OPTIMIZATION TECHNIQUES

Bayesian optimization

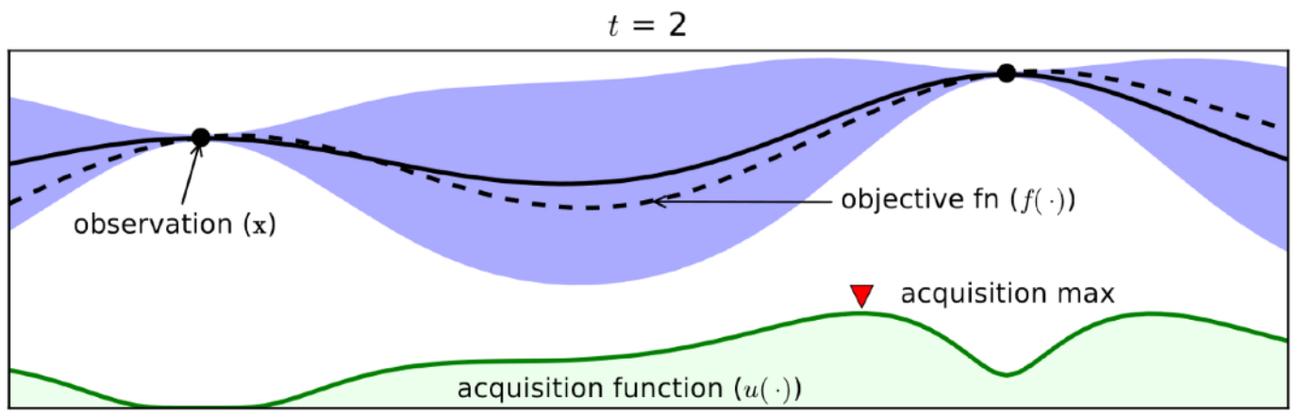
Prof. Giovanni Iacca

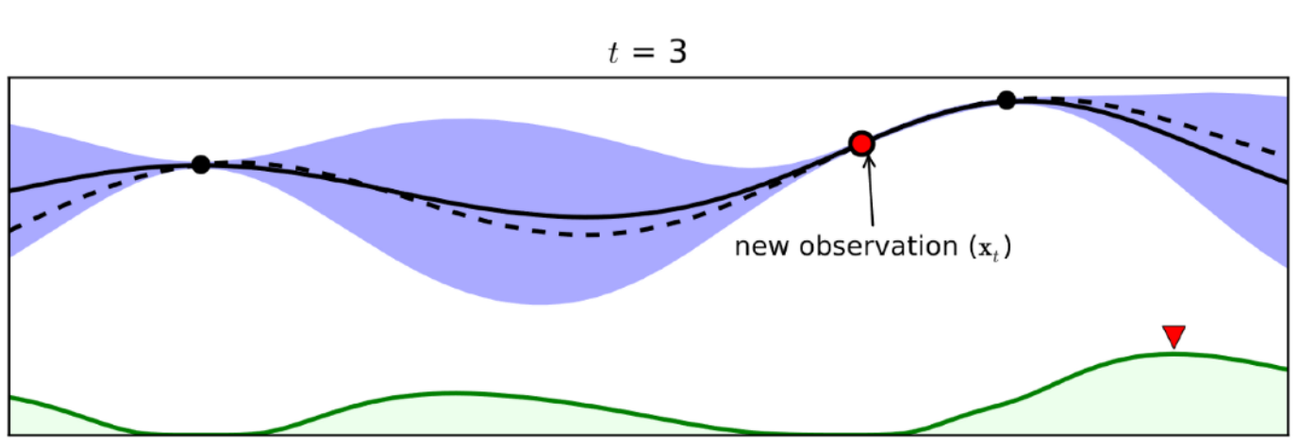
giovanni.iacca@unitn.it

UNIVERSITY OF TRENTO - Italy

Information Engineering
and Computer Science Department

Bayesian optimization





BAYESIAN OPTIMIZATION

PROBLEM STATEMENT (NOTE: WE ASSUME MAXIMIZATION)

Given the following optimization problem:

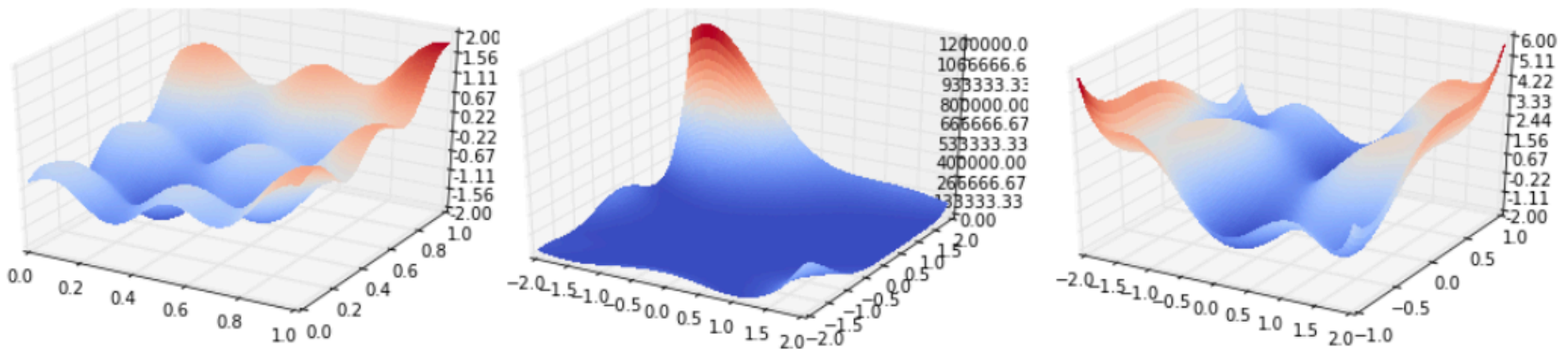
$$x^* = \text{argmax}\, f(x)$$

with the following constraints:

- $f(x)$ is a black-box for which no closed form is known (but, it should be smooth & continuous);
- gradient df/dx is not available;
- $f(x)$ **is expensive to evaluate**;
- (optional) uncertainty on observations y_i of $f(x)$, e.g., $y_i = f(x_i) + \varepsilon_i$ because of Poisson fluctuations;

The goal of BO is to find x^*, while **minimizing the number of evaluations** $f(x)$.

NOTE: if the constraints above do not hold, there is certainly a better optimization algorithm than Bayesian optimization (e.g., L-BFGS-B, Powell's method, etc.).



BAYESIAN OPTIMIZATION

THE ALGORITHM IN A NUTSHELL (MOCKUS, 1978)

1. Choose some *prior* over the space of possible objectives functions $f(x)$. Essentially, a **probabilistic model** for the objective $f(x)$ that will be used as *surrogate* for the optimization process.

For t from 1 to T (T being the max no. of iterations):

2. Given a number of observations $(x_i, y_i=f(x_i))$ for $i=1:t$, integrate out all possible true function evaluations to update the prior to form the *posterior* distribution over the objective function.
3. Optimize a *cheap* **acquisition function** (aka utility function or infill sampling criteria) $u(x)$ based on the *posterior* distribution for finding the next point x_{t+1} (and, in doing so, *exploit uncertainty to balance exploration against exploitation*):

$$x_{t+1}=\text{argmax } u(x)$$

4. Sample the next observation (i.e., perform function evaluation): $y_{t+1}=f(x_{t+1})$.

Essentially, BO transforms the problem of finding the optimum of an **expensive** function $f(x)$ into a series of sampling problems, where for each problem we want to find the sample that optimizes a given **cheap** acquisition function $u(x)$, where x is expected to optimize also $f(x)$.

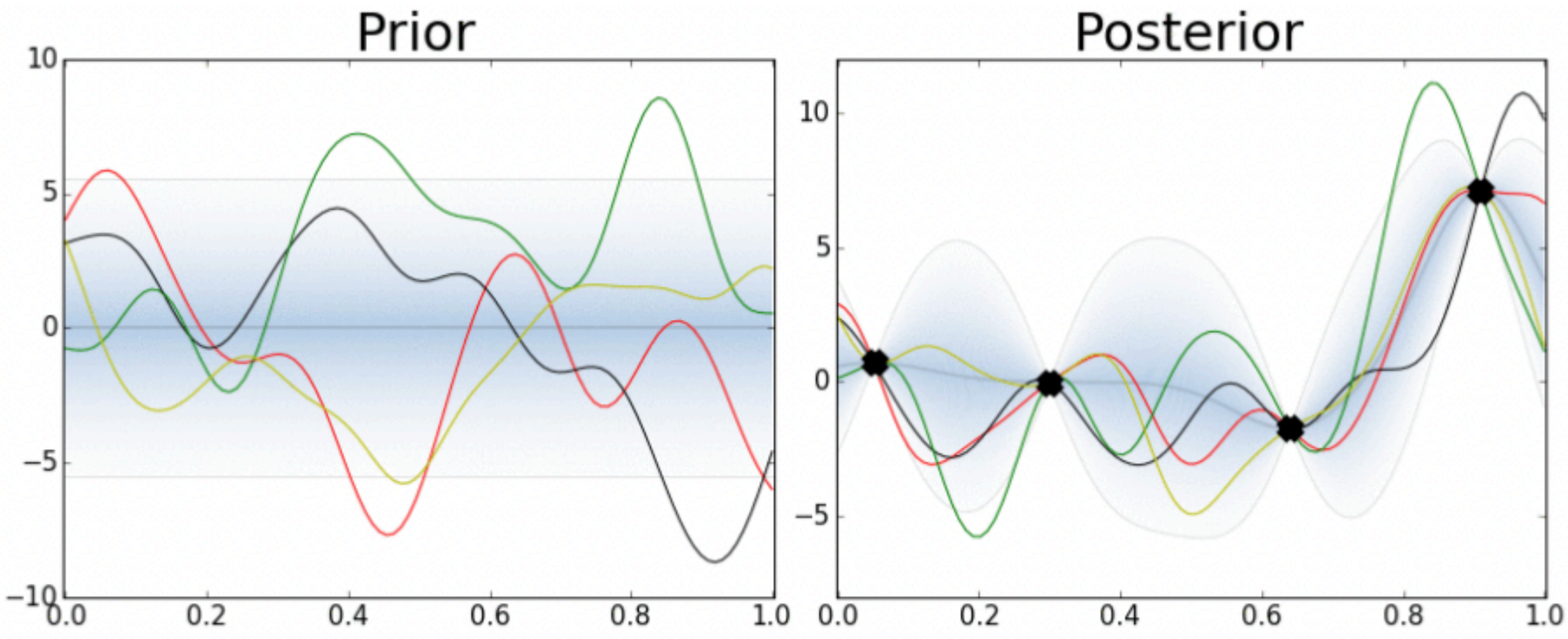
BAYESIAN OPTIMIZATION

DEFAULT SURROGATE CHOICE: GAUSSIAN PROCESSES (GP) REGRESSION

- Gaussian Processes (GP) are collections of random variables (of potentially infinite size), any finite number of which have (consistent) joint Gaussian distributions.

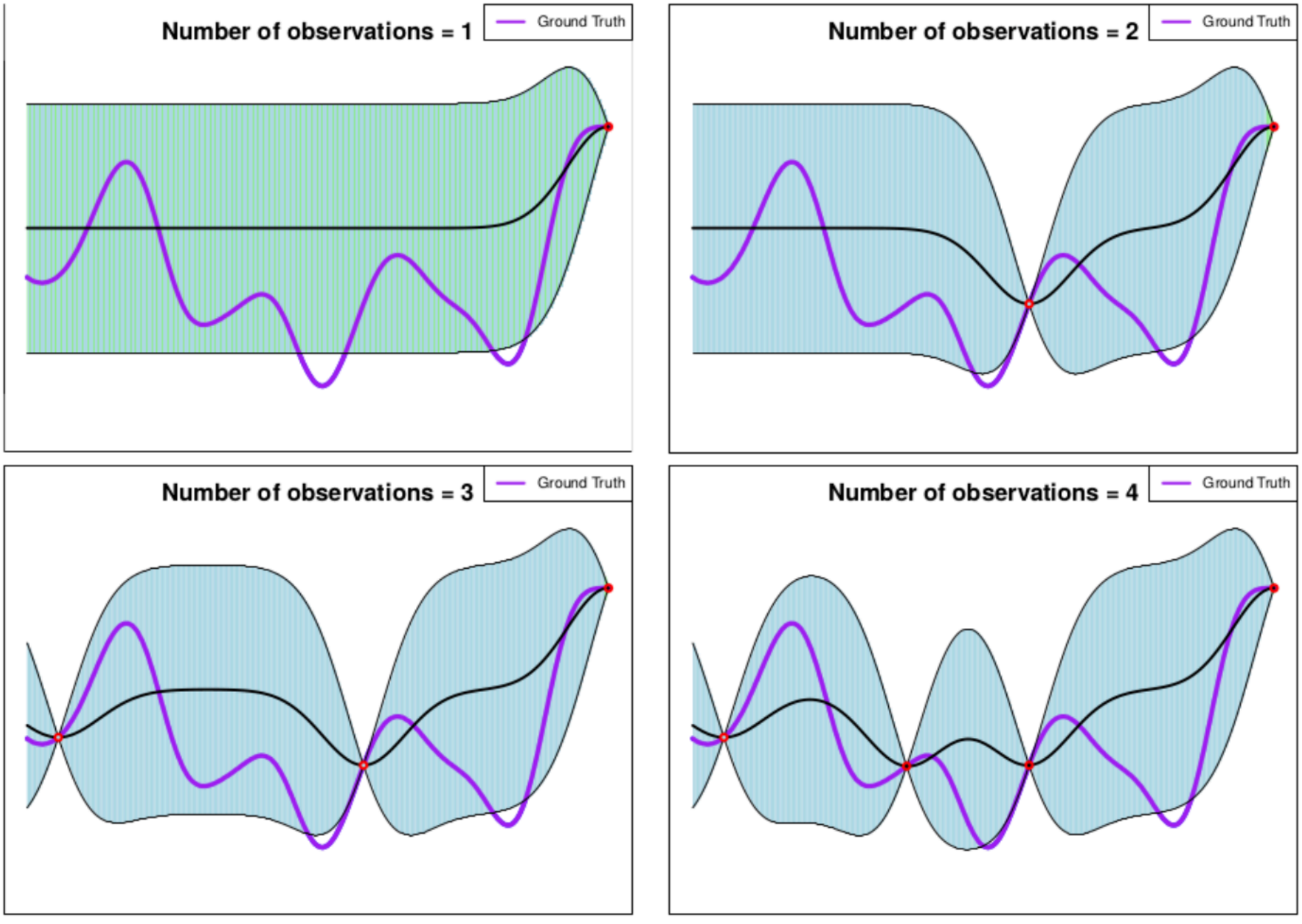
- Model: $f(x) \sim \mathrm{GP}(\mu(x), k(x, x'))$, fully determined by the mean function $m(x)$ and the covariance function $k(x, x'; \theta)$. The latter is also called kernel and often includes hyper-parameters to tune: it describes the correlation between any two points, i.e., the relationship between them, and is thus an implicit constraint on the way the resultant fit would look like.

- Posterior mean $\mu(x; \theta, D)$ and variance $\sigma(x; \theta, D)$ can be computed explicitly given a dataset D.



BAYESIAN OPTIMIZATION

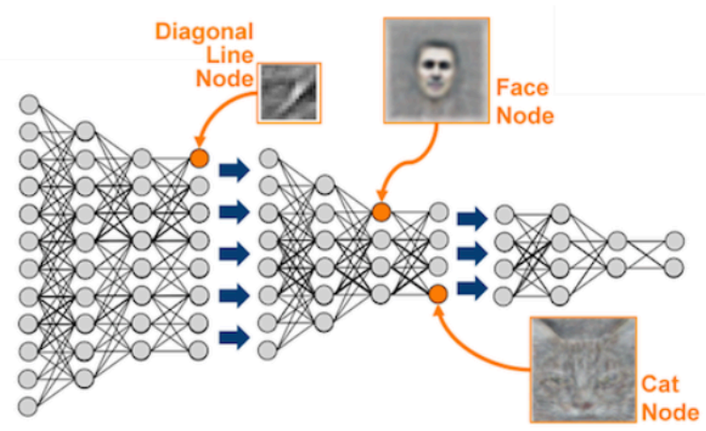
DEFAULT SURROGATE CHOICE: GAUSSIAN PROCESSES (GP) REGRESSION



BAYESIAN OPTIMIZATION

APPLICATIONS

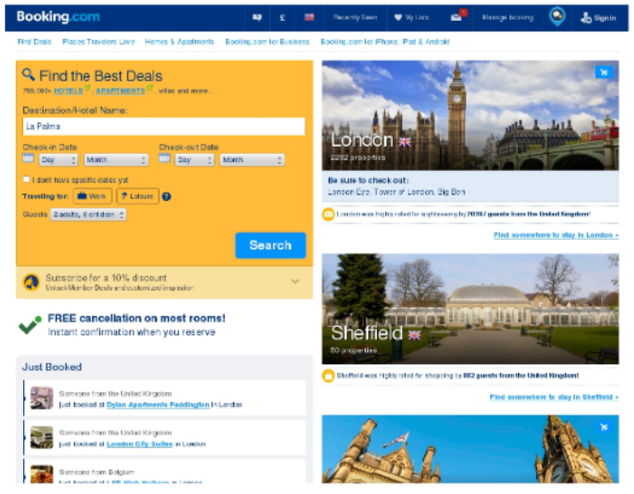
- Robotics, control systems, machine learning, life sciences (e.g., gene/protein optimization)
- Hyper-parameter tuning (tuning of optimization/ML model parameters)
- Model tuning, e.g. for high-energy physics simulations
- Optimization of compiler flags



Parameter tuning in ML algorithms

- Number of layers
- Number/kinds of units per layer
- Learning rates, etc.

Active Path Finding in Middle Level
Optimize the location of a sequence of waypoints in a map to navigate from a location to a destination.



Tuning websites with A/B testing
Optimize the web design to maximize sign-ups, downloads, purchases, etc.

BAYESIAN OPTIMIZATION

HYPER-PARAMETER TUNING THROUGH BO: A FAMOUS EXAMPLE

Bayesian Optimization in AlphaGo

Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver & Nando de Freitas

DeepMind, London, UK
yutianc@google.com

Abstract

During the development of AlphaGo, its many hyper-parameters were tuned with Bayesian optimization multiple times. This automatic tuning process resulted in substantial improvements in playing strength. For example, prior to the match with Lee Sedol, we tuned the latest AlphaGo agent and this improved its win-rate from 50% to 66.5% in self-play games. This tuned version was deployed in the final match. Of course, since we tuned AlphaGo many times during its development cycle, the compounded contribution was even higher than this percentage. It is our hope that this brief case study will be of interest to Go fans, and also provide Bayesian optimization practitioners with some insights and inspiration.

BAYESIAN OPTIMIZATION

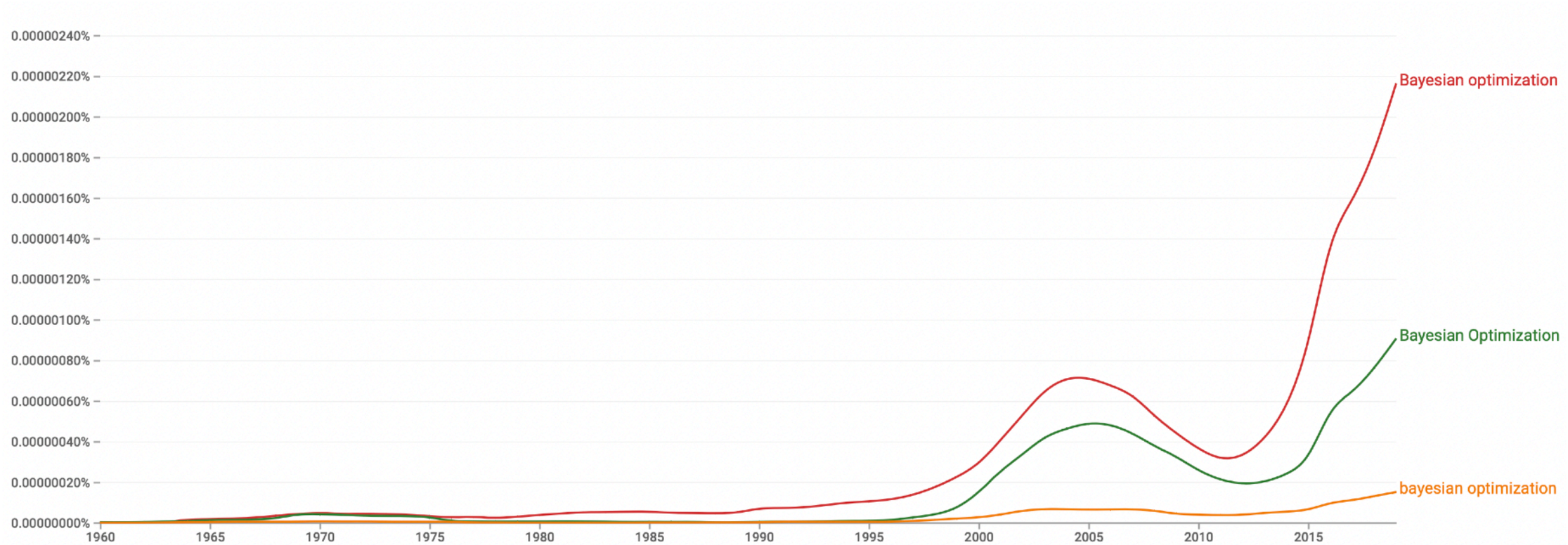
WIDELY AVAILABLE IN MANY SW PACKAGES/PROGRAMMING LANGUAGES

- Python
  - Spearmint https://github.com/JasperSnoek/spearmint
  - GPyOpt https://github.com/SheffieldML/GPyOpt
  - RoBO https://github.com/automl/RoBO
  - BOTorch https://botorch.org/
  - scikit-optimize https://github.com/MechCoder/scikit-optimize
  - Vanilla BO https://github.com/glouppe/talk-bayesian-optimisation
  - **Advanced version developed at Uber** https://github.com/uber-research/TuRBO
  - GPFlow (https://github.com/GPflow/GPflow) and GPyTorch (https://github.com/cornellius-gp/gpytorch) are GP regression libraries built on top of Tensorflow and PyTorch.
- C++
  - MOE https://github.com/yelp/MOE
- R
  - DiceOptim https://cran.r-project.org/web/packages/DiceOptim/index.html
  - laGP https://cran.r-project.org/web/packages/laGP/index.html
- Matlab
  - DACE http://www2.imm.dtu.dk/projects/dace/

BAYESIAN OPTIMIZATION

INCREASINGLY POPULAR FIELD

- Hot topic in Machine Learning.
- The BO workshop at NeurIPS is well stablished and it is a mini-conference itself.



BAYESIAN OPTIMIZATION

INCREASINGLY POPULAR FIELD

So popular that in the "Proposal for a Regulation laying down harmonized rules on artificial intelligence" from the EU Commission (https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence), a.k.a. "Artificial Intelligence Act", BO is explicitly mentioned within the definition of "AI".

Article 3

Definitions

For the purpose of this Regulation, the following definitions apply:

(1) 'artificial intelligence system' (AI system) means software that is developed with one or more of the techniques and approaches **listed in Annex I** and can, for a given set of human-defined objectives, generate outputs such as content, predictions, recommendations, or decisions influencing the environments they interact with.

ANNEX I

ARTIFICIAL INTELLIGENCE TECHNIQUES AND APPROACHES

referred to in Article 3, point 1

(a) Machine learning approaches, including supervised, unsupervised and reinforcement learning, using a wide variety of methods including deep learning;

(b) Logic- and knowledge-based approaches, including knowledge representation, inductive (logic) programming, knowledge bases, inference and deductive engines, (symbolic) reasoning and expert systems;

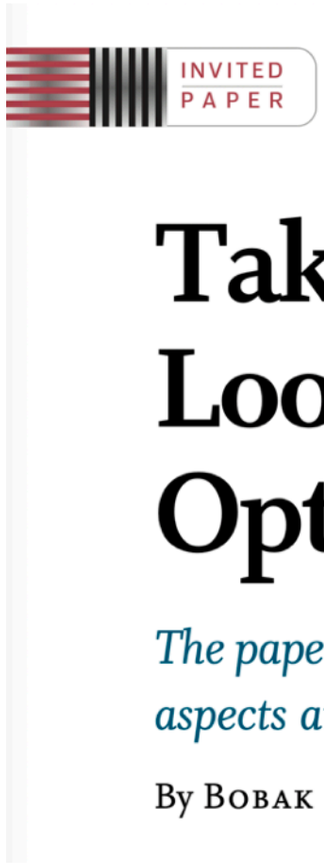
(c) Statistical approaches, **Bayesian estimation, search and optimization methods.**

BAYESIAN OPTIMIZATION

INCREASINGLY POPULAR FIELD

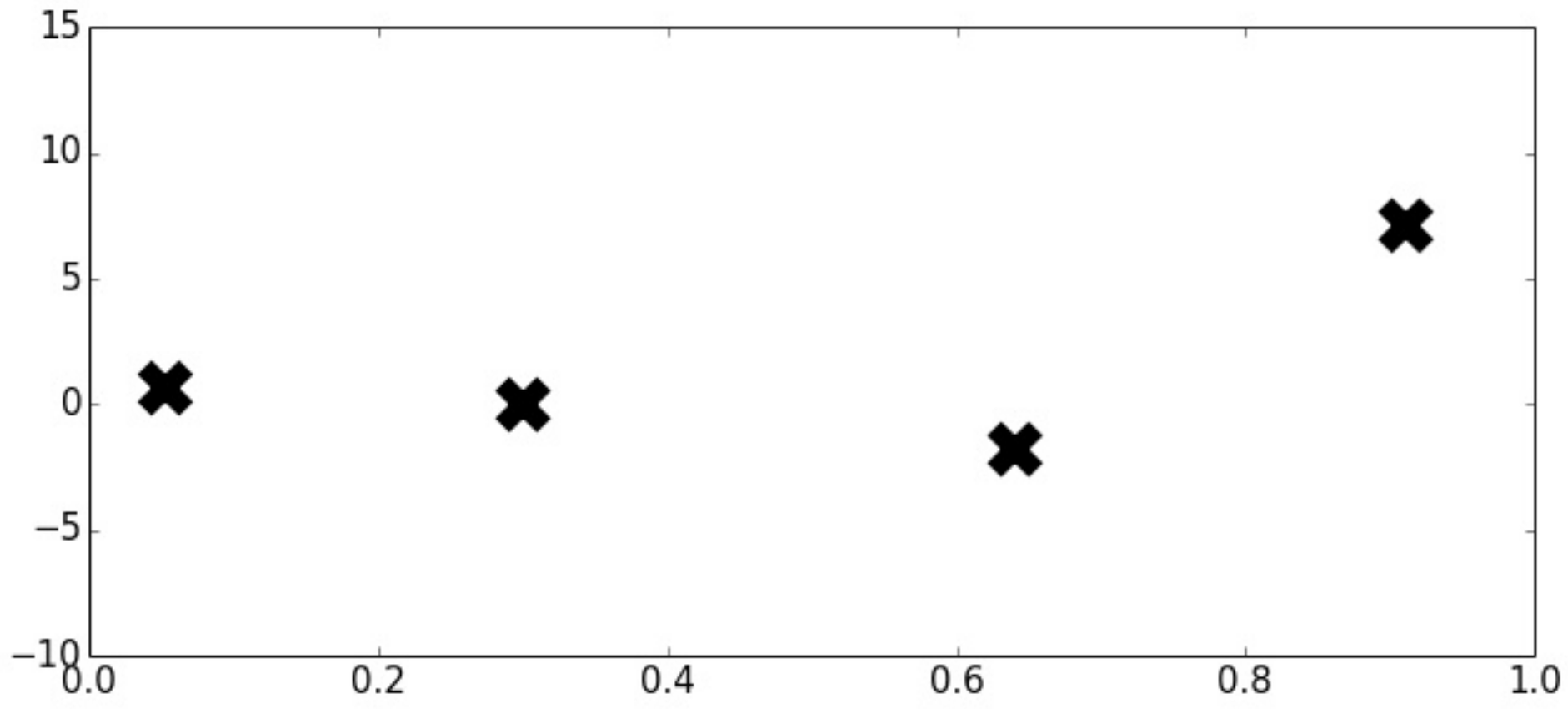
- Why this popularity?
  … Since allows to configure algorithms *without human intervention*.

- BO "takes the human out of the loop"!



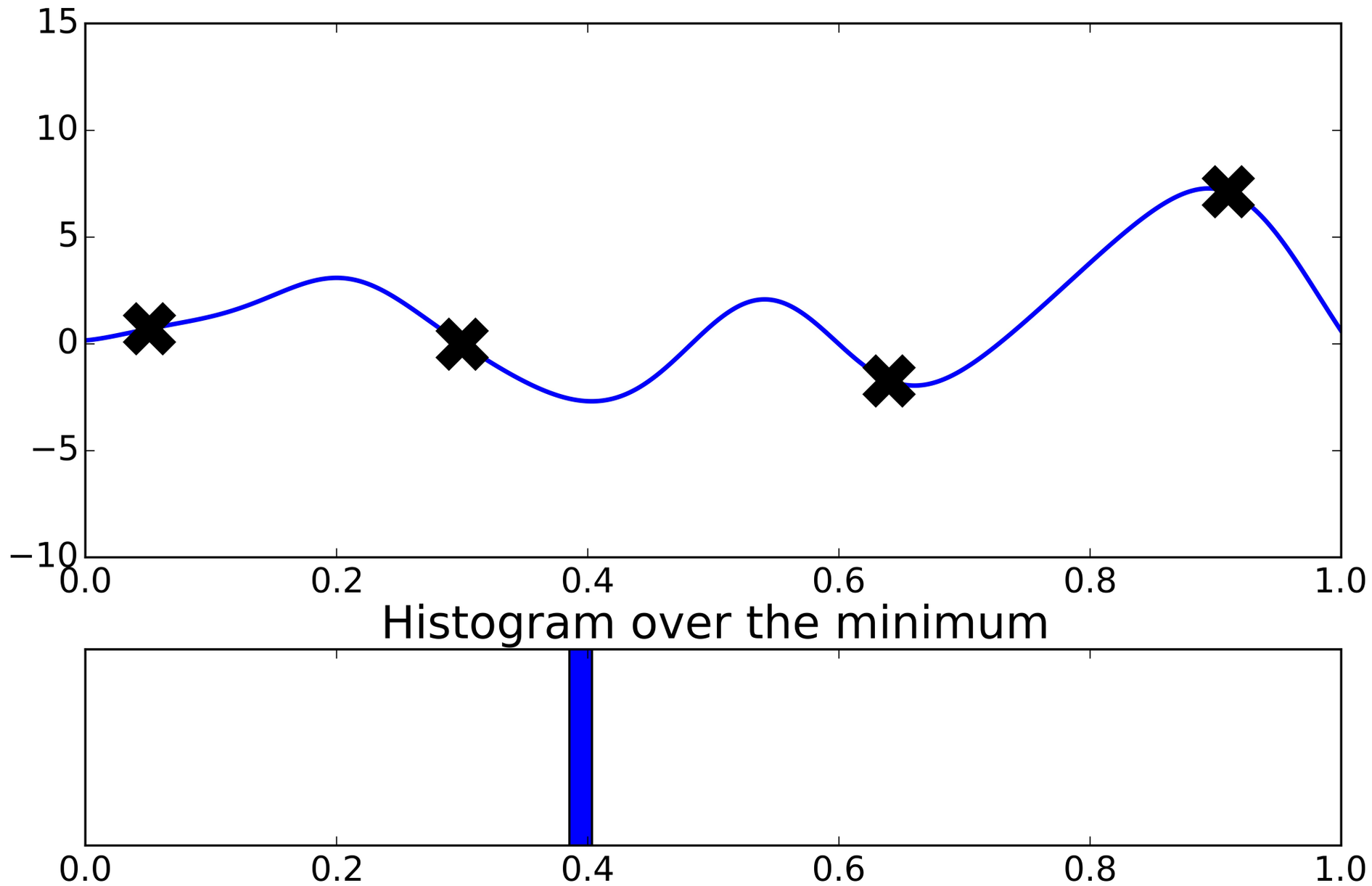
BAYESIAN OPTIMIZATION

INTUITION BEHIND IT - WE HAVE A FEW INITIAL FUNCTION EVALUATIONS



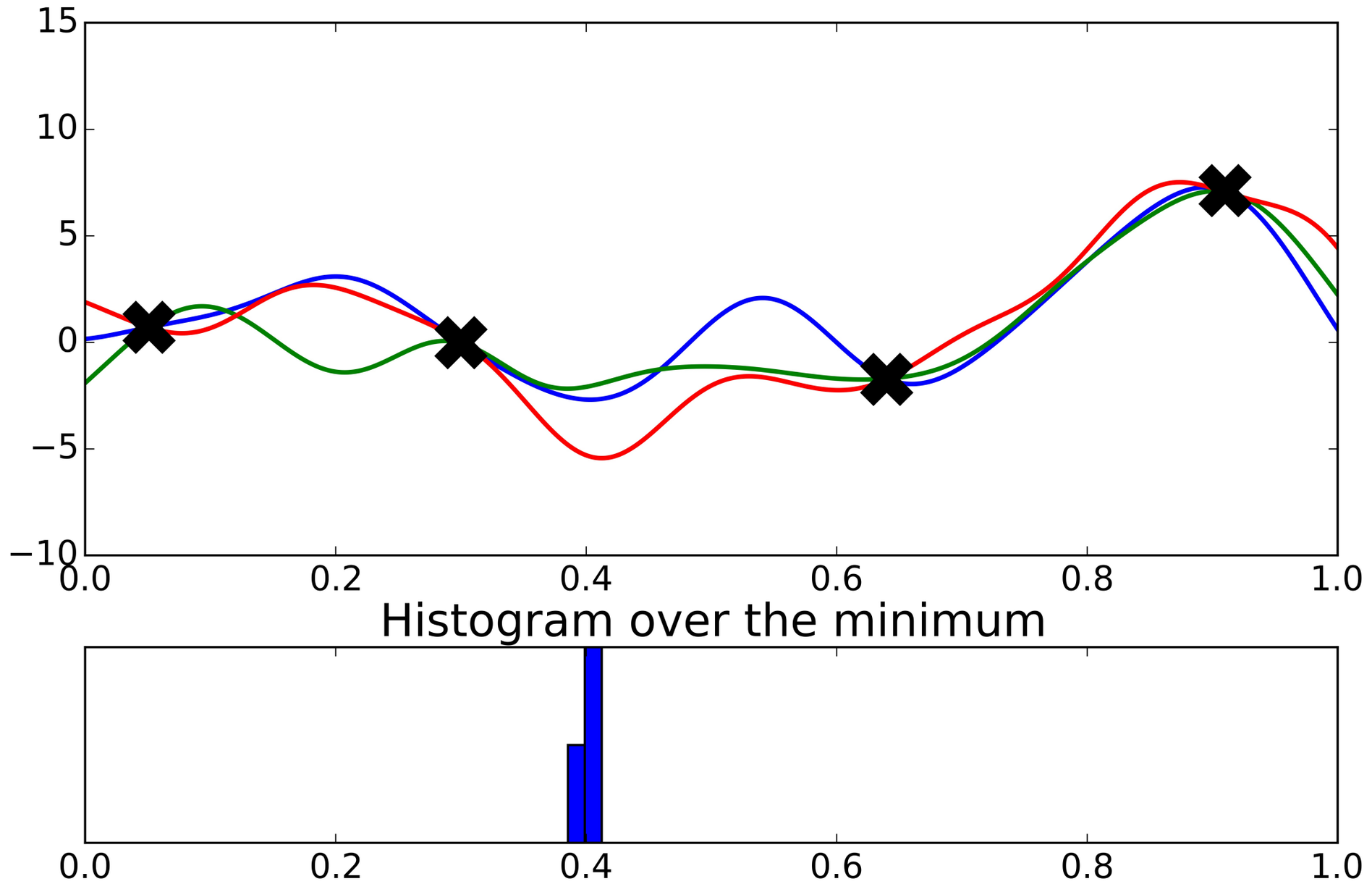
BAYESIAN OPTIMIZATION

INTUITION BEHIND IT - INTERPOLATE WITH 1 CURVE



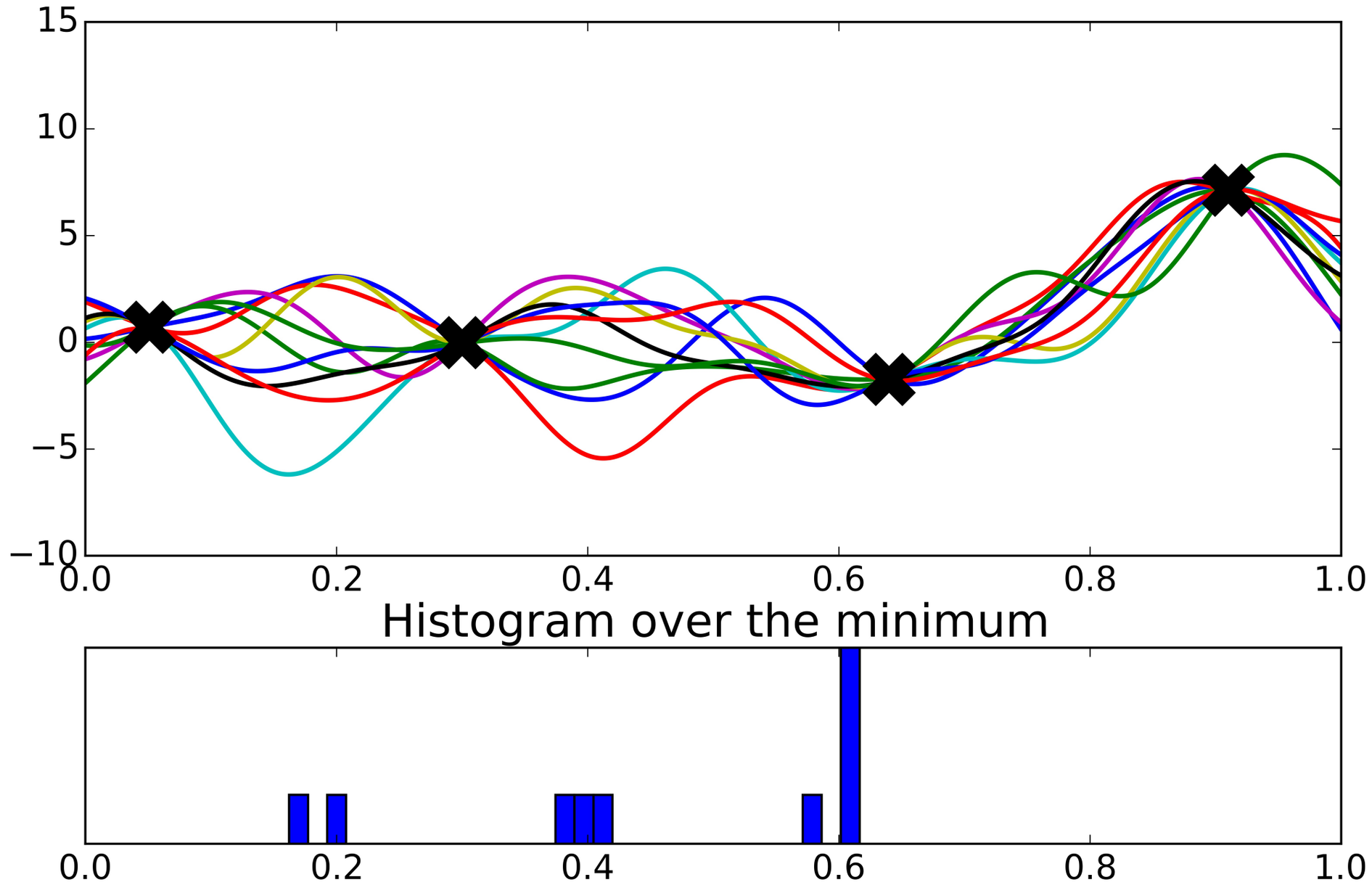
BAYESIAN OPTIMIZATION

INTUITION BEHIND IT - INTERPOLATE WITH 3 CURVES



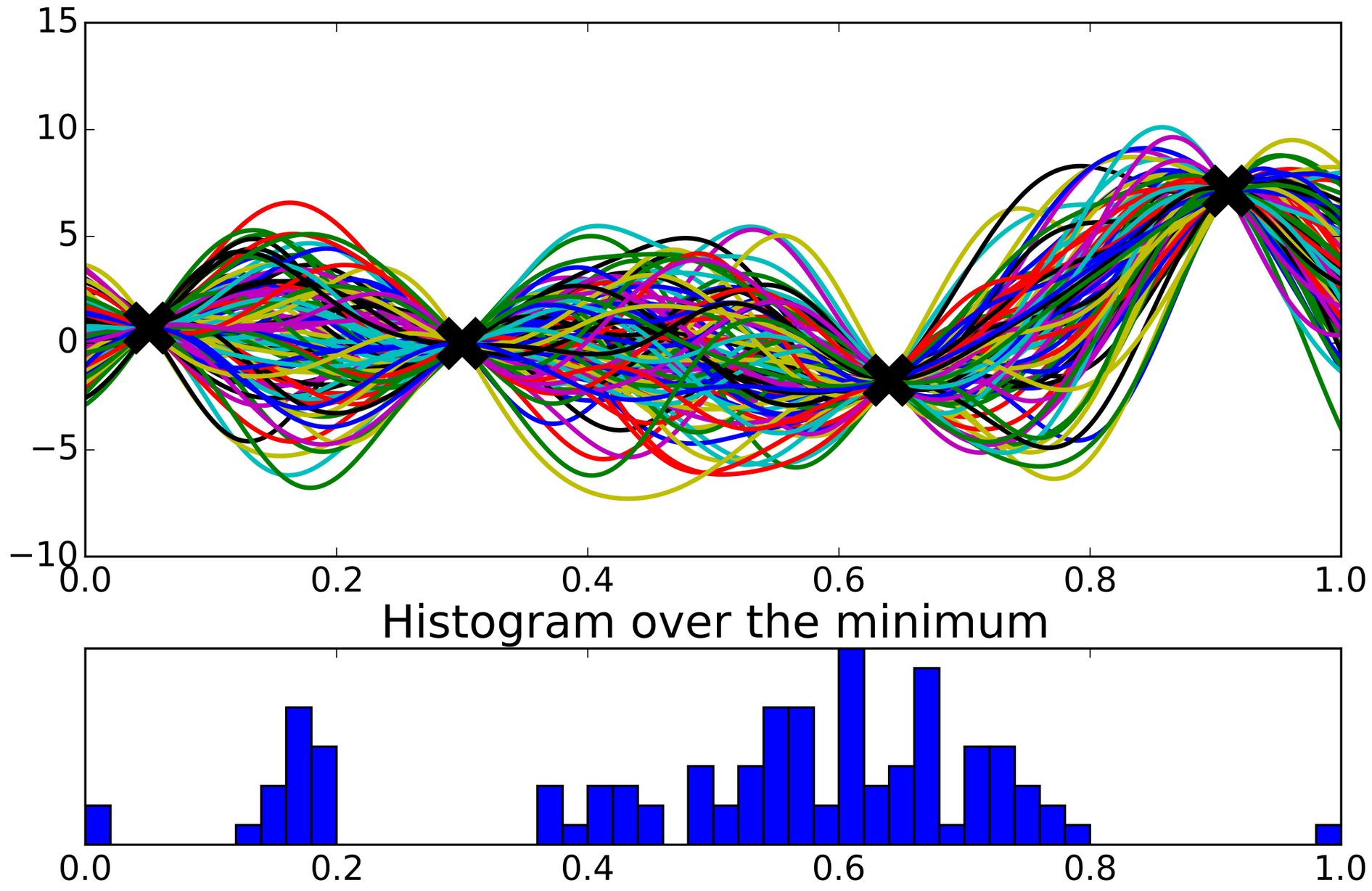
BAYESIAN OPTIMIZATION

INTUITION BEHIND IT - INTERPOLATE WITH 10 CURVES



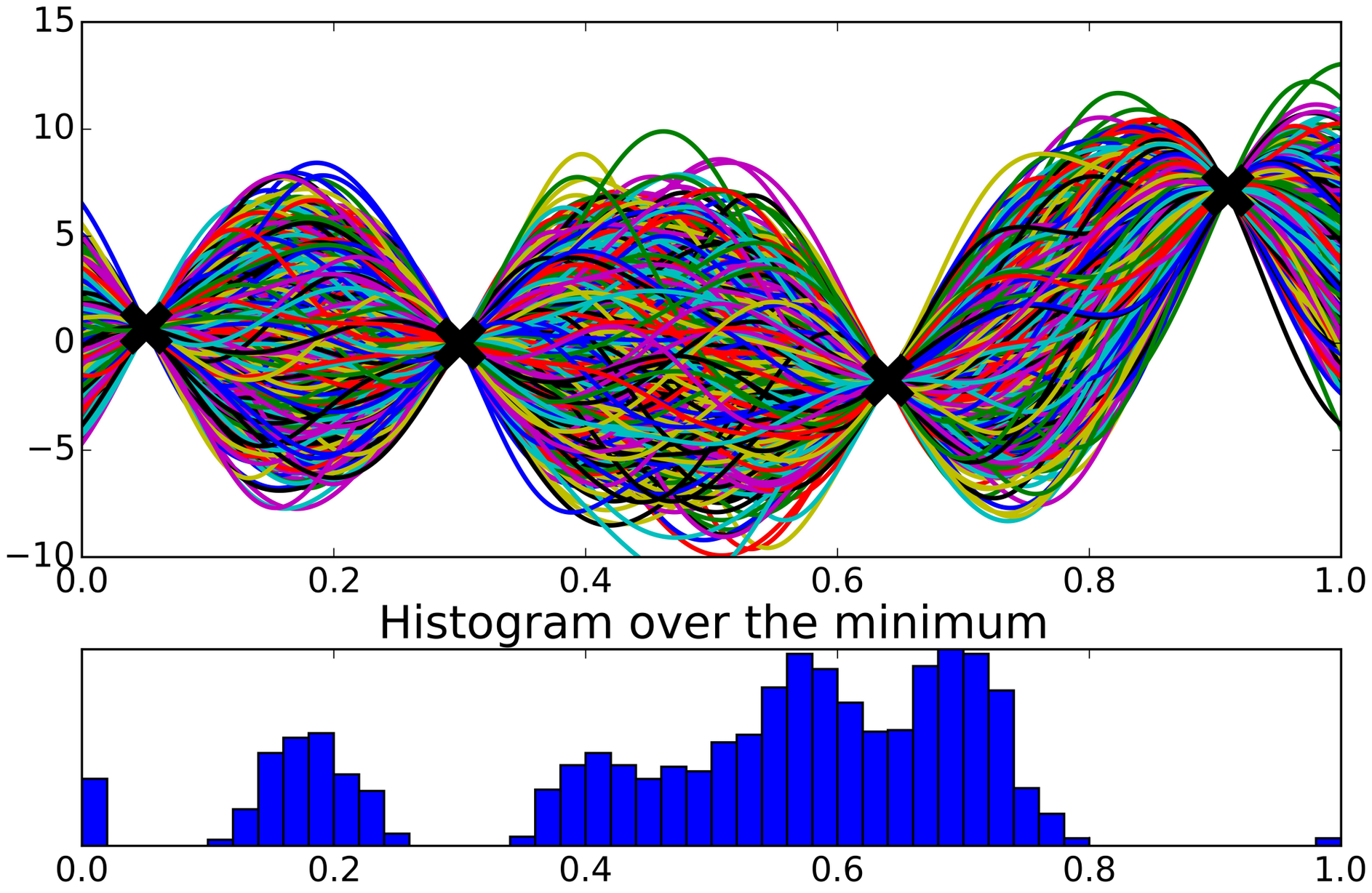
BAYESIAN OPTIMIZATION

INTUITION BEHIND IT - INTERPOLATE WITH 100 CURVES



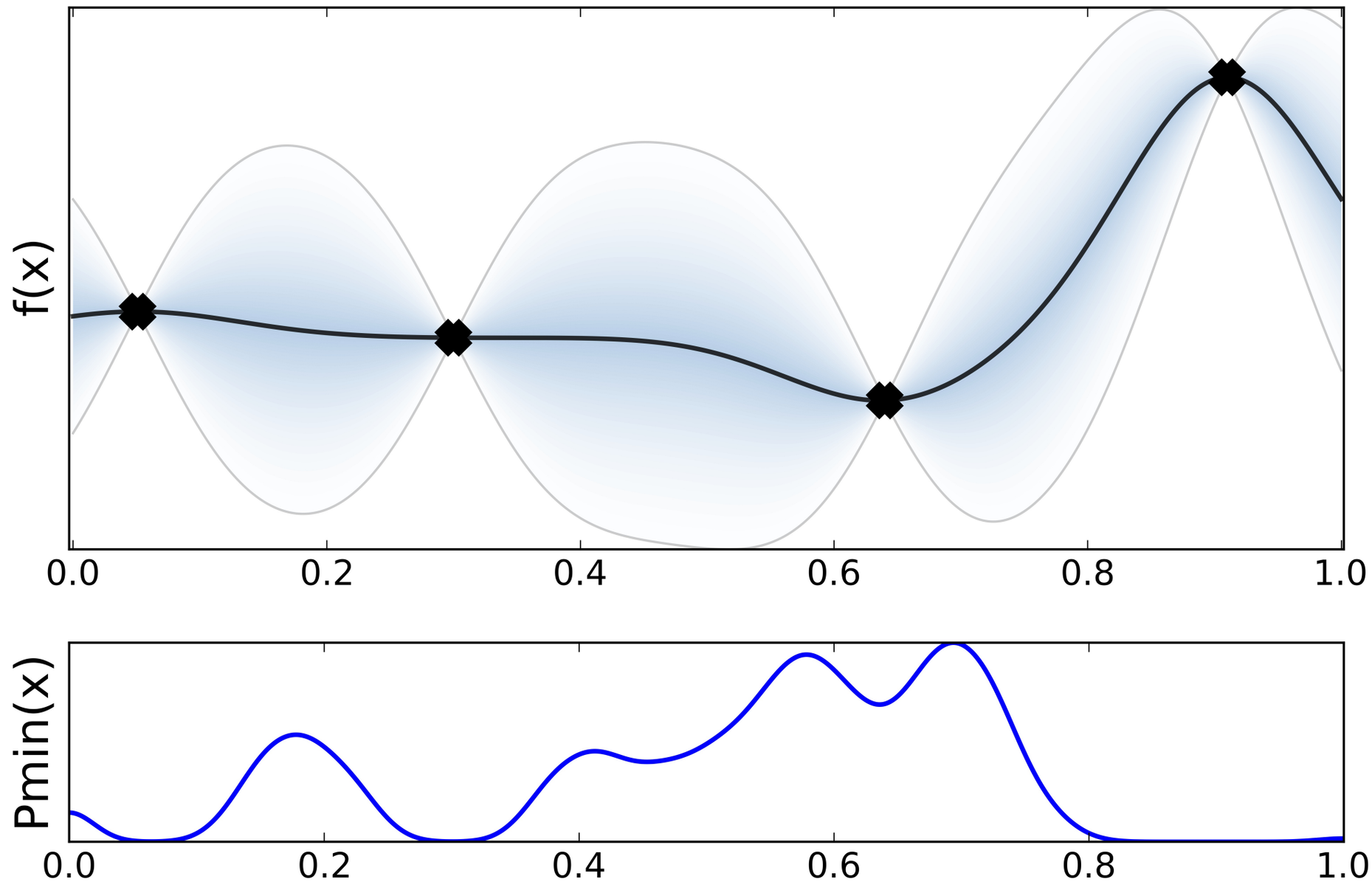
BAYESIAN OPTIMIZATION

INTUITION BEHIND IT - INTERPOLATE WITH 1000 CURVES



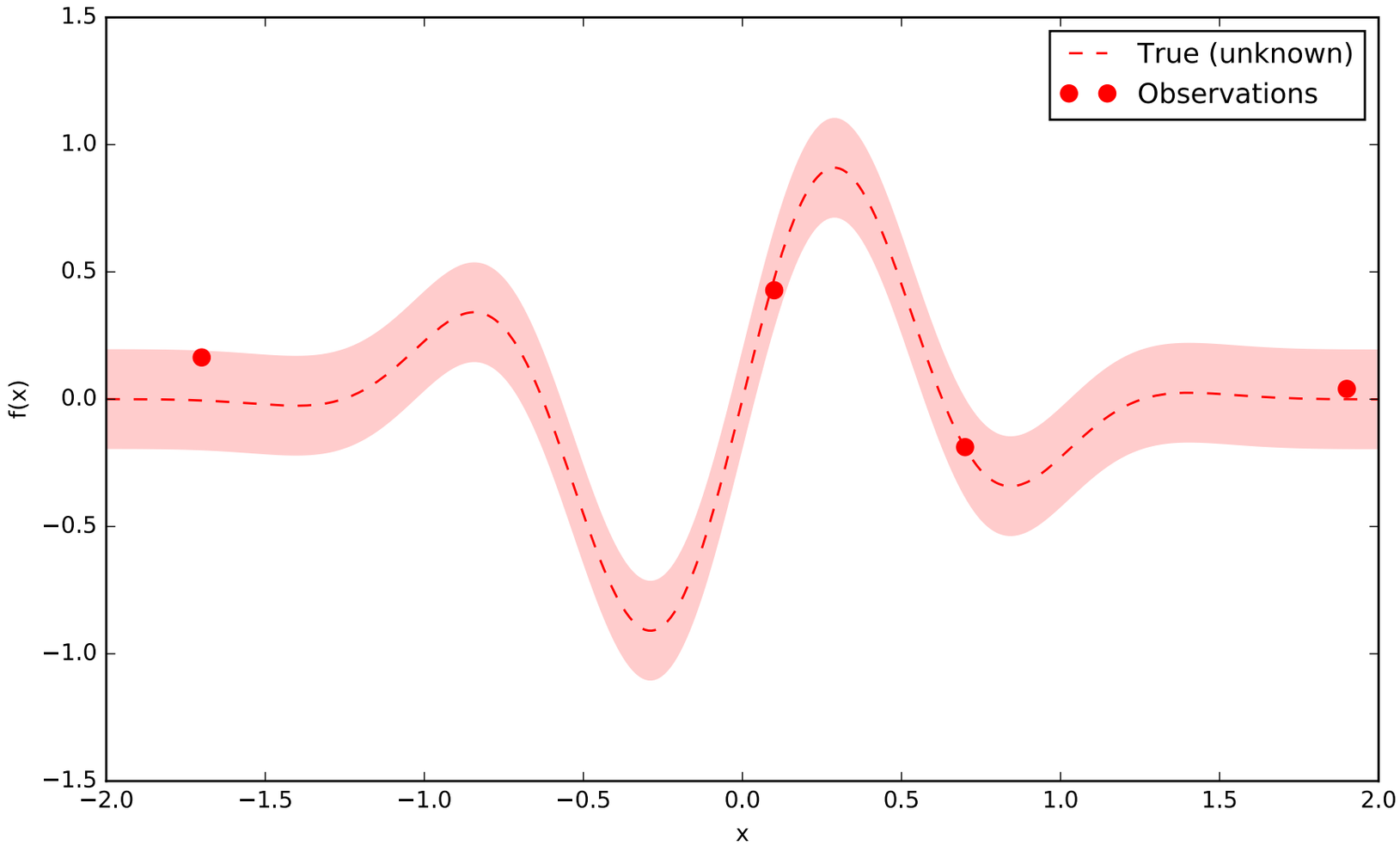
BAYESIAN OPTIMIZATION

INTUITION BEHIND IT - INTERPOLATE WITH "INFINITE" CURVES



BAYESIAN OPTIMIZATION

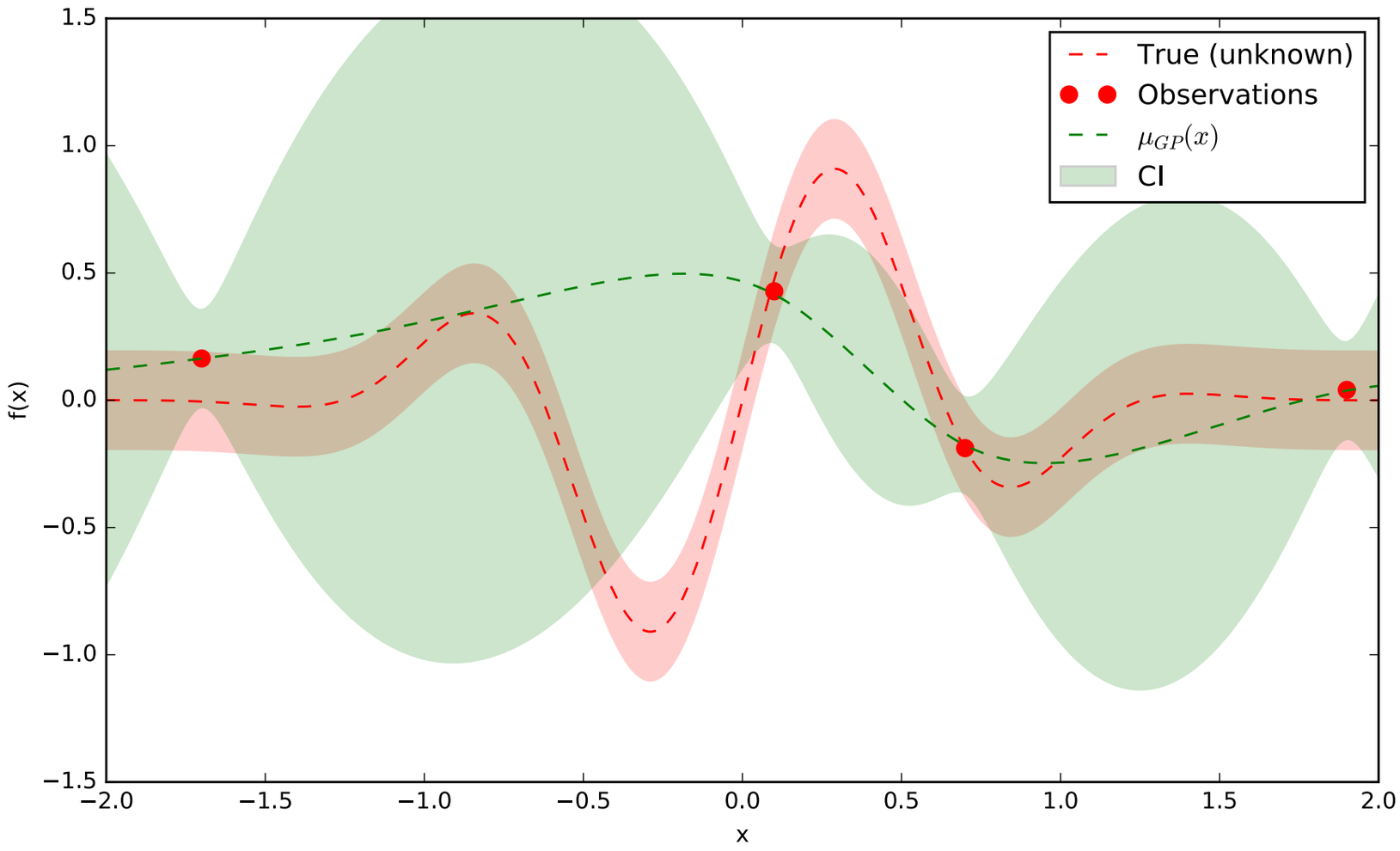
WHERE SHALL WE SAMPLE NEXT?



BAYESIAN OPTIMIZATION

BUILD A PROBABILISTIC MODEL FOR THE OBJECTIVE FUNCTION

This gives a *posterior* distribution over functions that could have generated the observed data.



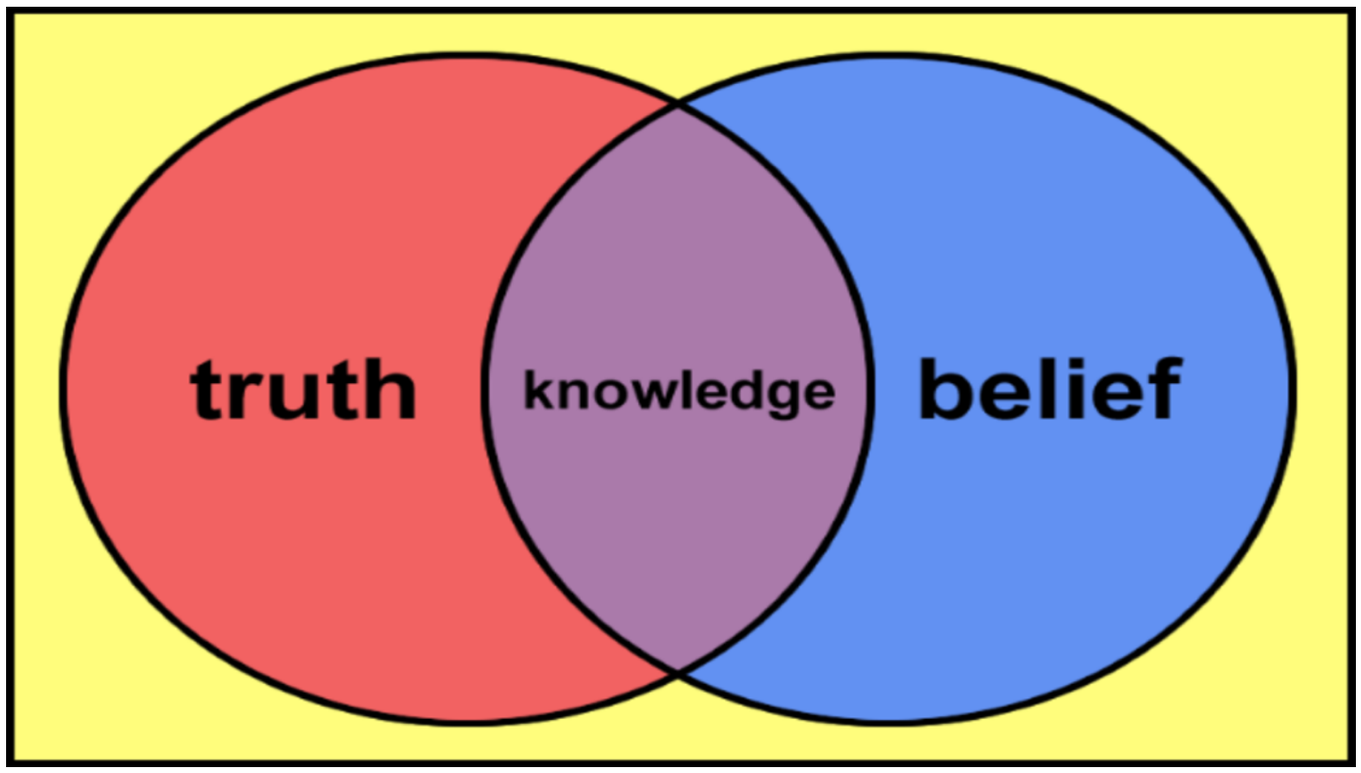
BAYESIAN OPTIMIZATION

BUILD A PROBABILISTIC MODEL FOR THE OBJECTIVE FUNCTION

- As mentioned, the probabilistic model is used as *surrogate* of $f(x)$ to carry out the optimization.

- The acquisition function is used to collect new data points satisfying some optimality criterion ("*optimization as decision*" paradigm).

- Such decision can be made as *inference* using the surrogate model: BO uses a probabilistic model able to calibrate both **epistemic** (aka systematic, i.e., due to lack of knowledge) and **aleatoric** (aka statistical, due to randomness) **uncertainty**.



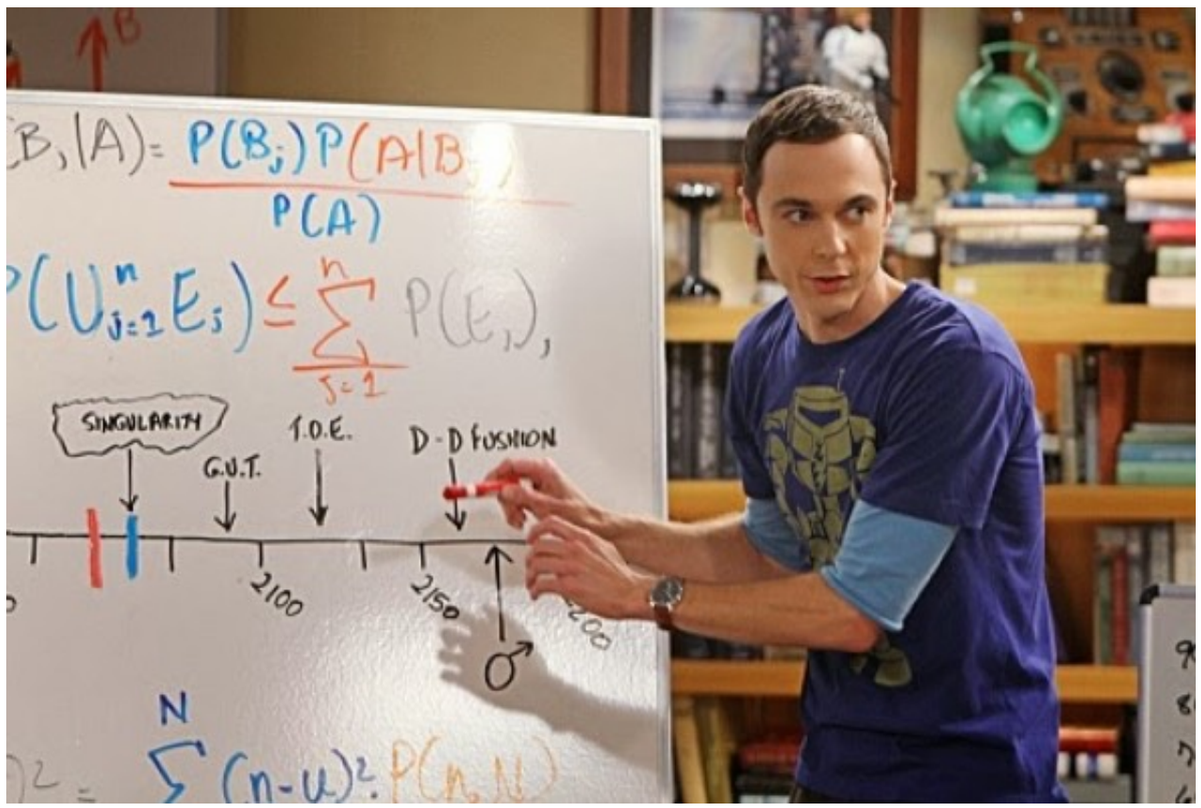
BAYESIAN OPTIMIZATION

WHAT IS BAYESIAN ABOUT BAYESIAN OPTIMIZATION?

- The Bayesian strategy treats the unknown objective function as a random function and places a *prior* over it. The prior captures our *beliefs* about the behavior of the function. As seen, it is usually defined by GP whose covariance function captures *assumptions* about the objective function.

- Function evaluations are treated as data. They are used to update the prior to form the *posterior* distribution over the objective function.

- The posterior distribution, in turn, is used to construct an *acquisition function* for querying the next point.



BAYESIAN OPTIMIZATION

ACQUISITION FUNCTIONS (NOTE: WE ASSUME MAXIMIZATION)

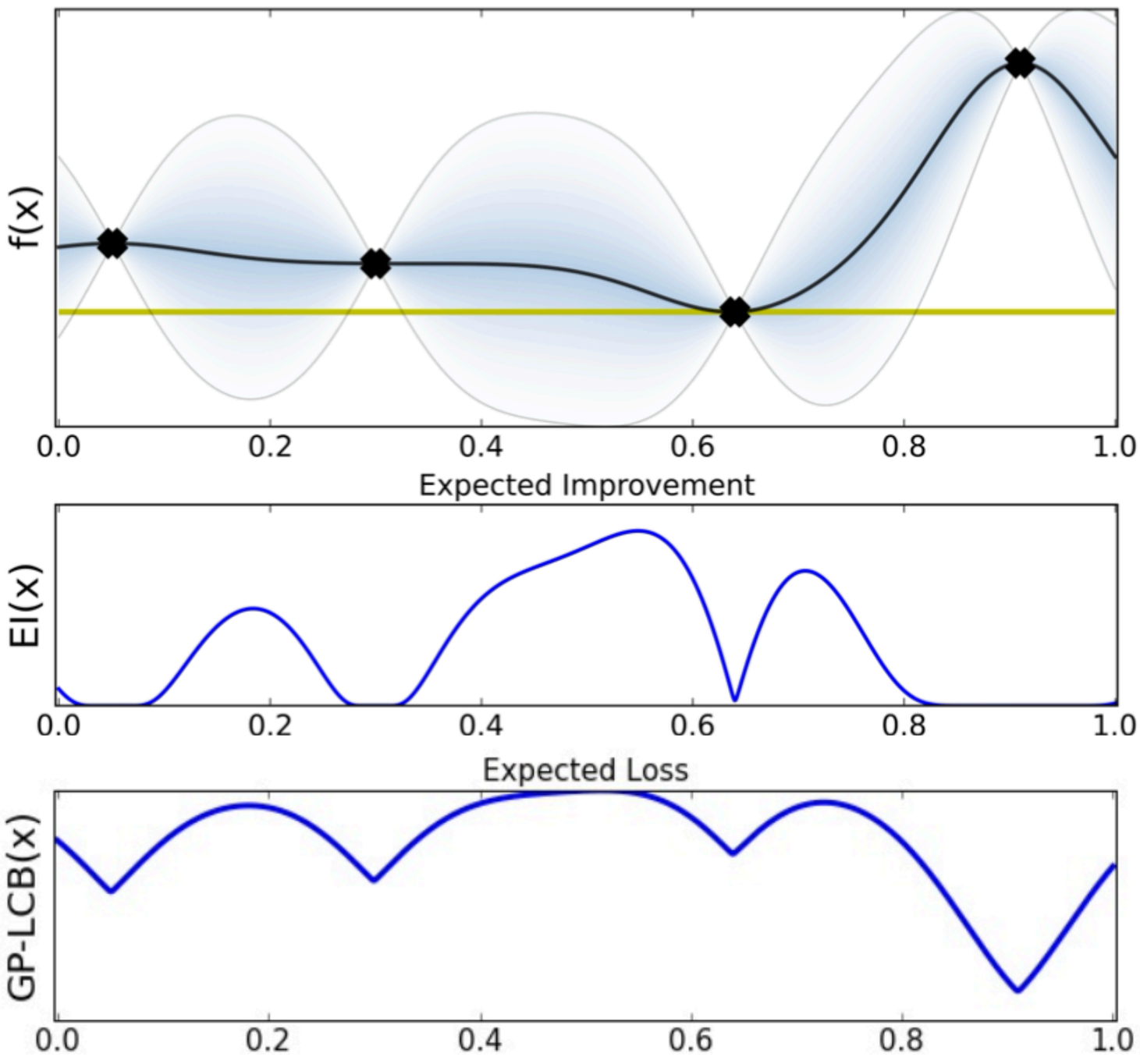
- An acquisition function $u(x)$ specifies which sample x should be tried next. Several possibilities:
  - Upper confidence bound $UCB(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$
    → for minimization: $LCB(x) = \mu_{GP}(x) - \kappa\sigma_{GP}(x)$
  - Probability of improvement $PI(x) = P(f(x) \geq f(x_t^+) + \kappa)$
  - Expected improvement $EI(x) = E[f(x) - f(x_t^+)]$ → most used acquisition
  - Maximum probability of improvement → first used acquisition, not so used in practice
  - Information-theoretic approaches (based on Thompson sampling aka probability matching)
  - … and many others!

  where x_t^+ is the best point observed so far.

- In most cases, acquisition functions provide knobs for controlling the exploration-exploitation trade-off. E.g., in the case of UCB the parameter κ controls the trade-off between:
  - Searching in regions where $\mu_{GP}(x)$ is high (exploitation)
  - Probing regions where uncertainty $\sigma_{GP}(x)$ is high (exploration)

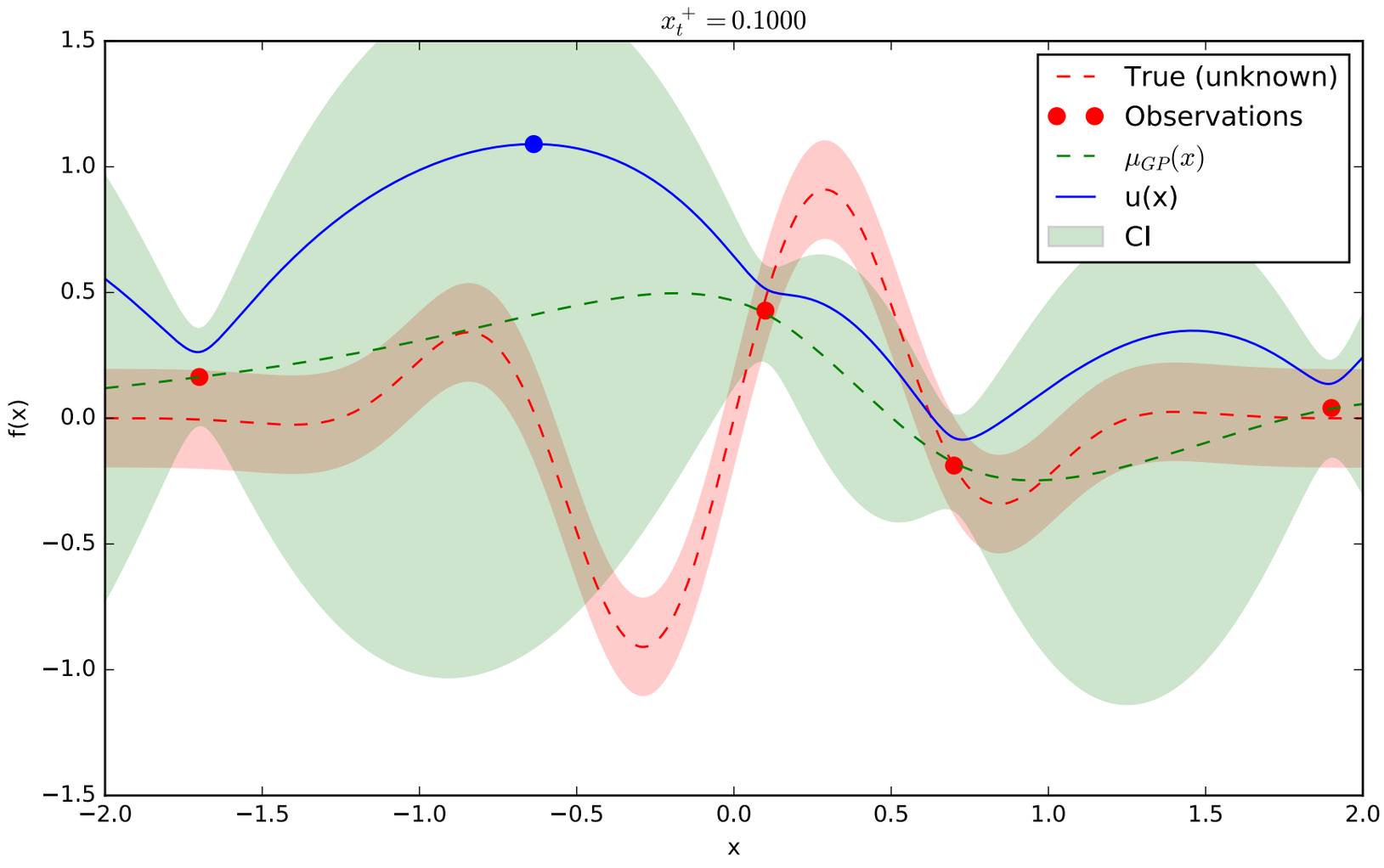
BAYESIAN OPTIMIZATION

ACQUISITION FUNCTIONS (NOTE: WE ASSUME MAXIMIZATION)



BAYESIAN OPTIMIZATION

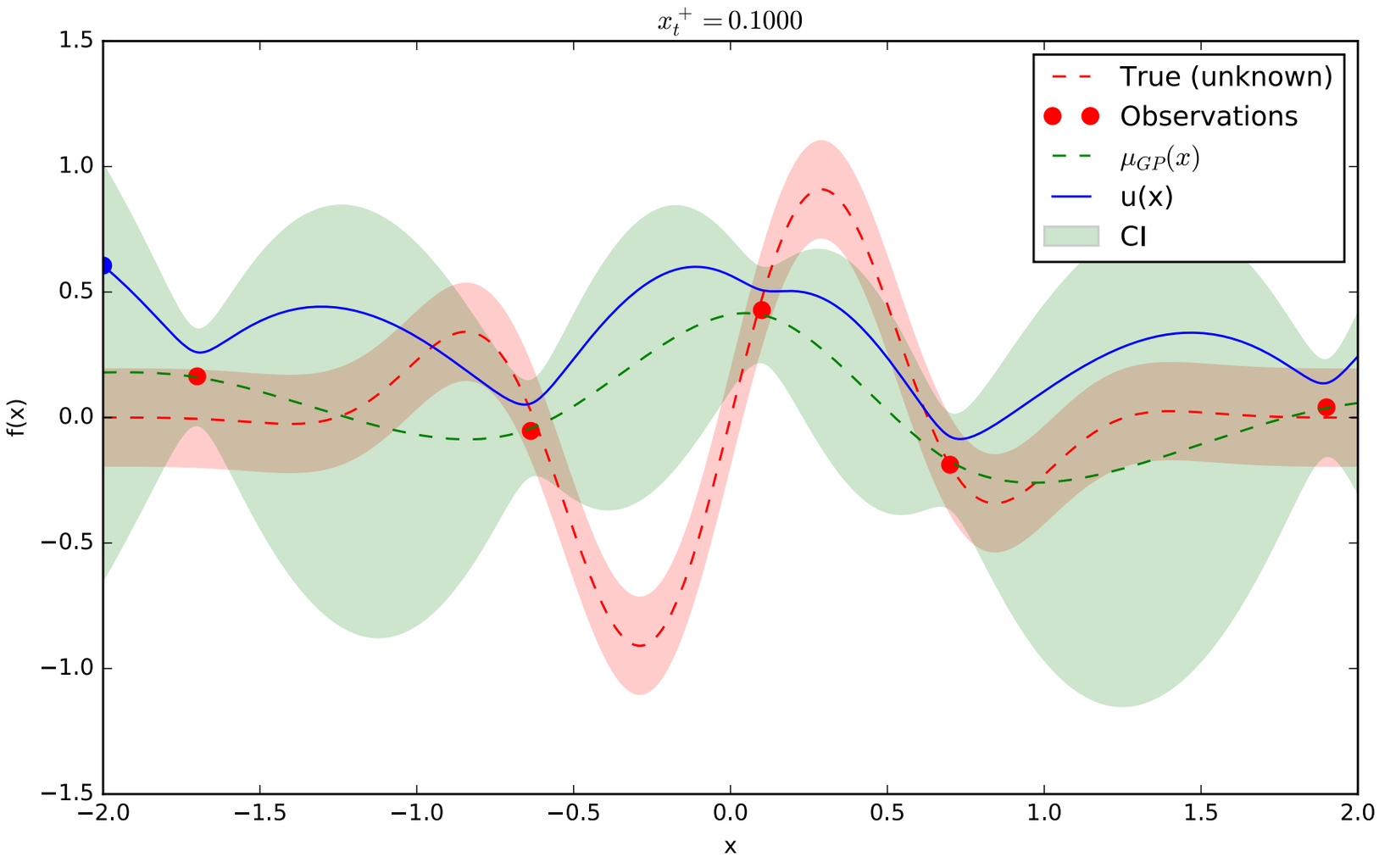
EXAMPLE (t = 0)



$$x_{t+1} = \arg\max_x \mathrm{UCB}(x)$$

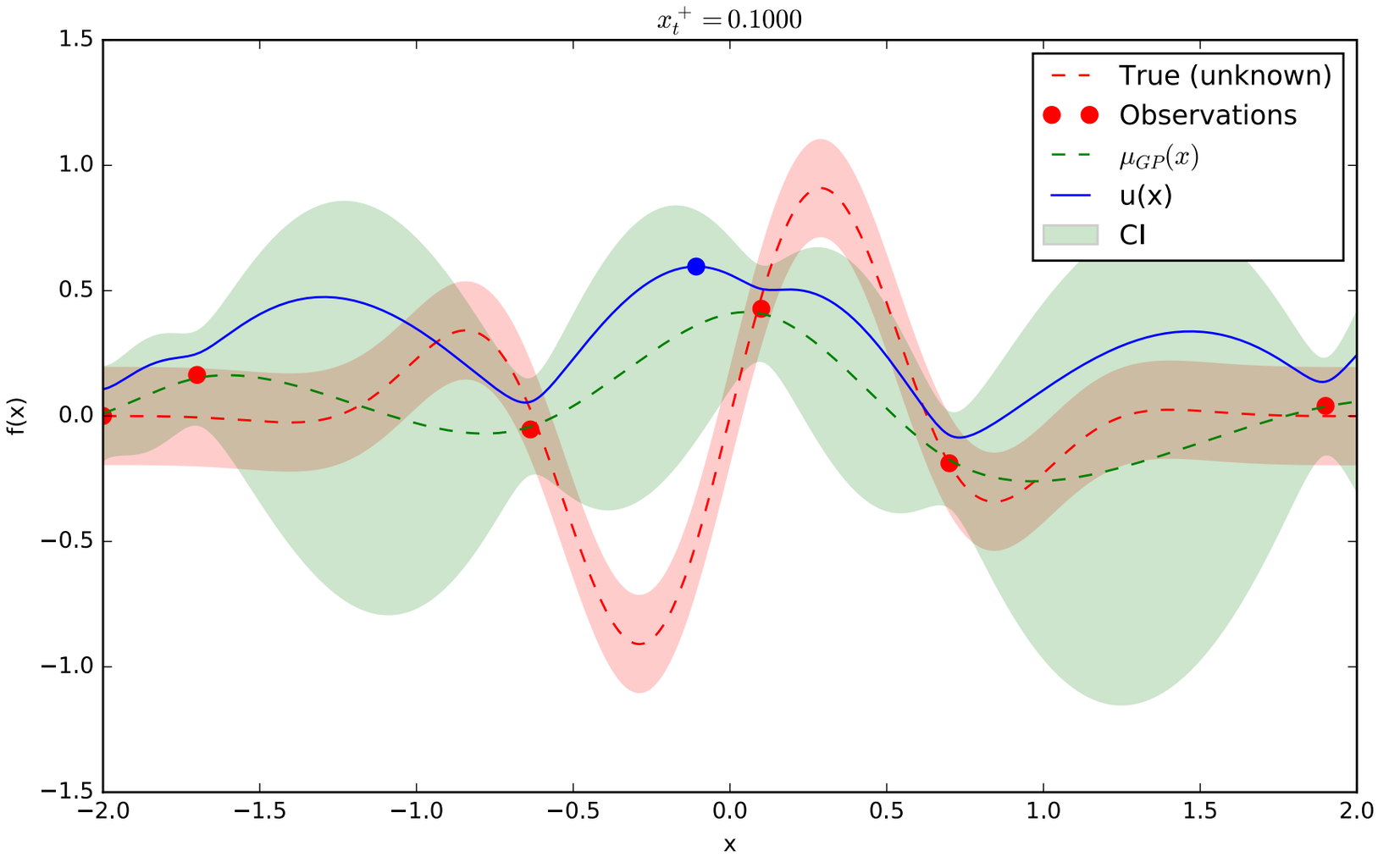
BAYESIAN OPTIMIZATION

EXAMPLE (t = 1)



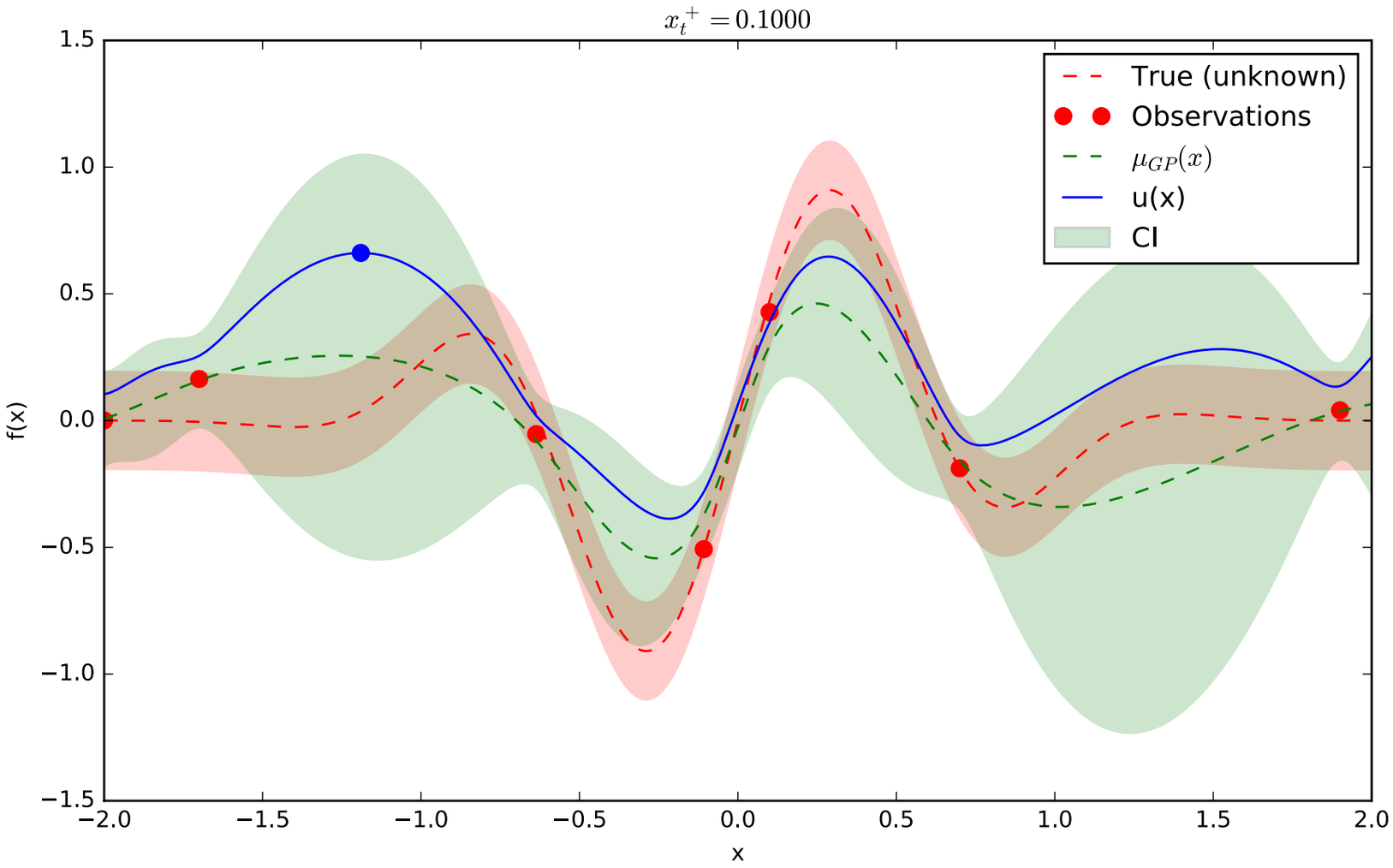
BAYESIAN OPTIMIZATION

EXAMPLE (t = 2)



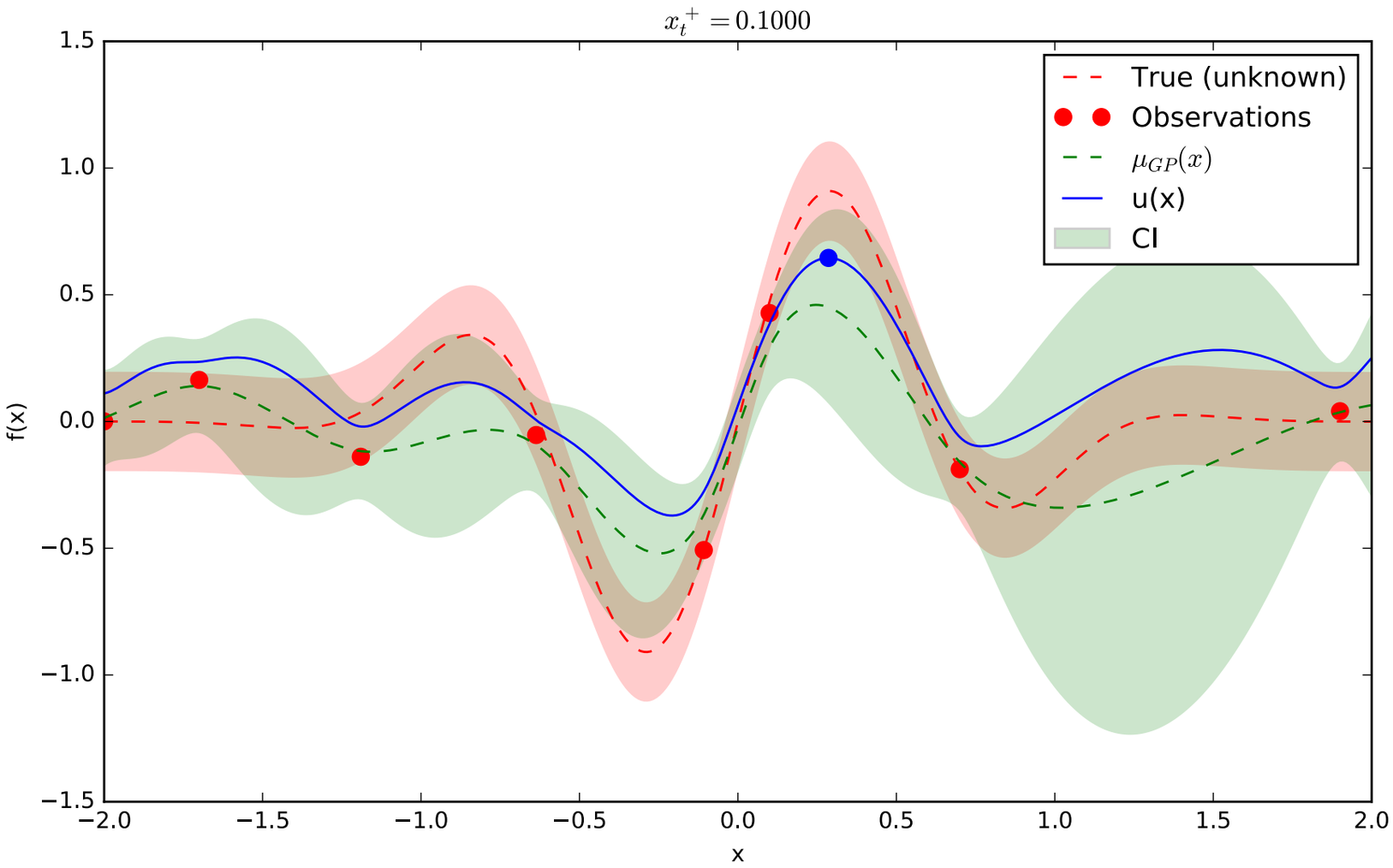
BAYESIAN OPTIMIZATION

EXAMPLE (t = 3)



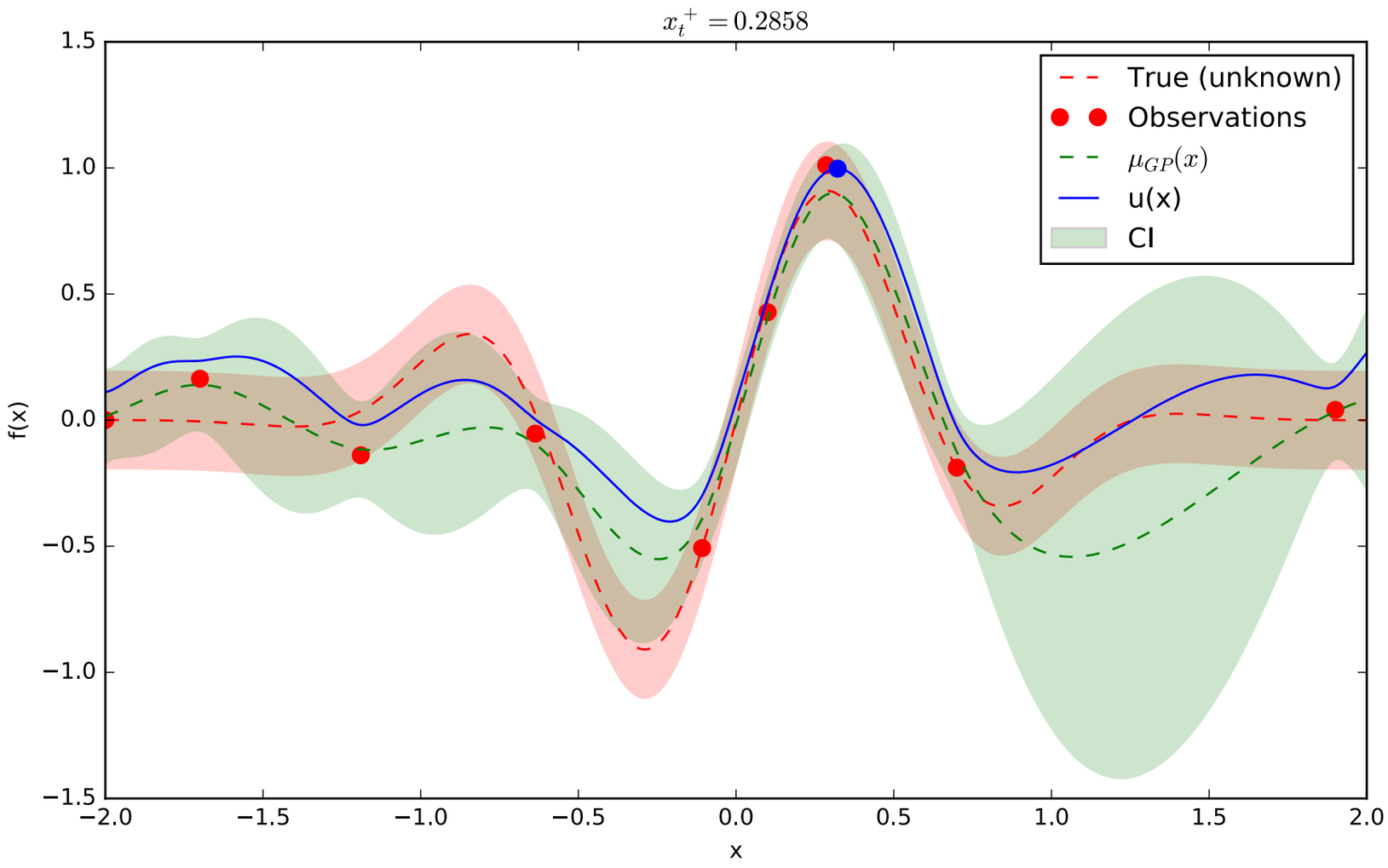
BAYESIAN OPTIMIZATION

EXAMPLE (t = 4)



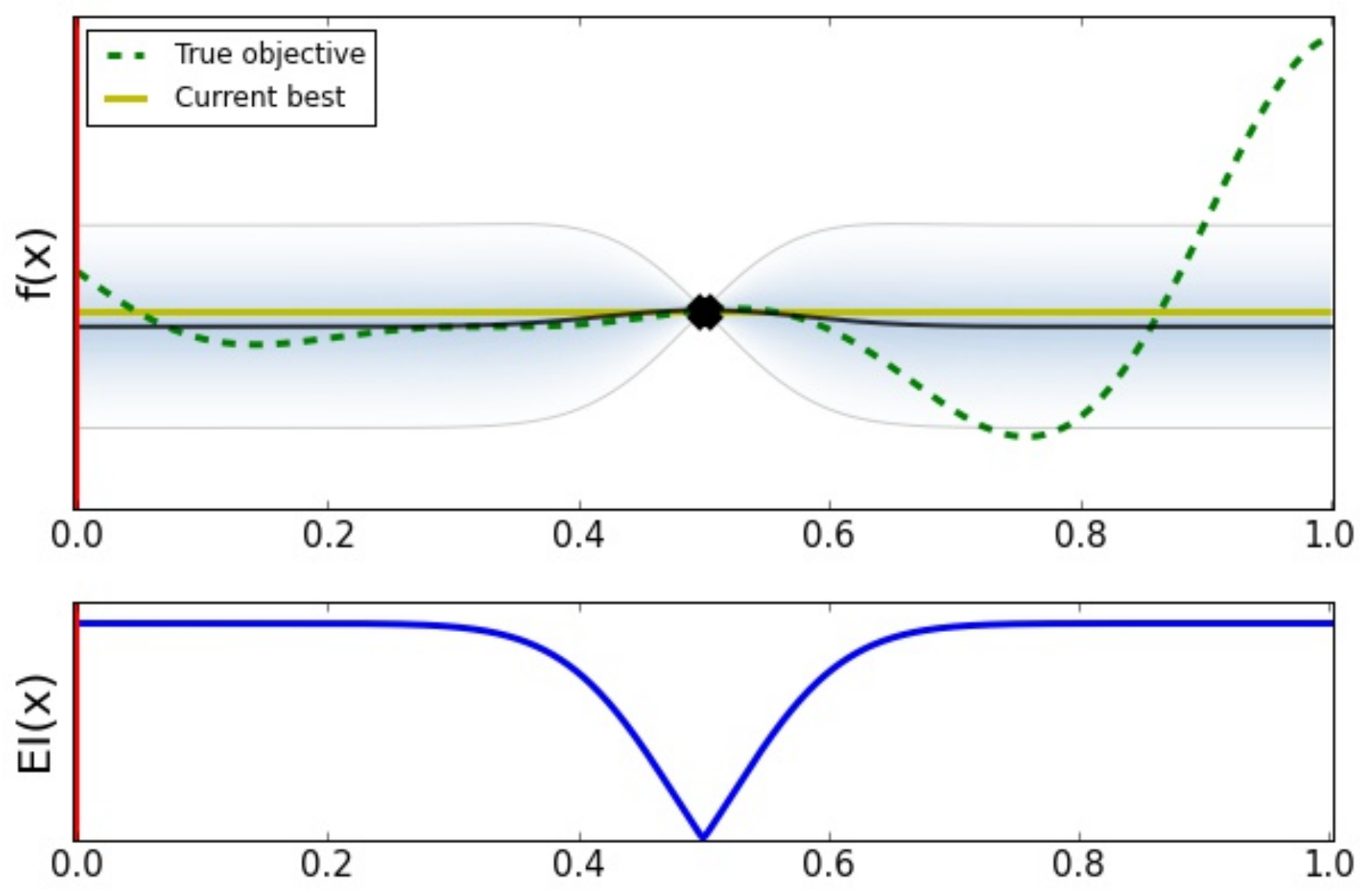
BAYESIAN OPTIMIZATION

EXAMPLE (t = 5)



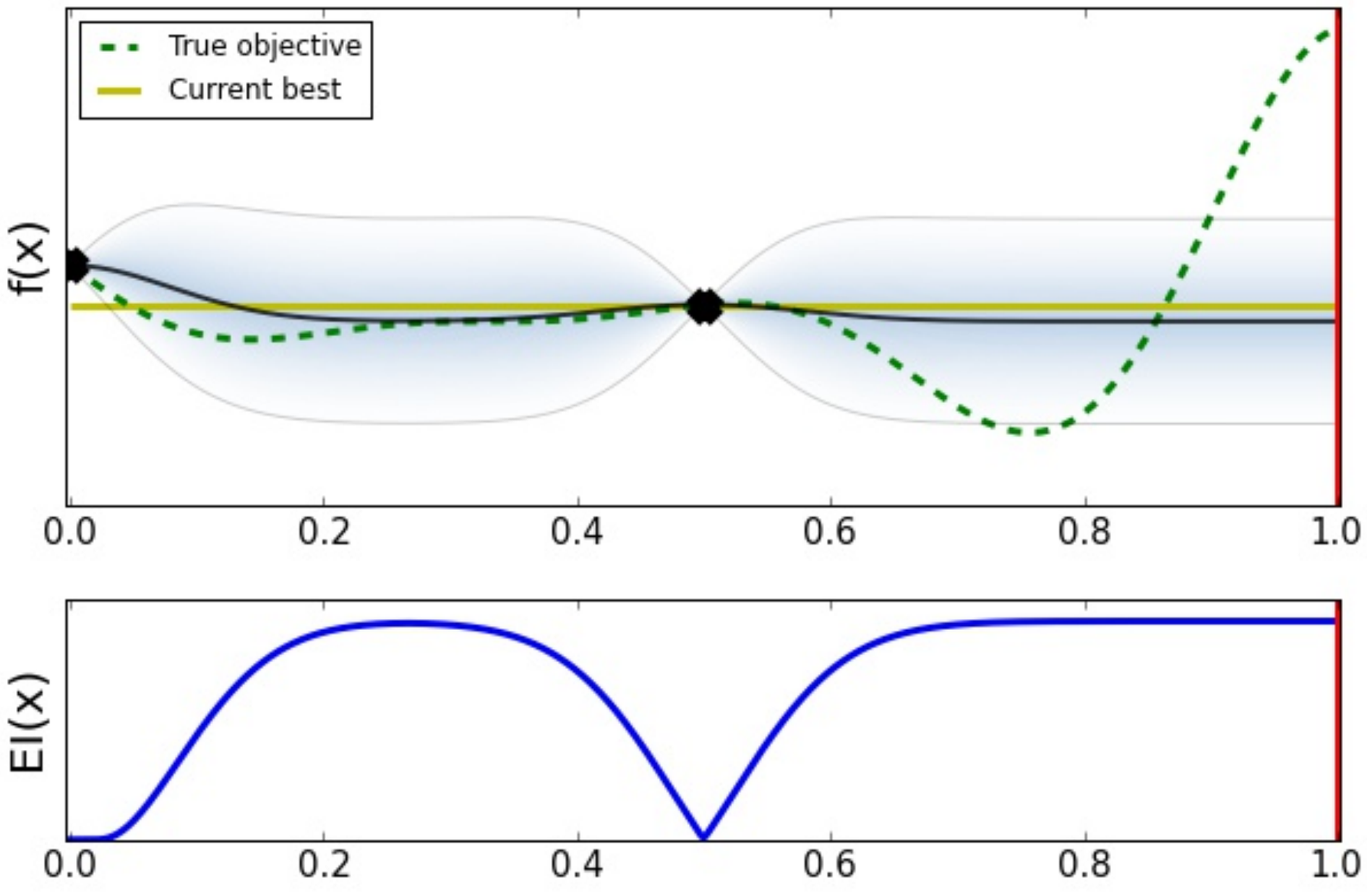
BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 0)



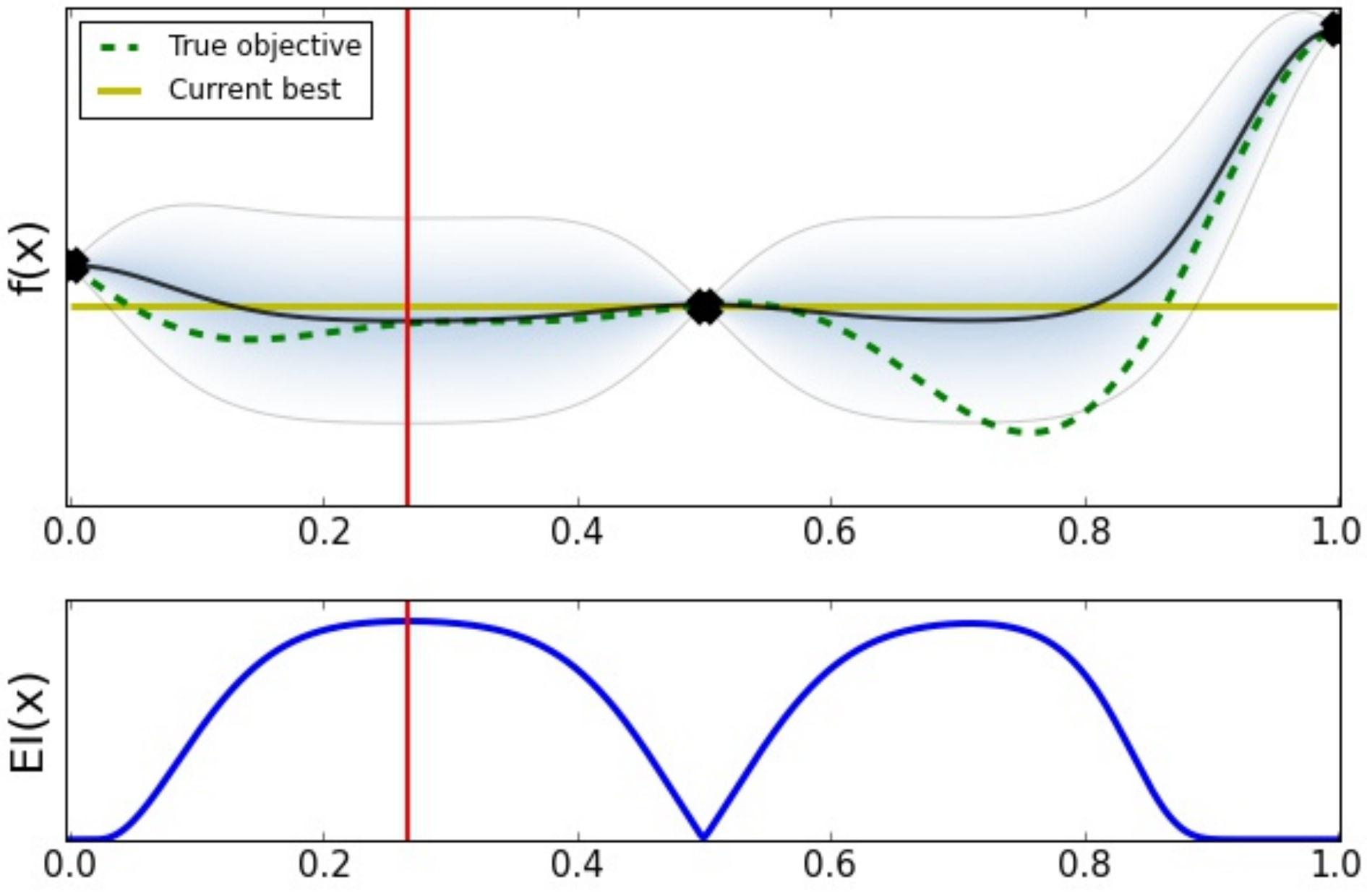
BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 1)



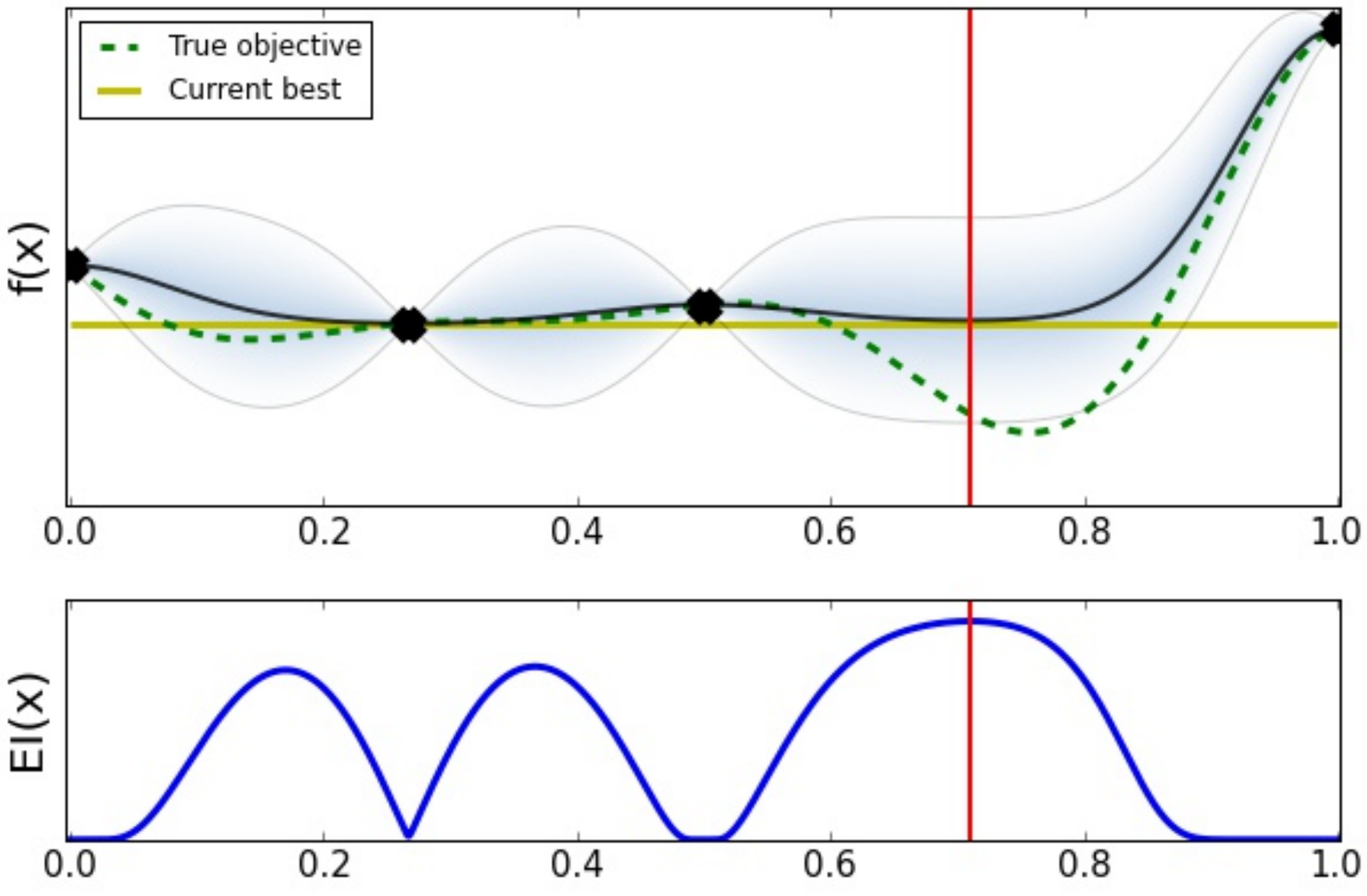
BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 2)



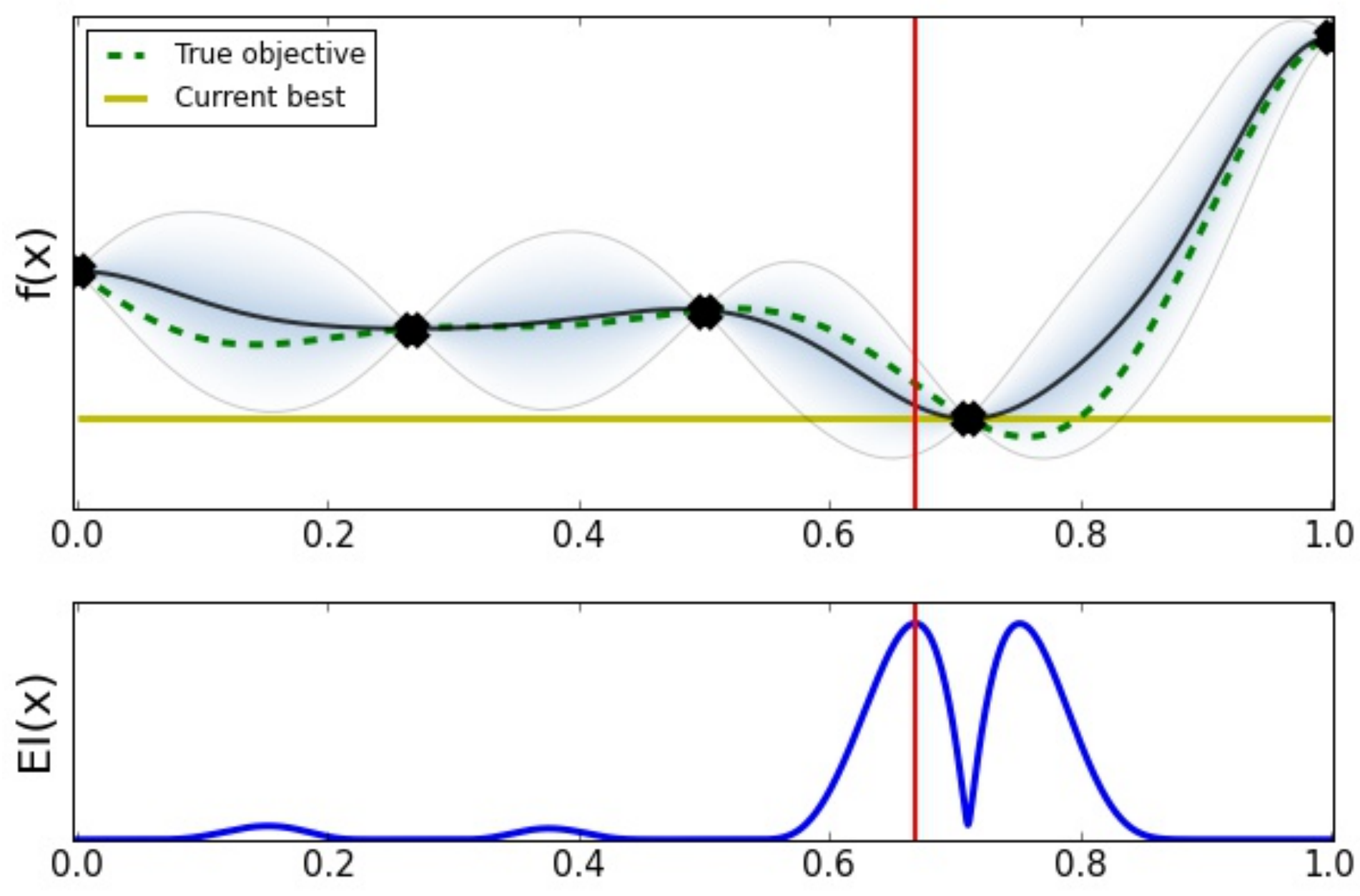
BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 3)



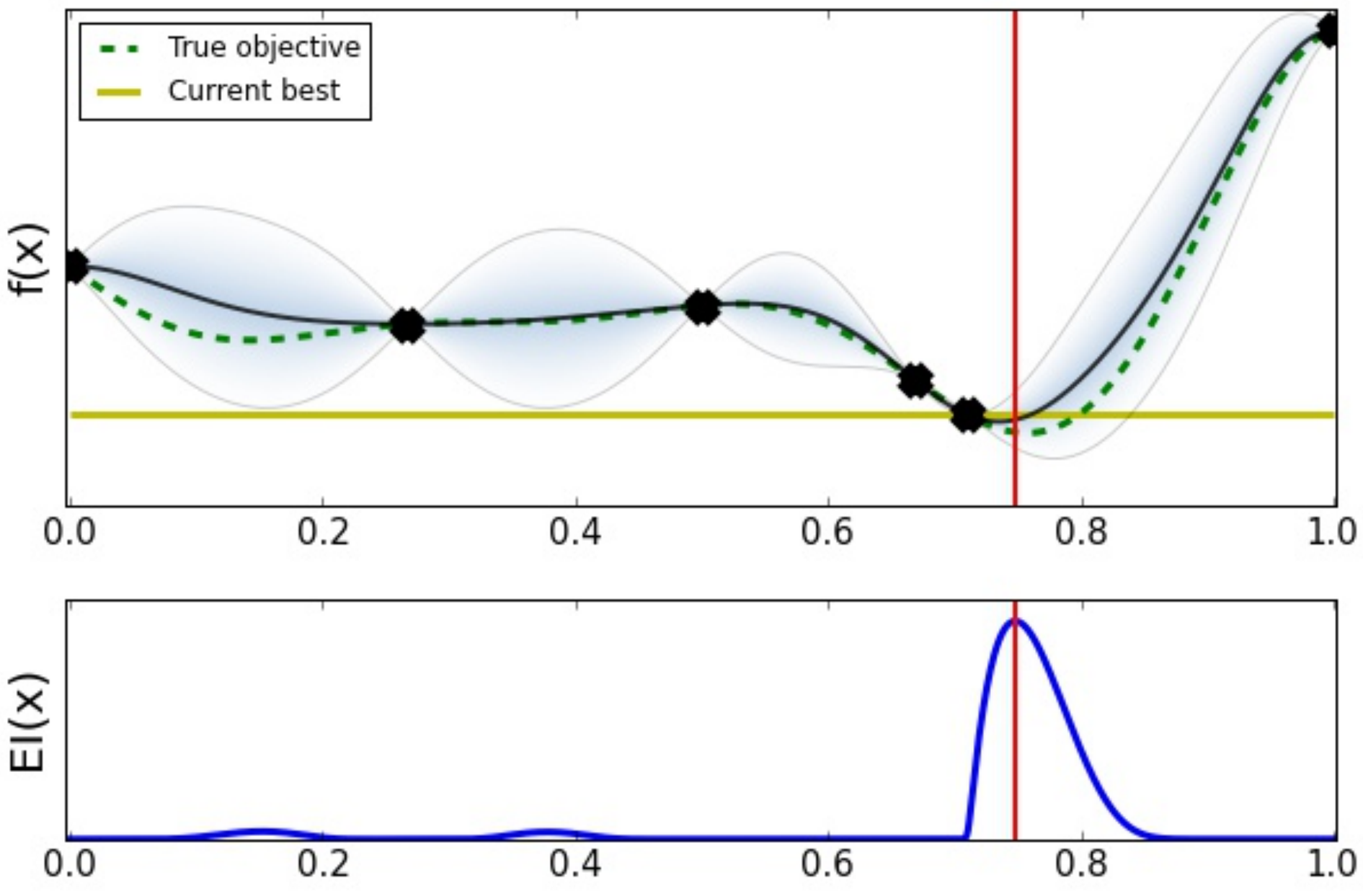
BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 4)



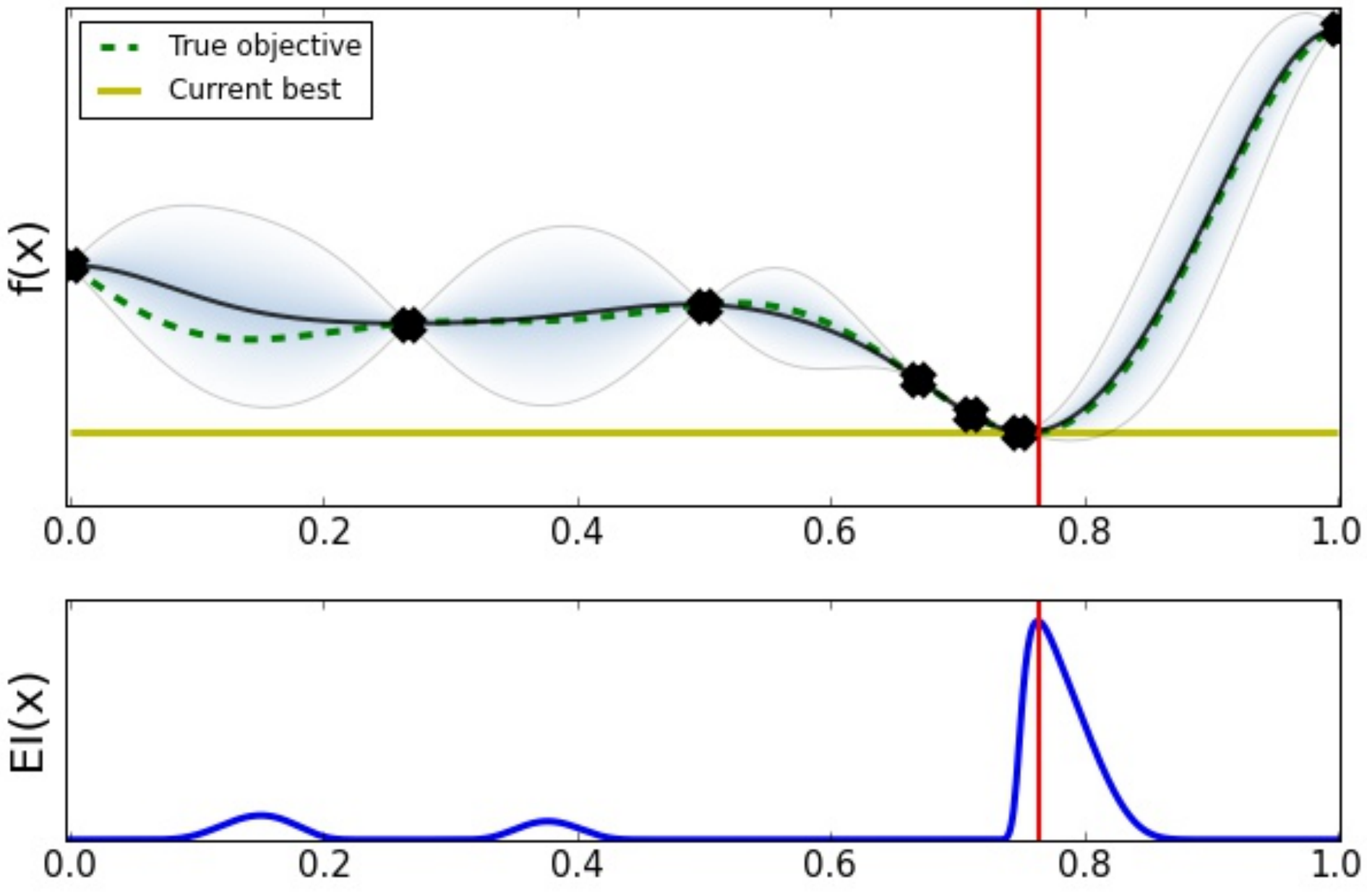
BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 5)



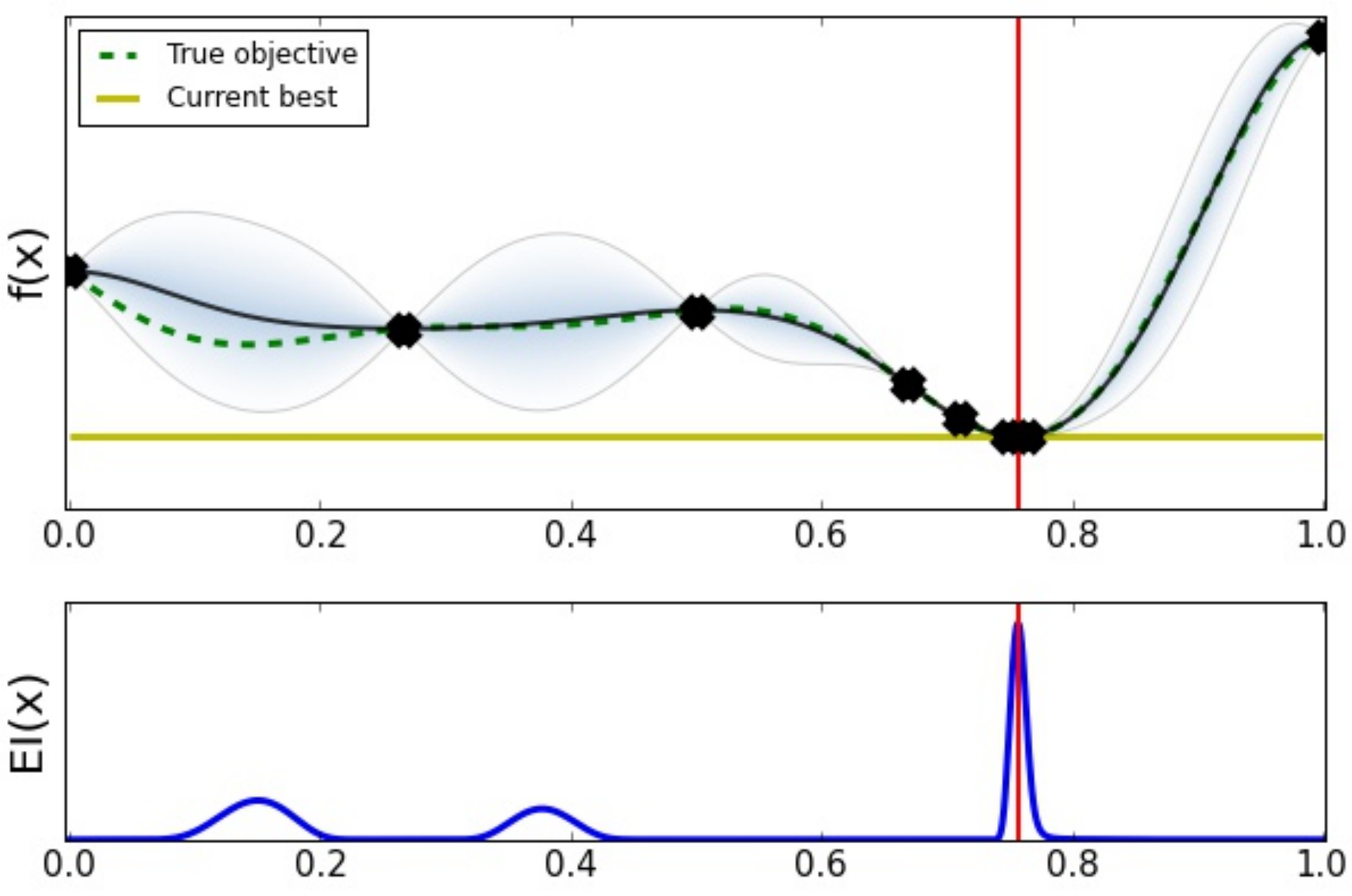
BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 6)



BAYESIAN OPTIMIZATION

ANOTHER EXAMPLE (t = 7)



BAYESIAN OPTIMIZATION

BAYESIAN OPTIMIZATION HAS PARAMETERS ITSELF!

Initialization scheme, i.e., how to select the initial points?

- One point in the centre of the domain.
- Uniformly selected random locations.
- More advanced methods (*we'll see some of them when we will talk about DOEs*):
  - Latin design, Halton sequences, determinantal point processes, etc.

The idea is always to start at some locations trying to minimize the initial model uncertainty.

BAYESIAN OPTIMIZATION

BAYESIAN OPTIMIZATION HAS PARAMETERS ITSELF!

- As seen, the surrogate is typically implemented via GP regression, which is fully determined by a mean function and a kernel. The de facto standard kernel is the *Squared Exponential* (aka *Radial Basis Function* or *Gaussian kernel*), with parameters σ and l:

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

  - Many other kernels are possible, with different properties (see e.g. https://www.cs.toronto.edu/~duvenaud/cookbook/).
  - How to choose the right kernel and optimize its hyper-parameters (if any)?

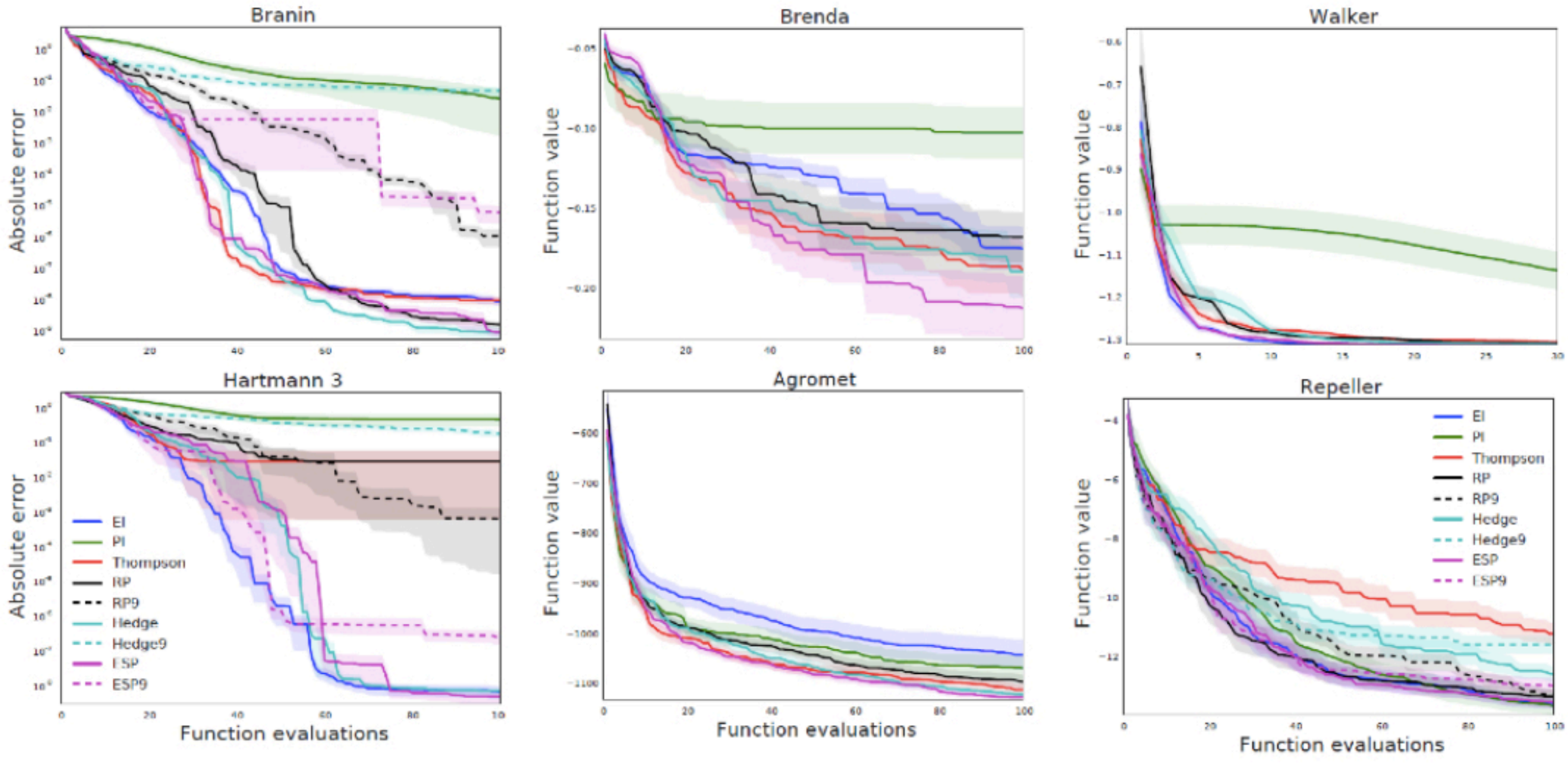
- Moreover, GPs is known to not scale so well to many observations and to high-dimensional data (this is because, in order to be fit, they need to calculate a matrix inversion). In general, GP works fine up to ~20 dimensions. Some alternatives are:
  - Sparse GP
  - Sequential model-based optimization (that, instead of GP, uses a tree-based model, such as random forests, or t-student processes)
  - Which model works best?

BAYESIAN OPTIMIZATION

BAYESIAN OPTIMIZATION HAS PARAMETERS ITSELF!

How to choose the acquisition function? → The best acquisition function depends on the problem and the level of exploration/exploitation required.

How to optimize it? → Typically, with gradient-descent methods (e.g., BFGS), DIRECT, or evolutionary algorithms (e.g., CMA-ES). In principle, these methods may also be used to directly optimize $f(x)$.



BAYESIAN OPTIMIZATION

CONCLUDING REMARKS

- Bayesian optimization provides a principled approach for optimizing an expensive function $f(x)$.
- It is a way of **encoding our beliefs** about a property of a function (e.g., where the optimum is, which kind of shape the function has, its smoothness, etc.).
- Two key elements: the **probabilistic model** and the **acquisition function**.
- Many choices in both cases, especially in terms of the acquisition function used.
- The key is to find a **good balance between exploration and exploitation**.
- Often very effective, provided it is itself properly configured.
- Limitation: **reduced scalability** in dimensions and number of evaluations (this is still a problem).
- Hot topic in Machine Learning research. Expect quick improvements!

BAYESIAN OPTIMIZATION

POSSIBLE EXTENSIONS

- Bayesian optimization is mostly meant for **continuous optimization**.

- However, it is possible to extend it to handle other kinds of optimization problems, such as **multi-objective problems**, or problems with **integer-valued and categorical variables.** Note that the majority of the kernels of GP assume real-valued variables. The solution is to create a transformation that allows a valid GP kernel to deal with integer and categorical-valued variables.

- **Bayesian optimization evaluated in paralle**l: standard BO evaluates one solution at a time. This may be inefficient. Parallel BO implementations have been recently proposed.

BAYESIAN OPTIMIZATION

FURTHER READING

- Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599.

- 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.

- https://distill.pub/2020/bayesian-optimization/

Questions?