Robust Sample-Efficient Learning in Uncertain Environments

Ilija Bogunovic
Learning and Adaptive Systems Group, ETH Zurich

IfA Seminar, Nov 2019

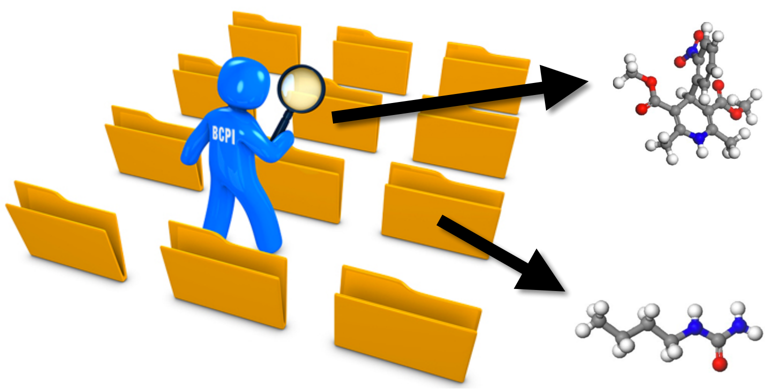
Learning &
Adaptive Systems

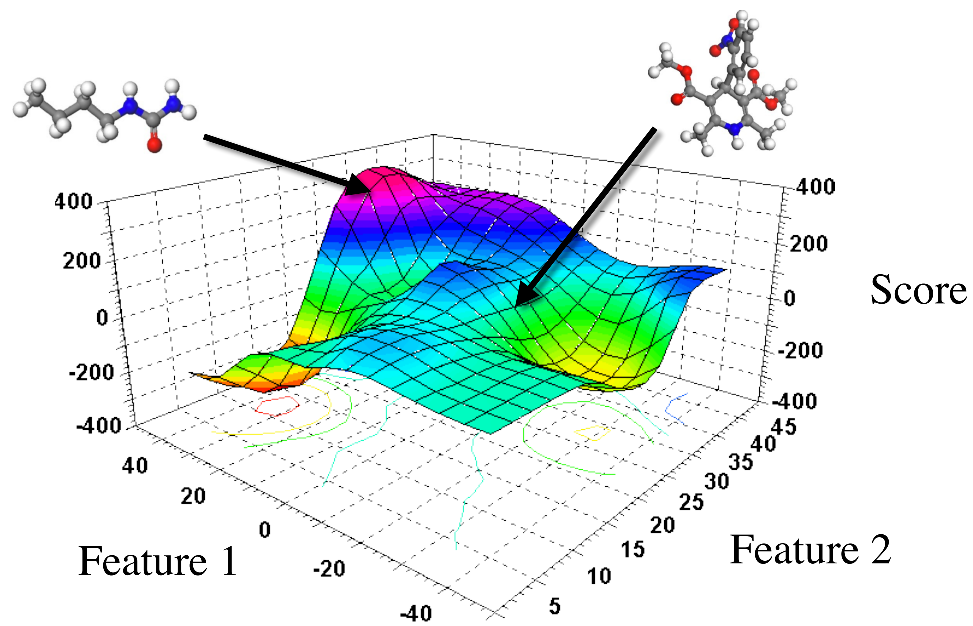
Learning & Adaptive Systems

ETH Zürich

Learning in Uncertain Environments

Expensive unknown function: who doesn't have one?





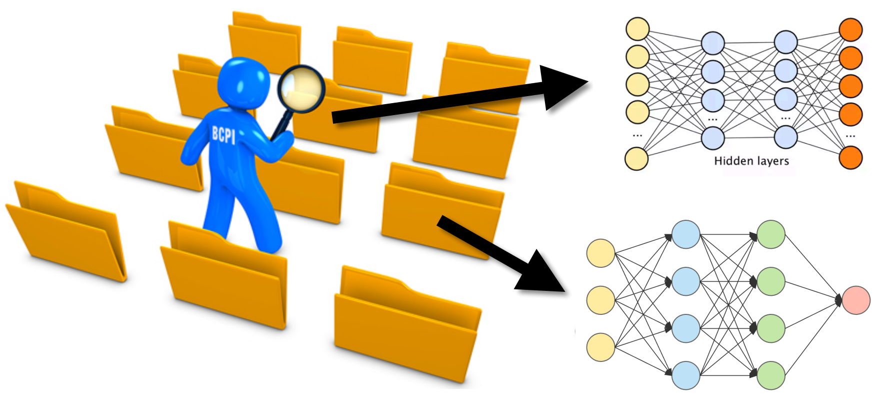
Molecular design

[Romero *el al.*'13,

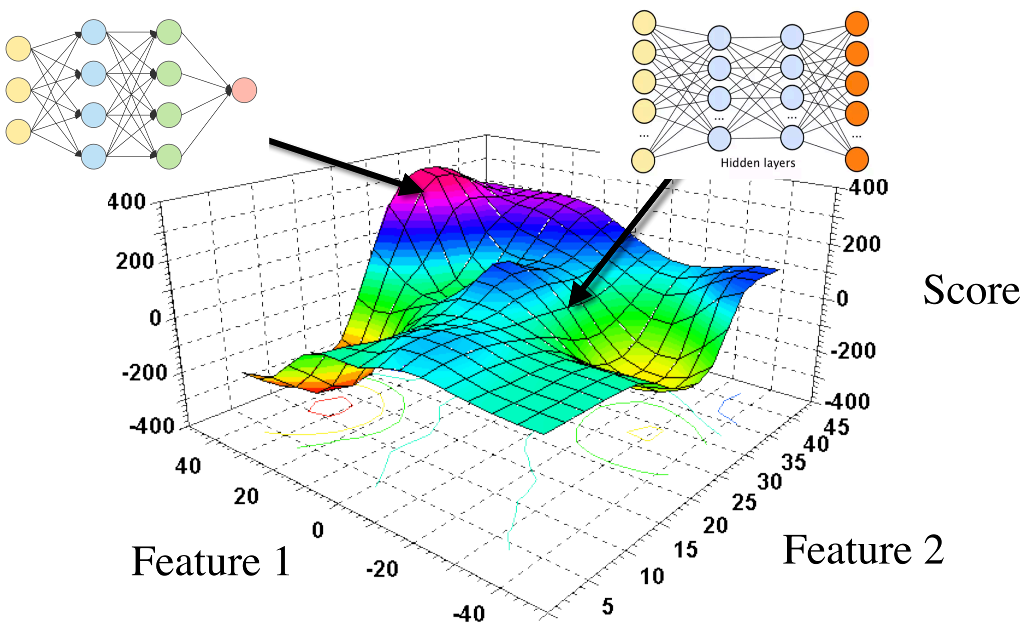
Gomez-Bombarelli *et al.*'17]

Learning in Uncertain Environments

Expensive unknown function: who doesn't have one?



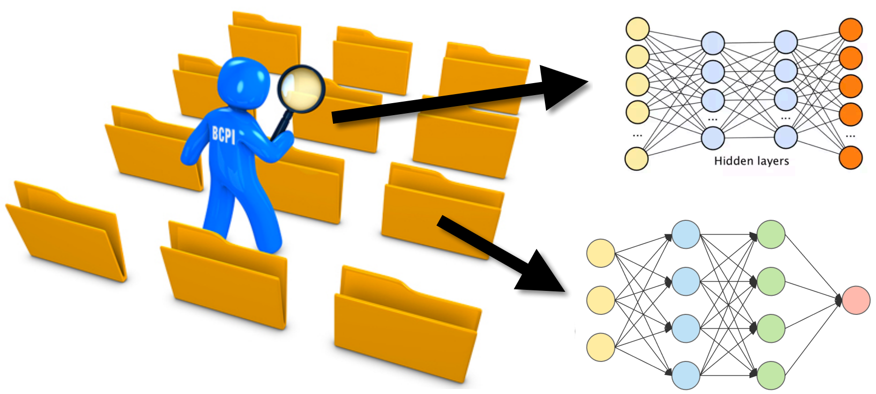
Molecular design
[Romero *el al.*'13,
Gomez-Bombarelli *et al.*'17]

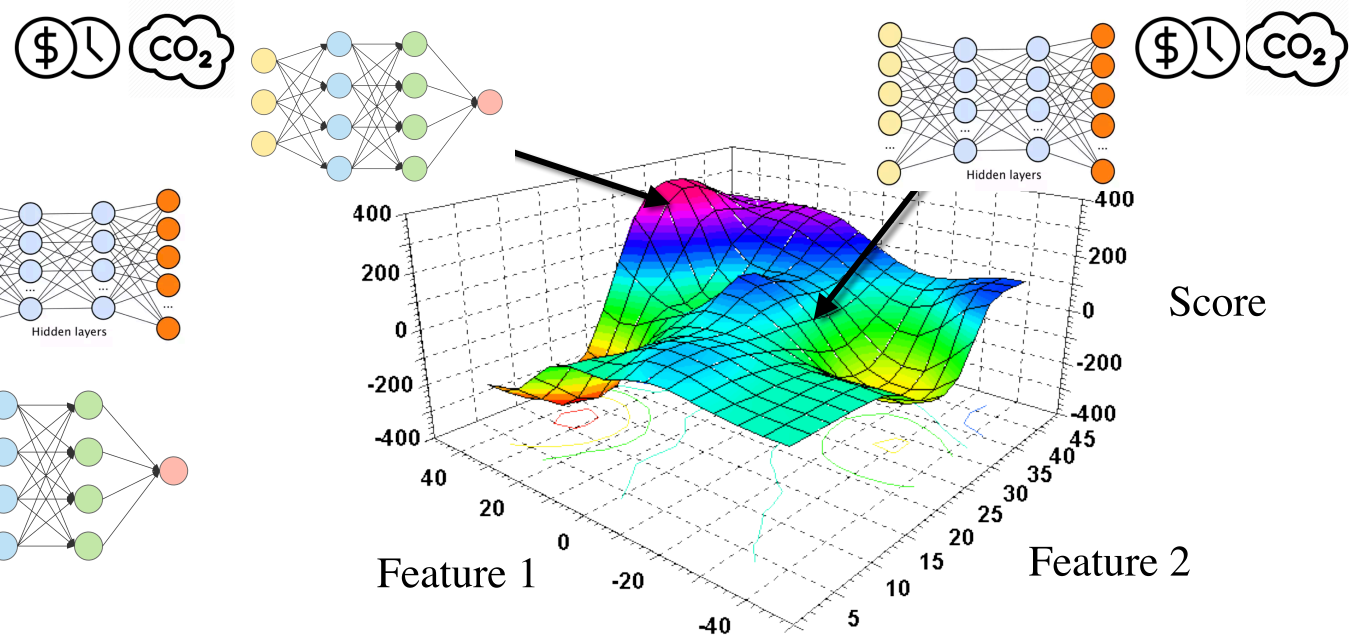


Automatic machine learning
[Snoek *et al.*'12]

Learning in Uncertain Environments

Expensive unknown function: who doesn't have one?



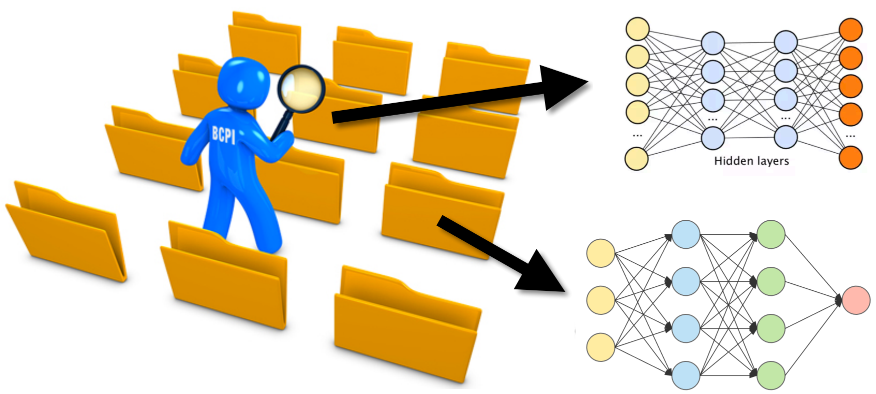


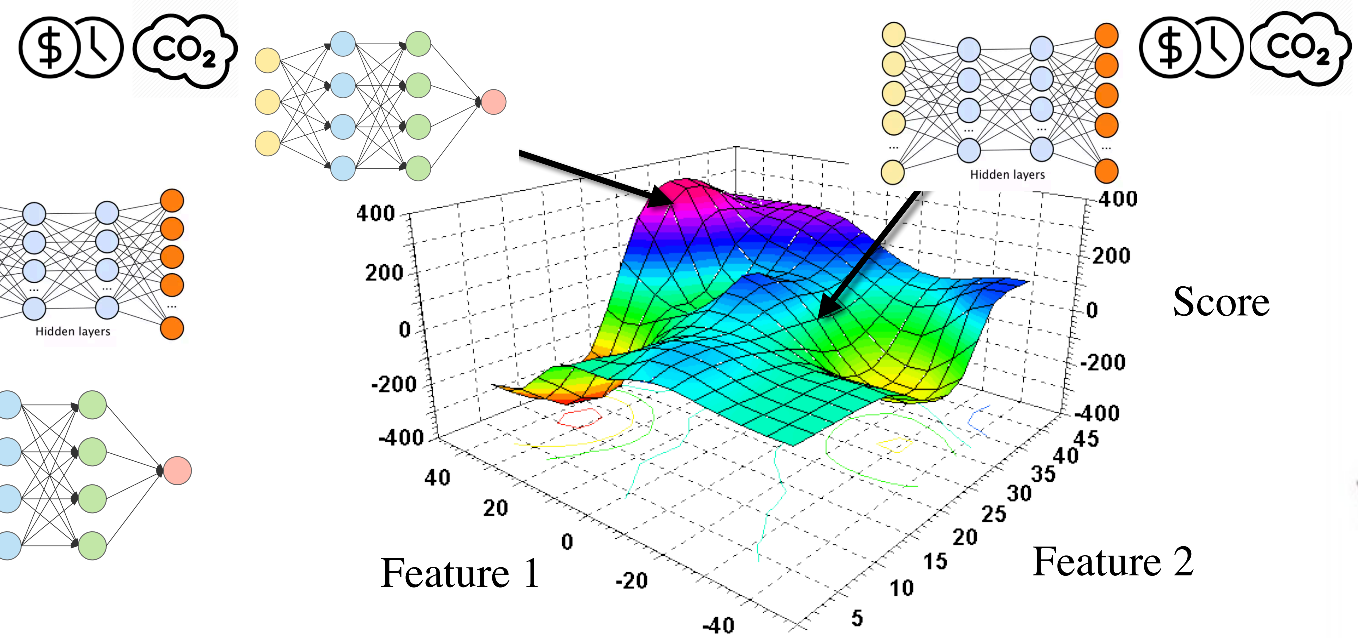
Molecular design
[Romero *el al.*'13,
Gomez-Bombarelli *et al.*'17]

Automatic machine learning
[Snoek *et al.*'12]

Learning in Uncertain Environments

Expensive unknown function: who doesn't have one?

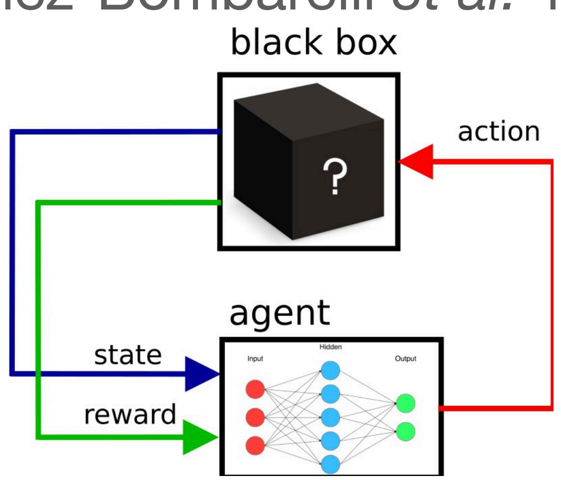




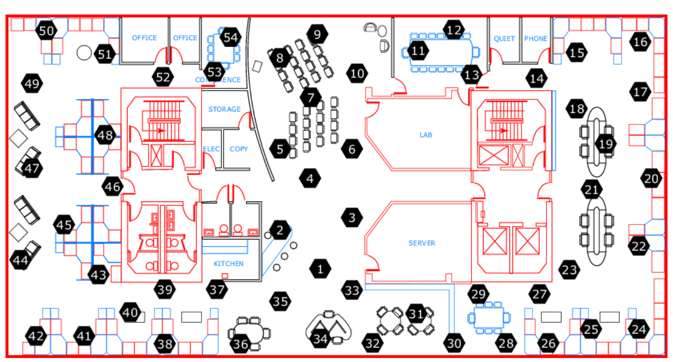
Molecular design
[Romero *el al.*'13, Gomez-Bombarelli *et al.*'17]

Automatic machine learning
[Snoek *et al.*'12]

Recommender systems and advertising
[Vanchinathan *et al.*'14]



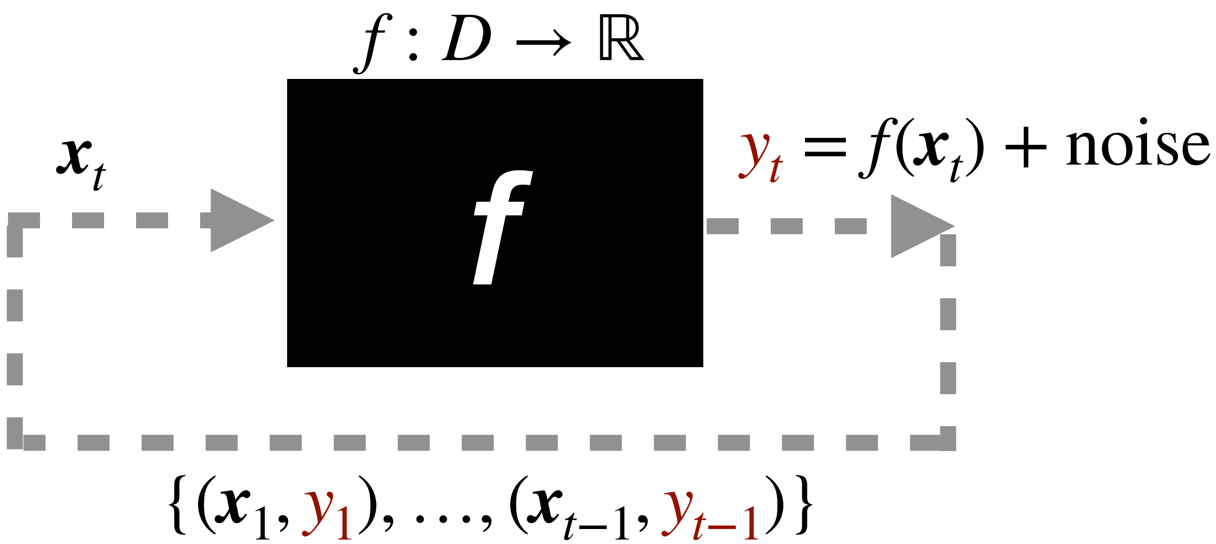
RL and control
[Brochu *et al.*'10]



Sensor nets
[Srinivas *et al.*'11]

AlphaGo
[Chen *et al.*'18]

Setting & Protocol

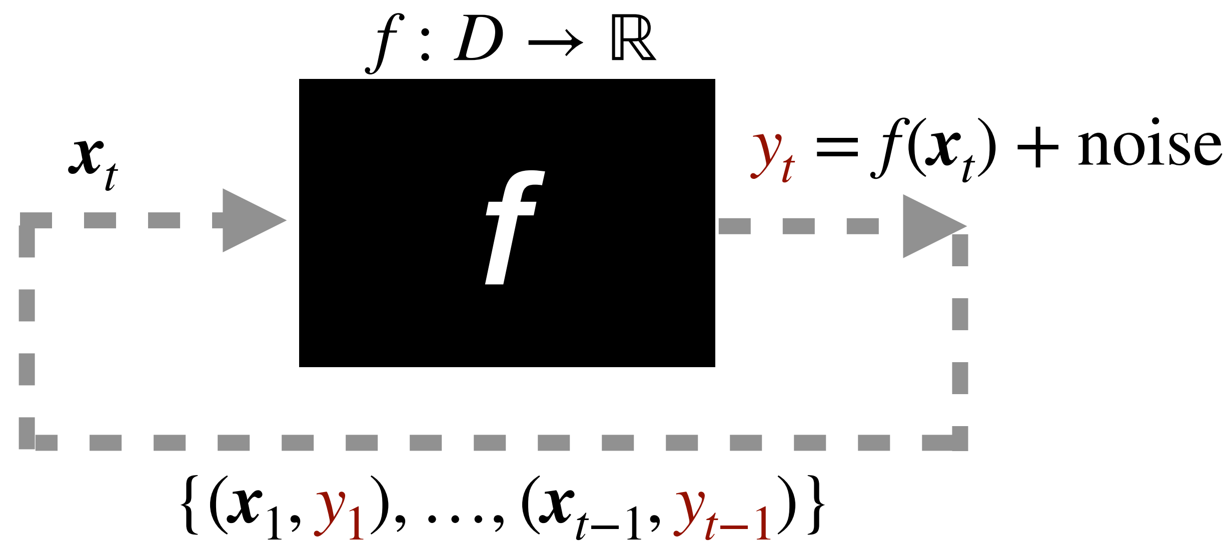


Black-box optimization: Sequentially optimize an unknown function

$$\text{maximize}_{\boldsymbol{x} \in D} \; f(\boldsymbol{x})$$

- **Protocol:** Choose $\boldsymbol{x}_t \in D$ and observe y_t at every t
- **Goal:** Find $\hat{\boldsymbol{x}}$ with low regret: $\max_{\boldsymbol{x} \in D} f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}})$

Setting & Protocol



Black-box optimization: Sequentially optimize an unknown function

$$\text{maximize}_{\boldsymbol{x} \in D} \; f(\boldsymbol{x})$$

- **Protocol:** Choose $\boldsymbol{x}_t \in D$ and observe y_t at every t
- **Goal:** Find $\hat{\boldsymbol{x}}$ with low regret: $\max_{\boldsymbol{x} \in D} f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}})$

Challenges:

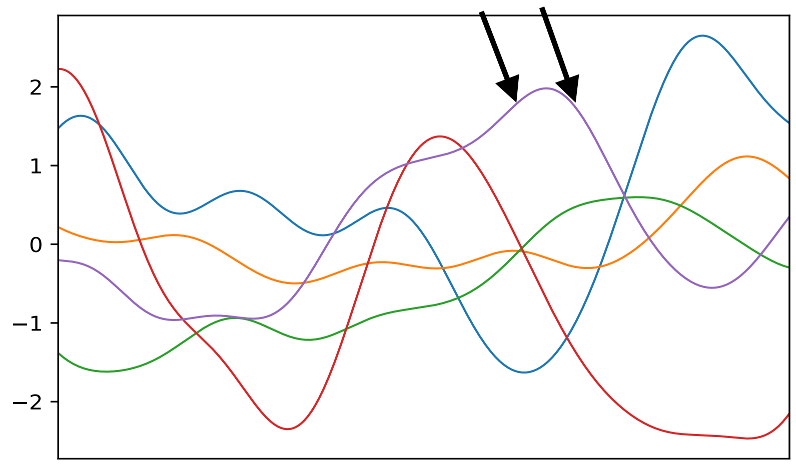
- f is *unknown* (no gradient information) and typically *multi-modal*
- Only *noisy* and *expensive* point evaluations are available

Gaussian Process Model

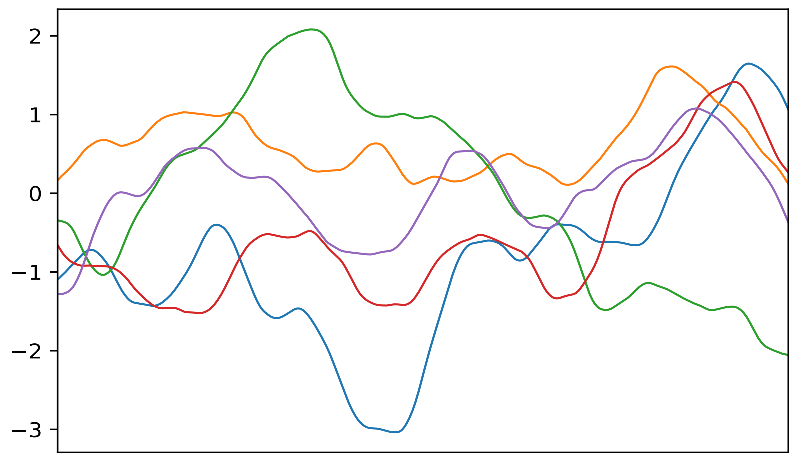
Smoothness: Gaussian processes, $f \sim \mathrm{GP}(\mathbf{0}, k(\cdot,\cdot))$

- Bayesian prior on the underlying $f(\cdot)$
- Smoothness properties encoded through **kernel** $k(\boldsymbol{x}, \boldsymbol{x}')$ (covariance function)
- Any finite set of function values $\{f(\boldsymbol{x}_i)\}_{i=1}^n$ is jointly Gaussian

Cov$[f(\boldsymbol{x}), f(\boldsymbol{x}')] = k(\boldsymbol{x}, \boldsymbol{x}')$



$k_{\mathrm{SE}}(\boldsymbol{x}, \boldsymbol{x}') = \exp\big(-(2l)^{-2}\|\boldsymbol{x}-\boldsymbol{x}'\|^2\big)$



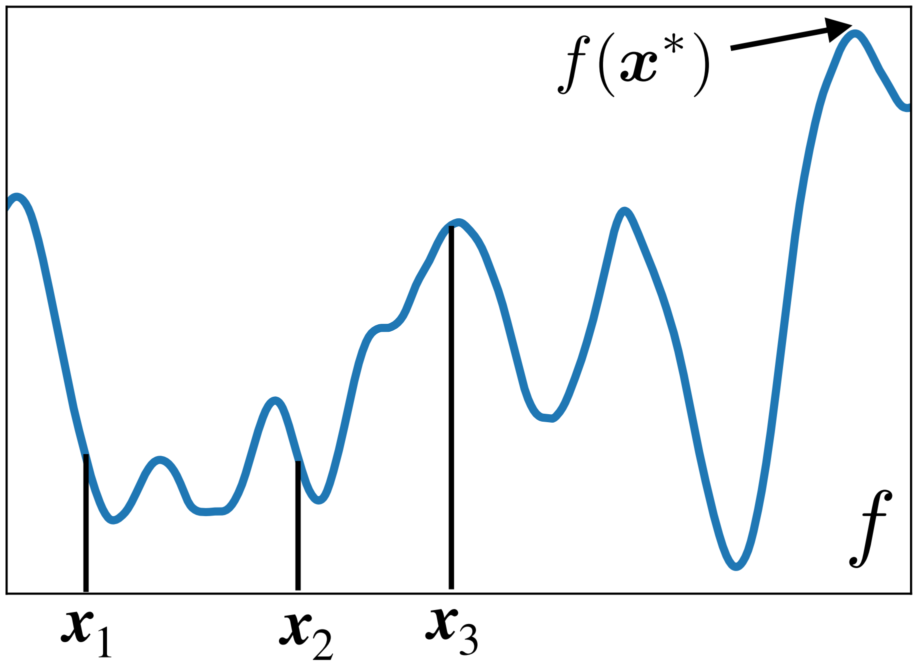
$k_{\mathrm{Mat}}(\boldsymbol{x}, \boldsymbol{x}')$

Gaussian Process Model

Smoothness: Gaussian processes, $f \sim \mathrm{GP}(\mathbf{0}, k(\cdot,\cdot))$

- Bayesian prior on the underlying $f(\cdot)$
- Smoothness properties encoded through **kernel** $k(\boldsymbol{x}, \boldsymbol{x}')$ (covariance function)
- Any finite set of function values $\{f(\boldsymbol{x}_i)\}_{i=1}^{n}$ is jointly Gaussian

$f \sim \mathrm{GP}(\mu(\cdot), k(\cdot,\cdot))$

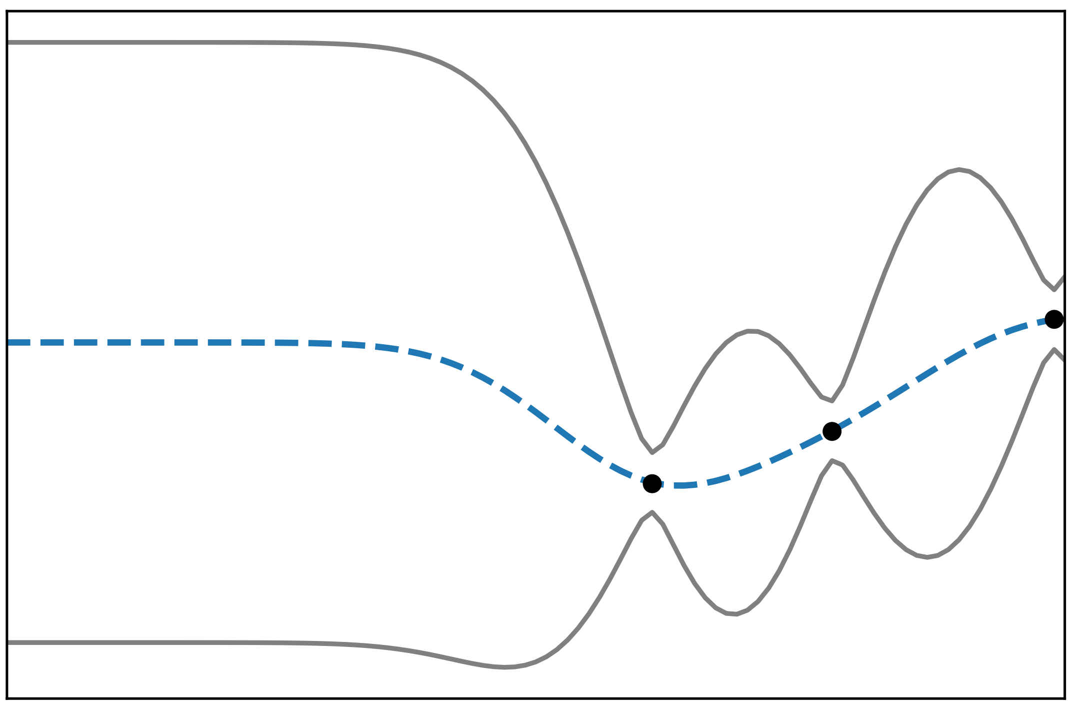


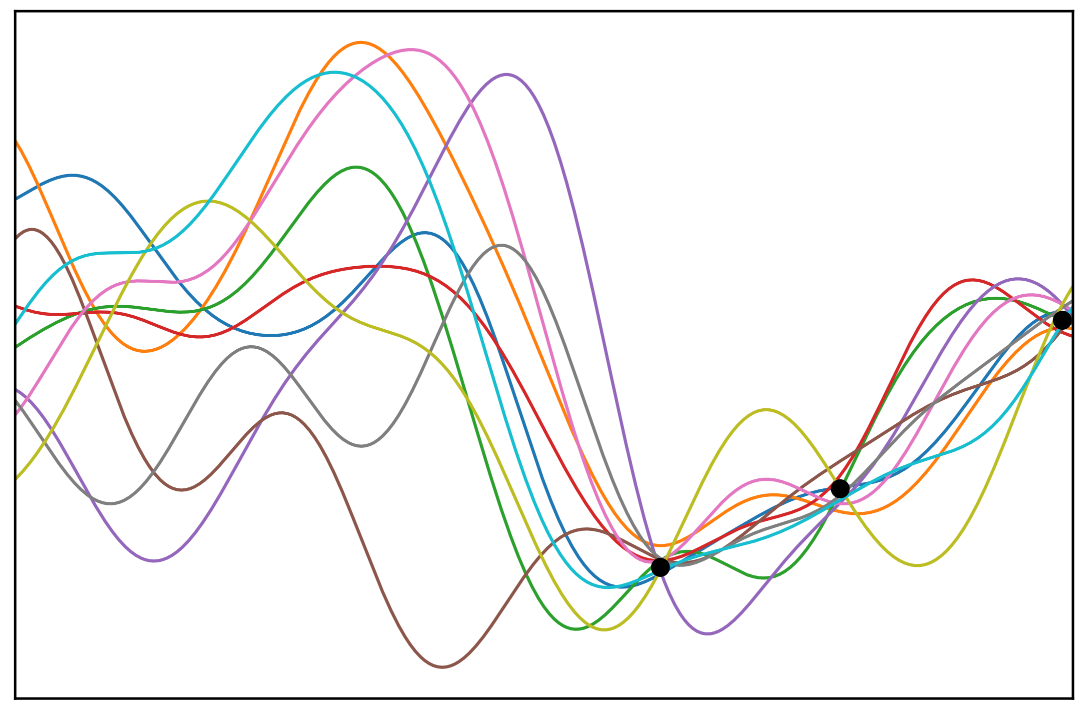
$$\begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ f(\boldsymbol{x}_3) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\boldsymbol{x}_1) \\ \mu(\boldsymbol{x}_2) \\ \mu(\boldsymbol{x}_3) \end{bmatrix}, \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & k(\boldsymbol{x}_1,\boldsymbol{x}_2) & k(\boldsymbol{x}_1,\boldsymbol{x}_3) \\ k(\boldsymbol{x}_2,\boldsymbol{x}_1) & k(\boldsymbol{x}_2,\boldsymbol{x}_2) & k(\boldsymbol{x}_2,\boldsymbol{x}_3) \\ k(\boldsymbol{x}_3,\boldsymbol{x}_1) & k(\boldsymbol{x}_3,\boldsymbol{x}_2) & k(\boldsymbol{x}_3,\boldsymbol{x}_3) \end{bmatrix} \right)$$

Gaussian Process Model

Model Learning: Gaussian processes, $\mathrm{GP}(\mathbf{0}, k(\cdot, \cdot))$

- **Closed form** posterior updates given previous data

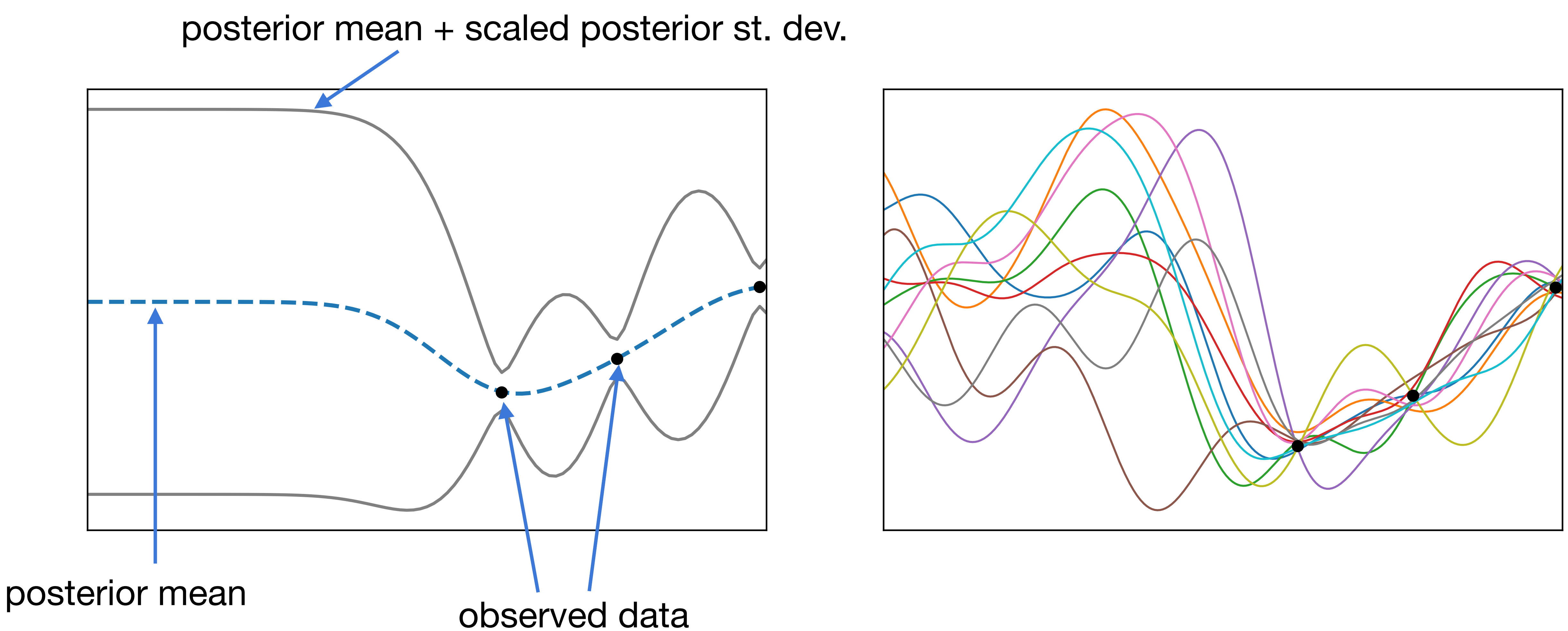




Gaussian Process Model

Model Learning: Gaussian processes, $\mathrm{GP}(\mathbf{0}, k(\cdot,\cdot))$

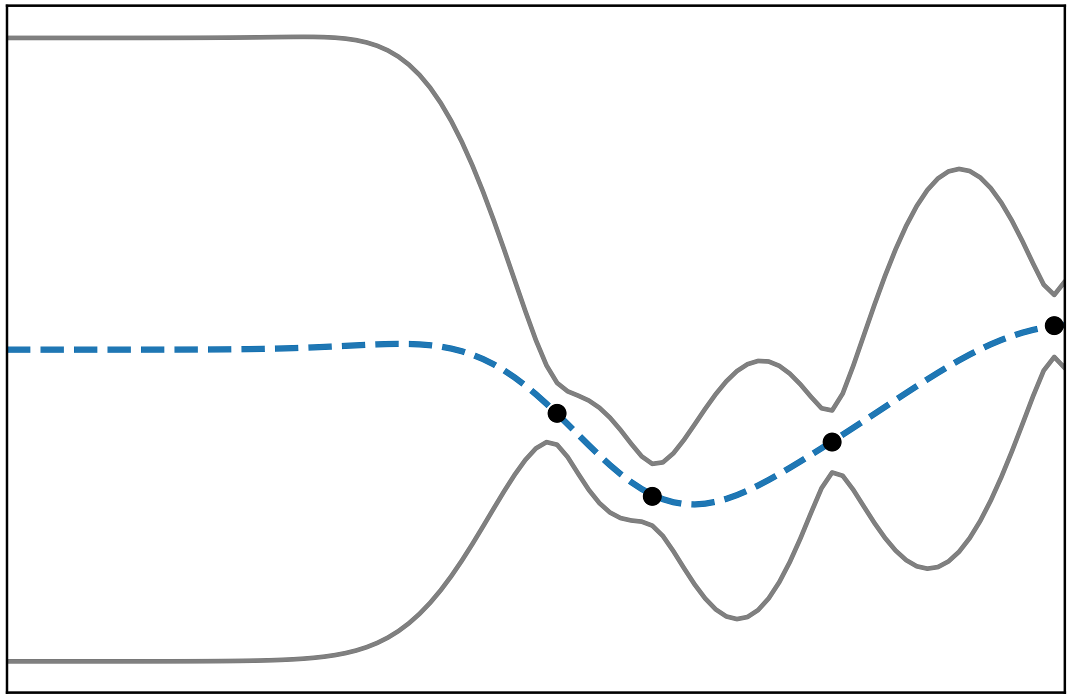
- **Closed form** posterior updates given previous data

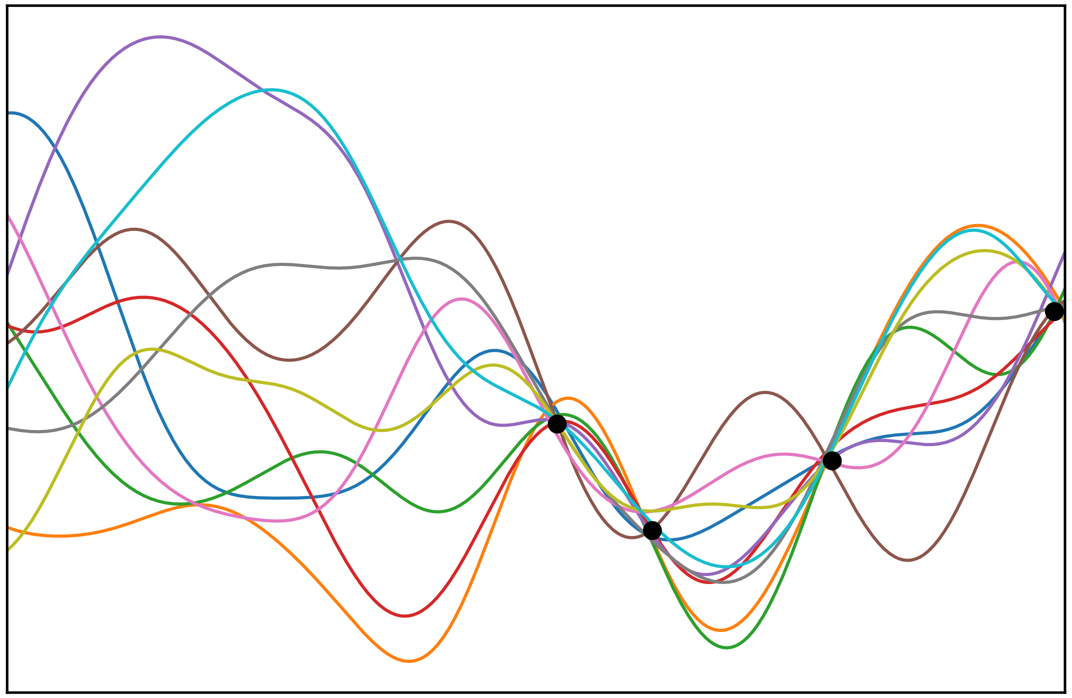


Gaussian Process Model

Model Learning: Gaussian processes, $\mathrm{GP}(\mathbf{0}, k(\cdot\,,\cdot\,))$

- **Closed form** posterior updates given previous data

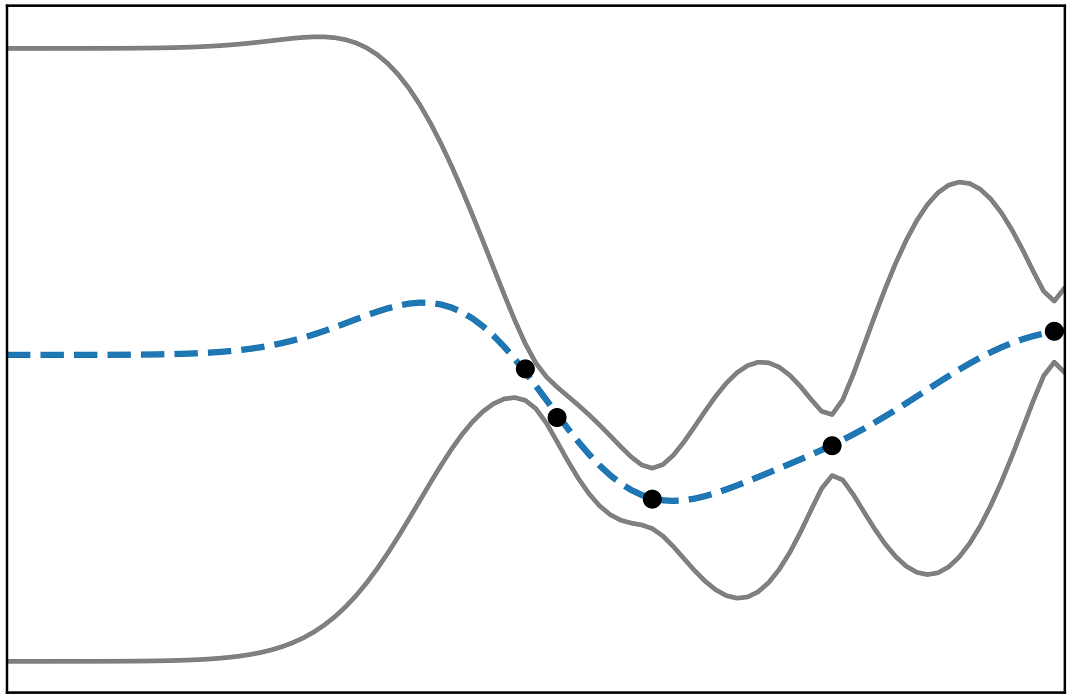


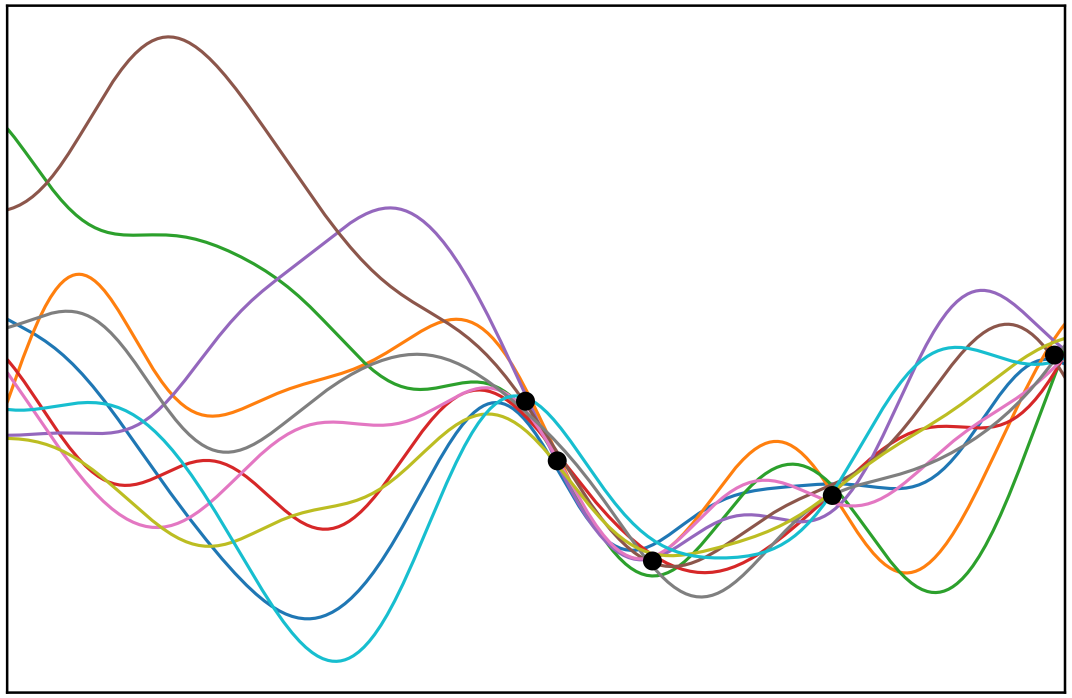


Gaussian Process Model

Model Learning: Gaussian processes, $\mathrm{GP}(\mathbf{0}, k(\cdot, \cdot))$

- **Closed form** posterior updates given previous data

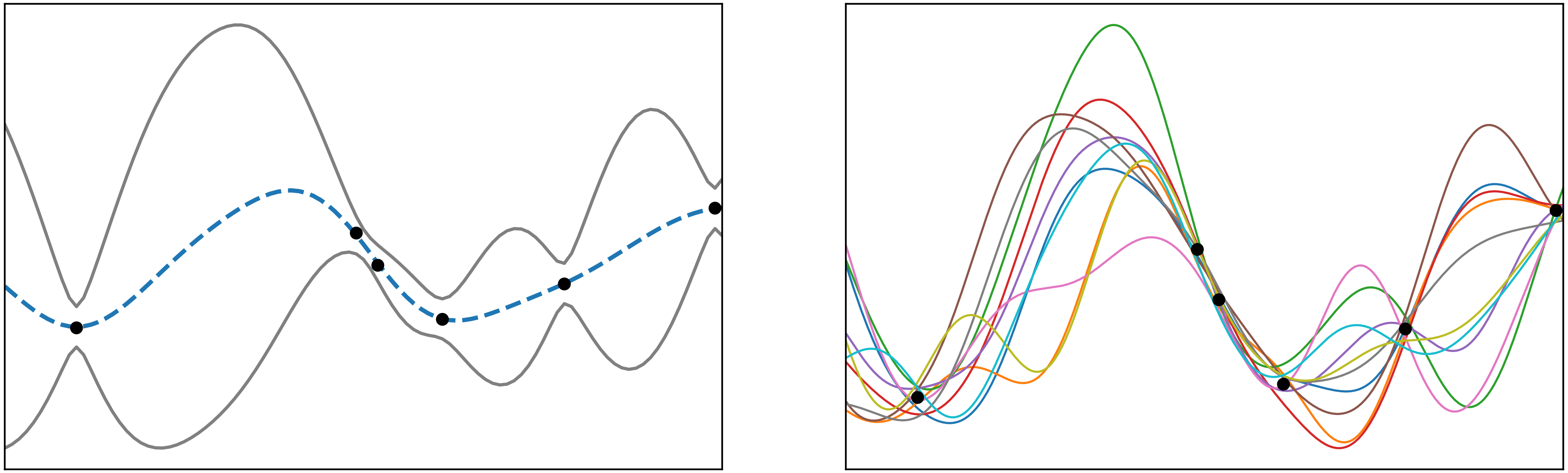




Gaussian Process Model

Model Learning: Gaussian processes, $\mathrm{GP}(\mathbf{0}, k(\cdot, \cdot))$

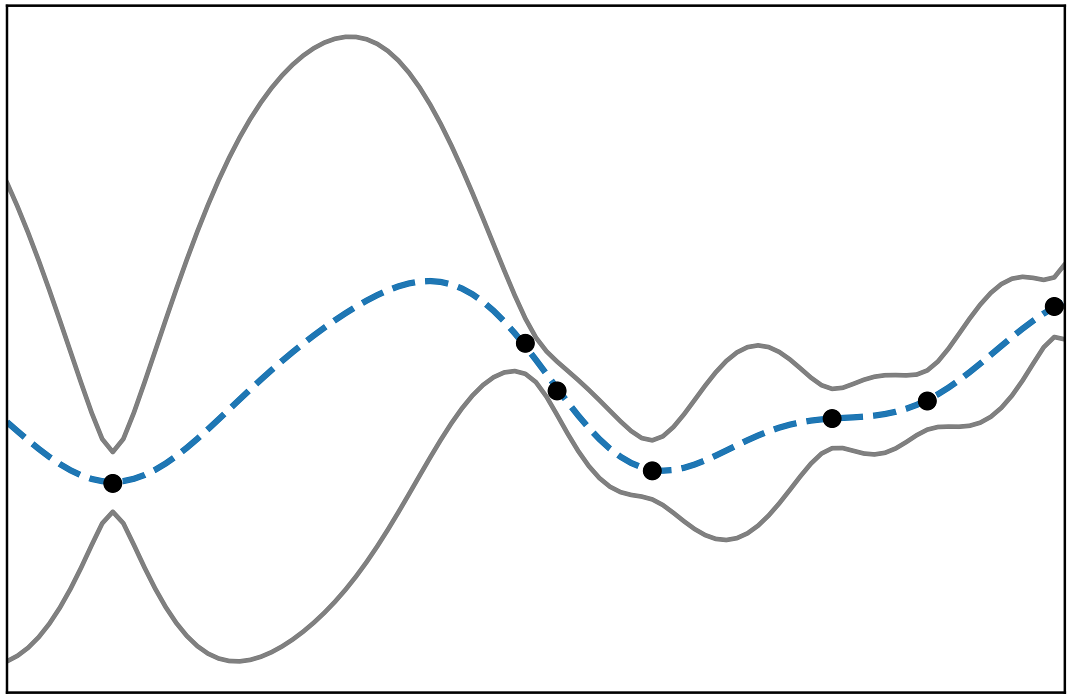
- **Closed form** posterior updates given previous data

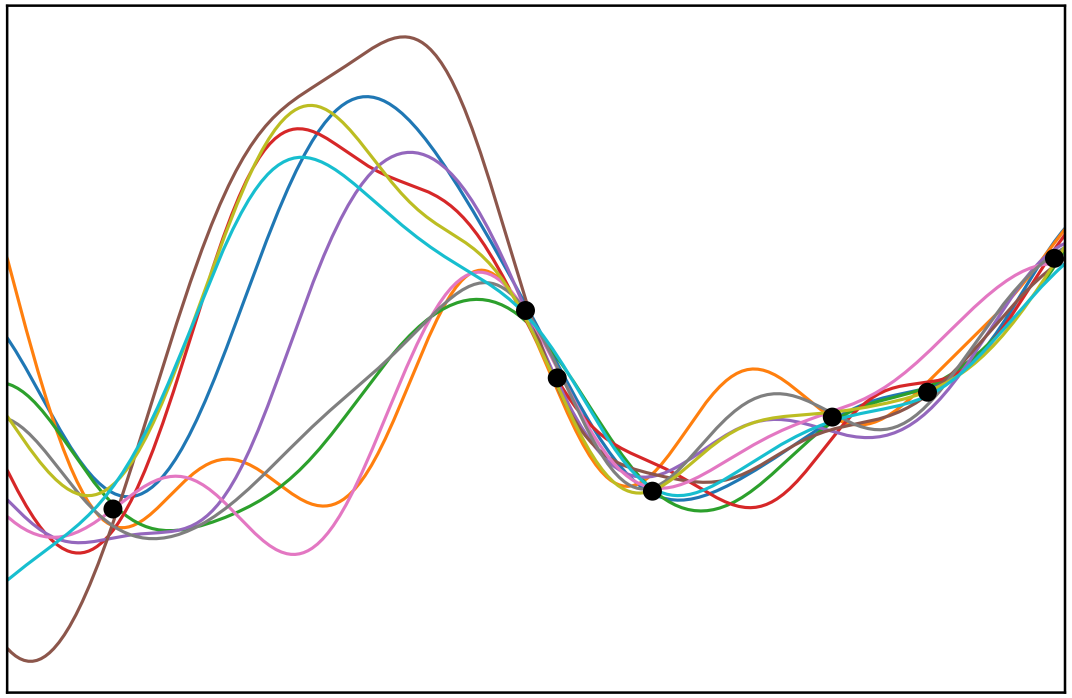


Gaussian Process Model

Model Learning: Gaussian processes, $\mathrm{GP}(\mathbf{0}, k(\cdot\,,\cdot\,))$

- **Closed form** posterior updates given previous data

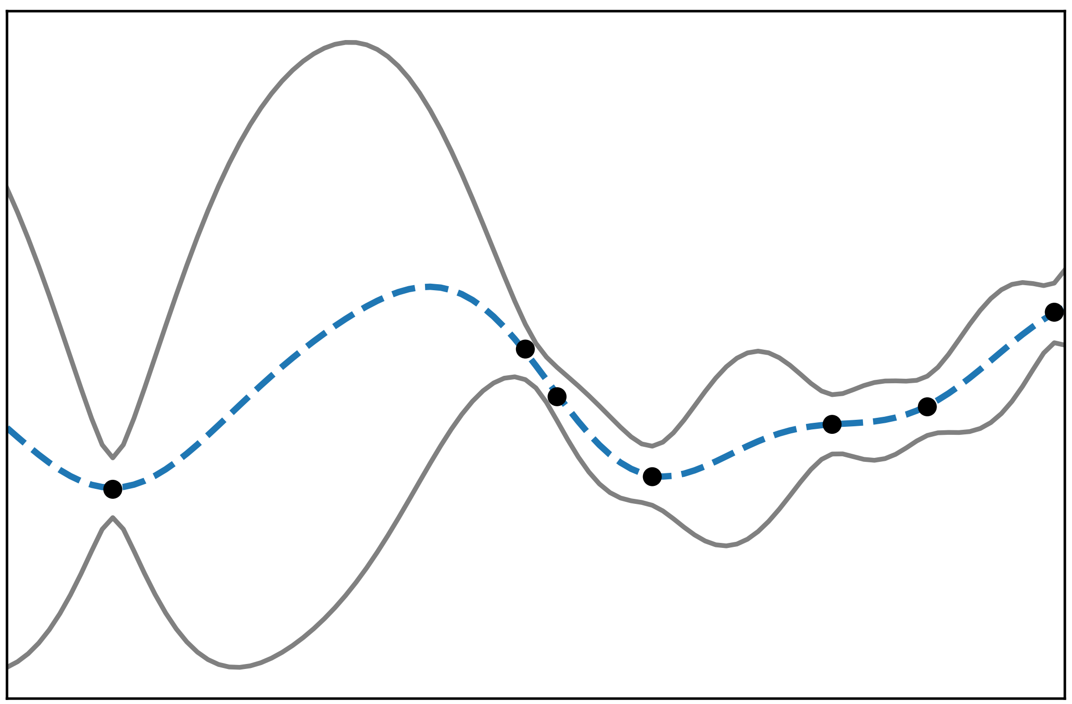


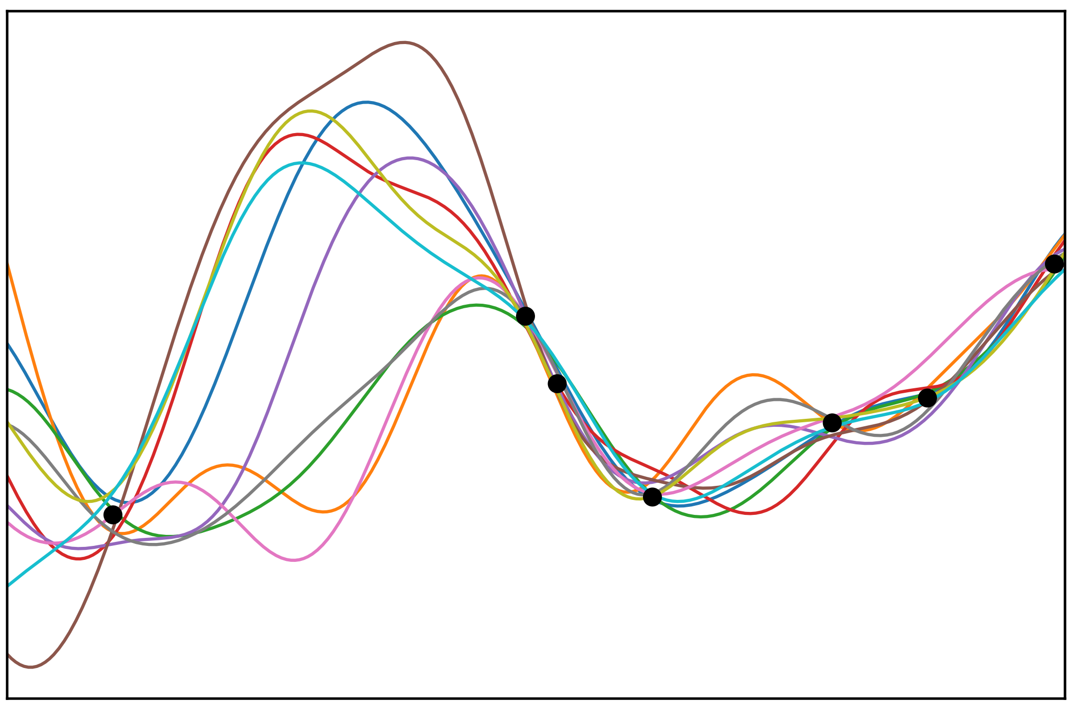


Gaussian Process Model

Model Learning: Gaussian processes, $\mathrm{GP}(\mathbf{0}, k(\cdot\,,\cdot\,))$

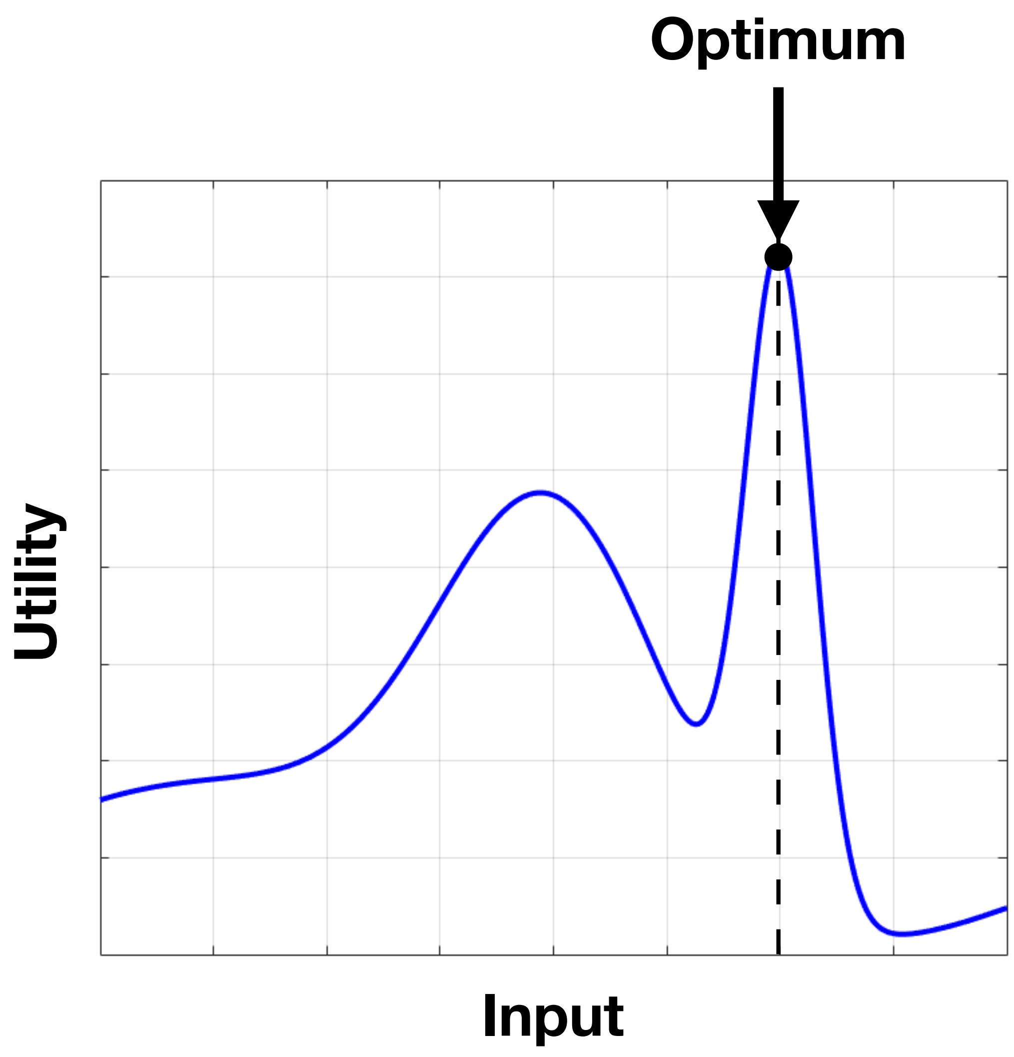
- **Closed form** posterior updates given previous data





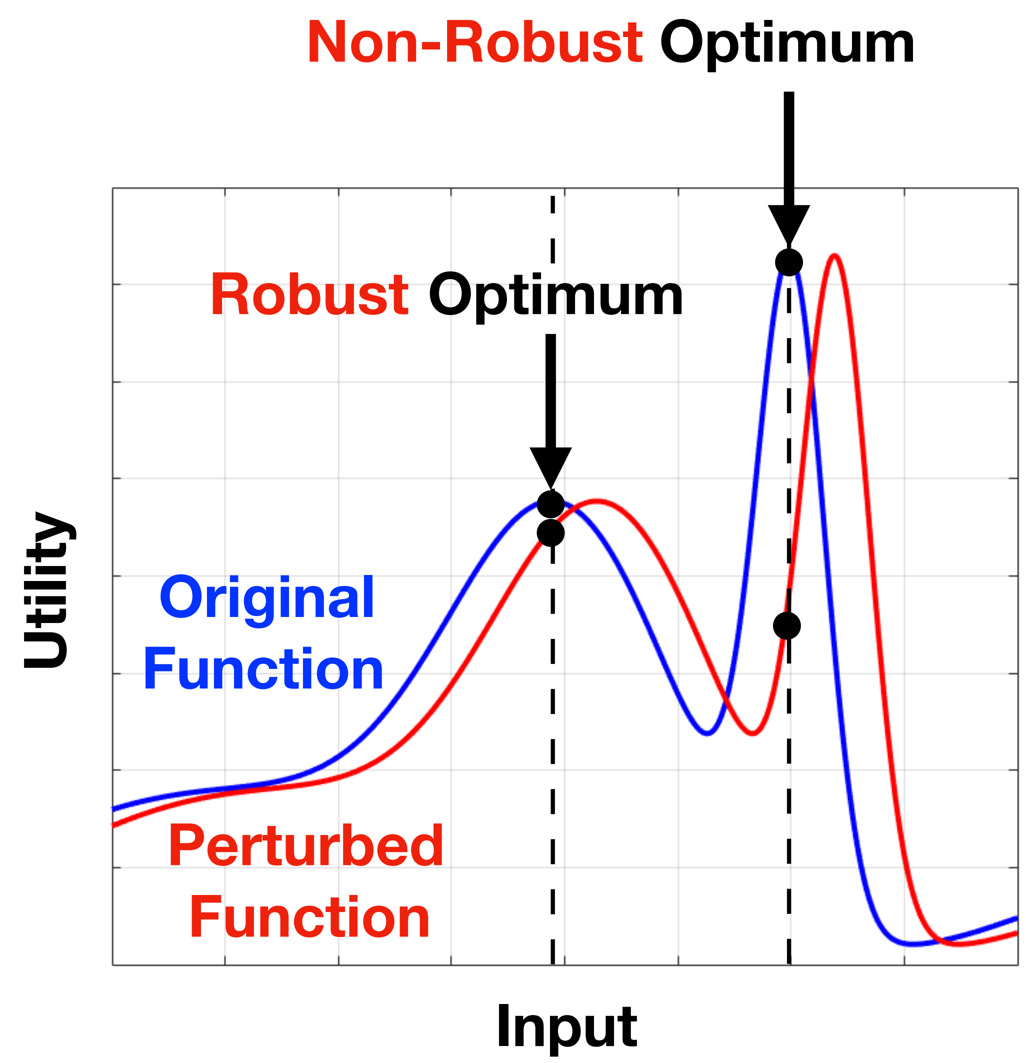
Robust Learning in Uncertain Environments

Example:



Robust Learning in Uncertain Environments

Example:

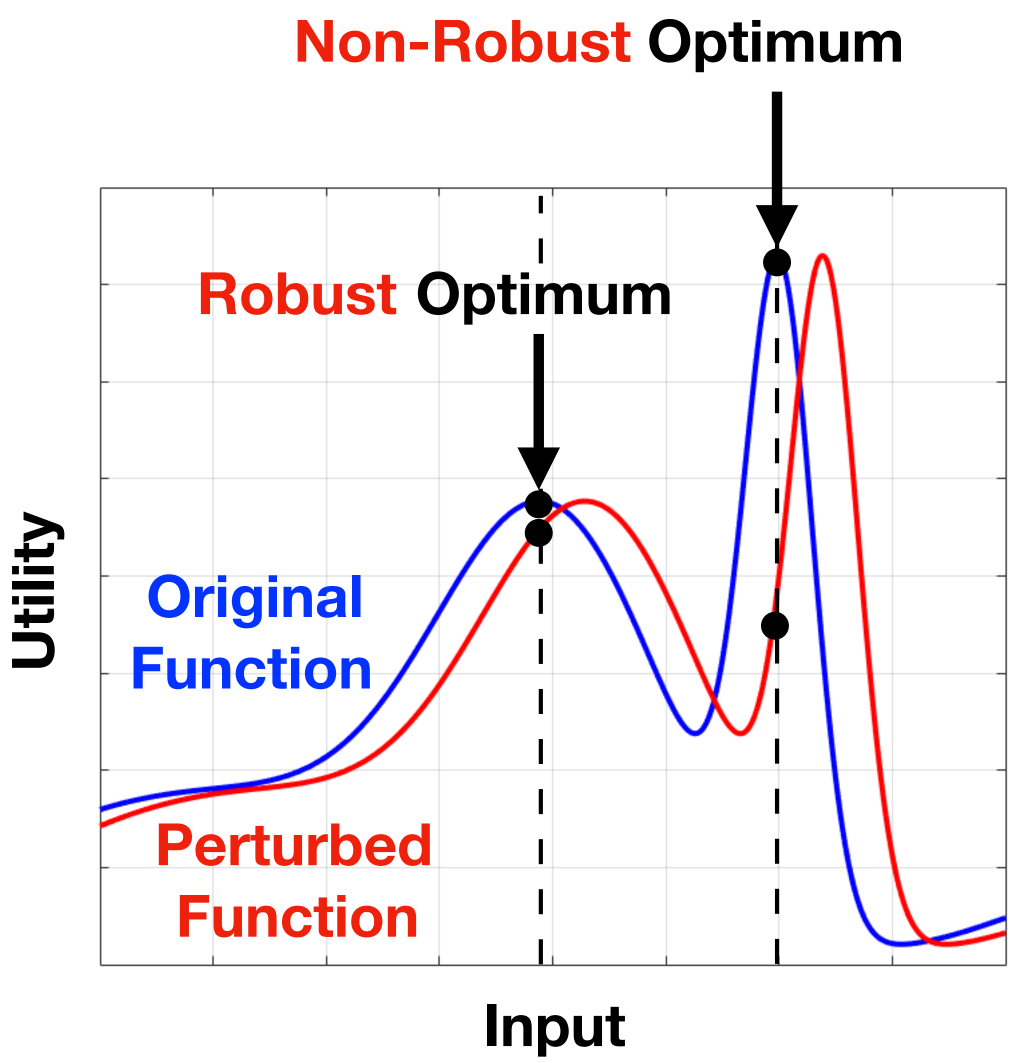


Robust Learning in Uncertain Environments

Robustness requirements:

- Changing environments
- Implementation errors
- Adversarial perturbations
- Training vs. test error in parameter tuning
- Simulator vs. physical world execution
- Model mismatch
- Corrupted data
- Competition in unknown games
- …

Example:



Problem Statement

Model: Assume Gaussian process model for some (known) kernel $k(\boldsymbol{x}, \boldsymbol{x}')$

Optimization goal: $\text{maximize}_{\boldsymbol{x} \in D} \, f(\boldsymbol{x})$

Procedure: At time *t*

1. Choose $\boldsymbol{x}_t$ and observe noisy sample

$$y_t = f(\boldsymbol{x}_t) + z_t, \quad z_t \sim \mathcal{N}(0, \sigma^2)$$

2. Update the GP posterior model with new observation

After T rounds, report final estimate $\hat{\boldsymbol{x}}_T$

Problem Statement

Model: Assume Gaussian process model for some (known) kernel $k(\boldsymbol{x}, \boldsymbol{x}')$

Optimization goal: $\text{maximize}_{\boldsymbol{x} \in D} f(\boldsymbol{x})$

Procedure: At time t

1. Choose $\boldsymbol{x}_t$ and observe noisy sample

$$y_t = f(\boldsymbol{x}_t) + z_t, \quad z_t \sim \mathcal{N}(0, \sigma^2)$$

2. Update the GP posterior model with new observation

After T rounds, report final estimate $\hat{\boldsymbol{x}}_T$

Also, corrupted observations:

$$y_t = f(\boldsymbol{x}_t) + z_t + a_t(\boldsymbol{x}_t)$$

[I.B., A. Krause, J. Scarlett (2019)]

Problem Statement

Model: Assume Gaussian process model for some (known) kernel $k(\boldsymbol{x}, \boldsymbol{x}')$

Optimization goal: $\text{maximize}_{x \in D} \, f(\boldsymbol{x})$

Procedure: At time *t*

1. Choose $\boldsymbol{x}_t$ and observe noisy sample

$$y_t = f(\boldsymbol{x}_t) + z_t, \quad z_t \sim \mathcal{N}(0, \sigma^2)$$

2. Update the GP posterior model with new observation

After T rounds, report final estimate $\hat{\boldsymbol{x}}_T$

Also, corrupted observations:

$$y_t = f(\boldsymbol{x}_t) + z_t + a_t(\boldsymbol{x}_t)$$

[I.B., A. Krause, J. Scarlett (2019)]

Robust Bayesian Optimization

Standard Optimization	$\max_{x \in D} f(x)$	Many works (e.g., Srinivas *et al.* ICML'11, I.B. *et al.* NeurIPS'16)	Molecular design
Robust Optimization (RO)	$\max_{x \in D} \min_{c \in C} f(x, c)$	I.B., J. Scarlett, S. Jegelka, V. Cevher (*NeurIPS*'18)	Robot pushing tasks
Mixed Strategy RO (MRO)	$\max_{P \in \Delta(D)} \min_{c \in C} \mathbb{E}_{x \sim P}[f(x, c)]$	P. G. Sessa, I.B., M. Kamgarpour, A. Krause (2019)	Trajectory planning for AVs
Distributionally RO (DRO)	$\max_{x \in D} \min_{Q \in \mathcal{U}} \mathbb{E}_{c \sim Q}[f(x, c)]$	J. Kirschner, I.B., S. Jegelka, A. Krause (2019)	Crop recommendation

Robust Bayesian Optimization

Standard Optimization	$\max_{x \in D} f(x)$	Many works (e.g., Srinivas *et al.* ICML'11, I.B. *et al.* NeurIPS'16)	Molecular design
Robust Optimization (RO)	$\max_{x \in D} \min_{c \in C} f(x, c)$	I.B., J. Scarlett, S. Jegelka, V. Cevher (*NeurIPS*'18)	Robot pushing tasks
Mixed Strategy RO (MRO)	$\max_{P \in \Delta(D)} \min_{c \in C} \mathbb{E}_{x \sim P}[f(x, c)]$	P. G. Sessa, I.B., M. Kamgarpour, A. Krause (2019)	Trajectory planning for AVs
Distributionally RO (DRO)	$\max_{x \in D} \min_{Q \in \mathcal{U}} \mathbb{E}_{c \sim Q}[f(x, c)]$	J. Kirschner, I.B., S. Jegelka, A. Krause (2019)	Crop recommendation

Robust Bayesian Optimization

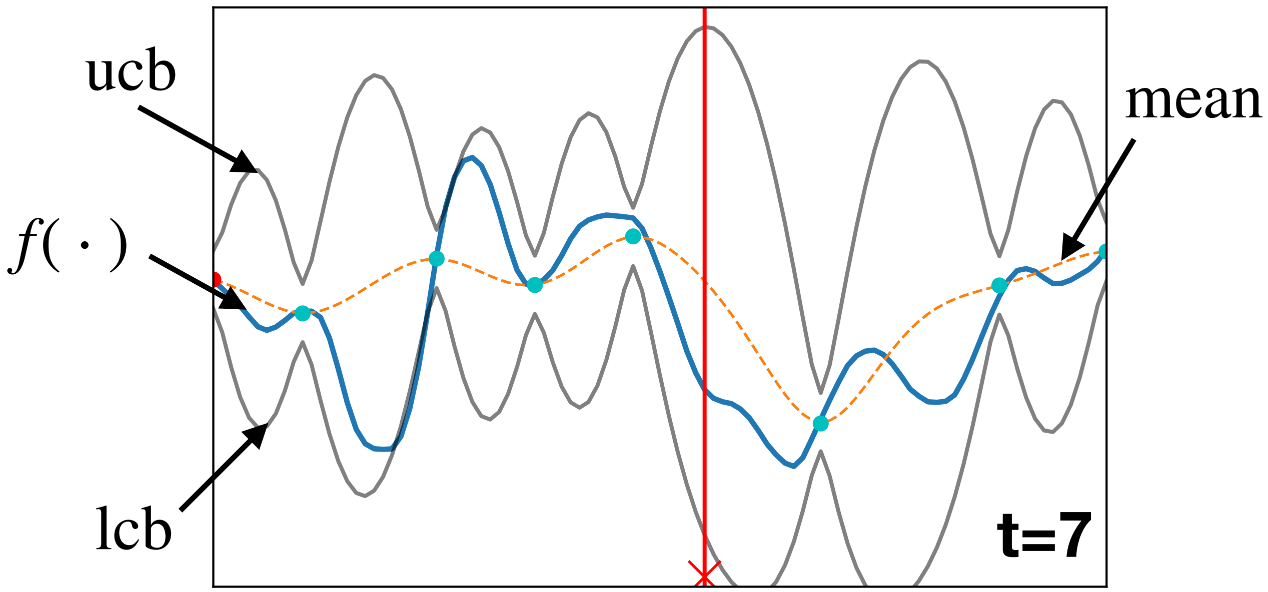
Standard Optimization	$\max_{x \in D} f(x)$	Many works (e.g., Srinivas *et al.* ICML'11, I.B. *et al.* NeurIPS'16)	Molecular design
Robust Optimization (RO)	$\max_{x \in D} \min_{c \in C} f(x, c)$	I.B., J. Scarlett, S. Jegelka, V. Cevher (*NeurIPS*'18)	Robot pushing tasks
Mixed Strategy RO (MRO)	$\max_{P \in \Delta(D)} \min_{c \in C} \mathbb{E}_{x \sim P}[f(x, c)]$	P. G. Sessa, I.B., M. Kamgarpour, A. Krause (2019)	Trajectory planning for AVs
Distributionally RO (DRO)	$\max_{x \in D} \min_{Q \in \mathcal{U}} \mathbb{E}_{c \sim Q}[f(x, c)]$	J. Kirschner, I.B., S. Jegelka, A. Krause (2019)	Crop recommendation

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \le f(\boldsymbol{x}) \le \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]
PI [Kushner'64]
EI [Mockus *et al.'78*]
GP-UCB [Srinivas *et al.*'11]
ES [Henning *et al.*'12]
GP-UCB-PE [Contal *et al.'13*]
BamSOO [Wang *et al.*'14]
PES [Hernandez-Lobato *et al.*'14]
MRS [Metzen'16]
GLASSES [Gonzalez *et al.*'15]
OPES [Hoffman & Ghahramani'15]
TruVaR [I.B. *et al.'16*]
MES [Wang & Jegelka'17]
FITBO [Ru *et al.*'18]
KG [Wu *et al.*'17]
the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

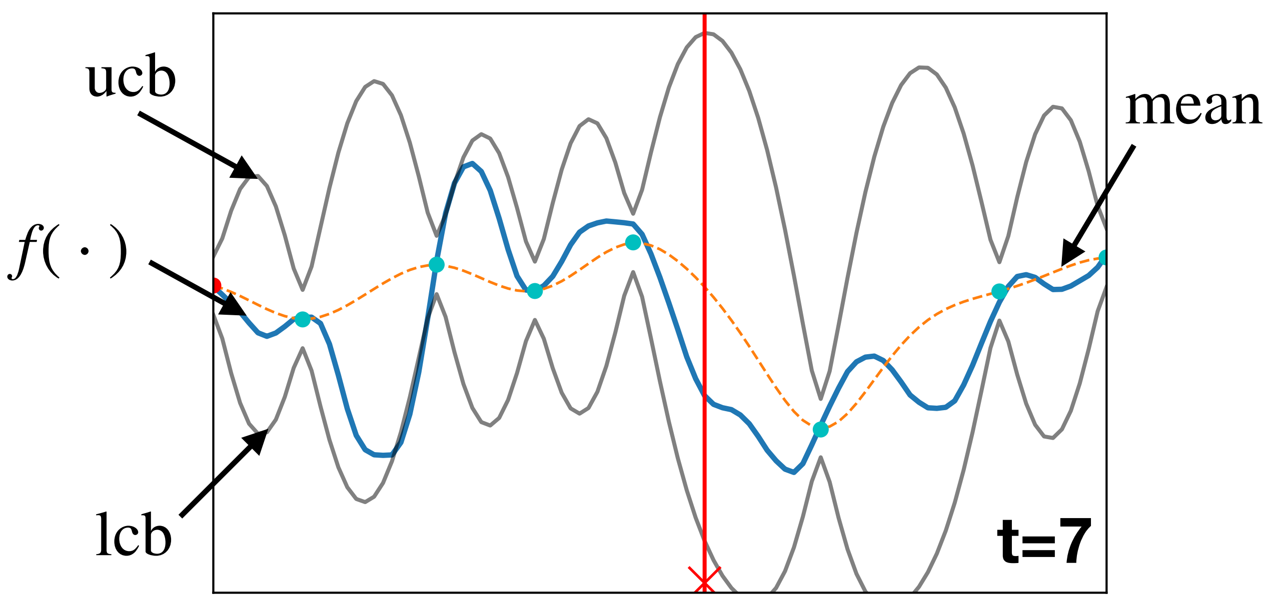
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]
PI [Kushner'64]
EI [Mockus *et al.'78*]
GP-UCB [Srinivas *et al.*'11]
ES [Henning *et al.*'12]
GP-UCB-PE [Contal *et al.'13*]
BamSOO [Wang *et al.*'14]
PES [Hernandez-Lobato *et al.*'14]
MRS [Metzen'16]
GLASSES [Gonzalez *et al.*'15]
OPES [Hoffman & Ghahramani'15]
TruVaR [I.B. *et al.'16*]
MES [Wang & Jegelka'17]
FITBO [Ru *et al.*'18]
KG [Wu *et al.*'17]
the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

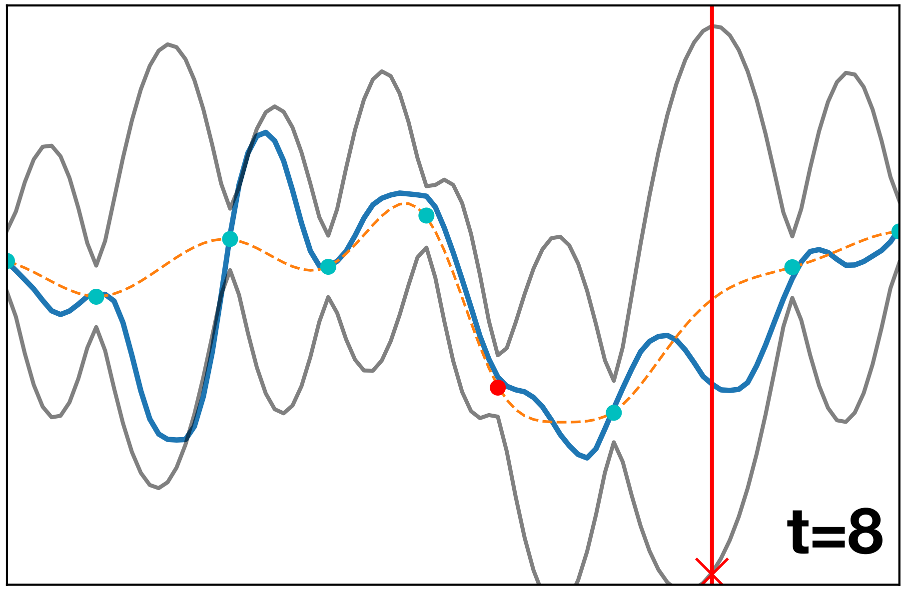
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time t, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]

PI [Kushner'64]

EI [Mockus *et al.'78*]

GP-UCB [Srinivas *et al.*'11]

ES [Henning *et al.*'12]

GP-UCB-PE [Contal *et al.'13*]

BamSOO [Wang *et al.*'14]

PES [Hernandez-Lobato *et al.*'14]

MRS [Metzen'16]

GLASSES [Gonzalez *et al.*'15]

OPES [Hoffman & Ghahramani'15]

TruVaR [I.B. *et al.'16*]

MES [Wang & Jegelka'17]

FITBO [Ru *et al.*'18]

KG [Wu *et al.*'17]

the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

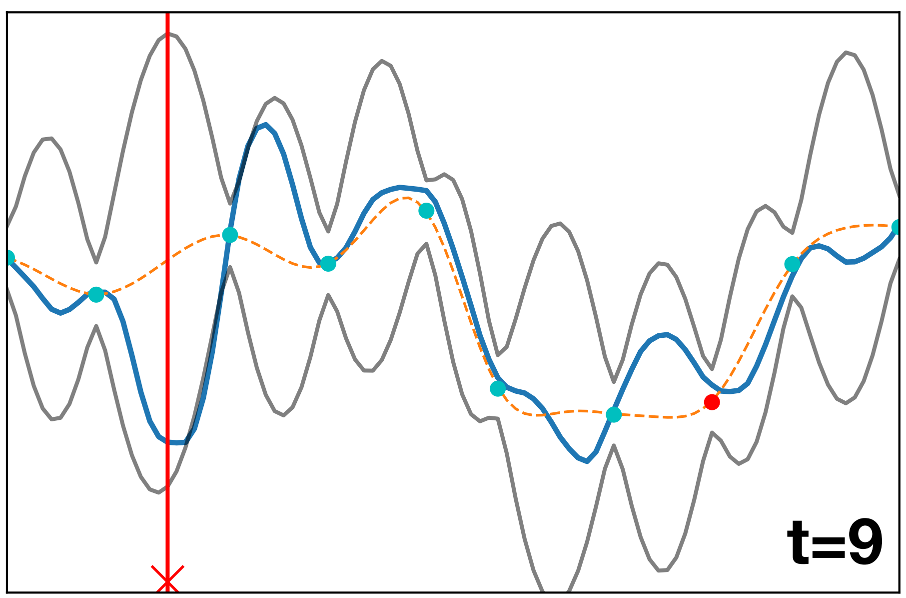
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time t, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]

PI [Kushner'64]

EI [Mockus *et al.'78*]

GP-UCB [Srinivas *et al.*'11]

ES [Henning *et al.*'12]

GP-UCB-PE [Contal *et al.'13*]

BamSOO [Wang *et al.*'14]

PES [Hernandez-Lobato *et al.*'14]

MRS [Metzen'16]

GLASSES [Gonzalez *et al.*'15]

OPES [Hoffman & Ghahramani'15]

TruVaR [I.B. *et al.'16*]

MES [Wang & Jegelka'17]

FITBO [Ru *et al.*'18]

KG [Wu *et al.*'17]

the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

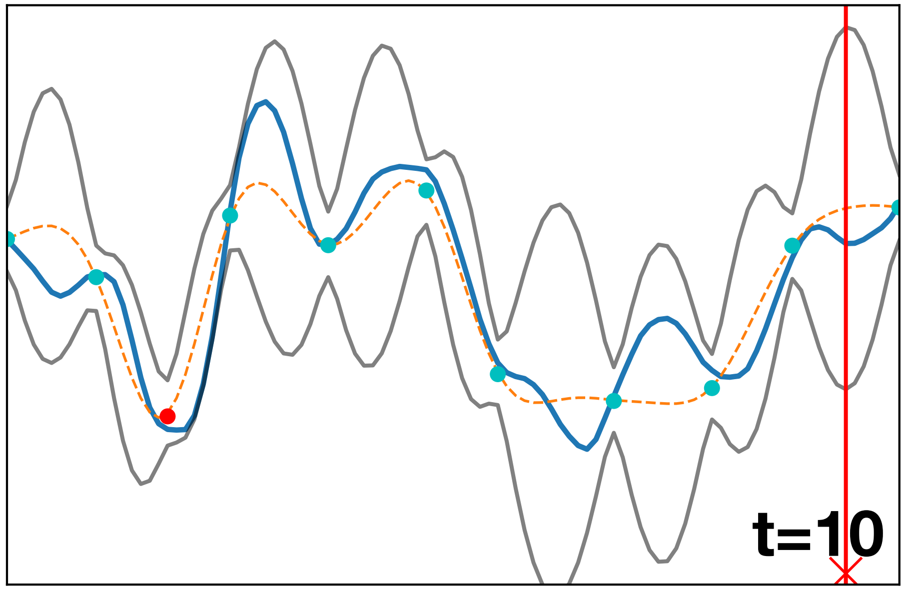
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]

PI [Kushner'64]

EI [Mockus *et al.'78*]

GP-UCB [Srinivas *et al.*'11]

ES [Henning *et al.*'12]

GP-UCB-PE [Contal *et al.'13*]

BamSOO [Wang *et al.*'14]

PES [Hernandez-Lobato *et al.*'14]

MRS [Metzen'16]

GLASSES [Gonzalez *et al.*'15]

OPES [Hoffman & Ghahramani'15]

TruVaR [I.B. *et al.'16*]

MES [Wang & Jegelka'17]

FITBO [Ru *et al.*'18]

KG [Wu *et al.*'17]

the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

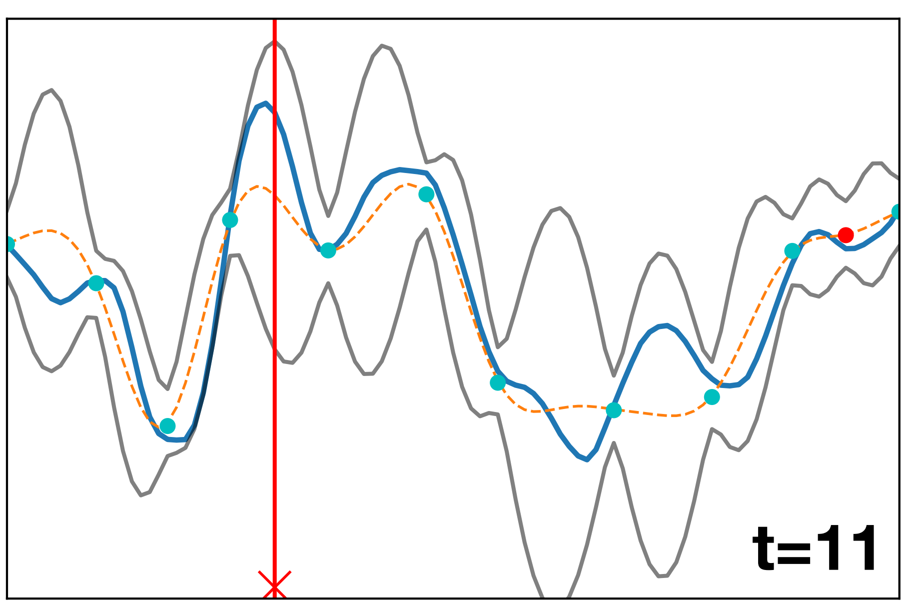
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]
PI [Kushner'64]
EI [Mockus *et al.'78*]
GP-UCB [Srinivas *et al.*'11]
ES [Henning *et al.*'12]
GP-UCB-PE [Contal *et al.'13*]
BamSOO [Wang *et al.*'14]
PES [Hernandez-Lobato *et al.*'14]
MRS [Metzen'16]
GLASSES [Gonzalez *et al.*'15]
OPES [Hoffman & Ghahramani'15]
TruVaR [I.B. *et al.'16*]
MES [Wang & Jegelka'17]
FITBO [Ru *et al.*'18]
KG [Wu *et al.*'17]
the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

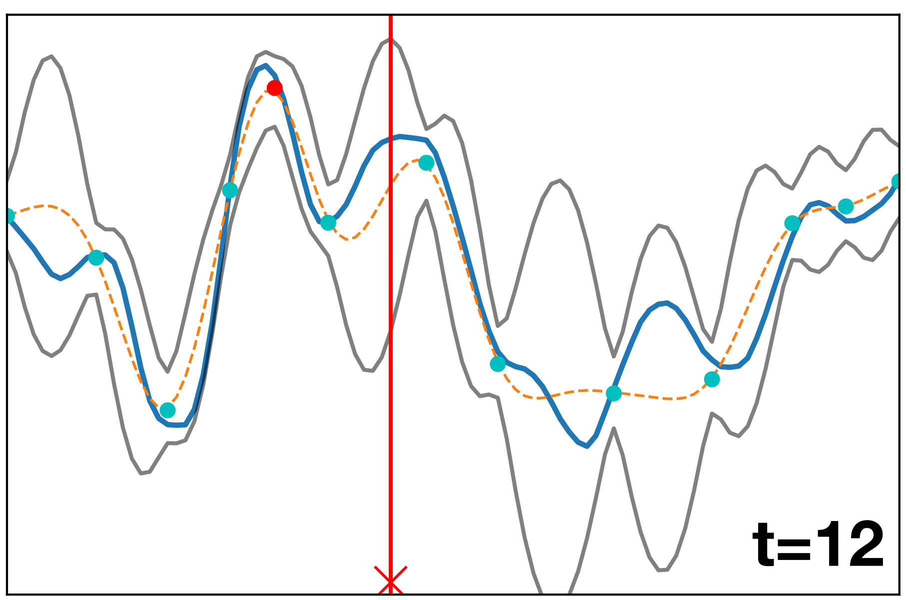
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]

PI [Kushner'64]

EI [Mockus *et al.'78*]

GP-UCB [Srinivas *et al.*'11]

ES [Henning *et al.*'12]

GP-UCB-PE [Contal *et al.'13*]

BamSOO [Wang *et al.*'14]

PES [Hernandez-Lobato *et al.*'14]

MRS [Metzen'16]

GLASSES [Gonzalez *et al.*'15]

OPES [Hoffman & Ghahramani'15]

TruVaR [I.B. *et al.'16*]

MES [Wang & Jegelka'17]

FITBO [Ru *et al.*'18]

KG [Wu *et al.*'17]

the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

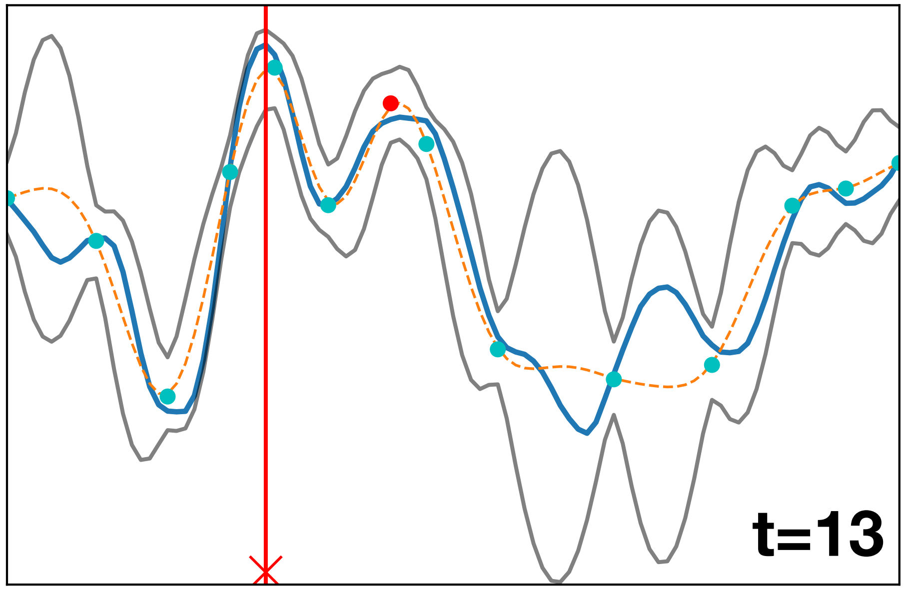
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]
PI [Kushner'64]
EI [Mockus *et al.'78*]
GP-UCB [Srinivas *et al.*'11]
ES [Henning *et al.*'12]
GP-UCB-PE [Contal *et al.'13*]
BamSOO [Wang *et al.*'14]
PES [Hernandez-Lobato *et al.*'14]
MRS [Metzen'16]
GLASSES [Gonzalez *et al.*'15]
OPES [Hoffman & Ghahramani'15]
TruVaR [I.B. *et al.'16*]
MES [Wang & Jegelka'17]
FITBO [Ru *et al.*'18]
KG [Wu *et al.*'17]
the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

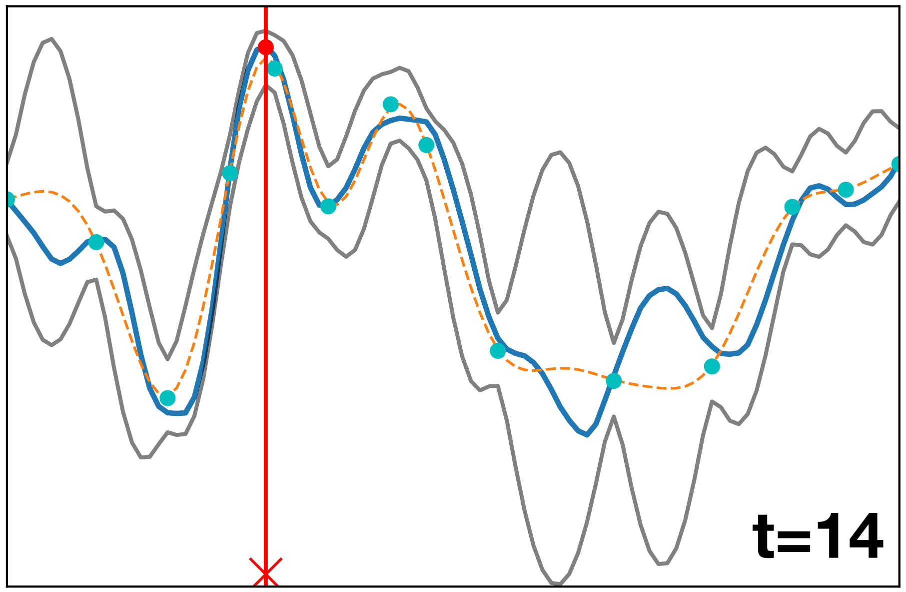
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]
PI [Kushner'64]
EI [Mockus *et al.'78*]
GP-UCB [Srinivas *et al.*'11]
ES [Henning *et al.*'12]
GP-UCB-PE [Contal *et al.'13*]
BamSOO [Wang *et al.*'14]
PES [Hernandez-Lobato *et al.*'14]
MRS [Metzen'16]
GLASSES [Gonzalez *et al.*'15]
OPES [Hoffman & Ghahramani'15]
TruVaR [I.B. *et al.'16*]
MES [Wang & Jegelka'17]
FITBO [Ru *et al.*'18]
KG [Wu *et al.*'17]
the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

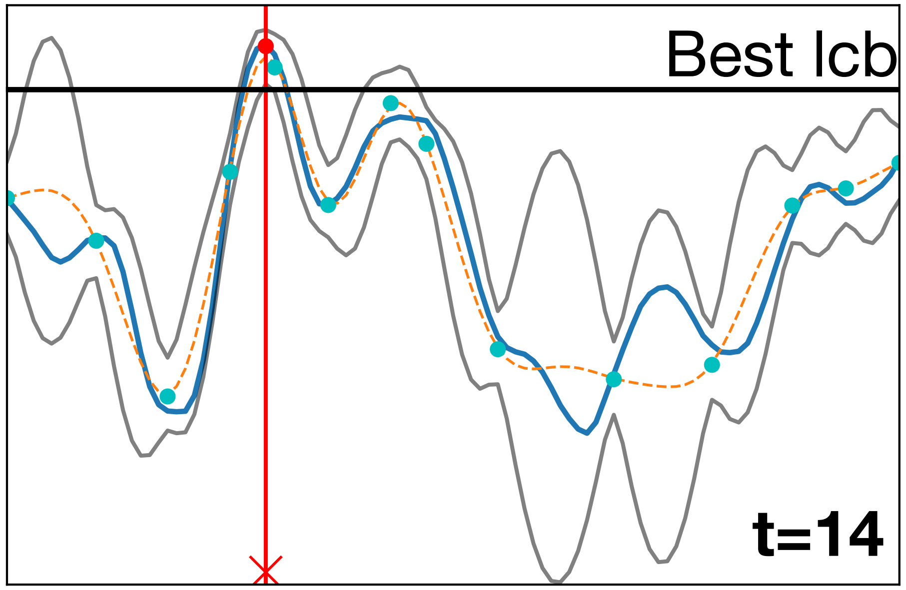
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]
PI [Kushner'64]
EI [Mockus *et al.'78*]
GP-UCB [Srinivas *et al.*'11]
ES [Henning *et al.*'12]
GP-UCB-PE [Contal *et al.'13*]
BamSOO [Wang *et al.*'14]
PES [Hernandez-Lobato *et al.*'14]
MRS [Metzen'16]
GLASSES [Gonzalez *et al.*'15]
OPES [Hoffman & Ghahramani'15]
TruVaR [I.B. *et al.'16*]
MES [Wang & Jegelka'17]
FITBO [Ru *et al.*'18]
KG [Wu *et al.*'17]
the list goes on…

Standard Optimization

Upper Confidence Bound (GP-UCB) [Srinivas *et al.*'11]

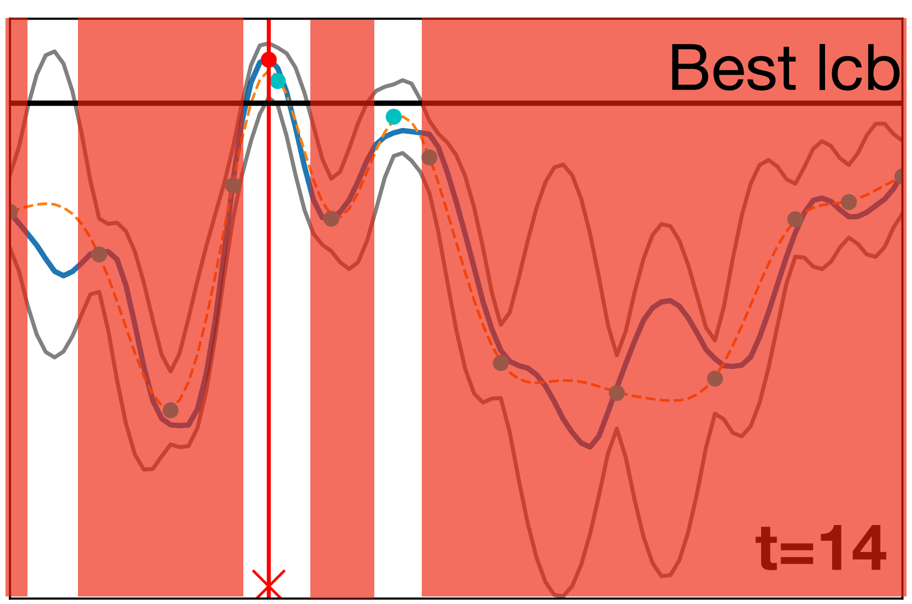
- Construct confidence bounds such that w.h.p.

$$\mathrm{lcb}_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mathrm{ucb}_t(\boldsymbol{x}), \quad \forall \boldsymbol{x}, t$$

- At time *t*, query the point

$$\boldsymbol{x}_t = \arg\max_{x \in D} \mathrm{ucb}_t(\boldsymbol{x})$$

and then observe y_t, and update the confidence bounds



Key idea: Optimism in the face of **uncertainty**

Others:

Thompson [Thompson '33]
PI [Kushner'64]
EI [Mockus *et al.'78*]
GP-UCB [Srinivas *et al.*'11]
ES [Henning *et al.*'12]
GP-UCB-PE [Contal *et al.'13*]
BamSOO [Wang *et al.*'14]
PES [Hernandez-Lobato *et al.*'14]
MRS [Metzen'16]
GLASSES [Gonzalez *et al.*'15]
OPES [Hoffman & Ghahramani'15]
TruVaR [I.B. *et al.'16*]
MES [Wang & Jegelka'17]
FITBO [Ru *et al.*'18]
KG [Wu *et al.*'17]
the list goes on…

Adversarially Robust Objective

[I.B., J.Scarlett, S. Jegelka, V. Cevher; ***NeurIPS*'18**]

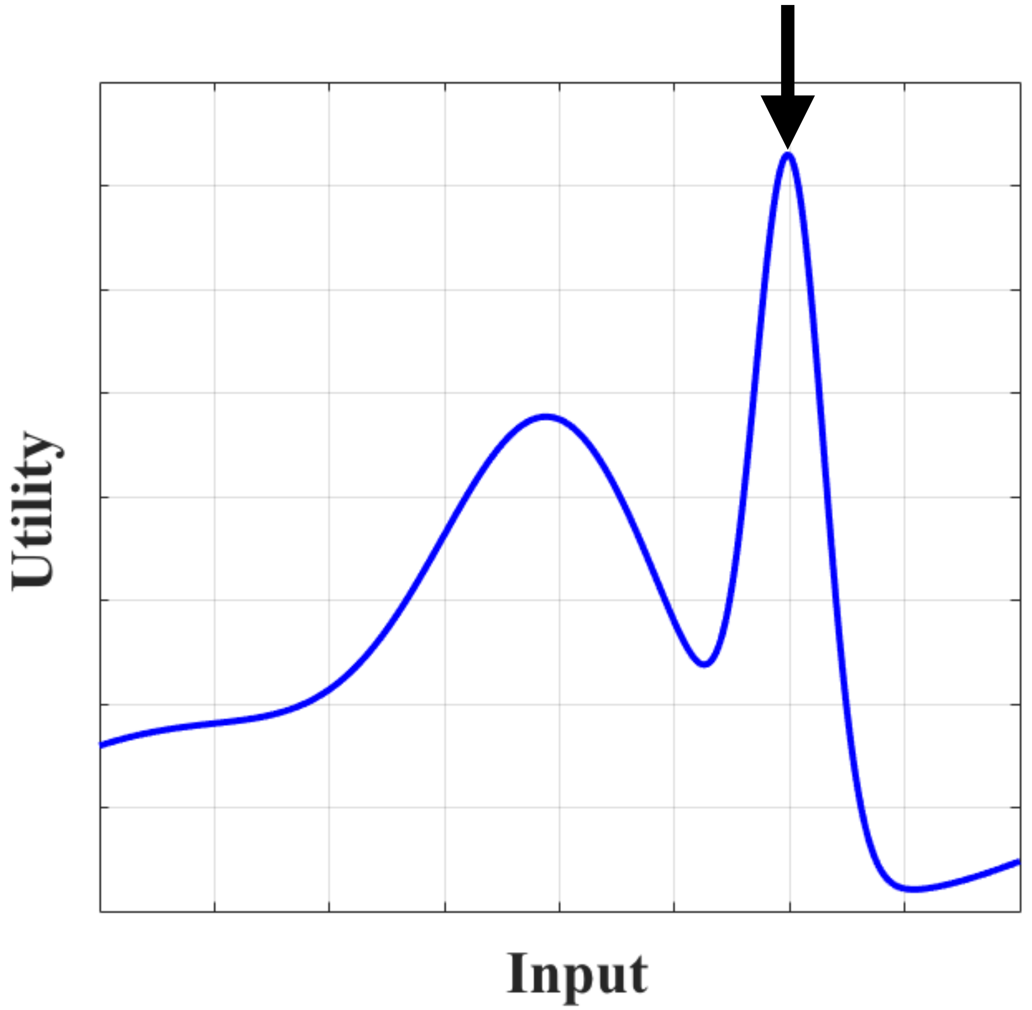
Robust problem:

$$x^* = \arg\max_{x \in D} \min_{\delta \in \Delta_\epsilon(x)} f(x+\delta)$$

Set of input perturbations:

$$\Delta_\epsilon(x) = \{x' - x \ : \ \mathrm{dist}(x, x') \leq \epsilon\}$$

Optimum



Adversarially Robust Objective

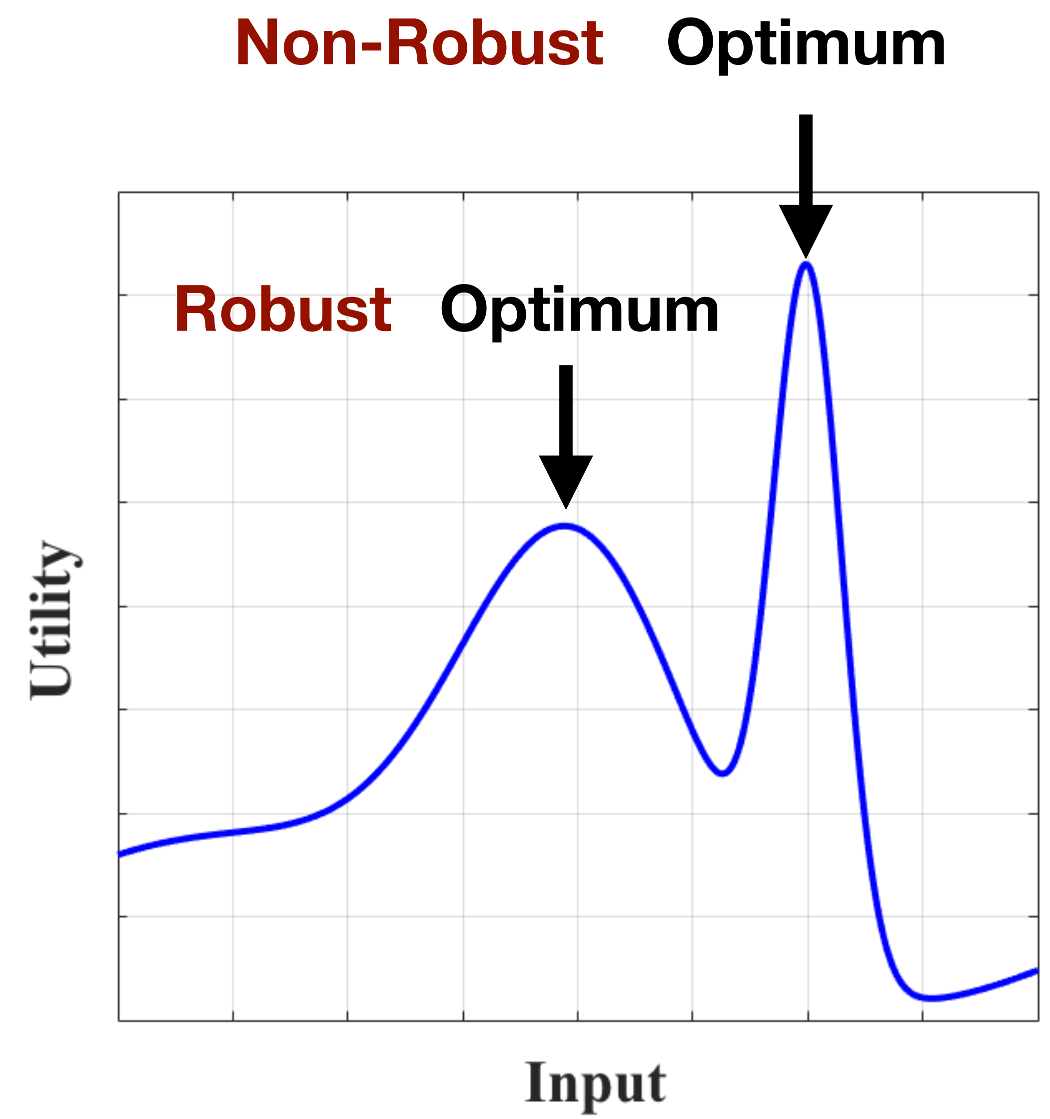
[I.B., J.Scarlett, S. Jegelka, V. Cevher; ***NeurIPS****’18*]

Robust problem:

$$x^* = \arg\max_{x \in D} \min_{\delta \in \Delta_\epsilon(x)} f(x+\delta)$$

Set of input perturbations:

$$\Delta_\epsilon(x) = \{x' - x \ : \ \mathrm{dist}(x, x') \le \epsilon\}$$



Adversarially Robust Objective

[I.B., J.Scarlett, S. Jegelka, V. Cevher; ***NeurIPS****'18*]

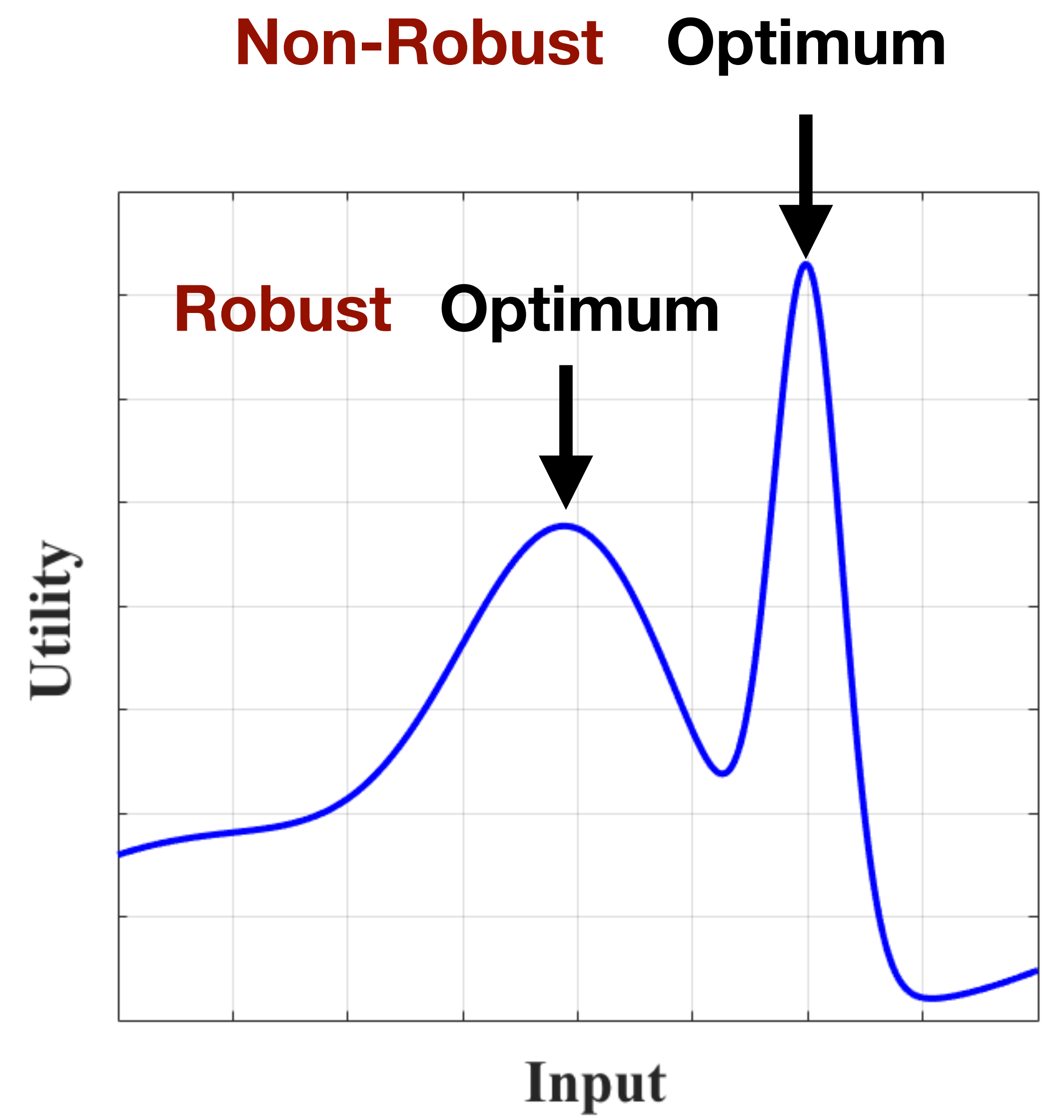
Robust problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in D} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})} f(\boldsymbol{x} + \boldsymbol{\delta})$$

Set of input perturbations:

$$\Delta_\epsilon(\boldsymbol{x}) = \left\{ \boldsymbol{x}' - \boldsymbol{x} \ : \ \mathrm{dist}(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon \right\}$$

Recall: $y_t = f(\boldsymbol{x}_t) + z_t$ (sub-Gaussian noise)



Adversarially Robust Objective

[I.B., J.Scarlett, S. Jegelka, V. Cevher; ***NeurIPS****'18*]

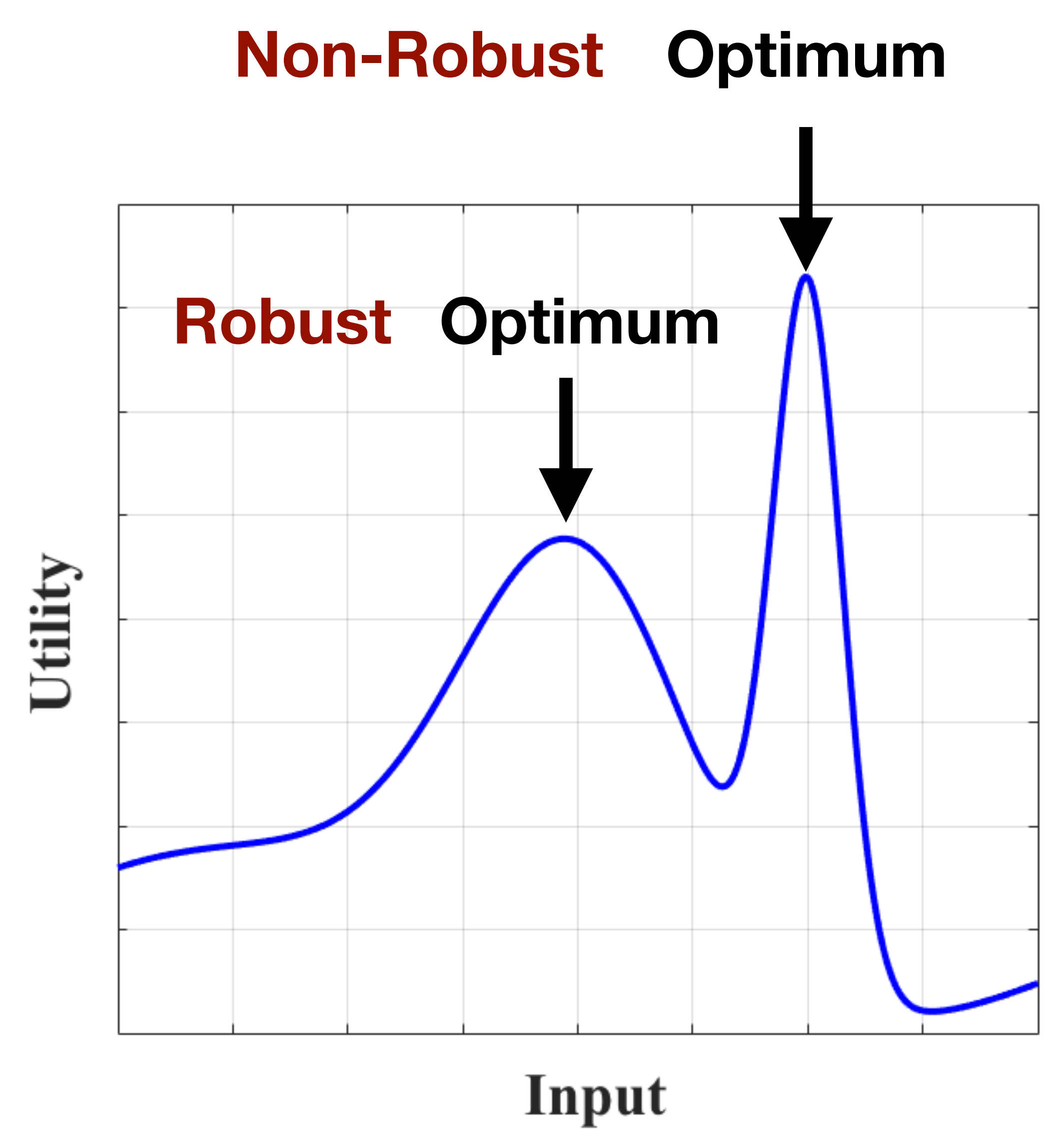
Robust problem:

$$x^* = \arg\max_{x \in D} \min_{\delta \in \Delta_\epsilon(x)} f(x+\delta)$$

Set of input perturbations:

$$\Delta_\epsilon(x) = \{x' - x \ : \ \mathrm{dist}(x, x') \le \epsilon\}$$

Recall: $y_t = f(x_t) + z_t$ (sub-Gaussian noise)

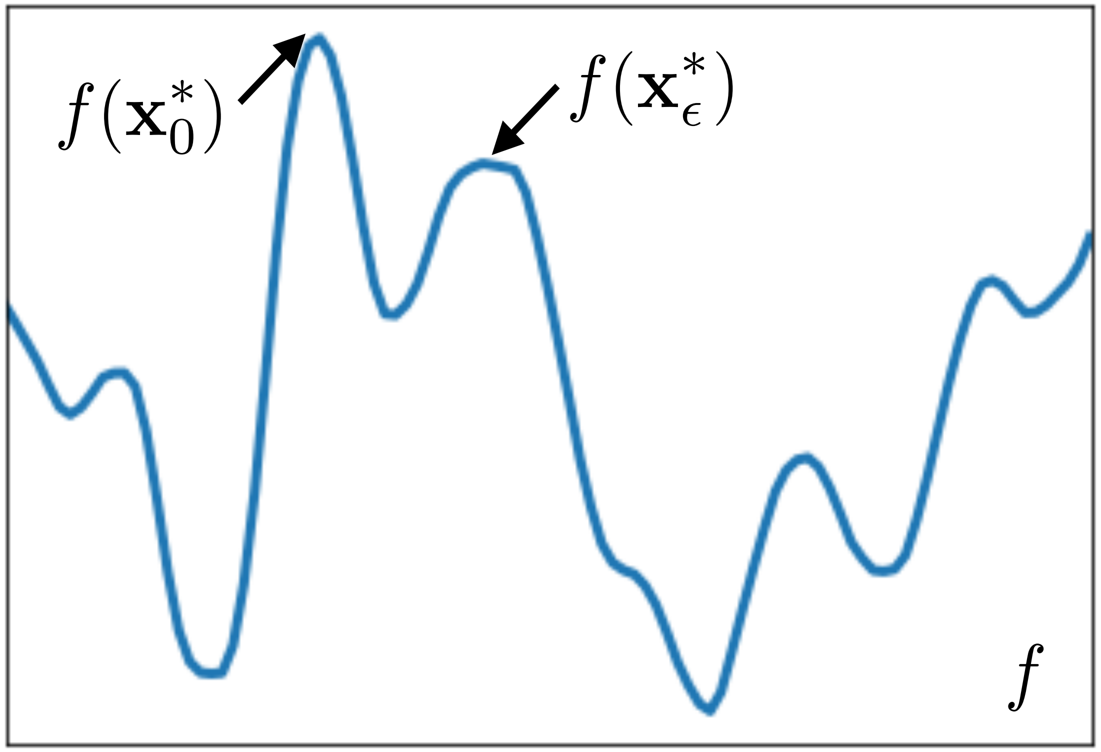


Goal: After T rounds report $x^{(T)}$ such that **regret** is small

$$\max_{x \in D} \min_{\delta \in \Delta_\epsilon(x)} f(x+\delta) - \min_{\delta \in \Delta_\epsilon(x^{(T)})} f(x^{(T)}+\delta)$$

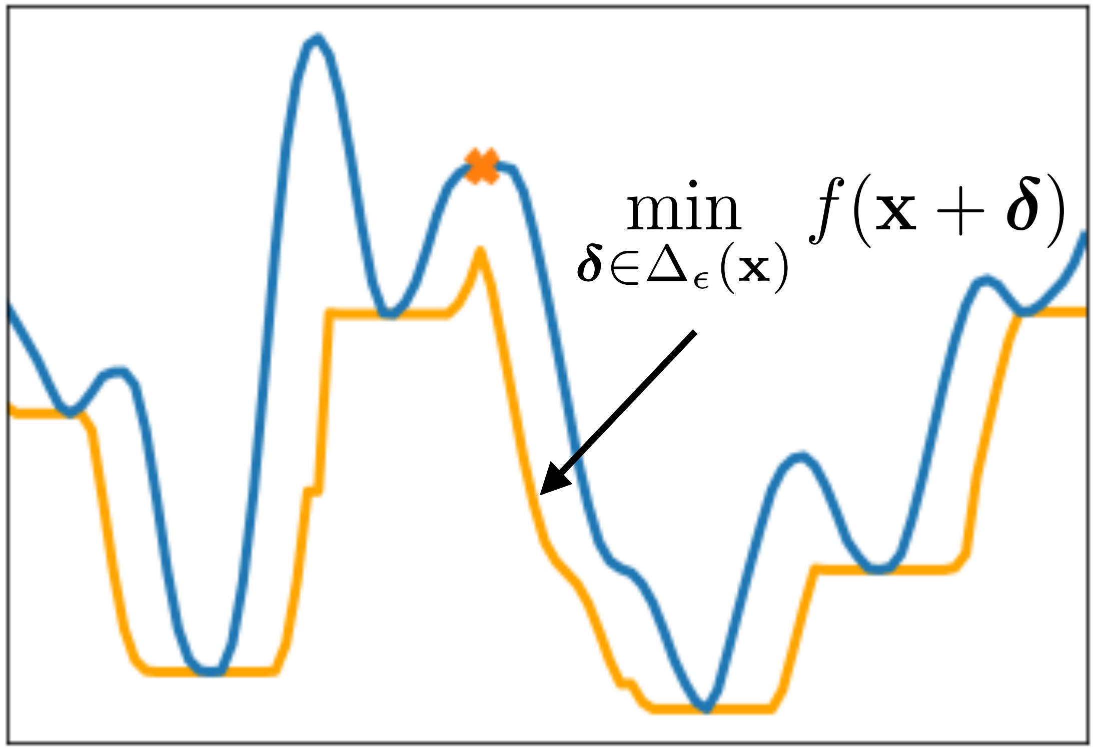
Challenge

Example: Standard BO methods can **fail** to achieve small regret



Challenge

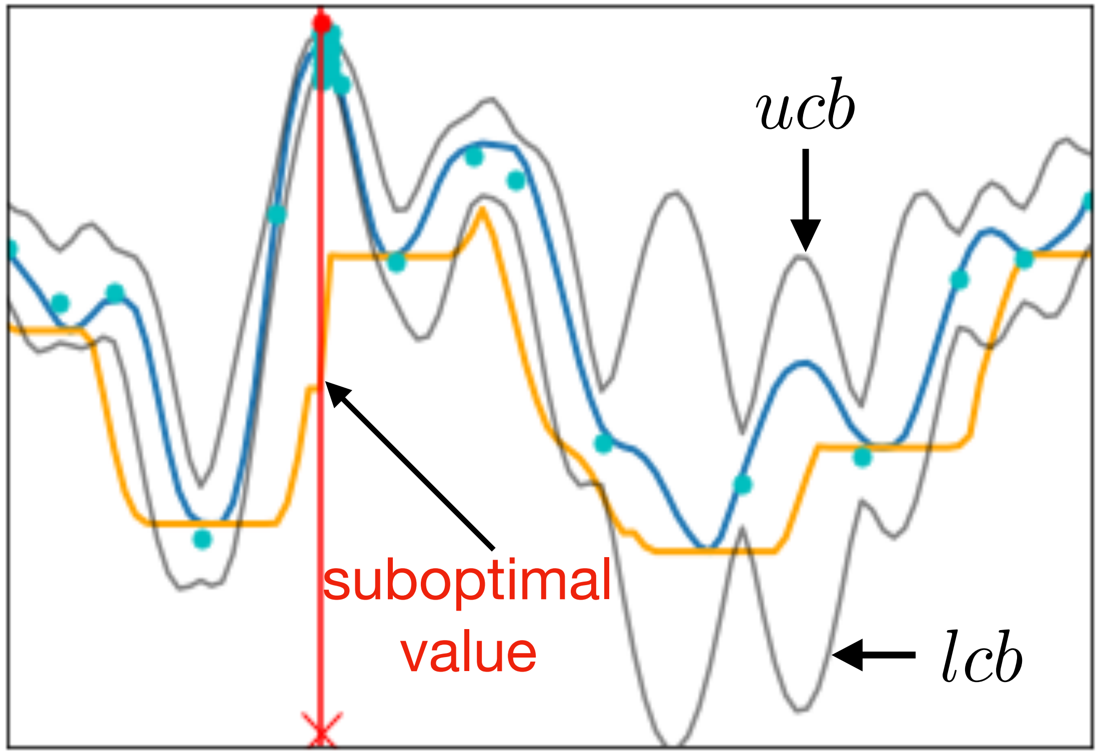
Example: Standard BO methods can **fail** to achieve small regret



$\epsilon = 0.06,\ d(x, x') = |x - x'|$

Challenge

Example: Standard BO methods can **fail** to achieve small regret



GP-UCB

StableOpt

At every round t:

- first: $$\tilde{x}_t = \operatorname*{argmax}_{x \in D} \min_{\delta \in \Delta_\epsilon(x)} \mathrm{ucb}_t(\boldsymbol{x} + \boldsymbol{\delta})$$
- second:

StableOpt

At every round t:

- first: $$\tilde{\boldsymbol{x}}_t = \operatorname*{argmax}_{\boldsymbol{x} \in D} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})} \mathrm{ucb}_t(\boldsymbol{x} + \boldsymbol{\delta})$$
- second: $$\boldsymbol{\delta}_t = \operatorname*{argmin}_{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\boldsymbol{x}}_t)} \mathrm{lcb}_t(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta})$$

StableOpt

At every round t:

- first: $\tilde{\boldsymbol{x}}_t = \underset{\boldsymbol{x} \in D}{\operatorname{argmax}} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})} \operatorname{ucb}_t(\boldsymbol{x}+\boldsymbol{\delta})$
- second: $\boldsymbol{\delta}_t = \underset{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\boldsymbol{x}}_t)}{\operatorname{argmin}} \operatorname{lcb}_t(\tilde{\boldsymbol{x}}_t+\boldsymbol{\delta})$
- observe $y_t = f(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t) + z_t$ and update model by including $\{(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t, y_t)\}$

StableOpt

At every round t:

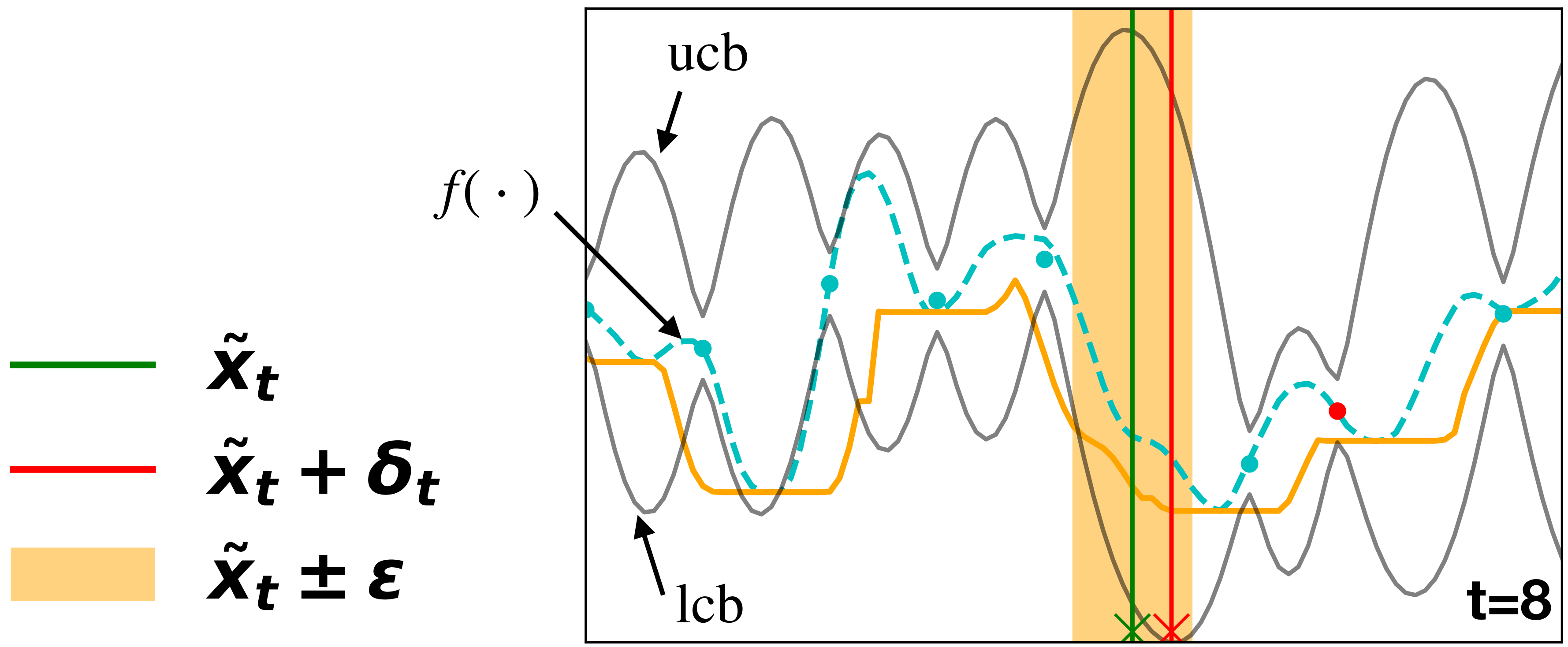
- first: $$\tilde{\boldsymbol{x}}_t = \operatorname*{argmax}_{\boldsymbol{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\boldsymbol{x})} \mathrm{ucb}_t(\boldsymbol{x}+\boldsymbol{\delta})$$
- second: $$\boldsymbol{\delta}_t = \operatorname*{argmin}_{\boldsymbol{\delta}\in\Delta_\epsilon(\tilde{\boldsymbol{x}}_t)} \mathrm{lcb}_t(\tilde{\boldsymbol{x}}_t+\boldsymbol{\delta})$$
- observe $y_t = f(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t) + z_t$ and update model by including $\{(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t, y_t)\}$

- Optimism in the face of uncertainty for choosing $\tilde{\boldsymbol{x}}_t$
- Pessimism in the face of uncertainty for choosing $\boldsymbol{\delta}_t$

StableOpt

At every round t:

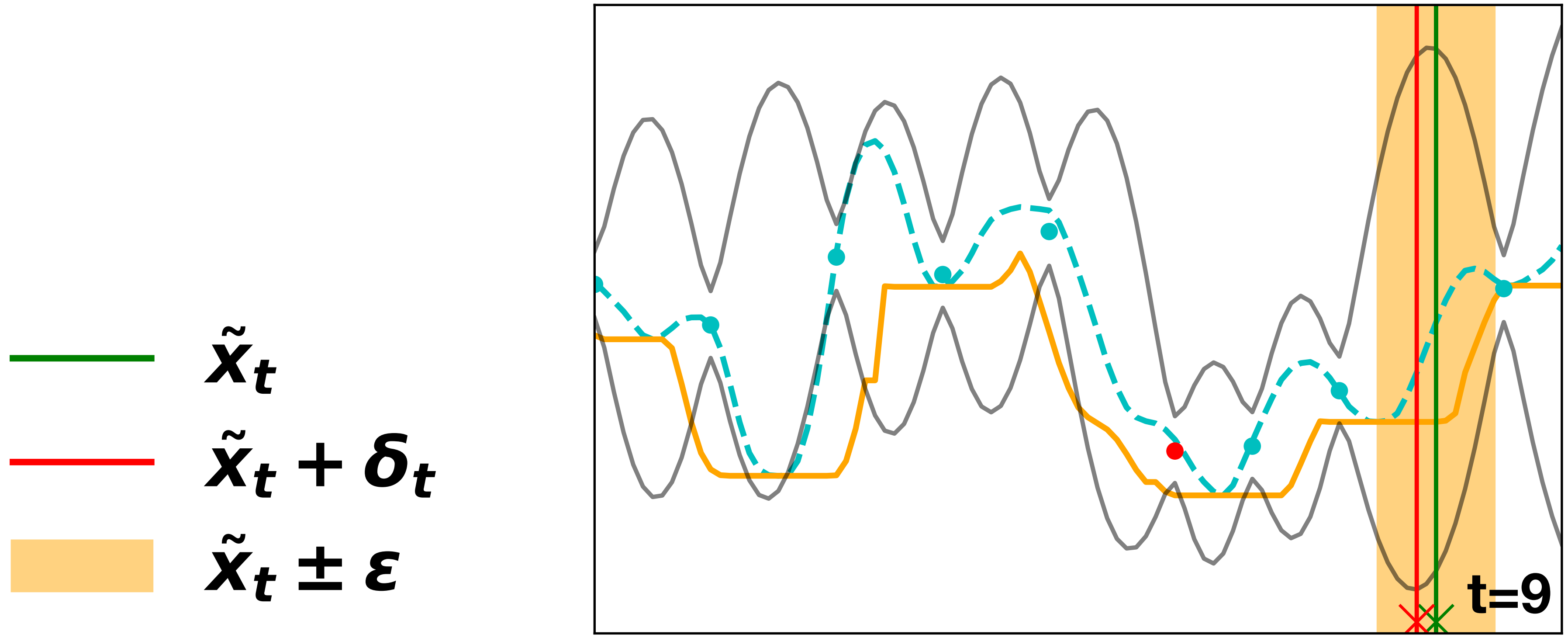
- first: $\tilde{\boldsymbol{x}}_t = \underset{x \in D}{\mathrm{argmax}} \min_{\delta \in \Delta_\epsilon(\boldsymbol{x})} \mathrm{ucb}_t(\boldsymbol{x}+\boldsymbol{\delta})$
- second: $\boldsymbol{\delta}_t = \underset{\delta \in \Delta_\epsilon(\tilde{x}_t)}{\mathrm{argmin}} \, \mathrm{lcb}_t(\tilde{\boldsymbol{x}}_t+\boldsymbol{\delta})$
- observe $y_t = f(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t) + z_t$ and update model by including $\{(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t, y_t)\}$



StableOpt

At every round t:

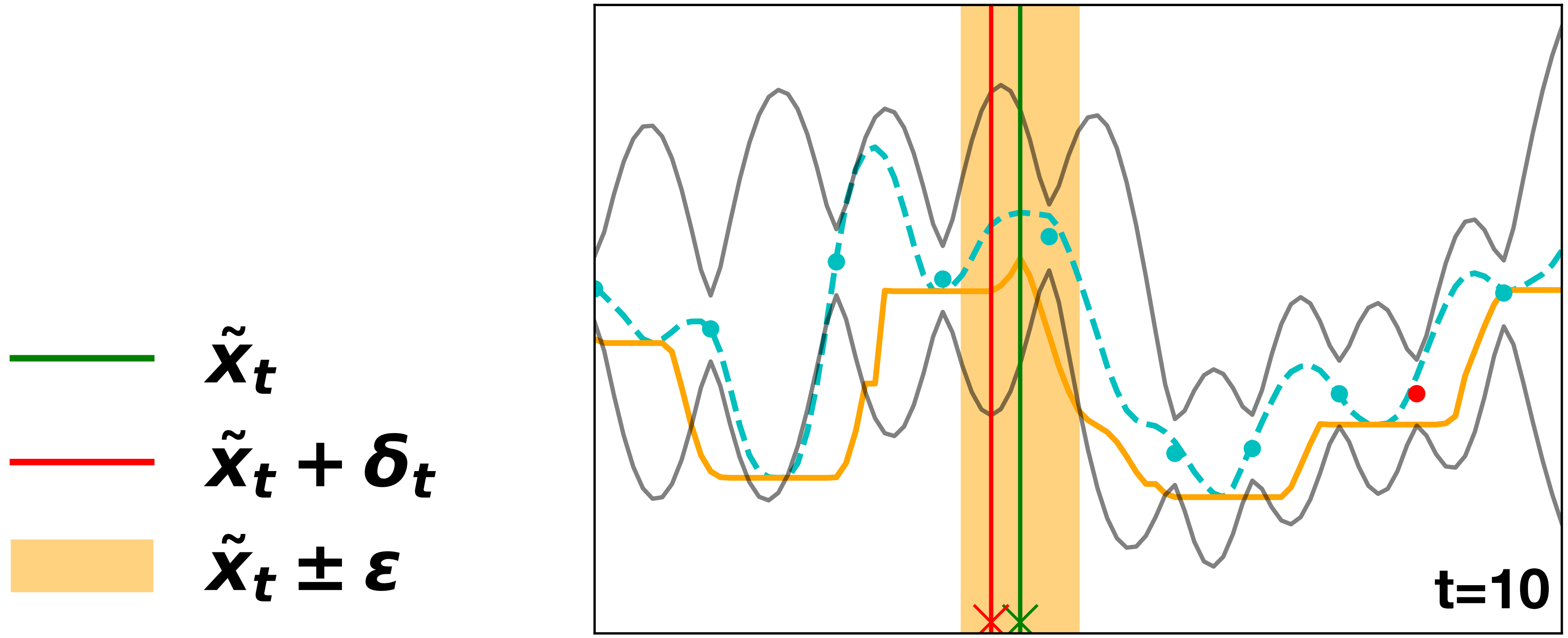
- first: $\tilde{\boldsymbol{x}}_t = \underset{\boldsymbol{x} \in D}{\operatorname{argmax}} \underset{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})}{\min} \operatorname{ucb}_t(\boldsymbol{x} + \boldsymbol{\delta})$
- second: $\boldsymbol{\delta}_t = \underset{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\boldsymbol{x}}_t)}{\operatorname{argmin}} \operatorname{lcb}_t(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta})$
- observe $y_t = f(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t) + z_t$ and update model by including $\{(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t, y_t)\}$



StableOpt

At every round t:

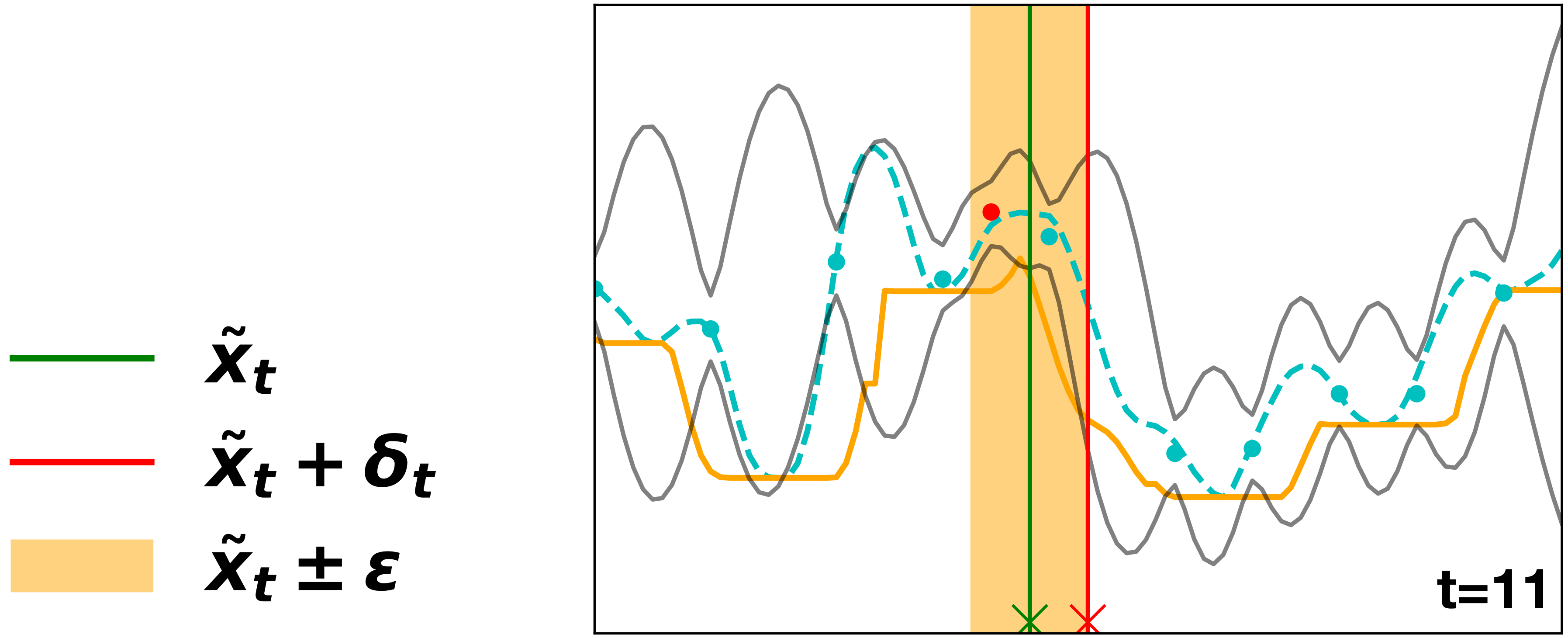
- first: $\tilde{x}_t = \underset{x \in D}{\operatorname{argmax}} \min_{\delta \in \Delta_\epsilon(x)} \operatorname{ucb}_t(x + \delta)$
- second: $\delta_t = \underset{\delta \in \Delta_\epsilon(\tilde{x}_t)}{\operatorname{argmin}} \operatorname{lcb}_t(\tilde{x}_t + \delta)$
- observe $y_t = f(\tilde{x}_t + \delta_t) + z_t$ and update model by including $\{(\tilde{x}_t + \delta_t, y_t)\}$



StableOpt

At every round *t*:

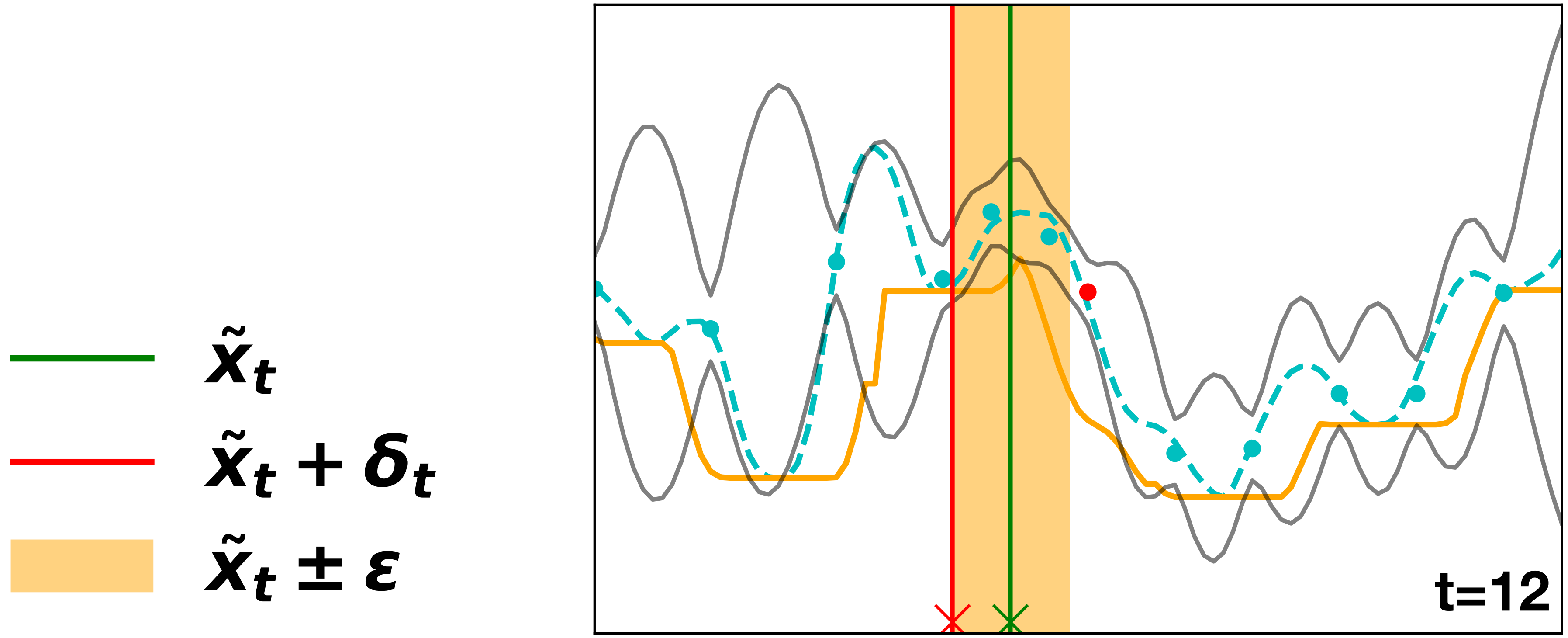
- first: $\tilde{\boldsymbol{x}}_t = \underset{\boldsymbol{x} \in D}{\operatorname{argmax}} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})} \operatorname{ucb}_t(\boldsymbol{x} + \boldsymbol{\delta})$
- second: $\boldsymbol{\delta}_t = \underset{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\boldsymbol{x}}_t)}{\operatorname{argmin}} \operatorname{lcb}_t(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta})$
- observe $y_t = f(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t) + z_t$ and update model by including $\{(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t, y_t)\}$



StableOpt

At every round t:

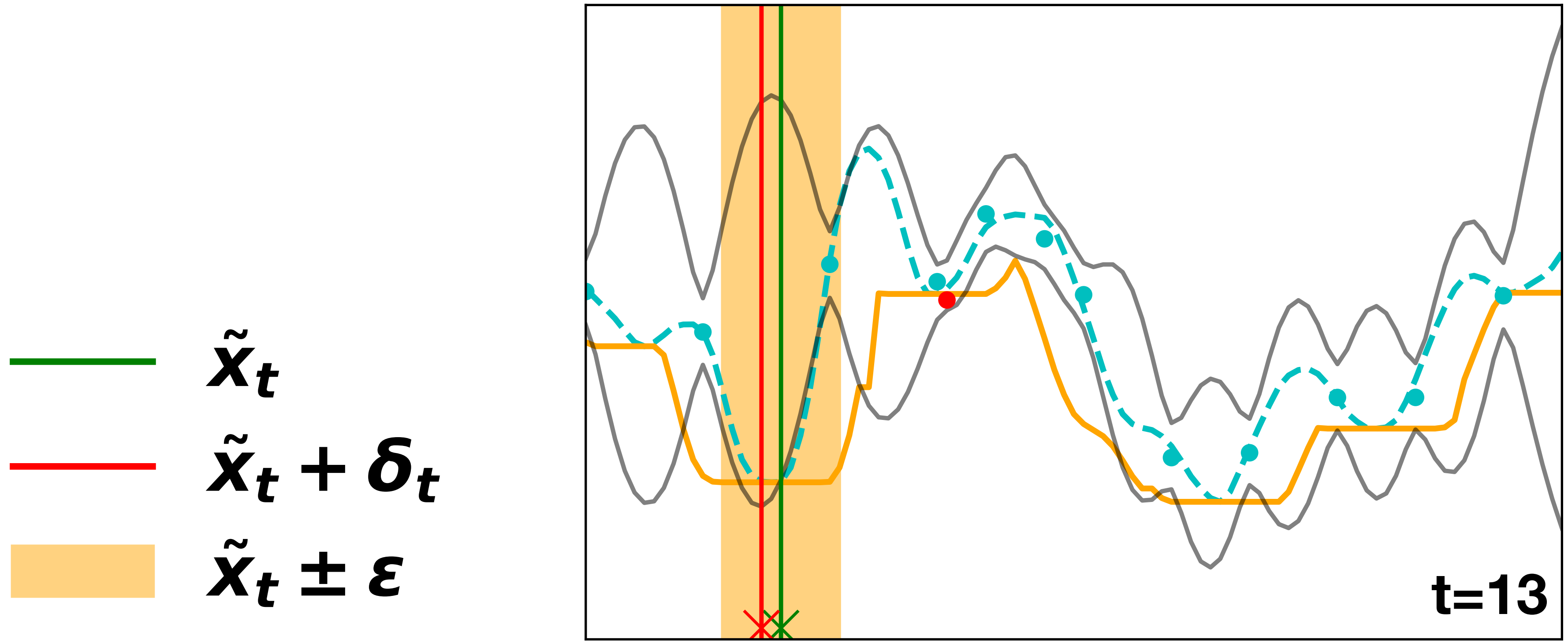
- first: $\tilde{x}_t = \underset{x \in D}{\operatorname{argmax}} \min_{\delta \in \Delta_\epsilon(x)} \mathrm{ucb}_t(x + \delta)$
- second: $\delta_t = \underset{\delta \in \Delta_\epsilon(\tilde{x}_t)}{\operatorname{argmin}} \mathrm{lcb}_t(\tilde{x}_t + \delta)$
- observe $y_t = f(\tilde{x}_t + \delta_t) + z_t$ and update model by including $\{(\tilde{x}_t + \delta_t, y_t)\}$



StableOpt

At every round t:

- first: $\tilde{x}_t = \underset{x \in D}{\operatorname{argmax}} \min_{\delta \in \Delta_\epsilon(x)} \operatorname{ucb}_t(x+\delta)$
- second: $\delta_t = \underset{\delta \in \Delta_\epsilon(\tilde{x}_t)}{\operatorname{argmin}} \operatorname{lcb}_t(\tilde{x}_t+\delta)$
- observe $y_t = f(\tilde{x}_t + \delta_t) + z_t$ and update model by including $\{(\tilde{x}_t + \delta_t, y_t)\}$



StableOpt

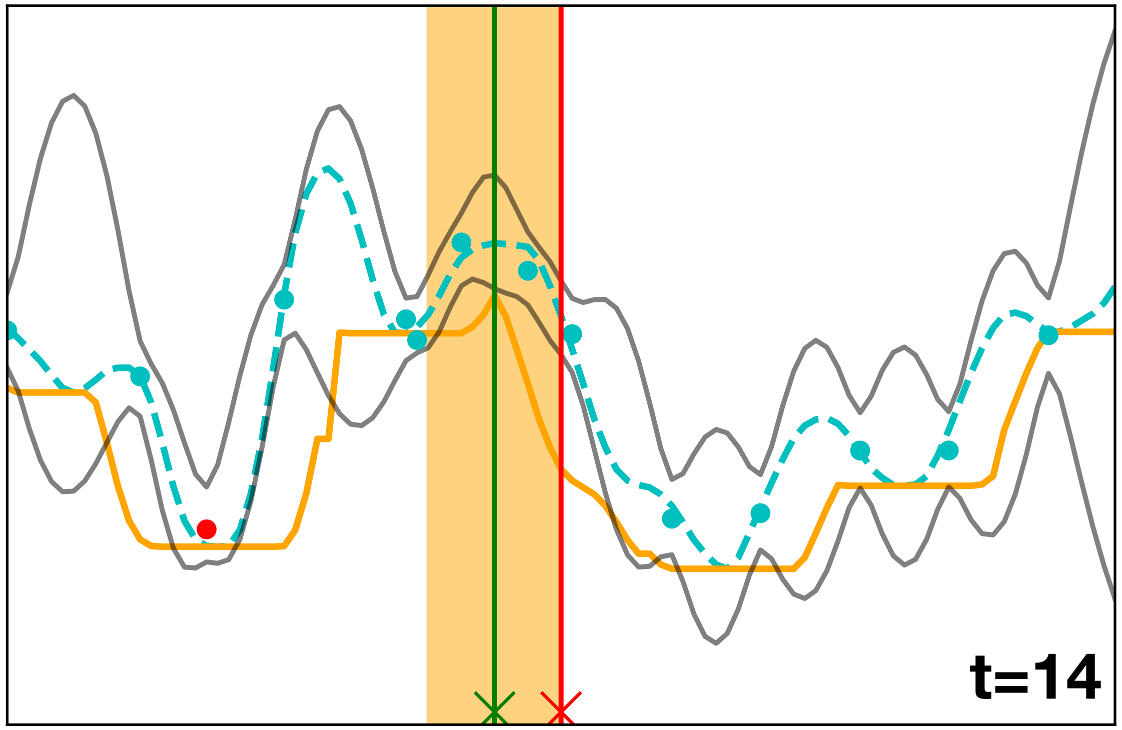
At every round t:

- first: $\tilde{\boldsymbol{x}}_t = \underset{\boldsymbol{x} \in D}{\operatorname{argmax}} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})} \operatorname{ucb}_t(\boldsymbol{x} + \boldsymbol{\delta})$
- second: $\boldsymbol{\delta}_t = \underset{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\boldsymbol{x}}_t)}{\operatorname{argmin}} \operatorname{lcb}_t(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta})$
- observe $y_t = f(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t) + z_t$ and update model by including $\{(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t, y_t)\}$

$\tilde{\boldsymbol{x}}_t$

$\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta}_t$

$\tilde{\boldsymbol{x}}_t \pm \boldsymbol{\epsilon}$



Theoretical guarantee

- Two distinct settings:
  - Bayesian setting (assume $f \sim \mathrm{GP}(\mathbf{0}, k(\cdot\,,\cdot))$)
  - Non-Bayesian setting (assume bounded norm $\|f\|_k$ in RKHS space)

Theorem: After running **StableOpt** for T rounds, if

$$T \gtrsim \frac{\gamma_T}{\eta^2}$$

then the point $\boldsymbol{x}^{(T)} = \tilde{\boldsymbol{x}}_{t^*}, \quad t^* = \operatorname*{argmax}_{t=1,\dots,T} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\boldsymbol{x}}_t)} \mathrm{lcb}_t(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta})$ satisfies w.h.p.:

$$\min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x}^{(T)})} f(\boldsymbol{x}^{(T)} + \boldsymbol{\delta}) \geq \max_{\boldsymbol{x} \in D} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})} f(\boldsymbol{x} + \boldsymbol{\delta}) - \eta \, .$$

γ_T : Kernel-dependent mutual information quantity [Srinivas *et al.*'11]

Theoretical guarantee

- Two distinct settings:
  - Bayesian setting (assume $f \sim \mathrm{GP}(\mathbf{0}, k(\cdot,\cdot))$)
  - Non-Bayesian setting (assume bounded norm $\|f\|_k$ in RKHS space)

Theorem: After running **StableOpt** for *T* rounds, if

$$T \gtrsim \frac{\gamma_T}{\eta^2}$$

then the point $\mathbf{x}^{(T)} = \tilde{\mathbf{x}}_{t^*}, \quad t^* = \operatorname*{argmax}_{t=1,\dots,T} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\tilde{\mathbf{x}}_t)} \mathrm{lcb}_t(\tilde{\mathbf{x}}_t + \boldsymbol{\delta})$ satisfies w.h.p.:

$$\min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x}^{(T)})} f(\mathbf{x}^{(T)} + \boldsymbol{\delta}) \geq \max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x} + \boldsymbol{\delta}) - \eta \,.$$

γ_T : Kernel-dependent mutual information quantity [Srinivas *et al.*'11]

$$\gamma_T = \max_{|A|\leq T} I(f; y_A)$$

- reduction in uncertainty about f
- bounds via submodular analysis

Theoretical guarantee

- Two distinct settings:
  - Bayesian setting (assume $f \sim \mathrm{GP}(\mathbf{0}, k(\cdot\,,\cdot))$)
  - Non-Bayesian setting (assume bounded norm $\|f\|_k$ in RKHS space)

Theorem: After running **StableOpt** for *T* rounds, if

$$T \gtrsim \frac{\gamma_T}{\eta^2}$$

then the point $\boldsymbol{x}^{(T)} = \tilde{\boldsymbol{x}}_{t^*}, \quad t^* = \operatorname*{argmax}_{t=1,\dots,T} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\boldsymbol{x}}_t)} \mathrm{lcb}_t(\tilde{\boldsymbol{x}}_t + \boldsymbol{\delta})$ satisfies w.h.p.:

$$\min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x}^{(T)})} f(\boldsymbol{x}^{(T)} + \boldsymbol{\delta}) \geq \max_{\boldsymbol{x} \in D} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\boldsymbol{x})} f(\boldsymbol{x} + \boldsymbol{\delta}) - \eta\ .$$

γ_T : Kernel-dependent mutual information quantity [Srinivas *et al.*'11]

$$\gamma_T = \max_{|A| \leq T} I(f; y_A)$$

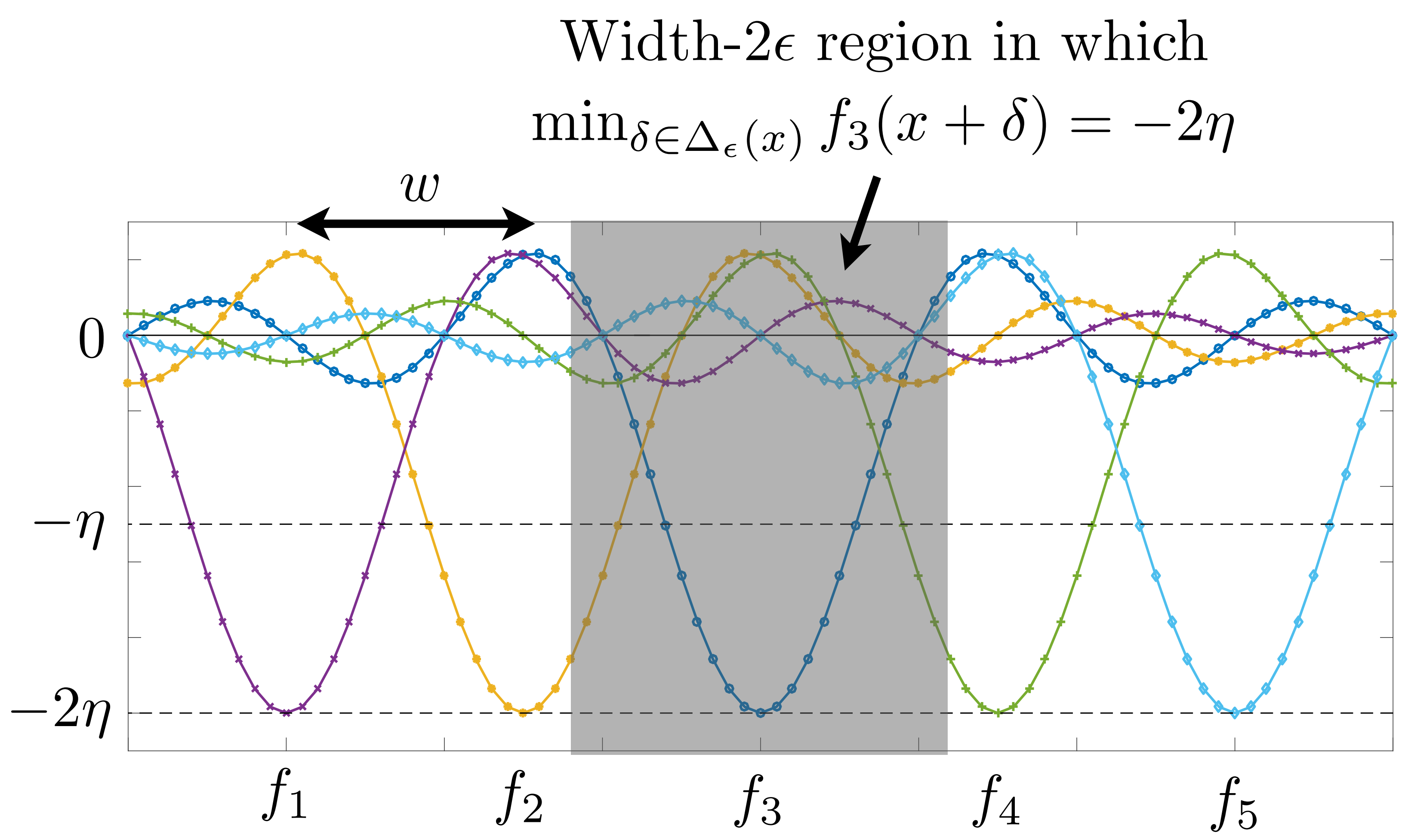
- reduction in uncertainty about f
- bounds via submodular analysis

- Special case: $T = O\big(\frac{1}{\eta^2}(\log\frac{1}{\eta})^{2d}\big)$ for squared exponential kernel in *d* dimensions

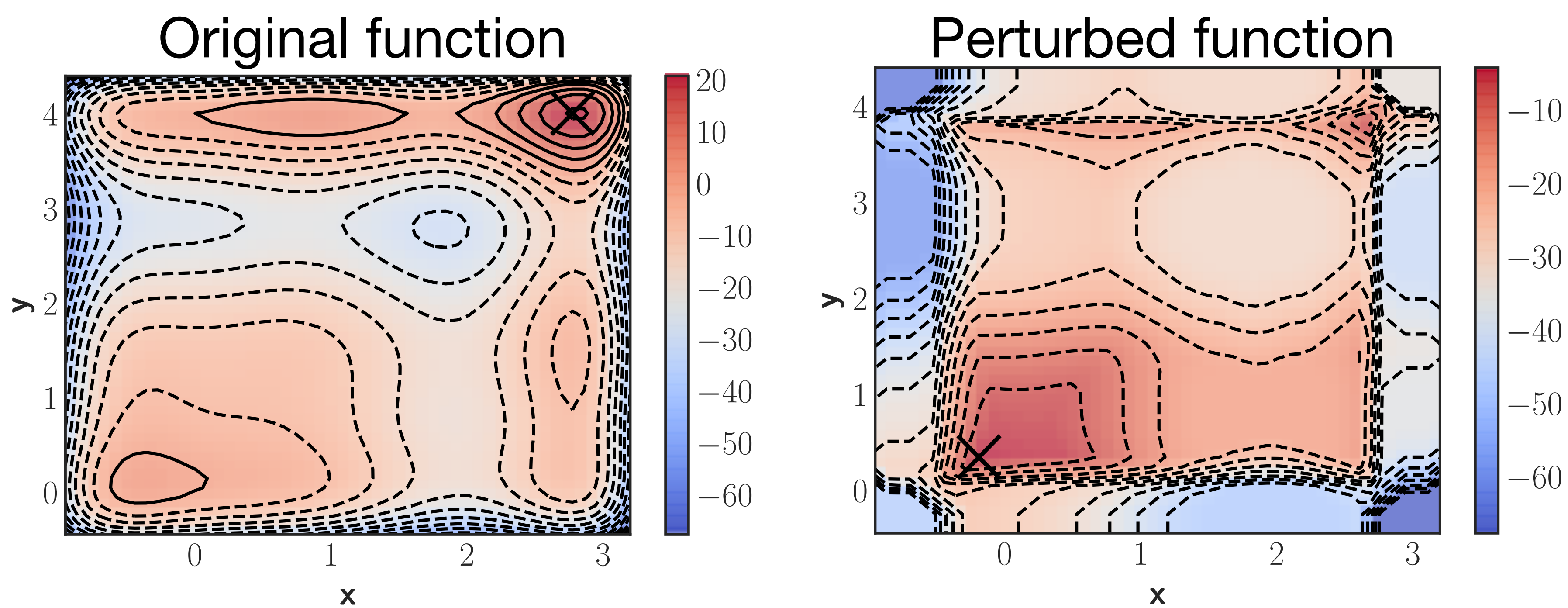
Lower bound

- Algorithm-independent lower bound (non-Bayesian setting):
  - For η regret squared exponential kernel requires $T = \Omega\left(\frac{1}{\eta^2}\left(\log \frac{1}{\eta}\right)^{d/2}\right)$

- Hard subset of functions used in proof (builds on [Scarlett *et al.* COLT'17])

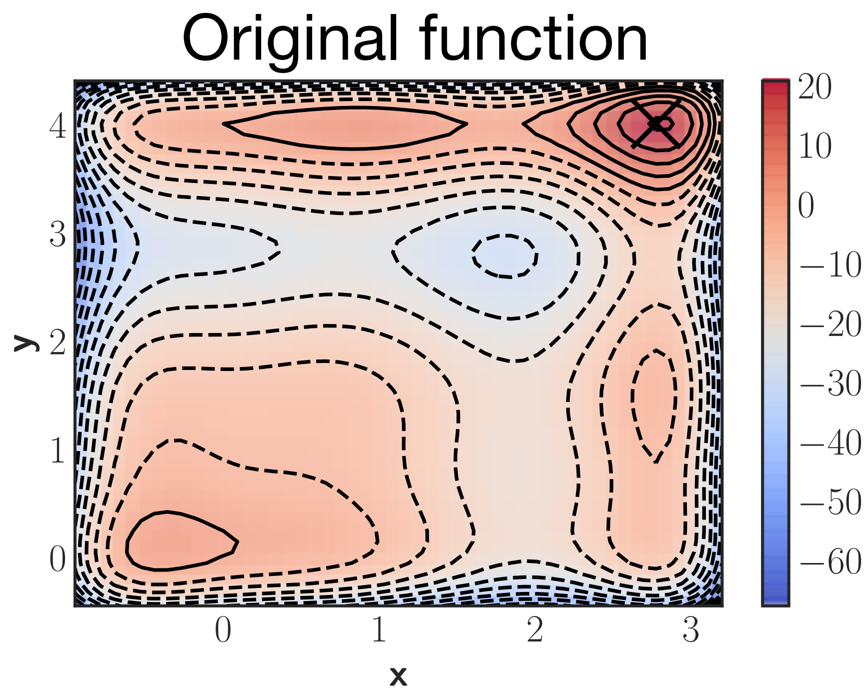


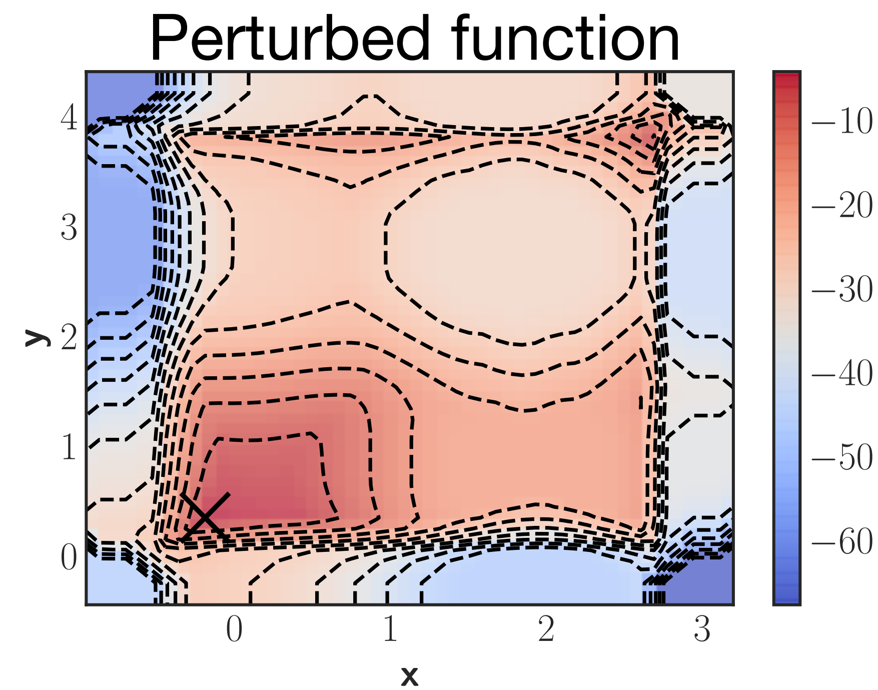
Numerical evidence



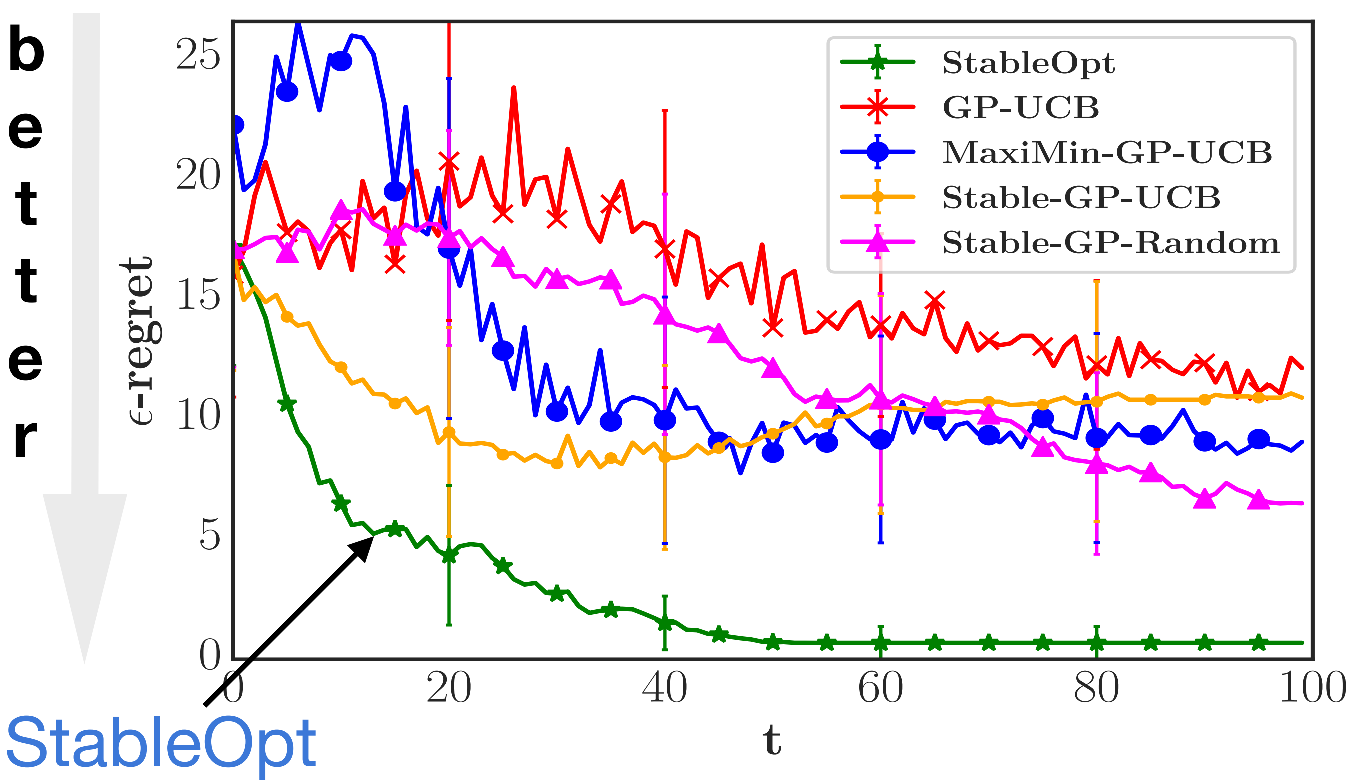
[Bertsimas *et al.*'10]

Numerical evidence





[Bertsimas *et al.*'10]

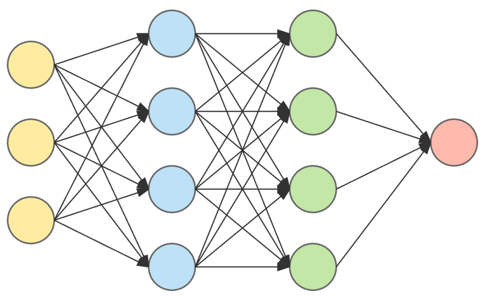


GP-UCB: standard BO representative

MaxiMin-GP-UCB: sampling and reporting $x_t = \arg\max_{x \in D} \min_{\delta \in \Delta_\epsilon(x)} \mathrm{ucb}_{t-1}(x + \delta)$

Stable-GP-Random: random sampling

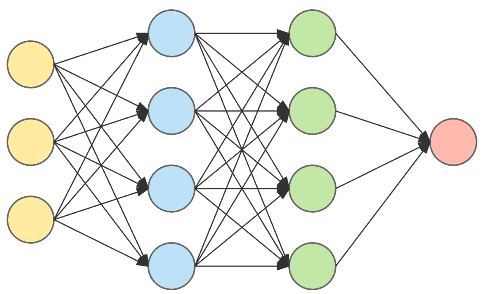
Variations



Robustness to unknown parameters:

- Goal: Choose $\boldsymbol{x}$ robust to different $\boldsymbol{\theta}$, $\max_{x \in D} \min_{\boldsymbol{\theta} \in \Theta} f(\boldsymbol{x}, \boldsymbol{\theta})$
- Application: Tuning hyperparameters robust to different data types

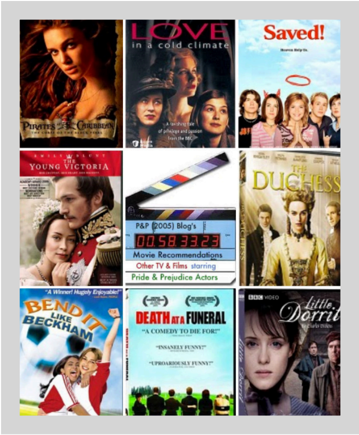
Variations



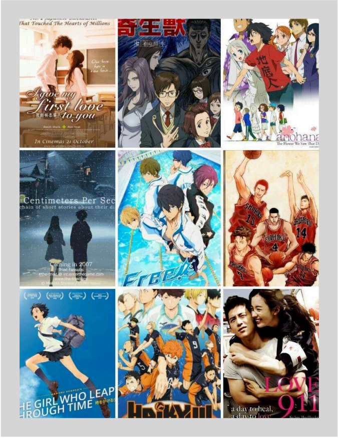
Robustness to unknown parameters:

- Goal: Choose $\boldsymbol{x}$ robust to different $\boldsymbol{\theta}$, $\max\limits_{x \in D} \min\limits_{\theta \in \Theta} f(\boldsymbol{x}, \boldsymbol{\theta})$
- Application: Tuning hyperparameters robust to different data types

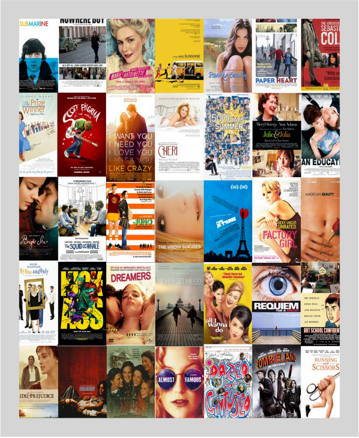
Robust group identification: Input space is partitioned into groups



G_1



G_2



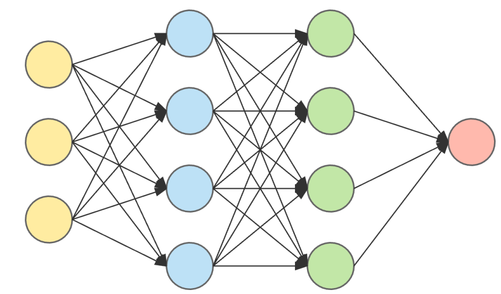
G_k

- Goal: Identify the group with the highest worst-case function value

$$\max_{G \in \mathcal{G}} \min_{x \in G} f(\boldsymbol{x})$$

- Application: Robust group movie recommendation

Variations

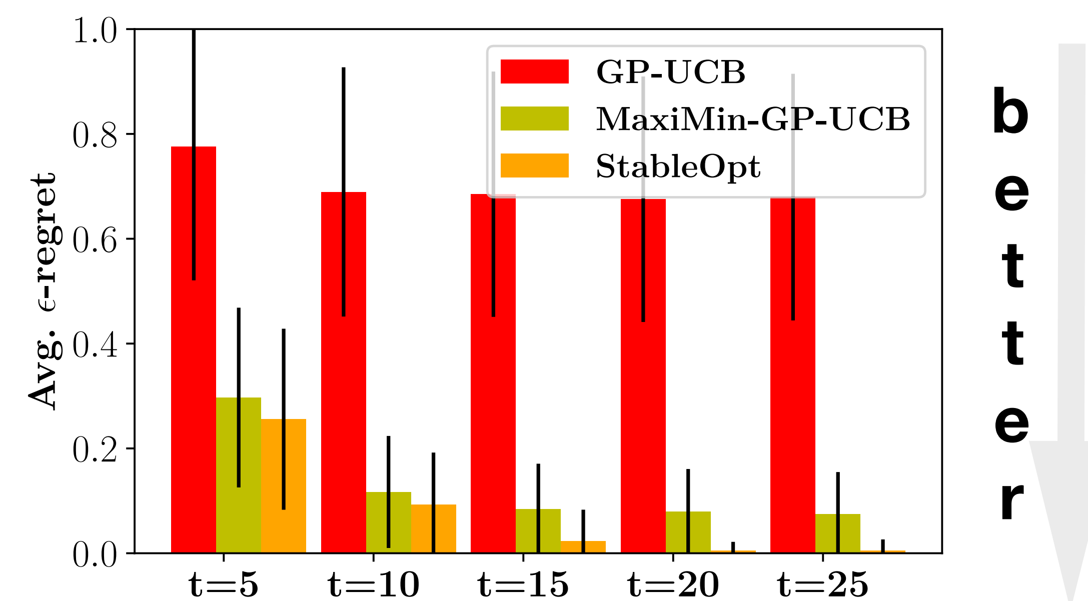


Robustness to unknown parameters:

- Goal: Choose $\boldsymbol{x}$ robust to different $\boldsymbol{\theta}$, $\max_{x \in D} \min_{\theta \in \Theta} f(\boldsymbol{x}, \boldsymbol{\theta})$
- Application: Tuning hyperparameters robust to different data types

Robust group identification: Input space is partitioned into groups

G_1 G_2 G_k



- Goal: Identify the group with the highest worst-case function value

$$\max_{G \in \mathcal{G}} \min_{x \in G} f(\boldsymbol{x})$$

- Application: Robust group movie recommendation

Robust robot pushing

Task: Find a good pre-image for pushing the object to the target location

[Wang *et al.* ICML'17]

$f(r_x, r_y, r_t) =$ distance(pushed object, target location)

$r_x, r_y \in [-5,5]$ $r_t \in [1,30]$

initial robot location pushing duration

Challenge: uncertainty of the precise target location

Goal: robustness vs. different potential locations

$$\boldsymbol{r} \in \arg\max_{r \in D} \min_{i \in [m]} f_i(\boldsymbol{r})$$

Robust robot pushing

Task: Find a good pre-image for pushing the object to the target location

[Wang *et al.* ICML'17]

$f(r_x, r_y, r_t) =$ distance(pushed object, target location)

$r_x, r_y \in [-5,5]$ $r_t \in [1,30]$

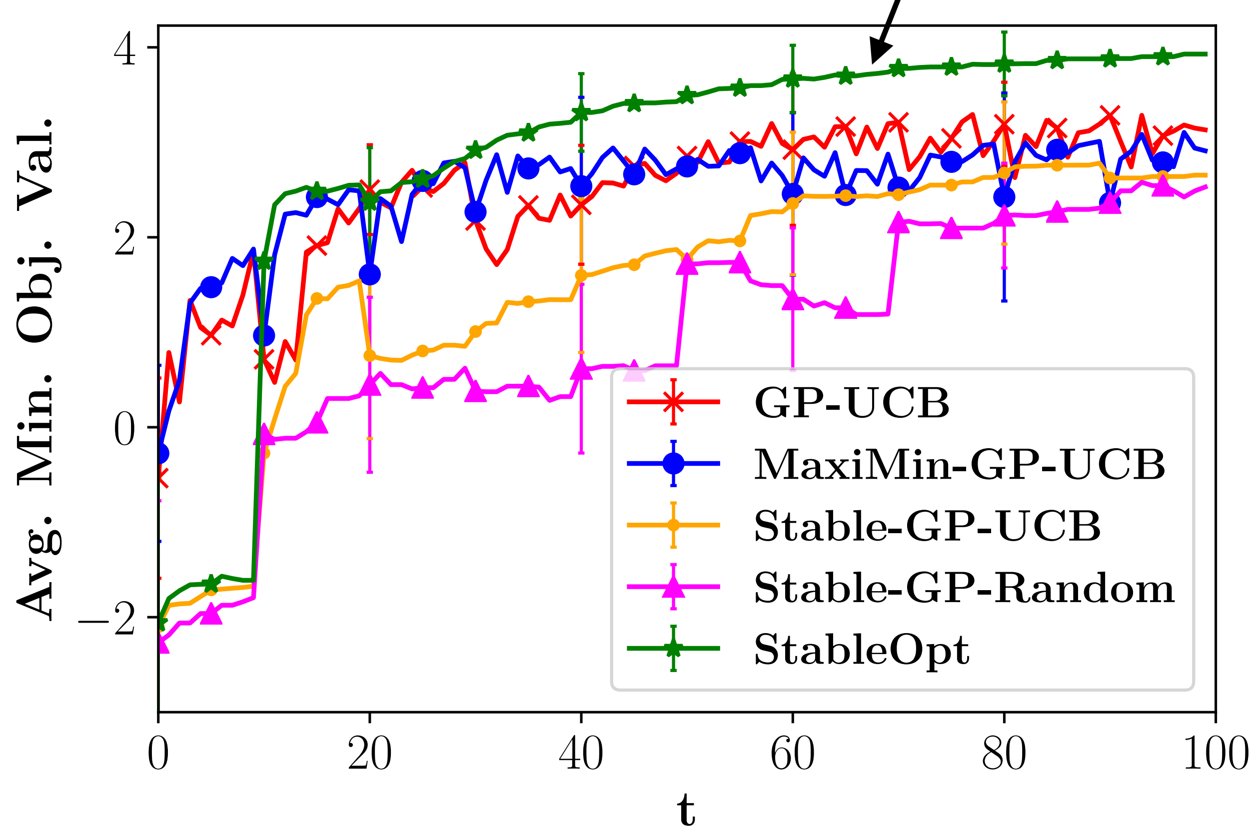
initial robot location   pushing duration

StableOpt

Challenge: uncertainty of the precise target location

Goal: robustness vs. different potential locations

$$\boldsymbol{r} \in \arg\max_{r \in D} \min_{i \in [m]} f_i(\boldsymbol{r})$$



better

Conclusion & Discussion

- Robust requirements in sample efficient learning
- Key steps:
  - Use **confidence bounds** to effectively prune the space
  - Perform **optimistic robust optimization**
  - **Explore pessimistically** to reduce uncertainty
- Future and current work:
  - **(Coming up!)** Various other robust optimization objectives
  - **(New!)** Repeated games with unknown reward function vs. adversarial players [P. G. Sessa, I.B., M. Kamgarpour, A. Krause, *NeurIPS'19*]
  - Robust Online Learning in Markov Decision Processes

Thank you! Acknowledgements

Adversarially Robust Optimization with Gaussian Processes

Ilija Bogunovic
LIONS, EPFL
ilija.bogunovic@epfl.ch

Jonathan Scarlett
National University of Singapore
scarlett@comp.nus.edu.sg

Stefanie Jegelka
MIT CSAIL
stefje@mit.edu

Volkan Cevher
LIONS, EPFL
volkan.cevher@epfl.ch

http://ilijabogunovic.com/

https://las.inf.ethz.ch/

Collaborators: P.G. Sessa (ETHZ), J. Kirschner (ETHZ)

Senior Collaborators: J. Scarlett (NUS), S. Jegelka (MIT), M. Kamgarpour (ETHZ), V. Cevher (EPFL), A. Krause (ETHZ)