Safe Bayesian Optimization for Tuning Particle Accelerators

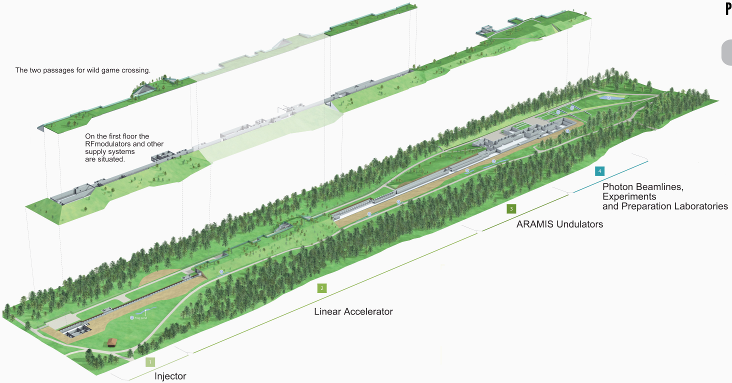
J. Kirschner, J. Coello, N. Hiller, J. Snuverink, M. Mutny, M. Nonnenmacher, R. Ischebeck, A. Krause

March 2nd, 2021

PACMAN progress meeting

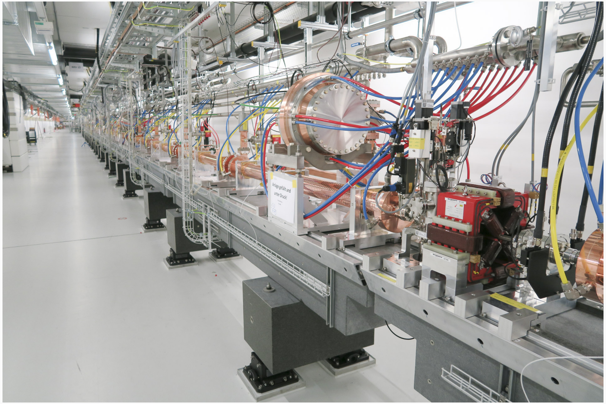
Task 1: Beam Optimization at SwissFEL

PAUL SCHERRER INSTITUT



Objective: Beam intensity

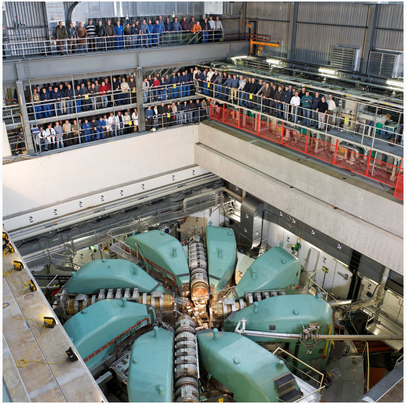
Task 1: Beam Optimization at SwissFEL



PAUL SCHERRER INSTITUT

Objective: Beam intensity

Task 2: Loss Minimization at HIPA



PAUL SCHERRER INSTITUT

Objective: Minimize proton losses

Optimizing Black-Box Functions

Formal objective:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

Optimizing Black-Box Functions

Formal objective:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

- Parameter space $\mathcal{X} \subset \mathbb{R}^d$

Optimizing Black-Box Functions

Formal objective:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

- Parameter space $\mathcal{X} \subset \mathbb{R}^d$
- Weak assumptions on f: *smoothness, but not convex, no analytical form, ...*

Optimizing Black-Box Functions

Formal objective:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

- ▷ Parameter space $\mathcal{X} \subset \mathbb{R}^d$
- ▷ Weak assumptions on f: *smoothness, but not convex, no analytical form, ...*

Get noisy point evaluations $y = f(x) + \epsilon$

- ▷ *Noisy Zero-Order Access / Blackbox*
- ▷ No access to gradients

Optimizing Black-Box Functions

Formal objective:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

- $\triangleright$ Parameter space $\mathcal{X} \subset \mathbb{R}^d$
- $\triangleright$ Weak assumptions on f: *smoothness, but not convex, no analytical form, ...*

Get noisy point evaluations $y = f(x) + \epsilon$

- $\triangleright$ *Noisy Zero-Order Access / Blackbox*
- $\triangleright$ No access to gradients
- $\triangleright$ Evaluations of f are 'expensive': *sample efficiency is important*

Optimization Techniques: Overview

Grid search

- Good for small problems with little noise

Optimization Techniques: Overview

Grid search

- Good for small problems with little noise

Local Methods

- Follow estimated gradient
- Nelder-Mead
- Random walk

Optimization Techniques: Overview

Grid search

- Good for small problems with little noise

Local Methods

- Follow estimated gradient
- Nelder-Mead
- Random walk

Evolutionary Algorithms

- Keep a population of candidates and refine
- E.g. CMA-ES, Particle Swarm

Optimization Techniques: Overview

Grid search

- Good for small problems with little noise

Local Methods

- Follow estimated gradient
- Nelder-Mead
- Random walk

Evolutionary Algorithms

- Keep a population of candidates and refine
- E.g. CMA-ES, Particle Swarm

Global Methods

- Bayesian Optimization **(this talk)**

Bayesian Optimization: Overview

For each step $t = 1, 2, 3, \ldots, n$,

1 : Compute *probabilistic model* of target

- ▷ Uses collected & prior data
- ▷ Gaussian process (GP) regression

Bayesian Optimization: Overview

For each step $t = 1, 2, 3, \ldots, n$,

1: Compute *probabilistic model* of target

- ▷ Uses collected & prior data
- ▷ Gaussian process (GP) regression

2: Choose *evaluation point*

- ▷ Score function based on model

Bayesian Optimization: Overview

For each step $t = 1, 2, 3, \ldots, n$,

1: Compute *probabilistic model* of target

- ▷ Uses collected & prior data
- ▷ Gaussian process (GP) regression

2: Choose *evaluation point*

- ▷ Score function based on model

3: Noisy measurement $y_t = f(x_t) + \epsilon$

- ▷ Update model

Bayesian Optimization: Overview

For each step $t = 1, 2, 3, \ldots, n$,

1: Compute *probabilistic model* of target

- ▷ Uses collected & prior data
- ▷ Gaussian process (GP) regression

2: Choose *evaluation point*

- ▷ Score function based on model

3: Noisy measurement $y_t = f(x_t) + \epsilon$

- ▷ Update model

Return: *return best measured or best predicted setting*.

Constraint Optimization

$$\arg\max_{x\in\mathcal{X}} f(x) \quad \text{s.t.} \quad g(x) \leq 0$$

Safe Optimization:

- Iterates need to satisfy $g(x_t) \leq 0$
- Apriori unknown constraint function g
- Measure at x and observe $f(x) + \epsilon$, $g(x) + \rho$
- Multiple constraints

Constraint Optimization

$$\arg\max_{x \in \mathcal{X}} f(x) \quad \text{s.t.} \quad g(x) \leq 0$$

Safe Optimization:

- Iterates need to satisfy $g(x_t) \leq 0$
- Apriori unknown constraint function g
- Measure at x and observe $f(x) + \epsilon$, $g(x) + \rho$
- Multiple constraints

At HIPA:

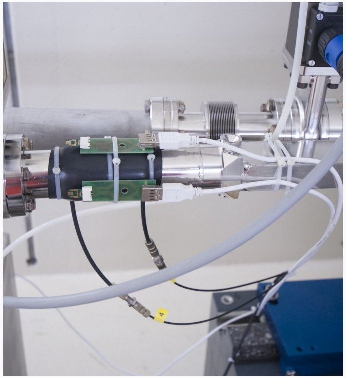
- Losses monitors, step-size constraints

Constraint Optimization

$$\arg\max_{x \in \mathcal{X}} f(x) \quad \text{s.t.} \quad g(x) \leq 0$$

Safe Optimization:

- Iterates need to satisfy $g(x_t) \leq 0$
- Apriori unknown constraint function g
- Measure at x and observe $f(x) + \epsilon$, $g(x) + \rho$
- Multiple constraints



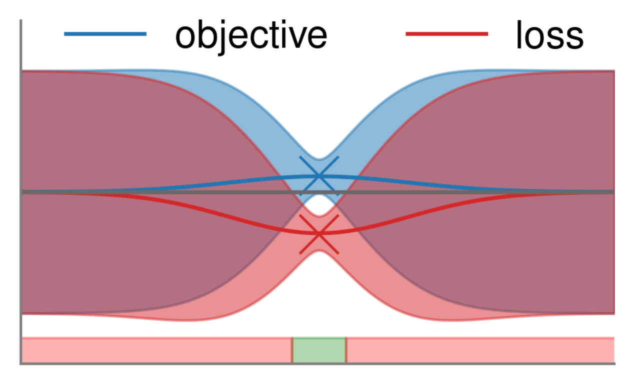
At HIPA:

- Losses monitors, step-size constraints

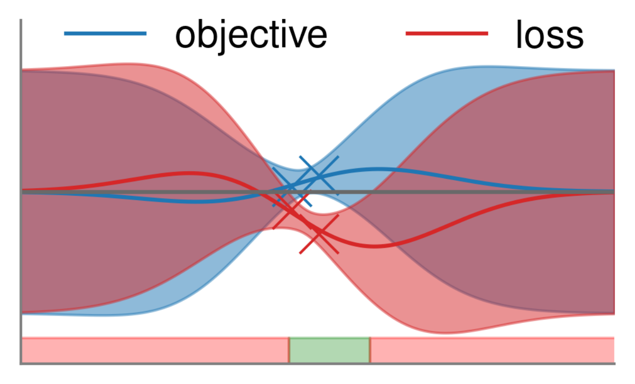
At SwissFEL:

- Avoid electron losses, minimum signal

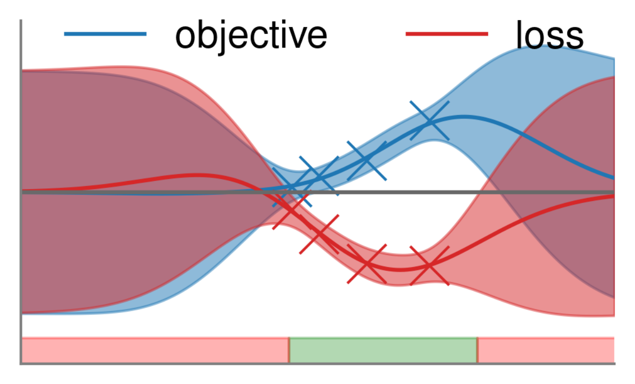
Safe Bayesian Optimization: Illustration



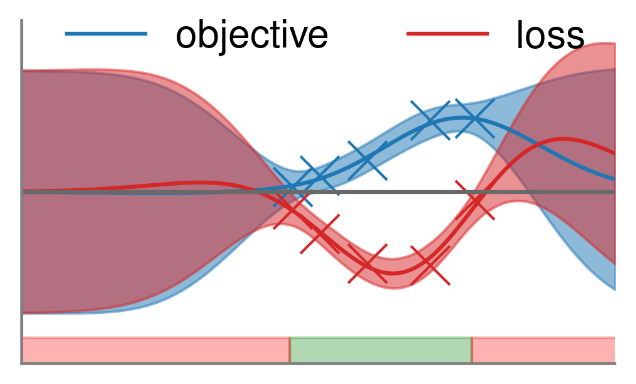
Safe Bayesian Optimization: Illustration



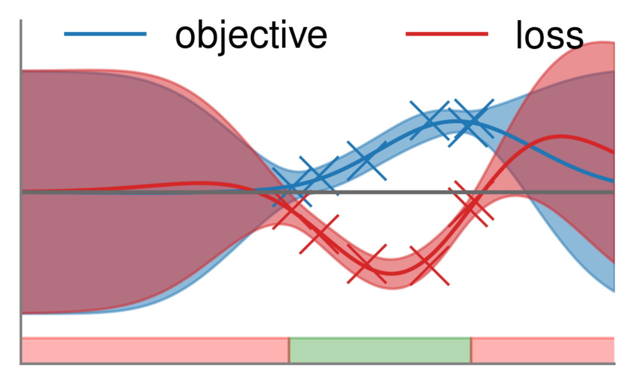
Safe Bayesian Optimization: Illustration



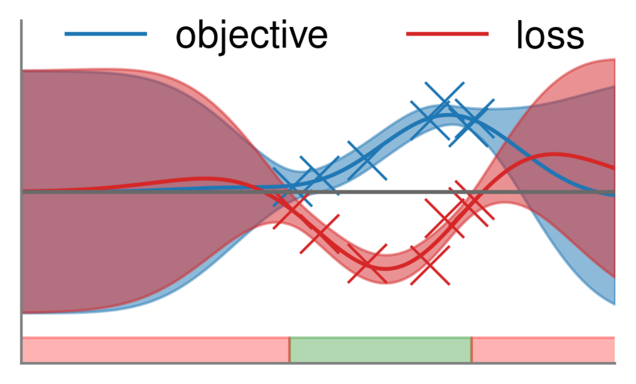
Safe Bayesian Optimization: Illustration



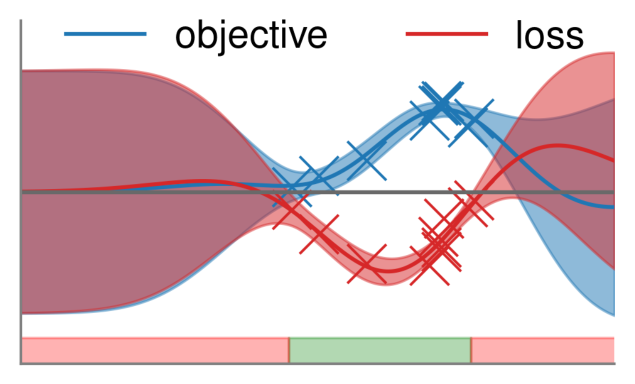
Safe Bayesian Optimization: Illustration



Safe Bayesian Optimization: Illustration



Safe Bayesian Optimization: Illustration



Scaling to High Dimensions (ICML paper)

$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1: **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

- ▷ Gaussian process / kernel regression

Scaling to High Dimensions (ICML paper)

$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1: **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

- Gaussian process / kernel regression

2: **Acquisition**: $x_t = \arg\max_x \alpha(x | \hat{f}_t, \hat{g}_t)$

- balances exploration-exploitation and safety
- augment data $\mathcal{D}_{t+1}$

Scaling to High Dimensions (ICML paper)

$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1: **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

- Gaussian process / kernel regression

Difficult optimization problem

2: **Acquisition**: $x_t = \arg\max_x \alpha(x | \hat{f}_t, \hat{g}_t)$

- balances exploration-exploitation and safety
- augment data $\mathcal{D}_{t+1}$

Scaling to High Dimensions (ICML paper)

$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1: **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

- Gaussian process / kernel regression

Restrict to 1d subspace $\mathcal{L}$

2: **Acquisition on $\mathcal{L}$:** $x_t = \arg\max_{x \in \mathcal{L}} \alpha(x | \hat{f}_t, \hat{g}_t)$

- balances exploration-exploitation and safety
- augment data $\mathcal{D}_{t+1}$

Scaling to High Dimensions (ICML paper)

$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1 : **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

- ▷ Gaussian process / kernel regression

2 : **Acquisition on** $\mathcal{L}$: $x_t = \arg\max_{x \in \mathcal{L}} \alpha(x | \hat{f}_t, \hat{g}_t)$

- ▷ balances exploration-exploitation and safety
- ▷ augment data $\mathcal{D}_{t+1}$

3 : **If** error$(\mathcal{L}) < \epsilon$:

New 1d subspace $\mathcal{L}$ at best point $\hat{x}_t$ in *random* direction

Scaling to High Dimensions (ICML paper)

$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1 : **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

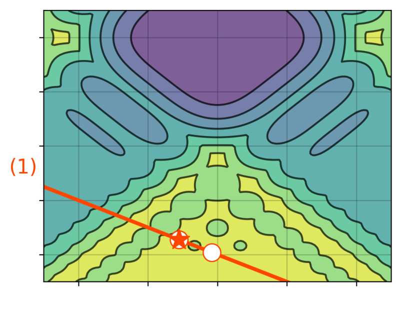
- Gaussian process / kernel regression

2 : **Acquisition on $\mathcal{L}$**: $x_t = \arg\max_{x \in \mathcal{L}} \alpha(x | \hat{f}_t, \hat{g}_t)$

- balances exploration-exploitation and safety
- augment data $\mathcal{D}_{t+1}$

3 : **If** error$(\mathcal{L}) < \epsilon$:

New 1d subspace $\mathcal{L}$ at best point $\hat{x}_t$ in *random* direction



Scaling to High Dimensions (ICML paper)

$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1 : **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

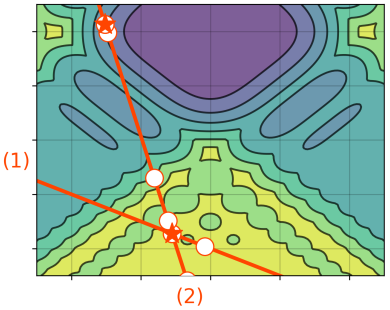
- ▷ Gaussian process / kernel regression

2 : **Acquisition on** $\mathcal{L}$: $x_t = \arg\max_{x \in \mathcal{L}} \alpha(x | \hat{f}_t, \hat{g}_t)$

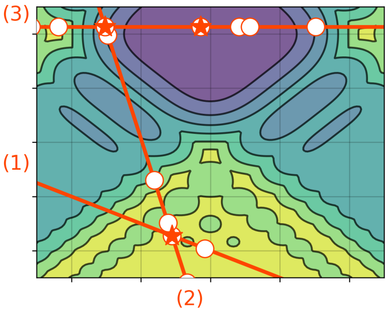
- ▷ balances exploration-exploitation and safety
- ▷ augment data $\mathcal{D}_{t+1}$

3 : **If** error$(\mathcal{L}) < \epsilon$:

New 1d subspace $\mathcal{L}$ at best point $\hat{x}_t$ in *random* direction



Scaling to High Dimensions (ICML paper)



$\mathcal{D}_0 = \{\}$

For $t = 1, 2, 3, \ldots$

1 : **Estimate:** $\hat{f}_t, \hat{g}_t | \mathcal{D}_t$

- ▷ Gaussian process / kernel regression

2 : **Acquisition on** $\mathcal{L}$: $x_t = \arg\max_{x \in \mathcal{L}} \alpha(x | \hat{f}_t, \hat{g}_t)$

- ▷ balances exploration-exploitation and safety
- ▷ augment data $\mathcal{D}_{t+1}$

3 : **If** $\text{error}(\mathcal{L}) < \epsilon$:

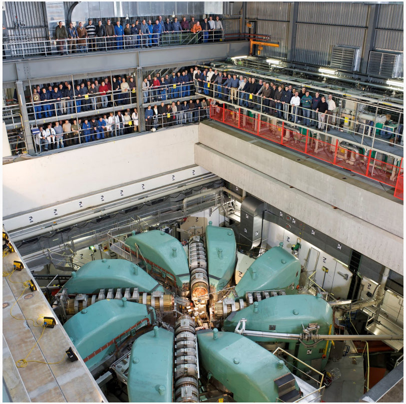
New 1d subspace $\mathcal{L}$ at best point $\hat{x}_t$ in *random* direction

the GUI

- by Jaime, Jochem, Nicole, J.

& contributions by Marco B., Nicolas L.

Results on HIPA



HIPA Tuning: Setup

Objective: Minimize combined losses (M4HIPA:VERL:2)

HIPA Tuning: Setup

Objective: Minimize combined losses (M4HIPA:VERL:2)

Tuning Parameters: 5-16 Quadrupole Magnets

HIPA Tuning: Setup

Objective: Minimize combined losses (M4HIPA:VERL:2)

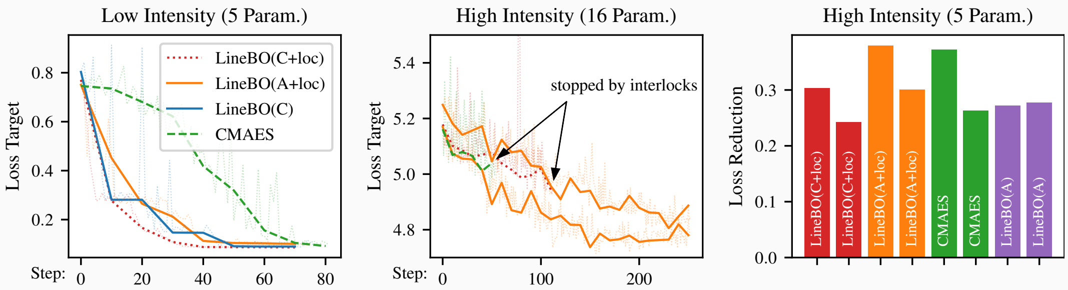
Tuning Parameters: 5-16 Quadrupole Magnets

Constraints:

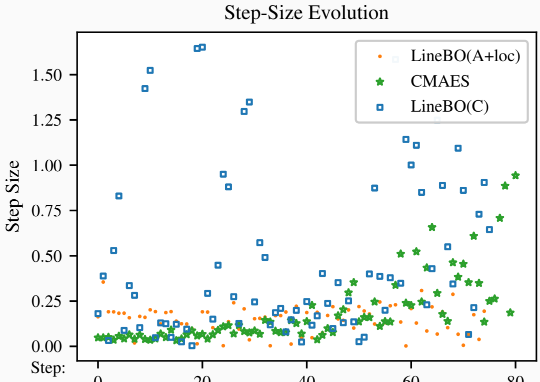
- ▷ About 200 loss monitors with individual warning levels
- ▷ Combined into a single constraint with $\max(\cdots)$.

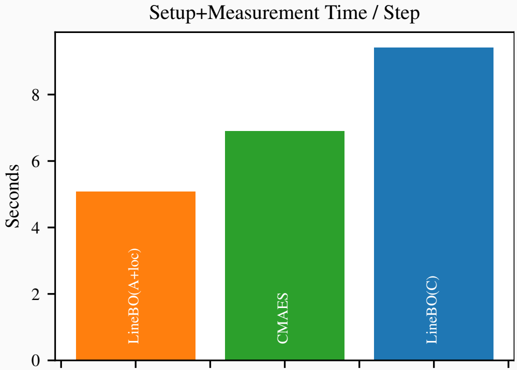
Effective control rate: $\sim$ 5 seconds / step

HIPA Tuning (Performance)

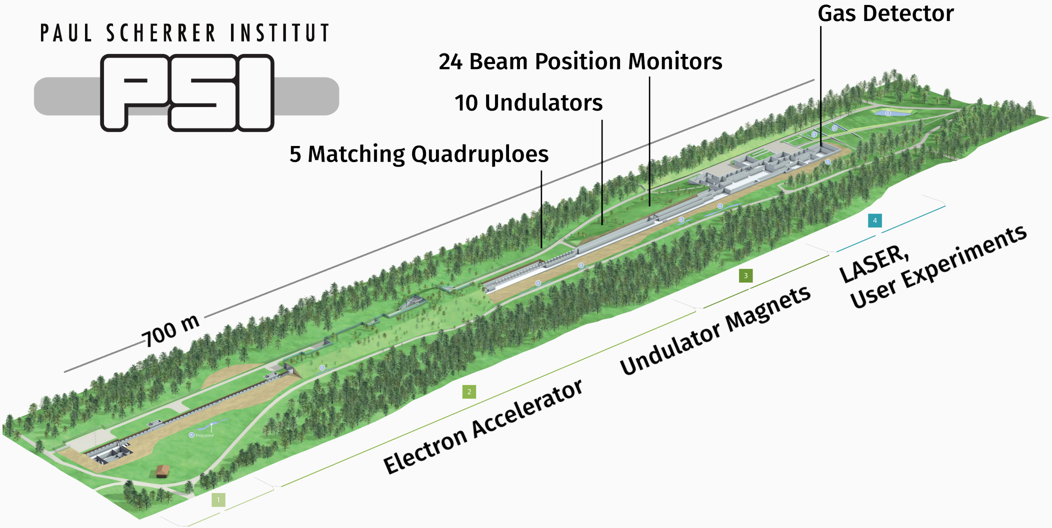


HIPA Tuning (Analysis)





Results on SwissFEL



SwissFEL Tuning: Setup

Objective: Shot-by-shot FEL intensity

SwissFEL Tuning: Setup

Objective: Shot-by-shot FEL intensity

Tuning Parameters: Quadrupole Magnets, Beam Position, Undulator K-Values

SwissFEL Tuning: Setup

Objective: Shot-by-shot FEL intensity

Tuning Parameters: Quadrupole Magnets, Beam Position, Undulator K-Values

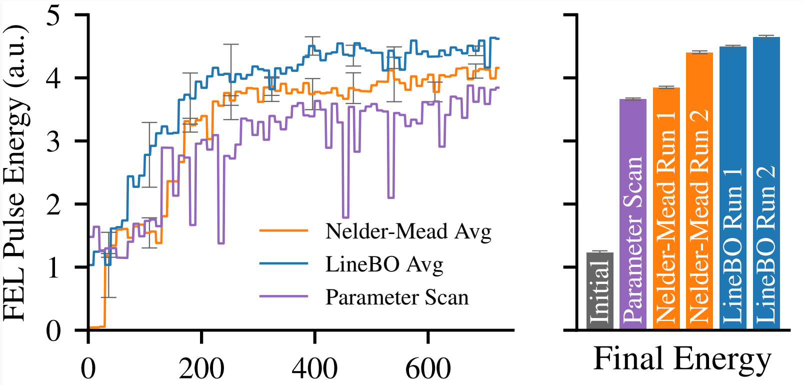
Constraints:

- ▷ Lower bound on intensity
- ▷ Loss monitors (not used because of technical difficulties)

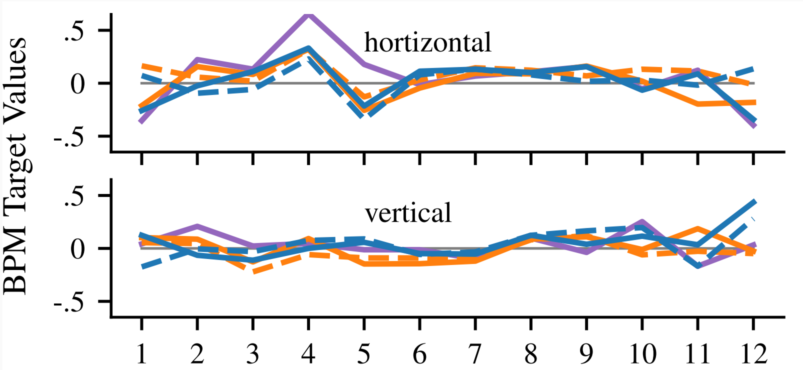
Effective control rate: 0.5-1 second / step

- ▷ Early test on new ATHOS beamline

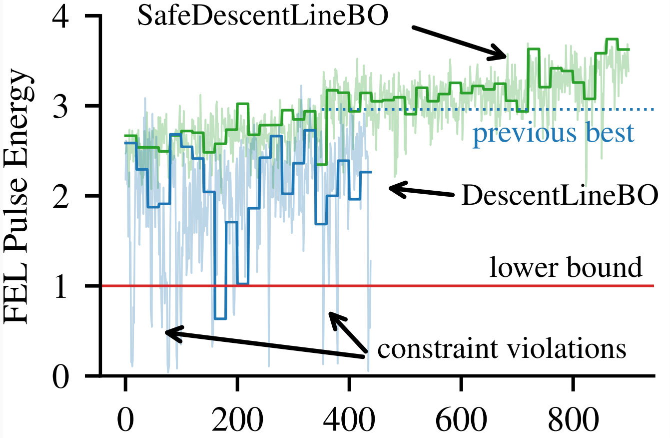
SwissFEL Tuning (24 parameters)



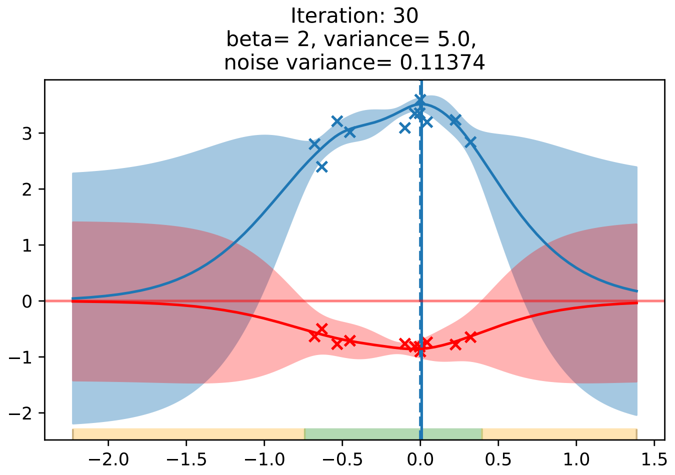
SwissFEL Tuning (24 parameters): Parameter Solutions



SwissFEL Tuning (24 parameters)



SwissFEL Tuning (Slice Plot)



Conclusion & On-Going Work

We found:

▷ Bayesian Optimization is feasible for safe tuning

▷ Relatively complex to set up

▷ CMA-ES often competitive performance, simpler to set up, but does not explicitly take constraints into account.

On-Going:

▷ Use new tools on startups

▷ Make GUI (more) production safe

▷ Code-base relatively complex, difficult to maintain? Merge with OCELOT?