Detector optimization

Detectors

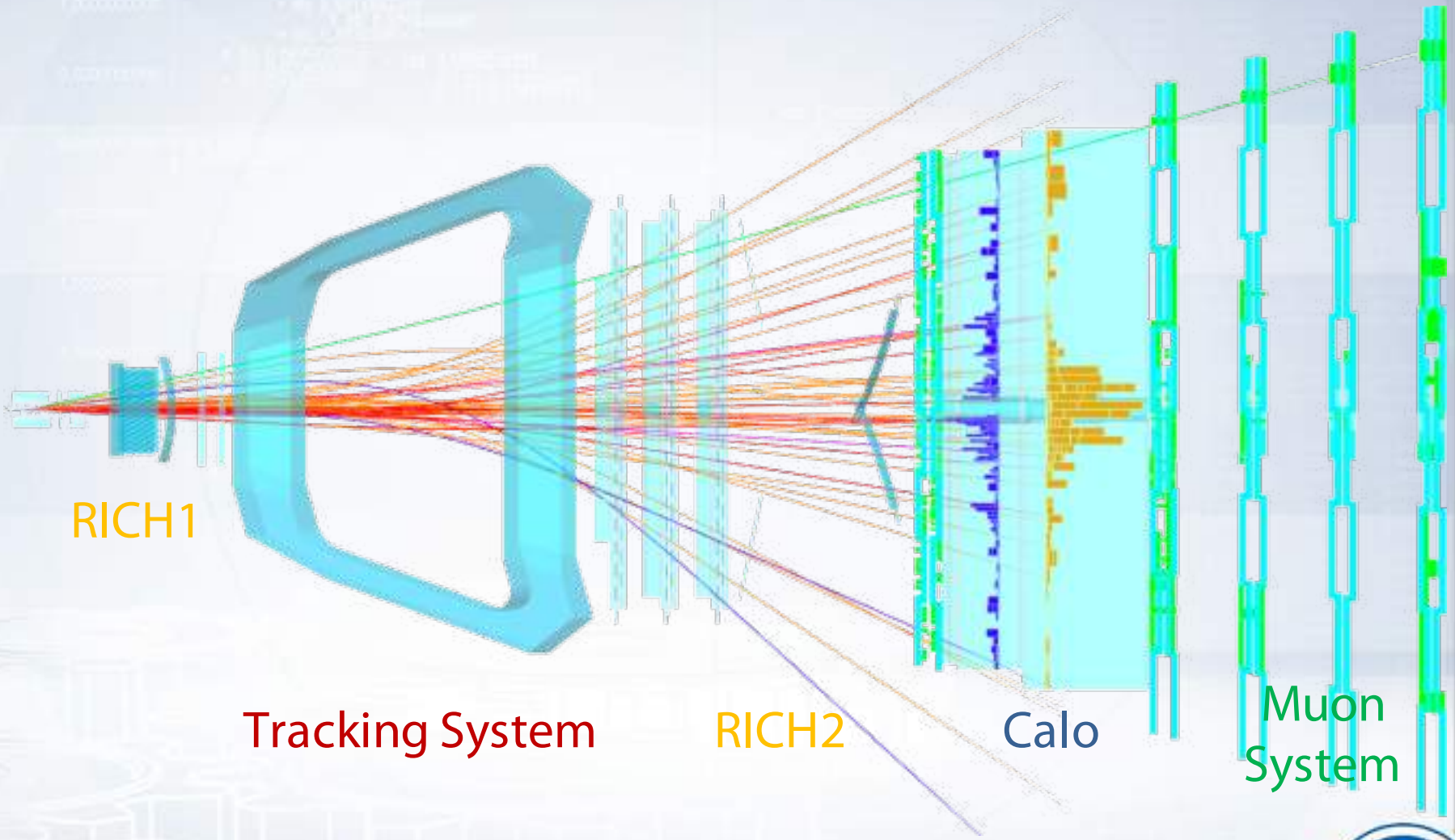
Detectors in high energy physics experiments are sophisticated and expensive.

The detectors consist of different parts: tracking system, Ring Image Cherenkov detector, electromagnetic and hadronic calorimeters, muon system.

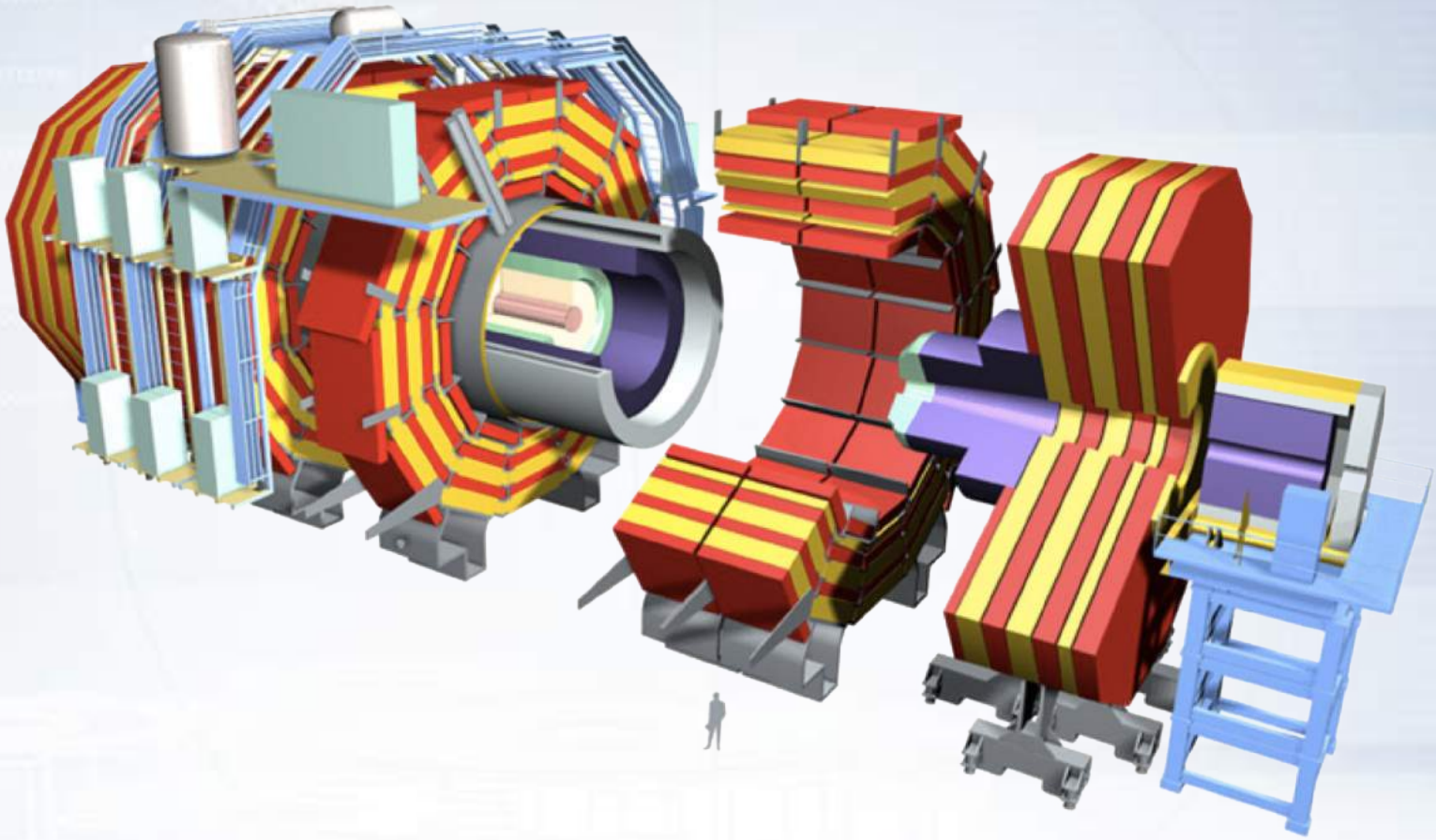
Each of these parts have thousands of sensors which register particles and measure their momentum and energy.

Layout of the sensors affects precision, efficiency and cost of the detectors.

LHCb detector

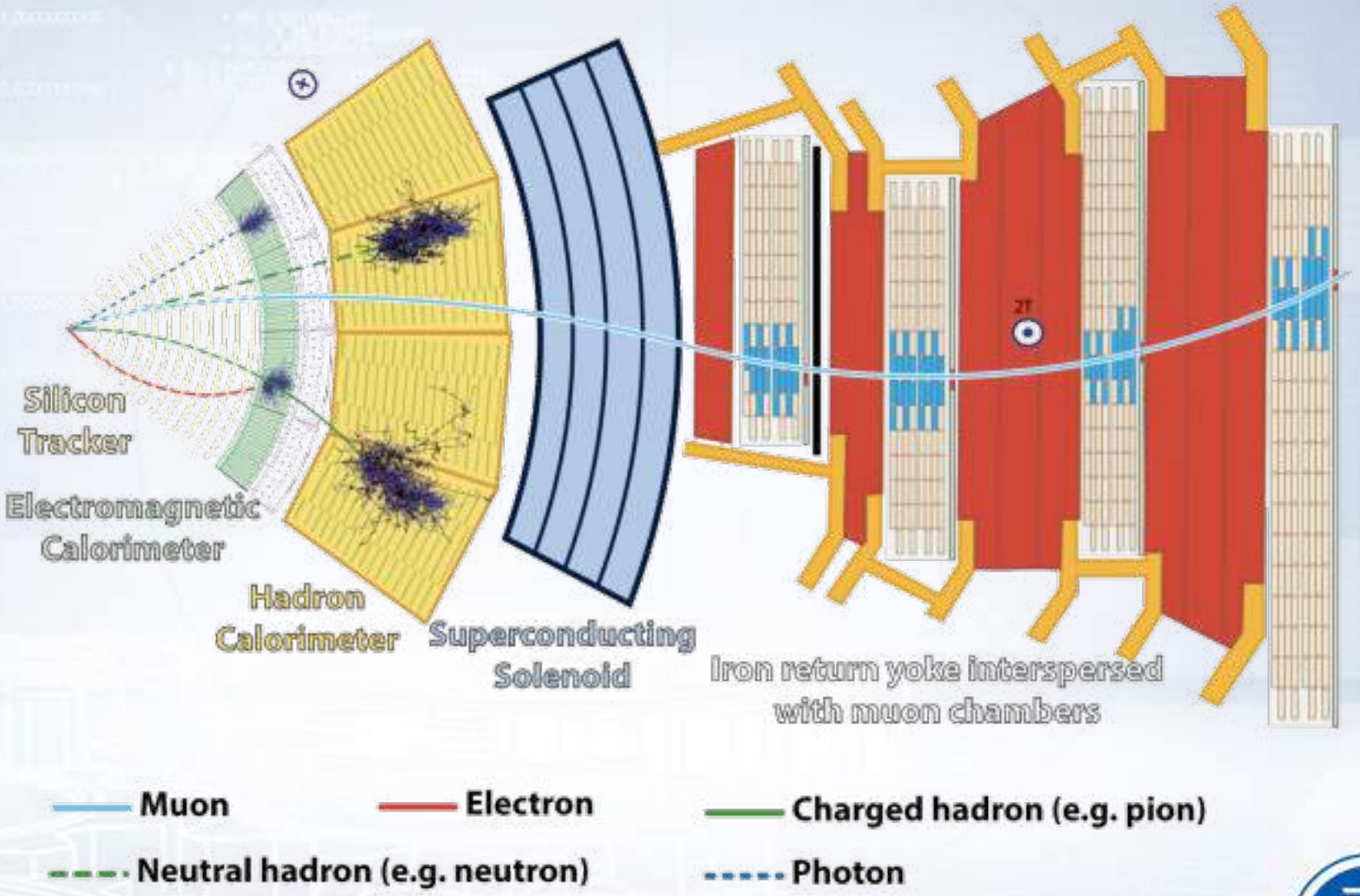


CMS detector



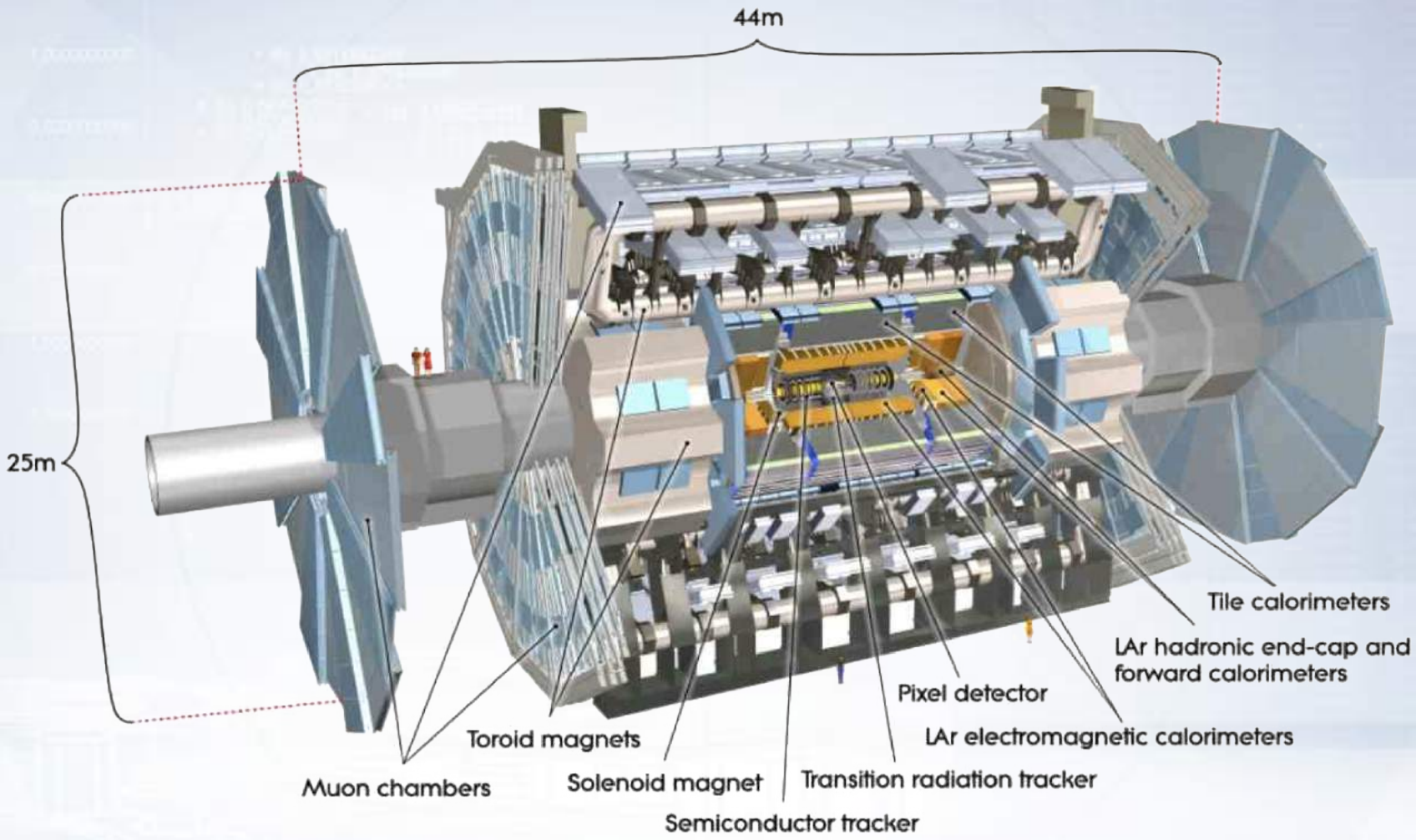
Hephy / http://www.hephy.at/user/friedl/diss/html/node8.html

CMS detector



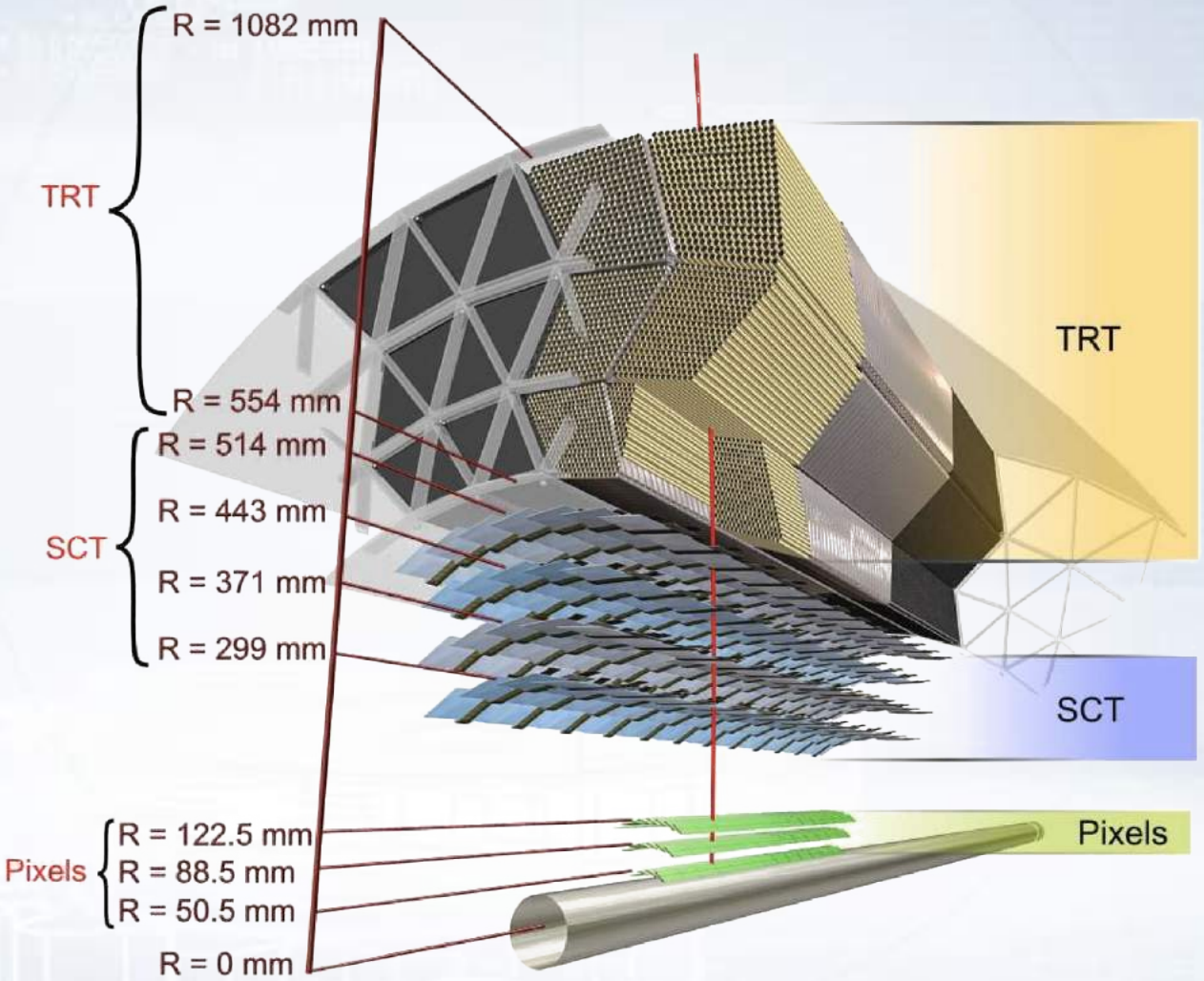
CMS / https://cds.cern.ch/record/2120661

ATLAS detector



ATLAS / http://inspirehep.net/record/834202/files/figure1.png

ATLAS inner tracking system



ATLAS / https://search.proquest.com/docview/1619175164?pq-origsite=gscholar

Detector optimization

The goal of the detector optimization is to find optimal layout of sensors in the detector.

To do this an objective function for optimization must be defined.

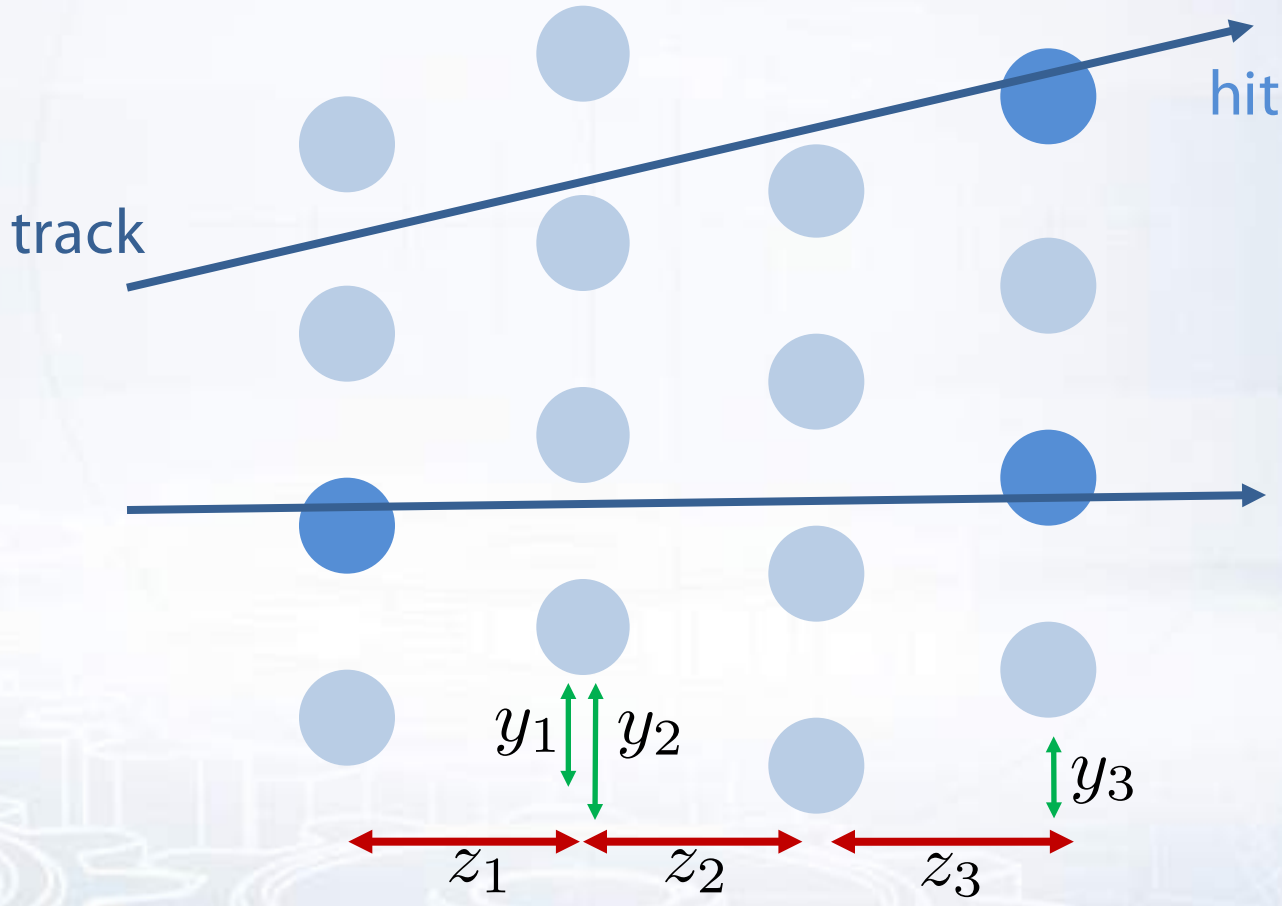
This function aggregates key values needed to be optimized: cost of the detector, track resolution, track reconstruction efficiency, precision of the momentum and energy reconstruction, etc..

Set of parameters that define the sensor layout and affect the objective function values must be selected.

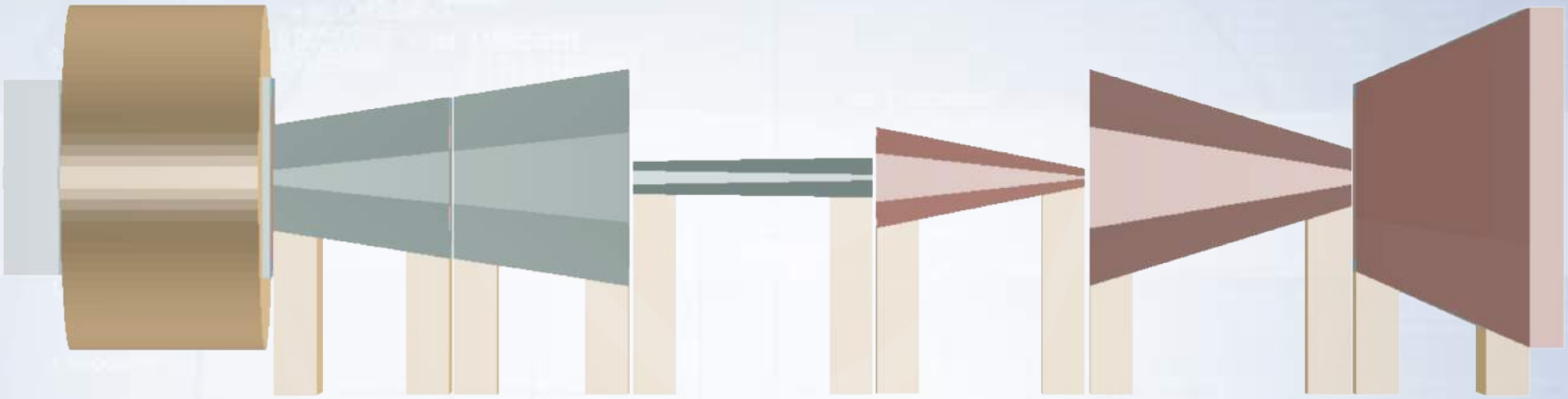
Search for the parameters values correspond to the optimum of the objective function.

Simple tracking system example

The tracking system geometry can be defined by the shifts between the layers: $y_1, y_2, y_3, z_1, z_2, z_3$. A possible objective function is the number of hits per one track.



SHiP muon shield example



The shield consists of eight magnets and each magnet is parameterized by seven values: length, width, etc..

The objective function depends on the physical performance of the shield (muon background) and its weight.

Baranov A. et. al., Optimising the Active Muon Shield for the SHiP Experiment at CERN, IOP Conf. Series: Journal of Physics: Conf. Series 934 (2017) 012050 doi :10.1088/1742-6596/934/1/012050

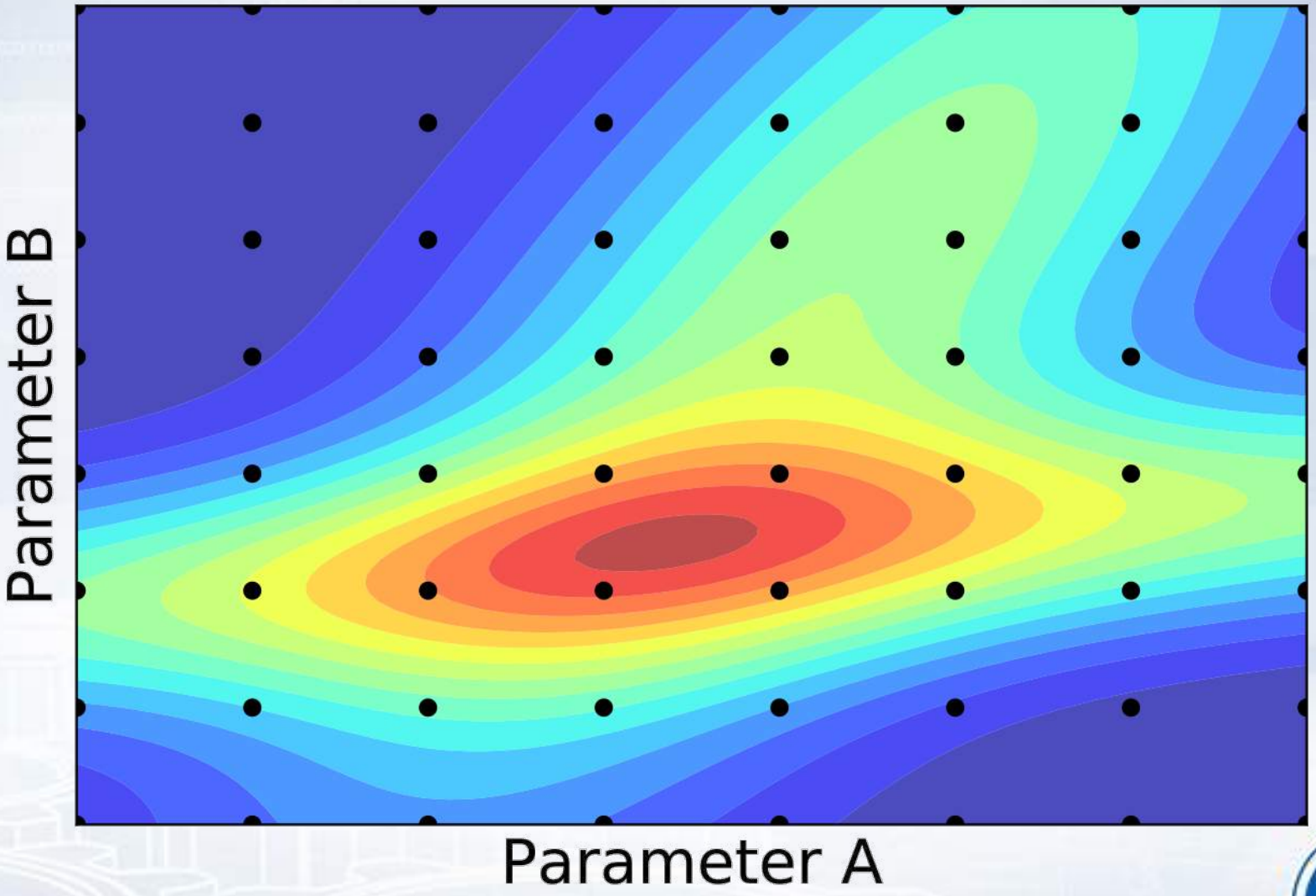
Grid search

Grid search is one of the most popular and the simplest methods of optimization.

It defines a grid of the parameter values and calculates the objective function values for each node in the grid.

The node with the highest/least function value is taken as the optimum.

Grid search



Grid search

Grid search is reasonable to use when the number of parameters is small.

The grid size exponentially grows with number of parameters to optimize.

It became more important when the objective function is expensive to evaluate. Thus, the grid search requires large computational resources.

Detectors optimization in high energy physics requires to run Monte-Carlo simulation of these detectors. Thus, the optimization requires large computational resources.

Bayesian optimization

Bayesian optimization is a method of finding the optimum of an expensive function.

The goal of the Bayesian optimization is to find the optimum of the objective function using as small number of the function calculations as possible.

Normal distribution

Normal distribution

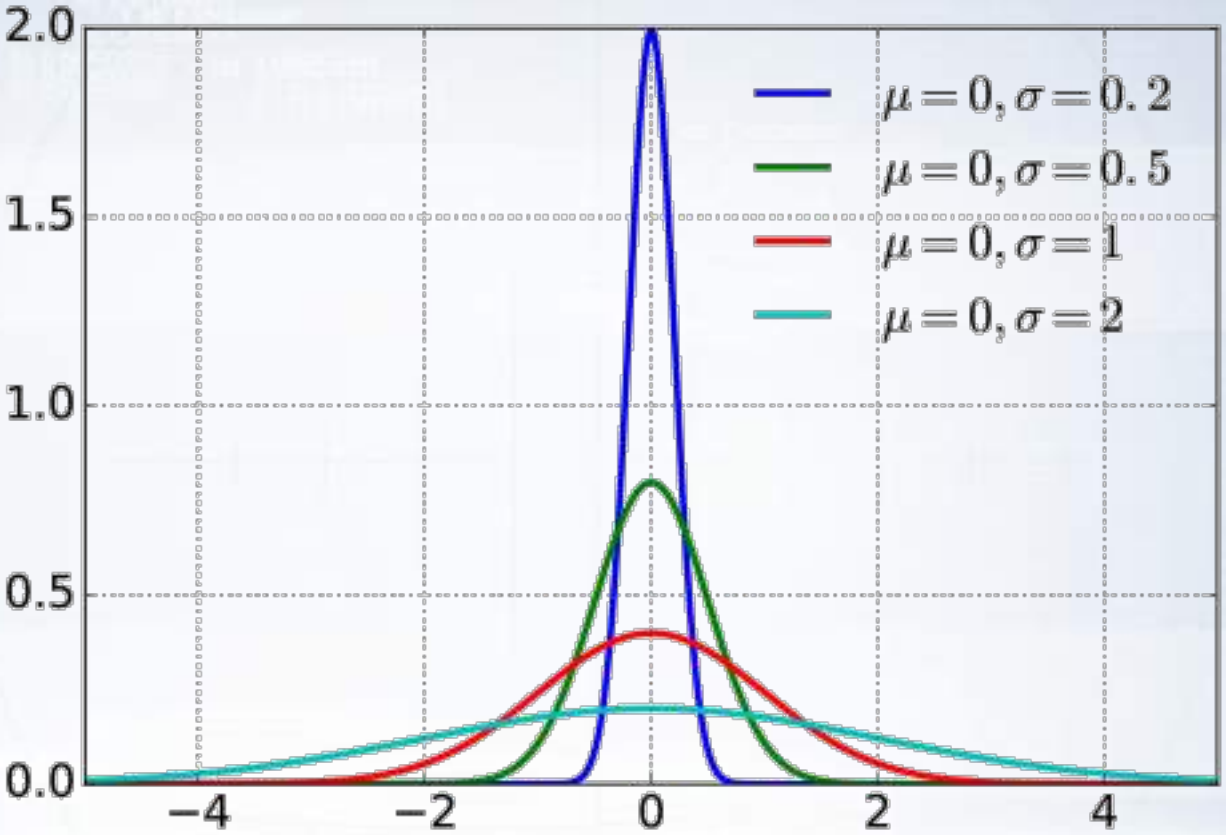
The normal (or Gaussian) distribution is continuous probability distribution. The probability density in 1D case is:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where

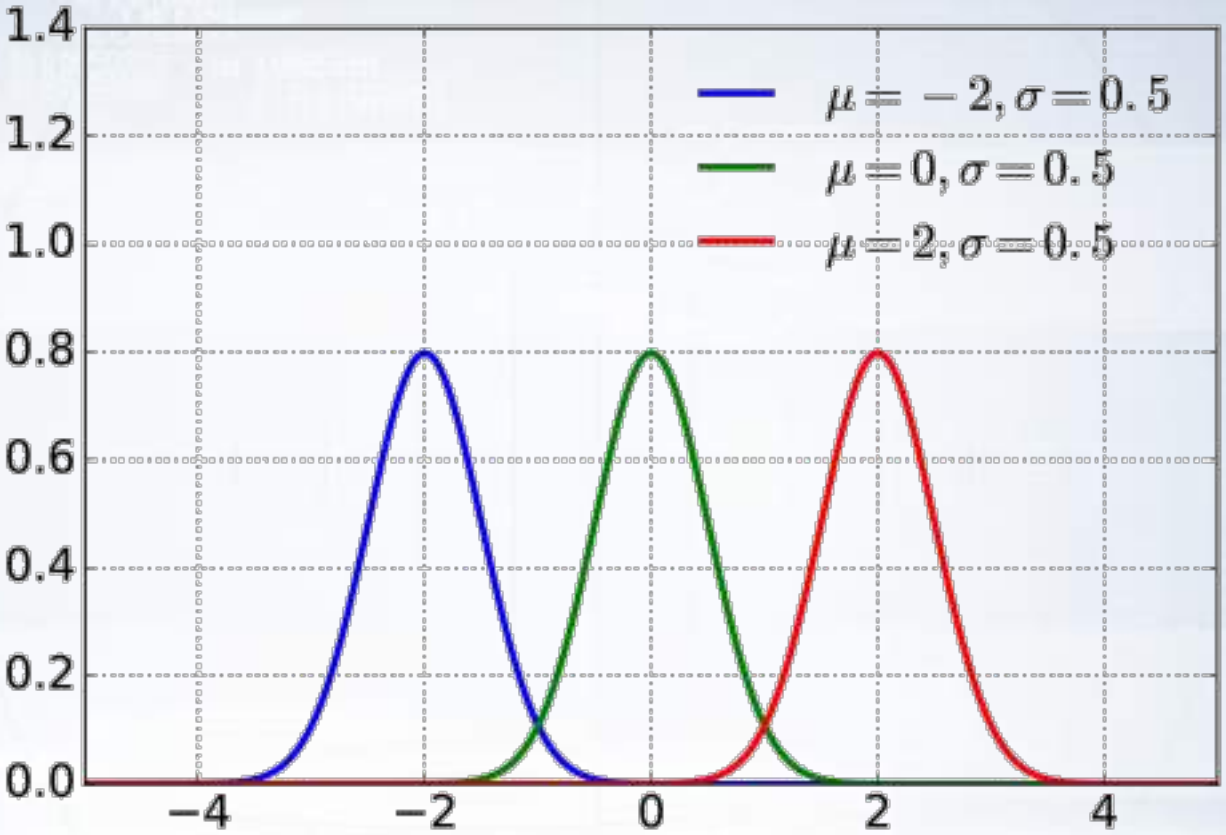
- x is the random value,
- $\mu = E[x]$ is the mean of the distribution,
- $\sigma^2 = E[(x-\mu)^2]$ is the variance,
- σ is the standard deviation.

Normal distribution



$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal distribution



$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Multivariate normal distribution

The multivariate normal (or Gaussian) distribution is a generalization of the one-dimensional normal distribution to higher dimensions. The probability density in N dimensions is:

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

where

- $x = (x_1, x_2, \ldots, x_N)^T$ is the N-dimensional random vector,
- $\mu = (\mu_1, \mu_2, \ldots, \mu_N)^T$ is the N-dimensional mean vector,
- Σ is the $N \times N$ covariance matrix,
- $|\Sigma| \equiv \det \Sigma$ is the determinant of Σ.

Multivariate normal distribution

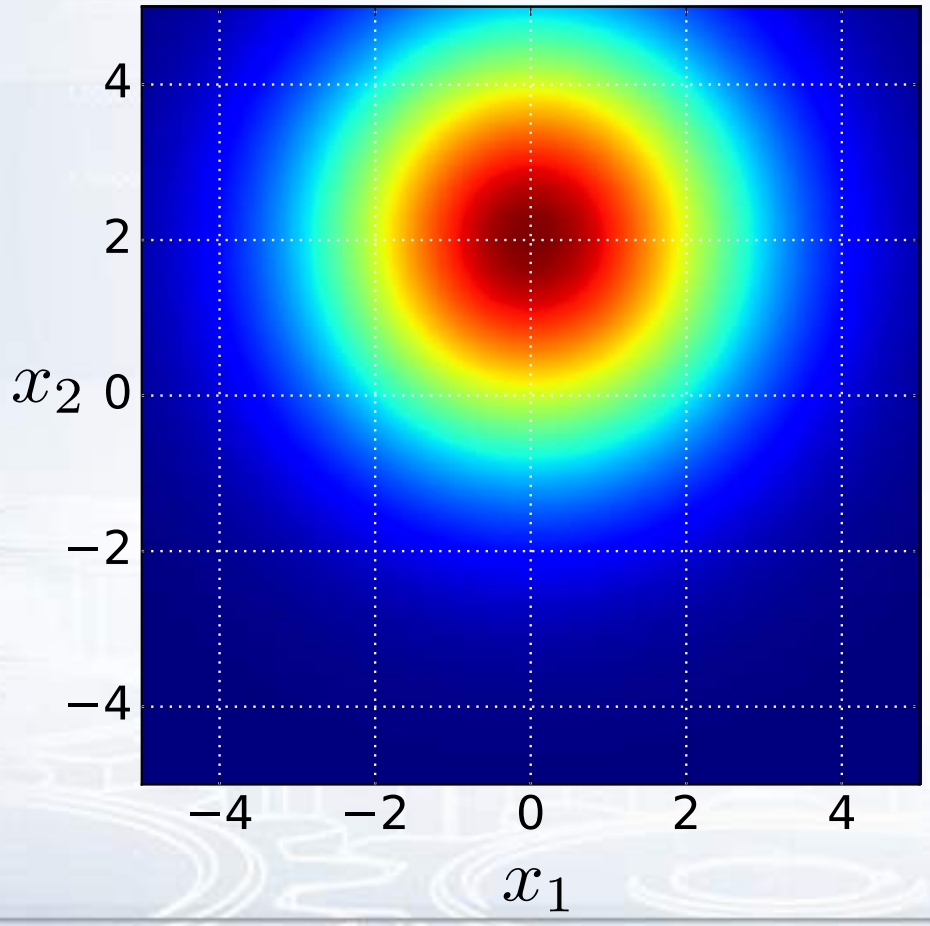
$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

- $x = (x_1, x_2, \ldots, x_N)^T$ is the N-dimensional random vector,
- $\mu = (\mu_1, \mu_2, \ldots, \mu_N)^T = (E[x_1], E[x_2], \ldots, E[x_N])^T$,
- $$\Sigma = \begin{pmatrix} Cov(x_1, x_1) & \ldots & Cov(x_1, x_N) \\ \vdots & \ddots & \vdots \\ Cov(x_N, x_1) & \ldots & Cov(x_N, x_N) \end{pmatrix},$$

where $Cov(x_i, x_j) = E[(x_i - \mu_i)(x_j - \mu_j)]$.

2D example

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

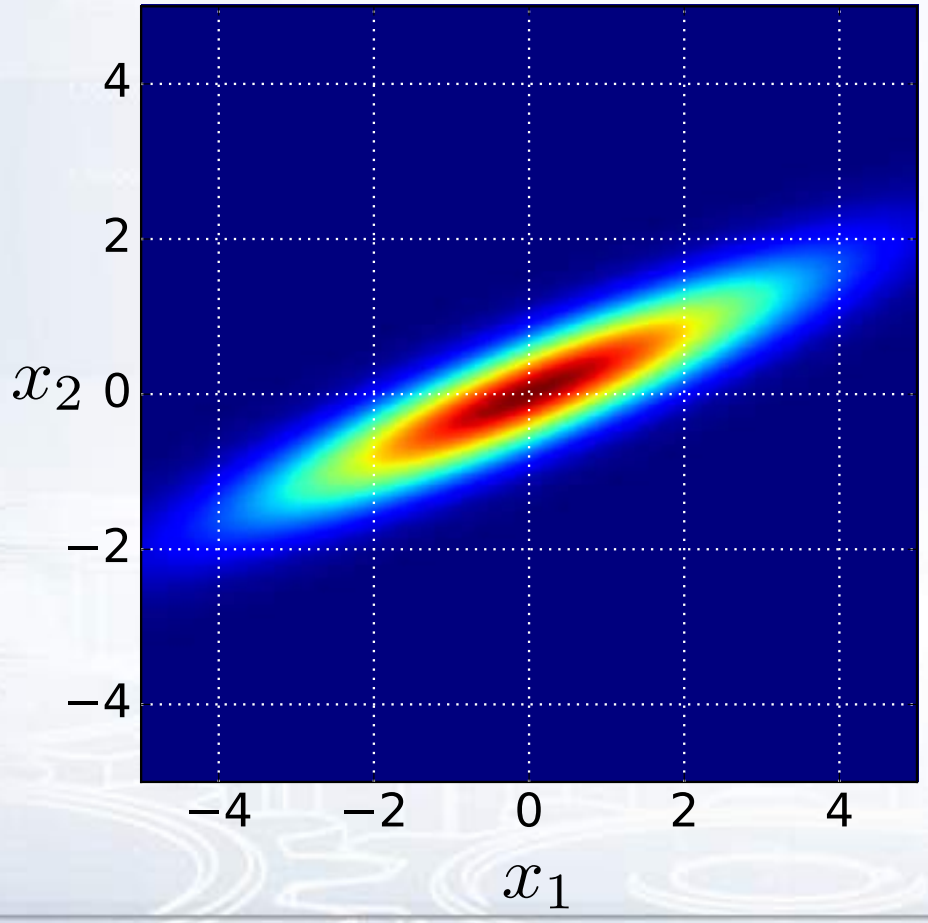


$$\mu = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$$

2D example

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$



$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}$$

Conditional distribution

Consider the multivariate normal distribution:

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Suppose the following:

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}$$

where

- x_a is a vector of size $k \times 1$,
- x_b is a vector of size $(N-k) \times 1$.

Conditional distribution

Consider the multivariate normal distribution:

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Suppose the following:

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}$$

Also:

$$\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

Conditional distribution

Then the conditional normal distribution is:

$$p(x_a|x_b) = \mathcal{N}(\mu_{a|b}, \Sigma_{a|b})$$

where

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b)$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

where

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}, \mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

Marginal distribution

Then the marginal normal distribution is:

$$p(x_a) = \int p(x_a, x_b) dx_b = \mathcal{N}(\mu_a, \Sigma_{aa})$$

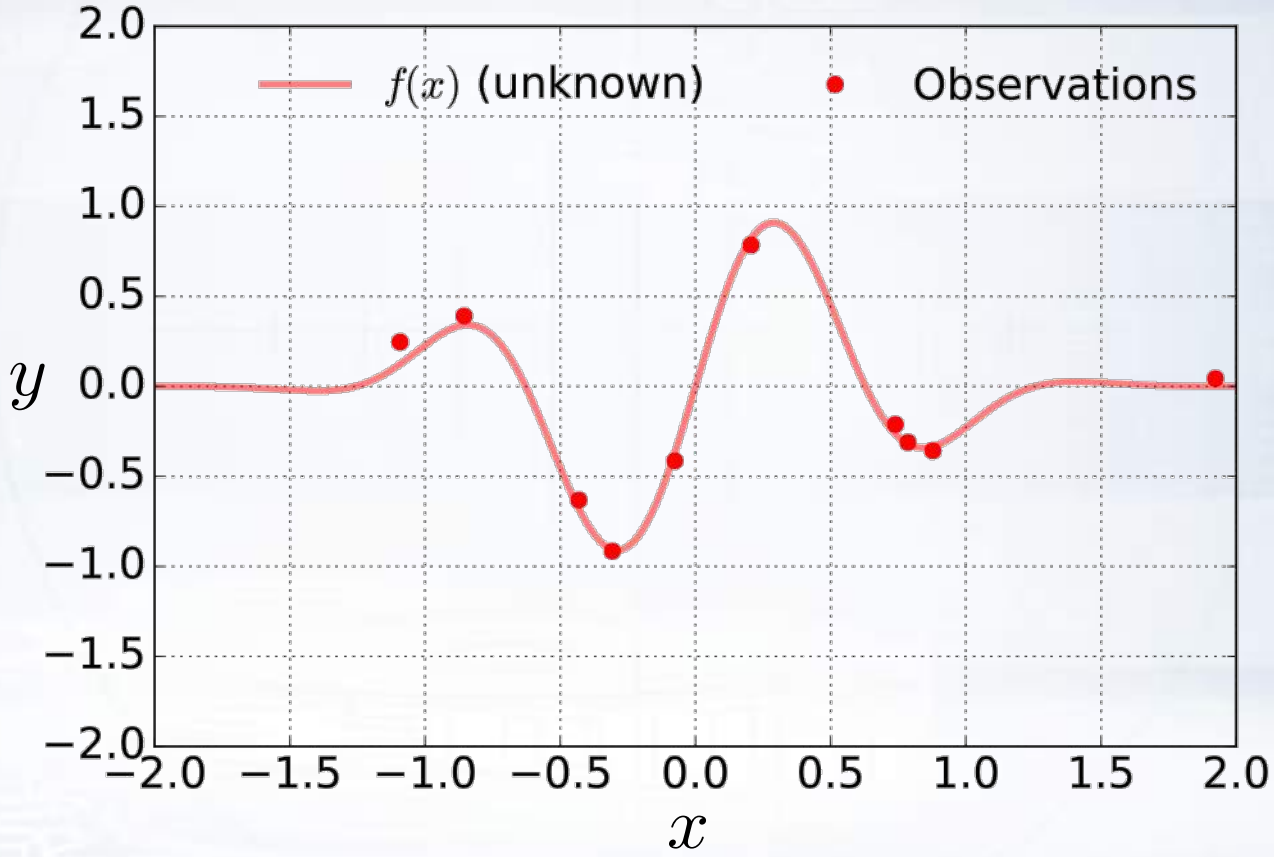
where

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}, \mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

Gaussian processes for regression

Regression example

Consider the following regression problem:



With training set $\{x_i, y_i\}_{i=0}^{N}$.

Gaussian processes

Take into account noise of the observations:

$$y = f + \varepsilon$$

where

- $y = (y_1, y_2, \ldots, y_N)^T$ is the vector of the target observations,
- $f = (f(x_1), f(x_2), \ldots, f(x_N))^T$ is the vector of the true function values,
- ε is noise of the observations.

Gaussian processes

In the Gaussian process assumption:

$$p(y|f) = \mathcal{N}(f, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(y-f)^T \Sigma^{-1} (y-f)}$$

where

- $y = (y_1, y_2, \ldots, y_N)^T$ is the vector of the target observations,
- $f = (f(x_1), f(x_2), \ldots, f(x_N))^T$ is the vector of the true function values,
- $\Sigma = \alpha \begin{pmatrix} 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \end{pmatrix}$ is the covariance matrix.

Gaussian processes

Also suppose that:

$$p(f) = \mathcal{N}(0, K) = \frac{1}{\sqrt{(2\pi)^N |K|}} e^{-\frac{1}{2}(f)^T K^{-1}(f)}$$

where

- $f = (f(x_1), f(x_2), \ldots, f(x_N))^T$ is the vector of the true function values,
- K is the covariance matrix.

The matrix K is chosen to express the property that, for points x_n and x_m that are similar, the values $f(x_n)$ and $f(x_m)$ will be more strongly correlated than for dissimilar points.

Covariance matrix

The matrix K is chosen to express the property that, for points x_n and x_m that are similar, the values $f(x_n)$ and $f(x_m)$ will be more strongly correlated than for dissimilar points.

$$K = \begin{pmatrix} Cov(f(x_1), f(x_1)) & \dots & Cov(f(x_1), f(x_N)) \\ \vdots & \ddots & \vdots \\ Cov(f(x_N), f(x_1)) & \dots & Cov(f(x_N), f(x_N)) \end{pmatrix}$$

where

$$Cov(f(x_i), f(x_j)) = k(x_i, x_j) = \sigma^2 e^{-d^2(x_i, x_j)}$$

where σ is constant and d is euclidean distance.

Gaussian processes

Then:

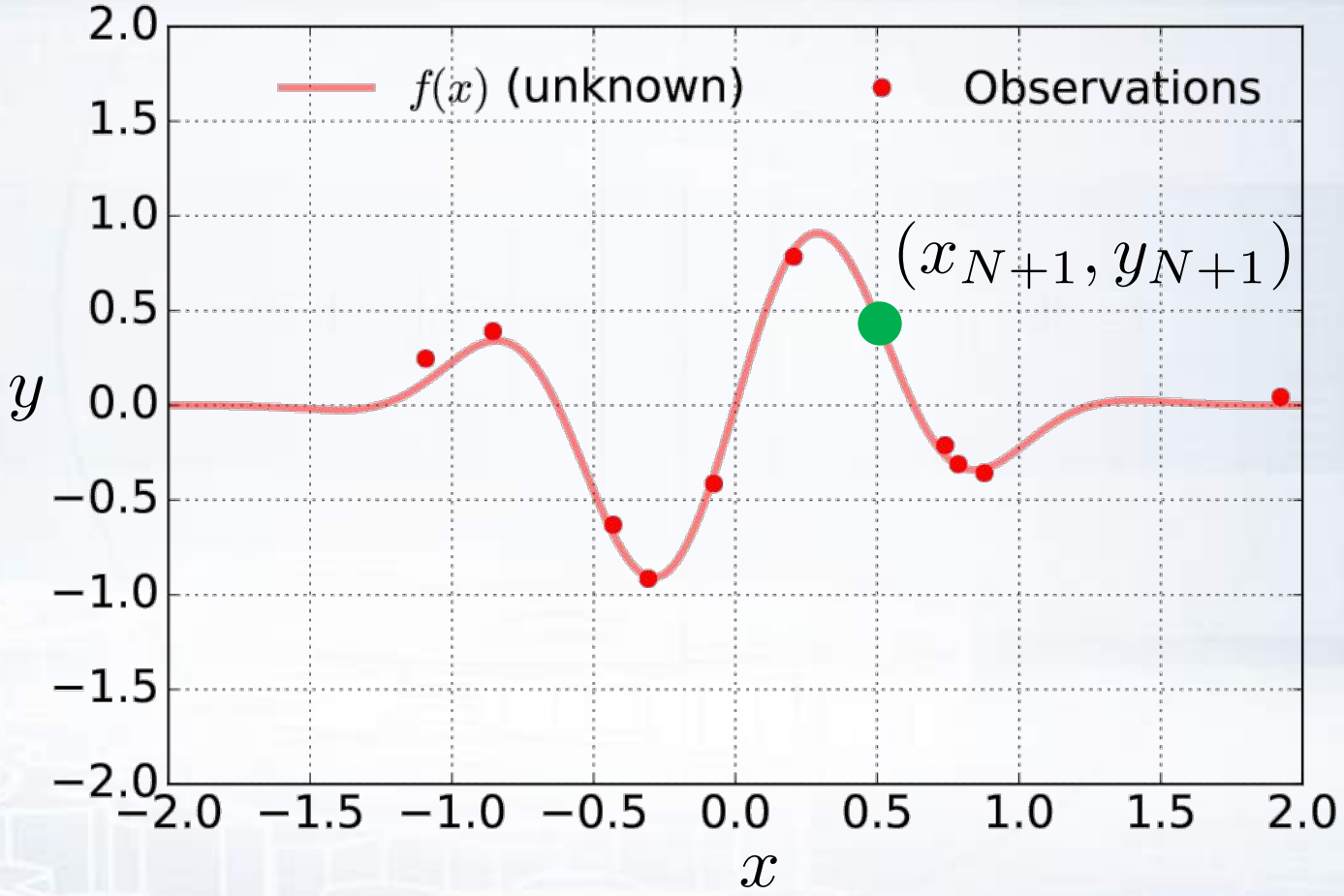
$$p(y) = \int p(y|f)p(f)df = \mathcal{N}(0, C)$$

where

- $C = K + \alpha \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{pmatrix}$ is the covariance matrix.

Regression example

Let's estimate y_{N+1} for a given x_{N+1}:



Gaussian processes

Then:

$$p(y_{N+1\times 1}) = (0, C_{N+1})$$

where

- $y_{N+1\times 1} = (y_1, y_2, \ldots, y_N, y_{N+1})^T$ is the vector of the target observations,
- $C_{N+1} = \begin{pmatrix} C_N & k \\ k^T & c \end{pmatrix}$ is the covariance matrix.

Gaussian processes

Then the conditional normal distribution:

$$p(y_{N+1}|y_{N\times 1}) = \mathcal{N}(\mu_{GP}(x_{N+1}), \sigma_{GP}(x_{N+1}))$$

where

- $\mu_{GP}(x_{N+1})$ is the mean value of the observation,
- $\sigma_{GP}(x_{N+1})$ is the standard deviation.

Conditional distribution

Then the conditional normal distribution is:

$$p(x_a|x_b) = \mathcal{N}(\mu_{a|b}, \Sigma_{a|b})$$

where

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b)$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

where

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}, \mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

Gaussian processes

Then the conditional normal distribution:

$$p(y_{N+1}|y_{N\times 1}) = \mathcal{N}(\mu_{GP}(x_{N+1}), \sigma_{GP}(x_{N+1}))$$

where

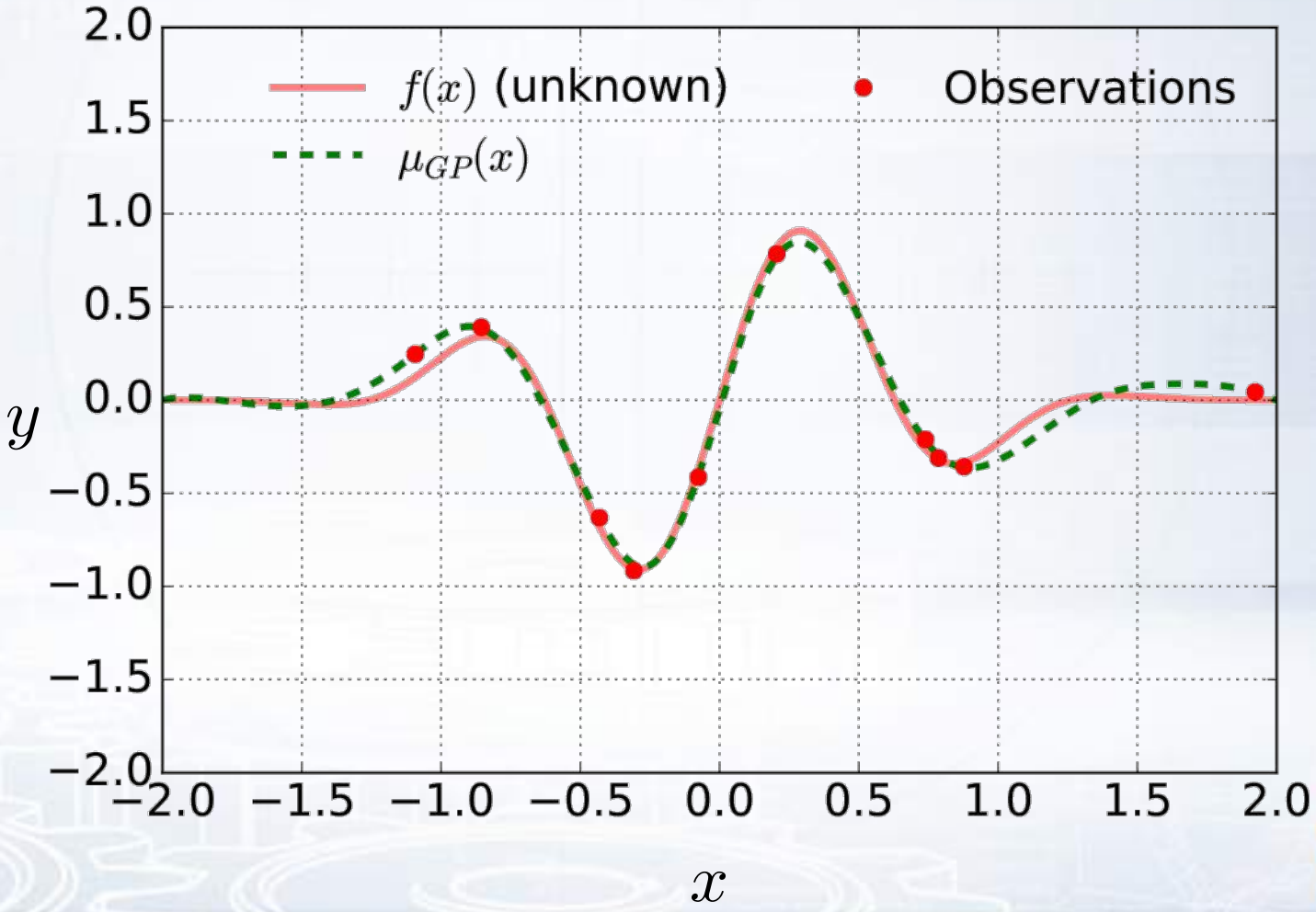
- $\mu_{GP}(x_{N+1})$ is the mean value of the observation,
- $\sigma_{GP}(x_{N+1})$ is the standard deviation.

$$\mu_{GP}(x) = k^T C_N^{-1} y_{N\times 1}$$

$$\sigma_{GP}(x) = c - k^T C_N^{-1} k$$

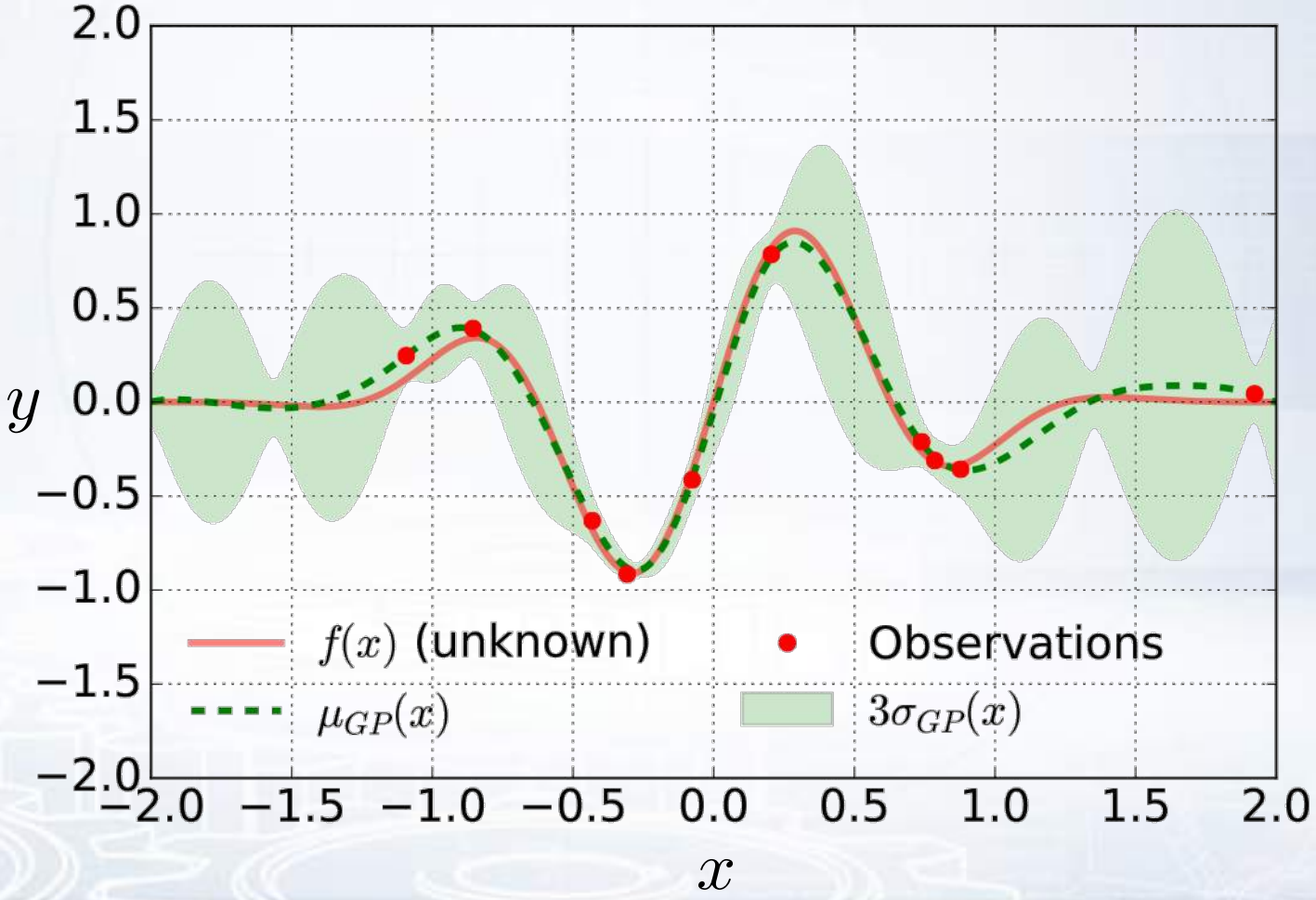
Regression example

$$\mu_{GP}(x) = k^T C_N^{-1} y_{N\times 1}$$



Regression example

$$\sigma_{GP}(x) = c - k^T C_N^{-1} k$$



Bayesian optimization
Part 1

Bayesian optimization

Bayesian optimization is a method of finding the optimum of expensive cost function.

This cost function is also called *objective* function and denoted as $f(x)$.

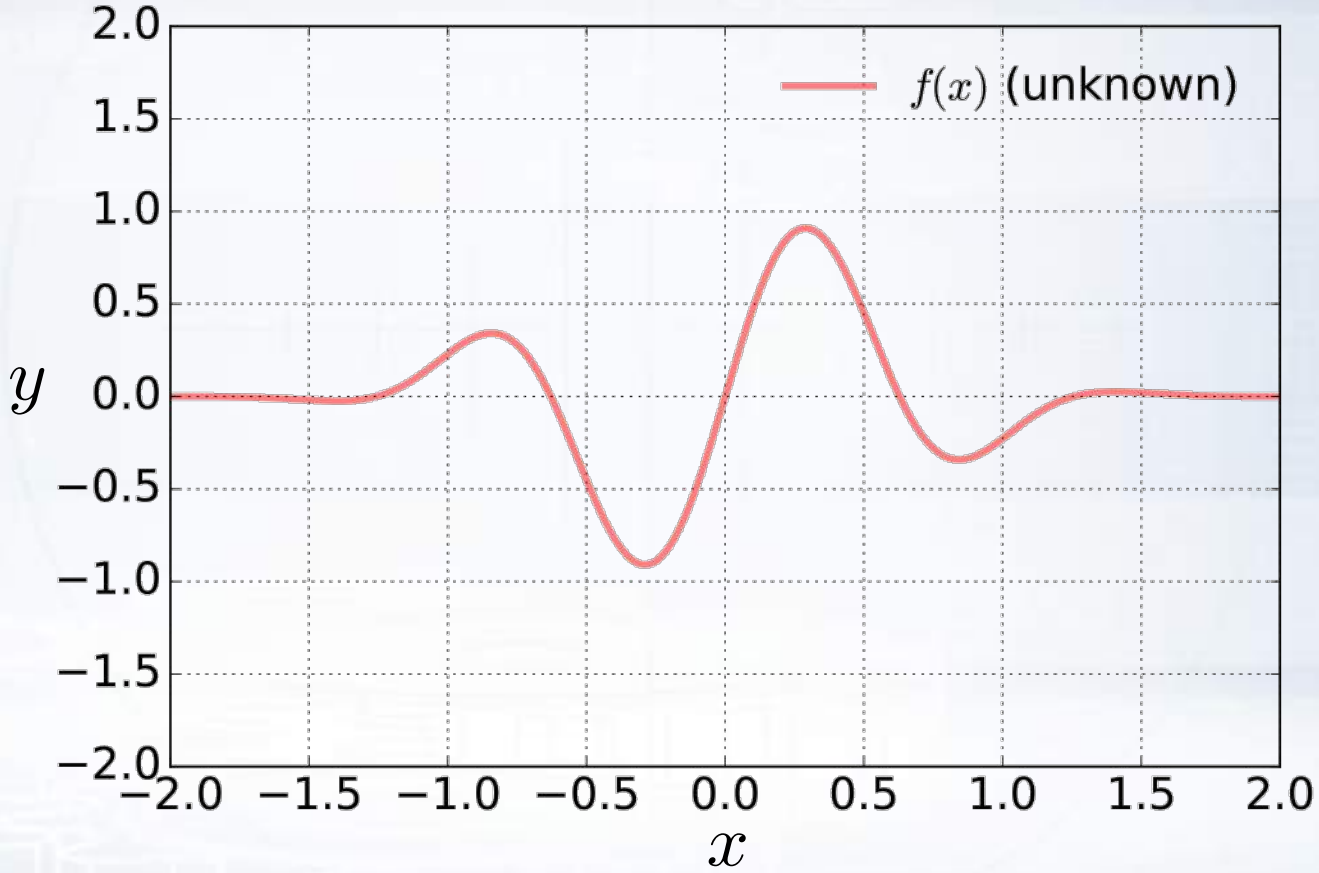
It supposed that calculation of $f(x)$ at one point is expensive.

The derivatives of the objective function are unknown.

The goal of the Bayesian optimization is to find the optimum of the objective function using as small number of the function calculations as possible.

Optimization example

Consider the following objective function $f(x)$:



The goal is to find its minimum.

Bayesian optimization

Algorithm:

1. Find the objective function approximation using previously calculated values $\{x_i, y_i\}_{i=1}^{N}$ solving a regression problem.
2. Using the approximation find the optimum point of an *acquisition* function $u_N(x) : x_{N+1} = \mathrm{argmax}_x u_N(x)$
3. Sample the objective function: $y_{N+1} = f(x_{N+1}) + \varepsilon_{N+1}$
4. Repeat the steps.

Acquisition function

There are variety of acquisition functions. One of them is Lower Confidence Bound (LCB) for the objective function minimization:

$$LCB(x) = \mu(x) - k\sigma(x)$$

where

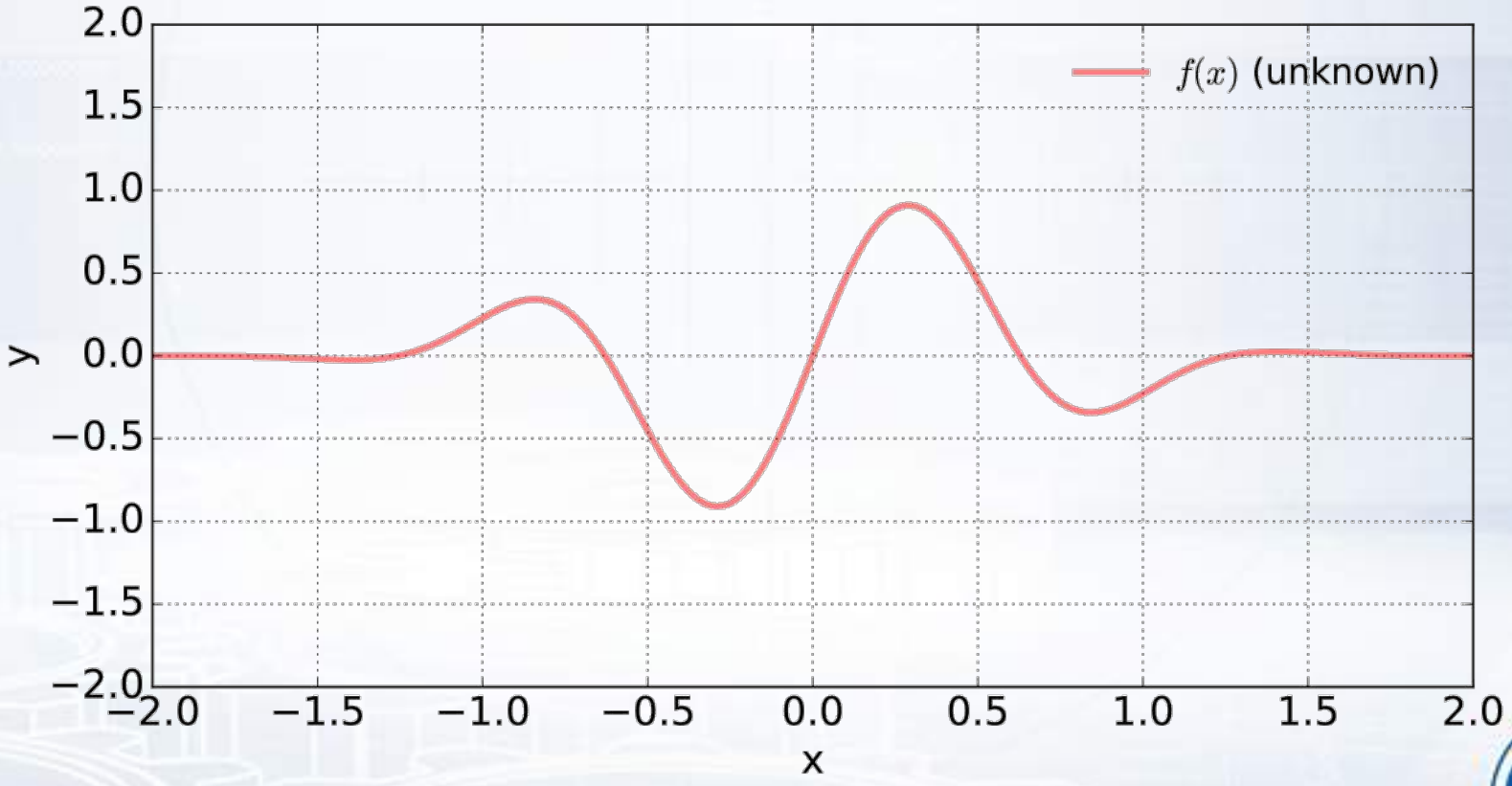
- $\mu(x)$ is mean value of the approximation of the objective function,
- $\sigma(x)$ is standard deviation of the approximation,
- k is adjustable parameter.

Upper Confidence Bound (LCB) for the objective function maximization:

$$UCB(x) = \mu(x) + k\sigma(x)$$

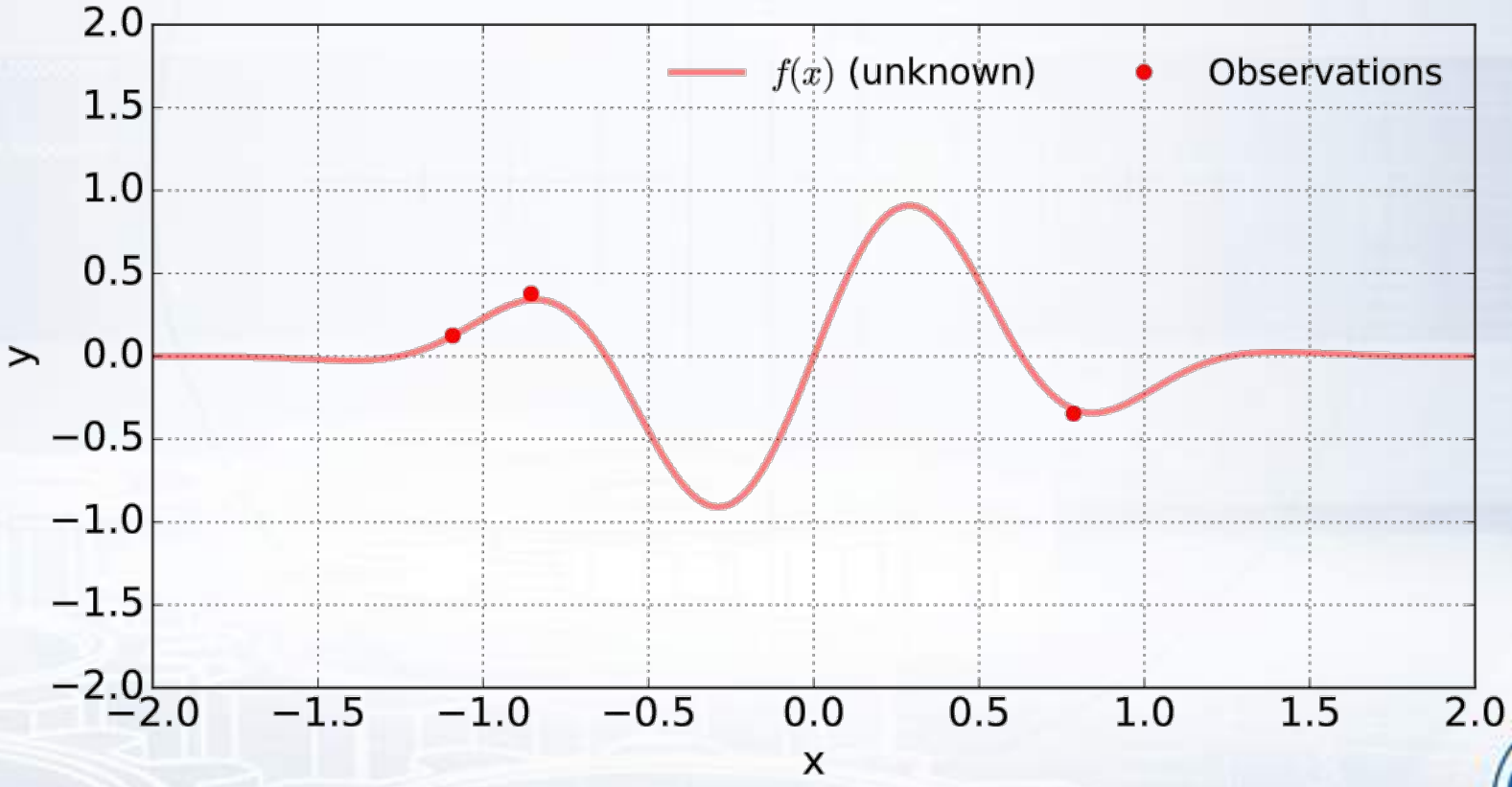
Optimization example

Consider the following objective function $f(x)$. The goal is to find its minimum.



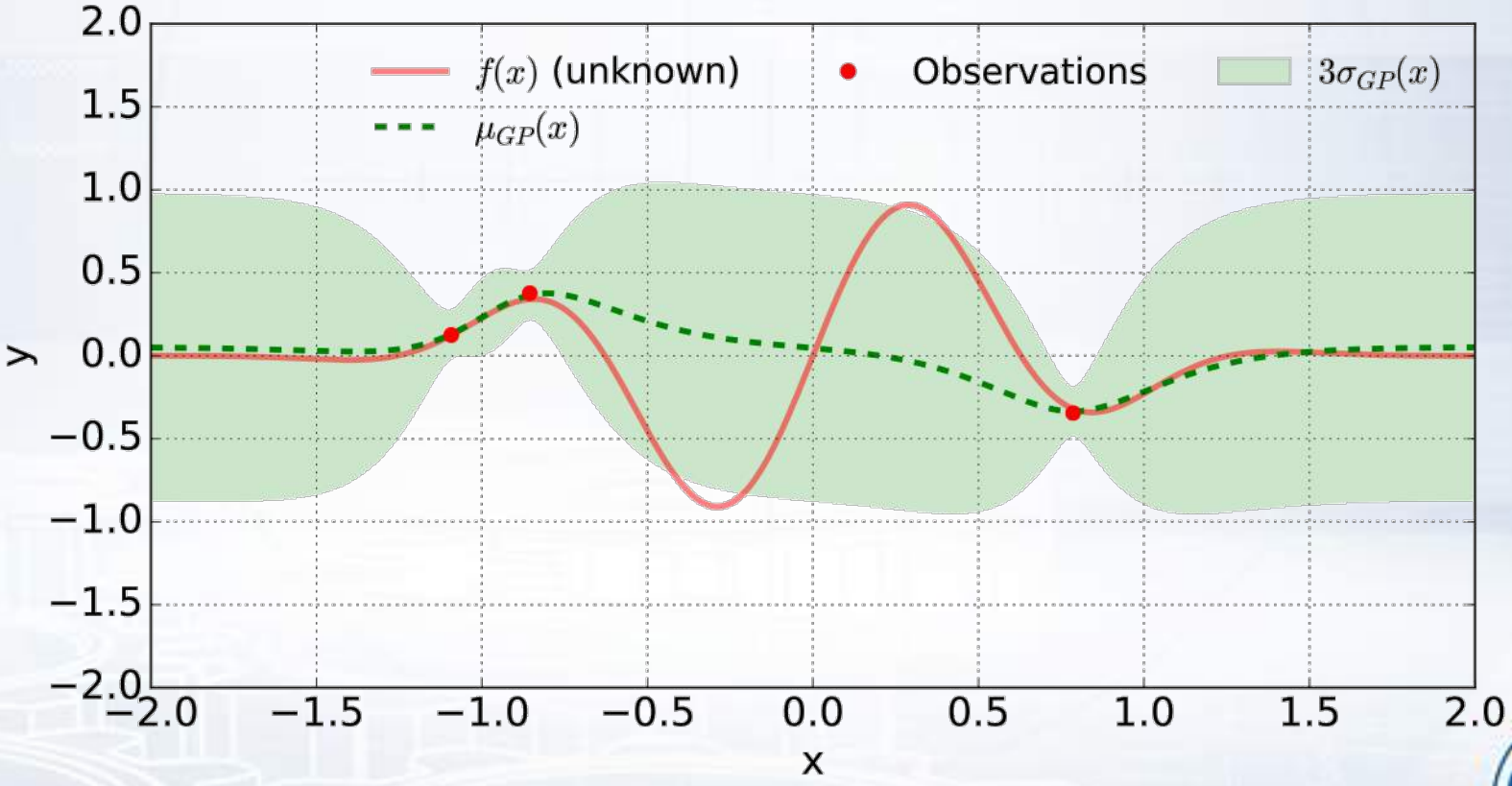
Optimization example

Lets start the optimization from three observations:



Optimization example

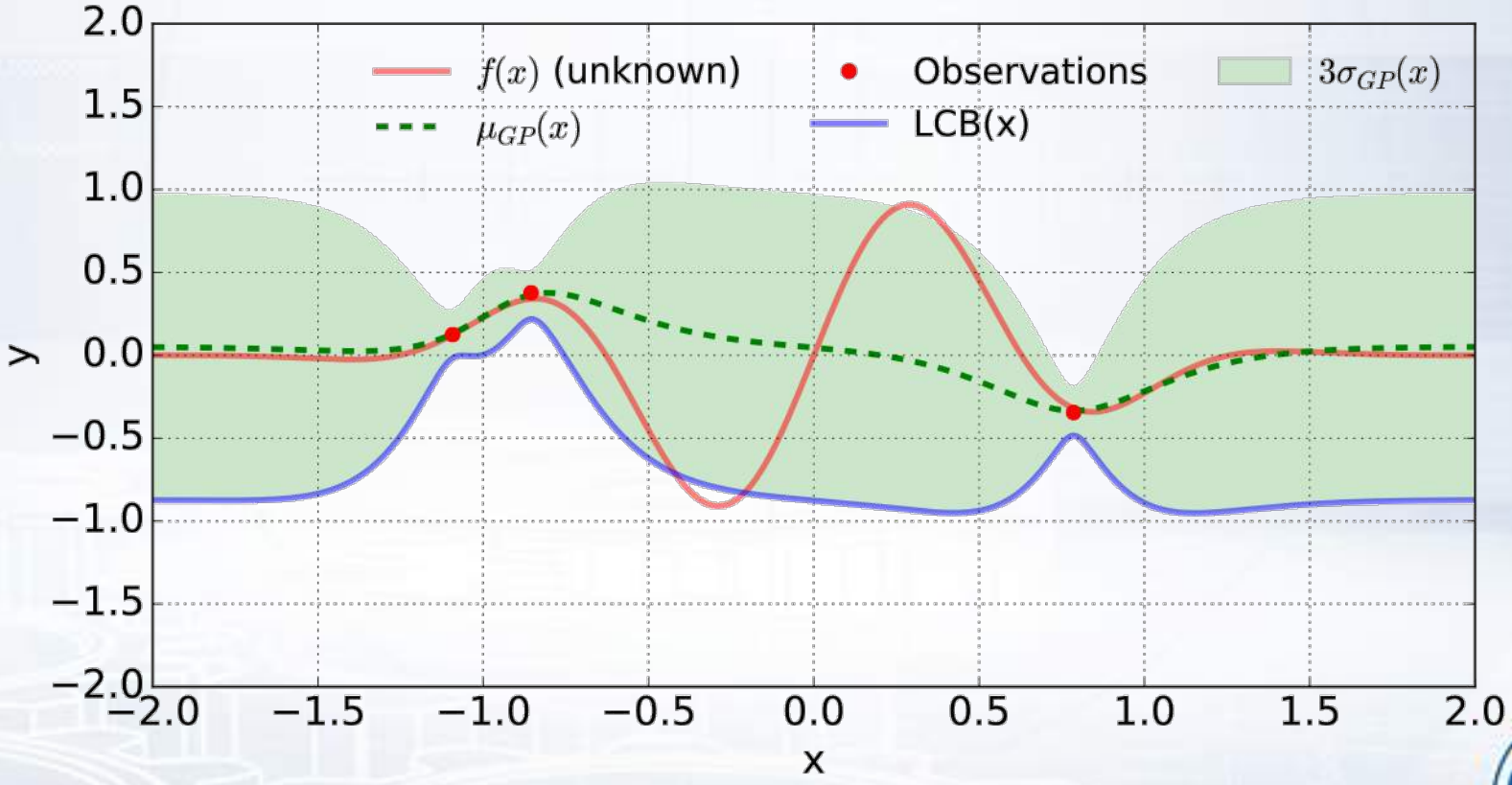
At the first step of the Bayesian optimization find approximation of the objective function $f(x)$ using Gaussian processes and known observations.



Optimization example

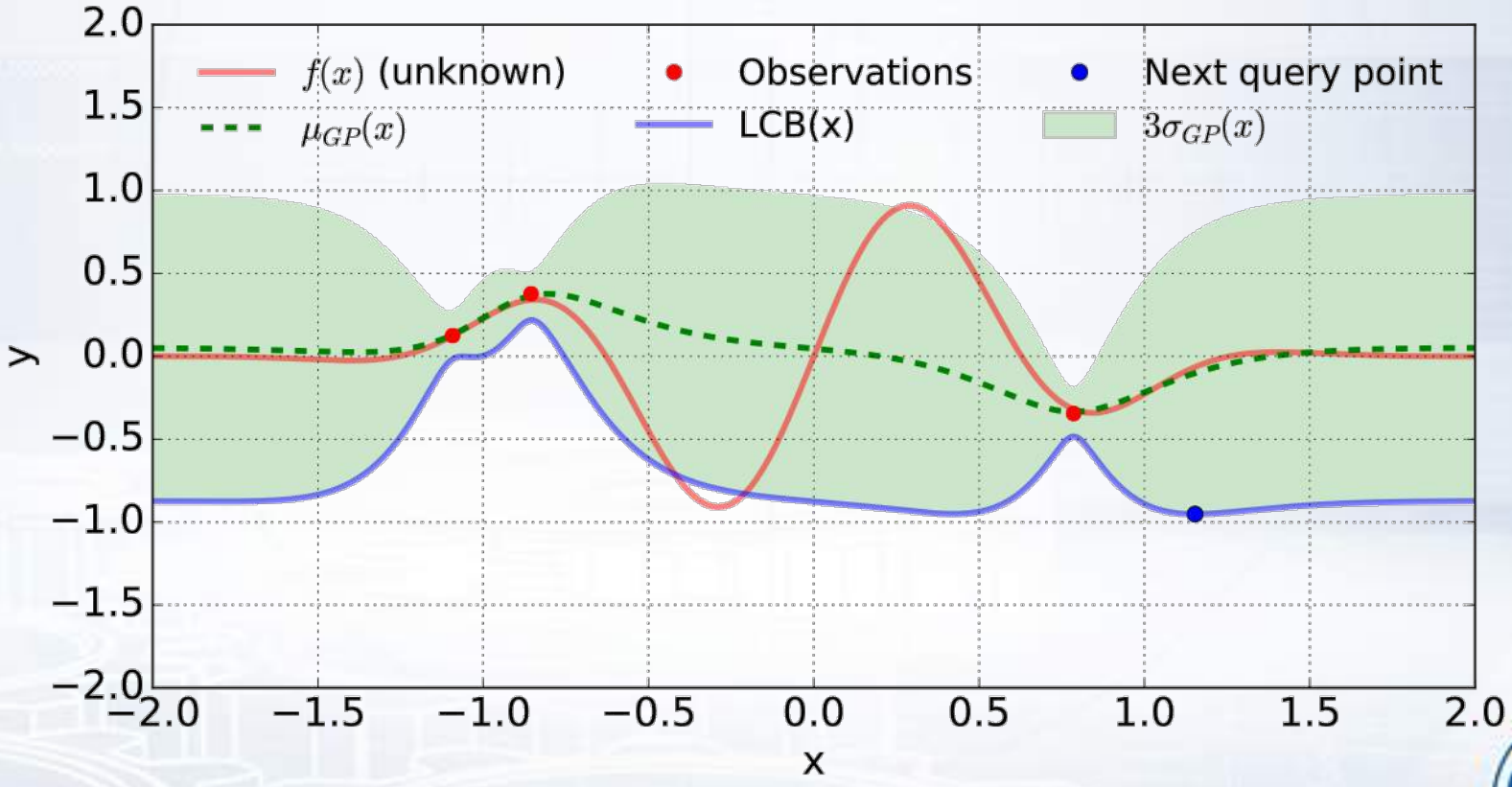
In this example the Lower Confidence Bound is used as the acquisition function:

$$LCB(x) = \mu_{GP}(x) - 3\sigma_{GP}(x)$$



Optimization example

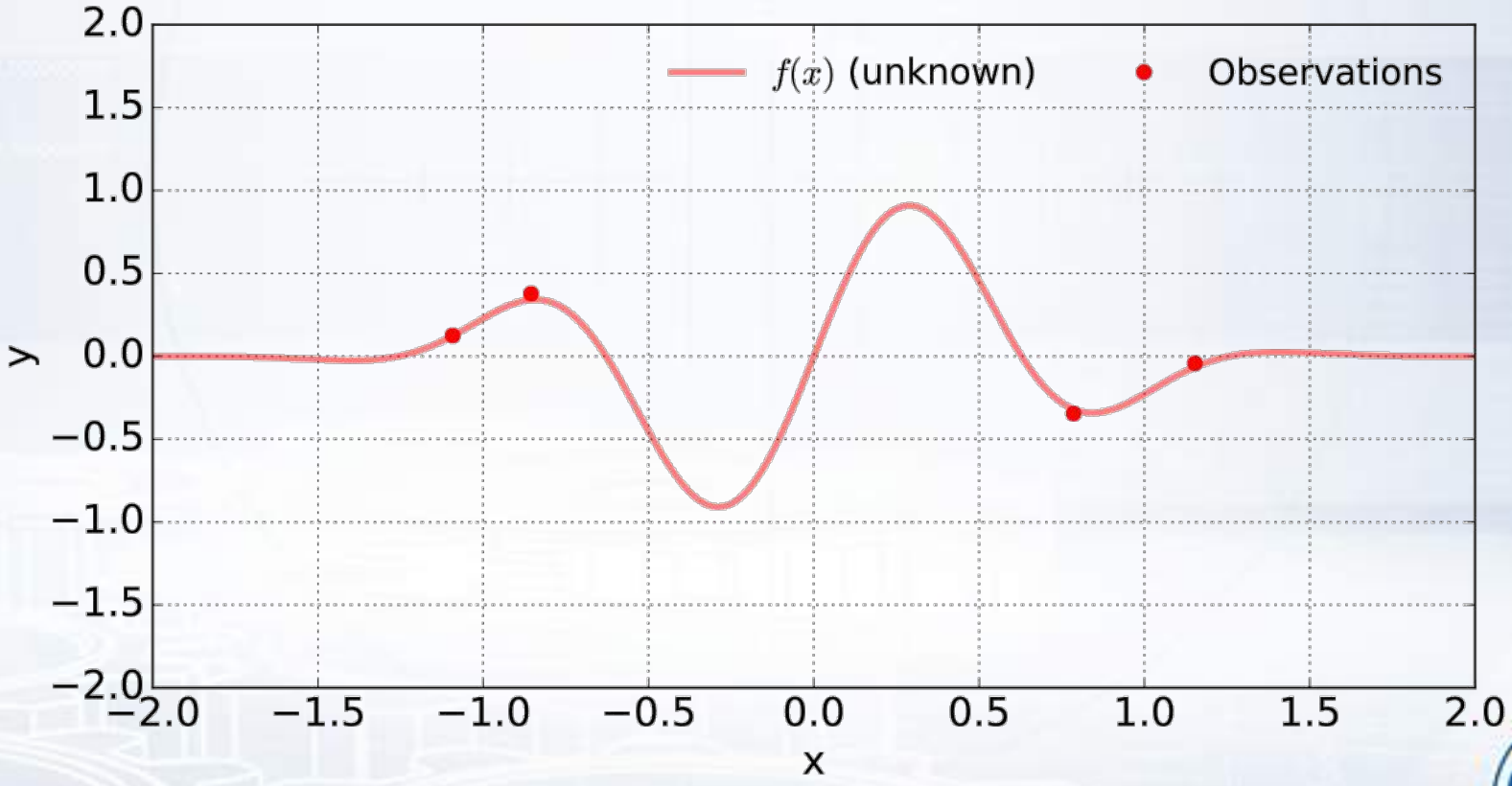
At the second step of the Bayesian optimization find minimum point x_4 of the acquisition function.



Optimization example

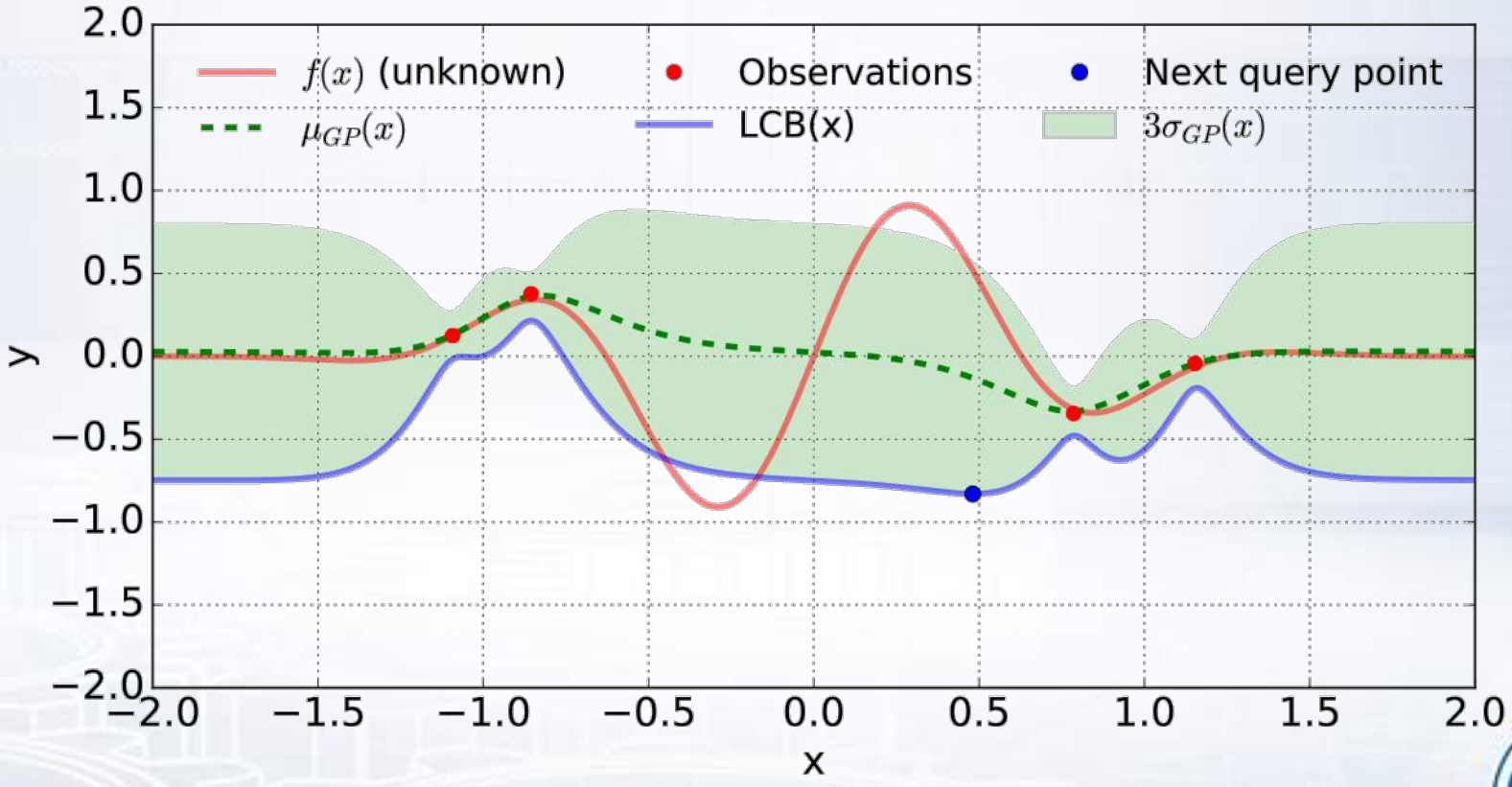
At the third step of the Bayesian optimization sample the objective function at found point x_4:

$$y_4 = f(x_4) + \varepsilon_4$$



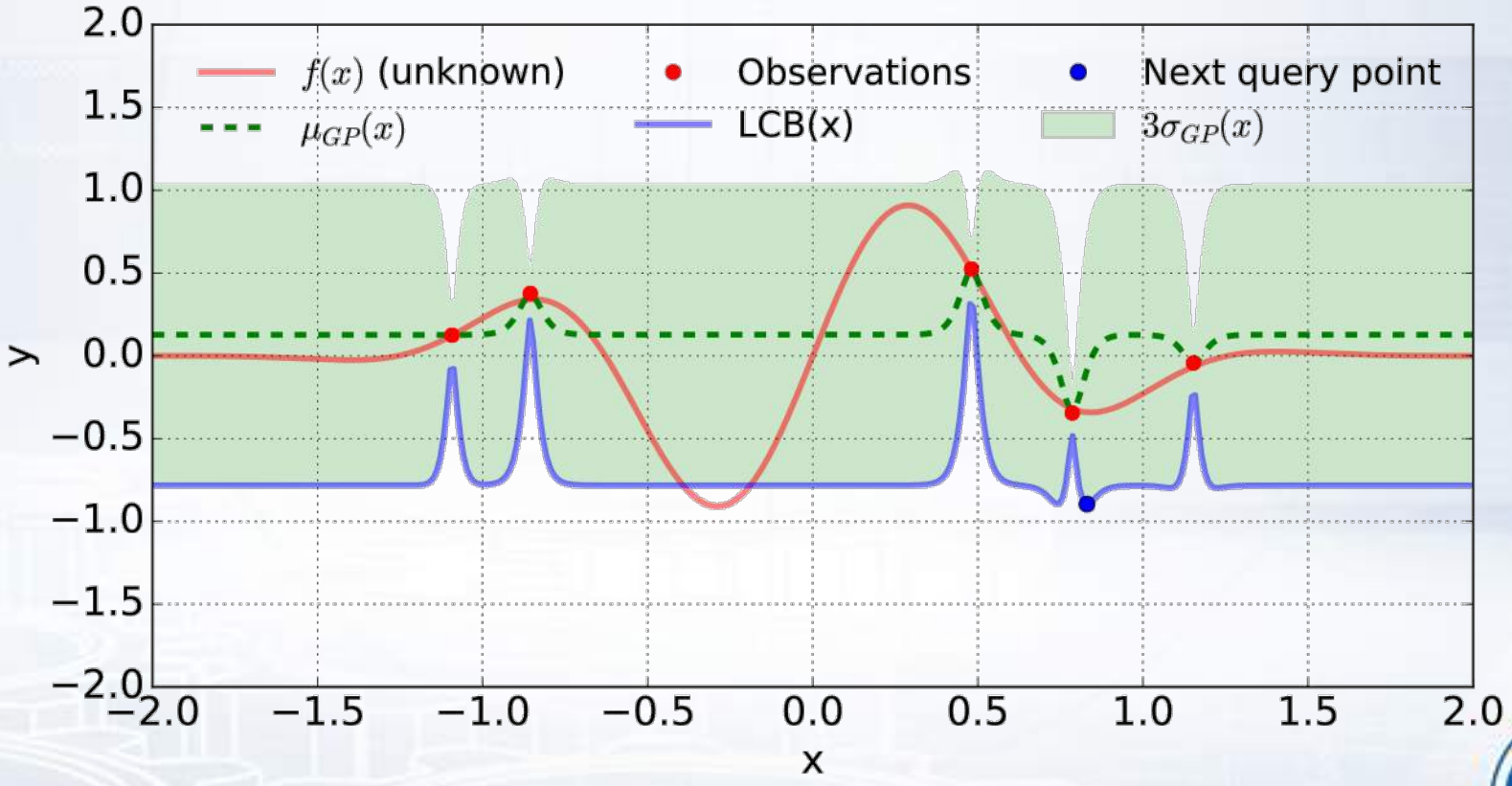
Optimization example

The 2nd iteration:



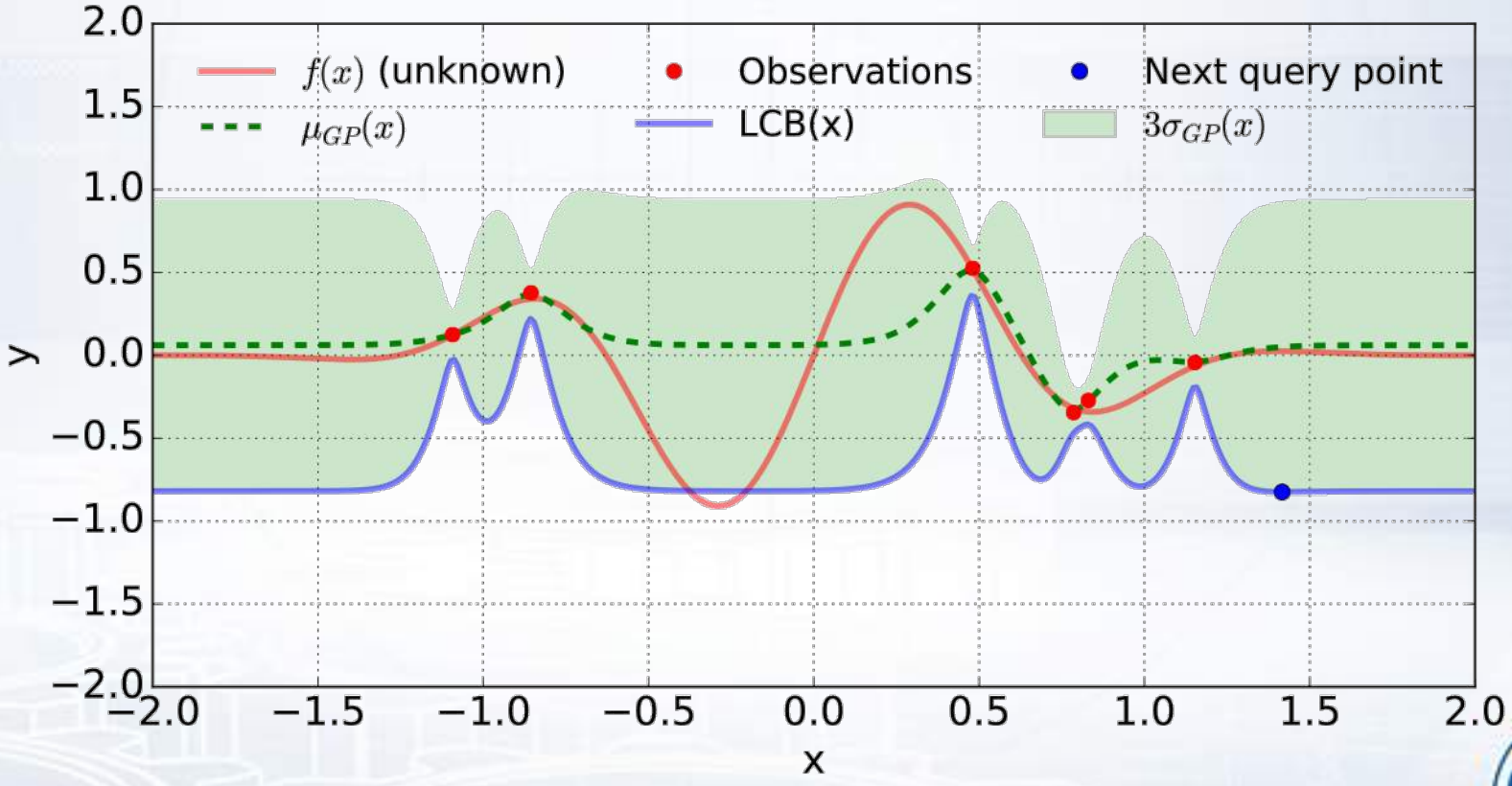
Optimization example

The 3rd iteration:



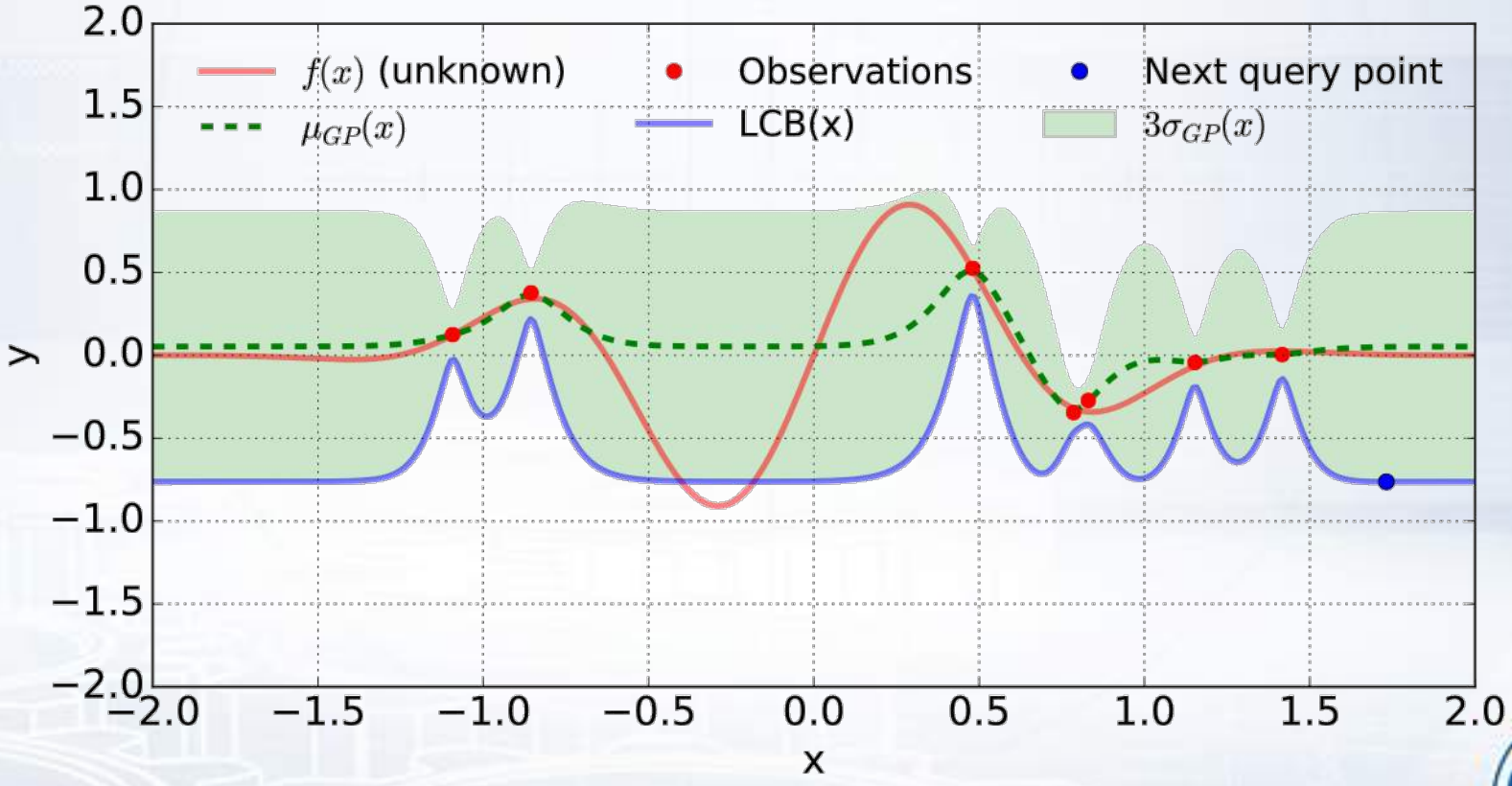
Optimization example

The 4th iteration:



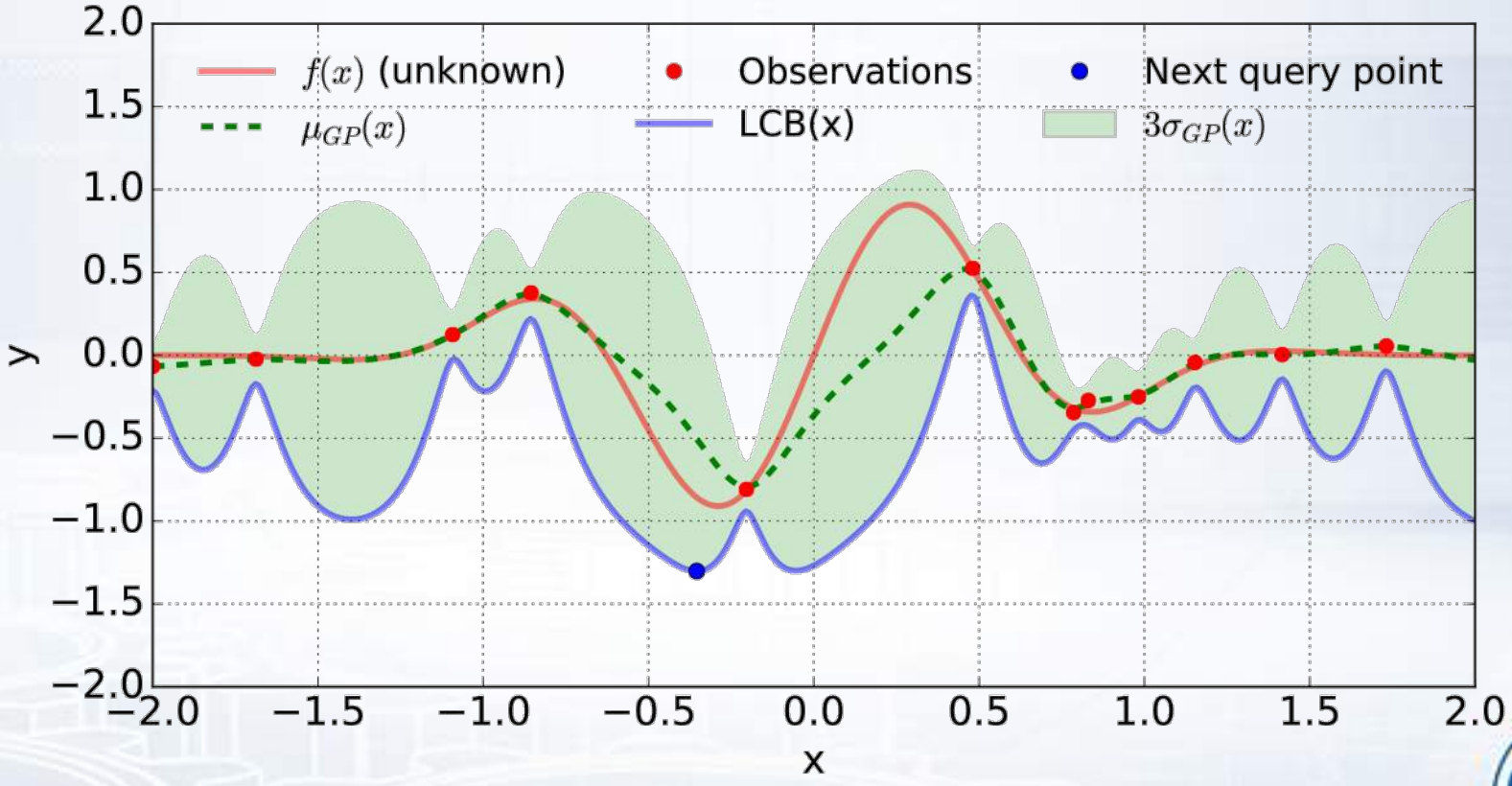
Optimization example

The 5th iteration:



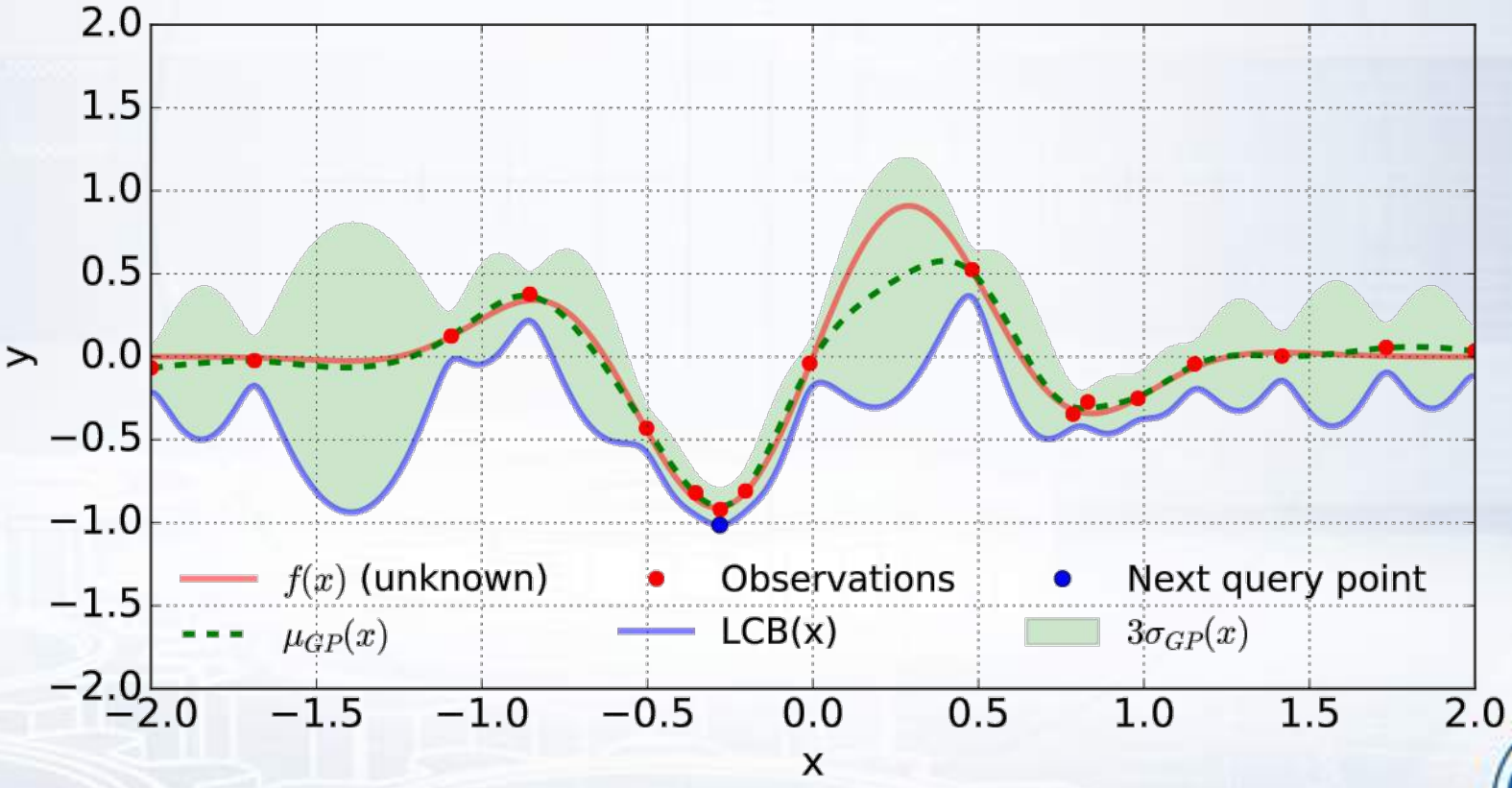
Optimization example

The 10th iteration:



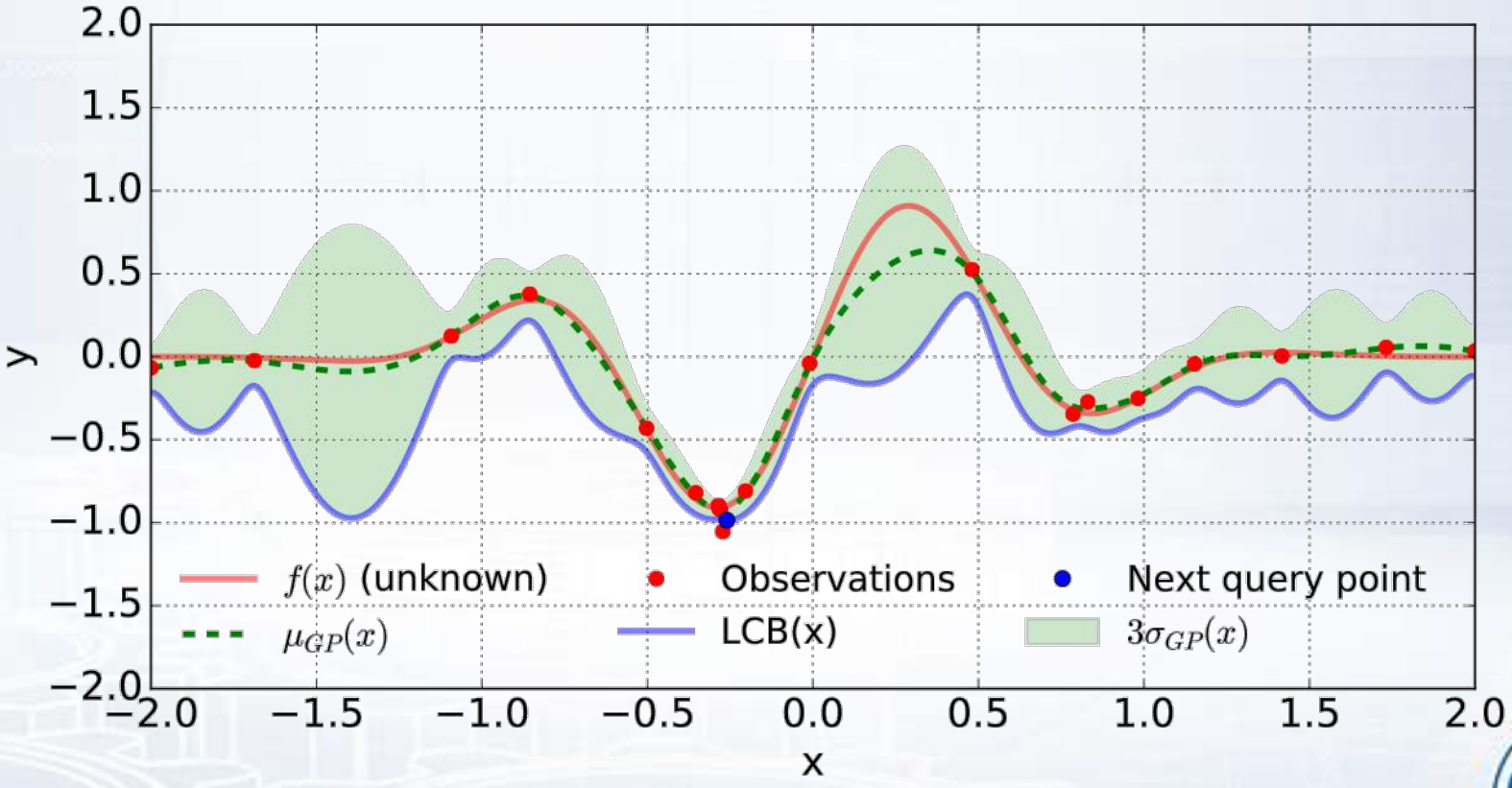
Optimization example

The 15th iteration:



Optimization example

The 20th iteration:



Bayesian optimization

Algorithm:

1. Find the objective function approximation using previously calculated values $\{x_i, y_i\}_{i=1}^{N}$ solving a regression problem.
2. Using the approximation find the optimum point of an *acquisition* function $u_N(x) : x_{N+1} = \mathrm{argmax}\ u_N(x)$
3. Sample the objective function: $y_{N+1} = f(x_{N+1}) + \varepsilon_{N+1}$
4. Repeat the steps.

Bayesian optimization
Part 2

Bayesian optimization

Bayesian optimization is a method of finding the optimum of expensive cost function.

This cost function is also called *objective* function and denoted as $f(x)$.

It supposed that calculation of $f(x)$ at one point is expensive.

The derivatives of the objective function are unknown.

The goal of the Bayesian optimization is to find the optimum of the objective function using as small number of the function calculations as possible.

Bayesian optimization

Algorithm:

1. Find the objective function approximation using previously calculated values $\{x_i, y_i\}_{i=1}^{N}$ and solving a regression problem.
2. Using the approximation find optimum point of an *acquisition* function $u_N(x) : x_{N+1} = \mathrm{argmax}_x u_N(x)$
3. Sample the objective function: $y_{N+1} = f(x_{N+1}) + \varepsilon_{N+1}$
4. Repeat the steps.

Bayesian optimization

Algorithm:

1. Find the objective function approximation using previously calculated values $\{x_i, y_i\}_{i=1}^{N}$ solving a regression problem.
2. Using the approximation find the optimum point of an *acquisition* function $u_N(x) : x_{N+1} = \mathrm{argmax}\ u_N(x)$
3. Sample the objective function: $y_{N+1} = f(x_{N+1}) + \varepsilon_{N+1}$
4. Repeat the steps.

Acquisition function

There are variety of acquisition functions. One of them is Lower Confidence Bound (LCB) for the objective function minimization:

$$LCB(x) = \mu_x - k\sigma(x)$$

where

- $\mu(x)$ is mean value of the approximation of the objective function,
- $\sigma(x)$ is standard deviation of the approximation,
- k is adjustable parameter.

Exploration-exploitation trade-off

The Bayesian optimization with Gaussian processes allows to balance between exploration and exploitation of the objective function.

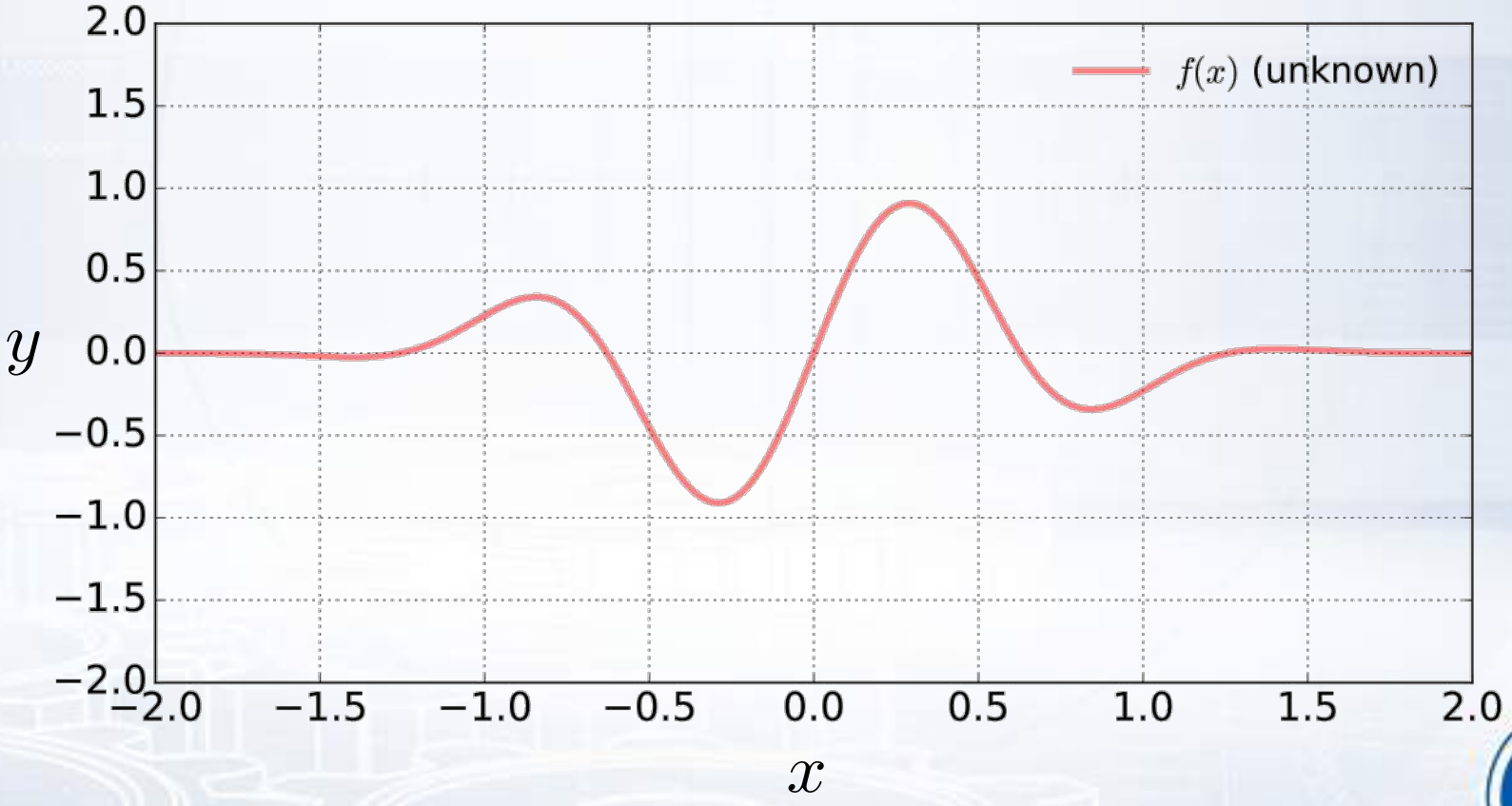
Exploration: choose a point with high variance at each iteration of the optimization.

Exploitation: choose a point with high/low mean at each iteration of the optimization.

Adjustable parameters of the acquisition function provide trade-off between exploration and exploitation.

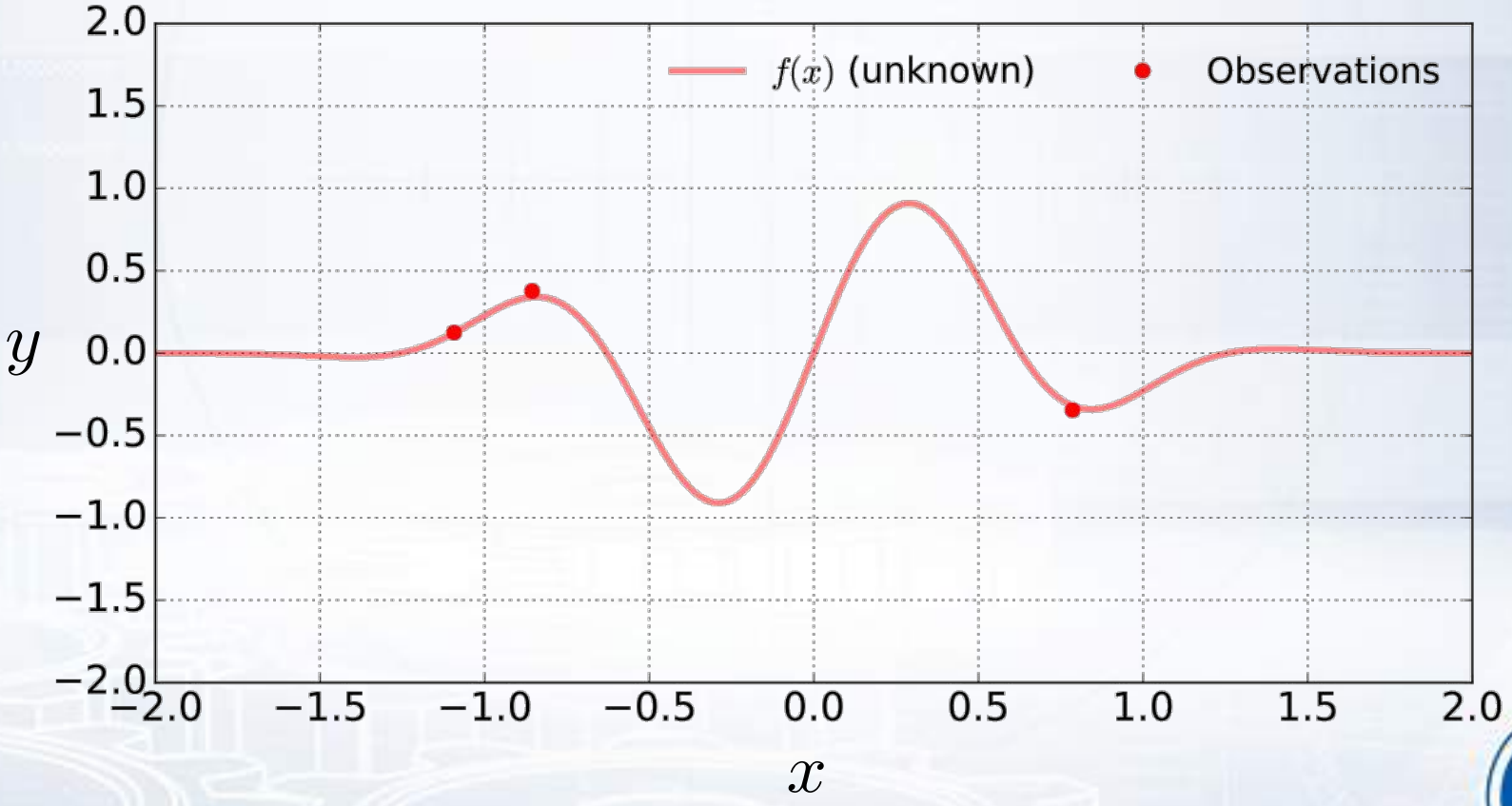
Optimization example

Consider the following objective function $f(x)$. The goal is to find its minimum.



Optimization example

Lets start the optimization from three observations:



Exploitation

Exploitation can be achieved with small values of k in $LCB(x)$:

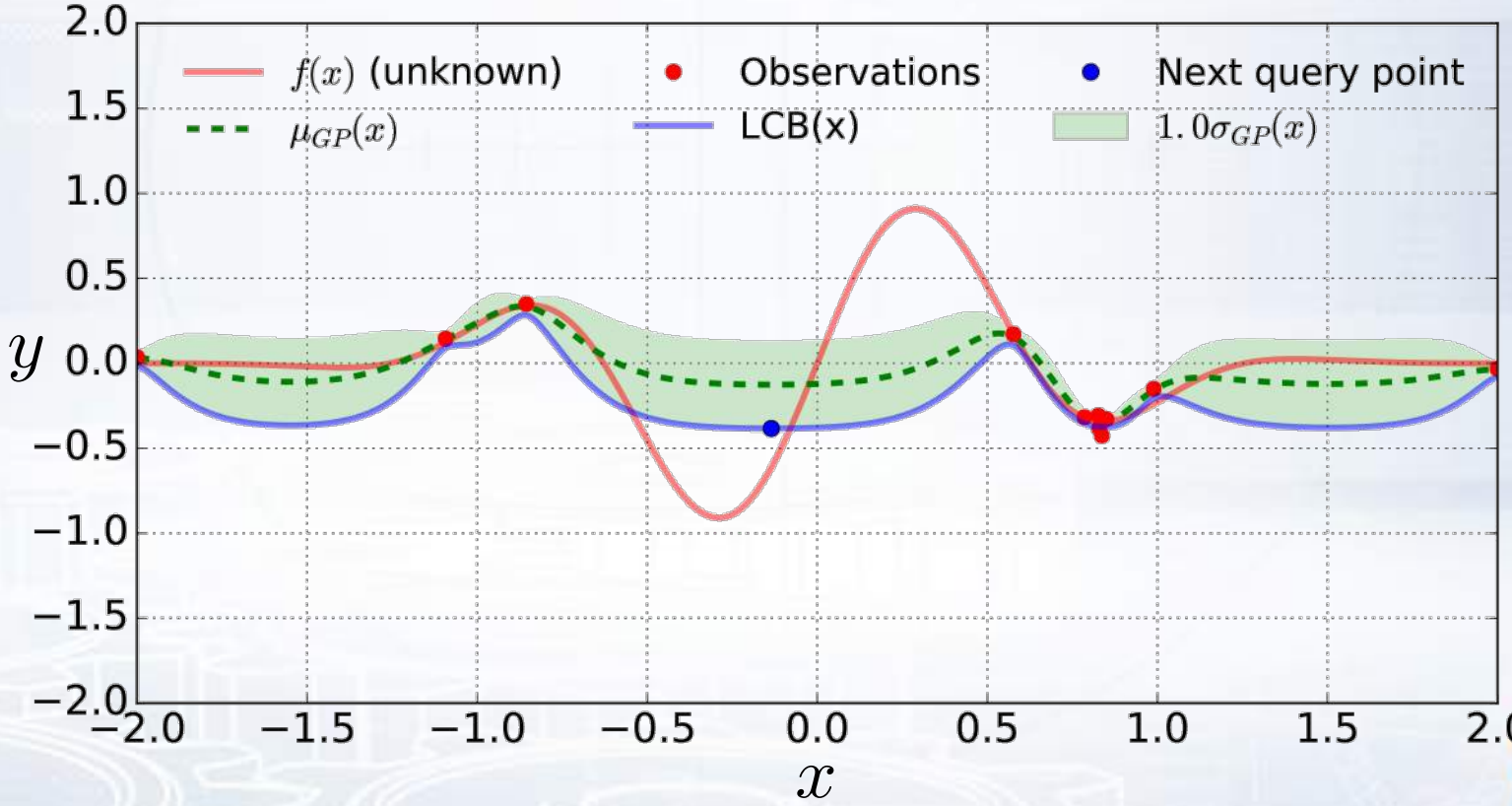
$$LCB(x) = \mu_{GP}(x) - 1.0 * \sigma_{GP}(x)$$

Exploitation

Exploitation can be achieved with small values of k in $LCB(x)$:

$$LCB(x) = \mu_{GP}(x) - 1.0 * \sigma_{GP}(x)$$

The 10th iteration:

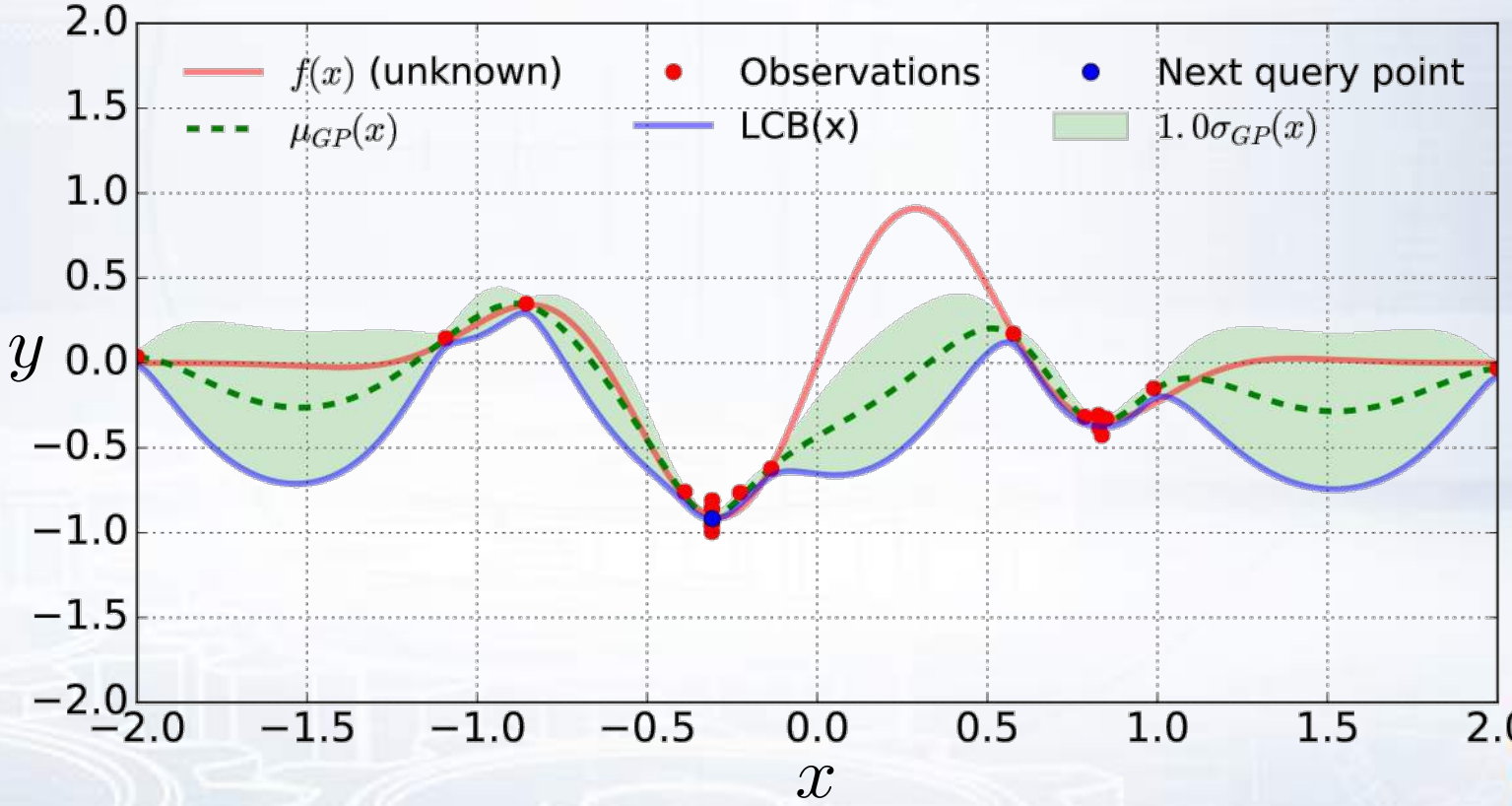


Exploitation

Exploitation can be achieved with small values of k in $LCB(x)$:

$$LCB(x) = \mu_{GP}(x) - 1.0 * \sigma_{GP}(x)$$

The 20th iteration:



Exploitation

- It takes a new point at each iteration close to found optimum of the objective function.
- There is no guarantee that this optimum is global.
- Other regions of the objective function are not explored.
- It needs less iterations than for exploration to find the optimum.

Exploration

Exploitation can be achieved with large values of k in $LCB(x)$:

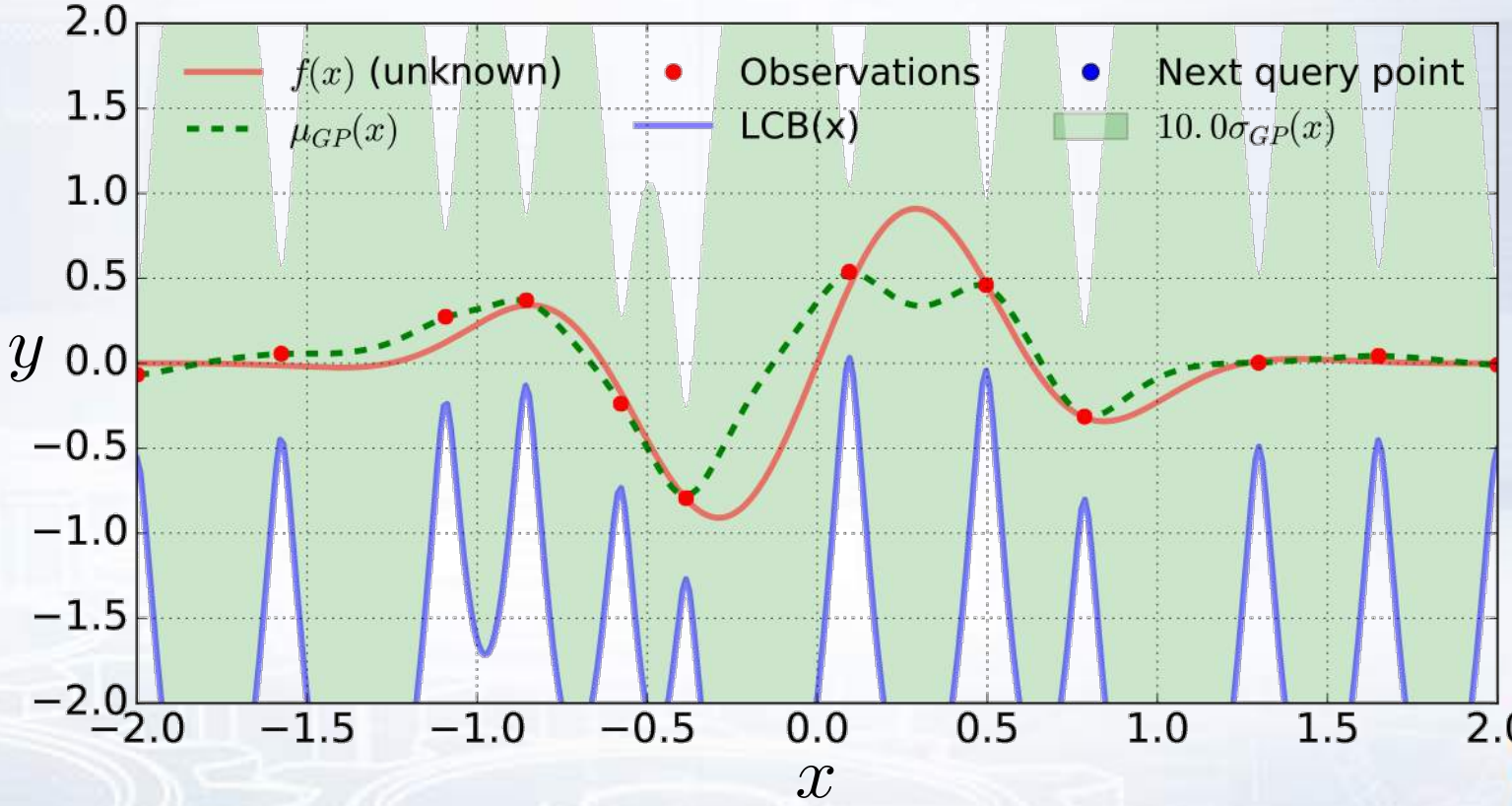
$$LCB(x) = \mu_{GP}(x) - 10.0 * \sigma_{GP}(x)$$

Exploration

Exploitation can be achieved with large values of k in $LCB(x)$:

$$LCB(x) = \mu_{GP}(x) - 10.0 * \sigma_{GP}(x)$$

The 10th iteration:

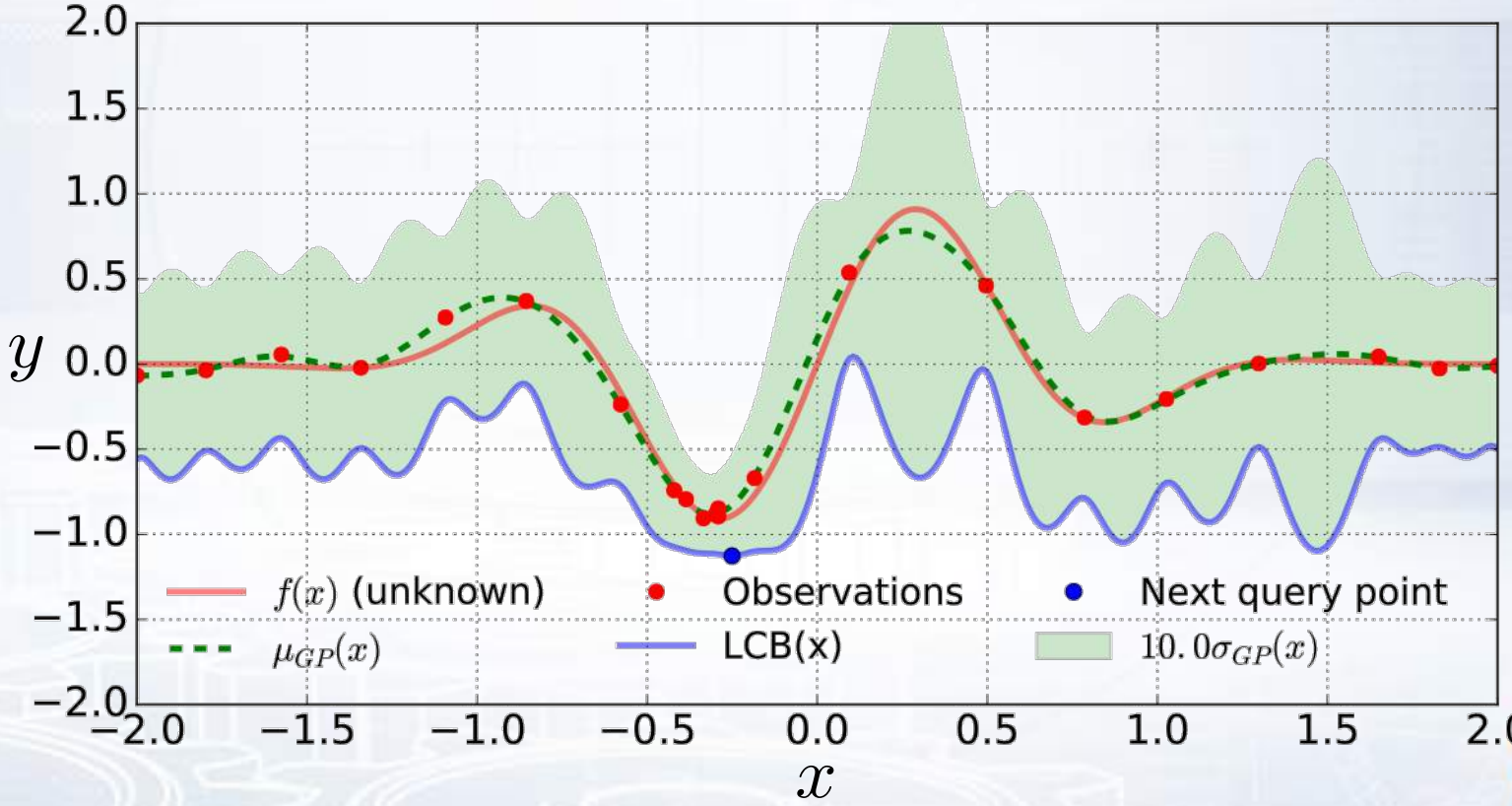


Exploration

Exploitation can be achieved with large values of k in $LCB(x)$:

$$LCB(x) = \mu_{GP}(x) - 10.0 * \sigma_{GP}(x)$$

The 20th iteration:

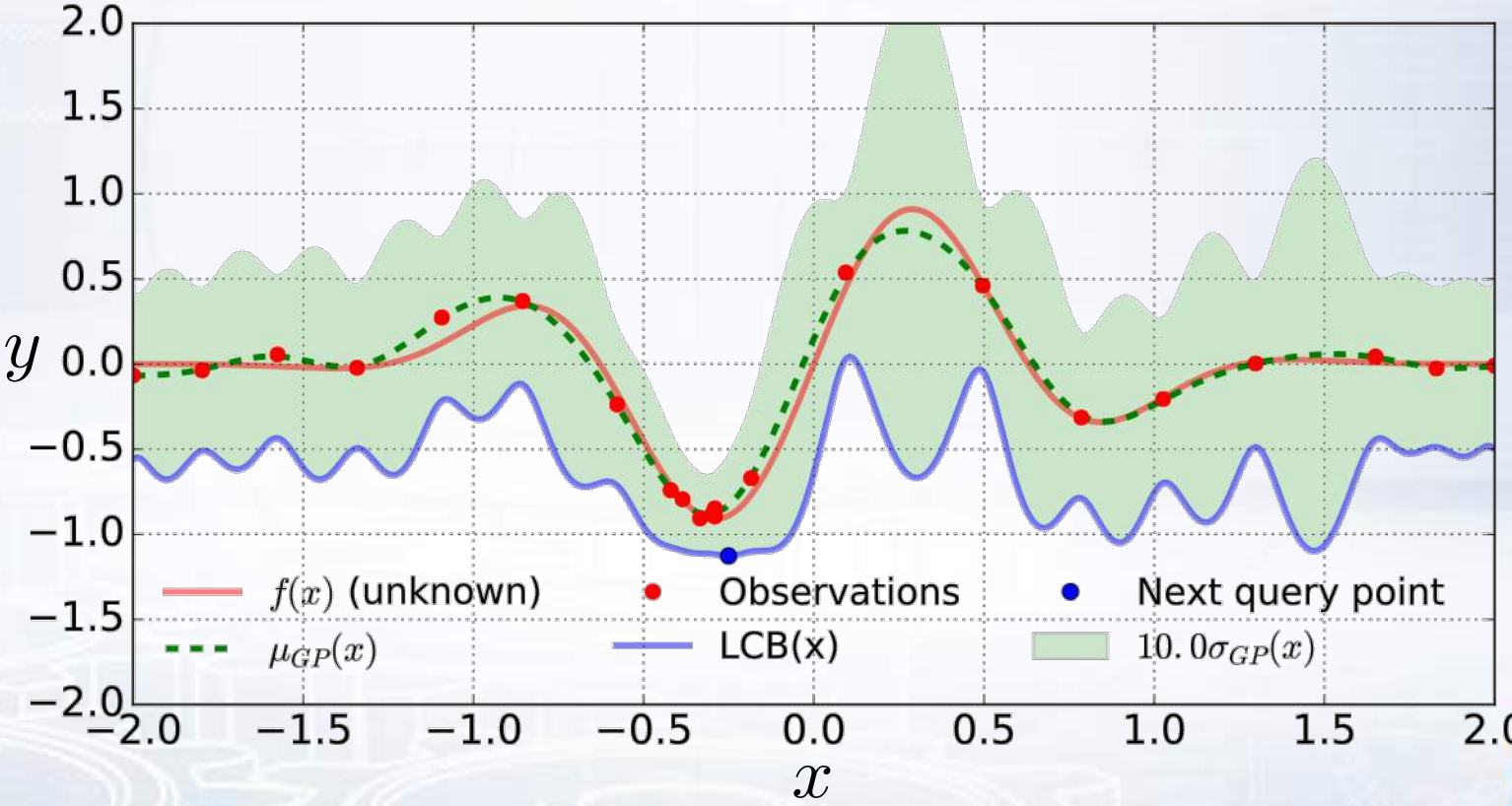


Exploration

- It takes a new point at each iteration where the variance of the approximation of the objective function is large.
- All regions of the objective function are explored.
- It is more likely that the found optimum is global.
- It needs more iterations than for exploitation to find the optimum.

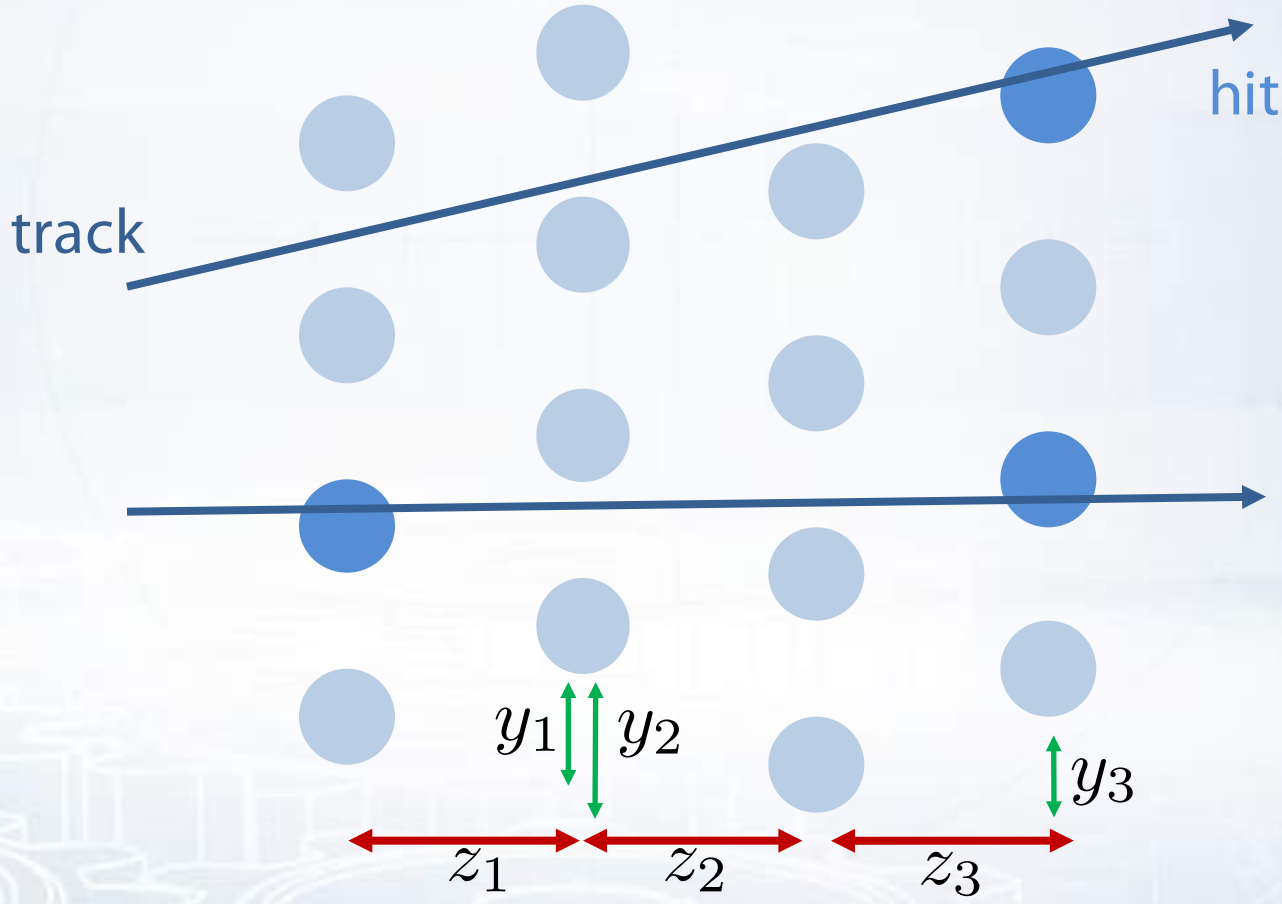
Bayesian optimization

Bayesian optimization is a method of finding the optimum of expensive cost function.



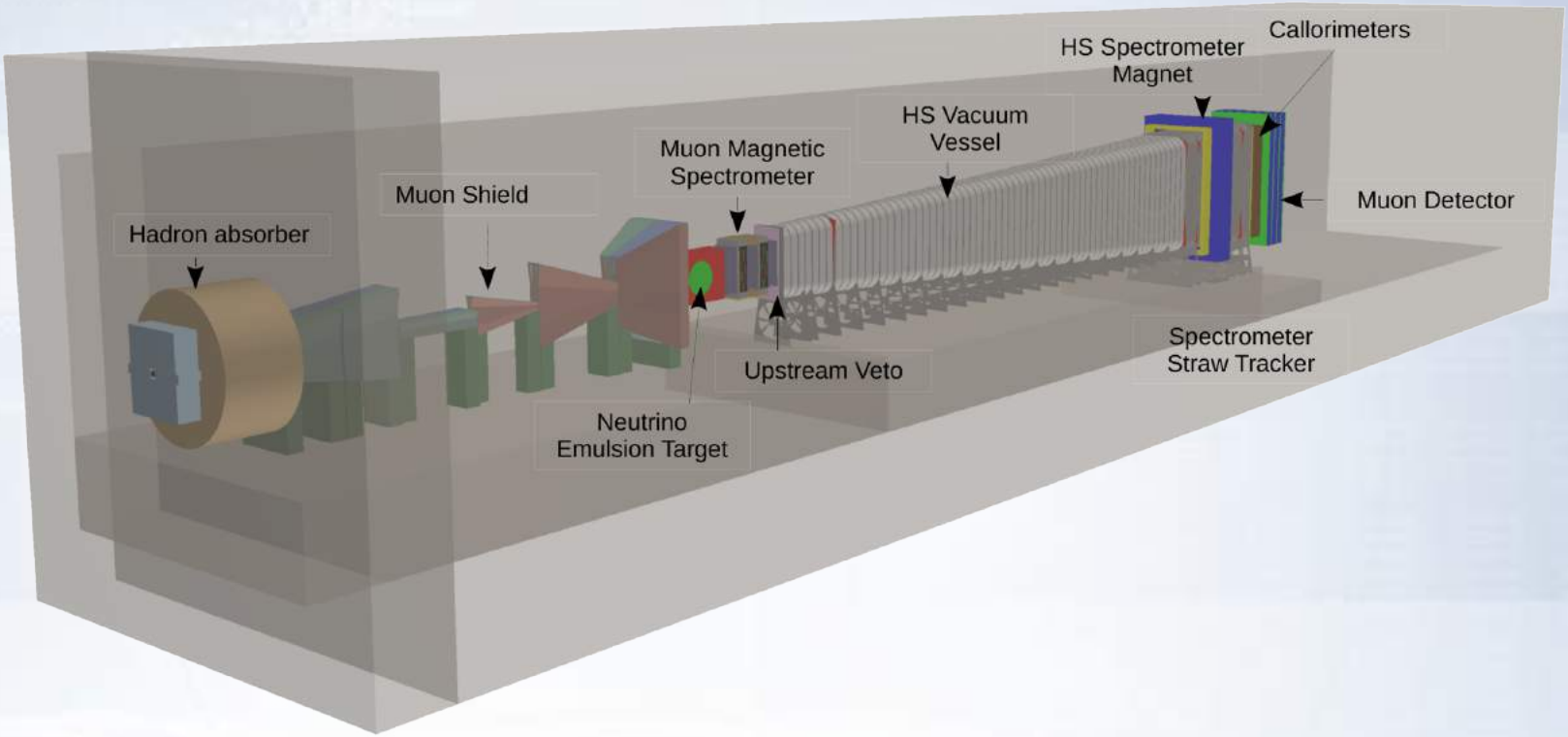
Detector optimization

In high energy physics Bayesian optimization with Gaussian processes can be used for detector design optimization.



Optimization examples

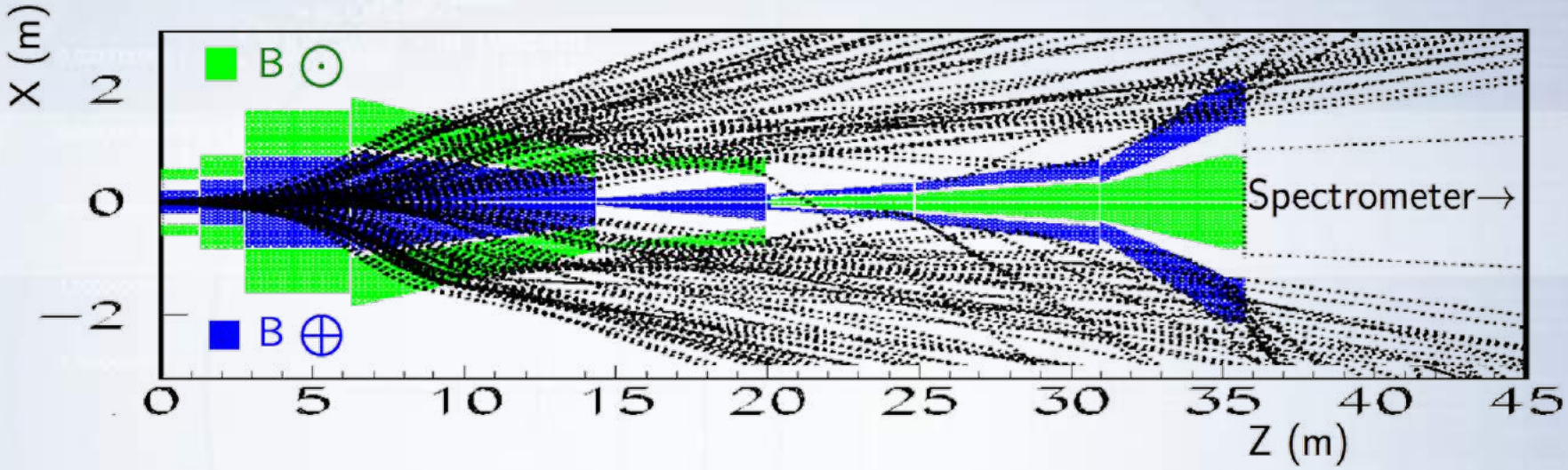
SHiP muon shield optimization



The SHiP experiment is a new general purpose fixed target facility proposed at the CERN SPS accelerator to search for new physics in the largely unexplored domain of very weakly interacting particles.

Physik-Institut, http://www.physik.uzh.ch/en/researcharea/ship.html

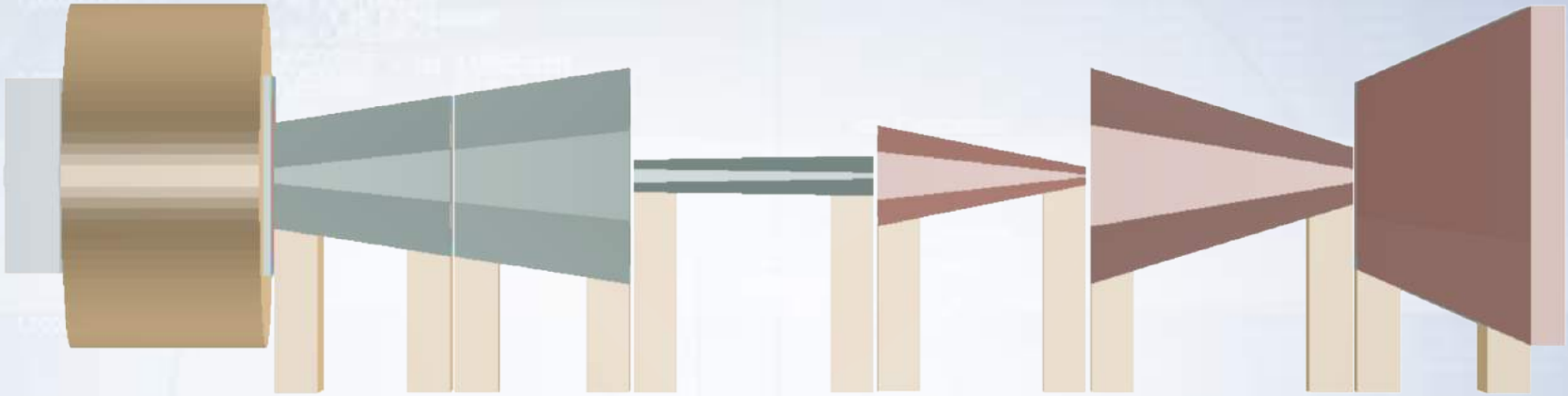
SHiP muon shield optimization



The muon shield is a critical component of the SHiP experiment, which deflects the high flux of muons produced in the target, that would represent a very serious background for the particle searches, away from the detector.

Baranov A. et. al., Optimising the Active Muon Shield for the SHiP Experiment at CERN, IOP Conf. Series: Journal of Physics: Conf. Series 934 (2017) 012050 doi :10.1088/1742-6596/934/1/012050

SHiP muon shield optimization

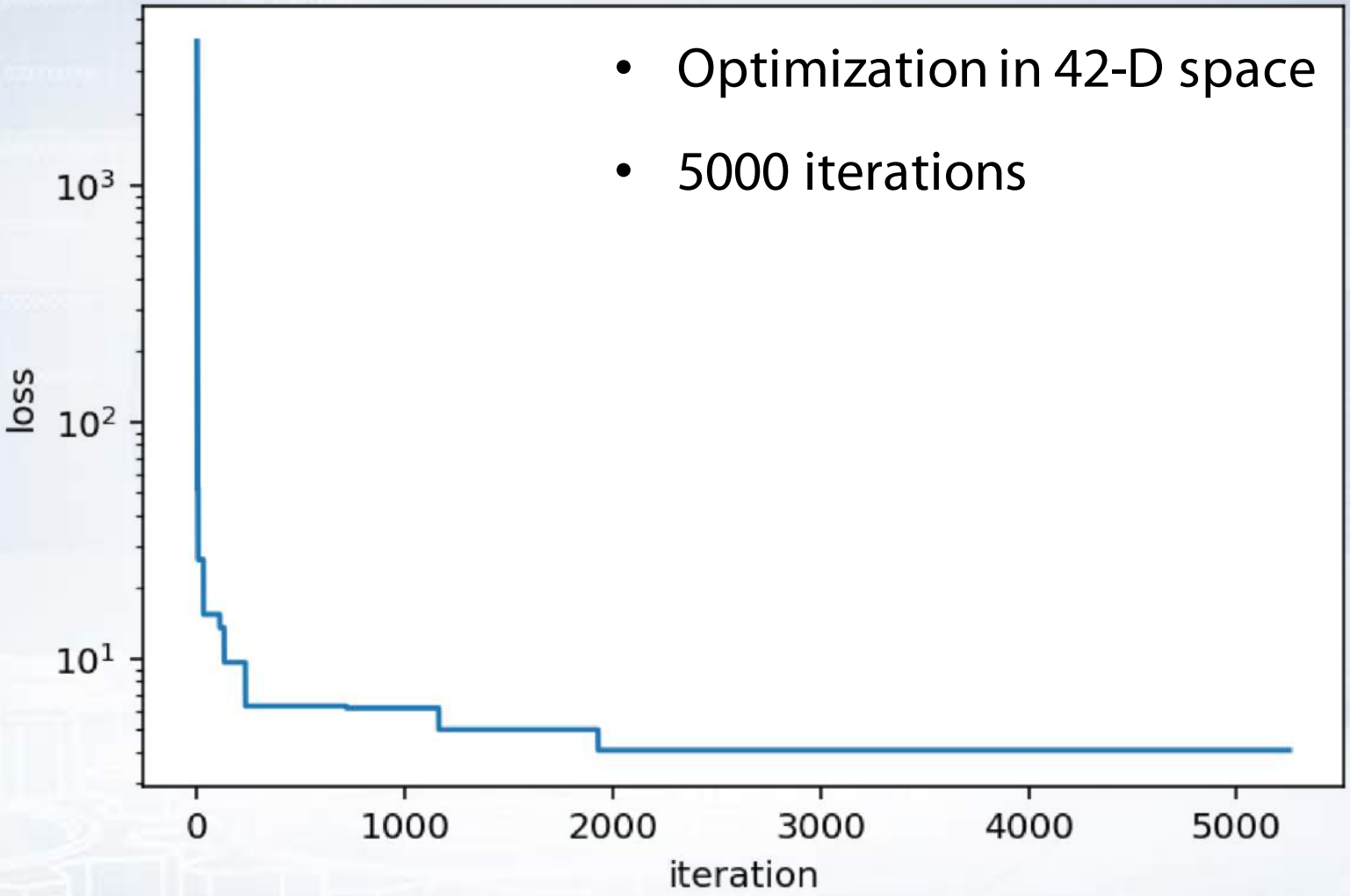


The shield consists of eight magnets and each magnet is parameterized by seven values: length, width, etc..

The loss function depends on the physical performance of the shield (muon background) and its weight.

Baranov A. et. al., Optimising the Active Muon Shield for the SHiP Experiment at CERN, IOP Conf. Series: Journal of Physics: Conf. Series 934 (2017) 012050 doi :10.1088/1742-6596/934/1/012050

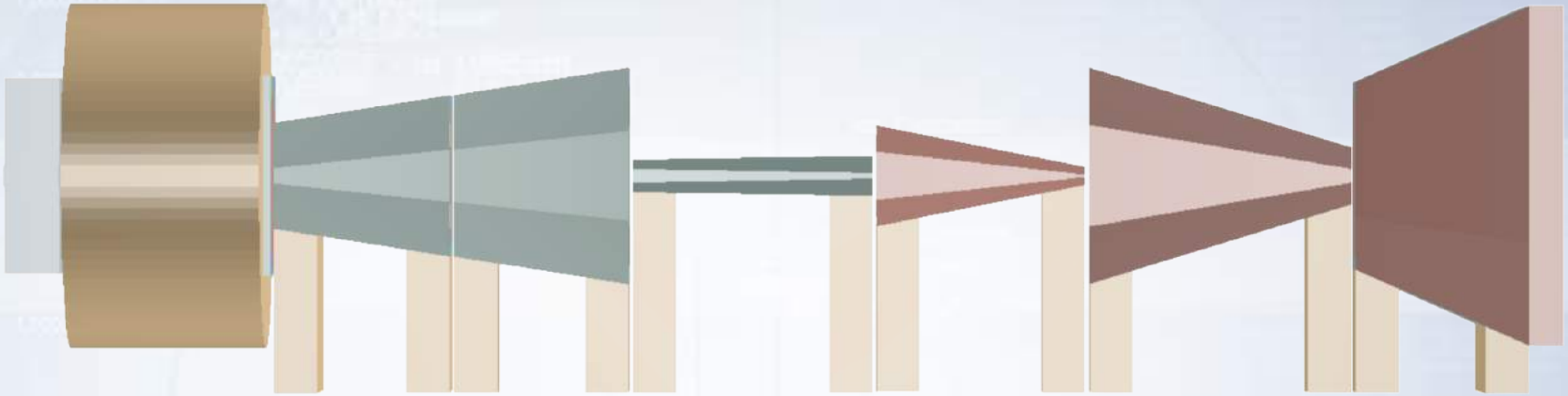
SHiP muon shield optimization



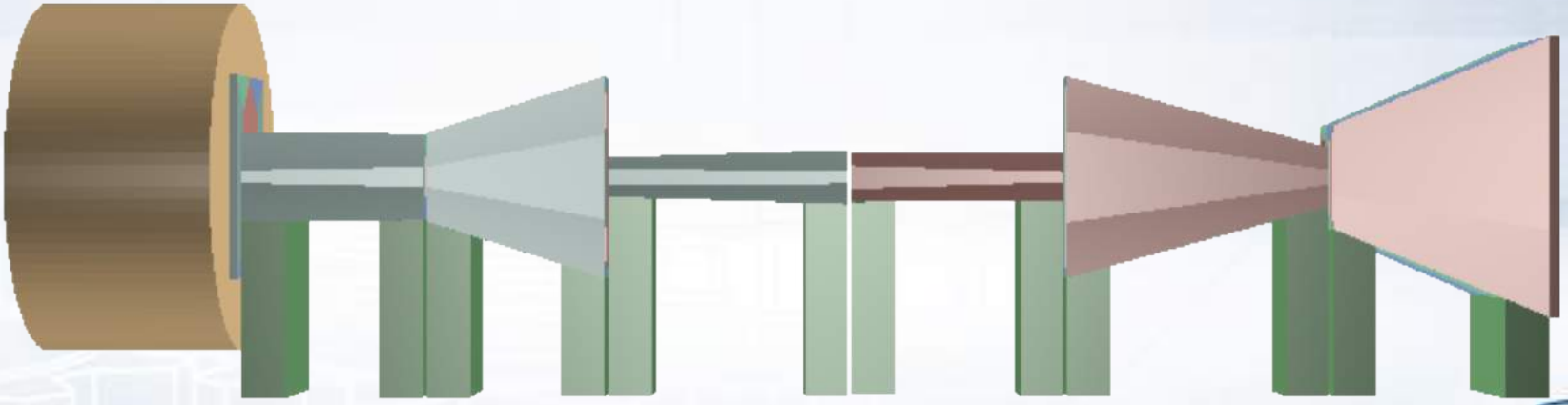
- Optimization in 42-D space
- 5000 iterations

Baranov A. et. al., Optimising the Active Muon Shield for the SHiP Experiment at CERN, IOP Conf. Series: Journal of Physics: Conf. Series 934 (2017) 012050 doi :10.1088/1742-6596/934/1/012050

SHiP muon shield optimization



The new muon shield design is 25% lighter



Baranov A. et. al., Optimising the Active Muon Shield for the SHiP Experiment at CERN, IOP Conf. Series: Journal of Physics: Conf. Series 934 (2017) 012050 doi :10.1088/1742-6596/934/1/012050

Collisions simulation optimization

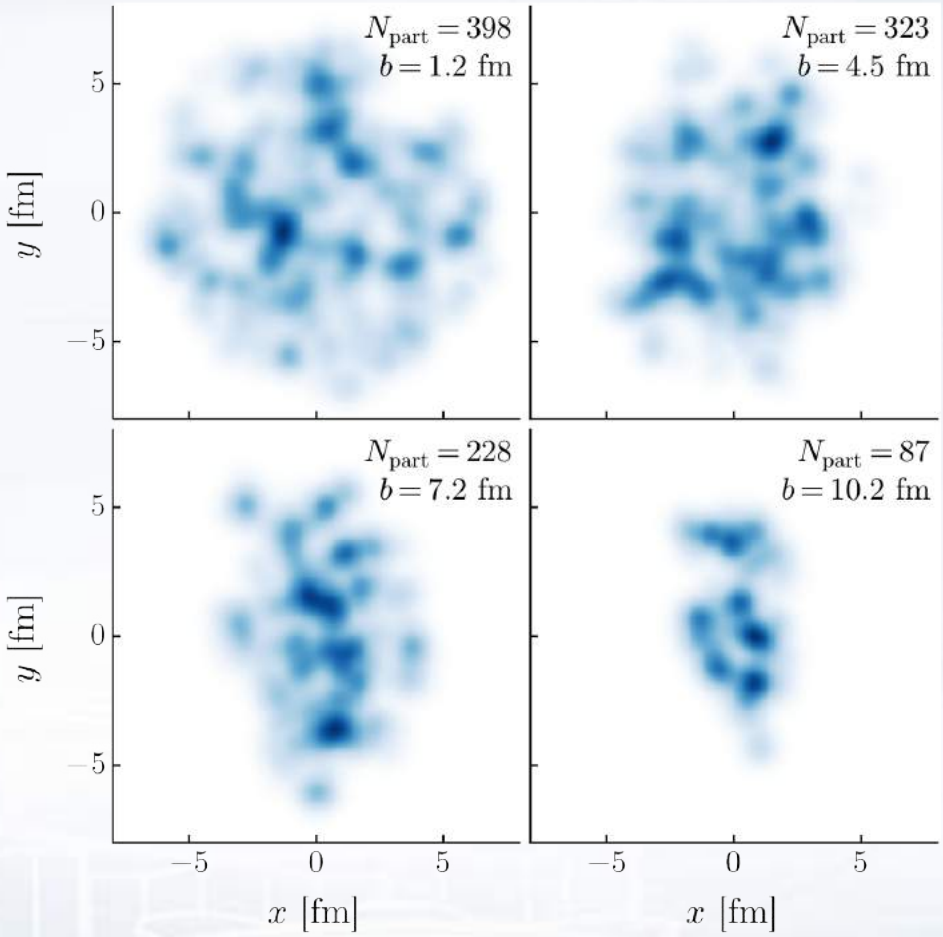
Bayesian methods are used to optimize parameters of heavy-ion collisions simulation.

Evaluating a simulation model for a single set of parameters requires thousands of individual event simulations, so direct optimization techniques quickly become intractable.

Jonah E. Bernhard, et. al., Applying Bayesian parameter estimation to relativistic heavy-ion collisions: simultaneous characterization of the initial state and quark-gluon plasma medium, Phys. Rev. C 94, 024907 (2016)

Collisions simulation optimization

Simulated examples of entropy density in the transverse plane for several typical Pb + Pb events:



Jonah E. Bernhard, et. al., Applying Bayesian parameter estimation to relativistic heavy-ion collisions: simultaneous characterization of the initial state and quark-gluon plasma medium, Phys. Rev. C 94, 024907 (2016)

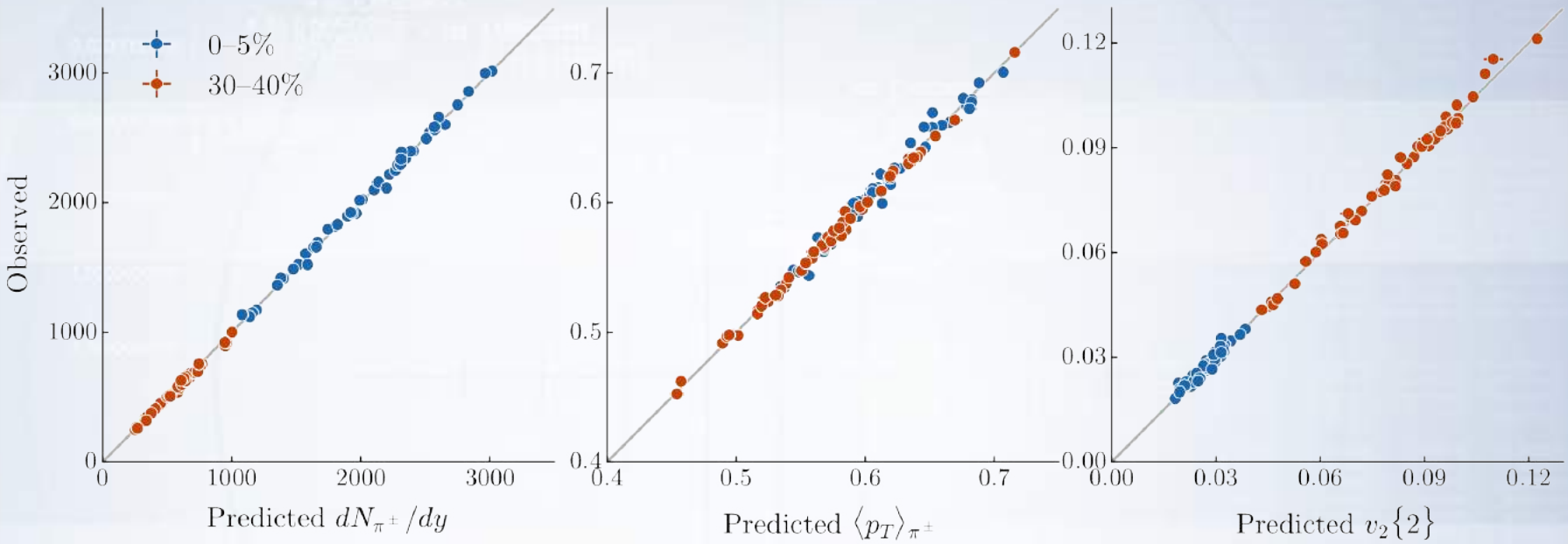
Collisions simulation optimization

Parameter	Description	Range
Norm	Overall normalization	100–250
p	Entropy deposition parameter	-1 to $+1$
k	Multiplicity fluct. shape	0.8–2.2
w	Gaussian nucleon width	0.4–1.0 fm
η/s hrg	Const. shear viscosity, $T < T_c$	0.3–1.0
η/s min	Shear viscosity at T_c	0–0.3
η/s slope	Slope above T_c	0–2 GeV^{-1}
ζ/s norm	Prefactor for $(\zeta/s)(T)$	0–2
T_{switch}	Particlization temperature	135–165 MeV

300 initial points are generated in the nine-dimensional parameter space and executed about 10k Pb + Pb events at each of the 300 points.

Jonah E. Bernhard, et. al., Applying Bayesian parameter estimation to relativistic heavy-ion collisions: simultaneous characterization of the initial state and quark-gluon plasma medium, Phys. Rev. C 94, 024907 (2016)

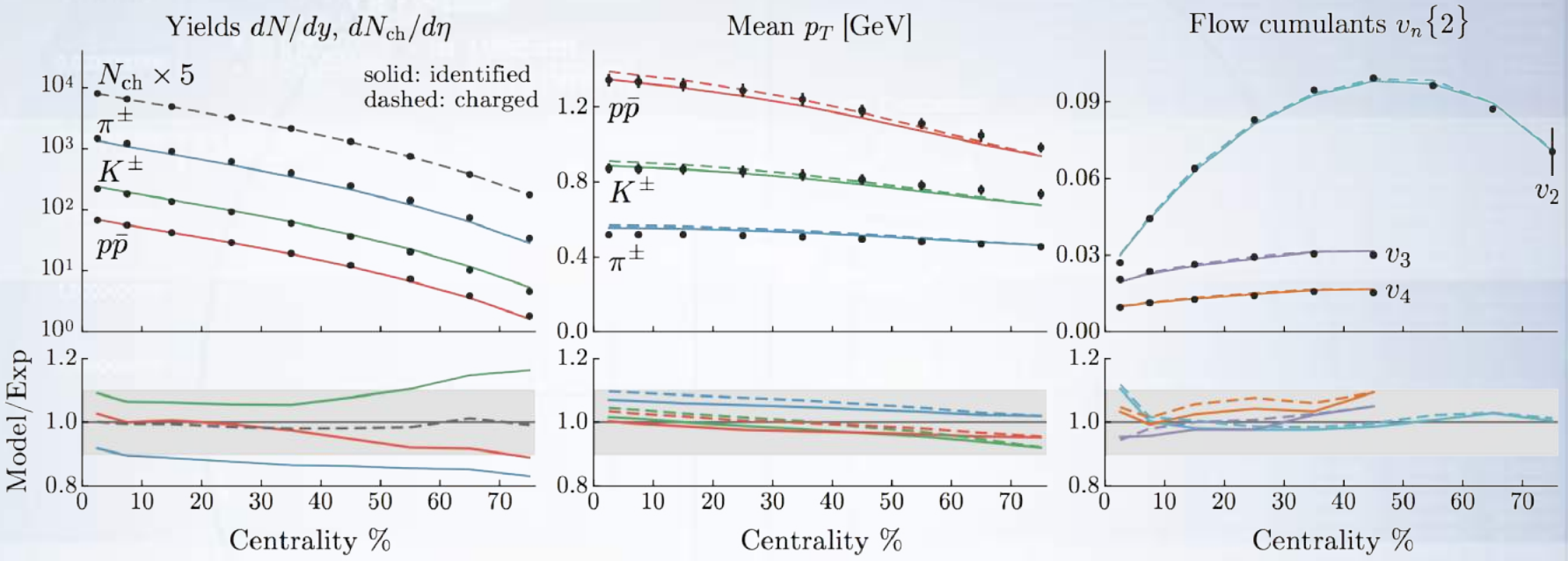
Collisions simulation optimization



Validation of Gaussian process emulator predictions. Each panel shows predictions compared to explicit model calculations at the 50 validation design points.

Jonah E. Bernhard, et. al., Applying Bayesian parameter estimation to relativistic heavy-ion collisions: simultaneous characterization of the initial state and quark-gluon plasma medium, Phys. Rev. C 94, 024907 (2016)

Collisions simulation optimization



Lines correspond to different simulation models with optimal parameters. Points are data from the ALICE experiment. Bottom: ratio of model calculations to data, where the gray band indicates ±10%.

Jonah E. Bernhard, et. al., Applying Bayesian parameter estimation to relativistic heavy-ion collisions: simultaneous characterization of the initial state and quark-gluon plasma medium, Phys. Rev. C 94, 024907 (2016)