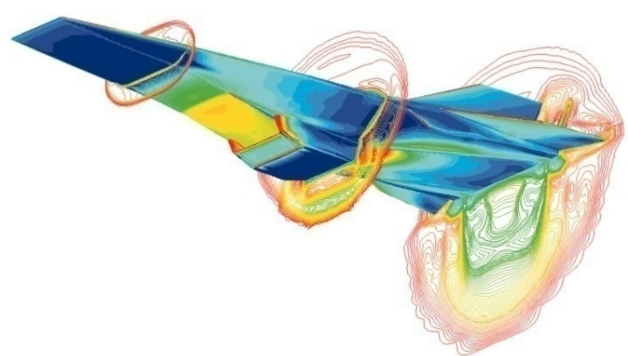
Engineering in the Age of Machine Learning

Barcelona, 30 October 2015
Roger Frigola



Machine Learning

- Create models based on data to make inferences/predictions.
- Based on statistics and computing.
- Reality is noisy and uncertain.

Engineering

- Create realities that had never existed before.
- Based on physics.
- Reality is mostly deterministic.

My Background

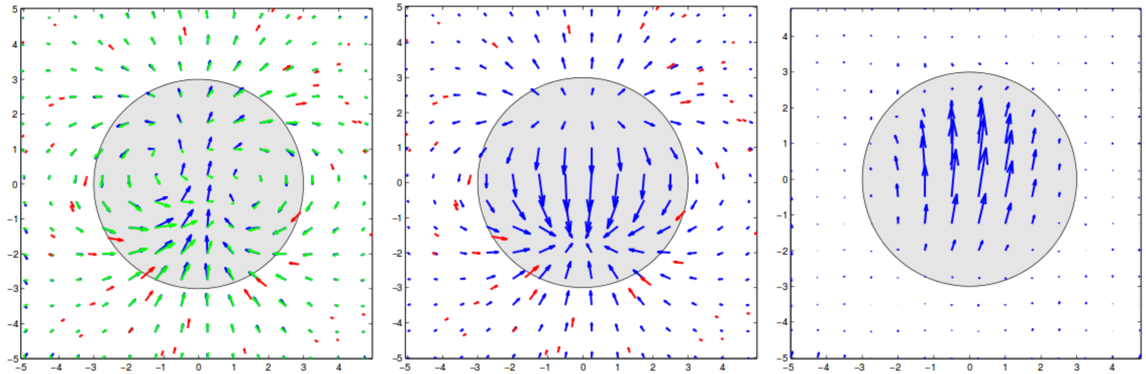
- Degrees in General and Aerospace Engineering
- AIRBUS (1.5 years) Fluid-Control-Structure Interaction
- McLaren F1 Team (4 years) Simulation, Modelling, Optimisation,
- PhD in Machine Learning
- Consulting at Ferrari F1, Red Bull F1, NZ America's Cup team

Outline

- Model-based Machine Learning
- Bayesian Inference
- Experiment Design and Optimisation
- The Future?

Model-Based Machine Learning

- Don't rely only on data. Use the knowledge we have about the world!

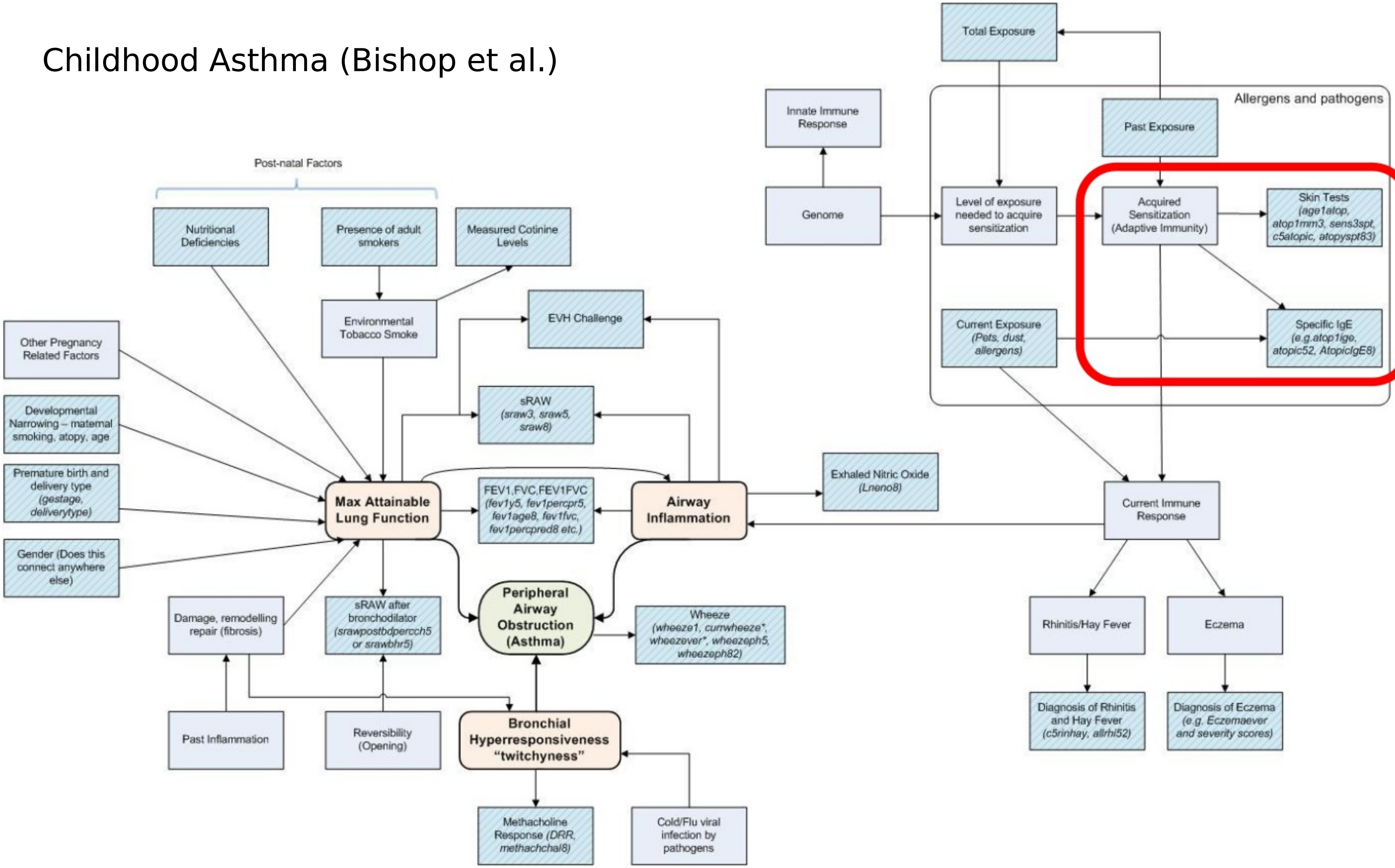


(a) Estimated **B**-field (b) Estimated **H**-field (c) Estimated **M**-field

Make customised assumptions!

Model-Based Machine Learning

Childhood Asthma (Bishop et al.)



Bayesian Inference for Engineering. Why?

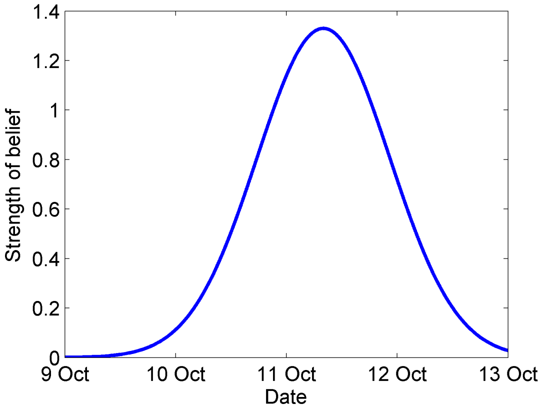
Introduction to Bayesian Modelling and Inference

Uses probability to *quantify uncertainty*.

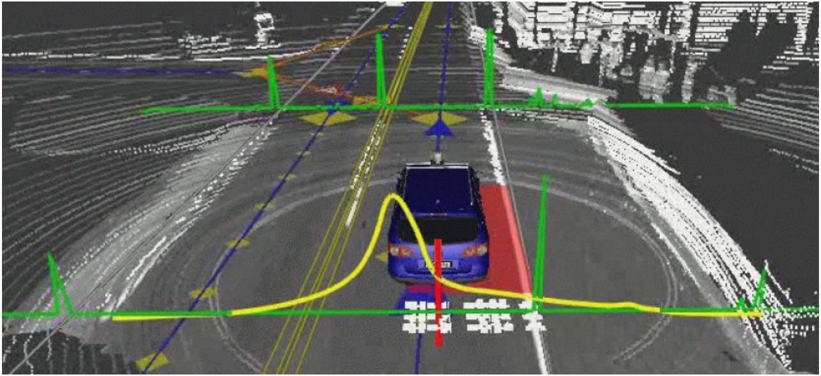
Related to information rather than repeated trials.

Uncertainty is subjective, it depends on what we have seen.

Subjective Uncertainty



Subjective Uncertainty



Stanford's self-driving car for the DARPA Urban Challenge (2007).

Bayesian Inference of Power and Drag

Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model

$$a_t = \frac{1}{mV_t} P - \frac{\rho V_t^2 S}{2m} C_d$$

e.g. with Gaussian noise

$$y_t = \mathcal{N}(a_t, \sigma^2)$$

$$\mathcal{D} = \{y_1, \ldots, y_N, V_1, \ldots, V_N\}$$

Bayesian Inference of Power and Drag

Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model

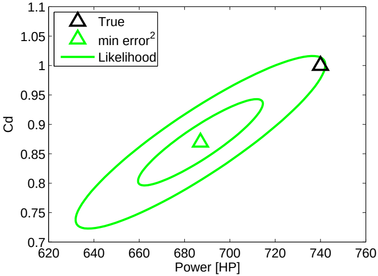
GOAL: find probability distribution of unknown parameters given data

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)\, p(\theta)}{p(\mathcal{D})}$$

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\, p(\theta)$$

Bayesian Inference of Power and Drag

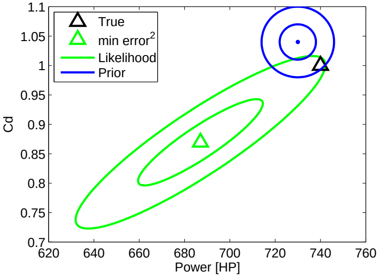
Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model



$$p(\mathcal{D} \mid \theta)$$

Bayesian Inference of Power and Drag

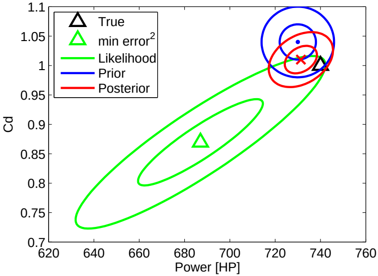
Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model



$p(\theta)$

Bayesian Inference of Power and Drag

Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model



$$p(\theta \mid \mathcal{D})$$

Bayesian Inference

One could say that we have used the prior as a regulariser to solve

$$\theta^* = \arg\min_\theta \; L(\theta, \mathcal{D}) + J(\theta)$$

But, we were simply looking for the posterior: $p(\theta \mid \mathcal{D})$.

No optimisation!

The posterior represents our uncertainty and tells us how to average different models.

Bayesian Inference

What is the outcome of Bayesian inference?

Thomas' indoor localisation example.

Posterior over parameters $\rightarrow$ posterior over identified systems.

In fact, we can find posteriors over many different kinds of objects: functions, genetic trees, English language sentences, etc.

Bayesian Inference: Making Predictions

The posterior represents our uncertainty over the parameters.

Any prediction can be found by *averaging* over the posterior

$$\begin{aligned} p(\mathrm{LapTime} \mid \mathcal{D}) &= \int p(\mathrm{LapTime}, \theta \mid \mathcal{D}) \, d\theta \\ &= \int p(\mathrm{LapTime} \mid \theta) \, p(\theta \mid \mathcal{D}) \, d\theta. \end{aligned}$$

Making Decisions Under Uncertainty

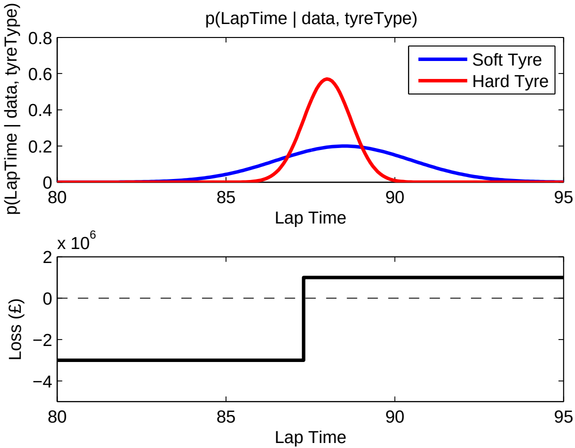
Bayesian inference provides probability distributions.

But, often, we can only take one action.

One solution: take the action that minimises the expected loss (aka risk) under the uncertainty provided by Bayesian inference.

$$a_{\mathrm{opt}} = \arg\min_{a} \int \mathrm{Loss}(a, \theta)\, p(\theta \mid \mathcal{D})\, d\theta$$

Making Decisions Under Uncertainty



Expected loss for hard tyre: $\$3.65 \cdot 10^5$

Expected loss for soft tyre: $\$-0.98 \cdot 10^5$

Over-fitting and Counting Parameters

Bayesian methods do not overfit because there is *no fitting!*

Inference is based on integration, i.e. averaging.

There is no statistical price to pay for adding more parameters.

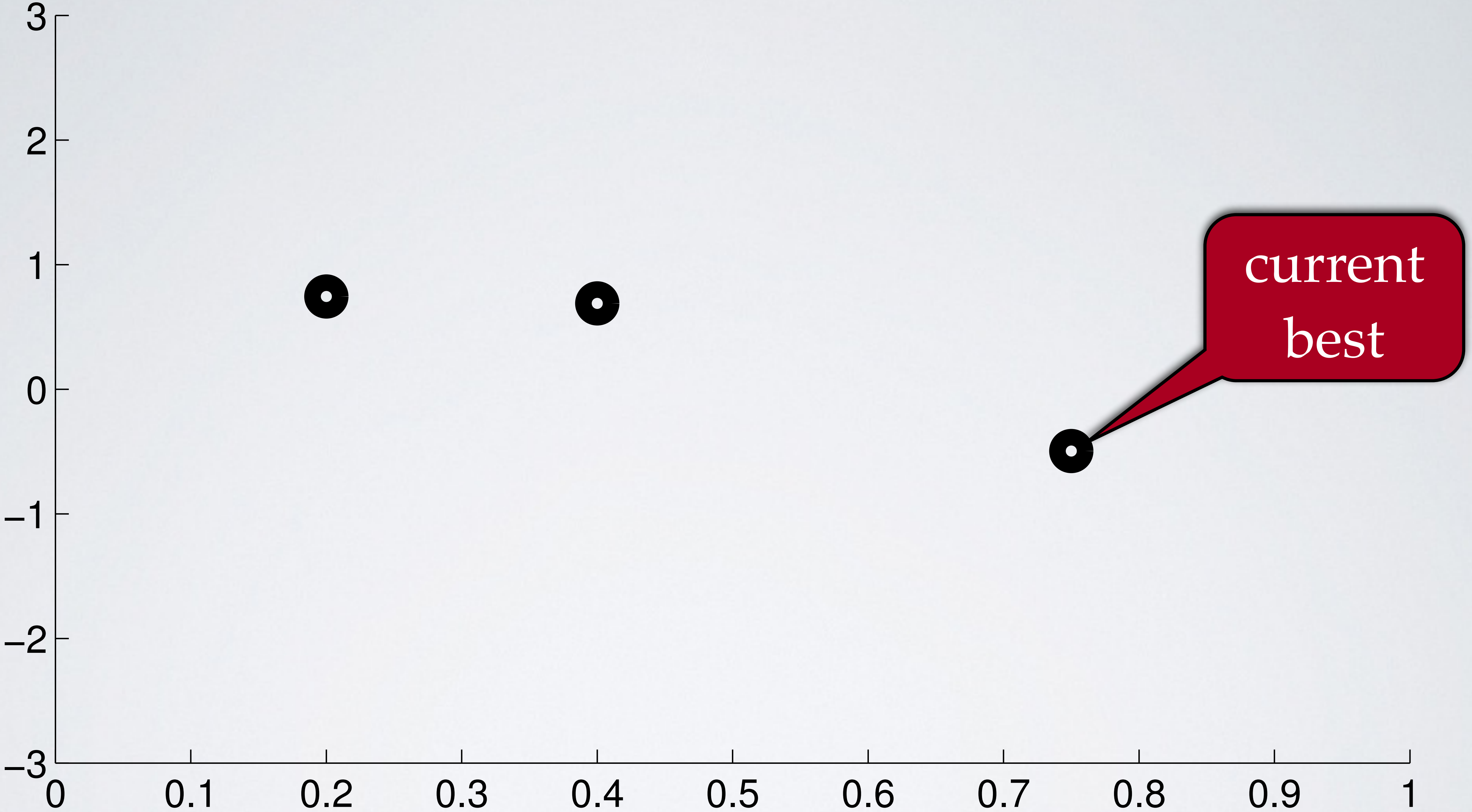
Nonidentifiability is not a problem when making predictions.

Experiment Design and Optimisation

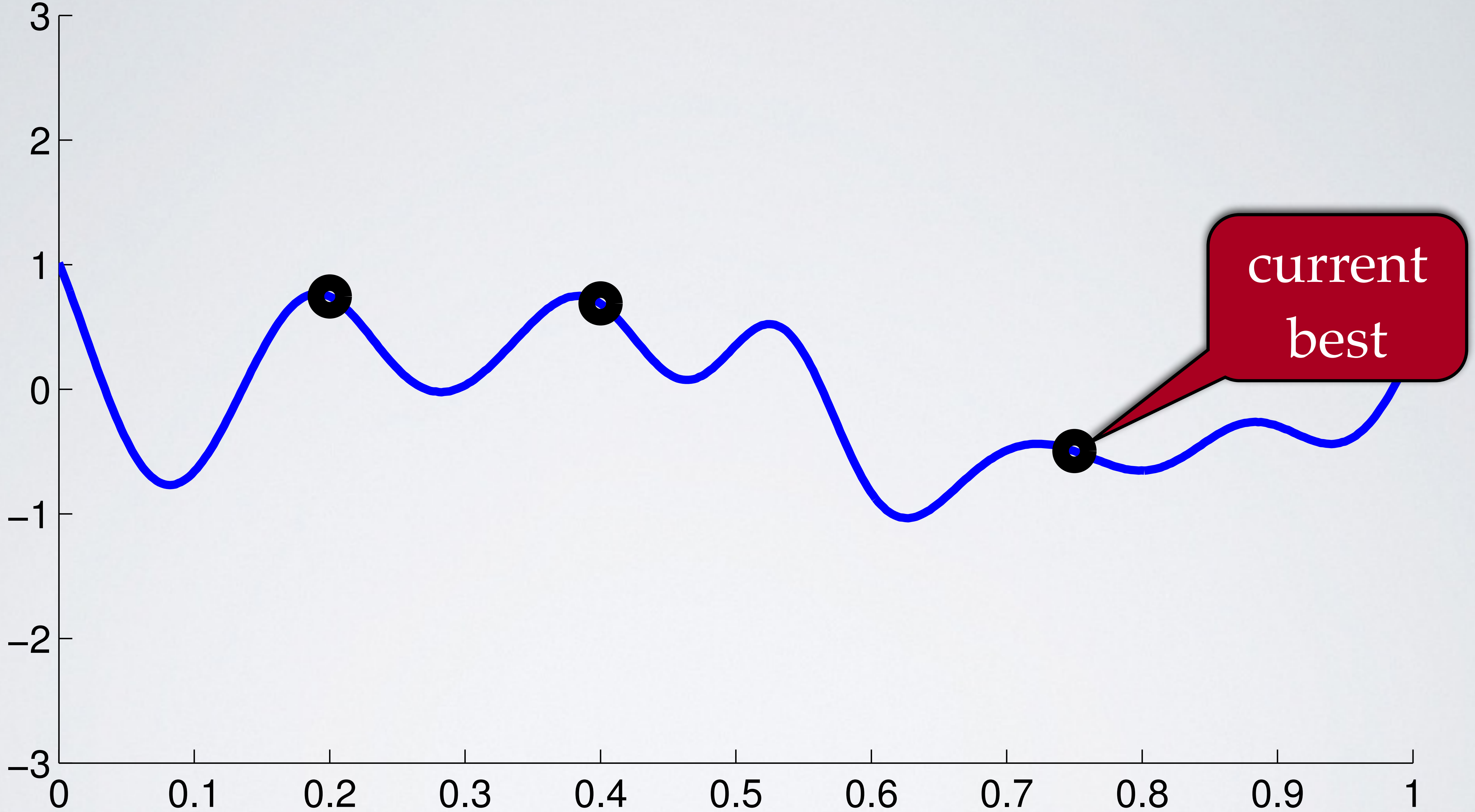
- In engineering we can run simulations or make prototypes.
- Which simulations to run? What prototypes to build?
- What is our goal?

- Following slides from Ryan Adams, A Tutorial on Bayesian Optimization for Machine Learning (2014).

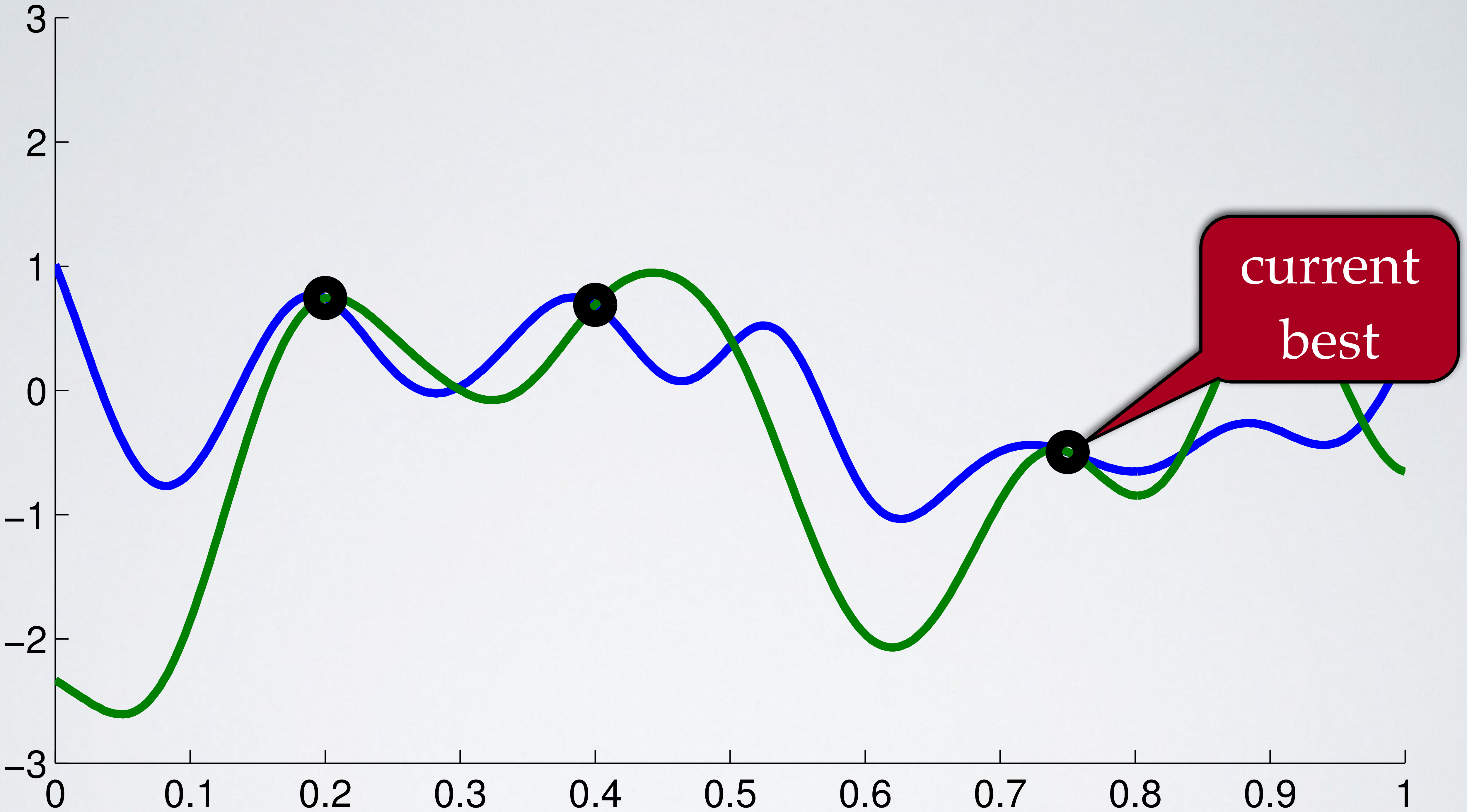
The Bayesian Optimization Idea



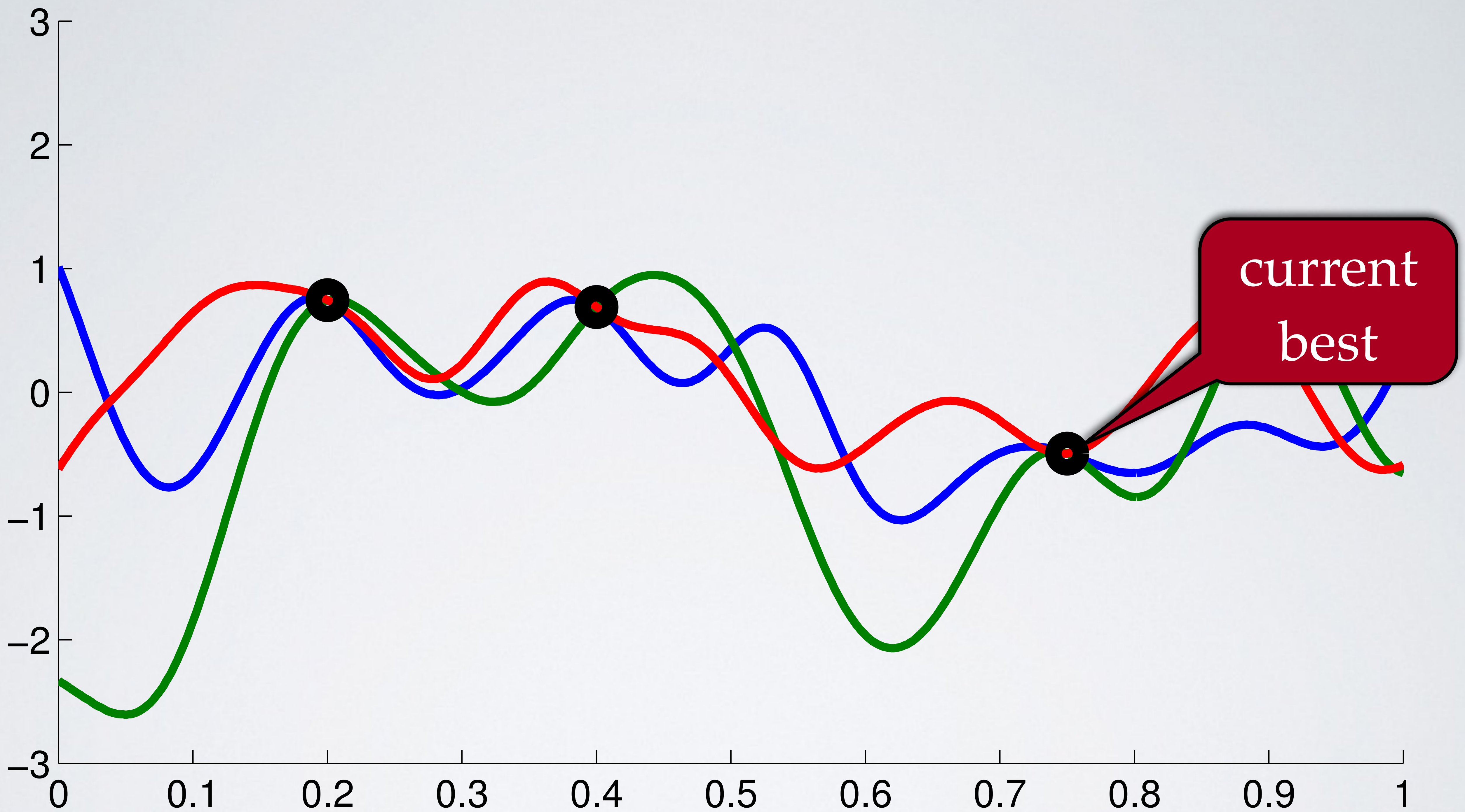
The Bayesian Optimization Idea



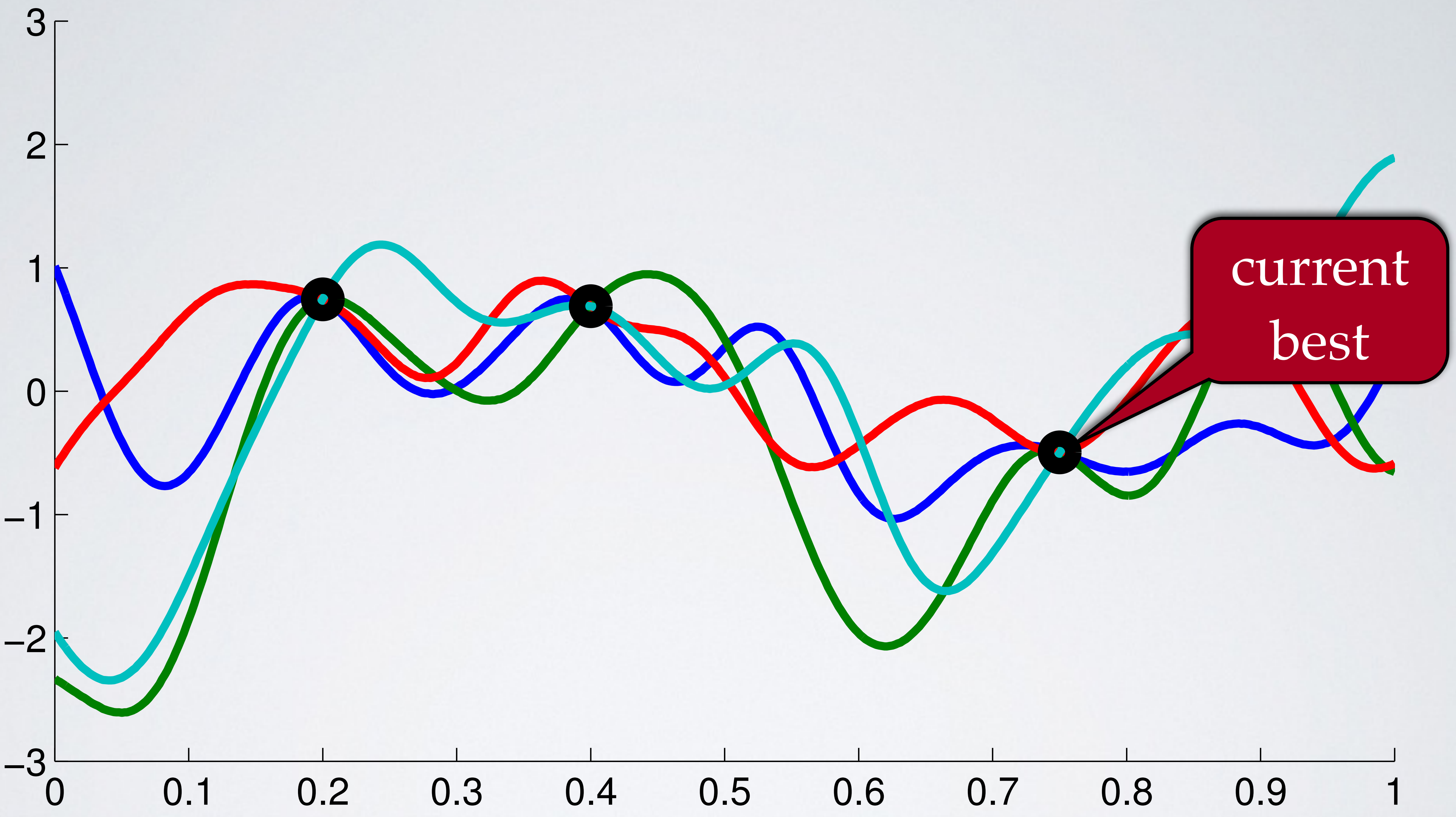
The Bayesian Optimization Idea



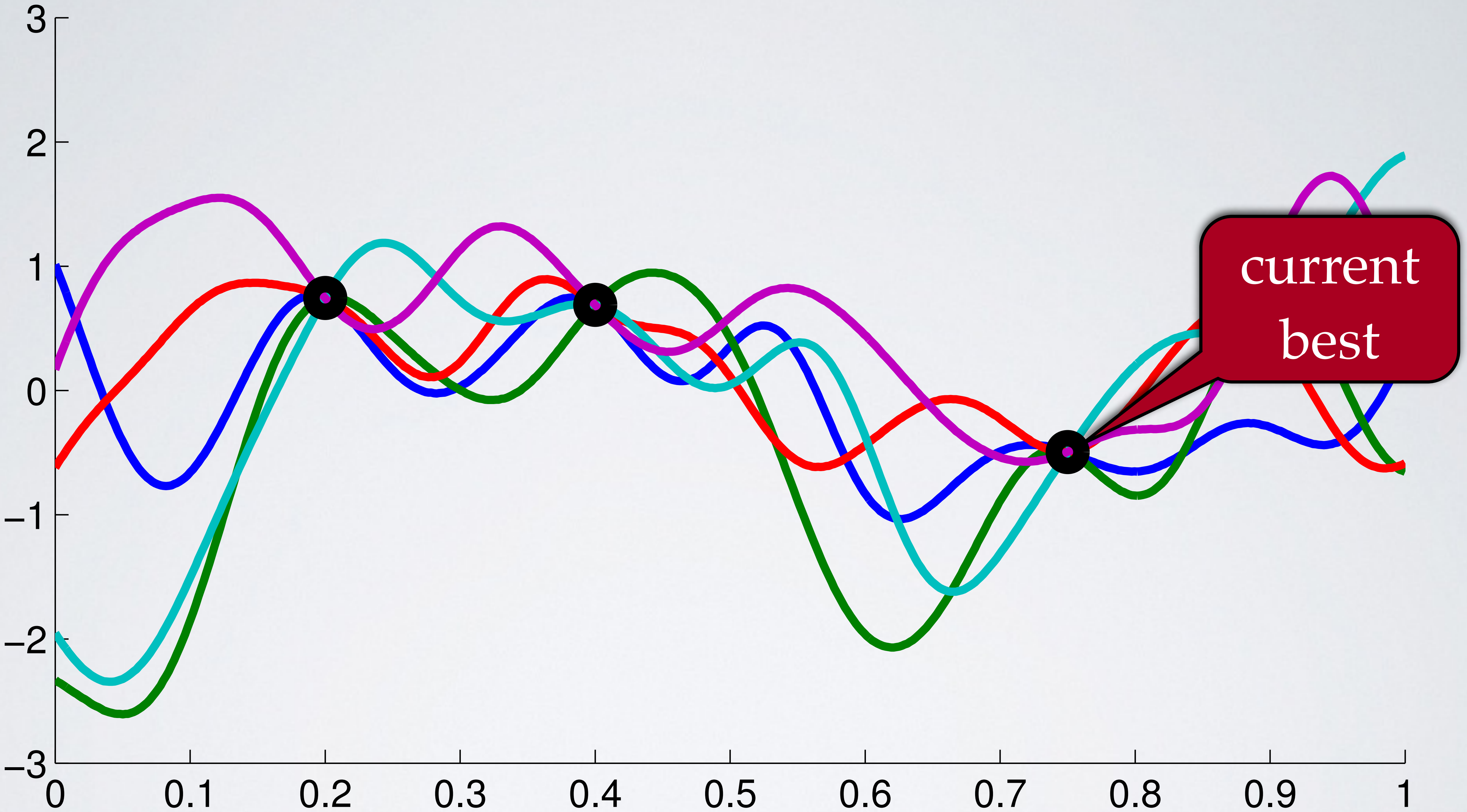
The Bayesian Optimization Idea



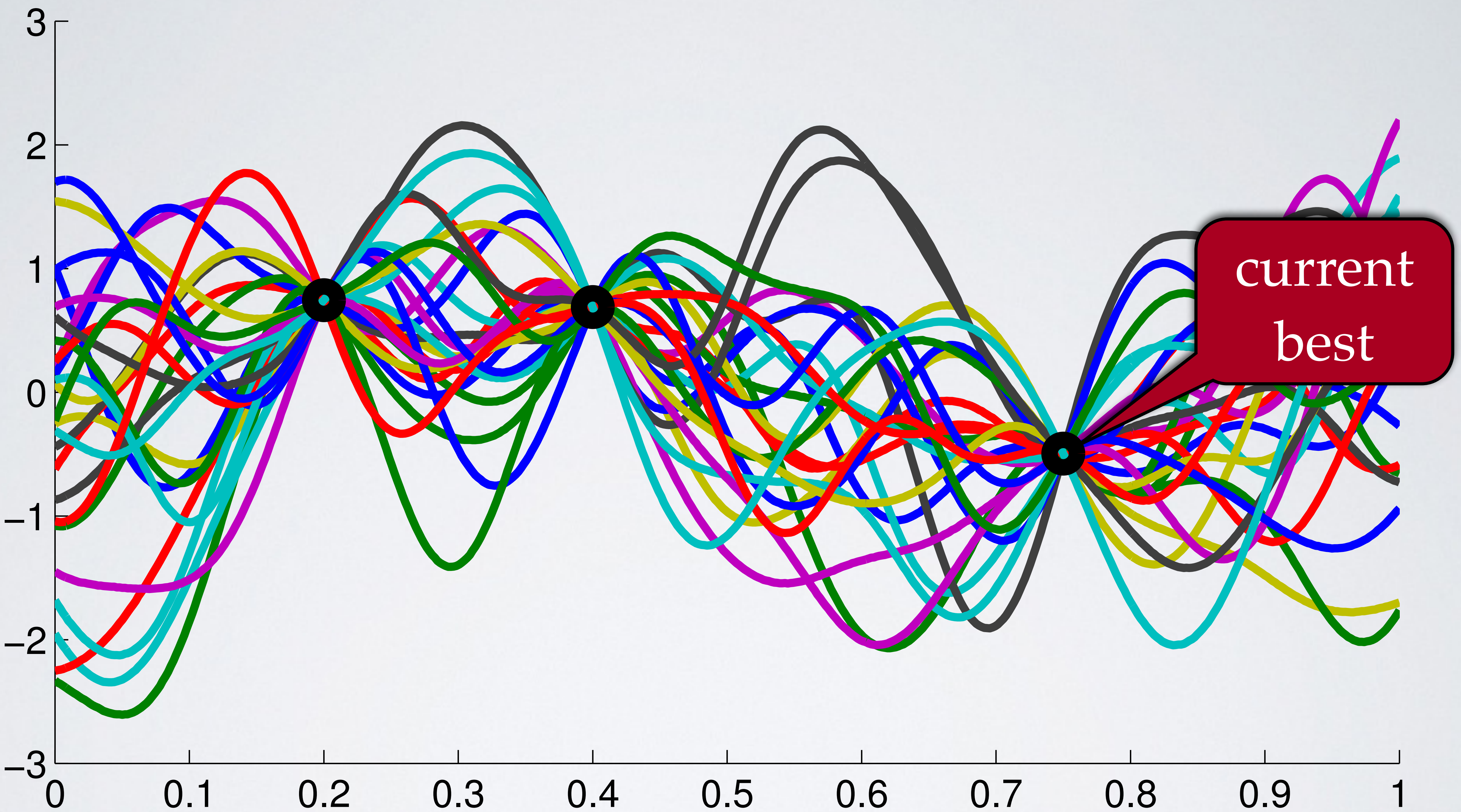
The Bayesian Optimization Idea



The Bayesian Optimization Idea

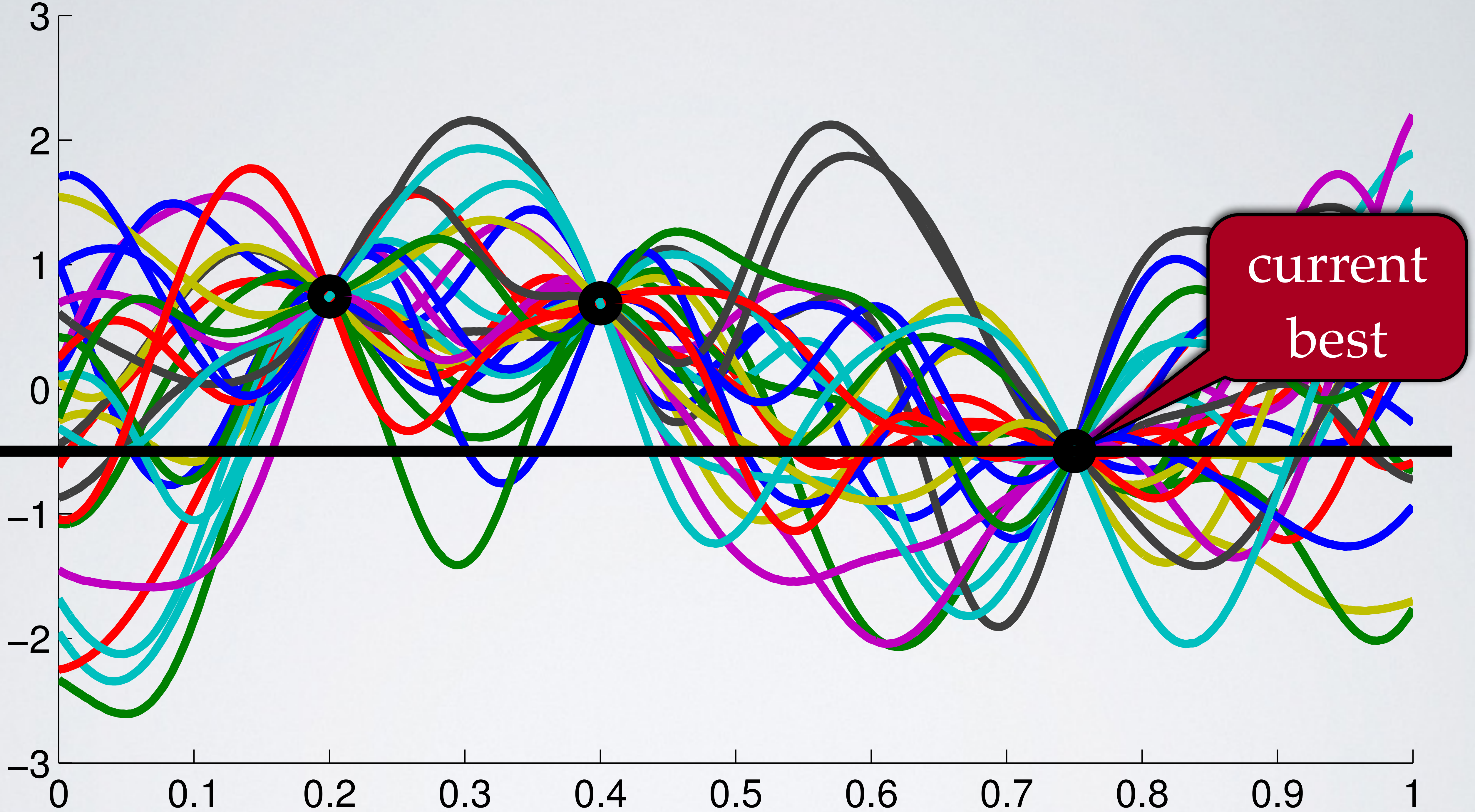


The Bayesian Optimization Idea



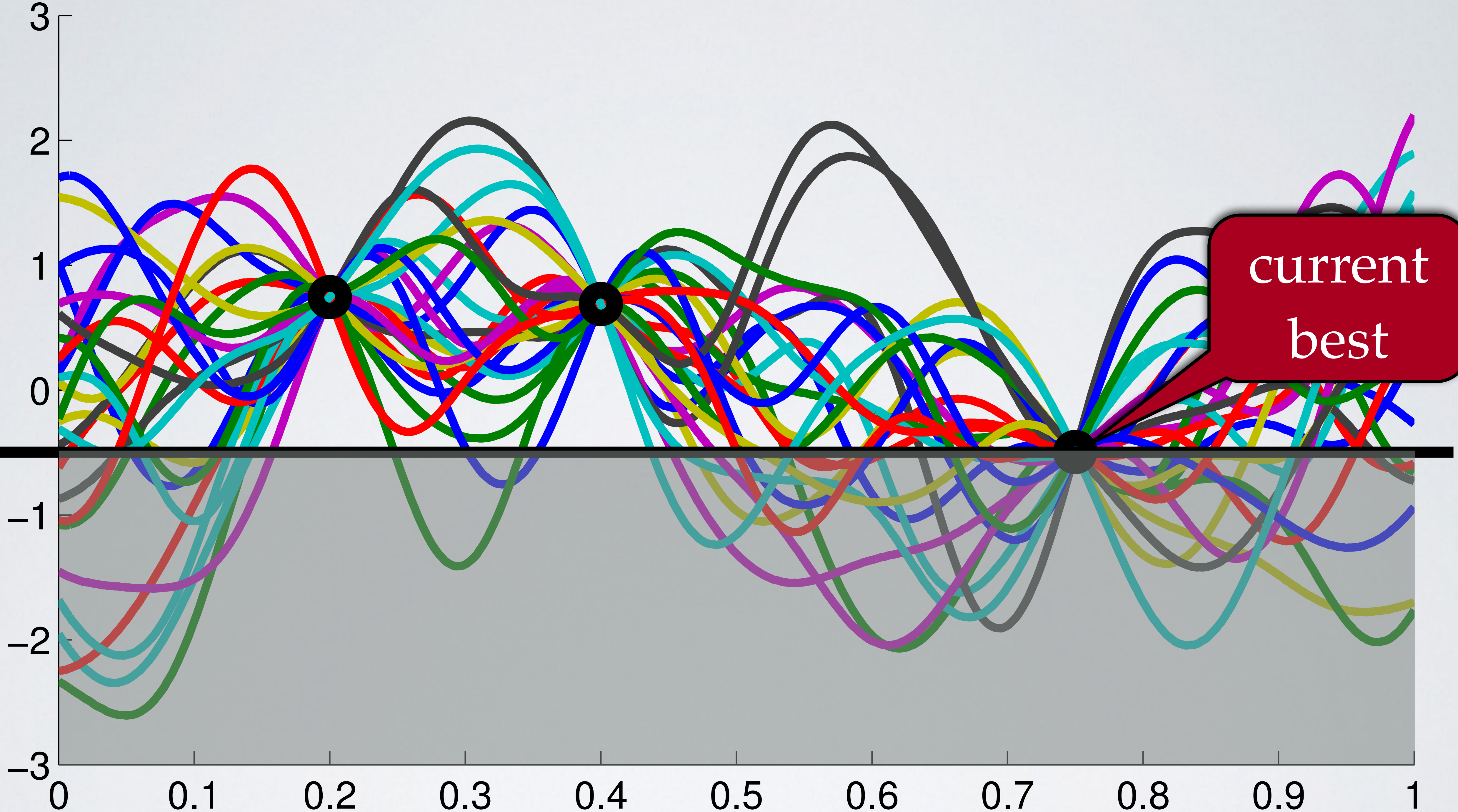
Where should we evaluate next
in order to improve the most?

The Bayesian Optimization Idea



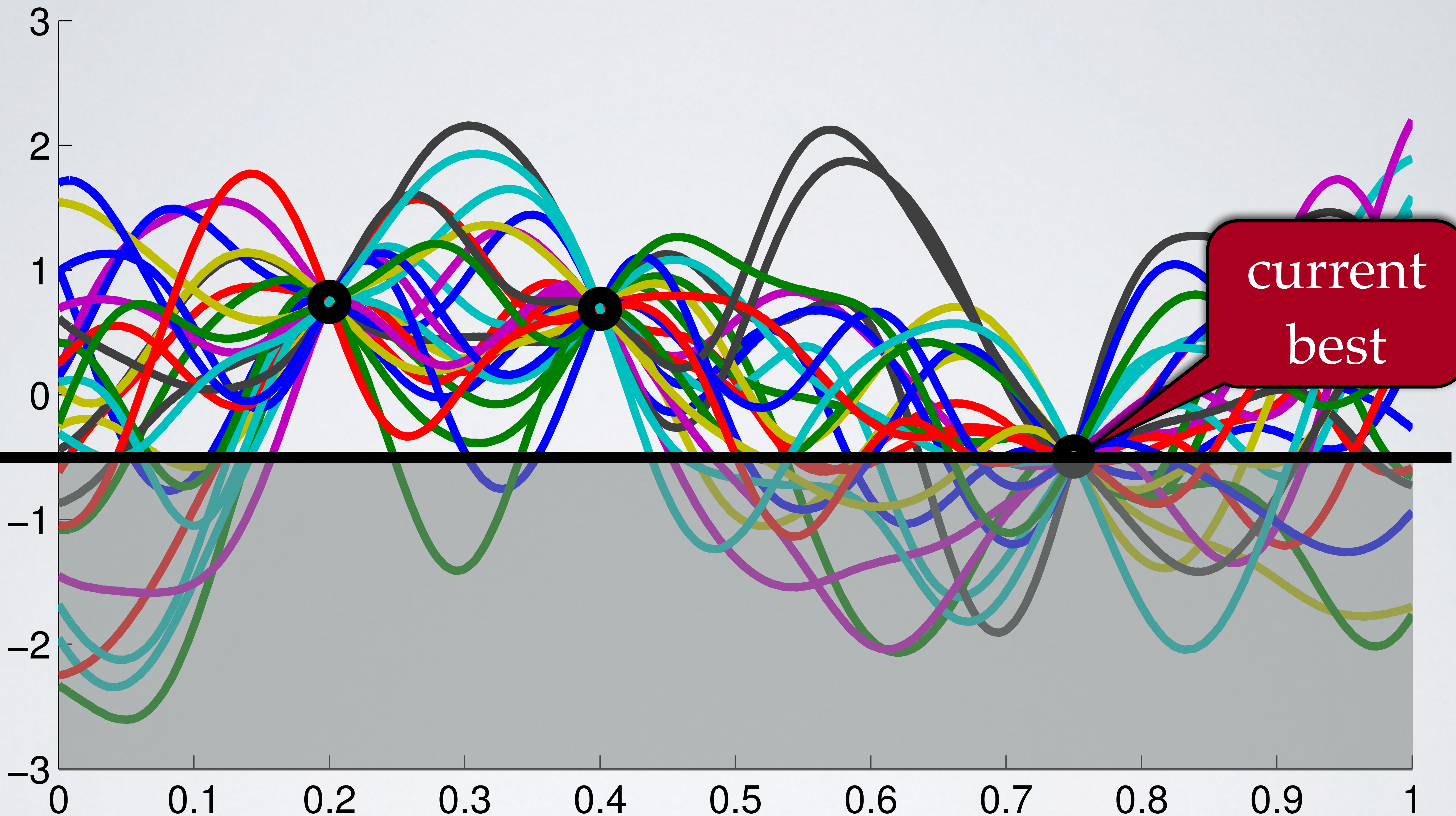
Where should we evaluate next
in order to improve the most?

The Bayesian Optimization Idea

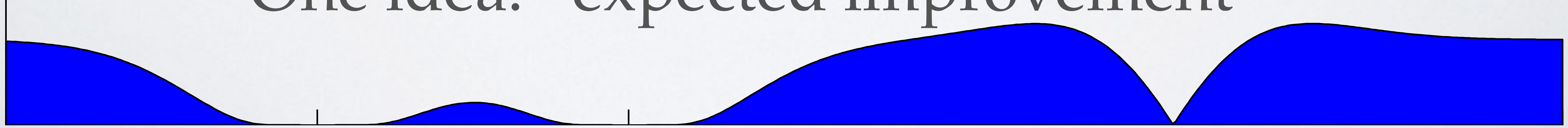


Where should we evaluate next
in order to improve the most?

The Bayesian Optimization Idea

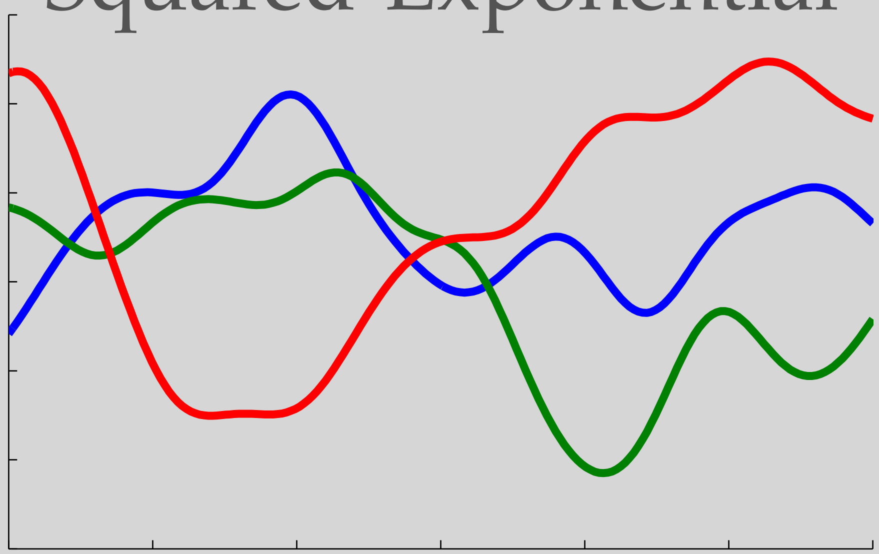


One idea: "expected improvement"



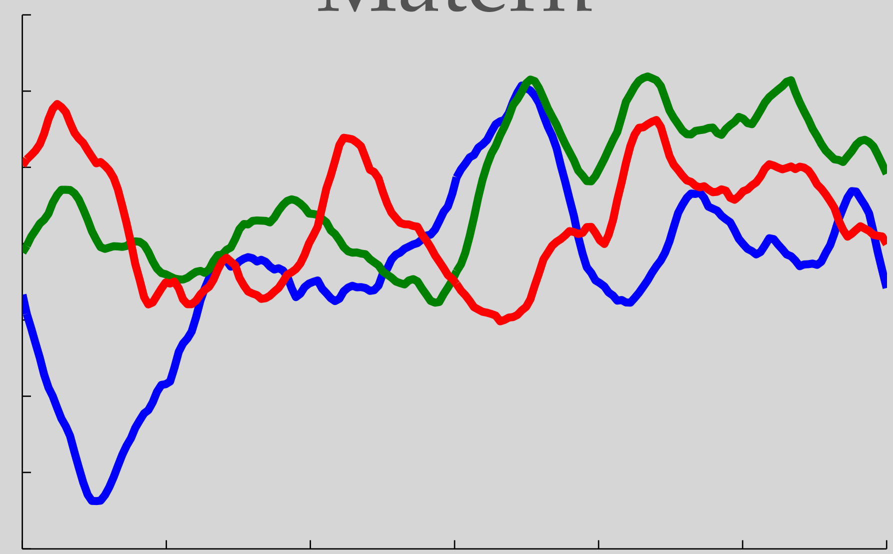
Examples of GP Covariances

Squared-Exponential



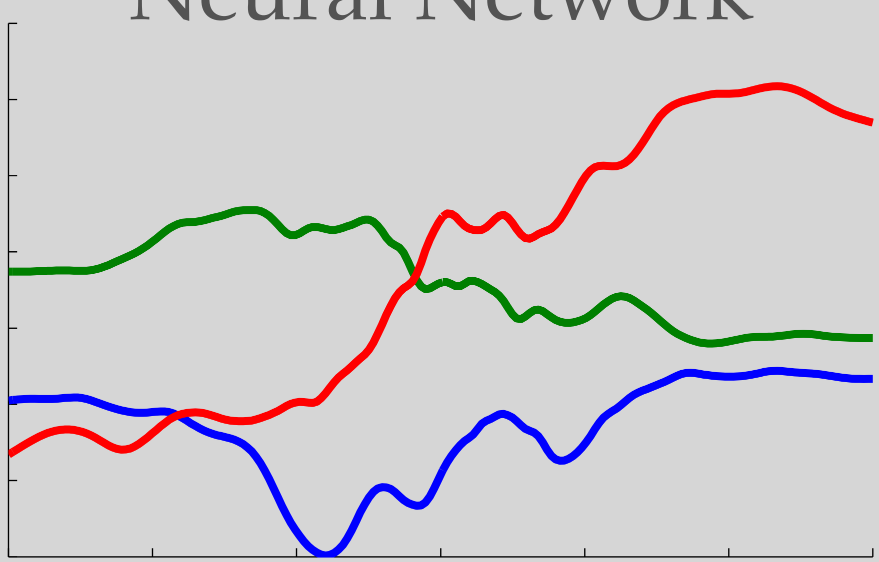
$$C(x,x') = \alpha \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_d - x'_d}{\ell_d}\right)^2\right\}$$

Matérn



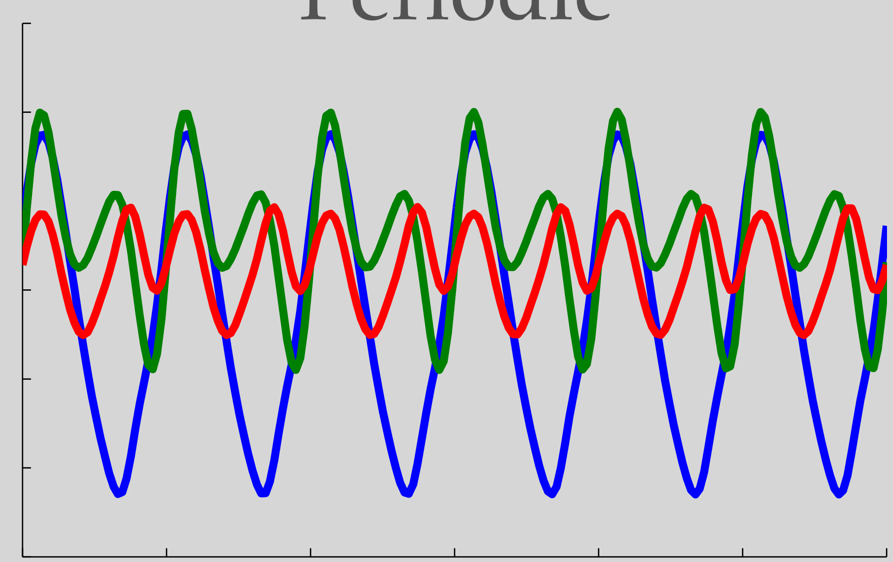
$$C(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\,r}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}\,r}{\ell}\right)$$

"Neural Network"



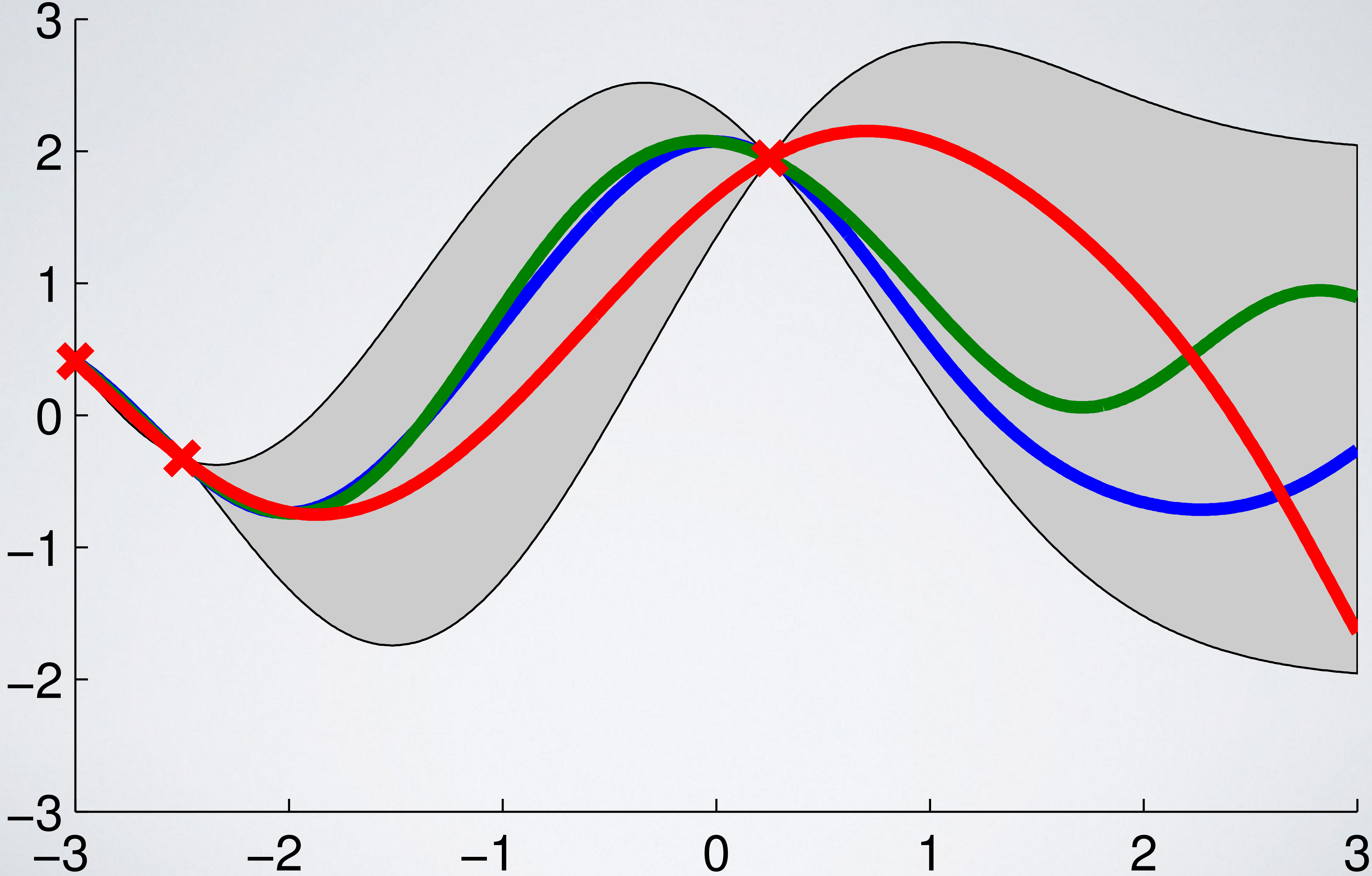
$$C(x,x') = \frac{2}{\pi}\sin^{-1}\left\{\frac{2x^\top\Sigma x'}{\sqrt{(1+2x^\top\Sigma x)(1+2x'^\top\Sigma x')}}\right\}$$

Periodic

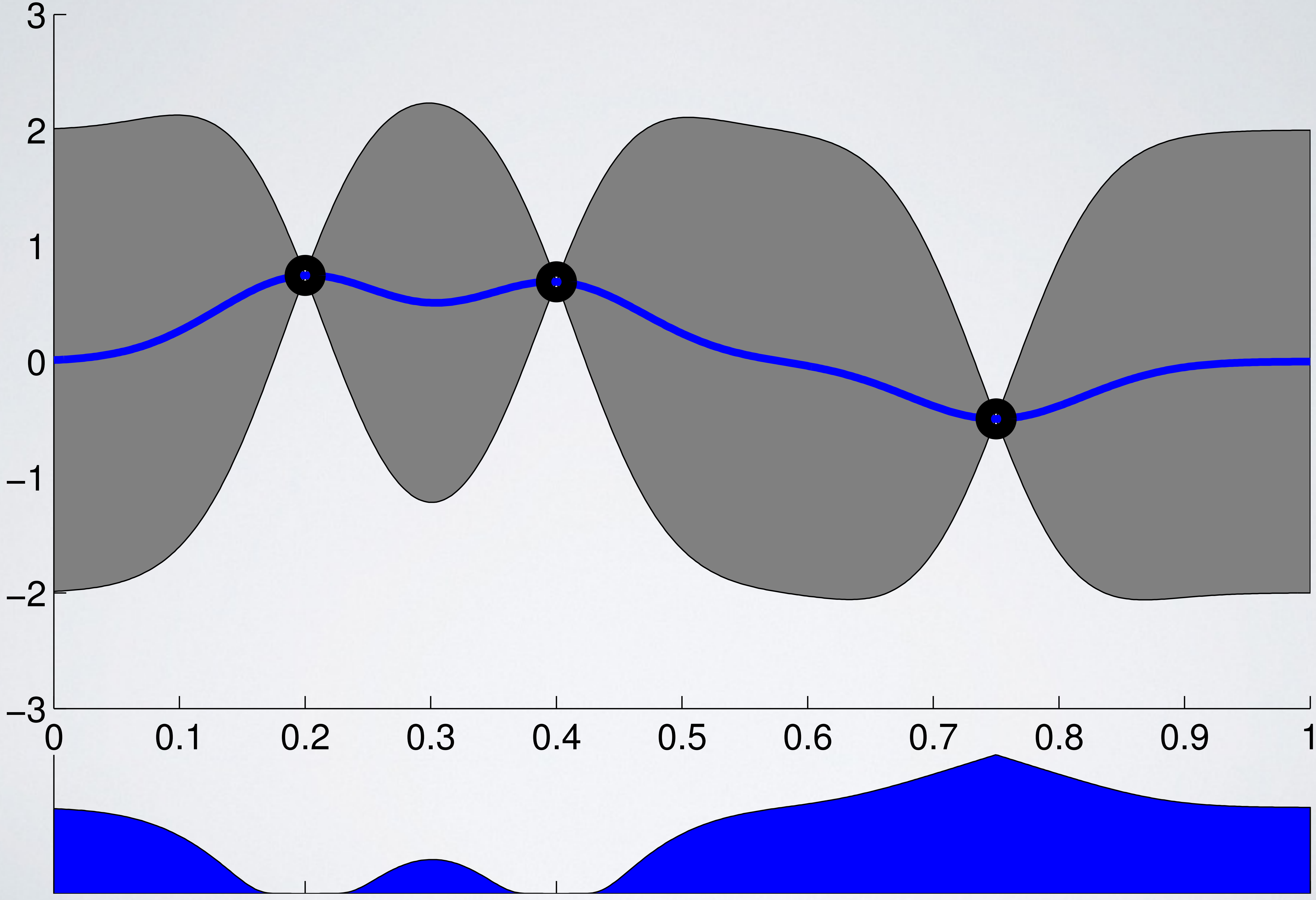


$$C(x,x') = \exp\left\{-\frac{2\sin^2\left(\frac{1}{2}(x-x')\right)}{\ell^2}\right\}$$

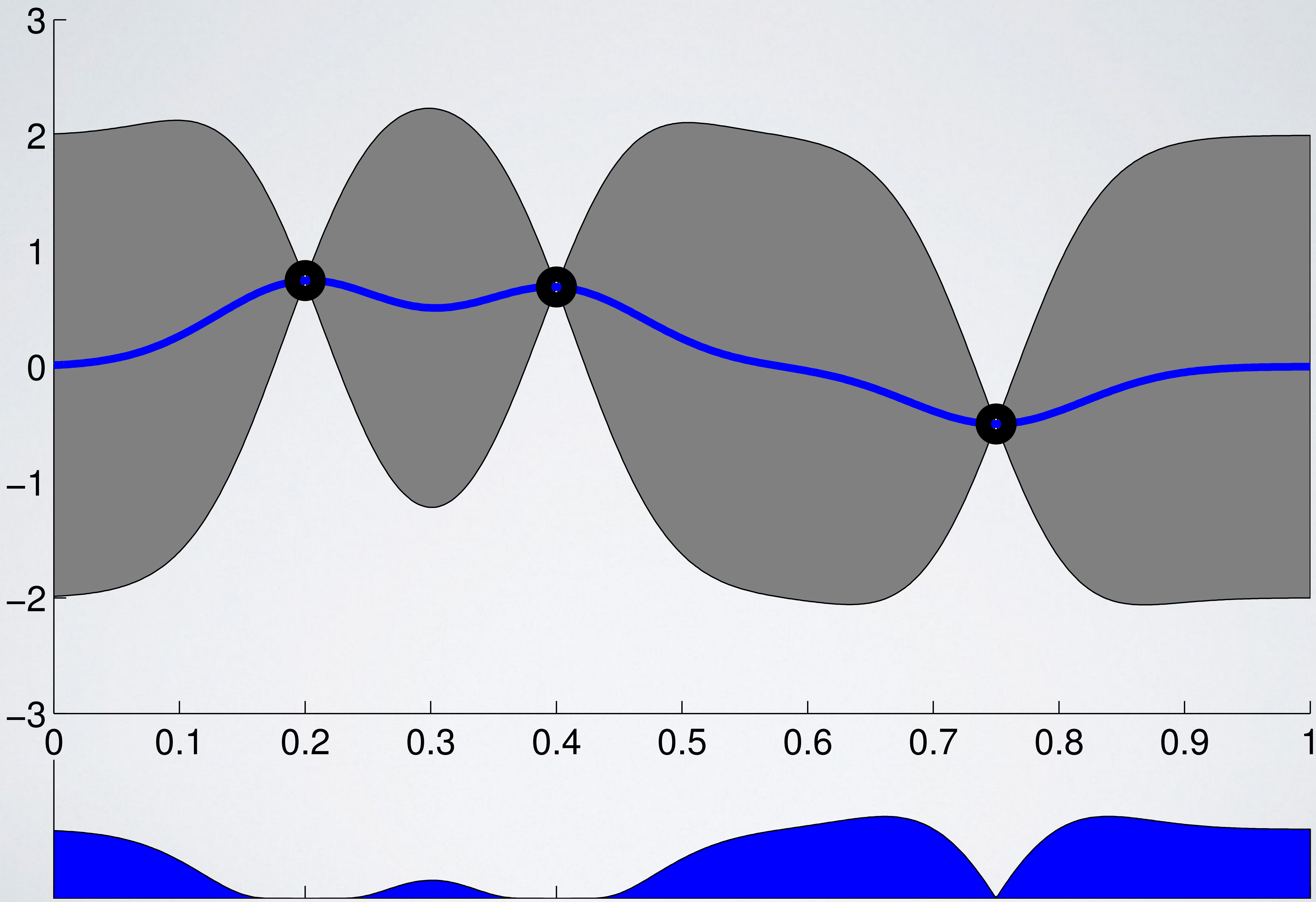
GPs Provide Closed-Form Predictions



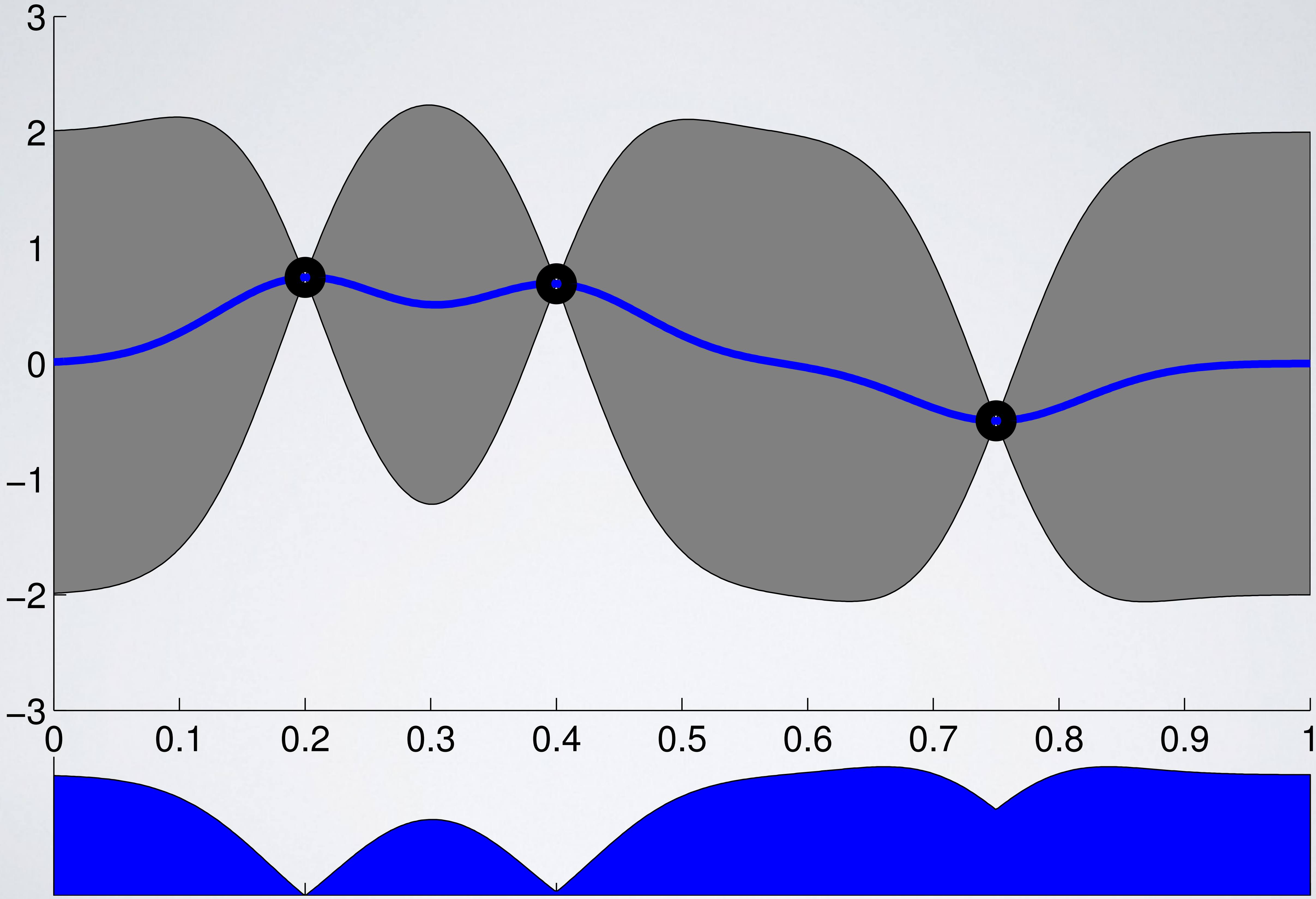
Probability of Improvement



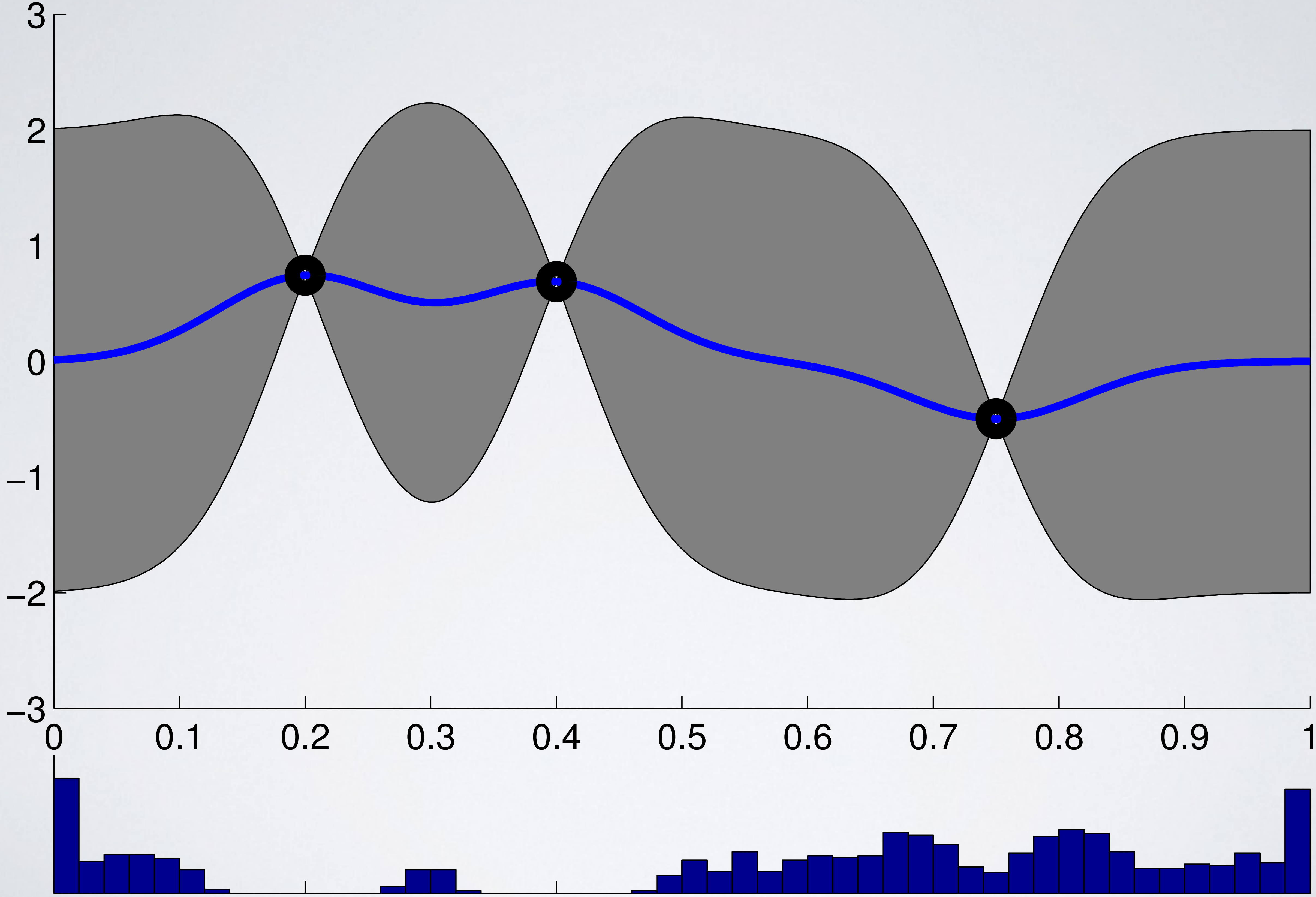
Expected Improvement



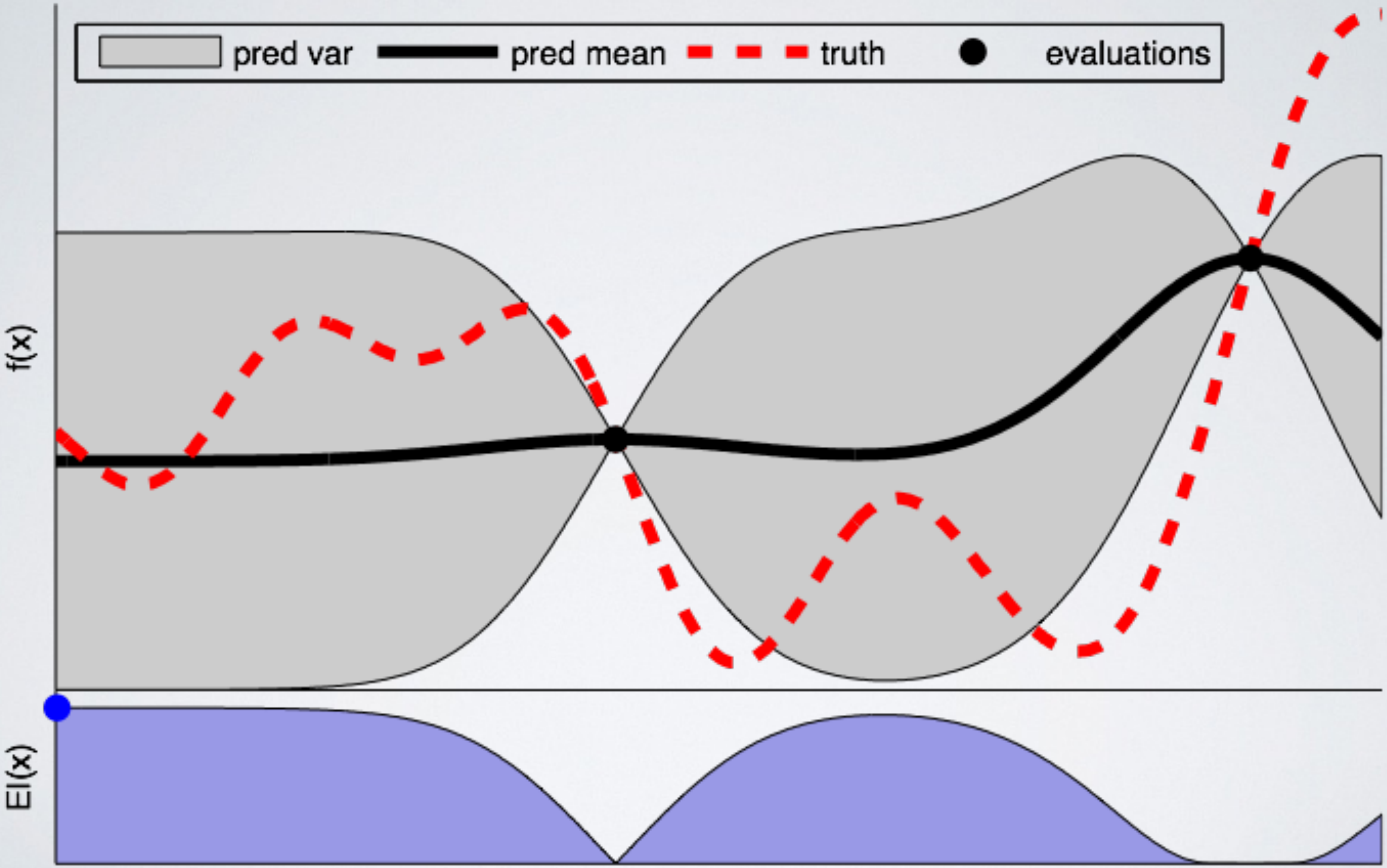
GP Upper (Lower) Confidence Bound



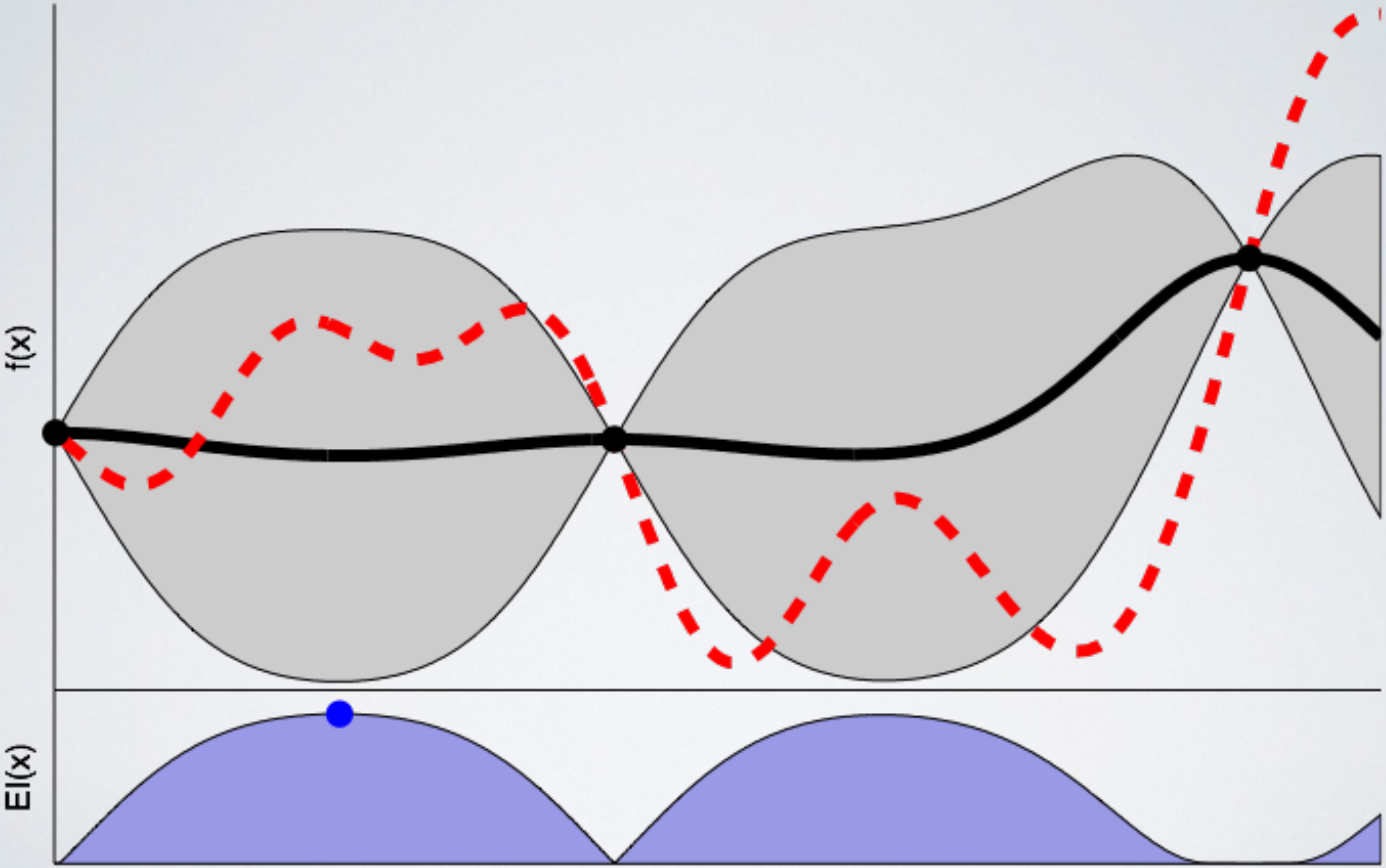
Distribution Over Minimum (Entropy Search)



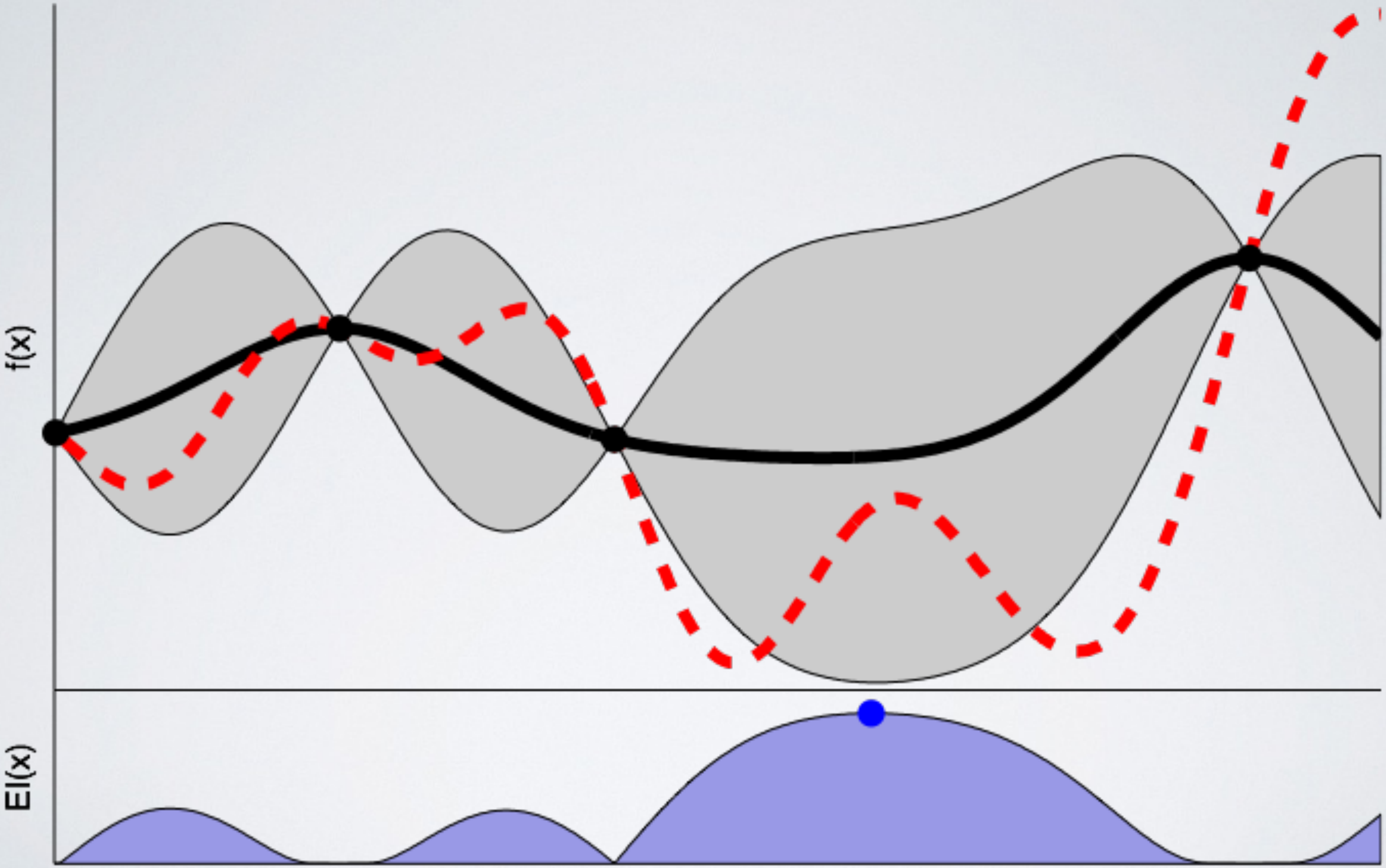
Illustrating Bayesian Optimization



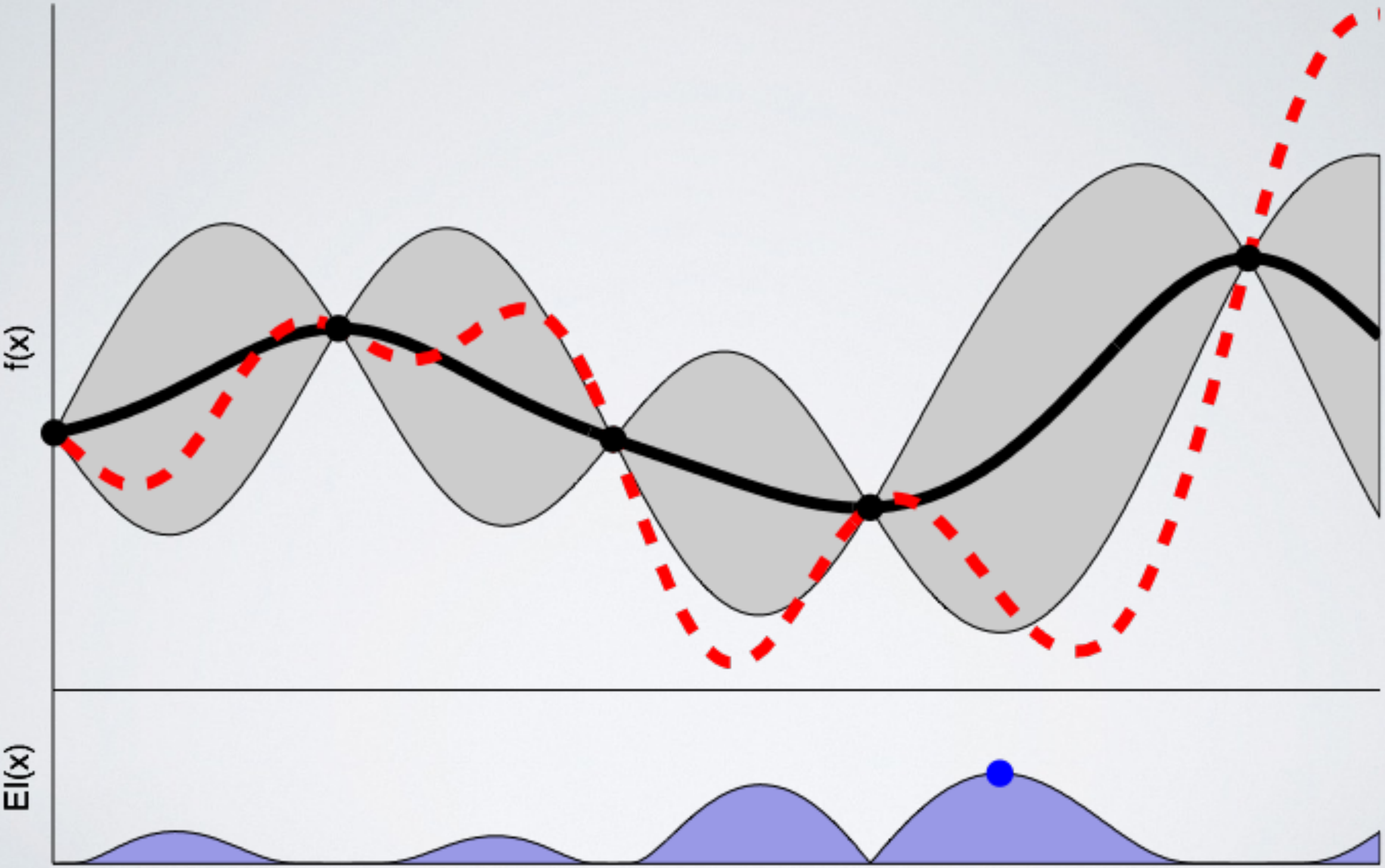
Illustrating Bayesian Optimization



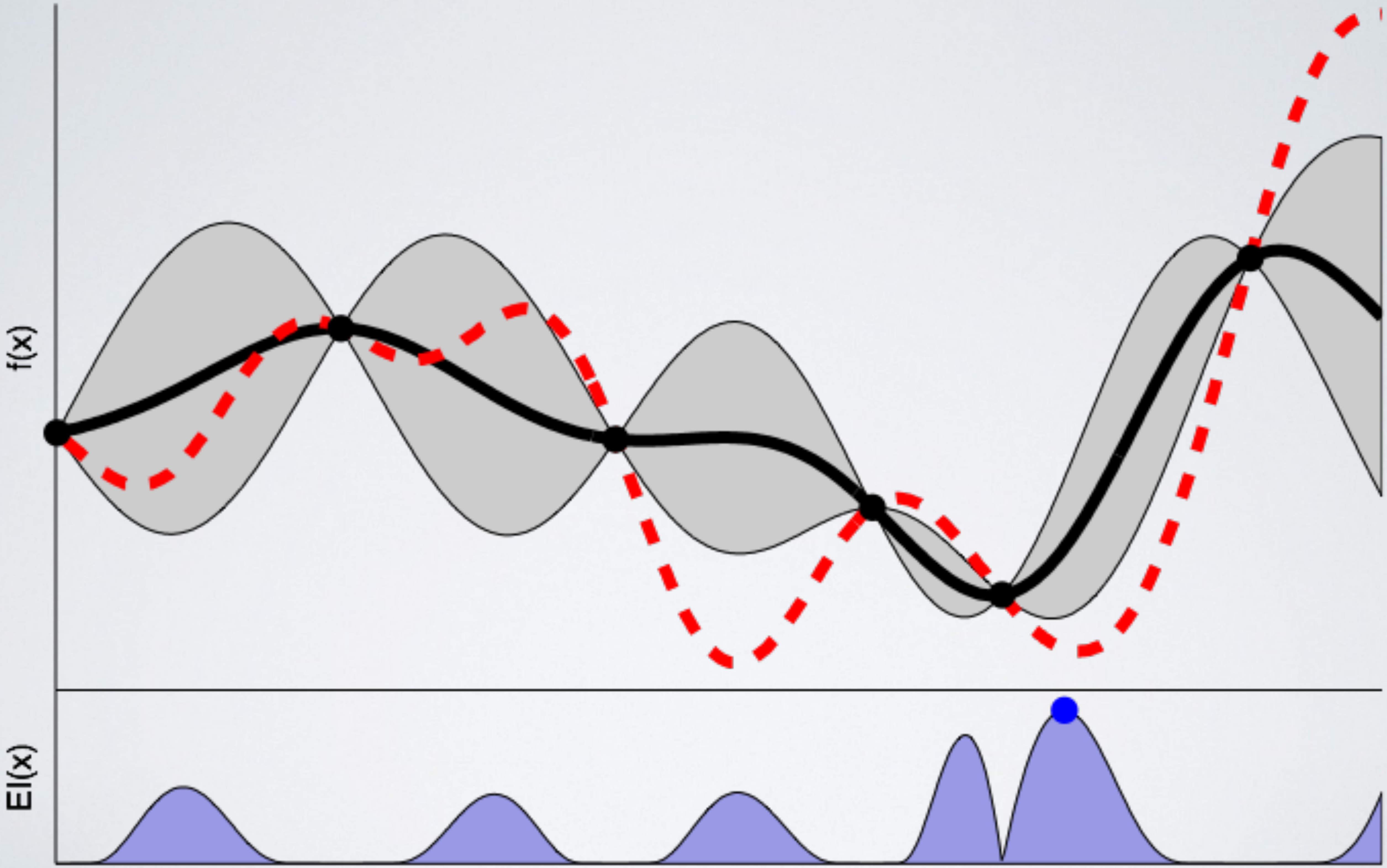
Illustrating Bayesian Optimization



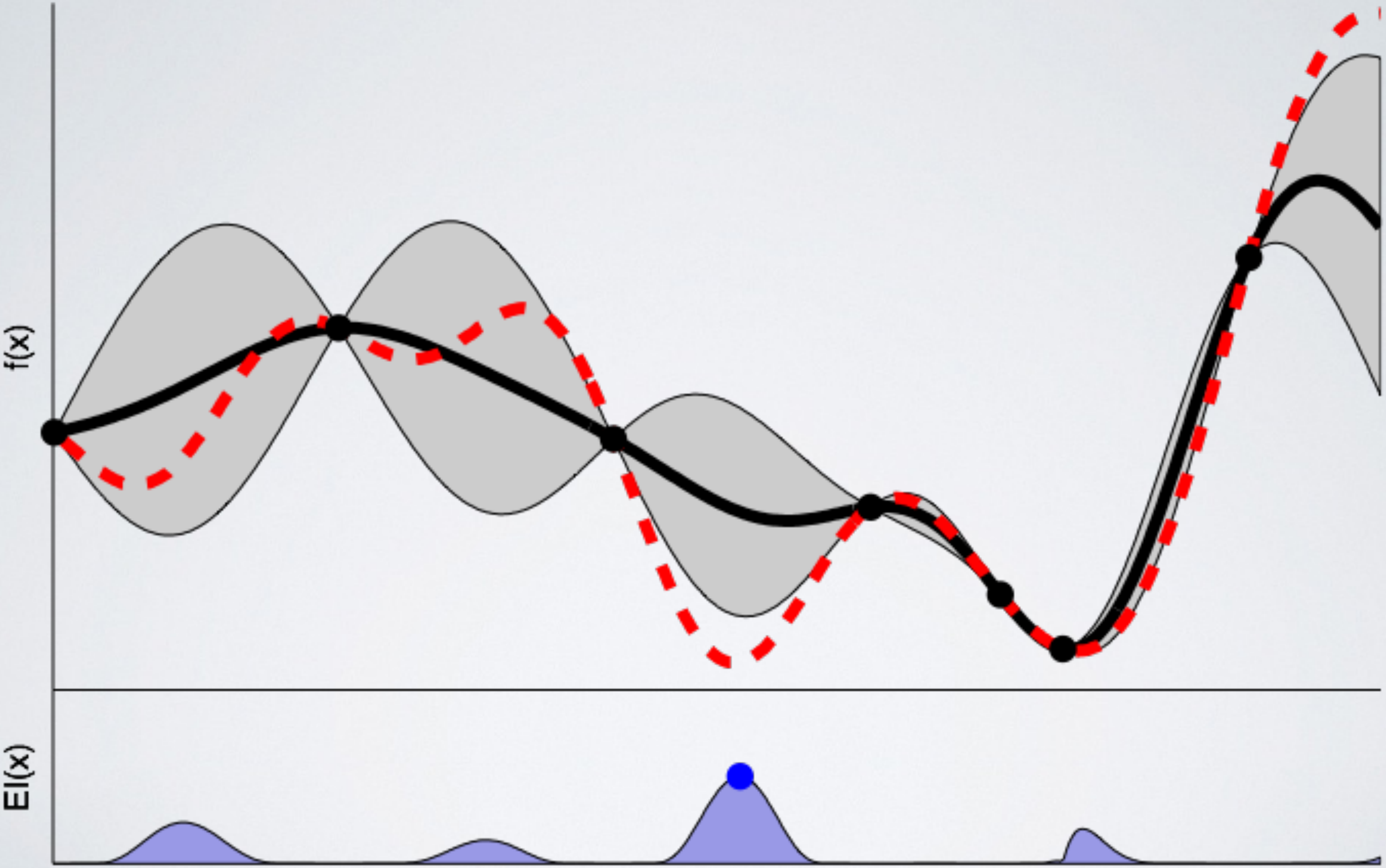
Illustrating Bayesian Optimization



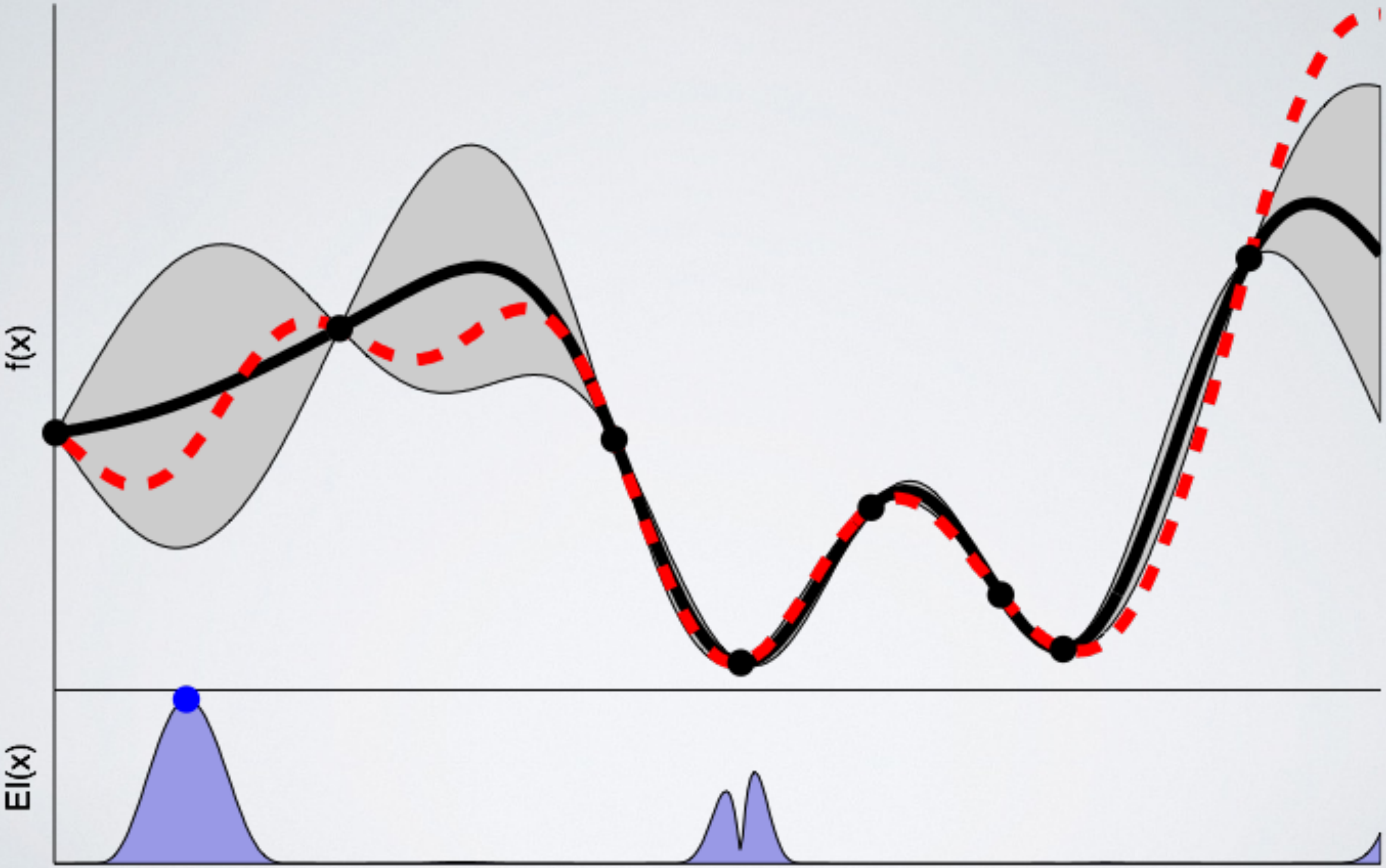
Illustrating Bayesian Optimization



Illustrating Bayesian Optimization



Illustrating Bayesian Optimization

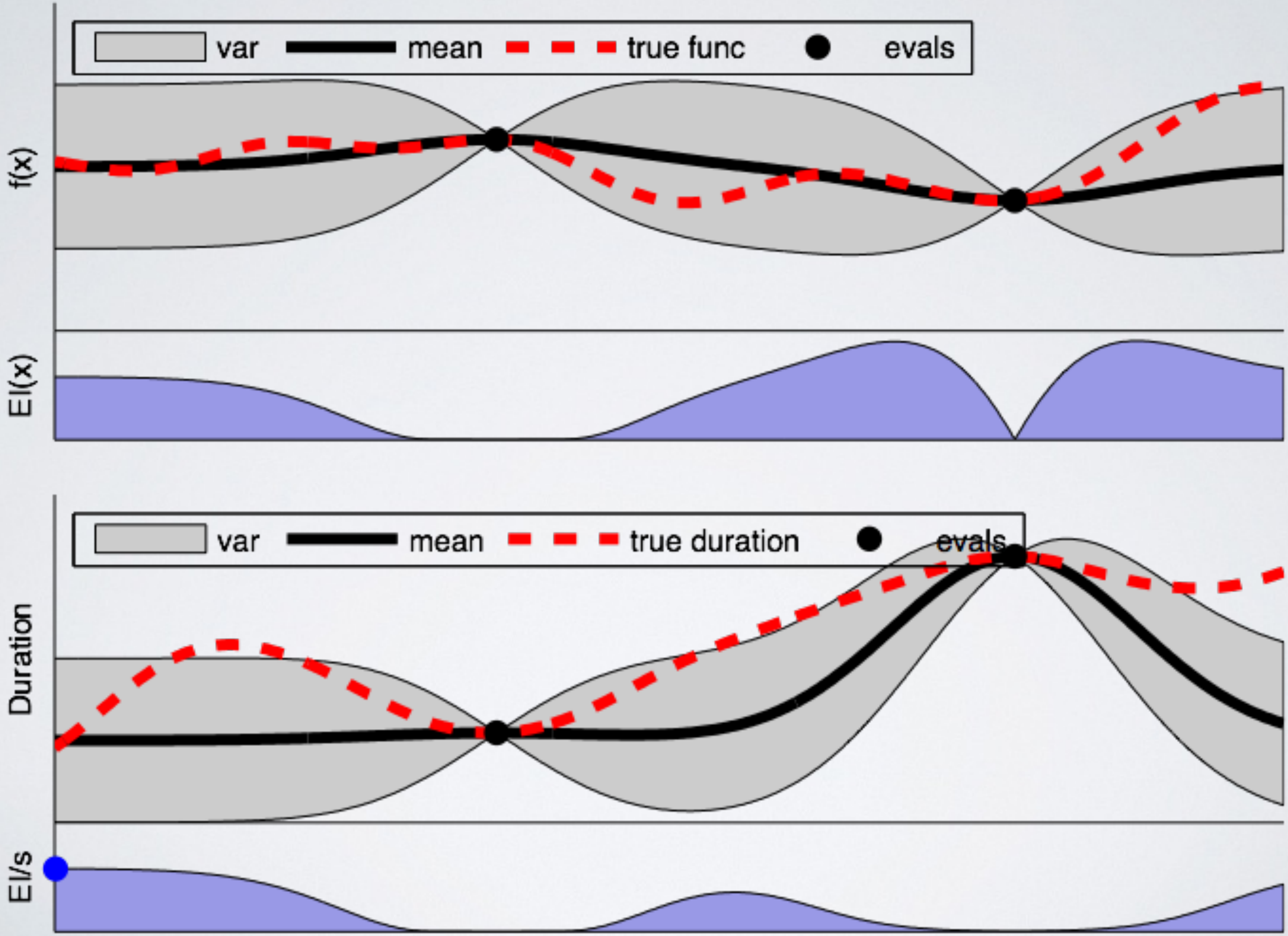


Why Doesn't Everyone Use This?

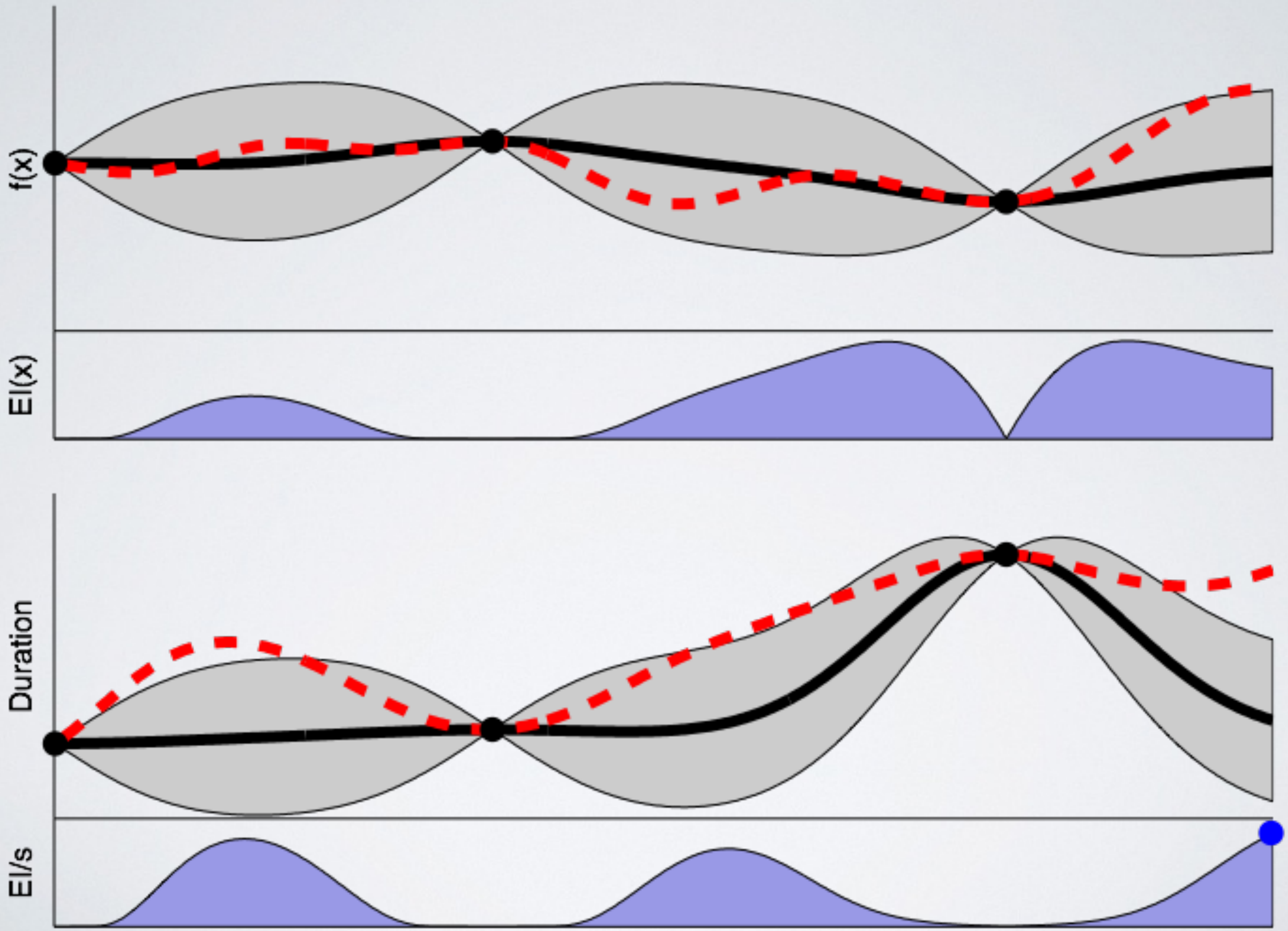
These ideas have been around for *decades*.
Why is Bayesian optimization in broader use?

- **Fragility and poor default choices.**
  Getting the function model wrong can be catastrophic.
- **There hasn't been standard software available.**
  It's a bit tricky to build such a system from scratch.
- **Experiments are run sequentially.**
  We want to take advantage of cluster computing.
- **Limited scalability in dimensions and evaluations.**
  We want to solve big problems.

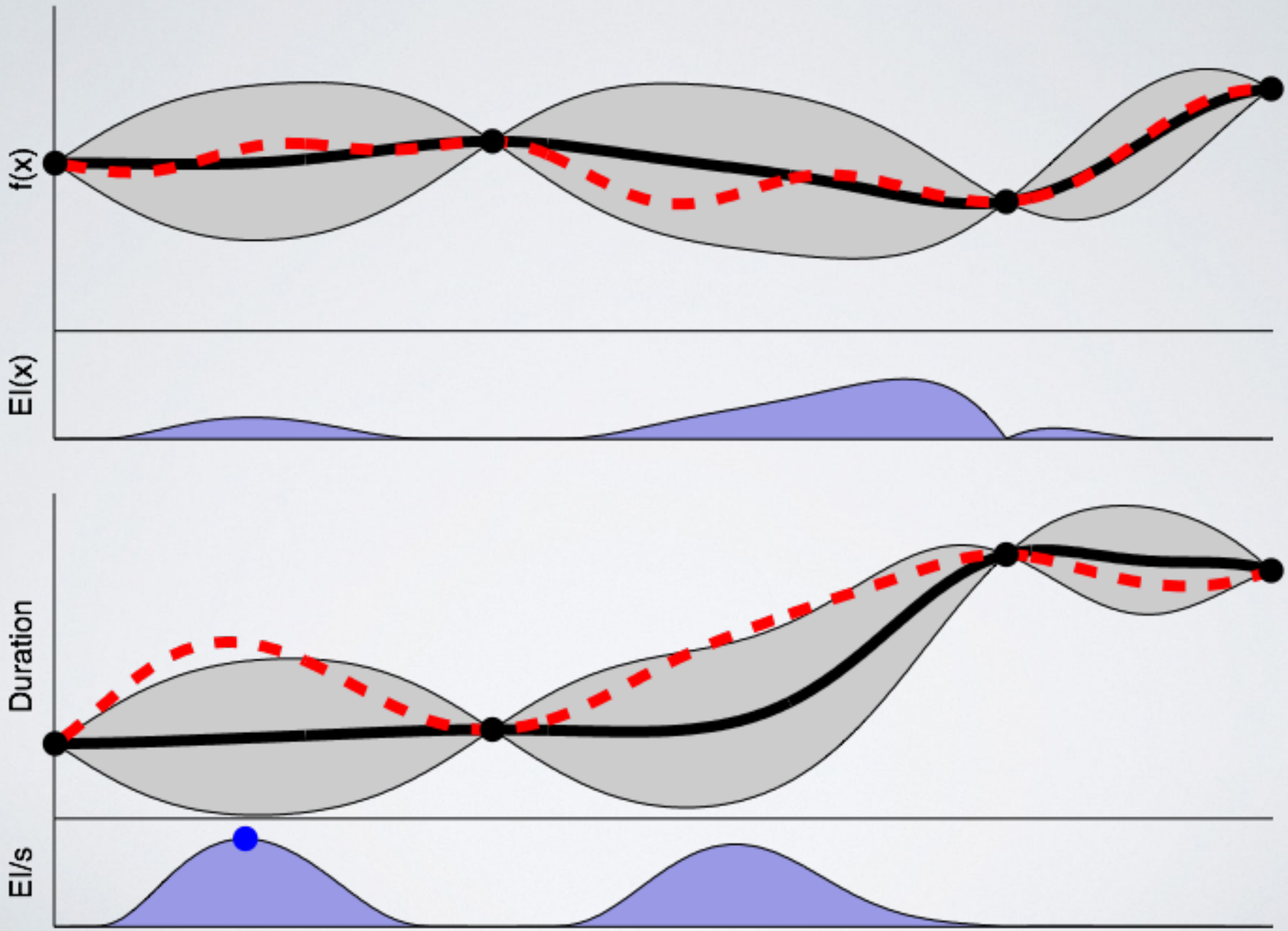
Expected Improvement per Second



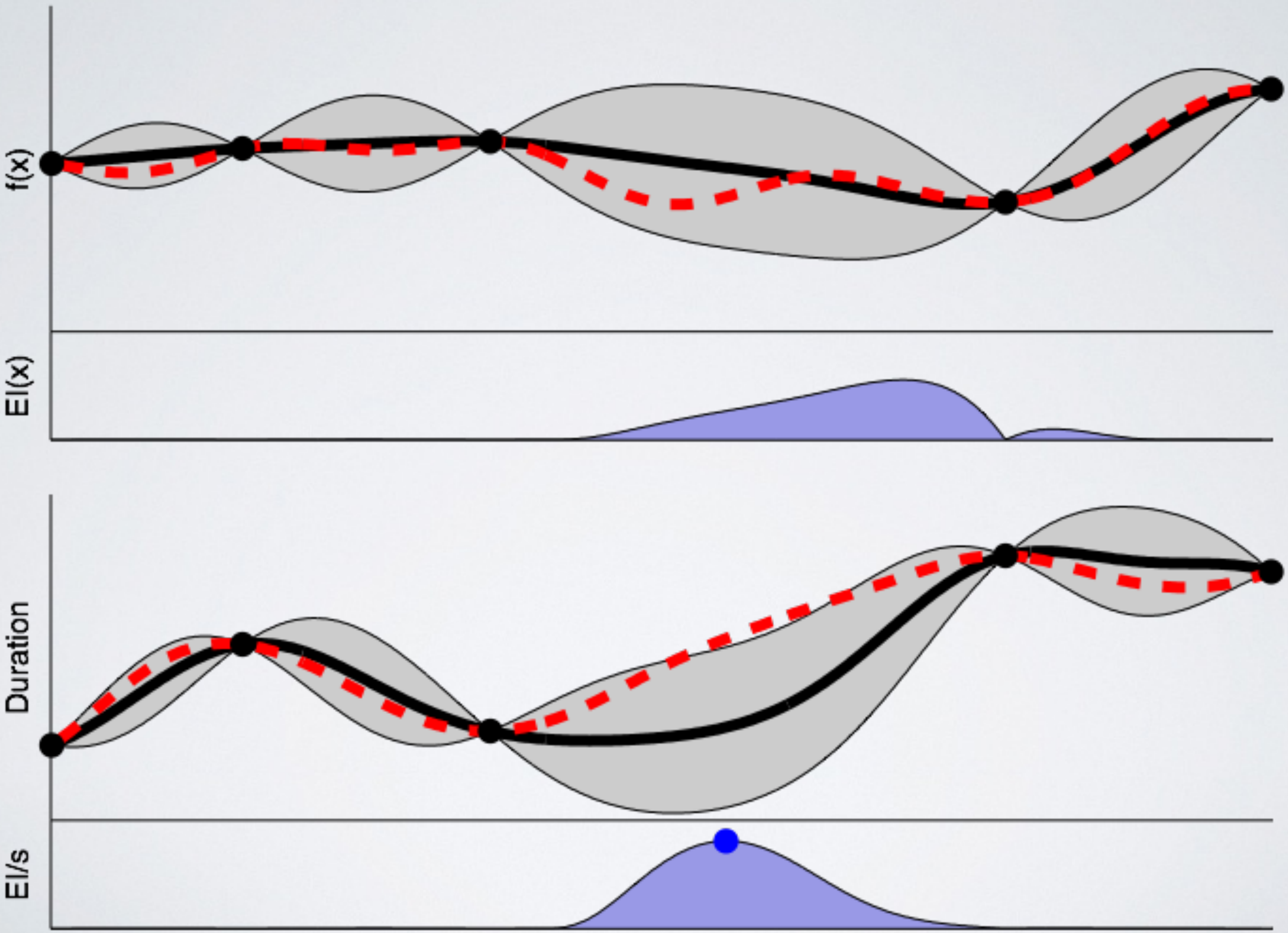
Expected Improvement per Second



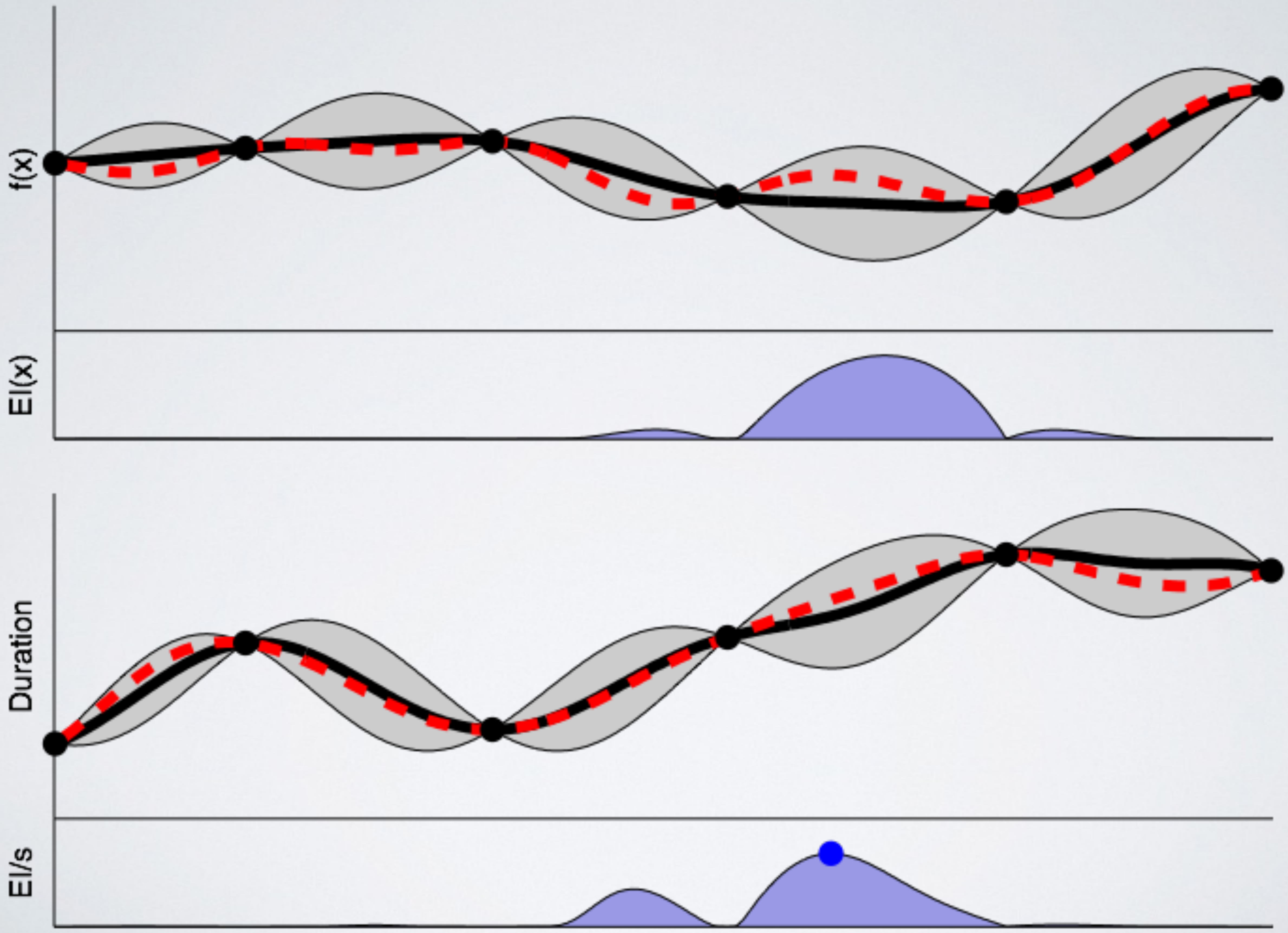
Expected Improvement per Second



Expected Improvement per Second

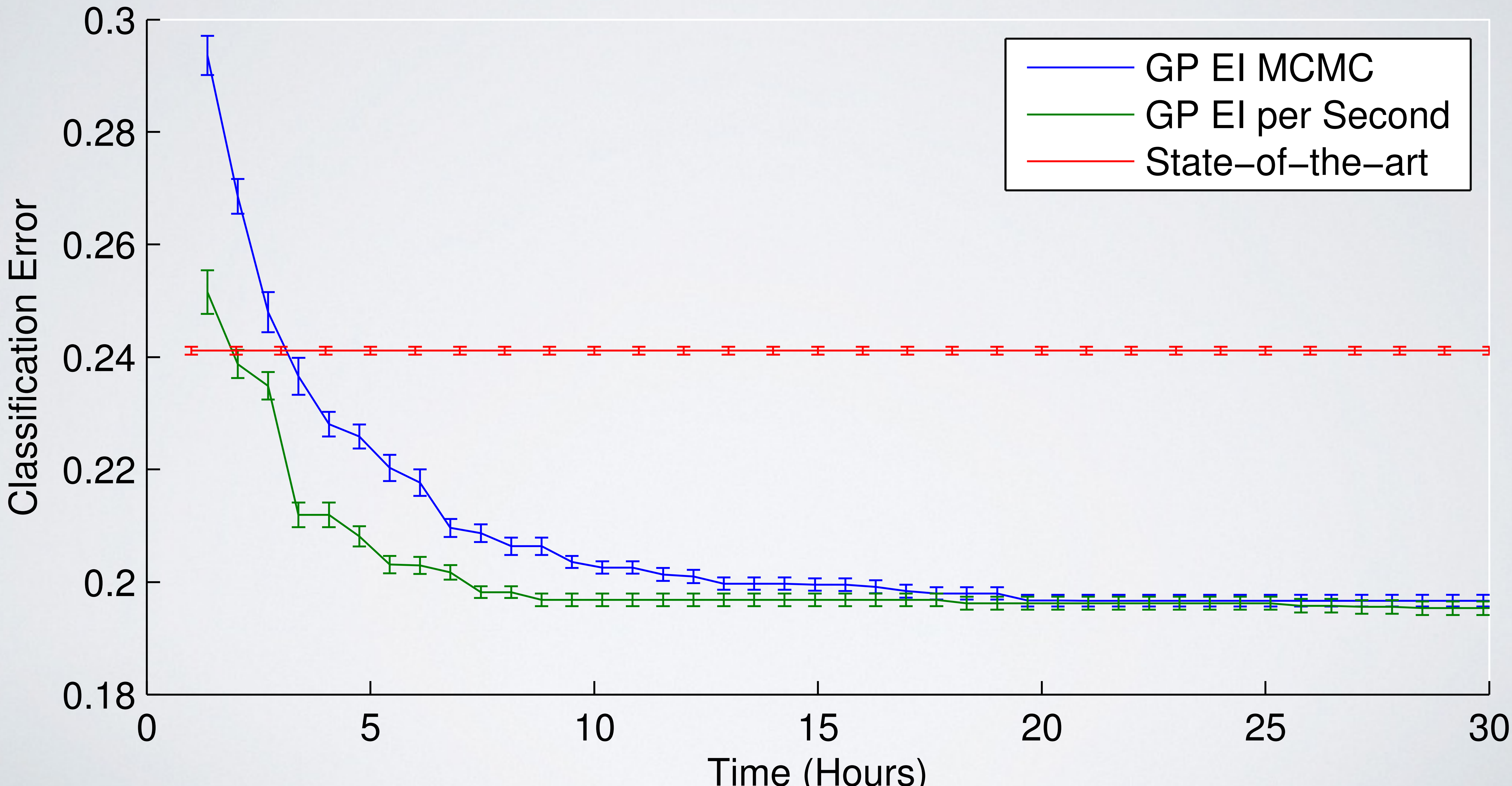


Expected Improvement per Second



Expected Improvement per Second

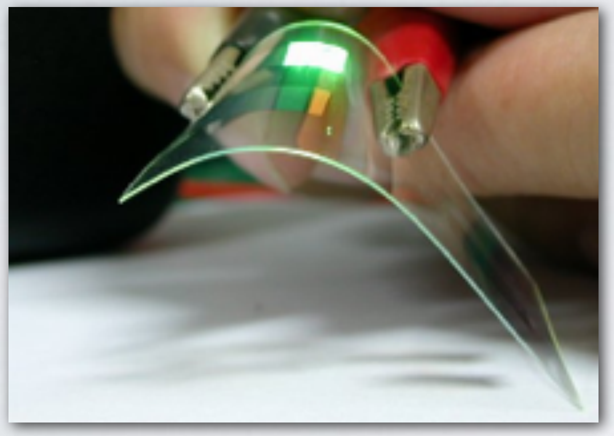
CIFAR10: Deep convolutional neural net (Krizhevsky)
Achieves 9.5% test error vs. 11% with hand tuning.



Snoek, Larochelle & RPA, NIPS 2012

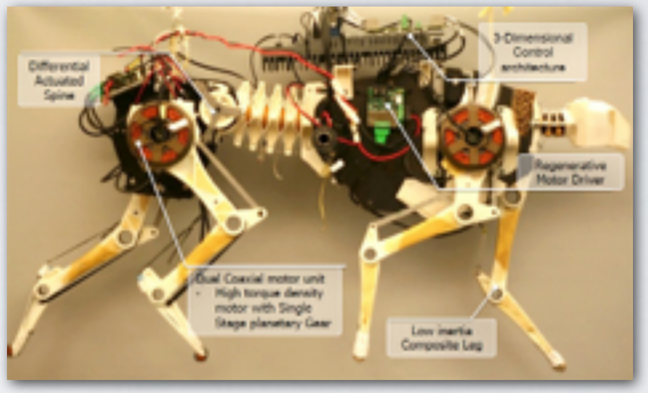
New Directions for Bayesian Optimization

SAMSUNG



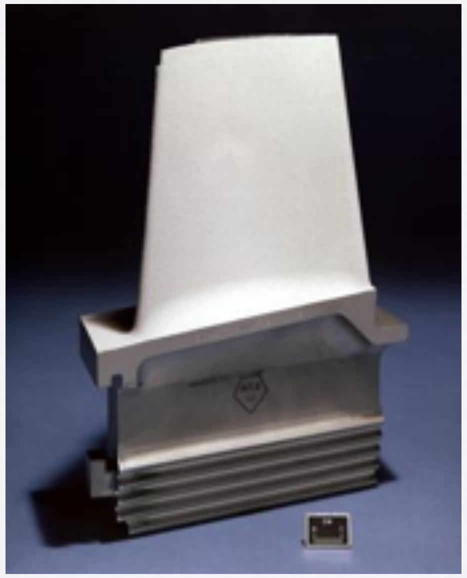
Finding new organic materials

Massachusetts Institute of Technology



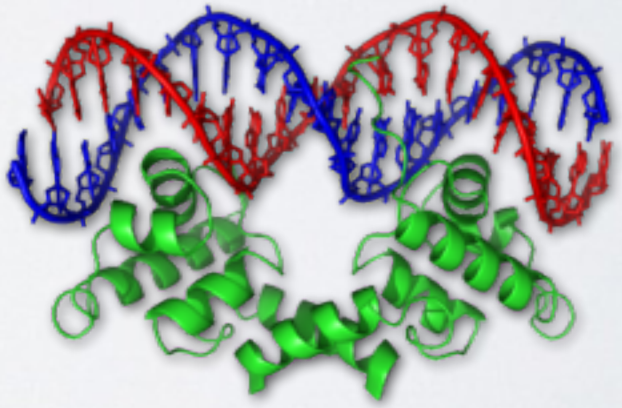
Optimizing robot control systems

Massachusetts Institute of Technology



Improving turbine blade design

BROAD INSTITUTE



Designing DNA for protein affinities

Challenges and Perspectives

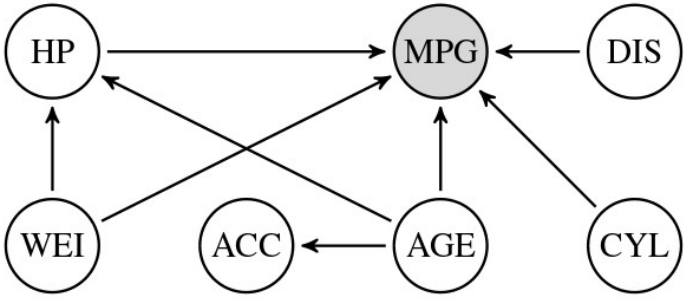
- Applying statistical methods to the global design process of a large engineering project is very difficult.
- We can attack many very useful smaller problems.
- Culture change: a point estimate shouldn’t be a valid answer anymore!

The Future of ML in Engineering?

- Probabilistic Programming for model-based ML

    1. Write down model with prior knowledge (e.g. science).
    2. Run *automated inference engine*.
    3. Analyse results, improve model and iterate.

- Causality



Lopez-Paz et al. (2015)

Thank You!

roger@rogerfrigola.com