Misspecified Gaussian Process Bandits

Ilija Bogunovic and Andreas Krause

NeurIPS, 2021

GP (kernel) bandits

- Problem setup:
  - Unknown reward function $f^\star : \mathscr{D} \to \mathbb{R}, \quad \mathscr{D} \subset \mathbb{R}^d$ over a known infinite and compact set of actions
  - Interaction over T rounds: Select action $x_t \in \mathscr{D}$ and obtain noisy (sub-Gaussian) observation $y_t = f^\star(x_t) + \eta_t$
- Goal: Minimize cumulative regret $R_T = \sum_{t=1}^{T} \left(\max_{x \in D} f^\star(x) - f^\star(x_t) \right)$ (optimizing while learning)

GP (kernel) bandits

- Problem setup:
  - Unknown reward function $f^\star : \mathscr{D} \to \mathbb{R}$, $\mathscr{D} \subset \mathbb{R}^d$ over a known infinite and compact set of actions
  - Interaction over T rounds: Select action $x_t \in \mathscr{D}$ and obtain noisy (sub-Gaussian) observation $y_t = f^\star(x_t) + \eta_t$
- Goal: Minimize cumulative regret $R_T = \sum_{t=1}^{T} \left(\max_{x \in D} f^\star(x) - f^\star(x_t) \right)$ (optimizing while learning)

- Hypothesis class:
  - Defined via positive semi-definite kernel $k(\cdot, \cdot)$ (e.g., squared exponential, Matern, NTK)
  - Smooth functions with bounded RKHS norm $\mathscr{F}_k(\mathscr{D}; B) = \{f \in \mathscr{H}_k(\mathscr{D}) : \|f\|_k \le B\}$
- Realizable setting: Assumed hypothesis class is the correct one, i.e., $f^\star \in \mathscr{F}_k(\mathscr{D}; B)$

GP (kernel) bandits

- Problem setup:
  - Unknown reward function $f^\star : \mathscr{D} \to \mathbb{R}$, $\mathscr{D} \subset \mathbb{R}^d$ over a known infinite and compact set of actions
  - Interaction over T rounds: Select action $x_t \in \mathscr{D}$ and obtain noisy (sub-Gaussian) observation $y_t = f^\star(x_t) + \eta_t$
- Goal: Minimize cumulative regret $R_T = \sum_{t=1}^{T} \left(\max_{x \in D} f^\star(x) - f^\star(x_t) \right)$ (optimizing while learning)

- Hypothesis class:
  - Defined via positive semi-definite kernel $k(\cdot,\cdot)$ (e.g., squared exponential, Matern, NTK)
  - Smooth functions with bounded RKHS norm $\mathscr{F}_k(\mathscr{D}; B) = \{ f \in \mathscr{H}_k(\mathscr{D}) : \|f\|_k \le B \}$
- Realizable setting: Assumed hypothesis class is the correct one, i.e., $f^\star \in \mathscr{F}_k(\mathscr{D}; B)$

- Applications: Molecular and material design, AutoML, recommender systems and advertising, sensor nets, etc.

Misspecified GP (kernel) bandits

- Realizability assumption may be too restrictive in real applications (kernel mismatch, hyperparameter estimation errors, etc.)

- Misspecified setting: Assumed hypothesis class is $\mathscr{F}_k(\mathscr{D}; B)$ but $f^\star \notin \mathscr{F}_k(\mathscr{D}; B)$

Misspecified GP (kernel) bandits

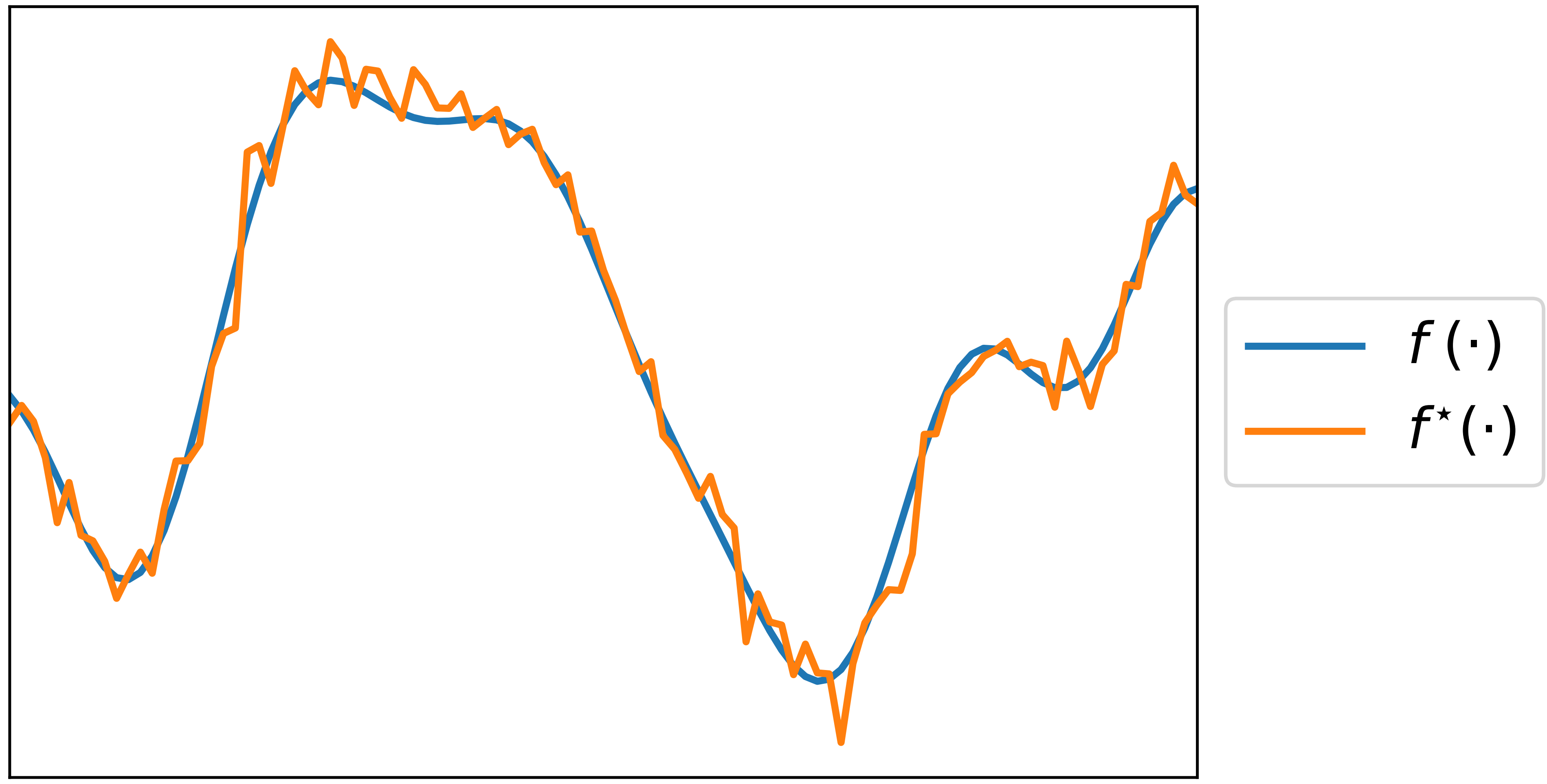
- Realizability assumption may be too restrictive in real applications (kernel mismatch, hyperparameter estimation errors, etc.)

- Misspecified setting: Assumed hypothesis class is $\mathscr{F}_k(\mathscr{D};B)$ but $f^\star \notin \mathscr{F}_k(\mathscr{D};B)$

- Assumption: The reward function can be uniformly approximated by a member from $\mathscr{F}_k(\mathscr{D};B)$

$$\min_{f \in \mathscr{F}_k(\mathscr{D};B)} \| f - f^\star \|_\infty \leq \epsilon$$

  - Misspecification rate $\epsilon > 0$ is unknown to the learner
  - Misspecification rate $\epsilon = 0$ recovers the realizable setting



Misspecified GP (kernel) bandits

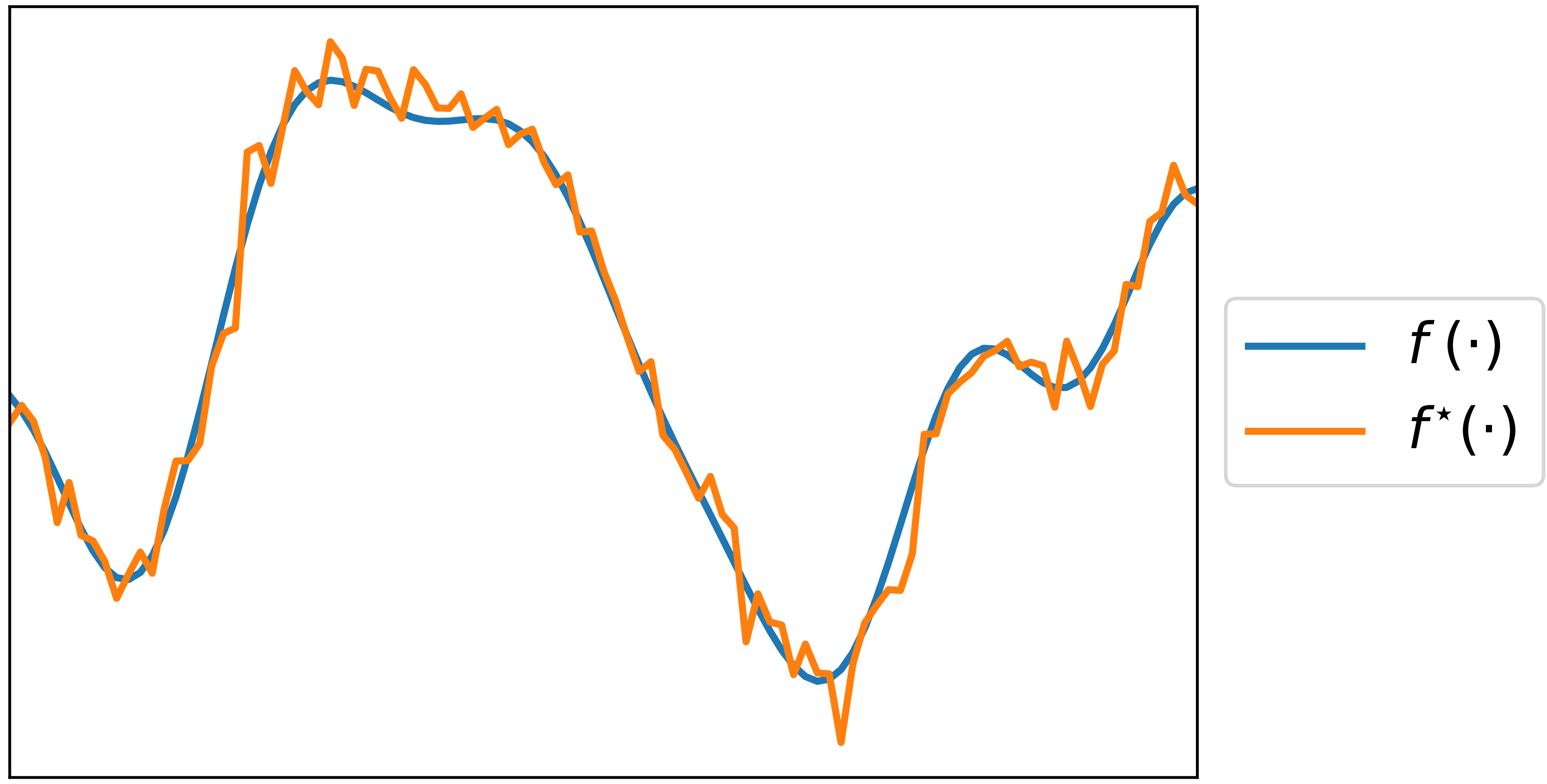
- Realizability assumption may be too restrictive in real applications (kernel mismatch, hyperparameter estimation errors, etc.)

- Misspecified setting: Assumed hypothesis class is $\mathscr{F}_k(\mathscr{D}; B)$ but $f^\star \notin \mathscr{F}_k(\mathscr{D}; B)$

- Assumption: The reward function can be uniformly approximated by a member from $\mathscr{F}_k(\mathscr{D}; B)$

$$\min_{f \in \mathscr{F}_k(\mathscr{D}; B)} \| f - f^\star \|_\infty \leq \epsilon$$

  - Misspecification rate $\epsilon > 0$ is unknown to the learner
  - Misspecification rate $\epsilon = 0$ recovers the realizable setting



Note: If $\| f - f' \|_k$ is small, then $\| f - f' \|_\infty$ is also small.

But the reverse does not need to hold !!

Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

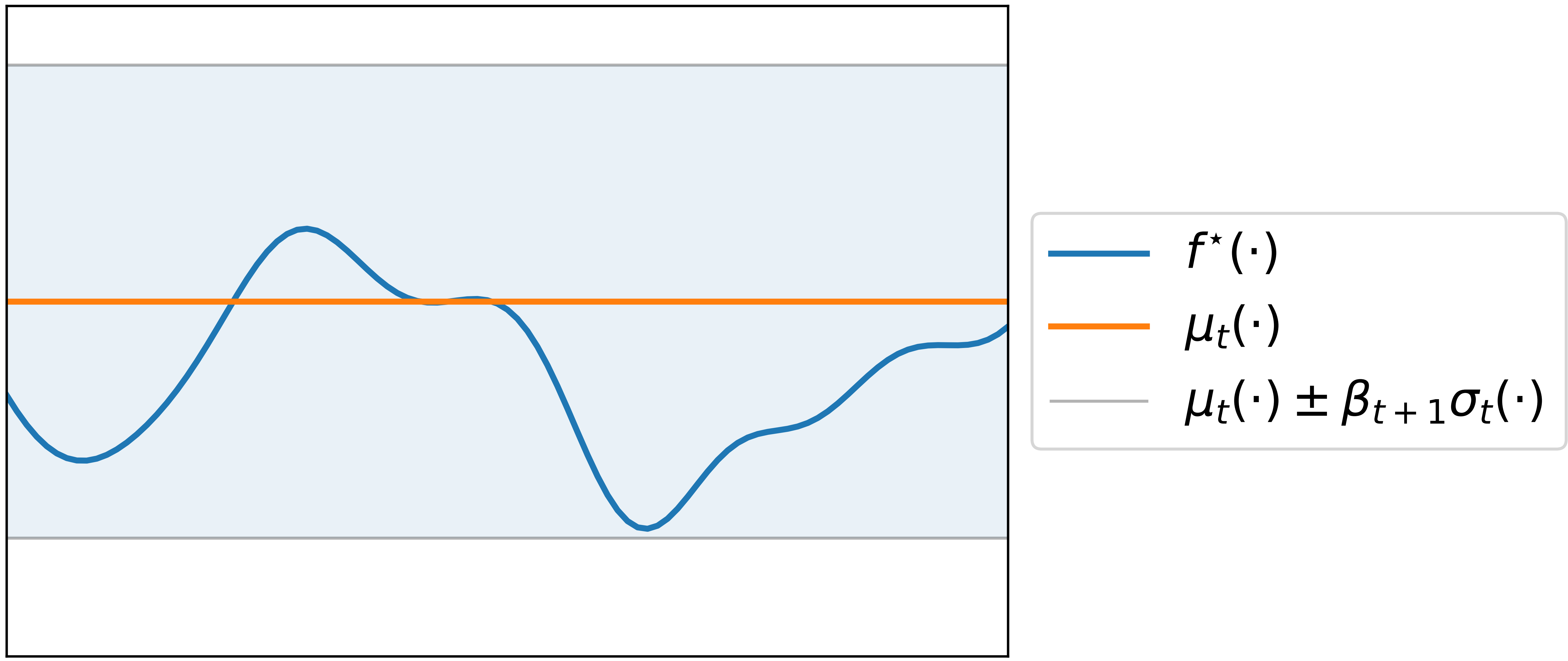
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\, \sigma_t(x) \;\le\; f(x) \;\le\; \mu_t(x) + \beta_{t+1}\, \sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t \in \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\, \sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]

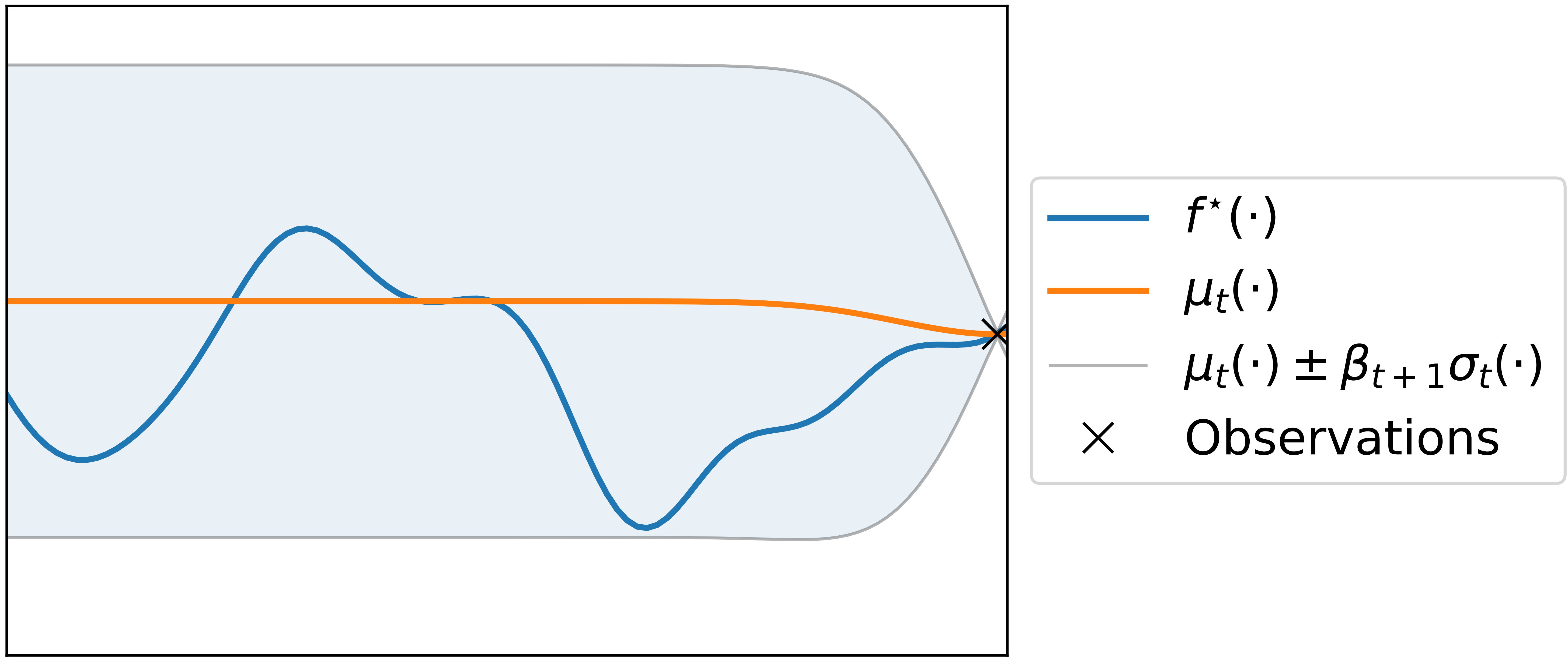


Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t \in \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$ [Srinivas *et al.*'10, ICML ToT Award]



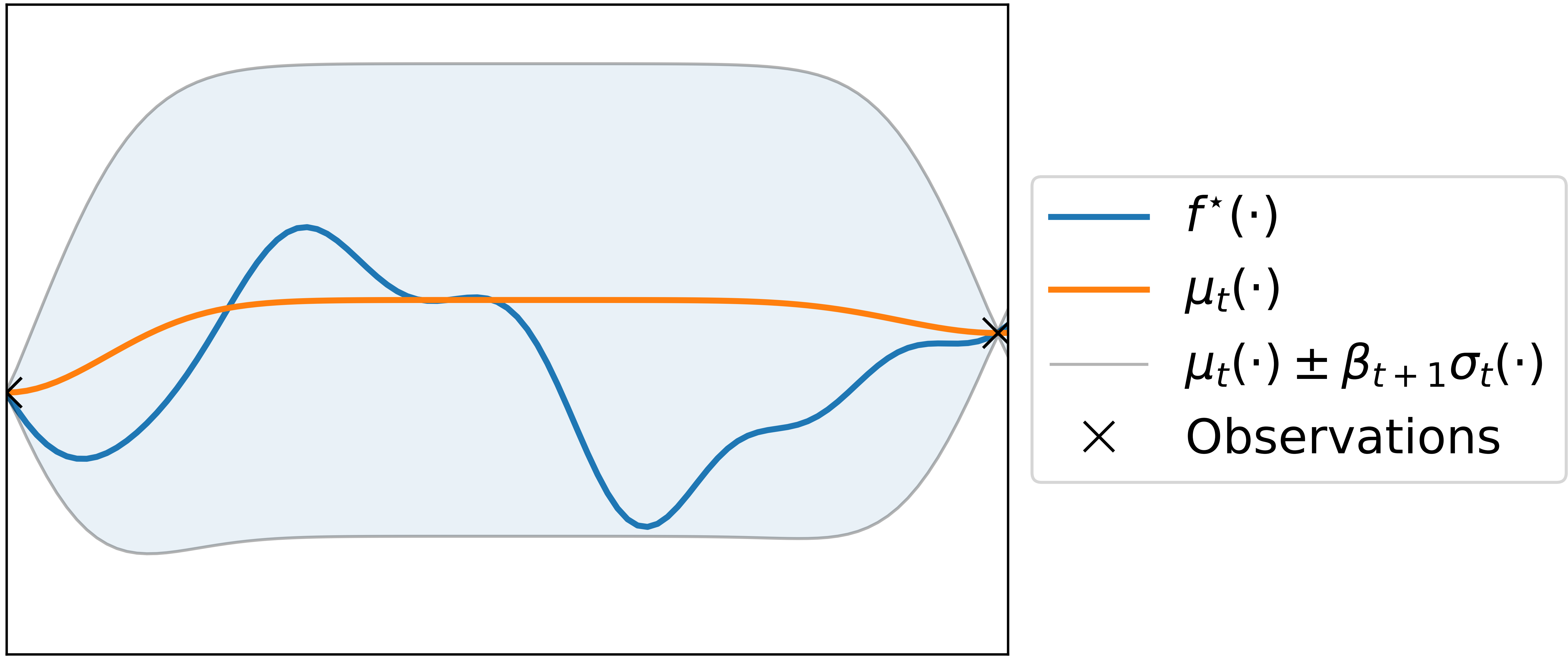
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



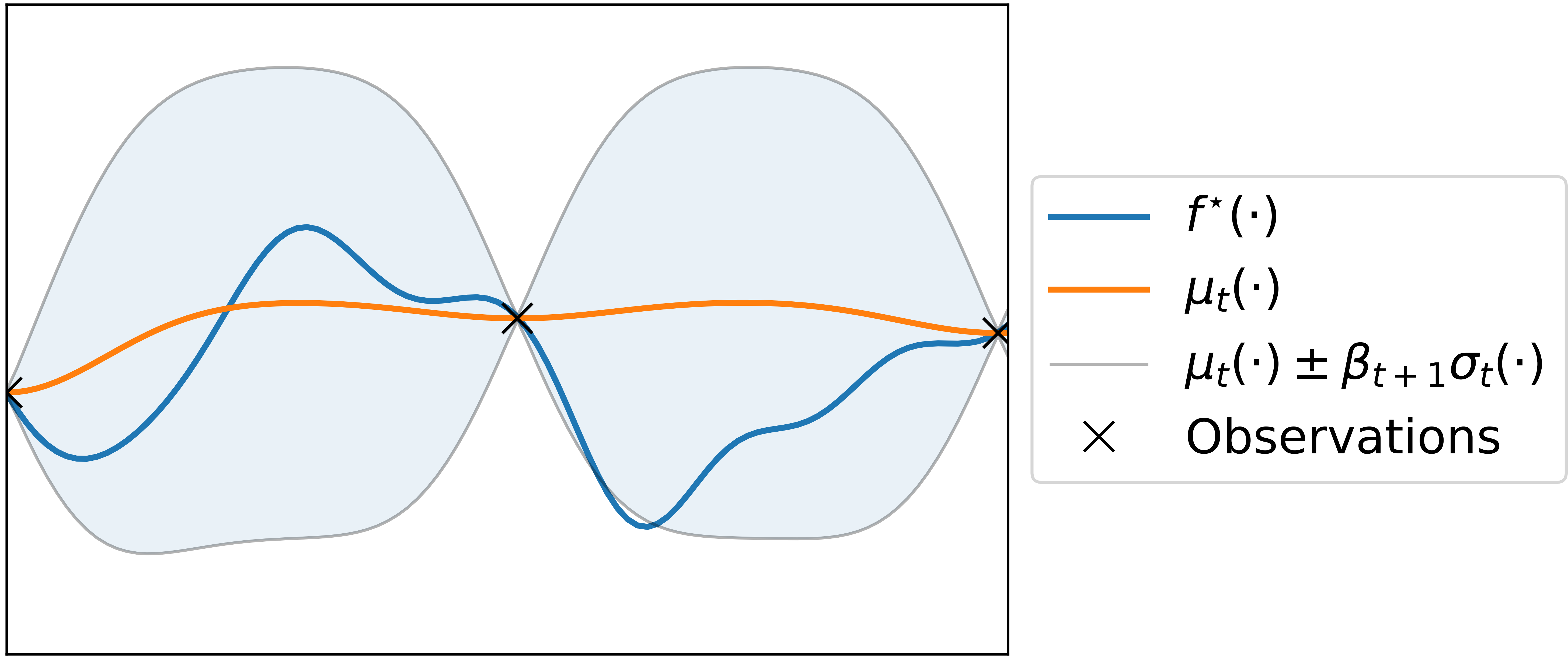
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



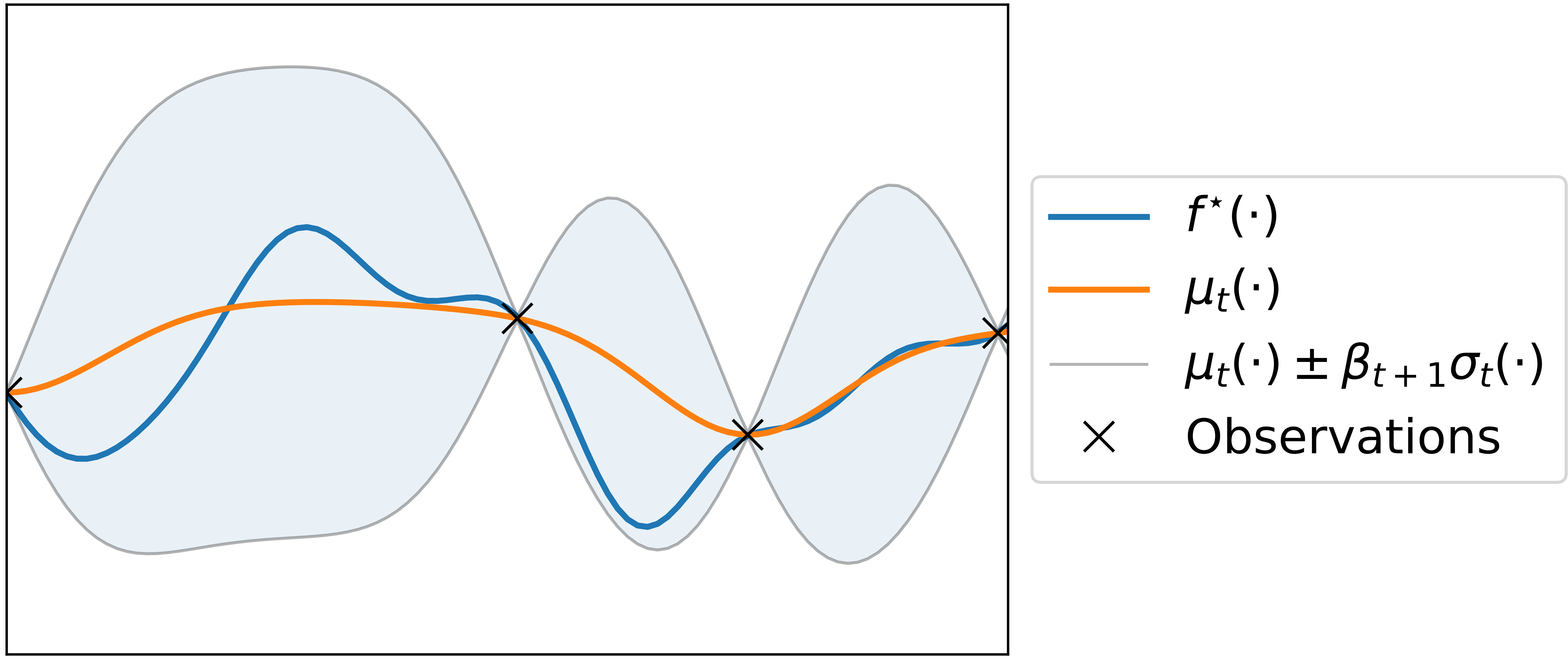
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



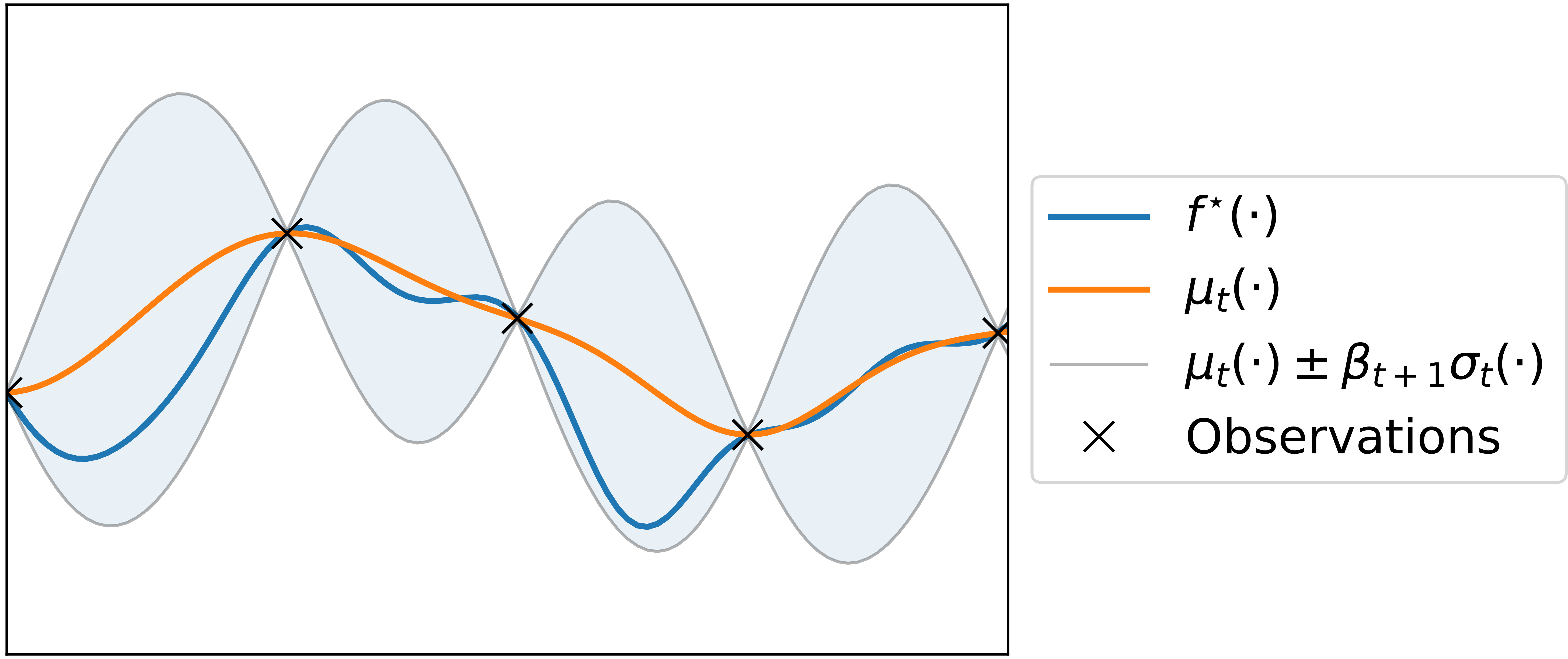
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \; + \; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



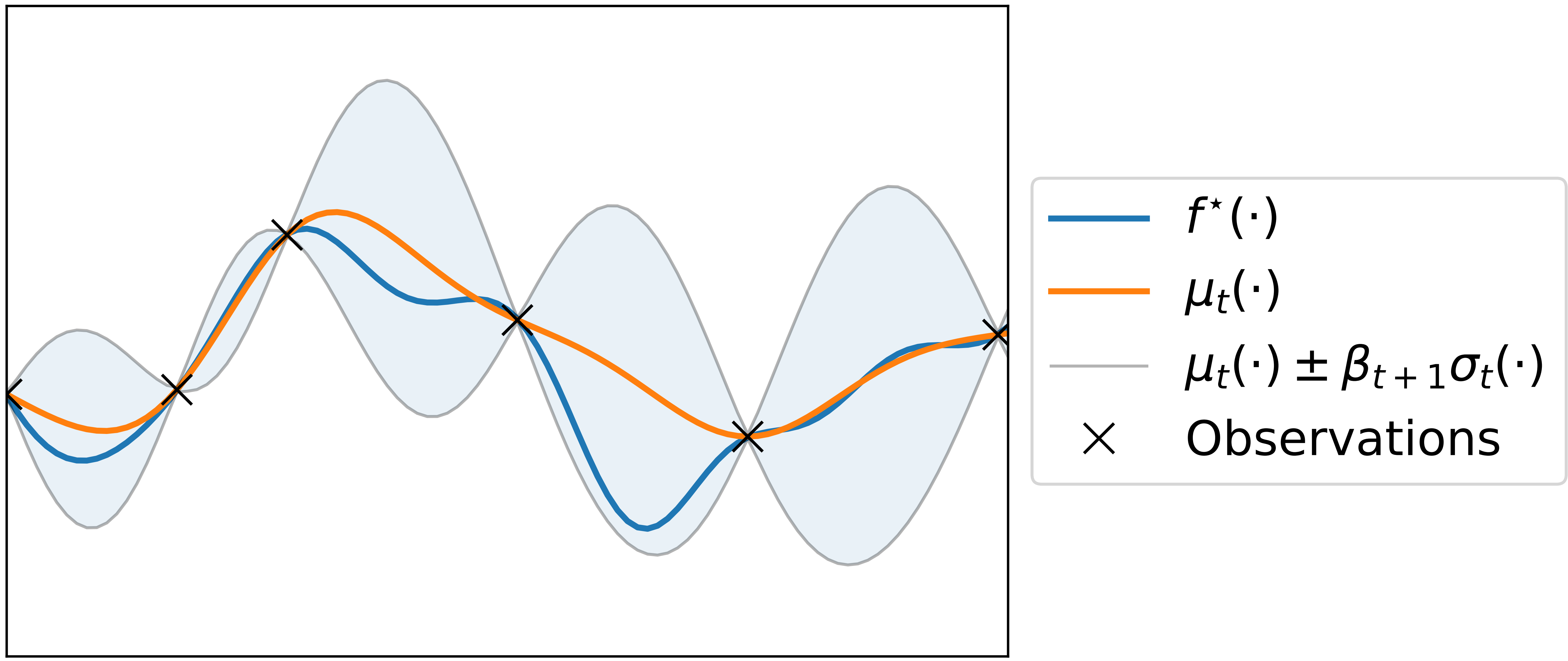
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



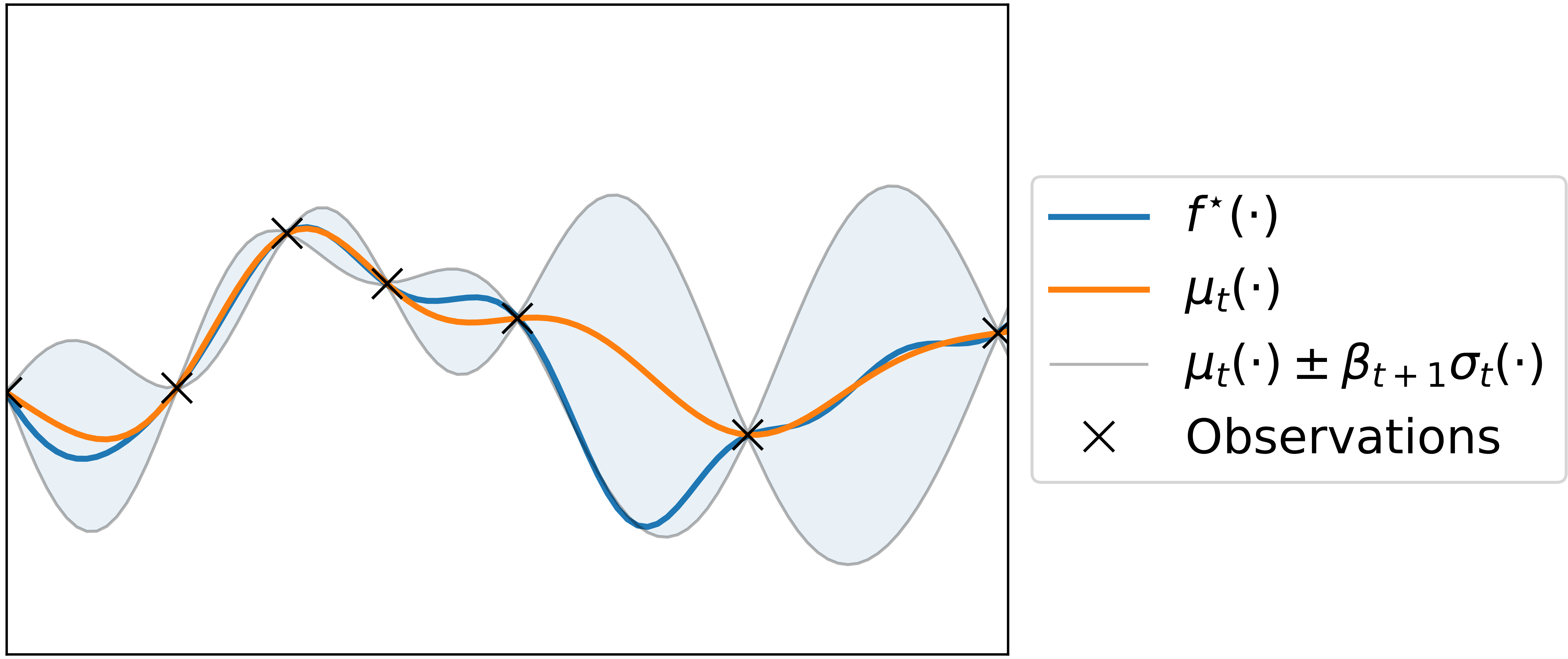
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



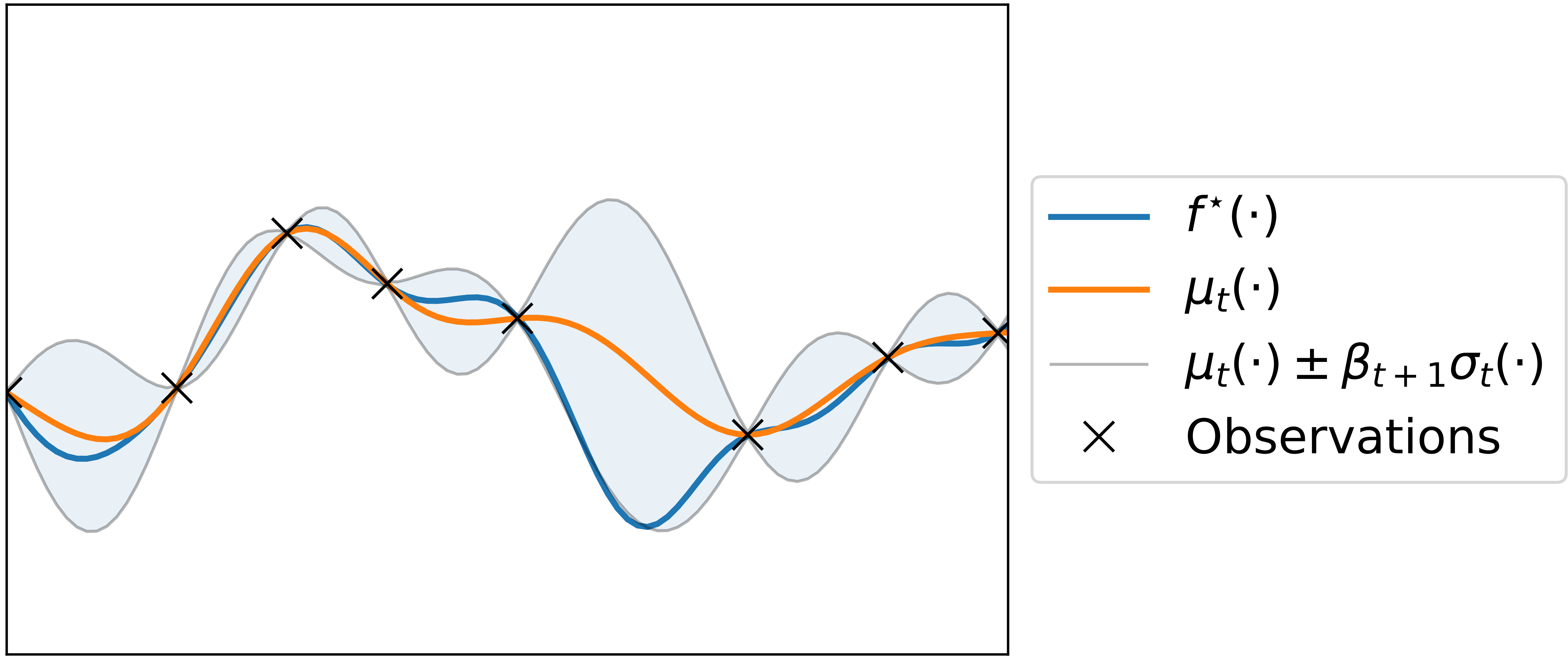
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



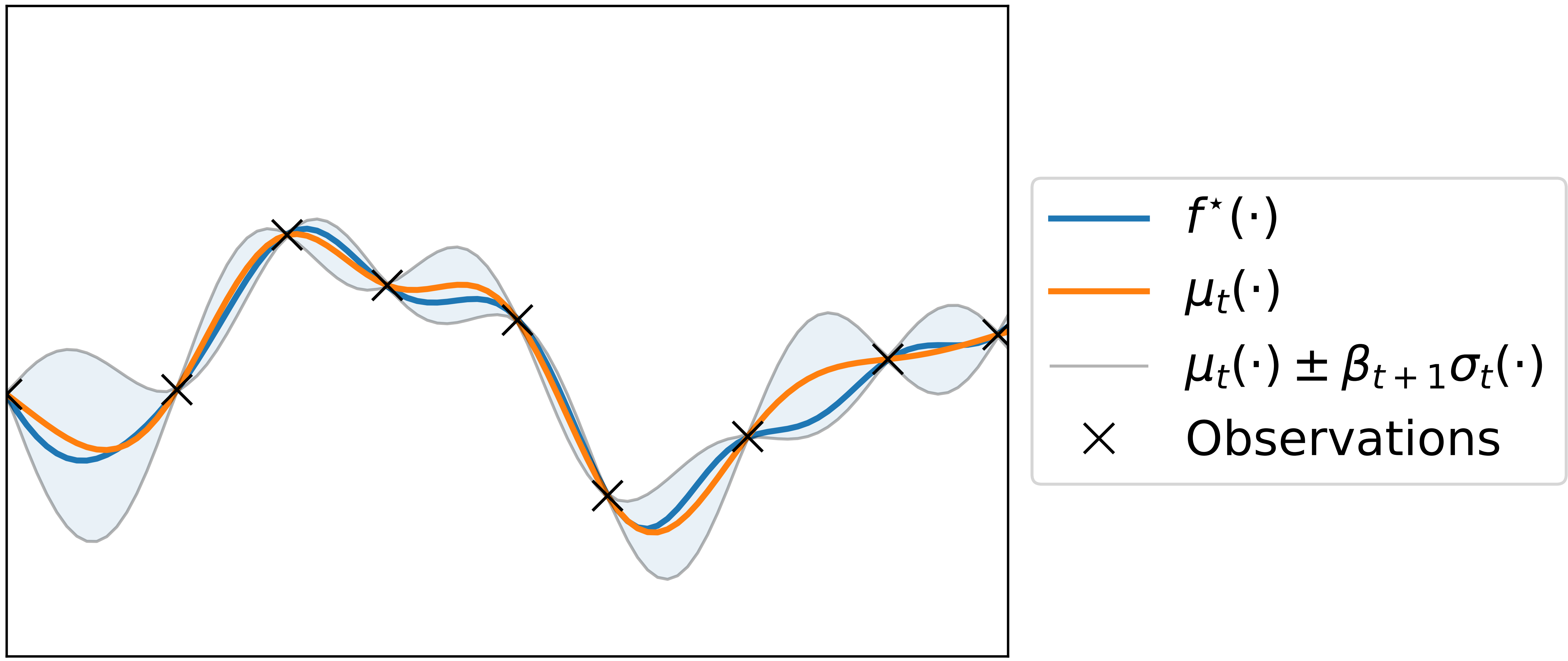
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



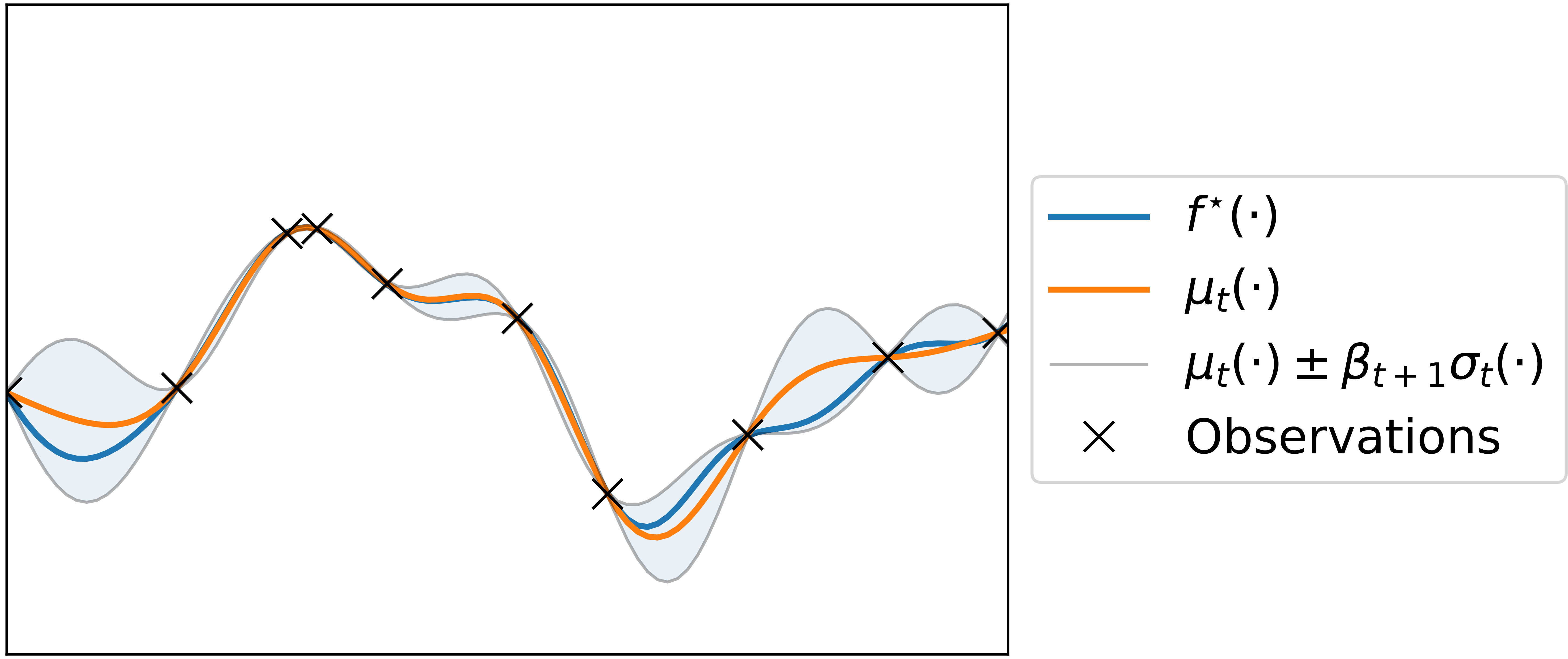
Background: Learning RKHS functions and optimistic approach

- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\le\; f(x) \;\le\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$
- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \; + \; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



Background: Learning RKHS functions and optimistic approach

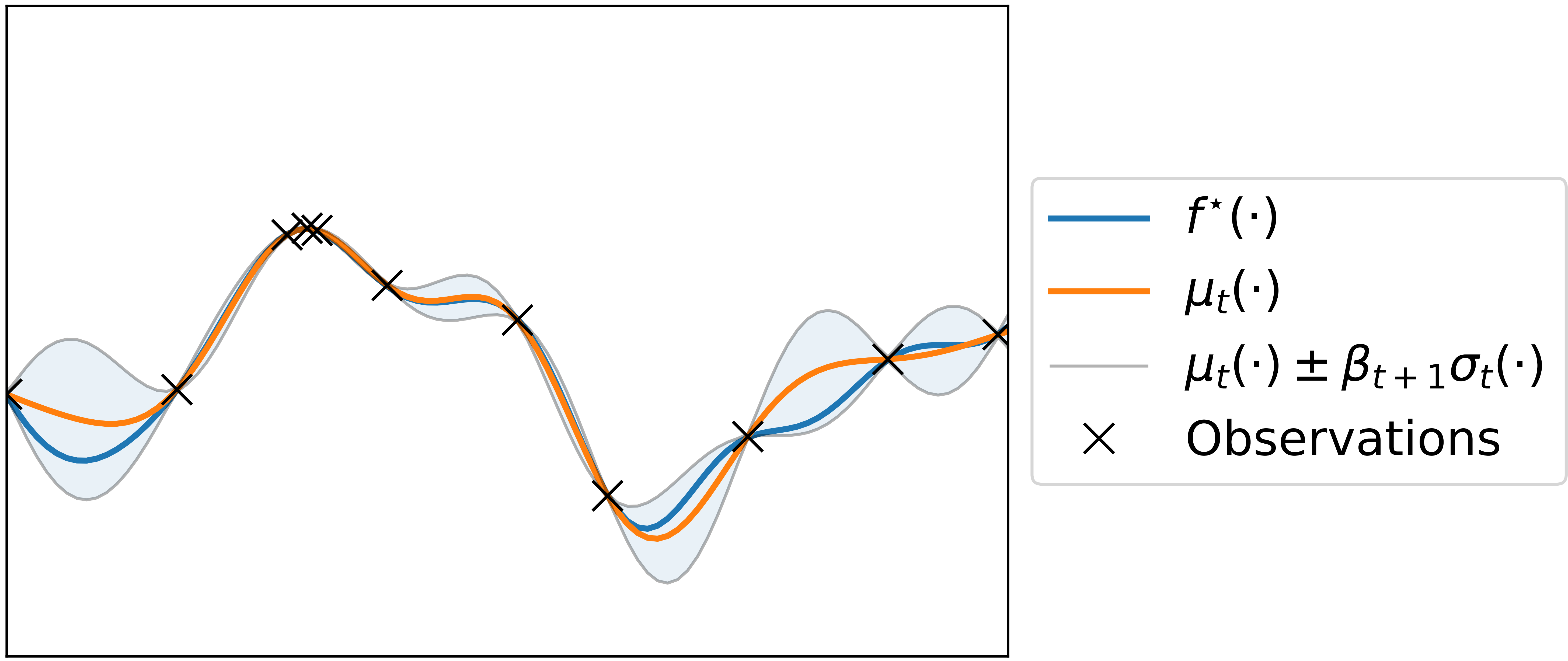
- Kernel ridge regression / Gaussian Process posterior to obtain function estimate $\mu_t(x)$
- Key idea: Use estimate $\mu_t(x)$ to guide exploration

[Srinivas *et al.*'10, Chowdhury *et al.*'17, Durand *et al.*'17]

- Construct confidence bounds such that w.h.p. $\mu_t(x) - \beta_{t+1}\,\sigma_t(x) \;\leq\; f(x) \;\leq\; \mu_t(x) + \beta_{t+1}\,\sigma_t(x) \qquad \forall x, t$

- GP-UCB algorithm maximizes UCB score: $x_t = \arg\max_{x \in D} \mu_t(x) \;+\; \beta_{t+1}\,\sigma_t(x)$

[Srinivas *et al.*'10, ICML ToT Award]



Known misspecification and optimistic approaches

- Misspecified mean estimator $\mu_t^\star(\cdot)$: Same as the one used in the realizable case => Misspecified (biased) model !!

Known misspecification and optimistic approaches

- Misspecified mean estimator $\mu_t^\star(\cdot)$: Same as the one used in the realizable case => Misspecified (biased) model !!
- Our first algorithm: Enlarged Confidence UCB algorithm (EC-UCB)

$$x_t \in \arg\max_{x \in D} \mu_t^\star(x) + \left(\beta_{t+1} + \frac{\epsilon\sqrt{t}}{\sqrt{\lambda}}\right) \sigma_t(x)$$

enlargement

Known misspecification and optimistic approaches

- Misspecified mean estimator $\mu_t^\star(\cdot)$: Same as the one used in the realizable case => Misspecified (biased) model !!

- Our first algorithm: Enlarged Confidence UCB algorithm (EC-UCB)

$$x_t \in \arg\max_{x \in D} \mu_t^\star(x) + \left(\beta_{t+1} + \frac{\epsilon\sqrt{t}}{\sqrt{\lambda}}\right) \sigma_t(x)$$

- Enlargement is due to model misspecification: For all x, t it holds

$$|\mu_t(\cdot) - \mu_t^\star(\cdot)| \leq \frac{\epsilon\sqrt{t}}{\sqrt{\lambda}}\sigma_t(\cdot)$$

  - $\mu_t(\cdot)$: Hypothetical mean estimator that corresponds to the (noisy) observations of the best in-class function, i.e., $f \in \arg\min_{f \in \mathcal{F}_k(\mathcal{D};B)} \| f - f^\star \|_\infty$
  - Intuition: Correlations among observations captured in the model increase the bias

Enlarged confidence UCB algorithm

- Recall assumptions:
  - Learner's hypothesis class $\mathcal{F}_k(\mathcal{D}; B)$; True reward function satisfies $\min_{f \in \mathcal{F}_k(\mathcal{D};B)} \| f - f^\star \|_\infty \leq \epsilon$

Theorem: EC-UCB achieves the following regret bound w.p. $1 - \delta$

$$R_T = \tilde{O}\Big(\underbrace{\big(B + \sqrt{\ln(1/\delta)}\big)\sqrt{\gamma_T T} + \gamma_T \sqrt{T}}_{\text{standard regret}} \quad + \quad T\epsilon\sqrt{\gamma_T} \Big).$$

due to misspecification
(unavoidable for any algorithm)

- T : time horizon / number of samples
- ϵ : misspecification parameter
- γ_T : kernel dependent mutual information quantity (measure of function class complexity)

  E.g., $\gamma_T = O\big((\log T)^{d+1}\big)$ for squared exponential kernel (Srinivas *et al.*'09)

Enlarged confidence UCB algorithm

- Recall assumptions:
  - Learner's hypothesis class $\mathcal{F}_k(\mathcal{D}; B)$; True reward function satisfies $\min_{f \in \mathcal{F}_k(\mathcal{D};B)} \| f - f^\star \|_\infty \leq \epsilon$

Theorem: EC-UCB achieves the following regret bound w.p. $1 - \delta$

$$R_T = \tilde{O}\Big(\underbrace{\big(B + \sqrt{\ln(1/\delta)}\big)\sqrt{\gamma_T T} + \gamma_T \sqrt{T}}_{\text{standard regret}} \quad + \quad T\epsilon\sqrt{\gamma_T} \Big).$$

due to misspecification
(unavoidable for any algorithm)

- T : time horizon / number of samples
- ϵ : misspecification parameter
- γ_T : kernel dependent mutual information quantity (measure of function class complexity)

  E.g., $\gamma_T = O\big((\log T)^{d+1}\big)$ for squared exponential kernel (Srinivas *et al.*'09)

- EC-UCB drawback: Requires knowing ϵ !!

Recall: $x_t = \arg\max_{x \in D} \mu_t^\star(x) \; + \; \left(\beta_{t+1} + \frac{\epsilon\sqrt{t}}{\sqrt{\lambda}}\right) \sigma_t(x)$

Main algorithm: Phased kernel uncertainty sampling

- Key idea: Use the uncertainty estimates to explore among “high-rewarding” actions
- Phased kernel uncertainty sampling: Runs in episodes of exponentially increasing length

Main algorithm: Phased kernel uncertainty sampling

- Key idea: Use the uncertainty estimates to explore among "high-rewarding" actions
- Phased kernel uncertainty sampling: Runs in episodes of exponentially increasing length

- For $t = 1, \dots, m_e$ do: %% m_e — episode length
  - Choose $x_t \in \arg\max_{x \in \mathcal{D}_e} \sigma^2_{t-1}(x)$
  - Update the uncertainty estimate $\sigma_t^2(\cdot)$ by using x_t

Main algorithm: Phased kernel uncertainty sampling

- Key idea: Use the uncertainty estimates to explore among "high-rewarding" actions
- Phased kernel uncertainty sampling: Runs in episodes of exponentially increasing length

- For $t = 1, \ldots, m_e$ do: %% m_e — episode length
  - Choose $x_t \in \arg\max_{x \in \mathcal{D}_e} \sigma_{t-1}^2(x)$
  - Update the uncertainty estimate $\sigma_t^2(\cdot)$ by using x_t
- Receive observations and compute $\mu_t^\star(\cdot)$
- Set $\mathcal{D}_{e+1} \leftarrow \left\{x \in \mathcal{D}_e \ : \ \mu_{m_e}^\star(x) + \beta_{m_e+1}\, \sigma_{m_e}(x) \ \geq \ \max_{x \in \mathcal{D}_e} \left(\mu_{m_e}^\star(x) - \beta_{m_e+1}\, \sigma_{m_e}(x)\right)\right\}$

Main algorithm: Phased kernel uncertainty sampling

- Key idea: Use the uncertainty estimates to explore among "high-rewarding" actions
- Phased kernel uncertainty sampling: Runs in episodes of exponentially increasing length

- For $t = 1, \ldots, m_e$ do: %% m_e — episode length
  - Choose $x_t \in \arg\max_{x \in \mathcal{D}_e} \sigma^2_{t-1}(x)$
  - Update the uncertainty estimate $\sigma^2_t(\cdot)$ by using x_t
- Receive observations and compute $\mu^\star_t(\cdot)$
- Set $\mathcal{D}_{e+1} \leftarrow \left\{x \in \mathcal{D}_e \;:\; \mu^\star_{m_e}(x) + \beta_{m_e+1}\,\sigma_{m_e}(x) \;\geq\; \max_{x \in \mathcal{D}_e}\left(\mu^\star_{m_e}(x) - \beta_{m_e+1}\,\sigma_{m_e}(x)\right)\right\}$
- Double the episode length, reset to $\sigma^2_0(\cdot)$, and return to the beginning

Main algorithm: Phased kernel uncertainty sampling

- Key idea: Use the uncertainty estimates to explore among "high-rewarding" actions
- Phased kernel uncertainty sampling: Runs in episodes of exponentially increasing length

> - For $t = 1, \ldots, m_e$ do: %% m_e — episode length
>   - Choose $x_t \in \arg\max_{x \in \mathcal{D}_e} \sigma^2_{t-1}(x)$
>   - Update the uncertainty estimate $\sigma_t^2(\cdot)$ by using x_t
> - Receive observations and compute $\mu_t^\star(\cdot)$
> - Set $\mathcal{D}_{e+1} \leftarrow \left\{x \in \mathcal{D}_e \;:\; \mu^\star_{m_e}(x) + \beta_{m_e+1}\,\sigma_{m_e}(x) \;\geq\; \max_{x \in \mathcal{D}_e} \left(\mu^\star_{m_e}(x) - \beta_{m_e+1}\,\sigma_{m_e}(x)\right)\right\}$
> - Double the episode length, reset to $\sigma_0^2(\cdot)$, and return to the beginning

- Note: $\mathcal{D}_e$ might not contain the global maximizer since the confidence bounds are invalid
- Phased elimination in the misspecified linear bandit (Lattimore *et al.*'20)

Phased kernel uncertainty sampling

Theorem: Phased uncertainty sampling achieves the following regret bound w.h.p.

$$R_T = \tilde{O}\Big(\underbrace{B\sqrt{\gamma_T T}}_{\text{standard regret}} \quad + \quad \underbrace{T\epsilon\sqrt{\gamma_T}}_{\text{due to misspecification}}\Big).$$

- Adapts to unknown misspecification parameter

Phased kernel uncertainty sampling

Theorem: Phased uncertainty sampling achieves the following regret bound w.h.p.

$$R_T = \tilde{O}\Big(\underbrace{B\sqrt{\gamma_T T}}_{\text{standard regret}} \quad + \quad \underbrace{T\epsilon\sqrt{\gamma_T}}_{\text{due to misspecification}}\Big) \, .$$

- Adapts to unknown misspecification parameter
- Recovers the linear misspecified setting and results (Lattimore *et al.*'20) when used with the linear kernel
- Contemporary work of Camilleri *et al.*'21 obtains similar results via a very different algorithm

Phased kernel uncertainty sampling

Theorem: Phased uncertainty sampling achieves the following regret bound w.h.p.

$$R_T = \tilde{O}\Big(\underbrace{B\sqrt{\gamma_T T}}_{\text{standard regret}} \quad + \quad \underbrace{T\epsilon\sqrt{\gamma_T}}_{\text{due to misspecification}}\Big).$$

- Adapts to unknown misspecification parameter
- Recovers the linear misspecified setting and results (Lattimore *et al.*'20) when used with the linear kernel
- Contemporary work of Camilleri *et al.*'21 obtains similar results via a very different algorithm
- Misspecified kernelized contextual setting (in the paper)
  - Based on the regret bound balancing strategy of Pacchiano *et al.*'20

Summary

Goal:

- Protect against model misspecification
  - Bandits (Ghosh *et al.*'17, Zanette *et al.*'20, Lattimore *et al.*'20, Neu *et al.*'20, Foster *et al.*'20), Contextual bandits (Foster *et al.*'20, Pacchiano *et al.*'20), RL (Jin *et al.*'20, Du *et al.*'19)
- Strong model assumptions are restrictive in practice

Our contributions:

- Complete treatment of the misspecified GP bandit optimization problem
- Practical algorithms inspired by the classical Bayesian optimization and ED acquisition functions
- Theoretical regret bounds in multiple settings

Paper ID: 28196