Bayesian Optimization for Experiment Design

Barnabas Poczos

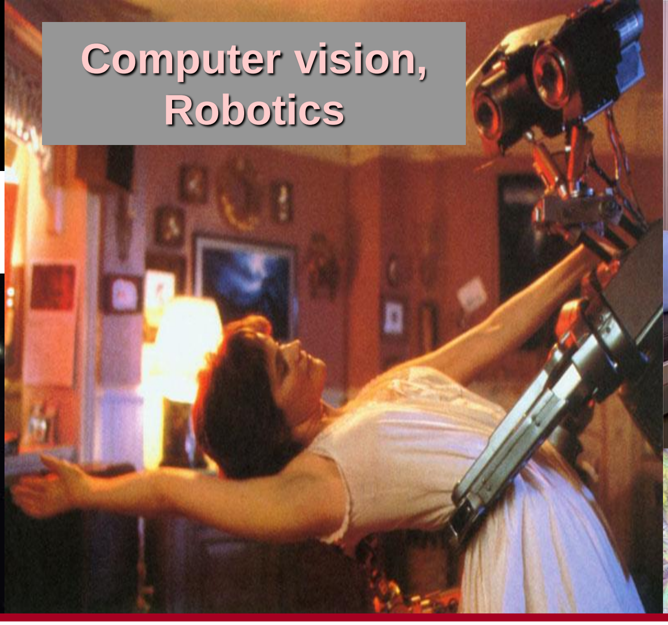
Carnegie Mellon University

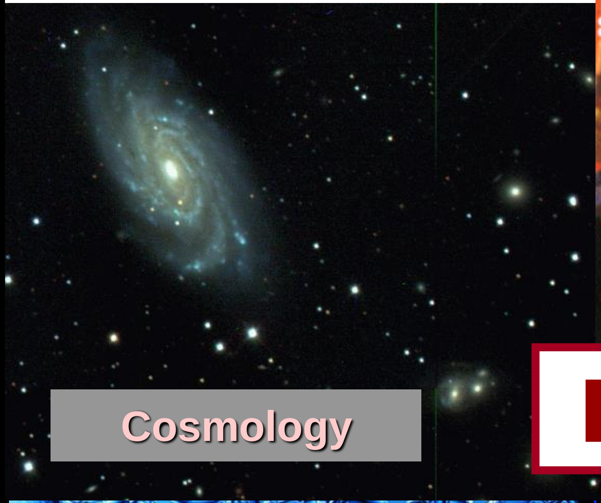
Machine Learning Department

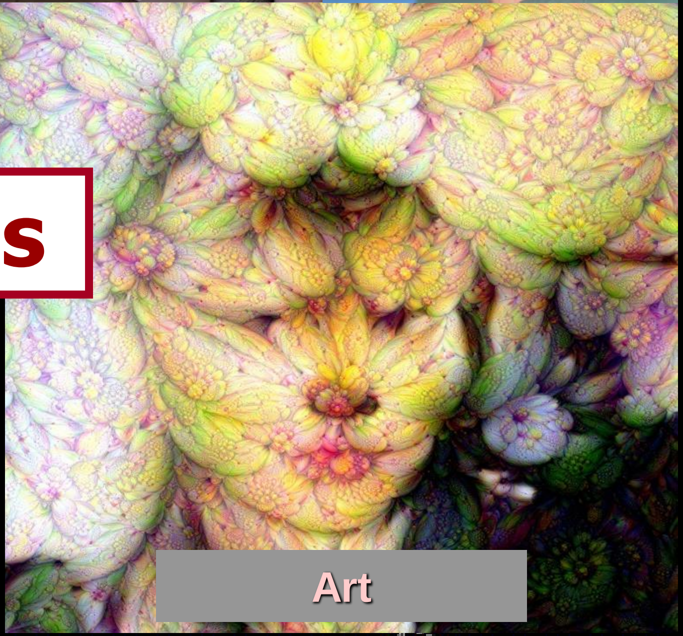
Auton Lab

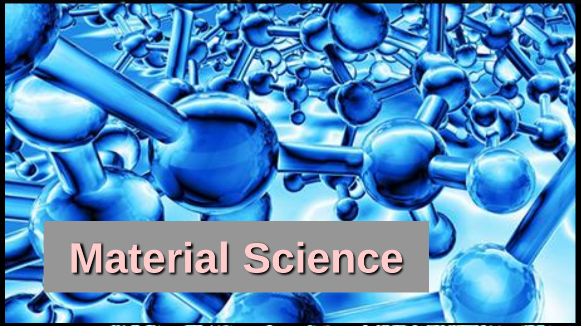
www.autonlab.org

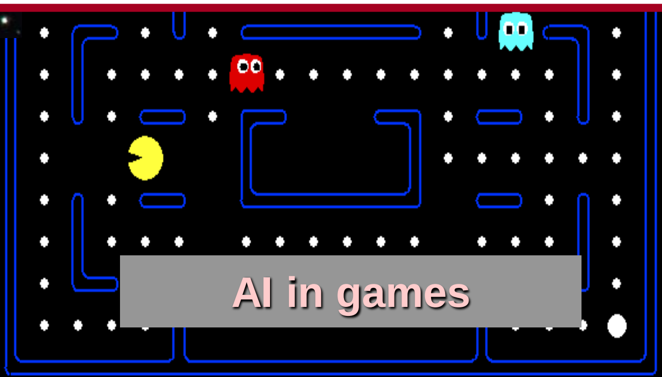
Barnabas Poczos

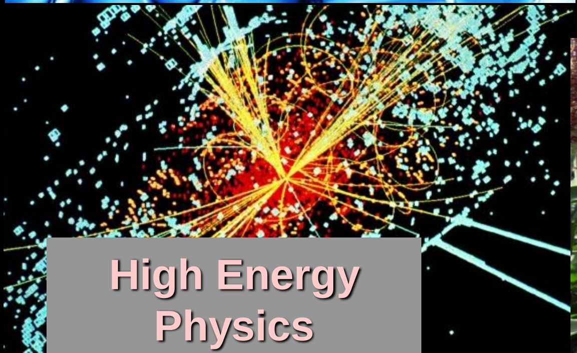


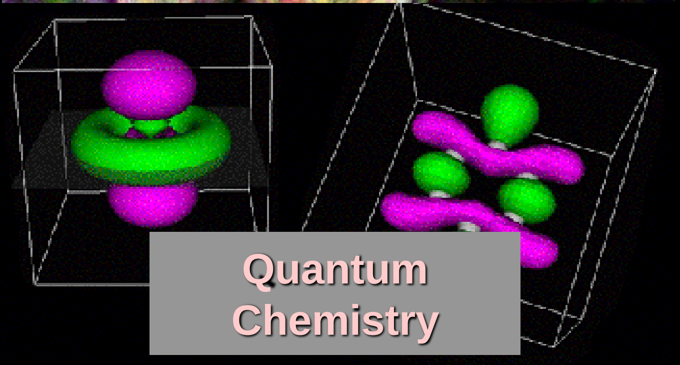










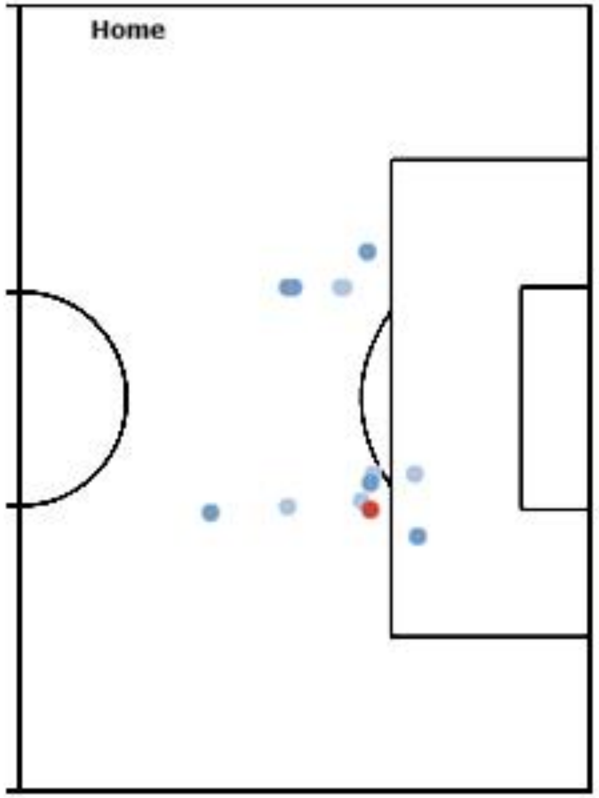


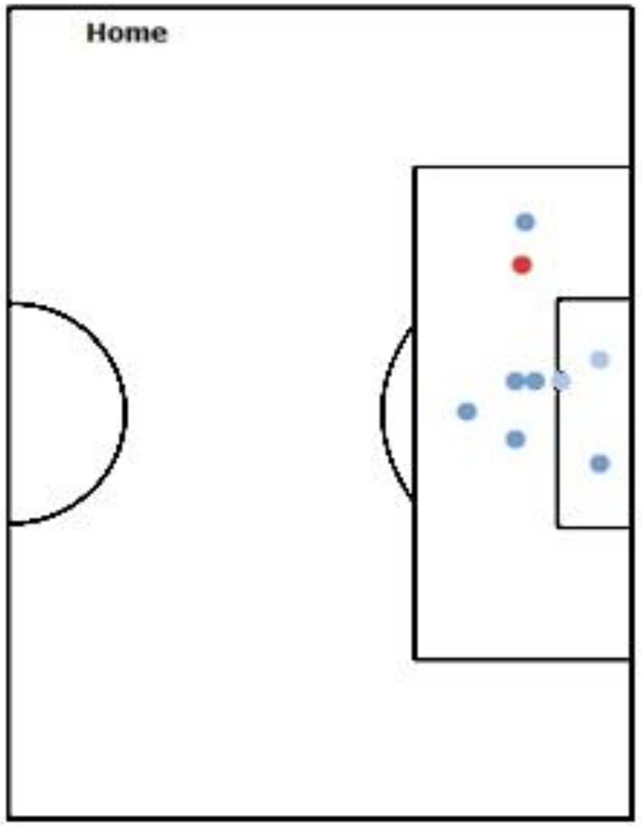
Contents

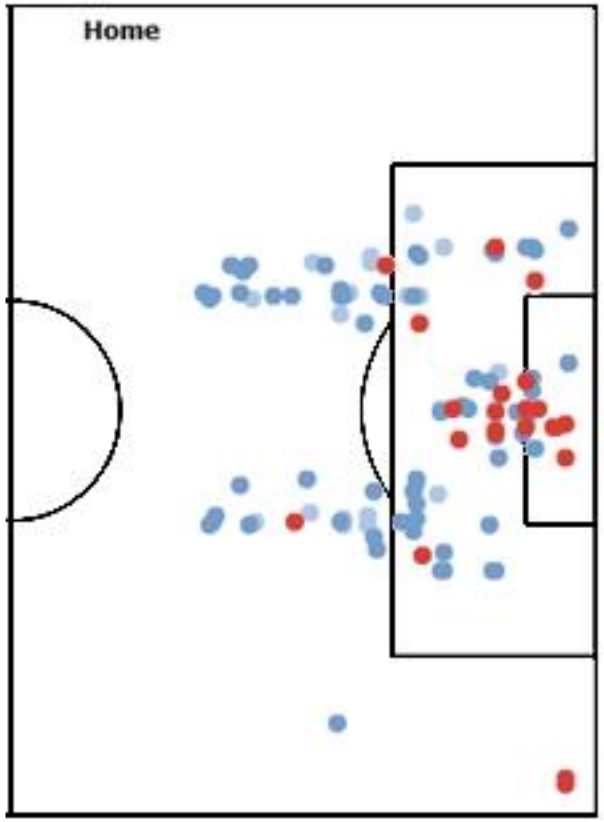
ML on Point Cloud Data

Distributional Data

Manchester United 07/08







Owen Hargreaves

Rio Ferdinand

Cristiano Ronaldo

Shot Type

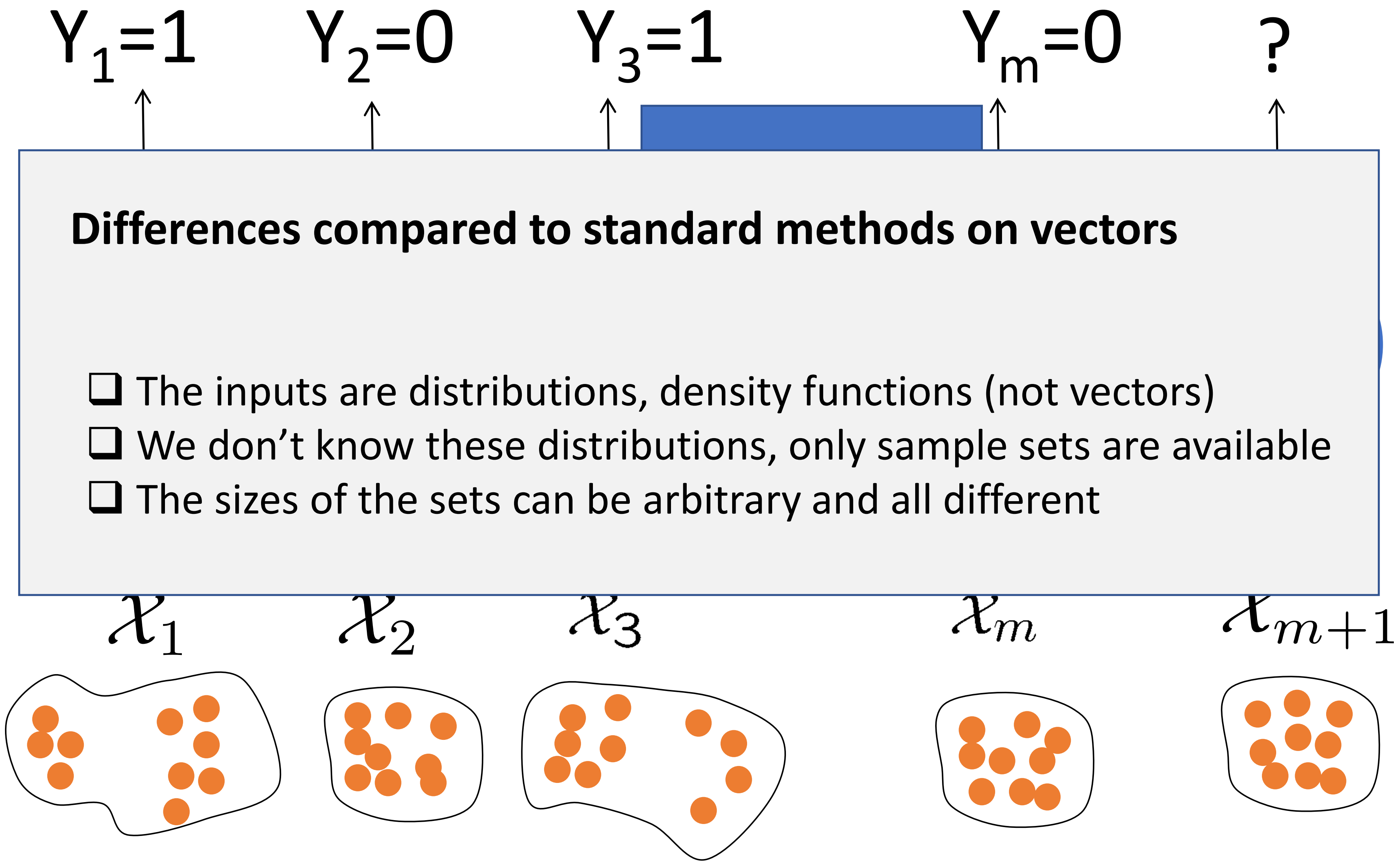
● Goals

● Shots on Goal

● Shots

www.juhokim.com/projects.php

Distribution Regression / Classification



Differences compared to standard methods on vectors

- ❑ The inputs are distributions, density functions (not vectors)
- ❑ We don't know these distributions, only sample sets are available
- ❑ The sizes of the sets can be arbitrary and all different

Distances / Divergences between Distributions

Euclidean: $D(p,q) = (\int (p(x) - q(x))^2 dx)^{1/2}$

Kullback-Leibler: $D(p,q) = KL(p,q) = \int p(x) \log \frac{p(x)}{q(x)} dx$

Renyi: $D(p,q) = R_\alpha(p\|q) = \frac{1}{\alpha-1} \log \int p^\alpha q^{1-\alpha}$

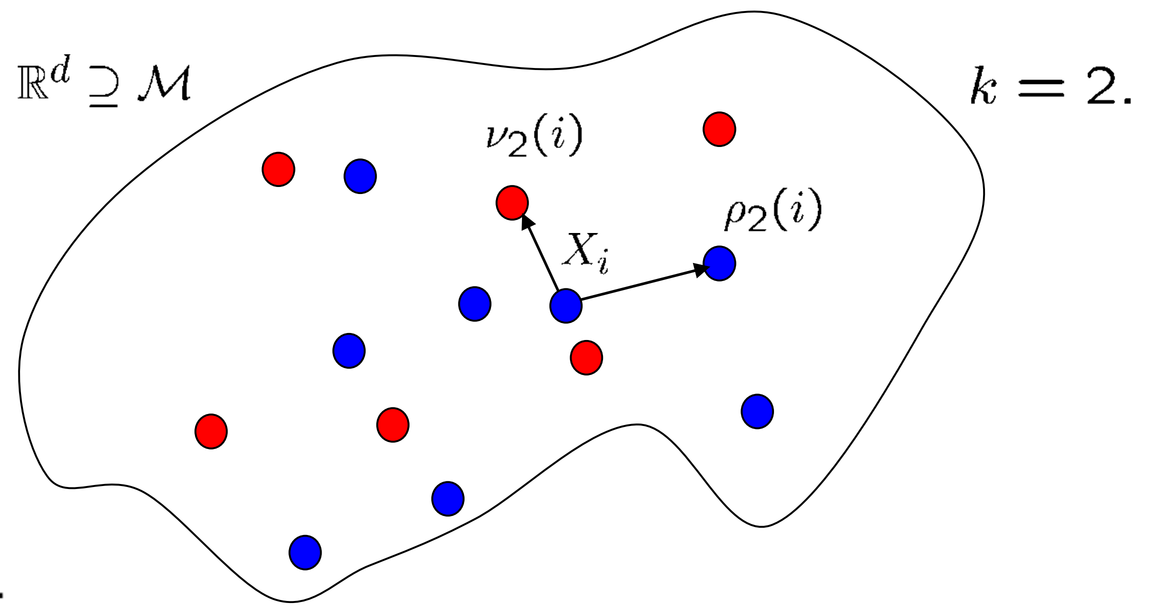
RÉNYI DIVERGENCE ESTIMATION

without density estimation

Using $X_{1:n} = \{X_1, \ldots, X_n\} \sim p \quad Y_{1:m} = \{Y_1, \ldots, Y_m\} \sim q$

Estimate divergence $R_\alpha(p\|q) \doteq \frac{1}{\alpha - 1} \log \int p^\alpha q^{1-\alpha}$

The Estimator



$k \geq 1$, fixed.

$\rho_k(i)$: the distance of the k-th nearest neighbor of X_i in $X_{1:n}$

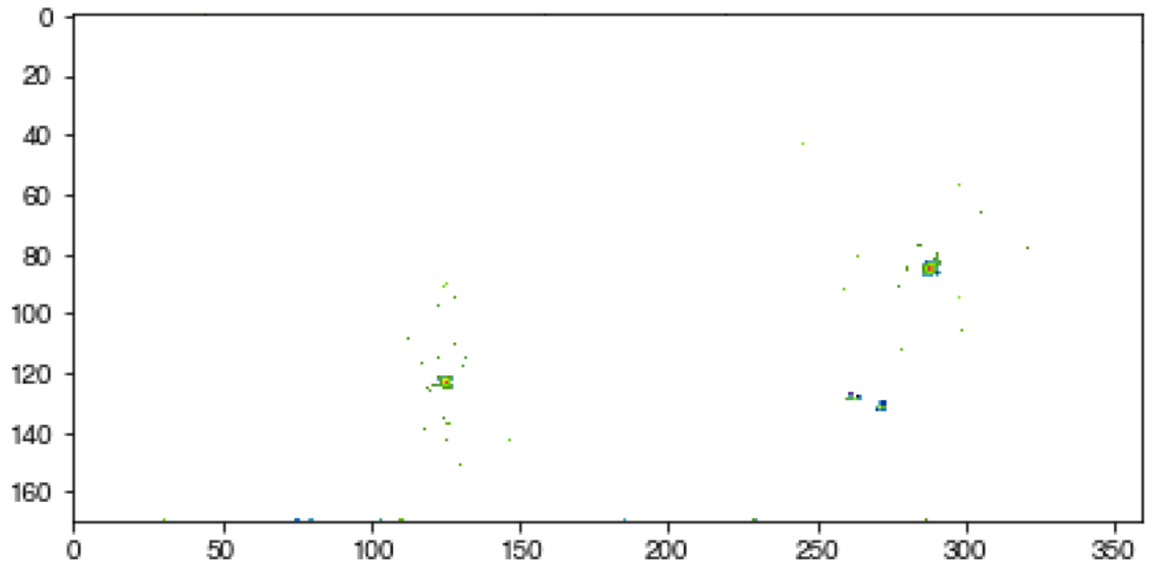
$\nu_k(i)$: the distance of the k-th nearest neighbor of X_i in $Y_{1:m}$

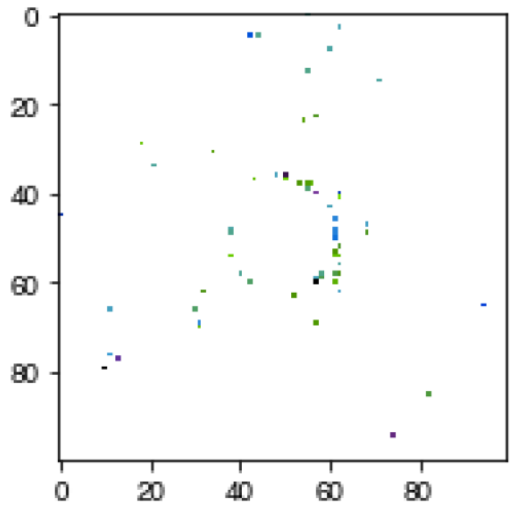
$$D_\alpha(p\|q) \doteq \int p^\alpha q^{1-\alpha}$$

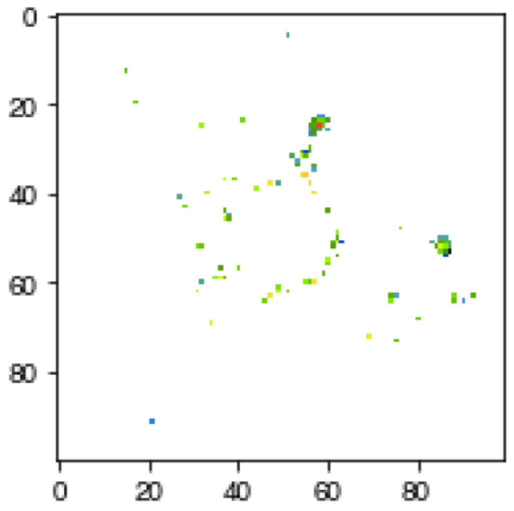
$$\widehat{D}_\alpha(X_{1:n}\|Y_{1:m}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(n-1)\rho_k^d(i)}{m\nu_k^d(i)}\right)^{1-\alpha}\frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}$$

Applications

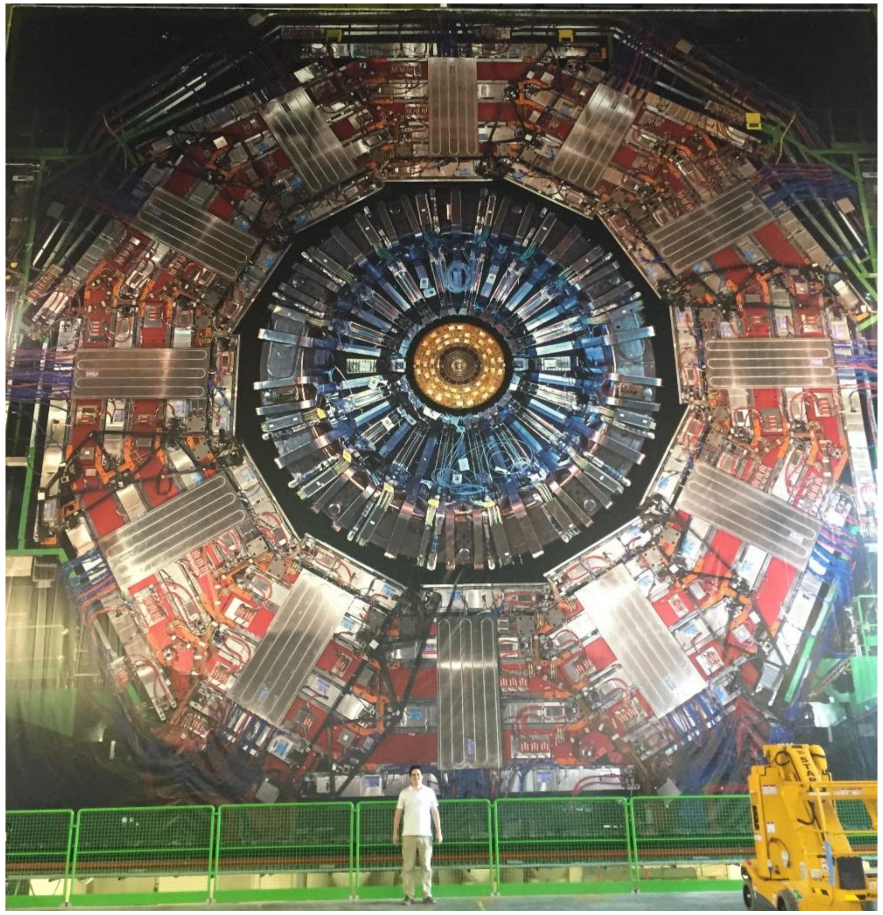
High Energy Physics



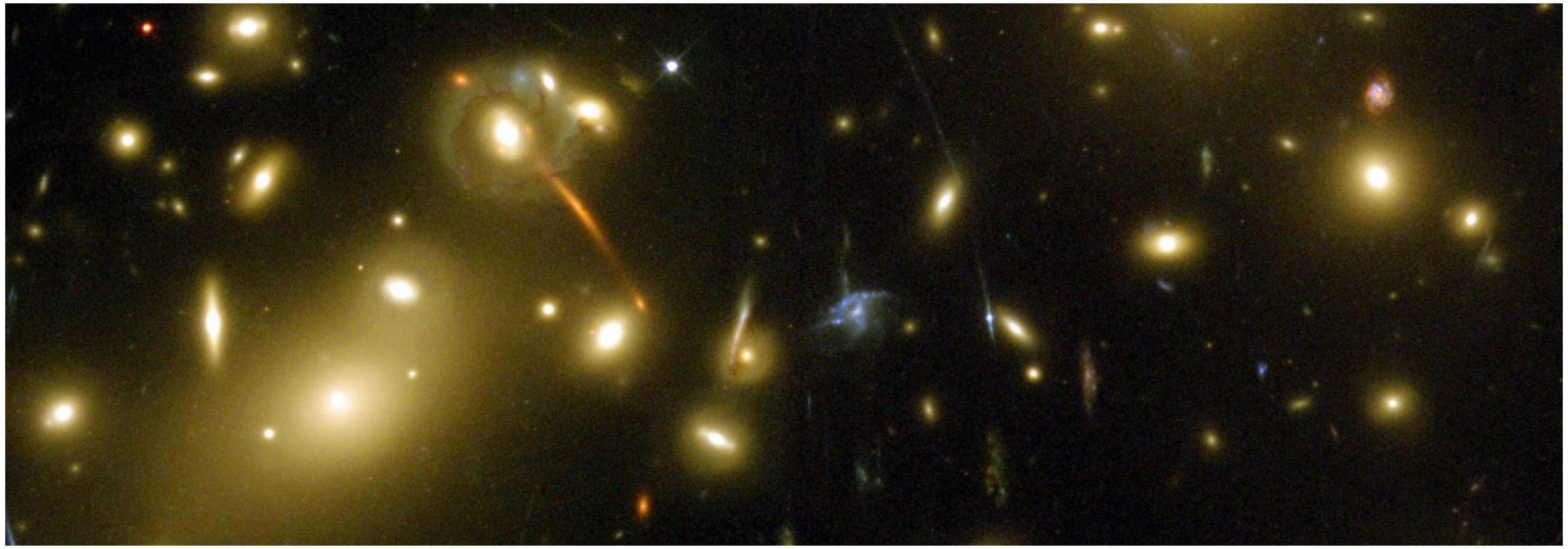




End-to-End Event Classification



Find new scientific “laws” in physics



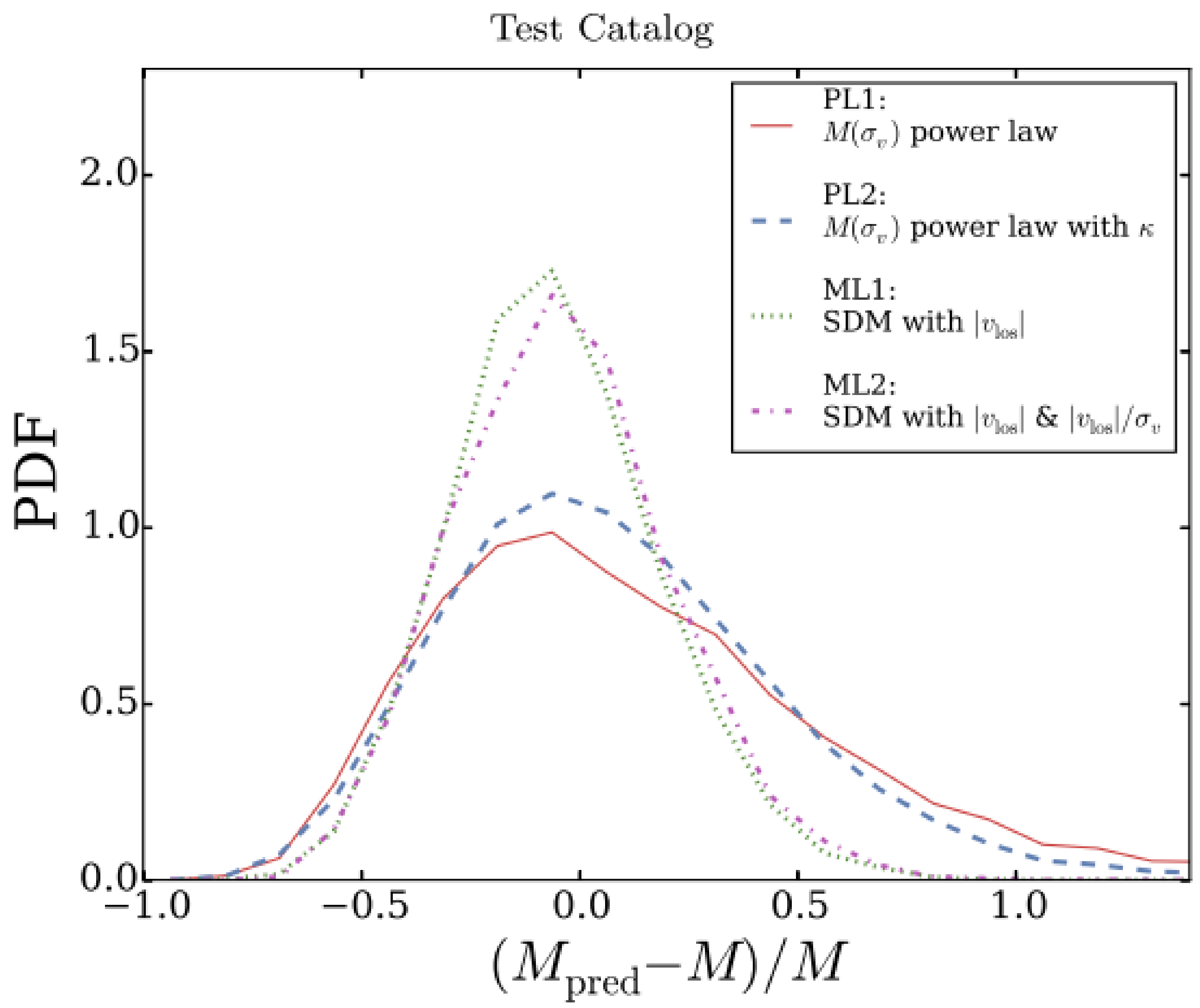
Goal: Estimate dynamical mass of galaxy clusters.

Importance: Galaxy clusters are being the largest gravitationally bound systems in the Universe. Dynamical mass measurements are important to understand the behavior of dark matter and normal matter.

Difficulty: We can only measure the velocity of galaxies not the mass of their cluster. Physicists estimate dynamical cluster mass from single velocity dispersion.

Our method: Estimate the cluster mass from the whole distribution of velocities rather than just a simple velocity distribution.

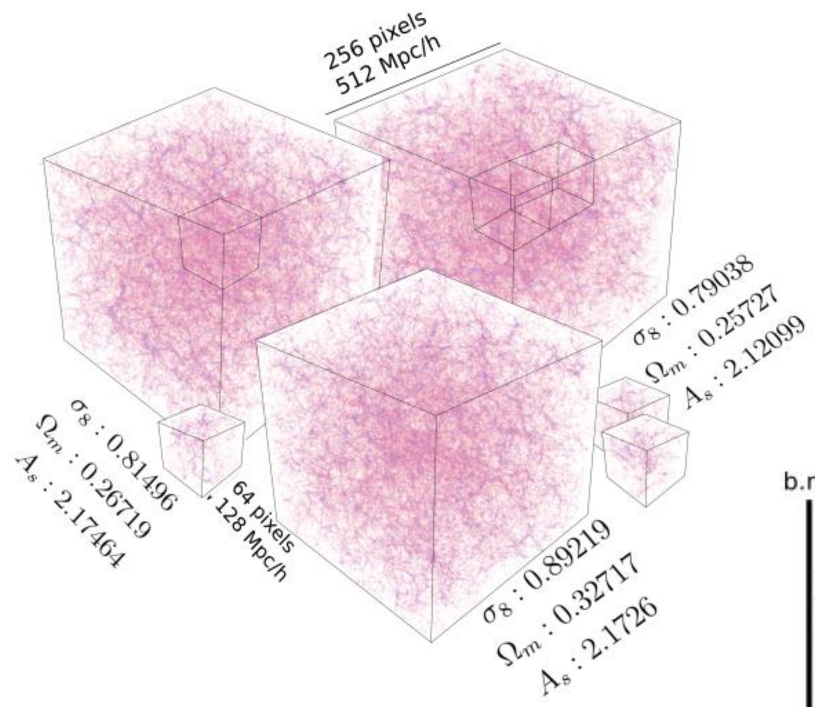
Find new scientific laws in physics

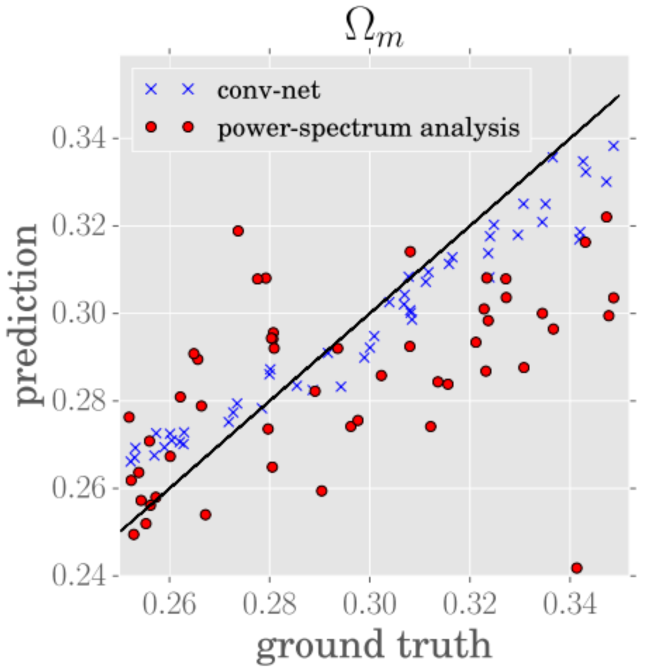


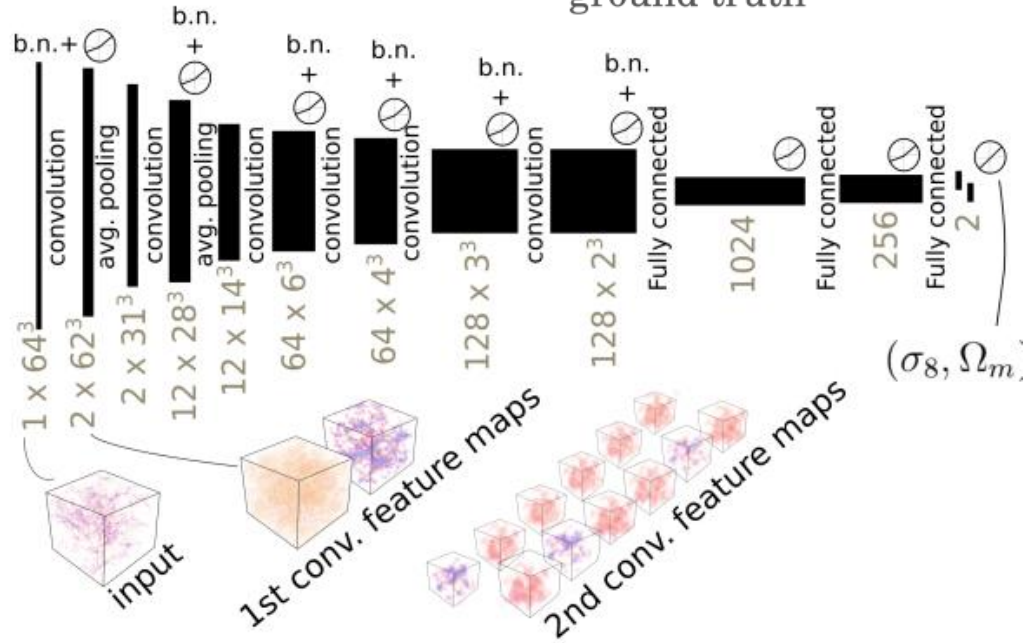
Michelle Ntampaka et al, A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters, APJ 2015

Find the parameters of Universe

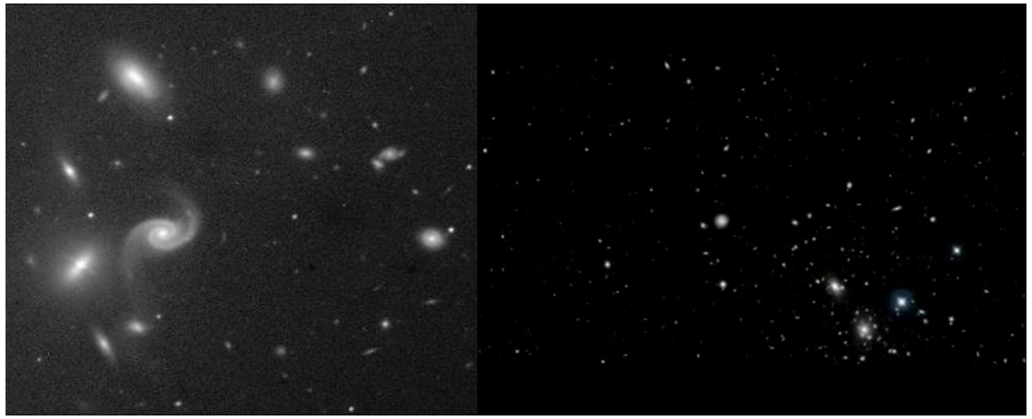
Given a distribution of particles, our goal is to predict the parameters of the simulated universe



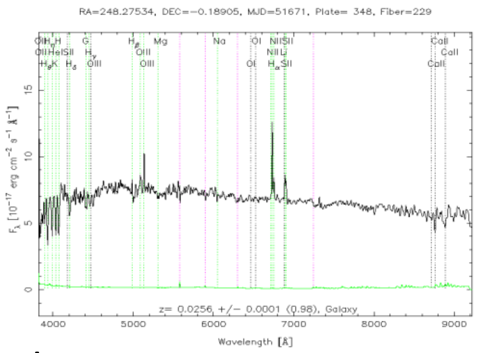




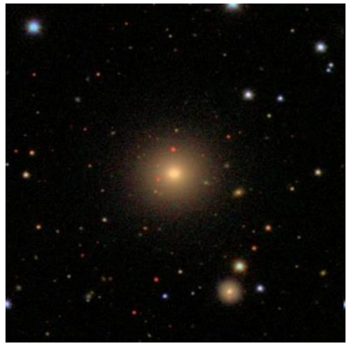
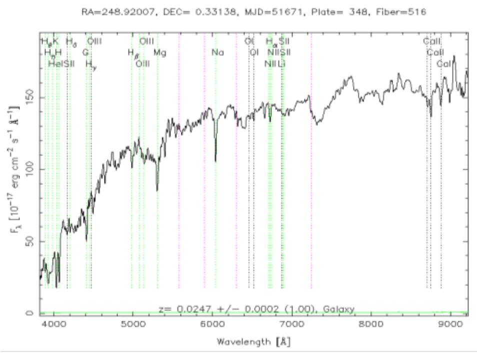
Find interesting Galaxy Clusters



Sloan Digital Sky Survey (SDSS)
- ❑ continuum spectrum
- ❑505 galaxy clusters (10-50 galaxies in each)
- ❑7530 galaxies


RA=248.27534, DEC=-0.18905, MJD=51671, Plate= 348, Fiber=229
z= 0.0256 +/- 0.0001 (0.98), Galaxy
Wavelength [Å]
F_λ [10^{-17} erg cm^{-2} s^{-1} Å^{-1}]

Blue galaxy



RA=248.92007, DEC= 0.33138, MJD=51671, Plate= 348, Fiber=516
z= 0.0247 +/- 0.0002 (1.00), Galaxy
Wavelength [Å]
F_λ [10^{-17} erg cm^{-2} s^{-1} Å^{-1}]

Red galaxy

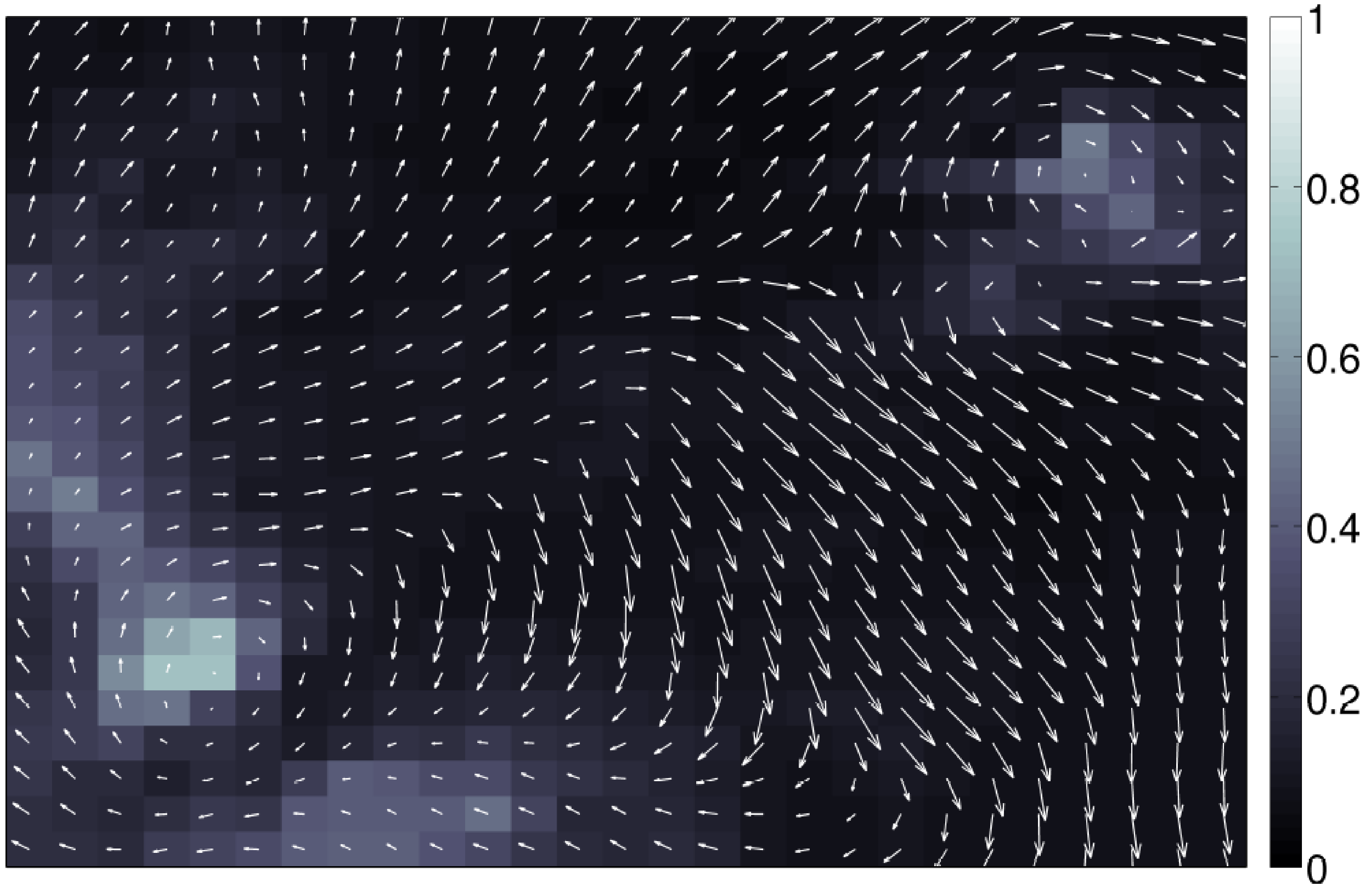
What are the most anomalous galaxy clusters?

The most anomalous galaxy cluster contains mostly
- ❑ star forming blue galaxies
- ❑ irregular galaxies

B. Póczos, L. Xiong & J. Schneider, UAI, 2011.

Credits: ESA, NASA

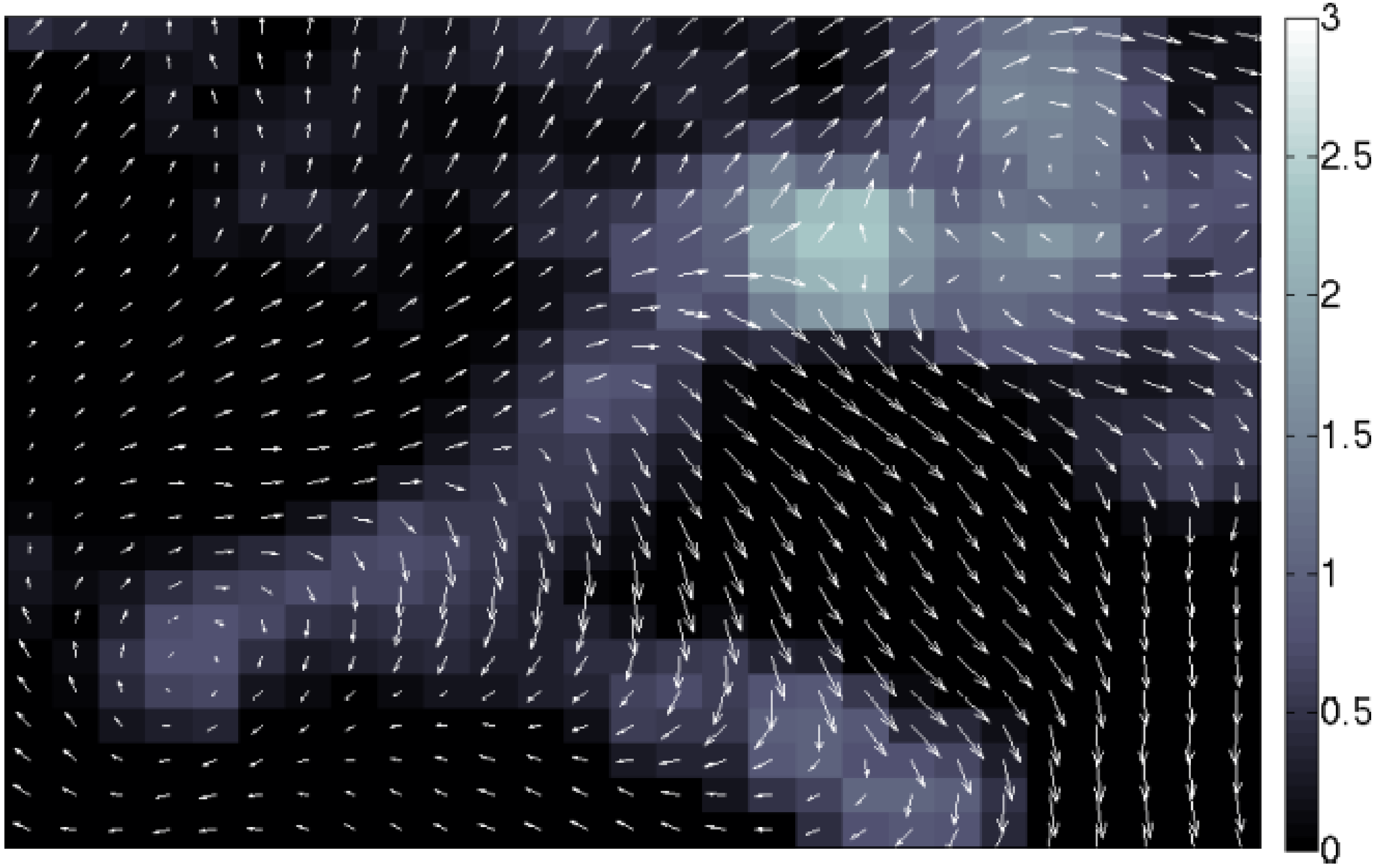
Finding Vortices in Simulated Fluid Flow



Classification probabilities

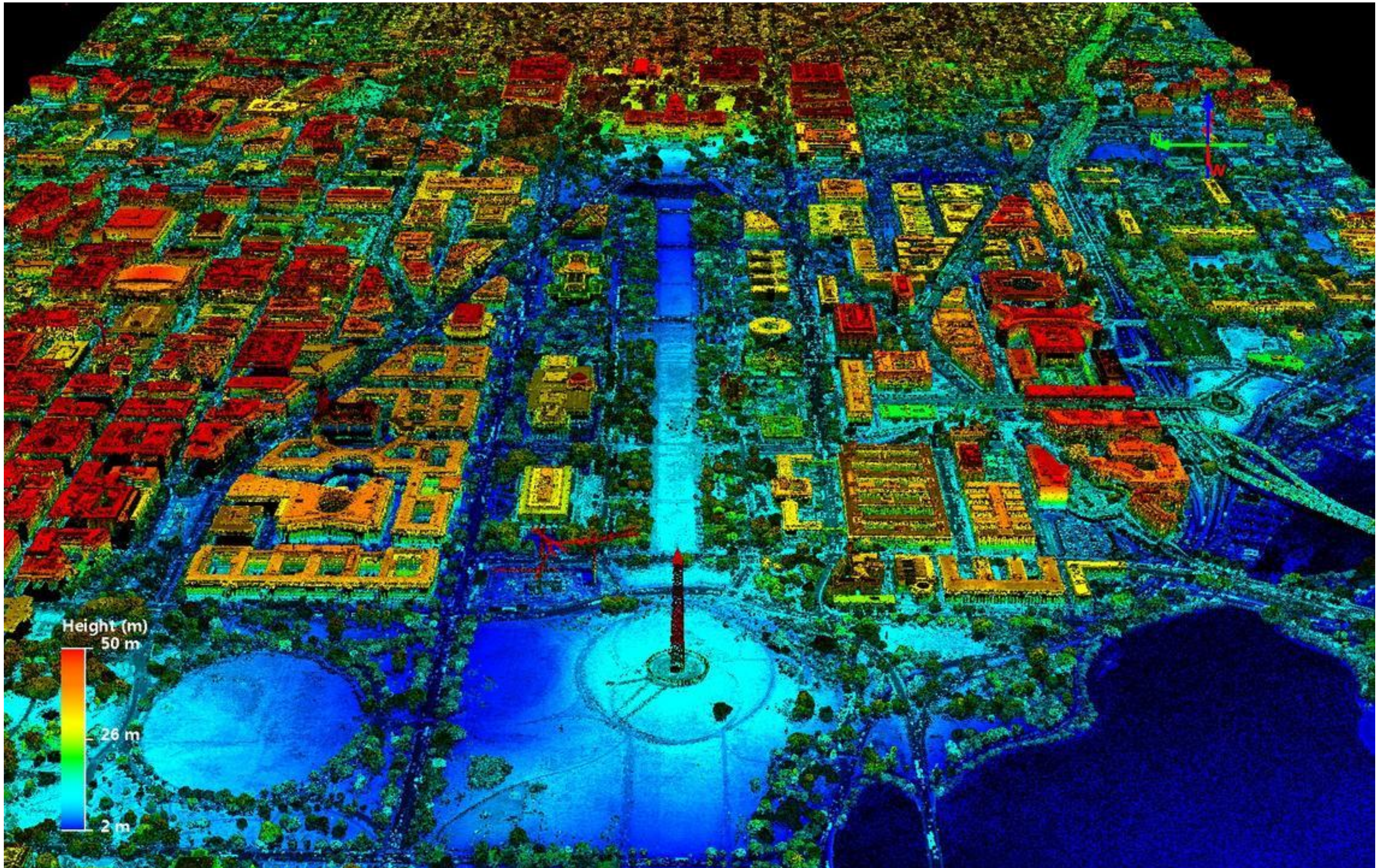
Find Interesting Phenomena in Turbulence Data

Anomaly detection



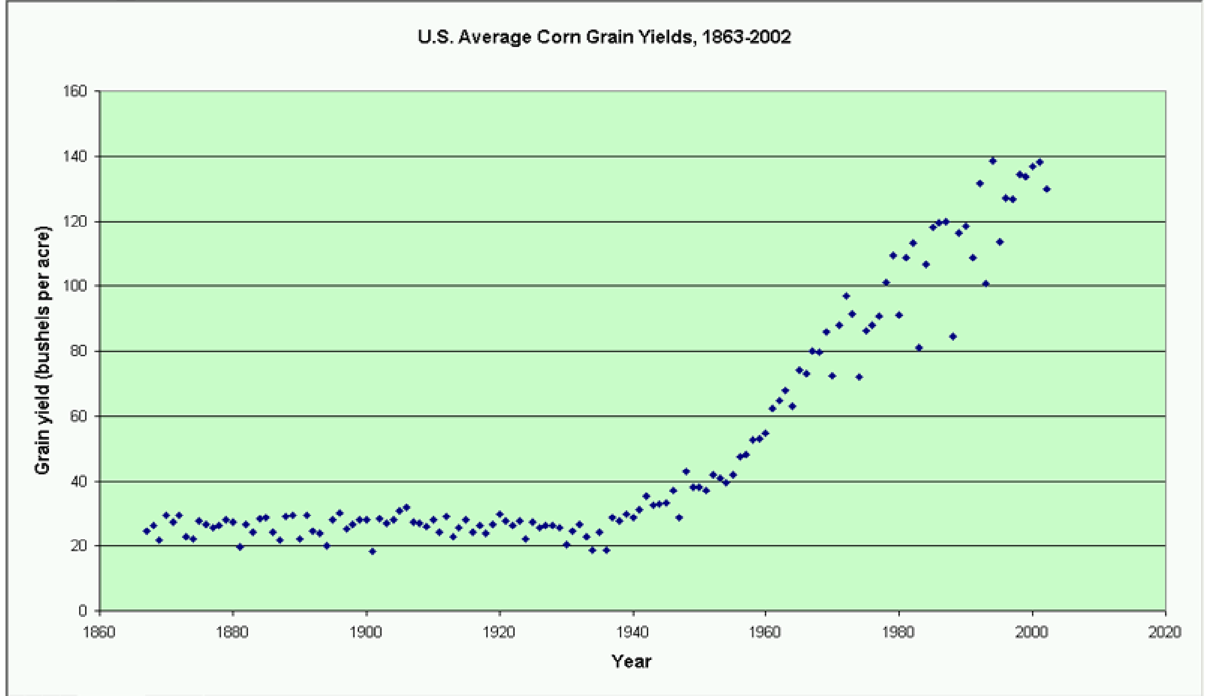
Anomaly scores

Lidar Point Cloud

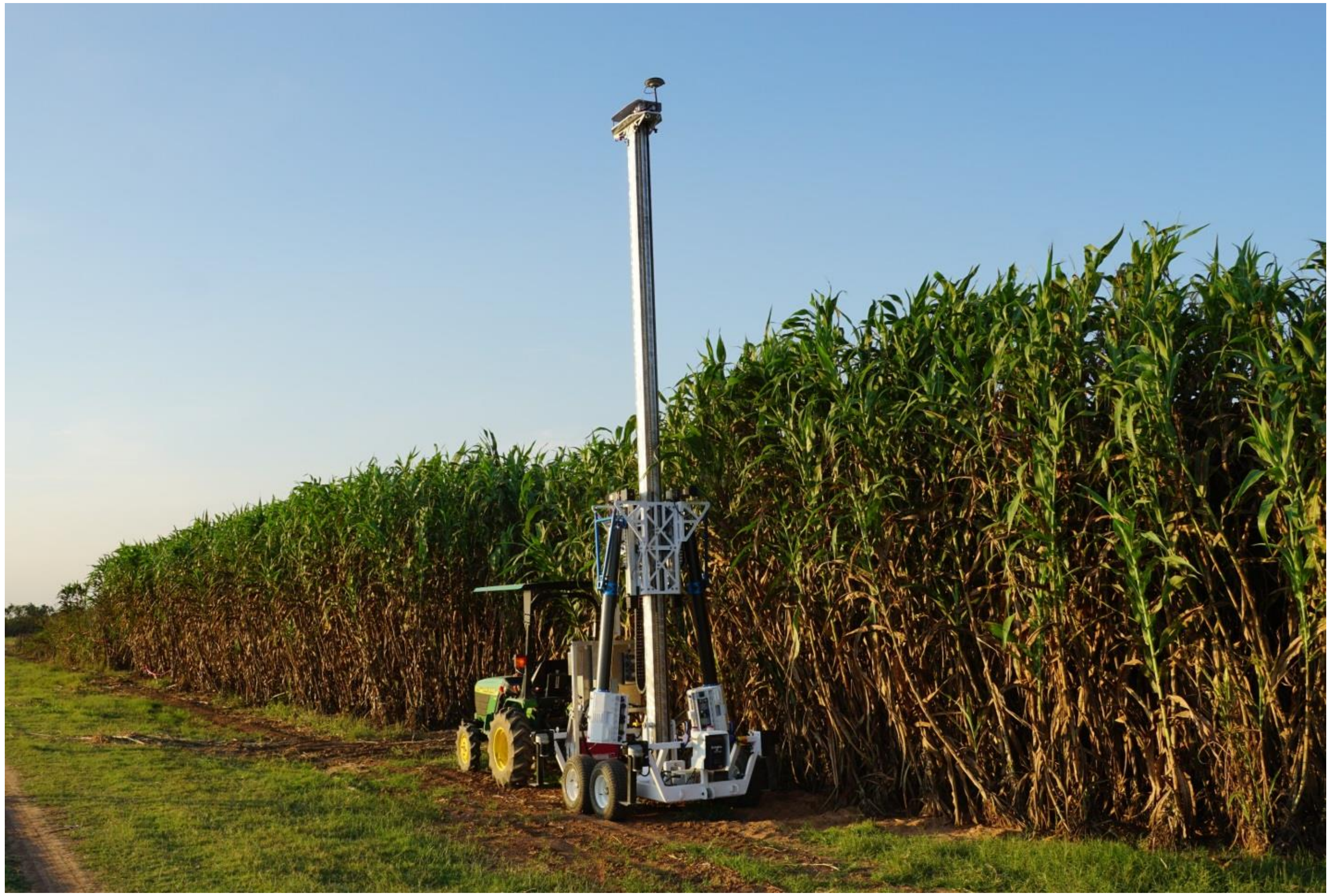


Agriculture

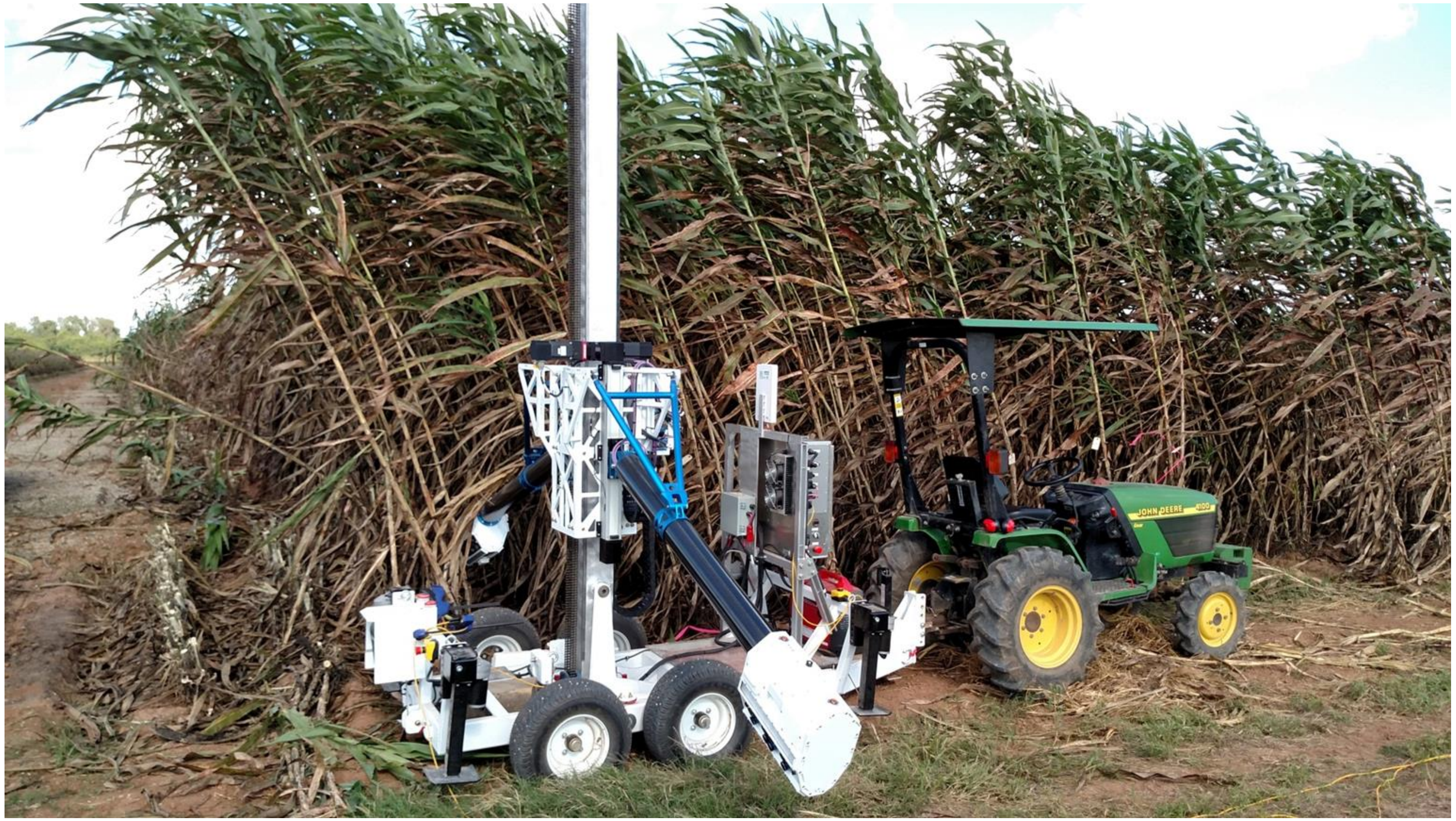
Recommend experiments (which plants to cross) to sorghum breeders.


U.S. Average Corn Grain Yields, 1863-2002
Grain yield (bushels per acre)
0
20
40
60
80
100
120
140
160
1860
1880
1900
1920
1940
1960
1980
2000
2020
Year

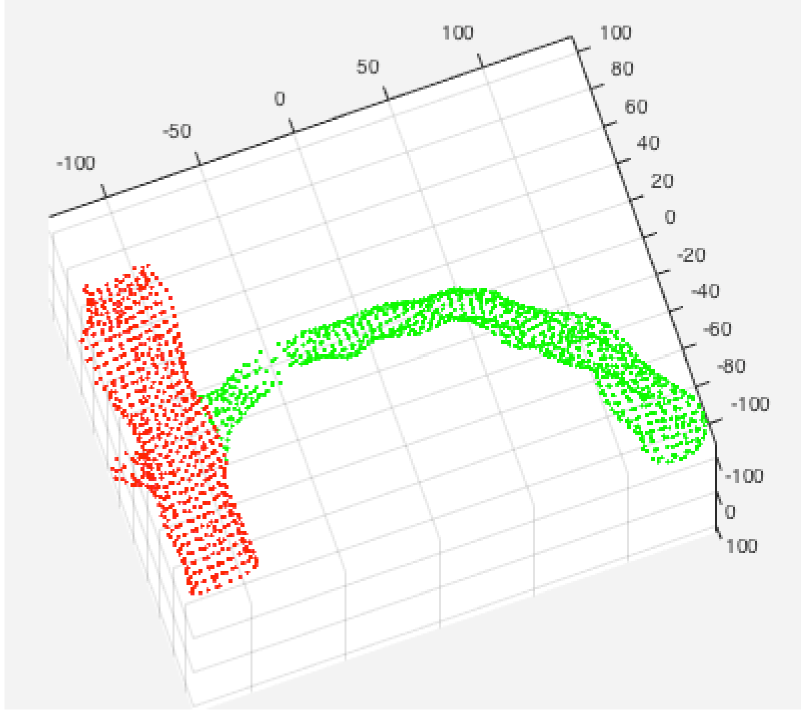
Surrogate robotic system in the field

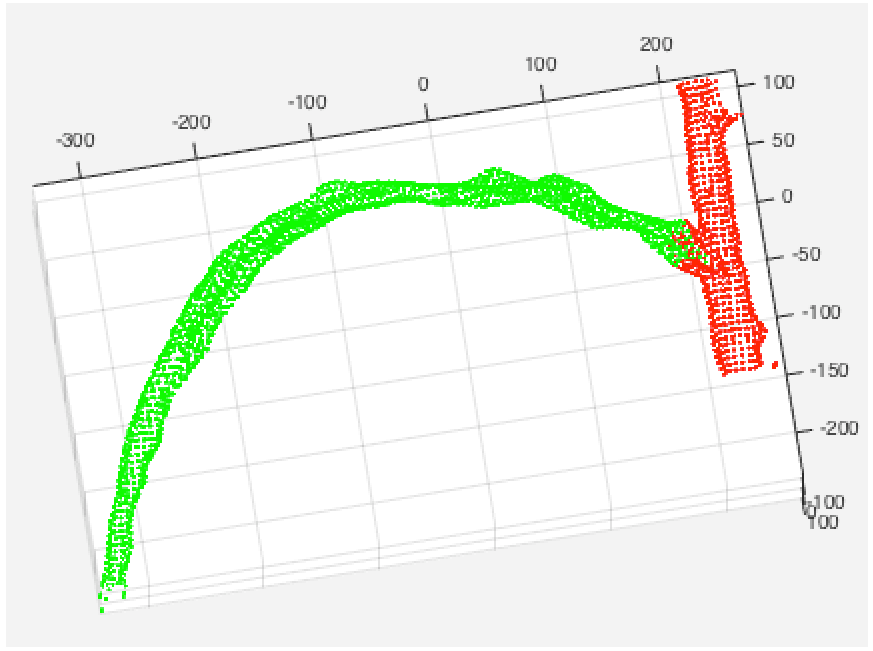


Surrogate robotic system in the field



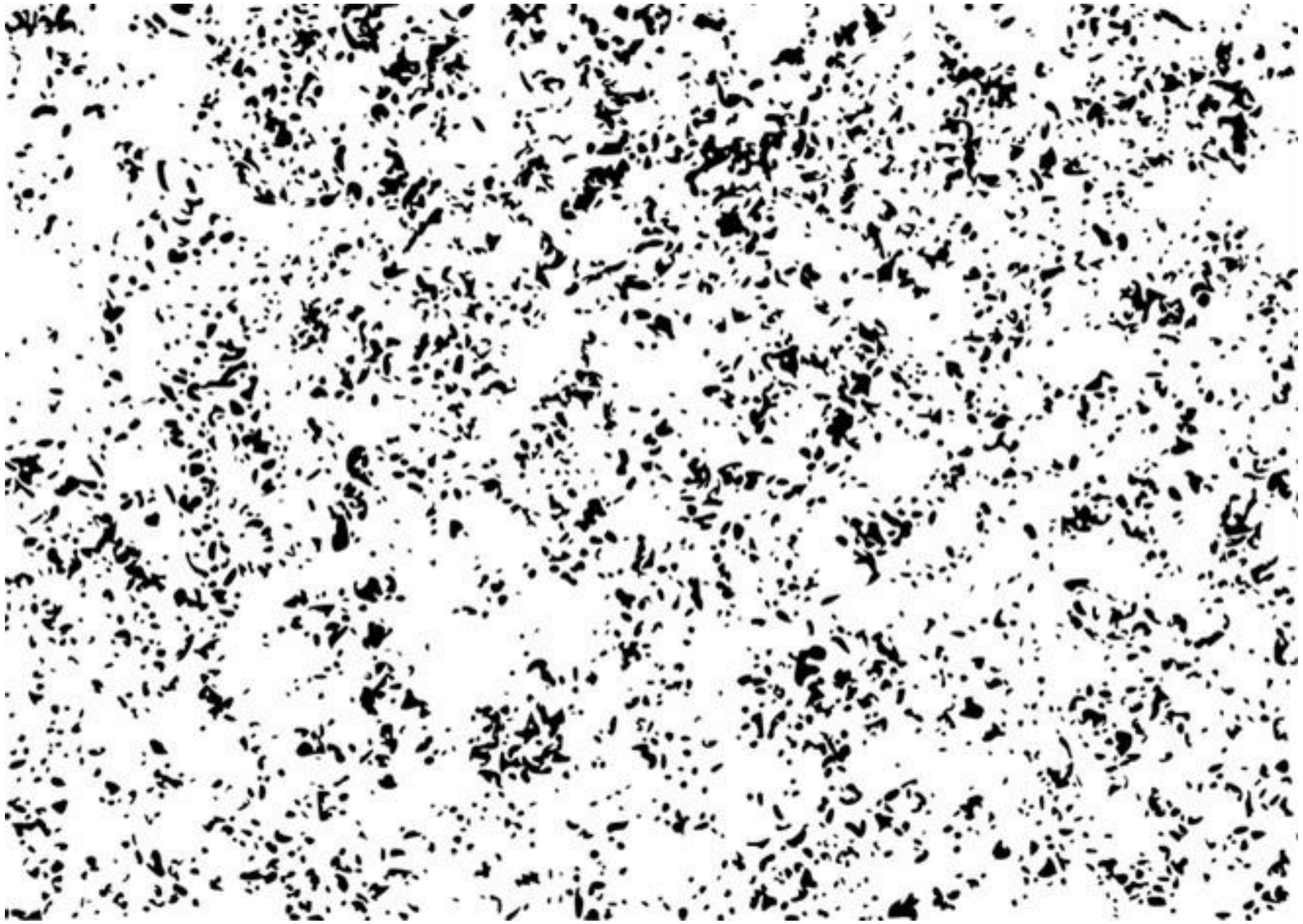
The surrogate system collecting data at the TAMU field site. The carriage supports two boom assemblies each one of which carries a sensor pod. The carriage slides up and down on the column allowing full scanning of a plant.





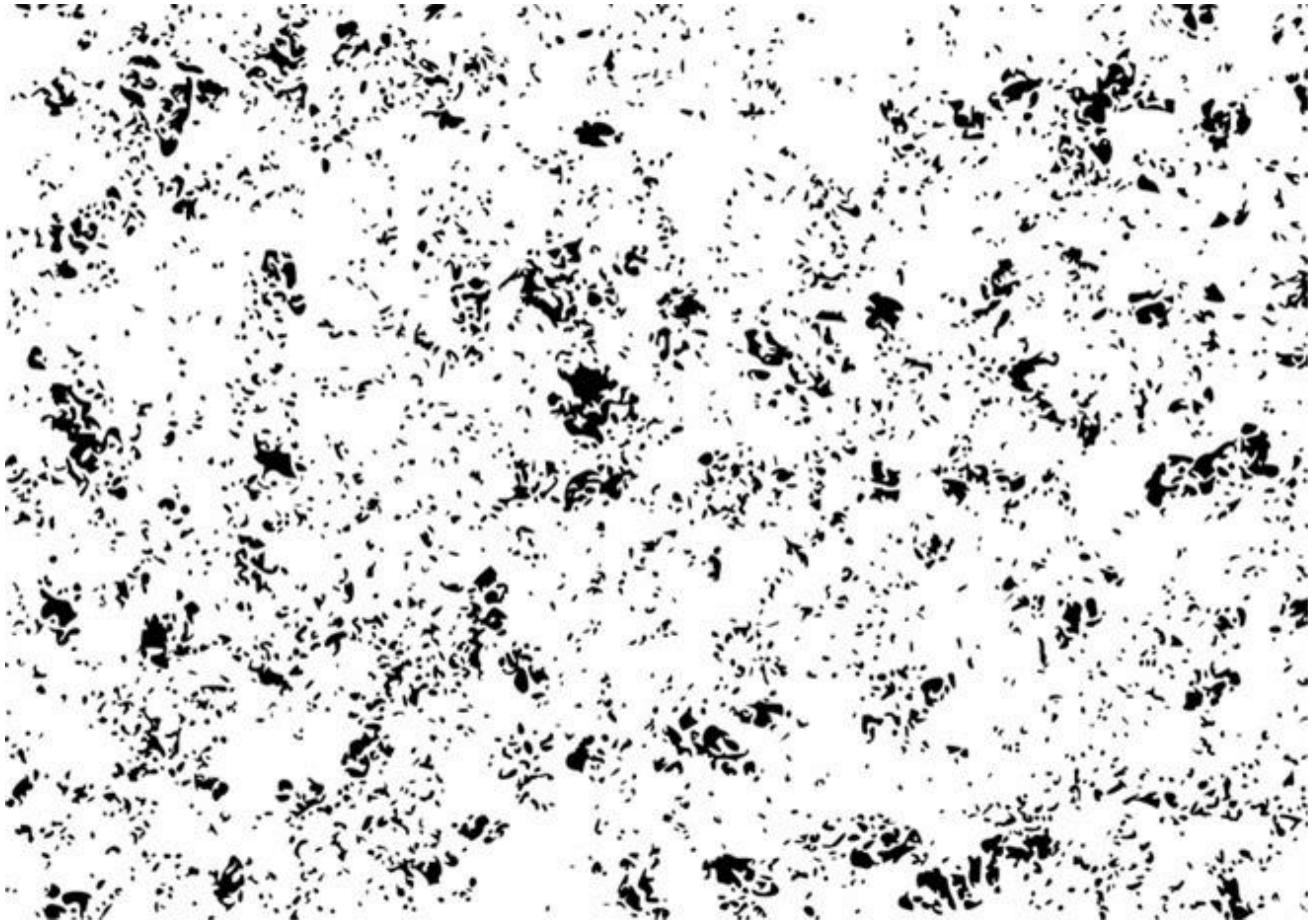
Name	Range	RMSE error
Leaf angle*	75.94	3.30 (4.35%)
Leaf radiation angle*	120.66	4.34 (3.60%)
Leaf length*	35.00	0.87 (2.49%)
Leaf width [max]	3.61	0.27 (7.48%)
Leaf width [average]	2.99	0.21 (7.02%)
Leaf area*	133.45	8.11 (6.08%)

Emerging images



Niloy J. Mitra, 2009

Emerging images



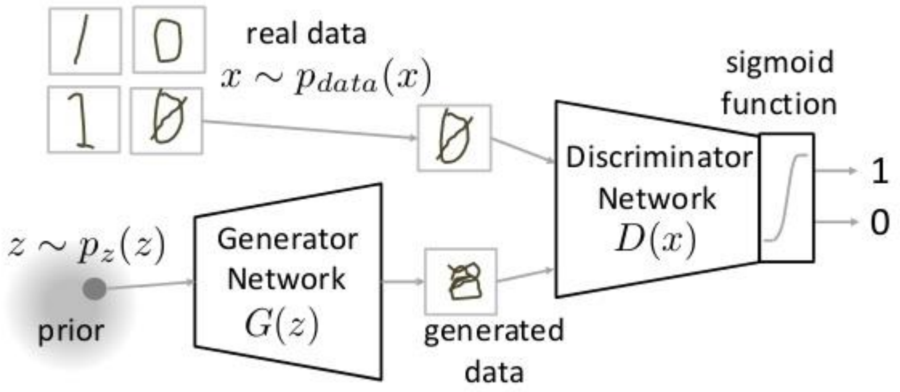
Niloy J. Mitra, 2009

Generative Adversarial Networks

Generating Adversarial Networks

$$\min_G \max_D V(D, G)$$

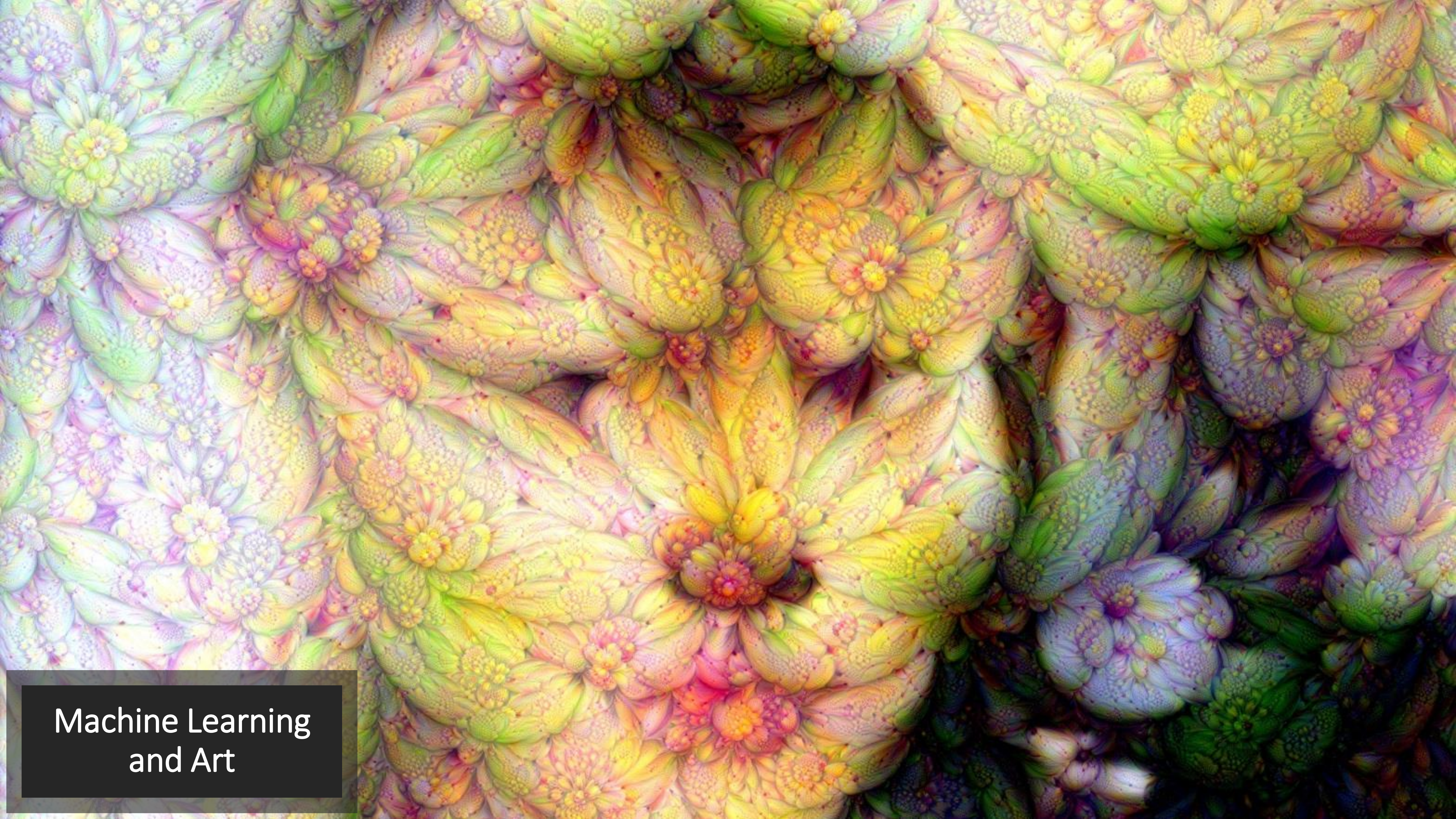
$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$



Generated fake celebrity images

Tero Karras, Timo Aila, Samuli Laine, ICLR 2018

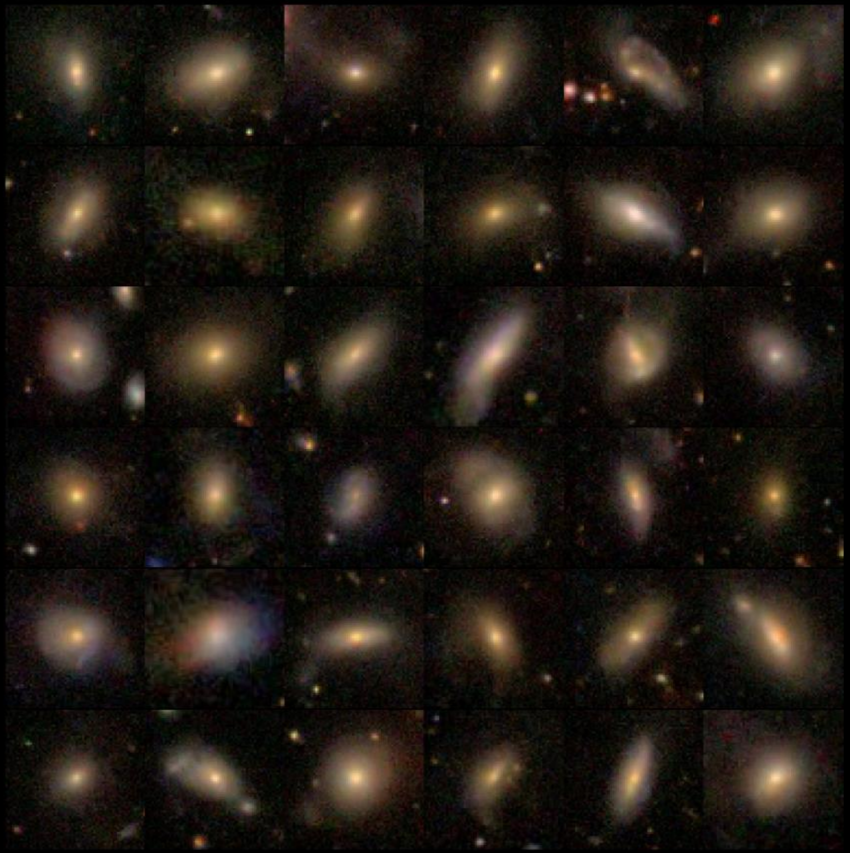
Generated fake living room images



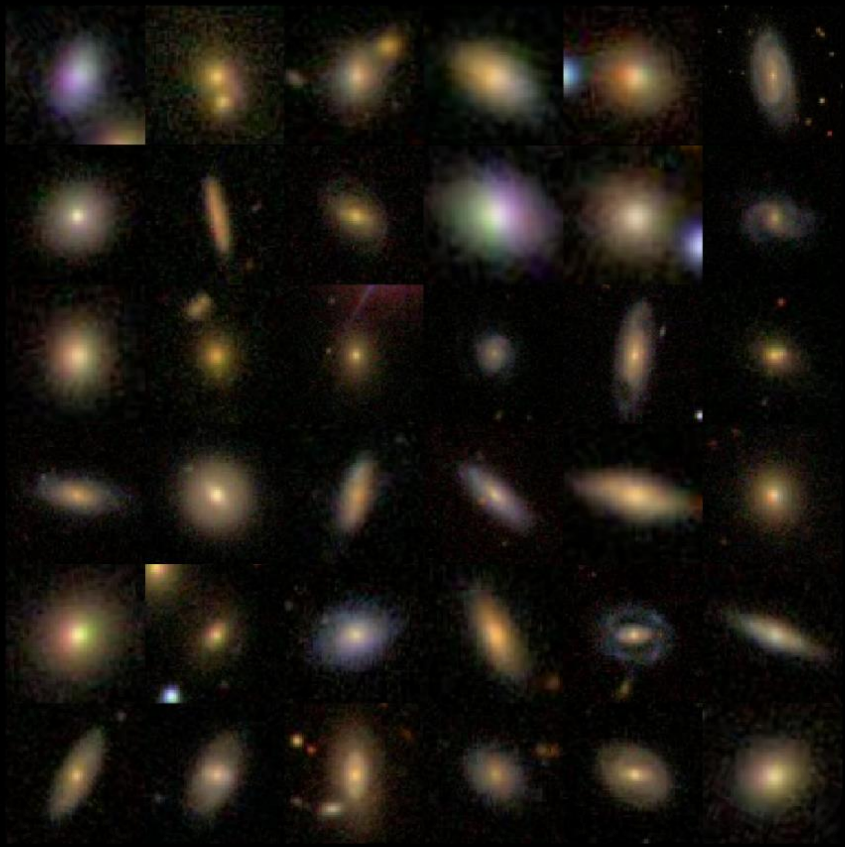
Machine Learning and Art

Ravanbakhsh, M., Lanusse, F., Mandelbaum, R., Schneider, J., and Poczos, B., AAAI 2017

visual Turing test

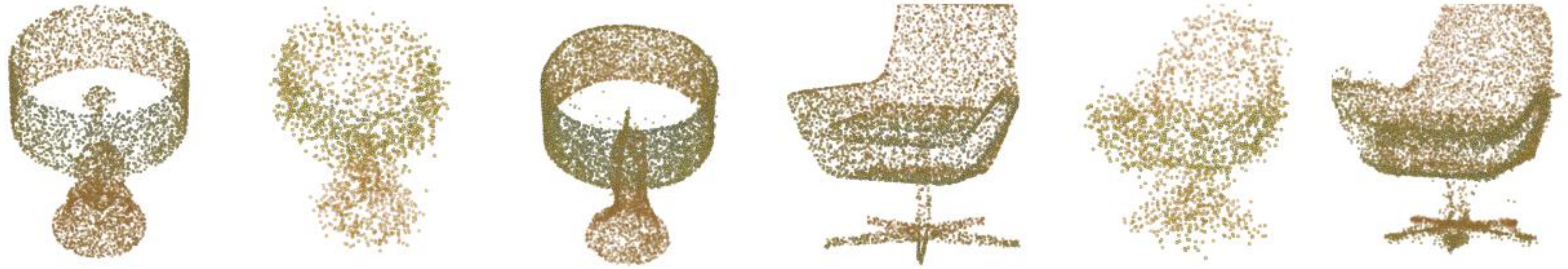


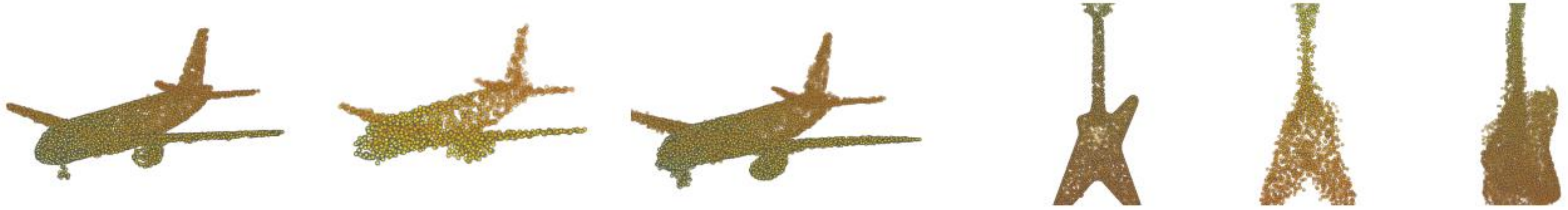
Mock - PixelCNN



Real - SDSS

Point Cloud GAN


(a) Lamp (b) Chair


(c) Plane (d) Guitar

For each object, from left to right is training data, Auto Encoder competitor, and PC-GAN. PC-GAN is better in capturing fine details like wheels of the airplanes or proper chair legs.

Theory of Generative Adversarial Networks

Goal: Suppose we observe n IID samples $X_1, ..., X_n \overset{IID}{\sim} P$ from a distribution P that is unknown but is assumed to lie in some regularity class $\mathcal{P}$.

We are interested in constructing an estimator $\widehat{P}^* : \{X_1, \ldots, X_n\} \to \mathcal{P}$ of P

with the goal of solving:

$$\widehat{P}_n^* \doteq \arg \min_{\widehat{P} \in \mathcal{F}_g} \sup_{f \in \mathcal{F}_d} \left| \mathbb{E}_{X \sim P} [f(X)] - \mathbb{E}_{X \sim \widehat{P}(X_1,...,X_n)} [f(X)] \right|$$

where $\mathcal{F}_d$ and $\mathcal{F}_g$ are called the *discriminator* and *generator* classes.

Question: When does $\lim_{n \to \infty} d(\widehat{P}_n^*, P) = 0$?

Question: What is the convergence rate of the convergence?

Statistical Results

Under some conditions

- $\mathcal{F}_d$ and $\mathcal{F}_g$ are Besov spaces
- the neural networks are large enough (i.e. can approximate functions in the above Besov places well)
- the neural networks are fully connected using ReLU activation function.
- We solve the minmax optimization problem perfectly,

Then the GAN is consistent and minimax optimal

Open Questions

- Statistical properties under less restrictive conditions?
- Results for convolutional neural nets?
- Best way for training GANs (i.e best way for solving the minmax optimization)?
- GANs on manifolds?
- Rare event generation?
- Generate uniform distribution on the support of the data?

Experiment Design

Experiment Design

Given a collection of results from past experiments, recommend new experiments to run.

- some experiments are expensive (time, money) and high-quality, some are cheap and low-quality.
- consider costs and expected information gain

Code available: https://github.com/dragonfly/dragonfly

Gaussian Processes

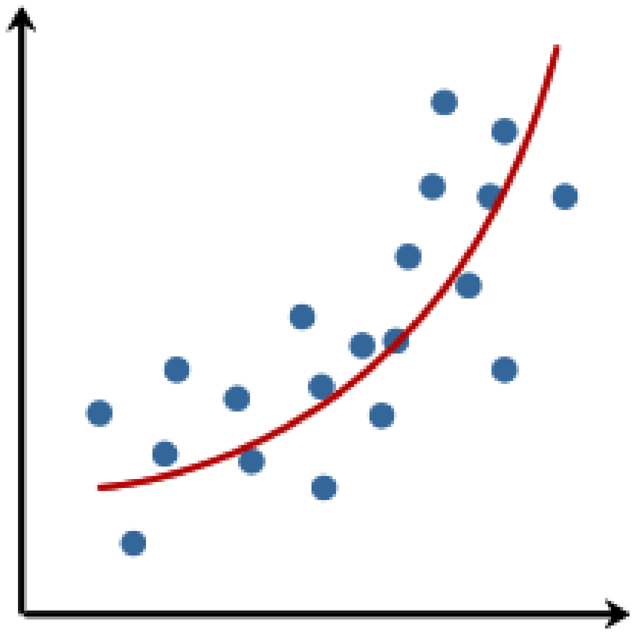
Why GPs for Regression?

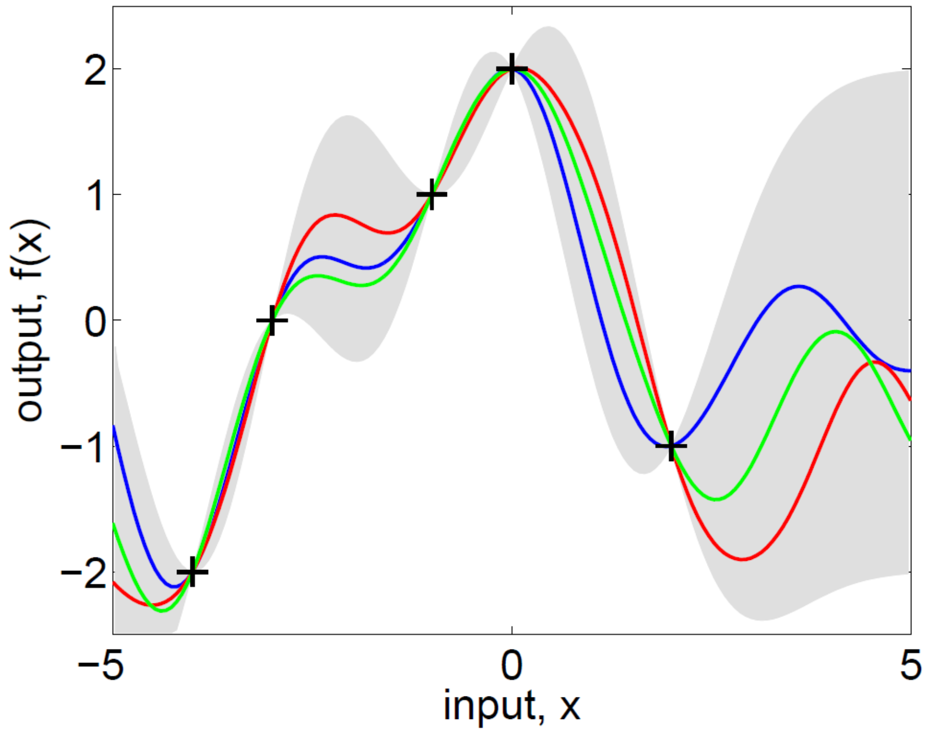
Regression methods:

Linear regression, support vector regression, kNN regression, deep neural networks, etc…

Motivation:

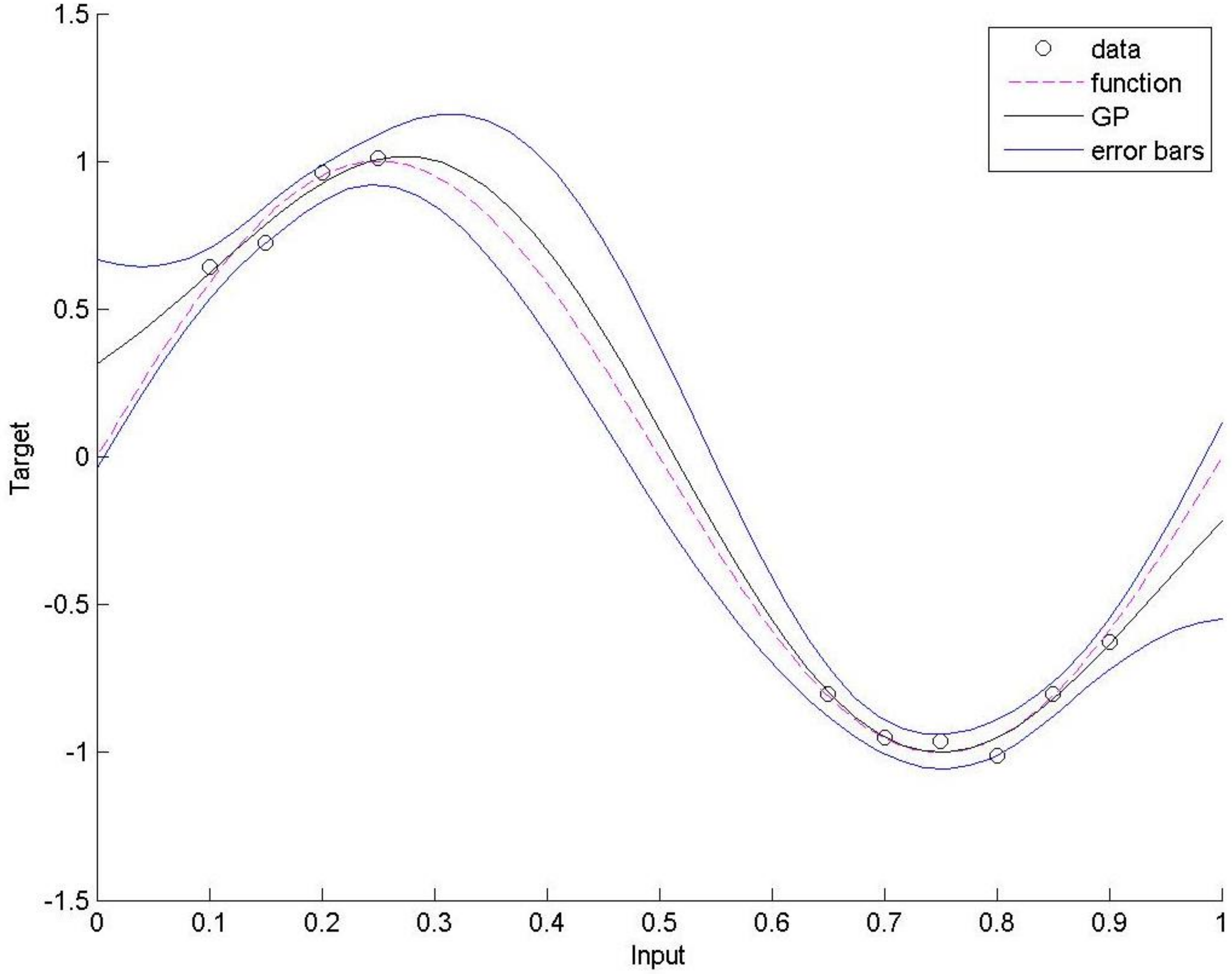
All the above regression methods give point estimates only. We would like a method that could also provide “confidence” during the estimation.





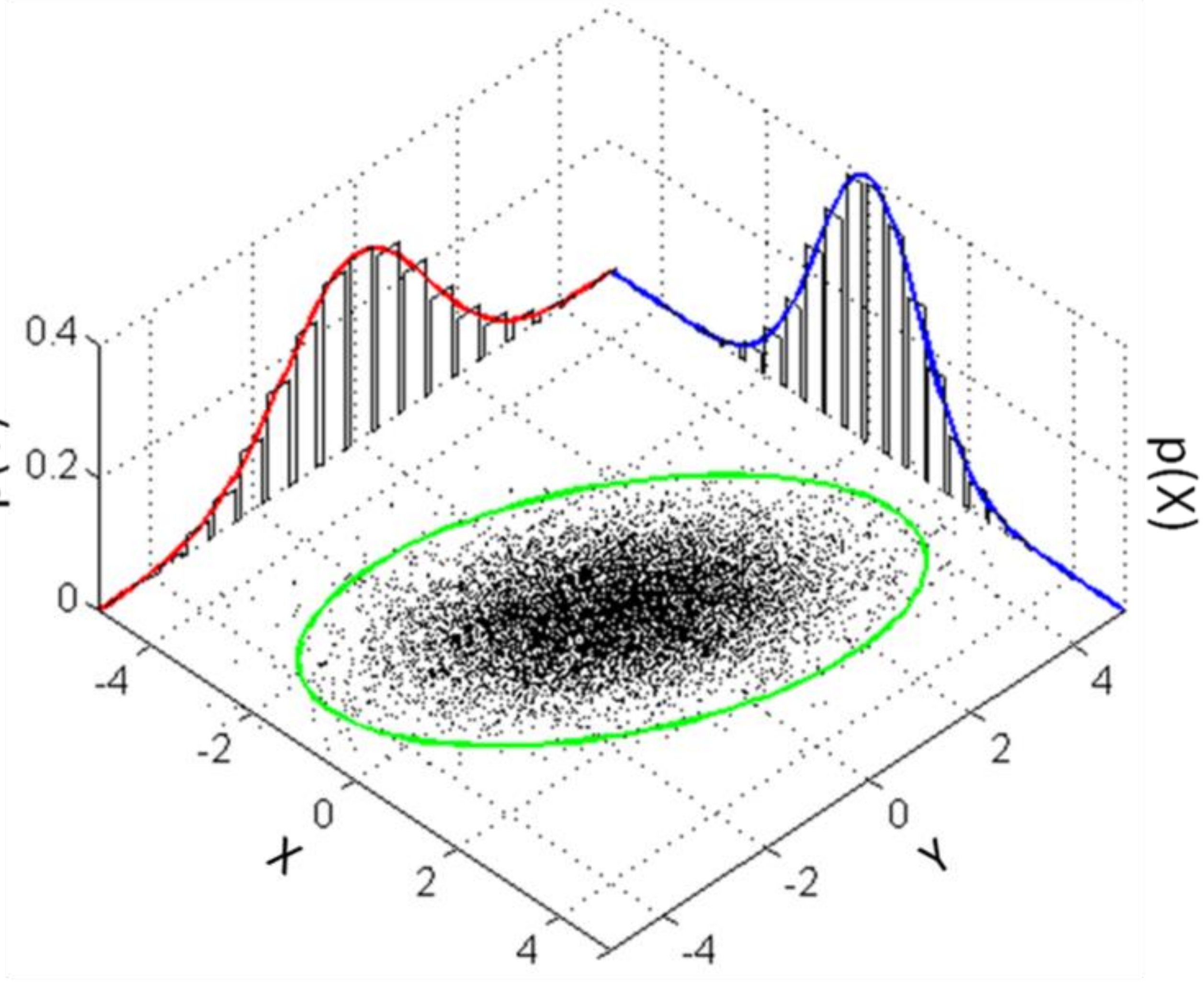
Sampling from GP: We can also use GP the generate samples from the posterior (red, blue, green)

Results: Sin function



Properties of Multivariate Gaussian Distributions

Marginal and Conditional distributions of Gaussians are Gaussian



$$p(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left\{\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Gaussian Processes

Definition: (Gaussian Processes)

GP is an (uncountably infinite) collection of random variables, s.t. any finite number of them have a joint Gaussian distribution

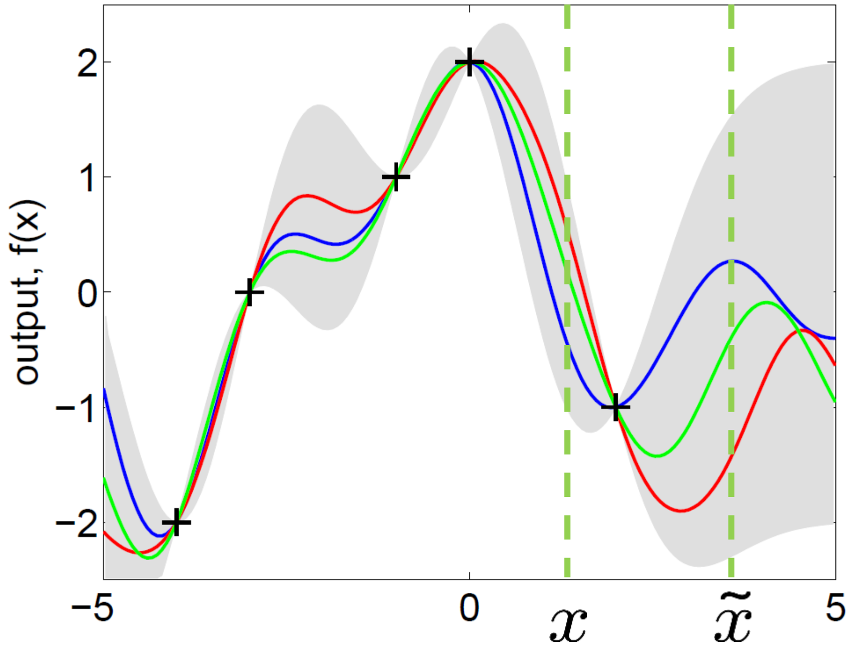
Notations:

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \tilde{\mathbf{x}})) \in \mathbb{R},\ \mathbf{x} \in \mathbb{R}^D$$

$$m(\mathbf{x}) = \mathbb{E}[f(x)] \in \mathbb{R},\ \text{(mean function)}$$

$$k(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbb{E}[(f(x) - m(\mathbf{x}))(f(\tilde{\mathbf{x}}) - m(\tilde{\mathbf{x}}))^T] \in \mathbb{R}$$

$$\text{(covariance function)}$$



Gaussian Process Pseudo Code

Inputs:

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \in \mathbb{R}^{n \times D},\ n \text{ training inputs}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^{n},\ n \text{ training targets}$$

$k(\cdot,\cdot) : \mathbb{R}^{D \times D} \to \mathbb{R}$ covariance function (kernel)

$\mathbf{x}_*$ test input

σ^2 noise level on the observations

$$[y(\mathbf{x}) = f(\mathbf{x}) + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2)]$$

Gaussian Process Pseudo Code (Continued)

1., $K \in \mathbb{R}^{n \times n}$ Gram matrix. $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

$$k(\mathbf{x}_*) = k_* = k(X, \mathbf{x}_*) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_*) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_*) \end{bmatrix} \in \mathbb{R}^n$$

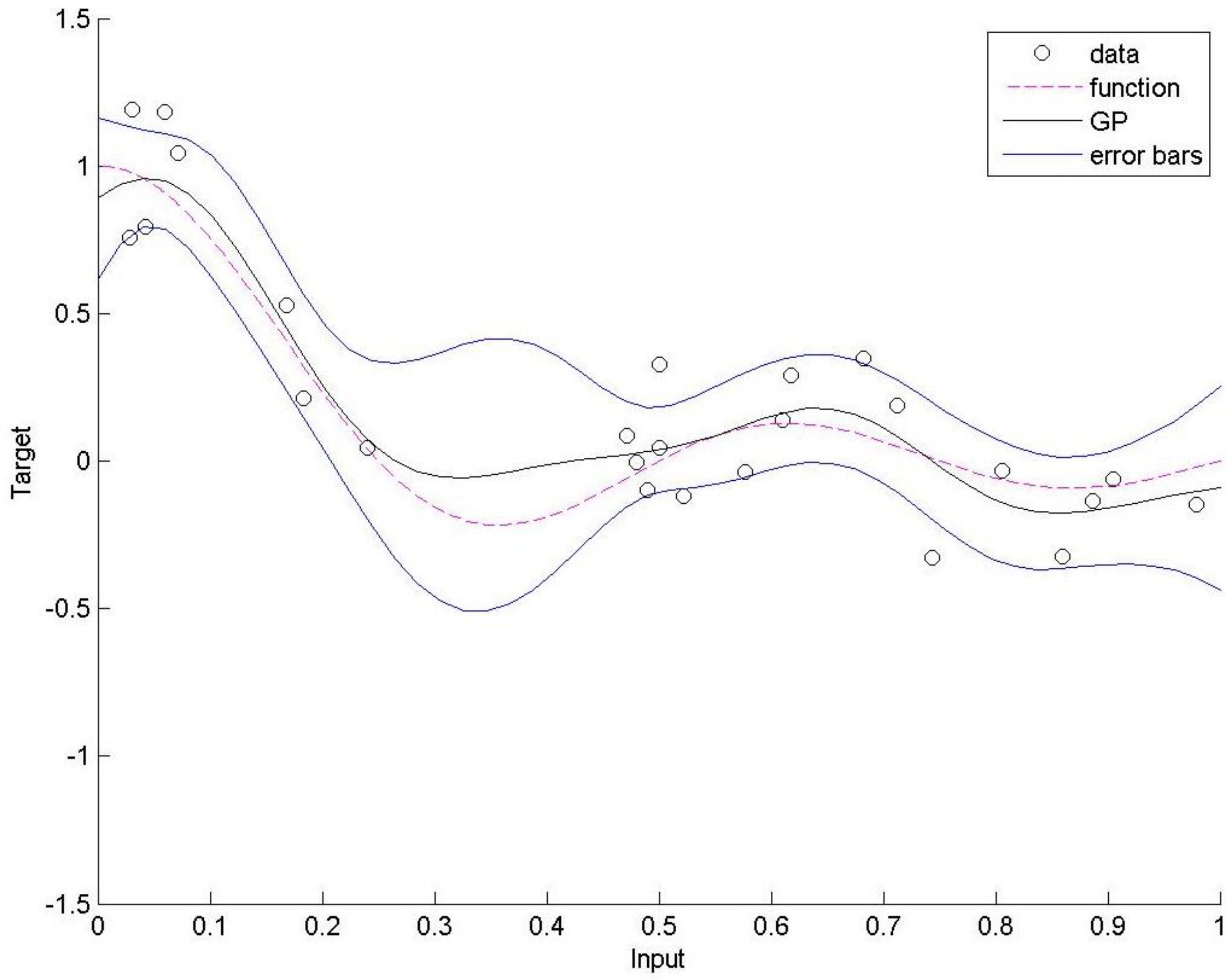
2., $\boldsymbol{\alpha} = (K + \sigma^2 \mathbf{I}_n)^{-1} \mathbf{y} \in \mathbb{R}^n$

3., $\bar{f}_* = k_*^T \boldsymbol{\alpha} \in \mathbb{R}$

4., $cov(f_*) = \underbrace{k(\mathbf{x}_*, \mathbf{x}_*)}_{\mathbb{R}} - \underbrace{k_*^T}_{\mathbb{R}^{1 \times n}} \underbrace{[K + \sigma^2 I_n]^{-1}}_{\mathbb{R}^{n \times n}} \underbrace{k_*}_{\mathbb{R}^n} \in \mathbb{R}$

Outputs: $\bar{f}_*$, $cov(f_*)$

Results: Sinc function



Take me home!

GPs can be used for

- nonlinear regression
- can provide "confidence" of the estimate
- and can sample from the posterior distribution of regression functions

GP Application:
Bayesian Optimization

Bayesian Optimization

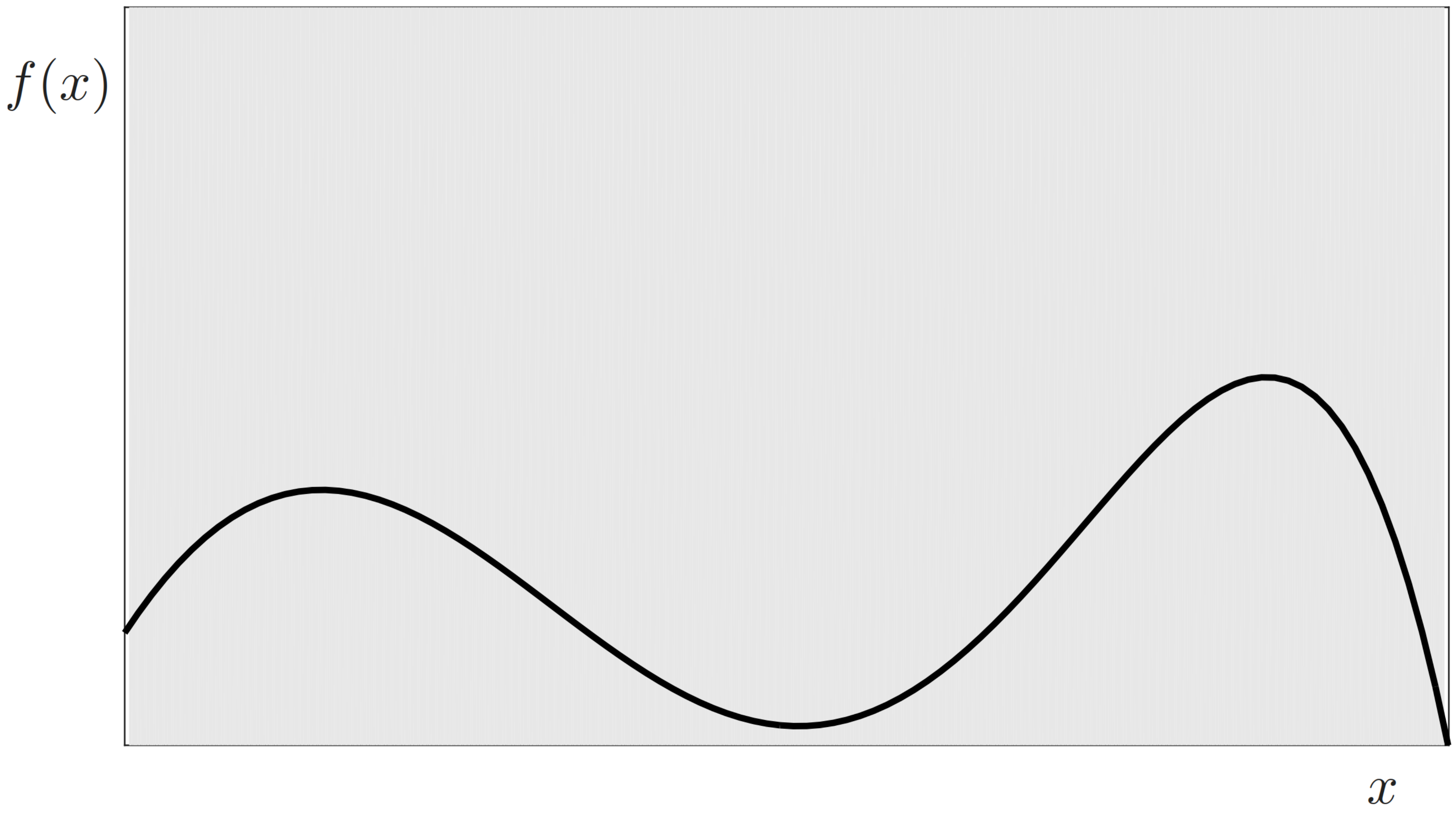
- Bayesian optimization (BO) is a popular recipe for
  - optimizing expensive black-box functions
  - where the goal is to find a global maximizer of the black-box objective function
  - We don’t even need to know the gradient of the objective function

- Bayesian optimization has been used for a variety of practical optimization tasks such as
  - hyperparameter tuning for machine learning algorithms,
  - experiment design,
  - online advertising,
  - …

- Often heuristics only
  - Big gaps in our theoretical understanding …
    - Convergence rate?
    - Dependence on dimension?

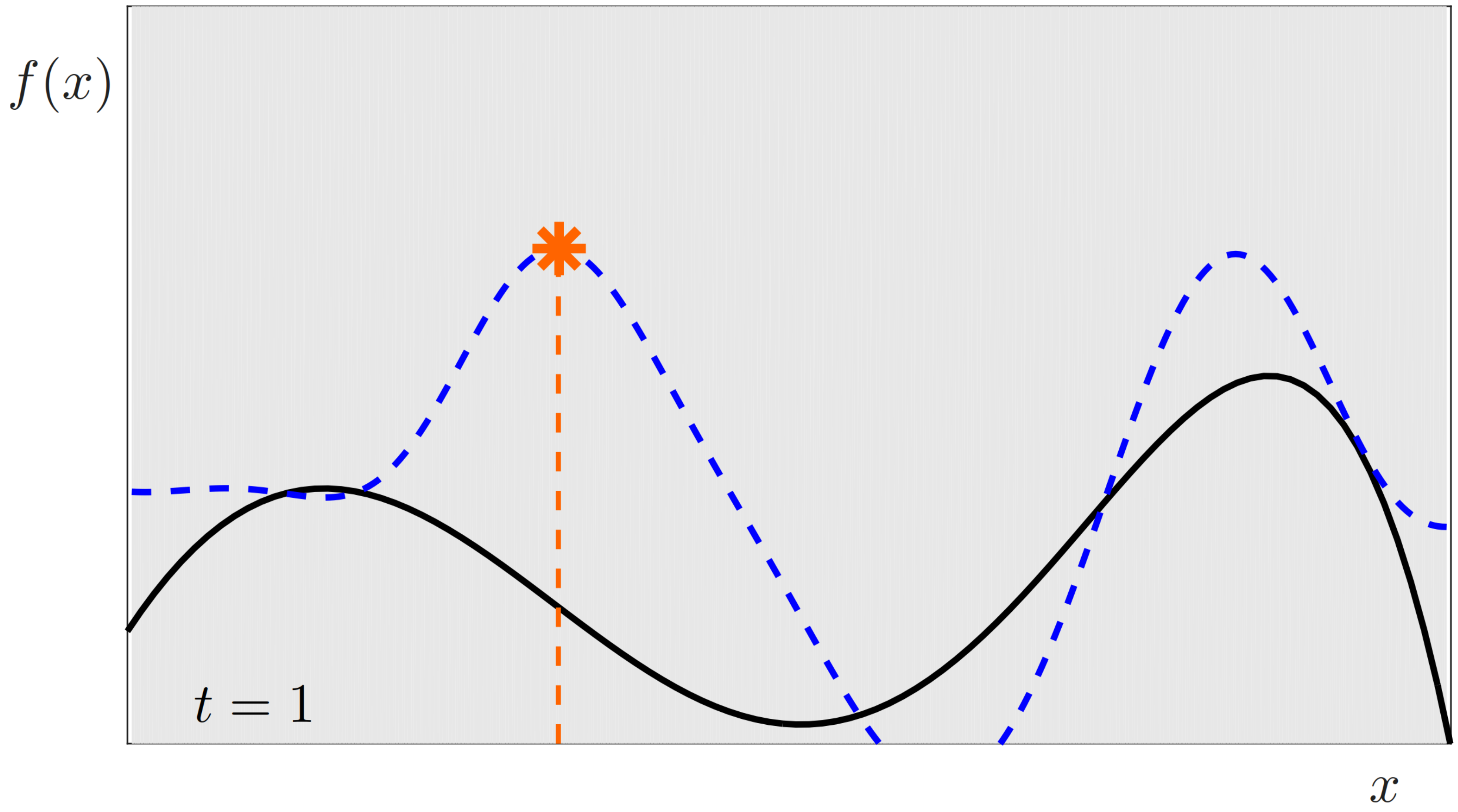
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



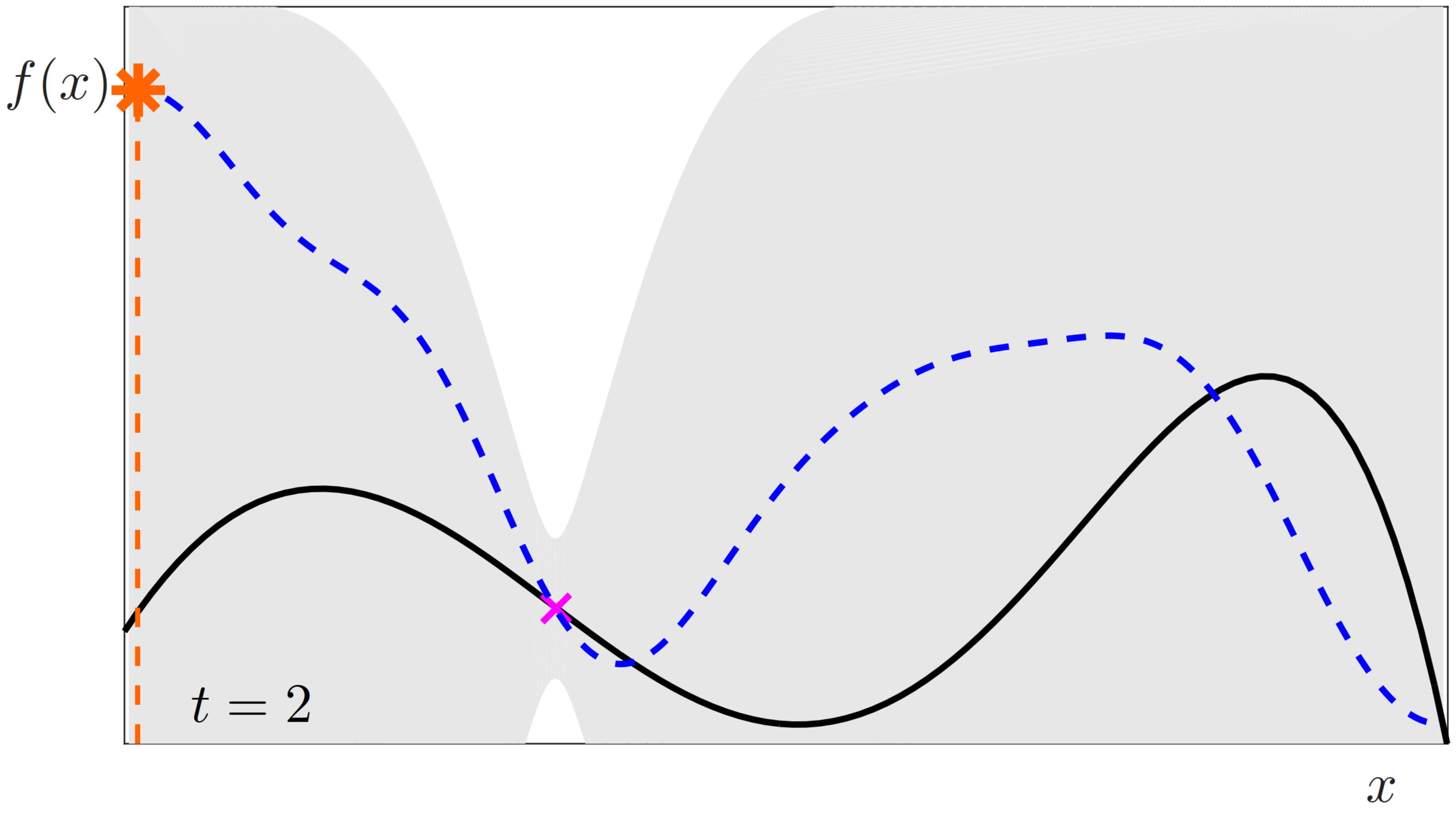
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



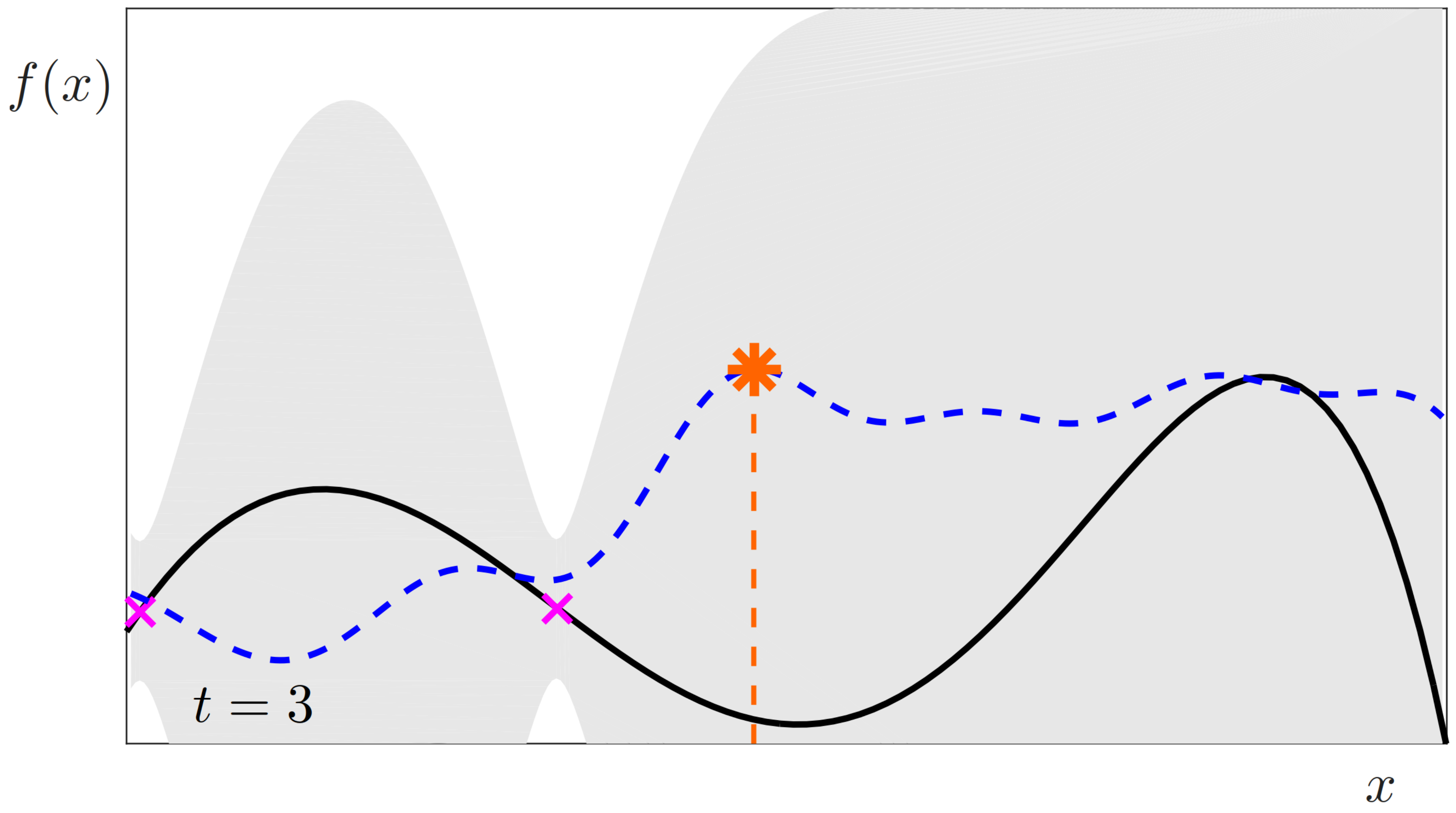
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



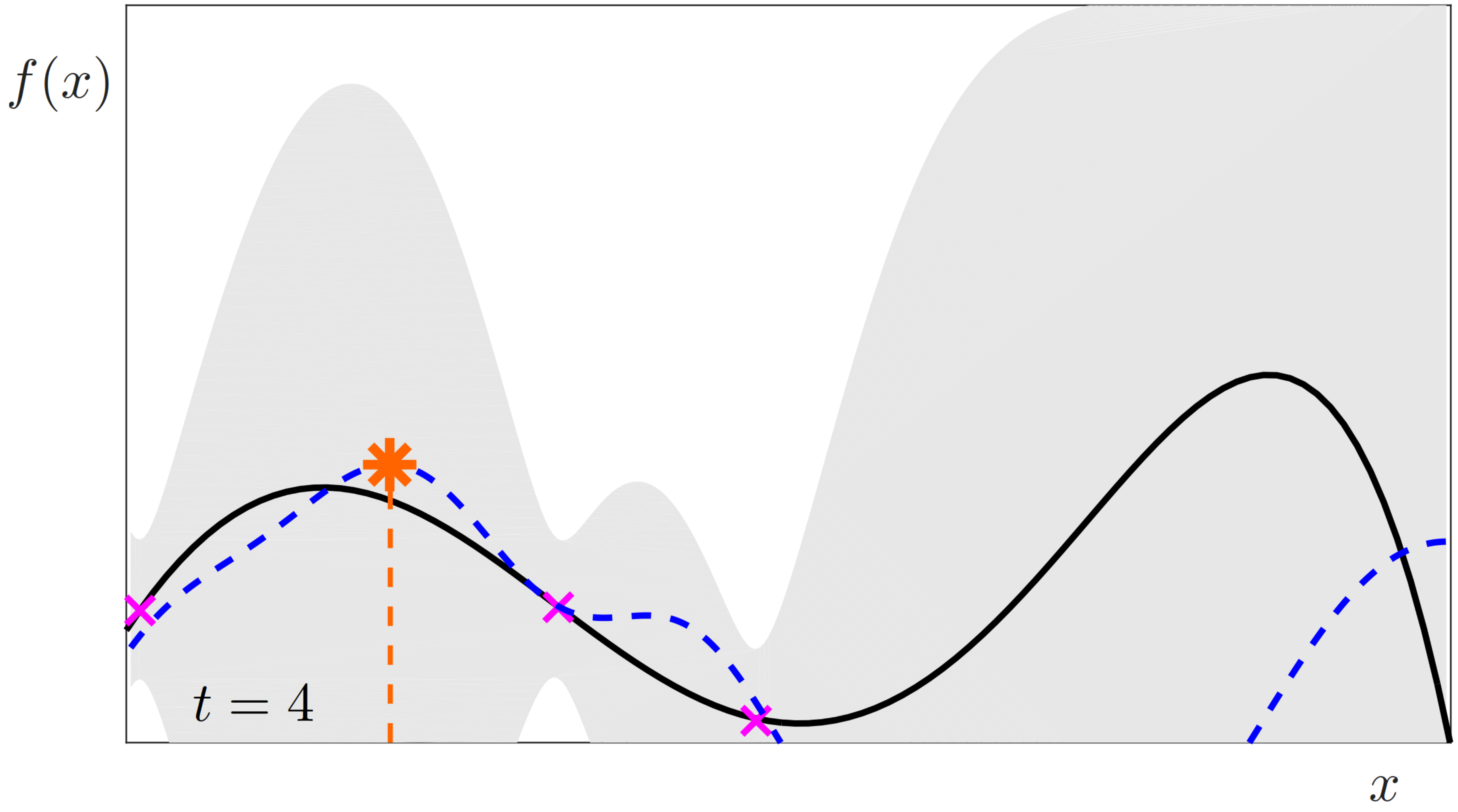
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



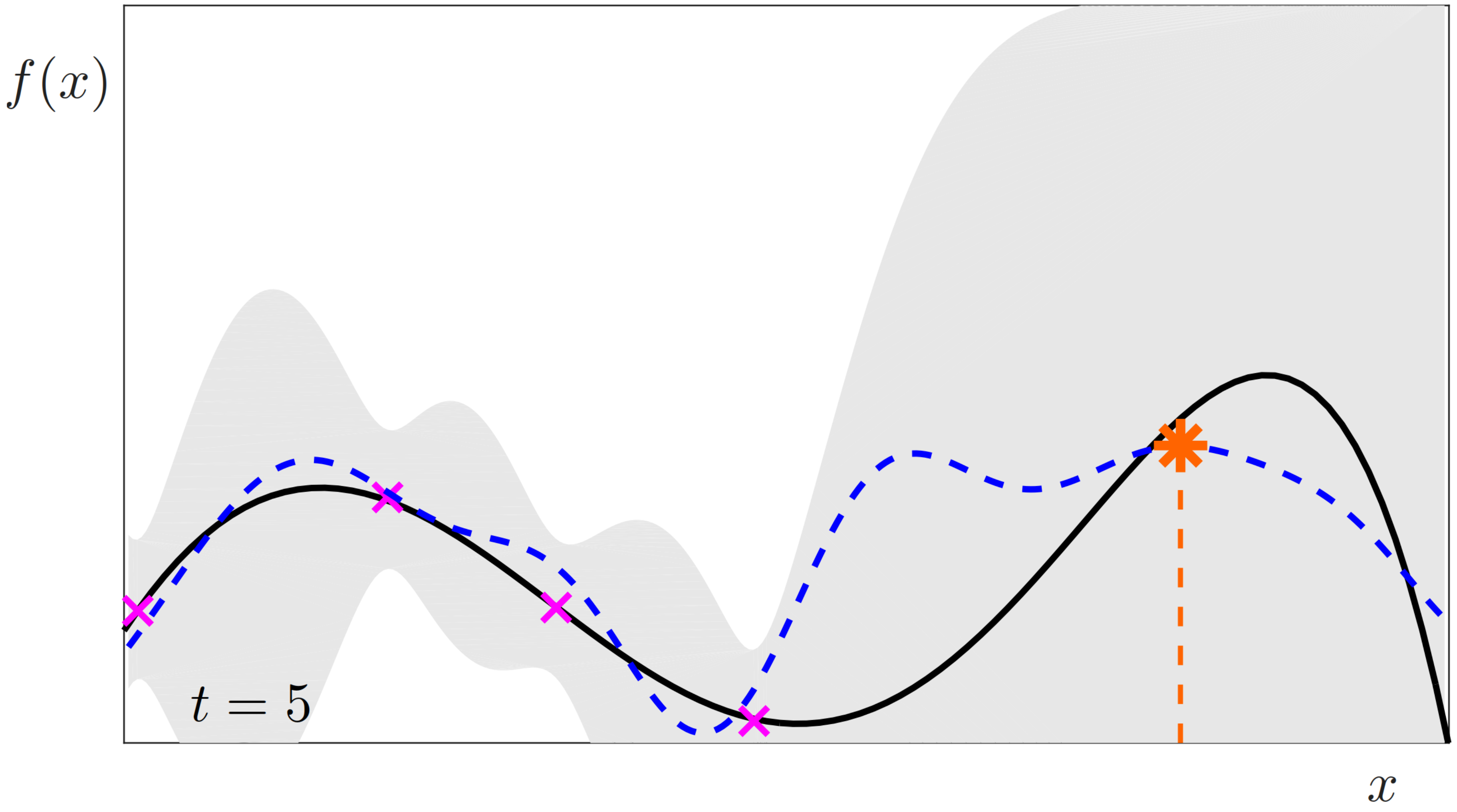
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



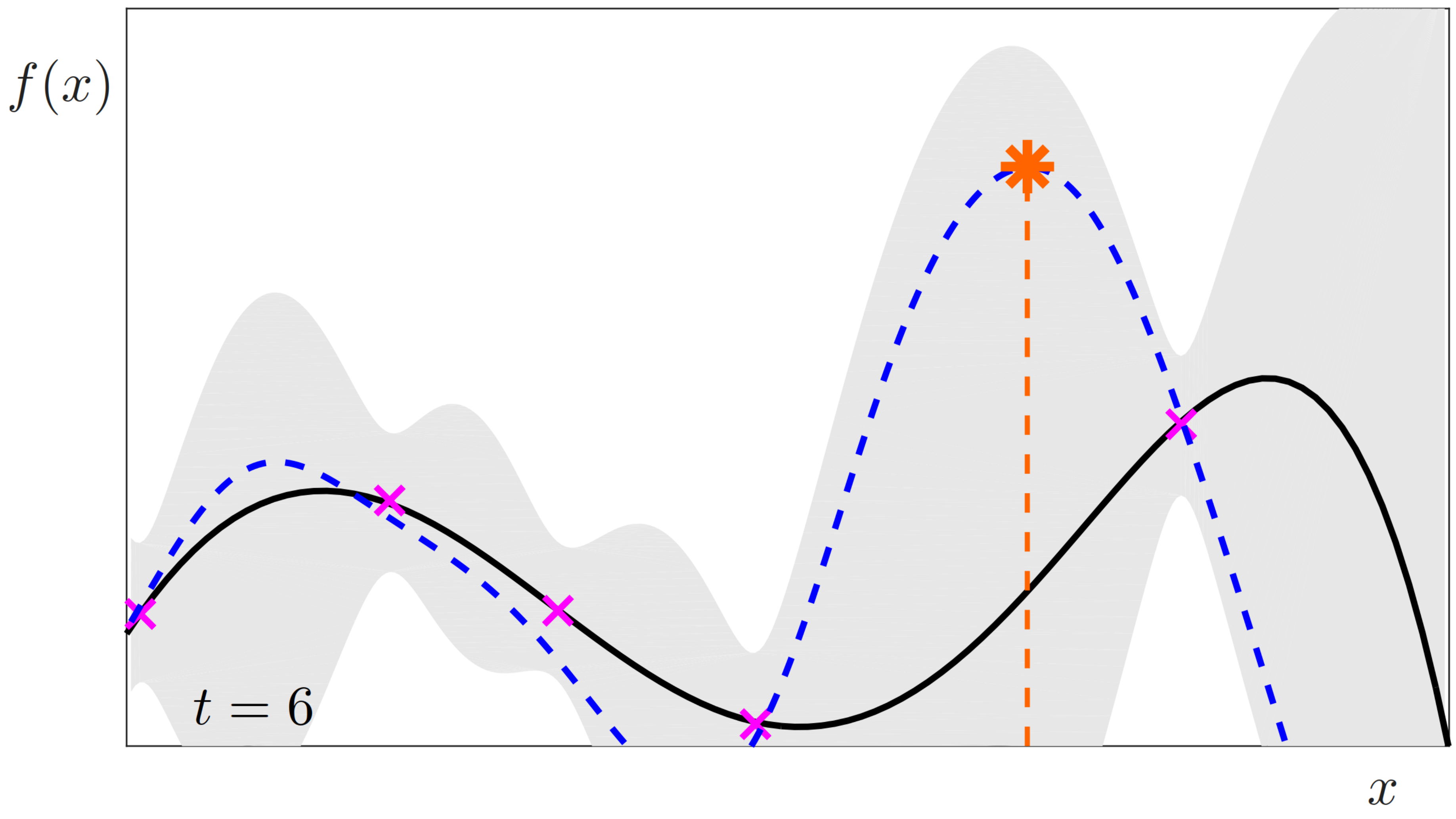
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



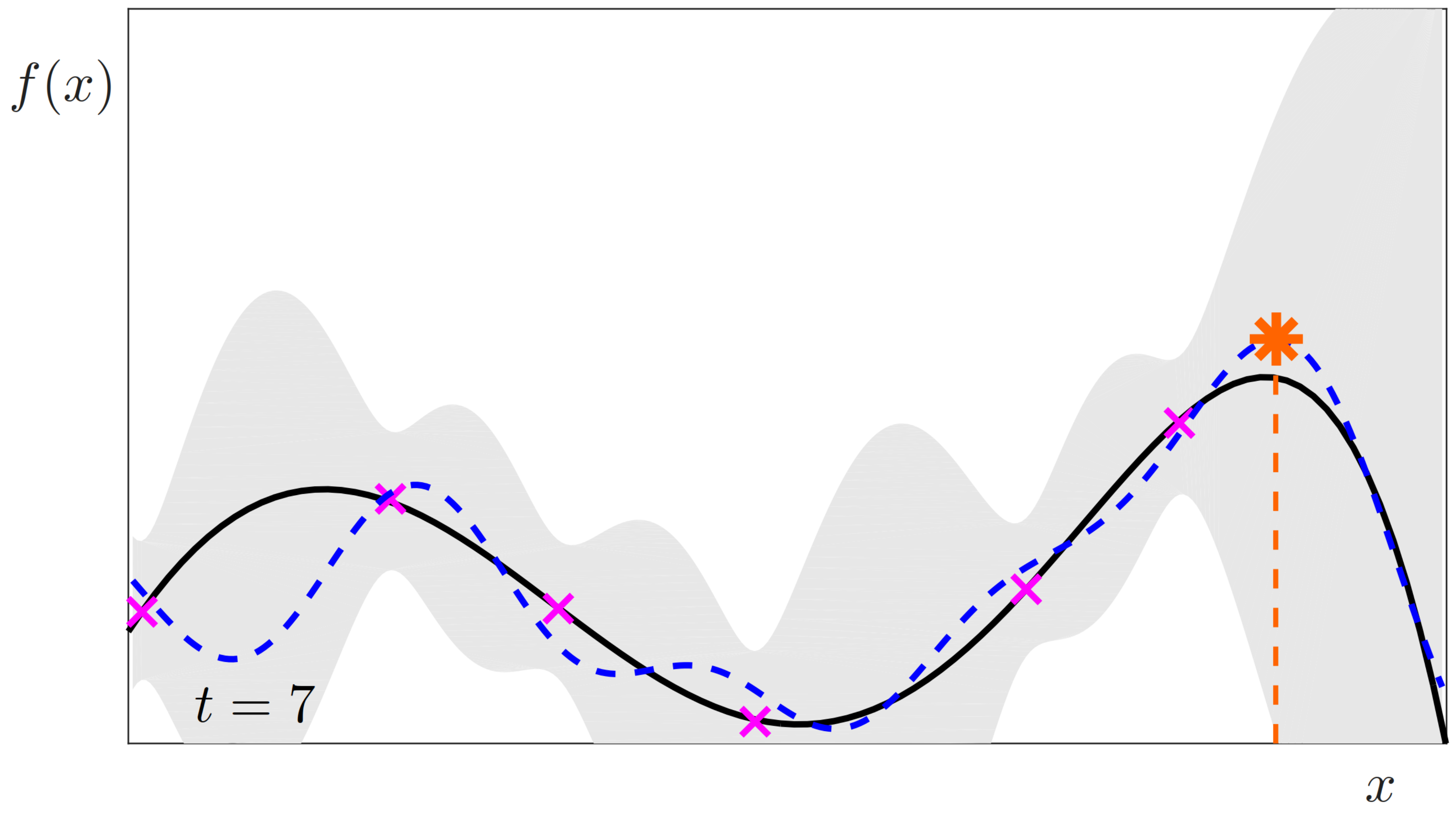
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



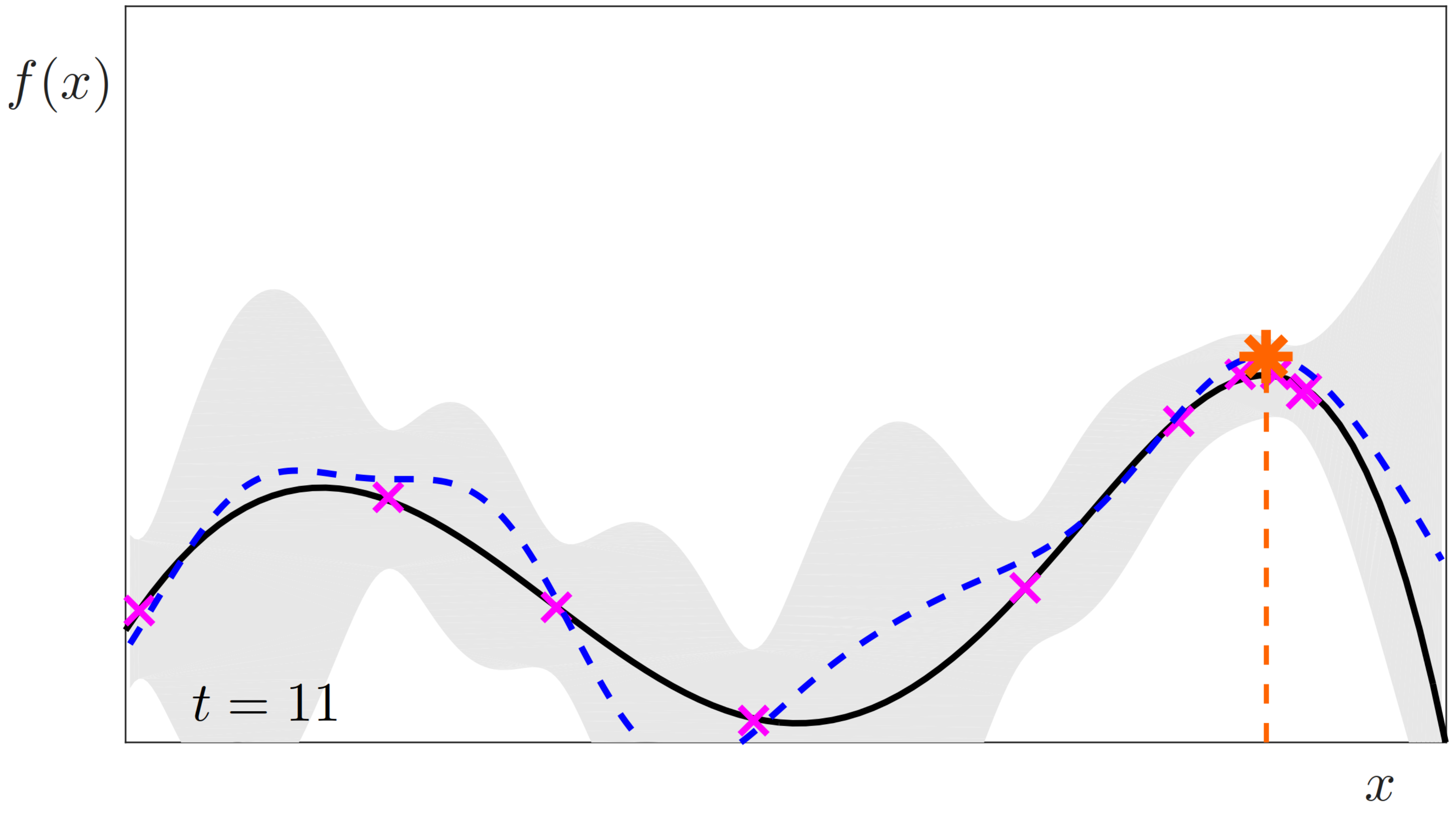
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



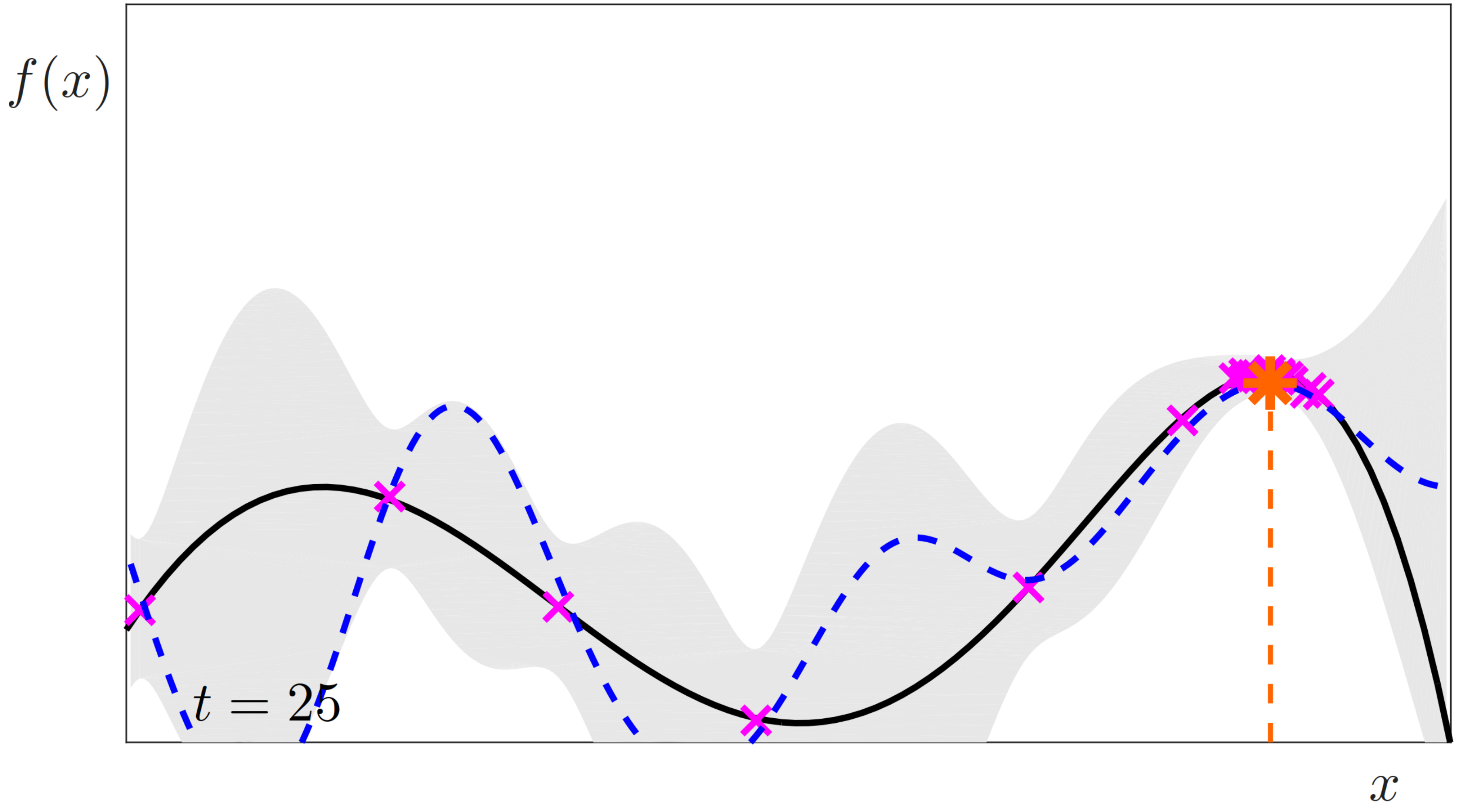
Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



Bayesian Optimization with Thompson Sampling

Goal: maximize function *f* using the results from previous experiments.



Bayesian Optimization

Other criteria for selecting $\boldsymbol{x_t}$:

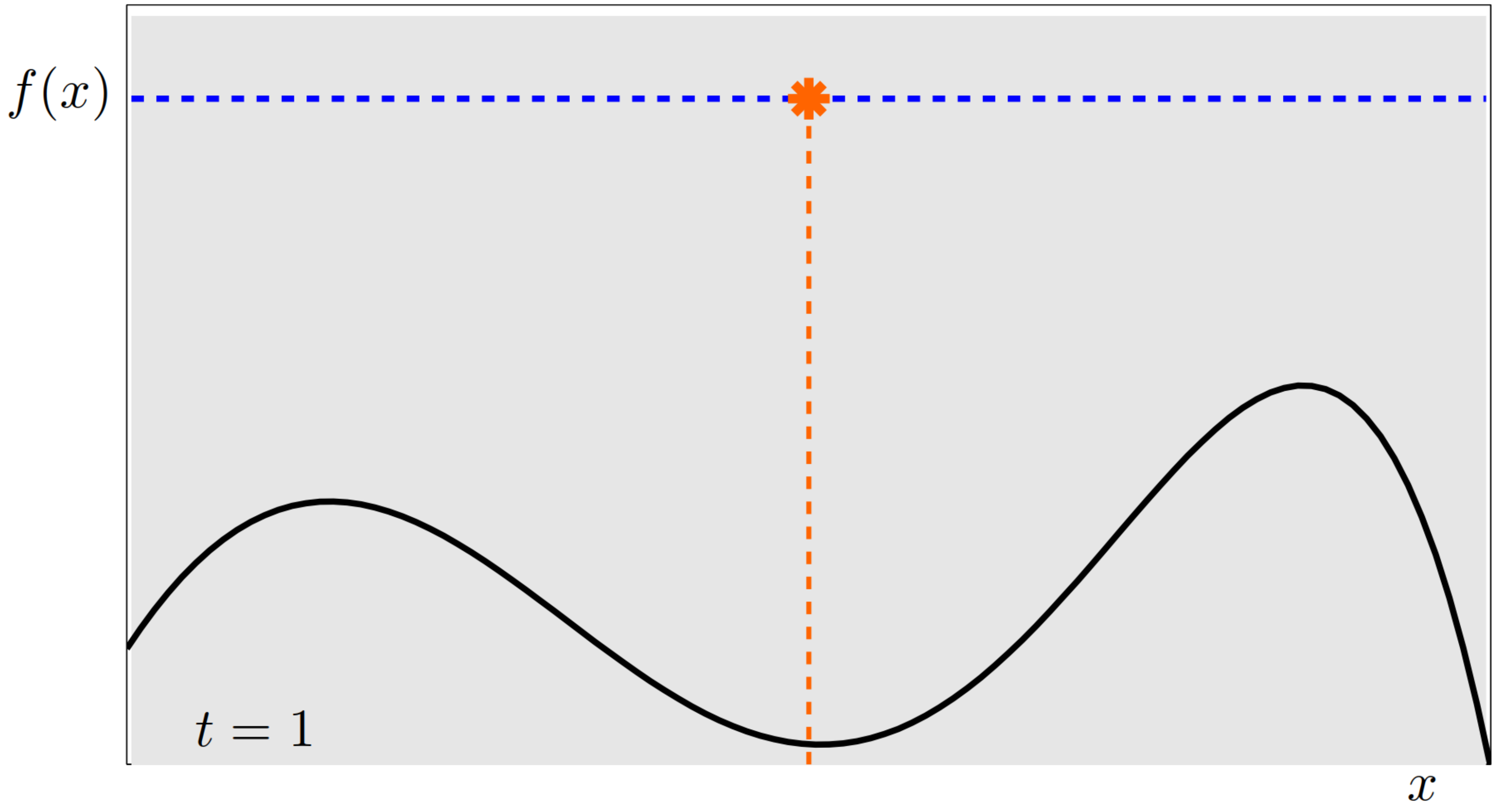
- ❑ Upper Confidence Bounds (Srinivas et al. 2010)
- ❑ Expected improvement (Jones et al. 1998)
- ❑ Probability of improvement (Kushner et al. 1964)
- ❑ Entropy search (Hernandez-Lobato et al. 2014, Wang et al. 2017)
- ❑ . . . and a few more.

Bayesian models for f :

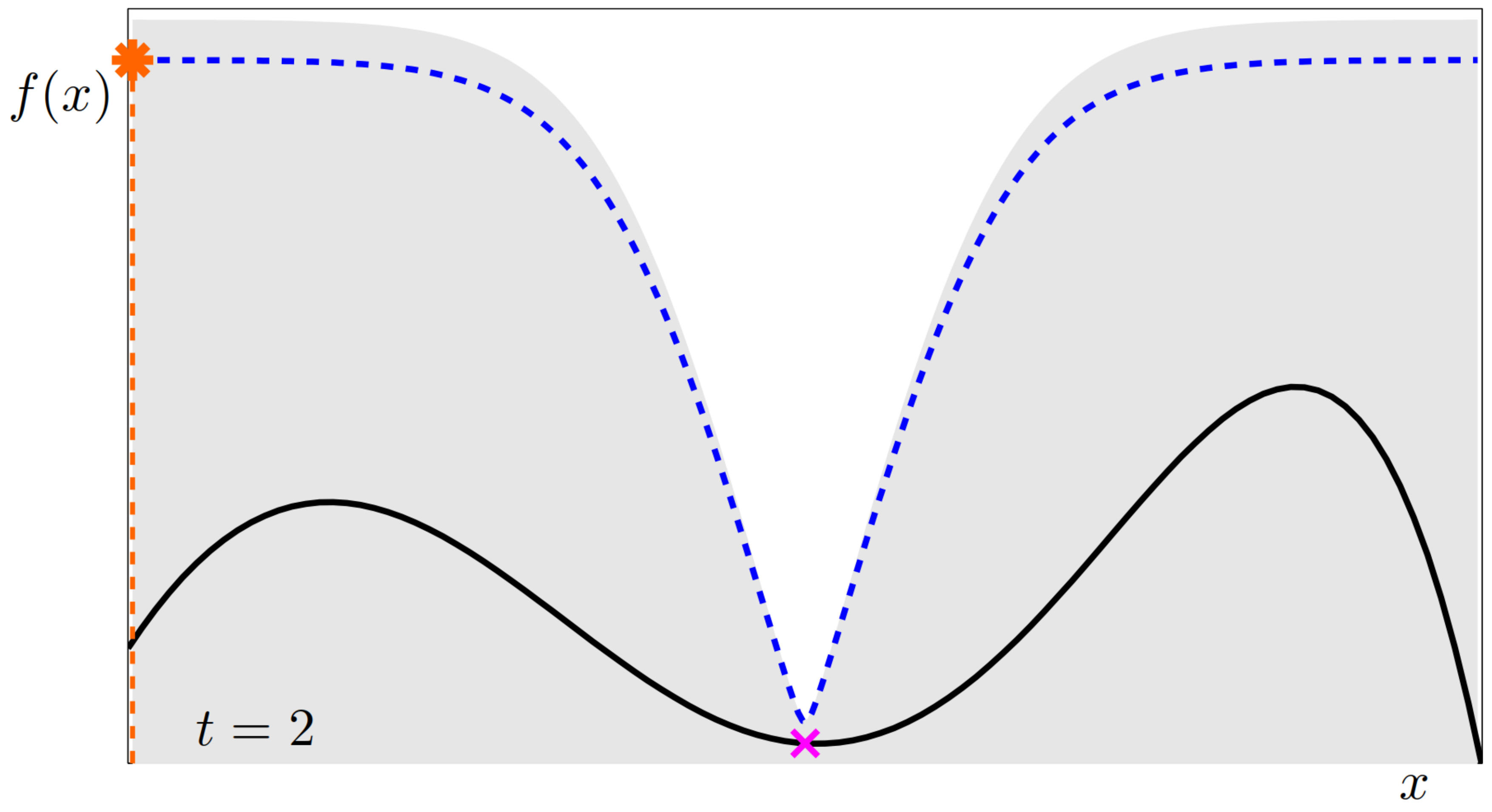
- ❑ Gaussian Processes (most popular)
- ❑ Neural networks (Snoek et al. 2015)
- ❑ Random forests (Hutter 2009)

Bayesian Optimization with Upper Confidence Bounds

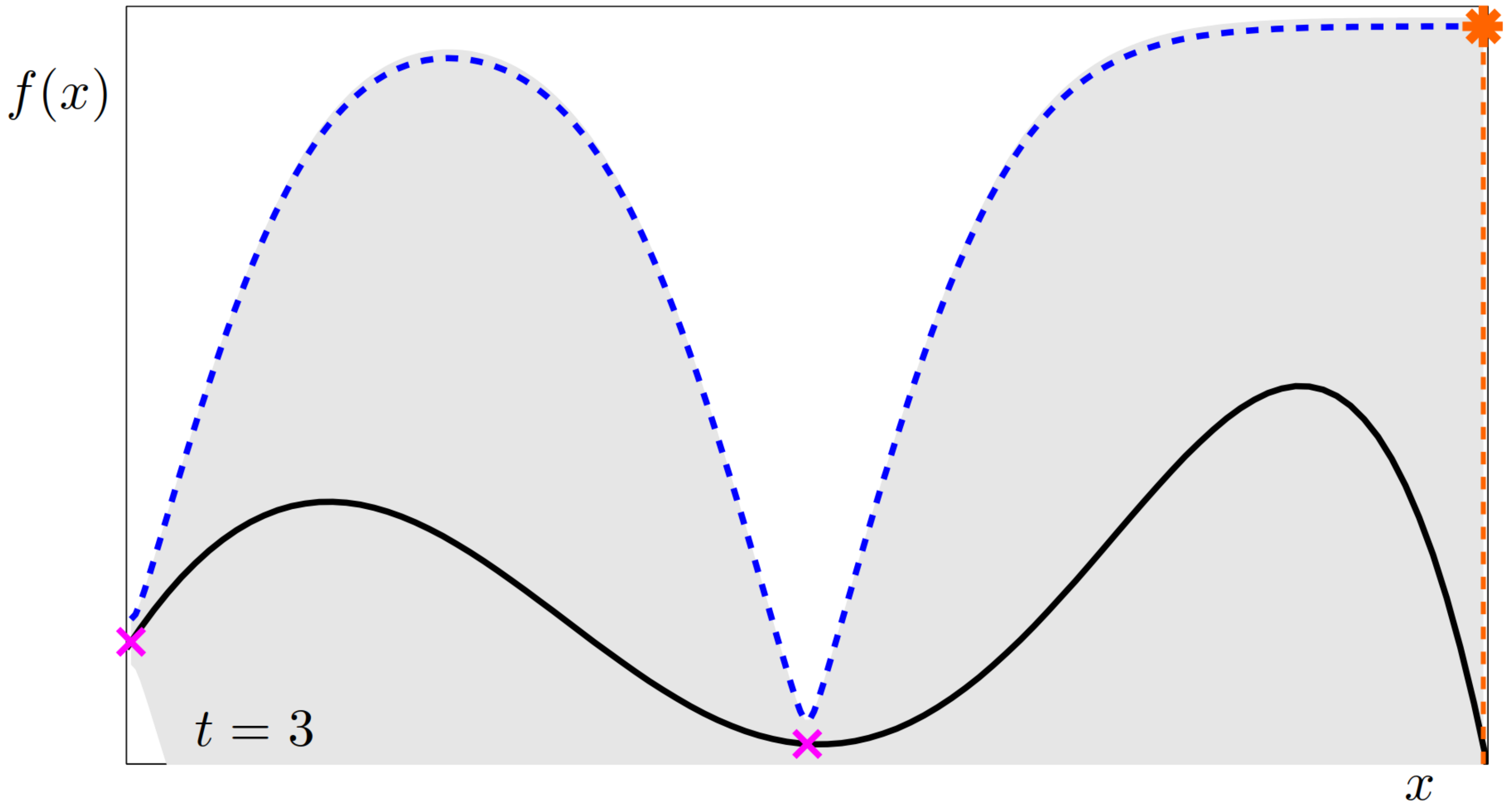
Goal: maximize function *f* using the results from previous experiments.



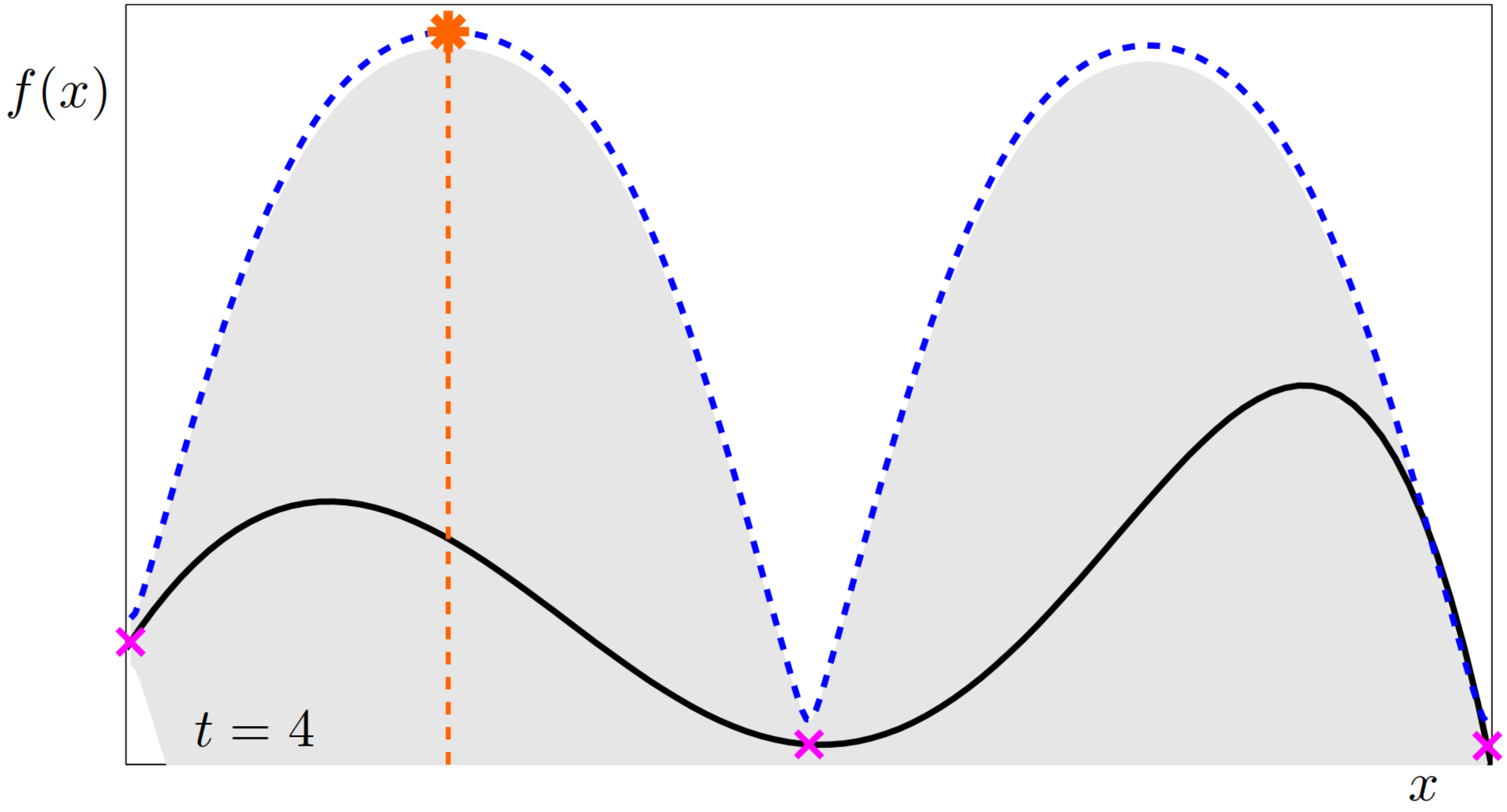
Gaussian Processes with Upper Confidence Bounds



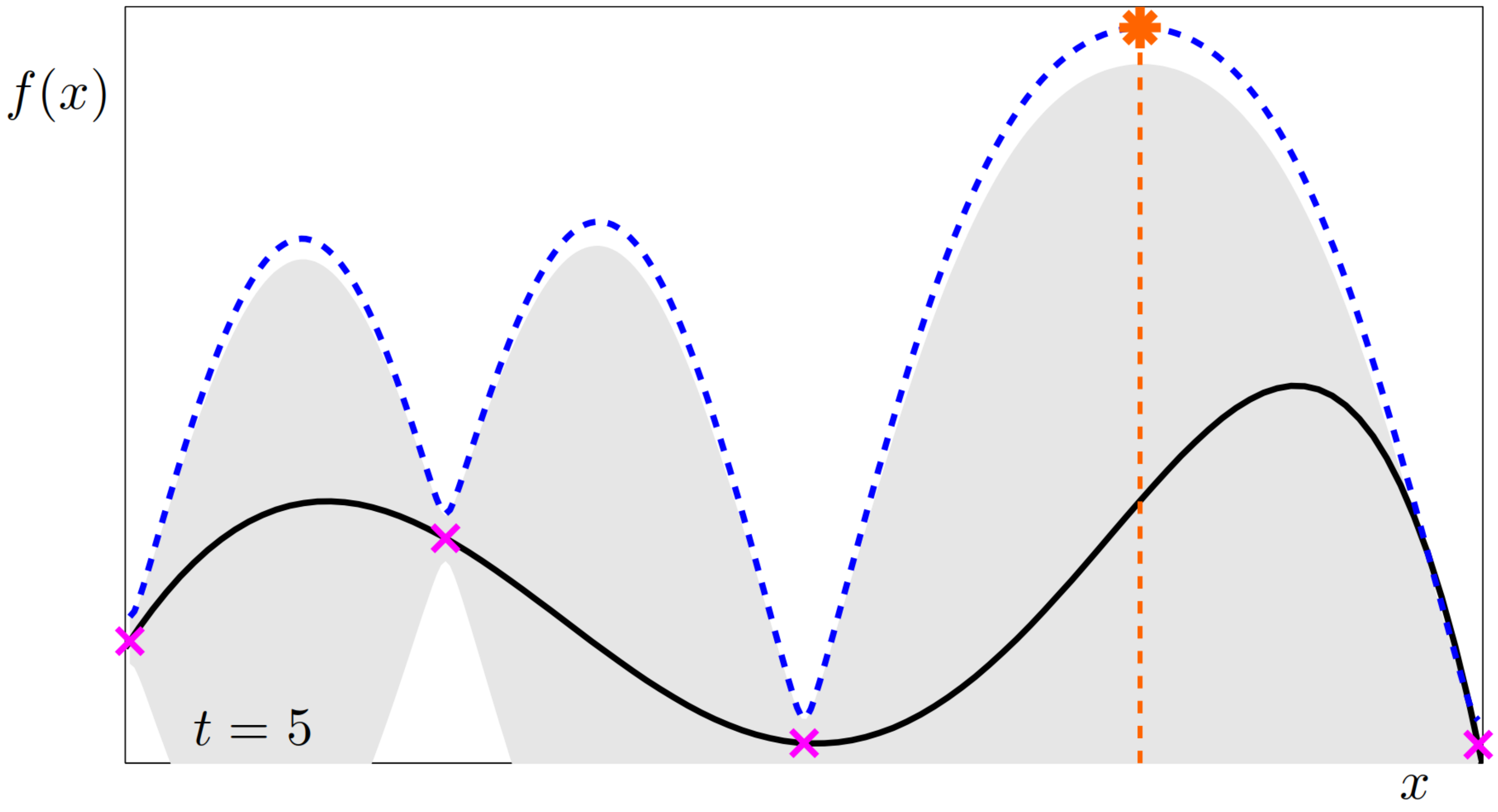
Gaussian Processes with Upper Confidence Bounds



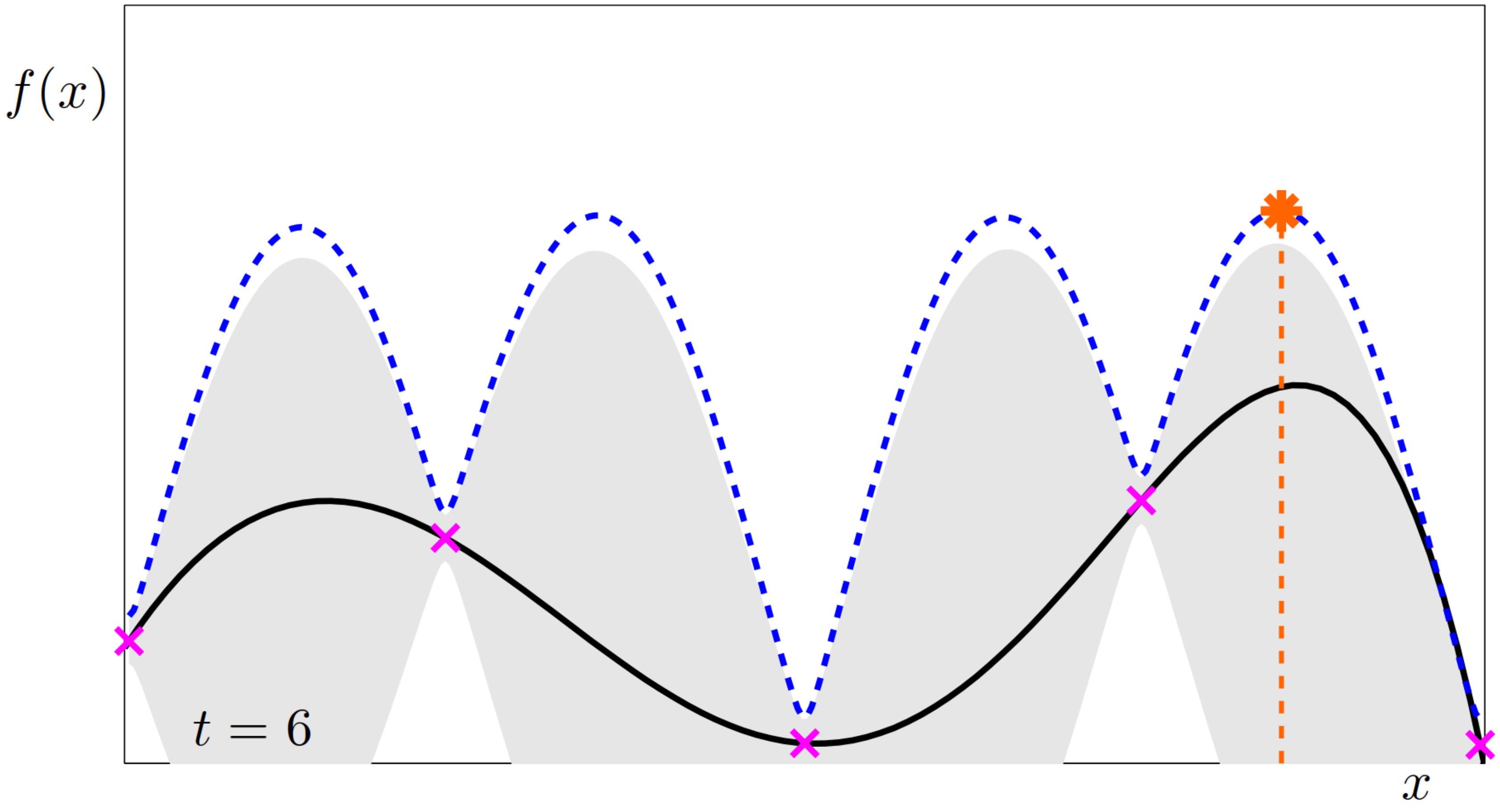
Gaussian Processes with Upper Confidence Bounds



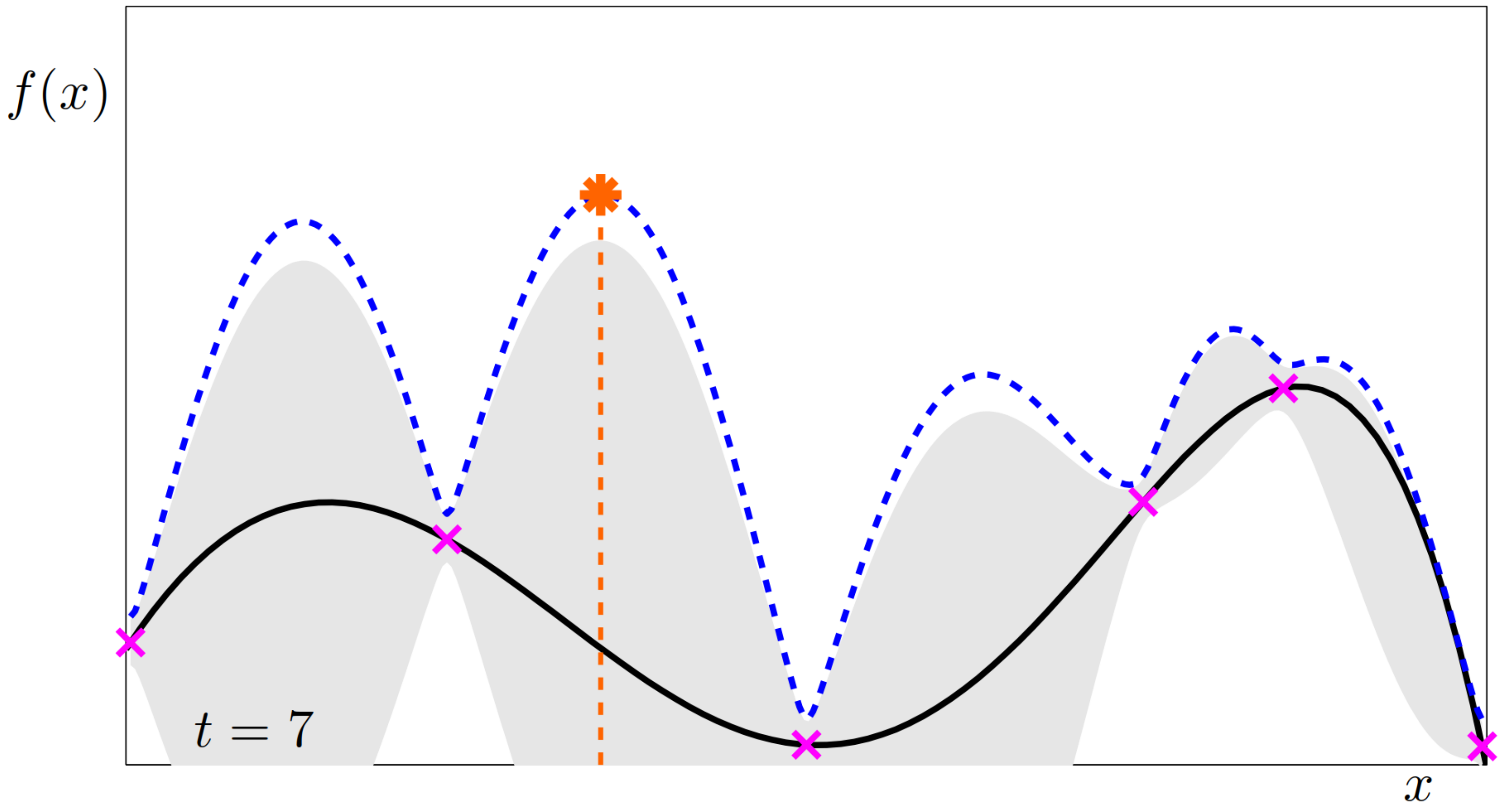
Gaussian Processes with Upper Confidence Bounds



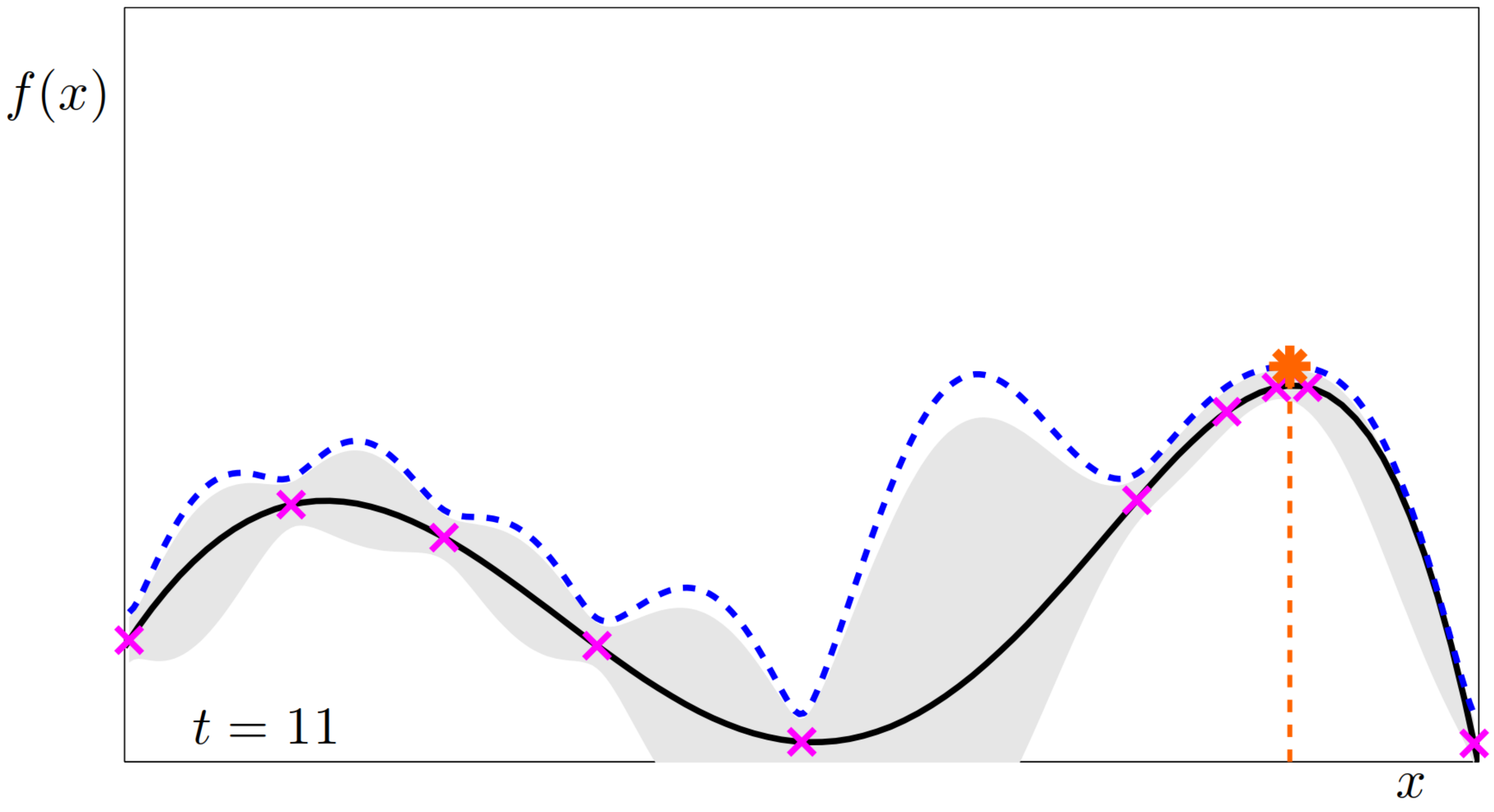
Gaussian Processes with Upper Confidence Bounds



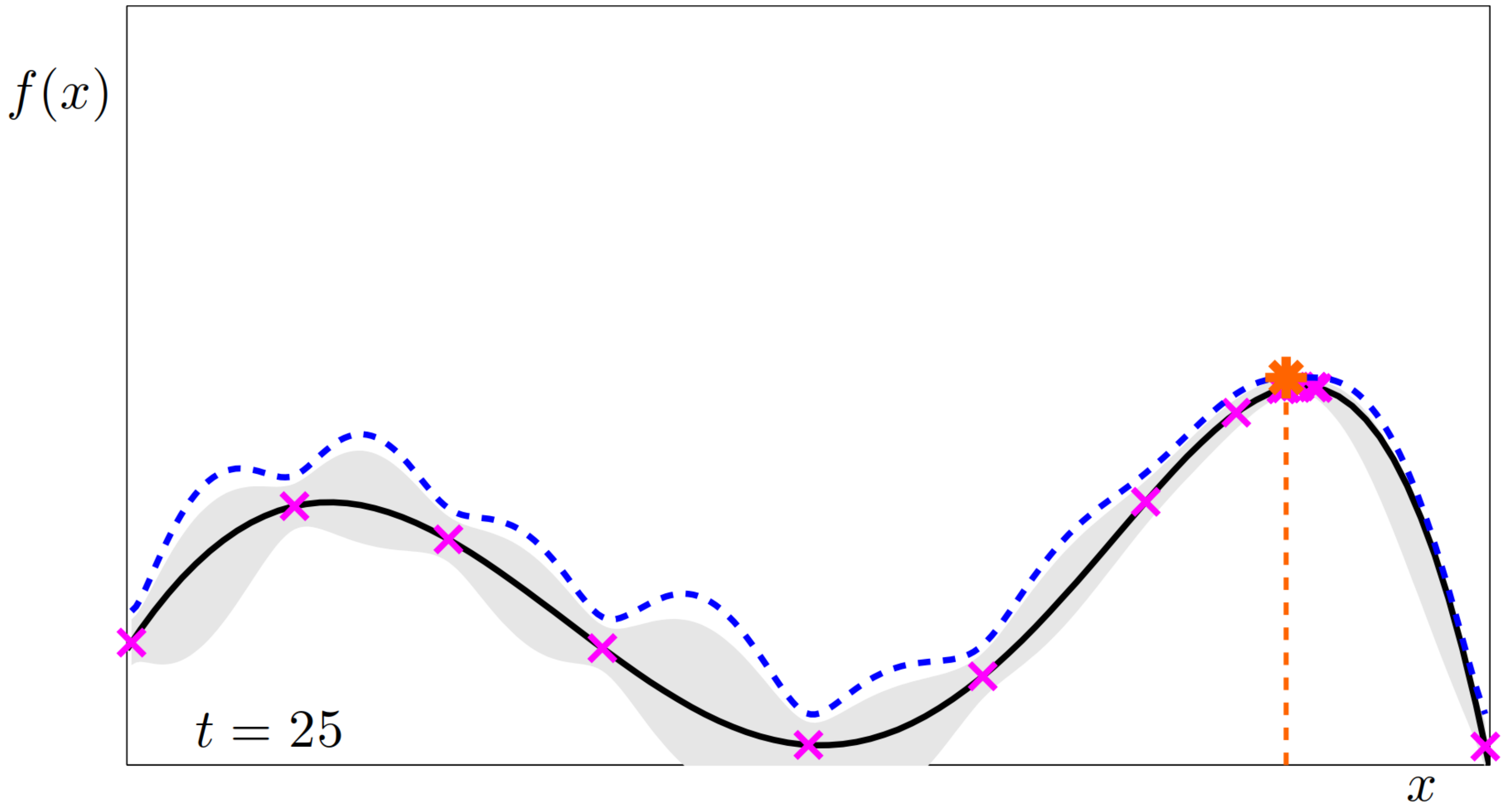
Gaussian Processes with Upper Confidence Bounds



Gaussian Processes with Upper Confidence Bounds



Gaussian Processes with Upper Confidence Bounds



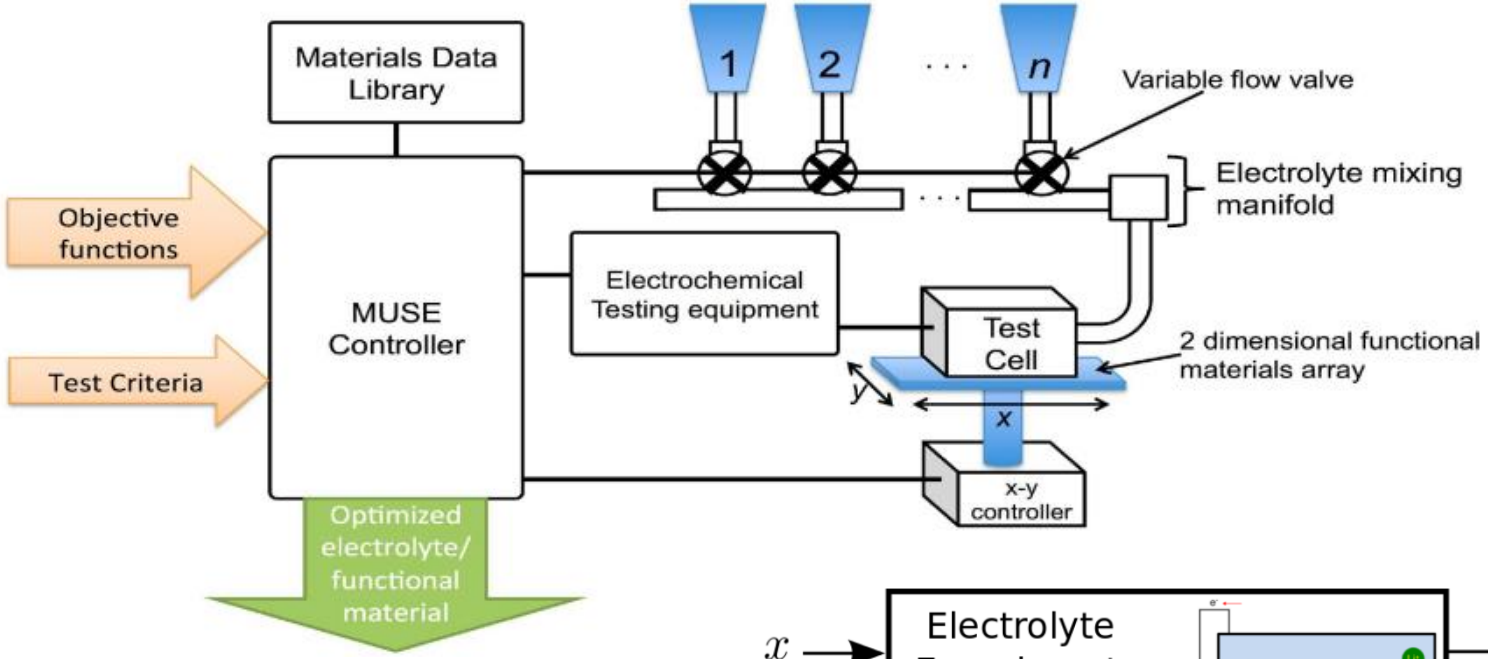
Take me home!

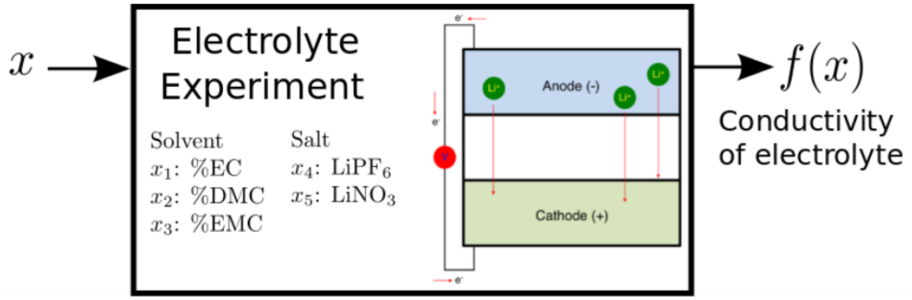
Bayesian Optimization can be used for function optimization

- When **function evaluation is expensive**
- **Gradient is not available**

Recommend experiments for creating better batteries

Machine Learning Unified Synchronous Experimentation (MUSE): **Rapid Autonomous Discovery/Optimization of Electrode and Electrolyte Materials (**joint project with Jay Whitacre & Venkat Viswanathan)


Materials Data Library
Objective functions
Test Criteria
MUSE Controller
Optimized electrolyte/ functional material
1
2
. . .
n
Variable flow valve
Electrolyte mixing manifold
Electrochemical Testing equipment
Test Cell
2 dimensional functional materials array
y
x
x-y controller



Goal: perform autonomous experimentation on electrolyte/electrochemically functional material combinations based on feedback from previous experiments.

1M KCl
50,000
0.1K
1642

Myopic Bayesian Design of Experiments via Posterior Sampling (MPS)

Setting

A decision maker chooses an action (experiment) $x \in \mathcal{X}$

then observes the outcome of the experiment $Y_x \sim P(\cdot|x, \theta^*)$.

where θ^* is an unknown parameter that we want to estimate from the outcome of the experiments.

After t rounds, the collected data: $D_t = \{(X_j, Y_{X_j})\}_{j=1}^{t}$

X_j = experiment parameter at iteration j, Y_{X_j}=result of the X_j experiment.

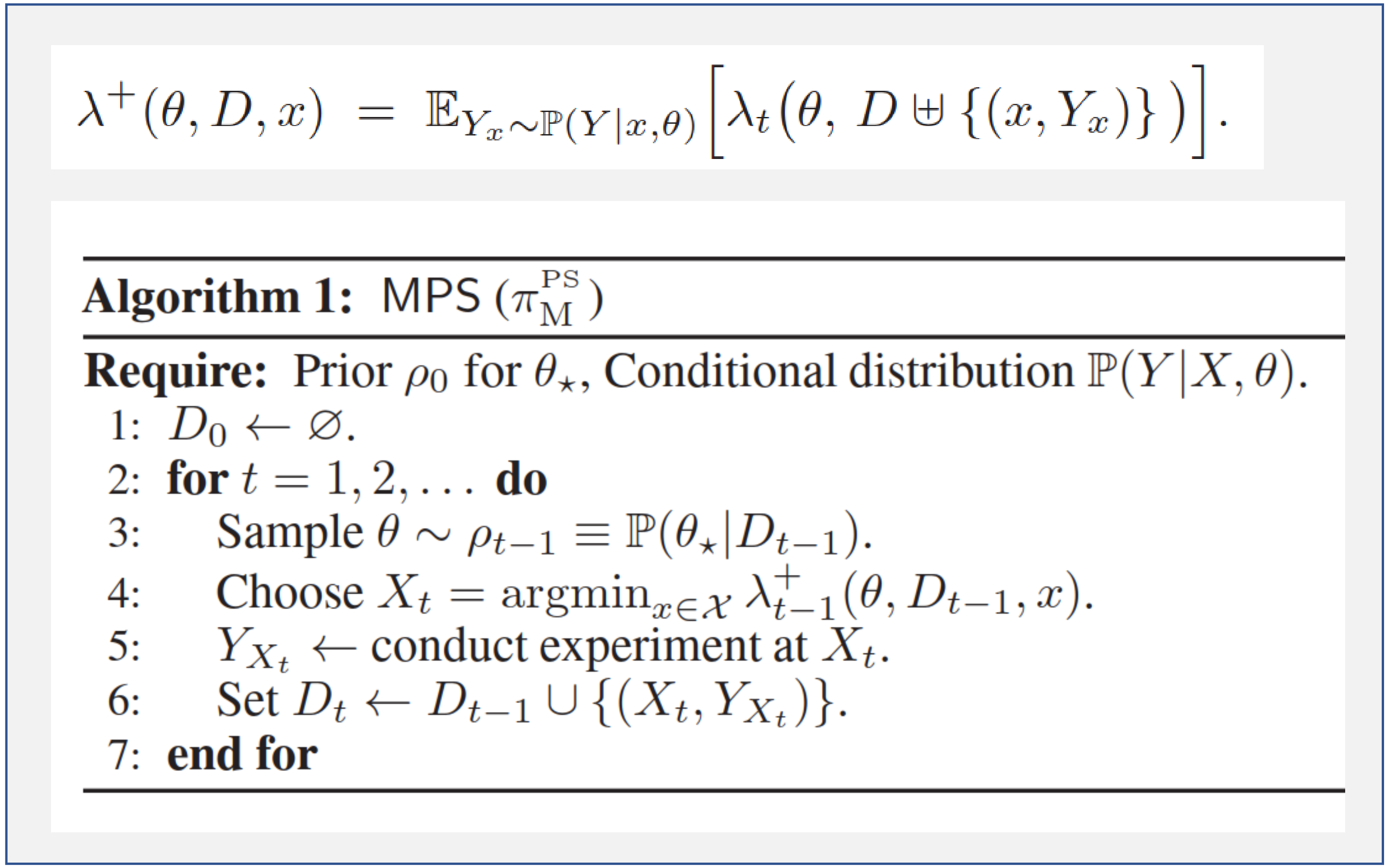
The goal: minimize a penalty function $\lambda(\theta^*, D)$ in θ

Example: $\theta = [f_{sol}(), f_{vis}(), \max_x f_{con}(x)]$ Solvation energy, viscosity, max conductivity

$$\lambda(\theta_\star, D_n) = \alpha \| f_{\text{sol}} - \hat{f}_{\text{sol}}(D_n) \|^2 + \beta \| f_{\text{vis}} - \hat{f}_{\text{vis}}(D_n) \|^2 + \gamma(\max f_{\text{con}} - \max_{X_t, t \le n} f_{\text{con}}(X_t)),$$

Myopic Bayesian Design of Experiments via Posterior Sampling (MPS)

After t rounds, the collected data: $D_t = \{(X_j, Y_{X_j})\}_{j=1}^{t}$ (Experiment parameters, results of experiments)



$$\lambda^+(\theta, D, x) \;=\; \mathbb{E}_{Y_x \sim \mathbb{P}(Y|x,\theta)}\Big[\lambda_t\big(\theta,\ D \uplus \{(x, Y_x)\}\big)\Big].$$

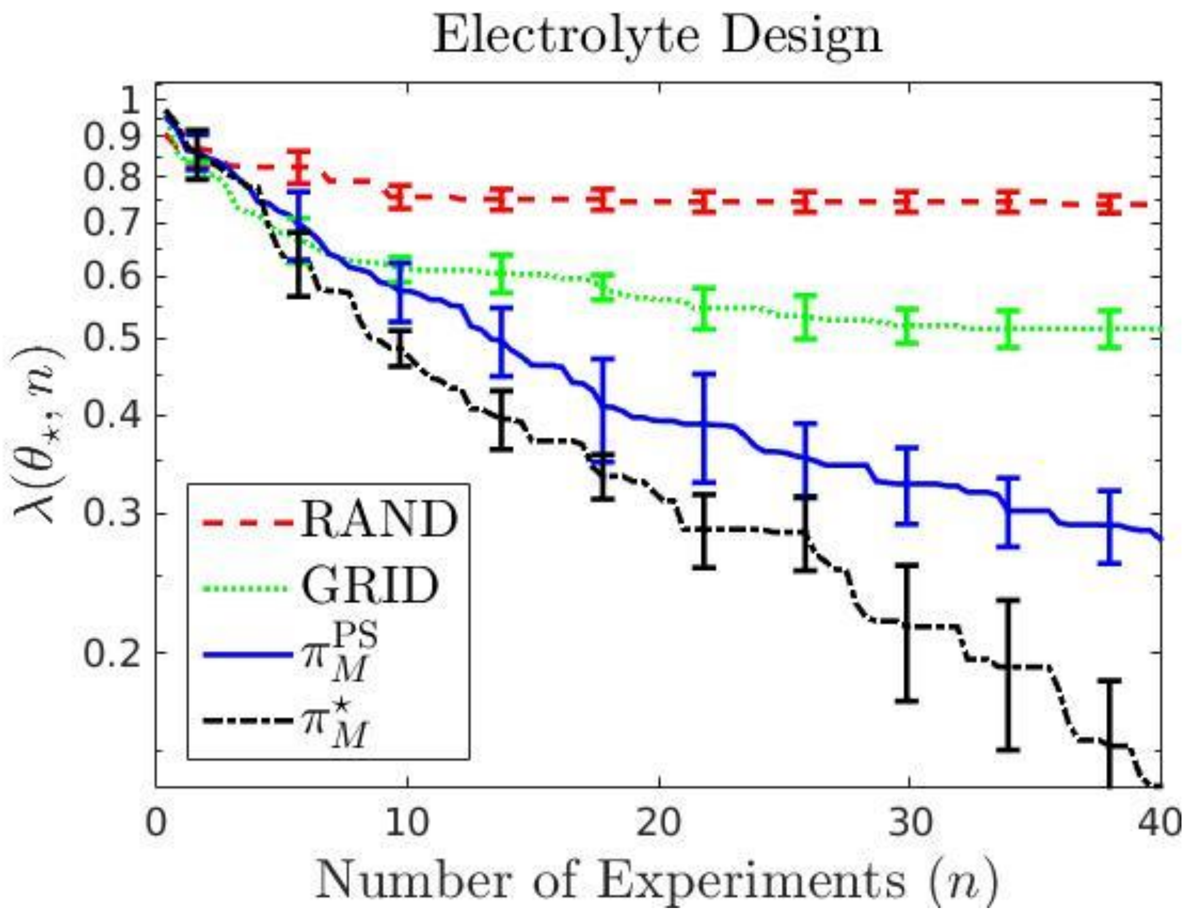
(predicted error after trying the experiment with parameters x)

```
Algorithm 1: MPS ($\pi_{\mathrm{M}}^{\mathrm{PS}}$)
Require: Prior $\rho_0$ for $\theta_\star$, Conditional distribution $\mathbb{P}(Y|X,\theta)$.
1: $D_0 \leftarrow \varnothing$.
2: for $t = 1, 2, \ldots$ do
3:     Sample $\theta \sim \rho_{t-1} \equiv \mathbb{P}(\theta_\star|D_{t-1})$.
4:     Choose $X_t = \mathrm{argmin}_{x\in\mathcal{X}}\, \lambda_{t-1}^{+}(\theta, D_{t-1}, x)$.
5:     $Y_{X_t} \leftarrow$ conduct experiment at $X_t$.
6:     Set $D_t \leftarrow D_{t-1} \cup \{(X_t, Y_{X_t})\}$.
7: end for
```

$$\lambda(\theta_\star, D_n) = \alpha\|f_{\mathrm{sol}} - \hat{f}_{\mathrm{sol}}(D_n)\|^2 + \beta\|f_{\mathrm{vis}} - \hat{f}_{\mathrm{vis}}(D_n)\|^2 + \gamma(\max f_{\mathrm{con}} - \max_{X_t, t\le n} f_{\mathrm{con}}(X_t)),$$

Myopic Bayesian Design of Experiments via Posterior Sampling (MPS)

Action space X: proportion of two solvents EC and EMC, molarity of the salt LiPF6, and T temperature

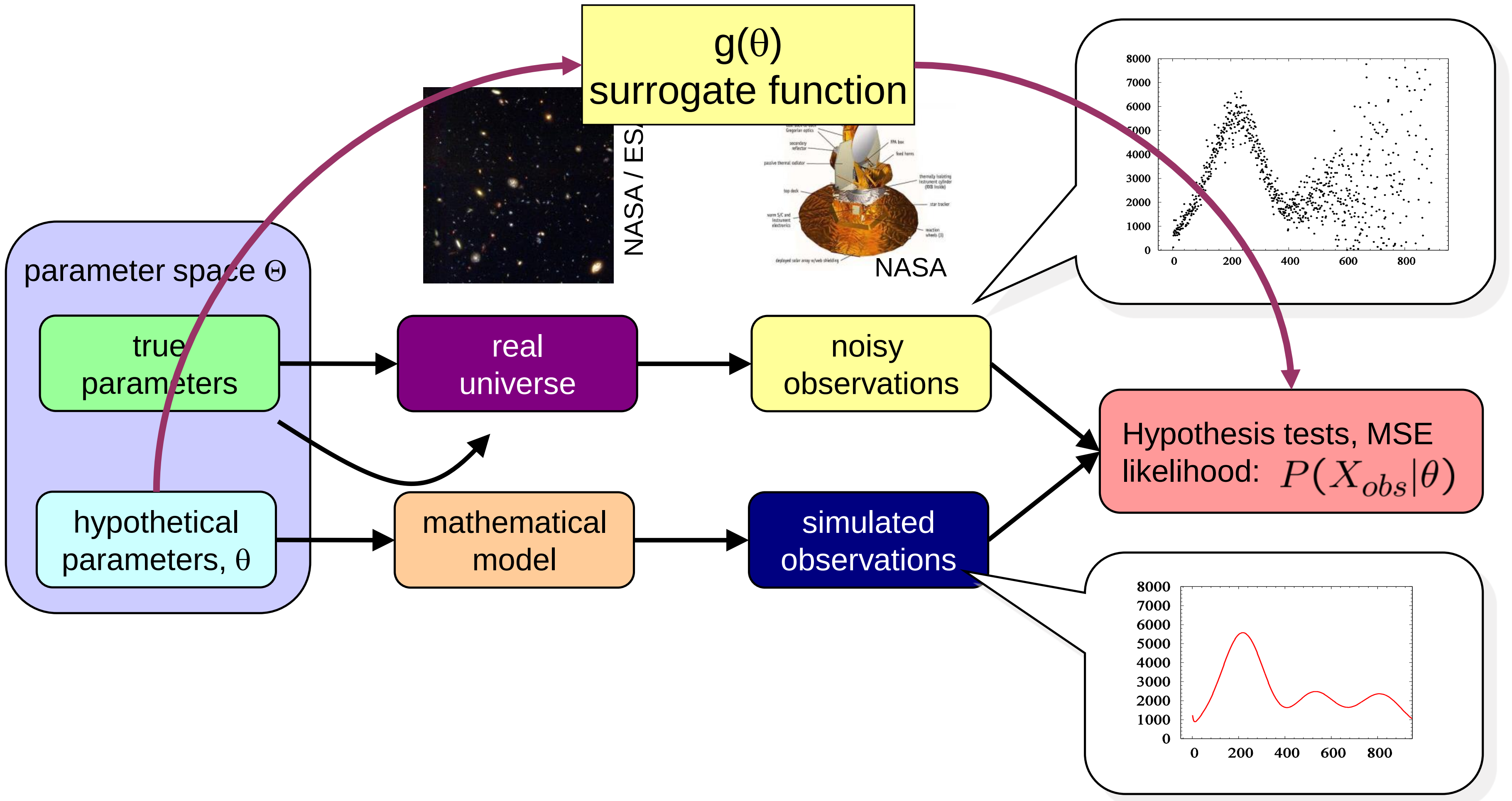


$$\lambda(\theta_\star, D_n) = \alpha \| f_{\mathrm{sol}} - \hat{f}_{\mathrm{sol}}(D_n) \|^2 + \beta \| f_{\mathrm{vis}} - \hat{f}_{\mathrm{vis}}(D_n) \|^2 + \gamma (\max f_{\mathrm{con}} - \max_{X_t, t \leq n} f_{\mathrm{con}}(X_t)),$$

Manufacturing: Recommend parameter settings for machines

Goal: Recommend settings for wire cutting machines

Find the True Parameters of the Universe



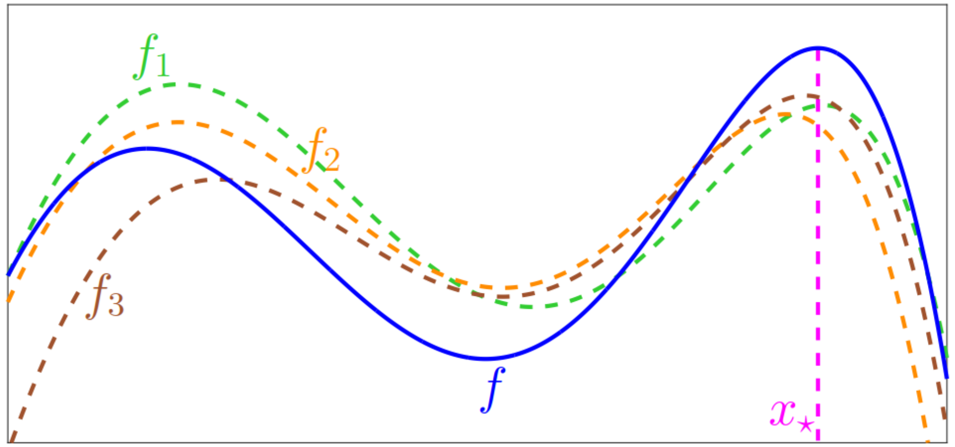
Question: How to search the parameter space? $\arg\max_{\theta \in \Theta} P(X_{obs}|\theta) = ?$

Solution: Learn a surrogate function and make experiment decisions using it

Multi-fidelity optimization

Multi-fidelity optimization

Desired function f is very expensive, but . . .
we have access to cheap approximations.



$f_1, f_2, f_3 \approx f$ which are cheaper to evaluate.

Example:

f: expensive target function
f_1:cheap computer simulation
f_2:cheap lab experiment
f_3:expensive lab experiment

Multi-fidelity application: Astrophysical Maximum Likelihood Inference

Data: We use **Type Ia supernova data** for maximum likelihood inference on 3 cosmological parameters:

- the Hubble constant $H_0 \in (60, 80)$,
- the dark matter fraction $\Omega_M \in (0, 1)$
- and dark energy fraction $\Omega_\Lambda \in (0, 1)$,

hence $d = 3$.

Likelihood function: Robertson-Walker metric (Requires numerical integration).

Multi-fidelity:

We construct a $p = 2$ dimensional multi-fidelity problem where we can choose between

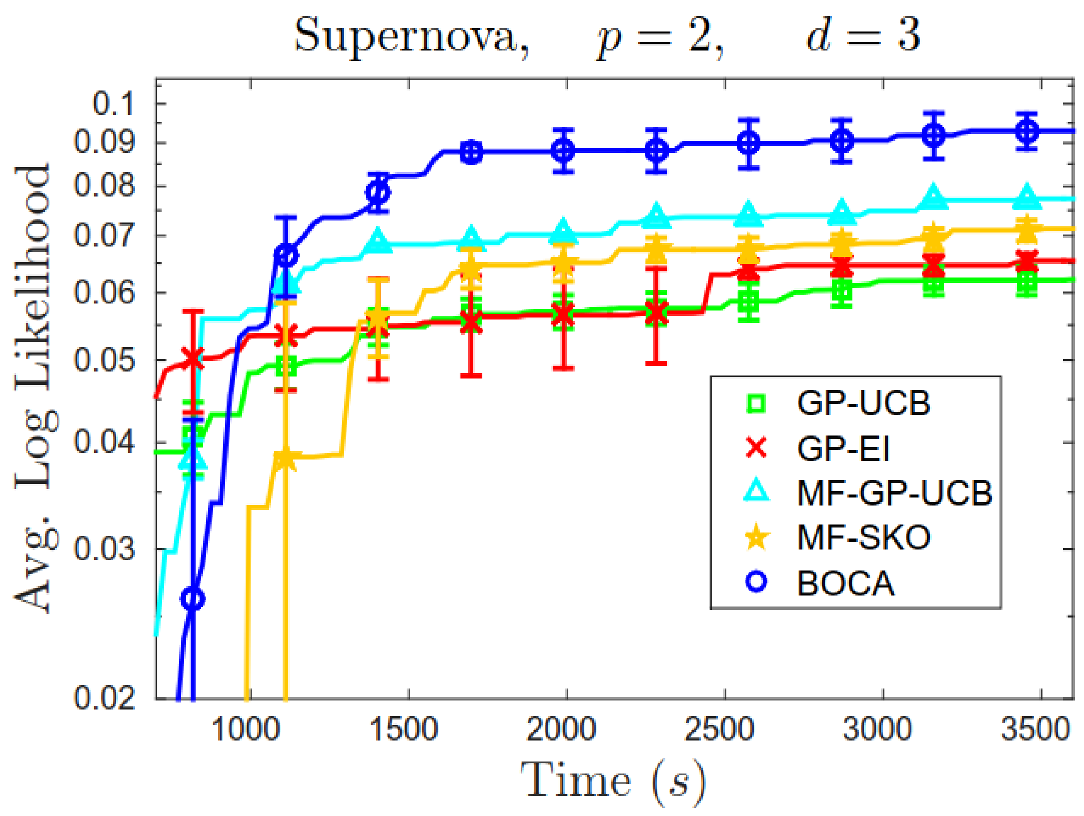
- **data set size $N \in [50, 192]$**
- and perform the **integration on grids of size $G \in$ [10^2, 10^6]** via the trapezoidal rule.

As the cost function for fidelity selection, we used **$\lambda(N, G) = NG$** as the computation time is linear in both parameters

Multi-fidelity application: Astrophysical Maximum Likelihood Inference

We plot the maximum average log likelihood against wall clock time as that is the cost in this experiment.

The plot includes the time taken by each method to tune the GPs and determine the next points/fidelities for evaluation.



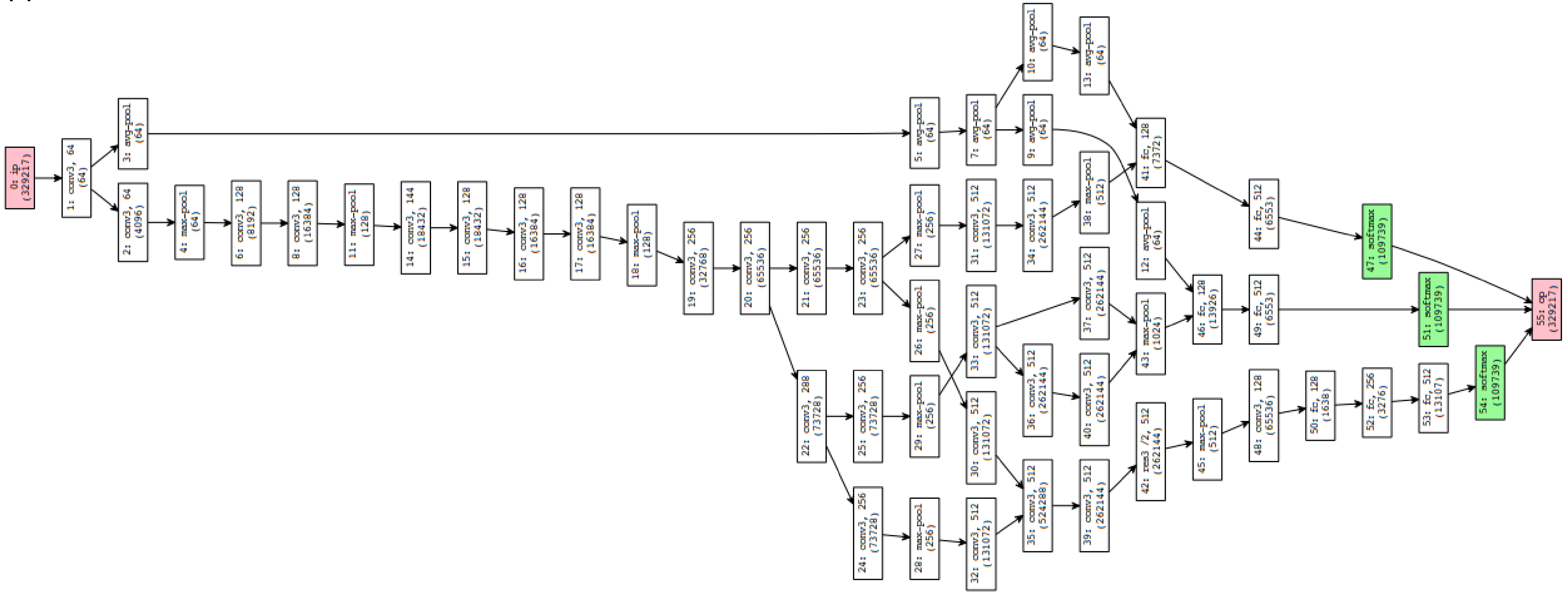
Bayesian Optimization with Continuous Approximations (BOCA)

We compare **BOCA** to the following four baselines:

I. GP-UCB,
II. the GP-EI criterion in BO (Jones et al., 1998),
III. MF-GP-UCB (Kandasamy et al., 2016a)
IV. and MF-SKO, the multi-fidelity sequential kriging optimisation method from Huang et al. (2006).

Neural Architecture Search with Bayesian Optimization and Optimal Transport

Multi-fidelity learning: Training a neural network on large dataset is very expensive, but sometimes cheap approximations are available.



Optimized neural network architecture on CIFAR 10 dataset

Neural Architecture Search

Method	Blog $(60K, 281)$	Indoor $(21K, 529)$	Slice $(54K, 385)$	Naval $(12K, 17)$	Protein $(46K, 9)$	News $(40K, 61)$	Cifar10 $(60K, 1K)$	Cifar10 $150K$ iters
RAND	0.780 $\pm$ 0.034	**0.115 $\pm$0.023**	0.758 $\pm$ 0.041	0.0103 $\pm$ 0.002	0.948 $\pm$ 0.024	**0.762 $\pm$0.013**	0.1342 $\pm$ 0.002	0.0914 $\pm$ 0.008
EA	0.806 $\pm$ 0.040	0.147 $\pm$ 0.010	0.733 $\pm$ 0.041	**0.0079 $\pm$0.004**	1.010 $\pm$ 0.038	**0.758 $\pm$0.038**	0.1411 $\pm$ 0.002	0.0915 $\pm$ 0.010
TreeBO	0.928 $\pm$ 0.053	0.168 $\pm$ 0.023	0.759 $\pm$ 0.079	0.0102 $\pm$ 0.002	0.998 $\pm$ 0.007	0.866 $\pm$ 0.085	0.1533 $\pm$ 0.004	0.1121 $\pm$ 0.004
NASBOT	**0.731 $\pm$0.029**	**0.117 $\pm$0.008**	**0.615 $\pm$0.044**	**0.0075 $\pm$0.002**	**0.902 $\pm$0.033**	**0.752 $\pm$0.024**	**0.1209 $\pm$0.003**	**0.0869 $\pm$0.004**

RAND: random search; EA (Evolutionary algorithm): TreeBO: a BO method which only searches over feed forward structures.

The first row gives the number of samples N and the dimensionality D of each dataset in the form (N, D).

The subsequent rows show the regression **MSE or classification error** (lower is better) on the test set for each method.

The last column is for Cifar10 where we took the best models found by each method in 24K iterations and trained it for 120K iterations. When we trained the VGG-19 architecture using our training procedure, we got test errors 0.1718 (60K iterations) and 0.1018 (150K iterations)

Take me home!

Methods for Multi-fidelity optimization

- Applications in Cosmology, Neural Network Search
- Code available

Multi-objective optimization

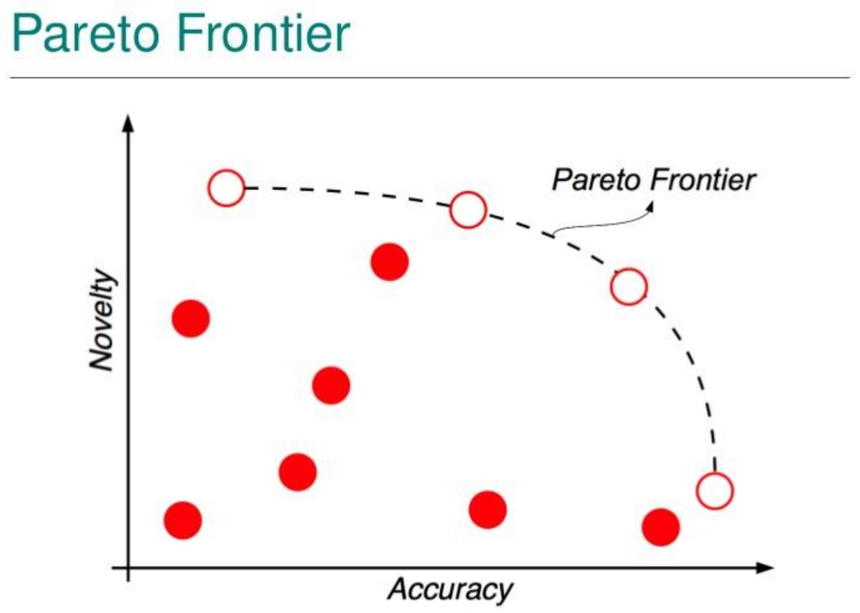
Many real world applications can be framed as multi-objective optimization problems, where we wish to simultaneously optimize for multiple criteria (**multi-objective optimization**).

- ❑ **Classification**: max True positive rate, max True negative rate, min False positive rate, min False negative rate, min computation cost, fast decision, min energy, min memory usage, min hard drive storage…

- ❑ **Drug discovery**: each evaluation of the functions is an in-vitro experiment
    - measure the solubility, toxicity and potency of a candidate example.
    - Goal: max solubility, max potency, min toxicity, min experiment cost.

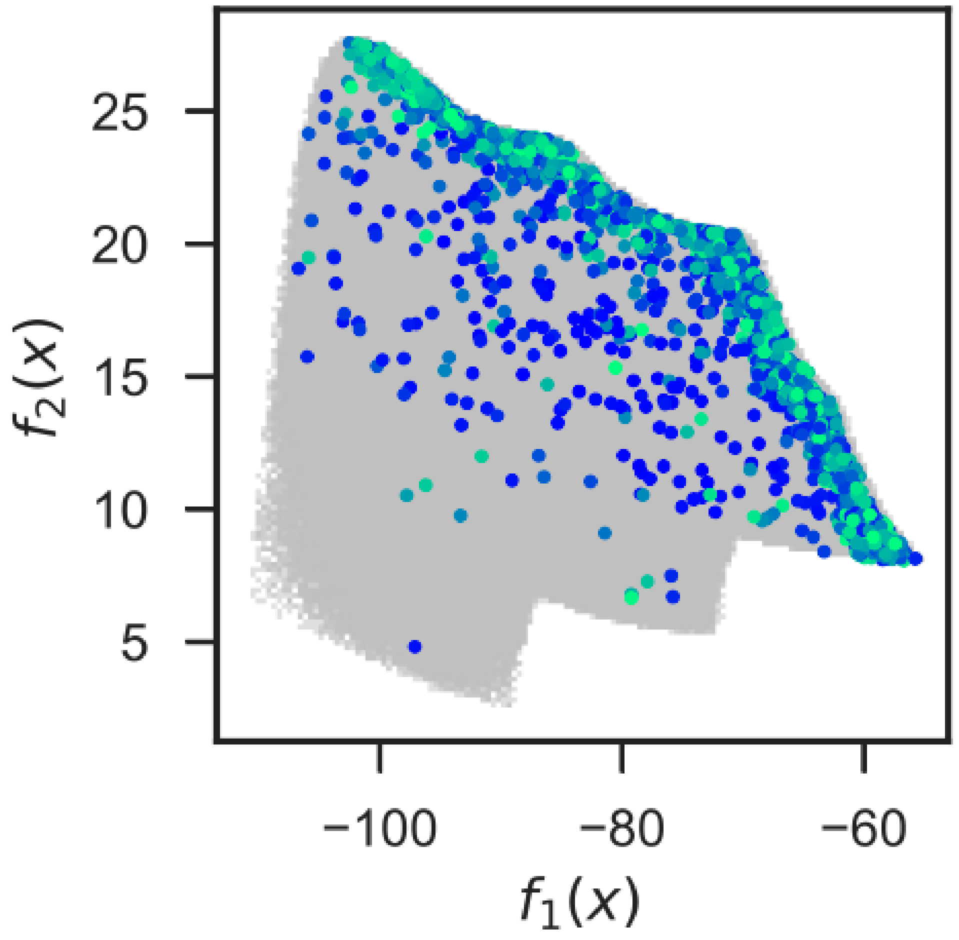
Multi-objective optimization

- **Goal**: recover (some part of the) Pareto front of these objectives with minimum number of experiments.



Goal: Maximize Objective 1 and Objective 2

Multi-objective optimization. Results:



Brighter colors were sampled in the later iterations

Take me home!

Bayesian Optimization based algorithm for multi-objective optimization

- Code is available!

Experiment Recommendation for Efficient Posterior Estimation

Query Efficient Posterior Estimation in Scientific Experiments via Bayesian Active Learning
Kirthevasan Kandasamy, Jeff Schneider, and Barnabas Poczos
Artificial Intelligence Journal, 2017

Posterior Estimation

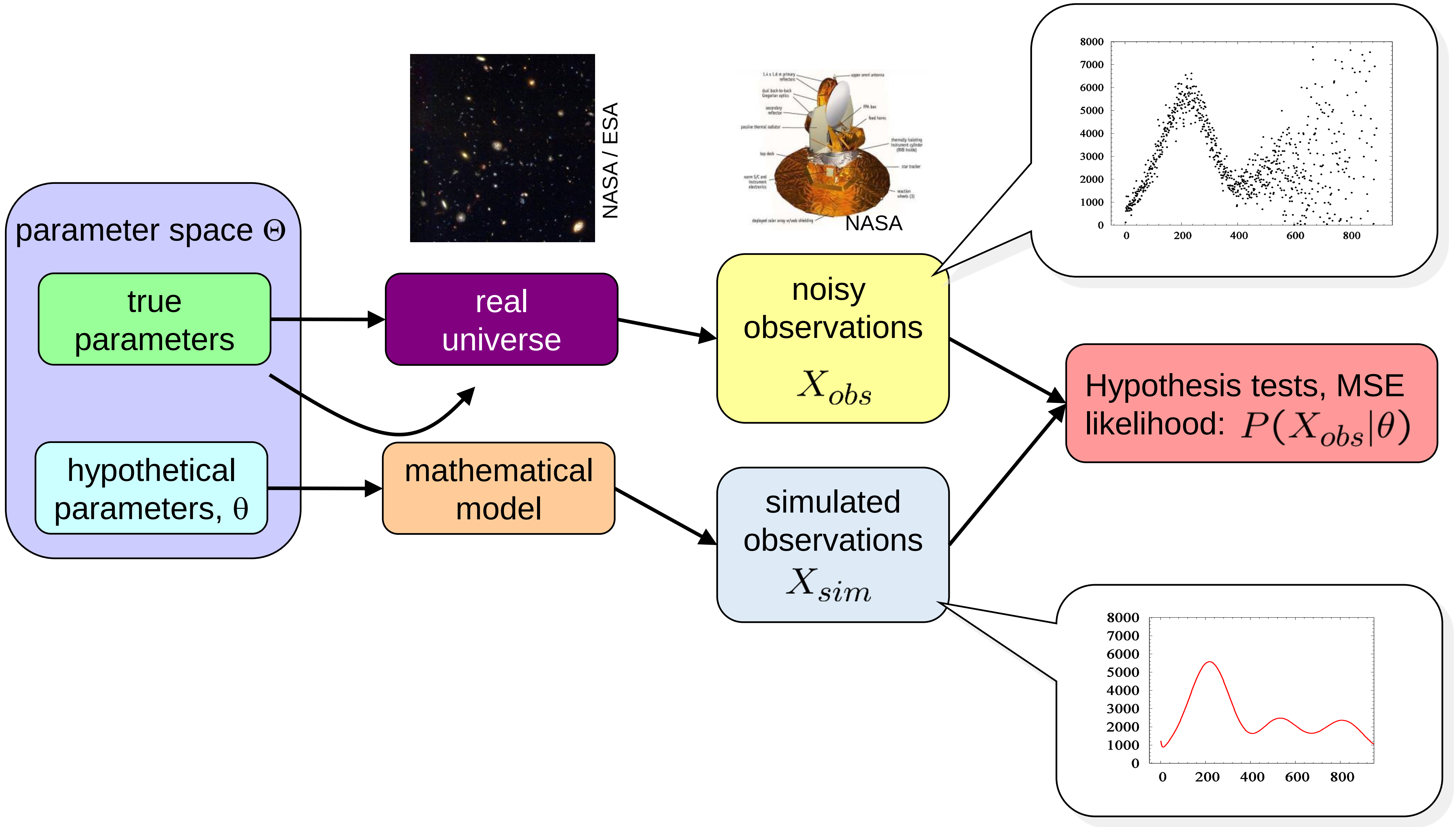
- Several cosmological phenomena are characterized by the **cosmological parameters** (e.g. Hubble constant, dark energy fraction).

- Given observations, we wish to **make inferences about the parameters**.

- Physicists have developed **simulation-based** probability models (= black box) of the Universe which can be used to **compute the likelihood function of cosmological parameters** for a given observation.

- Our goal is an efficient way to **estimate posterior densities as functions** when calls to this black box / **simulators are expensive**.

Likelihood of the Parameters of the Universe


parameter space Θ
true parameters
hypothetical parameters, θ
real universe
mathematical model
NASA / ESA
NASA
noisy observations X_{obs}
simulated observations X_{sim}
Hypothesis tests, MSE likelihood: $P(X_{obs}|\theta)$
8000
7000
6000
5000
4000
3000
2000
1000
0
0
200
400
600
800

Active Posterior Estimation

We have a **parameter space** Θ for the unknown parameters θ (cosmological constants

Let X_{obs} denote our observations (e.g. signals from telescopes).

For each $\theta \in \Theta$ we have the ability to **query an oracle** for the value of the likelihood $\mathcal{L}(\theta) = P(X_{obs}|\theta)$, but these **queries are expensive**.

Posterior distribution: $$P(\theta|X_{obs}) = \frac{P(X_{obs}|\theta)P(\theta)}{P(X_{obs})} \approx \mathcal{L}(\theta)P(\theta)$$

Goal: Obtain an estimate $\widehat{P}(\theta|X_{obs})$ as a function of θ while minimizing queries to the oracle.

Active Posterior Estimation

Goal: Obtain an estimate $\widehat{P}(\theta|X_{obs})$ while minimizing queries to the oracle.

$$P(\theta|X_{obs}) = \frac{P(X_{obs}|\theta)P(\theta)}{P(X_{obs})} \approx \mathcal{L}(\theta)P(\theta)$$

Training data:

$A_t = \{\theta_i, \mathcal{L}(\theta_i)\}_{i=1}^{t}$ all data collected up to iteration t.

Ideal Greedy Parameter Selection:

$$\theta_t = \arg\min_{\theta_+ \in \Theta} D\Big(P_{\theta|X_{obs}}(\theta) || \widehat{P}^{A_{t-1} \cup (\theta_+,\, \mathcal{L}(\theta_+))}(\theta)\Big)$$

Unknown!

Utility Function Based Active Posterior Estimation

$\star$ An iterative greedy algorithm that picks the next point based on the points we collected so far.

$\star$ At time step t, we have observations at $t-1$ points:

$$B_{t-1} = \{\theta_i, \ \log\Big(\mathcal{L}(\theta_i)P(\theta_i)\Big)\}_{i=1}^{t-1} \text{ all data collected up to iteration } t-1.$$

$\star$ Fit a GP using data B_{t-1}

$\star$ Design a utility function $u_t : \Theta \to \mathbb{R}$ using the GP.

$u_t(\theta)$ captures value/utility of querying at θ.

$\star$ Choose $\theta_t = \arg\max_{\theta \in \Theta} u_t(\theta)$.

$\star$ Repeat.

Negative Expected Divergence

Training data: $A_t = \{\theta_i,\ \mathcal{L}(\theta_i\}_{i=1}^{t}$

Ideal greedy experiment recommendation:

$$\theta_t = \arg\min_{\theta_+ \in \Theta} D\Big(P_{\theta|X_{obs}}(\theta) || \widehat{P}^{A_{t-1} \cup (\theta_+,\ \mathcal{L}(\theta_+))}(\theta)\Big)$$

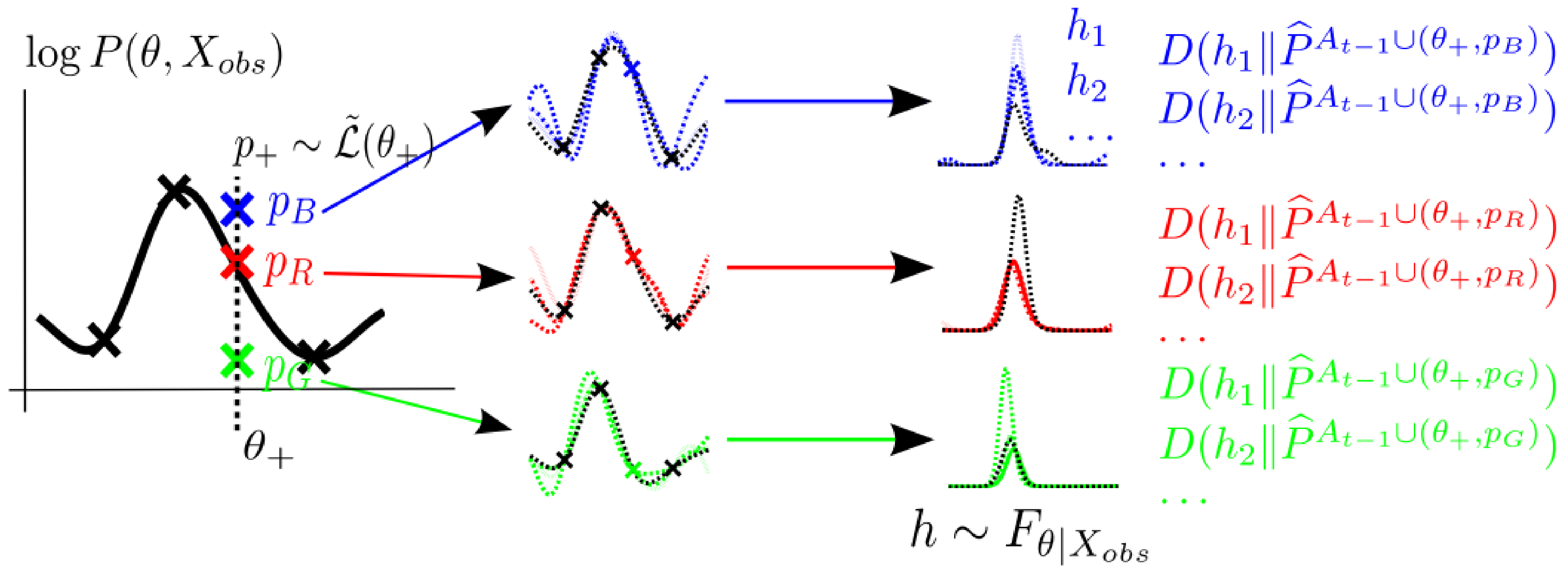
Negative Expected Divergence (NED)

- Ideally we should choose the point that results in the highest reduction in divergence **if we knew the likelihood and the true posterior** at that point.
- In NED, we choose the point with the **highest expected reduction** in the divergence.

$$u_t^{NED}(\theta_+) = -\mathbb{E}_{p+}\mathbb{E}_{h(\cdot)}\Big[D\Big(h(\theta) || \widehat{P}^{A_{t-1} \cup (\theta_+,\ p_+)}(\theta)\Big)\Big]$$

$$\theta_t = \arg\max_{\theta_+ \in \Theta} u_t^{NED}(\theta_+)$$

Negative Expected Divergence



$\log P(\theta, X_{obs})$
$p_+ \sim \tilde{\mathcal{L}}(\theta_+)$
p_B
p_R
p_G
θ_+
h_1
h_2
$\dots$
$D(h_1 \| \widehat{P}^{A_{t-1} \cup (\theta_+, p_B)})$
$D(h_2 \| \widehat{P}^{A_{t-1} \cup (\theta_+, p_B)})$
$\dots$
$D(h_1 \| \widehat{P}^{A_{t-1} \cup (\theta_+, p_R)})$
$D(h_2 \| \widehat{P}^{A_{t-1} \cup (\theta_+, p_R)})$
$\dots$
$D(h_1 \| \widehat{P}^{A_{t-1} \cup (\theta_+, p_G)})$
$D(h_2 \| \widehat{P}^{A_{t-1} \cup (\theta_+, p_G)})$
$\dots$
$h \sim F_{\theta | X_{obs}}$

Training data:

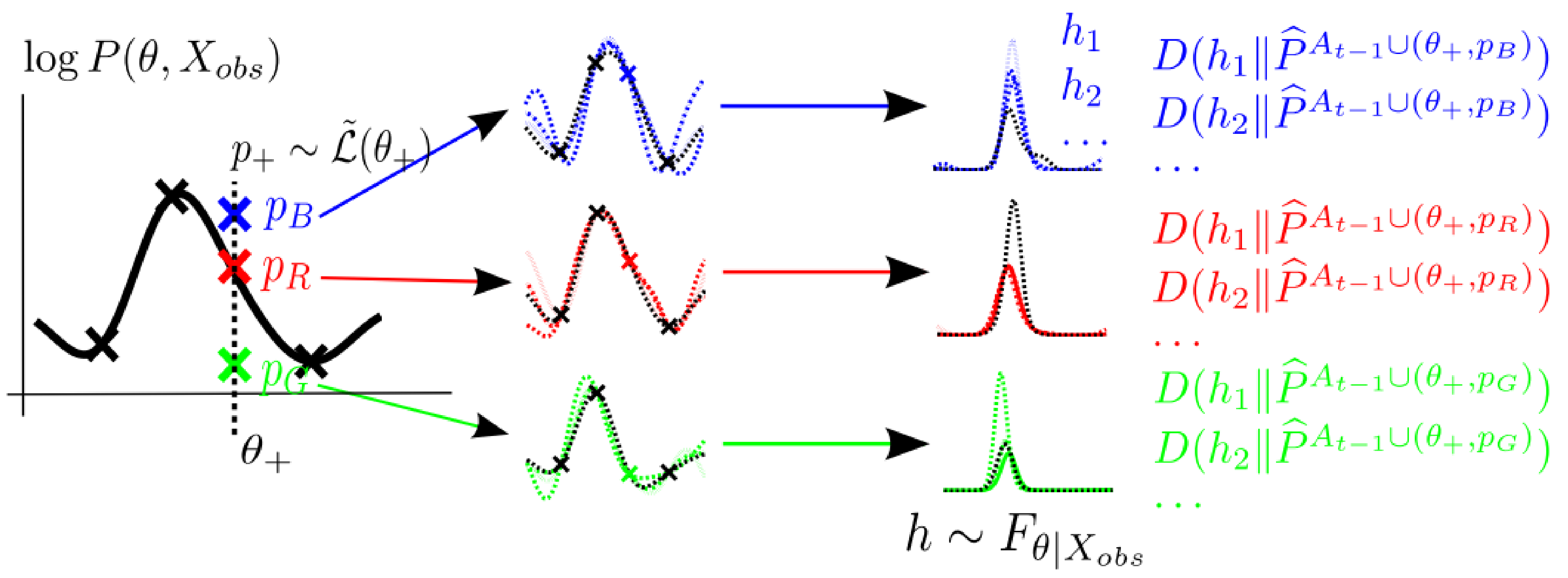
$$B_t = \{\theta_i,\ \log\left(\mathcal{L}(\theta_i)P(\theta_i)\right)\}_{i=1}^{t} \text{ all data collected up to iteration } t.$$

$$= \{\theta_i,\ \log P(\theta_i, X_{obs})\}_{i=1}^{t}$$

Negative Expected Divergence

- $\star$ Fit a GP on this data. $g(\theta) \sim GP(B_n) \quad \Rightarrow g(\theta) \approx \log P(\theta, X_{obs})$
- $\star$ Consider experiment in θ_+
- $\star$ sample from the above GP and evaluate in $\theta_+ \Rightarrow p_B, p_R, p_G$

Negative Expected Divergence


$\log P(\theta, X_{obs})$
$p_+ \sim \tilde{\mathcal{L}}(\theta_+)$
p_B
p_R
p_G
θ_+
h_1
h_2
$\dots$
$D(h_1\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_B)})$
$D(h_2\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_B)})$
$\dots$
$D(h_1\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_R)})$
$D(h_2\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_R)})$
$\dots$
$D(h_1\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_G)})$
$D(h_2\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_G)})$
$\dots$
$h \sim F_{\theta|X_{obs}}$

Negative Expected Divergence $u_t^{NED}(\theta_+) = -\mathbb{E}_{p+}\mathbb{E}_{h(\cdot)}\Big[D\Big(h(\theta)||\hat{P}^{A_{t-1}\cup(\theta_+,\, p_+)}(\theta)\Big)\Big]$

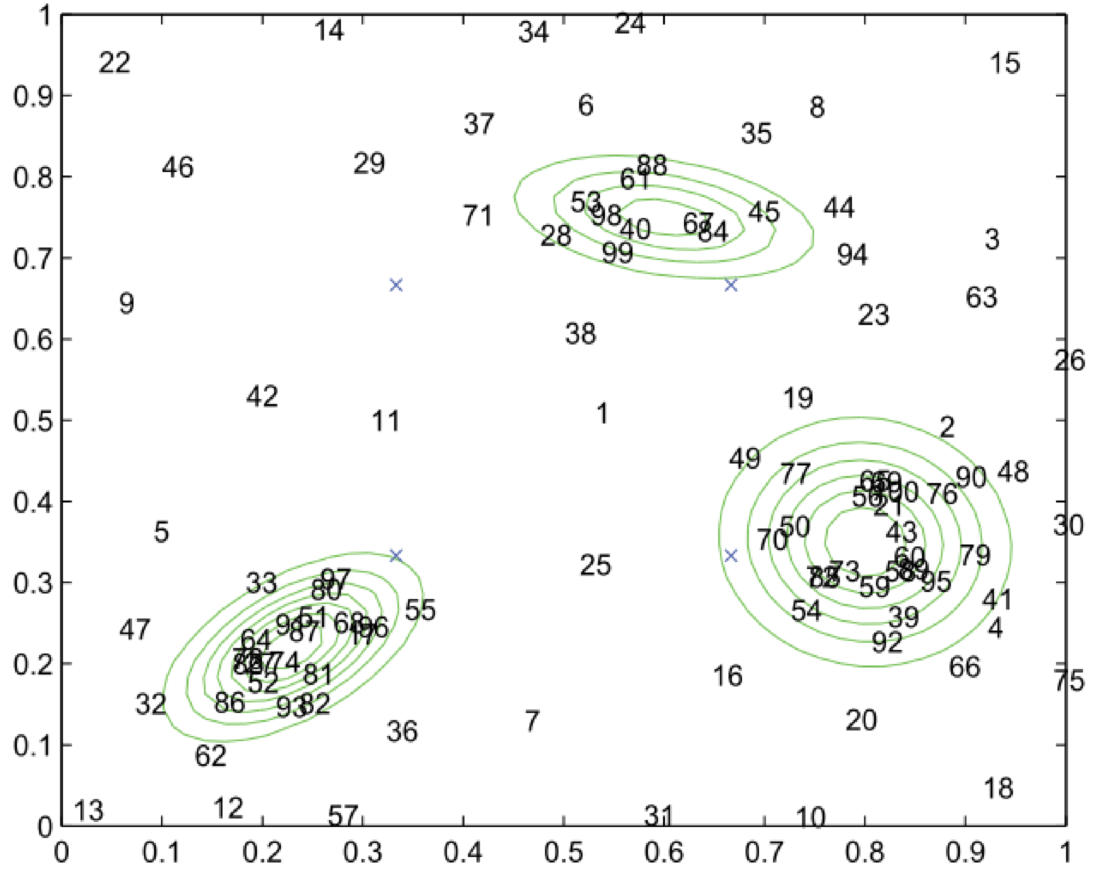
⋆ sample from the above GP and evaluate in $\theta_+ \Rightarrow p_B, p_R, p_G$

⋆ For p_B, we add (θ_+, p_B) as a hallucinated point to the $t-1$ and obtain an estimate of the posterior $\hat{P}^{A_{t-1}\cup(\theta_+,p_B)}$.

⋆ Next, we rebuild our GP using these t points.

⋆ We draw samples from the new GP, exponentiate, and normalise them to obtain samples h_i

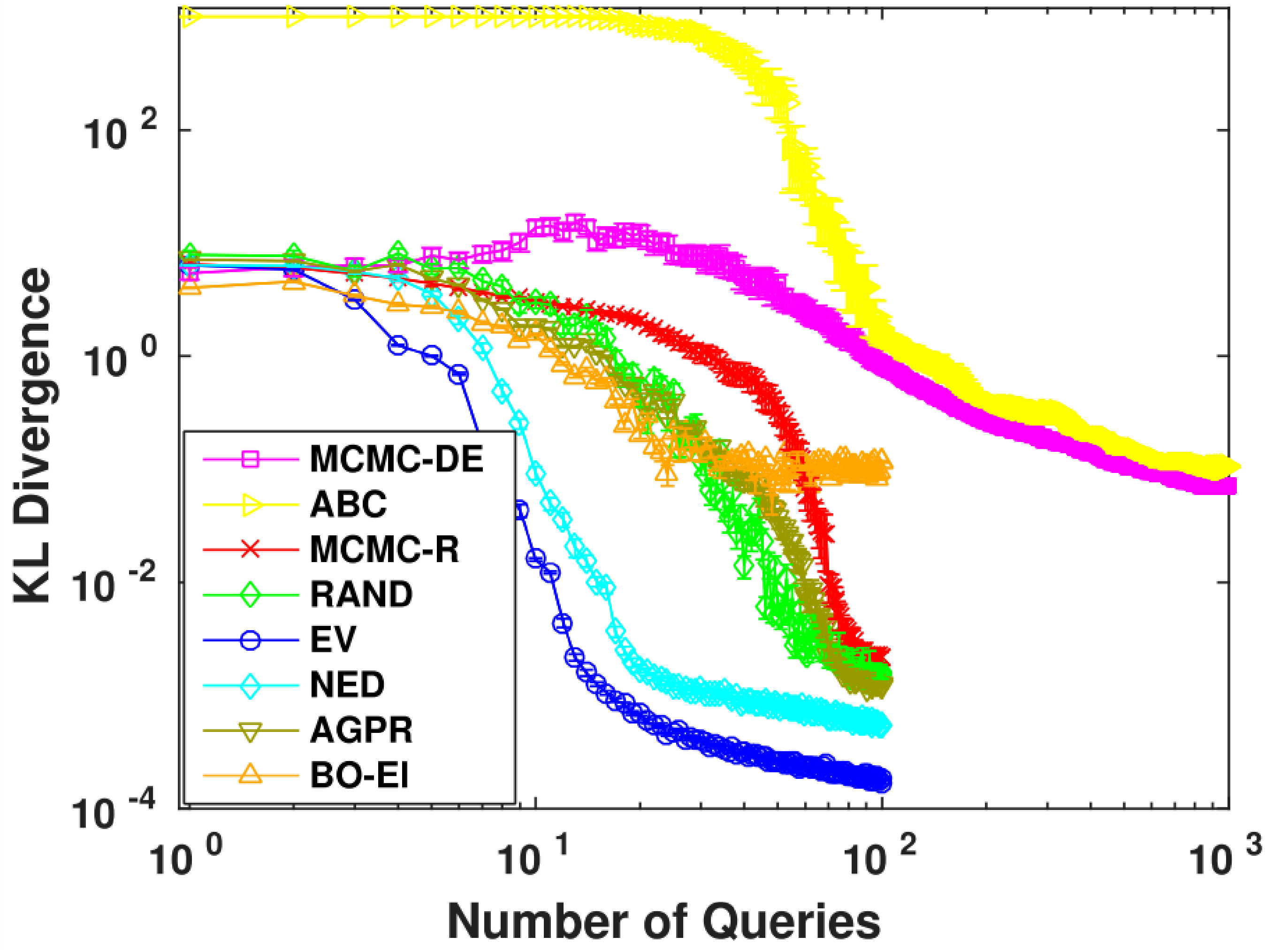
Experiments on Synthetic Data



- 100 points chosen in order by NED for a synthetic 2D experiment.
- The green contours are the true posterior.
- Initially the algorithm explores the space before focusing on high probability regions.

Experiments on Synthetic Data

KL divergence between the estimated posterior and the true posterior



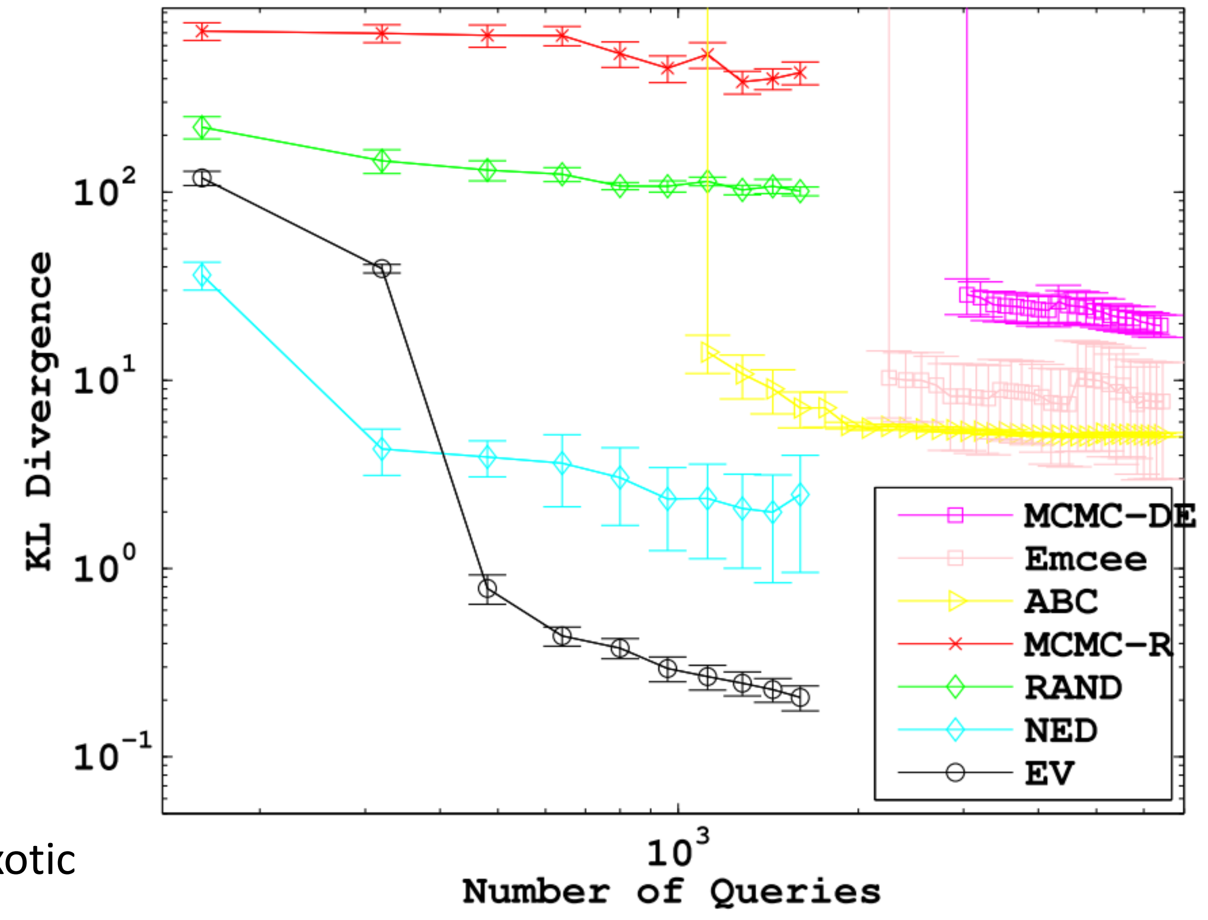
Comparison of NED/EV against MCMC-DE, ABC, MCMC-R and RAND for a 1D synthetic example

Estimating Cosmological Parameters using Type-1 Supernovae data

We use supernovae data for inference on 3 cosmological parameters:

- Hubble Constant ($H_0 \in (60, 80)$,
- Dark Matter Fraction $\Omega_M \in (0, 1)$
- Dark Energy Fraction $\Omega_\Lambda \in (0, 1)$.

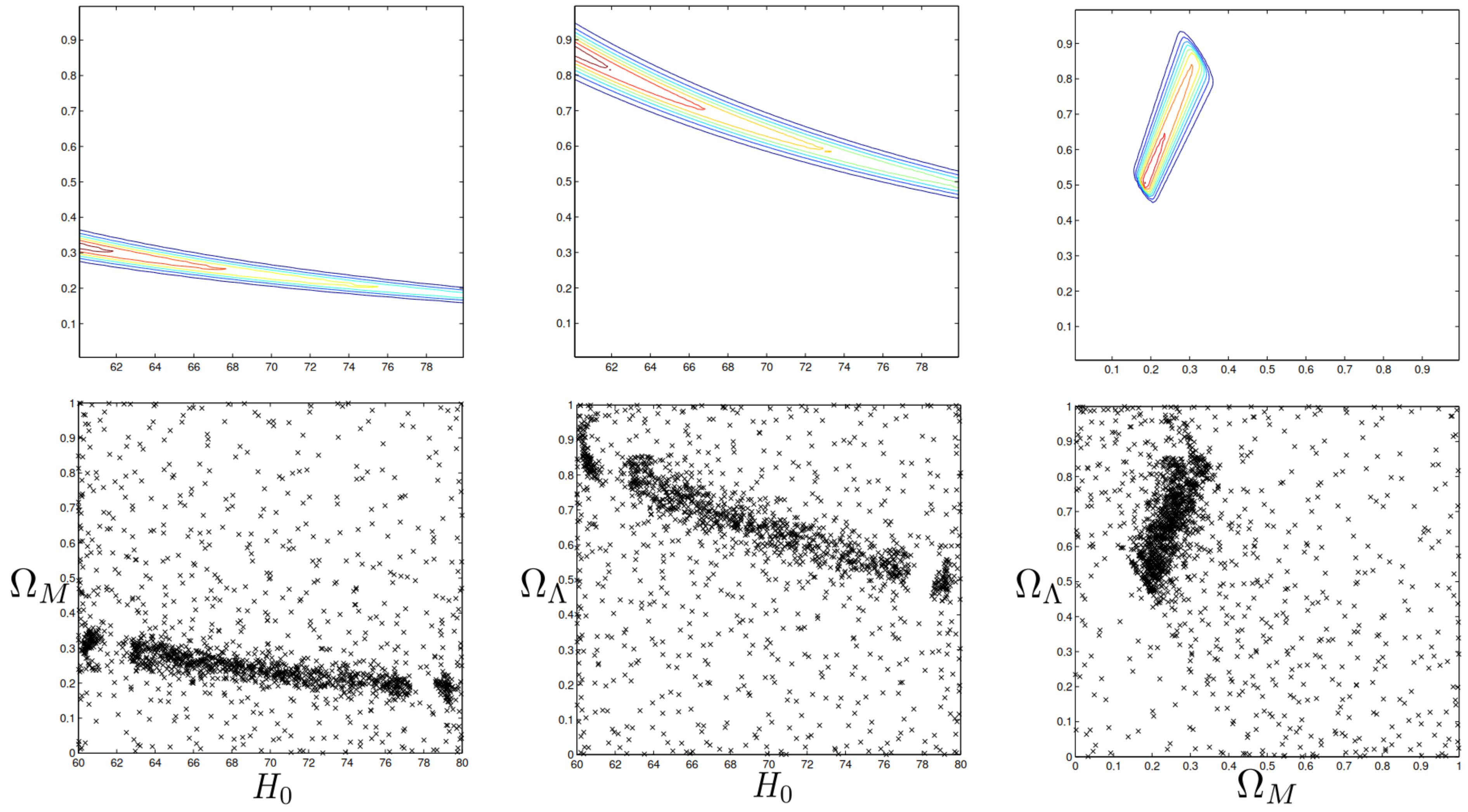
The likelihood for the experiment is given by the Robertson– Walker metric which models the distance to a supernova given the parameters and the observed red-shift.



Dataset is taken from: T. M. Davis et al. Scrutinizing Exotic Cosmological Models Using ESSENCE Supernova Data Combined with Other Cosmological Probes.
The Astrophysical Journal, pages 716–725, 2007.

KL was approximated via numeric integration.

Estimated Posterior Distributions



Projections of the points selected by EV (bottom row) and the marginal distributions (top row).

Thanks for your Attention!