# Bayesian Optimization

## Recent Advances

Joel Paulson
The H.C. “Slip” Slider Assistant Professor,
Department of Chemical and Biomolecular Engineering,
The Ohio State University

Sargent Centre Summer School on Bayesian Optimization, 2024

For copies of slides & code, see
https://github.com/joelpaulson/Sargent_Centre_BO_Summer_School_2024

# Recall: Bird's-eye View of Bayesian Optimization

while {budget not exhausted}

Fit a Bayesian machine learning model (usually Gaussian process regression) to observations {x, f(x)}

First talk

Find x that maximizes acquisition(x, posterior)

Second talk

Sample x & then observe f(x)

end

**This talk will focus on ways to modify the traditional setup of Bayesian optimization**

More Information

# Outline

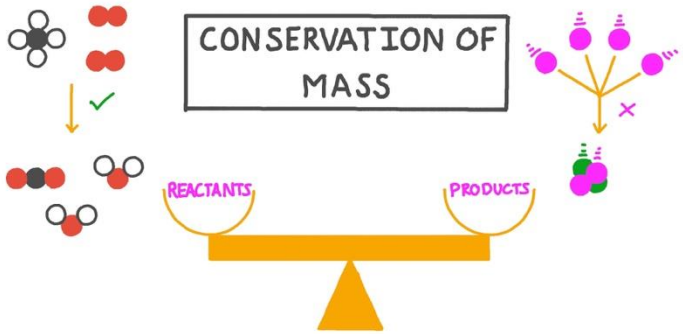
- What is standard Bayesian optimization missing?
  - Strong priors, known structure, safety considerations, uncertainty
- Beyond sequential & single objective problems
  - Parallel evaluations (synchronous & asynchronous), multi-objective optimization
- Beyond the black-box problem structure
  - Composites, function networks, multi-fidelity representations
- Beyond nominal optimization
  - Adversarial uncertainty, flexibility analysis, safety
- What is next?
  - Meta learning, preference learning, new ways to optimize acquisition, planning

# Outline

- What is standard Bayesian optimization missing?
  - Strong priors, known structure, safety considerations, uncertainty
- Beyond sequential & single objective problems
  - Parallel evaluations (synchronous & asynchronous), multi-objective optimization
- Beyond the black-box problem structure
  - Composites, function networks, multi-fidelity representations
- Beyond nominal optimization
  - Adversarial uncertainty, flexibility analysis, safety
- What is next?
  - Meta learning, preference learning, new ways to optimize acquisition, planning

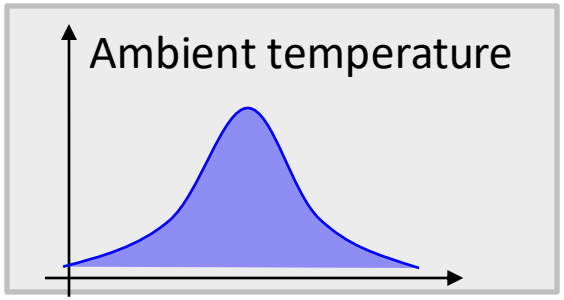
# What is standard Bayesian Optimization missing?

1. Strong prior information
   - Physical laws and bounds
   - Partially known structure



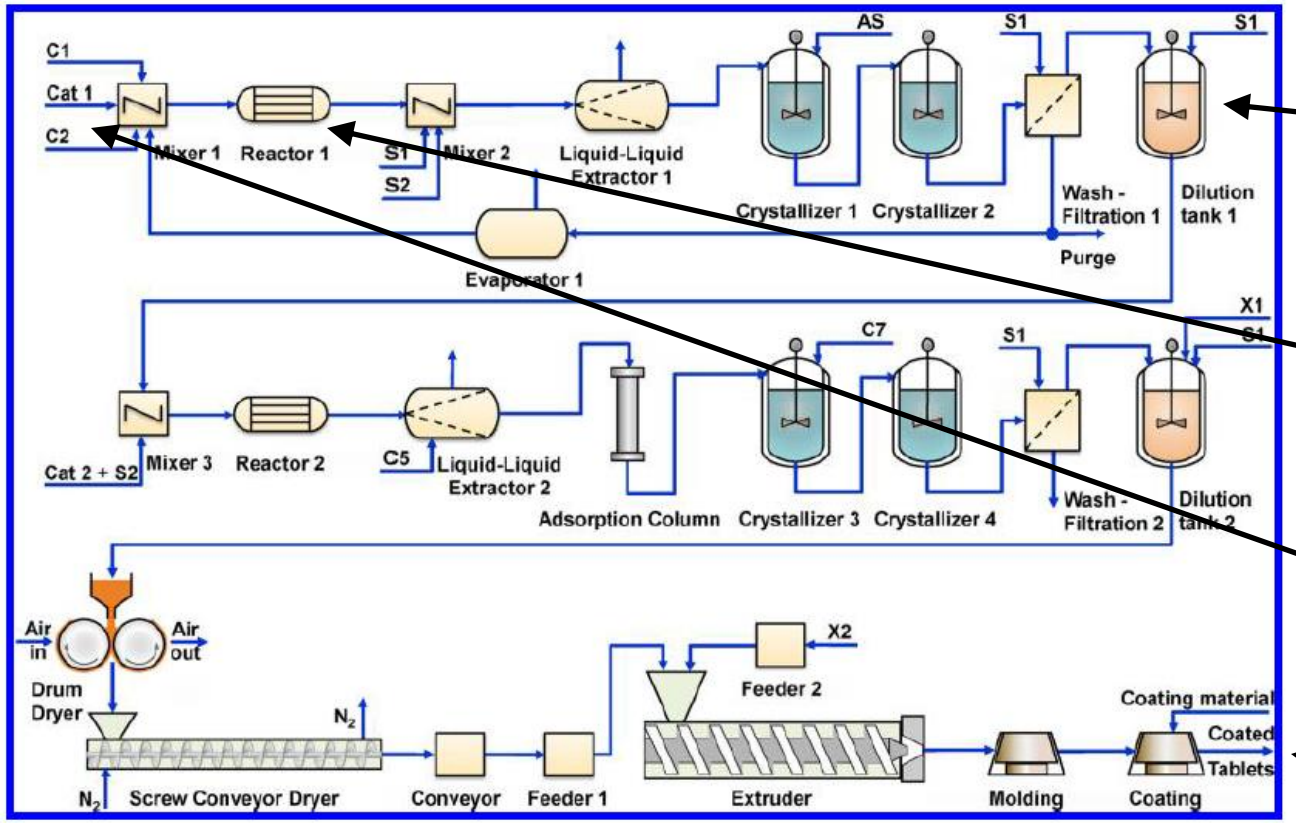

2. Safety considerations
   - Constraints on system behavior
   - Limitations on implementation

3. Large uncertainty in practice
   - Implementation errors
   - Perturbations in environment



# Example: Optimization of drug manufacturing process

- Process depicted below makes drug product from reactive precursors
- Multiple stages, each with its own set of design variables
- Goal is to maximize drug production under safety and quality constraints




Know about physics of reaction, crystallization, dilution; explicit description of overall flowsheet

Know certain restrictions exist on how we operate reactors to ensure equipment is safe

Can have large variation in feed quality (depending on vendor)

Have additional constraints on the quality of produced drug

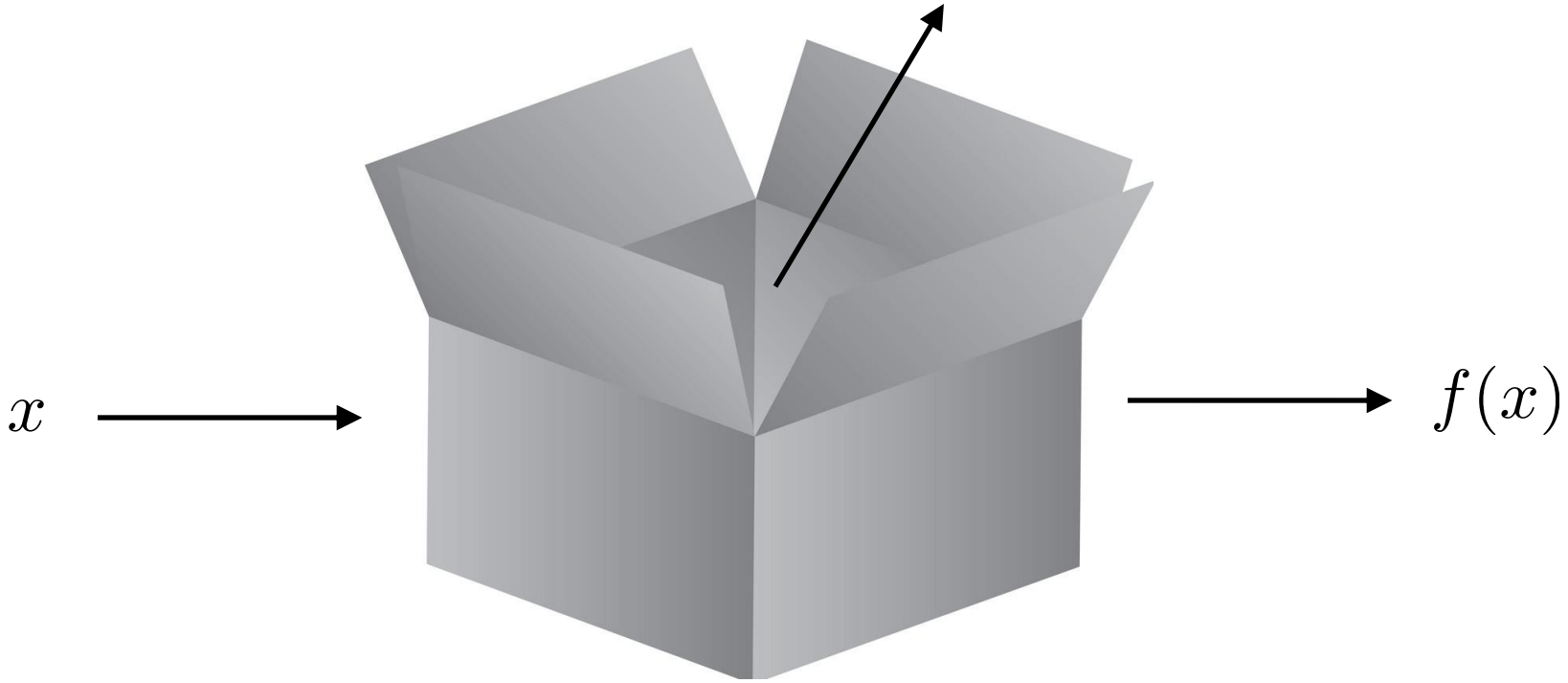
[Norvartis-MIT Center for Continuous Manufacturing]

# Can do better by peeking <u>inside</u> the box

"one should avoid learning what they already know"

(really just any additional information about problem and/or simulator)

**"knowledge" or "physics"**




*sometimes referred to as grey-box (or hybrid) optimization

Recent tutorial paper on grey-box BayesOpt: [Astudillo and Frazier, *Winter Simulation Conference*, 2021]

# The Principles of Bayesian Optimization are Extremely Flexible

- Let's NOT simply assume that our problem is of the form:

$$x^* = \text{argmax}_{x \in \Omega} f(x), \qquad f(x) \sim \mathcal{GP}(\mu_0, k_0)$$

- Even though a wide range of problems can be treated in this way, this may not be the best representation in practice

- We can revisit the assumed problem statement, similarly to what we did when analyzing the case of black-box constraints
  - We need to carefully think about the real-world problem setting
  - Silver lining…many of the principles we already saw can be repeatedly used

# The Principles of Bayesian Optimization are Extremely Flexible

Elicit some **prior distribution** on the functions of interest

while {budget not exhausted}

Find **information source** whose **value of information** is the largest in the set of options

Query corresponding **information source**

Update the **posterior distribution** of the functions

end

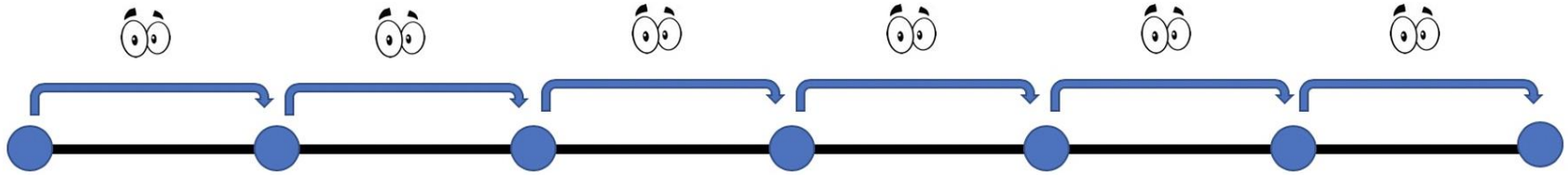
# Outline

- What is standard Bayesian optimization missing?
  - Strong priors, known structure, safety considerations, uncertainty
- Beyond sequential & single objective problems
  - Parallel evaluations (synchronous & asynchronous), multi-objective optimization
- Beyond the black-box problem structure
  - Composites, function networks, multi-fidelity representations
- Beyond nominal optimization
  - Adversarial uncertainty, flexibility analysis, safety
- What is next?
  - Meta learning, preference learning, new ways to optimize acquisition, planning

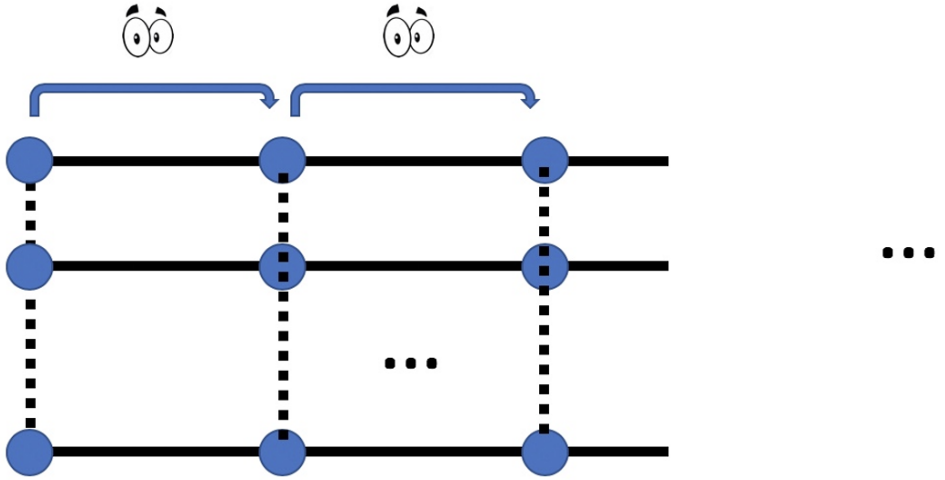
# Parallel Bayesian Optimization

# Sequential versus Parallel Evaluations

Standard (sequential) Bayesian optimization



Parallel (or batch) Bayesian optimization



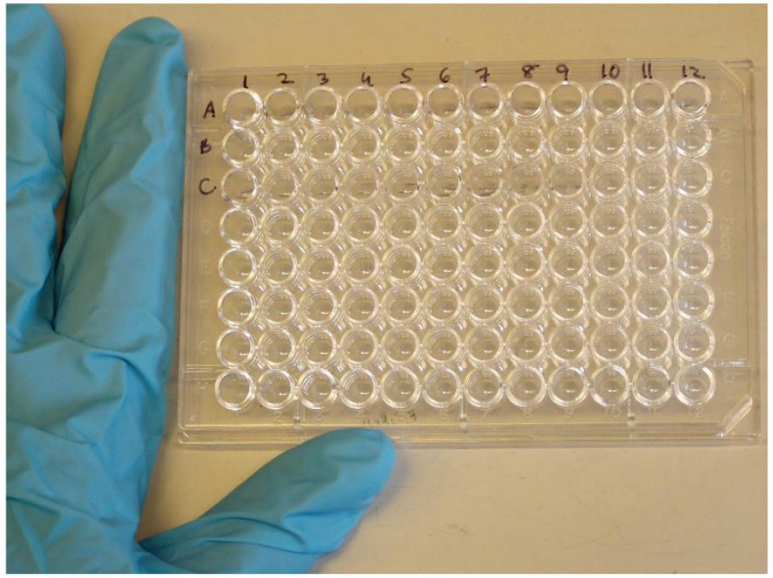
...

Cost of $f(x_n)$ ≈ Cost of $\{f(x_n^{(1)}), \ldots, f(x_n^{(q)})\}$
(potentially huge reduction in effective cost when # of batches q is large)

# Parallel Bayesian Optimization

- **Goal:** Previously evaluated $x_1, \ldots, x_n$ and observed $f(x_1), \ldots, f(x_n)$, can now collect new observations on a batch of $q$ points $\{x^{(1)}, \ldots, x^{(q)}\}$. Want to optimally design these points in a *simultaneous* fashion

- Many applications including optimizing computer code on multiple cores, well plates in biological experiments, A/B tests on the web, etc.



# Parallel Bayesian Optimization

- **Goal:** Previously evaluated $x_1, \ldots, x_n$ and observed $f(x_1), \ldots, f(x_n)$, can now collect new observations on a batch of $q$ points $\{x^{(1)}, \ldots, x^{(q)}\}$. Want to optimally design these points in a *simultaneous* fashion

- The best value that has been observed so far is (assuming goal is max):

$$f_n^* = \max\{f(x_1), \ldots, f(x_n)\}$$

- If we measure the new batch of points and then stop, the **expected value** of our new solution can be written as

$$\mathbb{E}_n\left[\max\{f_n^*, \max_{i=1,\ldots,q} f(x^{(i)})\}\right]$$

# Parallel Bayesian Optimization

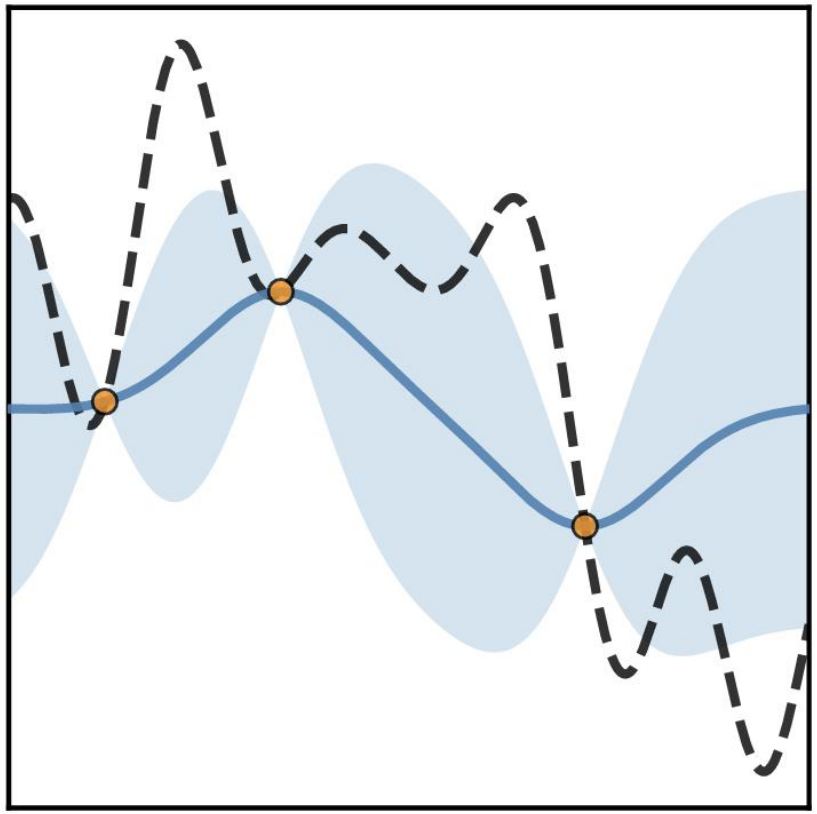
- The **parallel expected improvement** can then be expressed as

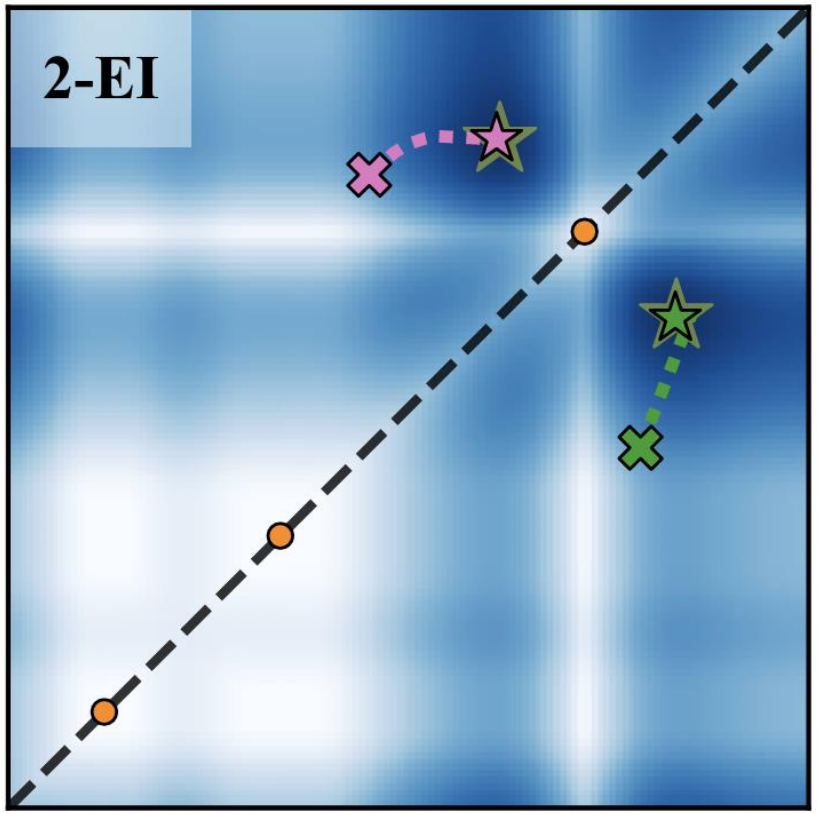
$$\mathrm{qEI}_n\left(x^{(1)}, \ldots, x^{(q)}\right) = \mathbb{E}_n\left[\max\{f_n^*, \max_{i=1,\ldots,q} f(x^{(i)})\}\right] - f_n^* = \mathbb{E}_n\left[\left(\max_{i=1,\ldots,q} f(x^{(i)}) - f_n^*\right)^+\right]$$

- For $q = 1$, we recover the closed-form expression of standard EI
- For $q = 2$, Ginsbourger et al., 2007 gives expression using bivariate normal CDFs
- For $q > 2$, need to resort to some type of Monte Carlo estimation scheme
  - Recent paper by Ament et al., *NeurIPS*, 2023 discusses an improved "log" formulation that ends up being much easier to optimize in practice

- Key point is that we are *jointly* choosing the locations of all the $q$ samples $\mathrm{argmax}_{x^{(1)},\ldots,x^{(q)}}\ \mathrm{qEI}_n\left(x^{(1)}, \ldots, x^{(q)}\right)$ → BoTorch is good at handling batching!

# Illustration of the optimization surface for q-EI with q = 2

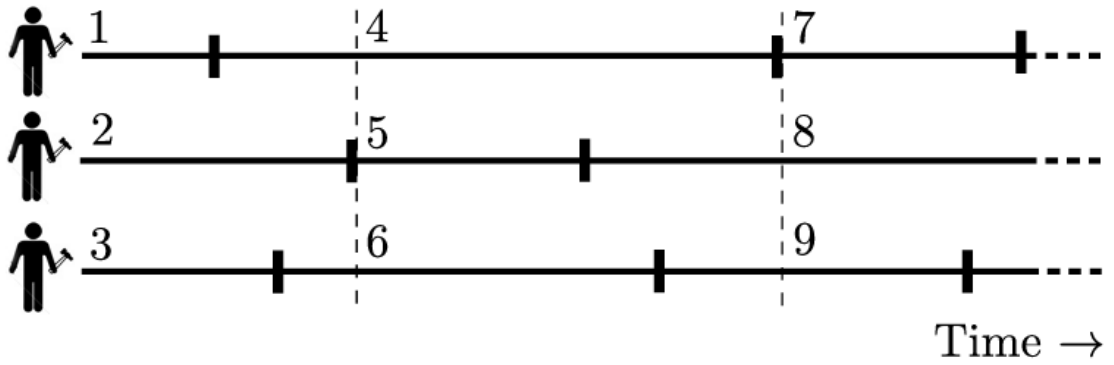




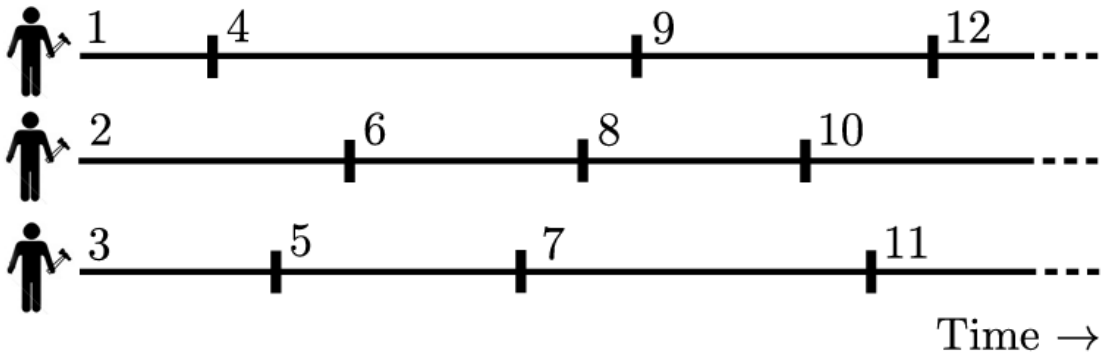

[Wilson et al. , The reparameterization trick for acquisition functions, 2017]

# What about asynchronous evaluations?

Synchronous settings with 3 workers




Asynchronous settings with 3 workers




[Kandasamy et al. , *ICAIS*, 2018]

# Thompson sampling nice way to deal with asynchronicity

**Algorithm:** Asynchronous Thompson sampling

**Require:** Prior GP $\mathcal{GP}(\mathbf{0}, \kappa)$.

1: $\mathcal{D}_1 \leftarrow \varnothing, \quad \mathcal{GP}_1 \leftarrow \mathcal{GP}(\mathbf{0}, \kappa)$.
2: **for** $j = 1, 2, \ldots$ **do**
3: Wait for a worker to finish.
4: $\mathcal{D}_j \leftarrow \mathcal{D}_{j-1} \cup \{(x', y')\}$ where $(x', y')$ are the worker's previous query/observation.
5: Compute posterior $\mathcal{GP}_j = \mathcal{GP}(\mu_{\mathcal{D}_j}, \kappa_{\mathcal{D}_j})$.
6: Sample $g \sim \mathcal{GP}_j, \quad x_j \leftarrow \operatorname{argmax} g(x)$.
7: Re-deploy worker to evaluate $f$ at $x_j$.
8: **end for**

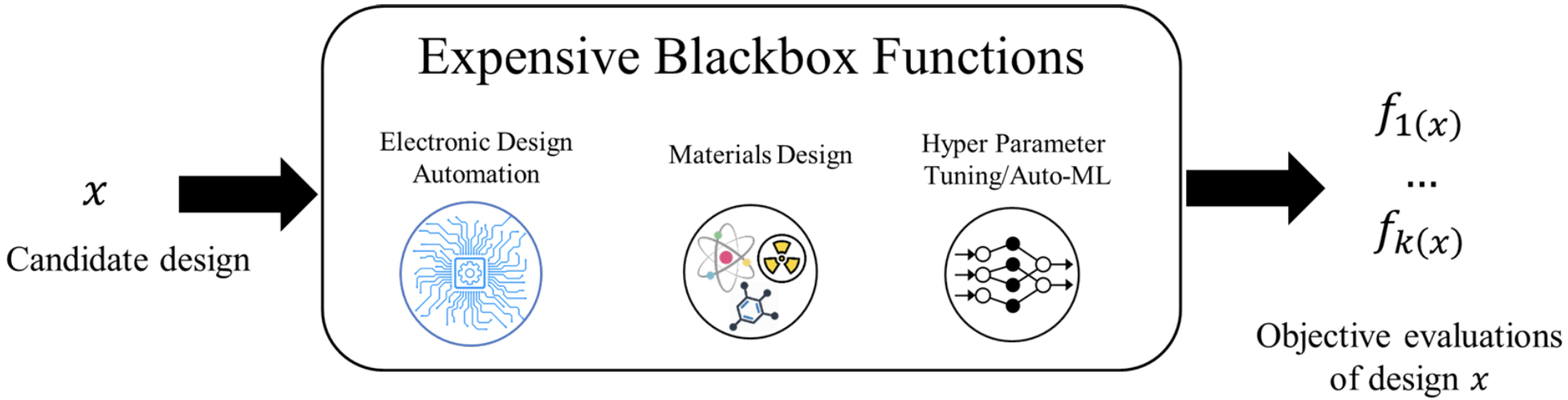
New evaluation can trigger as soon as worker is free

TS drawn independently, always accounts for all available information

[Kandasamy et al. , *ICAIS*, 2018]

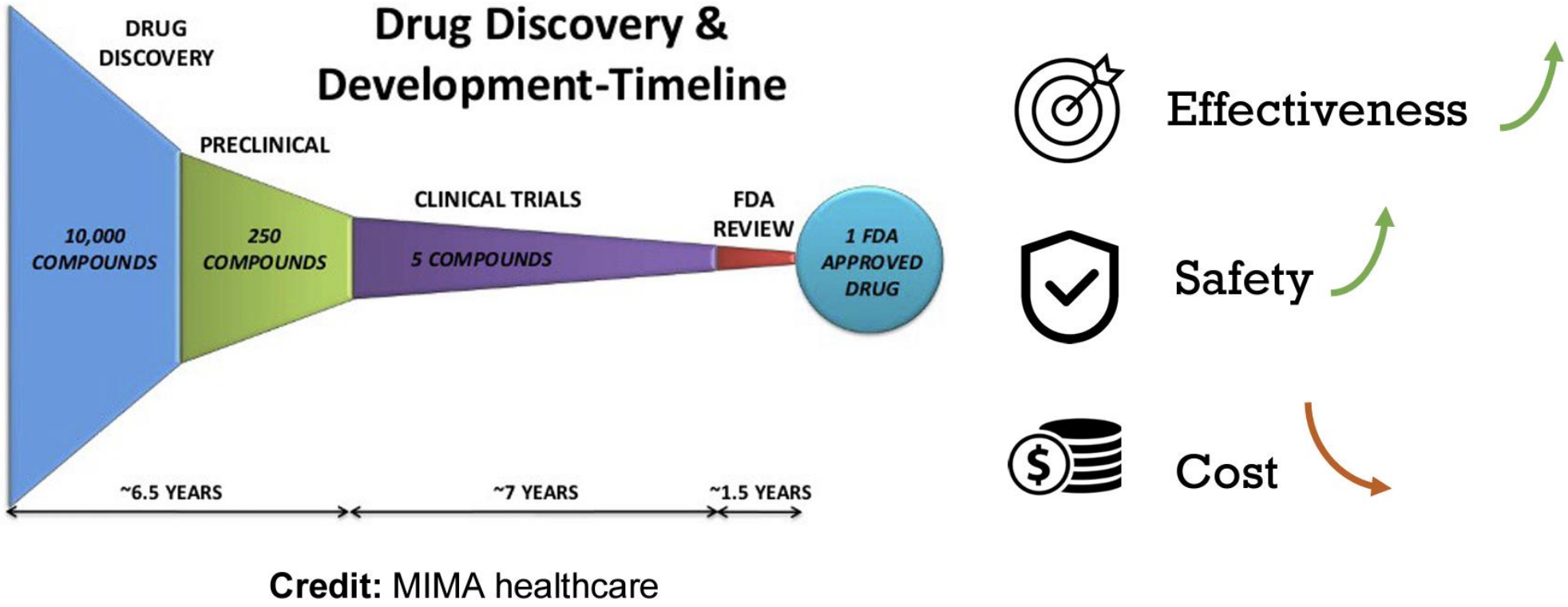
# Multi-Objective Bayesian Optimization

# Single versus Multi-Objective Evaluations




- **Goal:** find the designs that optimally tradeoff between different objectives using as little resources as possible
  - Objectives very often **conflict** with one another

# Example #1: Drug/Vaccine Design



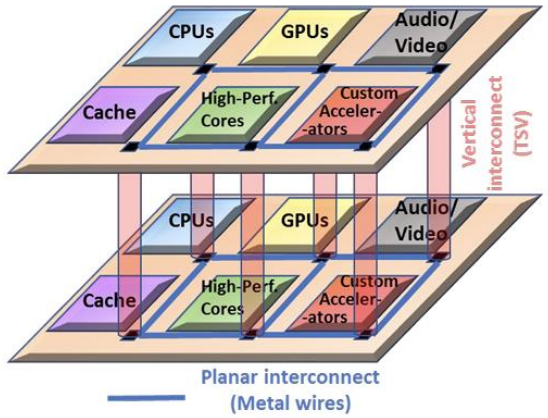

**Credit:** MIMA healthcare

- Accelerate the discovery of safe, effective, and low-cost designs

[Adapted from Aryan, Belakaria, and Doppa, Half-day tutorial, *AAAI*, 2022]

# Example #2: Hardware Design for Datacenters

**High-performance and Energy-efficient manycore chips**




## America's Data Centers Are Wasting Huge Amounts of Energy

By 2020, data centers are projected to consume roughly 140 billion kilowatt-hours annually, costing American businesses $13 billion annually in electricity bills and emitting nearly 150 million metric tons of carbon pollution

Report from Natural Resources Defense Council:.
https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IB.pdf

Performance ↑

Reliability ↑

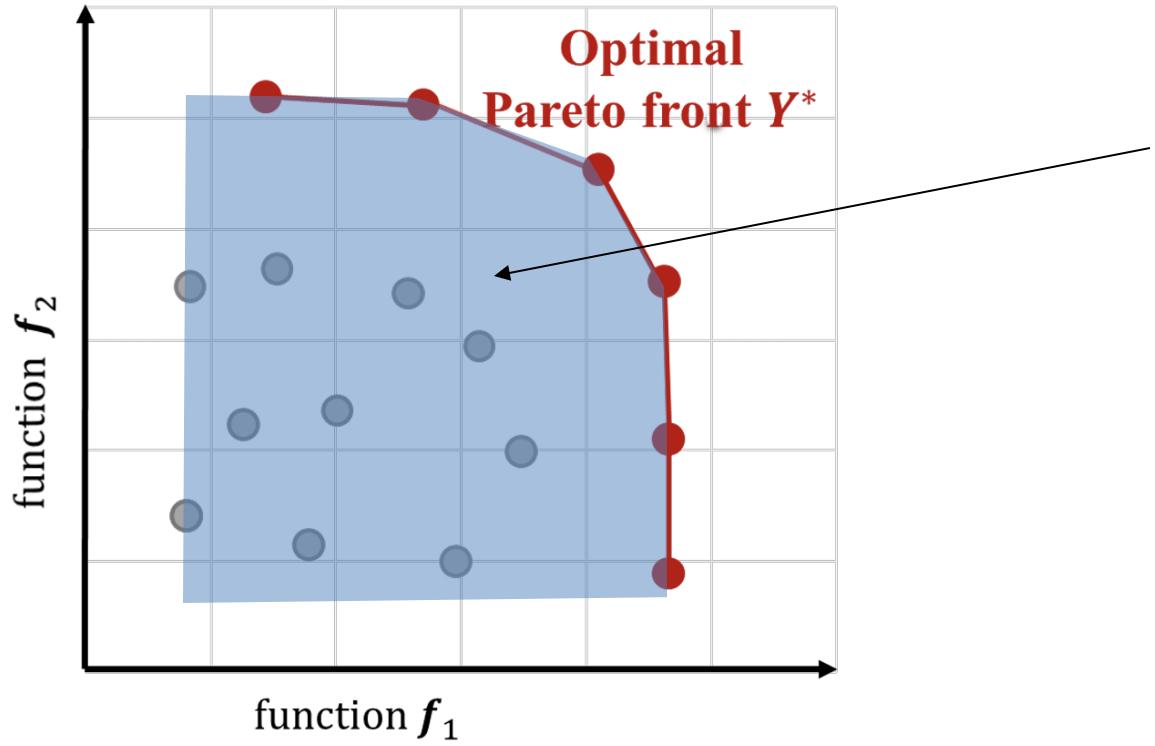
Power ↓

[Adapted from Aryan, Belakaria, and Doppa, Half-day tutorial, *AAAI*, 2022]

# What do we care about in multi-objective optimization?

The Pareto front & optimal set

- Set of inputs with optimal tradeoffs called the Pareto optimal set $X^*$
- Corresponding set of function values called the Pareto front $Y^*$




The "hypervolume" of the Pareto front provides some measure of quality
→ The bigger, the better

Huge amount of recent work on multi-objective BO…let's take a similar approach to what we have in the past (what would you do?)

[Adapted from Aryan, Belakaria, and Doppa, Half-day tutorial, *AAAI*, 2022]

# Expected Hypervolume Improvement (EHVI)

- Let $\mathrm{HV}(\mathcal{P}, \boldsymbol{r})$ be the hypervolume indicator of a finite approximation of the Pareto set $\mathcal{P}$ and a reference point $\boldsymbol{r} \in \mathbb{R}^K$ (bounds $\mathcal{P}$ from below)

- If we measure a new set of objective values $\boldsymbol{f}(\mathcal{X}_{\mathrm{cand}})$ at candidate points $\mathcal{X}_{\mathrm{cand}}$, then our **expected improvement** in the hypervolume is given by

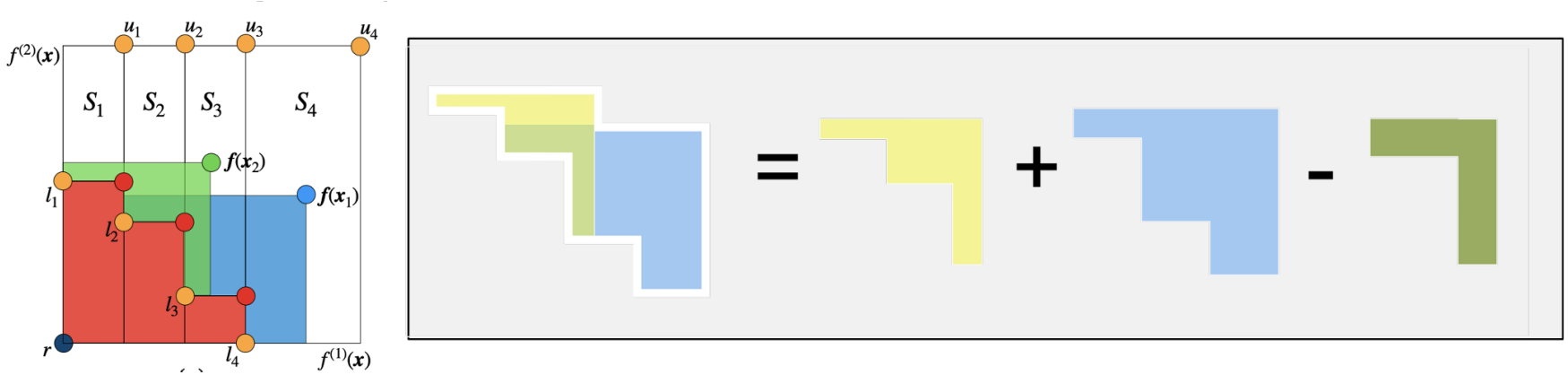
$$\mathrm{EHVI}(\mathcal{X}_{\mathrm{cand}}|\mathcal{D}) = \mathbb{E}[\mathrm{HV}(\mathcal{P} \cup \boldsymbol{f}(\mathcal{X}_{\mathrm{cand}}), \boldsymbol{r}) - \mathrm{HV}(\mathcal{P}, \boldsymbol{r})|\mathcal{D}]$$

Where the expectation is taken over the posterior $p(\boldsymbol{f}|\mathcal{D})$

  – Typically, use independent GPs for each objective function

# Example: Differentiable qEHVI [Daulton et al., 2020]

- Parallel form of EHVI using inclusion-exclusion principle




- Development of gradient estimator using Monte Carlo combined with the reparametrization trick
  - $y \sim \mathcal{N}(\mu, \Sigma)$ equal to $\mu + Lz$ where $z \sim \mathcal{N}(0\ I)$ and $\Sigma = LL^{\mathsf{T}}$

# Outline

- What is standard Bayesian optimization missing?
  - Strong priors, known structure, safety considerations, uncertainty
- Beyond sequential & single objective problems
  - Parallel evaluations (synchronous & asynchronous), multi-objective optimization
- Beyond the black-box problem structure
  - Composites, function networks, multi-fidelity representations
- Beyond nominal optimization
  - Adversarial uncertainty, flexibility analysis, safety
- What is next?
  - Meta learning, preference learning, new ways to optimize acquisition, planning

# Composite Functions

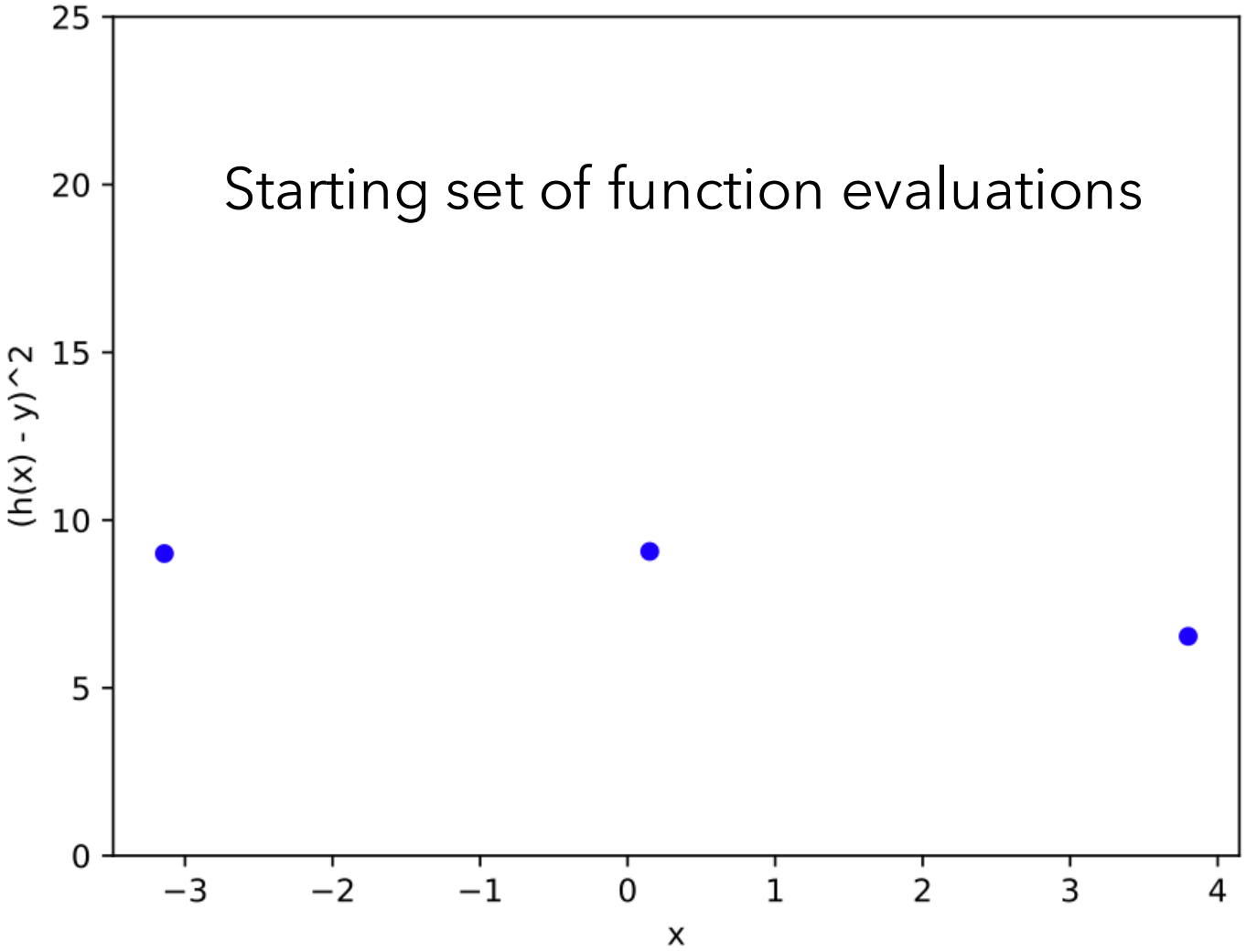
# Motivation: How to solve a calibration problem?

- Assume that we have the following variable declarations:
  - $x$ is a parameter of a black-box simulator (e.g., density functional theory)
  - $h(x)$ is the simulator's prediction given $x$
  - $y$ is our observed data that we would like our simulator to match

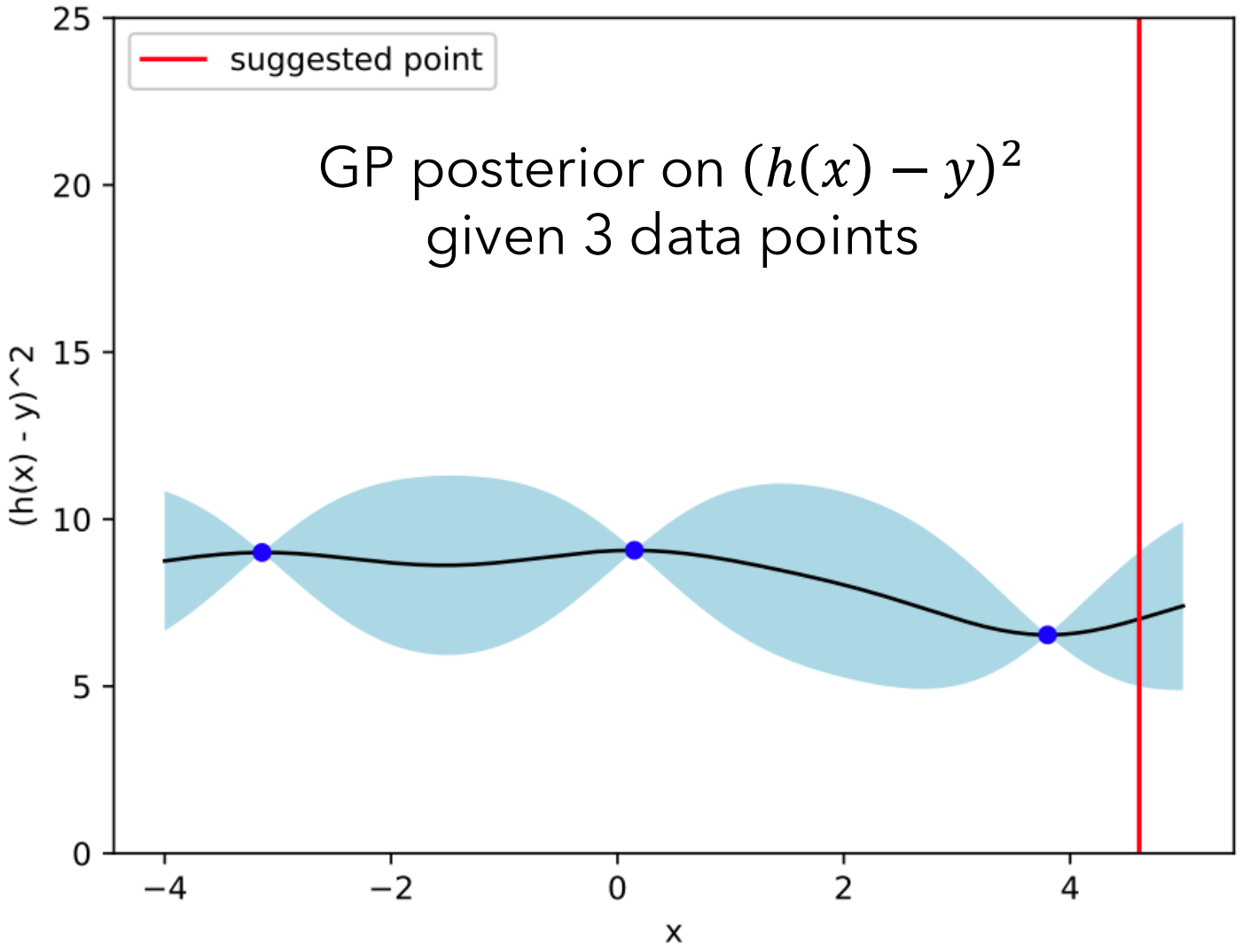
- To calibrate our simulation model, we want to solve

$$\min_x \ (h(x) - y)^2$$

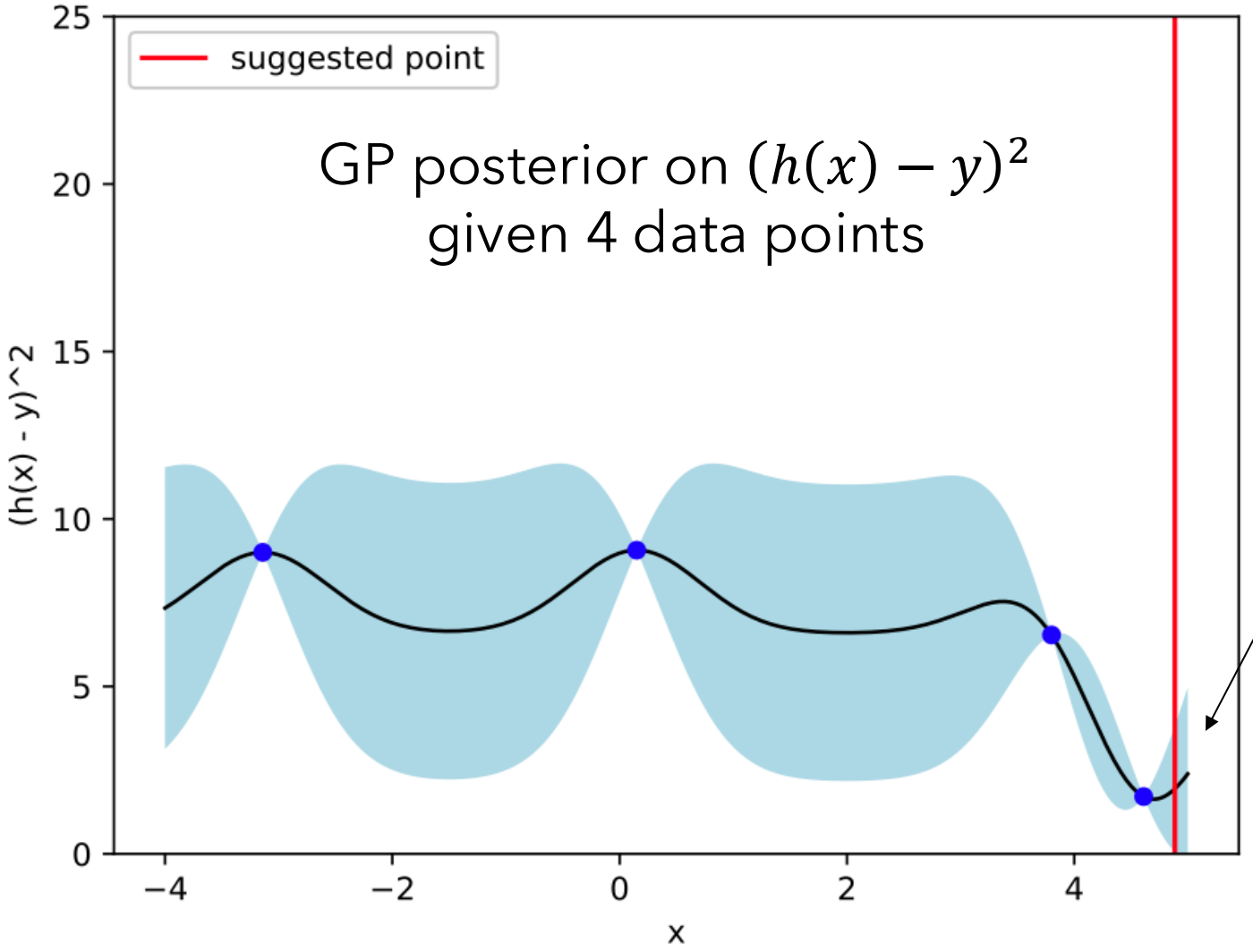
# Let's Solve Example using Standard Bayesian Optimization



# Let's Solve Example using Standard Bayesian Optimization

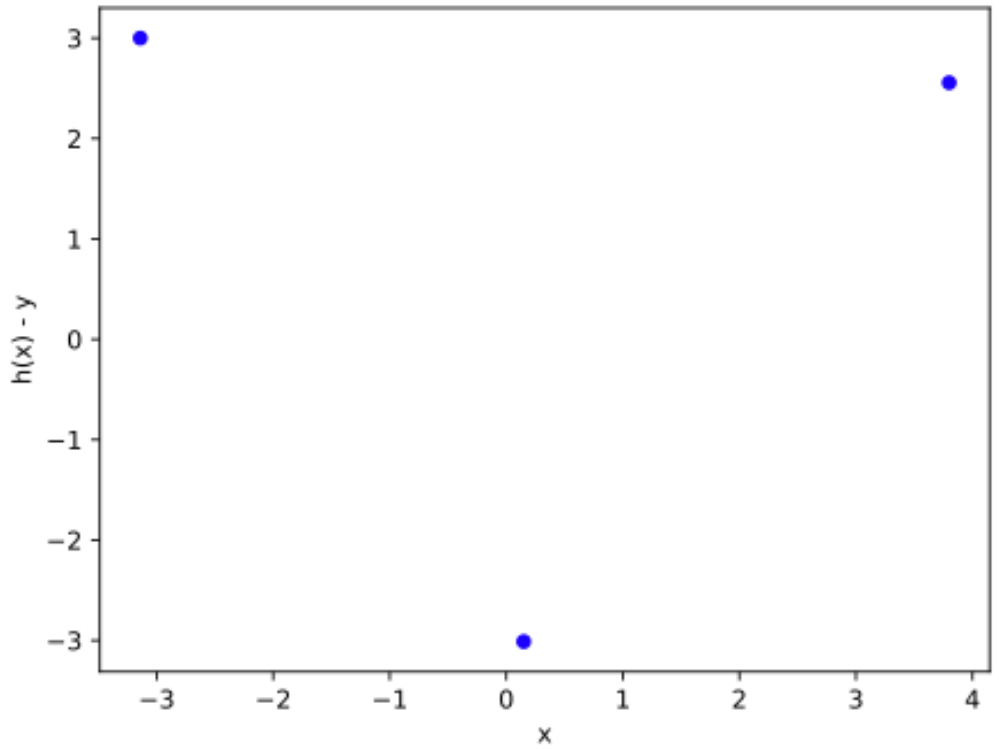


# Let's Solve Example using Standard Bayesian Optimization



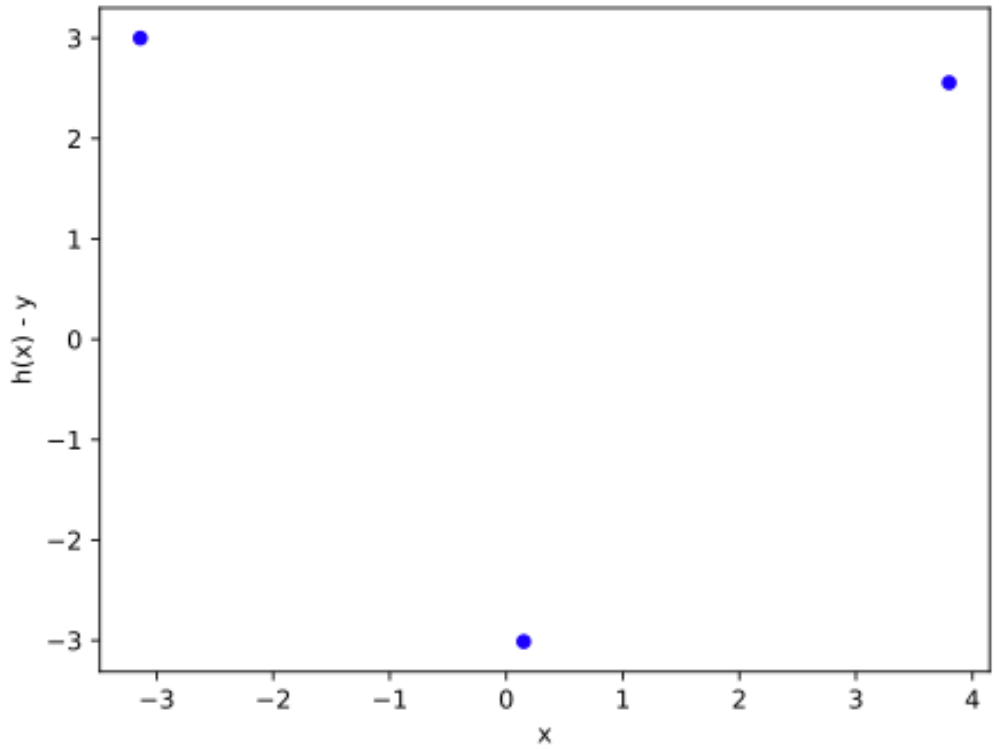

Notice how BO continues to sample in the right corner of the function

# Now, let's solve same problem by better leveraging the structure of the objective function $f(x)$

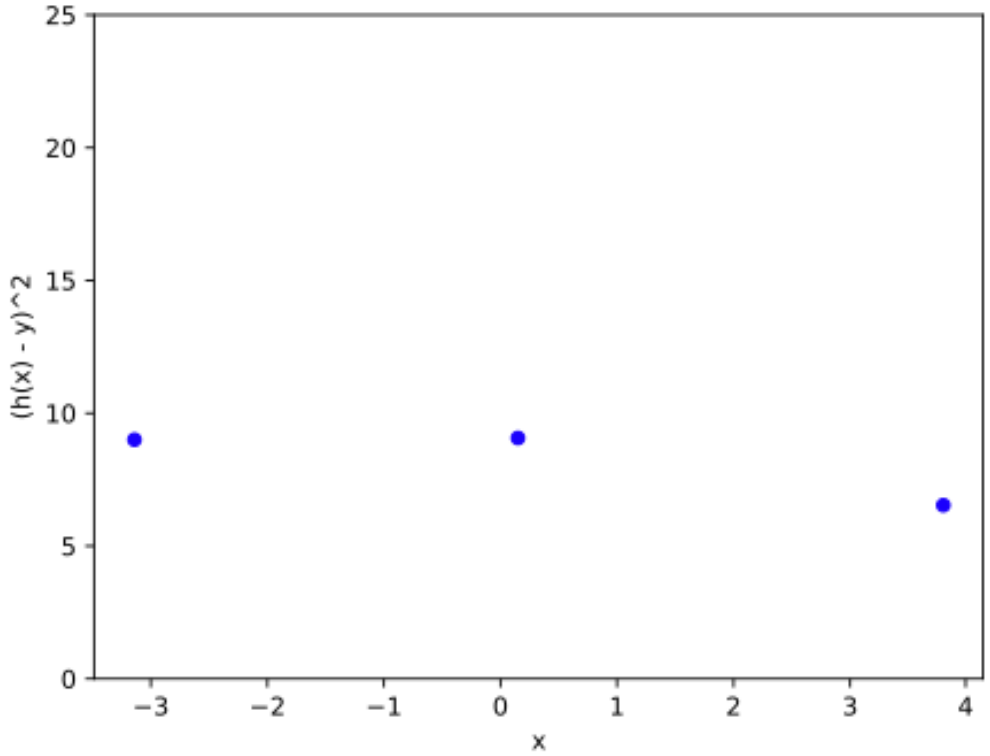



Evaluations of $h(x) - y$

# Now, let's solve same problem by better leveraging the structure of the objective function $f(x)$

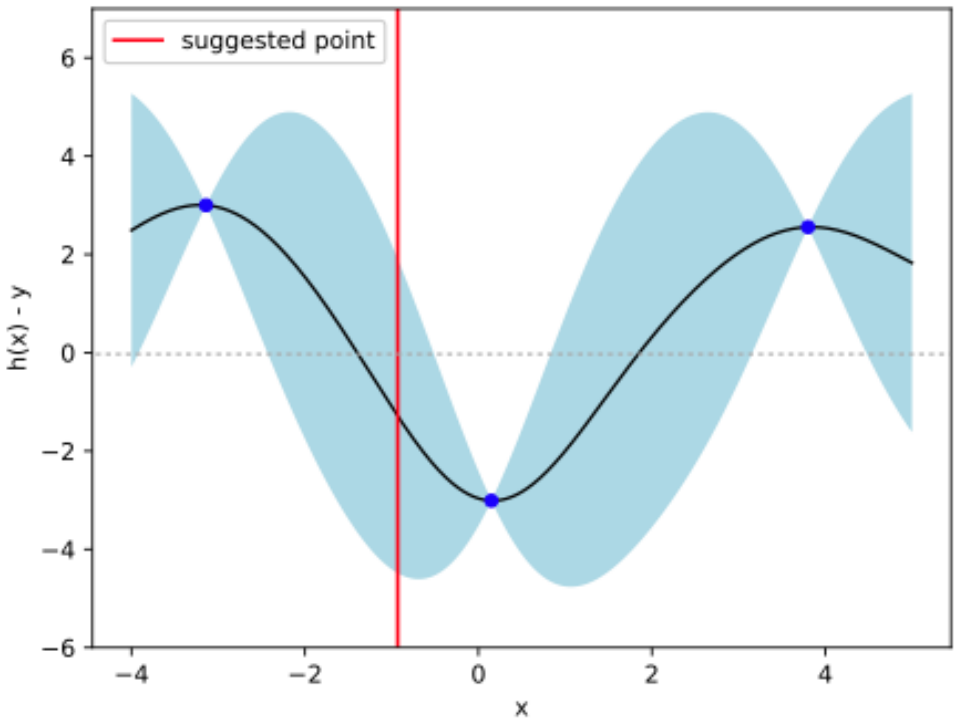



Evaluations of $h(x) - y$

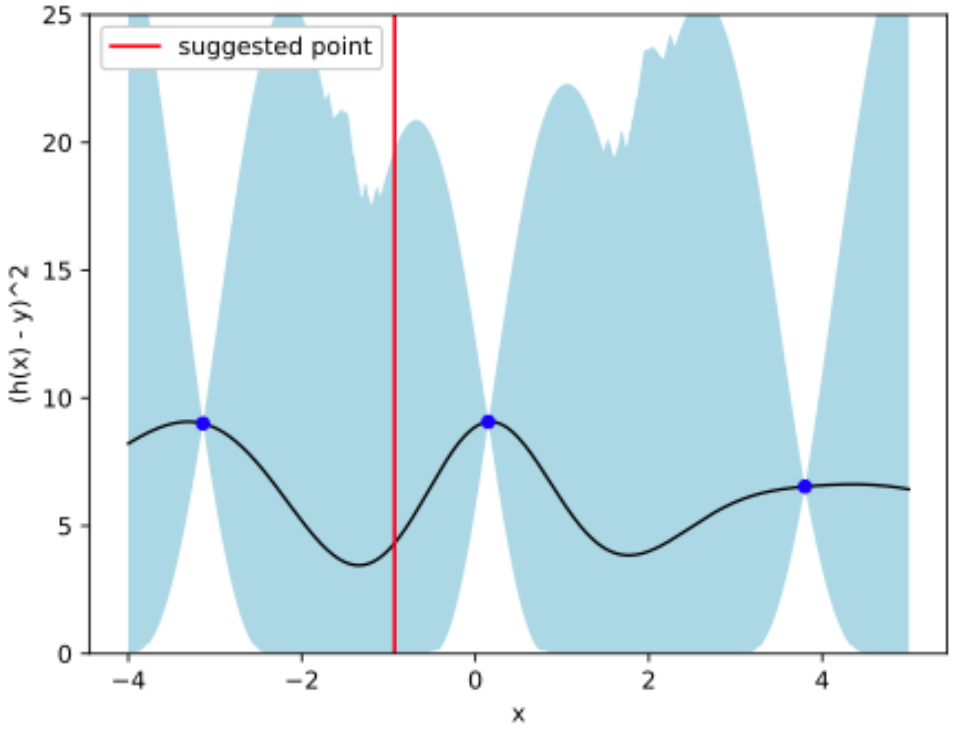



Evaluations of $(h(x) - y)^2$

# Now, let's solve same problem by better leveraging the structure of the objective function $f(x)$



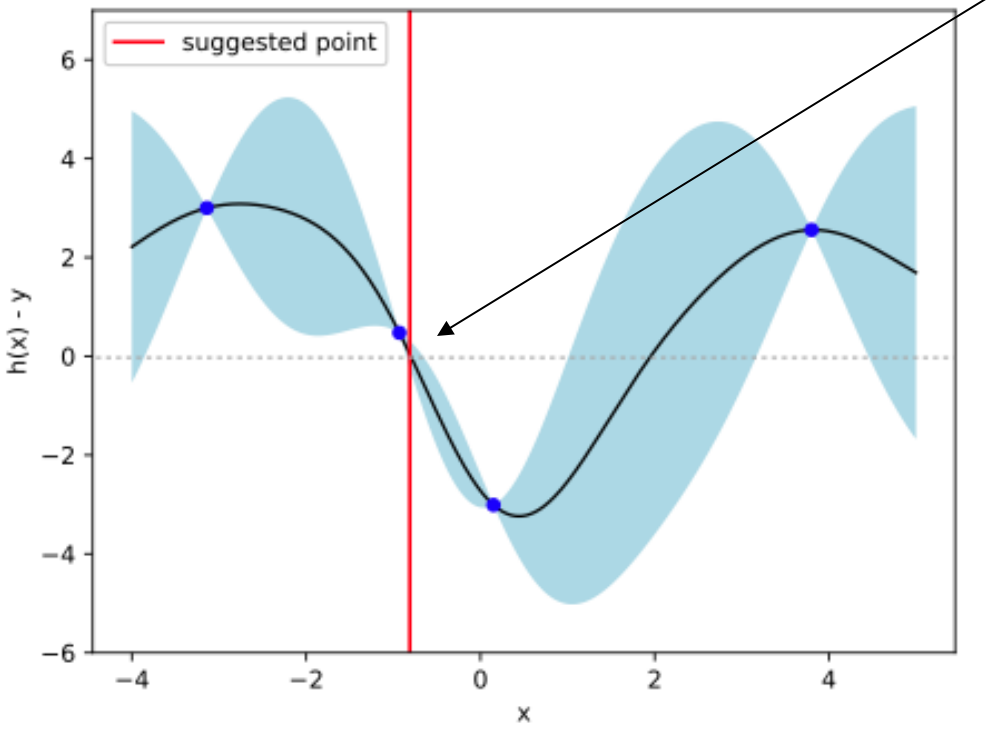

GP posterior on $h(x) - y$
using 3 data points



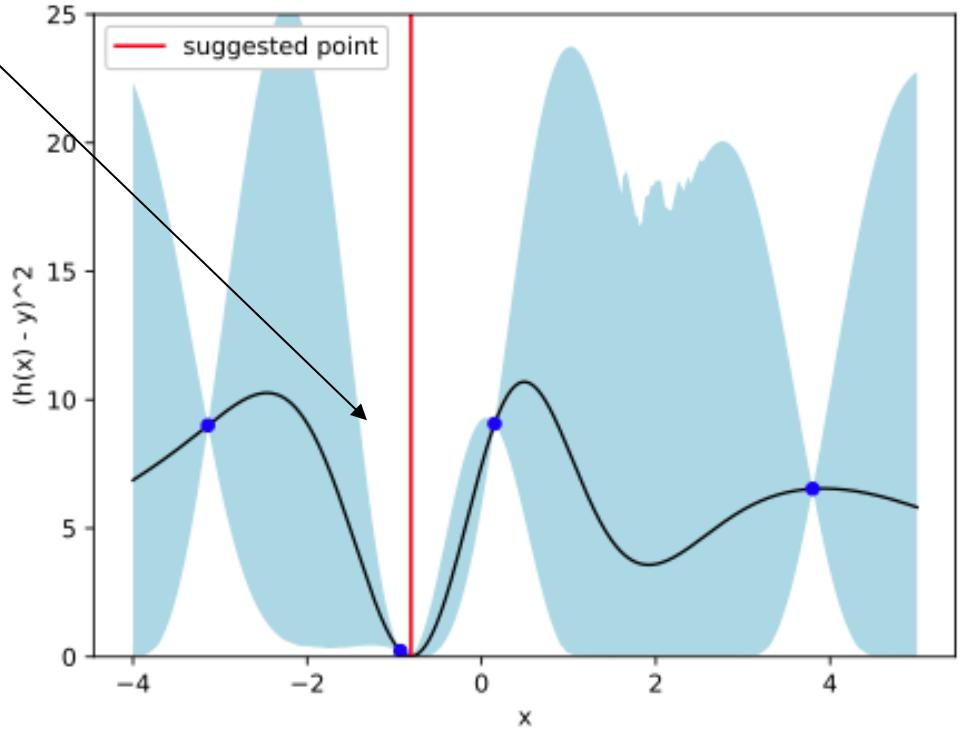

Implied posterior on $(h(x) - y)^2$
using 3 data points

# Now, let's solve same problem by better leveraging the structure of the objective function $f(x)$

Notice how new approach exploits positivity of the loss function to find a value of x that is more likely to make h(x) match y




GP posterior on $h(x) - y$ using 4 data points




Implied posterior on $(h(x) - y)^2$ using 4 data points

# Problem Setup: Bayesian Optimization of Composite Functions

- Consider problems of the form:

$$\max_{x \in \Omega} f(x)$$

Where $f(x) = g(h(x))$ is a composition of functions with properties:

- $h(x)$ is an expensive-to-evaluate, black-box, multi-output function
- $g(y)$ is a cheap-to-evaluate, differentiable, white-box function

[Astudillo and Frazier, *ICML*, 2019] *without constraints
[Paulson and Lu, *Computers & Chemical Engineering*, 2021] *with constraints

# Our Approach: The COBALT Method

Code available : https://github.com/joelpaulson/COBALT

while {budget not exhausted}

Fit **multi-output** Gaussian process regression to observations {x, **h(x)**}

Find x that maximizes a **new acquisition function**
COBALT(x) = E[{**g(h(x))** – f*}$^+$]

Sample x & then observe **h(x)**, f(x)

end

We have further modified acquisition to account for unknown constraints and noise

[Astudillo and Frazier, *ICML*, 2019] *without constraints
[Paulson and Lu, *Computers & Chemical Engineering*, 2021] *with constraints
[Lu and Paulson, *Journal of Process Control*, 2023] *with constraints and noise

# Composite Functions Arise in Many Practical Examples

- **Calibration of Expensive Black-box Forward Models**
  - h(x) = prediction of observed data as function of parameters
  - g(h(x)) = negative log-likelihood (+ regularization)

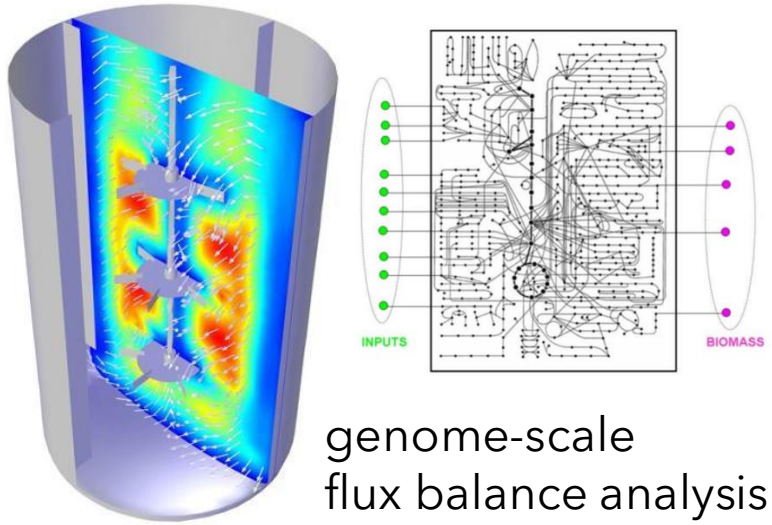
- **Materials Design**
  - h(x) = vector of different material attributes
  - g(h(x)) = combined performance measure over attributes

- **Process Flowsheet Optimization**
  - h(x) = reaction & separation efficiency as function of temp. and concentration
  - g(h(x)) = return on investment

# Genome-scale Bioreactor Model Calibration Test Problem

**Experimental measurements**

**Expensive computer simulation**




genome-scale
flux balance analysis

[Hanly, Urello, and Henson, 2012]

**Compare**

- Optimize simulation's parameters such that **log-likelihood** is maximized
  - six parameters related to bounds on extracellular uptake rate

# Genome-scale Bioreactor Model Calibration Test Problem

**Negative log-likelihood is a composite function:**

$$f(x) = g(h(x)) = \sum_{j=1}^{N} \log(0.0025 h_j^2(x)) + 400 h_j^{-2}(x)(y_j - h_j(x))^2$$

these terms are related to measurement noise term that depends on concentration

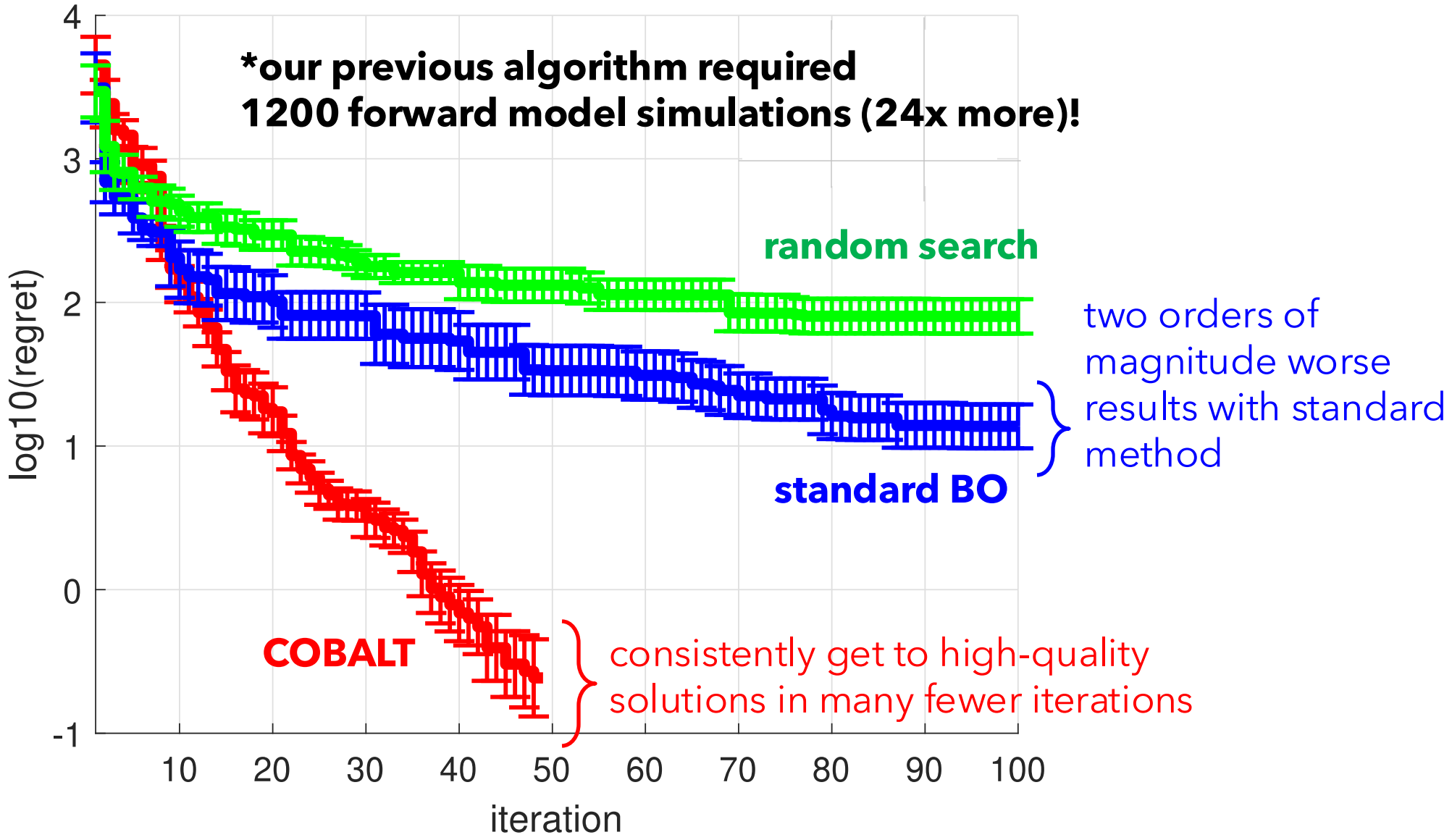
where $x$ is the vector of parameters that must be estimated

$y_j$ is the j$^{th}$ measured datapoint (e.g., extracellular concentrations)

$h_j(x)$ is the forward model prediction for the j$^{th}$ measurement

[Paulson et al., *PLOS One*, 2019]

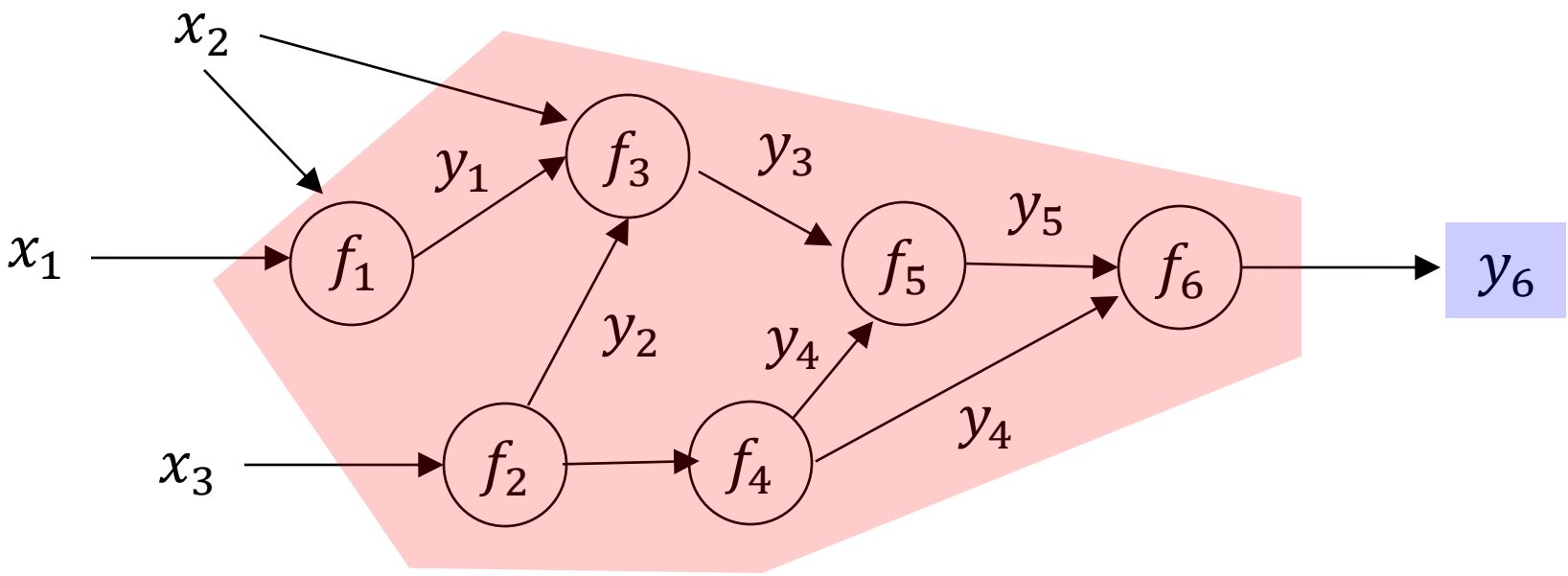
# Results: Log10(Regret) versus Number of Evaluations



# Function Networks

# Can further extend composites to networks of functions!

- The network corresponds to a **directed graph** with some number of nodes and edges, with a **single leaf node** corresponding to our objective




- Each function $f_1, \ldots, f_K$ can be modeled using a separate GP prior
  - Can encode any desired information through choice of prior including a given function is exactly known (prior kernel is zero function → no uncertainty)

[Astudillo and Frazier, *NeurIPS*, 2021]

# Bayesian Optimization for Function Networks

- Let $J(k)$ be the set of parent nodes of node $k$ and $I(k)$ be the subset of the decision variables impacting the function at node $k$

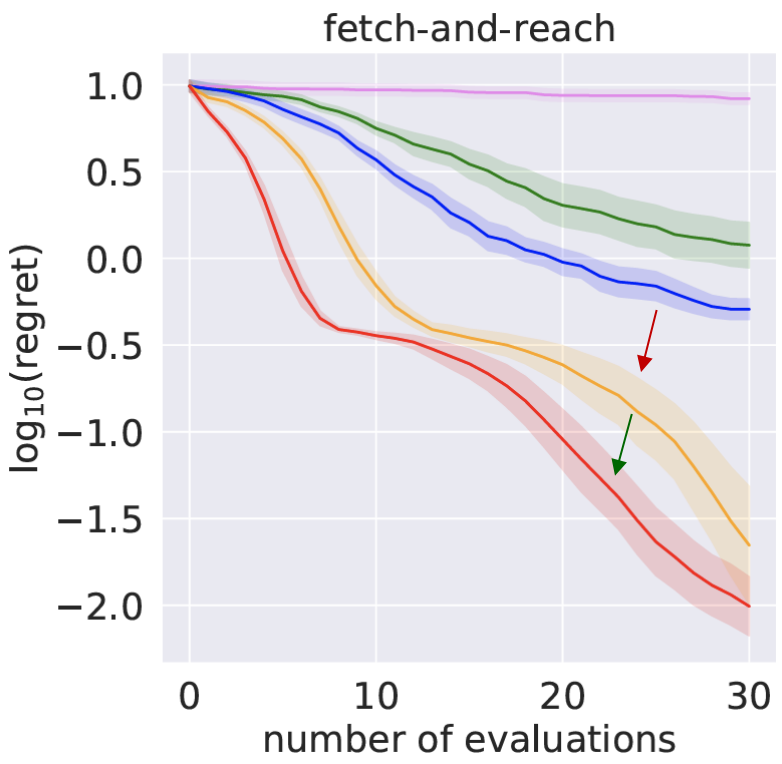
- Let $h_1(x), \ldots, h_K(x)$ be the values of the $K$ nodes in the function network when it is evaluated at $x$, which are defined recursively:

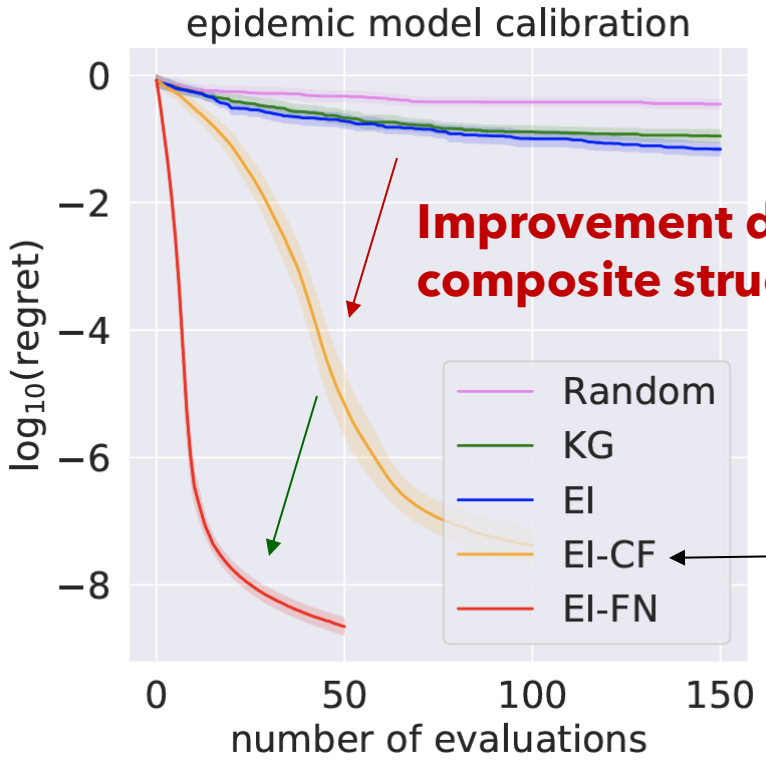
$$h_k(x) = f_k\left(x_{I(k)}, h_{J(k)}(x)\right), \qquad k = 1, \ldots, K$$

- The objective function is: $g(x) = h_K(x)$
  - GP posteriors for $f_1, \ldots, f_K$ imply a generally non-Gaussian posterior for $g$

- Multiple acquisition functions have been proposed:
  - Expected improvement for function networks (EIFN) [Astudillo and Frazier, *NeurIPS*, 2021]
  - Upper confidence bound-like approach (MCBO) [Sussex et al., *ICLR*, 2023]
  - Knowledge gradient with partial information (p-KGFN) [Buathong et al., *ICML*, 2024]

# Results: Comparison of EIFN to other methods




**Improvement due to composite structure**

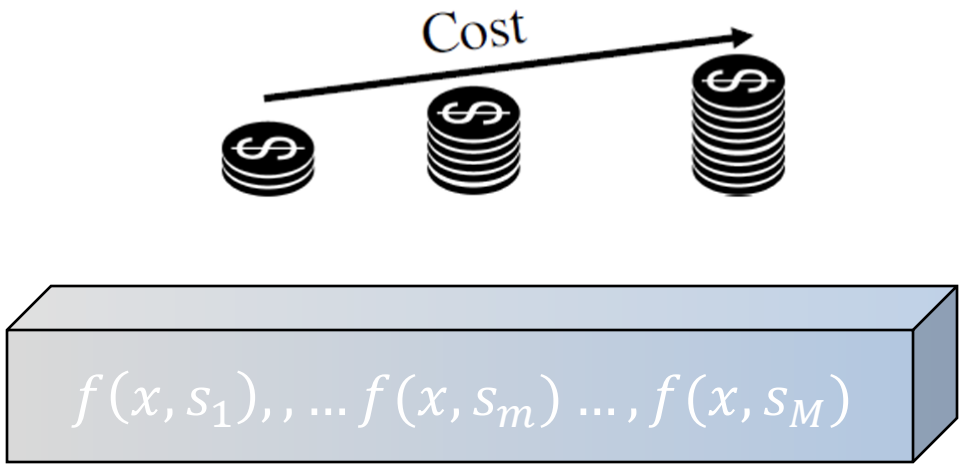
Unconstrained form of COBALT

**Further improvement due to network structure**

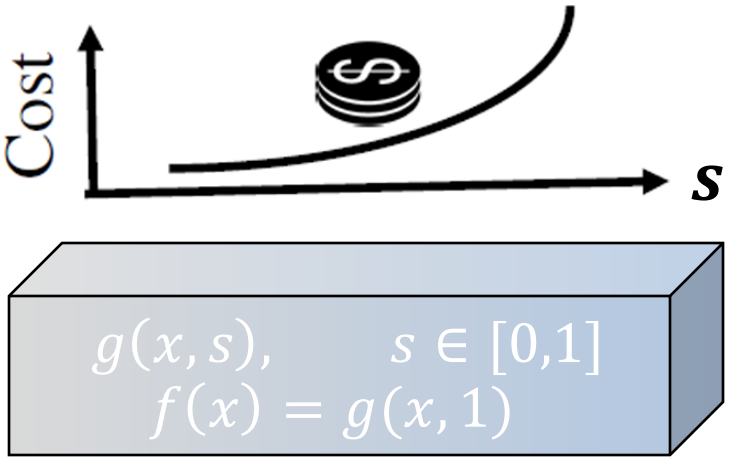
[Astudillo and Frazier, *NeurIPS*, 2021]

# Multi-Fidelity Representations

# Multi-fidelity Bayesian Optimization: The Problem



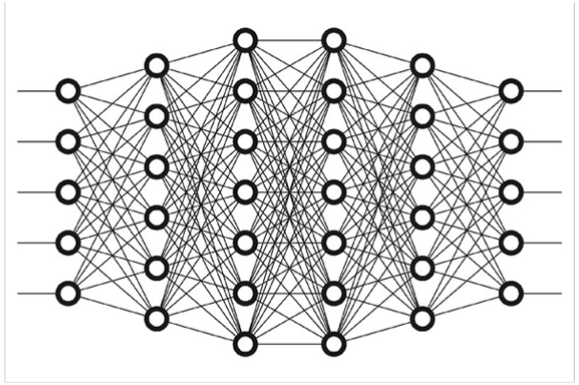

Discrete fidelities



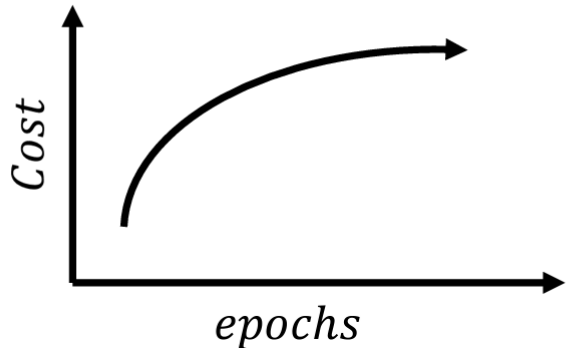

Continuous fidelities

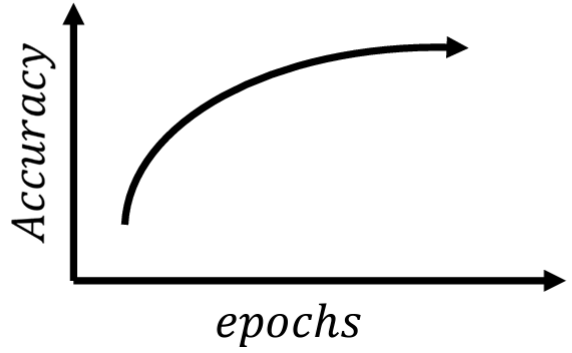
- **Goal:** (approximately) optimize the highest fidelity function by minimizing the total resources consumed across experiments
  - Must account for correlation across fidelities/tasks
  - Cost vs. accuracy tradeoff for the function approximations

# Example #1: Automated Machine Learning (AutoML)



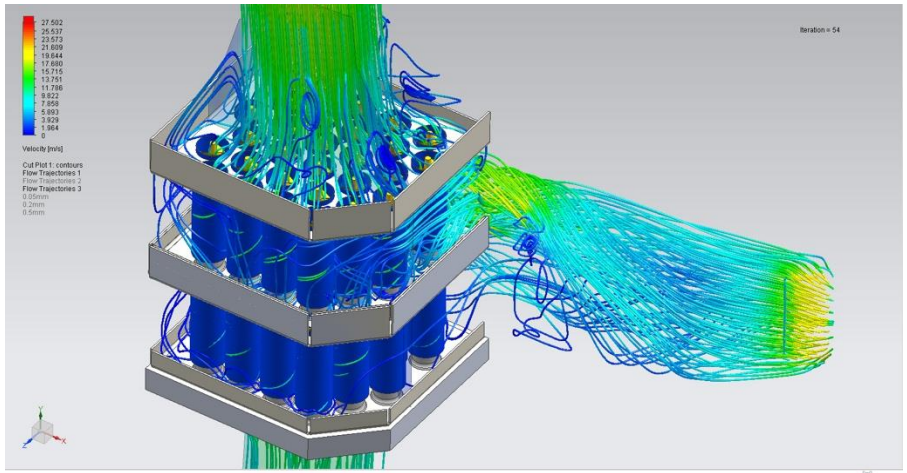
Cost vs. Accuracy trade-offs in evaluating hyperparameter configurations



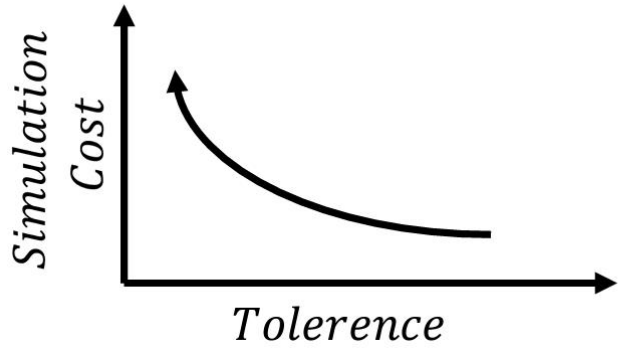



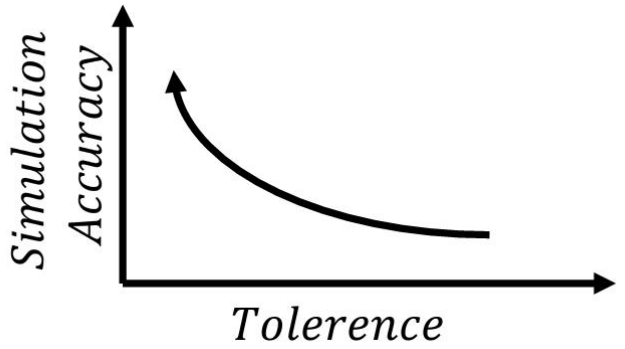

[Adapted from Aryan, Belakaria, and Doppa, Half-day tutorial, *AAAI*, 2022]

# Example #2: Engineering Design via Simulations



Cost vs. Accuracy trade-offs in evaluating hardware designs




[Adapted from Aryan, Belakaria, and Doppa, Half-day tutorial, *AAAI*, 2022]

# Multi-fidelity Bayesian Optimization: Key Challenges

- **Intuition:** We want to use inexpensive (low-fidelity) experiments to gain information and effectively "prune" the input space – save our expensive (high-fidelity) experiments for only most promising candidates

- **Modeling challenge:** How can we model multi-fidelity functions to allow for information sharing across tasks?

- **Reasoning challenge:** How can we simultaneously select the input location and the fidelity level at every iteration?

# Multi-fidelity Bayesian Optimization: Key Challenges

- **Intuition:** We want to use inexpensive (low-fidelity) experiments to gain information and effectively "prune" the input space – save our expensive (high-fidelity) experiments for only most promising candidates

- **Modeling challenge:** How can we model multi-fidelity functions to allow for information sharing across tasks?
  - Again, we can use GPs, but now need a different style of kernel…

- **Reasoning challenge:** How can we simultaneously select the input location and the fidelity level at every iteration?

# Multi-fidelity Bayesian Optimization: Key Challenges

- **Intuition:** We want to use inexpensive (low-fidelity) experiments to gain information and effectively "prune" the input space – save our expensive (high-fidelity) experiments for only most promising candidates

- **Modeling challenge:** How can we model multi-fidelity functions to allow for information sharing across tasks?
  - Again, we can use GPs, but now need a different style of kernel…

- **Reasoning challenge:** How can we simultaneously select the input location and the fidelity level at every iteration?
  - Need a "cost-aware" acquisition function that roughly measures value of information per unit cost of evaluation

# Multi-fidelity Gaussian Processes

- The core idea is to model the unknown function in an augmented space that includes the input and fidelity parameter(s), i.e.,

$$f(x, s) \sim \mathcal{GP}(\mu(\{x, s\}), k(\{x, s\}, \{x', s'\}))$$

- The question is what type of kernel should we use?
  - Cannot use a standard RBF kernel for multiple reasons (not directly applicable to unordered discrete variables & unlikely to capture realistic behavior)

- Multiple options, best choice will depend on application
  - For discrete fidelities and limited prior knowledge, I recommend treating the fidelity levels as "categorical" and using a **Mixed Single Task GP** (next slide)

# GP Model for Mixed Search Spaces

- Given continuous variables $x$ and categorical variables $s$, previous work has shown that the following kernel can be effective:

$$k(\{x,s\},\{x',s'\}) = k_{\text{cont},1}(x,x') + k_{\text{cat},1}(s,s') + k_{\text{cont},2}(x,x')k_{\text{cat},2}(s,s')$$

Where $k_{\text{cont},1}$, $k_{\text{cont},2}$ are kernels for the continuous variables and $k_{\text{cat},1}$, $k_{\text{cat},2}$ are kernels for the categorical variables (usually based on Hamming distance)

  – Additive kernel provides simple base (common trend over $x$ with $s$-dependent offset)
  – Product kernel allows for richer expression but may be hard to learn with limited data

- As one would expect, this kernel is not differentiable with respect to $s$ such that we must use mixed-integer optimization to jointly search over $\{x,s\}$

[Ru et al., *ICML*, 2020; CoCaBO]

# Acquisition Functions for Multi-fidelity Bayesian Optimization

- All the major acquisition functions have been extended to the multi-fidelity Bayesian optimization case (only some of them support continuous fidelities):
  - Expected improvement [Lam et al., Structures, Structural Dynamics, and Materials, 2015]
  - Upper confidence bound [Kandasamy et al., ICML, 2017]
  - Knowledge gradient [Poloczek et al., NeurIPS, 2017]
  - Entropy search [Takeno et al., ICML, 2020]

- Inevitably, they end up with an acquisition of the form below. KG and ES-like methods have very principled way to define the "potential improvement".

$$\alpha(x, s) = \text{Potential Improvement}(x, s)/\text{Cost}(x, s)$$

- When cost function is unknown, typically modeled with an independent GP

# Example: Multi-fidelity Optimization for Reinforcement Learning

**What information is available?**

High-fidelity
- 0.02 s integration time, 100 initial states

Lower-fidelities
- 0.04 s integration time, 40 initial states (5x savings)
- 0.02 s integration time, 10 initial states (10x savings)

**Not clear which fidelity is better approximation!**

**What is the problem?**

Reinforcement learning problem where we want to find best policy to keep cart upright by applying a forces to the left and right of cart that depend on states

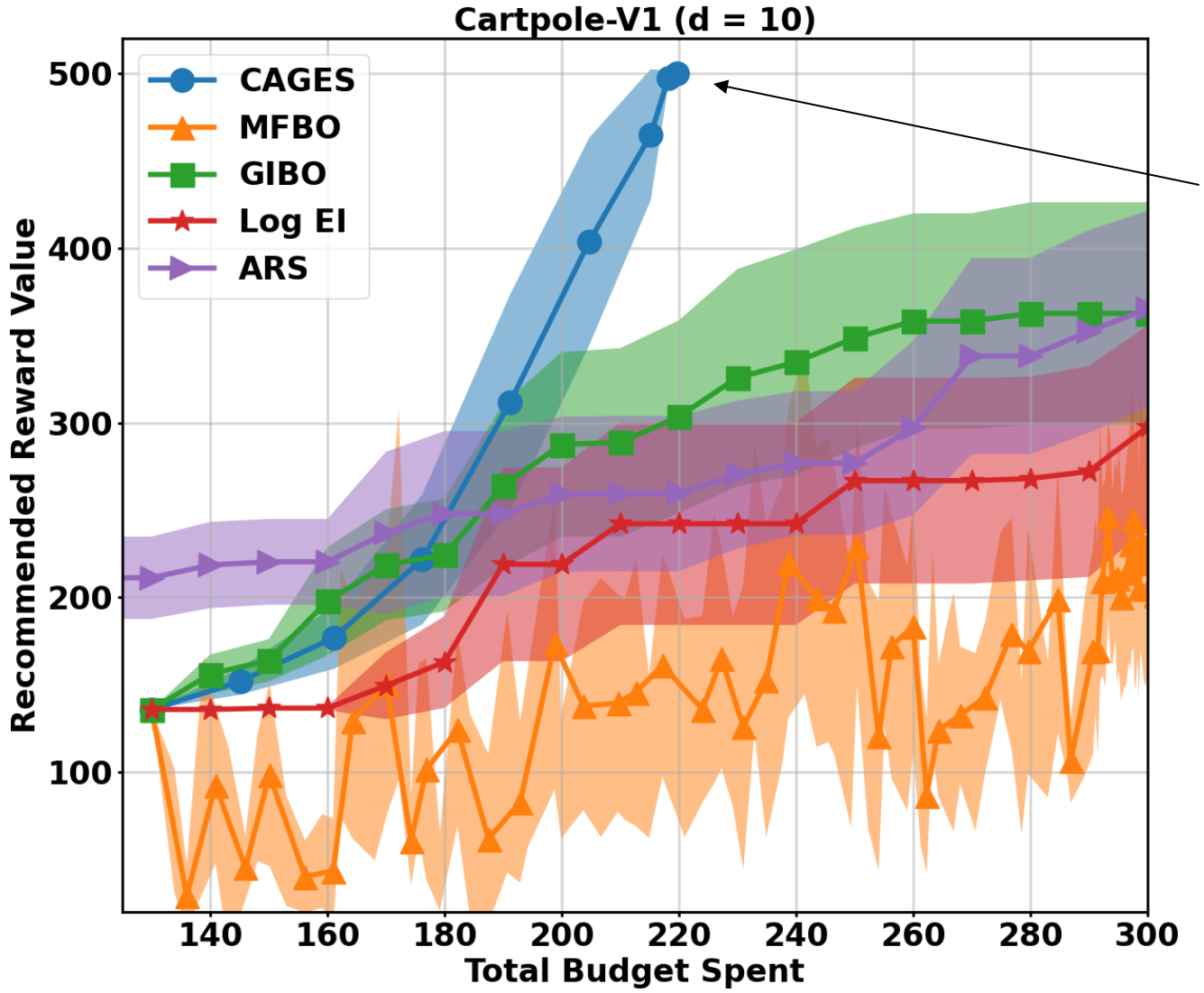
10d neural network policy

**Relatively high dimensional**
**→ Local version called CAGES (learn the gradient)**

[Tang and Paulson, *CDC*, 2024; CAGES]

# Example: Multi-fidelity Optimization for Reinforcement Learning

Results on OpenAI Gym Cartpole problem using CAGES



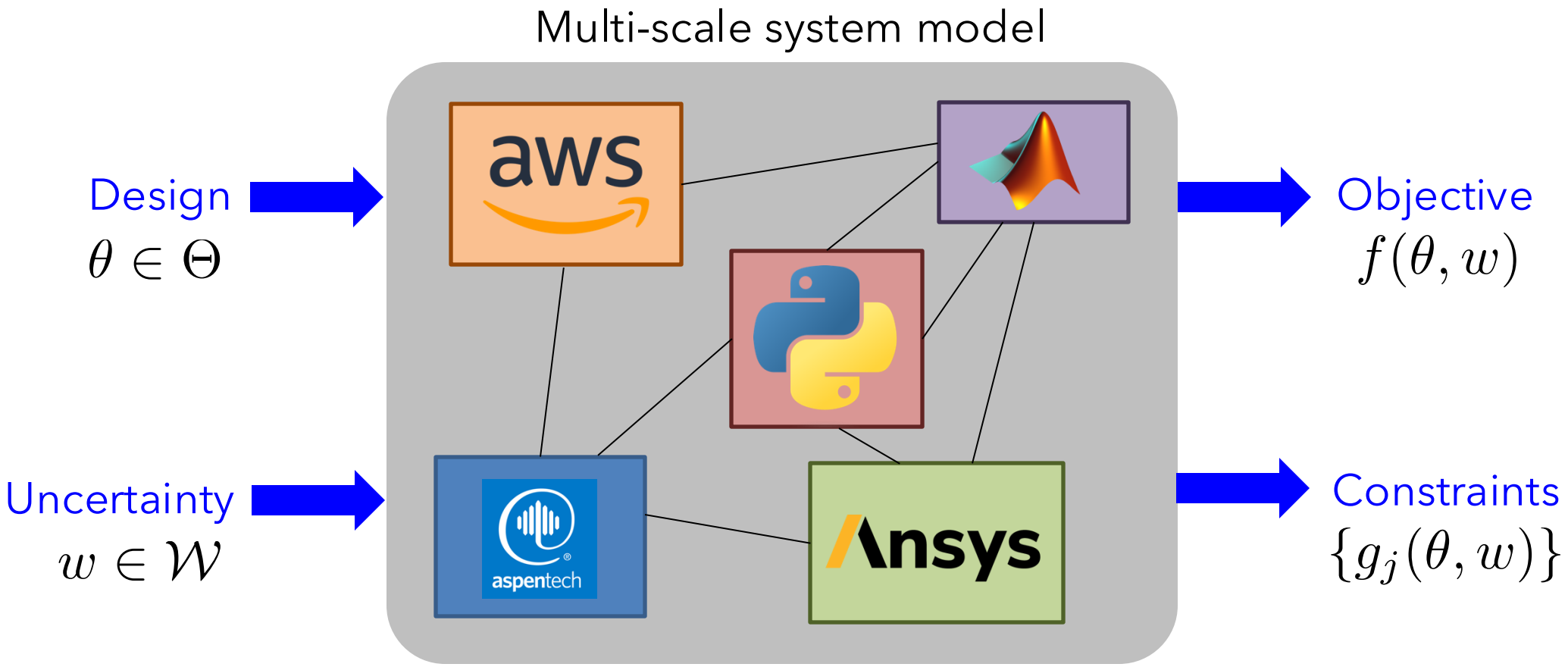

CAGES can find the best possible reward using ~220 budget, which substantially outperforms all other methods including (log)EI

# Outline

- What is standard Bayesian optimization missing?
  - Strong priors, known structure, safety considerations, uncertainty
- Beyond sequential & single objective problems
  - Parallel evaluations (synchronous & asynchronous), multi-objective optimization
- Beyond the black-box problem structure
  - Composites, function networks, multi-fidelity representations
- Beyond nominal optimization
  - Adversarial uncertainty, flexibility analysis, safety
- What is next?
  - Meta learning, preference learning, new ways to optimize acquisition, planning

# Robust Bayesian Optimization

# Robust Design of Expensive Multi-Scale Simulators




**Expensive (hours to days) & No-closed form equations available**

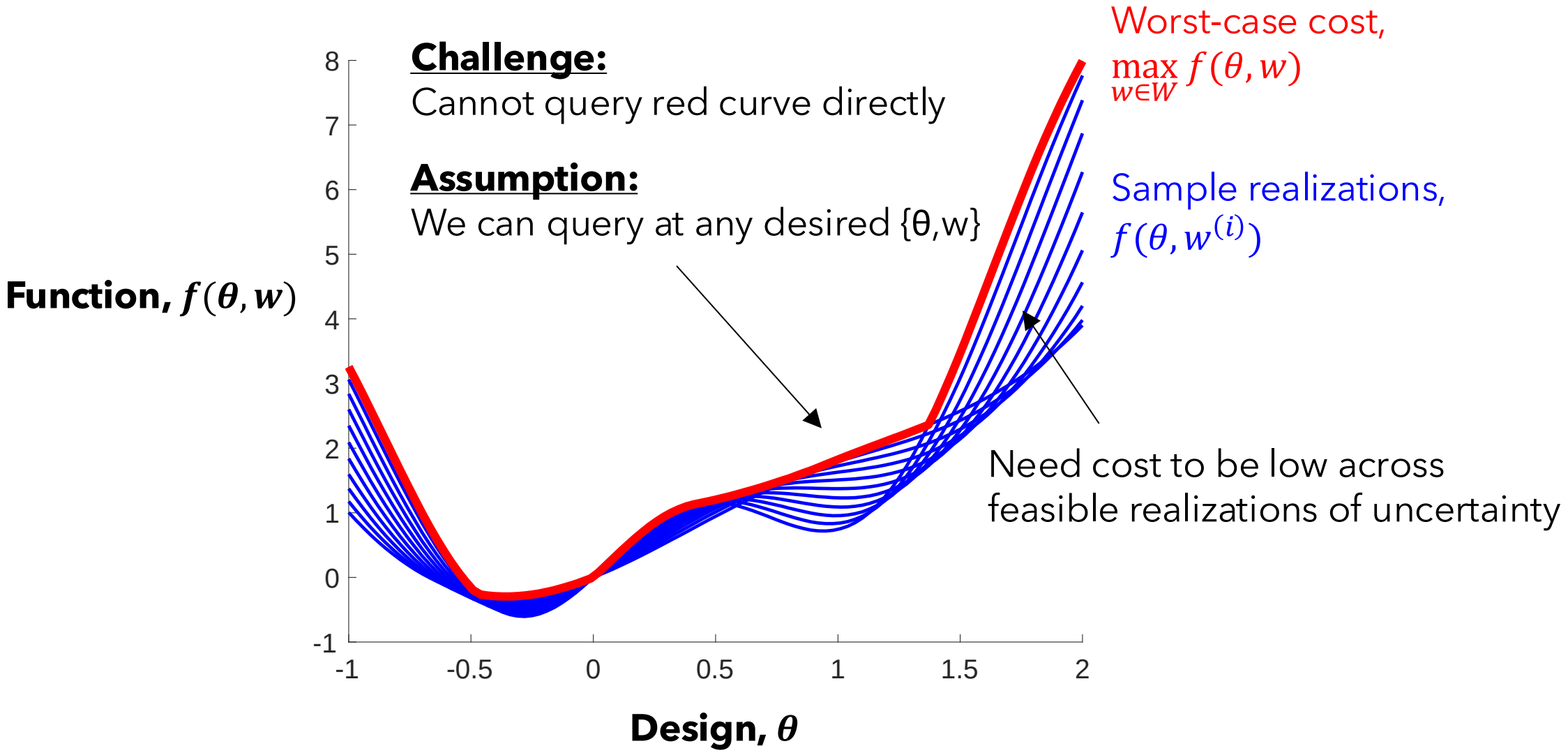
→ can only simulate at specific $\{\theta, w\}$ and observe function values

# Design Problem = *Robust* Global Optimization Problem

**How to solve** $\left\{ \begin{array}{ll} \min\limits_{\theta \in \Theta} \max\limits_{w \in \mathcal{W}} & f(\theta, w), \\ \text{subject to:} & g_j(\theta, w) \leq 0, \quad \forall (w, j) \in \mathcal{W} \times \mathcal{J} \end{array} \right\}$ **?**

- Algorithm should not use gradient of functions and must account for worst-case uncertainty (**constrained robust black-box optimization**)
- Algorithm should not get "stuck" at local solutions (**global optimization**)
- Algorithm should make the **fewest calls** to objective $f(\cdot)$ and constraints $g_j(\cdot)$ as possible since it is very expensive to evaluate

# Illustration of Worst-Case Values



# The MiMaReK Algorithm*

(roughly apply Bayesian optimization to both optimization levels)

**Algorithm 1** Minimax optimization via relaxation

1: Pick $\mathbf{x}_e^{(1)} \in \mathbb{X}_e$ and set $\mathcal{R}_e = \left\{\mathbf{x}_e^{(1)}\right\}$ and $i = 1$.

Start with 1 uncertainty

2: Compute

$$\mathbf{x}_c^{(i)} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \left\{ \max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c, \mathbf{x}_e) \right\}$$

Bayesian optimization over finite # of uncertainty points to get next design

3: Compute

$$\mathbf{x}_e^{(i+1)} = \arg \max_{\mathbf{x}_e \in \mathbb{X}_e} y(\mathbf{x}_c^{(i)}, \mathbf{x}_e)$$

Bayesian optimization at fixed design to get next uncertainty

4: If

$$y(\mathbf{x}_c^{(i)}, \mathbf{x}_e^{(i+1)}) - \max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c^{(i)}, \mathbf{x}_e) < \varepsilon_R$$

then return $\left\{\mathbf{x}_c^{(i)}, \mathbf{x}_e^{(i+1)}\right\}$ as an approximate solution to the initial minimax problem (1).

Else, append $\mathbf{x}_e^{(i+1)}$ to $\mathcal{R}_e$, increment $i$ by 1 and go to Step 1.

Add new uncertainties to set

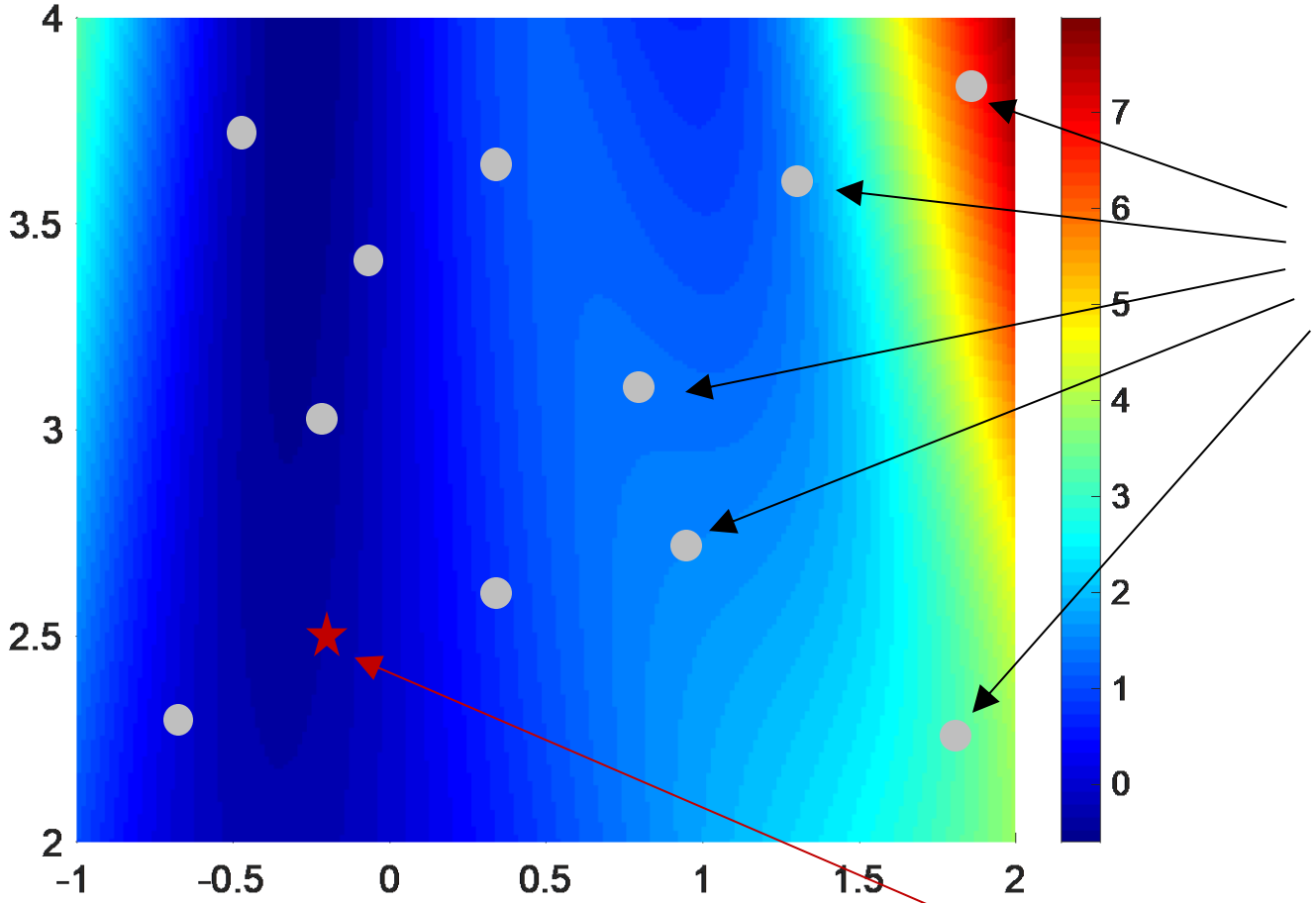
**Major challenge**

- Growing sample cost at each iteration

*[Marzat et al., *Journal of Global Optimization*, 2013]

# How Can We Better Limit the Number of Function Evaluations?

- Key Idea: Model {θ,*w*} Relationship Simultaneously




Obtain objective value for some initial {θ,w} values

Build probabilistic surrogate model to predict functions at new {θ,w} values

→ Use to query new {θ,w}; **HOW?**

# Sketch of the Derivation of ARBO

$$r_t^w = \max_{w \in \mathcal{W}} f(x_t, w) - \min_{x \in \mathcal{X}} \max_{w \in \mathcal{W}} f(x, w)$$ **Start with robust regret definition**

[Bound the function using upper and lower confidence bounds]

$$\leq \max_{w \in \mathcal{W}} \mathrm{ucb}_{t-1}(x_t, w) - \min_{x \in \mathcal{X}} \max_{w \in \mathcal{W}} \mathrm{lcb}_{t-1}(x, w)$$ ← **This sets selection rule**

[We can rewrite these quantities in terms of the selected samples]

$$= \mathrm{ucb}_{t-1}(x_t, w_t) - \max_{w \in \mathcal{W}} \mathrm{lcb}_{t-1}(x_t, w)$$

[We know that $\max_{w \in \mathcal{W}} \mathrm{lcb}_{t-1}(x_t, w) \geq \mathrm{lcb}_{t-1}(x_t, w)$ for any feasible w]

$$\leq \mathrm{ucb}_{t-1}(x_t, w_t) - \mathrm{lcb}_{t-1}(x_t, w_t)$$

[Relate the difference between upper and lower confidence bounds to variance]

$$= 2\kappa_t \sigma_{t-1}(x_t, w_t)$$ **Can show variance is decaying as iterations increase**

# How to Extend ARBO to Handle Worst-Case Constraints?

- Take advantage of **exact penalty functions**, which provide the necessary properties to demonstrate theoretical convergence results

- Two main modifications to the ARBO loop:

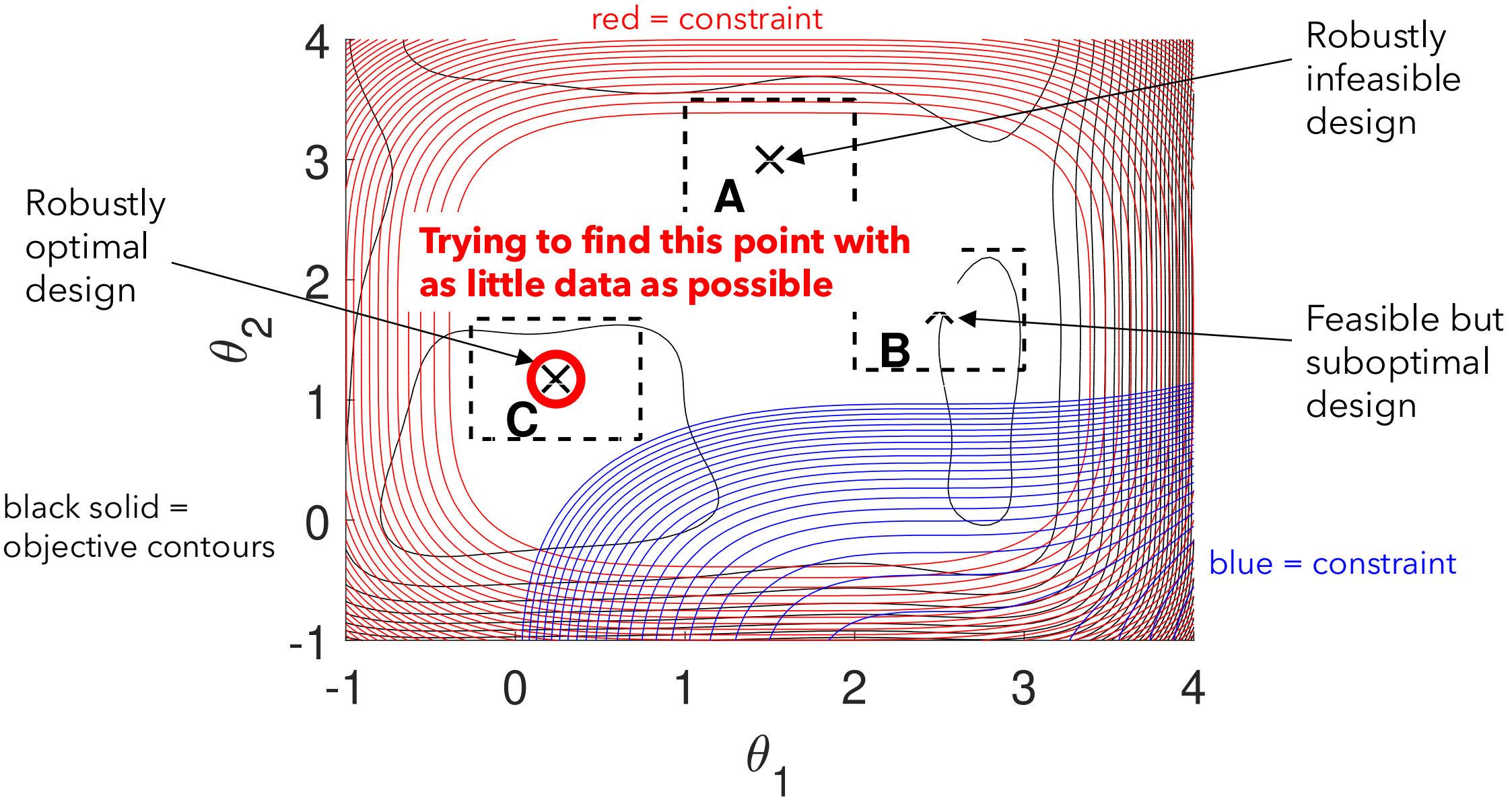
  1. Find $\theta$ that solves a minimax problem in terms of a lower confidence bound for a (non-smooth) penalty-based objective function

  2. Find set of $\{w_v\}$ that maximizes upper confidence bound of each unknown function $v \in \{f, g_1, \ldots, g_{|\mathcal{J}|}\}$ → evaluate at a set of points

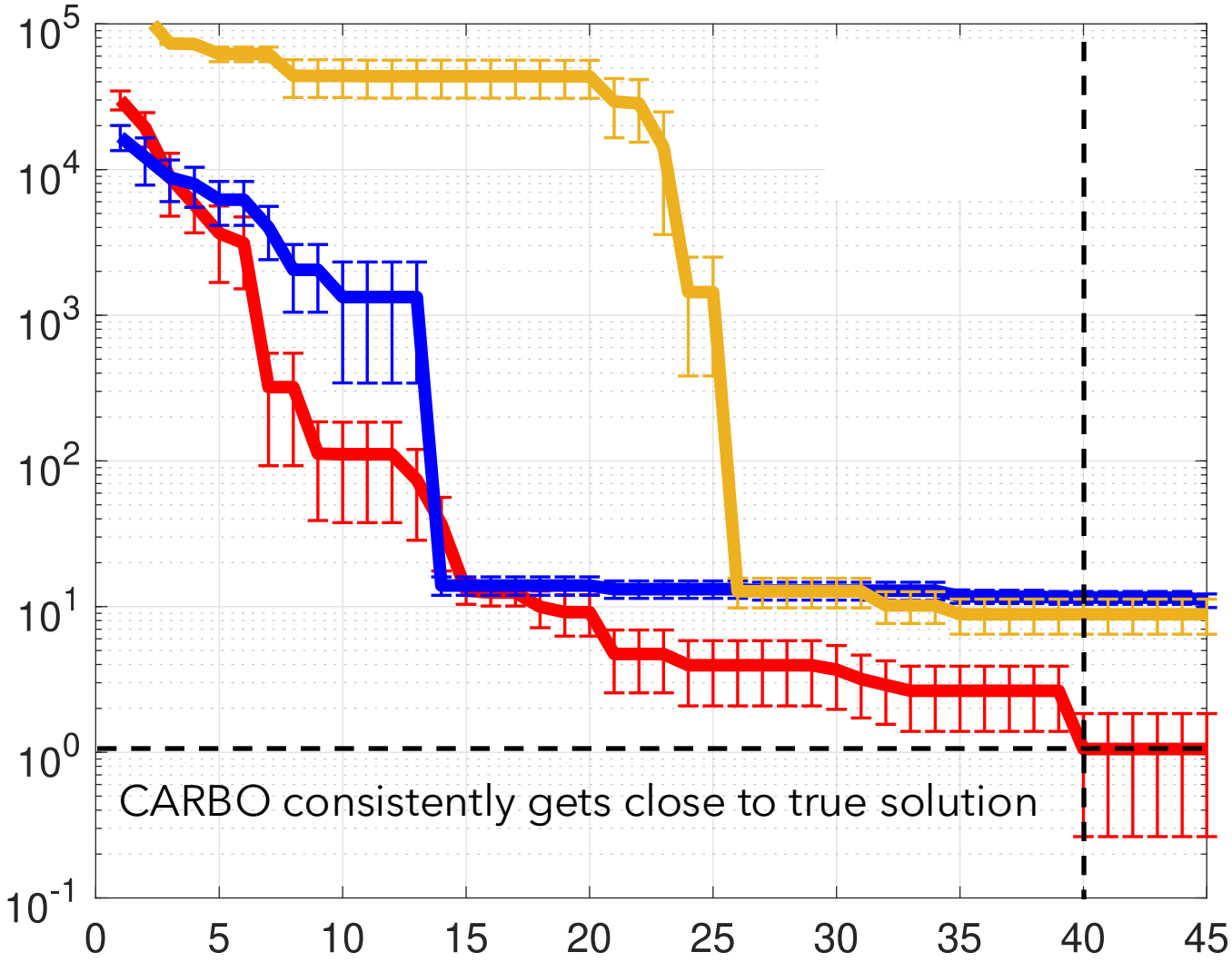
[Kudva, Sorourifar, and Paulson, *AIChE Journal*, 2022]

# Illustrative Example:

## Polynomial System with Implementation Errors



# Illustrative Example:
## Performance Comparison with Alternative Methods




[Bertsimas et al., *INFORMS Journal on Computing*, 2010]

- Local derivative-free approach to constrained robust optimization

**Bertsimas**

✖

>>100

# Bayesian Optimization for Flexibility Analysis

# Can we extend the ARBO concept to more optimization levels?

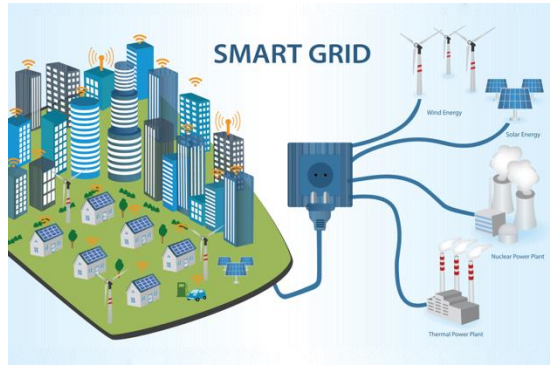
[For example, can we use it to perform flexibility analysis]

- The motivation for many next-generation manufacturing & energy systems of interest is their ability to operate in a <u>flexible</u> manner

Combined heat & power plant

Multiproduct chemical plants

Smart grid




- "Flexible" → ability to <u>adapt</u> to new, different, or <u>changing requirements</u>
  - Adapt = recourse, Changes = Parameters known in future (e.g., supply/demand)

- Given a design, how can we systematically test for & quantify flexibility?

# Can we extend the ARBO concept to more optimization levels?

[For example, can we use it to perform flexibility analysis]

- The flexibility test problem formulated as tri-level optimization*

Worst-case values for the uncertainties ($\theta$)

Worst-case constraint value ($j$, finite number)

$$\chi = \max_{\theta \in \Theta} \min_{z \in \mathcal{Z}} \max_{j \in \mathcal{J}} f_j(\theta, z)$$

Best-case value for recourse variables (z)

Key idea: Treat this as a composite operator and then combine with ARBO!

- $\chi \leq 0$: Verified feasible operation attainable over all $\Theta$ **(test passed)**

- $\chi > 0$: Verified feasible operation not obtained for part of $\Theta$ **(test failed)**

*[Halemane & Grossmann, *AIChE Journal*, 1983]

# BoFlex to the rescue!

**Algorithm 1** BoFlex: Bayesian Optimization for Black-Box Flexibility Tests

**Input:** Uncertain parameter domain $\Theta$;
Recourse variable domain $\mathcal{Z}$;
Kernel for GP prior $k((\boldsymbol{\theta}, \mathbf{z}, j), (\boldsymbol{\theta}', \mathbf{z}', j'))$;
Confidence interval parameters $\{\beta_t^{1/2}\}_{t\geq 0}$.

**for** $t = 0, 1, 2, \ldots$ **do**
  $\hat{\chi}_t^U \leftarrow \max_{\theta\in\Theta} \min_{\mathbf{z}\in\mathcal{Z}} \max_{j\in\mathcal{J}} u_t(\boldsymbol{\theta}, \mathbf{z}, j)$
  $\hat{\chi}_t^L \leftarrow \max_{\theta\in\Theta} \min_{\mathbf{z}\in\mathcal{Z}} \max_{j\in\mathcal{J}} l_t(\boldsymbol{\theta}, \mathbf{z}, j)$
  **if** $\hat{\chi}_t^U \leq 0$ **then**
    Declare test passed (system is *flexible*) and stop.
  **end if**
  **if** $\hat{\chi}_t^L > 0$ **then**
    Declare test failed (system is *not flexible*) and stop.
  **end if**
  $\boldsymbol{\theta}_{t+1} \leftarrow \text{argmax}_{\theta\in\Theta} \min_{\mathbf{z}\in\mathcal{Z}} \max_{j\in\mathcal{J}} u_t(\boldsymbol{\theta}, \mathbf{z}, j)$
  $\mathbf{z}_{t+1} \leftarrow \text{argmin}_{\mathbf{z}\in\mathcal{Z}} \max_{j\in\mathcal{J}} l_t(\boldsymbol{\theta}_{t+1}, \mathbf{z}, j)$
  Query noisy simulator: $\hat{f}_j(\boldsymbol{\theta}_{t+1}, \mathbf{z}_{t+1})$, $\forall j = 1, \ldots, q$
  Update GP model with new data collected in previous step
**end for**

See paper for details:

- Kudva, Tang, and Paulson, *Computers and Chemical Engineering*, 2024

- Notice use of both **upper** and **lower** bounds for selection of new points and for stopping criteria

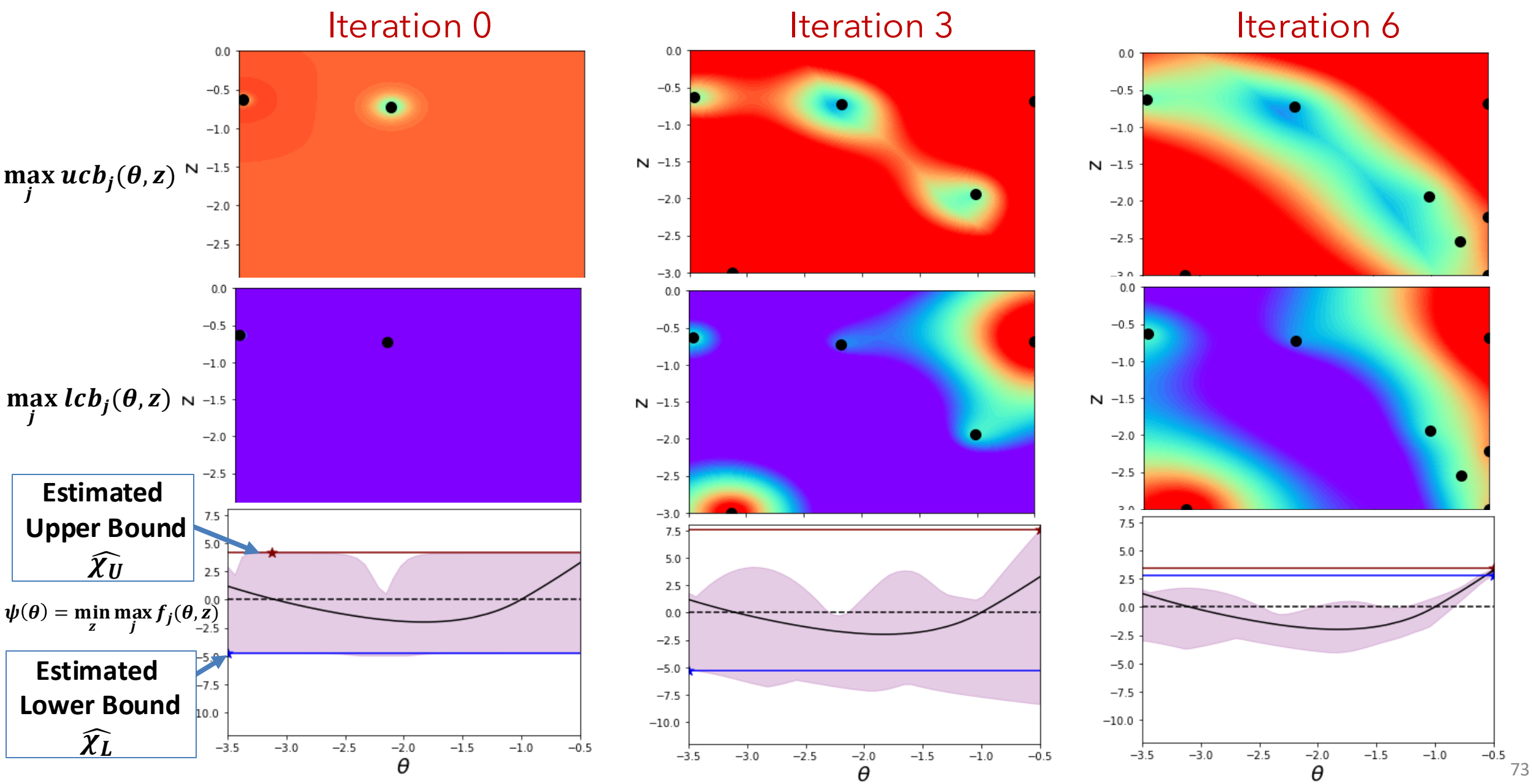
Code available on Github:

- https://github.com/PaulsonLab/BoFlex

[Kudva, Tang, Paulson, *CACE*, 2024]

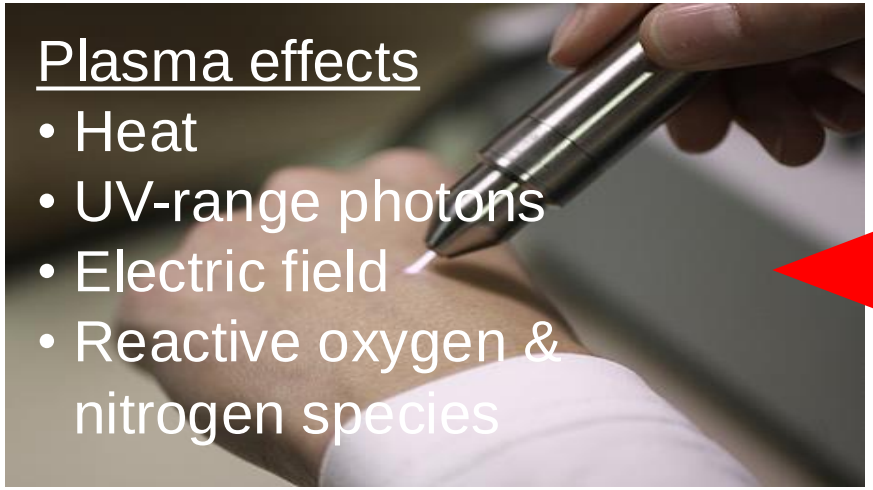
# BoFlex Illustration

https://github.com/PaulsonLab/BoFlex

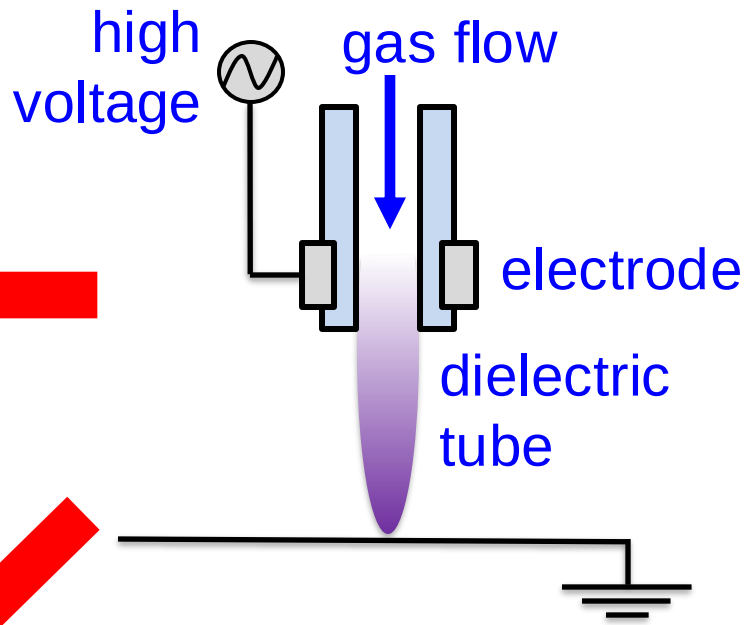


# Safe Bayesian Optimization

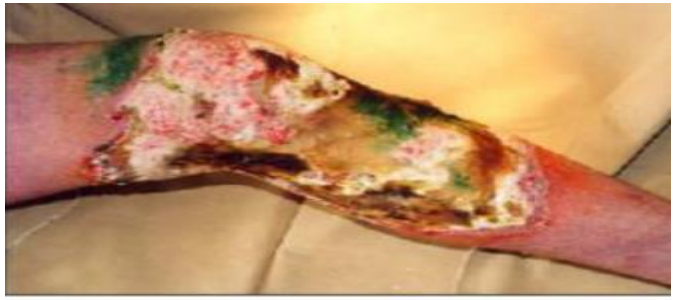
# Motivation: Problems with *safety-critical* constraints




kINPen, Griefswald, Germany




before plasma treatment

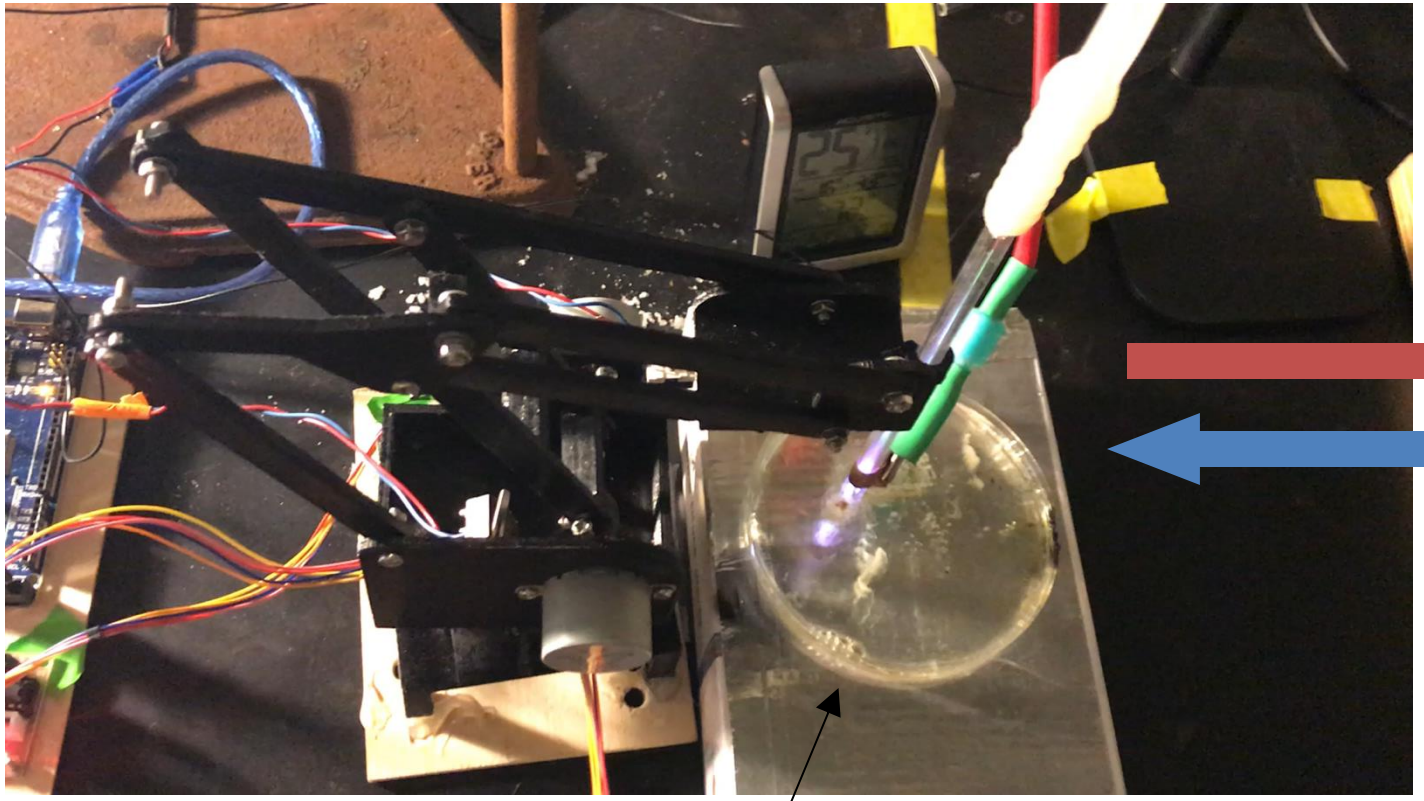


after 5 sessions

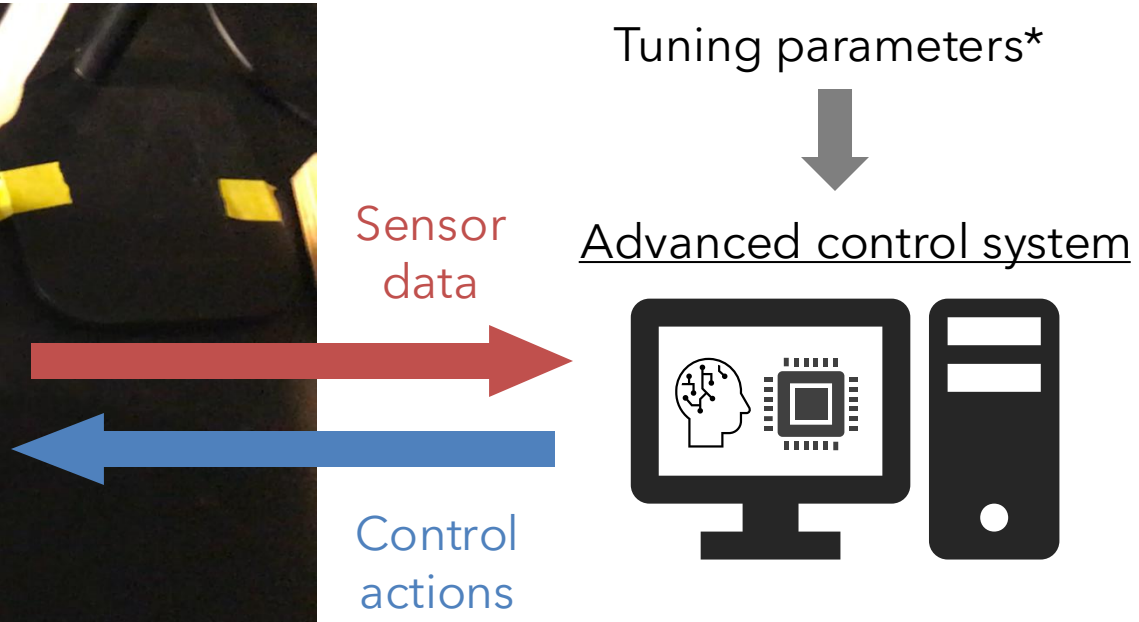
Fridman+, *Plasma Processes and Polymers*, 5:503-533, 2008

# Motivation: Problems with *safety-critical* constraints

- Atmospheric pressure plasma jets (APPJs) are portable device to deliver dose of cold plasma, which is a highly reactive medium with potential health benefits



Patient substrate (e.g., skin, wound, tumor)




*What are best tuning parameters?

Collaboration with Ali Mesbah, CBE, UC Berkeley

# Motivation: Problems with *safety-critical* constraints

- We can think of this as a constrained Bayesian optimization problem:

$$\min_{x \in \Omega} f^0(x) \text{ subject to } f^i(x) \geq 0 \text{ for all } i = 1, \ldots, m$$

$f^0(x)$ is our standard objective function (e.g., dose delivered)
$f^1(x), \ldots, f^m(x)$ are our safety-critical constraints (e.g., do not burn patient)

- We could apply one of the constrained Bayesian optimization methods that we saw in the previous set of slides, **why might that be bad?**

# Motivation: Problems with *safety-critical* constraints

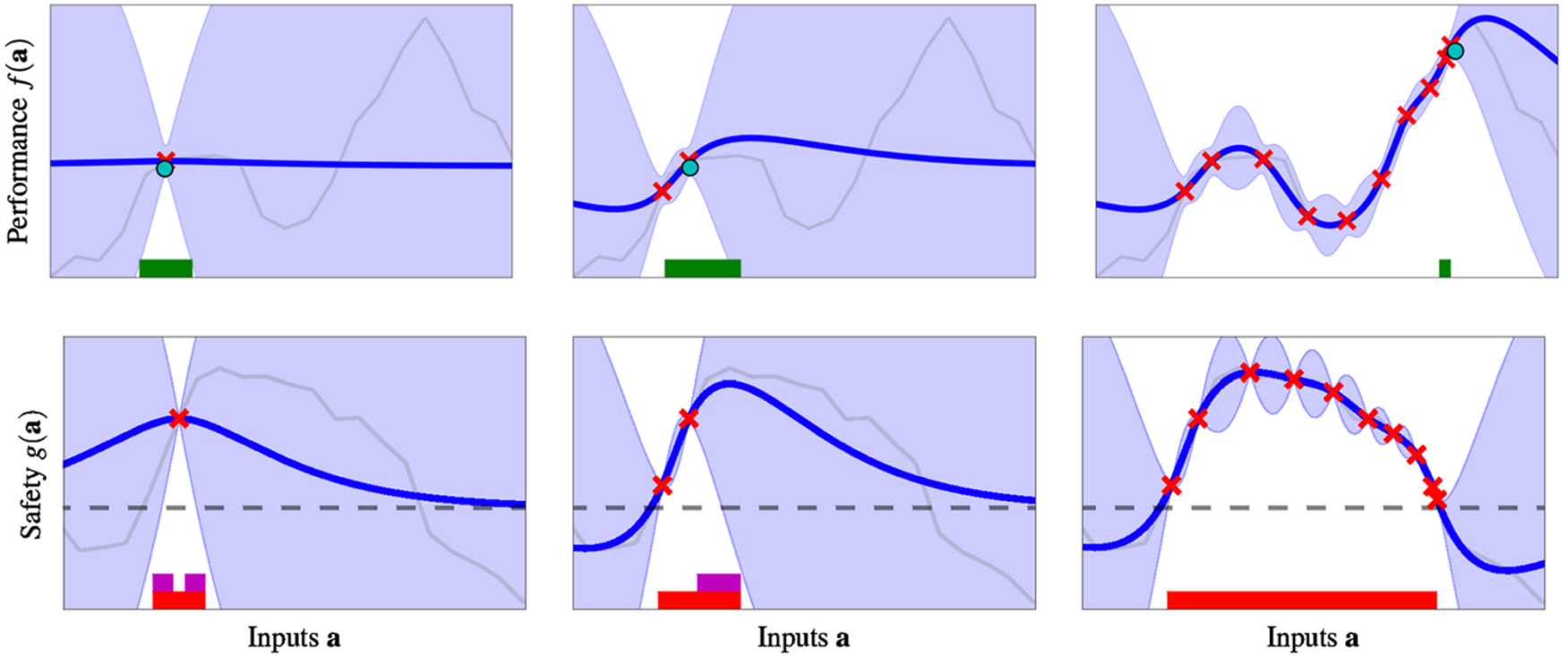
- We can think of this as a constrained Bayesian optimization problem:

$$\min_{x \in \Omega} f^0(x) \text{ subject to } f^i(x) \geq 0 \text{ for all } i = 1, \ldots, m$$

$f^0(x)$ is our standard objective function (e.g., dose delivered)
$f^1(x), \ldots, f^m(x)$ are our safety-critical constraints (e.g., do not burn patient)

- We could apply one of the constrained Bayesian optimization methods that we saw in the previous set of slides, **why might that be bad?**
  - We might violate constraints during our iteration process, which is not acceptable!
  - **Key challenge: Need to somehow balance exploration with safety**

# Illustration of SafeOpt (one approach)




**(a)** Initial, safe parameters. **(b)** Safe exploration. **(c)** After 10 evaluations: safe maximum found.

- Set of possible maximizers
- Current knowledge of safe set
- Set of possible "expanders" that could enlarge safety set

[Berkenkamp et al., *Machine Learning*, 2023]

# (Non-Exhaustive) Methods for Safe Bayesian Optimization

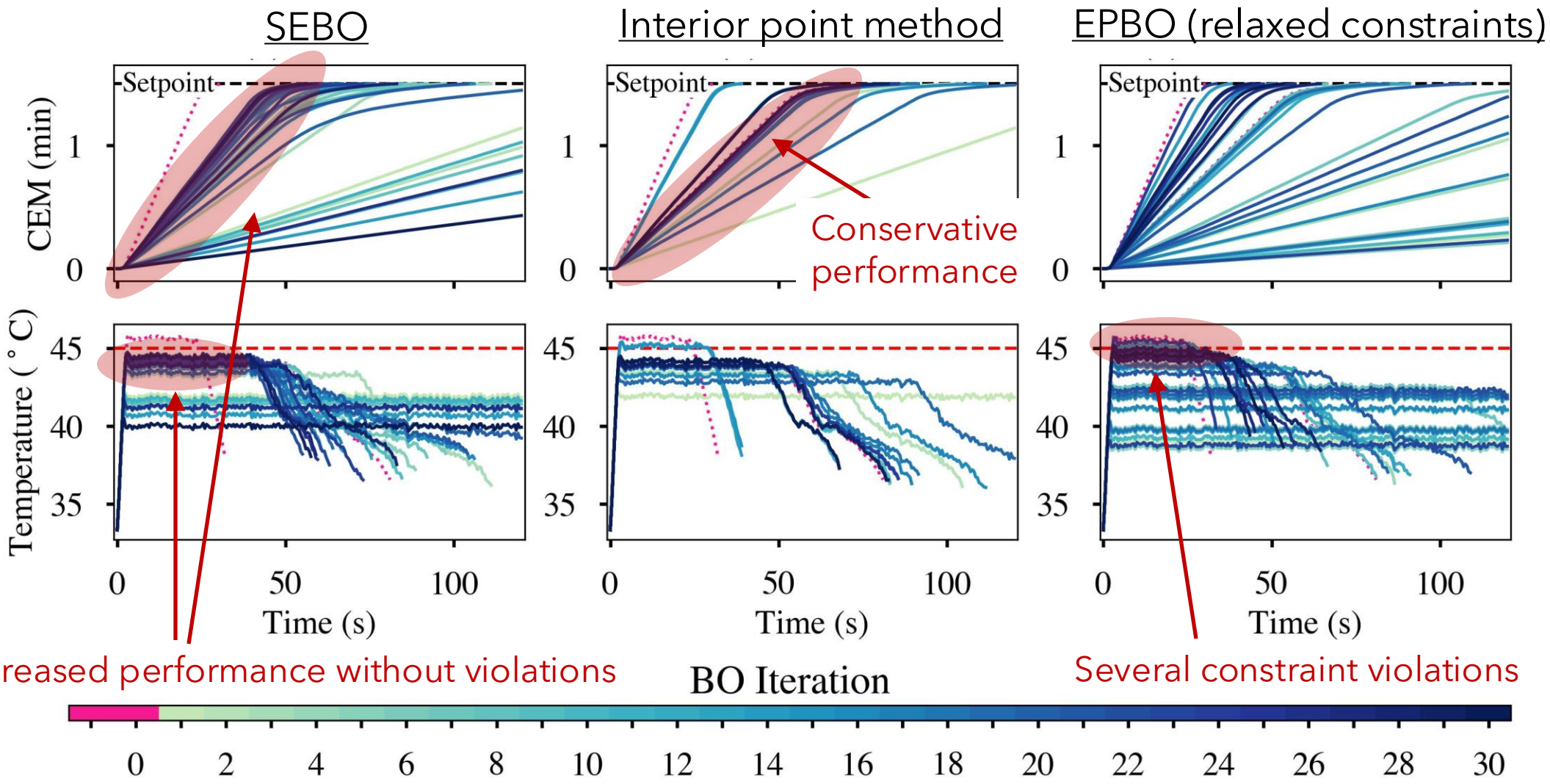
- SafeOpt [Berkenkamp et al., *Machine Learning*, 2023]
  - Uses confidence bounds to select most uncertain point between potential maximizers and potential expanders (a bit challenging to maximize acquisition function)

- Interior point methods [Krishnamoorthy and Doyle, *IEEE CSL*, 2022]
  - Uses confidence bounds with standard expected improvement acquisition function (simpler to implement, but does not guarantee growing safety set)

- Safe explorative Bayesian optimization (SEBO) [Chan, Paulson, and Mesbah, *CDC*, 2023]
  - Extends interior point approach to incentivize growing the safety set

- Information-theoretic exploration (ISE) [Bottero et al., *NeurIPS*, 2022]
  - Takes a max entropy search perspective to safe optimization

# Results: Delivered dose (CEM) and temperature safety constraints versus number of trials for different safe BO variants




[Chan, Paulson, and Mesbah, *CDC*, 2023]

# Outline

- What is standard Bayesian optimization missing?
  - Strong priors, known structure, safety considerations, uncertainty
- Beyond sequential & single objective problems
  - Parallel evaluations (synchronous & asynchronous), multi-objective optimization
- Beyond the black-box problem structure
  - Composites, function networks, multi-fidelity representations
- Beyond nominal optimization
  - Adversarial uncertainty, flexibility analysis, safety
- What is next?
  - Meta learning, preference learning, new ways to optimize acquisition, planning

# What is next?

- BayesOpt is optimal data acquisition strategy under certain assumptions
  - Gaussian process surrogate model is optimally calibrated model
  - Acquisition function derived from the true desired loss function $l(\mathcal{D}_n)$
  - We globally maximize the acquisition function when selecting next sample
  - We only have one remaining step to gather new data
- One could argue **none** of these assumptions are satisfied in practice
  - How to identify best possible model? What if we do not have strong prior?
  - Do we really know right acquisition function? What about preferences?
  - Use heuristics to maximize acquisition. What is impact on performance?
  - What if multiple steps remaining? Can we get away with greedy strategy?

[Paulson and Tsay, https://arxiv.org/pdf/2401.16373, 2024]

# What is next?

- BayesOpt is optimal data acquisition strategy under certain assumptions:
  - Gaussian process surrogate model is optimally calibrated model
  - Acquisition function derived from the true desired loss function $l(\mathcal{D}_n)$
  - We globally maximize the acquisition function when selecting next sample
  - We only have one remaining step to gather new data

- One could argue **none** of these assumptions are satisfied in practice:

**Meta learning**

[Volp et al., 2020] [Wang et al., 2022]
[Schur et al., 2023]

  - Do we really know right acquisition function? What about preferences?
  - Use heuristics to maximize acquisition. What is impact on performance?
  - What if multiple steps remaining? Can we get away with greedy strategy?

[Paulson and Tsay, https://arxiv.org/pdf/2401.16373, 2024]

# What is next?

- BayesOpt is optimal data acquisition strategy under certain assumptions:
  - Gaussian process surrogate model is optimally calibrated model
  - Acquisition function derived from the true desired loss function $l(\mathcal{D}_n)$
  - We globally maximize the acquisition function when selecting next sample
  - We only have one remaining step to gather new data

- One could argue **none** of these assumptions are satisfied in practice:

**Meta learning**

[Volp et al., 2020] [Wang et al., 2022]
[Schur et al., 2023]

**Preference learning**

[Bemporad and Piga, 2021]
[Lin et al., 2022]

  - Use heuristics to maximize acquisition. What is impact on performance?
  - What if multiple steps remaining? Can we get away with greedy strategy?

[Paulson and Tsay, https://arxiv.org/pdf/2401.16373, 2024]

# What is next?

- BayesOpt is optimal data acquisition strategy under certain assumptions:
  - Gaussian process surrogate model is optimally calibrated model
  - Acquisition function derived from the true desired loss function $l(\mathcal{D}_n)$
  - We globally maximize the acquisition function when selecting next sample
  - We only have one remaining step to gather new data

- One could argue **none** of these assumptions are satisfied in practice:

**Meta learning**
[Volp et al., 2020] [Wang et al., 2022]
[Schur et al., 2023]

**Preference learning**
[Bemporad and Piga, 2021]
[Lin et al., 2022]

**New optimization algorithms**
[Schweidtmann et al.,, 2021]
[Ament et al., 2023]

  - What if multiple steps remaining? Can we get away with greedy strategy?

[Paulson and Tsay, https://arxiv.org/pdf/2401.16373, 2024]

# What is next?

- BayesOpt is optimal data acquisition strategy under certain assumptions:
  - Gaussian process surrogate model is optimally calibrated model
  - Acquisition function derived from the true desired loss function $l(\mathcal{D}_n)$
  - We globally maximize the acquisition function when selecting next sample
  - We only have one remaining step to gather new data

- One could argue **none** of these assumptions are satisfied in practice:

**Meta learning**
[Volp et al., 2020] [Wang et al., 2022]
[Schur et al., 2023]

**Preference learning**
[Bemporad and Piga, 2021]
[Lin et al., 2022]

**New optimization algorithms**
[Schweidtmann et al.,, 2021]
[Ament et al., 2023]

**Planning under uncertainty**
[Lam et al., 2017] [Jiang et al., 2020]
[Wu et al., 2021] [Paulson et al., 2022]

[Paulson and Tsay, https://arxiv.org/pdf/2401.16373, 2024]

# What is next?

- BayesOpt is optimal data acquisition strategy under certain assumptions:
  - Gaussian process surrogate model is optimally calibrated model
  - Acquisition function derived from the true desired loss function $l(\mathcal{D}_n)$
  - We globally maximize the acquisition function when selecting next sample
  - We only have one remaining step to gather new data

- One could argue **none** of these assumptions are satisfied in practice:

**Meta learning**
[Volp et al., 2020] [Wang et al., 2022]
[Schur et al., 2023]

**Preference learning**
[Bemporad and Piga, 2021]
[Lin et al., 2022]

**New optimization algorithms**
[Schweidtmann et al.,, 2021]
[Ament et al., 2023]

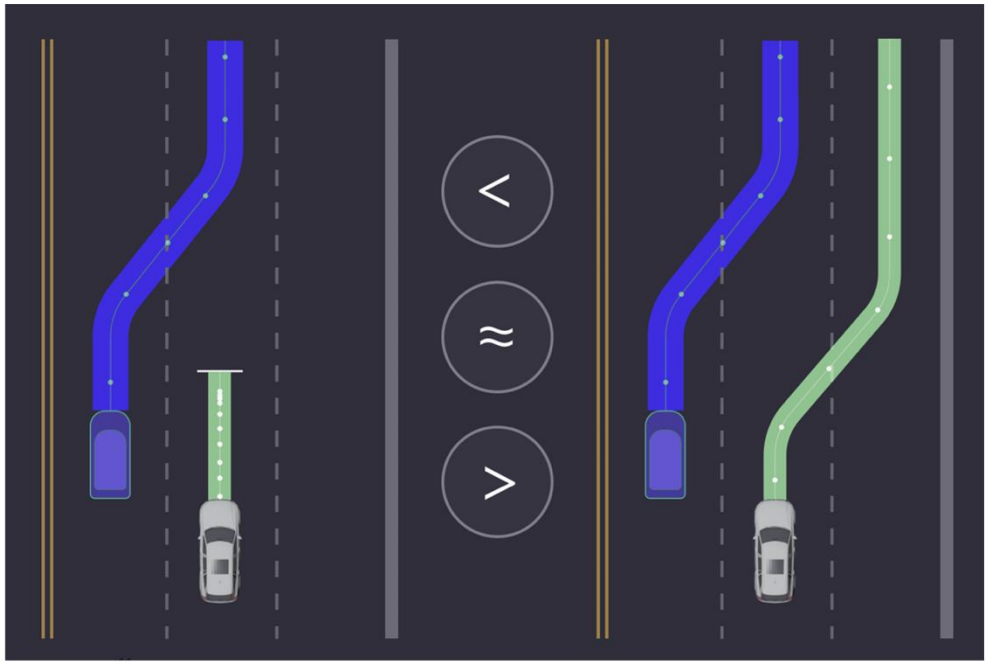
**Planning under uncertainty**
[Lam et al., 2017] [Jiang et al., 2020]
[Wu et al., 2021] [Paulson et al., 2022]

[Paulson and Tsay, https://arxiv.org/pdf/2401.16373, 2024]

# Problem Setting: Optimization given Preferences

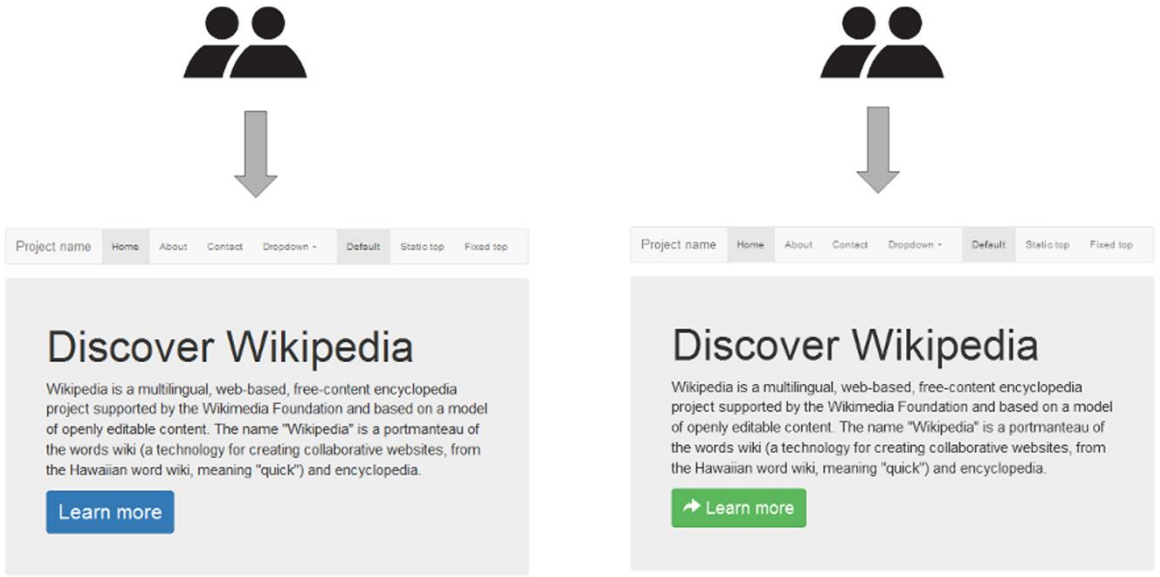
- Defining metrics amenable to optimization can be quite challenging in complex engineering systems (even when considering multiple objectives)
  - Comparison is almost always easier than rating for humans

Example: Path planning




[Dewancker et al., *NeurIPS Workshop*, 2017]

Example: Website design




Click rate: 52%

Click rate: 72%

[Gonzalez et al., *ICML*, 2019]

# Problem Setting: Optimization given Preferences

- Defining metrics amenable to optimization can be quite challenging in complex engineering systems (even when considering multiple objectives)
  - Comparison is almost always easier than rating for humans

- We can mathematically think of this problem as minimizing a latent function $\min_{x \in \Omega} g(x)$ but under condition that we cannot measure $g(x)$ directly at all
  - Can only infer information about $g$ through preference data
  - For example, might only observe if $g(x) < g(x')$ is true or not → outcome of duel $[x, x']$

- There has been interesting recent work on preferential Bayesian optimization, but far from solved in my opinion
  - Should we model a preference function, the latent function, or both?
  - Can / how might we use these ideas to manage the tradeoff between exploration and exploitation in our search algorithms?

# What is next?

- BayesOpt is optimal data acquisition strategy under certain assumptions:
  - Gaussian process surrogate model is optimally calibrated model
  - Acquisition function derived from the true desired loss function $l(\mathcal{D}_n)$
  - We globally maximize the acquisition function when selecting next sample
  - We only have one remaining step to gather new data

- One could argue **none** of these assumptions are satisfied in practice:

**Meta learning**
[Volp et al., 2020] [Wang et al., 2022]
[Schur et al., 2023]

**Preference learning**
[Bemporad and Piga, 2021]
[Lin et al., 2022]

**New optimization algorithms**
[Schweidtmann et al.,, 2021]
[Ament et al., 2023]

**Planning under uncertainty**
[Lam et al., 2017] [Jiang et al., 2020]
[Wu et al., 2021] [Paulson et al., 2022]

[Paulson and Tsay, https://arxiv.org/pdf/2401.16373, 2024]

# Better optimization of acquisition function is important!

## Unexpected Improvements to Expected Improvement for Bayesian Optimization


**Sebastian Ament**
Meta
ament@meta.com

**Samuel Daulton**
Meta
sdaulton@meta.com

**David Eriksson**
Meta
deriksson@meta.com

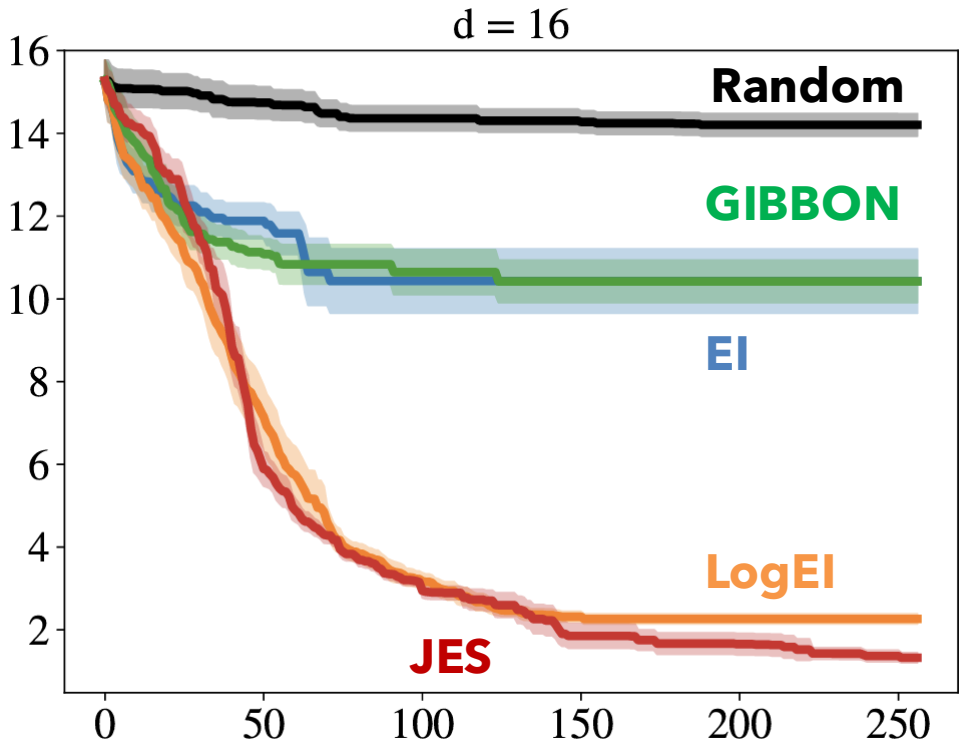
**Maximilian Balandat**
Meta
balandat@meta.com

**Eytan Bakshy**
Meta
ebakshy@meta.com



Expected Improvement (EI) is arguably the most popular acquisition function in Bayesian optimization and has found countless successful applications, but its performance is often exceeded by that of more recent methods. Notably, EI and its variants, including for the parallel and multi-objective settings, are challenging to optimize because their acquisition values vanish numerically in many regions. This difficulty generally increases as the number of observations, dimensionality of the search space, or the number of constraints grow, resulting in performance that is inconsistent across the literature and most often sub-optimal. Herein, we propose LogEI, a new family of acquisition functions whose members either have identical or approximately equal optima as their canonical counterparts, but are substantially easier to optimize numerically. We demonstrate that numerical pathologies manifest themselves in "classic" analytic EI, Expected Hypervolume Improvement (EHVI), as well as their constrained, noisy, and parallel variants, and propose corresponding reformulations that remedy these pathologies. Our empirical results show that members of the LogEI family of acquisition functions substantially improve on the optimization performance of their canonical counterparts and surprisingly, are on par with or exceed the performance of recent state-of-the-art acquisition functions, highlighting the understated role of numerical optimization in the literature.


Results on Ackley function




[Ament et al., *NeurIPS*, 2023]

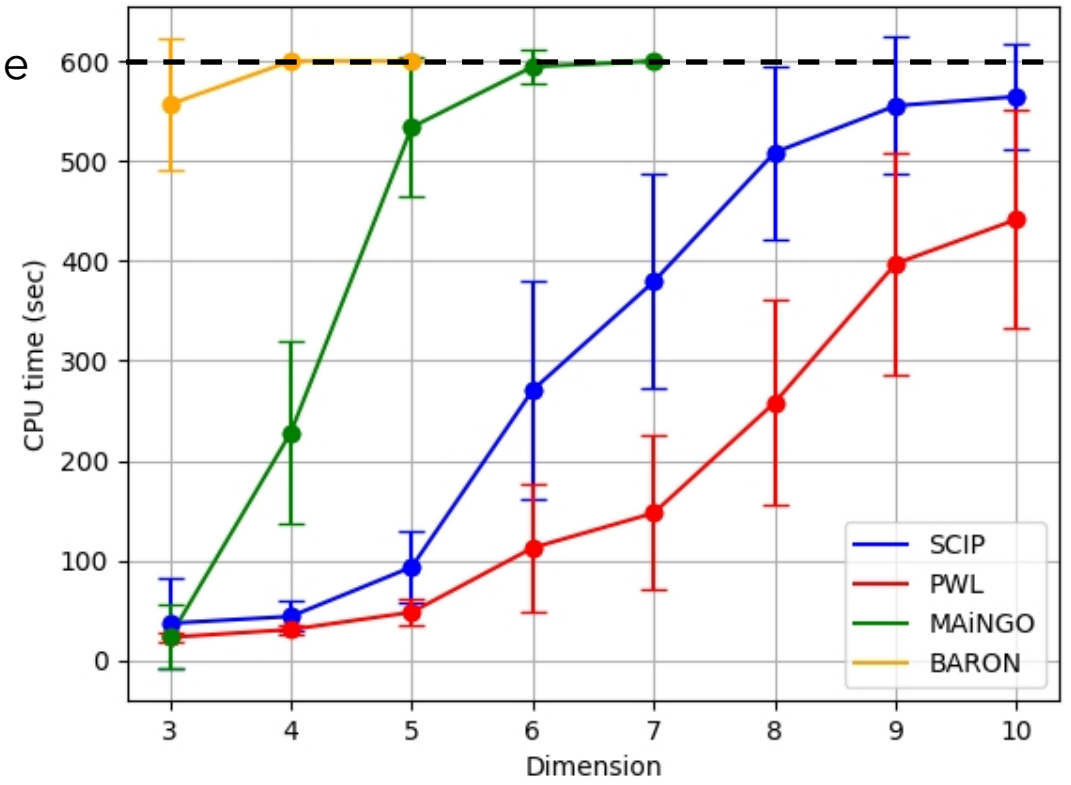
# Can we provide an exact guarantee of global optimality?

Non-convex optimization is NP-hard…but algorithms do exist (usually slow)

- We are developing tailored spatial branch-and-bound algorithms that exploit structure of GP and provably converge to global solution

**Commercial MINLP solver**

Max allowed solve time




**Custom GP optimizer**

**State-of-the-art solver**

**Our method**

**Collaboration with Calvin Tsay, Imperial College London**

# Summary

- Bayesian optimization (BO) is **efficient & flexible optimization** framework
  - meant for expensive black-box functions with possibly noisy observations
  - applicable to many problems in science, engineering, and beyond…
- BO can **automate tedious workflows**; free up humans for other tasks
  - save time and money, leading to long-term increase in productivity
- BO limited by black-box assumption; **big gains by peeking inside box**
  - grey-box philosophy: use all available information to accelerate convergence
- Lots of room for **developing new & better algorithms** for grey-box BO
  - theory/algorithm development is best informed from applications; need more!

# Thanks for your attention

# Questions?