User Priors for Hyperparameter Optimization

Luigi Nardi

Lund/Stanford/DBtune

October 12, 2022

LUND UNIVERSITY

dbtune

Speaker's Bio and Contact Info

- Current:
  - Assistant Professor at Lund University
  - Researcher at Stanford University
  - Founder & CEO at DBtune
- Previously: Imperial College London, UPMC (Paris), LAAS (Toulouse), Universidad Autónoma (Madrid), La Sapienza (Rome)
- Research: Bayesian optimization, AutoML
- Applications: HW design, compilers, CV, robotics, DBs
- Offices: E-huset (LU campus), Gates (Stanford campus)

Contact:

Luigi Nardi

luigi.nardi@cs.lth.se

cs.lth.se/luigi-nardi

GitHub: @luinardi

Twitter: @luiginardi

Research Team

Luigi Nardi
Assistant Professor Lund University
Researcher Stanford University

Artur Luis
Ph.D. Student

Erik Hellsten
Postdoc

Kenan Sehic
Postdoc

Simon Kristoffersson Lind
Ph.D. Student

Carl Hvarfner
Ph.D. Student

Leonard Papenmeier
Ph.D. Student

Collaborators

Leonardo Barbosa
Professor
UFMG

Marius Lindauer
Professor
Leibniz University Hannover

Kunle Olukotun
Professor
Stanford University

Frank Hutter
Professor
University of Freiburg

Matthias Mayr
Ph.D. student
Lund University

Volker Krueger
Professor
Lund University

Outline

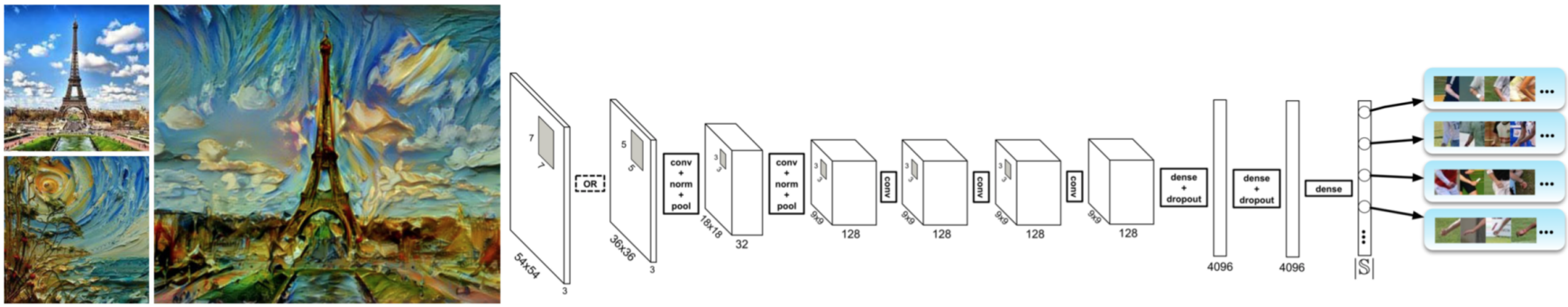
Outline

Machine Learning Meta-challenges

- ML models are getting more and more complex
  - Many parameters (e.g. deep neural networks)

Automate the selection of critical hyper-parameters (see also: Automated Machine Learning (AutoML) [Hutter et al., 2018])

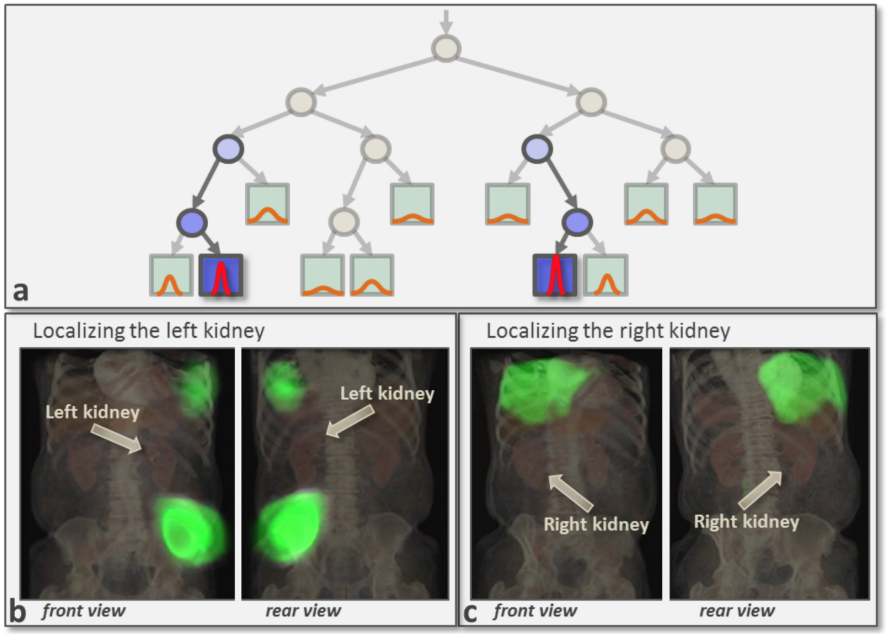
Example 1: Deep Neural Networks



Great interest in large neural networks

- When well-tuned, very successful for visual object identification, speech recognition, computational biology, etc.
- Big investments by Google, Facebook, Microsoft, etc.
- Hyper-parameters: Number of layers, weight regularization, layer size, which nonlinearity, batch size, learning rate schedule, momentum

Example 2: Detection and Localization in CT



Objective: Efficient anatomy detection and localization in computed tomography scans

- Regression Forests [Criminisi et al., 2013]
- Hyper-parameters: Number of trees, max depth, max features, class weights, criterion

Search for Good Hyper-parameters

- Define an objective function
  - Usually, we care about generalization performance
  - Cross validation to measure parameter quality

Search for Good Hyper-parameters

- Define an objective function
  - Usually, we care about generalization performance
  - Cross validation to measure parameter quality
- Standard search procedures:
  - Grid search
  - Random search – Very simple, works surprisingly well
  - Dark magic – A ninja developer using their intuition

Search for Good Hyper-parameters

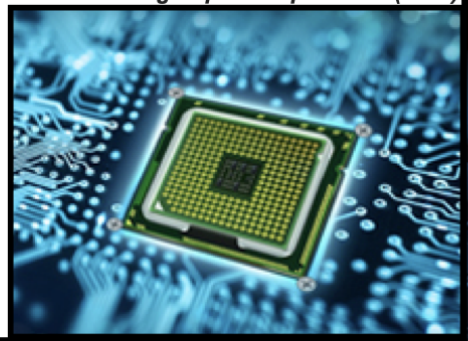
- Define an objective function
  - Usually, we care about generalization performance
  - Cross validation to measure parameter quality
- Standard search procedures:
  - Grid search
  - Random search – Very simple, works surprisingly well
  - Dark magic – A ninja developer using their intuition
- Painful:
  - Training may be very expensive (e.g., time or money)
  - Many training cycles
  - Possibly noisy

Search for Good Hyper-parameters

- Define an objective function
  - Usually, we care about generalization performance
  - Cross validation to measure parameter quality
- Standard search procedures:
  - Grid search
  - Random search – Very simple, works surprisingly well
  - Dark magic – A ninja developer using their intuition
- Painful:
  - Training may be very expensive (e.g., time or money)
  - Many training cycles
  - Possibly noisy
- Alternative approach: Bayesian optimization

Bayesian Optimization Applications

Nardi et al.
Practical Design Space Exploration (2019)



Source: https://techxplore.com/news/2020-02-machine-method-supercharge-battery-electric.html

Attia et al.
Closed-loop Optimization of Fast-charging Protocols for Batteries with Machine Learning, Nature (2020)

Chen et al.
Bayesian Optimization in AlphaGo (2018)

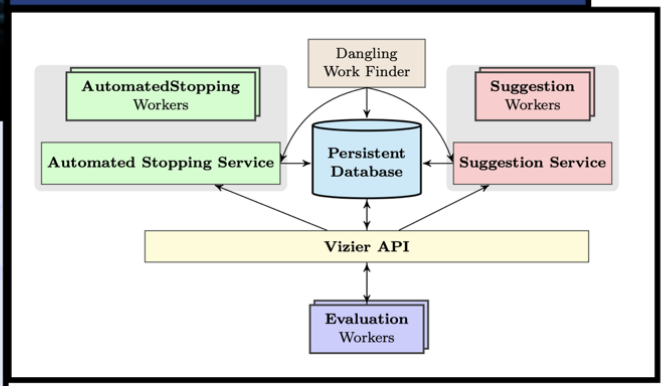
Source: https://becominghuman.ai/summary-of-the-alphago-paper-b55ce24d8a7c

Source: https://www.akira.ai/glossary/automated-machine-

Hutter et al.
Automated Machine Learning (2019)

Balandat et al.
BoTorch: Programmable Bayesian Optimization in PyTorch (2019)

Source: https://botorch.org/

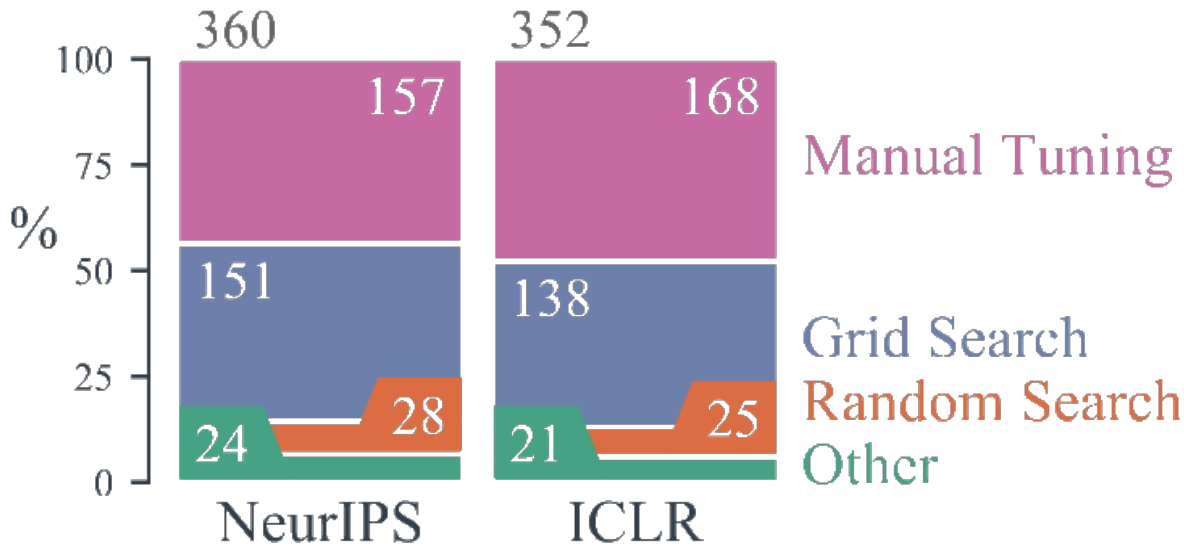


Golovin et al.
Google Vizier: A Service for Black-box Optimization (2017)

Motivation

BO is an effective tool for HPO, but:

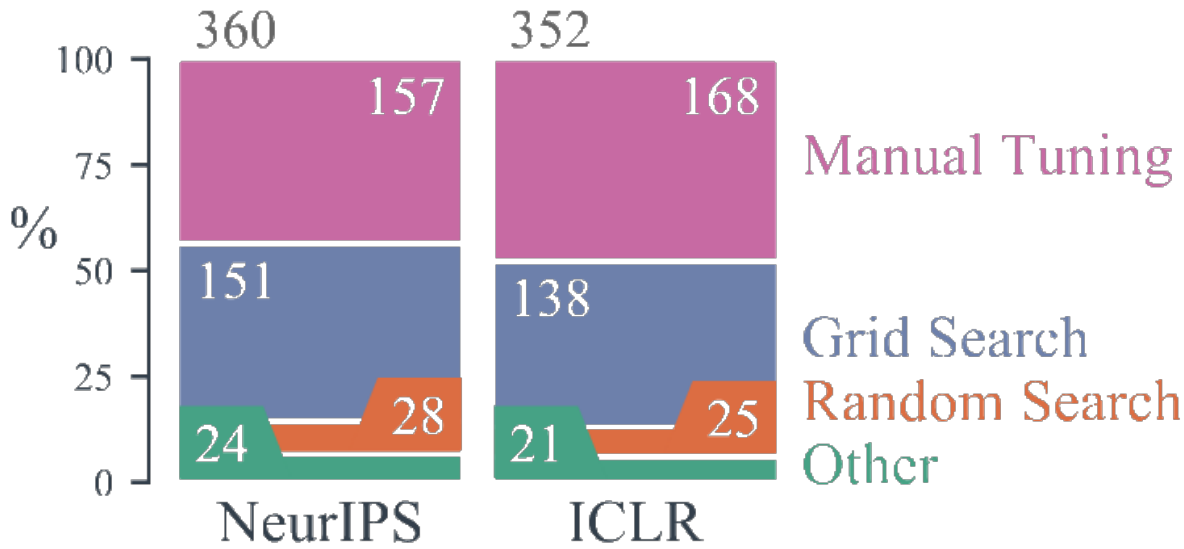
- Low adoption among practitioners [Bouthillier and Varoquaux, 2020]



Motivation

BO is an effective tool for HPO, but:

- Low adoption among practitioners [Bouthillier and Varoquaux, 2020]

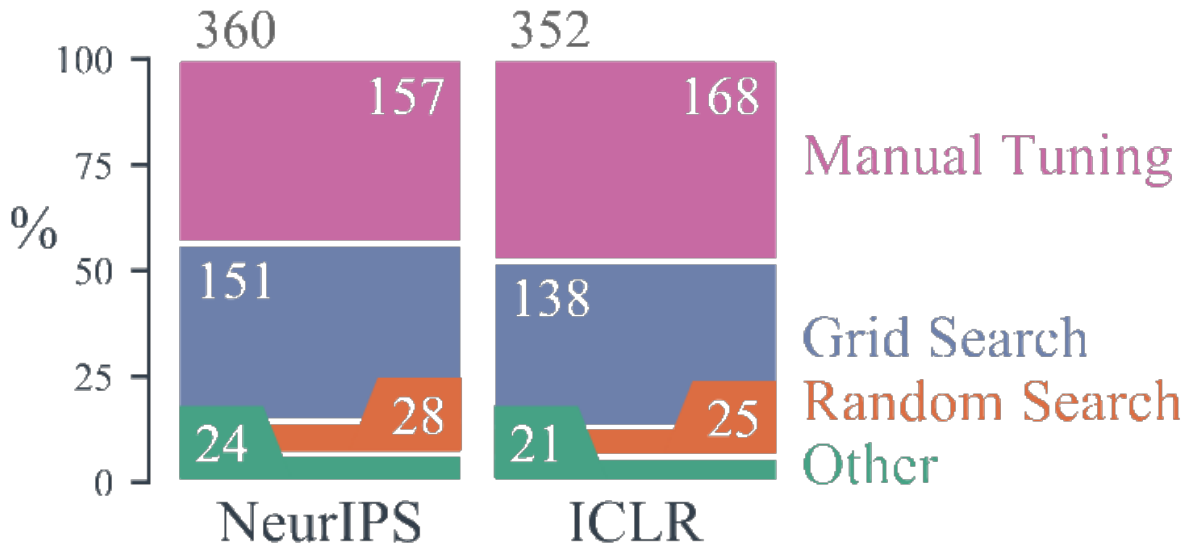


- Can waste evaluations on poor designs

Motivation

BO is an effective tool for HPO, but:

- Low adoption among practitioners [Bouthillier and Varoquaux, 2020]

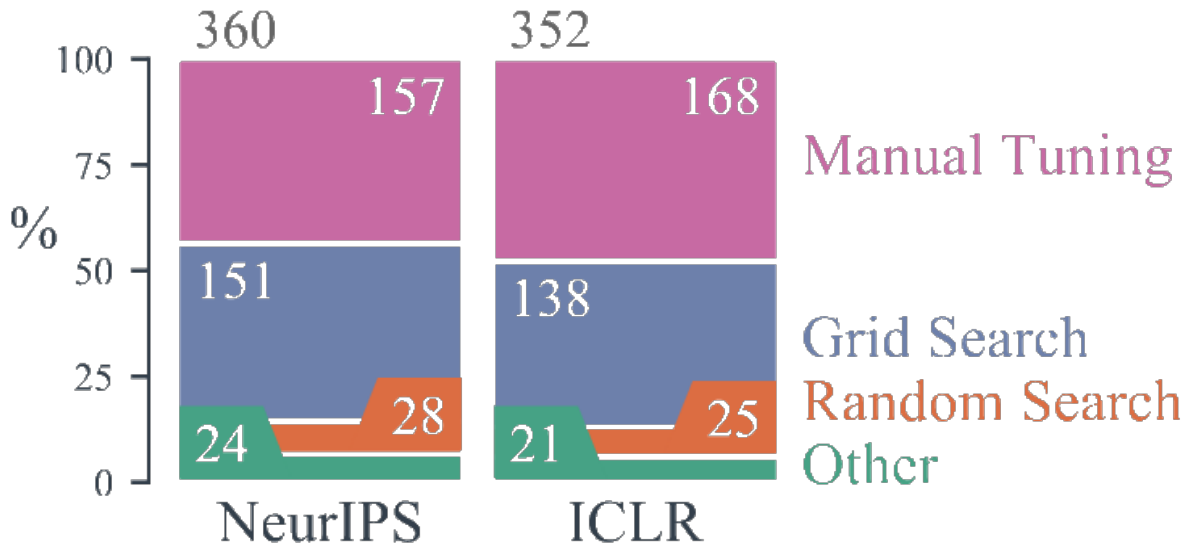


- Can waste evaluations on poor designs
- Disregards user beliefs on the location of the optimum

Motivation

BO is an effective tool for HPO, but:

- Low adoption among practitioners [Bouthillier and Varoquaux, 2020]

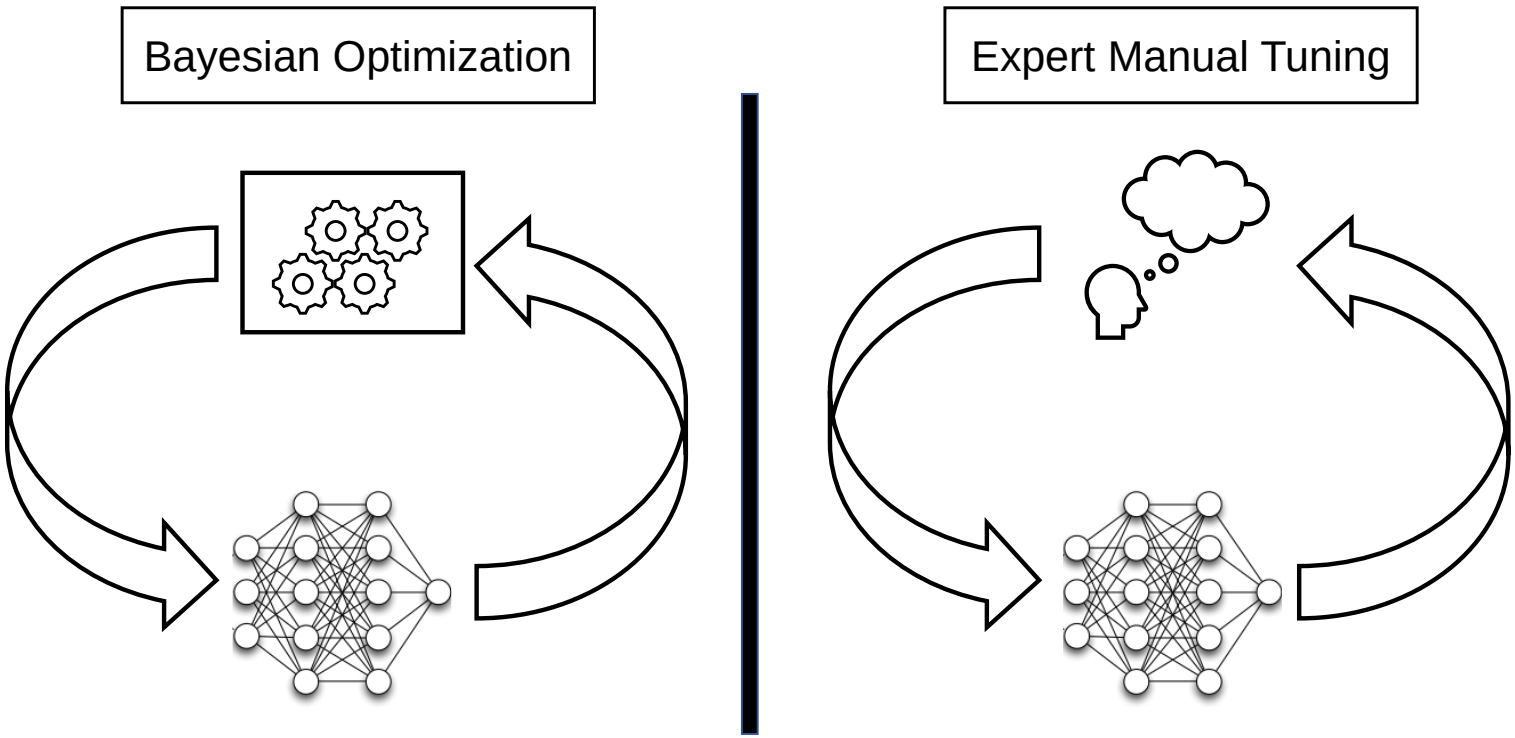


- Can waste evaluations on poor designs
- Disregards user beliefs on the location of the optimum

Can we incorporate expert knowledge in Bayesian optimization?

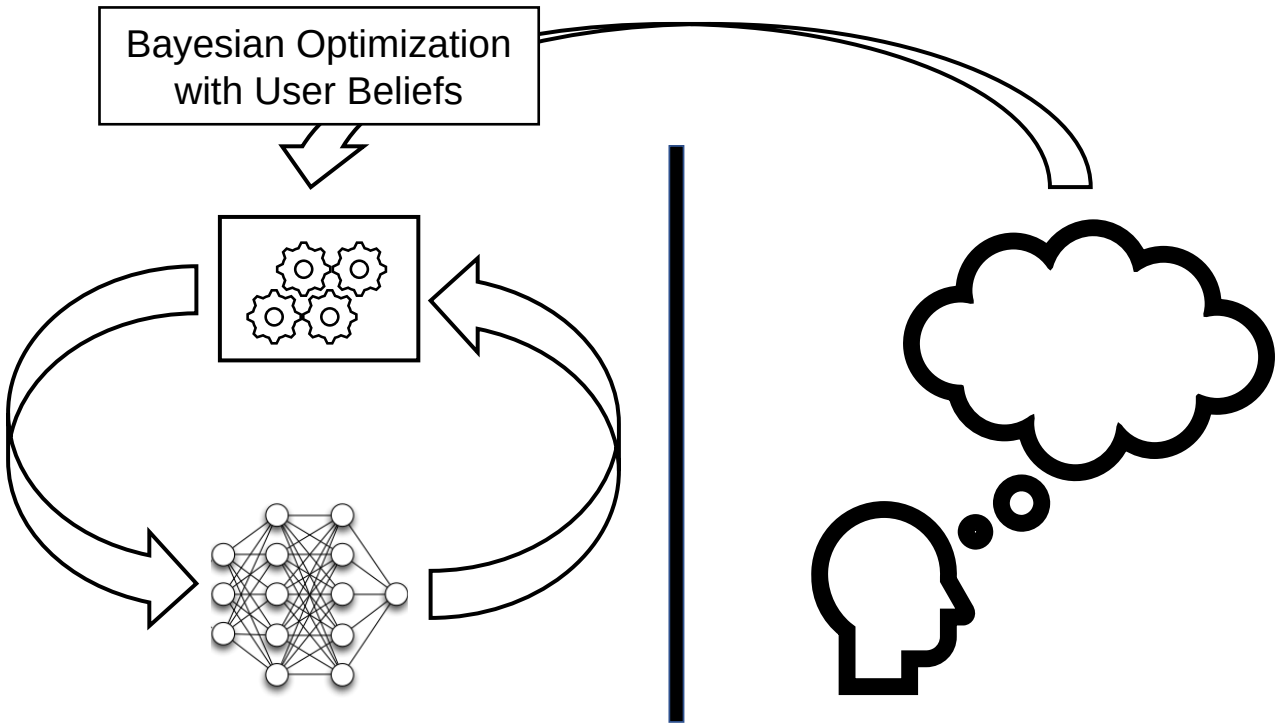
Motivation (Contd.)

ML practitioners typically choose between these approaches



Idea

Accelerate Bayesian optimization by allowing users to integrate beliefs that are typically reserved to manual search



Example belief: Adam's best learning rate is likely around 10^{-3}

Outline

Bayesian Optimization Setting

Setting

Globally optimize an objective function that is expensive to evaluate (e.g. cross-validation error for a massive DNN)

- Build a probabilistic surrogate model for the objective using outcomes of past experiments as training data

Bayesian Optimization Setting

Setting

Globally optimize an objective function that is expensive to evaluate (e.g. cross-validation error for a massive DNN)

- Build a probabilistic surrogate model for the objective using outcomes of past experiments as training data
- The model is cheaper to evaluate than the original objective

Bayesian Optimization Setting

Setting

Globally optimize an objective function that is expensive to evaluate (e.g. cross-validation error for a massive DNN)

- Build a probabilistic surrogate model for the objective using outcomes of past experiments as training data
- The model is cheaper to evaluate than the original objective
- Optimize cheap surrogate model to determine where to evaluate the true objective next

Bayesian Optimization Setting

Setting

Globally optimize an objective function that is expensive to evaluate (e.g. cross-validation error for a massive DNN)

- Build a probabilistic surrogate model for the objective using outcomes of past experiments as training data
- The model is cheaper to evaluate than the original objective
- Optimize cheap surrogate model to determine where to evaluate the true objective next
- Common surrogate: Gaussian Processes
  - But also Random Forests, Tree Parzen Estimators, Bayesian Neural Networks

Bayesian Optimization Setting (Contd.)

- Objective: Find global minimum of objective function g:

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x})$$

Bayesian Optimization Setting (Contd.)

- Objective: Find global minimum of objective function g:

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x})$$

- We can evaluate the objective g pointwise

Bayesian Optimization Setting (Contd.)

- Objective: Find global minimum of objective function g:

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x})$$

- We can evaluate the objective g pointwise
- But we do not have an easy functional form or gradients

Bayesian Optimization Setting (Contd.)

- Objective: Find global minimum of objective function g:

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x})$$

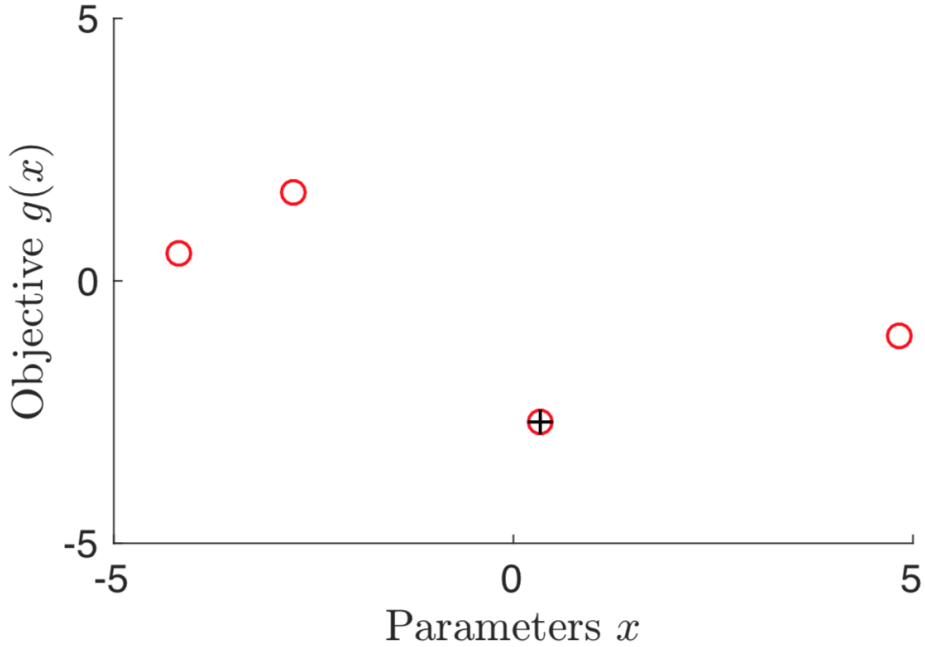
- We can evaluate the objective g pointwise
- But we do not have an easy functional form or gradients
- Observations may be noisy

Bayesian Optimization Setting (Contd.)

- Objective: Find global minimum of objective function g:

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x})$$

- We can evaluate the objective g pointwise
- But we do not have an easy functional form or gradients
- Observations may be noisy
- Evaluating g is costly (e.g. train a deep network)



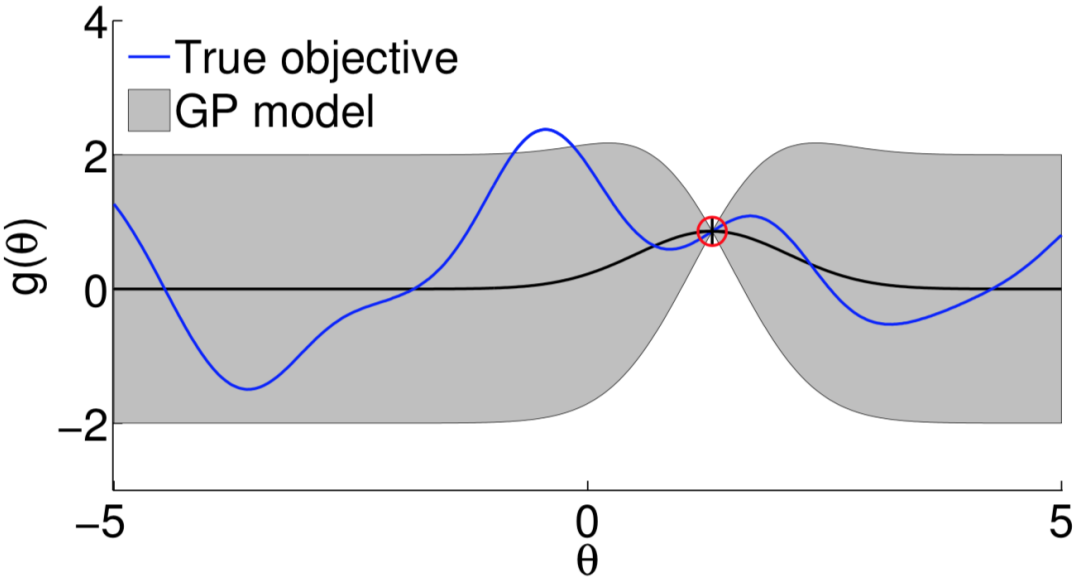
Bayesian Optimization Key Steps

- Avoid evaluating g an excessive number of times
  $\Longrightarrow$ approximate it using a surrogate model $\tilde{g}$
- $\tilde{g}$ is cheap to evaluate, e.g., Gaussian Process
- Find a global optimum $\tilde{g}(\boldsymbol{x}^*)$ of surrogate $\tilde{g}$
- Evaluate true objective g at $\boldsymbol{x}^*$

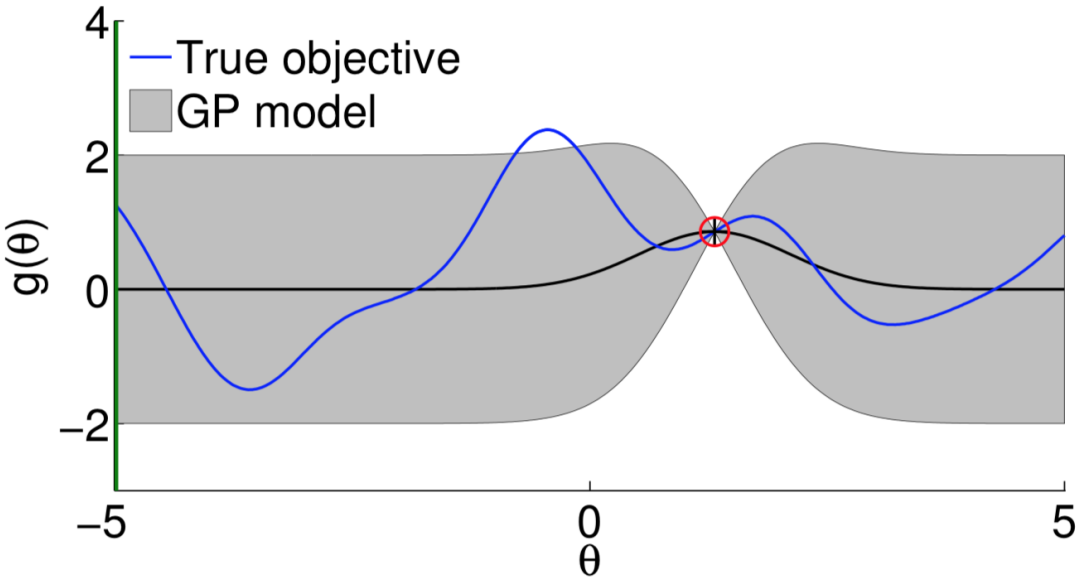
Bayesian Optimization Key Steps

- Avoid evaluating g an excessive number of times
  $\Longrightarrow$ approximate it using a surrogate model $\tilde{g}$
- $\tilde{g}$ is cheap to evaluate, e.g., Gaussian Process
- Find a global optimum $\tilde{g}(\boldsymbol{x}^*)$ of surrogate $\tilde{g}$
- Evaluate true objective g at $\boldsymbol{x}^*$
- Works well if $\tilde{g} \approx g$
- Usually not the case
  $\Rightarrow$ Repeat this cycle and keep updating $\tilde{g}$

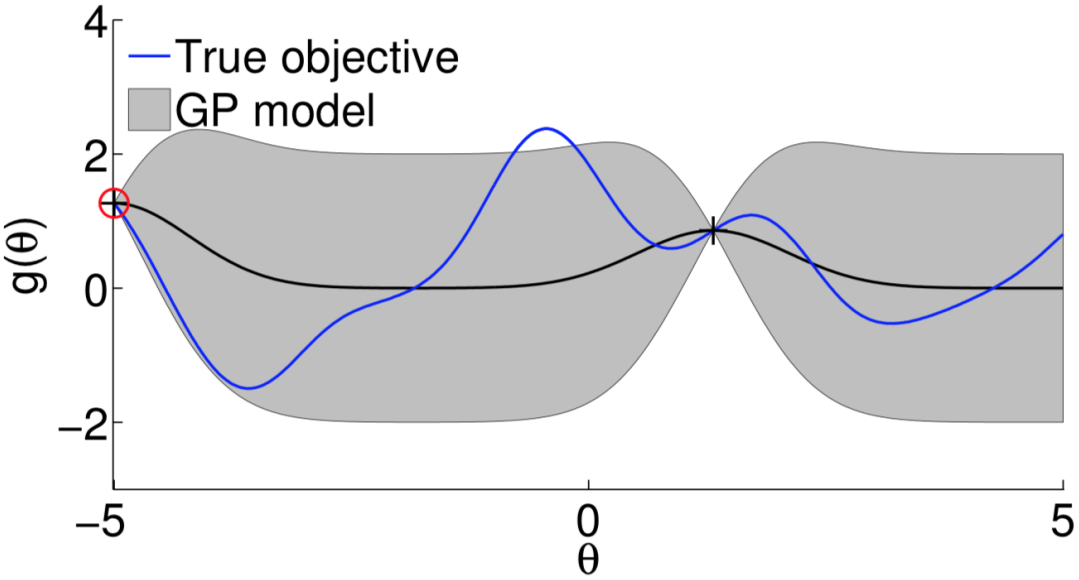
Bayesian Optimization: Illustration



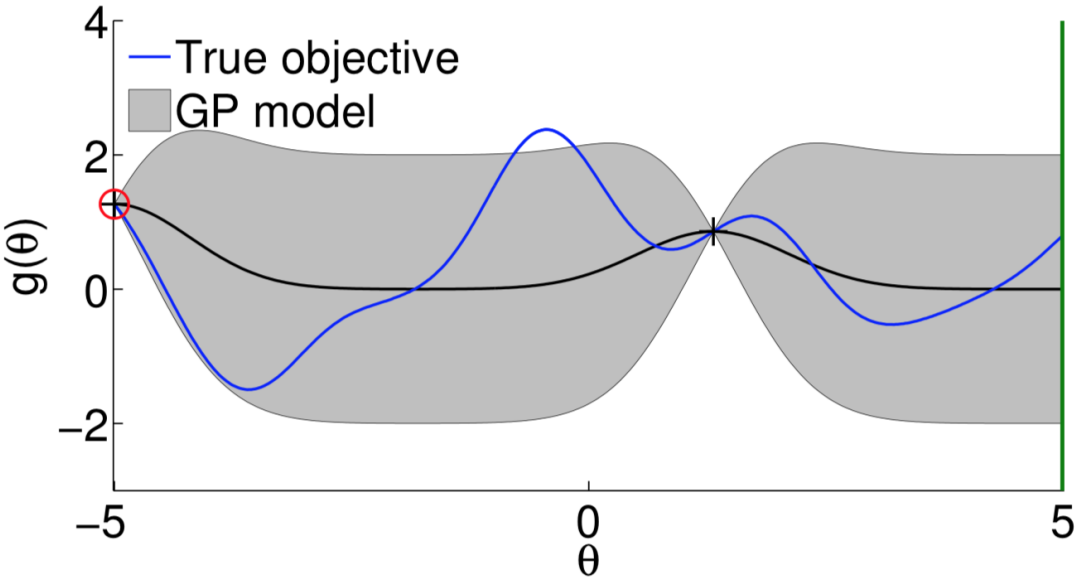
Bayesian Optimization: Illustration



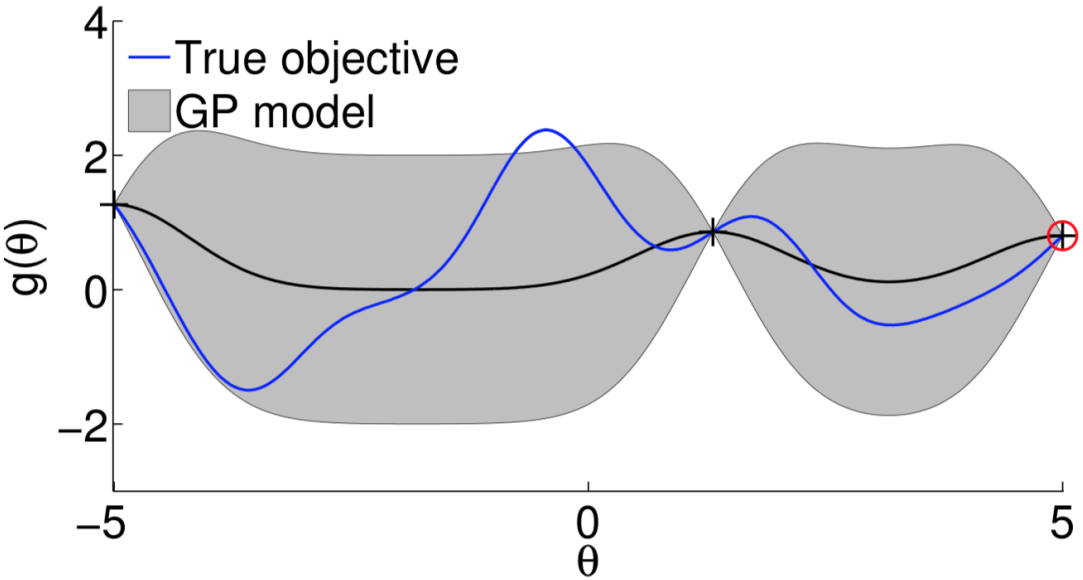
Bayesian Optimization: Illustration



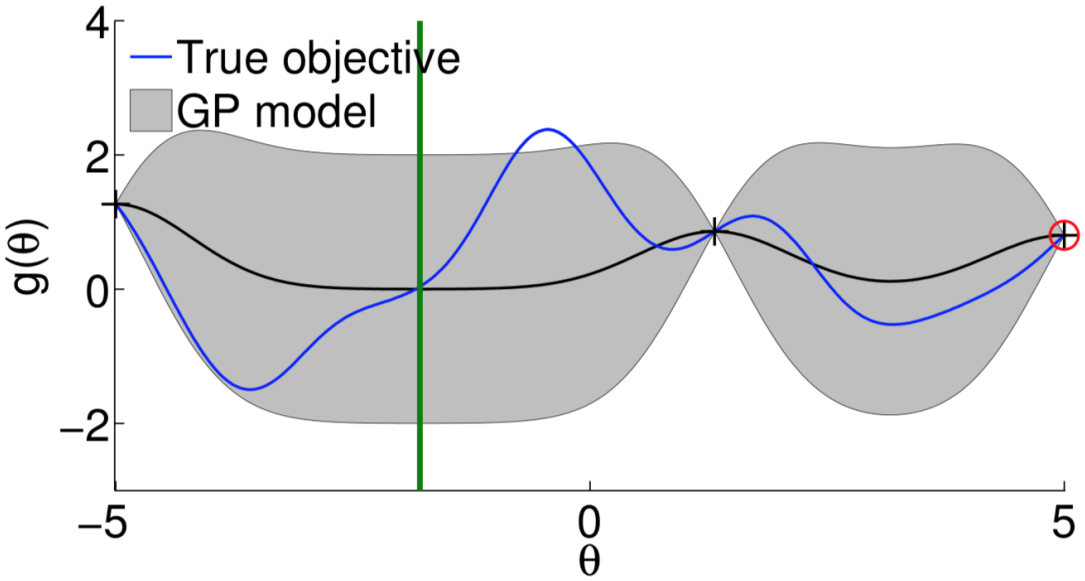
Bayesian Optimization: Illustration



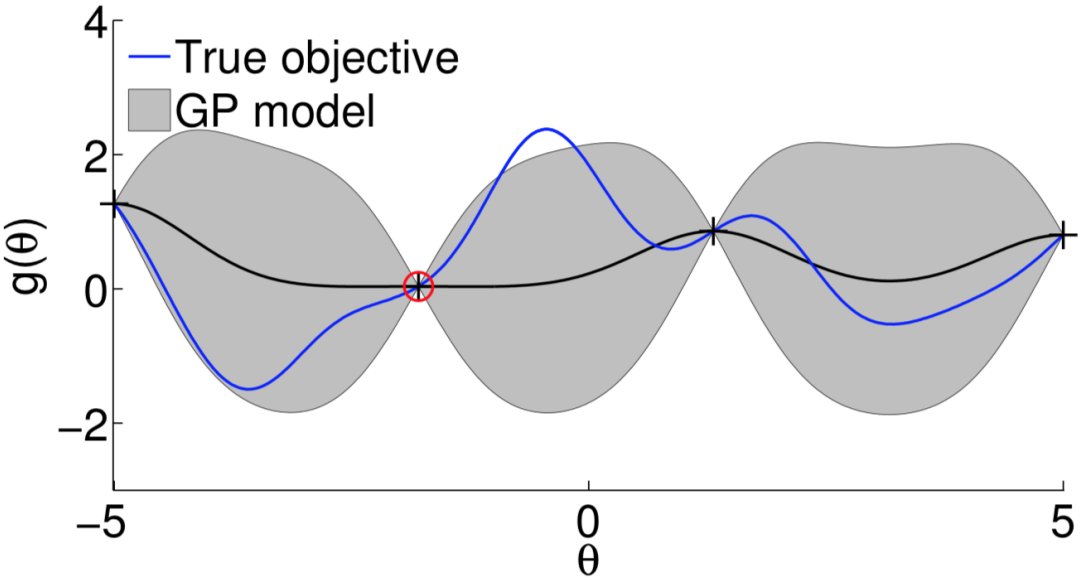
Bayesian Optimization: Illustration



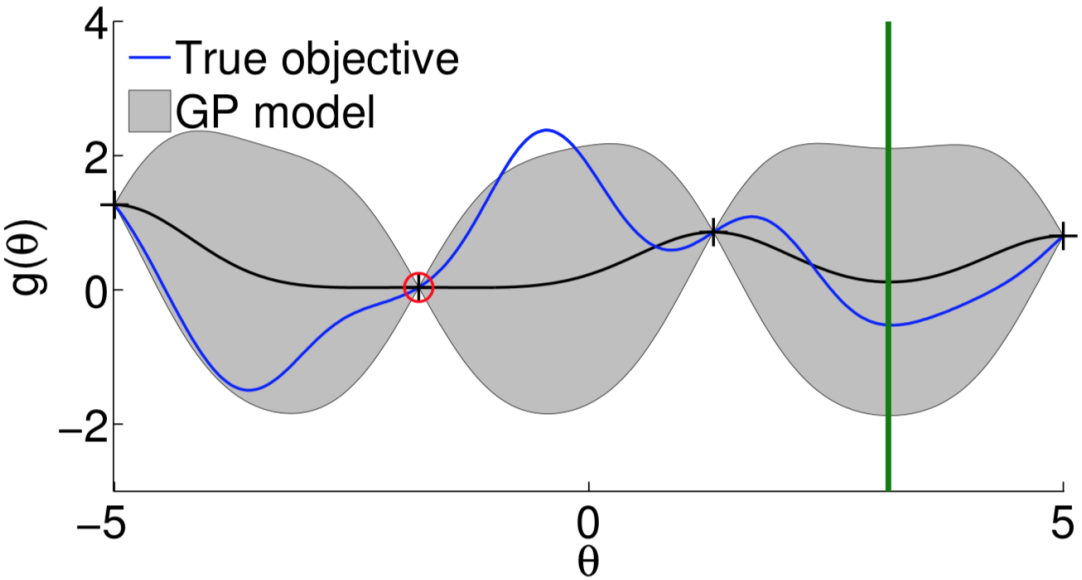
Bayesian Optimization: Illustration



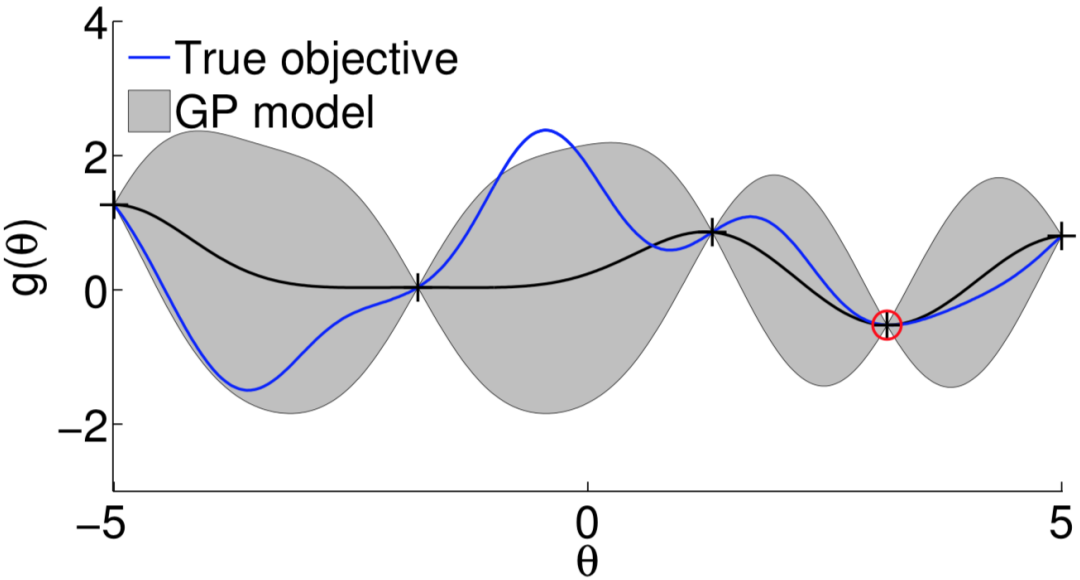
Bayesian Optimization: Illustration



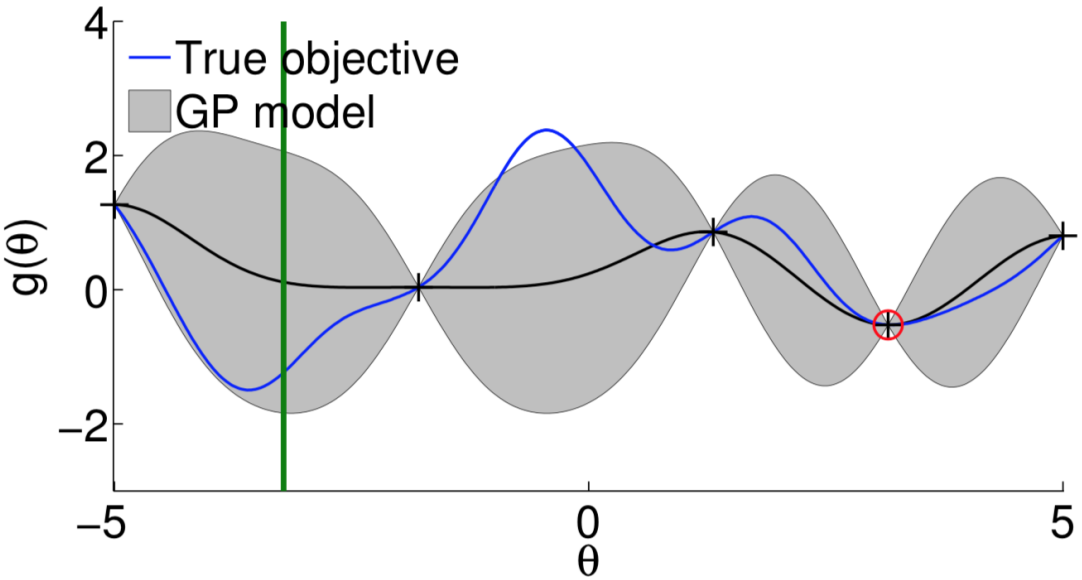
Bayesian Optimization: Illustration



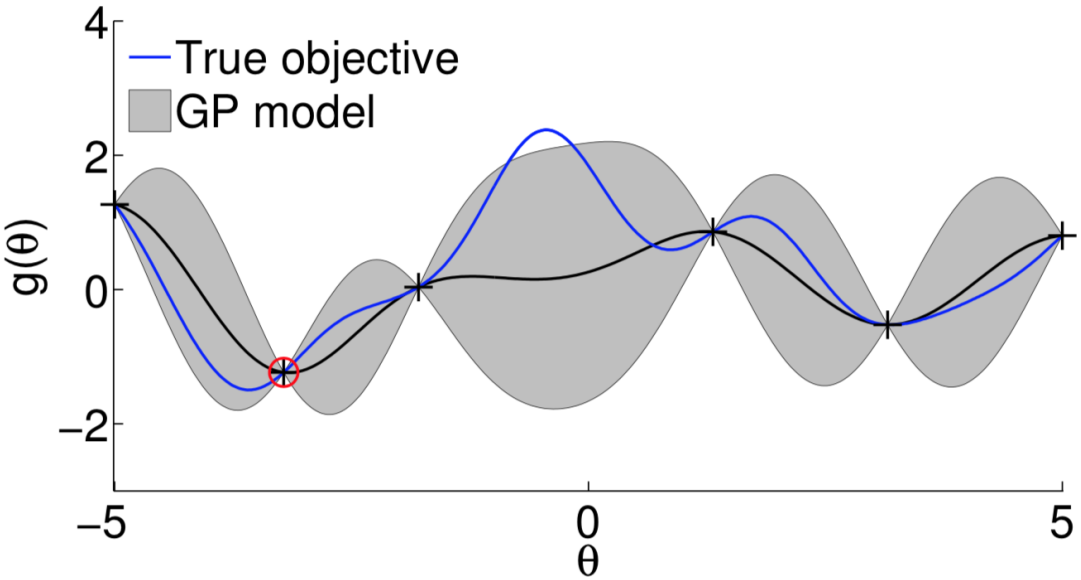
Bayesian Optimization: Illustration



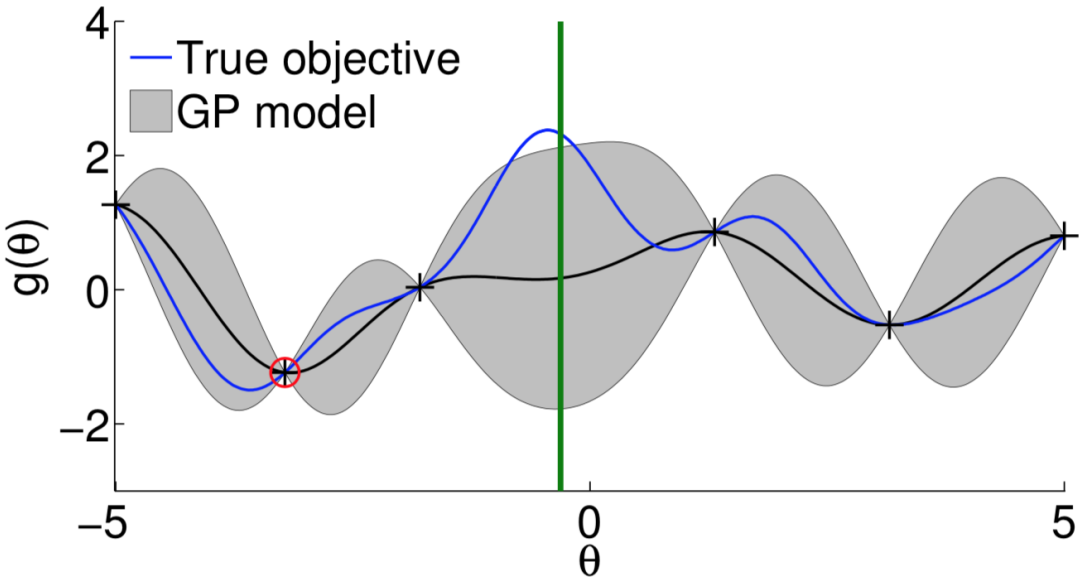
Bayesian Optimization: Illustration



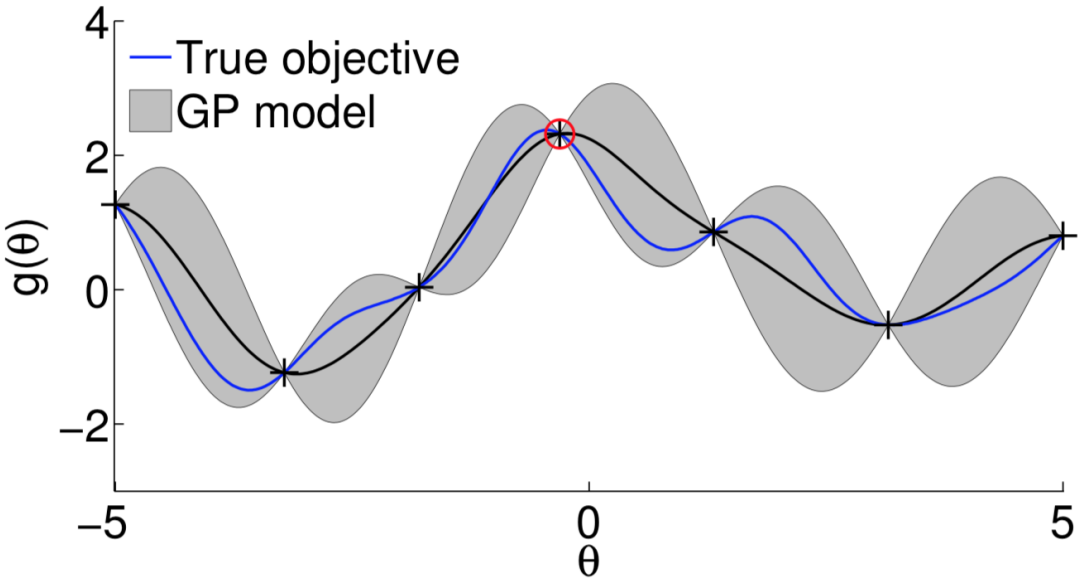
Bayesian Optimization: Illustration



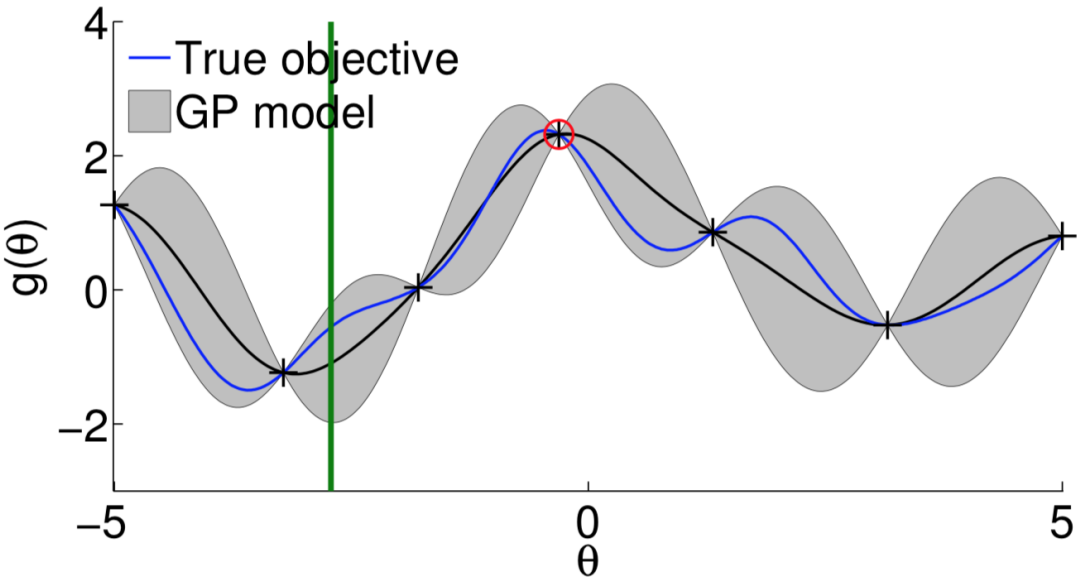
Bayesian Optimization: Illustration



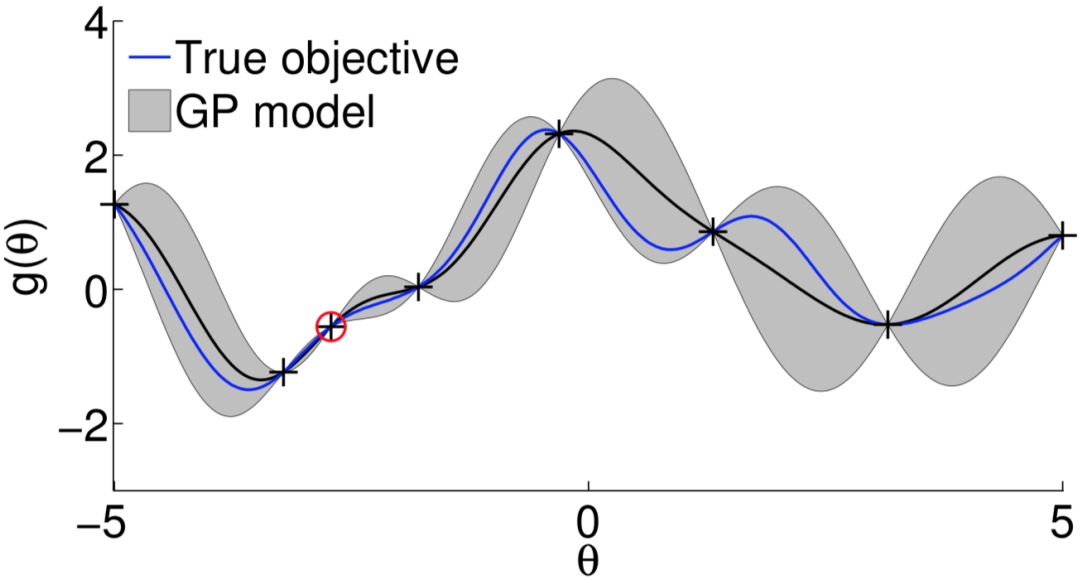
Bayesian Optimization: Illustration



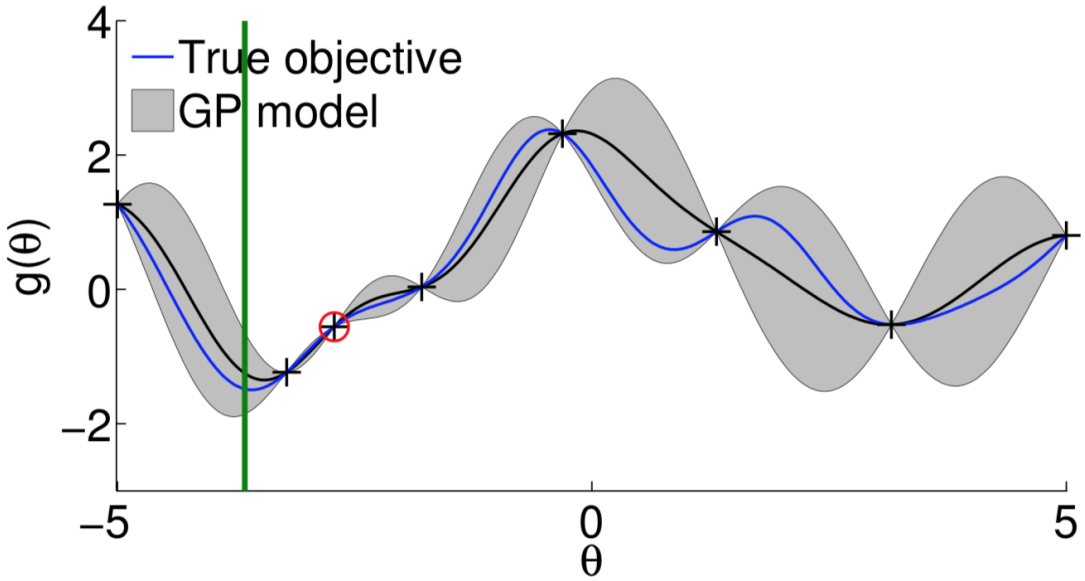
Bayesian Optimization: Illustration



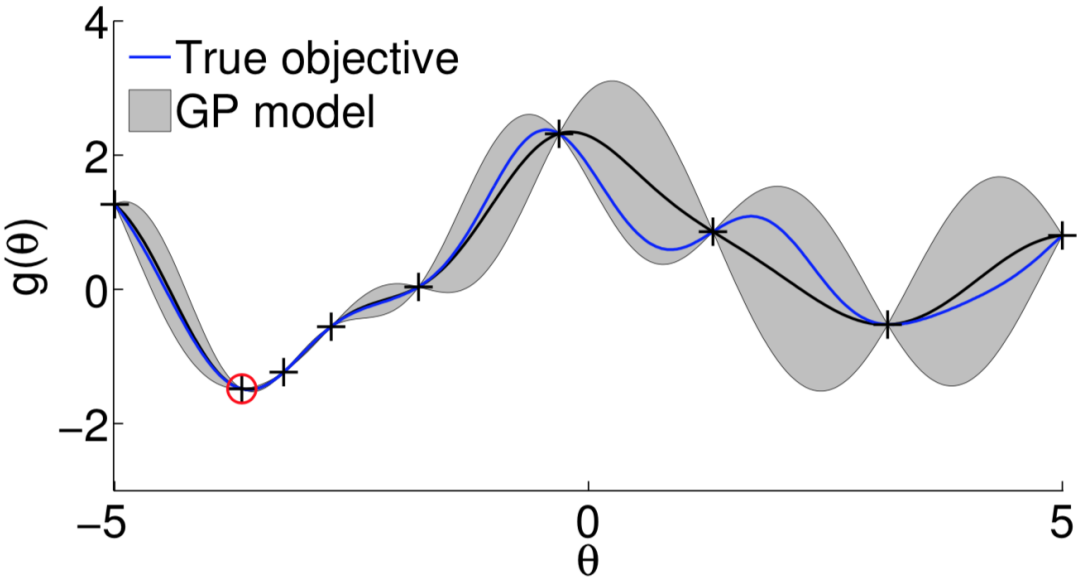
Bayesian Optimization: Illustration



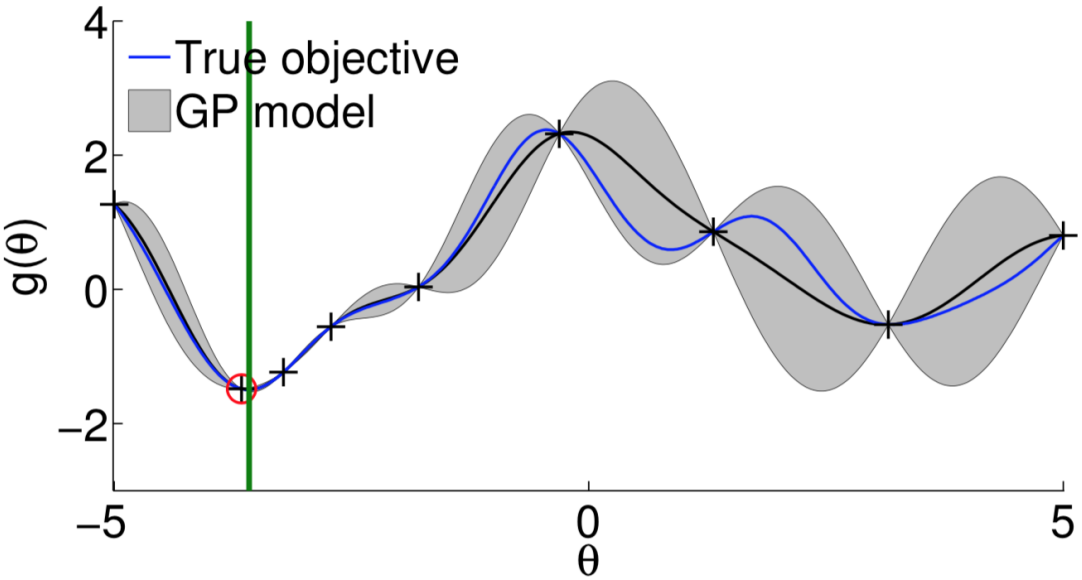
Bayesian Optimization: Illustration



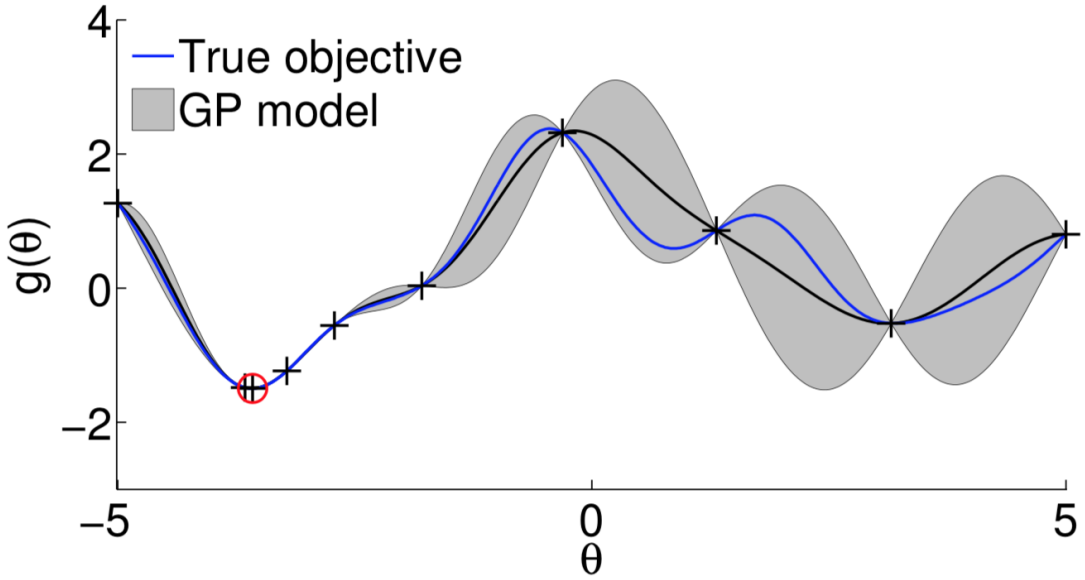
Bayesian Optimization: Illustration



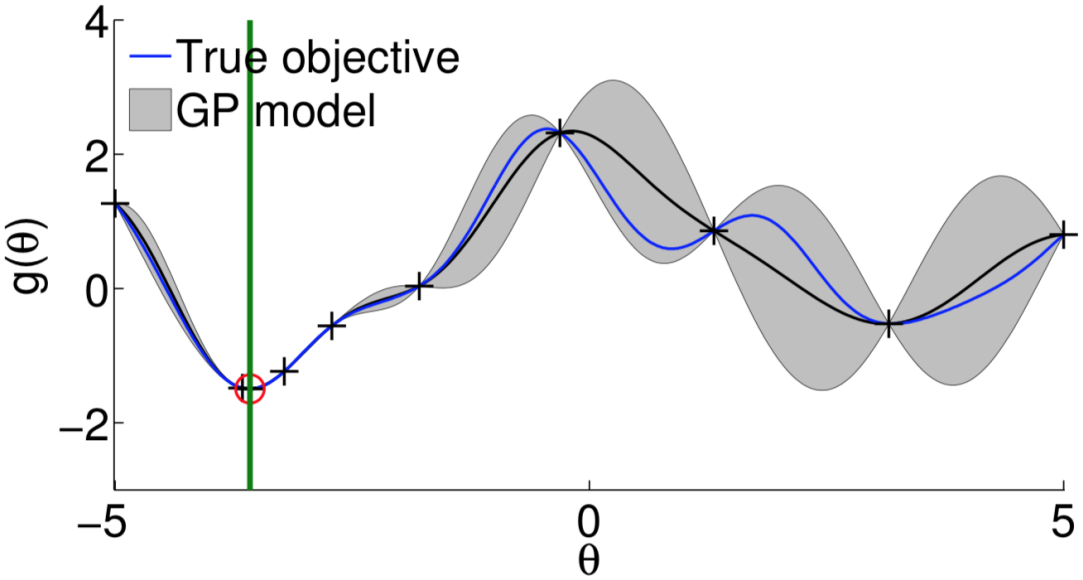
Bayesian Optimization: Illustration



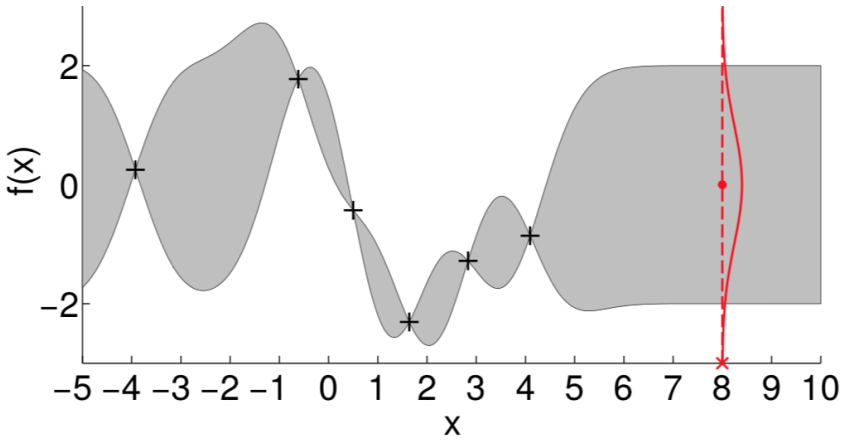
Bayesian Optimization: Illustration



Bayesian Optimization: Illustration

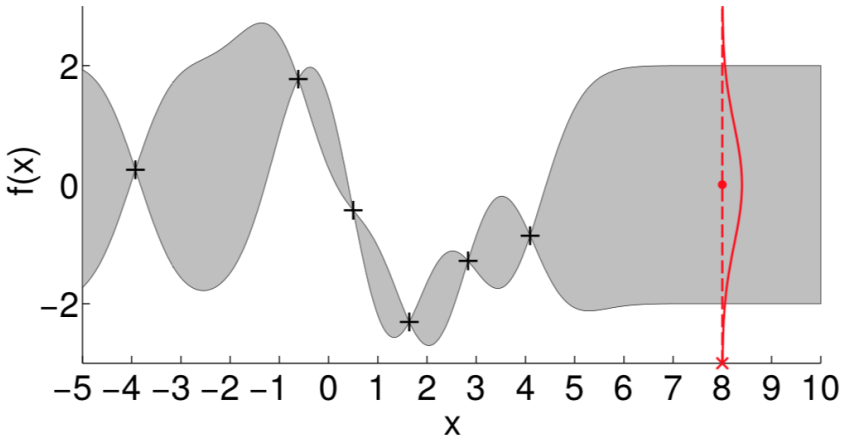


Using Uncertainty in Global Optimization



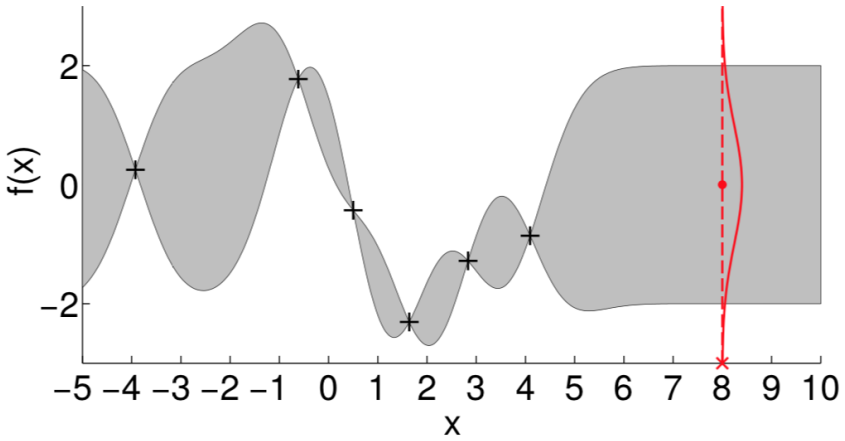
- Goal: choosing the next point to evaluate the true objective
- Find a good (global) optimum $\Longrightarrow$ Need to get out of local optima

Using Uncertainty in Global Optimization



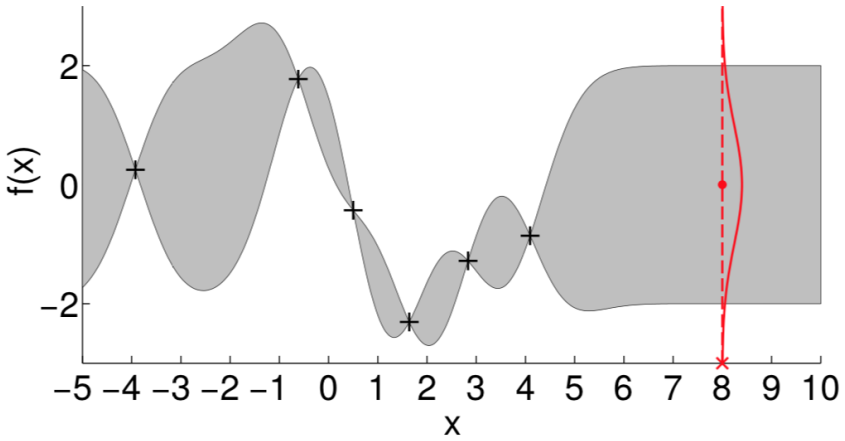
- Goal: choosing the next point to evaluate the true objective
- Find a good (global) optimum $\Longrightarrow$ Need to get out of local optima
- Extrapolate from collected observations

Using Uncertainty in Global Optimization



- Goal: choosing the next point to evaluate the true objective
- Find a good (global) optimum $\Longrightarrow$ Need to get out of local optima
- Extrapolate from collected observations
- GP gives us closed-form means and variances $\Longrightarrow$ Trade off exploration and exploitation
  - Exploration: seek places with high variance
  - Exploitation: seek places with low mean

Using Uncertainty in Global Optimization



- Goal: choosing the next point to evaluate the true objective
- Find a good (global) optimum $\Longrightarrow$ Need to get out of local optima
- Extrapolate from collected observations
- GP gives us closed-form means and variances $\Longrightarrow$ Trade off exploration and exploitation
  - Exploration: seek places with high variance
  - Exploitation: seek places with low mean
- Acquisition function α trades off exploration and exploitation for our proxy optimization

Gaussian Processes One-slide Review

- Due to Gaussian form
  $\Longrightarrow$ closed-form solutions for many useful questions about finite data
- Log likelihood:

$$\ln p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = -\frac{N}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{K}_{\boldsymbol{\theta}}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{y}$$

- Predictive distribution at test point y^{test}:

$$y^{test} \sim \mathcal{N}(\mu, \sigma^2)$$

$$\mu = \boldsymbol{k}_{\boldsymbol{\theta}}^T\boldsymbol{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{y} \qquad \sigma = k_{\theta}^{test} - \boldsymbol{k}_{\boldsymbol{\theta}}^T\boldsymbol{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{k}_{\boldsymbol{\theta}}$$

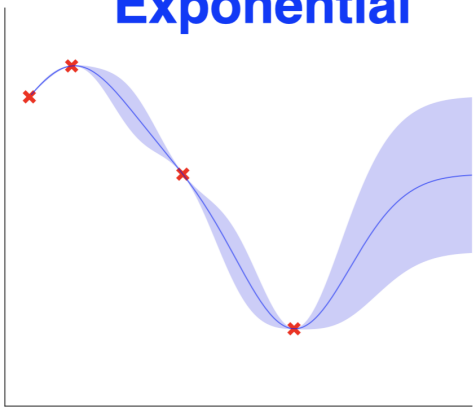
- We compute these matrices from the covariance function:

$$[\boldsymbol{K}_{\theta}]_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})$$

$$[\boldsymbol{k}_{\theta}]_i = K(\boldsymbol{x}^{test}, \boldsymbol{x}_i; \boldsymbol{\theta}) \qquad k_{\theta}^{test} = K(\boldsymbol{x}^{test}, \boldsymbol{x}^{test}; \boldsymbol{\theta})$$

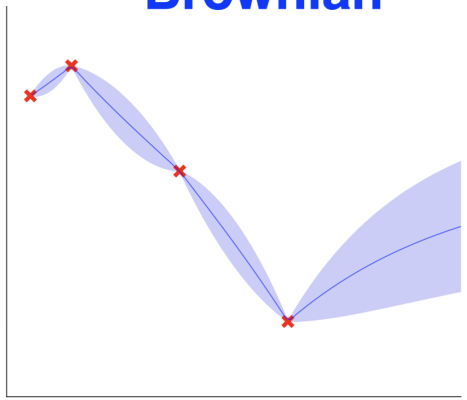
Examples of GP Kernels

Exponential



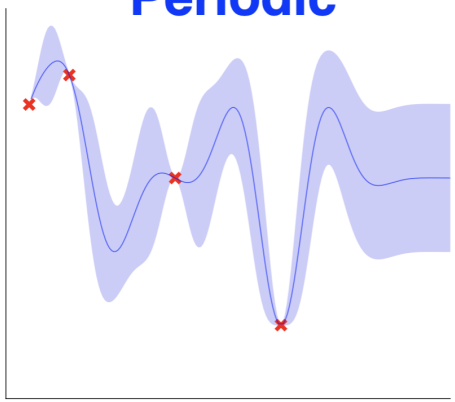
$$\sigma^2 e^{-\frac{1}{2l^2}(x_i - x_j)^2}$$

Brownian



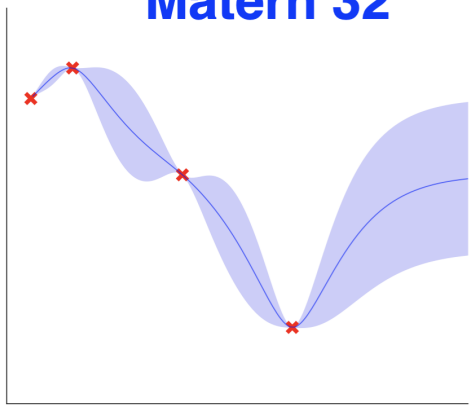
$$\sigma^2 e^{-\frac{1}{2l^2}|x_i - x_j|}$$

Periodic



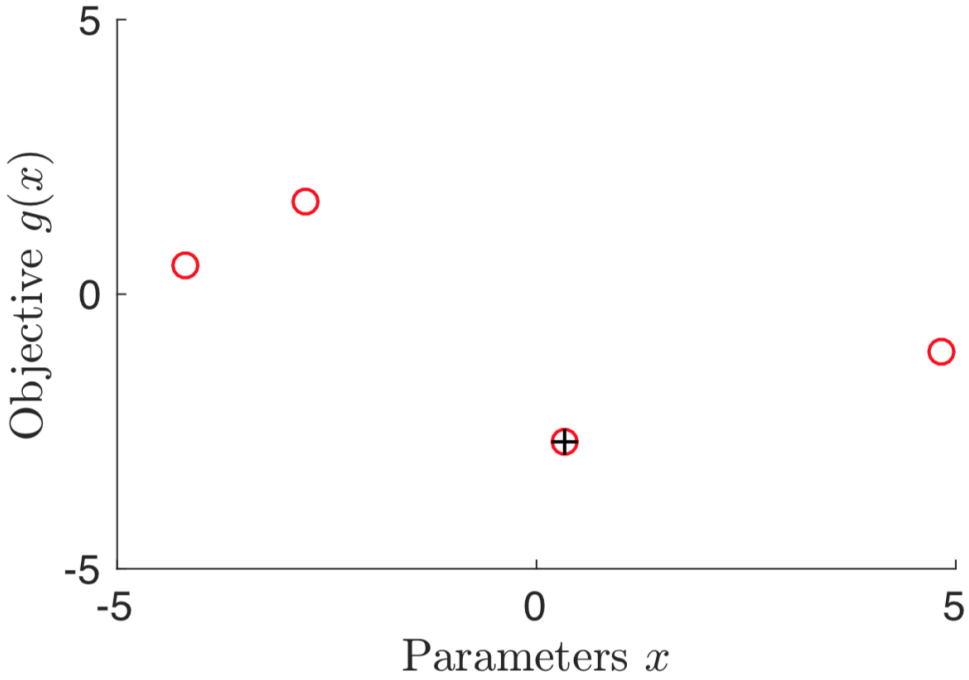
$$\sigma^2 \cos(\omega(x_i - x_j)) e^{-\frac{1}{2l^2}(x_i - x_j)^2}$$

Matérn 32

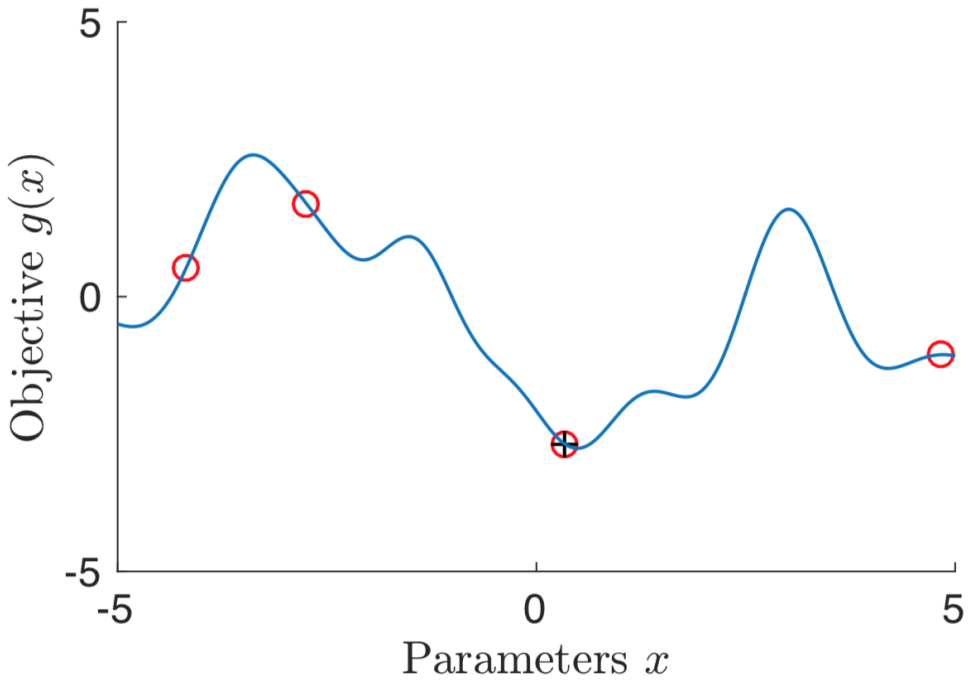


$$\sigma^2 \left(1 + \frac{\sqrt{3}\,|x_i - x_j|}{l}\right) e^{-\frac{\sqrt{3}\,|x_i - x_j|}{l}}$$

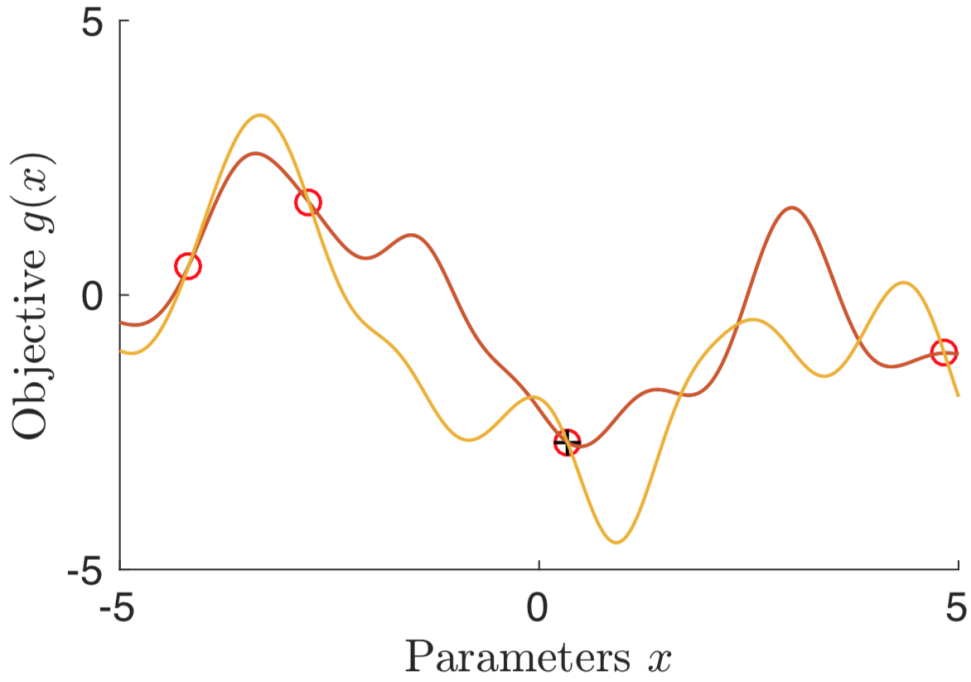
Where to Evaluate Next?



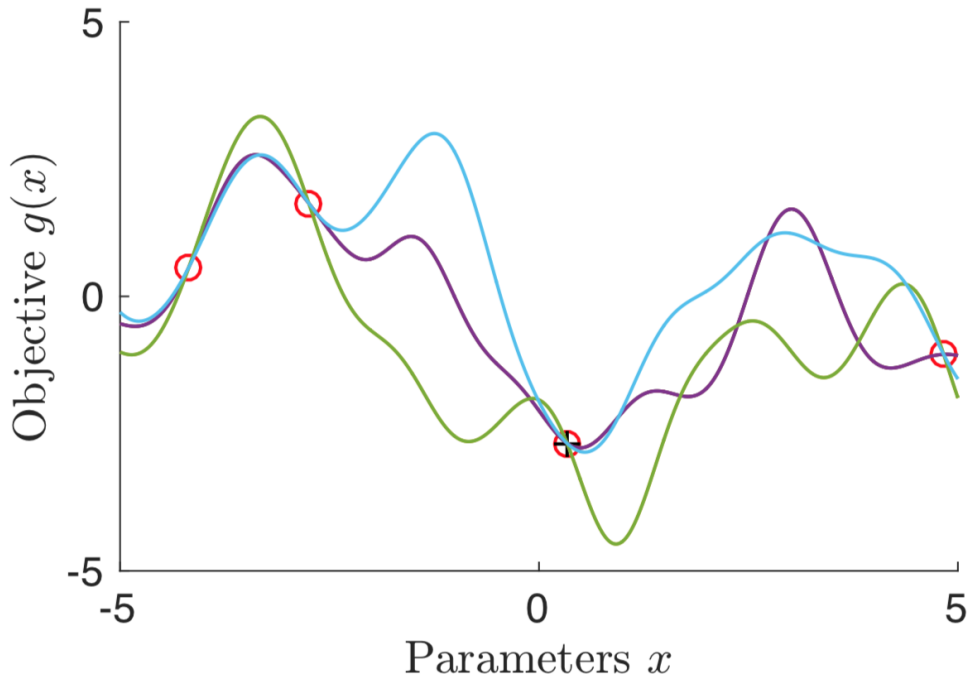
Where to Evaluate Next?



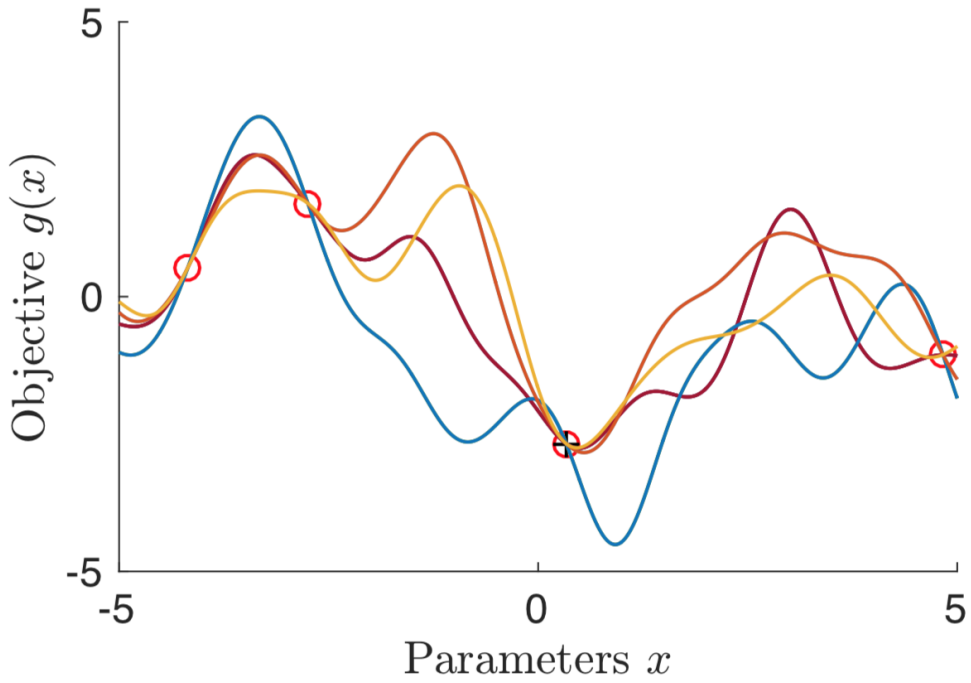
Where to Evaluate Next?



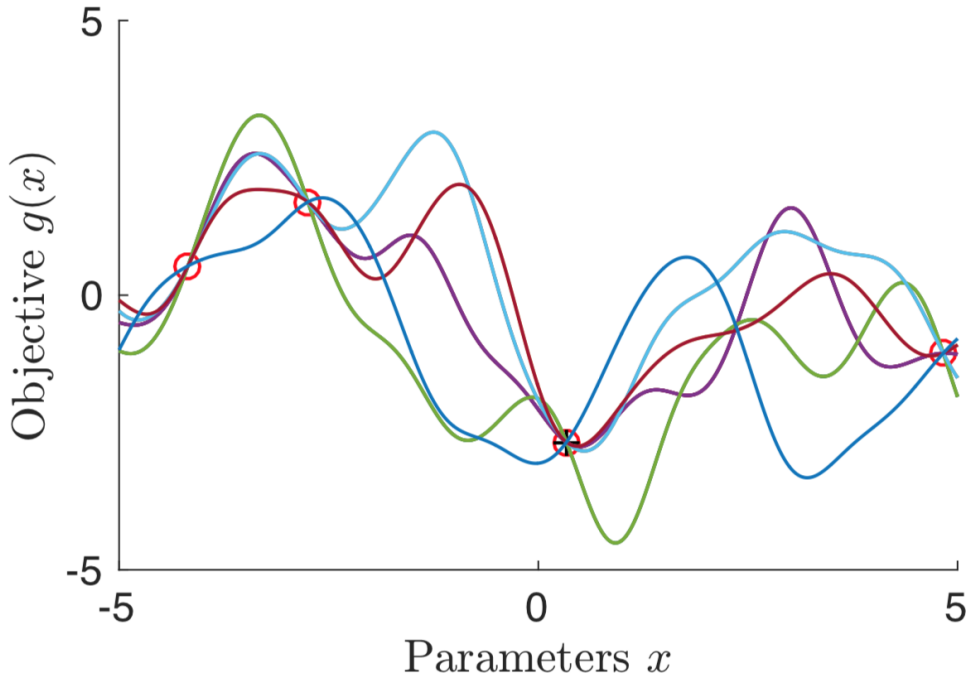
Where to Evaluate Next?



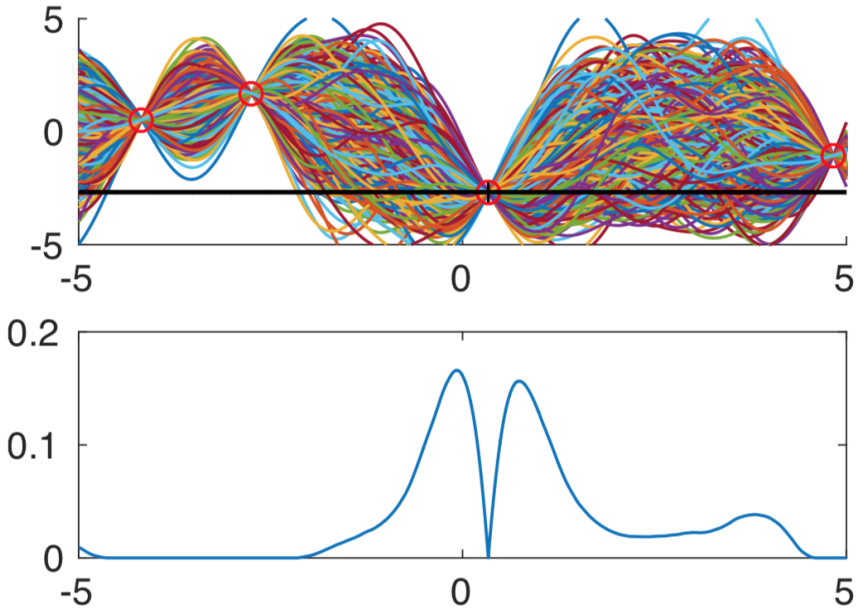
Where to Evaluate Next?



Where to Evaluate Next?



Where to Evaluate Next to Improve Most?



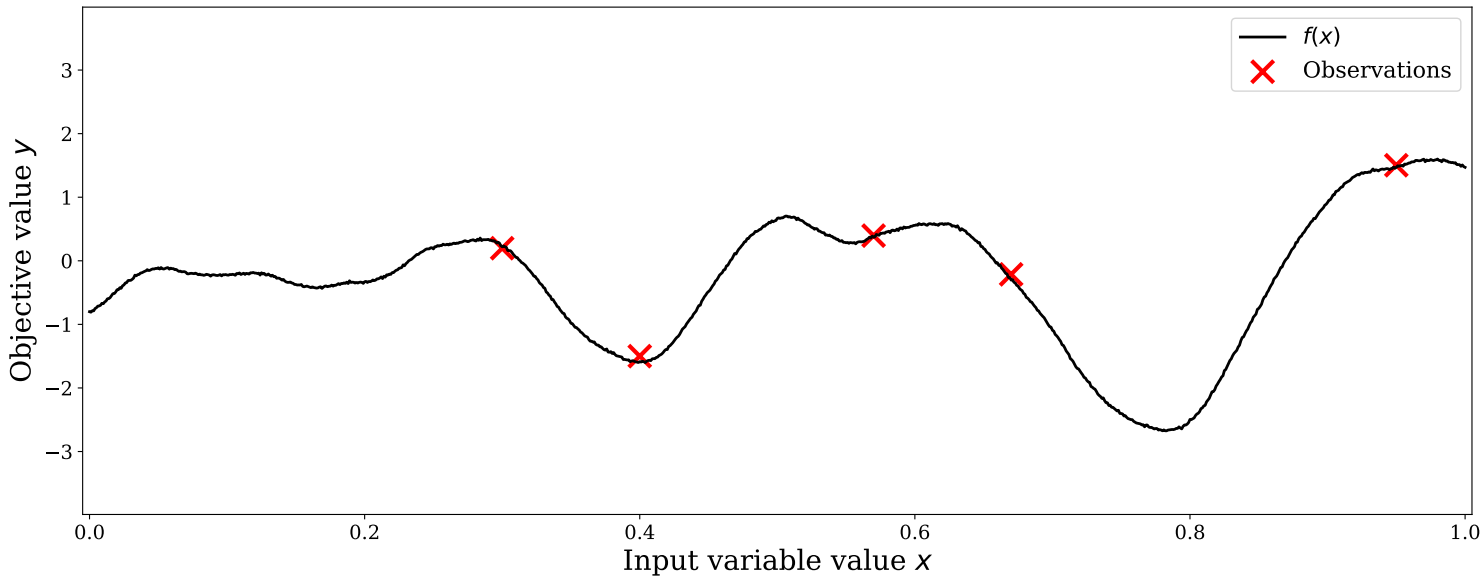
- Upper panel: Samples from a probabilistic model $\tilde{g}$
- Lower panel: Corresponding expected improvement over the best solution so far (black cross)
  - Evaluate g at the maximum of the expected improvement
  - Expected improvement is one possible acquisition function α

Expected Improvement (EI): Illustration

EI selects queries heuristically as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}\left[[f_n^* - f(\boldsymbol{x})]^+\right] = \textbf{[*closed form*]}$$

It queries points that are *likely good* or *uncertain*

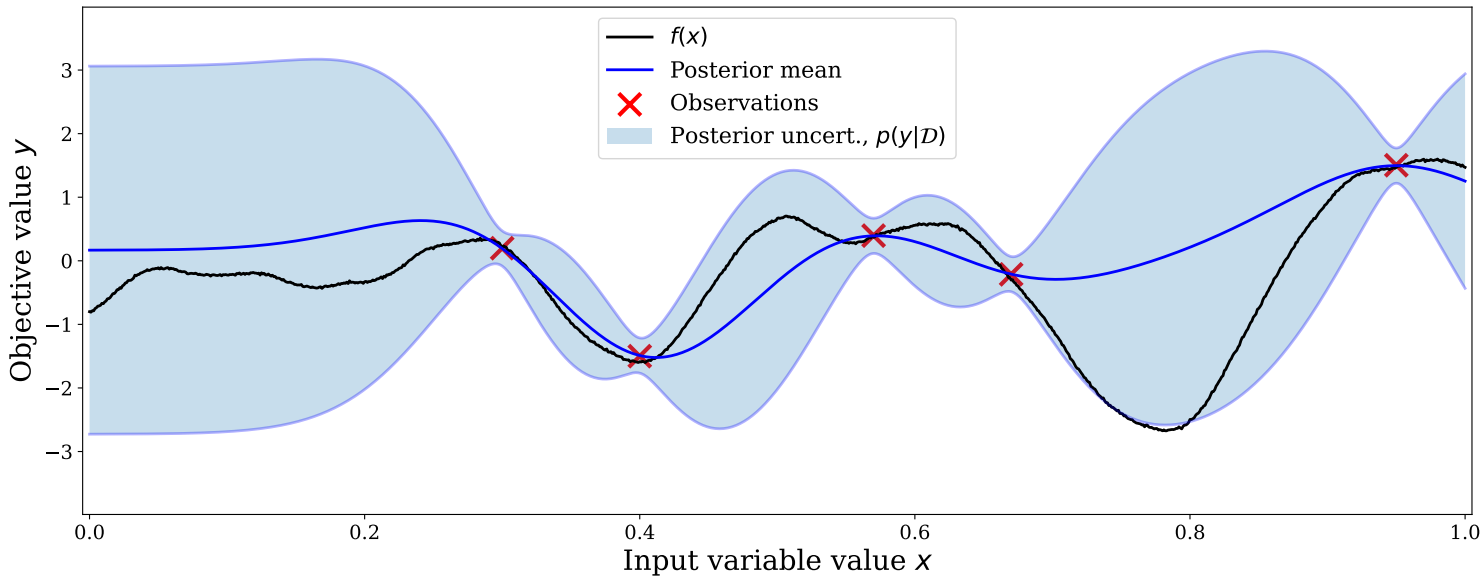


Expected Improvement (EI): Illustration

EI selects queries heuristically as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}\left[[f_n^* - f(\boldsymbol{x})]^+\right] = \textbf{[*closed form*]}$$

It queries points that are *likely good* or *uncertain*

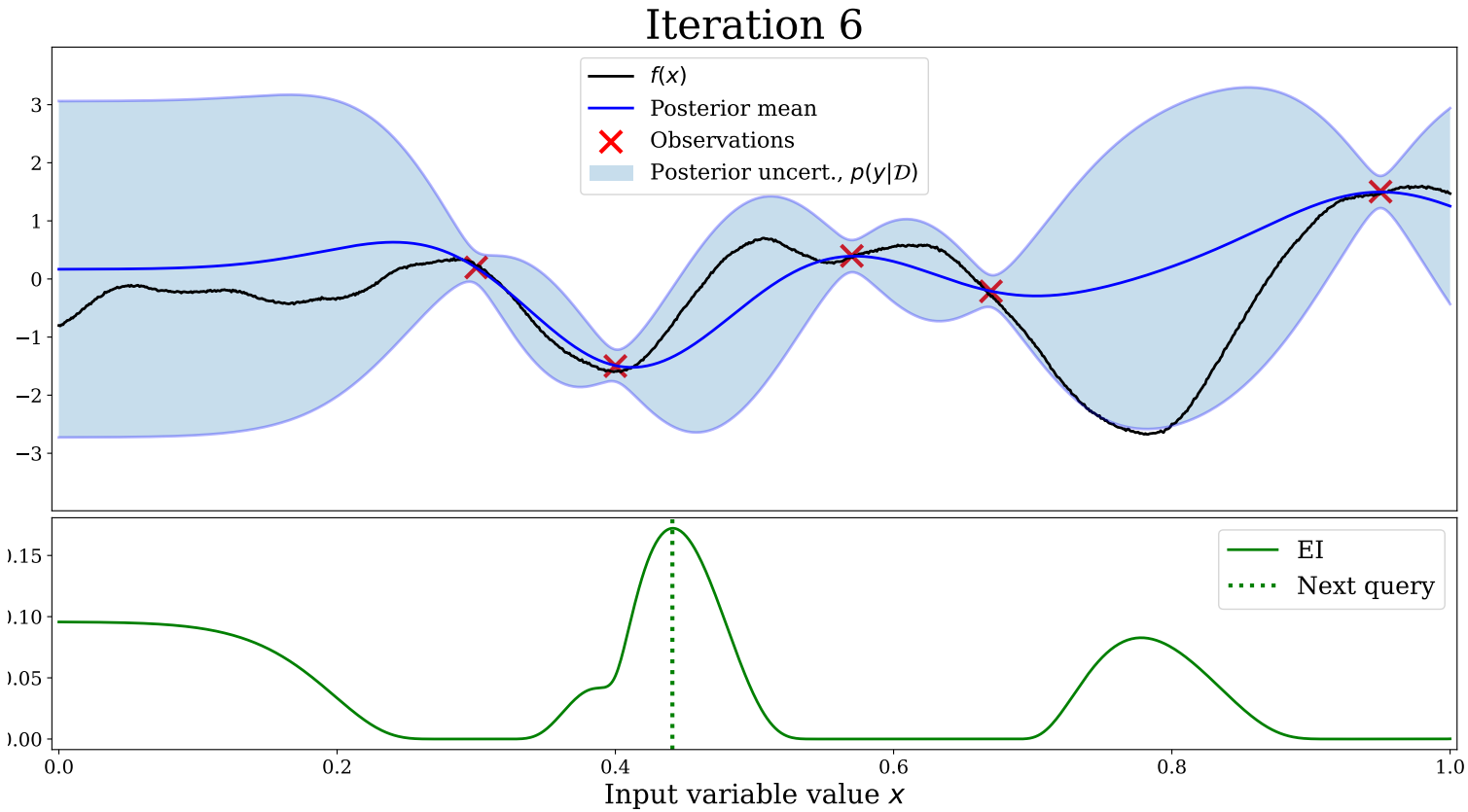


Expected Improvement (EI): Illustration

EI selects queries heuristically as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}\left[[f_n^* - f(\boldsymbol{x})]^+\right] = \textbf{[*closed form*]}$$

It queries points that are *likely good* or *uncertain*

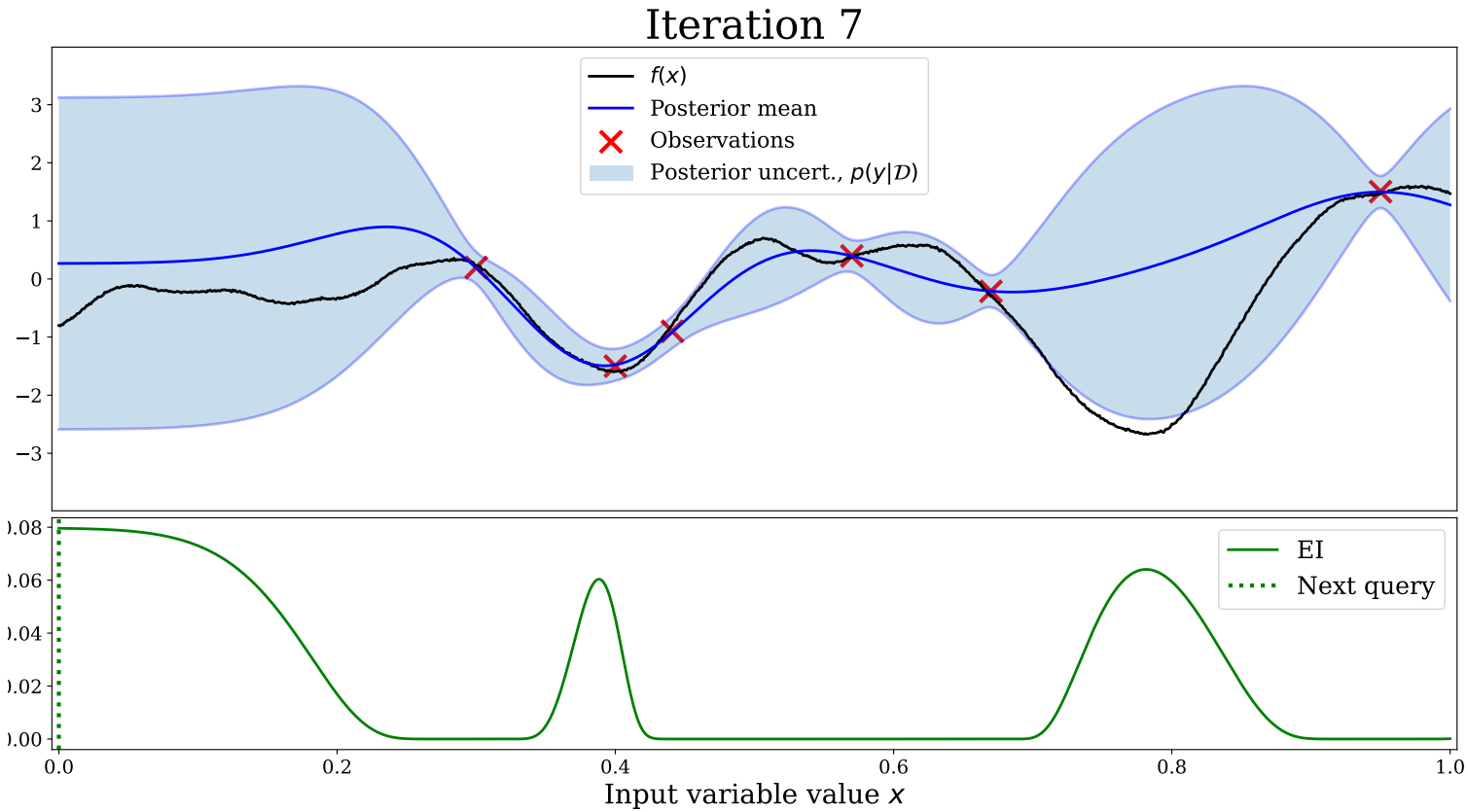


Expected Improvement (EI): Illustration

EI selects queries heuristically as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}\left[[f_n^* - f(\boldsymbol{x})]^+\right] = \textbf{[*closed form*]}$$

It queries points that are *likely good* or *uncertain*

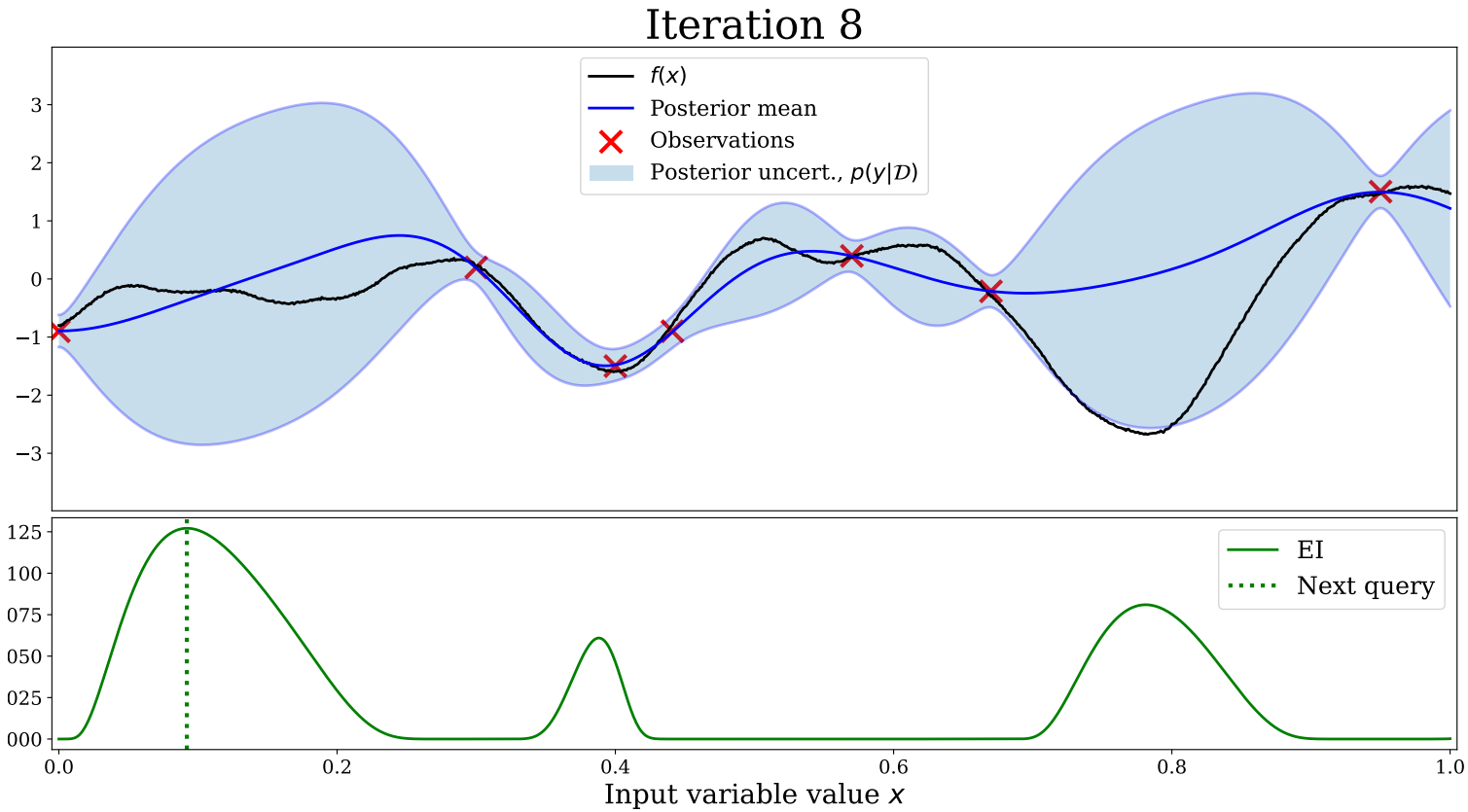


Expected Improvement (EI): Illustration

EI selects queries heuristically as

$$\boldsymbol{x}_{n+1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}\left[[f_n^* - f(\boldsymbol{x})]^+\right] = \textbf{[*closed form*]}$$

It queries points that are *likely good* or *uncertain*



Bayesian Optimization Pseudo-code

1: **Init**: Data set $\mathcal{D}_0 = \{\boldsymbol{X}_0, \boldsymbol{y}_0\}$
2: **for** $n = 1, 2, \ldots$ **do**
3: Update GP using data $\mathcal{D}_{n-1}$
4: Select $\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}} \alpha(\boldsymbol{x})$ by optimizing acquisition function
5: Query true objective g at $\boldsymbol{x}_n$
6: Augment data set $\mathcal{D}_n = \mathcal{D}_{n-1} \cup (\boldsymbol{x}_n, y_n)$
7: **end for**
8: **return** best input in $\mathcal{D}_n$: $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}} y(\boldsymbol{x})$

Closed-Form Acquisition Functions

- For all $\boldsymbol{x} \in \mathbb{R}^D$ the GP posterior gives a predictive mean $\mu(\boldsymbol{x})$ variance $\sigma^2(\boldsymbol{x})$. Define

$$\gamma(\boldsymbol{x}) = \frac{g(\boldsymbol{x}_{best}) - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$$

- Probability of Improvement [Kushner, 1964]:

$$\alpha_{PI}(\boldsymbol{x}) = \Phi(\gamma(\boldsymbol{x}))$$

- Expected Improvement [Mockus et al., 1978]:

$$\alpha_{EI}(\boldsymbol{x}) = \mathbb{E}[\max\{0, g(\boldsymbol{x}_{best}) - g(\boldsymbol{x})\}]$$

$$\implies \alpha_{EI}(\boldsymbol{x}) = \sigma(\boldsymbol{x})(\gamma(\boldsymbol{x})\Phi(\gamma(\boldsymbol{x})) + \mathcal{N}(\gamma(\boldsymbol{x})|0, 1))$$

- Lower Confidence Bound [Srinivas et al., 2010]:

$$\alpha_{LCB}(\boldsymbol{x}) = -(\mu(\boldsymbol{x}) - \kappa\sigma(\boldsymbol{x})), \kappa > 0$$

Test of Time Award – ICML 2020

Gaussian Process Upper Confidence Bound (GP-UCB) paper: [Srinivas et al., 2010]

We are thrilled to announce that the 2020 Test of Time Award goes to....

> Gaussian Process Optimization in the Bandit Setting:
> No Regret and Experimental Design

by: Niranjan Srinivas Andreas Krause Sham Kakade Matthias Seeger




> — **Abstract:** Many applications require optimizing an unknown, noisy function that is expensive to evaluate. We formalize this task as a multiarmed bandit problem, where the payoff function is either sampled from a Gaussian process (GP) or has low RKHS norm. We resolve the important open problem of deriving regret bounds for this setting, which imply novel convergence rates for GP optimization. We analyze GP-UCB, an intuitive upper-confidence based algorithm, and bound its cumulative regret in terms of maximal information gain, establishing a novel connection between GP optimization and experimental design. Moreover, by bounding the latter in terms of operator spectra, we obtain explicit sublinear regret bounds for many commonly used covariance functions. In some important cases, our bounds have surprisingly weak dependence on the dimensionality. In our experiments on real sensor data, GP-UCB compares favorably with other heuristical GP optimization approaches.

This paper brought together the fields of Bayesian optimization, bandits and experimental design by analyzing Gaussian process bandit optimization, giving a novel approach to derive finite-sample regret bounds in terms of a mutual information gain quantity. This paper has had profound impact over the past ten years, including the method itself, the proof techniques used, and the practical results. These have all enriched our community by sparking creativity in myriad subsequent works, ranging from theory to practice. When

Optimizing the Acquisition Function

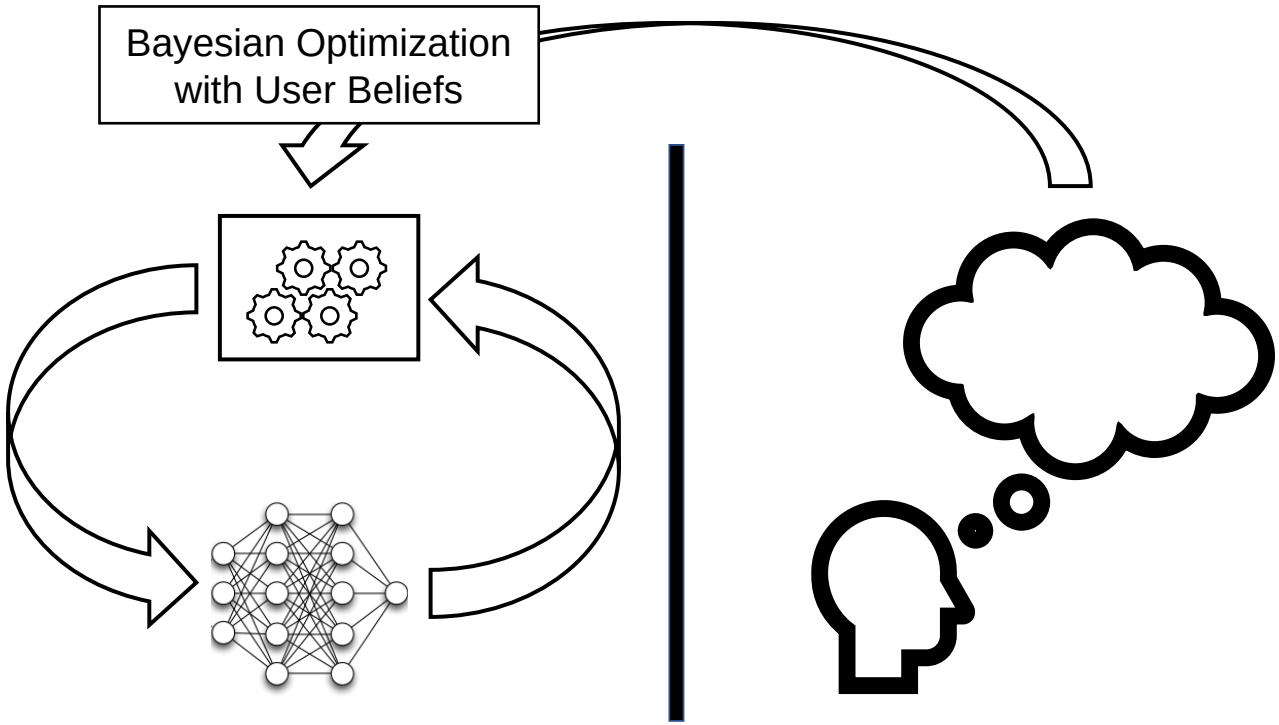
- Optimizing the acquisition function requires a global optimizer inside BO
- What have we gained?
- Evaluating the acquisition function is cheap compared to evaluating the true objective

  $\Longrightarrow$ We can afford evaluating it many times

Outline

The Idea Mentioned Earlier

Accelerate Bayesian optimization by allowing users to integrate beliefs that are typically reserved to manual search



Example belief: Adam's best learning rate is likely around 10^{-3}

Available Priors in Bayesian Optimization

Typically there are several ways prior knowledge can be injected in the optimization:

- Narrowing the search space

Available Priors in Bayesian Optimization

Typically there are several ways prior knowledge can be injected in the optimization:

- Narrowing the search space
- Biasing the initial design

Available Priors in Bayesian Optimization

Typically there are several ways prior knowledge can be injected in the optimization:

- Narrowing the search space
- Biasing the initial design
- Transforming the inputs and/or outputs

Available Priors in Bayesian Optimization

Typically there are several ways prior knowledge can be injected in the optimization:

- Narrowing the search space
- Biasing the initial design
- Transforming the inputs and/or outputs
- Using a prior over functions $p(f)$ via the GP kernel

Available Priors in Bayesian Optimization

Typically there are several ways prior knowledge can be injected in the optimization:

- Narrowing the search space
- Biasing the initial design
- Transforming the inputs and/or outputs
- Using a prior over functions $p(f)$ via the GP kernel
- Assuming monotonicity of the objective or non-stationary covariance

Available Priors in Bayesian Optimization

Typically there are several ways prior knowledge can be injected in the optimization:

- Narrowing the search space
- Biasing the initial design
- Transforming the inputs and/or outputs
- Using a prior over functions $p(f)$ via the GP kernel
- Assuming monotonicity of the objective or non-stationary covariance
- Transfer learning

Available Priors in Bayesian Optimization

Typically there are several ways prior knowledge can be injected in the optimization:

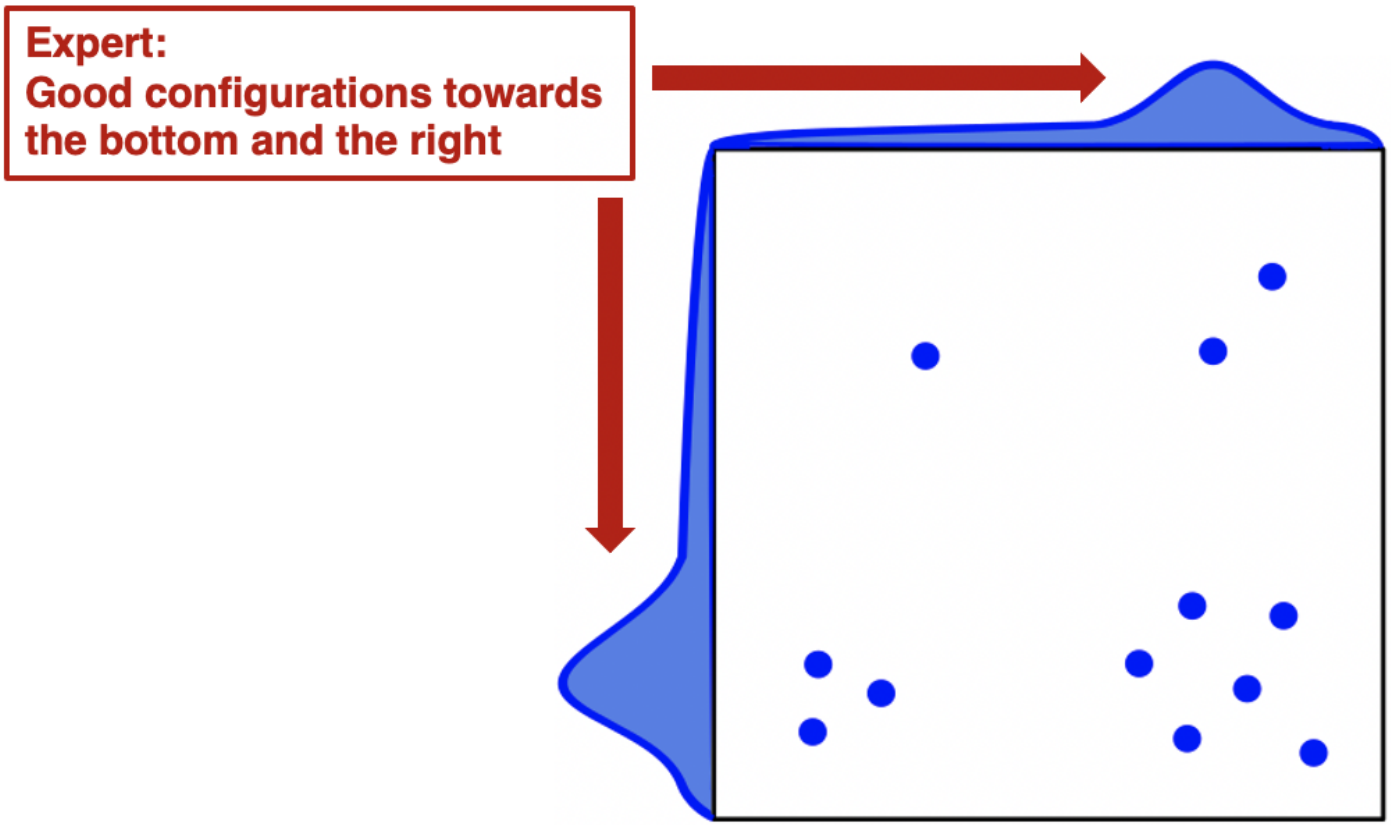
- Narrowing the search space
- Biasing the initial design
- Transforming the inputs and/or outputs
- Using a prior over functions $p(f)$ via the GP kernel
- Assuming monotonicity of the objective or non-stationary covariance
- Transfer learning

These are not aligned with the knowledge that users posses

Users know what ranges of hyperparameters tend to work best

Simple User Prior Belief

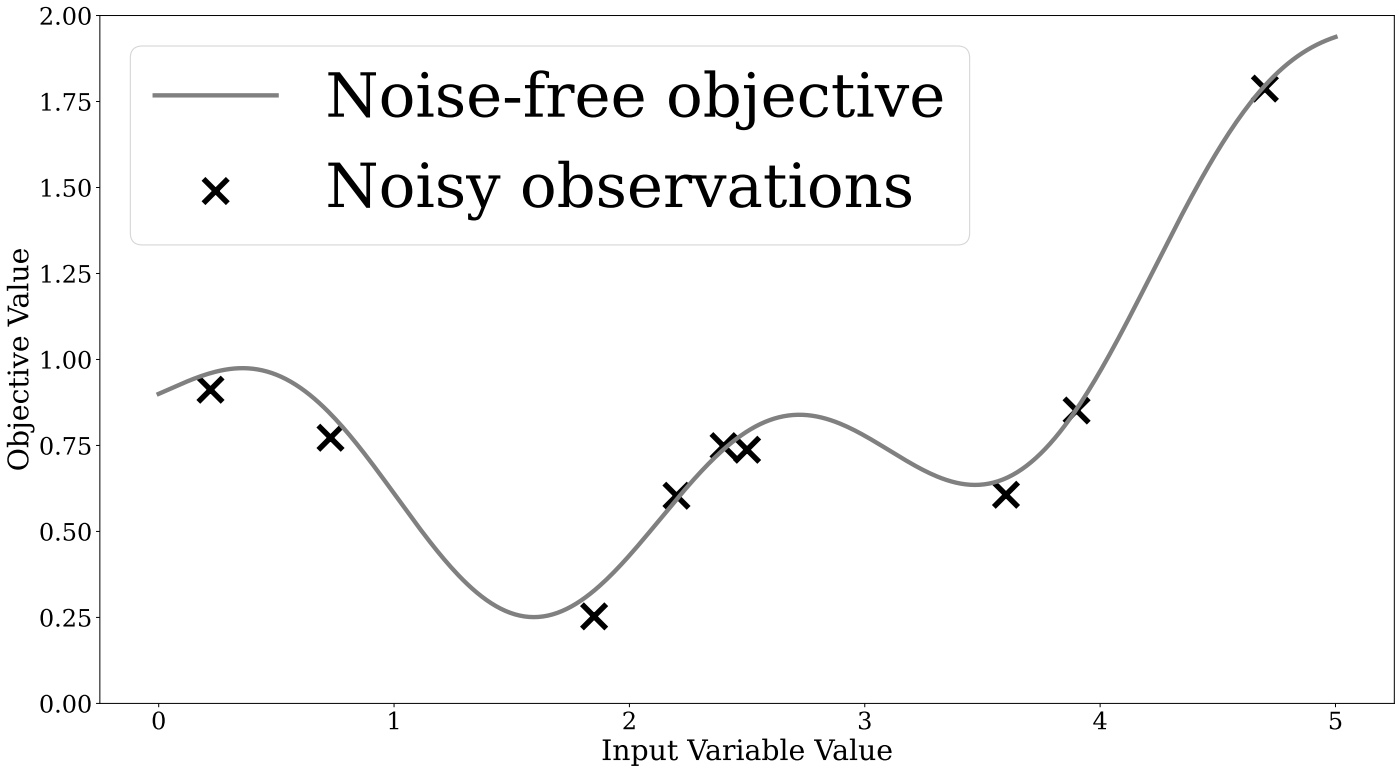
We propose one of the simplest forms of priors



Bayesian Optimization Problem Setting

Optimize a (noisy) black-box function f over $\mathcal{X} \subset \mathbb{R}^d$

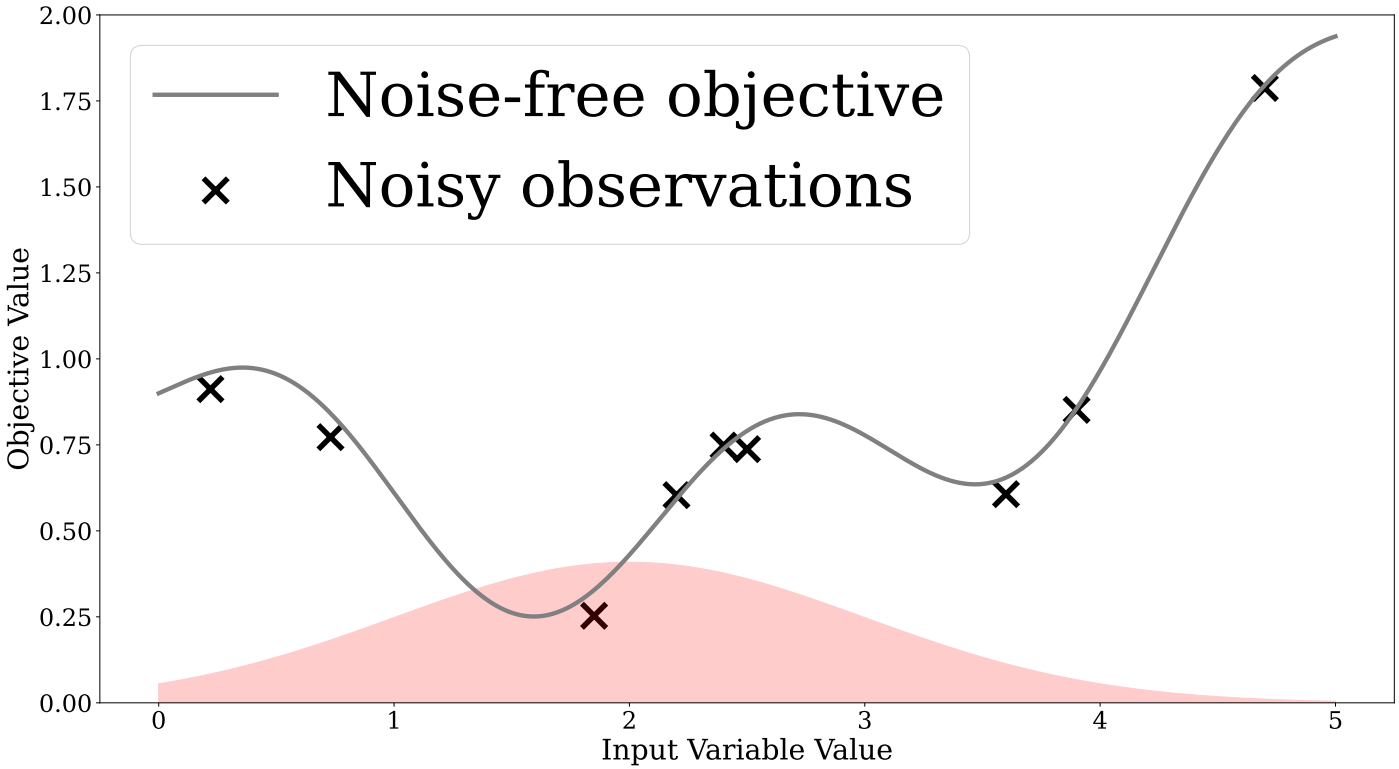
$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$$



BO with User Beliefs Problem Setting

User beliefs – A density on the location of the optimum

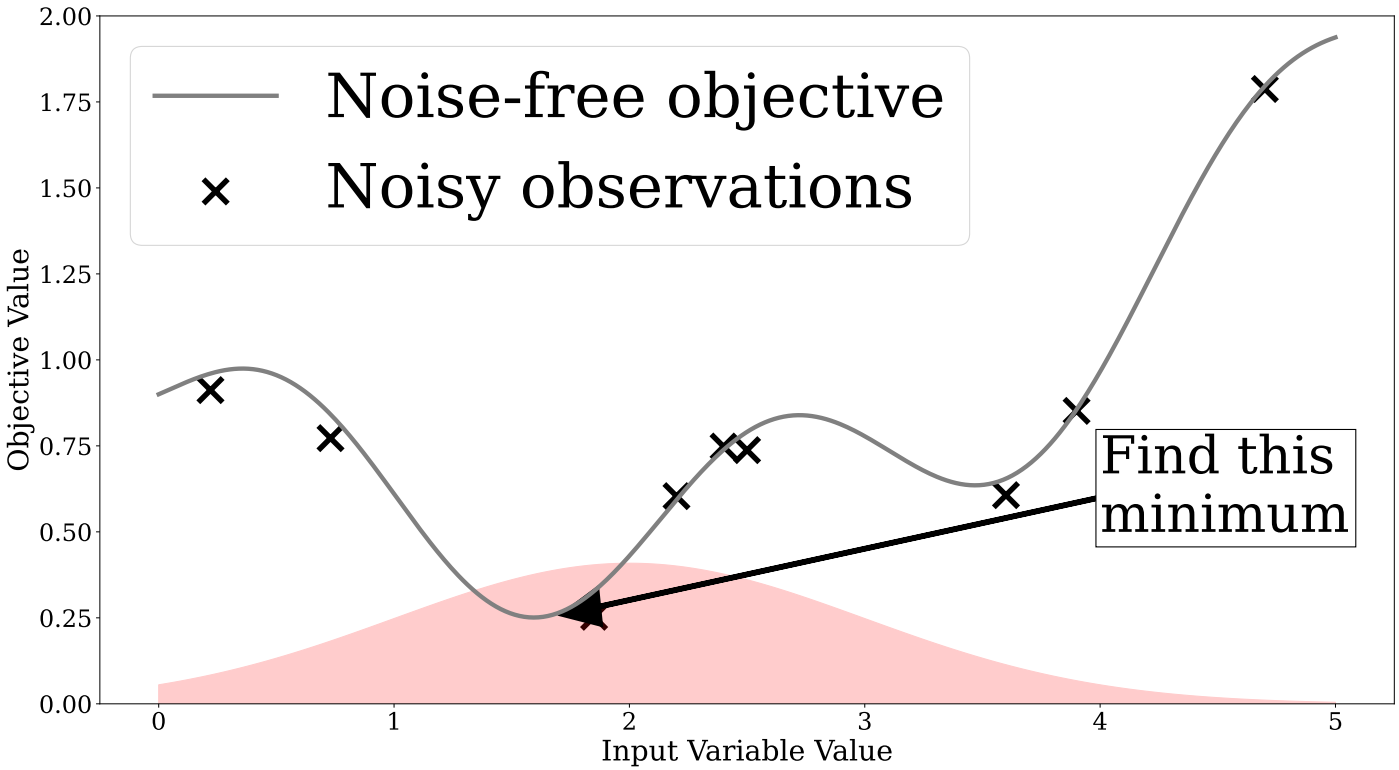
$$\pi(\boldsymbol{x}) = \mathbb{P}\left(f(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}')\right)$$



BO with User Beliefs Problem Setting

User beliefs – A density on the location of the optimum

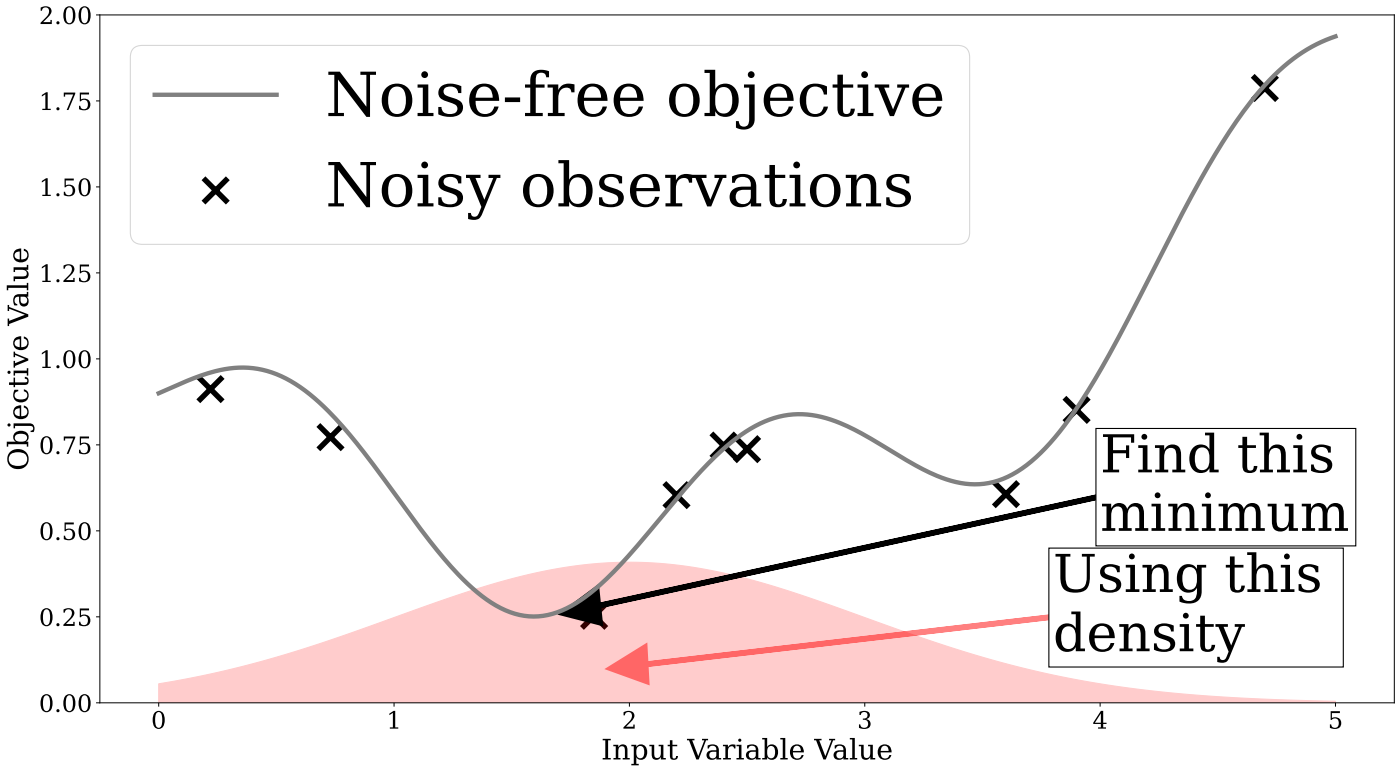
$$\pi(\boldsymbol{x}) = \mathbb{P}\left(f(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}')\right)$$



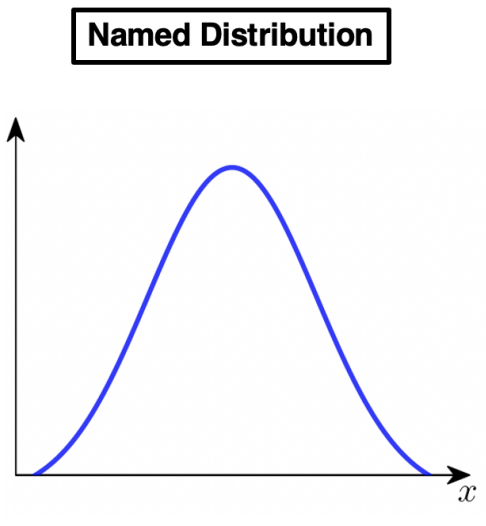
BO with User Beliefs Problem Setting

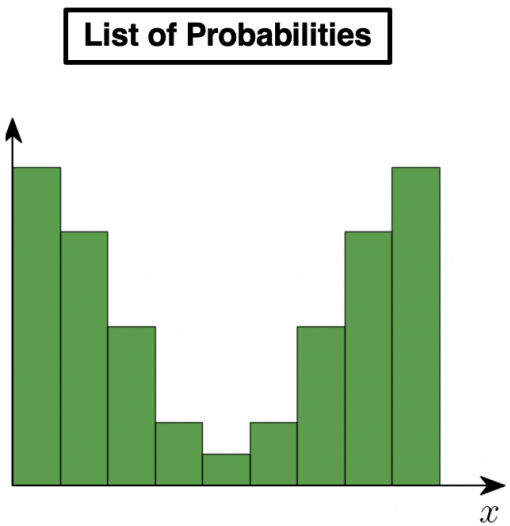
User beliefs – A density on the location of the optimum

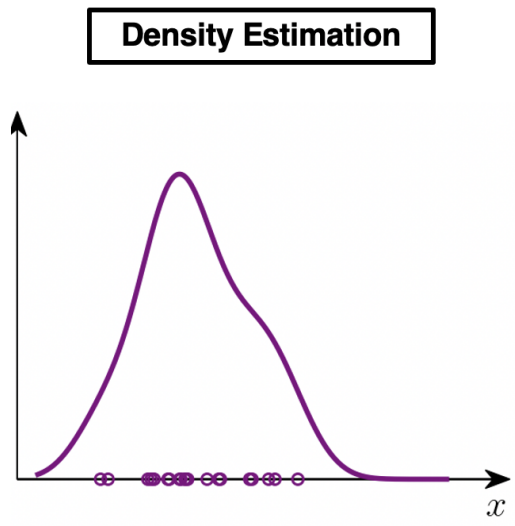
$$\pi(\boldsymbol{x}) = \mathbb{P}\left(f(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}')\right)$$



Examples of User Prior Beliefs







Approach Desiderata

- Incorporation of expert beliefs
- Robustness to poor priors
- Versatility across acquisition functions and priors

Prior-weighted Acquisition Function: πBO

Approach: Weight the acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ by the prior $\pi(\boldsymbol{x})$, raised to a time-dependent decay term $\beta/\boldsymbol{n}$

Prior-weighted Acquisition Function: πBO

Approach: Weight the acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ by the prior $\pi(\boldsymbol{x})$, raised to a time-dependent decay term β/n

The new decaying prior acquisition function takes into account $\pi(\boldsymbol{x})$ but gradually de-emphasize it

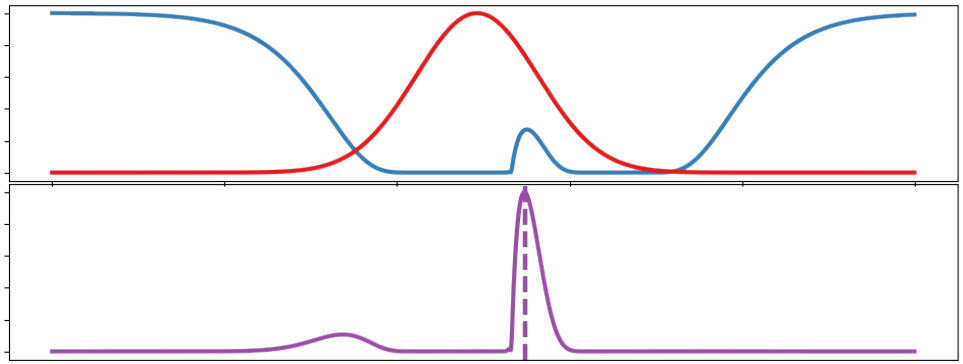
$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_\pi(\boldsymbol{x}, \mathcal{D}_n) = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n}$$

Prior-weighted Acquisition Function: πBO

Approach: Weight the acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ by the prior $\pi(\boldsymbol{x})$, raised to a time-dependent decay term β/n

The new decaying prior acquisition function takes into account $\pi(\boldsymbol{x})$ but gradually de-emphasize it

$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_\pi(\boldsymbol{x}, \mathcal{D}_n) = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n}$$

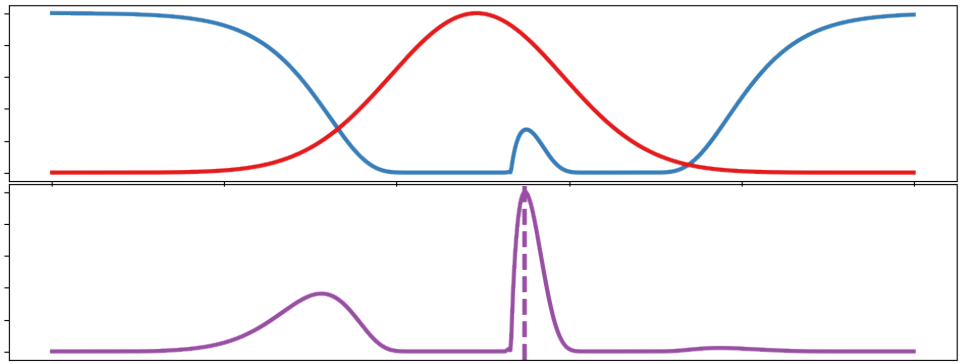


Prior-weighted Acquisition Function: πBO

Approach: Weight the acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ by the prior $\pi(\boldsymbol{x})$, raised to a time-dependent decay term β/n

The new decaying prior acquisition function takes into account $\pi(\boldsymbol{x})$ but gradually de-emphasize it

$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_\pi(\boldsymbol{x}, \mathcal{D}_n) = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n}$$

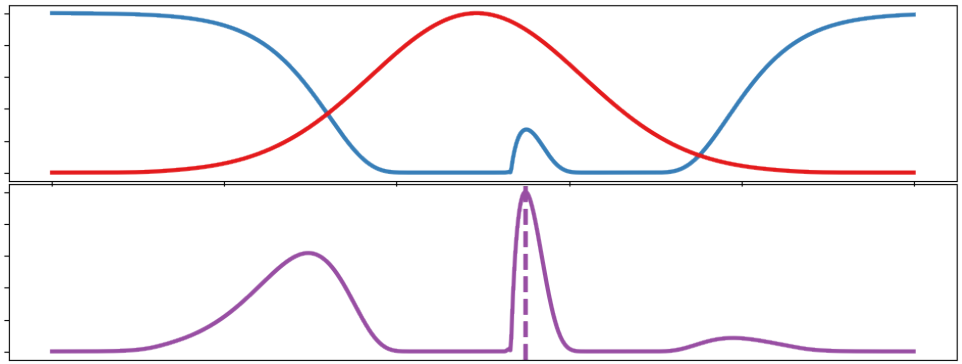


Prior-weighted Acquisition Function: πBO

Approach: Weight the acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ by the prior $\pi(\boldsymbol{x})$, raised to a time-dependent decay term β/n

The new decaying prior acquisition function takes into account $\pi(\boldsymbol{x})$ but gradually de-emphasize it

$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_\pi(\boldsymbol{x}, \mathcal{D}_n) = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n}$$

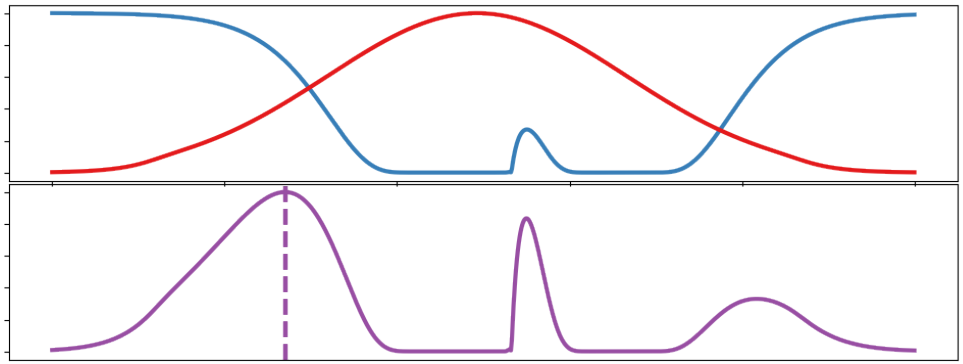


Prior-weighted Acquisition Function: πBO

Approach: Weight the acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ by the prior $\pi(\boldsymbol{x})$, raised to a time-dependent decay term β/n

The new decaying prior acquisition function takes into account $\pi(\boldsymbol{x})$ but gradually de-emphasize it

$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_\pi(\boldsymbol{x}, \mathcal{D}_n) = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n}$$

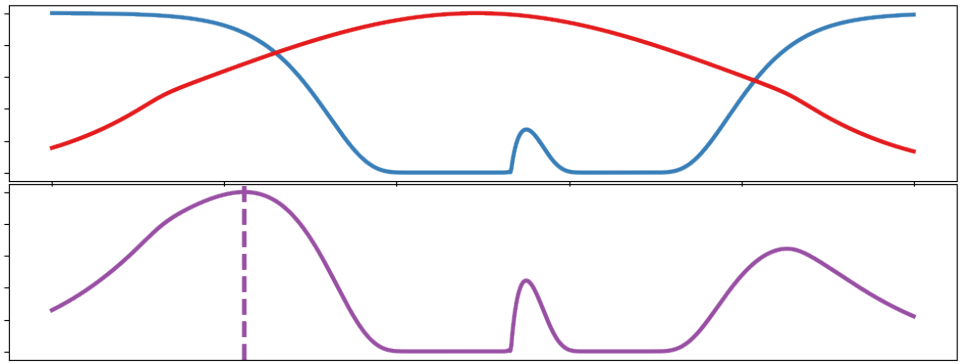


Prior-weighted Acquisition Function: πBO

Approach: Weight the acquisition function $\alpha(\boldsymbol{x}, \mathcal{D}_n)$ by the prior $\pi(\boldsymbol{x})$, raised to a time-dependent decay term β/n

The new decaying prior acquisition function takes into account $\pi(\boldsymbol{x})$ but gradually de-emphasize it

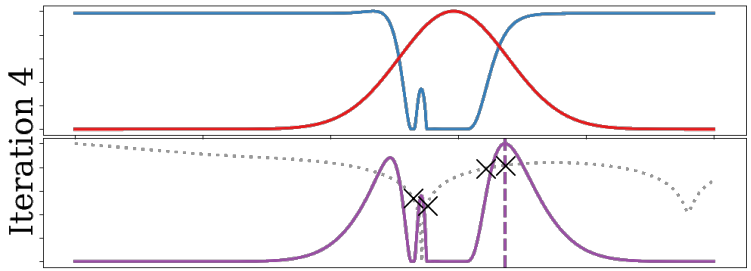
$$\boldsymbol{x}_n \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_\pi(\boldsymbol{x}, \mathcal{D}_n) = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x}, \mathcal{D}_n)\pi(\boldsymbol{x})^{\beta/n}$$



πBO Point Selection Illustrated

$\alpha(\boldsymbol{x}, \mathcal{D}_n)$, $\pi(\boldsymbol{x})^{\beta/n}$ and $\alpha_{\pi,n}(\boldsymbol{x}, \mathcal{D}_n)$ are visualized across iterations

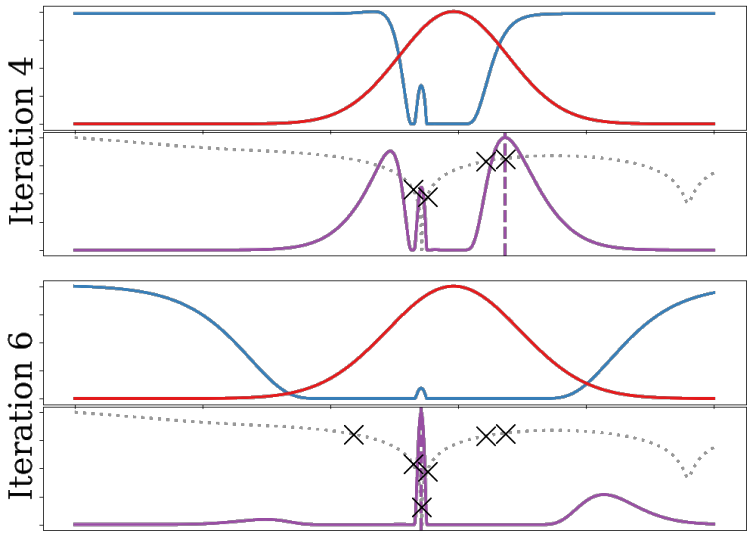
- High-probability regions are amplified early on



πBO Point Selection Illustrated

$\alpha(\boldsymbol{x}, \mathcal{D}_n)$, $\pi(\boldsymbol{x})^{\beta/n}$ and $\alpha_{\pi,n}(\boldsymbol{x}, \mathcal{D}_n)$ are visualized across iterations

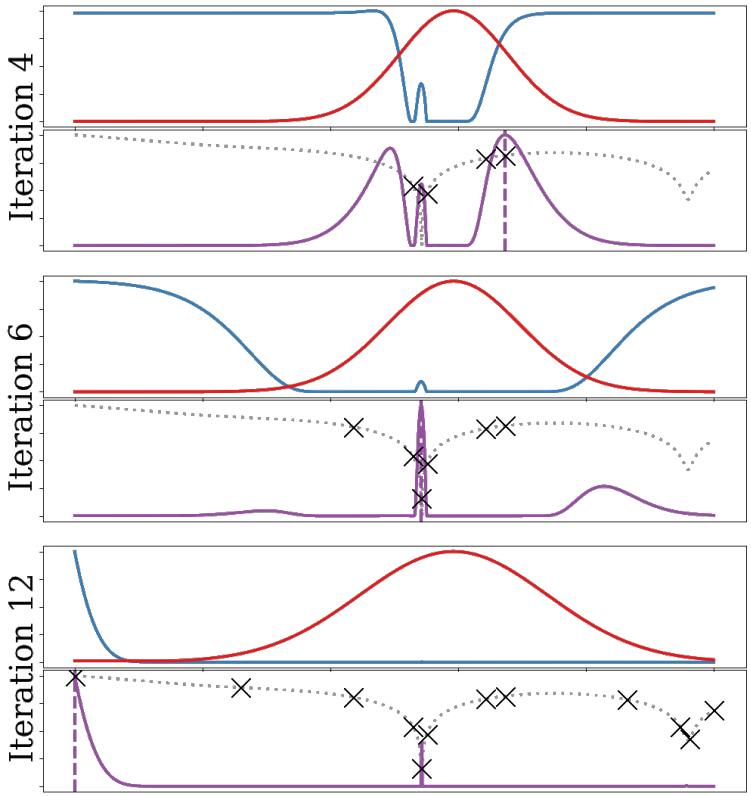
- High-probability regions are amplified early on
- The user-specified density becomes less peaked with time



πBO Point Selection Illustrated

$\alpha(\boldsymbol{x}, \mathcal{D}_n)$, $\pi(\boldsymbol{x})^{\beta/n}$ and $\alpha_{\pi,n}(\boldsymbol{x}, \mathcal{D}_n)$ are visualized across iterations

- High-probability regions are amplified early on
- The user-specified density becomes less peaked with time
- The rest of the search space is eventually explored



Theoretical Foundation of πBO

Theorem (Convergence of piBO)

The loss of πBO using EI is asymptotically equal to regular EI:

$$\mathcal{L}_n(EI_{\pi,n}, \mathcal{D}_n, \mathcal{H}_\ell(\mathcal{X}), R) \sim \mathcal{L}_n(EI_n, \mathcal{D}_n, \mathcal{H}_\ell(\mathcal{X}), R),$$

so we obtain a convergence rate for EI_π identical to EI [Bull, 2011]

$$\mathcal{L}_n(EI_{\pi,n}, \mathcal{D}_n, \mathcal{H}_\ell(\mathcal{X}), R) = \mathcal{O}(n^{-(\nu \wedge 1)/d}(\log n)^\gamma)$$

The convergence rate is independent of the choice of prior

Outline

Evaluation on Surrogate Tasks

We consider three surrogate tasks:

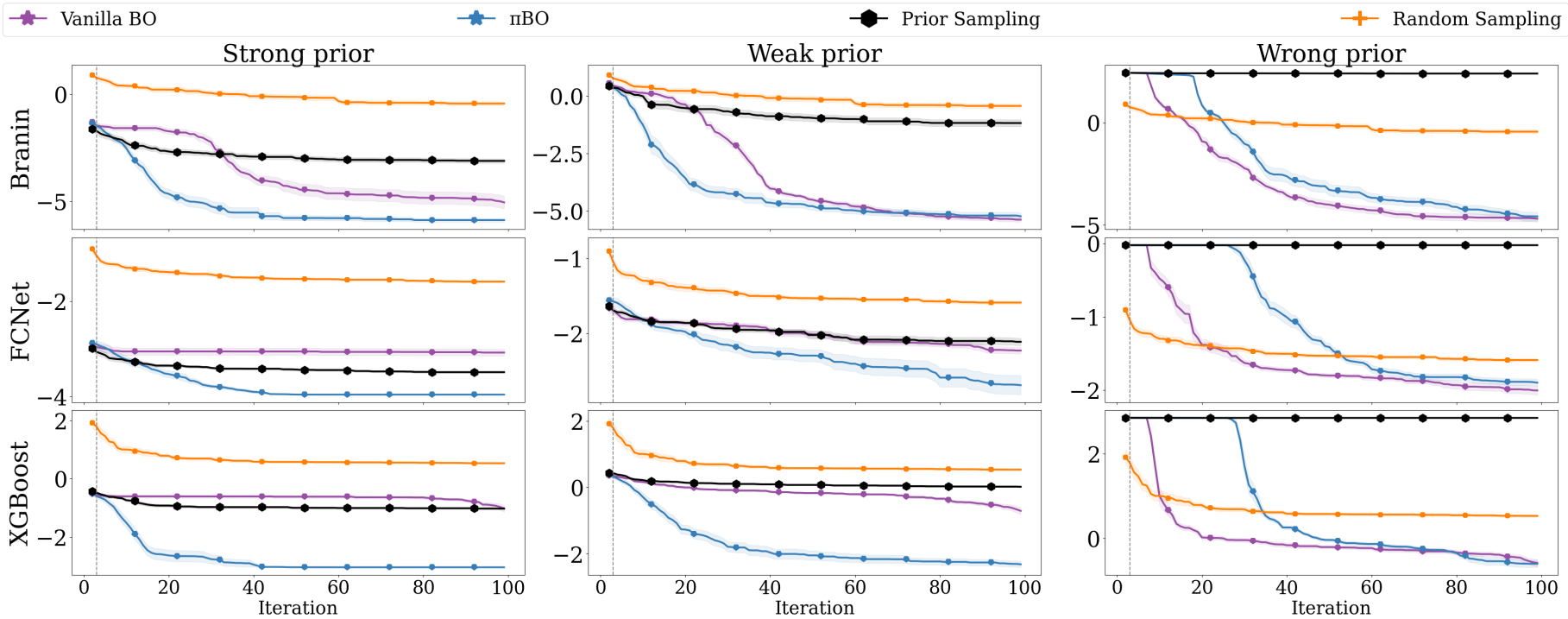
- Branin (2D)
- Profet FCNet (6D)
- Profet XGBoost (8D)

We consider three types of priors:

- Strong – Well located, high density on optimal region
- Weak – Well located, moderate density on optimal region
- Wrong – Poorly located, almost no density on optimum

Comparison to RS, prior sampling and prior-initialized BO (Spearmint [Snoek et al., 2012] + prior mode)

Evaluation on Surrogate Tasks – Results



Plots are in log scale

Takeaways:

- πBO is a superior option to BO with prior initialization
- πBO displays robustness to incorrect priors

Evaluation on MLP Classification Tasks

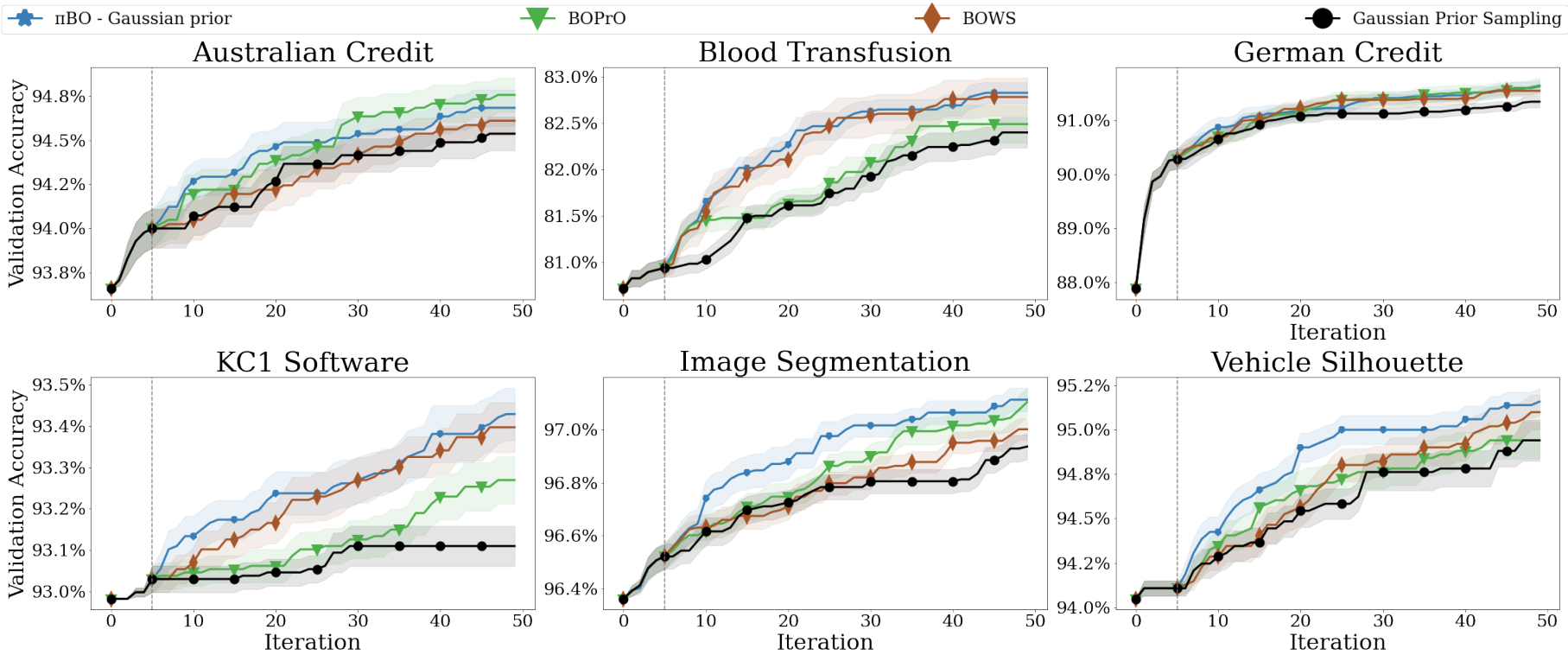
We compare πBO on six MLP classification tasks from the HPOBench benchmarking suite

Comparison against related work using priors:

- BOPrO [Souza et al., 2020a]
- BOWS [Ramachandran et al., 2020]
- Prior sampling

Priors are constructed as wide Gaussians around library default values

Evaluation on MLP Classification Tasks – Results



Takeaways:

- πBO yields superior performance on five out of six tasks
- BOPrO is competitive while BOWS struggles

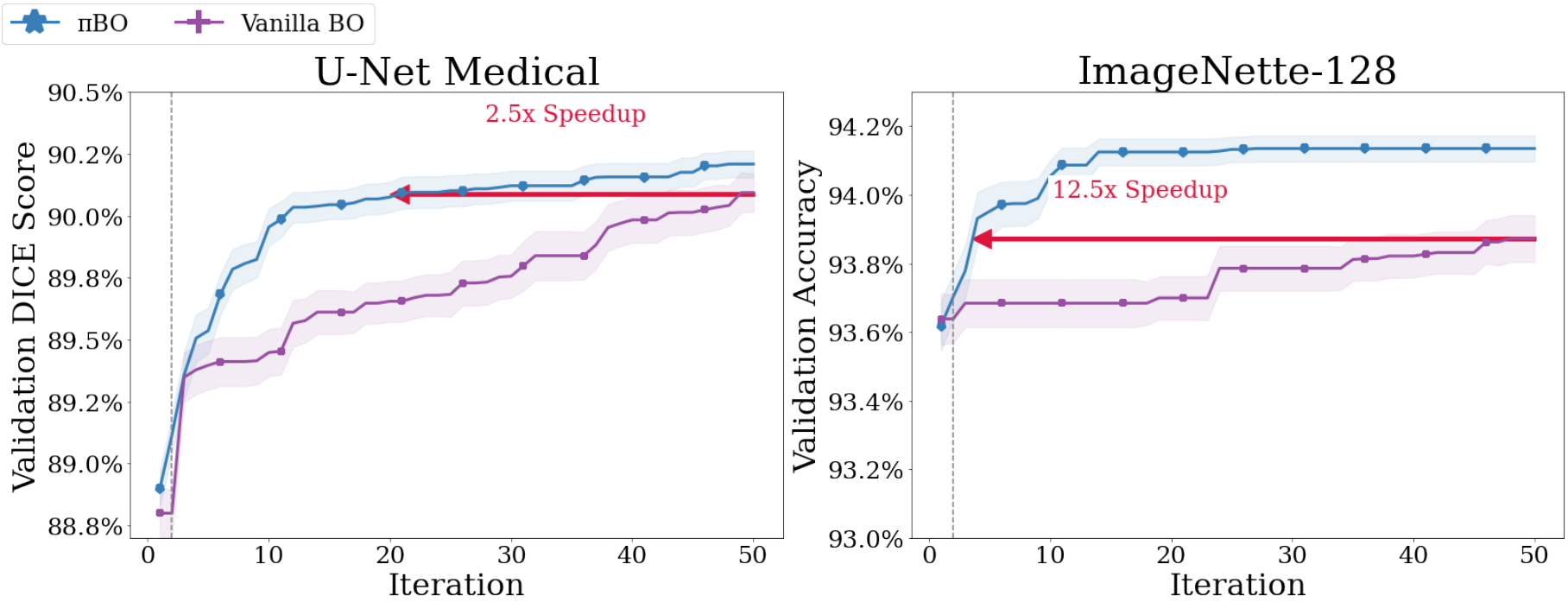
Evaluation on Computer Vision Tasks

We consider two popular CV architectures and tasks:

- U-Net (medical segmentation, 6D)
- ImageNette-128 (light-weight ImageNet, 6D)

We compare against vanilla BO provided with the default configuration

Evaluation on Computer Vision Tasks – Results



Takeaways:

- πBO yields superior performance on ImageNette establishing a new state-of-the-art at 94.14%
- Large speedups over prior-initialized BO (2.5× to 12.5×)

Outline

Exotic Applications #1: Compiler Autotuning

Integrated HyperMapper to 4 programming languages: Spatial [Nardi et al., 2019], HPVM [Ejjeh et al., 2022], TACO [Kjolstad et al., 2017], RISE/ELEVATE [Hagedorn et al., 2020]

- Autotuning of compilers for CPUs, GPUs, and FPGAs

Exotic Applications #1: Compiler Autotuning

Integrated HyperMapper to 4 programming languages: Spatial [Nardi et al., 2019], HPVM [Ejjeh et al., 2022], TACO [Kjolstad et al., 2017], RISE/ELEVATE [Hagedorn et al., 2020]

- Autotuning of compilers for CPUs, GPUs, and FPGAs
- For the Spatial language: Optimize runtime under the constraint that the design fits in the FPGA

Exotic Applications #1: Compiler Autotuning

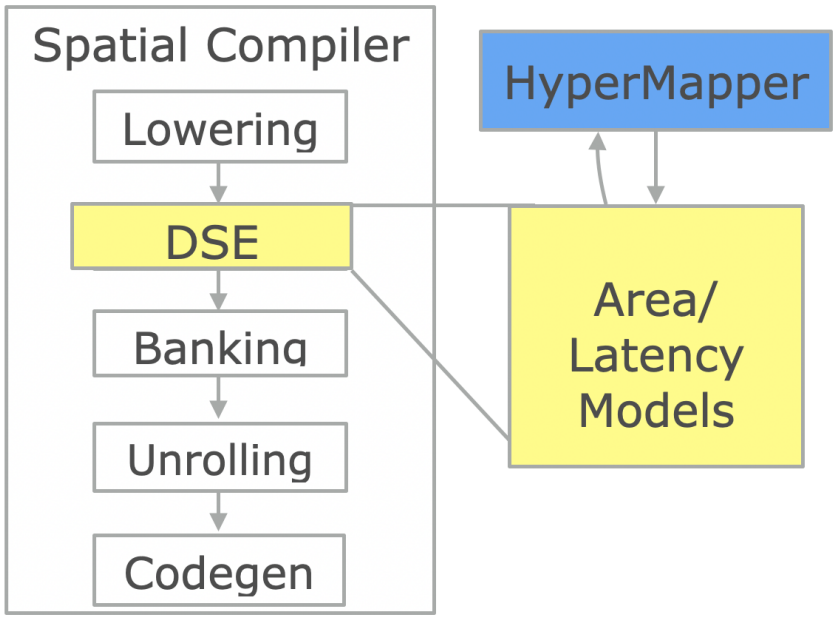
Integrated HyperMapper to 4 programming languages: Spatial [Nardi et al., 2019], HPVM [Ejjeh et al., 2022], TACO [Kjolstad et al., 2017], RISE/ELEVATE [Hagedorn et al., 2020]

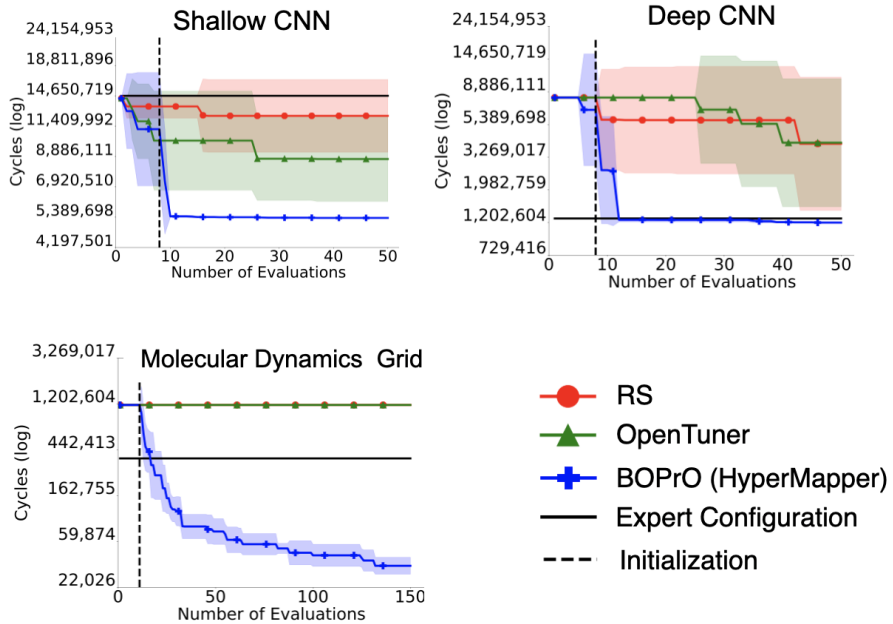
- Autotuning of compilers for CPUs, GPUs, and FPGAs
- For the Spatial language: Optimize runtime under the constraint that the design fits in the FPGA
- Expert Spatial programmer priors are available

Exotic Applications #1: Compiler Autotuning

Integrated HyperMapper to 4 programming languages: Spatial [Nardi et al., 2019], HPVM [Ejjeh et al., 2022], TACO [Kjolstad et al., 2017], RISE/ELEVATE [Hagedorn et al., 2020]

- Autotuning of compilers for CPUs, GPUs, and FPGAs
- For the Spatial language: Optimize runtime under the constraint that the design fits in the FPGA
- Expert Spatial programmer priors are available





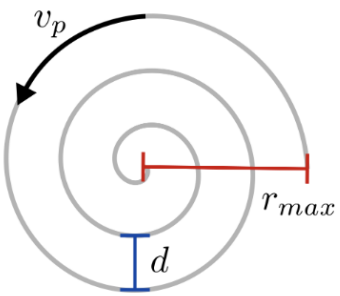
Exotic Applications #2: Robotics

Learning Skill-based Industrial Robot Tasks with User Priors [Mayr et al., 2022b, Mayr et al., 2022a, Mayr et al., 2021]

a) Peg Insertion

Learn insertion parameters

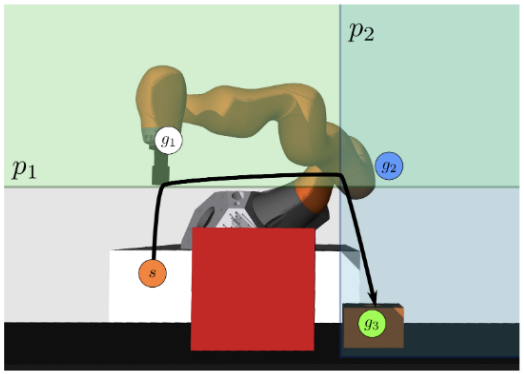
- Applied force
- Search motion
  - Path velocity
  - Maximum radius
  - Path Distance



b) Obstacle Avoidance

Learn suitable motion:

- Two intermediate goal points (y and z location)
- Two thresholds in the BT



▸ Presentation and ▸ video

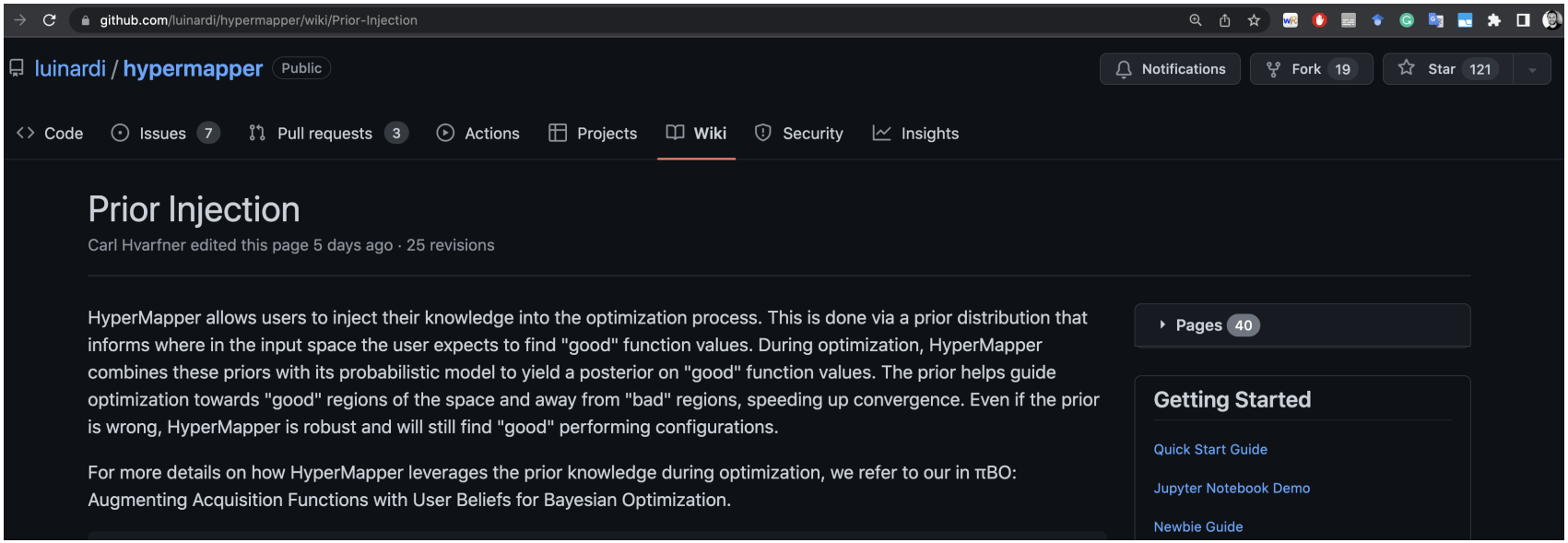
“I just want to use BO with priors”

The BO framework that my group maintains is HyperMapper: https://github.com/luinardi/hypermapper

HyperMapper implements both πBO [Hvarfner et al., 2022b] and BOPrO [Souza et al., 2020a]

SMAC supports πBO https://github.com/automl/SMAC3

Spearmint implementation is available under request



Conclusions

We have presented a human-centric approach to HPO

- πBO is flexible across myopic acquisition functions and surrogates
- Effective and conceptually simple
- Low-budget application settings benefit from user priors
- No expertise in BO or ML needed to inject user prior beliefs

Future Work

- What if we only have 10 evaluations?
  E.g. I can only train 10 models
- Use of multi-fidelity optimization with priors
- Integration of beliefs over optimal values
- Find a principled Bayesian approach for πBO
- Extend πBO to non-myopic acquisition functions

Material Related to This Lecture

Resources from our team:

- Practical design space exploration [Nardi et al., 2019]
- Prior-guided Bayesian optimization [Souza et al., 2020a]
- πBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization [Hvarfner et al., 2022b]
- PriorBand: HyperBand + Human Expert Knowledge [Mallik et al., 2022]

Other literature on priors:

- Incorporating expert prior in Bayesian optimisation via space warping [Ramachandran et al., 2020]
- Combining sequential model-based algorithm configuration with default-guided probabilistic sampling [Anastacio and Hoos, 2020]
- Incorporating expert prior knowledge into experimental design via posterior sampling [Li et al., 2020]
- Bayesian Optimization with Informative Covariance [Eduardo and Gutmann, 2022]

Recommended review on Bayesian optimization:

- A tutorial on Bayesian Optimization [Frazier, 2018]
- Taking the Human Out of the Loop: A Review of Bayesian Optimization [Shahriari et al., 2016]

Further Topics in BO

- Entropy-based acquisition functions
  [Hennig and Schuler, 2012, Hernández-Lobato et al., 2014, Wang and Jegelka, 2017, Hvarfner et al., 2022a]
- Non-myopic BO [Osborne et al., 2009]
- High-dimensional optimization
  [Wang et al., 2016, Papenmeier et al., 2022]
- Large-scale BO [Hutter et al., 2014, Eriksson et al., 2019]
- Non-GP BO:
  - Random Forests [Hutter et al., 2011]
  - Tree Parzen Estimators [Bergstra et al., 2011]
  - Bayesian Neural Networks [Springenberg et al., 2016]
- Constraints [Gardner et al., 2014, Gelbart et al., 2014]
- Automated machine learning [Hutter et al., 2018]
- Multi-objective, parallelizing
  [Paria et al., 2018, Snoek et al., 2012]
- Human-centric BO
  [Souza et al., 2020b, Hvarfner et al., 2022b]

References

[Anastacio and Hoos, 2020] Anastacio, M. and Hoos, H. H. (2020).
Combining sequential model-based algorithm configuration with default-guided probabilistic sampling.
In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 301–302.

[Bergstra et al., 2011] Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011).
Algorithms for hyper-parameter optimization.
In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.

[Bouthillier and Varoquaux, 2020] Bouthillier, X. and Varoquaux, G. (2020).
Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020.
PhD thesis, Inria Saclay Ile de France.

[Bull, 2011] Bull, A. D. (2011).
Convergence rates of efficient global optimization algorithms.
Journal of Machine Learning Research, 12(10).

[Criminisi et al., 2013] Criminisi, A., Robertson, D., Konukoglu, E., Shotton, J., Pathak, S., White, S., and Siddiqui, K. (2013).
Regression forests for efficient anatomy detection and localization in computed tomography scans.
Medical image analysis, 17(8):1293–1303.

[Eduardo and Gutmann, 2022] Eduardo, A. and Gutmann, M. U. (2022).
Bayesian optimization with informative covariance.
arXiv preprint arXiv:2208.02704.

[Ejjeh et al., 2022] Ejjeh, A., Medvinsky, L., Councilman, A., Nehra, H., Sharma, S., Adve, V., Nardi, L., Nurvitadhi, E., and Rutenbar, R. A. (2022).
HPVM2FPGA: Enabling true hardware-agnostic fpga programming.
In *IEEE International Conference on Application-specific Systems, Architectures, and Processors (ASAP)*.

References (cont.)

[Eriksson et al., 2019] Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019).
Scalable global optimization via local bayesian optimization.
Advances in neural information processing systems, 32.

[Frazier, 2018] Frazier, P. I. (2018).
A tutorial on bayesian optimization.

[Gardner et al., 2014] Gardner, J. R., Kusner, M. J., Xu, Z., Weinberger, K. Q., and Cunningham, J. P. (2014).
Bayesian optimization with inequality constraints.
In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–937–II–945. JMLR.org.

[Gelbart et al., 2014] Gelbart, M. A., Snoek, J., and Adams, R. P. (2014).
Bayesian optimization with unknown constraints.
arXiv preprint arXiv:1403.5607.

[Hagedorn et al., 2020] Hagedorn, B., Lenfers, J., Koehler, T., Qin, X., Gorlatch, S., and Steuwer, M. (2020).
Achieving high-performance the functional way: a functional pearl on expressing high-performance optimizations as rewrite strategies.
Proceedings of the ACM on Programming Languages, 4(ICFP):1–29.

[Hennig and Schuler, 2012] Hennig, P. and Schuler, C. J. (2012).
Entropy search for information-efficient global optimization.
Journal of Machine Learning Research, 13(Jun):1809–1837.

[Hernández-Lobato et al., 2014] Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014).
Predictive entropy search for efficient global optimization of black-box functions.
In *Advances in neural information processing systems*, pages 918–926.

References (cont.)

[Hutter et al., 2014] Hutter, F., Hoos, H., and Leyton-Brown, K. (2014).
An efficient approach for assessing hyperparameter importance.
In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages I–754–I–762. JMLR.org.

[Hutter et al., 2011] Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011).
Sequential model-based optimization for general algorithm configuration.
In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer.

[Hutter et al., 2018] Hutter, F., Kotthoff, L., and Vanschoren, J., editors (2018).
Automatic Machine Learning: Methods, Systems, Challenges.
Springer.
In press, available at http://automl.org/book.

[Hvarfner et al., 2022a] Hvarfner, C., Hutter, F., and Nardi, L. (2022a).
Joint entropy search for maximally-informed bayesian optimization.
In *Advances in Neural Information Processing Systems 35, NeurIPS 2022*.

[Hvarfner et al., 2022b] Hvarfner, C., Stoll, D., Souza, A., Nardi, L., Lindauer, M., and Hutter, F. (2022b).
πBO: Augmenting acquisition functions with user beliefs for bayesian optimization.
In *International Conference on Learning Representations*.

[Kjolstad et al., 2017] Kjolstad, F., Kamil, S., Chou, S., Lugato, D., and Amarasinghe, S. (2017).
The tensor algebra compiler.
Proceedings of the ACM on Programming Languages, 1(OOPSLA).

[Kushner, 1964] Kushner, H. J. (1964).
A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise.
Journal of Basic Engineering, 86(1):97–106.

References (cont.)

[Li et al., 2020] Li, C., Gupta, S., Rana, S., Nguyen, V., Robles-Kelly, A., and Venkatesh, S. (2020).
Incorporating expert prior knowledge into experimental design via posterior sampling.
arXiv preprint arXiv:2002.11256.

[Mallik et al., 2022] Mallik, N., Hvarfner, C., Stoll, D., Bergman, E., Janowski, M., Lindauer, M., Nardi, L., and Hutter, F. (2022).
Priorband: Hyperband + human expert knowledge.

[Mayr et al., 2022a] Mayr, M., Ahmad, F., Chatzilygeroudis, K., Nardi, L., and Krueger, V. (2022a).
Skill-based multi-objective reinforcement learning of industrial robot tasks with planning and knowledge integration.
arXiv preprint arXiv:2203.10033.

[Mayr et al., 2021] Mayr, M., Chatzilygeroudis, K., Ahmad, F., Nardi, L., and Krueger, V. (2021).
Learning of parameters in behavior trees for movement skills.
In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7572–7579. IEEE.

[Mayr et al., 2022b] Mayr, M., Hvarfner, C., Chatzilygeroudis, K., Nardi, L., and Krueger, V. (2022b).
Learning skill-based industrial robot tasks with user priors.
arXiv preprint arXiv:2208.01605.

[Mockus et al., 1978] Mockus, J., Tiesis, V., and Zilinskas, A. (1978).
The application of Bayesian methods for seeking the extremum, volume 2, pages 117–129.

[Nardi et al., 2019] Nardi, L., Koeplinger, D., and Olukotun, K. (2019).
Practical design space exploration.
In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE.

References (cont.)

[Osborne et al., 2009] Osborne, M. A., Garnett, R., and Roberts, S. J. (2009).
Gaussian processes for global optimization.
In *3rd international conference on learning and intelligent optimization (LION3)*, volume 2009.

[Papenmeier et al., 2022] Papenmeier, L., Poloczek, M., and Nardi, L. (2022).
Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces.
In *Advances in Neural Information Processing Systems 35, NeurIPS 2022*.

[Paria et al., 2018] Paria, B., Kandasamy, K., and Póczos, B. (2018).
A flexible multi-objective bayesian optimization approach using random scalarizations.
CoRR, abs/1805.12168.

[Ramachandran et al., 2020] Ramachandran, A., Gupta, S., Rana, S., Li, C., and Venkatesh, S. (2020).
Incorporating expert prior in bayesian optimisation via space warping.
Knowledge-Based Systems, 195:105663.

[Shahriari et al., 2016] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2016).
Taking the human out of the loop: A review of bayesian optimization.
Proceedings of the IEEE, 104(1):148–175.

[Snoek et al., 2012] Snoek, J., Larochelle, H., and Adams, R. P. (2012).
Practical bayesian optimization of machine learning algorithms.
In *Advances in neural information processing systems*, pages 2951–2959.

[Souza et al., 2020a] Souza, A., Nardi, L., Oliveira, L. B., Olukotun, K., Lindauer, M., and Hutter, F. (2020a).
Prior-guided bayesian optimization.

[Souza et al., 2020b] Souza, A., Nardi, L., Oliveira, L. B., Olukotun, K., Lindauer, M., and Hutter, F. (2020b).
Prior-guided bayesian optimization.
arXiv preprint arXiv:2006.14608.

References (cont.)

[Springenberg et al., 2016] Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. (2016).
Bayesian optimization with robust bayesian neural networks.
In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 4134–4142. Curran Associates, Inc.

[Srinivas et al., 2010] Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010).
Gaussian process optimization in the bandit setting: No regret and experimental design.
In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 1015–1022, USA. Omnipress.

[Wang et al., 2016] Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Feitas, N. (2016).
Bayesian optimization in a billion dimensions via random embeddings.
Journal of Artificial Intelligence Research, 55:361–387.

[Wang and Jegelka, 2017] Wang, Z. and Jegelka, S. (2017).
Max-value entropy search for efficient bayesian optimization.
In *International Conference on Machine Learning*, pages 3627–3635. PMLR.