# Automatic Hyper-parameter Tuning via Bayesian Optimization

**Janardhan Rao (Jana) Doppa**

School of EECS, Washington State University

# Motivation

- **Every machine learning algorithm has hyper-parameters**

  - **Perceptron algorithm:** learning rate, number of iterations
  - **SVM algorithm:** tradeoff parameter C, kernel parameter (degree for polynomial kernel, width for Gaussian kernel)
  - **Random Forests:** number of trees, depth of each tree
  - **Deep neural networks:** number of layers, weight regularization, layer size, which non-linearity, batch size, learning rate schedule, stopping conditions etc.
  - …

# Search for Good Hyper-parameters: Standard Practice

- **Define an objective function**
  - We care about generalization performance. Use cross-validation to measure parameter quality.
- **How do people currently search? Black magic.**
  - Grid search
  - Random search
  - Graduate student descent
- **What is wrong?**
  - Painful!
  - Computationally expensive – many training cycles
  - Possibly sub-optimal

# Bayesian Optimization: Key Ideas

- **Build a surrogate statistical model** based on past computational experiments
  - Assumption is that surrogate model is cheap to evaluate
- **Intelligently select the next experiment** (candidate solution) **using the statistical model**
  - Trade-off exploration and exploitation
  - Exploration corresponds to selecting candidates for which the statistical model is not confident (high variance)
  - Exploitation corresponds to selecting candidates for which the statistical model is highly confident (high mean)

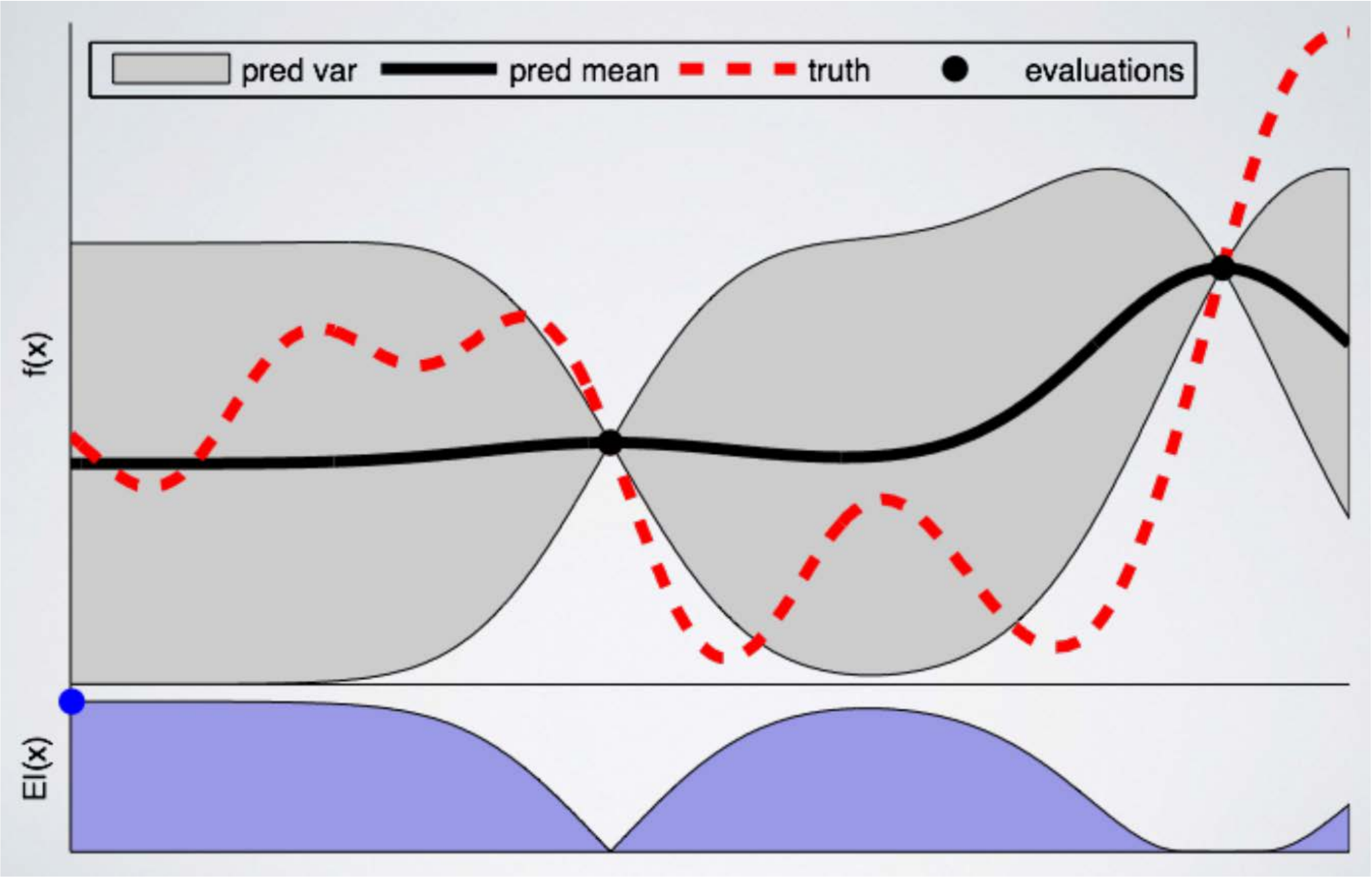
# Bayesian Optimization Framework: Key Elements

- **Surrogate statistical model**
  - Cheap to evaluate
  - Can quantify uncertainty of predictions (i.e., variance)
- **Acquisition function**
  - Scores candidate solutions (via mean and variance obtained from the statistical model) in terms of their usefulness
- **Optimizer**
  - Select the candidate that maximizes the acquisition function
  - Next candidate we should try

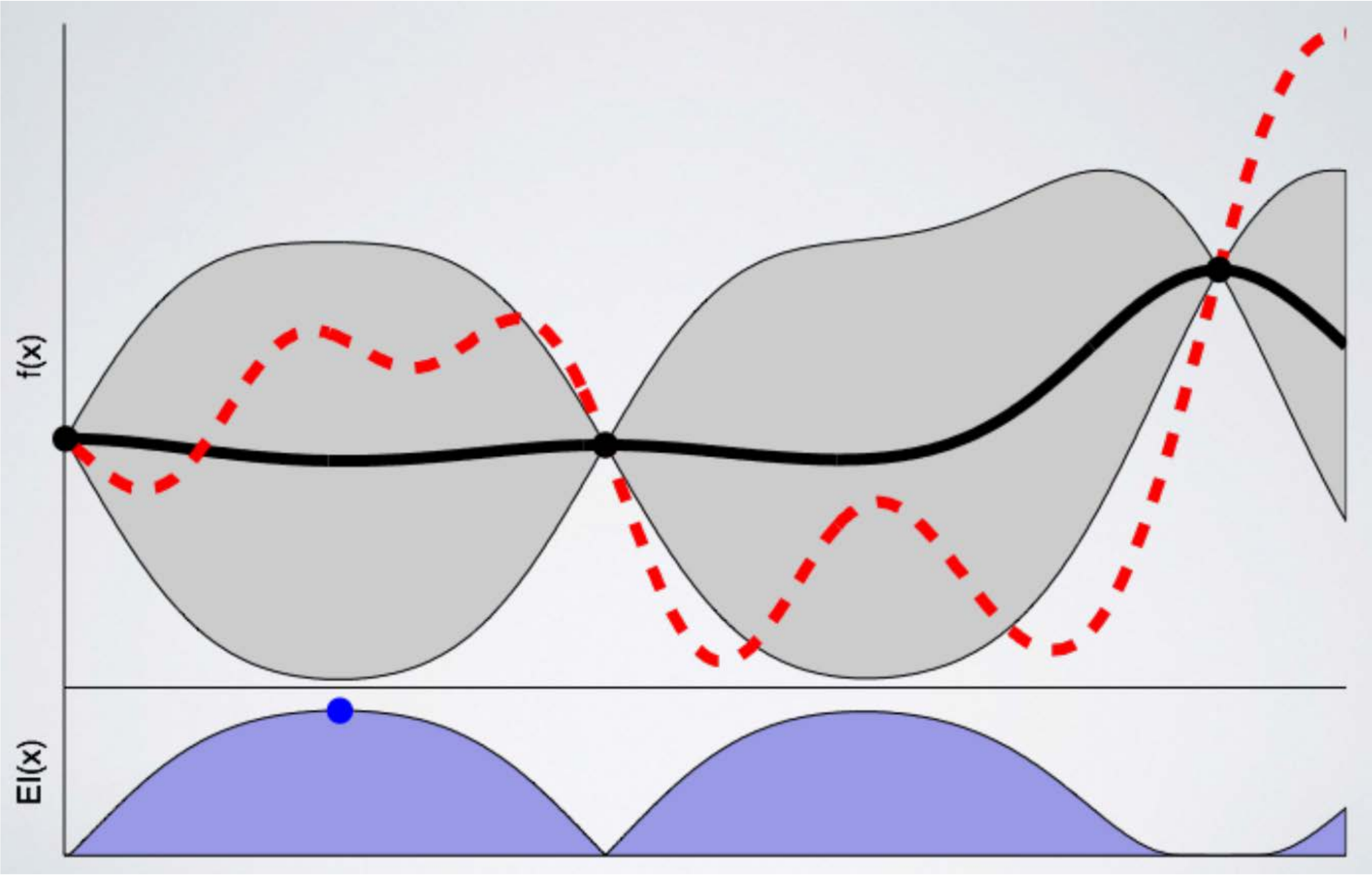
# Bayesian Optimization Framework: High-level Overview

- Initialize statistical model *F*
- Repeat the following steps for several iterations
  - Select the next candidate (say *x*) by optimizing the acquisition function *A(x)*
  - Run experiment with candidate *x* to compute its quality *y*
  - Update the statistical model *F* based on the new training example *(x, y)*
  - Update the best uncovered solution so far (say *x_{best}*)

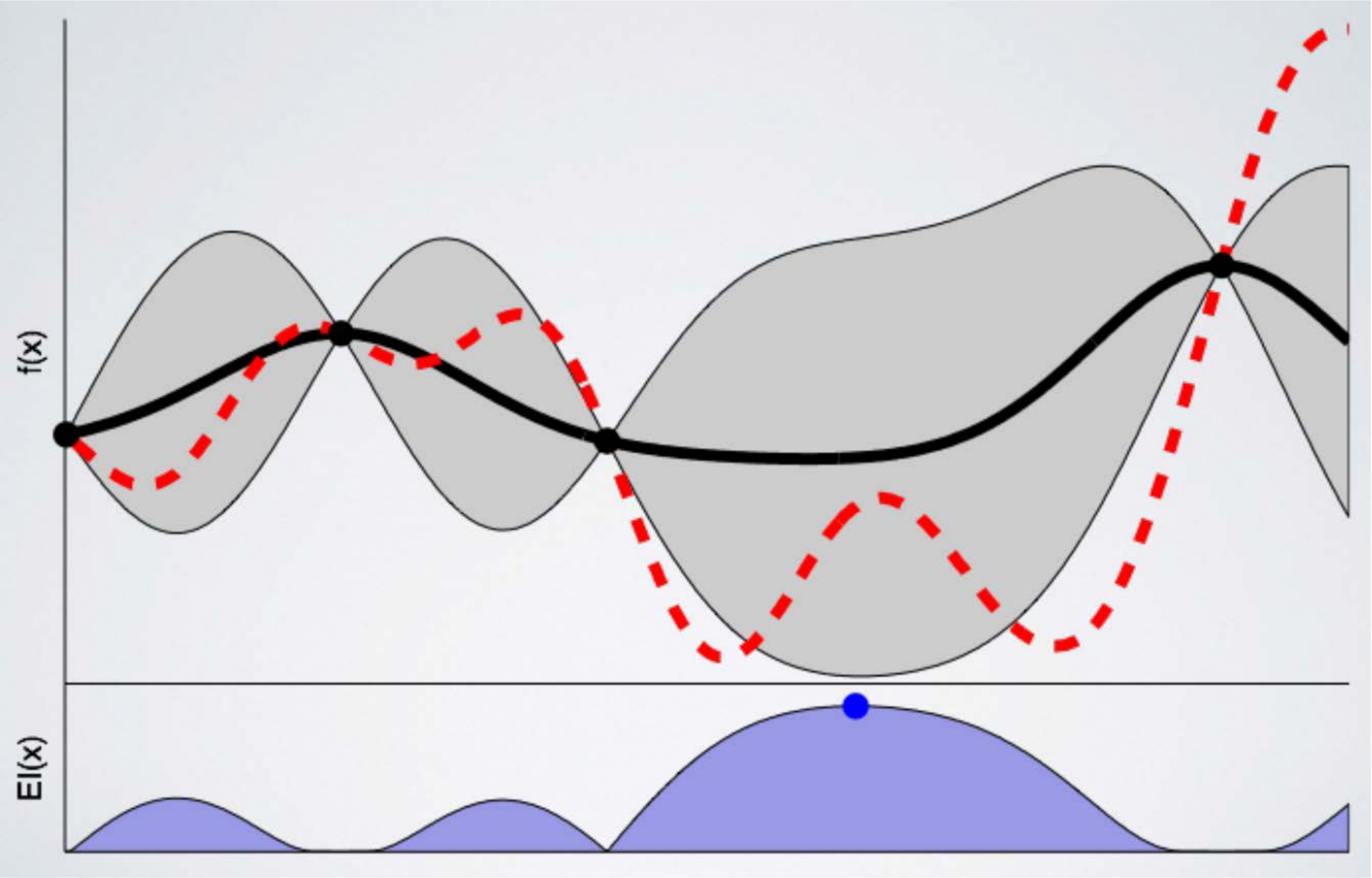
# Bayesian Optimization: Illustration



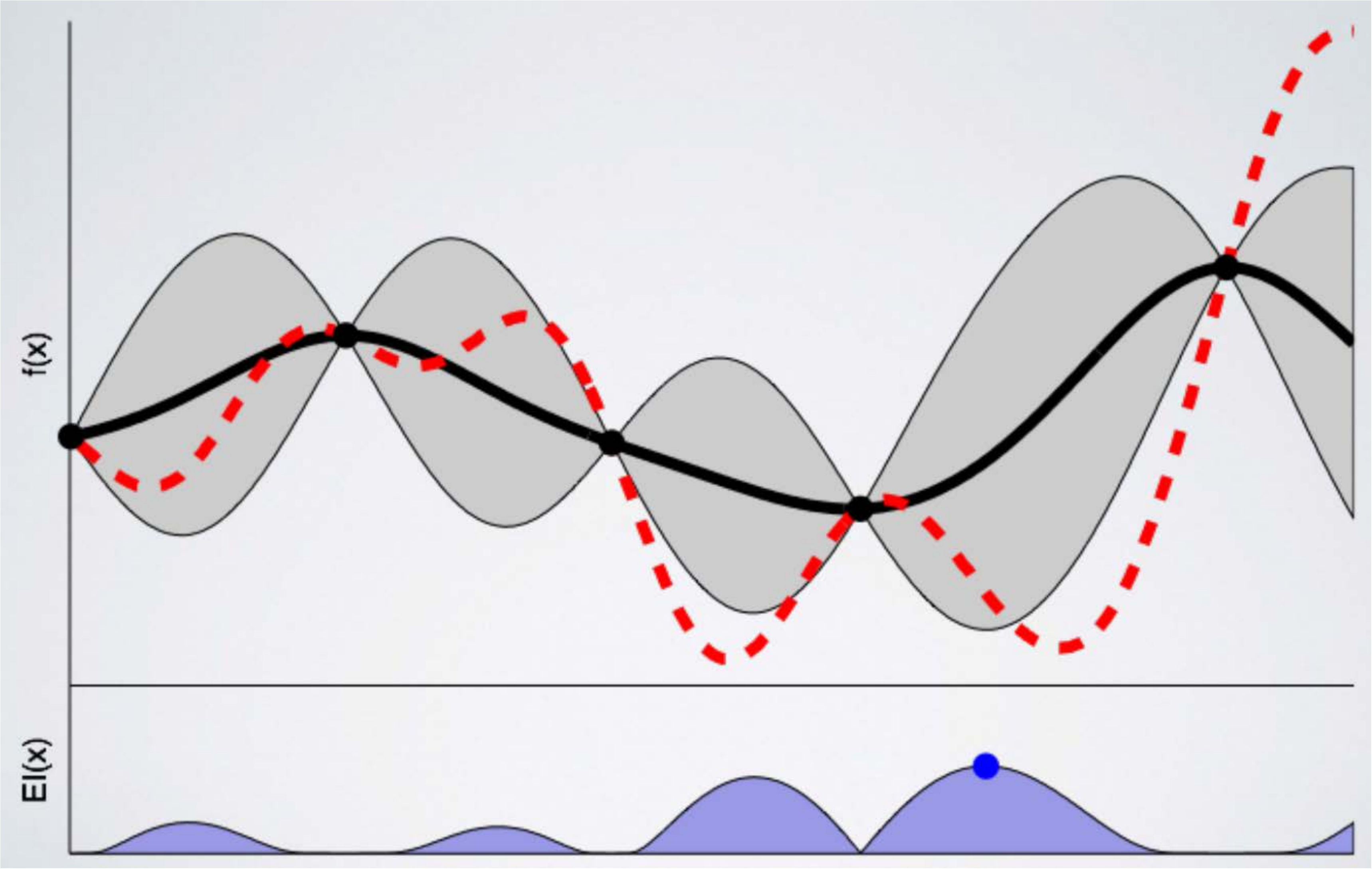
# Bayesian Optimization: Illustration



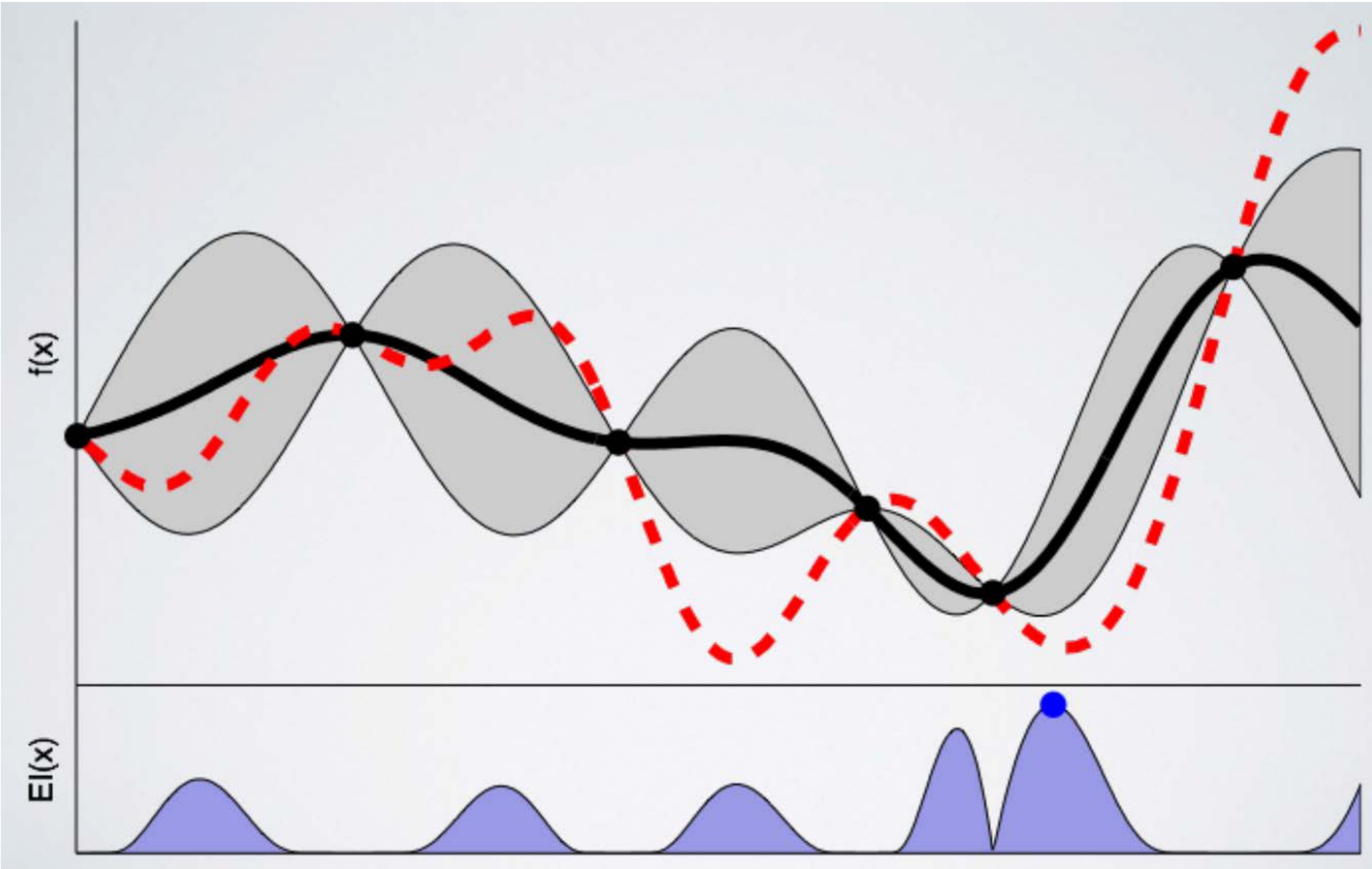
# Bayesian Optimization: Illustration



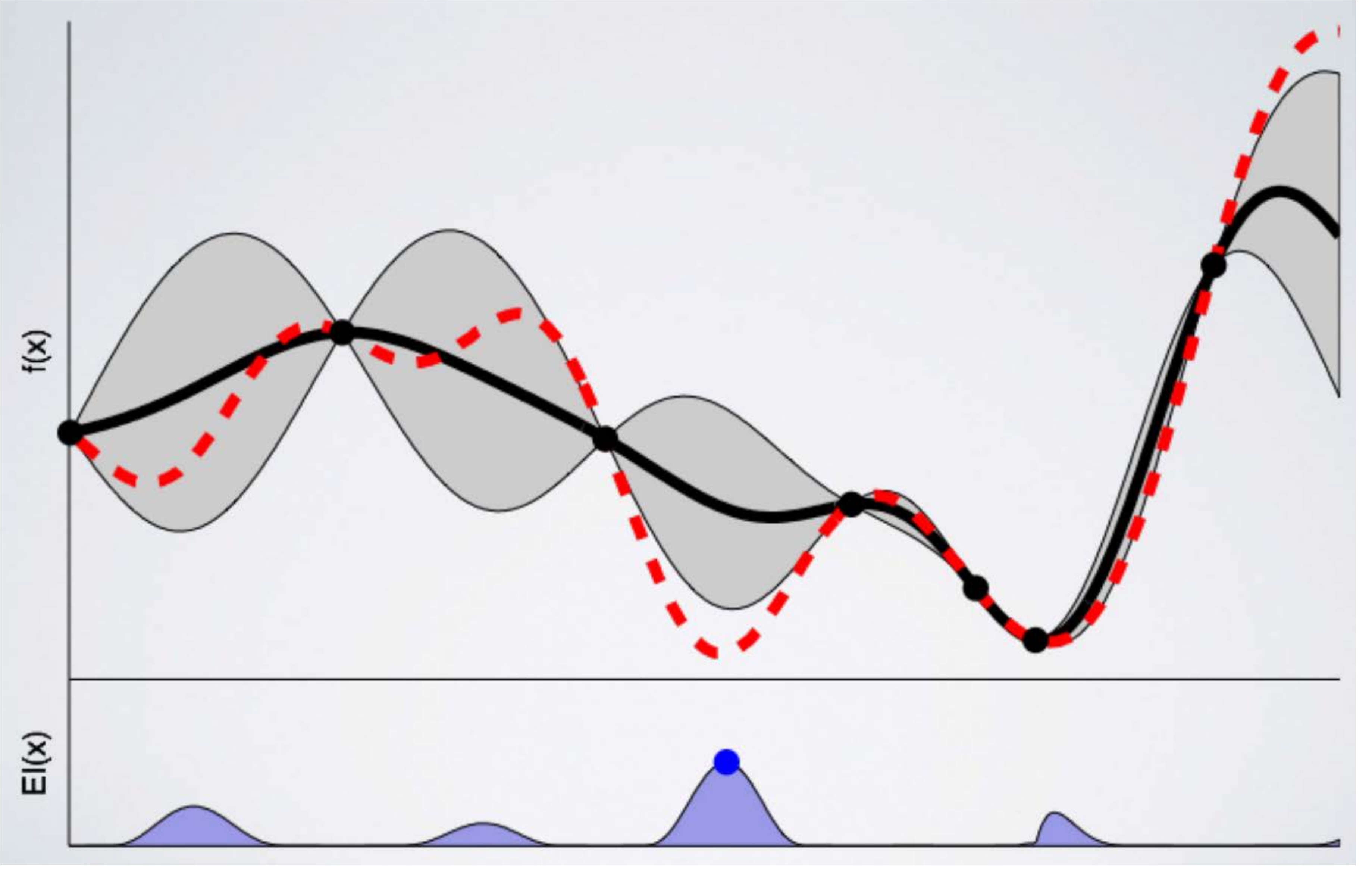
# Bayesian Optimization: Illustration



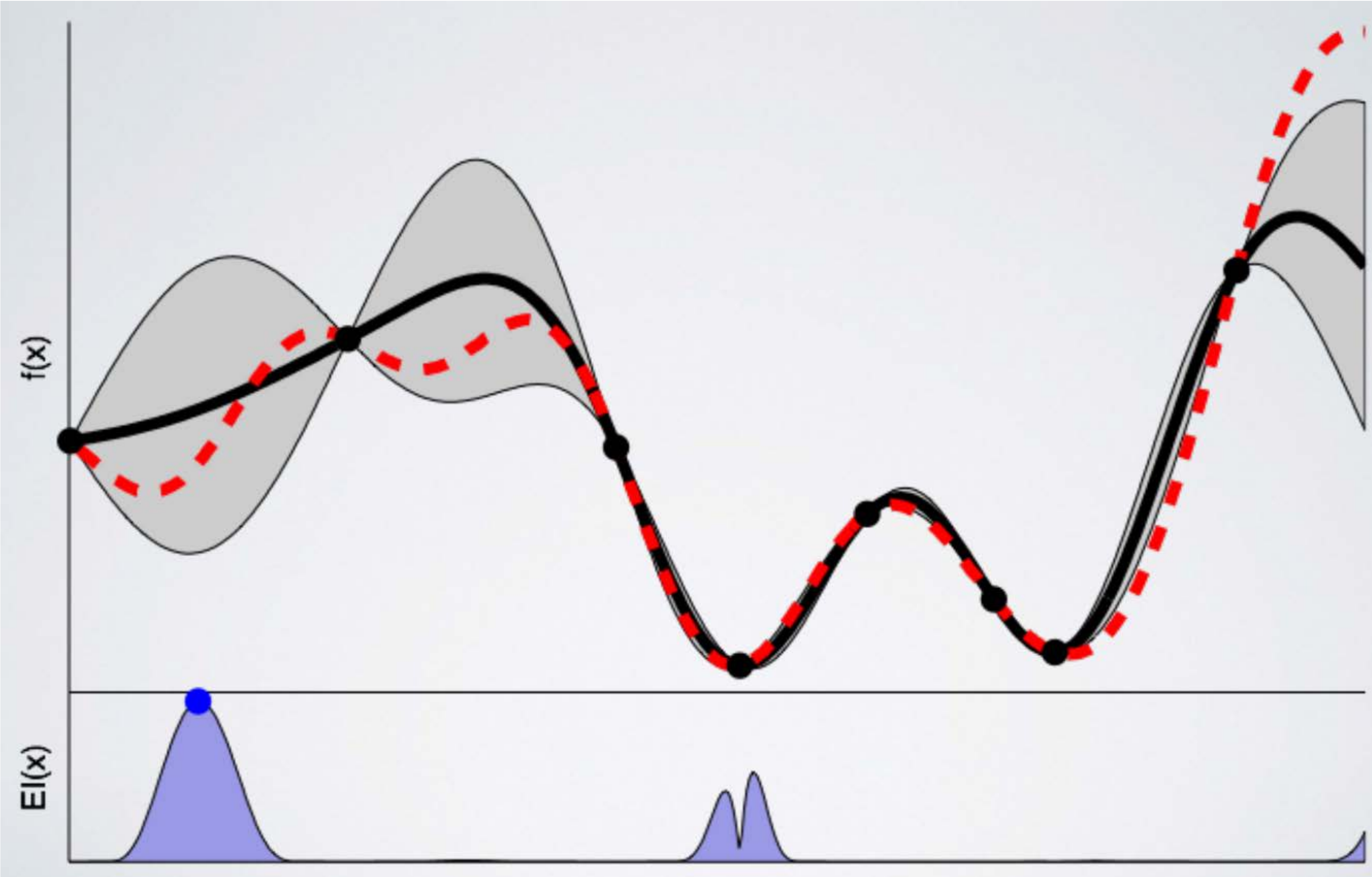
# Bayesian Optimization: Illustration



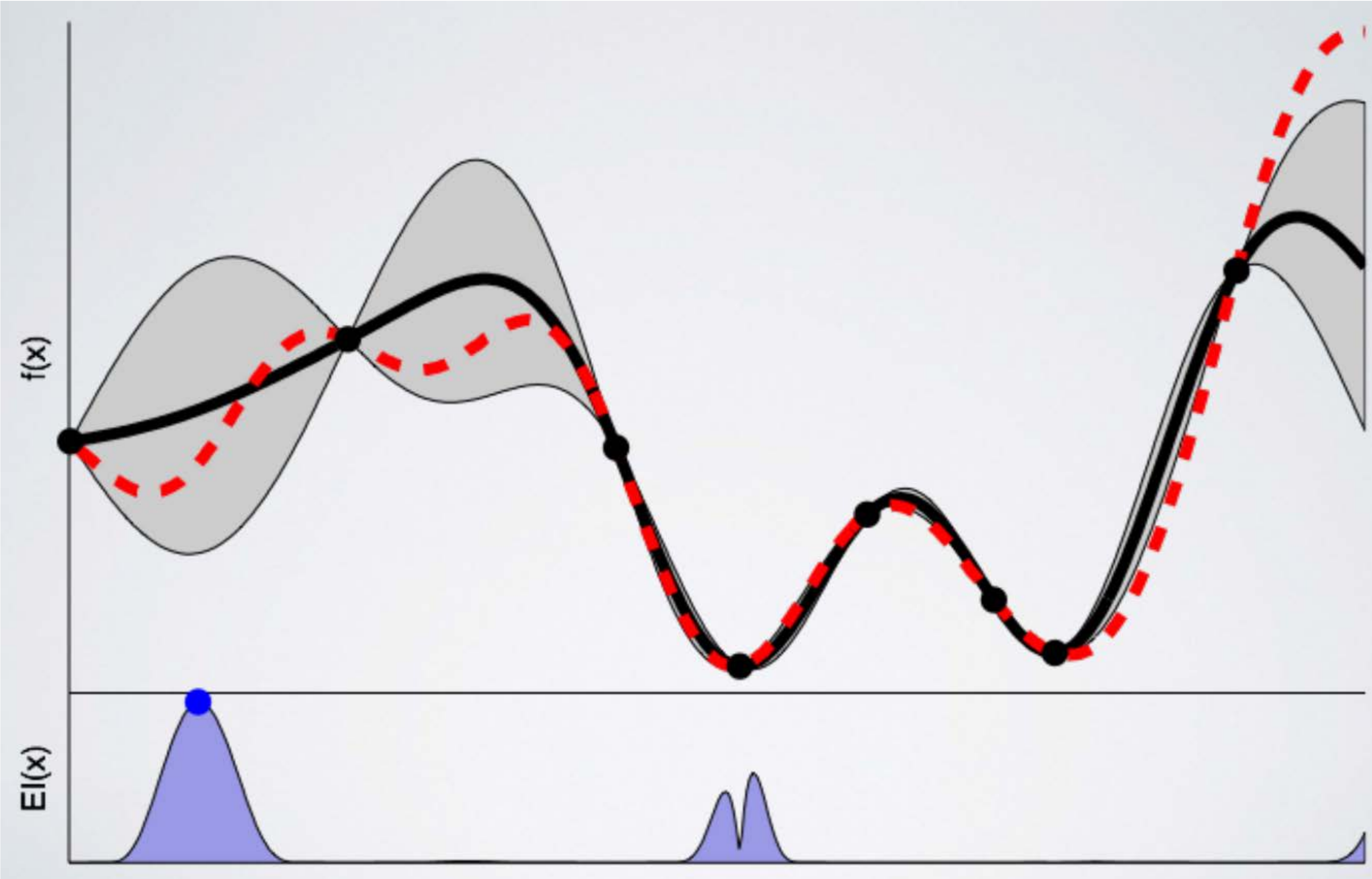
# Bayesian Optimization: Illustration



# Bayesian Optimization: Illustration



# Bayesian Optimization: Illustration



# Bayesian Optimization Framework: Key Elements

- **Surrogate statistical model**
  - Cheap to evaluate
  - Can quantify uncertainty of predictions (i.e., variance)
- **Acquisition function**
  - Scores candidate solutions (via mean and variance obtained from the statistical model) in terms of their usefulness
- **Optimizer**
  - Select the candidate that maximizes the acquisition function
  - Next candidate we should try

# Surrogate Model: Choices

- **Bayesian Regression Model**
- **Gaussian Process (GP) Model**
  - Very popular
- **Random Forests**
  - A bunch of regression trees
  - Mean = empirical mean of the predictions from all trees
  - Variance = empirical variance of the predictions from all trees
  - Simple implement
  - Often works well. You can try Mondrian Forests (reason about uncertainty) for a better choice
  - Becoming very popular lately (see SMAC software)

# Acquisition Function: Choices

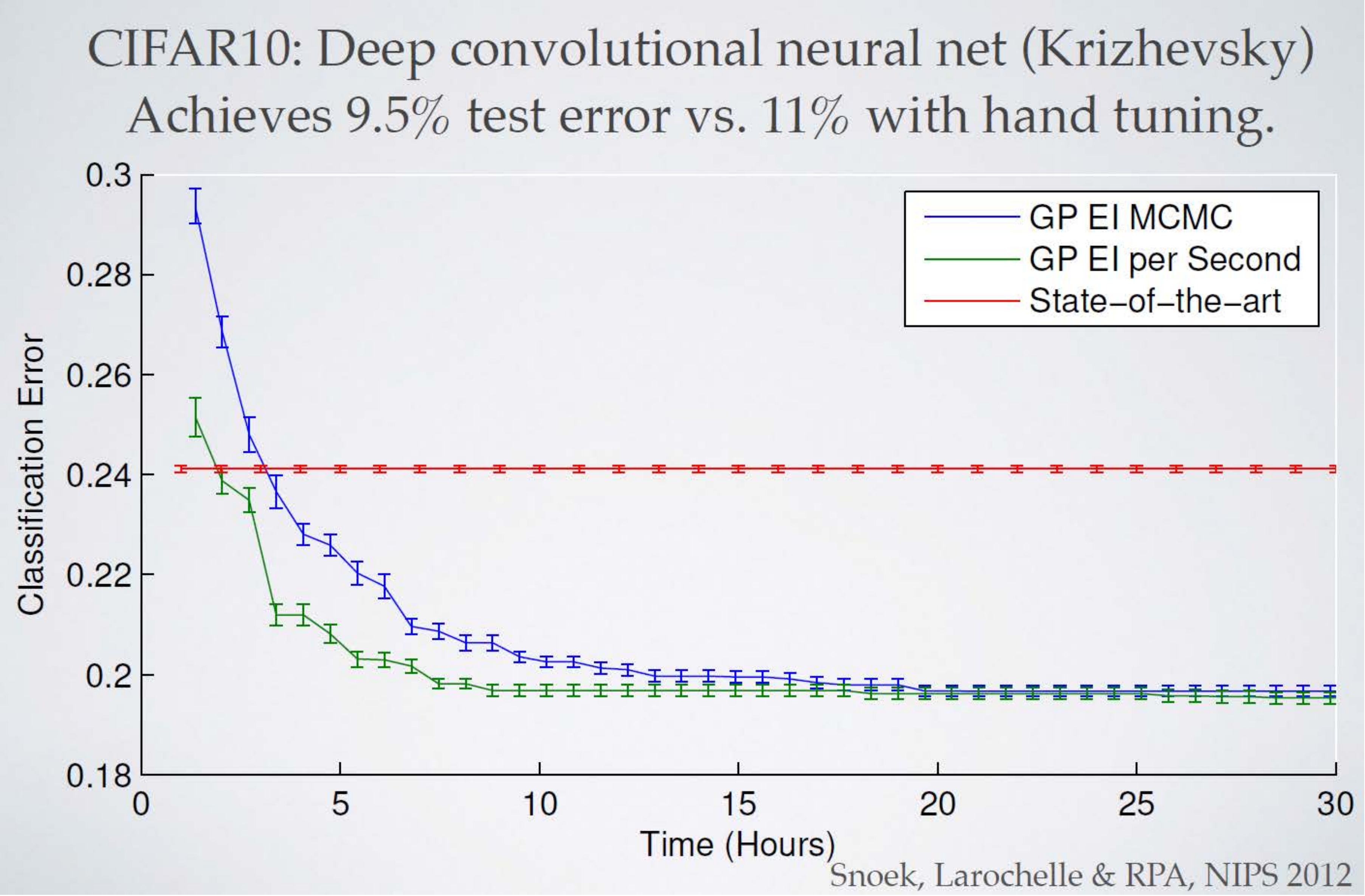
- Expected Improvement (EI)
- Probability of Improvement
- Upper Confidence Bound (UCB)

- EI and UCB are the most popular acquisition functions

# Optimizer: Choices

- Gradient descent with random restarts
- LBFGS
- DIvided RECTangles (DIRECT) algorithm
- Simultaneous Optimistic Optimization (SOO) algorithm
- …

# Example Results



# Papers and Software

- Jasper Snoek, Hugo Larochelle, Ryan P. Adams: Practical Bayesian Optimization of Machine Learning Algorithms. NIPS 2012: 2960-2968
  - https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf
- Spearmint Software
  - https://github.com/JasperSnoek/spearmint