Bayesian Optimization with a Budget and Unknown Costs

Daniel Jiang
joint work with Raul Astudillo (former intern), Max Balandat, Eytan Bakshy, Peter Frazier

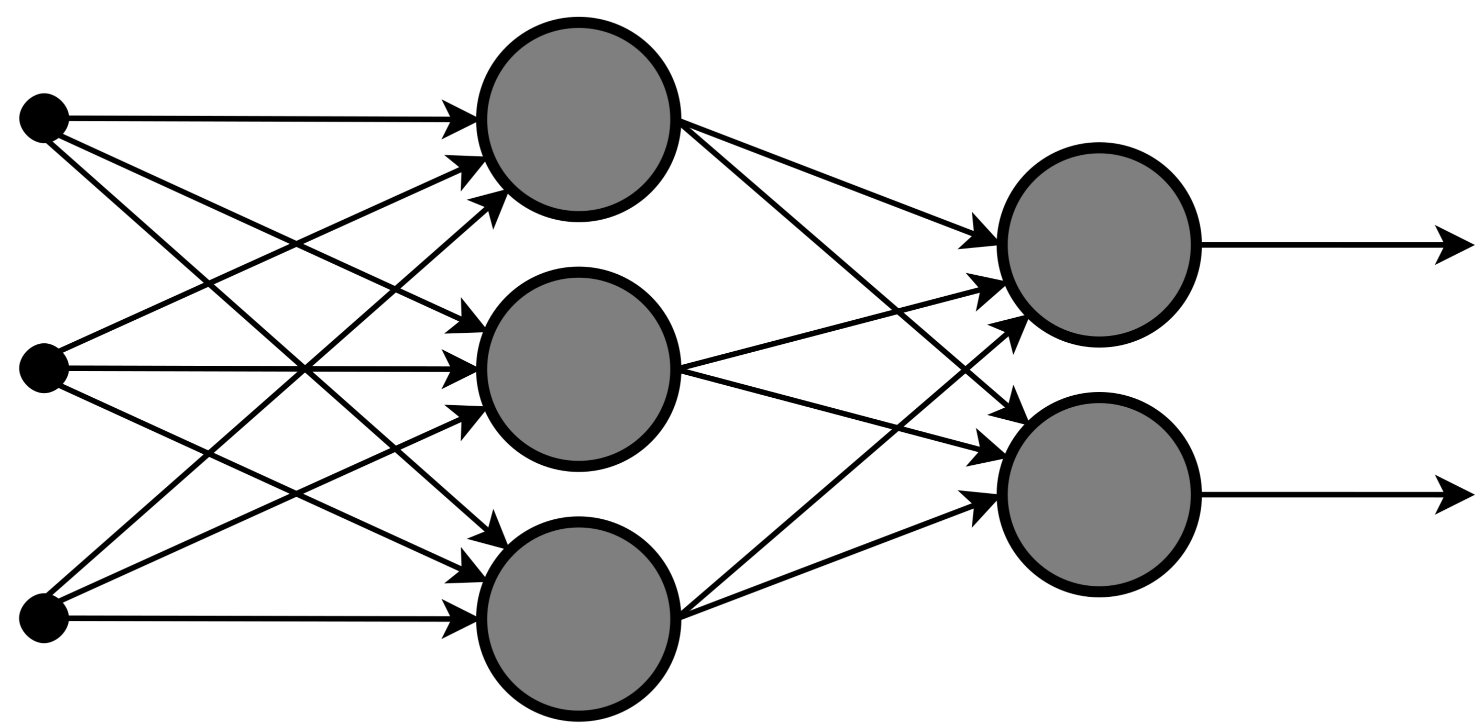
Introduction to the Bayesian optimization problem

- We have an **expensive-to-evaluate** objective function $f : \mathbb{R}^d \to \mathbb{R}$
  - Typically, assume that f is a continuous function, but otherwise no structure
  - No gradients
  - Evaluations can either be **exact** or noisy, i.e., $y = f(x) + \epsilon$
- Goal:
  - $\max_{x \in \mathbb{X}} f(x)$, where $\mathbb{X} \subset \mathbb{R}^d$ is usually not too complicated (box bounds)
  - Small number of evaluations (think 20 to 60)
- When does this type of problem arise?

Resources: Frazier, 2008; Ax documentation

Optimizing ML algorithms (AutoML)

- Hyper-parameter tuning of ML models [1]
  - Learning rates, momentum parameters
- Neural network architectures
- One of the primary applications of BO in industry



[1] Snoek et al., 2012

Online experimentation (A/B testing)

- Compare multiple versions of a system by running an experiment [2, 3]
- Despite the name “A/B-test,” usually need to search through many more than just two variants of the system
  - E.g., value model tuning for ranking/recommendation algorithms [3]
- Experiments can run for days or even weeks (costly in terms of time)
- Testing a new feature can lead to loss of revenue (costly in terms of $$)

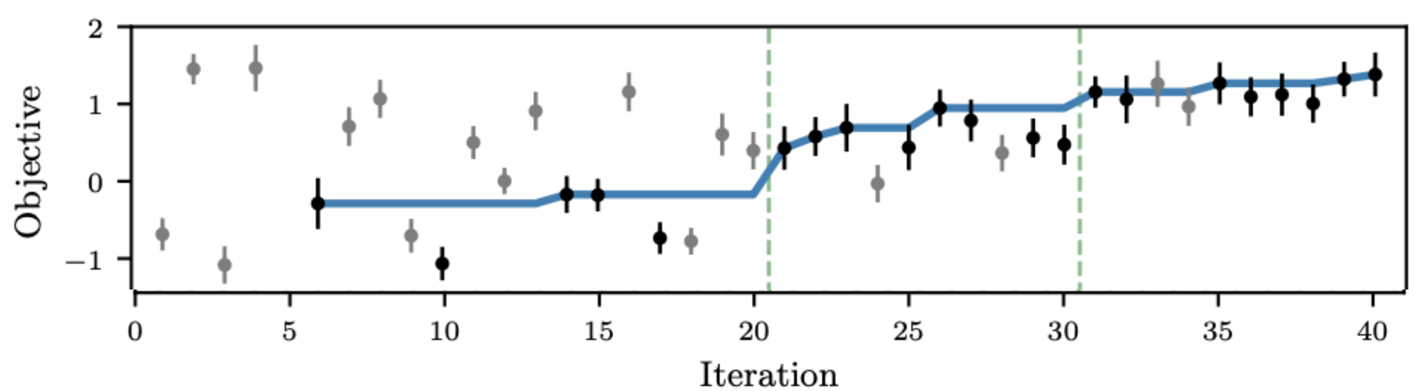


Figure from [3]

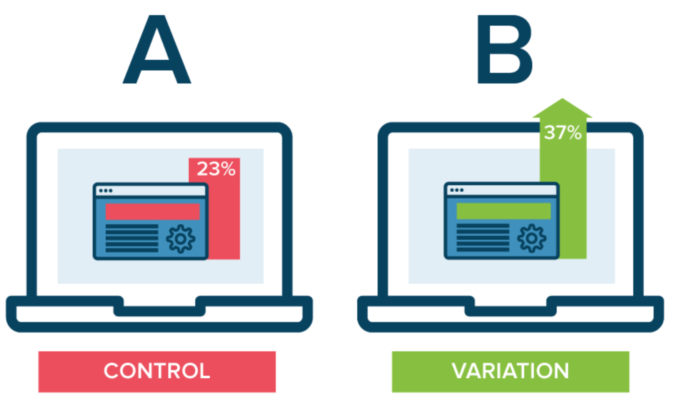


Figure from [4]

[2] Letham et al., 2019, [3] Letham & Bakshy, 2019, [4] optimizely.com

Expensive physics simulations

- Coverage and capacity optimization in cellular networks
  - Decision is where to place cell towers such that coverage is maximized
  - Expensive to simulate a particular configuration of cell tower placement

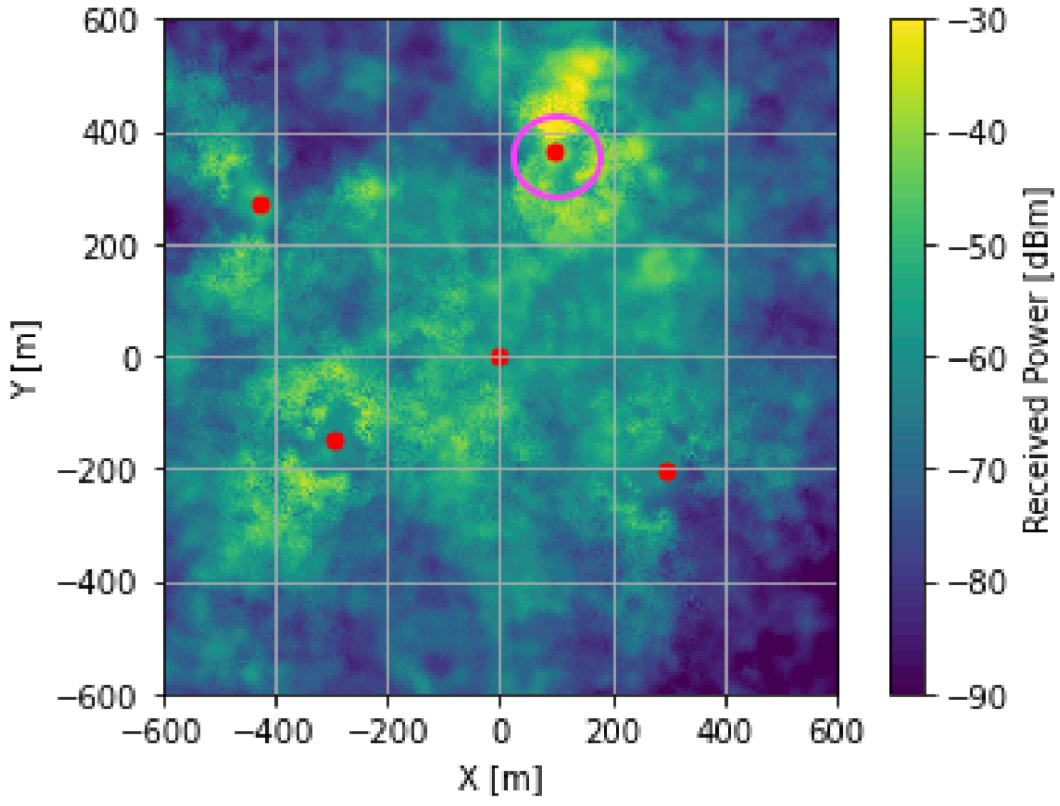


Figure from [5]

[5] Dreifuerst et al., 2021

Lab science experiments (chemistry, biology, materials)

Article | Published: 03 February 2021

Bayesian reaction optimization as a tool for chemical synthesis

Benjamin J. Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I. Martinez Alvarado, Jacob M. Janey, Ryan P. Adams & Abigail G. Doyle

Nature **590**, 89–96(2021) | Cite this article

Discovering high-performance broadband and broad angle antireflection surfaces by machine learning

Sajad Haghanifar,[1] Michael McCourt,[2] Bolong Cheng,[2] Jeffrey Wuenschell,[3] Paul Ohodnicki,[3] and Paul W. Leu[1,4,5,*]

[1]Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, Pennsylvania 15261, USA
[2]SigOpt, San Francisco, California 94104, USA
[3]National Energy Technology Laboratory, U.S. Department of Energy, Pittsburgh, Pennsylvania 15236, USA
[4]Department of Mechanical Engineering and Materials Science, University of Pittsburgh, Pittsburgh, Pennsylvania 15261, USA
[5]Department of Chemical Engineering, University of Pittsburgh, Pittsburgh, Pennsylvania 15261, USA
*Corresponding author: pleu@pitt.edu

Received 10 January 2020; revised 10 May 2020; accepted 13 May 2020 (Doc. ID 387938); published 9 July 2020

Creating glasswing butterfly-inspired durable antifogging superomniphobic supertransmissive, superclear nanostructured glass through Bayesian learning and optimization†

Sajad Haghanifar,[a] Michael McCourt,[b] Bolong Cheng,[b] Jeffrey Wuenschell,[c] Paul Ohodnicki[c] and Paul W. Leu[*ade]

Article | Published: 19 February 2020

Closed-loop optimization of fast-charging protocols for batteries with machine learning

Peter M. Attia, Aditya Grover, Norman Jin, Kristen A. Severson, Todor M. Markov, Yang-Hung Liao, Michael H. Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, Patrick K. Herring, Muratahan Aykol, Stephen J. Harris, Richard D. Braatz, Stefano Ermon & William C. Chueh

Nature **578**, 397–402(2020) | Cite this article

Article | Open Access | Published: 07 December 2018

Discovering de novo peptide substrates for enzymes using machine learning

Lorillee Tallorin, JiaLei Wang, Woojoo E. Kim, Swagat Sahu, Nicolas M. Kosa, Pu Yang, Matthew Thompson, Michael K. Gilson, Peter I. Frazier, Michael D. Burkart & Nathan C. Gianneschi

Nature Communications **9**, Article number: 5253 (2018) | Cite this article

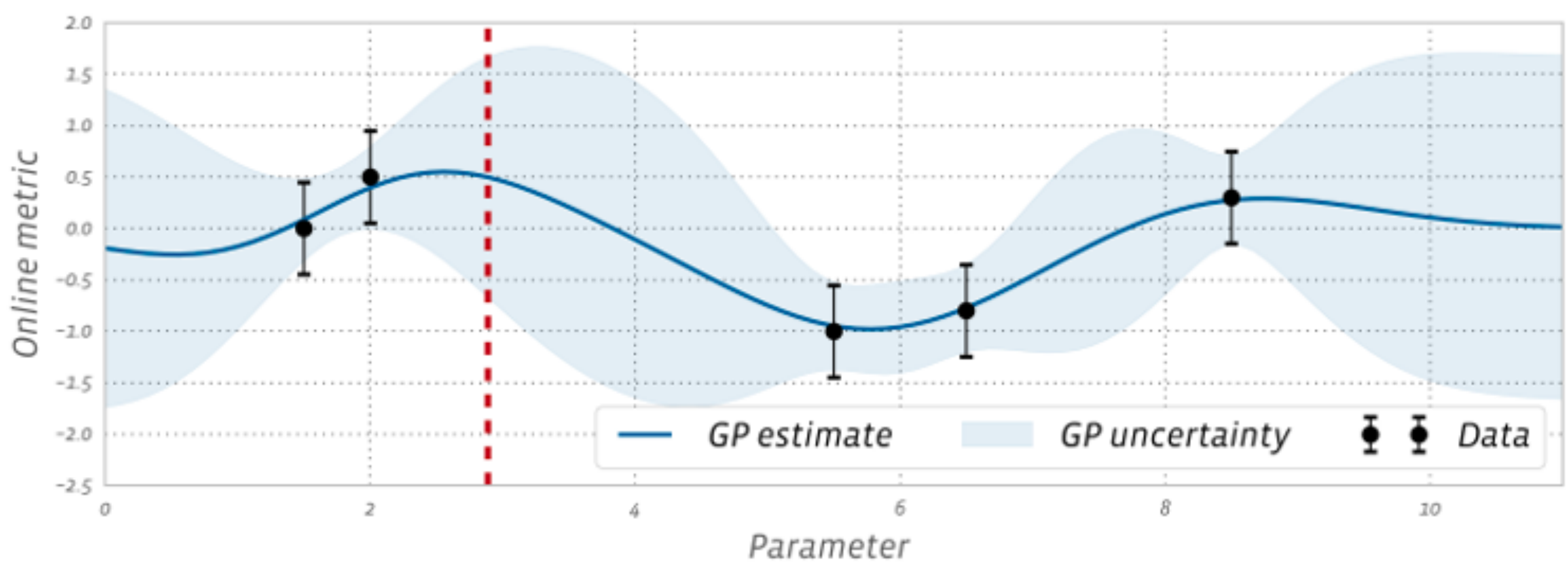
High-throughput in vivo mapping of RNA accessible interfaces to identify functional sRNA binding sites

Mia K. Mihailovic, Jorge Vazquez-Anderson, Yan Li, Victoria Fry, Praveen Vimalathas, Daniel Herrera, Richard A. Lease, Warren B. Powell & Lydia M. Contreras

Nature Communications **9**, Article number: 4084 (2018) | Cite this article

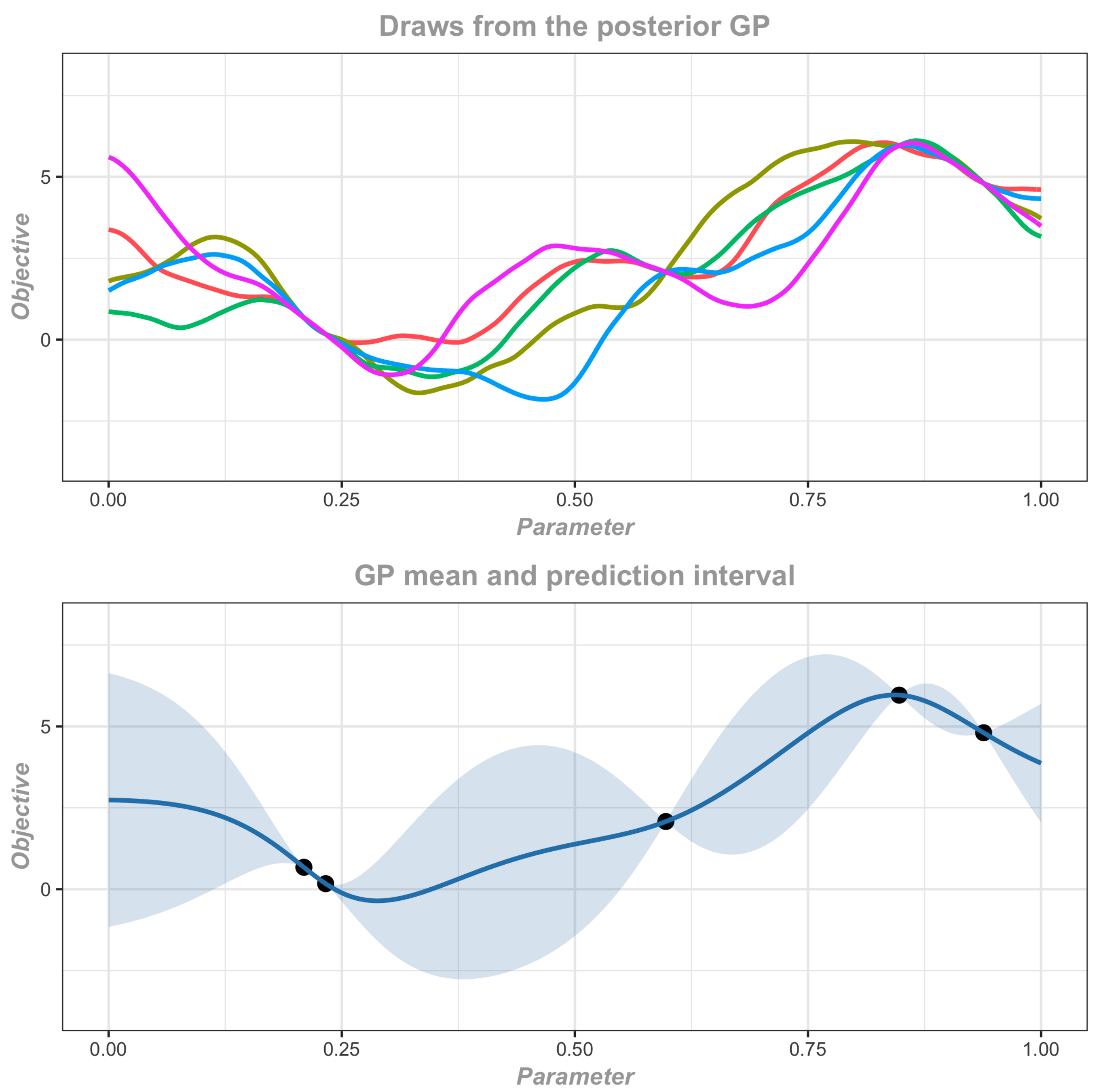
Quick outline of Bayesian optimization (GP model)

- **Step 1:** Build a surrogate model (usually, a Gaussian process) that allows the experimenter to *quantify uncertainty* about their knowledge of the function f given the observed data so far.
  - By doing so, the experimenter can trade-off exploration (trying new points) and exploitation (focusing in around the optimal)



Resources: Ax documentation (ax.dev)

Quick outline of Bayesian optimization (GP model)



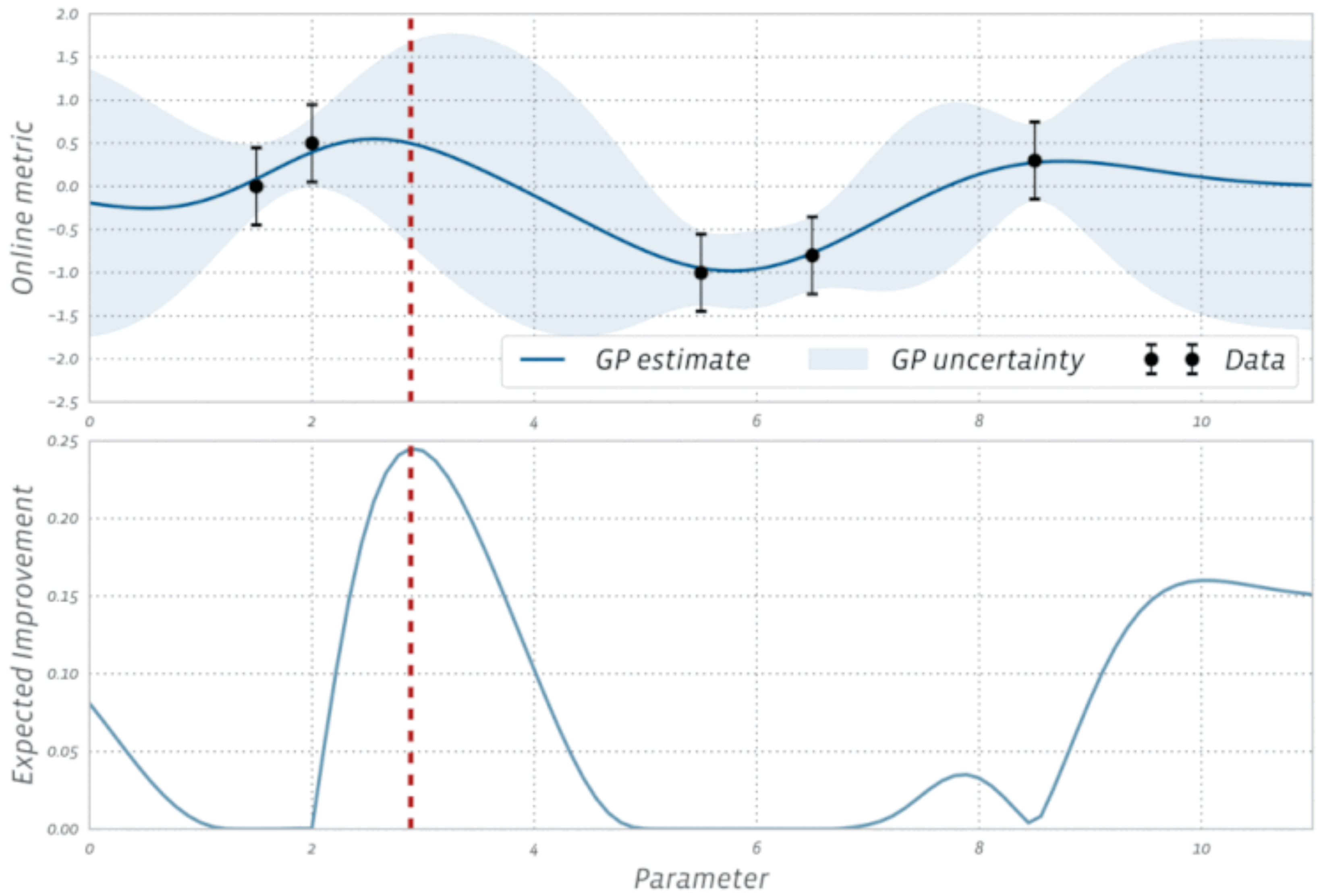
Resources: Ax documentation (ax.dev)

Quick outline of Bayesian optimization (acquisition)

- **Step 2:** Decide where to sample next by maximizing an *acquisition function*. There are many ways to explore; an acquisition function encodes this strategy
  - The acquisition function places value on points and implies a sampling policy
  - Expected improvement is one such possibility: $\mathrm{EI}(x) = \mathbf{E}\left[\max(f(x) - f_{\mathrm{best}}, 0)\right]$, where f_{best} is the best observation so far

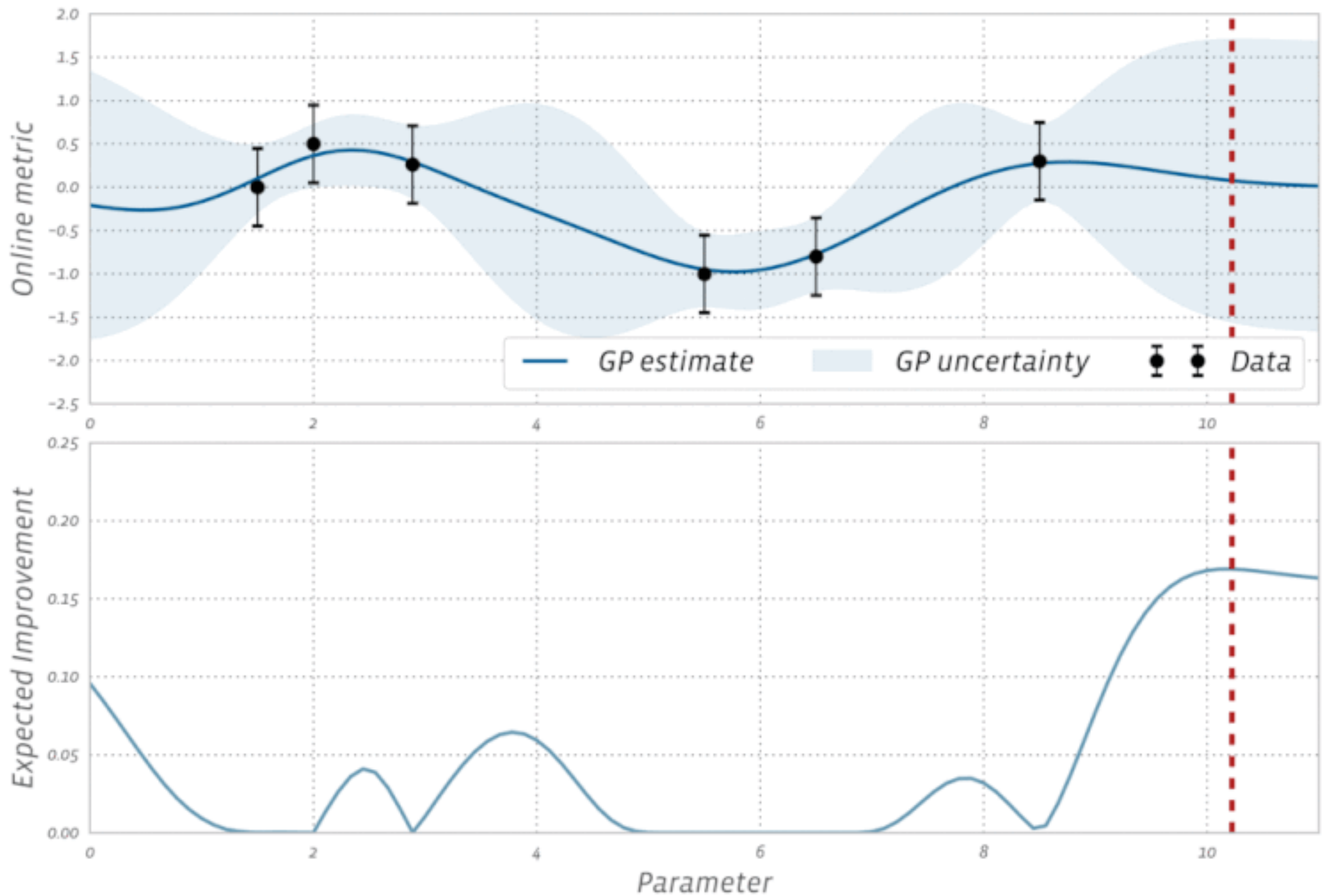
Resources: Ax documentation (ax.dev)

Quick outline of Bayesian optimization (EI)



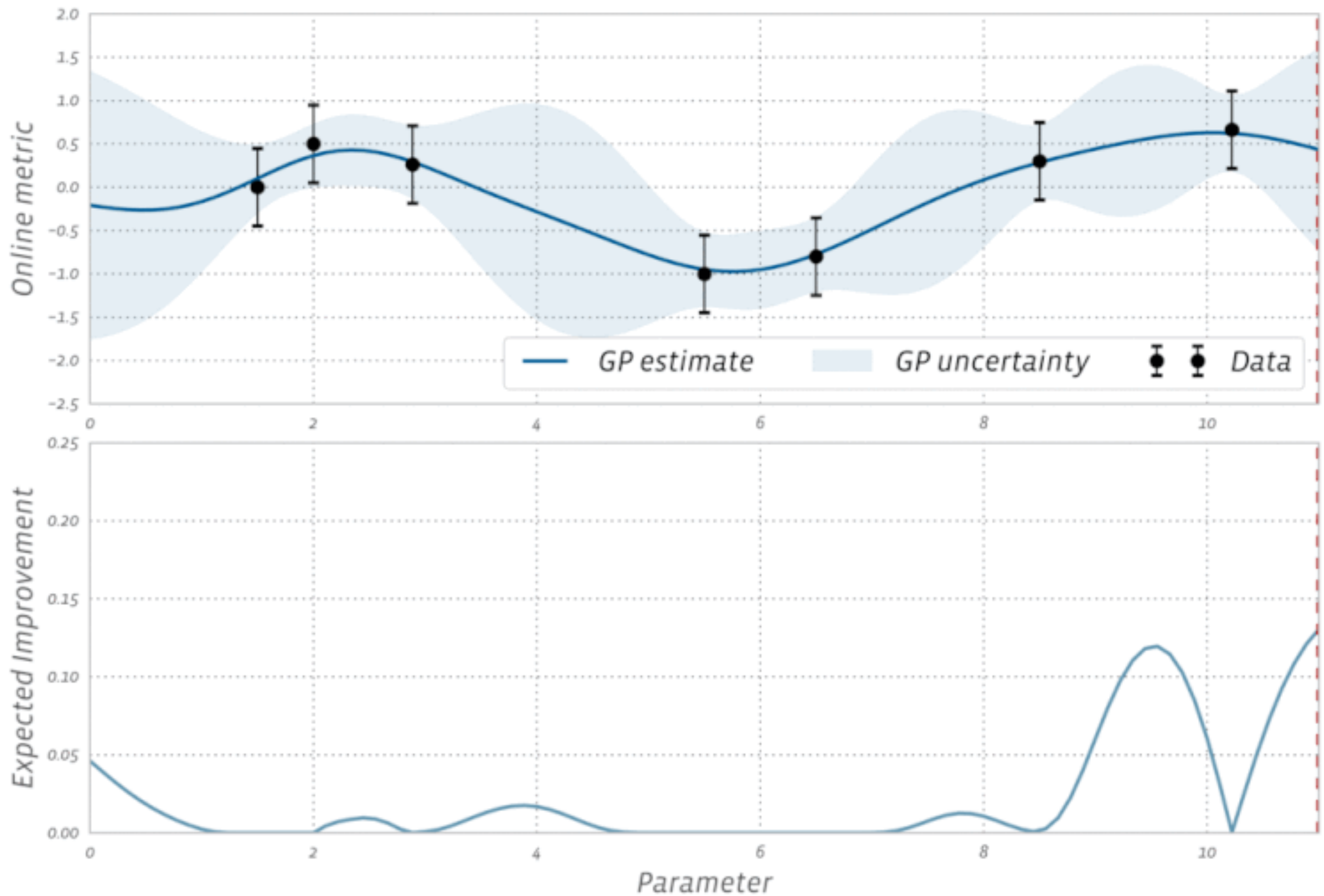
Resources: Ax documentation (ax.dev)

Quick outline of Bayesian optimization (EI)



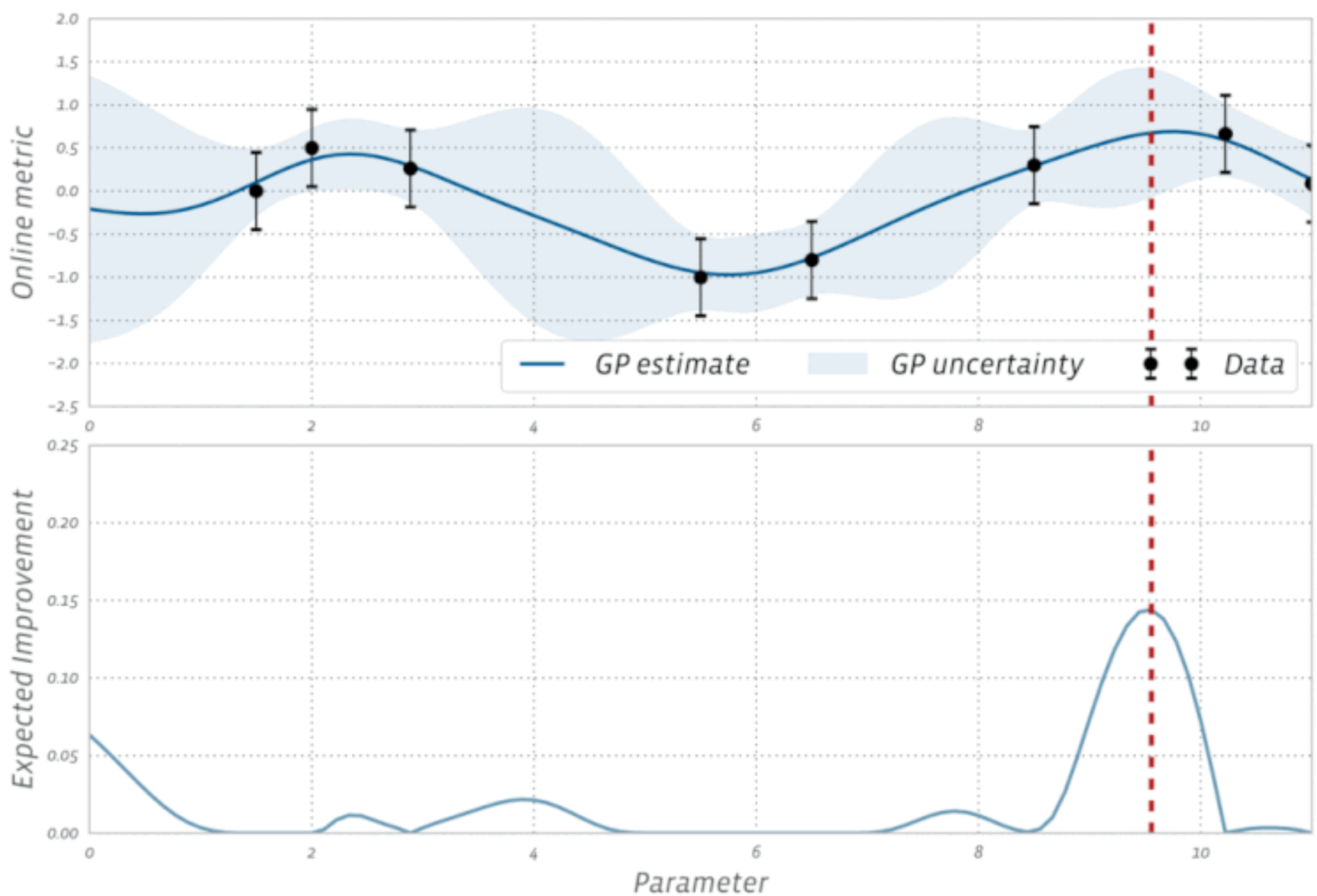
Resources: Ax documentation (ax.dev)

Quick outline of Bayesian optimization (EI)



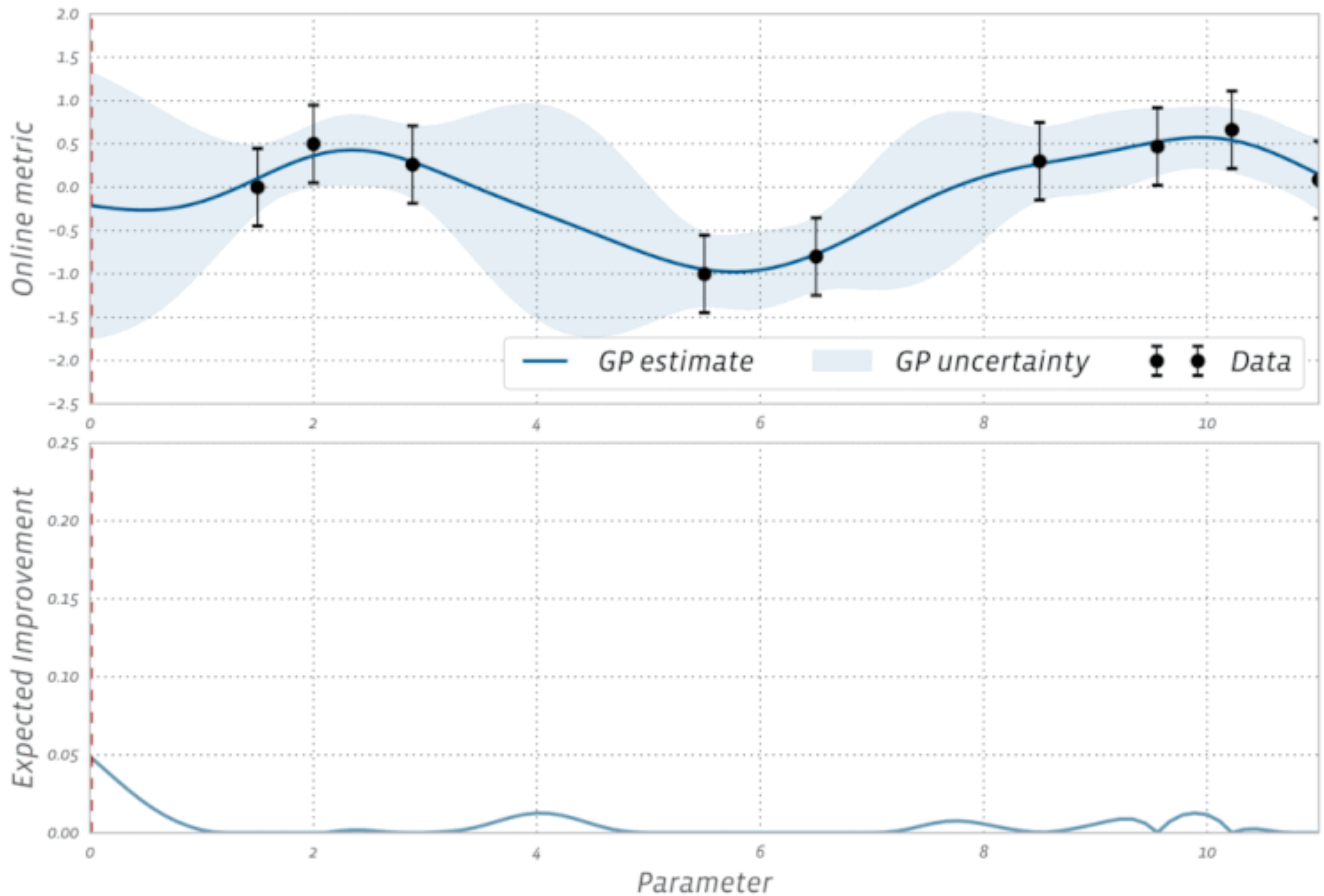
Resources: Ax documentation (ax.dev)

Quick outline of Bayesian optimization (EI)



Resources: Ax documentation (ax.dev)

Quick outline of Bayesian optimization (EI)



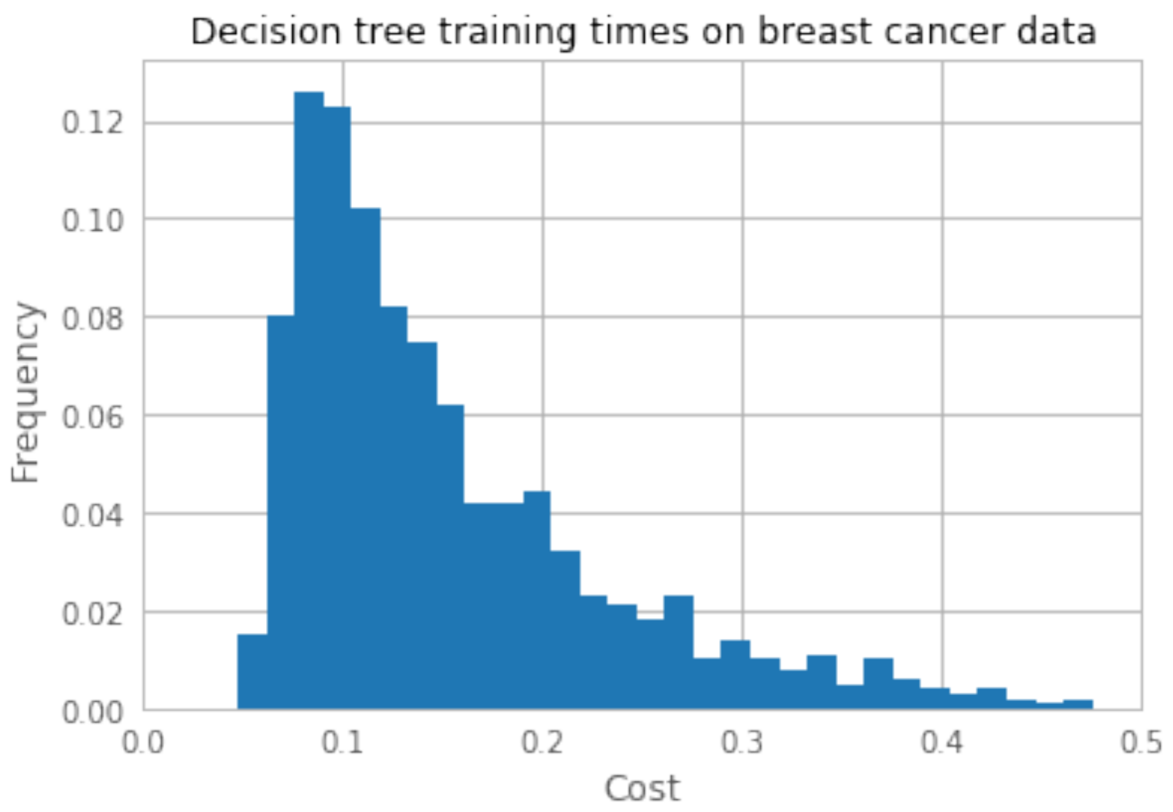
Resources: Ax documentation (ax.dev)

Outline

- ~~Review of Bayesian optimization (BO)~~
- Bayesian optimization with a budget and costs
    - Value-to-cost ratio methods
- Our new acquisition function
    - Optimization of multi-step, differentiable trees
- Numerical results
- Open source code in BoTorch

What is the standard BO formulation missing?

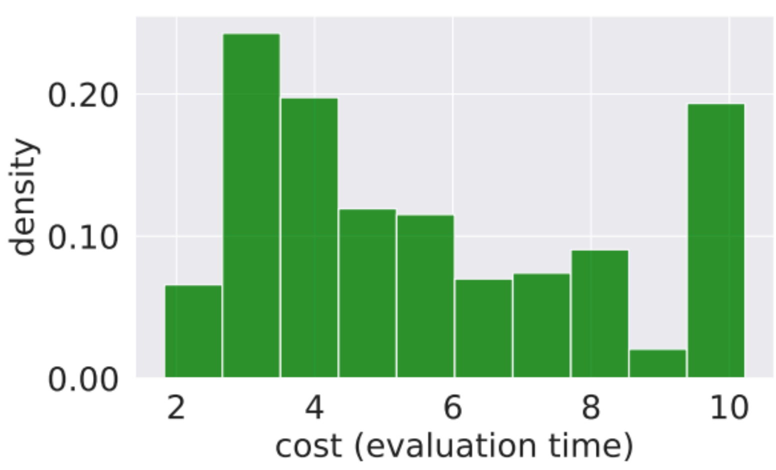
- It assumes that evaluations of f take identical effort (or equally expensive), but… **evaluation costs are often heterogeneous!**



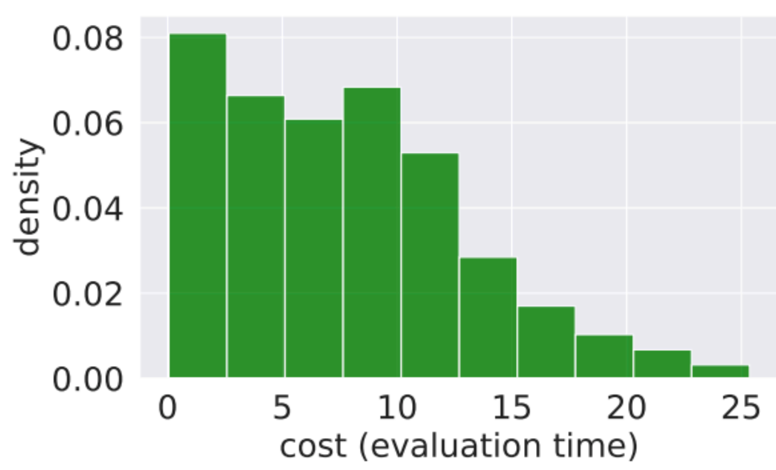
In a sample-efficient setting, exploiting 8-10x cheaper training times is potentially very impactful.

At the same time, we don't necessarily know the cost function a priori.

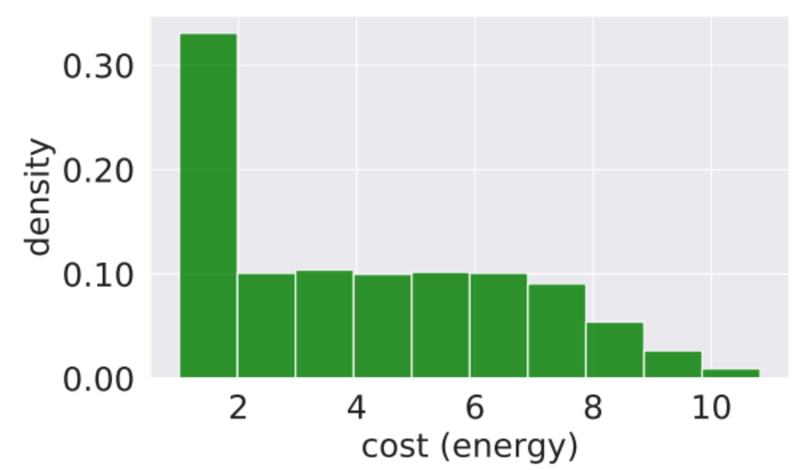
Evaluation times from three open-source examples



Online latent Dirichlet allocation algorithm for topic modeling (2-10 hours!)



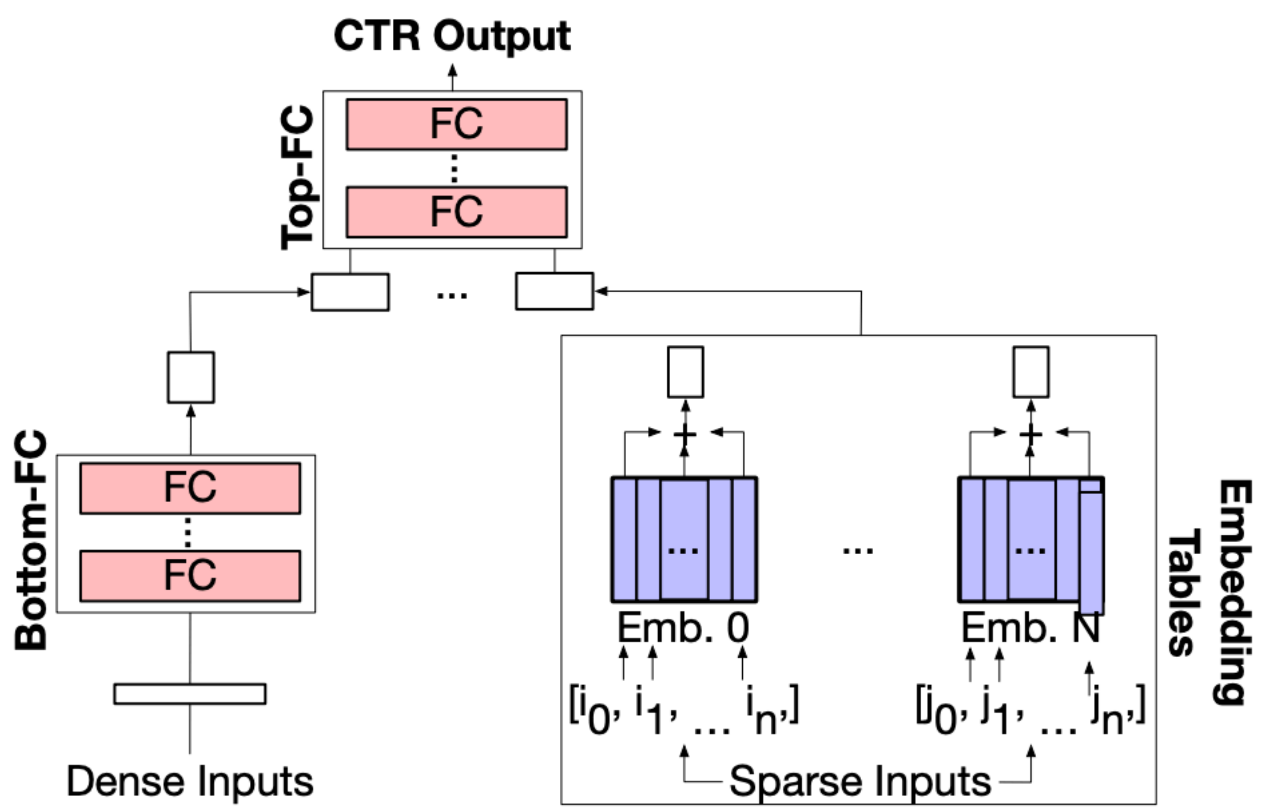
Random forest on the Boston housing dataset



Energy-aware robot pushing benchmark example

Real-world applications where cost is heterogeneous

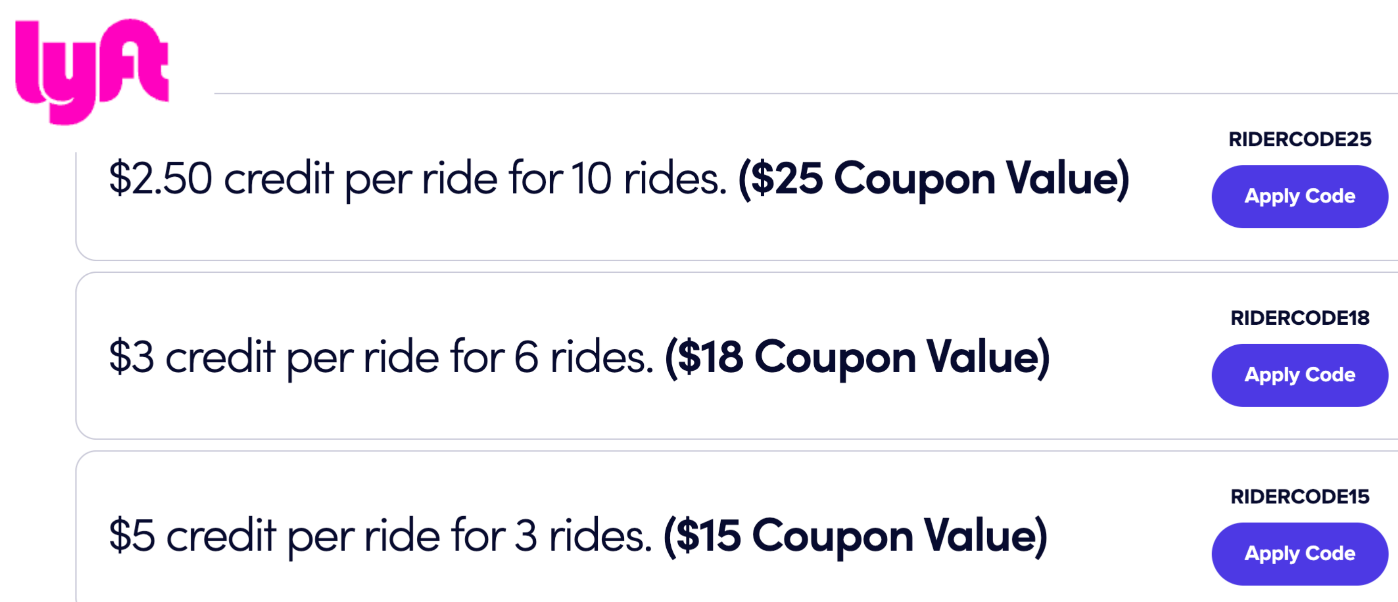
- AutoML for large-scale recommender systems, which consume the majority of inference cycles in Meta's production data centers [6]
- Sparse features are usually converted to dense features via an embeddings.
- Tuning the embedding dimensionality affects the model's training time



[6] Gupta et al., 2020 (https://arxiv.org/pdf/1906.03109.pdf)

Real-world applications where cost is heterogeneous

- In online experimentation, some variants may be more expensive (consider testing prices, coupons, or some other type of promotion)



Our problem: Bayesian optimization with costs / budgets

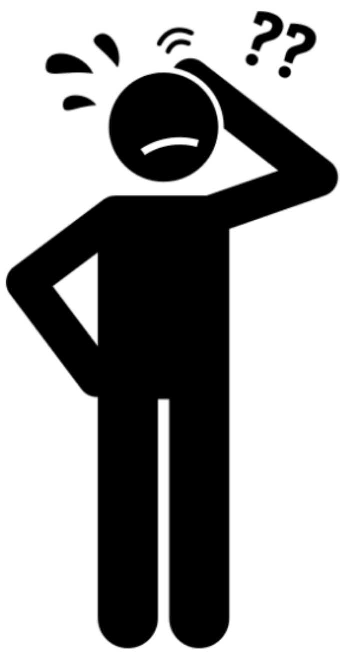
- Consider the standard BO setting of $\max_{x \in \mathbb{X}} f(x)$, but an evaluation of f at a point x incurs a cost $c(x)$
- The cost function $c(x)$ is **also unknown** and must be learned.
  - Our observations after each step is $(x, f(x), c(x))$.
- We care about the best configuration found after some **experimentation budget** B is exhausted:
  - Find a policy that sequentially selects points $\{x_i\}_{i=1}^{N_B}$ such that the best observation is as large as possible and N_B is the last iteration n where the experimentation budget is satisfied: $\sum_i^n c(x_i) \leq B$.

Differences from standard BO

- There is a new dimension to the exploration-exploitation trade-off:
  - Need to **learn the cost function**
- Need to reason about cost-learning through **uncertainty estimates** of the cost
  - Interestingly aspect: learning about the cost incurs the cost itself
- Optimal behavior when the **budget is high** vs when the **budget is low**

Thought experiment

- Consider a point x where, based on the surrogate:
  - $f(x)$ seems good and has significant upside
  - $c(x)$ is medium but with high uncertainty
- Should we evaluate?
  - If the observed $c(x)$ is high, we might exhaust our budget.
  - If the observed $c(x)$ turns out to be low cost, we might have just found a new region of cheap + high performing points.



*These complex features point to a **planning-based solution**, but first let's take detour and look at a simple class of heuristics.*

The “value over cost” paradigm

- All existing work in this area uses a form of **value to cost ratio.**
- Typically, an existing acquisition function is **divided by the cost function** to determine a point that gives the maximum “value” per unit cost:
  - e.g., “EI / cost” samples the point $x^* = \mathrm{argmax}_x \; \mathrm{EI}(x)/c(x)$
  - Variants:
    - $x^* = \mathrm{argmax}_x \; \mathrm{EI}(x)/\mathbf{E}[c(x)],$
    - $x^* = \mathrm{argmax}_x \; \mathrm{EI}(x)/c(x)^{\alpha}$
  - Snoek et al., 2012; Swersky et al., 2013; Kandasamy et al., 2016; 2017; Poloczek et al., 2017; Song et al.,2019; Wu et al., 2020; Lee et al., 2020b
  - Decent performance in many practical settings, but not always

EI / cost can be arbitrarily bad

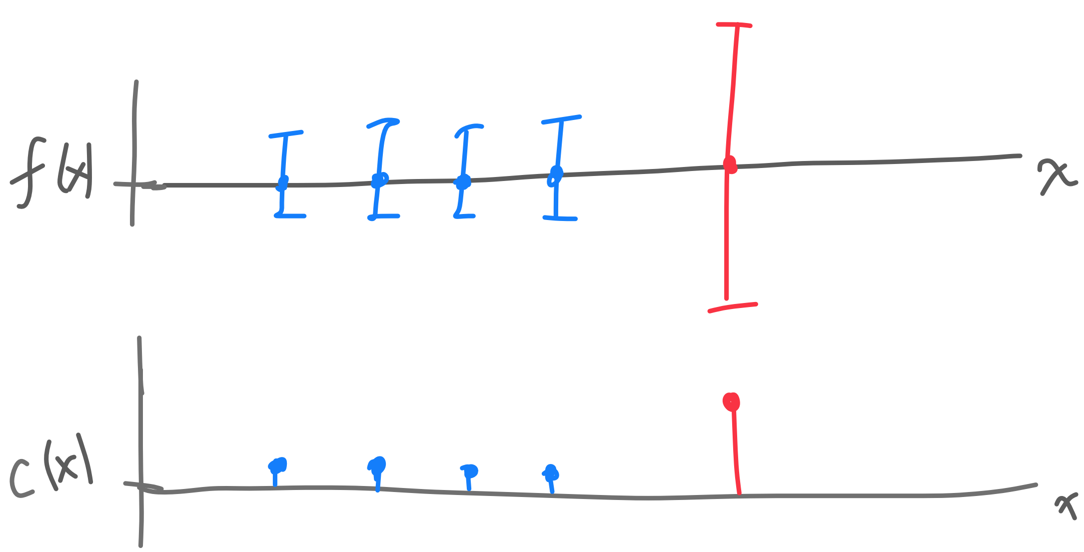
- **Theorem:** *The approximation ratio provided by the EI/cost policy is arbitrarily bad.*
  - Let $\mathcal{S}$ be a set of initial observations $\{(x, f(x))\}$ (let's suppose $c(x)$ is known).
  - Let $V^*(\mathcal{S})$ be the value (expected best value of the function f found within the budget) of the optimal policy and let $V^{\text{EI/cost}}(\mathcal{S})$ be the value of the "EI / cost" policy, $\text{argmax}_x \, \text{EI}(x)/c(x)$.
  - Then, for any (large) $\alpha > 0$, there exists a BO problem instance (a prior probability distribution over objective and cost functions, a budget, and a set of initial observations $\mathcal{S}$) where:

$$V^*(\mathcal{S}) > \alpha V^{\text{EI/cost}}(\mathcal{S}) \, .$$

- **What does this mean?** Setting α to be large, you can find problem instances where the performance of EI/cost is less than $1/\alpha$ times the optimal performance.
- **Does this happen in practice?** When there exist points with small / near-zero costs, this phenomenon happens often (anecdotally).

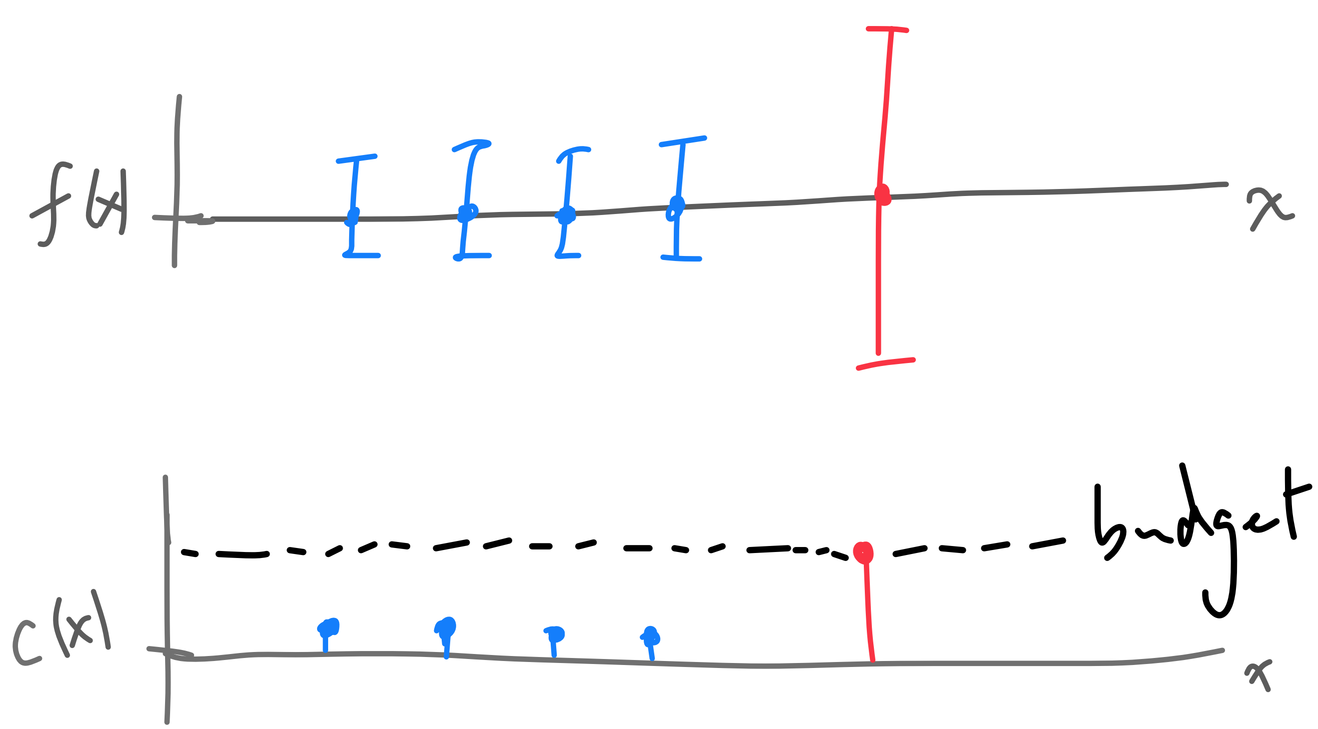
Proof via construction of bad instance

- Suppose we have a function f with a finite domain and the following prior.
- All points are normally distributed and have mean 0.
- Two types of points: many low-variance, low-cost point and one high-variance, high-cost point.
  - low-variance, low-cost: variance is ϵ^2, cost of evaluation $c(x) = \epsilon$
  - high-variance, high-cost: variance is 1, cost of evaluation is $c(x) = 1 + \delta$
- Suppose that the total evaluation budget is also $1 + \delta$.



Proof via construction of bad instance

- Two possible policies:
  - Policy 1: “always measure the low-variance arms”
  - Policy 2: “measure the high-variance arm once”



Proof via construction of bad instance

- Let $Z \sim \mathcal{N}(0,1)$. Acquisition values for the EI / cost policy:
  - low-variance, low-cost: $\mathrm{EI}(x)/c(x) = \mathbf{E}[(f(x) - 0)^+]/\epsilon = \mathbf{E}[\epsilon Z^+]/\epsilon = \mathbf{E}[Z^+]$
  - high-variance, high-cost: $\mathrm{EI}(x)/c(x) = \mathbf{E}[(f(x) - 0)^+]/(1+\delta) = \mathbf{E}[Z^+]/(1+\delta)$
- So, the EI / cost policy prefers the low cost points at the beginning**.**
  - After measuring the first point, the high-cost point is no longer feasible, so EI / cost must be the first type of policy **(“**always selects low-cost points**”)**
  - With some calculations, we can show that: $\lim_{\epsilon \to 0} V^{\mathrm{EI/cost}}(\mathcal{S}) = 0$
- The “measure the high-variance arm once” has value $\mathbf{E}[Z^+] > 0$
  - The value is independent of ϵ

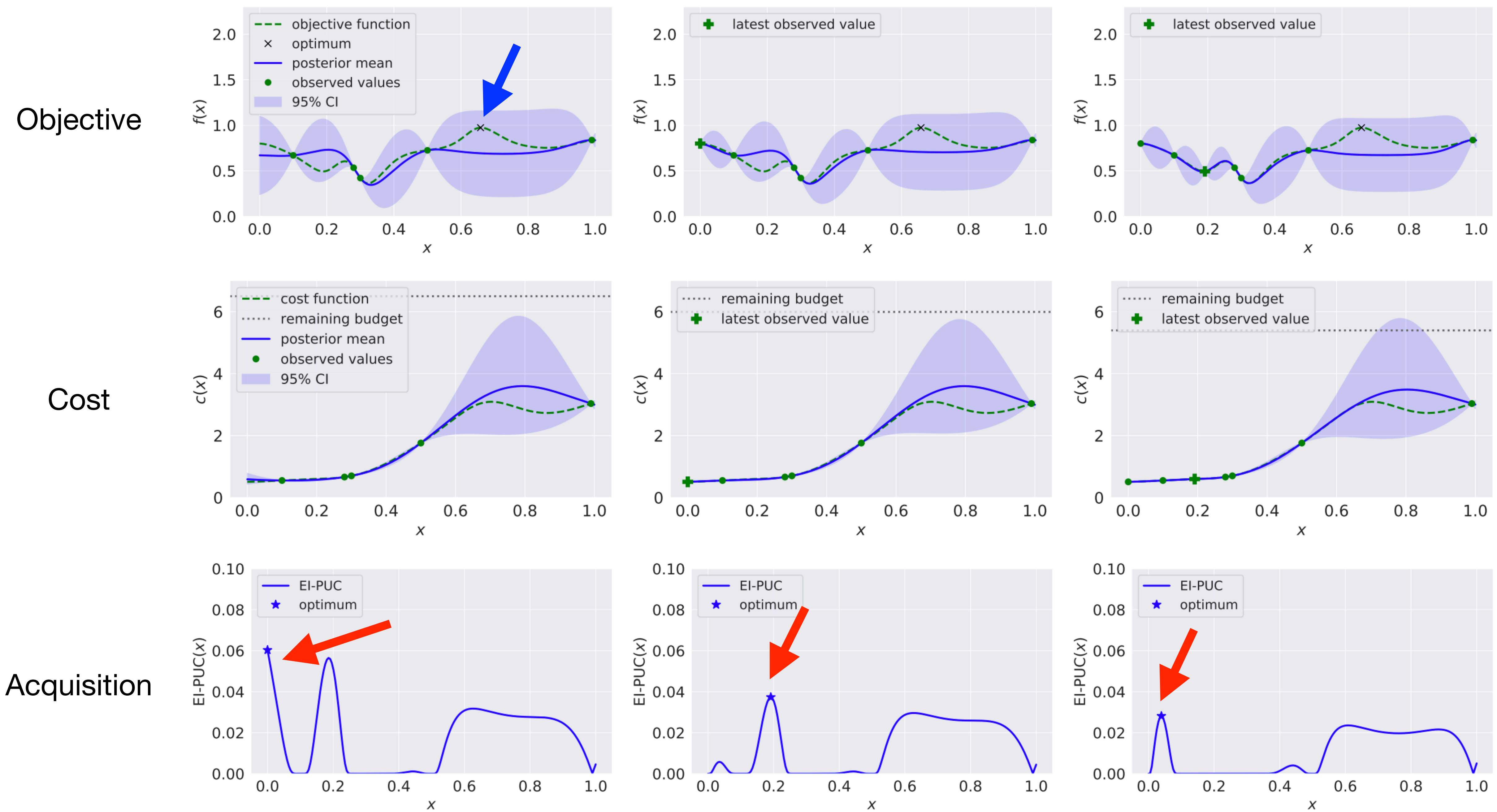
Proof via construction of bad instance

Conclusion:

- Optimal policy: Policy 2 (“measure the high-variance arm once”)
- EI/cost policy: Policy 1 “repeatedly measure the low-variance arms”
- $\lim_{\epsilon \to 0} V^{\mathrm{EI/cost}}(\mathcal{S}) = 0$

Illustrative example (EI / cost)

- In this synthetic setting, EI / cost tends to measure the low cost points.



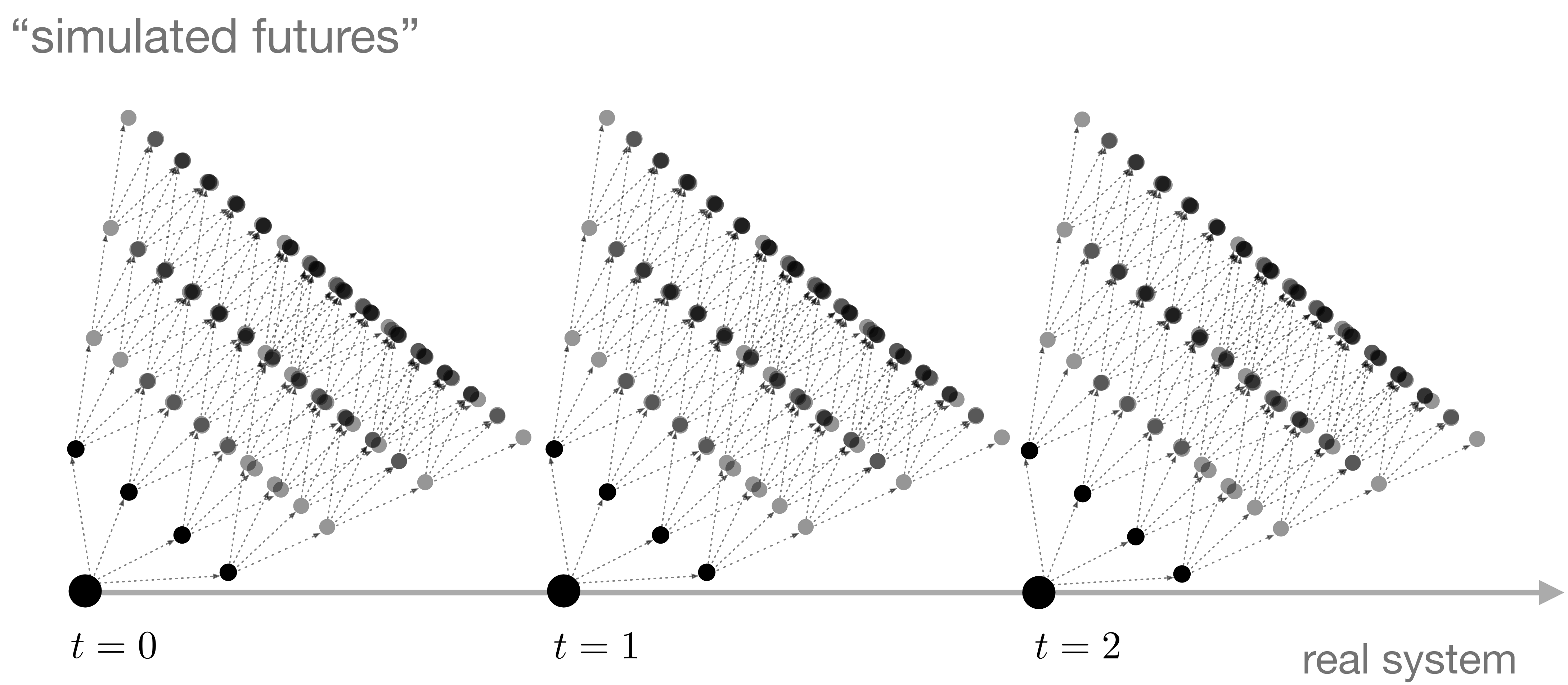
MDP formulation of the problem

- In order to properly manage the various trade-offs, we need to **plan ahead**!
- Dynamic programming equation, considers both objective & cost-learning:
  - Let $\mathcal{S}_n = \{(x_i, y_i = f(x_i), z_i = c(x_i))\}$ denote the **set of data** up to step n
  - The decision at step is the point to measure: $x_n = \pi_n(\mathcal{S}_{n-1})$
  - Objective: $V^*(\mathcal{S}) = \sup_{\pi \in \Pi} \mathbf{E}^{\pi}_{\mathcal{S}} \left[u(\mathcal{S}_{N_B}) - u(\mathcal{S}_0) \right]$, where:
    - $u(\mathcal{S}_n) = \max_{(x,y,z) \in \mathcal{S}_n} y$ (we care about the best observed point)
    - $N_B = \sup \left\{ k : \sum_{j=1}^{i} z_j \leq B \right\}$ (budget is still available)
    - Expectation is over the sequences of random observation sets $\{\mathcal{S}_k\}$

Challenges of the MDP

- State space is **highly intractable**:
  - Continuous
  - Grows with the number of observed points so far
  - High-dimensional, but even worse, the state is a set
    - Ordering of observations don't matter
- Traditional ADP / RL approaches to solve this problem will be challenging because they often require approximating a *function of the state* (whether value / policy)
- Attempts at non-myopic BO have largely focused on
  - Heuristic approximations: Gonzalez et al, 2016
  - Horizon of two: Wu & Frazier, 2019, Zhang et al., 2021
  - Rollout policies: Lam et al., 2016, Lee et al., 2020, Yue & Kontar, 2020, Lee et al, 2021

Our approach: decision trees

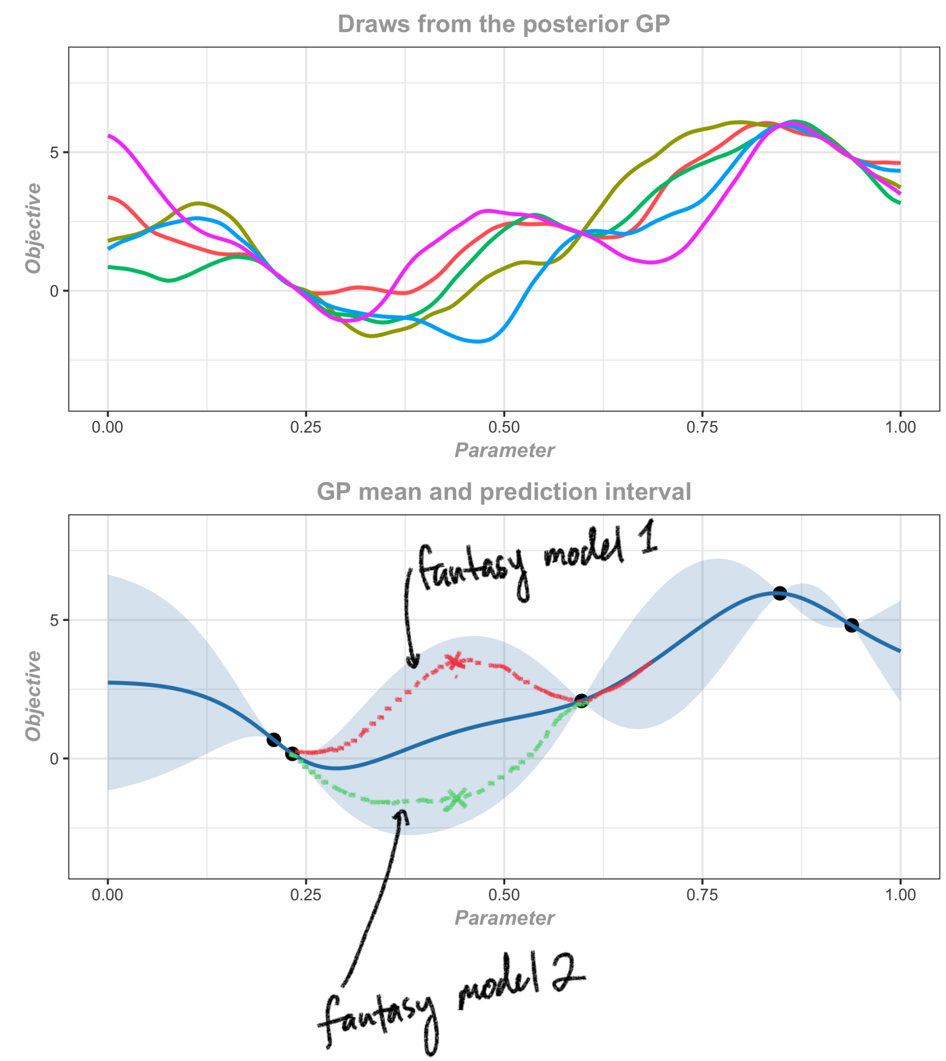


We ***avoid*** the complexities of commonly-used, value-function-based ADP/RL by:

- using a discrete representation of the future
- re-optimizing after each period

Optimizing via differentiable trees

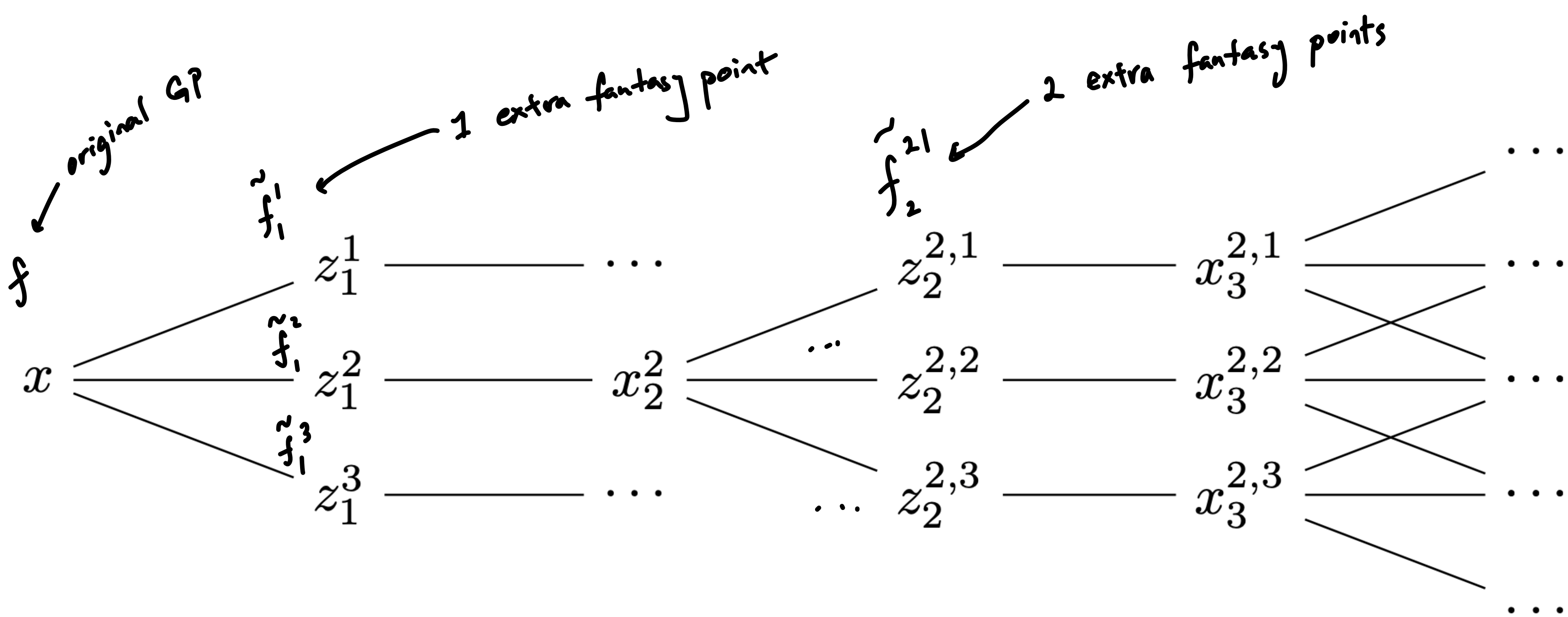
- At each period, we optimize a differentiable decision tree [7]
- Based on the notion of **fantasizing from the GP**:
  - Suppose we want to know the effect on our knowledge of measuring at x
  - Sample a "fantasy observation" $\tilde{y} \sim f(x)$ from GP
  - Add $(x, \tilde{y})$ to GP training data
  - Condition on the full data to get *fantasy GP* $\tilde{f}$
    - Full, optimizable model that comes with uncertainty estimates!
- Suppose we want to consider the value of a policy π
  - Repeatedly fantasize and apply π to each fantasy GP to get an estimated value of the policy



[7] Jiang*, **J.***, Balandat*, Karrer, Gardner, Garnett, 2020

Optimizing via differentiable trees

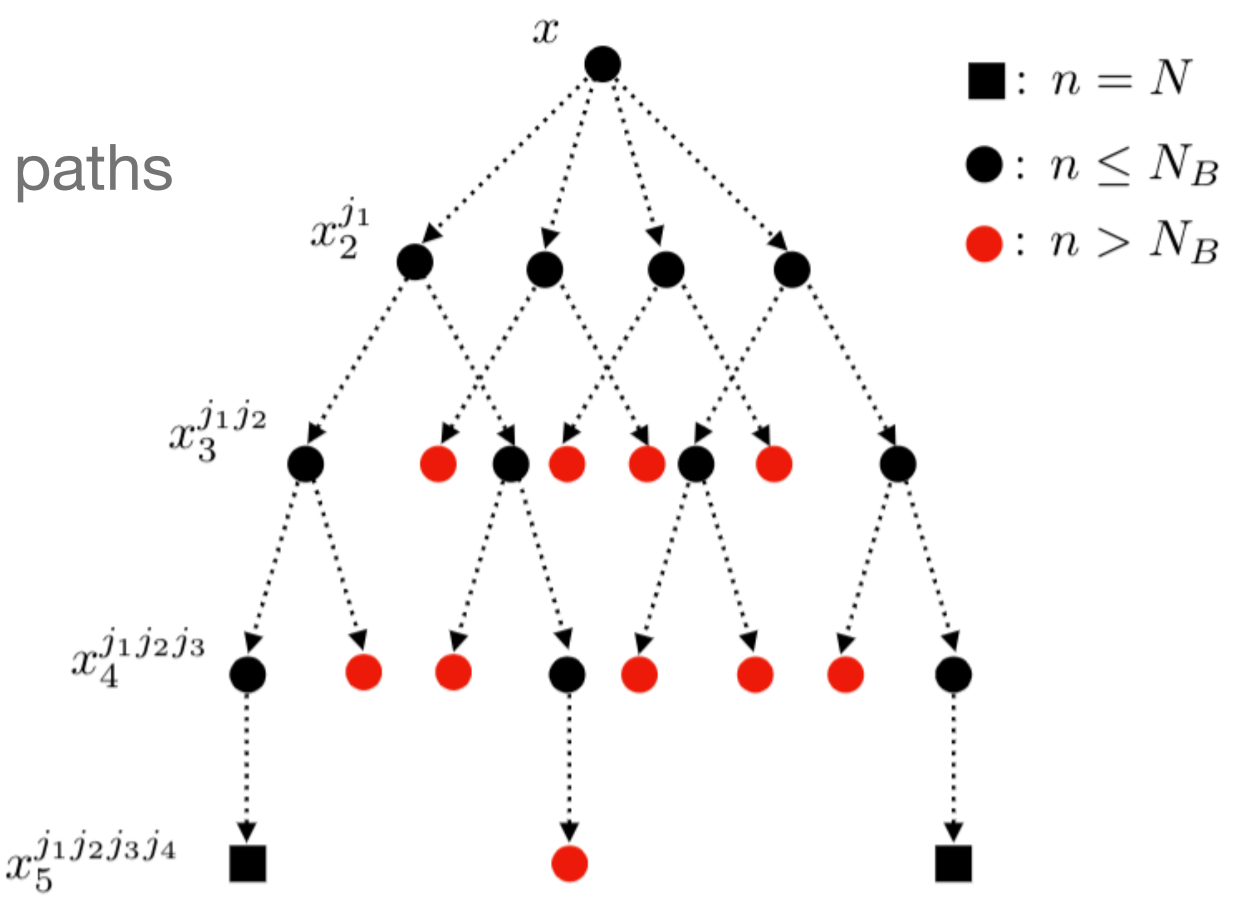
- Using the "reparameterization trick" we can write $\tilde{y} = \mu(x) + L(x)\,z$, where
  - $\mu(x)$ is the posterior mean of the GP,
  - $L(x)L(x)^T = \Sigma(x)$, the posterior covariance of the GP,
  - and $z \sim N(0, I)$ — no dependence on x!
- We can (auto-) differentiate through the tree wrt $\mathbf{x}$ for fixed $\mathbf{z}$, using BoTorch [8] / PyTorch



[7] Jiang*, **J.***, Balandat*, Karrer, Gardner, Garnett, 2020
[8] Balandat, Karrer, **J.**, Daulton, Letham, Wilson, Bakshy, 2020

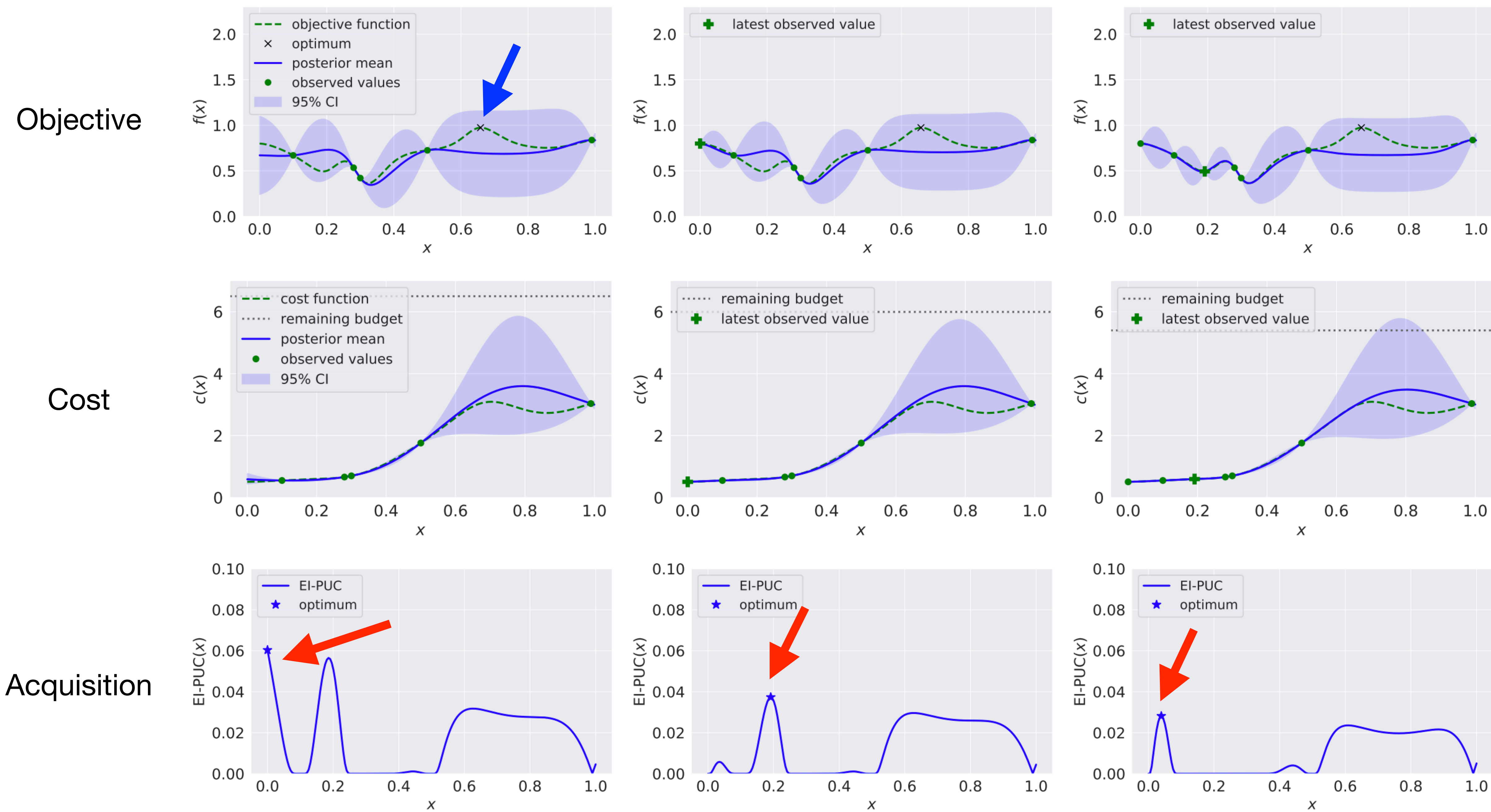
Visualization of one such tree

- Original MDP formulation has a random horizon (ends when no budget left)
- Zero subsequent value along paths in the tree where the budget is exhausted
- Partial rewards generated along those paths



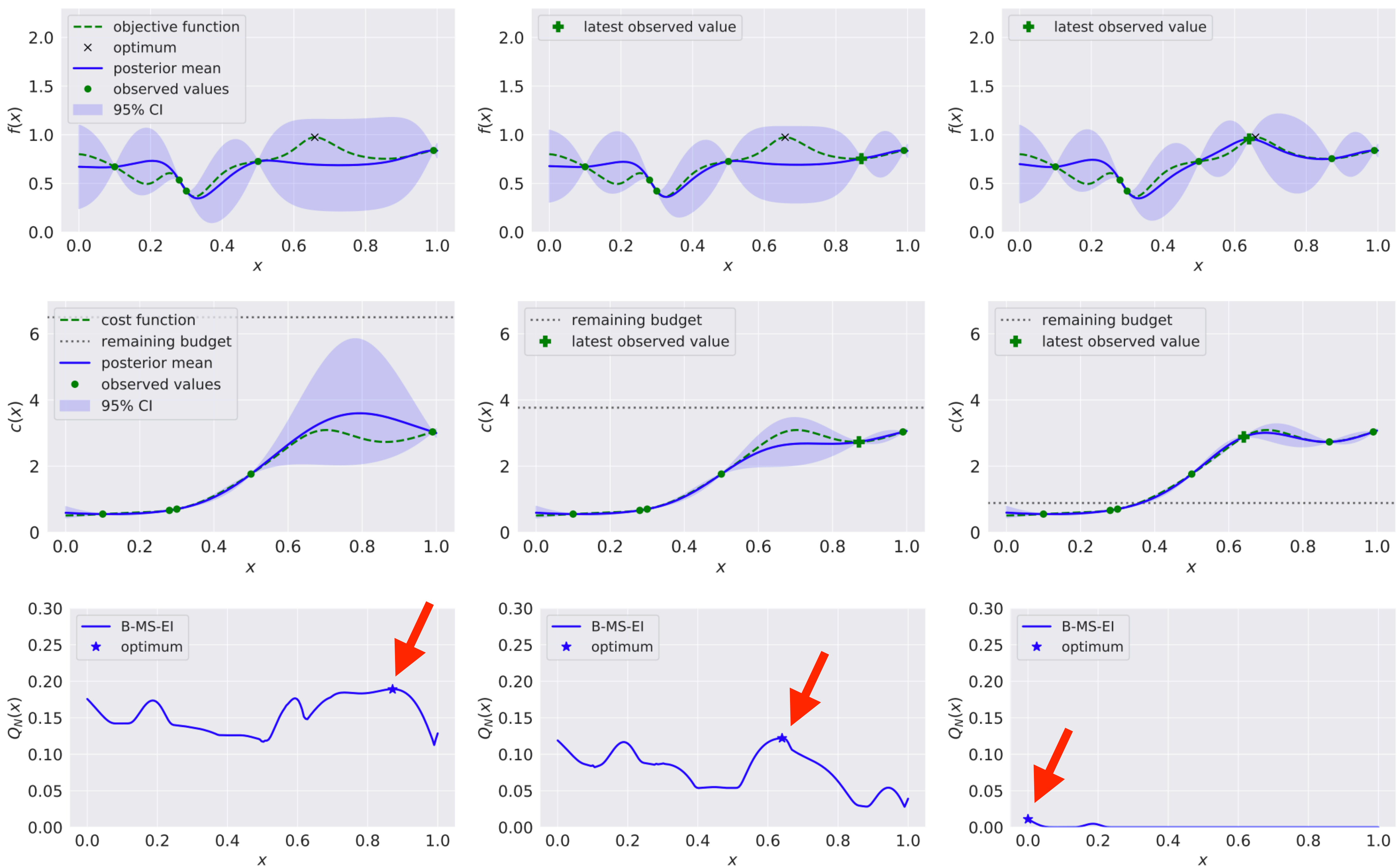
Recall the previous example (EI / cost)

- In this synthetic setting, EI / cost tends to measure the low cost points.

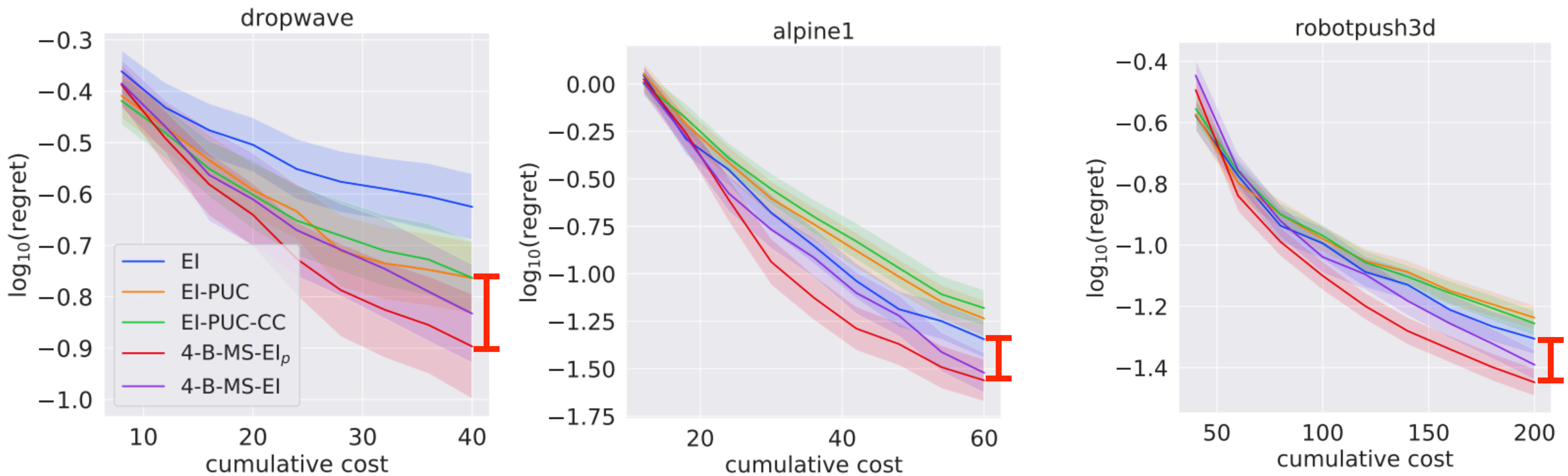


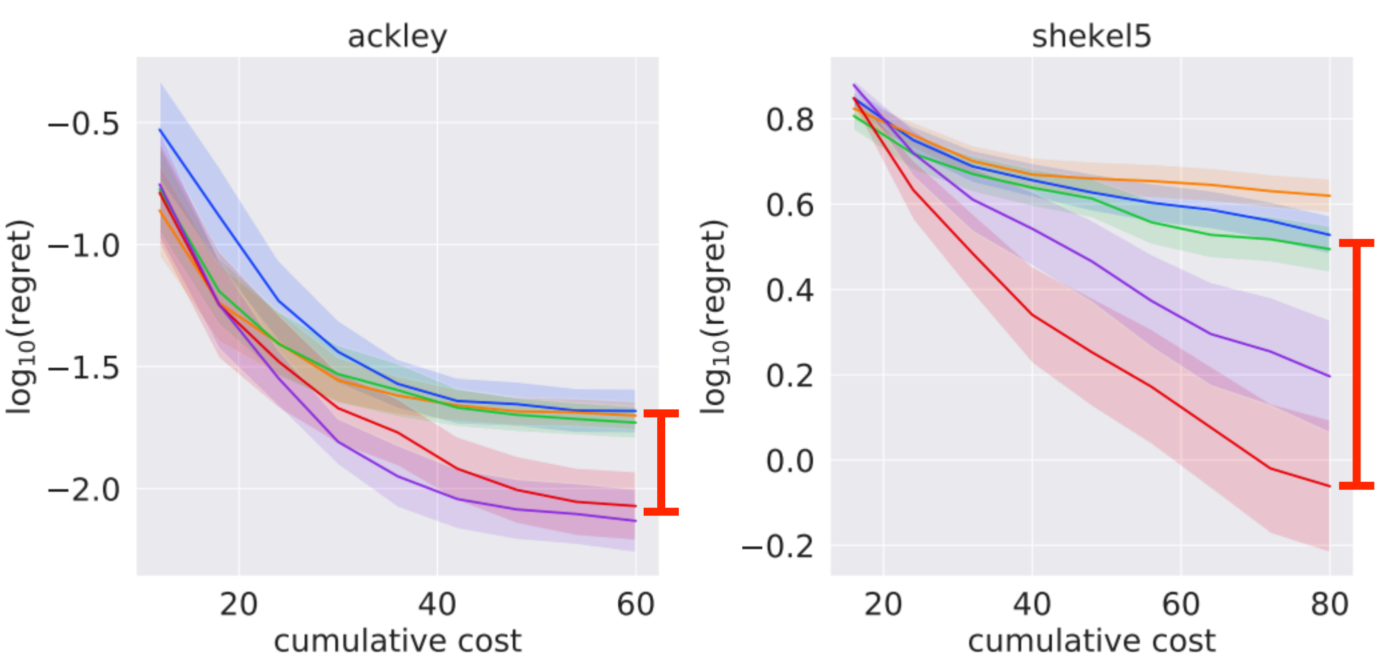
Illustrative example (our new acquisition function)

- But our method will try to learn the cost function and find the optimum.



Empirical results and industry collaborations





- Results from Meta collaborations:
  - 5x reduction in cost to achieve the same objective quality
  - Made available in company-wide AutoML platform

Other computational notes

- Based on open-source code in BoTorch [8]

- Batched (vectorized) computations involving fantasy GP models in PyTorch
  - A 3 layer tree with n_1, n_2, n_3 nodes is represented by a $n_3 \times n_2 \times n_1 \times 1$ GP model
  - Batched linear algebra tensor operations that exploit parallelization and hardware acceleration

[8] Balandat, Karrer, **J.**, Daulton, Letham, Wilson, Bakshy, 2020

Drawbacks of the approach and future work

- Could we allow for cheaper proxies to also be used?
  - Future work: extension to the multi-fidelity setting
- Decision trees do not scale with horizon
  - Our methodology only allows us to look ahead 4-6 steps (still a major improvement upon existing non-myopic BO methods)
  - Acquisition optimization times range from 42sec - 7min, compared to less than 30sec for baseline acquisitions.
  - Future work: improved multi-step methodologies?
- Heuristic budget pacing rules to solve shorter-horizon problems
  - Future work: long / unknown horizons?
- Requires at each step re-optimization due to discretized decision tree
  - Future work: policy reuse?

Thank you! Questions?

Please feel free to email me (danielrjiang@gmail.com) for additional comments / discussion!