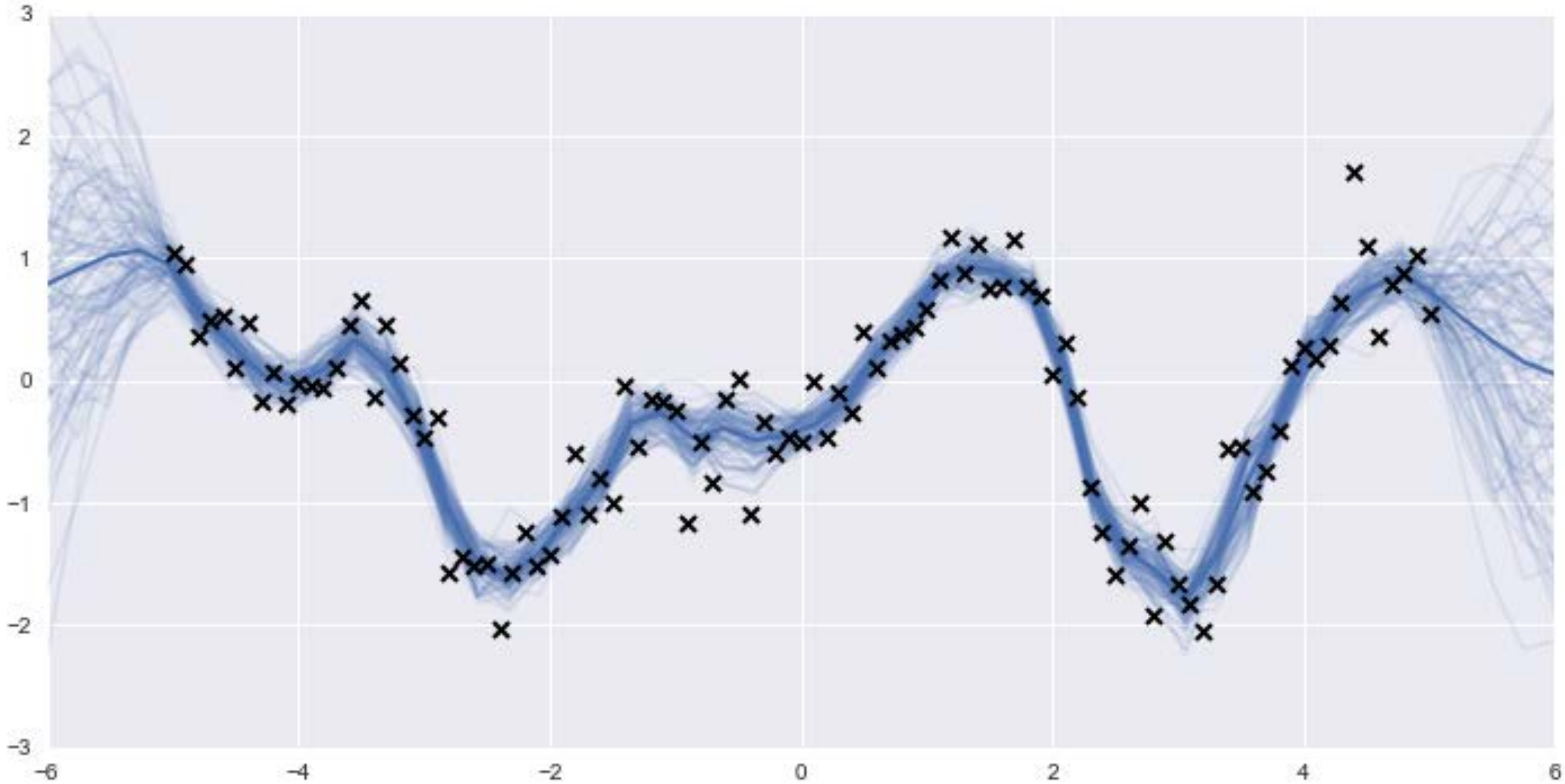
ENM 540: Data-driven modeling and probabilistic scientific computing

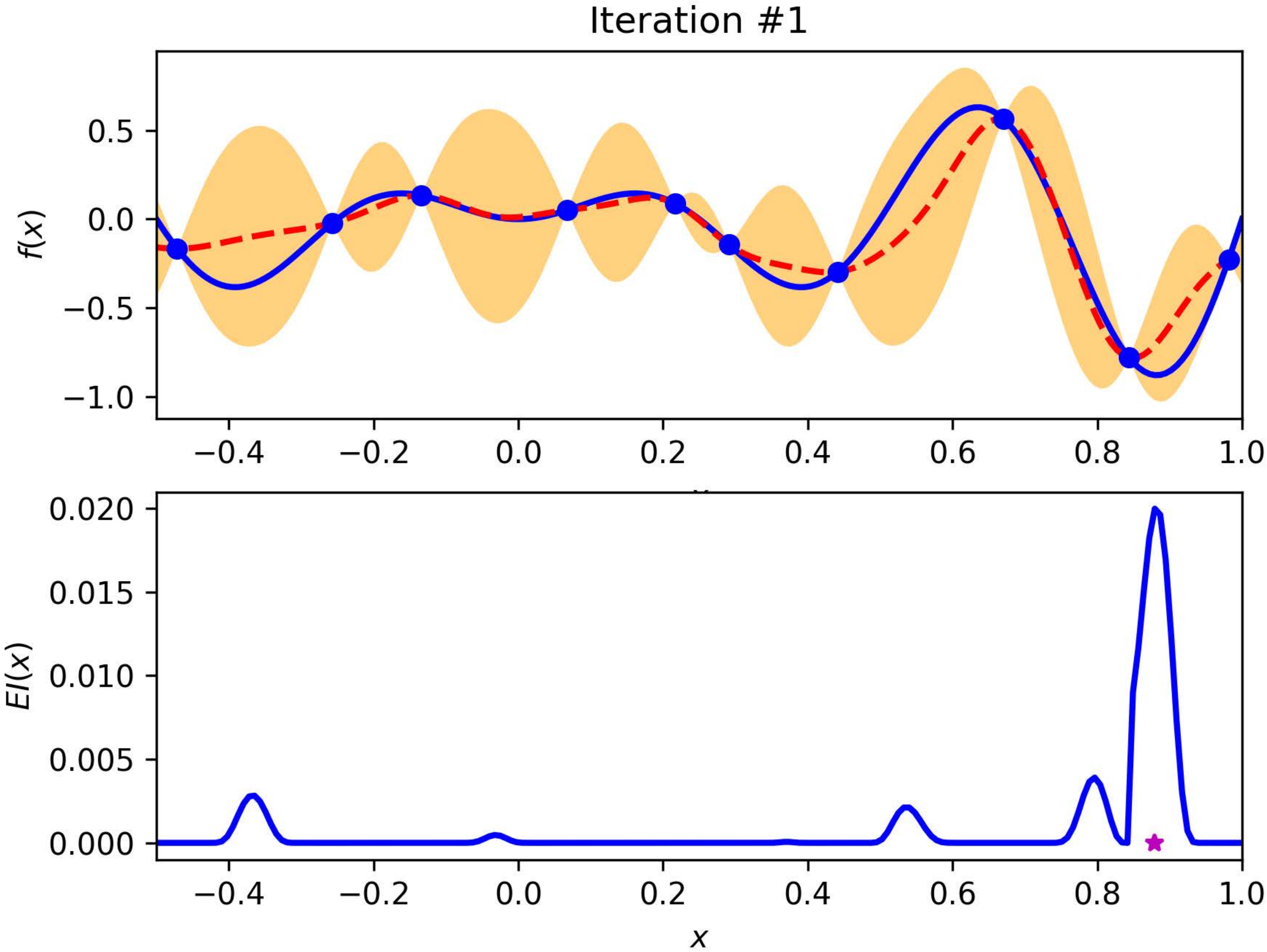
Active learning & Bayesian optimization
Applications

Paris Perdikaris
March 15, 2018

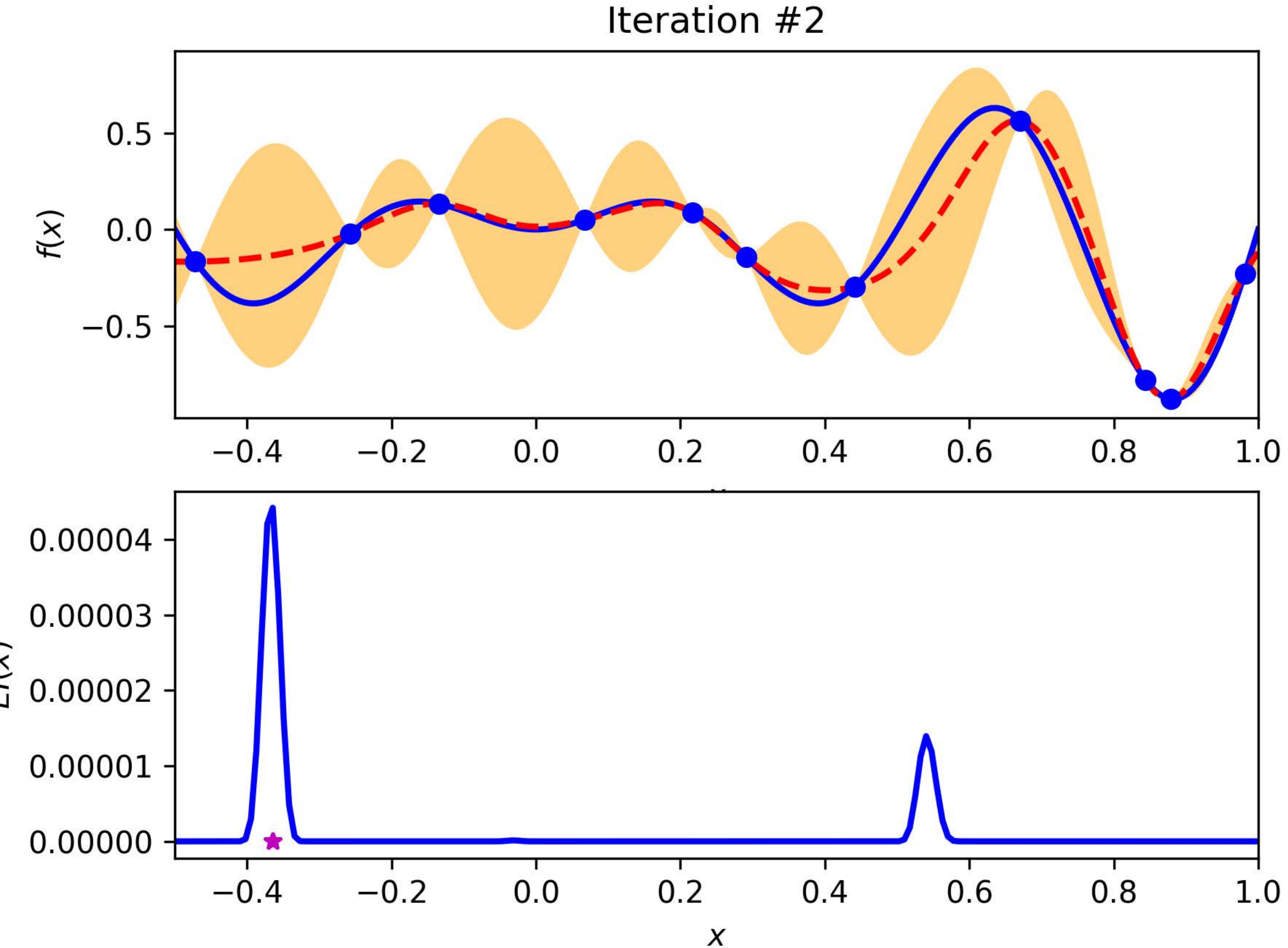
Penn
UNIVERSITY *of* PENNSYLVANIA



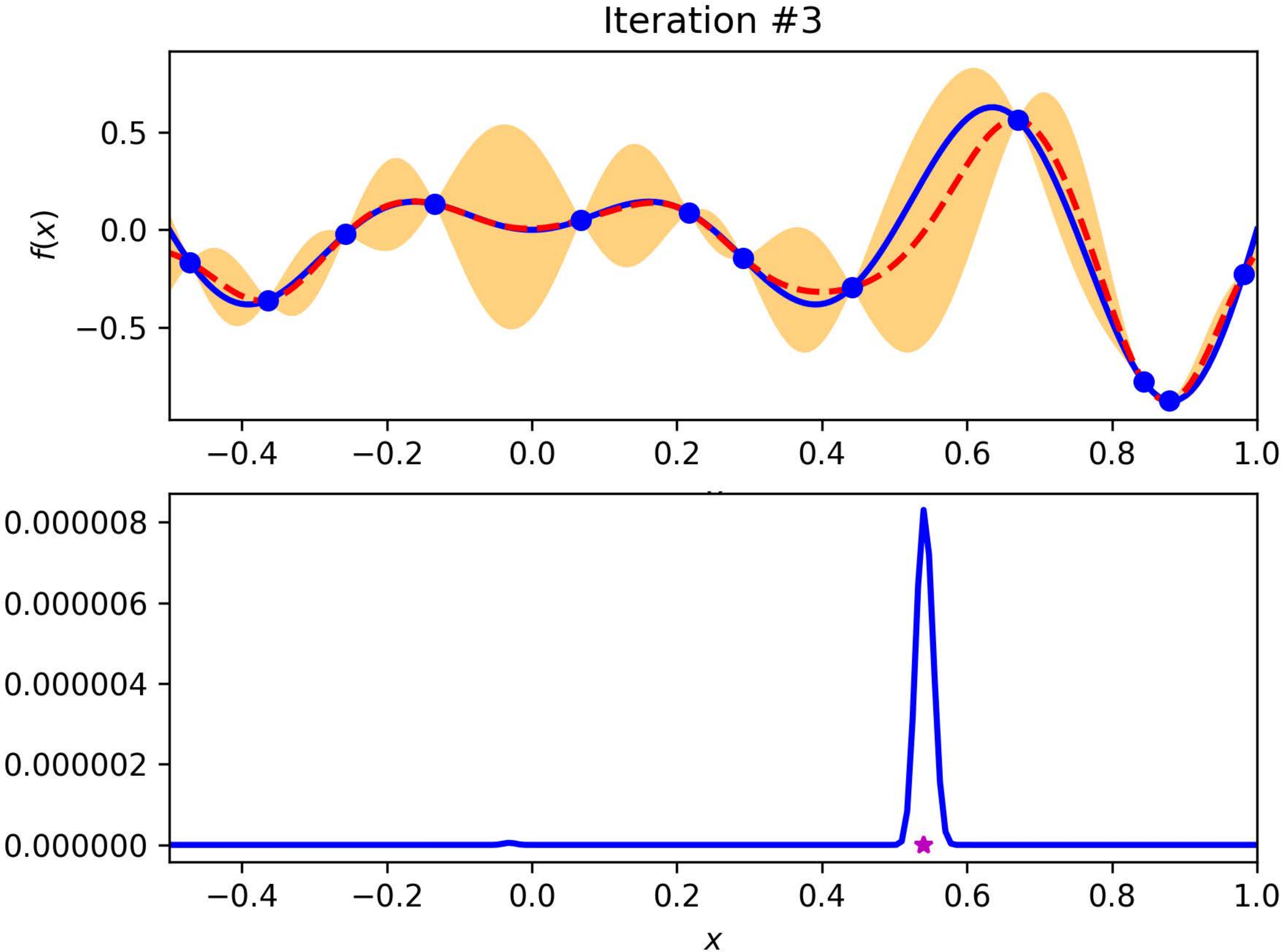
3
2
1
0
−1
−2
−3
−6
−4
−2
0
2
4
6



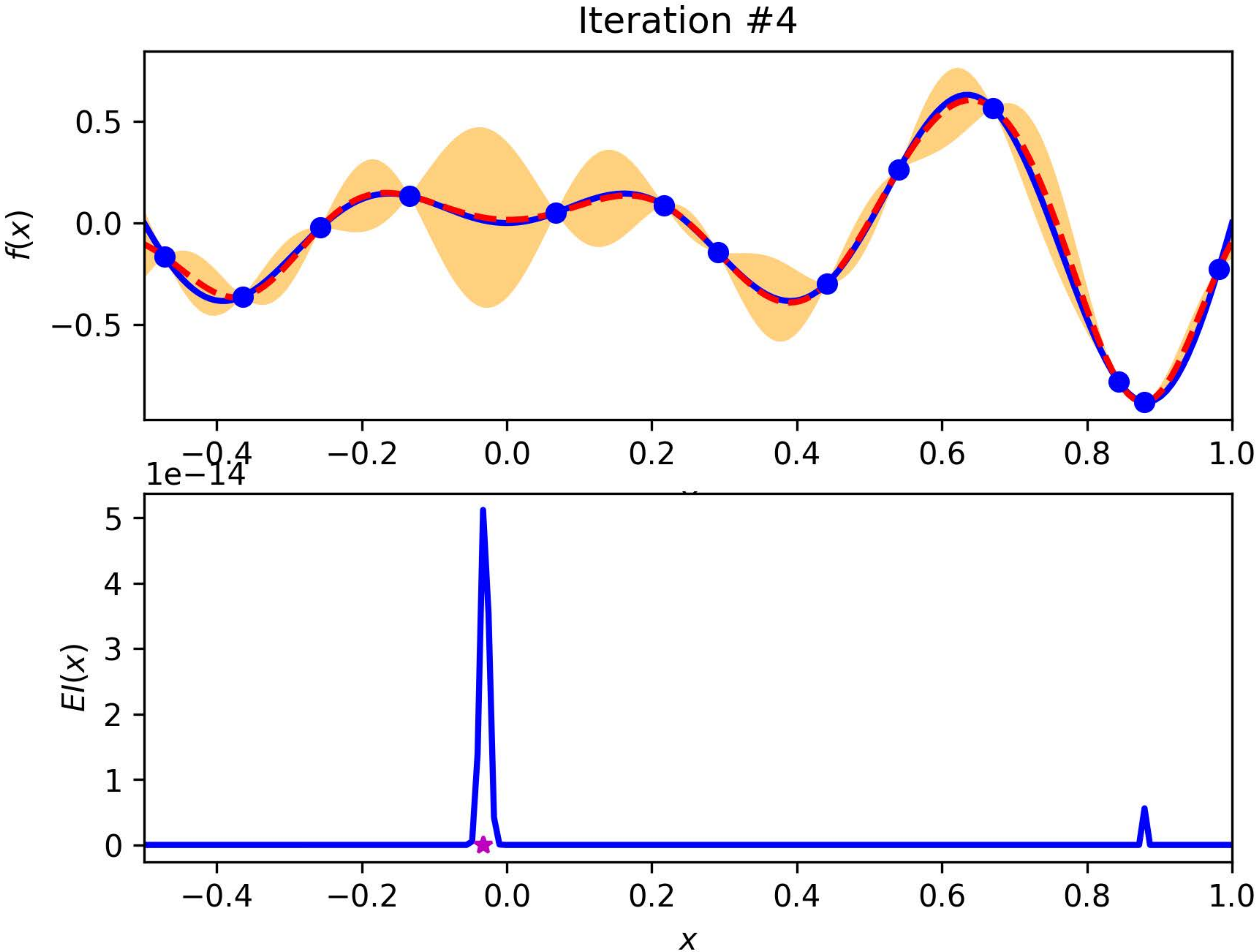
Iteration #1
f(x)
0.5
0.0
−0.5
−1.0
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.020
0.015
0.010
0.005
0.000
x



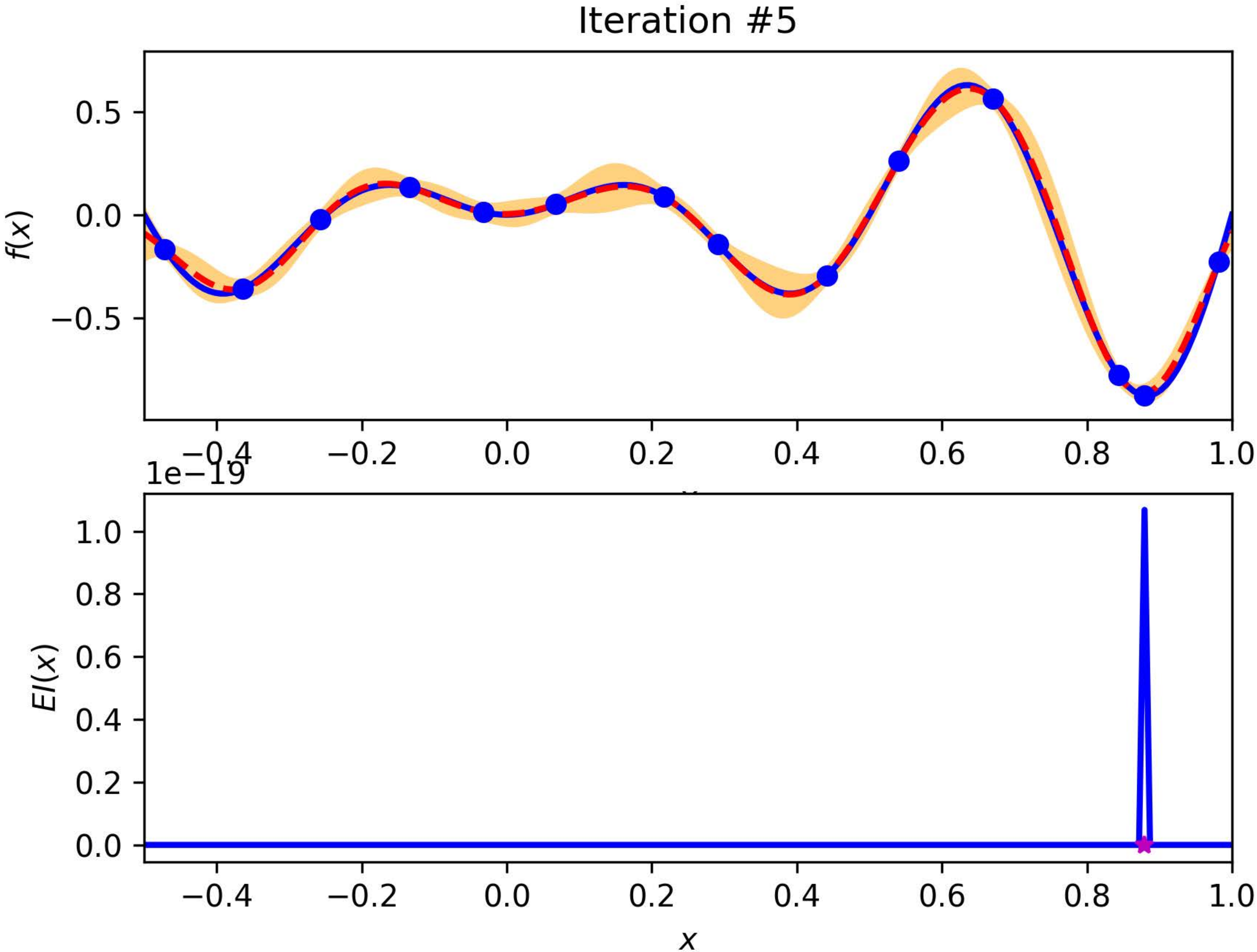
Iteration #2
0.5
0.0
−0.5
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
0.00004
0.00003
0.00002
0.00001
0.00000
x



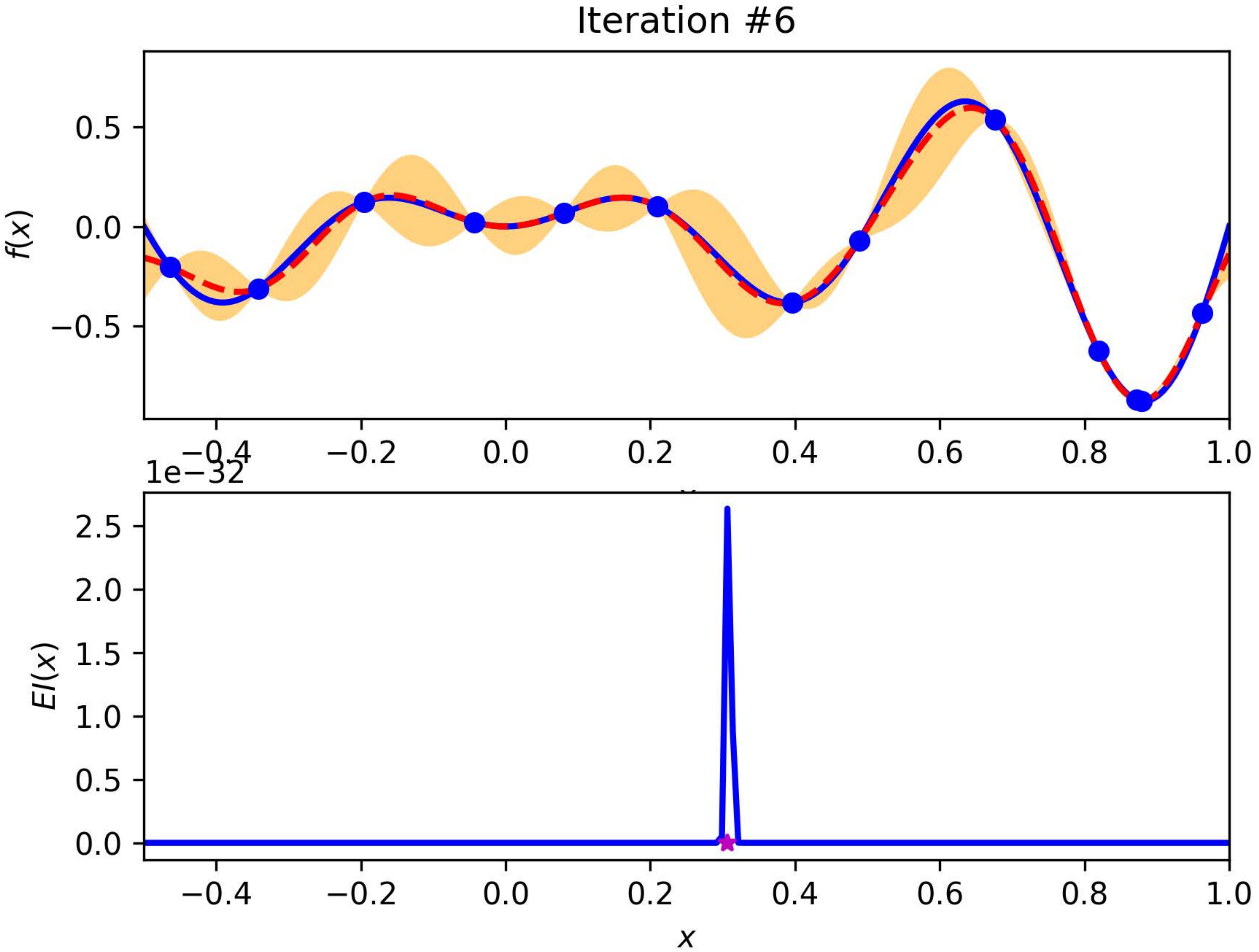
Iteration #3
f(x)
0.5
0.0
−0.5
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
0.000008
0.000006
0.000004
0.000002
0.000000
x



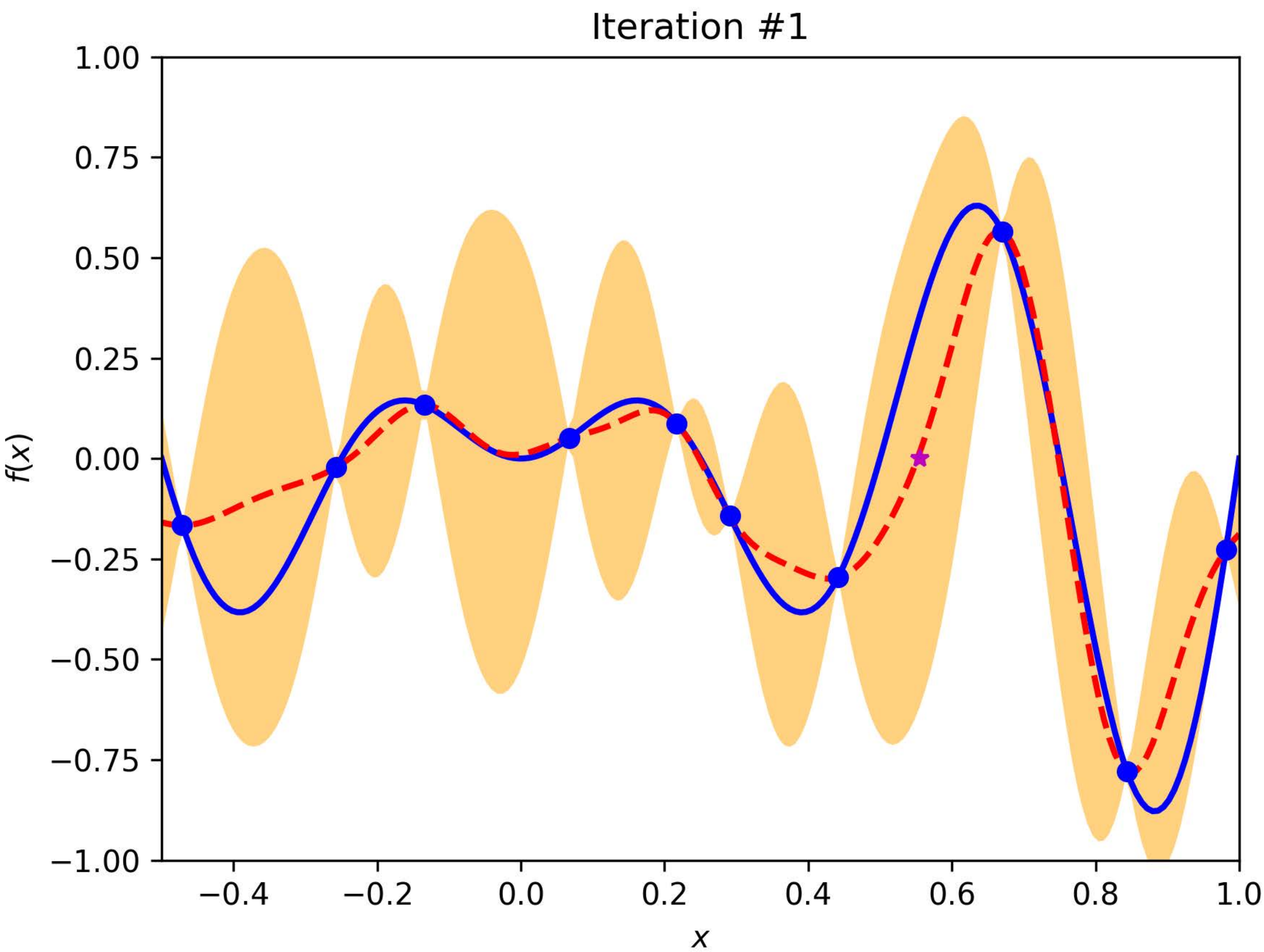
Iteration #4
0.5
0.0
−0.5
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1e−14
5
4
3
2
1
0
EI(x)
x



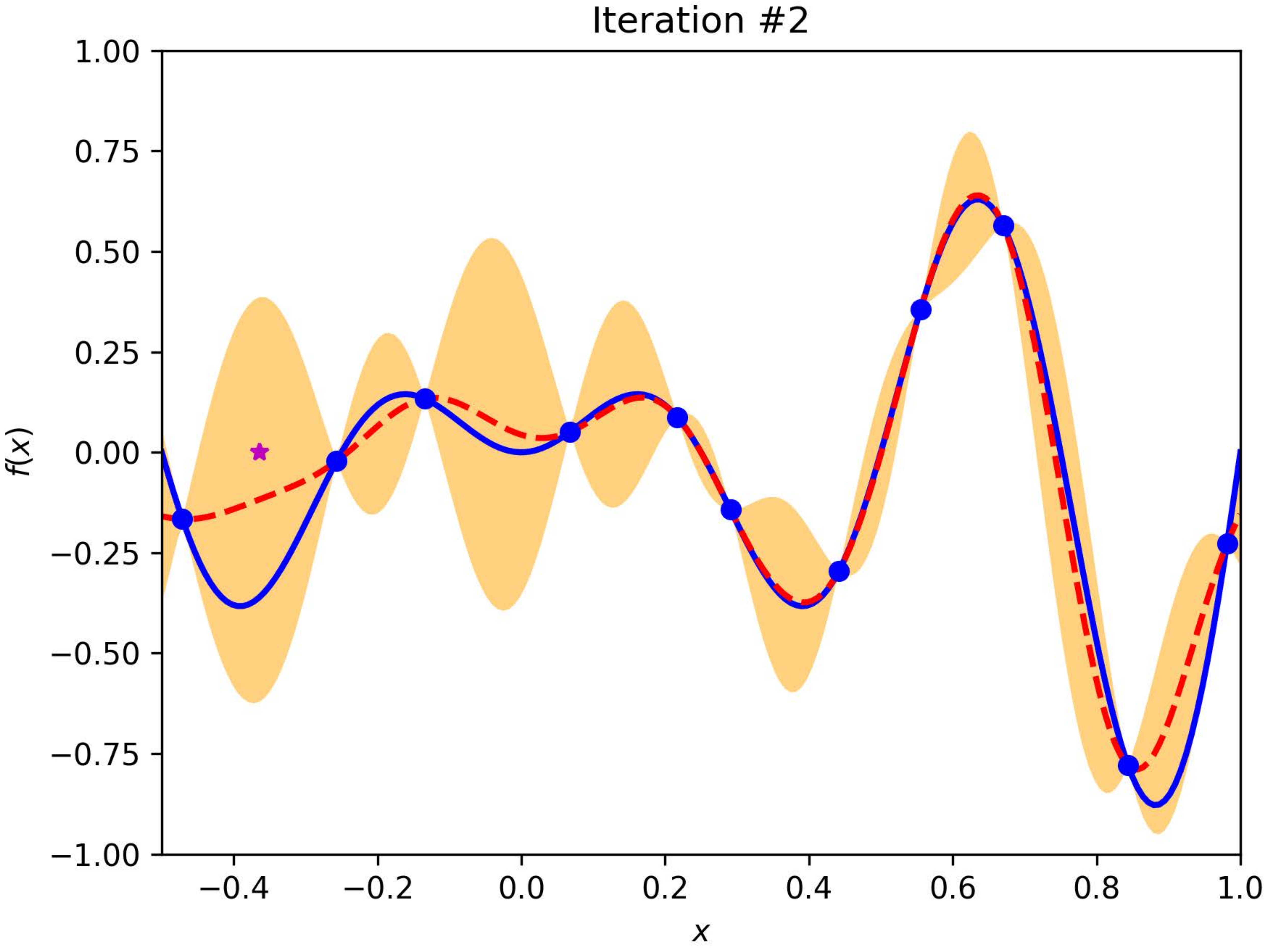
Iteration #5
f(x)
0.5
0.0
−0.5
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1e−19
EI(x)
1.0
0.8
0.6
0.4
0.2
0.0
x



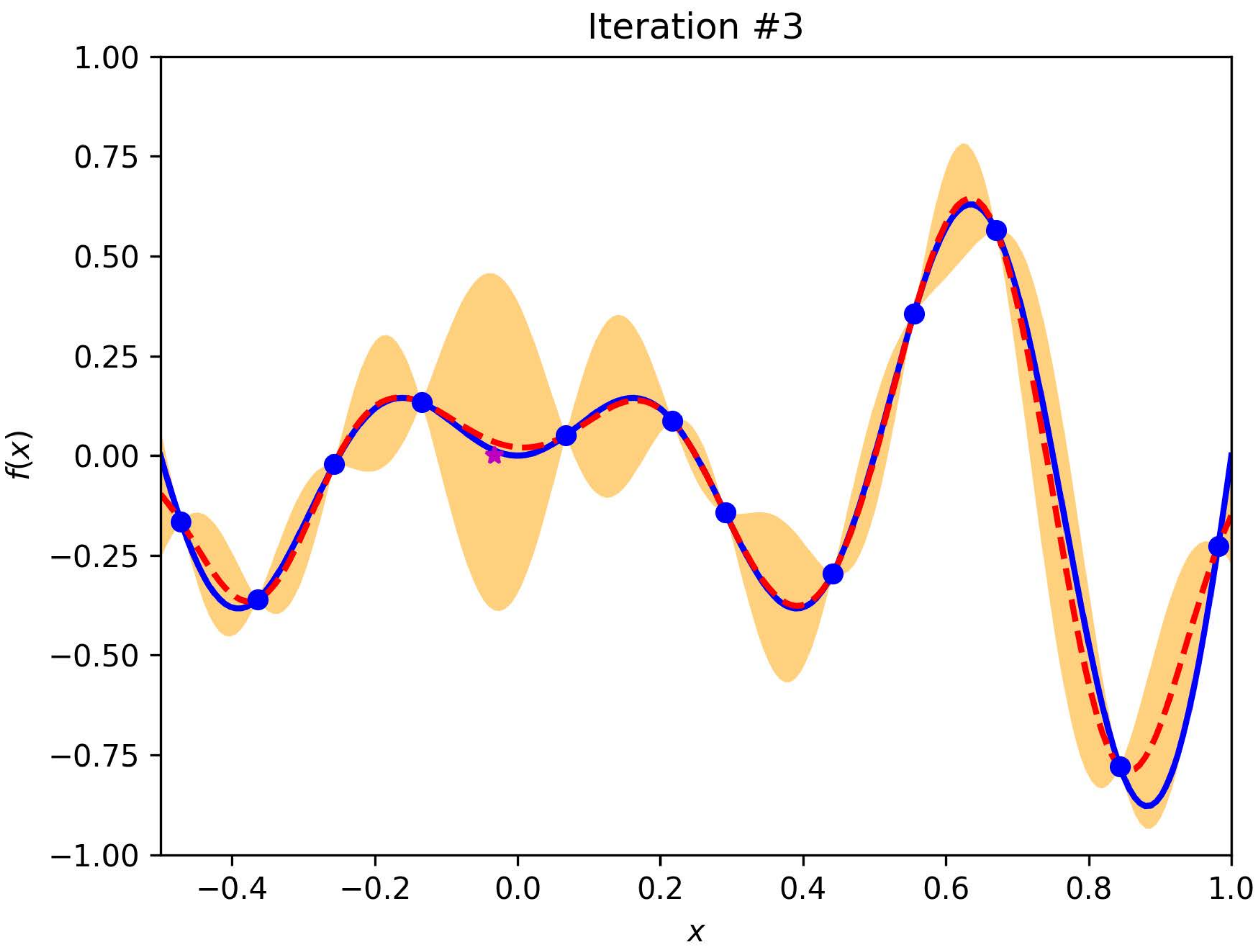
Iteration #6
f(x)
0.5
0.0
−0.5
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x
1e−32
2.5
2.0
1.5
1.0
0.5
0.0
EI(x)
x



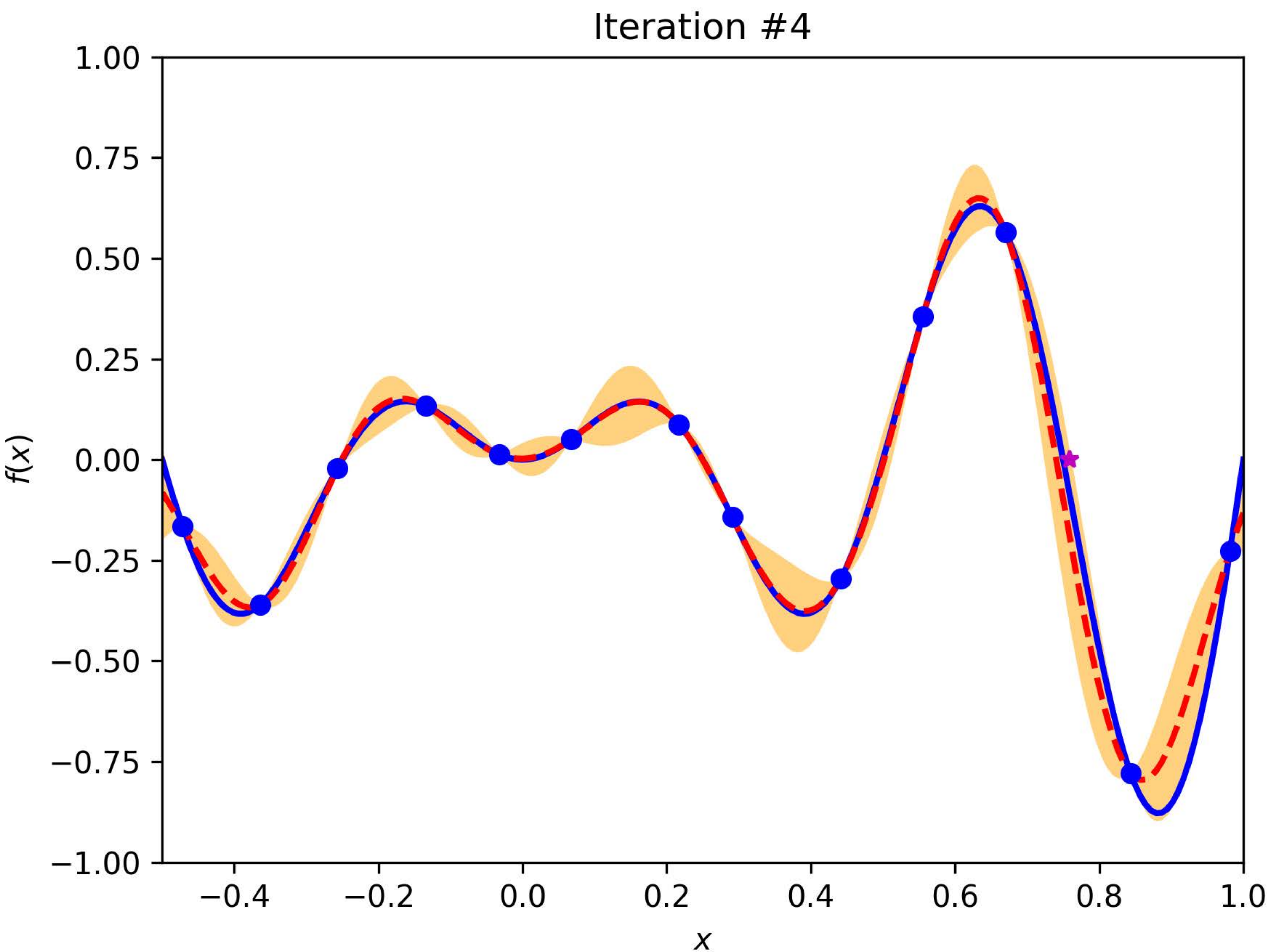
Iteration #1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x



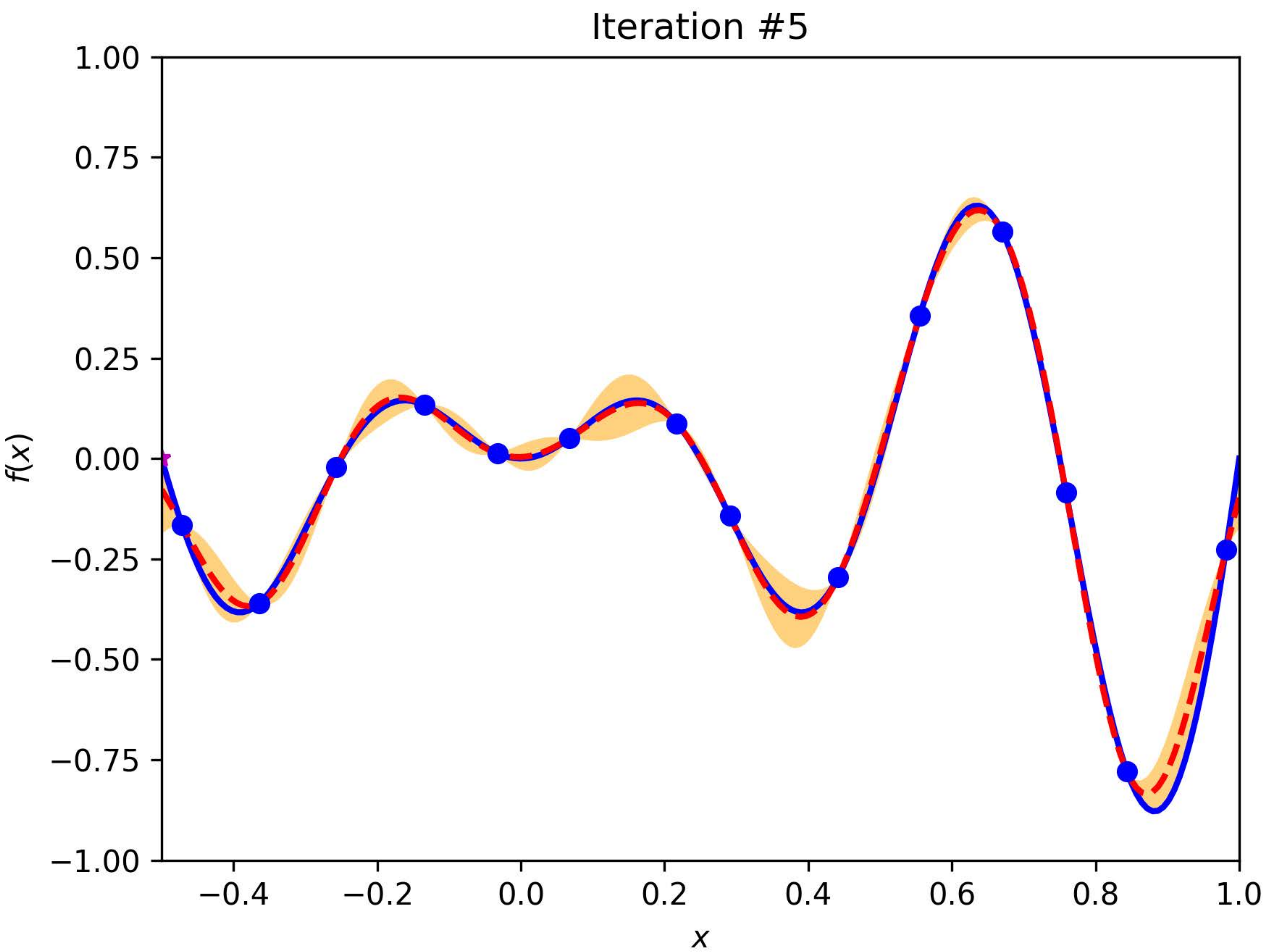
Iteration #2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x



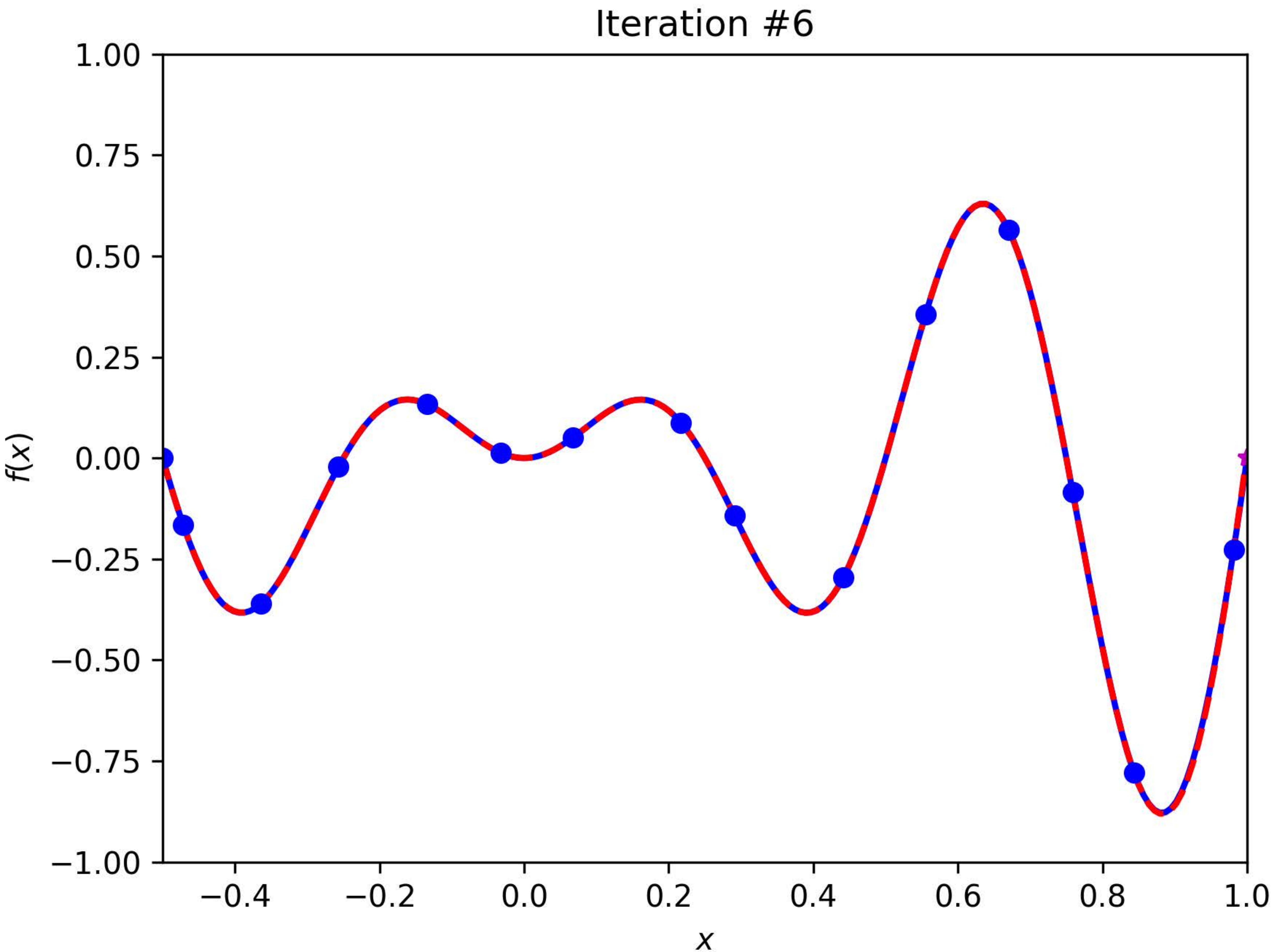
Iteration #3
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x



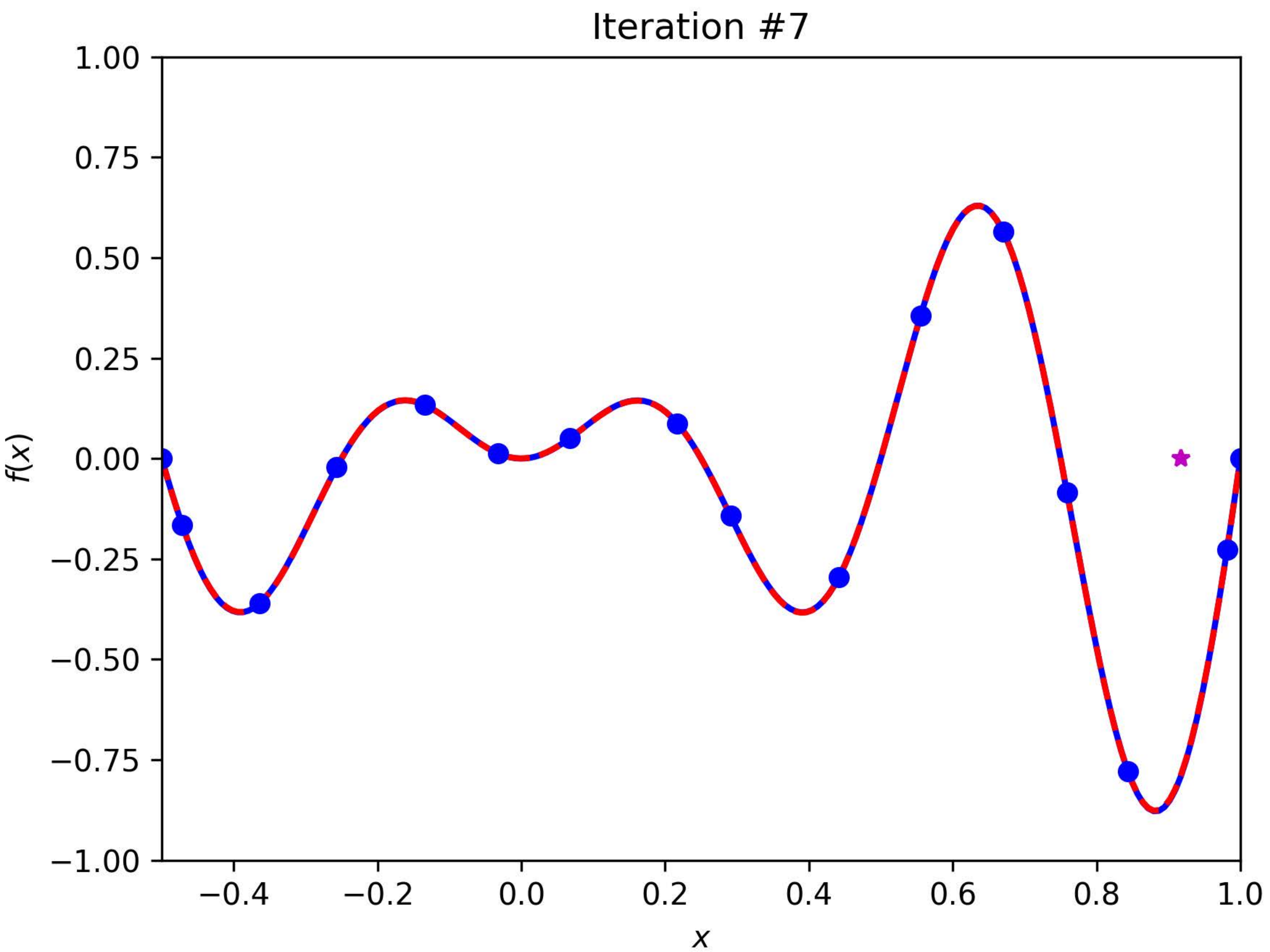
Iteration #4
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x



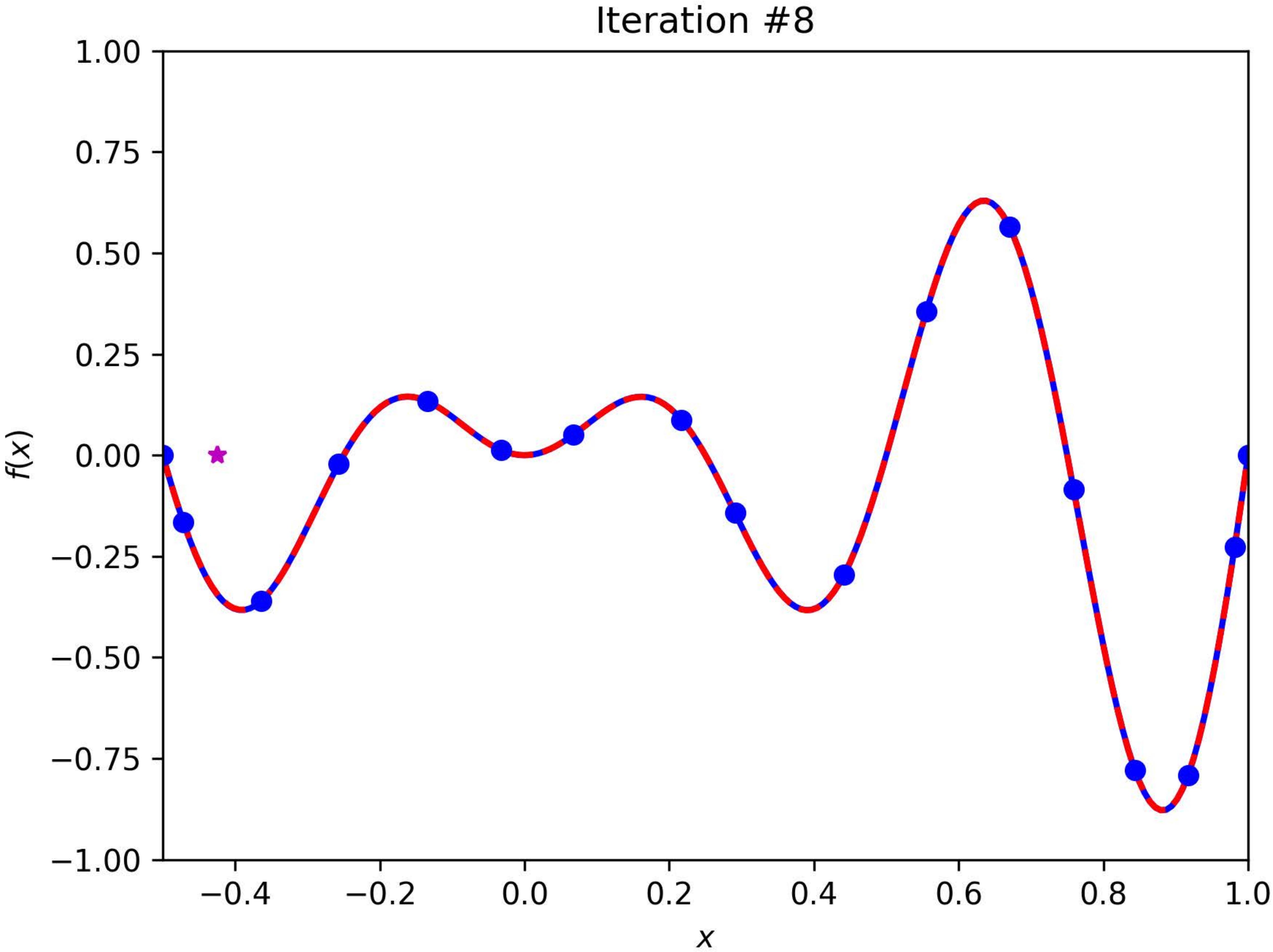
Iteration #5
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x



Iteration #6
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x



Iteration #7
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x



Iteration #8
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(x)
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x

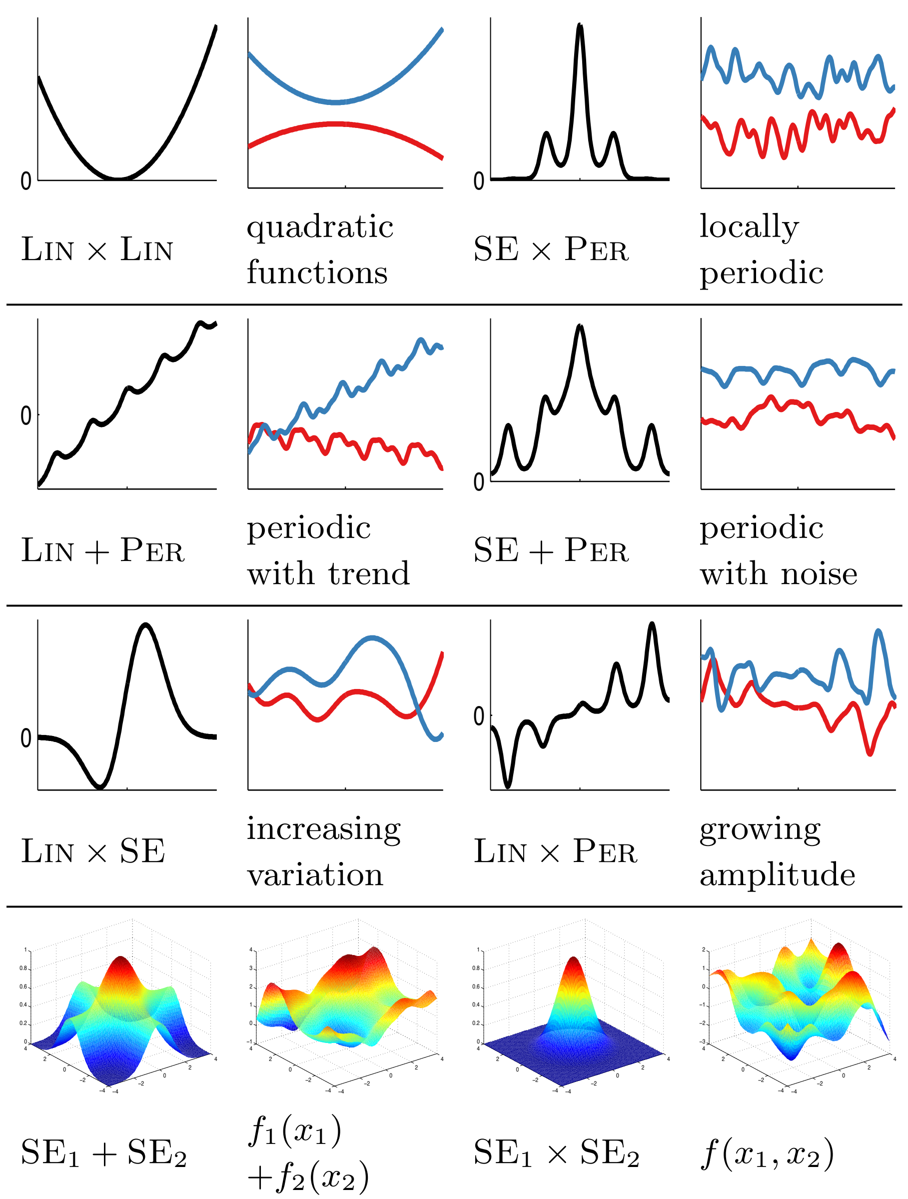
Applications to model selection

The Kernel Cookbook:

Advice on Covariance functions

by **David Duvenaud**

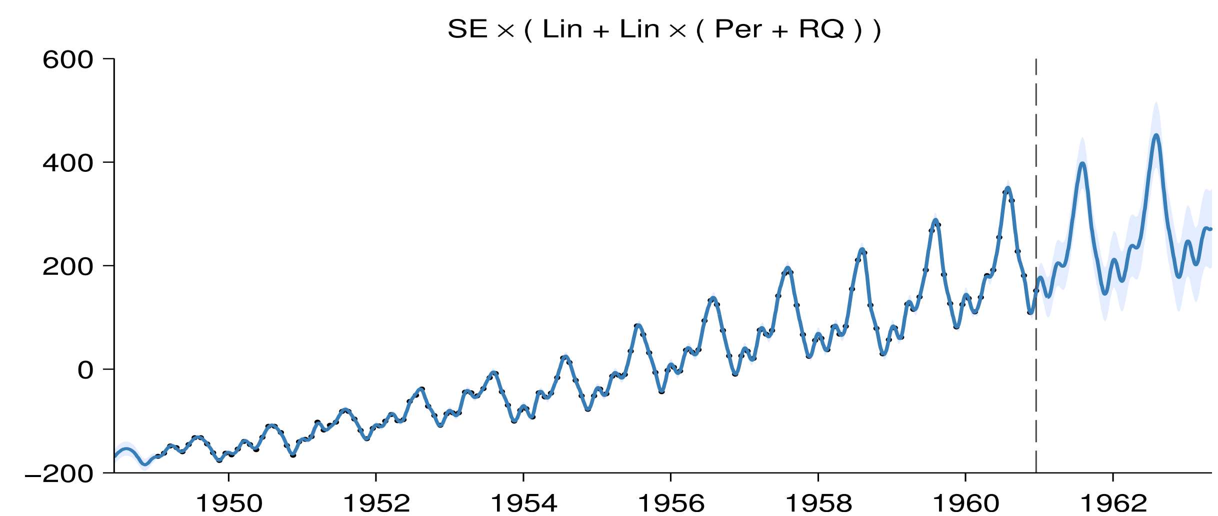
https://www.cs.toronto.edu/~duvenaud/cookbook/



The Automatic Statistician

An artificial intelligence for data science

https://automaticstatistician.com/index/



$$K_g(\mathcal{M}, \mathcal{M}'; \theta_g, \mathbf{X}) = \sigma^2 \exp\left(-\frac{1}{2}\frac{\bar{d}_{\mathrm{H}}^2(\mathcal{M}, \mathcal{M}'; \mathbf{X})}{\ell^2}\right)$$

Malkomes, G., Schaff, C., & Garnett, R. (2016). Bayesian optimization for automated model selection. In Advances in Neural Information Processing Systems (pp. 2900-2908).

Practical Bayesian Optimization of Machine Learning Algorithms

Jasper Snoek
Department of Computer Science
University of Toronto
jasper@cs.toronto.edu

Hugo Larochelle
Department of Computer Science
University of Sherbrooke
hugo.larochelle@usherbrooke.edu

Ryan P. Adams
School of Engineering and Applied Sciences
Harvard University
rpa@seas.harvard.edu

Applications to control

PILCO: A Model-Based and Data-Efficient Approach to Policy Search

Marc Peter Deisenroth MARC@CS.WASHINGTON.EDU
Department of Computer Science & Engineering, University of Washington, USA

Carl Edward Rasmussen CER54@CAM.AC.UK
Department of Engineering, University of Cambridge, UK

Gaussian Processes for Data-Efficient Learning in Robotics and Control

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen

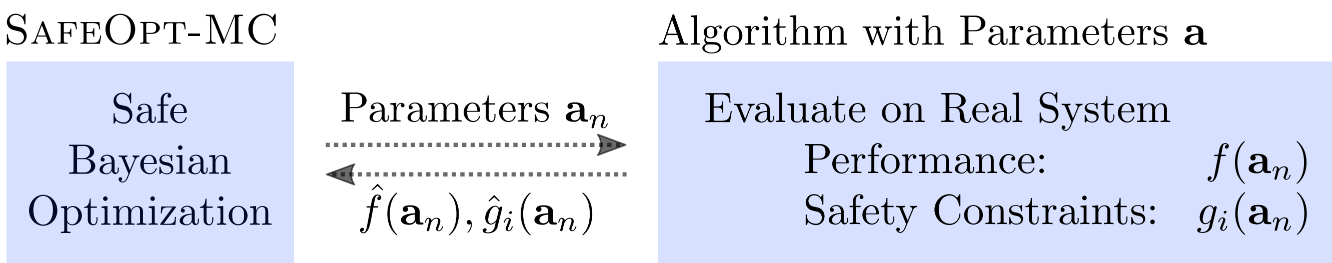
Applications to control

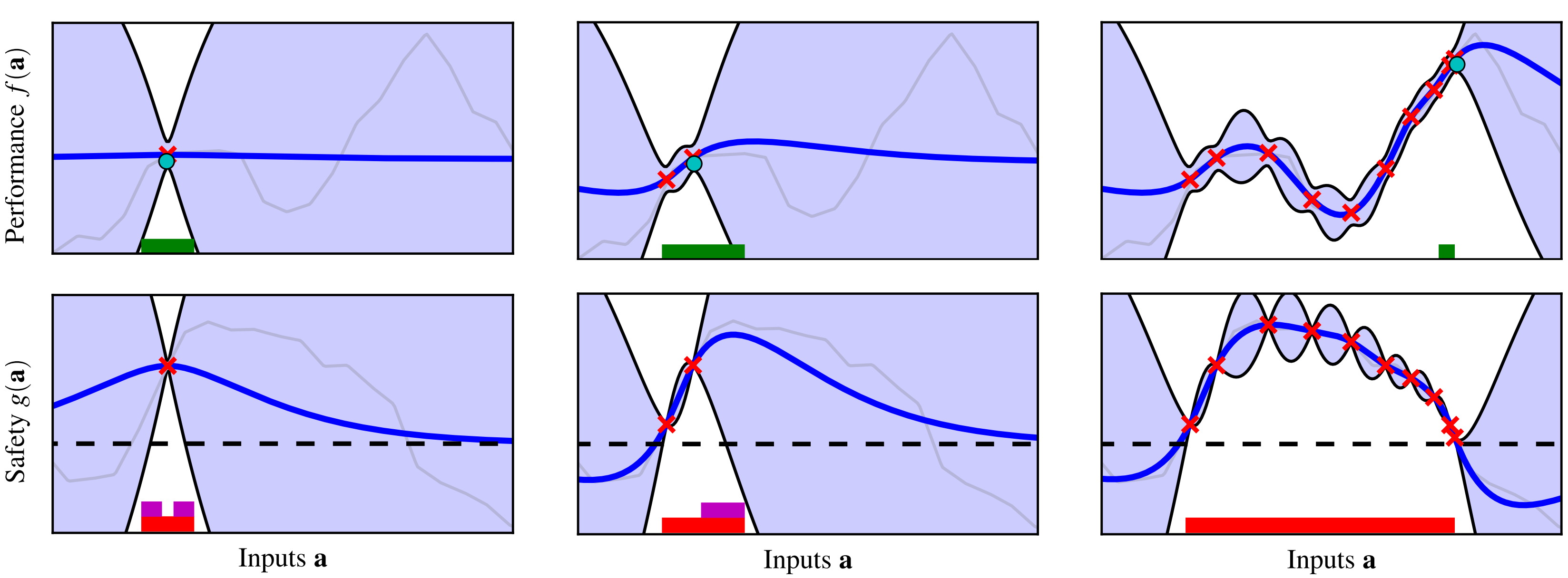
Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics

Felix Berkenkamp*, Andreas Krause*, and Angela P. Schoellig†
*Learning & Adaptive Systems Group, Department of Computer Science, ETH Zurich, Switzerland
†Dynamic Systems Lab, Institute for Aerospace Studies, University of Toronto, Canada
Email: *{befelix, krausea}@ethz.ch, †schoellig@utias.utoronto.ca

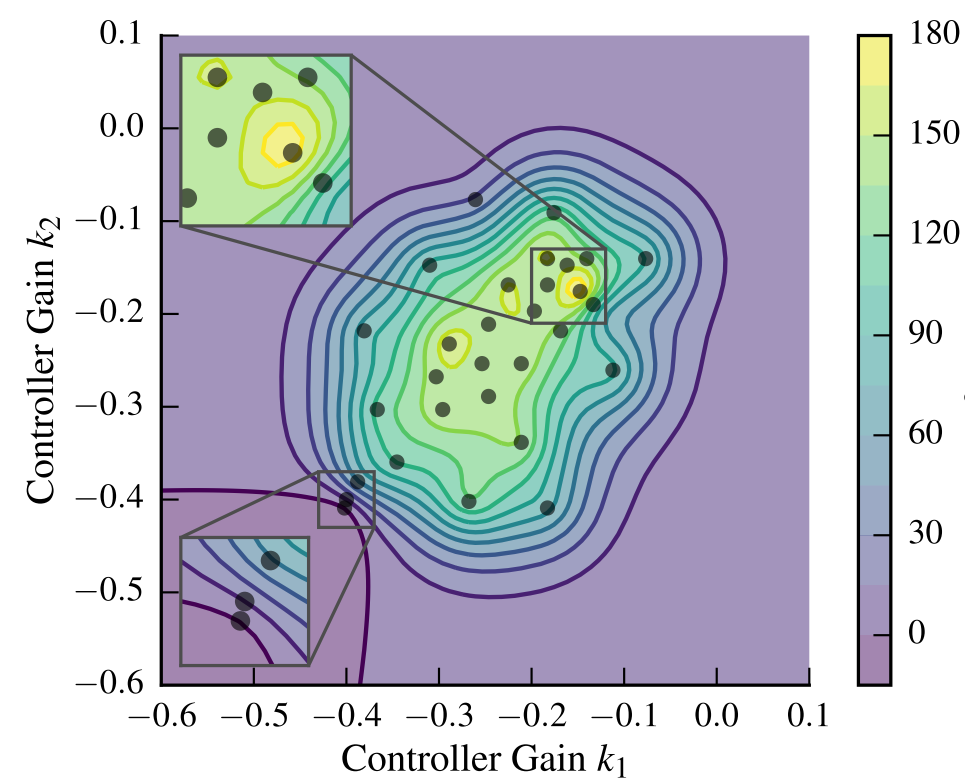
Safe Controller Optimization for Quadrotors with Gaussian Processes

Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause





(d) Initial, safe parameters.
(e) Safe exploration.
(f) After 10 evaluations: safe maximum found.



Example application: *Calibration of blood flow simulations*

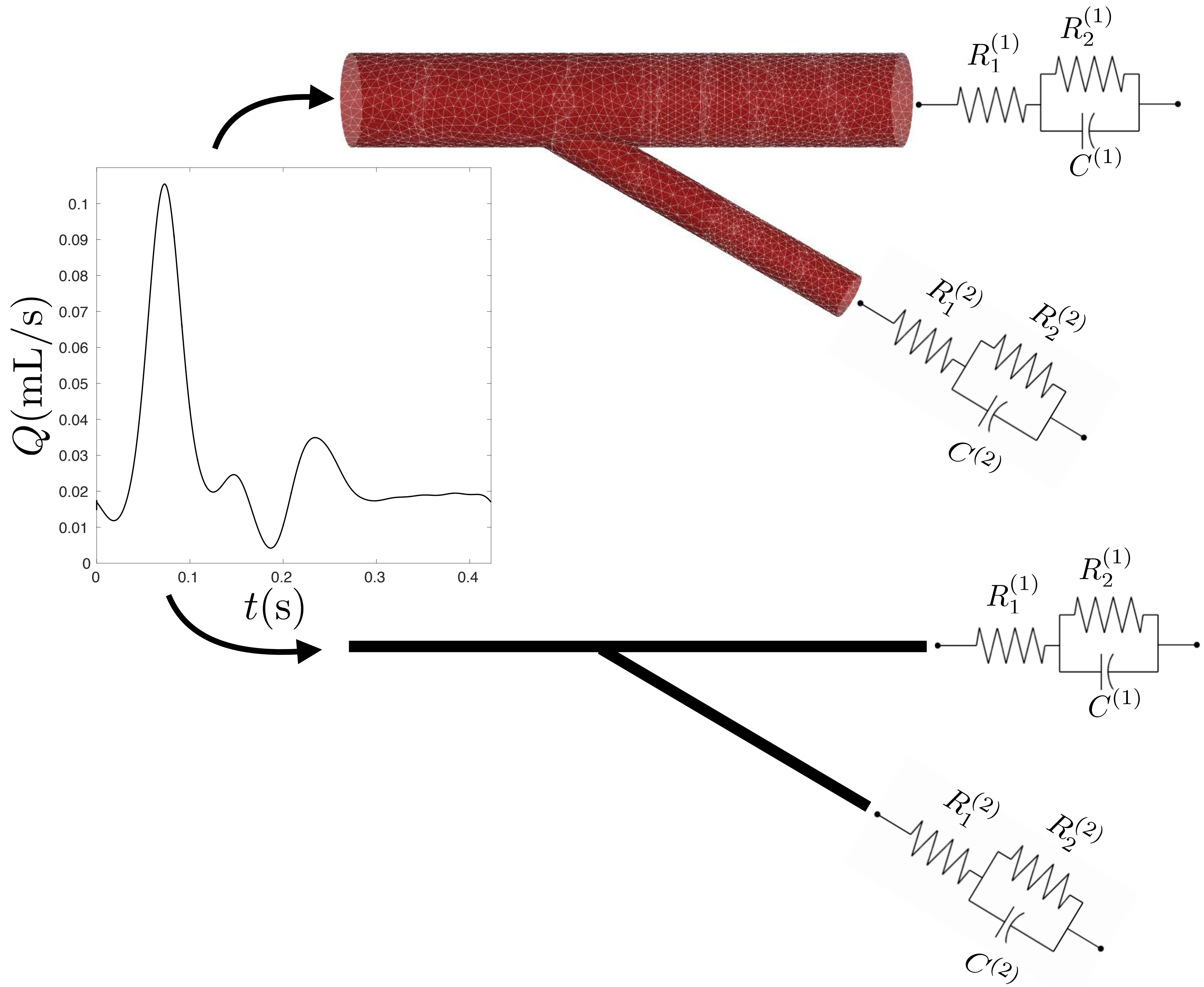
Goal:
Calibrate the outflow boundary condition parameters to match a target inlet systolic pressure, i.e.,

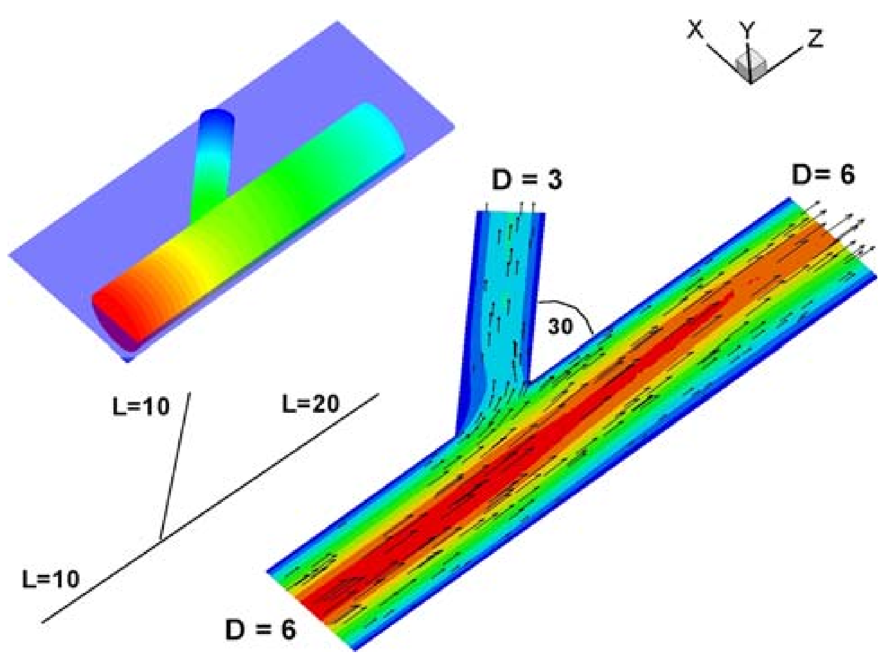
$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} |p_s^\star - p_s(\mathbf{x})|^2,$$

$$\mathbf{x} = [R_T^{(1)}, R_T^{(2)}]$$

$$\mathcal{X} = [10^{10}, 10^{11}] \times [10^{11}, 10^{12}]$$

$$p_s^\star = 47\text{mmHg}$$

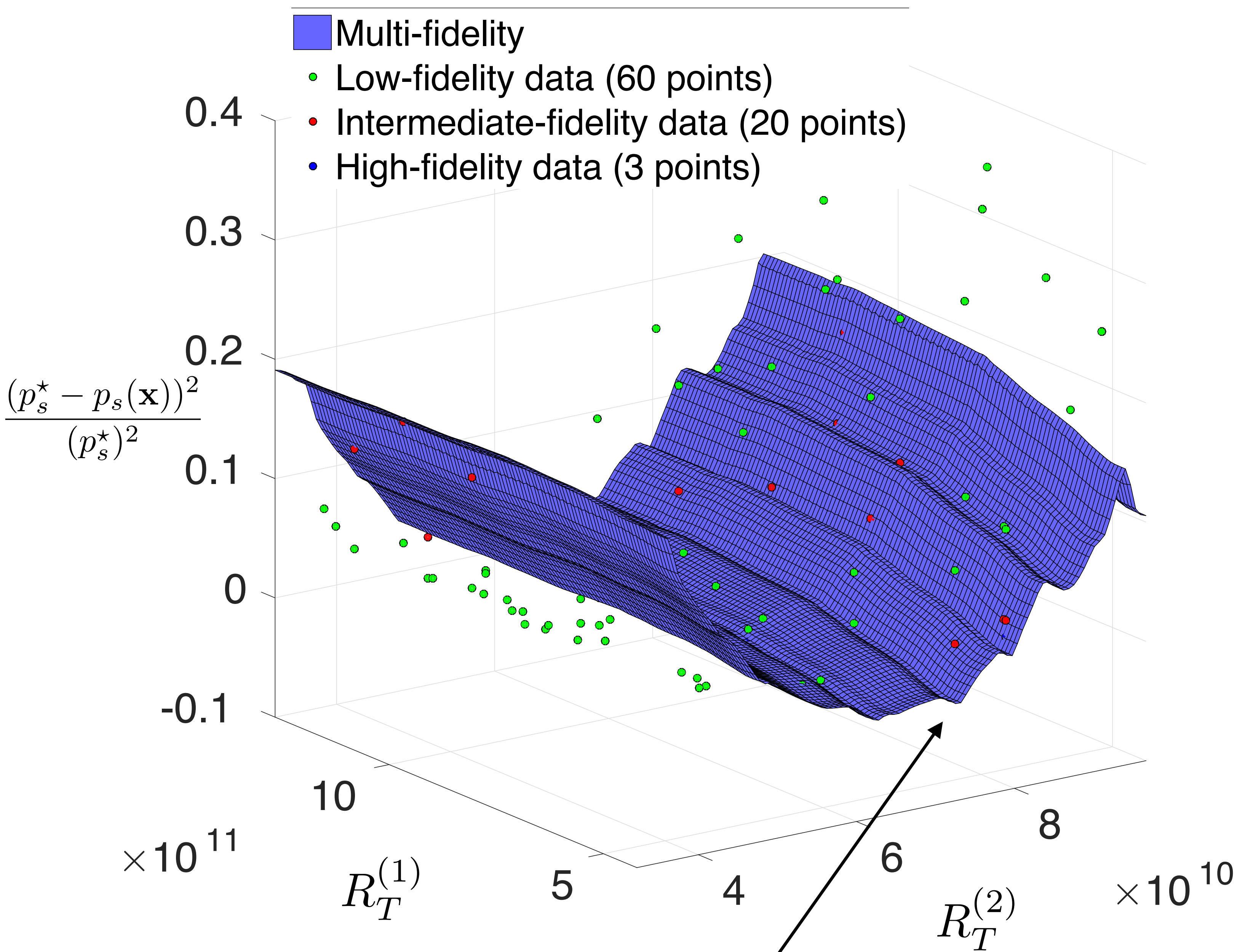


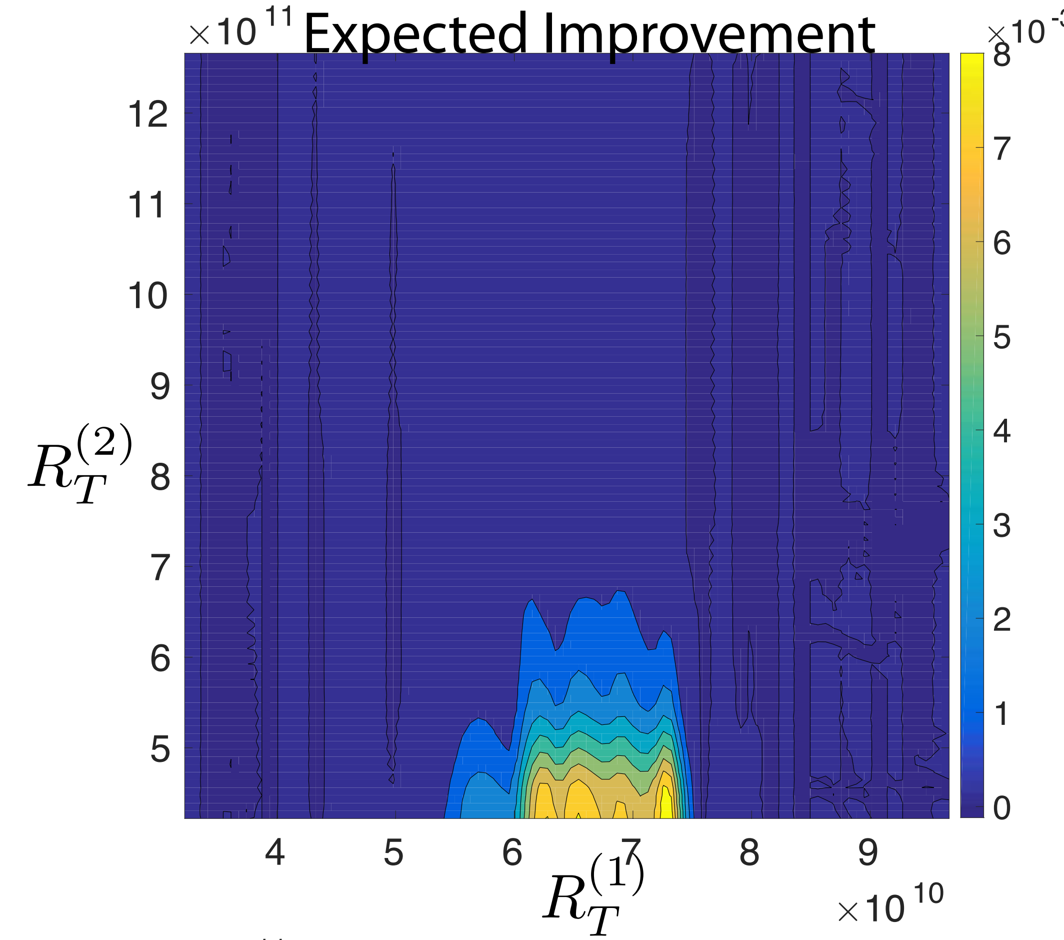


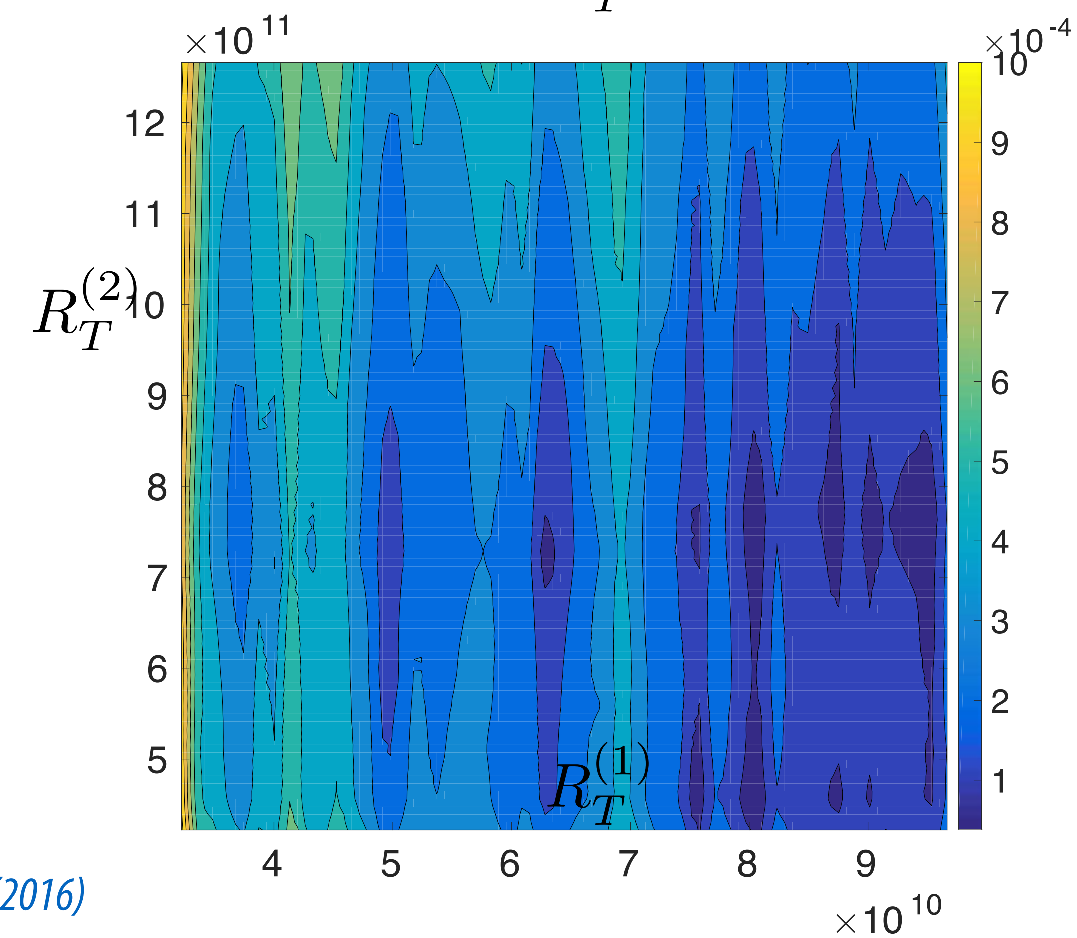
Multi-fidelity approach:
1.) 3D Navier-Stokes (spectral/hp elements, rigid artery) - high fidelity O(hrs)
2.) Non-linear 1D-FSI (DG, compliant artery) - intermediate fidelity O(mins)
3.) Linearized 1D-FSI solver around an inaccurate reference state - low fidelity O(s)

P. Perdikaris, and G.E Karniadakis. "Model inversion via multi-fidelity Bayesian optimization." J. R. Soc. Interface (2016)

Example application: *Calibration of blood flow simulations*







Decreased the relative error to $\mathcal{O}(10^{-3})$ after 3 iterations of BO, mainly sampling the lowest fidelity (cheapest) solver.

P. Perdikaris, and G.E Karniadakis. "Model inversion via multi-fidelity Bayesian optimization." J. R. Soc. Interface (2016)

Things we didn't cover

- Non-Gaussian likelihoods/noise models
- Approximate inference techniques (Variational inference, MCMC)
- Classification

- Open source software:
  - GPy
  - GPflow
  - GPML
  - GPyOpt
  - GPflowOpt

See the course website for links:

`https://www.seas.upenn.edu/~enm540/syllabus/`