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

Zi Wang

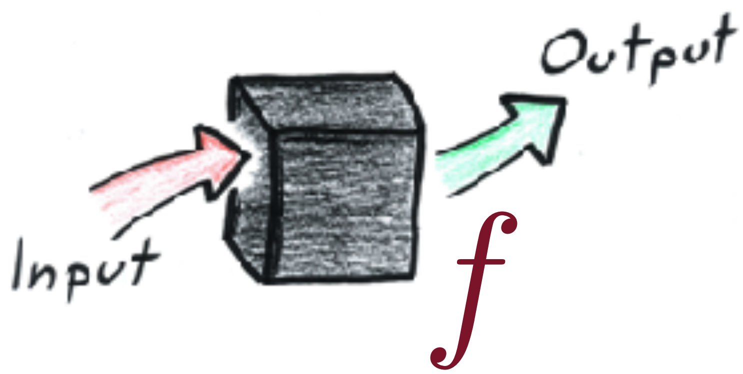
Microsoft Research AI Breakthroughs Workshop, Sep 18

Joint work with Beomjoon Kim, Leslie Pack Kaelbling

Blackbox Function Optimization

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

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



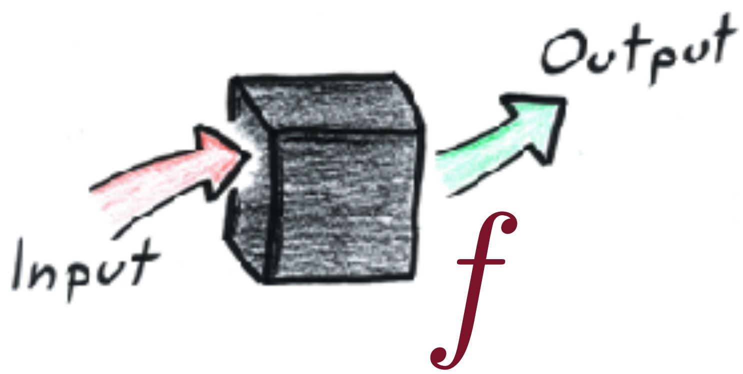
Challenges:

- 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

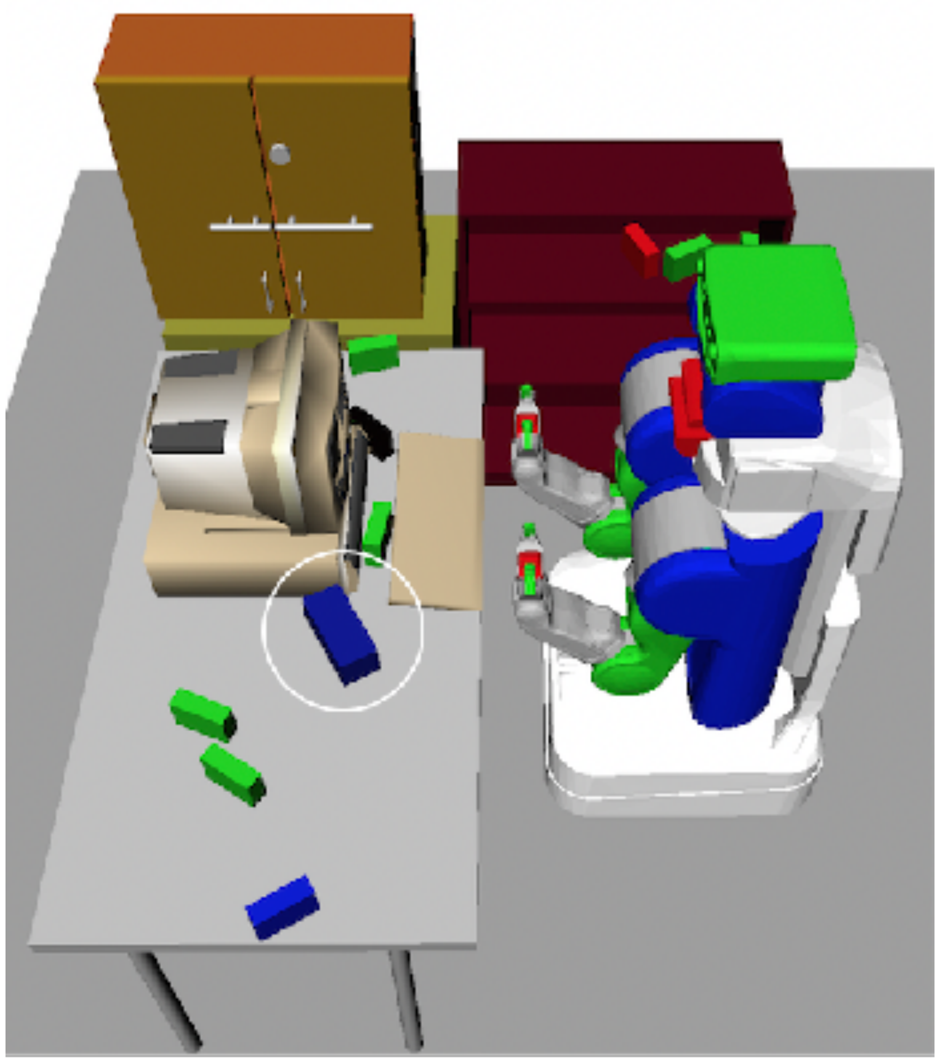
a.k.a derivative-free optimization,
experimental design,
(continuous) multi-armed bandit



Goal: $x^* = \operatorname{argmax} f(x)$

box $\mathscr{X} \subset \mathbb{R}^d$

or $|\mathscr{X}| = M$



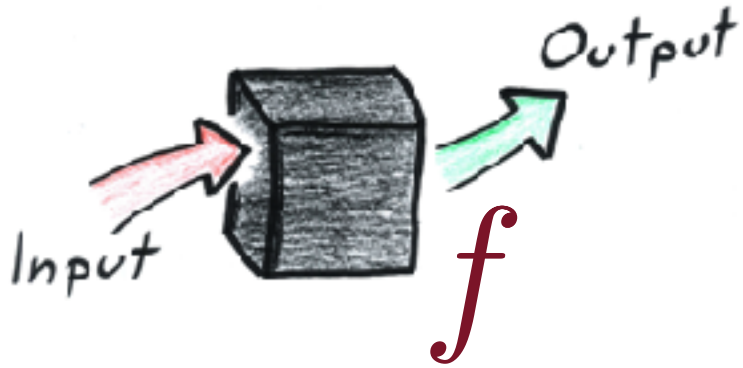
Challenges:

- 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)

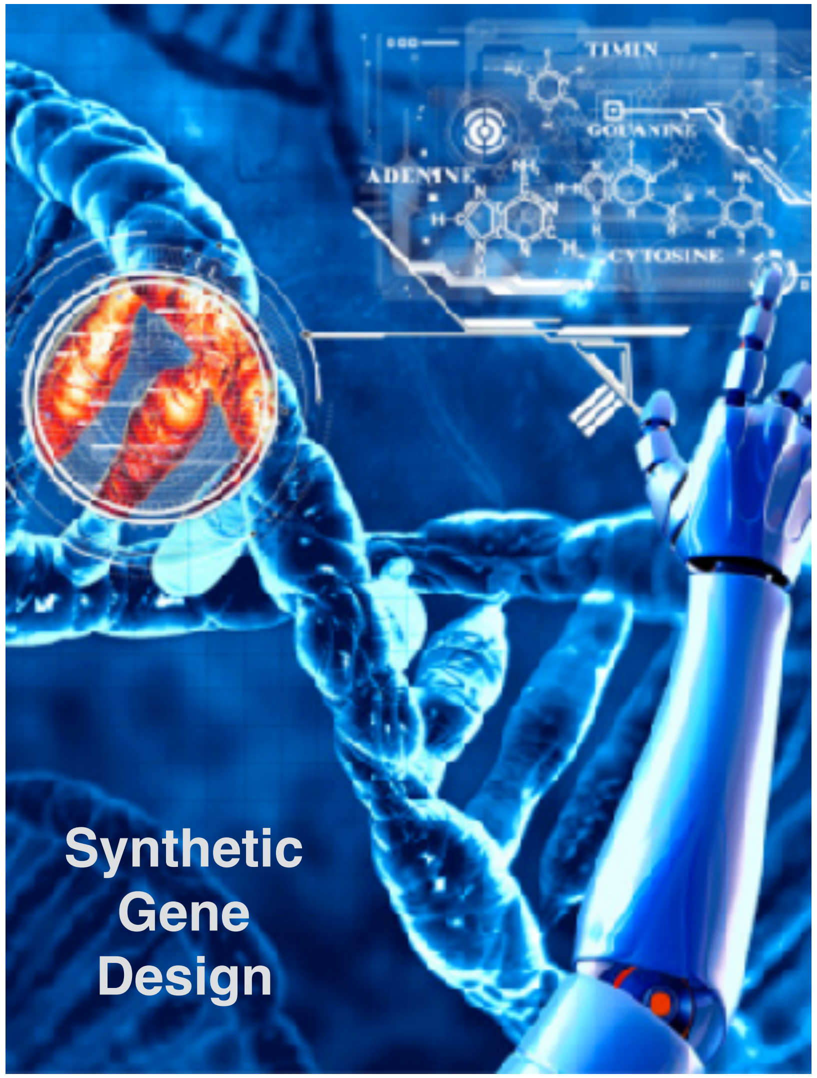
Blackbox Function Optimization

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



Goal: $x^* = \text{argmax}\, f(x)$

box $\mathcal{X} \subset \mathbb{R}^d$
or $|\mathcal{X}| = M$



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)

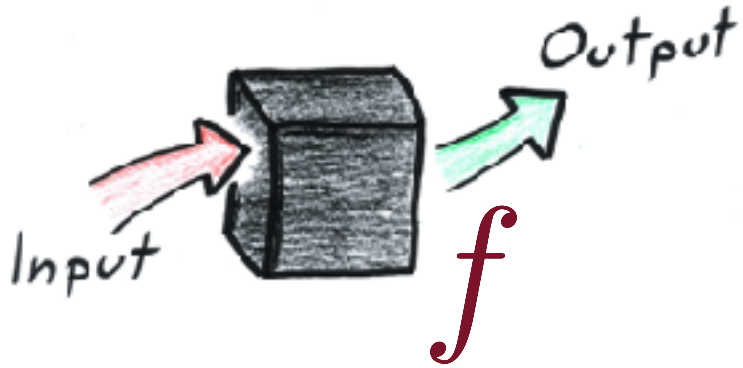
Blackbox Function Optimization

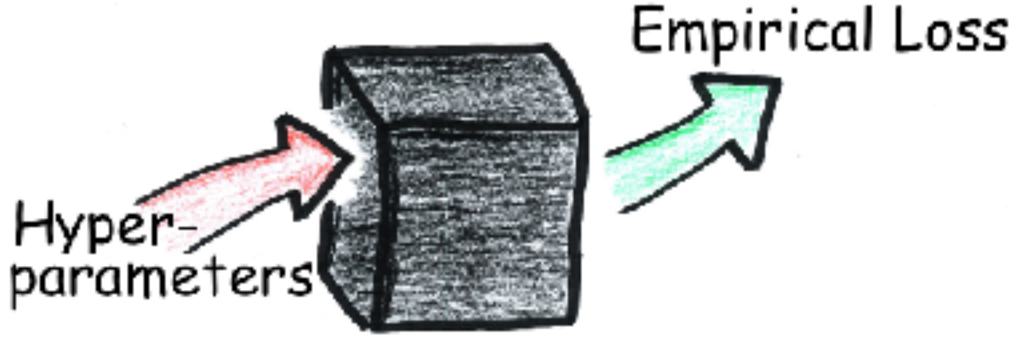
a.k.a derivative-free optimization,
experimental design,
(continuous) multi-armed bandit

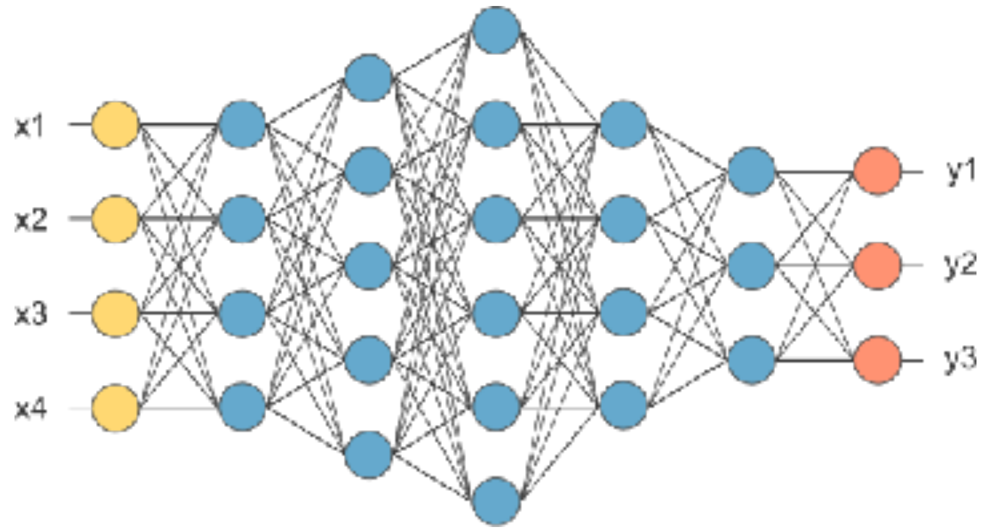
Goal: $x^* = \text{argmax}\, f(x)$

box $\mathcal{X} \subset \mathbb{R}^d$

or $|\mathcal{X}| = M$







Challenges:

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

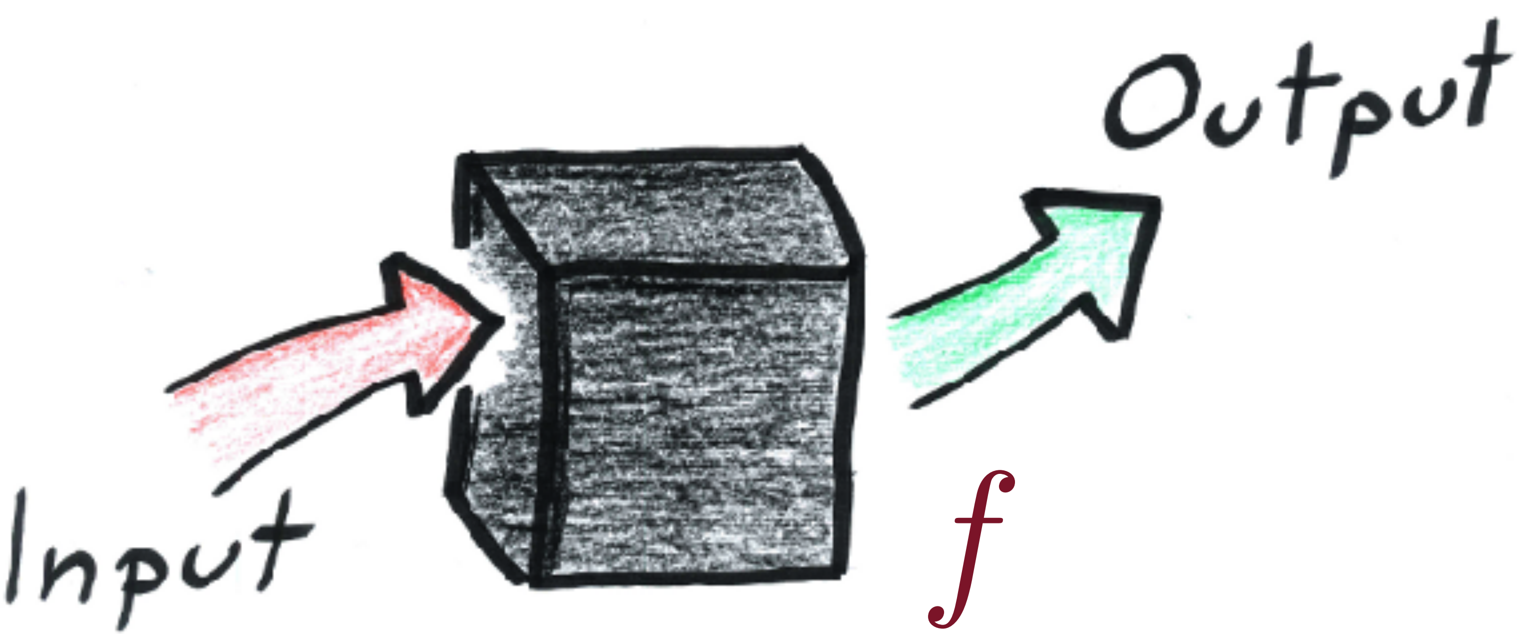
(Kim et al., 2017; Snoek et al., 2012; Gonzalez et al., 2015)

Bayesian Optimization

Idea: build a probabilistic model of the function f

LOOP

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

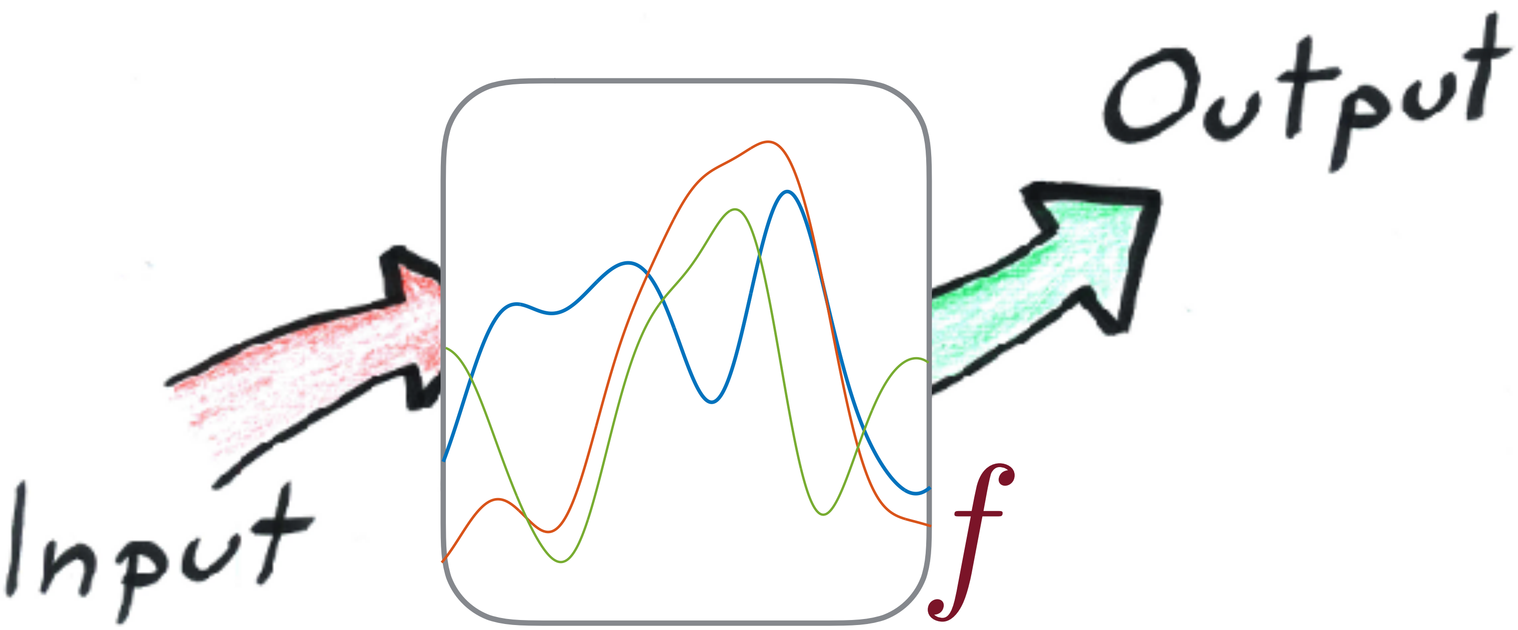


Bayesian Optimization

Idea: build a probabilistic model of the function f

LOOP

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

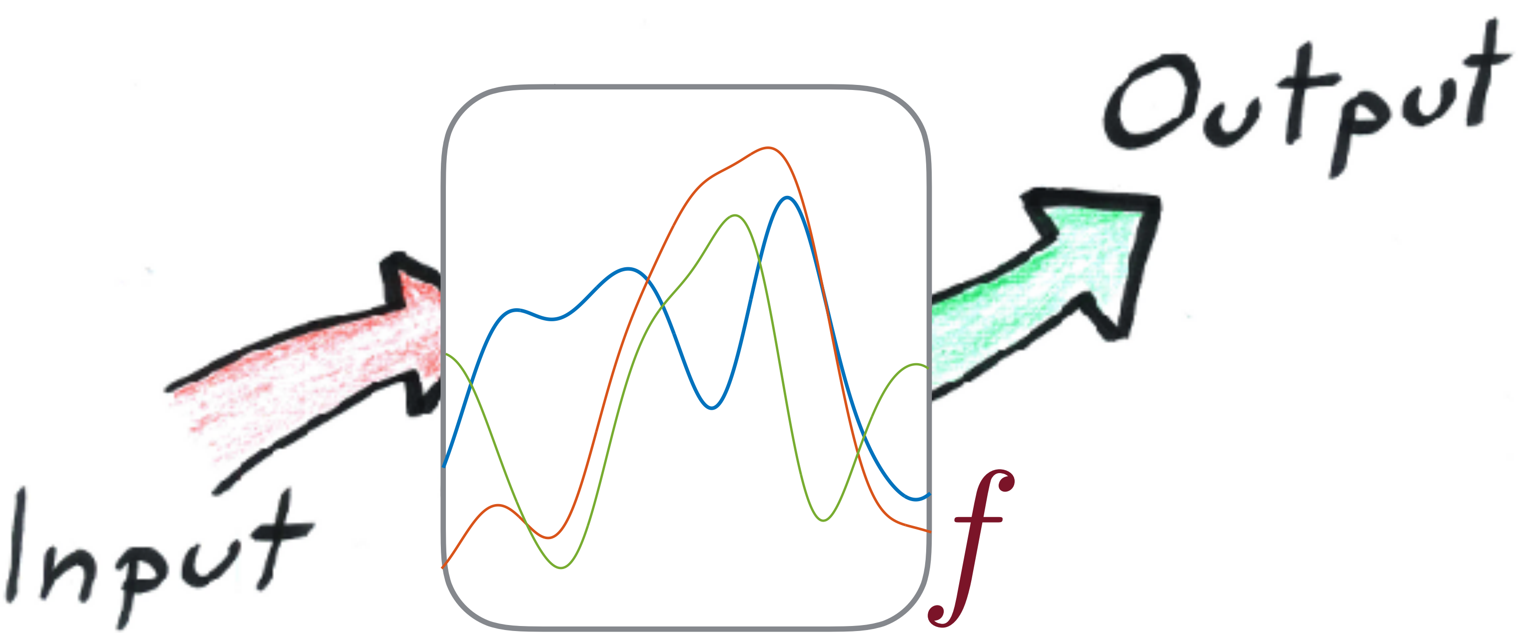


Bayesian Optimization

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



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

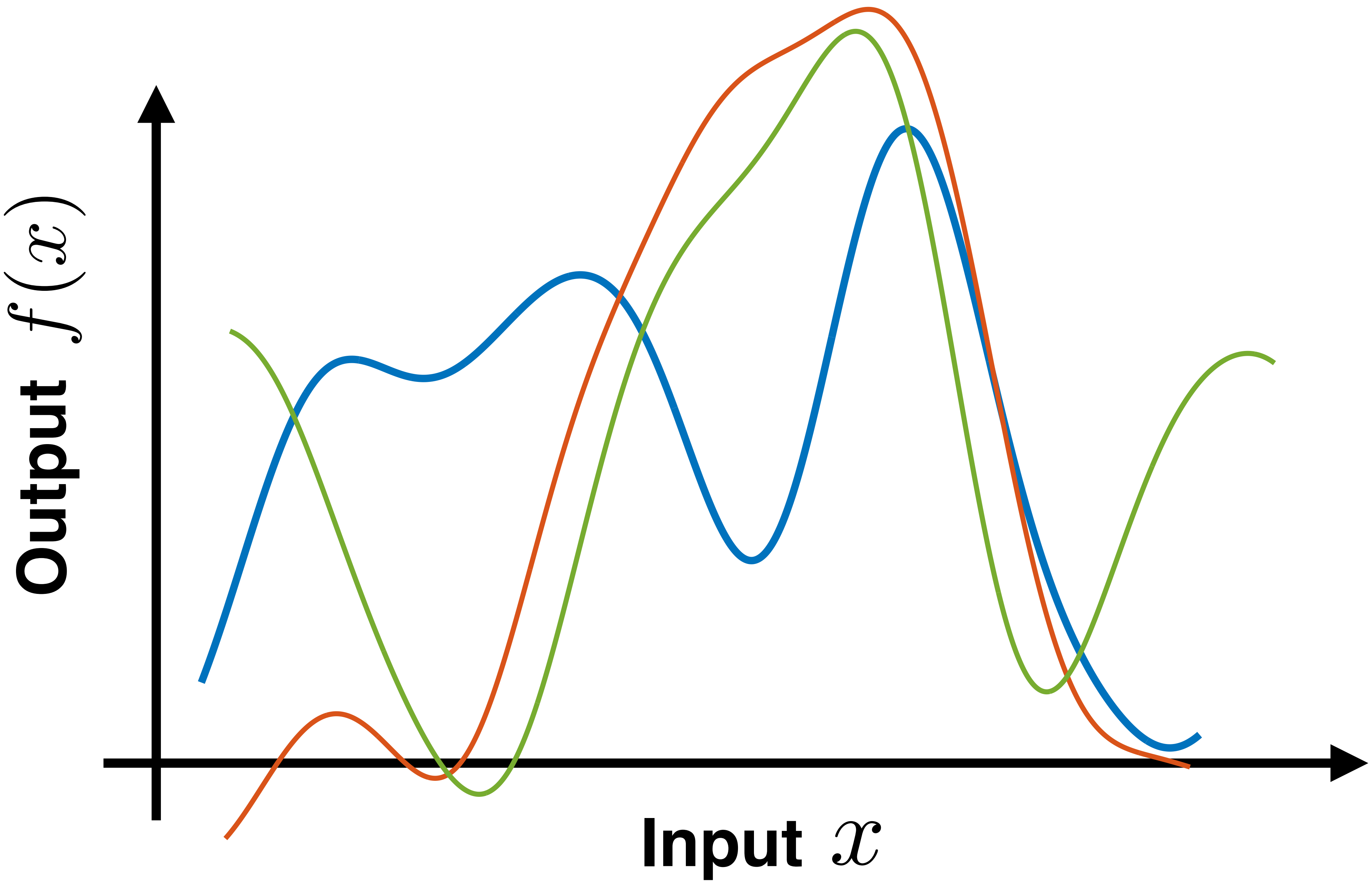
$$\Downarrow$$

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

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

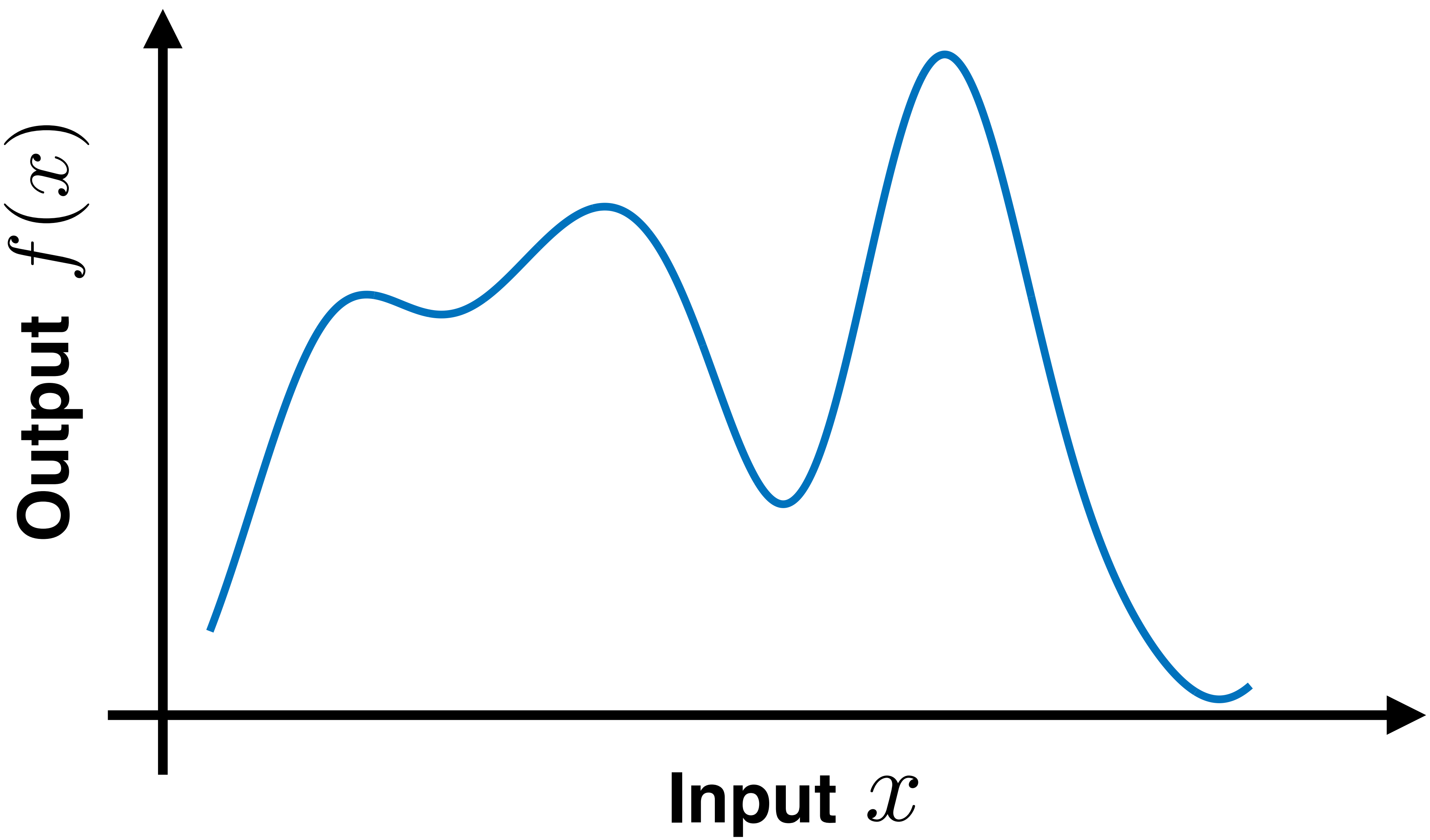
Gaussian Processes (GPs)

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



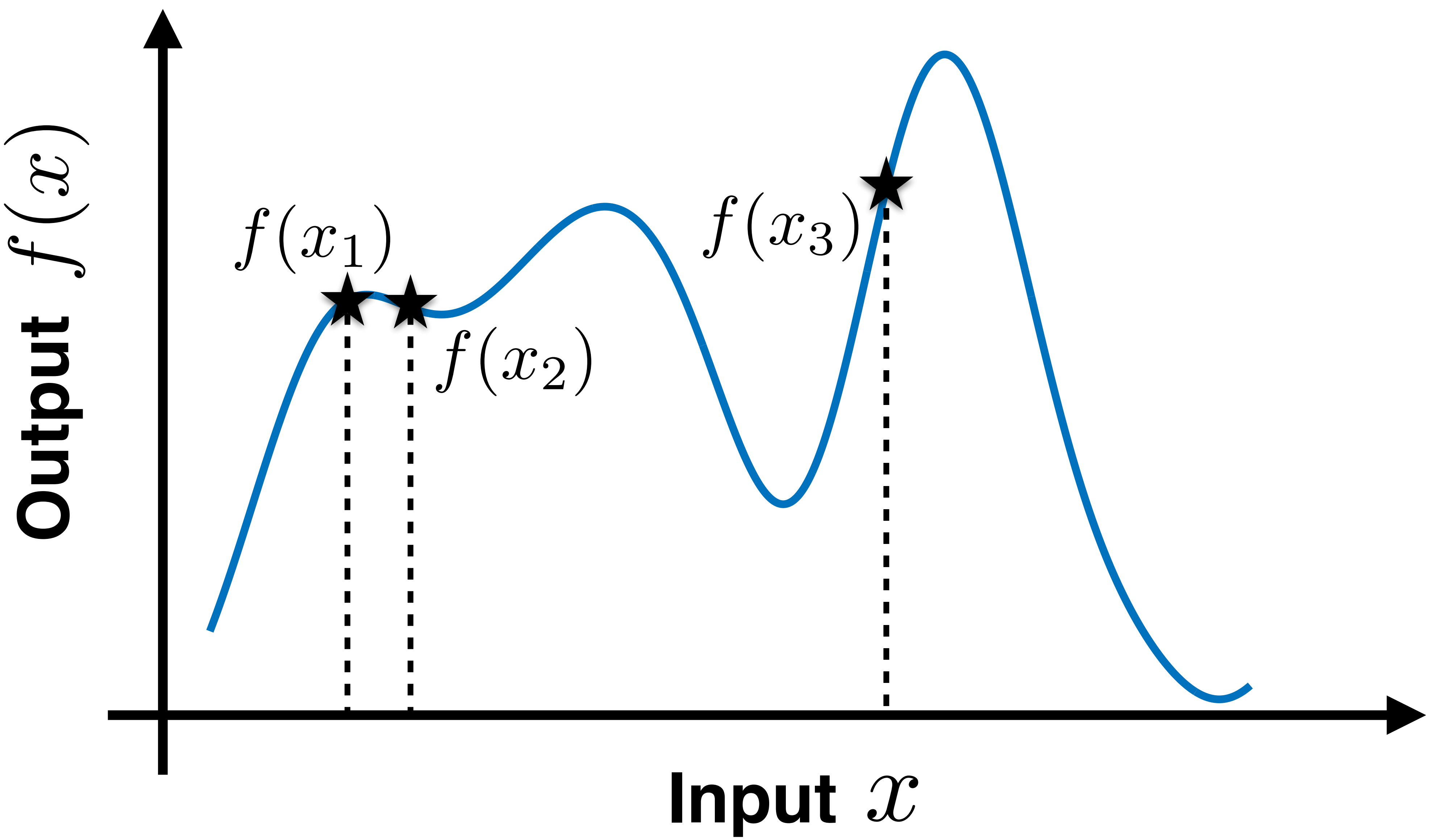
Gaussian Processes (GPs)

- 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 $\mu(\cdot)$

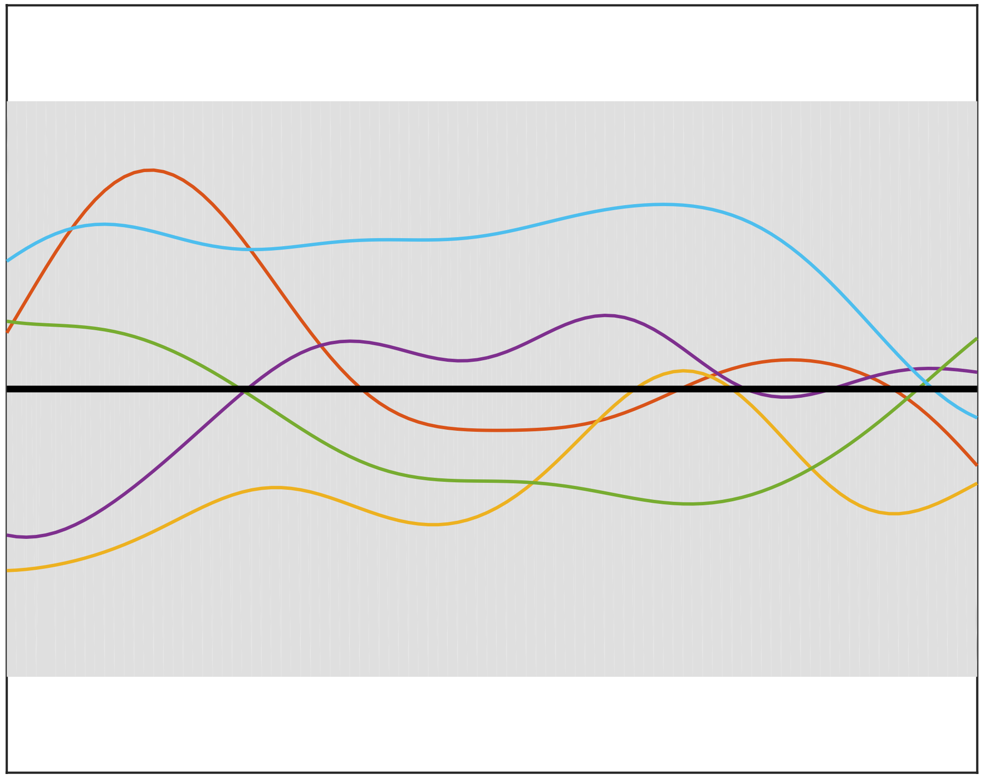
$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1), & \cdots, & k(x_1, x_n) \\ \vdots, & & \vdots \\ k(x_n, x_1), & \cdots, & k(x_n, x_n) \end{bmatrix} \right)$$

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

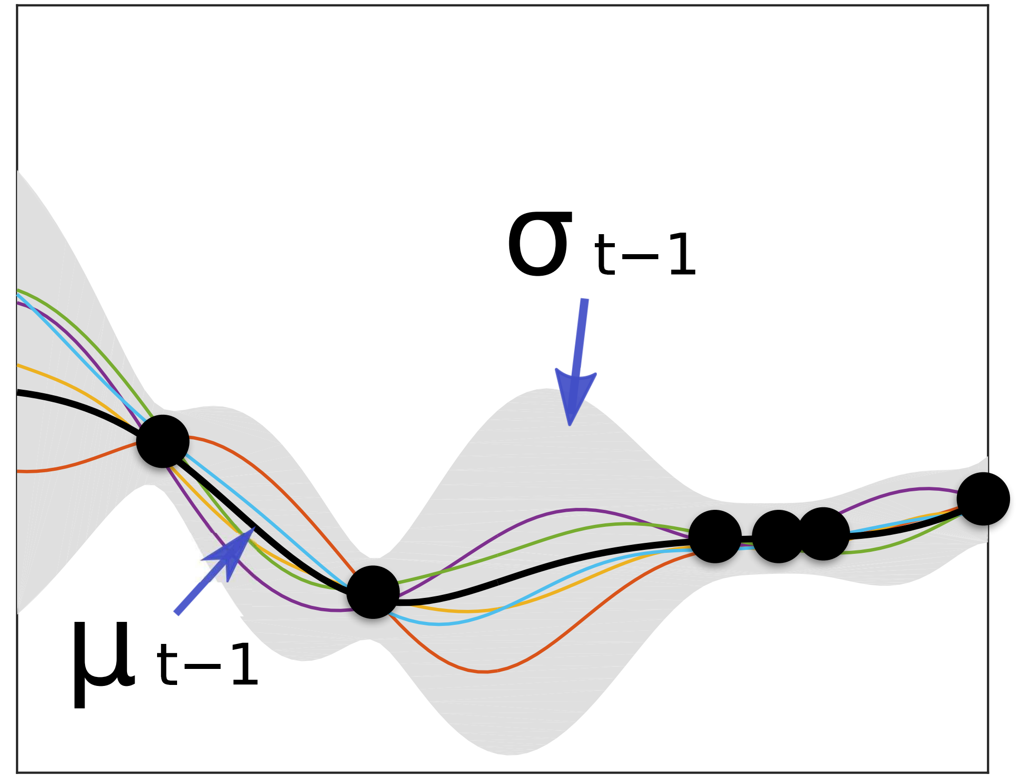
$$y_\tau = f(x_\tau) + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2)$$

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

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

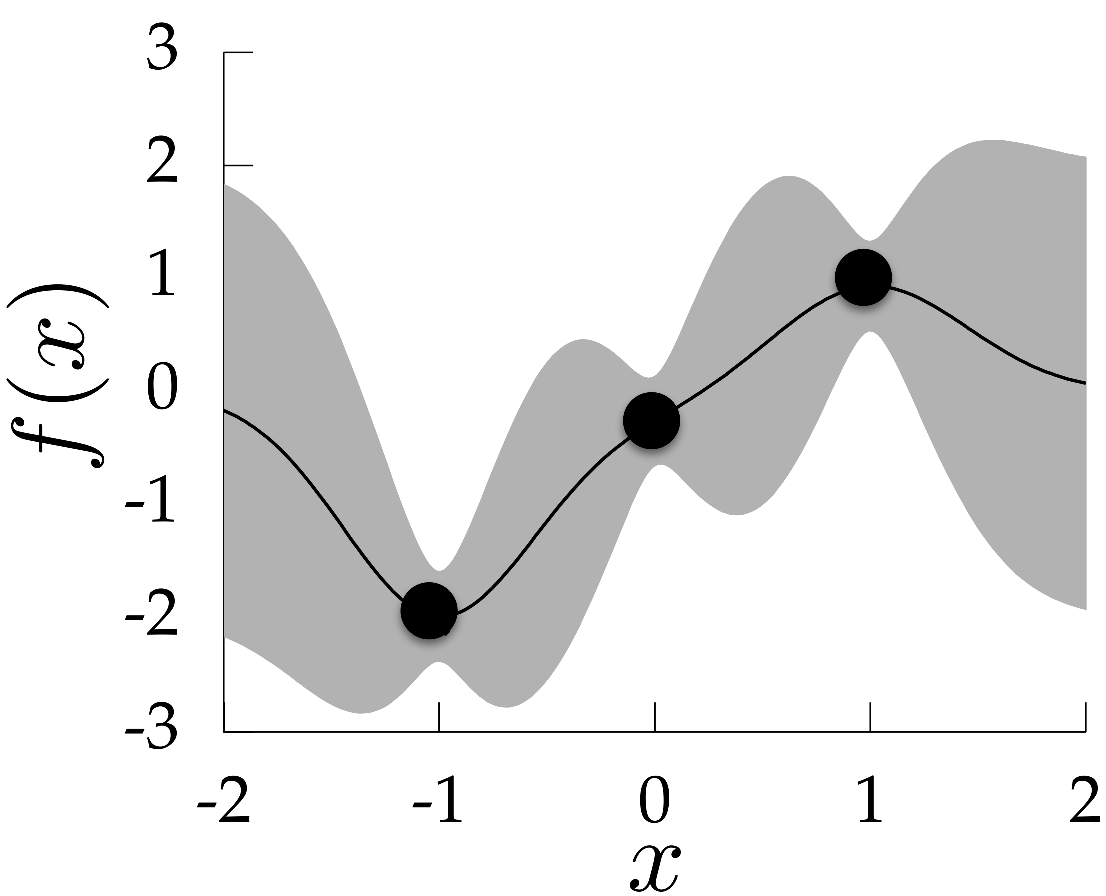
$$\sigma_{t-1}(x)^2 = k(x,x) - k_{t-1}(x)^{\mathrm{T}}(K_{t-1} + \sigma^2 I)^{-1} k_{t-1}(x)$$

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

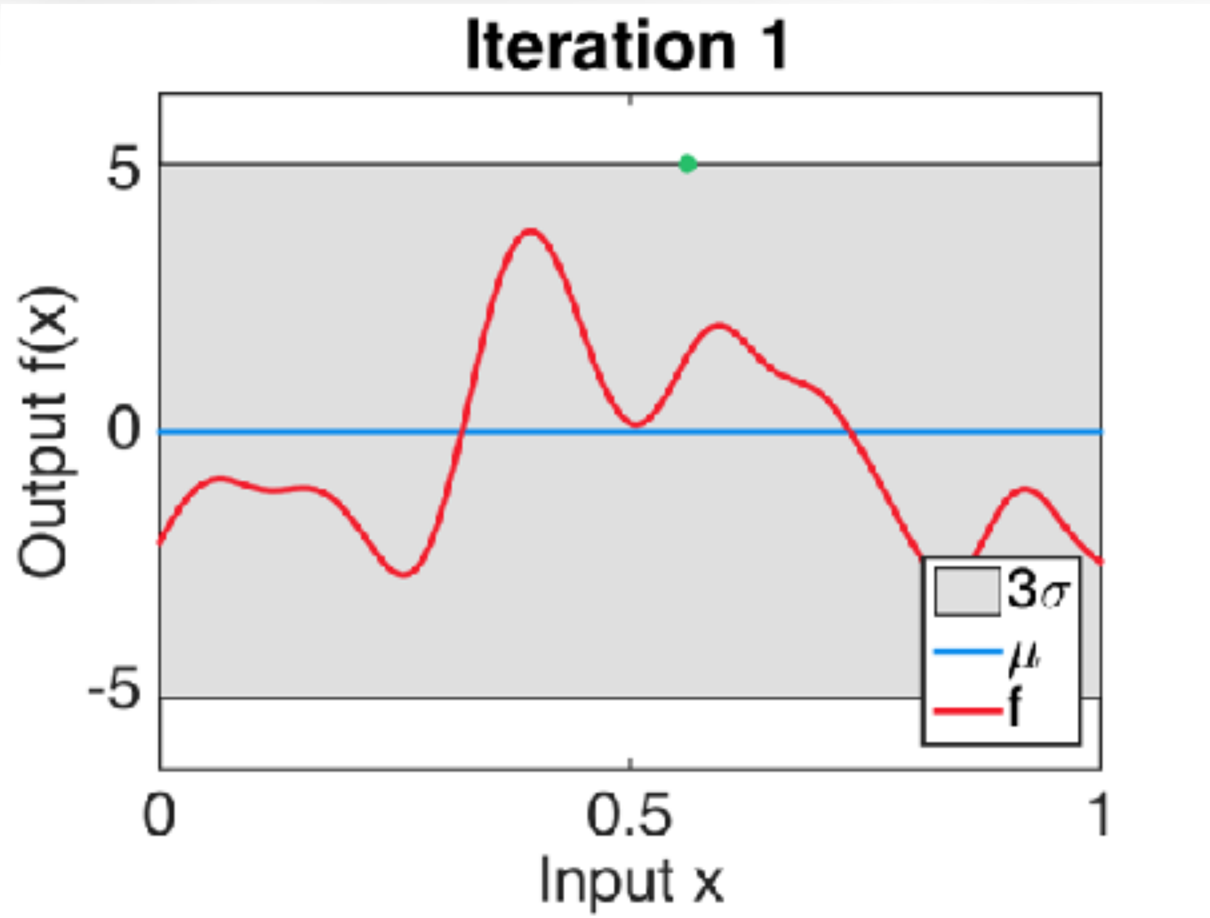


$$x^* = \underset{\mathcal{X}\subset\mathbb{R}^d}{\operatorname{argmax}} f(x)$$

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

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

Gaussian process upper confidence bound



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

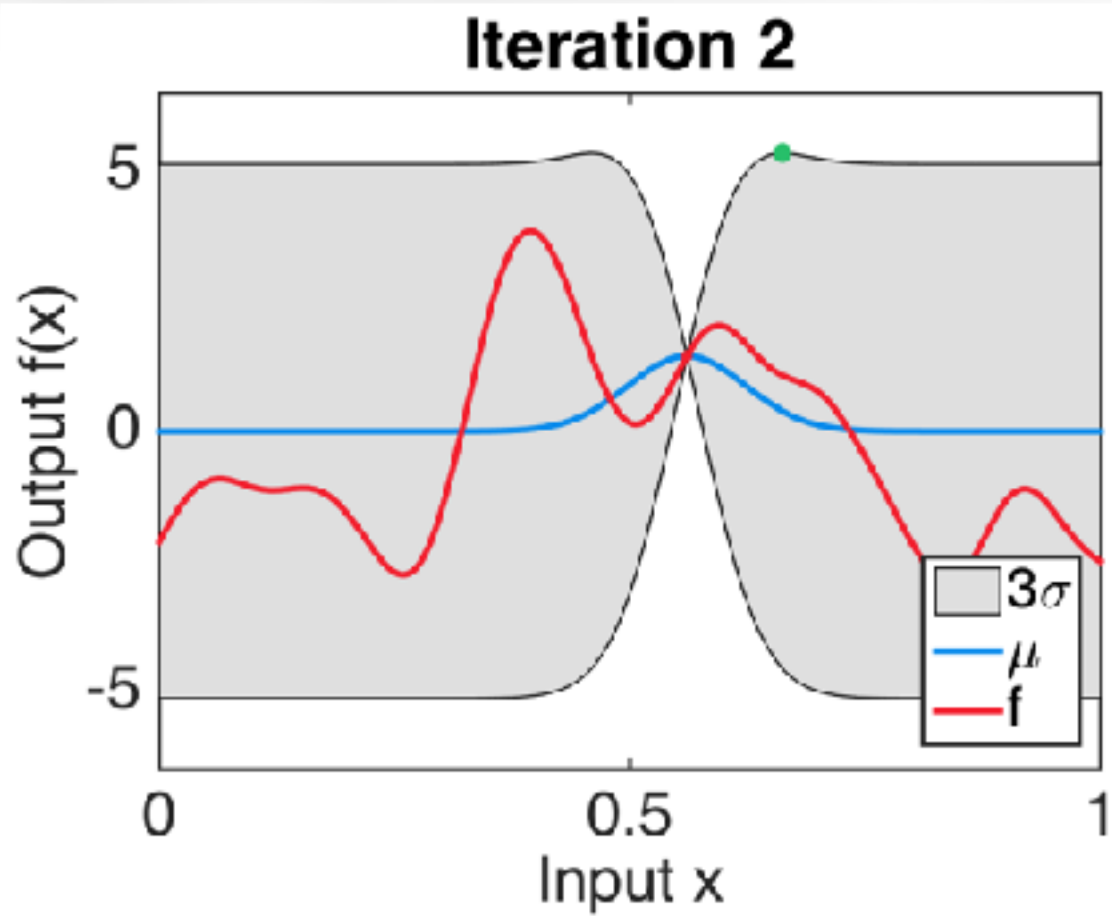
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

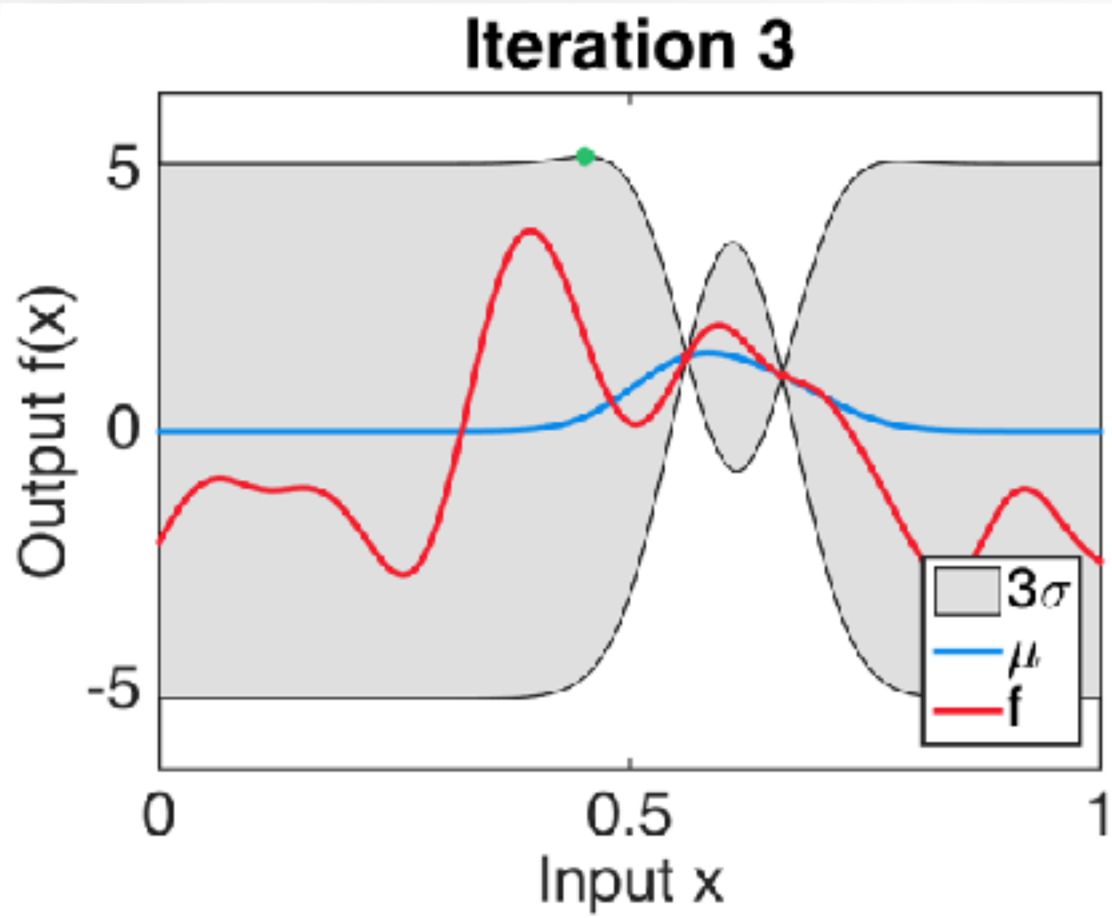
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

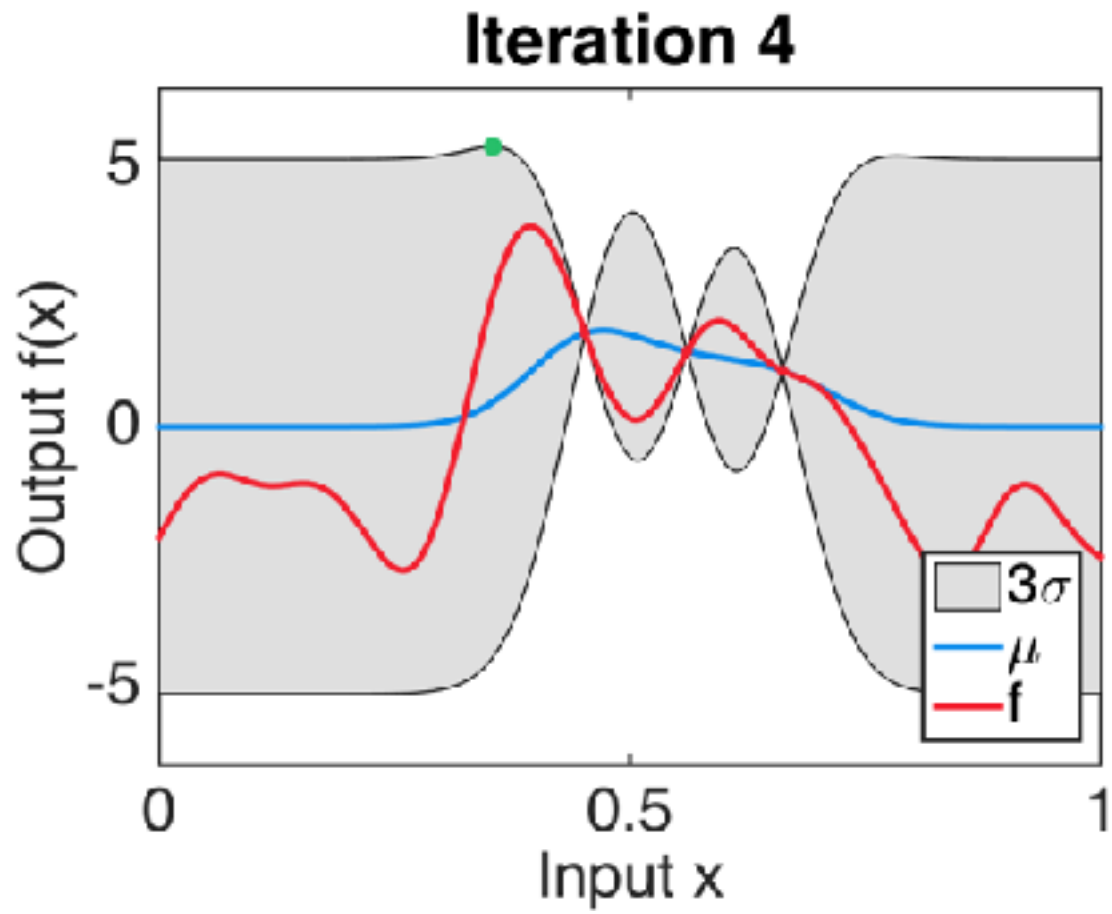
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

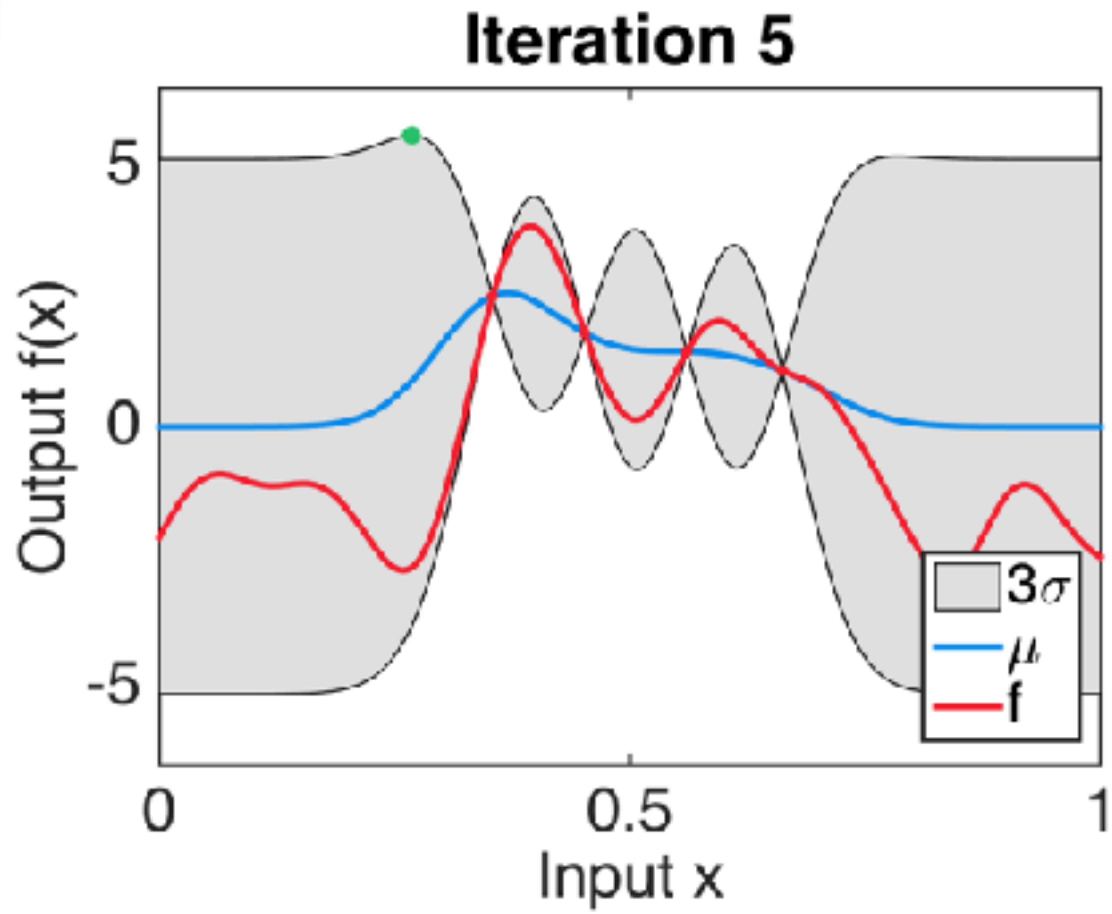
At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- 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: $f \sim GP(\mu, k)$

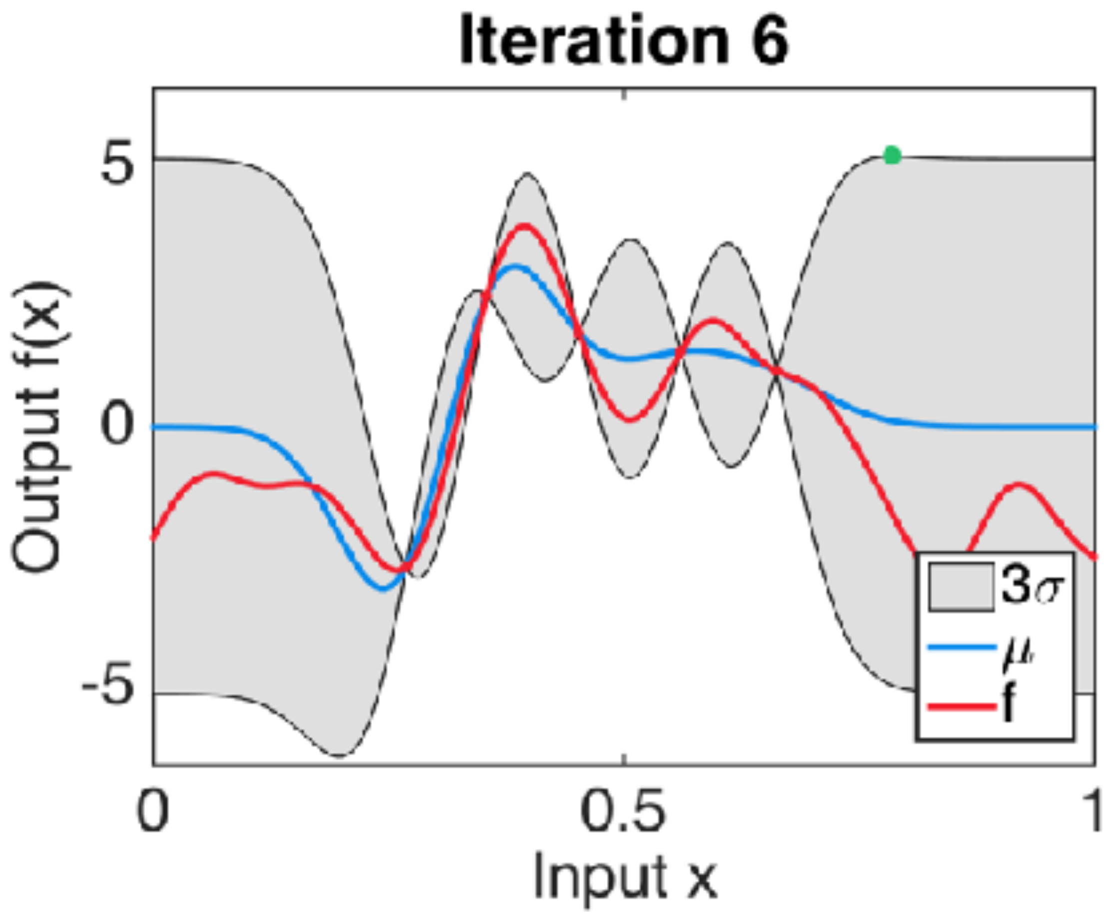
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

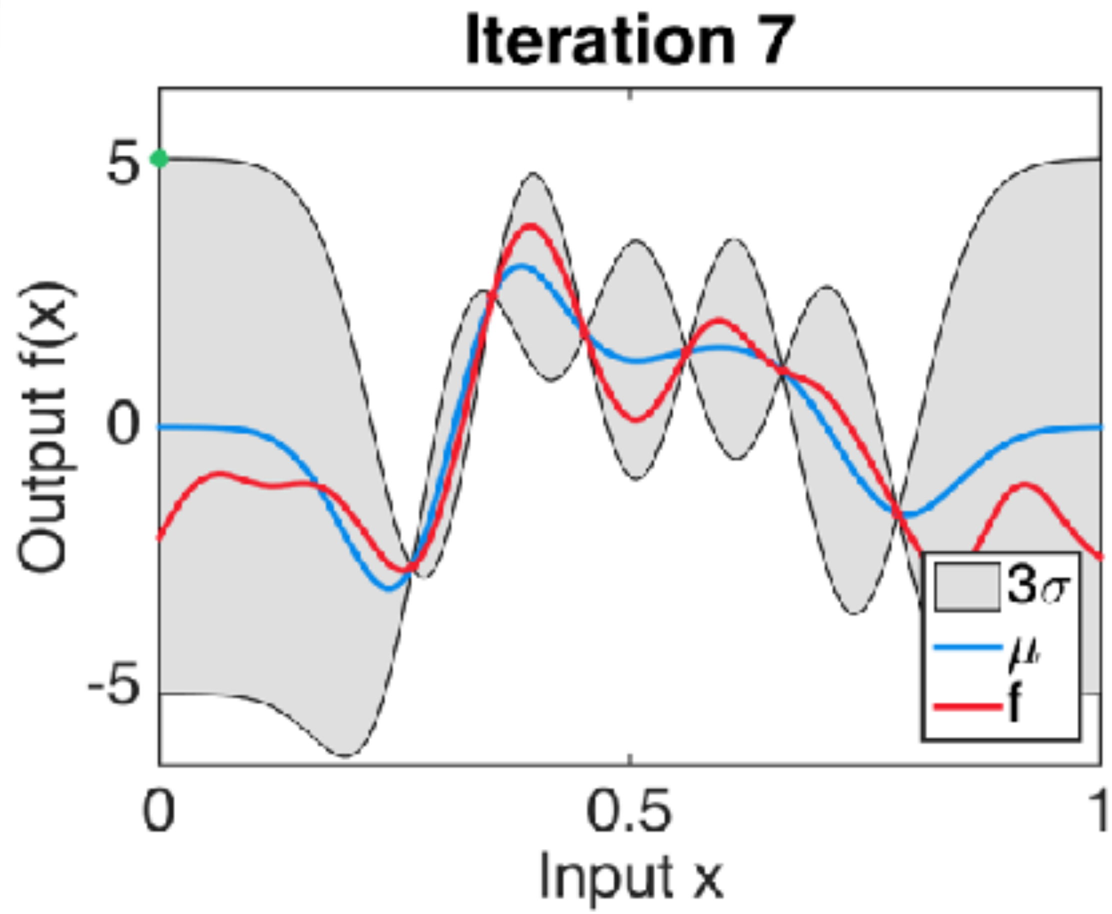
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

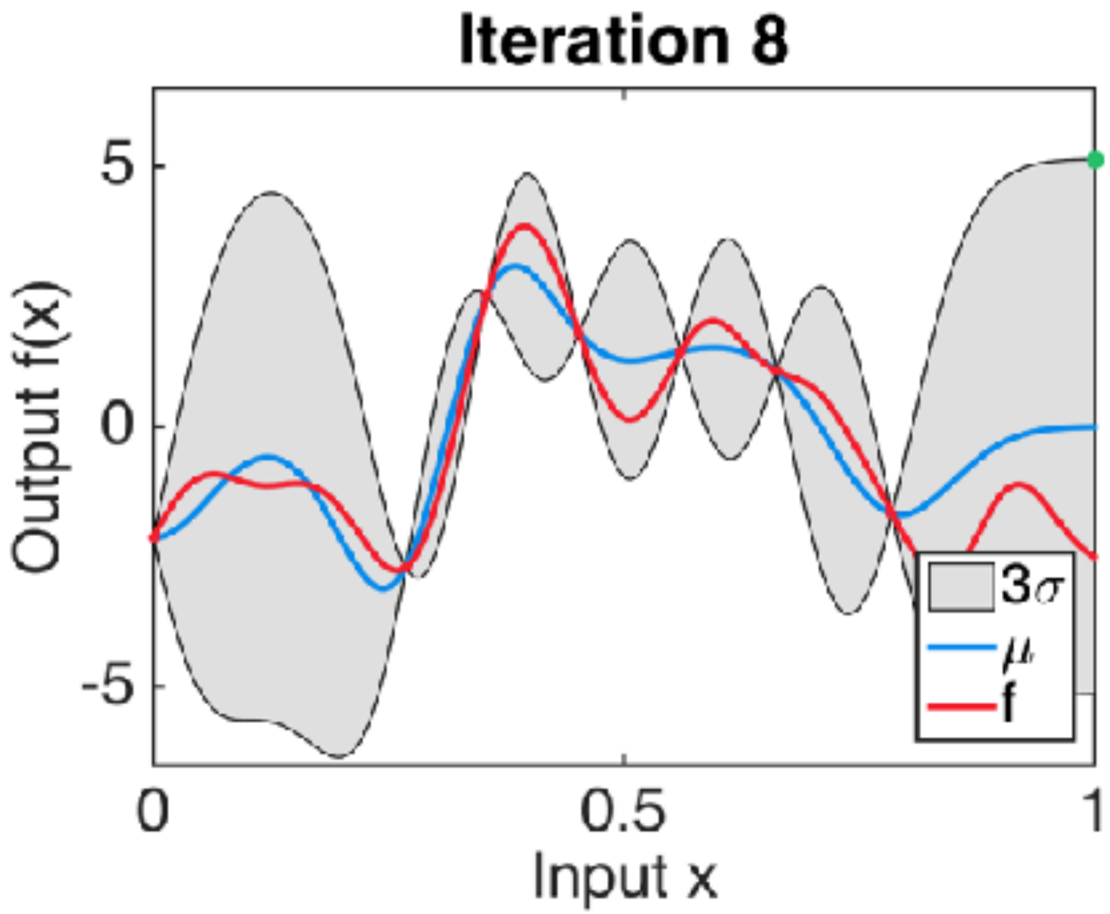
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

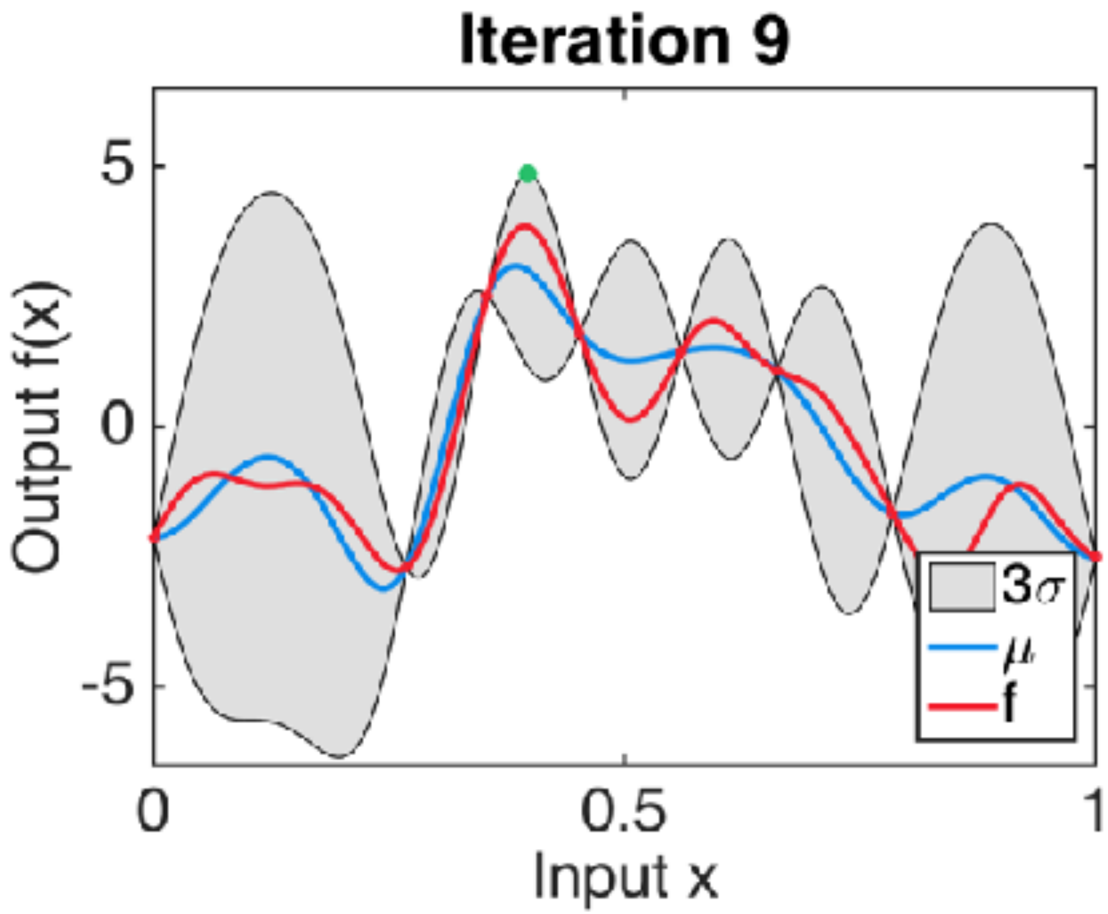
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

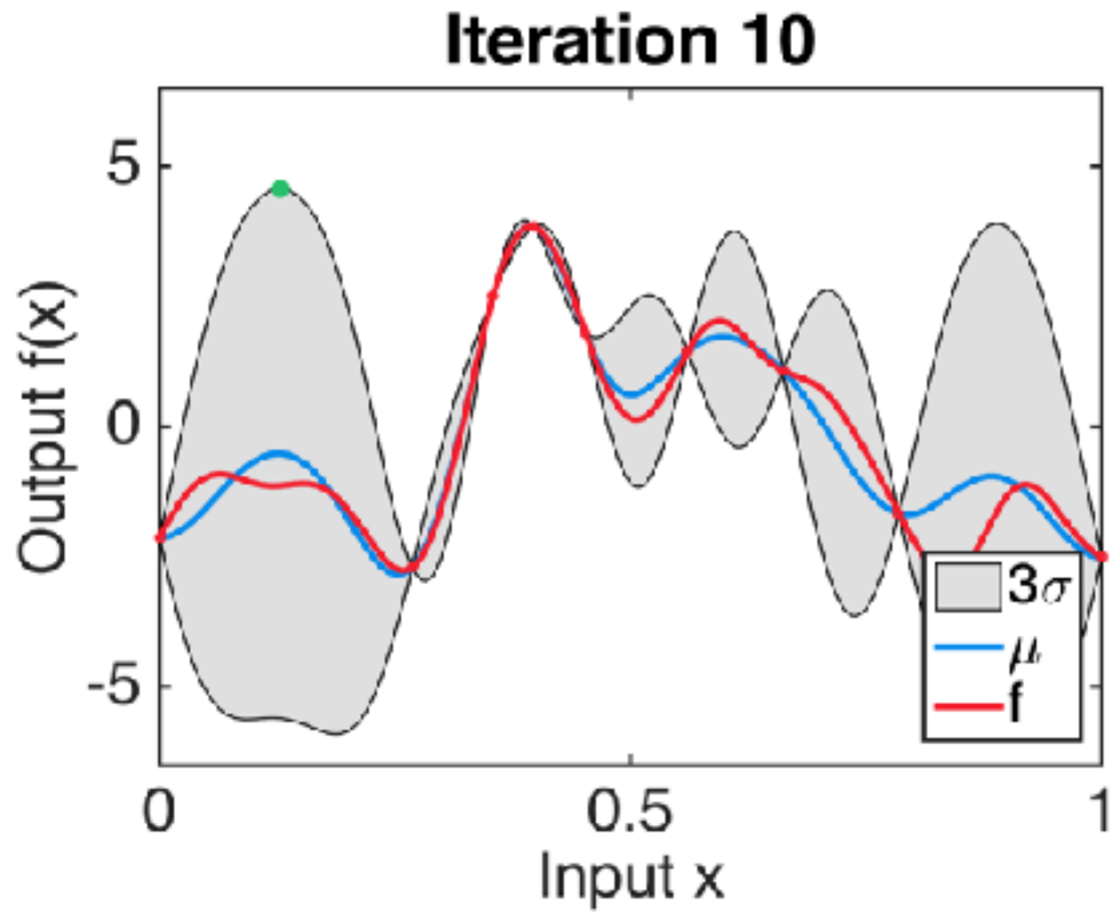
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

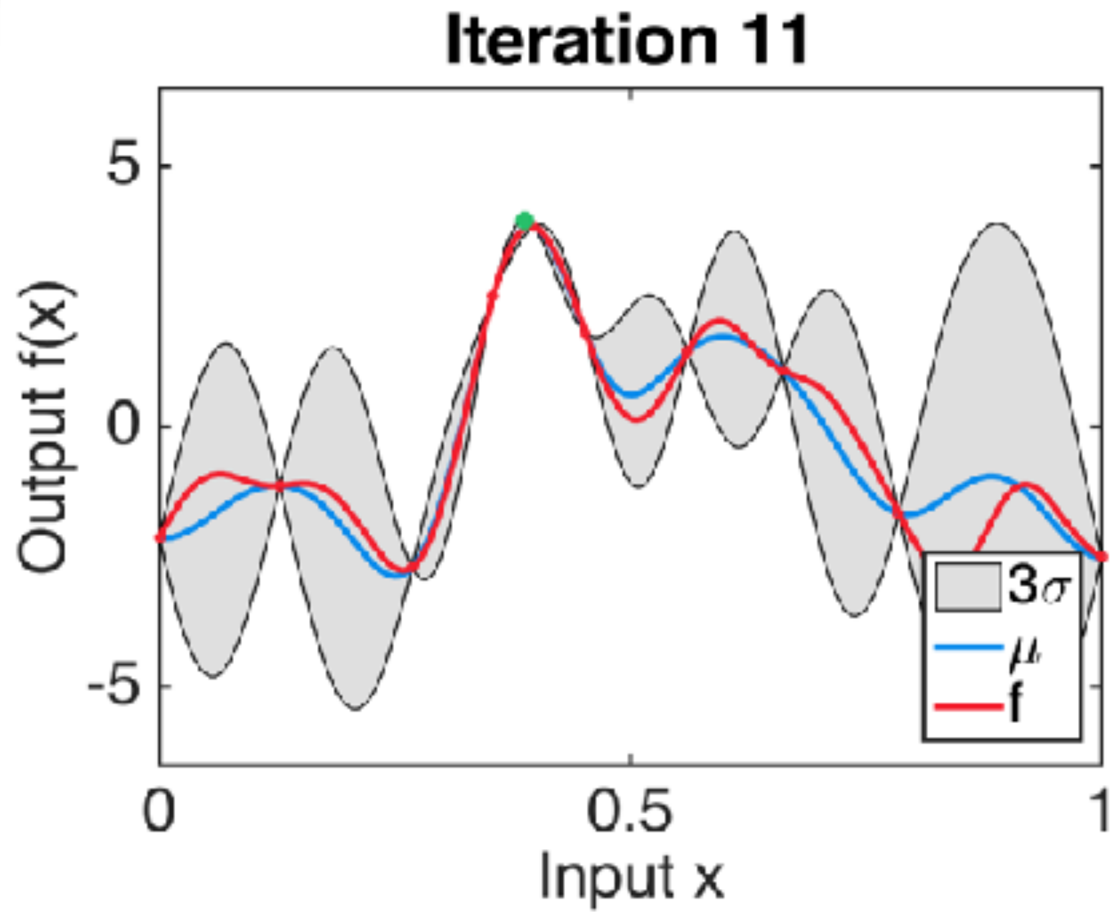
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

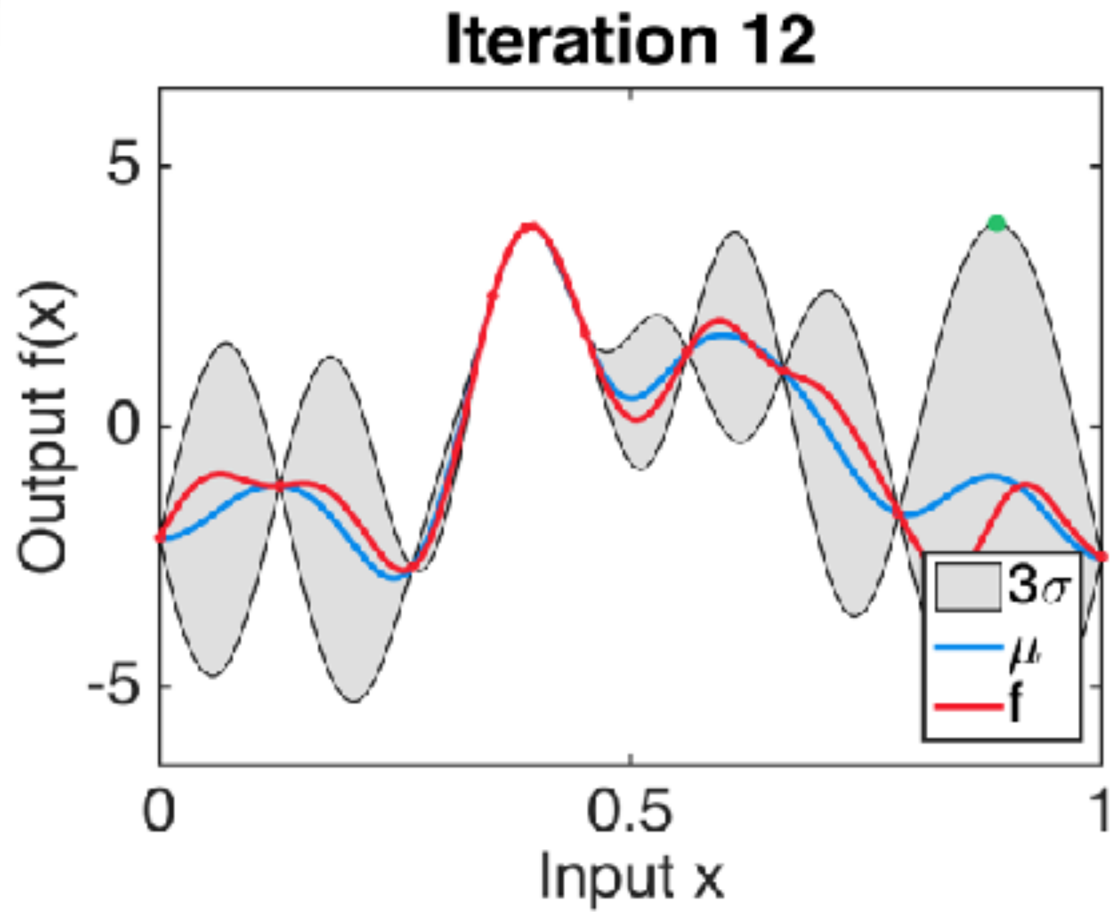
At iteration t,

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

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

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

Gaussian process upper confidence bound



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

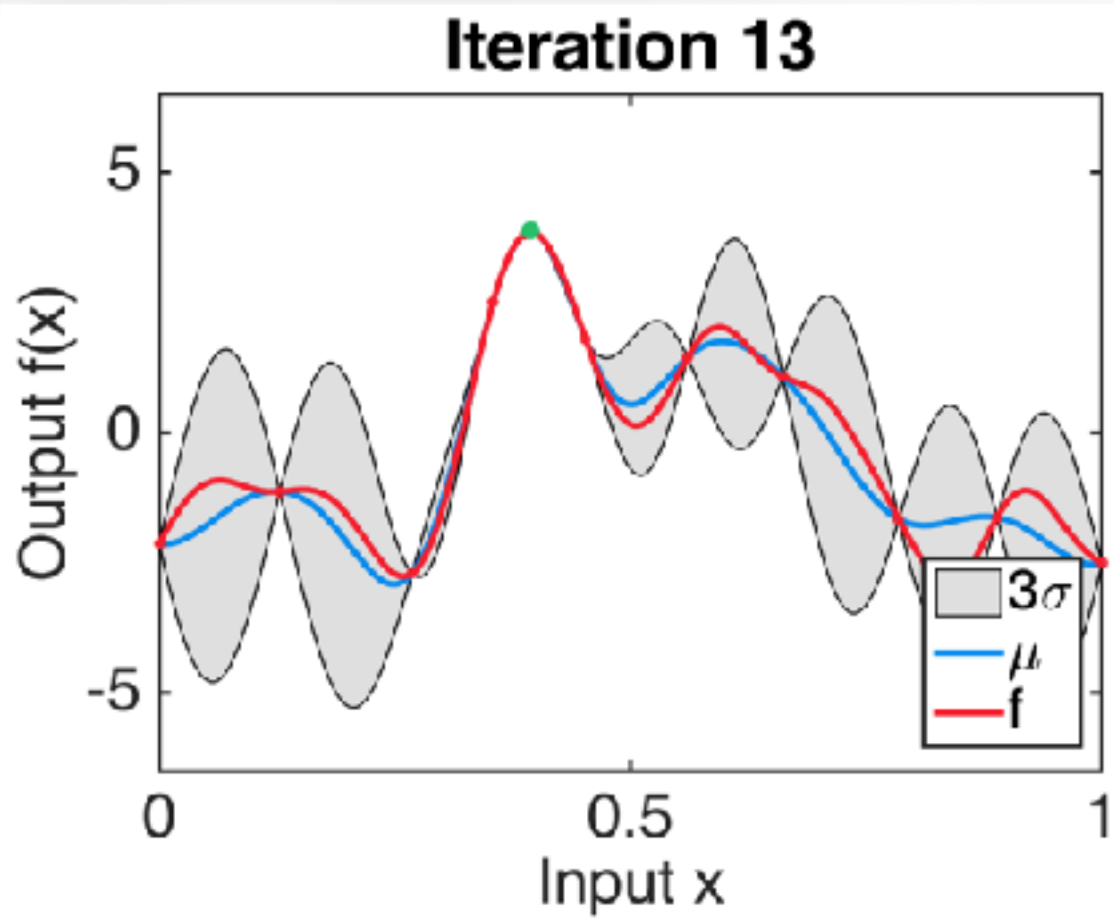
At iteration t,

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

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

(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

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

(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

Zi Wang, MIT CSAIL; Bolei Zhou, MIT CSAIL; Stefanie Jegelka, MIT CSAIL

Max-value Entropy Search for Efficient Bayesian Optimization

Zi Wang, Stefanie Jegelka

Batched High-dimensional Bayesian Optimization via Structural Kernel Learning

Zi Wang, Chengtao Li, Stefanie Jegelka, Pushmeet Kohli

Batched Large-scale Bayesian Optimization in High-dimensional Sp

Zi Wang, MIT CSAIL; Clement Gehring, MIT CSAIL; Pushmeet Kohli, DeepMind; Stefanie J

[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
- …

this talk

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

[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

Which comes first?
Data or prior?

- Regret bounds exist only when prior is assumed given

Bayesian optimization with an unknown prior

Estimate "prior" from data
- maximum likelihood
- hierarchical Bayes

Which comes first?
Data or prior?

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

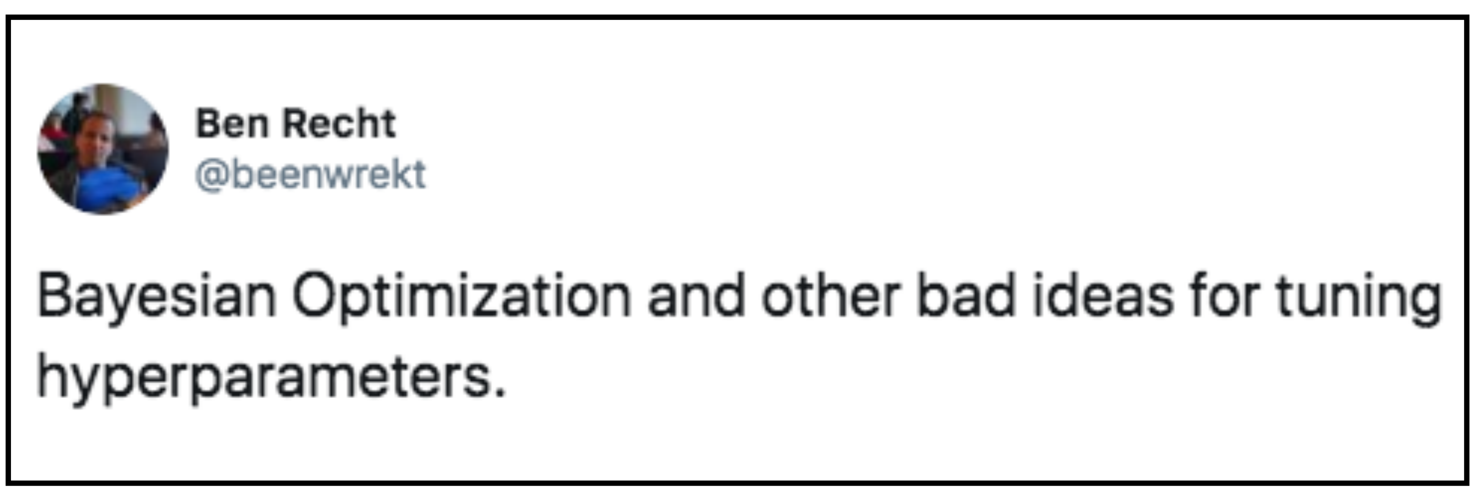
Bayesian optimization with an unknown prior

Estimate “prior” from data

- maximum likelihood
- hierarchical Bayes

Which comes first? Data or prior?

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



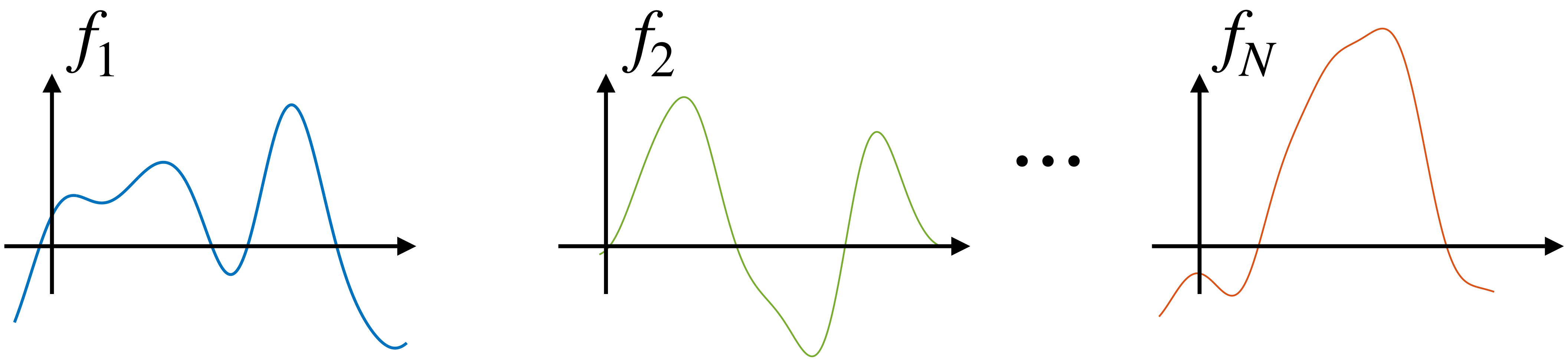
Bayesian optimization with an unknown prior

- 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

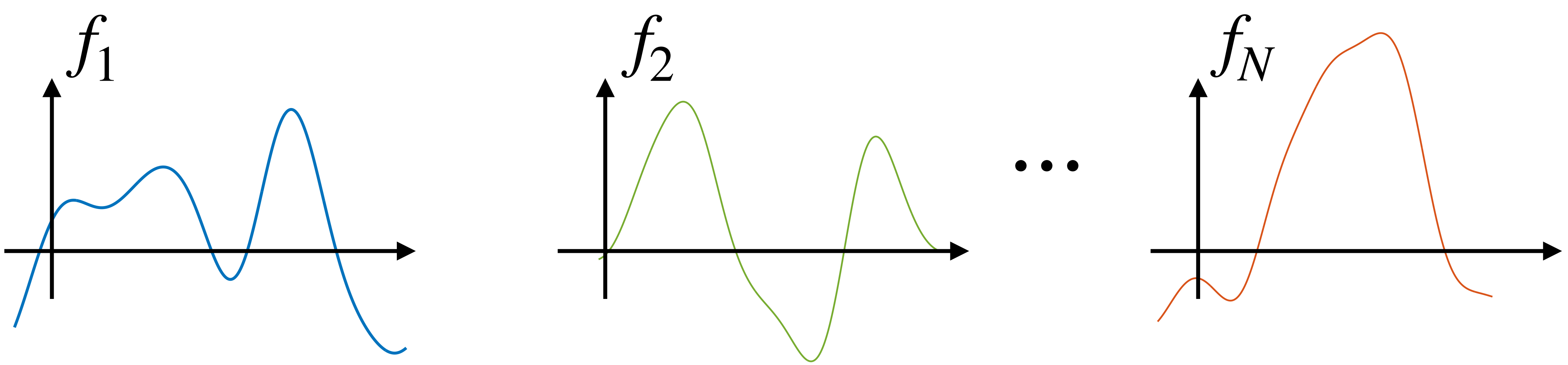
- Our idea: learn the "prior" from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$



[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 $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$

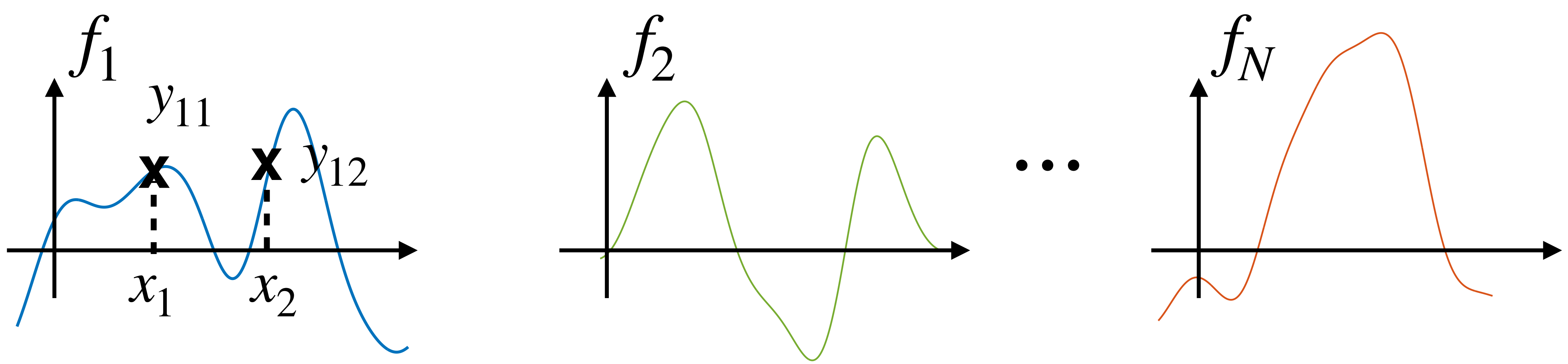


$[x_1, \cdots, x_M]$ $\quad y_{ij} \sim \mathcal{N}(f_i(x_j), \sigma^2)$

[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 $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$

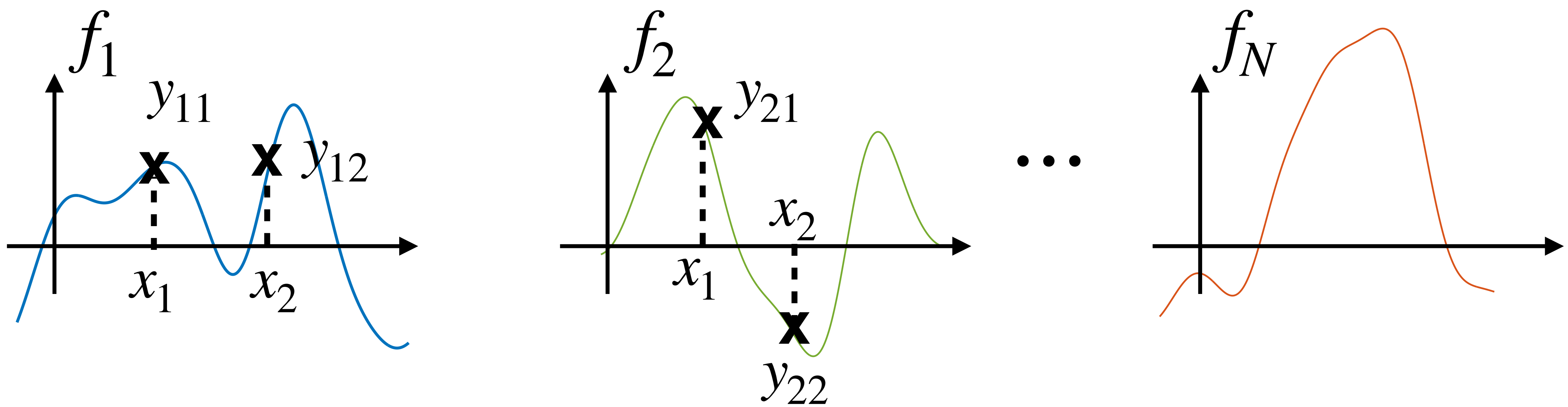


$[x_1, \cdots, x_M]$ $y_{ij} \sim \mathcal{N}(f_i(x_j), \sigma^2)$

[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 $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$

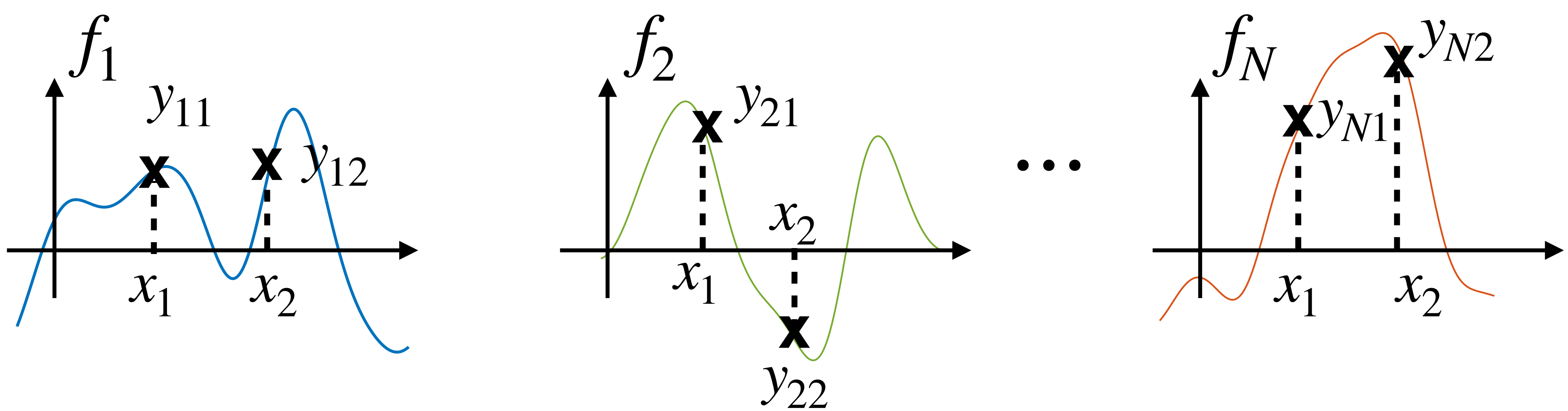


$[x_1, \cdots, x_M]$ $\quad y_{ij} \sim \mathcal{N}(f_i(x_j), \sigma^2)$

[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 $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$



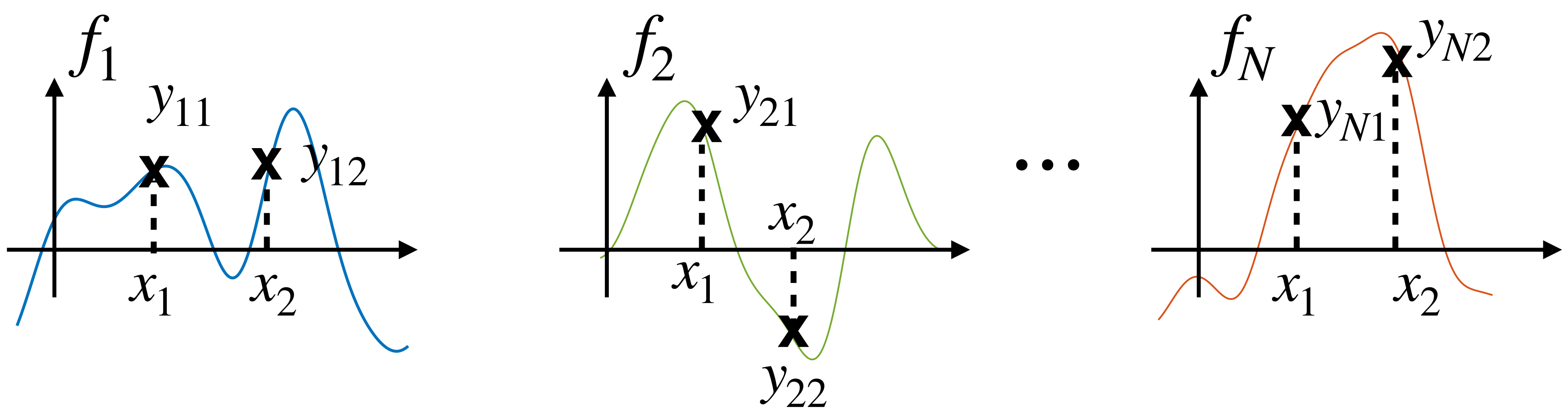
$[x_1, \cdots, x_M]$ $\quad$ $y_{ij} \sim \mathcal{N}(f_i(x_j), \sigma^2)$

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

Bayesian optimization with an unknown prior

meta / multi-task / transfer learning

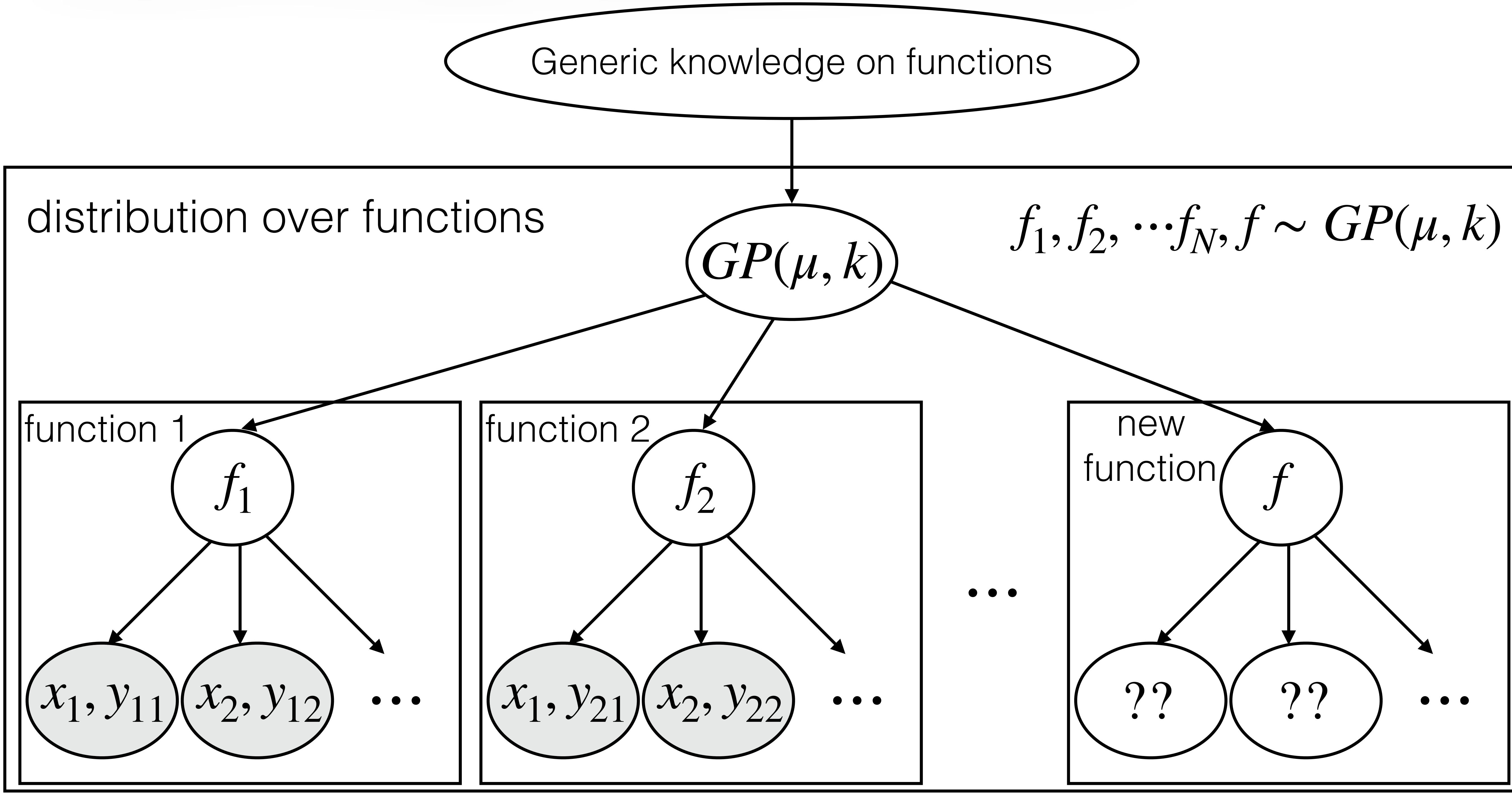
- Our idea: learn the "prior" from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$



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

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

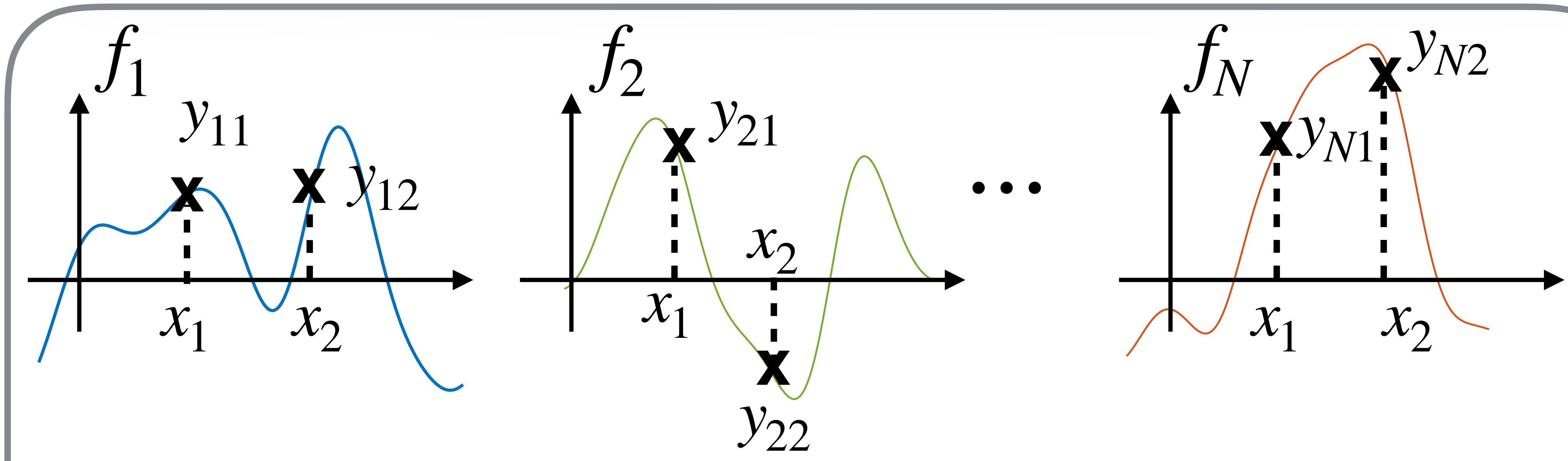
Bayesian optimization with an unknown prior



observations of function values

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

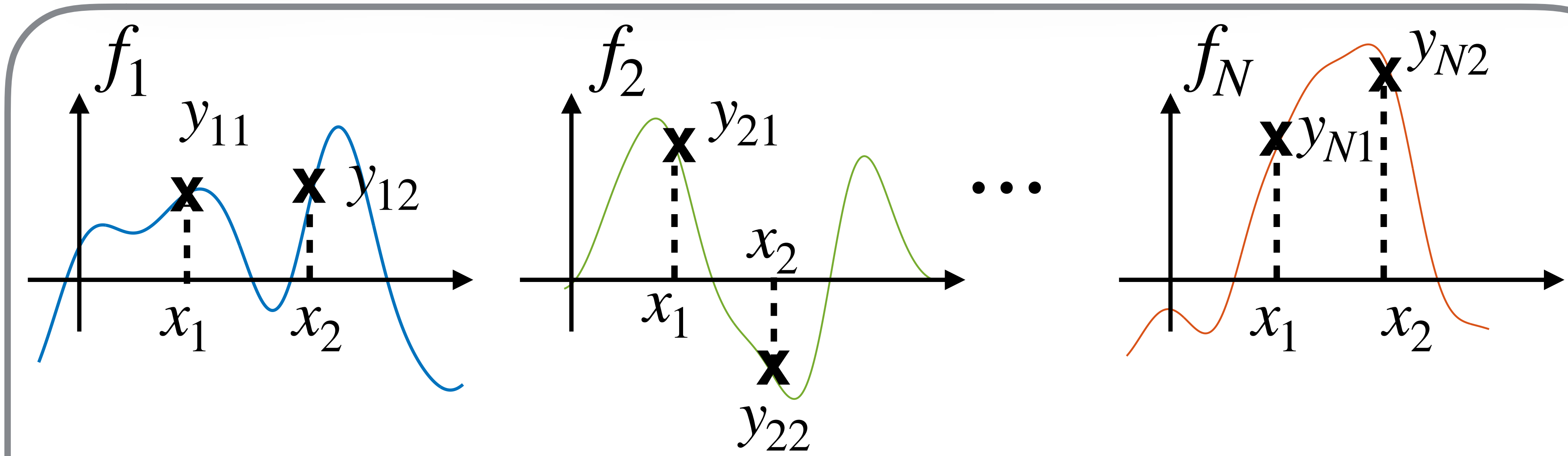
Meta Bayesian optimization with an unknown prior



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

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

Meta Bayesian optimization with an unknown prior

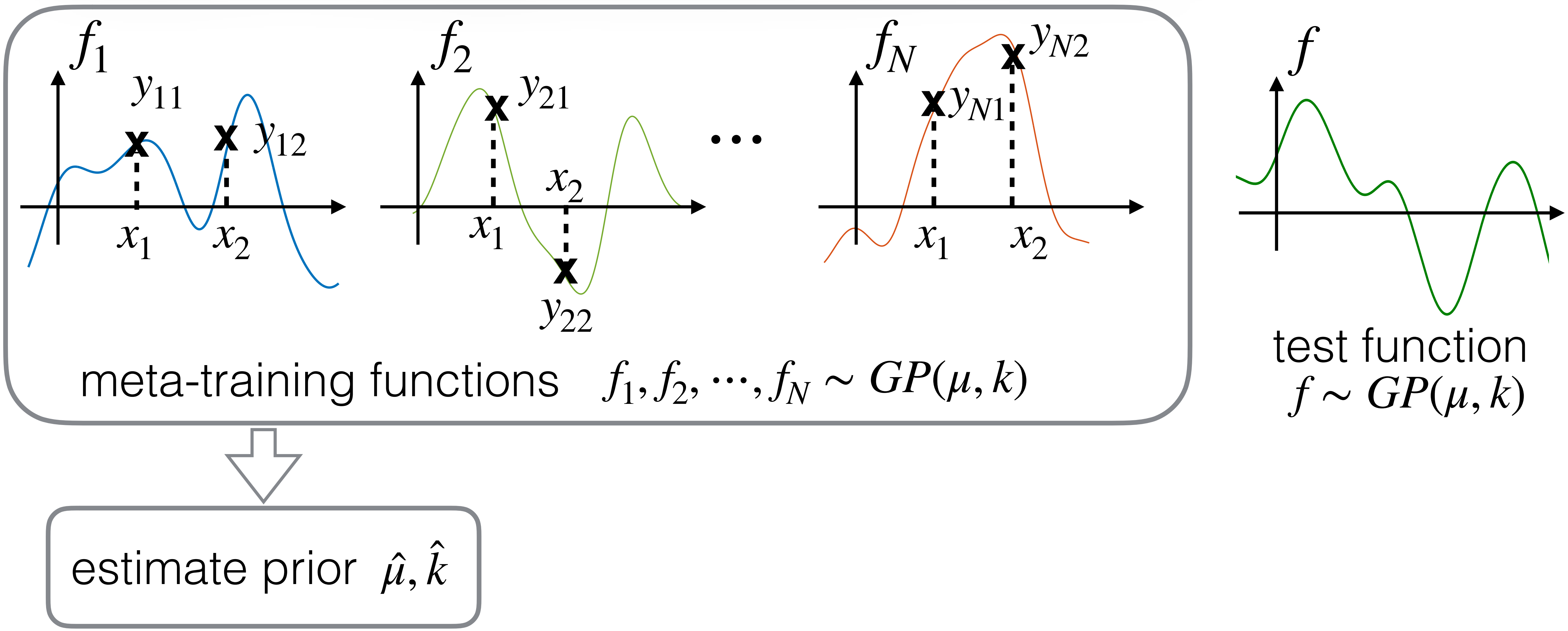


meta-training functions $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$

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

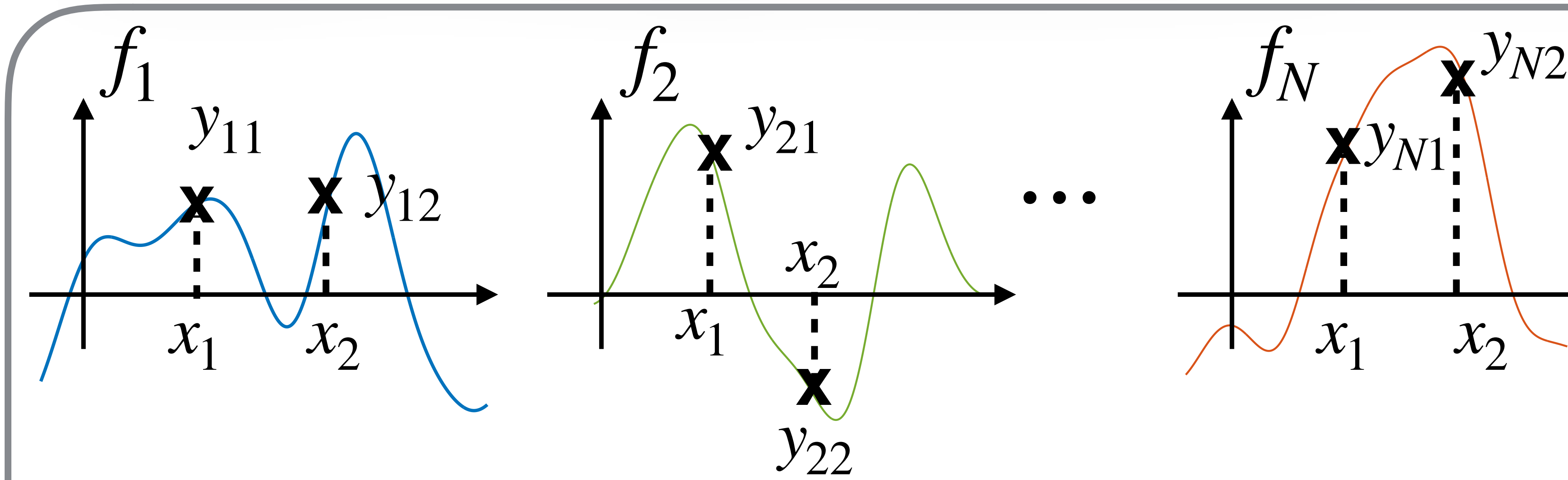
[Wang*&Kim*&Kaelbling, NIPS 2018]

Meta Bayesian optimization with an unknown prior

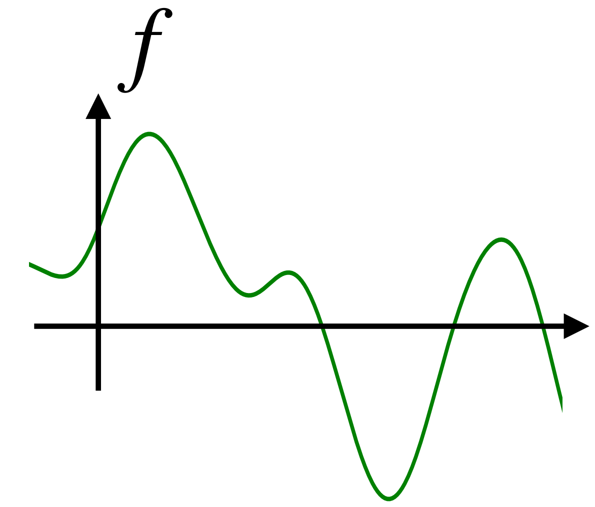


[Wang*&Kim*&Kaelbling, NIPS 2018]

Meta Bayesian optimization with an unknown prior



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



test function
$f \sim GP(\mu, k)$

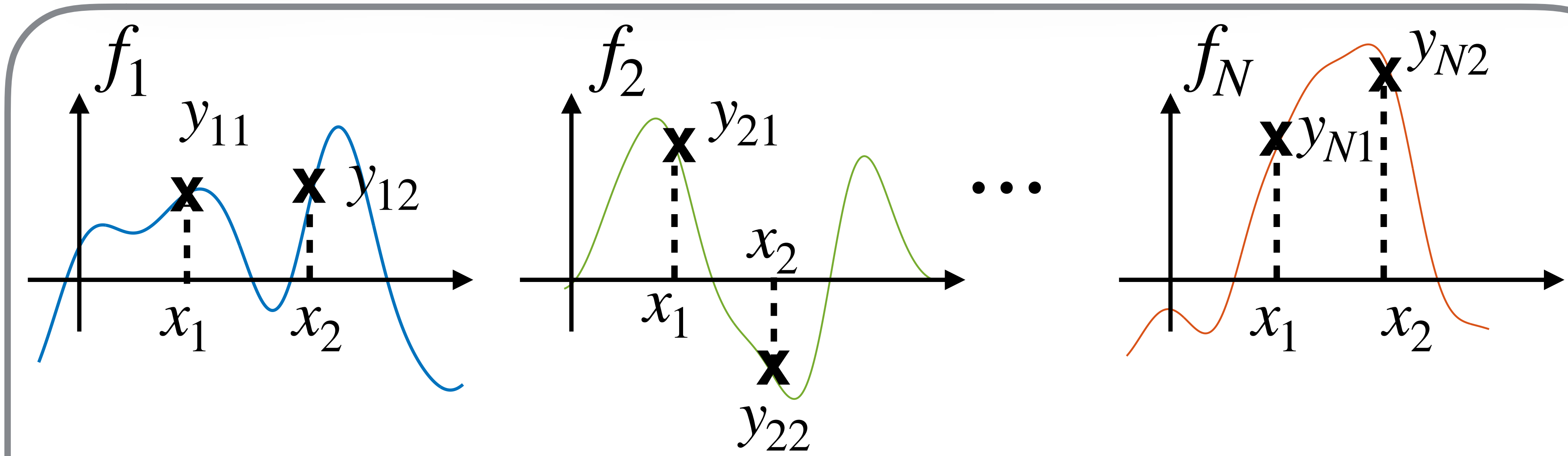
estimate prior $\hat{\mu}, \hat{k}$

Optimize test function f with GP-UCB

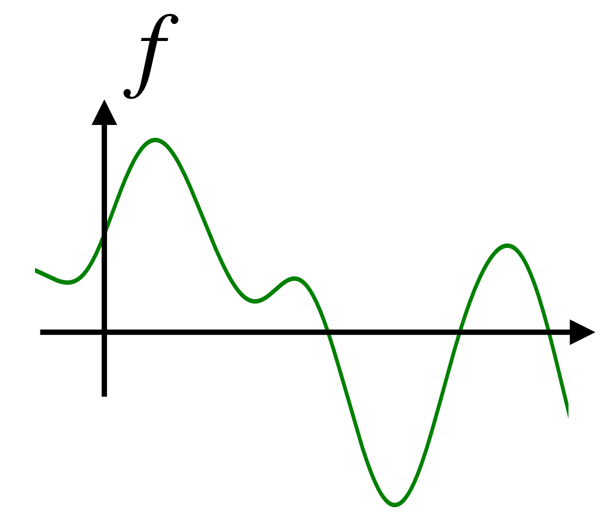
$$x_t = \text{argmax } \hat{\mu}_{t-1}(x) + \beta_t \hat{\sigma}_{t-1}(x)$$

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

Meta Bayesian optimization with an unknown prior



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



test function
$f \sim GP(\mu, k)$

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

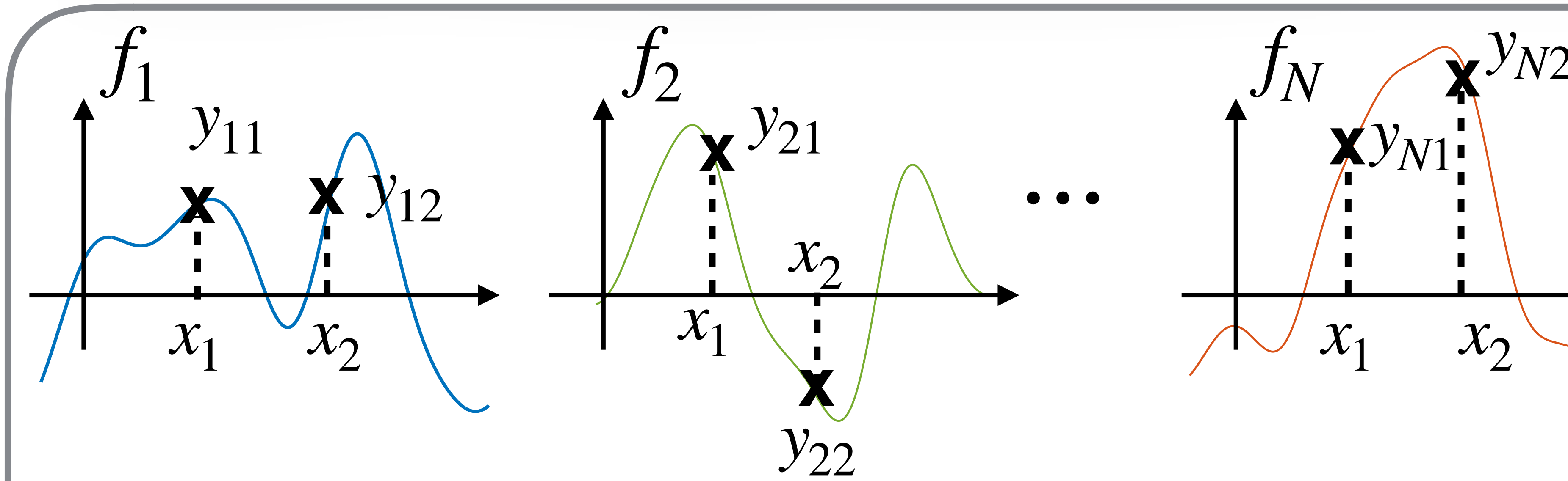
Optimize test function f with GP-UCB

$$x_t = \text{argmax}\ \hat{\mu}_{t-1}(x) + \beta_t \hat{\sigma}_{t-1}(x)$$

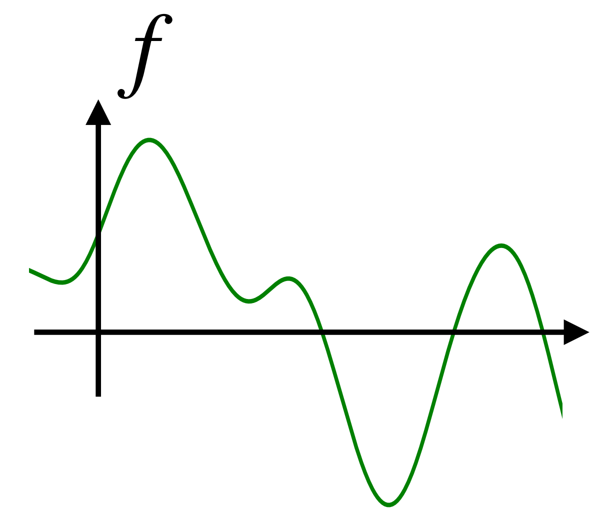
$$\beta_t = \frac{\left(6(N-3+t+2\sqrt{t\log\frac{6}{\delta}} + 2\log\frac{6}{\delta})/(\delta N(N-t-1))\right)^{\frac{1}{2}} + (2\log(\frac{3}{\delta}))^{\frac{1}{2}}}{(1-2(\frac{1}{N-t}\log\frac{6}{\delta})^{\frac{1}{2}})^{\frac{1}{2}}}$$

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

Meta Bayesian optimization with an unknown prior



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



test function
$f \sim GP(\mu, k)$

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

Optimize test function f with GP-UCB

$$x_t = \text{argmax}\ \hat{\mu}_{t-1}(x) + \beta_t \hat{\sigma}_{t-1}(x)$$

$\beta_t(N, \delta)$

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)

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

[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

T observations on the test function f

[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

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

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)

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 $1 - \delta$,

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)

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)

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 $1 - \delta$,

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 $|\mathscr{X}| = M$

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

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

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

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

Gaussian distribution

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

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

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

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

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

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

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

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

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

unbiased posterior estimator

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

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

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

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

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

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

Missing entries? Matrix completion.

unbiased prior estimator

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

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

unbiased posterior estimator

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

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

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

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

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

unbiased posterior estimator

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

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

bias correction

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

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

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

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

Missing entries? Matrix completion.

unbiased prior estimator

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

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

unbiased posterior estimator

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

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

$\sim$ **Wishart distribution**

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

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

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

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

Missing entries? Matrix completion.

unbiased prior estimator

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

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

unbiased posterior estimator

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

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

Continuous input space

assume basis functions are given!

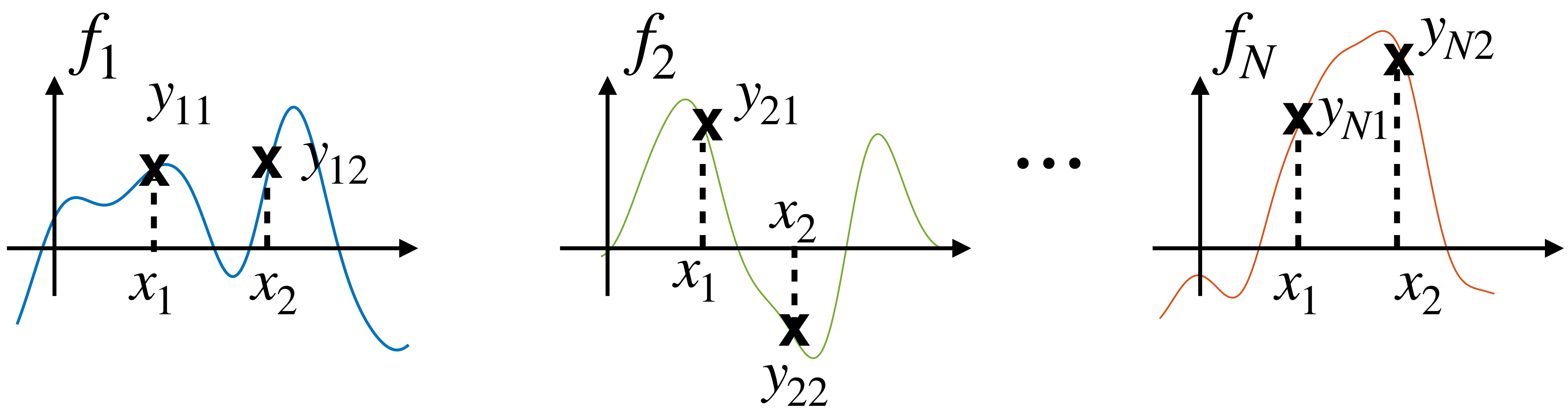
basis functions

mean parameter

covariance parameter

$$f = \Phi(x)^T W \sim GP(\mu, k) \qquad W \sim \mathcal{N}(\boldsymbol{u}, \Sigma)$$

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

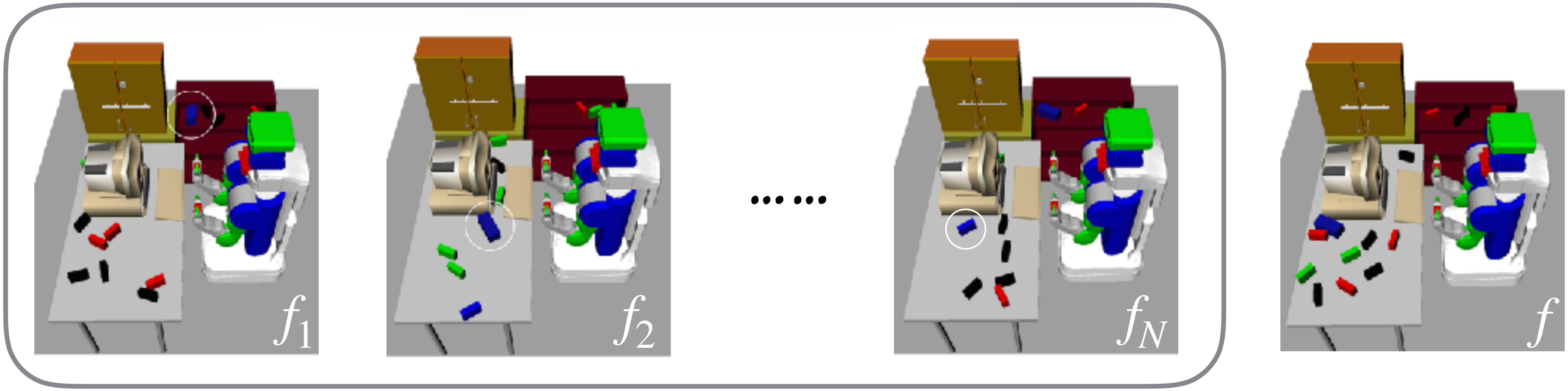


$$\Phi(X)^T W_i = Y_i \implies \boxed{W_i = (\Phi(X)\Phi(X)^T)^{-1}\Phi(X)Y_i}$$

estimate its mean and covariance to construct GP prior

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

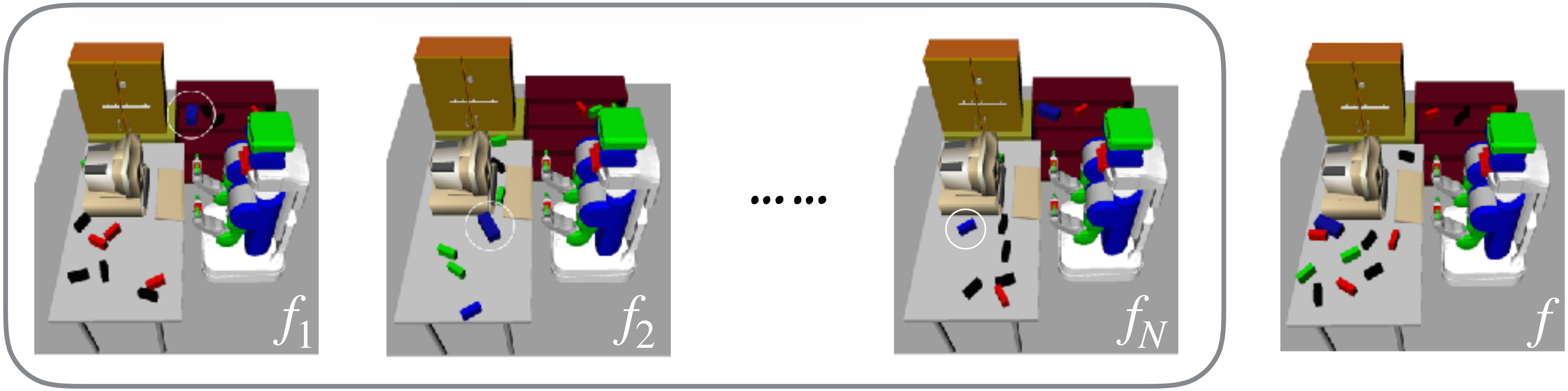
Empirical results on block picking and placing



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

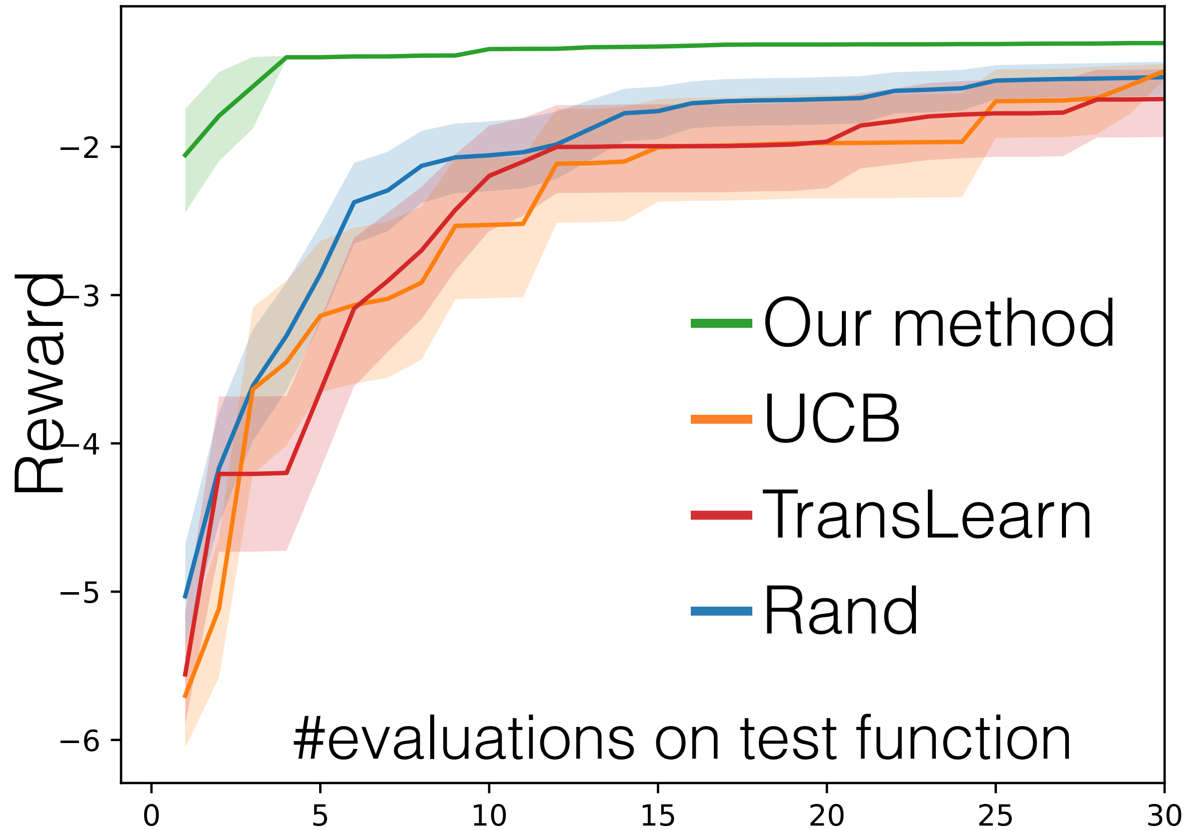
test function

Empirical results on block picking and placing



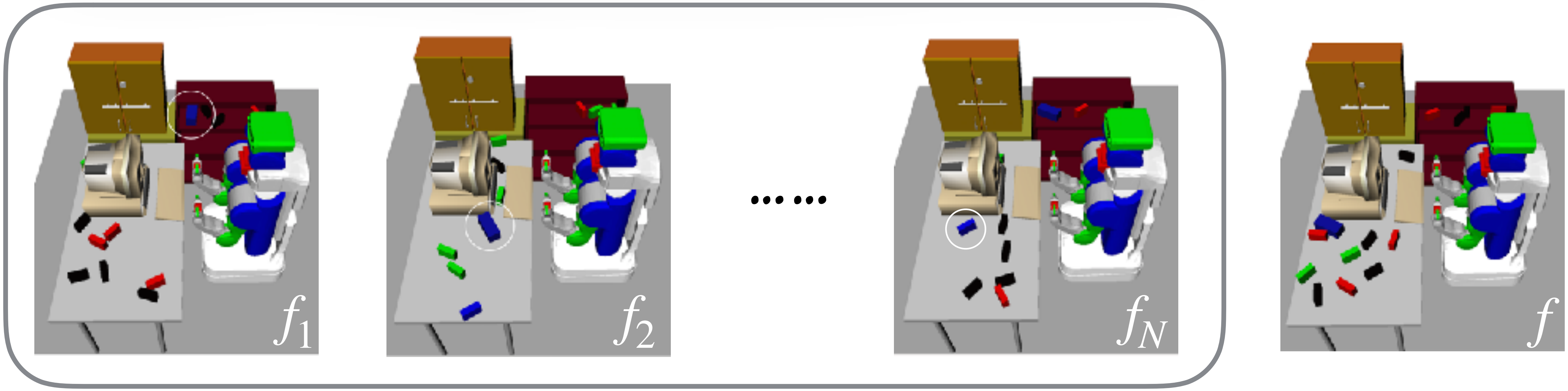
meta-training data $N = 1500, M = 1000$

test function

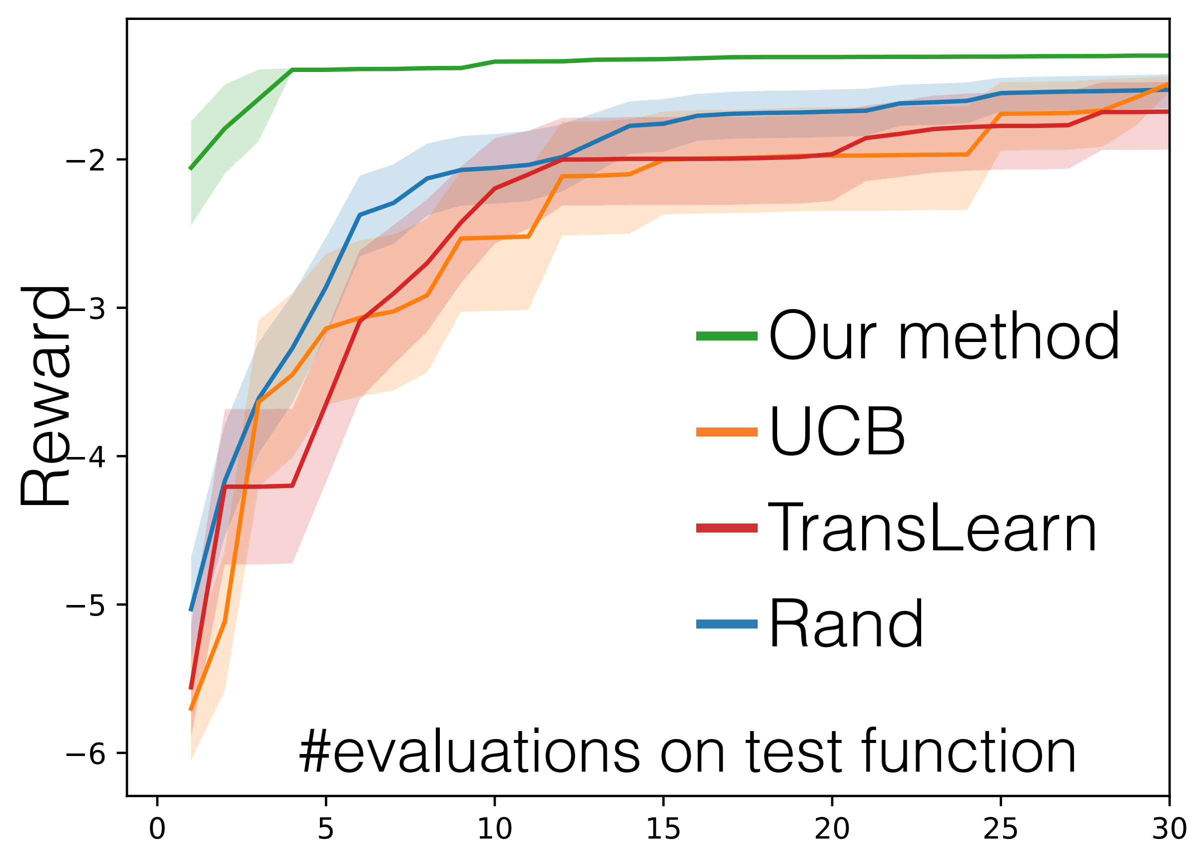


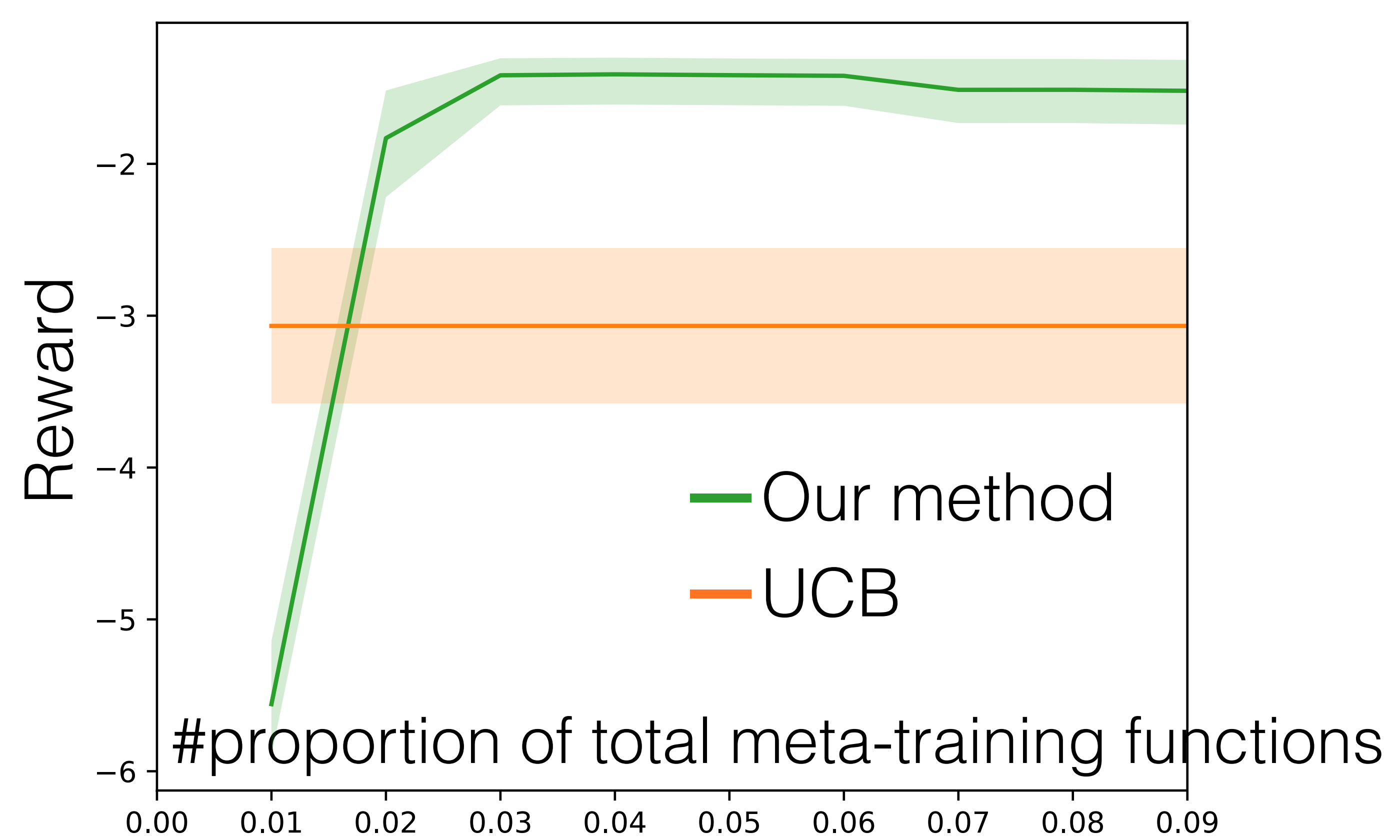
[Wang*&Kim*&Kaelbling, NIPS 2018]

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?