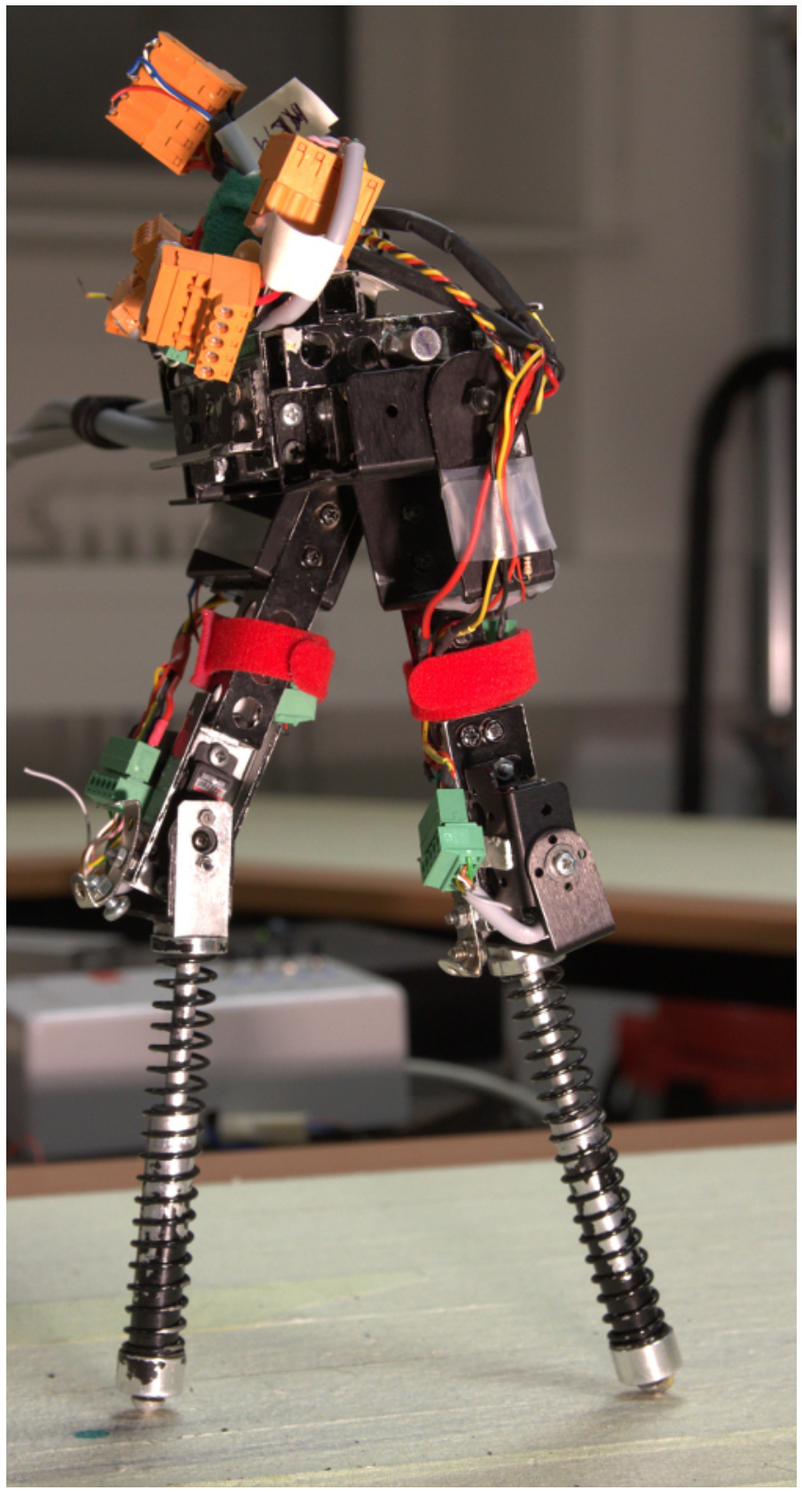
Bayesian Optimization and How to Scale It Up

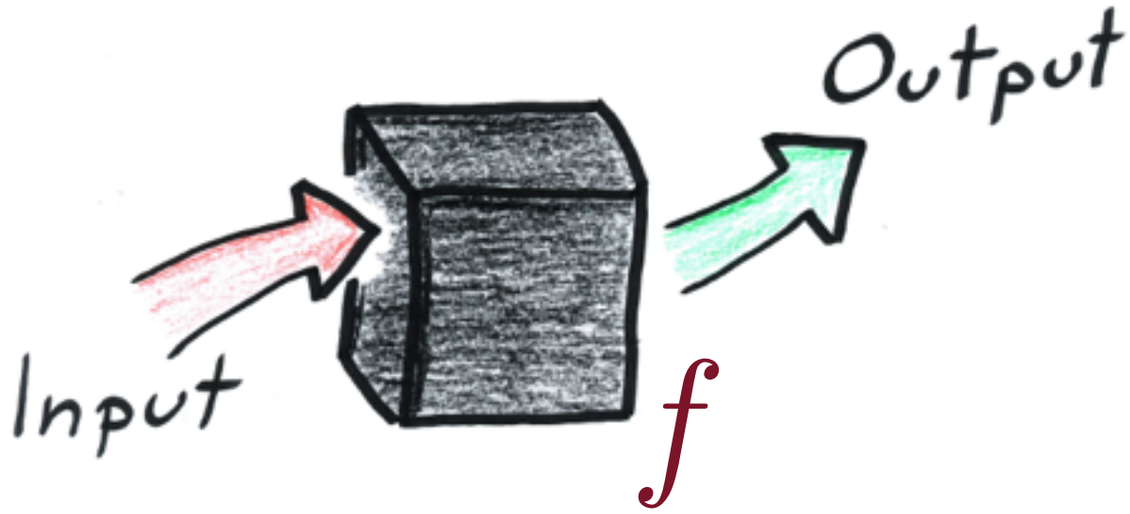
Zi Wang

CS Colloquium, USC

Joint work with Stefanie Jegelka, Bolei Zhou, Chengtao Li, Clement Gehring (MIT) and Pushmeet Kohli (DeepMind)

Blackbox Function Optimization



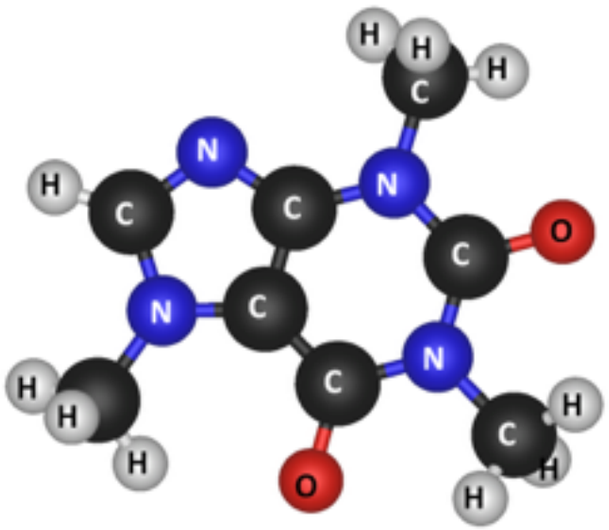


Goal:

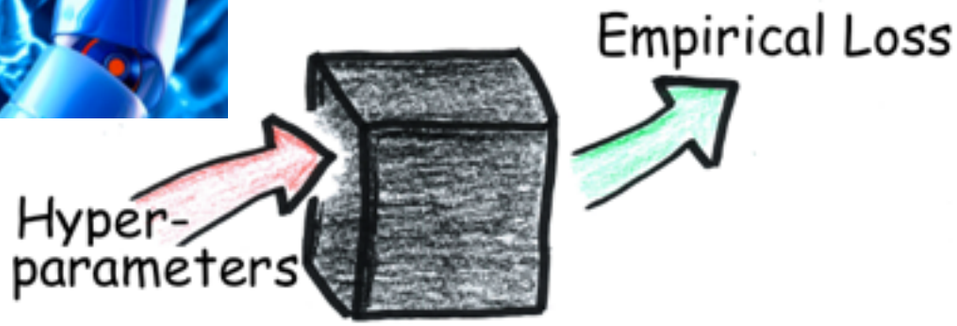
$$x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}}\ f(x)$$

(Calandra et al., 2015)

Blackbox Function Optimization

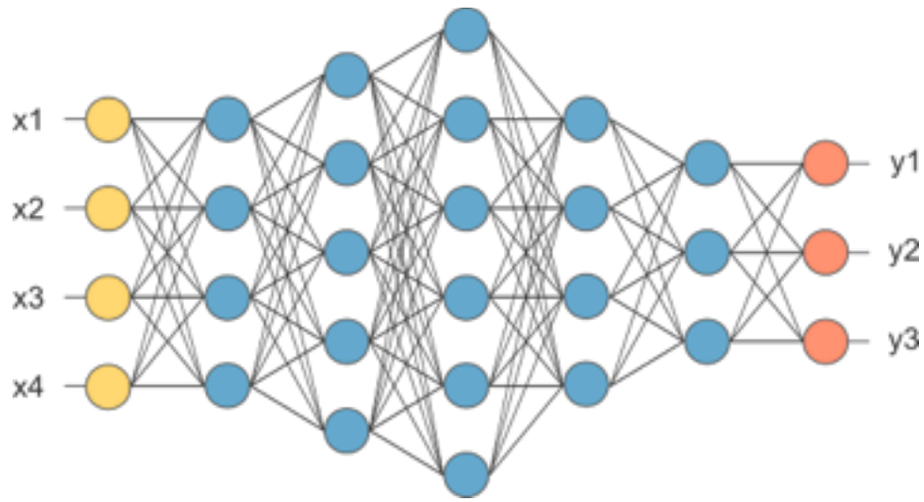


Goal: $x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}}\ f(x)$



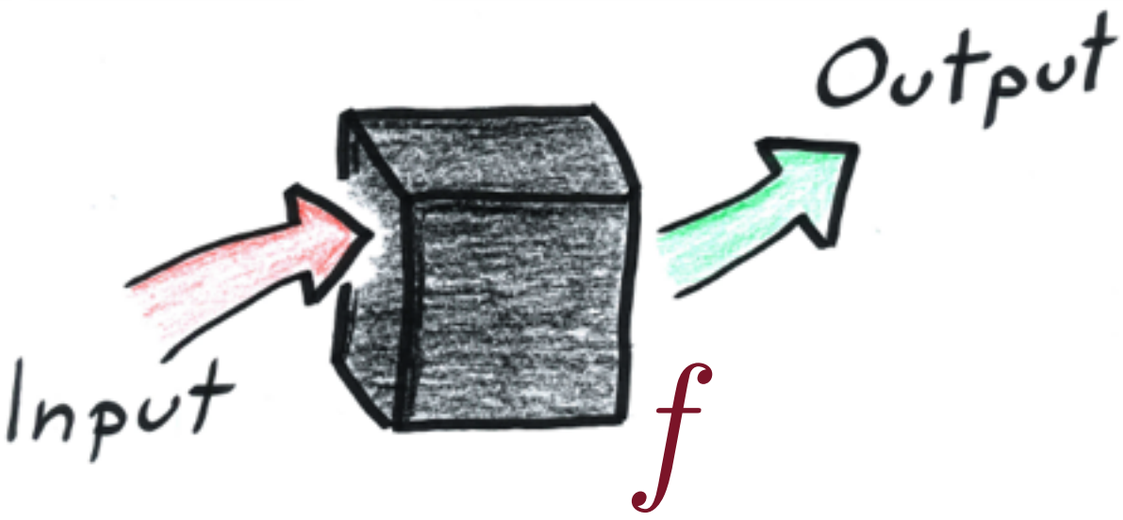
Challenges:

- f is expensive to evaluate
- f is non-convex
- no gradient information
- evaluations can be noisy



(Snoek et al., 2012;Gonzalez et al., 2015; Hernández-Lobato et al., 2017)

Blackbox Function Optimization

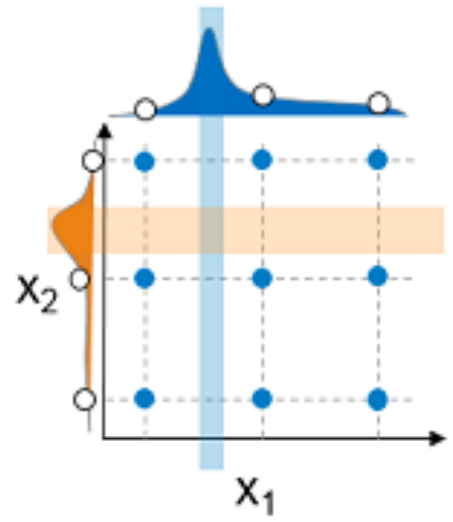


Goal: $x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} \ f(x)$

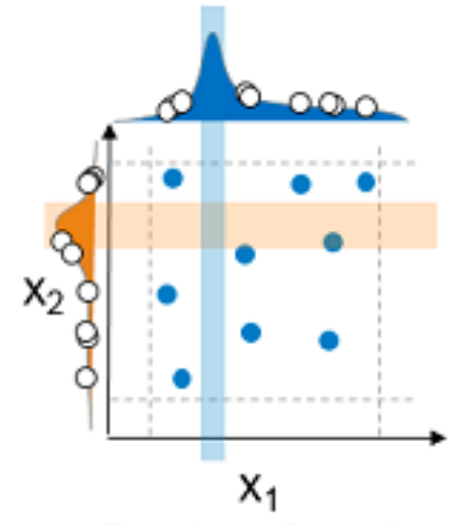
Challenges:

- f is expensive to evaluate
- f is non-convex
- no gradient information
- evaluations can be noisy

Grid search?



Random search?



Many evaluations are wasted!

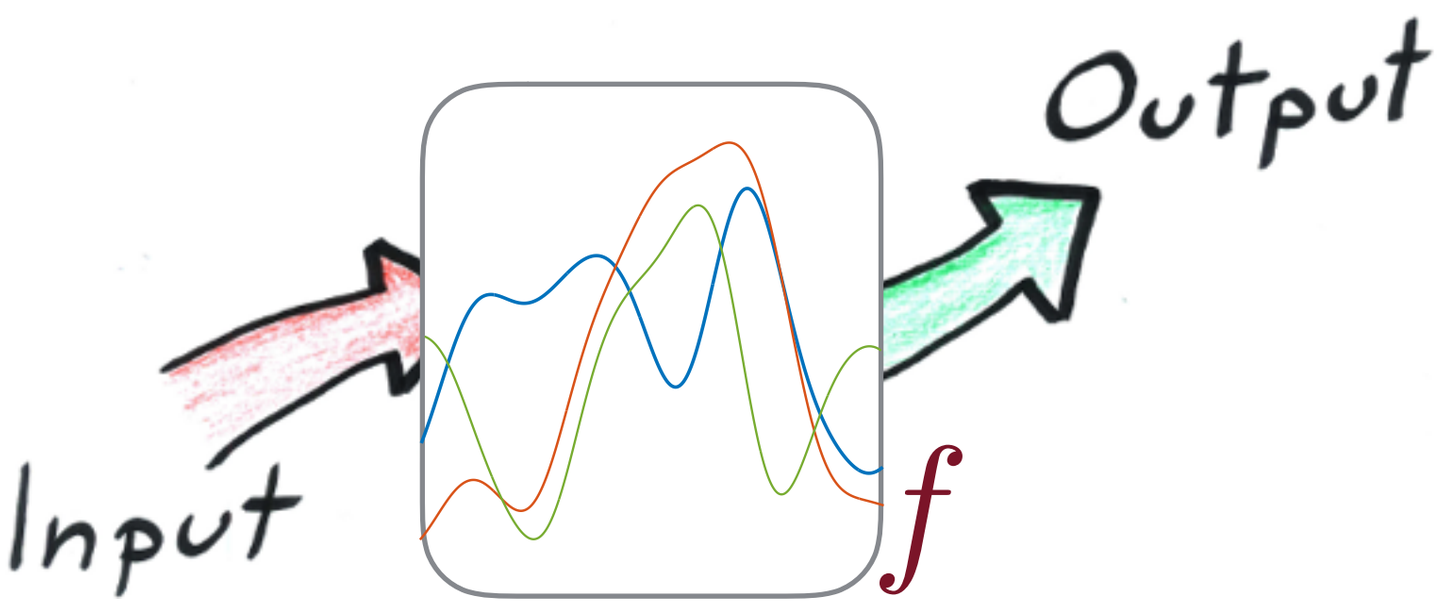
(Koch, 2016)

Bayesian Optimization

Idea: build a probabilistic model of the function f

LOOP

- choose new query point(s) to evaluate
  decision criterion: acquisition function $\alpha_t(\cdot)$
- update model



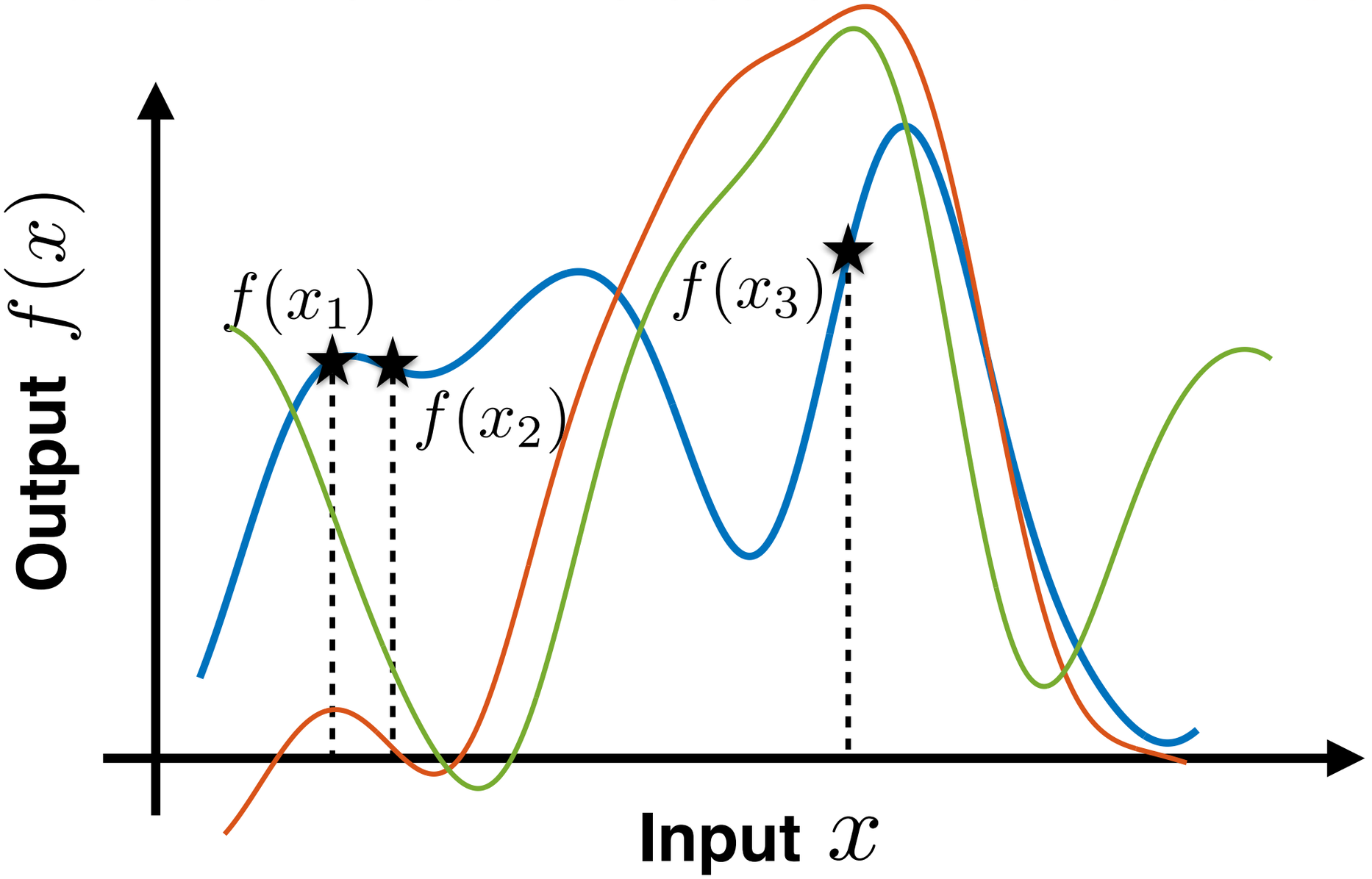
$$x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}}\ f(x)$$

$$x_t = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}}\ \alpha_t(x)$$

$$t = 1, \cdots, T$$

Gaussian Processes (GPs)

- probability distribution over functions
- any finite set of function values is a multi-variate Gaussian



Gaussian Processes (GPs)

- probability distribution over functions
- any finite set of function values is a multi-variate Gaussian
- kernel function $k(\cdot,\cdot)$; mean function $\mu(\cdot)$

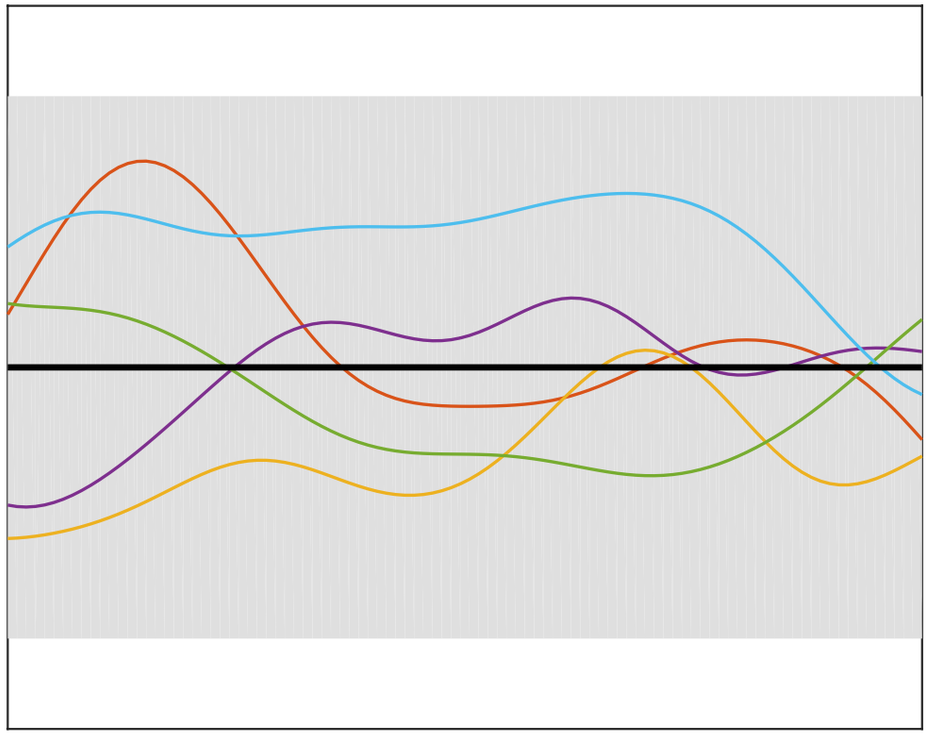
$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1), & \cdots, & k(x_1,x_n) \\ \vdots, & & \vdots \\ k(x_n,x_1), & \cdots, & k(x_n,x_n) \end{bmatrix} \right)$$

- function $f \sim GP(\mu, k)$; observe noisy output at x_τ

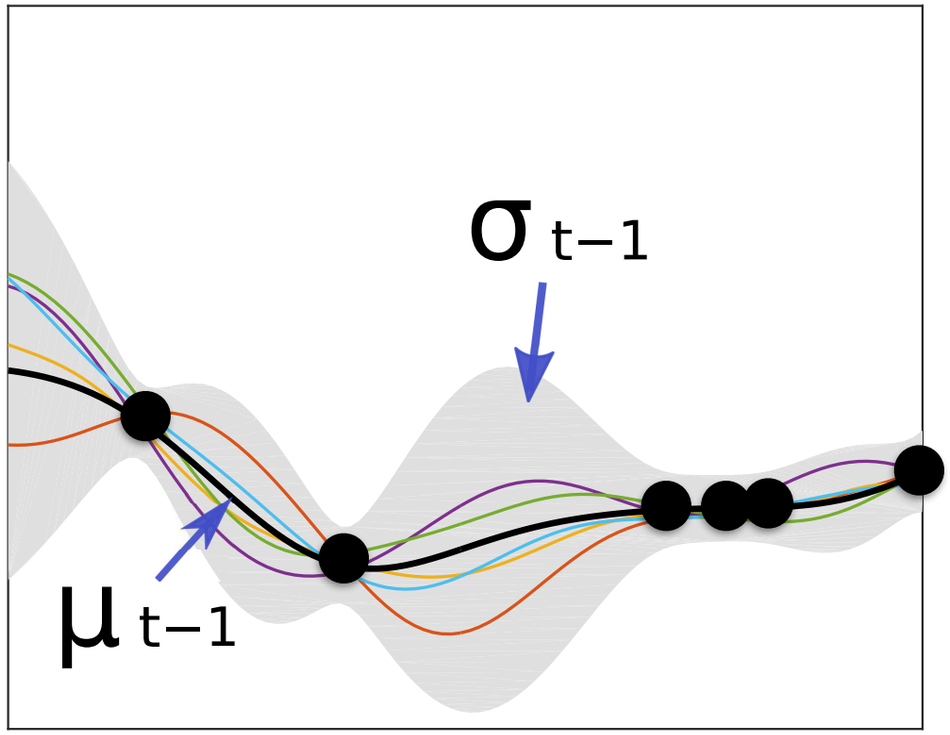
$$y_\tau = f(x_\tau) + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Gaussian Processes (GPs)

Samples from the prior



Samples from the posterior



Given observations $D_t = \{(x_\tau, y_\tau)\}_{\tau=1}^{t-1}$, predict posterior mean and variance in closed form via conditional Gaussian

$$\mu_{t-1}(x) = k_{t-1}(x)^{\mathrm{T}}(K_{t-1} + \sigma^2 I)^{-1} y_{t-1}$$

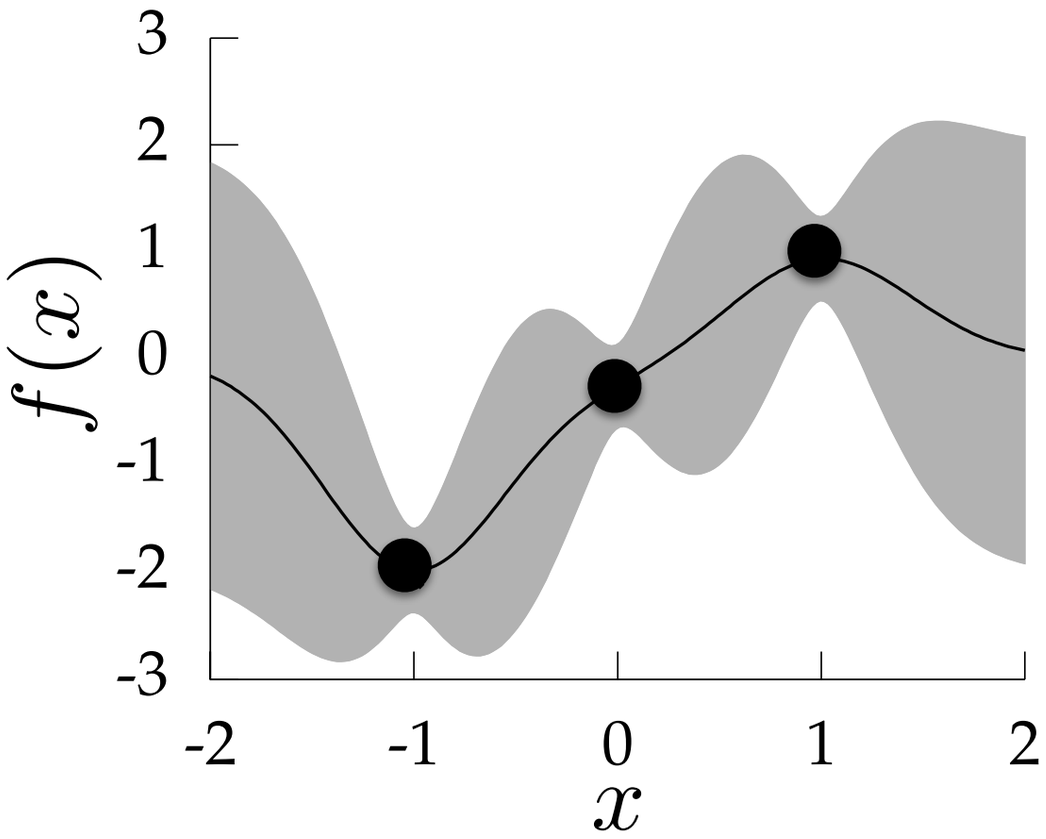
$$\sigma_{t-1}(x)^2 = k(x,x) - k_{t-1}(x)^{\mathrm{T}}(K_{t-1} + \sigma^2 I)^{-1} k_{t-1}(x)$$

Bayesian Optimization

Idea: build a probabilistic model of the function f

LOOP

- choose new query point(s) to evaluate
  decision criterion: acquisition function $\alpha_t(\cdot)$
- update model

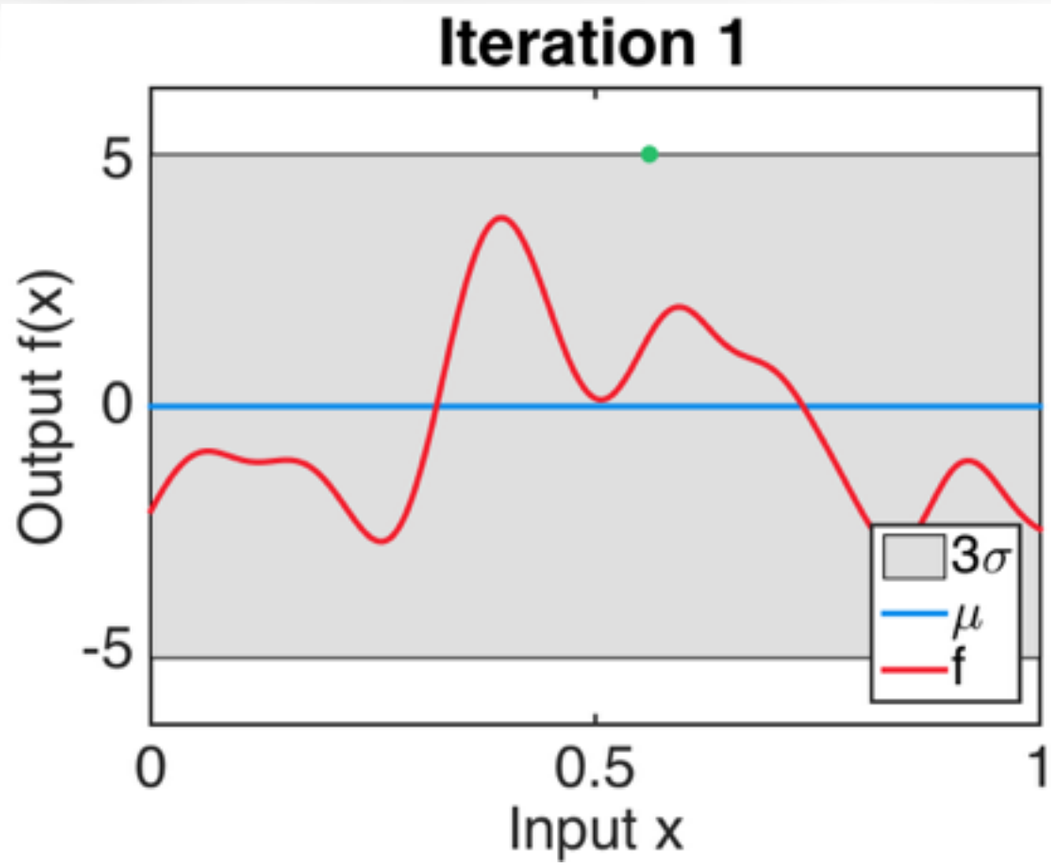


$$x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}}\ f(x)$$

$$x_t = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}}\ \alpha_t(x)$$

$$t = 1, \cdots, T$$

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

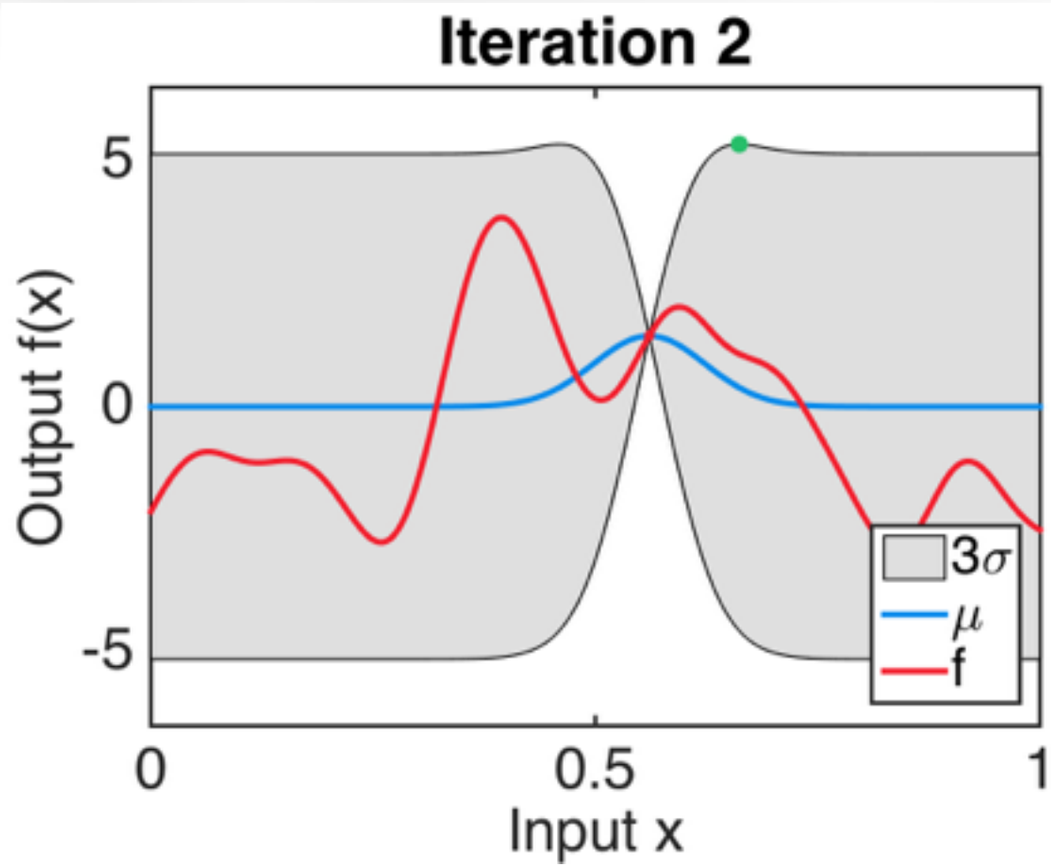
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

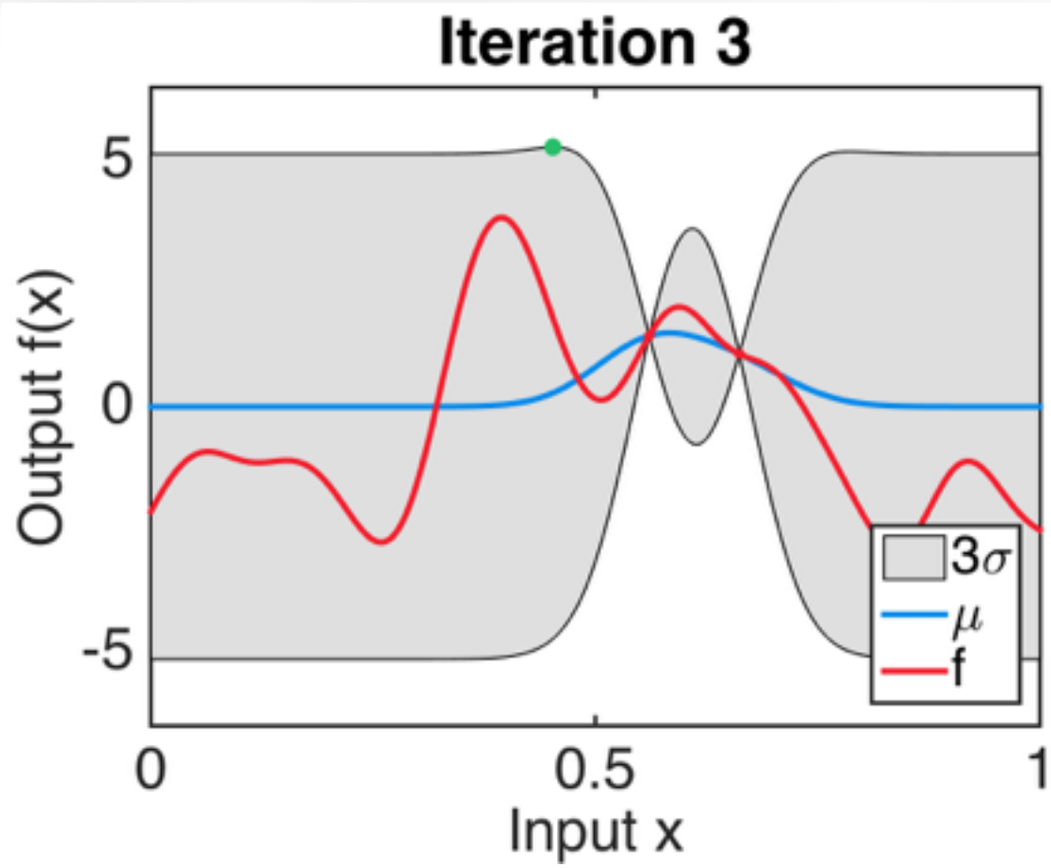
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

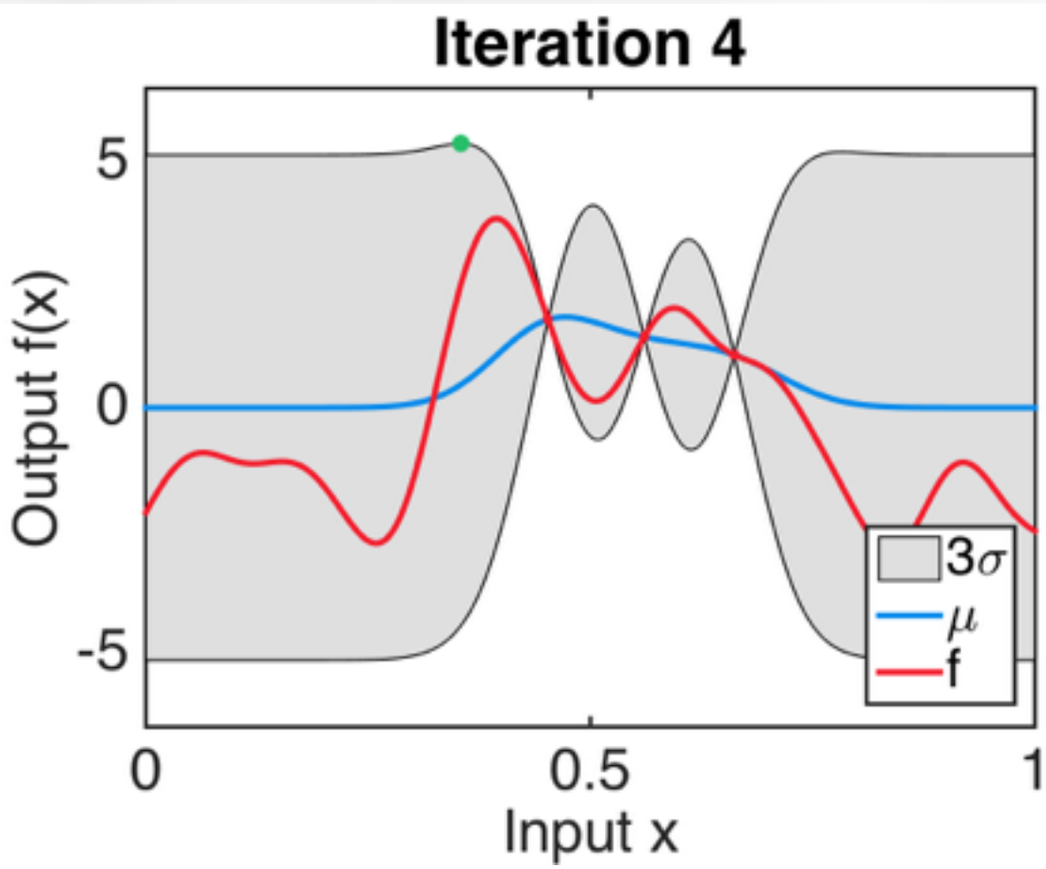
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

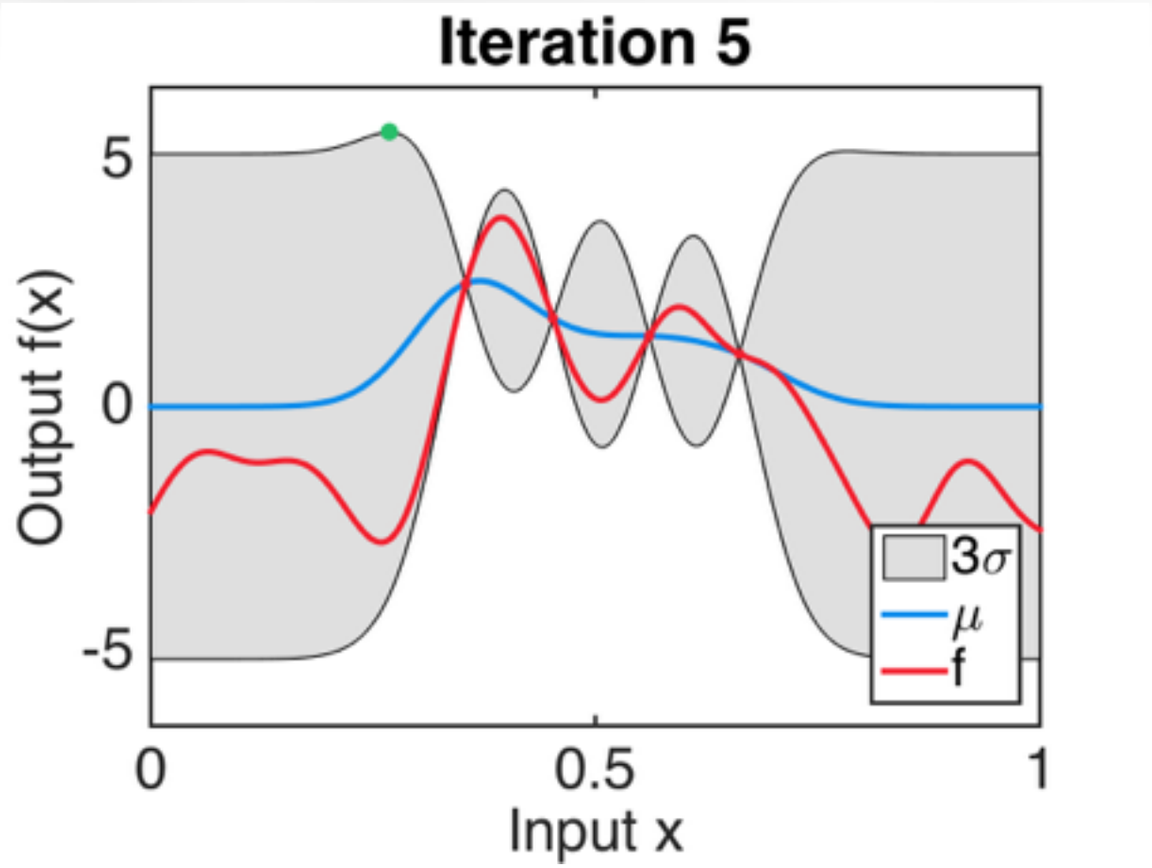
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

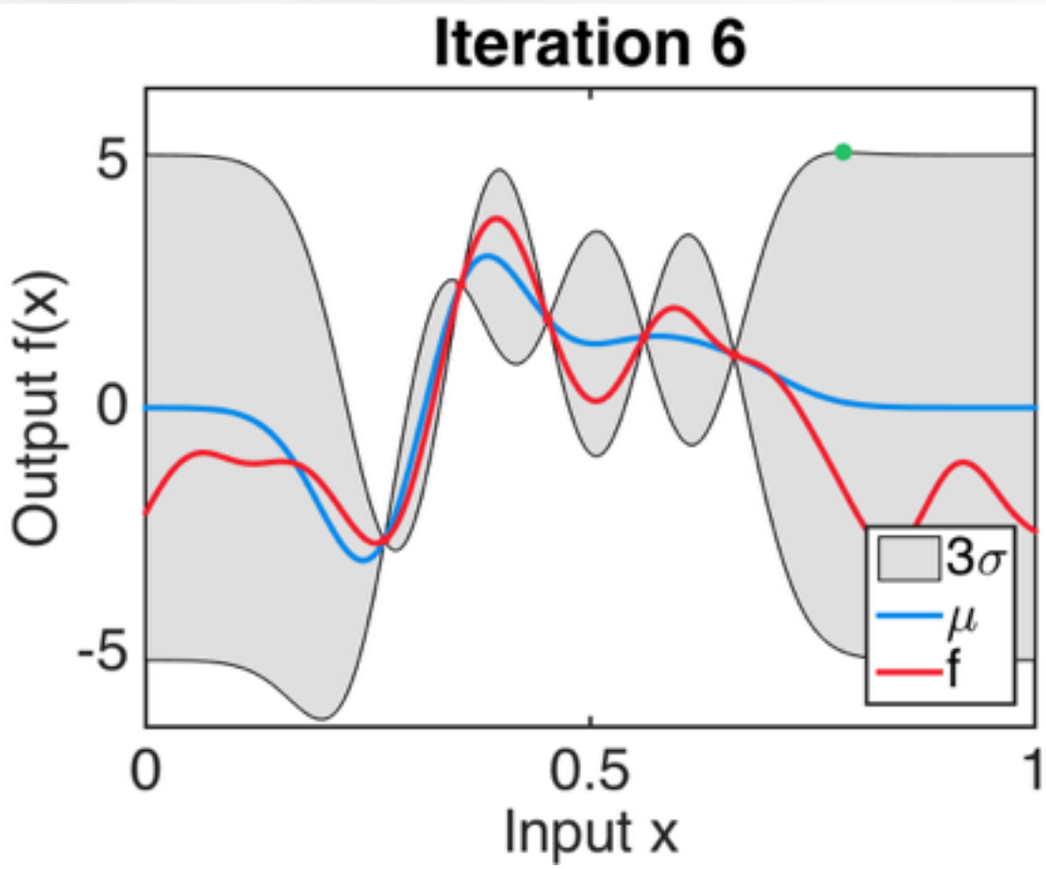
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

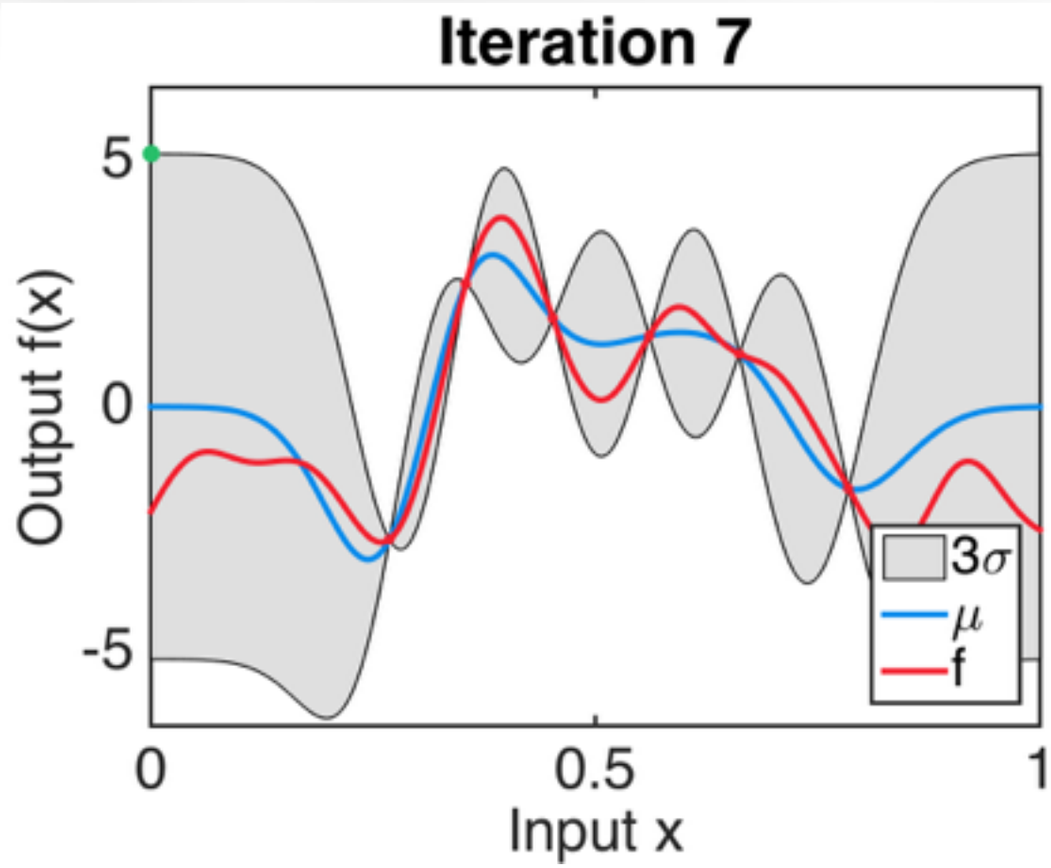
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

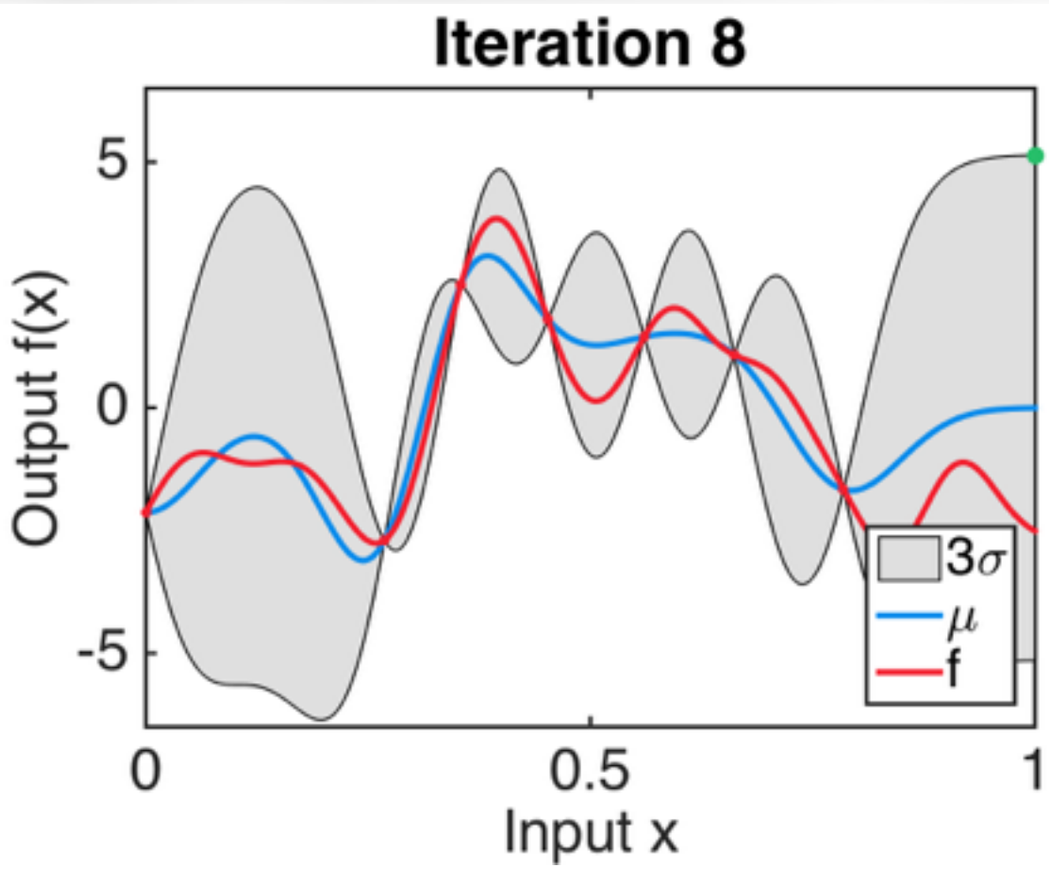
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

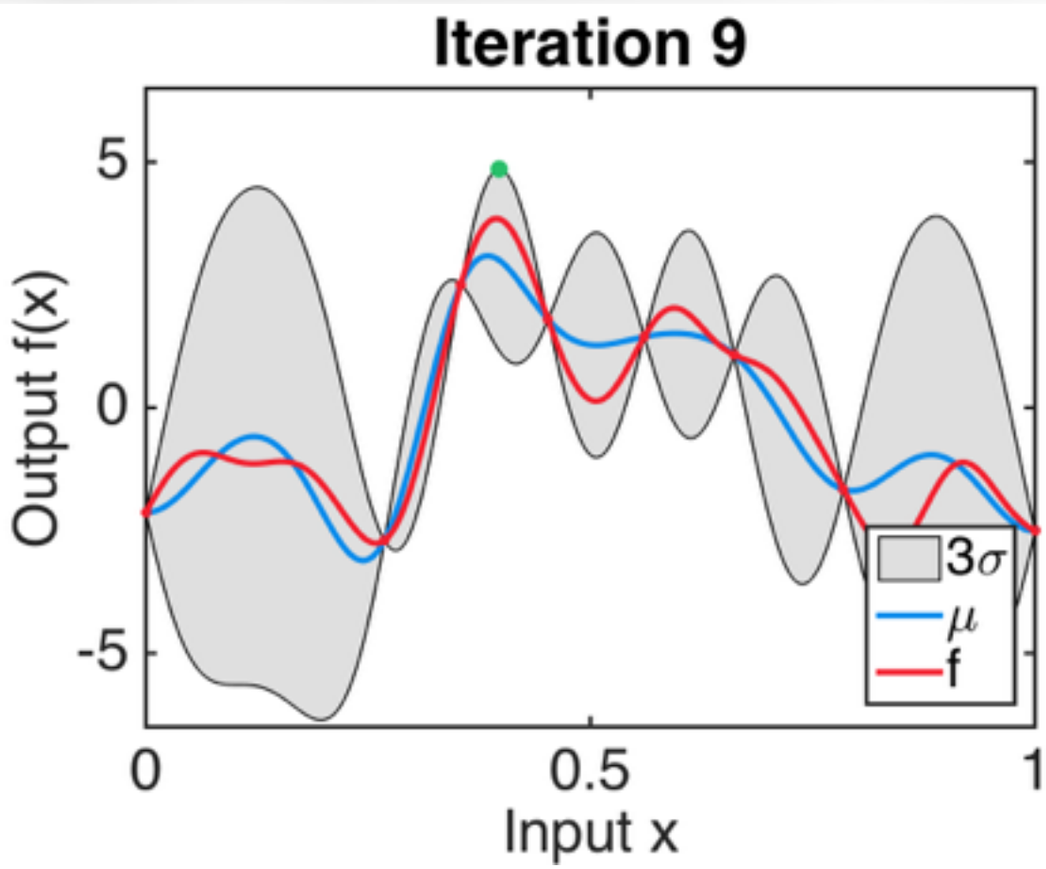
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

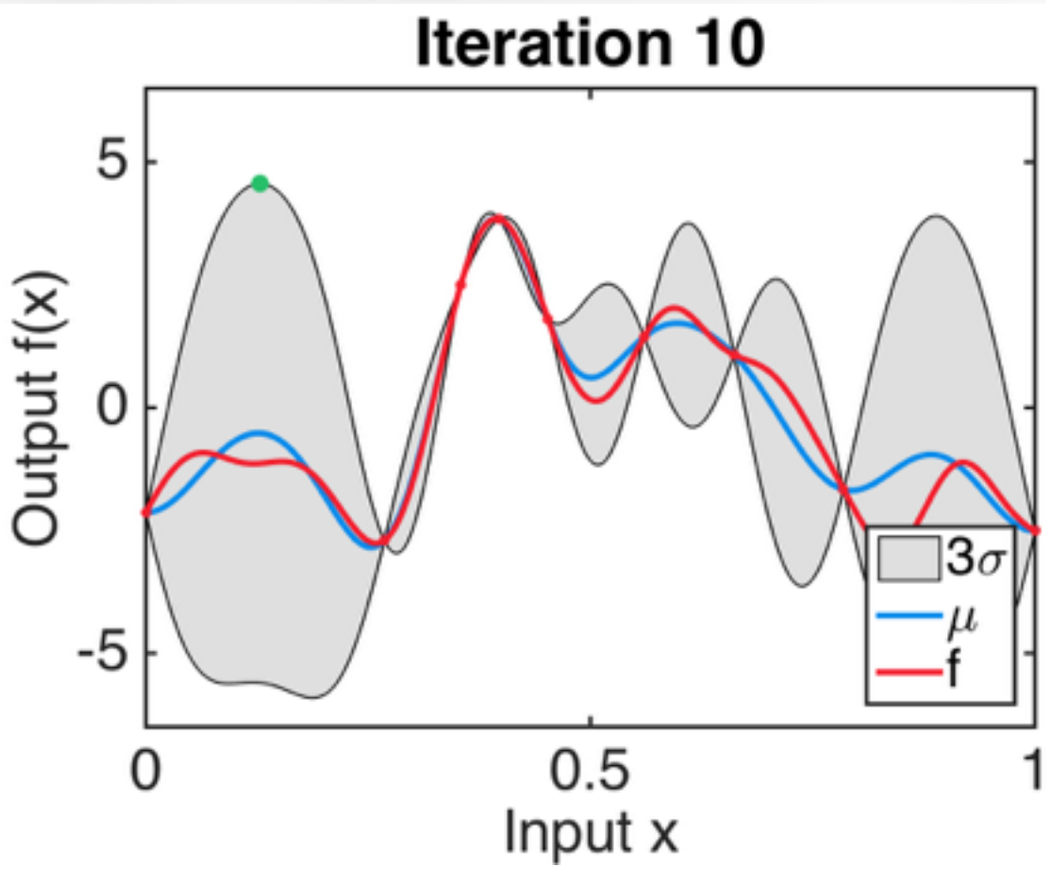
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

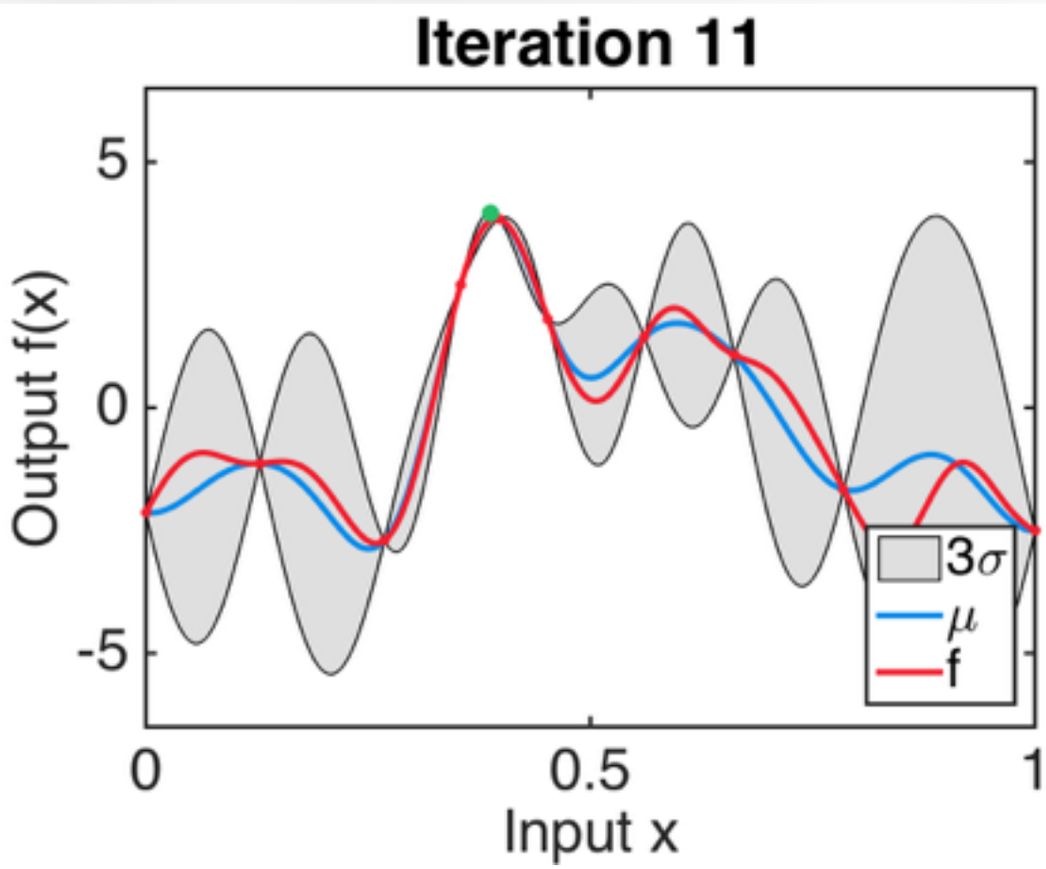
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

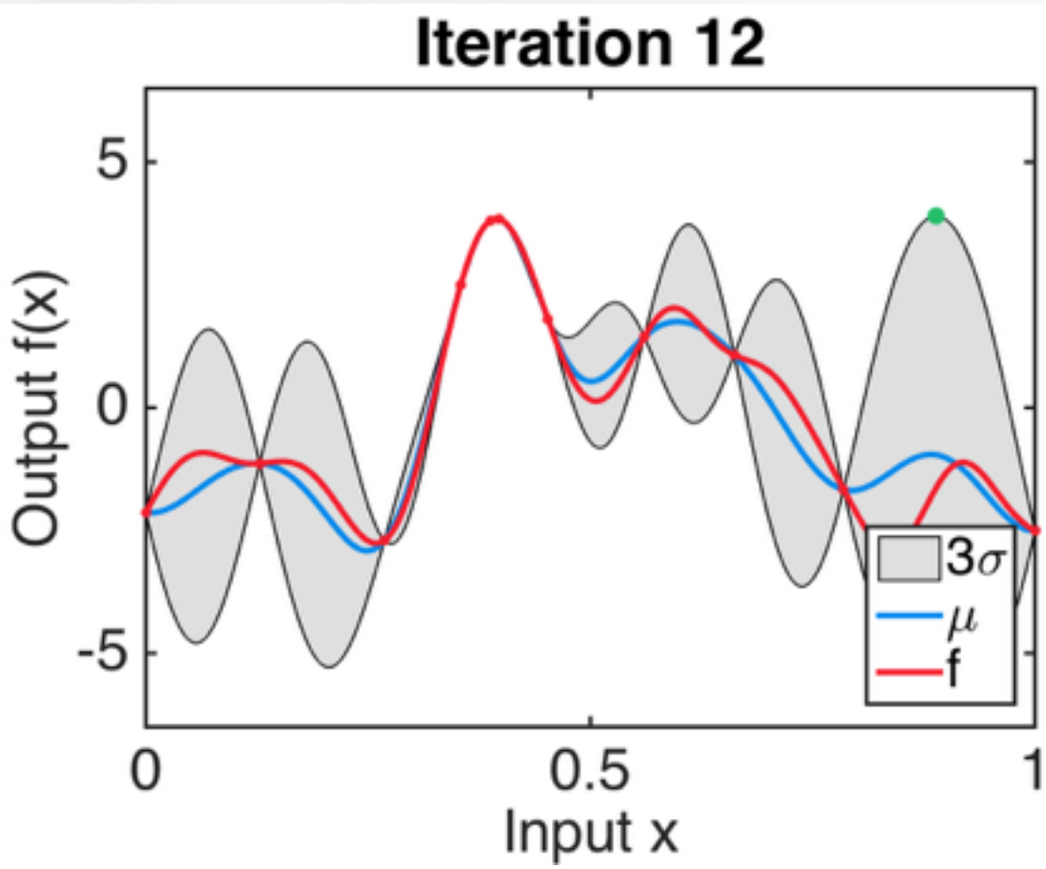
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

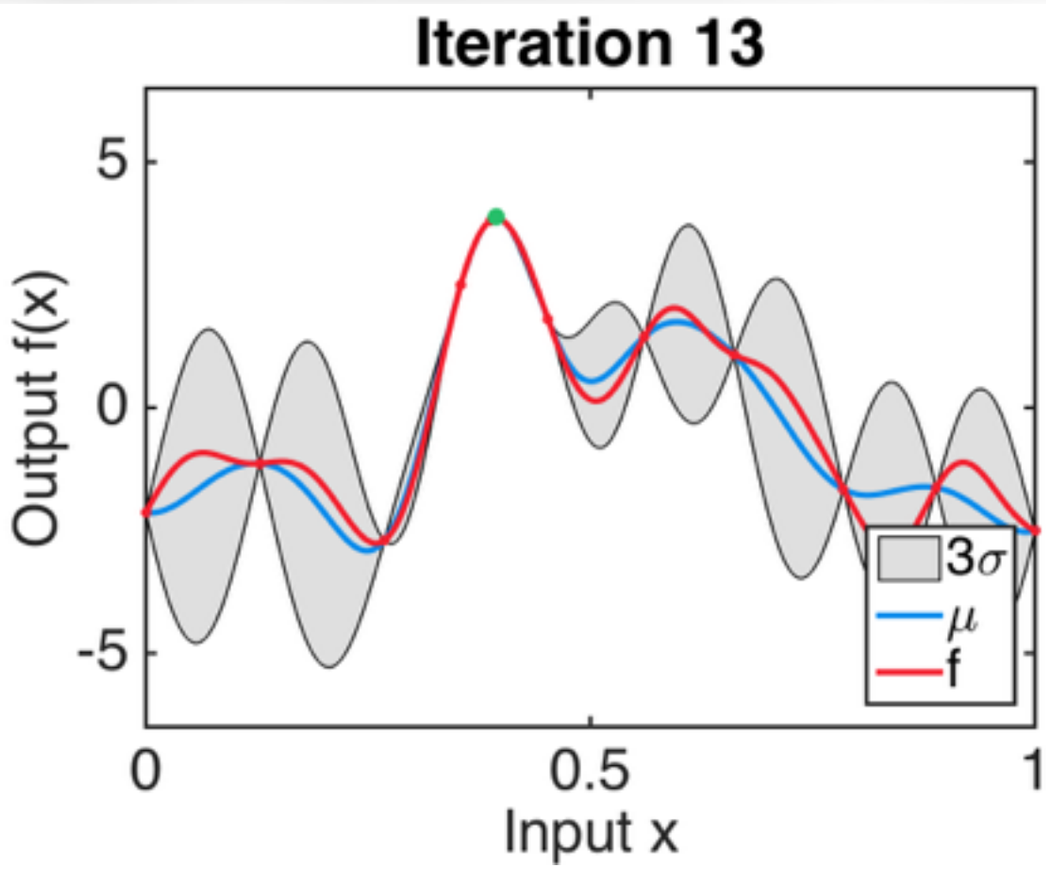
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

GP-UCB: an example of Bayesian Optimization



Prior: $f \sim GP(\mu, k)$

At iteration t,

β is hard to tune in practice

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

(Auer, 2002; Srinivas et al., 2010)

Challenges in Bayesian Optimization

LOOP

- choose new query point(s) to evaluate
  decision criterion: acquisition function
- update model

- Effective and efficient acquisition function
- High dimensional input space
- Large-scale observations
- Parallel evaluations

Roadmap

- Better acquisition function

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

- Scaling up input dimensions

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

- Scaling up observations & parallel queries

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

Roadmap

- Better acquisition function

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

- Scaling up input dimensions

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

- Scaling up observations & parallel queries

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

Entropy Search and Predictive Entropy Search

$$\underset{x_t \in \mathfrak{X}}{\text{maximize}}\ \alpha_t(x_t) \qquad t = 1, \cdots, T$$

Point to query

Location of global optimum

Observed Data

$$\alpha_t(x) = I(\{x, y\}; x_* \mid D_t)$$

$$I(a; b) = H(a) - H(a|b)$$
$$= H(b) - H(b|a)$$

$$= H(p(x_* \mid D_t)) - \mathbb{E}_y[H(p(x_* \mid D_t \cup \{x, y\}))]$$

$$= H(p(y \mid D_t, x)) - \mathbb{E}_{x_*}[H(p(y \mid x_*, D_t, x))]$$

x^* can be high-dimensional: $\alpha_t(x)$ costly to estimate!

(Hennig & Schuler, 2012; Hernandez-Lobato et al., 2014)

Max-value Entropy Search

Point to query

Observed Data

~~$\alpha_t(x) = I(\{x, y\}; x_* \mid D_t)$~~ ~~Location of global optimum~~

Input space

Output space

$$\alpha_t(x) = I(\{x, y\}; y_* \mid D_t)$$ **1-D Global max-value**

$$= H(p(y \mid D_t, x)) - \mathbb{E}_{y_*}\left[H(p(y \mid y_*, D_t, x))\right]$$

Gaussian

Truncated Gaussian

$y \leq y_*$

$$\approx \frac{1}{K} \sum_{y_* \in Y_*} \left[\frac{\gamma_{y_*}(x)\psi(\gamma_{y_*}(x))}{2\Psi(\gamma_{y_*}(x))} - \log(\Psi(\gamma_{y_*}(x))) \right]$$

Something Closed-form

How to sample y_* ?

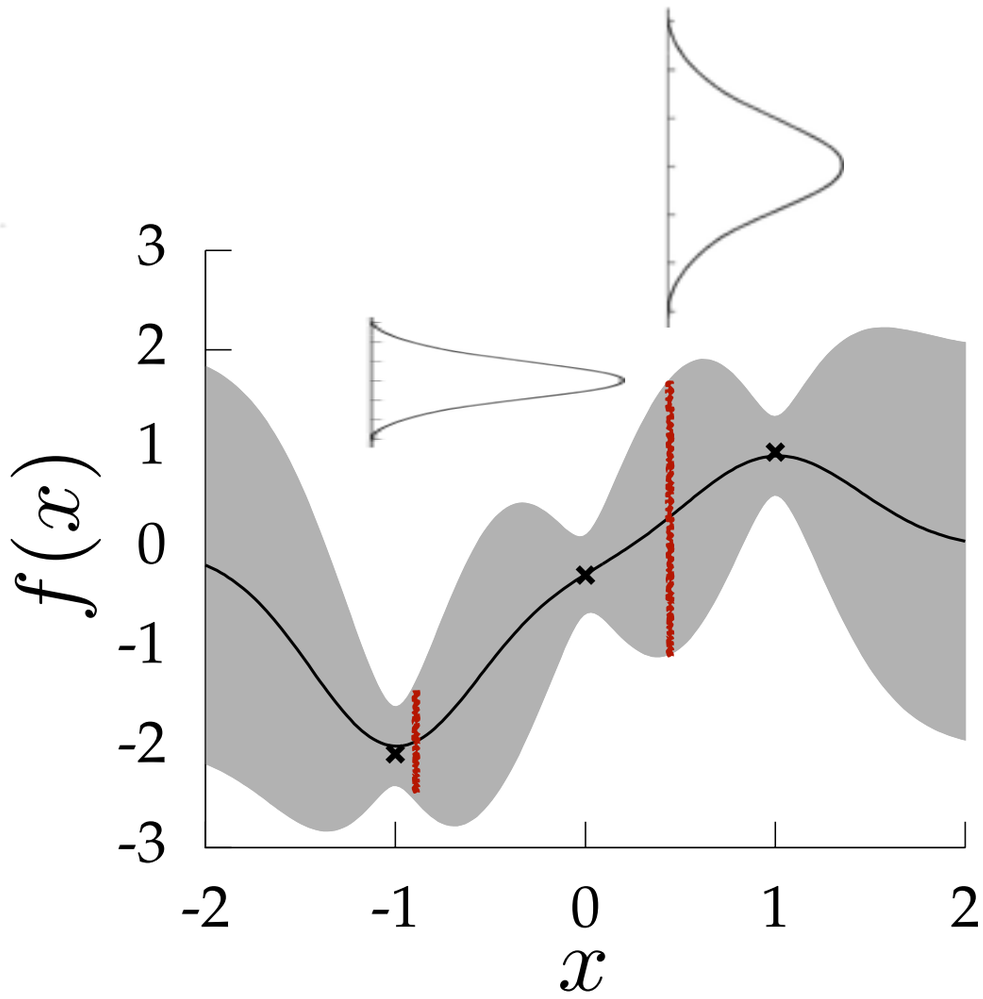
(Wang&Jegelka, 2017; Hoffman&Zoubin, 2015)

Sample y_* with a Gumbel Distribution

Intuition: each f(x) is a Gaussian

Fisher-Tippett-Gnedenko Theorem

The maximum of a set of i.i.d. Gaussian variables is asymptotically described by a Gumbel distribution.

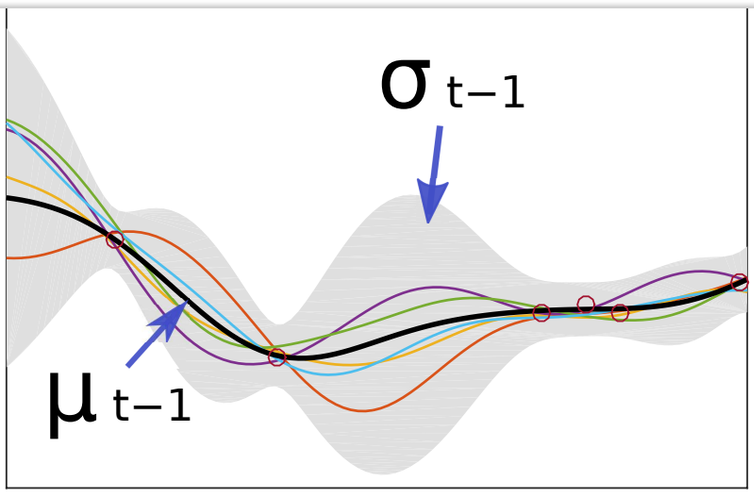


- Sample representative points
- Approximate the max-value of the representative points by a Gumbel distribution

(Wang&Jegelka, 2017)

Max-value Entropy Search (MES)

Posterior estimation



Sample a set of y_*

Define acquisition function

$$\alpha_t(x) = I(\{x, y\}; y_* \mid D_t)$$

Evaluate f at

$$x_t = \arg\max \alpha_t(x)$$

$t \leftarrow t + 1$

(Wang&Jegelka, 2017)

Max-value Entropy Search

Point to query

Observed Data

$$\alpha_t(x) = I(\{x, y\}; x_* \mid D_t)$$ Location of global optimum

Input space

Output space

$$\alpha_t(x) = I(\{x, y\}; y_* \mid D_t)$$ **1-D Global max-value**

$$\approx \frac{1}{K} \sum_{y_* \in Y_*} \left[\frac{\gamma_y(x)\psi(\gamma_{y_*}(x))}{2\Psi(\gamma_{y_*}(x))} - \log(\mathrm{Jacobian}(\cdot)) \right]$$

Something Closed-form

What do we lose?

How to understand the acquisition function?

(Wang&Jegelka, 2017; Hoffman&Zoubin, 2015)

MES v.s. PES

- Predictive entropy search (PES) samples x_*

(Hernandez-Lobato et al., 2014)

Location of global optimum

- Max-value entropy search (MES) samples y_*

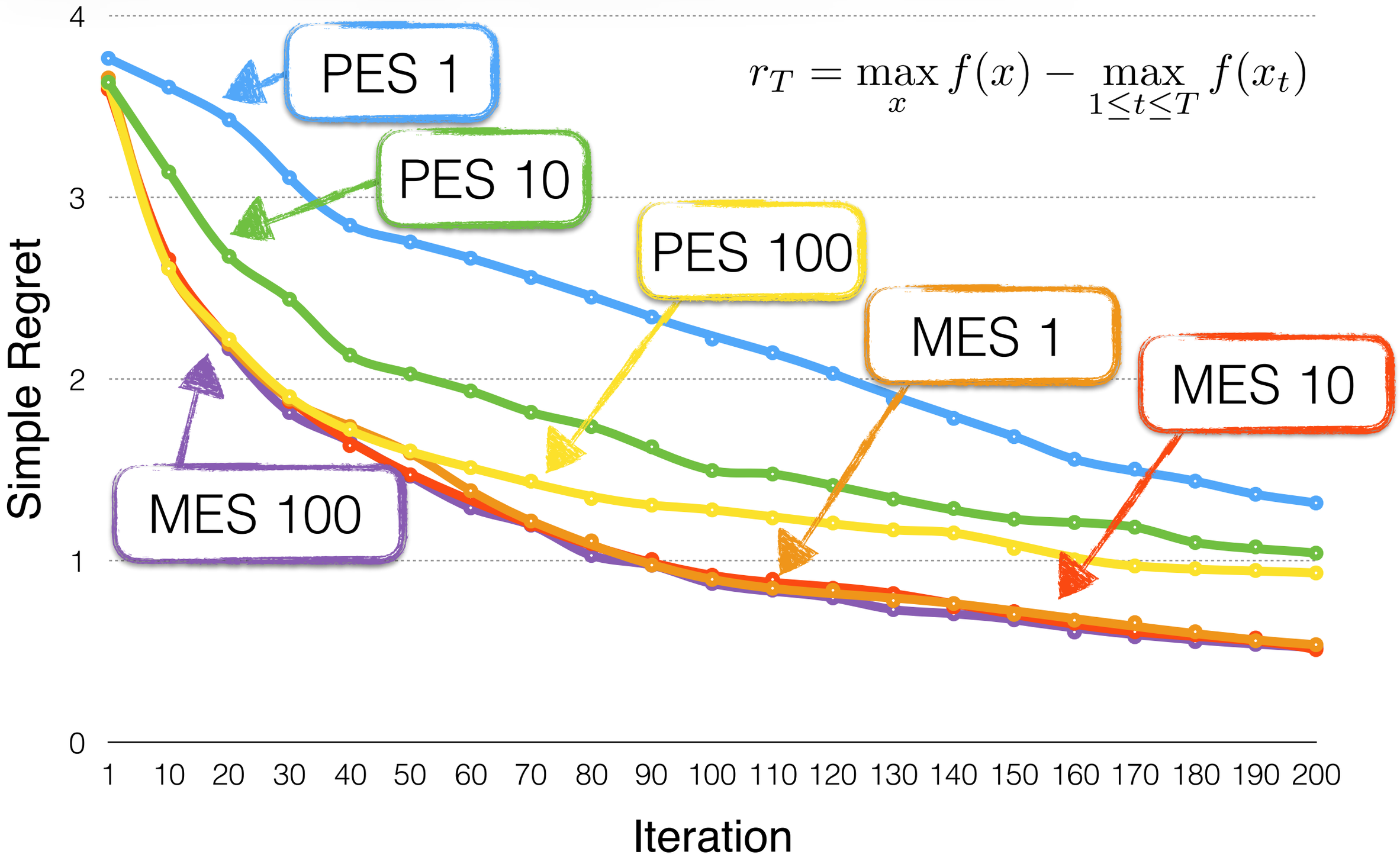
(Wang&Jegelka, 2017)

1-D Global max-value

Performance criterion: simple regret

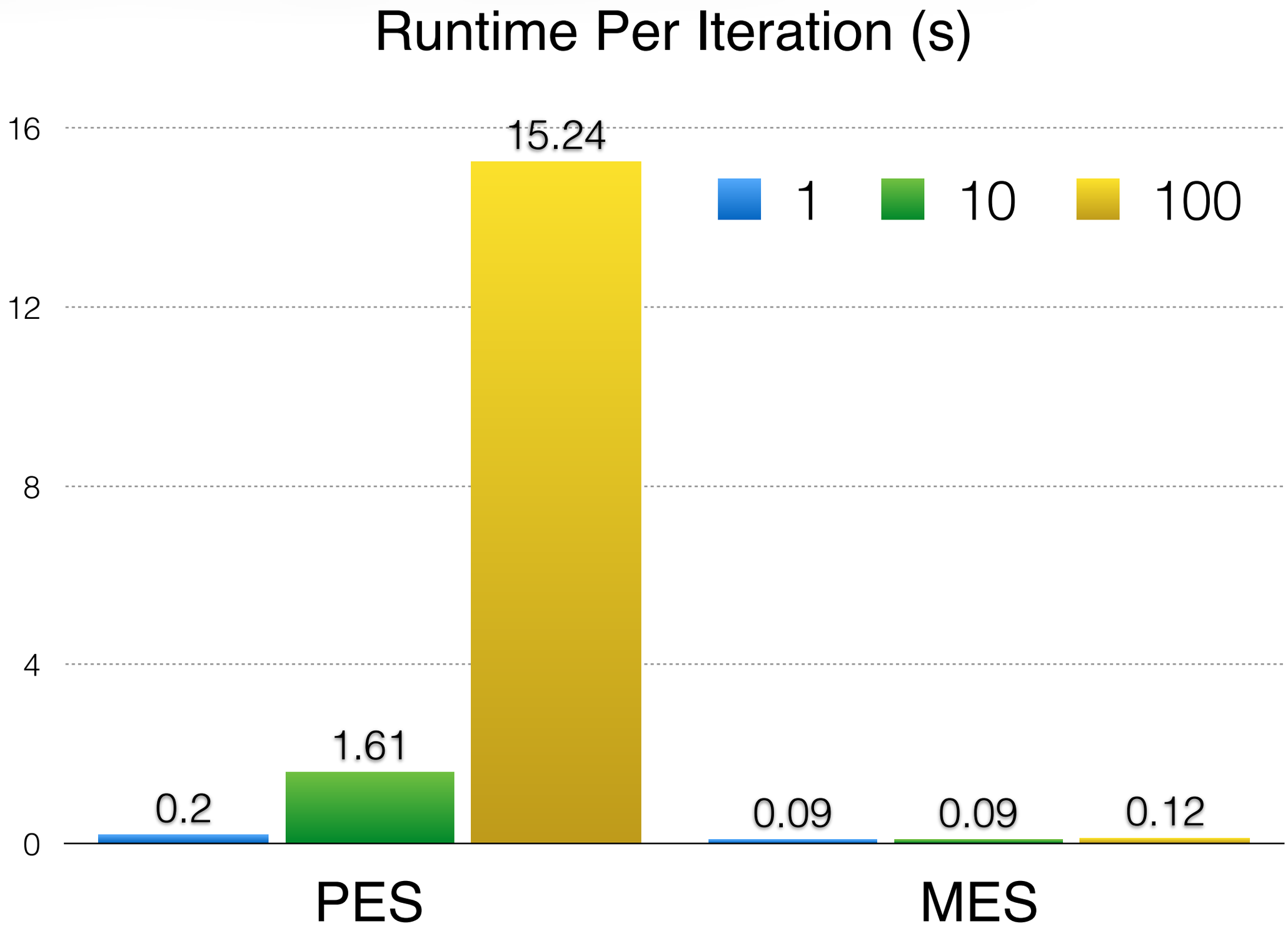
$$r_T = \max_x f(x) - \max_{1 \le t \le T} f(x_t)$$

Faster and Better Empirical Results than PES



(Wang&Jegelka, 2017)

Faster and Better Empirical Results than PES



(Wang&Jegelka, 2017)

Regret Bound for Max-value Entropy Search

$$r_T = \max_x f(x) - \max_{1 \le t \le T} f(x_t)$$

Theorem (Wang&Jegelka, 2017)

Let F be the cumulative probability distribution for the maximum of $GP(\mu, k)$, $f_* = \max_{\boldsymbol{x} \in \mathfrak{X}} f(\boldsymbol{x})$, and $w = F(f_*) \in (0, 1)$. With probability $1 - \delta$, and $T' = \sum_{i=1}^{T} \log_w \frac{\delta}{2\pi_i}$ number of iterations, MES with a single y_* sample achieves

$$r_{T'} \le O(\sqrt{\frac{(\log T)^{d+2}}{T}})$$

First regret bound for an entropy search method.

Max-value Entropy Search

Point to query

Observed Data

$$\alpha_t(x) = I(\{x, y\}; x_* \mid D_t)$$ Location of global optimum

Input space

Output space

$$\alpha_t(x) = I(\{x, y\}; y_* \mid D_t)$$ **1-D Global max-value**

$$\approx \frac{1}{K} \sum_{y_* \in Y_*} \left[\frac{\gamma_y(x)\psi(\gamma_{y_*}(x))}{2\Psi(\gamma_{y_*}(x))} - \log(\mathrm{Id}\,\gamma_{y_*}(x)) \right]$$

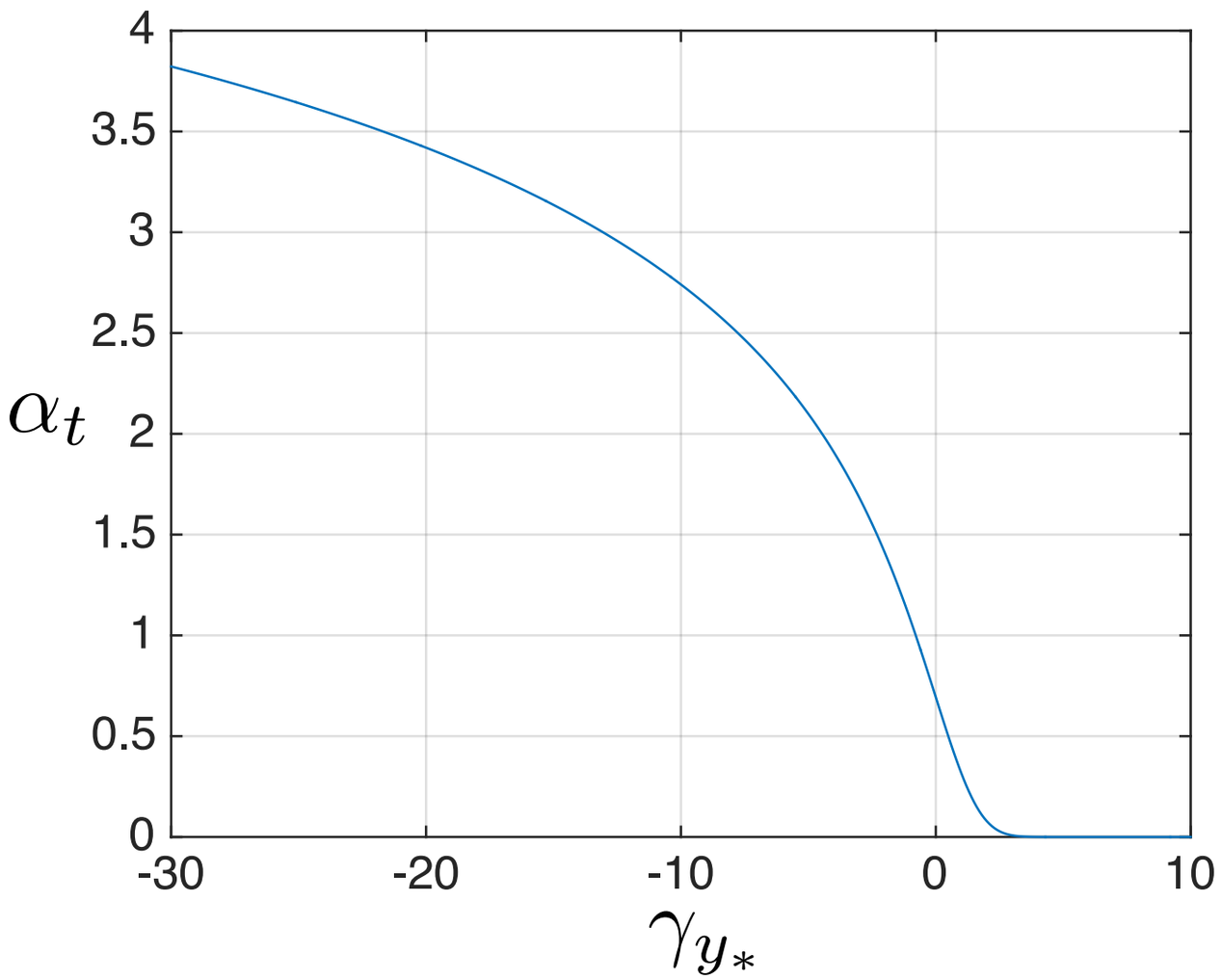
Something Closed-form

How to understand the acquisition function?

(Wang&Jegelka, 2017; Hoffman&Zoubin, 2015)

Understanding the acquisition function in MES

$$\alpha_t(x) \approx \alpha_t(\gamma_{y_*}(x)) - \log(\Psi(\gamma_{y_*}(x)))$$



$$\gamma_{y_*}(x) = \frac{y_* - \mu_{t-1}(x)}{\sigma_{t-1}(x)}$$

So, $\underset{x}{\text{maximize}}\ \alpha_t(x)$

is equivalent to

$\underset{x}{\text{minimize}}\ \gamma_{y_*}(x)$.

(Wang&Jegelka, 2017)

Connections to other acquisition functions

$$\gamma_{y_*}(x) = \frac{y_* - \mu_{t-1}(x)}{\sigma_{t-1}(x)}$$

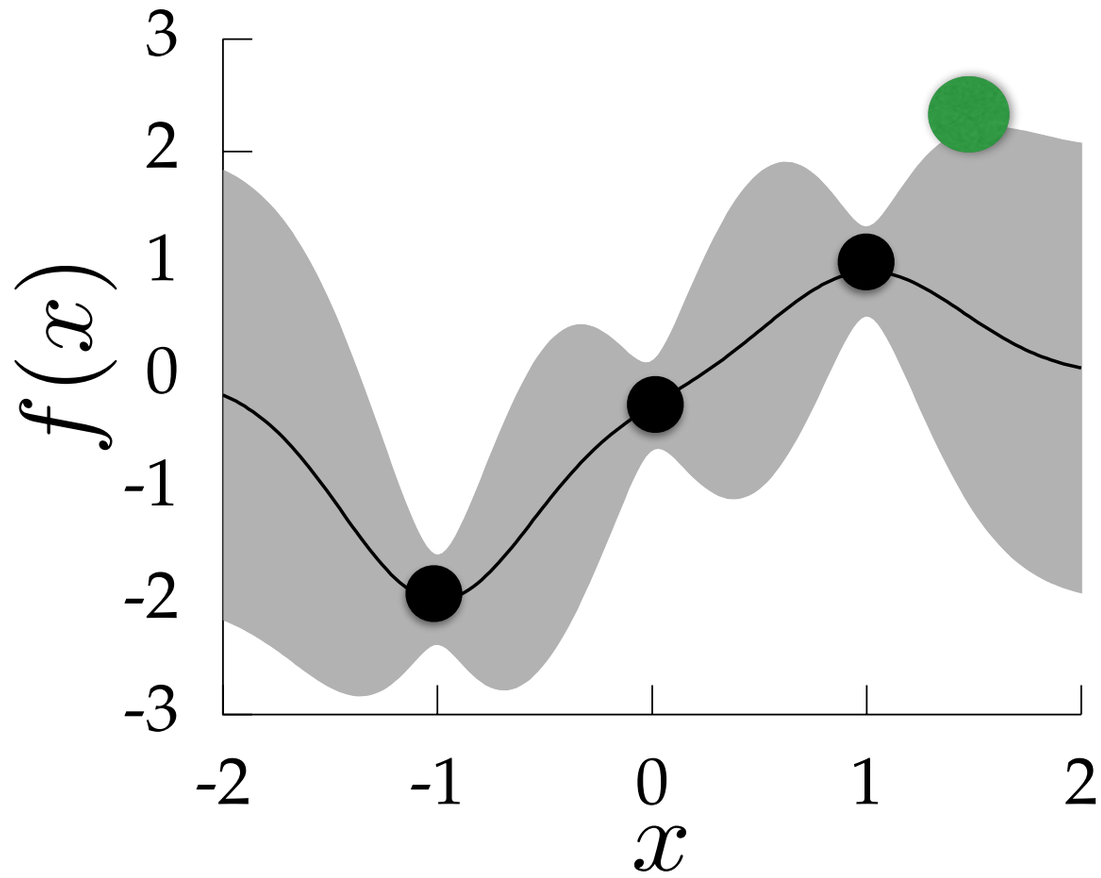
$$\underset{x}{\text{minimize}}\ \gamma_{y_*}(x) \qquad \text{MES}$$

Recall GP-UCB (Auer, 2002; Srinivas et al., 2010)

$$\alpha_t(x) = \mu_{t-1}(x) + \beta\ \sigma_{t-1}(x)$$

Set $\beta = \min_x \gamma_{y_*}(x)$

GP-UCB = MES.



(Wang&Jegelka, 2017)

Connections to other acquisition functions

Lemma (Wang&Jegelka, 2017; Wang&Zhou&Jegelka, 2016)

The following acquisition functions are equivalent:

- MES with a single sample y_* to estimate $\alpha_t(x)$
- GP-UCB (upper confidence bound, Srinivas et al., 2010) with a specific parameter setting
- PI (probability of improvement, Kushner, 1964) with a specific parameter setting.

Roadmap

Matlab open sourced on Github;
Python implemented in GPflowOpt (Knudde et al., 2017).

- Better acquisition function: *Max-value Entropy Search*

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

- Scaling up input dimensions

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

- Scaling up observations & parallel queries

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

Roadmap

- Better acquisition function: *Max-value Entropy Search*

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

- Scaling up input dimensions

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

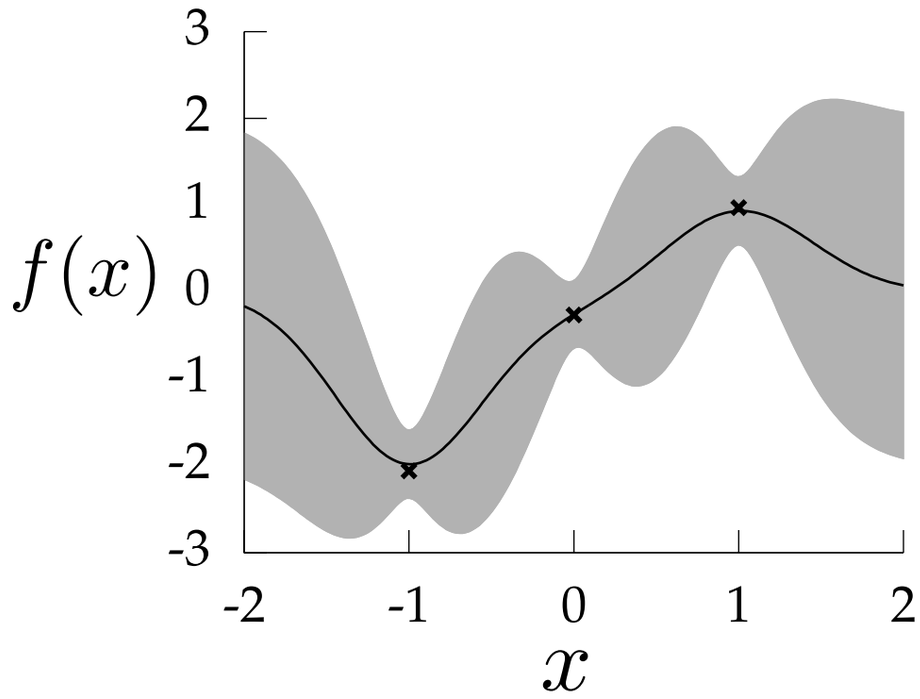
- Scaling up observations & parallel queries

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

Challenges in high-dimensional BO

- optimizing non-convex acquisition functions in high dimensions
  computationally challenging

- estimating a nonlinear function in high input dimensions: need more observations
  statistically challenging

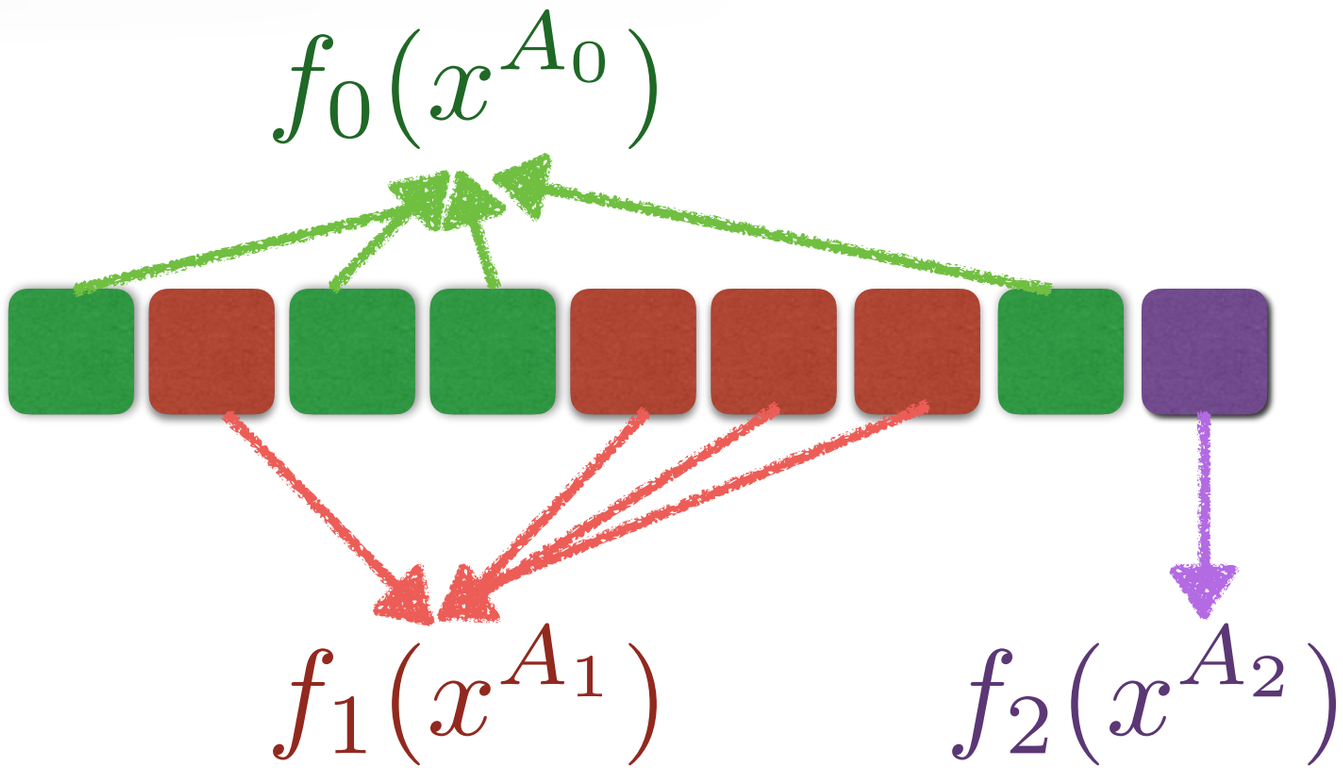


$$\text{regret} = O(\sqrt{\frac{(\log T)^{d+2}}{T}})$$

(Wang et al., 2013; Djolonga el al., 2013; Kandasamy et al., 2015)

Possible solution: additive Gaussian processes

$$f(x) = \sum_{m \in [M]} f_m(x^{A_m})$$



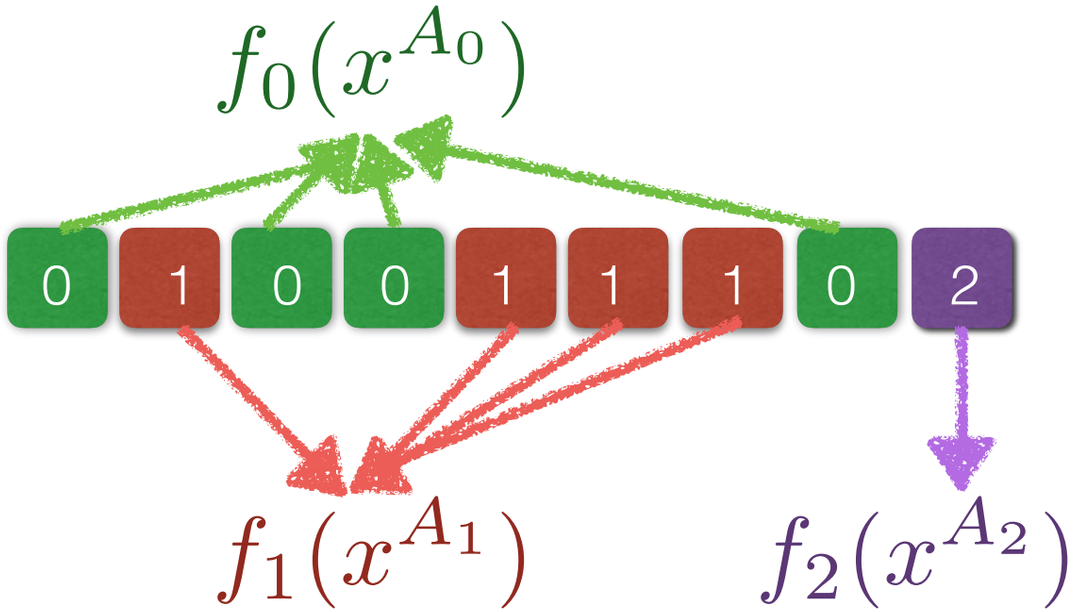
- optimize acquisition function block-wise
  computational efficiency
- lower-complexity functions
  statistical efficiency

What is the additive structure?

(Hastie&Tibshirani, 1990; Kandasamy et al., 2015)

Structural Kernel Learning

$$f = f_0 + f_1 + f_2$$



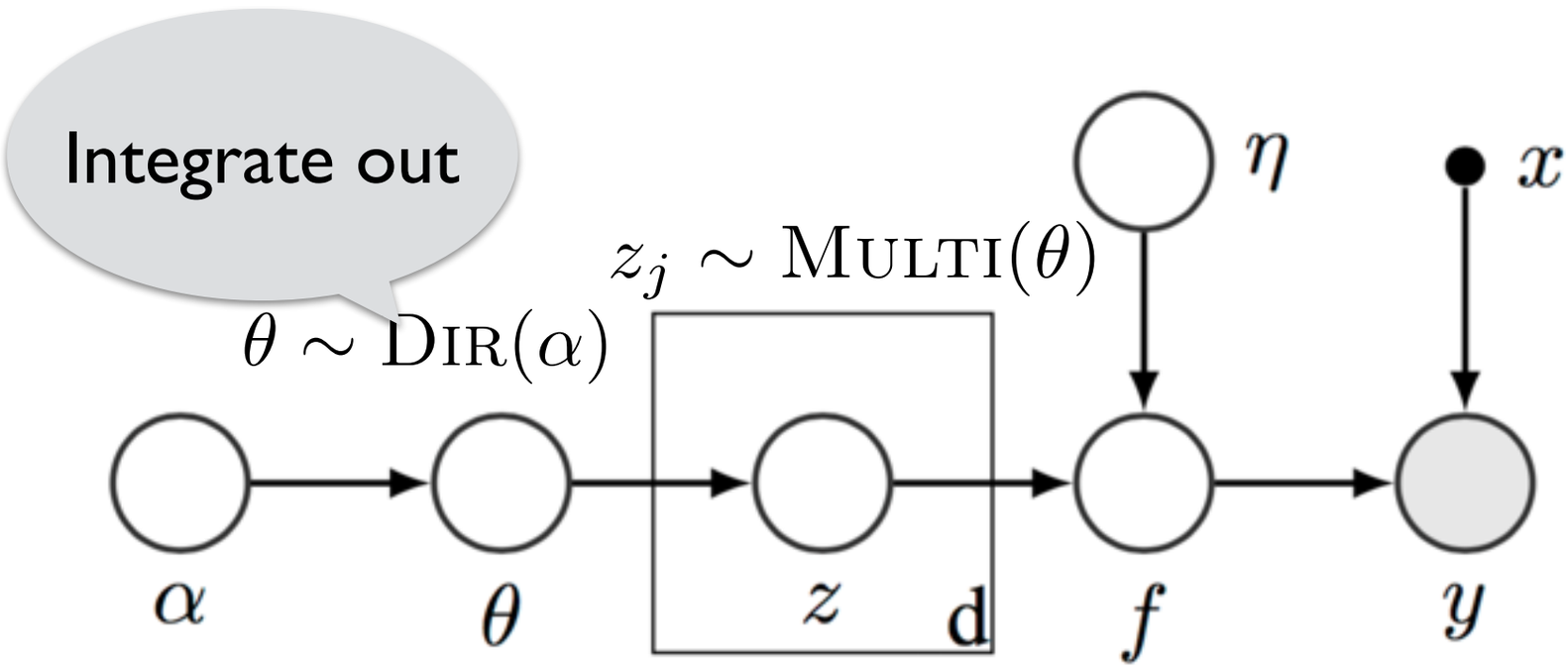
Example:

z = [0 1 0 0 1 1 1 0 2]

Learn the assignment!

Key idea:

Dirichlet prior on z



Learn posterior

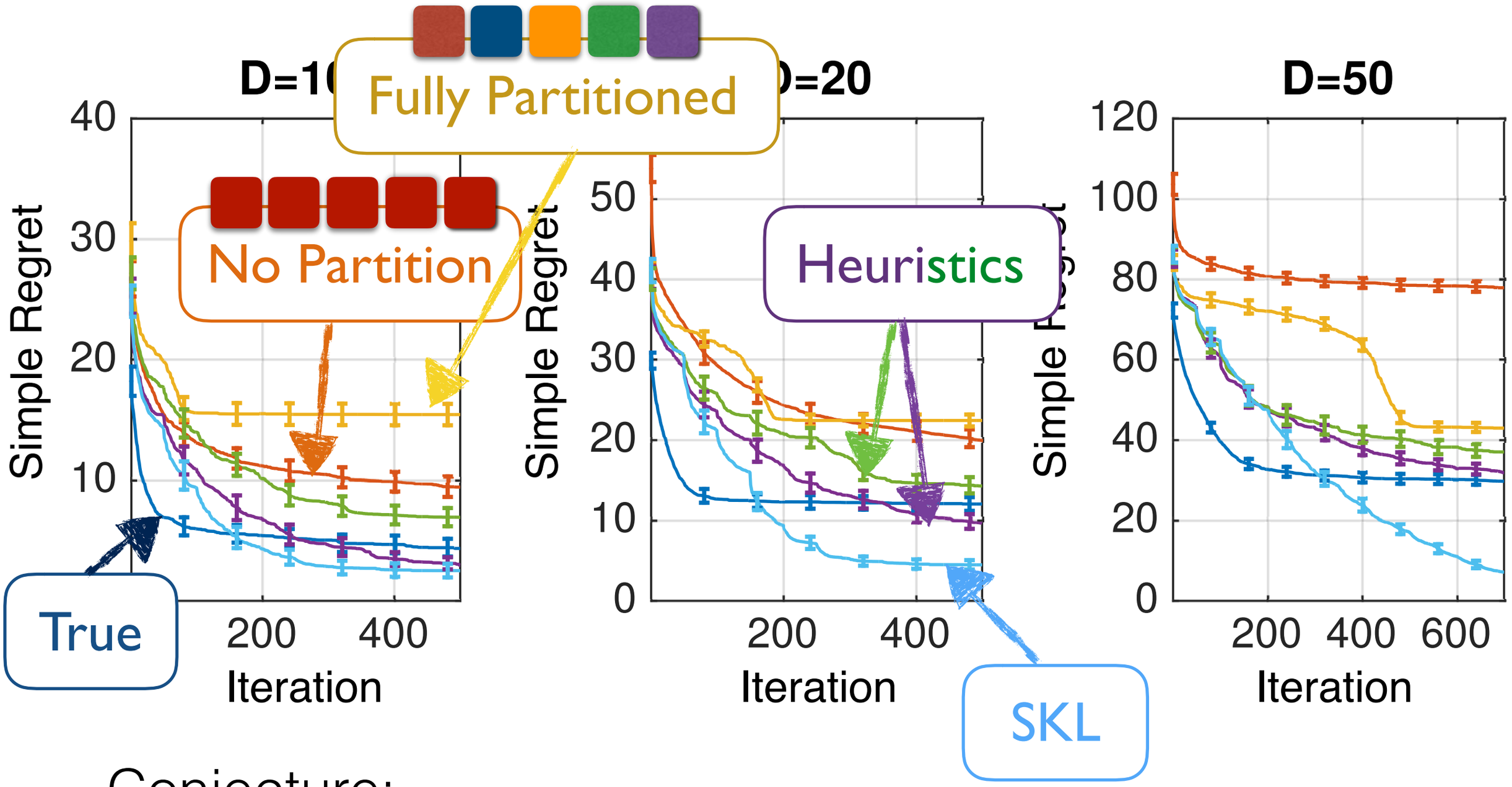
$$p(z \mid D_n; \alpha)$$

via Gibbs sampling.

easy updates

(Wang et al., 2017)

Empirical Results of Structural Kernel Learning

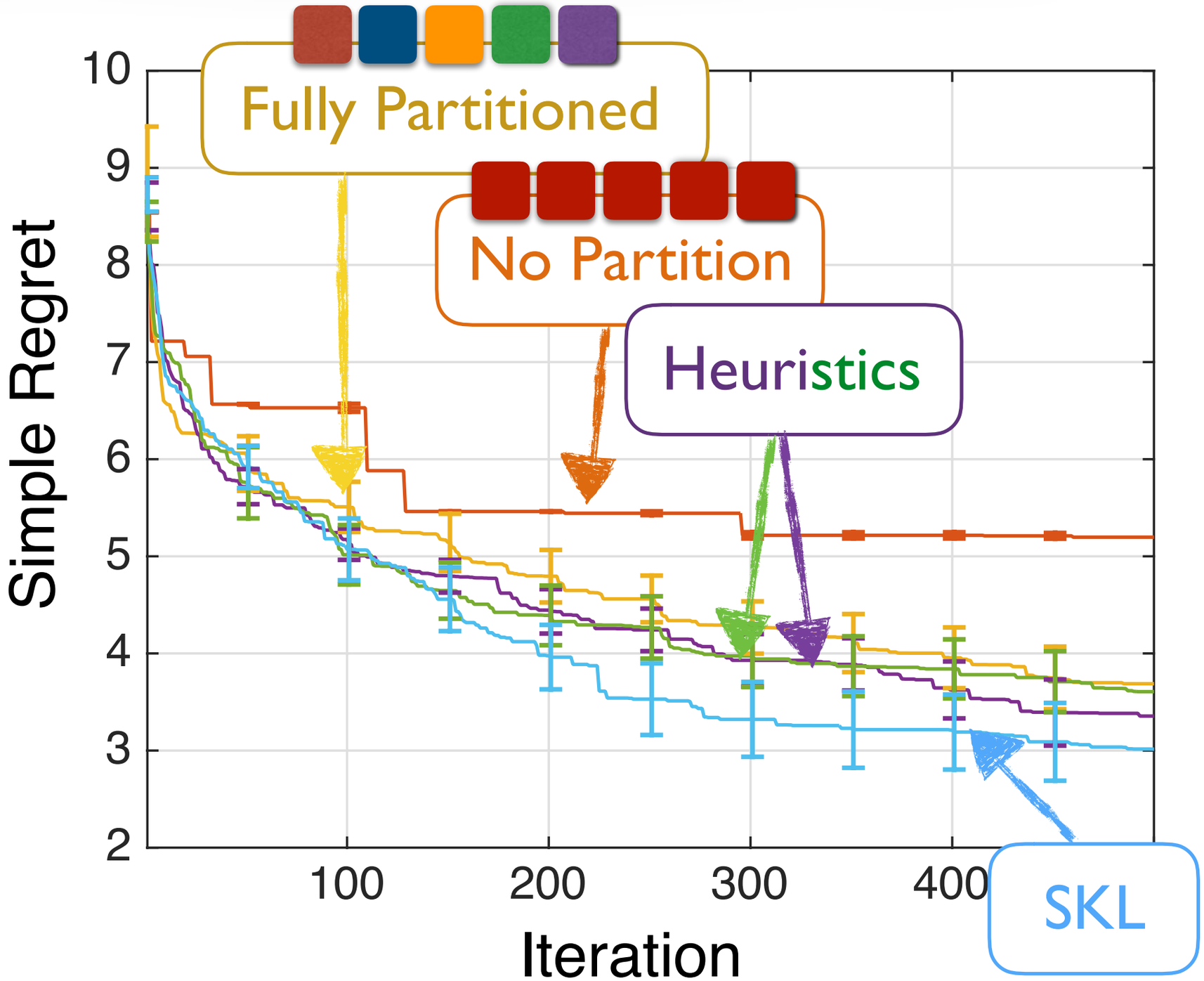


Conjecture:

SKL is better than True because of more exploration

(Wang et al., 2017)

Empirical Results of Structural Kernel Learning

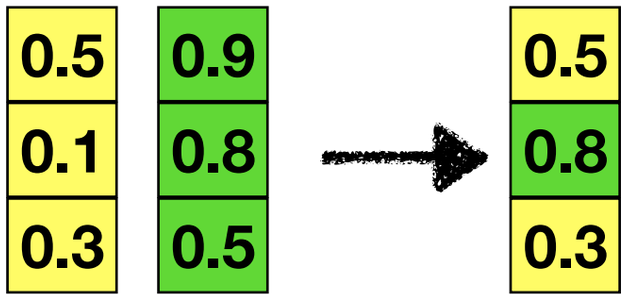


(Wang et al., 2017)

Interesting Connection to Evolutionary Algorithms

Evolutionary/Genetic algorithms:

- maintain ensemble of promising points
- new points from exchanging coordinates of good points randomly

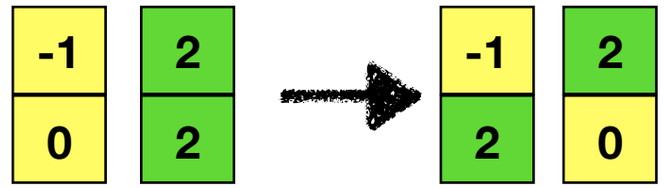


Interesting Connection to Evolutionary Algorithms

BO with additive GPs toy example: 2D

observed good points: [-1,0], [2,2]

query points: [-1,2], [2,0]



learned instead of completely random coordinate partition!

(Wang et al., 2017)

Roadmap

- Better acquisition function: *Max-value Entropy Search*

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

- Scaling up input dimensions via Structural Kernel Learning

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

- Scaling up observations & parallel queries

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

Roadmap

- Better acquisition function: *Max-value Entropy Search*

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

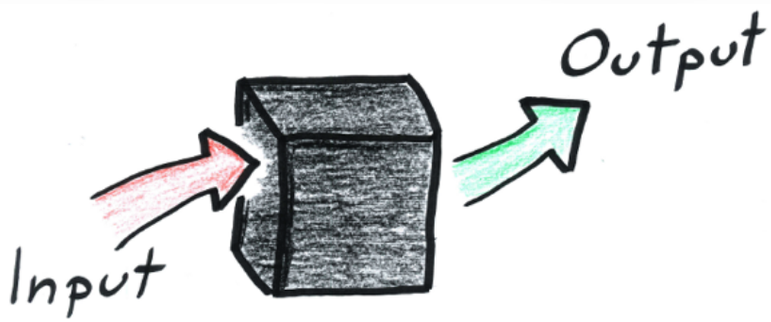
- Scaling up input dimensions via Structural Kernel Learning

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

- Scaling up observations & parallel queries

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

Large-scale Bayesian Optimization



Goal: $x^* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} f(x)$

- x is high dimensional

  $\Longrightarrow$ Large-scale observations are necessary
- f can be evaluated in parallel

  $\Longrightarrow$ Large-scale observations are available

Challenge:

In Gaussian processes, posterior predictions are

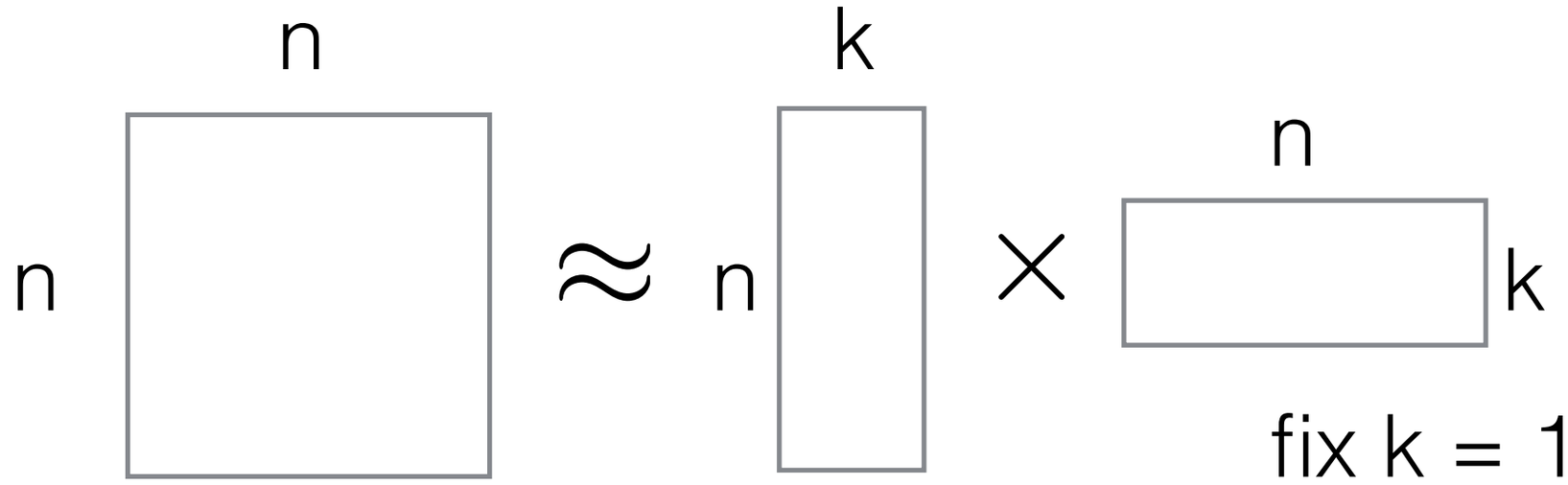
$$\mu_n(x) = k_n(x)^{\mathrm{T}} (K_n + \sigma^2 I)^{-1} y_n$$

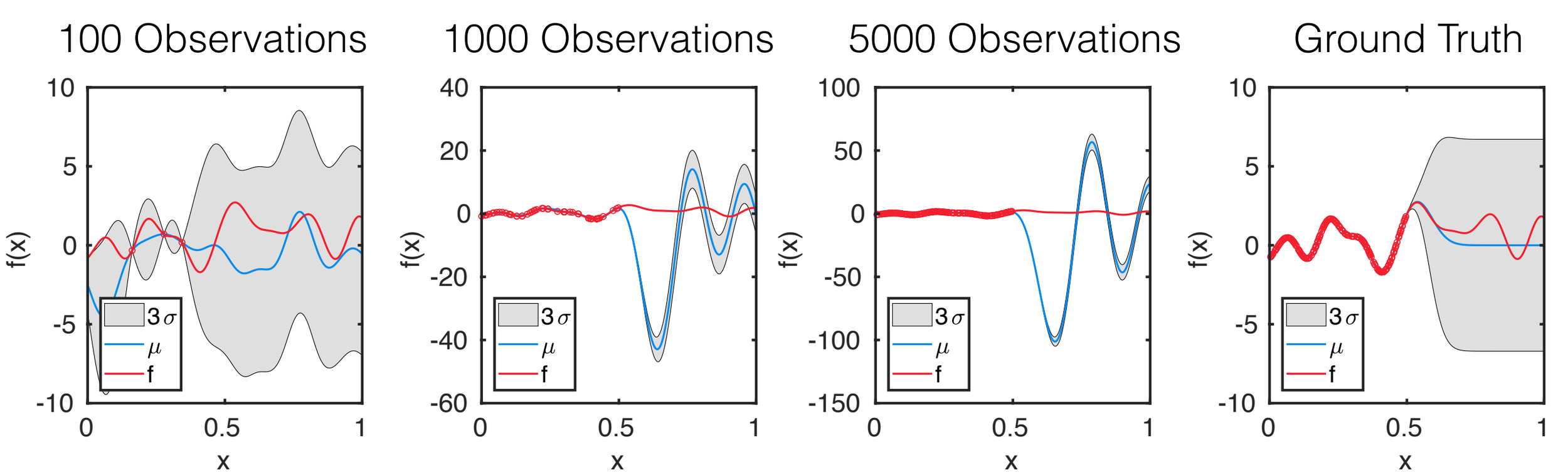
$$k_n(x, x') = k(x, x') - k_n(x)^{\mathrm{T}} (K_n + \sigma^2 I)^{-1} k_n(x')$$

inversion of large n x n matrix $O(n^3)$

Challenges in Scaling Up Observations

- low-rank approximation? e.g. random features





(Snoek et al., 2015)

Requirements for Large-scale High-dim BO

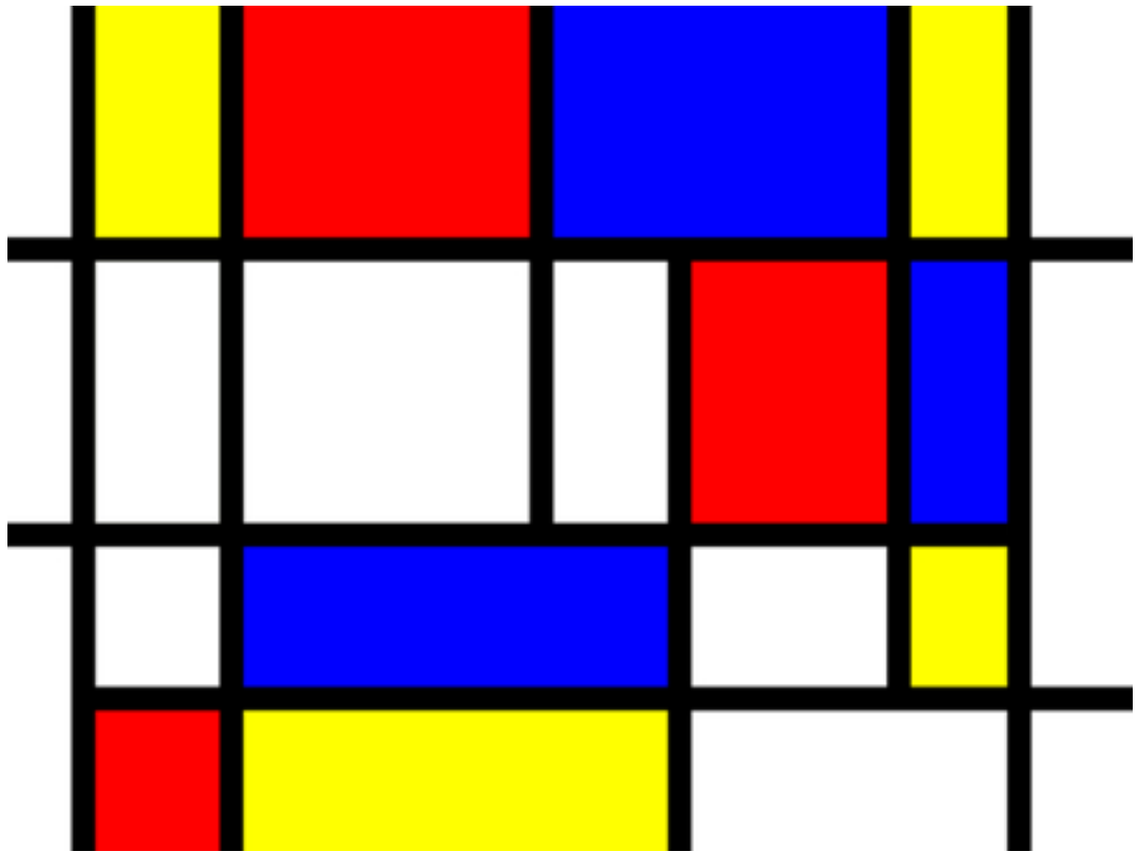
We need:

- ☐ good uncertainty estimation
- ☐ efficient computation for large-scale observations
- ☐ ability to recommend a batch of query points
- ☐ ability to handle high dimensional inputs

Overview of Ensemble Bayesian Optimization (EBO)

LOOP

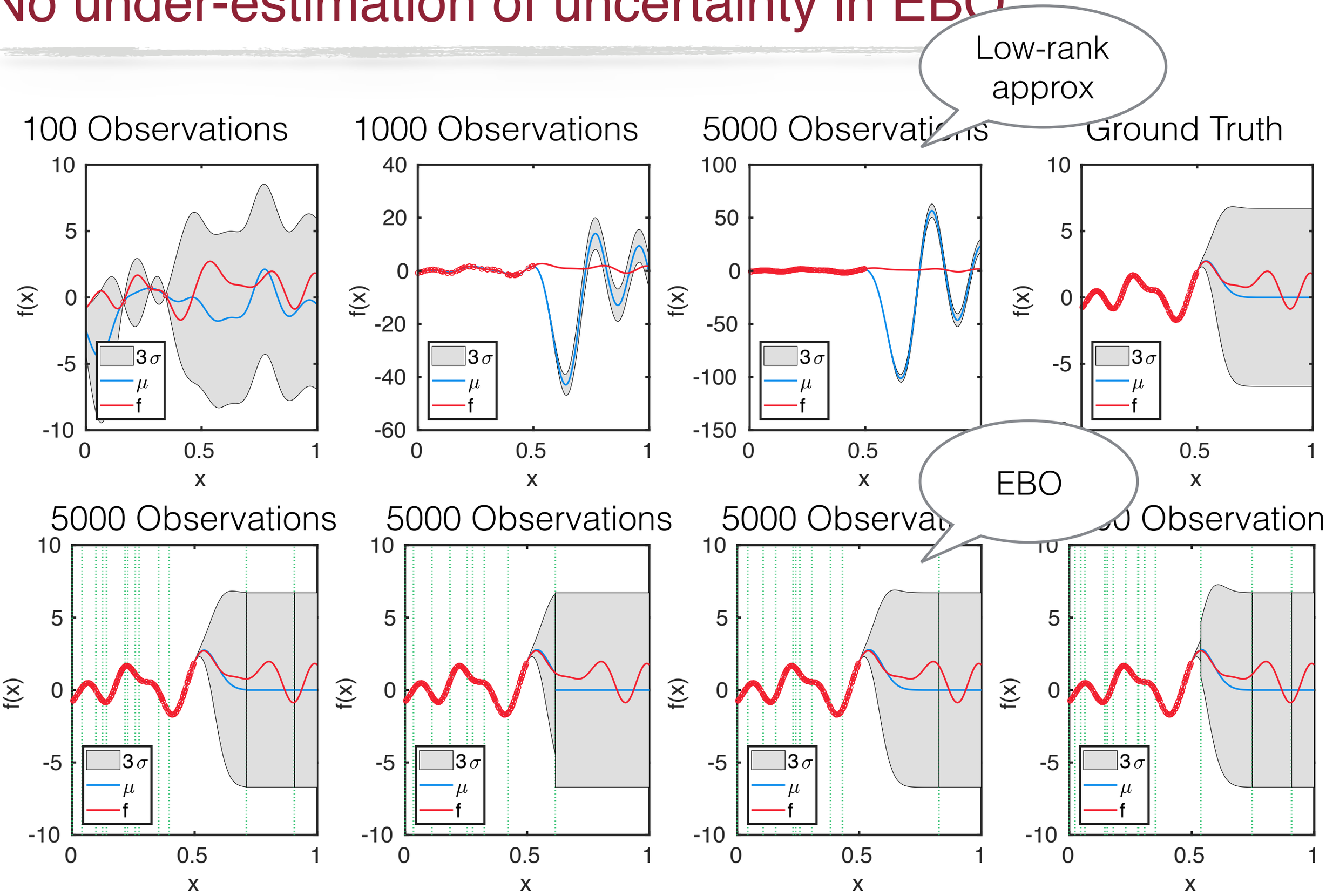
- Randomly partition the input space via a Mondrian process
- Learn an additive GP in each part and recommend BO query points
- Aggregate GPs from all the parts and filter BO query points



Highly parallelizable

(Lakshminarayanan et al., 2015; Wang et al., 2017)

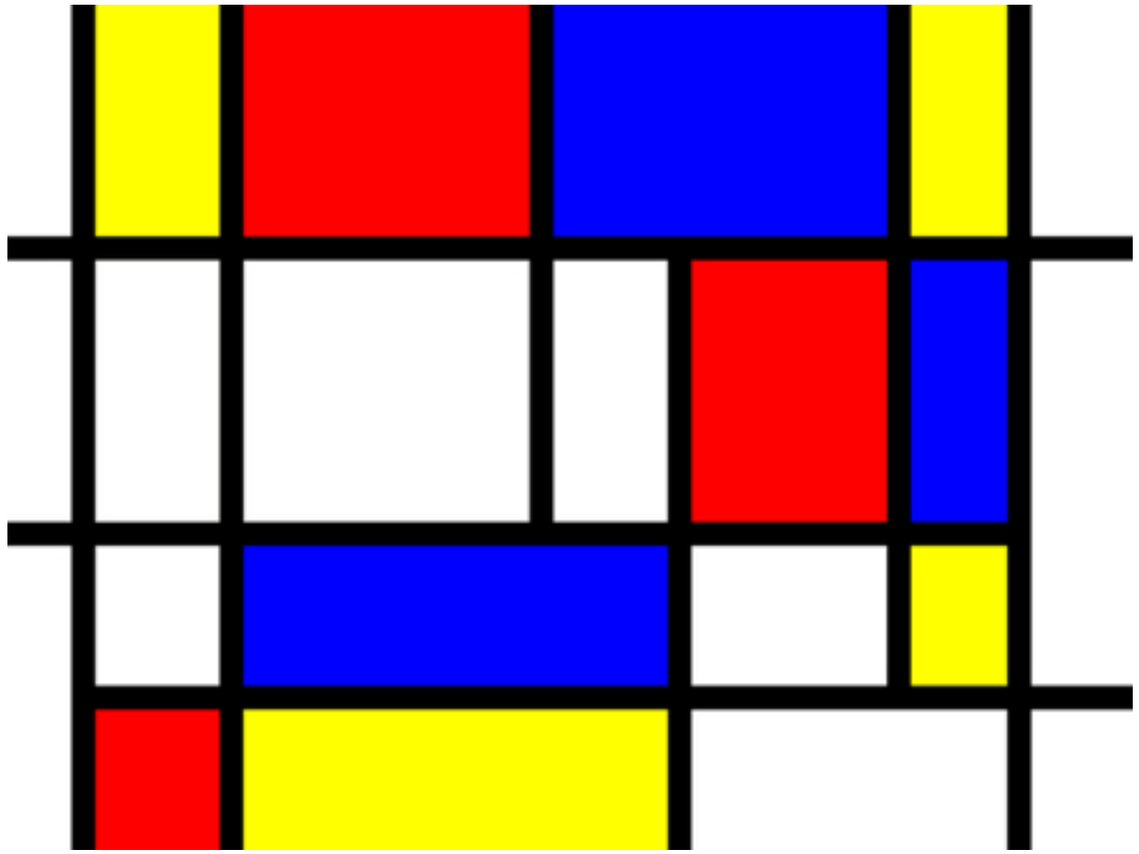
No under-estimation of uncertainty in EBO



Overview of Ensemble Bayesian Optimization (EBO)

LOOP

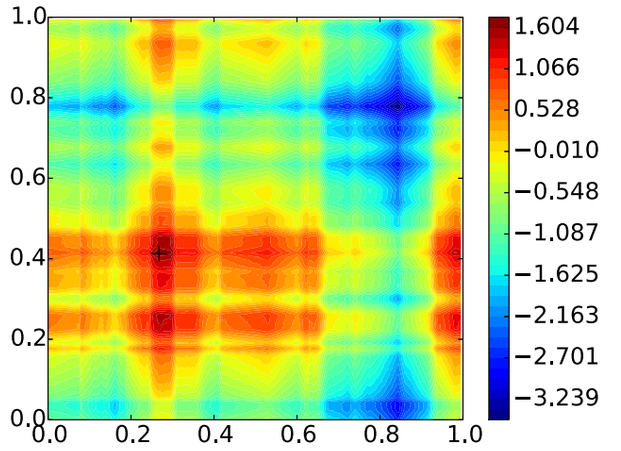
- Randomly partition the input space via a Mondrian process
- Learn an additive GP in each part and recommend BO query points
- Aggregate GPs from all the parts and filter BO query points

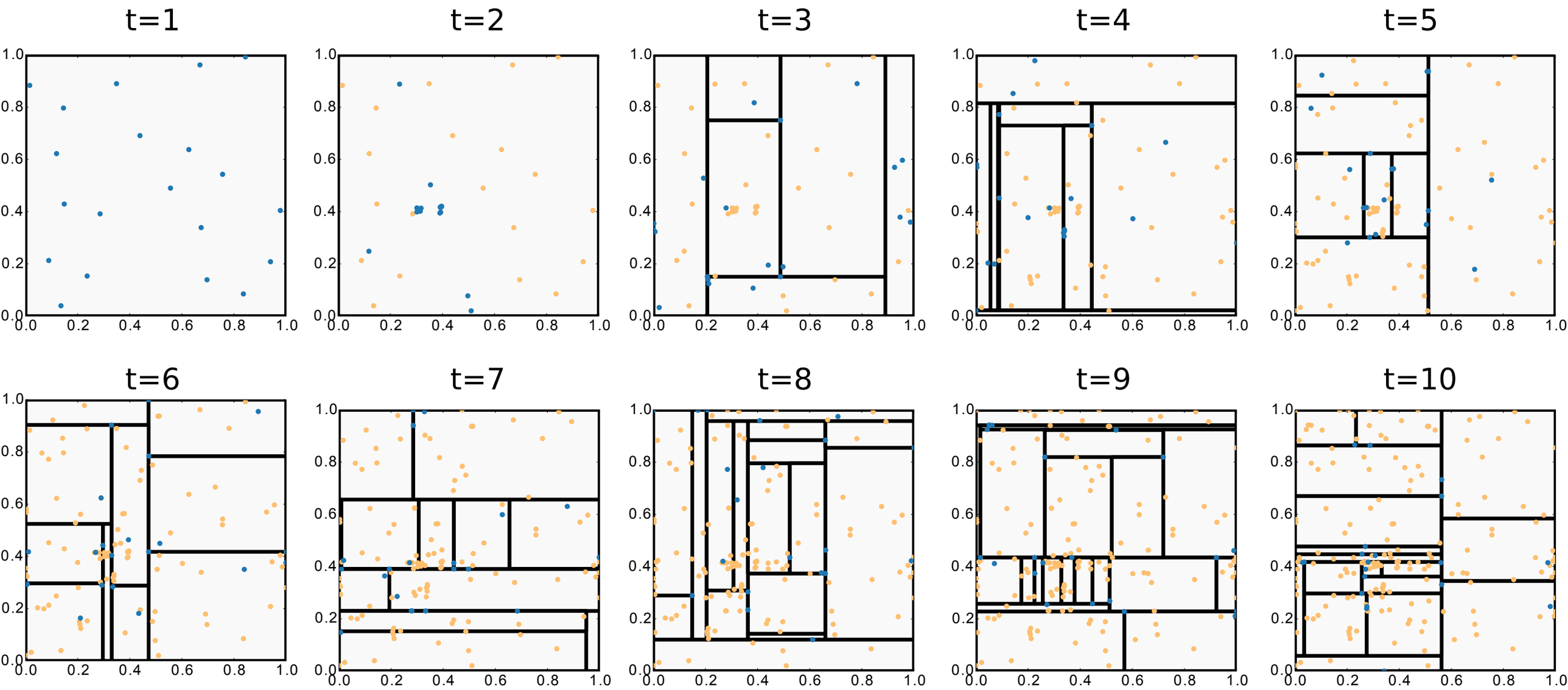


☑ uncertainty estimation
☑ large-scale observations
☑ batch queries
☑ high dimensional inputs

(Lakshminarayanan et al., 2015; Wang et al., 2017)

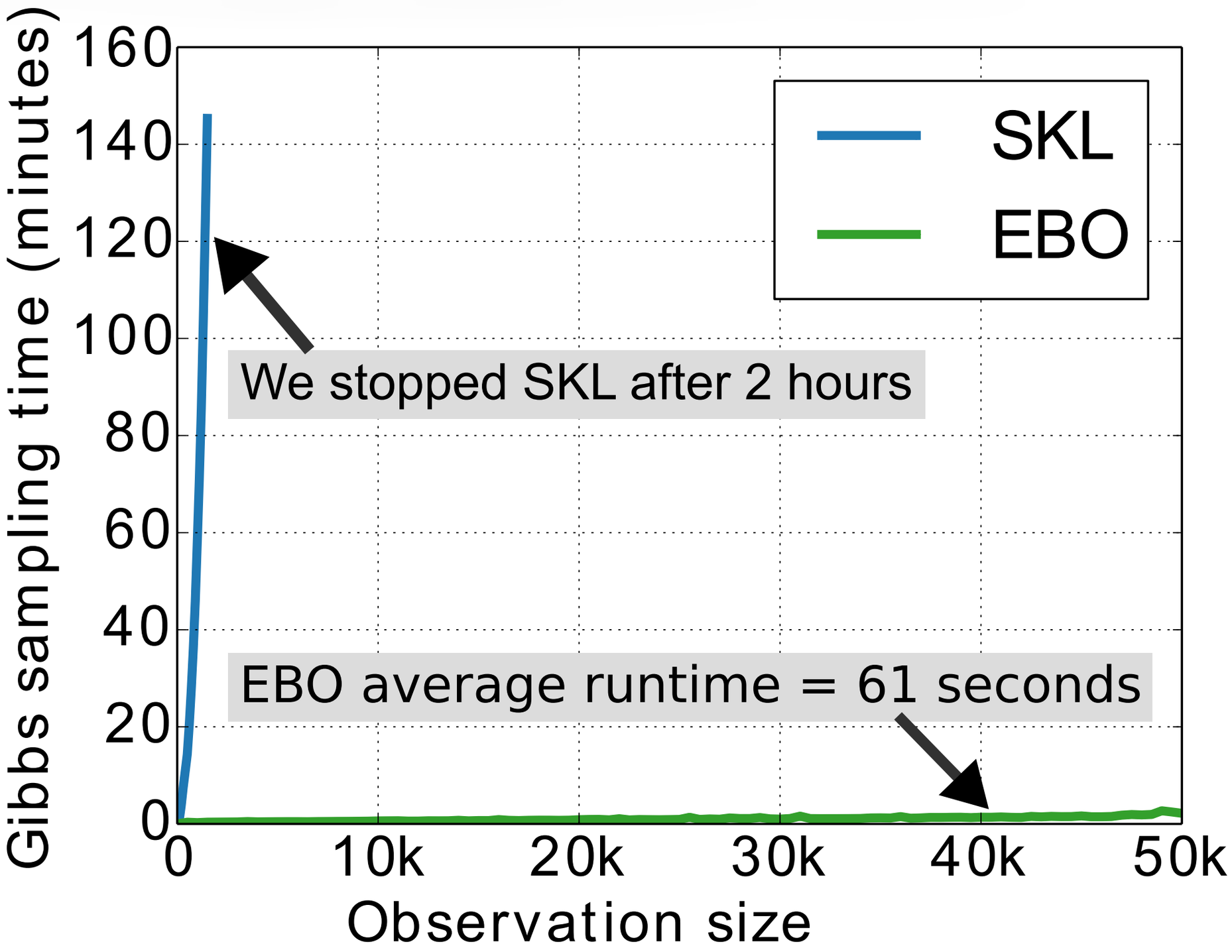
Ensemble Bayesian Optimization





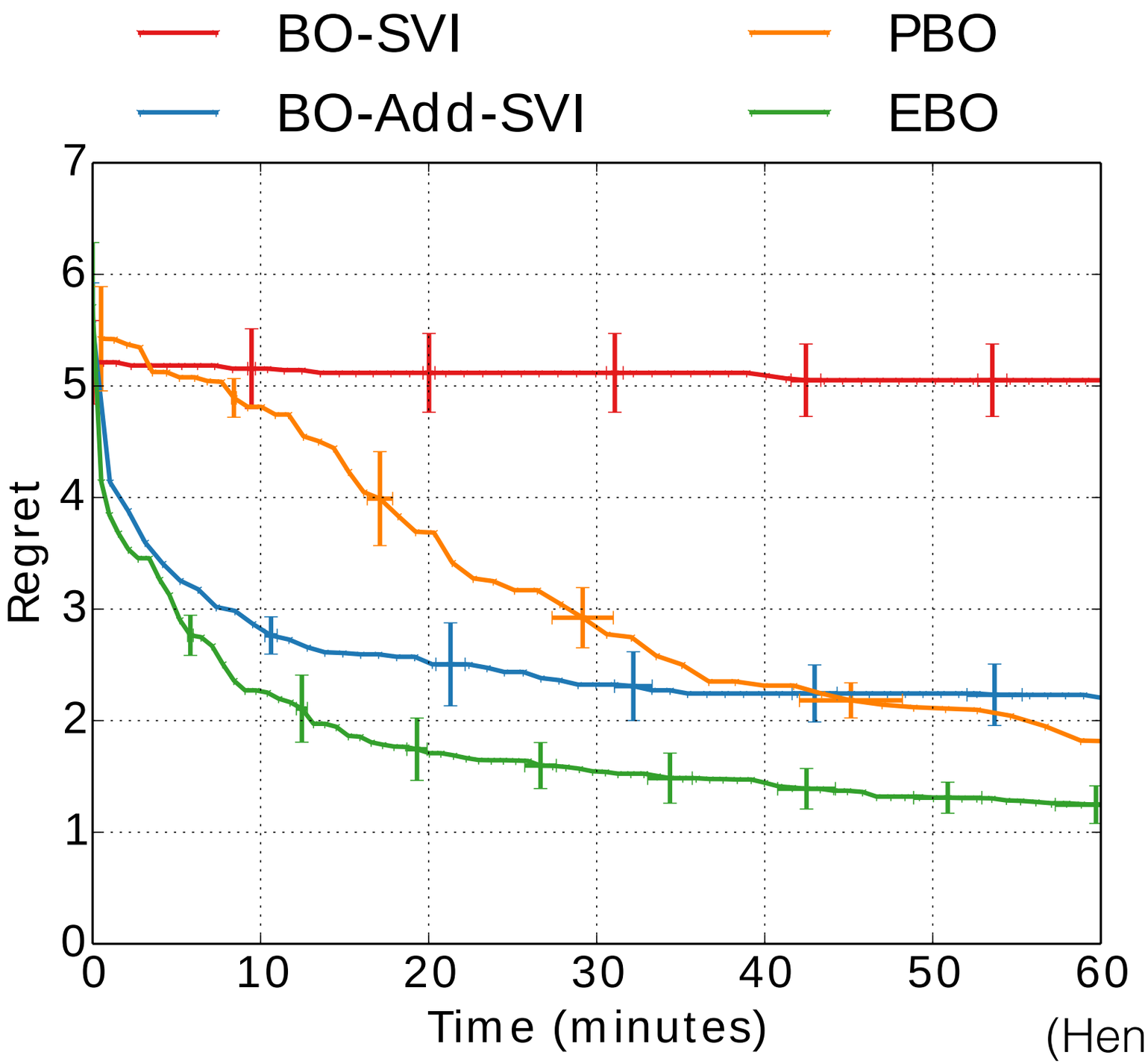
(Wang et al., 2017)

Empirical Results on Running Time



(Wang et al., 2017)

Empirical Results on Quality



(Hensman et al., 2013
Wang et al., 2017)

Roadmap

- Better acquisition function: *Max-value Entropy Search*

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

- Scaling up input dimensions via Structural Kernel Learning

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

- Scaling up observations & parallel queries:

  Ensemble Bayesian Optimization

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

Summary

Challenges in Gaussian Processes & Bayesian Optimization: high dimensions, many observations, parallelization

- Better acquisition function: *Max-value Entropy Search*

  (Zi Wang, Stefanie Jegelka, ICML 2017; Zi Wang, Bolei Zhou, Stefanie Jegelka, AISTATS 2016)

- Scaling up input dimensions via Structural Kernel Learning

  (Zi Wang, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli, ICML 2017)*

- Scaling up observations & parallel queries:

  Ensemble Bayesian Optimization

  (Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka, arXiv 2017)

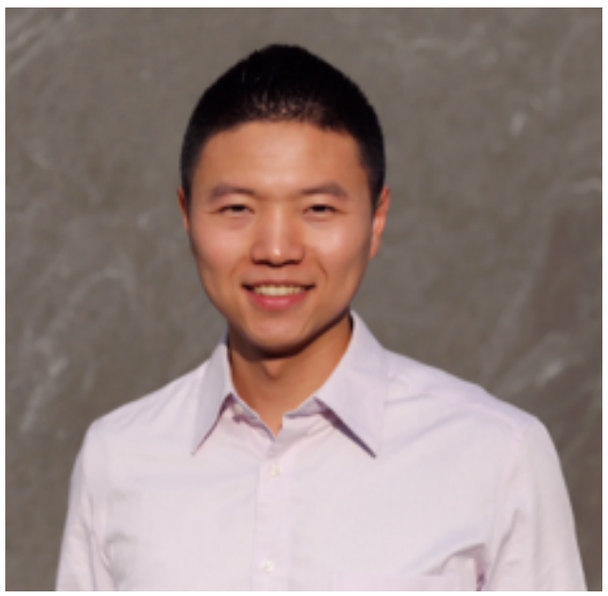
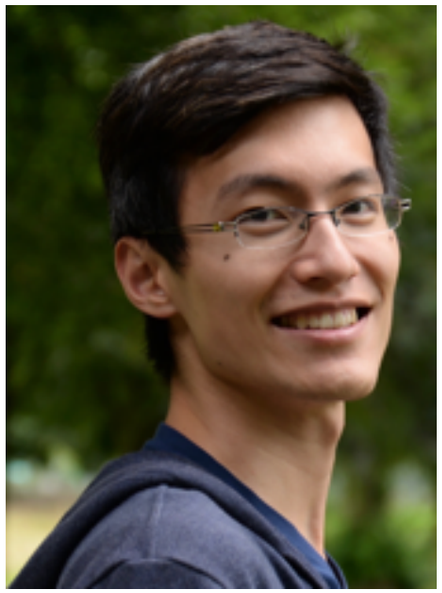
References

- *Zi Wang, Bolei Zhou, Stefanie Jegelka.* Optimization as Estimation with Gaussian Processes in Bandit Settings. *AISTATS 2016.*
- *Zi Wang, Stefanie Jegelka.* Max-value entropy search for efficient Bayesian Optimization. *ICML 2017.*
- *Zi Wang*, Chengtao Li*, Stefanie Jegelka, Pushmeet Kohli.* Batched High-dimensional Bayesian Optimization via Structural Kernel Learning. *ICML 2017.*
- *Zi Wang, Clement Gehring, Pushmeet Kohli, Stefanie Jegelka.* Ensemble Bayesian Optimization. *arXiv, 2017.*

Acknowledgement






Stefanie Jegelka, Bolei Zhou, Chengtao Li, Clement Gehring, Leslie Kaelbling, Tomas Lozano-Perez (MIT) and Pushmeet Kohli (DeepMind)

Thank you!

Questions?