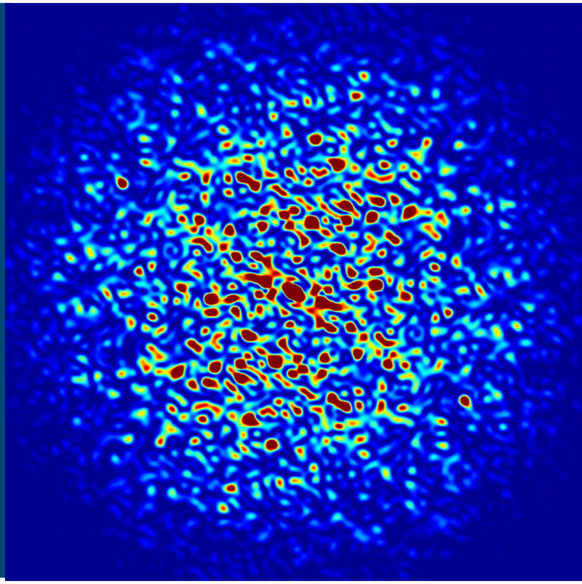


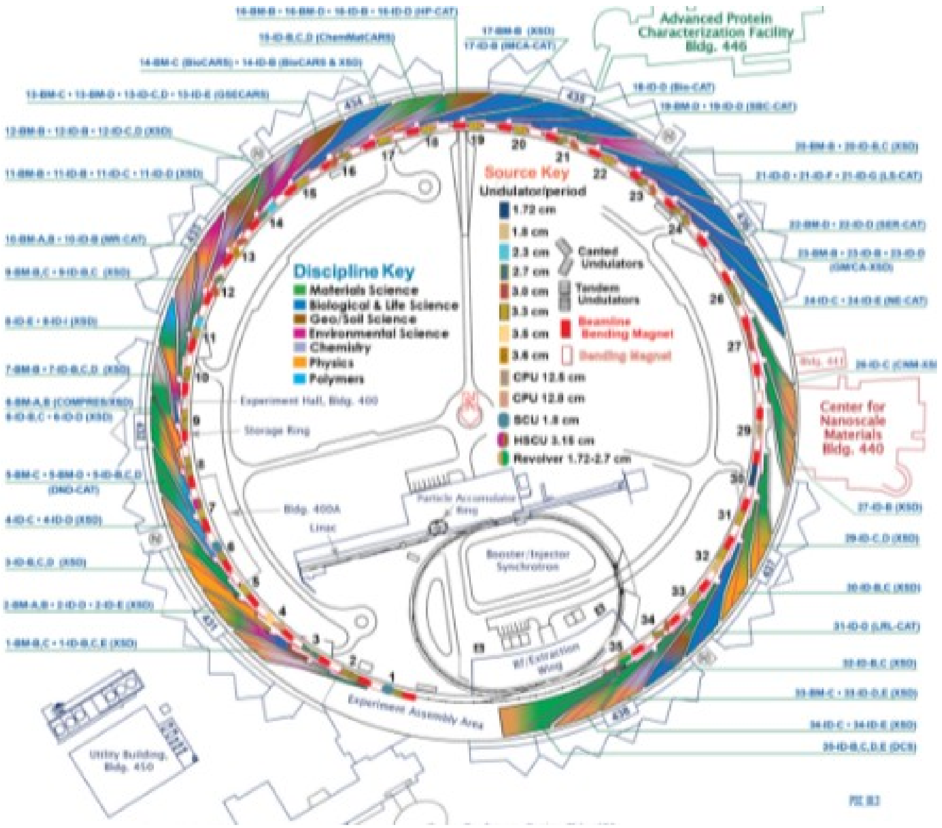
Simultaneous optimization of dynamic acceptance and lifetime for APS

Yipeng Sun, Louis Emery, Hairong Shang, and Xiaobiao Huang
Advanced Photon Source, Argonne National Laboratory

AI for Particle Accelerators, X-ray Beamlines, and Electron Microscopy Workshop
Nov. 1 - 3, 2021, at Argonne National Laboratory

Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

Advanced Photon Source (APS)



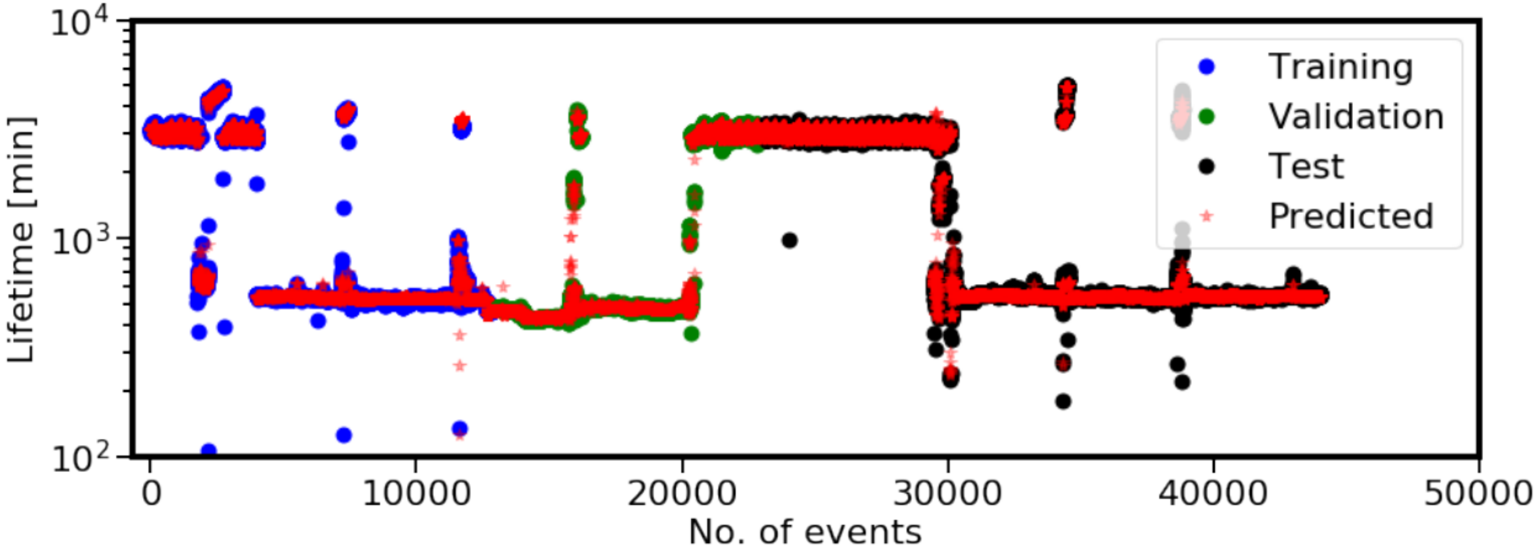
Lemont, IL
Commissioned 1995, 7GeV, 102mA
3nm emittance, 1104m

~68 beamlines
More than ~5000 users

Figures from: https://www.aps.anl.gov/

Storage ring beam lifetime

Storage ring beam lifetime, is defined as the time for beam decaying to 1/e of initial charge. Particles are lost due to physical apertures, where motion excited by scattering, photon emission, **nonlinear beam dynamics**

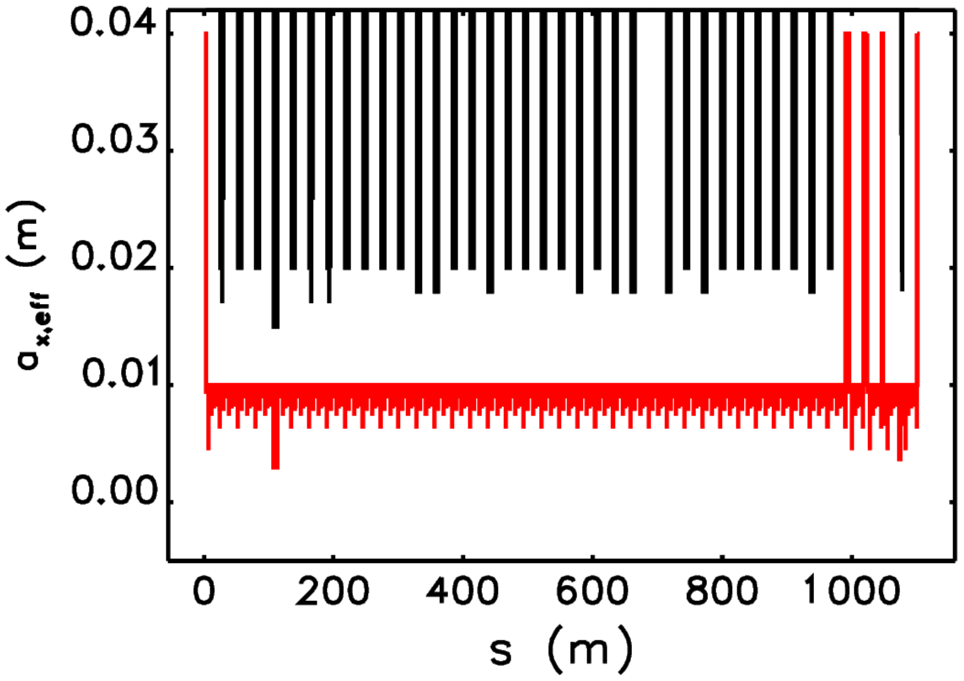


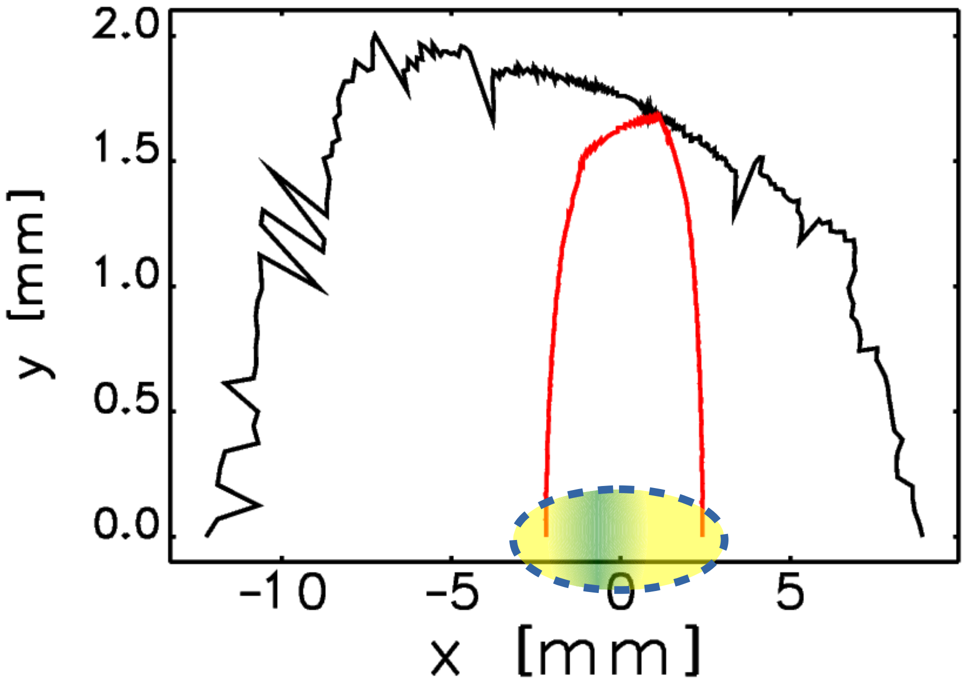
Artificial neural network prediction of APS beam lifetime [1], operation data of 3 months

- Total stored beam current; Number of RF buckets filled; Vacuum pressure;
- Transverse emittance ratio; RF gap voltage; energy loss from insertion devices;
- **Linear chromaticity in both transverse planes (nonlinear beam dynamics)**.

[1] Yipeng Sun, TUPAB060, in IPAC2021

Storage ring dynamic acceptance, critical for injection efficiency to fill the storage ring

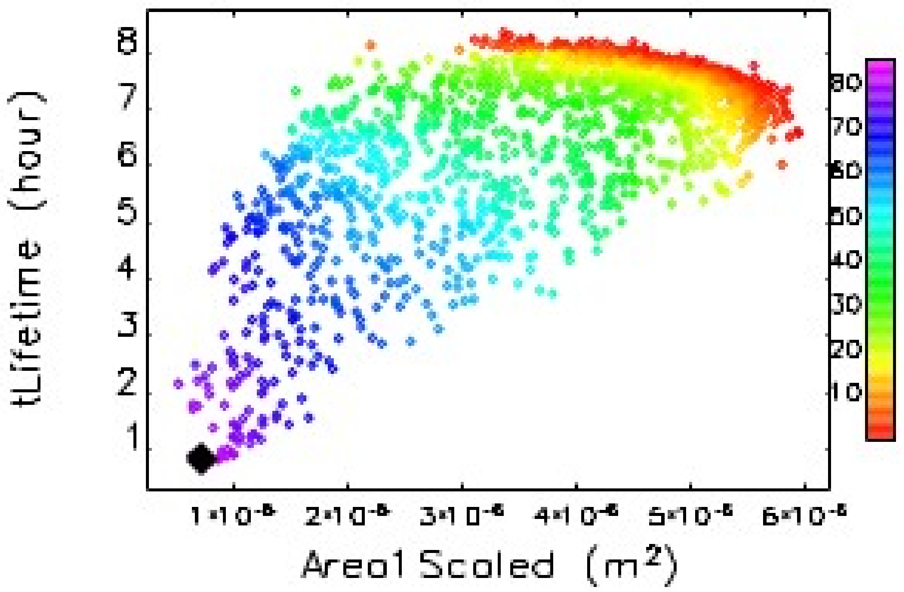


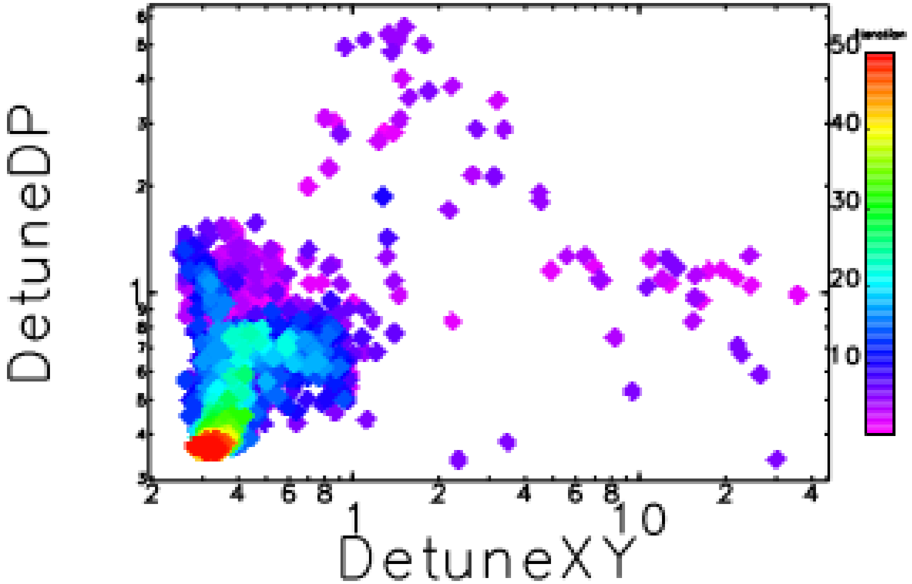


Stronger magnets/undulators requires smaller horizontal/vertical physical apertures
Red: APS-U; Black: APS

Dynamic acceptance is determined by:
dynamic aperture (beam dynamics)
Plus physical apertures
Red: APS-U; Black: APS

Nominal offline optimizations approach



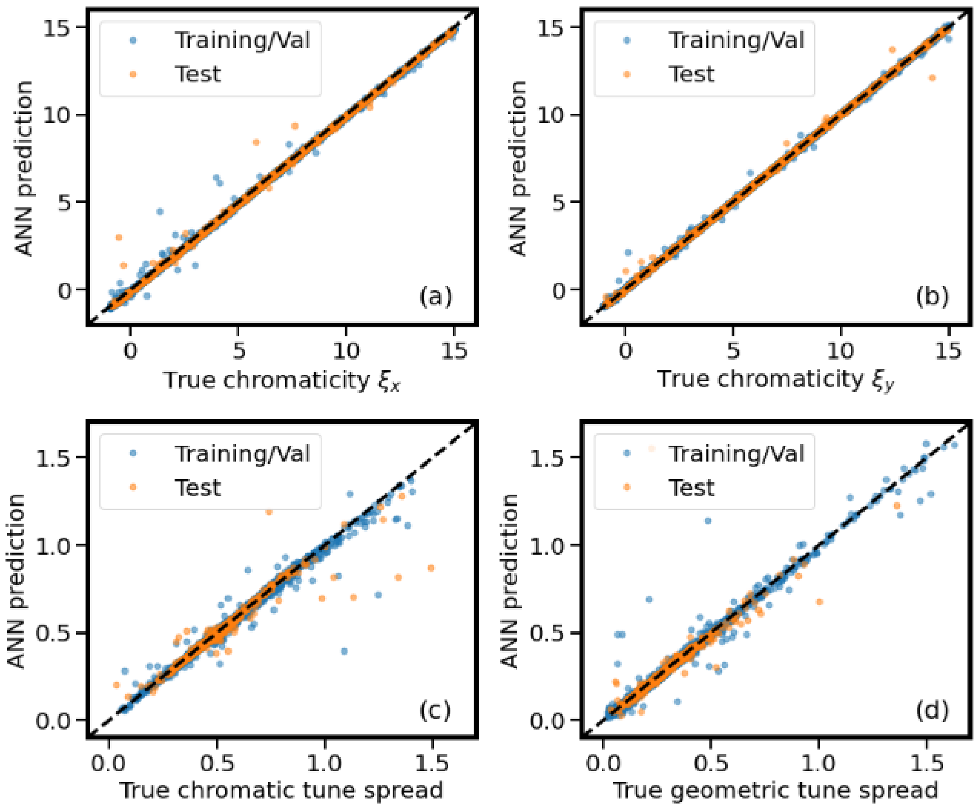


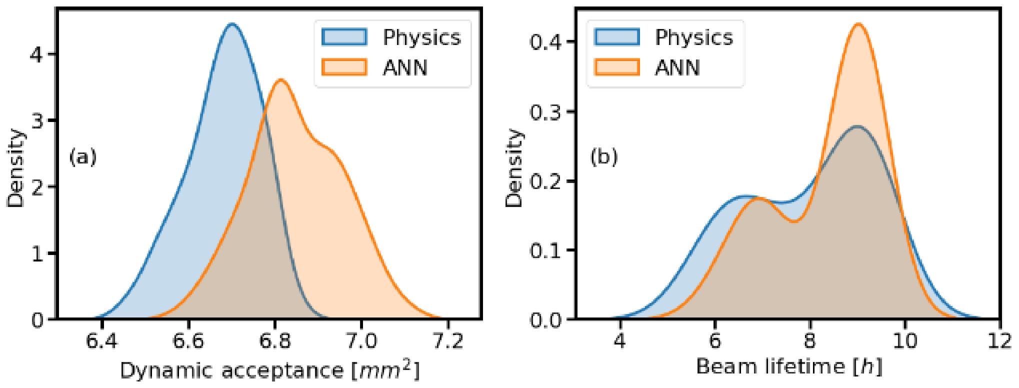
Genetic algorithm to optimize local momentum and dynamic acceptance, chromatic tunes footprint
M. Borland et al., ANL/APS/LS-319 (2010).

Particle swarm optimizations to optimize detuning of on- and off-momentum bunches
Y.-P. Sun et al., in NAPAC2016 (2016).

Offline optimized solutions verified by APS operations, with good agreement on lifetime and injection eff.

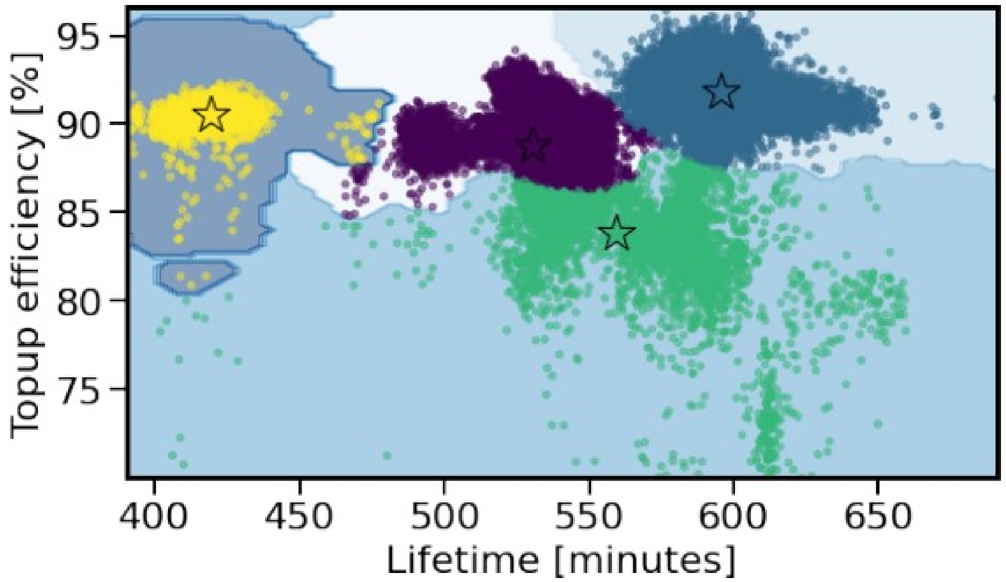
Neural network as surrogate for physics model

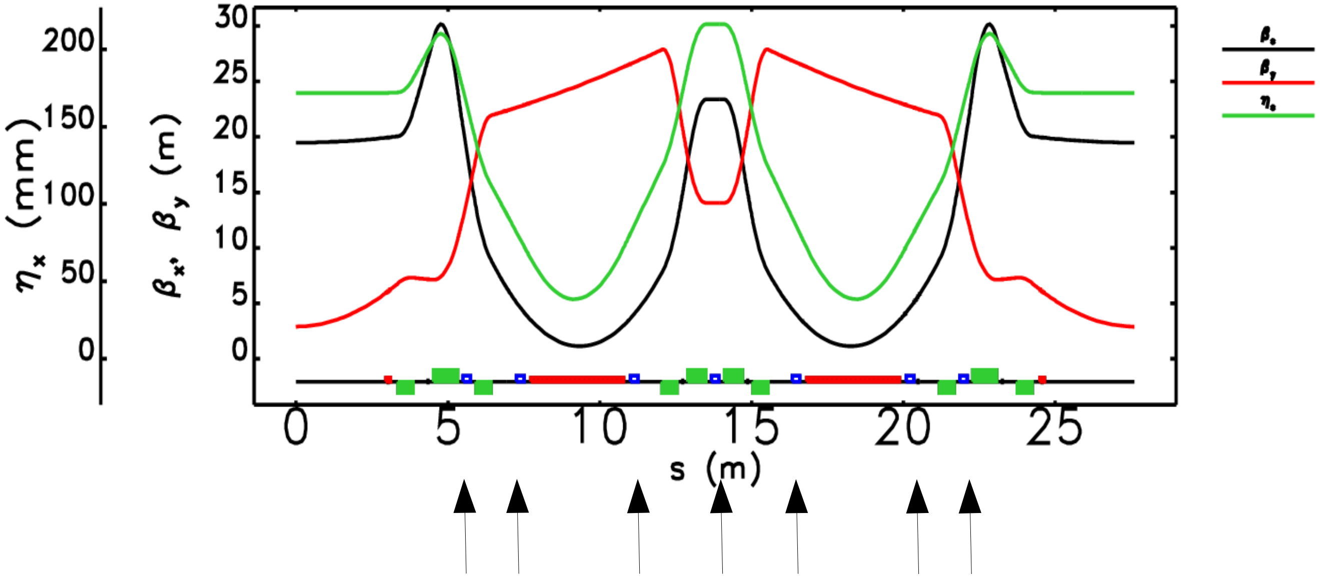




- Neural network may accurately represent the complex physics model
- The ANN model optimized solution slightly outperforms the best solution from extremely time-consuming physics based optimizations
- May not work well for small training dataset

Optimization objectives and variables





APS operation data of lifetime and top-up efficiency, 24 bunch mode

Unsupervised learning [1] by K-means clustering and naive Bayes classifier, showing trade off between two objectives

[1] Yipeng Sun, TUPAB060, in IPAC2021

There are many variables affecting beam lifetime and injection efficiency.

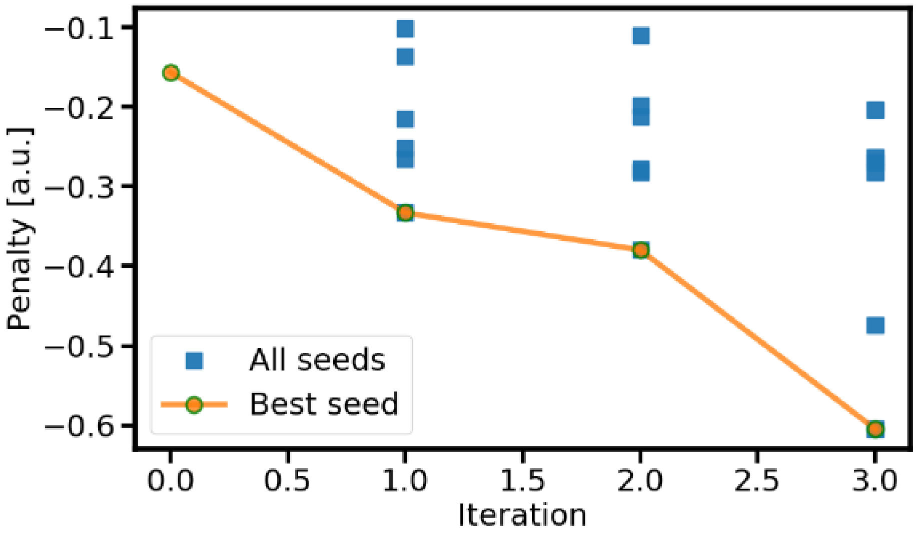
In this work, focus on nonlinear beam dynamics (configurations of nonlinear sextupole magnets, 7 per cell for APS). Employing null space of chromaticity response matrix for fixed linear chromaticity.

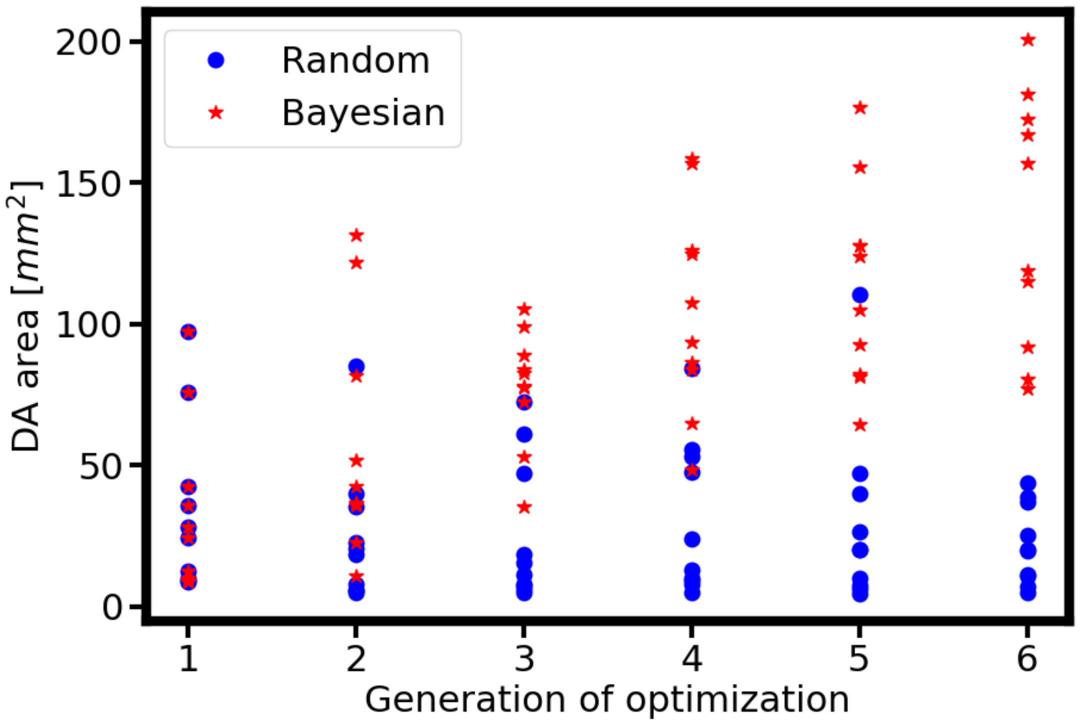
Why Bayesian optimization and online optimization

Online optimization
- Model-independent; there is always difference between ideal model and real machine, due to errors etc.
- Some instantly observable parameters taking long computing time

Bayesian optimization
- Grid/Random search suitable for simulation based approach, given enough computing resources
- Bayesian optimization providing directed search, sampling candidates near potential global optimum; may be preferred for online optimizations when objectives are expensive to evaluate
- Gaussian process regression: working well on a relatively small training dataset; easier to be interpreted; the predictions contain information on uncertainties

Why Bayesian optimization and online optimization



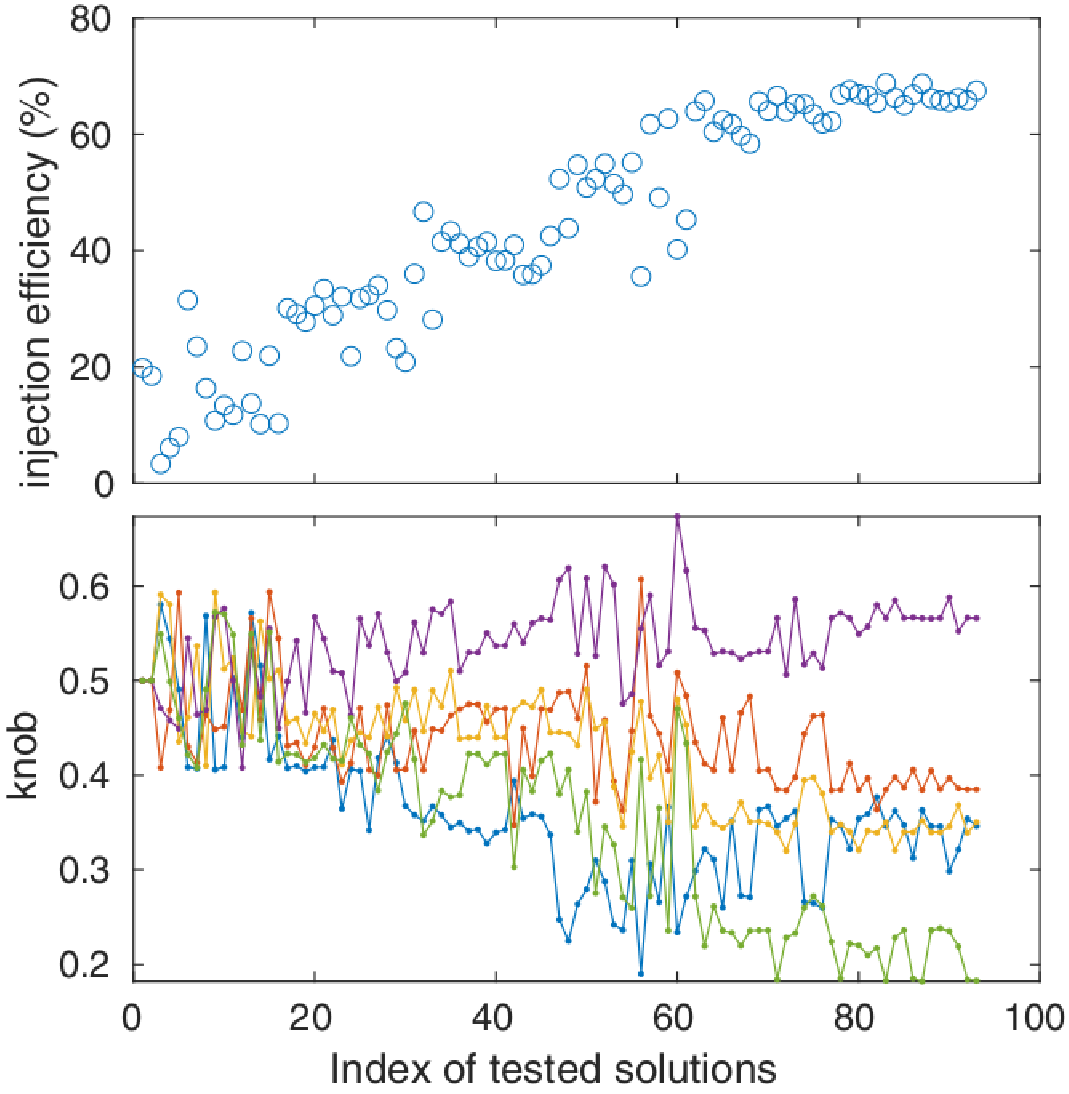


Online Optimizations [1] on APS injection efficiency of a single bunch, by tuning 28 families of sextupole magnets in sectors hosting the minimum physical aperture and the four injection kickers, random optimization.

Random optimization, compared to Bayesian optimization, on APS dynamic acceptance, 10 seeds for each of 6 generations. Bayesian optimization selected 10 seeds out of 300 candidates.

[1] Yipeng Sun, TUPAB058, in IPAC2021

Online optimization of dynamic acceptance [1]

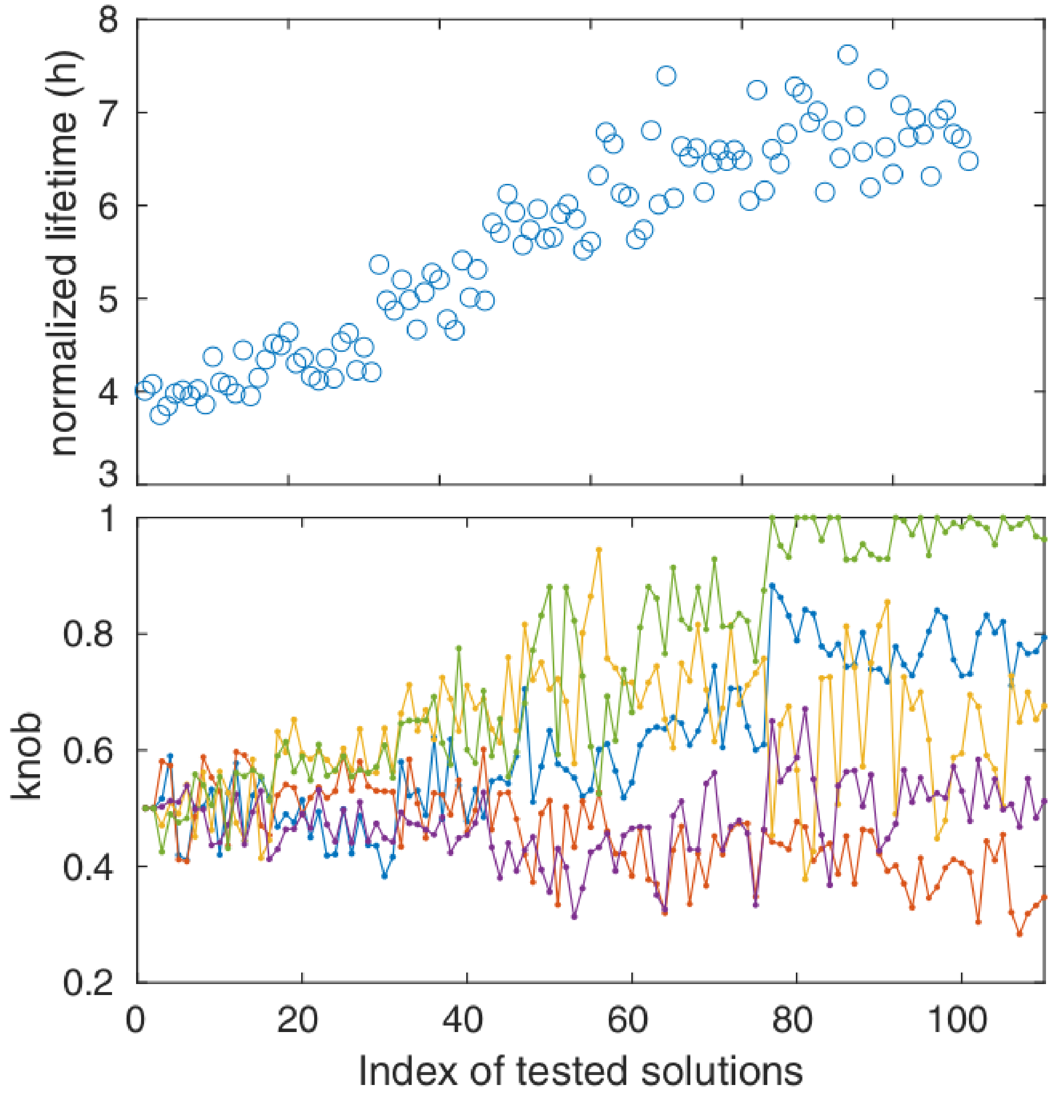


- Null space matrix (5 by 7) for fixed linear chromaticity.
- Sextupole strength change up to 30%
- Kicker bump size was decreased to 55% to lower the initial injection efficiency to 32%
- Measurement noise 1%

- Random sampling with genetic algorithm for diversity
- Gaussian process regression to evaluate random samples
- Acquisition function employs expected improvement as criteria
- 6 generations in optimization, 15 seed for each generation

[1] Xiaobiao Huang, Louis Emery, Hairong Shang, Yipeng Sun, PRAB 24, 082802 (2021)

Online optimization of beam lifetime [1]

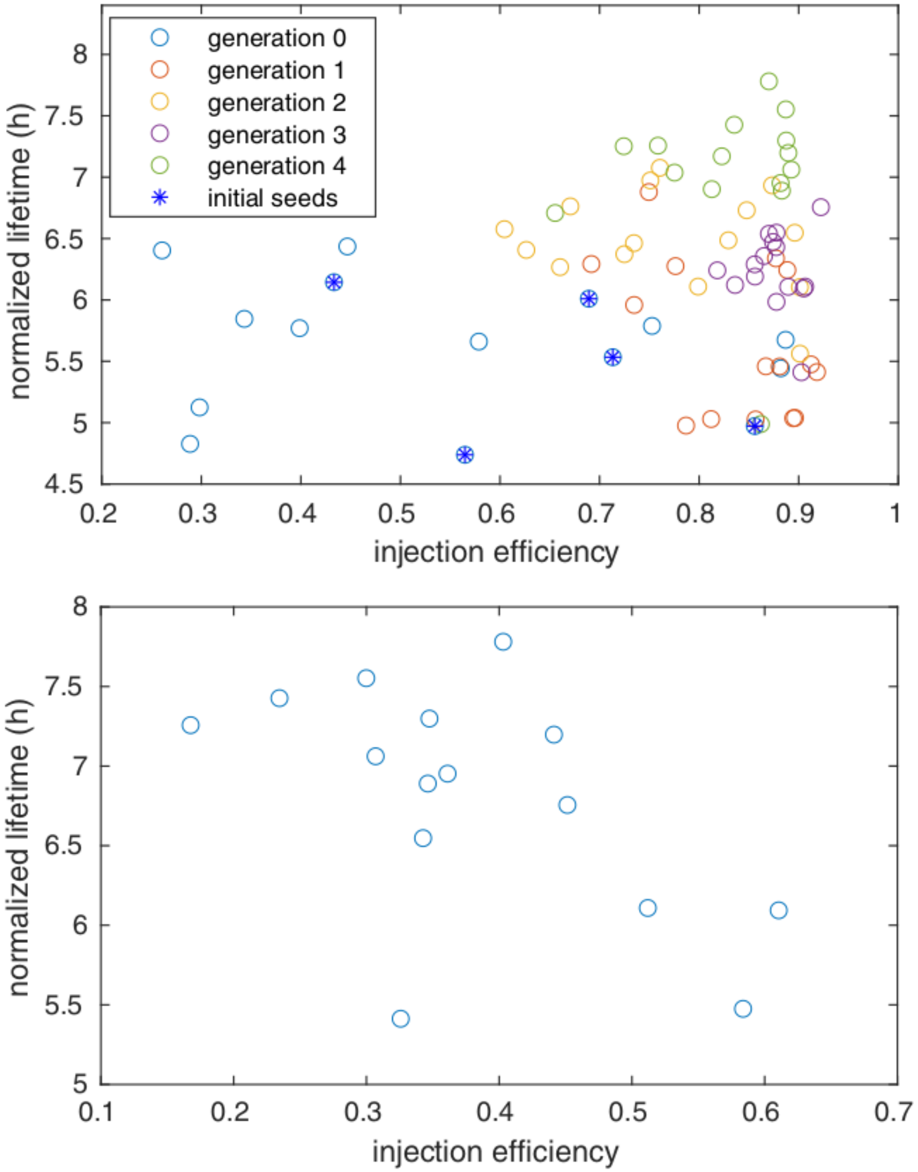


- Same optimization procedure and algorithms

- Start with solution of DA optimization
- 24 mA in 6 bunches, 20 seconds beam current data with linear fit
- Lifetime normalized by beam current and coupling ratio
- Measurement noise 1%

[1] Xiaobiao Huang, Louis Emery, Hairong Shang, Yipeng Sun, PRAB 24, 082802 (2021)

Simultaneous online optimizations of dynamic acceptance and beam lifetime [1]



- Start with solution of previous optimizations
- Initial procedure: evaluate seed one by one
- Optimized procedure: for 15 seeds of each generation, evaluate lifetime without beam dump; then evaluate injection efficiency

- Top: optimization results with closed kicker bump
- Bottom: 15 best seeds, re-measured with open kicker bump

[1] Xiaobiao Huang, Louis Emery, Hairong Shang, Yipeng Sun, PRAB 24, 082802 (2021)

Acknowledgment

- Michael Borland, Ryan Lindberg, Vadim Sajaev, Bob Soliday, Aimin Xiao, C.Y. Yao and other AOP group members for discussions
- Some simulations used PELEGANT on weed cluster at APS
- Several Python libraries (*NumPy*, *SciPy*, *scikit-learn*) are employed for artificial neural network algorithms development, naive Bayes classifier algorithm development, and data processing. The figures are generated using the *matplotlib* libraries in the Python code, and *sddsplot*.
- Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

Thank you for your attention

Copyright