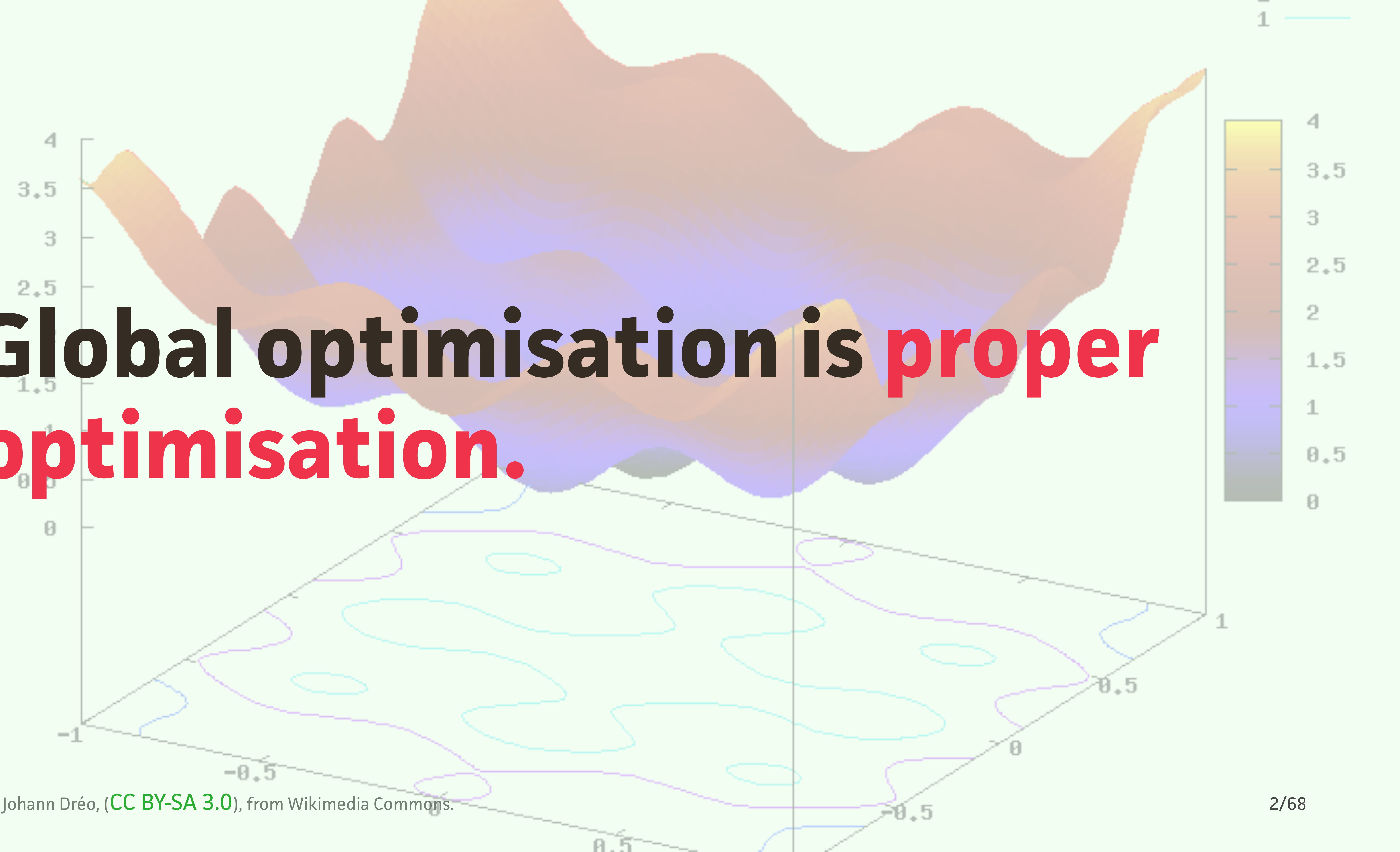
BAYESIAN OPTIMISATION

IS PROBABILISTIC NUMERICS

Michael A Osborne, @maosbot



Global optimisation is **proper optimisation.**

By Johann Dréo, (CC BY-SA 3.0), from Wikimedia Commons.

Exploitation

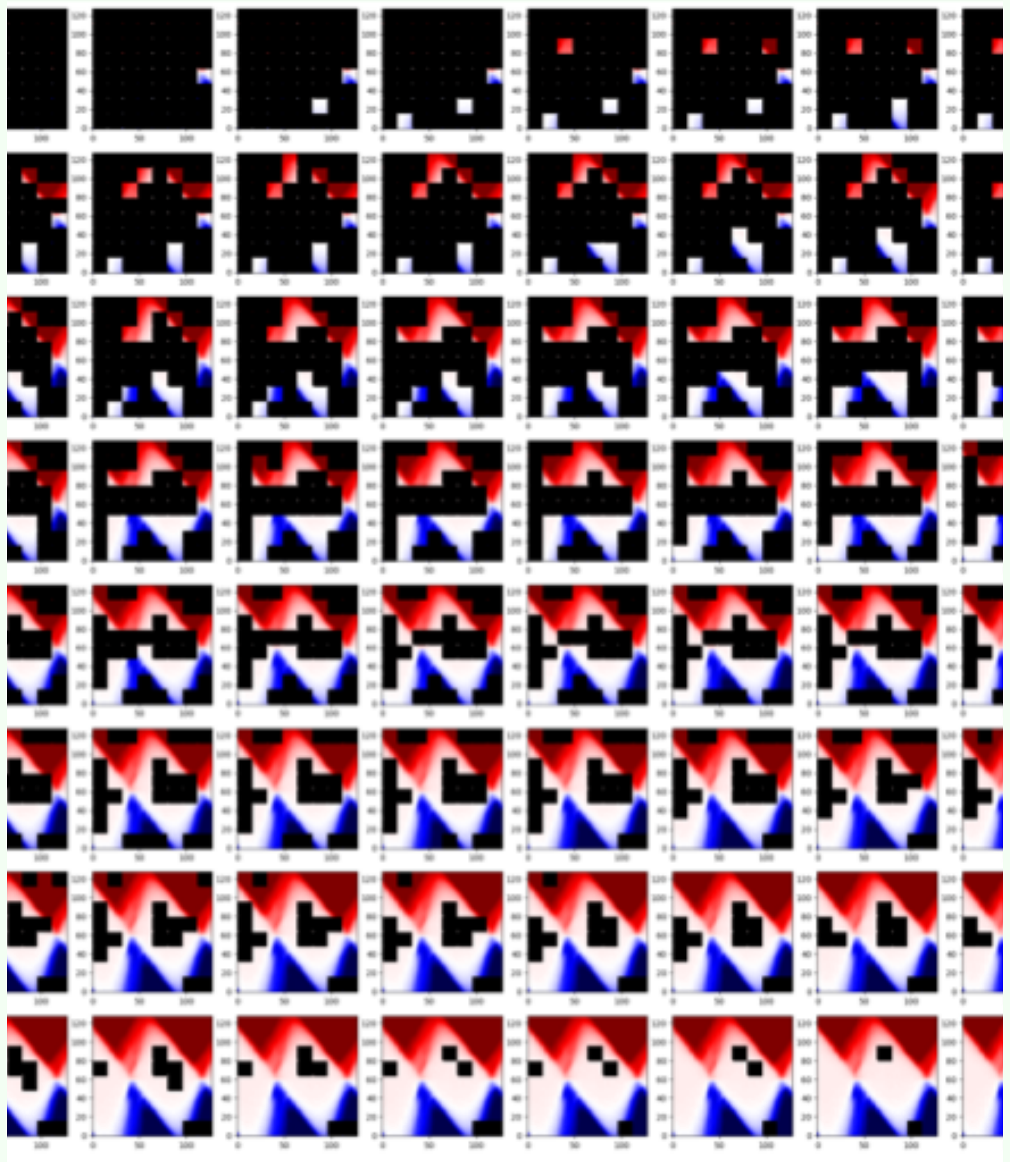
By Tony Hisgett from Birmingham, UK (Oil Pump, uploaded by tm) [CC BY 2.0], via Wikimedia Commons.

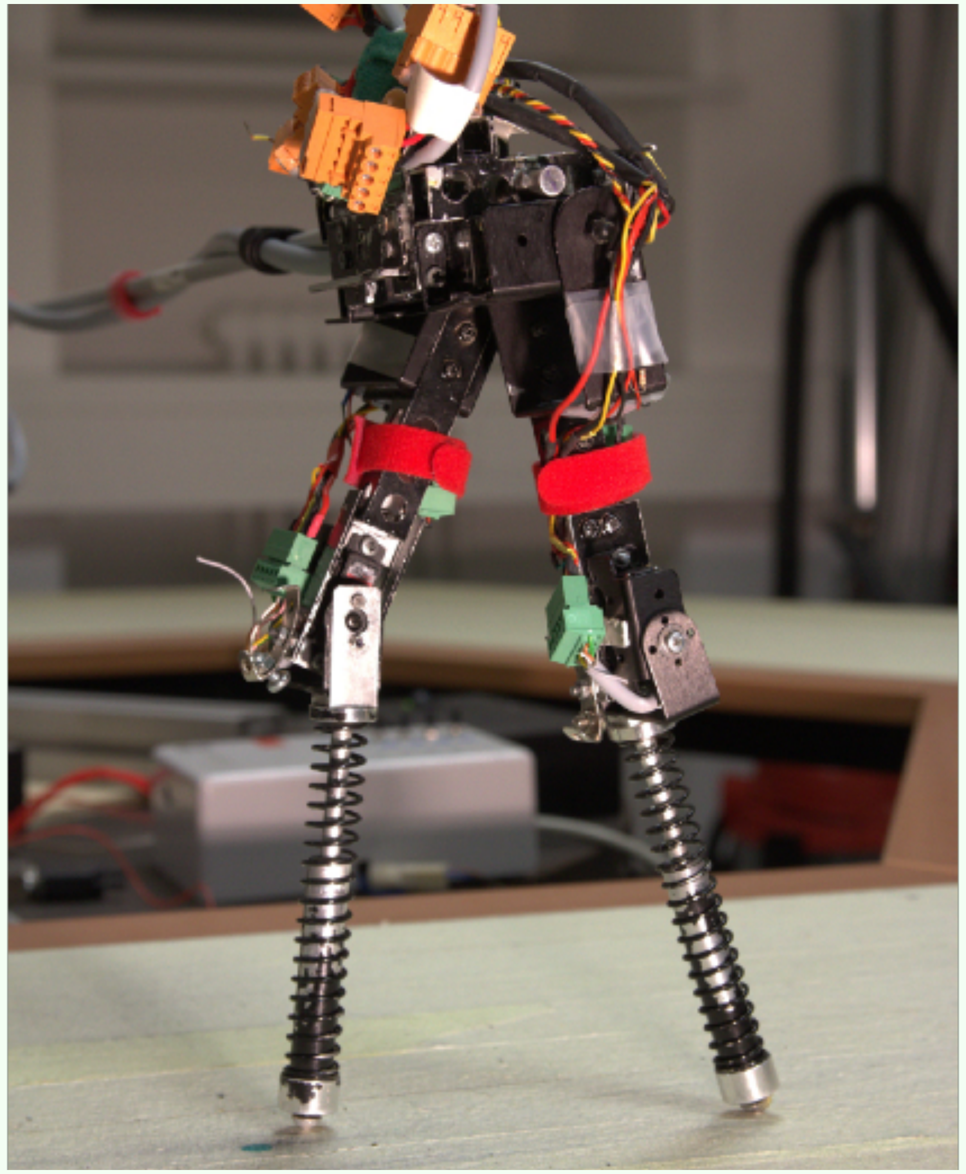
Exploration

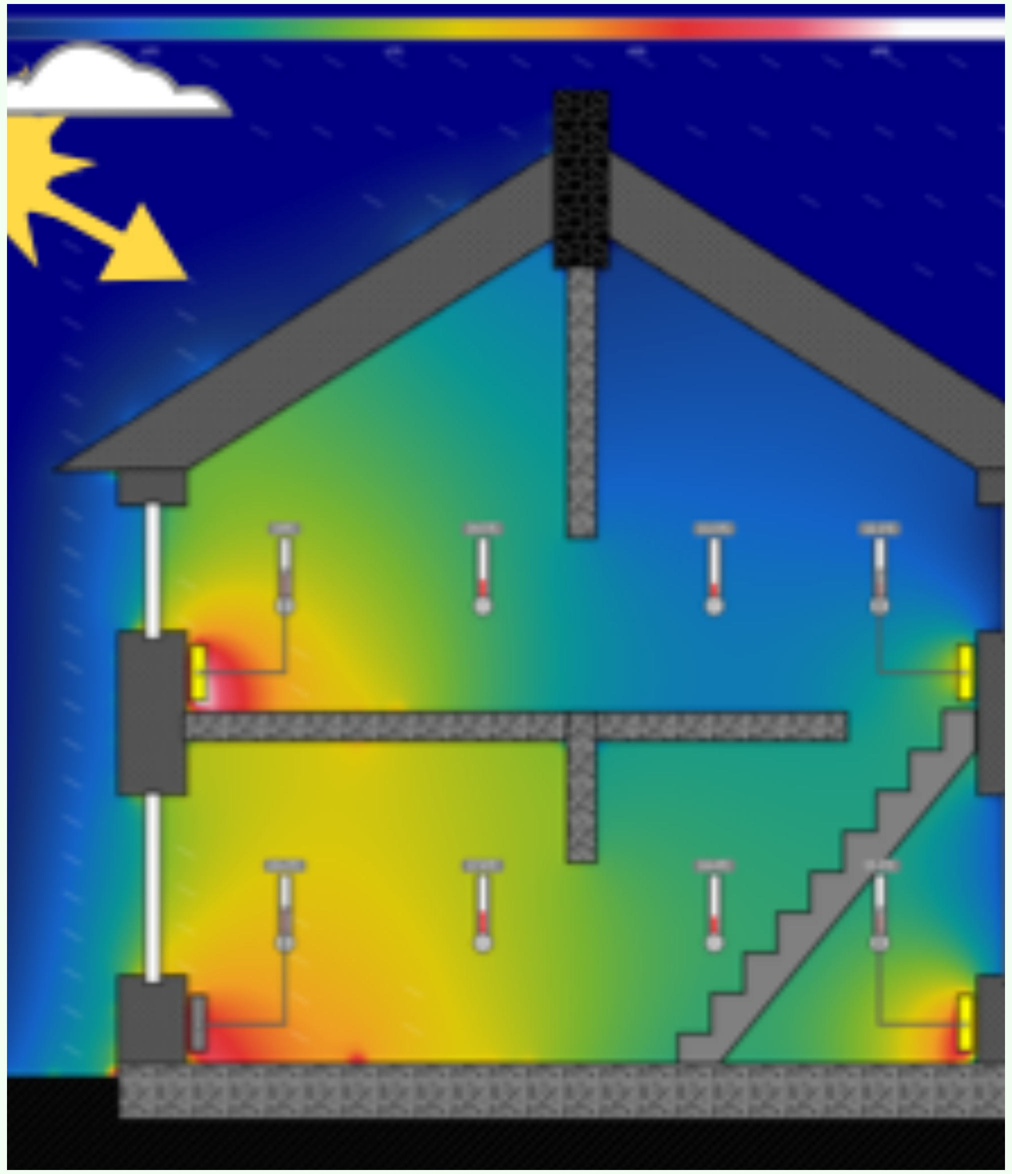
Relative to local optimisation, global optimisation:

1. is less amenable to **theory**;
2. requires higher **overhead**; and
3. overhead costs scale more poorly in **dimension**.

Global optimisation is widely used.







Source: Rainforth, Le, van de Meent, Osborne, & Wood (2016); Calandra, Gopalan, Seyfarth, Peters, Deisenroth (2014).

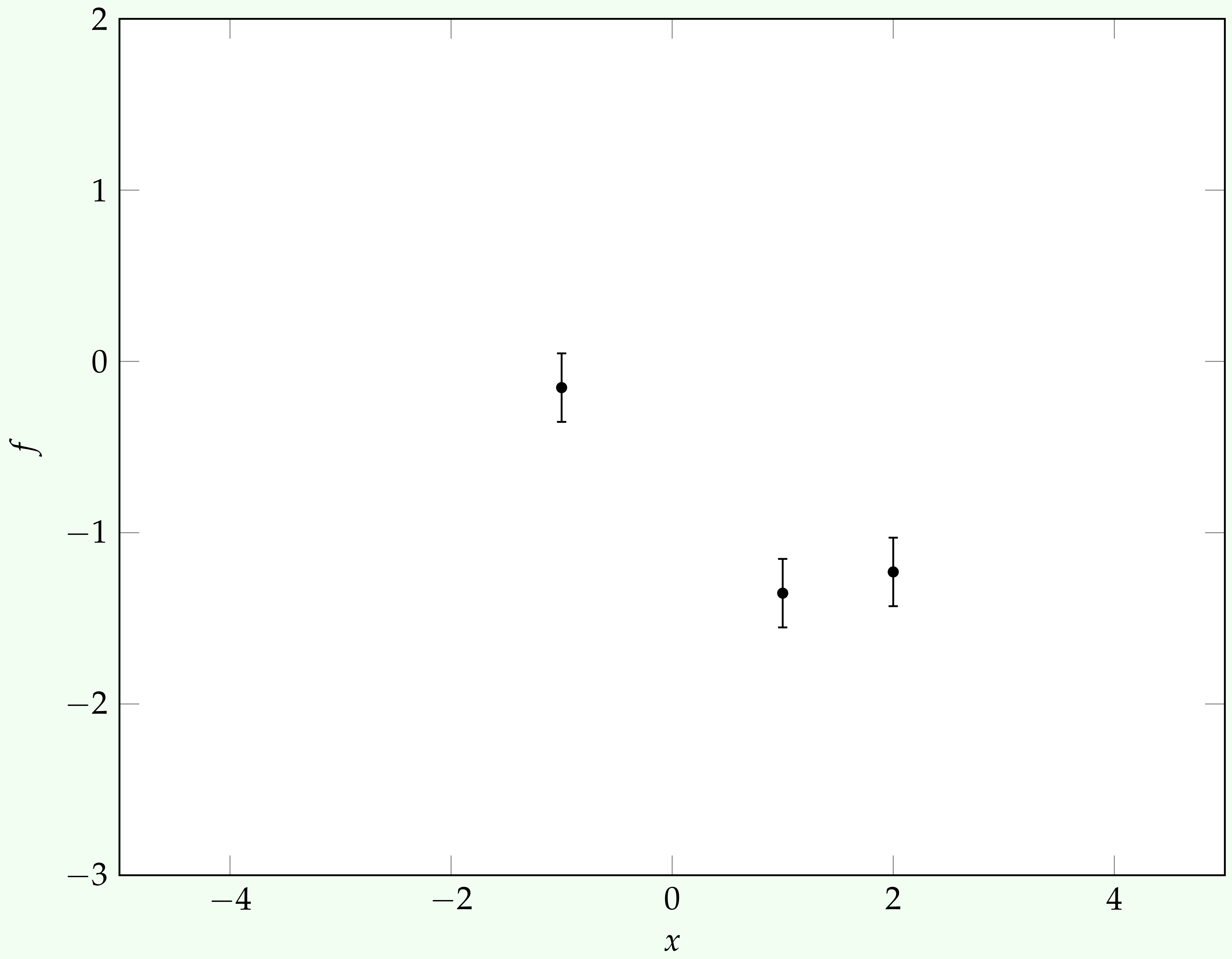
Machine learning treats algorithms as agents.

Probabilistic numerics treats numeric algorithms as agents.

An agent receives data, predicts, & then makes decisions.

In global optimisation:

data = ?;
predictand = ?; &
decisions = ?.


2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x

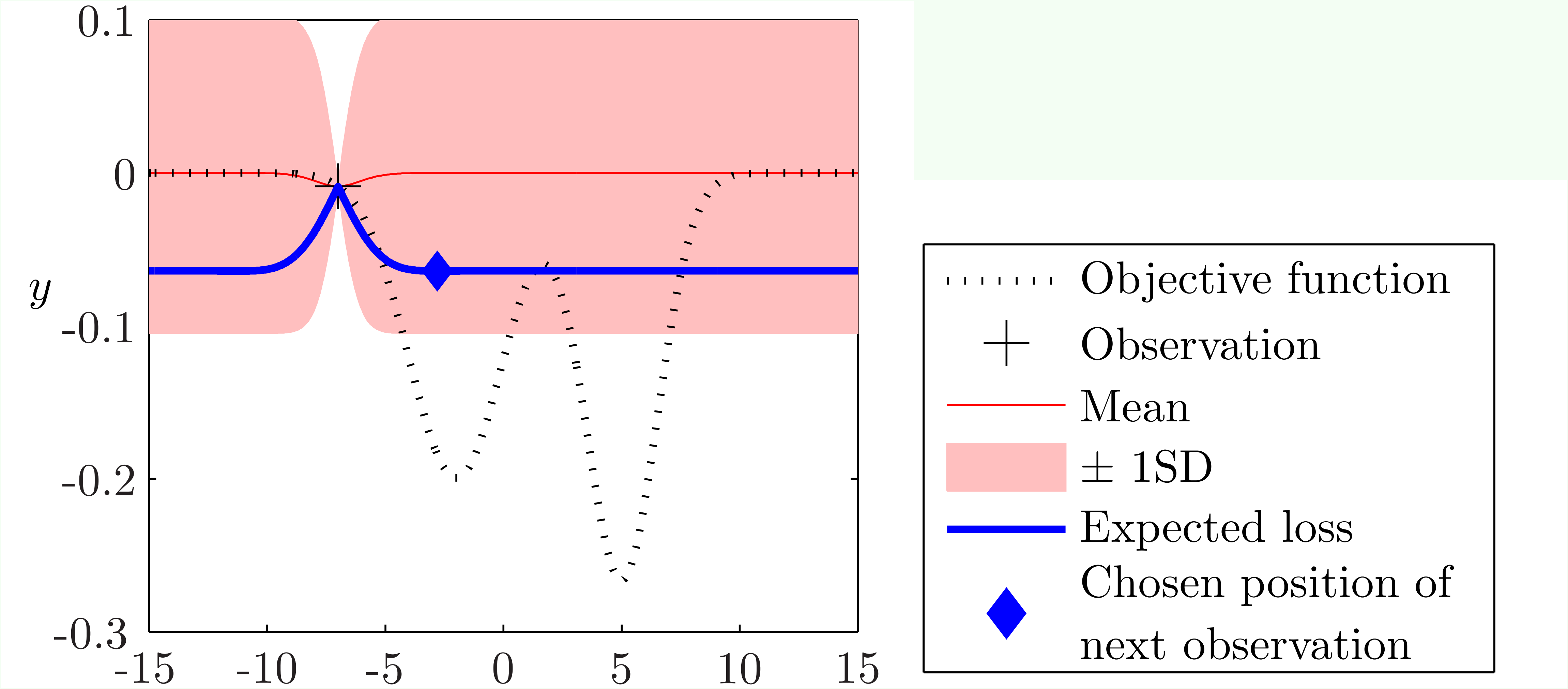
In global optimisation:

data = evaluations;

predictand = minimiser; &

decisions = locations.

Bayesian optimisation is probabilistic numerics for global optimisation.



An agent is defined by its **prior** and loss function.

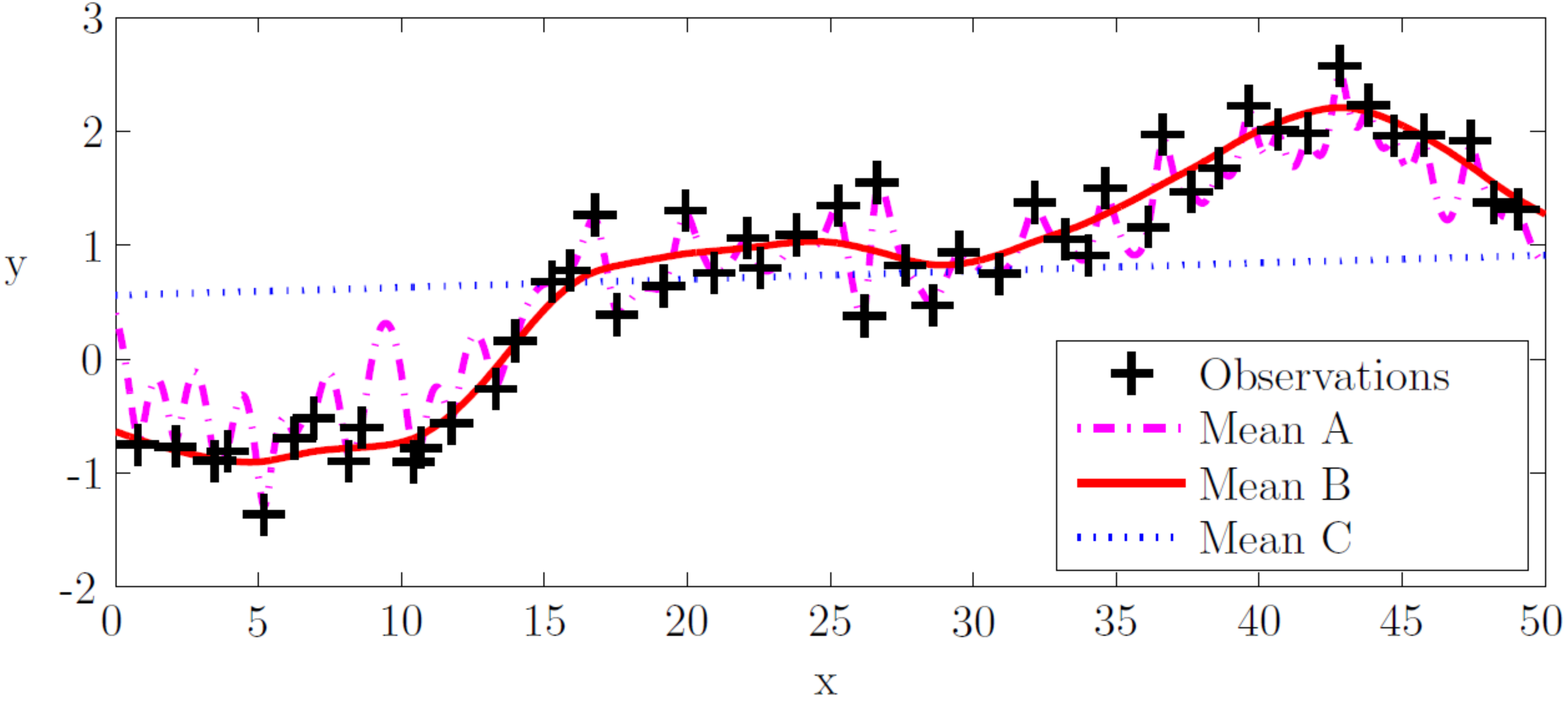
The surrogate is the prior for the objective: options include

Gaussian processes,

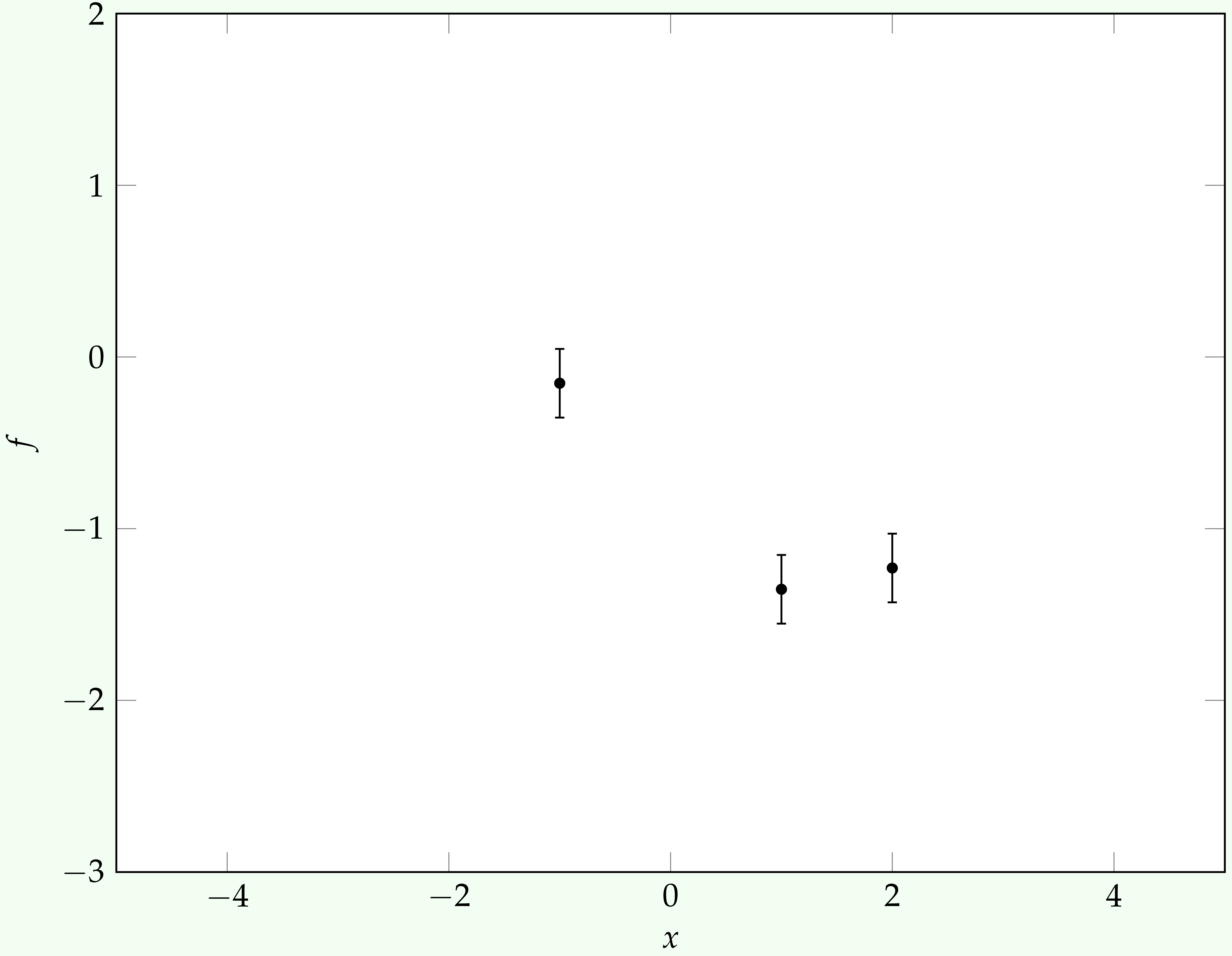
random forests,

tree-structured Parzen (density) estimators and

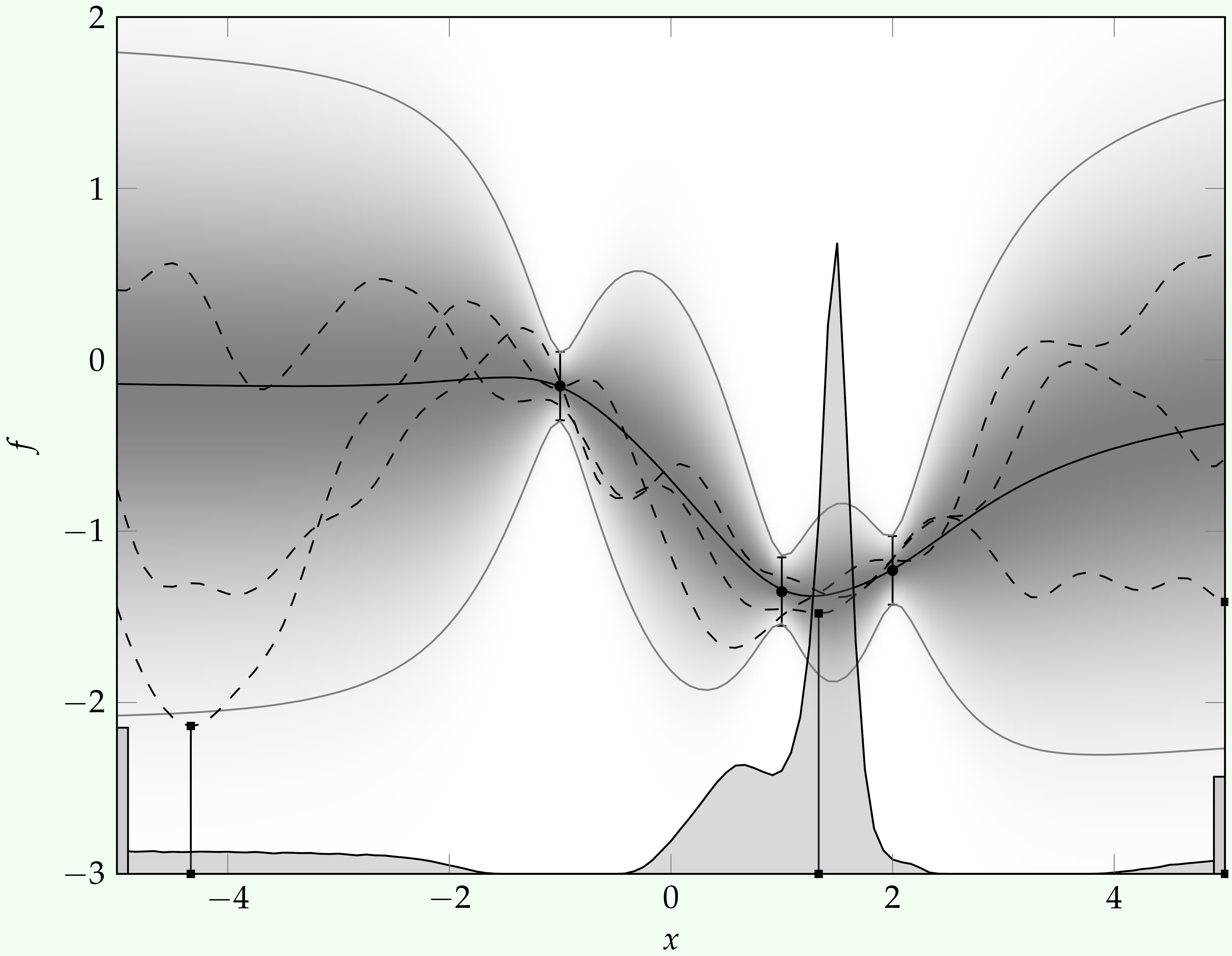
Bayesian neural networks.


3
2
1
0
-1
-2
y
0
5
10
15
20
25
30
35
40
45
50
x
Observations
Mean A
Mean B
Mean C

TO IMPROVE OPTIMISATION, IMPROVE YOUR SURROGATE.

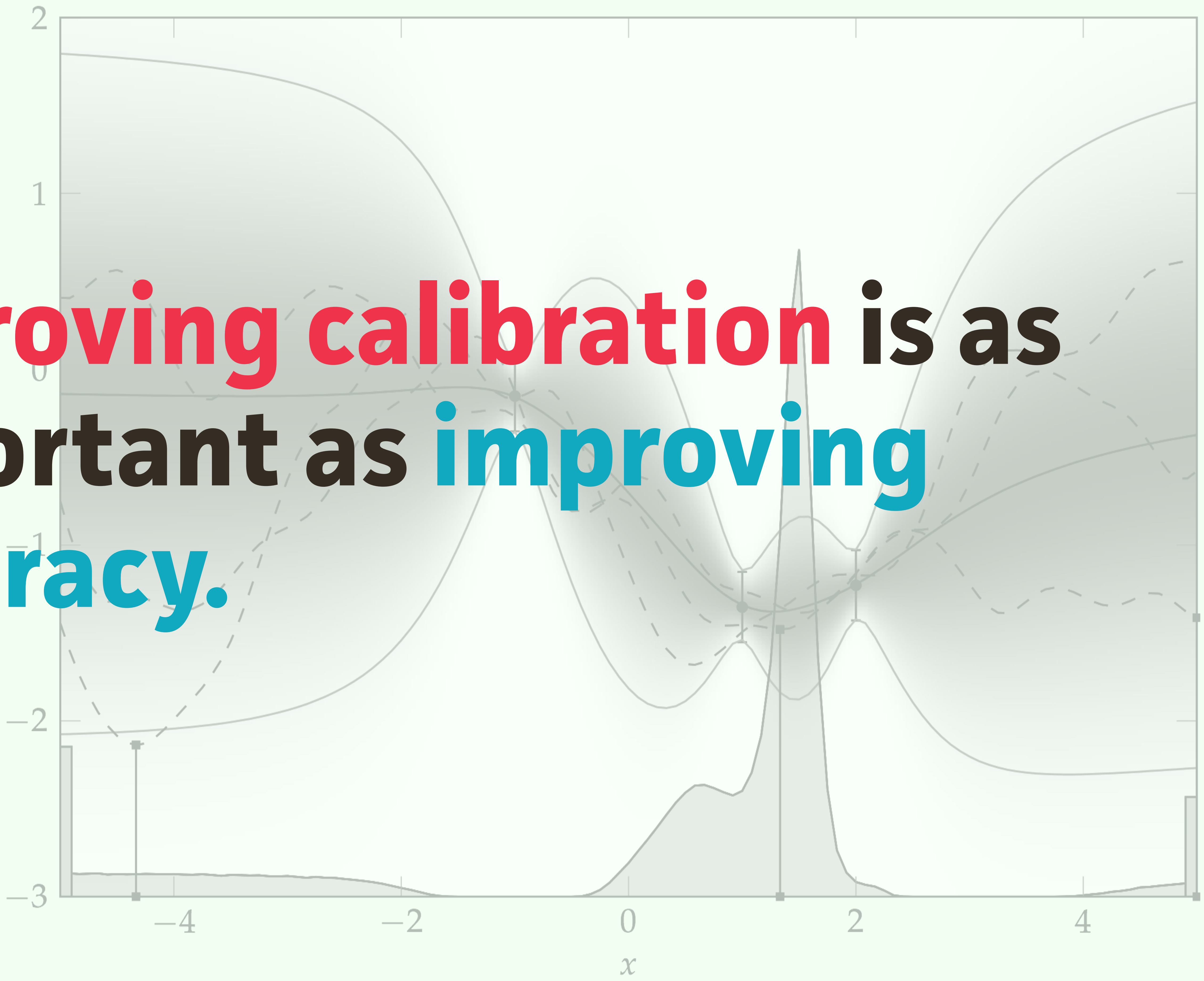


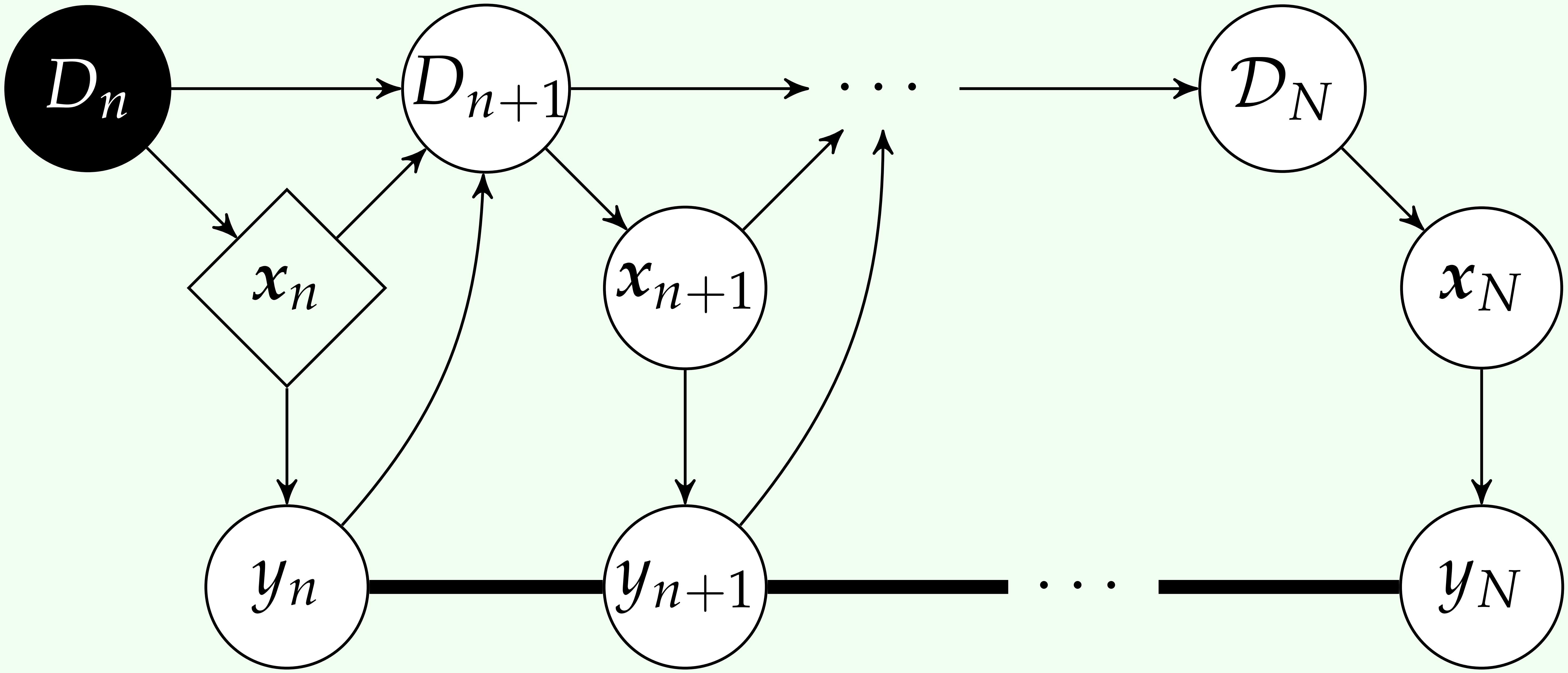
2
1
0
−1
−2
−3
−4
−2
0
2
4
f
x



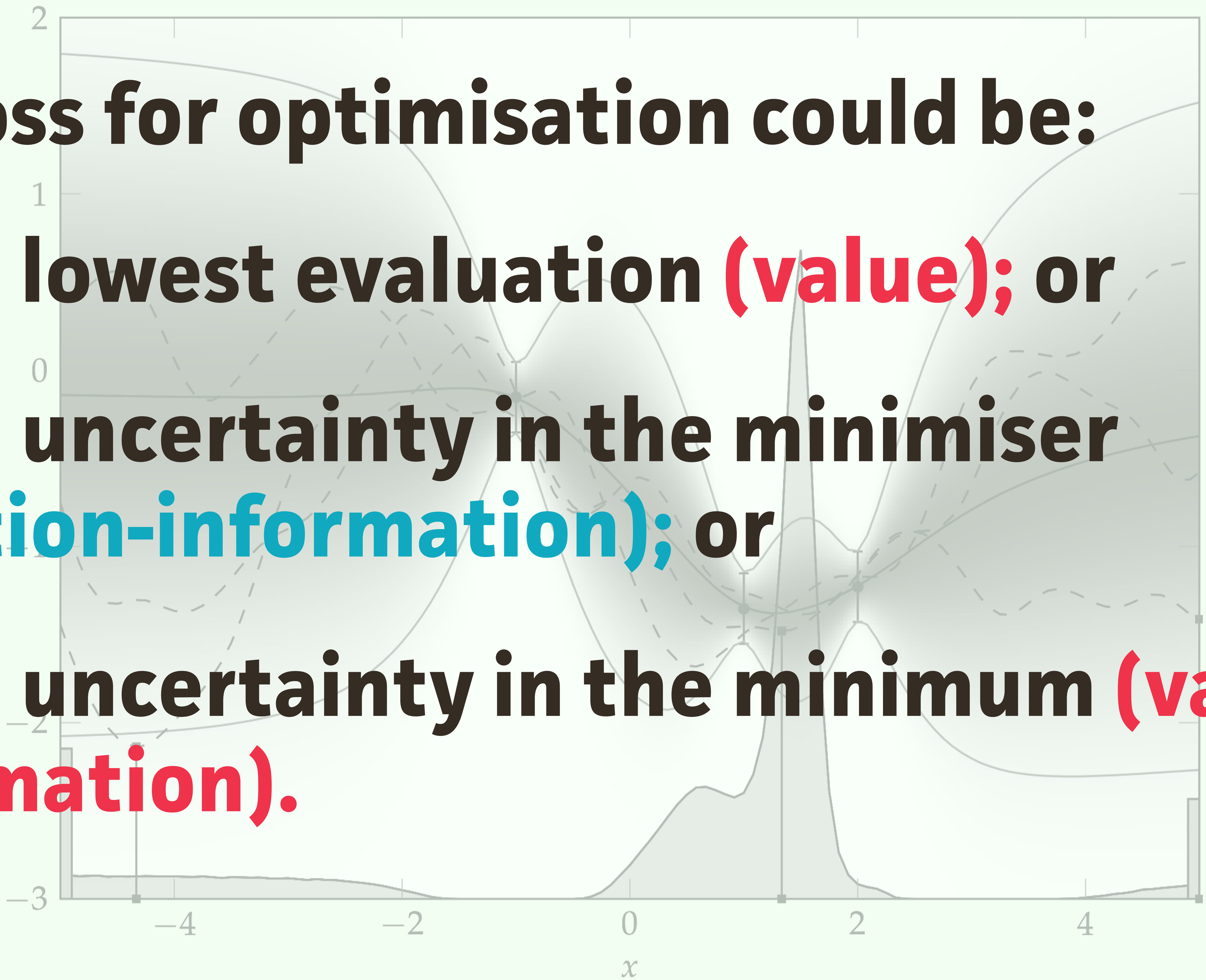
2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x

Improving calibration is as important as improving accuracy.




D_n
D_{n+1}
$\cdots$
$\mathcal{D}_N$
$\boldsymbol{x}_n$
$\boldsymbol{x}_{n+1}$
$\boldsymbol{x}_N$
y_n
y_{n+1}
$\cdots$
y_N

What should we pick as the **loss function** for **optimisation?**



The loss for optimisation could be:

1. the lowest evaluation **(value)**; or

2. the uncertainty in the minimiser **(location-information)**; or

3. the uncertainty in the minimum **(value-information)**.

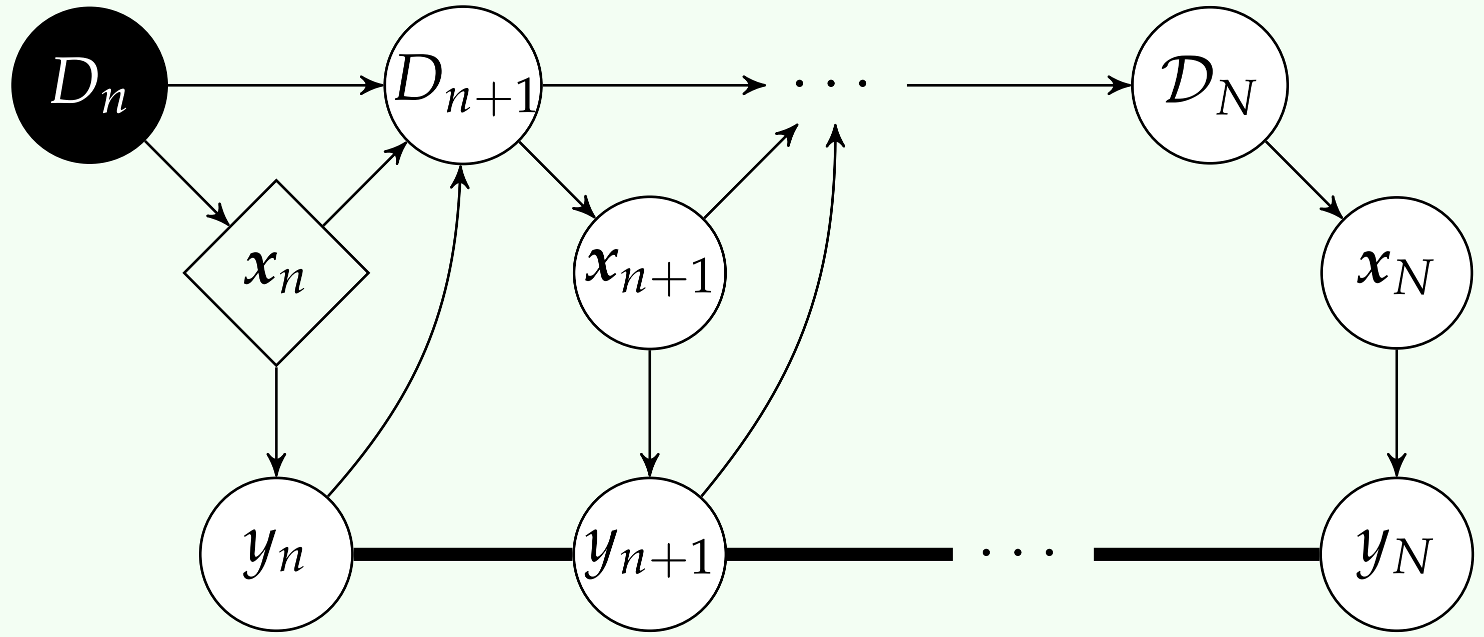
1. **Value:** $\lambda_{\mathrm{VL}} := y_N$.

2. **Location-information:**

$$\lambda_{\mathrm{LIL}} := \mathbb{H}(\boldsymbol{x}_* \mid \boldsymbol{x}_N, y_N, \mathcal{D}_N).$$

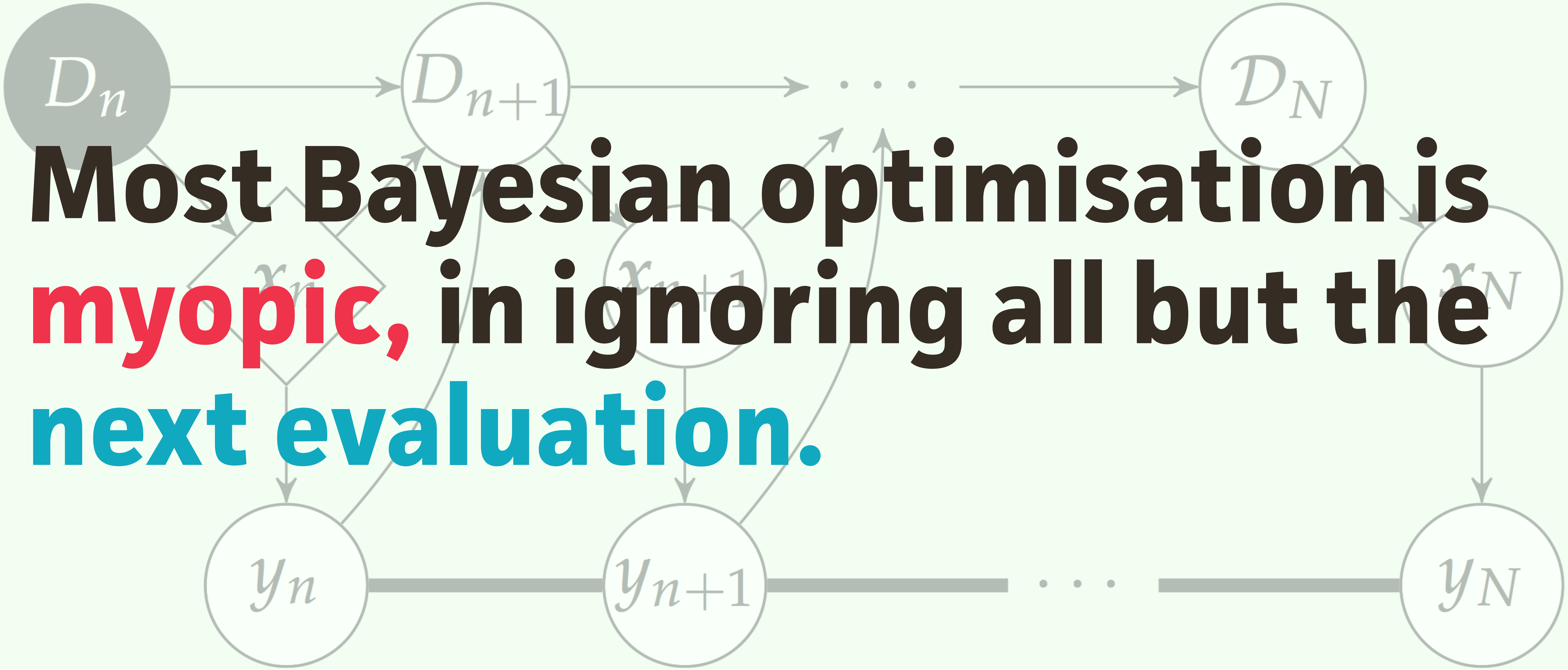
2. **Value-information:**

$$\lambda_{\mathrm{VIL}} := \mathbb{H}(y_* \mid \boldsymbol{x}_N, y_N, \mathcal{D}_N).$$



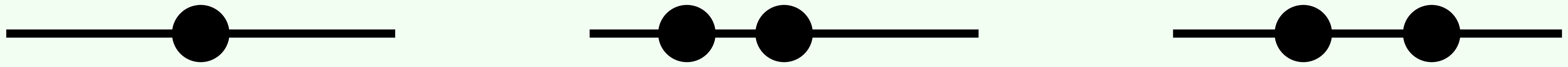
The minimiser is $\boldsymbol{x}_*$ and the minimum y_*.

An acquisition function is an expected loss function.

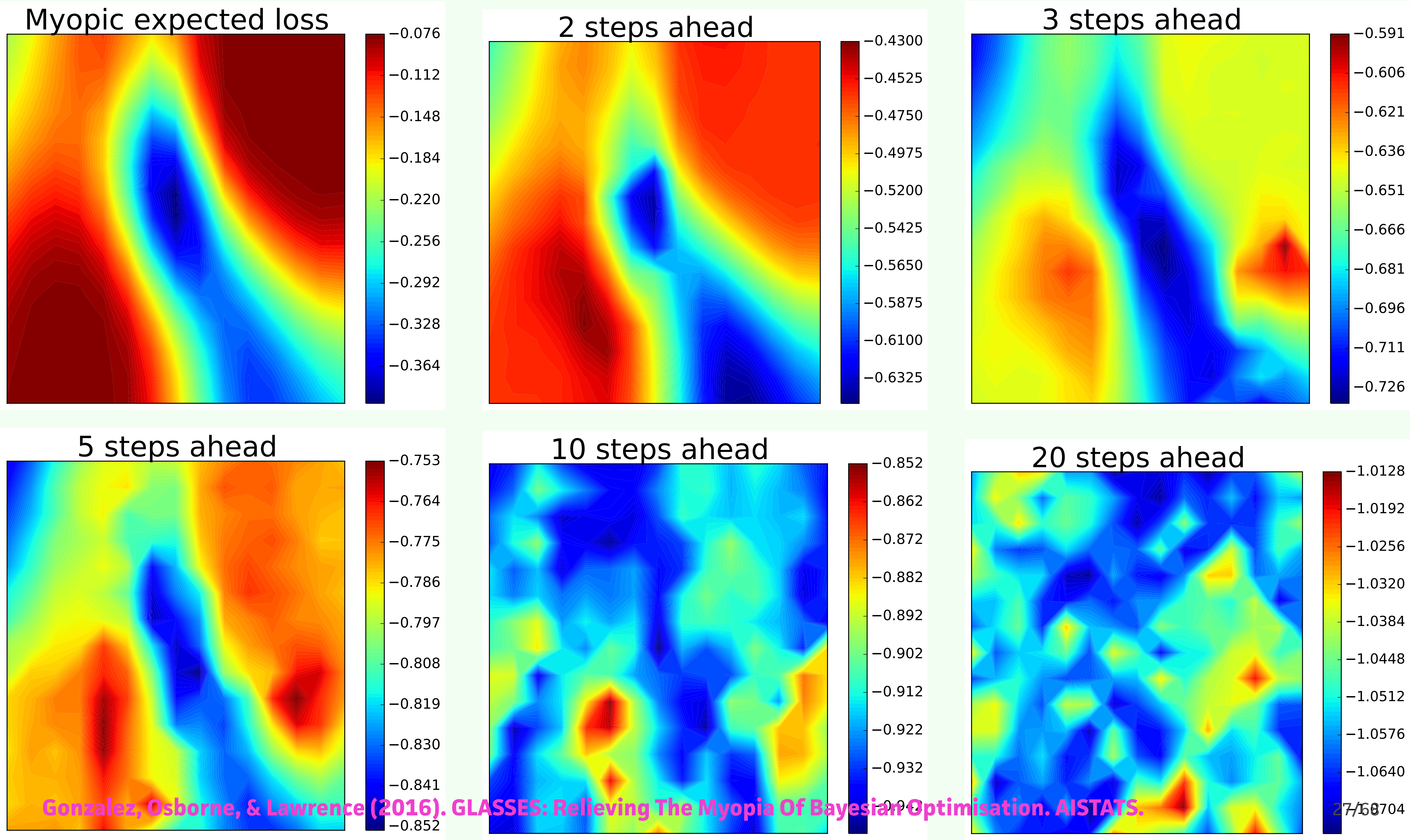


Most Bayesian optimisation is myopic, in ignoring all but the next evaluation.

Myopia can lead to insufficient exploration.



On the other hand, any flaws of a surrogate are magnified by non-myopia.



Gonzalez, Osborne, & Lawrence (2016). GLASSES: Relieving The Myopia Of Bayesian Optimisation. AISTATS.

With a myopic strategy, the acquisition function is

$$\alpha(\boldsymbol{x}_n \mid \mathcal{D}_n) = \mathbb{E}\big(\lambda(\boldsymbol{x}_n, y_n, \mathcal{D}_n)\big)$$
$$= \int \lambda(\boldsymbol{x}_n, y_n, \mathcal{D}_n)\, p(y_n \mid \mathcal{D}_n)\, \mathrm{d}y_n.$$

The next evaluation location will be

$$\boldsymbol{x}_n = \arg\min_{\boldsymbol{x}} \alpha(\boldsymbol{x} \mid \mathcal{D}_n).$$

$$\boldsymbol{x}_n = \arg\min_{\boldsymbol{x}} \alpha(\boldsymbol{x} \mid \mathcal{D}_n).$$

We have succeeded in turning optimisation into optimisation.

The acquisition function:

is less expensive than the objective;

gives us gradients and Hessians; and

need not be optimised exactly.

Expected improvement

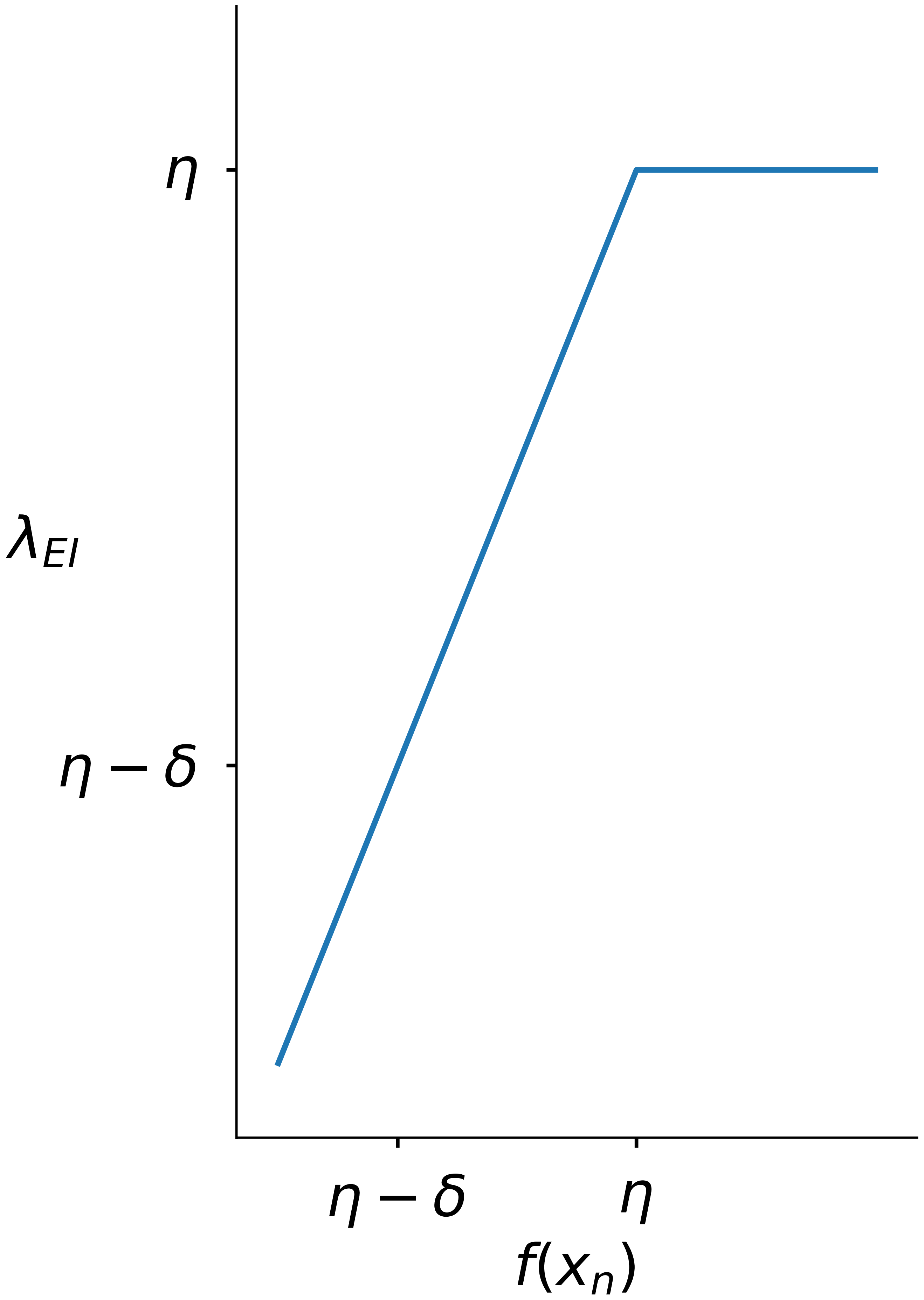
is a myopic approximation to the value loss:

$$\begin{aligned} & \lambda_{\mathrm{VL}}(\boldsymbol{x}_N, f(\boldsymbol{x}_N), \mathcal{D}_N) \\ \simeq \; & \lambda_{\mathrm{EI}}(\mathcal{D}_{n+1}) \\ \coloneqq \; & \min_{i \in \{0, \ldots, n\}} f(\boldsymbol{x}_i). \end{aligned}$$

Defining the lowest function value available at the nth step as

$$\eta \coloneqq \min_{i \in \{0, \ldots, n-1\}} f(\boldsymbol{x}_i),$$

we can simply rewrite the loss as $\lambda_{\mathrm{EI}}(\mathcal{D}_{n+1}) = \min\{\eta, f(\boldsymbol{x}_n)\}$.



If we have a Gaussian posterior for the next evaluation,

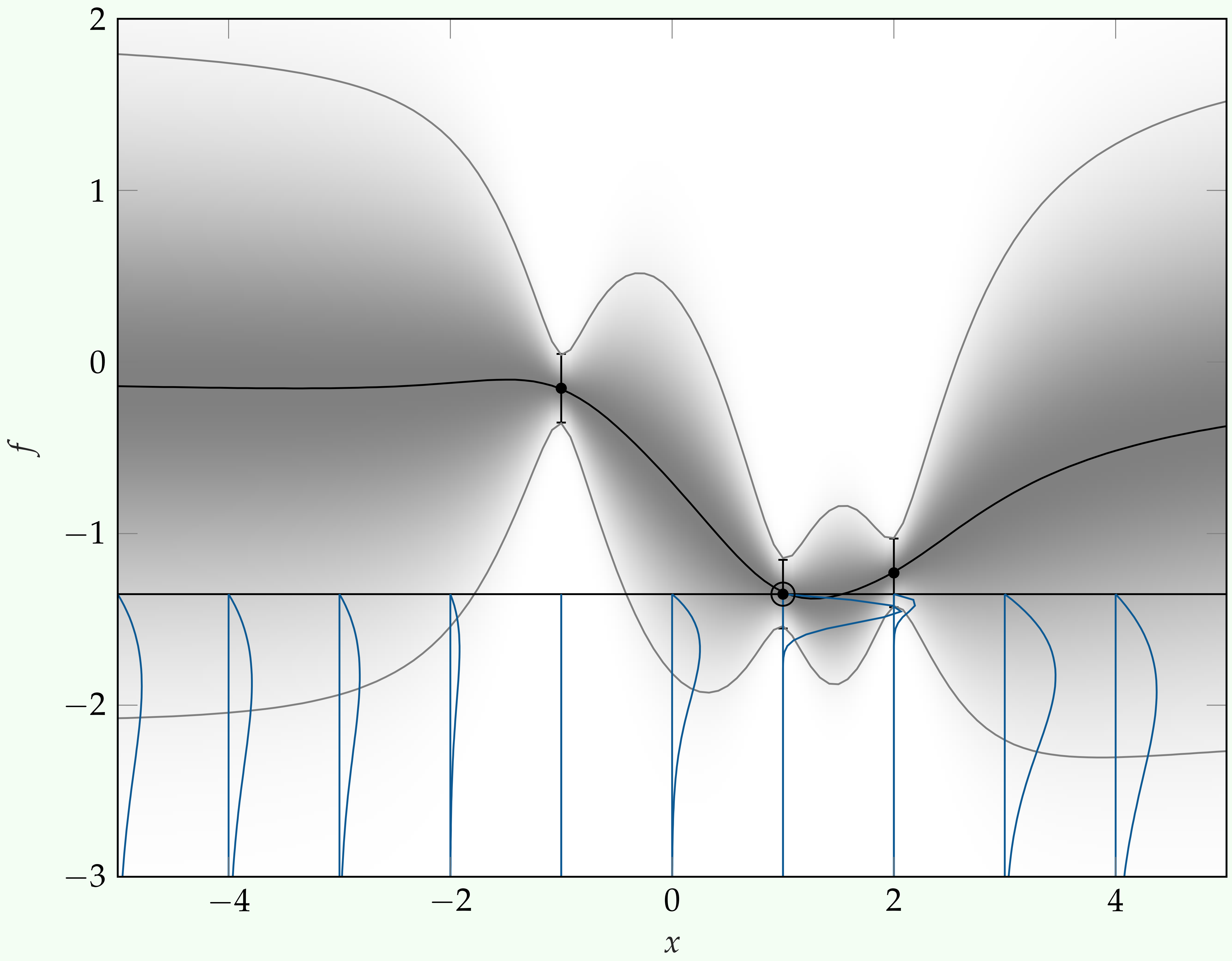
$$p\big(f(\boldsymbol{x}_n) \mid \mathcal{D}_n\big) := \mathcal{N}\big(f(\boldsymbol{x}_n); m(\boldsymbol{x}_n), V(\boldsymbol{x}_n)\big),$$

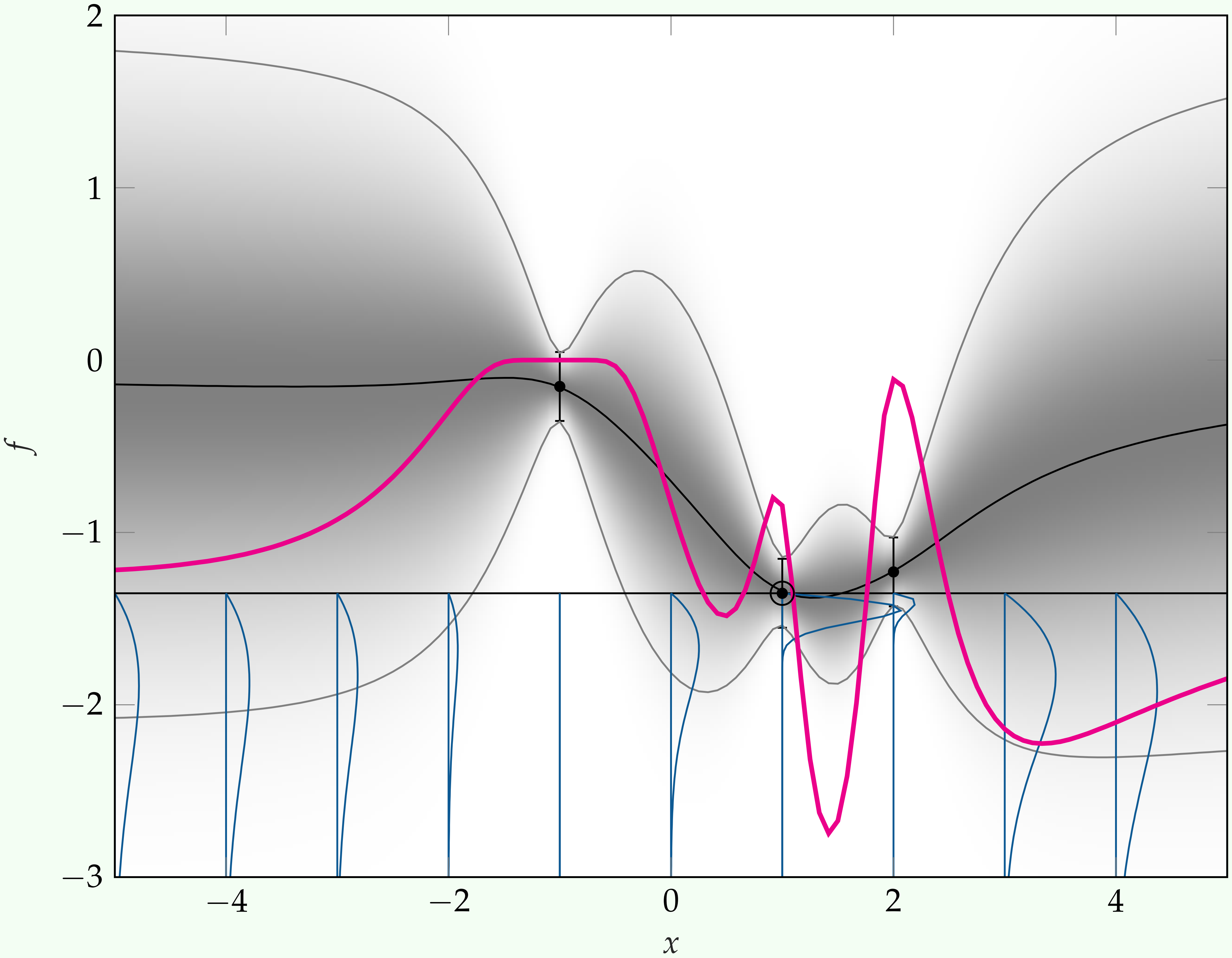
the **expected improvement acquisition function** is

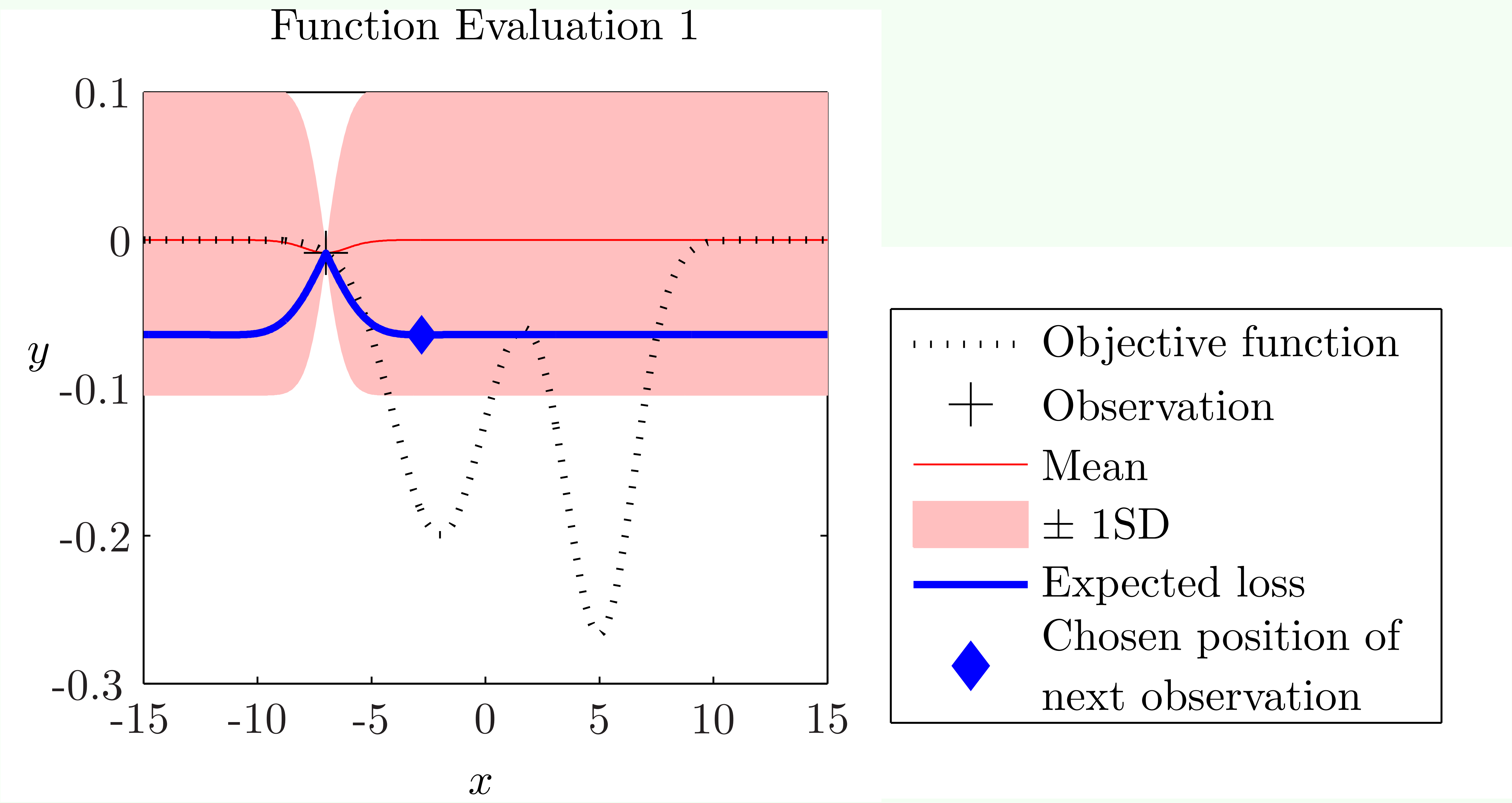
$$\begin{aligned}
\alpha_{\mathrm{EI}}(\boldsymbol{x}_n) := \quad & \mathbb{E}\big(\lambda_{\mathrm{EI}}\big)(\boldsymbol{x}_n) - \eta \\
= \quad & \int_{-\infty}^{\eta} \big(f(\boldsymbol{x}_n) - \eta\big) p\big(f(\boldsymbol{x}_n) \mid \mathcal{D}_n\big)\, \mathrm{d}f(\boldsymbol{x}_n) \\
= \quad & -V(\boldsymbol{x}_n)\mathcal{N}\big(\eta; m(\boldsymbol{x}_n), V(\boldsymbol{x}_n)\big) \\
& + \big(m(\boldsymbol{x}_n) - \eta\big)\, \Phi\big(\eta; m(\boldsymbol{x}_n), V(\boldsymbol{x}_n)\big).
\end{aligned}$$

Source: Močkus, Tiesis, and Žilinskas. "The Application of Bayesian Methods for Seeking the Extremum". 1978.

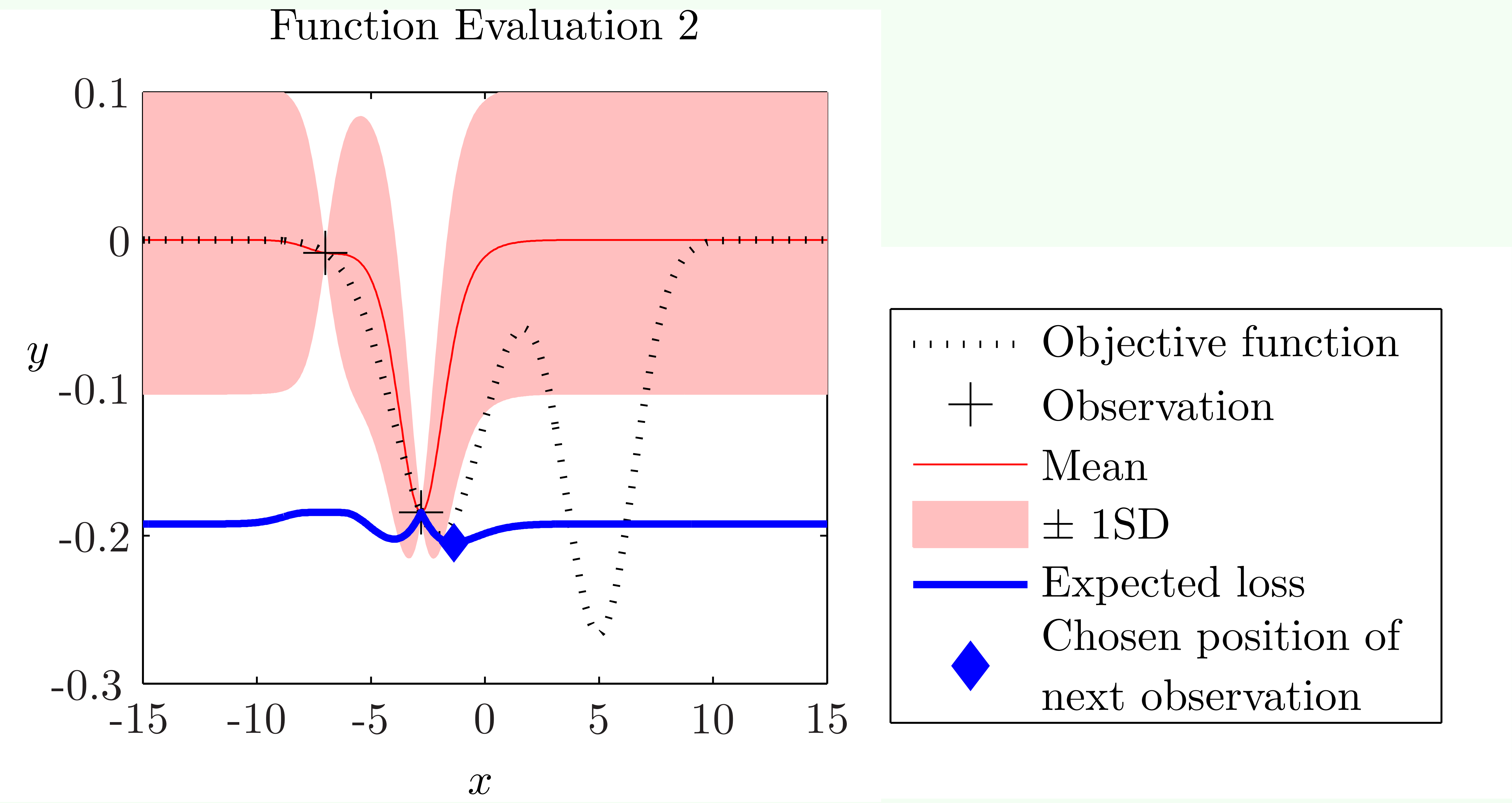
$$\alpha_{\mathrm{EI}}(\boldsymbol{x}_n) = \int_{-\infty}^{\eta} \big(f(\boldsymbol{x}_n) - \eta\big) p\big(f(\boldsymbol{x}_n) \mid \mathcal{D}_n\big) \, \mathrm{d}f(\boldsymbol{x}_n)$$


2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x

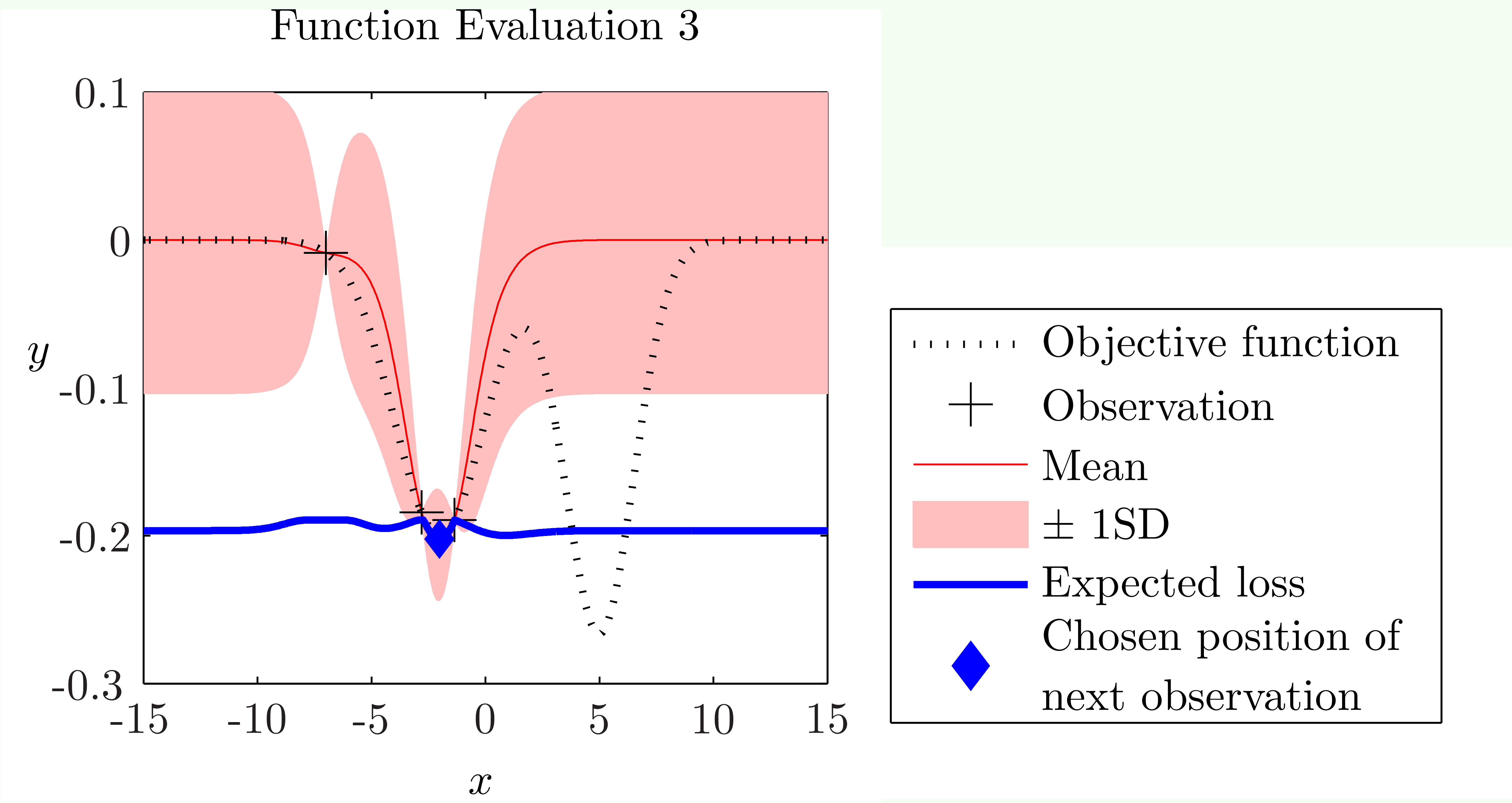

2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x

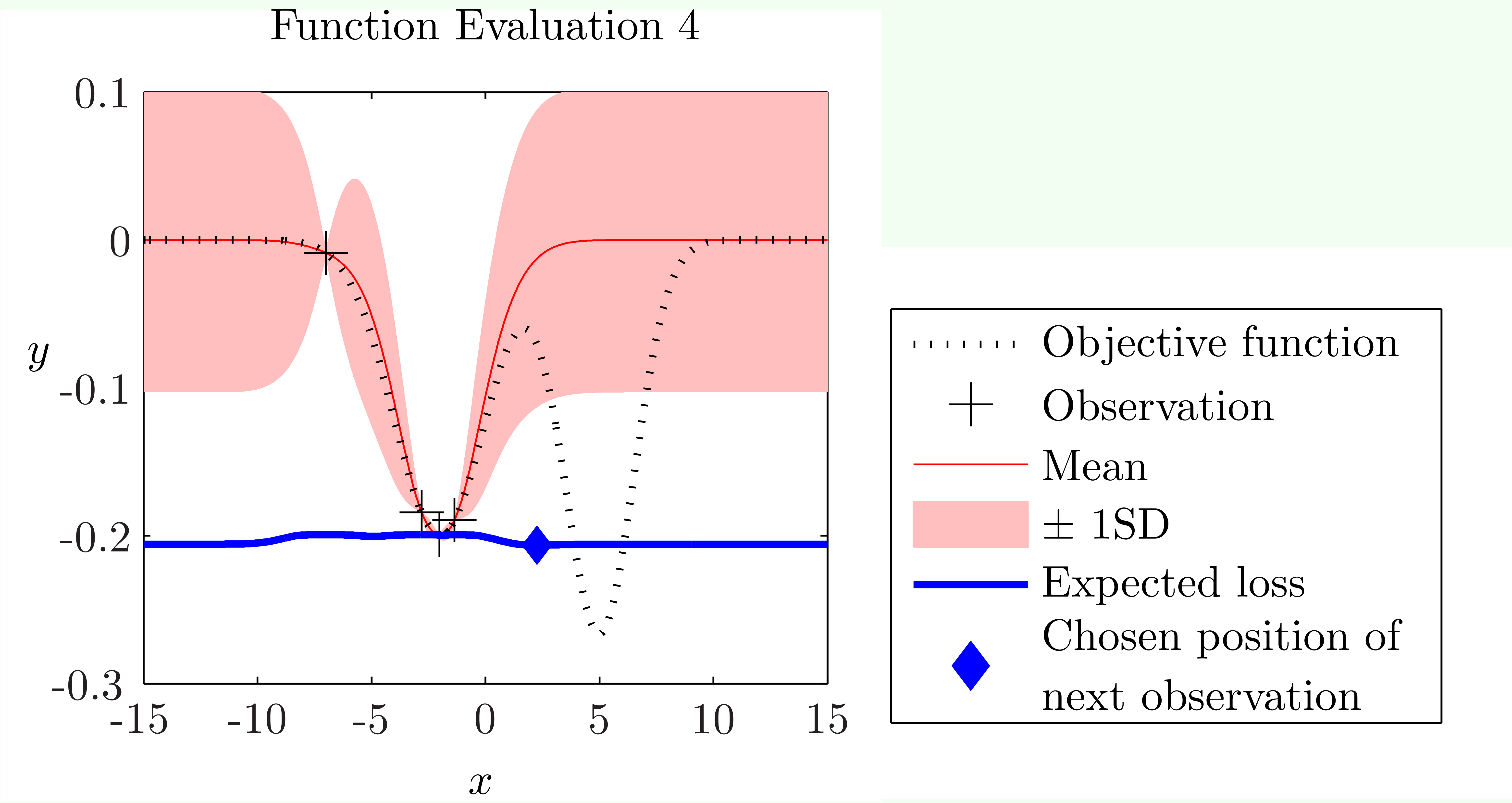


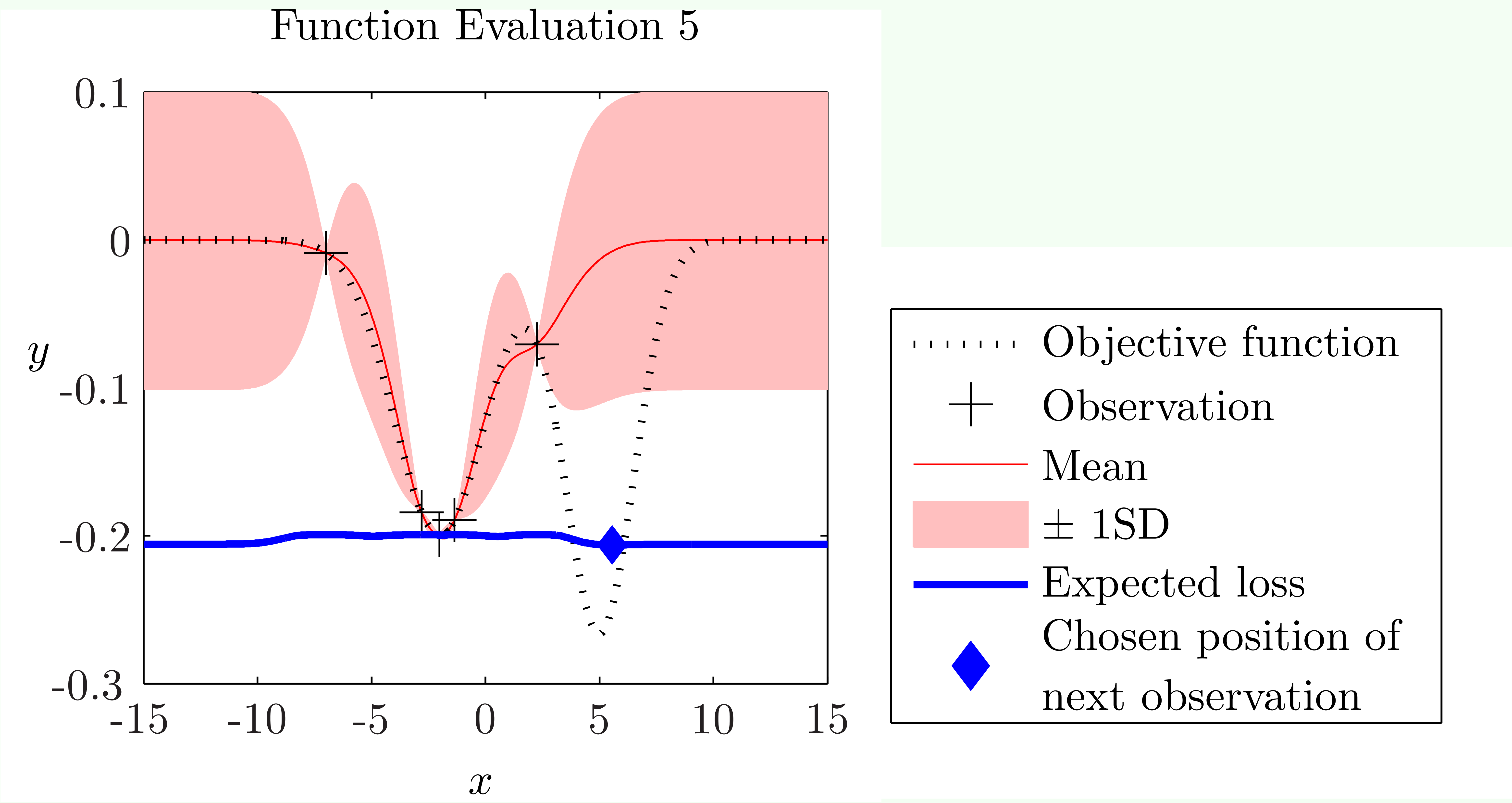
Function Evaluation 1
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation

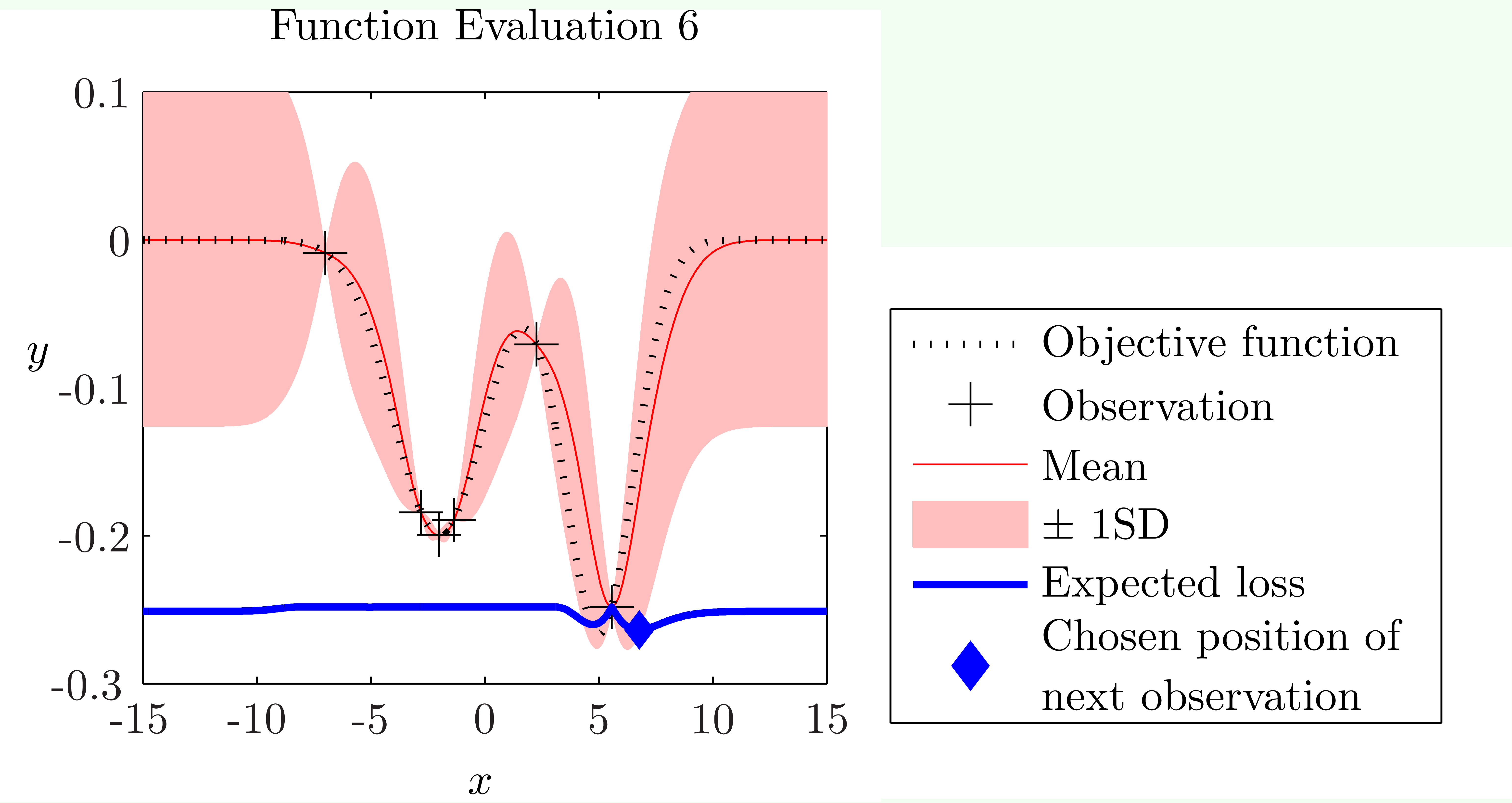


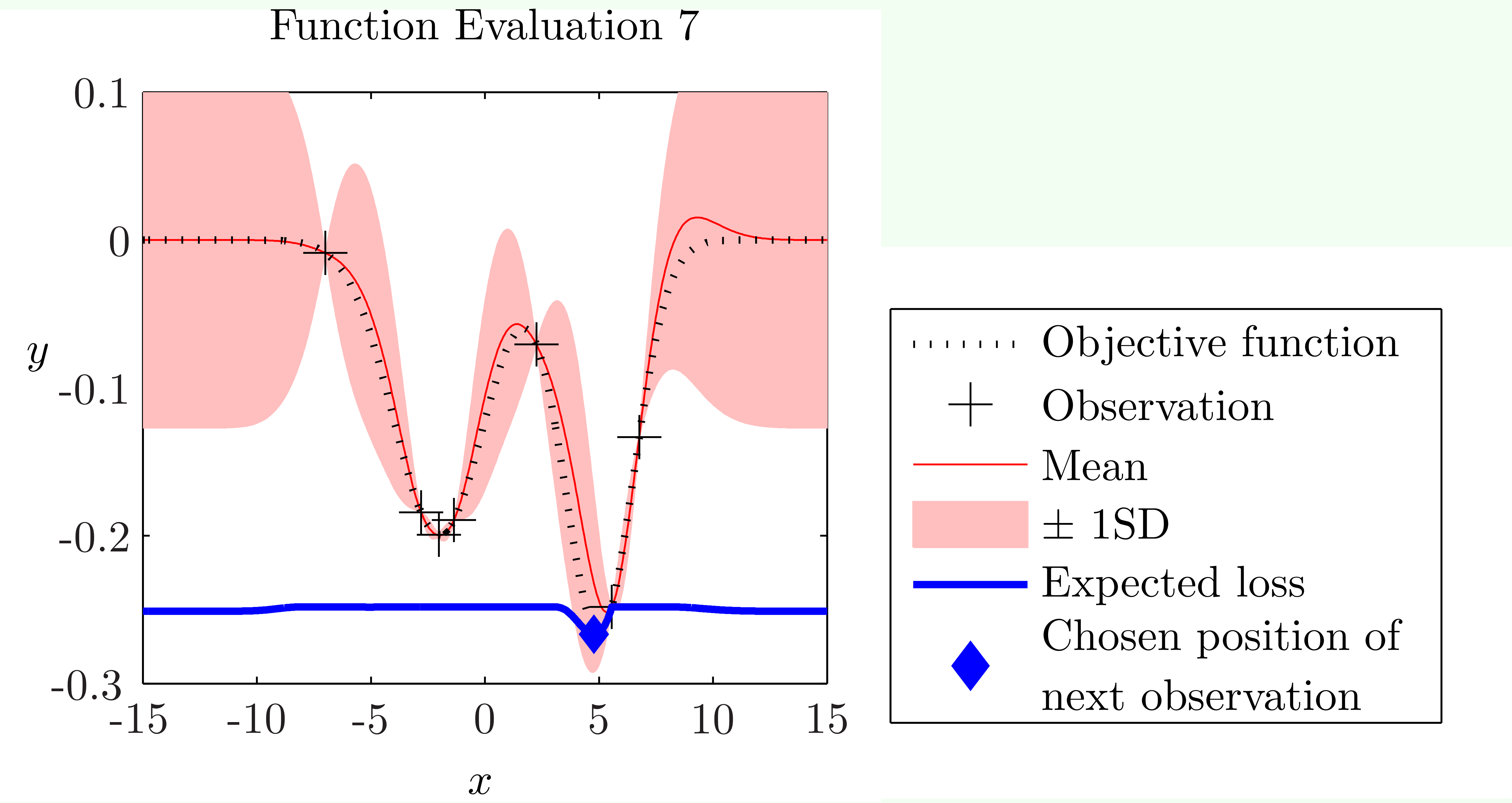
Function Evaluation 2
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation

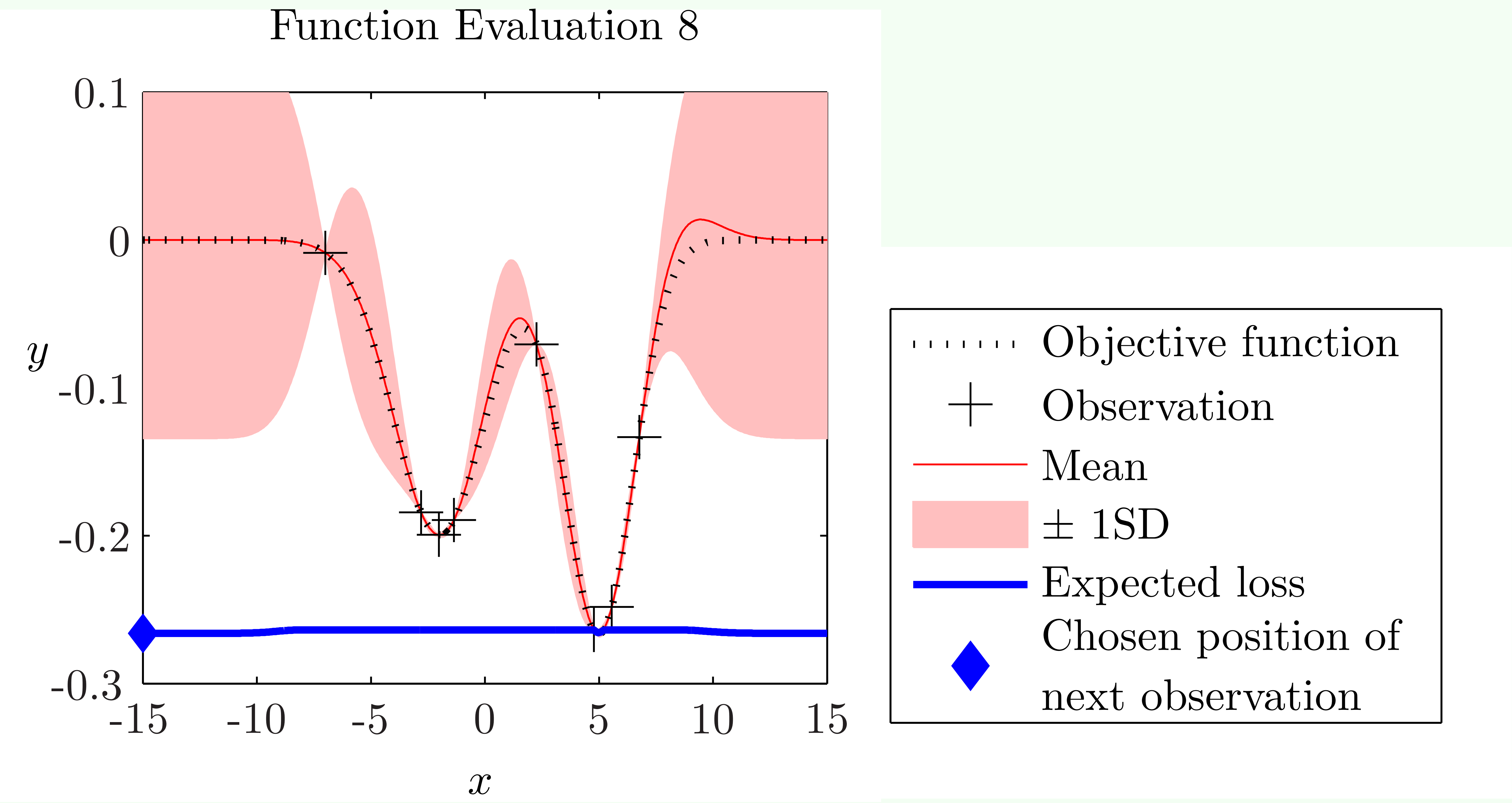

Function Evaluation 3
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation


Function Evaluation 4
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation

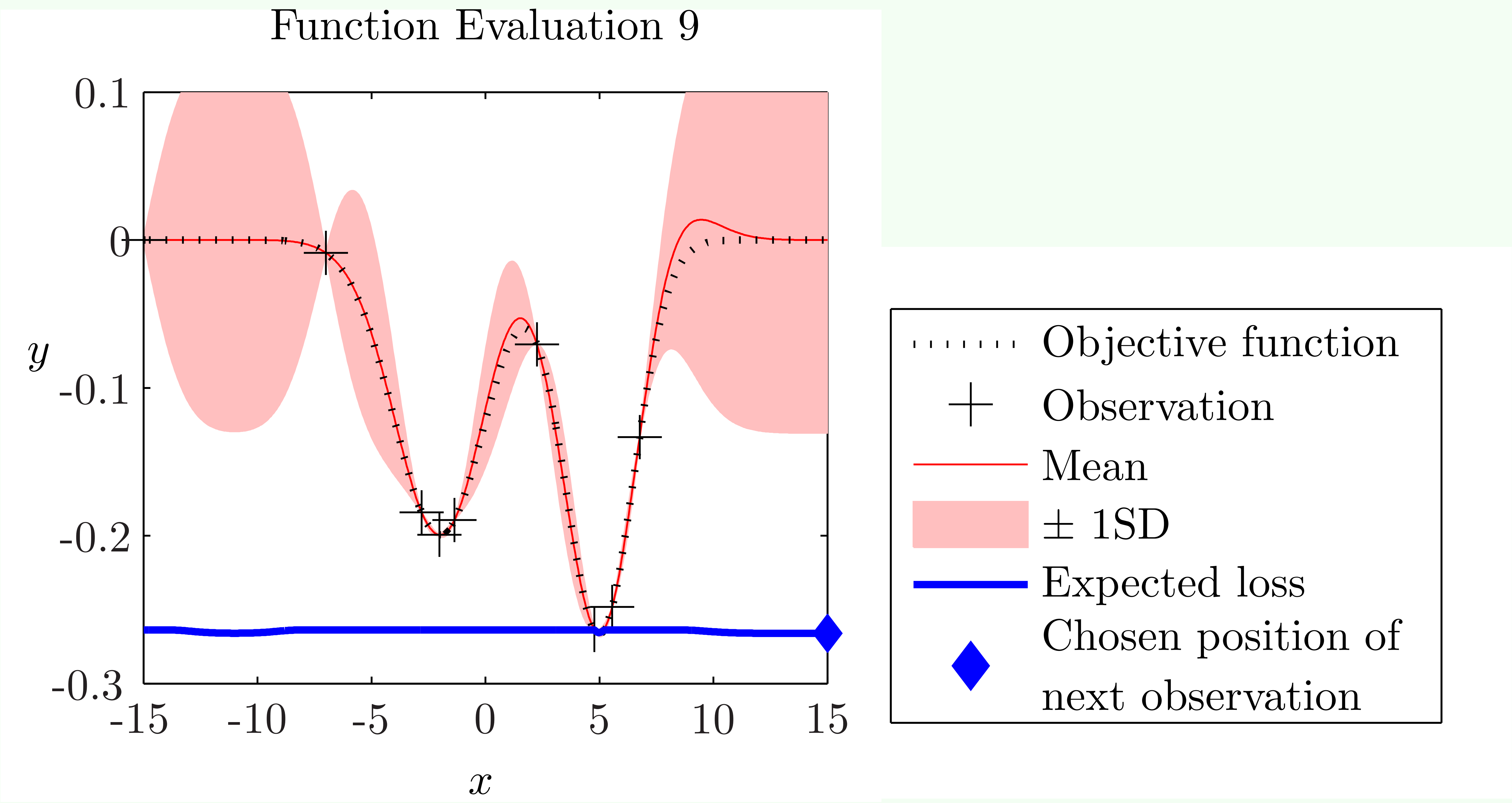

Function Evaluation 5
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation


Function Evaluation 6
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation

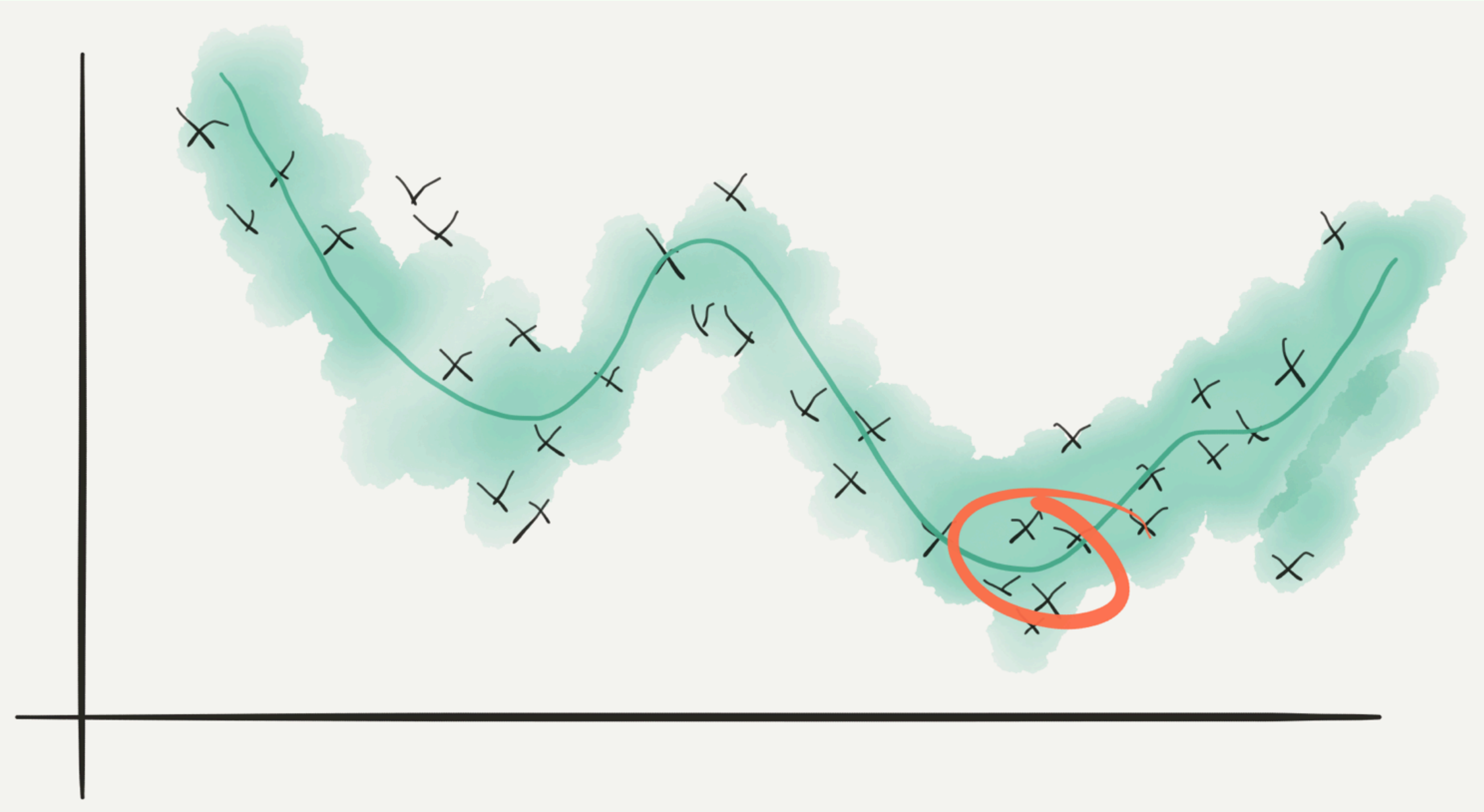

Function Evaluation 7
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation



Function Evaluation 8
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation


Function Evaluation 9
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation

If our evaluations are noisy, the best evaluation (η) is also probably the most noise-corrupted.



Probability of improvement

defines (for $\mathbb{I}$ the indicator function) the myopic loss

$$\lambda_{n,\mathrm{PI}}(\mathcal{D}_{n+1}) \coloneqq \mathbb{I}\big(f(\boldsymbol{x}_n) \geq \eta\big).$$

The probability of improvement acquisition function is hence

$$\alpha_{n,\mathrm{PI}}(\boldsymbol{x}_n) \coloneqq \mathbb{E}\big(\lambda_{n,\mathrm{PI}}(\mathcal{D}_{n+1})\big) = P\big(f(\boldsymbol{x}_n) \geq \eta \mid \mathcal{D}_n\big).$$

Source: H. J. Kushner. "A new method of locat- ing the maximum point of an arbitrary multipeak curve in the presence of noise" (1964).

Probability of improvement

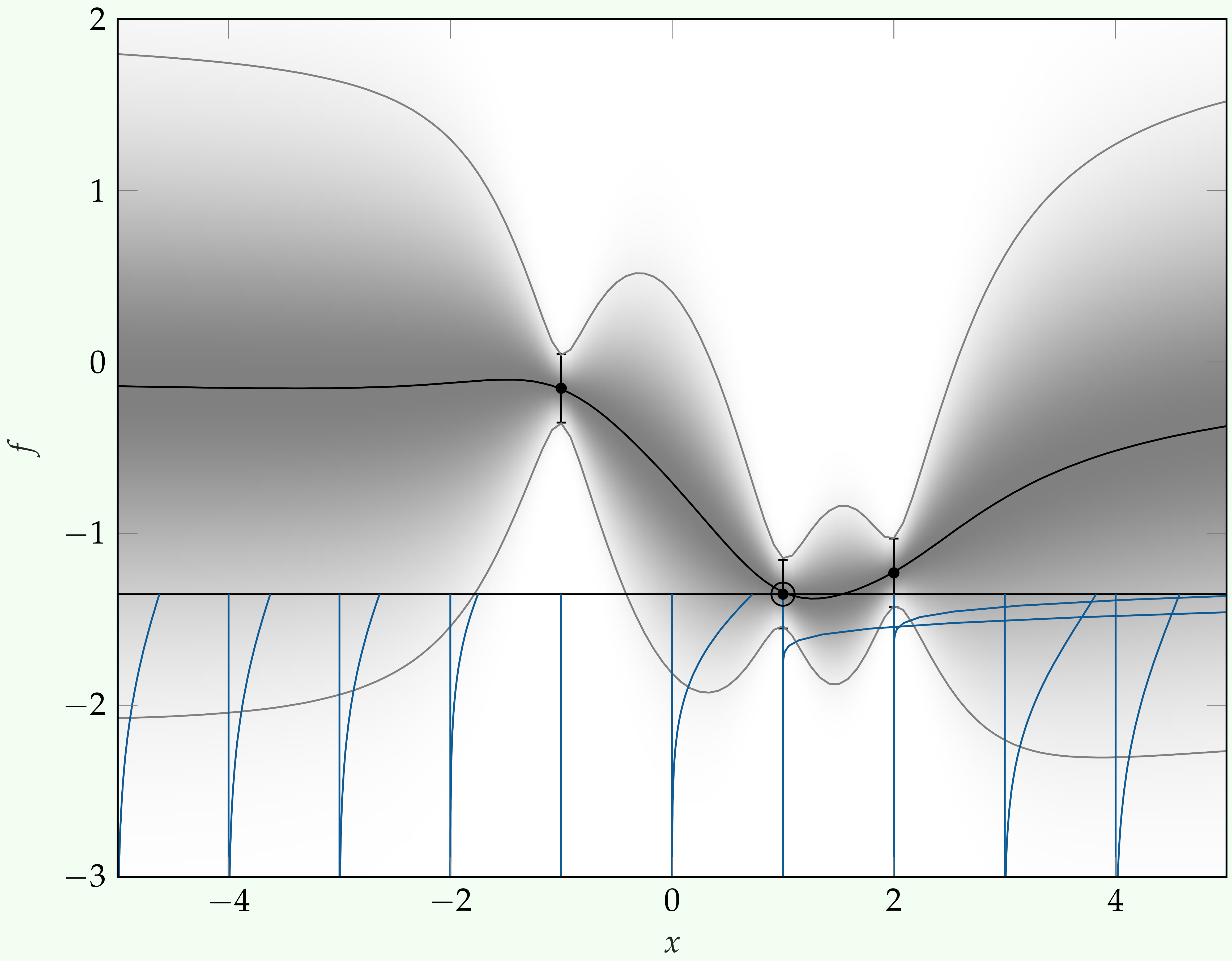
defines a myopic loss (for $\mathbb{I}$ the indicator function)

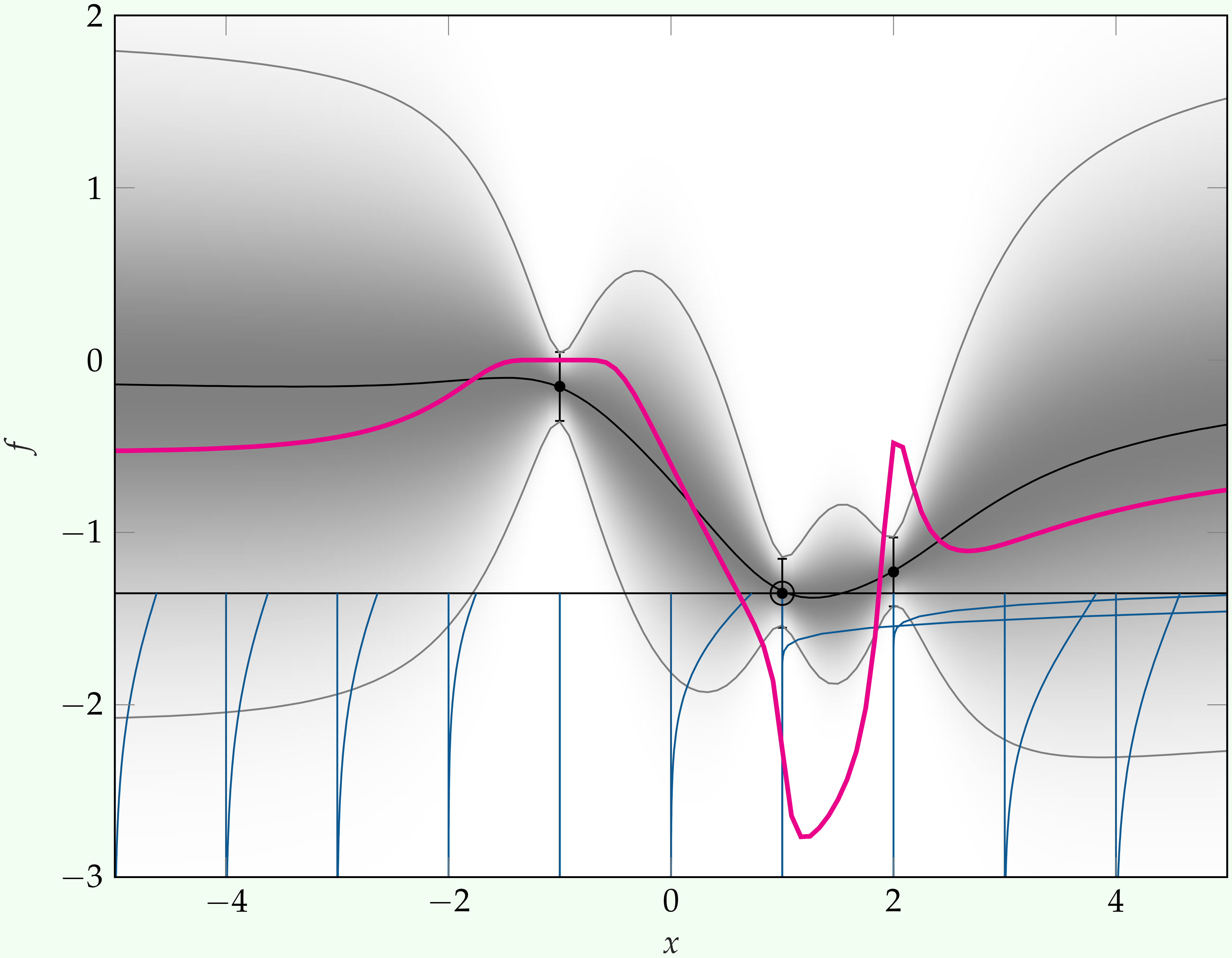
$$\lambda_{n,\mathrm{PI}}(\mathcal{D}_{n+1}) := \mathbb{I}\big(f(\boldsymbol{x}_n) \geq \eta\big).$$

The probability of improvement acquisition function is hence

$$\alpha_{n,\mathrm{PI}}(\boldsymbol{x}_n) := \mathbb{E}\big(\lambda_{n,\mathrm{PI}}(\mathcal{D}_{n+1})\big) = P\big(f(\boldsymbol{x}_n) \geq \eta \mid \mathcal{D}_n\big).$$

PI values incremental improvement every step.


2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x



2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x

Upper confidence bound

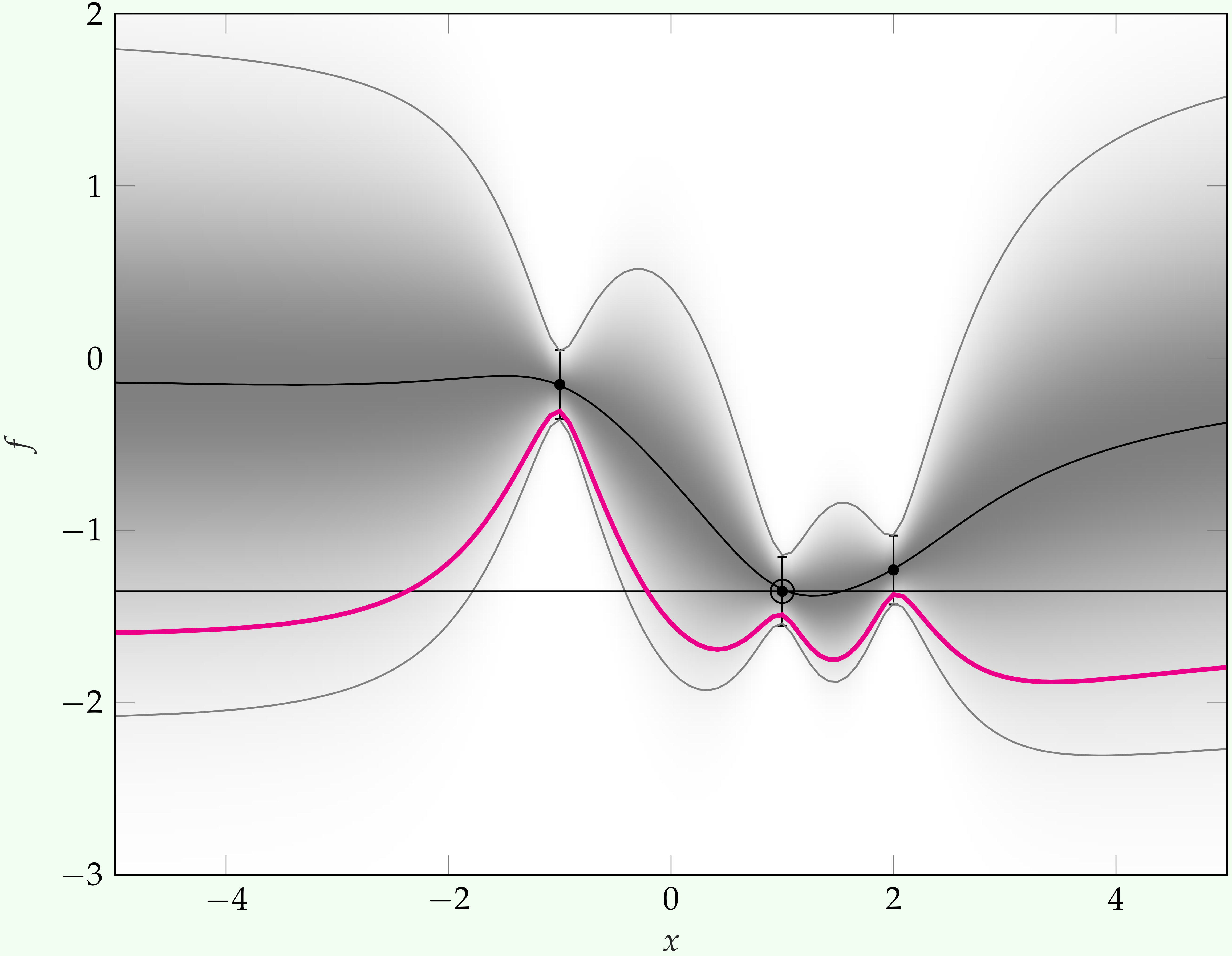
is the myopic acquisition function

$$\alpha_{\mathrm{UCB}}(\boldsymbol{x}_n) := m(\boldsymbol{x}_n) - \beta_n V(\boldsymbol{x}_n)^{\frac{1}{2}}.$$

given a surrogate with mean $m(\boldsymbol{x}_n)$ and variance $V(\boldsymbol{x}_n)$.

It is difficult to reconcile UCB with a defensible loss function.

Source: N. Srinivas et al. “Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design”. 2010.


2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x

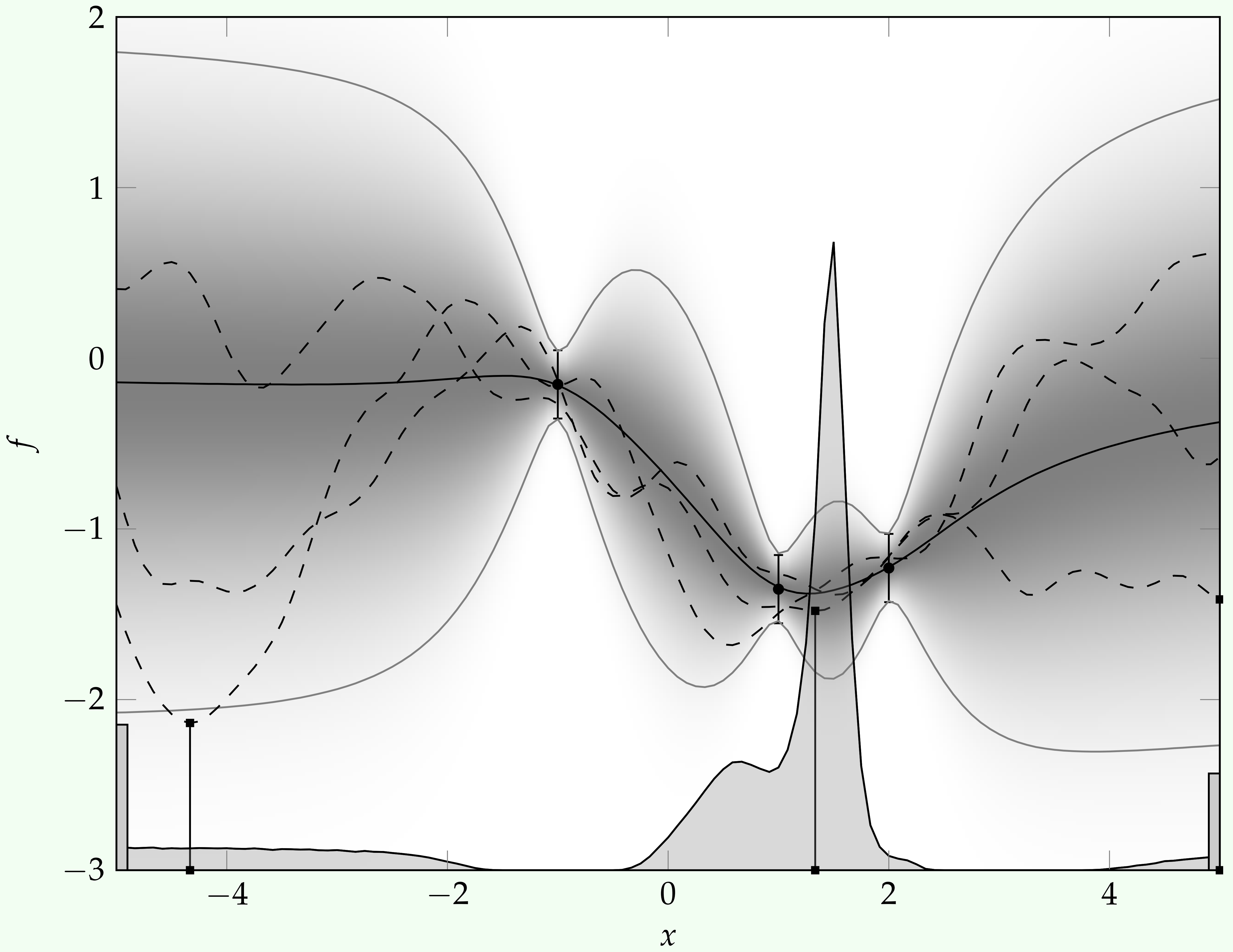
Information-theoretic methods

give alternative myopic implementations of value-information and location-information losses:

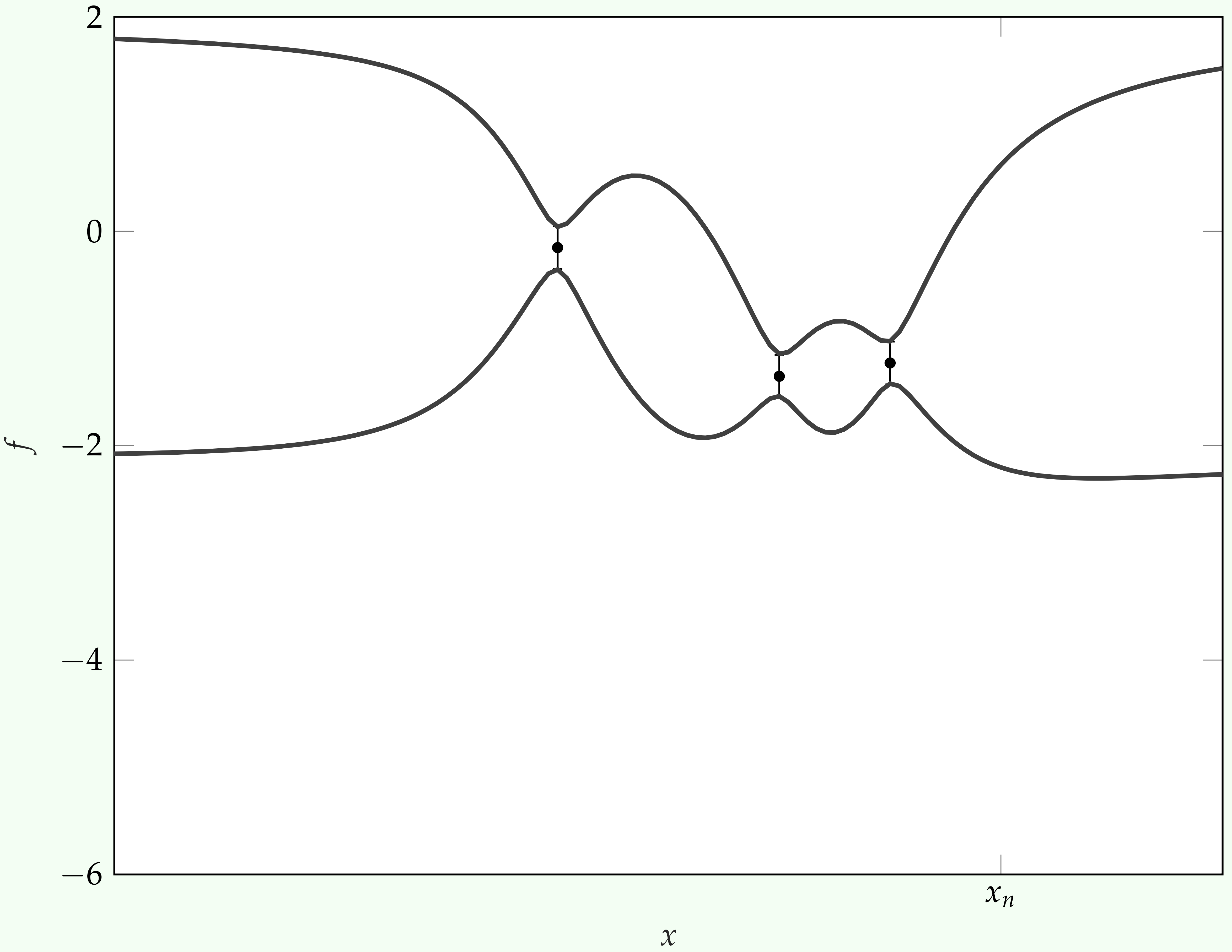
$$\begin{aligned} \alpha_{\text{LIL}} &:= \mathbb{E}_{y_n}\, \mathbb{H}(\boldsymbol{x}_* \mid y_n, \boldsymbol{x}_n, \mathcal{D}_n) \quad \text{and} \\ \alpha_{\text{VIL}} &:= \mathbb{E}_{y_n}\, \mathbb{H}(y_* \mid y_n, \boldsymbol{x}_n, \mathcal{D}_n). \end{aligned}$$

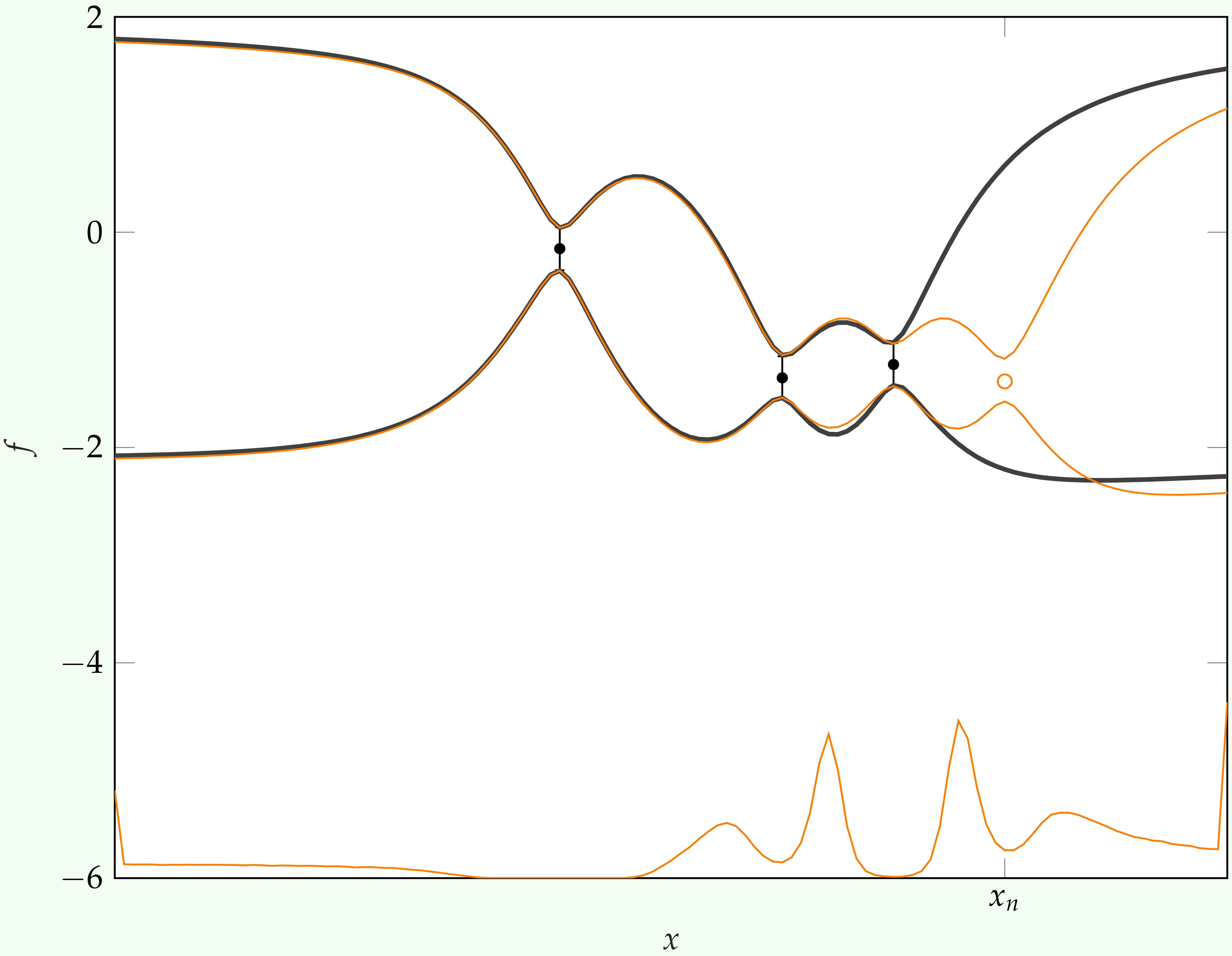
These methods tend to be more exploratory, helping performance.

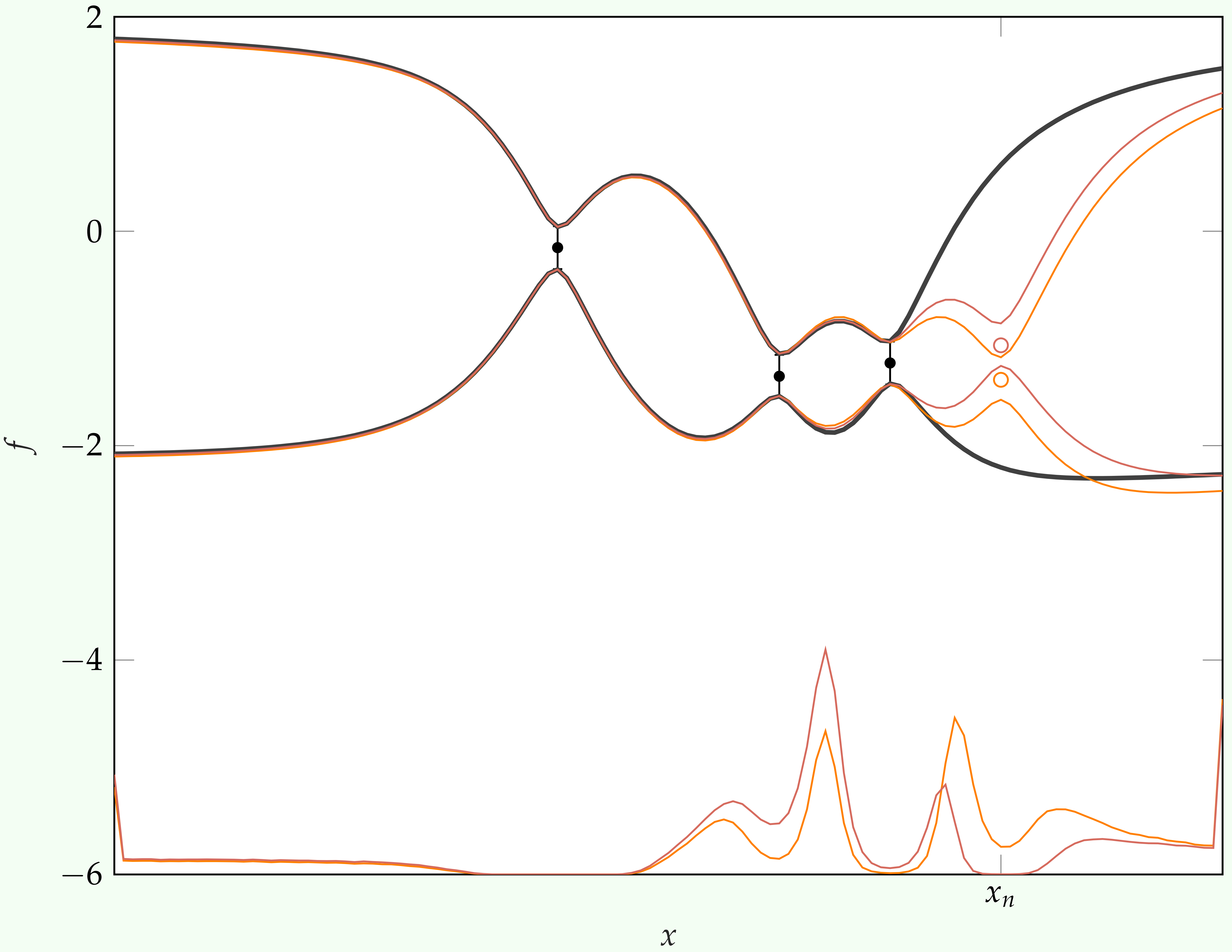
Source: Villemonteix, Vazquez, & Walter (2009); Hennig & Schuler (2012); Hernández-Lobato et al. (2015); Hoffman & Ghahramani (2015); Wang & Jegelka (2017); Ru et al. (2018).



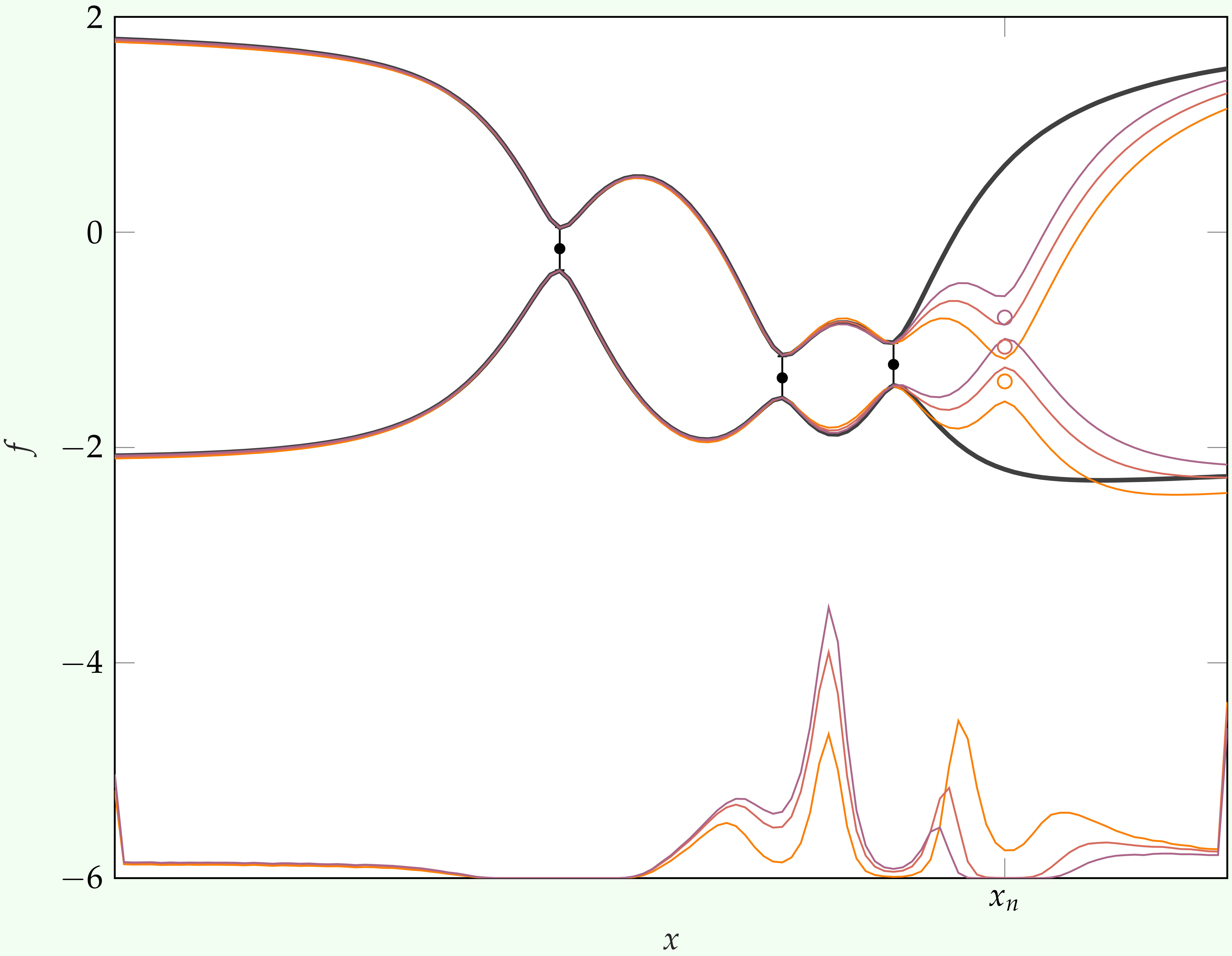
2
1
0
−1
−2
−3
f
−4
−2
0
2
4
x

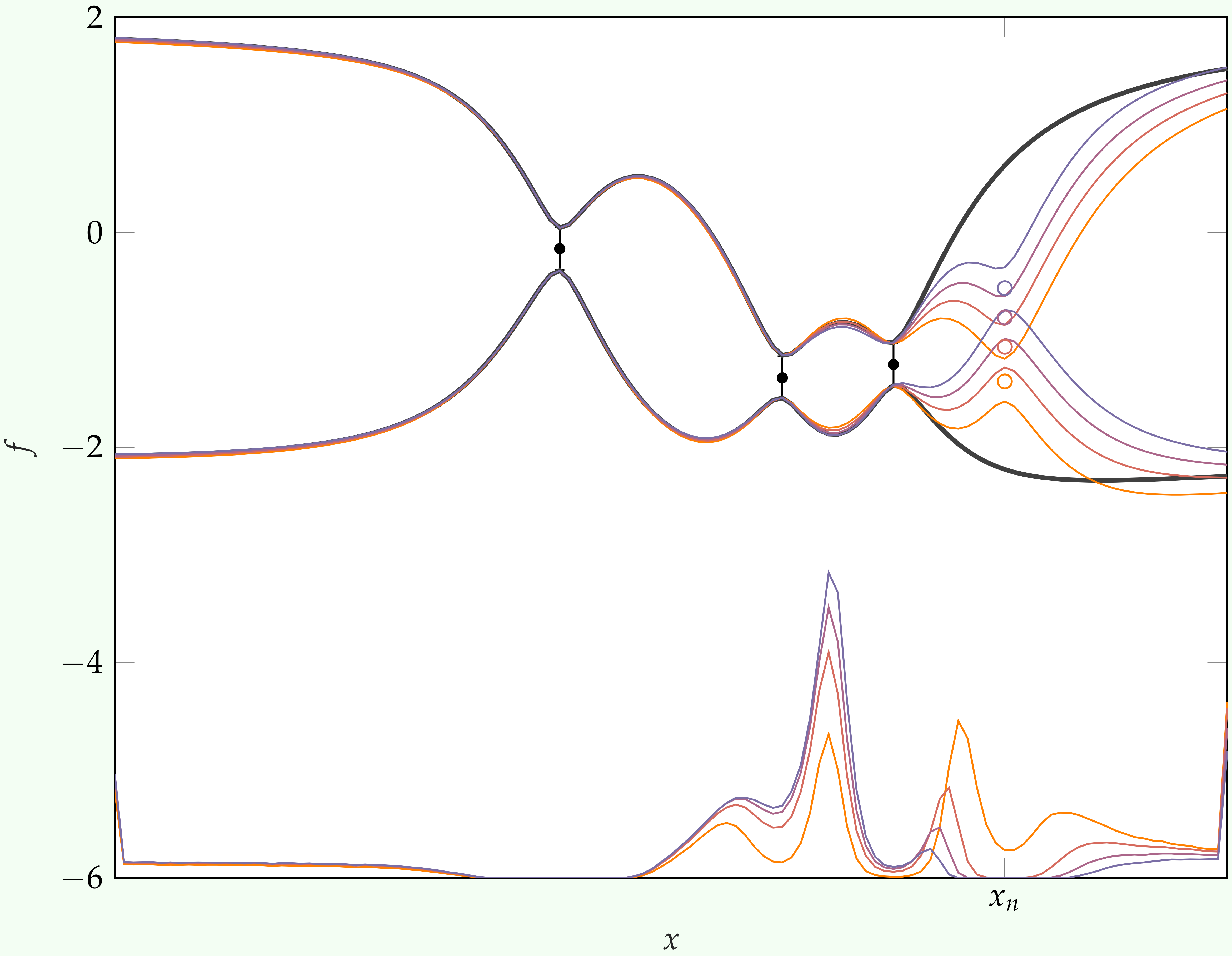

2
0
−2
−4
−6
f
x_n
x


2
0
−2
−4
−6
f
x
x_n

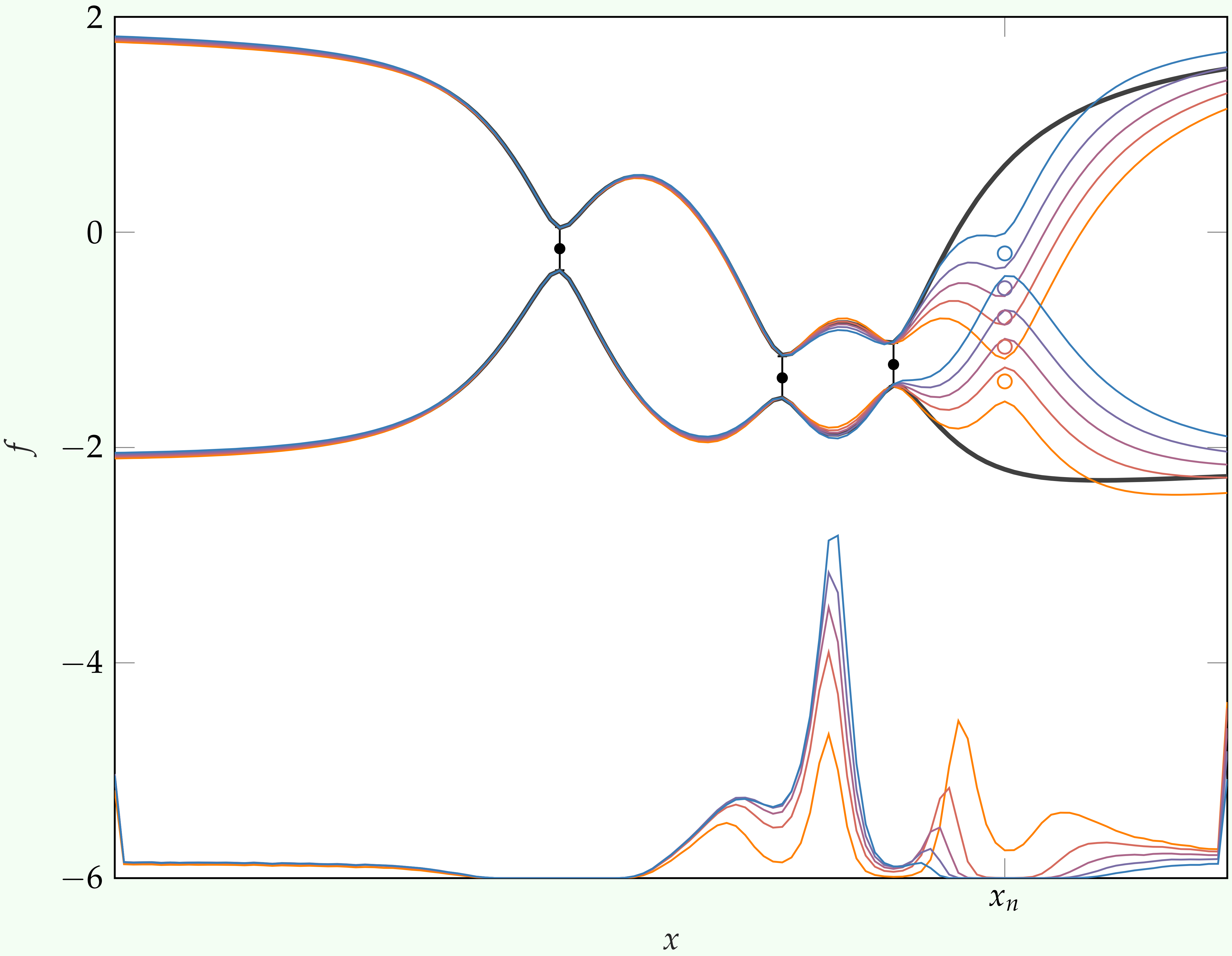


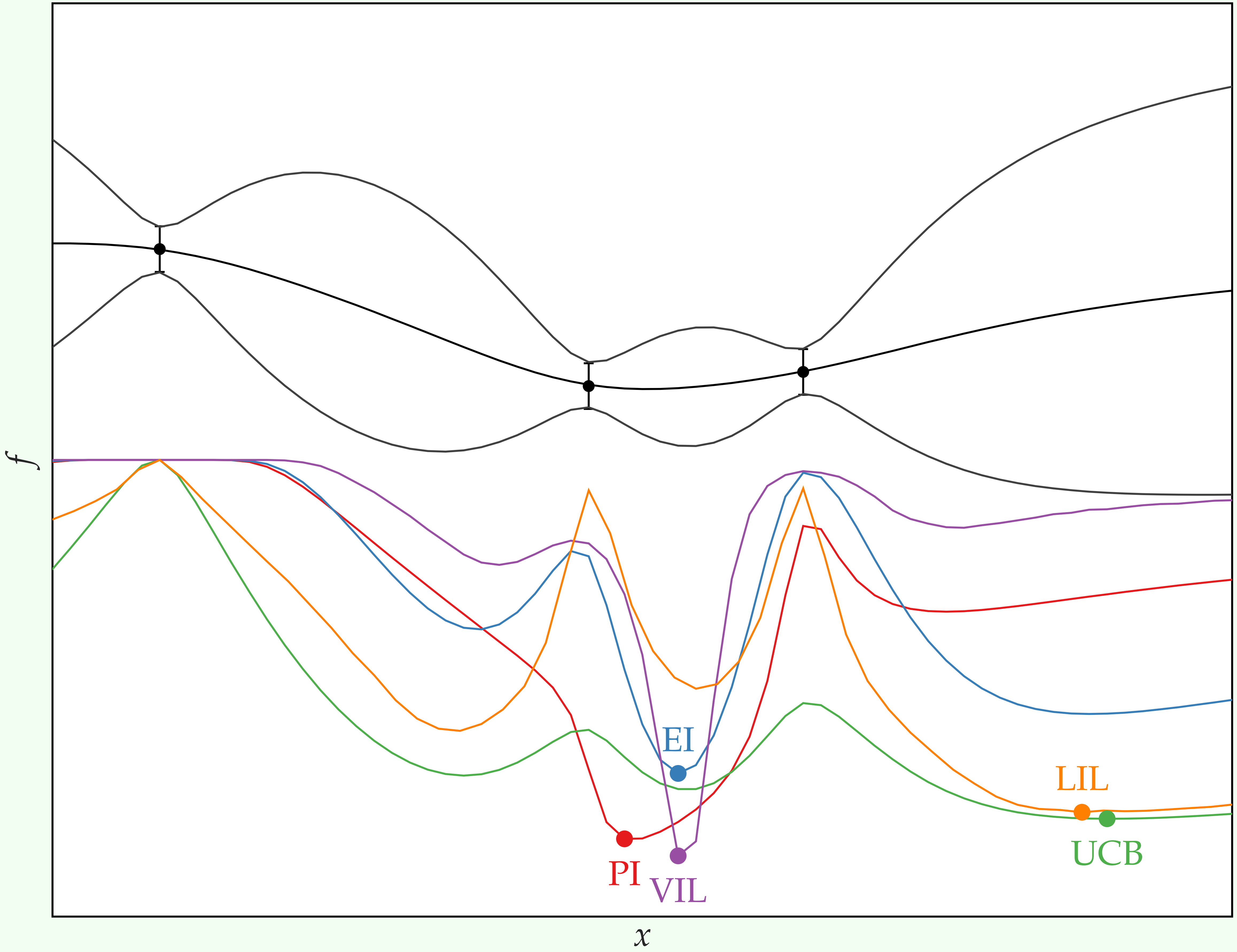
2
0
−2
−4
−6
f
x
x_n


2
0
−2
−4
−6
f
x_n
x



2
0
−2
−4
−6
f
x_n
x


2
0
−2
−4
−6
f
x
x_n

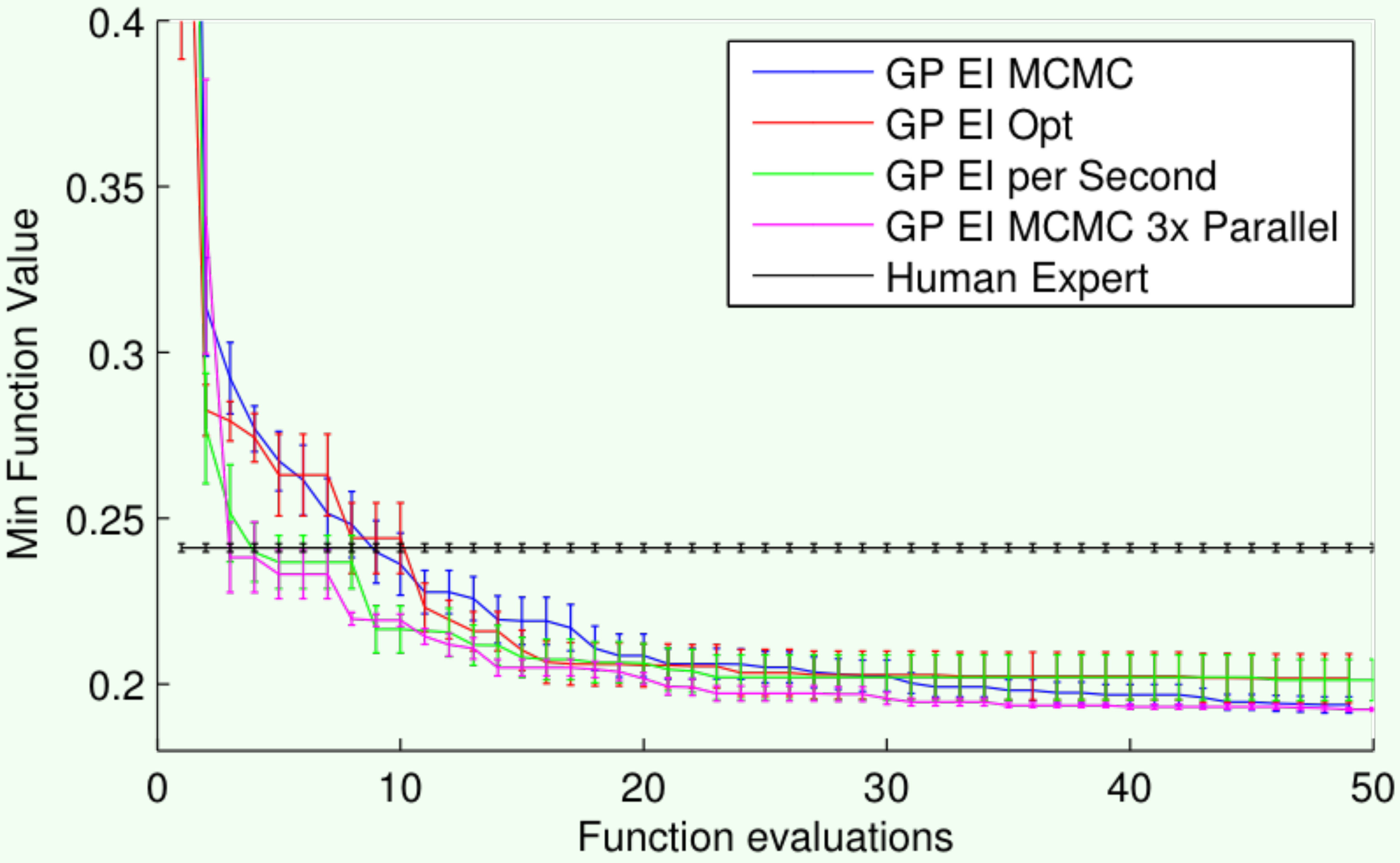

f
x
EI
LIL
PI
VIL
UCB

Bayesian optimisation of hyperparameters is used in AutoML.

Whetlab
We make machine learning better and faster, automatically.

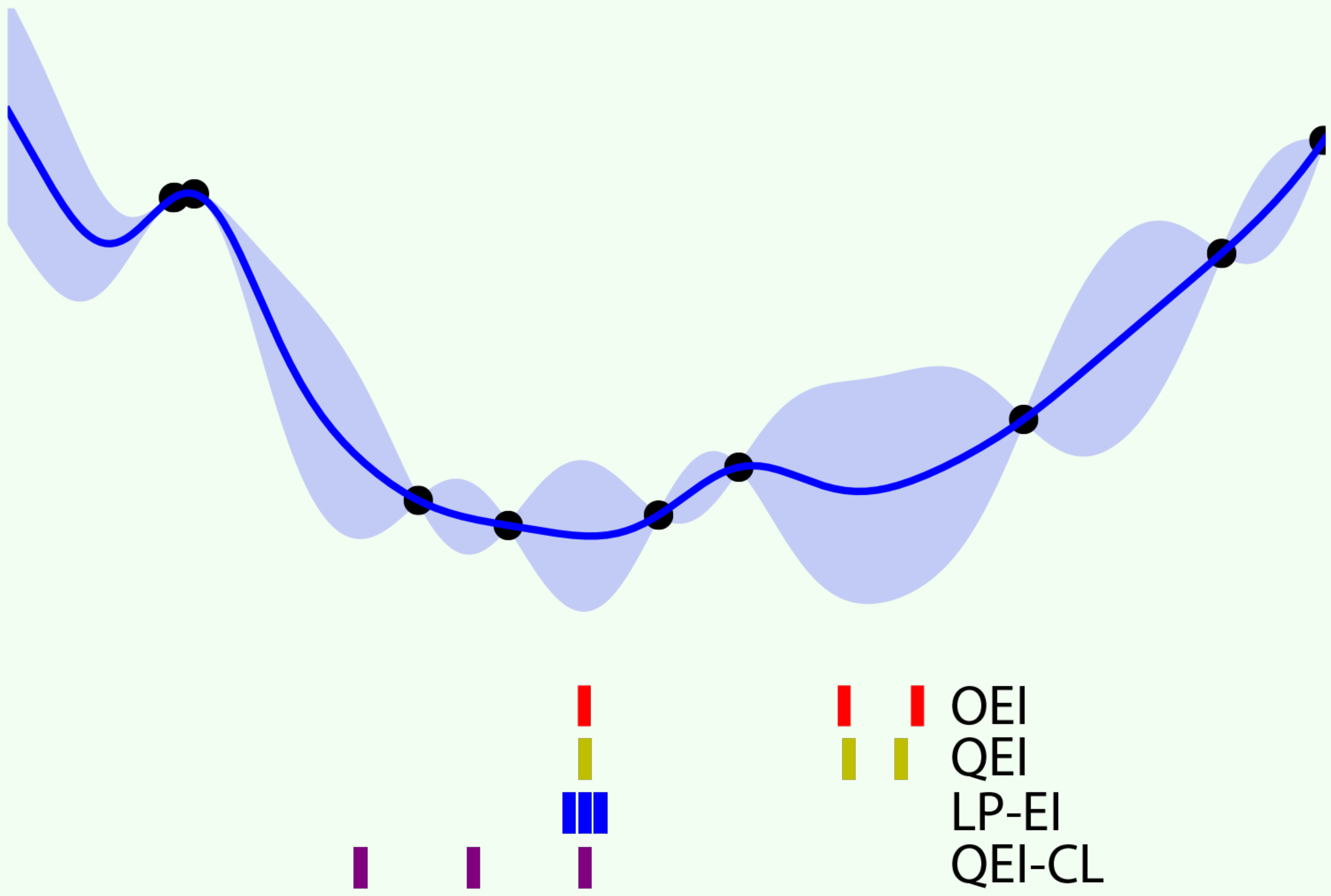
DataRobot

MIND FOUNDRY
Applied Machine Learning



Source: Snoek, Larochelle and Adams (2012), "Practical Bayesian optimization of machine learning algorithms". NIPS.

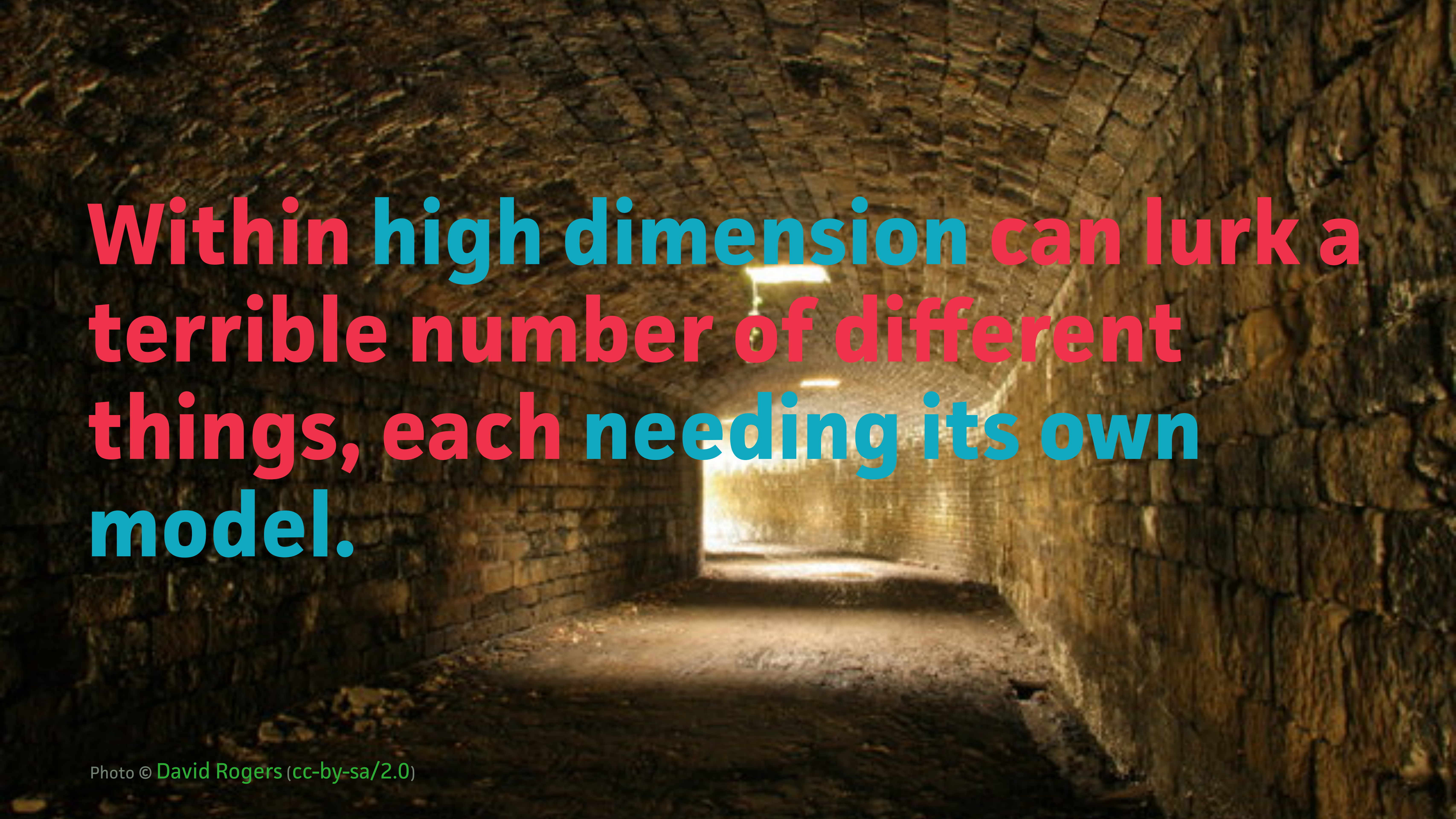
Batch Bayesian optimisation is run in parallel.



Rontsis, Osborne, and Goulart. “Distributionally Robust Optimization Techniques in Batch Bayesian Optimization” (2017).

Hyperparameter optimisation is often treated as a black-box optimisation problem.

It is difficult to imagine a more white-box problem than one where you have full access to the problem's source code.



Within high dimension can lurk a terrible number of different things, each needing its own model.

Hyperparameters should usually be marginalised, not optimised.


Huge thanks to Roman Garnett & Philipp Hennig.