



Regret bounds for meta Bayesian optimization with an **unknown** Gaussian process prior

Zi Wang

Microsoft Research AI Breakthroughs Workshop, Sep 18

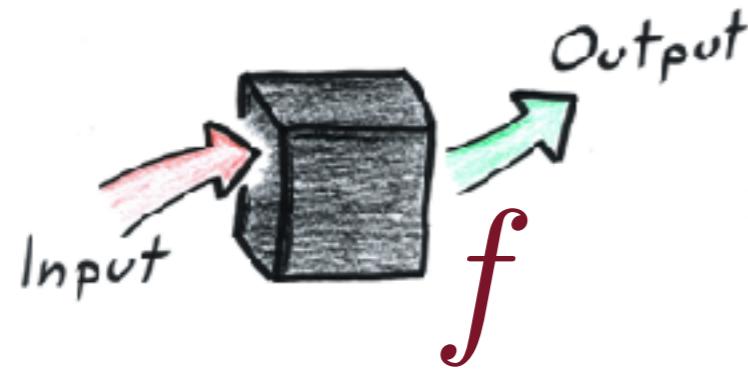
Joint work with Beomjoon Kim, Leslie Pack Kaelbling



Blackbox Function Optimization

a.k.a derivative-free optimization,
experimental design,
(continuous) multi-armed bandit

Goal: $x^* = \operatorname{argmax}_{\substack{\text{box } \mathcal{X} \subset \mathbb{R}^d \\ \text{or } |\mathcal{X}| = M}} f(x)$



Challenges:

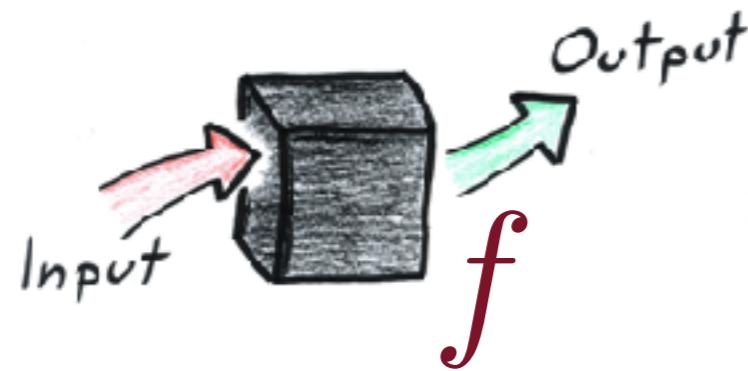
- f is expensive to evaluate
- f is multi-peak
- no gradient information
- evaluations can be noisy

(Kim et al., 2017; Snoek et al., 2012; Gonzalez et al., 2015)

Blackbox Function Optimization

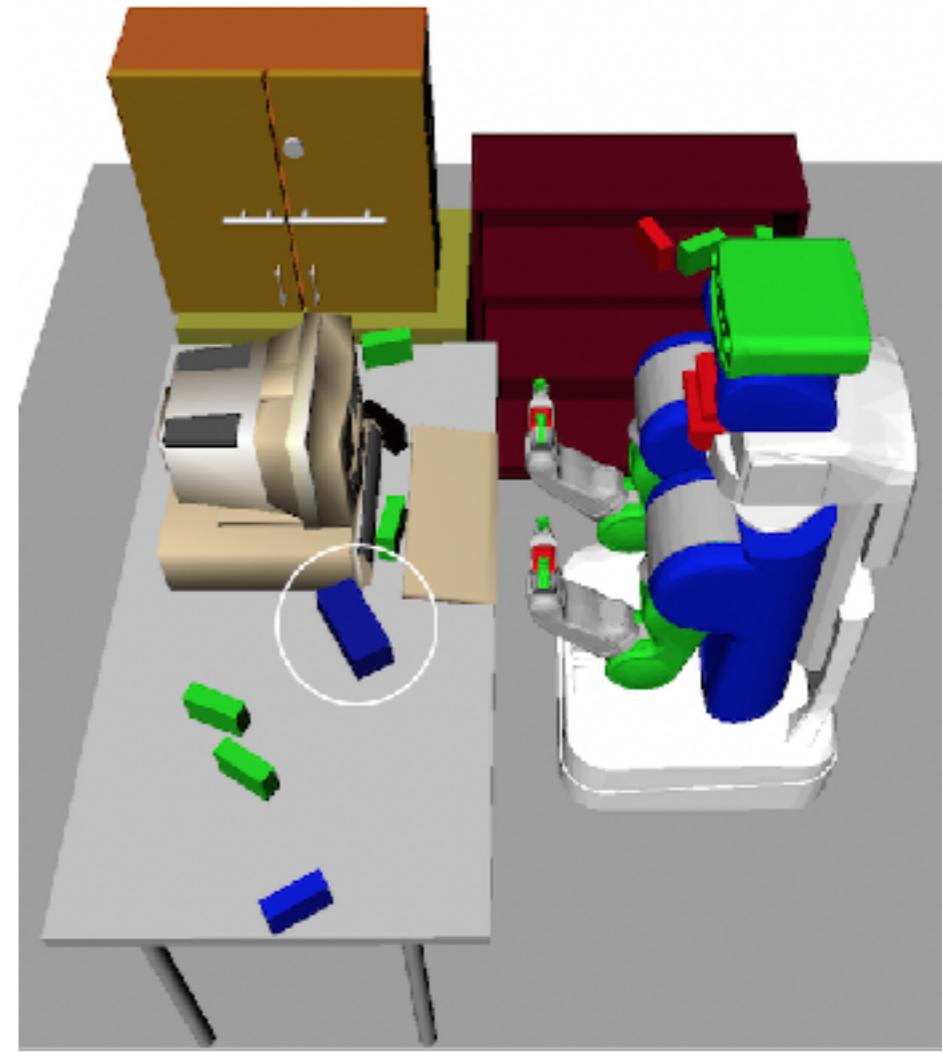
a.k.a derivative-free optimization,
experimental design,
(continuous) multi-armed bandit

Goal: $x^* = \operatorname{argmax}_{\text{box } \mathcal{X} \subset \mathbb{R}^d \text{ or } |\mathcal{X}| = M} f(x)$



Challenges:

- f is expensive to evaluate
- f is multi-peak
- no gradient information
- evaluations can be noisy

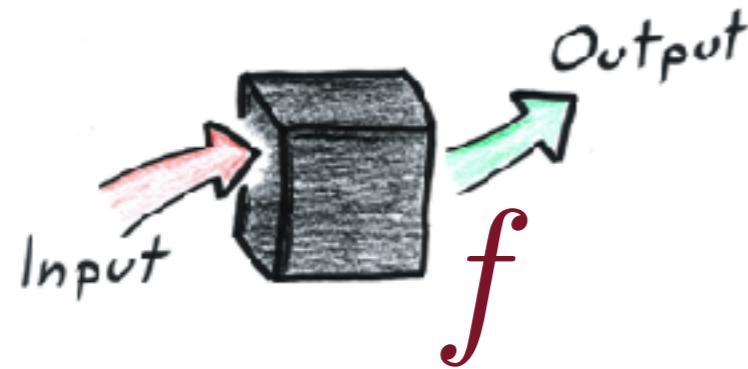


(Kim et al., 2017; Snoek et al., 2012; Gonzalez et al., 2015)

Blackbox Function Optimization

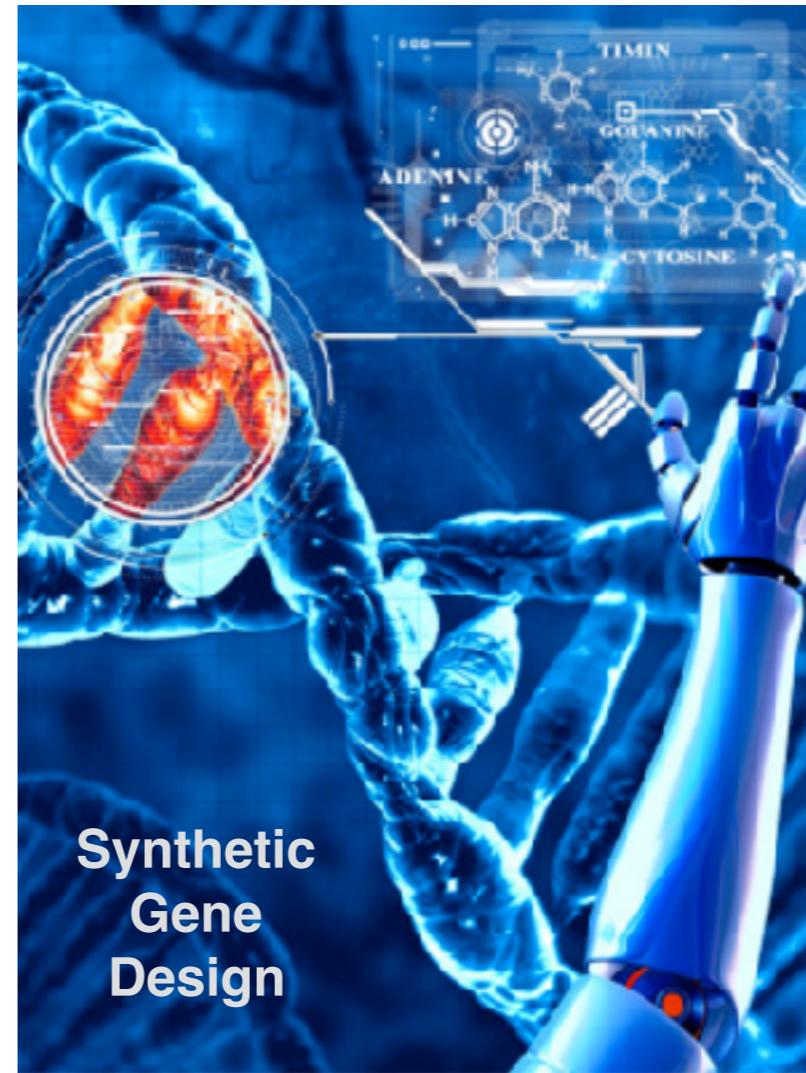
a.k.a derivative-free optimization,
experimental design,
(continuous) multi-armed bandit

Goal: $x^* = \operatorname{argmax}_{\substack{\text{box } \mathcal{X} \subset \mathbb{R}^d \\ \text{or } |\mathcal{X}| = M}} f(x)$



Challenges:

- f is expensive to evaluate
- f is multi-peak
- no gradient information
- evaluations can be noisy

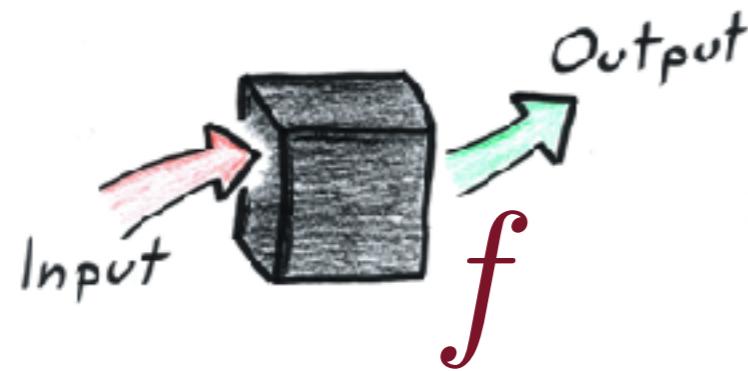


(Kim et al., 2017; Snoek et al., 2012; Gonzalez et al., 2015)

Blackbox Function Optimization

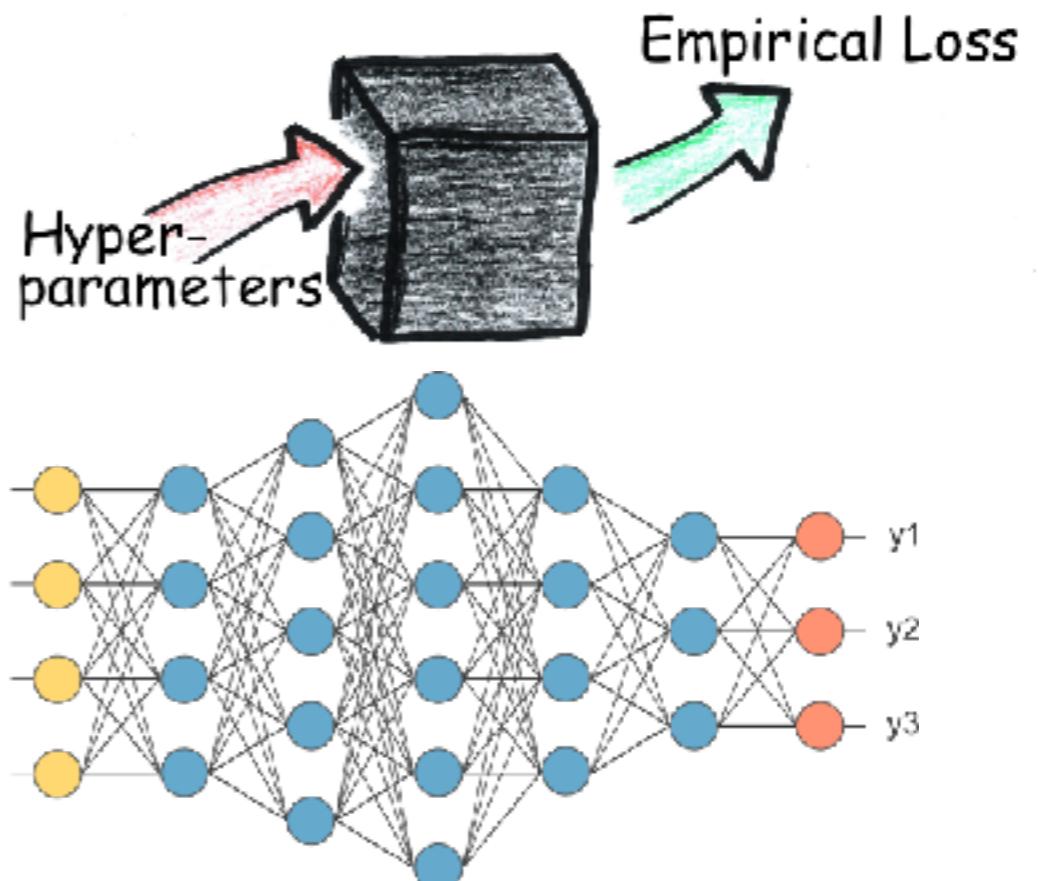
a.k.a derivative-free optimization,
experimental design,
(continuous) multi-armed bandit

Goal: $x^* = \operatorname{argmax}_{\text{box } \mathcal{X} \subset \mathbb{R}^d \text{ or } |\mathcal{X}| = M} f(x)$



Challenges:

- f is expensive to evaluate
- f is multi-peak
- no gradient information
- evaluations can be noisy



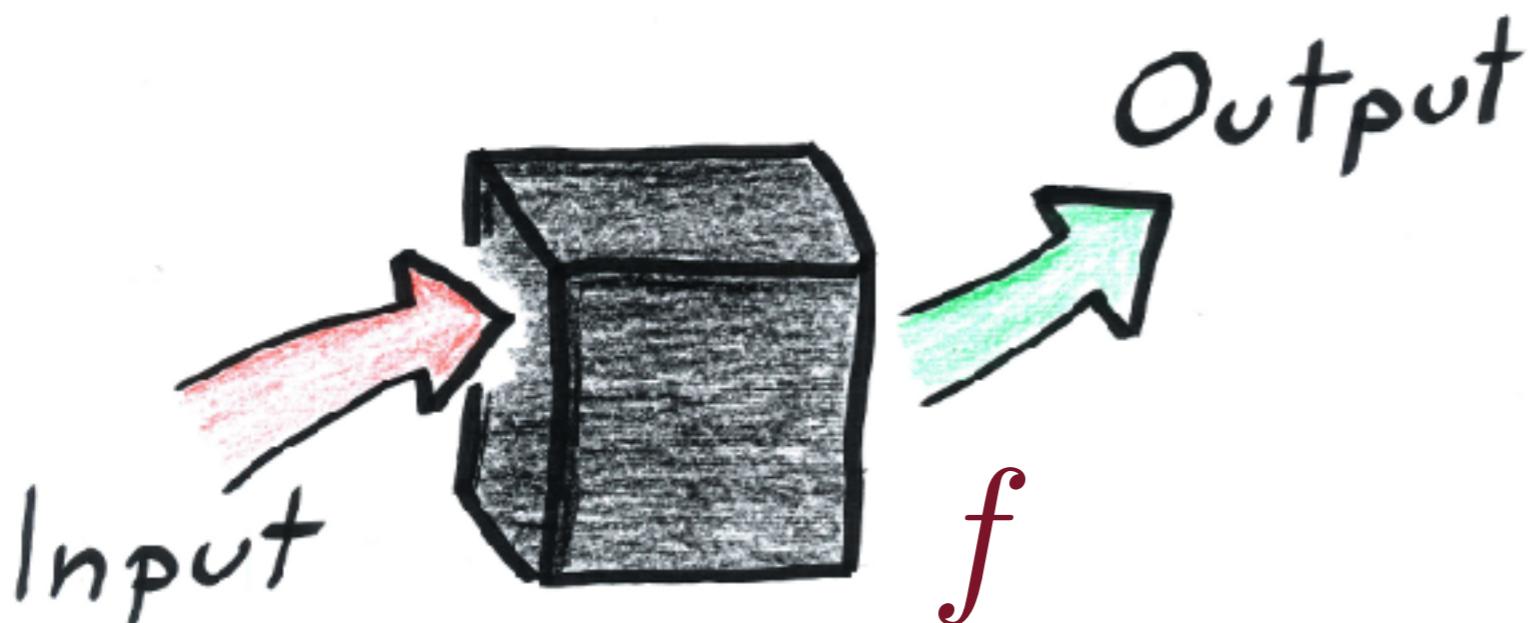
(Kim et al., 2017; Snoek et al., 2012; Gonzalez et al., 2015)

Bayesian Optimization

Idea: build a **probabilistic model** of the function f

LOOP

- choose new query point(s) to evaluate
- update model

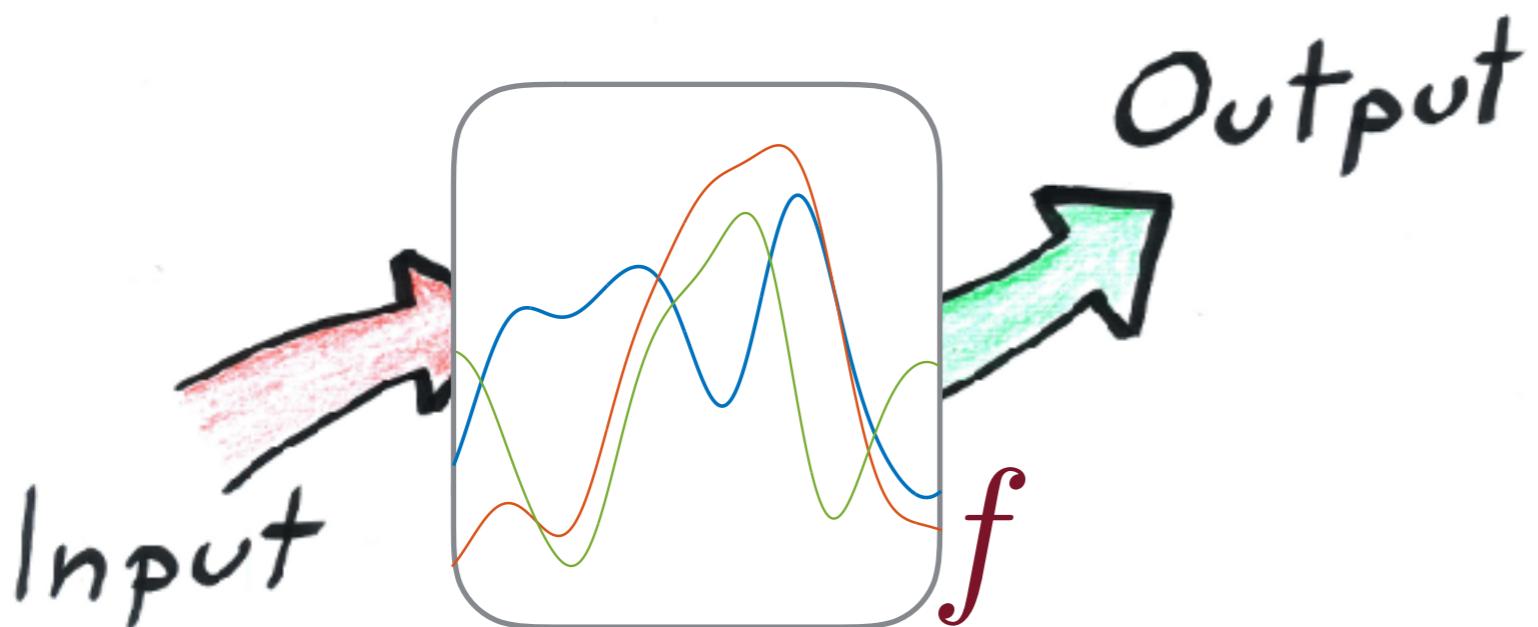


Bayesian Optimization

Idea: build a **probabilistic model** of the function f

LOOP

- choose new query point(s) to evaluate
- update model

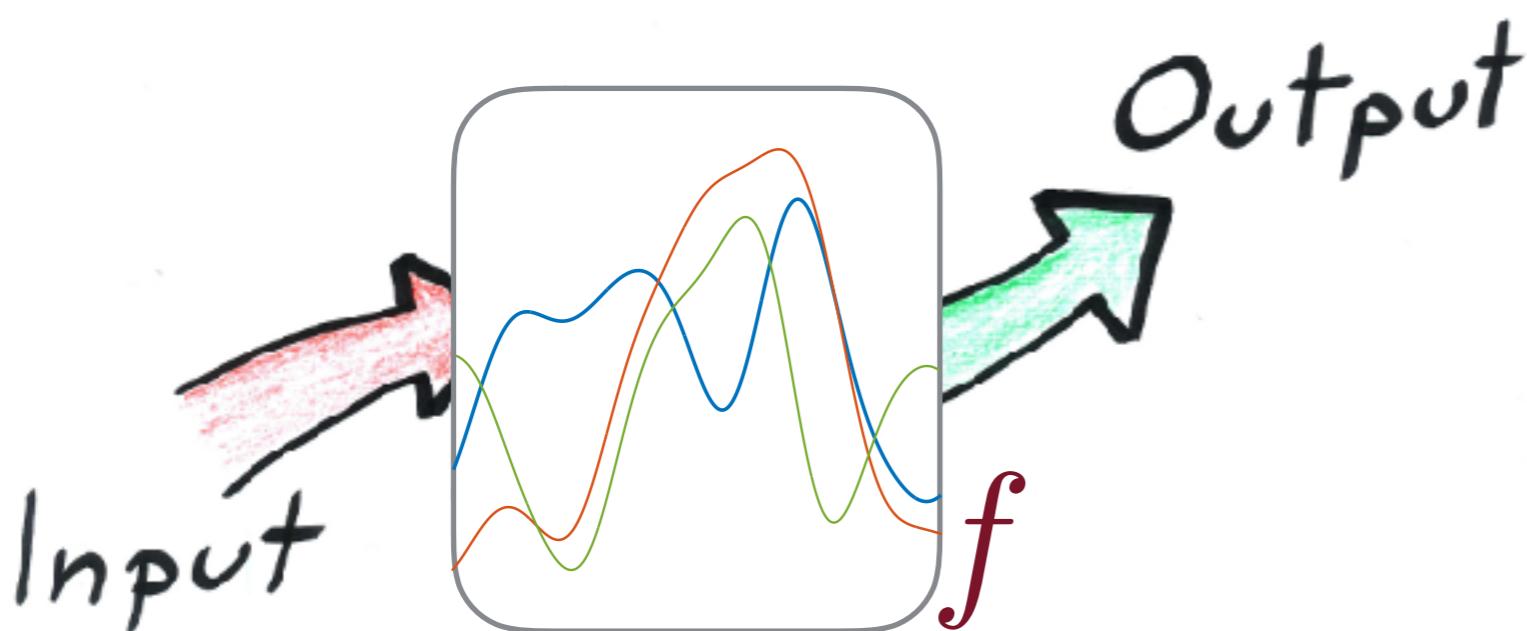


Bayesian Optimization

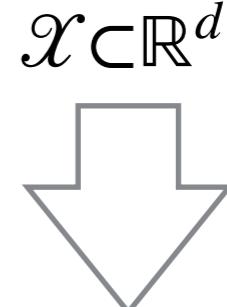
Idea: build a **probabilistic model** of the function f

LOOP

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



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

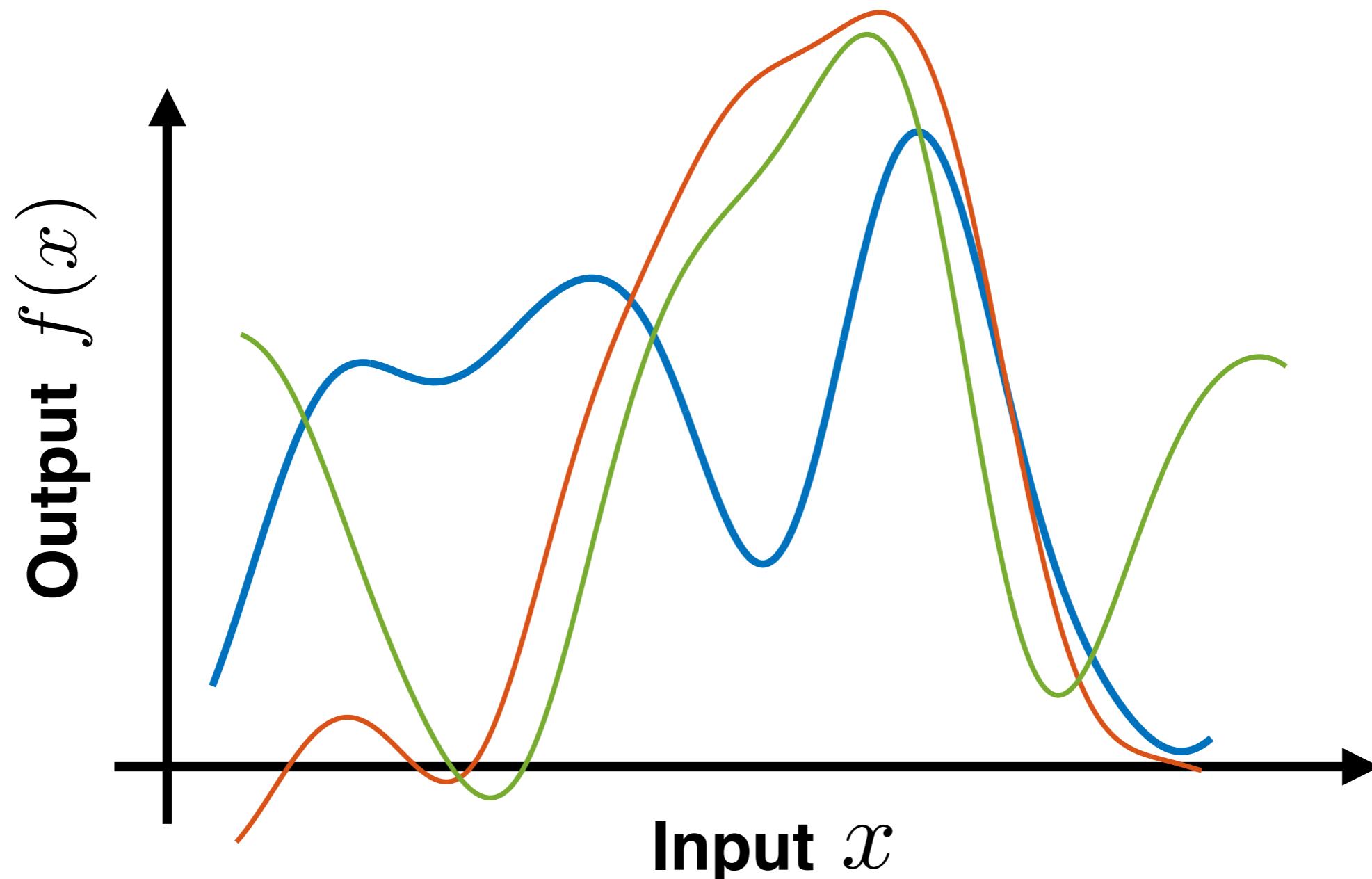


$$x_t = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} \alpha_t(x)$$

$$t = 1, \dots, T$$

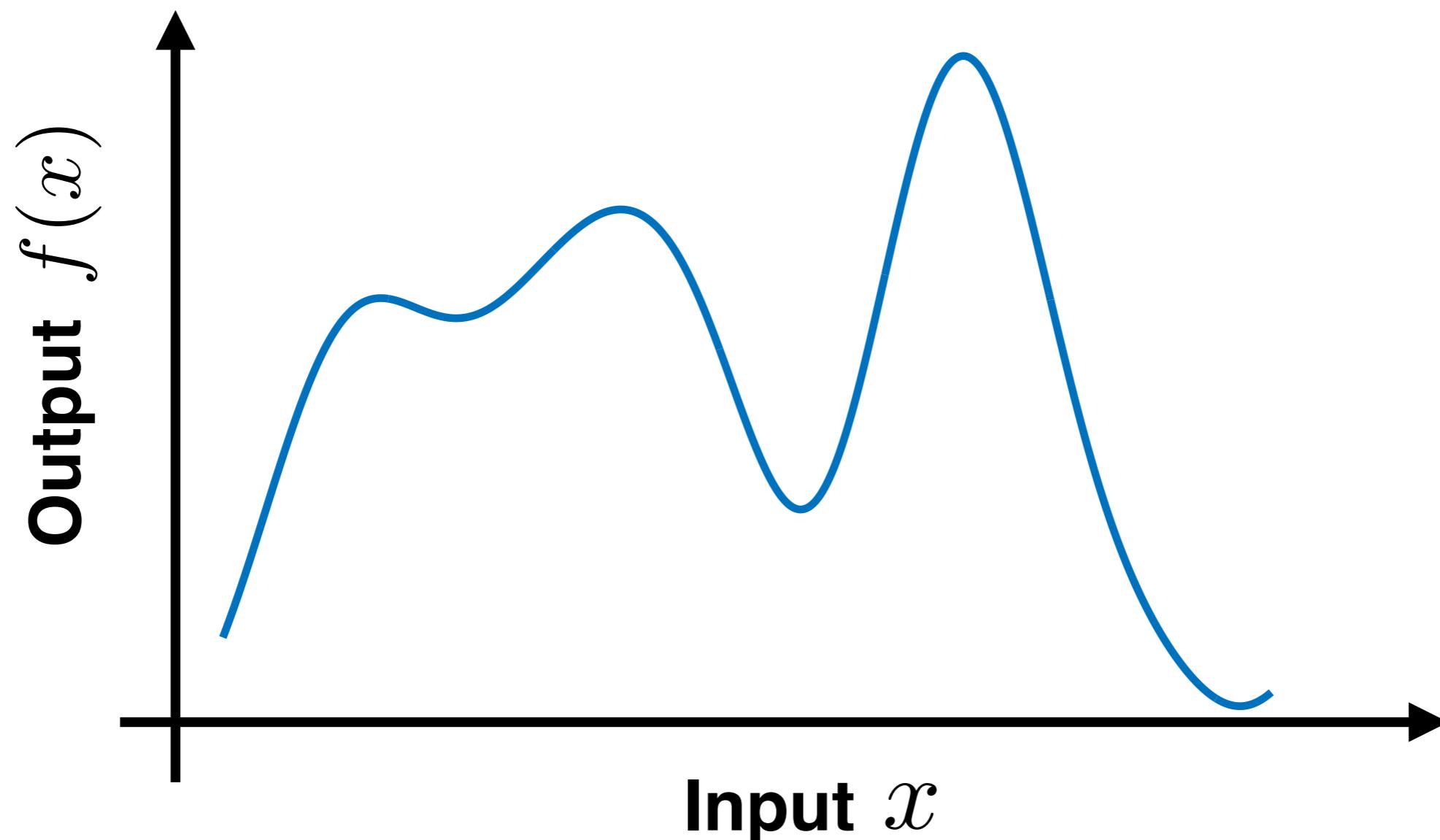
Gaussian Processes (GPs)

- probability distribution over functions
- any finite set of function values has a multivariate Gaussian distribution



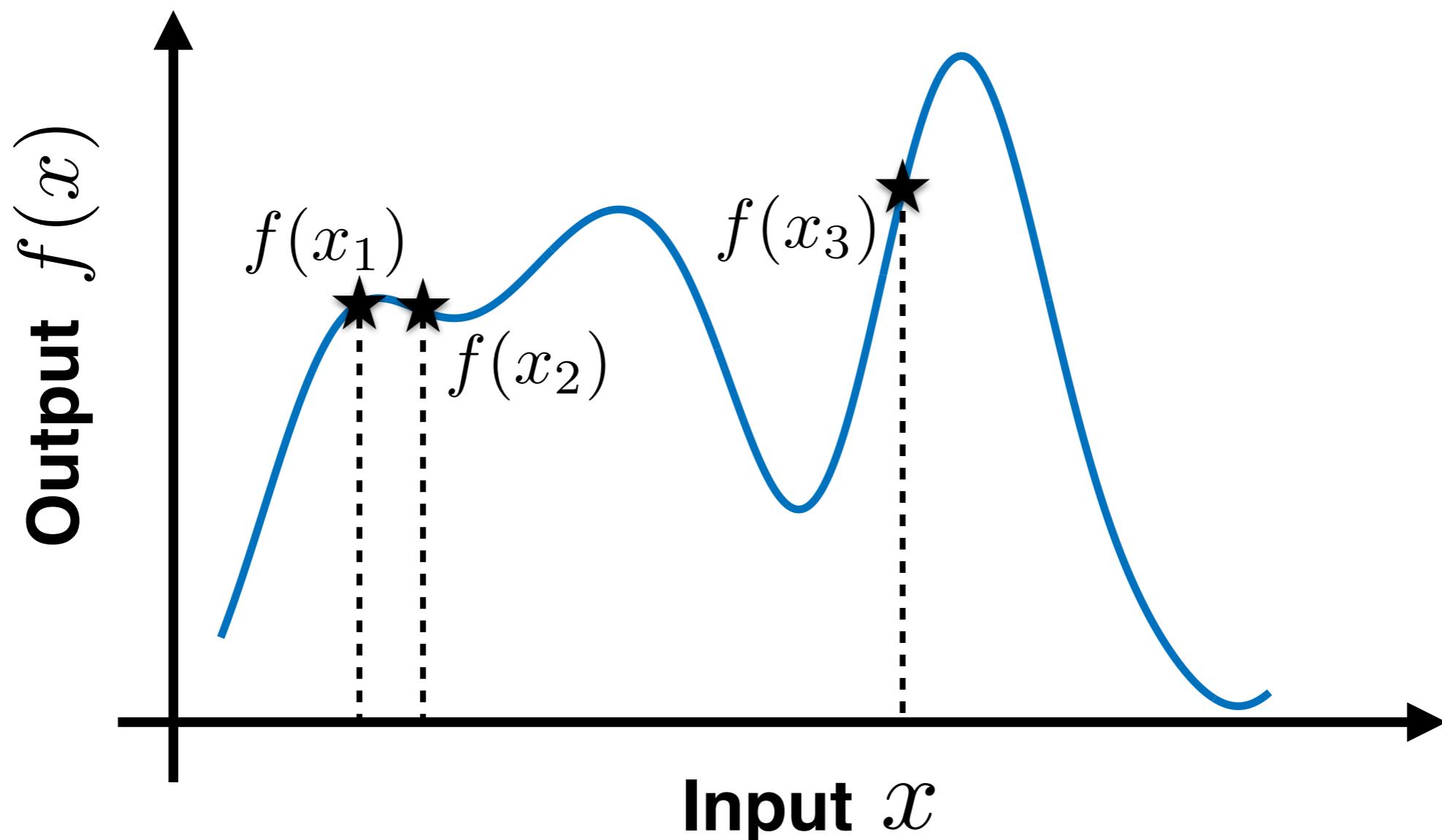
Gaussian Processes (GPs)

- probability distribution over functions
- any finite set of function values has a multivariate Gaussian distribution



Gaussian Processes (GPs)

- probability distribution over functions
- any finite set of function values has a multivariate Gaussian distribution



Gaussian Processes (GPs)

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

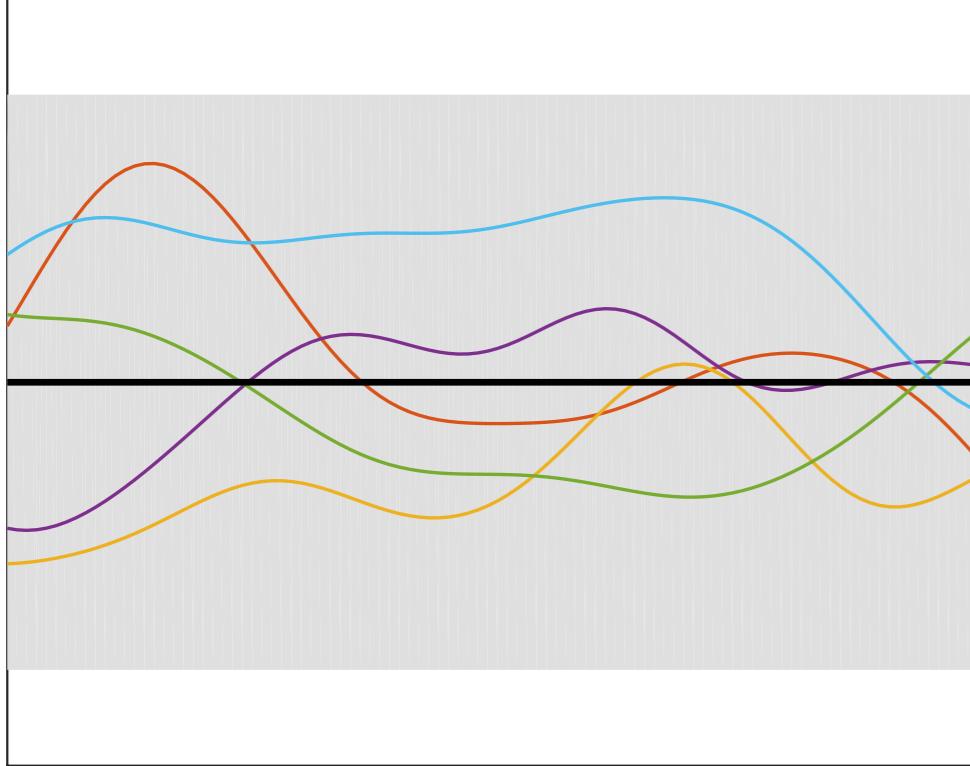
$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1), & \cdots, & k(x_1, x_n) \\ \vdots, & & \vdots \\ k(x_n, x_1), & \cdots, & k(x_n, x_n) \end{bmatrix} \right)$$

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

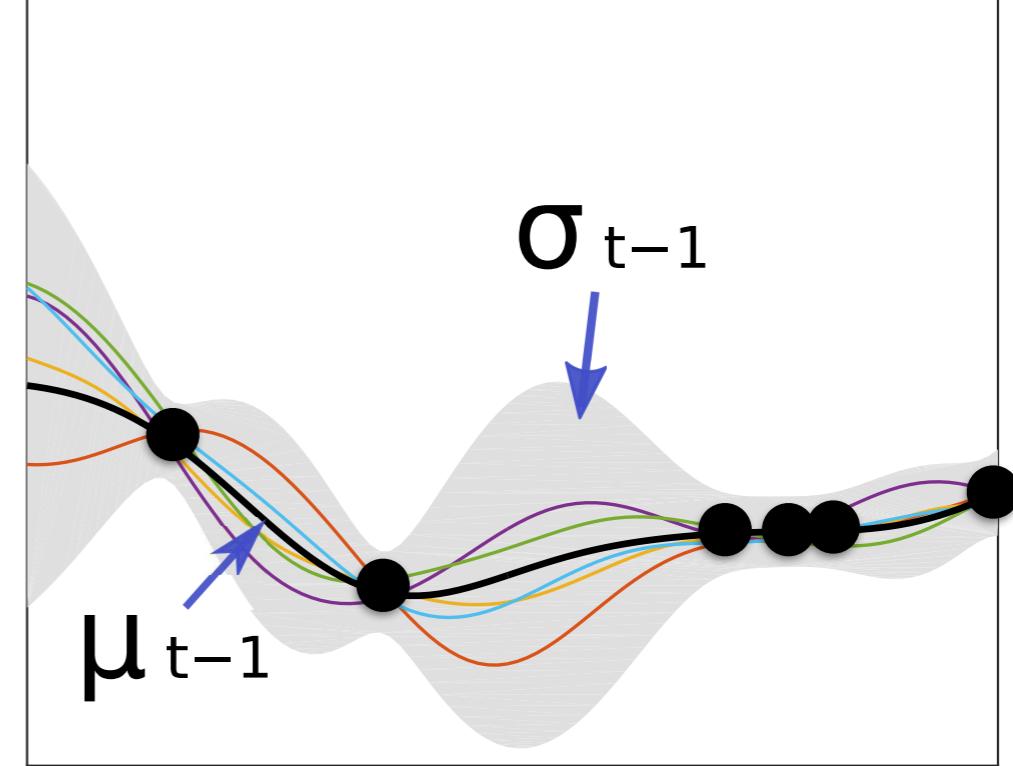
$$y_\tau = f(x_\tau) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

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

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

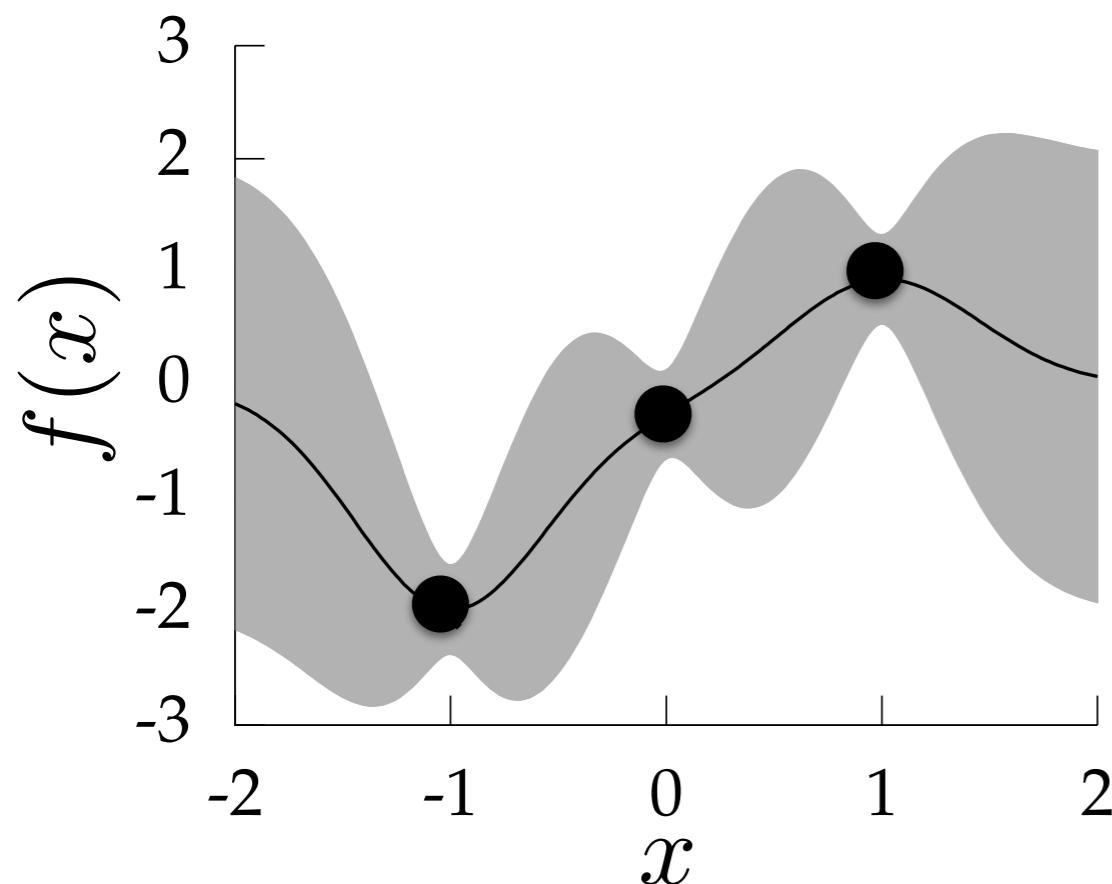
$$\sigma_{t-1}(x)^2 = k(x, x) - k_{t-1}(x)^T(K_{t-1} + \sigma^2 I)^{-1}k_{t-1}(x)$$

Bayesian Optimization

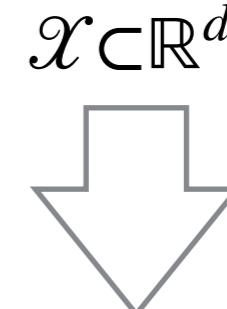
Idea: build a **probabilistic model** of the function f

LOOP

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



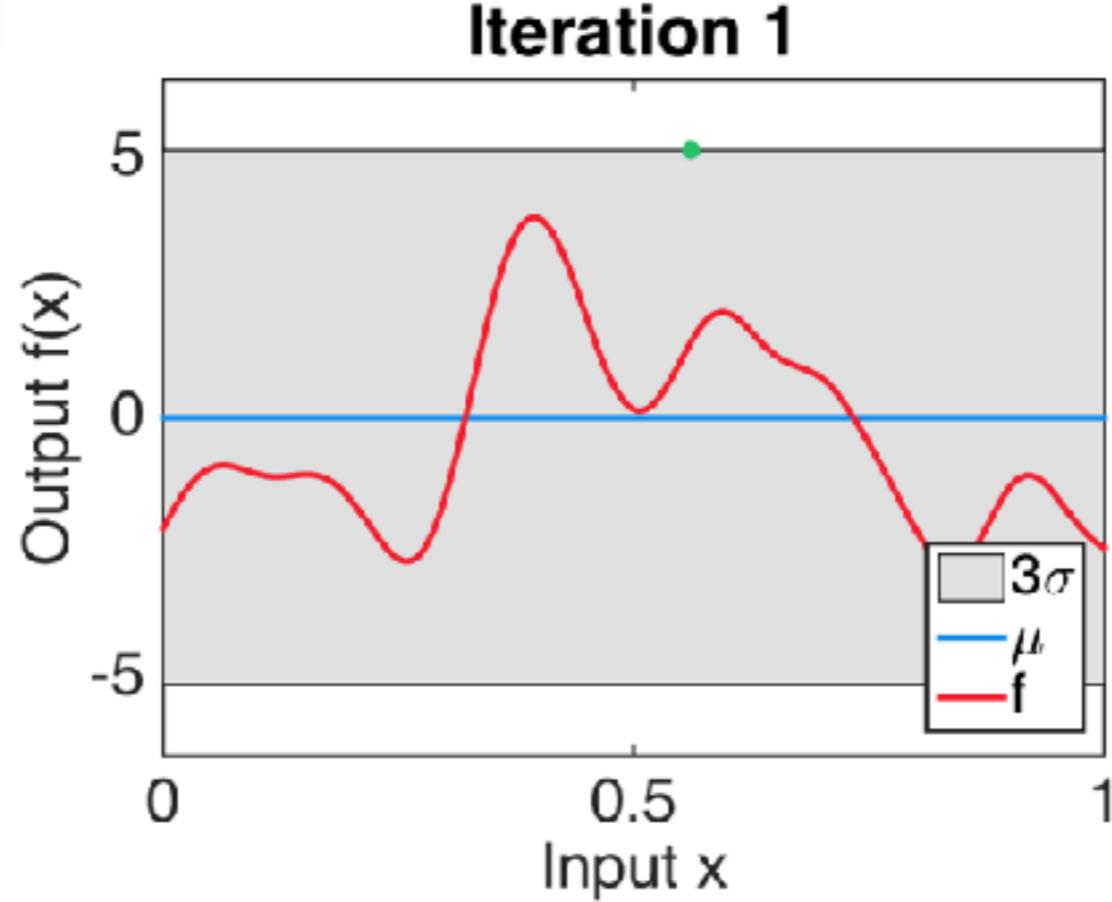
$$x^* = \operatorname{argmax}_{\mathcal{X} \subset \mathbb{R}^d} f(x)$$



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

$$t = 1, \dots, T$$

Gaussian process upper confidence bound



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

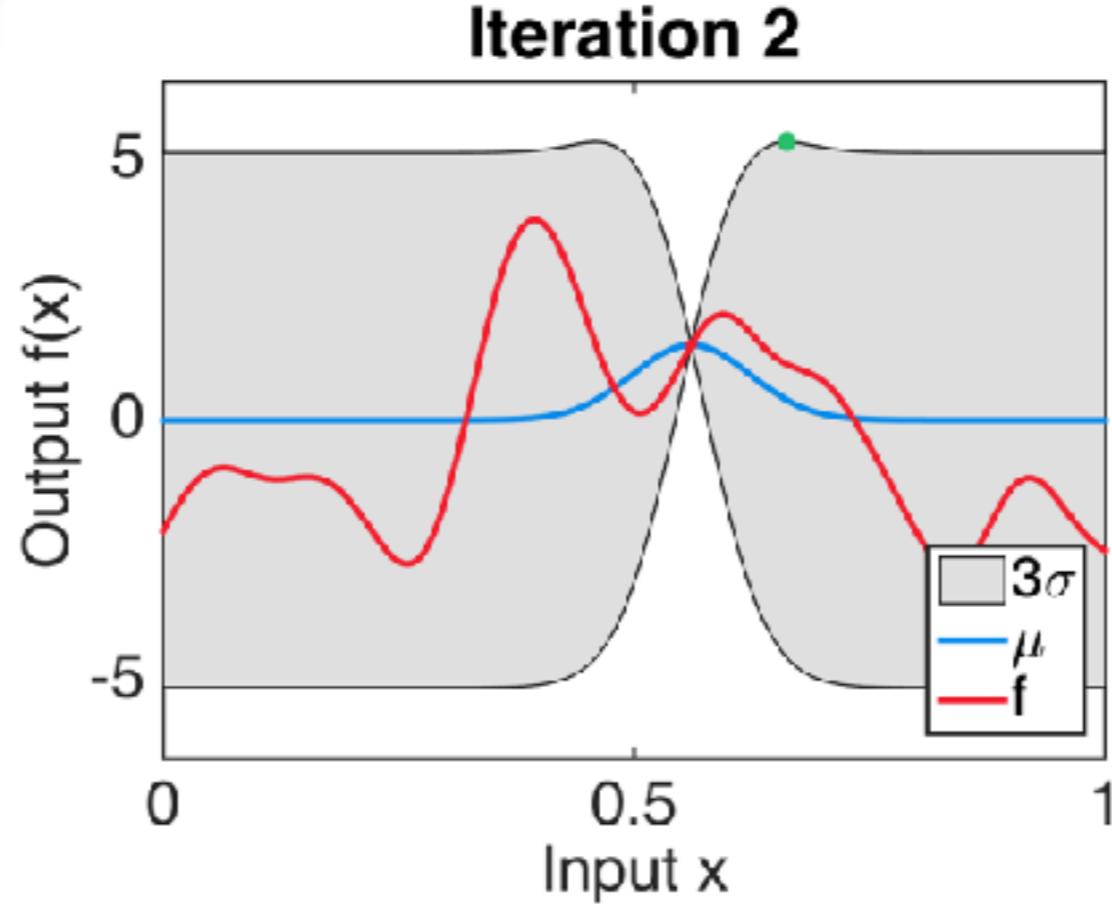
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

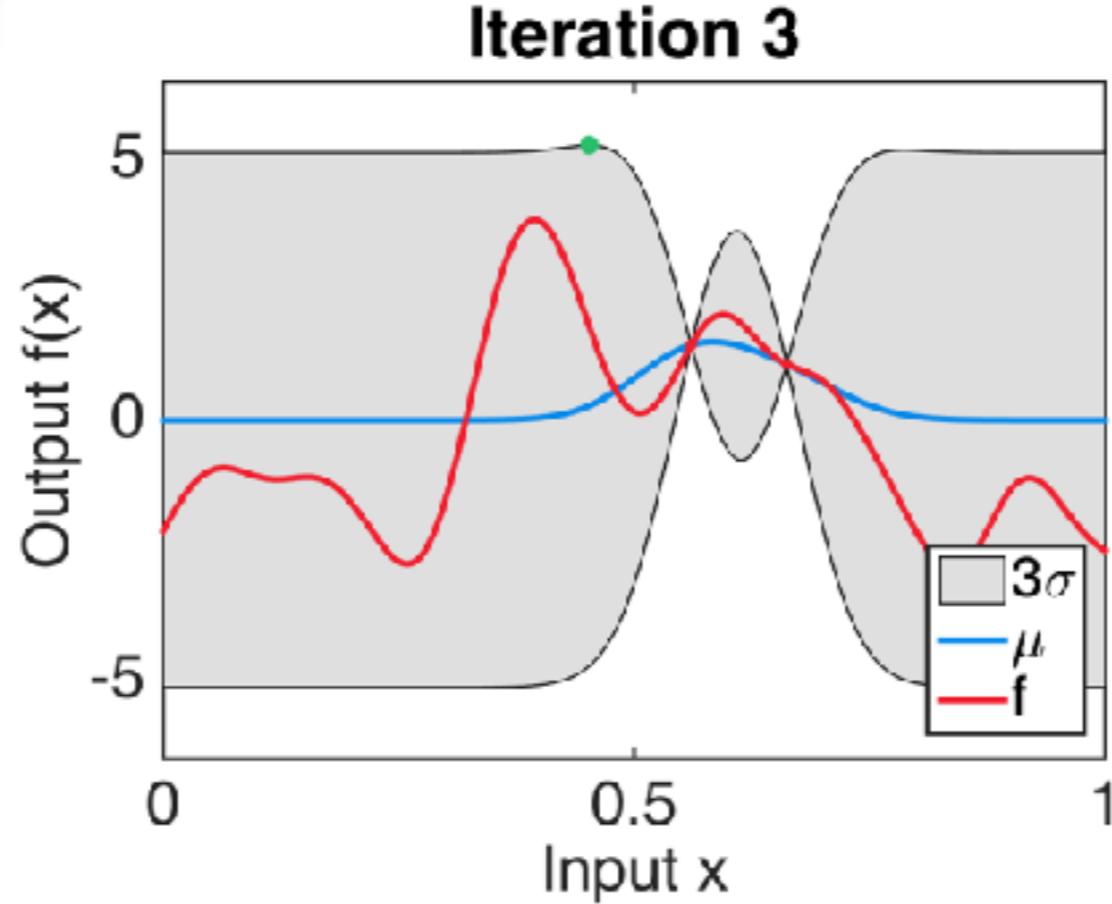
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

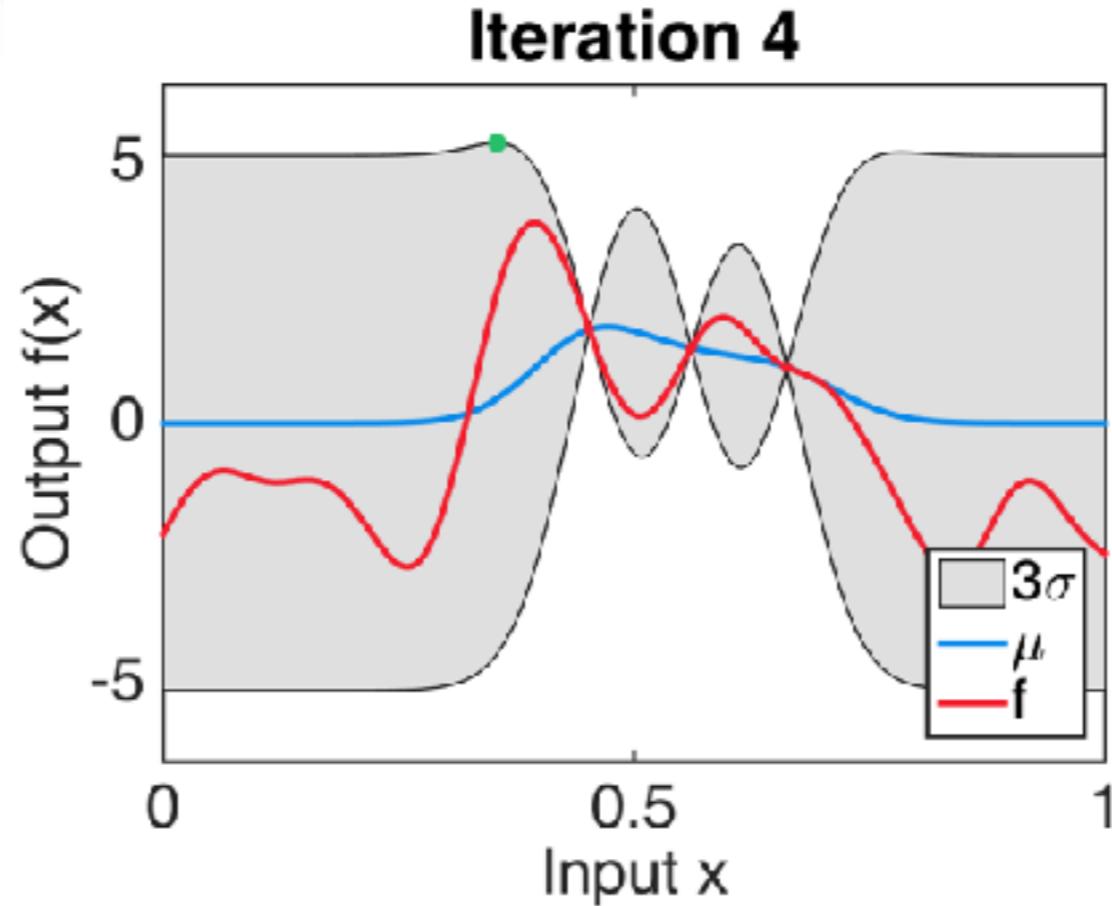
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

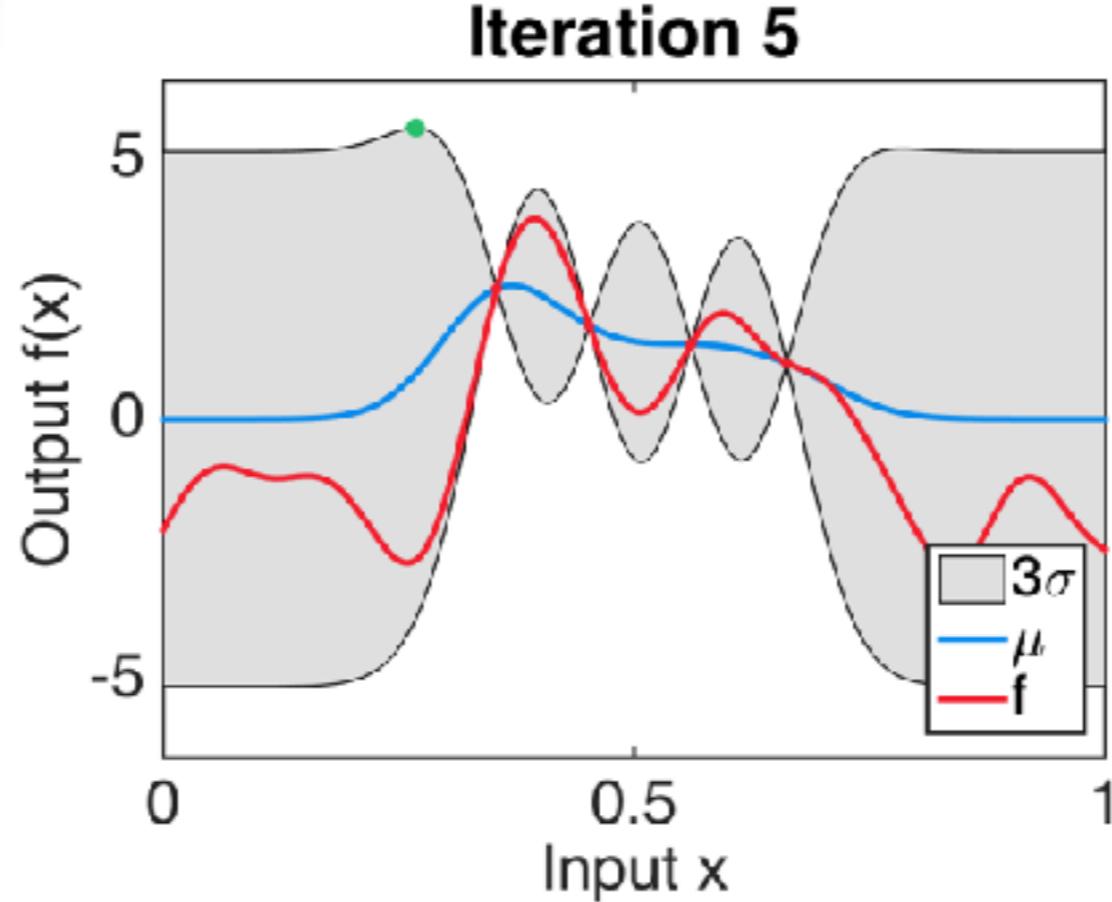
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

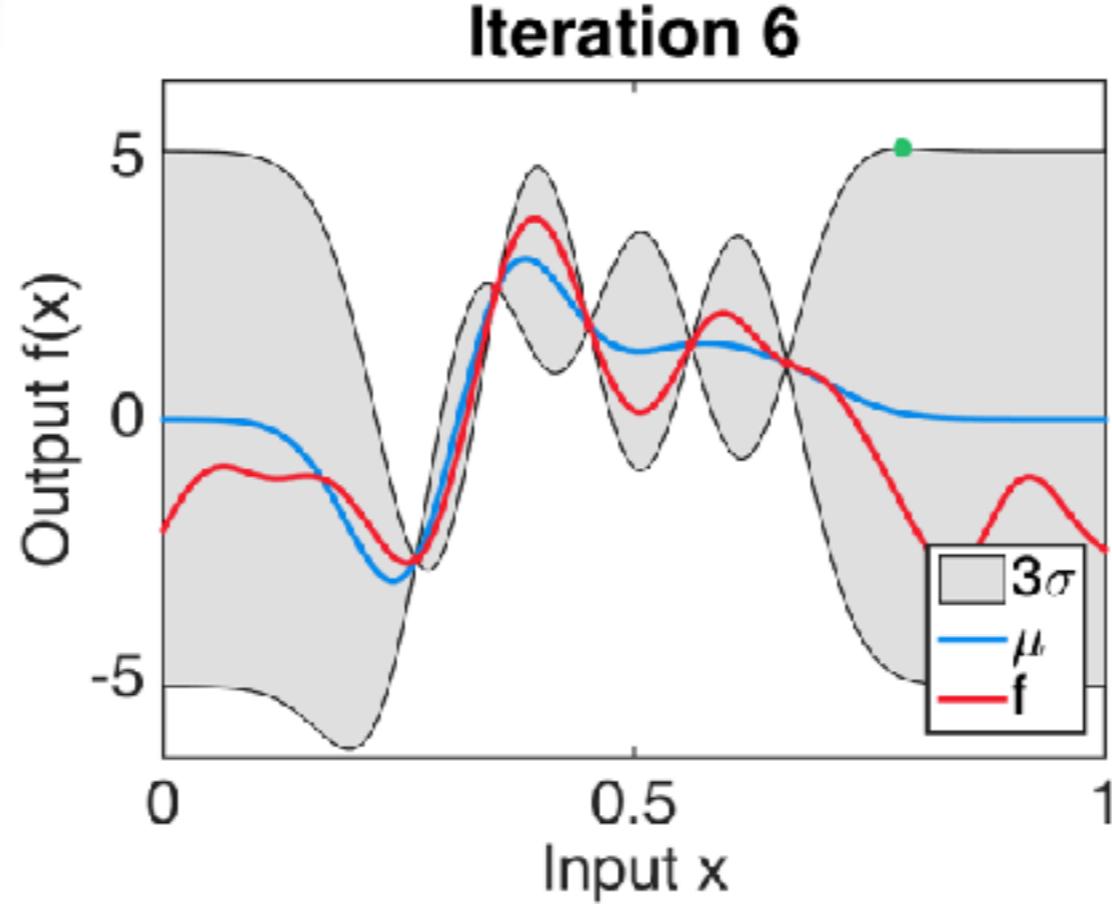
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

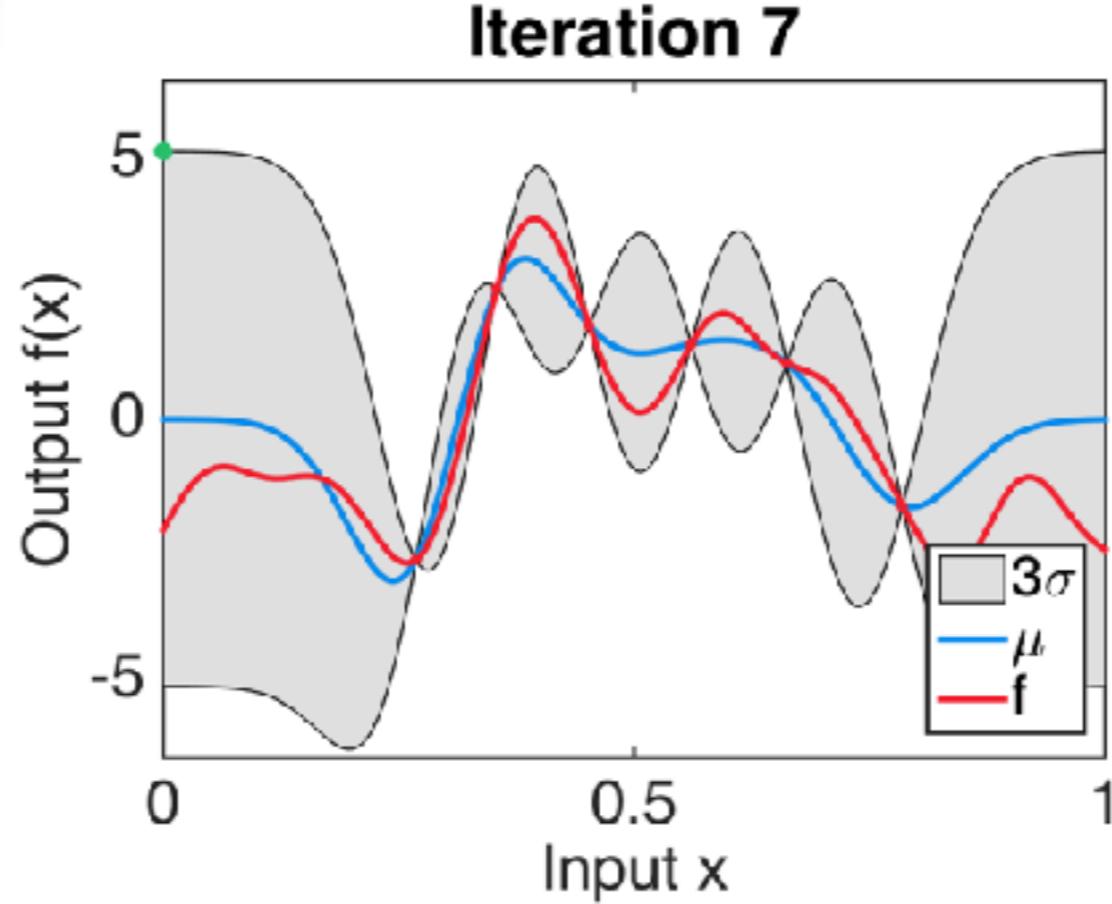
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

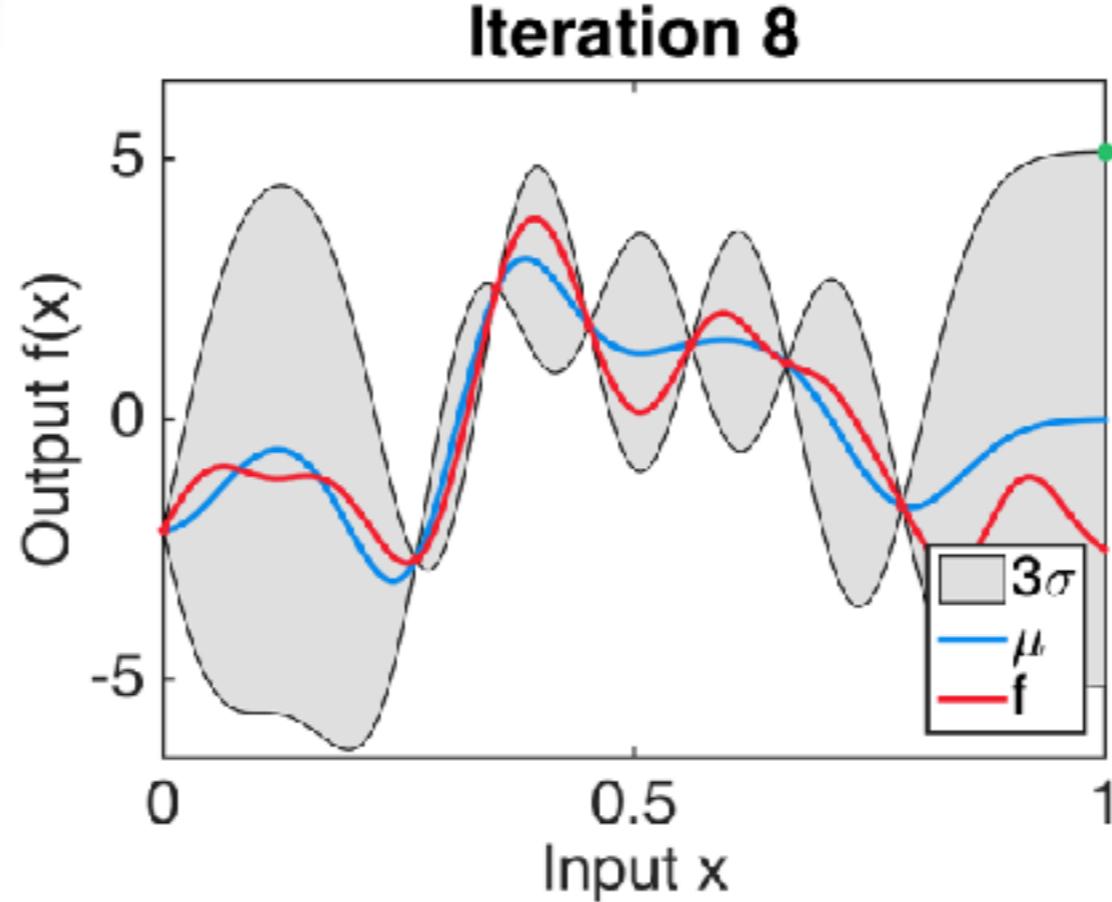
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

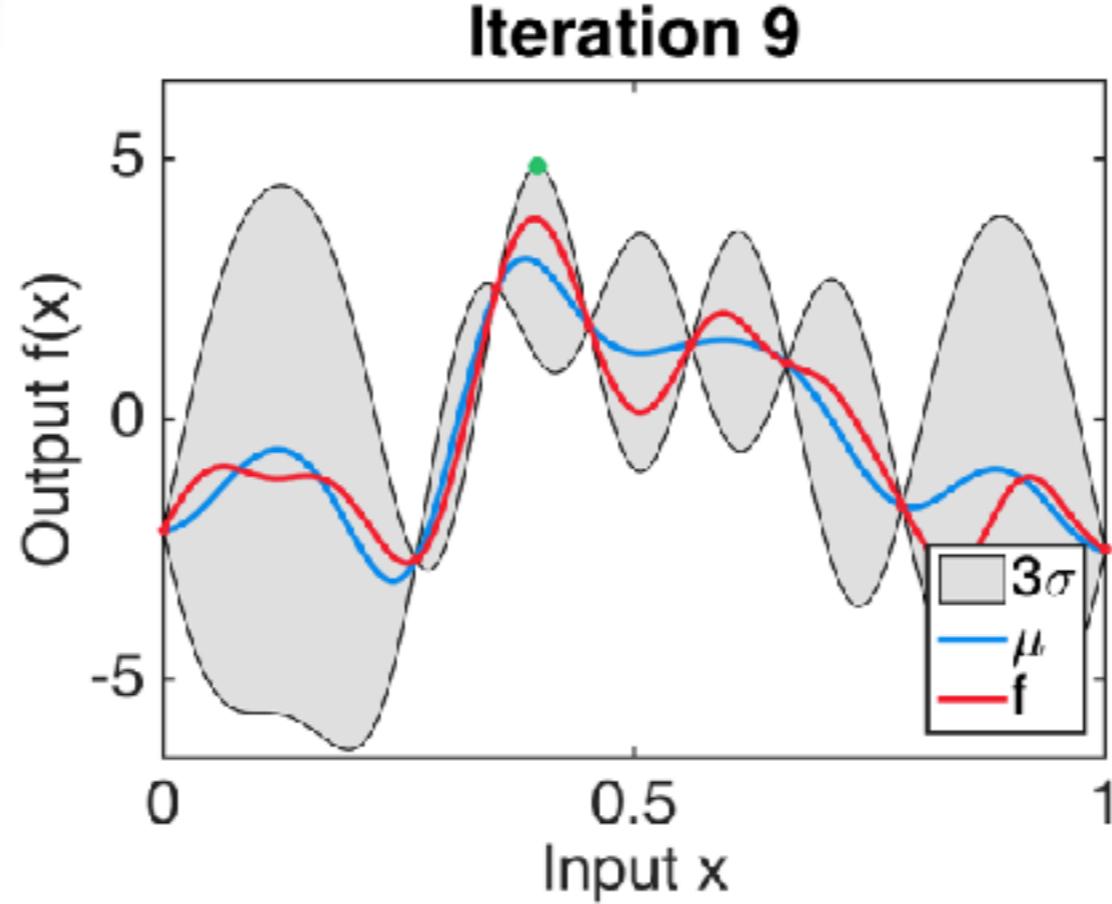
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

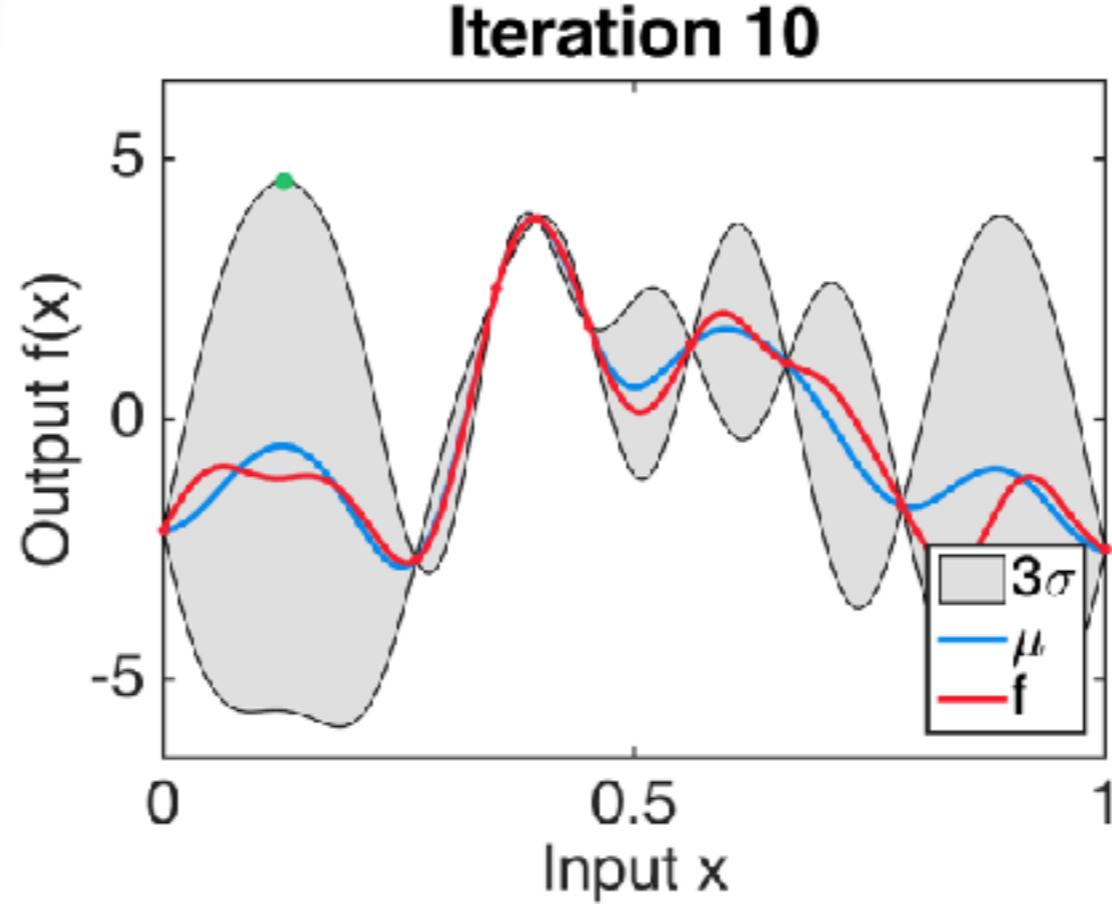
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

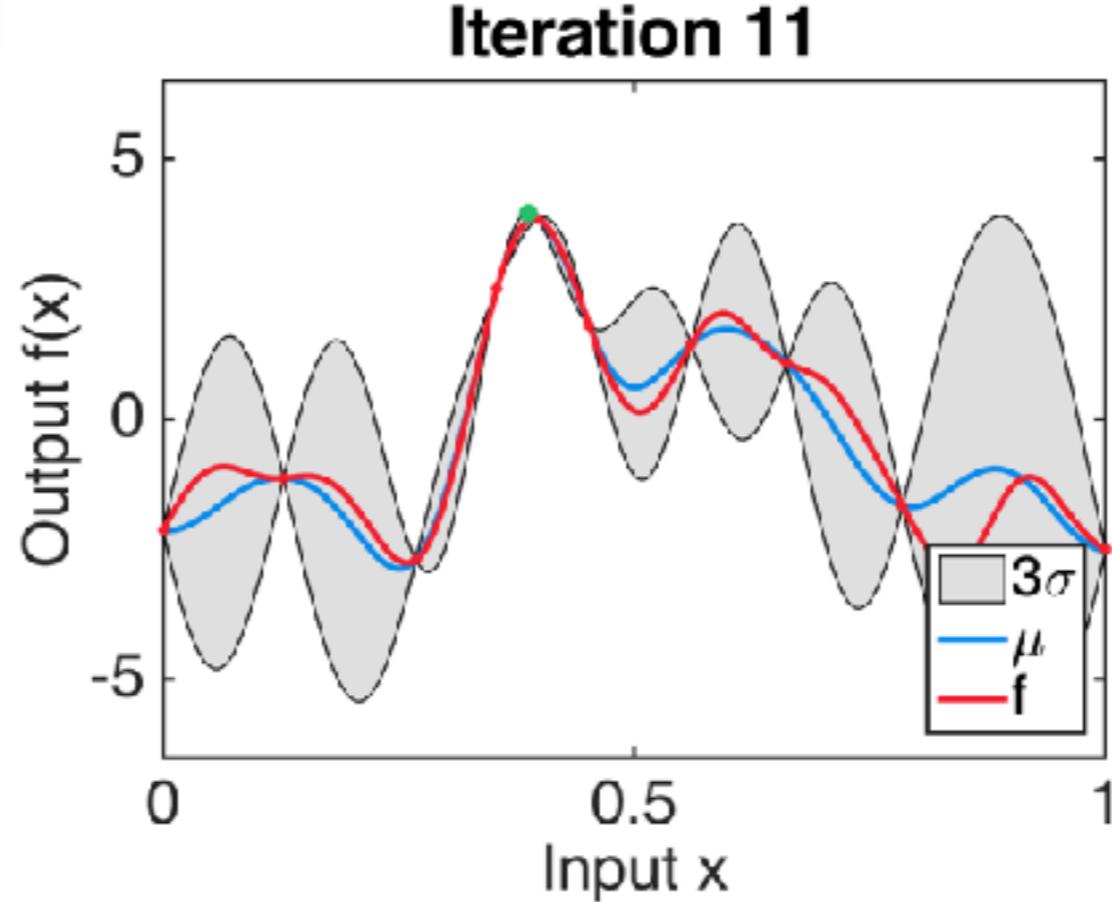
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

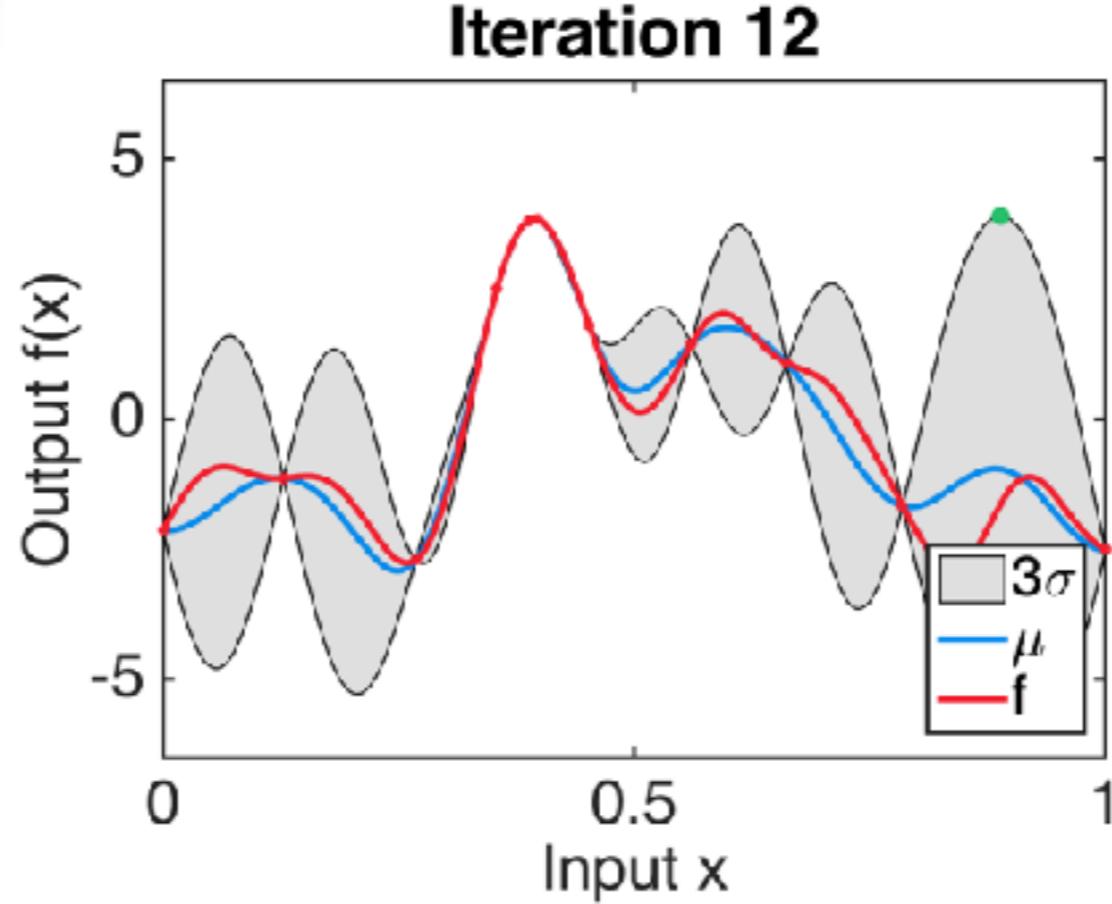
At iteration t ,

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

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

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

Gaussian process upper confidence bound



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

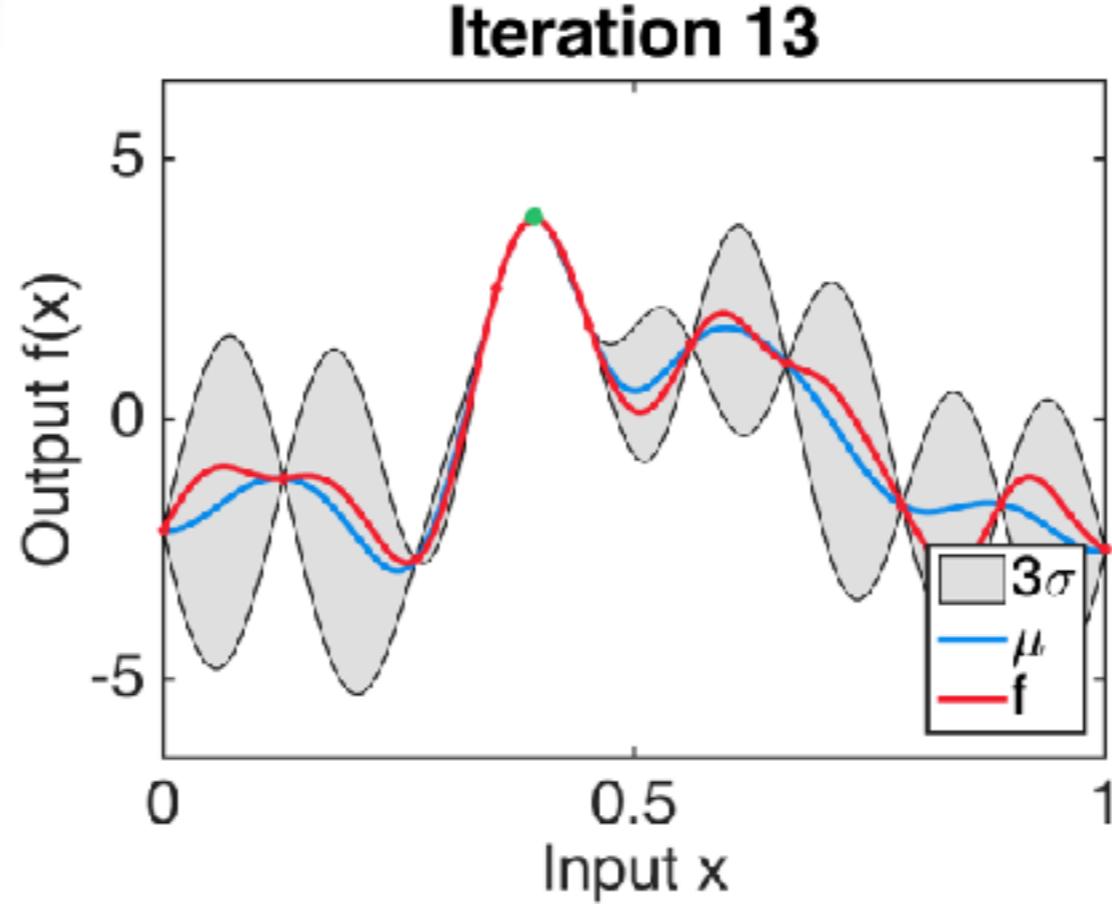
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

At iteration t ,

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

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

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

Challenges in Bayesian optimization

- better acquisition function
- high dimensional input space
- large scale observations
- batch selection of queries
- prior estimation
- ...

Challenges in Bayesian optimization

check out ziw.fyi

- better acquisition function
- high dimensional input space
- large scale observations
- batch selection of queries
- prior estimation
- ...

| Optimization as Estimation with Gaussian Processes in Bandit Settings | Max-value Entropy Search for Efficient Bayesian Optimization |
|---|---|
| Zi Wang MIT CSAIL | Yi Zhou MIT CSAIL |
| Abstract | Abstract |
| <p>Recently, there has been rising interest in Bayesian optimization—the optimization of UNKNOWN OBJECTS WITH UNKNOWN COSTS usually expressed by a Gaussian Process (GP) prior. We study an optimization strategy that directly uses an estimate of the gradient of the function. This strategy offers both practical and theoretical advantages: no smoothness needs to be assumed. However, we establish three limitations to the popular GP-LCB and GP-PI strategies. Our approach can be understood as automatically and adaptively trading off exploration and exploitation in GP-UCB and GP-PI. We illustrate the effects of this adaptive tuning via bench tests on the mean as well as an extensive empirical evaluation on robotics and vision tasks, demonstrating the robustness of this strategy for a range of performance metrics.</p> | <p>Entropy Search (ES) and Predictive Entropy Search (PES) are popular and empirically successful Bayesian Optimization techniques. They aim at a competing information-theoretic motivation, and maximize the information gained about the key value of the unknown function; yet, both are plagued by the expensive computation for estimating entropies. We propose a new criterion, Max-value Entropy Search (MES), that instead uses the information about the maximum function values. We show relations of MES to other Bayesian optimization methods, and establish a regret bound. We observe that MES minimizer improves the good empirical performance of ES/PES, while considerably lightening the computational burden. In particular, MES is much more robust to the number of samples used for computing the entropy and hence more efficient for higher-dimensional problems.</p> |
| Batched High-dimensional Bayesian Optimization via Structural Kernel Learning | Batched Large-scale Bayesian Optimization in High-dimensional Space |
| Zi Wang ^{*†} , Chengtao Li ^{*†} , Stefanie Jegelka [†] , Prasanna Kothli [†] | Zi Wang MIT CSAIL Clement Gehring MIT CSAIL Prasanna Kothli DeepMind Stefanie Jegelka MIT CSAIL |
| Abstract | Abstract |
| <p>Optimization of high-dimensional black-box functions is an extremely challenging problem. While Bayesian optimization has emerged as a popular approach for optimizing black-box functions, its applicability has been limited to low-dimensional problems due to its computational and statistical challenges arising from high-dimensional settings. In this paper, we propose to tackle those challenges by (1) assuming a latent additive structure of the function and inferring it properly for more efficient and effective BO, and (2) performing multiple evaluations in parallel to reduce the number of iterations to converge the method. Our novel approaches use the latent structure with ODE sampling and various non-standard nonparametric Gaussian process priors. Experimental validation on different datasets shows that our proposed approach is competitive with state-of-the-art solvers.</p> | <p>Bayesian optimization (BO) has become an effective approach for black-box function optimization problems when function evaluations are expensive and the optimum can be achieved within a relatively small number of queries. However, many cases, such as the ones with high-dimensional inputs, may require a much larger number of observations for optimization. Despite an abundance of BO techniques have been limited to merely a few thousand observations. In this paper, we propose extensible Bayesian optimization (EBO) to address three current challenges in BO simultaneously: (1) large-scale observations; (2) high-dimensional input spaces; and (3) selection of confidence bounds (an essential part of the functions), and expensive or inaccurate Gaussian</p> |
| [Wang&Jegelka, ICML 2017] | [Wang&Zhou&Jegelka, oral@AISTATS 2016] |
| [Wang&Gehring&Kohli&Jegelka, AISTATS 2018] | [Wang*&Li*&Jegelka&Kohli, ICML 2017] |

Challenges in Bayesian optimization

check out ziw.fyi

- better acquisition function
- high dimensional input space
- large scale observations
- batch selection of queries
- prior estimation
- ...

[Wang*&Kim*&Kaelbling, spotlight@NIPS 2018]

this talk

| Optimization as Estimation with Gaussian Processes in Bandit Settings | Max-value Entropy Search for Efficient Bayesian Optimization |
|---|---|
| Zi Wang MIT CSAIL | Brian Zhou MIT CSAIL |
| Abstract Recently, there has been rising interest in Bayesian optimization—the optimization of UNKNOWN OBJECTS WITH UNKNOWN COSTS—expressed by a Gaussian Process (GP) prior. We study an optimization strategy that directly uses an estimate of the gradient of the function. This strategy offers both practical and theoretical advantages: no gradient estimator needs to be implemented. However, we establish three shortcomings to the popular GP-LCB and GP-PI strategies. Our approach can be understood as automatically and adaptively trading off exploration and exploitation in GP-UCB and GP-PI. We illustrate the effects of this adaptive tuning through experiments on the mean as well as an extensive empirical evaluation on robotics and vision tasks, demonstrating the robustness of this strategy for a range of performance metrics. | Abstract Entropy Search (ES) and Predictive Entropy Search (PES) are popular and empirically successful Bayesian Optimization techniques. They aim at a competing information-theoretic motivation, and maximize the information gained about the argmax of the unknown function; yet, both are plagued by the expensive computation for estimating entropies. We propose a new criterion, Max-value Entropy Search (MES), that instead uses the information about the maximum function value. We show relations of MES to other Bayesian optimization methods, and establish a regret bound. We observe that MES minimizer improves the good empirical performance of ES/PES, while considerably lightening the computational burden. In particular, MES is much more robust to the number of samples used for computing the entropy and hence more efficient for higher-dimensional problems. |
| Batched High-dimensional Bayesian Optimization via Structural Kernel Learning | Batched Large-scale Bayesian Optimization in High-dimensional Space |
| Zi Wang [*] , Chengtao Li [*] , Stefanie Jegelka [†] , Prasanna Kothli [‡] | Zi Wang MIT CSAIL Clement Gehring MIT CSAIL Prasanna Kothli DeepMind Stefanie Jegelka MIT CSAIL |
| Abstract Optimization of high-dimensional black-box functions is an extremely challenging problem. While Bayesian optimization has emerged as a popular approach for optimizing black-box functions, its applicability has been limited to low-dimensional problems due to its computational and statistical challenges arising from high-dimensional settings. In this paper, we propose to tackle those challenges by (1) assuming a latent additive structure of the function and inferring it properly for more efficient and effective BO, and (2) performing multiple evaluations in parallel to reduce the number of iterations to cover the search. Our novel approaches use the latent structure with O(1m) sampling variance to standardize nonstationary Gaussian process priors. Experimental validation on different datasets shows that our proposed approach is competitive with state-of-the-art solvers. | Abstract Bayesian optimization (BO) has become an effective approach for black-box function optimization problems when function evaluations are expensive and the optimum can be achieved within a relatively small number of queries. However, many cases, such as the ones with high-dimensional inputs, may require a much larger number of observations for optimization. Despite an abundance of BO techniques, no parallel experiments, e.g., using parallel experiments, have been limited to merely a few thousand observations. In this paper, we propose extensible Bayesian optimization (EBO) to address three current challenges in BO simultaneously: (1) large-scale observations; (2) high-dimensional input spaces; and (3) selections of confidence bounds (an essential part of the functions), and expensive or inaccurate Gaussian |

[Wang&Jegelka, ICML 2017]

[Wang&Zhou&Jegelka, oral@AISTATS 2016]

[Wang&Gehring&Kohli&Jegelka, AISTATS 2018]

[Wang*&Li*&Jegelka&Kohli, ICML 2017]

Bayesian optimization with an unknown prior

Estimate “prior” from data

- maximum likelihood
- hierarchical Bayes

Bayesian optimization with an unknown prior

Estimate “prior” from data

- maximum likelihood
- hierarchical Bayes

Which comes first?
Data or prior?



Bayesian optimization with an unknown prior

Estimate “prior” from data

- maximum likelihood
- hierarchical Bayes
- Regret bounds exist only when prior is assumed given

**Which comes first?
Data or prior?**



Bayesian optimization with an unknown prior

Estimate “prior” from data

- maximum likelihood
- hierarchical Bayes

- Regret bounds exist only when prior is assumed given
- bad settings of priors make BO perform badly and seem to be a bad approach

**Which comes first?
Data or prior?**



Bayesian optimization with an unknown prior

Estimate “prior” from data

- maximum likelihood
- hierarchical Bayes

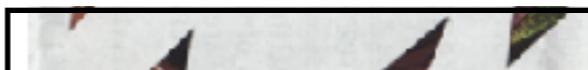
- Regret bounds exist only when prior is assumed given
- bad settings of priors make BO perform badly and seem to be a bad approach

Which comes first?
Data or prior?



Ben Recht
[@beenwrekt](https://twitter.com/beenwrekt)

Bayesian Optimization and other bad ideas for tuning hyperparameters.



About

Embracing the Random

Kevin Jamieson and Ben Recht • Jun 23, 2016

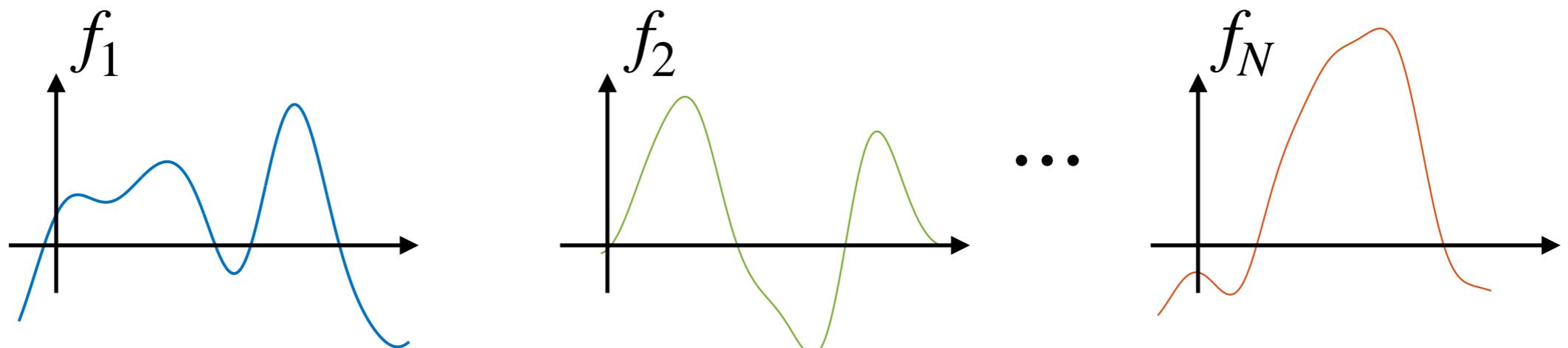
Bayesian optimization with an unknown prior

- Our idea: learn the “prior” from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior

[Wang*&Kim*&Kaelbling, NIPS 2018]

Bayesian optimization with an unknown prior

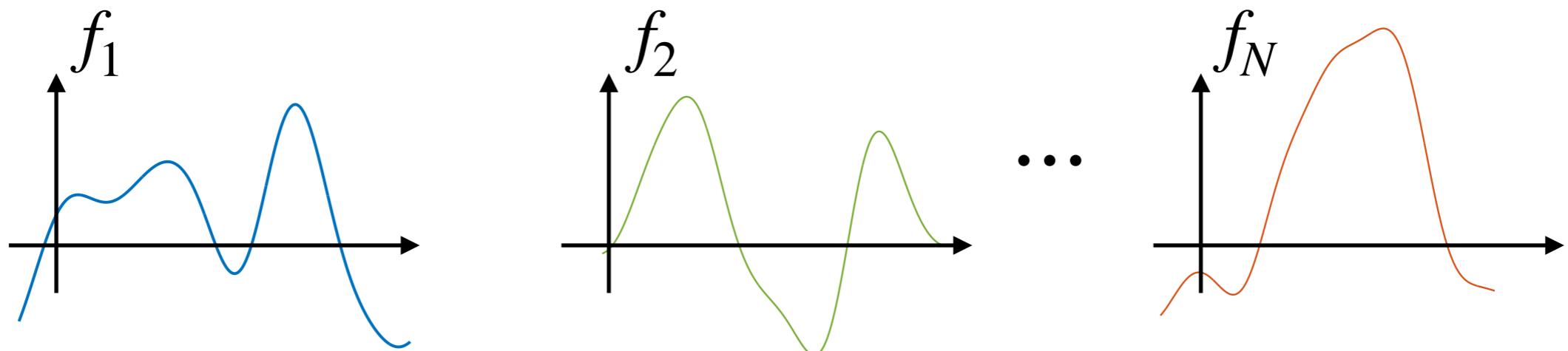
- Our idea: learn the “prior” from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \dots, f_N \sim GP(\mu, k)$



[Wang*&Kim*&Kaelbling, NIPS 2018]

Bayesian optimization with an unknown prior

- Our idea: learn the “prior” from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \dots, f_N \sim GP(\mu, k)$

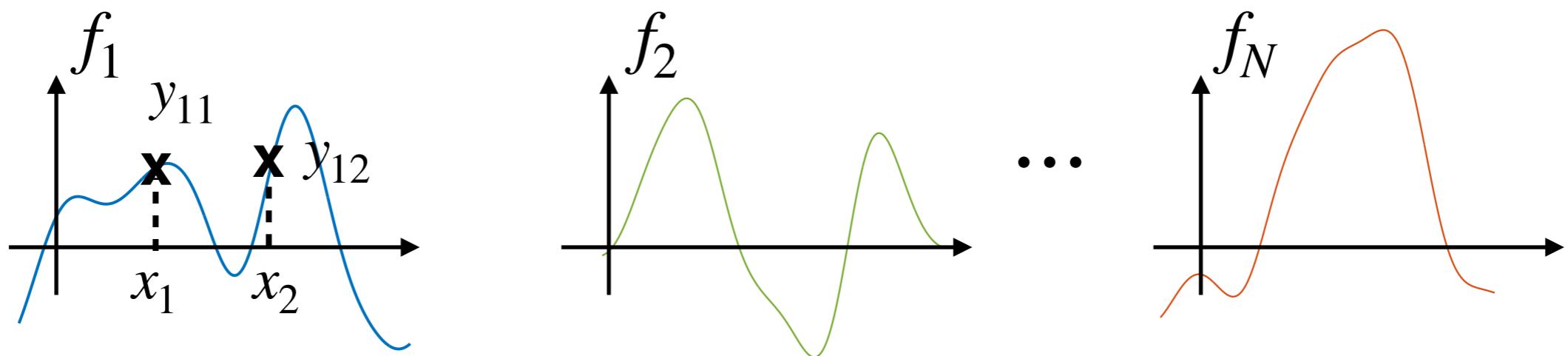


$$[x_1, \dots, x_M] \quad y_{ij} \sim \mathcal{N}(f_i(x_j), \sigma^2)$$

[Wang*&Kim*&Kaelbling, NIPS 2018]

Bayesian optimization with an unknown prior

- Our idea: learn the “prior” from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \dots, f_N \sim GP(\mu, k)$

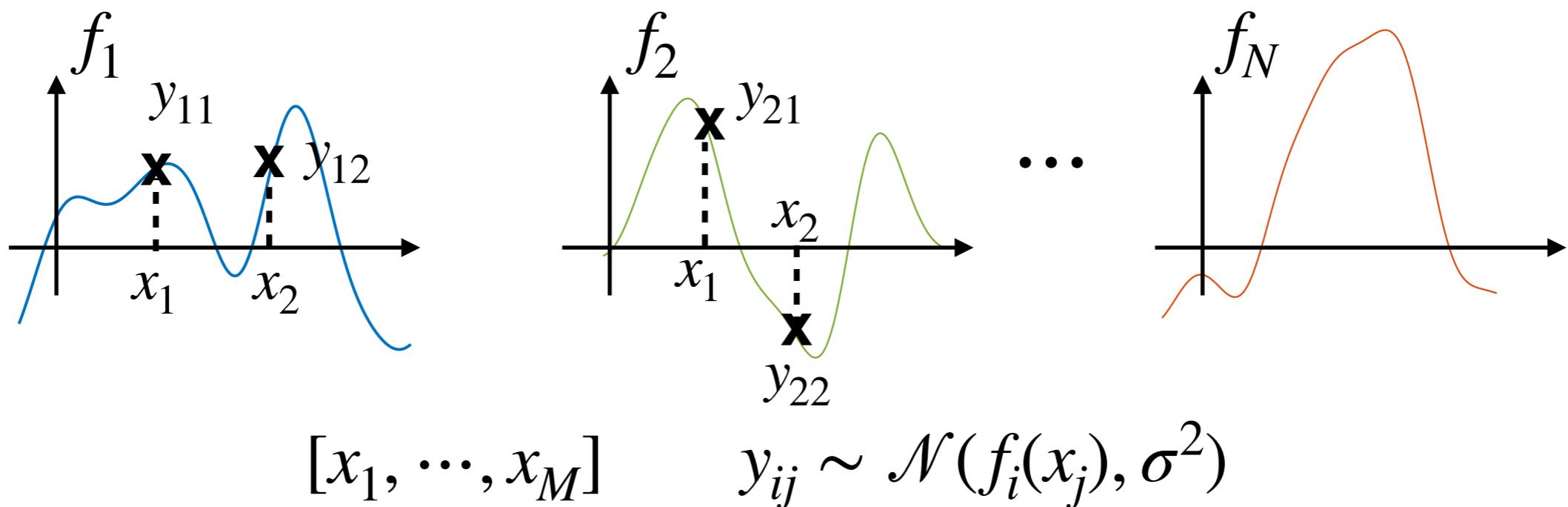


$$[x_1, \dots, x_M] \quad y_{ij} \sim \mathcal{N}(f_i(x_j), \sigma^2)$$

[Wang*&Kim*&Kaelbling, NIPS 2018]

Bayesian optimization with an unknown prior

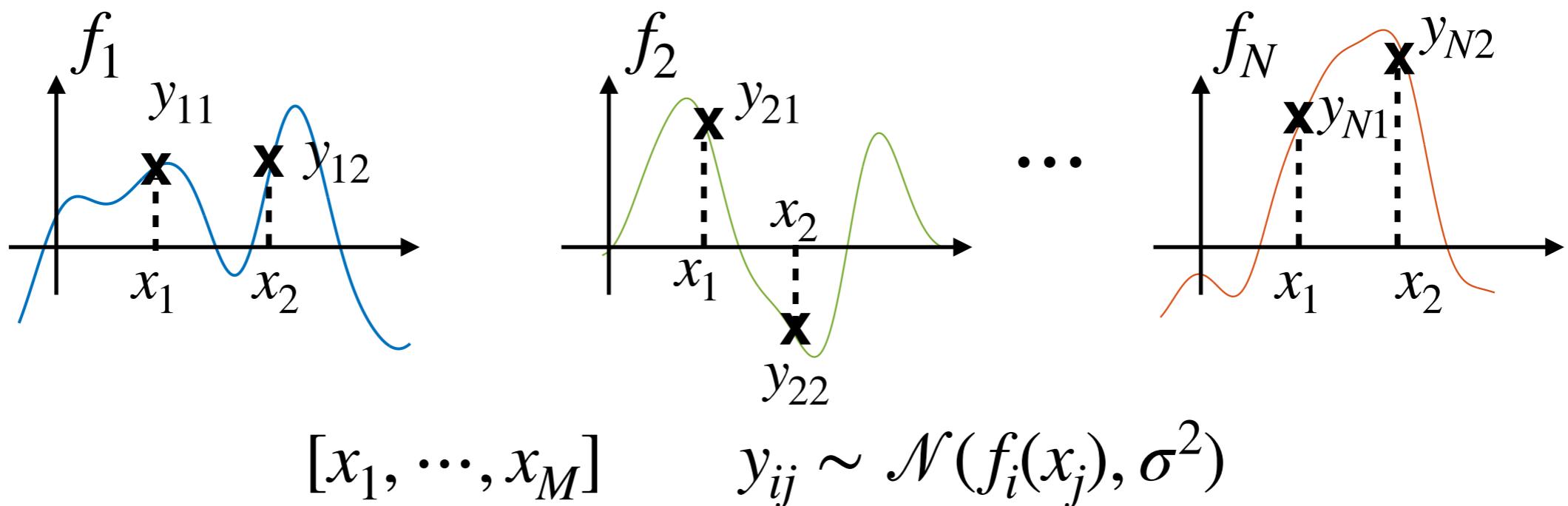
- Our idea: learn the “prior” from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \dots, f_N \sim GP(\mu, k)$



[Wang*&Kim*&Kaelbling, NIPS 2018]

Bayesian optimization with an unknown prior

- Our idea: learn the “prior” from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \dots, f_N \sim GP(\mu, k)$

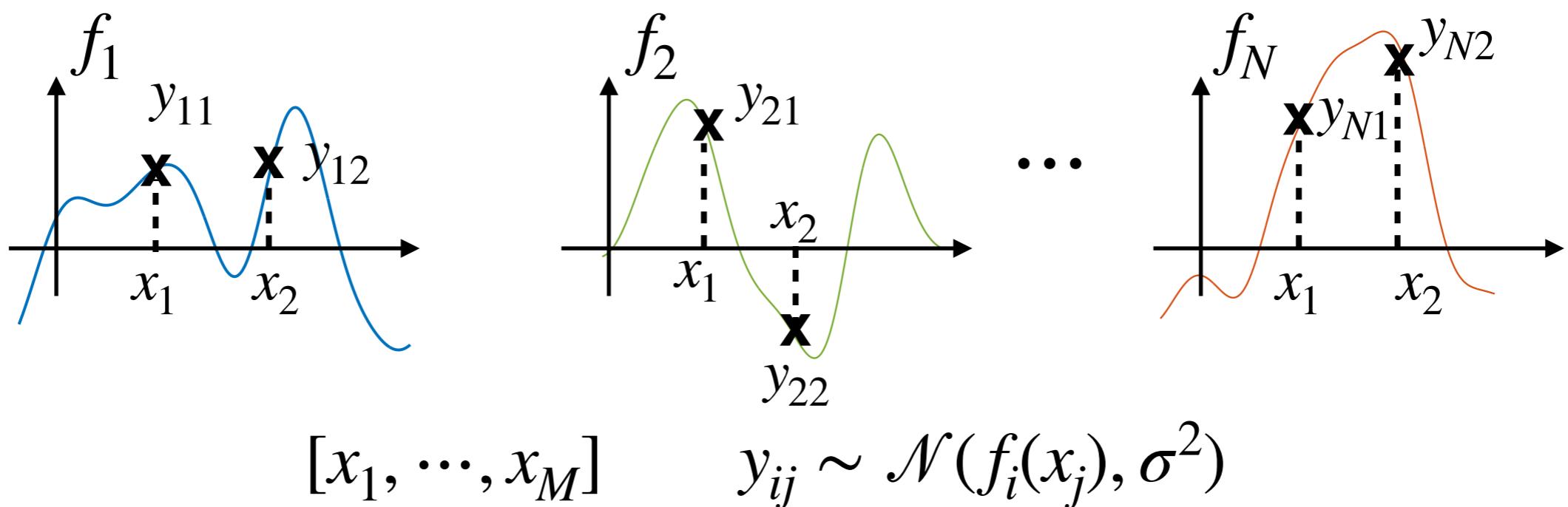


[Wang*&Kim*&Kaelbling, NIPS 2018]

Bayesian optimization with an unknown prior

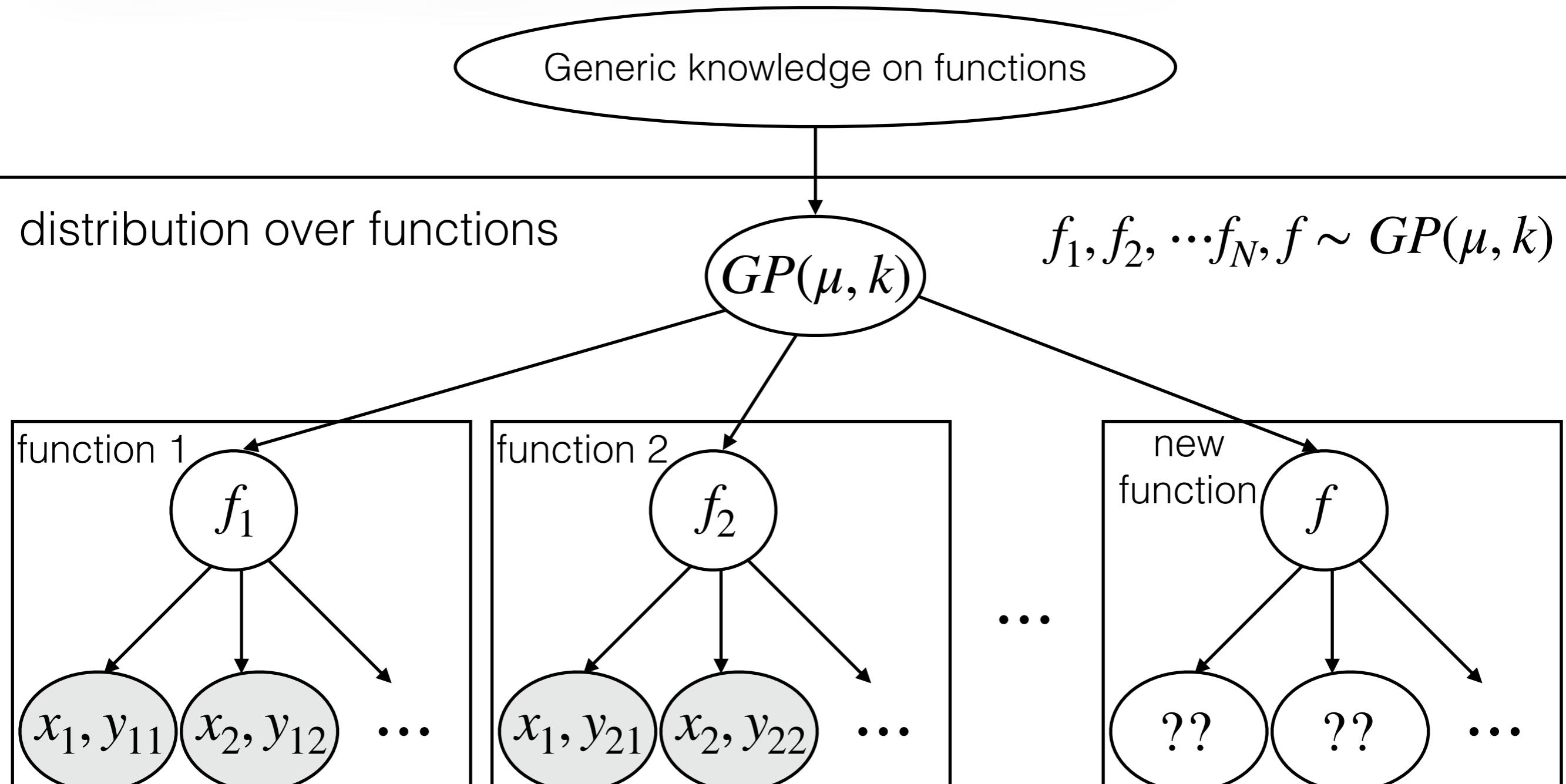
meta / multi-task / transfer learning

- Our idea: learn the “prior” from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \dots, f_N \sim GP(\mu, k)$



[Wang*&Kim*&Kaelbling, NIPS 2018]

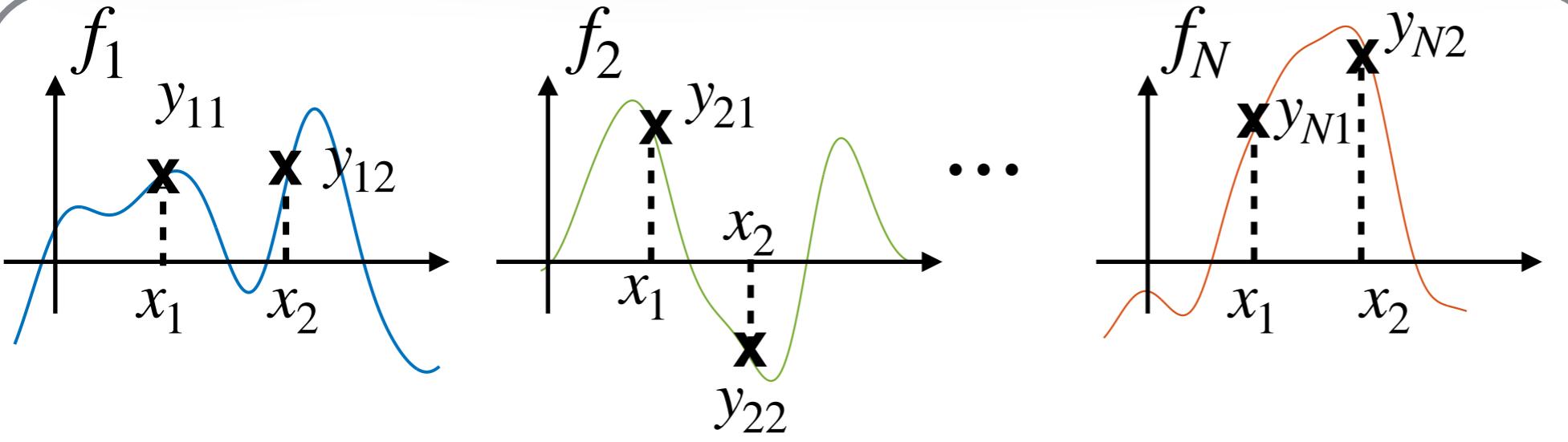
Bayesian optimization with an unknown prior



observations of
function values

[Wang*&Kim*&Kaelbling, NIPS 2018]

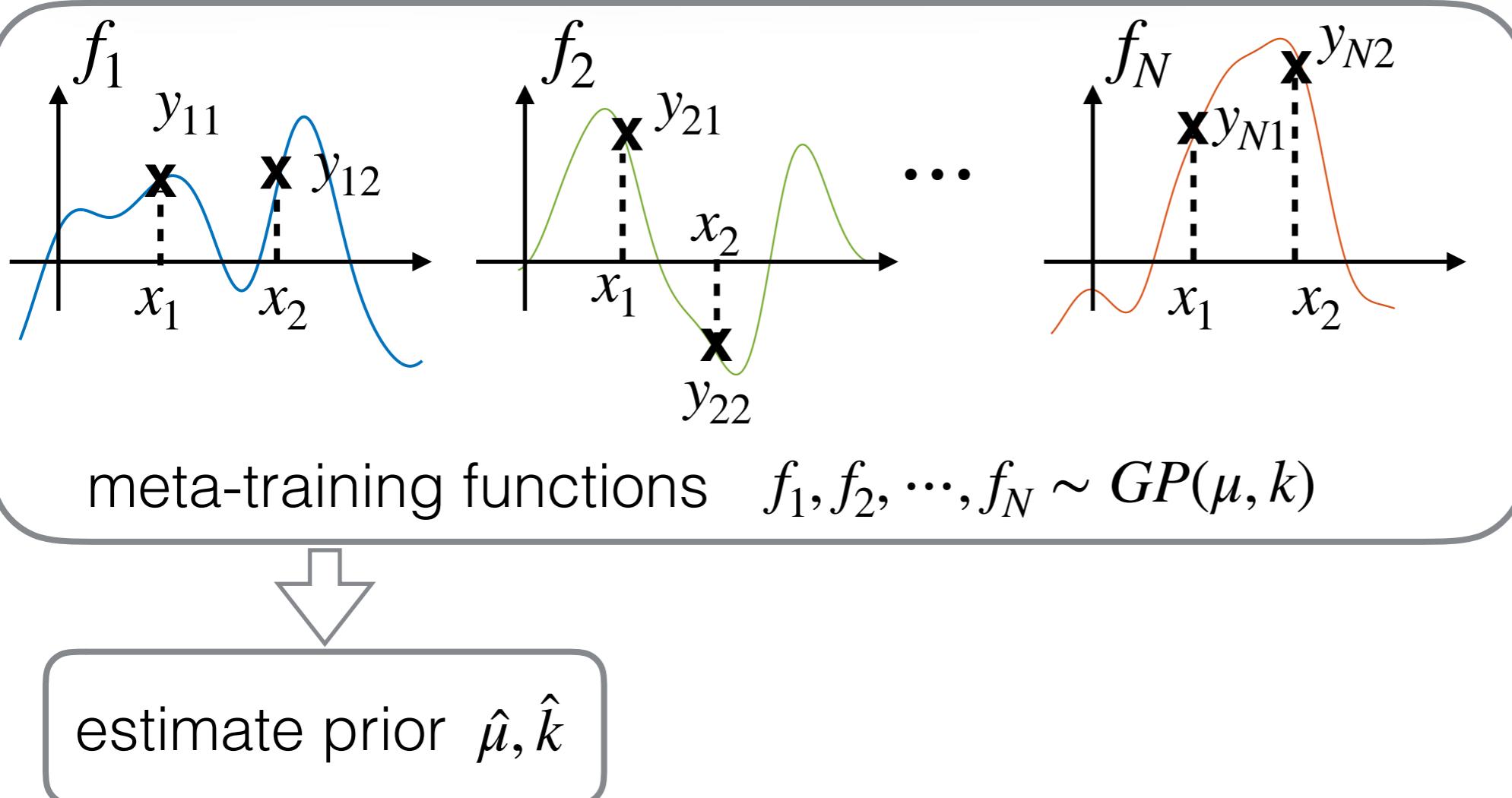
Meta Bayesian optimization with an unknown prior



meta-training functions $f_1, f_2, \dots, f_N \sim GP(\mu, k)$

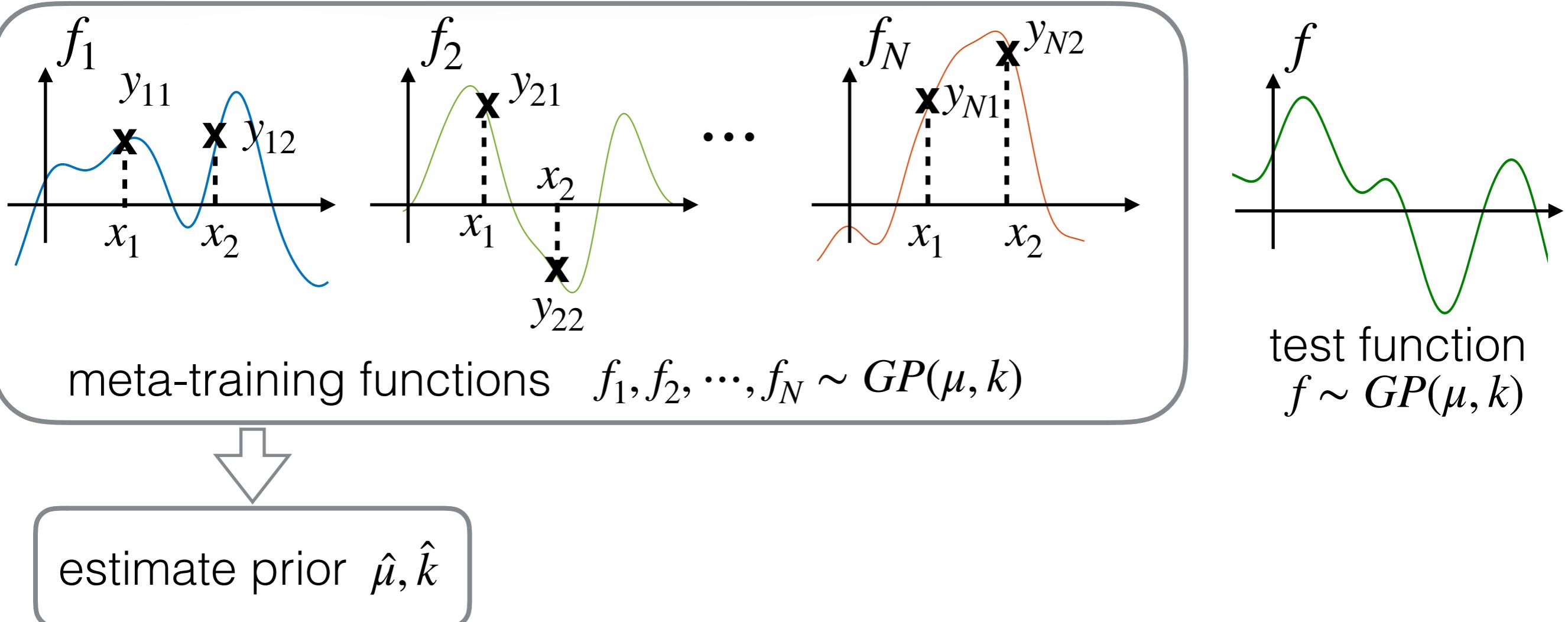
[Wang*&Kim*&Kaelbling, NIPS 2018]

Meta Bayesian optimization with an unknown prior



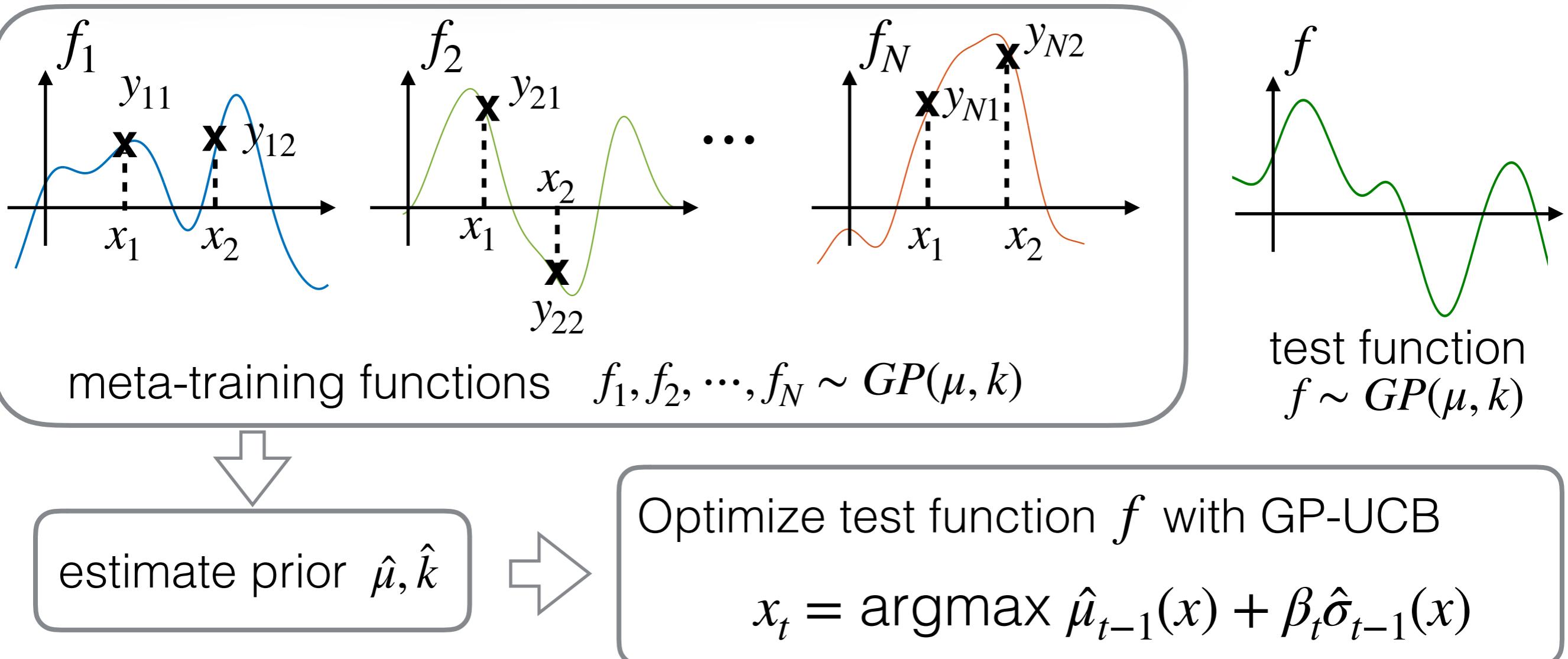
[Wang*&Kim*&Kaelbling, NIPS 2018]

Meta Bayesian optimization with an unknown prior



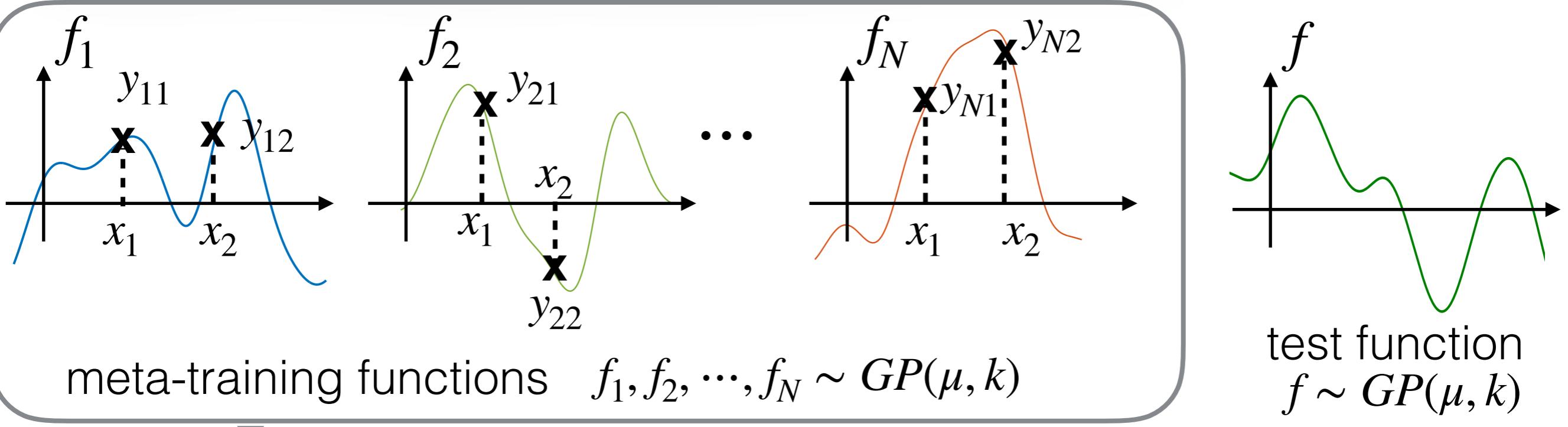
[Wang*&Kim*&Kaelbling, NIPS 2018]

Meta Bayesian optimization with an unknown prior



[Wang*&Kim*&Kaelbling, NIPS 2018]

Meta Bayesian optimization with an unknown prior



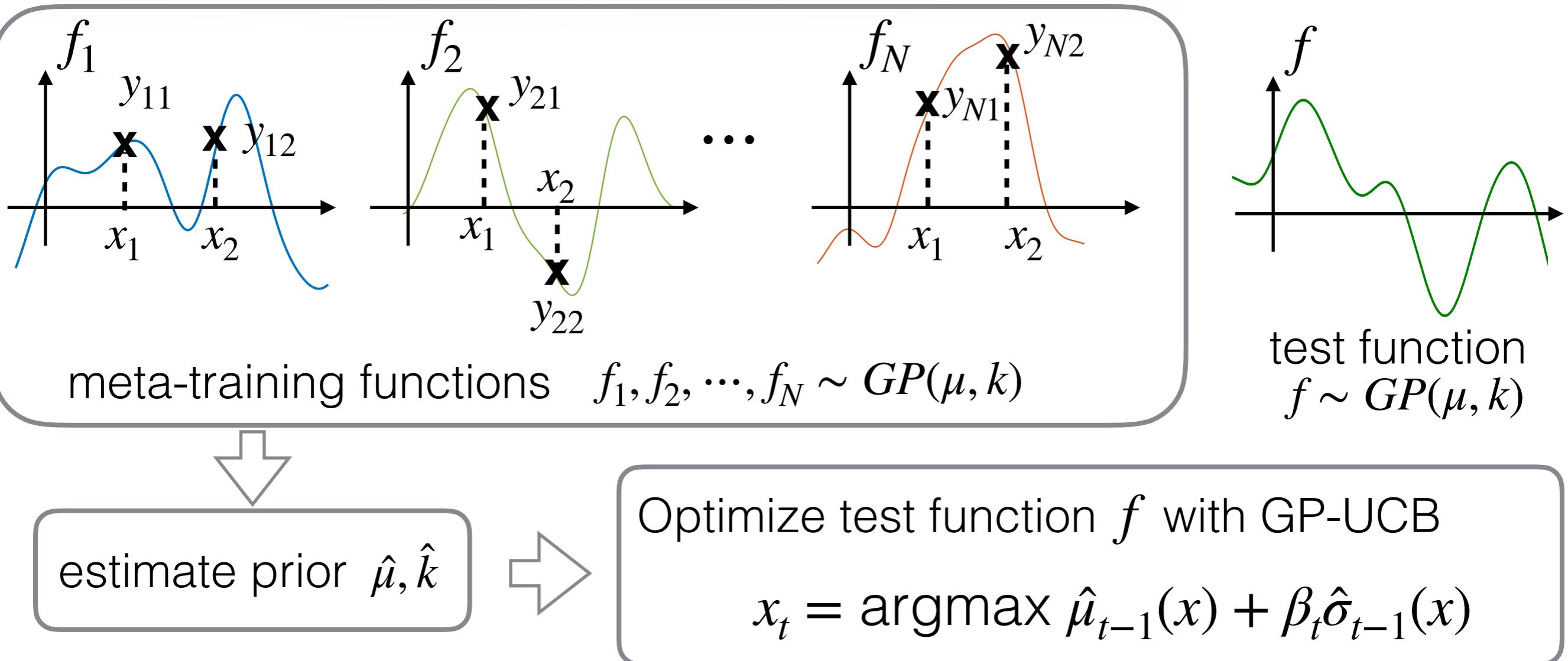
estimate prior $\hat{\mu}, \hat{k}$



$$\beta_t = \frac{\left(6(N-3+t+2\sqrt{t \log \frac{6}{\delta}} + 2 \log \frac{6}{\delta}) / (\delta N(N-t-1))\right)^{\frac{1}{2}} + (2 \log(\frac{3}{\delta}))^{\frac{1}{2}}}{(1 - 2(\frac{1}{N-t} \log \frac{6}{\delta})^{\frac{1}{2}})^{\frac{1}{2}}}$$

[Wang*&Kim*&Kaelbling, NIPS 2018]

Meta Bayesian optimization with an unknown prior



$$\beta_t = \frac{(6(N-3+t+2\sqrt{t})}{\beta_t(N, \delta)}$$

| N | δ | t | $\beta_t(N, \delta)$ |
|------|----------|-----|----------------------|
| 50 | 0.01 | 10 | 19.0 |
| 100 | 0.01 | 10 | 9.7 |
| 1000 | 0.01 | 10 | 4.5 |

[Wang*&Kim*&Kaelbling, NIPS 2018]

Regret bound for meta BO with unknown prior

Theorem (informal)

$$\text{simple regret: } r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$$

Important assumptions:

- meta-training functions come from the same prior
- enough number of meta-training functions $N \geq O(\max(T, M))$
- kernel function is bounded

[Wang*&Kim*&Kaelbling, NIPS 2018]

Regret bound for meta BO with unknown prior

Theorem (informal)

simple regret: $r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$

Important assumptions:

- meta-training functions come from the same prior
- enough number of meta-training functions $N \geq O(\max(T, M))$
- kernel function is bounded

#observations on each
training function

Regret bound for meta BO with unknown prior

Theorem (informal)

simple regret: $r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$

Important assumptions:

- meta-training functions come from the same prior
- enough number of meta-training functions $N \geq O(\max(T, M))$
- kernel function is bounded

#observations on each
training function

T observations on the test function f

[Wang*&Kim*&Kaelbling, NIPS 2018]

Regret bound for meta BO with unknown prior

Theorem (informal)

$$\text{simple regret: } r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$$

Important assumptions:

- meta-training functions come from the same prior
- enough number of meta-training functions $N \geq O(\max(T, M))$
- kernel function is bounded

#observations on each
training function

Given T observations on the test function f , with probability $1 - \delta$,

$$\text{simple regret } r_T \leq O\left(\left(\sqrt{\frac{1}{N-T}} + C\right)\left(\sqrt{\frac{\log T}{T}} + \sigma^2\right)\right)$$

[Wang*&Kim*&Kaelbling, NIPS 2018]

Regret bound for meta BO with unknown prior

Theorem (informal)

$$\text{simple regret: } r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$$

Important assumptions:

- meta-training functions come from the same prior
- enough number of meta-training functions $N \geq O(\max(T, M))$
- kernel function is bounded

#observations on each
training function

Given T observations on the test function f , with probability $1 - \delta$,

$$\text{simple regret } r_T \leq O\left(\left(\sqrt{\frac{1}{N-T}} + C\right)\left(\sqrt{\frac{\log T}{T}} + \sigma^2\right)\right)$$

constant
depending on δ

[Wang*&Kim*&Kaelbling, NIPS 2018]

Regret bound for meta BO with unknown prior

Theorem (informal)

$$\text{simple regret: } r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$$

Important assumptions:

- meta-training functions come from the same prior
- enough number of meta-training functions $N \geq O(\max(T, M))$
- kernel function is bounded

#observations on each
training function

Given T observations on the test function f , with probability $1 - \delta$,

$$\text{simple regret } r_T \leq O\left(\left(\sqrt{\frac{1}{N-T}} + C\right)\left(\sqrt{\frac{\log T}{T}} + \sigma^2\right)\right)$$

constant
depending on δ

observation noise

[Wang*&Kim*&Kaelbling, NIPS 2018]

Regret bound for meta BO with unknown prior

Theorem (informal)

$$\text{simple regret: } r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$$

Important assumptions:

- meta-training functions come from the same prior
- enough number of meta-training functions $N \geq O(\max(T, M))$
- kernel function is bounded

#observations on each
training function

Given T observations on the test function f , with probability $1 - \delta$,

$$\text{simple regret } r_T \leq O\left(\left(\sqrt{\frac{1}{N-T}} + C\right)\left(\sqrt{\frac{\log T}{T}} + \sigma^2\right)\right)$$

constant
depending on δ linear kernel observation noise

[Wang*&Kim*&Kaelbling, NIPS 2018]

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \mathcal{N}(\mu(\mathcal{X}), \frac{1}{N}(k(\mathcal{X}) + \sigma^2 I))$$

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1} (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T)^T (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T) \sim \mathcal{W}\left(\frac{1}{N-1}(k(\mathcal{X}) + \sigma^2 I), N-1\right)$$

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1} (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T)^T (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T) \sim \text{Wishart distribution}$$

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1} (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T)^T (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T) \sim \text{Wishart distribution}$$

unbiased posterior estimator

$$\hat{\mu}_t(x) = \hat{\mu}(x) + \hat{k}(x, \mathbf{x}_t) \hat{k}(\mathbf{x}_t, \mathbf{x}_t)^{-1} (\mathbf{y}_t - \hat{\mu}(\mathbf{x}_t))$$

$$\hat{\sigma}_t^2(x, x') = \frac{N-1}{N-t-1} \left(\hat{k}(x, x') - \hat{k}(x, \mathbf{x}_t) \hat{k}(\mathbf{x}_t, \mathbf{x}_t)^{-1} \hat{k}(\mathbf{x}_t, x') \right)$$

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1} (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T)^T (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T) \sim \text{Wishart distribution}$$

unbiased posterior estimator

$$\hat{\mu}_t(x) = \text{can bound } |\hat{\mu}_t(x) - \mu_t(x)|$$

$$\hat{\sigma}_t^2(x, x') = \frac{N-1}{N-t-1} \left(\hat{k}(x, x') - \hat{k}(x, \mathbf{x}_t) \hat{k}(\mathbf{x}_t, \mathbf{x}_t)^{-1} \hat{k}(\mathbf{x}_t, x') \right)$$

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1} (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T)^T (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T) \sim \text{Wishart distribution}$$

unbiased posterior estimator

$$\hat{\mu}_t(x) = \text{can bound } |\hat{\mu}_t(x) - \mu_t(x)|$$

$$\hat{\sigma}_t^2(x, x') = \frac{N-1}{N-t-1} \left(\hat{k}(x, x') - \hat{k}(x, \mathbf{x}_t) \hat{k}(\mathbf{x}_t, \mathbf{x}_t)^{-1} \hat{k}(\mathbf{x}_t, x') \right)$$

bias correction

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1} (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T)^T (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T) \sim \text{Wishart distribution}$$

unbiased posterior estimator

$$\hat{\mu}_t(x) = \text{can bound } |\hat{\mu}_t(x) - \mu_t(x)|$$

$$\hat{\sigma}_t^2(x, x') = \frac{N-1}{N-t-1} \left(\hat{k}(x, x') - \hat{k}(x, \mathbf{x}_t) \hat{k}(\mathbf{x}_t, \mathbf{x}_t)^{-1} \hat{k}(\mathbf{x}_t, x') \right)$$

\sim Wishart distribution

Discrete input space $|\mathcal{X}| = M$

Prior estimation with meta training data $\{(x_j, y_{ij})\}_{j=1}^M\}_{i=1}^N$

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Missing entries? Matrix completion.

unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T \mathbf{1}_N \sim \text{Gaussian distribution}$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1} (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T)^T (Y - \mathbf{1}_N \hat{\mu}(\mathcal{X})^T) \sim \text{Wishart distribution}$$

unbiased posterior estimator

$$\hat{\mu}_t(x) = \text{can bound } |\hat{\mu}_t(x) - \mu_t(x)|$$

$$\hat{\sigma}_t^2(x, x') = \text{can bound } \frac{\hat{\sigma}_t^2(x)}{\sigma_t^2(x) + \sigma^2}$$

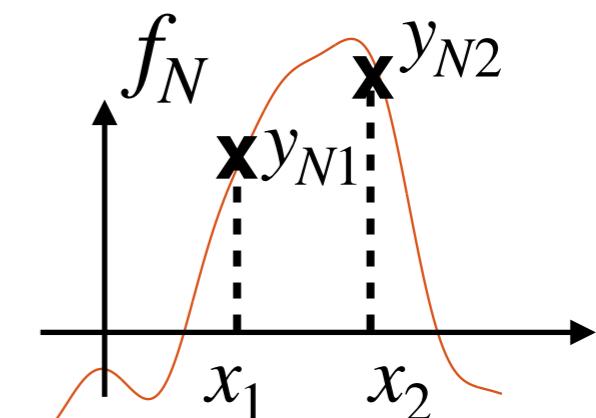
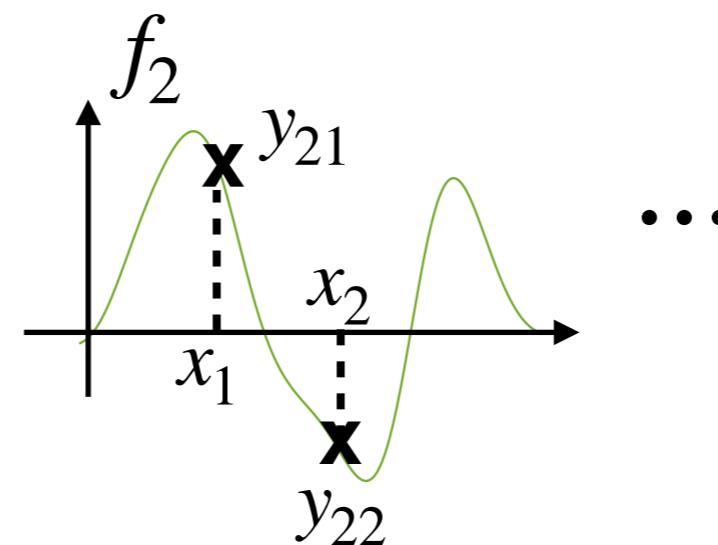
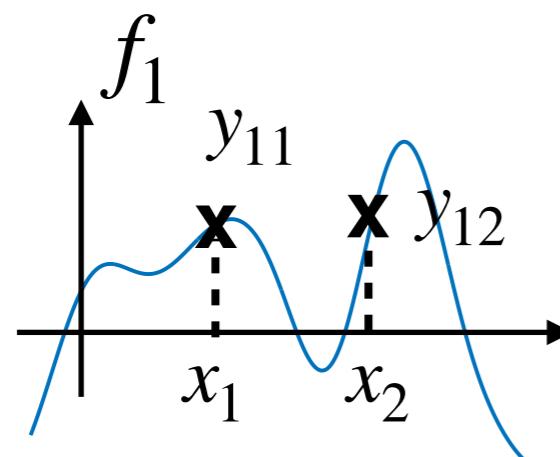
Continuous input space

assume basis functions are given!

basis functions
 $f = \Phi(x)^T W \sim GP(\mu, k)$

mean parameter
 $W \sim \mathcal{N}(\mathbf{u}, \Sigma)$

$$\mu(x) = \Phi(x)^T \mathbf{u} \quad k(x, x') = \Phi(x)^T \Sigma \Phi(x')$$

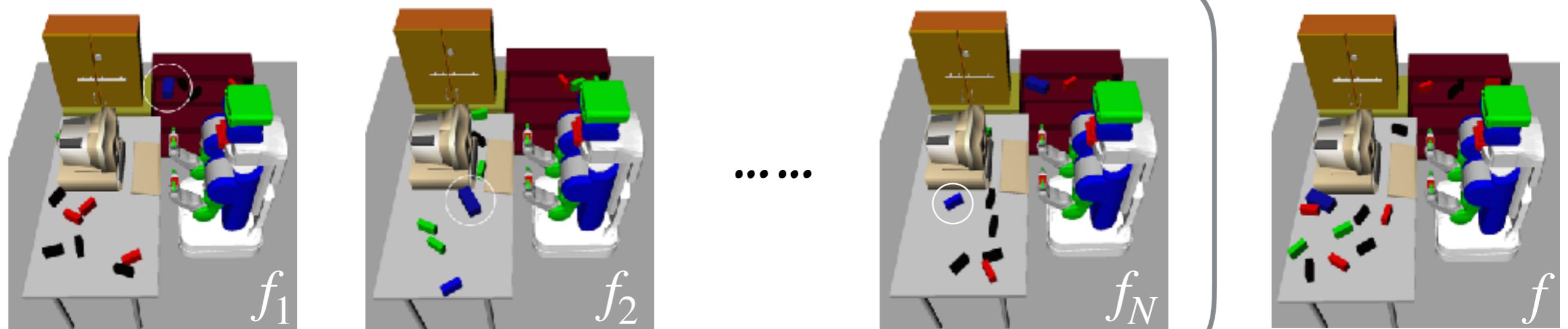


$$\Phi(X)^T W_i = Y_i \implies W_i = (\Phi(X)\Phi(X)^T)^{-1}\Phi(X)Y_i$$

estimate its mean and covariance to construct GP prior

[Neal, 1996; Wang*&Kim*&Kaelbling, NIPS 2018]

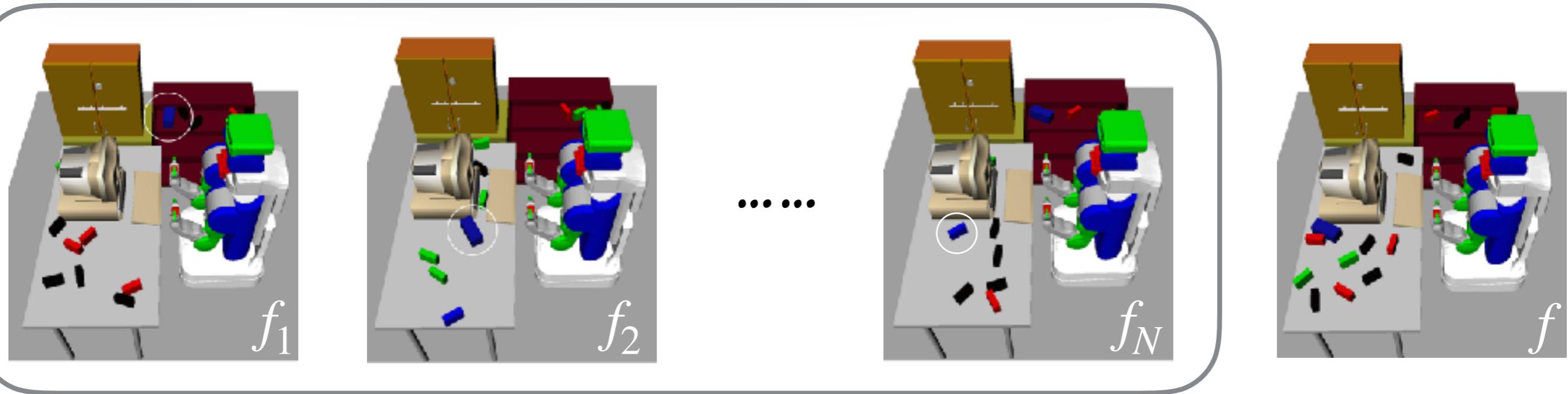
Empirical results on block picking and placing



meta-training data $N = 1500, M = 1000$

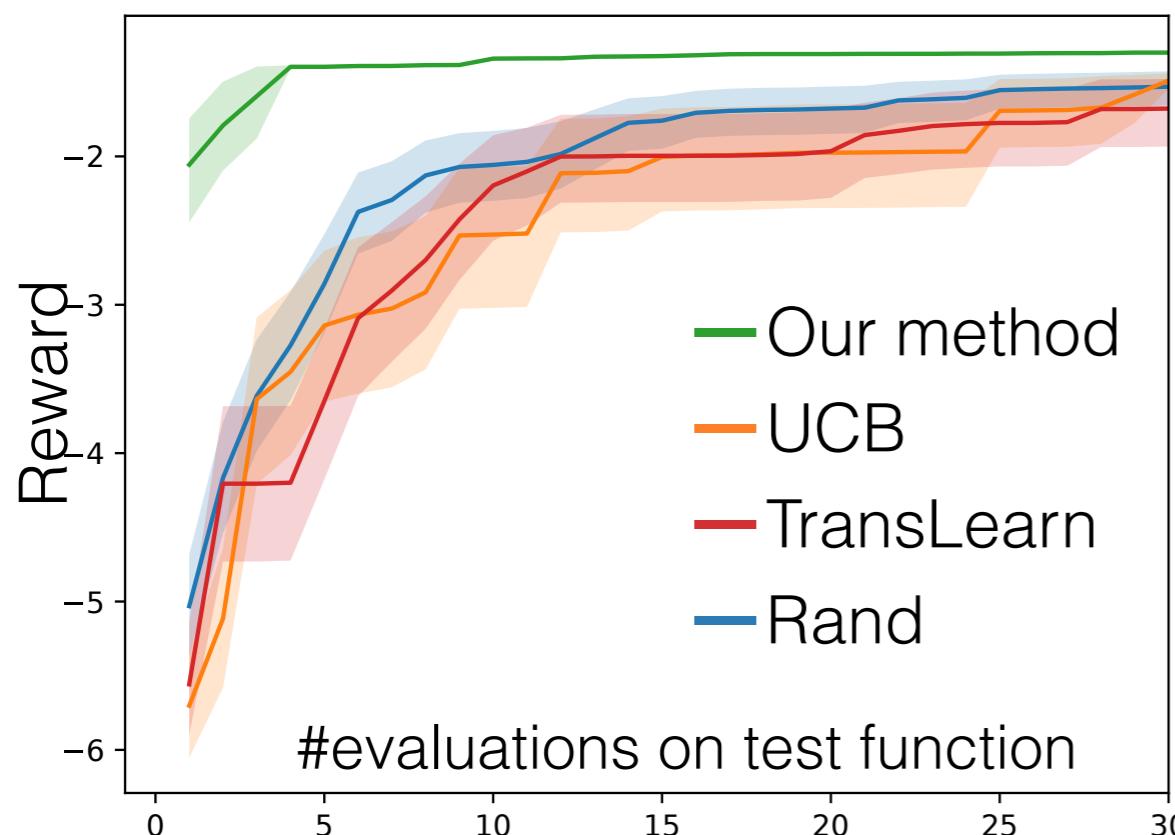
test function

Empirical results on block picking and placing

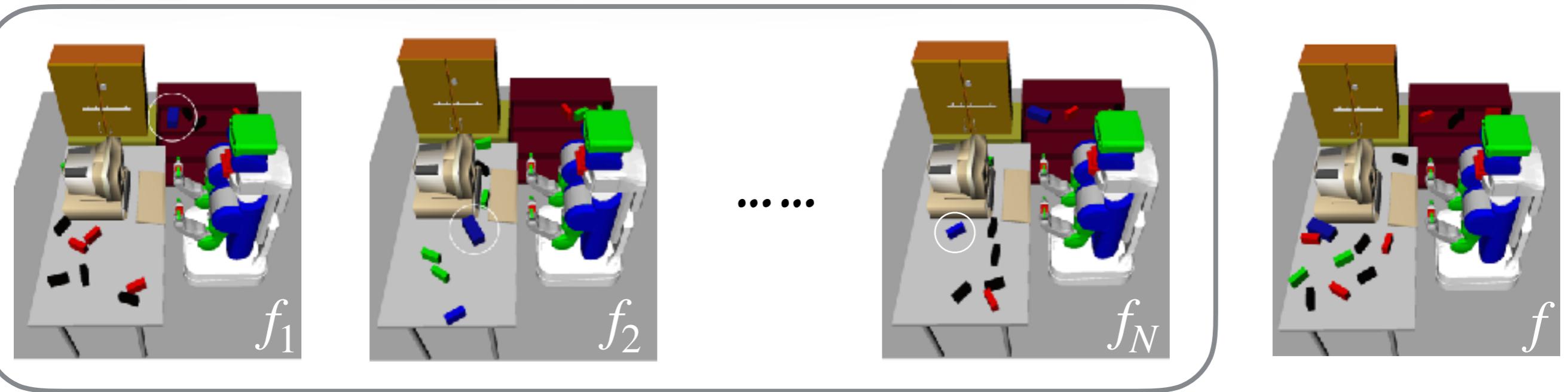


meta-training data $N = 1500, M = 1000$

test function

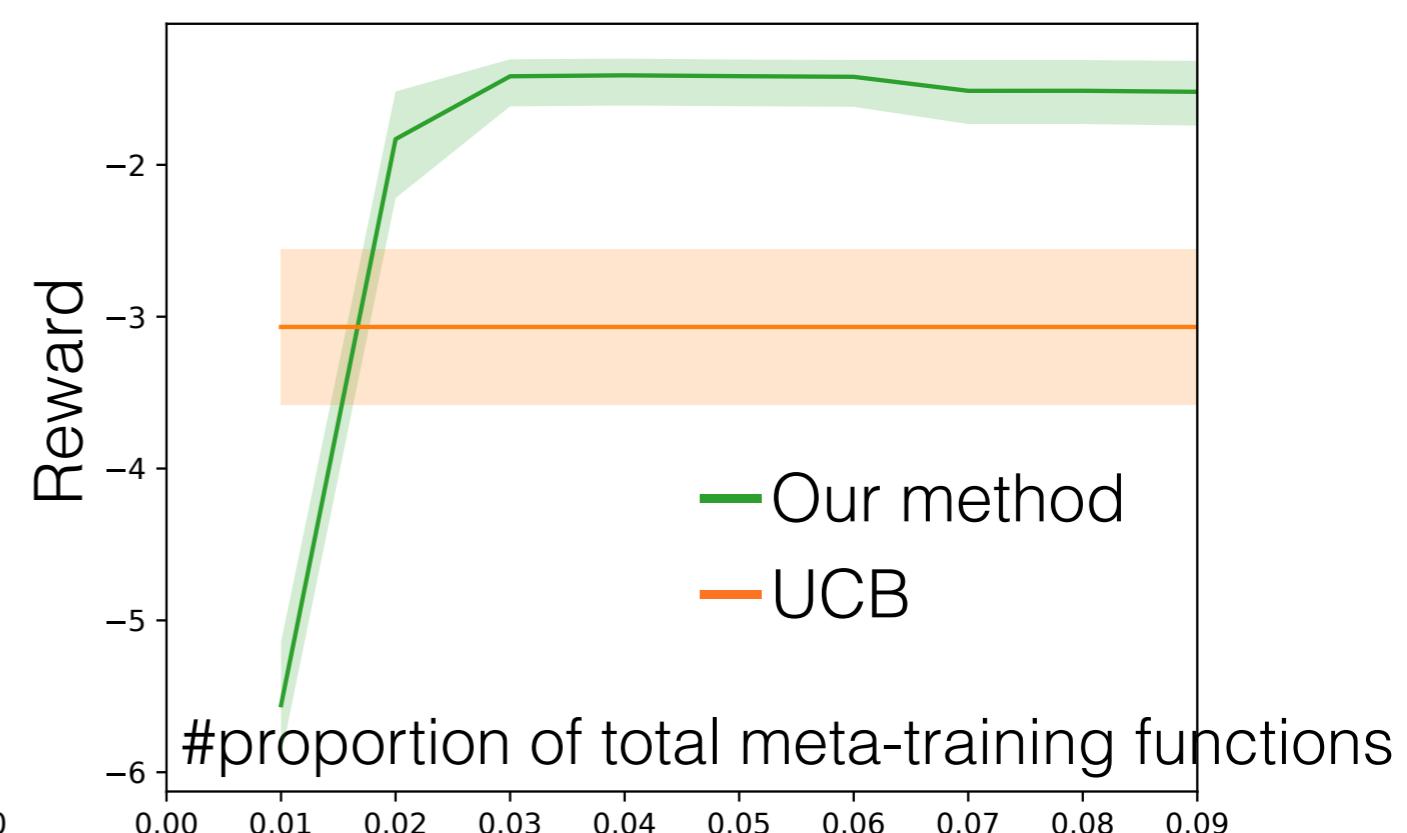
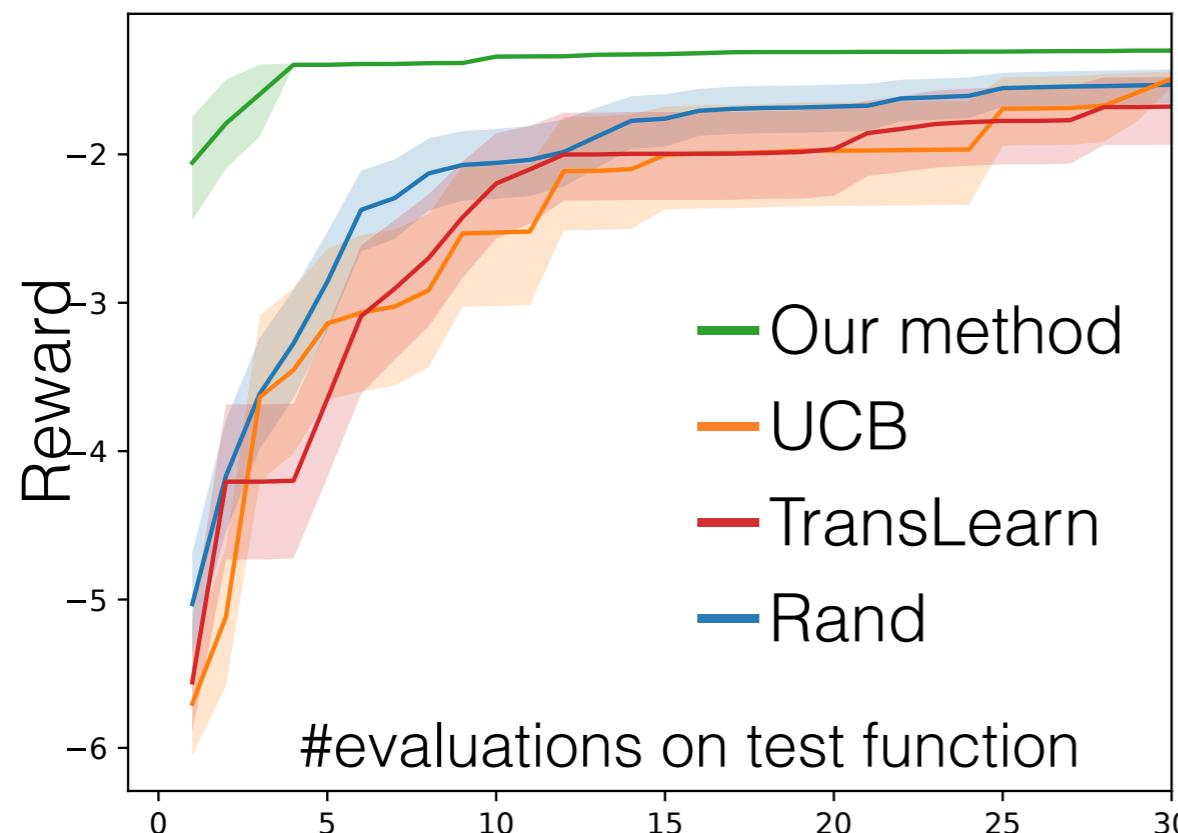


Empirical results on block picking and placing



meta-training data $N = 1500, M = 1000$

test function



Summary

- a regret bound for meta Bayesian optimization with an unknown prior **but with assumptions on available data**
- future directions
 - what if basis functions are unknown?
 - goodness-of-fit test for functions?
 - how to reduce the dependency on large N?
$$N = O(\max(T, M))$$
 - are there better estimators than unbiased ones?