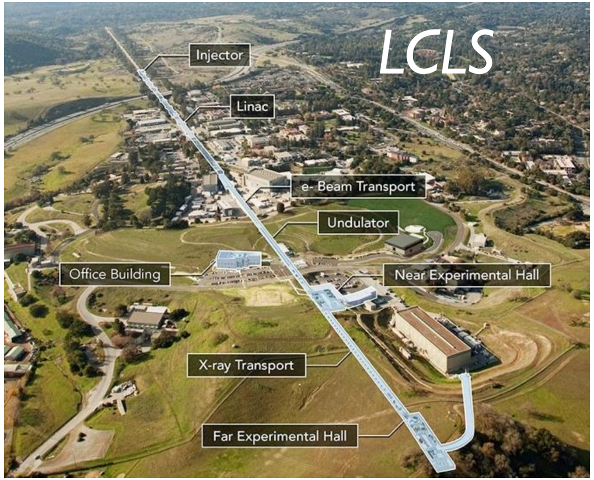
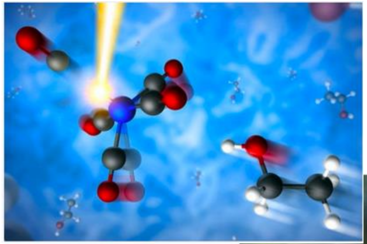
ICAP'24
October 5, 2024

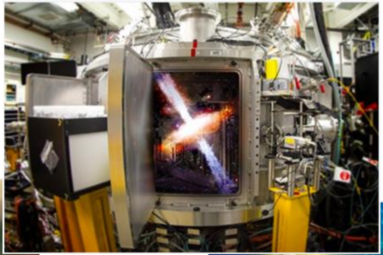
Integration of system models and machine learning for online optimization and characterization of accelerators

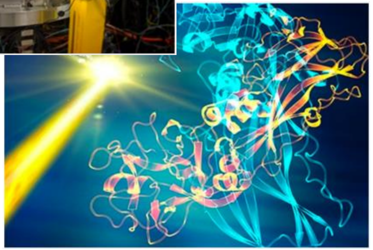
Auralee Edelen
Machine Learning Department Head, Accelerator Research Division
SLAC National Accelerator Laboratory
edelen@slac.stanford.edu

R. Roussel, D. Ratner, D. Kennedy, Y. Yazar, E. Cropp, C. Mayes, J. Bellister, Z. Zhu, Z. Zhang, C. Emma, S. Miskovich, W. Neiswanger,, T. Boltz, J.P. Gonzalez-Aguilera, and many many other collaborators





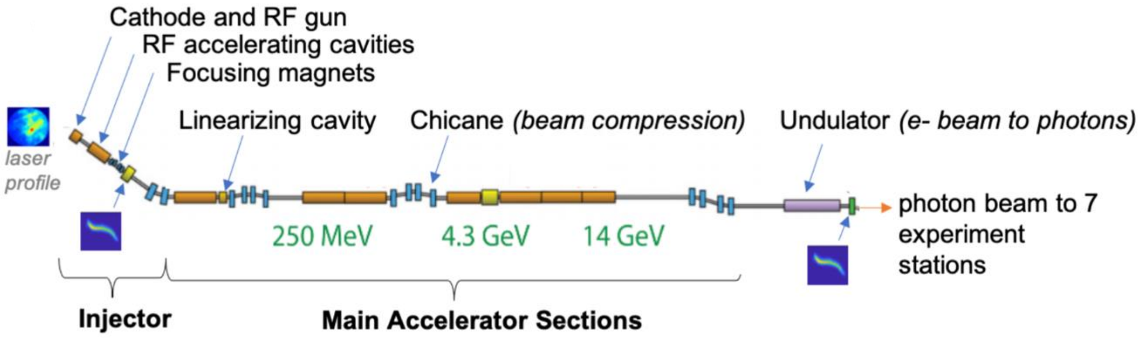


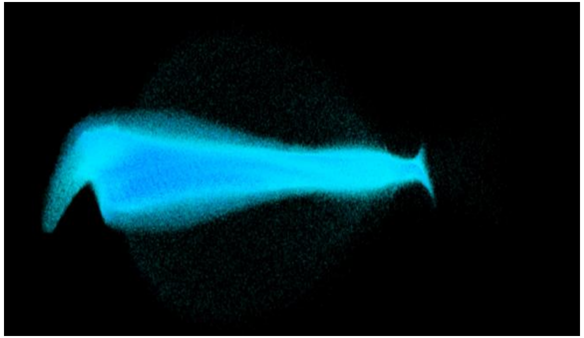


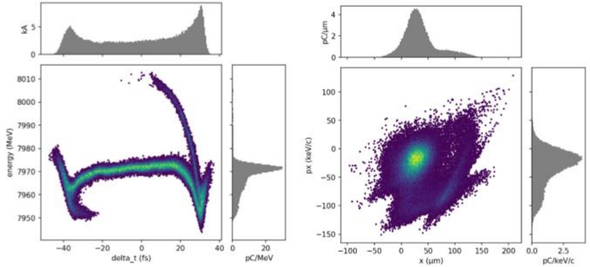
beam duration, average energy, energy spread, pulse intensity adjusted for each experiment

https://lcls.slac.stanford.edu

Detailed beam phase space customization required for different experiments





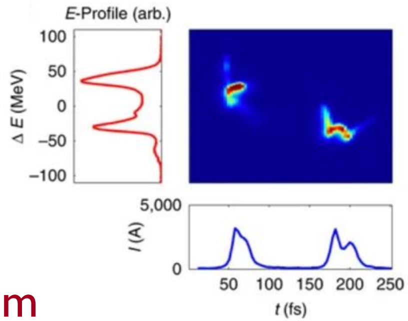


Beam exists in 6-D position-momentum phase space

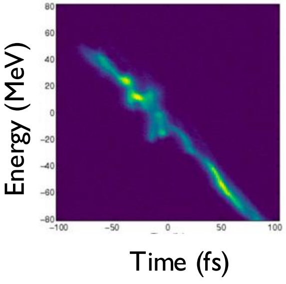
Incomplete information: measure 2-D projections or reconstruct based on perturbations of upstream controls (e.g. tomography)

Dozens-to-hundreds of controllable variables and hundreds-of-thousands to monitor

Increasingly dynamic control needed during experiments



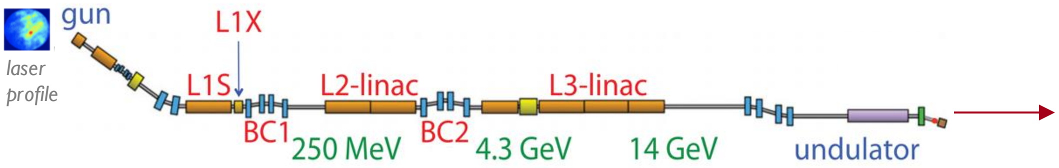
A. Marinelli, et al., Nat. Commun. 6, 6369 (2015)



A. Marinelli, IPAC'18

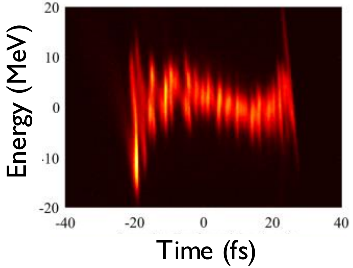
Nonlinear, high-dimensional optimization/control problem

wide spectrum of tuning needs

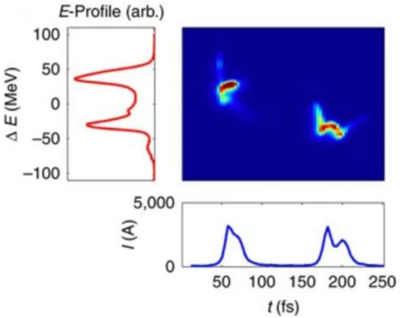


photon beam to 7 experiment stations

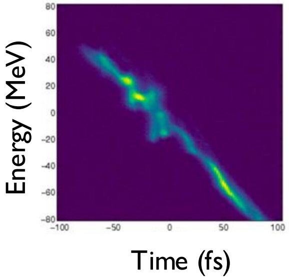
J. Qiang et al, PRAB (2017)



A. Marinelli, et al., Nat. Commun. 6, 6369 (2015)



A. Marinelli, IPAC'18



Rapid beam customization

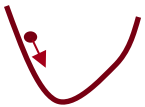
Achieve new configurations + unprecedented beam parameters

Fine control to maintain stability within tolerances

Tuning approaches leverage different amounts of data / previous knowledge → suitable under different circumstances

less ← assumed knowledge of machine → more

Model-Free Optimization

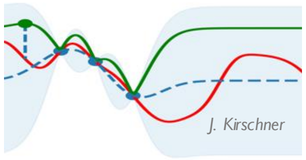


Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

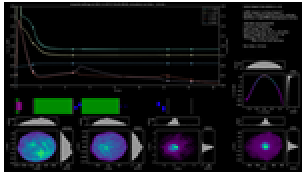


Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



→ provide initial guess (i.e. warm start)
→ provide insight to operators
→ model-based control

ML system models +
inverse models

General strategy: start with sample-efficient methods that do well on new systems, then build up to more data-intensive and heavily model-informed approaches.

Tuning approaches leverage different amounts of data / previous knowledge → suitable under different circumstances

less ← assumed knowledge of machine → more

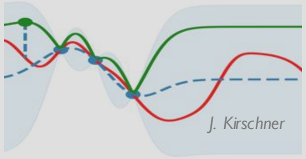
Model-Free Optimization

Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

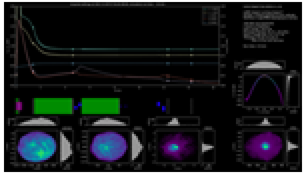


Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



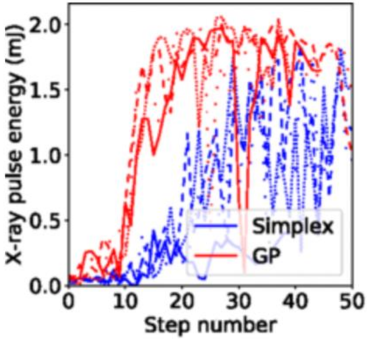
→ provide initial guess (i.e. warm start)
→ provide insight to operators
→ model-based control

ML system models + inverse models

General strategy: start with sample-efficient methods that do well on new systems, then build up to more data-intensive and heavily model-informed approaches.

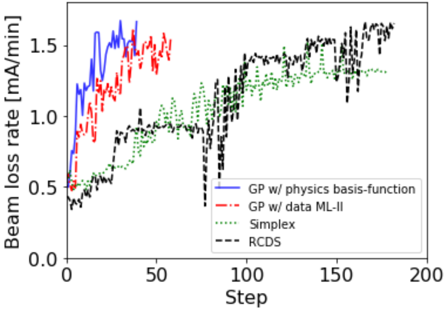
Many successes with Bayesian Optimization *(+ algorithmic improvements)*

FEL pulse energy tuning at LCLS (w/ physics-based kernel)



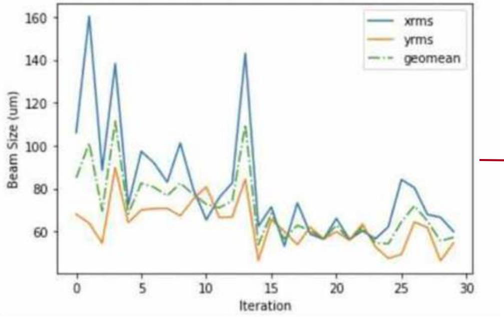
Duris et. al. PRL , **2020**

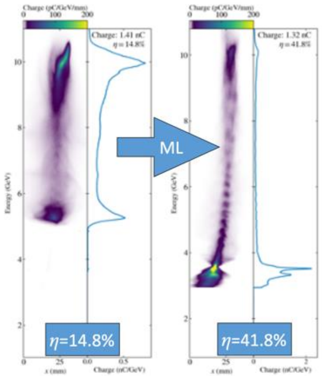
Loss rate tuning at SPEAR3 (w/ physics-based kernel)



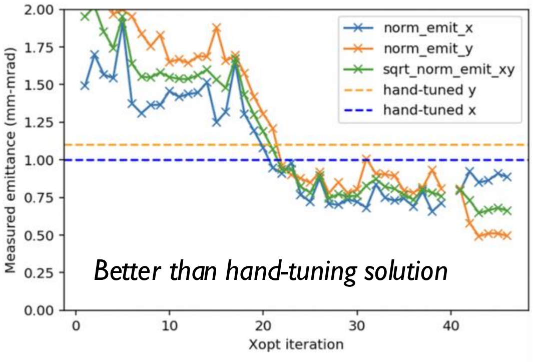
Hanuka et. al. PRAB , **2021**

Sextupole tuning at FACET-II
2x efficiency of acceleration in plasma

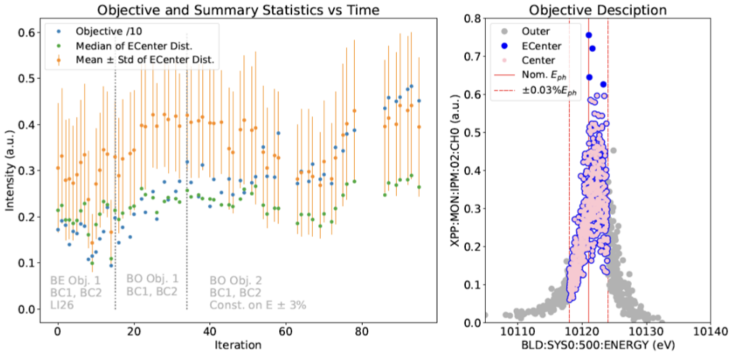




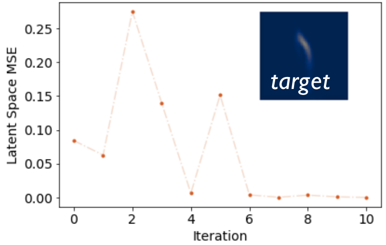
Emittance tuning for LCLS-II injector

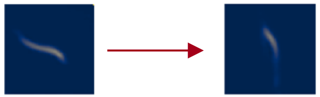


Tuning on monochrometer signal



Longitudinal phase space tuning on LCLS





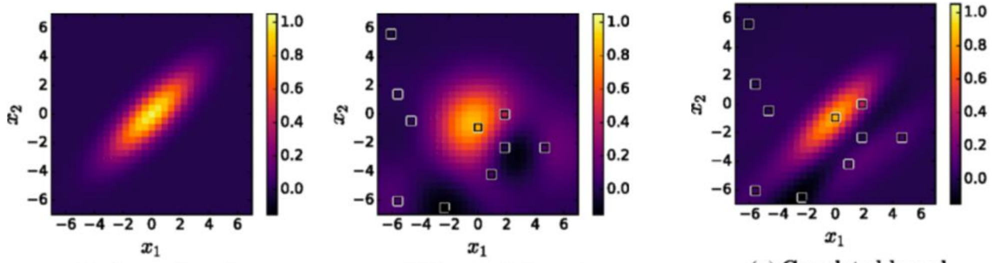
Algorithms being implemented/distributed in Xopt: https://github.com/xopt-org/Xopt

Comprehensive review of BO for accelerators: R. Roussel, et al., PRAB (2024) https://journals.aps.org/prab/abstract/10.1103/PhysRevAccelBeams.27.084801

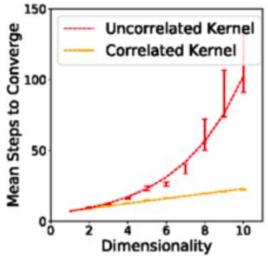
Physics-Aware Bayesian Optimization: Correlated Kernel

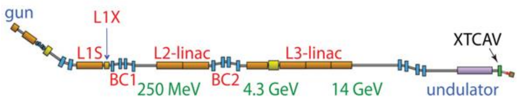
J. Duris et al., PRL, 2020
A. Hanuka, et al., PRAB, 2021

→ Design Gaussian Process kernel from expected correlations between inputs (e.g. quadrupole magnets)



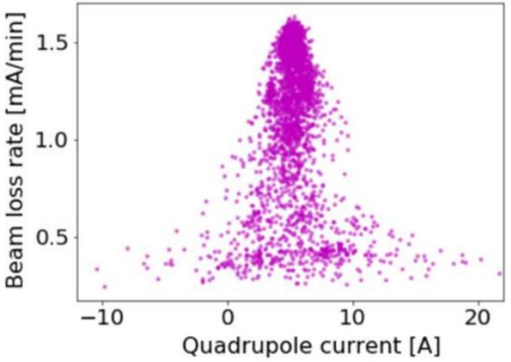
(a) Ground truth (b) Isotropic kernel (c) Correlated kernel

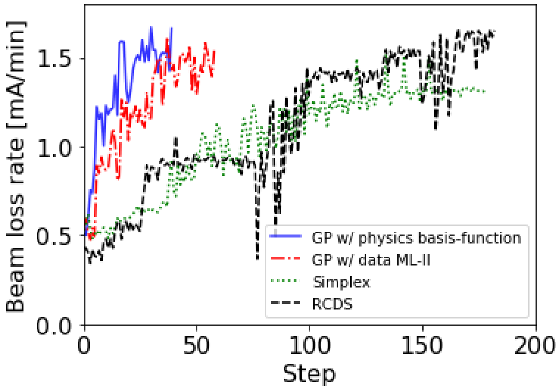


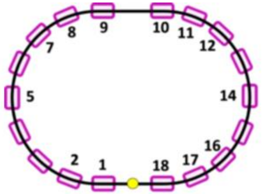


FEL tuning @LCLS

→ Take the Hessian of model at expected optimum to get the correlations







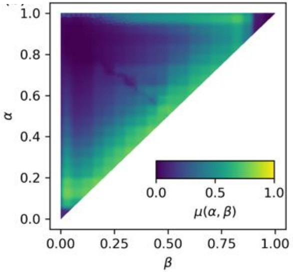
vertical emittance tuning @SPEAR3

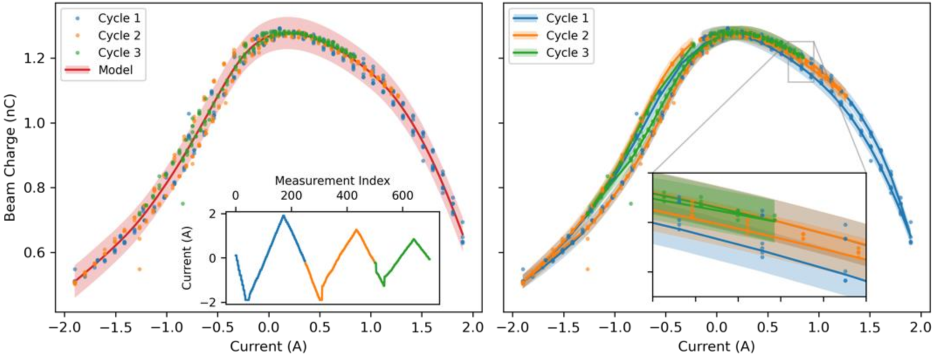
No measured data needed ahead of time, just a physics model of system

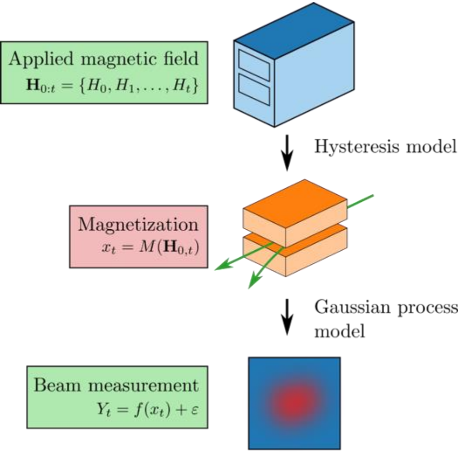
Including correlation between inputs enables increased sample-efficiency and results in faster optimization
→ kernel-from-Hessian enables easy computation of correlations even in high dimension

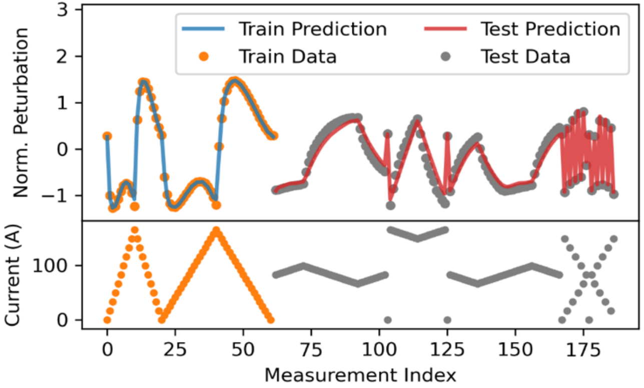
Addressing Magnetic Hysteresis with Differentiable Physics Models

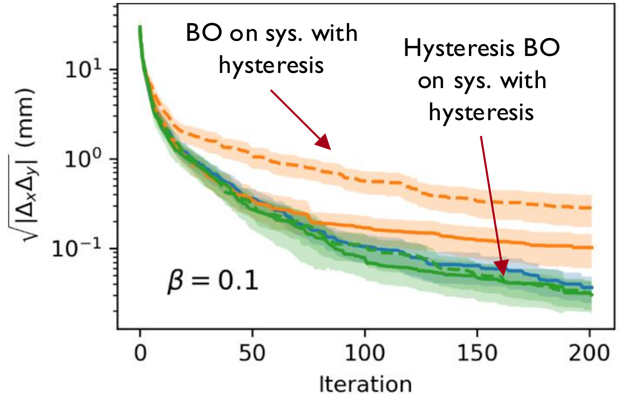
Learn both hysteresis properties and beam response simultaneously









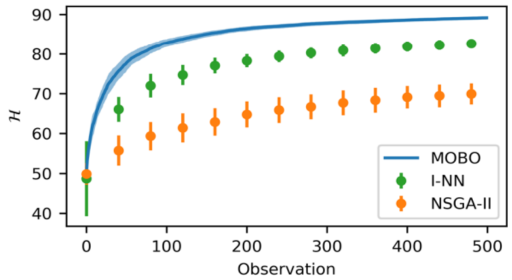


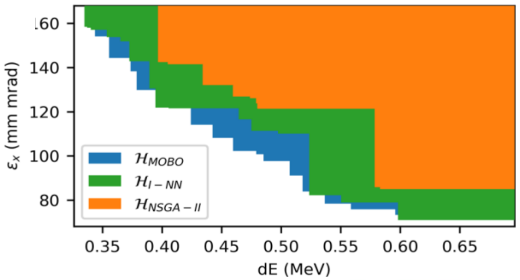
R. Roussel, et. al. Phys. Rev. Lett. ***128****, 204801*

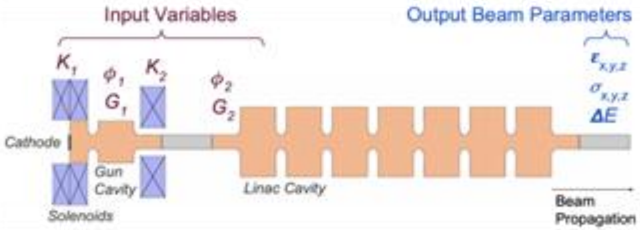
Differentiable physics model + Gaussian process enables in-situ characterization of hysteresis and faster magnet tuning

Multi-Objective Bayesian Optimization

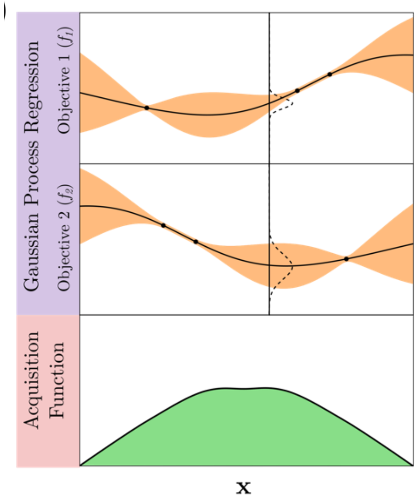
Roussel et. al. PRAB **2021**

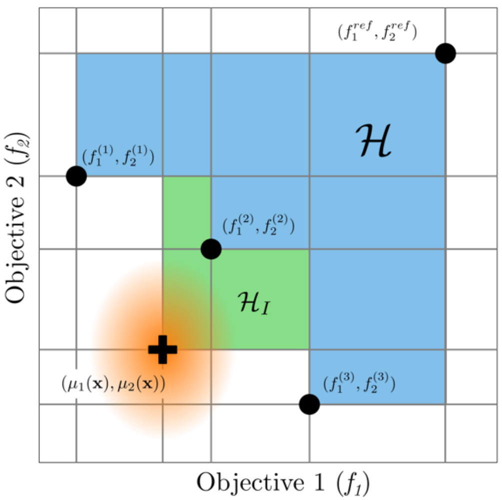


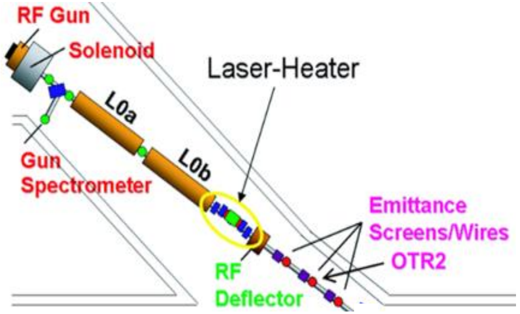


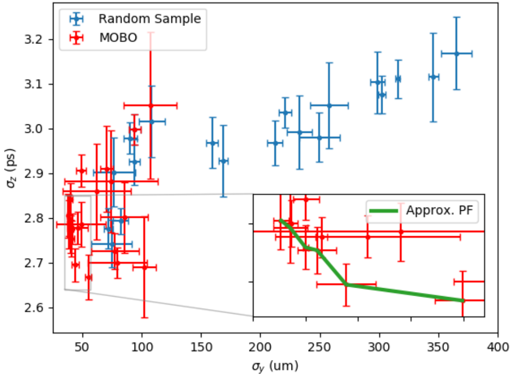


Simulation study with the AWA injector







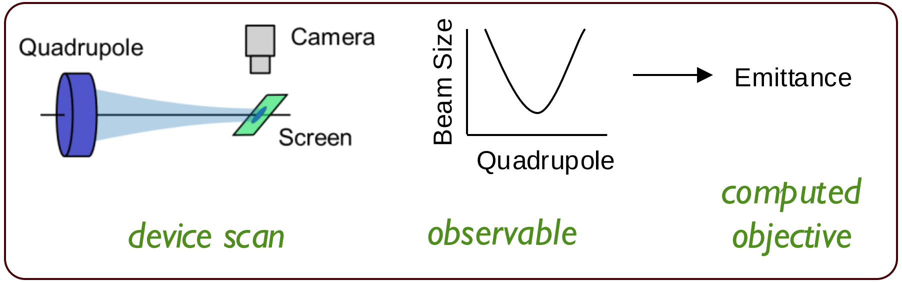


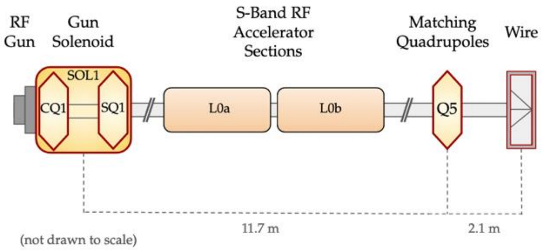
Experimental demo with the LCLS injector

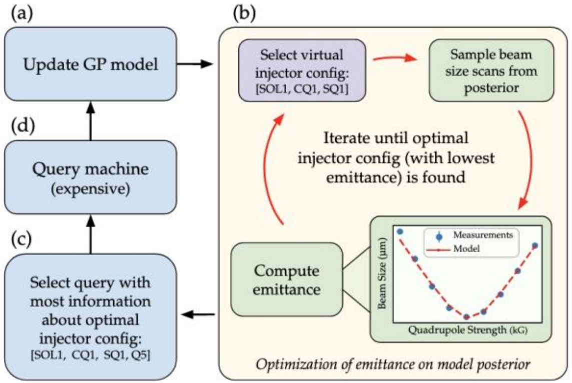
Multi-objective Bayesian optimization enables efficient, direct examination of experimental tradeoffs

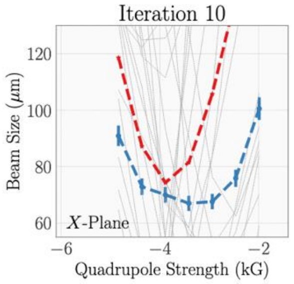
Optimization with Virtual Objectives

- Many objectives require layered scans or optimization problems
- Instead learn model from scratch online and do scan on model
- Bayesian Algorithm Execution (BAX) → 20x speedup in tuning

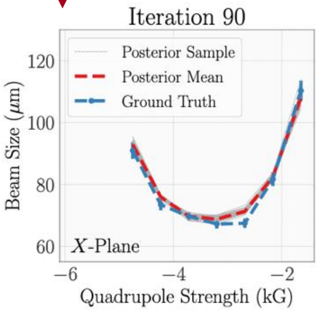


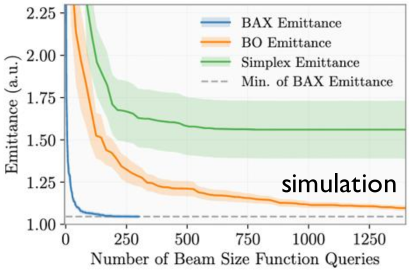


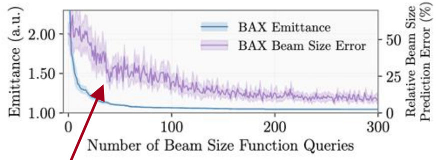




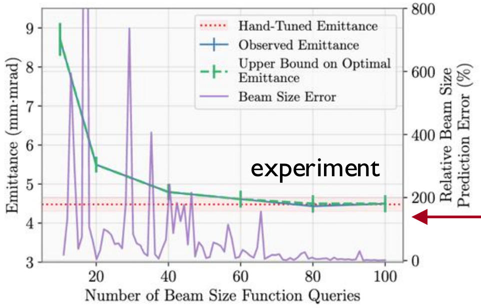
model is learned on-the-fly







Convergence of beam size prediction error gives practical indicator of convergence



20x faster tuning than standard BO, equivalent or better solution than hand-tuning

S. Miskovich, MLST, 2024

BAX enables a paradigm shift in how optimization problems with complicated scans or other indirect measurements are handled

Argonne NATIONAL LABORATORY

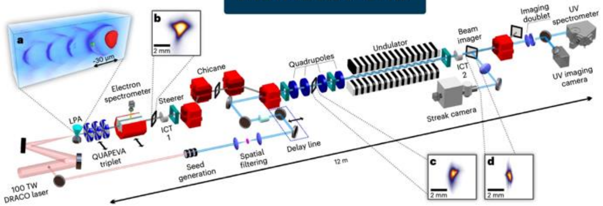
SLAC NATIONAL ACCELERATOR LABORATORY

Fermilab

Xopt

Badger GUI interface

BERKELEY LAB



Brookhaven National Laboratory

ESRF The European Synchrotron

DESY.

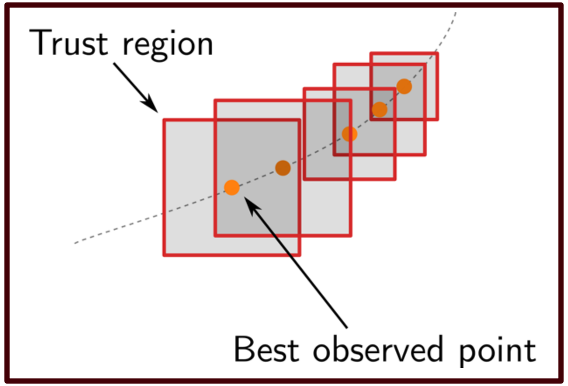
Roussel, et al. IPAC 2023 THPL164

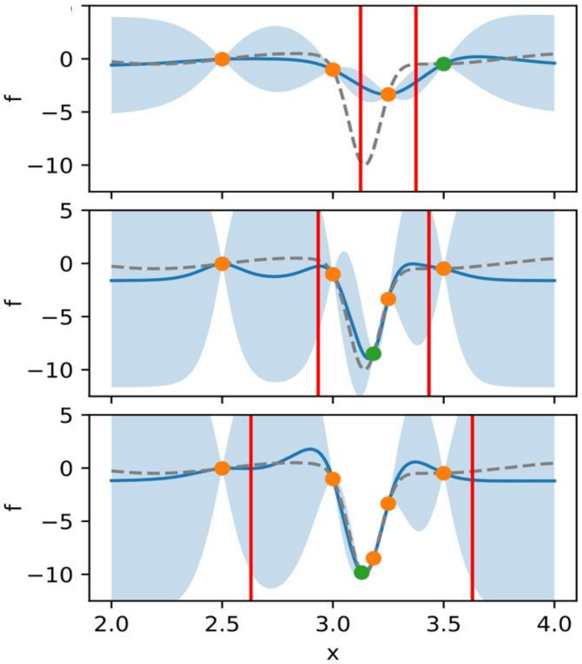
https://github.com/xopt-org/Xopt

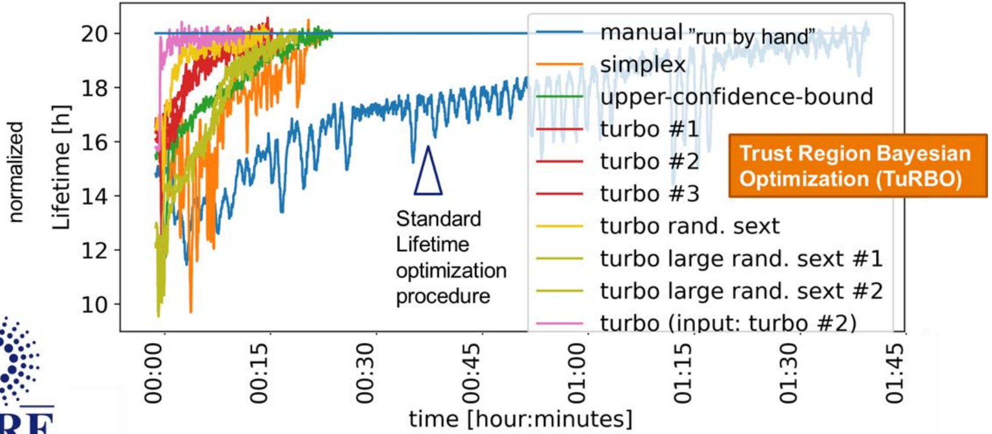
We welcome collaborators → contact us!

Common software tools (Xopt, Badger) enables rapid transfer between facilities and algorithmic progress
Also working to link accelerator and photon beamline tuning

Trust Region Bayesian Optimization







ESRF
The European Synchrotron

S. Maria Liuzzo, ICALEPCS 2023, MO3AO01

ESRF for lifetime optimization:

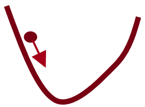
- 50x faster than human operator
- Achieved best lifetime yet observed at ESRF (41 hours)
- Now used in regular operation

Trust region BO enables efficient extension to very high dimensional problems with narrow ranges of stability

Tuning approaches leverage different amounts of data / previous knowledge → suitable under different circumstances

less ← assumed knowledge of machine → more

Model-Free Optimization

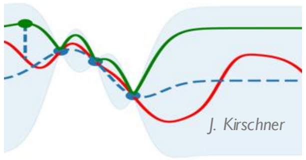


Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

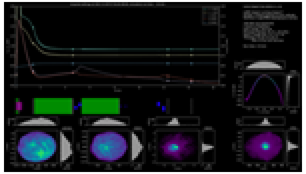


Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



→ provide initial guess (i.e. warm start)
→ provide insight to operators
→ model-based control

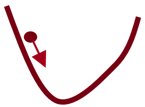
ML system models +
inverse models

General strategy: start with sample-efficient methods that do well on new systems, then build up to more data-intensive and heavily model-informed approaches.

Tuning approaches leverage different amounts of data / previous knowledge → suitable under different circumstances

less ← assumed knowledge of machine → more

Model-Free Optimization

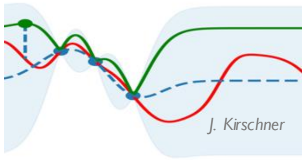


Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

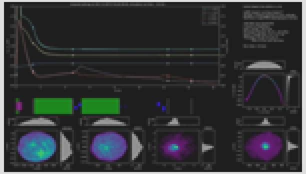


Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



→ provide initial guess (i.e. warm start)
→ provide insight to operators
→ model-based control

ML system models +
inverse models

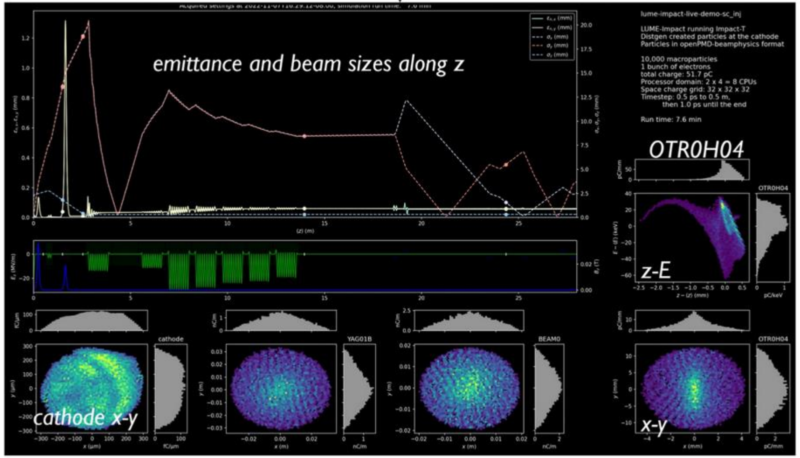
General strategy: start with sample-efficient methods that do well on new systems, then build up to more data-intensive and heavily model-informed approaches.

Combining BO with Warm Starts from Online Physics Models

Used combination of online physics simulation and Bayesian optimization algorithms to aid LCLS-II injector commissioning

Readings from machine via EPICS

injector settings, laser profile from VCC image

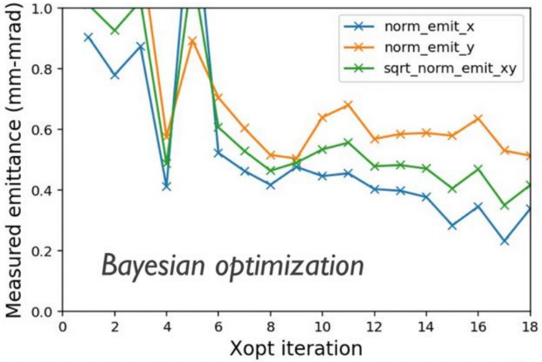


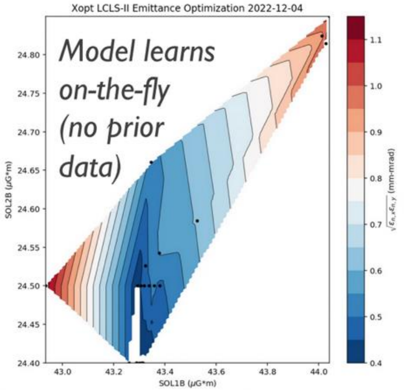
LCLS-II live sim: run on HPC and display in control room

Updates every 3-8 mins, space charge included, uses LUME-IMPACT

Adjust settings / ranges with insight from predictions

Hand over to ML-based optimization for fine tuning





Best emittance yet obtained during LCLS-II injector commissioning

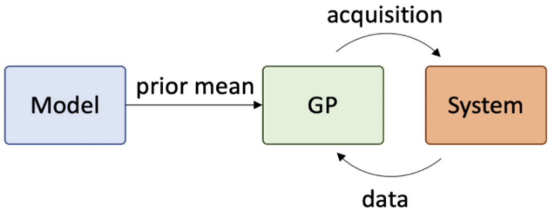
despite extensive previous hand-tuning

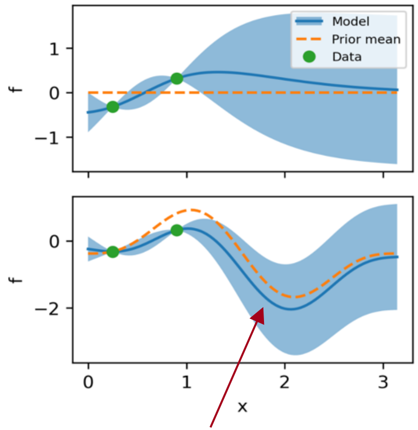
Physicists' intuition aided by detailed online physics model → simple example of how a "virtual accelerator" can aid tuning

HPC enables fundamentally new capabilities in what can be realistically simulated online

Leveraging Online Models for Faster Optimization

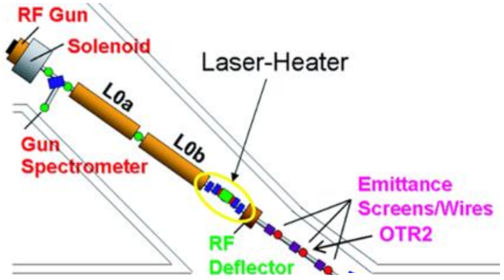
Combining existing models with BO
→ **important for scaling up to higher dimension**

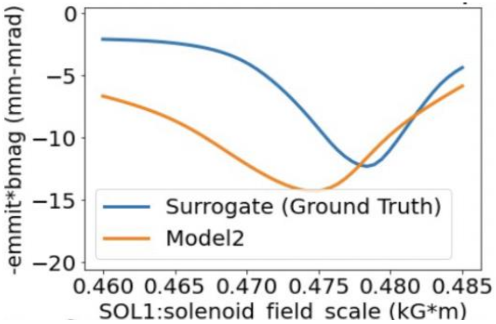




model prediction returns to prior

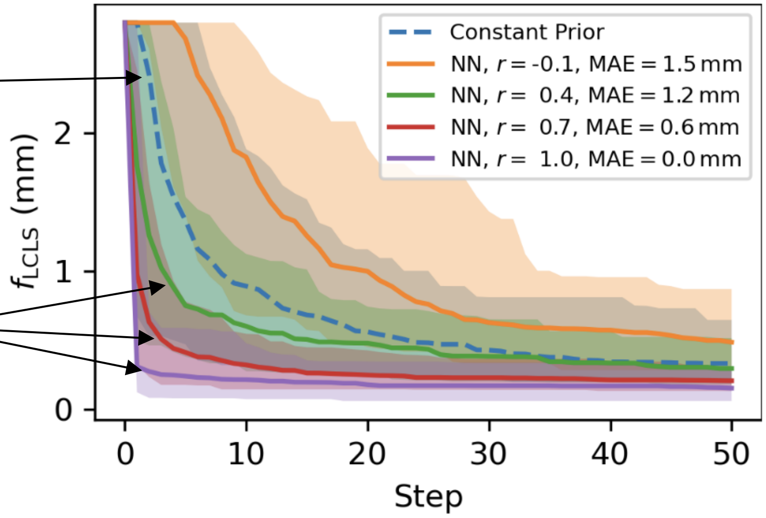
Prototyped on LCLS injector
variables: *solenoid, 2 corrector quads, 6 matching quads*
objective: *minimize emittance and matching parameter*





regular Bayesian optimization

prior mean from models with different fidelity



Even prior mean models with substantial inaccuracies provide a boost in optimization speed

https://arxiv.org/abs/2403.03225 https://arxiv.org/abs/2211.09028

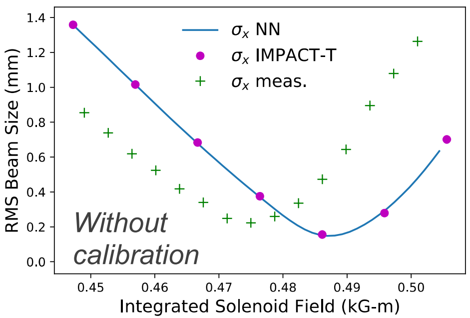
Finding Sources of Error Between Simulations and Measurements

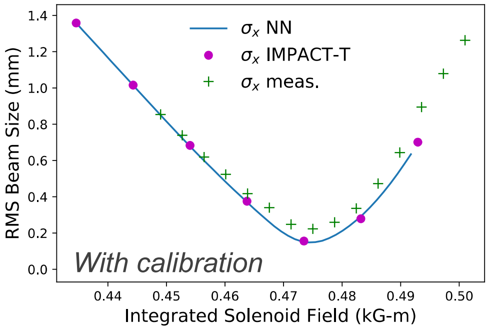
Many non-idealities not included in physics simulations:

- **static error sources** (e.g. magnetic field nonlinearities, physical offsets)
- **time-varying changes** (e.g. temperature-induced phase calibrations)

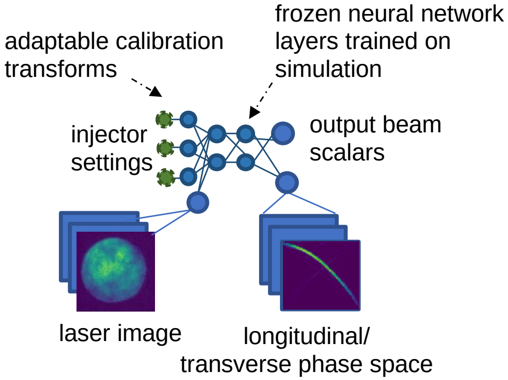
Want to identify these to get better understanding of machine performance

à ***ML model allows fast / automatic exploration of error sources in high dimension***





Example: calibration offset in injector solenoid strength found automatically with neural network model (trained first in simulation, then calibrated to machine)



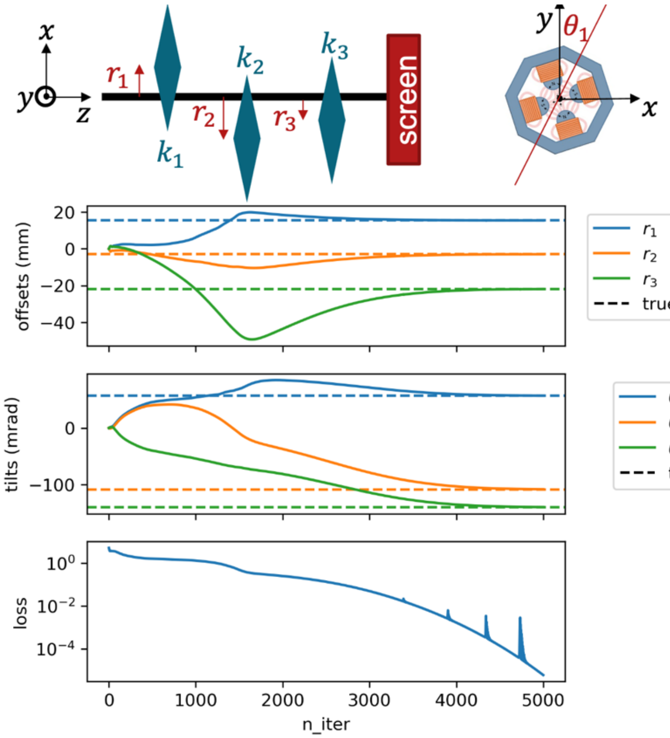
Inputs
Laser radius
Laser spot sizes
Pulse length
Charge
Solenoid
L0A phase
L0B phase
SQ quad
CQ quad
6 matching quads

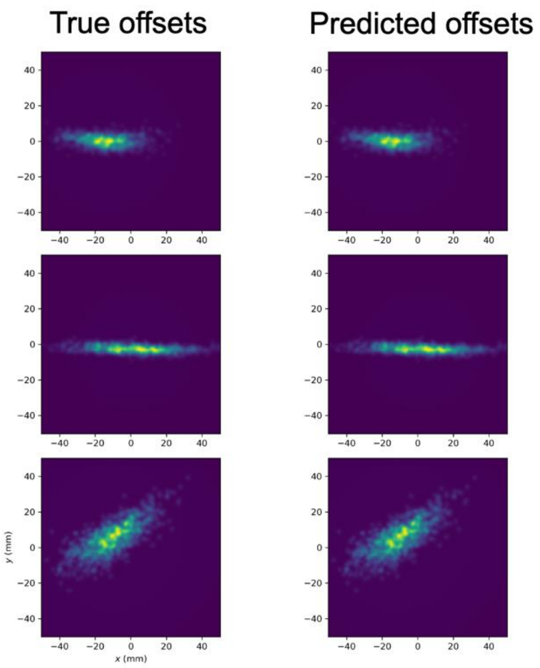
Outputs
Beam size (x,y)
Emittance (x,y)
Bunch length

Speed and differentiability of ML models enables rapid identification of error sources between idealized physics simulations and real machine

Finding Sources of Error Between Simulations and Measurements

Same approach can be used with differentiable physics simulations





Present limitations:

- Nonlinear collective effects (space charge, CSR)
- Computational scaling
- Expand to full EM cavity/magnet descriptions?

→ Looking to community to expand tools for differentiable sims!
(e.g. Cheetah, SciBmad)

J.P. Gonzalez-Aguilera
https://accelconf.web.cern.ch/ipac2023/pdf/WEPA065.pdf

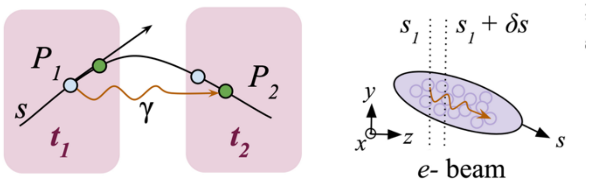
Differentiable simulations allow direct learning of calibrations while being constrained by the expected physics

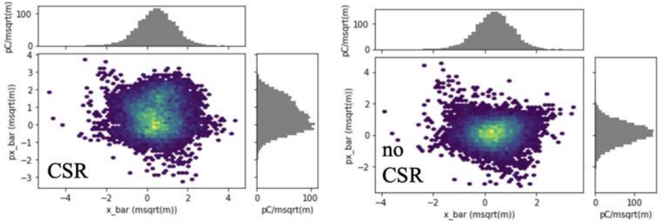
Embedding surrogates in tracking calculations

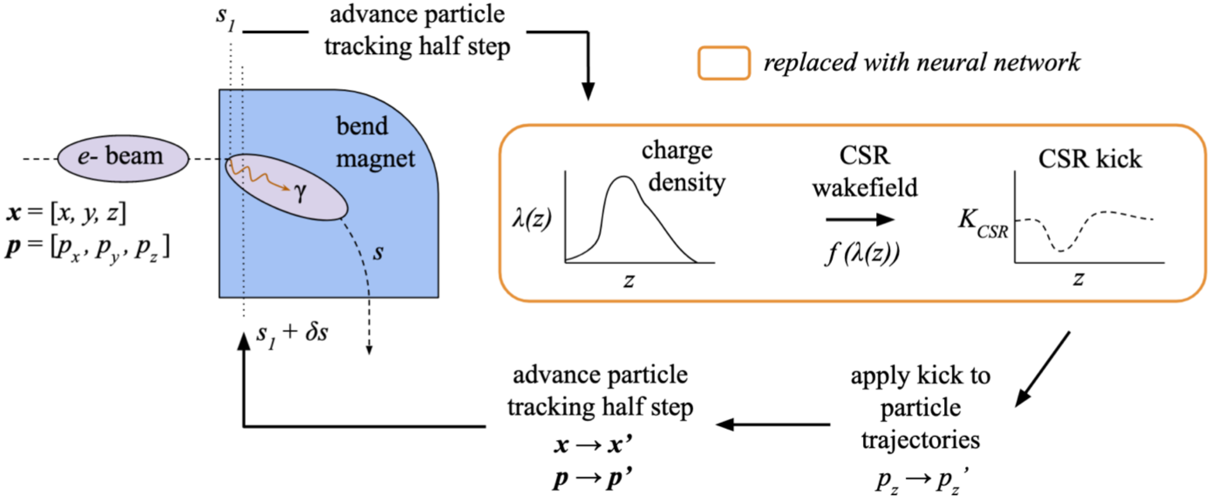
Coherent Synchrotron Radiation (CSR) impacts beam quality (critical for Free Electron Laser performance)

CSR computationally intensive to simulate, even for 1D effect

Solution: replace wakefield calculation in tracking step with a neural network to gain both speed and differentiability







Trained fully-connected, feed-forward network

Trained on >1M samples from 10k different initial beam distributions (generated from start-to-end LCLS sims with random linac settings)

Edelen, et al., IPAC'22 https://accelconf.web.cern.ch/ipac2022/papers/wepoms013.pdf

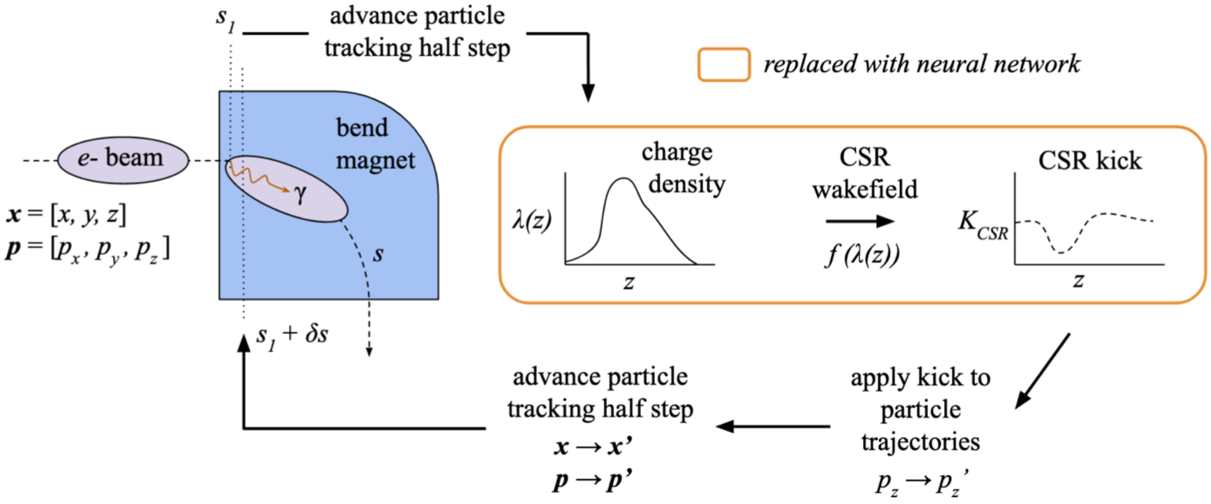
Embedding surrogates in tracking calculations

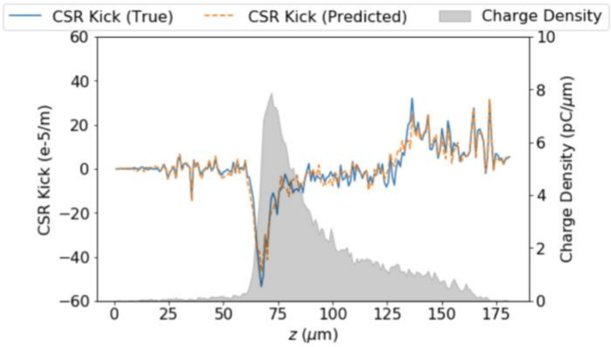
Coherent Synchrotron Radiation (CSR) impacts beam quality (critical for Free Electron Laser performance)

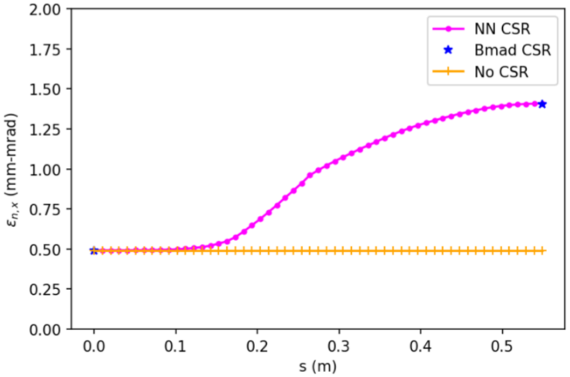
CSR computationally intensive to simulate, even for 1D effect

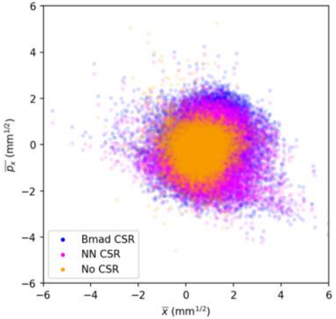
Solution: replace wakefield calculation in tracking step with a neural network to gain both speed and differentiability

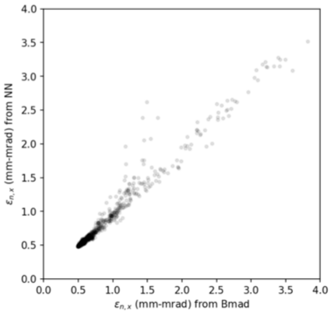
→ Accurately replicates main effect (better than excluding CSR)
→ 10X faster than running with 1D CSR routine











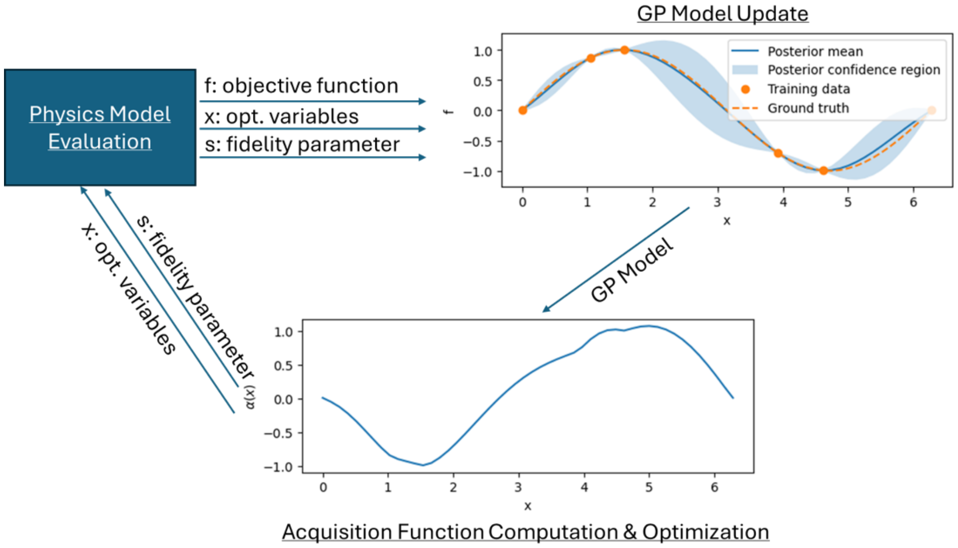
Edelen, et al., IPAC'22 https://accelconf.web.cern.ch/ipac2022/papers/wepoms013.pdf

Multi-fidelity Model Calibration

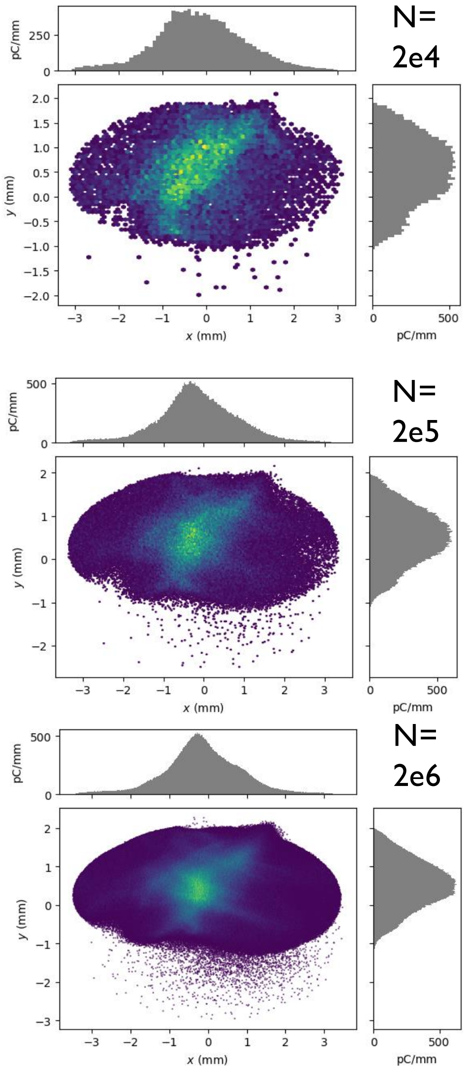
Want to efficiently probe possible model errors and obtain uncertainty estimates.

Multi-fidelity Bayesian optimization:

- Learn correlations between different model fidelities
- Use BO to select model fidelity and next optimization variables



Number of Particles (N)	2e4	2e5	2e6
Space Charge Grid Size	16	32	64
Execution time	~1 min	~2.5 min	~25 min
σ_x (um)	1026	1018	1017
σ_y (um)	654	623	614
Norm x emit (um)	9.26	8.87	8.77

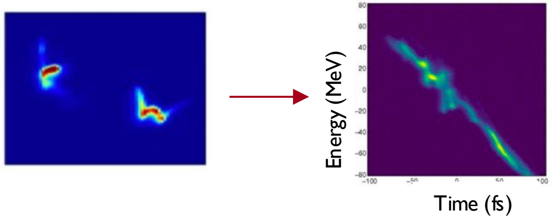


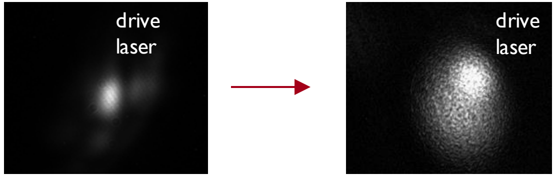
E. Cropp

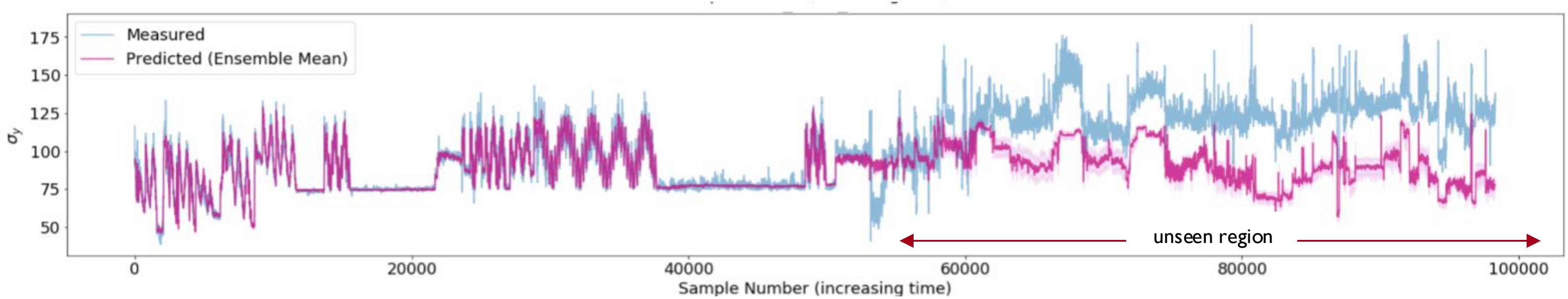
Distribution Shift is a Major Challenge in Particle Accelerators

Many sources of change over time:

- Deliberate changes in beam configuration (e.g. beam charge)
- Unintended drift in initial conditions (including in unobservable variables), diurnal temperature/humidity changes, etc
- Time-dependent action of feedback systems







Example: beam size prediction and uncertainty estimates under drift from a neural network
Uncertainty estimate from neural network ensemble does not cover prediction error, but does give a qualitative metric for uncertainty

Reliable uncertainty estimates and model adaptation methods are key for putting online models to use operationally

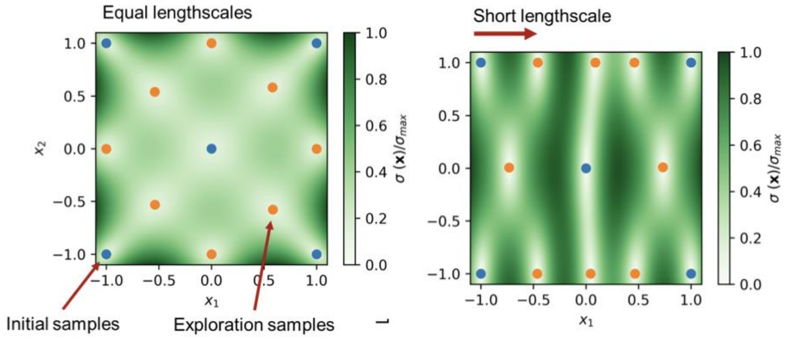
Efficient Characterization with Bayesian Exploration

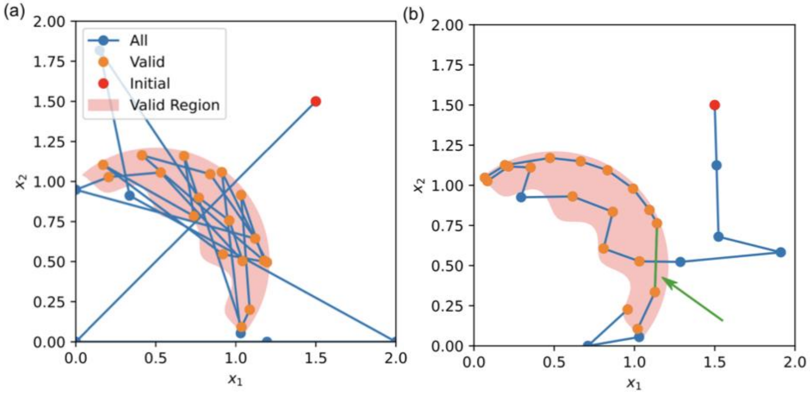
$$\alpha(\boldsymbol{x}) = \sigma(\boldsymbol{x}) \prod_{i=1}^{N} p_i(g_i(\boldsymbol{x}) \geq h_i)\, \Psi(\boldsymbol{x}, \boldsymbol{x_0})$$

R. Roussel et. al.
Nat. Comm. **2021**

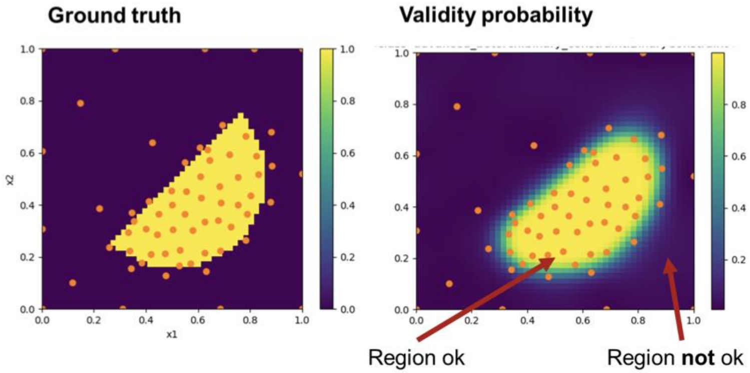
proximal biasing

adaptive sampling





learning constraints



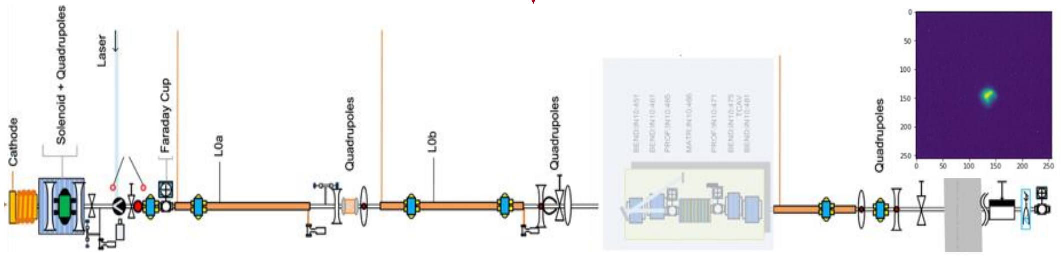
Enables sample-efficient characterization of high-dimensional spaces, while respecting both input and output constraints

Bayesian Exploration for Efficient Characterization

Automatic Exploration
(constrained to useful values of emittance and match)

Comprehensive ML Models of Injector

Setting changes on 10 variables (solenoid, bucking coil, corrector quads and matching quads)



FACET-II Injector

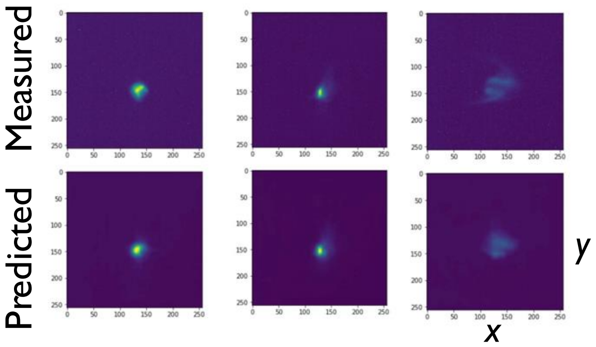
x-y emit, match, and beam images

- Used Bayesian Exploration for efficient high-dimensional characterization (10 variables) of emittance and match at 700pC: **2 hrs for 10 variables compared to 5 hrs for 4 variables with N-D parameter scan (~8x faster)**

- Data was used to train neural network model of injector response predicting x-y beam images. GP ML model from exploration predicts emittance and match.

- **Example of integrated cycle between characterization, modeling, and optimization → now want to extend to larger system sections and new setups**

transverse phase space



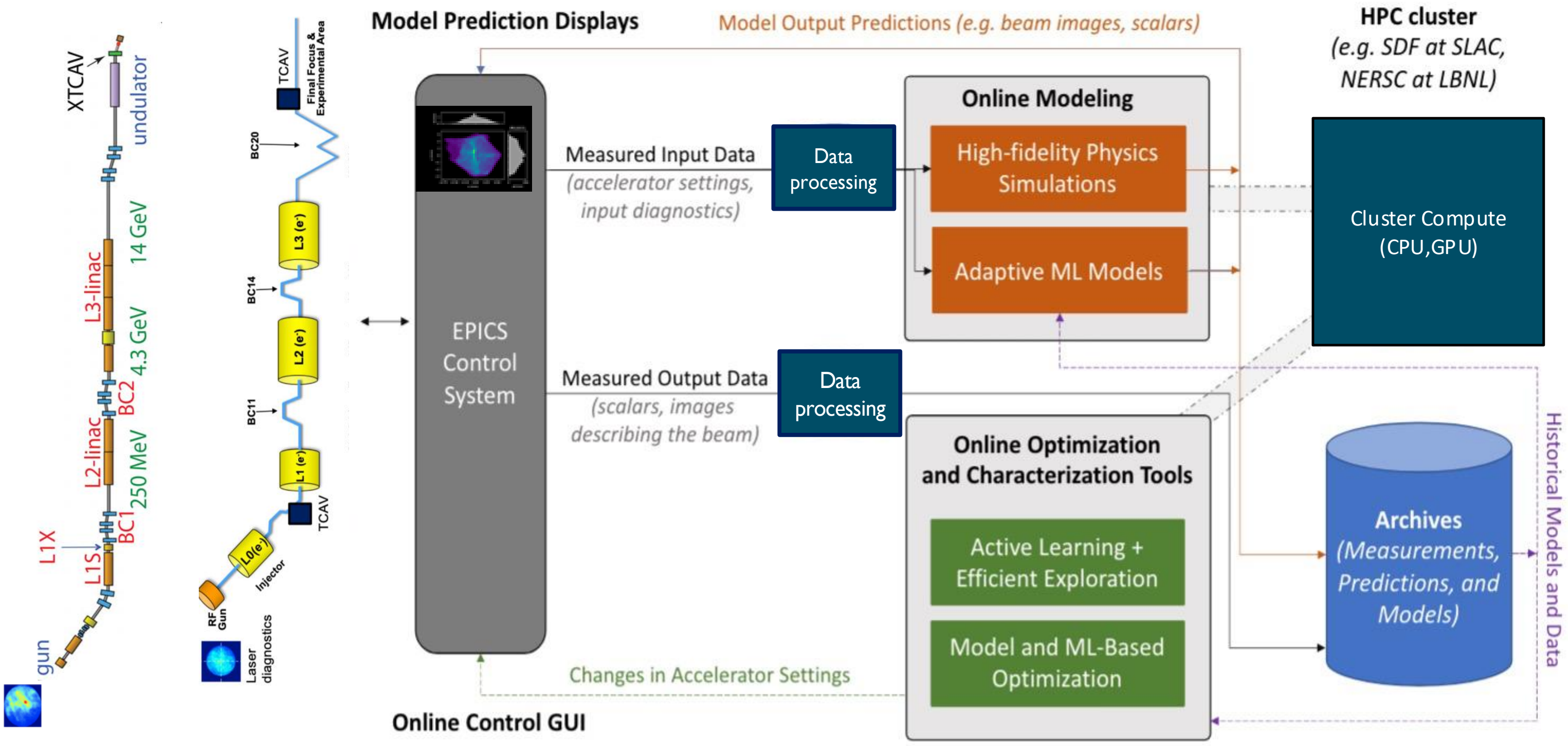
https://www.nature.com/articles/s41467-021-25757-3

Use of Bayesian exploration to generate training data was sample-efficient, reduced burden of data cleaning, and resulted in a well-balanced distribution for the training data set over the input space. ML models were immediately useful for optimization.

Goal: Full Integration of AI/ML Optimization, Data-Driven Modeling, and Physics Simulations

Working on a ***facility-agnostic*** ecosystem for online simulation, ML modeling, and AI/ML driven characterization/optimization

Will enable system-wide application to aid operations, and help drive AI/ML development *(e.g. higher dimensionality, robustness, combining algorithms efficiently)*



Making good progress toward this vision with open-source, modular software tools

Digital Twin Infrastructure

Ecosystem of modular tools (can use independently)

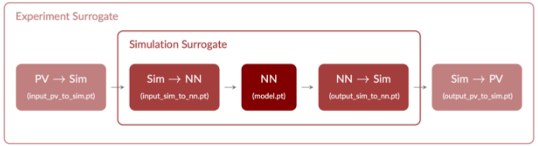
LUME – simulation interfaces/wrappers in Python

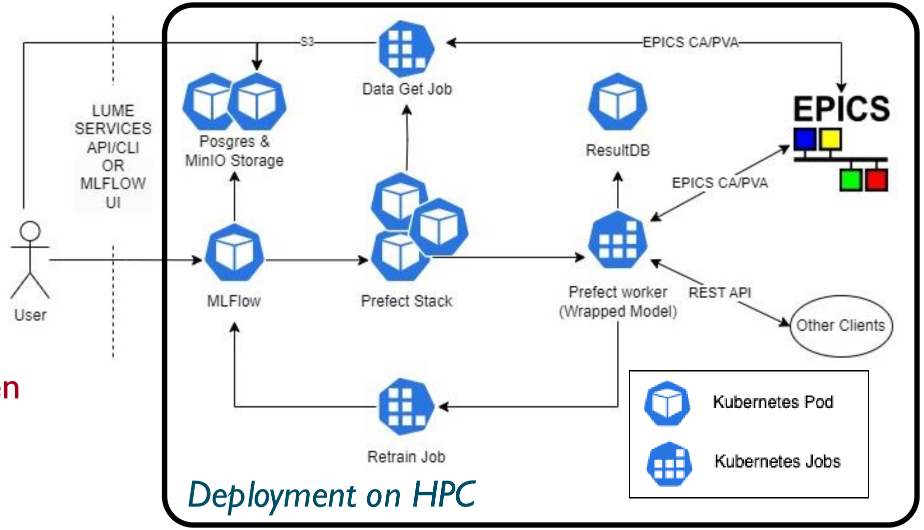
lume-model – wraps ML models, facilitates calibration

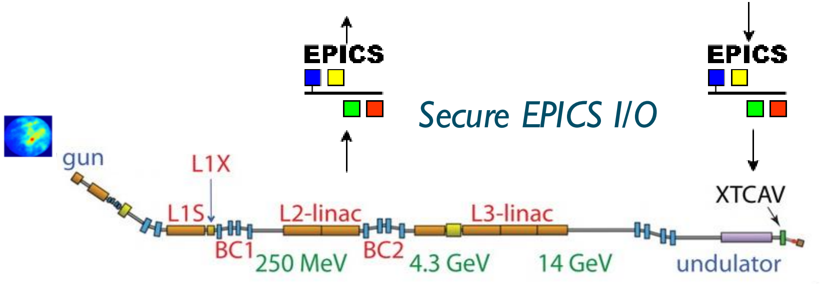
distgen – flexible creation of beam distributions

Integration with MLFlow for MLOps

https://www.lume.science/





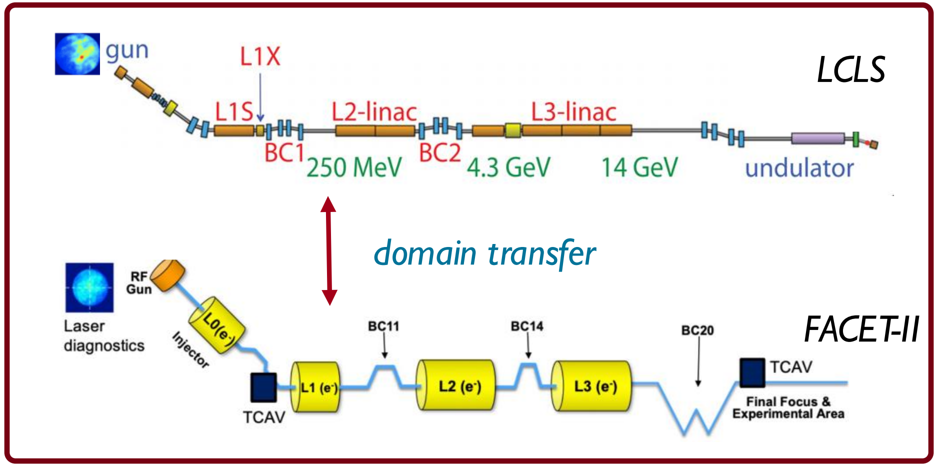


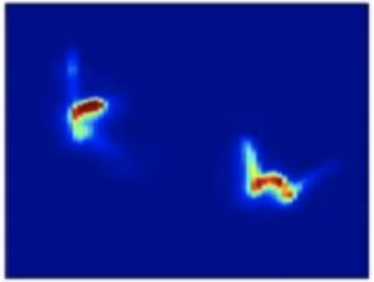
- Live physics simulations and ML models now linked between SLAC's HPC system (S3DF) and control system
  → *run with Kubernetes and Prefect*
- Working with NERSC to swap between S3DF/NERSC resources
- Beginning work on MLOps aspects that will be used in continual learning research
- SLAC is part of CAMPA project for end-to-end virtual accelerators → working on shared set of tools/standards/interfaces

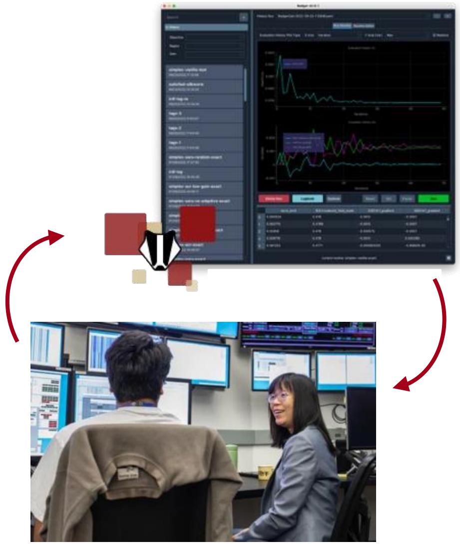
Substantial progress on deploying ML and Physics-based models and integrating with HPC in a portable way

Summary/Conclusions

- Many successes with Bayesian optimization and variants → *many ways of incorporating system models for improved performance in BO*
- Have deliberately targeted ML-based approaches that don't require large amounts of data and are readily transferrable between systems
- Online system models that combine physics simulations and ML being deployed → *scaling up toward comprehensive digital twins*
- Differentiable physics simulations including nonlinear collective effects and hybrid ML models are a major area of need *(e.g. Cheetah, SciBmad)*
- Deployment infrastructure and shared community software tools are essential
- Increasingly working to combine system models and online optimization to enable more detailed control



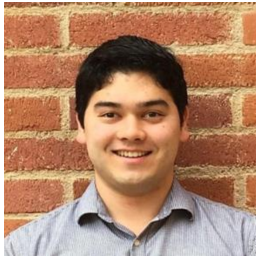
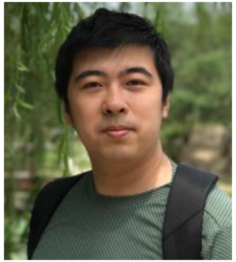




Thanks for your attention!
Any questions?











Thanks to the core team at SLAC working on various AIML technologies and infrastructure!

Thanks to many other collaborators not shown!

Backups

Reinforcement Learning

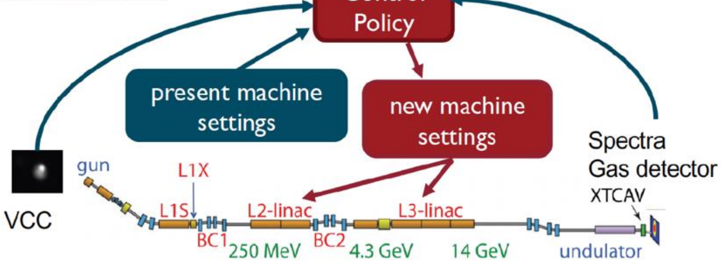
RL can help address a different set of needs than BO:

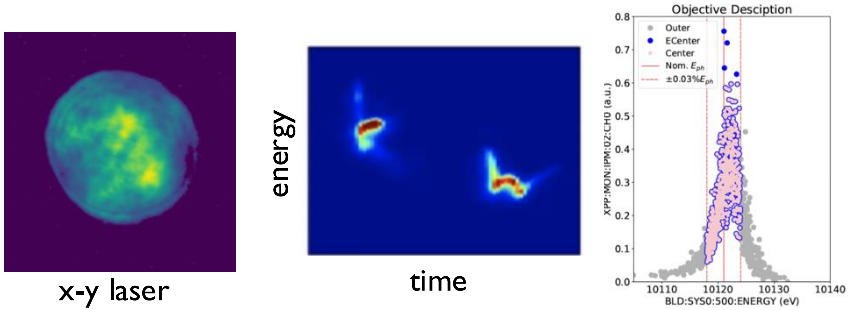
- Use global machine information, more historical data
- Treat as a dynamical system *(many time-dependent processes/feedbacks + drift)*
- Address demands for fast dynamic control from users

Suitability of accelerator tuning problems for RL:

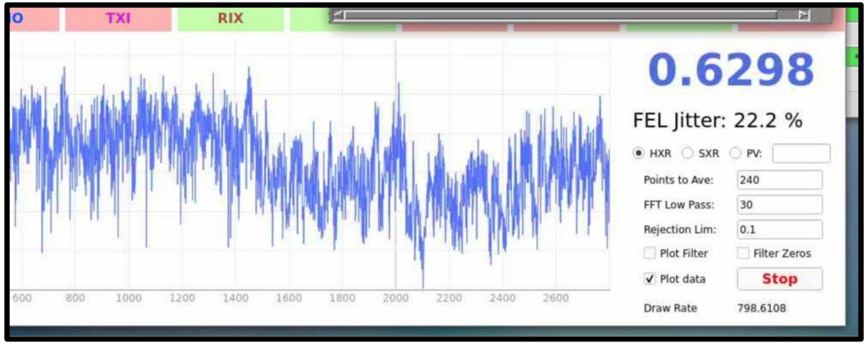
- Many variables, multi-modal signals (images, scalars, time series)
- Continuous state/action spaces (similar to robotics)
- Have physics models/simulators for many problems

Nagabandi, et al., 2019





Variety of high dimensional signals for states, objectives



120 Hz FEL pulse intensity

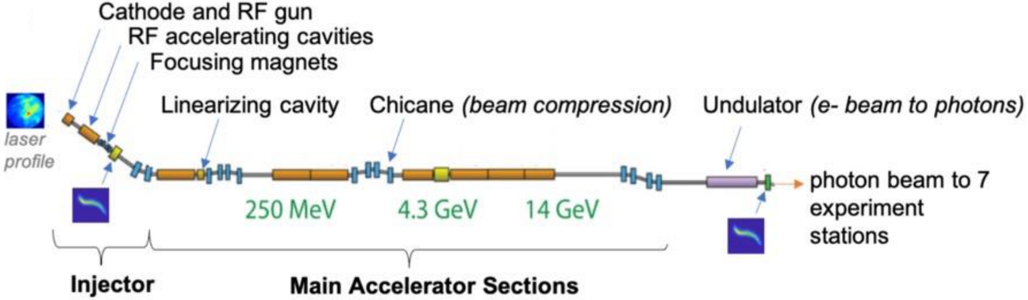
Nonlinear instability → sensitive to dynamic processes

(e.g. trajectory feedback, cooling, LLRF control)

Reinforcement Learning

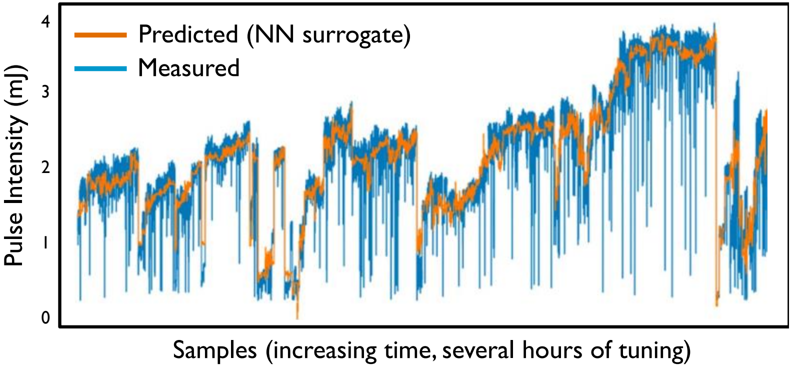
- FEL is sensitive to focusing, trajectory; perturbing beam/feedbacks too much results in beam losses
- Using data-driven surrogates and differentiable sims to train agents
- Iteratively add more data, targets and variables:
  - Photon pulse intensity
  - Beam phase space images, spectra
  - Focusing magnets, RF cavities, undulator
- Similar accelerator designs may enable facility-agnostic agents?

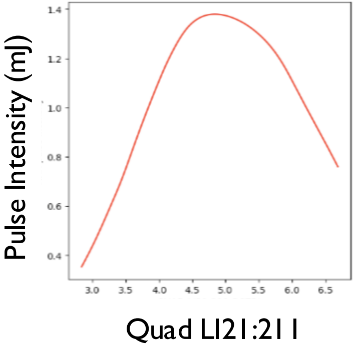
  Starting to explore with EuXFEL



~28 focusing magnets for FEL pulse intensity

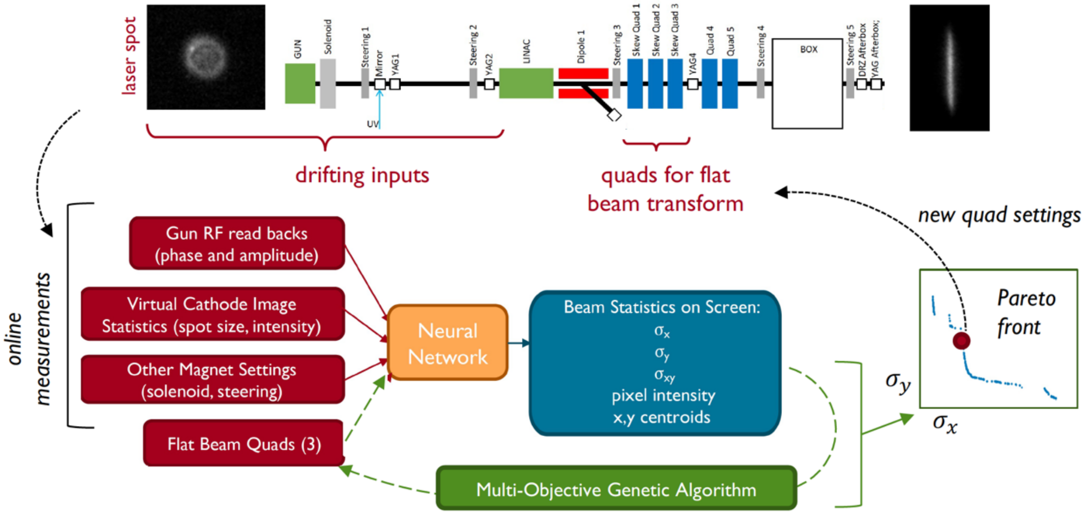
(many more variables to include: steering, rf cavities, undulator, drive laser)



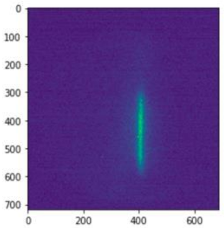


Example: Warm Starts from Online Models

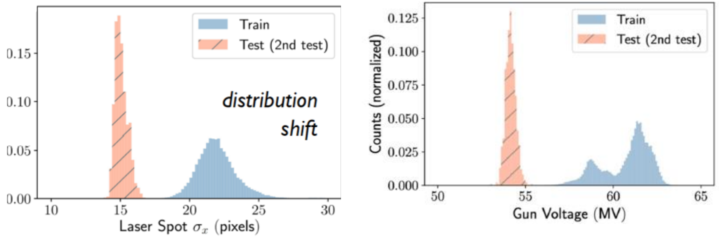
E. Cropp et al., in preparation

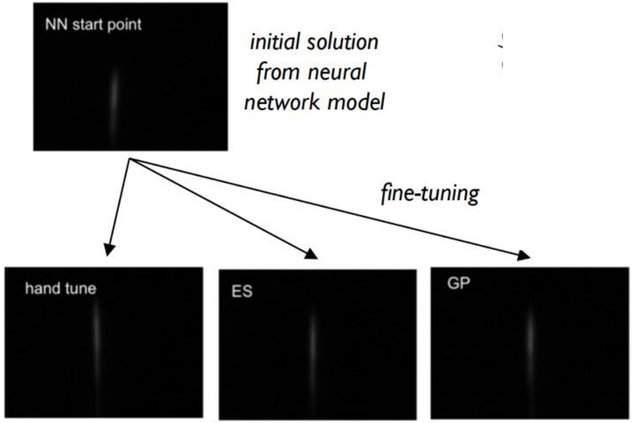


- Round-to-flat beam transforms are challenging to optimize → 2019 study explored ability of a learned model to help
- Trained neural network model to predict fits to beam image, based on archived data
- Tested online multi-objective optimization over model (3 quad settings) given present readings of other inputs
- Used as warm start for other optimizers
- Trained DDPG Reinforcement Learning agent and tested on machine under different conditions than training



Can work even under distribution shift





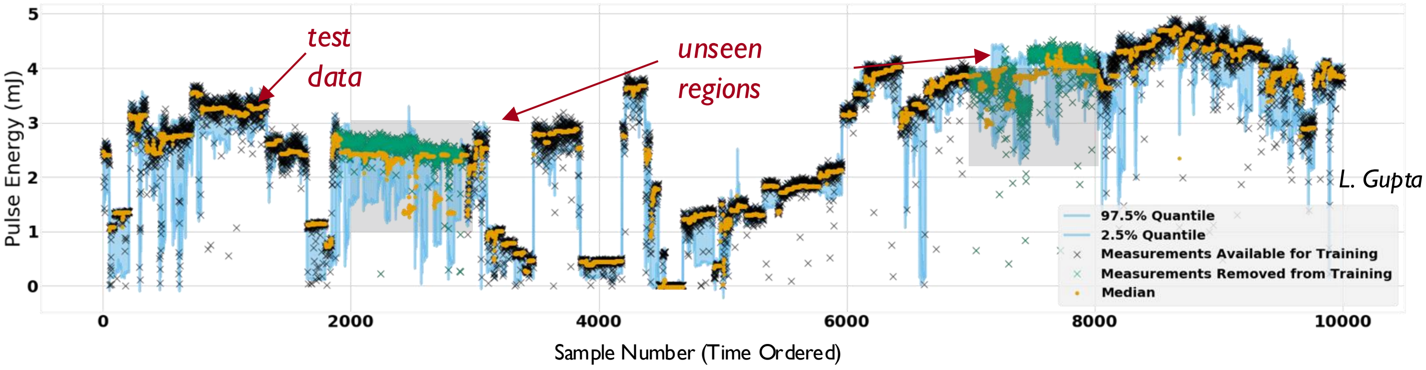
Hand-tuning in seconds vs. tens of minutes

Boost in convergence speed for other algorithms

Uncertainty Quantification / Robust Modeling

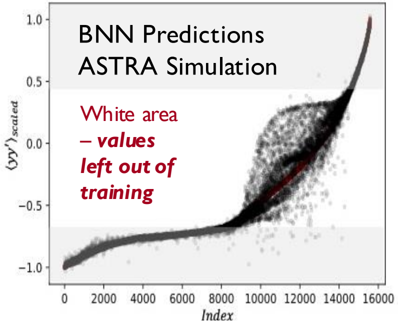
Essential for decision making under uncertainty (e.g. safe opt., intelligent sampling, virtual diagnostics)

Current approaches
- Ensembles
- Gaussian Processes
- Bayesian NNs
- Quantile Regression



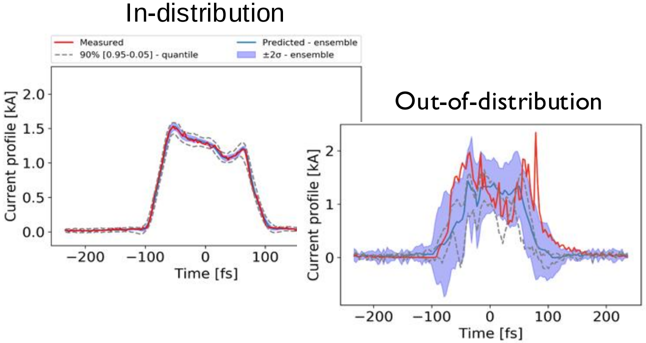
L. Gupta

Neural network with quantile regression predicting FEL pulse energy at LCLS



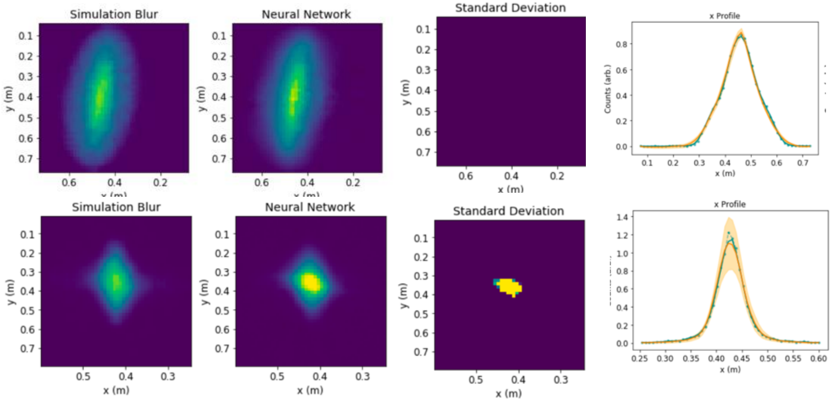
Scalar parameters for the LCLS-II injector (Bayesian neural network)

A. Mishra et. al., PRAB, 2021

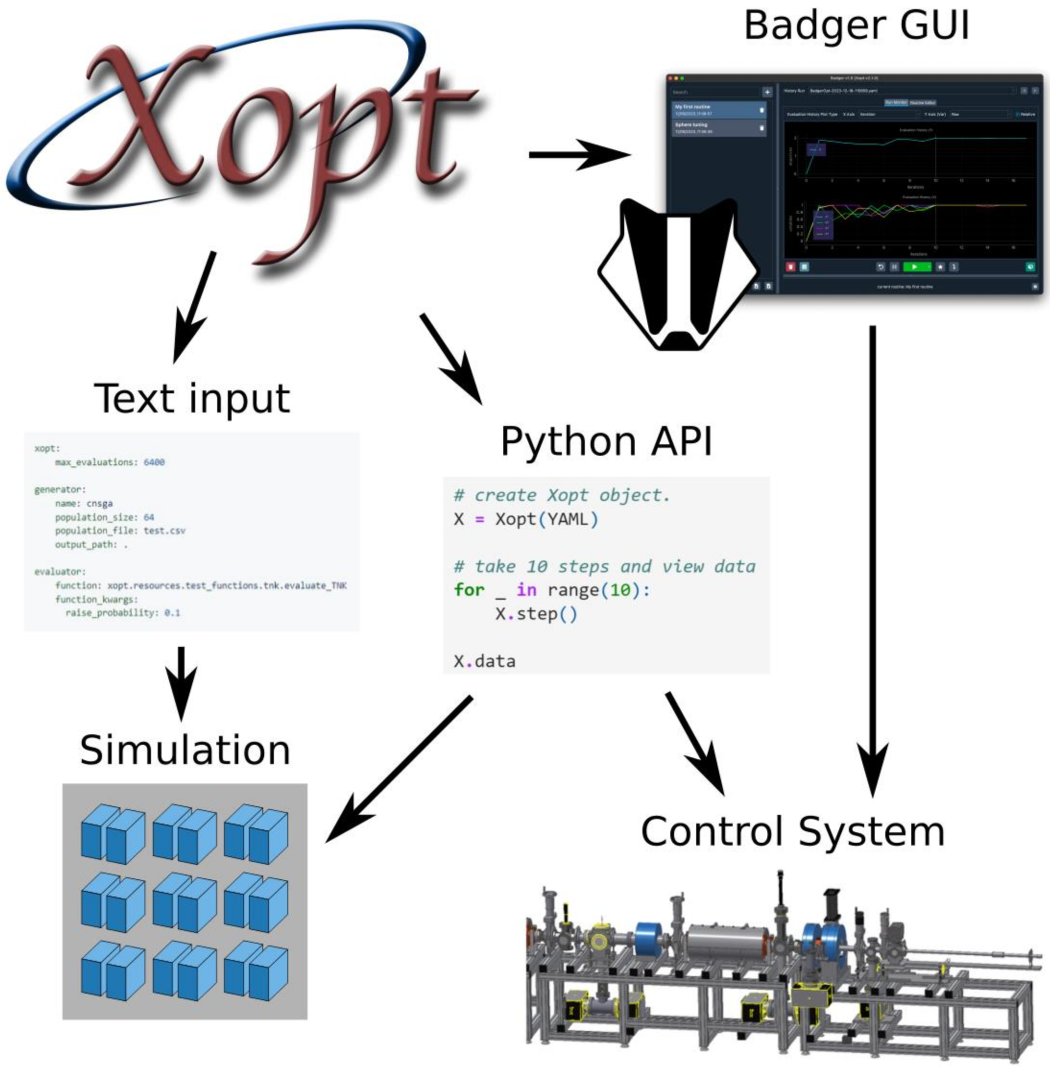


longitudinal phase space (quantile regression + ensemble)

O. Convery, et al., PRAB, 2021



LCLS injector transverse phase space (ensemble)


Badger GUI
Text input
xopt:
max_evaluations: 6400
generator:
name: cnsga
population_size: 64
population_file: test.csv
output_path: .
evaluator:
function: xopt.resources.test_functions.tnk.evaluate_TNK
function_kwargs:
raise_probability: 0.1
Python API
# create Xopt object.
X = Xopt(YAML)
# take 10 steps and view data
for _ in range(10):
X.step()
X.data
Simulation
Control System

Deployment: Xopt and Badger

Xopt: houses optimization algorithms

```yaml
xopt:
    max_evaluations: 6400

generator:
    name: cnsga
    population_size: 64
    population_file: test.csv
    output_path: .

evaluator:
    function: xopt.resources.test_functions.tnk.evaluate_TNK
    function_kwargs:
      raise_probability: 0.1

vocs:
    variables:
        x1: [0, 3.14159]
        x2: [0, 3.14159]
    objectives: {y1: MINIMIZE, y2: MINIMIZE}
    constraints:
        c1: [GREATER_THAN, 0]
        c2: [LESS_THAN, 0.5]
    linked_variables: {x9: x1}
    constants: {a: dummy_constant}
```

```python
# create Xopt object.
X = Xopt(YAML)

# take 10 steps and view data
for _ in range(10):
    X.step()

X.data
```

Python interface

Many optimization algorithms
- *Genetic algorithms (NSGA-II, etc.)*
- *Nelder-Mead Simplex*
- *Bayesian Optimization*
- *Bayesian Exploration*
- *Trust-region BO*
- *Learned output constrained BO*
- *Interpolating BO*

Badger GUI interface

User interface, I/O with machine

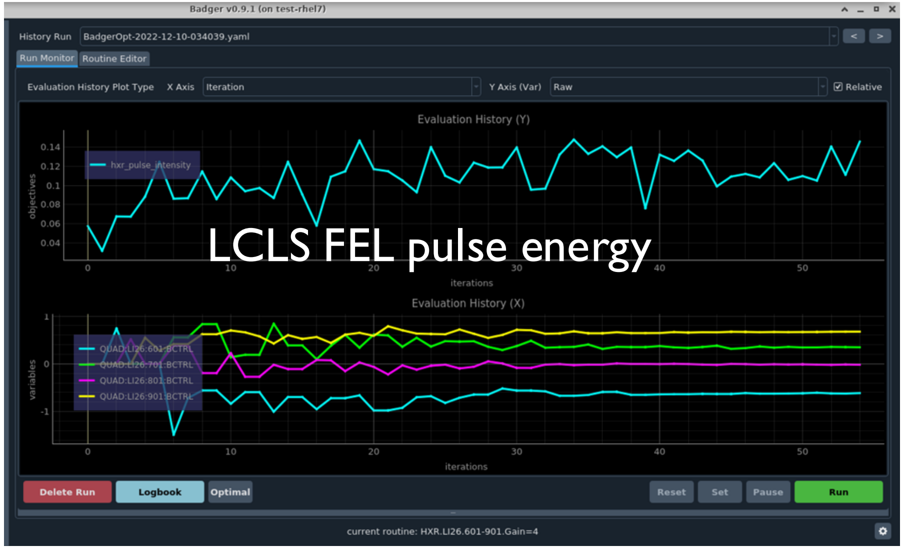
https://christophermayes.github.io/Xopt/

https://christophermayes.github.io/Xopt/algorithms/

https://github.com/slaclab/Badger

→ Has been used for online optimization at numerous facilities (LCLS/LCLS2, FACET-II, ESRF, AWA, NSLS-II, FLASHForward)

→ Working to make interoperable with other software (e.g. Gymnasium)



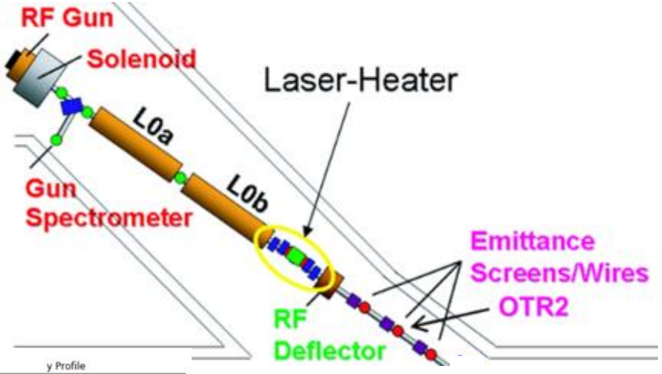
0.04 to 0.14 mJ in SXR → 15% better than hand-tuning

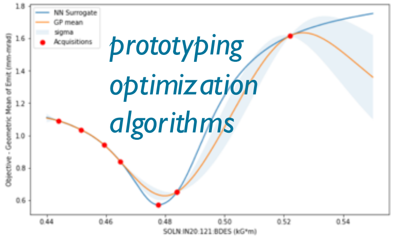
41hr → best lifetime observed ever (in record speed of 15 minutes)
injection efficiency improved by 5%

- Can specify constraints on settings and outputs (e.g. avoid dark current, beam losses, etc)
- Trust-region method allows conservative high-dimensional tuning (e.g. used >100 sextupoles at ESRF)
- Working on integrating global model priors → not learning from scratch each time
- Working to make compatible with RL problems + gymnasium

In Regular Use: Injector Surrogate Model at LCLS

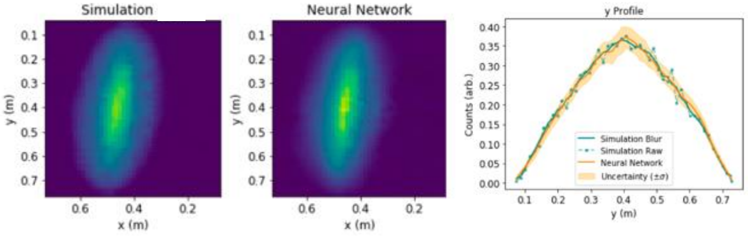
- ML models trained on detailed physics simulations with nonlinear collective effects
- Accurate over a wide range of settings → calibrate to match machine measurements
- Provide initial parameters for downstream model



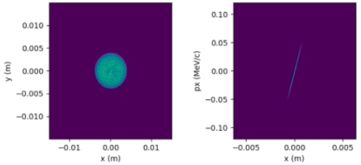


prototyping optimization algorithms

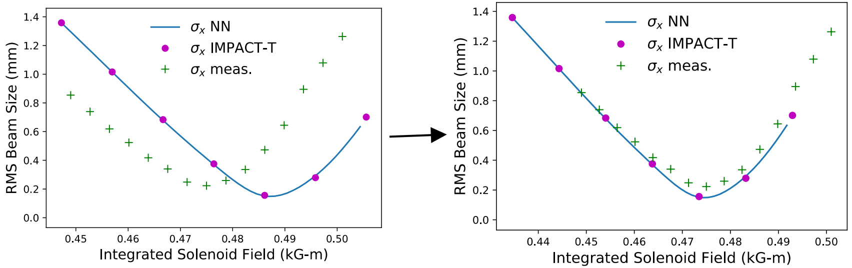
ML model matches simulation under interpolation



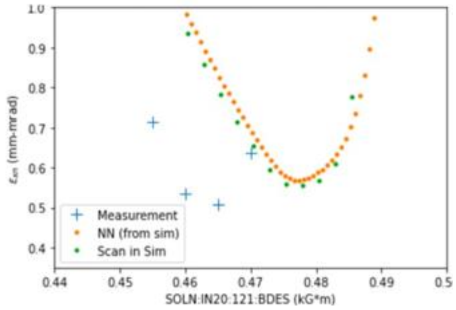
Simulation and ML model trained on it are qualitatively similar to measurements under interpolation (setting combinations reasonable distance from training set)



interactive model widget and visualization tools

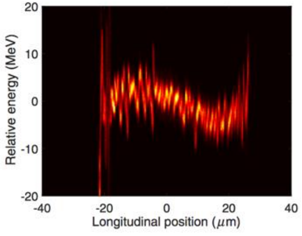


Automatic adaptation of models and identification of sources of deviation between simulations and as-built machine

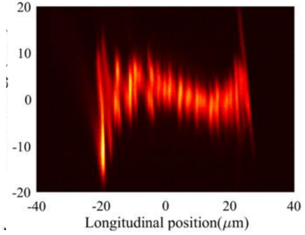


ML models trained on simulations and measurements have enabled fast prototyping of new optimization algorithms, facilitated rapid model adaptation under new conditions, and can directly aid online tuning and operator decision making *computationally expensive simulations*

Simulation



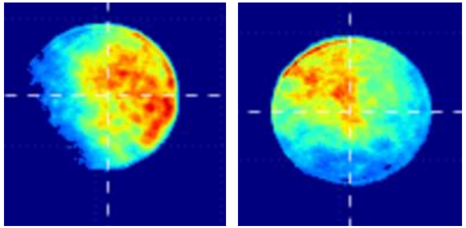
Measurement



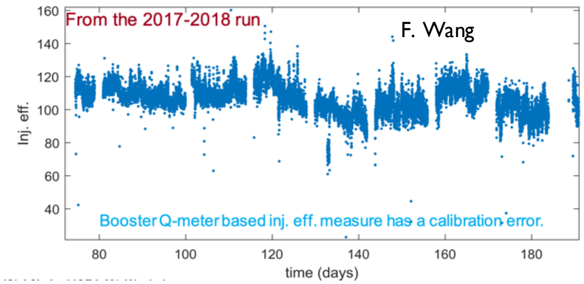
"10 hours on thousands of cores at the NERSC"

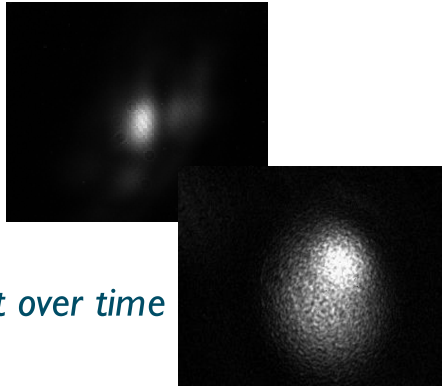
J. Qiang, et al., PRSTAB30, 054402, 2017

fluctuations/noise (e.g. initial beam conditions)

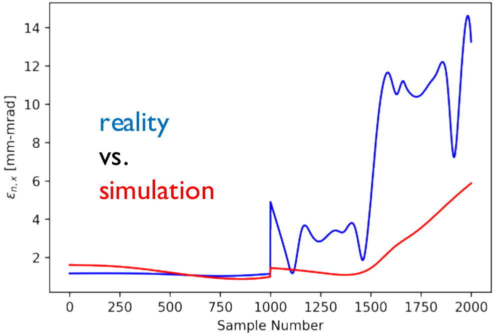


hidden variables / sensitivities

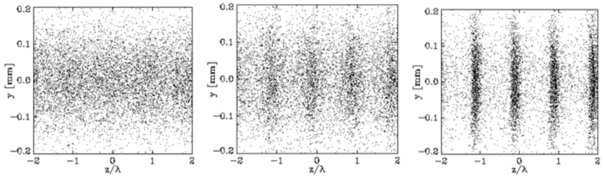




drift over time



many small, compounding sources of uncertainty

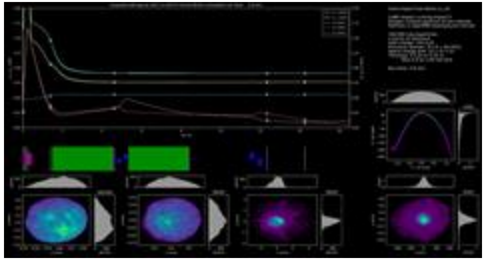


nonlinear effects / instabilities

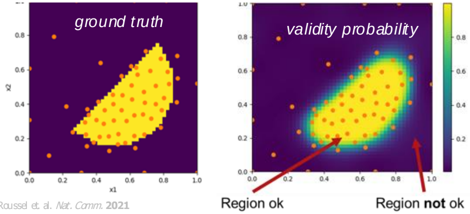
Broad Research Program at SLAC in AI/ML for Accelerators

(1) Developing new approaches for accelerator optimization/characterization and faster higher-fidelity system modeling, (2) developing portable software tools to support end-to-end AI/ML workflows, (3) helping integrating these into regular use

Online prediction with physics sims and **fast/accurate ML system models**



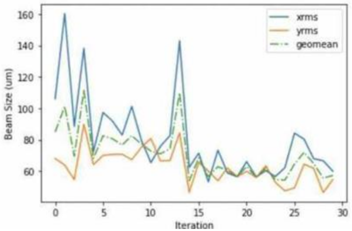
Efficient, safe optimization algorithms



Roussel et. al. *Nat. Comm.* 2021

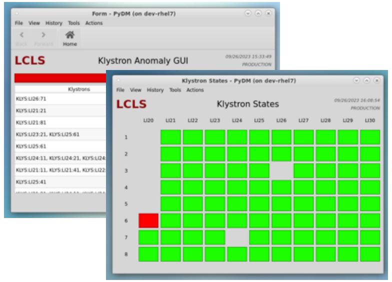
Output constraints learned on-the-fly

Adhere to constraints and balance multiple targets

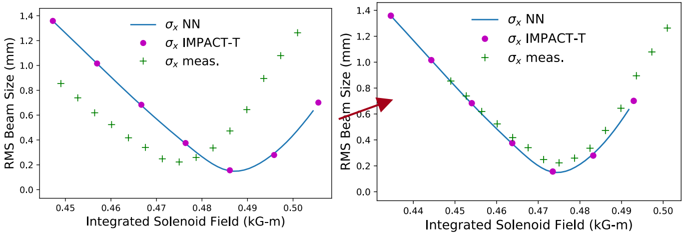


Challenging problems: e.g. sextupole tuning

Anomaly detection

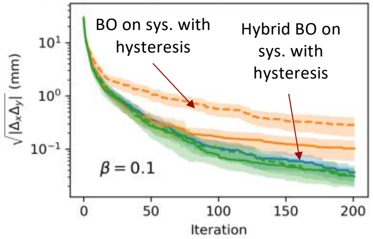


Adaptation of models and **identification of sources of deviation** between simulations and as-built machine



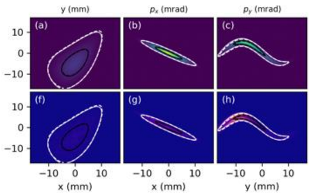
Combining physics and ML for better performance

Hysteresis-aware optimization



Roussel et. al. *PRL.* 2022

Differentiable simulations + ML for 6D phase space reconstruction

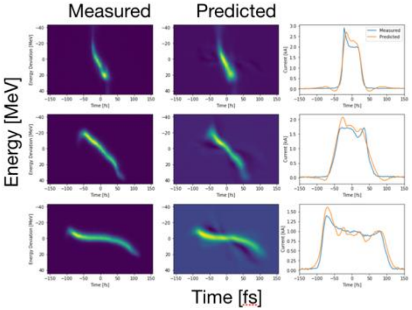


Roussel et. al. *PRL.* 2023

ML-enhanced diagnostics

Rapid analysis/virtual diagnostics

Shot-to-shot predictions at beam rate



C. Emma, et al. – PRAB **21**, 112802 (2018)

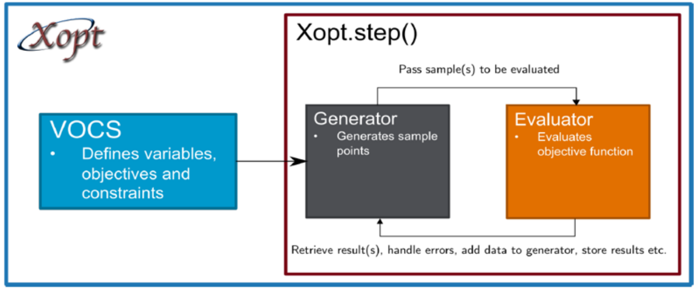
Many solutions put into reusable open-source software (e.g. Xopt/Badger) demoed at many facilities

AI/ML enables fundamentally new capabilities across a broad range of applications → highly promising from initial demos.

Modular, Open-Source Software Development

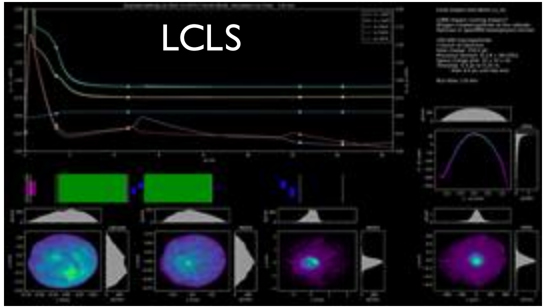
Community development of **re-usable, reliable, flexible software tools** for AI/ML workflows has been essential to maximize return on investment and ensure transferability between systems

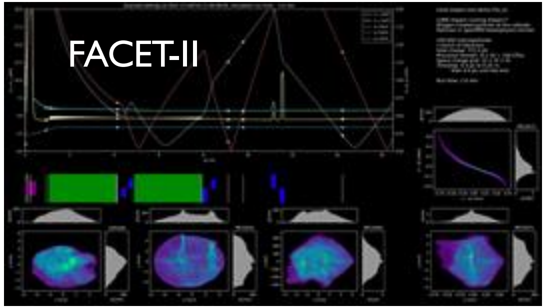
Modularity has been key: separating different parts of the workflow + using shared standards




```
vocs:
  name: TNK_test
  variables:
    x1: [0, 3.14159]
    x2: [0, 3.14159]
  objectives: {y1: MINIMIZE}
  constraints:
    c1: [GREATER_THAN, 0]
    c2: ['LESS_THAN', 0.5]
```

```
algorithm:
  name: bayesian_exploration
  options:
      n_initial_samples: 5
      n_steps: 25
      generator_options:
          batch_size: 1
          #sigma: [[0.01, 0.0],
          use_gpu: False
```





Online Impact-T simulation and live display; trivial to get running on FACET-II using same software tools as the LCLS injector

Different software for different tasks:

Optimization algorithm driver *(e.g. Xopt)*

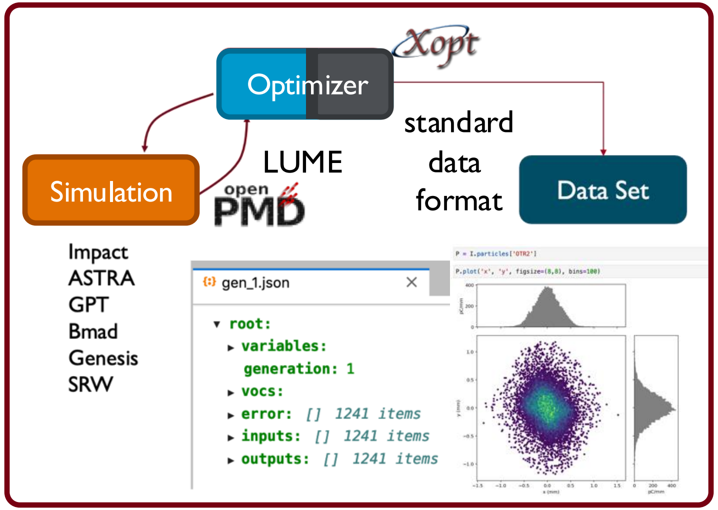
Visual control room interface *(e.g. Badger)*

Simulation drivers *(e.g. LUME)*

Standards model descriptions, data formats, and software interfaces *(e.g. openPMD)*

Online model deployment *(LUME-services)*

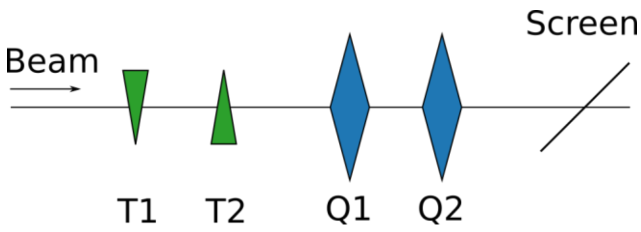
More details at https://www.lume.science/



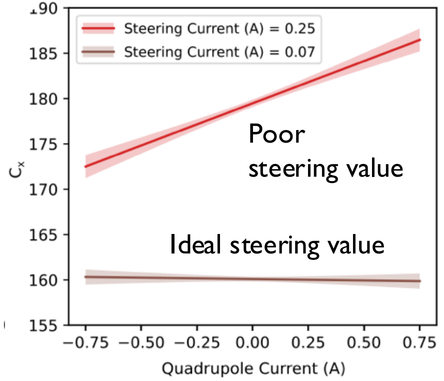
Modular open-source software has been essential for our work.

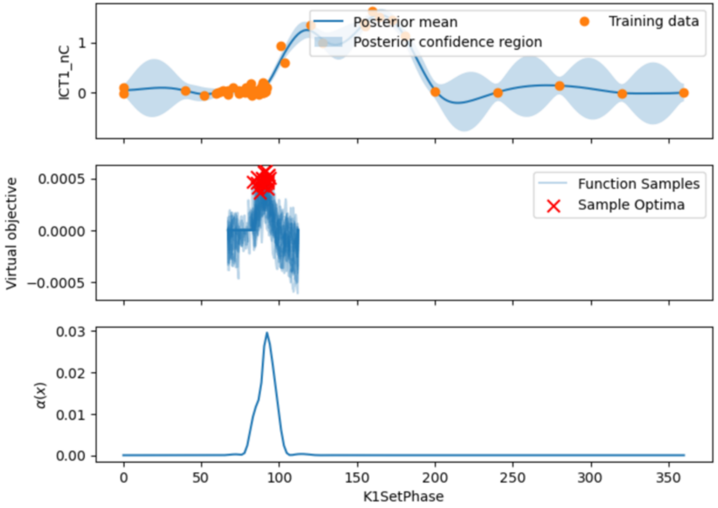
Further Automation

- Chaining together automation of sub-tasks and measurements
- RF /laser timing scans, beamline alignment, smart sampling for measurements

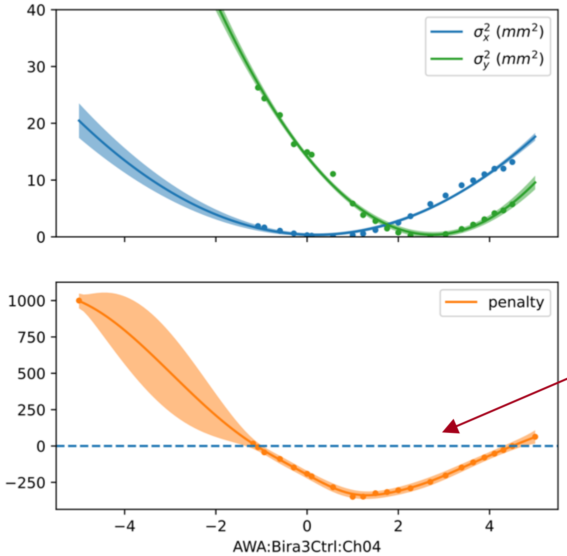


Automated beam alignment
→ 20-30 minutes by hand
→ 5 minutes with BAX





Automated determination of gun phase with BAX



Smart sampling for emittance measurements with Bayesian Exploration

Beam bounding box penalty

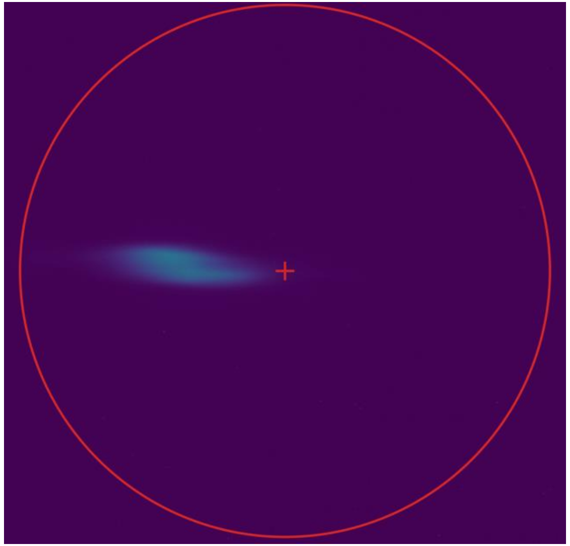
R. Roussel, D. Kennedy

Incorporating Constraints

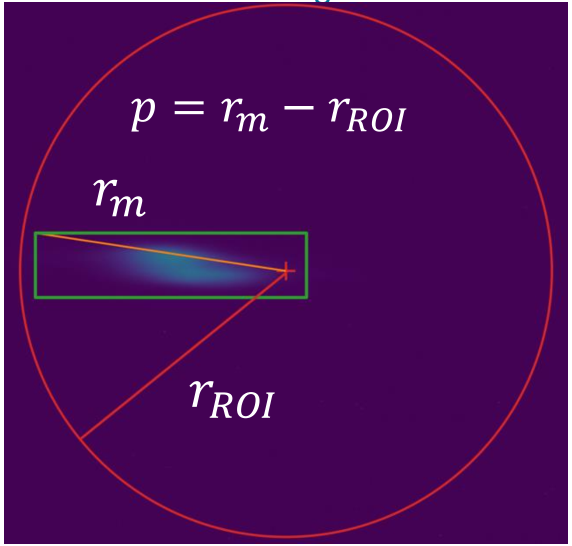
We want to ensure during measurements that the beam stays on screen

→ Define a **smoothly varying** penalty function to act as a constraint

Define a circular ROI

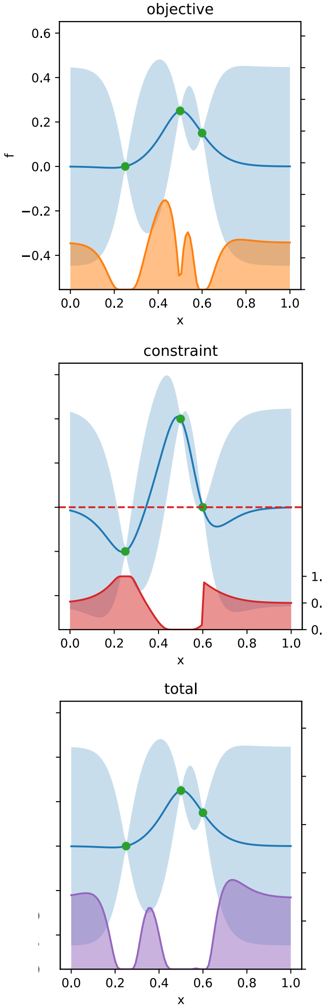


Measure maximum distance from the ROI center to bounding box corners



Constraint: $p \leq 0$

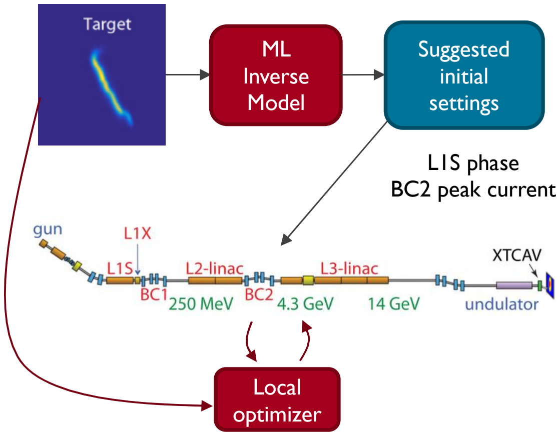
Other examples: Beam losses, dark current production, emittance, etc.



Gardner et. al. ICML 2014

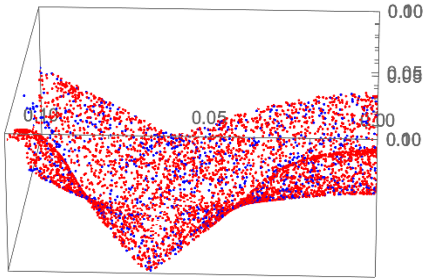
See R. Roussel et al., PRAB (2024) https://journals.aps.org/prab/abstract/10.1103/PhysRevAccelBeams.27.084801

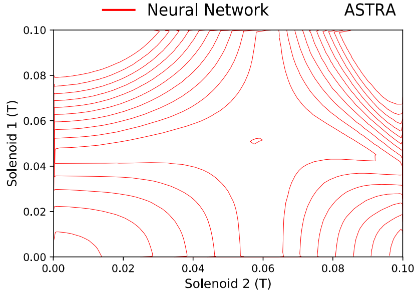
Warm starts for optimization



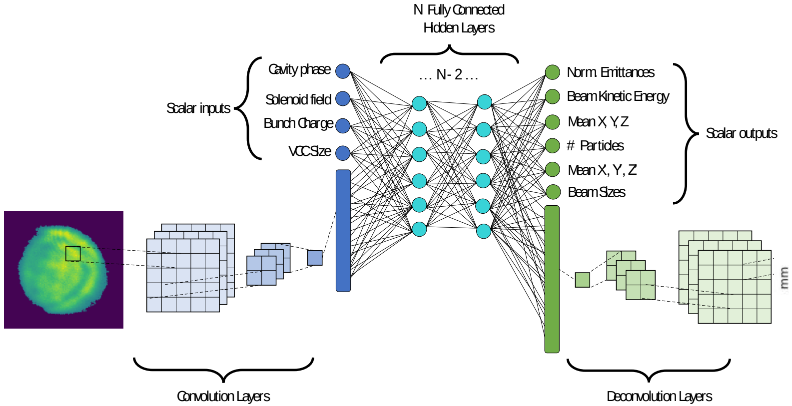
A. Scheinker, A. Edelen, et al, PRL, 2018

Smooth interpolation
Example σ_x surface from 2D scan, LCLS-II Injector

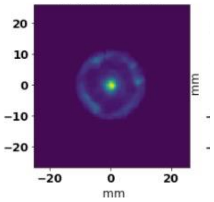


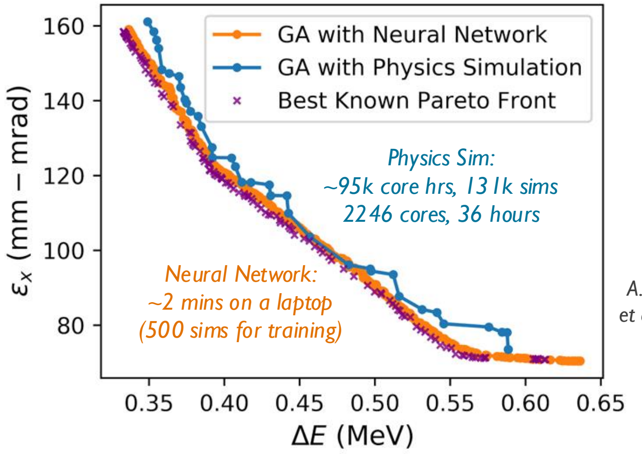


A. Edelen et al., NeurIPS 2019



L. Gupta, et al, MLST, 2021





A. Edelen et al., PRAB, 2020

Include high-dimensional input information à better output predictions

Surrogate-boosted design optimization