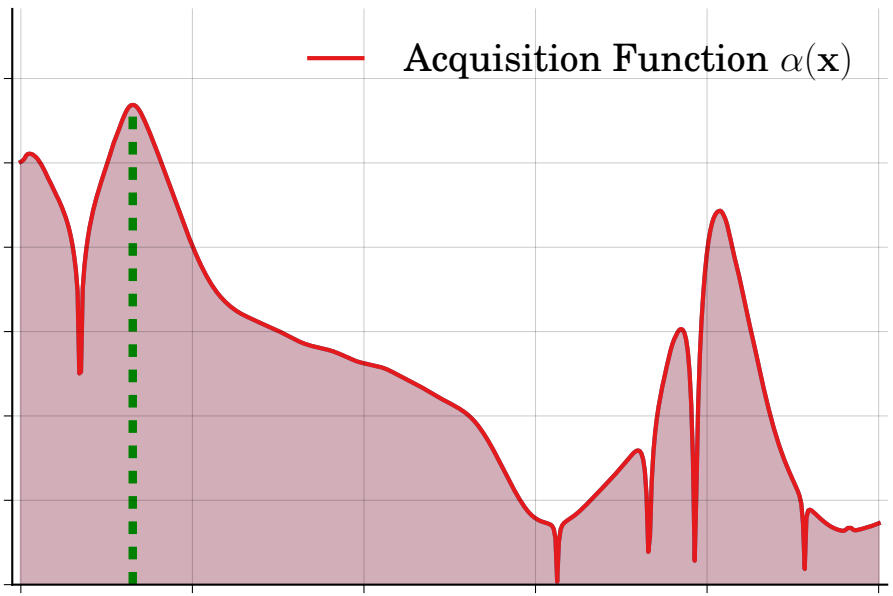
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

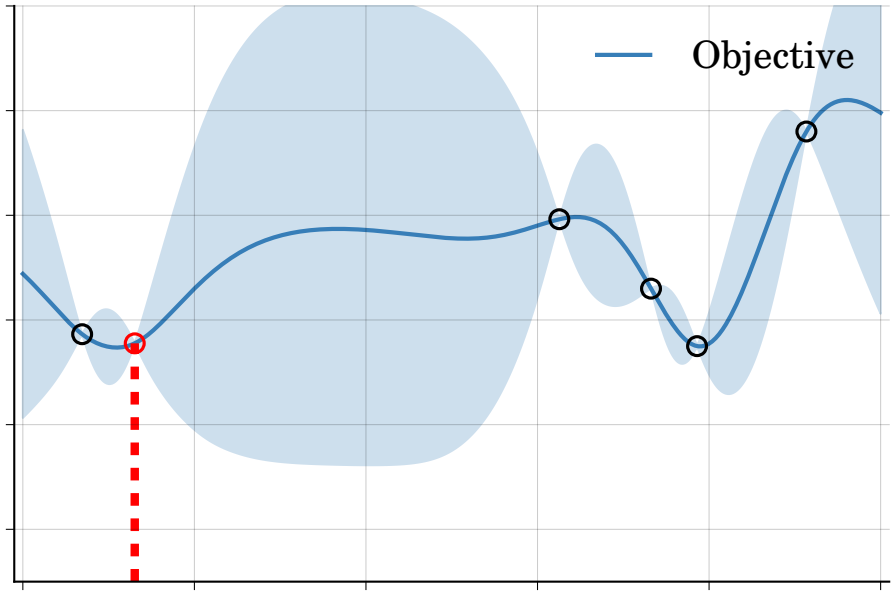
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

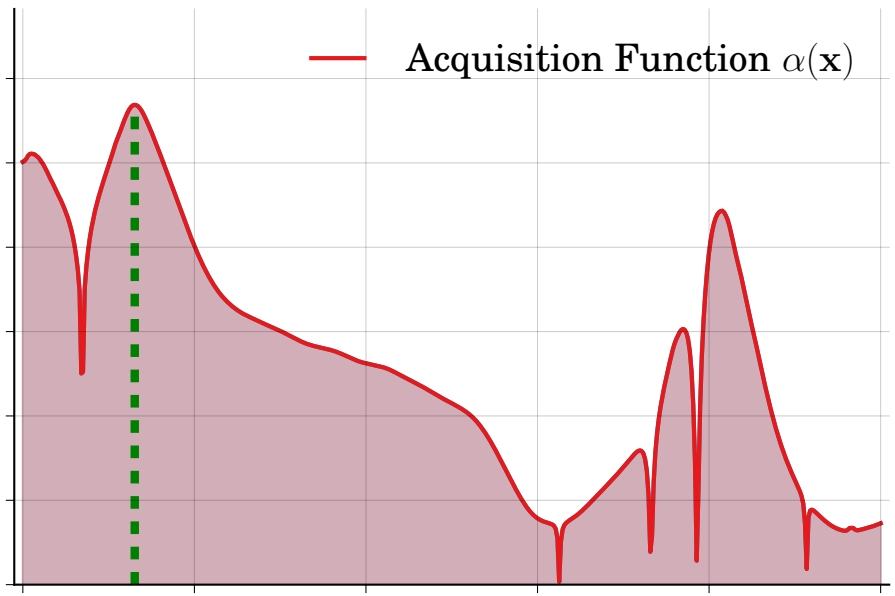
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

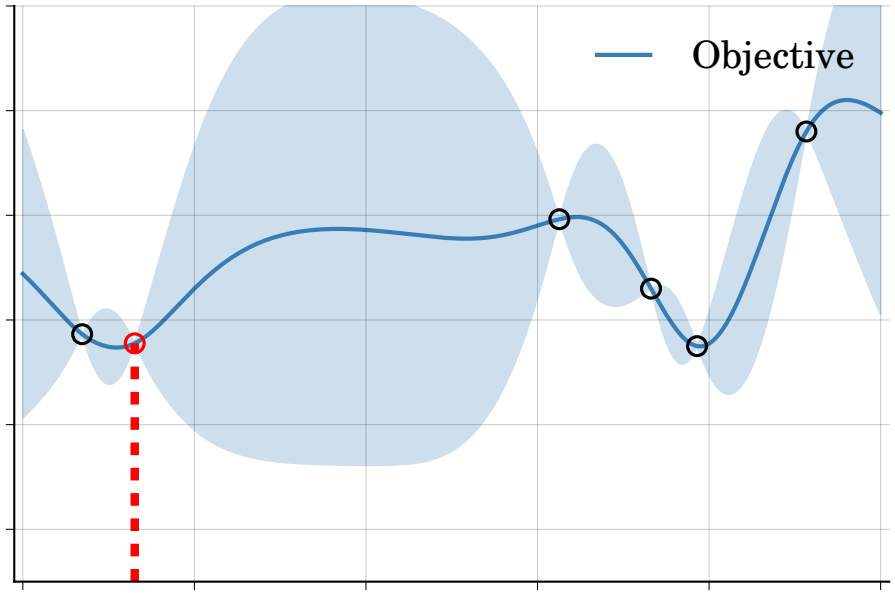
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

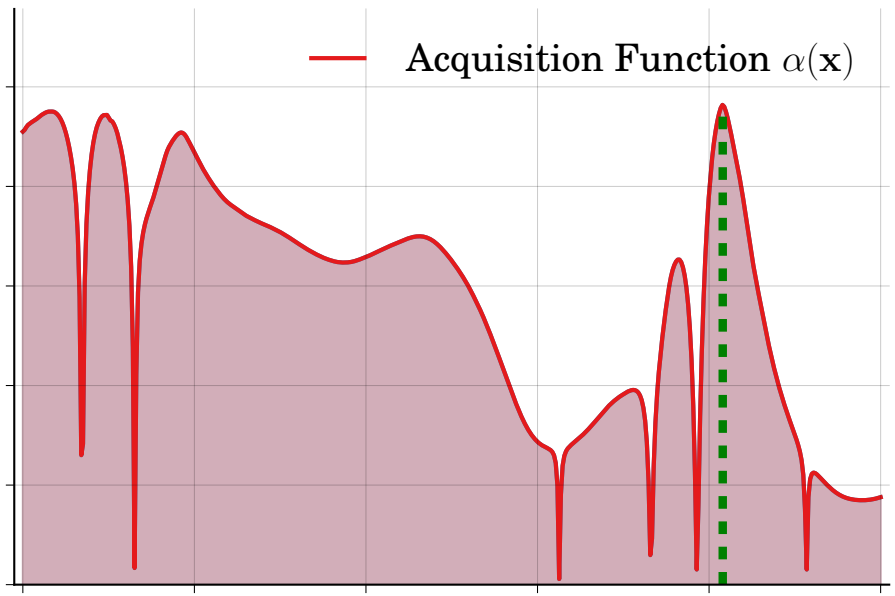
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

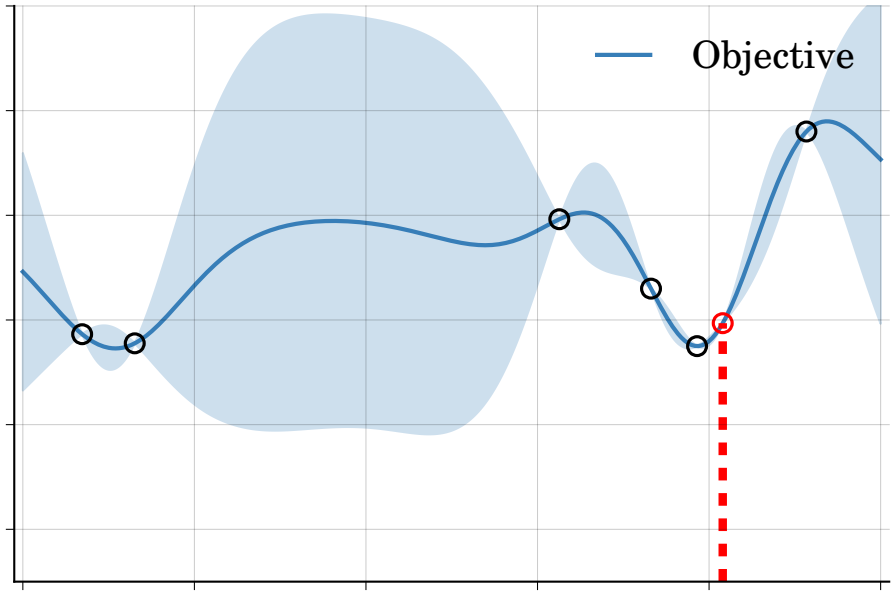
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

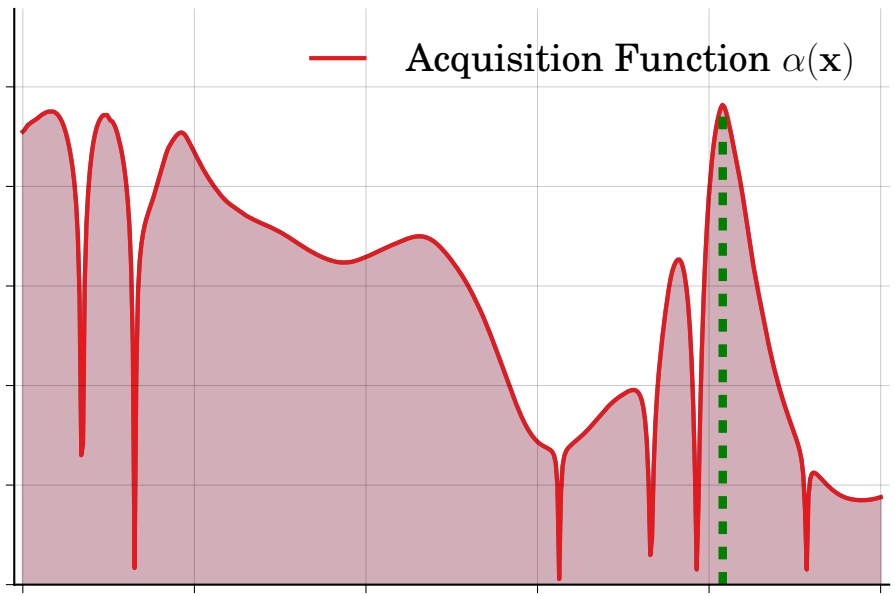
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

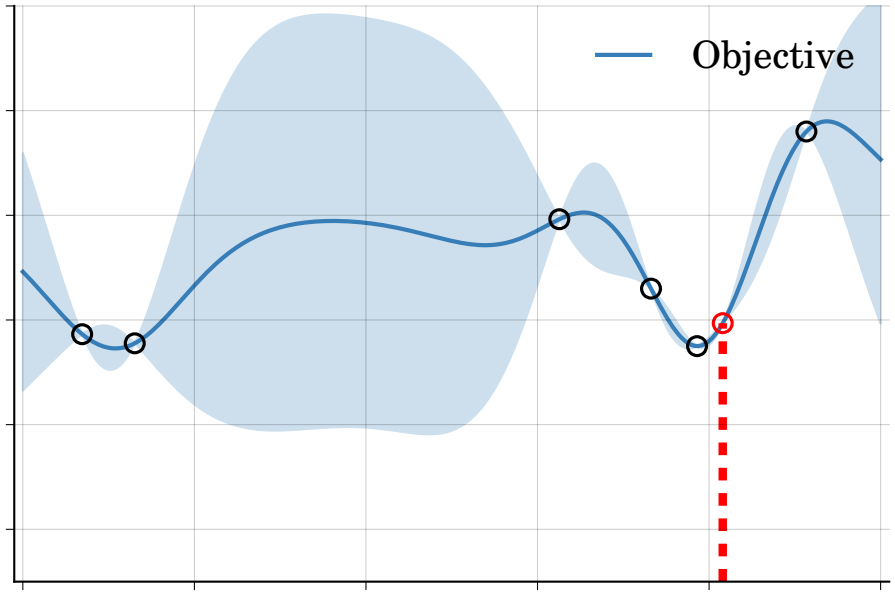
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

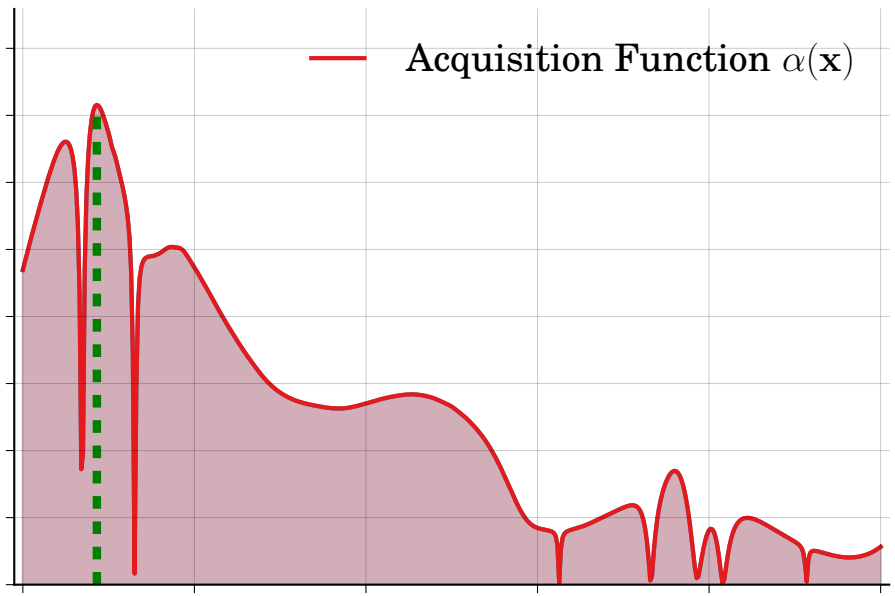
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

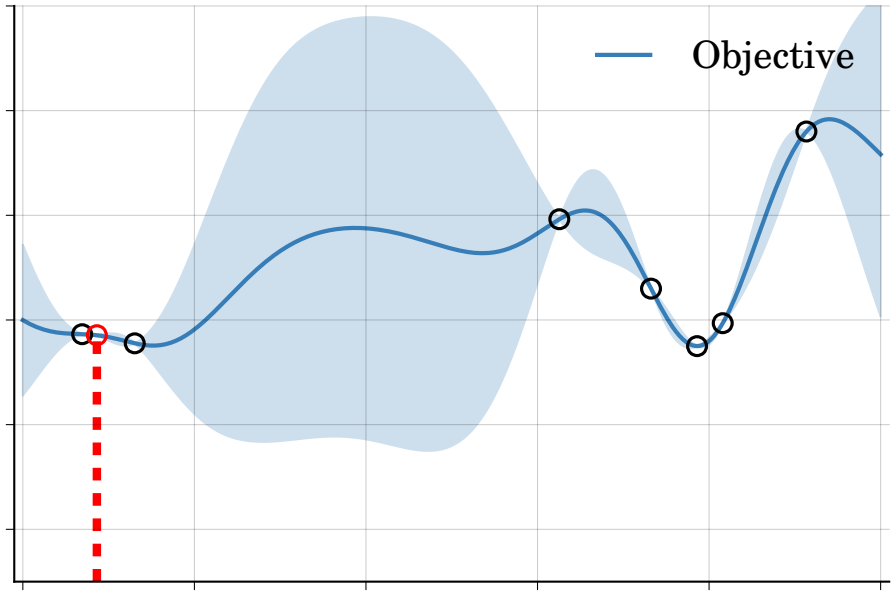
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

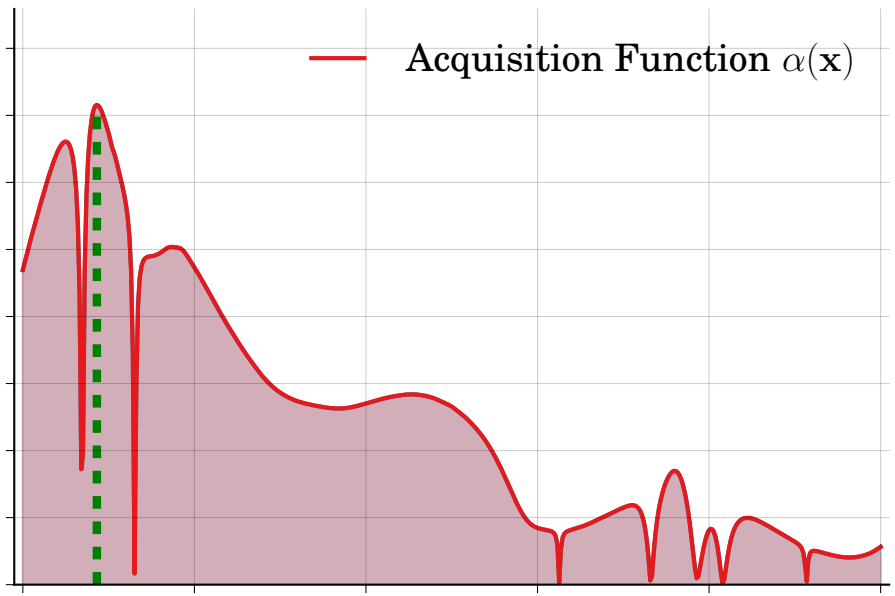
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

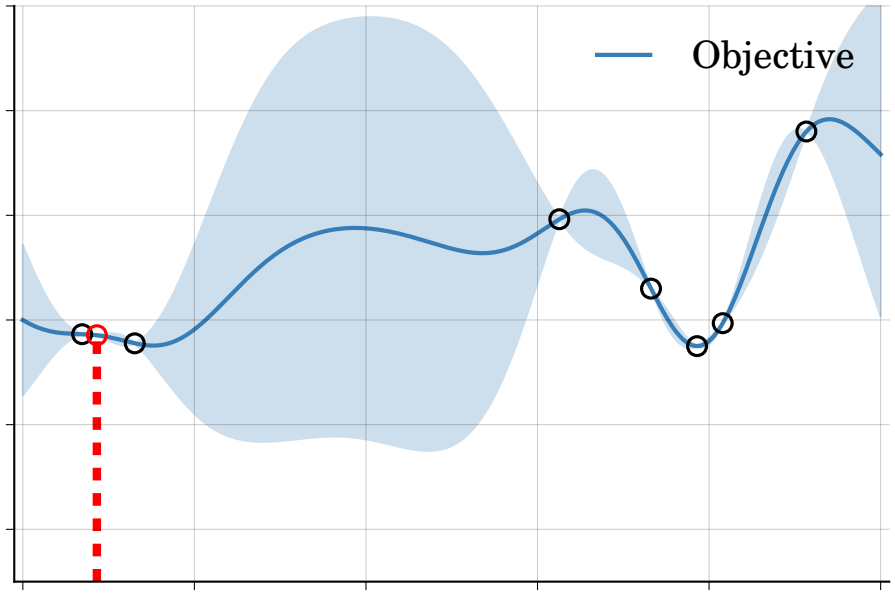
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

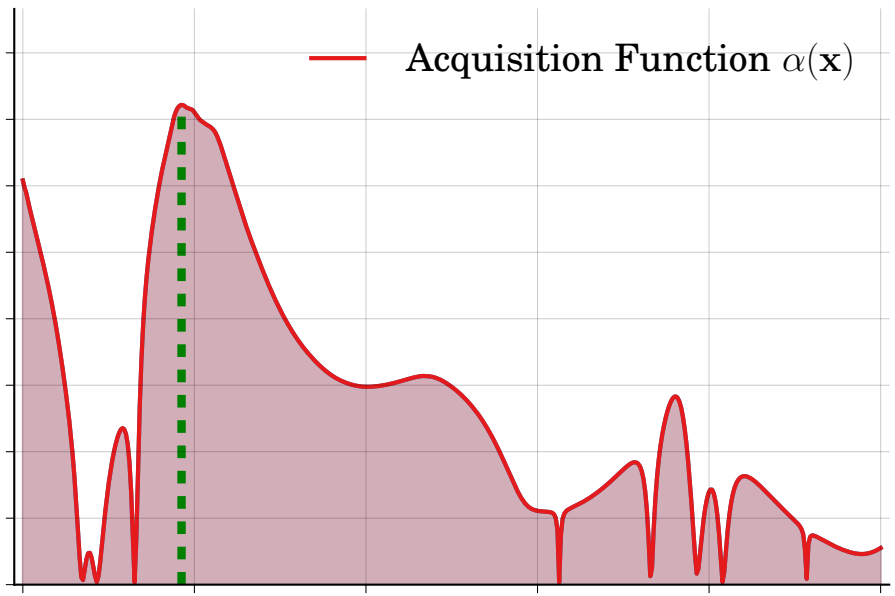
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

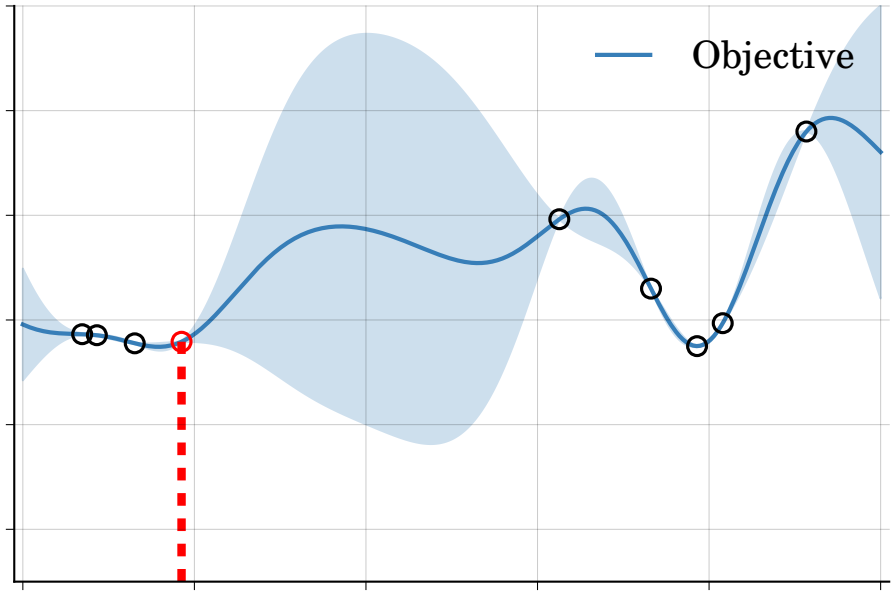
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

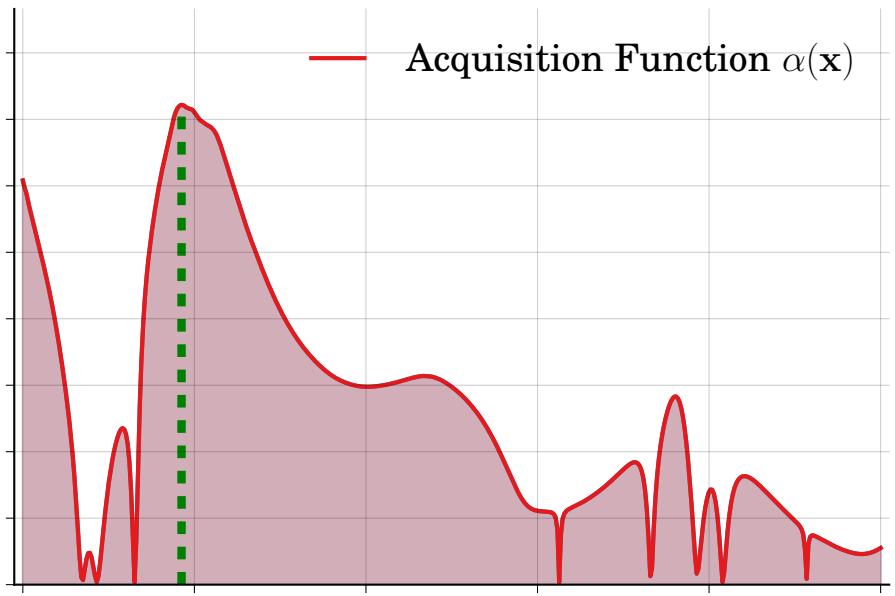
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

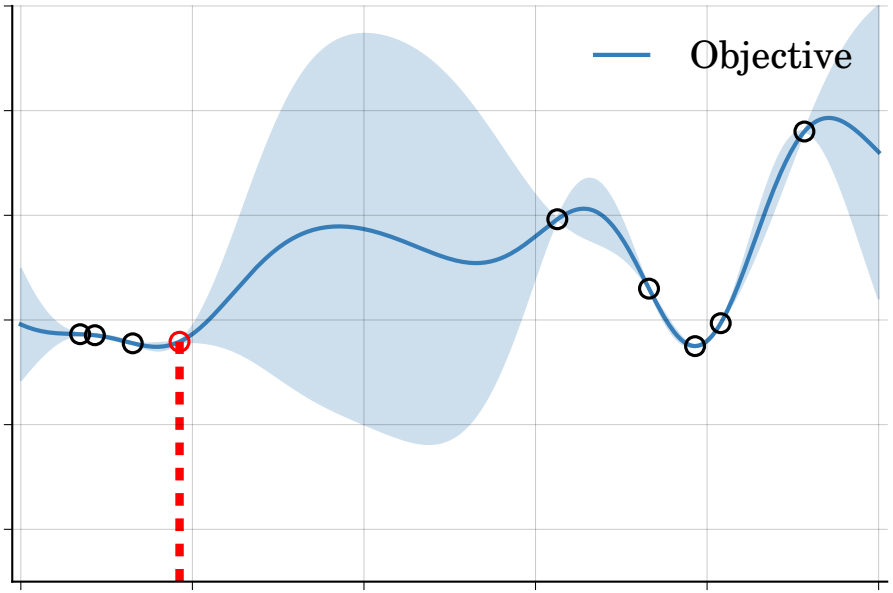
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

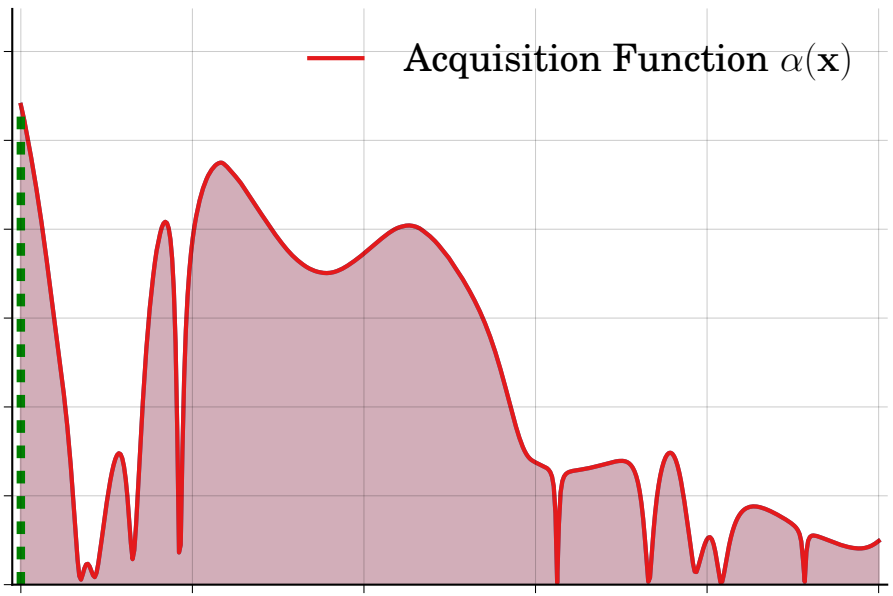
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

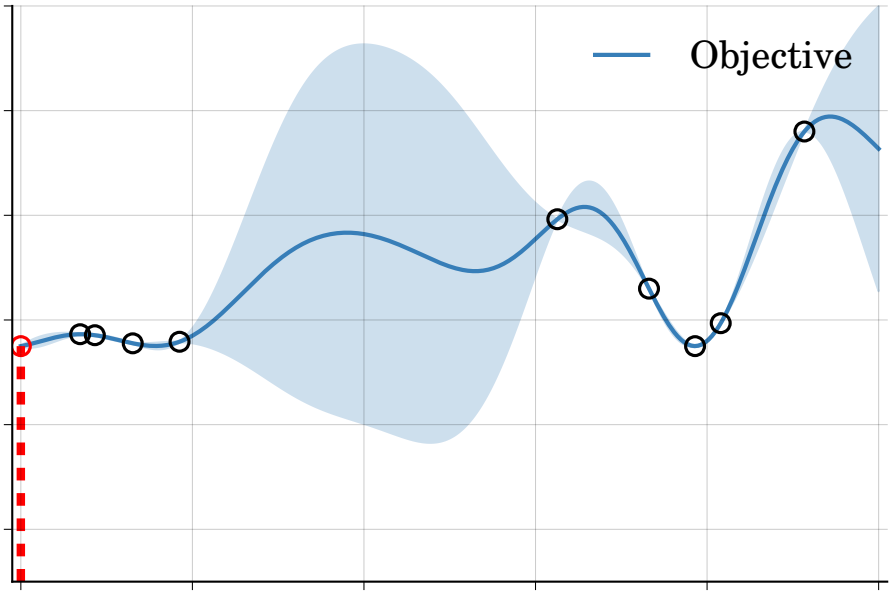
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

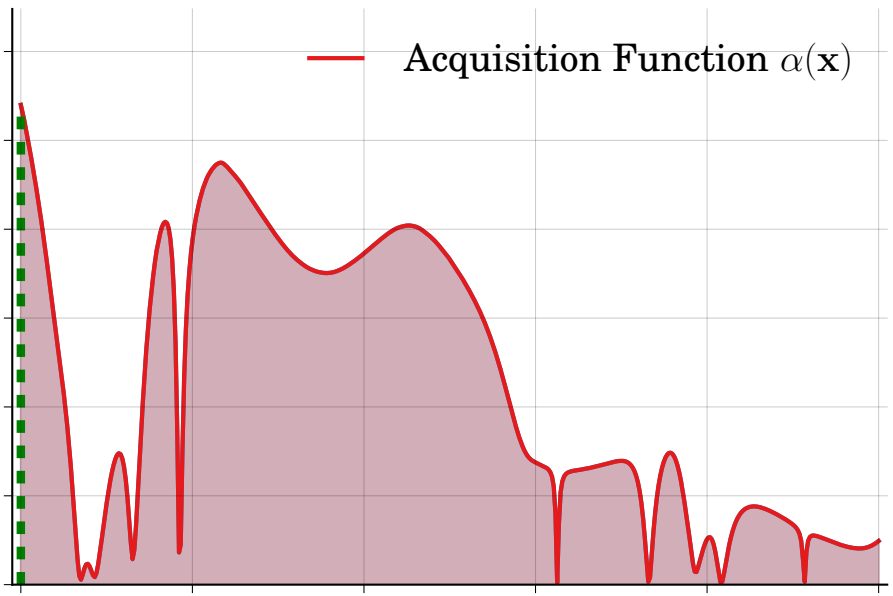
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

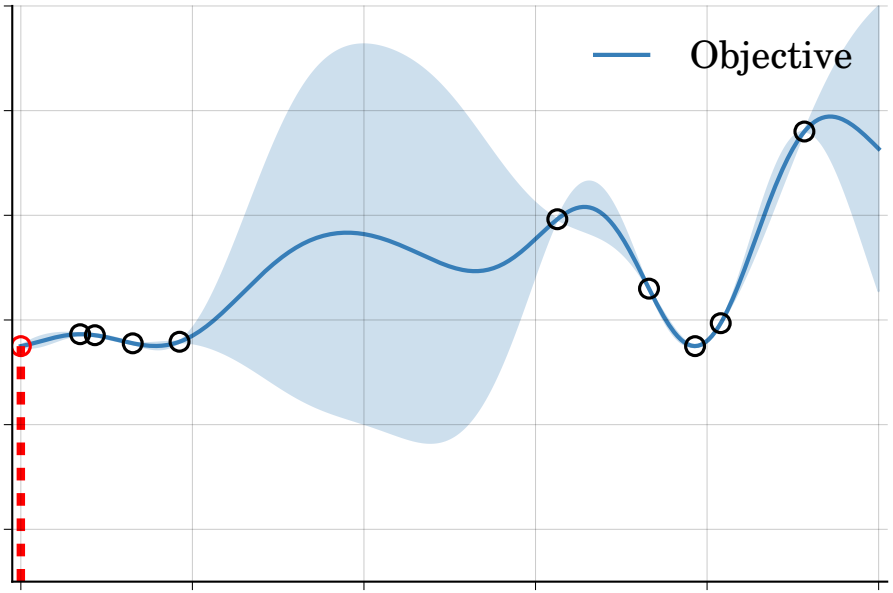
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

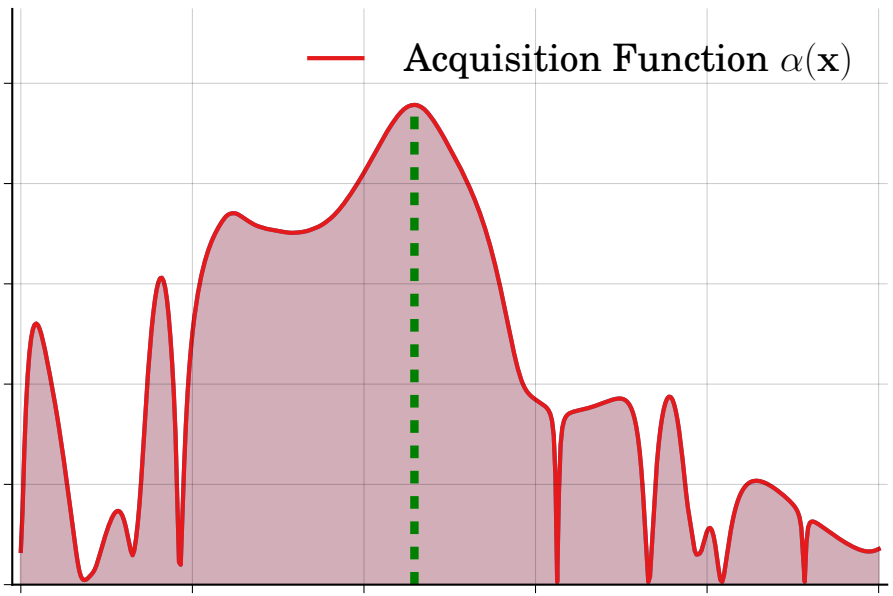
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

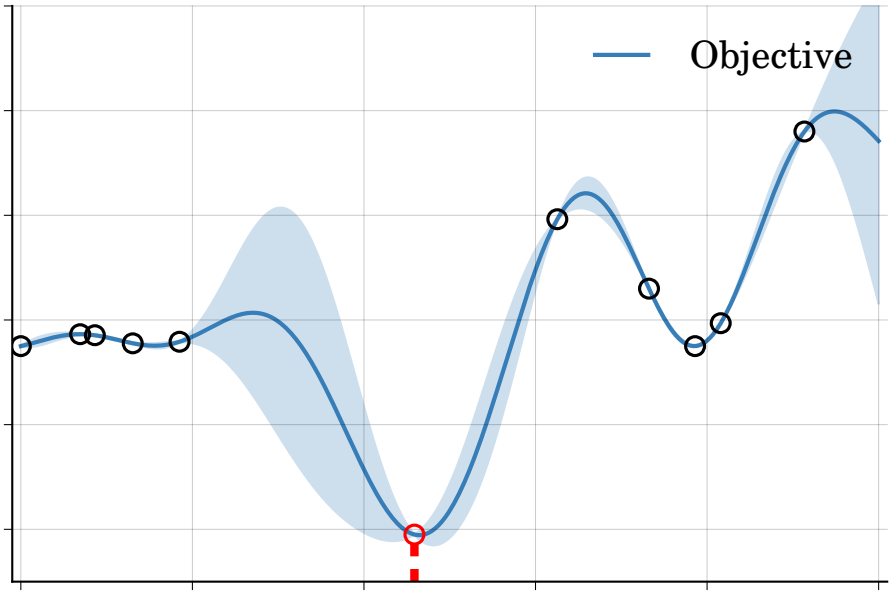
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

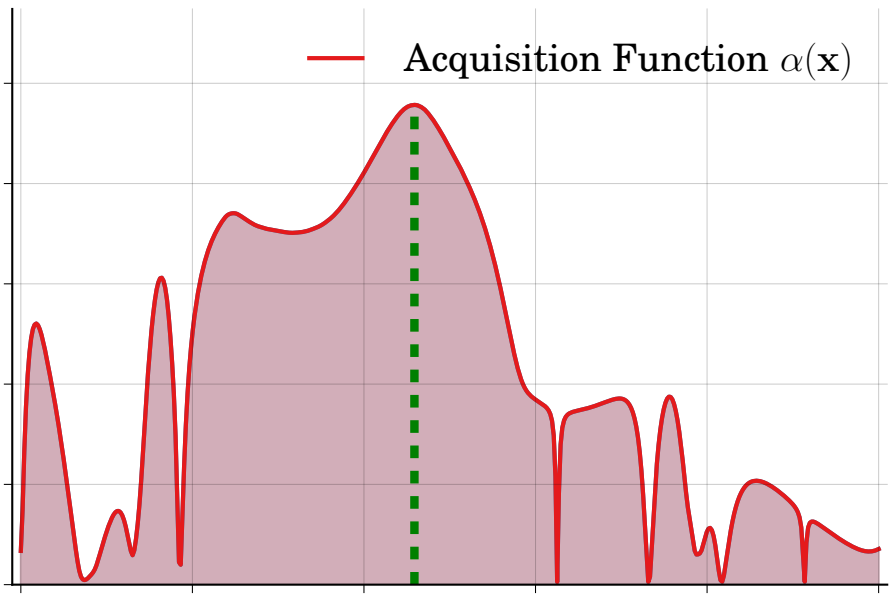
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

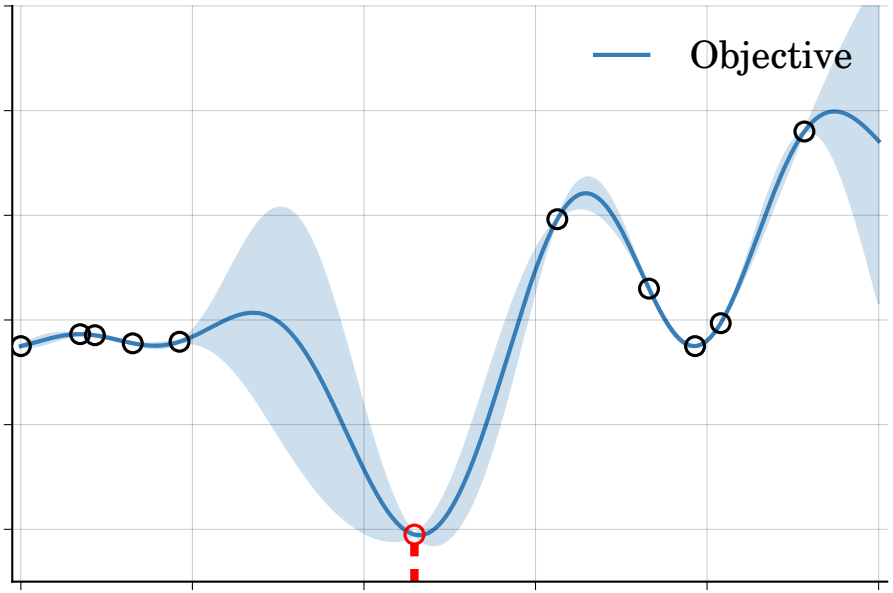
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

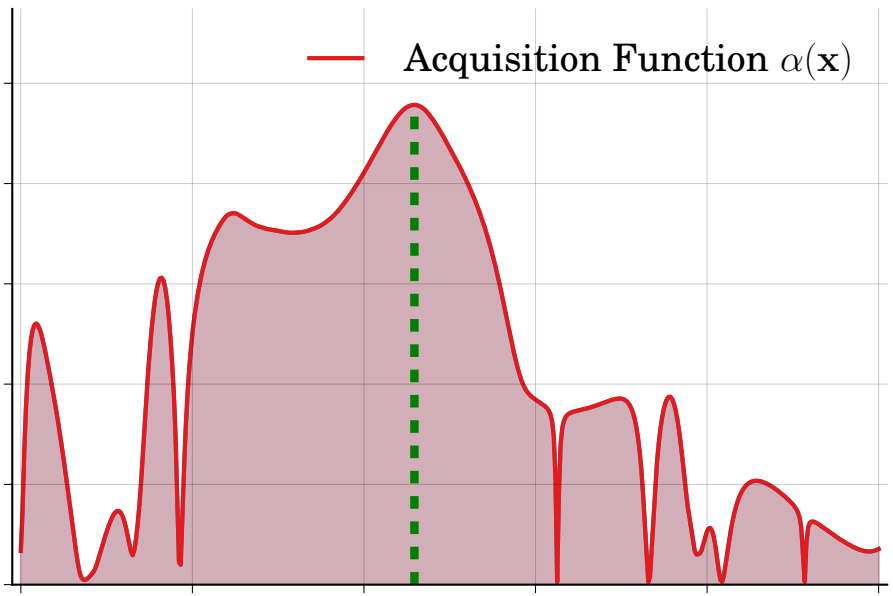
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Bayesian Optimization in Practice





1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

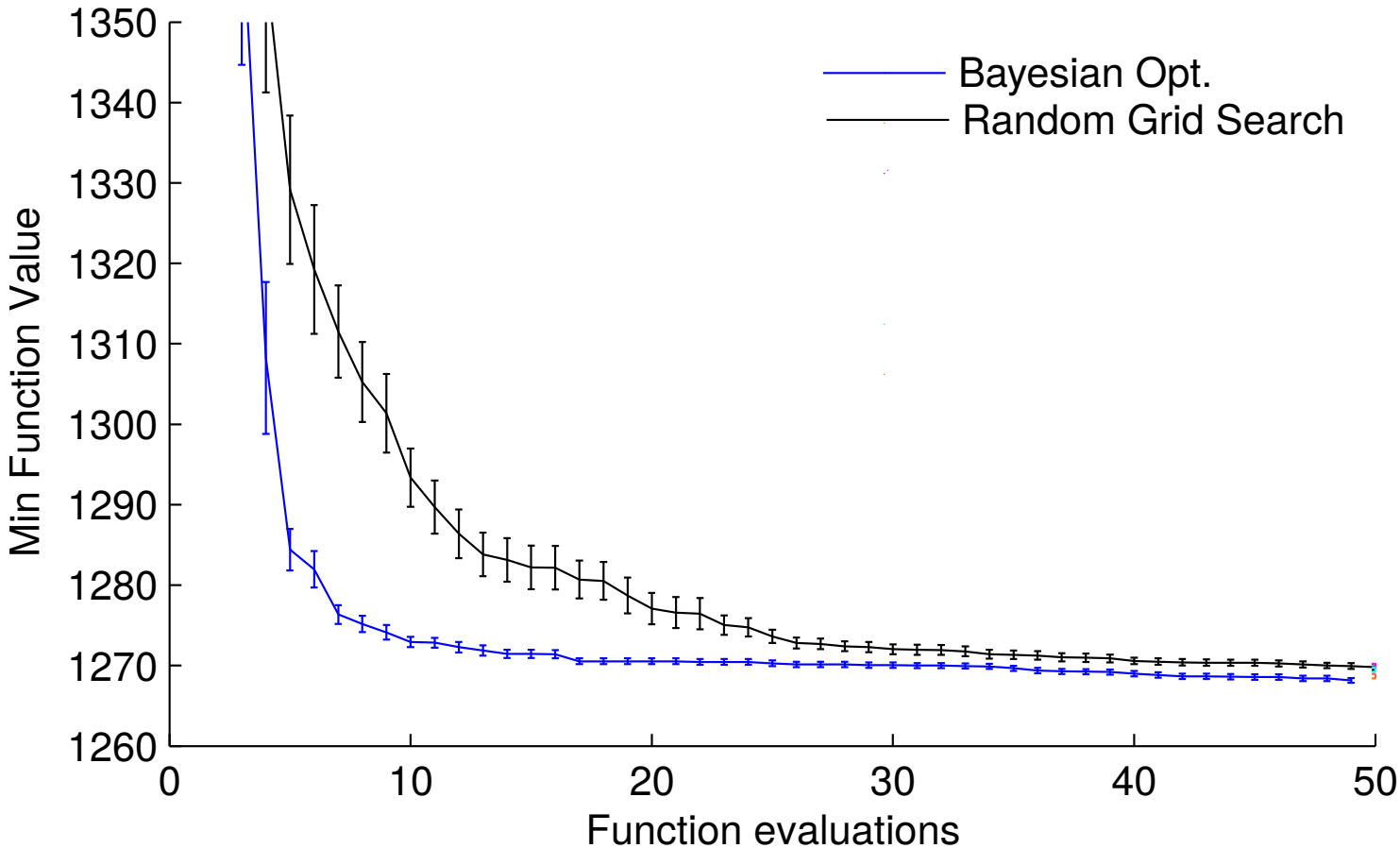
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

The model guides the search focusing on the most-promising regions of the input space!

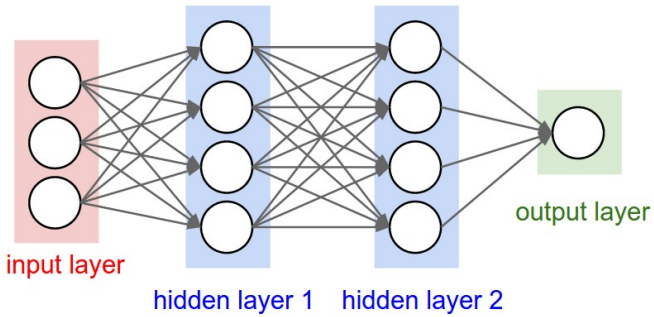
Bayesian Optimization vs. Uniform Exploration

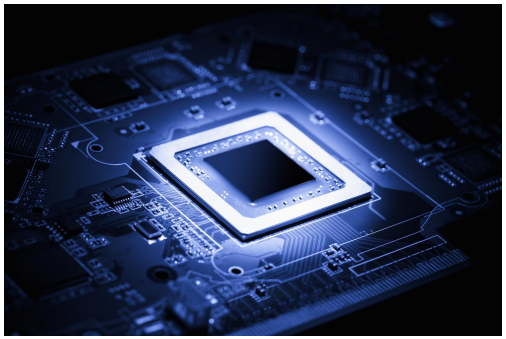


Tuning LDA on a collection of Wikipedia articles (Snoek *et al.*, 2012).

Several Objectives and Constraints

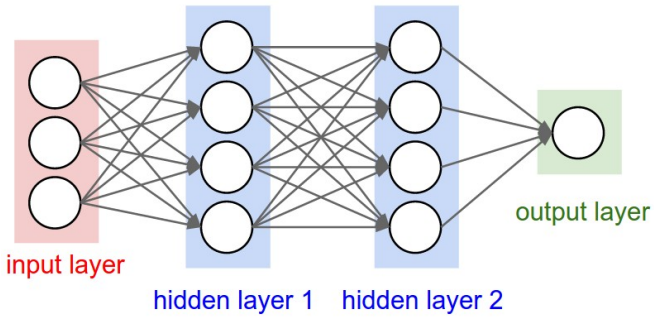
Optimal design of **hardware accelerator** for neural network predictions.

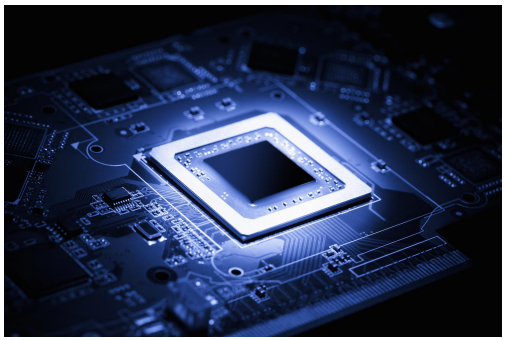




Several Objectives and Constraints

Optimal design of **hardware accelerator** for neural network predictions.



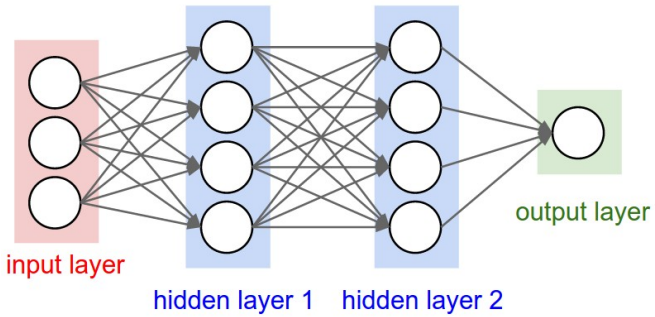


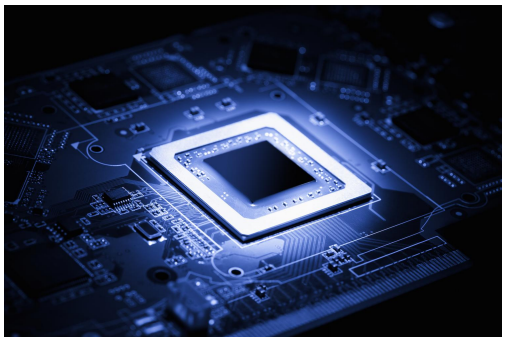
Goals:

- Minimize **prediction error**.
- Minimize **prediction time**.

Several Objectives and Constraints

Optimal design of **hardware accelerator** for neural network predictions.





Goals:

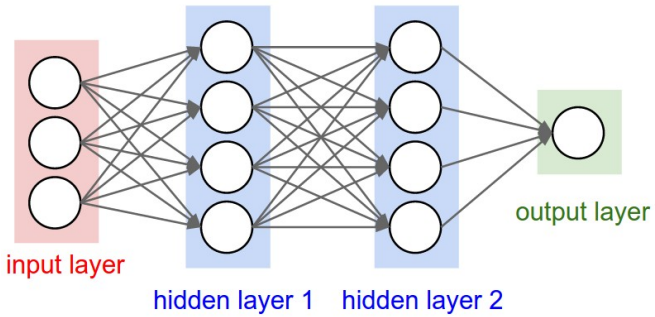
- Minimize **prediction error**.
- Minimize **prediction time**.

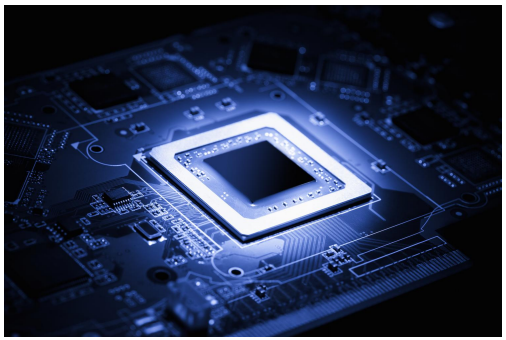
Constrained to:

- **Chip area** below a value.
- **Power consumption** below a level.

Several Objectives and Constraints

Optimal design of **hardware accelerator** for neural network predictions.



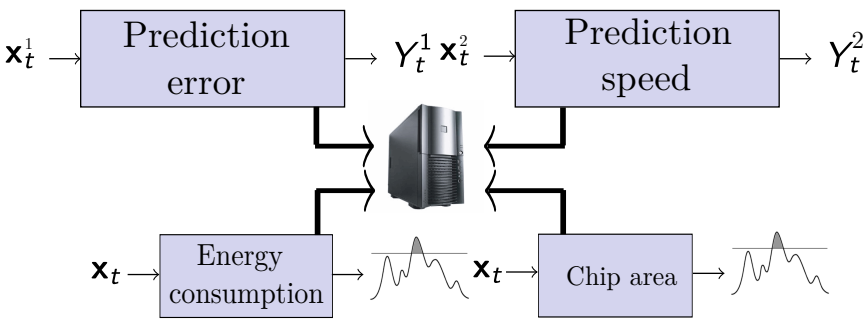


Goals:

- Minimize **prediction error**.
- Minimize **prediction time**.

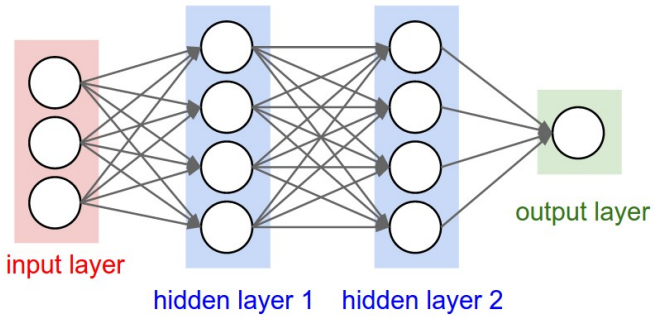
Constrained to:

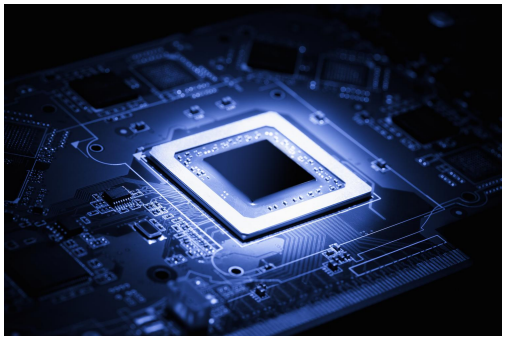
- **Chip area** below a value.
- **Power consumption** below a level.



Several Objectives and Constraints

Optimal design of **hardware accelerator** for neural network predictions.



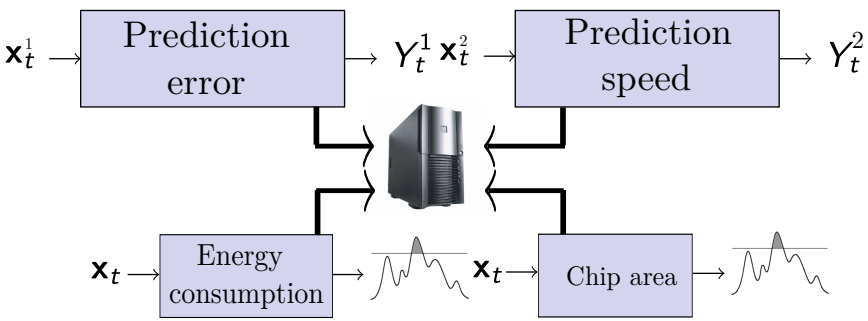


Goals:

- Minimize **prediction error**.
- Minimize **prediction time**.

Constrained to:

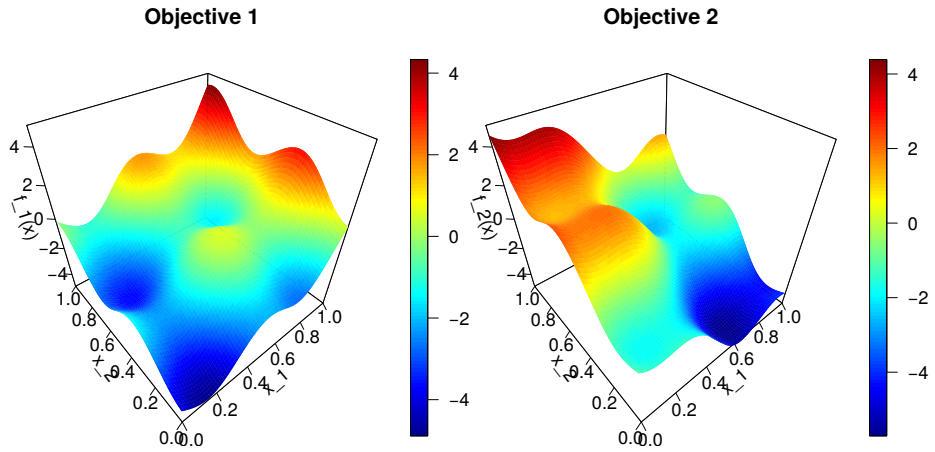
- **Chip area** below a value.
- **Power consumption** below a level.



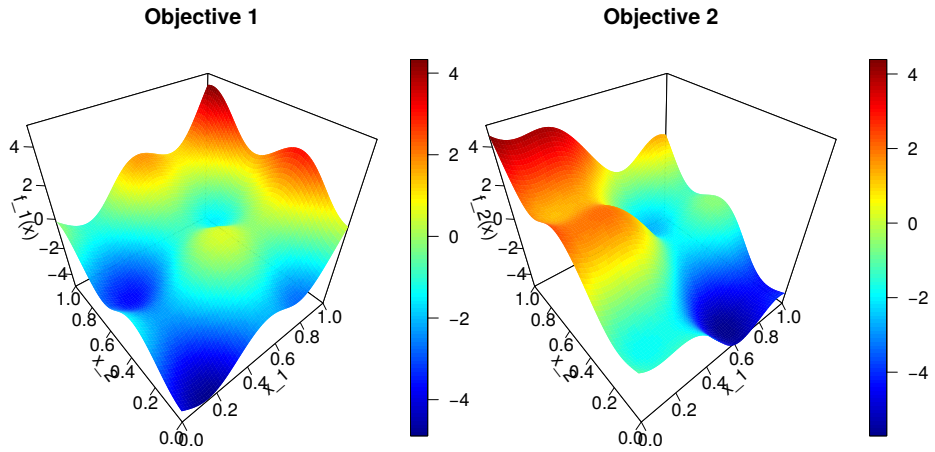
Challenges:

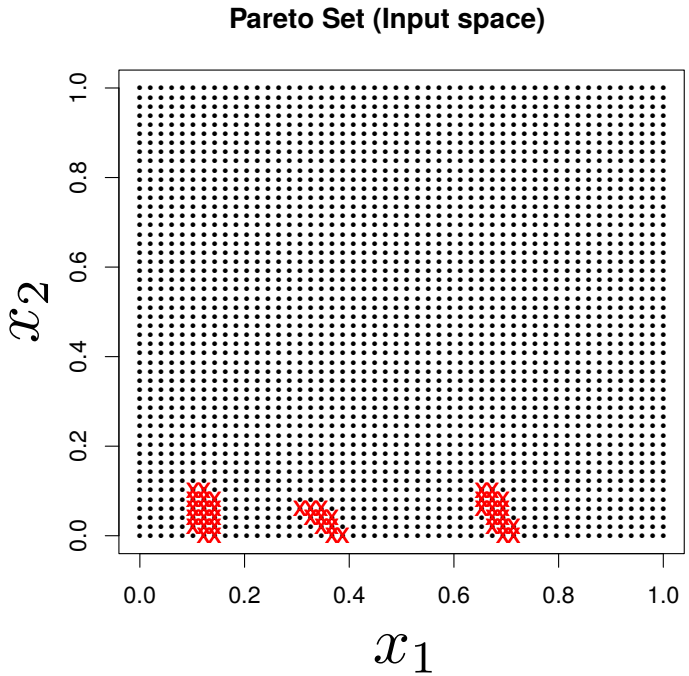
- **Complicated** constraints.
- **Conflictive** objectives.

Constrained Multi-Objective Optimization

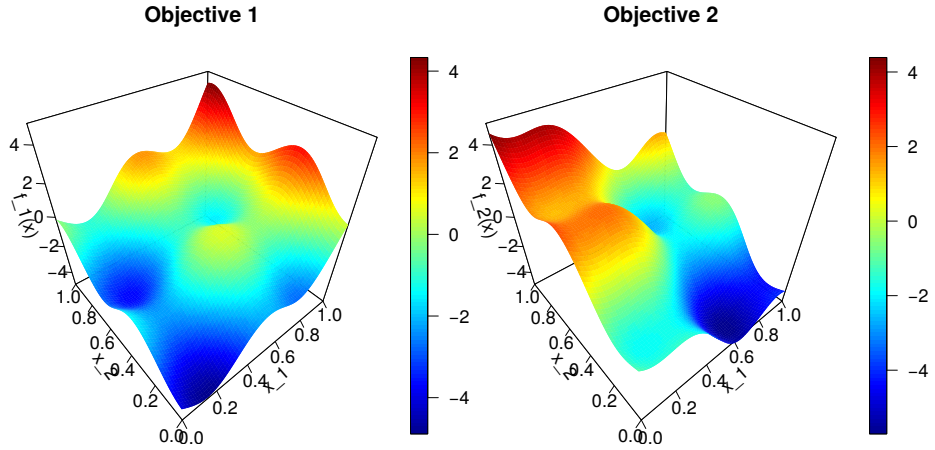


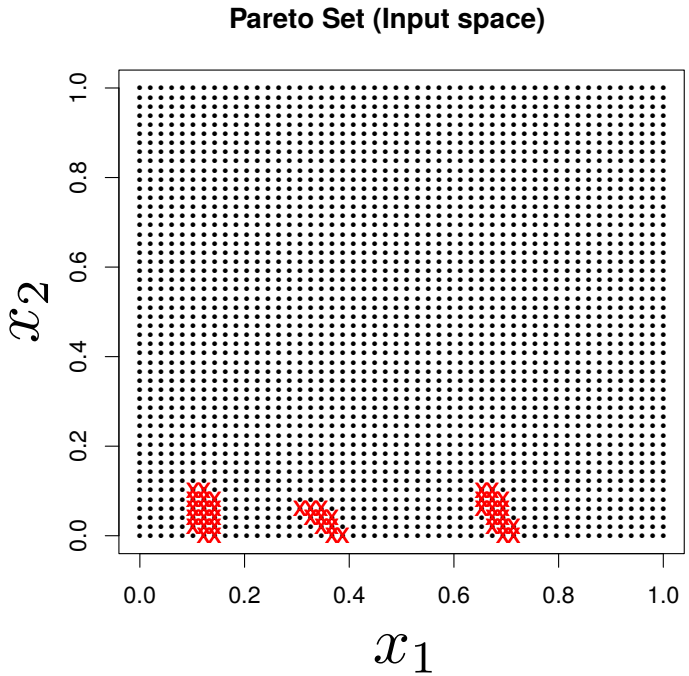
Constrained Multi-Objective Optimization

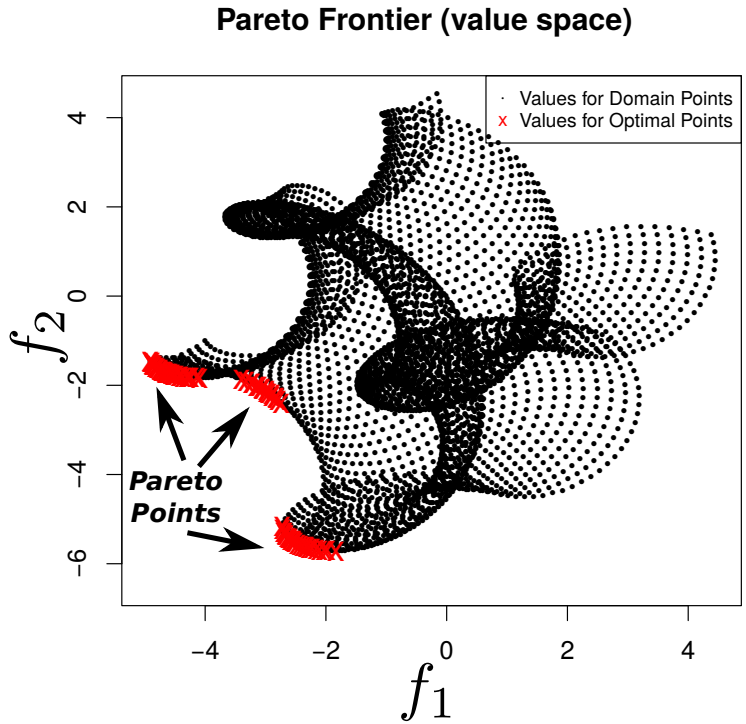




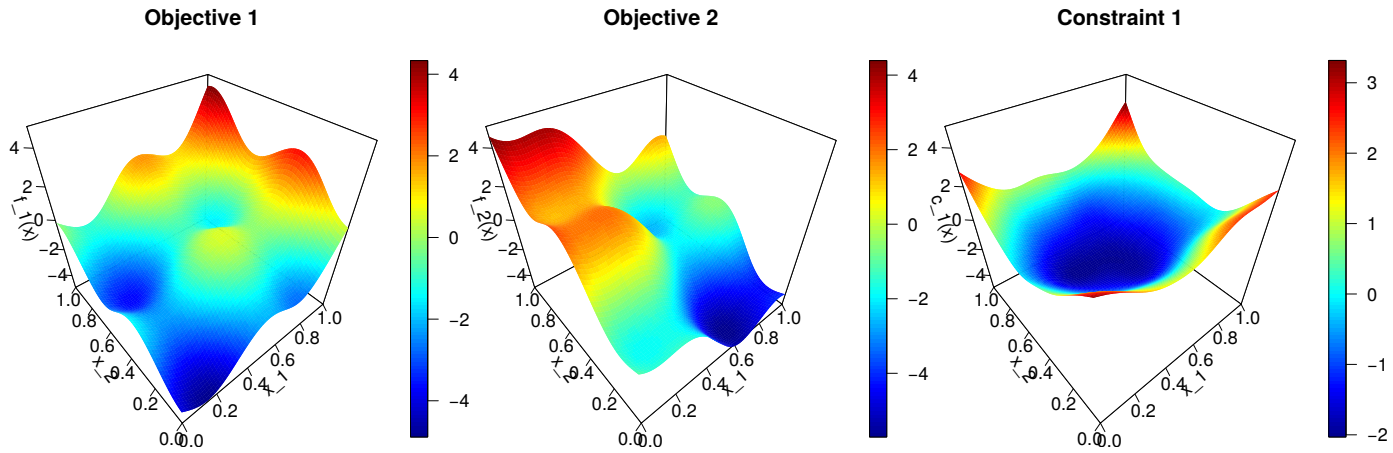
Constrained Multi-Objective Optimization

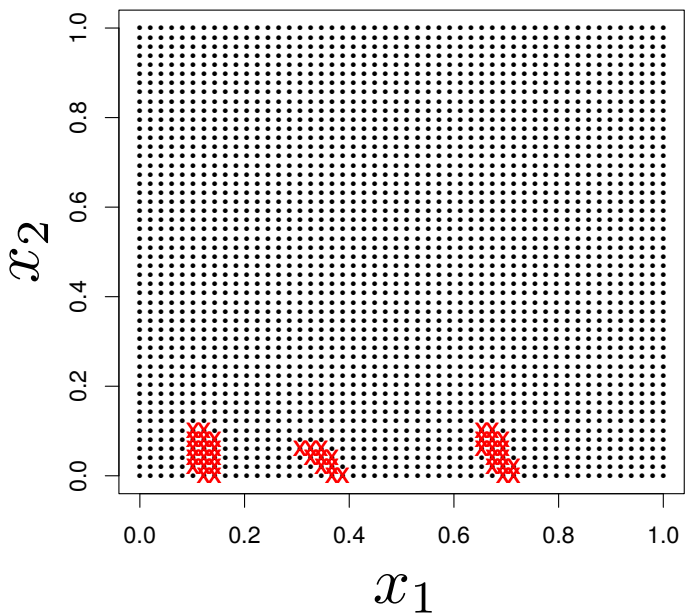


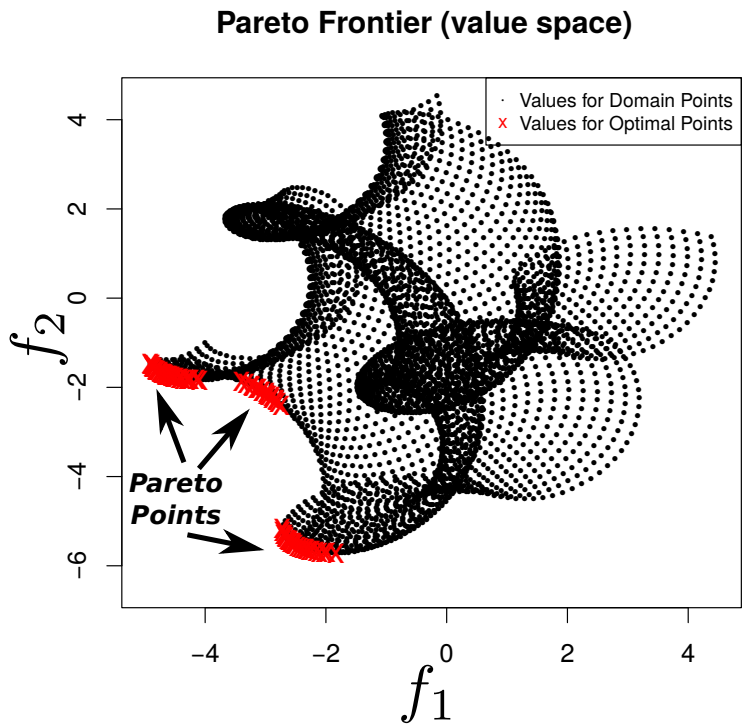




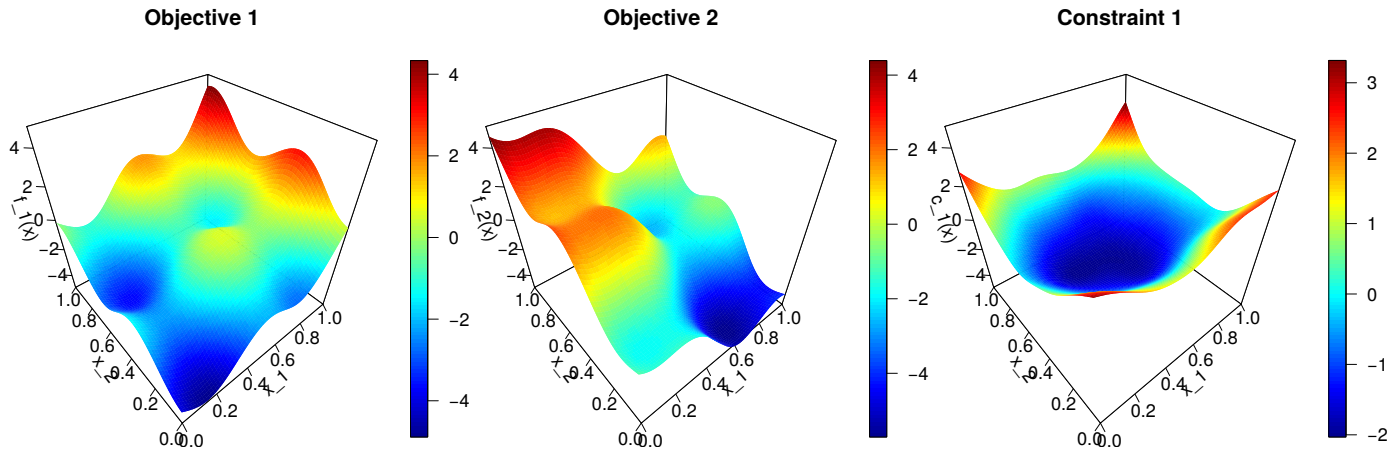
Constrained Multi-Objective Optimization

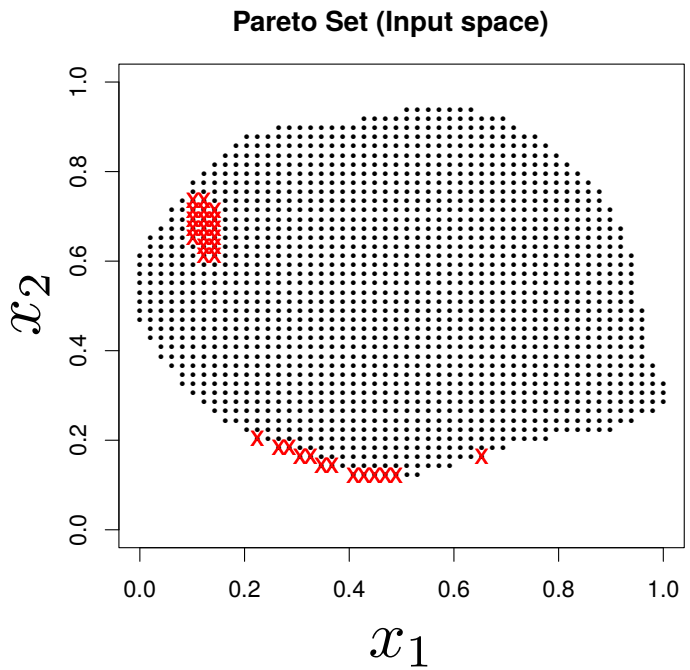


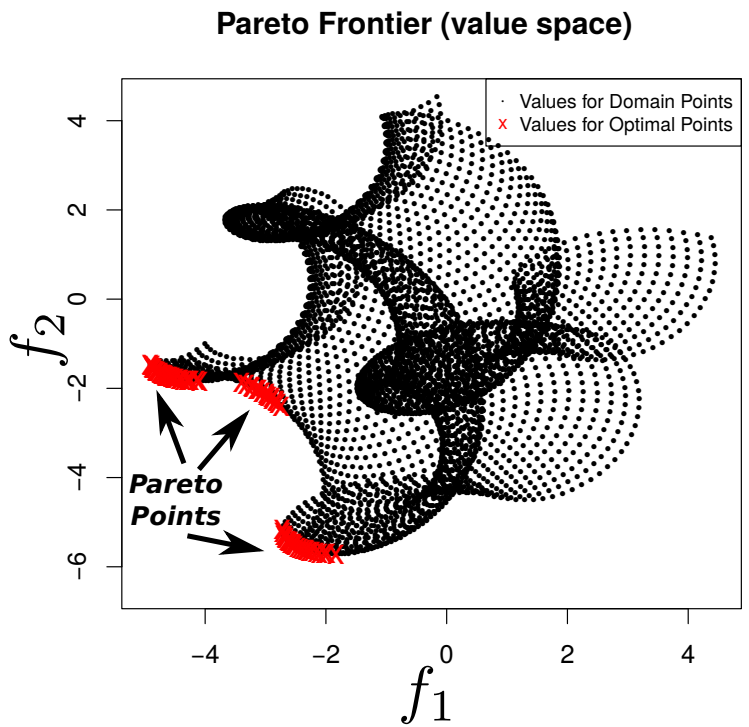




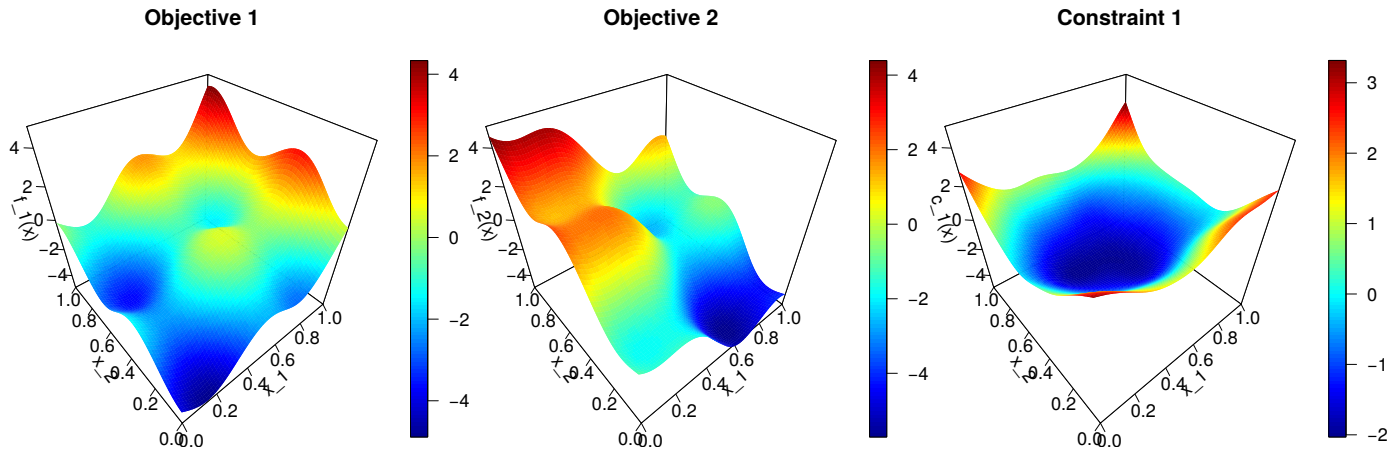
Constrained Multi-Objective Optimization

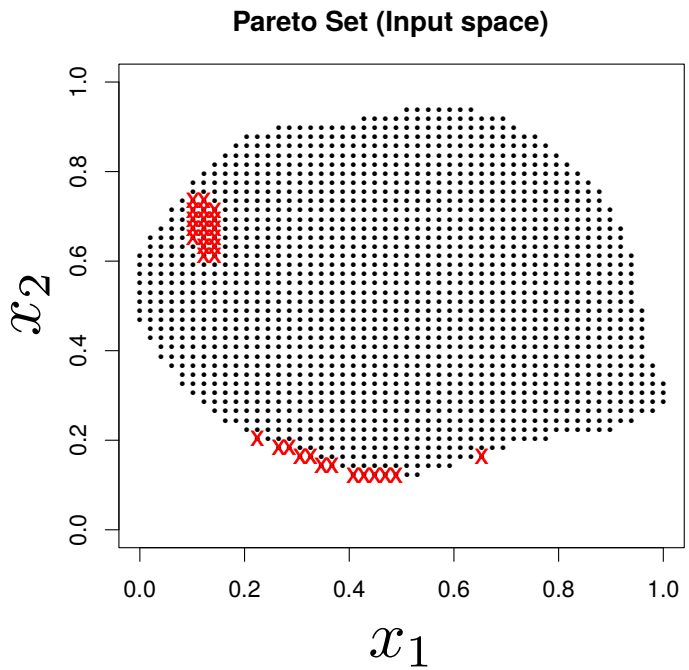


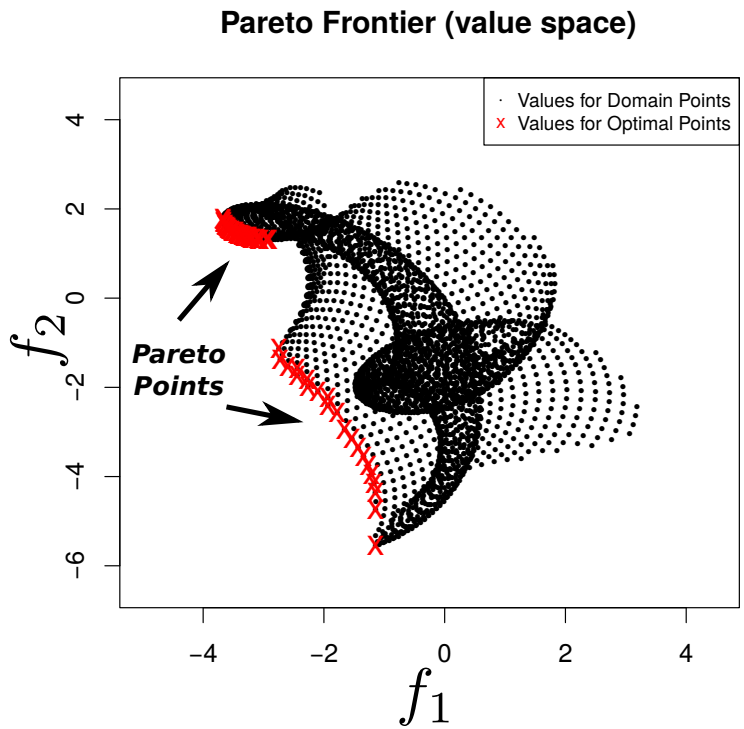




Constrained Multi-Objective Optimization







Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information-based Approach

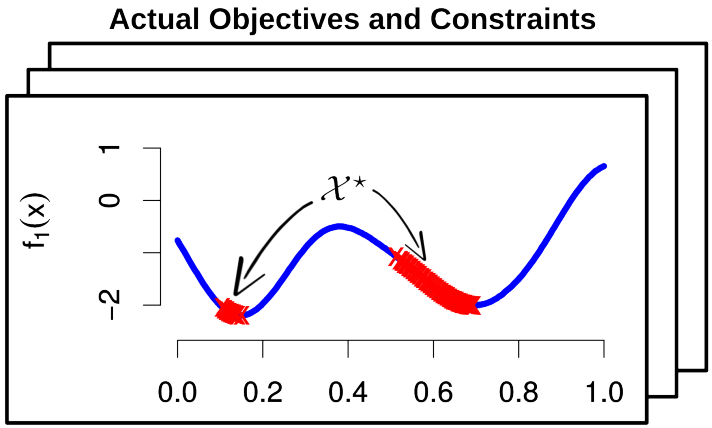
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

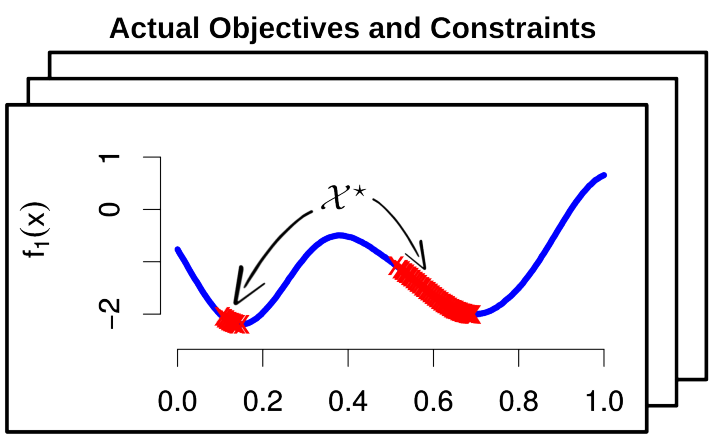
Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

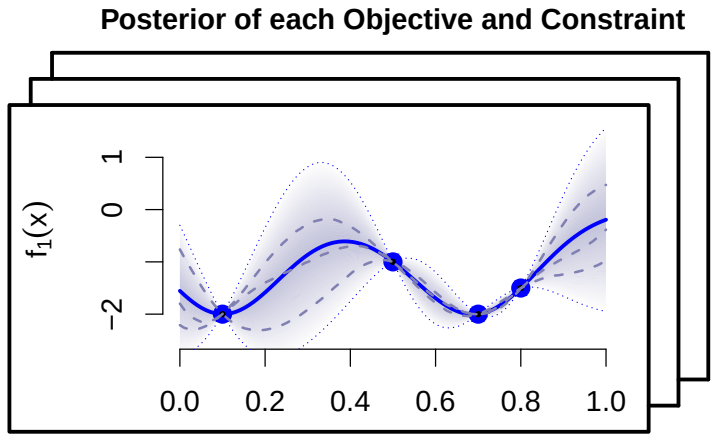


Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

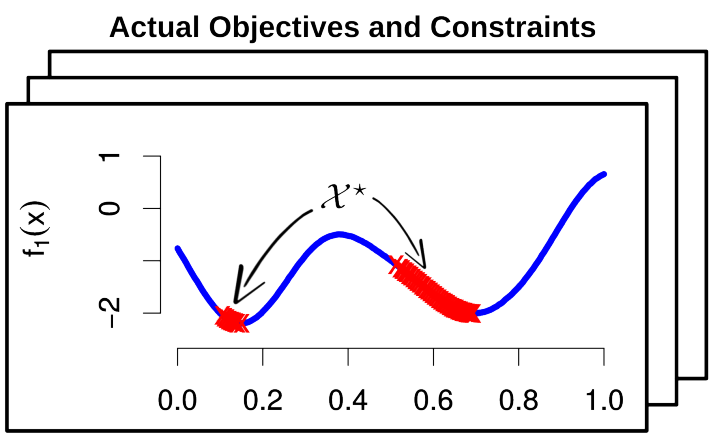


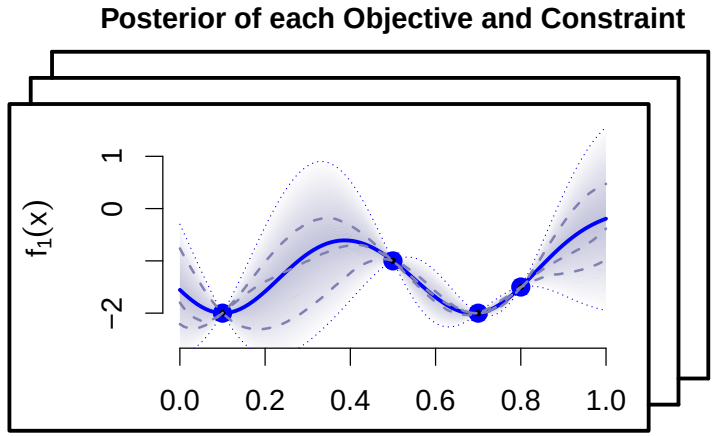


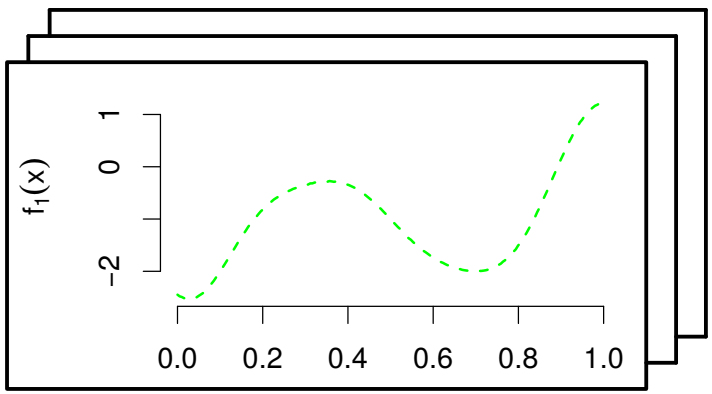
Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.



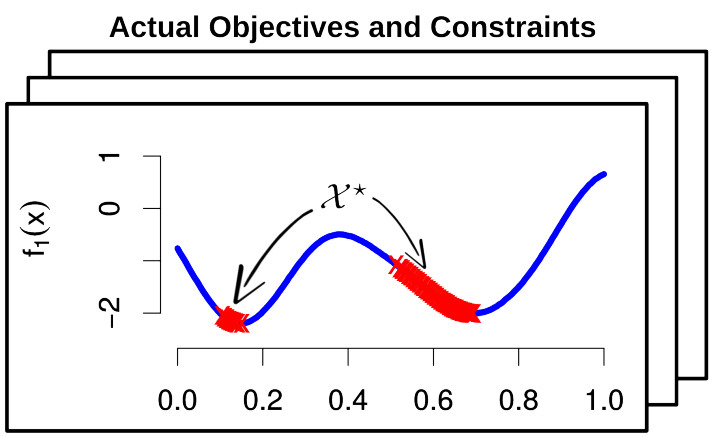


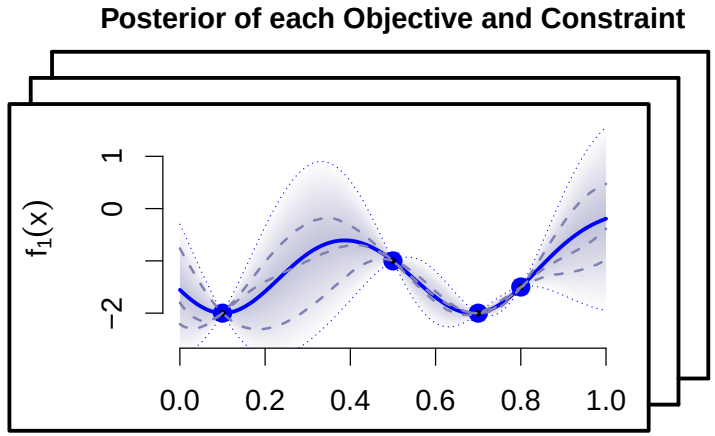


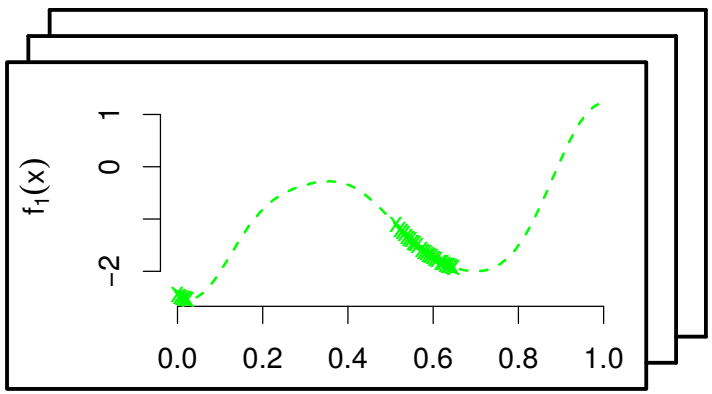
Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.



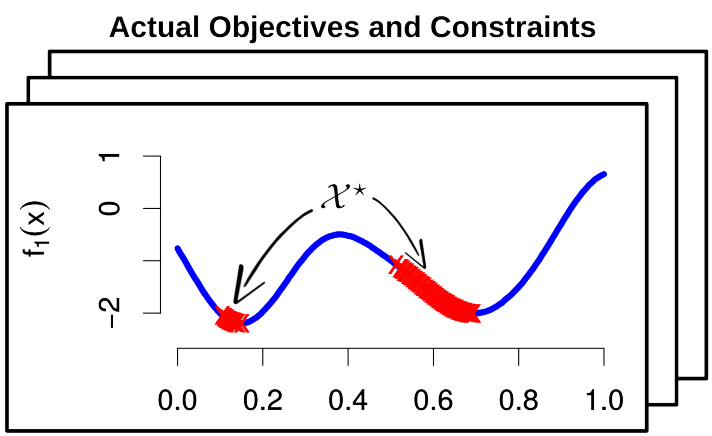


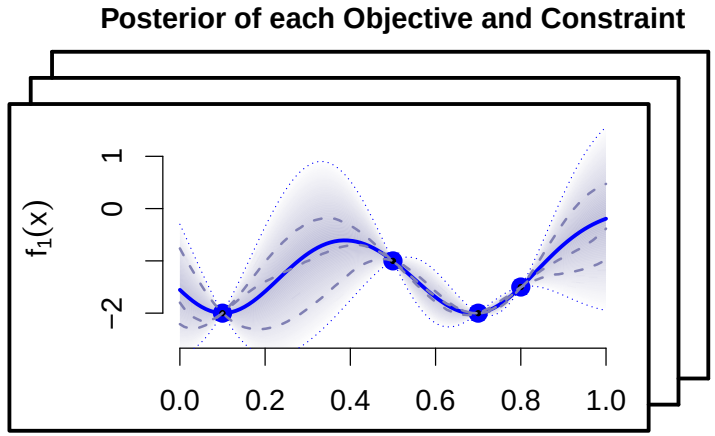


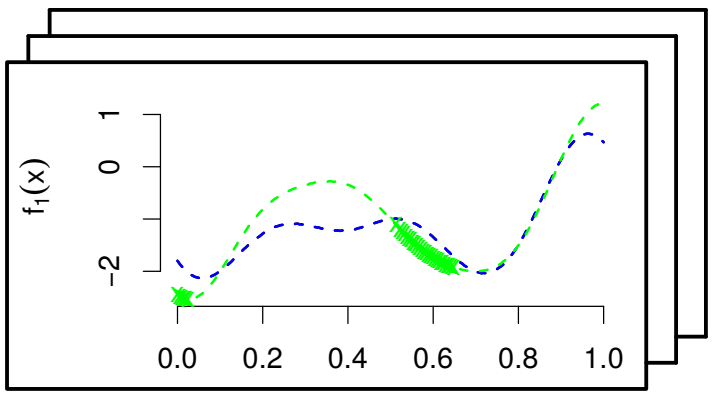
Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.



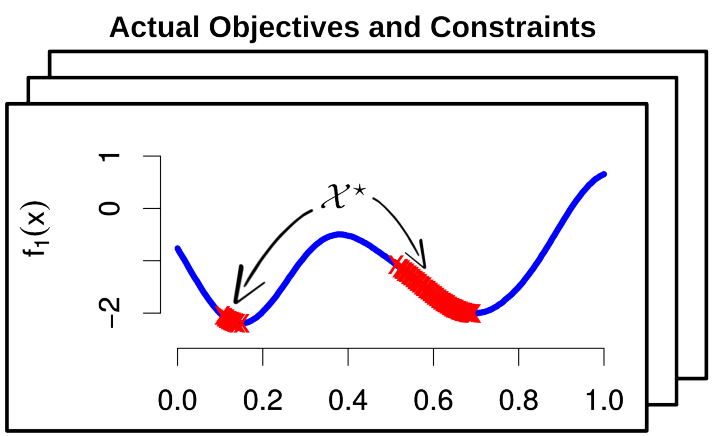


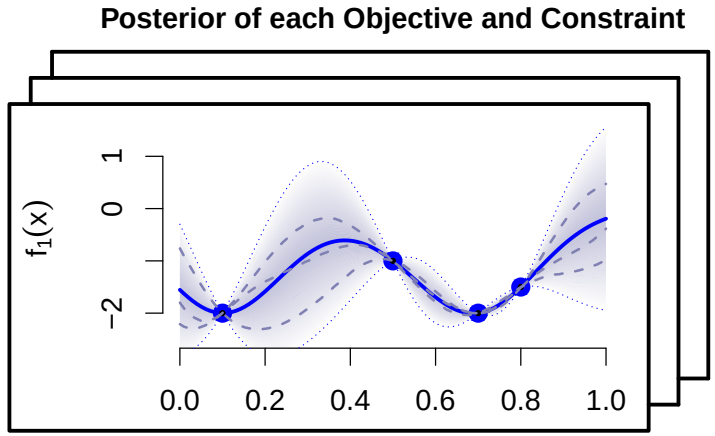


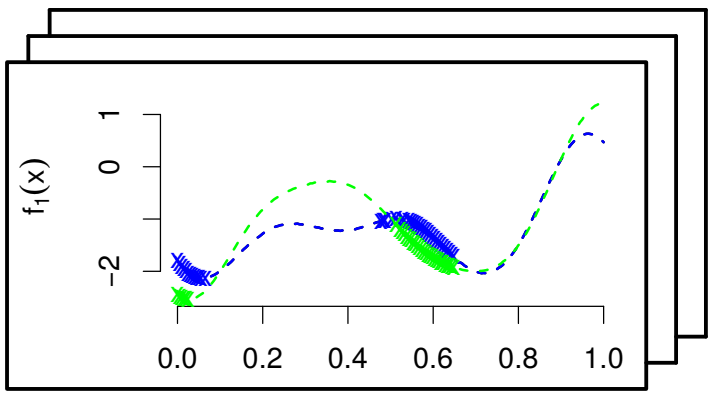
Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.



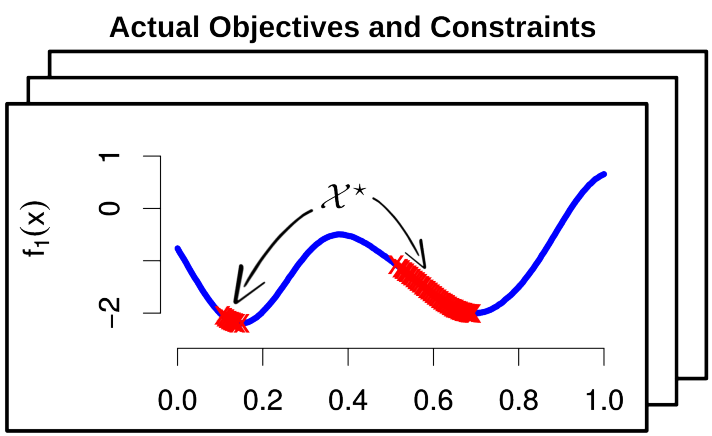


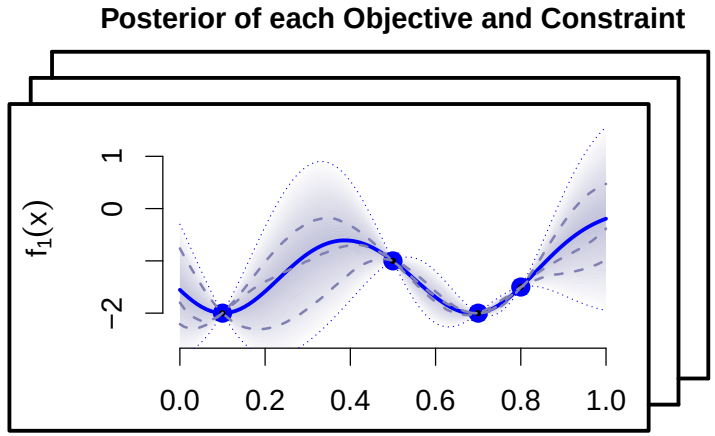


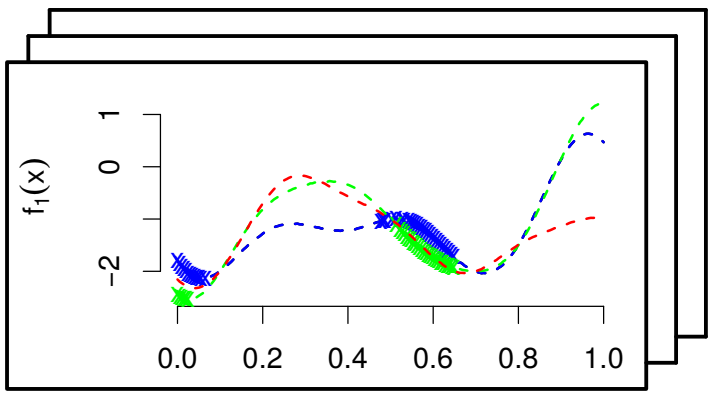
Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.



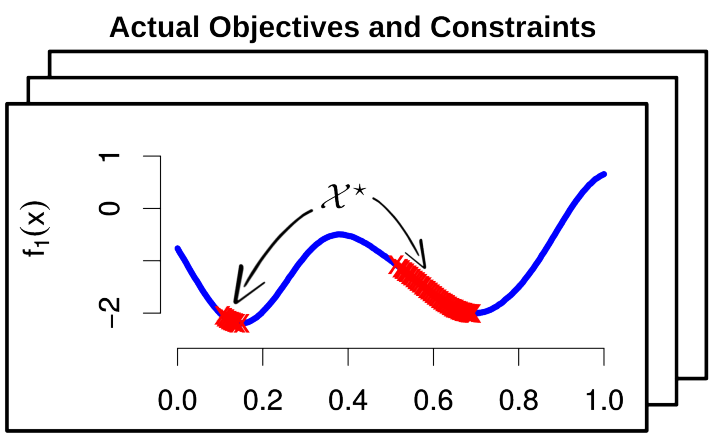


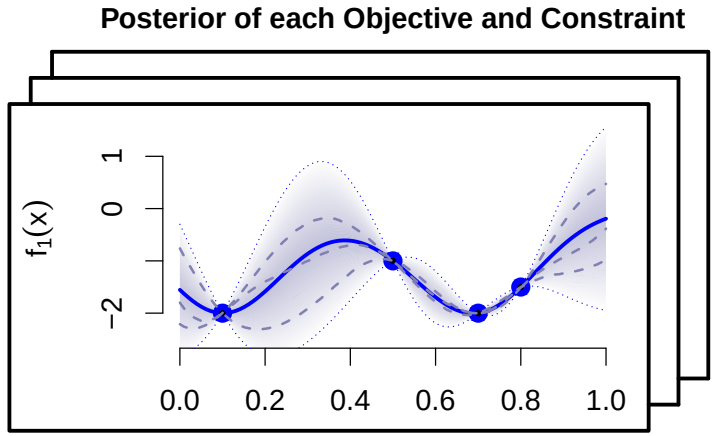


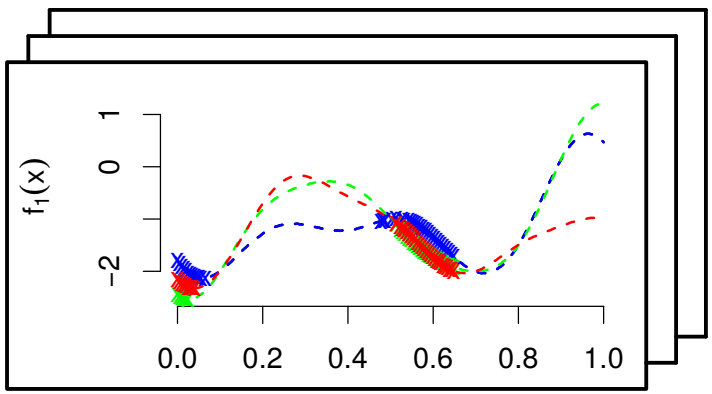
Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.



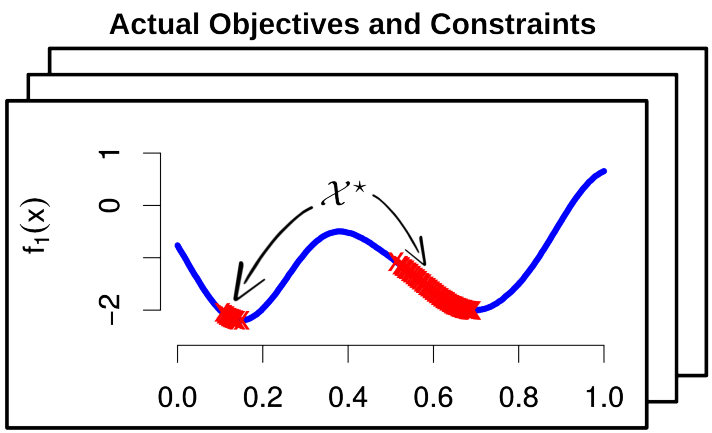


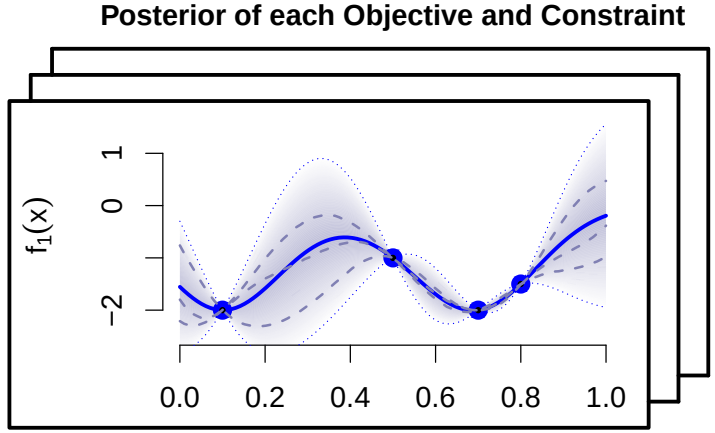


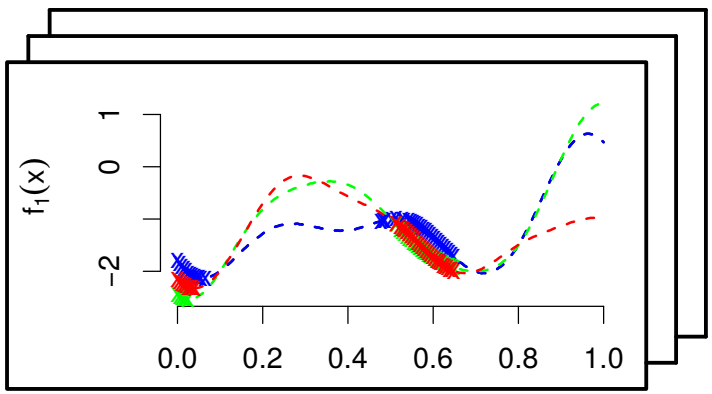
Information-based Approach

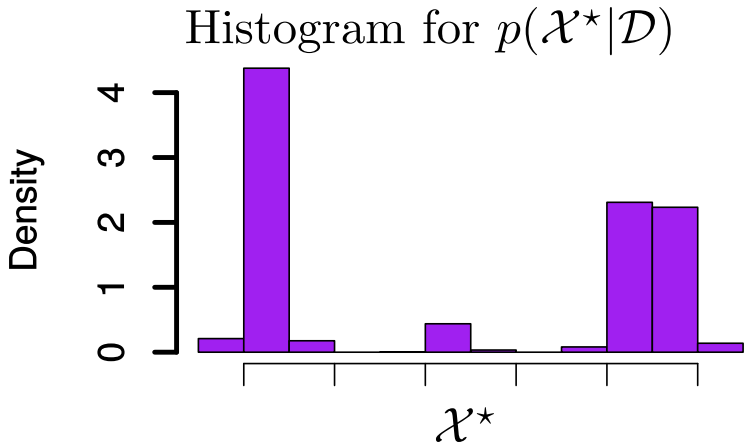
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.









Information-based Approach

The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

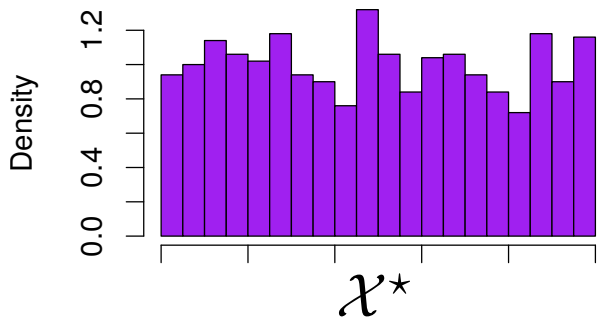
Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

Information-based Approach

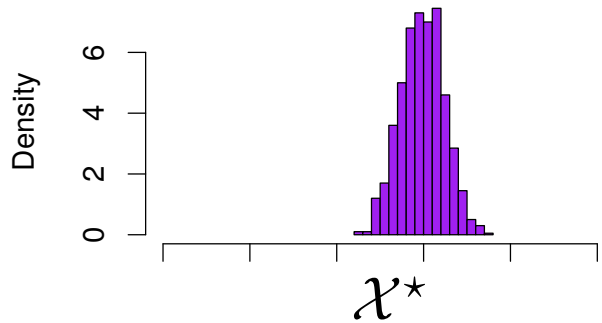
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

High Entropy
Low Information



Low Entropy
High Information

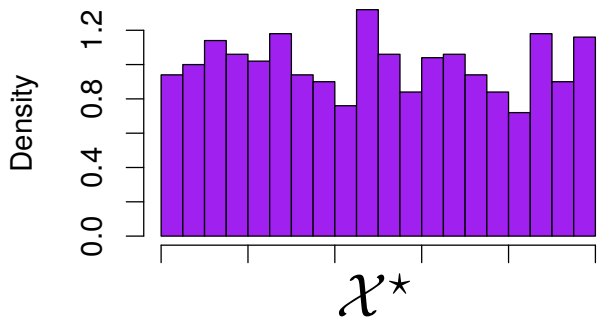


Information-based Approach

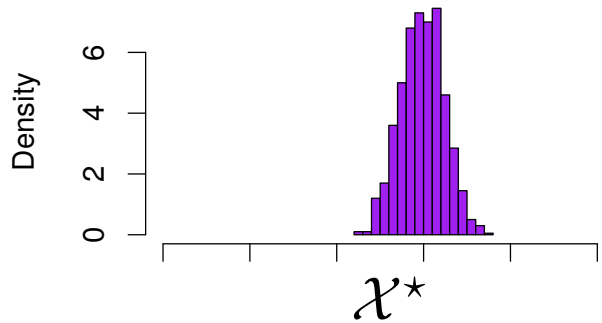
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

High Entropy
Low Information



Low Entropy
High Information



The acquisition function is

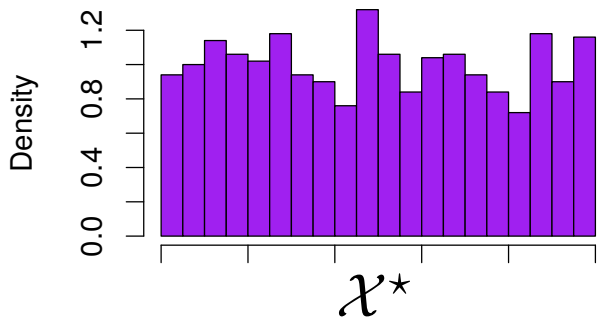
$$\alpha(\mathbf{x}) = \mathsf{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{y}}\left[\mathsf{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{x}, \mathbf{y}\}\right] \middle| \mathcal{D}_t, \mathbf{x}\right] \quad (1)$$

Information-based Approach

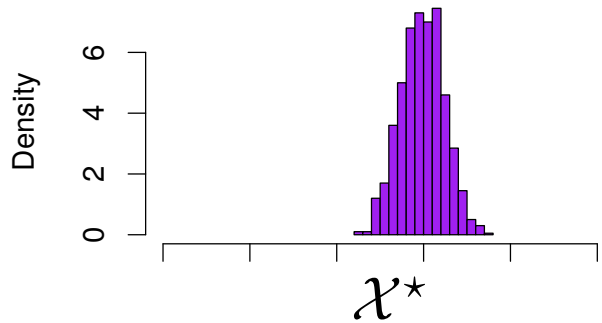
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

High Entropy
Low Information



Low Entropy
High Information



The acquisition function is

$$\alpha(\mathbf{x}) = \mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{x}, \mathbf{y}\}\right] \middle| \mathcal{D}_t, \mathbf{x}\right] \quad (1)$$

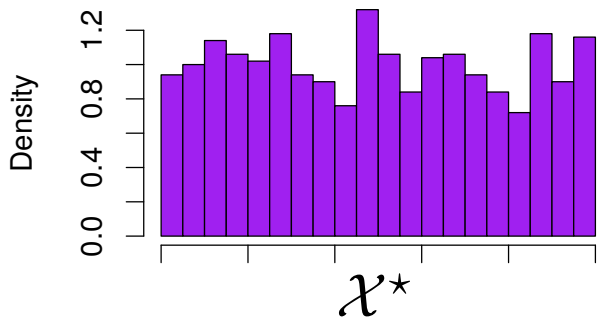
How much we know about $\mathcal{X}^\star$now.

Information-based Approach

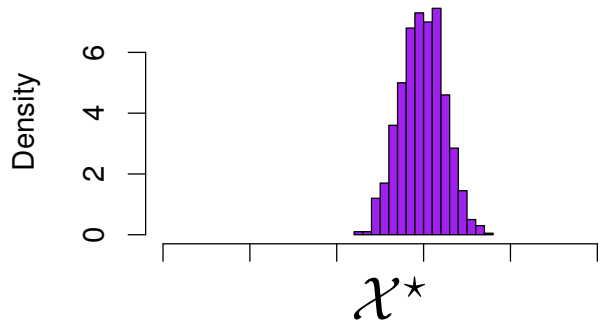
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

High Entropy
Low Information



Low Entropy
High Information



The acquisition function is

$$\alpha(\mathbf{x}) = \mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{x}, \mathbf{y}\}\right] \middle| \mathcal{D}_t, \mathbf{x}\right] \quad (1)$$

How much we know about $\mathcal{X}^\star$now.

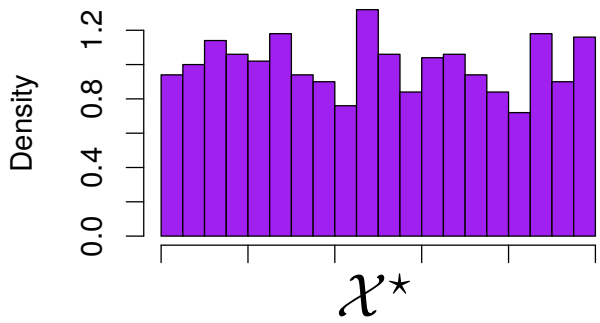
How much we will know about $\mathcal{X}^\star$after collecting y at $\mathbf{x}$.

Information-based Approach

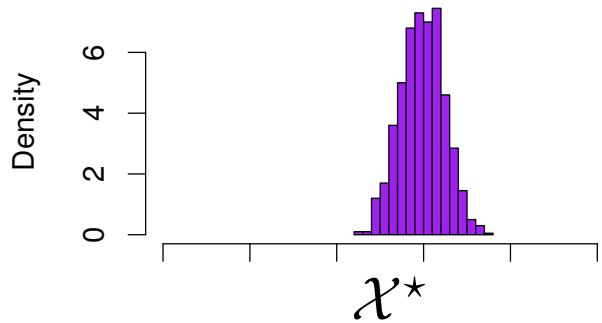
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

High Entropy
Low Information



Low Entropy
High Information



The acquisition function is

$$\alpha(\mathbf{x}) = \mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{x}, \mathbf{y}\}\right] \middle| \mathcal{D}_t, \mathbf{x}\right] \quad (1)$$

How much we know about $\mathcal{X}^\star$now.

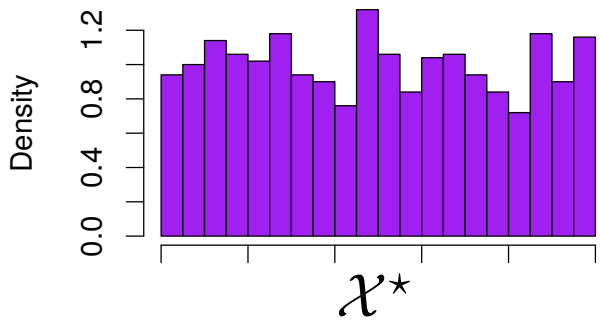
How much we will know about $\mathcal{X}^\star$after collecting y at $\mathbf{x}$.

Information-based Approach

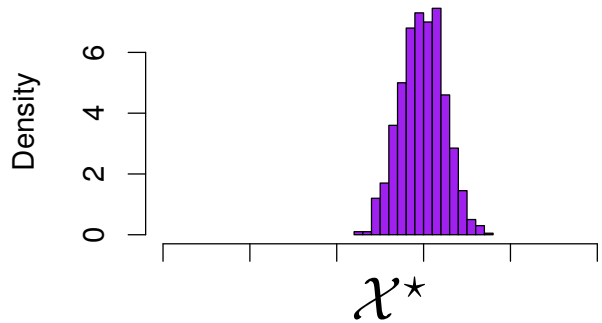
The Pareto set $\mathcal{X}^\star$ in the feasible space is a **random variable**!

Information is measured by the **entropy** of $p(\mathcal{X}^\star|\mathcal{D}_N)$.

High Entropy
Low Information



Low Entropy
High Information



The acquisition function is

$$\alpha(\mathbf{x}) = \mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{x}, \mathbf{y}\}\right] \middle| \mathcal{D}_t, \mathbf{x}\right] \quad (1)$$

How much we know about $\mathcal{X}^\star$now.

How much we will know about $\mathcal{X}^\star$after collecting y at $\mathbf{x}$.

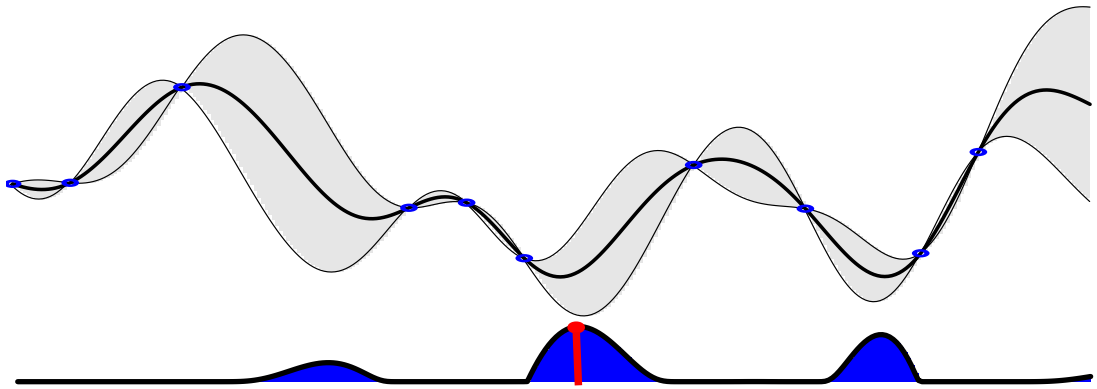
Computing (1) is **very difficult in practice**!

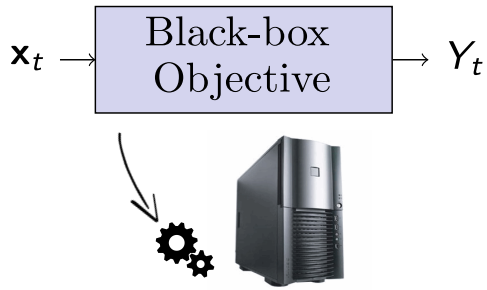
Parallel Bayesian Optimization

Traditional Bayesian optimization is **sequential**!

Parallel Bayesian Optimization

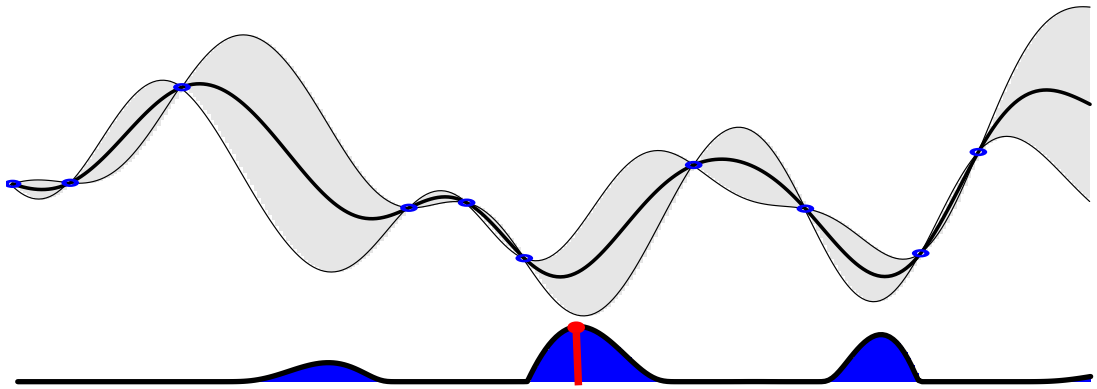
Traditional Bayesian optimization is **sequential**!

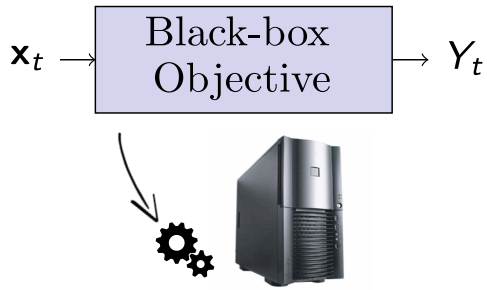




Parallel Bayesian Optimization

Traditional Bayesian optimization is **sequential**!

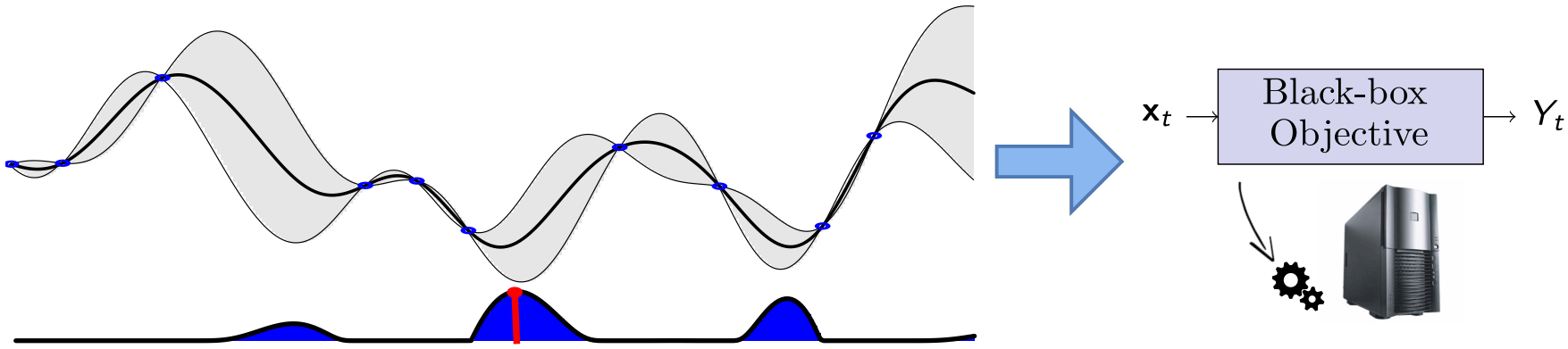




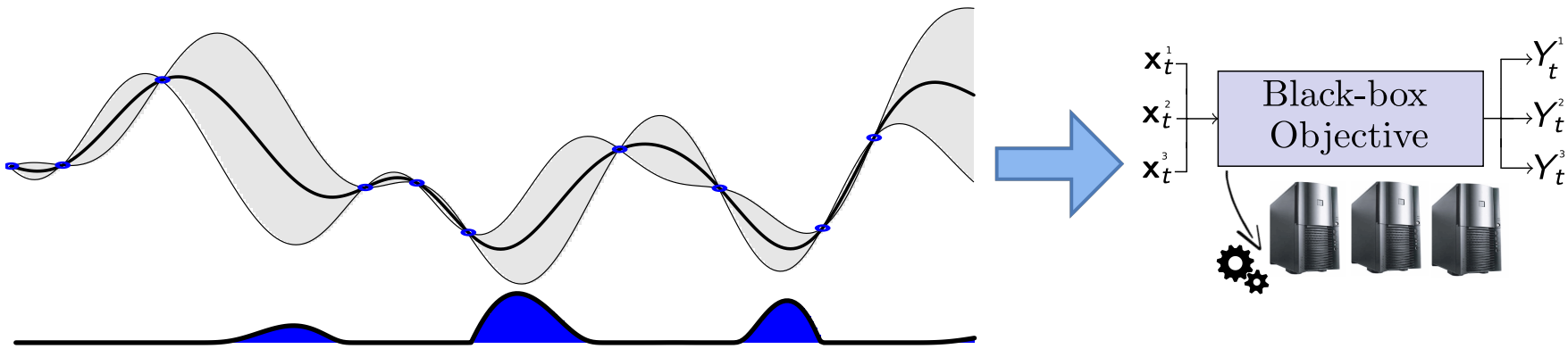
Computing clusters let us do **many things** at once!

Parallel Bayesian Optimization

Traditional Bayesian optimization is **sequential**!

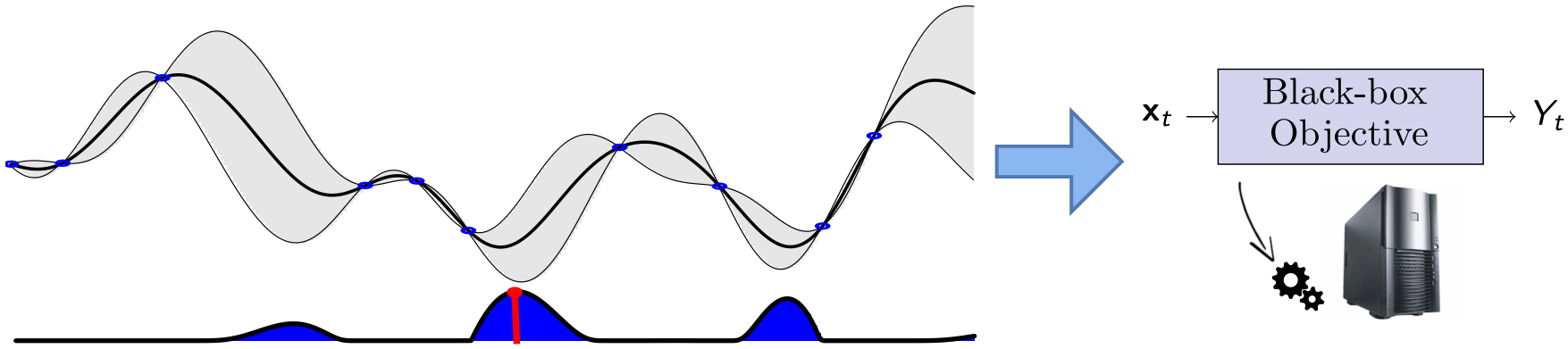


Computing clusters let us do **many things** at once!

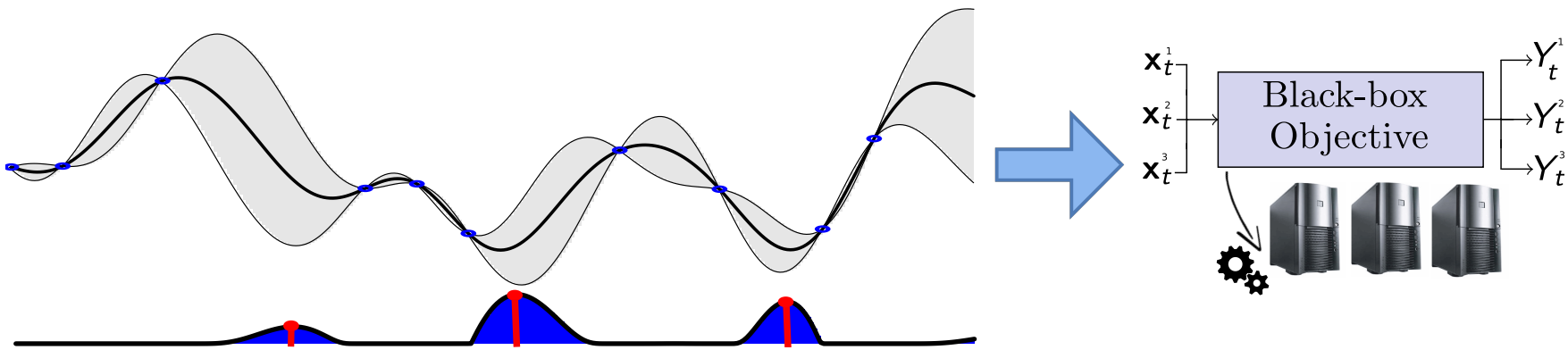


Parallel Bayesian Optimization

Traditional Bayesian optimization is **sequential**!

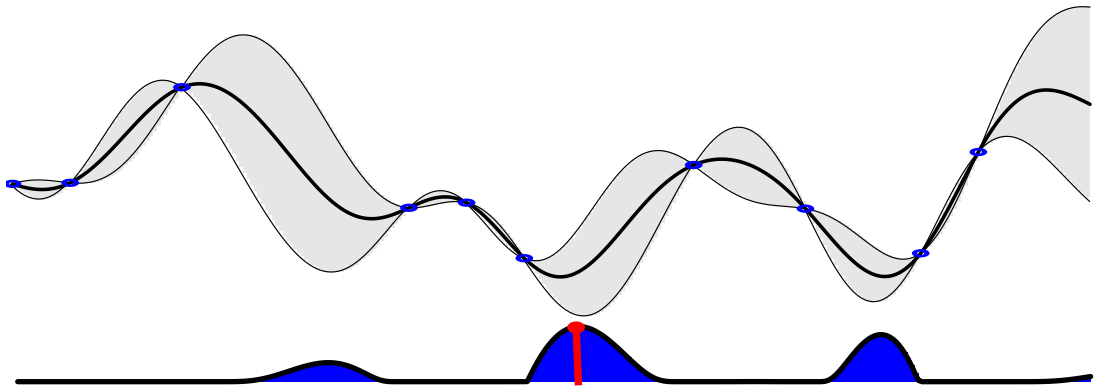


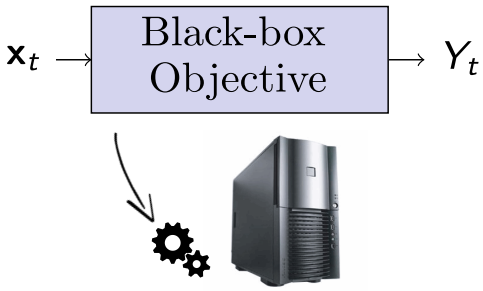
Computing clusters let us do **many things** at once!



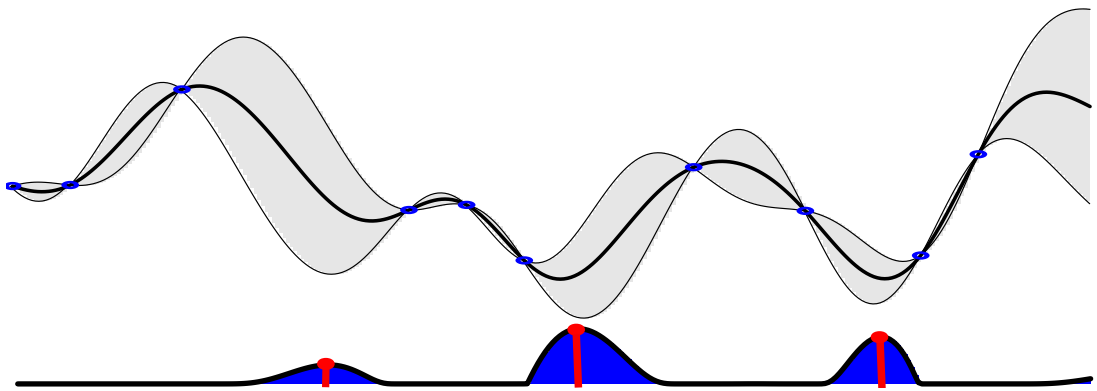
Parallel Bayesian Optimization

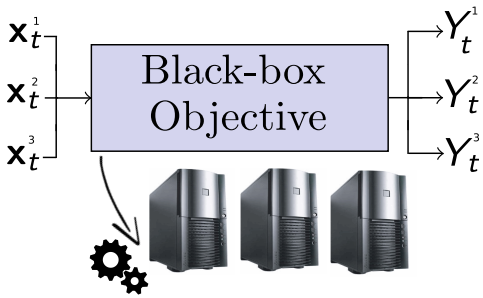
Traditional Bayesian optimization is **sequential**!





Computing clusters let us do **many things** at once!





Parallel experiments should be highly informative but different!

Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

$$\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{Y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{Y}, \mathbf{X}\}\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathbf{Y}, \mathcal{X}^\star) \qquad \text{(Parallel ESMOC)}$$

Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

$$\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{Y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{Y}, \mathbf{X}\}\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathbf{Y}, \mathcal{X}^\star) \qquad \text{(Parallel ESMOC)}$$

$$\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}\right] - \mathbb{E}_{\mathcal{X}^\star}\left[\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}, \mathcal{X}^\star\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathcal{X}^\star, \mathbf{Y}) \qquad \text{(Parallel PESMOC)}$$

Parallel Predictive Entropy Search

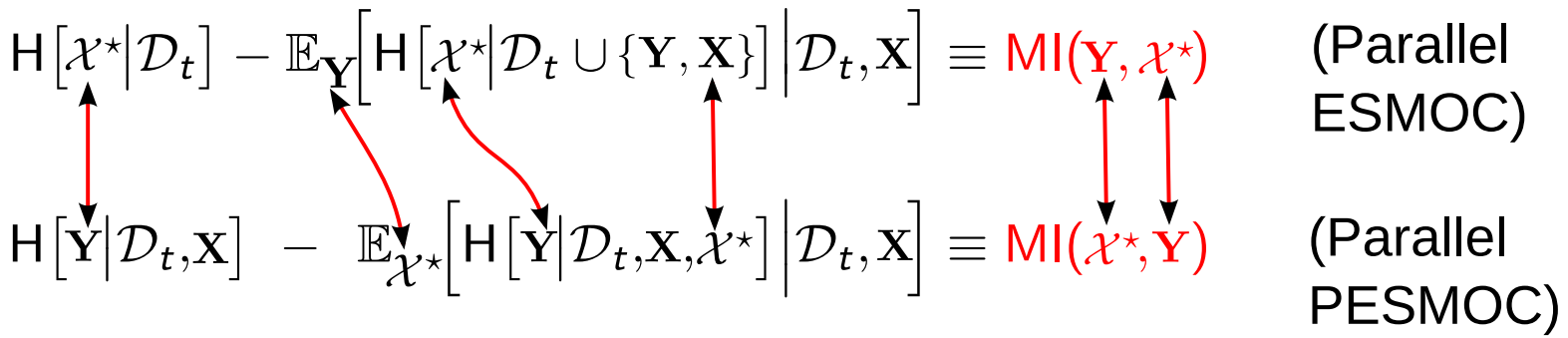
Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

$$\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{Y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{Y}, \mathbf{X}\}\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathbf{Y}, \mathcal{X}^\star) \quad \text{(Parallel ESMOC)}$$

$$\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}\right] - \mathbb{E}_{\mathcal{X}^\star}\left[\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}, \mathcal{X}^\star\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathcal{X}^\star, \mathbf{Y}) \quad \text{(Parallel PESMOC)}$$

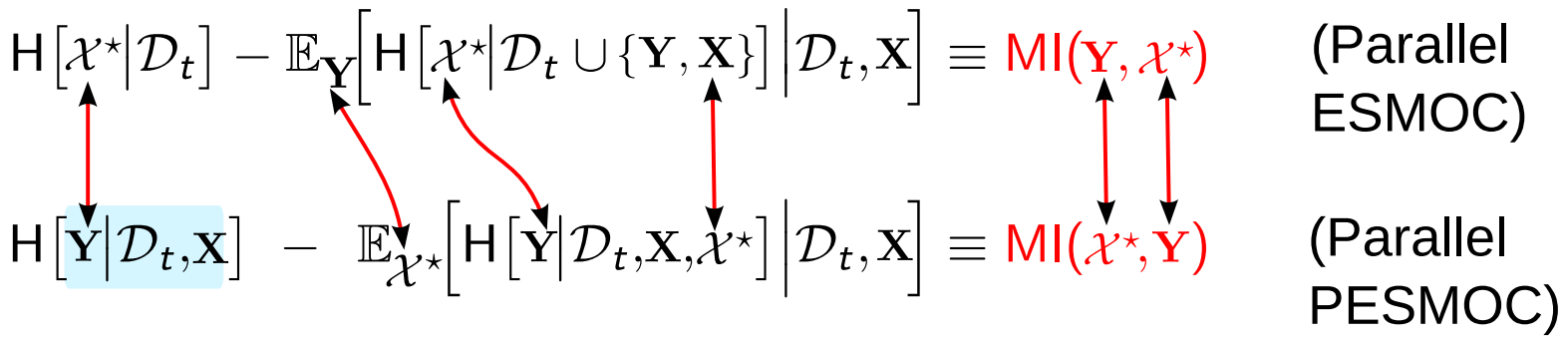
Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.



Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.



Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

$$\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{Y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{Y}, \mathbf{X}\}\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathbf{Y}, \mathcal{X}^\star)$$

(Parallel ESMOC)

$$\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}\right] - \mathbb{E}_{\mathcal{X}^\star}\left[\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}, \mathcal{X}^\star\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathcal{X}^\star, \mathbf{Y})$$

(Parallel PESMOC)

Gaussian distribution

Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

$$\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{Y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{Y}, \mathbf{X}\}\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathbf{Y}, \mathcal{X}^\star)$$

(Parallel ESMOC)

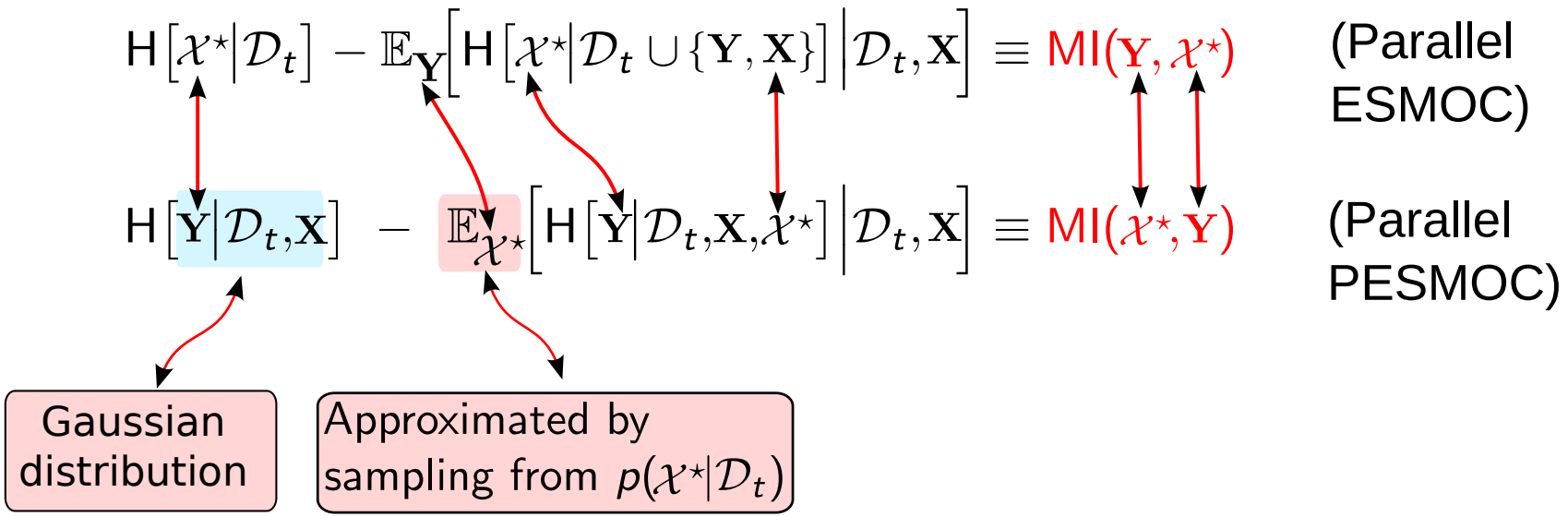
$$\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}\right] - \mathbb{E}_{\mathcal{X}^\star}\left[\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}, \mathcal{X}^\star\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathcal{X}^\star, \mathbf{Y})$$

(Parallel PESMOC)

Gaussian distribution

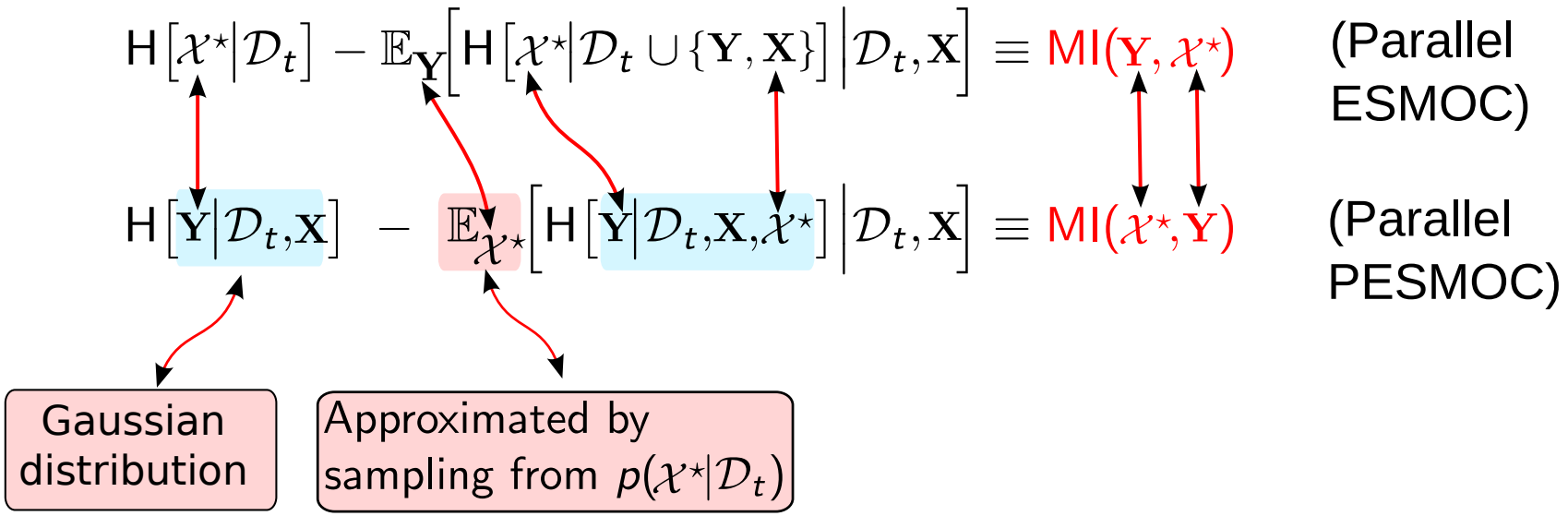
Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.



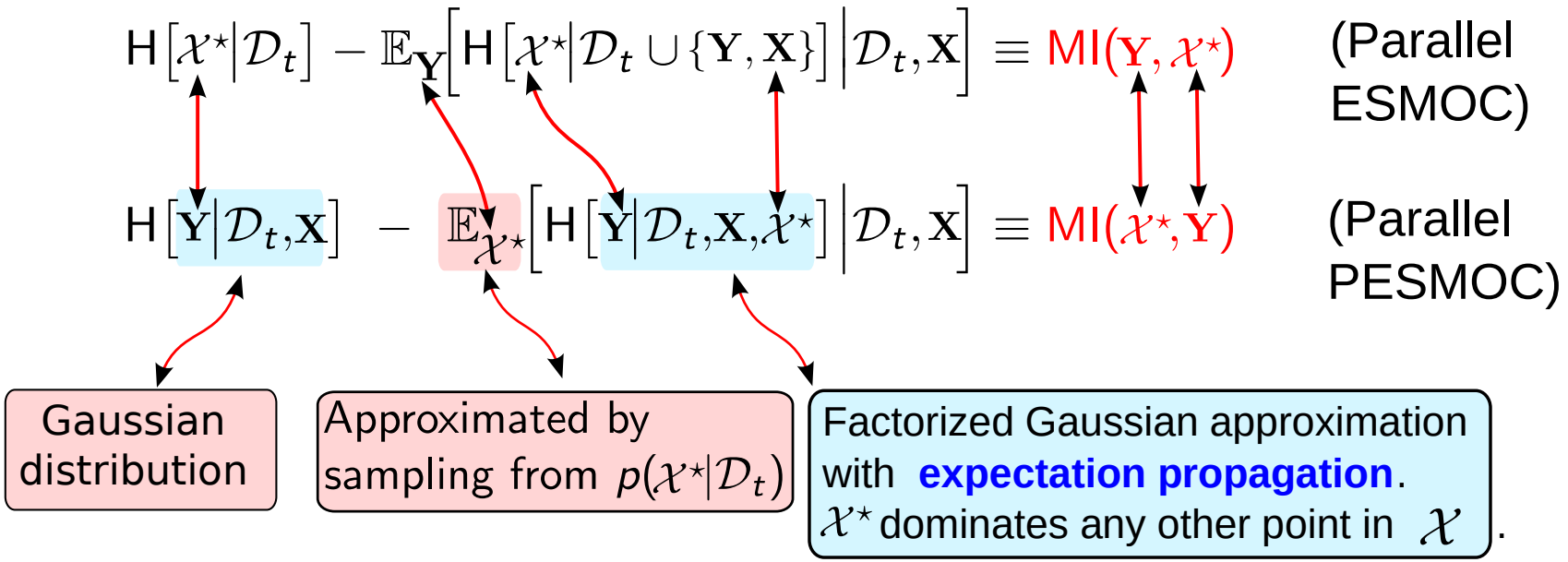
Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.



Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.



(Minka, 2001)

Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

$$\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{Y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{Y}, \mathbf{X}\}\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathbf{Y}, \mathcal{X}^\star)$$

(Parallel ESMOC)

$$\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}\right] - \mathbb{E}_{\mathcal{X}^\star}\left[\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}, \mathcal{X}^\star\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathcal{X}^\star, \mathbf{Y})$$

(Parallel PESMOC)

Gaussian distribution

Approximated by sampling from $p(\mathcal{X}^\star|\mathcal{D}_t)$

Factorized Gaussian approximation with **expectation propagation**.

$\mathcal{X}^\star$ dominates any other point in $\mathcal{X}$.

$$\alpha(\mathbf{x}) \approx \sum_{c=1}^{C} \log|\mathbf{V}_c^{\mathrm{PD}}(\mathbf{X})| - \frac{1}{M}\sum_{m=1}^{M}\left(\sum_{c=1}^{C}\log|\mathbf{V}_c^{\mathrm{CPD}}(\mathbf{X}|\mathcal{X}_m^\star)|\right) + \sum_{k=1}^{K}\log|\mathbf{V}_k^{\mathrm{PD}}(\mathbf{X})| - \frac{1}{M}\sum_{m=1}^{M}\left(\sum_{k=1}^{K}\log|\mathbf{V}_k^{\mathrm{CPD}}(\mathbf{X}|\mathcal{X}_m^\star)|\right)$$

(Minka, 2001)

Parallel Predictive Entropy Search

Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.

$$\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t\right] - \mathbb{E}_{\mathbf{Y}}\left[\mathrm{H}\left[\mathcal{X}^\star \middle| \mathcal{D}_t \cup \{\mathbf{Y}, \mathbf{X}\}\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathbf{Y}, \mathcal{X}^\star)$$

(Parallel ESMOC)

$$\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}\right] - \mathbb{E}_{\mathcal{X}^\star}\left[\mathrm{H}\left[\mathbf{Y} \middle| \mathcal{D}_t, \mathbf{X}, \mathcal{X}^\star\right] \middle| \mathcal{D}_t, \mathbf{X}\right] \equiv \mathrm{MI}(\mathcal{X}^\star, \mathbf{Y})$$

(Parallel PESMOC)

Gaussian distribution

Approximated by sampling from $p(\mathcal{X}^\star|\mathcal{D}_t)$

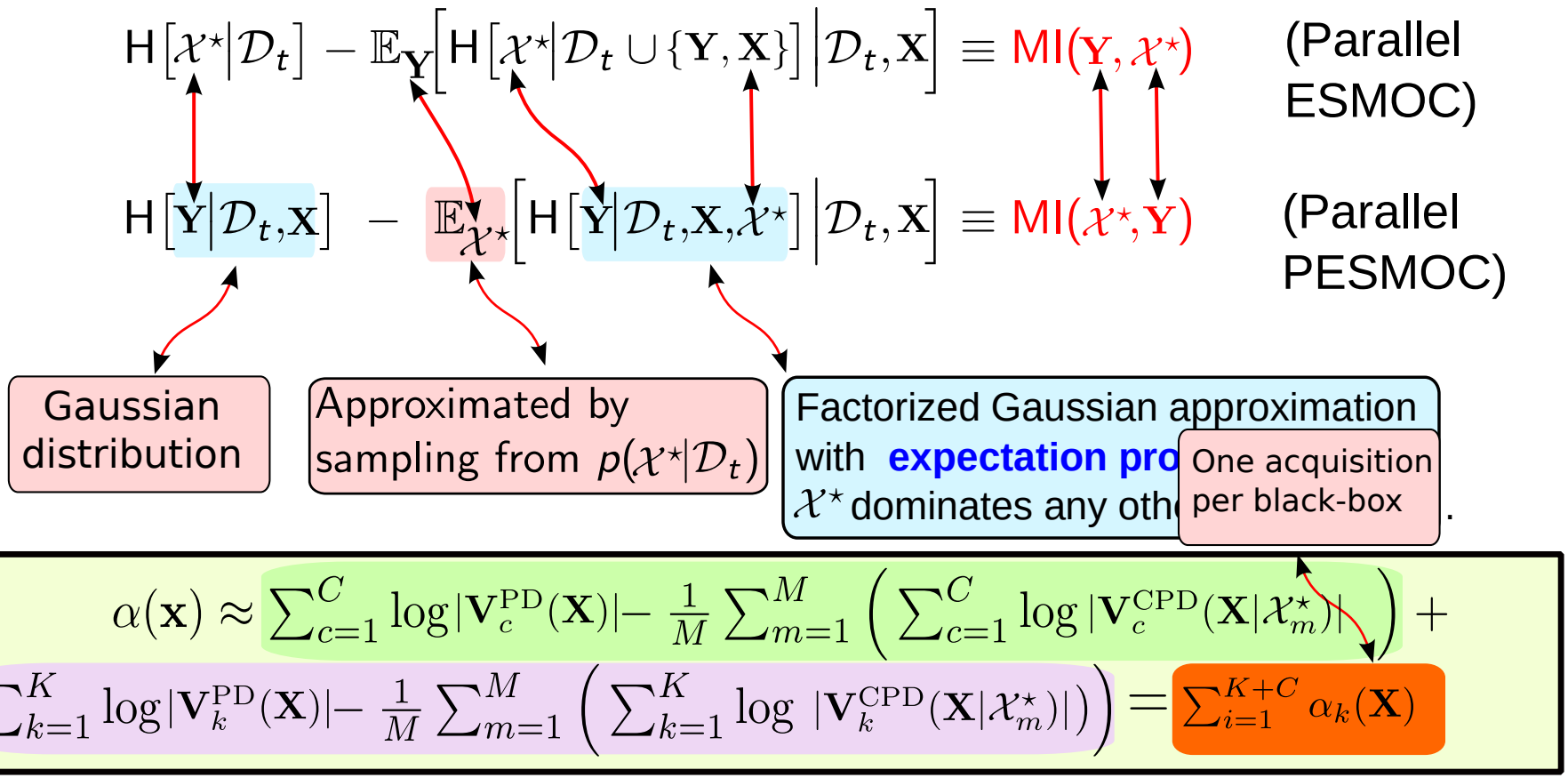
Factorized Gaussian approximation with **expectation propagation**. $\mathcal{X}^\star$ dominates any other point in $\mathcal{X}$.

$$\alpha(\mathbf{x}) \approx \sum_{c=1}^{C} \log|\mathbf{V}_c^{\mathrm{PD}}(\mathbf{X})| - \frac{1}{M}\sum_{m=1}^{M}\left(\sum_{c=1}^{C}\log|\mathbf{V}_c^{\mathrm{CPD}}(\mathbf{X}|\mathcal{X}_m^\star)|\right) + \sum_{k=1}^{K}\log|\mathbf{V}_k^{\mathrm{PD}}(\mathbf{X})| - \frac{1}{M}\sum_{m=1}^{M}\left(\sum_{k=1}^{K}\log|\mathbf{V}_k^{\mathrm{CPD}}(\mathbf{X}|\mathcal{X}_m^\star)|\right) = \sum_{i=1}^{K+C}\alpha_k(\mathbf{X})$$

(Minka, 2001)

Parallel Predictive Entropy Search

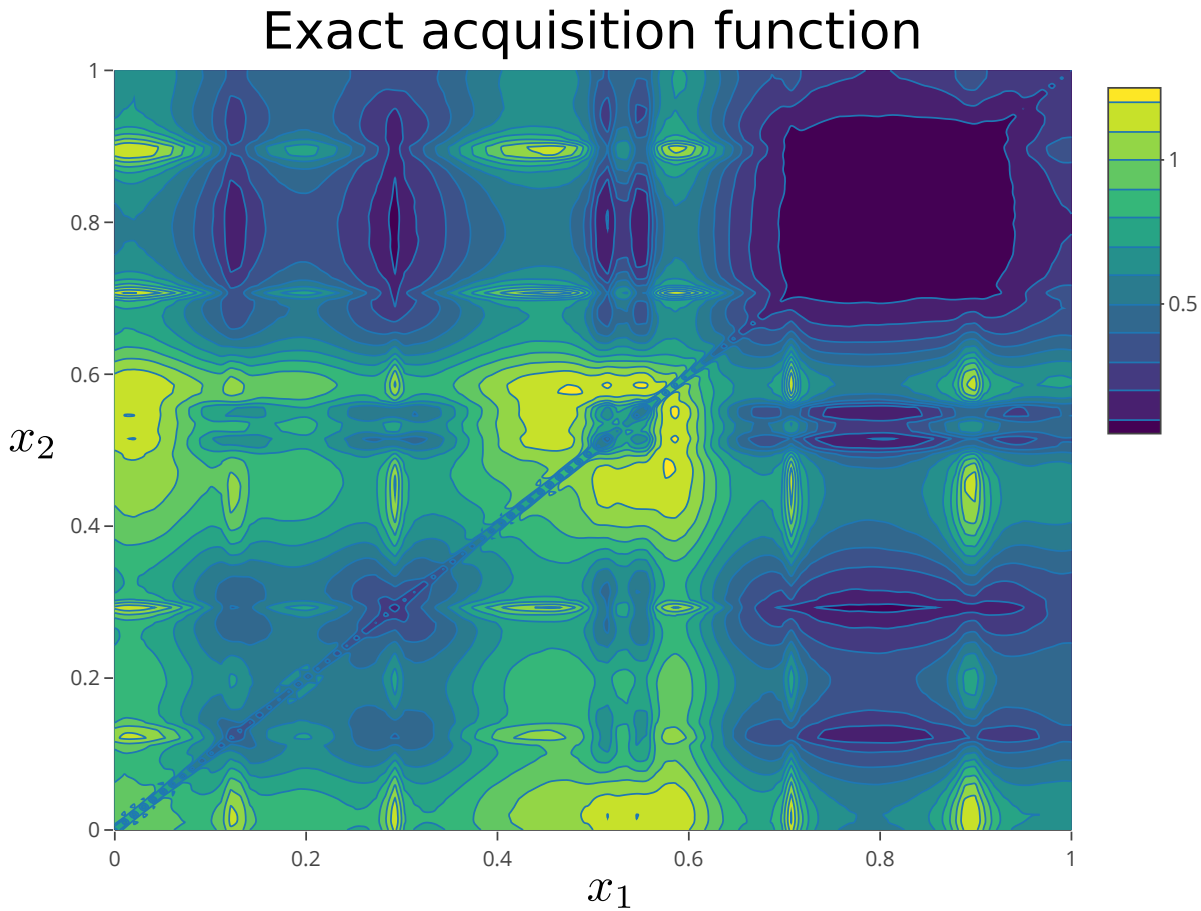
Choose a set of Q points $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ to minimize the entropy of $\mathcal{X}^\star$.



(Minka, 2001)

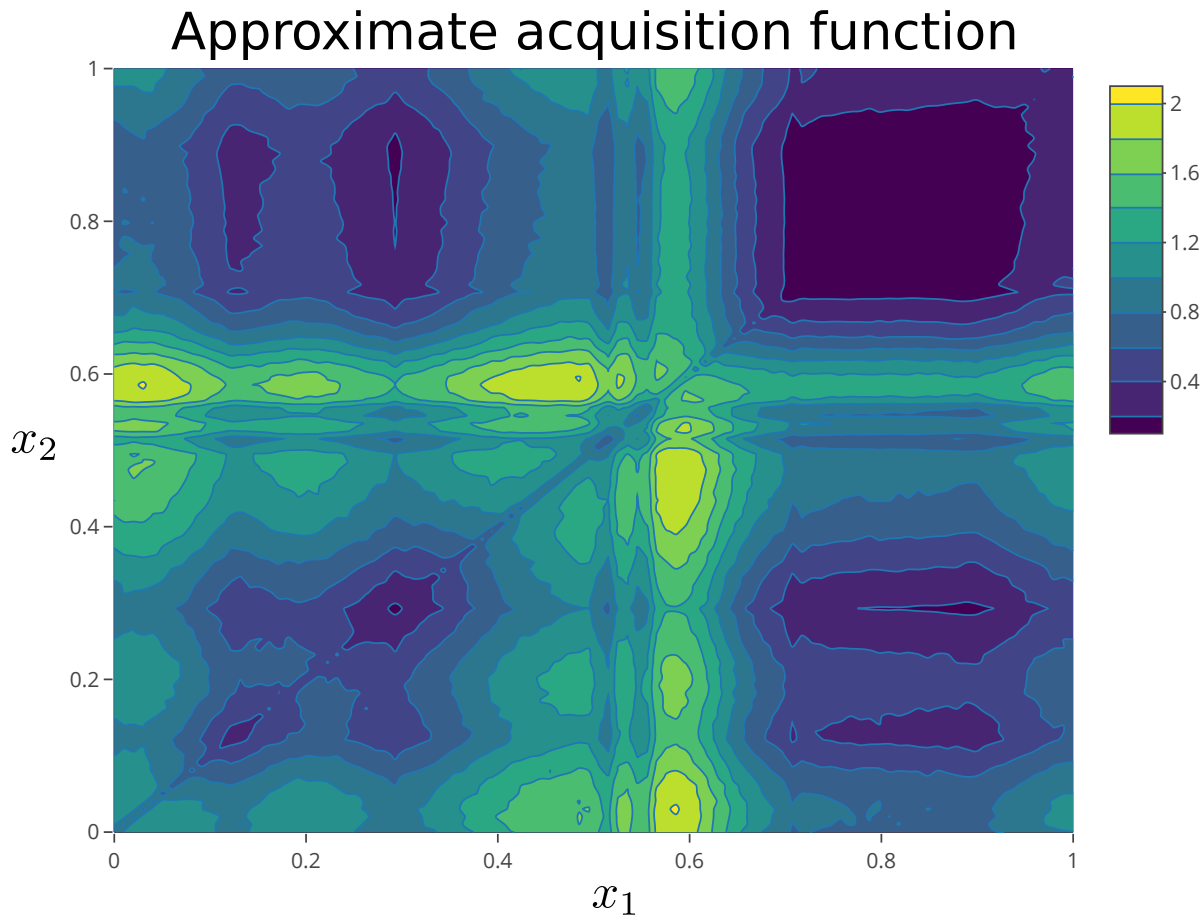
Comparison of Exact and Approximate Acquisition

1 dimensional problem, Batch size $= 2$:



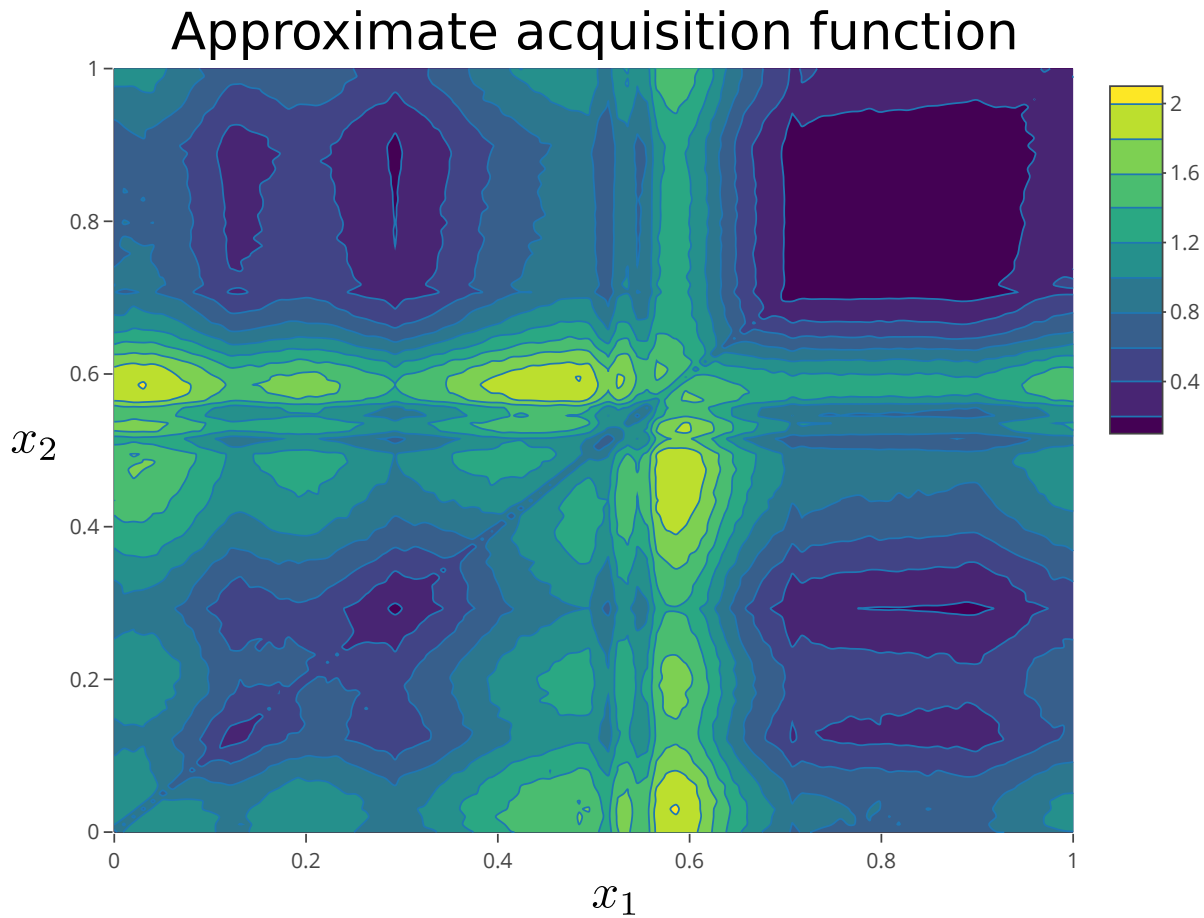
Comparison of Exact and Approximate Acquisition

1 dimensional problem, Batch size $= 2$:



Comparison of Exact and Approximate Acquisition

1 dimensional problem, Batch size $= 2$:



The acquisition is symmetric and the approximation is large where the exact acquisition is large, as expected!

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.
    - Repeats the optimization of the acquisition B times.

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.
  - Repeats the optimization of the acquisition B times.
  - Models are updated. Pending points assigned the predictive mean.

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.
  - Repeats the optimization of the acquisition B times.
  - Models are updated. Pending points assigned the predictive mean.

Expected to be very expensive for large batch sizes B!

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.
  - Repeats the optimization of the acquisition B times.
  - Models are updated. Pending points assigned the predictive mean.

  Expected to be very expensive for large batch sizes B!

- Expected Hyper-volume Improvement Strategies:

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.
  - Repeats the optimization of the acquisition B times.
  - Models are updated. Pending points assigned the predictive mean.

  Expected to be very expensive for large batch sizes B!

- Expected Hyper-volume Improvement Strategies:
  - Two versions: qEHVI (noiseless evals.) and qNEHVI (niosy evals.).

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.
  - Repeats the optimization of the acquisition B times.
  - Models are updated. Pending points assigned the predictive mean.

  Expected to be very expensive for large batch sizes B!

- Expected Hyper-volume Improvement Strategies:
  - Two versions: qEHVI (noiseless evals.) and qNEHVI (niosy evals.).
  - Constraints incorporated by multiplying by the feasibility prob.

Related Methods

- Parallel Sequential: Transforms any sequential BO method into a batch method.
  - Repeats the optimization of the acquisition B times.
  - Models are updated. Pending points assigned the predictive mean.

  Expected to be very expensive for large batch sizes B!

- Expected Hyper-volume Improvement Strategies:
  - Two versions: qEHVI (noiseless evals.) and qNEHVI (niosy evals.).
  - Constraints incorporated by multiplying by the feasibility prob.
  - Expectations approximated by Monte Carlo.

Related Methods

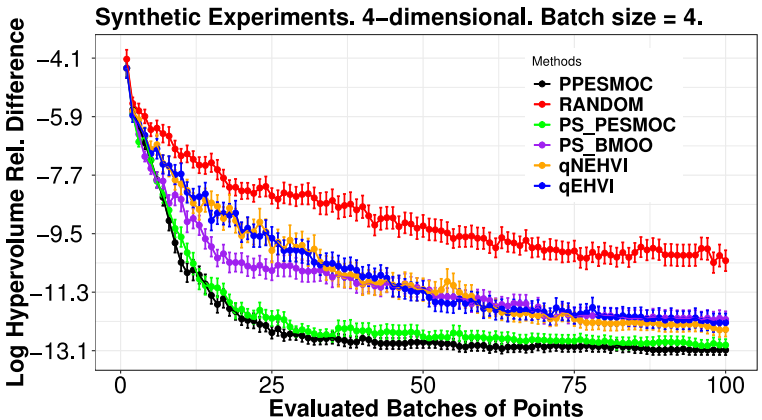
- Parallel Sequential: Transforms any sequential BO method into a batch method.
  - Repeats the optimization of the acquisition B times.
  - Models are updated. Pending points assigned the predictive mean.

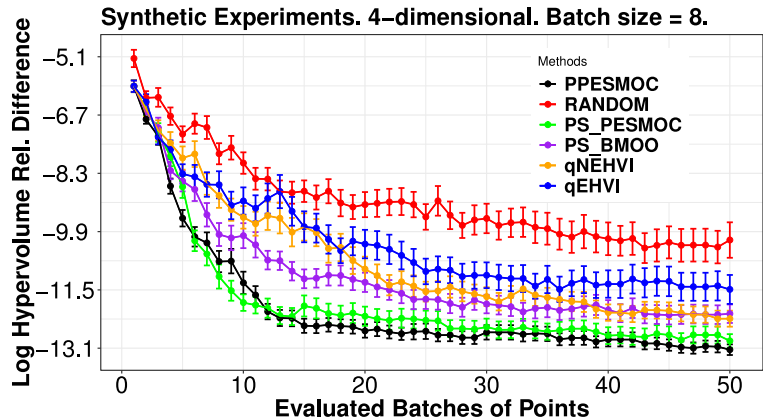
Expected to be very expensive for large batch sizes B!

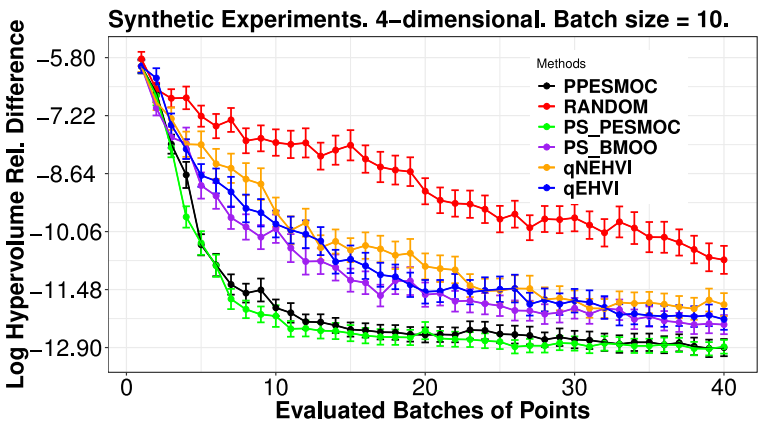
- Expected Hyper-volume Improvement Strategies:
  - Two versions: qEHVI (noiseless evals.) and qNEHVI (niosy evals.).
  - Constraints incorporated by multiplying by the feasibility prob.
  - Expectations approximated by Monte Carlo.

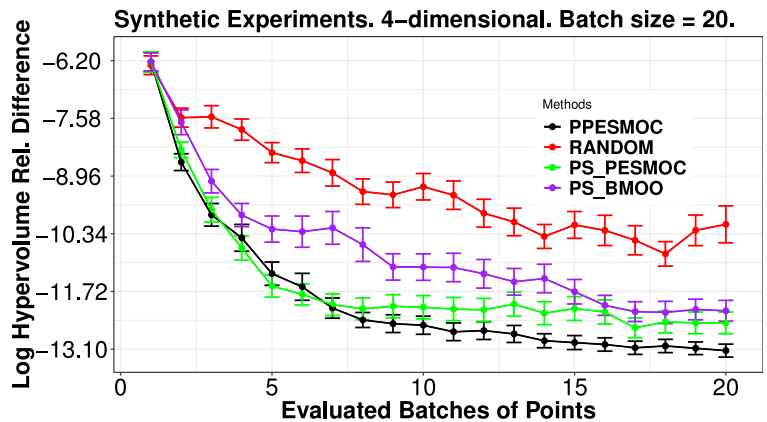
MC approximation is zero after a few evaluations and they have high cost w.r.t. B (even exponential)!

Synthetic Experimnets

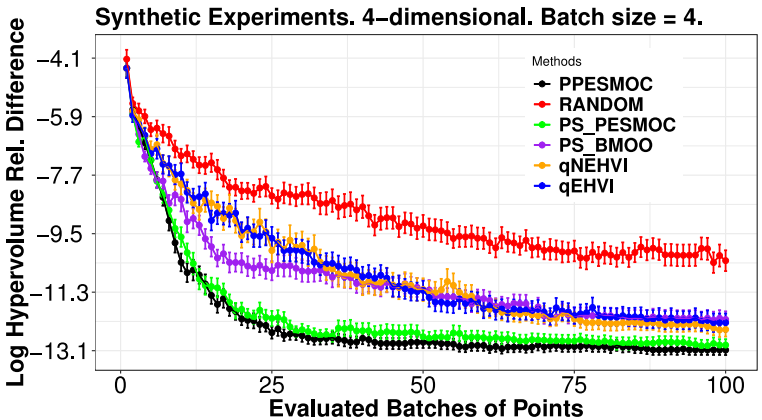


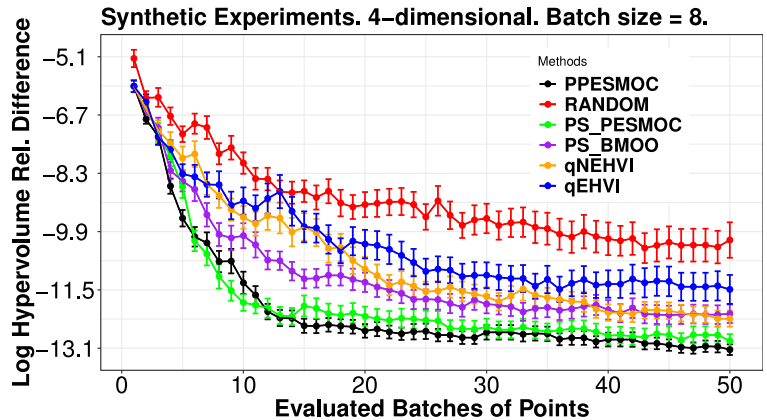


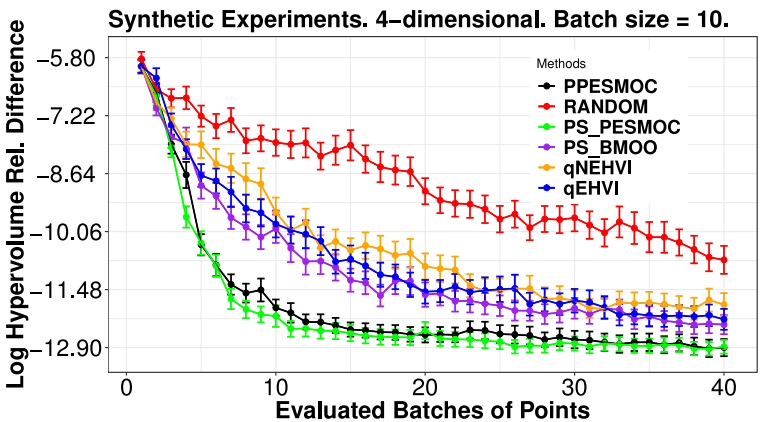


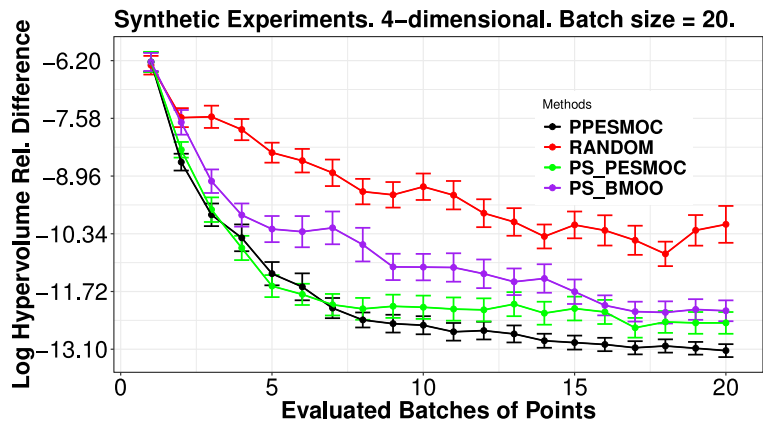


Synthetic Experimnets









PPESMOC performs better than or similar to the other strategies!

Time to Choose the next Batch

Table 1
Mean of the time in seconds to choose the next batch of points by PPESMOC and the parallel sequential approaches. For $B = 50$, underlined results are significantly different with respect to PPESMOC results according to the Wilcoxon test at $\alpha = 0.05$.

Method	$B = 4$	$B = 8$	$B = 10$	$B = 20$	$B = 50$
PPESMOC	696.0 ± 26.9	912.74 ± 26.3	957.3 ± 25.7	1045.7 ± 30.53	**1269.35 26.62**
PS_PESMOC	191.5 ± 7.0	$\mathbf{347.2 \pm 6.0}$	$\mathbf{405.49 \pm 5.8}$	$\mathbf{801.05 \pm 27.8}$	1957.72 34.1
PS_BMOO	379.4 ± 13.1	551.1 ± 21.7	593.86 ± 18.0	897.4 ± 29.6	1870.42 42.77
qEHVI	$\mathbf{65.2 \pm 1.8}$	417.9 ± 21.9	1174.9 ± 54.3		
qNEHVI	89.5 ± 2.3	401.4 ± 23.9	1169.4 ± 56.1		

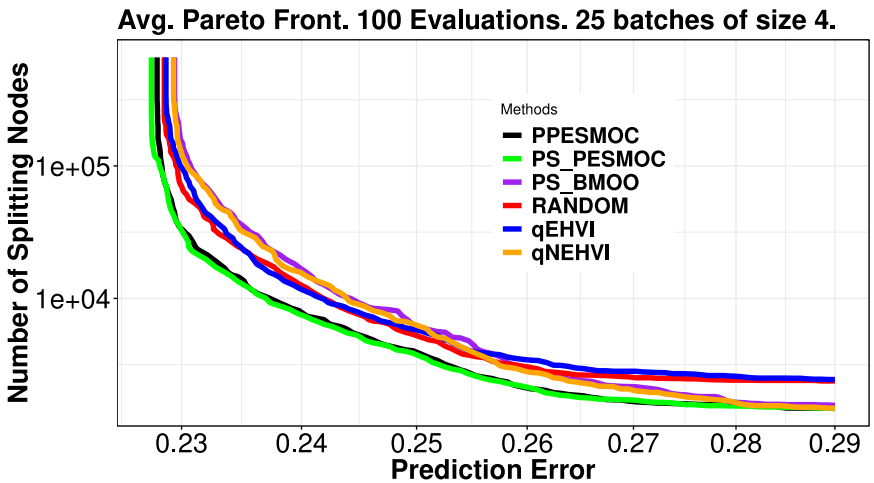
Time to Choose the next Batch

Table 1
Mean of the time in seconds to choose the next batch of points by PPESMOC and the parallel sequential approaches. For $B = 50$, underlined results are significantly different with respect to PPESMOC results according to the Wilcoxon test at $\alpha = 0.05$.

Method	$B = 4$	$B = 8$	$B = 10$	$B = 20$	$B = 50$
PPESMOC	696.0 ± 26.9	912.74 ± 26.3	957.3 ± 25.7	1045.7 ± 30.53	**1269.35 26.62**
PS_PESMOC	191.5 ± 7.0	$\mathbf{347.2 \pm 6.0}$	$\mathbf{405.49 \pm 5.8}$	$\mathbf{801.05 \pm 27.8}$	1957.72 34.1
PS_BMOO	379.4 ± 13.1	551.1 ± 21.7	593.86 ± 18.0	897.4 ± 29.6	1870.42 42.77
qEHVI	$\mathbf{65.2 \pm 1.8}$	417.9 ± 21.9	1174.9 ± 54.3		
qNEHVI	89.5 ± 2.3	401.4 ± 23.9	1169.4 ± 56.1		

PPESMOC scales significantly better w.r.t. the batch size B for large values of B!

Optimal Ensemble on the German Dataset



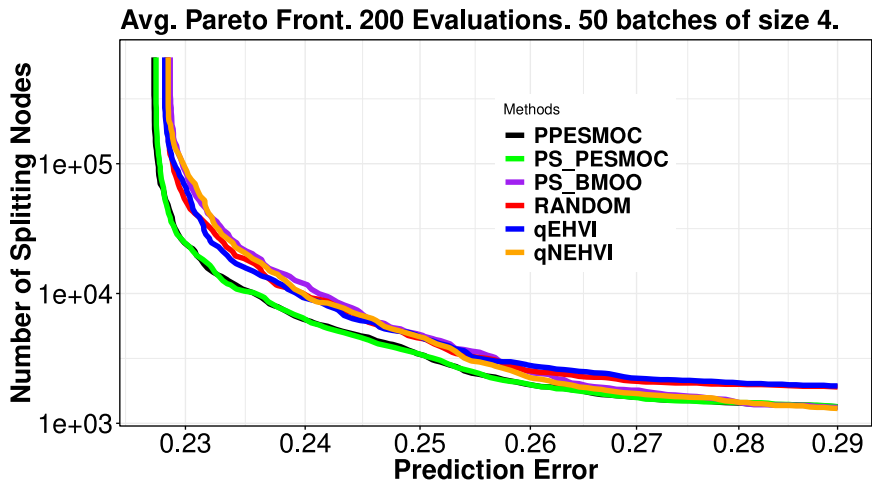
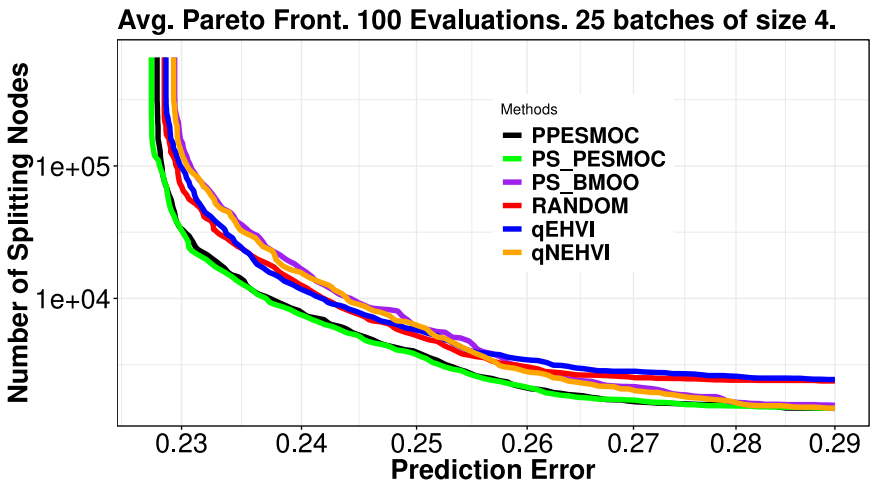


Table 2

Average hyper-volume in the task of finding an optimal ensemble of trees. Underlined results are significantly different with respect to PPESMOC results according to the Wilcoxon test at $\alpha = 0.05$.

# Eval.	PPESMOC	PS_PESMOC	PS_BMOO	P_RANDOM	qNEHVI	qEHVI
100	0.325 ± 0.007	**0.327 ± 0.007**	0.295 ± 0.014	0.298 ± 0.009	0.299 ± 0.011	0.294 ± 0.013
200	0.334 ± 0.005	**0.335 ± 0.006**	0.313 ± 0.010	0.310 ± 0.007	0.3154 ± 0.008	0.309 ± 0.010

Optimal Ensemble on the German Dataset



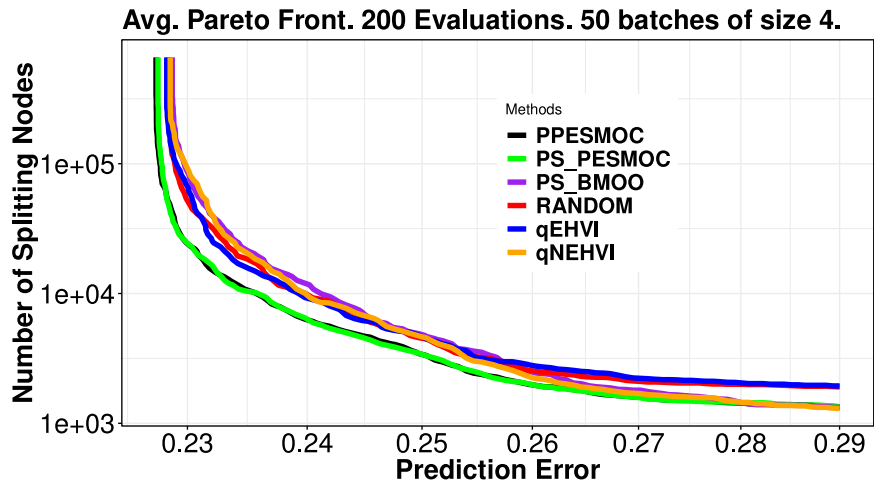


Table 2
Average hyper-volume in the task of finding an optimal ensemble of trees. Underlined results are significantly different with respect to PPESMOC results according to the Wilcoxon test at $\alpha = 0.05$.

# Eval.	PPESMOC	PS_PESMOC	PS_BMOO	P_RANDOM	qNEHVI	qEHVI
100	0.325 ± 0.007	**0.327 ± 0.007**	0.295 ± 0.014	0.298 ± 0.009	0.299 ± 0.011	0.294 ± 0.013
200	0.334 ± 0.005	**0.335 ± 0.006**	0.313 ± 0.010	0.310 ± 0.007	0.3154 ± 0.008	0.309 ± 0.010

PPESMOC performs better than or similar to the other strategies!

Optimal Neural Network on MNIST

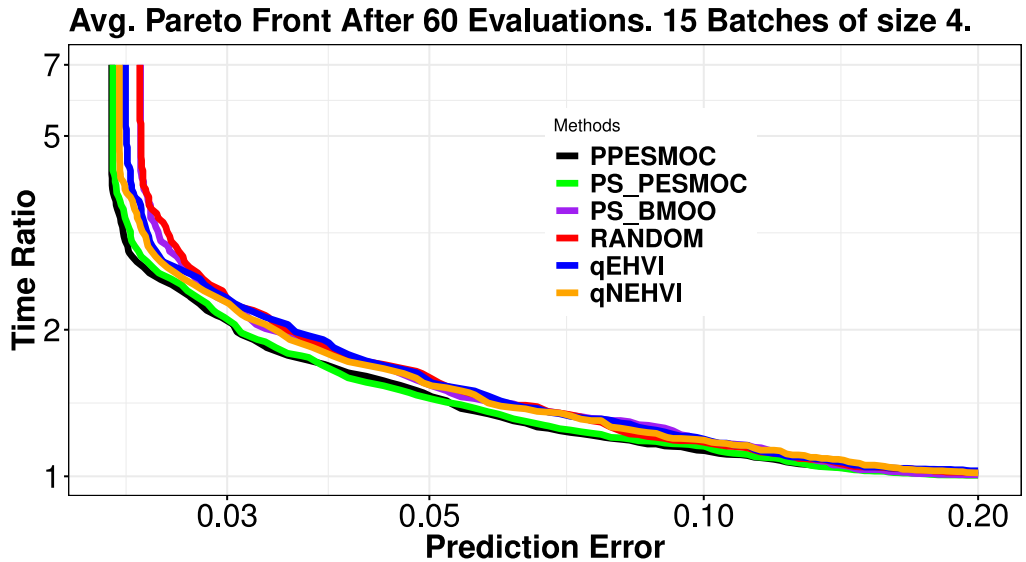


Table 3
Avg. hyper-volume of each method in the neural network experiment. Underlined results are significantly different with respect to PPESMOC results according to the Wilcoxon test at $\alpha = 0.05$.

# Eval.	PPESMOC	PS_PESMOC	PS_BMOO	P_RANDOM	qNEHVI	qEHVI
60	**1.020 ± 0.014**	1.014 ± 0.029	0.982 ± 0.095	0.993 ± 0.035	0.999 ± 0.050	0.996 ± 0.041

Optimal Neural Network on MNIST

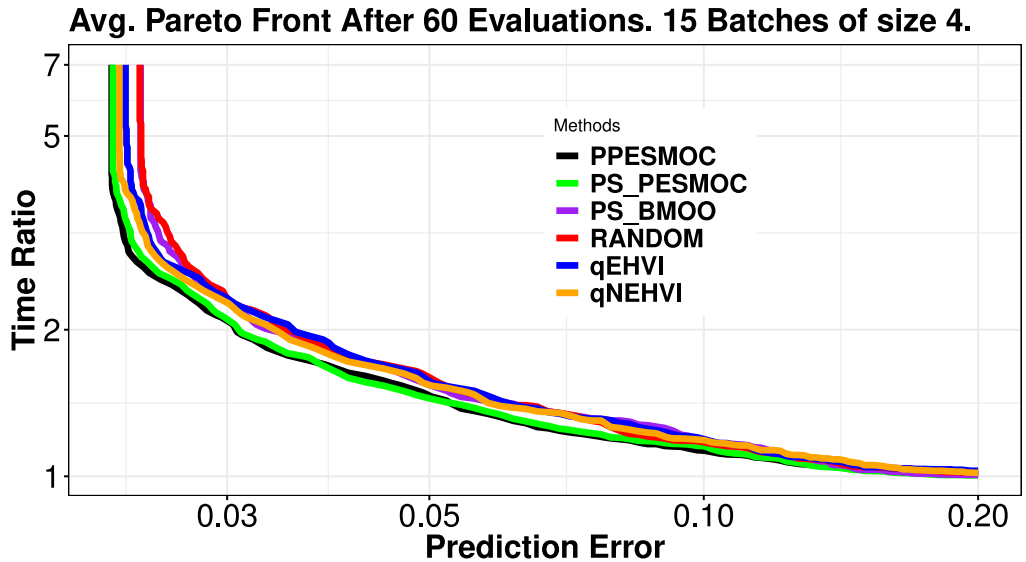


Table 3
Avg. hyper-volume of each method in the neural network experiment. Underlined results are significantly different with respect to PPESMOC results according to the Wilcoxon test at $\alpha = 0.05$.

# Eval.	PPESMOC	PS_PESMOC	PS_BMOO	P_RANDOM	qNEHVI	qEHVI
60	**1.020 ± 0.014**	1.014 ± 0.029	0.982 ± 0.095	0.993 ± 0.035	0.999 ± 0.050	0.996 ± 0.041

PPESMOC performs slithgly better than the other strategies!

Conclusions

- PPESMOC is the first entropy-based BO method for several objectives and constraints and parallel black-box evaluations.

Conclusions

- PPESMOC is the first entropy-based BO method for several objectives and constraints and parallel black-box evaluations.
- PPESMOC performs similar or better than other methods from the literature having a smaller computational cost w.r.t. the batch size.

Conclusions

- PPESMOC is the first entropy-based BO method for several objectives and constraints and parallel black-box evaluations.
- PPESMOC performs similar or better than other methods from the literature having a smaller computational cost w.r.t. the batch size.
- If the batch size B is small, Parallel Sequential methods based on PESMOC may be the better approach.

Conclusions

- PPESMOC is the first entropy-based BO method for several objectives and constraints and parallel black-box evaluations.
- PPESMOC performs similar or better than other methods from the literature having a smaller computational cost w.r.t. the batch size.
- If the batch size B is small, Parallel Sequential methods based on PESMOC may be the better approach.

Thank you for your attention!

Partially funded by the Autonomous Community of Madrid

References I

- Daulton, S., Balandat, M., & Bakshy, E. (2020). Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In Advances in neural information processing systems (pp. 9851–9864).
- Daulton, S., Balandat, M., & Bakshy, E. (2021). Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In Advances in neural information processing systems (pp. 2187–2200).
- Feliot, P., Bect, J., & Vazquez, E. (2017). A Bayesian approach to constrained single-and multi-objective optimization. Journal of Global Optimization, 67, 97–133.
- Garrido-Merchán, E., & Hernández-Lobato, D. (2019). Predictive entropy search for multi-objective Bayesian optimization with constraints. Neurocomputing, 361, 50–68.
- Snoek, J., Larochelle, H., & Adams, R. (2012). Practical Bayesian optimization of machine learning algorithms. In Advances in neural information processing systems (pp. 2951–2959).

References II

- Villemonteix, J., Vazquez, E., & Walter, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. Journal of Global Optimization, 44(509).
- Tu, B., Gandy, A., Kantas, N., & Shafei, B. (2022). Joint entropy search for multi-objective bayesian optimization. Advances in Neural Information Processing Systems, 35, 9922-9938.

Conditional Predictive Distribution I

The predictions must be compatible with $\mathcal{X}^\star$!

$$p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) = \int \underbrace{p(\mathbf{y}|\mathbf{f})}_{\text{Noise}} \underbrace{p(\mathbf{f}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)}_{\text{Noiseless pred. dist.}} \, d\mathbf{f}$$

Conditional Predictive Distribution I

The predictions must be compatible with $\mathcal{X}^\star$!

$$p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) = \int \underbrace{p(\mathbf{y}|\mathbf{f})}_{\text{Noise}} \underbrace{p(\mathbf{f}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)}_{\text{Noiseless pred. dist.}} \, d\mathbf{f}$$

Noiseless conditional predictive distribution:

$$p(\mathbf{f}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) \propto \int \underbrace{p(\mathbf{f}|\mathbf{x}, \mathcal{F})}_{\text{Black-box values at } \mathbf{x}} \times \underbrace{p(\mathcal{F}|\mathcal{D})}_{\text{Post. dist.}} \times \underbrace{p(\mathcal{X}^\star|\mathcal{F})}_{\text{Guarantees } \mathcal{X}^\star \text{ optimal.}} d\mathcal{F}$$

where $\mathcal{F}$ informally representes all potential black-box functions.

Conditional Predictive Distribution I

The predictions must be compatible with $\mathcal{X}^\star$!

$$p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) = \int \underbrace{p(\mathbf{y}|\mathbf{f})}_{\text{Noise}} \underbrace{p(\mathbf{f}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)}_{\text{Noiseless pred. dist.}} \, d\mathbf{f}$$

Noiseless conditional predictive distribution:

$$p(\mathbf{f}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) \propto \int \underbrace{p(\mathbf{f}|\mathbf{x}, \mathcal{F})}_{\text{Black-box values at } \mathbf{x}} \times \underbrace{p(\mathcal{F}|\mathcal{D})}_{\text{Post. dist.}} \times \underbrace{p(\mathcal{X}^\star|\mathcal{F})}_{\text{Guarantees } \mathcal{X}^\star \text{ optimal.}} d\mathcal{F}$$

where $\mathcal{F}$ informally representes all potential black-box functions.

- Unconditional posterior.

Conditional Predictive Distribution I

The predictions must be compatible with $\mathcal{X}^\star$!

$$p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) = \int \underbrace{p(\mathbf{y}|\mathbf{f})}_{\text{Noise}} \underbrace{p(\mathbf{f}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)}_{\text{Noiseless pred. dist.}} \, d\mathbf{f}$$

Noiseless conditional predictive distribution:

$$p(\mathbf{f}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) \propto \int \underbrace{p(\mathbf{f}|\mathbf{x}, \mathcal{F})}_{\text{Black-box values at } \mathbf{x}} \times \underbrace{p(\mathcal{F}|\mathcal{D})}_{\text{Post. dist.}} \times \underbrace{p(\mathcal{X}^\star|\mathcal{F})}_{\text{Guarantees } \mathcal{X}^\star \text{ optimal.}} d\mathcal{F}$$

where $\mathcal{F}$ informally representes all potential black-box functions.

- Unconditional posterior.
- Takes value 1 if $\mathcal{X}^\star$ is optimal given $\mathcal{F}$ and zero otherwise.

Conditional Predictive Distribution II

The factor that guarantees optimality is:

$$p(\mathcal{X}^\star|\mathcal{F}) = \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\left[\prod_{c=1}^{C} \Theta(\text{cons}_c(\mathbf{x}^\star)) \right] \left[\prod_{\mathbf{x}' \in \mathcal{X}} \Omega(\mathbf{x}', \mathbf{x}^\star) \right] \right)$$

Conditional Predictive Distribution II

The factor that guarantees optimality is:

$$p(\mathcal{X}^\star|\mathcal{F}) = \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\left[\prod_{c=1}^{C} \Theta(\text{cons}_c(\mathbf{x}^\star)) \right] \left[\prod_{\mathbf{x}' \in \mathcal{X}} \Omega(\mathbf{x}', \mathbf{x}^\star) \right] \right)$$

- Takes value 0 if $\mathbf{x}^\star$ is infeasible and zero othersie!

Conditional Predictive Distribution II

The factor that guarantees optimality is:

$$p(\mathcal{X}^\star|\mathcal{F}) = \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\left[\prod_{c=1}^{C} \Theta(\text{cons}_c(\mathbf{x}^\star)) \right] \left[\prod_{\mathbf{x}' \in \mathcal{X}} \Omega(\mathbf{x}', \mathbf{x}^\star) \right] \right)$$

- Takes value 0 if $\mathbf{x}^\star$ is infeasible and zero othersie!
- Takes value 0 if $\mathbf{x}^\star$ is dominated by feasible $\mathbf{x}'$ and zero otherwise.

Conditional Predictive Distribution II

The factor that guarantees optimality is:

$$p(\mathcal{X}^\star|\mathcal{F}) = \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \left(\left[\prod_{c=1}^{C} \Theta(\text{cons}_c(\mathbf{x}^\star)) \right] \left[\prod_{\mathbf{x}' \in \mathcal{X}} \Omega(\mathbf{x}', \mathbf{x}^\star) \right] \right)$$

- Takes value 0 if $\mathbf{x}^\star$ is infeasible and zero othersie!
- Takes value 0 if $\mathbf{x}^\star$ is dominated by feasible $\mathbf{x}'$ and zero otherwise.

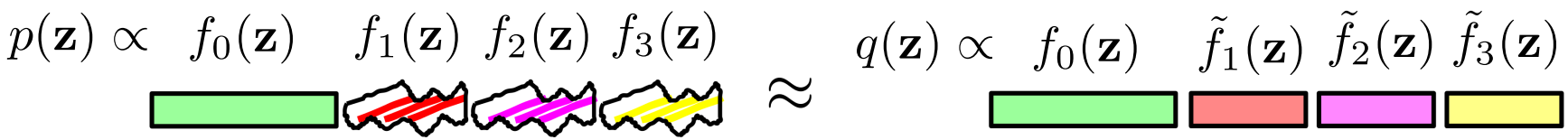
The factors are all step functions! The set $\mathcal{X}$ is approximated using the evaluations!

Expectation Propagation

Approximates $\boxed{p(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} f_j(\boldsymbol{z})}$ with $\boxed{q(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} \tilde{f}_j(\boldsymbol{z})}$

Expectation Propagation

Approximates $p(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} f_j(\boldsymbol{z})$ with $q(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} \tilde{f}_j(\boldsymbol{z})$



Expectation Propagation

Approximates $\boxed{p(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} f_j(\boldsymbol{z})}$ with $\boxed{q(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} \tilde{f}_j(\boldsymbol{z})}$

$p(\mathbf{z}) \propto f_0(\mathbf{z}) \; f_1(\mathbf{z}) \; f_2(\mathbf{z}) \; f_3(\mathbf{z}) \quad \approx \quad q(\mathbf{z}) \propto f_0(\mathbf{z}) \; \tilde{f}_1(\mathbf{z}) \; \tilde{f}_2(\mathbf{z}) \; \tilde{f}_3(\mathbf{z})$

The $\tilde{f}_j$ are tuned by minimizing the KL-divergence

$$\mathrm{KL}[\hat{p}_j || q] \quad \text{for } j = 1, \ldots, N, \quad \text{where} \quad \begin{array}{rcl} \hat{p}_j(\boldsymbol{z}) & \propto & f_j(\boldsymbol{z}) \prod_{i \neq j} \tilde{f}_i(\boldsymbol{z}) \\ q(\boldsymbol{z}) & \propto & \tilde{f}_j(\boldsymbol{z}) \prod_{i \neq j} \tilde{f}_i(\boldsymbol{z}) \end{array}.$$

Expectation Propagation

Approximates $\boxed{p(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} f_j(\boldsymbol{z})}$ with $\boxed{q(\boldsymbol{z}) \propto f_0(\boldsymbol{z}) \prod_{j=1}^{N} \tilde{f}_j(\boldsymbol{z})}$

$p(\mathbf{z}) \propto f_0(\mathbf{z}) \; f_1(\mathbf{z}) \; f_2(\mathbf{z}) \; f_3(\mathbf{z}) \approx q(\mathbf{z}) \propto f_0(\mathbf{z}) \; \tilde{f}_1(\mathbf{z}) \; \tilde{f}_2(\mathbf{z}) \; \tilde{f}_3(\mathbf{z})$

The $\tilde{f}_j$ are tuned by minimizing the KL-divergence

$$\mathrm{KL}[\hat{p}_j || q] \quad \text{for } j = 1, \ldots, N, \quad \text{where} \quad \begin{array}{rcl} \hat{p}_j(\boldsymbol{z}) & \propto & f_j(\boldsymbol{z}) \prod_{i \neq j} \tilde{f}_i(\boldsymbol{z}) \\ q(\boldsymbol{z}) & \propto & \tilde{f}_j(\boldsymbol{z}) \prod_{i \neq j} \tilde{f}_i(\boldsymbol{z}) \end{array}.$$

The latent variables z are in our case the objectives and the constraints values at each $x^\star$ and each x'!

Optimal Ensemble of Decision Trees

- Dataset: German Credit

Optimal Ensemble of Decision Trees

- Dataset: German Credit
  - Number of instances: 1000

Optimal Ensemble of Decision Trees

- Dataset: German Credit
  - Number of instances: 1000
  - Number of features: 20

Optimal Ensemble of Decision Trees

- Dataset: German Credit
  - Number of instances: 1000
  - Number of features: 20
- Ensemble Parameters:

Optimal Ensemble of Decision Trees

- Dataset: German Credit
  - Number of instances: 1000
  - Number of features: 20
- Ensemble Parameters:
  - Ensemble size, random chosen attributes considered at each split, minimum number of samples required to split a node, sub-sampling probability, fraction of labels changed.

Optimal Ensemble of Decision Trees

- Dataset: German Credit
  - Number of instances: 1000
  - Number of features: 20
- Ensemble Parameters:
  - Ensemble size, random chosen attributes considered at each split, minimum number of samples required to split a node, sub-sampling probability, fraction of labels changed.
- Objectives: Ensemble size in log-number of nodes and prediction error (10-fold-cv).

Optimal Ensemble of Decision Trees

- Dataset: German Credit
  - Number of instances: 1000
  - Number of features: 20
- Ensemble Parameters:
  - Ensemble size, random chosen attributes considered at each split, minimum number of samples required to split a node, sub-sampling probability, fraction of labels changed.
- Objectives: Ensemble size in log-number of nodes and prediction error (10-fold-cv).
- Constraints: time for predictions sped-up at least 25% when using a dynamic pruning technique.

Optimal Neural Network on MNIST

- Dataset: MNIST

Optimal Neural Network on MNIST

- Dataset: MNIST
  - Number of instances: 60,000

Optimal Neural Network on MNIST

- Dataset: MNIST
  - Number of instances: 60,000
  - Number of features: 28x28=784

Optimal Neural Network on MNIST

- Dataset: MNIST
  - Number of instances: 60,000
  - Number of features: 28x28=784
- Ensemble Parameters:

Optimal Neural Network on MNIST

- Dataset: MNIST
  - Number of instances: 60,000
  - Number of features: 28x28=784
- Ensemble Parameters:
  - Hidden layers, neurons per layer, learning rate, dropout rate, ℓ_1 penalty, ℓ_2 penalty, memory partition, loop unrolling.

Optimal Neural Network on MNIST

- Dataset: MNIST
  - Number of instances: 60,000
  - Number of features: 28x28=784
- Ensemble Parameters:
  - Hidden layers, neurons per layer, learning rate, dropout rate, ℓ_1 penalty, ℓ_2 penalty, memory partition, loop unrolling.
- Objectives: network error and prediction time (validation set).

Optimal Neural Network on MNIST

- Dataset: MNIST
  - Number of instances: 60,000
  - Number of features: 28x28=784
- Ensemble Parameters:
  - Hidden layers, neurons per layer, learning rate, dropout rate, ℓ_1 penalty, ℓ_2 penalty, memory partition, loop unrolling.
- Objectives: network error and prediction time (validation set).
- Constraints: chip area below threshold.

Potential Extensions / Improvements

- Incorporate information about the Pareto front $\mathcal{Y}^\star$ (JES):

Potential Extensions / Improvements

- Incorporate information about the Pareto front $\mathcal{Y}^\star$ (JES):
  - Conditioning to $\mathcal{Y}^\star$, which can be done simply by updating each GP.

Potential Extensions / Improvements

- Incorporate information about the Pareto front $\mathcal{Y}^\star$ (JES):
  - Conditioning to $\mathcal{Y}^\star$, which can be done simply by updating each GP.
  - Measuring $I(\{\mathcal{X}^\star, \mathcal{Y}^\star\}; \mathbf{Y})$ is expected to improve results!

Potential Extensions / Improvements

- Incorporate information about the Pareto front $\mathcal{Y}^\star$ (JES):
  - Conditioning to $\mathcal{Y}^\star$, which can be done simply by updating each GP.
  - Measuring $I(\{\mathcal{X}^\star, \mathcal{Y}^\star\}; \mathbf{Y})$ is expected to improve results!
- Use decoupled information for evaluation:

Potential Extensions / Improvements

- Incorporate information about the Pareto front $\mathcal{Y}^\star$ (JES):
  - Conditioning to $\mathcal{Y}^\star$, which can be done simply by updating each GP.
  - Measuring $I(\{\mathcal{X}^\star, \mathcal{Y}^\star\}; \mathbf{Y})$ is expected to improve results!
- Use decoupled information for evaluation:
  - Easily identifies on which black-box to evaluate each batch.

Potential Extensions / Improvements

- Incorporate information about the Pareto front $\mathcal{Y}^\star$ (JES):
  - Conditioning to $\mathcal{Y}^\star$, which can be done simply by updating each GP.
  - Measuring $I(\{\mathcal{X}^\star, \mathcal{Y}^\star\}; \mathbf{Y})$ is expected to improve results!
- Use decoupled information for evaluation:
  - Easily identifies on which black-box to evaluate each batch.
  - Requires optimizing one acquisition per black-box.

Potential Extensions / Improvements

- Incorporate information about the Pareto front $\mathcal{Y}^\star$ (JES):
  - Conditioning to $\mathcal{Y}^\star$, which can be done simply by updating each GP.
  - Measuring $I(\{\mathcal{X}^\star, \mathcal{Y}^\star\}; \mathbf{Y})$ is expected to improve results!
- Use decoupled information for evaluation:
  - Easily identifies on which black-box to evaluate each batch.
  - Requires optimizing one acquisition per black-box.
  - Expected to give better results if more informative black-boxes.