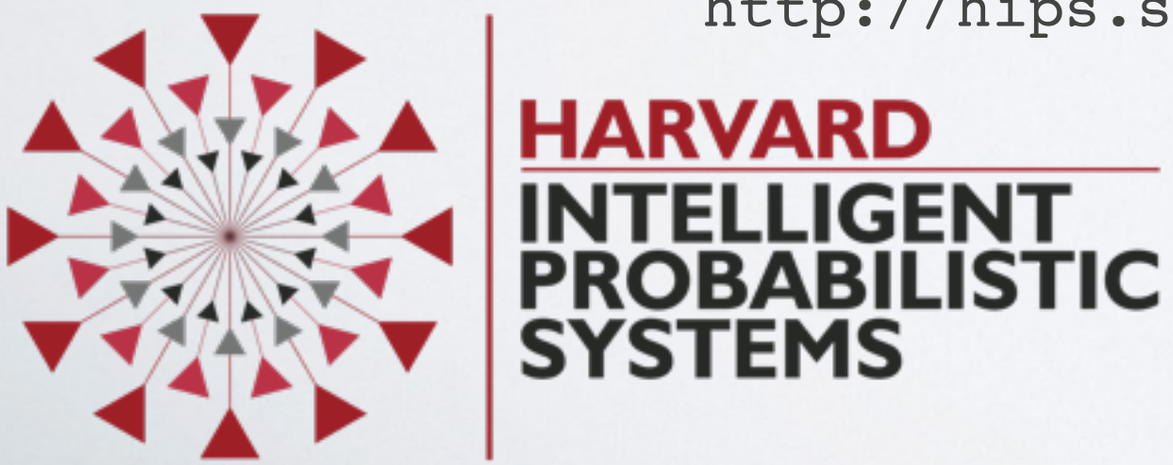
# A Tutorial on Bayesian Optimization for Machine Learning

Ryan P. Adams
School of Engineering and Applied Sciences
Harvard University

**Presented by the University of Toronto CSC411/2515 Fall 2018 Teaching Staff**

**Source - https://www.iro.umontreal.ca/~bengioy/cifar/NCAP2014-summerschool/slides/Ryan_adams_140814_bayesopt_ncap.pdf**

http://hips.seas.harvard.edu




# Machine Learning Meta-Challenges

- **Increasing Model Complexity**
  More flexible models have more parameters.

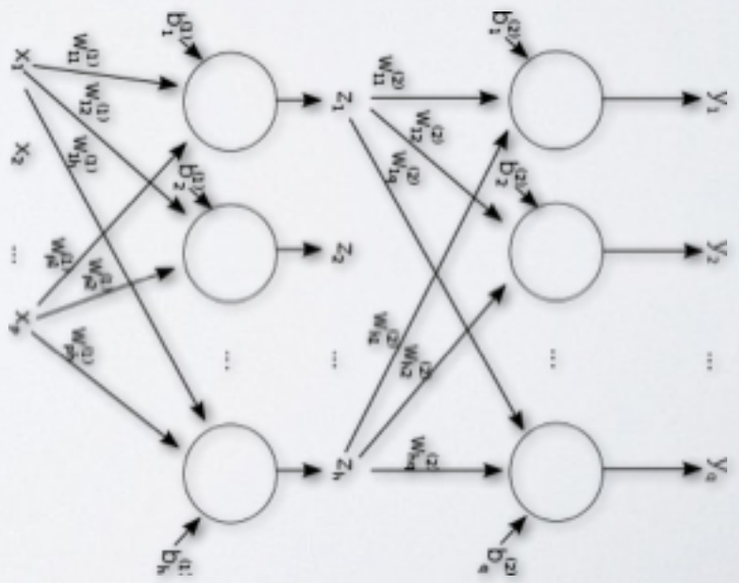
- **More Sophisticated Fitting Procedures**
  Non-convex optimization has many knobs to turn.

- **Less Accessible to Non-Experts**
  Harder to apply complicated techniques.

- **Results Are Less Reproducible**
  Too many important implementation details are missing.

# Example: Deep Neural Networks

- Resurgent interest in large neural networks.
- When well-tuned, very successful for visual object identification, speech recognition, comp bio, ...
- Big investments by Google, Facebook, Microsoft, etc.
- Many choices: number of layers, weight regularization, layer size, which nonlinearity, batch size, learning rate schedule, stopping conditions



# Search for Good Hyperparameters?

- **Define an objective function.**
  Most often, we care about generalization performance.
  Use cross validation to measure parameter quality.

- **How do people currently search? Black magic.**
  Grid search
  Random search
  Grad student descent

- **Painful!**
  Requires many training cycles.
  Possibly noisy.

# Can We Do Better? Bayesian Optimization

- **Build a probabilistic model for the objective.**
  Include hierarchical structure about units, etc.

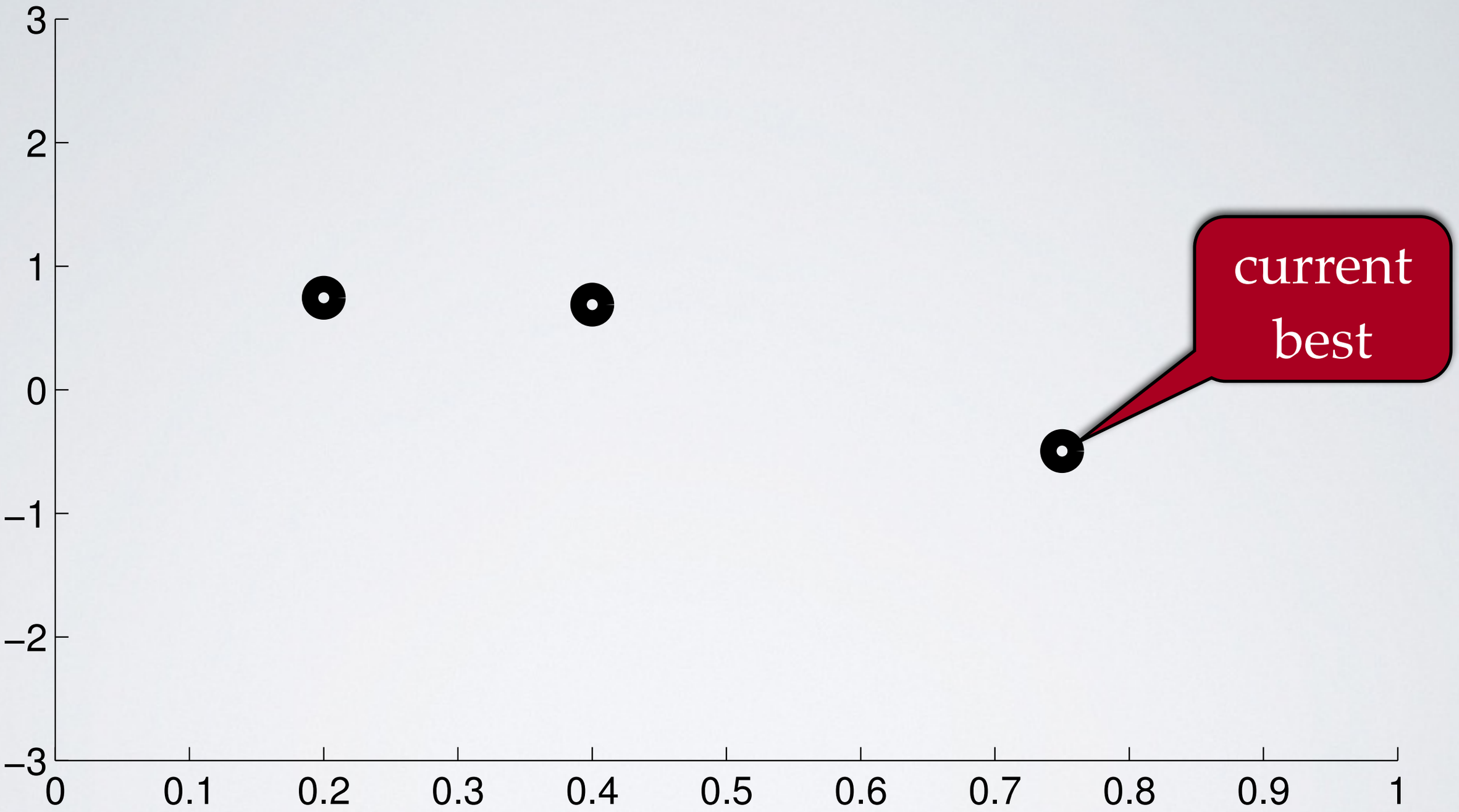
- **Compute the posterior predictive distribution.**
  Integrate out all the possible true functions.
  We use Gaussian process regression.

- **Optimize a cheap proxy function instead.**
  The model is much cheaper than that true objective.

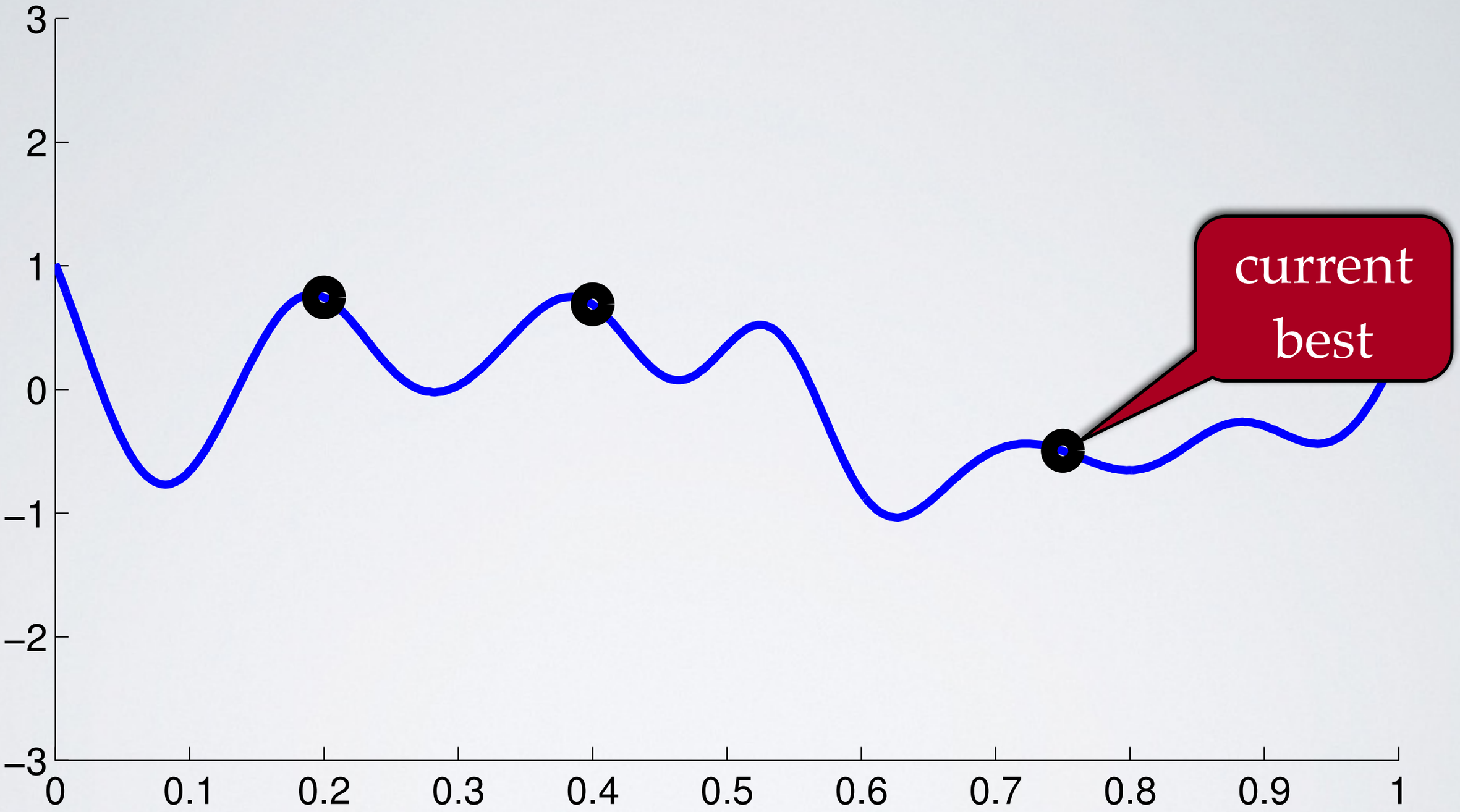
*The main insight:*

Make the proxy function exploit uncertainty to balance **exploration** against **exploitation**.

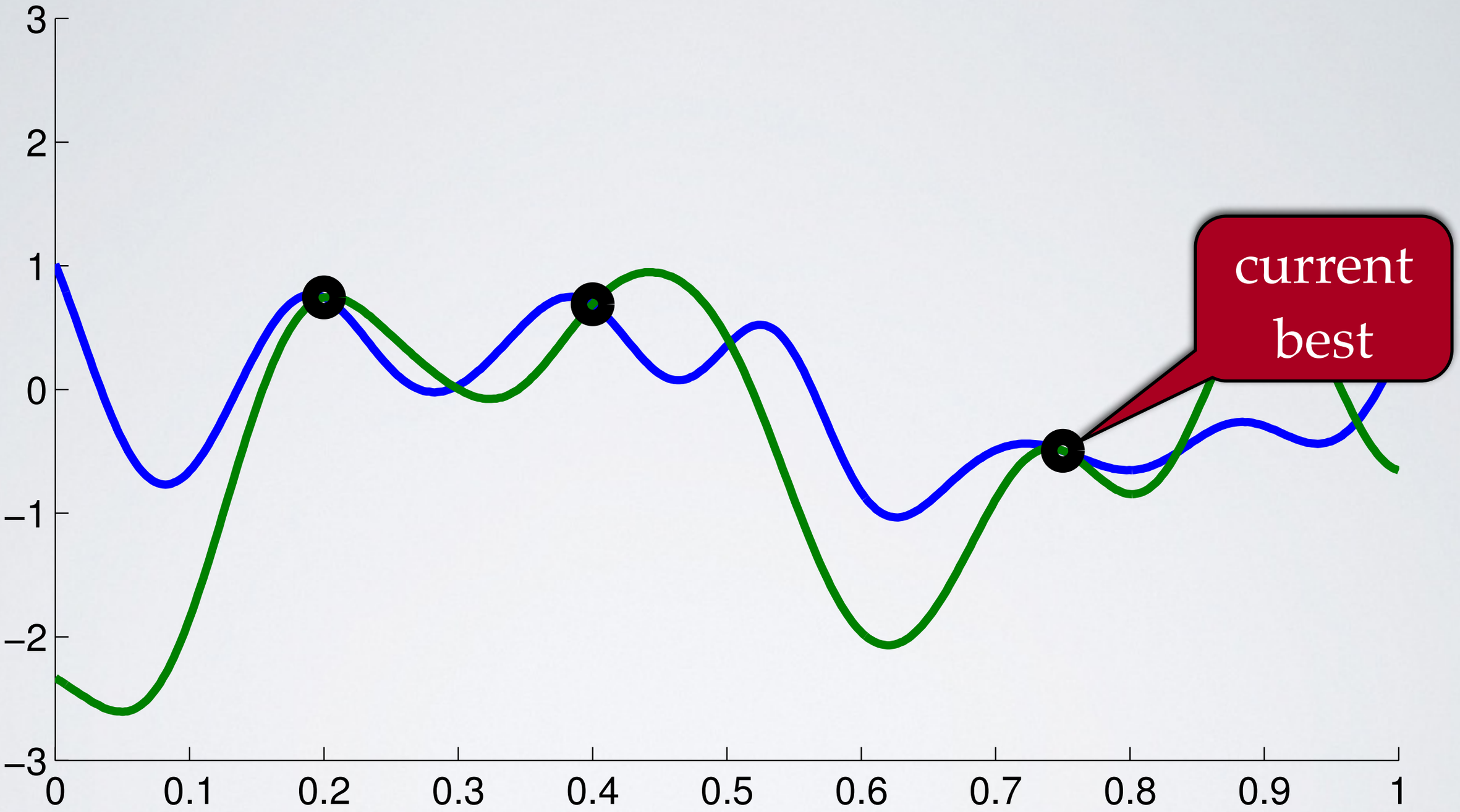
# The Bayesian Optimization Idea



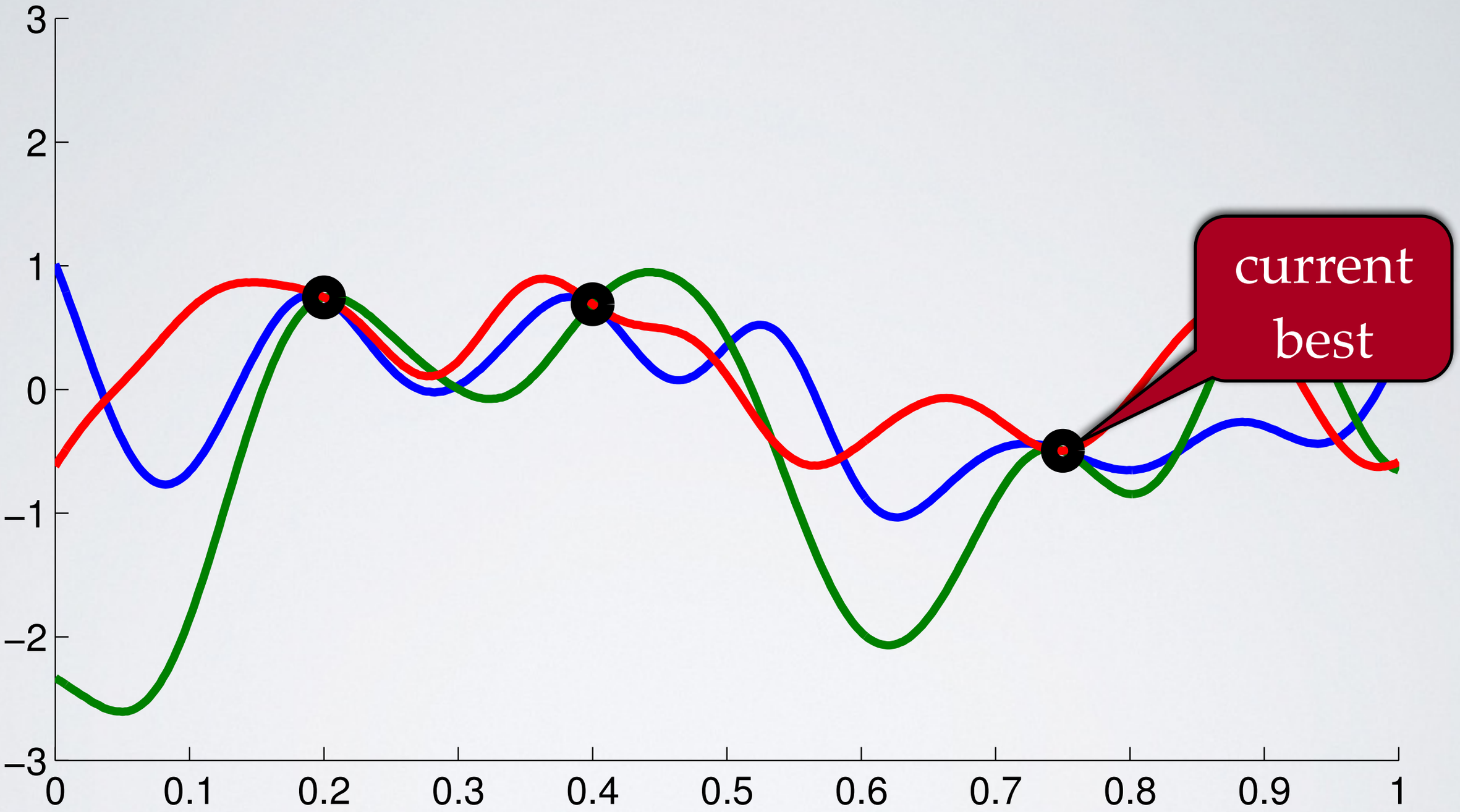
# The Bayesian Optimization Idea



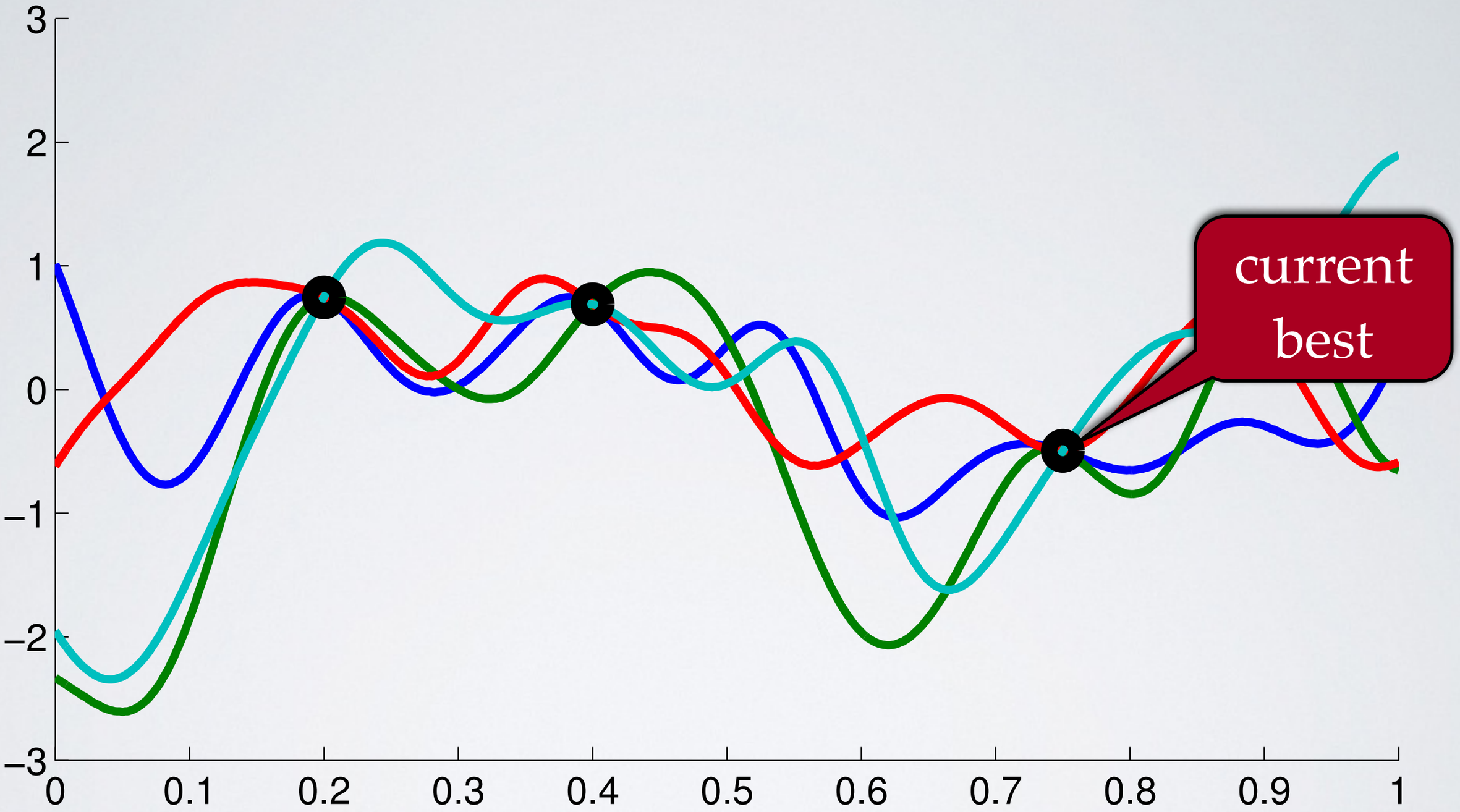
# The Bayesian Optimization Idea



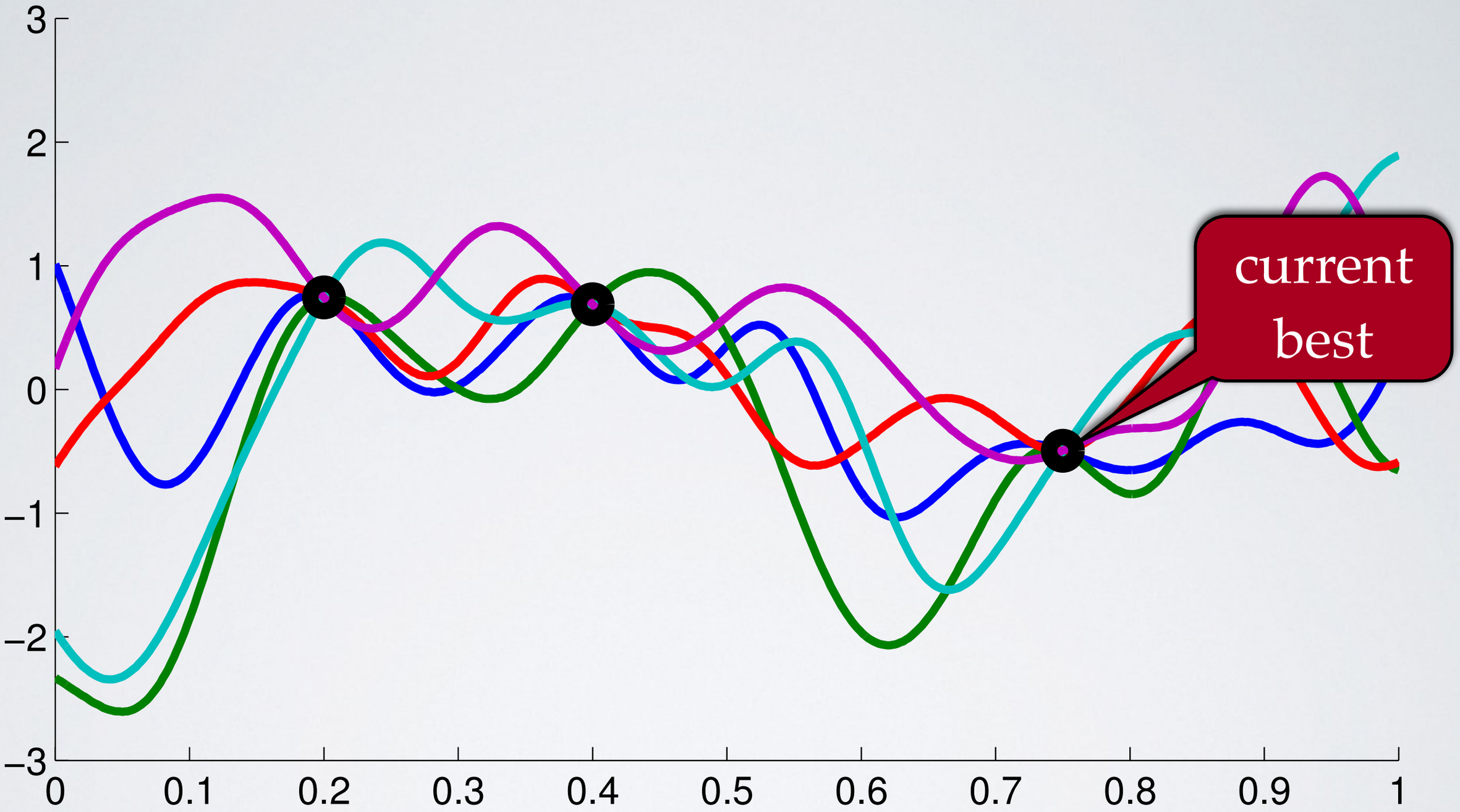
# The Bayesian Optimization Idea



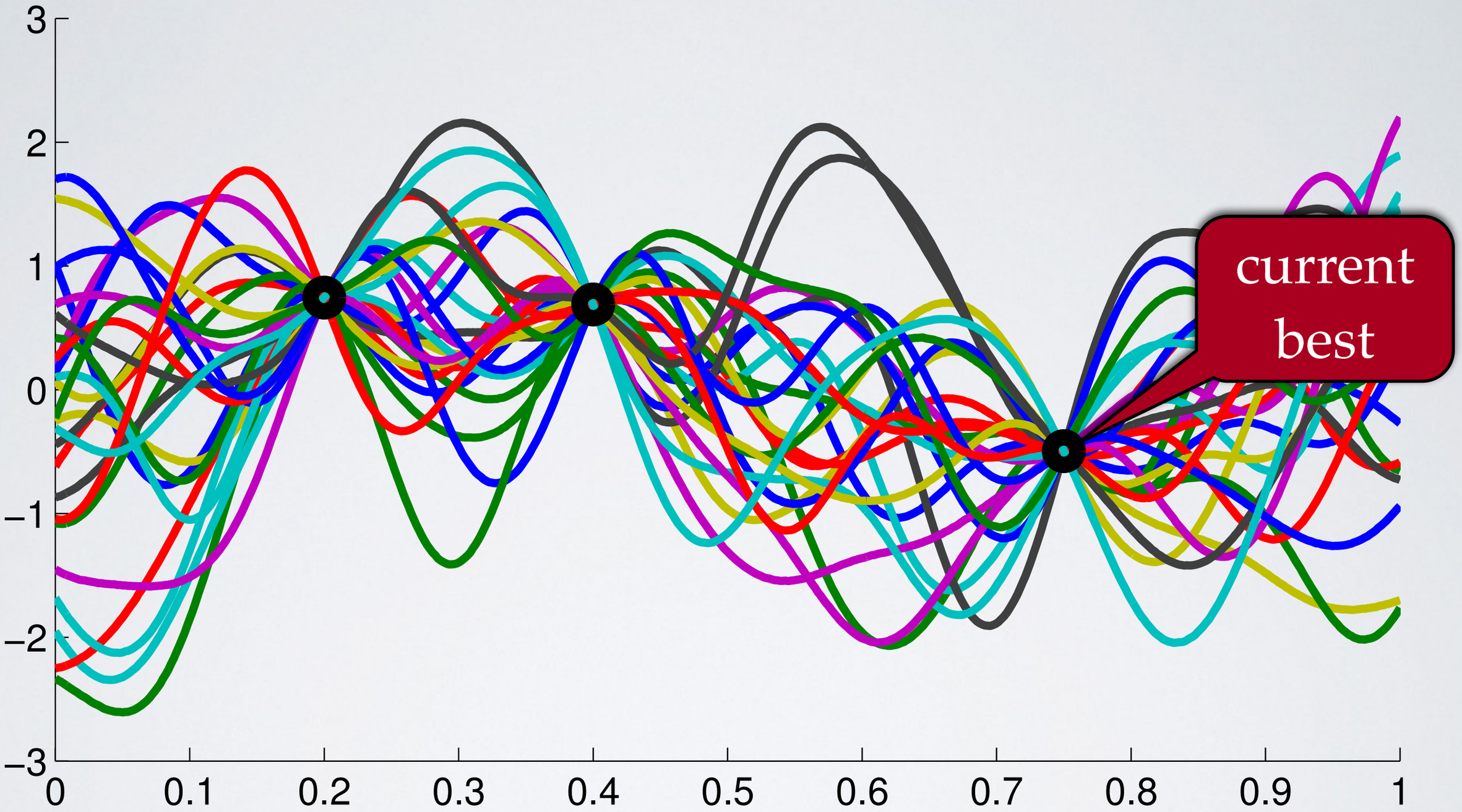
# The Bayesian Optimization Idea



# The Bayesian Optimization Idea

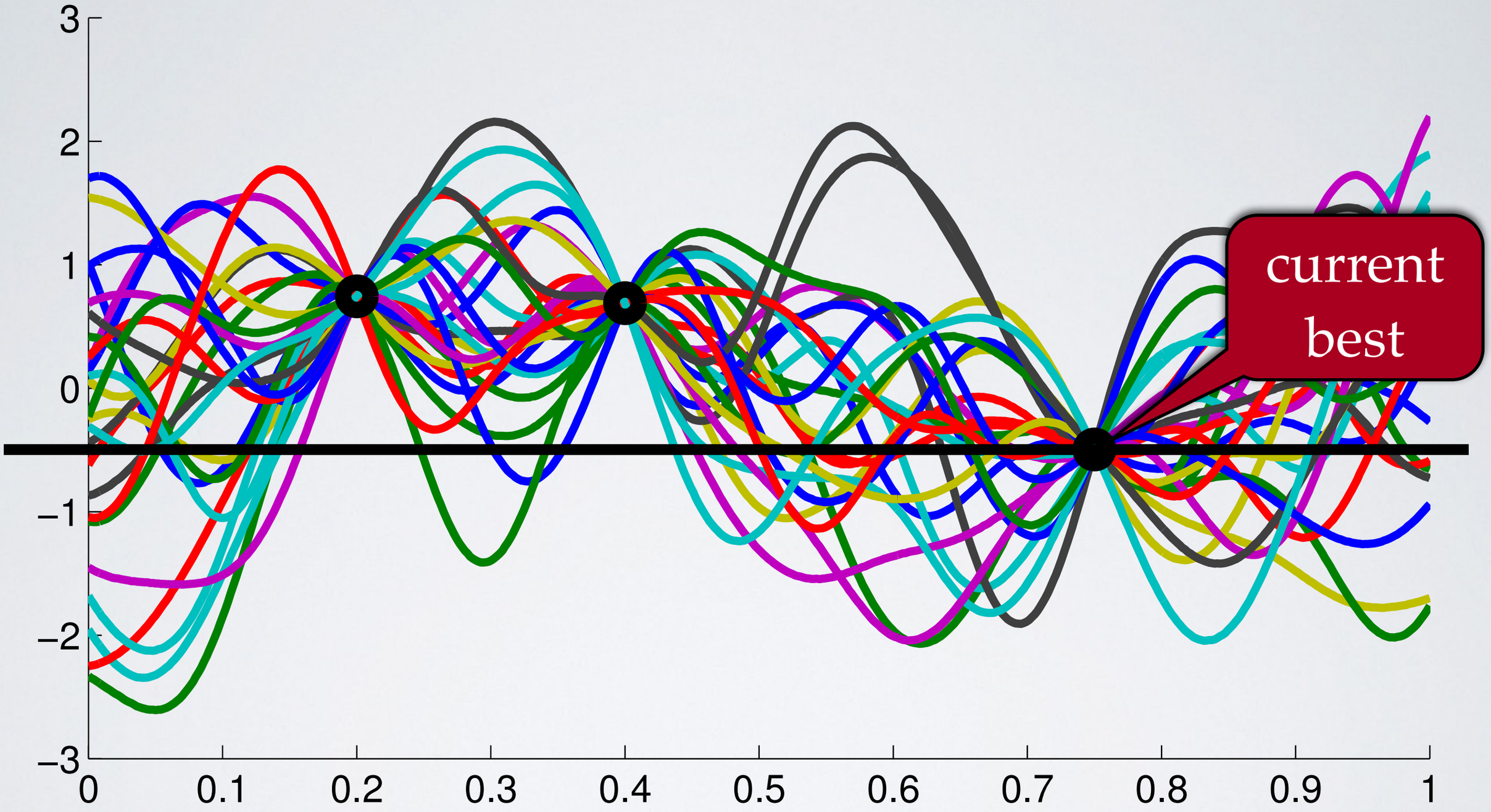


# The Bayesian Optimization Idea



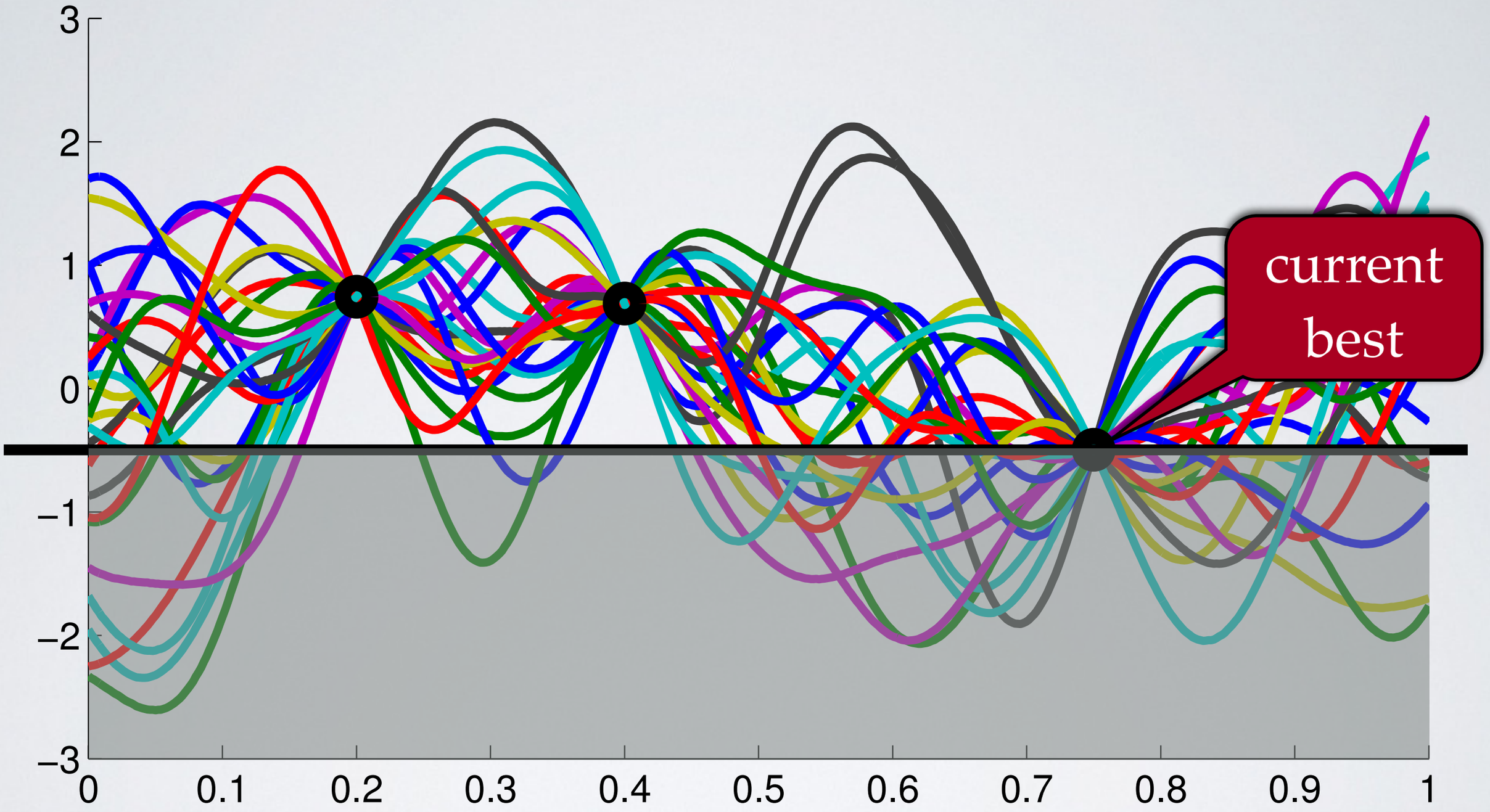

Where should we evaluate next
in order to improve the most?

# The Bayesian Optimization Idea



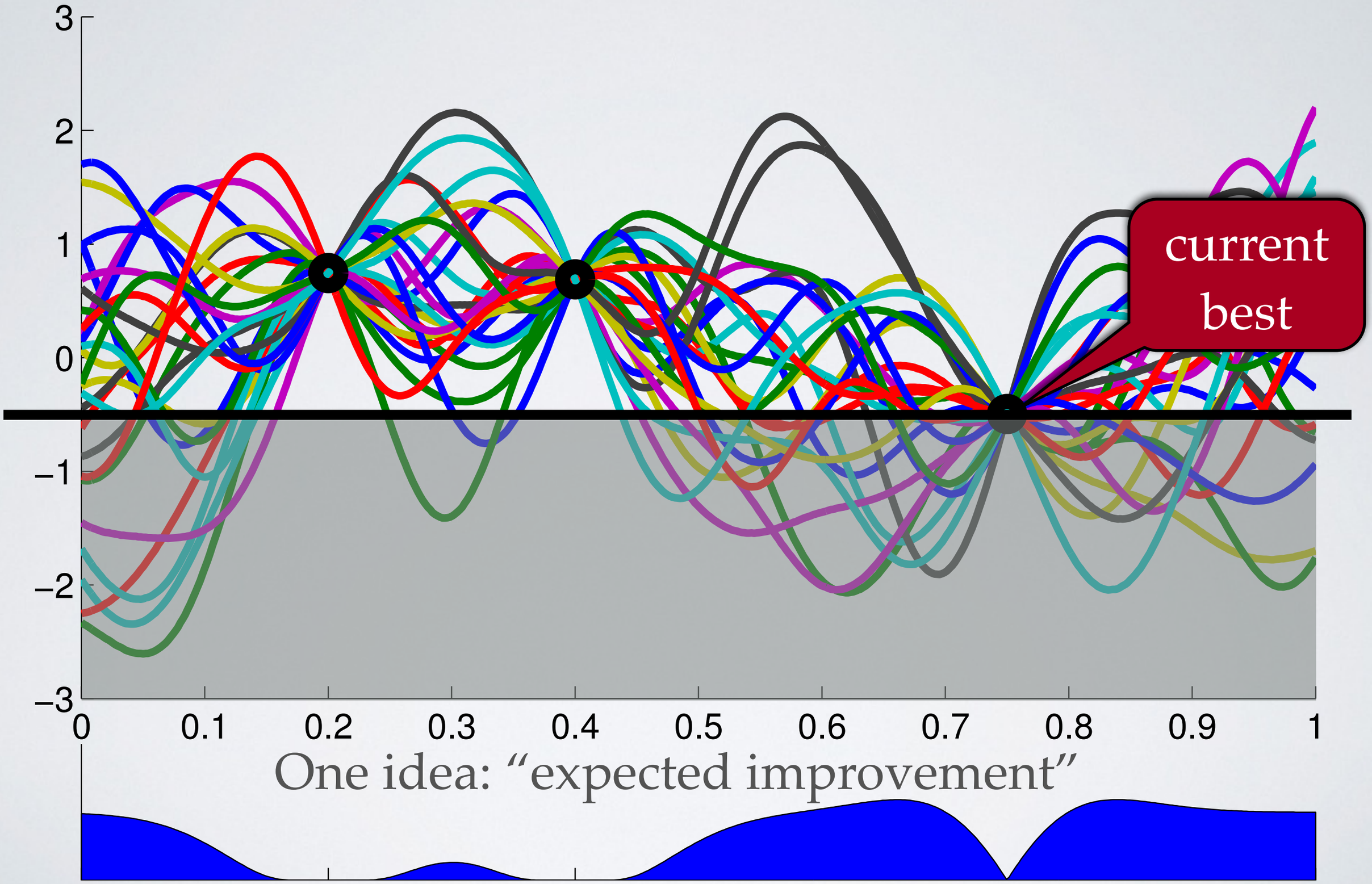

Where should we evaluate next
in order to improve the most?

# The Bayesian Optimization Idea




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

# The Bayesian Optimization Idea




One idea: “expected improvement”

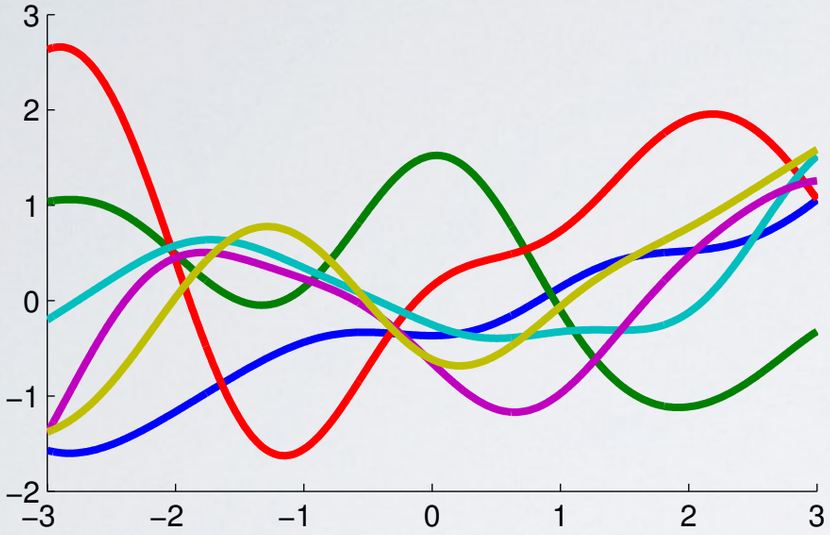
# Today's Topics

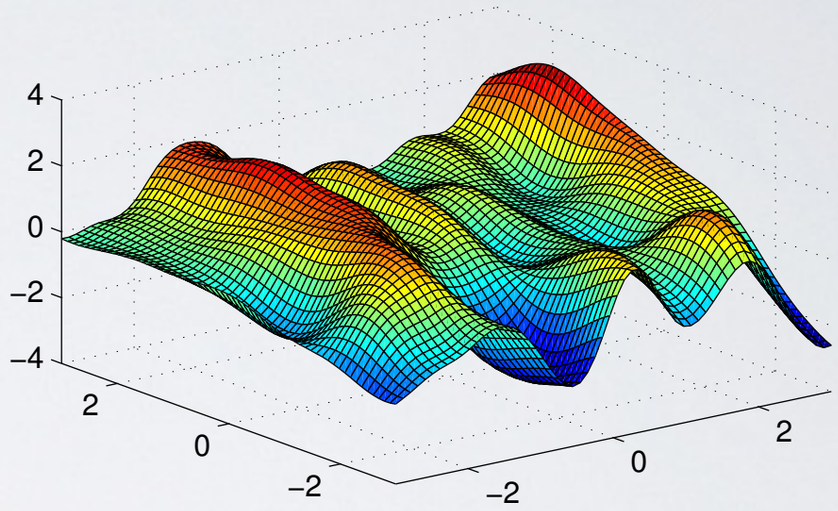
- Review of Gaussian process priors
- Bayesian optimization basics
- Managing covariances and kernel parameters
- ~~Accounting for the cost of evaluation~~
- ~~Parallelizing training~~
- ~~Sharing information across related problems~~
- ~~Better models for nonstationary functions~~
- ~~Random projections for high-dimensional problems~~
- ~~Accounting for constraints~~
- ~~Leveraging partially-completed training runs~~

**The crossed out material is not necessary to cover in tutorial.**

# Gaussian Processes as Function Models

Nonparametric prior on functions specified in terms of a positive definite kernel.

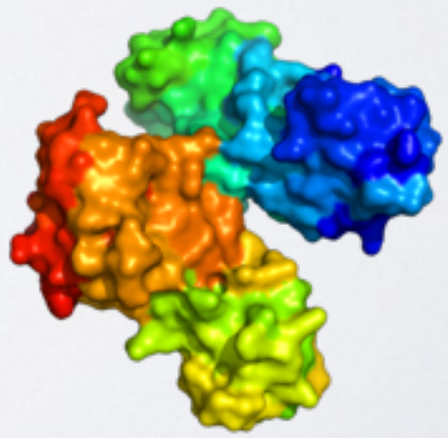



```
the quick brown fox
jumps over the lazy dog
```

strings (Haussler 1999)

permutations (Kondor 2008)



proteins (Borgwardt 2007)

# Gaussian Processes

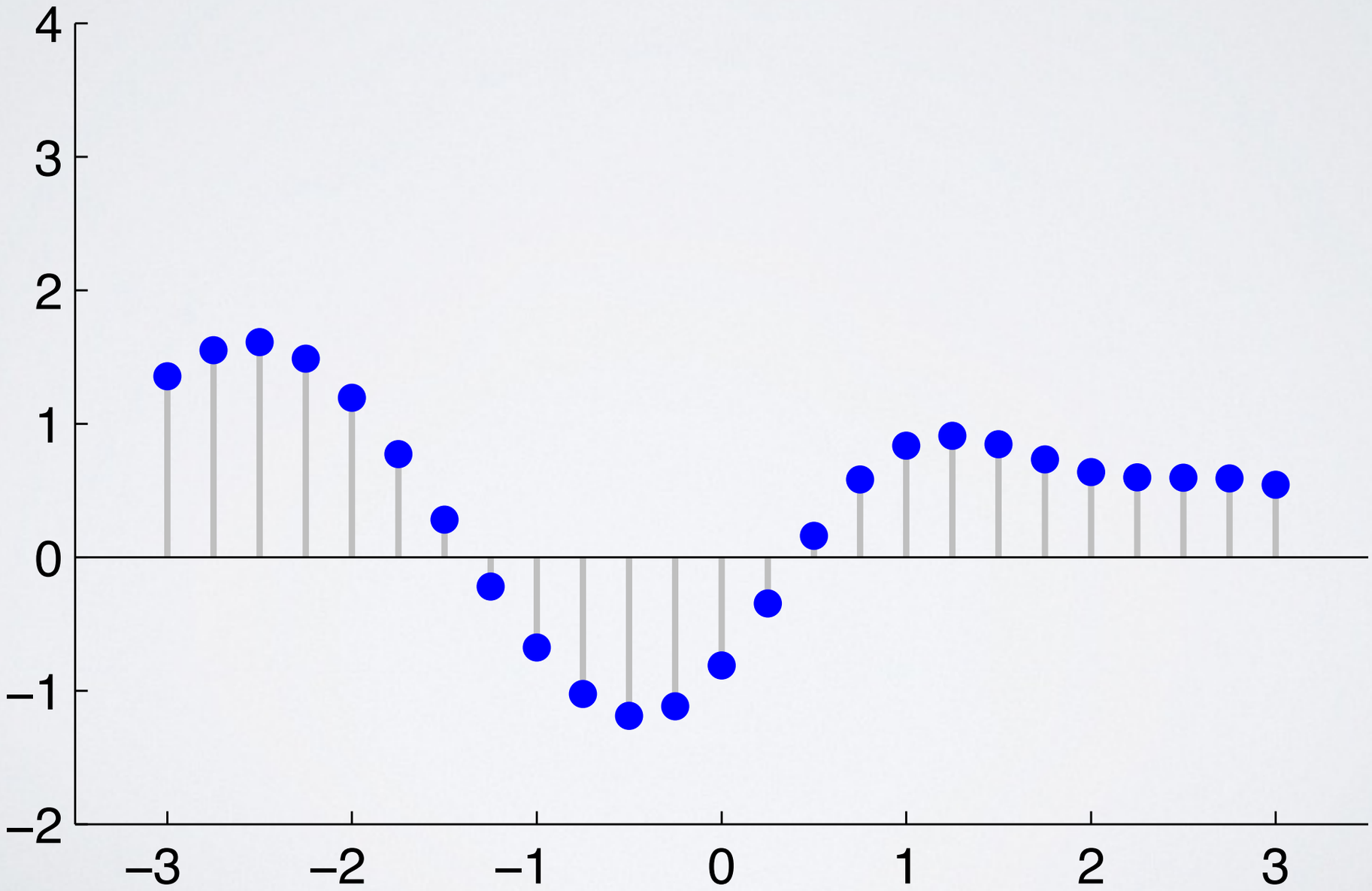
- Gaussian process (GP) is a distribution on functions.
- Allows tractable Bayesian modeling of functions without specifying a particular finite basis.
- Input space (where we're optimizing) $\mathcal{X}$
- Model scalar functions $f : \mathcal{X} \to \mathbb{R}$
- Positive definite covariance function $C : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
- Mean function $m : \mathcal{X} \to \mathbb{R}$

# Gaussian Processes

Any finite set of $N$ points in $\mathcal{X}$, $\{x_n\}_{n=1}^N$ induces a homologous $N$-dimensional Gaussian distribution on $\mathbb{R}^N$, taken to be the distribution on $\{y_n = f(x_n)\}_{n=1}^N$

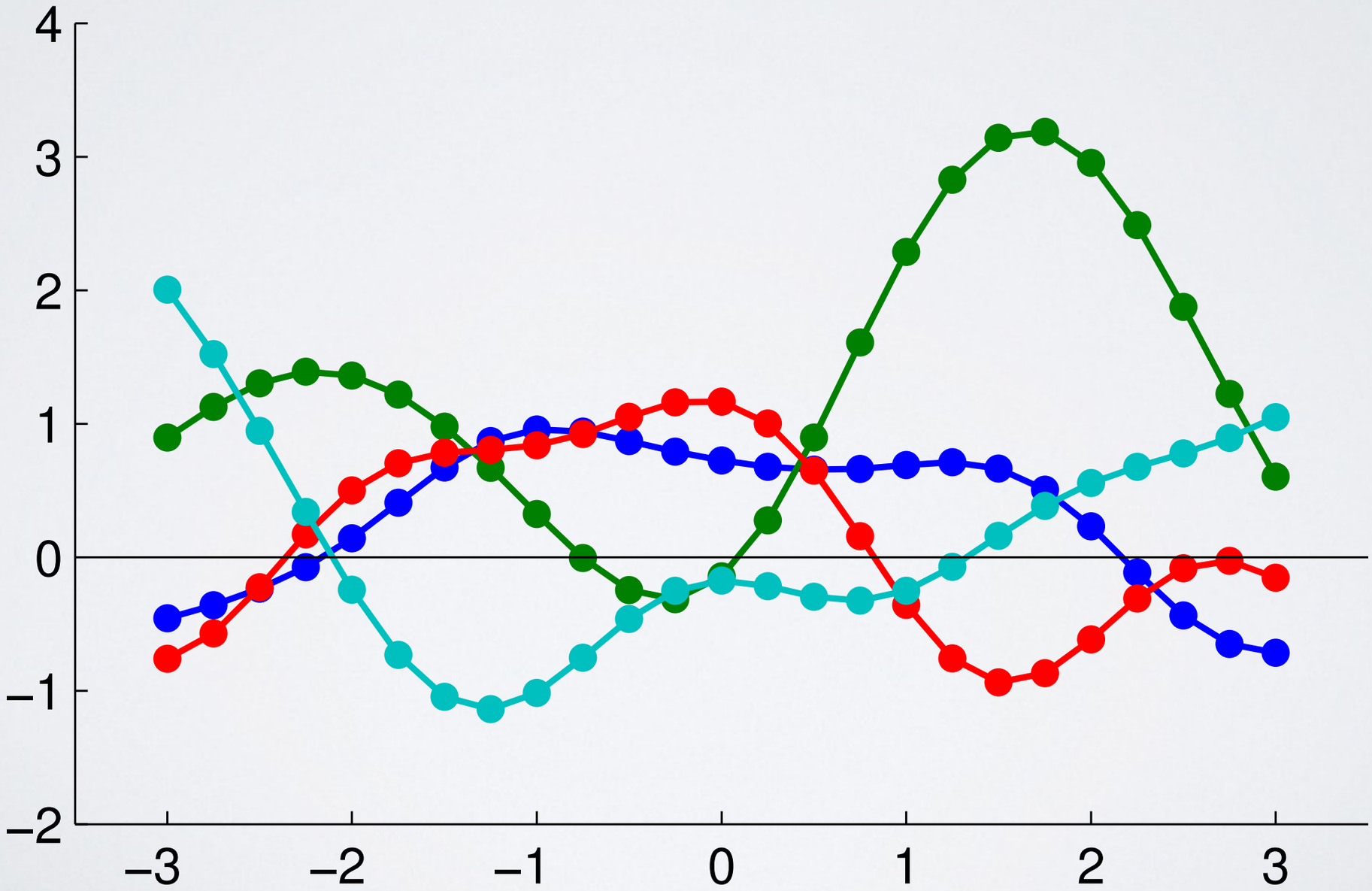
# Gaussian Processes

Any finite set of $N$ points in $\mathcal{X}$, $\{x_n\}_{n=1}^N$ induces a homologous $N$-dimensional Gaussian distribution on $\mathbb{R}^N$, taken to be the distribution on $\{y_n = f(x_n)\}_{n=1}^N$



# Gaussian Processes

Any finite set of $N$ points in $\mathcal{X}$, $\{x_n\}_{n=1}^N$ induces a homologous $N$-dimensional Gaussian distribution on $\mathbb{R}^N$, taken to be the distribution on $\{y_n = f(x_n)\}_{n=1}^N$



# Gaussian Processes

- Due to Gaussian form, closed-form solutions for many useful questions about finite data.
- Marginal likelihood:

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

- Predictive distribution at test points $\{\boldsymbol{x}_m\}_{m=1}^M$:

$$\boldsymbol{y}^{\mathsf{test}} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$$

$$\boldsymbol{m} = \boldsymbol{k}_\theta^\mathsf{T} \boldsymbol{K}_\theta^{-1} \boldsymbol{y} \qquad \boldsymbol{\Sigma} = \boldsymbol{\kappa}_\theta - \boldsymbol{k}_\theta^\mathsf{T} \boldsymbol{K}_\theta^{-1} \boldsymbol{k}_\theta$$

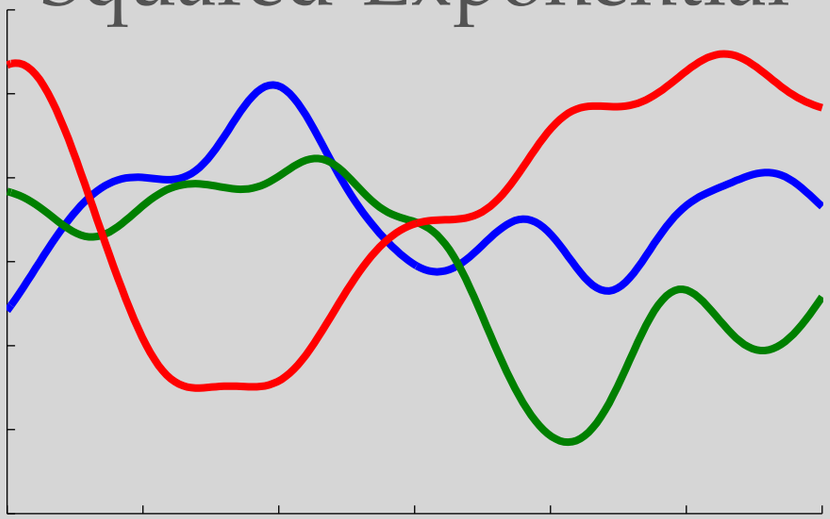
- We compute these matrices from the covariance:

$$[\boldsymbol{K}_\theta]_{n,n'} = C(\boldsymbol{x}_n, \boldsymbol{x}_{n'}\,;\,\theta)$$

$$[\boldsymbol{k}_\theta]_{n,m} = C(\boldsymbol{x}_n, \boldsymbol{x}_m\,;\,\theta) \qquad [\boldsymbol{\kappa}_\theta]_{m,m'} = C(\boldsymbol{x}_m, \boldsymbol{x}_{m'}\,;\,\theta)$$

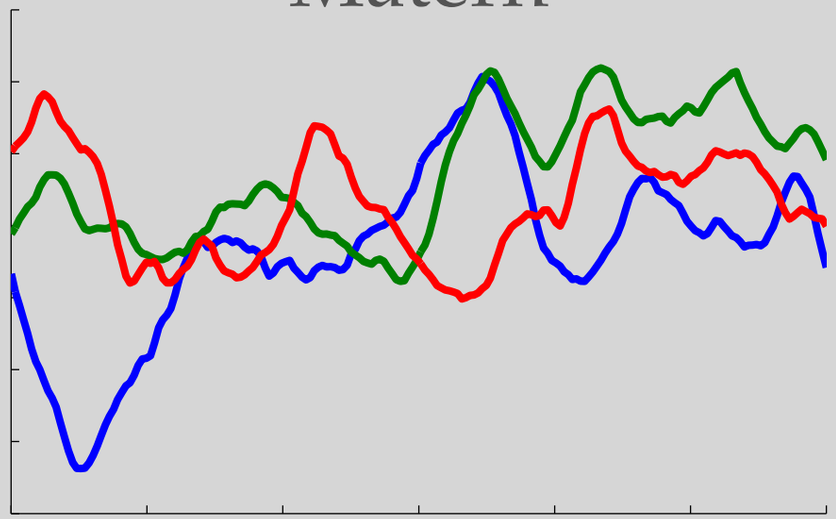
# Examples of GP Covariances

## Squared-Exponential



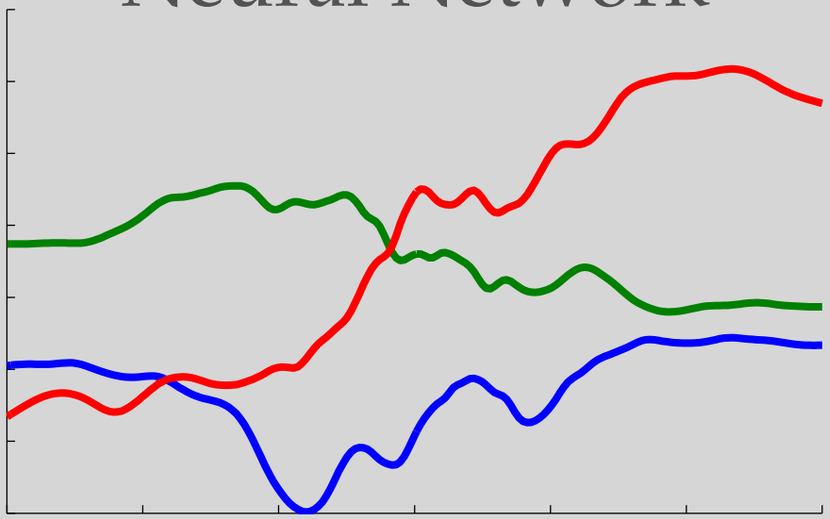
$$C(x, x') = \alpha \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_d - x'_d}{\ell_d}\right)^2\right\}$$

## Matérn



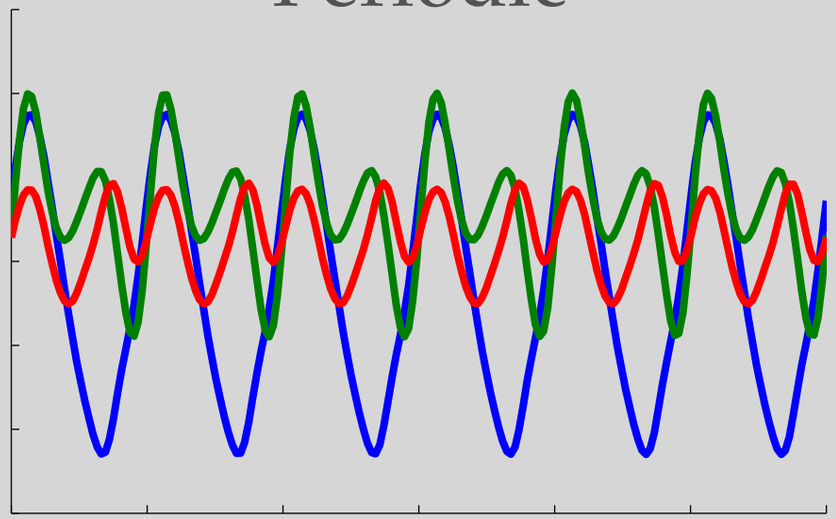
$$C(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\, r}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}\, r}{\ell}\right)$$

## "Neural Network"



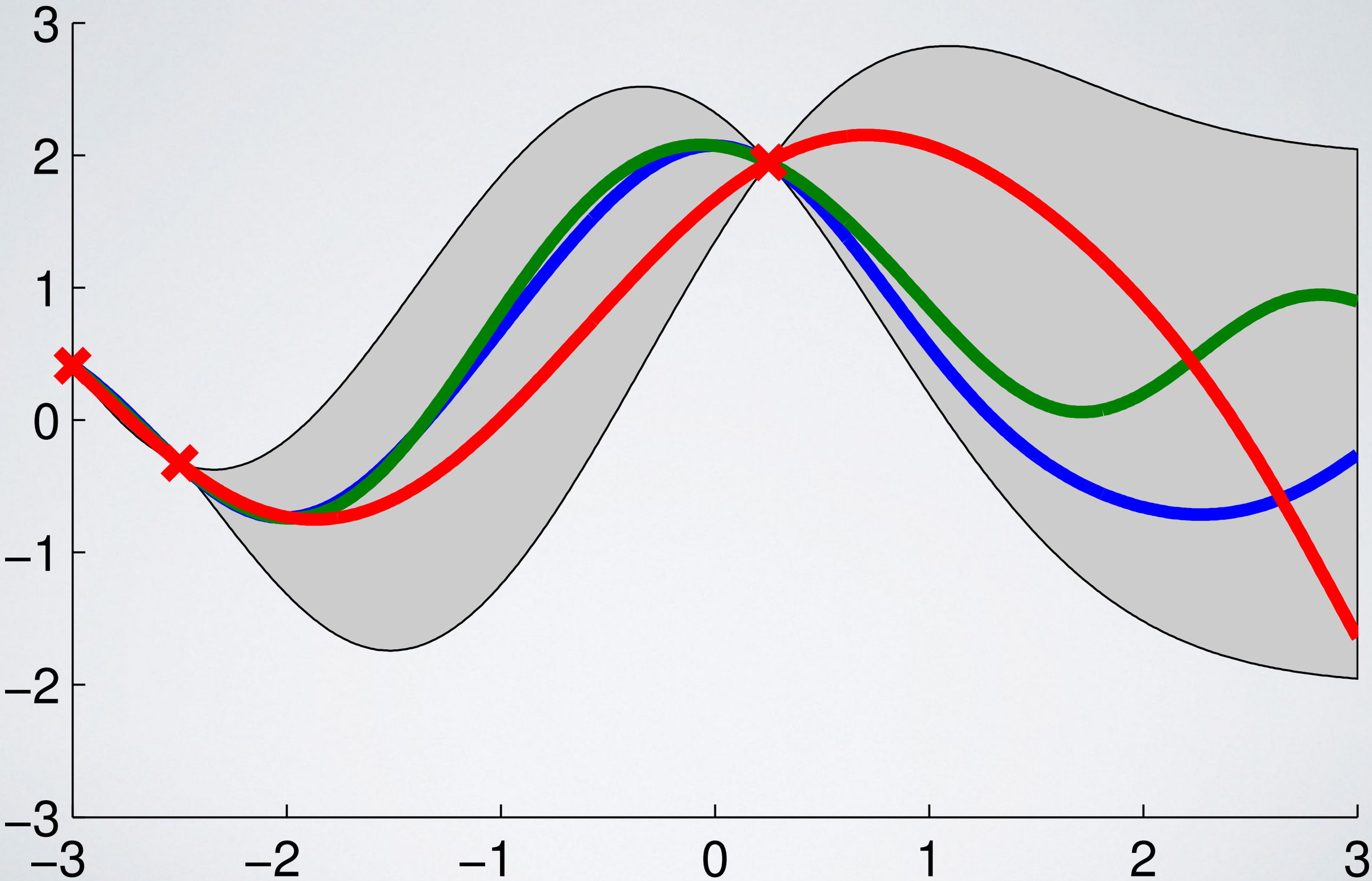
$$C(x, x') = \frac{2}{\pi}\sin^{-1}\left\{\frac{2x^{\mathsf{T}}\Sigma x'}{\sqrt{(1 + 2x^{\mathsf{T}}\Sigma x)(1 + 2x'^{\mathsf{T}}\Sigma x')}}\right\}$$

## 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



# Using Uncertainty in Optimization

- Find the minimum: $x^\star = \arg\min_{x \in \mathcal{X}} f(x)$
- We can evaluate the objective pointwise, but do not have an easy functional form or gradients.
- After performing some evaluations, the GP gives us easy closed-form marginal means and variances.
- **Exploration**: Seek places with high variance.
- **Exploitation**: Seek places with low mean.
- The **acquisition function** balances these for our proxy optimization to determine the next evaluation.

# Closed-Form Acquisition Functions

- The GP posterior gives a predictive mean function $\mu(x)$ and a predictive marginal variance function $\sigma^2(x)$

$$\gamma(x) = \frac{f(x_{\mathsf{best}}) - \mu(x)}{\sigma(x)}$$

- Probability of Improvement (Kushner 1964):

$$a_{\mathsf{PI}}(x) = \Phi(\gamma(x))$$

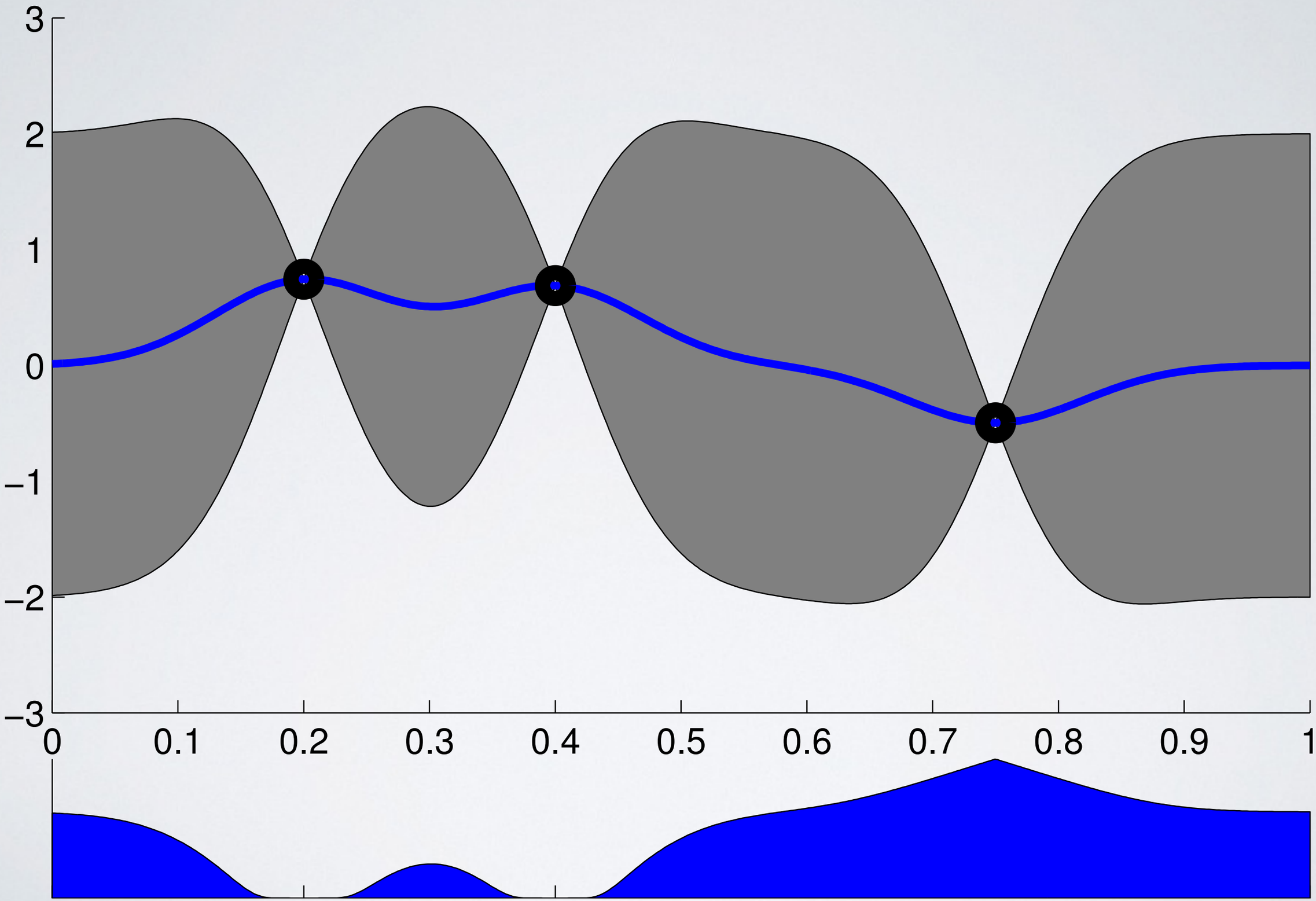
- Expected Improvement (Mockus 1978):

$$a_{\mathsf{EI}}(x) = \sigma(x)(\gamma(x)\Phi(\gamma(x)) + \mathcal{N}(\gamma(x)\,;\,0,1))$$

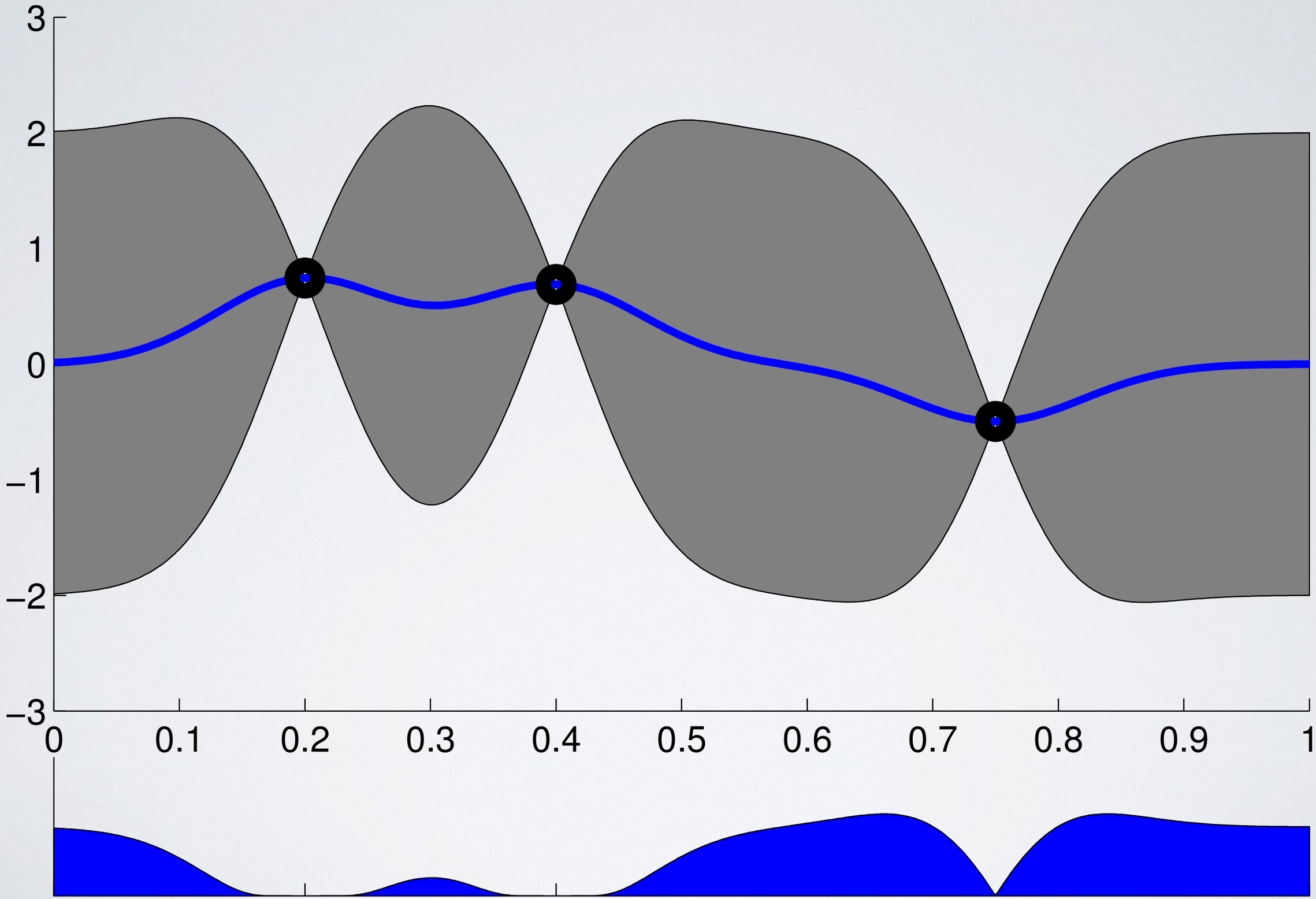
- GP Upper Confidence Bound (Srinivas et al. 2010):

$$a_{\mathsf{LCB}}(x) = \mu(x) - \kappa\,\sigma(x)$$

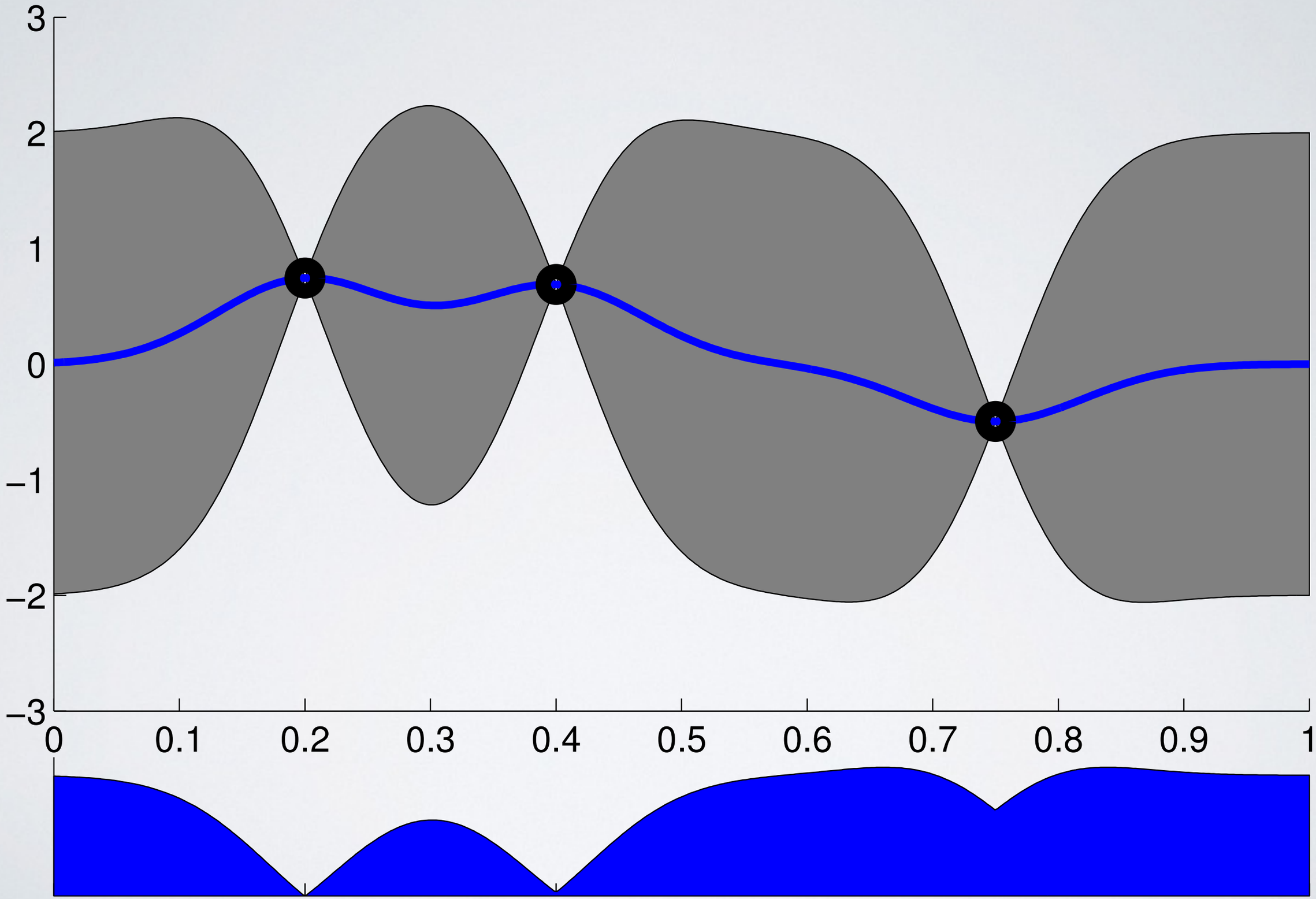
# Probability of Improvement



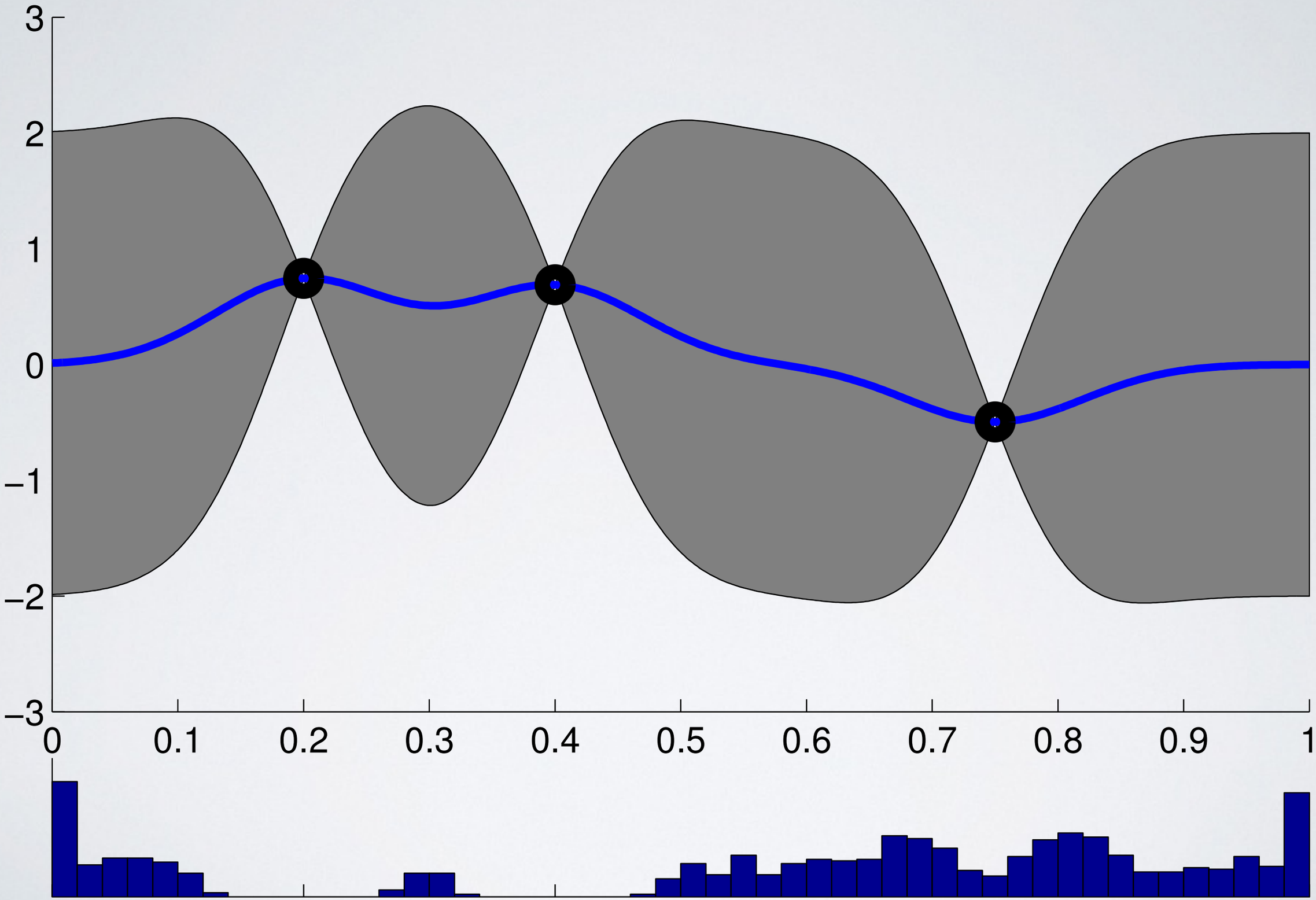
# Expected Improvement



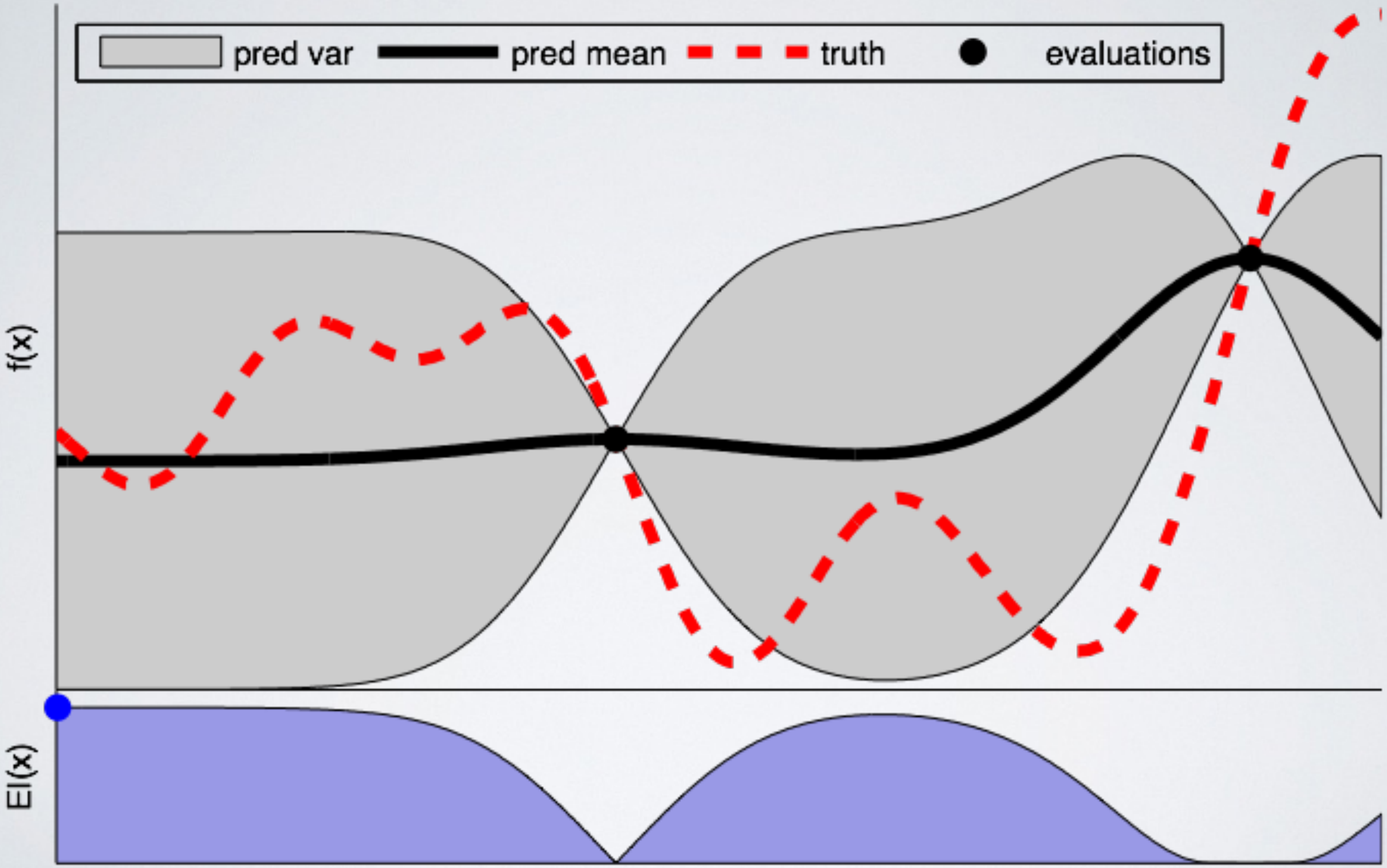
# GP Upper (Lower) Confidence Bound



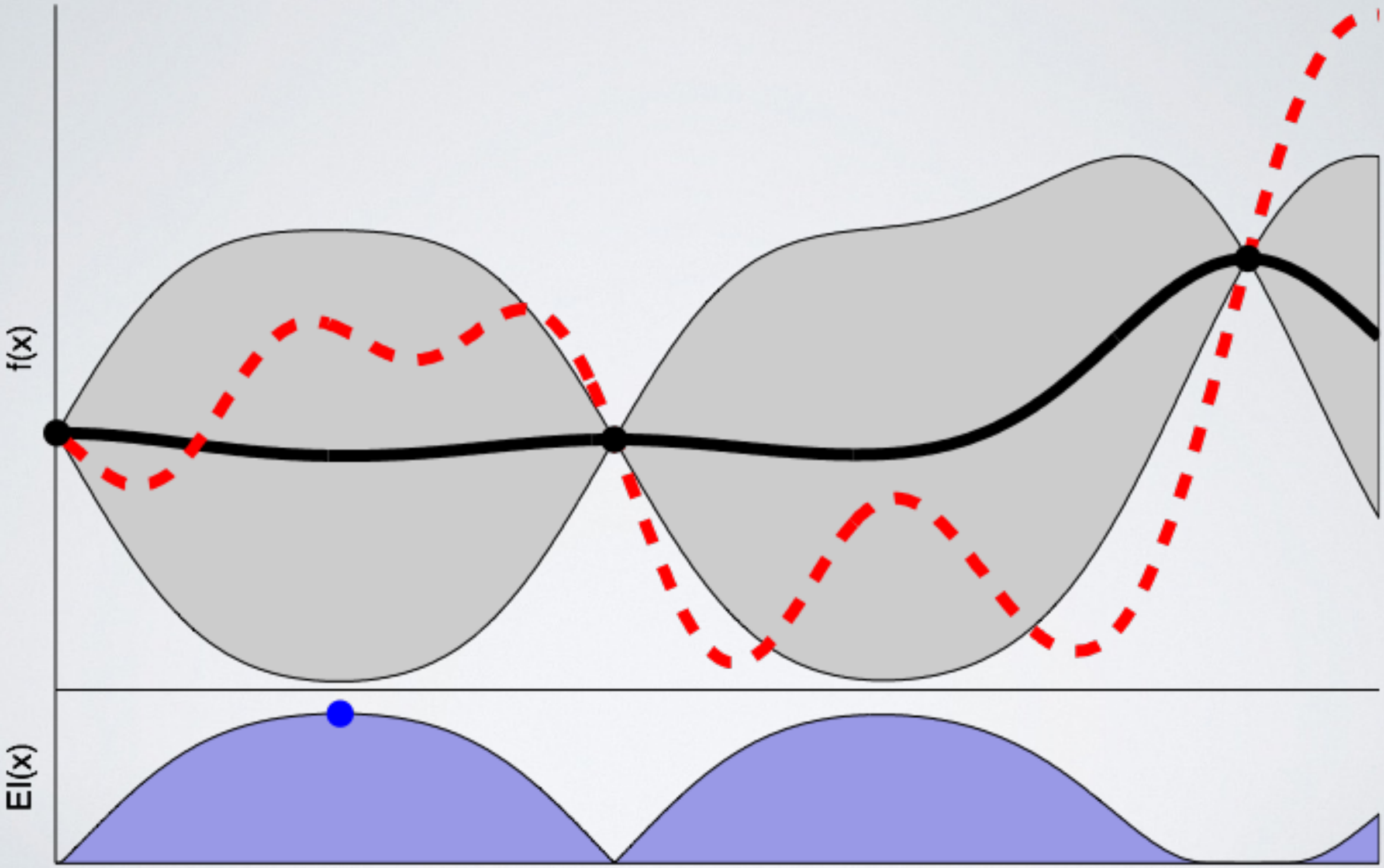
# Distribution Over Minimum (Entropy Search)



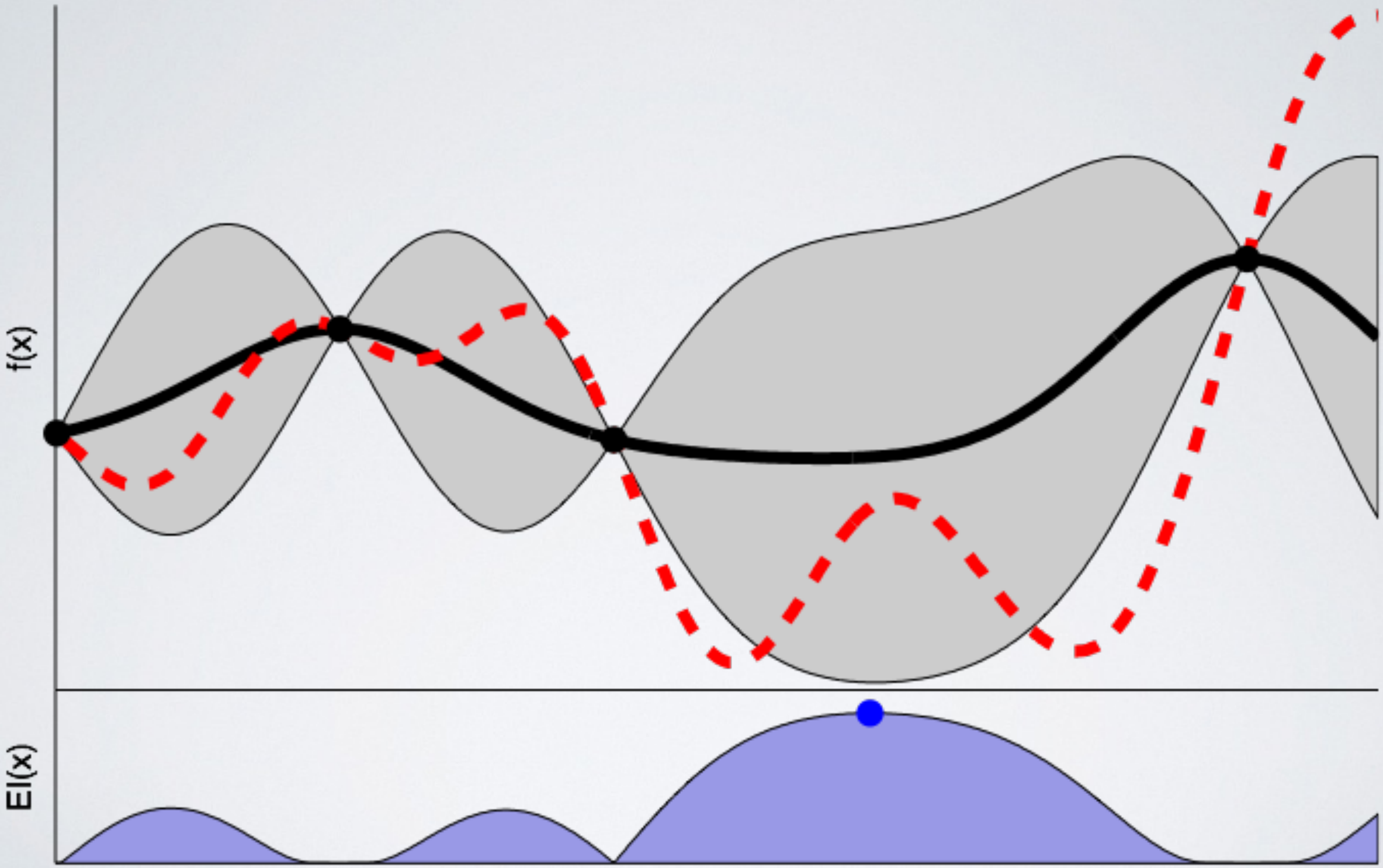
# Illustrating Bayesian Optimization



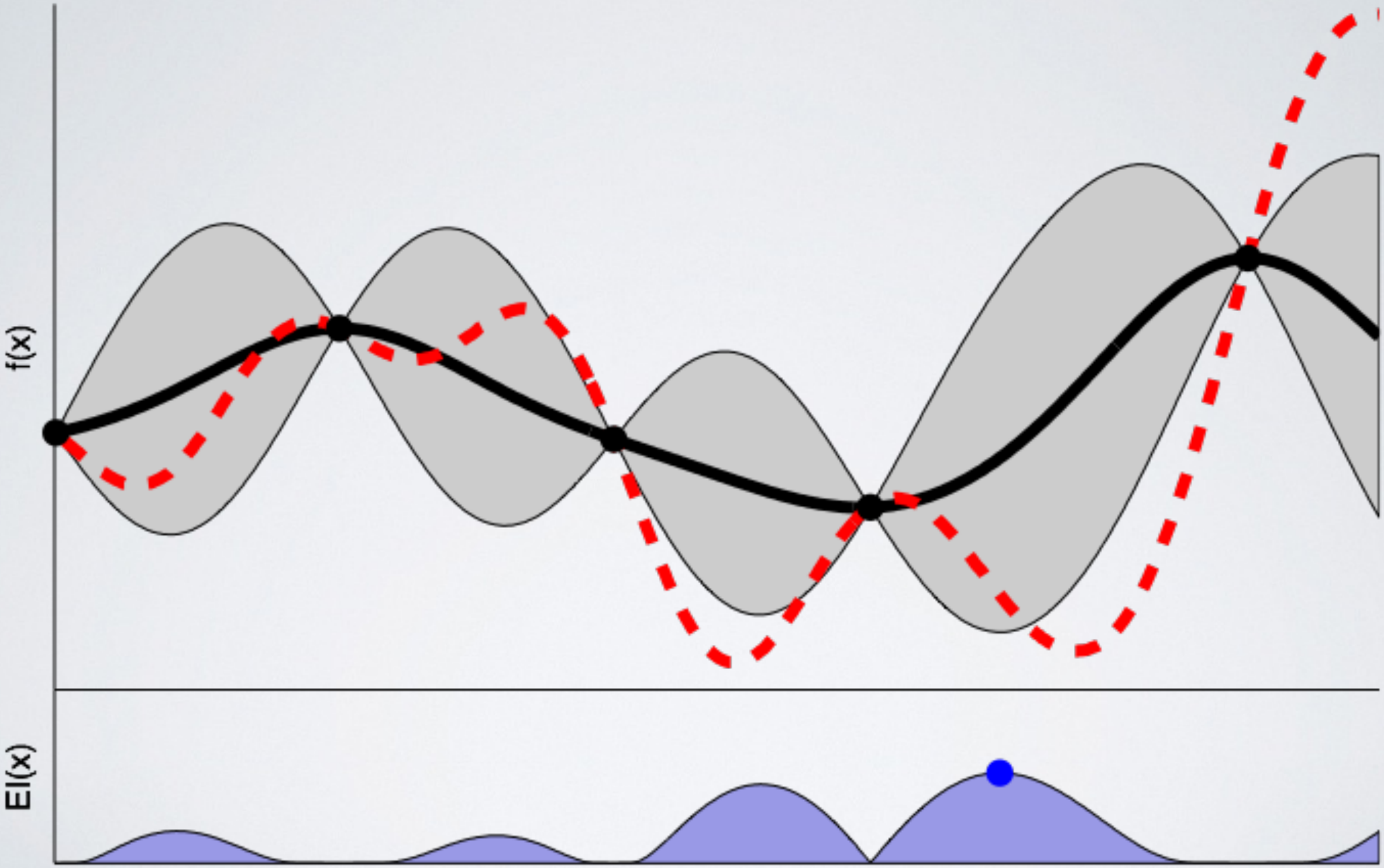
# Illustrating Bayesian Optimization



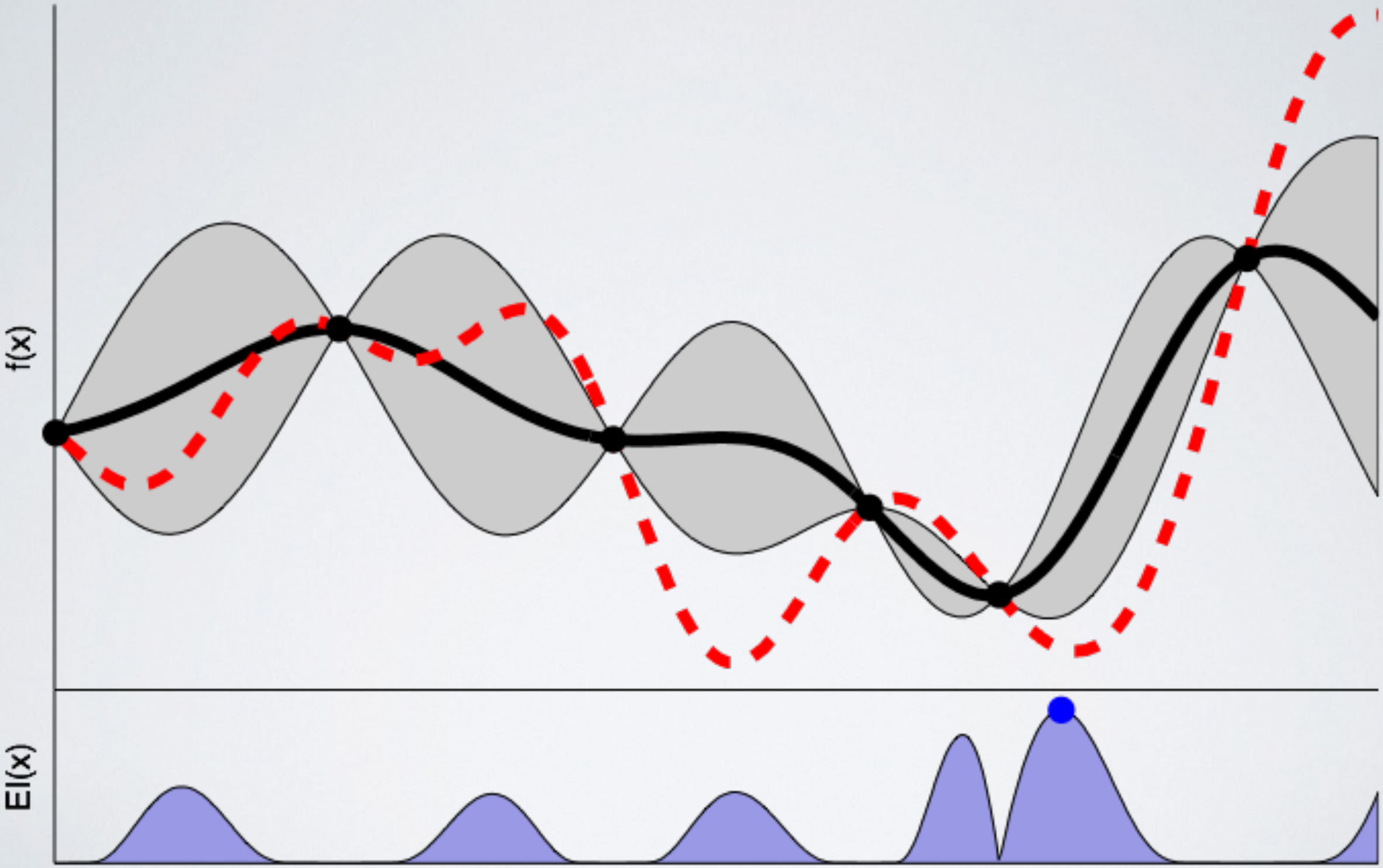
# Illustrating Bayesian Optimization



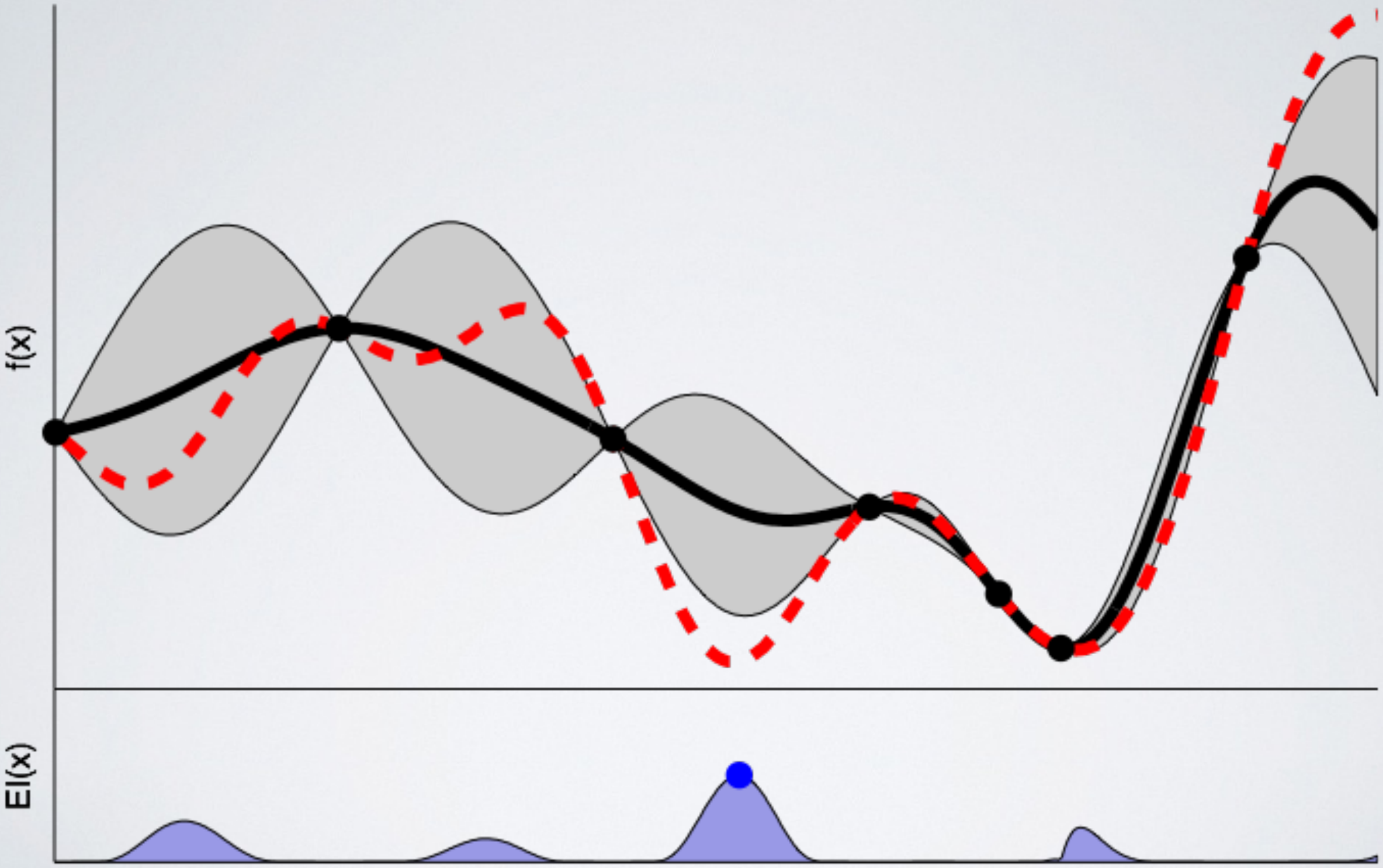
# Illustrating Bayesian Optimization



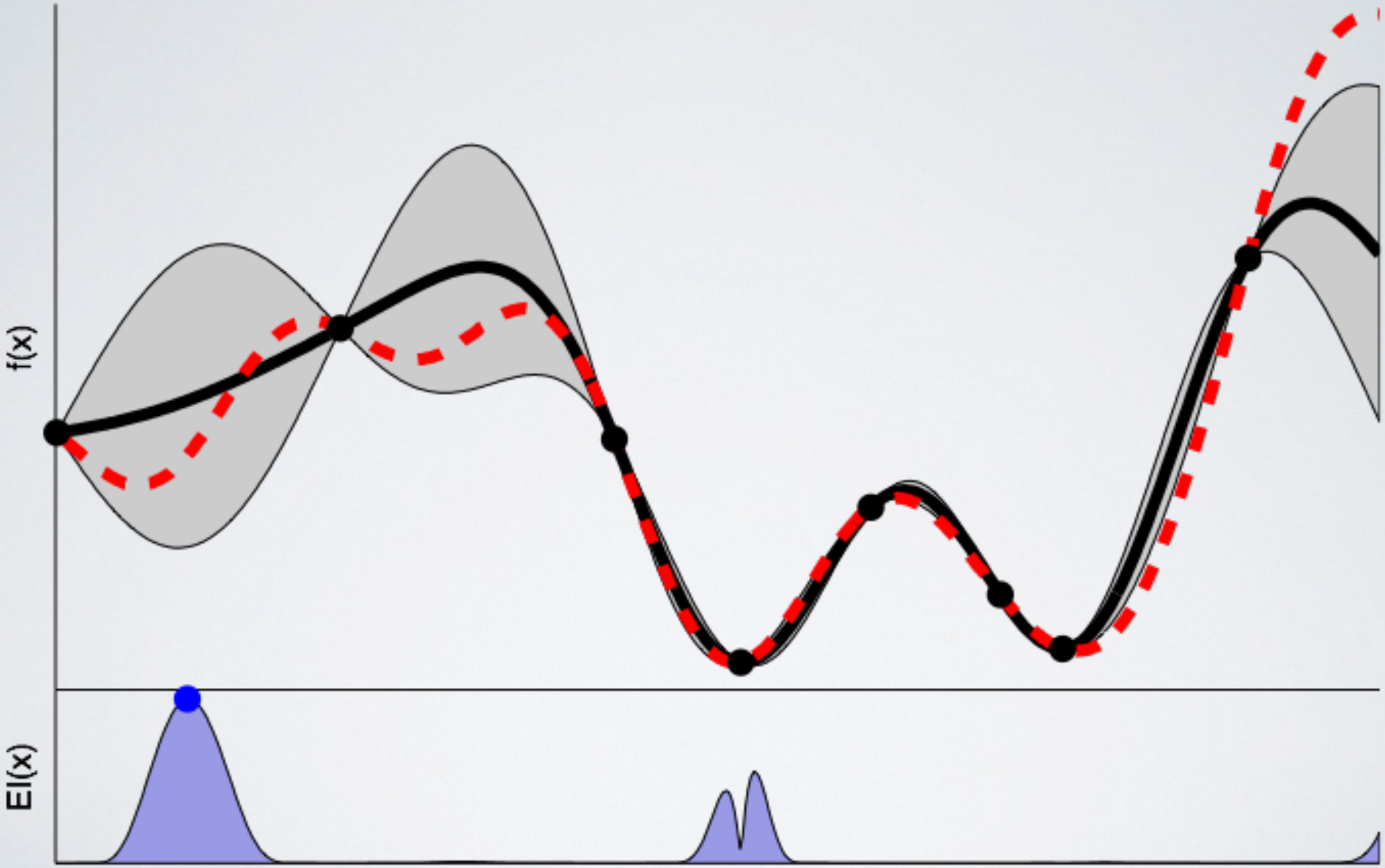
# Illustrating Bayesian Optimization



# 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.

# Fragility and Poor Default Choices

*Ironic Problem:*

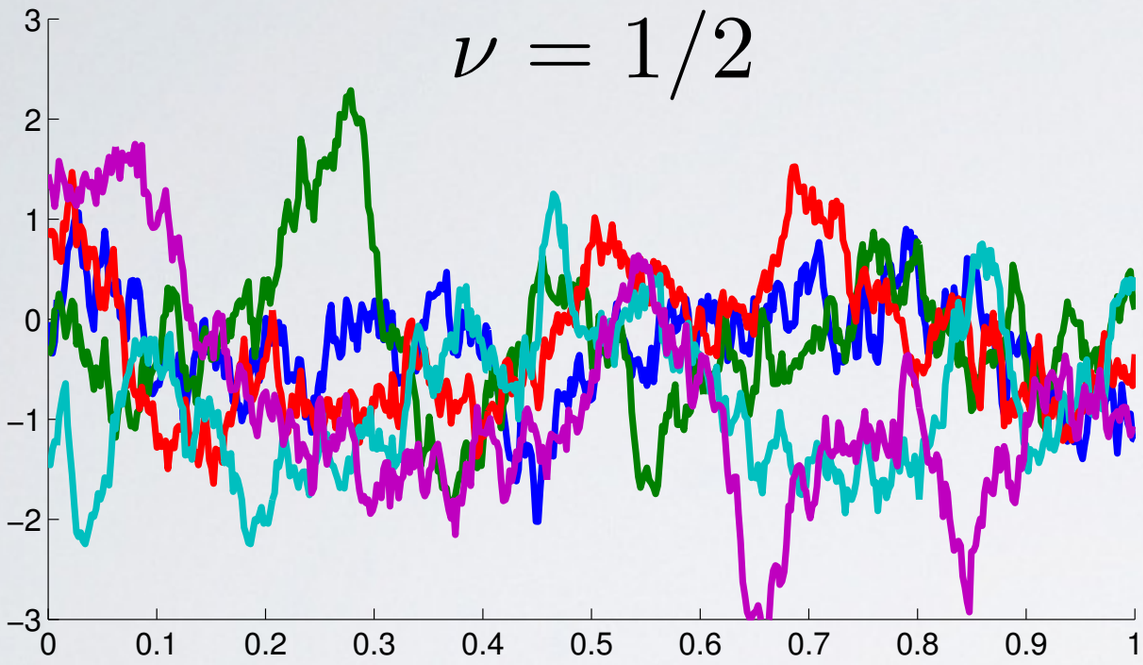
Bayesian optimization has its own hyperparameters!

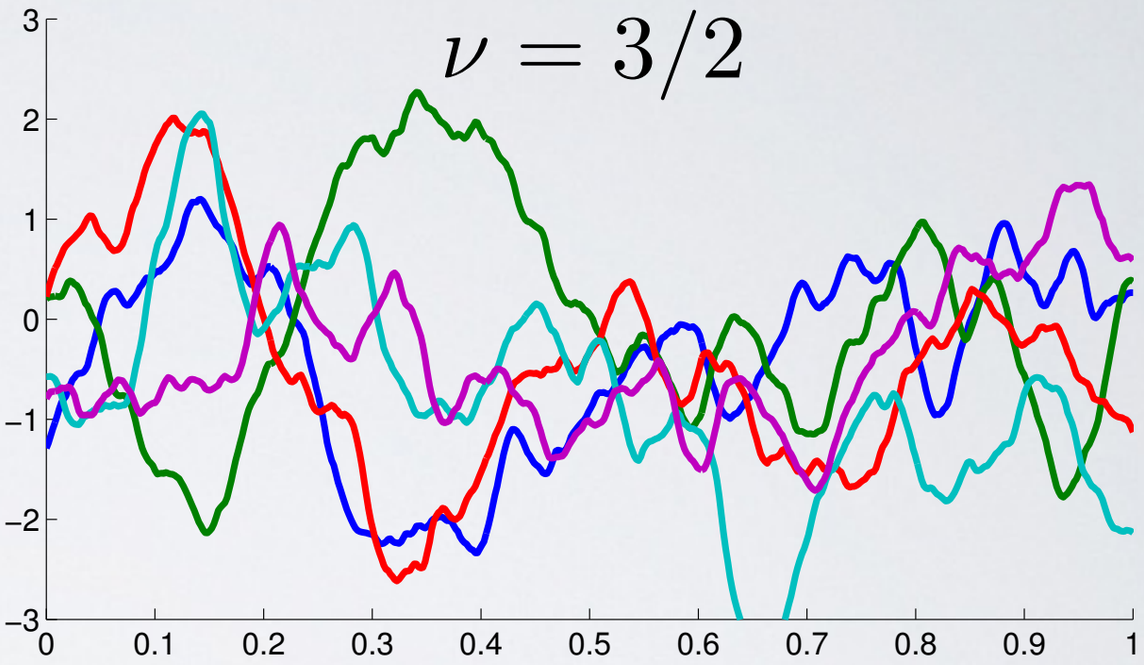
- **Covariance function selection**
  This turns out to be crucial to good performance.
  The default choice for regression is *way* too smooth.
  Instead: use adaptive Matèrn 3/5 kernel.

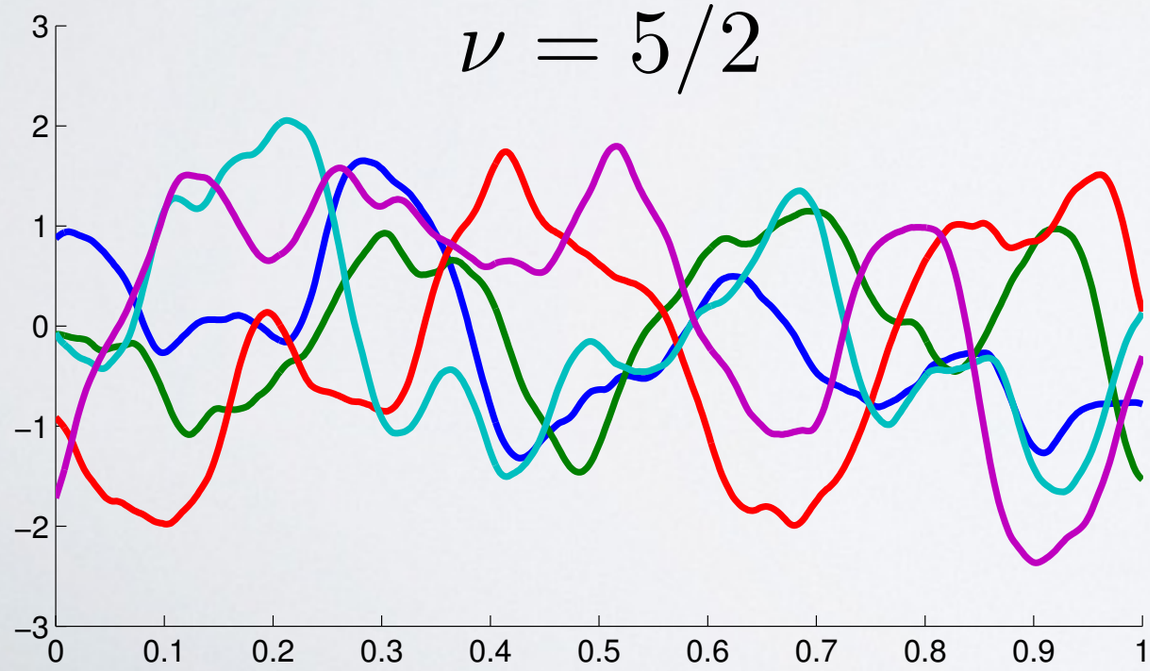
- **Gaussian process hyperparameters**
  Typical empirical Bayes approach can fail horribly.
  Instead: use Markov chain Monte Carlo integration.
  Slice sampling means no additional parameters!

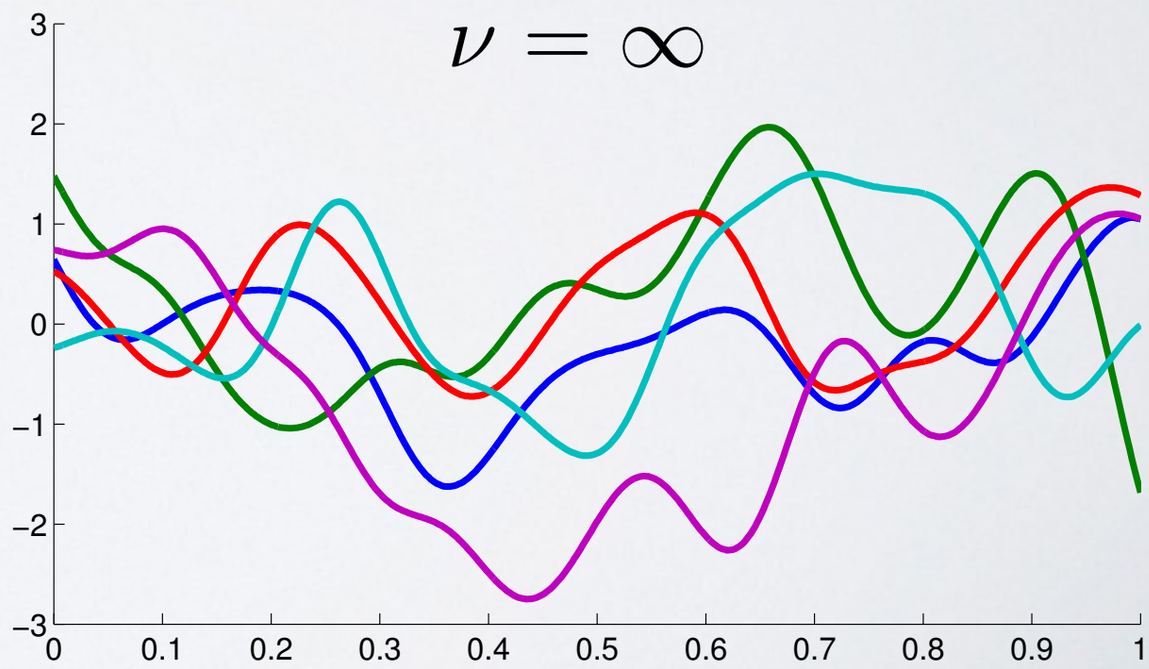
# Covariance Function Choice

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



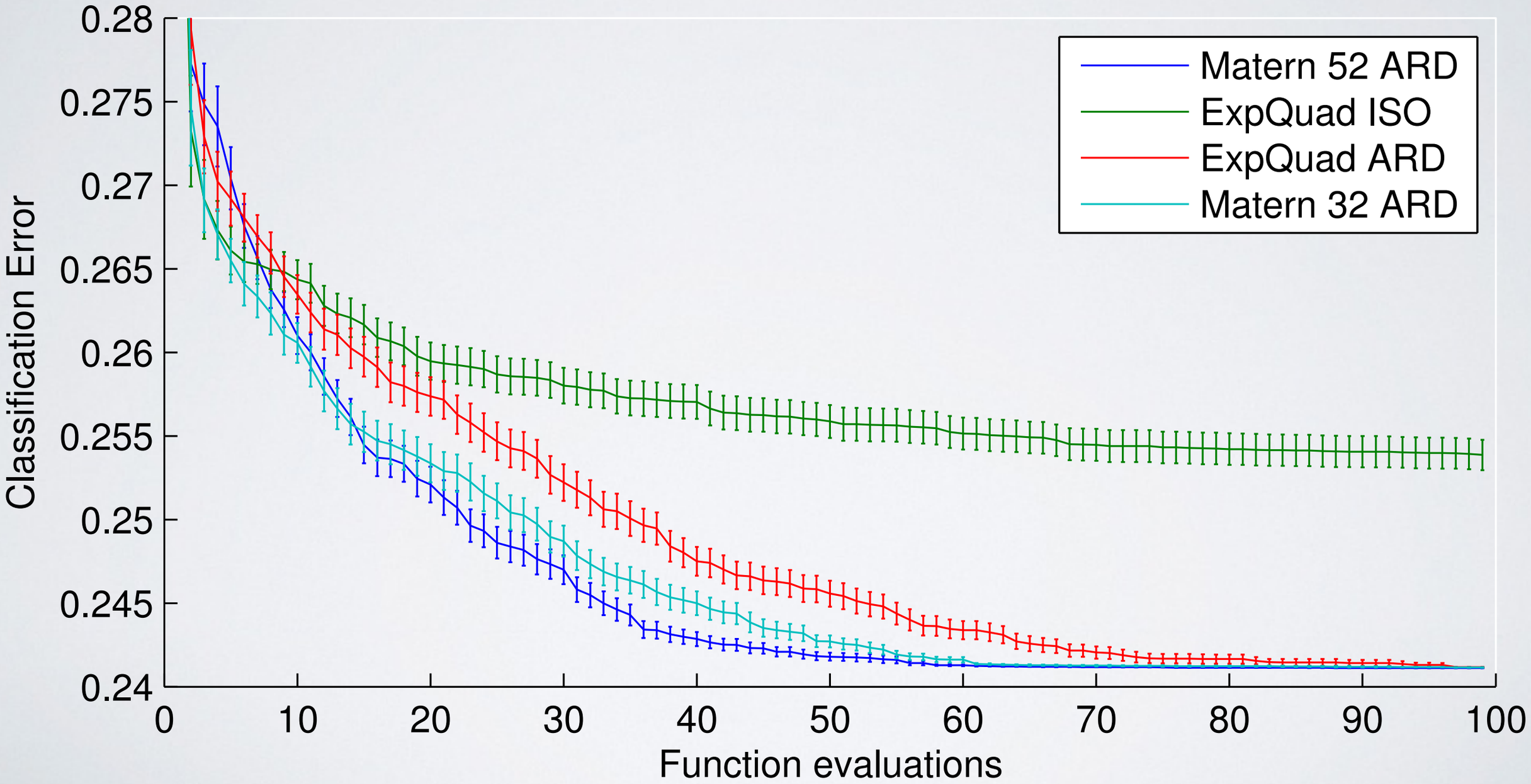



# Choosing Covariance Functions

## Structured SVM for Protein Motif Finding
## Miller et al (2012)




Snoek, Larochelle & RPA, NIPS 2012

# MCMC for GP Hyperparameters

- Covariance hyperparameters are often optimized rather than marginalized, typically in the name of convenience and efficiency.
- Slice sampling of hyperparameters (e.g., Murray and Adams 2010) is comparably fast and easy, but accounts for uncertainty in length scale, mean, and amplitude.
- Integrated Acquisition Function:

$$\hat{a}(x) = \int a(x\,;\,\theta)\, p(\theta \,|\, \{x_n, y_n\}_{n=1}^N)\, \mathrm{d}\theta$$

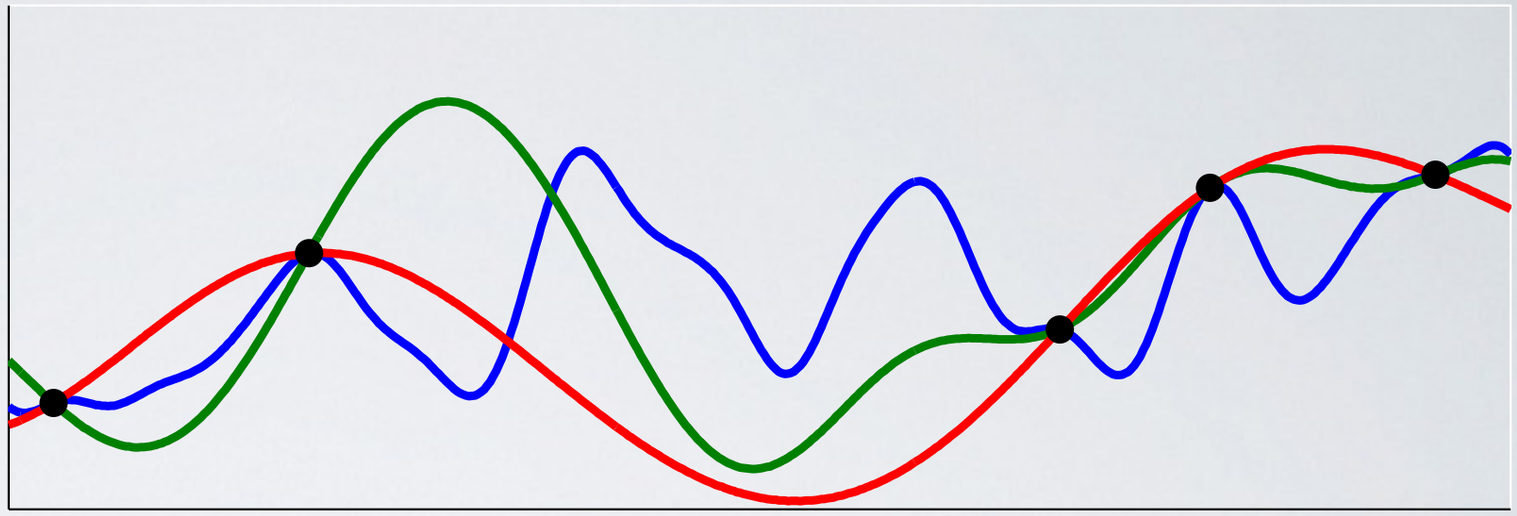
$$\approx \frac{1}{K} \sum_{k=1}^{K} a(x\,;\,\theta^{(k)}) \qquad \theta^{(k)} \sim p(\theta \,|\, \{x_n, y_n\}_{n=1}^N)$$

- For a theoretical discussion of the implications of inferring hyperparameters with BayesOpt, see recent work by Wang and de Freitas (http://arxiv.org/abs/1406.7758)

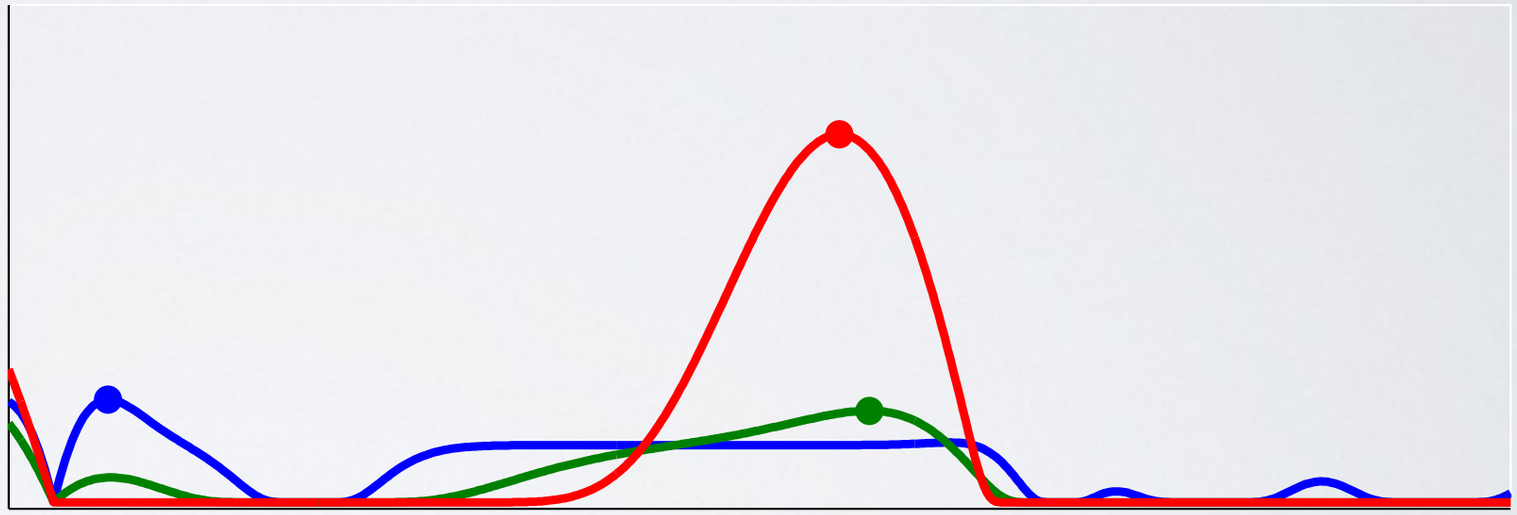
Snoek, Larochelle & RPA, NIPS 2012

# Integrating Out GP Hyperparameters

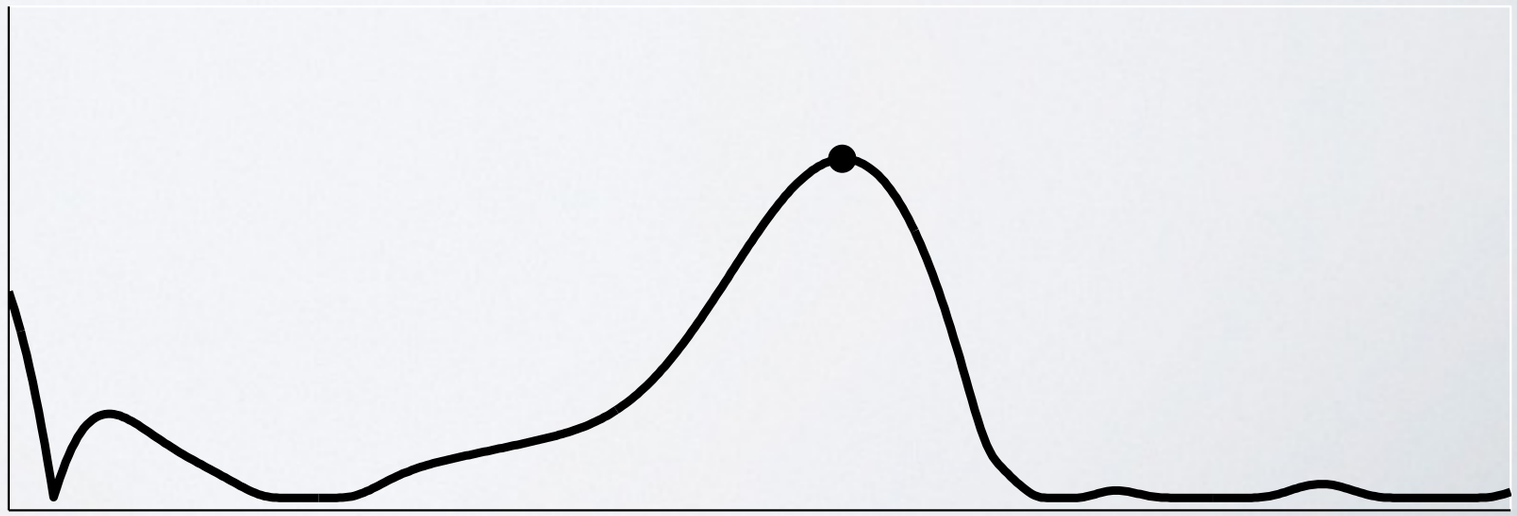
Posterior samples with three different length scales



Length scale specific expected improvement

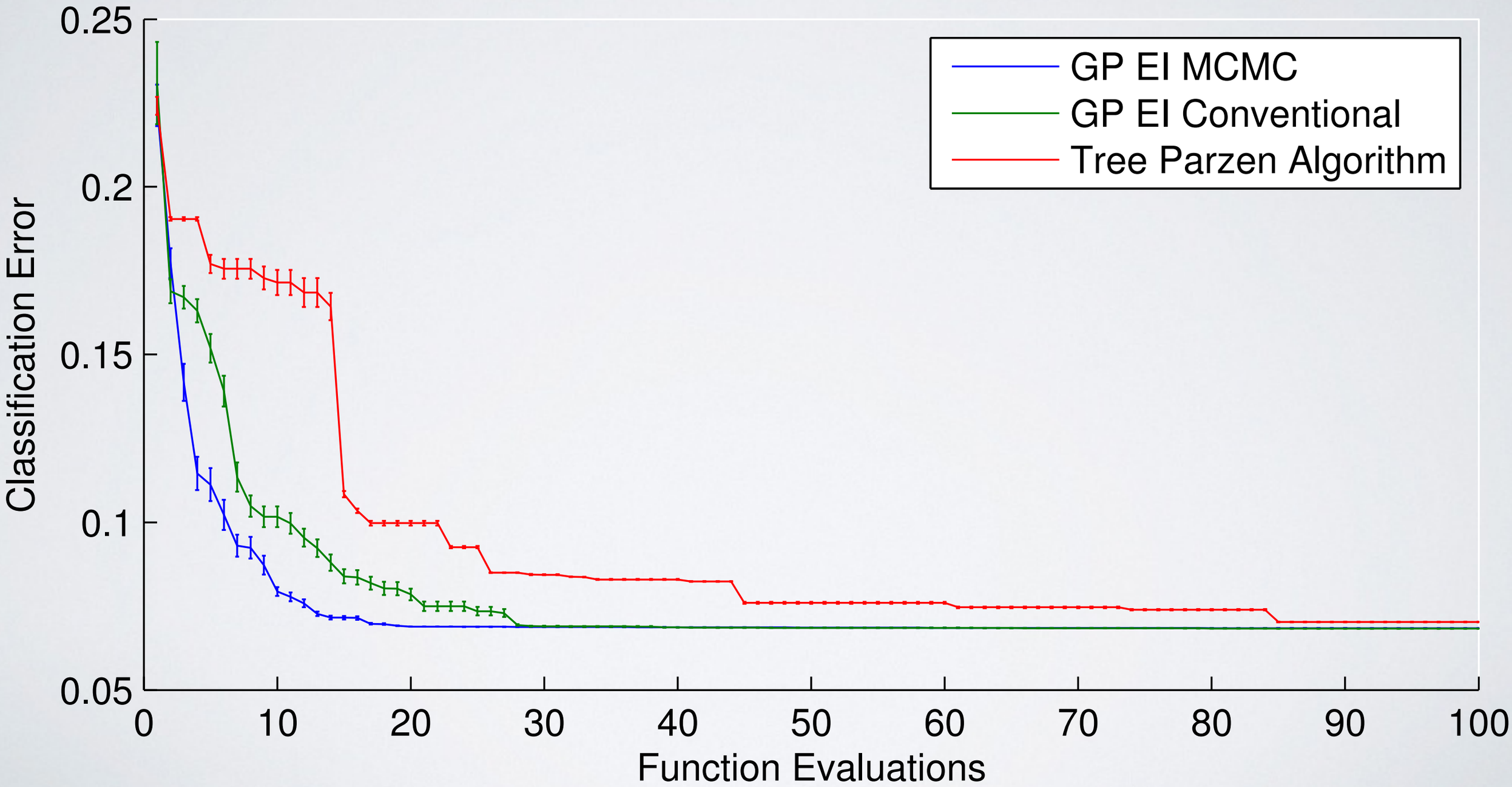


Integrated expected improvement



# MCMC for Hyperparameters

Logistic regression for handwritten digit recognition




Snoek, Larochelle & RPA, NIPS 2012

**The original source for these slides has advanced topics beyond this point. If interested please reference:**

**https://www.iro.umontreal.ca/~bengioy/cifar/NCAP2014-summerschool/slides/Ryan_adams_140814_bayesopt_ncap.pdf**