ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf Informatik Computer Science

Bayesian optimization and Gaussian process bandits: Theory and Applications

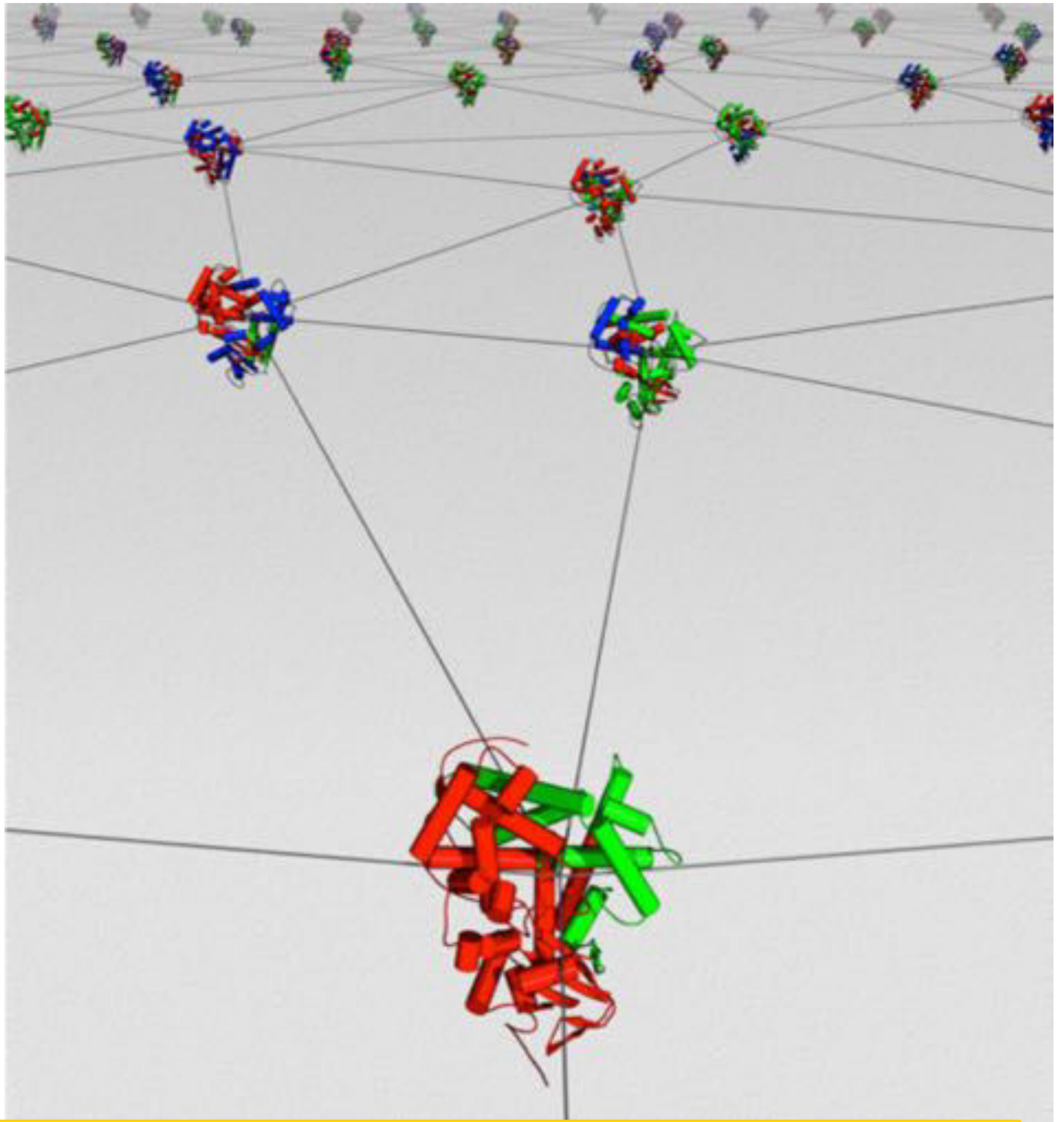
Andreas Krause

Summer school on Mathematics of Deep Learning
Berlin

Navigating the Protein Fitness Landscape

[Romero, K, Arnold PNAS '13]

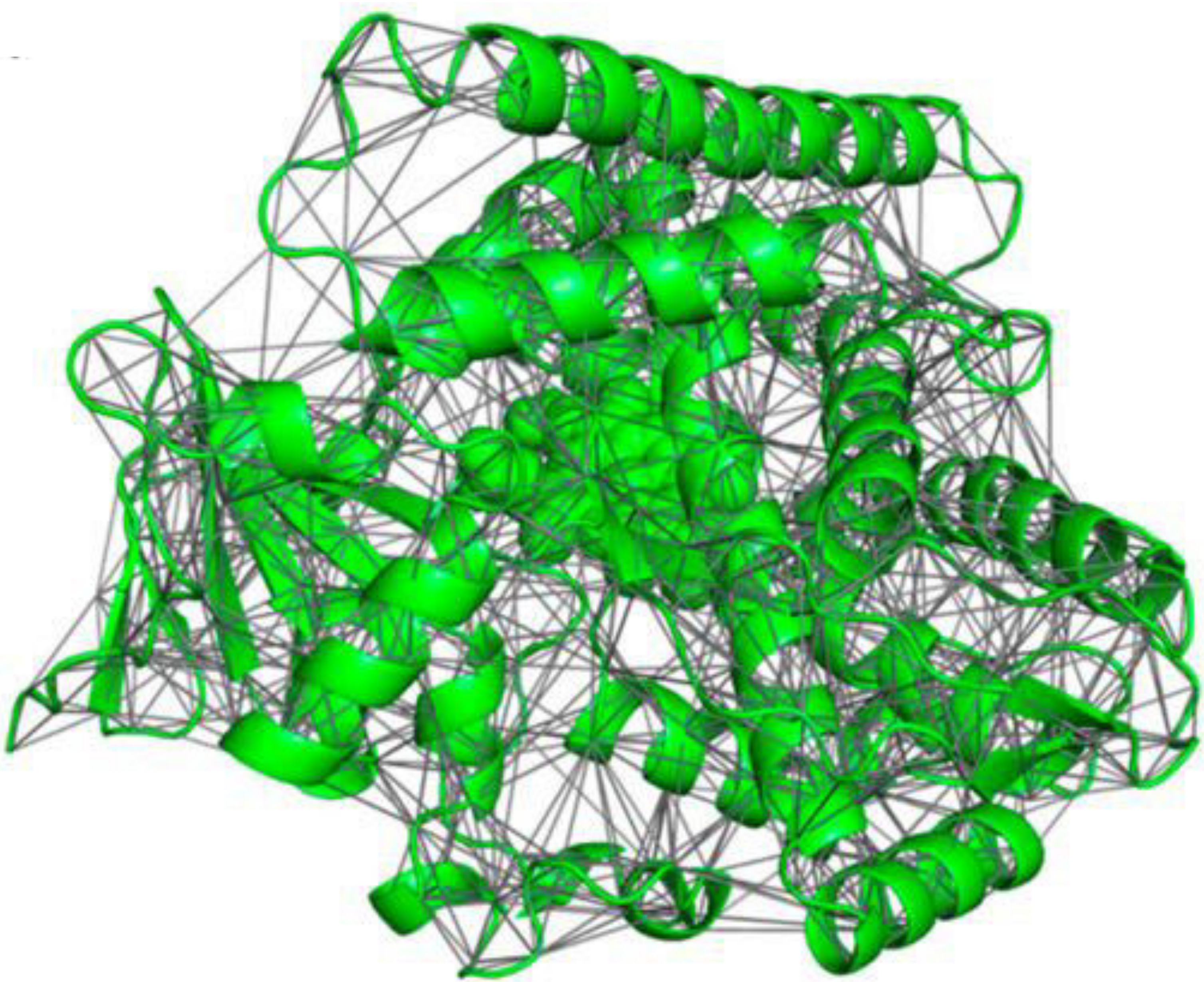
- Want to engineer proteins with desirable properties
  - Vaccine design
  - Contrast agents
  - ...
- Need experiments!
- **Sequence space is vast**



How can we design experiments to find good sequences?

Designing P450s chimeras

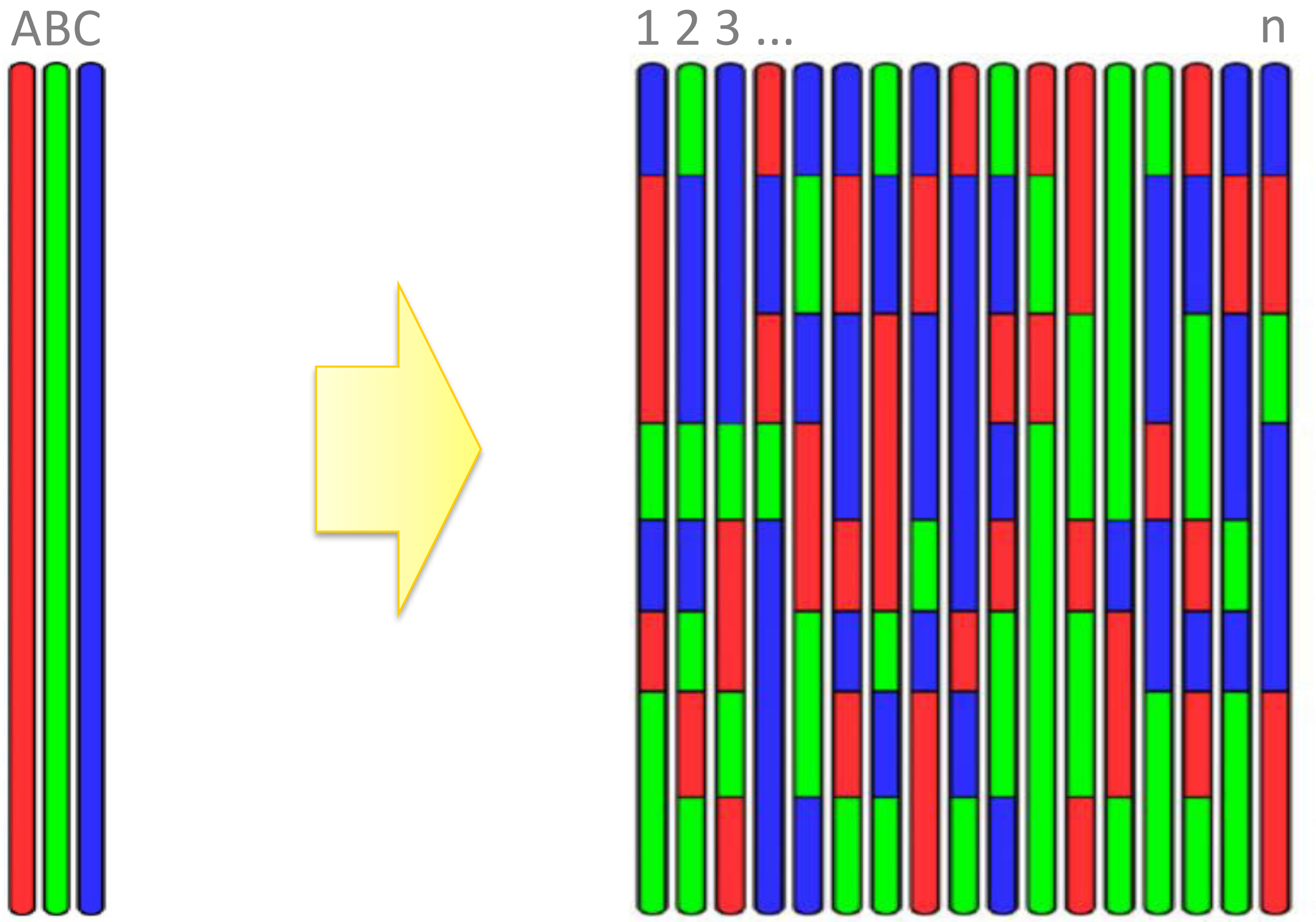
[Romero, K, Arnold PNAS ‘13]



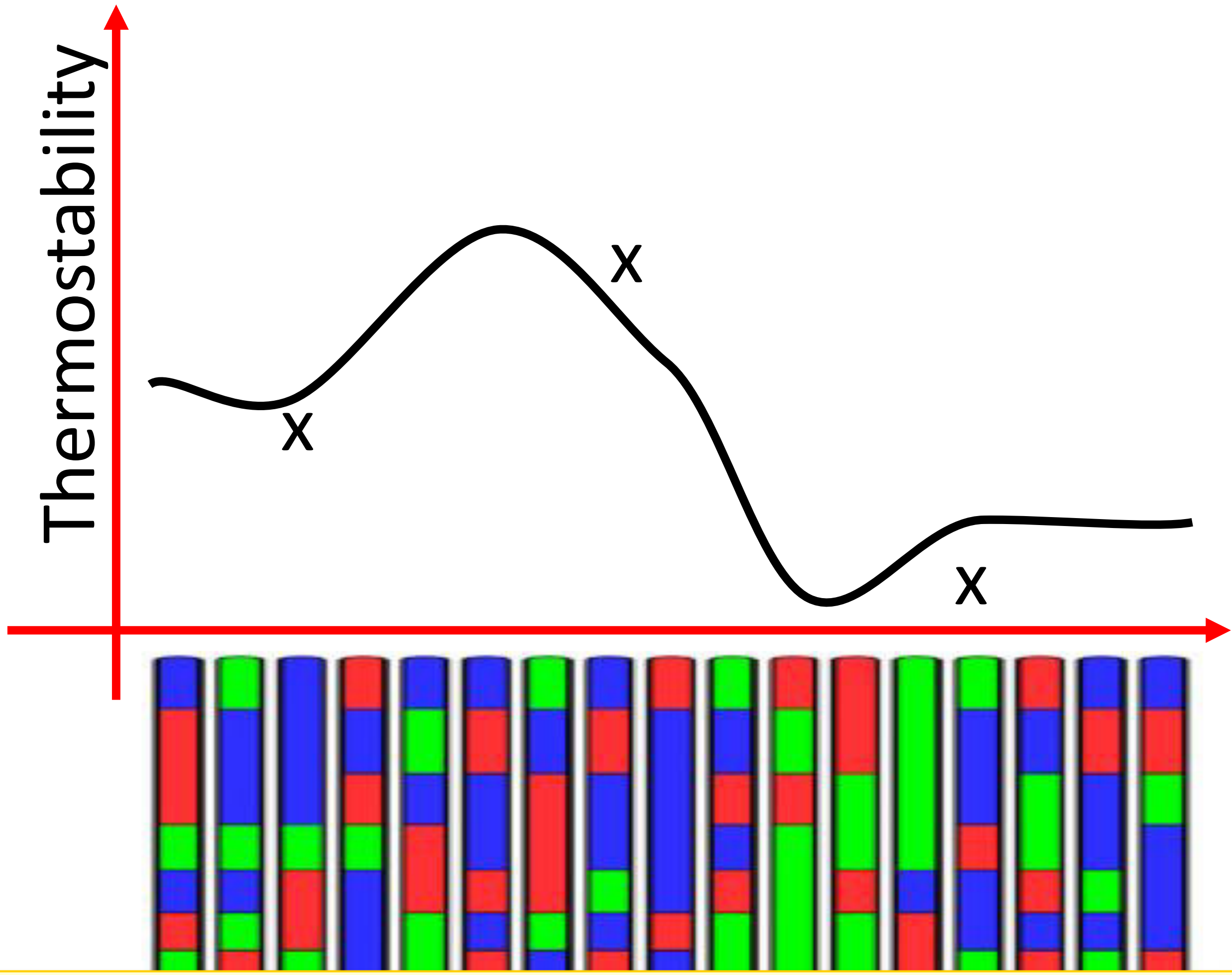
Design space

Parent sequences

Candidate designs



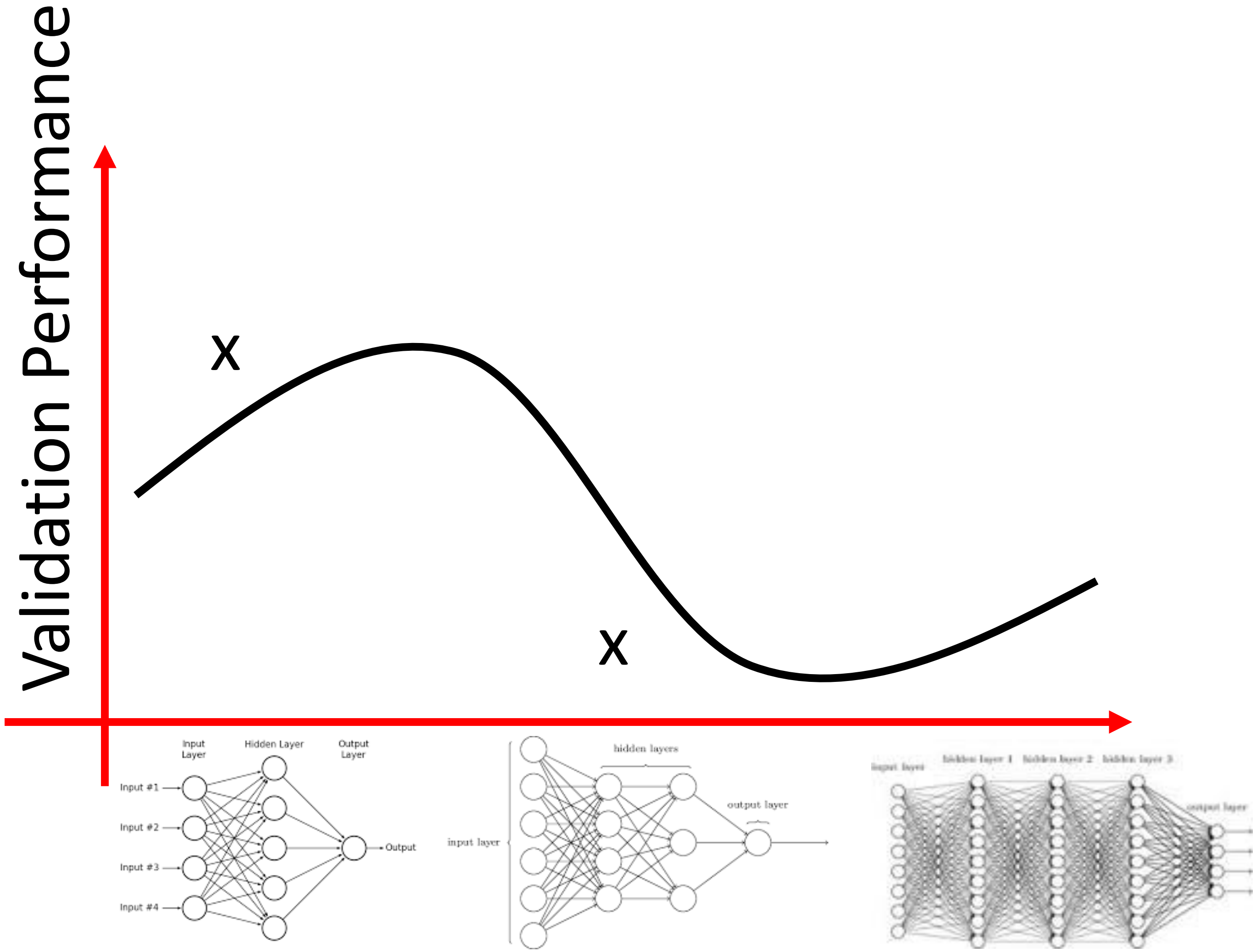
Protein Fitness Landscape



How can we experiment to learn and optimize thermostability?

Automatic Machine Learning

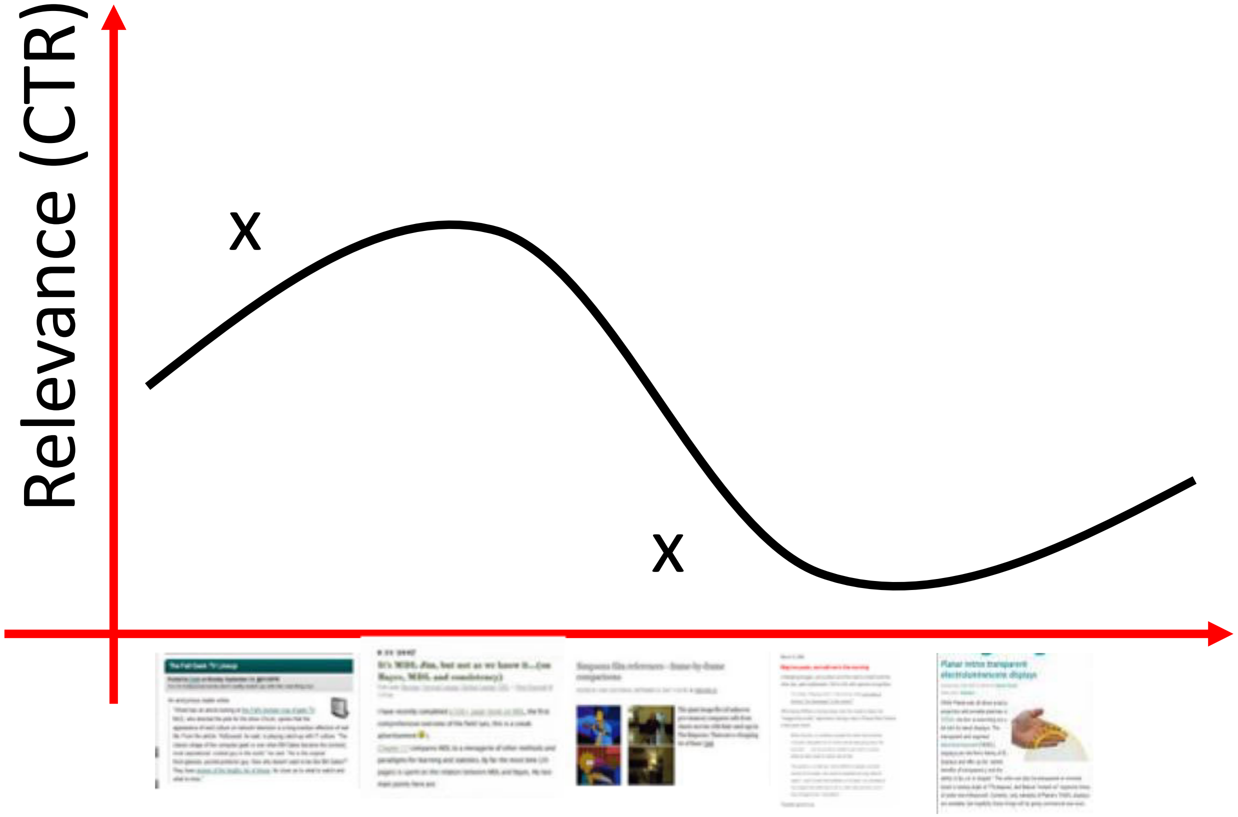
[Cf. Snoek et al'12; Google Vizier, Golovin et al '17]



How can we automatically tune model & hyperparameters?

Explore-exploit in Recommendation

[cf Li et al'10, Vanchinathan et al '14]



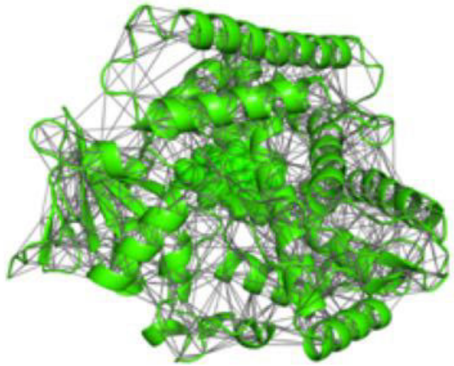
Economics Politics Sport

How can we recommend to learn and optimize relevance?

Exploration—Exploitation Tradeoffs

Numerous applications require trading experimentation (exploration) and optimization (exploitation)

- Experimental design
- Recommender systems
- Online advertising
- Automatic ML
- Robotic control



Often:

- #alternatives >> #trials
- experiments are noisy & expensive
- similar alternatives have similar performance

Can one exploit this regularity?

Outline

- Motivating Examples and Problem Setting
- Review of Gaussian processes
- GP Bandits and Bayesian optimization
- More complex settings
  - Parallelization
  - Multi-task / contextual optimization
  - Level sets
  - Multi-objective optimization
  - High dimensions
  - Constraints and "Safe" Bayesian optimization

k-armed (stochastic) bandits

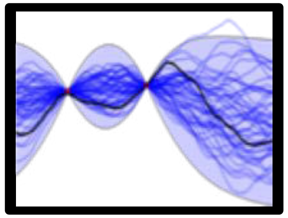
f_1 f_2 f_3 … f_k

- Sequentially allocate T tokens to k "arms" of a slot machine
- Each time: pick arm i; get iid payoff with (unknown) mean f_i
- Want to maximize the expected cumulative reward
- Classical model of exploration – exploitation tradeoff
  - Has been extensively studied (since Robbins '52)
  - In same cases, can calculate *optimal* allocation (Gittins '79)
  - Tight bounds on cumulative regret (Auer et al '02, …)
  - Very successful in applications (e.g., drug trials, scheduling, …)
- Typically assume every "arm" is tried multiple times

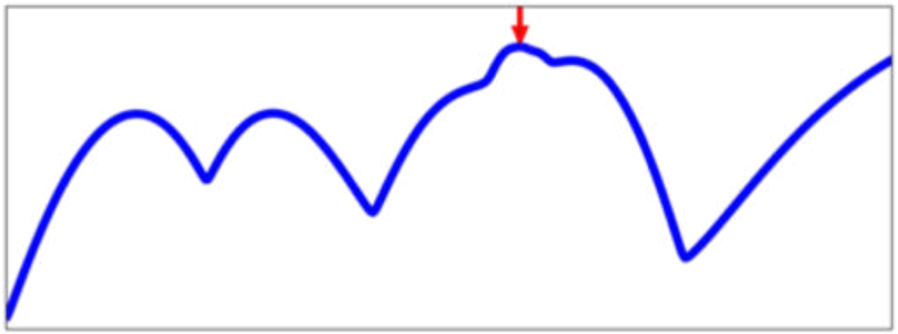
∞-armed bandits

- In many domains, number of choices is very large
  - Space of parameters for possible lab experiments or NN architectures
  - Recommender systems
  - Policy parameters for robotic control
- Can't even try every choice once!
- Classical algorithms don't scale, and guarantees become useless
- Substantial work on "structured" bandits (linear, Lipschitz, combinatorial, networked, etc.)

Another viewpoint:
Bayesian Optimization [Močkus ’75]

$x_t \longrightarrow$  $\longrightarrow \quad y_t = f(x_t) + \epsilon_t$

Acquisition function



Expected/most prob. improvement [Močkus *et al.* ’78,’89], Information gain about maximum [Villemonteix *et al.* ’09], Knowledge gradient [Powell *et al.* ’10], Predictive Entropy Search [Hernández-Lobato *et al.* ’14], TruVaR [Bogunovic et al’17], Max Value Entropy Search [Wang et al’17]

Bandits vs Bayesian optimization

(Stochastic) Bandits

- **Finite** [Robbins '52, Gittins '79, Auer et al '02...]
  Linear objectives [Dani *et al.* '08; Rusmevichientong & Tsitsiklis '08],
  Lipschitz objectives [Slivkins *et al.* '08, Bubeck *et al.* '08], ...

- Strong theory
- Not as „flexible"
- (Often) Frequentist
- Contextual, dueling, ...

Bayesian optimization

- Sample **Bayesian (GP) model** of *f* acc. to
  Expected Improvement [Močkus *et al.* '78],
  Most Probable Improvement [Močkus '89],
  Information gain about maximum [Villemonteix *et al.* '09],
  Knowledge gradient [Powell *et al.* '10],...

- Little theory
- Highly configurable
- Bayesian
- Parallel, multi-fidelity, ...

Combine insights to get best of both worlds

Learning to optimize

- Given: Set of possible inputs *D*; noisy black-box access to unknown function $f \in \mathcal{F}, \quad f : D \to \mathbb{R}$
- Task: Choose inputs $x_1, \ldots, x_T$ from *D*
  After each selection, observe $y_t = f(x_t) + \varepsilon_t$

Cumulative regret: $R_T = \sum_{t=1}^{T} \Big(\max_x f(x) - f(x_t) \Big)$

Sublinear if $R_T/T \to 0$

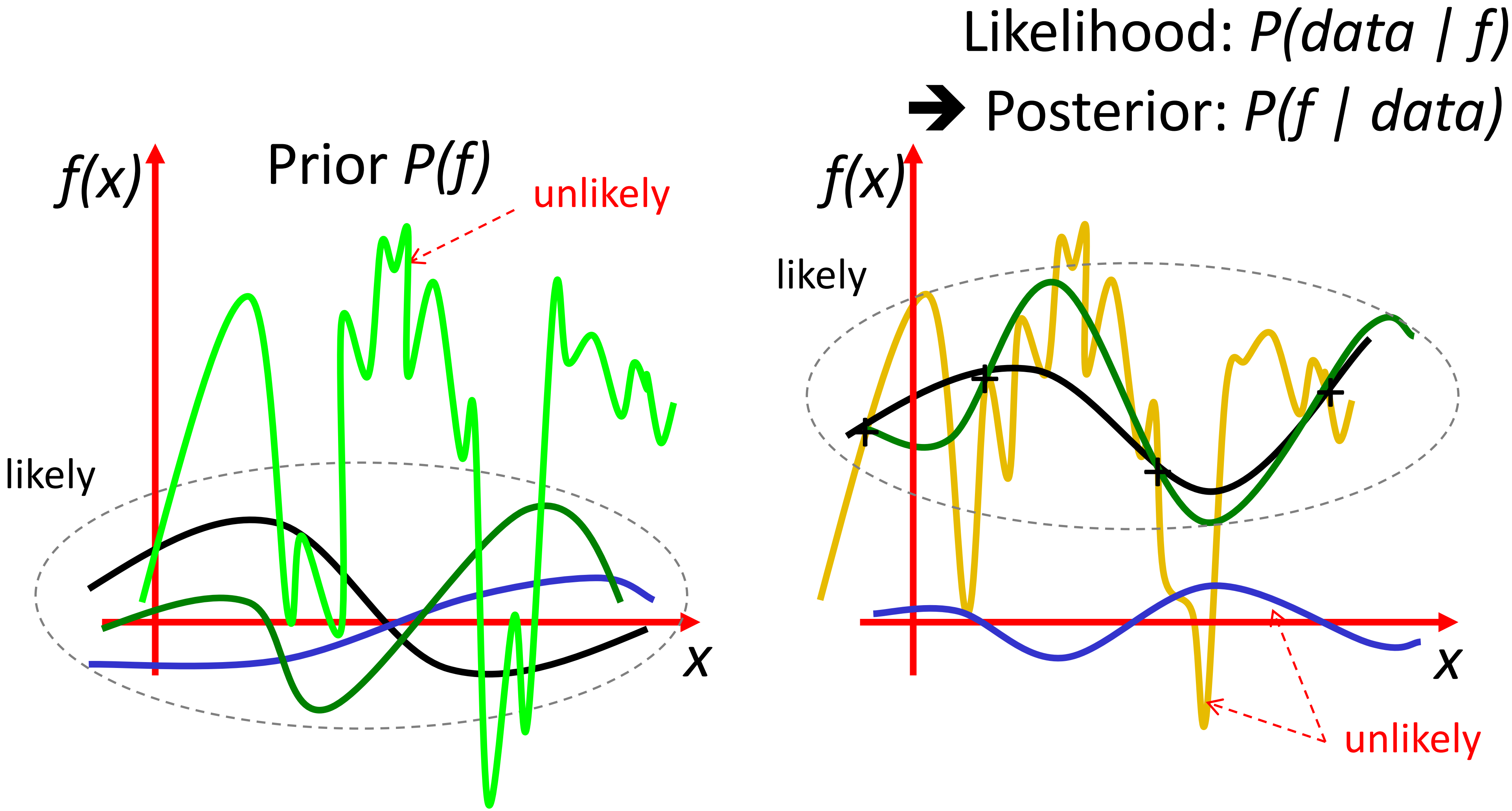
Simple regret: $S_T = \min_{t \in \{1,\ldots T\}} \Big(\max_x f(x) - f(x_t) \Big)$

Note that $S_T \leq R_T/T$

Brief review of Gaussian Processes

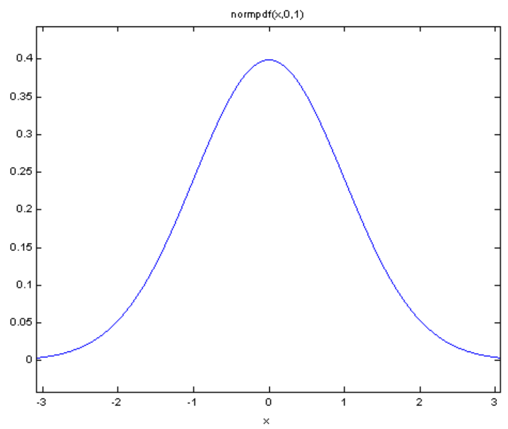
Gaussian processes

[c.f. Rasmussen & Williams 2006]

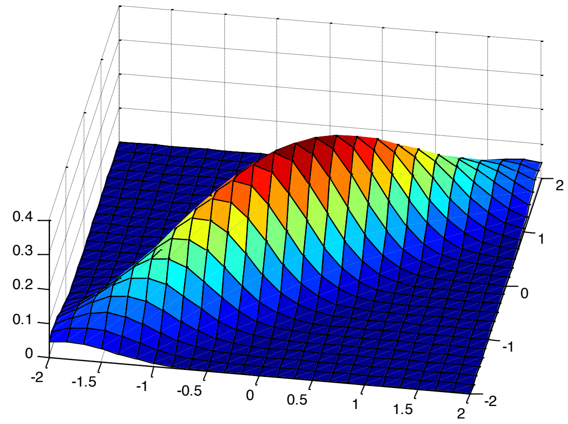


Predictive uncertainty + tractable inference

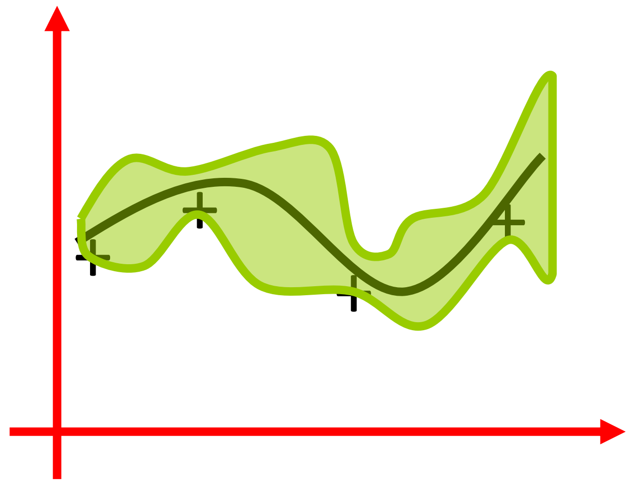
Gaussian Processes



Normal dist.
(1-D Gaussian)



Multivariate normal
(n-D Gaussian)



Gaussian process
(∞-D Gaussian)

- Gaussian process (GP) = normal distribution over *functions*
- Finite marginals are multivariate Gaussians
- Closed form formulae for Bayesian posterior update exist
- Parameterized by *covariance function* $K(x,x') = Cov(f(x),f(x'))$

Gaussian process

A Gaussian Process (GP) is an

(infinite) set of random variables, indexed by some set X

i.e., for each x in X there's a random variable Y_x

There exists functions $\mu : X \to \mathbb{R}$ $\mathcal{K} : X \times X \to \mathbb{R}$

such that for all $A \subseteq X$, $A = \{x_1, \ldots, x_k\}$

it holds that

$$Y_A = [Y_{x_1}, \ldots, Y_{x_k}] \sim \mathcal{N}(\mu_A, \Sigma_{AA})$$

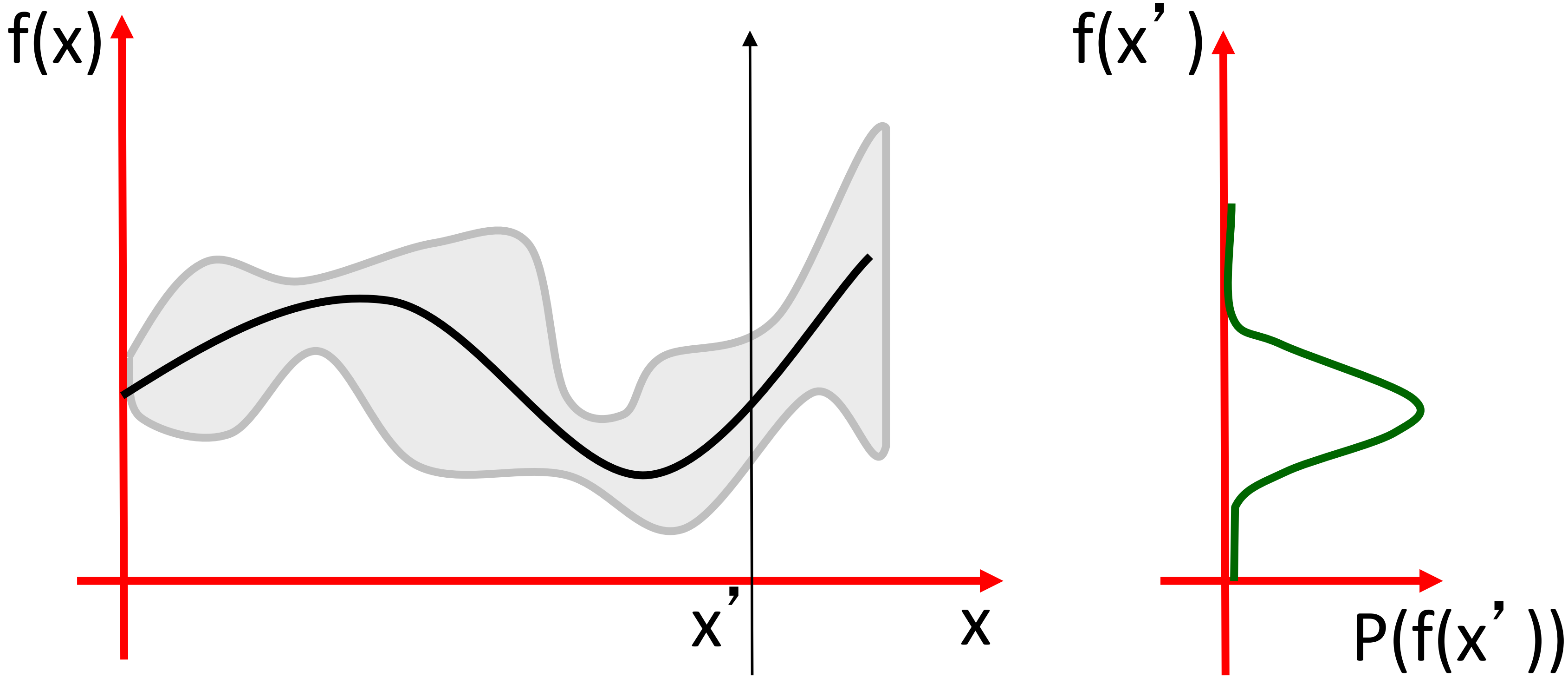
where

$$\Sigma_{AA} = \begin{pmatrix} \mathcal{K}(x_1, x_1) & \mathcal{K}(x_1, x_2) & \ldots & \mathcal{K}(x_1, x_k) \\ \vdots & & & \vdots \\ \mathcal{K}(x_k, x_1) & \mathcal{K}(x_k, x_2) & \ldots & \mathcal{K}(x_k, x_k) \end{pmatrix} \quad \mu_A = \begin{pmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_k) \end{pmatrix}$$

K is called kernel (covariance) function

μ is called mean function

Predictive confidence in GPs



Typically, only care about marginals, i.e.,

$$P(f(x)) = \mathcal{N}(f(x); \mu_t(x), \sigma_t^2(x))$$

Parameterized by covariance function *K(x,x ') = Cov(f(x),f(x '))*

Kernel functions

- K must be symmetric

  $K(x,x') = K(x',x)$ for all x, x'

- K must be positive definite

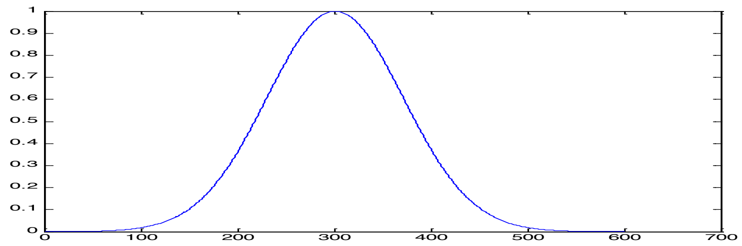
  For all A: Σ_{AA} is positive definite matrix

- Kernel function K: assumptions about correlation!

- Decades of research in ML on kernels for different data types (vectors, graphs, sets, sequences, …)

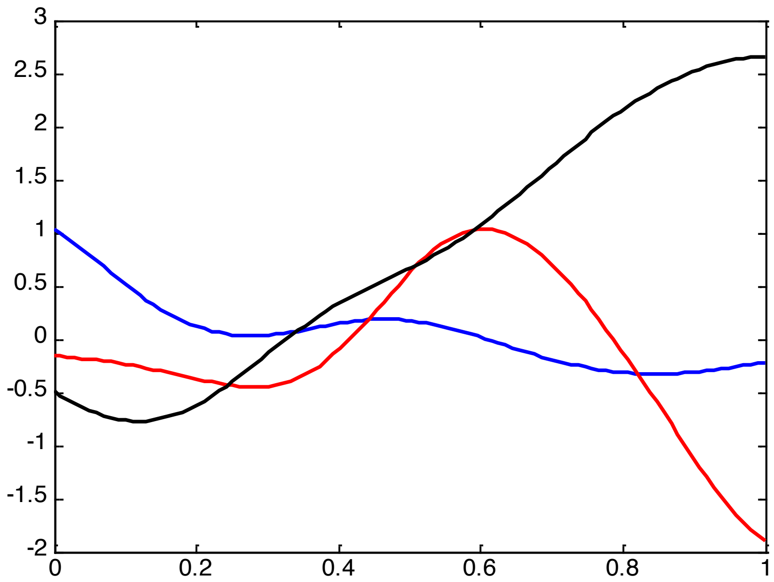
Kernel functions: Examples

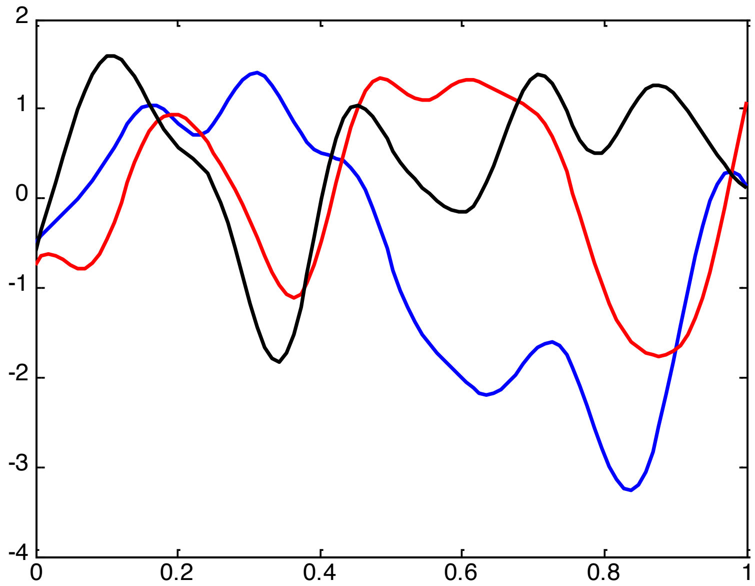
- Squared exponential kernel
  $K(x,x') = \exp(-(x-x')^2/h^2)$



Distance $|x-x'|$

Samples from P(f)



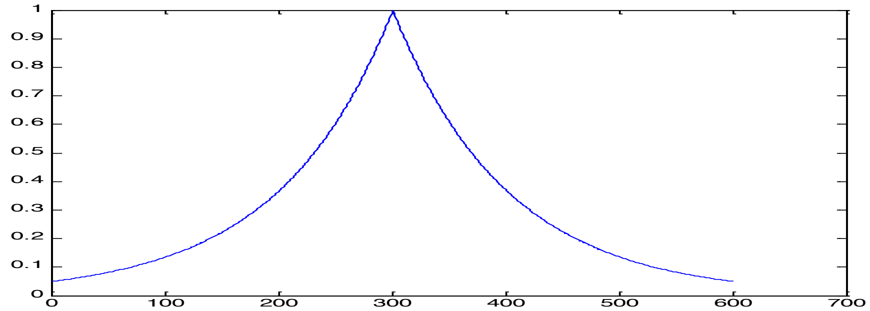


Bandwidth h=.3

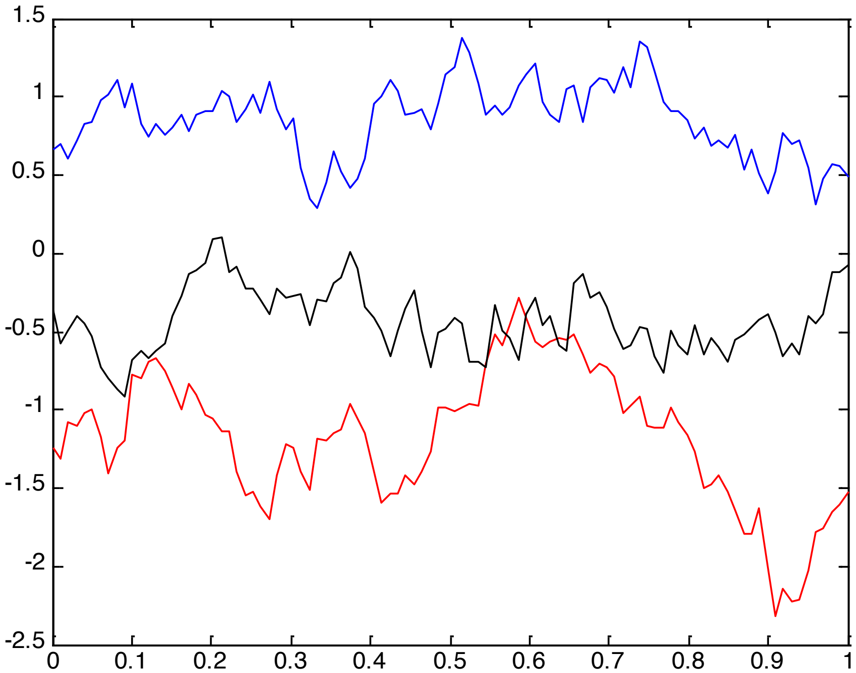
Bandwidth h=.1

Kernel functions: Examples

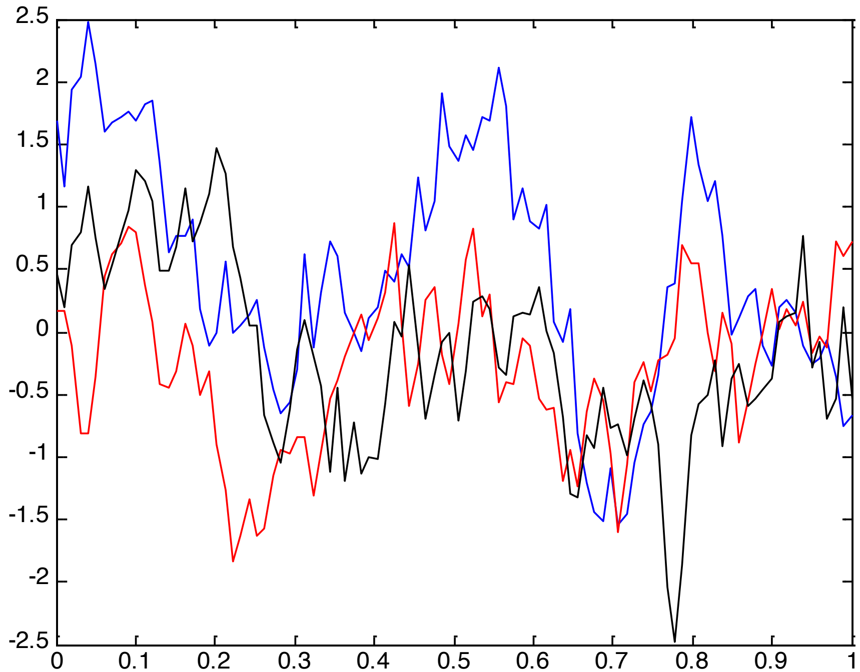
- Exponential kernel
  $K(x,x') = \exp(-|x-x'|/h)$



Distance $|x-x'|$



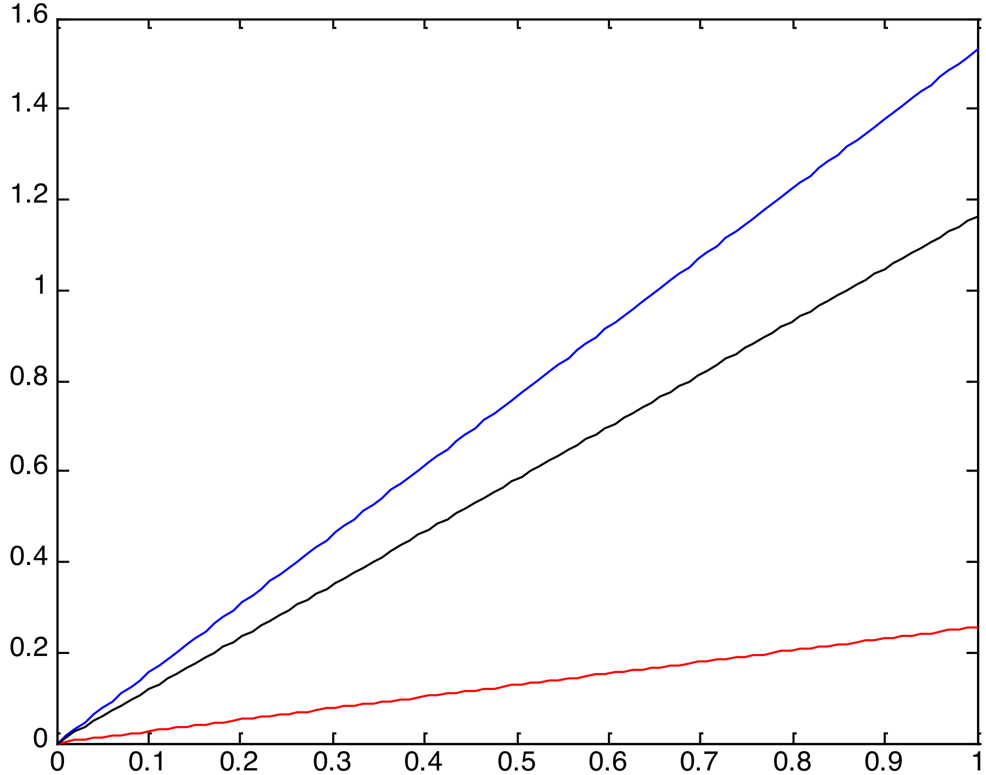
Bandwidth h=1



Bandwidth h=.3

Kernel functions: Examples

- Linear kernel:
  $K(x,x') = x^T x'$

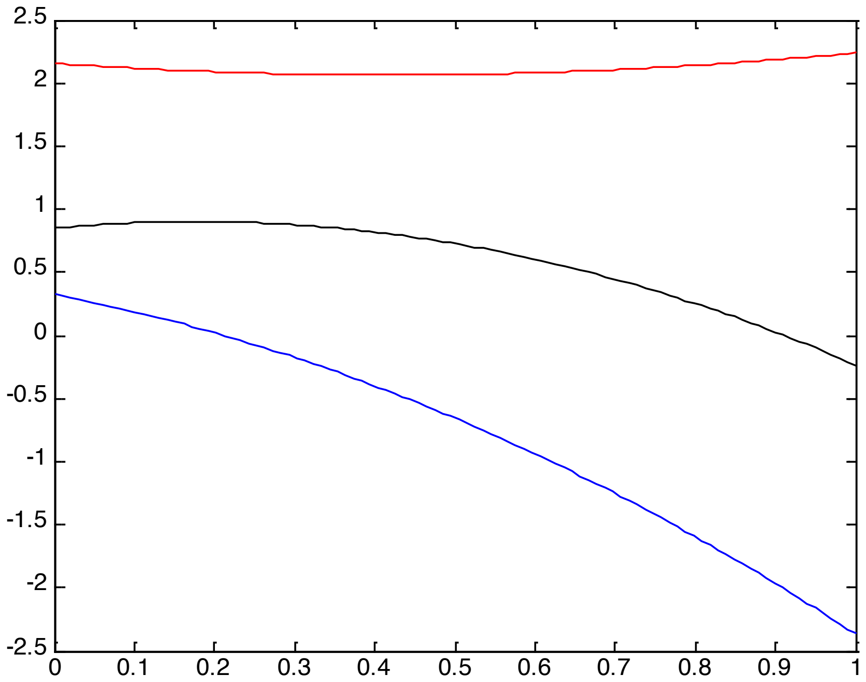


- Corresponds to (Bayesian) linear regression!

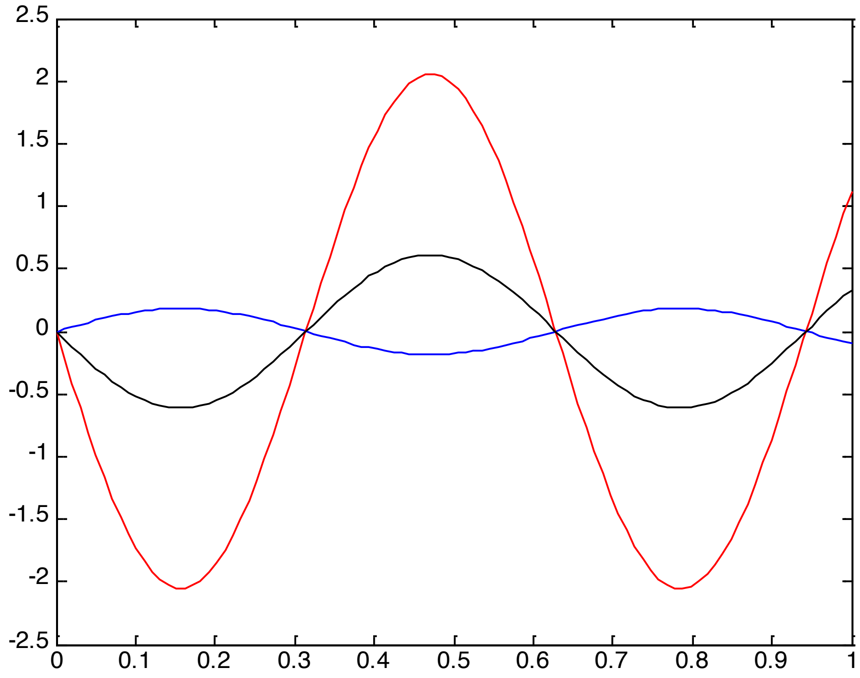
Kernel functions: Examples

- Linear kernel with features:
  $K(x,x') = \Phi(x)^T\Phi(x')$

E.g., $\Phi(x) = [0,x,x^2]$

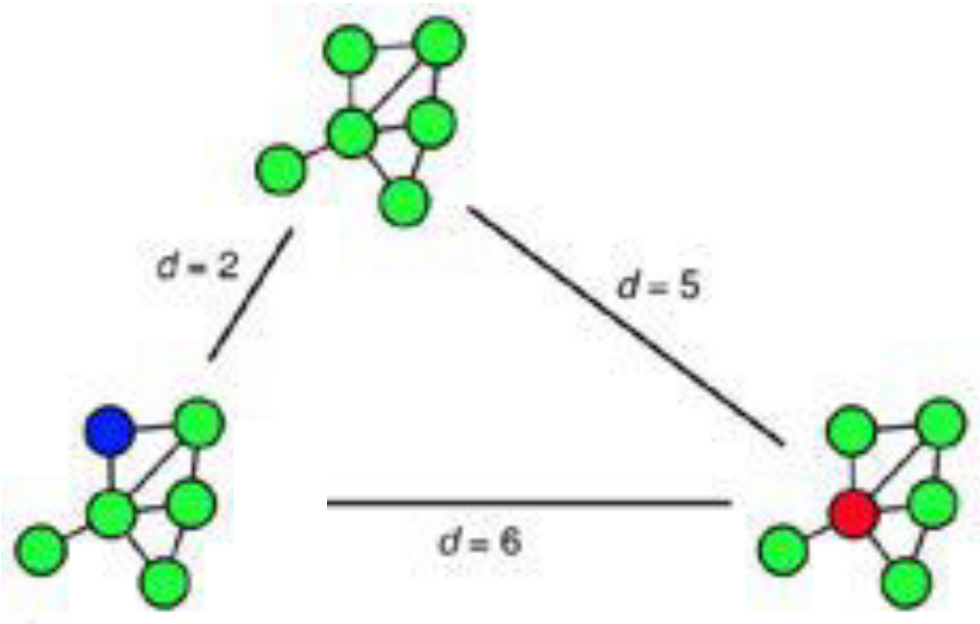


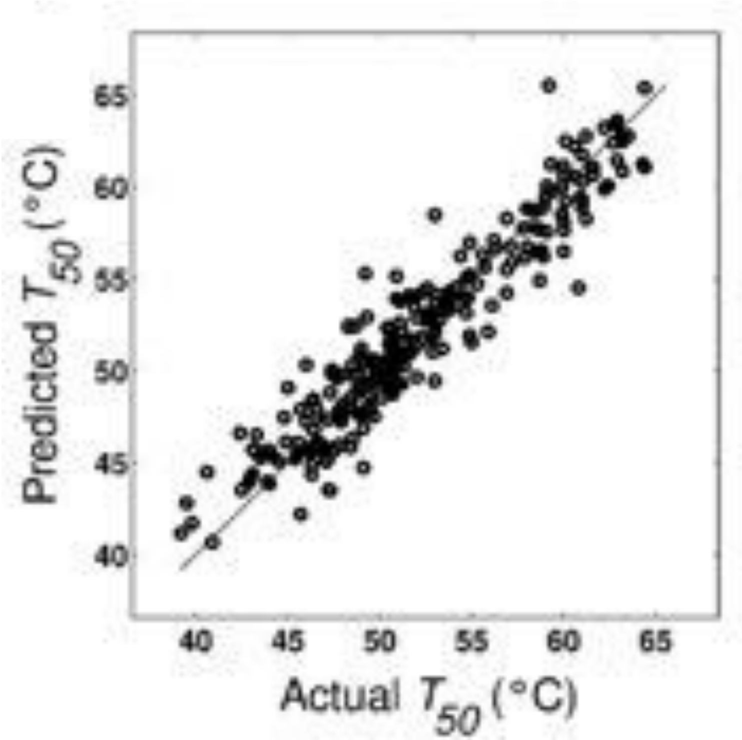
E.g., $\Phi(x) = \sin(x)$



Application: Protein Engineering

[with Romero, Arnold, PNAS '13]





Making predictions with GPs

- Suppose $P(f) = GP(f; \mu, k)$

  and we observe $y_i = f(\mathbf{x}_i) + \epsilon_i$ $\quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$

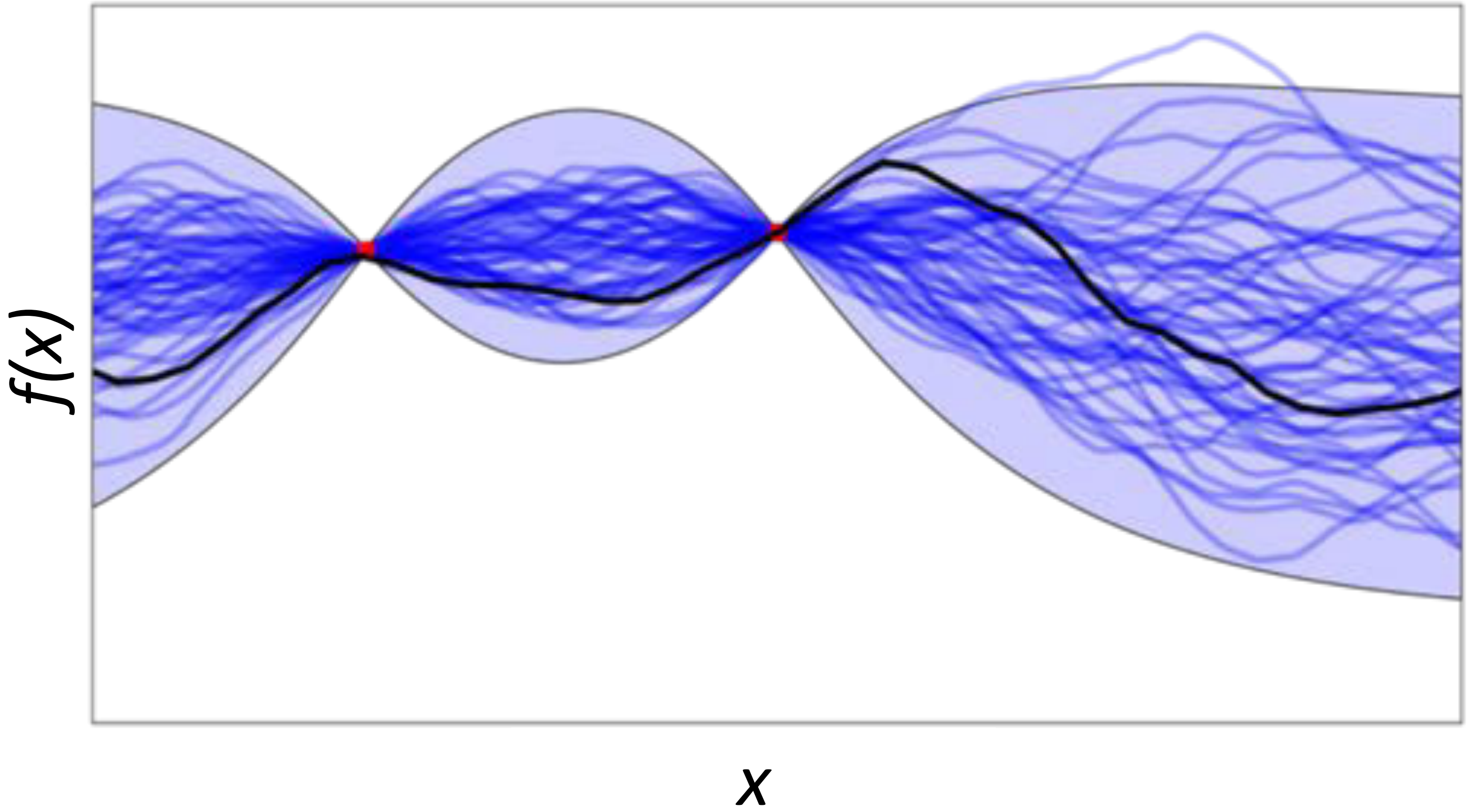
  $A = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
- Then the posterior is also a GP:

$$\boxed{\begin{aligned} &P(f \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, y_1, \ldots, y_n) = GP(f; \mu', k') \\ &\mu'(\mathbf{x}) = \mu(\mathbf{x}) + \mathbf{\Sigma}_{x,A}(\mathbf{\Sigma}_{AA} + \sigma^2 \mathbf{I})^{-1}(\mathbf{y}_A - \mu_A) \\ &k'(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{\Sigma}_{x,A}(\mathbf{\Sigma}_{AA} + \sigma^2 \mathbf{I})^{-1} \mathbf{\Sigma}_{A,x'} \end{aligned}}$$

- Thus, predictive distribution for some test point:

$$P(f(\mathbf{x}) \mid \mathbf{x}_1, \ldots, \mathbf{x}_k, y_1, \ldots, y_k) = \mathcal{N}(f(\mathbf{x}); \mu'(\mathbf{x}), k'(\mathbf{x}, \mathbf{x}))$$

GP Inference Illustration



Where do we get the kernel (parameters) from?

- Prior knowledge
- Empirical Bayes (maximizing marginal likelihood)
- Integrating over hyperparameters
- Online hyperparameter adaptation (→ JMLR 2019)

- For now, assume kernel is given

Active Learning and Optimization with Gaussian Processes

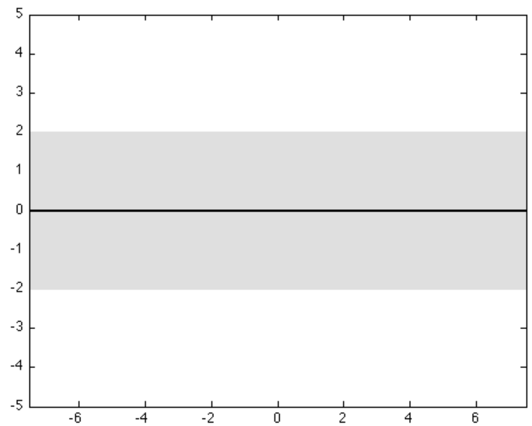
How do we quantify utility? Information gain

[c.f., Lindley '56]

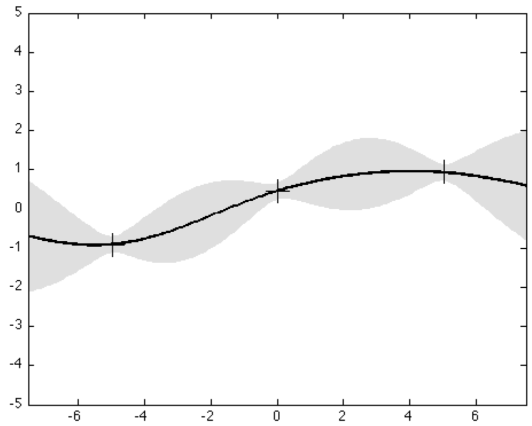
- Set *D* of points to evaluate *f* at
- Find *S* ⊆ *D* maximizing information gain:

$$F(S) = \underbrace{H(f)}_{\text{Uncertainty of } f \text{ before evaluation}} - \underbrace{H(f \mid y_S)}_{\text{Uncertainty of } f \text{ after evaluation}} = I(f; y_S) = \frac{1}{2} \log |I + \sigma^{-2} \Sigma_{SS}|$$

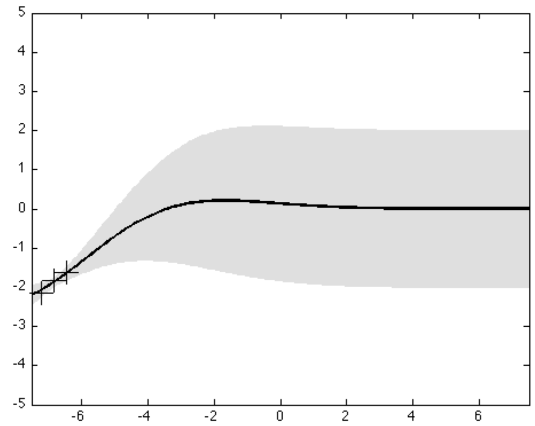
Noisy obs.
at locations *S*



prior



high infogain



low infogain

Optimizing mutual information

[cf Shewry & Wynn ’87]

- Mutual information F(S) is NP-hard to optimize
- Simple strategy: Greedy algorithm. For $S_t = \{x_1, \ldots, x_t\}$

$$x_{t+1} = \arg\max_{x \in D} F(S_t \cup \{x\})$$

$$= \arg\max_{x \in D} H(y_x \mid y_{S_t}) - H(y_x \mid f)$$

$$= \arg\max_{x \in D} \sigma^2_{x|S_t}$$

Constant for fixed noise variance

Entropy is monotonic in variance

Side note: Submodularity of Mutual Information [cf K & Guestrin '05]

- Mutual information F(S) is monotone submodular:

$$\forall x \in D \;\; \forall A \subseteq B \subseteq D : F(A \cup \{x\}) - F(A) \geq F(B \cup \{x\}) - F(B)$$

- Greedy algorithm provides constant-factor approximation [Nemhauser et al'78]

$$F(S_T) \geq \left(1 - \frac{1}{e}\right) \max_{S \subseteq D, |S| \leq T} F(S)$$

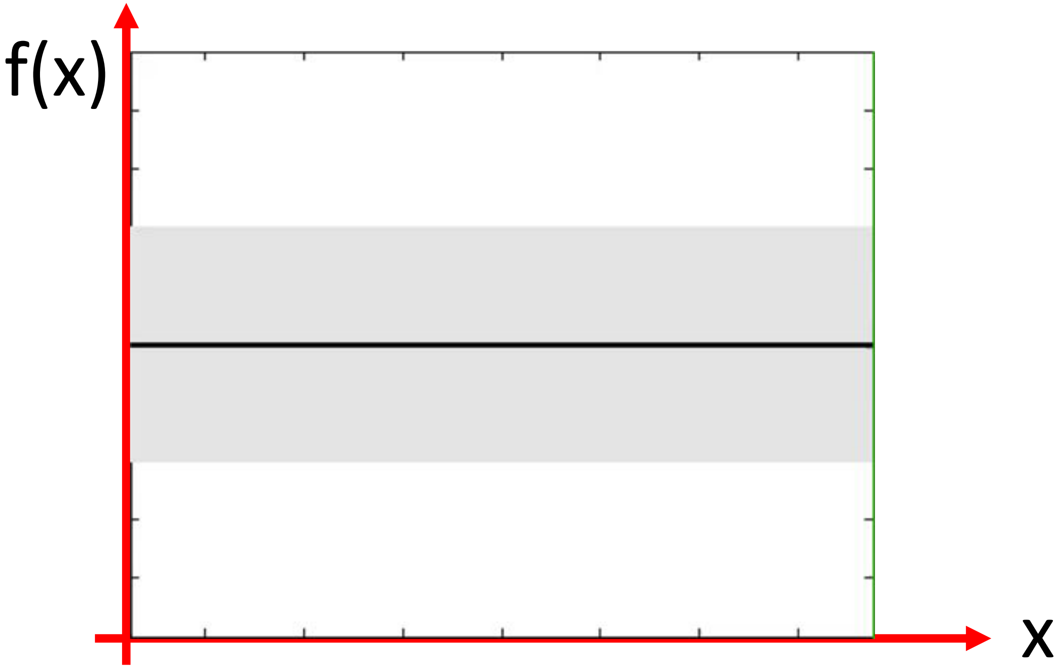
- I.e., uncertainty sampling is near-optimal!

Active Learning: Uncertainty sampling

Pick: $$x_t = \arg\max_{x \in D} \sigma_{t-1}^2(x)$$

In active learning, we reduce uncertainty everywhere

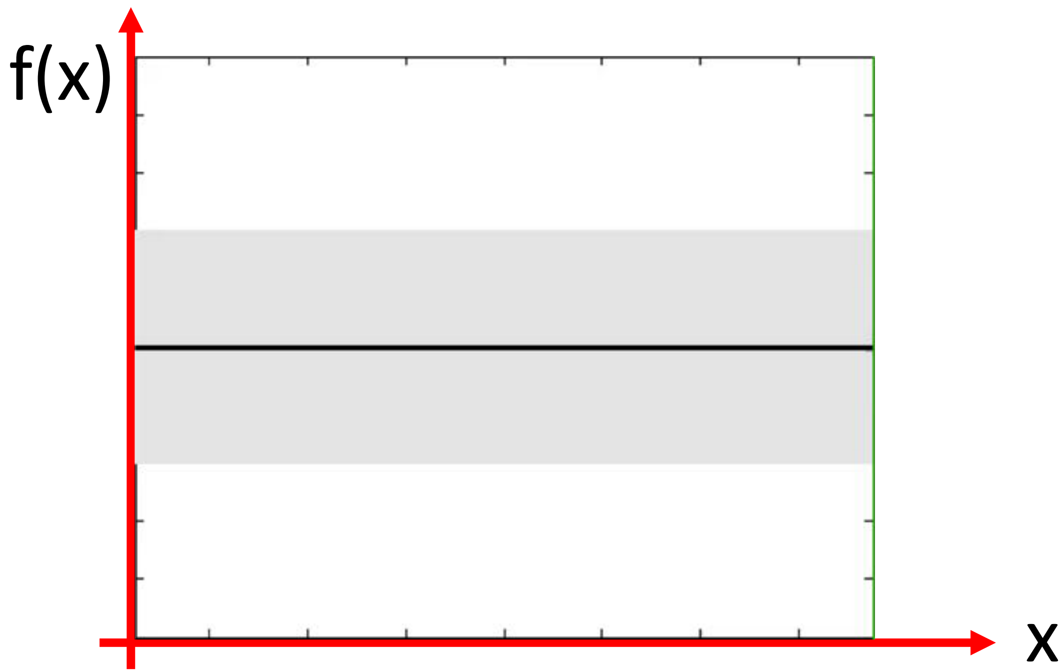
In Bayesian optimization, only care about maximum!



Wastes samples by exploring f everywhere!

Exploiting only

Pick: $$x_t = \arg\max_{x \in D} \mu_{t-1}(x)$$

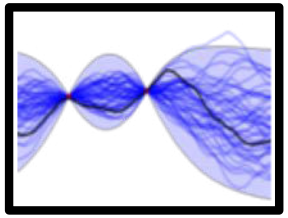


Gets stuck in local optima!

Bayesian Optimization

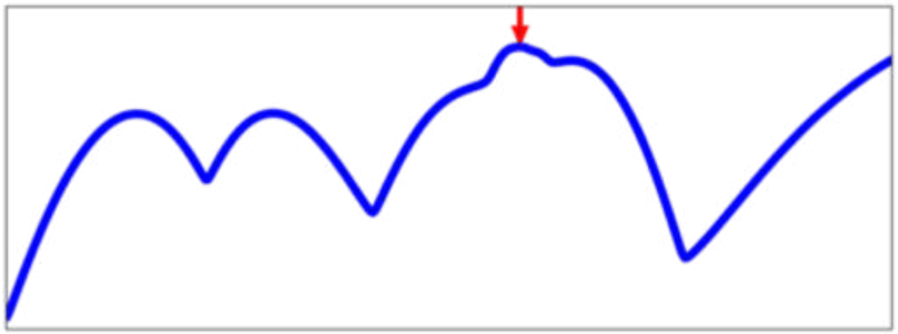
[Močkus *et al.* '78]

$x_t \longrightarrow$



$\longrightarrow \quad y_t = f(x_t) + \epsilon_t$

Acquisition function



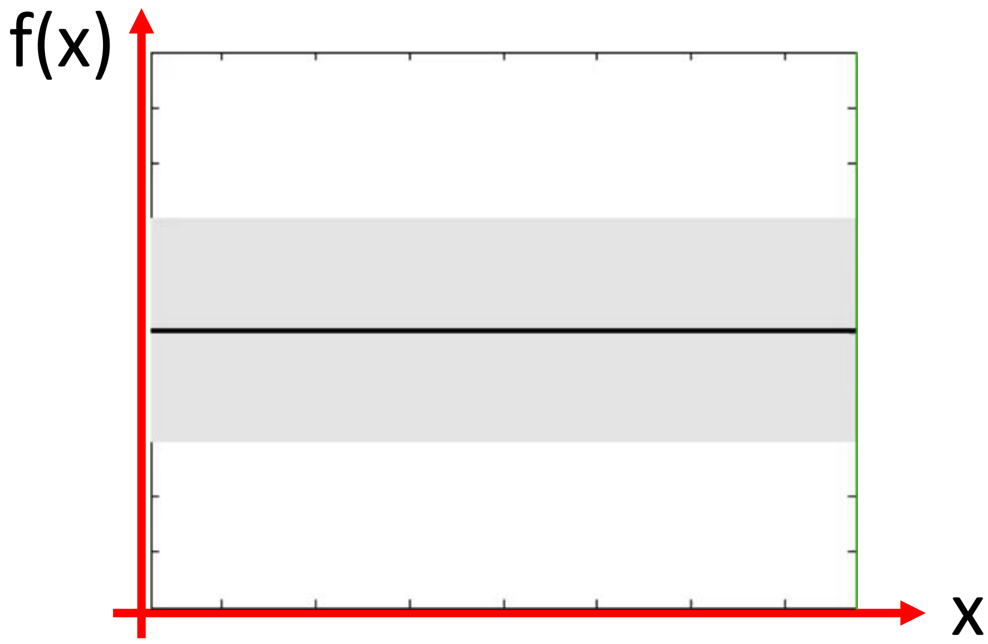
Expected/most prob. improvement [Močkus *et al.* '78,'89], Information gain about maximum [Villemonteix *et al.* '09], Knowledge gradient [Powell *et al.* '10], Predictive Entropy Search [Hernández-Lobato *et al.* '14], TruVaR [Bogunovic et al'17], Max Value Entropy Search [Wang et al'17]

Gaussian process bandit optimization

Goal: Pick inputs x_1,x_2,... s.t.

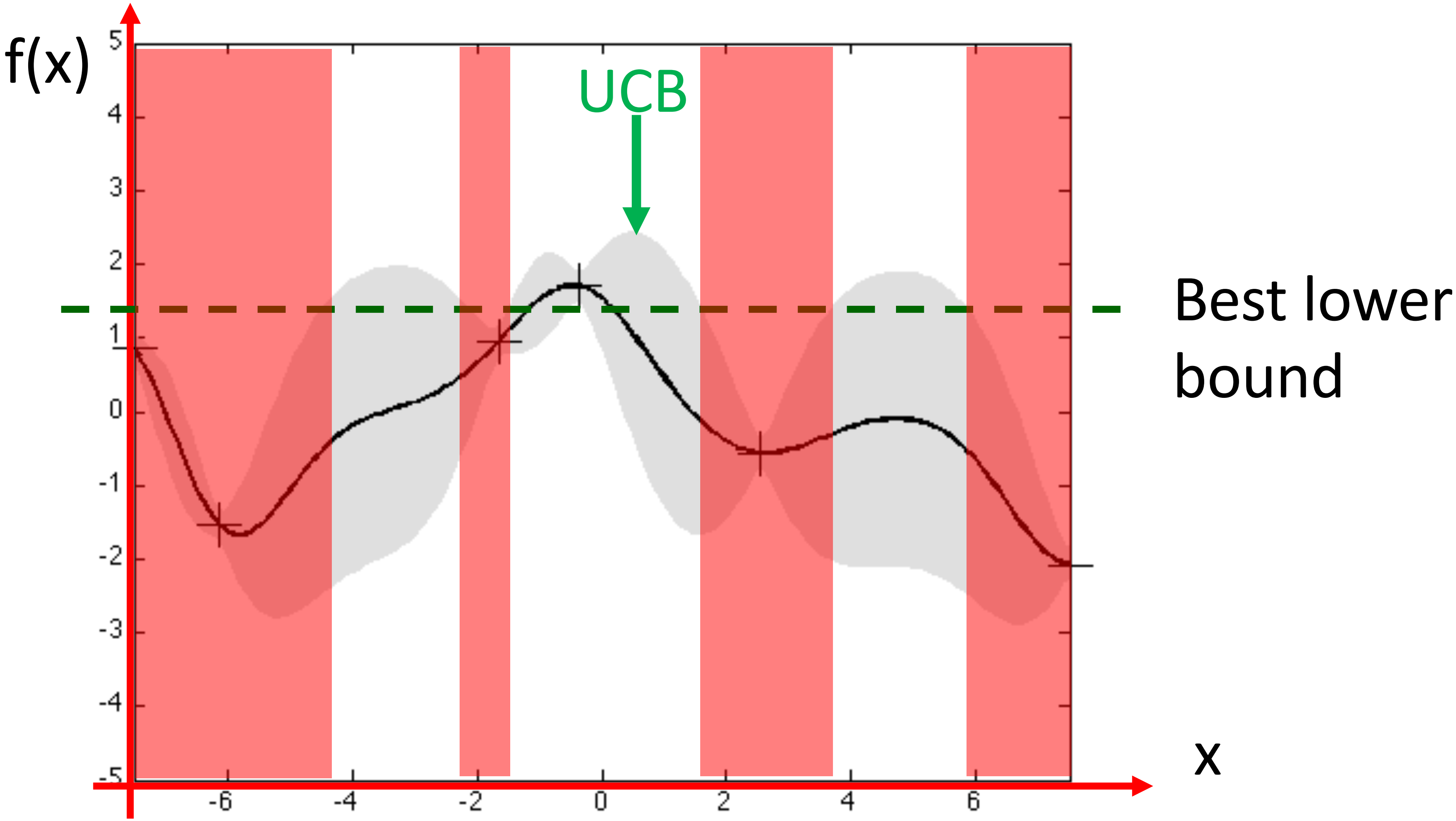
$$\frac{1}{T}\sum_{t=1}^{T}[f(x^*) - f(x_t)] \rightarrow 0$$

Average regret



How should we pick samples to minimize our regret?

Avoiding unnecessary samples



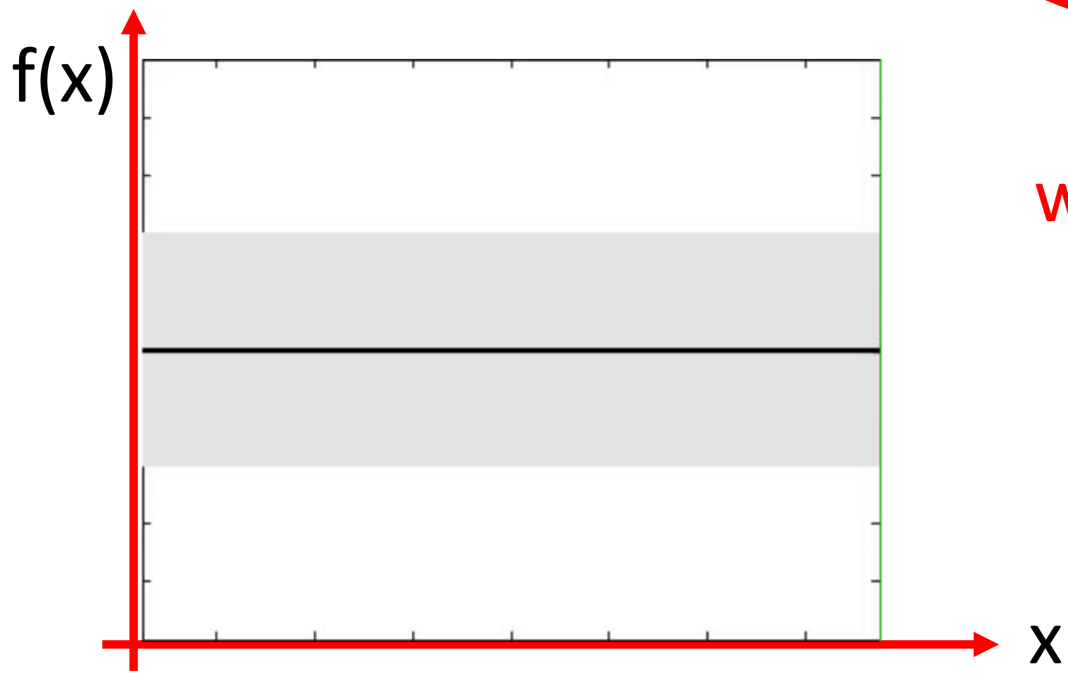
Key insight: Never need to sample where upper confidence limit < best lower bound!

Upper confidence sampling (GP-UCB)

[use in Bandits: e.g., Auer et al '02, Dani'08, …]

Pick input that maximizes upper confidence bound:

$$\boxed{x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)}$$



How should we choose β_t?

Naturally trades off exploration and exploitation

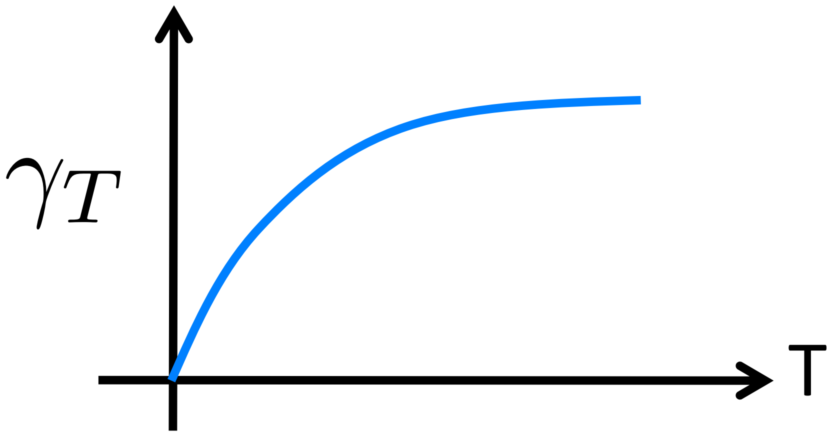
Does not waste samples (with high probability)

Information capacity of GPs

- Will see that regret bounds depend on how **quickly** we can **gain information**
- Mathematically: $$\gamma_T = \max_{|A| \le T} I(f; y_A)$$

$$I(f; y_A) = H(f) - H(f \mid y_A)$$

Optimized in active learning/ uncertainty sampling



Performance of GP-UCB

Theorem [Srinivas, Krause, Kakade, Seeger IEEE IT'12]

If we choose β_t = O(log t), then

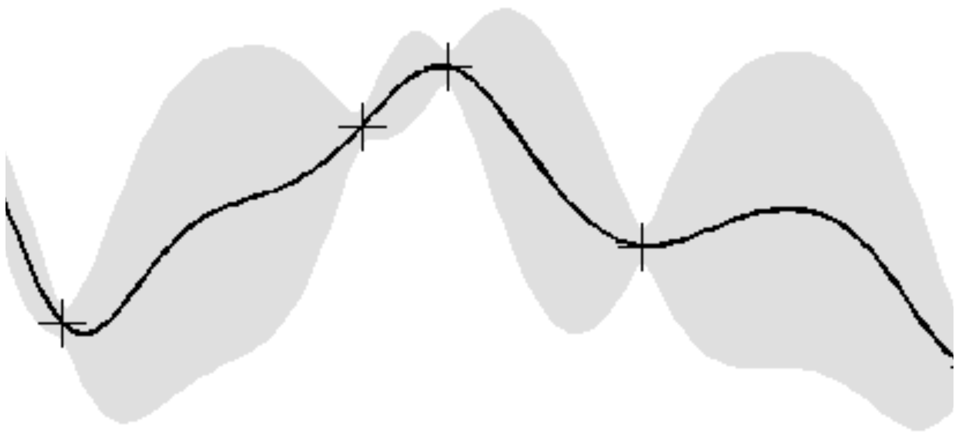
$$\frac{1}{T}\sum_{t=1}^{T}[f(x^*) - f(x_t)] = \mathcal{O}^*\left(\sqrt{\frac{\gamma_T}{T}}\right)$$

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

Information capacity / DOF …

High-level argument

- True function contained in confidence bounds w.h.p.
- Instantaneous regret bounded by confidence interval at UCB action



- Bound cumulative regret by sum of (scaled) squared predictive variances at evaluation points
- Latter is bounded by the log determinant (= mutual information) of selected points

Performance of GP-UCB

Theorem [Srinivas, Krause, Kakade, Seeger IEEE IT'12]

If we choose $\beta_t = O(\log t)$, then

$$\frac{1}{T}\sum_{t=1}^{T}[f(x^*) - f(x_t)] = \mathcal{O}^*\left(\sqrt{\frac{\gamma_T}{T}}\right)$$

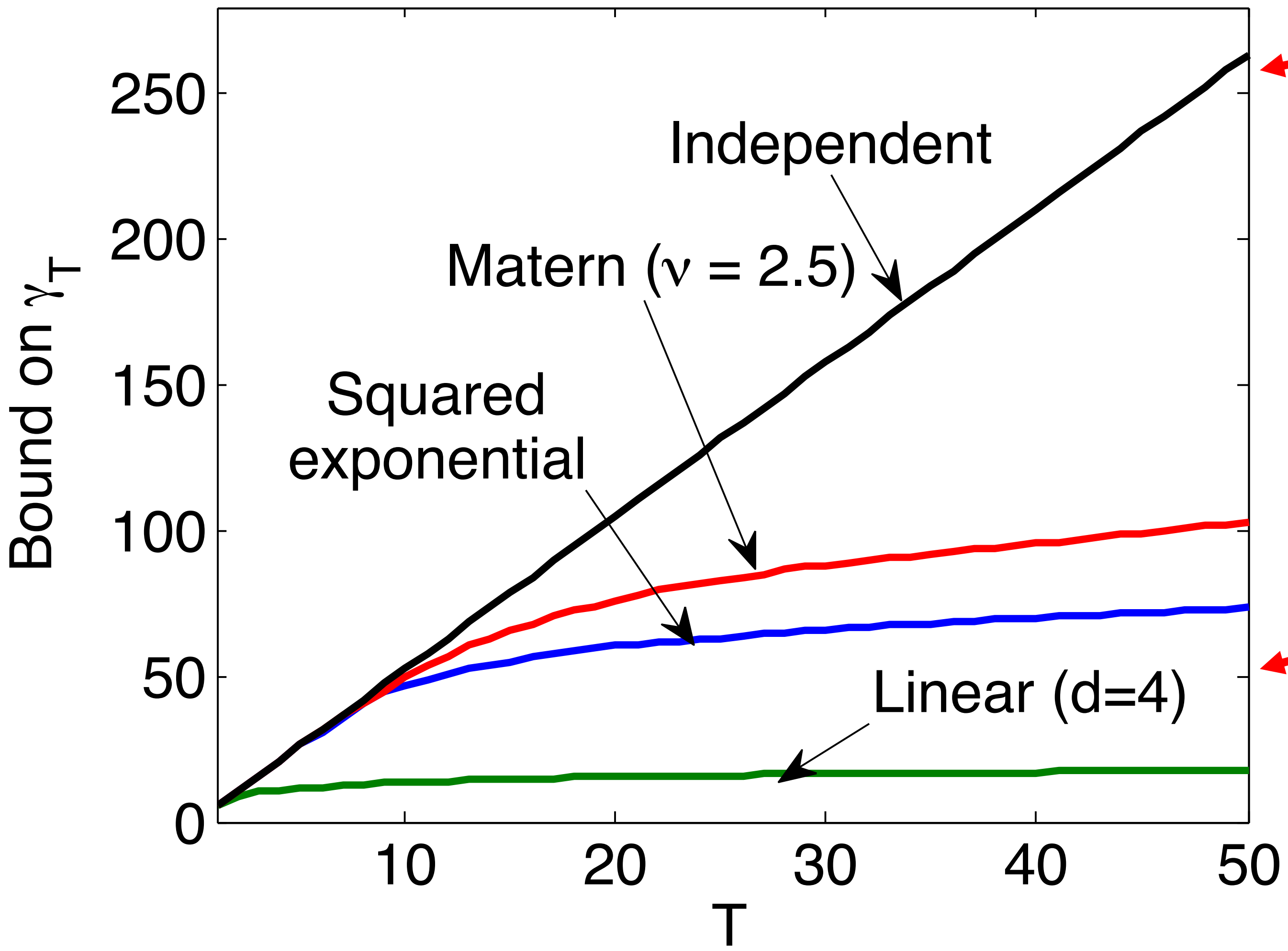
Hereby $$\gamma_T = \max_{|A|\leq T} I(f; y_A)$$

Information capacity / DOF …

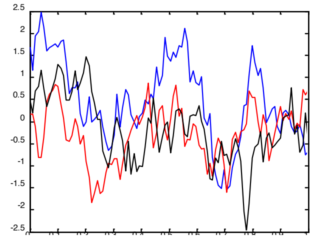
The slower γ_T grows, the easier is f to learn

Key question: **How quickly does γ_T grow??**

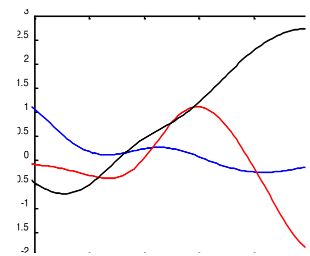
Growth of information gain



<u>Hard</u>: little/no diminishing returns



<u>Easy</u>: strong diminishing returns



Can exploit *submodularity* of mutual info. to compute tight data-dependent bounds

Bounds for common kernels

[Srinivas, Krause, Kakade, Seeger ICML'10; IEEE Trans. IT'12]

Theorem: For the following kernels, we have:

- Linear: $\gamma_T = \mathcal{O}(d \log T)$; $\quad \frac{R_T}{T} = \mathcal{O}^*\left(\frac{d}{\sqrt{T}}\right)$
- Squared-exponential: $\gamma_T = \mathcal{O}((\log T)^{d+1})$;
$$\frac{R_T}{T} = \mathcal{O}^*\left(\frac{(\log T)^{d+1}}{\sqrt{T}}\right)$$
- Matérn with $\nu > 2$, $\gamma_T = \mathcal{O}(T^{\frac{d(d+1)}{2\nu+d(d+1)}} \log T)$;
$$\frac{R_T}{T} = \mathcal{O}^*\left(T^{\frac{\nu+d(d+1)}{2\nu+d(d+1)}-1}\right)$$

Smoothness of *f* helps battle curse of dimensionality!

Our bounds crucially rely on submodularity of γ_T

Robustness?

- So far, have assumed
  - objective f is drawn from a known GP prior
  - Noise is iid Gaussian with known variance
- Robustness w.r.t. these assumptions??

Reproducing Kernel Hilbert Spaces (RKHS)

- Given kernel $k : D \times D \to \mathbb{R}$, consider functions

  $$f(x) = \sum_i \alpha_i k(x_i, x) \quad \text{where} \quad \alpha_i \in \mathbb{R}, x_i \in D$$

  with inner product $\langle f, g \rangle = \sum_{i,j} \alpha_i^f \alpha_j^g k(x_i^f, x_j^g)$

  and norm $||f|| = \sqrt{\langle f, f \rangle}$

- A Reproducing Kernel Hilbert Space (RKHS) is

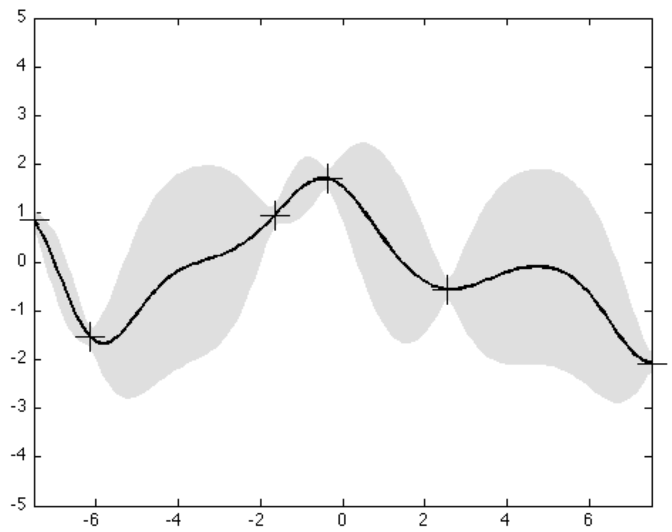
$$H_k(D) = \left\{ f : D \to \mathbb{R}, f(x) = \sum_i \alpha_i k(x_i, x) \text{ s.t. } ||f|| < \infty \right\}$$

Frequentist confidence intervals for GPs?

Theorem: [w Srinivas, Kakade, Seeger'10;

Want: cf Abbasi-Yadkori'12]

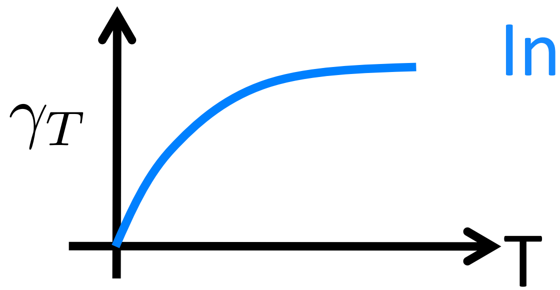
$$\Pr\Big(\forall x, t : f(x) \in [\mu_t(x) \pm \beta_t \sigma_t(x)]\Big) \geq 1 - \delta$$



$$O\left(||f||_k \gamma_t \log^3 t \log \frac{1}{\delta}\right)$$

"Complexity" of *f*
(RKHS norm)

$$\gamma_T = \underbrace{\max_{|A| \leq T} I(f; y_A)}_{\text{Information capacity}}$$



What if f is not from a GP?

[Srinivas, Krause, Kakade, Seeger ICML'10; IEEE Trans. IT'12]

- In practice, f may not be Gaussian

Theorem: Let f lie in the RKHS of kernel K with $\|f\|_K^2 \leq B$, and let the noise be bounded almost surely by σ. Choose $\beta_t = \mathcal{O}(2B + \gamma_t \log^3 t)$. Then w. high probability

$$\frac{R_T}{T} = \mathcal{O}\left(\sqrt{\frac{\beta_T \gamma_T}{T}}\right)$$

- Don't need to know the "true prior"
- Intuitively, the bound depends on the "complexity" of the function through its RKHS norm

Side note: Lower Bounds?

- Upper bounds tight for Gaussian kernel [Scarlett ICML ' 18]
- Open whether they can be improved for Matern kernel

Side note: Optimizing the acquisition function

- GP-UCB requires solving the problem

$$x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)$$

- This is generally non-convex ☹

- In low-D, can use Lipschitz-optimization (DIRECT, etc.)
- In high-D, can use gradient ascent (based on random initialization)
- More later

Beyond Basic BO: More Complex Settings

Confidence based sampling

- Key idea behind GP-UCB
  - Utilize high-probability bounds on function value to constrain sampling
  - Information-capacity bounds problem complexity

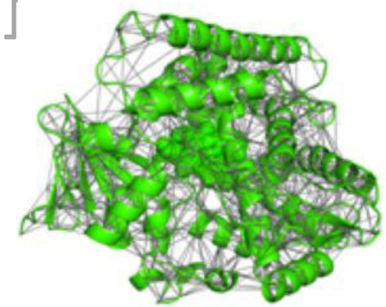
- Can generalize to more complex settings
  - Parallelizing exploration tradeoffs
  - Context / Side-information
  - Multi-objective optimization
  - Level-set identification
  - High-dimensions
  - Constraints

Beyond Basic BO:

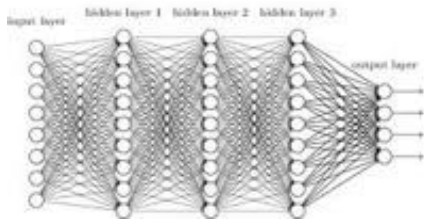
Parallel Exploration/ Delayed Feedback

Parallelizing exploration—exploitation tradeoffs

[cf Azimi et al ‘10; Ginsbourger et al ’10]



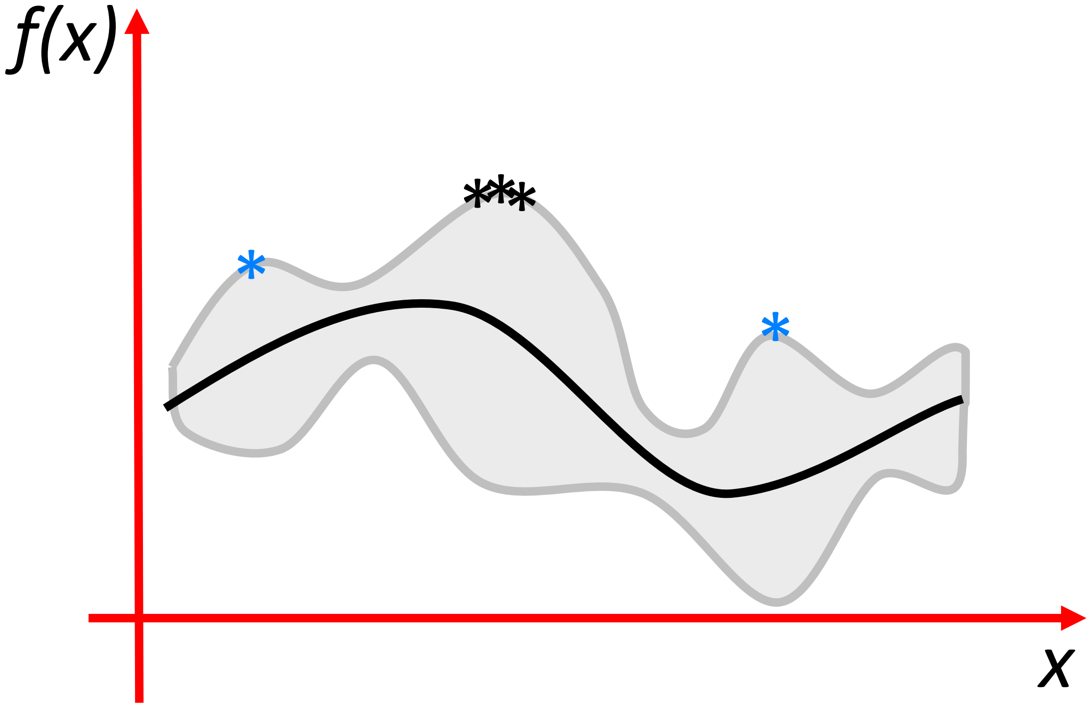
- Basic algorithm is fully sequential
  - Needs $y_1, \ldots, y_t$ to choose x_{t+1}
- In many applications, wish to perform batch of multiple (say B) evaluations in parallel



- *How should we choose the batch?*
- *How much “informational speedup” can we get?*

Naïve approaches

- Pick a single query and run this experiment *B* times?
  - Intuitively seems like we could do better
- Pick top *B* queries in GP-UCB criterion?
  - Problem: likely close to one another, no diversity

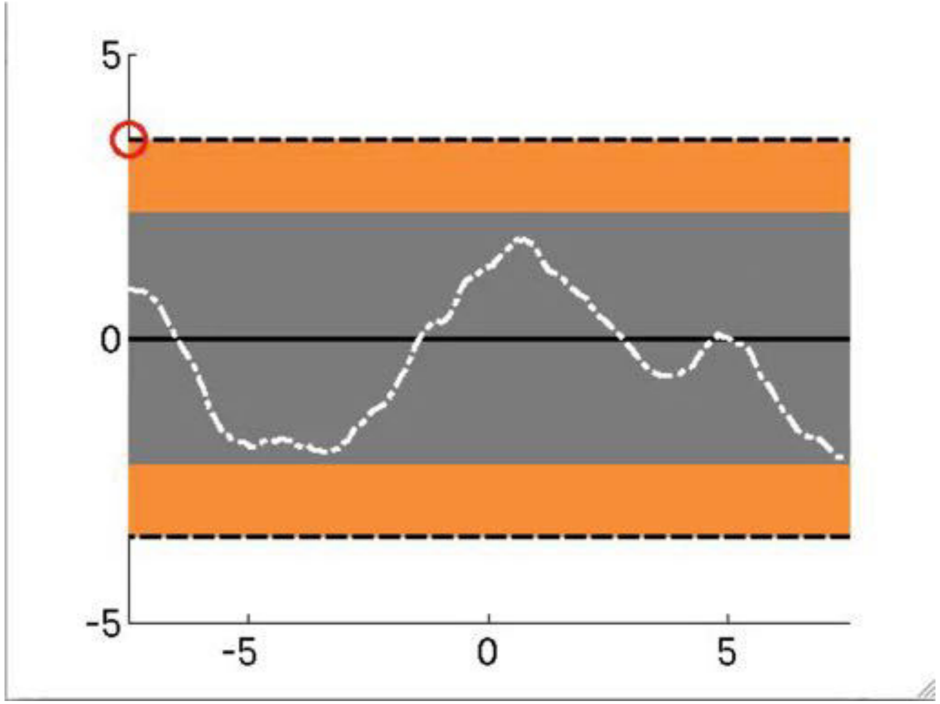


Batch Mode GP Optimization

[cf. "Kriging Believer", Ginsbourger et al '10]

$$\boxed{x_t = \arg\max_{x \in D} \left[\mu_{t_b - 1}(x) + \beta_t^{(1/2)} \sigma_{t-1}(x) \right]}$$

- Update $\sigma_{t-1}(x)$ after each selection
- This scheme anticipates information we are gaining
- Must be careful to avoid overconfidence!



GP-BUCB mode guarantees

Theorem [Desautels, K, JMLR '14]

For sufficiently regular kernels, can choose

$$B = \mathcal{O}(\log(T))$$

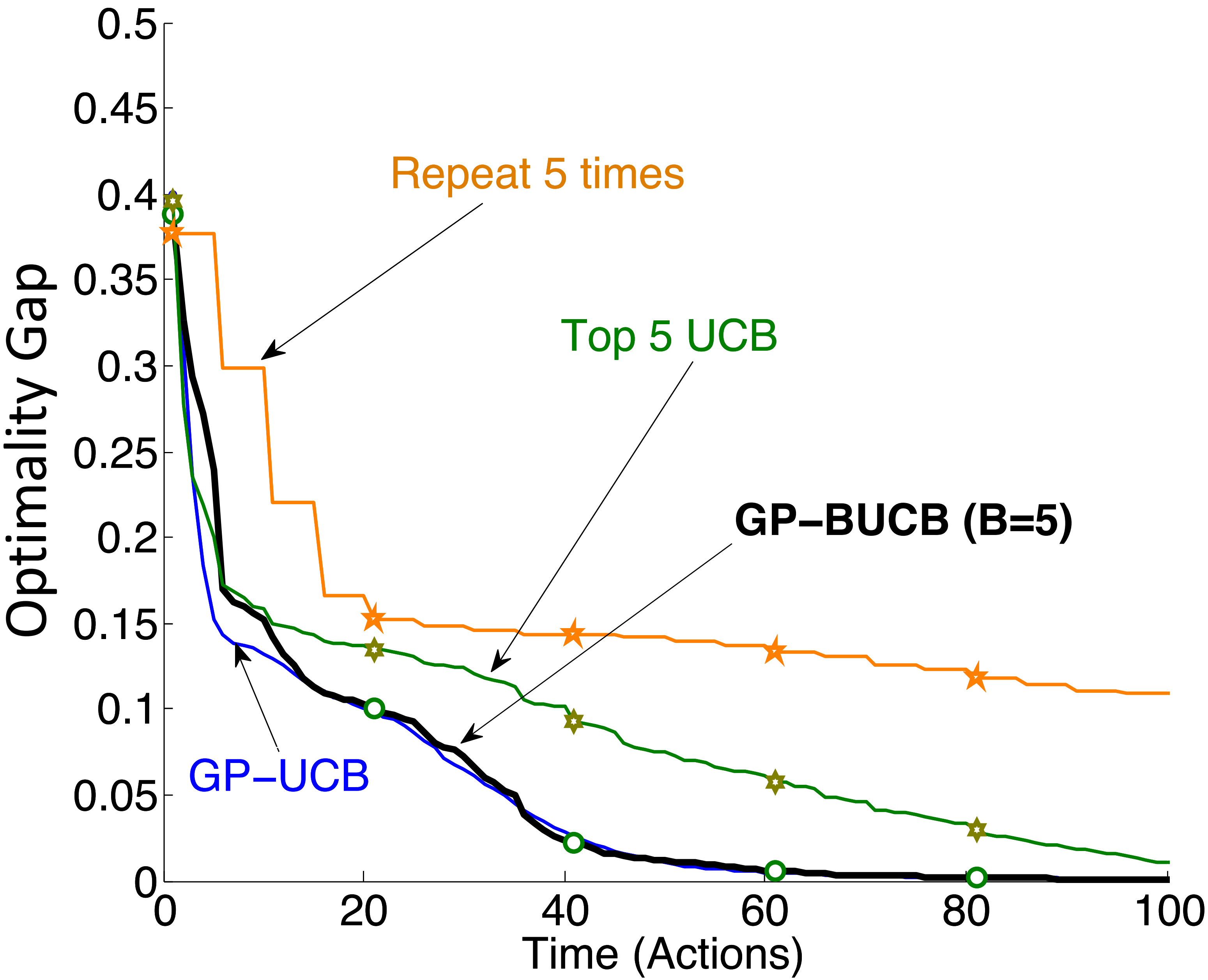
and nevertheless obtain regret

$$R_T^{\text{batch}} = c(d, K) \cdot R_T^{\text{seq}} + \mathcal{O}(\text{polylog}(T, B))$$

Independent of B and T!

➔ Near-linear speedup in convergence rate!

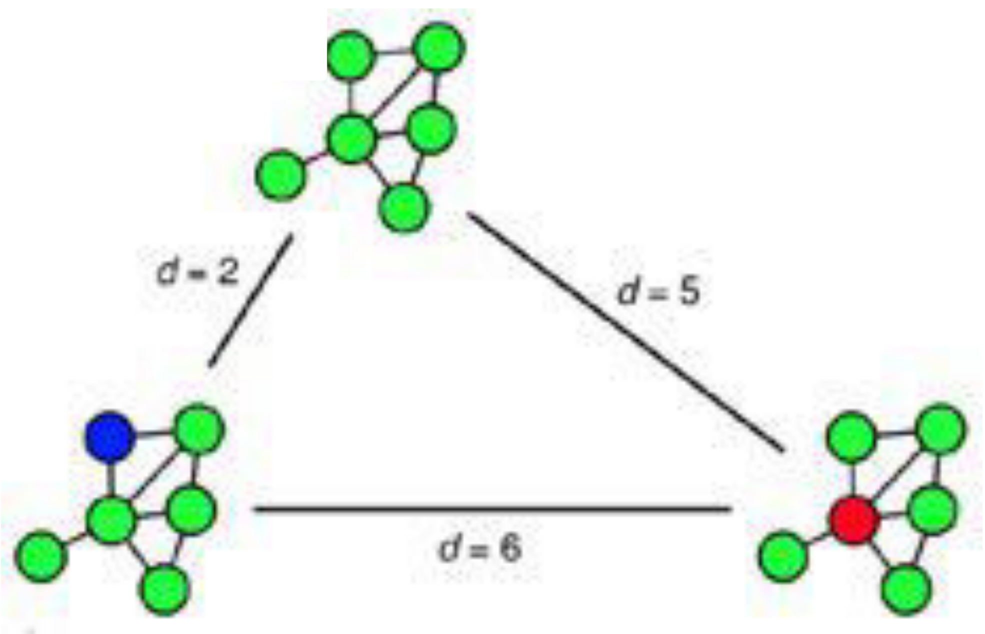
Simulation Results

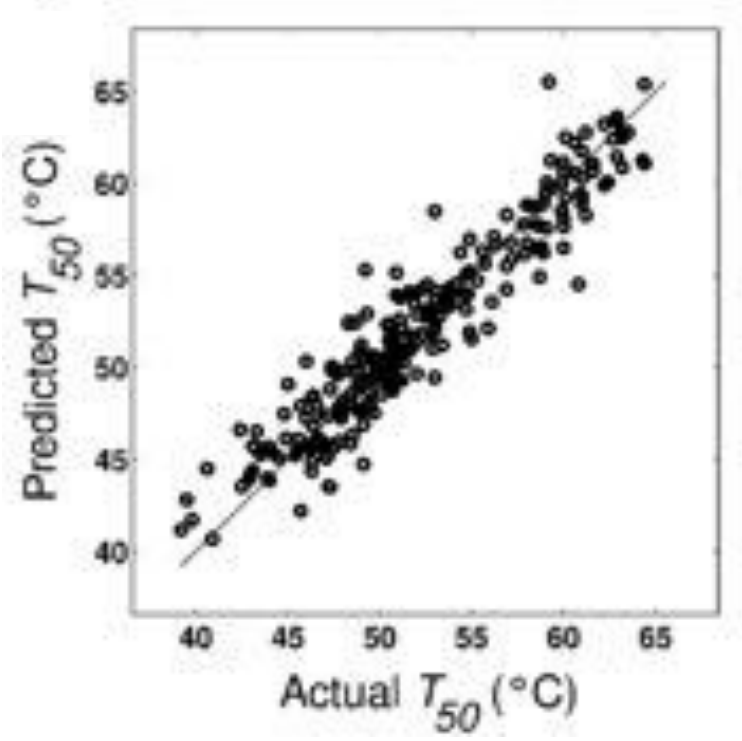


Application: Protein Engineering

[with Romero, Arnold, PNAS '13]

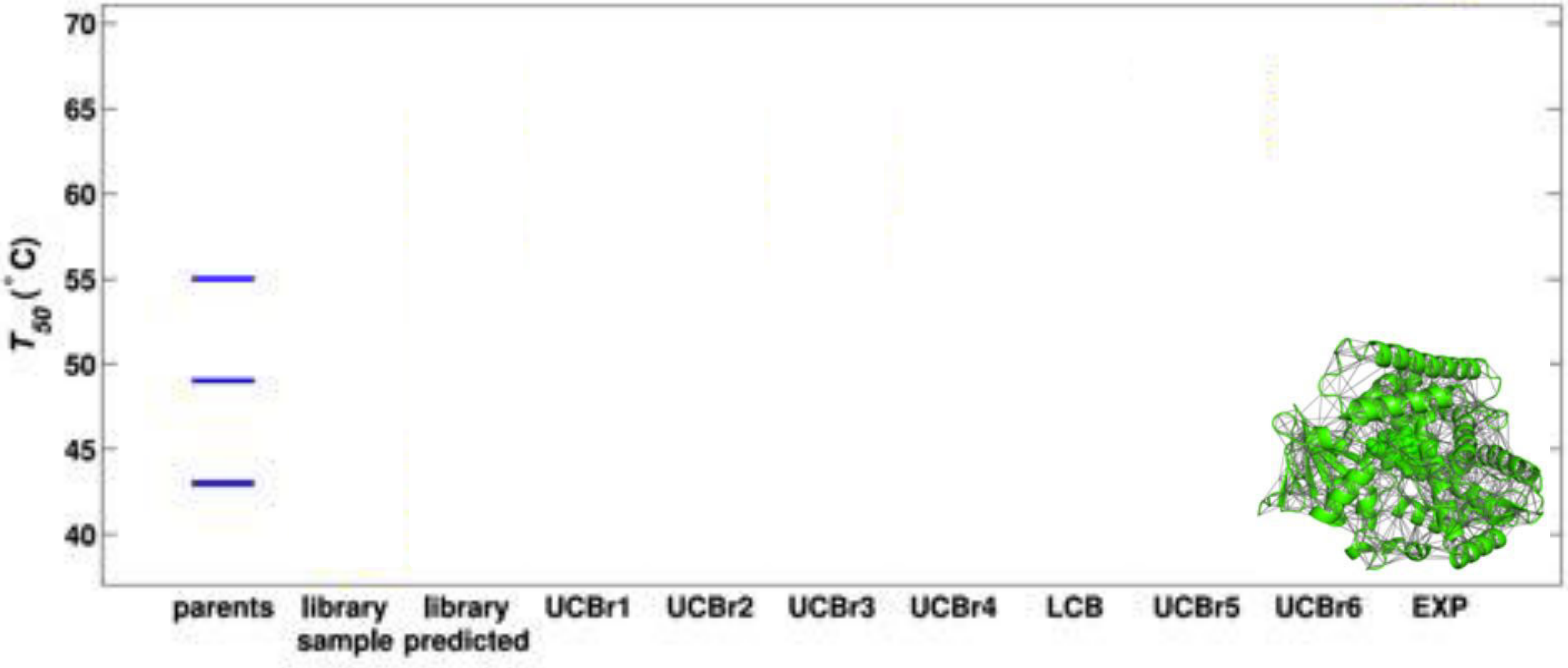
- Task: Design cytochrome P450s chimeras
- Action: Experiment with protein sequence
- Feedback: Thermostability T_{50}
- Kernel: Structure-based kernel function





Wet-lab results

[w Romero, Arnold PNAS '13]



- Identification of new thermostable P450s chimera
- **5.3C more stable than best published sequence!**

Google Vizier: A Service for Black-Box Optimization

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, D. Sculley

[KDD 2017]

“…, Vizier defaults to using Batched Gaussian Process Bandits [8]"

“Vizier is used across Google to optimize hyperparameters of machine learning models, both for research and production models. Our implementation scales to service the entire hyperparameter tuning workload across Alphabet, which is extensive."

“Vizier has made notable improvements to production models underlying many Google products, resulting in measurably better user experiences for over a billion people."

Other strategies

- Multi-point expected improvement [Schonlau ‘97]
- Simulation matching [Azimi et al ’10]
- DPP sampling [Kathuria et al ‘16]

Beyond Basic BO:

Multi-task/ Contextual BO

Contextual GP bandits

In each round t do:

- Observe context $z_t \in Z$
- Pick $x_t \in D$
- Observe $y_t = f(x_t, z_t) + \epsilon_t$
- Incur regret $r_t = \sup_x f(x, z_t) - f(x_t, z_t)$

Modeled as GP

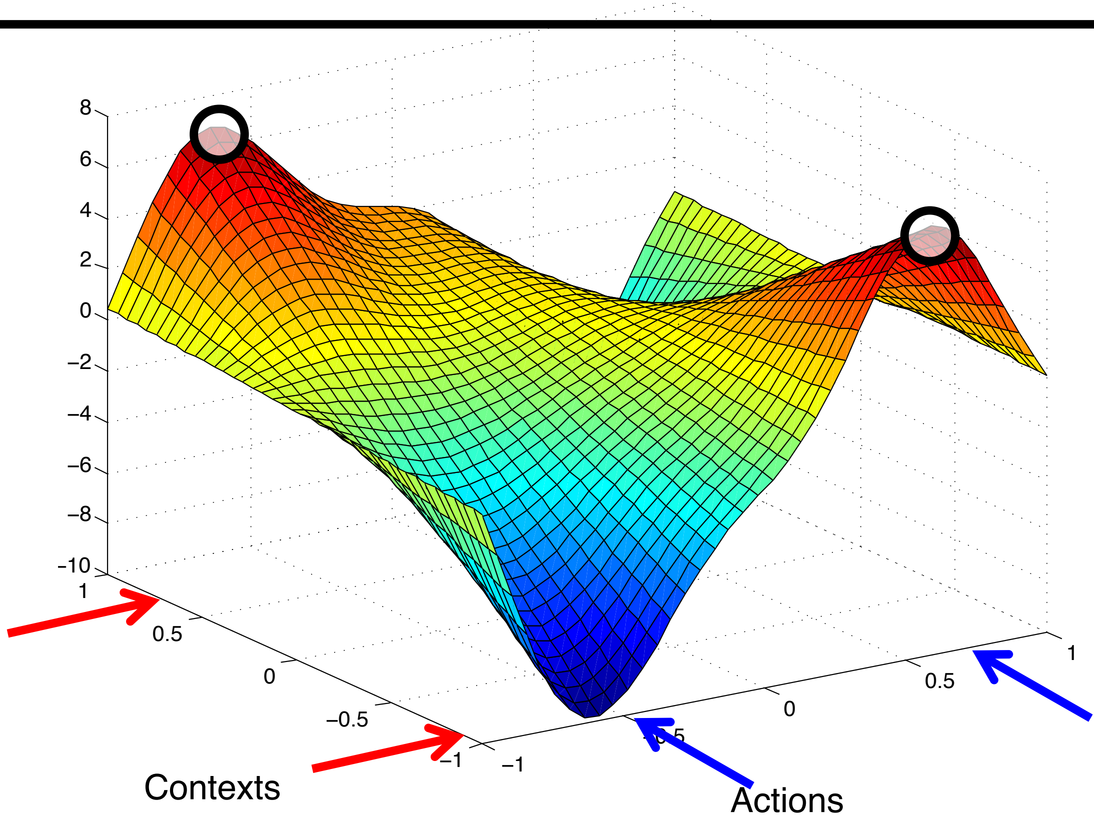
- Cumulative contextual regret $R_T = \sum_{t=1}^{T} r_t$
- Obtaining sublinear regret $R_T/T \to 0$ requires learning optimal mapping from contexts to actions!

CGP-UCB

[generalizes LinUCB: Li et al'10]

Pick input that maximizes upper confidence bound at current context

$$\boxed{x_t = \arg\max_{x \in D} \mu_{t-1}(x, z_t) + \beta_t \sigma_{t-1}(x, z_t)}$$



Where do we get the kernel from?

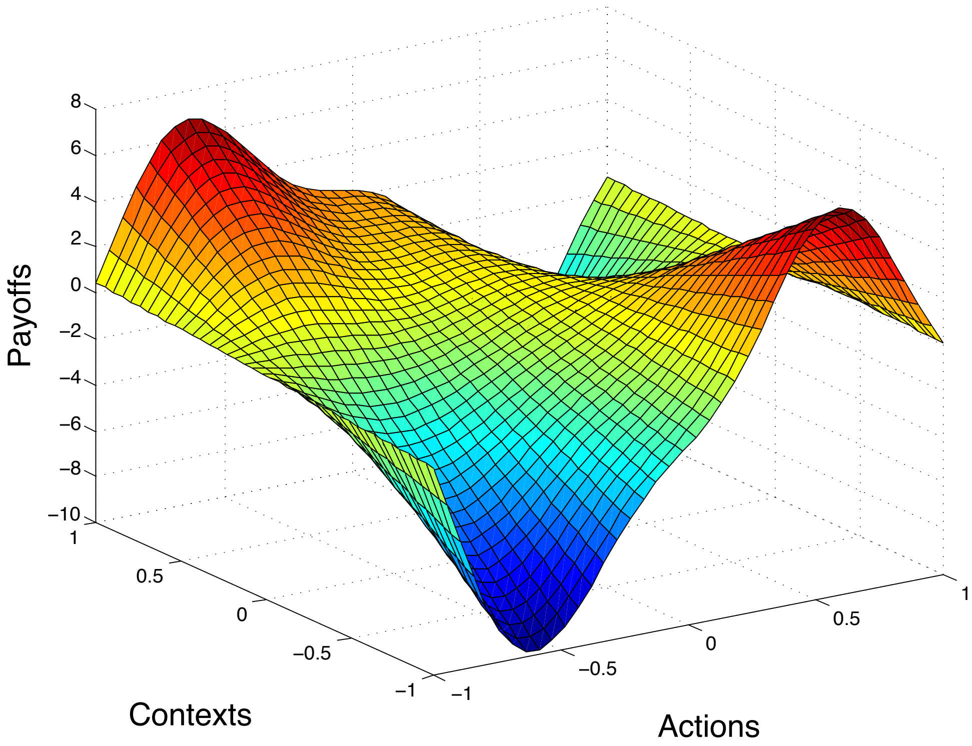
In principle, can choose any kernel on $D \times Z$

Suppose we have kernels $k_D(x, x')$ and $k_Z(z, z')$

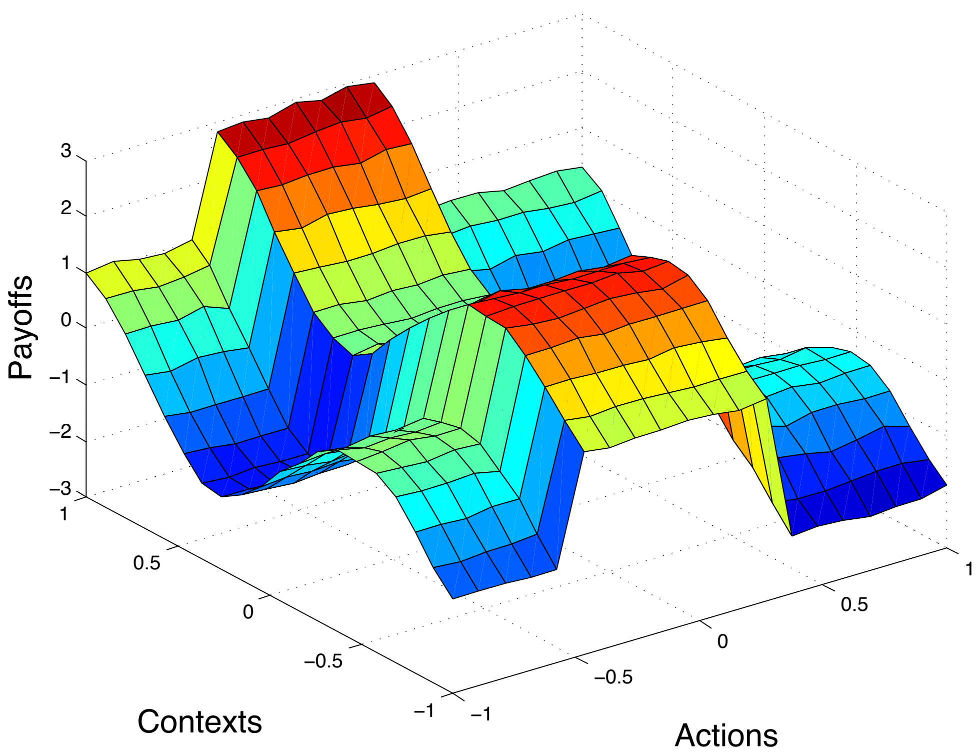
Can compose to kernel on $D \times Z$ through

- Multiplication: $k\big((x, z), (x', z')\big) = k_D(x, x') \cdot k_Z(z, z')$
- Addition: $k\big((x, z), (x', z')\big) = k_D(x, x') + k_Z(z, z')$

Examples



Product



Addition

Can bound the information gain for composite kernels based on that of constituent kernels [cf K & Ong NIPS '11]

Performance of CGP-UCB

Theorem [K, Ong NIPS '11]

If we choose $\beta_t = O(\log t)$, then

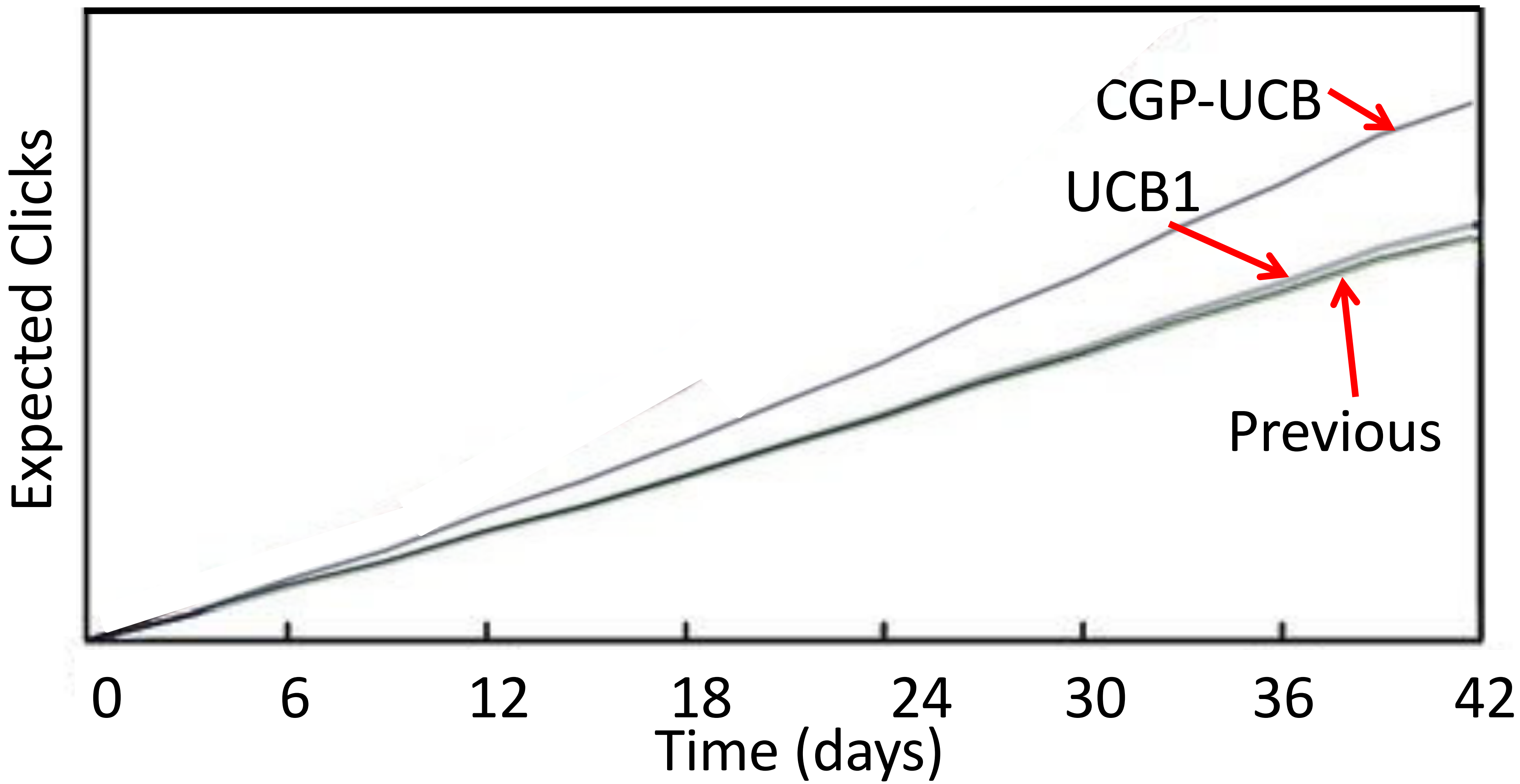
$$\frac{1}{T}\sum_{t=1}^{T}[f(x_t^*, z_t) - f(x_t, z_t)] = \mathcal{O}^*\left(\sqrt{\frac{\gamma_T}{T}}\right)$$

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

Thus, information gain even bounds stronger notion of contextual regret!

Book recommendation results

[w Nikolic, Vanchinathan, de Bona, RecSys '14]



Beyond Basic BO:

Multiple objectives

Multi-objective performance optimization

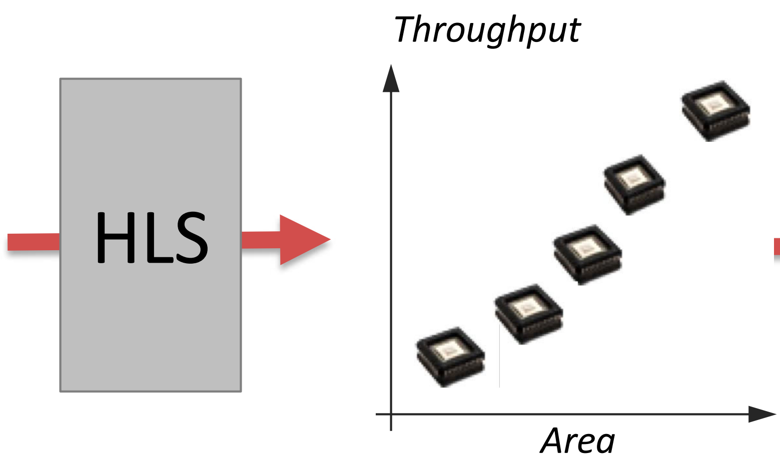
[w Zuluaga, Sergent, Püschel, JMLR '16]

- Protein structure optimization
  - Trade binding affinity & thermostability
- Empirical algorithmics
  - Trade performance & memory footprint
- Design of special purpose hardware
  - Trade area, throughput, energy, runtime ...
- ...

- **Evaluation can be costly and noisy**

High-level synthesis for high-performance computing

High-level specification



Best

Throughput

Latency

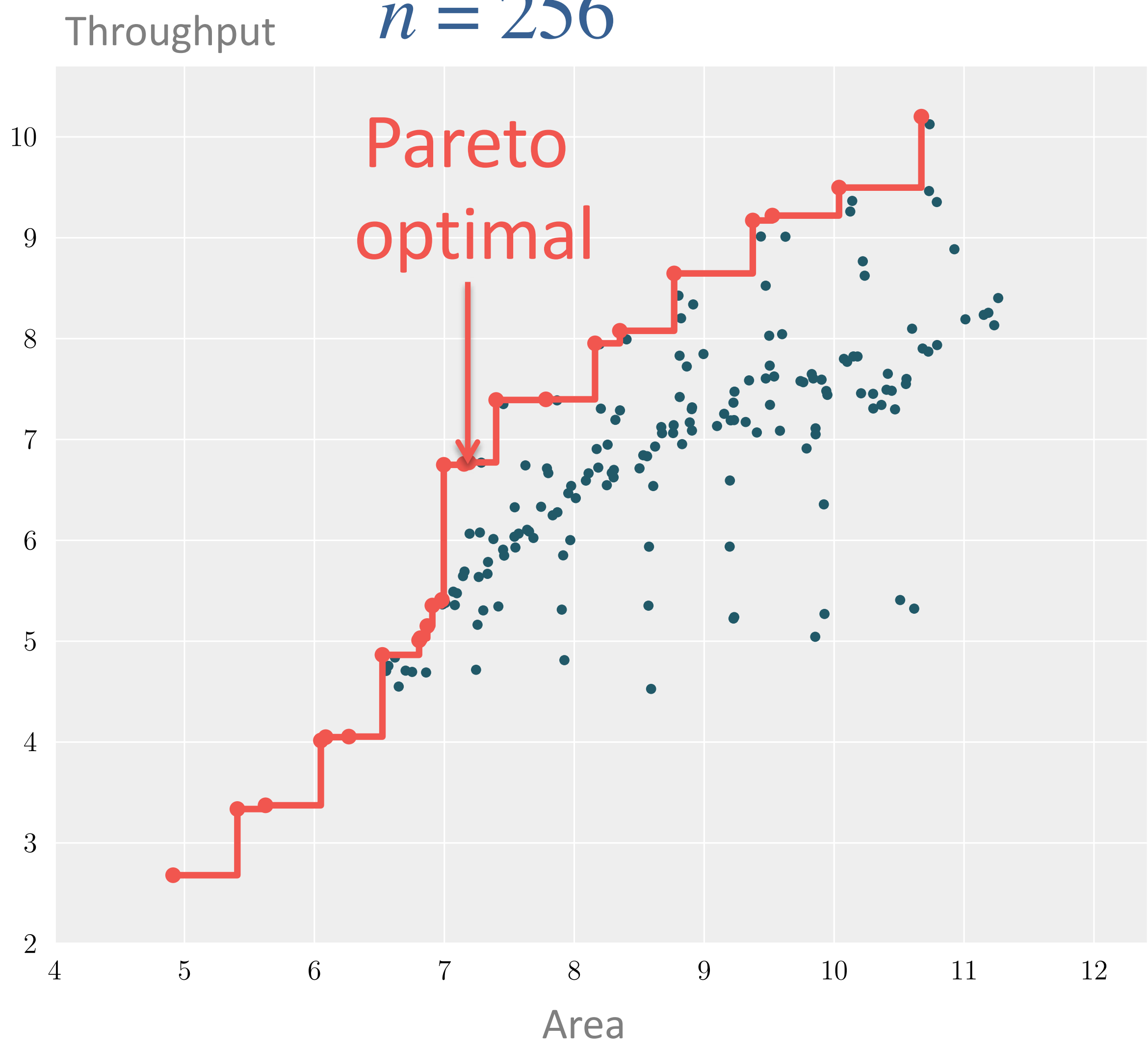
Area

Energy

Exploring the Design Space

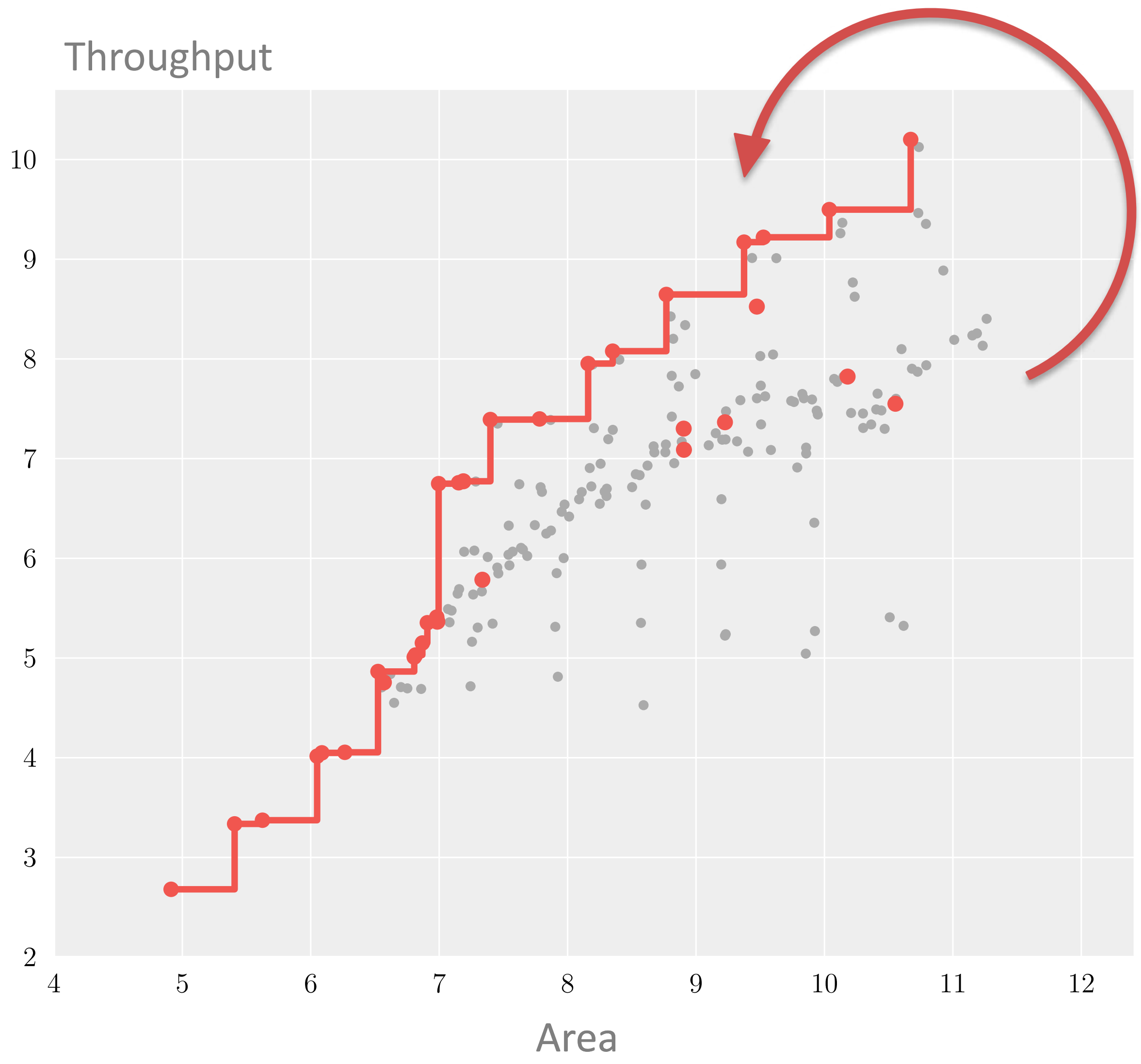
Sorting Networks

$n = 256$

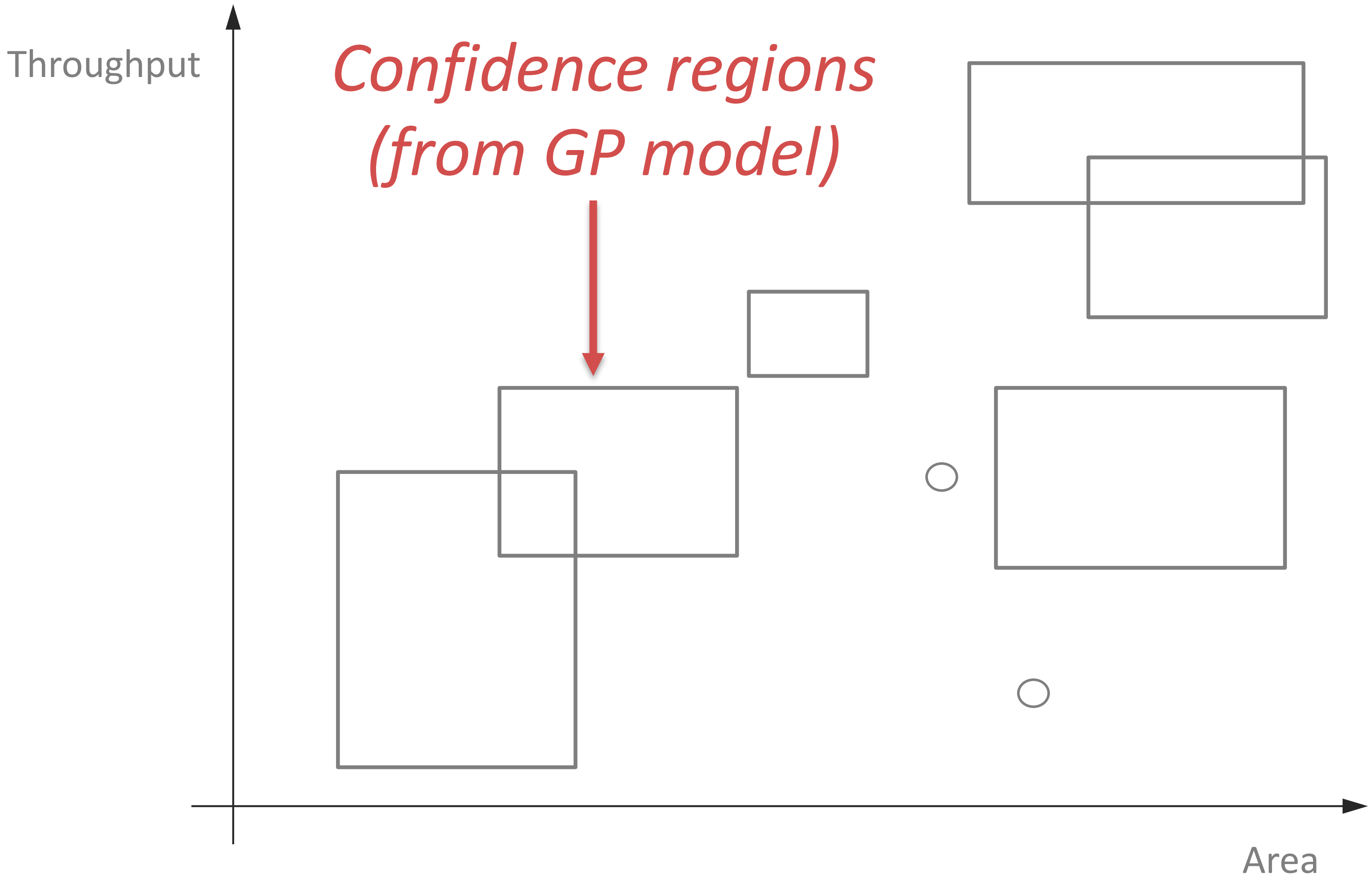


Goal

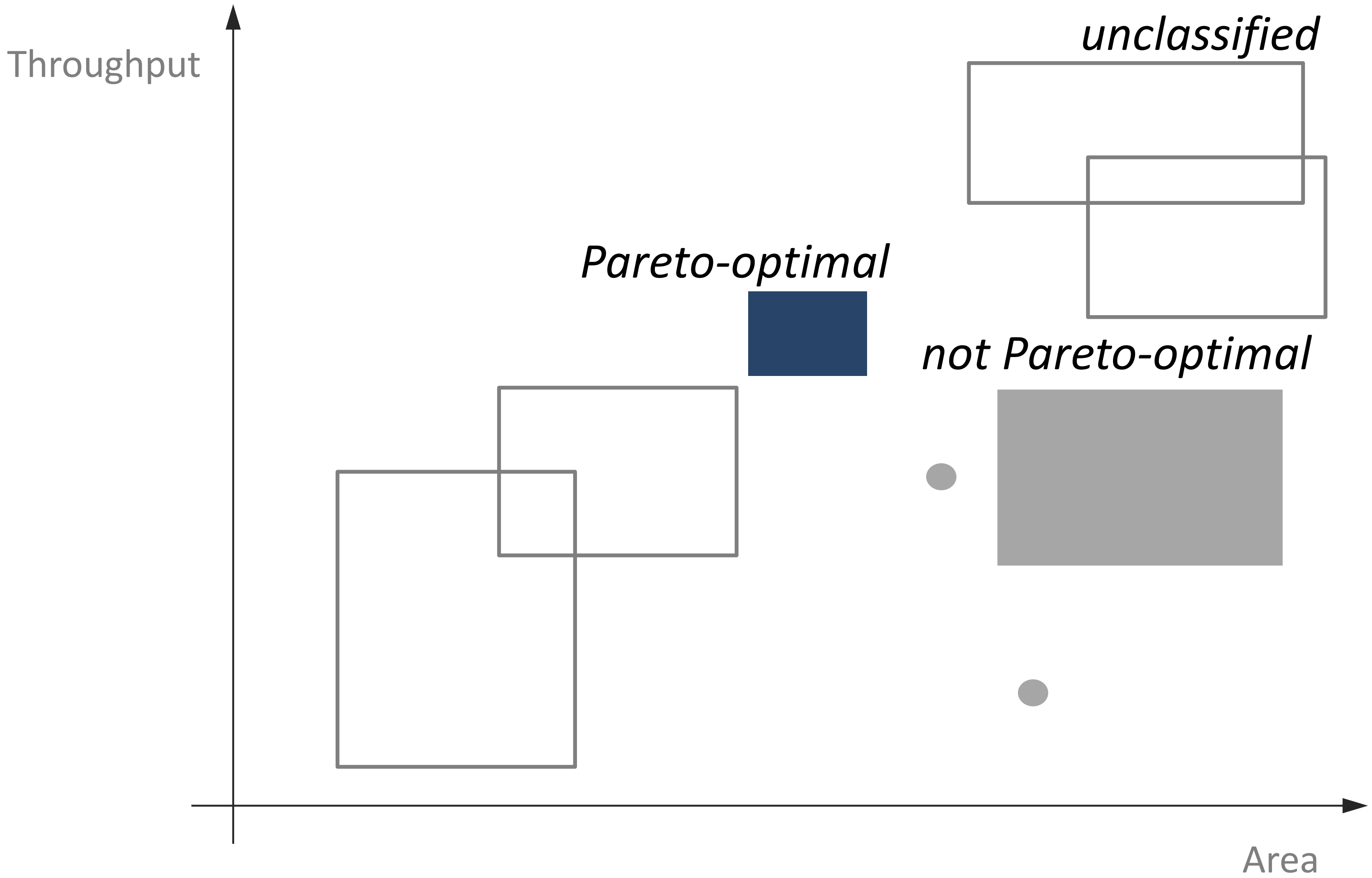
Sample as few designs as possible *to predict* Pareto optimal designs



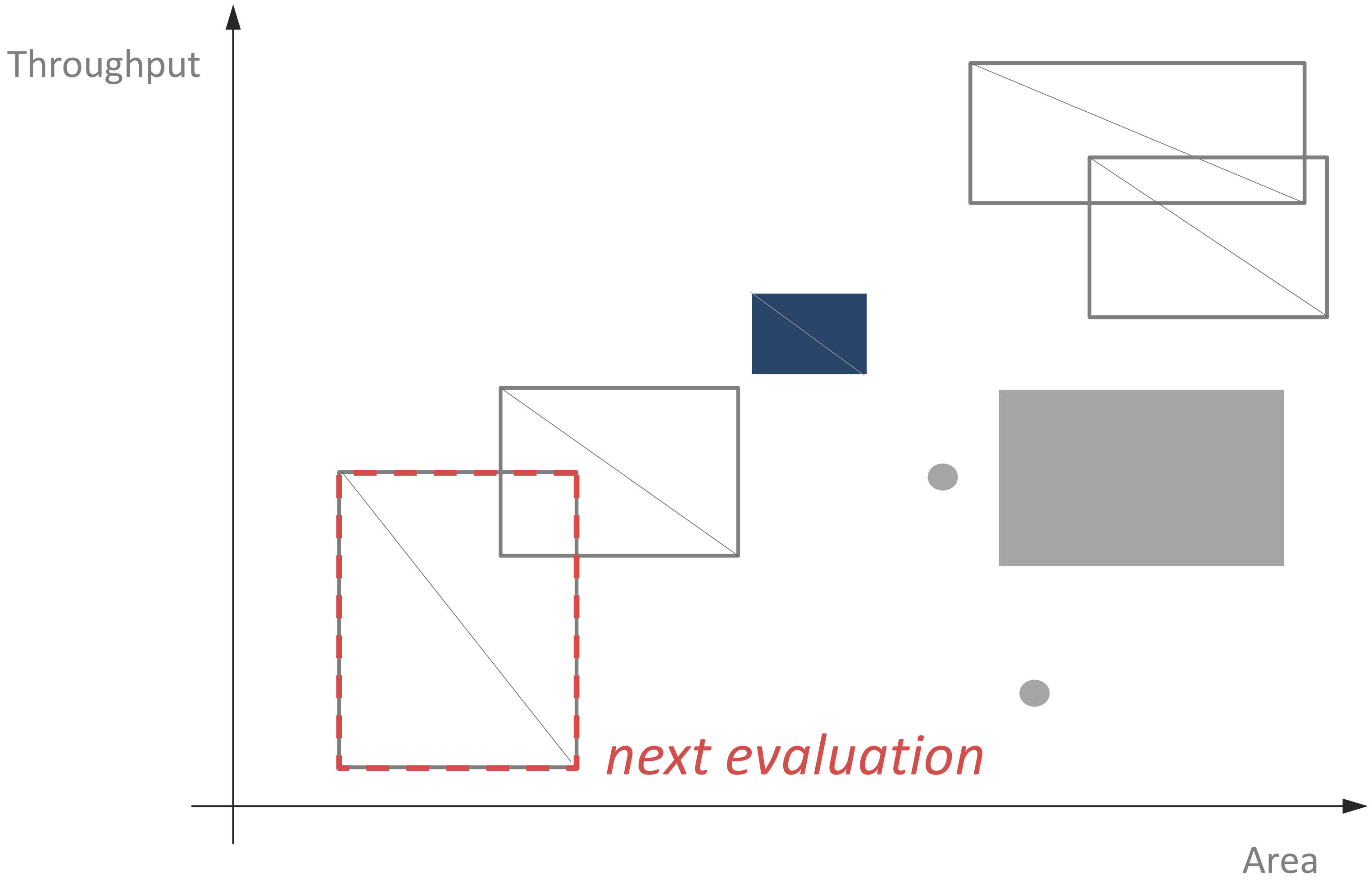
Running the Algorithm: Modeling



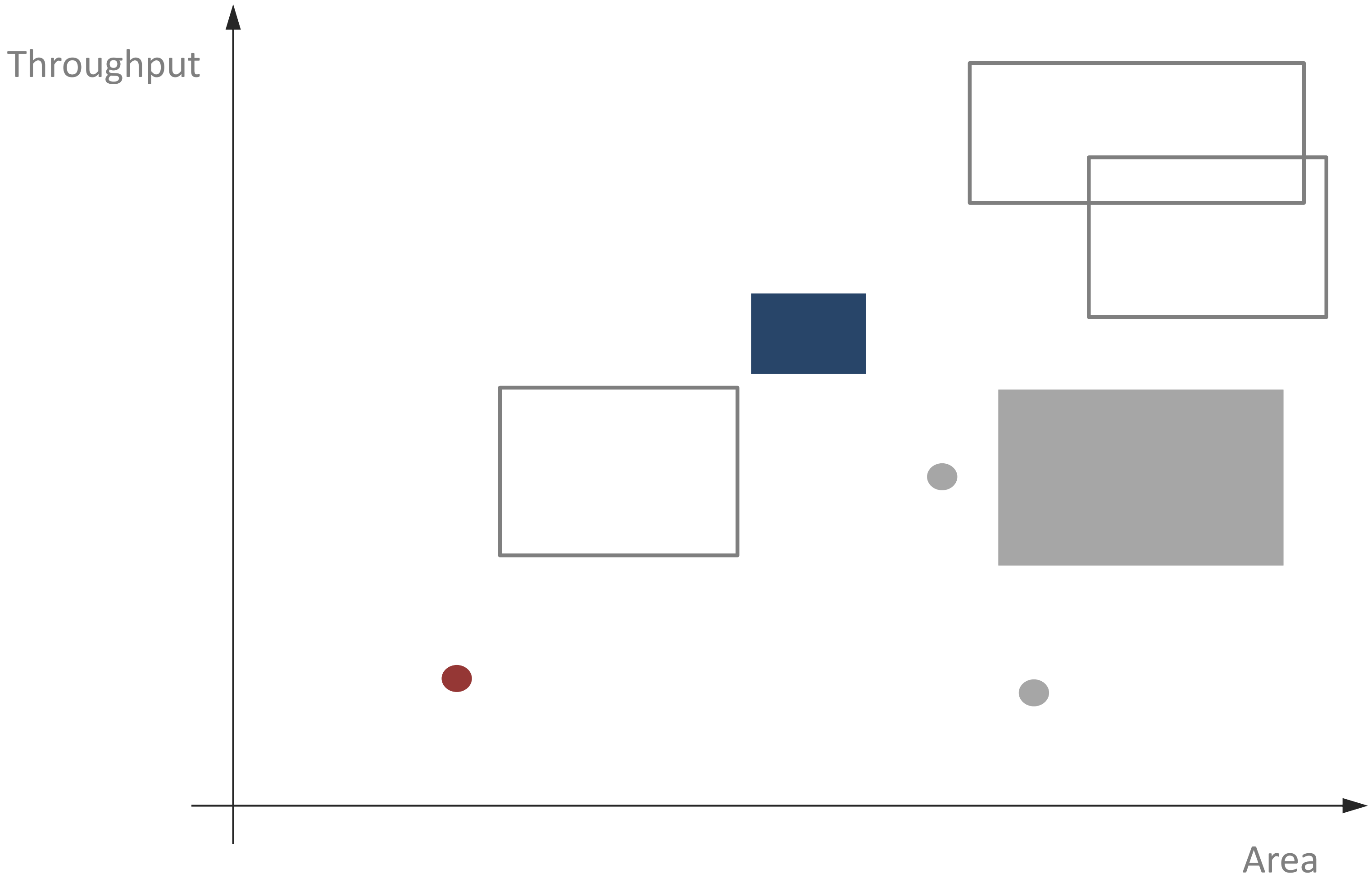
Running the Algorithm: Classification



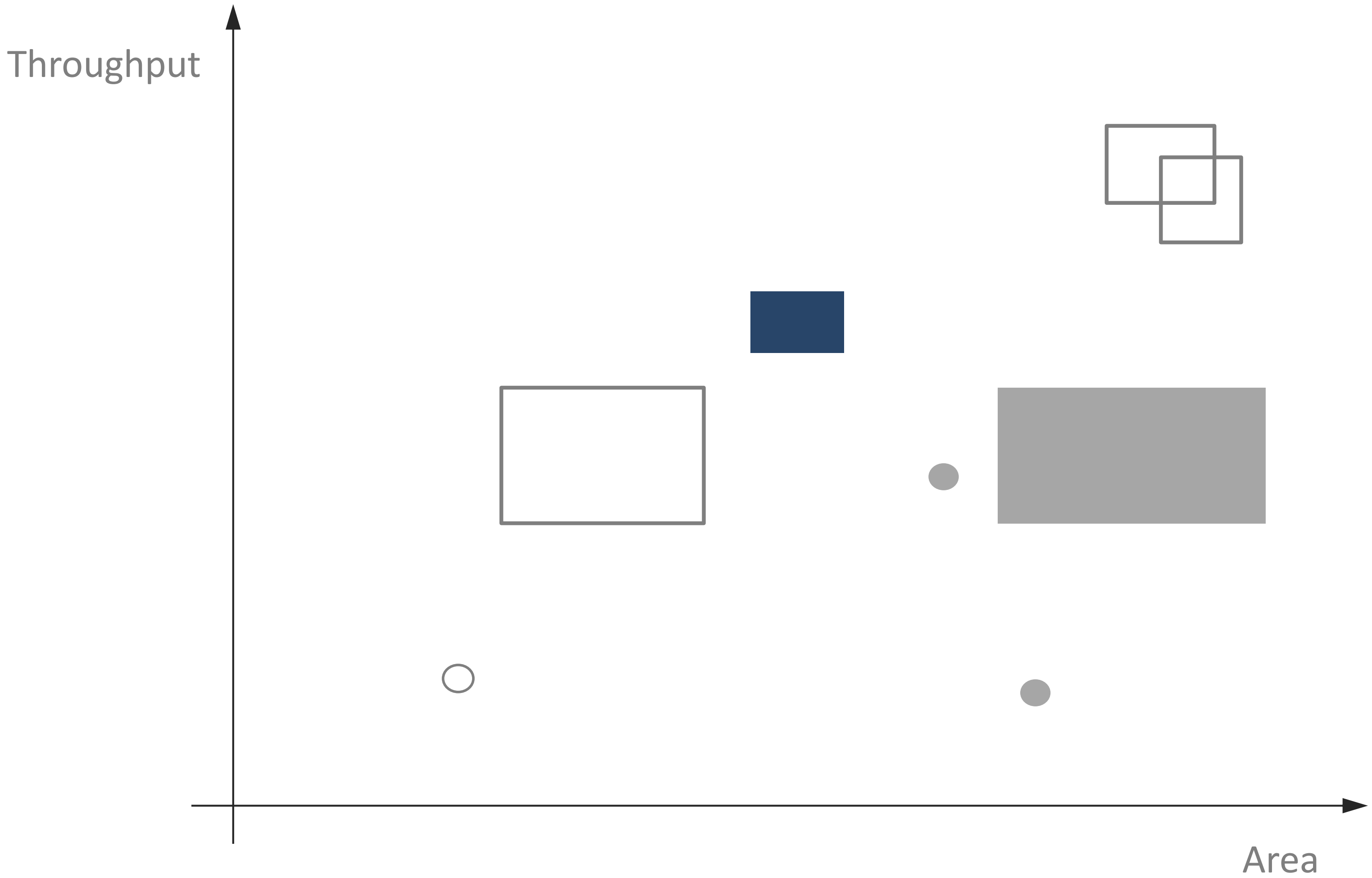
Running the Algorithm: Sampling



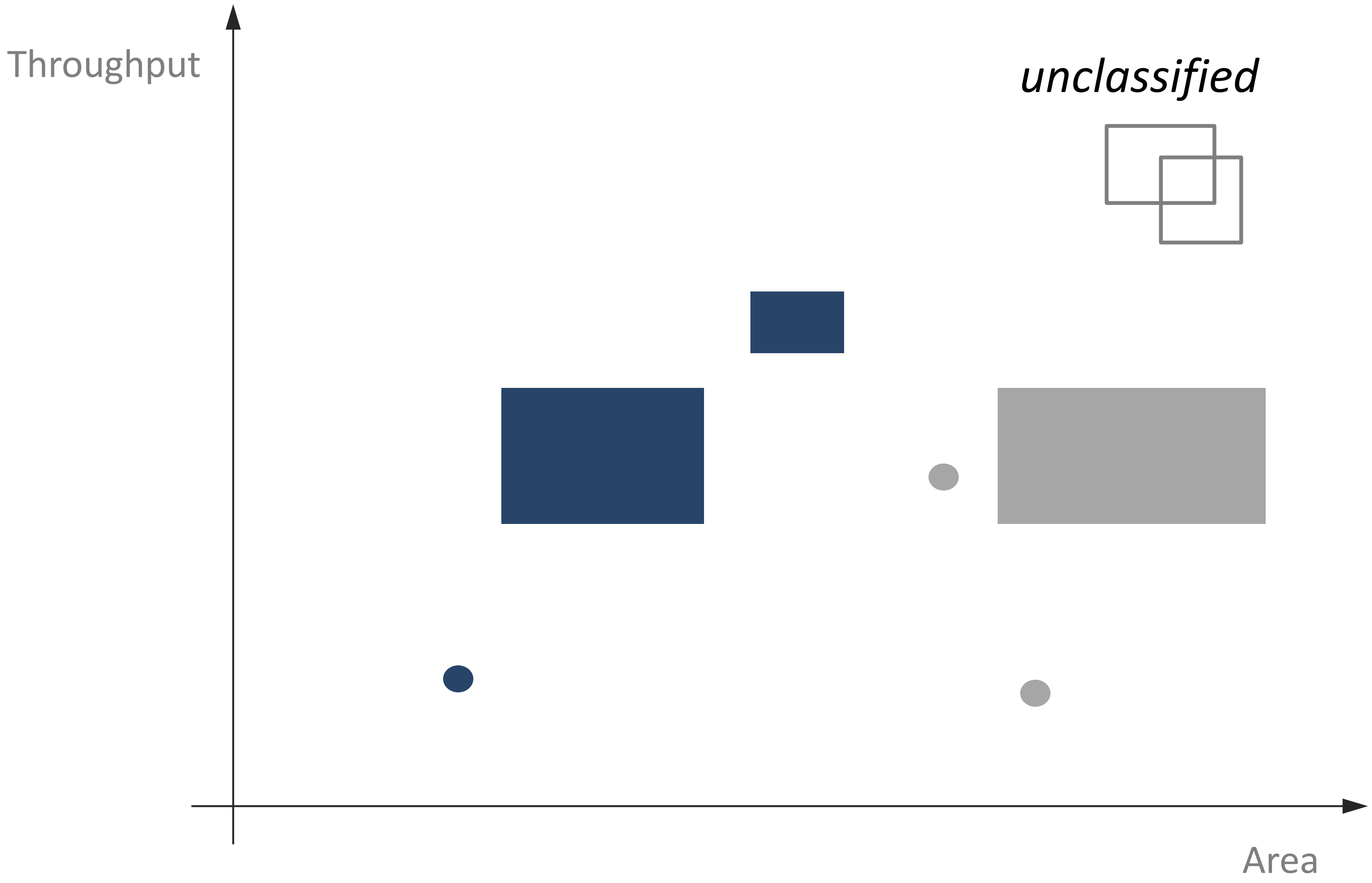
Running the Algorithm: Evaluation



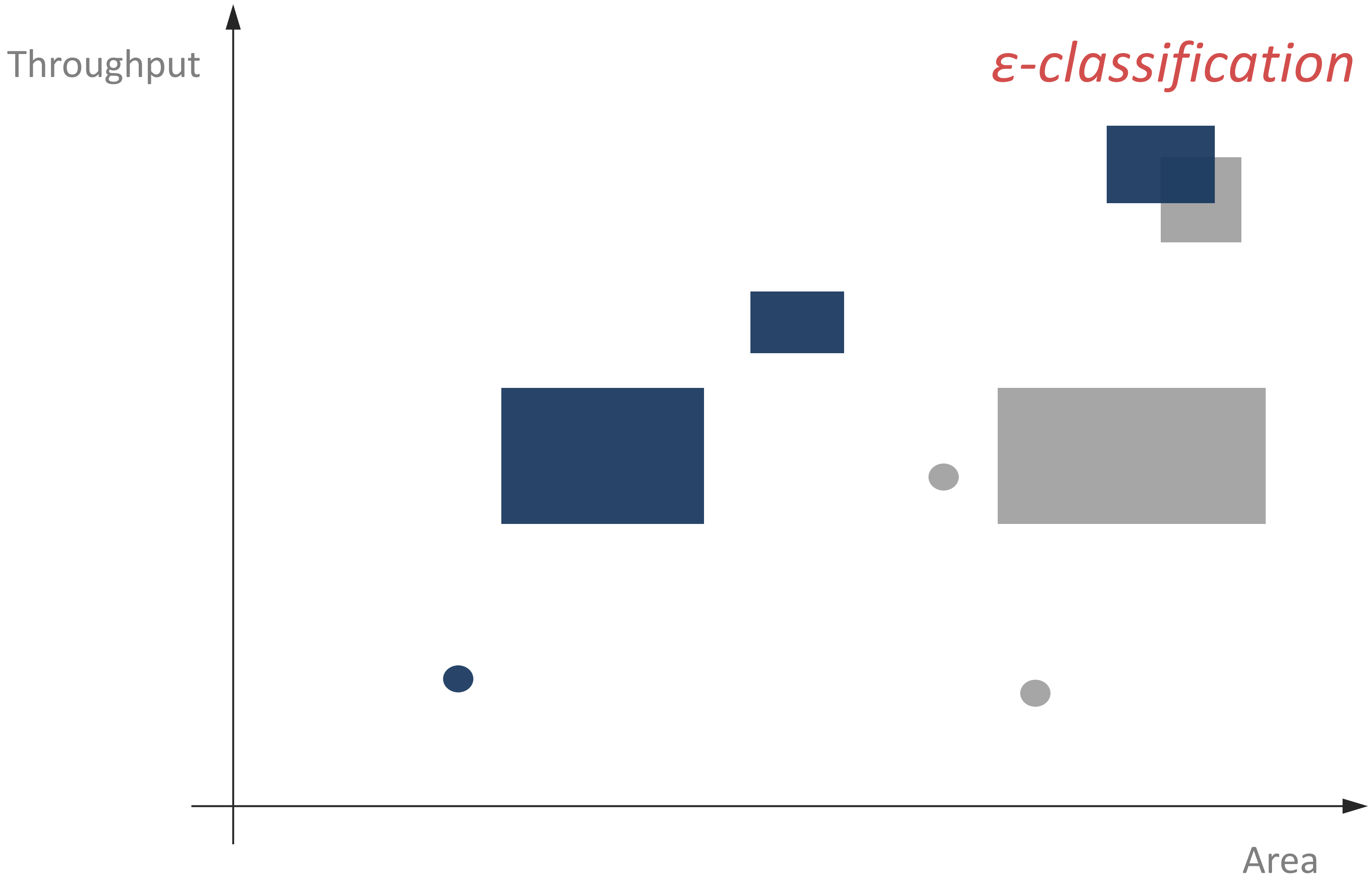
Running the Algorithm: Modeling



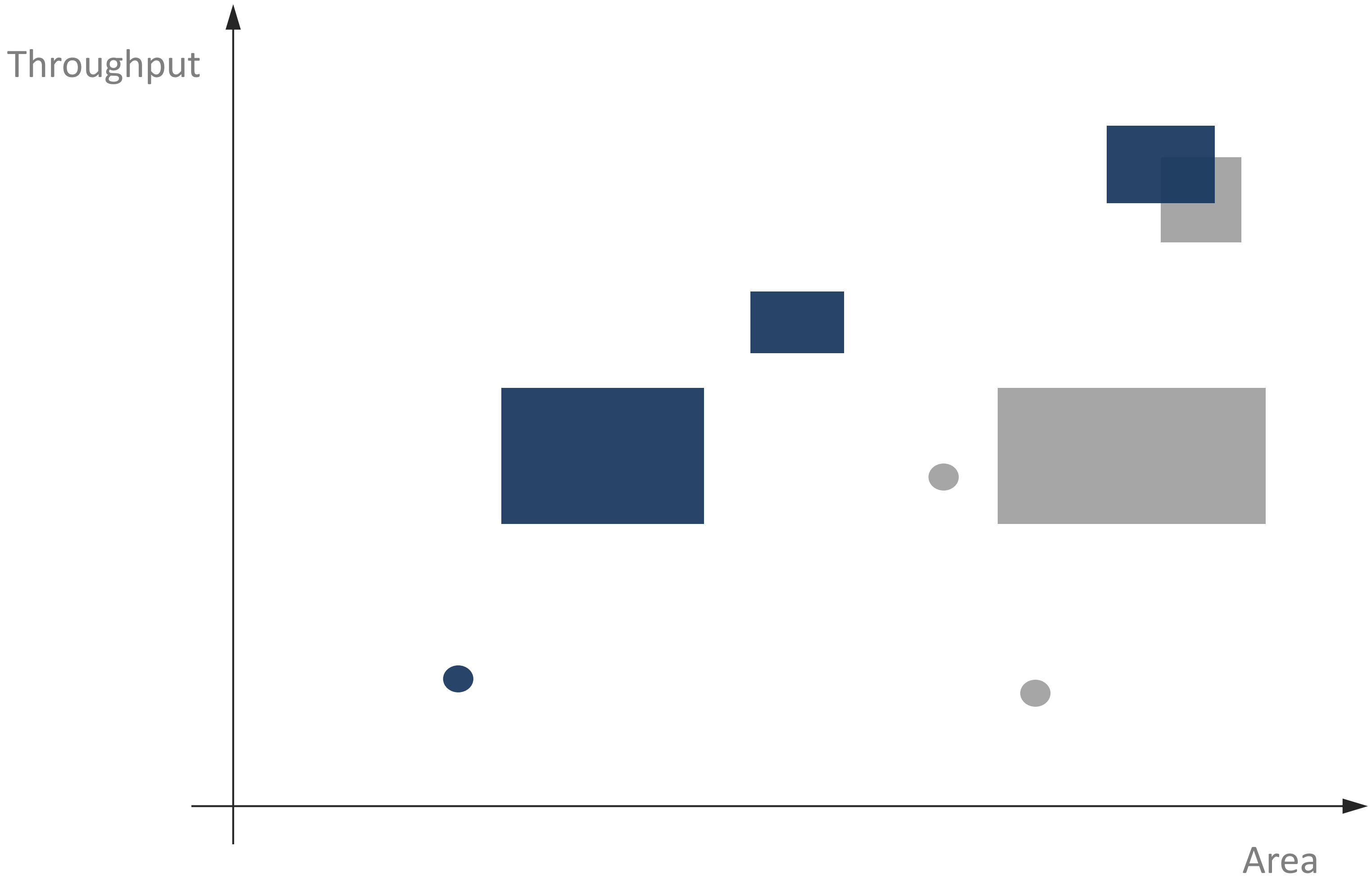
Running the Algorithm: Classification



Running the Algorithm: Tolerances



Running the Algorithm: Termination



The PAL Algorithm

Input: design space E; GP prior $\mu_{0,i}, \sigma_0, k_i$ for all $1 \leq i \leq n$; ϵ; β_t for $t \in \mathbb{N}$

Output: predicted-Pareto set $\hat{P}$

$P_0 = \emptyset, N_0 = \emptyset, U_0 = E$ {classification sets}

$S_0 = \emptyset$ {evaluated set}

$R_0(\boldsymbol{x}) = \mathbb{R}^n$ for all $\boldsymbol{x} \in E$

$t = 0$

repeat

— *Modeling* —

Obtain $\boldsymbol{\mu}_t(\boldsymbol{x})$ and $\boldsymbol{\sigma}_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in E$ $\{\boldsymbol{\mu}_t(\boldsymbol{x}) = \boldsymbol{y}(\boldsymbol{x})$ and $\boldsymbol{\sigma}_t(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in S_t\}$

$R_t(\boldsymbol{x}) = R_{t-1}(\boldsymbol{x}) \cap Q_{\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t, \beta_{t+1}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in E$

— *Classification* —

$P_t = P_{t-1}, \; N_t = N_{t-1}, \; U_t = U_{t-1}$

for all $\boldsymbol{x} \in U_t$ **do**

if there is no $\boldsymbol{x}' \neq \boldsymbol{x}$ such that $\min(R_t(\boldsymbol{x})) + \boldsymbol{\epsilon} \preceq \max(R_t(\boldsymbol{x}')) - \boldsymbol{\epsilon}$ **then**

$P_t = P_t \cup \{\boldsymbol{x}\}, \; U_t = U_t \setminus \{\boldsymbol{x}\}$

else if there exists $\boldsymbol{x}' \neq \boldsymbol{x}$ such that $\max(R_t(\boldsymbol{x})) - \boldsymbol{\epsilon} \preceq \max(R_t(\boldsymbol{x}')) + \boldsymbol{\epsilon}$ **then**

$N_t = N_t \cup \{\boldsymbol{x}\}, \; U_t = U_t \setminus \{\boldsymbol{x}\}$

end if

end for

— *Sampling* —

Find $w_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in (U_t \cup P_t) \setminus S_t$

Choose $\boldsymbol{x}_{t+1} = \arg\max_{x \in (U_t \cup P_t) \setminus S_t} \{w_t(\boldsymbol{x})\}$

$t = t + 1$

Sample $\boldsymbol{y}_t(\boldsymbol{x}_t) = \boldsymbol{f}(\boldsymbol{x}_t) + \boldsymbol{\nu}_t$

until $U_t = \emptyset$

$\hat{P} = P_t$

Theoretical Guarantee

Given a target error η, PAL is guaranteed to stop in less than T iterations:

Theorem 1. *Let* $\delta \in (0, 1)$. *Running* PAL *with* $\beta_t = 2\log(n|E|\pi^2 t^2/(6\delta))$, *the following holds with probability* $1 - \delta$.
To achieve a maximum hypervolume error of η, *it is sufficient to choose*

$$\epsilon = \frac{\eta(n-1)!}{2na^{n-1}},$$

where $a = \max_{\boldsymbol{x} \in E, 1 \leq i \leq n}\{\sqrt{\beta_1 k_i(\boldsymbol{x}, \boldsymbol{x})}\}$.
In this case, the algorithm terminates after at most T *iterations, where* T *is the smallest number satisfying*

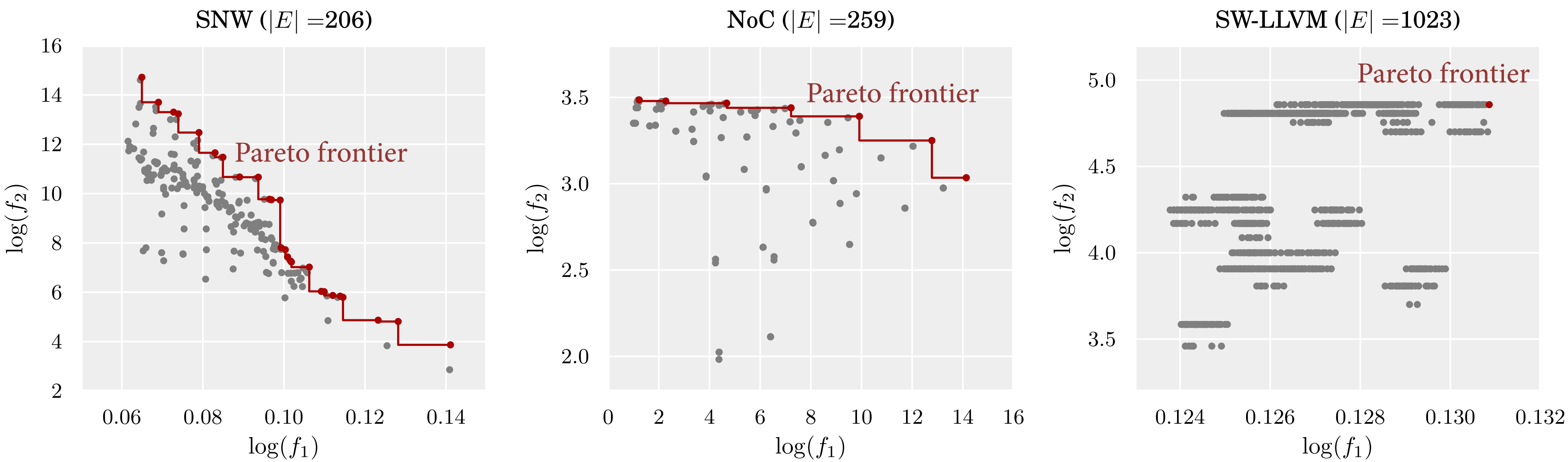
$$\boxed{\sqrt{\frac{T}{C_1 \beta_T \gamma_T}} \geq \frac{na^{n-1}}{\eta(n-1)!}}.$$

Here, $C_1 = 8/\log(1-\sigma^{-2})$, *and* γ_T *depends on the type of kernel used.*

Same γ_T !

Experiments: Data Sets

Sorting networks | *Network on Chip* | *Compiler settings*

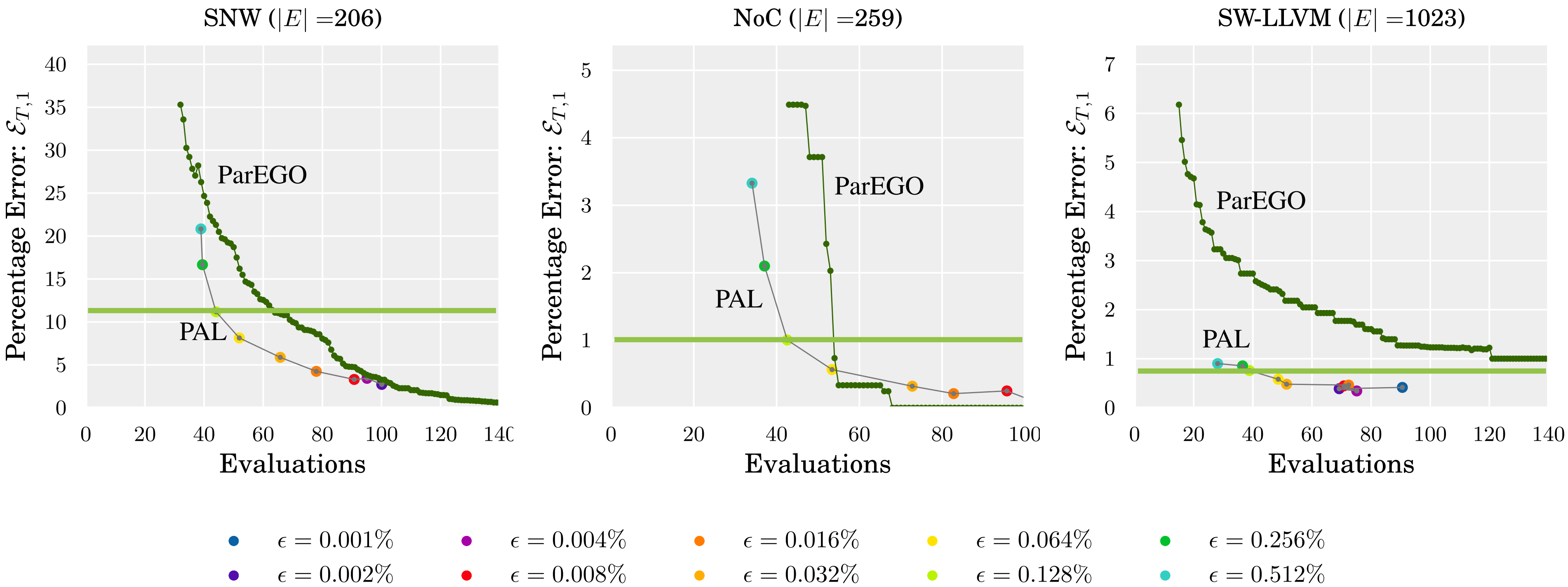


Marcela Zuluaga, Andreas Krause, Peter Milder, Markus Püschel.
Streaming Sorting Networks. LCTES 2012

Oscar Almer, Nigel Topham, Björn Franke.
A Learning- Based Approach to the Automated Design of MP-SoC Networks. ARCS 2011

N. Siegmund, S. Kolesnikov, C. Kastner , S. Apel, D. Batory, M. Rosenmuller, and G. Saake
Predicting Performance via Automated Feature-Interaction Detection. ICSI 2012

Experimental results

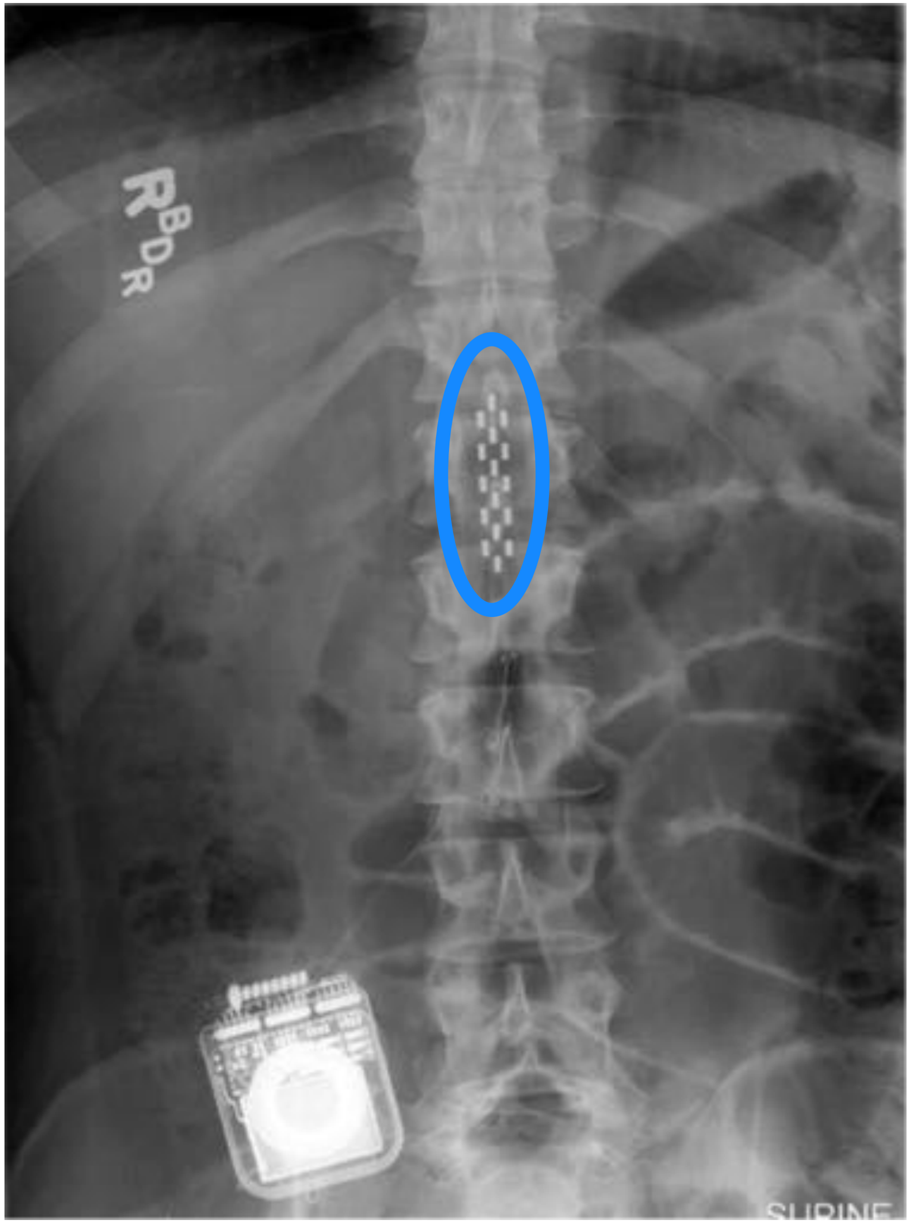


Beyond Basic BO:

Constraints and “Safe” Exploration

Therapeutic Spinal Cord Stimulation

[w Sui, Gotovos, Burdick '15; w Desautels, Burdick '14]



[S. Harkema,
The Lancet, Elsevier]

Safe Controller Tuning

[with Berkenkamp, Schoellig ICRA '16]

Tuning the Swiss Free Electron Laser

[with Kirschner, Mutny, Ischebeck et al ’18]

PAUL SCHERRER INSTITUT

PSI

Illustration

Tracking performance	$\max_{\theta} f(\theta)$	**Few experiments**
Safety constraint	$g(\theta) \geq 0$	**Safety for all experiments**

Safe Bayesian Optimization

θ_t

(Noisy) Reward

$$f(\theta_t) + \epsilon_t^f$$

$$g(\theta_t) + \epsilon_t^g$$

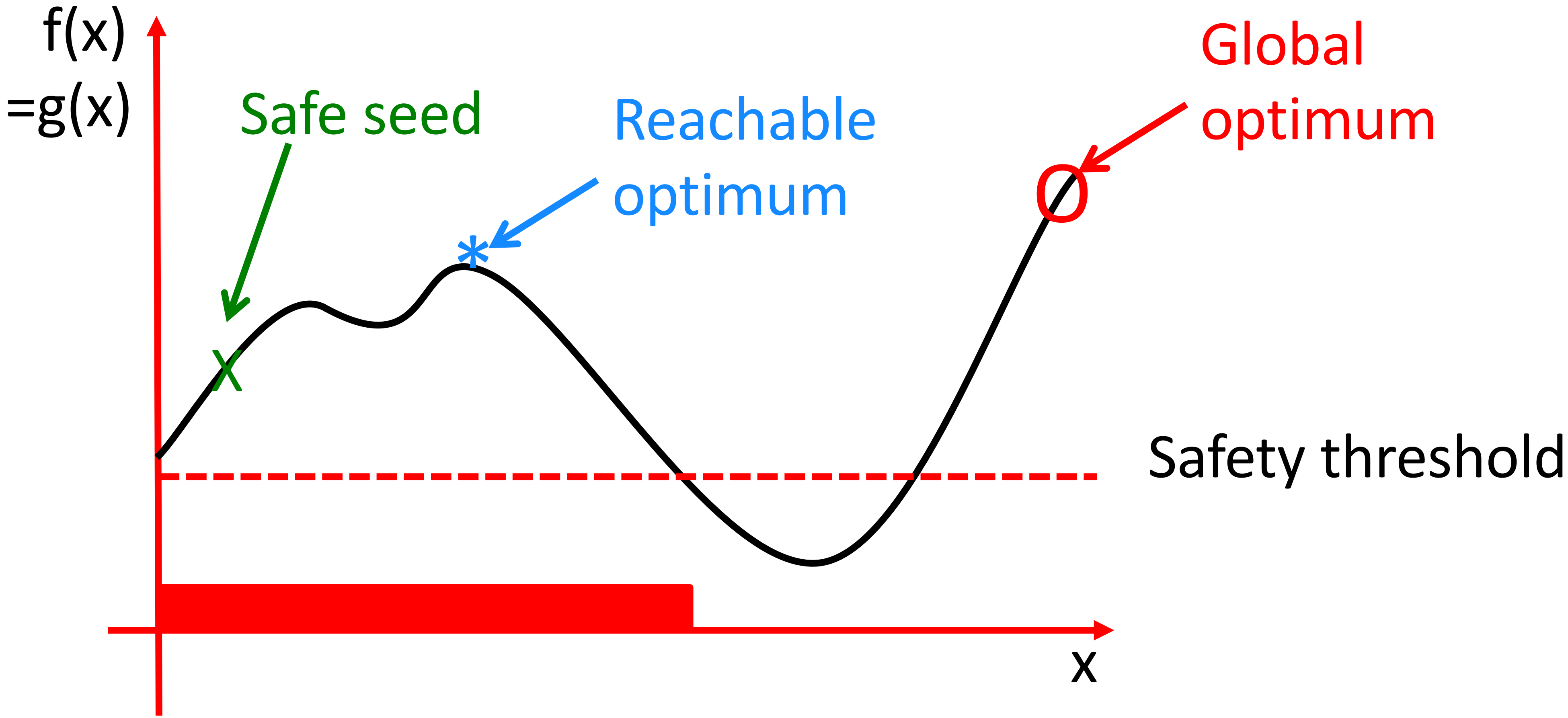
(Noisy) Constraint

UNKNOWN!

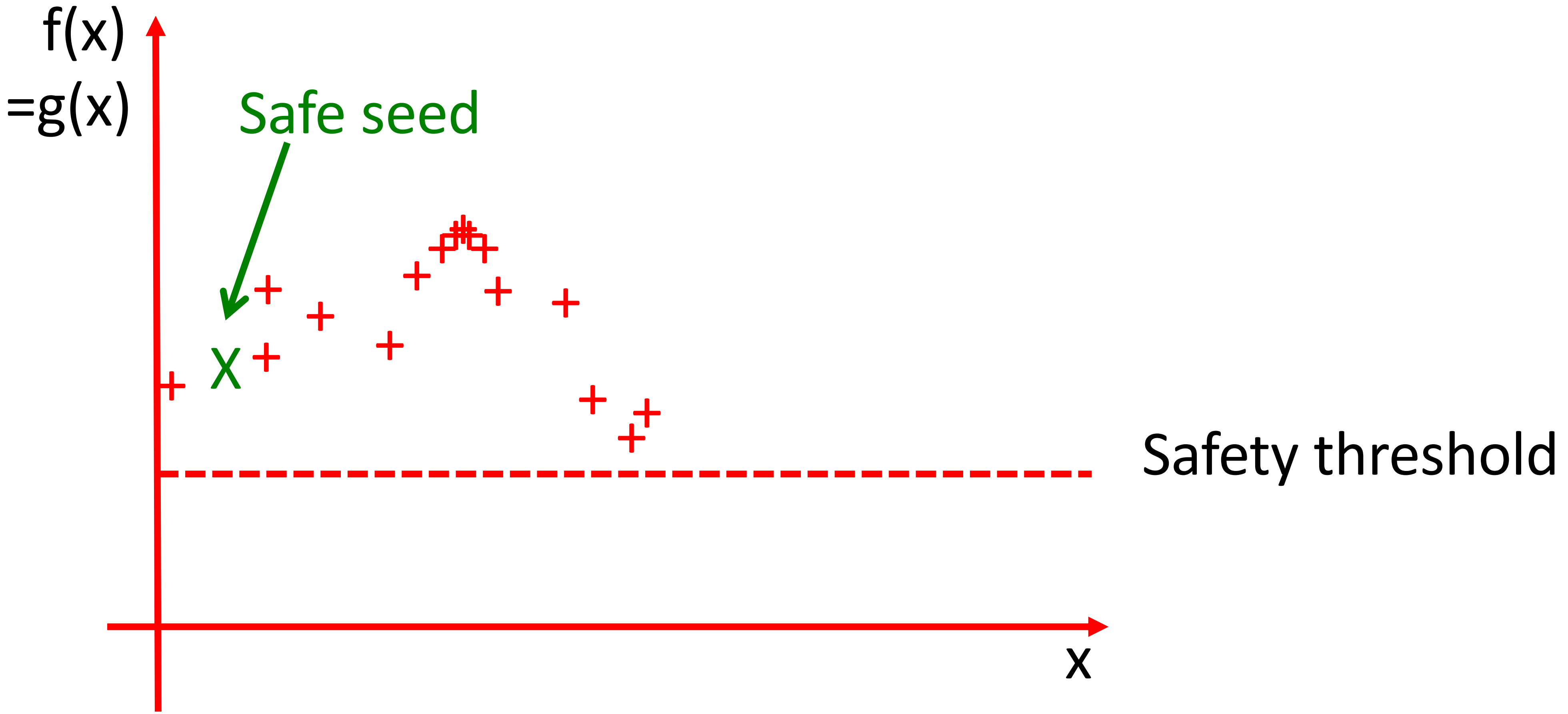
Goal: $$\max_{\theta} f(\theta) \text{ s.t. } g(\theta) \geq 0$$

Safety: $$g(\theta_t) \geq 0 \text{ for all } t$$

Safe optimization



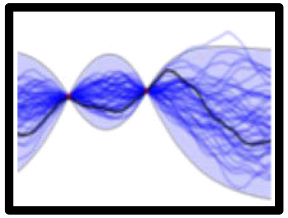
Safe optimization



Starting point: Bayesian Optimization

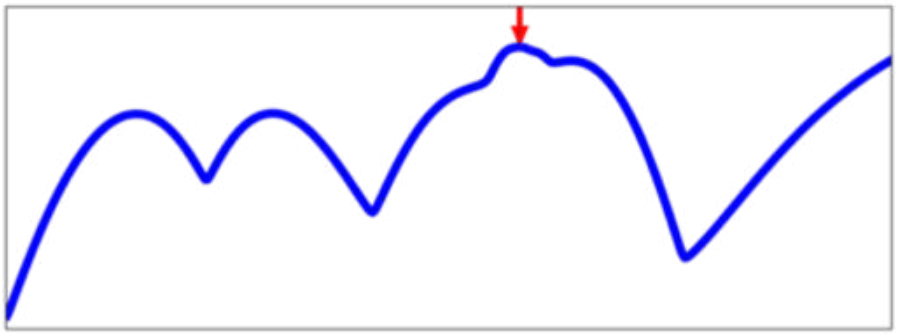
[Močkus '75]

x_t →



→ $y_t = f(x_t) + \epsilon_t$

Acquisition function



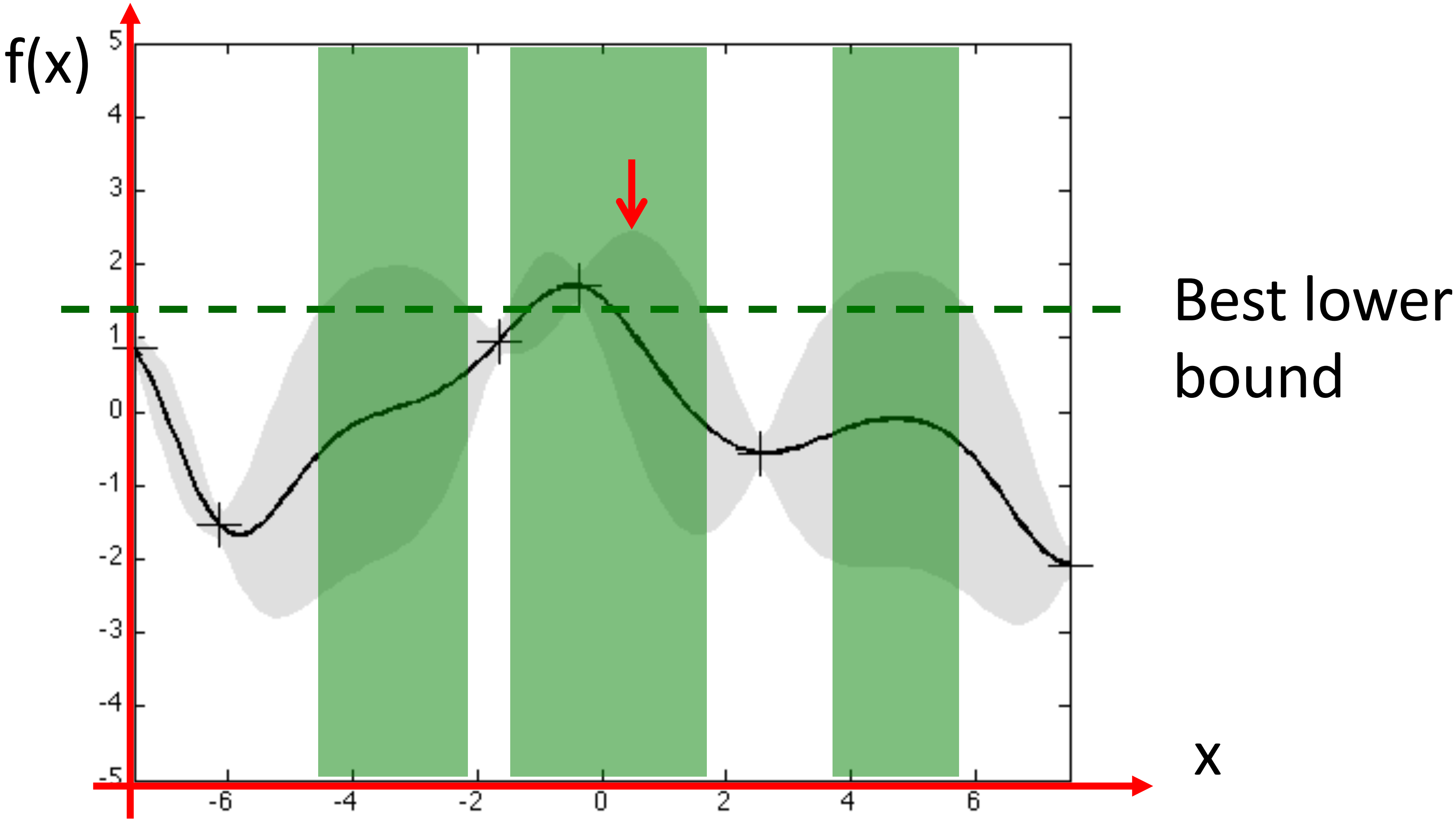
Unconstrained

Expected/most prob. improvement [Močkus *et al.* '78,'89], Information gain about maximum [Villemonteix *et al.* '09], Knowledge gradient [Powell *et al.* '10], Predictive Entropy Search [Hernández-Lobato *et al.* '14], TruVaR [Bogunovic et al'17], Max Value Entropy Search [Wang et al'17]

Constraints / Multiple Objectives

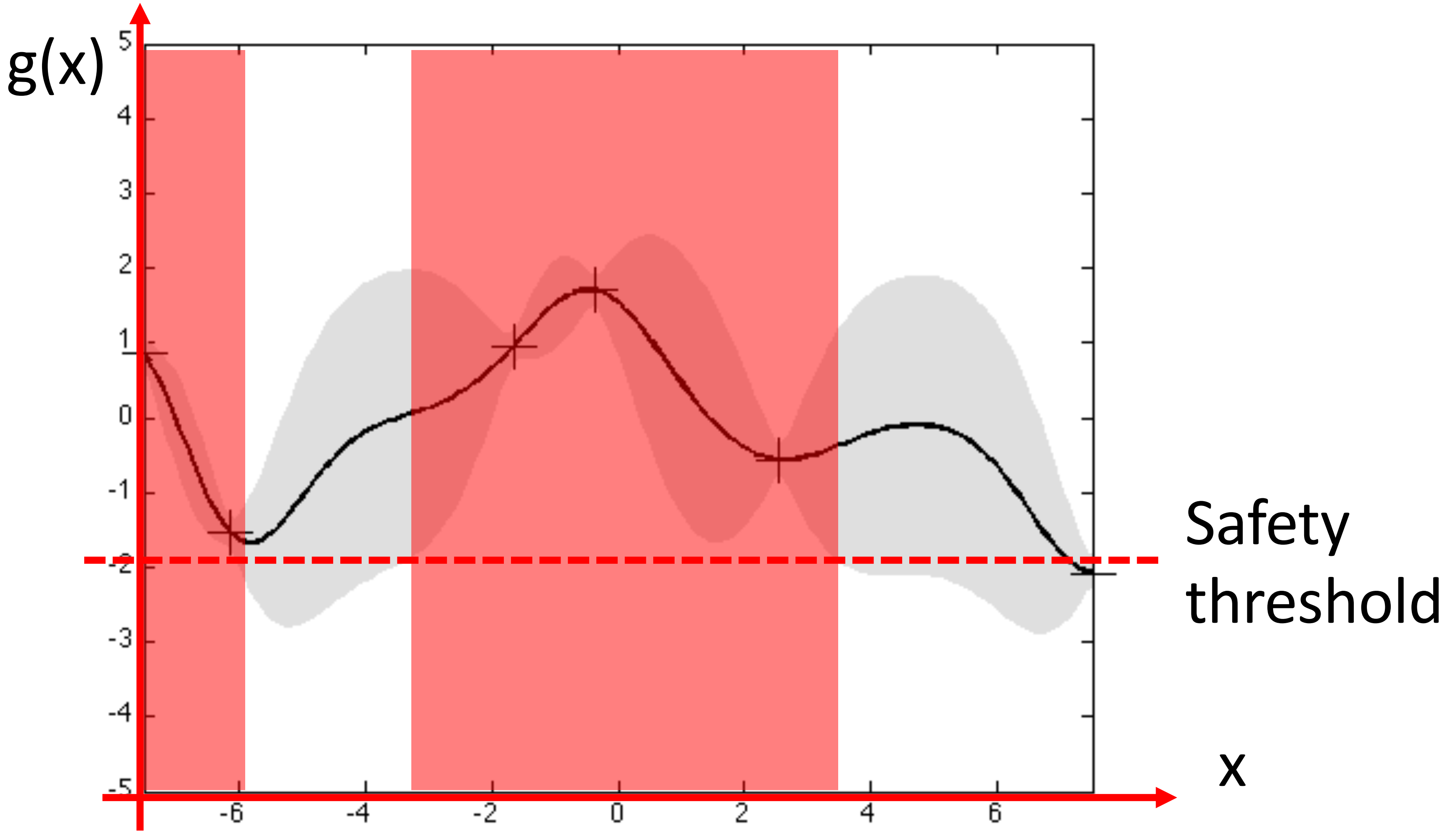
[Snoek *et al.* '13, Gelbart *et al.* '14, Gardner *et al.* '14, Zuluaga et al. '16]

Plausible maximizers



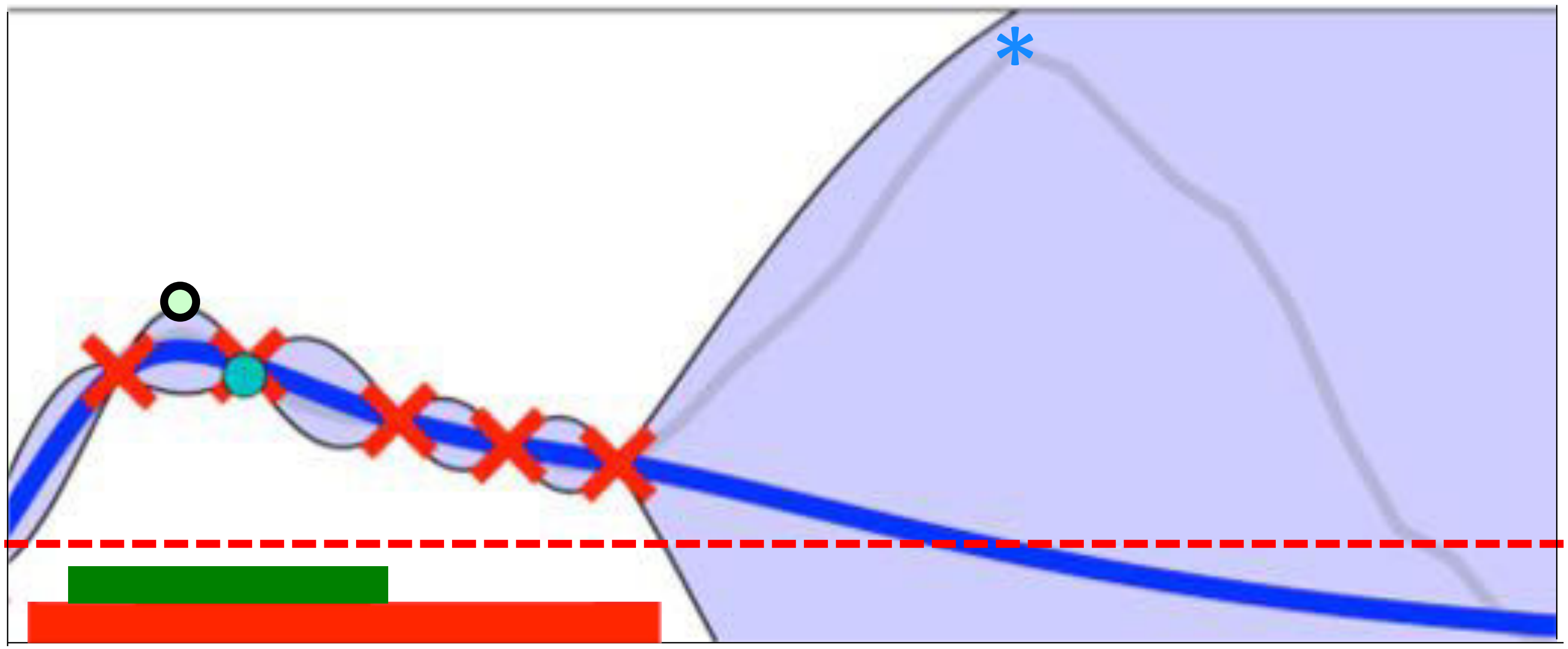
Focus exploration where
upper confidence bound ≥ best lower bound!

Certifying Safety



Statistically certify safety where lower bound > threshold!

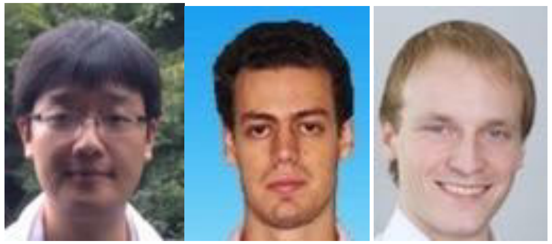
First Attempt: SafeUCB



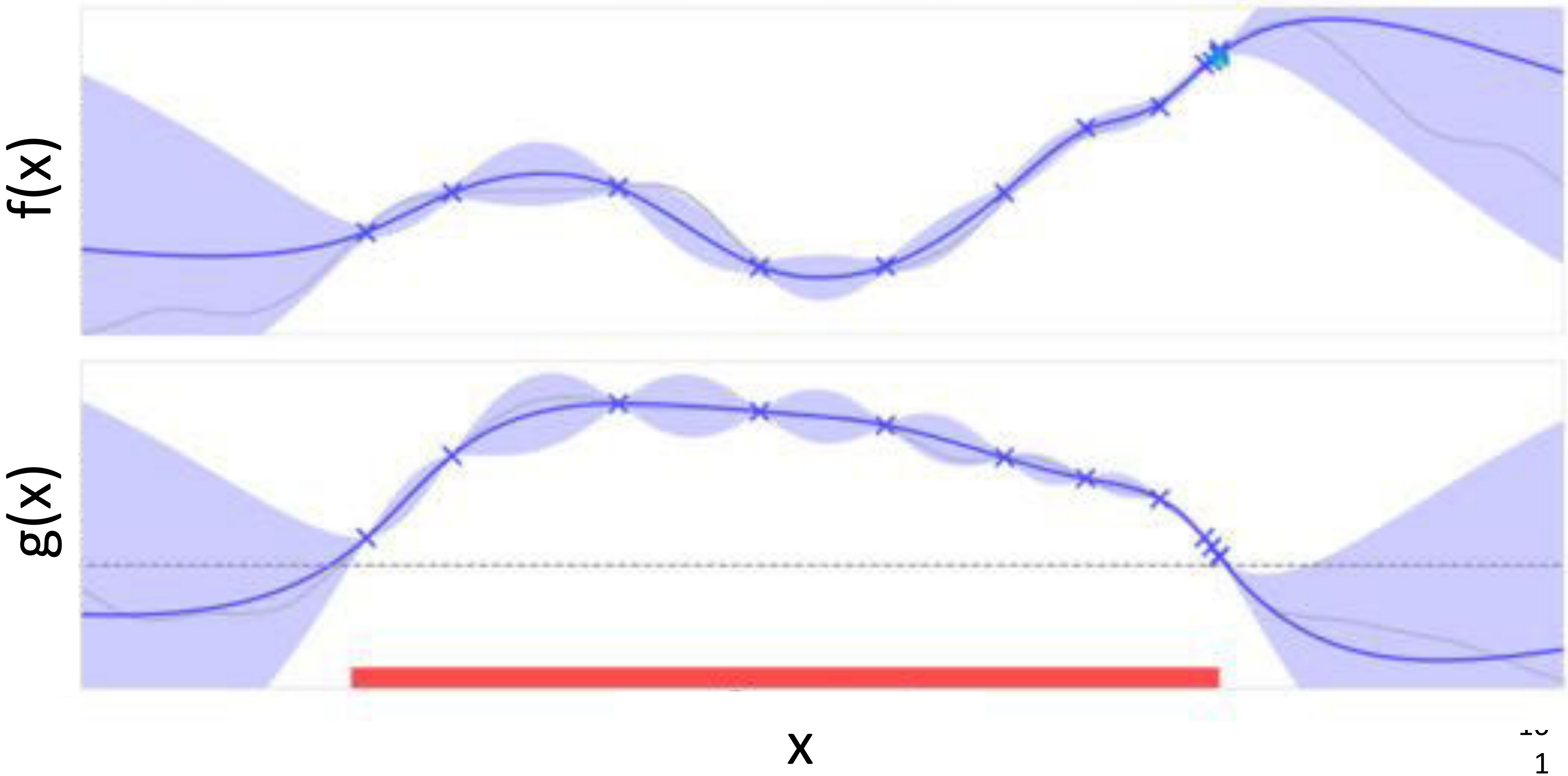
Maximize acquisition function (GP-UCB, EI, …) over certified safe domain

➔ Gets stuck in local optima!

SafeOpt



[Sui, Gotovos, Burdick, K ICML'15], [Berkenkamp, Schoellig K'16]



SafeOpt Guarantees

[with Sui, Gotovos, Burdick ICML'15; Berkenkamp Schoellig K'16]

Theorem (informal):

Under suitable conditions on the kernel and on *f,g*, there exists a function *T(ε,δ)* such that for any *ε>0* and *δ>0*, it holds with probability at least *1-δ* that

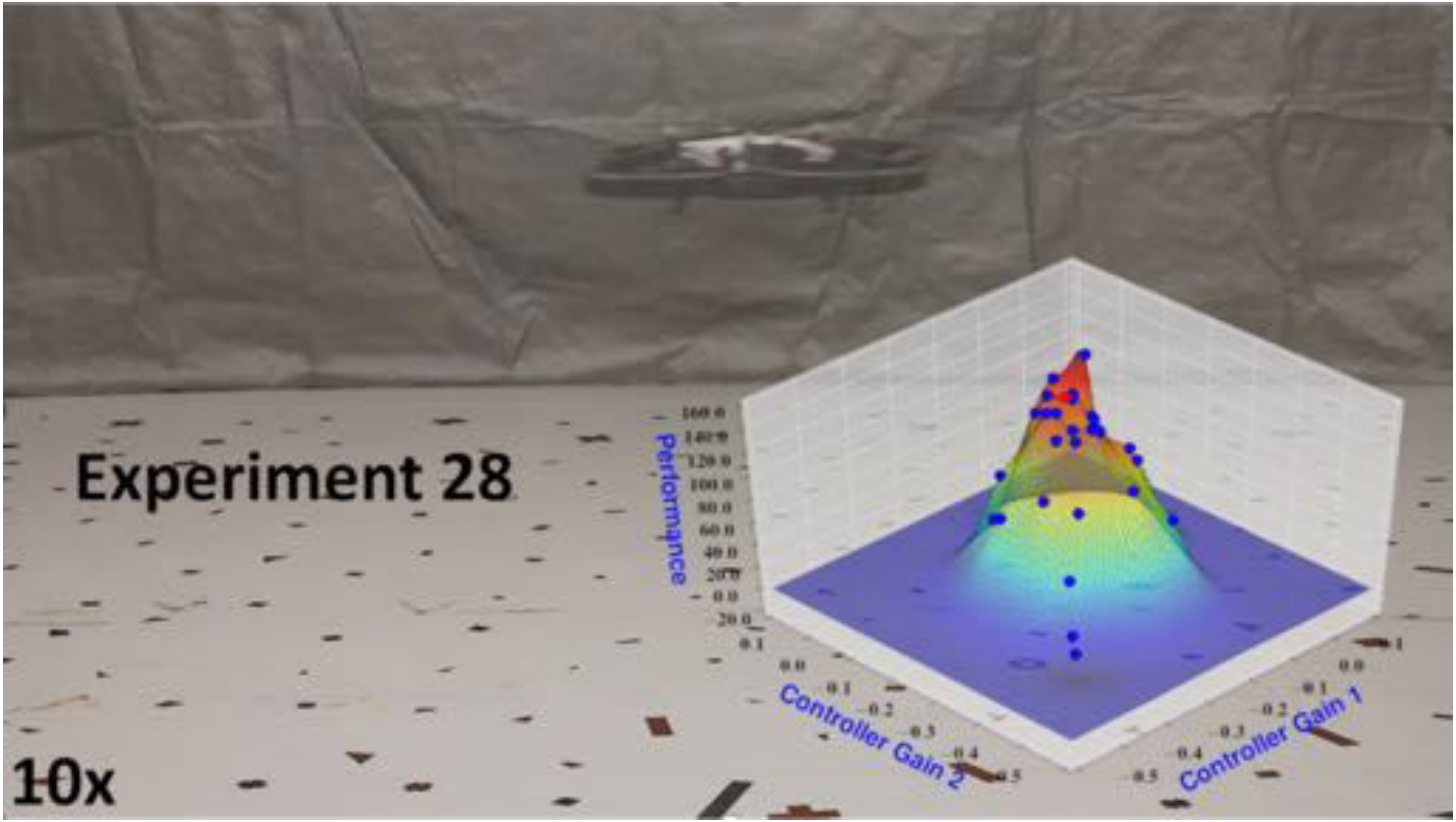
1) SafeOpt never makes an unsafe decision
2) After at most *T(ε,δ)* iterations, it found an ε-optimal reachable point

For Gaussian kernel:
(fixed domain & dim.)

$$T(\varepsilon, \delta) \in \tilde{O}\left((||f||_k + ||g||_k) \frac{\log^3 1/\delta}{\varepsilon^2} \right)$$

Safe controller tuning

[with Berkenkamp, Schoellig ICRA '16]



Transfer learning / handling context

[cf K & Ong NIPS‘11; Berkenkamp, Schöllig, K ‘16]

unknown

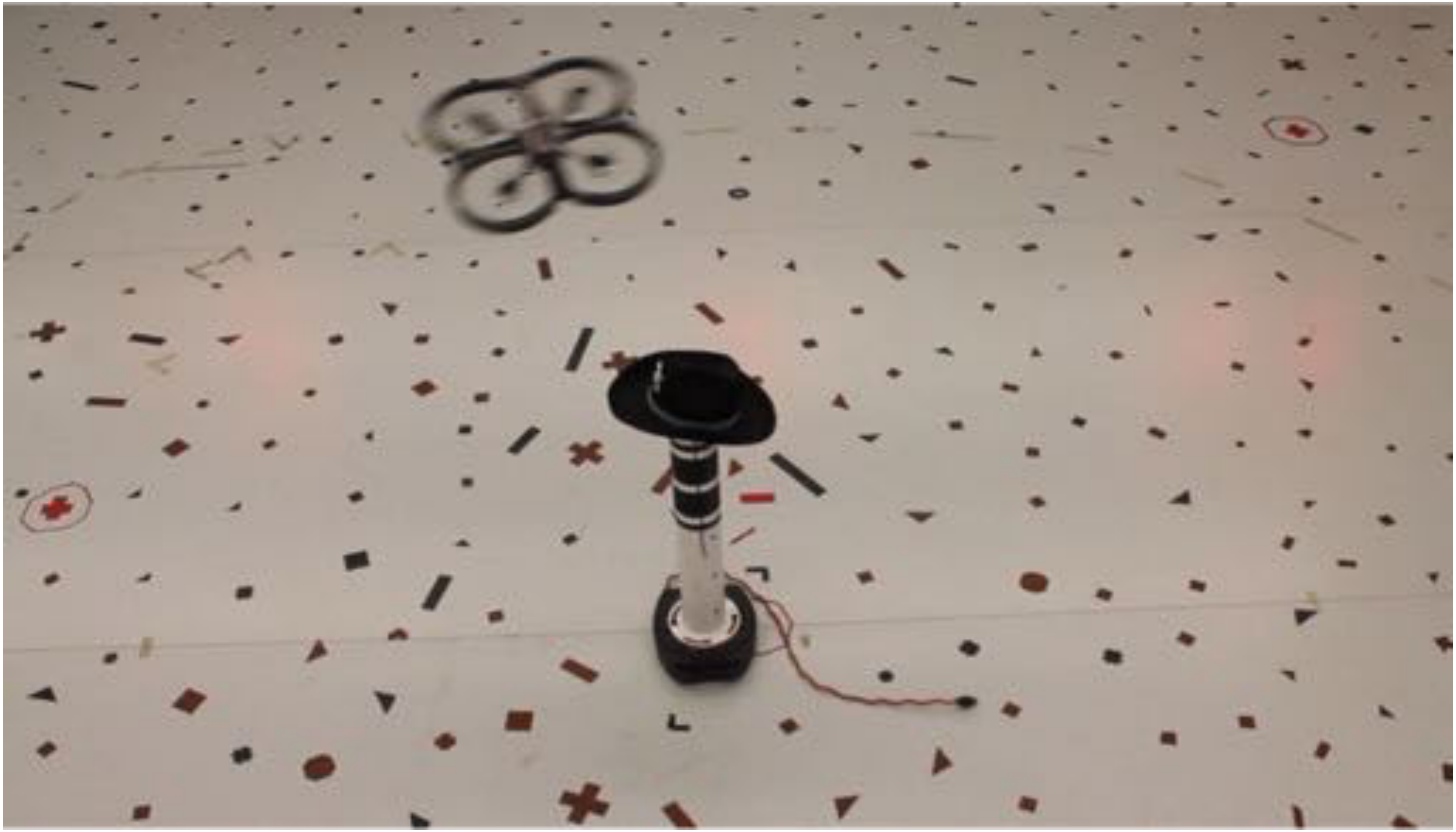
$$\max_{\mathbf{x} \in D} f(\mathbf{x},\quad) \text{ s.t. } g_i(\mathbf{x},\quad) \geq 0 \text{ for } i \in \{1, \ldots, m\}$$

x: chosen by algorithm

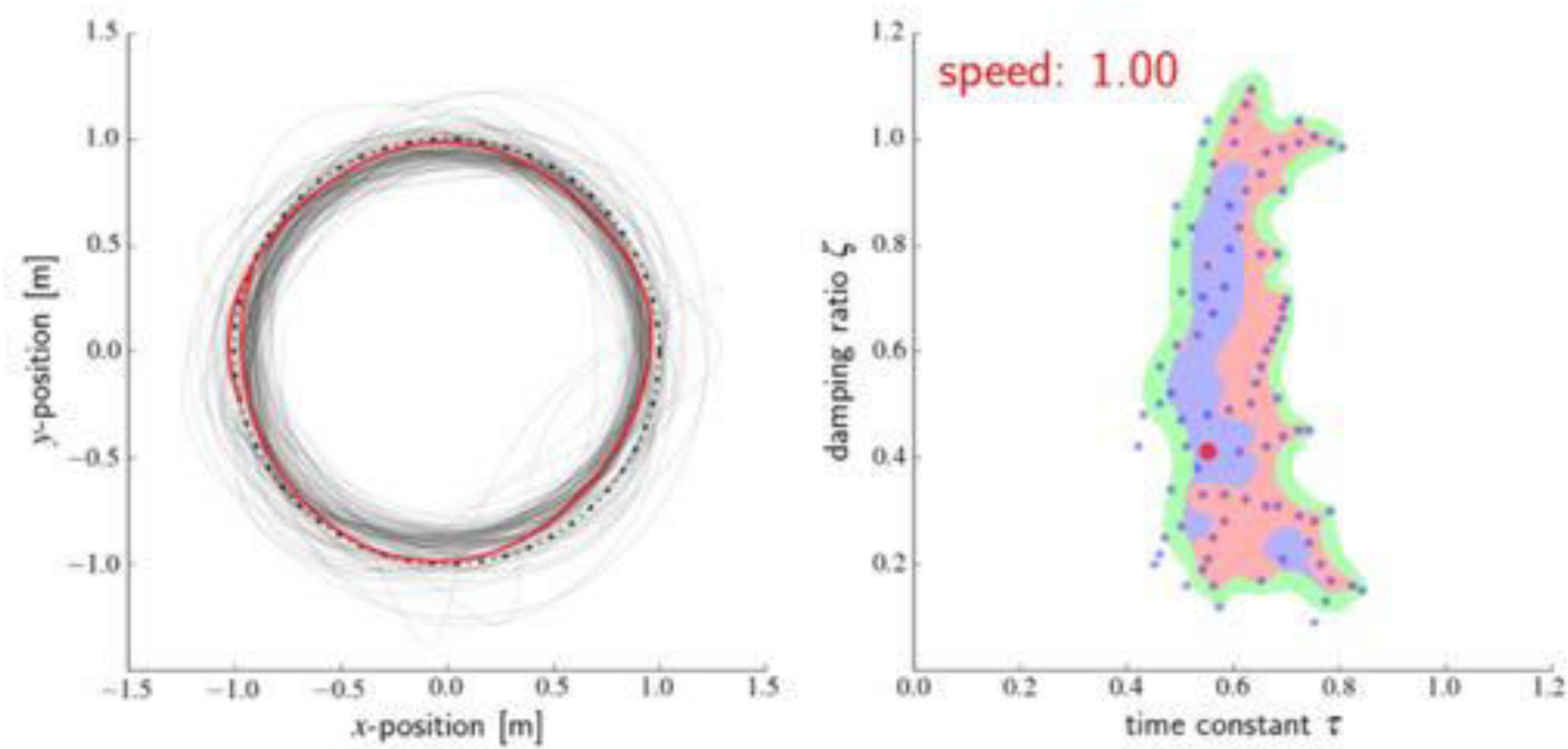
z: given as input (context)

Optimization at 1 m/s

[with Berkenkamp, Schoellig ICRA ’16]

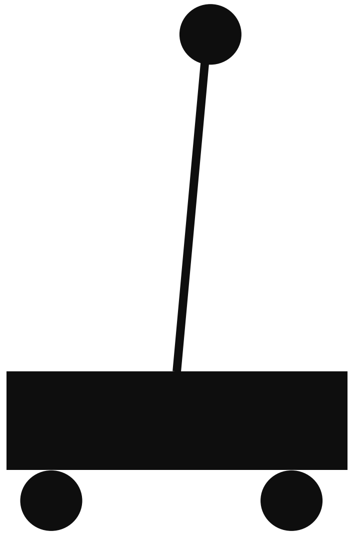


Knowledge transfer to higher speeds



Exploration: Virtual vs Physical

[w Marco, Berkenkamp, Hennig, Schöllig, Schaal, Trimpe, ICRA'17]



Expensive, but accurate

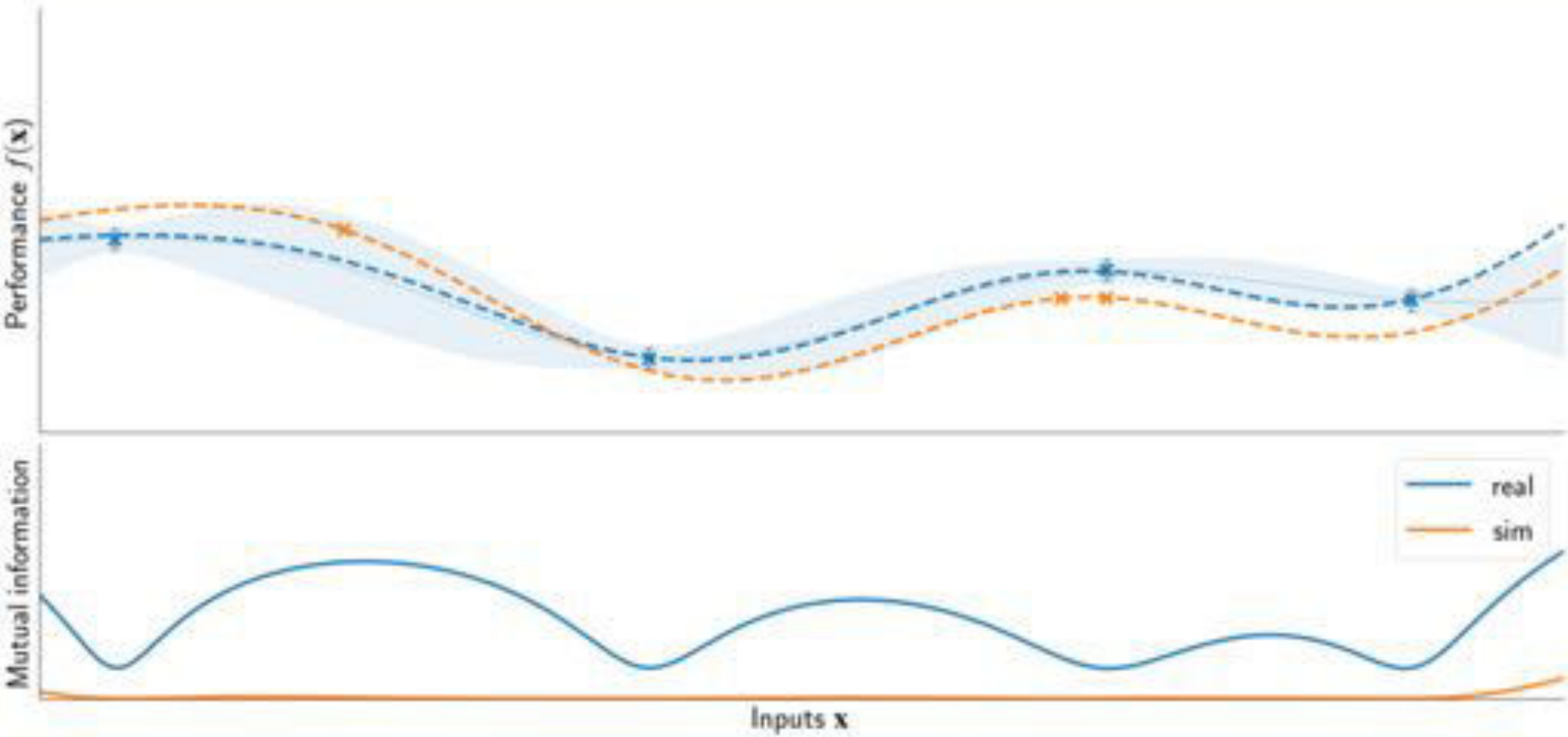
$$f(x)$$

Cheap, but biased

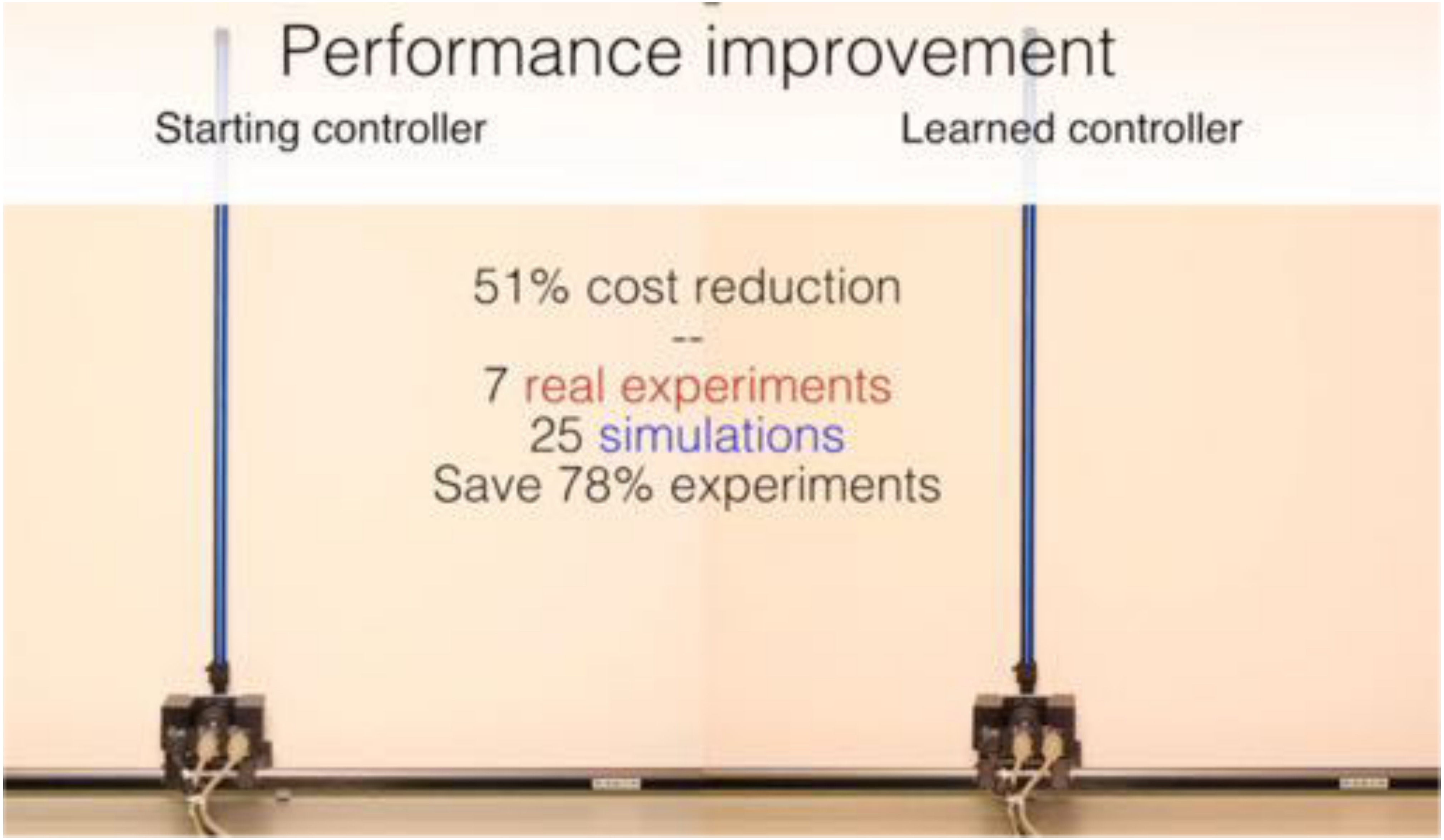
$$\hat{f}(x) = f(x) + \delta(x)$$

Multiple sources of information

$$\hat{f}(x) = f(x) + \delta(x)$$



Exploration: Virtual vs Physical

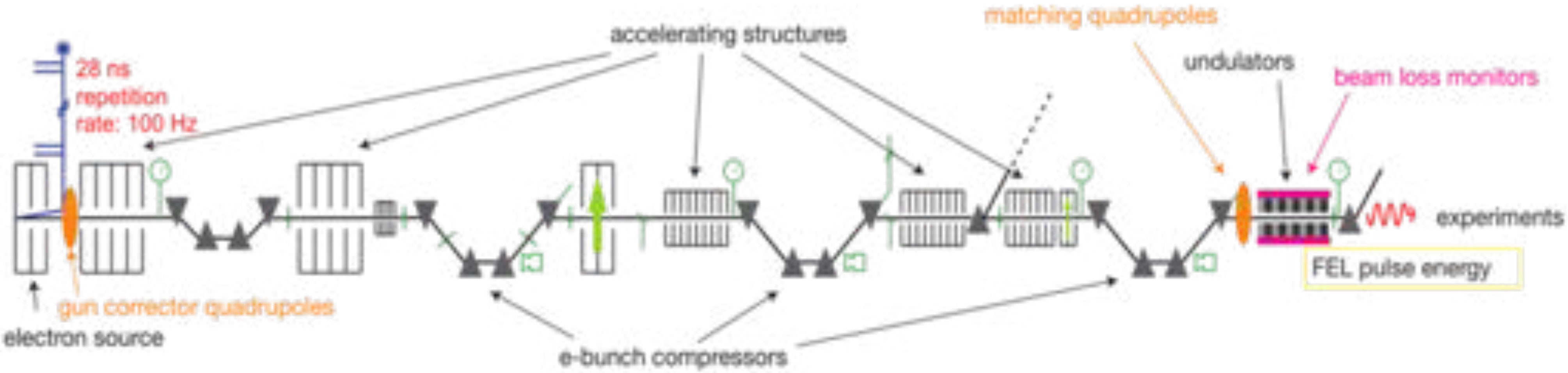


The Swiss Free Electron Laser

PAUL SCHERRER INSTITUT

Tuning SwissFEL

[w Kirschner, Mutny, Ischebeck et al'18]

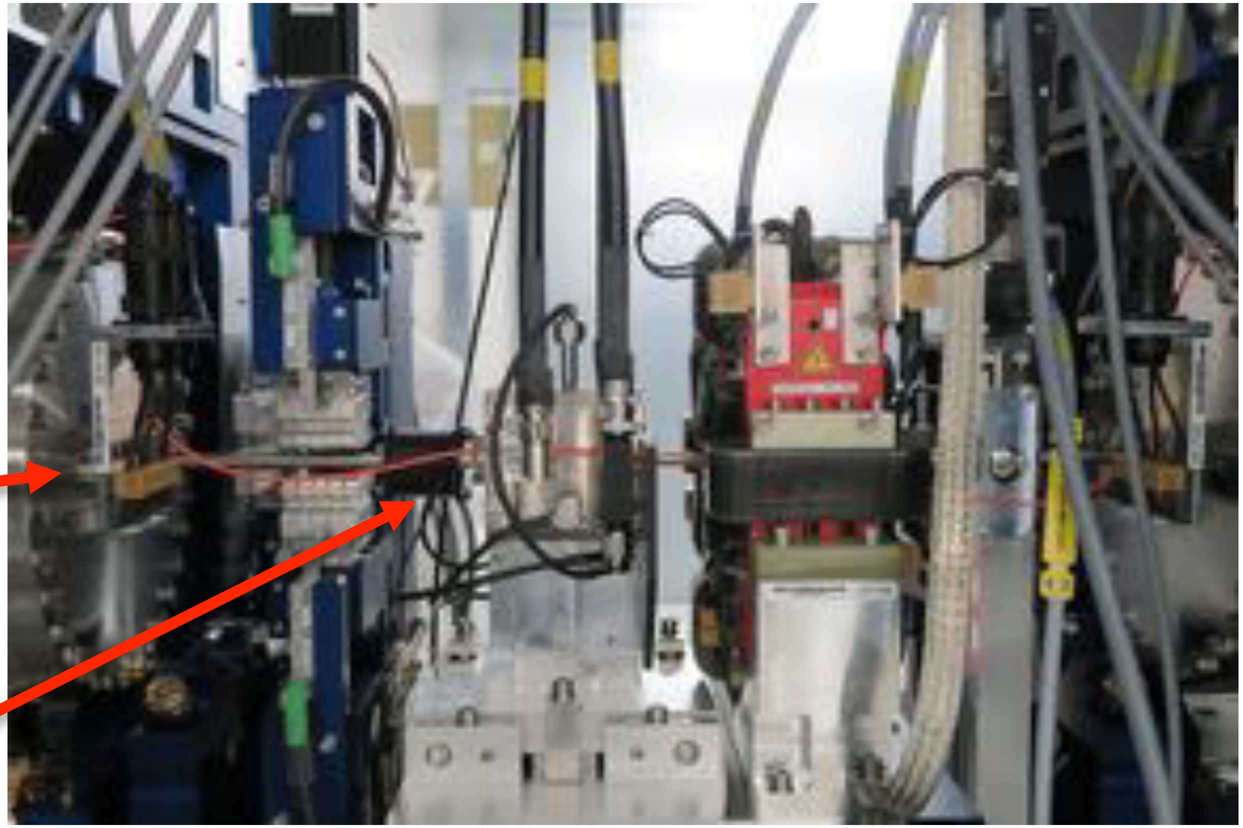


[c.f., McIntire, Ratner, Ermon '16]

Challenge: Safety Constraints

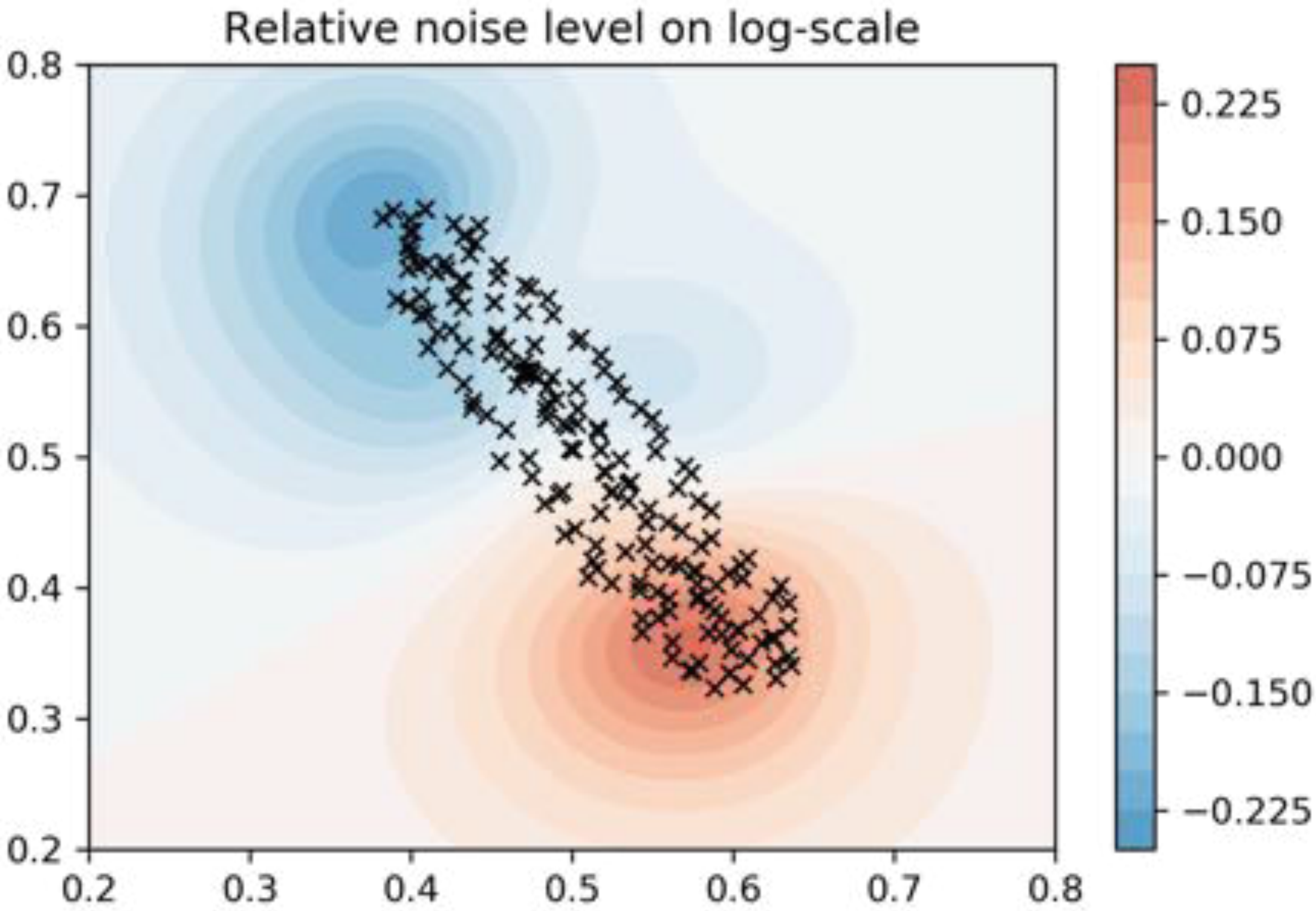
Vacuum Chamber of Undulator Module

Beam Loss Monitor



Radiation damage leads to loss of the magnetization
➔ Undulators need to be replaced

Challenge: Heteroscedastic Noise



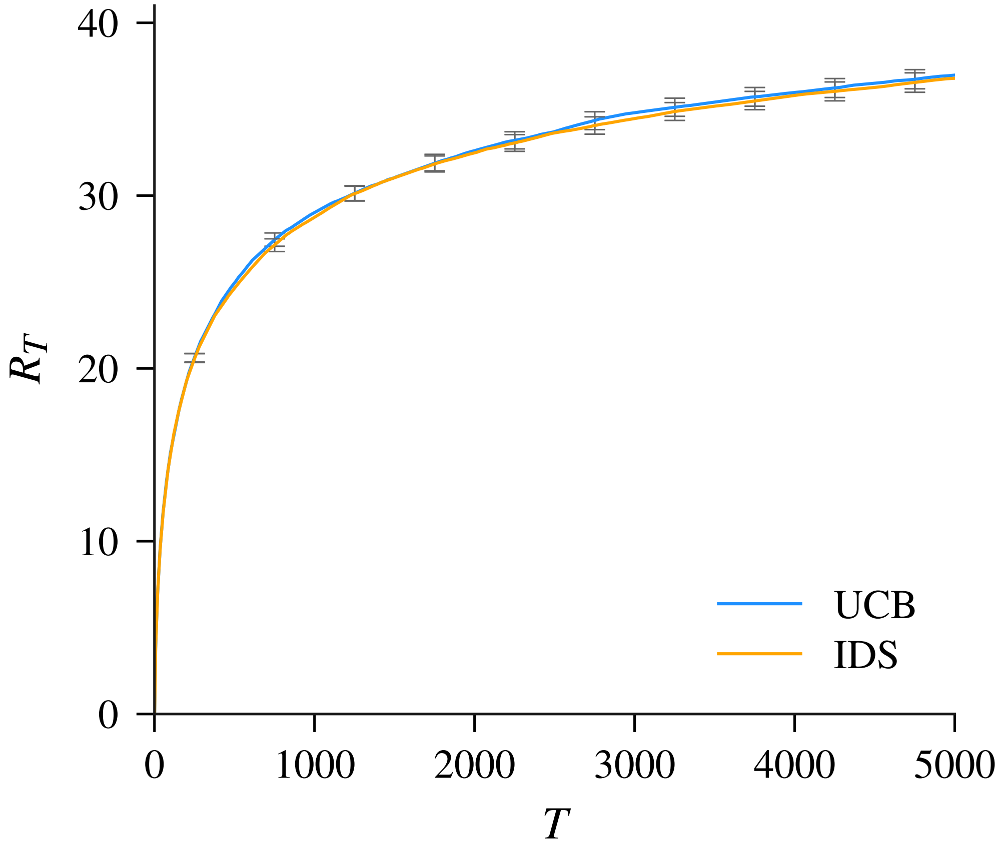
➔ Information Directed Sampling $\quad \mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} \frac{\Delta_t(\mathbf{x})}{I_t(\mathbf{x})}$

[w Kirschner '18, cf., Russo & van Roy '14]

Example: Heteroscedastic bandits

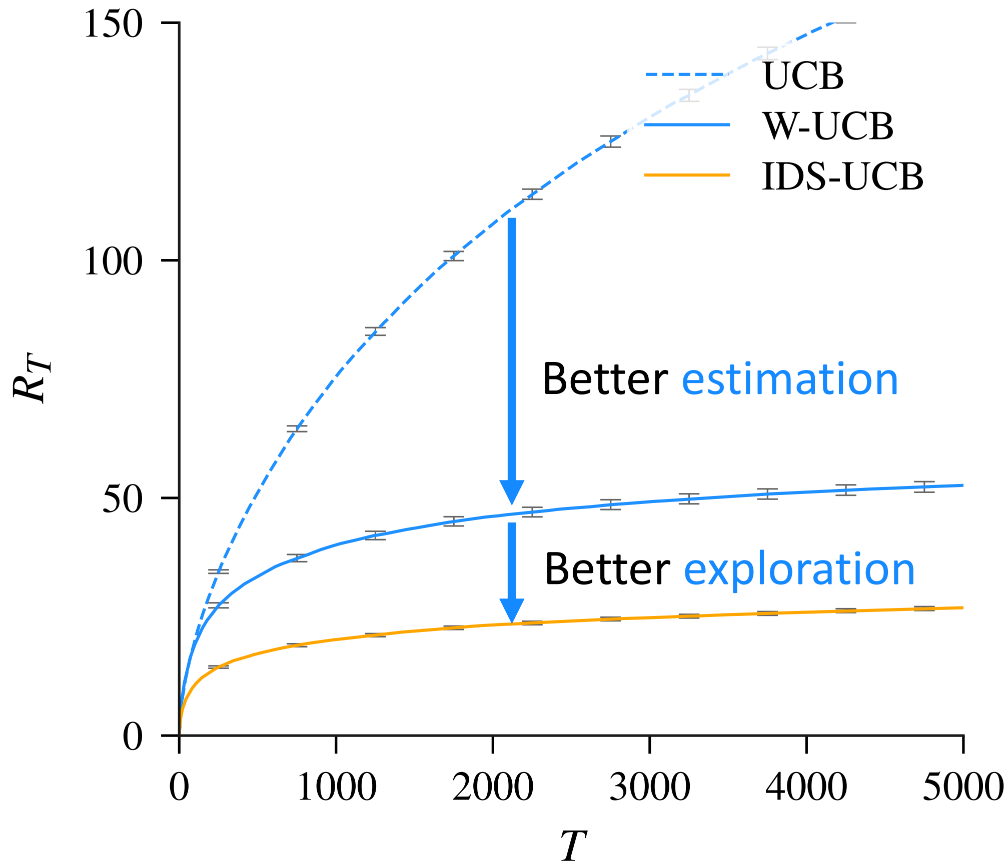
homoscedastic

$$y_t = f(\mathbf{x}_t) + \varepsilon_t$$



heteroscedastic

$$y_t = f(\mathbf{x}_t) + \varepsilon_t(\mathbf{x}_t)$$



IDS obtains significantly lower regret than UCB in case of heteroscedastic noise

Application: Exploration in Deep RL

[Nikolov, Kirschner, Berkenkamp, K, ICLR 2019]

Heteroscedasticity is everywhere in reinforcement learning

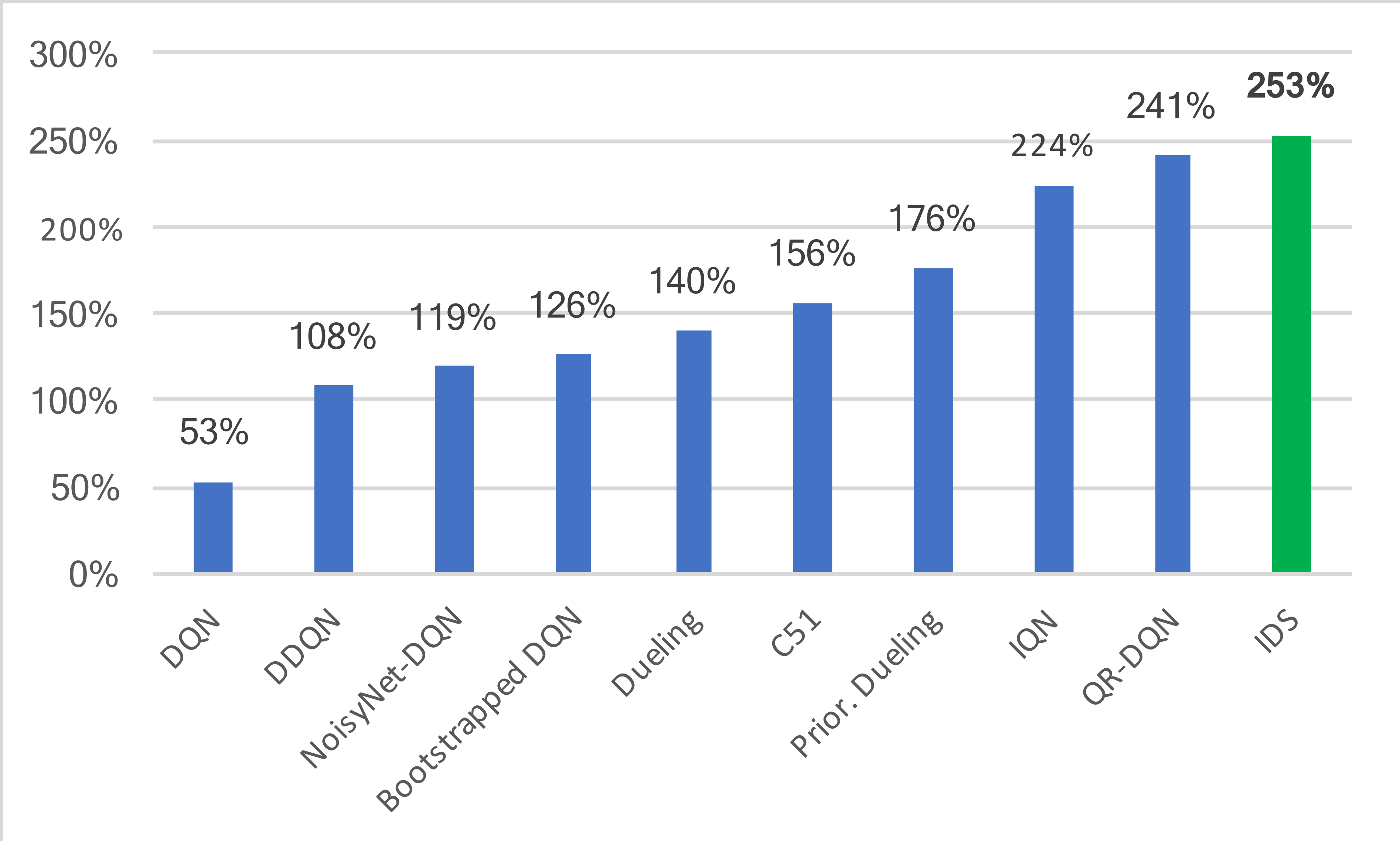
- Heteroscedastic reward functions
- Stochasticity in the transition model
- Aliasing due to partial observability
- Evolving TD targets

We propose IDS as a novel criterion for exploration in RL

- Bayesian deep learning to estimate the return distribution (Categorical DQN [Bellemare et al. 2017])
- Extract confidence intervals to estimate the instantaneous regret

IDS for Deep RL on Atari Games

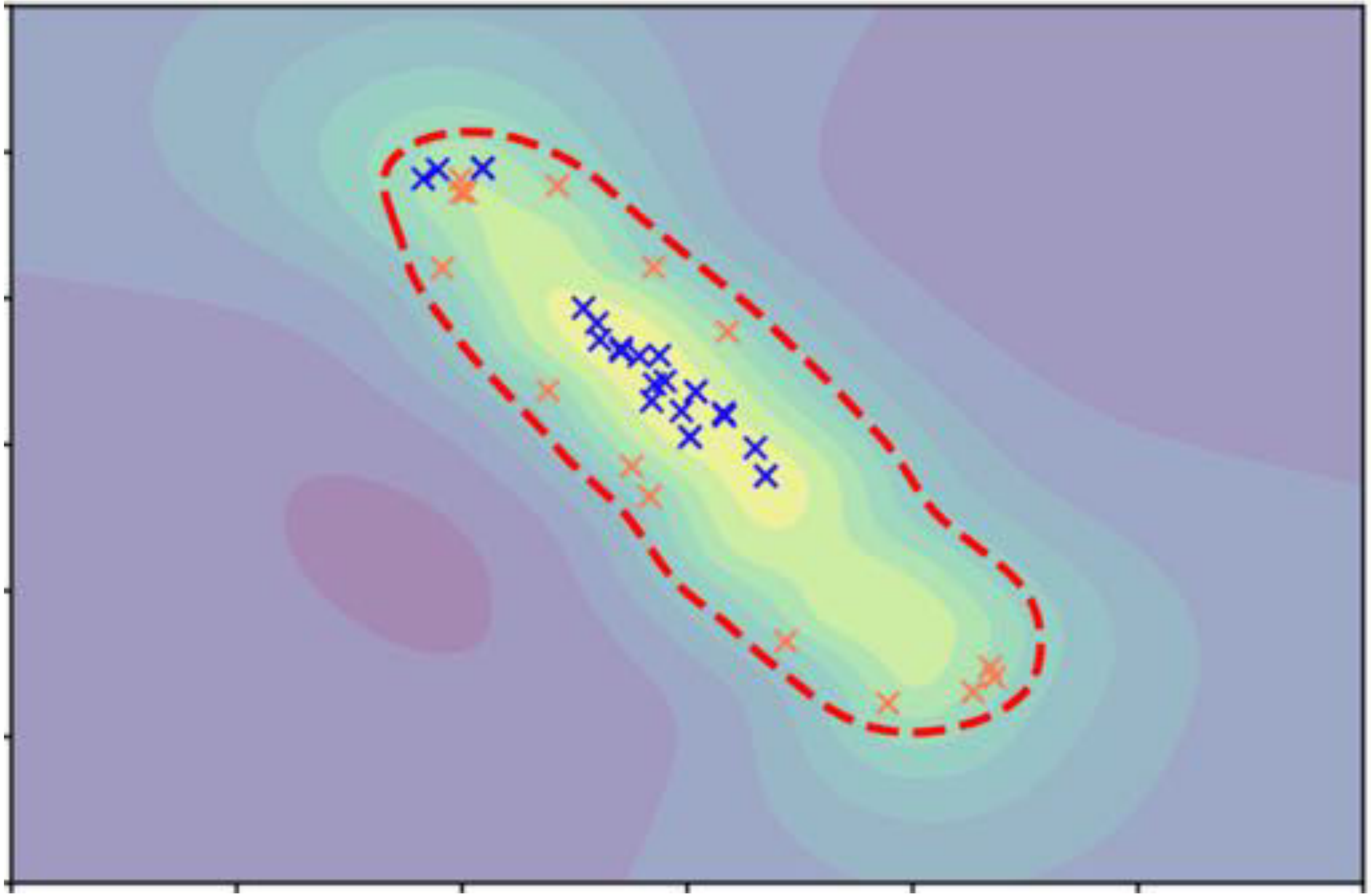
[with Nikolov, Kirschner, Berkenkamp, ICLR 2019]



More Challenges

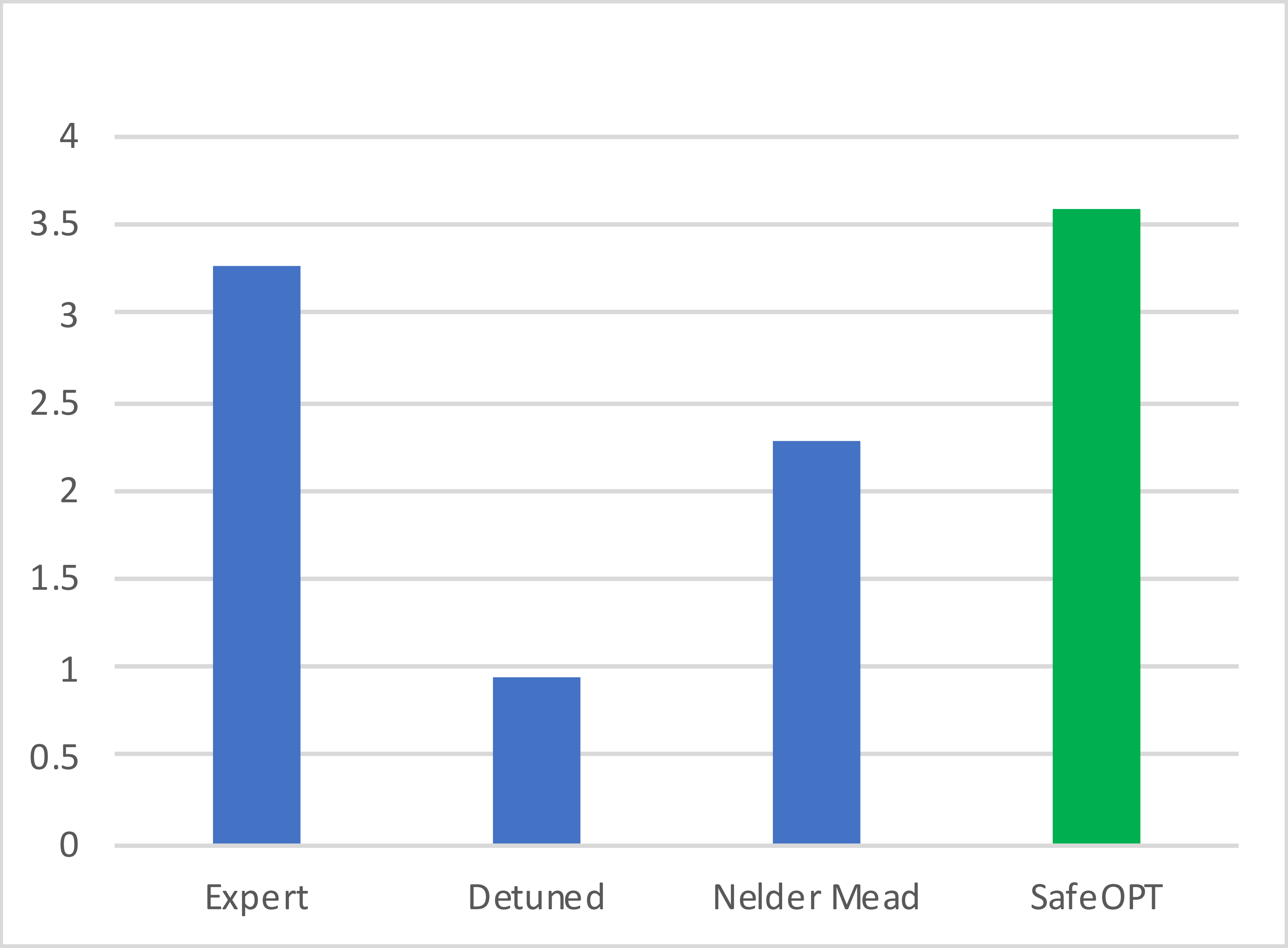
- Safety constraints crucial
- Heteroscedastic noise
- Need to contextualize to user requirements
- Simulations *slower* than physical experiment
- Variable dimensionality (2-100s)
- “Movement constraints” for parameter changes
- …

Tuning SwissFEL



Performance

[with Kirschner, Mutny, Hiller, Ischebeck ‘18]



Beyond Basic BO:

Outlook & Further topics

Outlook: Further topics

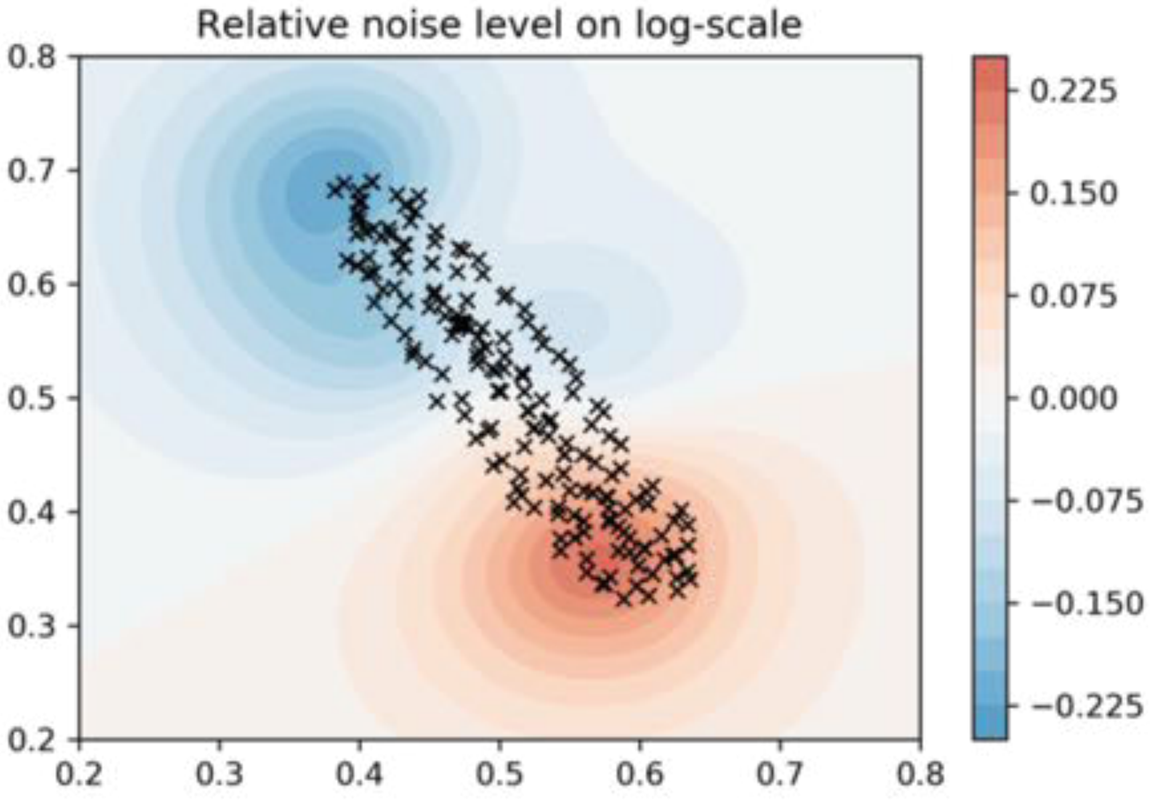
- Exploiting gradient information
- Heteroscedastic noise
- Dealing with high dimensions
- Efficient kernel approximations and beyond GPs

Exploiting gradient information

- In some applications, (noisy) gradient information may be available
- These correspond to linear observations
  → posterior is still a Gaussian process
- May have to be careful in choice of acquisition function [Wu et al NIPS '17]

Heteroscedastic Noise

$$\mathrm{Var}(y(\mathbf{x})) = g(\mathbf{x})$$

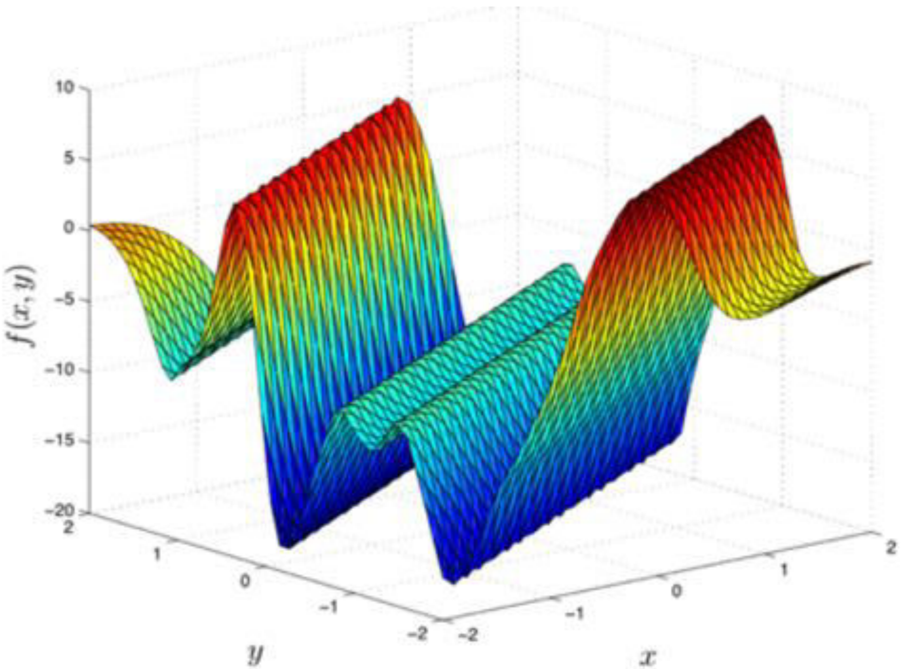


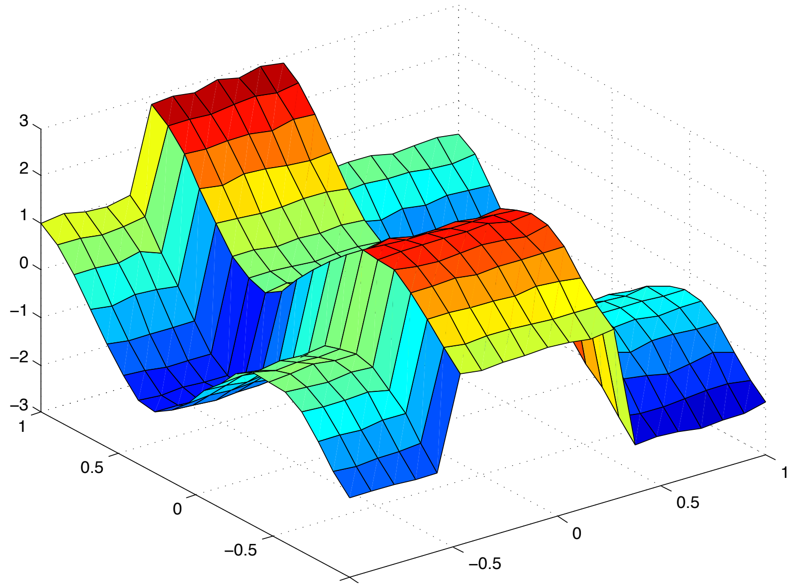
➔ Information Directed Sampling

[w Kirschner ‘18, cf., Russo & van Roy ’14]

High-dimensions

Statistical and computational challenges → need assumptions





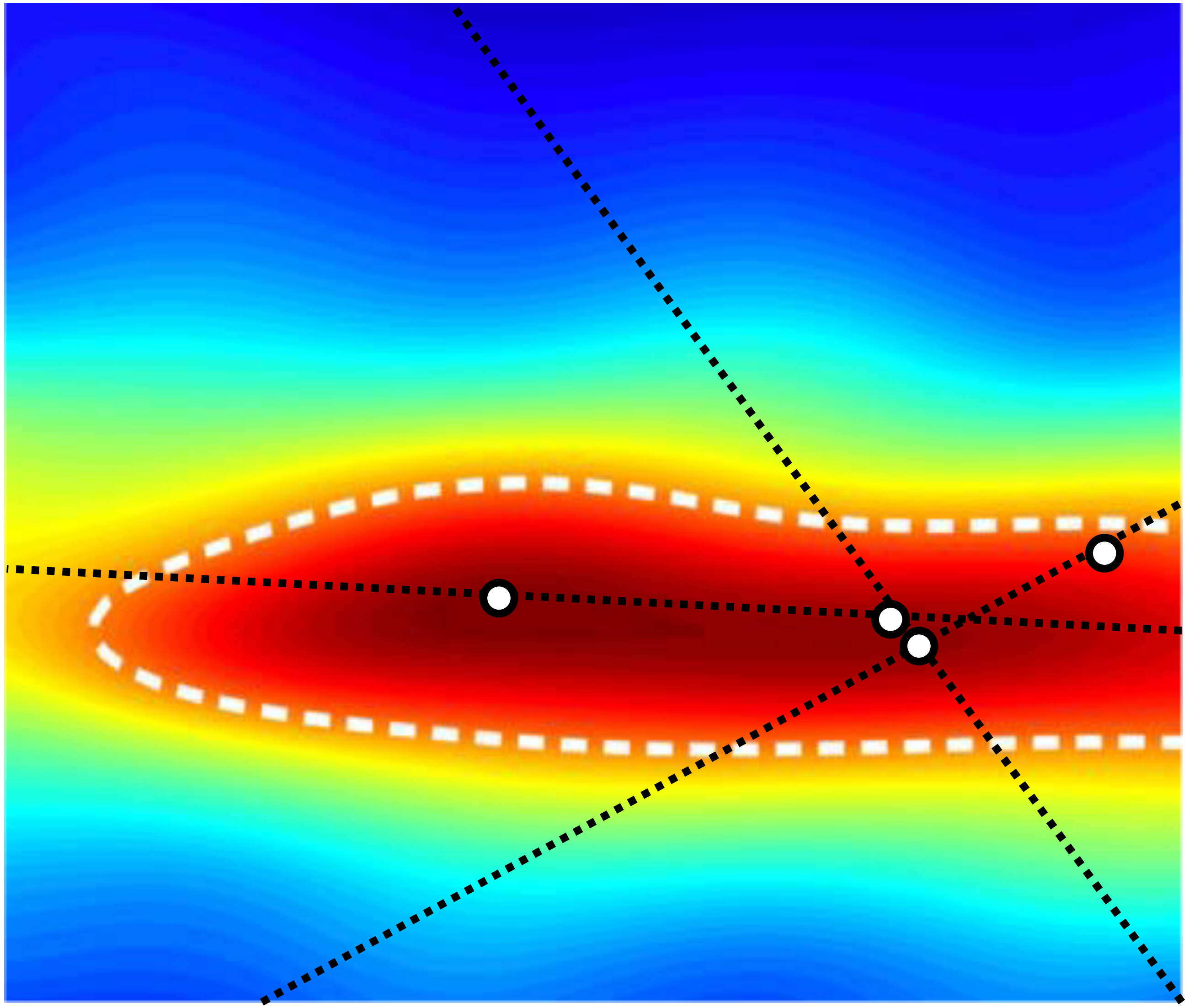
$$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x}) \quad \mathbf{A} \in \mathbb{R}^{d\times D}$$

$$f(\mathbf{x}) = \sum_i f_i(x_i)$$

[Wang et al'13, Djolonga & K'13, …]

[Kandasamy et al '15, Rolland et al '18, Mutny & K '18]

LineBO



Guarantees in high dimensions

[with Mutny, Kirschner, Hiller, Ischebeck ICML '19]

- We develop a novel algorithm – LINEBO
  - Solve a sequence of one-dimensional Bayesian optimization problems on one dimensional subspaces
- For random subspaces, can guarantee simultaneously
  - Global convergence (at Lipschitz rates, automatically *adapting to intrinsic dimension*)
  - Local convergence (at *fast rates* in case of locally strongly convex functions)
- Can also (heuristically) use more informed directions

Efficient kernel approximations

- Naively, predictions in GPs require Cholesky decompositions of T x T matrices → $O(T^3)$
- Considerable work in efficient approximations
  - Data-independent (Fourier features, …)
  - Data-dependent (pseudo-inputs, Nystrom approximation, DNN basis functions…)
  - Can provably reduce to O(T polylog(T)) for generalized additive GPs while still yielding no regret [Mutny & K NIPS ‘18]

- Much recent work on replacing GPs with neural nets [cf Springenberg et al NIPS’16, Garnelo et al ICML’18]

Conclusions

- Bridging bandits and Bayesian optimization
- Key idea: Exploit confidence bounds to constrain sampling

  Parallelization, Context, Multi-objective, Level sets,
  Active search and discovery, Safety constraints …
- Performance bounds based on information capacity, bounded via submodularity
- Strong performance on real-world problems

References

- Abbasi-Yadkori, Y. Online Learning for Linearly Parametrized Control Problems. PhD thesis, 2012.
- Abdelrahman, H., Berkenkamp, F., Poland, J., Krause, A.. Bayesian Optimization forMaximum Power Point Tracking in Photovoltaic Power Plants. ECC 2016
- Almer, O., Topham, N., & Franke, B. (2011, February). A learning-based approach to the automated design of MPSoC networks. In ICACS (pp. 243-258). Springer, Berlin, Heidelberg.
- Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2-3), 235-256.
- Azimi, Javad, Alan Fern, and Xiaoli Z. Fern. "Batch bayesian optimization via simulation matching." Advances in Neural Information Processing Systems. 2010.
- Berkenkamp, F., Schoellig, A. P., & Krause, A. (2016, May). Safe controller optimization for quadrotors with Gaussian processes. In Robotics and Automation (ICRA), 2016
- Berkenkamp, F., Schoellig, A.P. and Krause, A., (2019). No-Regret Bayesian Optimization with Unknown Hyperparameters. Journal of Machine Learning Research, 20, p.50.
- Bogunovic, I., Scarlett, J., Krause, A., & Cevher, V. (2016). Truncated variance reduction: A unified approach to Bayesian optimization and level-set estimation. NIPS
- Bubeck, S., Munos, R., Stoltz, G., & Szepesvári, C. (2011). X-armed bandits. Journal of Machine Learning Research, 12(May), 1655-1695.
- Cesa-Bianchi, N., & Lugosi, G. (2012). Combinatorial bandits. Journal of Computer and System Sciences, 78(5), 1404-1422.
- Dani, V., Hayes, T. P., & Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback.
- Desautels, T., Krause, A., & Burdick, J. W. (2014). Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. JMLR

- Djolonga, J., Krause, A., & Cevher, V. (2013). High-dimensional gaussian process bandits. In Advances in Neural Information Processing Systems (pp. 1025-1033).
- Frazier, Peter, Warren Powell, and Savas Dayanik. "The knowledge-gradient policy for correlated normal beliefs." INFORMS journal on Computing 21.4 (2009): 599-613.
- Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., & Cunningham, J. P. (2014, June). Bayesian Optimization with Inequality Constraints. In ICML (pp. 937-945).
- Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M., & Teh, Y. W. (2018). Neural processes. arXiv preprint arXiv:1807.01622.
- Garnett, R., Osborne, M. A., & Hennig, P. (2013). Active learning of linear embeddings for Gaussian processes. arXiv preprint arXiv:1310.6740.
- Gelbart, M. A., Snoek, J., & Adams, R. P. (2014). Bayesian optimization with unknown constraints. arXiv preprint arXiv:1403.5607.
- Ginsbourger, D., Le Riche, R., & Carraro, L. (2010). Kriging is well-suited to parallelize optimization. In Computational intelligence in expensive optimization problems
- Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society. Series B (Methodological), 148-177.
- Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., & Sculley, D. (2017, August). Google vizier: A service for black-box optimization. KDD
- Gotovos, A., Casati, N., Hitz, G., & Krause, A. (2013, August). Active learning for level set estimation. In IJCAI (pp. 1344-1350).
- Herbert Robbins. Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 58(5):527–535, 1952.

- Hernández-Lobato, J. M., Gelbart, M. A., Hoffman, M. W., Adams, R. P., & Ghahramani, Z. (2015). Predictive entropy search for bayesian optimization with unknown constraints.
- Hernández-Lobato, J. M., Hoffman, M. W., & Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. NIPS
- Hitz, G., Gotovos, A., Garneau, M. É., Pradalier, C., Krause, A., & Siegwart, R. Y. (2014, May). Fully autonomous focused exploration for robotic environmental monitoring. ICRA
- Jones, Donald R., Matthias Schonlau, and William J. Welch. "Efficient global optimization of expensive black-box functions." Journal of Global optimization 13.4 (1998): 455-492.
- Kandasamy, K., Schneider, J., & Póczos, B. (2015, June). High dimensional Bayesian optimisation and bandits via additive models. ICML
- Kathuria, T., Deshpande, A., & Kohli, P. (2016). Batched gaussian process bandit optimization via determinantal point processes. NIPS
- Kirschner, J., & Krause, A. (2018). Information Directed Sampling and Bandits with Heteroscedastic Noise. COLT
- Kirschner, J., Mutný, M., Hiller, N., Ischebeck, R. and Krause, A., (2019). Adaptive and Safe Bayesian Optimization in High Dimensions via One-Dimensional Subspaces. ICML
- Kleinberg, R., Slivkins, A., & Upfal, E. (2008, May). Multi-armed bandits in metric spaces. STOC
- Krause, A. & Guestrin, C. (2005). Near-optimal Nonmyopic Value of Information in Graphical Models, Proc. Uncertainty in Artificial Intelligence (UAI)
- Krause, A., & Ong, C. S. (2011). Contextual gaussian process bandit optimization. In Advances in Neural Information Processing Systems (pp. 2447-2455).
- Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010, April). A contextual-bandit approach to personalized news article recommendation. WWW

- Lindley, D. V. (1956). On a measure of the information provided by an experiment. The Annals of Mathematical Statistics, 986-1005.
- Marco, A., Berkenkamp, F., Hennig, P., Schoellig, A. P., Krause, A., Schaal, S., & Trimpe, S. (2017, May). Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In Robotics and Automation (ICRA)
- Močkus, J. "On Bayesian methods for seeking the extremum." Optimization Techniques IFIP Technical Conference. Springer, Berlin, Heidelberg, 1975.
- Mockus, J. (1989). The Bayesian approach to local optimization. In Bayesian Approach to Global Optimization(pp. 125-156). Springer, Dordrecht.
- Mutny, M. & Krause, A. (2018) Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. In Neural Information Processing Systems (NIPS)
- Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—I. Mathematical programming, 14(1), 265-294.
- Nikolov, N., Kirschner, J., Berkenkamp, F. & Krause A. (2019). Information-Directed Exploration for Deep Reinforcement Learning. ICLR
- Rasmussen, C. E., & Williams, C. K. (2006). Gaussian processes for machine learning. 2006. The MIT Press, Cambridge, MA, USA, 38, 715-719.
- Rolland, P., Scarlett, J., Bogunovic, I., & Cevher, V. (2018). High-Dimensional Bayesian Optimization via Additive Models with Overlapping Groups. arXiv preprint arXiv:1802.07028.
- Romero, P. A., Krause, A., & Arnold, F. H. (2013). Navigating the protein fitness landscape with Gaussian processes. Proc. National Academy of Sciences, 110(3), E193-E201.
- Rusmevichientong, P., & Tsitsiklis, J. N. (2010). Linearly parameterized bandits. Mathematics of Operations Research, 35(2), 395-411.

- Russo, D., & Van Roy, B. (2014). Learning to optimize via information-directed sampling. In Advances in Neural Information Processing Systems (pp. 1583-1591).
- Scarlett, J. (2018). Tight Regret Bounds for Bayesian Optimization in One Dimension. ICML
- Schonlau, M. (1997). Computer experiments and global optimization.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104(1), 148-175.
- Shewry, M. C., & Wynn, H. P. (1987). Maximum entropy sampling. Journal of applied statistics, 14(2), 165-170.
- Siegmund, N., Kolesnikov, S. S., Kästner, C., Apel, S., Batory, D., Rosenmüller, M., & Saake, G. (2012, June). Predicting performance via automated feature-interaction detection. ICSE
- Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. (2012) "Practical bayesian optimization of machine learning algorithms." Advances in neural information processing systems. 2012.
- Springenberg, J. T., Klein, A., Falkner, S., & Hutter, F. (2016). Bayesian optimization with robust bayesian neural networks. NIPS
- Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. W. (2012). Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory, 58(5), 3250-3265.
- Sui, Y., Gotovos, A., Burdick, J., & Krause, A. (2015, June). Safe exploration for optimization with Gaussian processes. In International Conference on Machine Learning (pp. 997-1005).
- Swersky, K., Snoek, J., & Adams, R. P. (2013). Multi-task bayesian optimization. In Advances in neural information processing systems (pp. 2004-2012).

- Vanchinathan, H. P., Nikolic, I., De Bona, F., & Krause, A. (2014, October). Explore-exploit in top-n recommender systems via gaussian processes. ACM RecSys
- Villemonteix, J., Vazquez, E., & Walter, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. Journal of Global Optimization, 44(4), 509.
- Wang, Z., & Jegelka, S. (2017). Max-value entropy search for efficient Bayesian optimization. arXiv preprint arXiv:1703.01968.
- Wang, Z., Zoghi, M., Hutter, F., Matheson, D., & De Freitas, N. (2013, August). Bayesian Optimization in High Dimensions via Random Embeddings. In IJCAI (pp. 1778-1784).
- Wu, J., Poloczek, M., Wilson, A. G., & Frazier, P. (2017). Bayesian optimization with gradients. In Advances in Neural Information Processing Systems (pp. 5267-5278).
- Zuluaga, M., Krause, A., & Püschel, M. (2016). ε-pal: an active learning approach to the multi-objective optimization problem. JMLR
- Zuluaga, M., Krause, A., Milder, P., & Püschel, M. (2012, June). Smart design space sampling to predict pareto-optimal solutions. In ACM SIGPLAN Notices (Vol. 47, No. 5, pp. 119-128).