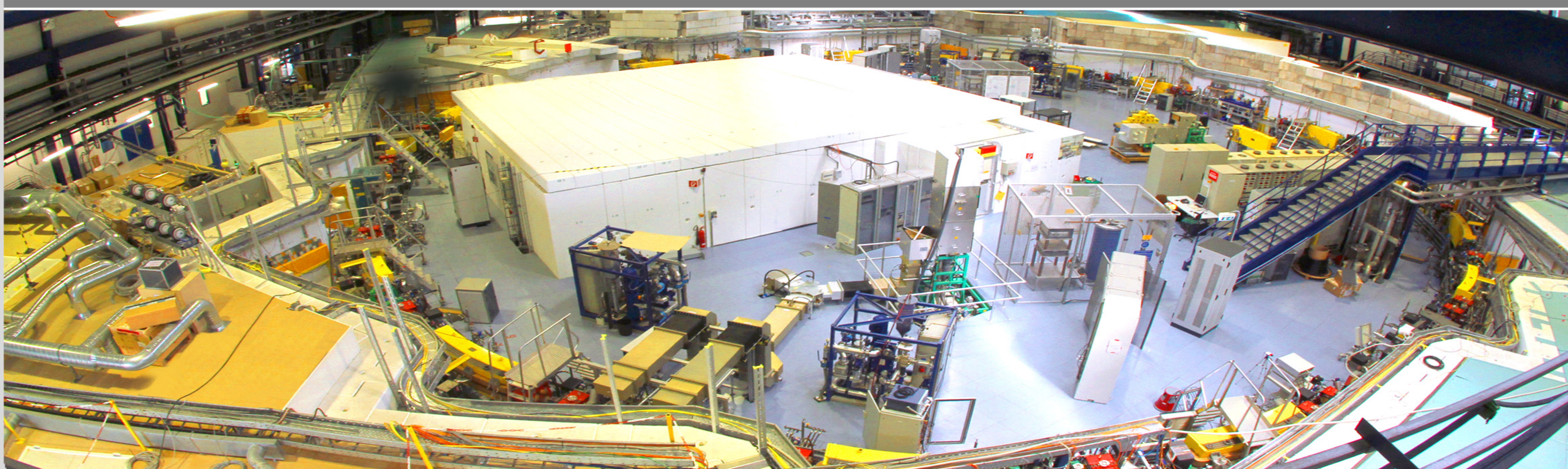
Bayesian Optimization of Injection Efficiency at KARA using Gaussian Processes

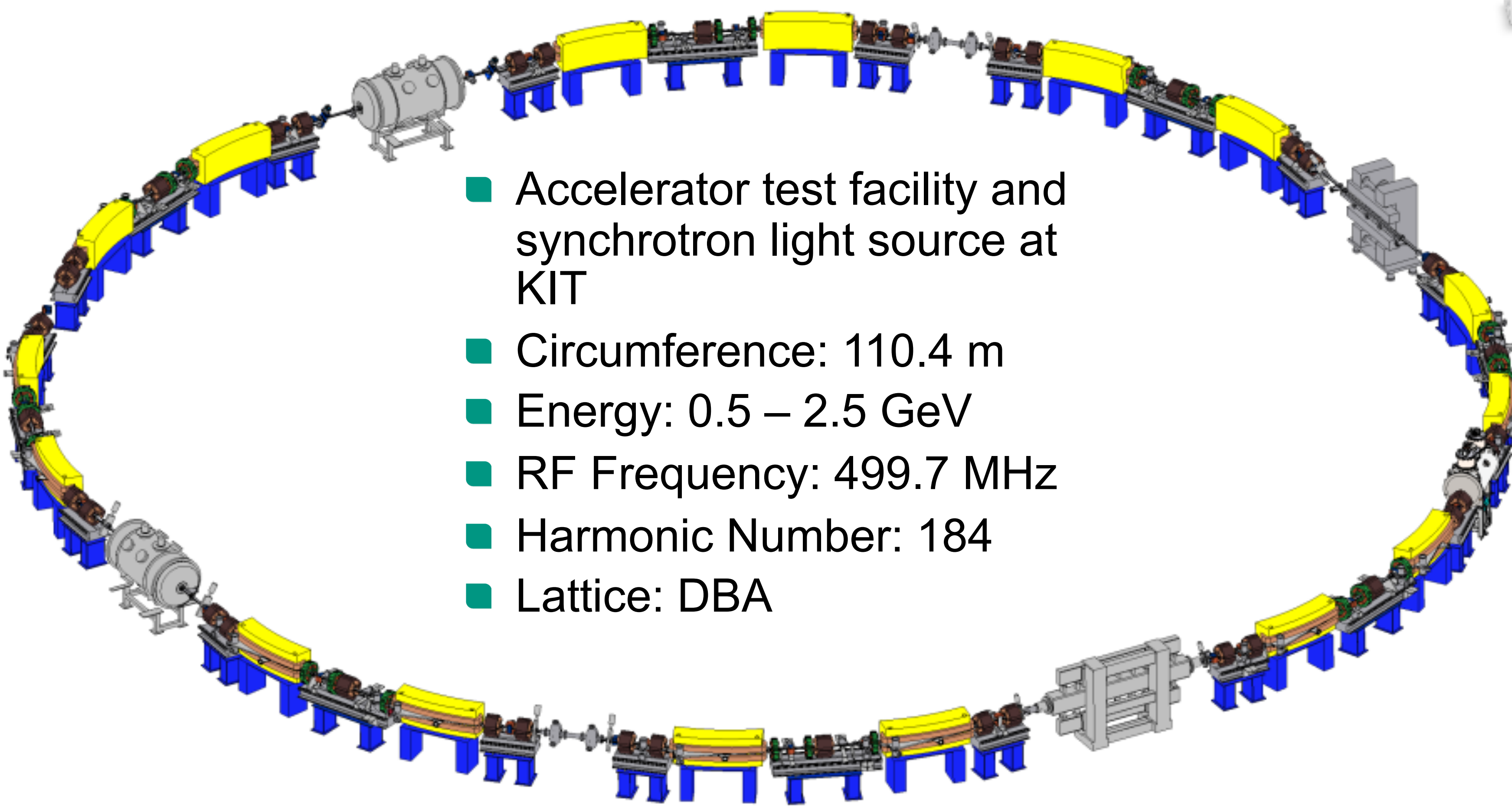
11.05.2020

Chenran Xu

Institute for Beam Physics and Technology (IBPT)

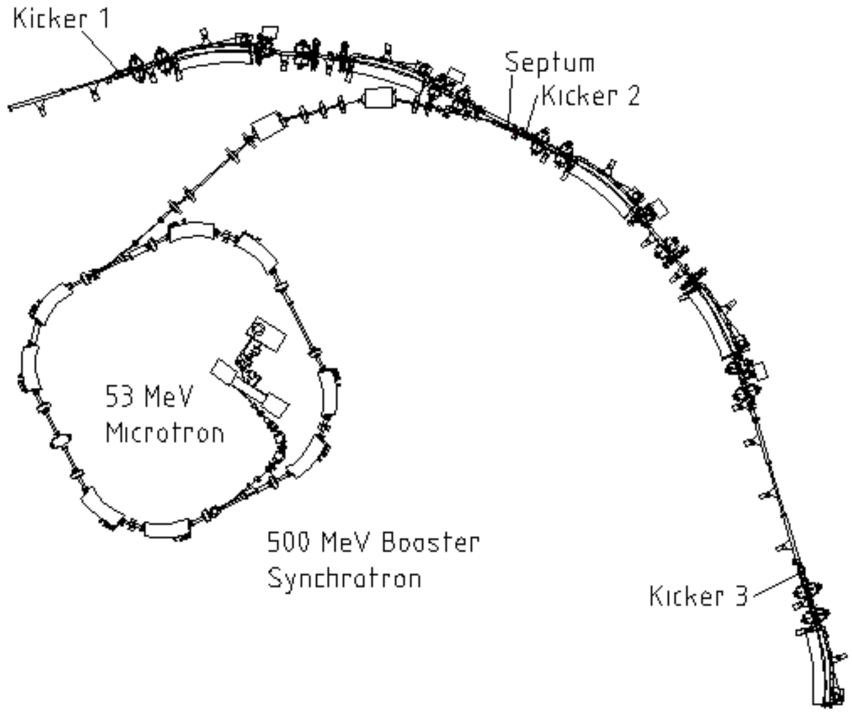


KARA Storage Ring

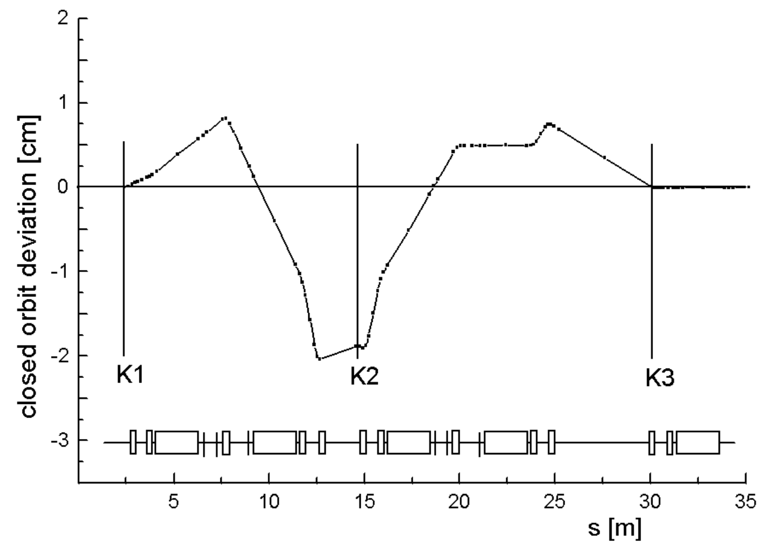


- Accelerator test facility and synchrotron light source at KIT
- Circumference: 110.4 m
- Energy: 0.5 – 2.5 GeV
- RF Frequency: 499.7 MHz
- Harmonic Number: 184
- Lattice: DBA

KARA Injection Scheme



- Injection bump with 3 kickers



- Injector:
  - Rep. rate: 1 Hz
  - Microtron: to 53 MeV
  - Booster: to 500 MeV

Figure: D. Einfeld, The Injection Scheme for the ANKA Storage Ring, 1998

KARA Injection Scheme

- Tuning Parameters
  - RF frequency, Corrector magnets…
  - Kicker magnets, injection septum...
- Problem of manual tuning
  - Rely on experience
  - Time consuming
  - Can be stuck in local optima
- Motivate Bayesian optimization
  - Converge to global optimum
  - Fast tuning
  - Successfully implemented at LCLS, SwissFEL to tune FEL Performance

- Injection bump with 3 kickers

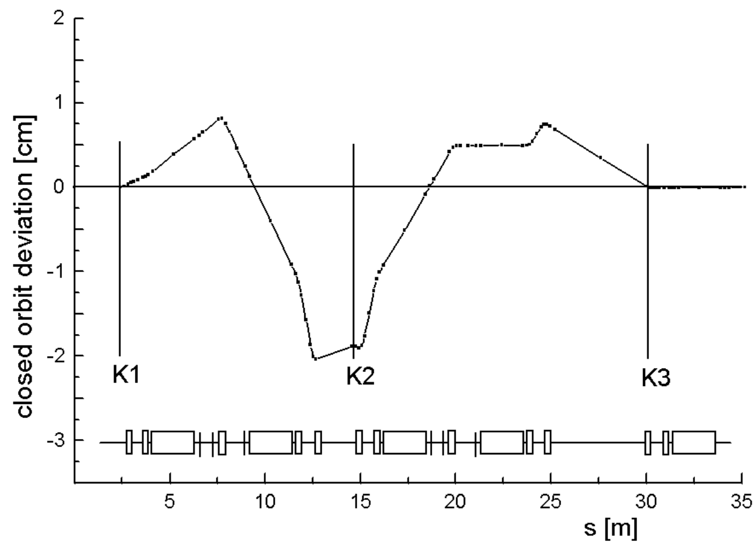


Figure: D. Einfeld, The Injection Scheme for the ANKA Storage Ring, 1998

Bayesian Optimization (BayesOpt) in a nutshell

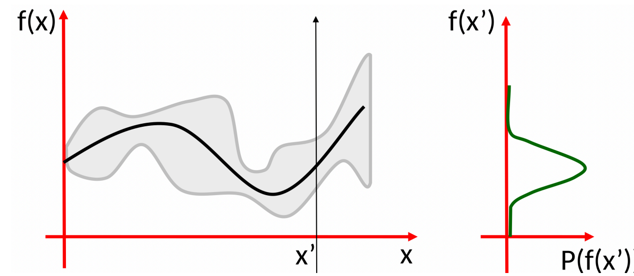
- Goal:
  - Global optimization of expensive to evaluate, black-box function
- Algorithm:
  - Build **probabilistic model** of the objective
    - Often use **Gaussian Process (GP)**
  - Use **acquisition function** to suggest next evaluation points
  - Sample new data, augment data set and update posterior probability
- Advantages:
  - Find optimum in minimum number of steps
    - more efficient than the Genetic Algorithm
    - able to incorporate prior knowledge
  - Explicitly model the observation noise

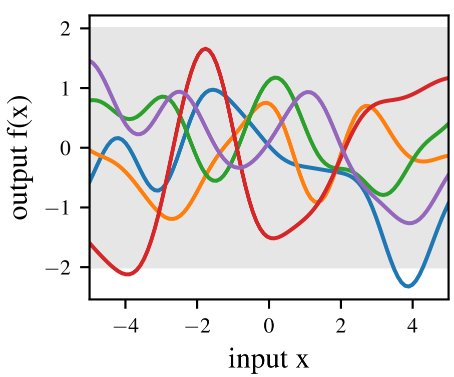
Gaussian Process

- In short:
  a powerful non-parametric function approximator
- $f(x) \sim \mathcal{GP}\big(\mu(x), k(x,x')\big)$ is fully specified by
  - $\mu(x)$: the mean function
    - Commonly set to constant zero
  - $k(x,x')$: the covariance function (kernel)
    - Measure of similarity of two close points
    - Example: Radial Basis Function (RBF)

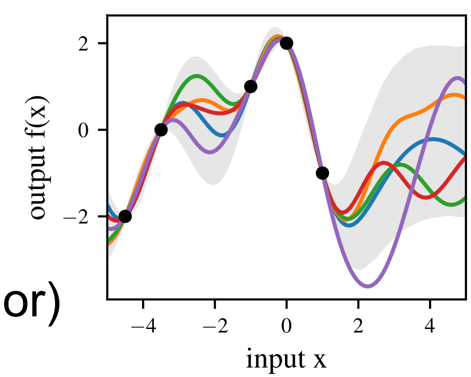
$$k(x,x') = \sigma^2 \exp\left(-\frac{\| x - x' \|^2}{2l^2}\right)$$

    - And many other possible kernels.





(prior)



(posterior)

Acquisition Functions

- Functions to guide the search & tradeoff exploration-exploitation
  - **UCB** (Upper Confidence Bound):
    - $a(x) = \mu(\mathrm{x}) + \beta * \sigma(x)$

exploitation term

exploration term

tradeoff parameter

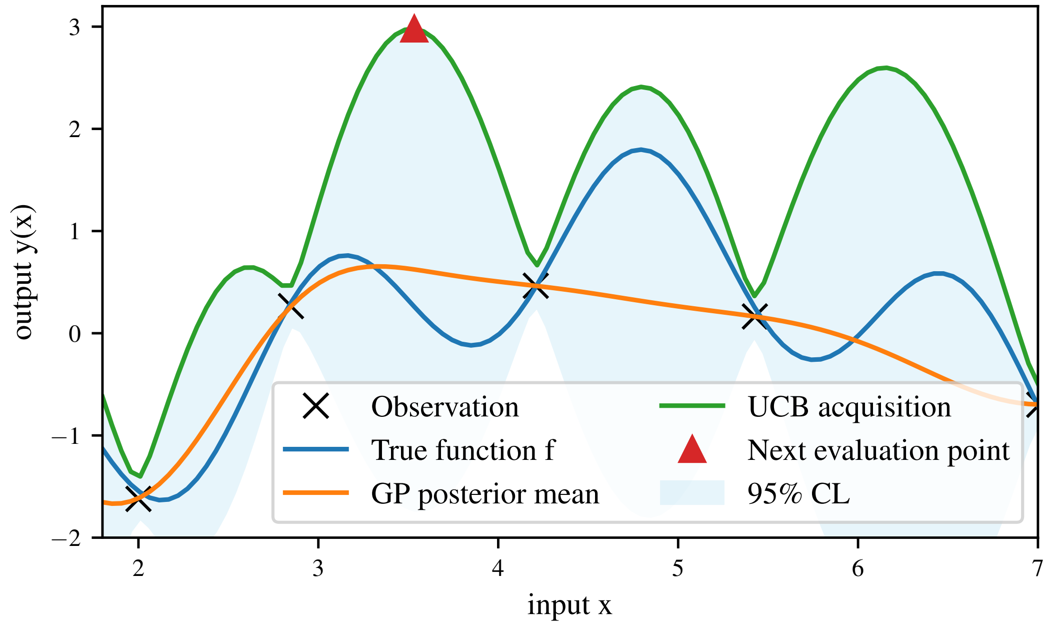
  - EI (Expected Improvement)
  - MPI (Maximum Probability of Improvement)
  - Entropy Search…

Bayesian Optimization

- Example using UCB acquisition function

$$a(x) = \mu(x) + \beta * \sigma(x) \ ,$$

Iteration 1:



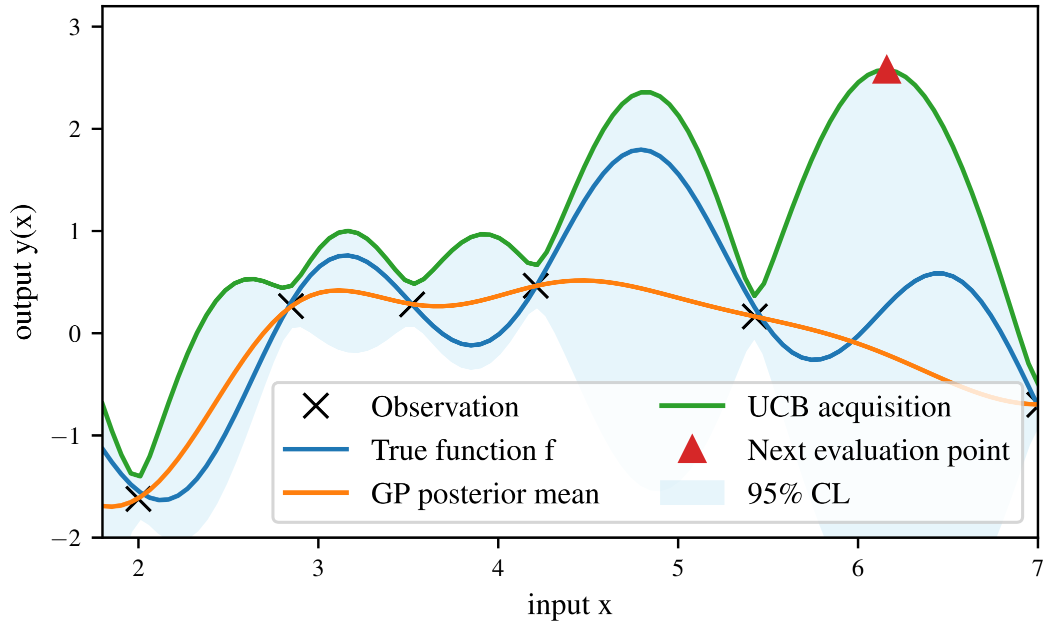
Here, $\beta = 2$ corresponding to a 95% CL

Bayesian Optimization

- Example using UCB acquisition function

$$a(x) = \mu(x) + \beta * \sigma(x) \quad ,$$

Iteration 2:



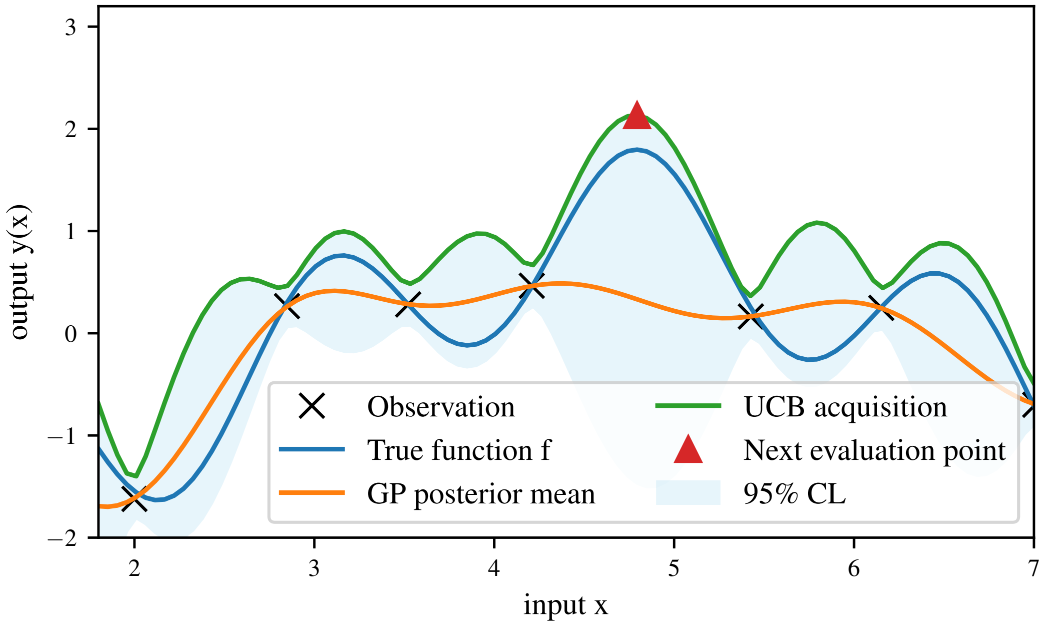
Here, $\beta = 2$ corresponding to a 95% CL

Bayesian Optimization

- Example using UCB acquisition function

$$a(x) = \mu(x) + \beta * \sigma(x) \ ,$$

Iteration 3:



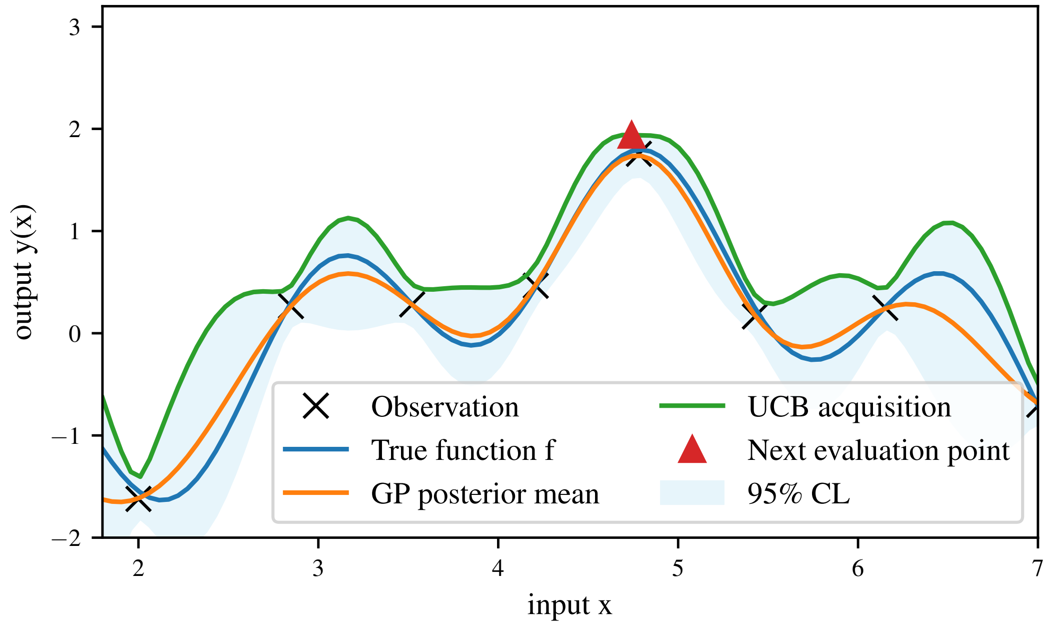
Here, $\beta = 2$ corresponding to a 95% CL

Bayesian Optimization

- Example using UCB acquisition function

$$a(x) = \mu(x) + \beta * \sigma(x) \ ,$$

Iteration 4:



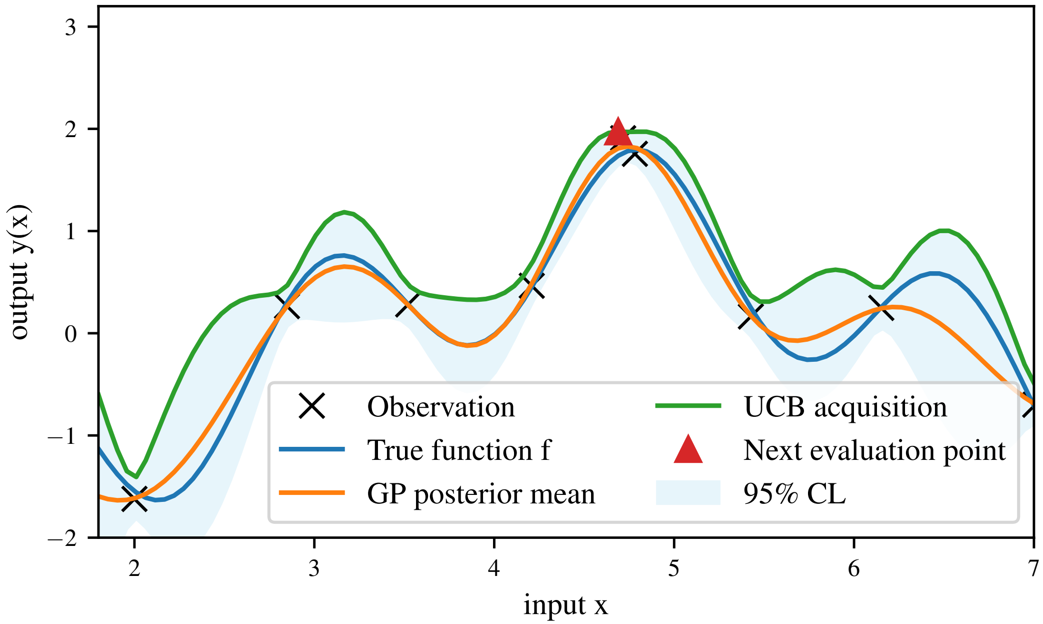
Here, $\beta = 2$ corresponding to a 95% CL

Bayesian Optimization

- Example using UCB acquisition function

$$a(x) = \mu(x) + \beta * \sigma(x) \ ,$$

Iteration 5:



Here, $\beta = 2$ corresponding to a 95% CL

Problem Statement & Implementation

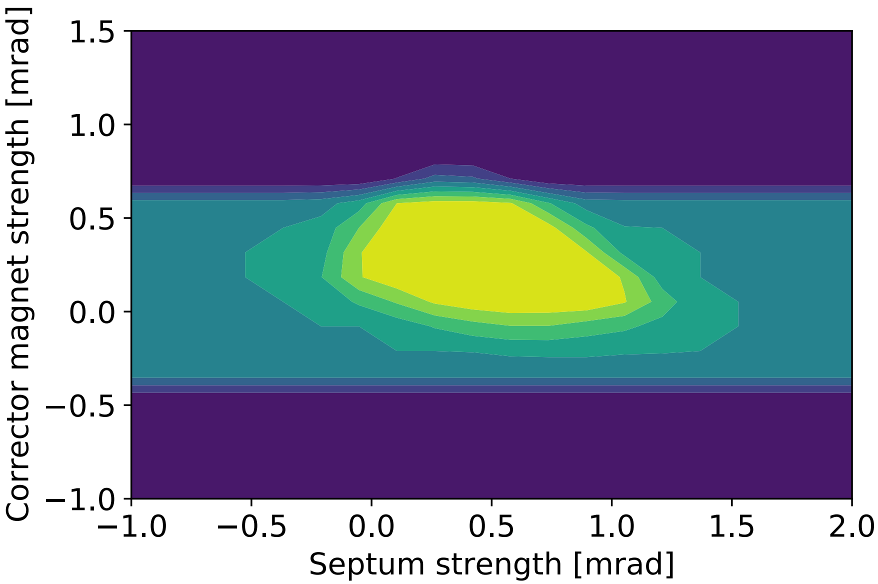
- Goal:
  - Optimize injection efficiency by tuning storage ring parameters
- Objective:
  - Injection Efficiency := (ring injected current – lost stored current) /booster extraction current
- Implementation
  - Python package, use GPy[1] for building GP model
  - Proof of principle using simulation model in Accelerator Toolbox (AT)[1]
- Implementation at KARA:
  - Use control system to read-back and set new machine parameter values (pyepics)

[1]. A. Terebilo, "Accelerator Modeling with MATLAB Accelerator Toolbox", 2001

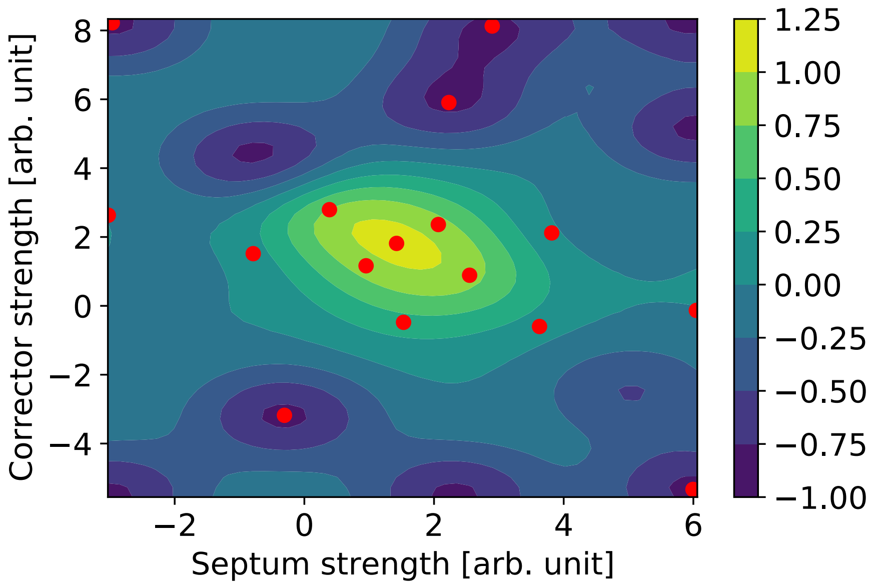
[2]. Gpy : A Gaussian process framework in python, http://github.com/SheffieldML/GPy

Simulation Results

- Build KARA ring model in AT
- Track injected & stored beam from injection point, for hundred turns.



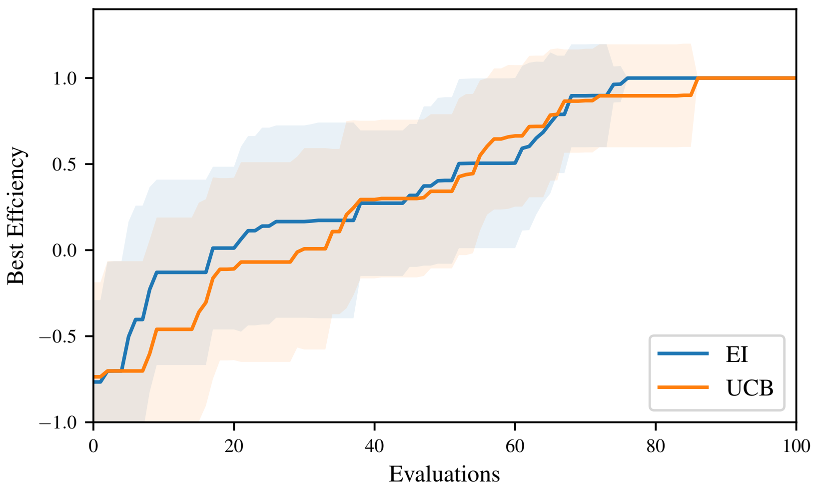
Grid scan of parameter space
Time : ~ 1 hour



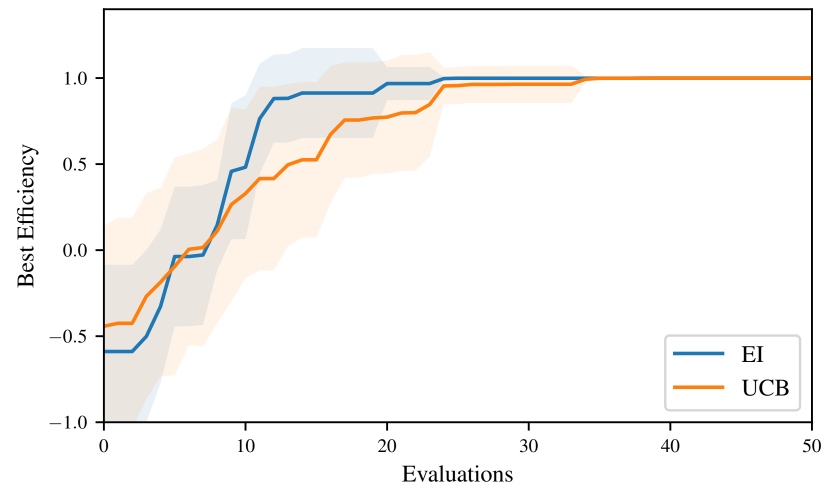
Bayesian Optimization
Time : ~ 3 minutes

Simulation Results

$d = 3$



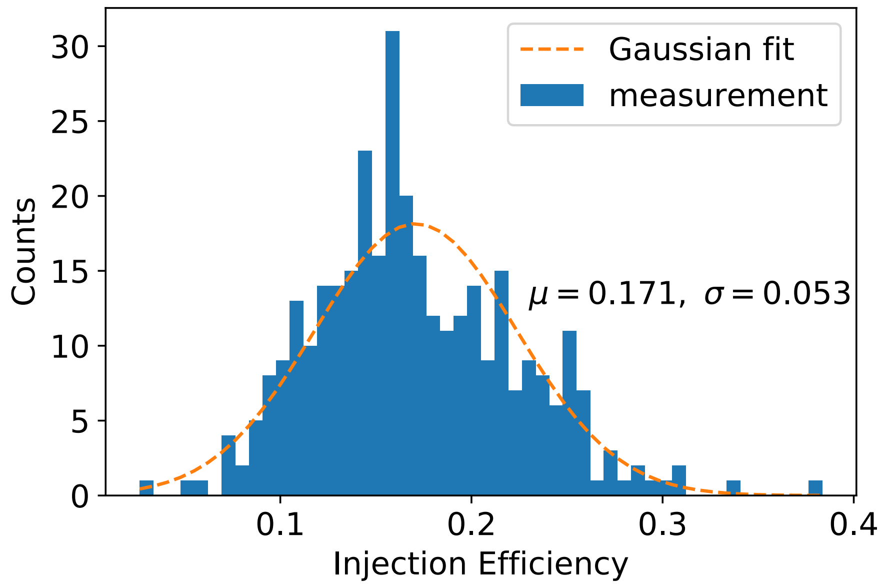
$d = 6$



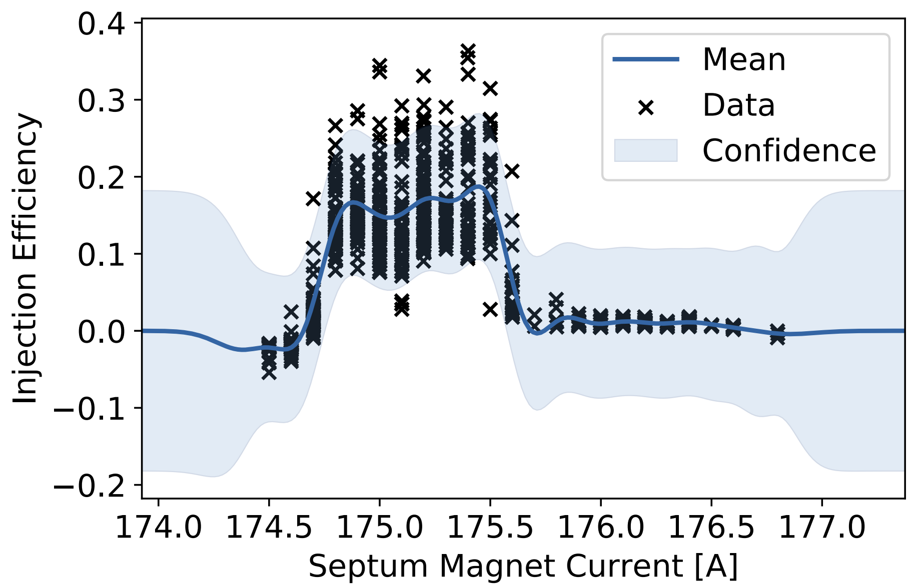
Reward averaged over 10 runs.
BayesOpt is able to optimize the simulation problem relatively fast.

Pre-processing

- Measure uncertainty of injection efficiency with fixed machine parameter settings, determine the noise level.



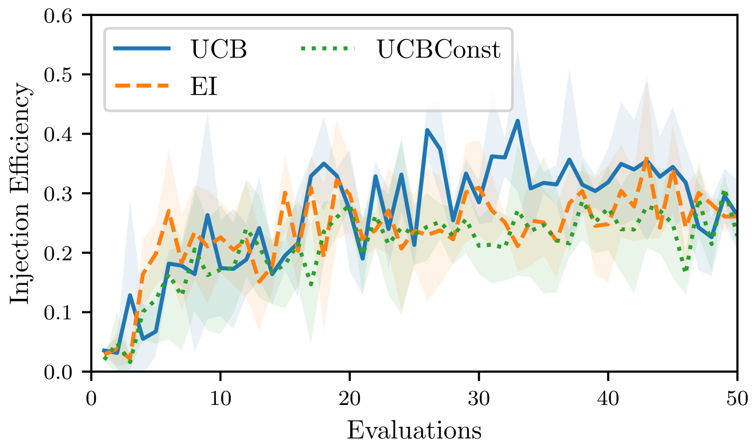
- 1-d scan of parameter space to get an estimate of
  - feasible parameter region
  - length scale in RBF covariance
- Example: scan of injection septum magnet strengths



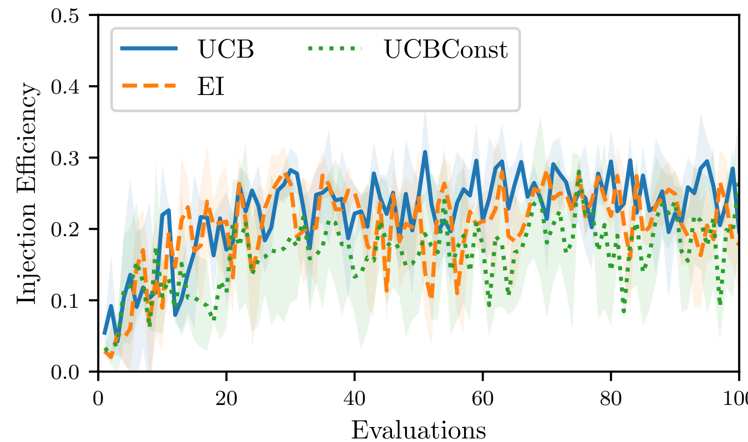
Implementation at KARA

- Different acquisition functions
  - Detune the injection condition to same starting point

$d = 6$



$d = 9$

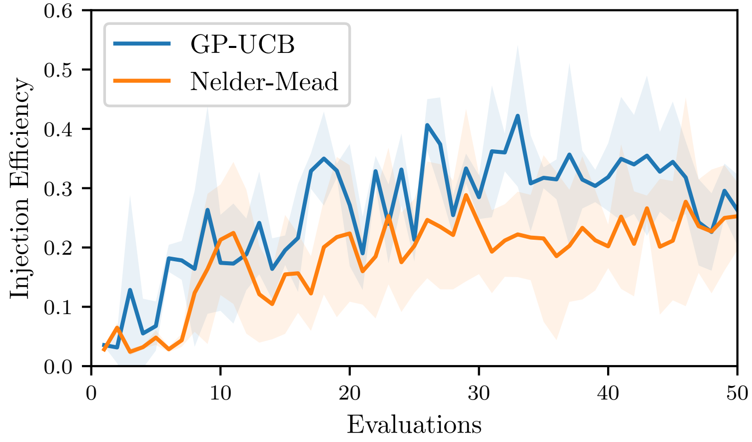


All three acquisition functions achieved similar results
Optmization time : ~10 s per-step ➔ 10 min for 50 evaluations
Most of the time is for settling & evaluation of objective
Computation time per-step: < 1s

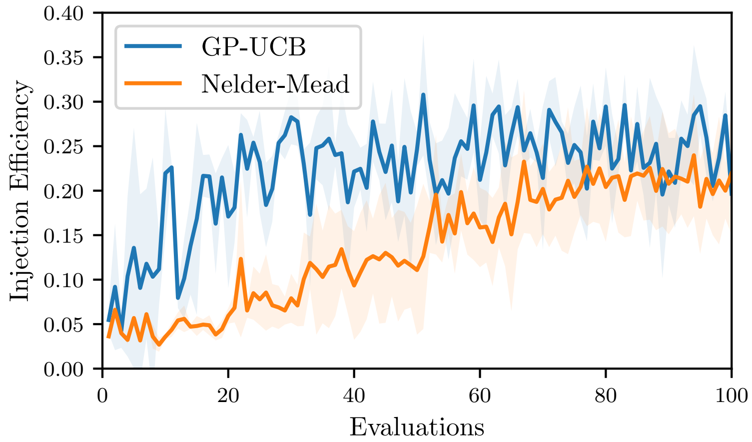
Implementation at KARA

- Benchmark against Nelder-Mead (NM) algorithm
  - Detune the injection condition to same starting point

$d = 6$



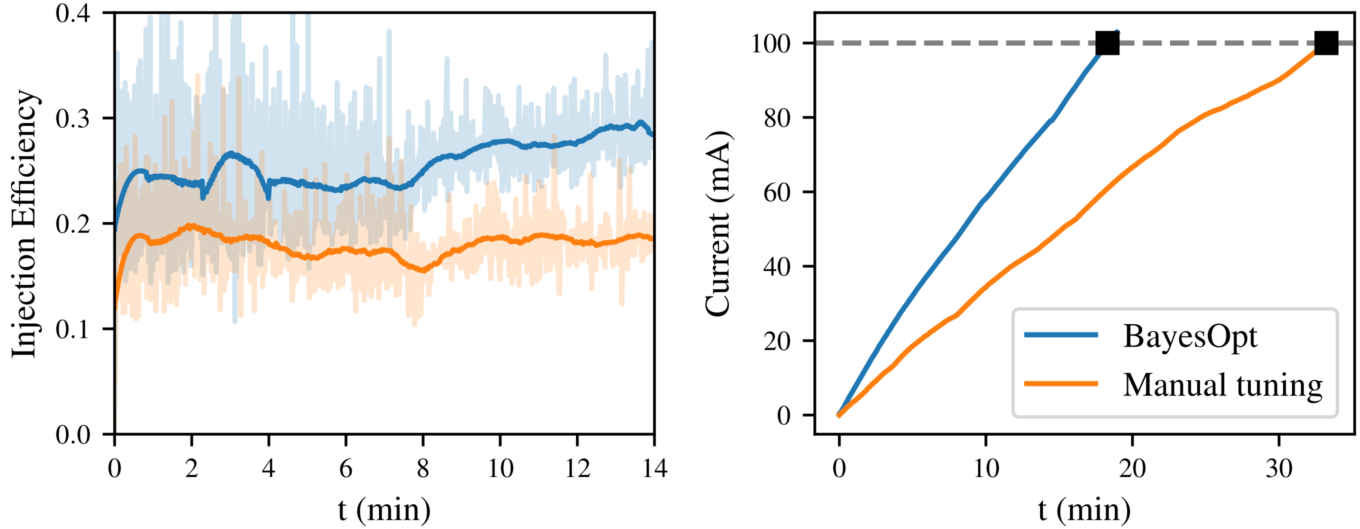
$d = 9$



- NM needs to be restarted, otherwise it can get stuck
- BayesOpt optimizes faster than NM.

Implementation at KARA

- Possible usage scenario: optimize & inject



- Injection time up to 100 mA :
  - Bayesian optimization: ~ 18 min
  - Manual Tuning: ~ 30 min

Summary & Outlook

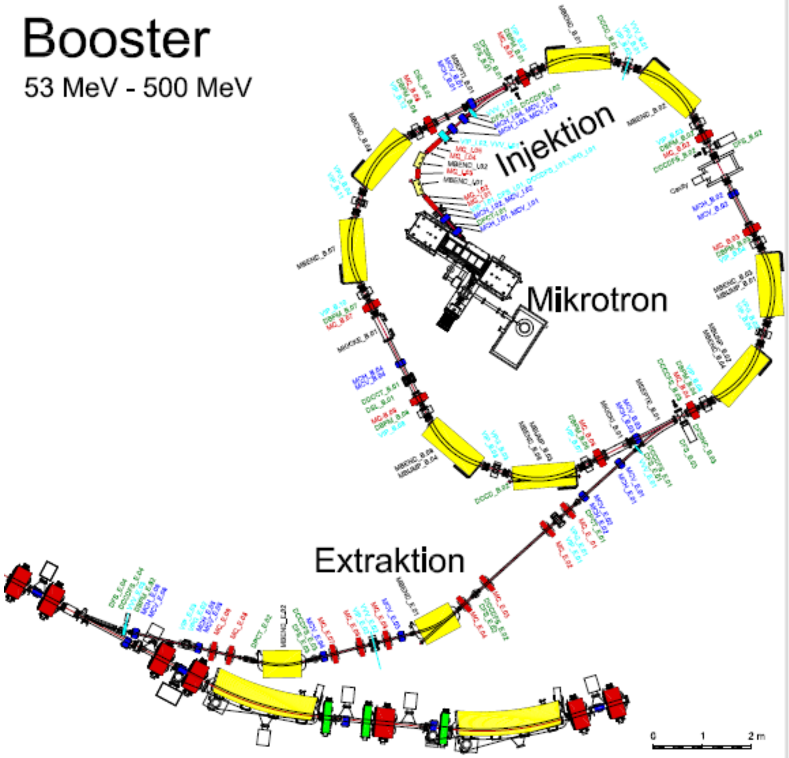
- Works well for both the simulation model & real KARA machine
- Can find optimum in small number of steps in $d < 9$ parameter space
- More robust to noise, and faster than Nelder-Mead
- Achieved better injection efficiency than manual tuning
- Injection optimization half-automated
  - still need to tune pre-accelerators

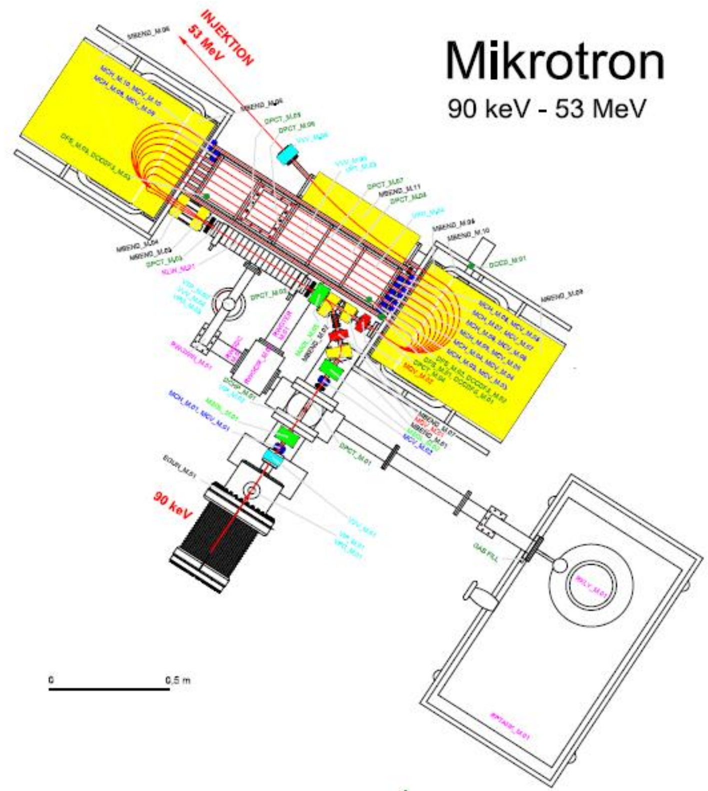
- Outlook:
  - Include more tuning parameters & expand use cases
    - also for other optics: low α_c, negative α_c…
  - Incorporate safety constraints
    - Safe BayesOpt: learn safe conditions along with optimization
  - Contextual GP
    - Include non-controllable parameters: current, temperature…

Thank you for your attention!

Backup

Introduction KARA Injector



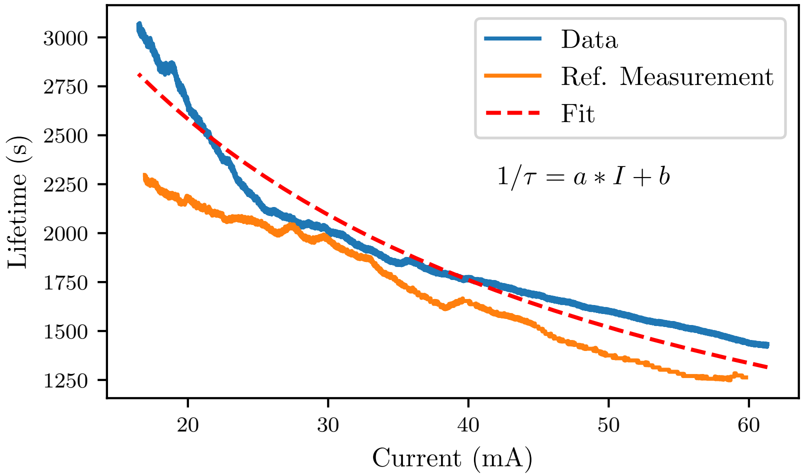


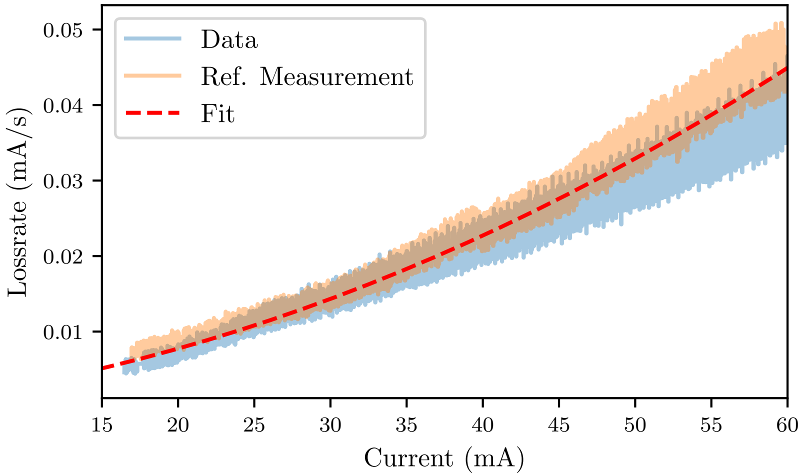
- Energy: 500 MeV
- Circumference: 24 m
- Harmonic Number: 44
- Rep. Rate: 1 Hz

- Energy: 53 MeV
- RF Frequency: 2.999 GHz
- Number of Turns: 10

Pre-processing

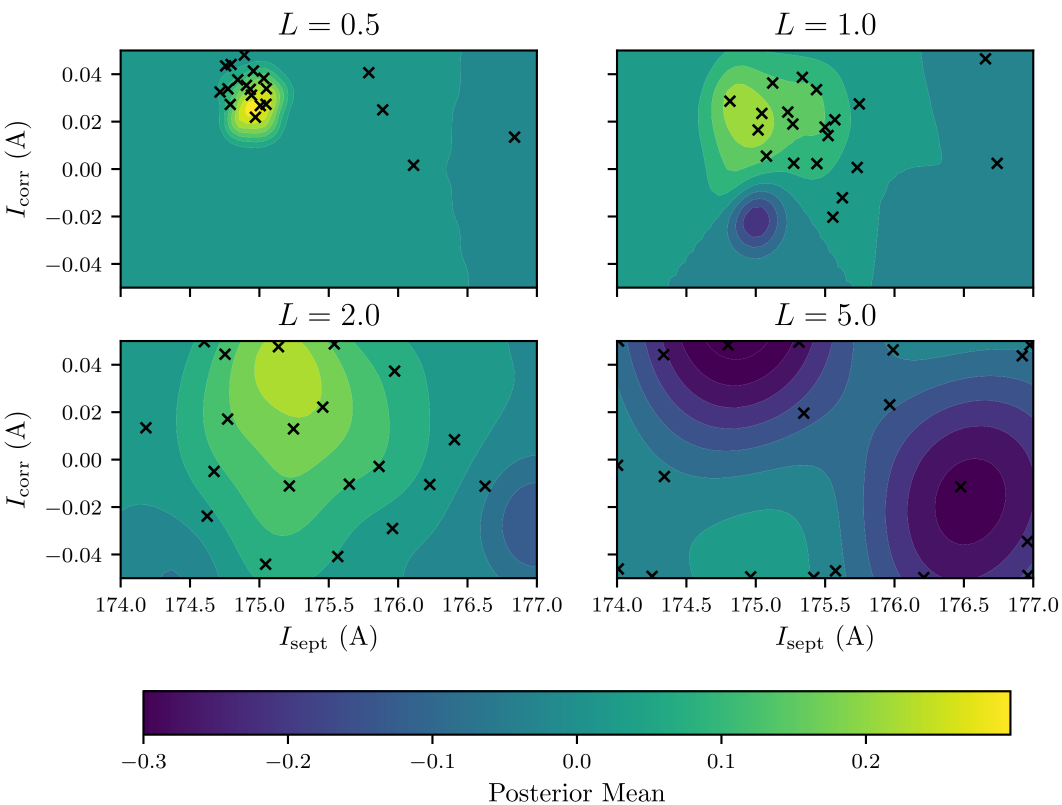
- Measure current dependency of lifetime
  - Correct decay rate from the injection efficiency using fitted data
  - Touschek dominant at 0.5 GeV





Implementation at KARA

- Effect of hyperparameter:
  - Length-scale: $k(x, x') = \sigma^2 \exp\left(-\frac{\|x-x'\|^2}{2L^2}\right)$

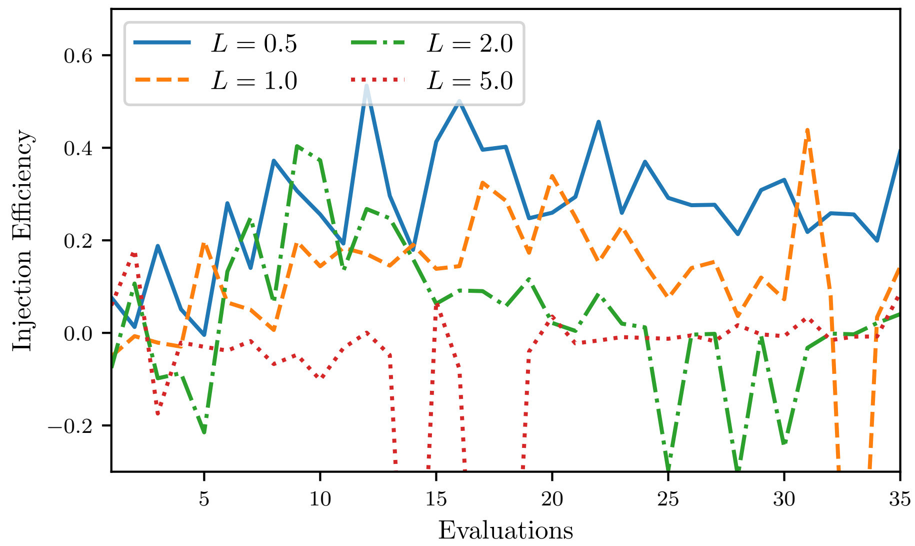


large L:
➔ large structure, correlation between distant points
➔ may overlook optimum

small L:
➔ small structure, correlation only between nearby points
➔ may overfit & optimize locally

Implementation at KARA

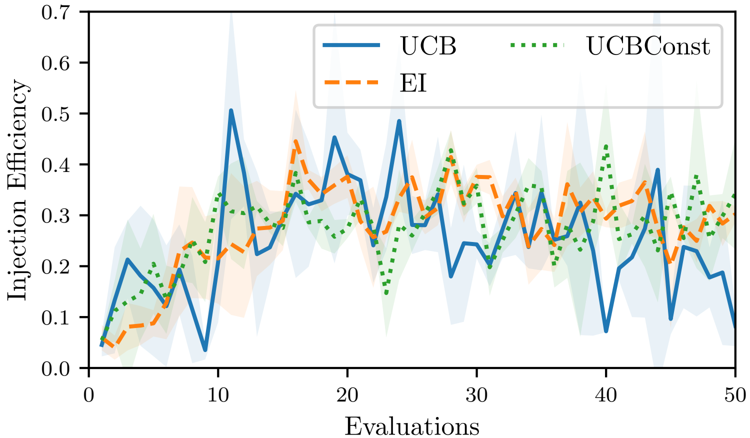
- Performance of different length-scales

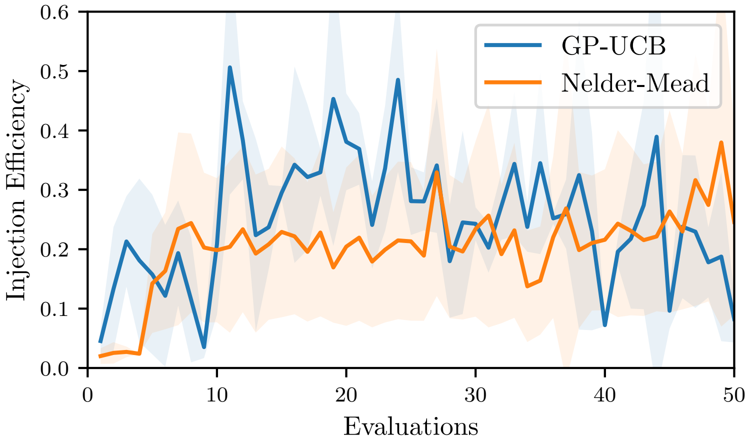


Due to the exploration behavior of GP, unaccounted beam loss can happen.

Implementation at KARA

- $d = 3$ Results





Implementation at KARA

- NM can get stuck due to noisy evaluations and needs to be restarted