# AI Research and Applications Forum: Bayesian Optimization

Bayesian Optimization

Johanna Menn, M.Sc.
Institute for Data Science in Mechanical Engineering

Center for Artificial Intelligence

RWTH AACHEN UNIVERSITY

INTERNET OF PRODUCTION

RWTH AACHEN UNIVERSITY

slido

Join at **slido.com**
**#AiForum**
or visit
https://app.sli.do/event/ddHceNA9PzG4YgyVSnJ21F



Institute for Data Science in Mechanical Engineering

RWTH AACHEN UNIVERSITY

# Agenda

16:30 - 16:45 General introduction to the event series
16:45 - 17:30 Tutorial on Bayesian optimization
17:30 - 19:30 Poster session + networking with food and drinks

# AI Research & Applications Forum: Providing Space for Exchange about AI

**What?**

AI gatherings on AI methods including a tutorial and poster sessions

Three events

**Target Audience:**

AI researchers and practioners at RWTH *and beyond*

**Key objectives:**

Connect researchers from theory and application.
Present AI methods and possible applications.

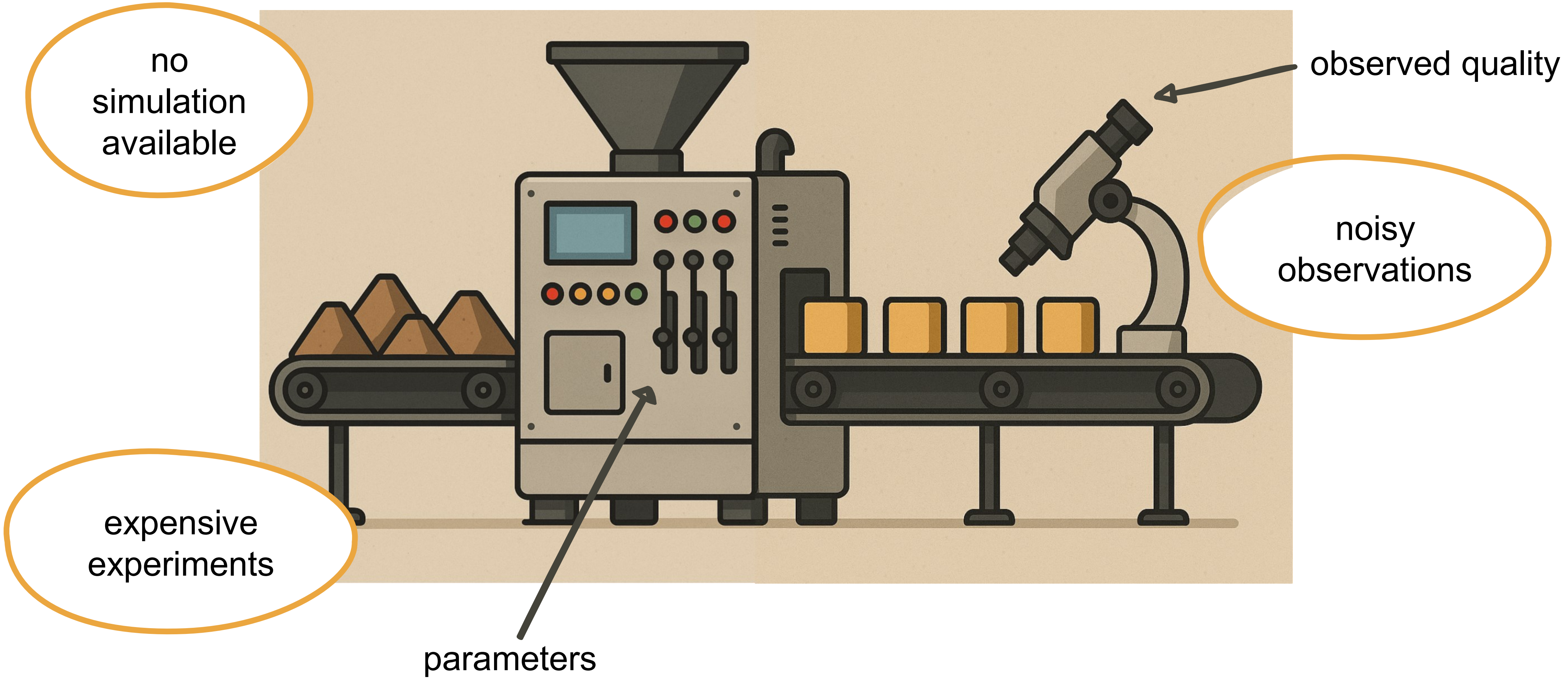
**IOP:**
Platform to show AI related work and share with lager AI community

**AI-Center:**
Connect to practioners at RWTH and find new applications

# Tutorial on Bayesian Optimization

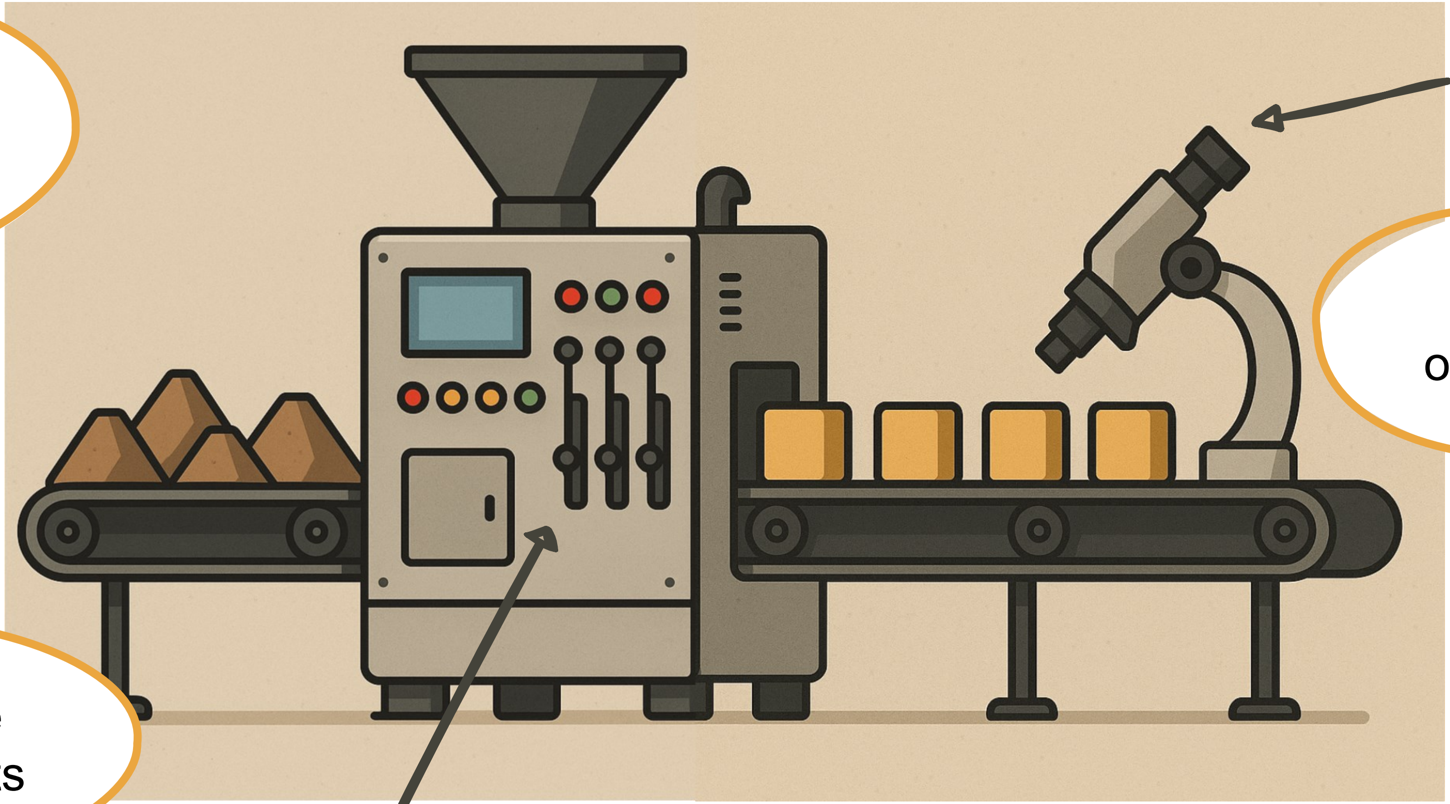
Questions in the end please ☺

# Motivation




**Goal:** Find the parameter setting that leads to the best product quality with a small number of experiments

# Problem formulation



no simulation available

observed quality
$y = J(\theta^*) + \epsilon$

noisy observations

expensive experiments

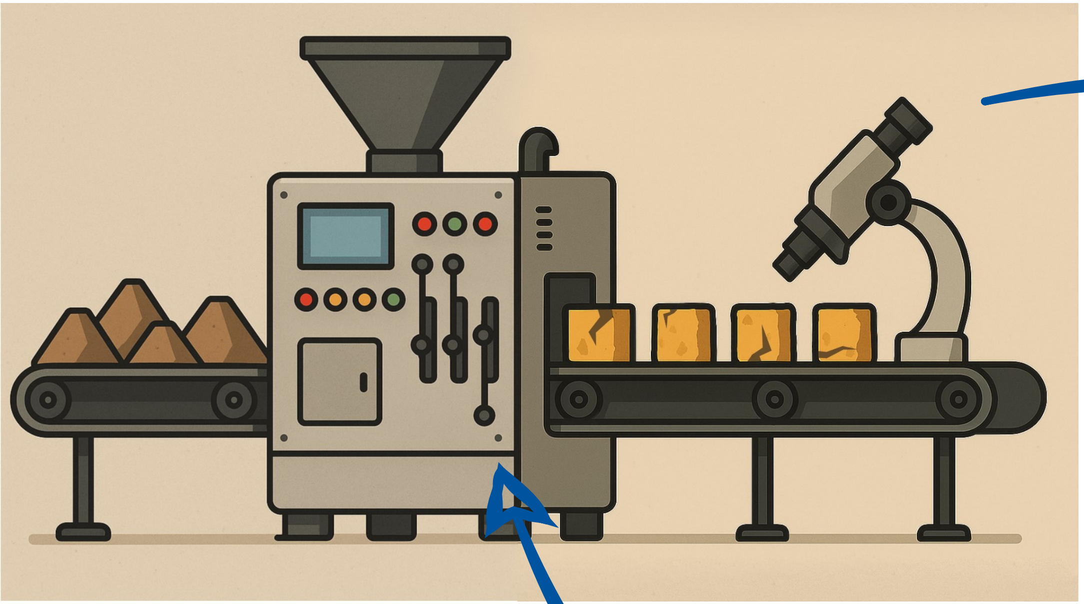
parameters $\theta^*$

**Goal:** Find the parameter setting that leads to the best product quality with a small number of experiments
***Formally:*** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathbb{E}[J(\theta)]$

# How does BO work?

## We query experiments iteratively

**Goal:** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}}\ \mathbb{E}[J(\theta)]$



observed quality
$y_i = J(\theta_i) + \epsilon_i$

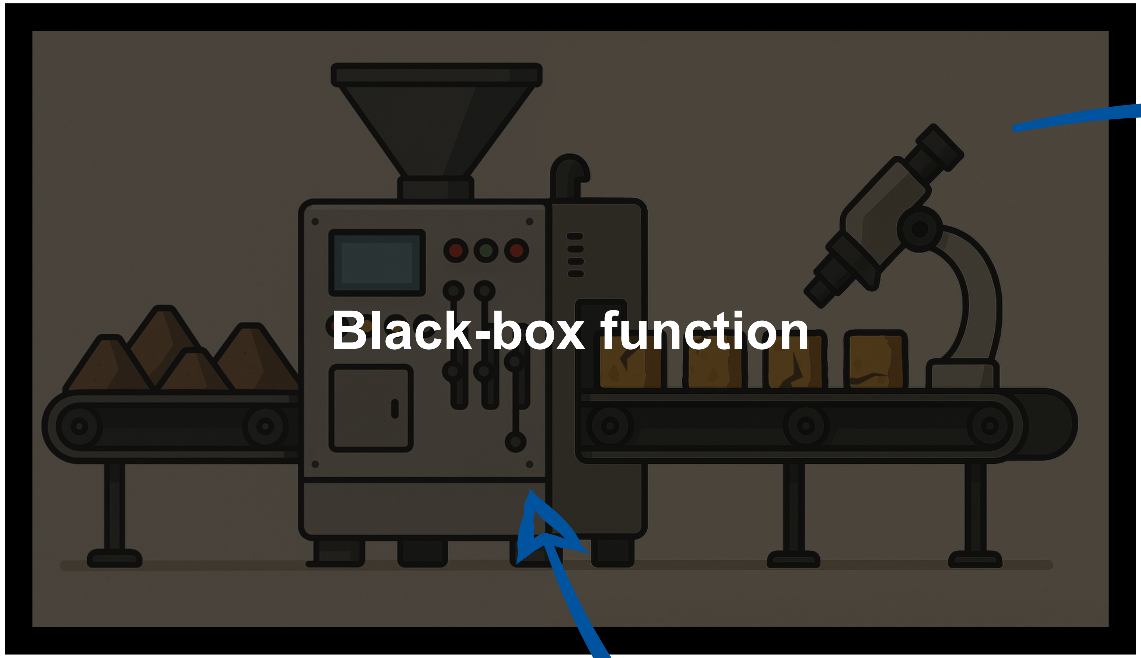
**Bayesian Optimization**

parameters $\theta_i$

# How does BO work?

## View experiment as a black-box function

**Goal:** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathbb{E}[J(\theta)]$



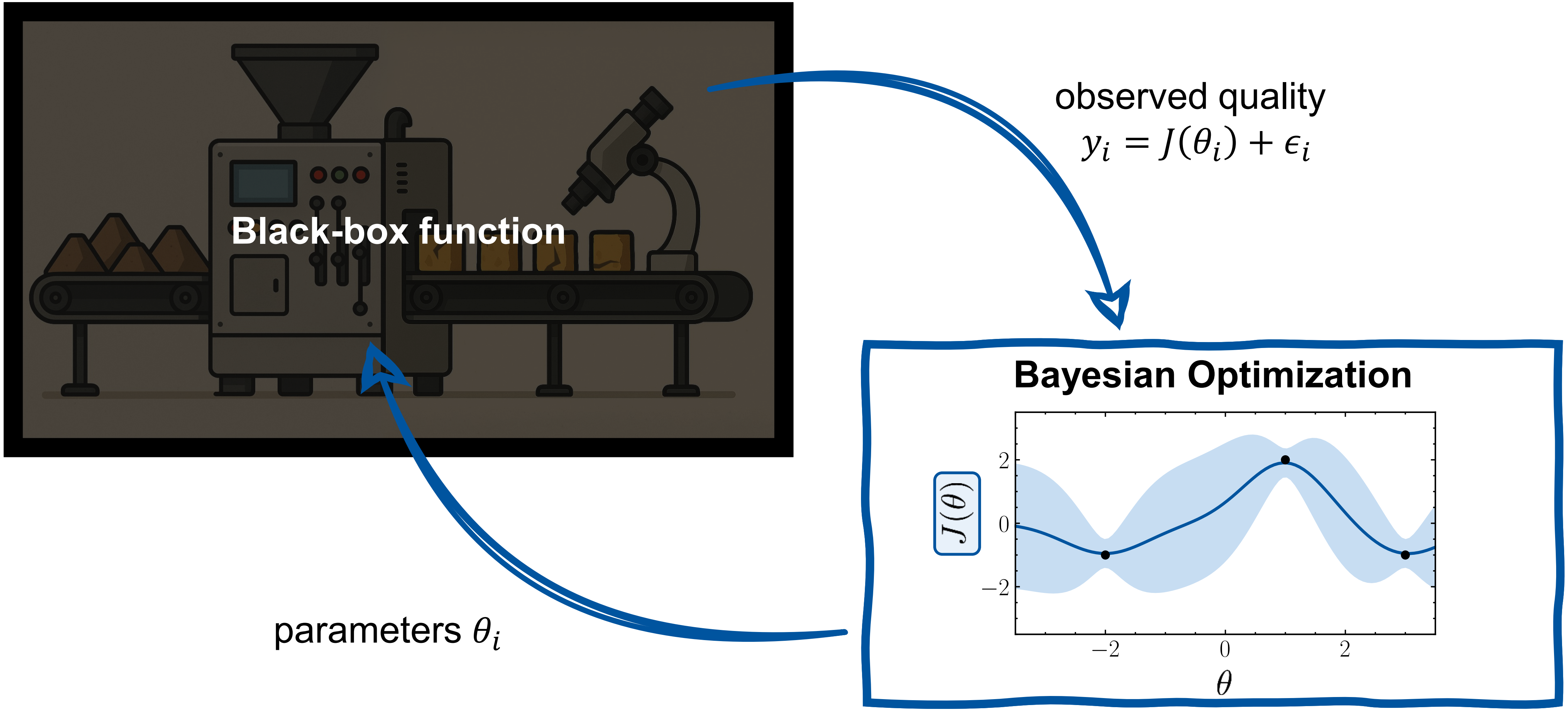

observed quality
$y_i = J(\theta_i) + \epsilon_i$

**Bayesian Optimization**

parameters $\theta_i$

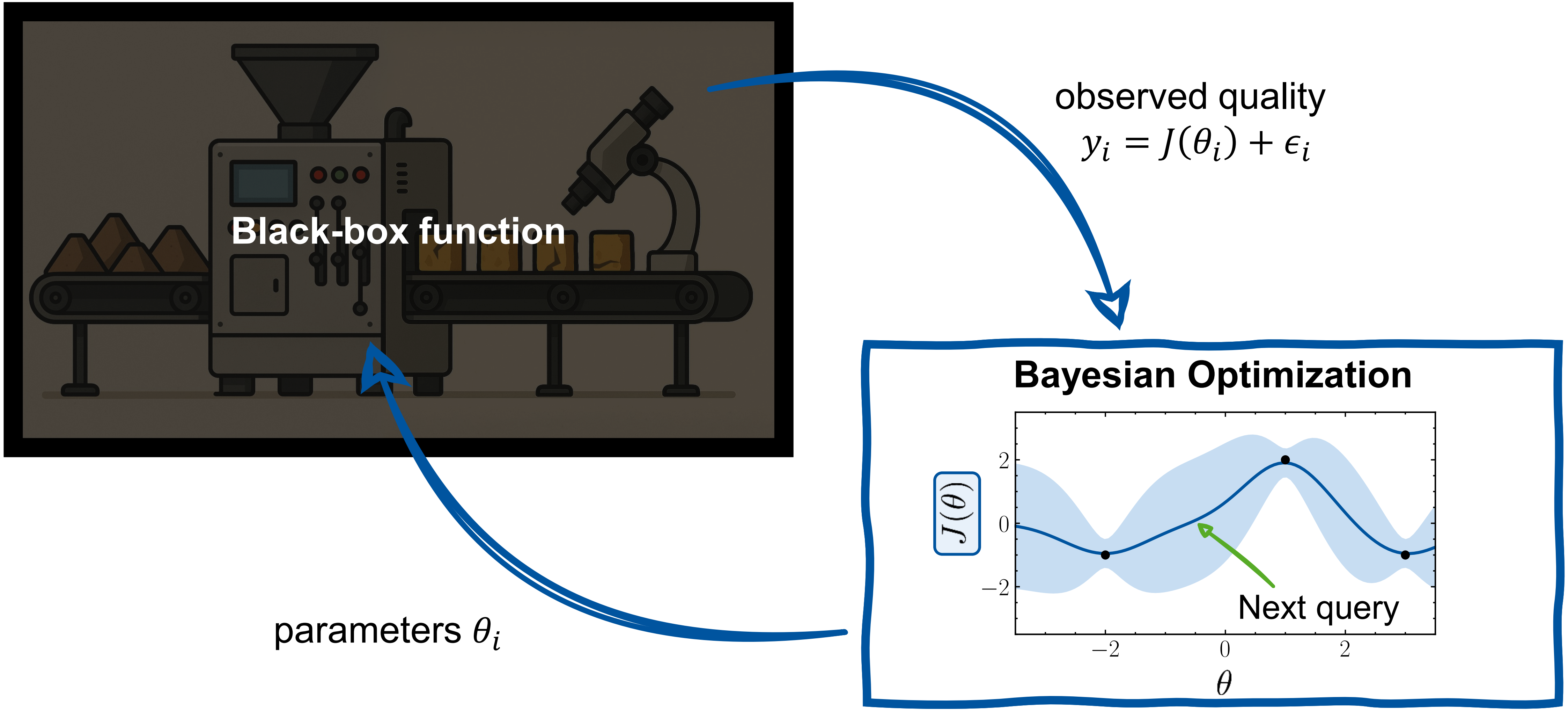
# How does BO work?

**Model the black-box function over the parameter space with a probabilistic surrogate model**



## How does BO work?

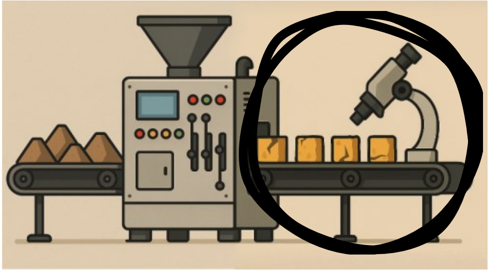
### Use the model to suggest next parameters to query (acquisition function)



# Main ingredients of Bayesian optimization

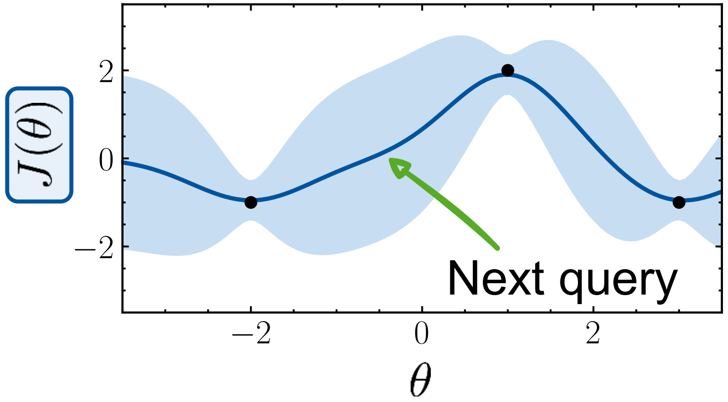
$\xrightarrow{\theta}$ **Black-box function** $\xrightarrow{y = \boxed{J(\theta)} + \epsilon}$

**Goal:** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}} \, \mathbb{E}[J(\theta)]$



**1) Observations** / evaluations / experiments

- Noisy evaluation as $J_i = J(\theta_i) + \epsilon_i$ , $\epsilon_i \sim \mathcal{N}(0, \sigma_{\mathrm{n}}^2)$, iid
- **Data set** from observations: store all relevant information
  $\mathcal{D}_i = \{(\theta_1, J_1), \dots, (\theta_i, J_i)\}$




**2) Probabilistic ML model**

- Model belief about the unknown function

**3) Acquisition function**

- Determine the next evaluation $\theta_{i+1} = \underset{\theta \in \Theta}{\operatorname{argmax}} \, \alpha(\theta \mid \mathcal{D}_i)$
- Expresses trade-off between exploration (test new solutions) and exploitation (use what is known to be “best”)



[1] Garnett, Roman, “Bayesian Optimization”, Cambridge University Press, 2023

01_bo_tutorial.ipynb

# Main ingredients of Bayesian optimization

$\xrightarrow{\theta}$ **Black-box function** $\xrightarrow{y = \boxed{J(\theta)} + \epsilon}$

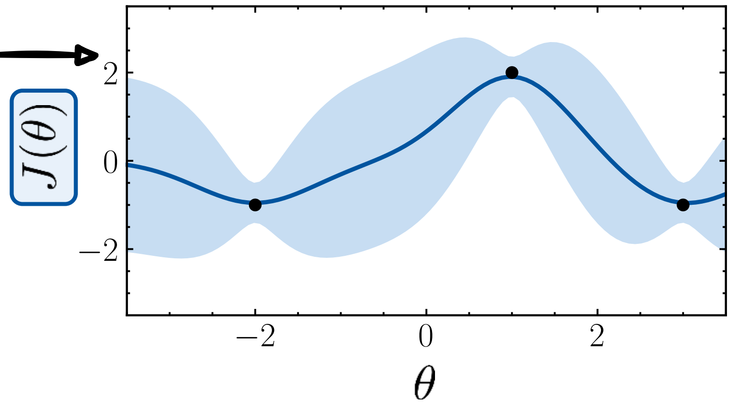
**Goal:** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathbb{E}[J(\theta)]$

**1) Observations** / evaluations / experiments:
- Noisy evaluation as $J_i = J(\theta_i) + \epsilon_i\,,\ \ \epsilon_i \sim \mathcal{N}(0, \sigma_{\mathrm{n}}^2)$, iid
- **Data set** from observations: store all relevant information
  $\mathcal{D}_i = \{(\theta_1, J_1), \ldots, (\theta_i, J_i)\}$

usually a Gaussian process

**2) Probabilistic ML model**
- Model belief about the unknown function

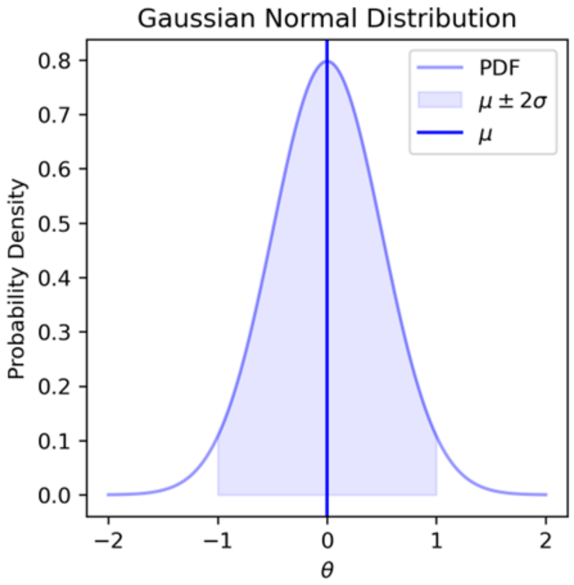
$$\boxed{J(\theta)} \sim \mathcal{GP}(m(\theta), k(\theta, \theta'))$$

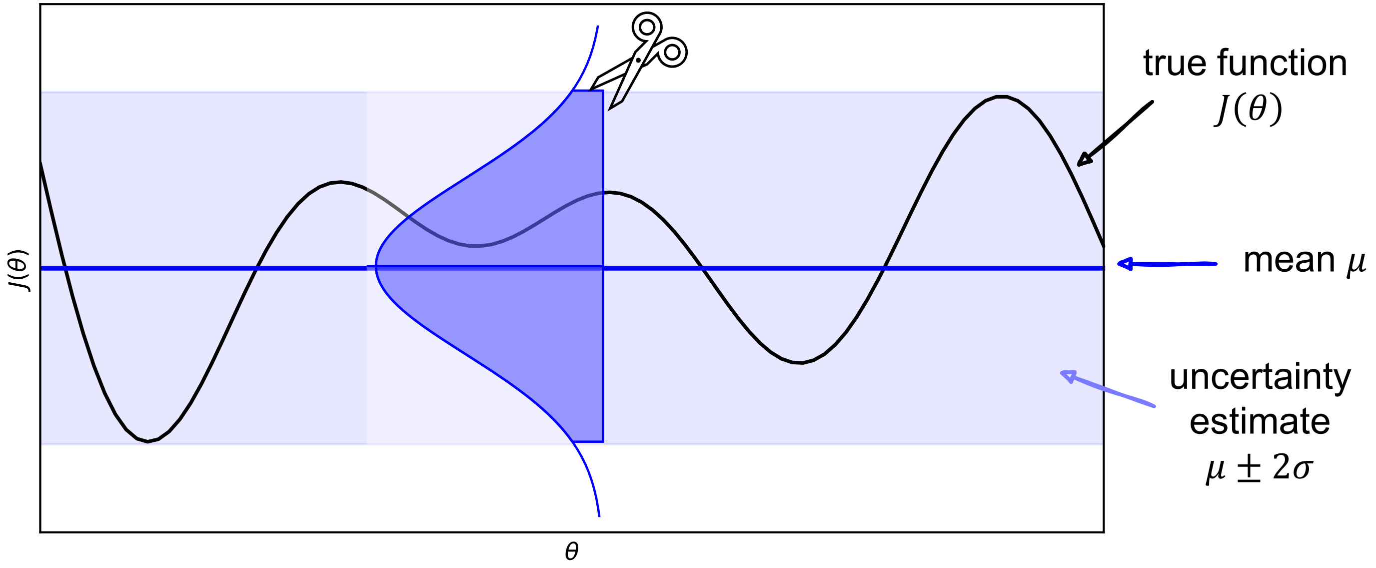



**3) Acquisition function**
- Determine the next evaluation $\theta_{i+1} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \alpha(\,\theta \mid \mathcal{D}_i\,)$
- Expresses trade-off between exploration (test new solutions) and exploitation (use what is known to be "best")


[1] .

# Gaussian process: An infinite dimensional normal distribution

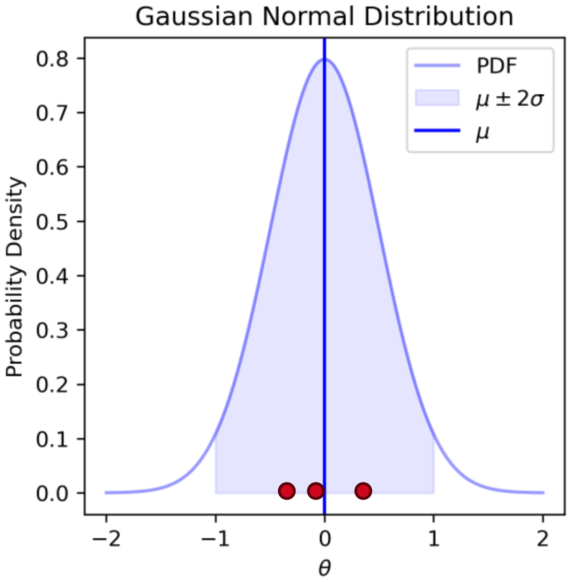


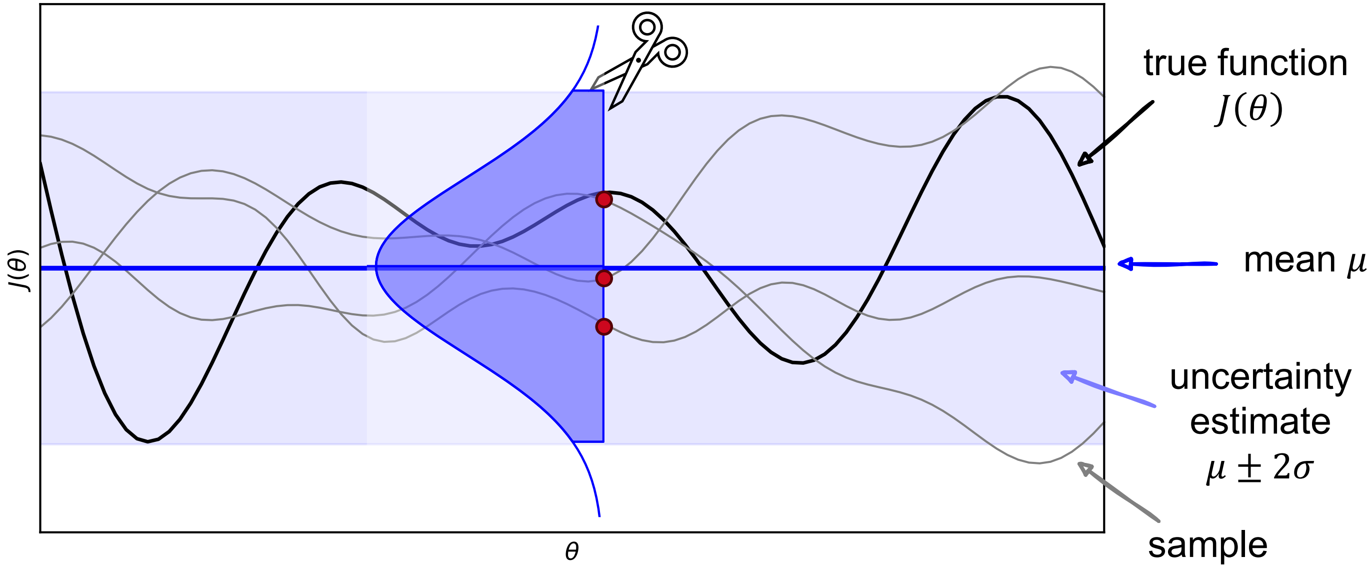






[1] Rasmussen, C. E., Williams, C. K. I, Gaussian Processes for Machine Learning. The MIT Press, 2006.

# Gaussian process: An infinite dimensional normal distribution



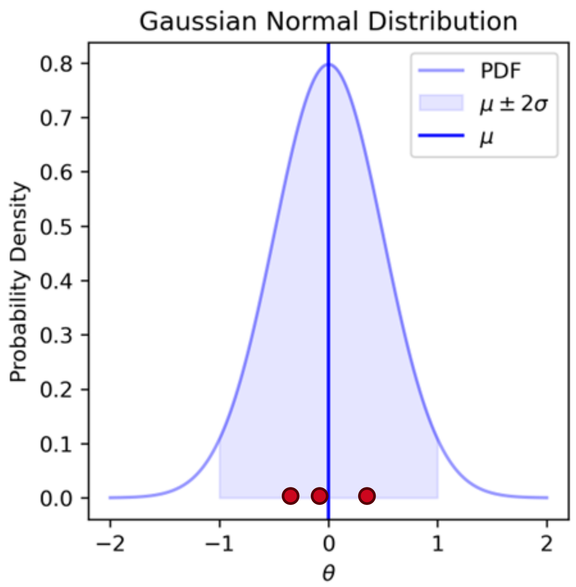

**Key take aways:**

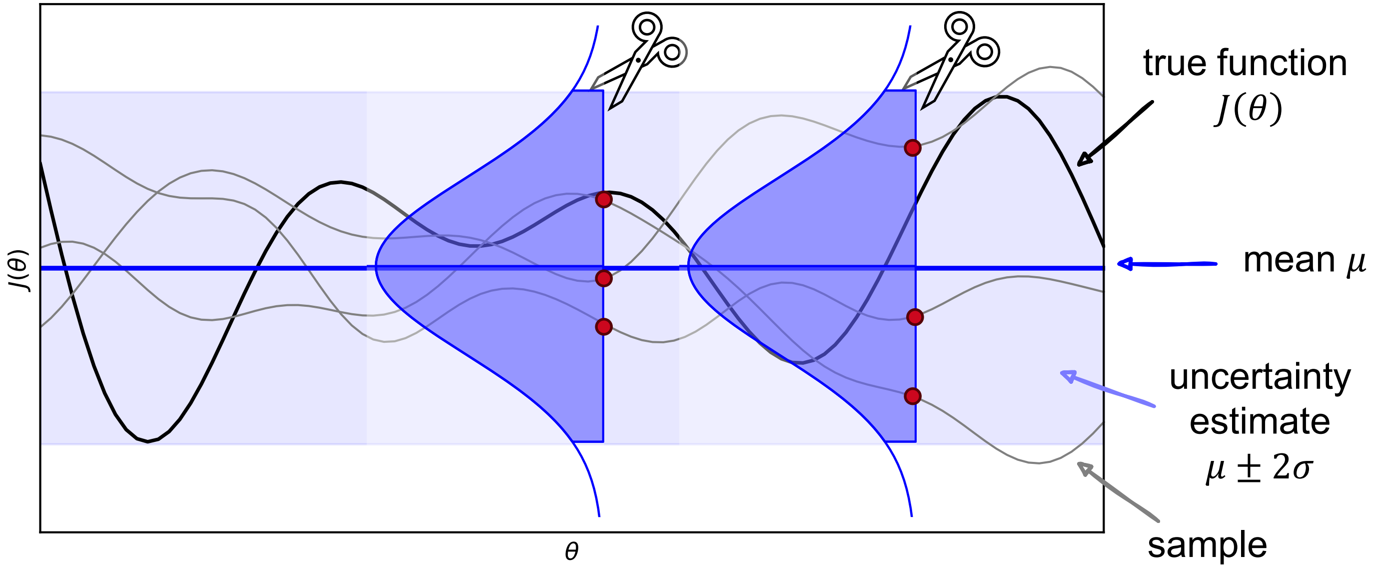
- GPs are distributions over functions



[1] Rasmussen, C. E., Williams, C. K. I, Gaussian Processes for Machine Learning. The MIT Press, 2006.

# Gaussian process: An infinite dimensional normal distribution




$$\boxed{J(\theta)} \sim \mathcal{GP}(m(\theta), k(\theta, \theta'))$$

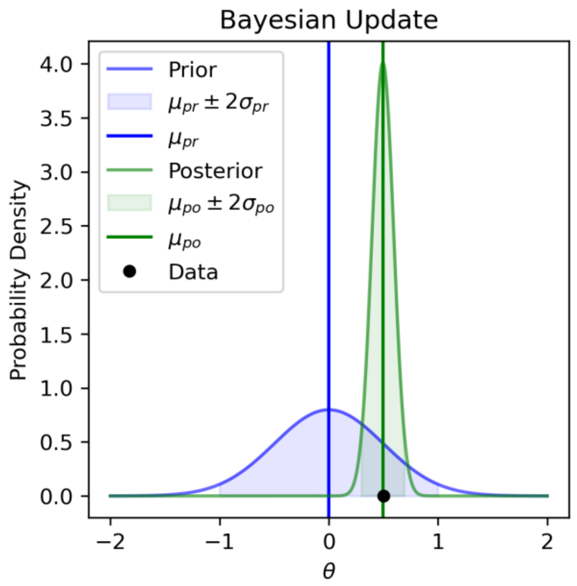
**Key take aways:**

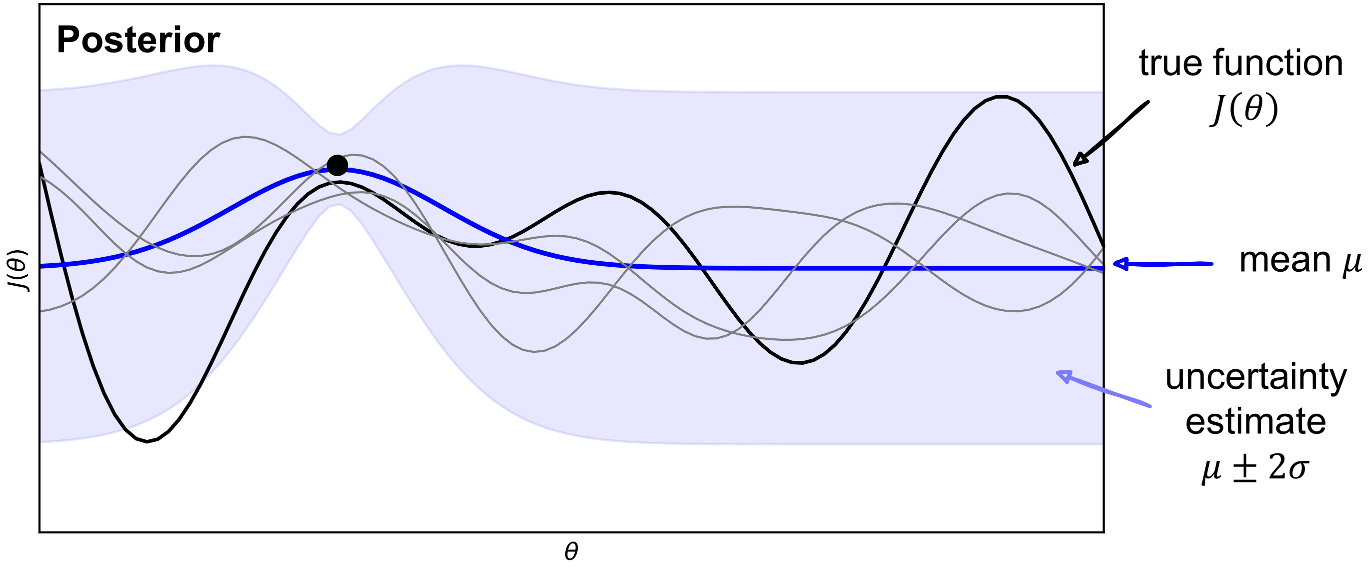
- GPs are distributions over functions
- Defined by mean and covariance (kernel) function
- Kernel ↔ smoothness/structure of functions



[1] Rasmussen, C. E., Williams, C. K. I, Gaussian Processes for Machine Learning. The MIT Press, 2006.

# Gaussian process: An infinite dimensional normal distribution




$$\boxed{J(\theta)} \sim \mathcal{GP}(m(\theta), k(\theta, \theta'))$$

## Gaussian Process Regression

**Prior distribution**
$p(f)$
$\Updownarrow$
$p(\hat{y}|\hat{\theta})$

Inference using
**Bayes theorem**

likelihood
$p(y|\,\theta, f)$

data
$(\boldsymbol{y}, \boldsymbol{\theta})$

**Posterior distribution**
$p(f\,|\,\boldsymbol{y}, \boldsymbol{\theta})$
$\Updownarrow$
$p(\hat{y}|\hat{\theta}, \boldsymbol{y}, \boldsymbol{\theta})$

Likelihood describes our observation model, $y_i = J(\theta_i) + \epsilon_i, \epsilon \sim \mathcal{N}(0, \sigma_n)$

## Key take aways:

- GPs are distributions over functions
- Defined by mean and covariance (kernel) function
- Kernel ↔ smoothness/structure
- **Uncertainty low near data, high elsewhere**



[1] Rasmussen, C. E., Williams, C. K. I, Gaussian Processes for Machine Learning. The MIT Press, 2006.

# Main ingredients of Bayesian optimization

$\theta$ → Black-box function → $y = \boxed{J(\theta)} + \epsilon$

**Goal:** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathbb{E}[J(\theta)]$

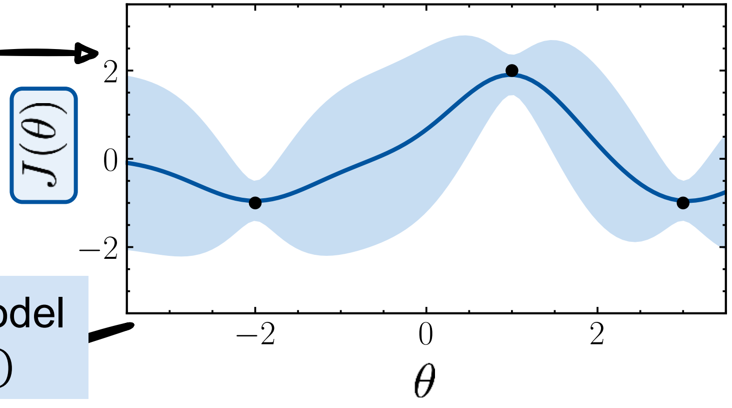
**1) Observations** / evaluations / experiments:
- Noisy evaluation as $J_i = J(\theta_i) + \epsilon_i\ ,\ \epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$, iid
- **Data set** from observations: store all relevant information
  $\mathcal{D}_i = \{(\theta_1, J_1), \dots, (\theta_i, J_i)\}$

**2) Probabilistic ML model**

usually a Gaussian process

- Model belief about the unknown function

$\boxed{J(\theta)} \sim \mathcal{GP}(m(\theta), k(\theta, \theta'))$




Uses model of $J(\theta)$

**3) Acquisition function**
- Determine the next evaluation $\theta_{i+1} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \alpha(\, \theta \mid \mathcal{D}_i\,)$
- Expresses trade-off between exploration (test new solutions) and exploitation (use what is known to be "best")

Let's look at the acquisition function in more detail!





01_bo_tutorial.ipynb

# Choosing the next sample location: The acquisition function

- To choose the next sample, we aim to solve $\theta_{i+1} = \underset{\theta\in\Theta}{\text{argmax}}\, \alpha(\theta \mid \mathcal{D}_i)$

Uses **model** of $J(\theta)$, **not** true $J(\theta)$!

*Sampling and thus optimization is more straightforward!*

This is also a global optimization problem.

*What have we gained?*

There exist various acquisition functions:

1) **Probability of Improvement** [1]

$$\alpha_{\mathrm{PI}}(\theta \mid \mathcal{D}_t) = \mathbb{P}[J(\theta) > J(\theta_{\mathrm{best}})]$$

2) **Expected Improvement** [2]

$$\alpha_{\mathrm{EI}}(\theta \mid \mathcal{D}_t) = \mathbb{E}[\max\{0,\ J(\theta) - J(\theta_{\mathrm{best}})\}]$$

3) **Max-Value Entropy Search** [3]

$$\alpha_{\mathrm{MES}}(\theta \mid \mathcal{D}_t) = H(p(J|\mathcal{D}_t, \theta)) - \mathbb{E}[H(p(J|\mathcal{D}_t, \theta, J^*))]$$

4) **Upper-Confidence-Bound** [1,4]

$$\alpha_{\mathrm{UCB}}(\theta \mid \mathcal{D}_t) = \mu_{\mathcal{D}_t}(\theta) + \beta_t^{1/2}\sigma_{\mathcal{D}_t}(\theta)$$

**...**

**Acquisition function**

**Bayesian optimization**

1: $\mathcal{GP}(0, k)$, $\Theta \in \mathbb{R}^d$, $\mathcal{D}_0 = \emptyset$
2: **for** $i = 1, 2, \ldots$ to $T$ **do**
3: Train GP model with $\mathcal{D}_{i-1}$
4: $\theta_i \leftarrow \textsc{Query}(\mathcal{D}_{i-1})$
5: $J_i \leftarrow \textsc{Observation}(\theta_i)$
6: $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \{(\theta_i, J_i)\}$


[1] Kushner, H. J. "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise." Journal of Basic Engineering 86.1, 1964.
[2] Močkus, Jonas. "On Bayesian methods for seeking the extremum." Optimization Techniques IFIP Technical Conference Novosibirsk, 1975.
[3] Wang, Zi, and Stefanie Jegelka. "Max-value entropy search for efficient Bayesian optimization." International Conference on Machine Learning, 2017.
[3] Srinivas, Niranjan, et al. "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design." International Conference on Machine Learning, 2010.

# Bayesian optimization on a standard benchmark

Let's test BO on a standard benchmark!

**Objective:**

- Hartmann 6D – a six-dimensional objective function with six local minima

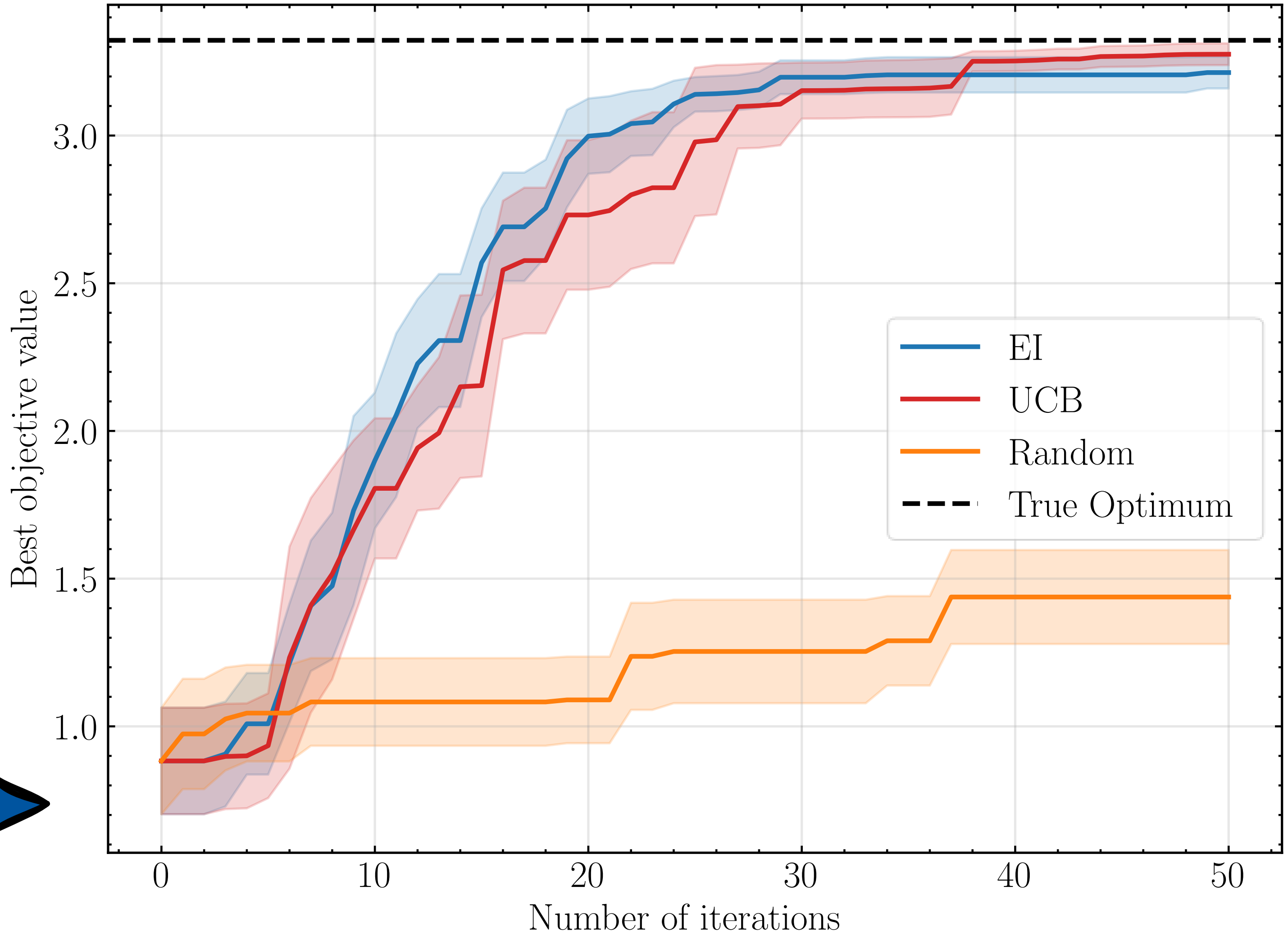
$$J(\theta) = \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(\theta_j - P_{ij})^2\right)$$

**Comparison:**

- BO with Expected Improvement (EI)
- BO with UCB ($\beta = 2$)
- Random search

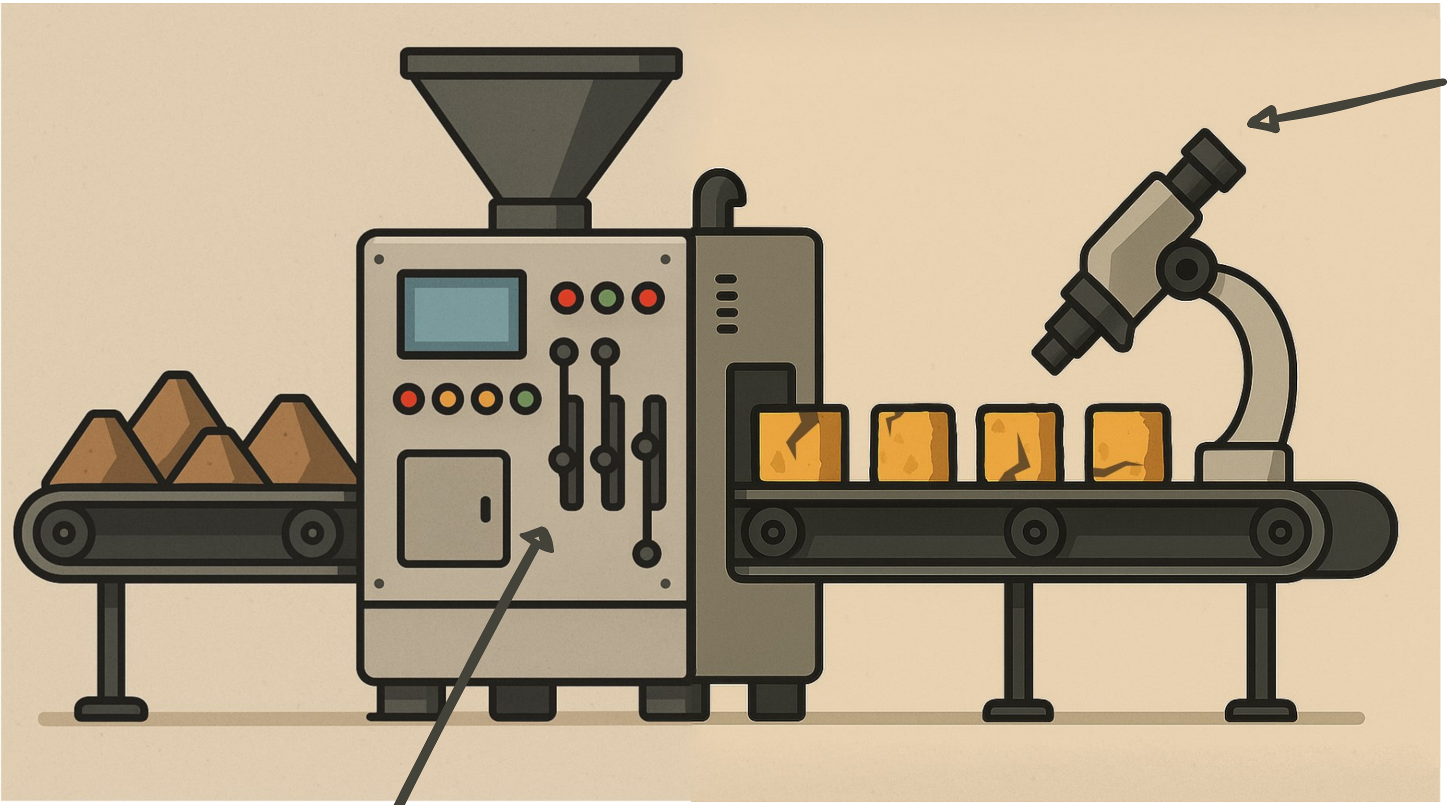
**Setup:**

- Feasible domain as $[0,1]^6$
- 6 initial points
- 50 iteration



# Bayesian Optimization in Application

# Bayesian Optimization in Application



observed quality
$y = J(\theta) + \epsilon$

parameters $\theta$

**BO Checklist:**

1. Parameters $\theta$
2. Parameter bounds $\Theta$
3. Objective $J(\theta)$
4. Experiment design
5. Experiment budget

**Goal:** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}} \, \mathbb{E}[J(\theta)]$

# Bayesian Optimization in Application

observed quality
$y = J(\theta) + \epsilon$

parameters $\theta$

**BO Checklist:**

1. Parameters $\theta$
   → 3 parameters $\theta = [\theta_1, \theta_2, \theta_3]$, continuous
2. Parameter bounds $\Theta$
   → Lever settings from 0 (down) to 3 (up) $\Theta = [0,3]^3 \subset \mathbb{R}^3$
3. Objective $J(\theta)$
   → surface quality measurement
4. Experimental design
   → produce 4 parts per experiment and measure the average quality of the last 3 parts
5. Experiment budget
   → 10*dim = 30

**Goal:** Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathbb{E}[J(\theta)]$



02_bo_application_example.ipynb

# Extensions of Bayesian Optimization

# Extensions of Bayesian Optimization

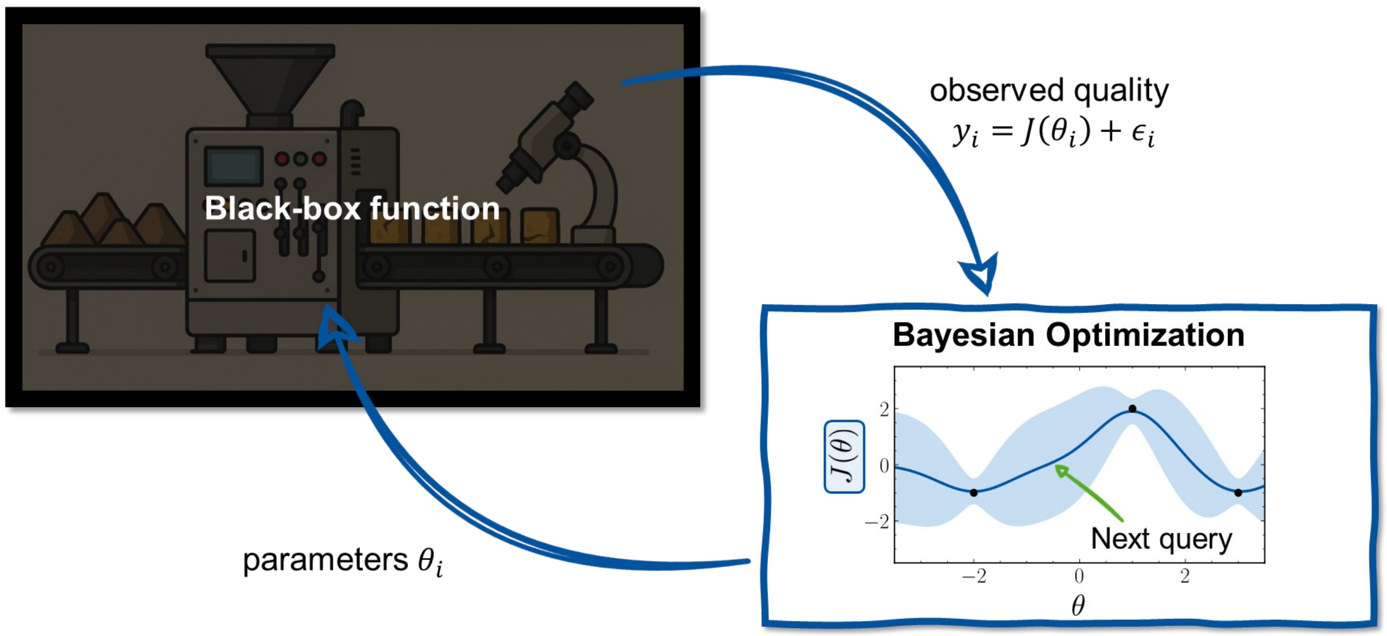
| Challenge | BO Extension |
|---|---|
| High-dimensions | Local BO [1,2] |
| Experiments fail | Crash constraints [3,4] |
| Constraints | Constrained BO [5] |
| Safety requirements | Safe BO [6,7] |
| Something is changing | Time-varying/ Event-triggered BO [8,9] |
| Multiple experiments in parallel | Batch BO [10] |
| Non-measurable objective function | Preferential BO [11] |
| Information from multiple sources | Multi-fidelity BO [12] |

[1] Eriksson, David, et al. "Scalable global optimization via local Bayesian optimization." Advances in Neural Information Processing Systems 32, 2019.
[2] Müller, Sarah, Alexander von Rohr, and Sebastian Trimpe. "Local policy search with Bayesian optimization." Advances in Neural Information Processing Systems 34, 2021.
[3] Stenger, David, and Dirk Abel. "Benchmark of Bayesian optimization and metaheuristics for control engineering tuning problems with crash constraints." arXiv preprint arXiv:2211.02571, 2022
[4] Marco, Alonso, et al. "Robot learning with crash constraints." IEEE Robotics and Automation Letters, 2021
[5] Gardner, Jacob R., et al. "Bayesian optimization with inequality constraints." International Conference on Machine Learning, 2014.
[6] Berkenkamp, Felix, Angela P. Schoellig, and Andreas Krause. "Safe controller optimization for quadrotors with Gaussian processes." IEEE International Conference on Robotics and Automation, 2016
[7] Fiedler, Christian, et al. "On Safety in Safe Bayesian Optimization." Transactions on Machine Learning Research, 2024
[8] Brunzema, Paul, Alexander Von Rohr, and Sebastian Trimpe. "On controller tuning with time-varying Bayesian optimization." IEEE Conference on Decision and Control, 2022.
[9] Brunzema, Paul, et al. "Event-triggered time-varying Bayesian optimization." Transactions on Machine Learning Research, 2025.
[10] González, Javier, et al. "Batch Bayesian optimization via local penalization." Artificial intelligence and statistics, 2016.
[11] González, Javier, et al. "Preferential bayesian optimization." International Conference on Machine Learning, 2017.
[12] Marco, Alonso, et al. "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization." 2017 IEEE International Conference on Robotics and Automation, 2017.

# High-dimensional multifidelity Bayesian Optimization

# Event-Triggered Safe Bayesian Optimization

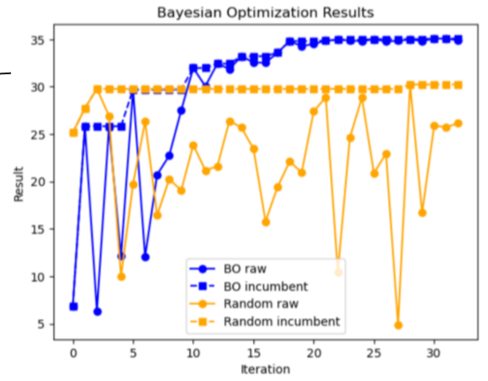
# Summary




## Bayesian Optimization in Application

**BO Checklist:**

1. Parameters $\theta$
2. Parameter bounds $\Theta$
3. Objective $J(\theta)$
4. Experiment design
5. Experiment Budget




## Three Ingredients of Bayesian Optimization

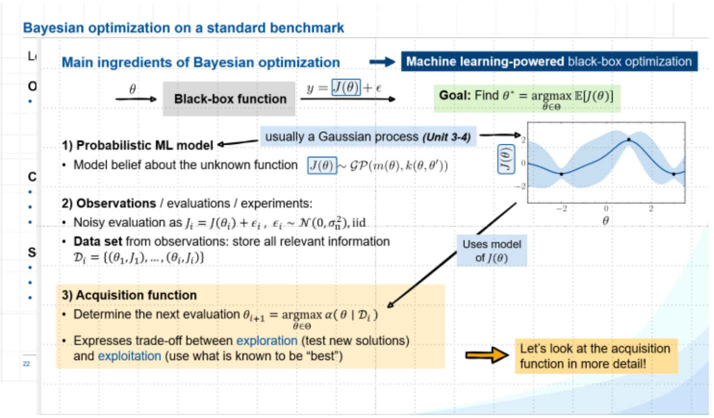
observations

acquisition functions

surrogate model

## Extensions of Bayesian Optimization

# Advertisement MOOC

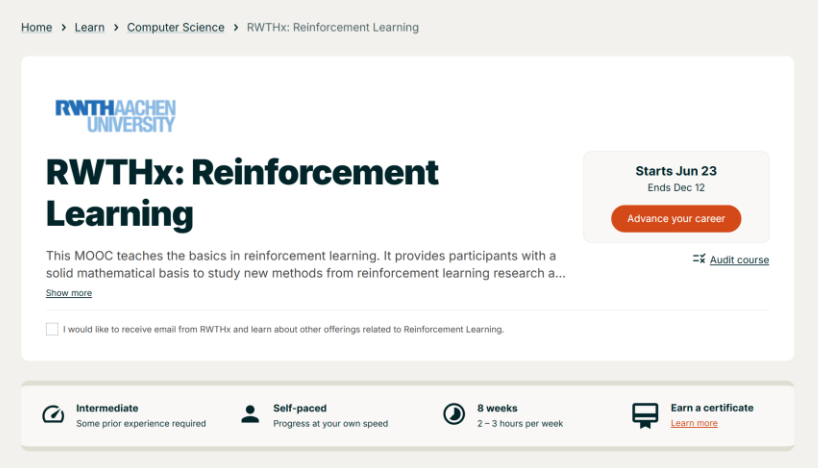


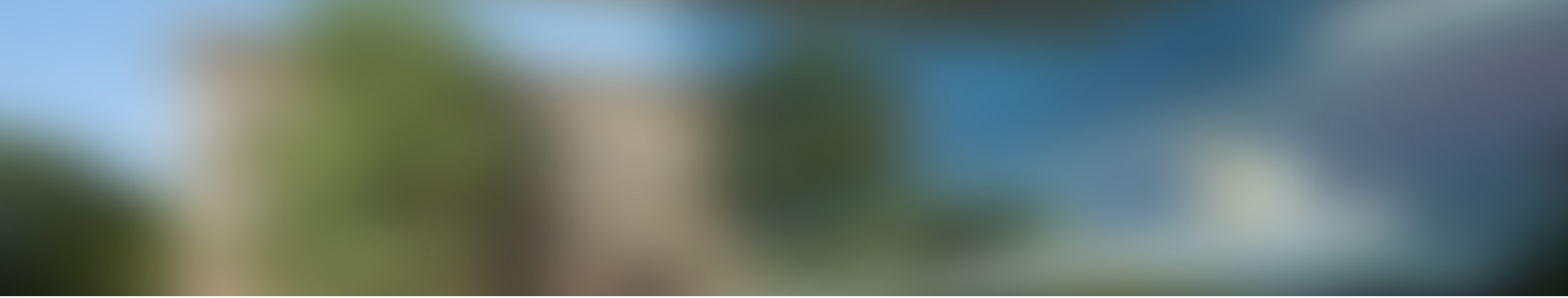

## Coming soon:
## RWTHx: Learning-based control MOOC

Thanks to Paul Brunzema, for providing slides from this MOOC for this tutorial.

## Already existing MOOC:
## RWTHx: Reinforcement Learning

https://www.edx.org/learn/computer-science/rwth-aachen-university-reinforcement-learning-2





# Thank you for your attention and enjoy the poster session!