

# Geoff: Applications and Developments in 2024 (and 2025)

- Penny Madysa<sup>1</sup> Hendrik Alsmeier<sup>2</sup> Sabrina Appel<sup>1</sup> Martin Bajzek<sup>1</sup>  
 Rolf Findeisen<sup>2</sup> Sebastian Hirt<sup>2</sup> Daniel Kallendorf<sup>2</sup> Erika Kazantseva<sup>1</sup>  
 Eric Lenz<sup>2</sup> Maik Pfefferkorn<sup>2</sup> Stephane Pietri<sup>1</sup> Christoph Reinwald<sup>2</sup>  
 Helmut Weick<sup>1</sup>

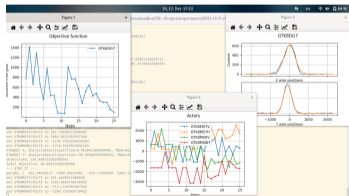
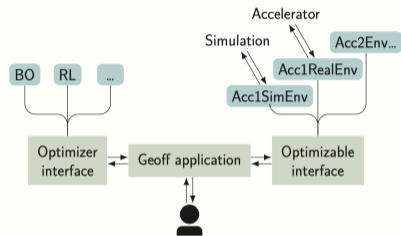
RL4AA Workshop,  
4 April 2025

---

<sup>1</sup>GSI, Darmstadt

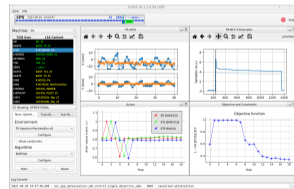
<sup>2</sup>TU Darmstadt

- <https://gitlab.cern.ch/geoff/>
- Python-based framework to **harmonize libraries** for optimization, optimal control and RL
- configures and runs agents and environments
- supports plotting, data logging, and is **agnostic of the controls system**
- flexible, extensible, maintainable



← runs in the terminal ...

... and in a GUI! →



- migrated from *Gym* to *Gymnasium*
- extensive rewrite, lots of docs
- core packages & GUI app updated
- environments updated by machine experts via migration guide →
- API for Bayesian optimization (`CustomOptimizerProvider`)

## Next Steps:

- update to Gymnasium 1.0, NumPy 2
- PoC use at CEA Saclay, which doesn't have a unified controls system

Acc-Py Documentation server » cermml-coi 0.9.2 documentation » User Guide » [Migration Guide for COI v0.9](#) previous | next | modules | index

Theme  |

### Table of Contents

Migration Guide for COI v0.9

- Minimum Python Version is now 3.9
- New Gymnasium Step API
- New API for Single-Objective Optimization
- Changes to Environment Reseeding
- New Rendering API
- Render Mode Changes
- New Registration API
- Revamp of the Abstract Base Classes
- Miscellaneous Minor Breaking Changes
- Deprecations

Previous topic  
Optimizing Points on an LSA Function

Next topic  
Code Examples

This Page  
Show Source

## Migration Guide for COI v0.9

**See also:**

[Changelog for Unreleased Version](#)  
List of all changes, including backwards-compatible ones, released in this version.

[v21 to v26 Migration Guide](#)  
Corresponding migration guide of the Gymnasium package.

Version 0.9 of this package is the first to be based on [Gymnasium](#) rather than its predecessor, [Gym](#). The new package changes several core aspects of its API, which this package has to adapt to. At the same time, the opportunity has been taken to introduce more breaking changes than in several previous major version bumps.

This page collects all changes that are considered breaking and how to upgrade your code to the new version.

### Minimum Python Version is now 3.9

Both the COI and Gymnasium have **dropped support for Python 3.7**. Gymnasium now requires at least Python 3.8, the COI require at least Python 3.9 [1]. While the two versions are largely backwards-compatible in the context where the COI typically get used, Python 3.9 has added a **lot of deprecation warnings**. We strongly recommend to run your code in [Development Mode](#) (added in Python 3.7).

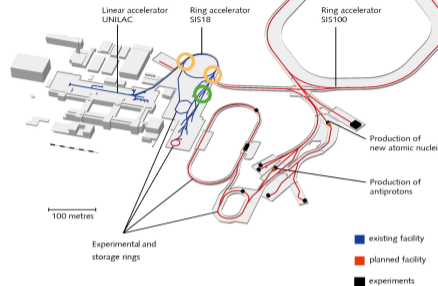
Please refer to their respective release notes to see if anything you require has been removed:

- [What's New In Python 3.8](#)
- [What's New In Python 3.9](#)

[1] The reason for the discrepancy is that Acc-Py never supported Python 3.8 and has already added support for Python 3.11. We see little value in supporting Python 3.8, but please contact us if you require it.

## Automation & Optimization with Python

- injection loss minimization (SIS18)
- beam steering (TK, X2)
- closed-orbit correction for non-standard optics (SIS18)
- slow extraction loss minimization (SIS18)
- beam steering and focusing (FRS)



## Methods of Interest

- BOBYQA, Bayesian optimization (BO)
- physics-informed BO
- multi-objective Bayesian optimization (MOBO)
- mixed-method BO + extremum seeking
- data-driven model predictive control (MPC)
- reinforcement learning (RL)

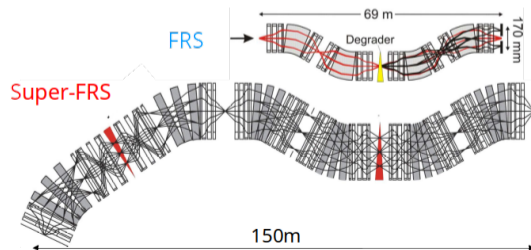
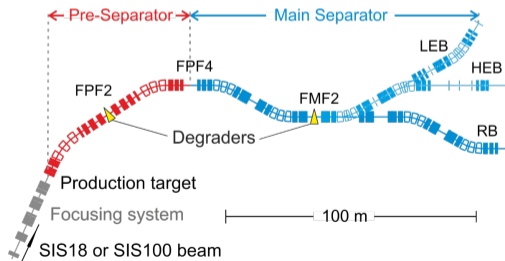
## Work power

- one scientific staff financed by EURO-LABS
- one master student from PLUS (Salzburg)
- several master/PhD students from TUDa (Darmstadt)

# Fragment Separator (FRS) and Super-FRS

- investigates nuclear structure of exotic nuclei
- produces, separates and identifies exotic nuclei
- sends them to downstream experiments or storage ring
- Super-FRS: higher acceptance, more complex (4× more magnets), gain factors between 1000 ( $^{12}\text{C}$ ) and 7500 ( $^{132}\text{Sn}$ )
- automation of operational tasks essential

	$B\rho_{\text{max}}$	$\Delta p/p$	$\Delta\Phi_x, \Delta\Phi_y$	resolving power	gain factor	
					$^{19}\text{C}$	$^{132}\text{Sn}$
FRS	18 Tm	1.0 %	$\pm 13, \pm 13$ mrad	1500	1	1
Super-FRS	20 Tm	2.5 %	$\pm 40, \pm 20$ mrad	1500	5	10
				including primary rate	250	20 000



## Motivation

- scale: manual setup too time consuming
- complexity: different optical modes per user
- accuracy: scaling for same optics but different  $B\rho$  not accurate enough

## Observables

- profile grid histograms
- phase space spectra from tracking detectors

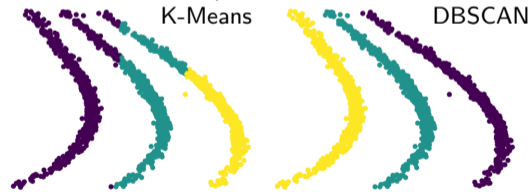
## Machine Interface

- LSA database (trim steerers, magnets)
- FESA (SIS18 monitoring)
- Go4 (experiment-side TPC, current grids)

## Optimization Goals

- 0<sup>th</sup> order: center the beam  
(vertical steerers, main dipoles)
- 1<sup>st</sup> order: set focus, dispersion (quadrupoles)
- 2<sup>nd</sup> order: minimize aberrations (sextupoles)
- 3<sup>rd</sup> order: minimize aberrations  
(octupoles, S-FRS only)

Figure: Track classification (simulation with disabled sextupoles)



(D. Kallendorf, master's thesis)

# FRS and Super-FRS: Observables

## Central spot:

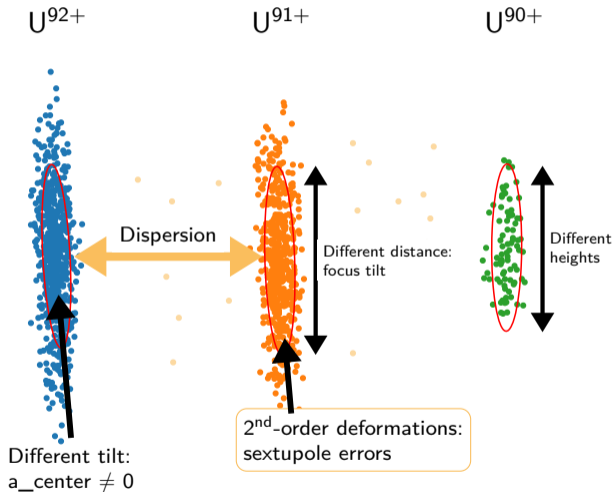
- Twiss parameters
- Non-dispersive 1<sup>st</sup>-order transfer matrix

## Outer spots:

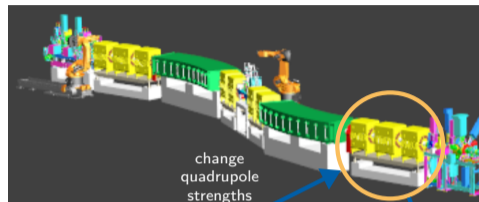
- 1<sup>st</sup>-/2<sup>nd</sup>-order dispersive transfer matrix elements

**Goal:** Bring the spectra as close as possible to desired 1st order parameters

⇒ optimize **direct observables**, not individual matrix elements!



## Illustration of FRS Dispersive Area

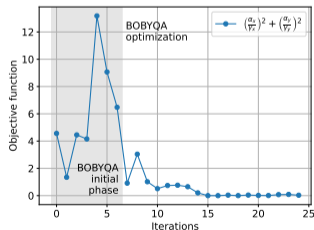


### Proof of principle:

1. **steer beams** at target
2. **tune quadrupoles** to focus beam (central spot upright)
3. **tune sextupoles** to remove focal-plane tilt and “banana” deformation

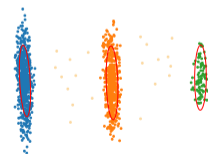
### Next step

- use other observables
- tune entire 1<sup>st</sup>-order optics
- include dispersion



### TPC Data Analysis

evaluation of beam spot slope



(M. Bajzek, E. Kazantseva, S. Pietri, H. Weick)

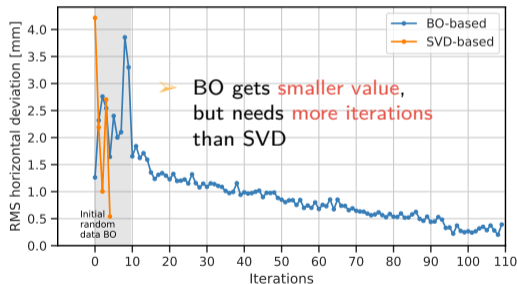
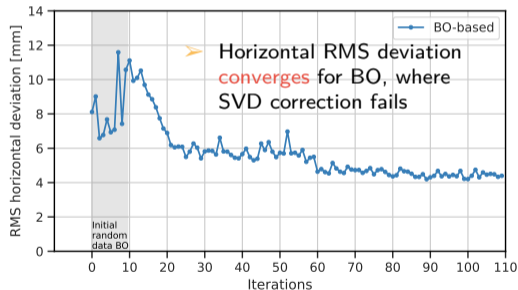


# SIS18 Closed-Orbit Correction

- broken-symmetry high-transition-energy SIS18 optics: **standard technique fails**
- BO provides complementary approach
- additionally handles **BPM noise** better

Current work:

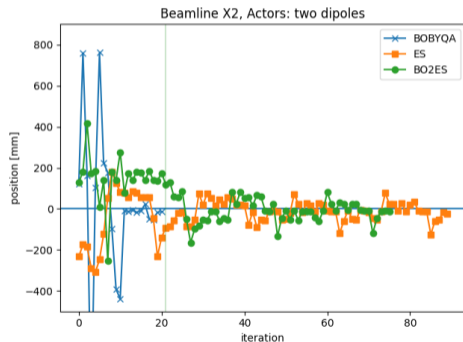
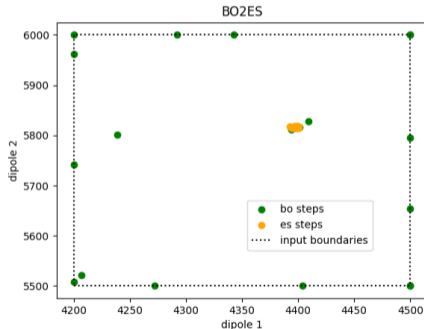
physics-informed covariance kernels for orbit model between BPMs



(V. Isensee, master's thesis)

# Exploration of Novel Optimizers at the X2 Beamline

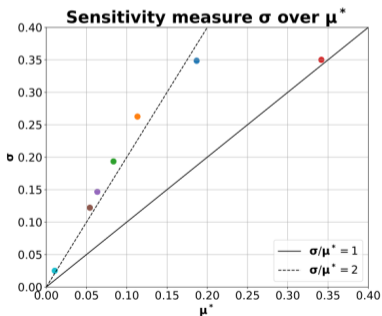
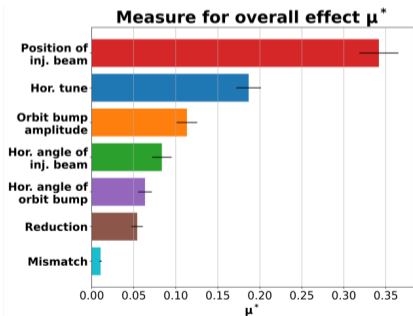
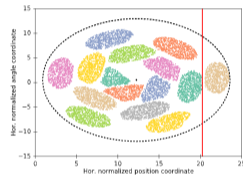
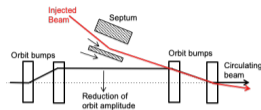
- goal: combine gradient-free global + gradient-based local optimization
- BO2ES: two phases of optimization
  1. Bayesian Optimization (efficient global exploration)
  2. Extremum Seeking (finetuning of best candidate)



(H. Alsmeier, R. Findeisen, S. Hirt, E. Lenz, M. Pfefferkorn)

# SIS18 Multi-Turn Injection

- Liouville's theorem: beam can only go into free phase space
- Competing goals:
  - maximize gain factor factor (reach space charge limit)
  - minimize losses at septum (cause vacuum degradation)
- MTI model implemented in **Xsuite** for fast tracking

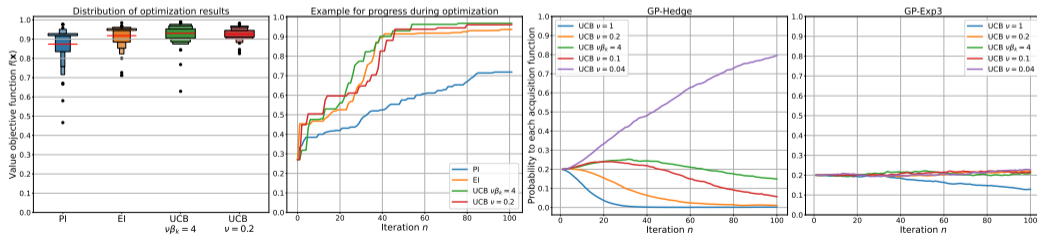
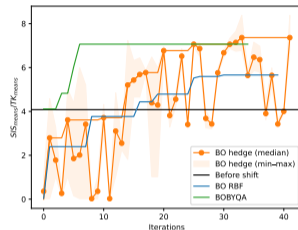


- beam position + tune most important
- all parameters have higher order dynamics (shown by  $\sigma/\mu^*$ )

(C. Reinwald, master's thesis)

# SIS18 Multi-Turn Injection

- various BO kernels and acquisition functions studied
- combine multiple answers to make BO more general
- Hedging* algorithm prefers most exploitative answer



(C. Reinwald, master's thesis)

- explore sample-efficient control algorithms
- based on XSuite simulation of SIS18 MTI
- for more information, see [poster](#) by Benjamin Halilovic!

**Robust Multi-Turn Injection at SIS18 via Gaussian Process Model Predictive Control**

**Benjamin Halilovic<sup>1</sup>**  
**Sabrina Appel<sup>1</sup>**  
**Simon Hirlander<sup>2</sup>**

**abstract**  
In advanced accelerator facilities like the FAIR in Darmstadt, ensuring stable and efficient multi-turn injection is vital for achieving high-intensity beams. However, conventional control methods often lack the adaptability needed to handle rapidly changing beam dynamics, leading to suboptimal performance.

**methodology**  
To address this shortcoming, a data-driven Gaussian Process Model Predictive Control (GP-MPC) framework is employed, leveraging real-time updates to capture and predict complex injection behavior more accurately. Additionally, we systematically also analyze the controller's behavior in response to noise in the incoming beam intensity and emittance, assessing its robustness against such perturbations.

**problem statement**  
**description**  
SIS18 injection limits FAIR intensity due to the vacuum degradation triggered by losses. MTI must fulfil 3-levels' objectives and benefits must not overlap, attempts to maximize injections while minimizing loss.

**simulation**  
SIS18 MTI model implemented in XSuite sensitivity analysis shows higher-order effects.

**results**  
**TE current**  
**particle noise scenario with fixed emittance noise**

**discussion**  
Our results demonstrate that GP-MPC reliably controls multi-turn injection in both noise-free and noise-affected scenarios. A measured injection current near 5000 was reached and integrated into the simulation, enabling GP-MPC's robustness under realistic, noise-induced conditions. High computational costs arise from the repeated inversion of covariance matrices.

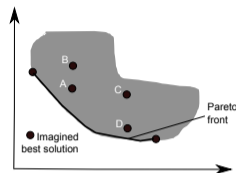
**contact**  
benjamin.halilovic@stud.gsi.de  
sabina.appel@gsi.de

**references**  
[1] Halilovic, B. and Appel, S. (2024) Robust multi-turn injection control via Gaussian Process Model Predictive Control. In Proceedings of the 2024 IEEE Conference on Decision and Control, 2024, pp. 1-6.  
[2] Halilovic, B. and Appel, S. (2024) Robust multi-turn injection control via Gaussian Process Model Predictive Control. In Proceedings of the 2024 IEEE Conference on Decision and Control, 2024, pp. 1-6.  
[3] Halilovic, B. and Appel, S. (2024) Robust multi-turn injection control via Gaussian Process Model Predictive Control. In Proceedings of the 2024 IEEE Conference on Decision and Control, 2024, pp. 1-6.

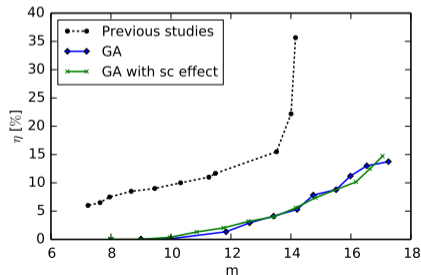
(S. Appel, B. Halilovic)

# Multi-Objective Bayesian Optimization

- competing goals: **maximize** number of injections, **minimize** loss
- **Pareto front**: set of solutions strictly better than others
- prior art: simulation with genetic algorithms (S. Appel)



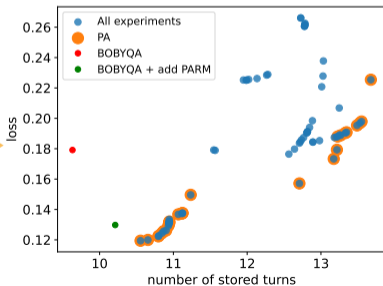
## Simulation with Genetic Algorithms



Optimization of injector beam line is missing



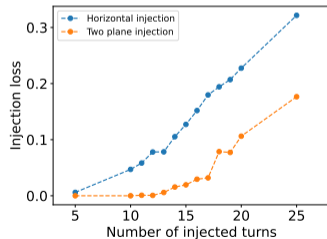
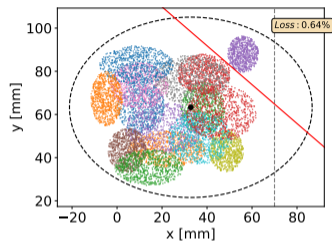
## Measurements



# Possible upgrade for SIS18 injection

- SIS18 uses wide range of ions (from protons to uranium)
  - ⇒ injection via charge exchange is impractical
- possible upgrade: **two-plane injection**
  - tilt injection septum
  - ⇒ use vertical phase space
  - ⇒ overlap beamlets without violating Liouville's theorem
- Drawbacks:
  - requires replacement of injection line and area
  - doubles number of injection parameters

(S.Appel)



## Summary

- successful proof of concept at GSI, endorsement from leadership
- Gymnasium migration completed!
- widespread use at CERN

## Outlook

- ecosystem keeps shifting: Gymnasium  $\rightarrow$  1.0, NumPy  $\rightarrow$  2.0
- make dynamic loading of environments more convenient
- GSI contributions to GUI app development
- more experiments at GSI and CEA Paris-Saclay