Bayesian Optimization

Why & How?

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

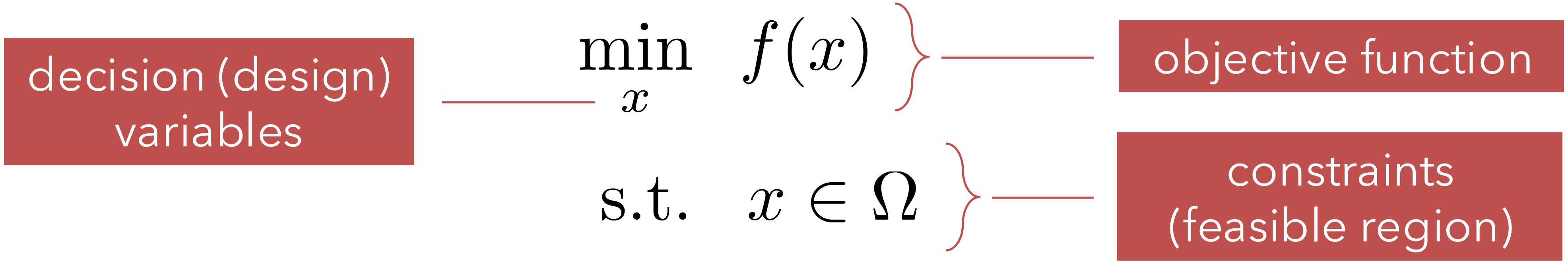
Outline

- Introduction to Bayesian optimization
  - White-box vs. black-box, prevalence of expensive functions, bird's eye view
- Improvement-based acquisition functions
  - Expected improvement, knowledge gradient
- Information-theoretic acquisition functions
  - Predictive, max-value, and joint entropy search
- Constrained Bayesian optimization
  - Expected improvement with constraints, exact penalty methods
- Practical considerations
  - Optimizing the acquisition function, kernel adaptation, acquisition scheduling

Outline

- Introduction to Bayesian optimization
  - White-box vs. black-box, prevalence of expensive functions, bird's eye view
- Improvement-based acquisition functions
  - Expected improvement, knowledge gradient
- Information-theoretic acquisition functions
  - Predictive, max-value, and joint entropy search
- Constrained Bayesian optimization
  - Expected improvement with constraints, exact penalty methods
- Practical considerations
  - Optimizing the acquisition function, kernel adaptation, acquisition scheduling

What is an Optimization Problem?



decision (design) variables

$$\min_{x} \quad f(x)$$

$$\text{s.t.} \quad x \in \Omega$$

objective function

constraints (feasible region)

- Optimization problems are **pervasive** in every application domain
  - differentiate problems based on characteristics → determine what solver to use

- There are a huge number of available optimization algorithms; difficult to *a priori* know the best one but we can eliminate some options

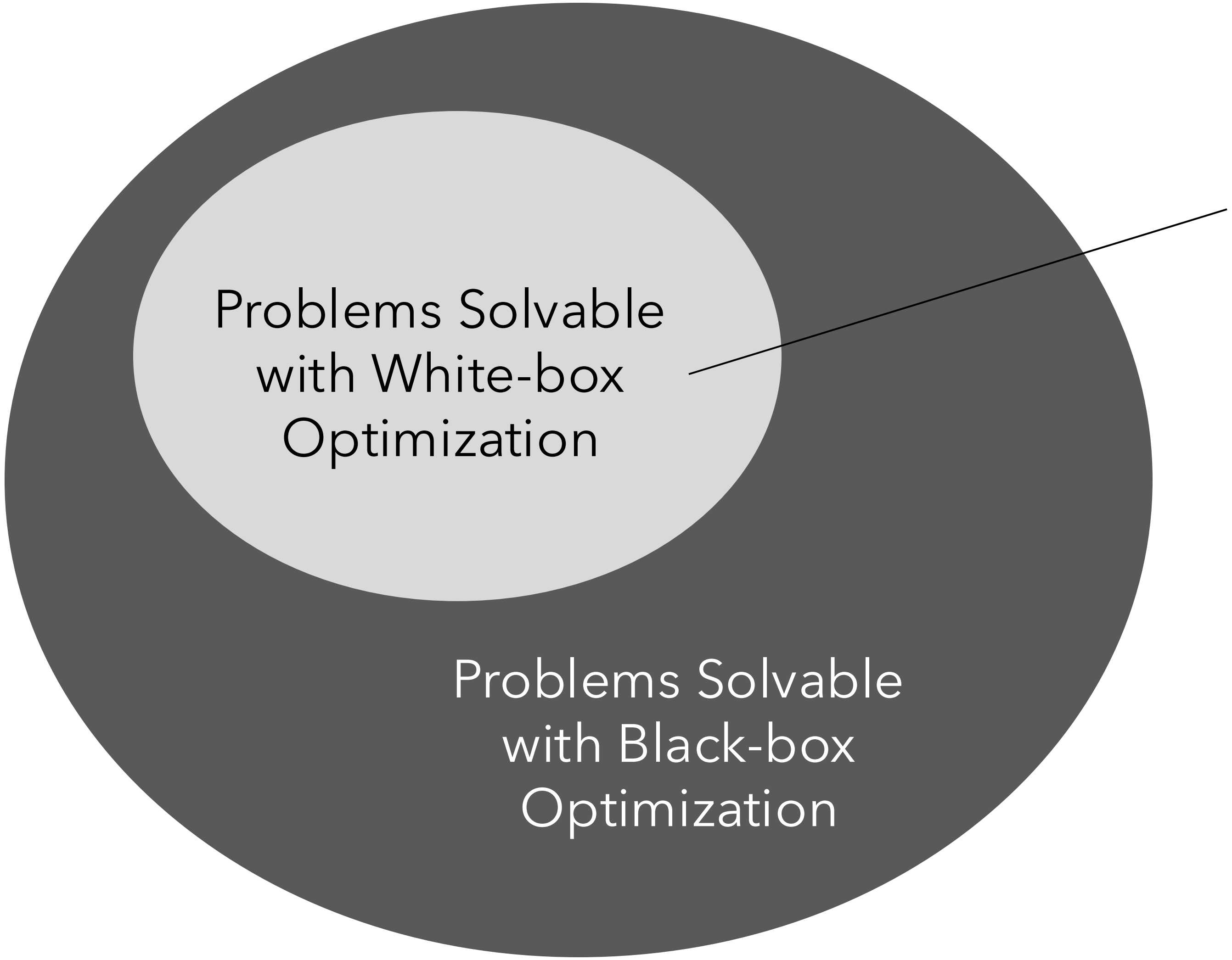
How to Classify Optimization Algorithms?

- A simple way to “partition” the algorithms into two major buckets are “white-box” and “black-box” (i.e., not white box)

- White-box means that we need an “equation-oriented model” of the system so that the mathematical structure of $f(x)$ and Ω satisfy certain important assumptions
  - The exact assumptions depend on the method, but they will typically require the functions to be differentiable and/or easy to build relaxations of them

- Any method that only requires evaluations of $f(x)$ and $x \in \Omega$ at specific points can then be classified as “black box”

How to Classify Optimization Algorithms?


Problems Solvable with White-box Optimization
Problems Solvable with Black-box Optimization

Since white-box algorithms make stronger assumptions, they can only be used to tackle a subset of problems when compared to black-box algorithms

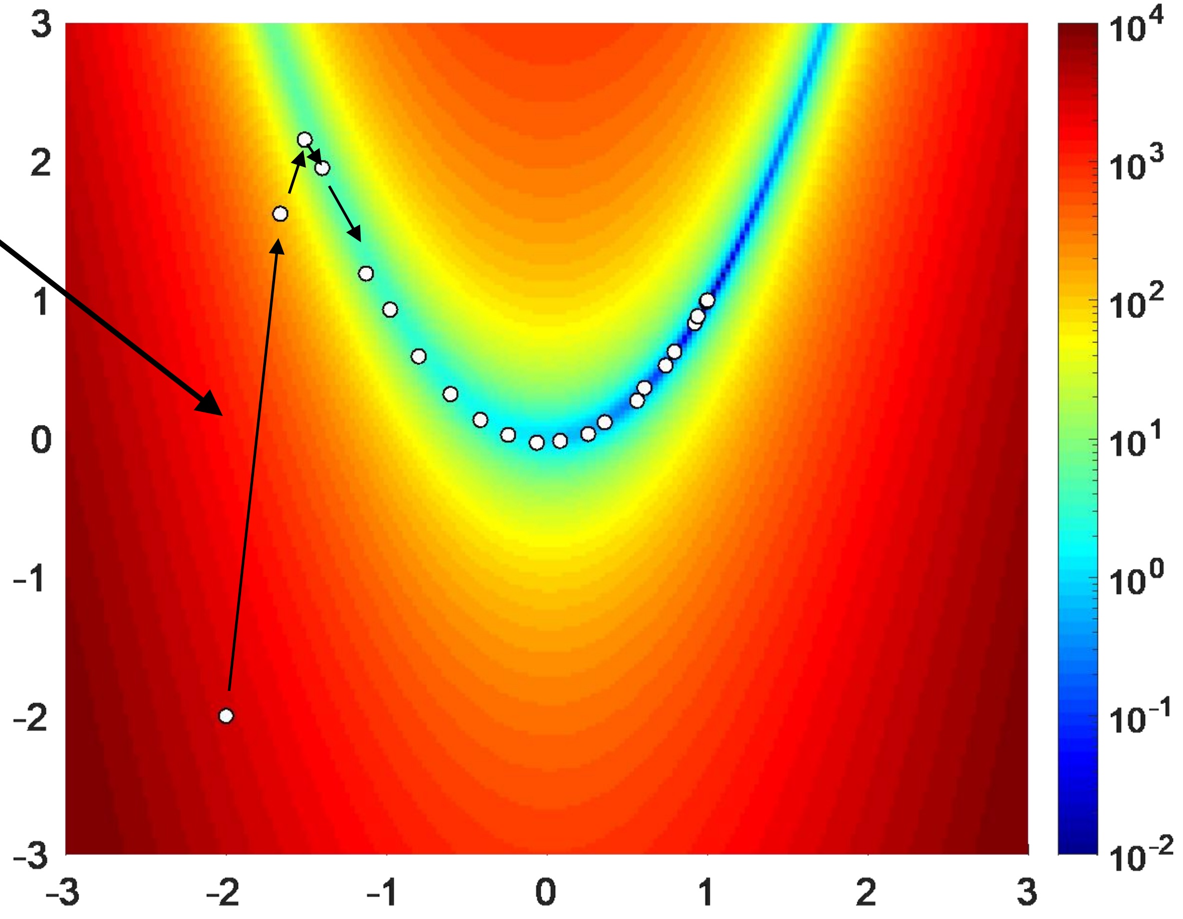
→ The main value of black-box methods are their generality (not necessarily efficient)

Example of White-Box Optimization: Newton's Method

Use derivatives to take step toward reducing objective, i.e.,

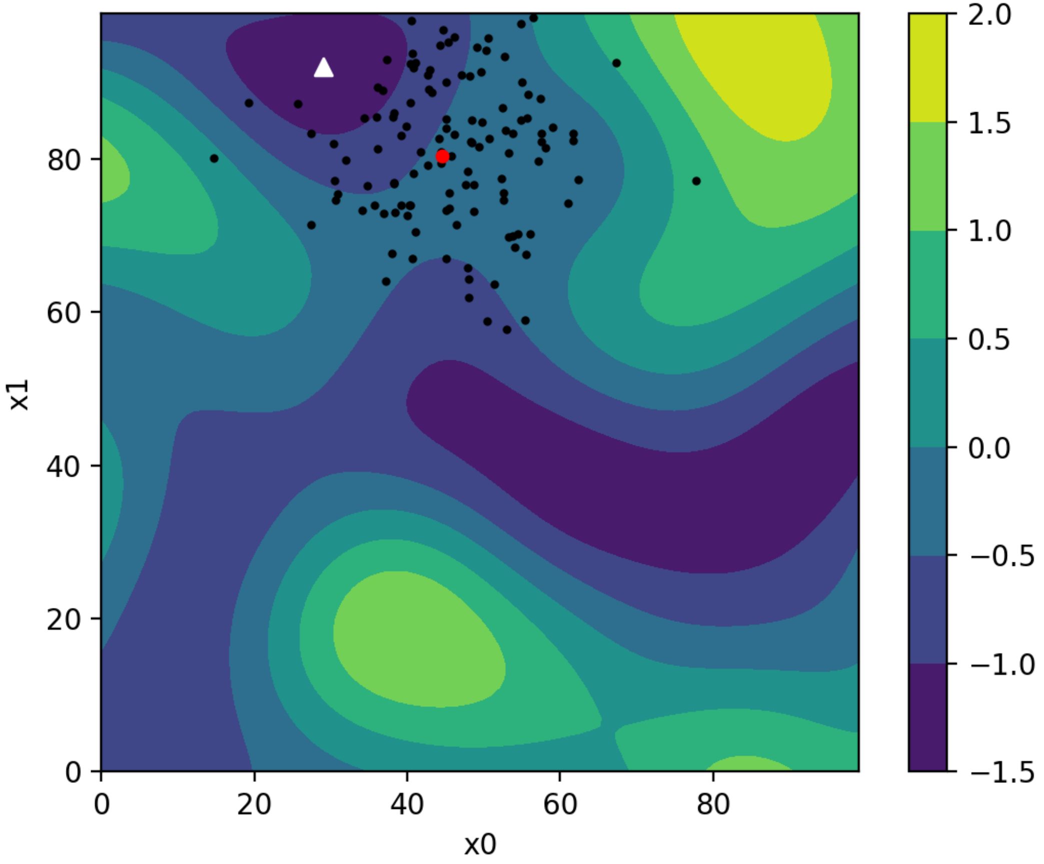
$$x_{k+1} = x_k - \alpha_k \underline{\left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)}$$

This type of algorithm is "local" (requires initial guess) & requires ability to compute derivatives (expensive when the structure of the function is unknown)



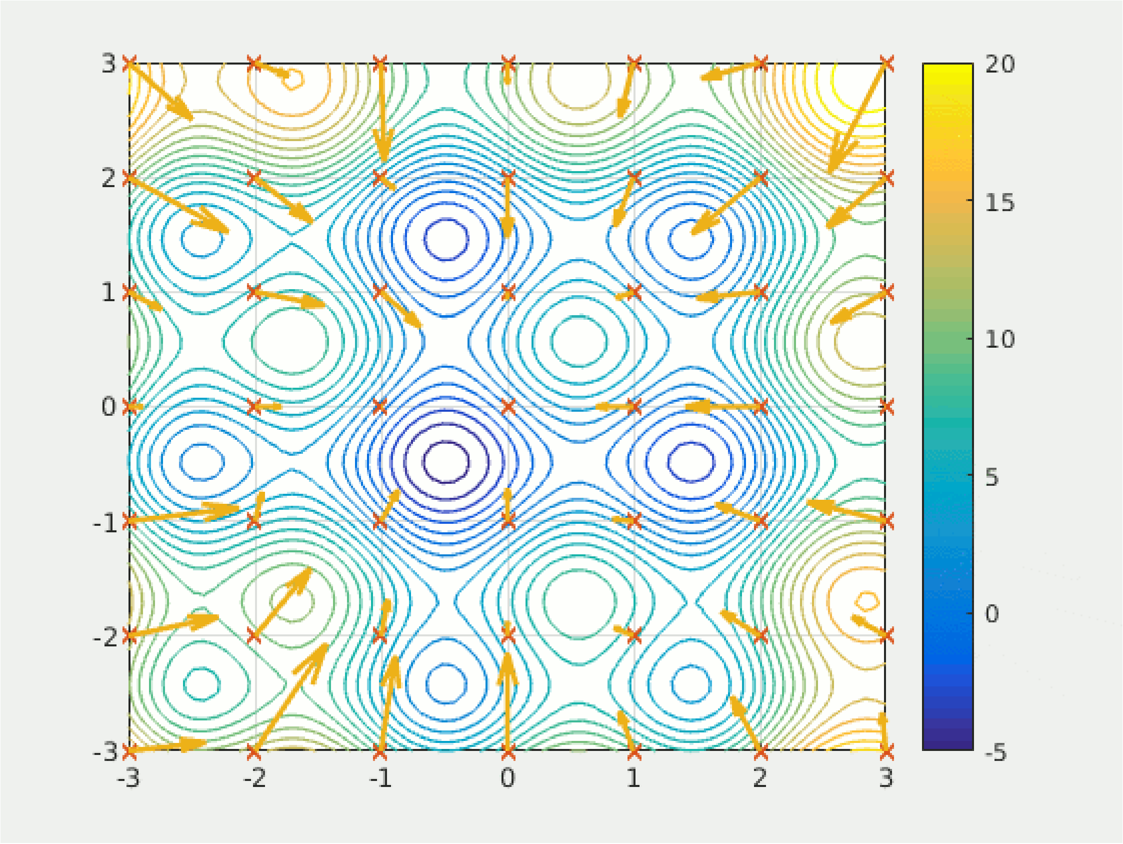
Examples of Black-Box (Derivative-Free) Optimization

Covariance Matrix Adaptive Evolutionary Strategy (CMA-ES)



https://thurinj.github.io/CMA-ES.html

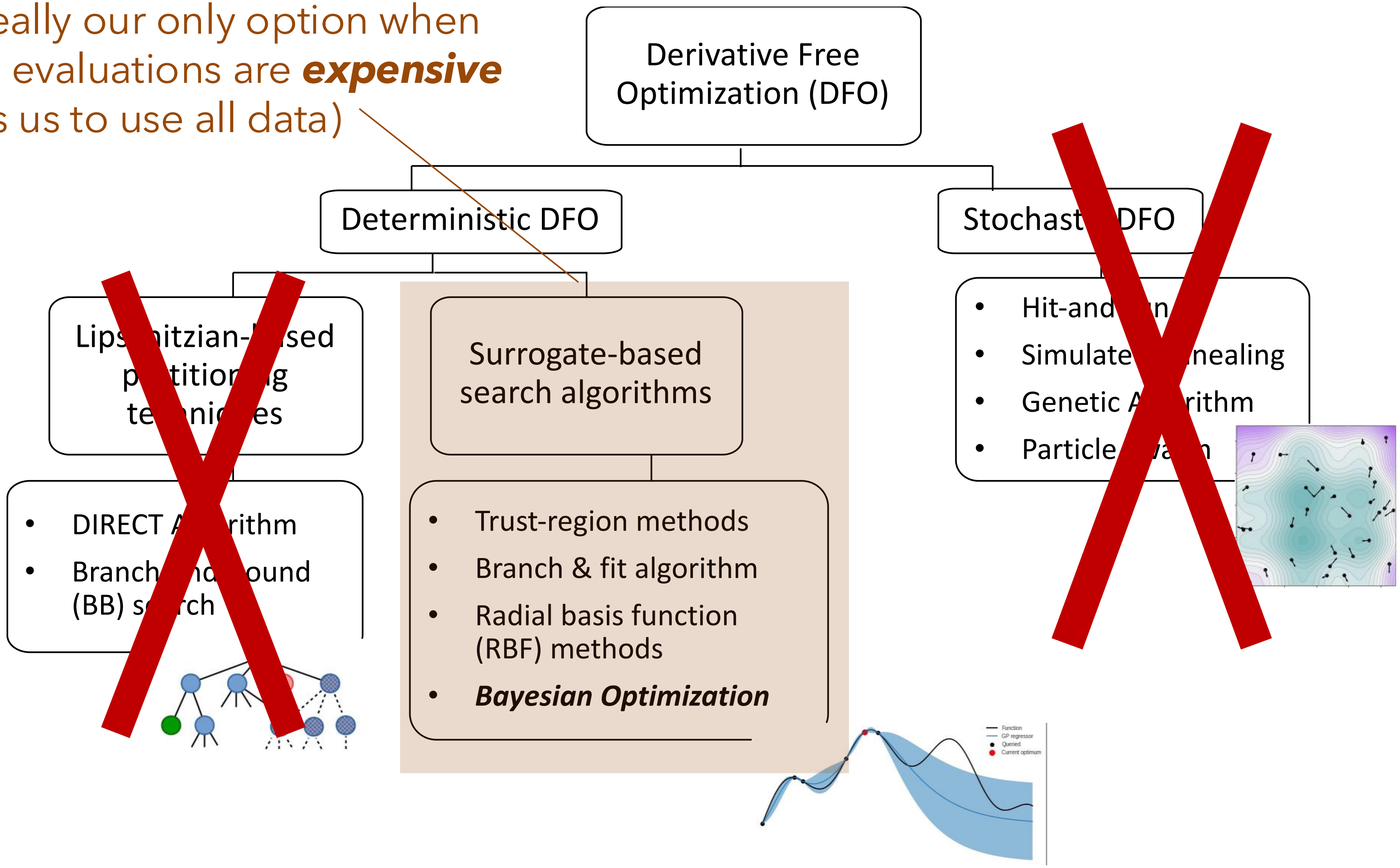
Particle Swarm Optimization (PSO)



https://en.wikipedia.org/wiki/Particle_swarm_optimization

Many derivative-free optimization methods, which to choose?

This is really our only option when function evaluations are ***expensive*** (enables us to use all data)



Derivative Free Optimization (DFO)

Deterministic DFO

~~Stochastic DFO~~

~~Lipschitzian-based partitioning techniques~~

- ~~DIRECT Algorithm~~
- ~~Branch and bound (BB) search~~

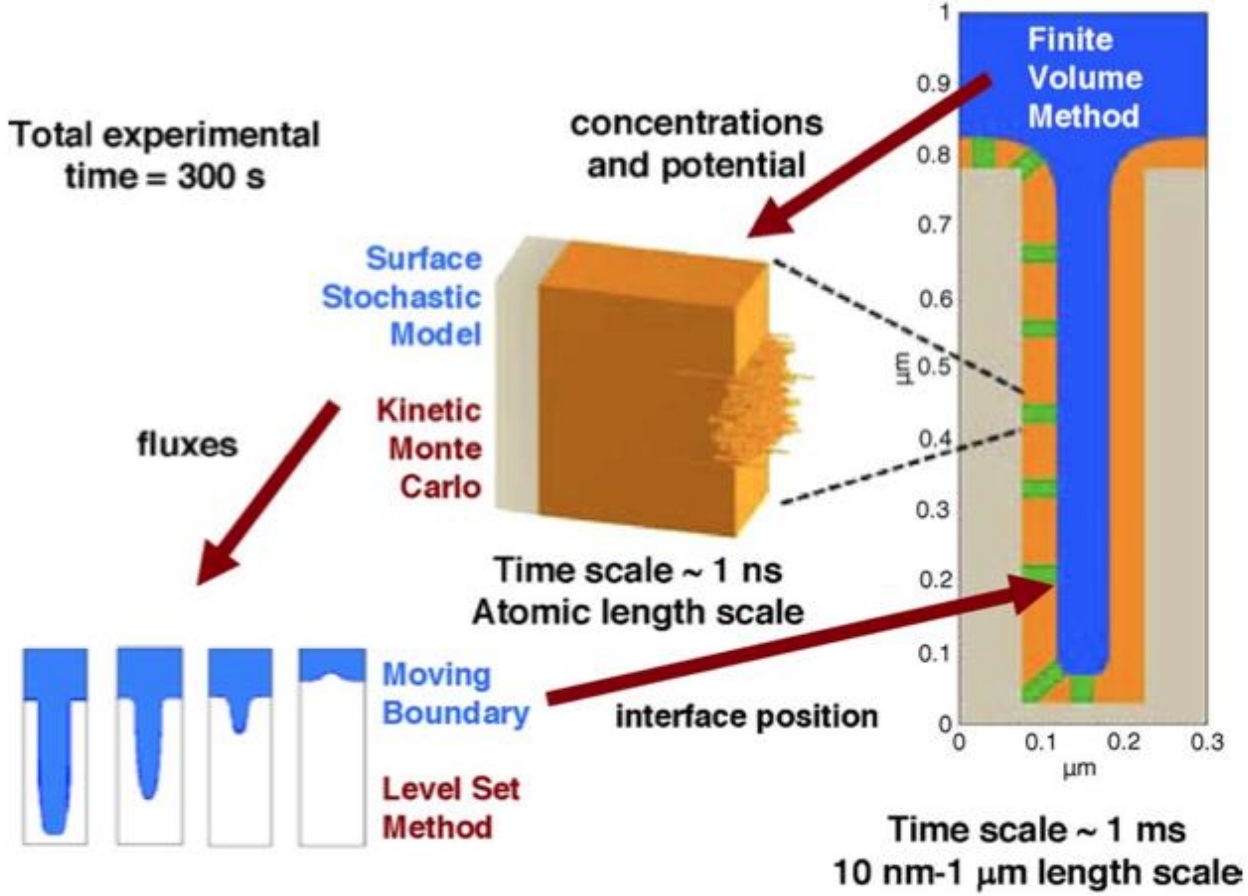
Surrogate-based search algorithms

- Trust-region methods
- Branch & fit algorithm
- Radial basis function (RBF) methods
- ***Bayesian Optimization***

- ~~Hit-and-run~~
- ~~Simulated annealing~~
- ~~Genetic Algorithm~~
- ~~Particle swarm~~

Expensive functions, they are everywhere

- **Optimizing multi-scale simulation models**



- **Objective:**

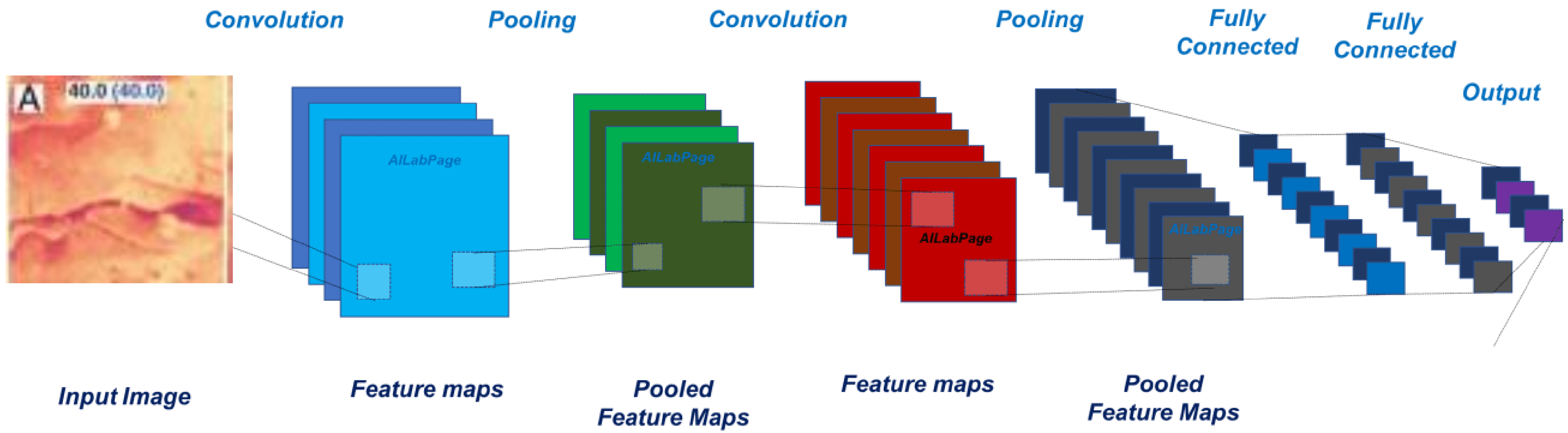
  Minimize surface roughness

- **Design variables:**

  Chemical additive concentrations & reaction temperature

Expensive functions, they are everywhere

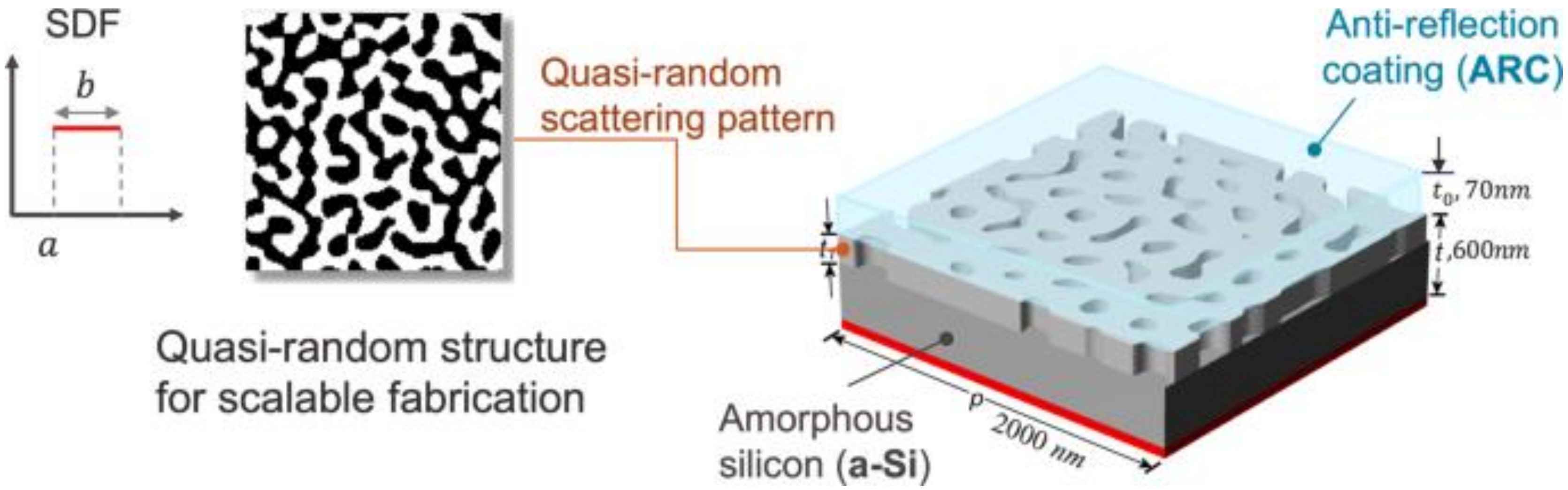
- **Automated machine learning**



- **Objective:** Maximize classification accuracy for image-based chemical sensor
- **Design variables:** Number of layers, number of nodes per layer, learning rates, regularization penalties, activation functions, etc.

Expensive functions, they are everywhere

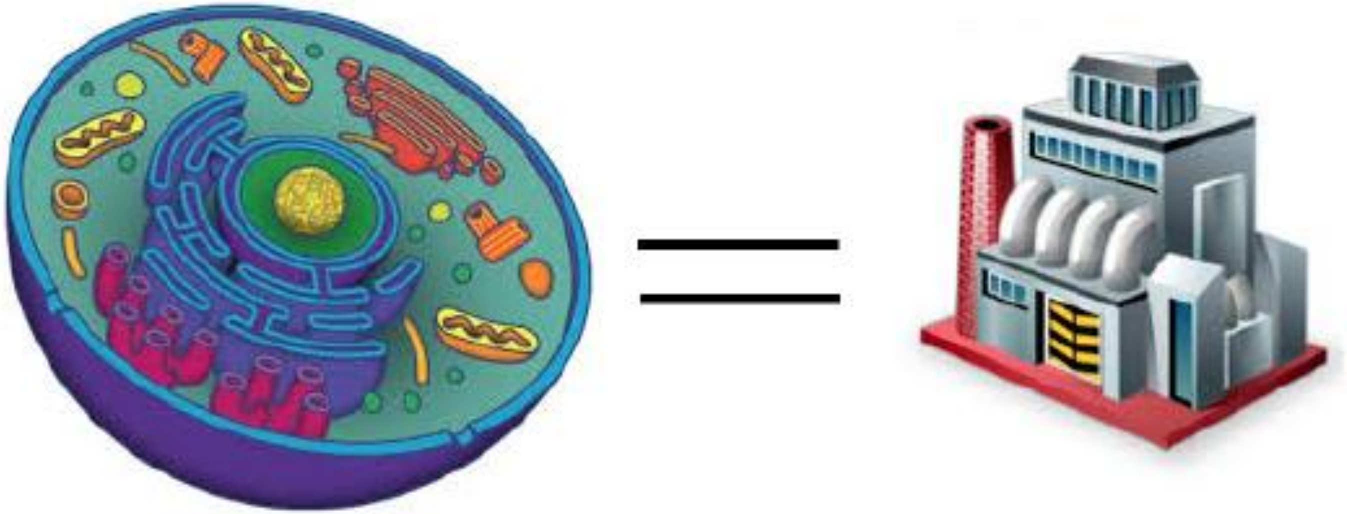
- **Material and drug discovery**


SDF
b
a
Quasi-random scattering pattern
Quasi-random structure for scalable fabrication
Anti-reflection coating (ARC)
t_0, 70nm
t,600nm
t_s
Amorphous silicon (a-Si)
p
2000 nm

- **Objective:** Maximize light adsorption in quasi-random solar cell
- **Design variables:** Type of amorphous silicon (a-Si), light trapping pattern for fabrication, & overall thickness

Expensive functions, they are everywhere

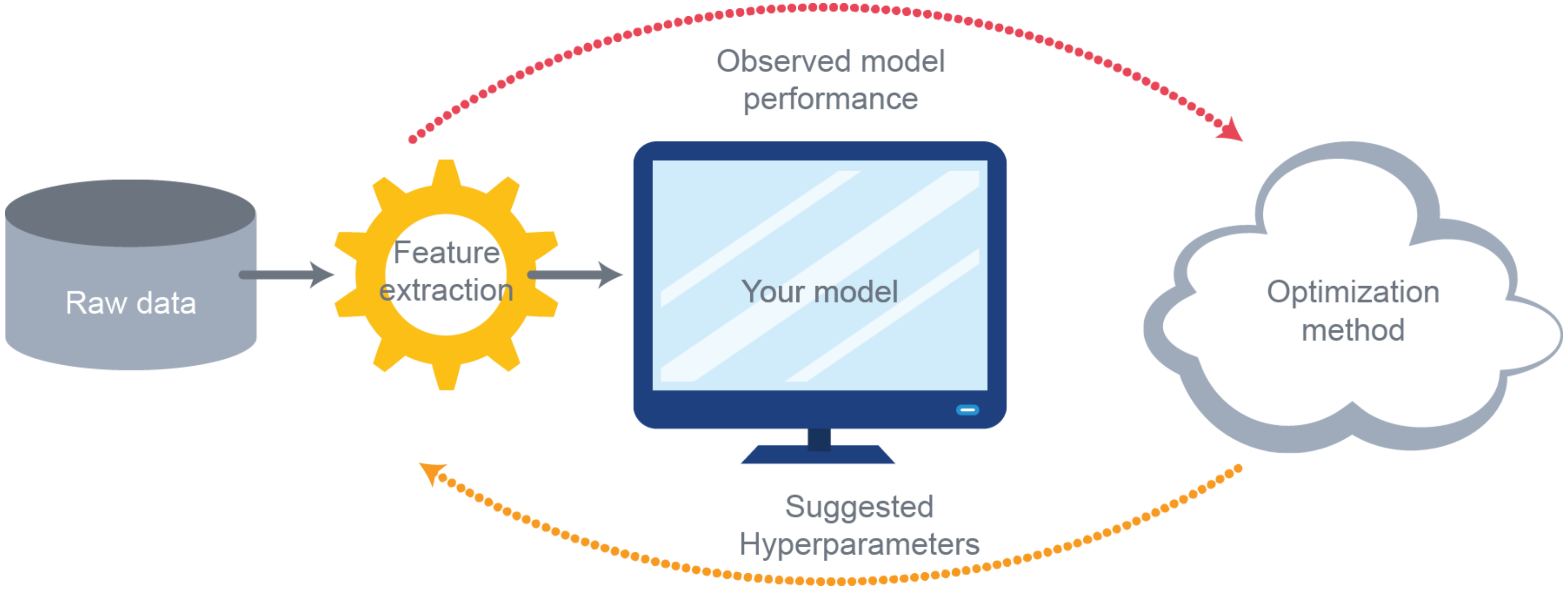
- **Design of experiments: Gene optimization**



  - **Objective:** Maximize efficiency of the cell factory to make product (e.g., proteins)
  - **Design variables:** Gene sequence (e.g., ATTGGTUGA…) & culture conditions (e.g,. pH)

Expensive functions, they are everywhere

- **Tuning hyperparameters in optimization codes**



- **Objective:** Minimize solution time for family of scheduling/planning problems
- **Design variables:** Algorithmic parameters in solver (e.g., CPLEX has 76 design parameters)

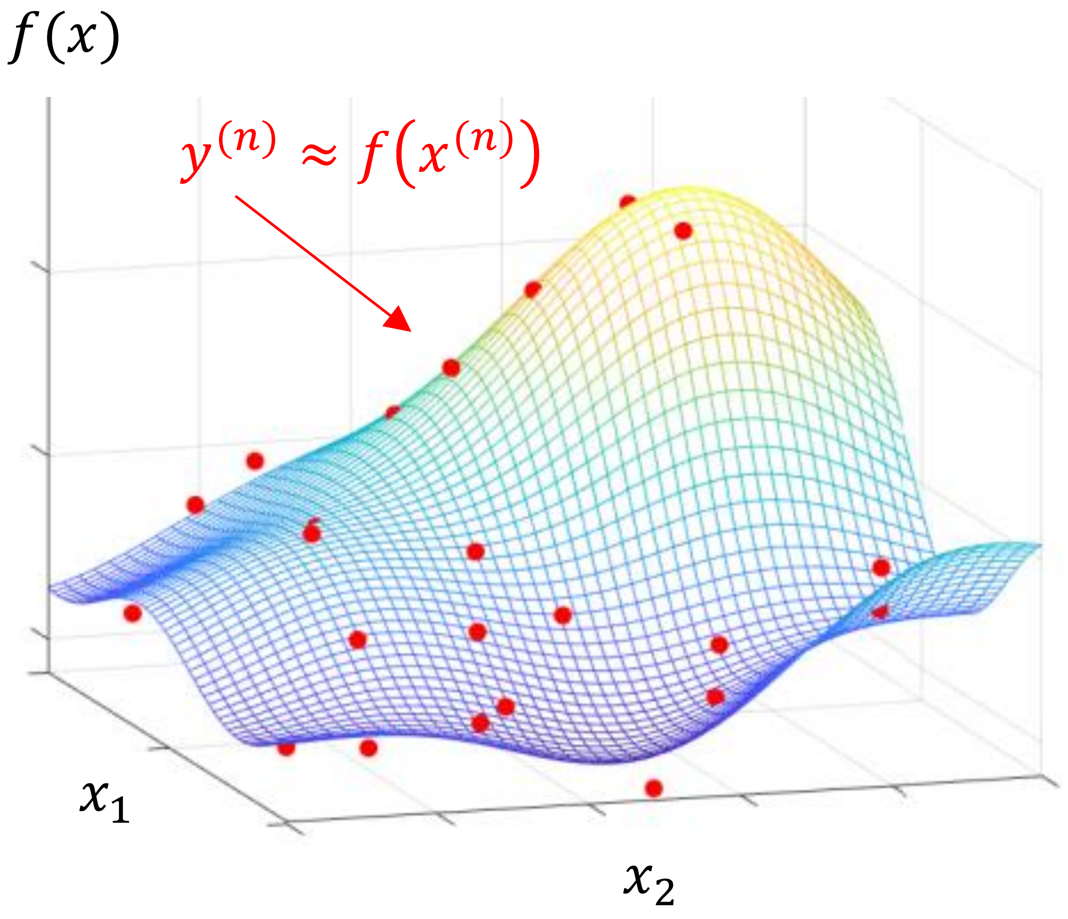
https://sigopt.com/blog/common-problems-in-hyperparameter-optimization/

Expensive functions, they are everywhere

- **Many other problems:**
    - Robotics, aerospace, control, reinforcement learning
    - Tuning websites with A/B testing
    - Calibrating expensive simulators to experimental data
    - etc.…

Standard Goal in Bayesian Optimization:

Optimize functions f : $\mathbb{R}^d \to \mathbb{R}$ that are:

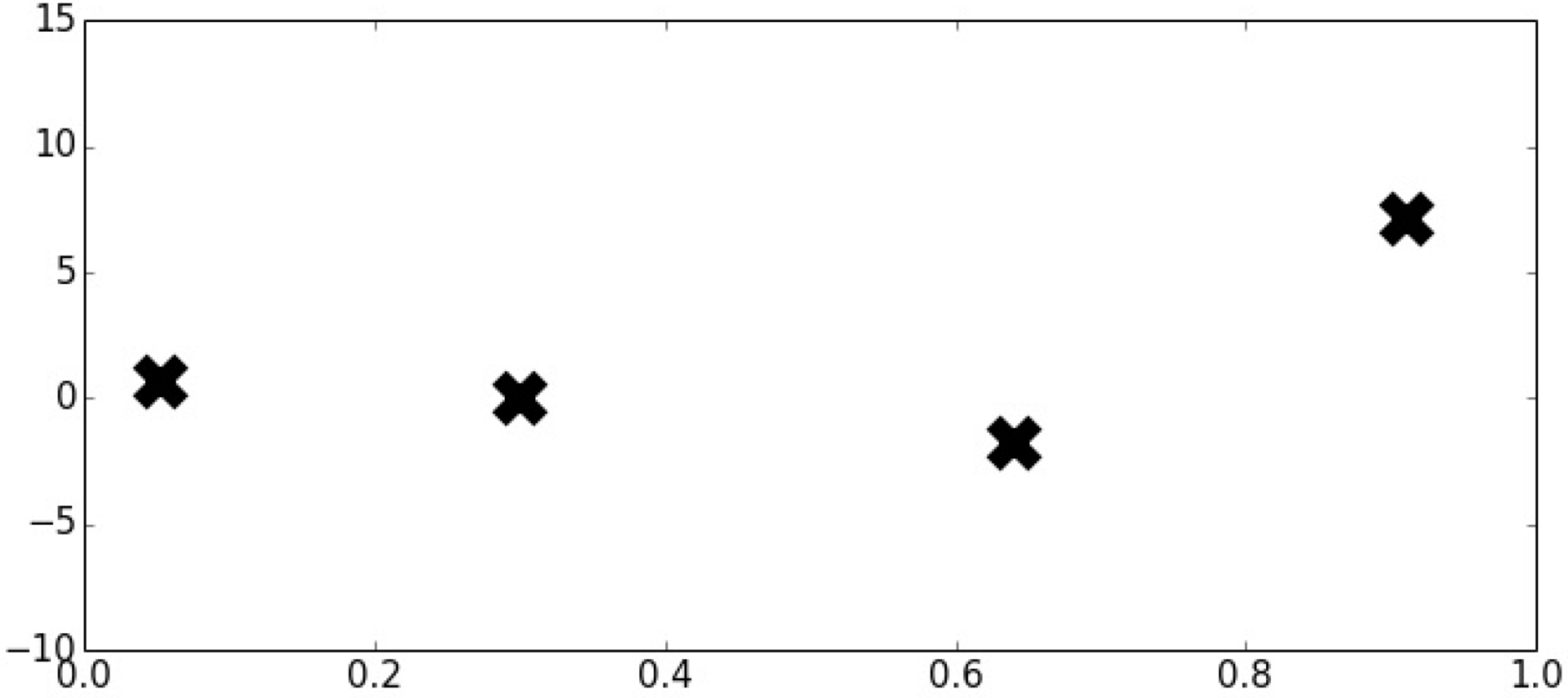


- f(·) is explicitly unknown & non-convex
  – lacks known special structure, e.g., convexity
- f(·) is derivative-free
  – cannot simply get gradients
- f(·) is expensive to evaluate
  – # of evaluations is **severely limited**
- f(·)'s evaluations may be noisy
  – noise independent & ~normally distributed, but unknown variance

*We will deal with black-box constraints later

Illustrative example to build some intuition

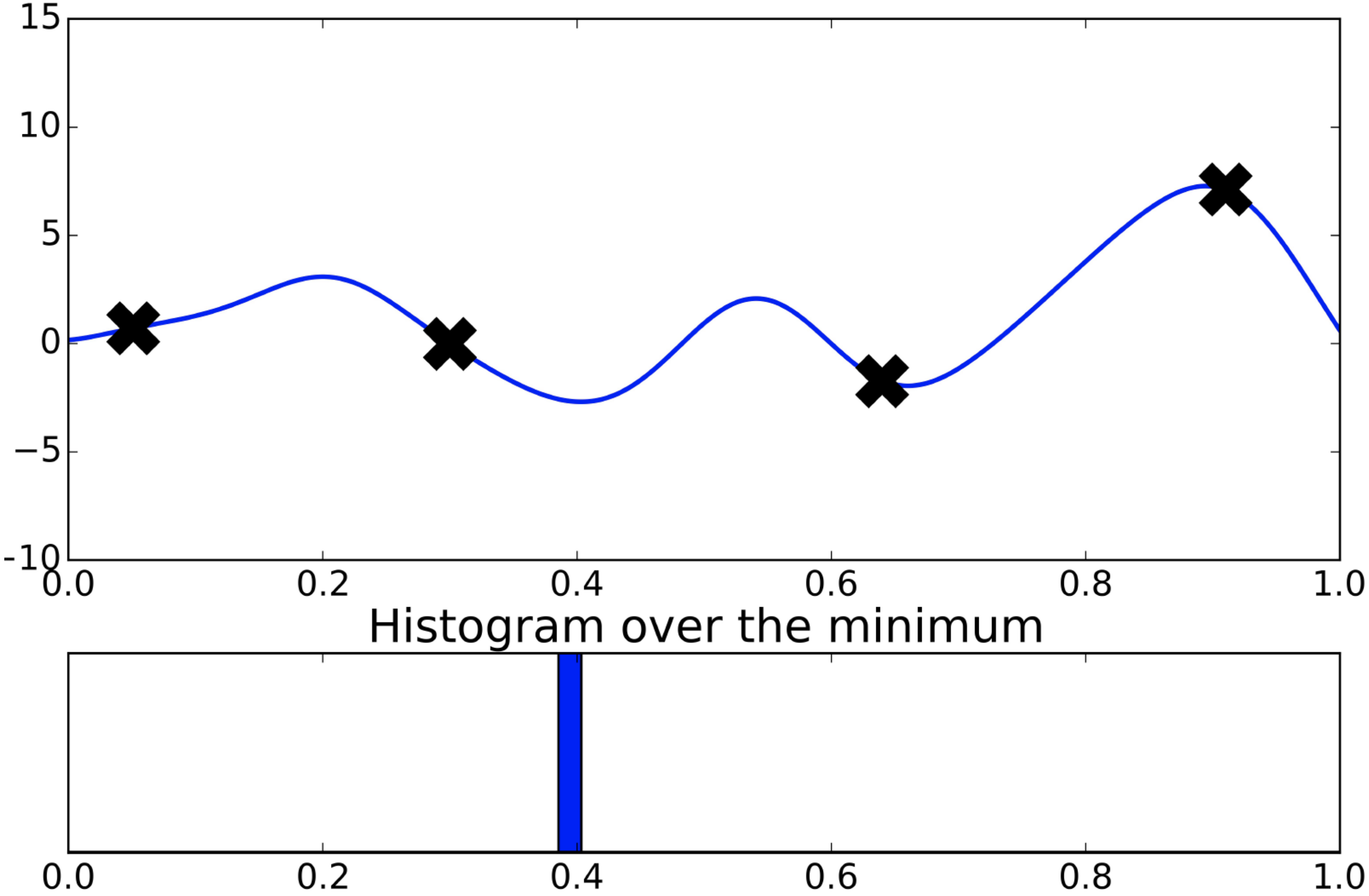
We have four function evaluations



- Where is the minimum of the function f(·)?
- Where should we take our next evaluation?

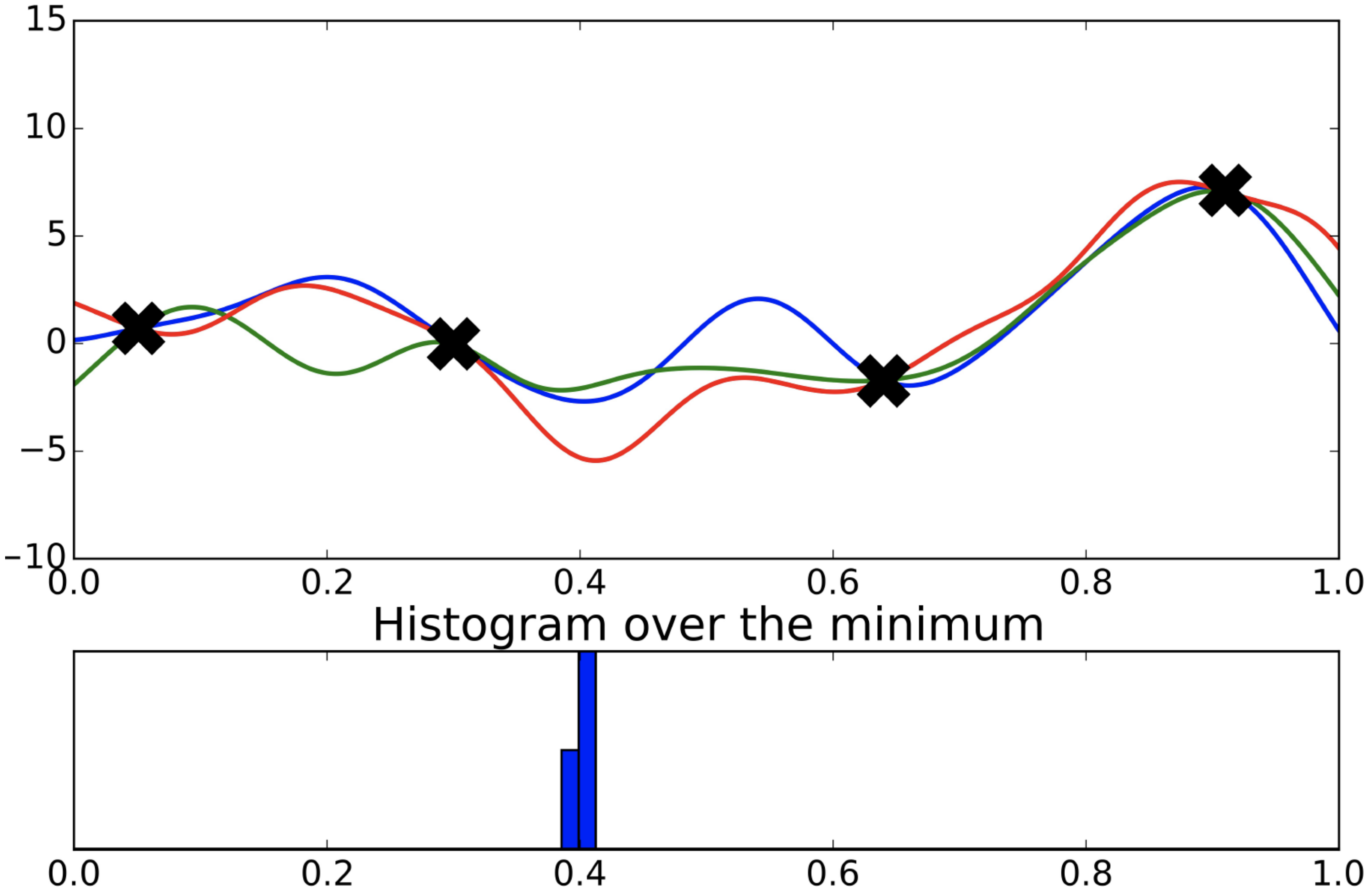
Intuitive solution, fit a surrogate model

One curve; which one should we select?



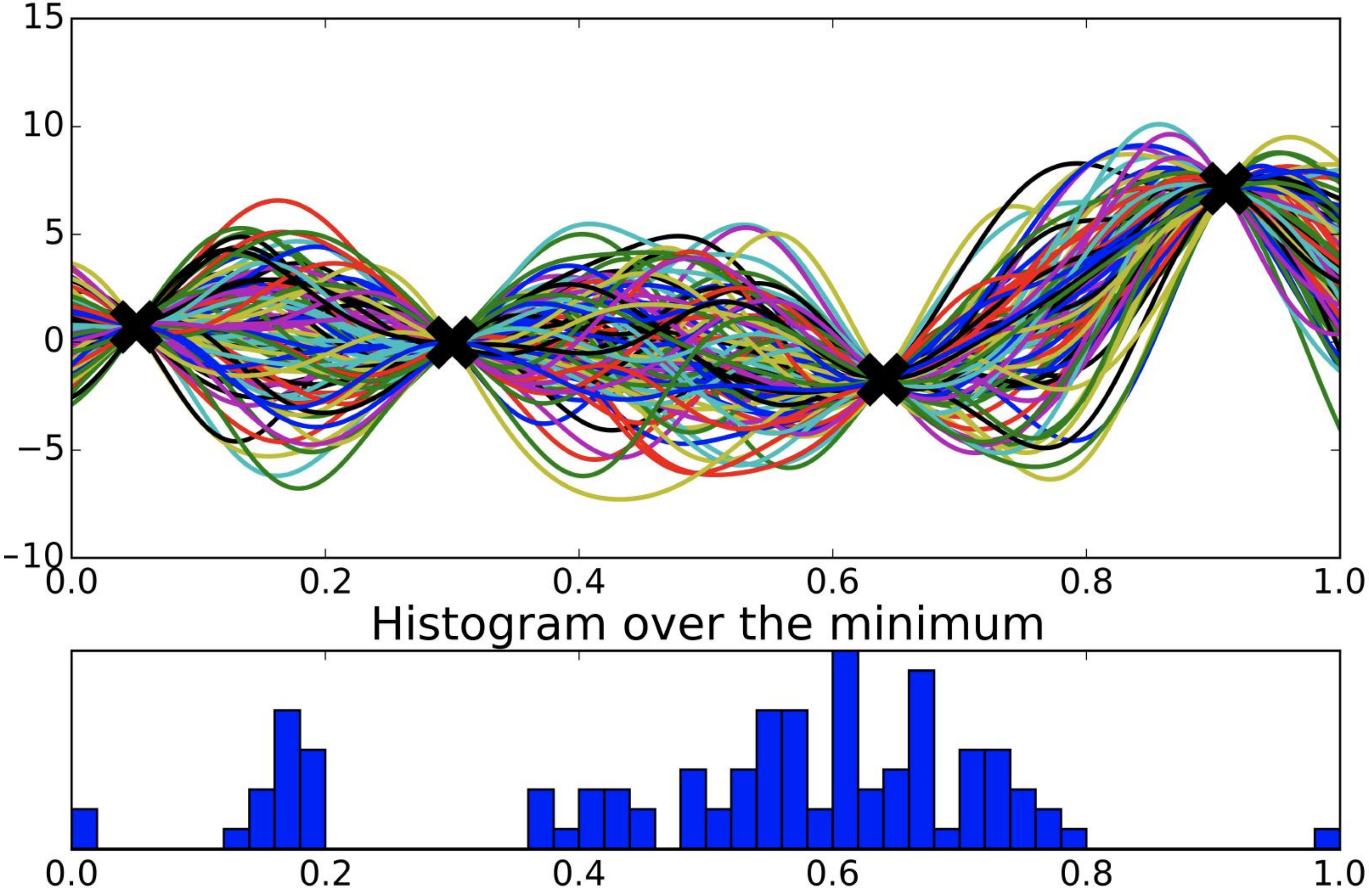
Intuitive solution, fit a surrogate model

Three curves



Intuitive solution, fit a surrogate model

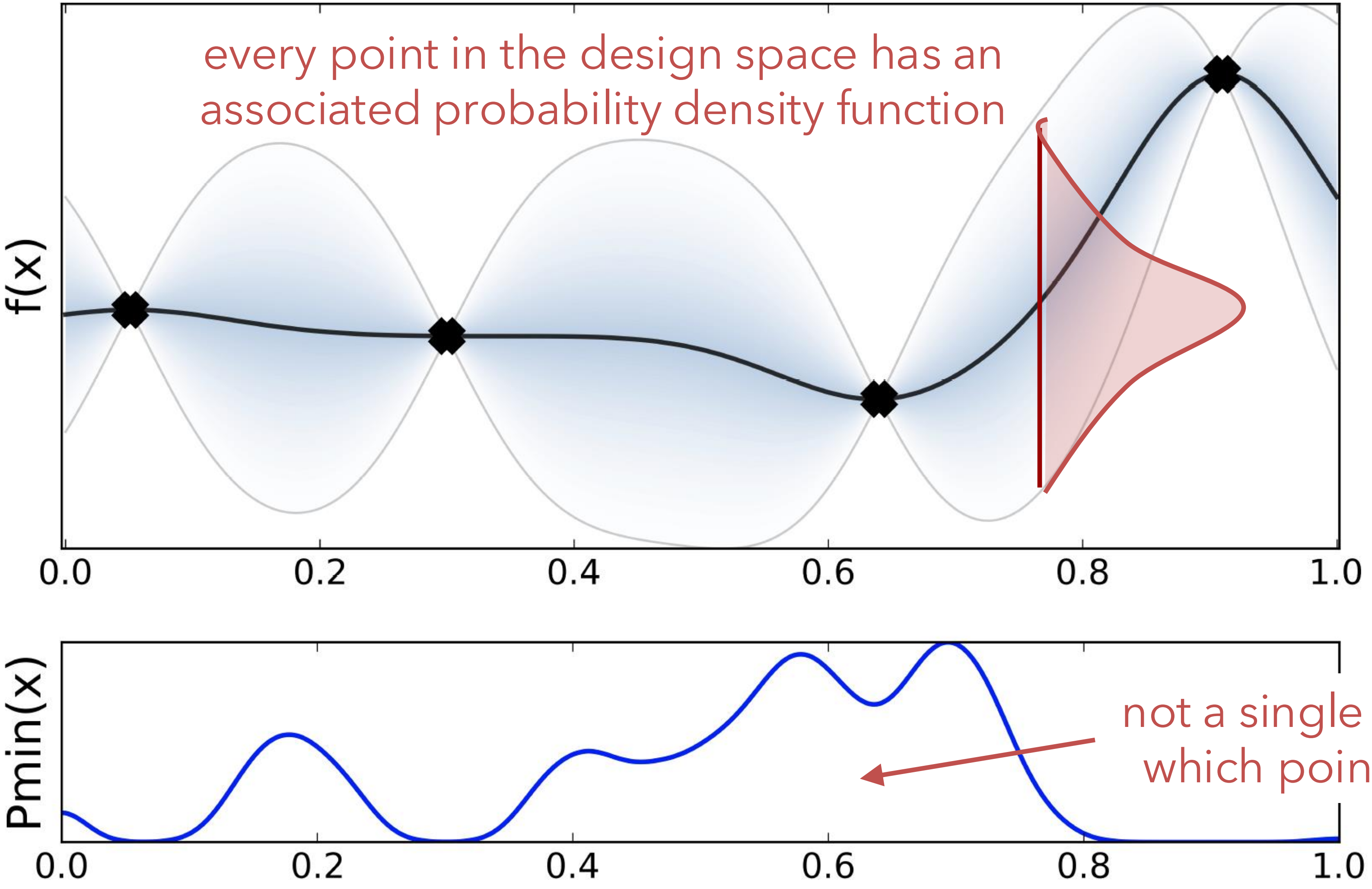
One hundred curves



Intuitive solution, fit a surrogate model

Infinite curves

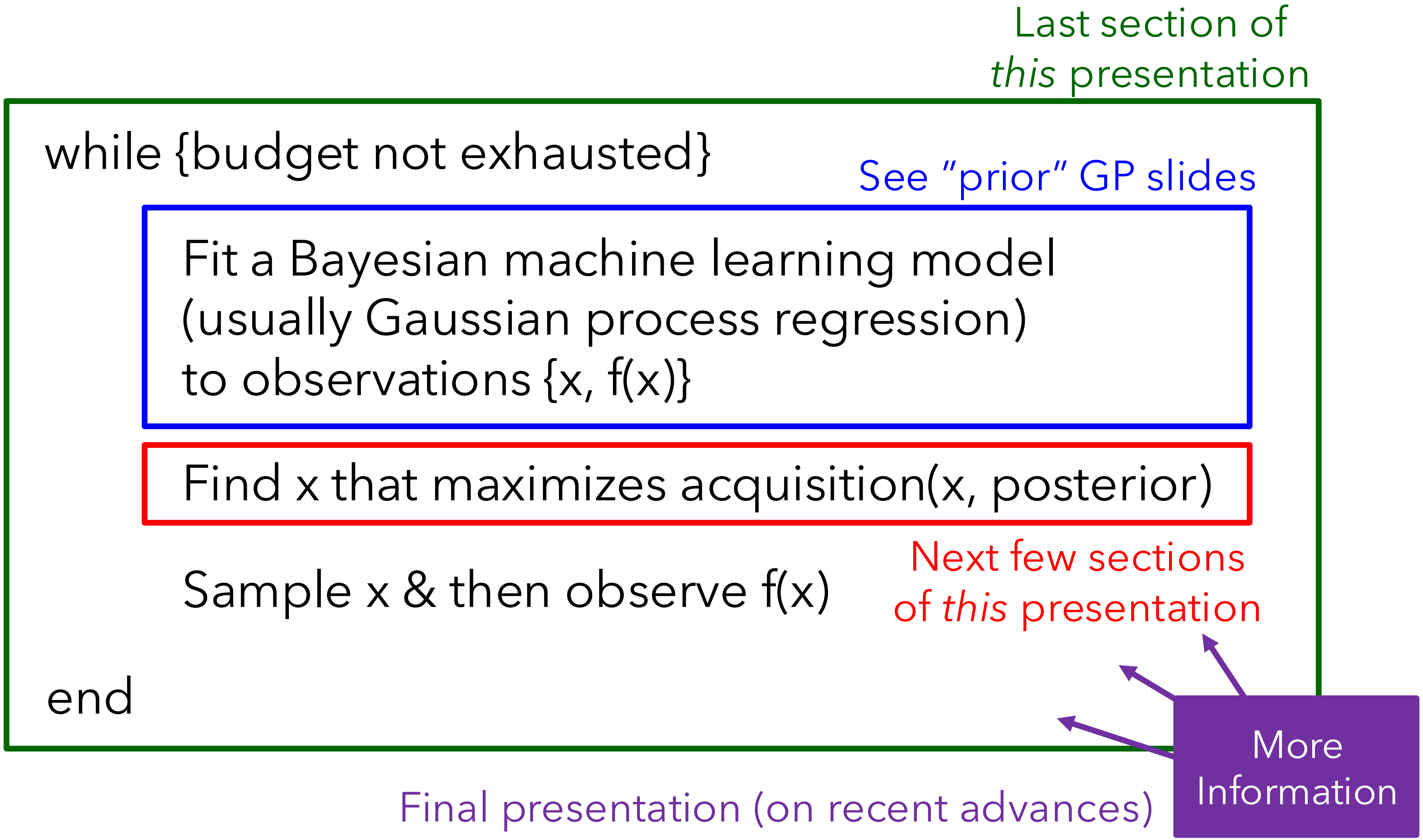
(Need the help of information theory to properly define models + metrics)



every point in the design space has an associated probability density function

not a single minimum, which point to take?

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)}

    Find x that maximizes acquisition(x, posterior)

    Sample x & then observe f(x)
end
```

Last section of *this* presentation

See "prior" GP slides

Next few sections of *this* presentation

More Information

Final presentation (on recent advances)

Outline

- Introduction to Bayesian optimization
  - White-box vs. black-box, prevalence of expensive functions, bird’s eye view
- Improvement-based acquisition functions
  - Expected improvement, knowledge gradient
- Information-theoretic acquisition functions
  - Predictive, max-value, and joint entropy search
- Constrained Bayesian optimization
  - Expected improvement with constraints, exact penalty methods
- Practical considerations
  - Optimizing the acquisition function, kernel adaptation, acquisition scheduling

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)}

    Find x that maximizes acquisition(x, posterior)

    Sample x & then observe f(x)

end
```

Assume our goal is to minimize f(x)
[same idea as maximize, just replace with -f(x)]

How to Define an Acquisition Function α_n?

- When properly selected, the value of $\alpha_n(x)$ at any $x \in \Omega$ should be a good measure of the (expected) benefit of querying f at that point in future
  - Must depend on the posterior distribution of $f|y_{1:n}$

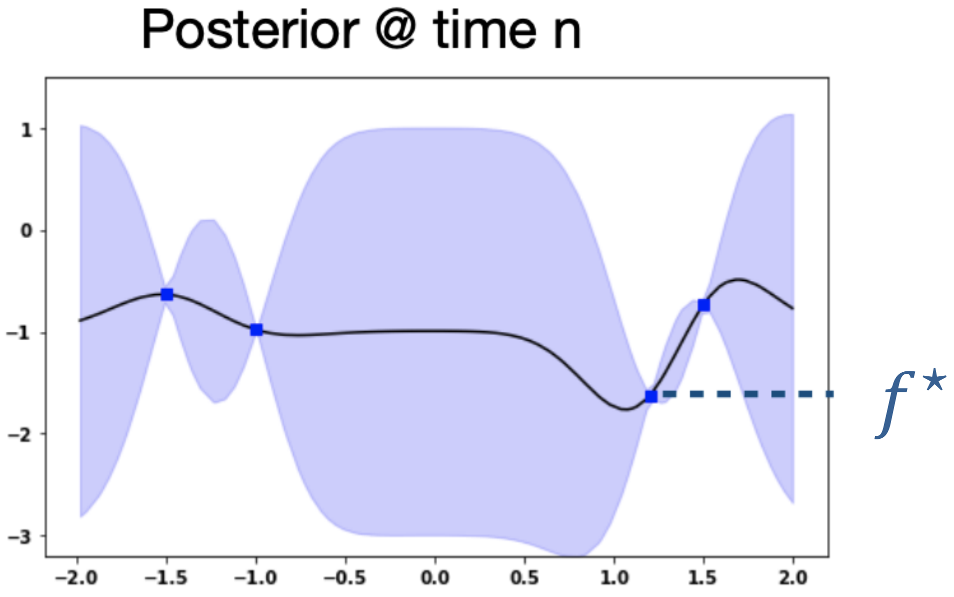
- This implies we should like to preferentially sample at the point that produces the highest possible value of the acquisition function:

$$x_{n+1} \in \mathrm{argmax}_{x\in\Omega}\, \alpha_n(x)$$

- It is expected for this problem to be much cheaper to solve since, unlike f, we have some equation-based form for α_n

Let's Start with Expected Improvement (EI) Acquisition Function

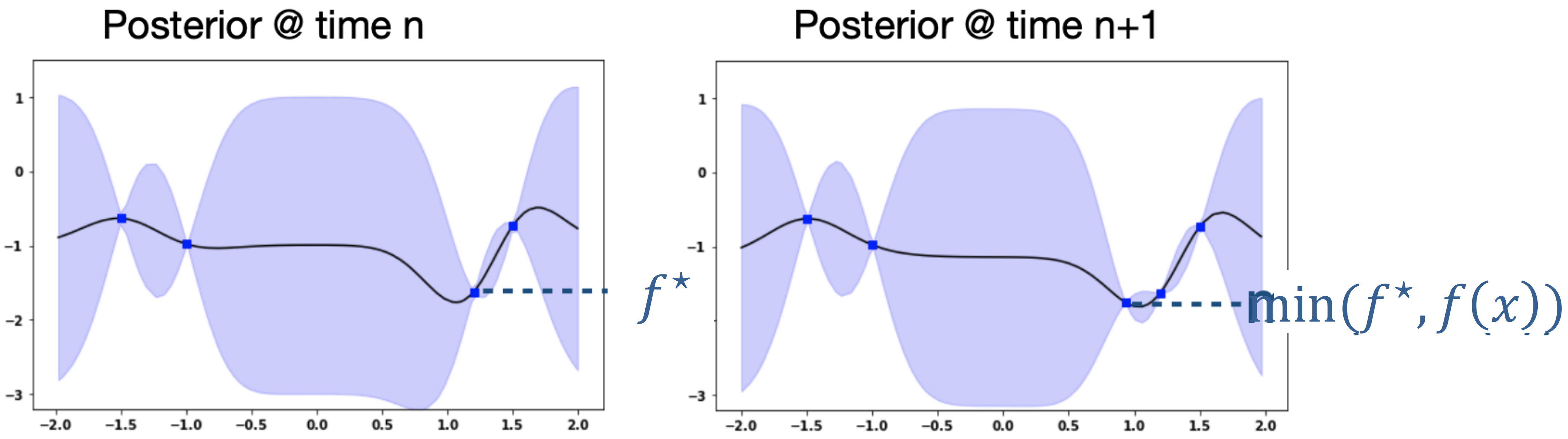
[Mockus 1989; Jones, Schonlau, and Welch 1998]



- Loss if we stop now: $f^\star$

Let’s Start with Expected Improvement (EI) Acquisition Function

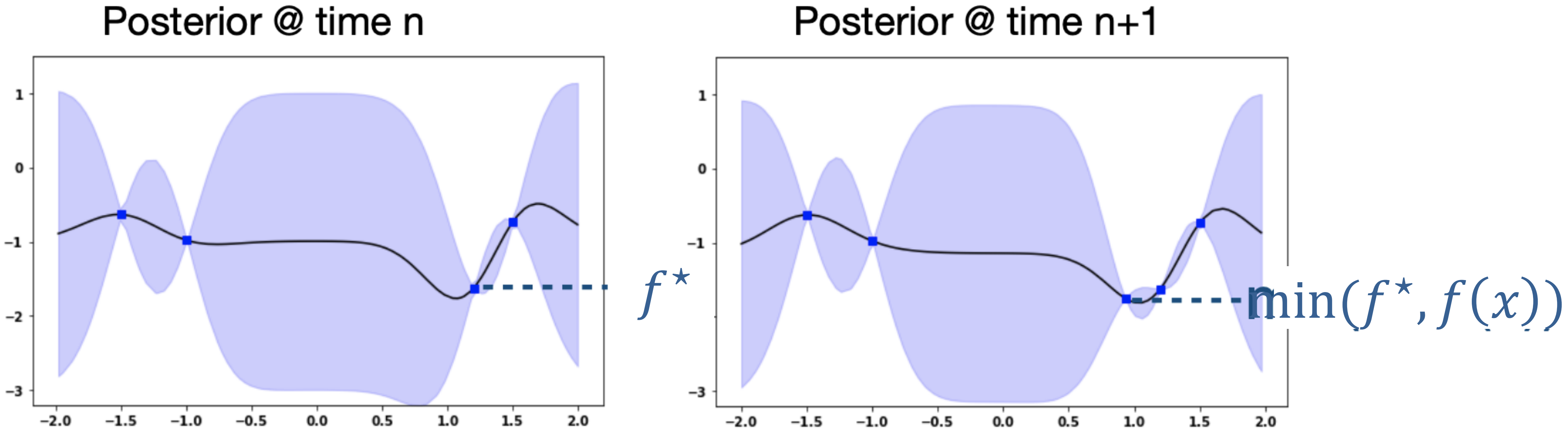
[Mockus 1989; Jones, Schonlau, and Welch 1998]



- Loss if we stop now: $f^\star$
- Loss if we stop after sampling at $f(x)$: $\min(f^\star, f(x))$

Let's Start with Expected Improvement (EI) Acquisition Function

[Mockus 1989; Jones, Schonlau, and Welch 1998]



- Loss if we stop now: $f^\star$
- Loss if we stop after sampling at $f(x)$: $\min(f^\star, f(x))$
- Expected reduction in loss due to sampling: $\mathbb{E}_n[f^\star - \min(f^\star, f(x))]$

Let's Start with Expected Improvement (EI) Acquisition Function

[Mockus 1989; Jones, Schonlau, and Welch 1998]

$$\mathrm{EI}_n(x) = \mathbb{E}_n\{f^\star - \min(f^\star, f(x))\}$$

$$= \mathbb{E}_n\{\max\{f^\star - f(x), 0\}\}$$

$$= \underbrace{\mathbb{E}_Z\{\max\{f^\star - \mu_n(x) - \sigma_n(x)Z\ , 0\}\}}_{\text{Integral can be carried out analytically using integration by parts}}$$

Closed-form Expression Expected Improvement

Standard normal cumulative distribution function (CDF)

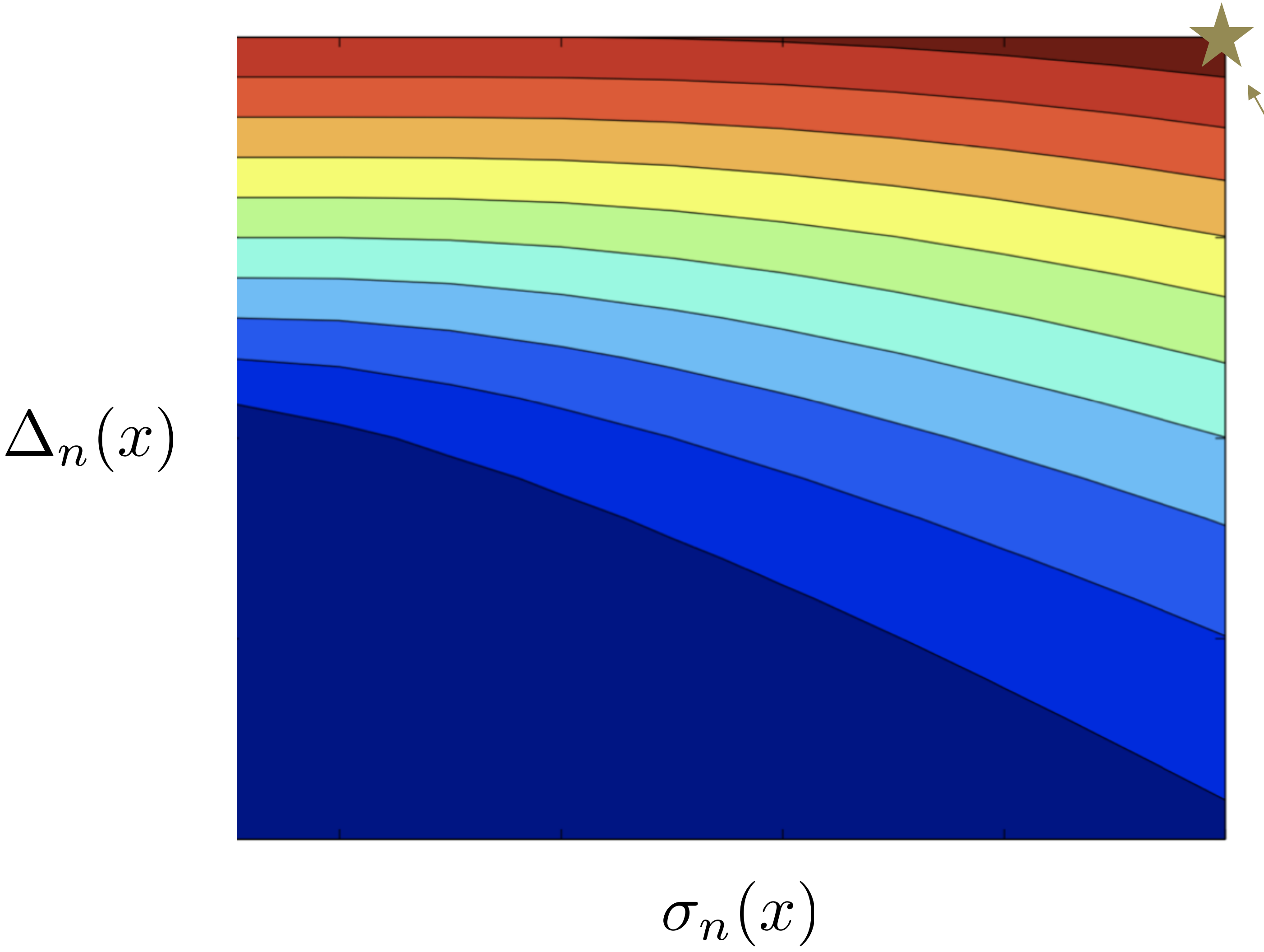
Standard normal probability density function (PDF)

$$\mathrm{EI}_n(x) = \Delta_n(x)\Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) + \sigma_n(x)\phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right)$$

Where $\Delta_n(x) = f_n^\star - \mu_n(x)$ is expected quality

[Jones et al., 1998; Frazier, 2018]

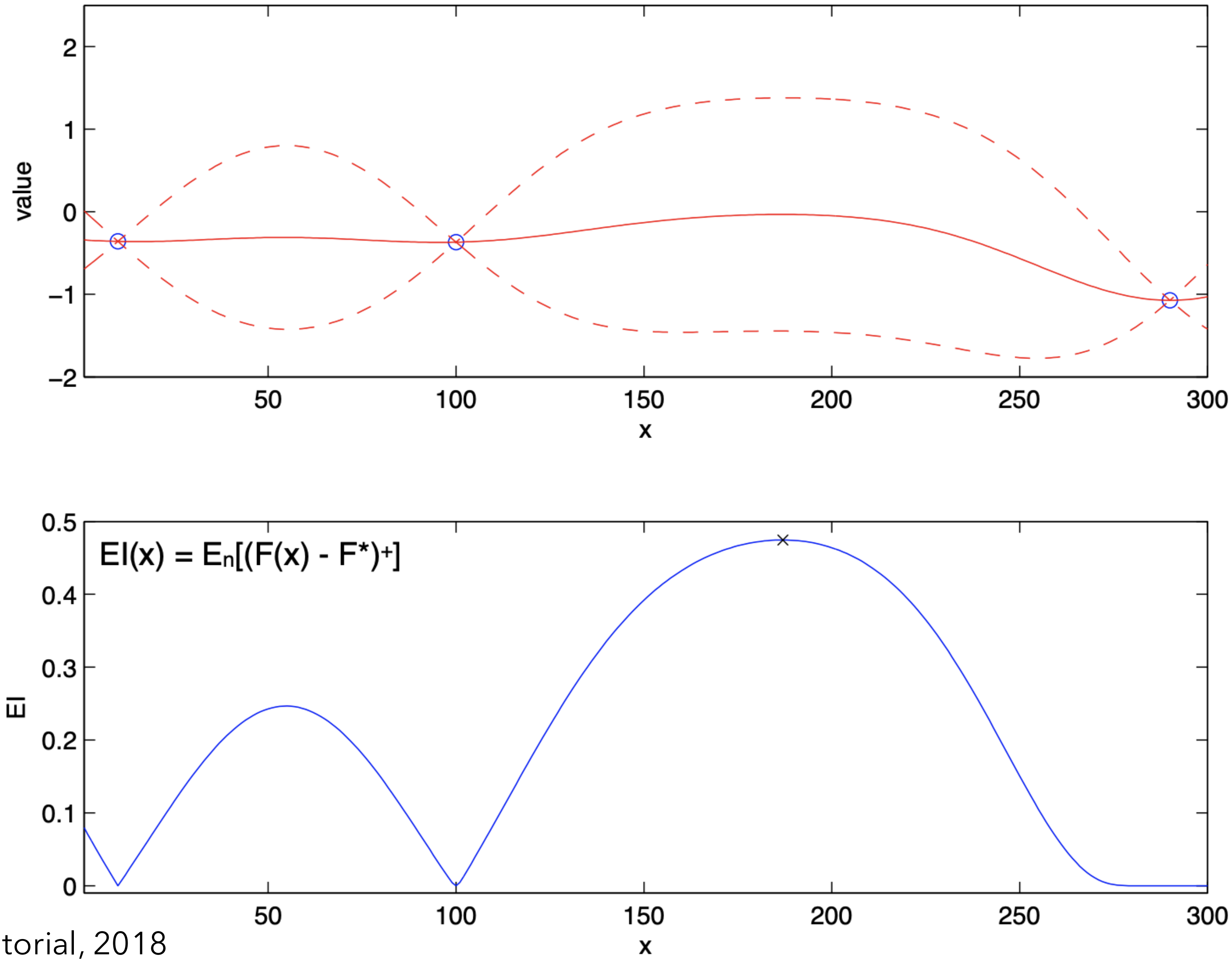
EI Tradeoffs Exploitation ($\Delta_n(x)$) vs. Exploration ($\sigma_n(x)$)



Best to have both high quality and high uncertainty

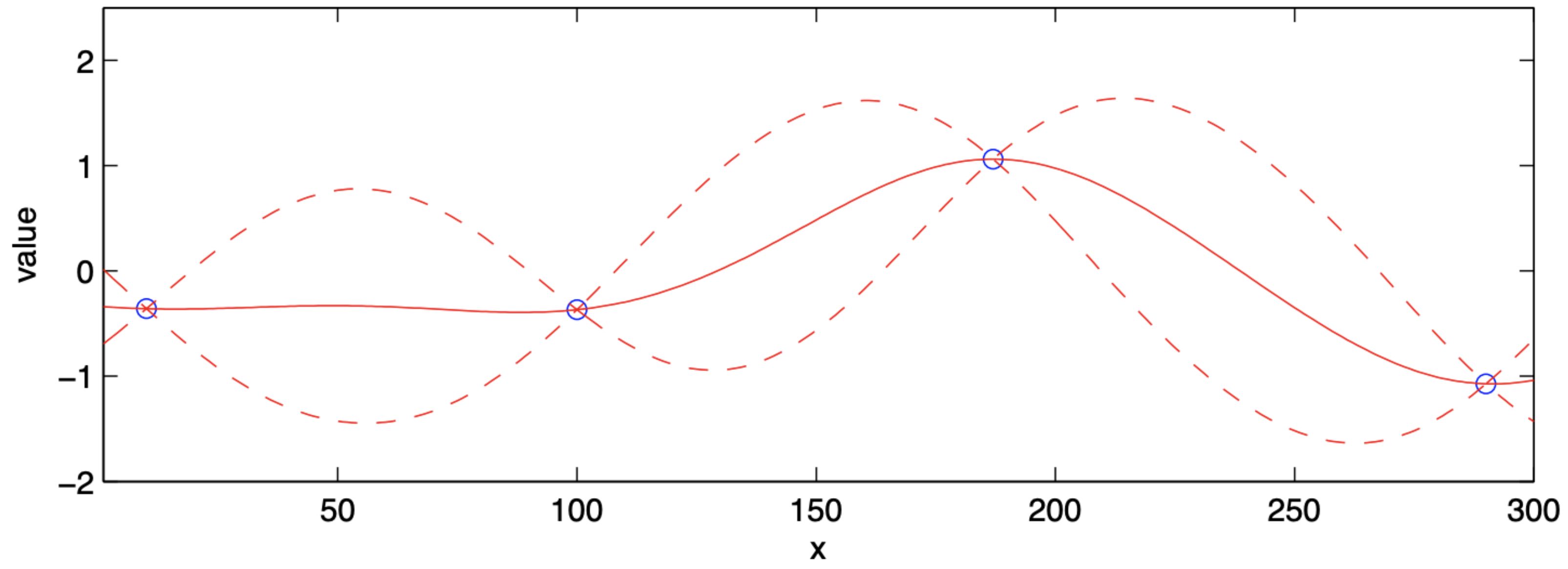
[Jones et al., 1998; Frazier, 2018]

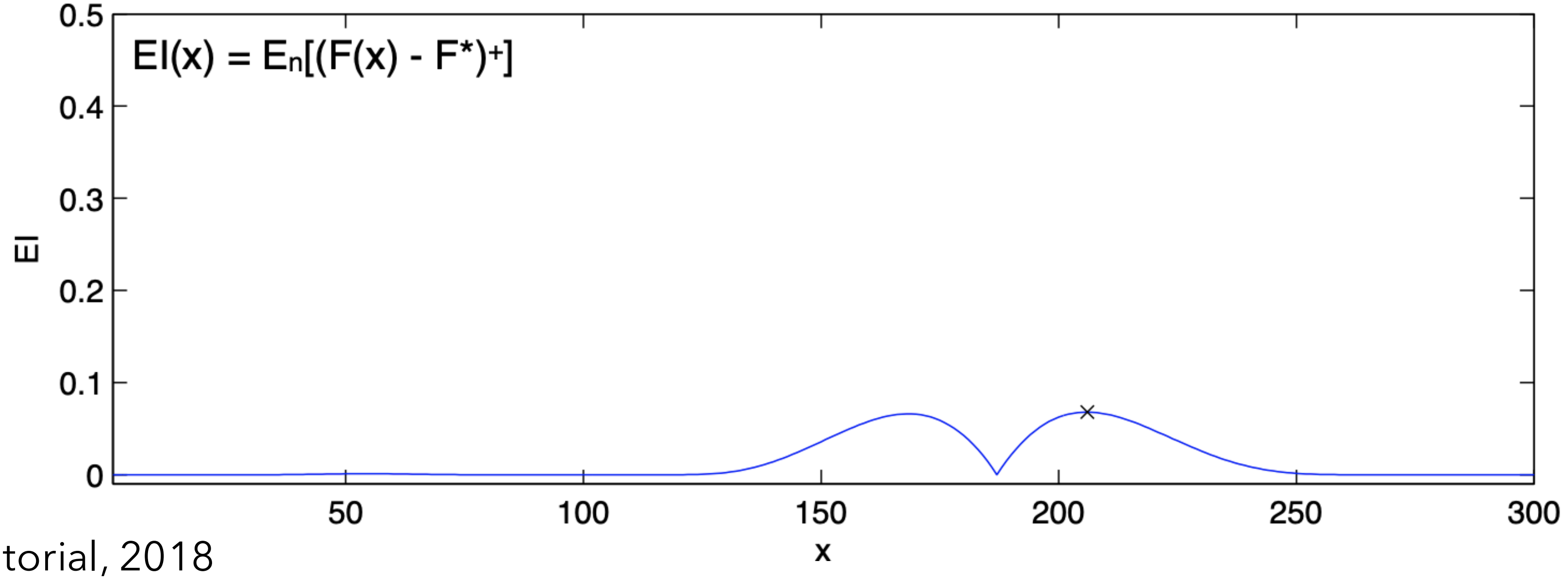
Quick Example of EI for Maximizing 1-Dimensional Objective



P. Frazier, INFORMS Tutorial, 2018

Quick Example of EI for Maximizing 1-Dimensional Objective

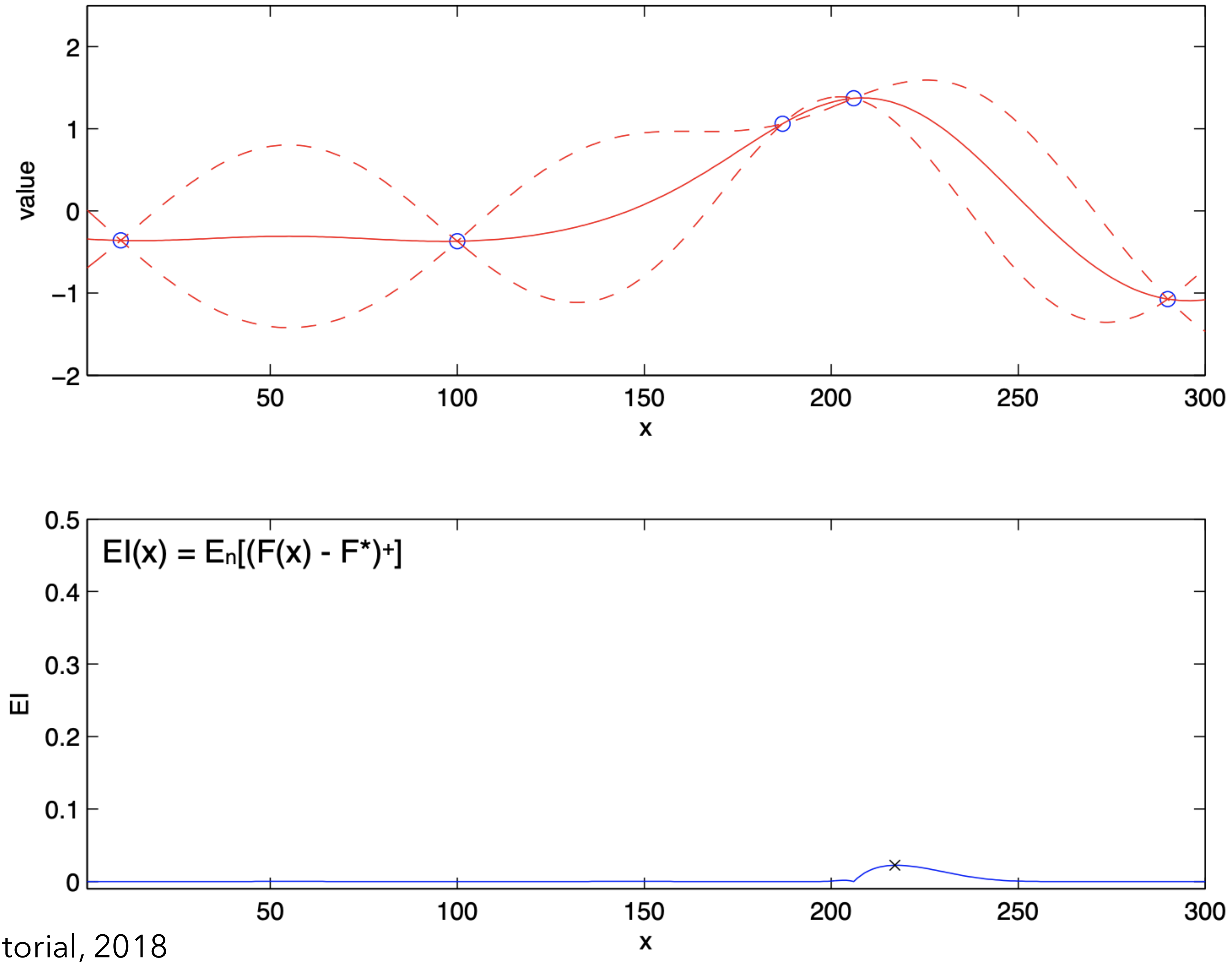




$$EI(x) = E_n[(F(x) - F^*)^+]$$

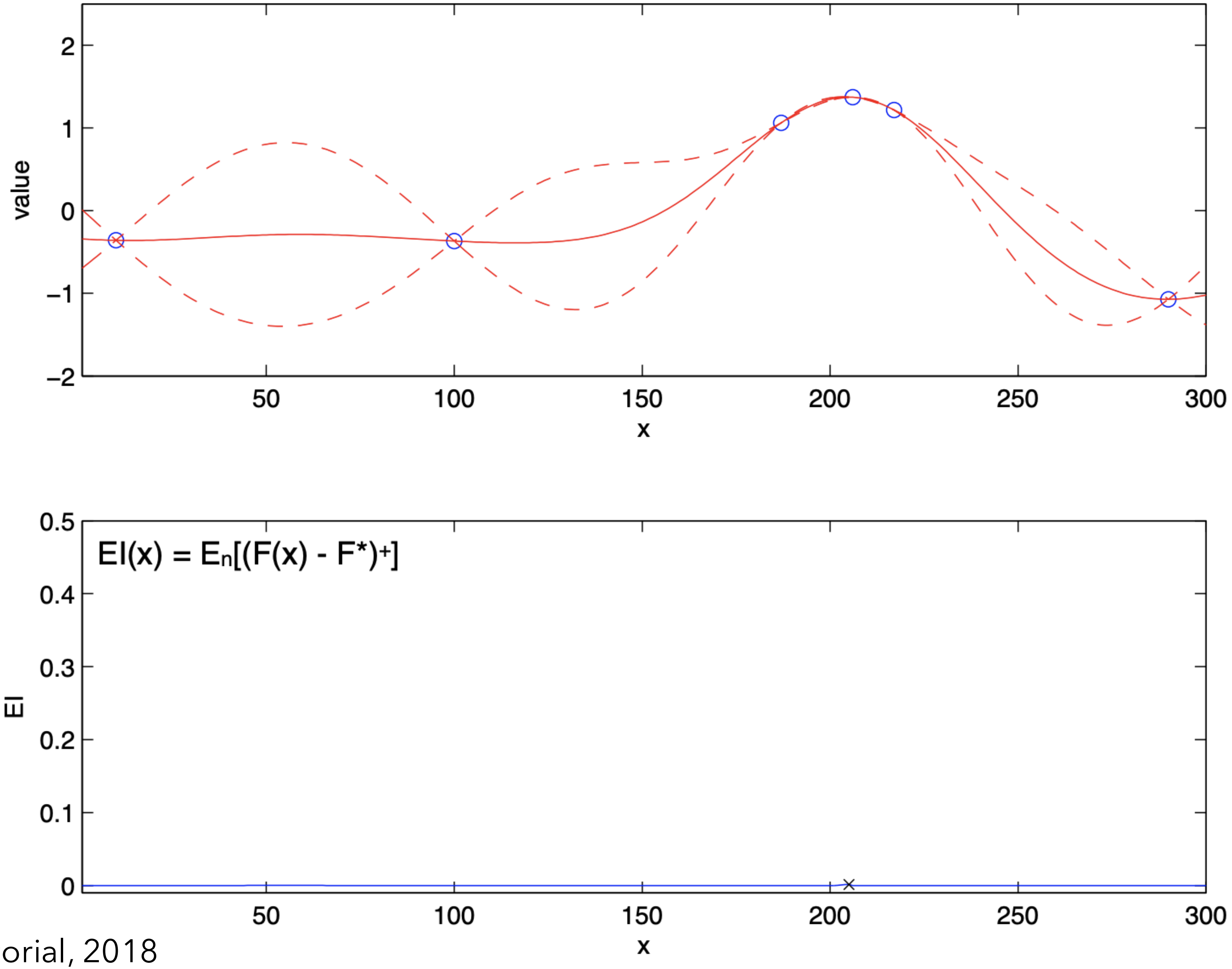
P. Frazier, INFORMS Tutorial, 2018

Quick Example of EI for Maximizing 1-Dimensional Objective



P. Frazier, INFORMS Tutorial, 2018

Quick Example of EI for Maximizing 1-Dimensional Objective



$EI(x) = E_n[(F(x) - F^*)^+]$

P. Frazier, INFORMS Tutorial, 2018

Thought Experiment

- What should expected improvement (EI) reduce to when the variance of the prediction is zero everywhere?
  - – We no longer have uncertainty, so we no longer need to sequentially search (simply find the minimum of the mean function)
  - – We can think of traditional optimization as placing a GP prior with perfectly known mean function $\mu_0(x) = f(x)$ (and zero variance/covariance), then running EI one step to find the "true" minimum
  - – EI in some sense generalizes traditional "white-box" optimization to the unknown "black-box" setting → attempts to be information-optimal

Is EI Optimal in any Sense?

- Yes, it turns out that EI is Bayes-optimal under some assumptions:
  - There is no noise in the observations of the objective function
  - We are only willing to select previously evaluated point as final solution
  - We are risk neutral (i.e., we value a random outcome according to its expected value, hence $\mathbb{E}[\text{Reduction in Loss}]$)
  - This is our last evaluation

Why is this assumption needed?

In general, we must solve a sequential decision-making problem

- The loss that we calculated previously is only a function of the next sample that we take; however, in general, we have a budget of N remaining samples $\{x_1, x_2, \ldots, x_N\}$
- Furthermore, every sample that we take yields more data, such that we have more information to make our next decision
- We can formulate this as a stochastic optimal control problem where our state is current data, action is next sample, and immediate reward is reduction in loss

Best (finite-budget) sampling strategy is policy that optimizes the value function (total loss reduction)

- Policy: $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_N\}$, $x_k = \pi_k(\mathcal{D}_{k-1})$
- Value function: $V_{\boldsymbol{\pi}}(\mathcal{D}_0) = \mathbb{E}\left[\sum_{k=1}^{N} r(\mathcal{D}_{k-1}, \mathcal{D}_k)\right]$, r(·) is loss reduction
- Optimal policy: $V^\star(\mathcal{D}_0) = V_{\boldsymbol{\pi}^\star}(\mathcal{D}_0) = \max_{\boldsymbol{\pi}\in\Pi} V_{\boldsymbol{\pi}}(\mathcal{D}_0)$
- Solution expressed using dynamic programing:

$$V_k(\mathcal{D}) = \max_{x\in\Omega} \mathbb{E}_{\mathcal{D}^+}\left[r(\mathcal{D}, \mathcal{D}^+) + V_{k-1}(\mathcal{D}^+)\right] \quad , \quad \forall k = 1, \ldots, N$$

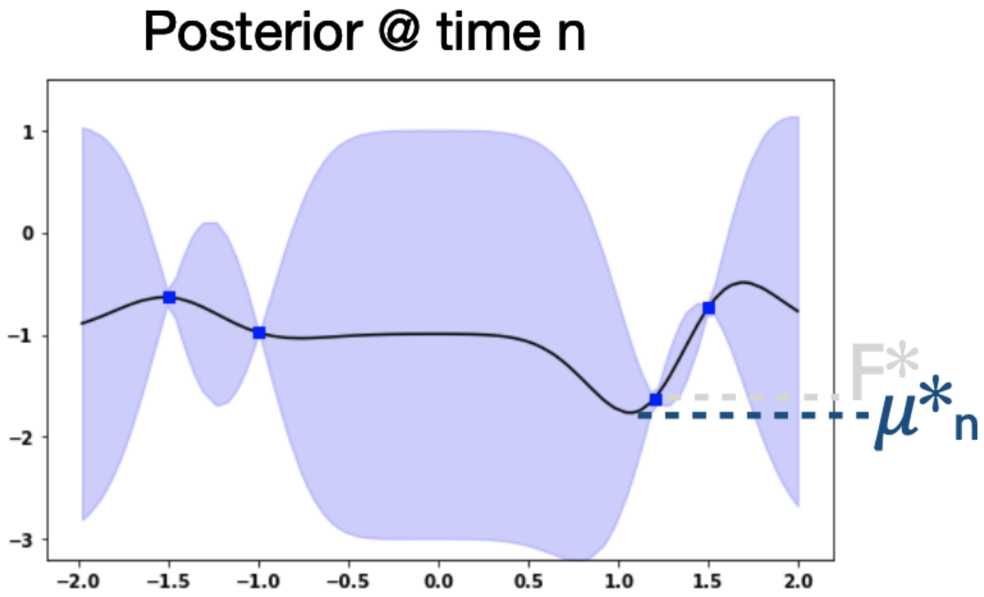
[Paulson et al, *Conference on Decision & Control*, 2022]

Let's Drop Two of These Assumptions

- Yes, it turns out that EI is Bayes-optimal under some assumptions:
  - ~~There is no noise in the observations of the objective function~~
  - ~~We are only willing to select previously evaluated point as final solution~~
  - We are risk neutral (i.e., we value a random outcome according to its expected value, hence $\mathbb{E}[\text{Reduction in Loss}]$)
  - This is our last evaluation

Yields Knowledge Gradient (KG) acquisition function, what should be loss?

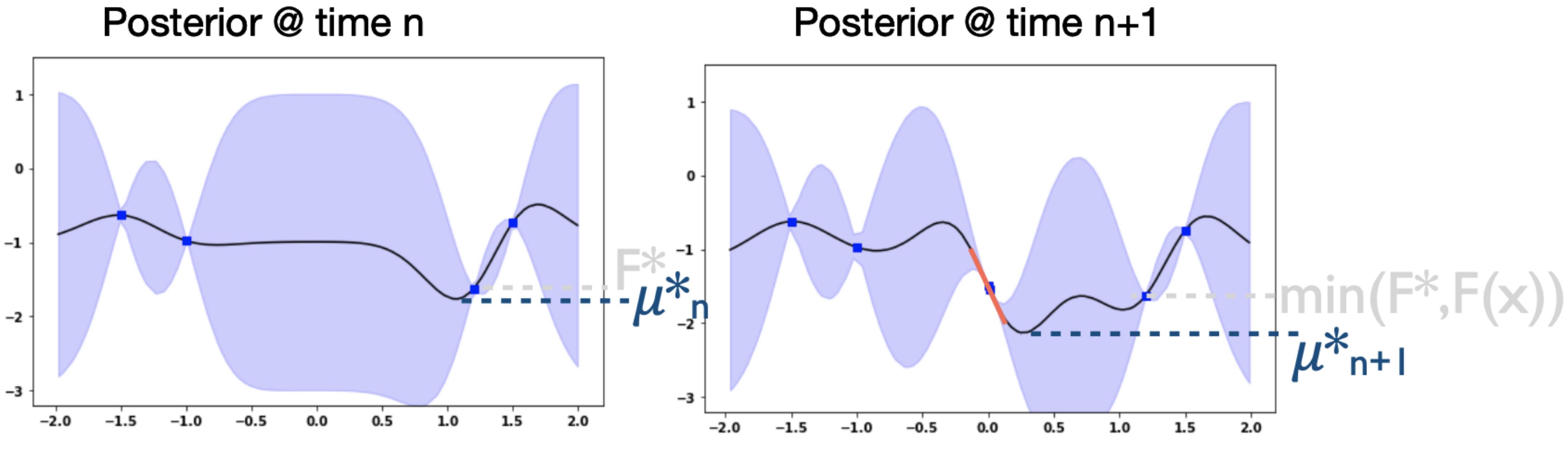
Knowledge Gradient (KG) Acquisition Function



- Loss if we stop now: $\mu_n^\star = \min_{x \in \Omega} \mu_n(x)$

[Frazier et al. , *SIAM Journal on Optimization and Control*, 2009]

Knowledge Gradient (KG) Acquisition Function

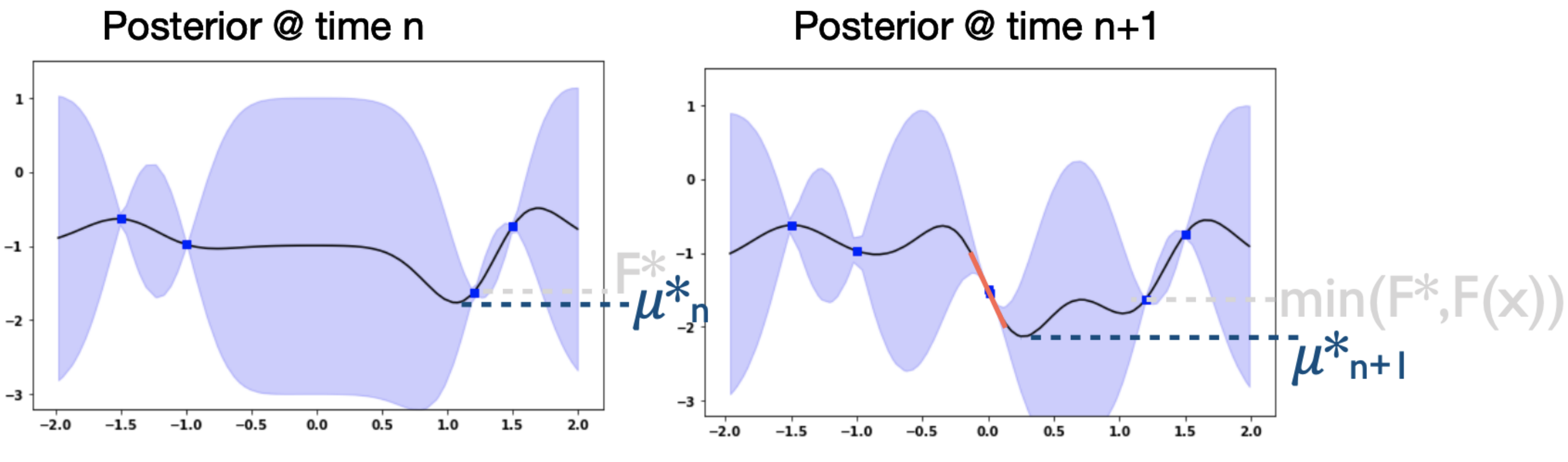


- Loss if we stop now: $\mu_n^\star = \min_{x \in \Omega} \mu_n(x)$

- Loss if we stop after sampling $f(x)$: $\mu_{n+1}^\star = \min_{x \in \Omega} \mu_{n+1}(x)$

[Frazier et al. , *SIAM Journal on Optimization and Control*, 2009]

Knowledge Gradient (KG) Acquisition Function



- Loss if we stop now: $\mu_n^\star = \min_{x \in \Omega} \mu_n(x)$
- Loss if we stop after sampling $f(x)$: $\mu_{n+1}^\star = \min_{x \in \Omega} \mu_{n+1}(x)$
- Expected reduction in loss due to sampling: $\mathbb{E}_n[\mu_n^\star - \mu_{n+1}^\star \mid \text{sample } x]$

[Frazier et al. , *SIAM Journal on Optimization and Control*, 2009]

Do you see any challenges with KG?

- The main disadvantage of KG is that we lose the analytic formula that we were able to derive for EI
  - Makes is harder to maximize than EI, as we now have a two-stage optimization

- Are there alternative strategies for handling observation noise?
  - Yes, many other strategies exist including:
    - Modifications to EI to mitigate impact of noise
    - Alternative acquisition functions

Use a “plug-in” value for the incumbent

- Computing EI with noise is challenging because we no longer know the incumbent $f^\star$; a simple way to address this is to replace with an alternative value such as the GP estimate of the best function value

$$\hat{f}^\star = \min_{x \in \Omega} \mu_n^\star$$

- The main issue with this approach is that it tends to underestimate the improvement potential in regions of Ω with large uncertainty
  - Because mean smooths out noise, potentially ignores regions where there could be significant improvement due to high variance in $f(x)$ → over-exploitation

[Gramacy et al. , *Bayesian Statistics*, 2011]

Noisy Expected Improvement (NEI)

- The noisy expected improvement function

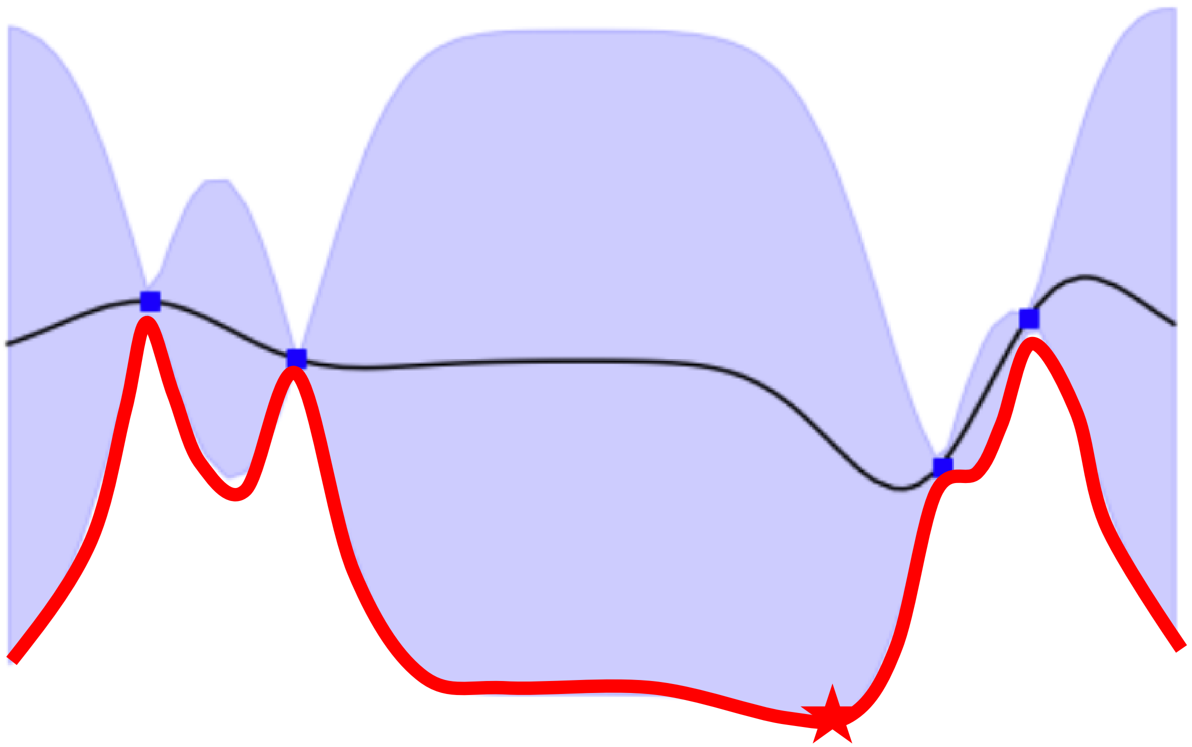
$$\mathrm{NEI}(x|\mathcal{D}) = \mathbb{E}_{\mathbf{f}^n}[\mathrm{EI}(x|\mathbf{f}^n)|\mathcal{D}] = \int_{\mathbf{f}^n} \mathrm{EI}(x|\mathbf{f}^n)p(\mathbf{f}^n|\mathcal{D})\,d\mathbf{f}^n$$

Where $\mathrm{EI}(x|\mathbf{f}^n)$ denotes the standard EI expression assuming noiseless observations of the function $\mathbf{f}^n$ and $p(\mathbf{f}^n|\mathcal{D}) \sim N(\boldsymbol{\mu}_f^n, \Sigma_f^n)$ is the GP posterior prediction of the function values given noisy data $\mathcal{D}$

- By incorporating noise into the decision-making process, NEI promotes a more exploratory behavior, especially in early iterations when the model is still uncertain about the objective function landscape.

[Letham et al. , *Bayesian Analysis*, 2017]

Alternative Acquisition: Lower Confidence Bound (LCB)



- Simple idea: Just directly minimize a lower bound on the function

$$\min_{x \in \Omega} \mu_n(x) - \sqrt{\beta_{n+1}} \sigma_n(x)$$

We can establish rigorous bounds on "regret" for LCB

- Lower confidence bound: $l_n(x) = \mu_{n-1}(x) - \sqrt{\beta_n}\sigma_{n-1}(x)$
- Upper confidence bound: $u_n(x) = \mu_{n-1}(x) + \sqrt{\beta_n}\sigma_{n-1}(x)$

- Assume that true function satisfies $f(x) \in [l_n(x), u_n(x)]$
  (can prove this holds with high probability for sufficiently large β_n)

- Performance measure: Regret r_n defined as distance to optimal solution:

$$r_n = f(x_n) - f(x^\star)$$

[Srinivas et al. , *IEEE Transactions on Information Theory,* 2012]

We can establish rigorous bounds on “regret” for LCB

- The following sequence of inequalities hold:

$$r_n = f(x_n) - \min_{x \in \Omega} f(x)$$

[Definition of regret]

$$\leq u_n(x_n) - \min_{x \in \Omega} f(x)$$

[Property of upper bound]

$$\leq u_n(x_n) - \min_{x \in \Omega} l_n(x)$$

[Property of lower bound]

$$= u_n(x_n) - l_n(x_n)$$

[Definition of our sample choice $x_n = \mathrm{argmin}_x\, l_n(x)$]

$$= 2\sqrt{\beta_n}\sigma_{n-1}(x_n)$$

[Difference between bounds given by standard deviation]

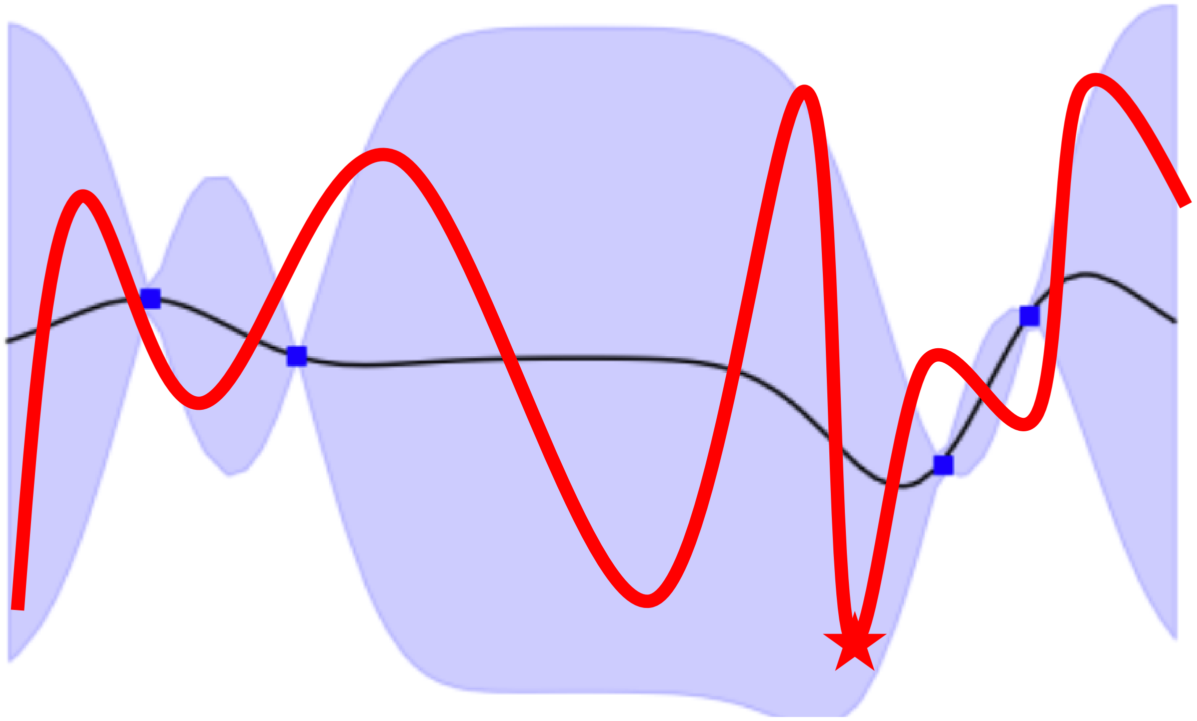
[Srinivas et al. , *IEEE Transactions on Information Theory,* 2012]

Be careful with LCB in practice…

- Although theoretical values for $\{\beta_n\}$ exist, they are often too conservative to be very useful in practice → not great short-term performance

- In published experiments, it is common to set $\sqrt{\beta_n} \approx 2$ as a heuristic, which can result in reasonable performance in certain problems; however, performance can be quite sensitive to the choice of β_n for some problems

Alternative Acquisition: Thompson Sampling (TS)



- Minimize random sample of the GP, i.e., $f^{(n)} \sim \mathcal{GP}(\mu_n(x), \sigma_n^2(x))$

$$\min_{x \in \Omega} f^{(n)}(x)$$

Efficient generation of *differentiable* TS from GP posteriors

- Interesting recent work showing how to get an efficient, differentiable TS
    - – I highly recommend the following paper that goes in-depth on GP sampling:
        - [Wilson et al., "Efficiently sampling functions from GP posteriors", *ICML*, 2020]

- Core argument is one can use Matheron's rule to break down posterior sample into a "prior" and "update" step that has the following form for GPs

$$(f \mid \boldsymbol{y})(\cdot) \approx \underbrace{\sum_{i=1}^{l} w_i \phi_i(\cdot)}_{\text{weight-space prior}} + \underbrace{\sum_{j=1}^{n} v_j k\left(\cdot, \boldsymbol{x}_j\right)}_{\text{function-space update}}, \qquad \boldsymbol{v} = \left(\boldsymbol{K}_{n,n} + \sigma_n^2 \boldsymbol{I}\right)^{-1}(\boldsymbol{y} - \boldsymbol{\Phi w} - \underbrace{\boldsymbol{\varepsilon}}_{\substack{\text{random noise realizations,} \\ \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I})}})$$

Outline

- Introduction to Bayesian optimization
  - White-box vs. black-box, prevalence of expensive functions, bird's eye view
- Improvement-based acquisition functions
  - Expected improvement, knowledge gradient
- Information-theoretic acquisition functions
  - Predictive, max-value, and joint entropy search
- Constrained Bayesian optimization
  - Expected improvement with constraints, exact penalty methods
- Practical considerations
  - Optimizing the acquisition function, kernel adaptation, acquisition scheduling

A Quick Primer on Information Theory

- The concept of information entropy was introduced by Claude Shannon in his 1948 paper "A Mathematical Theory of Communication"

- In essence, for a discrete random variable X that takes values in Ω, we can compute its entropy as follows

$$H[X] = \mathbb{E}[-\log p(X)] = -\sum_{x \in \Omega} p(x) \log p(x)$$

- The entropy $H[X]$ quantifies the average level of uncertainty or information associated with the variable's potential outcomes
  - Generalizes to continuous & multivariate distributions → differential entropy

Can we use the notion of entropy to define our acquisition function in Bayesian optimization?

- Yes!
  - Similar to the idea of the improvement-based acquisitions, we can now attempt to decrease the entropy = increase our information by querying new points

- Expected information gain (EIG) – expected decrease in entropy – over posterior for global optima $x^\star$ if we were to query f at point x is:

$$\alpha^{\mathrm{ES}}(x) = H[p(x^*|\mathcal{D})] - \mathbb{E}_{p(y_x|\mathcal{D})}\left[H[p(x^*|\mathcal{D} \cup \{(x, y_x)\})]\right]$$

Entropy of posterior over global optima $x^\star$

Expected entropy of posterior over global optima $x^\star$…if we were to query at location x

[P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *JMLR*, 2012]

Entropy search acquisition is challenging to compute!

[Predictive entropy search (PES) to the rescue]

- Two difficulties
  - Need to compute $p(x^*|\mathcal{D} \cup \{(x, y_x)\})$ for many different values
  - Neither entropy expression can be computed analytically

- Luckily, we can use a nice trick to simplify computation
  - Notice that $\alpha^{\mathrm{ES}}(x)$ equivalent to mutual information (MI) between $x^\star|\mathcal{D}$ and $y_x|\mathcal{D}$
  - MI is symmetric such that we can reverse the order and obtain:

$$\alpha^{\mathrm{PES}}(x) = H[p(y_x|\mathcal{D})] - \mathbb{E}_{p(x^\star|\mathcal{D})}\big[H[p(y_x|\mathcal{D}, x^\star)]\big]$$

We can derive analytic expression using GP

Generate MC samples before optimization by finding optima of TS

Generate approximate samples using expectation propagation

Entropy search concept easily generalizes to other statistics

- In general, can write out MI for some statistic of unknown function $S(f)$:

$$\begin{aligned}\alpha^S(x) &= \mathrm{MI}(y_x; S(f)|x, \mathcal{D}) \\ &= H[p(y_x|\mathcal{D})] - \mathbb{E}_{p(S(f)|\mathcal{D})}[p(y_x|x, \mathcal{D}, S(f))]\end{aligned}$$

- Predictive entropy search (PES) sets $S(f) = x^\star$
  - Hernández-Lobato et al., *NeurIPS*, 2014

- Max-value entropy search (MES) sets $S(f) = y^\star = \max_{x \in \Omega} f(x)$
  - Wang and Jegelka, *ICML*, 2017

- Joint entropy search (JES) sets $S(f) = (x^\star, y^\star)$
  - Hvarfner, Hutter, and Nardi, *NeurIPS*, 2022

Visual Illustration of PES, MES, and JES

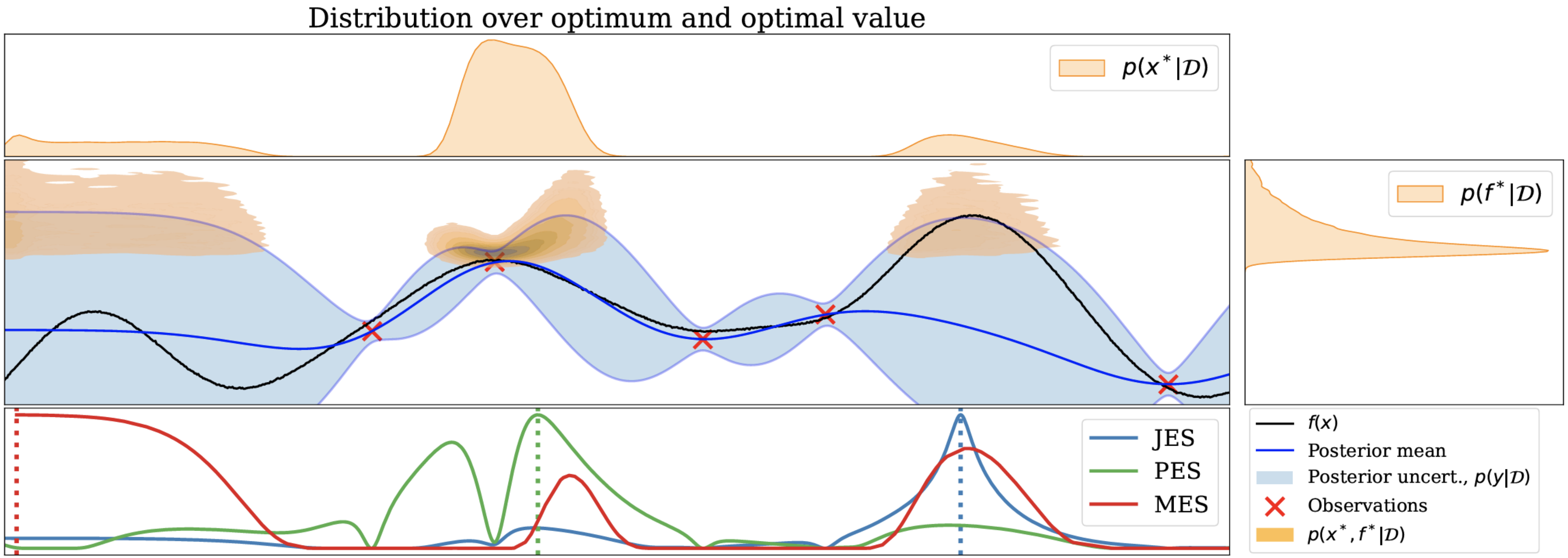


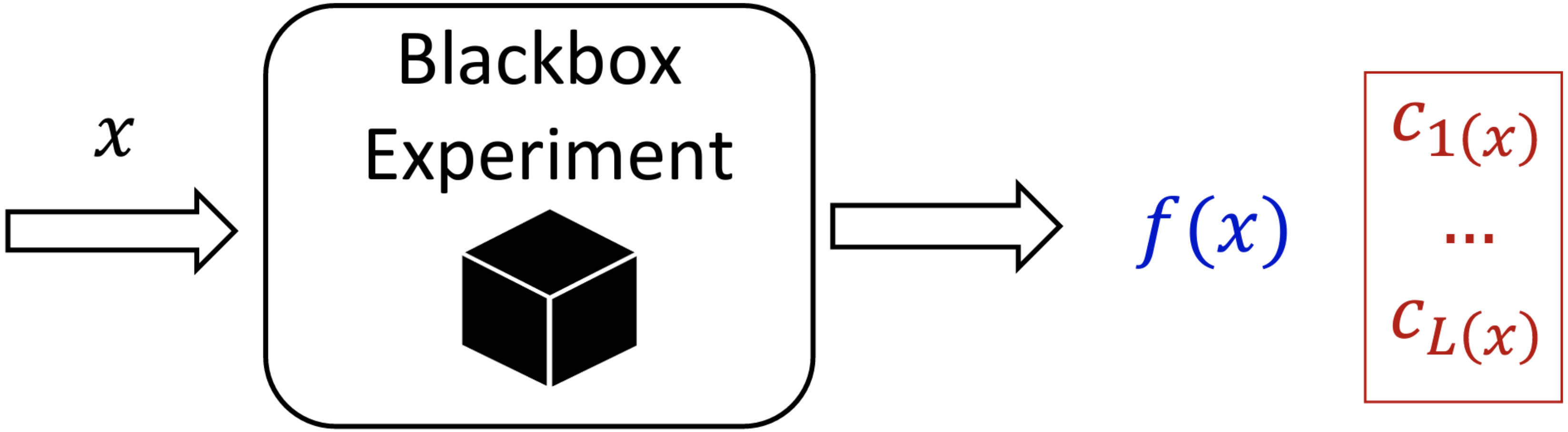
Figure 1: The densities considered by ES/PES (top), MES (right) and JES (center) on a one-dimensional toy example. The multimodal density $p(\boldsymbol{x}^*, f^*)$ is reduced to a heavy-tailed density over f^* for the density used by MES (right), which does not capture the multi-modality of the density over the optimum. The density over $\boldsymbol{x}^*$ used by PES (top) does not capture the apparent exploration/exploitation trade-off that exists between the modes. The acquisition functions and their next point selections are highlighted with dashed lines (bottom).

Hvarfner, Hutter, and Nardi, Joint Entropy Search for Maximally-Informed Bayesian Optimization, *NeurIPS*, 2022

Outline

- Introduction to Bayesian optimization
  – White-box vs. black-box, prevalence of expensive functions, bird's eye view
- Improvement-based acquisition functions
  – Expected improvement, knowledge gradient
- Information-theoretic acquisition functions
  – Predictive, max-value, and joint entropy search
- Constrained Bayesian optimization
  – Expected improvement with constraints, exact penalty methods
- Practical considerations
  – Optimizing the acquisition function, kernel adaptation, acquisition scheduling

What about expensive black-box constraints?



x

Blackbox Experiment

$f(x)$

$c_{1}(x)$

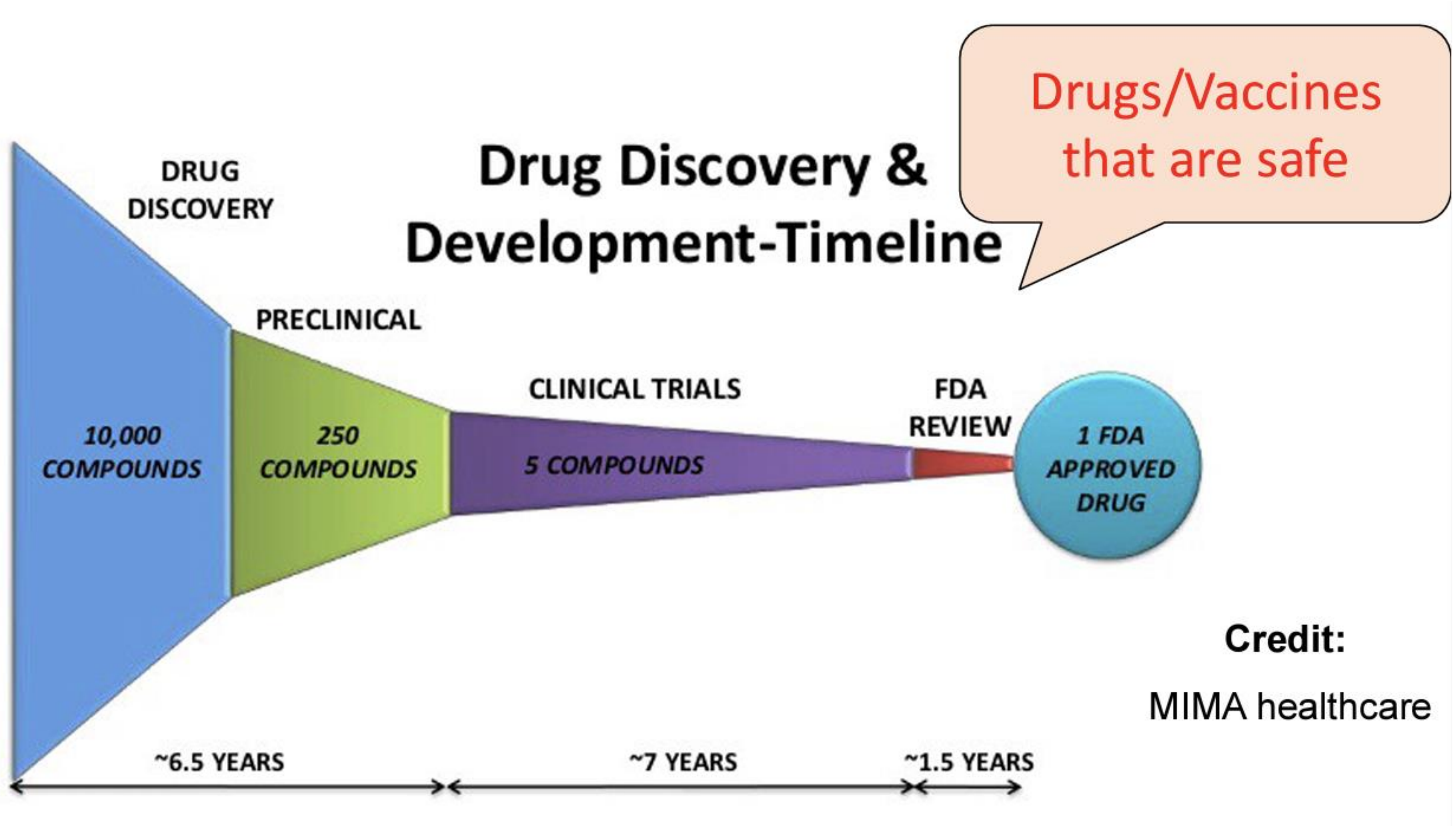
…

$c_{L}(x)$

Objective and constraints evaluation of design x

- **Goal:** find the constrained optima that minimizes the objective subject to satisfying constraints across the entire design space Ω
  - Evaluations of objective and constraint could be **coupled** or **uncoupled**

Example #1: Drug/Vaccine Design

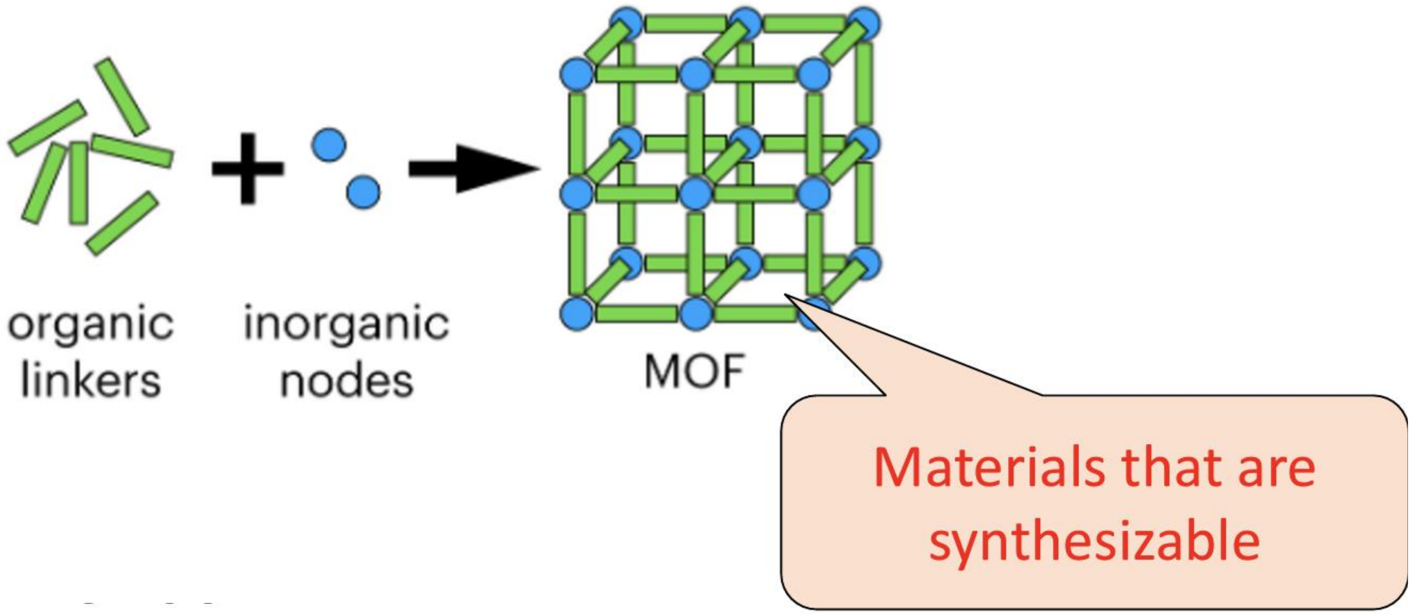


Credit:

MIMA healthcare

- Accelerate the discovery of safe and promising designs

Example #2: Design of Nanoporous Materials



Materials that are synthesizable

- Sustainability applications:
  - Storing gases (e.g., hydrogen powered cars)
  - Separating gases (e.g., CO_2 from flue gas in power plant)
  - Detecting gases (e.g., pollutants in outdoor spaces)

BO with Black-Box Constraints: Key Challenges

- **Modeling challenge:** How to model the black-box constraints?
  - GP models can be straightforwardly trained for each constraint
  - Need to be a bit careful when the objective and/or constraints are correlated or there are a large number of constraints → efficient multi-output GPs

- **Reasoning challenge:** How to select the next input location using simultaneous information from all objective and constraint models?
  - Especially in the case that we have no feasible inputs that satisfy constraints from the previous experiments (also must be careful with observation noise!)

Expected Improvement with Constraints (EIC)

Feasibility indicator function
= 1 if $c(x) \leq 0$ and 0 otherwise

Improvement over our best incumbent value $f_n^* = \min_{i=1,\ldots,n} f(x_i)$ s.t. $c(x_i) \leq 0$

$$\begin{aligned}\mathrm{EIC}(x) &= \mathbb{E}_n\{\mathbf{1}_{\{c(x)\leq 0\}}(x)\max\{0, f_n^\star - f(x)\}\} \\ &= \mathbb{E}_n\{\mathbf{1}_{\{c(x)\leq 0\}}(x)\}\mathbb{E}_n\{\max\{0, f_n^\star - f(x)\}\} \\ &= \underbrace{\Pr_n\{c(x) \leq 0\}}_{\Phi\left(-\frac{\mu_n^c(x)}{\sigma_n^c(x)}\right)}\underbrace{\mathrm{EI}_n(x)}\end{aligned}$$

Conditional independence of objective and constraints*

Standard expected improvement value that has analytic solution (shown on previous slide)

Probability of feasibility;
analytic solution available for GP model using normal CDF

*Only holds when constraints / objective are independently modeled

Gardner et al., 2014.

Expected Improvement with Constraints (EIC)

- If a best feasible incumbent $f^\star$ exists, we can use previous expression

$$\mathrm{EIC}(x; f^\star, \mathcal{D}) = \mathrm{EI}(x; f^\star, \mathcal{D}) \prod_{i=1}^{L} \mathbb{P}(c_i(x) \le 0 | \mathcal{D})$$

- If $f^\star$ does not exist, EIC is not well-defined. This issue is often addressed by maximizing the probability of constraint satisfaction $\prod_{i=1}^{L} \mathbb{P}(c_i(x) \le 0 | \mathcal{D})$
  - – This is not ideal behavior, as it ignores learning about f in these iterations

- Suffers from a pathology when dealing with equality constraints $c_i(x) = 0$
  - – Must convert to two-sided inequality → $\mathbb{P}(c_i(x) \le 0 | \mathcal{D}) \cdot \mathbb{P}(c_i(x) \ge 0 | \mathcal{D})$, which can be extremely small especially when no feasible solutions known

Exact Penalty Bayesian Optimization (EPBO)*

- **Idea:** Take an LCB (optimistic) perspective of the problem by relaxing the objective and constraints with high probability
  - Can build upon theory for deterministic constrained optimization where certain types of penalty functions are known to preserve *exact* solutions
  - Extend standard LCB theory to prove convergence to *constrained* global optima

- **Advantages:**
  - Does not depend on a potentially undefined incumbent $f^\star$
  - Naturally handles both equality and inequality constraints
  - Theoretical convergence guarantees, even in the presence of noise

***C. Lu and J.A. Paulson. "No-Regret Bayesian Optimization with Unknown Equality and Inequality Constraints using Exact Penalty Functions." *IFAC-PapersOnLine*, 2022.**

EPBO Acquisition Function: Penalized Lower Confidence Bound

Convergence holds for sufficiently large ρ, can set $\rho = \infty$ in practice (hard constraints)

$$\begin{aligned}
\min_{x,\epsilon} \quad & \mu_{f,t}(x) - \beta_{t+1}^{1/2}\sigma_{f,t}(x) + \rho\|\epsilon\|_1, \\
\text{s.t.} \quad & |\mu_{h,t}(x)| - \beta_{t+1}^{1/2}\sigma_{h,t}(x) \le \epsilon_h, \\
& \mu_{g,t}(x) - \beta_{t+1}^{1/2}\sigma_{g,t}(x) \le \epsilon_g, \\
& \epsilon = [\epsilon_h^\top, \epsilon_g^\top]^\top \ge 0, \\
& x \in \Omega,
\end{aligned}$$

No penalty whenever we can find a mean prediction <u>within a confidence band</u> set by the predicted standard deviation

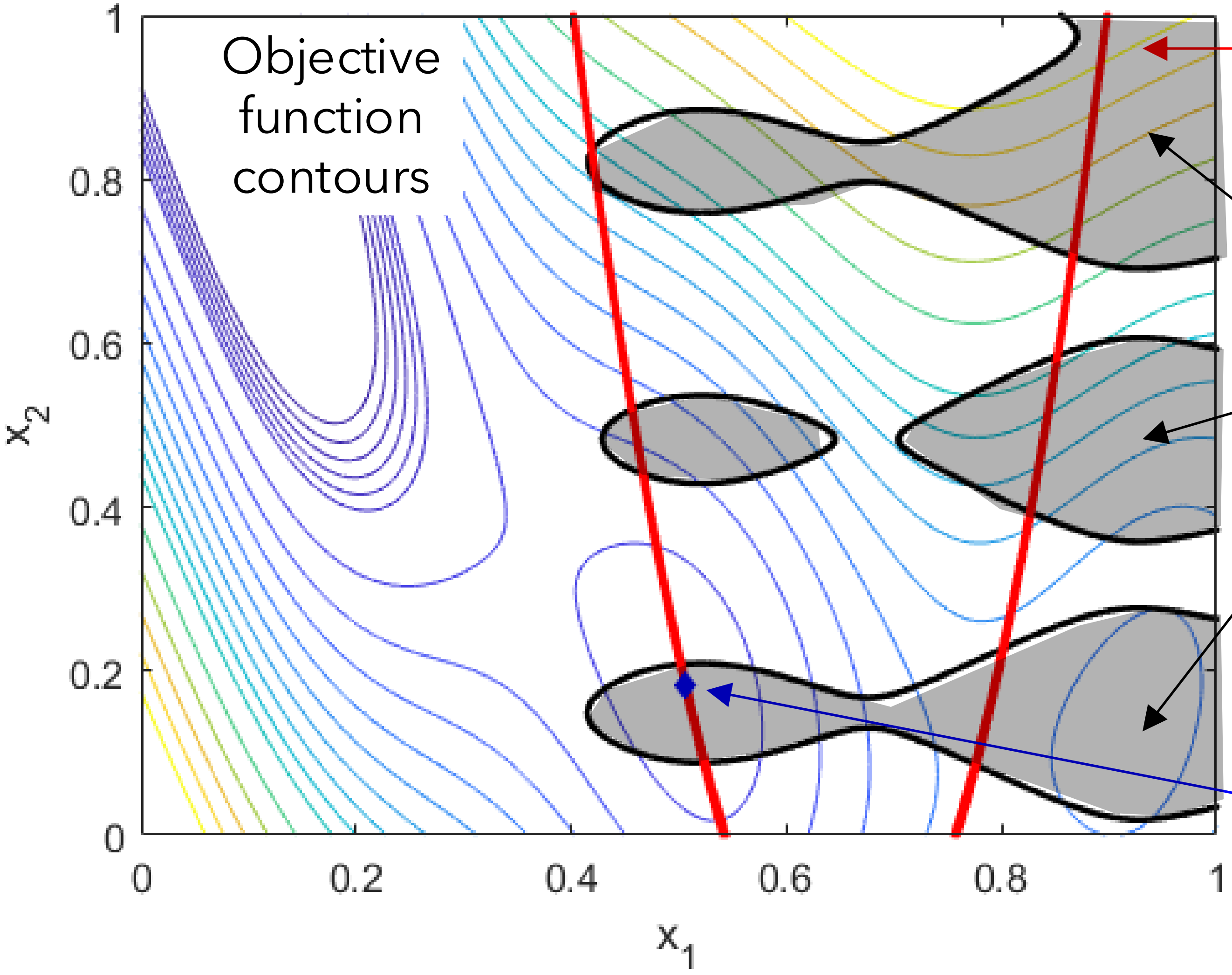
$$\mu_{h,t}(x) \in \sqrt{\beta_{t+1}}[-\sigma_{h,t}(x), \sigma_{h,t}(x)]$$

We can interpret inequality constraint representation as a <u>high probability relaxation</u> of the true feasible region

$$\{x : g(x) \le 0\} \subseteq \{x : \mu_{g,t}(x) \le \beta_{t+1}^{1/2}\sigma_{g,t}(x)\}$$

Example Problem with Equality and Inequality Constraints:

Modified Branin Function



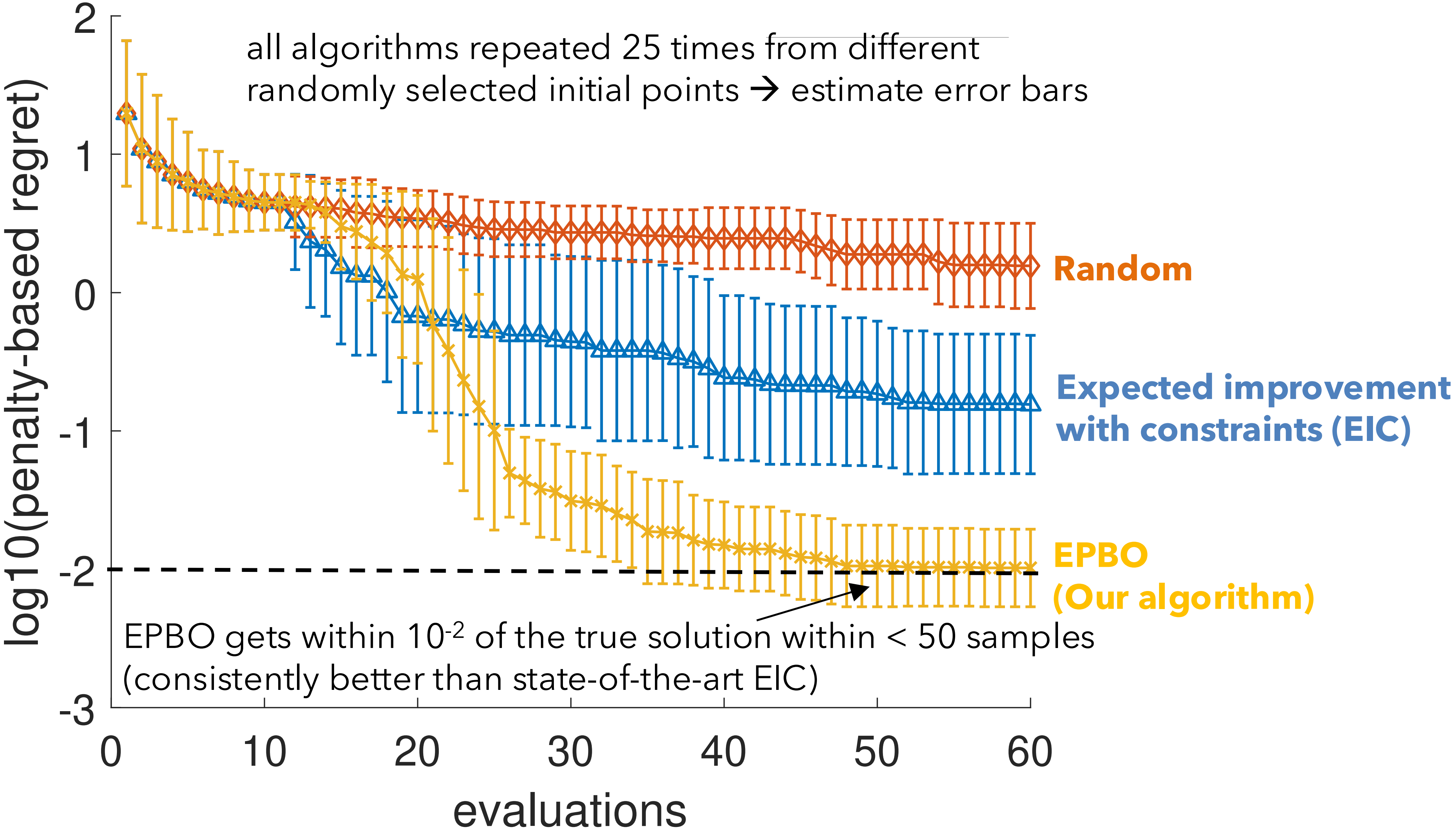
Objective function contours

Equality constraint

Non-convex inequality constraint

True (unknown) global minimum

EPBO versus EIC: Results on Modified Branin Function



all algorithms repeated 25 times from different randomly selected initial points → estimate error bars

EPBO gets within 10^{-2} of the true solution within < 50 samples (consistently better than state-of-the-art EIC)

Can we use information theory for the constrained setting?

- Yes, PES with constraints (PESC) showed good performance on problems related to meta-optimization of machine learning and sampling methods

Now this is *constrained* optimizer

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

Vector of all evaluations

- Advantage: Nicely handles the *decoupled evaluation* case wherein $\alpha(\cdot)$ becomes a sum of individual acquisition functions (one per function)
  - For example, imagine you are designing a cookie recipe and want to find the lowest calorie cookie possible that meets a constraint on the take (>95% of test subjects like it)
  - Much easier for us to estimate calories (objective) than the taste (constraint) function, and these two functions can be evaluated at different inputs

- Disadvantage: Main disadvantage is its difficultly to implement, as it involves several approximations that can become numerically unstable

Outline

- Introduction to Bayesian optimization
  – White-box vs. black-box, prevalence of expensive functions, bird’s eye view

- Improvement-based acquisition functions
  – Expected improvement, knowledge gradient

- Information-theoretic acquisition functions
  – Predictive, max-value, and joint entropy search

- Constrained Bayesian optimization
  – Expected improvement with constraints, exact penalty methods

- Practical considerations
  – Optimizing the acquisition function, kernel adaptation, acquisition scheduling

Two Major Tasks at Each Iteration in BO

1. Train hyperparameters of Gaussian process model(s)
   - The more accurate estimates we achieve for kernel + hyperparameters, the better the decision we can make for the next sample
   - In practice, we re-optimize the hyperparameters at every iteration (given the most recent data) by maximizing the log-likelihood function
   - A trick to reduce cost is to “warm start” the initial guess for the hyperparameters, which works well once they have roughly “stabilized”

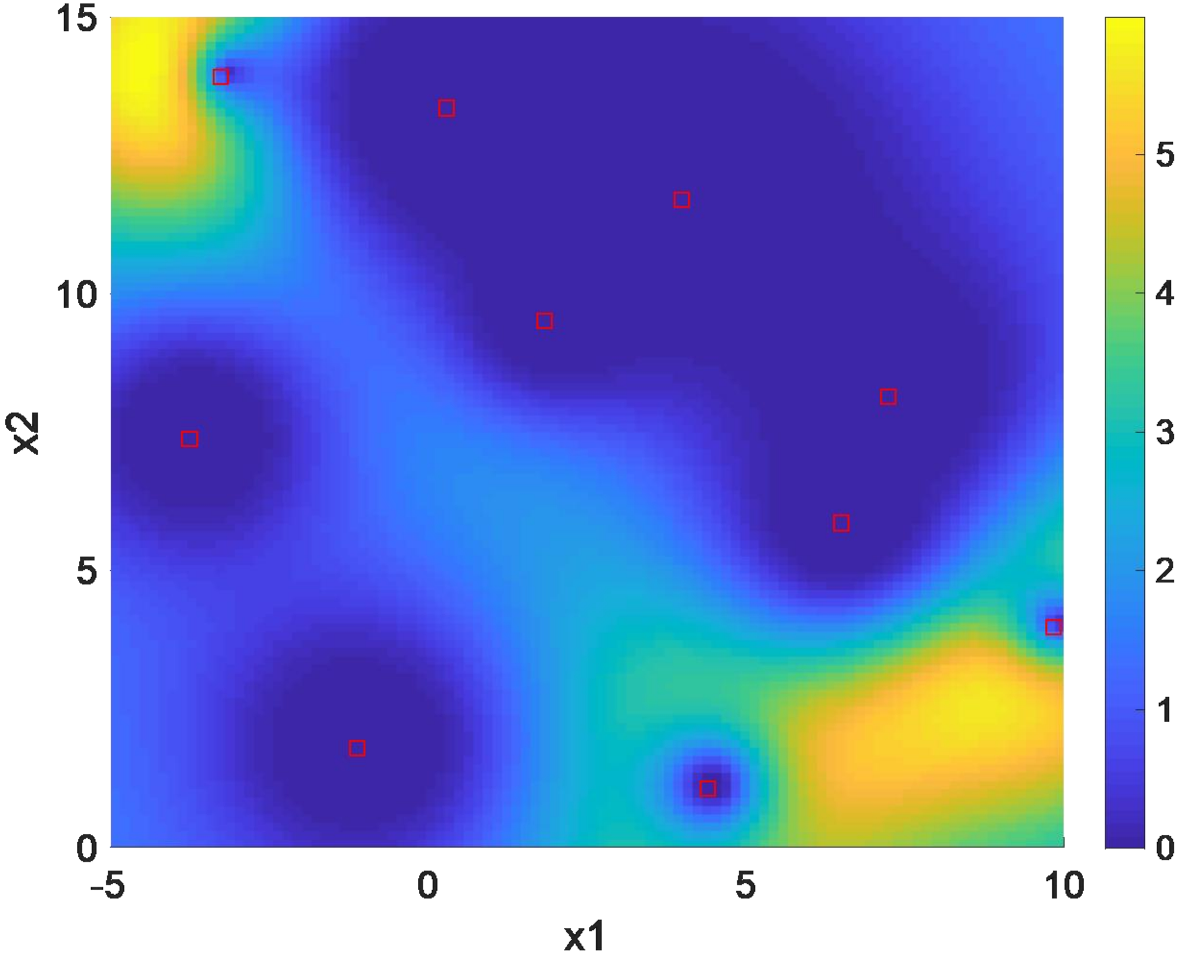
2. Maximize the acquisition function, $x_{n+1} \in \mathrm{argmax}_{x \in \Omega}\, \alpha_n(x)$
   - Many methods exist, no consensus in literature (use your favorite method)
   - Complexity of GP model + operators in acquisition function both important

Common Practice for Maximizing Acquisition Function

- We expect $\alpha_n(x)$ to be non-convex but almost always differentiable

- Easily apply local optimization methods (e.g., L-BFGS, IPOPT)

- Important to perform some type of multi-start procedure to globalize
  - How effective is this method?



How to Measure Performance?

- Most BO papers use simple regret, which is the minimum over the recommended point after a finite number of iterations

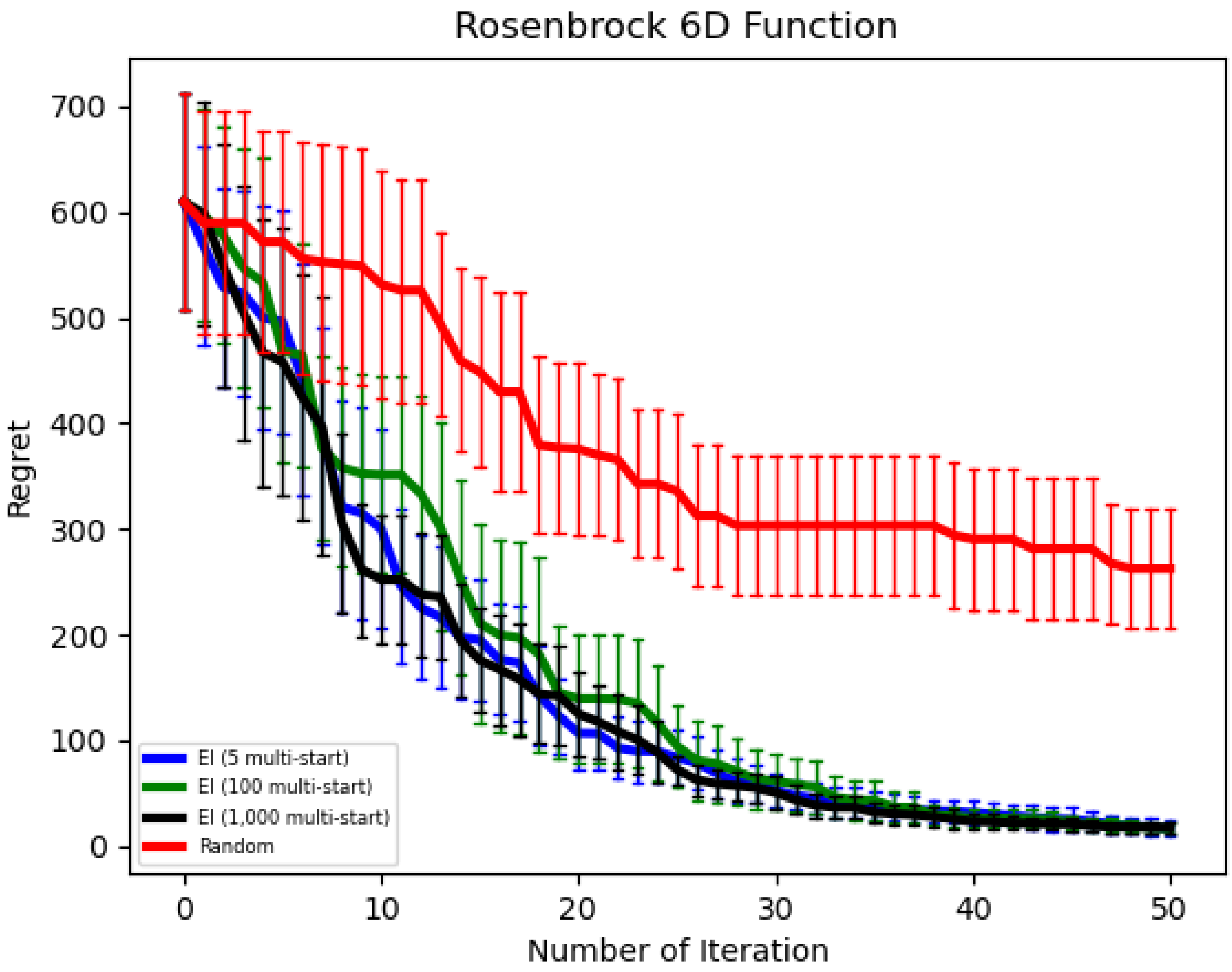
$$\text{SimpleRegret}_n = \min_{i \in \{1,\ldots,n\}} r_i = \min_{i \in \{1,\ldots,n\}} \{ f(x_i) - \underline{f(x^\star)} \}$$

Use best known solution when true solution unknown

- Since the regret sequence depends on the initial data, it is very common to do 50-100 replicates of the entire procedure (with initial data randomly generated) to assess average performance
  - We cannot conclude that every run performs – not obvious what the distribution of performance is either, so often do not try to estimate

Common Practice for Maximizing Acquisition Function

[100 replicates over initial data]



random

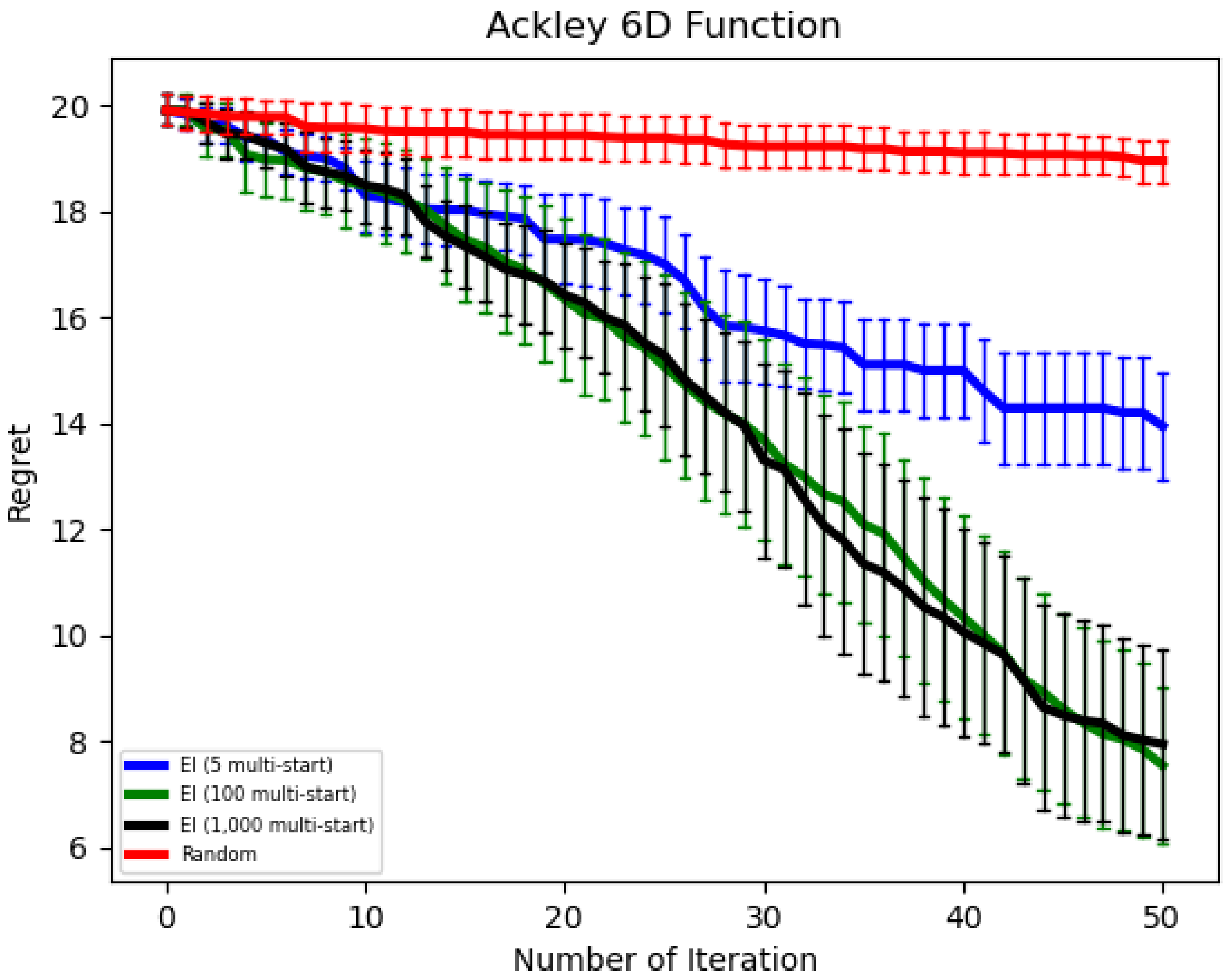
5 restarts

100 restarts

1000 restarts

Common Practice for Maximizing Acquisition Function

[100 replicates over initial data]



random

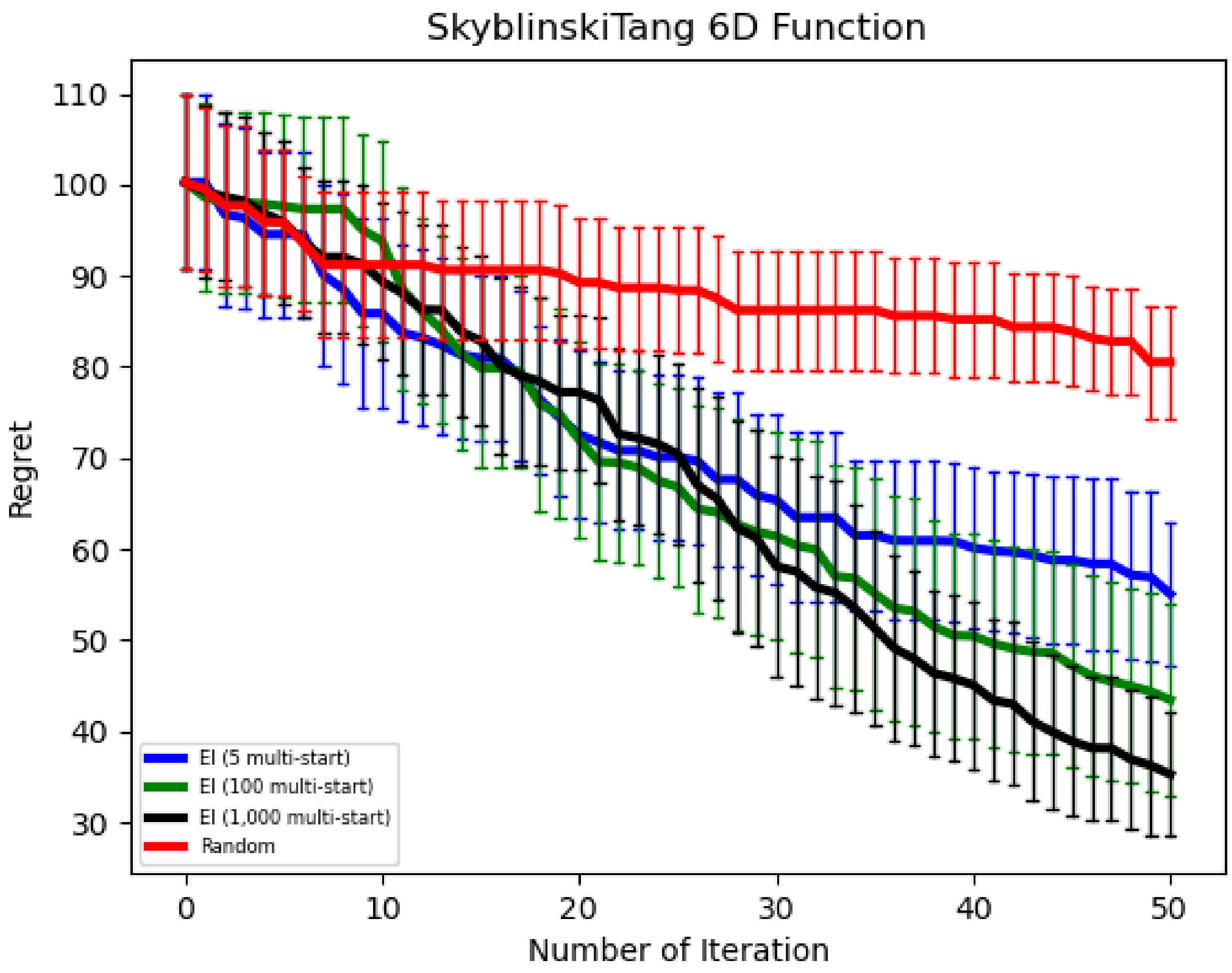
5 restarts

100 restarts

1000 restarts

Common Practice for Maximizing Acquisition Function

[100 replicates over initial data]



random

5 restarts

100 restarts

1000 restarts

Why Not Global Optimization?

Short Answer: Existing Methods Struggle

- McCormick relaxations end up being very weak for the posterior mean and covariance functions since they are the sum over several terms that can have alternating sign

- Would be great if we could (cheaply) construct underestimators that are not too weak → active research area in my group

- Since multi-start is trivially parallelizable (run local solves in parallel), it seems that is best approach to use for now
  - First-order methods, like Adam, also can take advantage of GPU acceleration, so end up having fast wall-times in practice

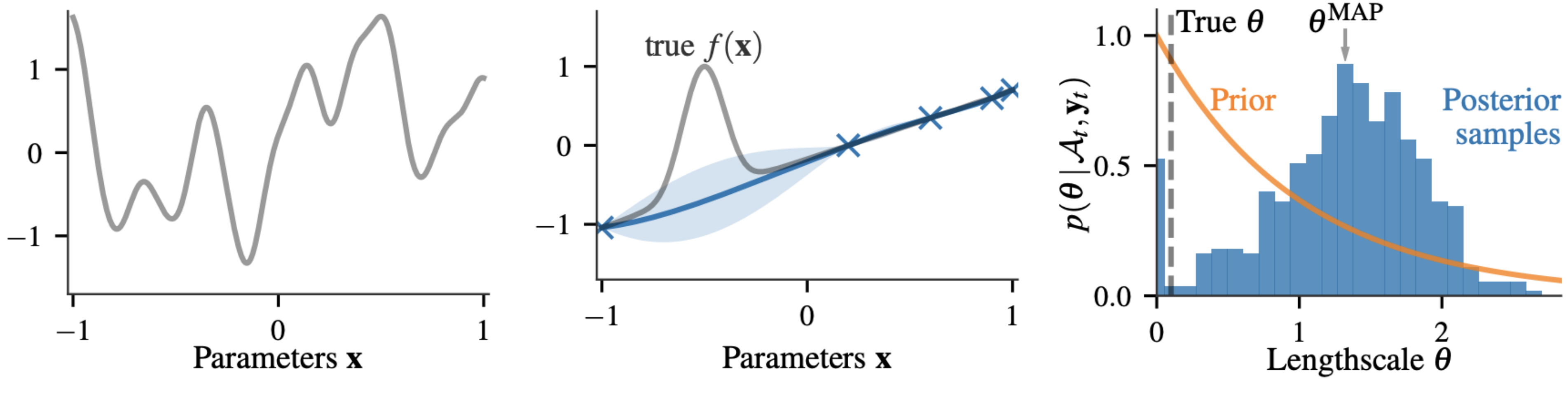
Some Adaptive Modifications to (Potentially) Improve Performance

Dynamic Kernel Selection

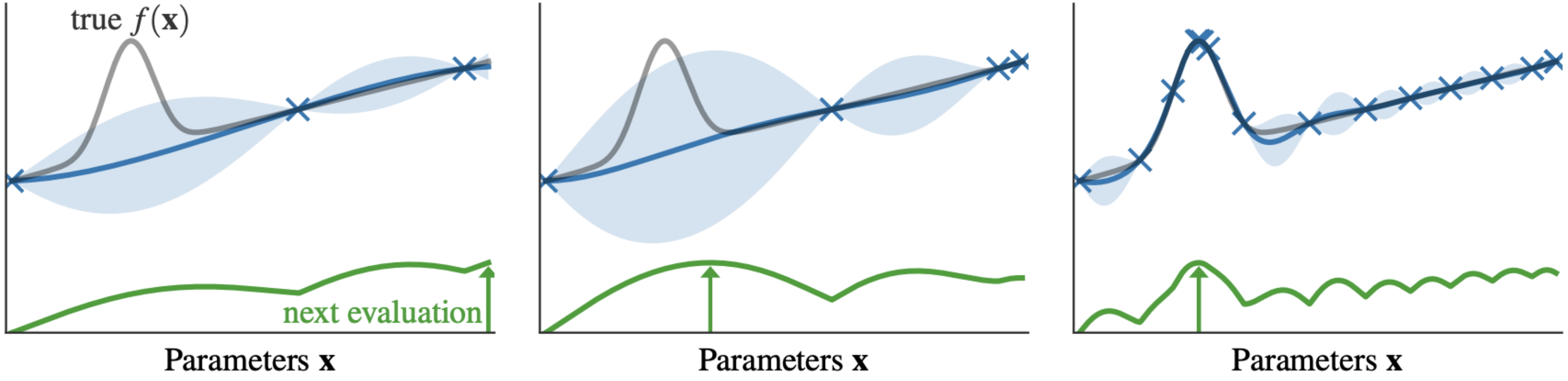
- Normally, we pick a single kernel and keep it fixed at every iteration
- There has been recent work suggesting that some performance gains can be obtained by training multiple GP models (with different kernels) and using some criteria to dynamically select the one to use at each iteration
- Heuristic in nature, so more work needed to find best ways to systematically select between GP models → usually based on some random select process to induce more exploration in the BO process

Roman, Ibai, et al. *BayesOpt 2014: NIPS Workshop on Bayesian Optimization*, 2014

Adaptation of Kernel Hyperparameters



(a) Sample from GP prior. (b) GP estimate (RKHS). (c) Lengthscale distribution.



(a) Stuck in local optimum. (b) Expanding the function class. (c) Global optimum found.

Berkenkamp, et al. *Journal of Machine Learning Research*, 2019

Dynamic Scheduling of Acquisition Functions

- General practice in the BO literature has been for a practitioner to pick their favorite acquisition and use it for the entire optimization process

- Relatively recent work has discussed the value of adopting an adaptive sampling that chooses different acquisition functions at different iterations instead of attempting to pick "the best one"

- Different sampling strategies exist. Simple one is to assume m acquisition functions with weights $\{w_i\}_{i=1}^{m}$. The probability of sampling i^{th} acquisition is $w_i / \sum_{i=1}^{n} w_i$ and, if the acquisition takes a successful move, then the weight is updated

Kandasamy, et al. *Journal of Machine Learning Research*, 2020

Exploitative Guard Against Model Misspecification

[especially for information-theoretic approaches]

- In the paper that introduced the JES acquisition function, noted information-theoretic acquisitions geared toward reduce uncertainty in location of optimum

- Thus, information-theoretic acquisitions are less likely to query the perceived optimum than other approaches like EI, which can greatly impact their performance in the case of a misspecified surrogate model

- To remedy this, they propose to use a γ-exploit approach that implies, with probability γ, the algorithm will query the point that maximizes the posterior mean to confirm its belief on the optimum location
  - If the model is misspecified, these exploitative steps help the algorithm to reconsider its beliefs rather than continuing to act based on faulty ones → suggest $\gamma = 0.1$

Hvarfner, Hutter, and Nardi, Joint Entropy Search for Maximally-Informed Bayesian Optimization, *NeurIPS*, 2022

Extra material (if time permits)*

Knowledge Gradient Maximization [More Advanced Optimization]

Let’s recall the Knowledge Gradient (KG) acquisition function

- Earlier in this slide deck, we saw that the KG acquisition function is:

$$\begin{aligned} \mathrm{KG}_n(x) &= \mathbb{E}_n\{\mu_n^\star - \mu_{n+1}^\star | x_{n+1} = x\} \\ &= \mathbb{E}_n \left\{ \cancel{\mu_n^\star} - \min_{x' \in \Omega} \mu_{n+1}(x') | x_{n+1} = x \right\} \end{aligned}$$

- Our goal is to maximize this function, so can ignore constant and convert the max to a min (due to the -1)

Maximization of KG is a two-stage stochastic program

$$\min_{x\in\Omega}\mathbb{E}_n\left\{\min_{x'\in\Omega}\mu_{n+1}(x')|x_{n+1}=x\right\}$$

How can we express the expectation in terms of things we can compute?

$$\mu_{n+1}(x')=\mu_n(x')+\tilde{\sigma}_n(x',x_{n+1})Z, \qquad Z\sim\mathcal{N}(0,1)$$

$$\tilde{\sigma}_n(x,x')=\frac{k_n(x,x')}{\sqrt{k_n(x',x')+\sigma^2}}$$

Maximization of KG is a two-stage stochastic program

$$\min_{x_{n+1}\in\Omega} \mathbb{E}_Z \left\{ \min_{x'\in\Omega} \{\mu_n(x') + \tilde{\sigma}_n(x', x_{n+1}) Z \right\}$$

- Two main approaches for solving this problem:

1. Stochastic gradient descent (SGD) (+ envelope theorem)
2. Sample average approximation (SAA)

this has become more popular recently

Sample average approximation for KG acquisition

$$\min_{x_{n+1}\in\Omega} \frac{1}{N}\sum_{i=1}^{N}\left\{\min_{x^{(i)}\in\Omega}\{\mu_n(x^{(i)})+\tilde{\sigma}_n(x^{(i)},x_{n+1})Z_i\right\}$$

↓

The point that minimizes the next mean function depends on our sample selection

$$\min_{\underset{\text{“here-and-now”}}{x_{n+1}},\underbrace{x^{(1)},\dots,x^{(N)}}_{\text{“wait-and-see”}}\in\Omega} \frac{1}{N}\sum_{i=1}^{N}\{\mu_n(x^{(i)})+\tilde{\sigma}_n(x^{(i)},x_{n+1})Z_i\}$$