IEEE Conference on Artificial Intelligence 2026

Sustainable Hyperparameter Optimization

Laurens Bliek

Department of Industrial Engineering & Innovation Sciences
Eindhoven University of Technology (TU/e)

Sasan Amini

Data Science Institute
Hasselt University

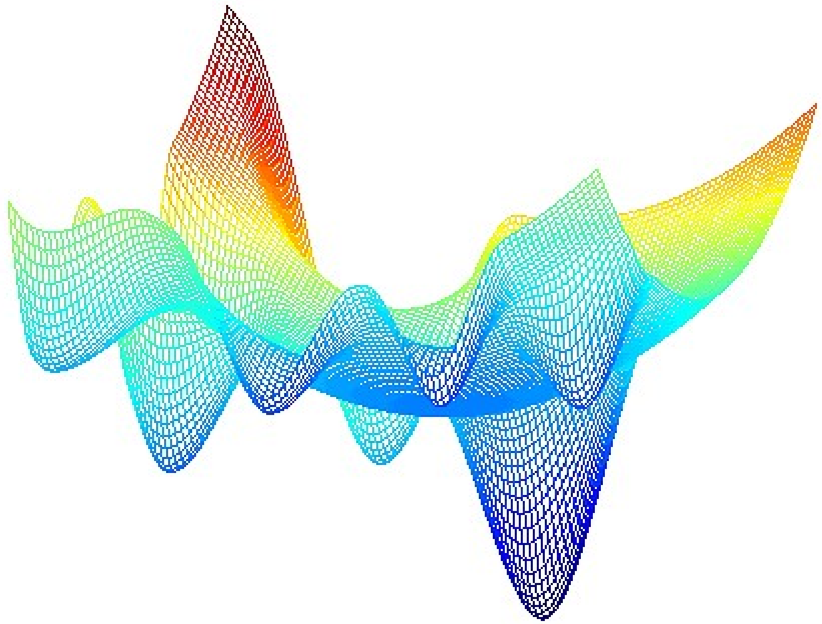
Overview

- Part I: HPO (15:30-17:00)

- **Problem definition**
- **Overview of HPO algorithms**
- **Introduction to Bayesian Optimization**
- **Main ideas in multi-objective BO**

- Part II: Sustainable HPO (17:30-20:00)

- **Short intro Sustainable AI**
- **Sustainable AI in practice**
- **Exercise**

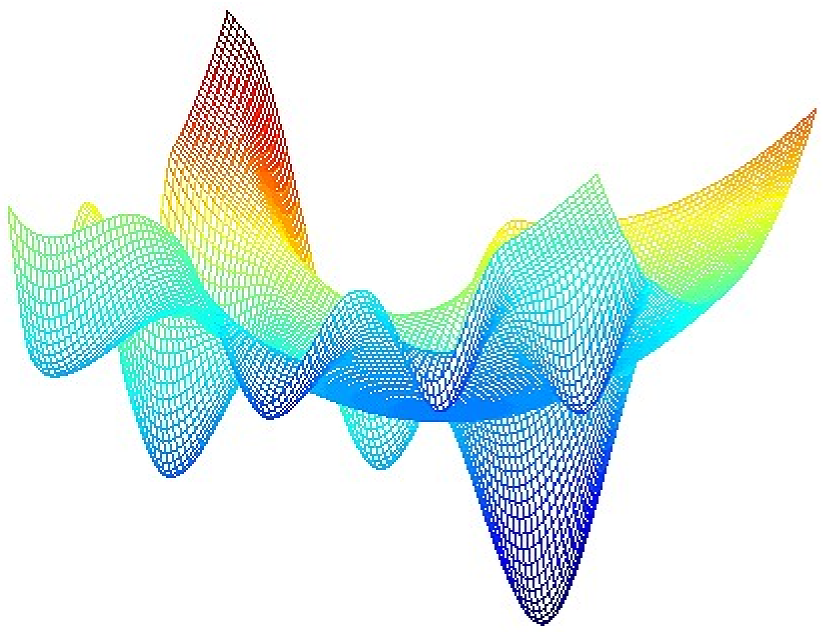


If you want to work ahead on part 2:

You can find the tutorial code here:
https://github.com/ai-for-decision-making-tue/SustainableHPO_example → (scan)

Overview

- Part I: HPO (15:30-17:00)
  - Problem definition
  - Overview of HPO algorithms
  - Introduction to Bayesian Optimization
  - Main ideas in multi-objective BO
- Part II: Sustainable HPO (17:30-20:00)
  - Short intro Sustainable AI
  - Sustainable AI in practice
  - Exercise



Hyperparameter Optimization

Parameters: learned from data; e.g.:

- weights in a neural network,
- Intercept and slope coefficients in linear regression,
- …

Hyperparameters: set before training; e.g.:

- learning rate and number of layers in a neural network,
- number of trees and tree depth in a random forest model,
- …

Hyperparameters can make or break model performance.
The same model, with different hyperparameter settings, may yield drastically different results.

Hyperparameter Optimization

The goal is to find the hyperparameter setting that gives the best validation performance:

$$\min_{x \in \mathcal{X}} f(x)$$

where:

- *x*: hyperparameters
- *f(x)*: validation error (expensive to evaluate)

Why Is Hyperparameter Optimization Difficult?

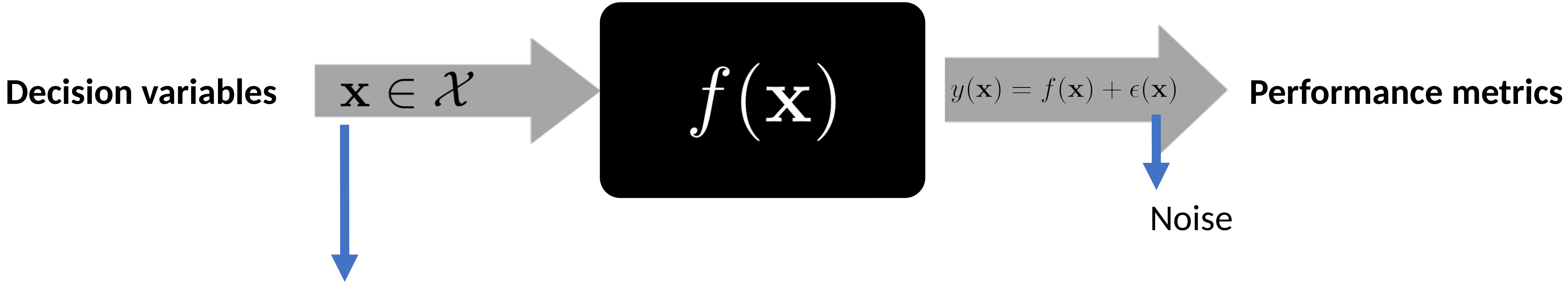
- Expensive (training can take minutes → days)
- No gradient information (black-box)
- Stochastic (noise from training)
- The search space is often complex and heterogeneous

Problem definition

HPO belongs to the class of **expensive black-box optimization** problems.

- only input/output information is available.

-Needs simulator/experiment
-Expensive to evaluate (effort, money, time)



Small number of function evaluations/samples possible

How do people solve this problem?

- **Trial-and-error**
- **Grid search**
- **Random search**
- **Evolutionary strategies**

Computationally intensive

For expensive black-box problems, we need methods that **learn from limited data** and **efficiently search the solution space** to find good solutions with fewer evaluations.

- **Bayesian Optimization**

Bayesian Optimization

Bayesian Optimization is a sequential strategy for optimizing expensive black-box functions using a probabilistic surrogate model*.

Key components:

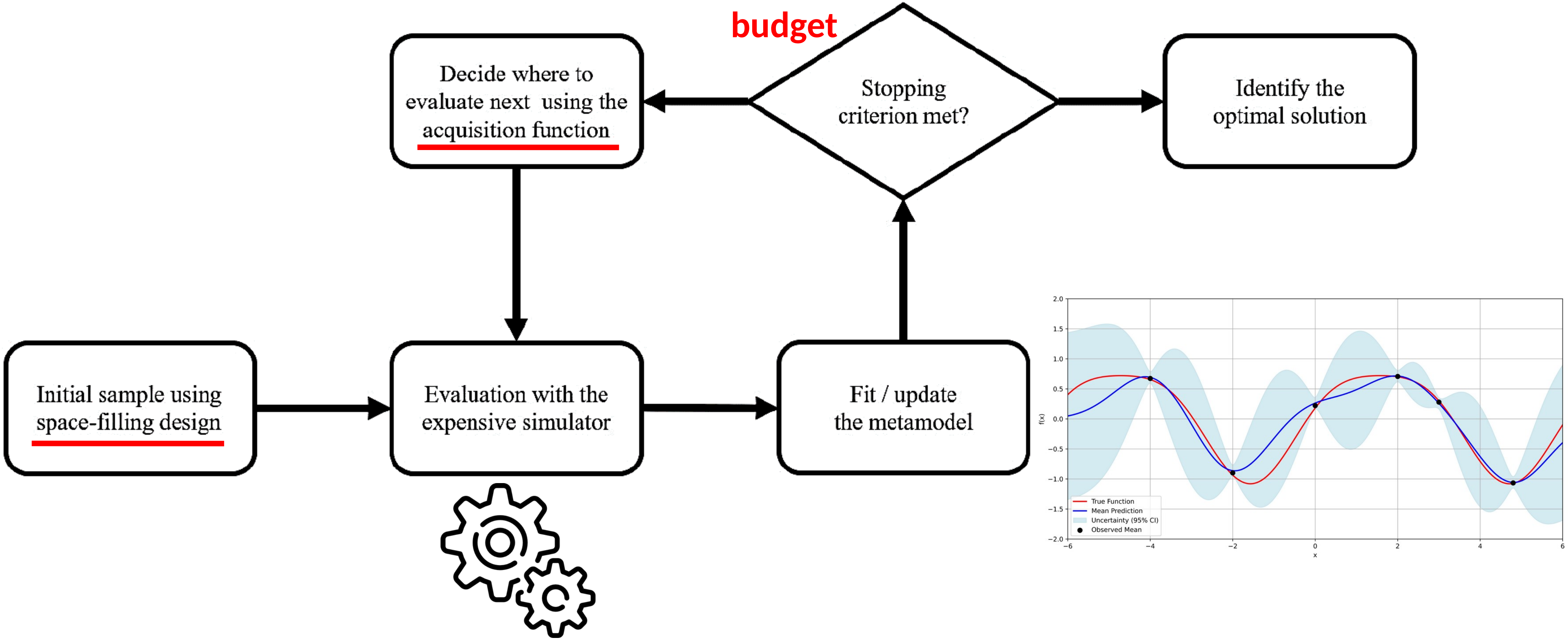
- Surrogate model (e.g., Gaussian Process),
- Acquisition function (guides next evaluation),
- Sequential updates based on observed data.

✓ **Well-suited for hyperparameter optimization**

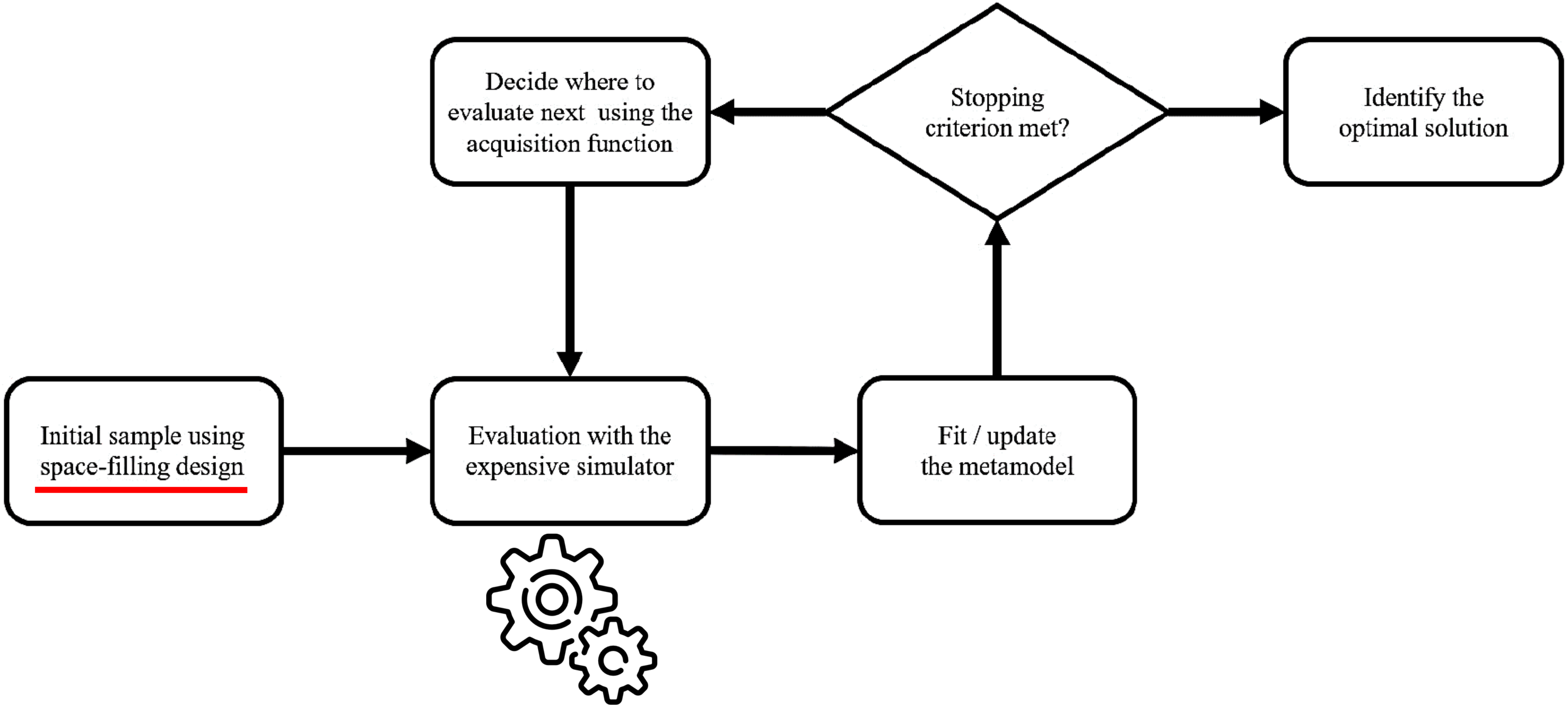
*Also called metamodel, or response surface model

General steps in Bayesian Optimization

Limited computational budget



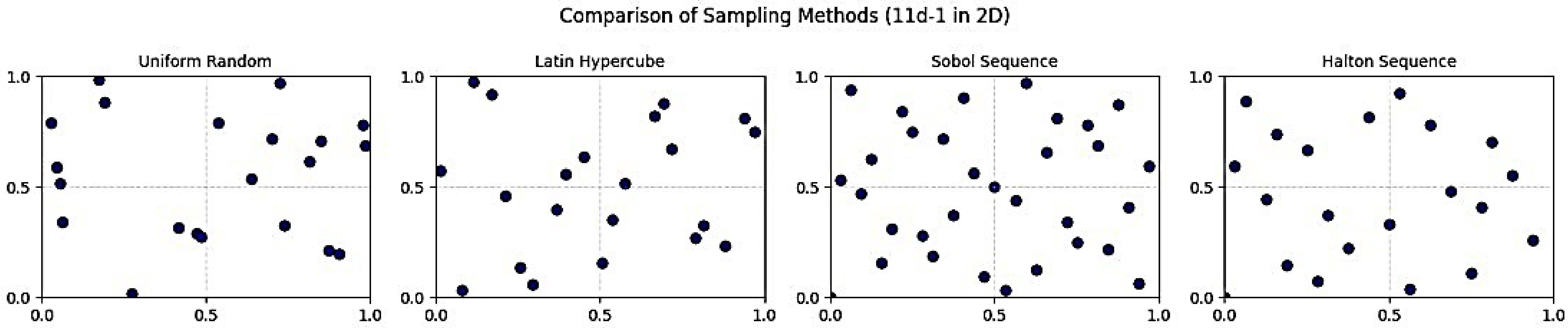
General steps in Bayesian Optimization



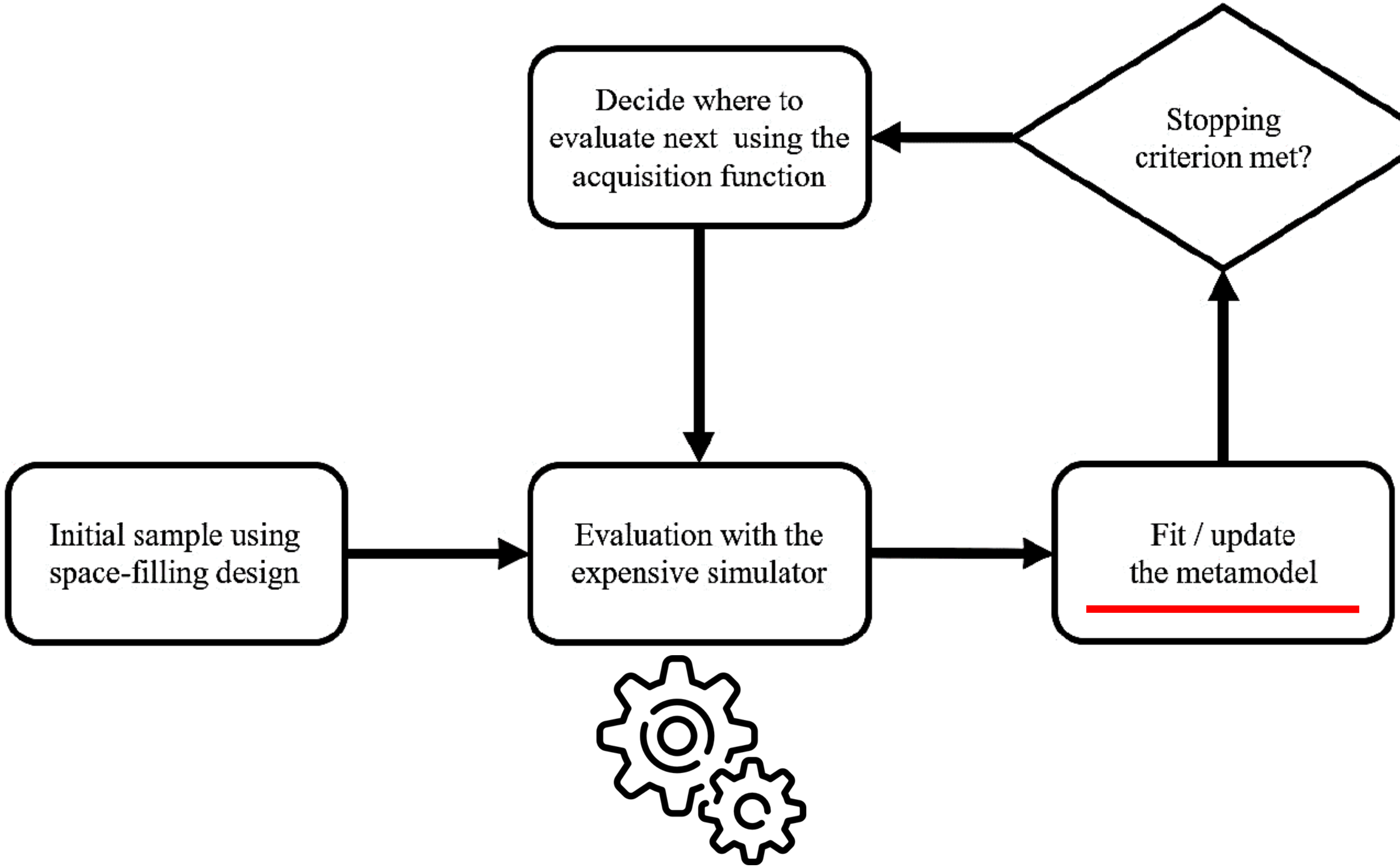
Initial sample

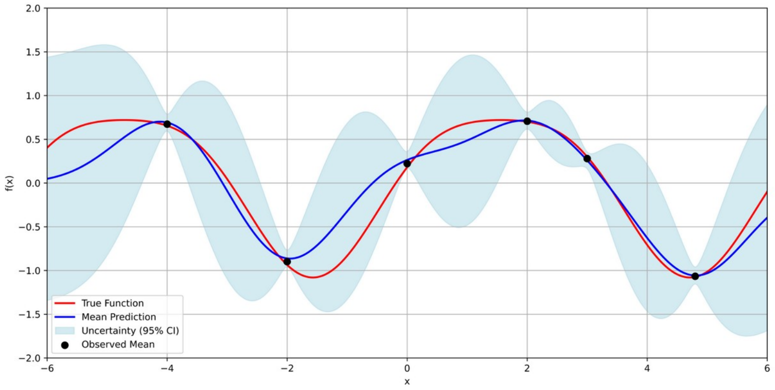
It has been shown that the success of metamodel-based optimization algorithms in achieving fast convergence to the optimal solution largely depends on the **quality of the initial data** used to train the metamodels.

The purpose of this step is to create a **space-filling** set of design points, i.e. a set of points that effectively covers the entire space without leaving significant gaps or underrepresented regions.



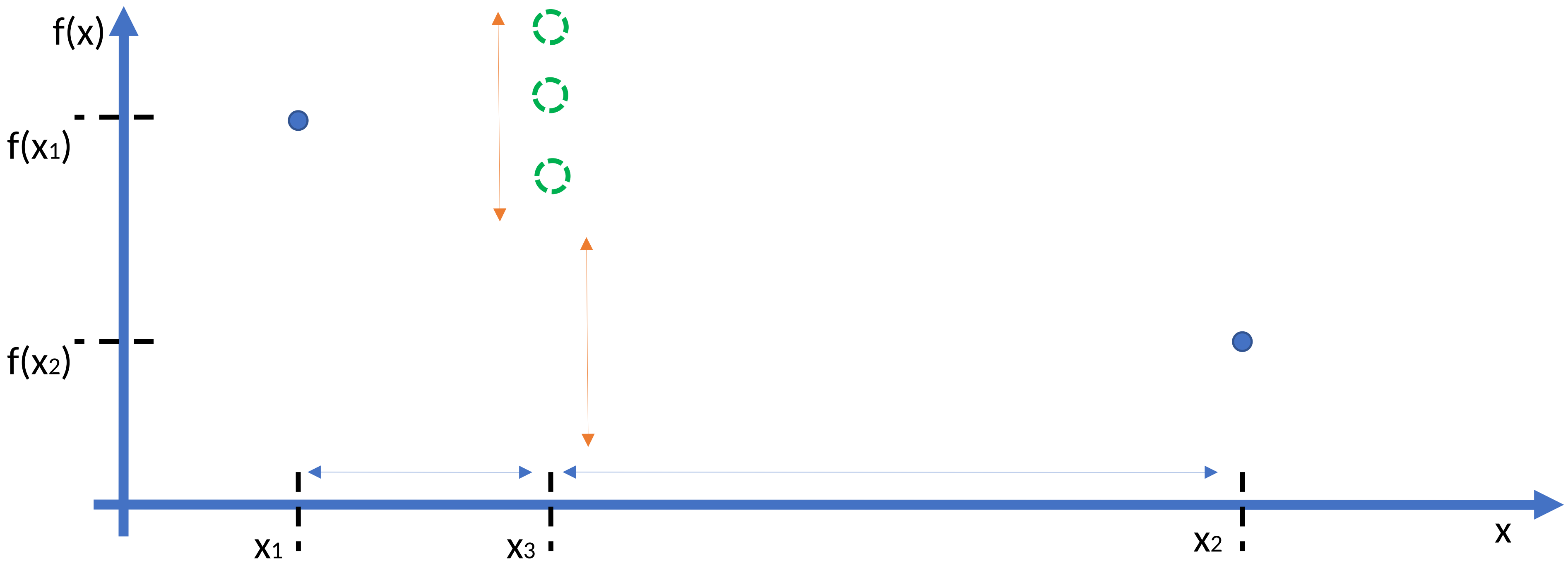
General steps in Bayesian Optimization





Gaussian Processes Regression

GPR is a spatial interpolation method used to obtain predictions at unsampled locations based on observed data. It assumes that the **distance between sample points** reflects a spatial correlation that can be used to **explain variation in the surface**.



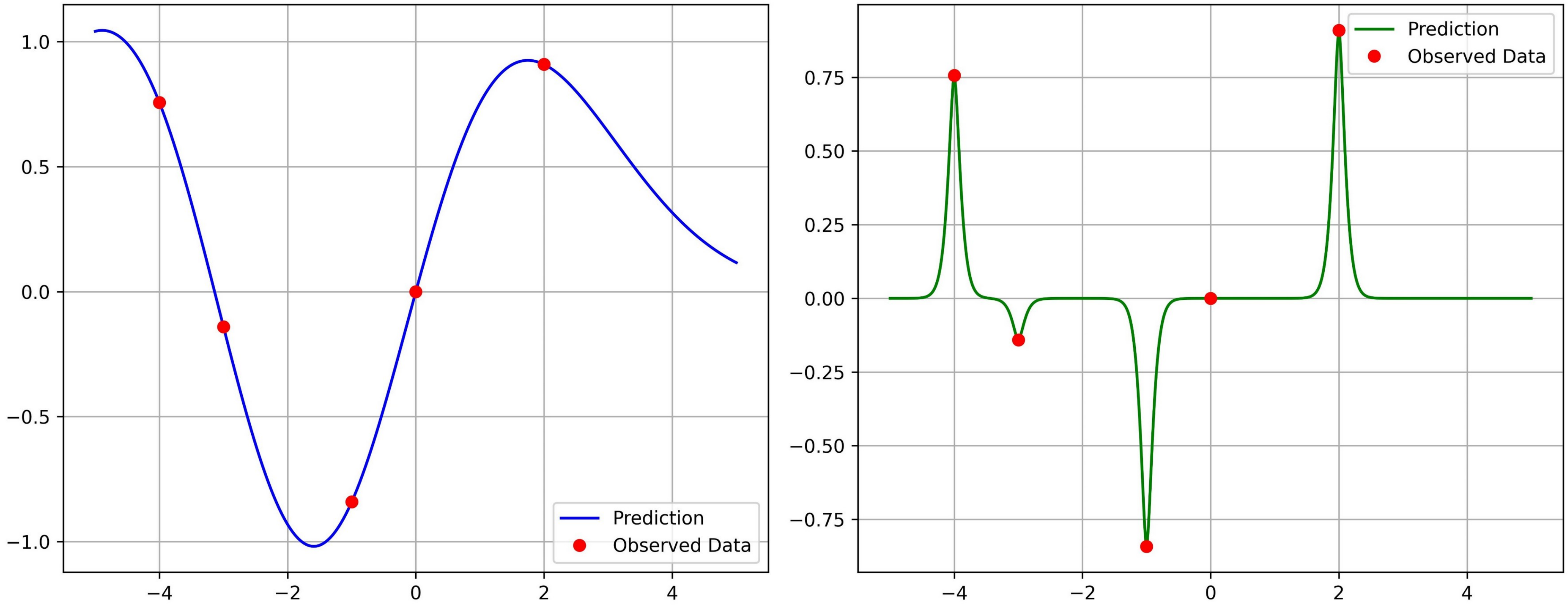
GPR

The **kernel function** quantifies how the function values (**outputs**) at different **input** locations are related.

- **Data points that are close to each other in the solution space are more likely to have similar output values, so they tend to have high correlation.**

Many different types of kernels (covariance functions) can be used; the choice of covariance function directly impacts the model’s ability to capture the underlying function.

GPR



Same observations, different predictions

GPR

Some of commonly used kernel functions in the literature are:
the squared exponential, Matern, …

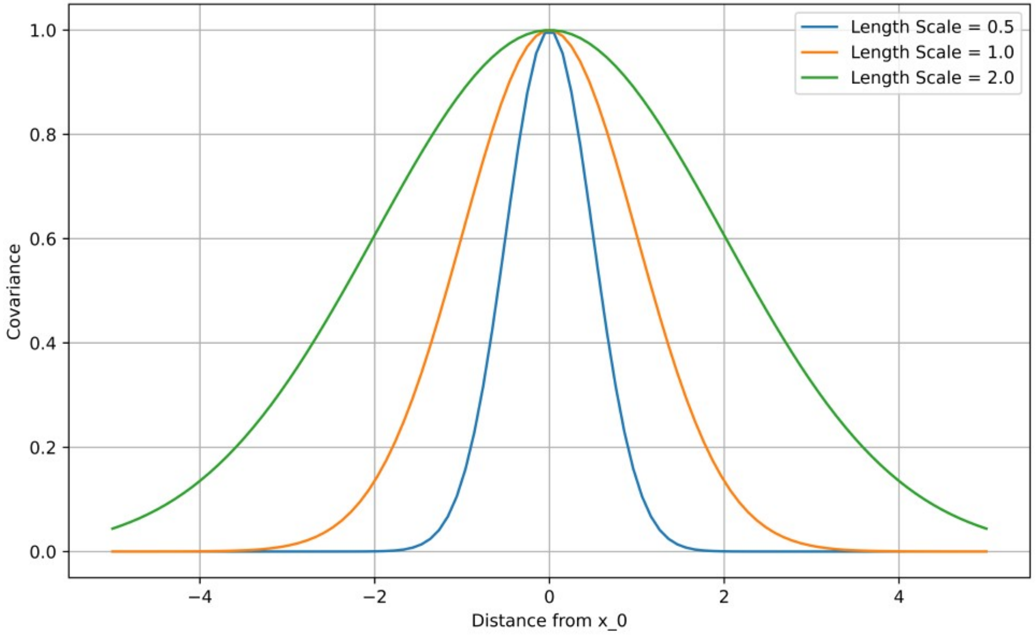
$$k_{SE}(.,.) = \sigma^2 exp(-\frac{d^2}{2l^2})$$ → squared exponential

$$k_{M_{3/2}}(.,.) = \sigma^2(1+\frac{\sqrt{3}d}{l})exp(-\frac{\sqrt{3}d}{l})$$ → Matern

$$k_{M_{5/2}}(.,.) = \sigma^2(1+\frac{\sqrt{5}d}{l}+\frac{5d^2}{3l^2})exp(-\frac{\sqrt{5}d}{l})$$

signal variance

length scale



For ease of interpretation, the plot assumes $\sigma^2 = 1$; hence, the covariance values are equal to the correlation values.

Hyperparameters

➢ **Gaussian Process is non-parametric**

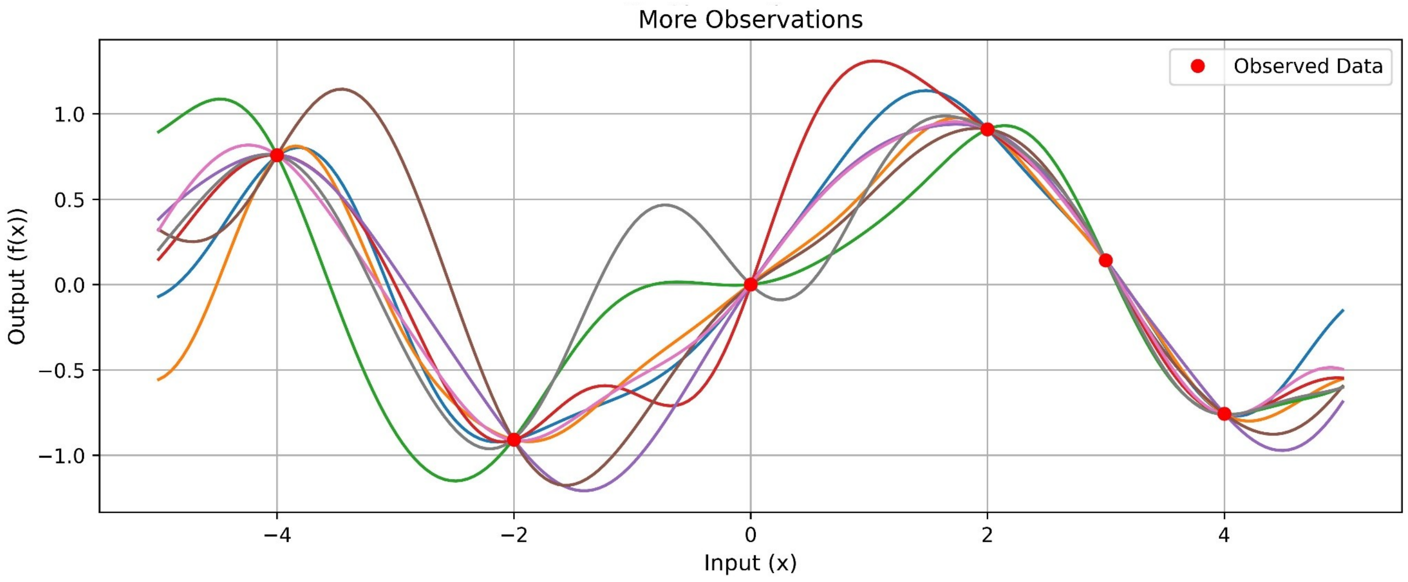
GPR, a Bayesian reasoning method

GPR model don’t give us a single function.
Instead, they define a **distribution over all possible functions** that fit our observed data.

New observation

Prior belief

Posterior belief



GPR

Short demo:

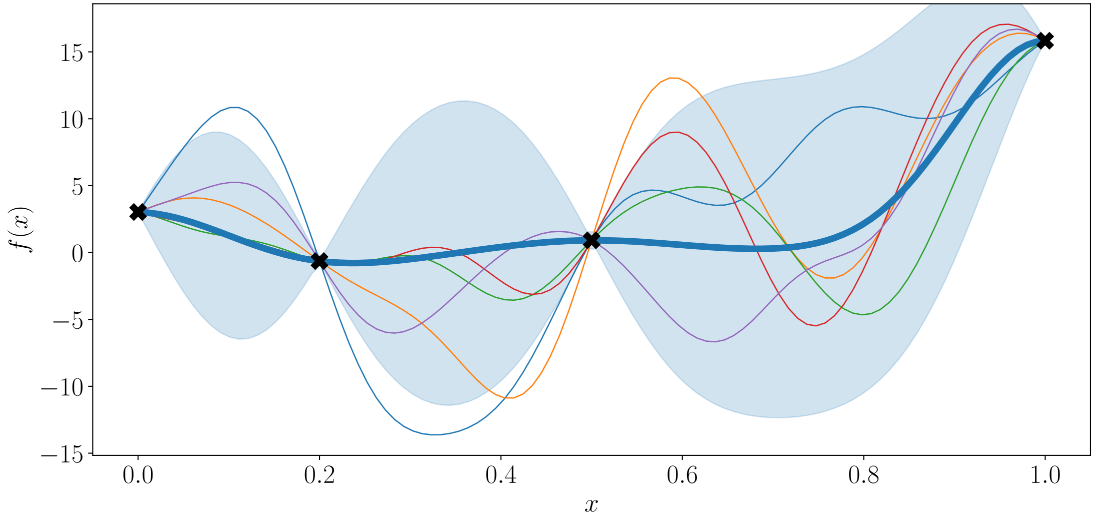
https://smlbook.org/GP/

GPR, a Bayesian reasoning method

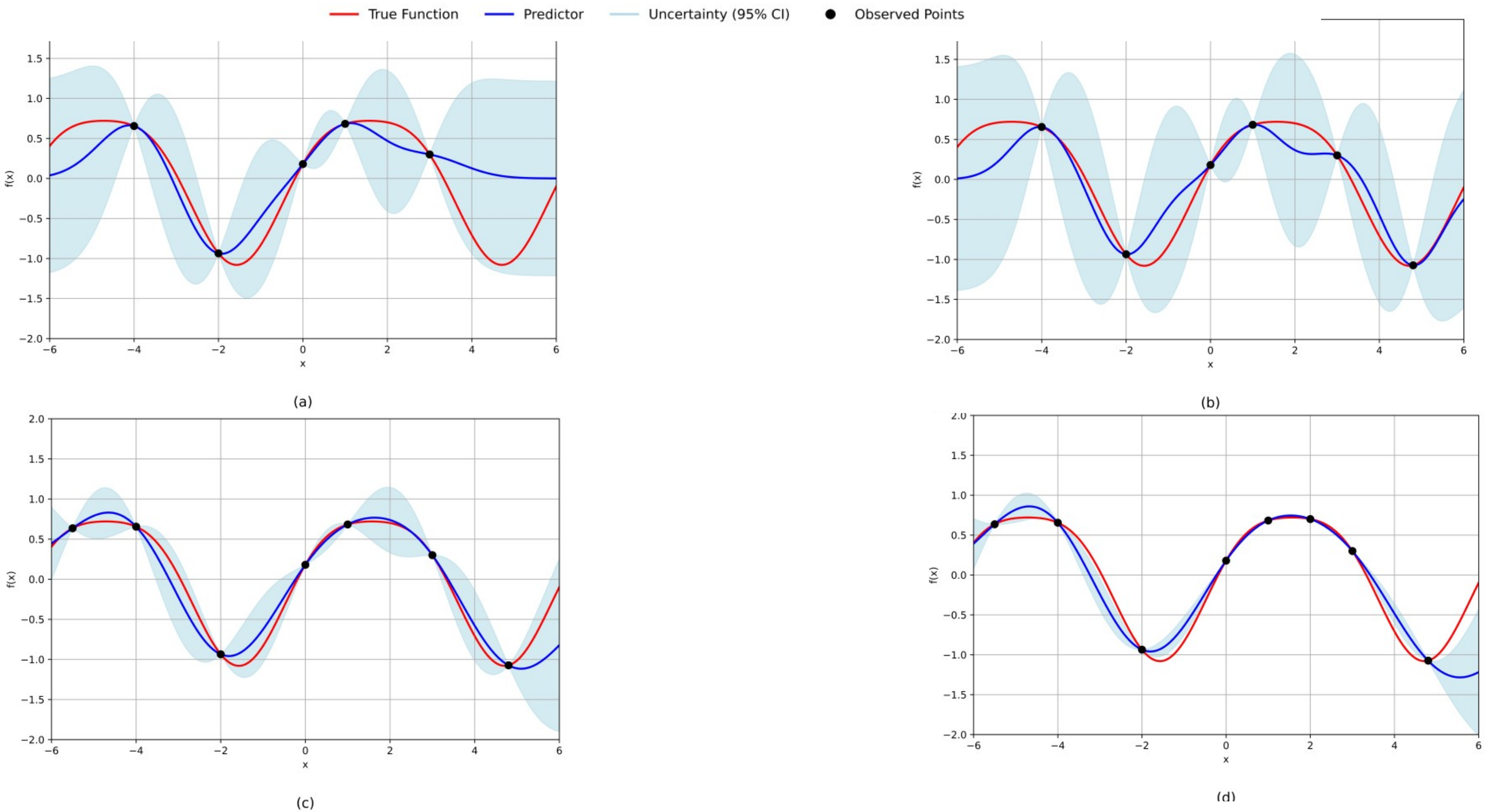
What happens during the training of the GPR model?

The model is conditioned on (limited) I/O dataset

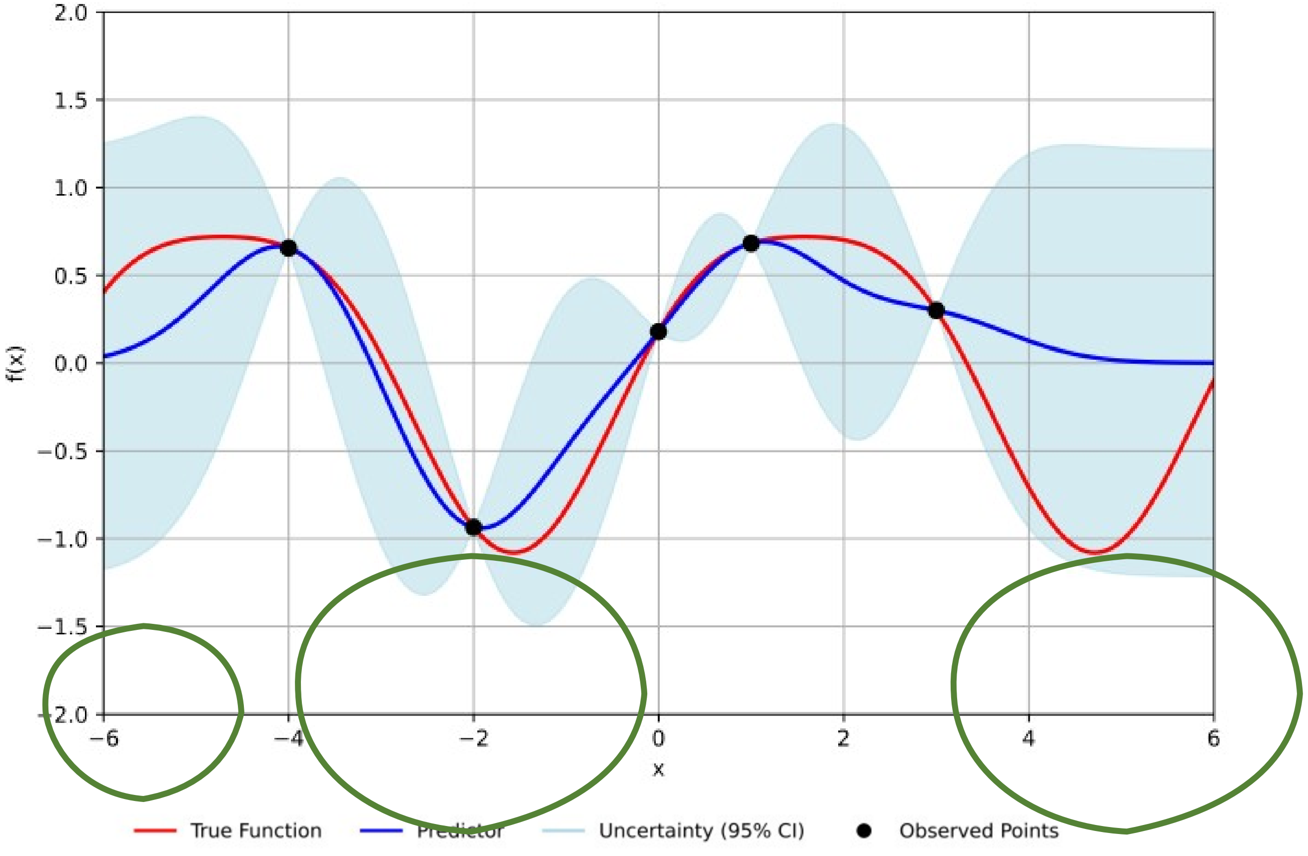
- Kernel's hyperparameters are optimized (using, e.g., maximum likelihood estimation)
- At observed input locations: predictor goes through the observed $\bar{\boldsymbol{y}}$
- At unobserved input locations:
  - Extrinsic uncertainty (= metamodel uncertainty): many outcomes still possible, but not equally likely
  - Mean of all possible outcomes = predictor



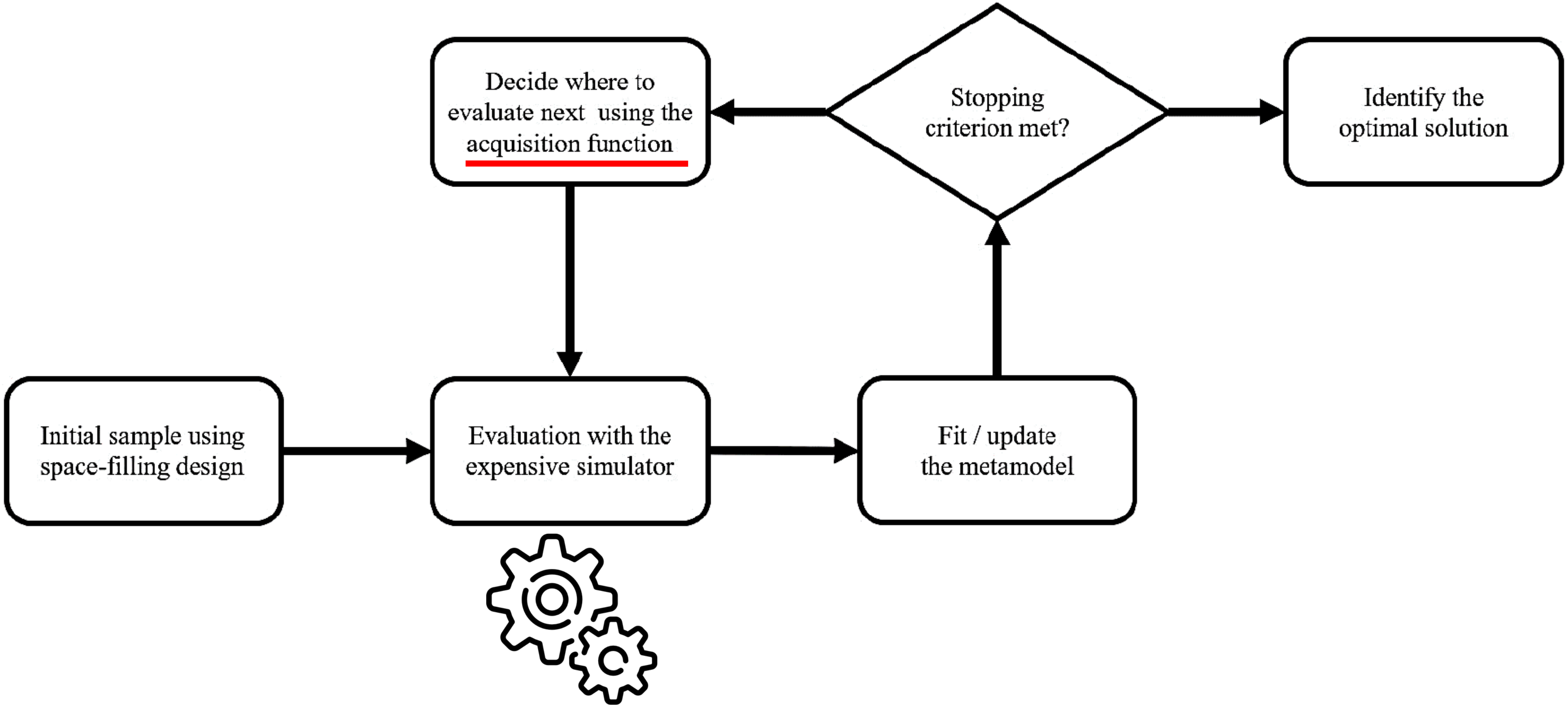
Progressive improvement of the model



How does this information help us to find the optima?



General steps in Bayesian Optimization



Acquisition function

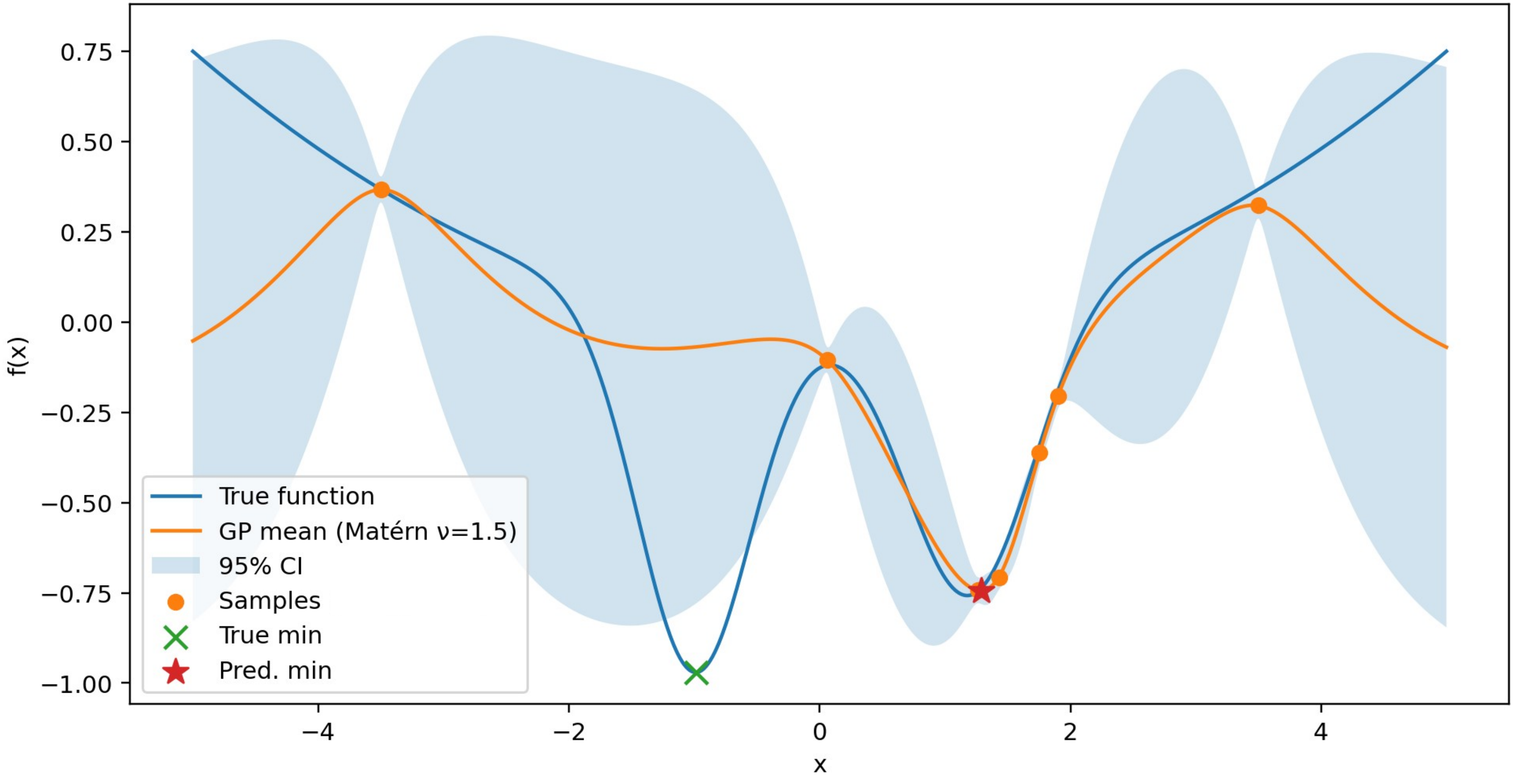
Trade off: Exploitation versus Exploration

Exploitation of the search space: sampling in areas with promising predictor values,
Exploration of the search space: sampling in areas with high predictor uncertainty.

Too much exploitation may lead to **premature convergence**

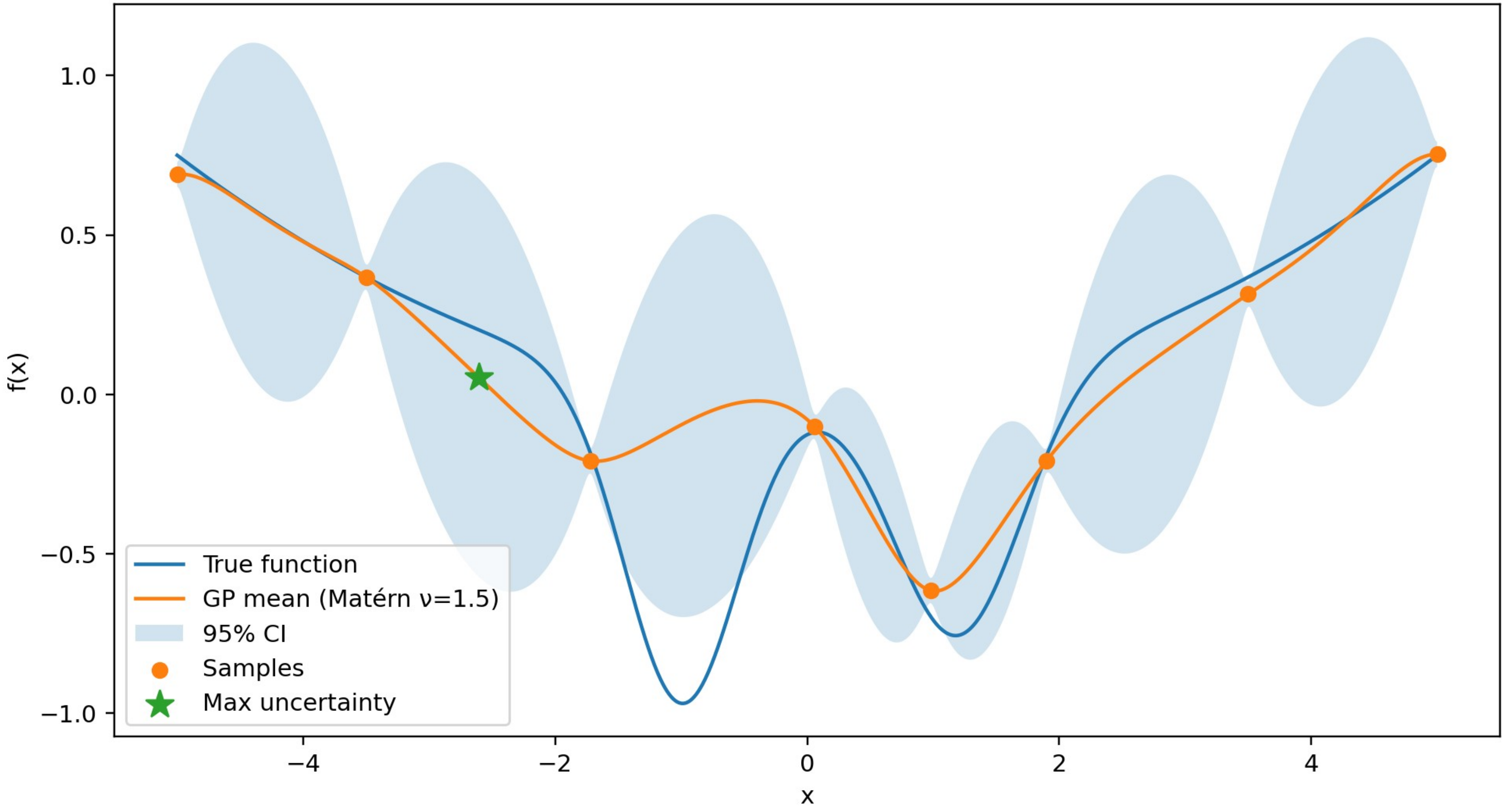
Too much exploration may **slow down convergence**

Acquisition function



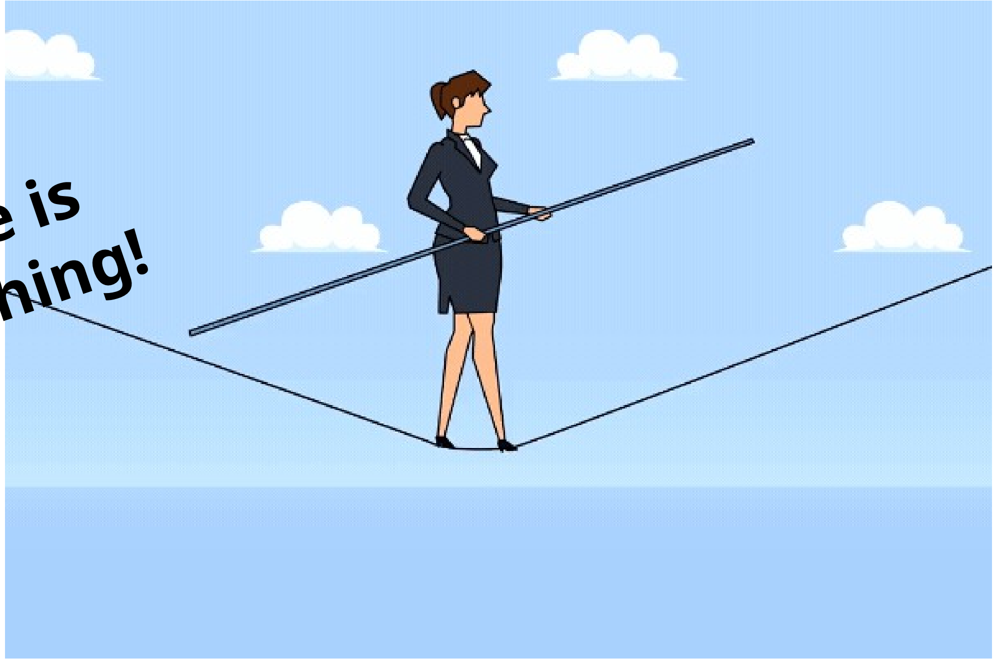
Too much exploitation → **stuck in local optimum**

Acquisition function



Too much exploration → **slow convergence**

Acquisition function

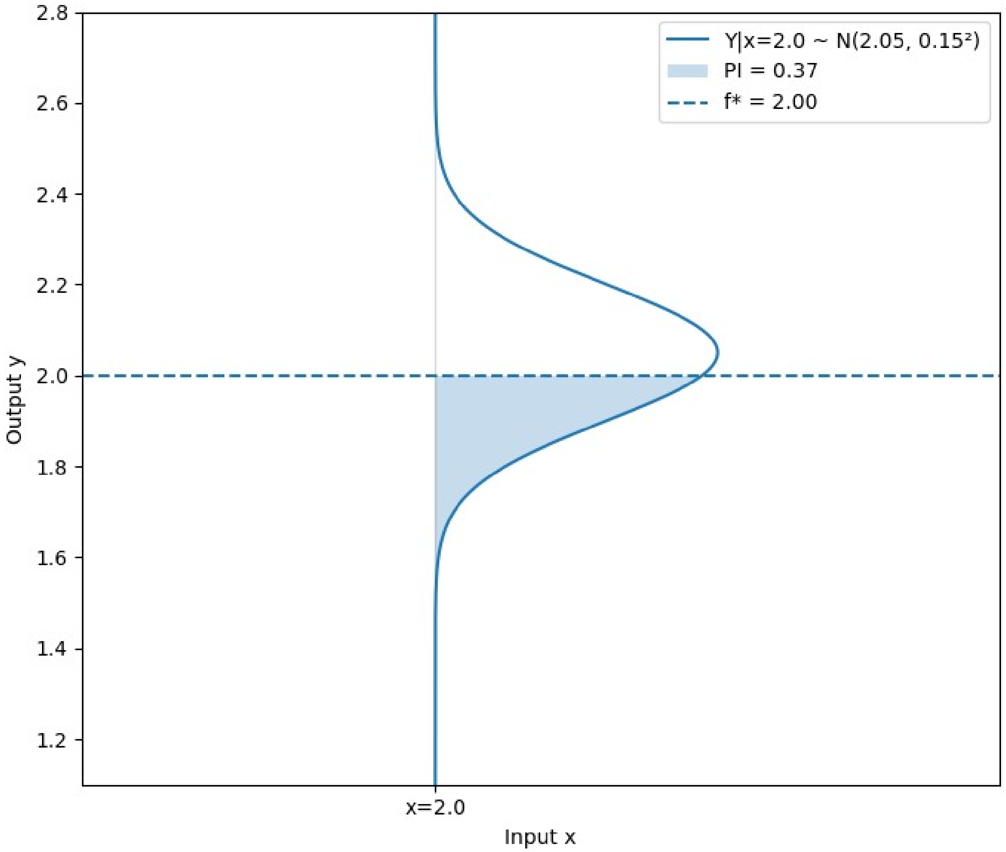


General rule:

During the **early stages** of optimization, **exploration** is often prioritized; As the optimization progresses, the surrogate model becomes more accurate, and the algorithm shifts towards **exploitation**.

Acquisition function

Probability of Improvement:



$$PI(\mathbf{x}) = Pr\left(f(\mathbf{x}) \leq f_{min}\right)) = \Phi\left(\frac{f_{min} - \mu(\mathbf{x})}{s(\mathbf{x})}\right)$$

- PoI does not estimate the **magnitude of the improvement** over the current best observed value.

Acquisition function

Expected Improvement:

$$EI(\mathbf{x}) = \underbrace{\underbrace{[f_{min} - \mu(\mathbf{x})]}_{\text{Potential gain}} \, \underbrace{\Phi\left(\frac{f_{min} - \mu(\mathbf{x})}{s(\mathbf{x})}\right)}_{\text{probability of gain}}}_{\text{the expected gain}} + \underbrace{s(\mathbf{x})\phi\left(\frac{f_{min} - \mu(\mathbf{x})}{s(\mathbf{x})}\right)}_{\text{Exploration}}$$

Encourages sampling where the mean looks promising.

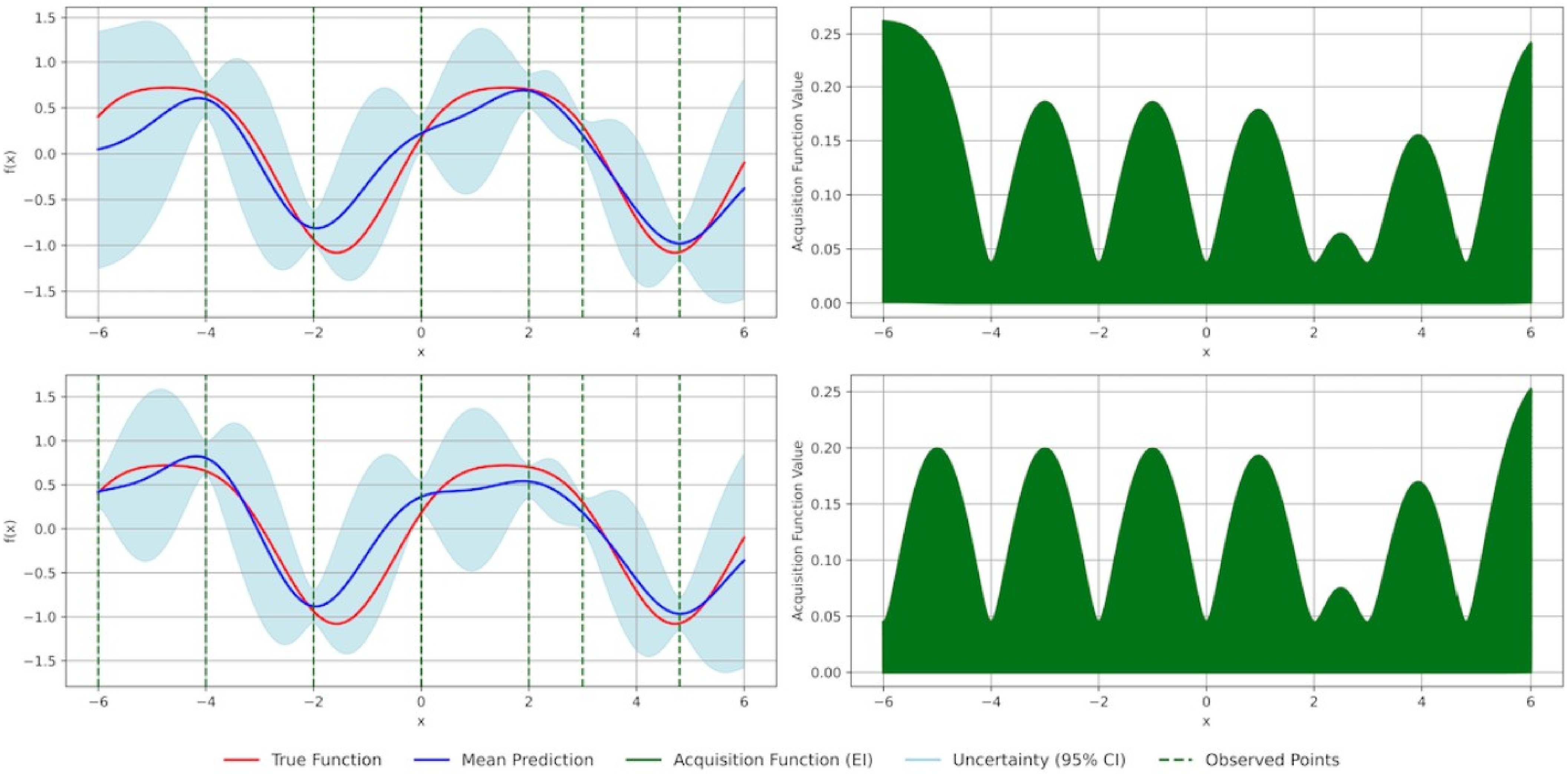
Encourages sampling where the model is uncertain.

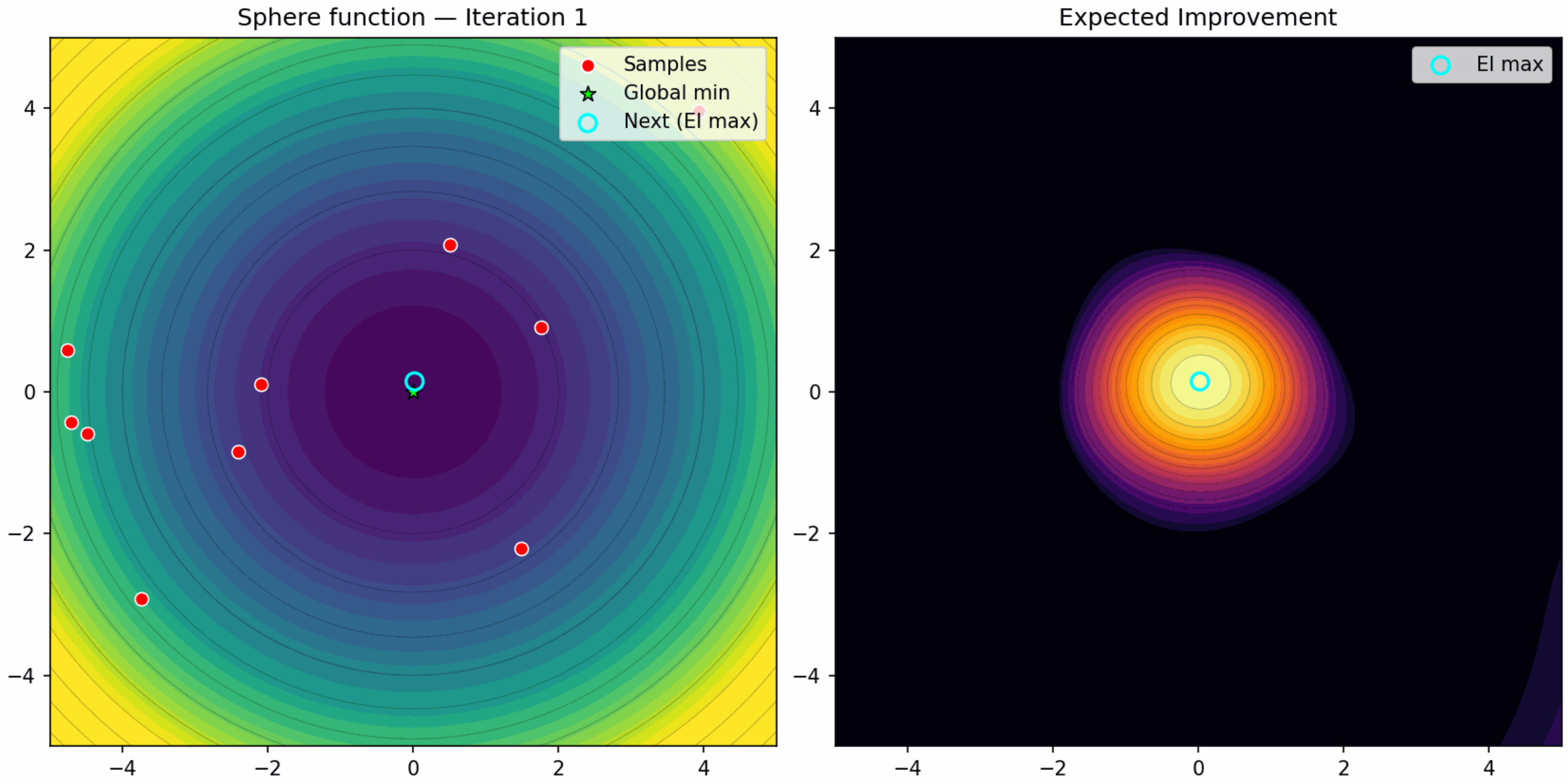
Acquisition function

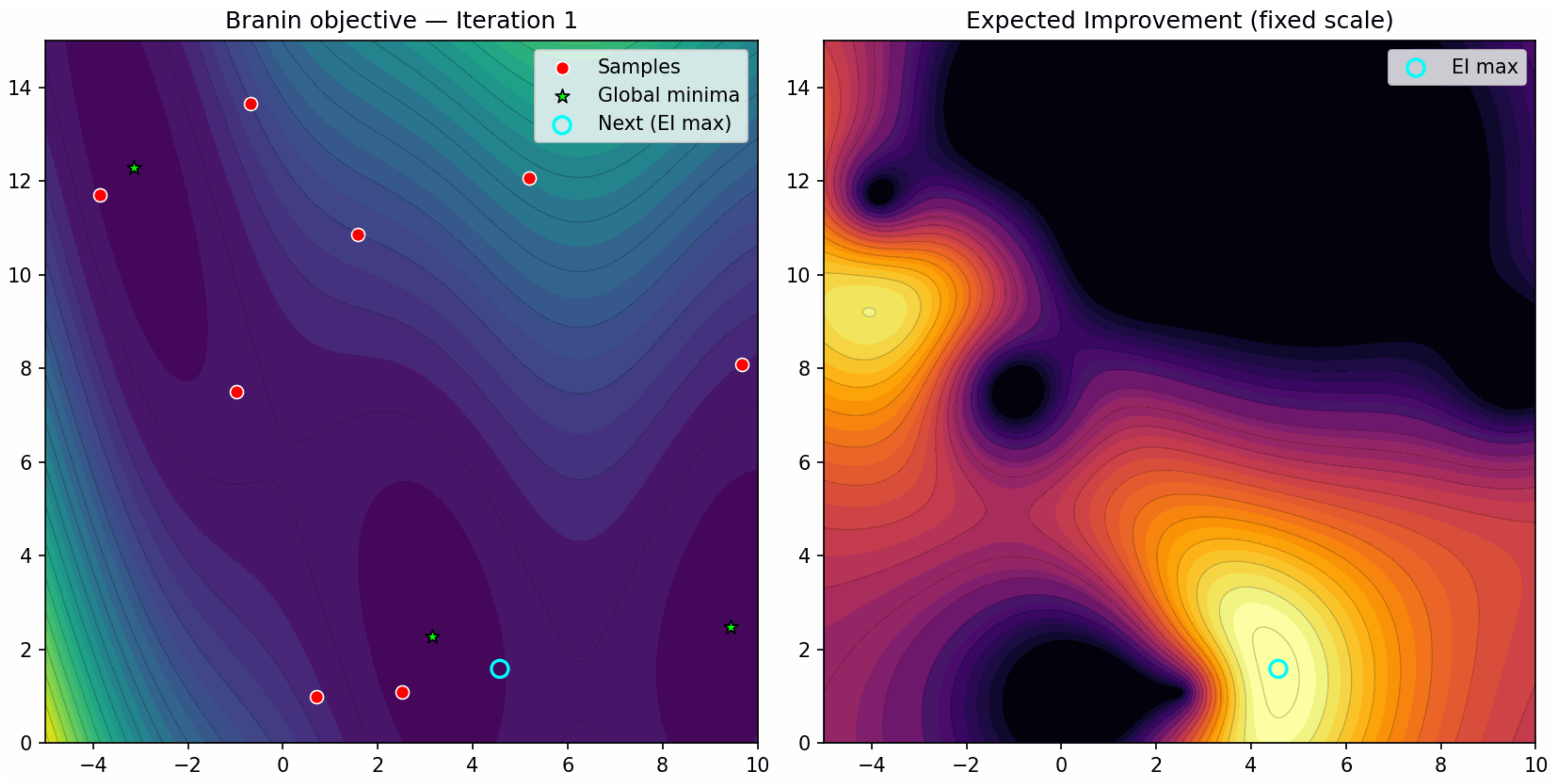
$$EI(\mathbf{x}) = [f_{min} - \mu(\mathbf{x})]\,\Phi\left(\frac{f_{min}-\mu(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\,\phi\left(\frac{f_{min}-\mu(\mathbf{x})}{s(\mathbf{x})}\right)$$

Situation	$\mu(x)$	$\sigma(x)$	Dominant term	Behavior
Very promising mean, small uncertainty	$\ll f^*$	small	$(f^* - \mu)\Phi(z)$	Exploit known good area
High uncertainty, mean around f^*	$\approx f^*$	large	$\sigma(x)\phi(z)$	Explore uncertain region
Poor mean, low uncertainty	$\gg f^*$	small	both terms small	Ignore (neither exploit nor explore)

Acquisition function




Sphere function — Iteration 1
Samples
Global min
Next (EI max)
Expected Improvement
EI max
4
2
0
−2
−4


Branin objective — Iteration 1
Samples
Global minima
Next (EI max)
Expected Improvement (fixed scale)
EI max
0
2
4
6
8
10
12
14
−4
−2
0
2
4
6
8
10

Acquisition function

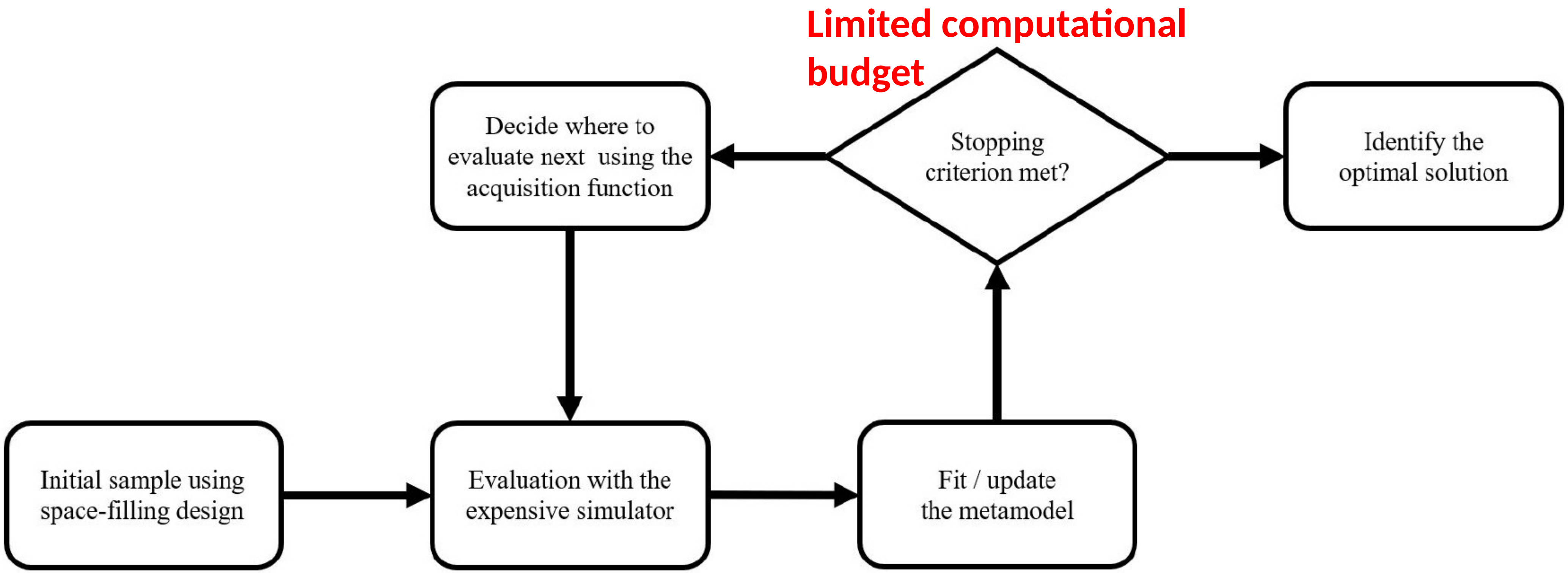
Optimization of the acquisition function

$$x_{t+1} = \arg\max \alpha(x),$$

where α is non-convex, multi-modal, **cheap to evaluate**, and often differentiable.

- Heuristic algorithms such as adaptive random search,
- Evolutionary methods,
- Local optimization approaches with multi-start,
- Discretizing the search space into a finite (but potentially large) set of candidate points.

General steps in Bayesian Optimization



Identification of the best solution

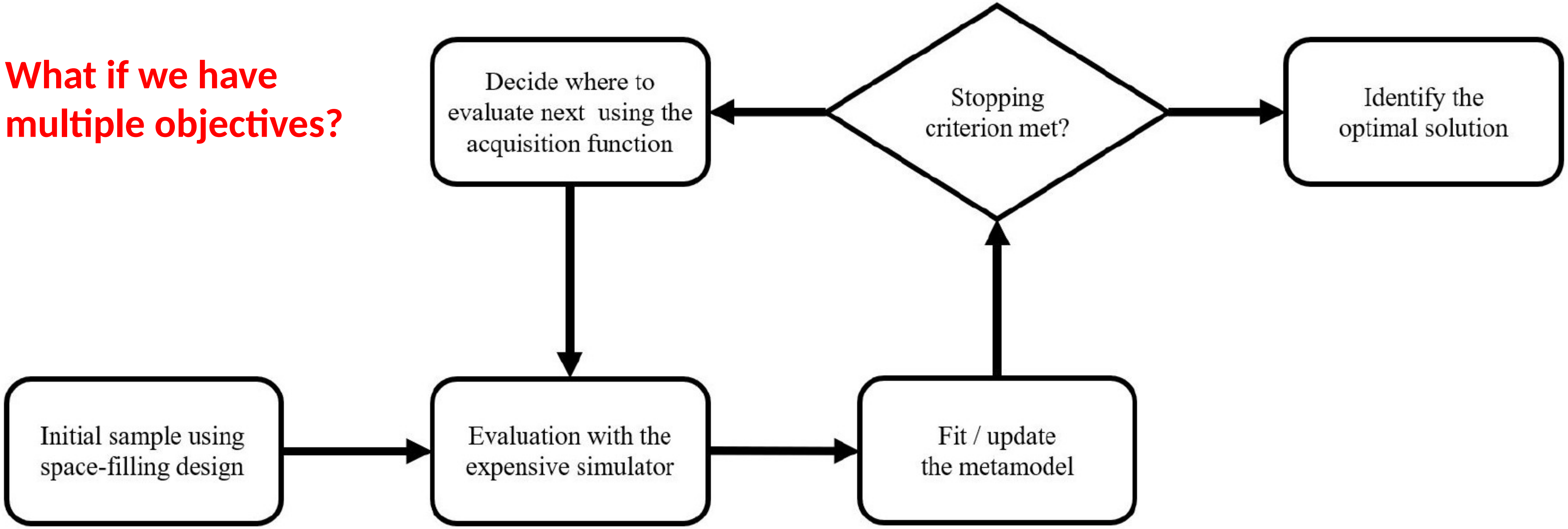
Once the evaluation budget (number of function calls or simulation runs) is depleted, the algorithm must select one solution to recommend.

Strategy	Description	Potential Pitfalls
Best observed point	Choose best solution among all evaluated points.	Algorithm may never *sample* the global optimum
Best predicted mean	Choose best predicted solution over the search space.	May **over trust model**
Resampling of candidates	Use few promising points; re-simulate to reduce noise.	Requires extra budget

Uncertainty ignored → reporting a single point without credible interval can mislead decision-makers.

General steps in Bayesian Optimization

What if we have multiple objectives?



Multi-objective Hyperparameter Optimization

The goal is to find the hyperparameter setting that gives the best validation performance:

$$\min_{x \in \mathcal{X}} f(x)$$

where:

- *x*: hyperparameters
- *f(x)*: validation error (expensive to evaluate)

$$\min_{x \in \mathcal{X}} \quad (f(x),\ g(x))$$

Where:

- *g(x)*: secondary performance measure (e.g.; training time or energy consumption)

Multi-objective optimization

Objectives are typically **conflicting,**

- Improving one degrades another (e.g., Cost vs Quality, Error vs Training time)

No single solution optimizes all objectives simultaneously,

- We seek trade-off solutions

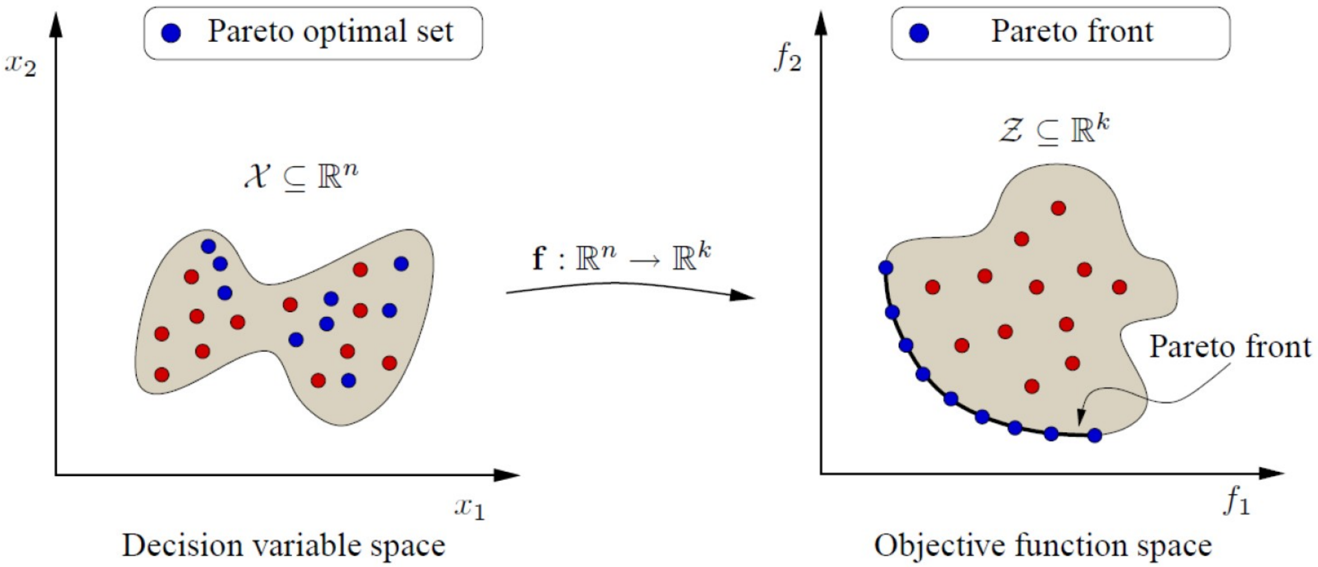
To formalize these trade-offs, we define a dominance relation between solutions:

Dominance Relation:

$$j \prec i \text{ iff } \forall l \in \{1, 2, \ldots, H\}, J_{jl} \leq J_{il} \text{ and}$$
$$\exists l \in \{1, 2, \ldots, H\}, J_{jl} < J_{il}.$$

Multi-objective optimization

A solution is called **non-dominated** if no other feasible solution dominates it; a.k.a. **Pareto optimal set.** The set of all such solutions constitutes the **Pareto front.** Without any preference information from the decision maker, the goal in MO optimization is to approximate the Pareto front**.**



Multi-objective optimization

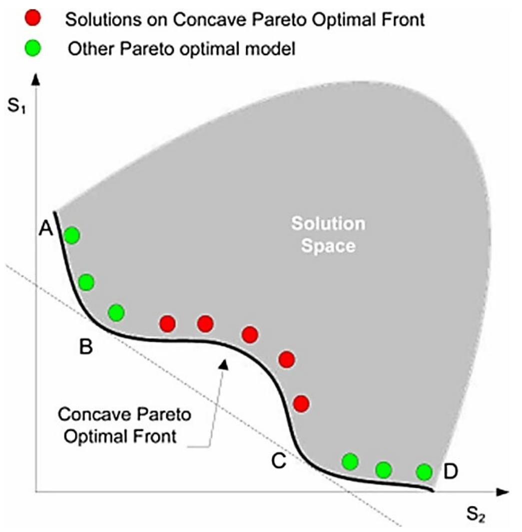
- Fit **one GPR model per objective**.
- A few model all objectives jointly using multivariate GPR, but this is less common.
- **Common approach:** scalarize the problem and fit a single model.

Metamodel

Weighted Sum:

$$S_{\mathrm{WS}}(\mathbf{f}(x)) = \sum_{m=1}^{M} w_m f_m(x), \quad w_m \geq 0, \ \sum_m w_m = 1.$$

Pros: simple, differentiable, easy to randomize weights in BO.
Con: recovers only convex regions of the Pareto front.



Augmented Tchebycheff:

$$S_{\mathrm{TCH}}(\mathbf{f}(x)) = \max_m w_m \left|f_m(x) - z_m^*\right| \quad \left(+ \rho \sum_m w_m |f_m(x) - z_m^*|\right),$$

Pro: handles non-convex fronts

Multi-objective optimization

Acquisition function

Single-objective acquisition functions:

Adapted for multi-objective problems by:

- Applying the function to each objective separately, or
- Using a scalarized combination of objectives.

Multi-objective acquisition functions:

Measure the contribution of new points to the Pareto front.

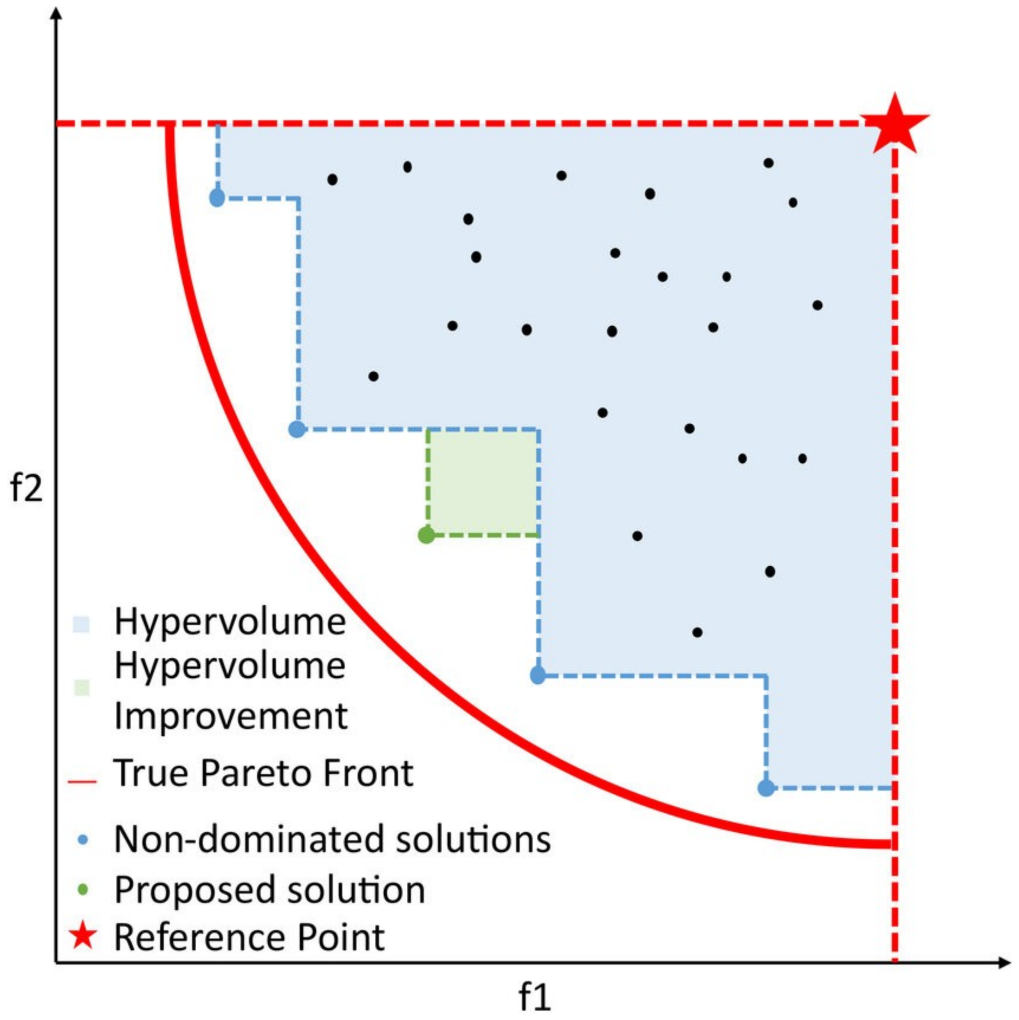
Hypervolume Improvement:

The hypervolume (HV) quantifies the volume of objective space dominated by the current Pareto set and bounded by a reference point *r*.

$$HV(\mathcal{P}) = \mathrm{Vol}\Big(\bigcup_{x \in \mathcal{P}} [f_1(x), r_1] \times \cdots \times [f_M(x), r_M]\Big) \quad \longrightarrow \quad \mathrm{HVI}(x) = \max\big(HV(\mathcal{P} \cup \{x\}) - HV(\mathcal{P}), 0\big).$$

Hypervolume improvement at a candidate x

Multi-objective optimization



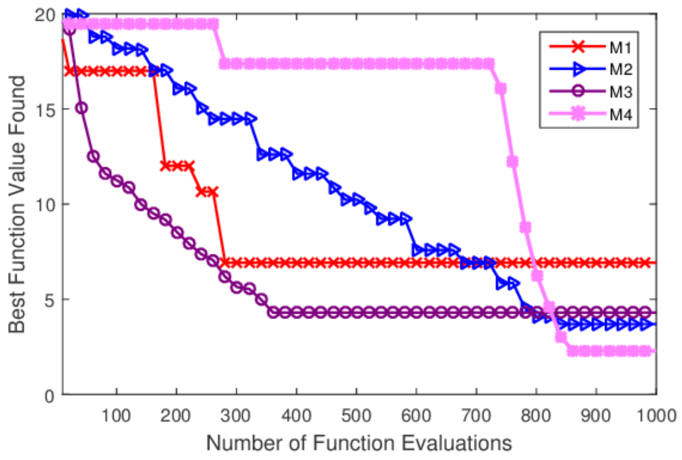
- HV measures the dominated region
  larger HV ⇒ better front (minimization).
- HVI(x) is the volume gain from adding x to the front.
- Expected Hypervolume Improvement (EHVI) integrates this gain under GP uncertainty.

Key Insights

When is Bayesian Optimization Useful?

- Each function evaluation is expensive (e.g., simulation, experiment, model training),
- The function is a black box; no gradients or analytical form,
- The search space is continuous and low- to medium-dimensional.

BO does not guarantee to find the global optimum.
Its goal is to discover a good solution with the fewest experiments.



True global optima are <u>unknown</u> in practice,

<u>Studies often report:</u>

- The best value found so far over iterations, and
- Convergence curves comparing different algorithms.

Break until 17:30h

Up next: Sustainable hyperparameter optimization

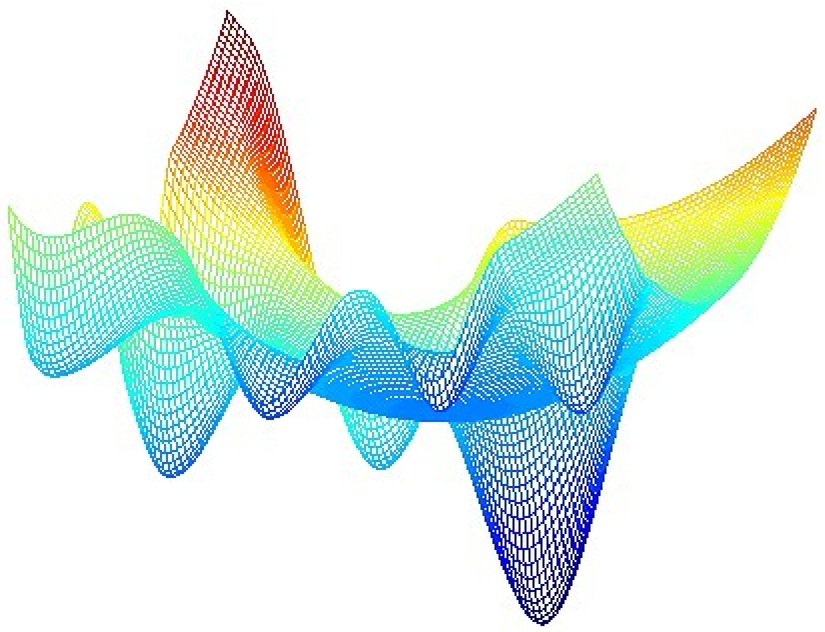
Overview

- ❑ **Part I: HPO (15:30-17:00)**

- **Problem definition**
- **Overview of HPO algorithms**
- **Introduction to Bayesian Optimization**
- **Main ideas in multi-objective BO**

- ❑ **Part II: Sustainable HPO (17:30-20:00)**

- **Short intro Sustainable AI**
- **Sustainable AI in practice**
- **Exercise**

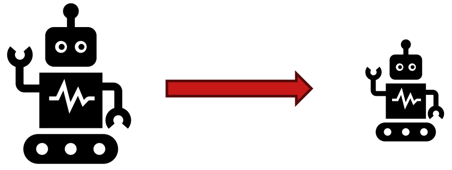


Sustainable AI

AI for Sustainability: using AI for renewable energy, biodiversity, e.g.

- pollution detection
- wind farm design
- animal face recognition

Sustainability of AI: reduce energy use (and other resources) of AI, e.g.



- Smaller models (pruning, quantization)
- Energy-efficient chips (quantum/neuromorphic/photonic/edge computing)
- Re-using data (meta-learning, 0-shot learning, transfer learning)
- Datacenter scheduling and load shifting
- Efficient hyperparameter optimization
- …

Sustainable AI as part of Sustainable IT

Use IT to reduce the environmental footprint of other sectors (mobility, etc.)

Reduce the environmental footprint of IT

Sustainable AI as part of responsible AI

Trustworthiness

Transparency

Privacy

Human-centric

Interpretability

Explainability

Fairness

Sustainability

Robustness

Why Sustainable AI?

BBC

Home News US Election Sport Business Innovation Culture Arts Travel Earth Video Live

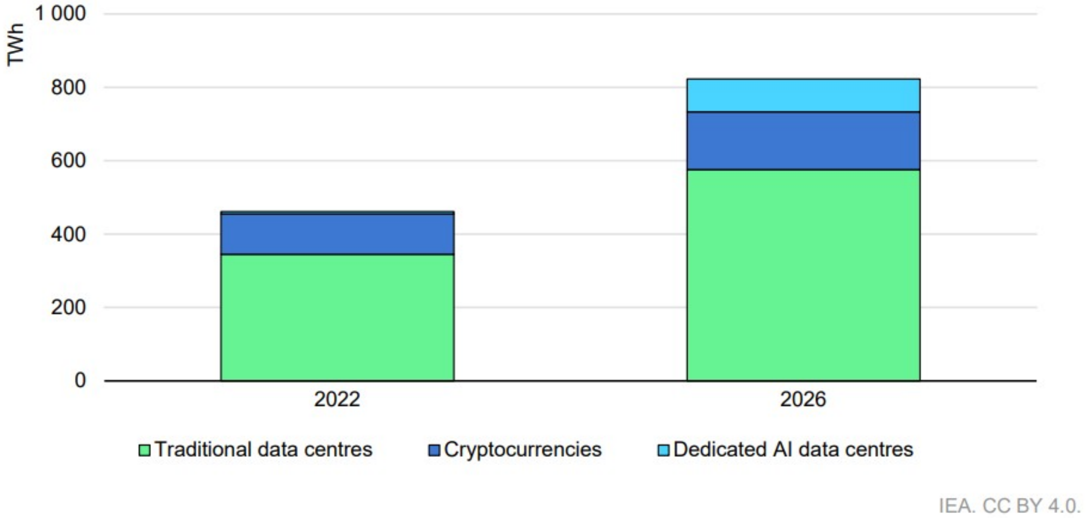
AI drives 48% increase in Google emissions

3 July 2024

Share Save

Imran Rahman-Jones
Technology reporter

Estimated electricity demand from traditional data centres, dedicated AI data centres and cryptocurrencies, 2022 and 2026, base case



Note: Data centre electricity demand excludes consumption from data network centres.

Ambitious startup plans to put AI data centers in offshore wind turbines

Why Microsoft And Amazon Are Turning To Nuclear Power For AI

How the AI data center boom is breathing new life into transforming dirty, old coal plants

Bloomberg

Extreme Heat, Drought Drive Opposition to AI Data Centers

For more than a year, Spain has been struggling with drought that has sent water levels in dams below historical averages, prompting local...

26 juil. 2023

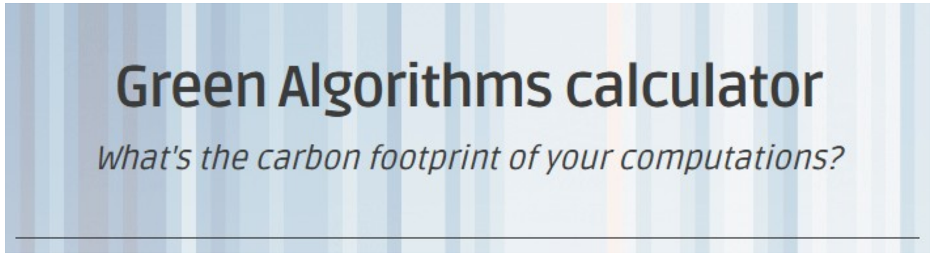
DW

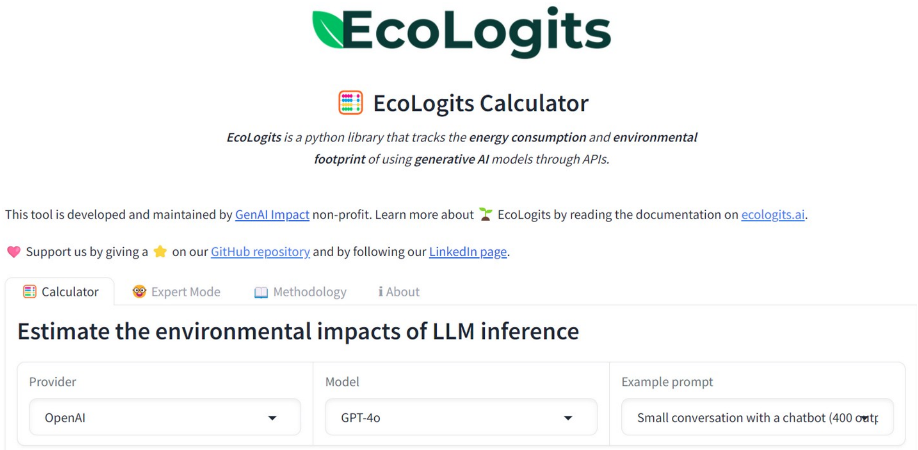
E-waste from AI computers could 'escalate beyond control'

Researchers predicting a thousand-fold increase in e-waste from AI computer servers by 2030 called for recycling strategies to reduce the...

28 oct. 2024

Sustainable AI measurement tools (not exhaustive!)





Tool	Link
Carbontracker	https://github.com/lfwa/carbontracker
Experiment impact tracker	https://github.com/Breakend/experiment-impact-tracker/tree/master)
Eco2AI	https://github.com/sb-ai-lab/Eco2AI
MLCO2	https://mlco2.github.io/impact/
CodeCarbon	http://codecarbon.io/
Green Algorithms	http://calculator.green-algorithms.org/
EcoLogits	https://huggingface.co/spaces/genai-impact/ecologits-calculator

Sustainable AI in practice

Let's address the Sustainability of AI in two ways:

- Efficient hyperparameter optimization using Bayesian optimization instead of grid search or manual tuning)
- Finding low-energy AI models

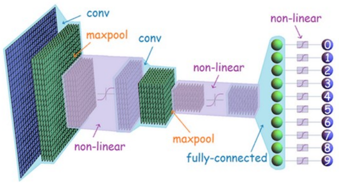
Sustainable AI in practice

You can find the tutorial code here:
https://github.com/ai-for-decision-making-tue/SustainableHPO_example → (scan)

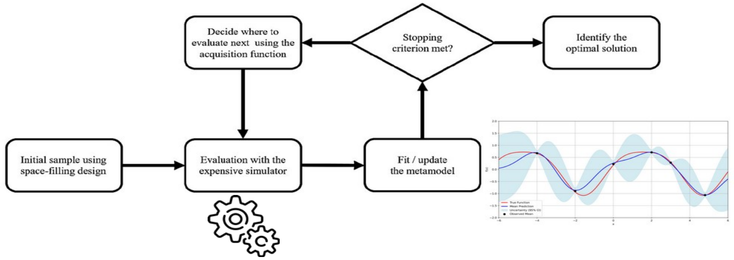
What are we going to do

Part 1

Train and evaluate a
convolutional neural network (CNN)



Optimize hyperparameters for accuracy



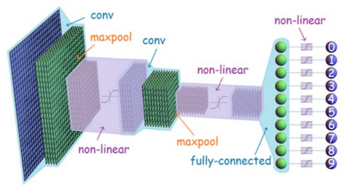
Sustainable AI in practice

You can find the tutorial code here:
https://github.com/ai-for-decision-making-tue/SustainableHPO_example → (scan)

What are we going to do

Part 2

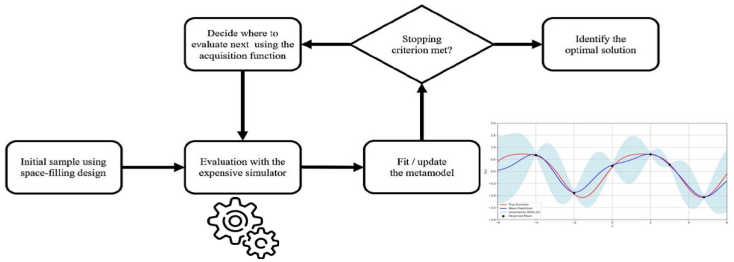
Train and evaluate a convolutional neural network (CNN)



Measure energy

Carbontracker

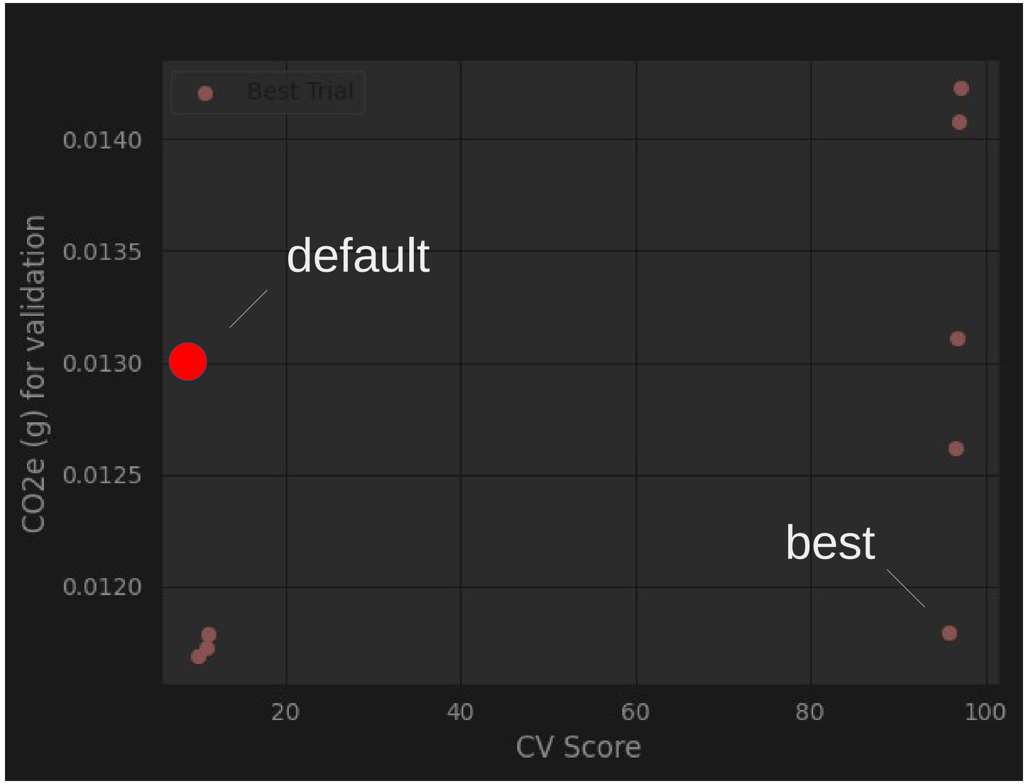
Optimize hyperparameters for both accuracy and energy



Sustainable AI in practice

You can find the tutorial code here:
https://github.com/ai-for-decision-making-tue/SustainableHPO_example → (scan)

What the end result will be (spoilers):



- Even for a simple example with very limited resources:
- **Increase accuracy by almost 10x** compared to default
- **AND**
- **Reduce CO2 emissions by 10%** compared to default

- Almost 20% CO2 reduction compared to only focusing on accuracy
- Almost 10x as accurate compared to only focusing on CO2