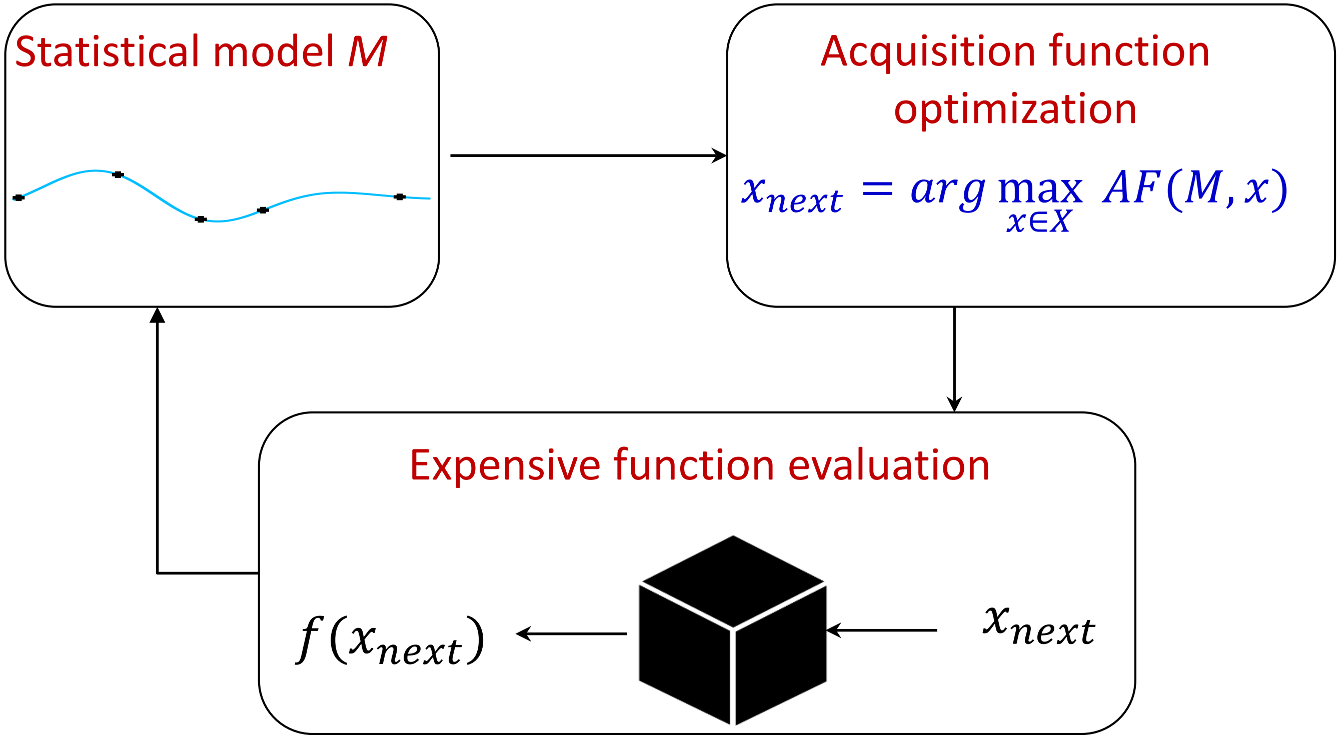
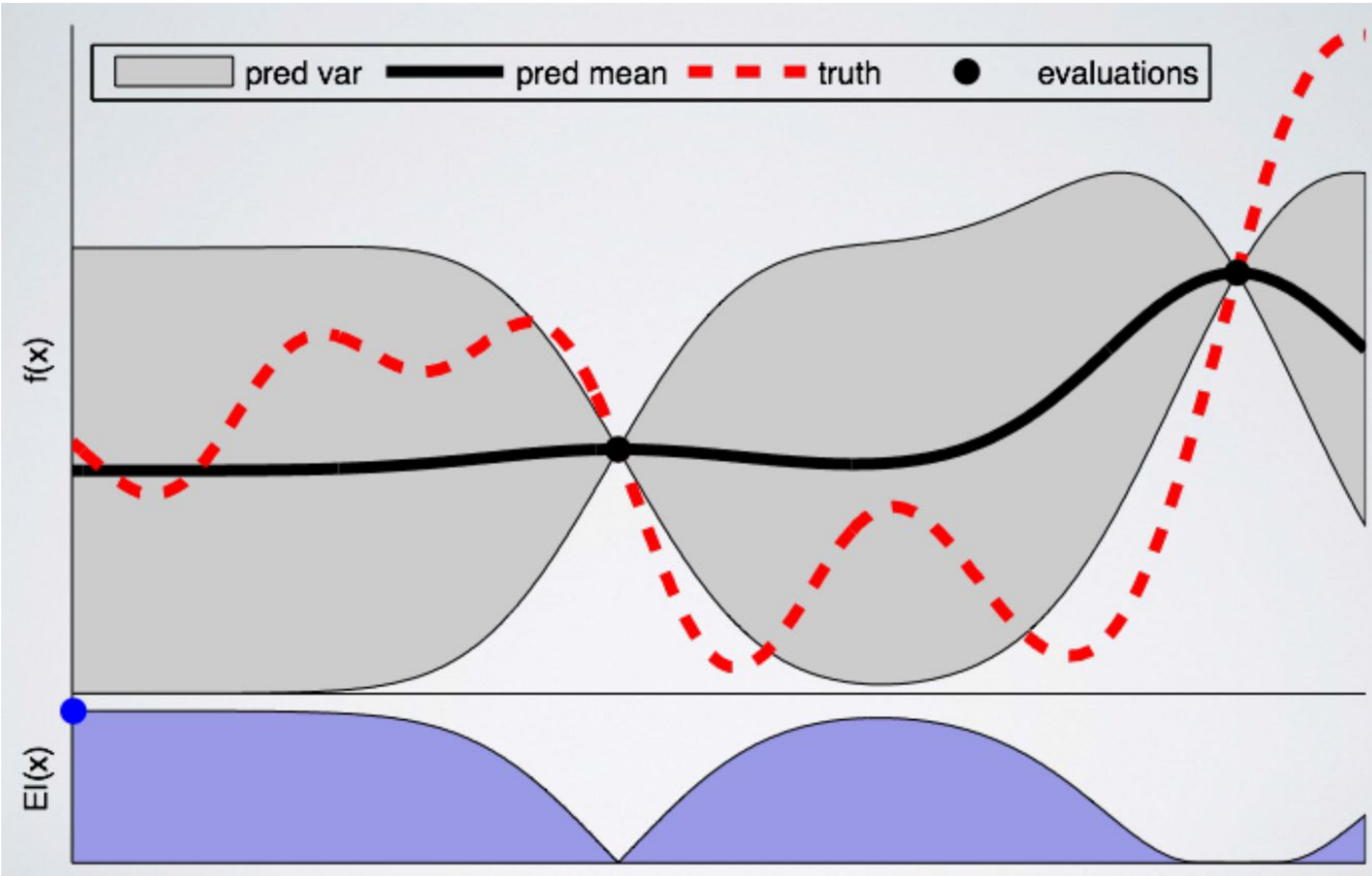
Background on Gaussian Processes and Single-Objective Bayesian Optimization

Bayesian Optimization: Key Idea

- Build a surrogate statistical model and use it to intelligently search the space
  - Replace expensive queries with cheaper queries
  - Use uncertainty of the model to select expensive queries



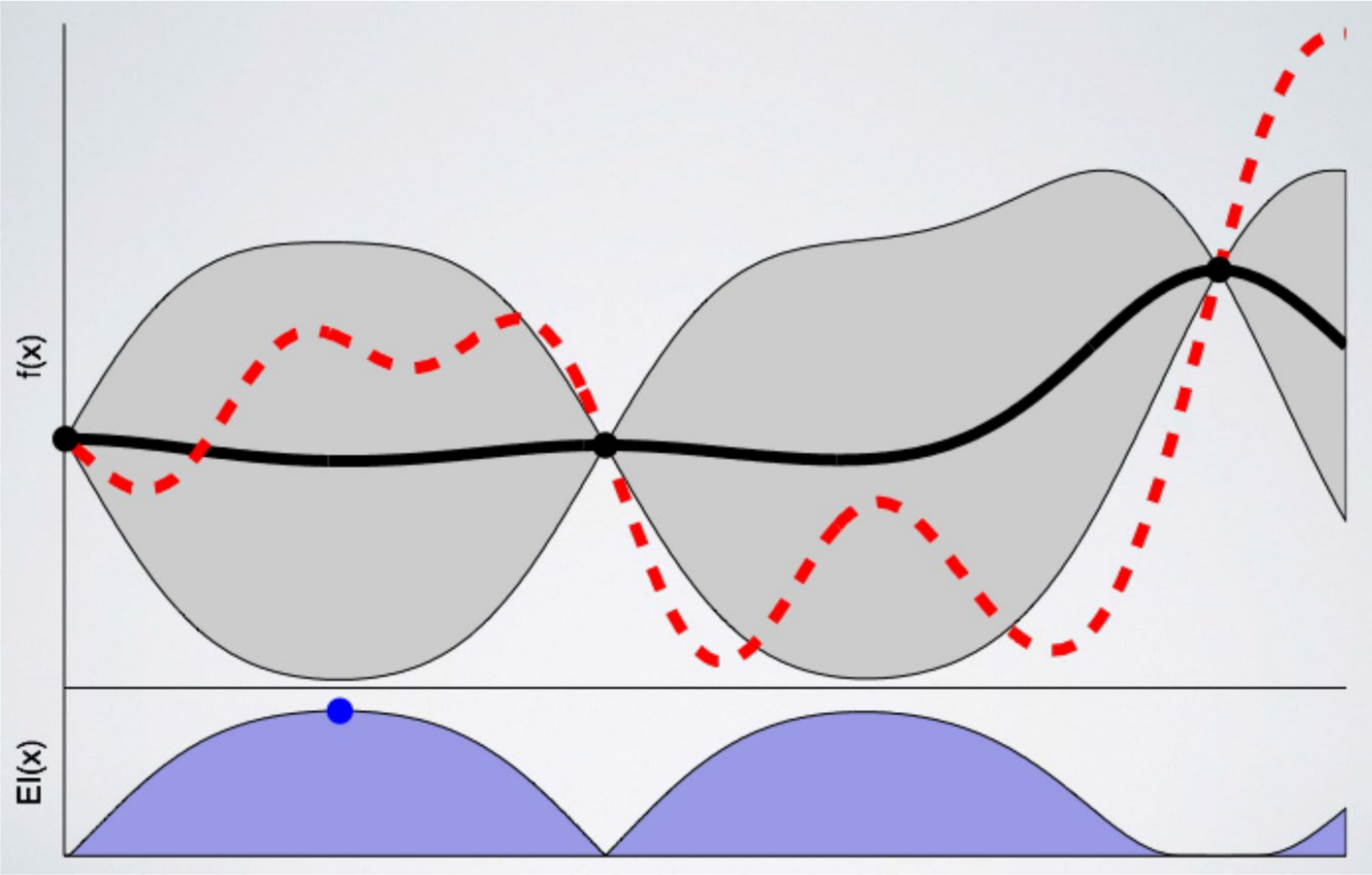
Bayesian Optimization: Illustration



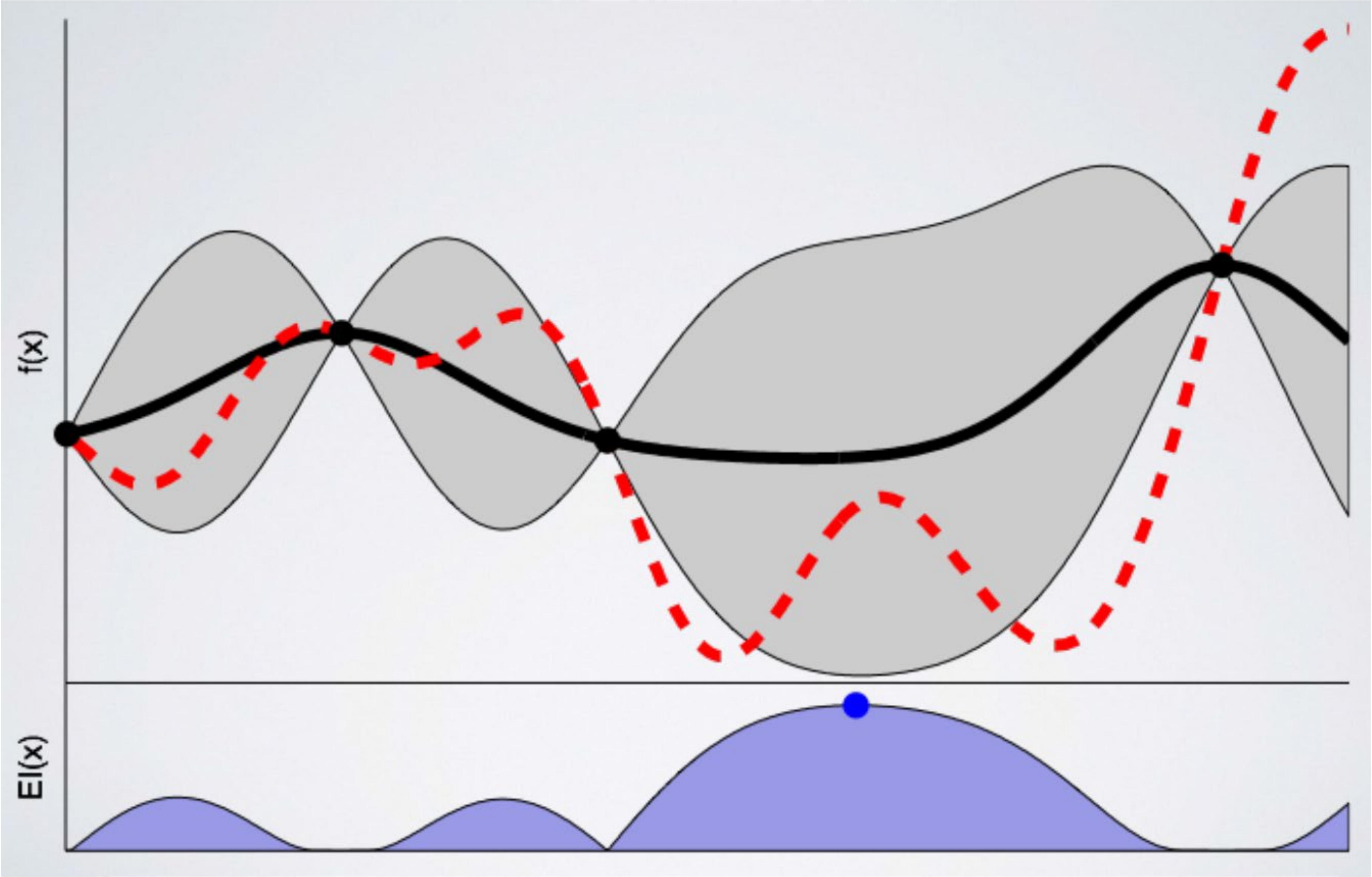
Credit: Ryan Adams

https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/tutorials/tut8_adams_slides.pdf

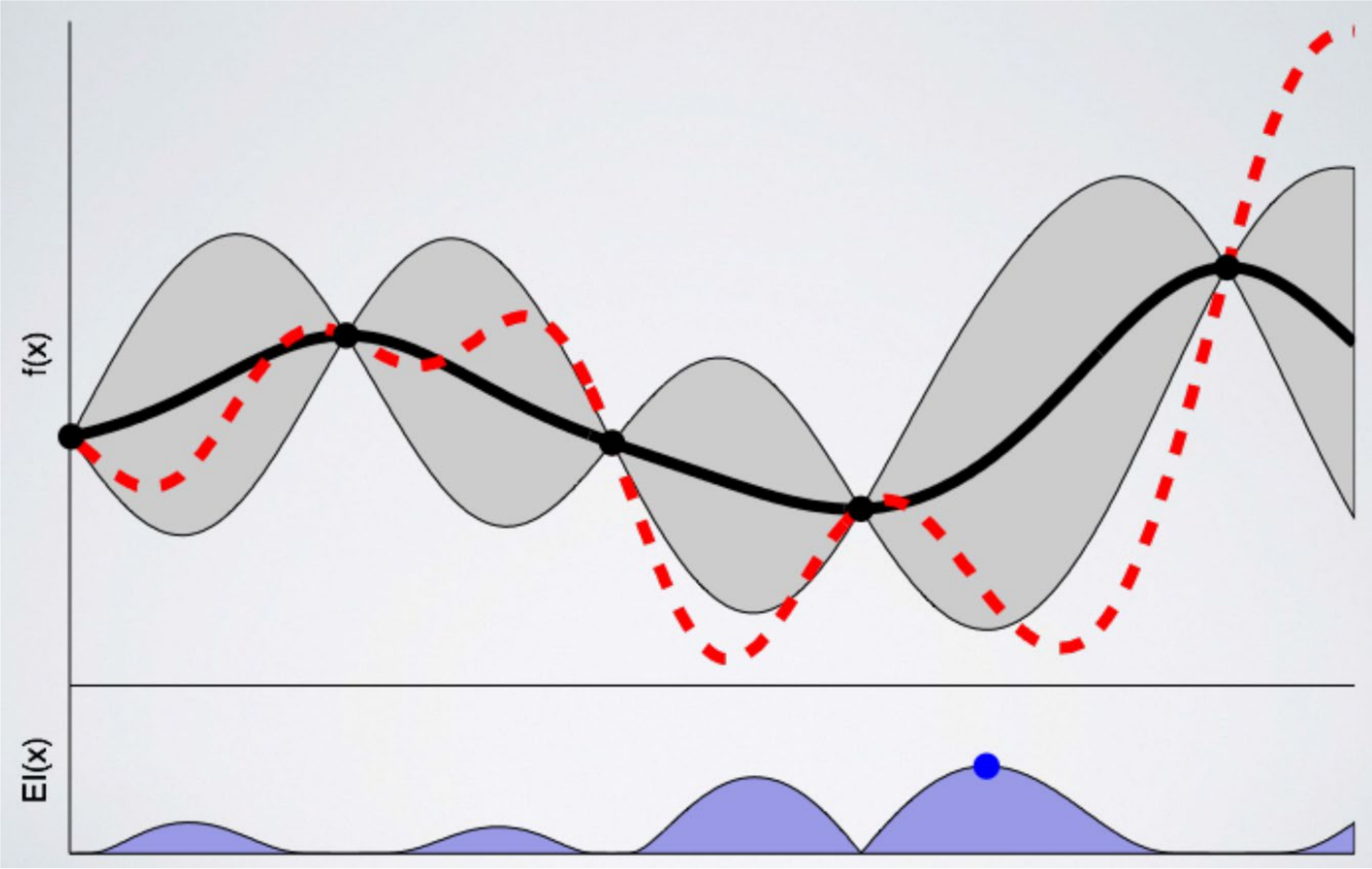
Bayesian Optimization: Illustration



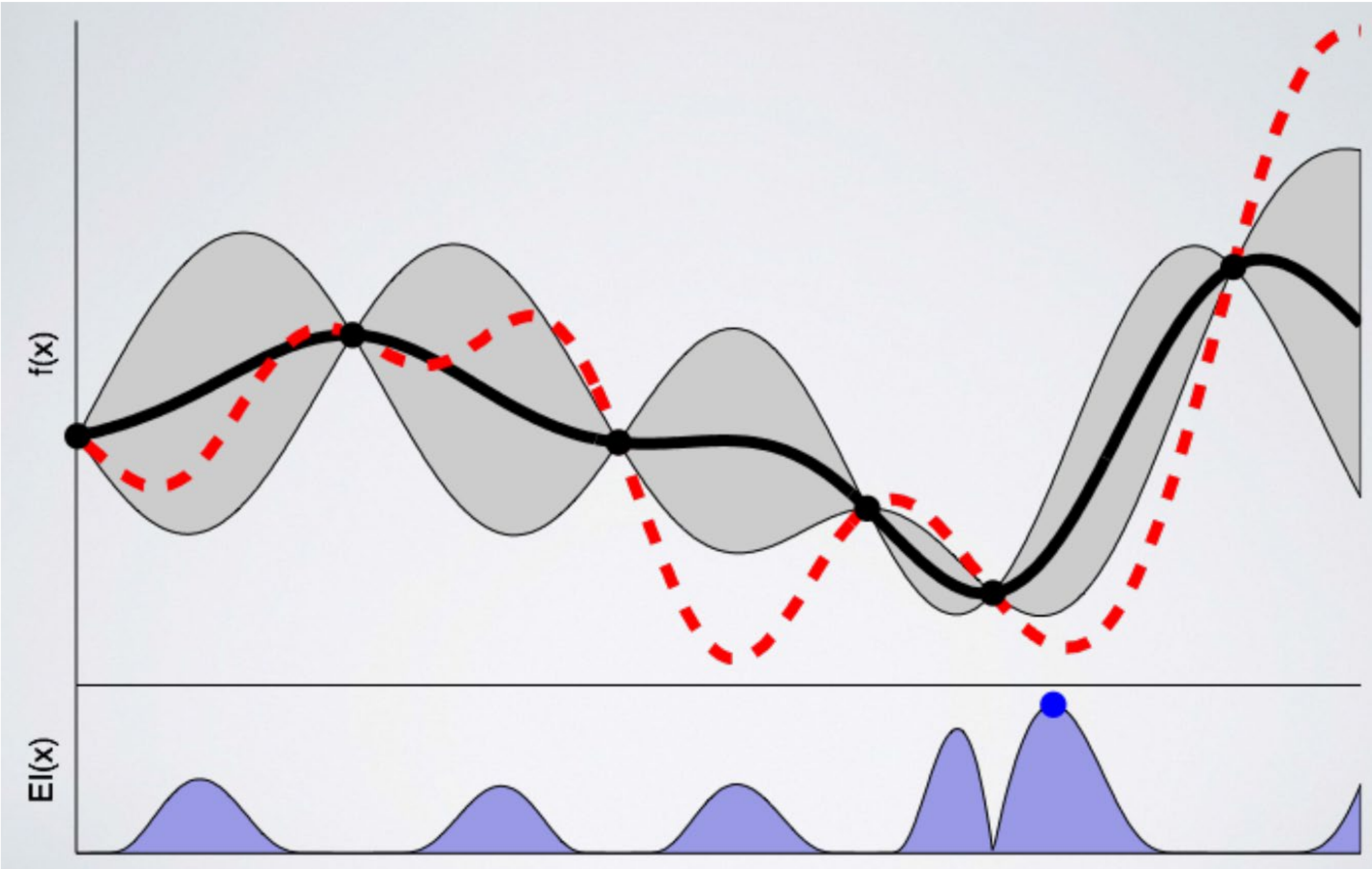
Bayesian Optimization: Illustration



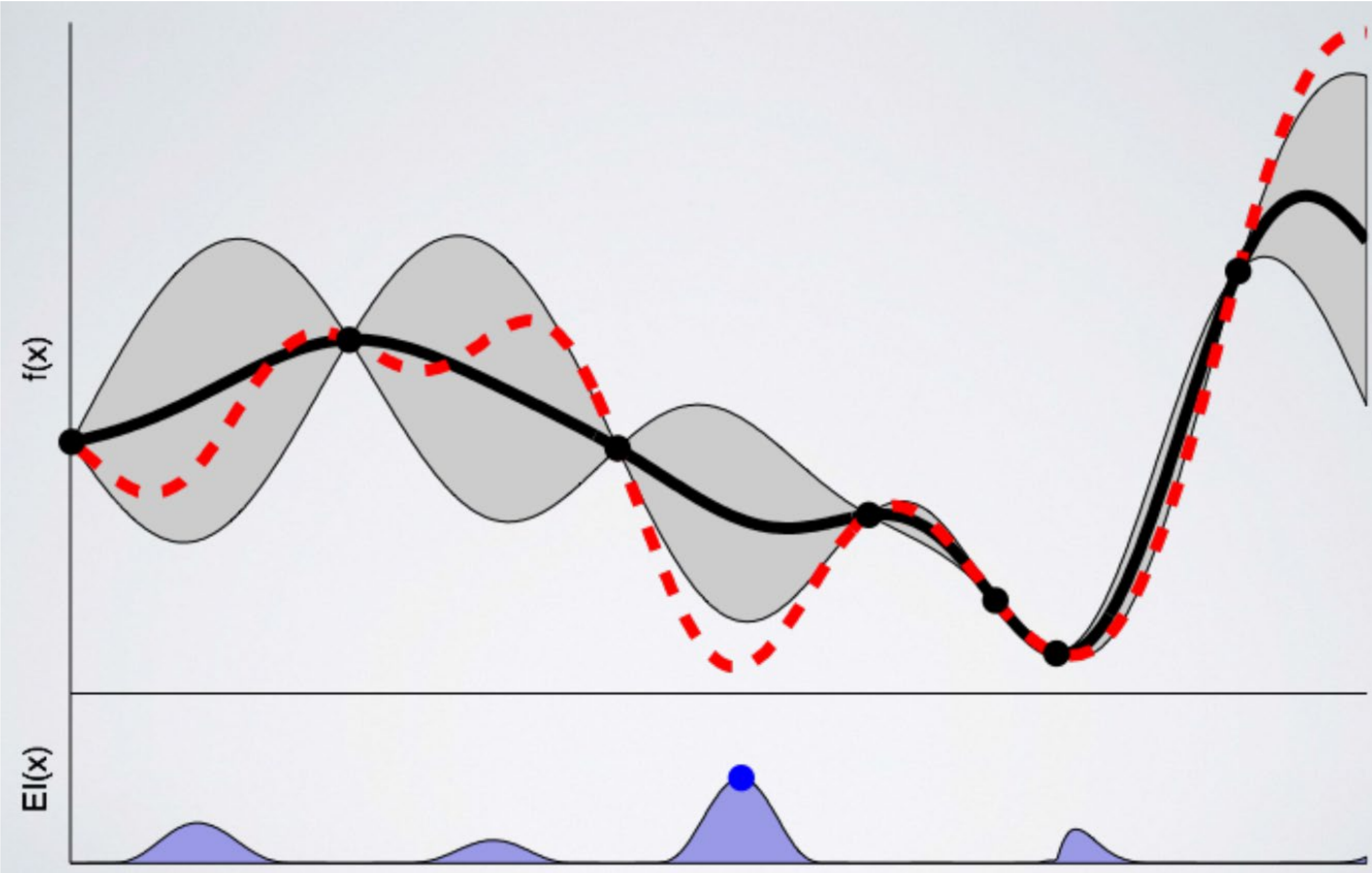
Bayesian Optimization: Illustration



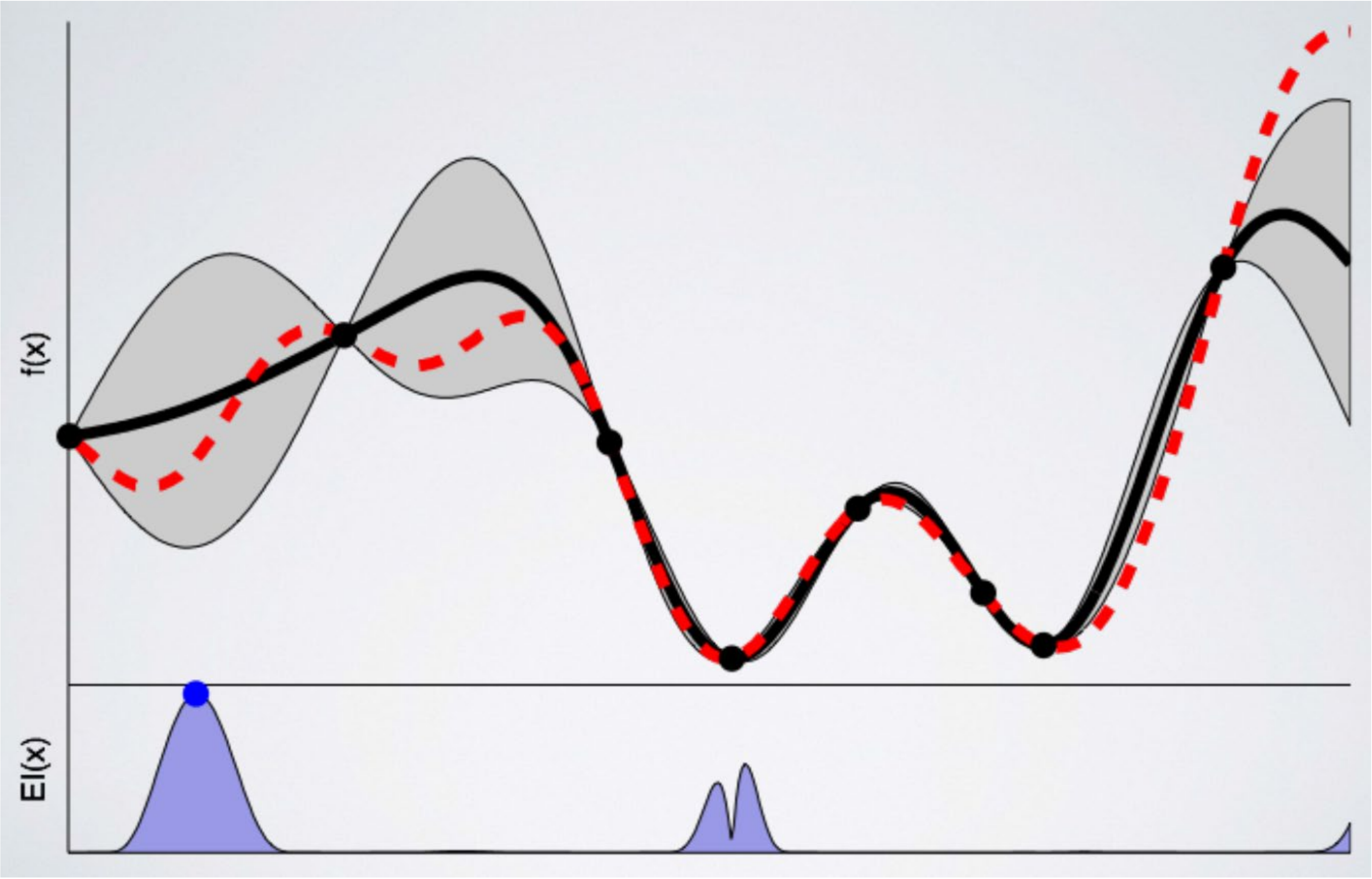
Bayesian Optimization: Illustration



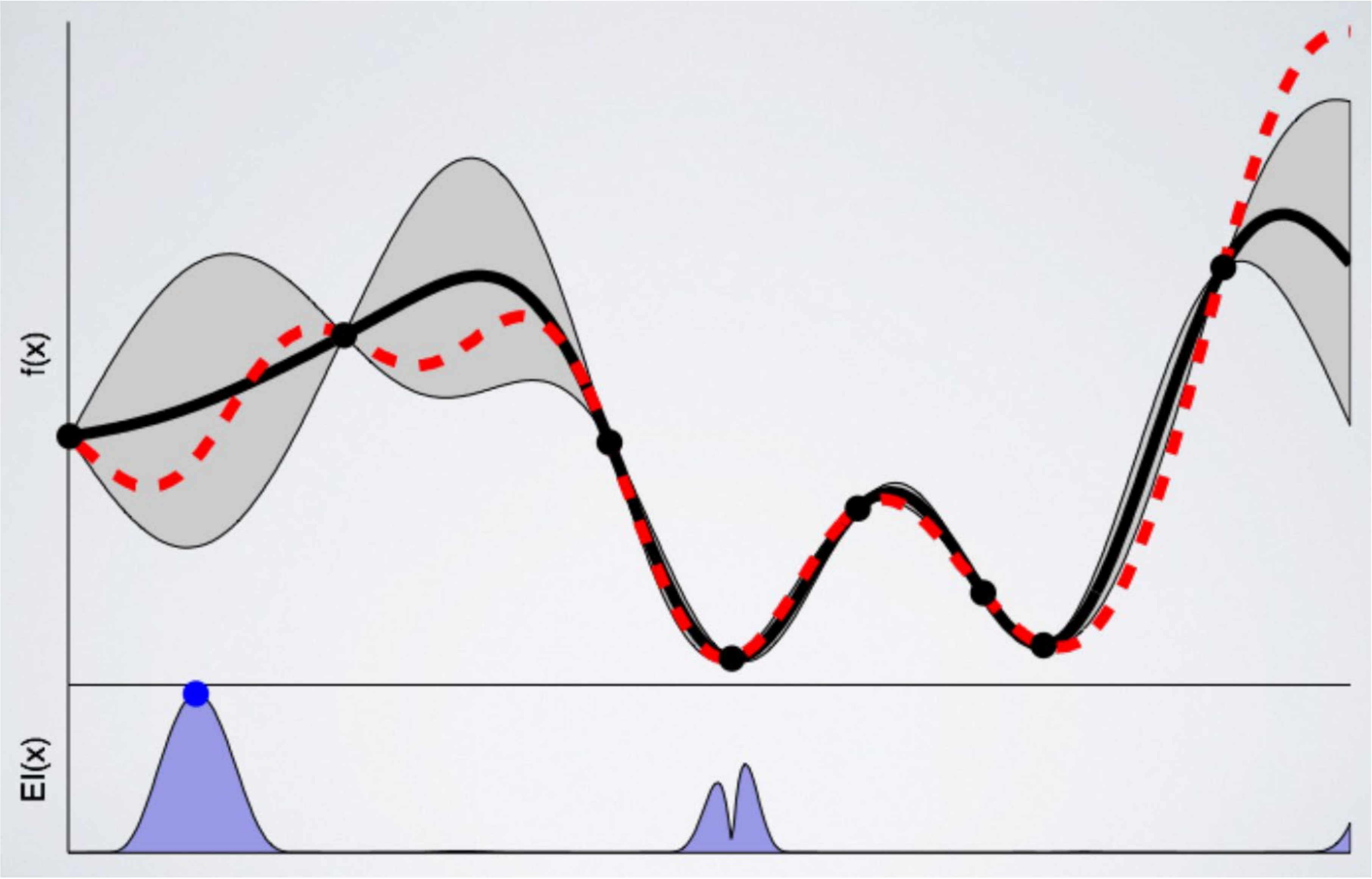
Bayesian Optimization: Illustration



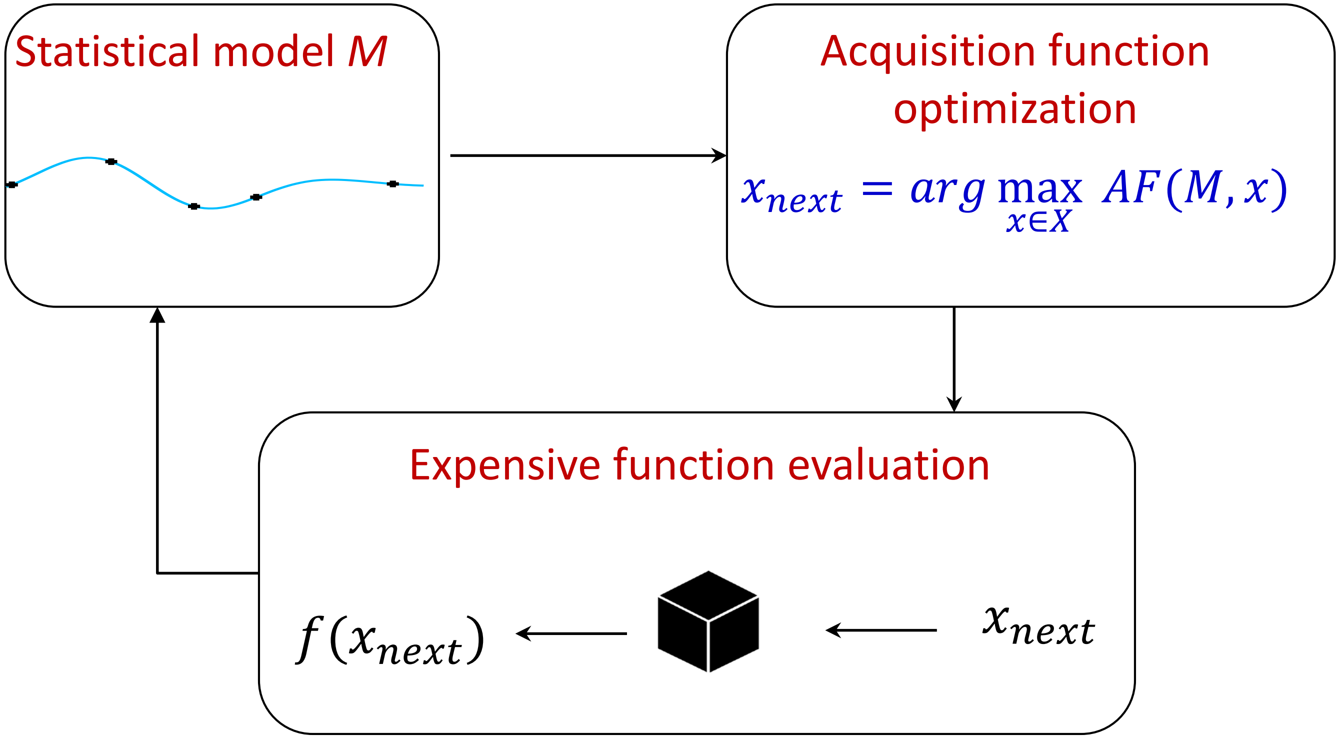
Bayesian Optimization: Illustration



Bayesian Optimization: Illustration

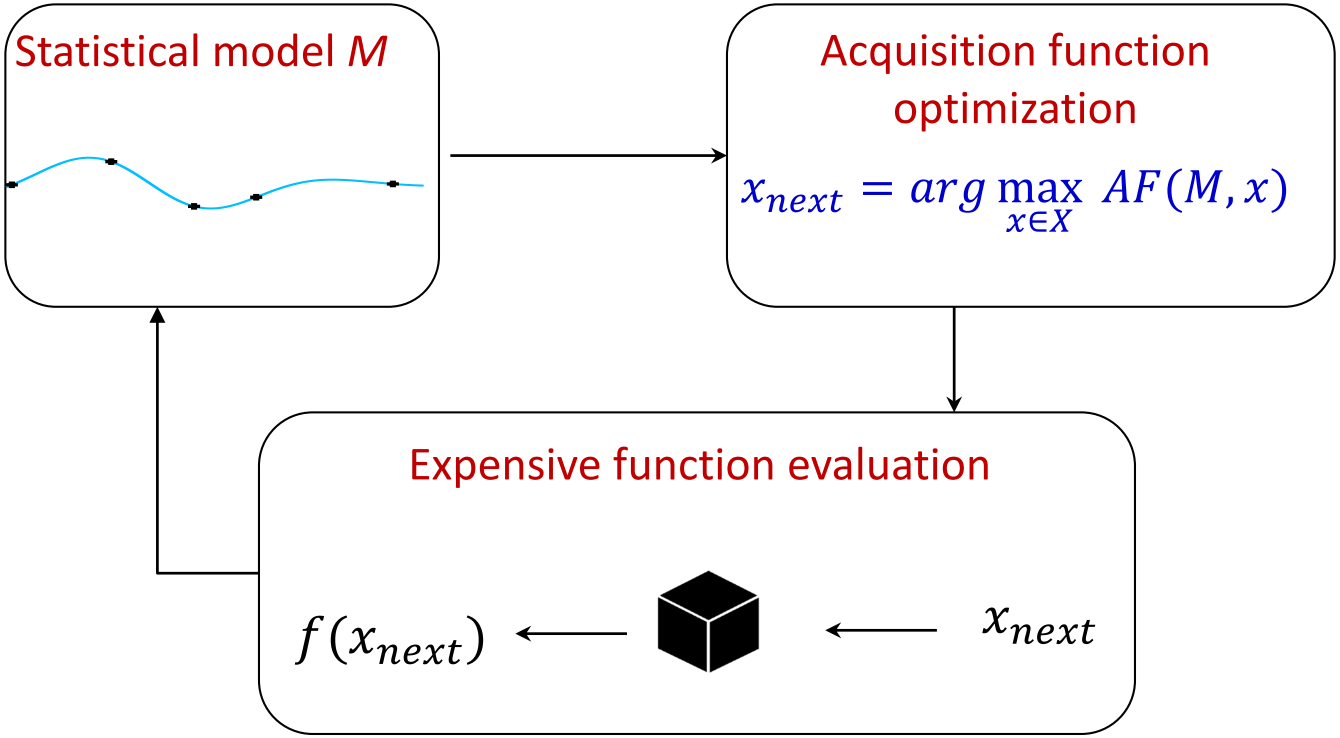


Bayesian Optimization: Three Key Elements



- Statistical model (e.g., Gaussian process)
- Acquisition function (e.g., Expected improvement)
- Acquisition function optimizer (e.g., local search)

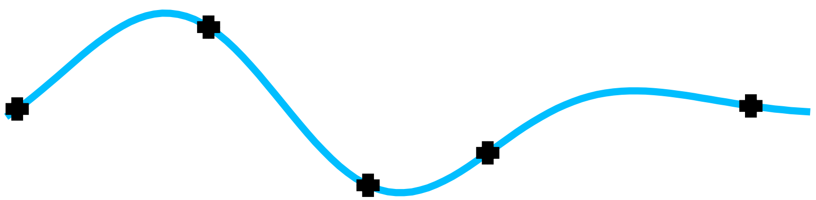
Bayesian Optimization: Three Key Elements



- Statistical model (e.g., Gaussian process)
- Acquisition function (e.g., Expected improvement)
- Acquisition function optimizer (e.g., local search)

BO needs a Probabilistic Model

- To make predictions on unknown input
- To quantify the uncertainty in predictions



- One popular class of such models are Gaussian Processes (also called GPs)

Gaussian Processes: What and Why?

Non-parametric, Bayesian and Kernel driven model

Flexibility

Principled uncertainty estimation

Specification of prior beliefs about rich function classes

Gaussian Process

- **Stochastic process definition**
  - Given any set of input points $\{x_1, x_2, \dots, x_m\}$, the output values follows a multi-variate Gaussian distribution

$$[f(x_1), f(x_2), f(x_3), \dots, f(x_m)] \sim \mathcal{N}(0, \Sigma)$$

- The covariance matrix Σ is given by a kernel function $k(x, x')$, i.e., $\Sigma_{ij} = k\left(x_i, x_j\right)$
  - Kernel captures the similarity between x and x'[1]
  - GPs are fully characterized by the kernel function[2]

Footnotes

1. For people aware of SVMs, it is the same kernel function.
2. Technically, there is also the mean function, but it is not as interesting for most applications.

Gaussian Process: Inference

- **Inference:** Given training data $\{(x_1, y_1), (x_2, y_2), \ \dots \ (x_m, y_m)\}$, the prediction for an unseen point x^*

$$\text{Prediction}(x^*) \sim \mathcal{N}(y^*, \sigma^*)$$

$$y^* = \ \boldsymbol{k}^* \, \boldsymbol{K}^{-1} \, \boldsymbol{Y}$$

$$\sigma^* = k(x^*, x^*) - \boldsymbol{k}^* \, \boldsymbol{K}^{-1} \boldsymbol{k}^*$$

$$\boldsymbol{k}^* = [k(x^*, x_1), k(x^*, x_2), \dots, k(x^*, x_m)]$$

$$\boldsymbol{K}_{\boldsymbol{ij}} = k\left(x_i, x_j\right)$$

Gaussian Process: Training

- Training procedure: searching for (kernel) hyper-parameters by optimizing the marginal log-likelihood

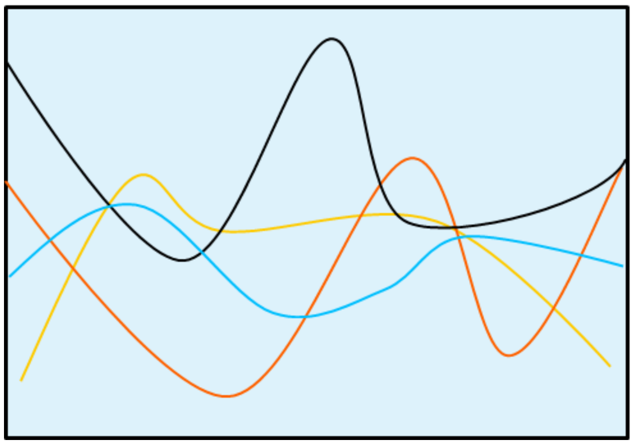
$$\log p(y) = -\frac{1}{2}\boldsymbol{Y}^T\boldsymbol{K}^{-1}\boldsymbol{Y} - \frac{1}{2}\log\det(\boldsymbol{K}) - \frac{n}{2}\log 2\pi$$

- Choice of kernel $k(x, x')$ is critical for good performance
  - Allows to incorporate domain knowledge (e.g., Morgan fingerprints in chemistry)
  - Matern kernel is a popular choice for continuous spaces

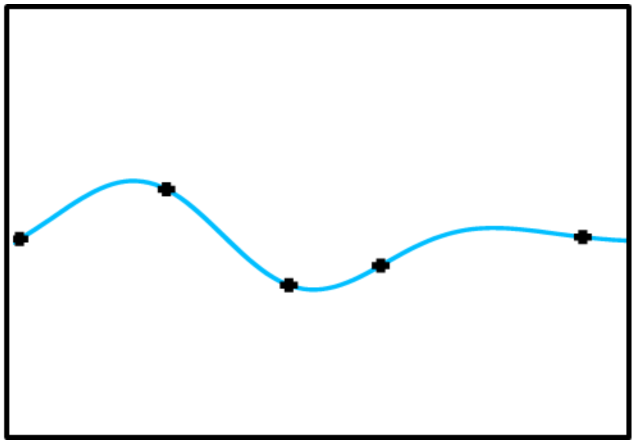
Gaussian Process: Two Views

- Function space view: distribution over functions
  - Function class is characterized by kernel

Prior



Posterior



- Weight space view: Bayesian linear regression in kernel's feature space

$$f(x) = w^T \tau(x)$$

$$k(x, x') = < \tau(x), \tau(x') >$$

Gaussian Processes: Challenges and Solutions

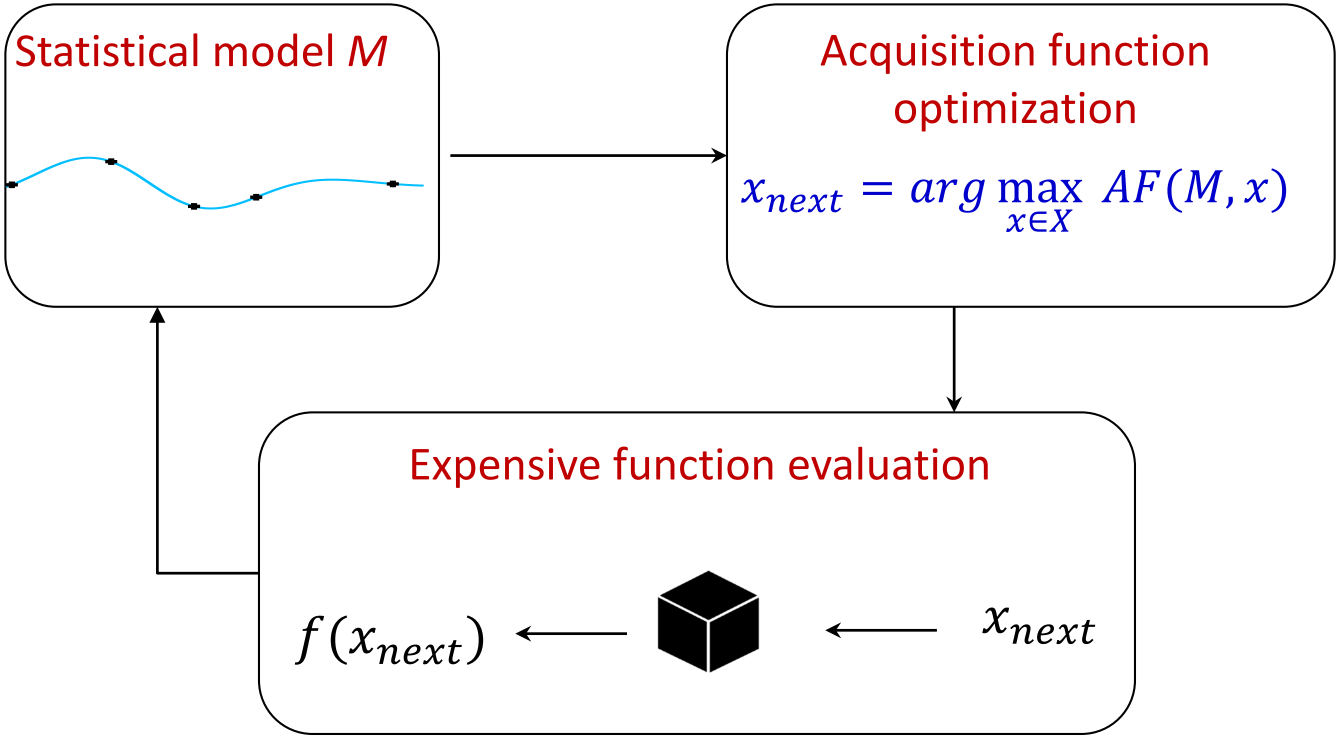
- Scalability: naive time complexity $O(n^3)$

$$\log p(y) = -\frac{1}{2}\boldsymbol{Y}^{\boldsymbol{T}}\boldsymbol{K}^{-\boldsymbol{1}}\boldsymbol{Y} - \frac{1}{2}\log\det(\boldsymbol{K}) - \frac{n}{2}\log 2\pi$$

  - Solution: Sparse Gaussian processes

- Non-Gaussian likelihoods
  - No closed form expression, e.g., classification setting
  - Solution: Approximate inference

Bayesian Optimization: Three Key Elements

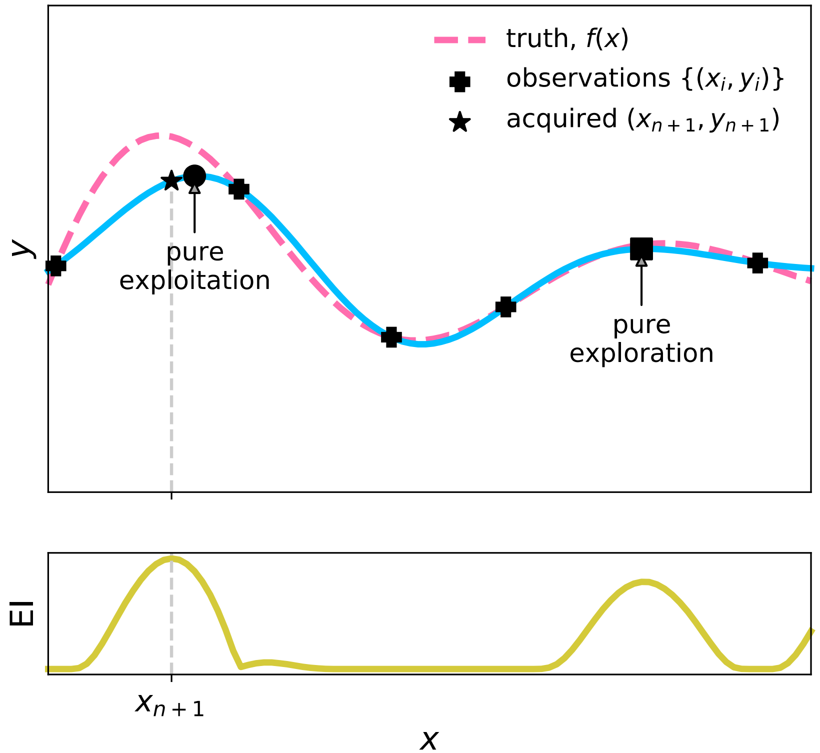


- Statistical model (e.g., Gaussian process)
- Acquisition function (e.g., Expected improvement)
- Acquisition function optimizer (e.g., local search)

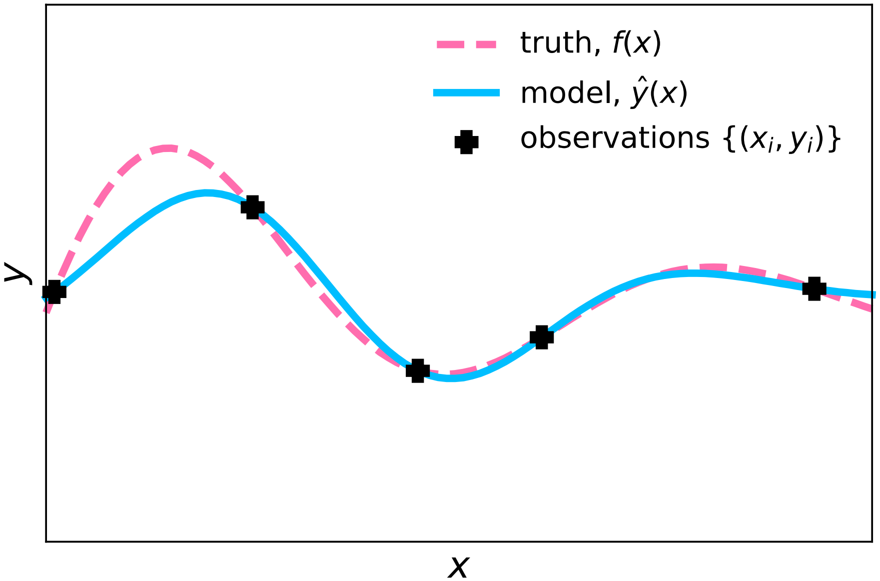
Acquisition Function

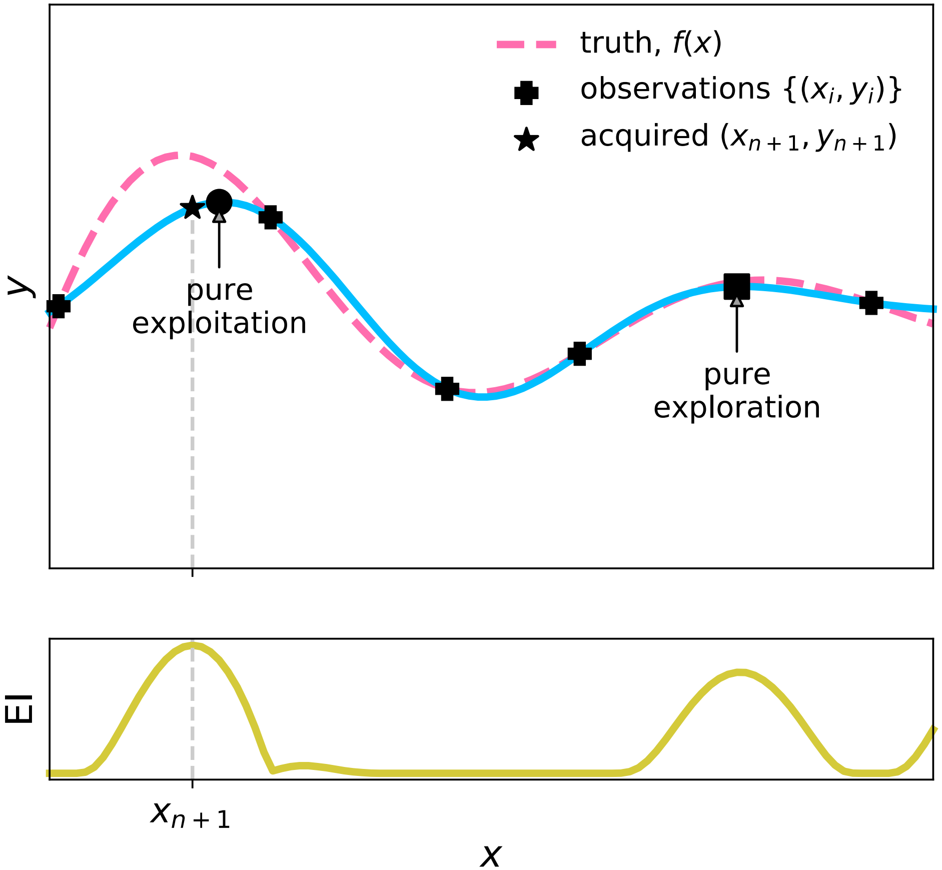
- **Intuition:** captures utility of evaluating an input

- **Challenge:** trade-off exploration and exploitation
  - Exploration: seek inputs with high variance
  - Exploitation: seek inputs with high mean



Acquisition Function: Illustration





Acquisition Function: Examples

- Upper Confidence Bound (UCB)
  - Selects input that maximizes upper confidence bound

$$\boxed{AF(x) = y^*(x) + \beta\, \sigma^*(x)}$$

- Expected Improvement (EI)
  - Selects input with highest expected improvement over the incumbent
- Thompson Sampling (TS)
  - Selects optimizer of a function sampled from the surrogate model's posterior
- Knowledge Gradient

Information-Theoretic Acquisition Functions

- Key principle: select inputs for evaluation which provide maximum information about the optimum

- Concretely, pick observations which quickly decrease the entropy of distribution over the optimum

$$AF(x) = \text{ Expected decrease in entropy}$$
$$AF(x) = H(\alpha \mid D) - E_y[H(\alpha | D \cup \{x, y\}$$
$$= \text{Information Gain}(\alpha; y)$$

- Design choices of α leads to different algorithms

Information-Theoretic Acquisition Functions

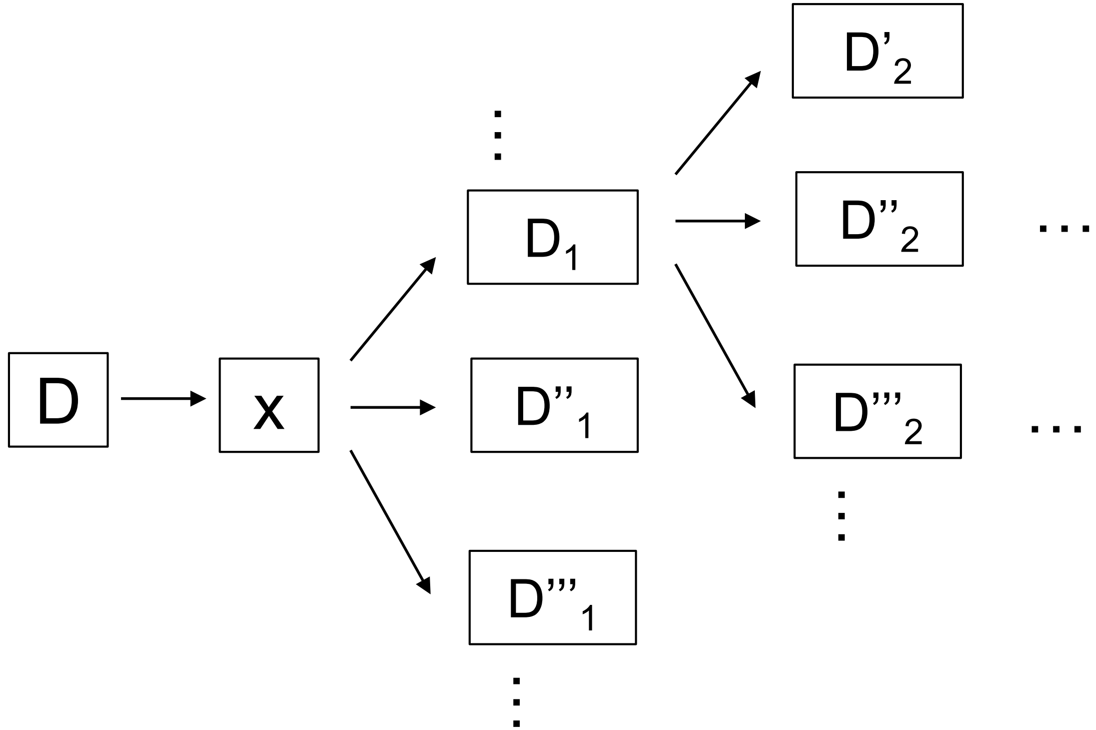
- Design choices of α leads to different algorithms

$$AF(x) = \text{Expected decrease in entropy}$$
$$AF(x) = H(\alpha \mid D) - E_y[H(\alpha | D \cup \{x, y\}]$$
$$= \text{Information Gain}(\alpha; y)$$

- α as input location of optima x^*
  - Entropy Search (ES) / Predictive Entropy Search (PES)
  - Intuitive but requires expensive approximations
- α as output value of optima y^*
  - Max-value Entropy Search (MES) and it's variants
  - Computationally cheaper and more robust

Non-Myopic / Lookahead Acquisition Functions

- Myopic acquisition functions (e.g., EI) reason about immediate utility
- Non-myopic variants consider BO as a MDP and reason about longer decision horizons



Non-Myopic / Lookahead Acquisition Functions

- Non-myopic variants consider BO as MDP and reason about longer decision horizons

$$u_k(x|D) = u_1(x|D) + E_y\left[\max_{x'} u_{t-1}(x'|D \cup \{x, y\})\right]$$

Bellman Recursion

Non-Myopic / Lookahead Acquisition Functions

- Non-myopic variants consider BO as MDP and reason about longer decision horizons

$$u_k(x|D) = u_1(x|D) + E_y\left[\max_{x'} u_{t-1}(x'|D \cup \{x, y\})\right]$$

- Challenge: curse of dimensionality

$$u_k(x|D) = u_1(x|D) + E_y\left[\max_{x1}\{u(x_1|D_1) + E_{y1}[\max_{x2}\{u(x_2|D_2)\ \ldots.\}]\}\right]$$

Non-Myopic / Lookahead Acquisition Functions

- Non-myopic variants consider BO as MDP and reason about longer decision horizons

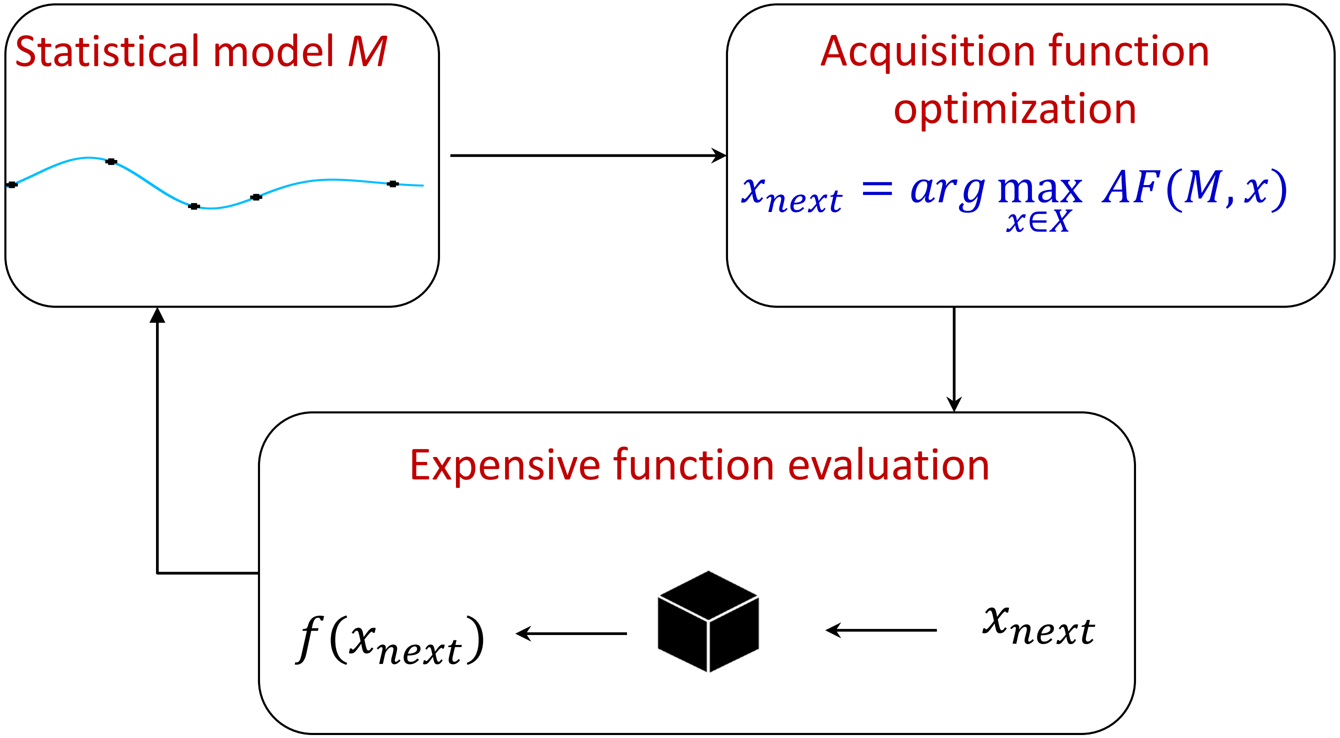
$$u_k(x|D) = u_1(x|D) + E_y\left[\max_{x'} u_{t-1}(x'|D \cup \{x, y\})\right]$$

- Challenge: curse of dimensionality

$$u_k(x|D) = u_1(x|D) + E_y\left[\max_{x1}\{u(x_1|D_1) + E_{y1}[\max_{x2}\{u(x_2|D_2) \ldots.\}]\}\right]$$

- Some solutions
  - Multi-step lookahead policies with approximations
  - Rollout based approximate dynamic programming

Bayesian Optimization: Three Key Elements



- Statistical model (e.g., Gaussian process)
- Acquisition function (e.g., Expected improvement)
- Acquisition function optimizer (e.g., local search)

Acquisition Function Optimizer

- Challenge: non-convex/multi-modal optimization problem

- Commonly used approaches
  - Space partitioning methods (e.g., DIRECT, LOGO)
  - Gradient based methods (e.g., Gradient descent)
  - Evolutionary search (e.g., CMA-ES)

BO Software: BoTorch

- Scalability via automatic differentiation
  - PyTorch/GpyTorch
- Monte-Carlo acquisition functions
  - Express acquisition functions as expectations of utility functions
  - Compute expectations via Monte-Carlo sampling
  - Use the reparameterization trick to make acquisition functions differentiable
- Other software: Trieste (based on TensorFlow)
- Not actively maintained: GPyOpt, Spearmint

Questions ?