BNL Collaboration Meeting
August 22, 2024

Applications of international ML experiences in the Linac to RHIC chain

Auralee Edelen (on behalf of the whole team)
edelen@slac.stanford.edu

Relevant Developments from 4th ICFA Workshop

FPGA / RL at KARA

- Real-time training and deployment for betatron oscillations and microbunching instability
- Likely many practical aspects to learn from them

Many labs building out digital twin / ML-ops tools and infrastructure

- Trying to unify to common standards/tools/infrastructure (ISIS, SLAC, FNAL, ORNL, Jlab, RadiaSoft)
- Continual learning → demonstration at ALS for beam size correction

Differentiable simulations and modular ML models are clearly a major path forward for digital twins, model calibration, advanced diagnostics and integration into control

- Bmad-X demonstrations for 4D and 6D phase space reconstruction; Bmad Julia project
- Cheetah → effort to build python-wrapped simulation tool for fast/approximate differentiable simulations
- LBNL → integrate modular surrogates into plasma simulation chain (e.g. just for plasma stages)

Many exciting applications of LLMs and computer vision (elog + measured data synthesis, LLM verbal commands for tuning, literature → chatbot, visual ID of beamline elements in tunnel)

Badger/Xopt being used at a lot of facilities: SLAC, AWA, ESRF, DESY

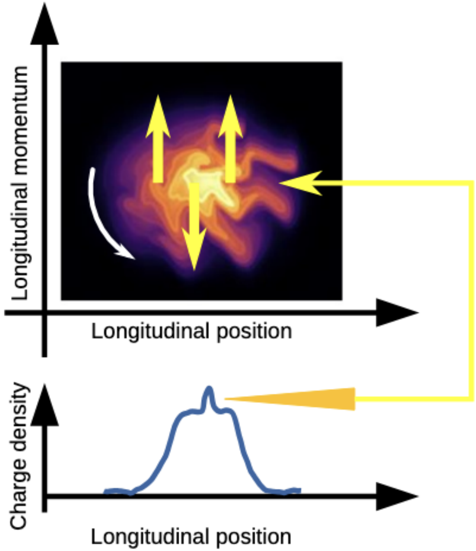
Timetable/slides

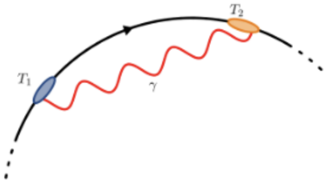
Hard problem: Microbunching Instability

Unstable coherent synchrotron radition (THz) production

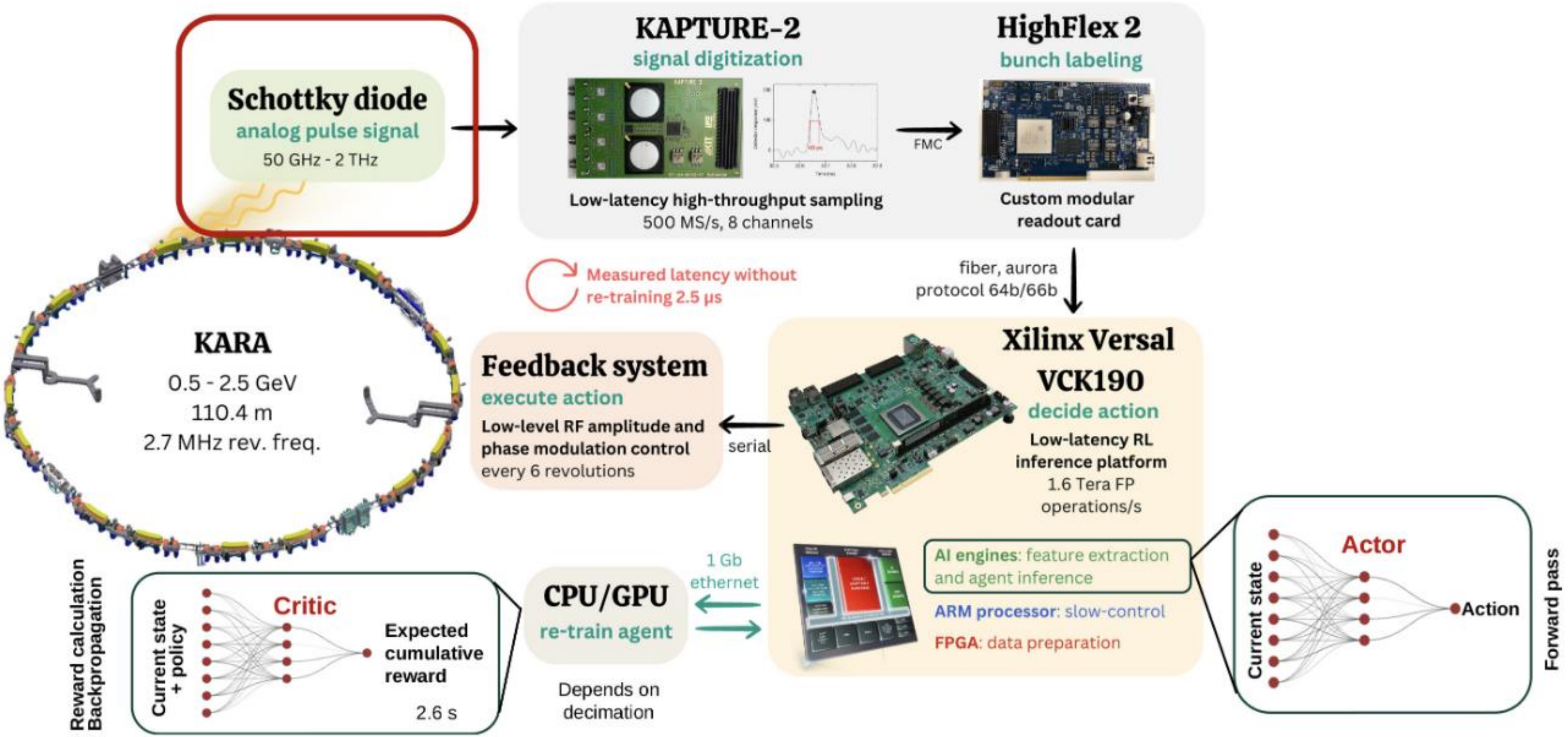
- Self-interaction of bunch with emitted radiation
- Nonlinear dynamics, several timescales/frequency components
- Main timescales: $O(10\,\mu s)$, $O(10\,ms)$, with $T_s = O(100\,\mu s)$
- Expensive to simulate!

Perfect candidate for real time RL!





https://www.indico.kr/event/47/contributions/537/attachments/500/1173/presentation.pdf



Schottky diode
analog pulse signal
50 GHz - 2 THz

KAPTURE-2
signal digitization
Low-latency high-throughput sampling
500 MS/s, 8 channels

FMC

HighFlex 2
bunch labeling
Custom modular readout card

fiber, aurora protocol 64b/66b

Measured latency without re-training 2.5 μs

KARA
0.5 - 2.5 GeV
110.4 m
2.7 MHz rev. freq.

Feedback system
execute action
Low-level RF amplitude and phase modulation control every 6 revolutions

serial

Xilinx Versal VCK190
decide action
Low-latency RL inference platform
1.6 Tera FP operations/s

AI engines: feature extraction and agent inference
ARM processor: slow-control
FPGA: data preparation

Actor
Current state
Action
Forward pass

1 Gb ethernet

CPU/GPU
re-train agent
Depends on decimation

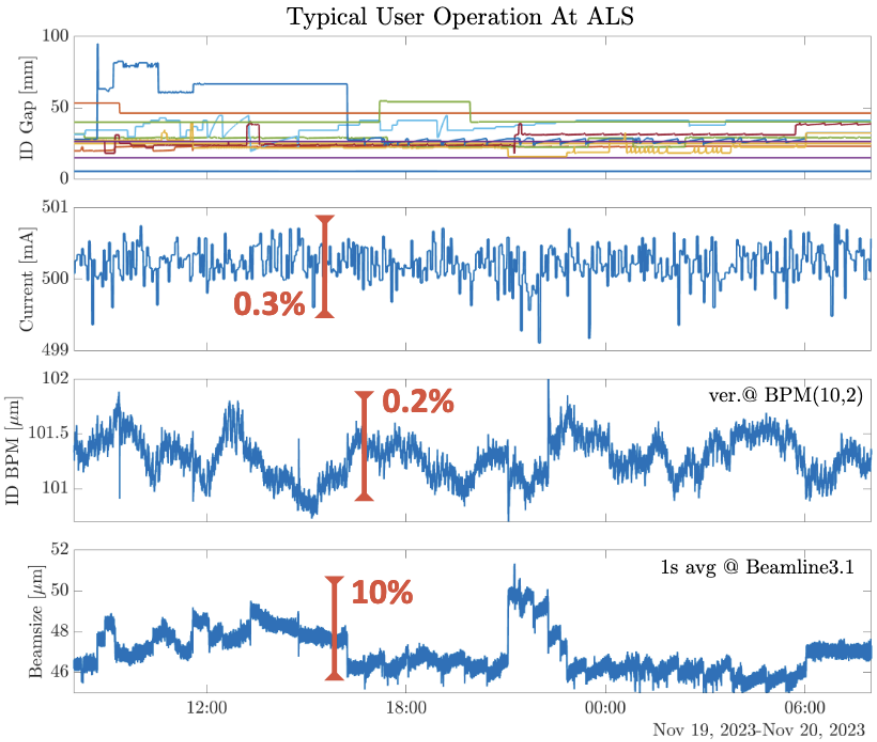
Critic
Current state + policy
Expected cumulative reward
2.6 s
Reward calculation Backpropagation

Courtesy Andrea Santamaria Garcia

https://www.indico.kr/event/47/contributions/537/attachments/500/1173/presentation.pdf

Electron Beam Stability at the Advanced Light Source

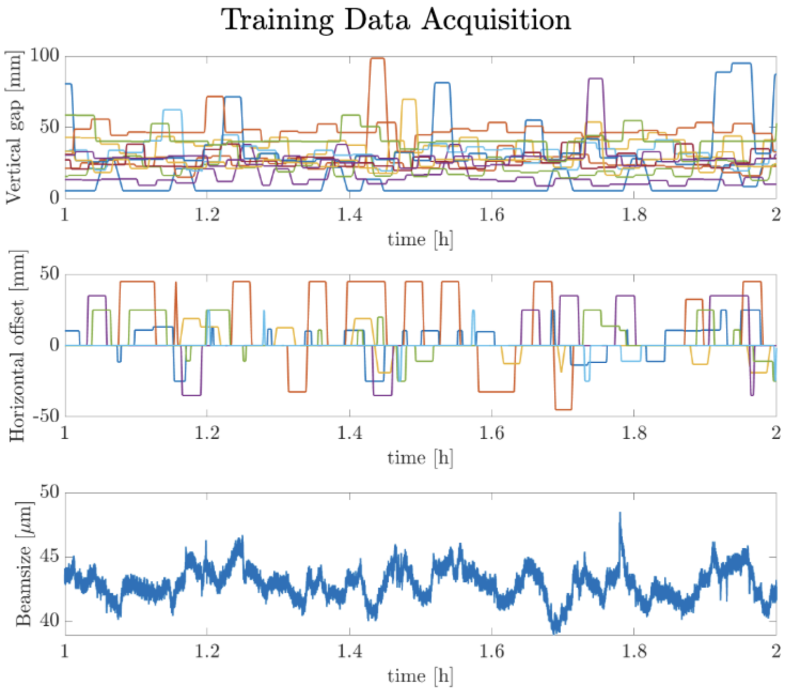
- Beam Current:
  - Top-off operation keeps current variations below 1mA
- Beam Position:
  - Orbit feedback and ID feed-forwards stabilize source positions to sub-micron level
- Beam Size:
  - ID skew quadrupole feed-forwards stabilize source size
  - Requires lookup tables



https://www.indico.kr/event/47/contributions/529/attachments/513/1191/ALS_beam_size_control.pdf

Acquiring Training Data

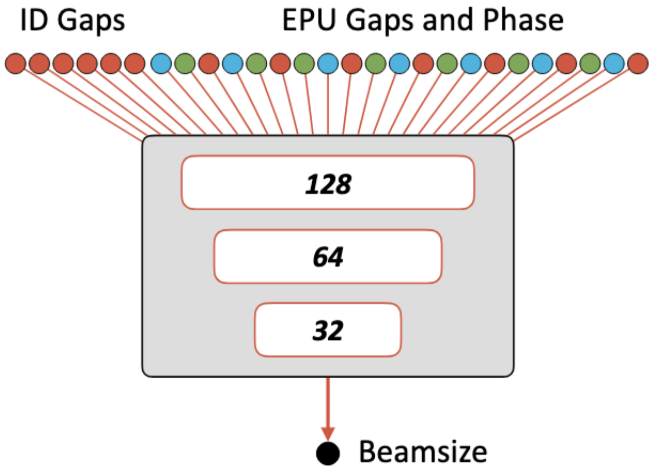
- Data Sampling:
  - Derived from two years of user operation data
  - Ensures representative operational conditions
- Data Acquisition and Recording:
  - Gathered during accelerator physics shifts
  - Independent exercise of each insertion device
  - All ID read-backs and beam size recorded at 10Hz
  - EPICS based archiver system
  - 12-hour, 27 ID parameters (466k x 27 samples)
- Operational Challenges:
  - High value of AP time leads to nighttime shifts
  - ID setup not optimized for fast ramping (ID amplifier trips, local ID FF trips)
  - Implementation of watchdog with for operational oversight very important

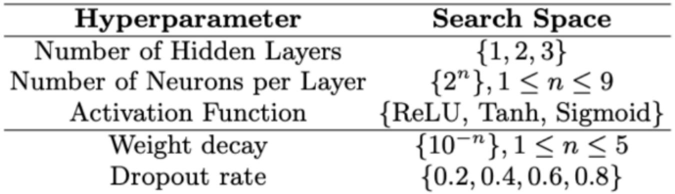


https://www.indico.kr/event/47/contributions/529/attachments/513/1191/ALS_beam_size_control.pdf

Neural Network Architecture

- Model Input/Output:
  - 27 ID input parameters
  - 1 beam size prediction output
  - Dispersion wave used to correct beamsize
- Studied Neural Network Types:
  - RNN, CNN, ***MLP***
- MLP Hyperparameter Search:
  - Number of hidden layers: 3
  - Neurons per Layer: 128/64/32
  - Activation Function: Tanh
- Final Hyperparameter Search:
  - Weight decay: 1E-3
  - Dropout rate: 0.2
- Takes about 15min on RTX2060 GPU



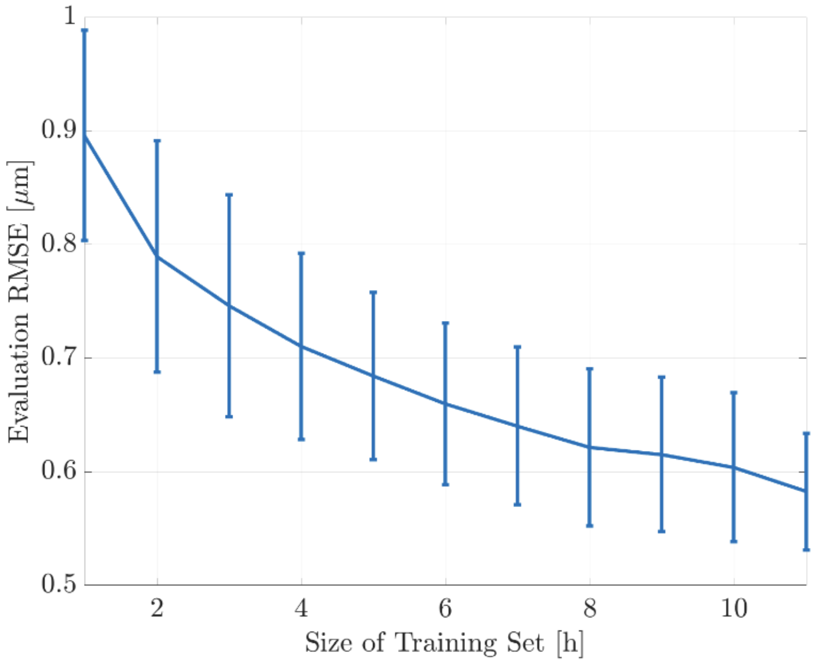


Hyperparameter	Search Space
Number of Hidden Layers	$\{1, 2, 3\}$
Number of Neurons per Layer	$\{2^n\}, 1 \leq n \leq 9$
Activation Function	{ReLU, Tanh, Sigmoid}
Weight decay	$\{10^{-n}\}, 1 \leq n \leq 5$
Dropout rate	$\{0.2, 0.4, 0.6, 0.8\}$

https://www.indico.kr/event/47/contributions/529/attachments/513/1191/ALS_beam_size_control.pdf

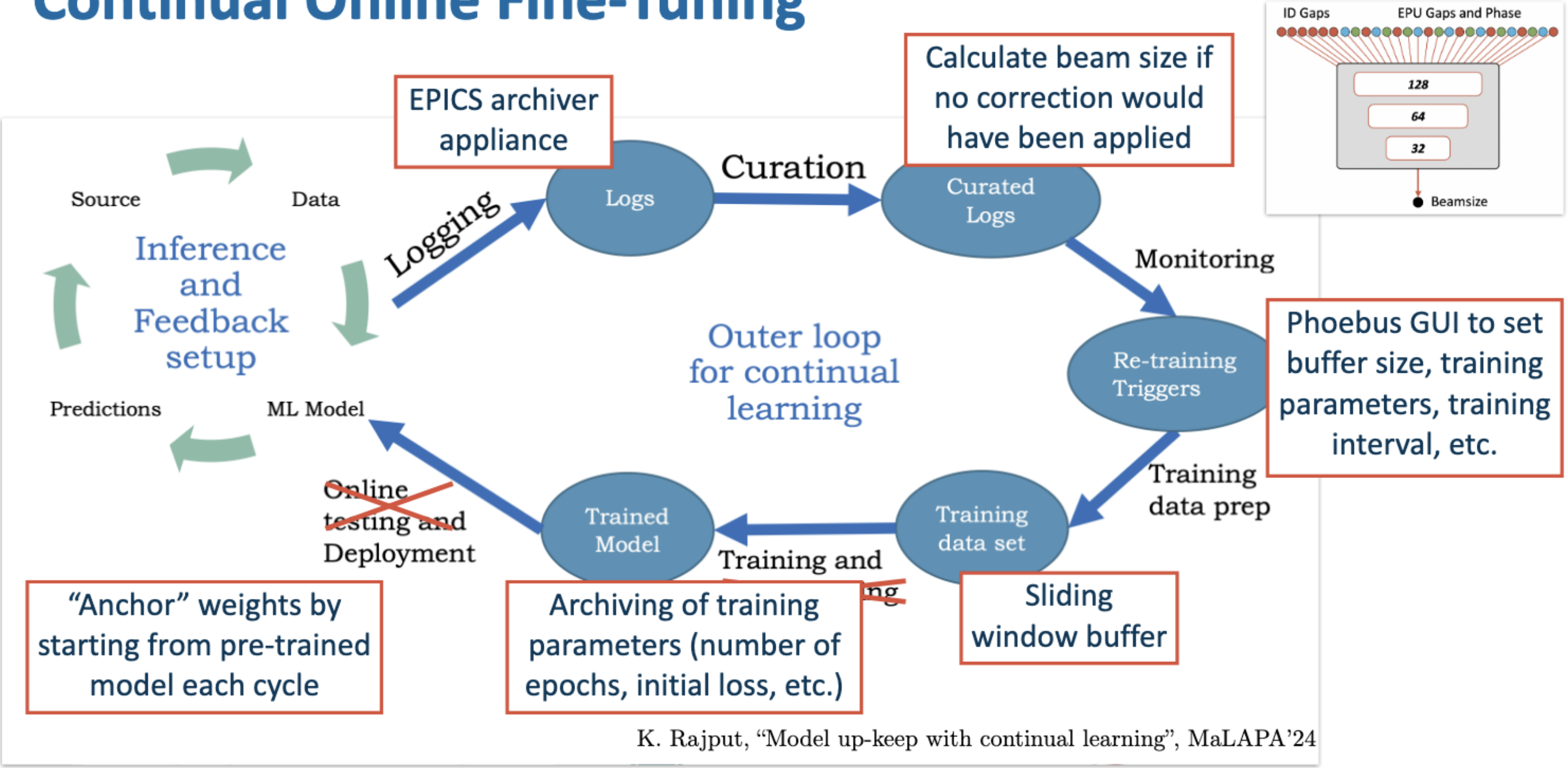
Impact of Training Data Size on Model Performance

- How much training time is required for *perfect* model?
- Chronologically Split Data:
  - Can not randomly select datapoints for evaluation (oversampling at 10Hz)
- Evaluation Procedure:
  - Remove 1h randomly from the data set for evaluation
  - Choose [1,...11]h for training
  - 10 seed for each configuration
  - Evaluate RMSE on evaluation data
- Observed Convergence:
  - Reasonable convergence at first
  - Trend suggests infeasible amount of data required to reach noise level



https://www.indico.kr/event/47/contributions/529/attachments/513/1191/ALS_beam_size_control.pdf

Continual Online Fine-Tuning



EPICS archiver appliance

Calculate beam size if no correction would have been applied

Phoebus GUI to set buffer size, training parameters, training interval, etc.

Sliding window buffer

Archiving of training parameters (number of epochs, initial loss, etc.)

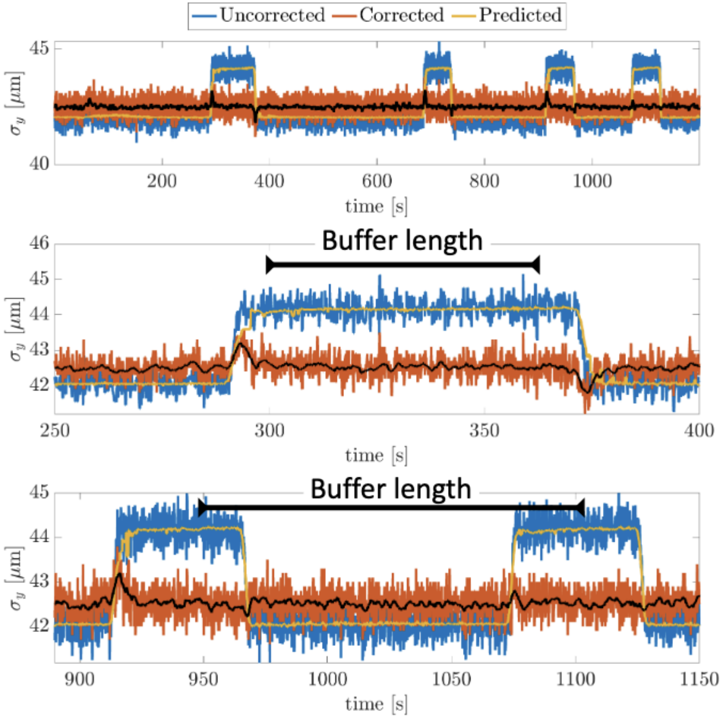
“Anchor” weights by starting from pre-trained model each cycle

K. Rajput, “Model up-keep with continual learning”, MaLAPA’24

https://www.indico.kr/event/47/contributions/529/attachments/513/1191/ALS_beam_size_control.pdf

Continual Online Fine-Tuning

- Online Fine-Tuning:
  - Circular buffer to record model input
  - Train base model on data in buffer only
  - Start from base model each cycle to avoid runaway
  - Uncorrected beamsize calculated with DWP
- Parameters:
  - Typically 1k samples in buffer
  - Takes less then 100 epochs and about 1s
- Feedback vs Feedforward:
  - Online retraining acts as feedback
  - Buffer size controls impact of FB vs. FF



https://www.indico.kr/event/47/contributions/529/attachments/513/1191/ALS_beam_size_control.pdf

Digital Twin Infrastructure

Ecosystem of modular tools (can use independently)

LUME – simulation interfaces/wrappers in Python

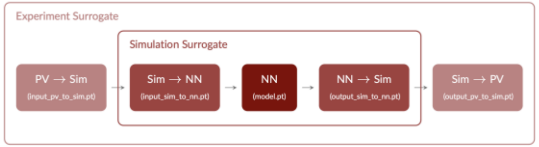
lume-model – wraps ML models, facilitates calibration

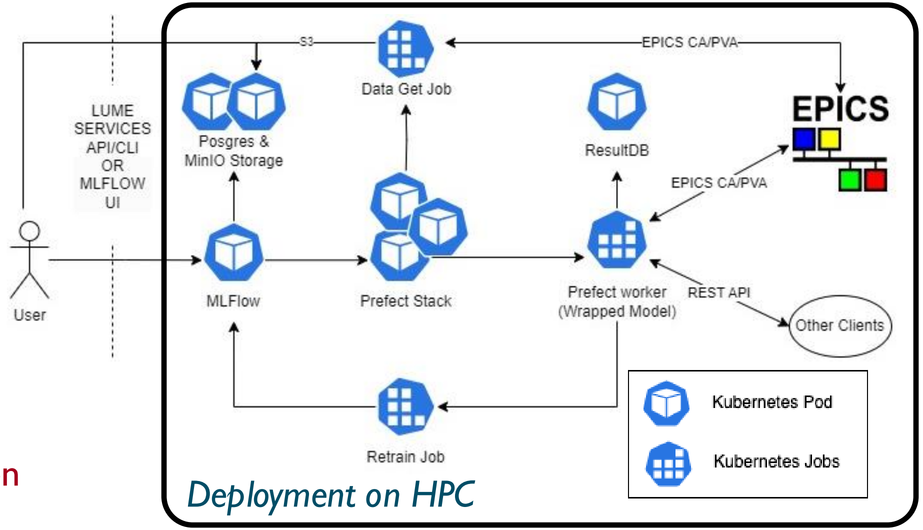
lume-services – online model deployment and orchestration

distgen – flexible creation of beam distributions

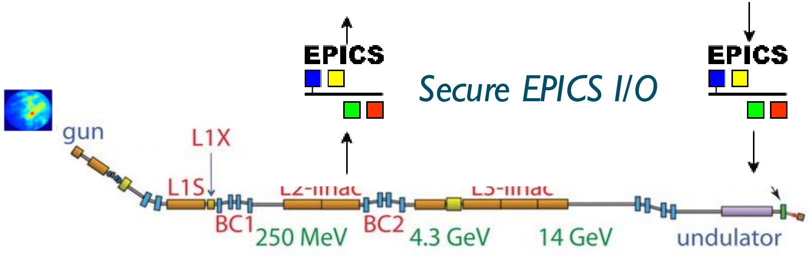
Integration with MLFlow for MLOps

https://www.lume.science/


Experiment Surrogate
Simulation Surrogate
PV → Sim (input_pv_to_sim.pt)
Sim → NN (input_sim_to_nn.pt)
NN (model.pt)
NN → Sim (output_sim_to_nn.pt)
Sim → PV (output_pv_to_sim.pt)



- 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

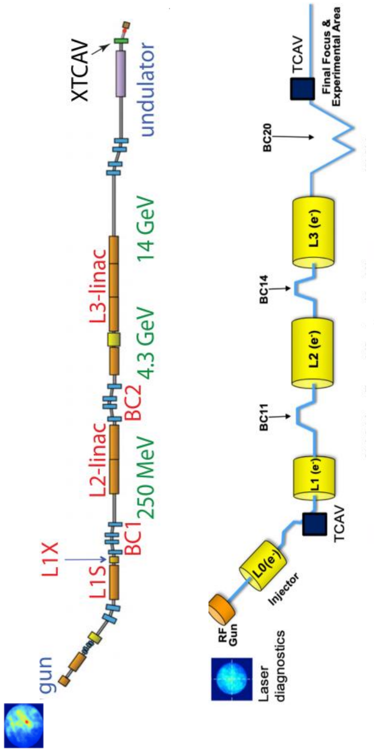


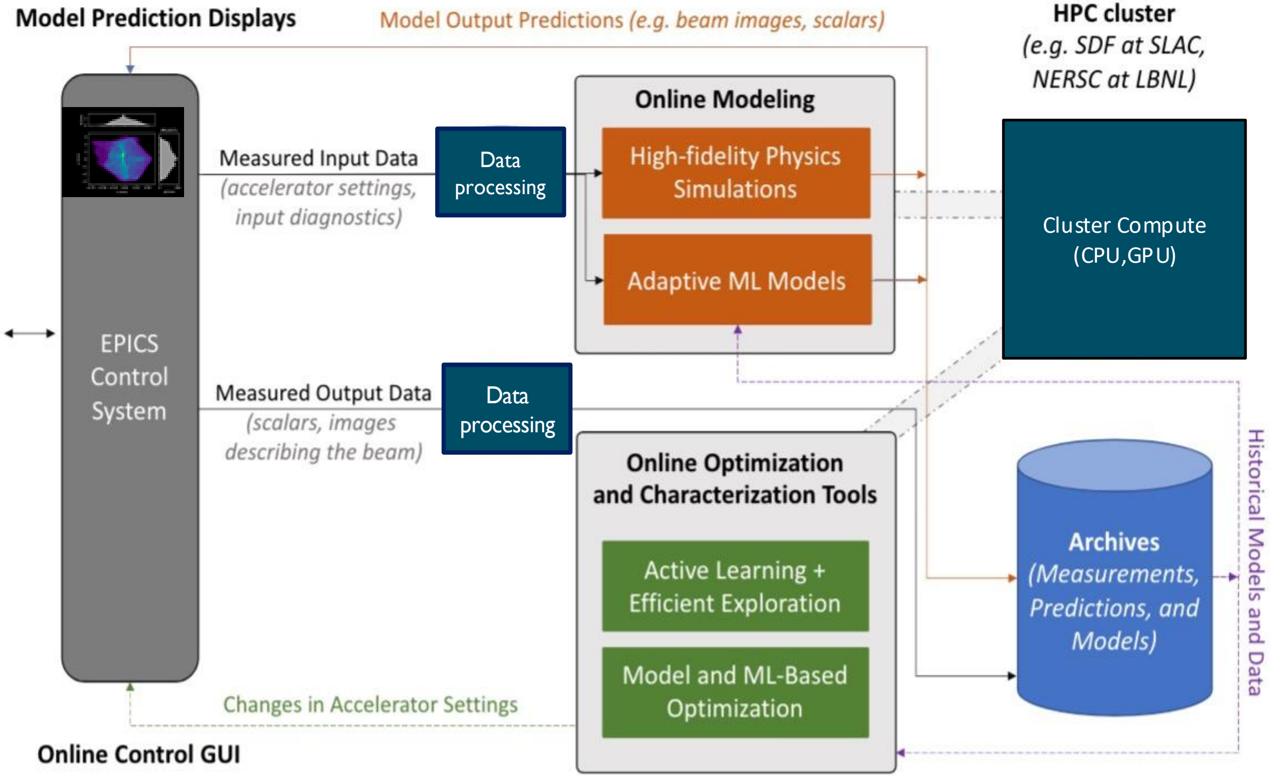
Substantial progress on deploying ML and Physics-based models and integrating with HPC in a portable way

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)*





Model Prediction Displays

Model Output Predictions *(e.g. beam images, scalars)*

HPC cluster

(e.g. SDF at SLAC, NERSC at LBNL)

EPICS Control System

Measured Input Data

(accelerator settings, input diagnostics)

Data processing

Online Modeling

High-fidelity Physics Simulations

Adaptive ML Models

Cluster Compute (CPU,GPU)

Measured Output Data

(scalars, images describing the beam)

Data processing

Online Optimization and Characterization Tools

Active Learning + Efficient Exploration

Model and ML-Based Optimization

Archives

(Measurements, Predictions, and Models)

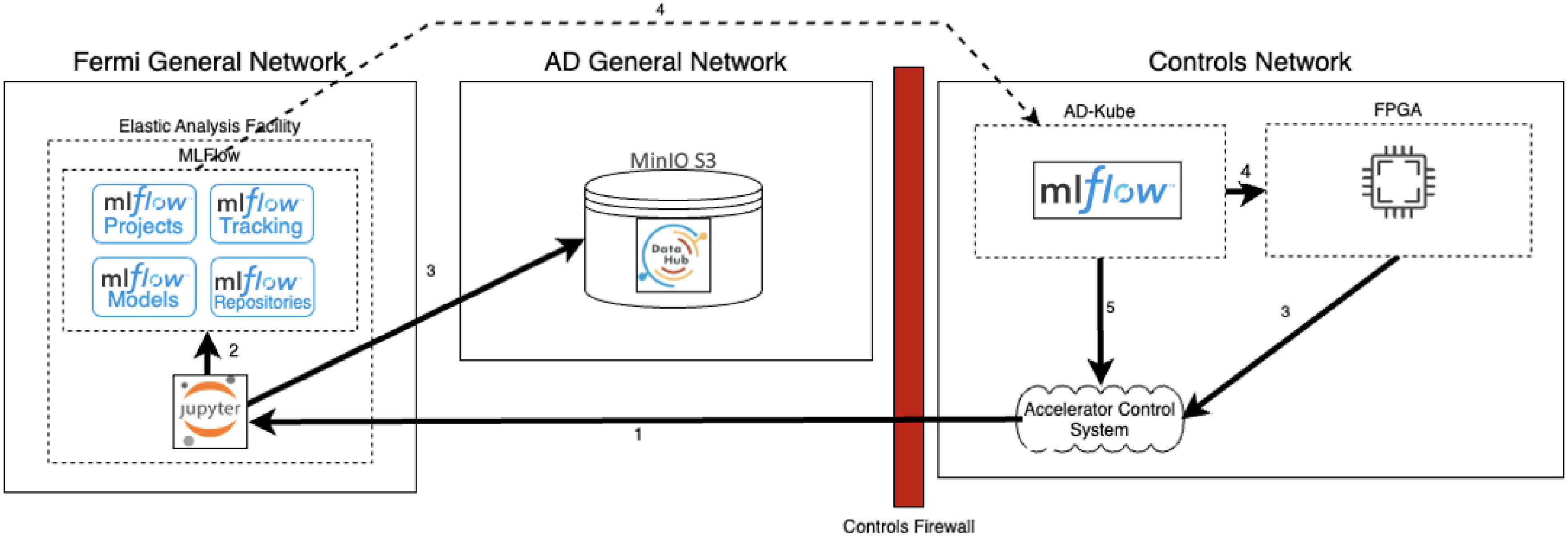
Historical Models and Data

Changes in Accelerator Settings

Online Control GUI

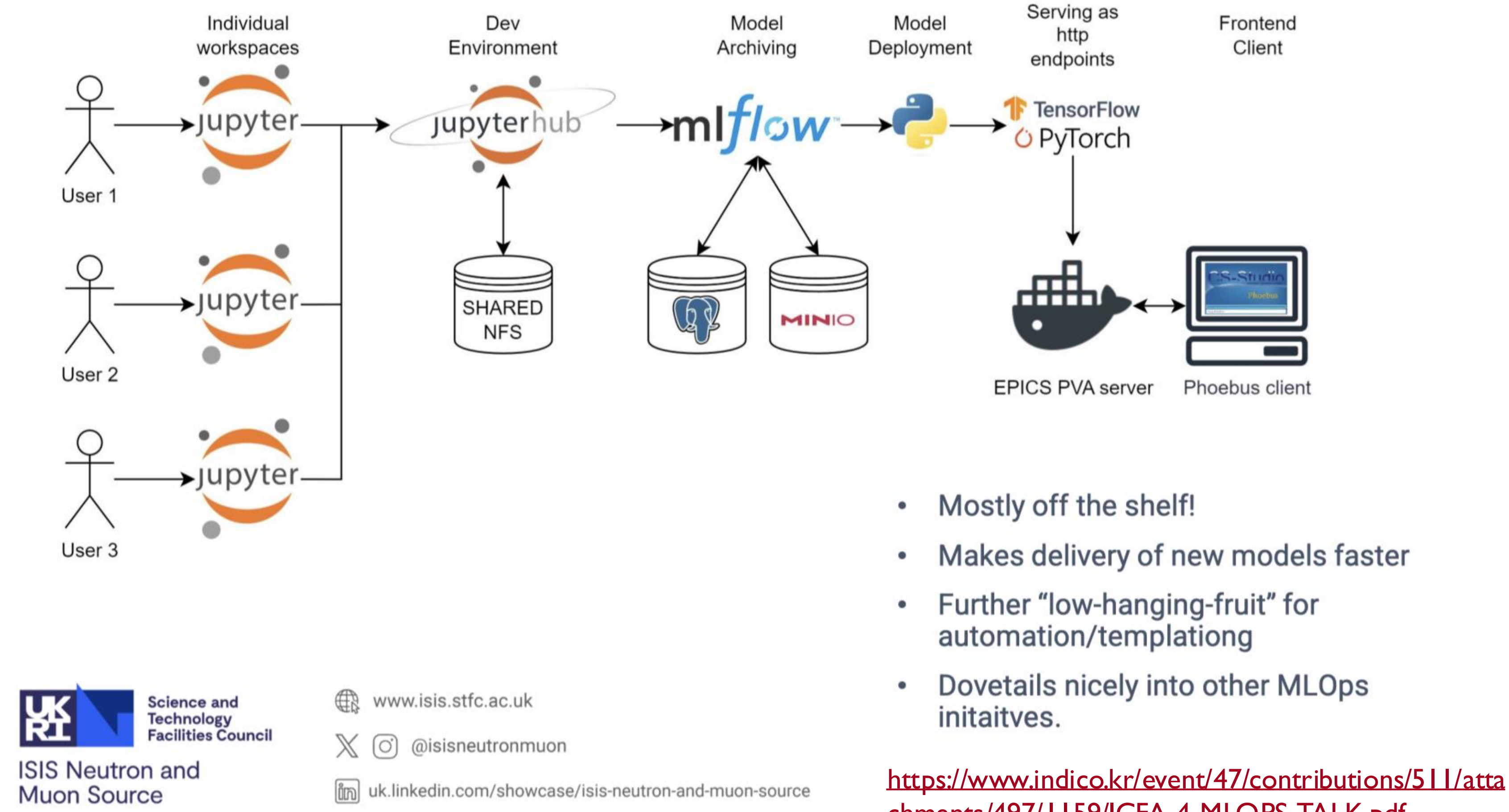
Making good progress toward this vision with open-source, modular software tools

ML Inference Infrastructure - FNAL



https://www.indico.kr/event/47/contributions/534/attachments/477/1110/Presentation4-malapa.pptx

ML Inference Infrastructure - ISIS



- Mostly off the shelf!
- Makes delivery of new models faster
- Further "low-hanging-fruit" for automation/templationg
- Dovetails nicely into other MLOps initaitves.

Science and Technology Facilities Council

ISIS Neutron and Muon Source

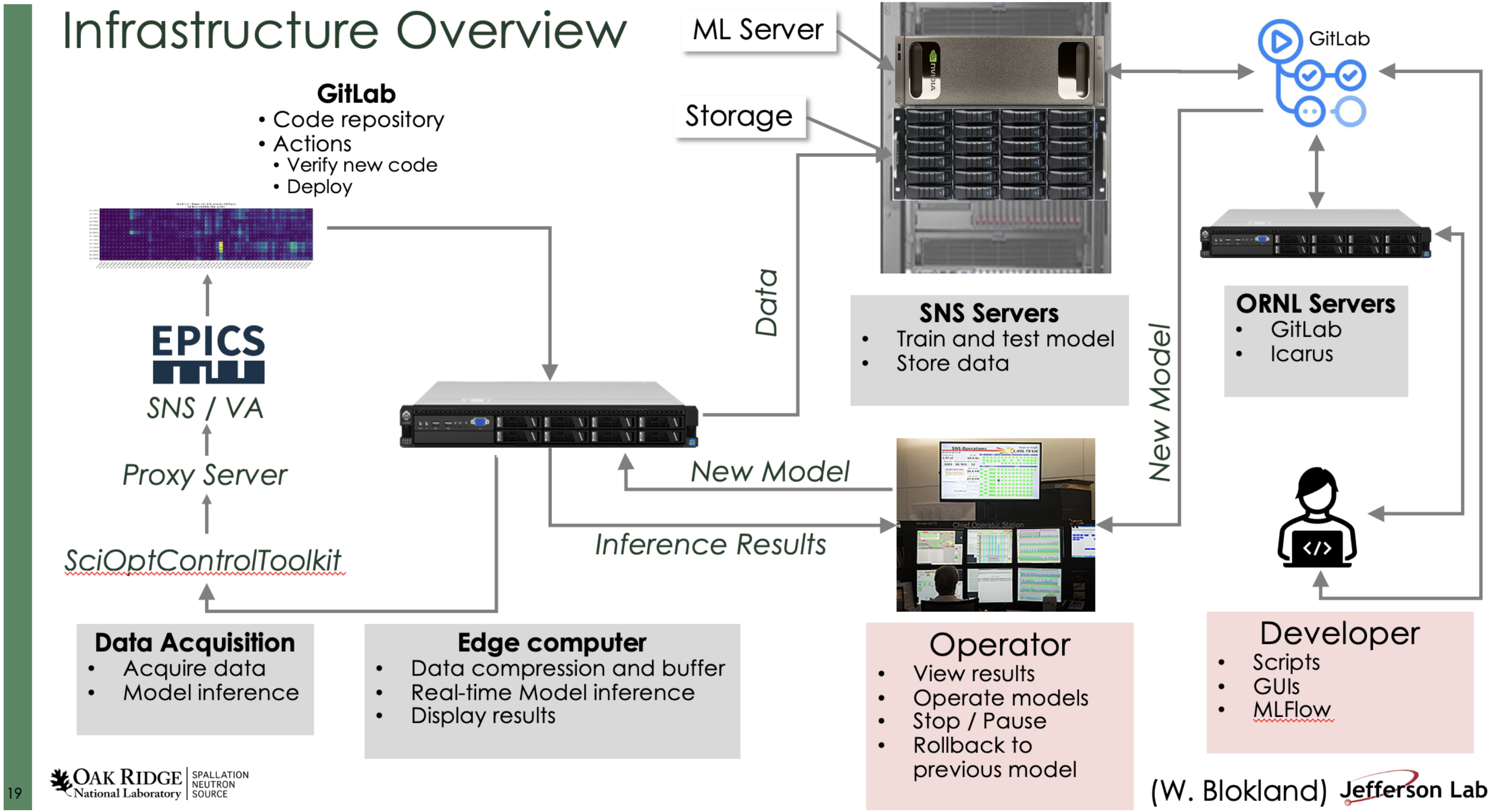
www.isis.stfc.ac.uk

@isisneutronmuon

uk.linkedin.com/showcase/isis-neutron-and-muon-source

https://www.indico.kr/event/47/contributions/511/attachments/497/1159/ICFA-4-MLOPS-TALK.pdf

Digital Twin Infrastructure – ORNL / JLab



Infrastructure Overview

GitLab
- Code repository
- Actions
  - Verify new code
  - Deploy

ML Server

Storage

GitLab

Data

New Model

New Model

Inference Results

EPICS

SNS / VA

Proxy Server

SciOptControlToolkit

SNS Servers
- Train and test model
- Store data

ORNL Servers
- GitLab
- Icarus

Data Acquisition
- Acquire data
- Model inference

Edge computer
- Data compression and buffer
- Real-time Model inference
- Display results

Operator
- View results
- Operate models
- Stop / Pause
- Rollback to previous model

Developer
- Scripts
- GUIs
- MLFlow

OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE

(W. Blokland) Jefferson Lab

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

less ← assumed knowledge of machine → more

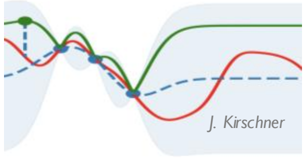
Model-Free Optimization

Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization



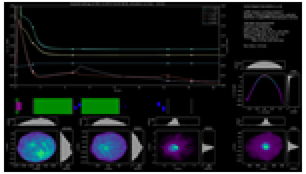
J. Kirschner

Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



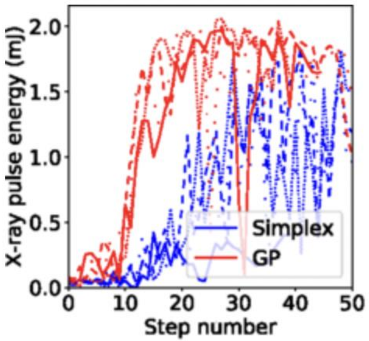
→ provide initial guess (i.e. warm start)
→ provide insight to operators
→ model-based control

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.

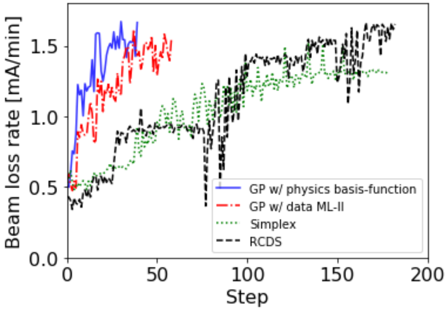
Many successes with Bayesian Optimization *(+ algorithmic improvements)*

FEL pulse energy tuning at LCLS



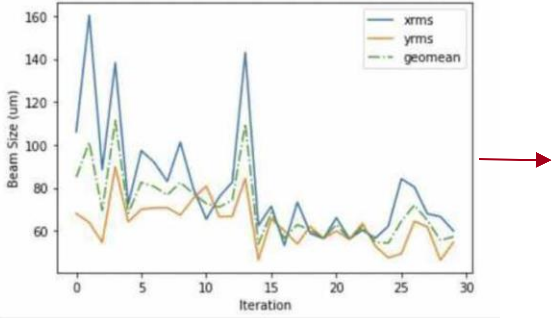
Duris et. al. PRL , **2020**

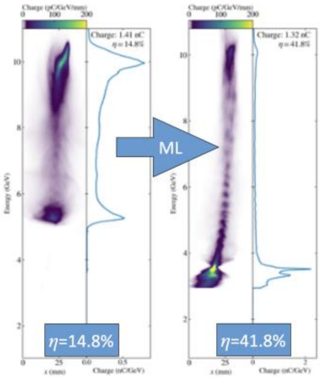
Loss rate tuning at SPEAR3



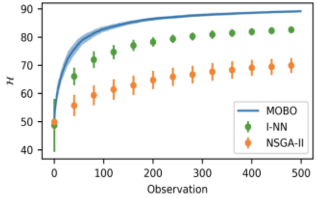
Hanuka et. al. PRAB , **2021**

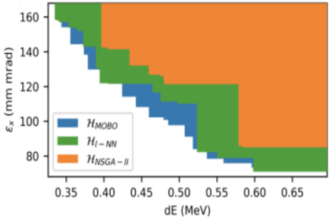
Sextupole tuning at FACET-II
2x efficiency of acceleration in plasma



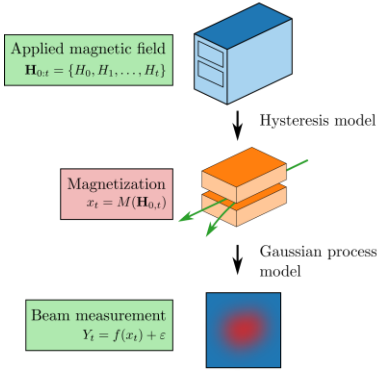


Multi-objective Bayesian Optimization





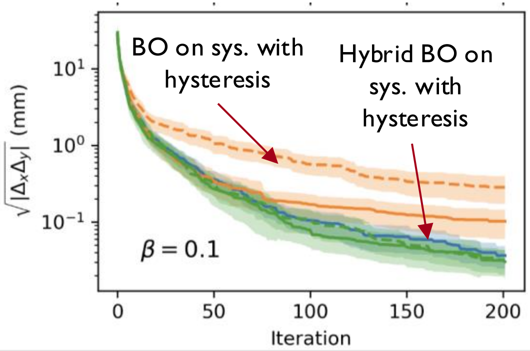
Roussel et. al. PRAB , **2021**



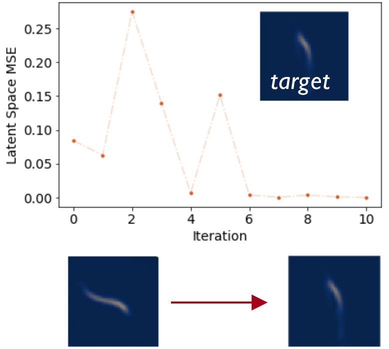
Applied magnetic field
$\mathbf{H}_{0:t} = \{H_0, H_1, \ldots, H_t\}$
Hysteresis model
Magnetization
$x_t = M(\mathbf{H}_{0,t})$
Gaussian process model
Beam measurement
$Y_t = f(x_t) + \varepsilon$

Roussel et. al. PRL , **2022**

Higher-precision optimization possible when including hysteresis effects in model



Longitudinal phase space tuning on LCLS



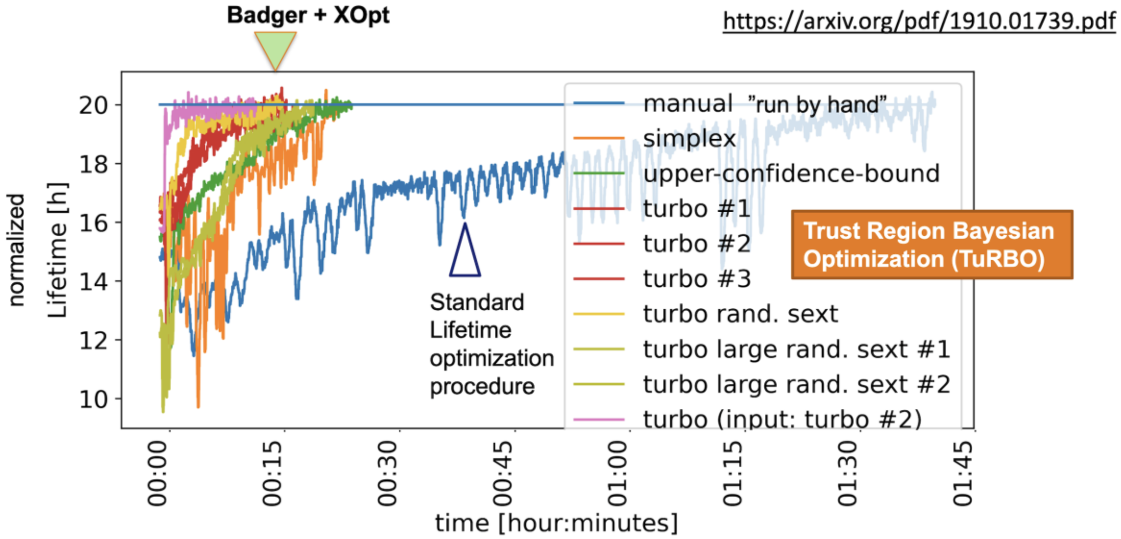
Algorithms being implemented/distributed in Xopt: https://github.com/ChristopherMayes/Xopt
Comprehensive review of advanced BO for particle accelerators: https://arxiv.org/html/2312.05667v2

Xopt

ESRF Results

The Trust region BO (TuRBO) method is now in regular use for lifetime maximization.

10-15 MINUTES INSTEAD OF 100 MINUTES FOR EQUIVALENT OPTIMIZATION



Badger + XOpt

https://arxiv.org/pdf/1910.01739.pdf

Standard Lifetime optimization procedure

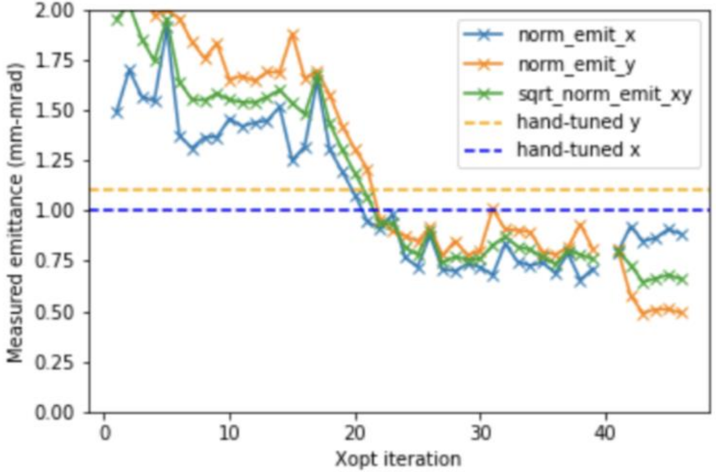
Trust Region Bayesian Optimization (TuRBO)

Slide courtesy S.M. Liuzzo et al., LER 2024

https://indico.cern.ch/event/1326603/contributions/5773963/attachments/2799269/4894488/LER_2024.pdf

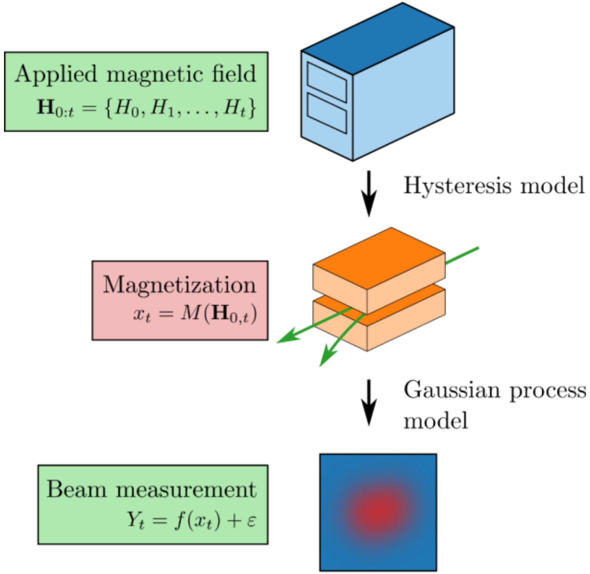
Bayesian optimization of emittance at LCLS-II

- **Repeatedly used Xopt and regular BO for emittance tuning on OTR0H04**
- Used pyemittance for adaptive emittance measurements
  - Tuning parameters are SOL1, SOL2, SQ1, CQ1, SQ2, CQ2
  - Working on including matching into the objective



Combining GP Modeling with Differentiable Physics

Learn both hysteresis properties and beam response simultaneously using two step modeling



Applied magnetic field $\mathbf{H}_{0:t} = \{H_0, H_1, \ldots, H_t\}$

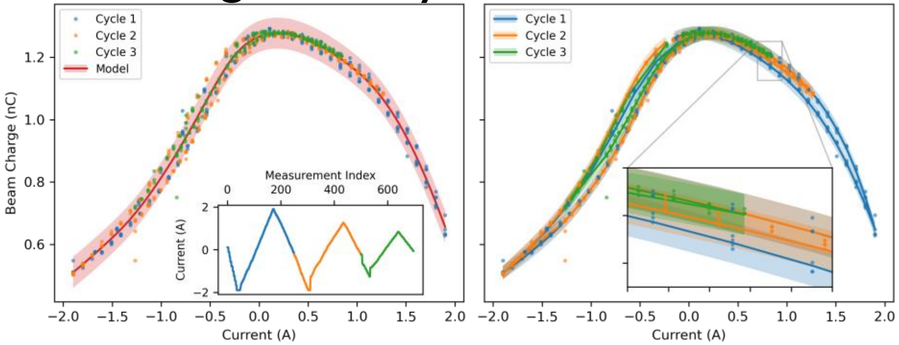
Hysteresis model

Magnetization $x_t = M(\mathbf{H}_{0,t})$

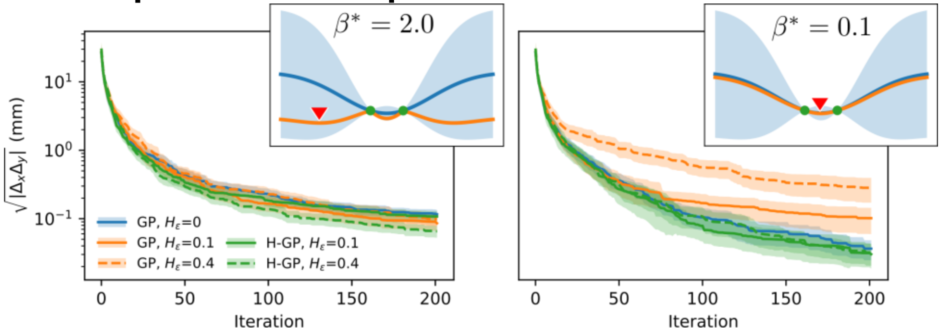
Gaussian process model

Beam measurement $Y_t = f(x_t) + \varepsilon$

Modeling accuracy increases



Optimization performance increases

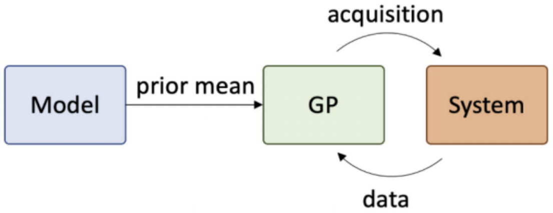


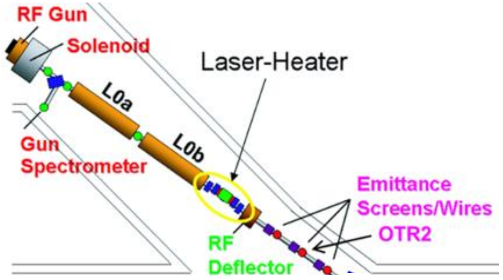
R. Roussel, et. Al. Phys. Rev. Lett. **128**, 204801

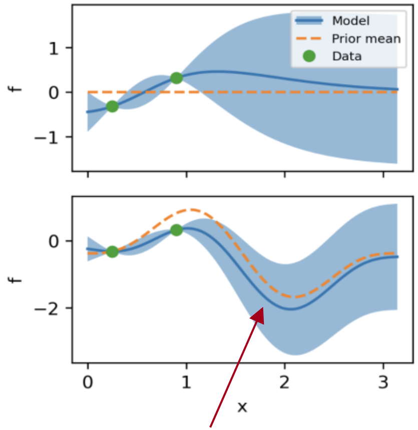
Leveraging Online Models for Faster Optimization

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

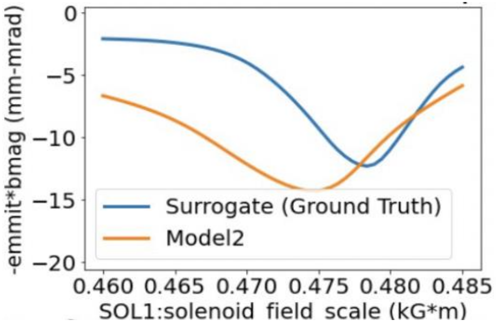
Prototyped on LCLS injector
variables: *solenoid, 2 corrector quads, 6 matching quads*
objective: *minimize emittance and matching parameter*





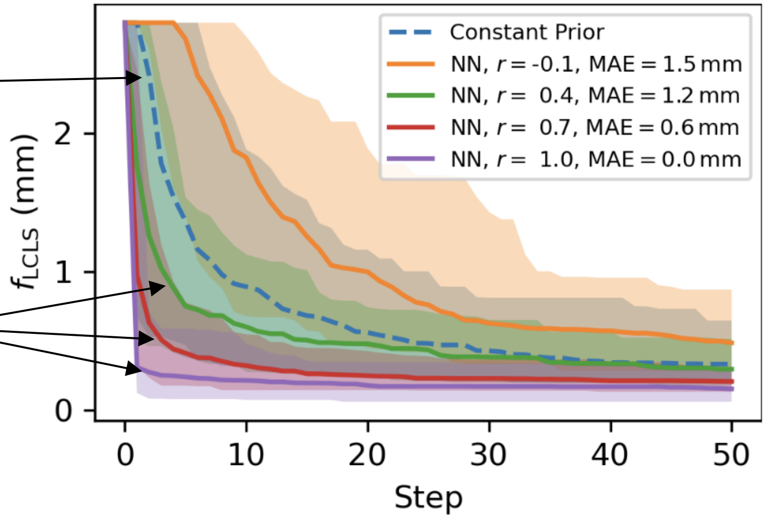


model prediction returns to prior



regular Bayesian optimization

prior mean from models with different fidelity



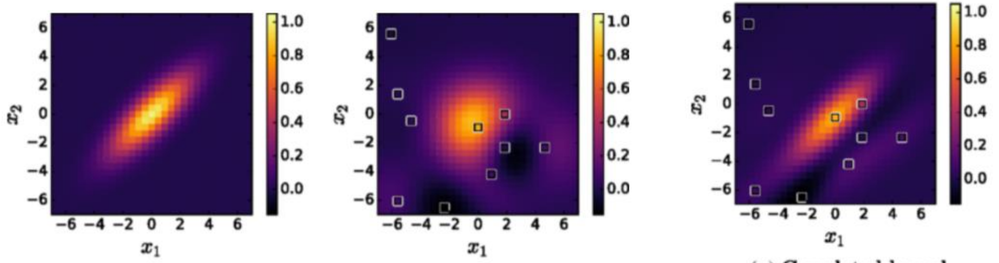
Even prior mean models with substantial inaccuracies provide a boost in optimization speed

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

Bayesian Optimization with Correlated Kernel

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

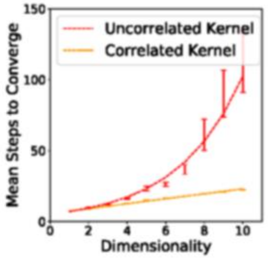
→ Design Gaussian Process kernel from expected correlations between inputs (e.g. quadrupole magnets)

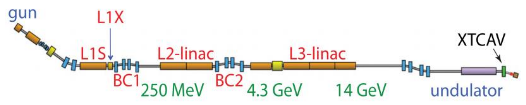


(a) Ground truth

(b) Isotropic kernel

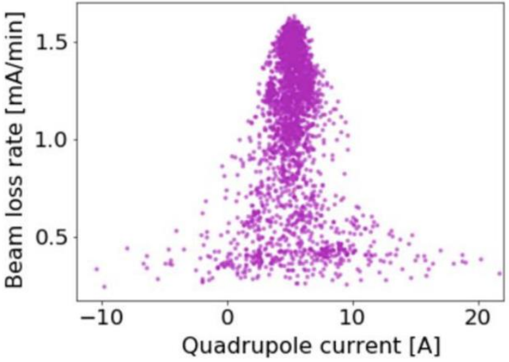
(c) Correlated kernel

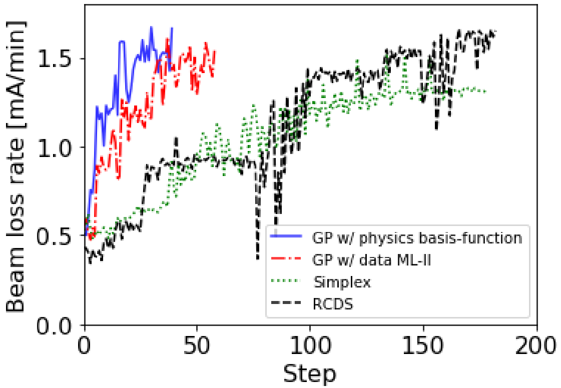


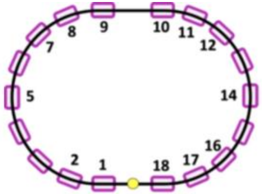


FEL tuning @LCLS

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







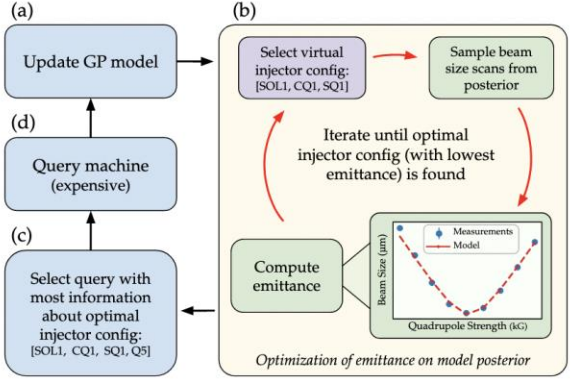
vertical emittance tuning @SPEAR3

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

Efficient Emittance Optimization with Virtual Objectives

- **Instead of tuning on costly emittance measurements directly: learn a fast-executing model online for beam size while optimizing** → *learn on direct observables (e.g. beam size); do inferred "measurements" (e.g. emittance)*
- **New algorithmic paradigm leveraging "Bayesian Algorithm Execution" (BAX) for 20x speedup in tuning**


(a) Update GP model

(d) Query machine (expensive)

(c) Select query with most information about optimal injector config: [SOL1, CQ1, SQ1, Q5]

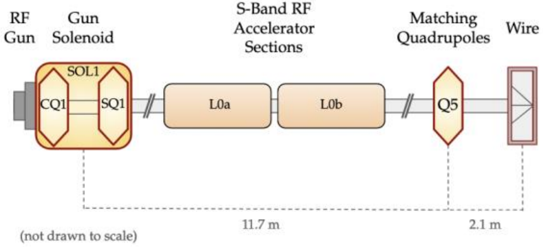
(b) Select virtual injector config: [SOL1, CQ1, SQ1]

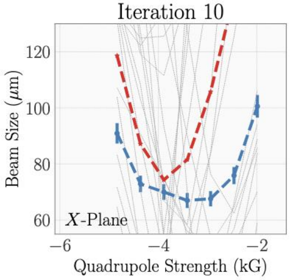
Sample beam size scans from posterior

Iterate until optimal injector config (with lowest emittance) is found

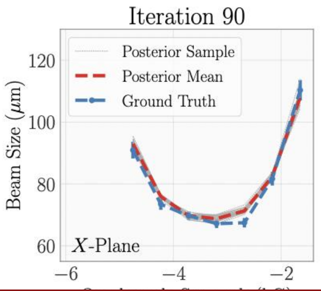
Compute emittance

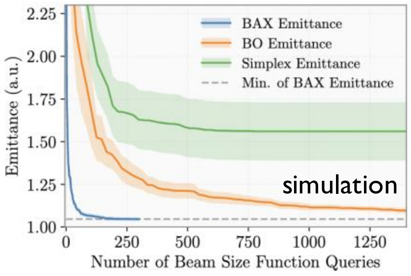
Optimization of emittance on model posterior

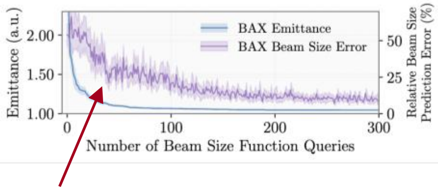




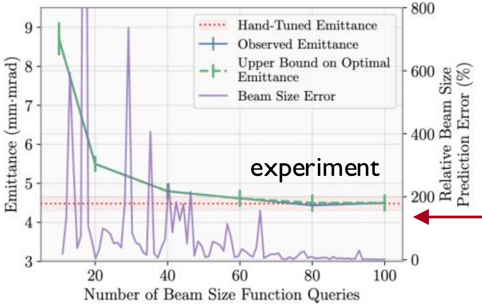
model is learned on-the-fly







Convergence of beam size prediction error gives practical indicator of optimization convergence *(no need to do direct emittance measurement until the end)*



Found equivalent quality to hand-tuning in about 70 iterations *(estimate this would take a few minutes with computationally optimized routine)*

https://arxiv.org/abs/2209.04587

Paradigm shift in how tuning on indirectly computed beam measurements (such as emittance) is done, with 20x improvement over standard method for emittance tuning. → ***Now working to integrate into operations.***

→ ***Also now working to incorporate more informative global models /priors rather than learning the model from scratch each time.***

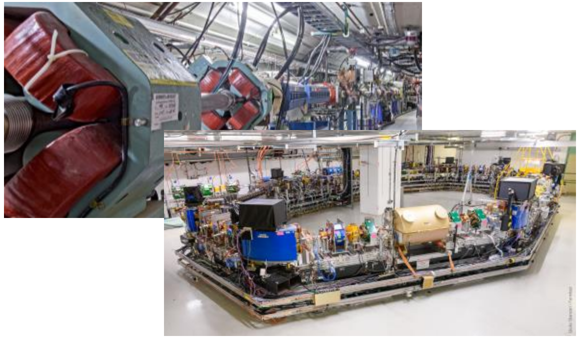
Autonomous Control: Xopt/Badger Contributions

R. Roussel

Argonne NATIONAL LABORATORY

SLAC NATIONAL ACCELERATOR LABORATORY

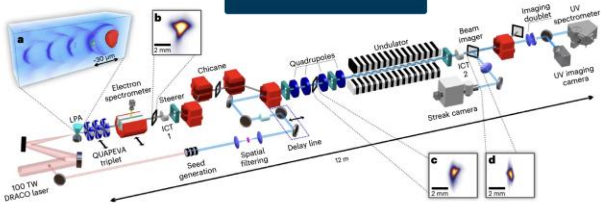
Fermilab



Xopt

Badger GUI interface

BERKELEY LAB



Roussel, R., et al. (IPAC 2023)

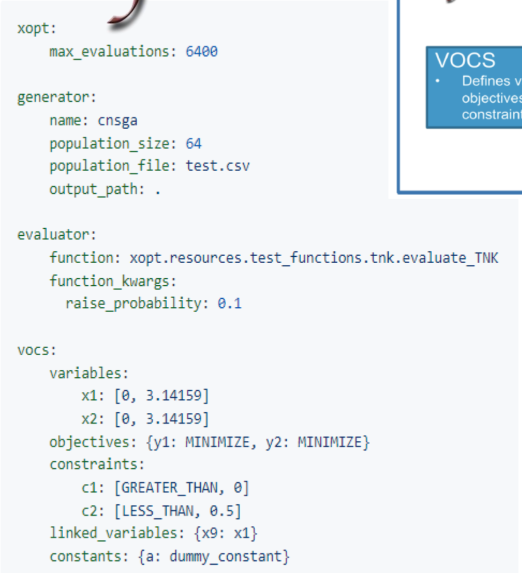
Brookhaven National Laboratory

ESRF The European Synchrotron

DESY.

Deployment: Xopt and Badger

Xopt: houses optimization algorithms



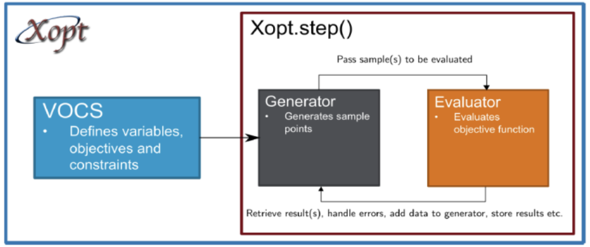
```
xopt:
    max_evaluations: 6400

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 interface



Xopt.step()

Pass sample(s) to be evaluated

VOCS
- Defines variables, objectives and constraints

Generator
- Generates sample points

Evaluator
- Evaluates objective function

Retrieve result(s), handle errors, add data to generator, store results etc.

Many optimization algorithms

- *Genetic algorithms (NSGA-II, etc.)*
- *Nelder-Mead Simplex*
- *Bayesian Optimization*
- *Bayesian Exploration*
- *Trust-region BO*
- *Learned output constrained BO*
- *Interpolating BO*
- *See more BO algorithm details/capabilities here: https://arxiv.org/abs/2312.05667*

Badger GUI interface

User interface, I/O with machine

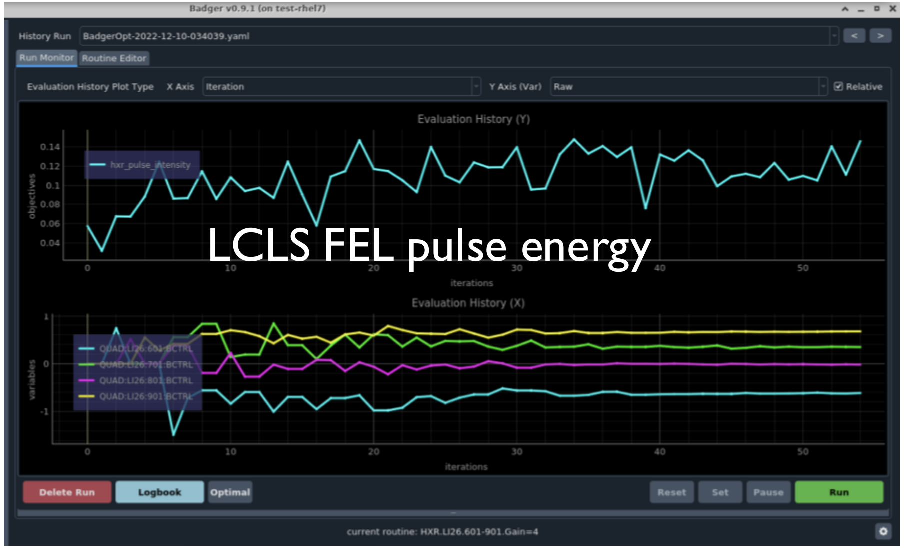
https://christophermayes.github.io/Xopt/

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



Badger v0.9.1 (on test-rhel7)

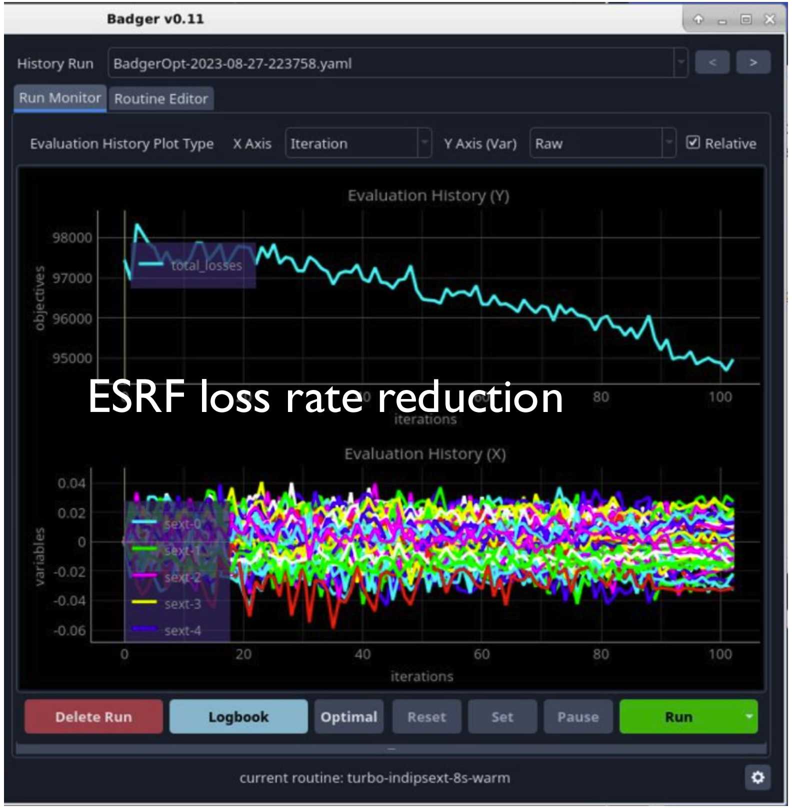
History Run BadgerOpt-2022-12-10-034039.yaml

Evaluation History Plot Type X Axis Iteration Y Axis (Var) Raw ☑ Relative

current routine: HXR.LI26.601-901.Gain=4

LCLS FEL pulse energy

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



Badger v0.11

History Run BadgerOpt-2023-08-27-223758.yaml

Evaluation History Plot Type X Axis Iteration Y Axis (Var) Raw ☑ Relative

current routine: turbo-indipsext-8s-warm

ESRF loss rate reduction

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

Reinforcement Learning

Appealing for moving toward large-scale, comprehensive control of accelerators

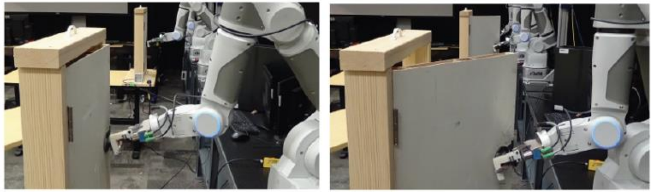
→ Many similarities to robotics applications

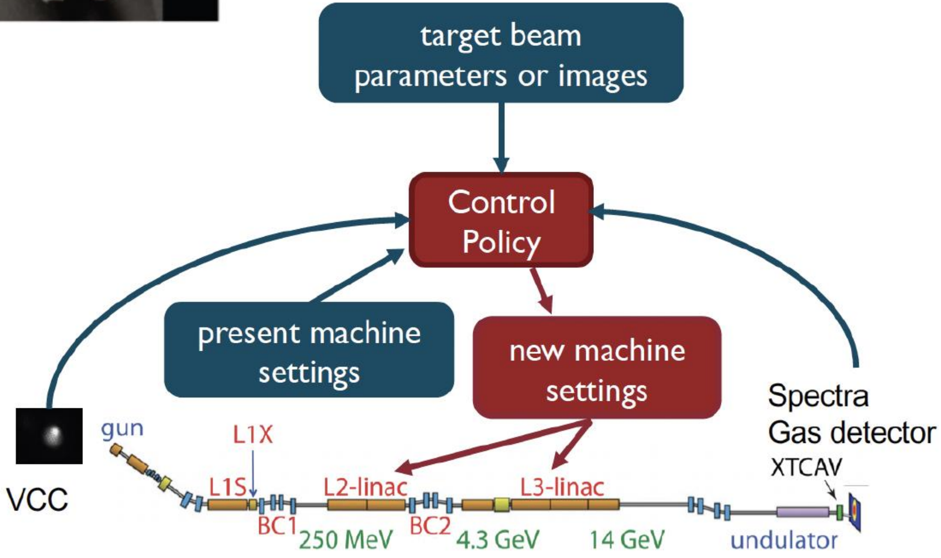
→ Ability to learn from many observations

→ Multi-modal, high-dimensional data

Nagabandi, et al., 2019

Gu, et al., 2016





target beam parameters or images

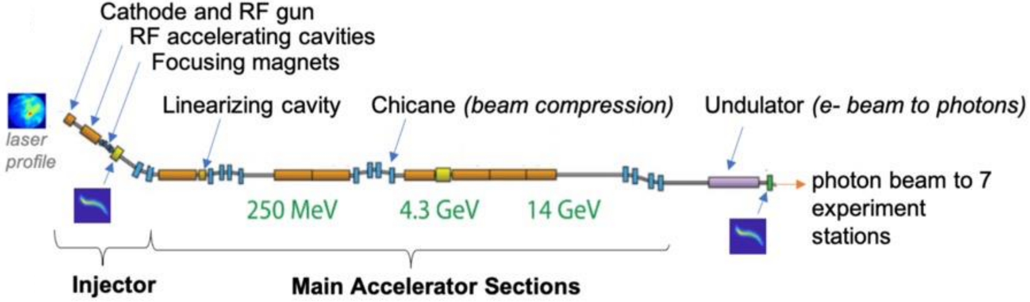
Control Policy

present machine settings

new machine settings

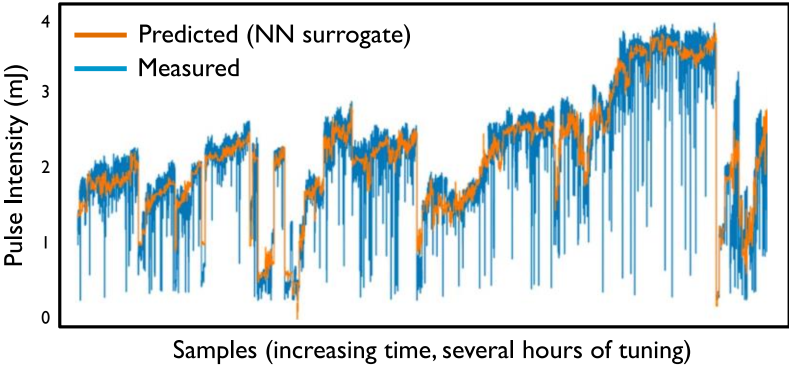
Reinforcement Learning

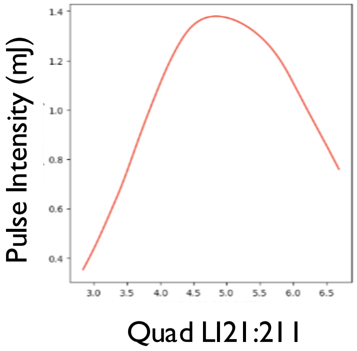
- Appealing for moving toward large-scale, comprehensive control of accelerators
- Free Electron Laser at LCLS is sensitive to focusing, trajectory; perturbing beam/feedbacks too much results in beam losses
- Using data-driven surrogates and differentiable sims to train agents
- Iteratively add more data, targets and variables:
  - Photon pulse intensity
  - Beam phase space images, spectra
  - Focusing magnets, RF cavities, undulator
- Similar accelerator designs → facility-agnostic agents?


Cathode and RF gun
RF accelerating cavities
Focusing magnets
laser profile
Linearizing cavity
Chicane (beam compression)
Undulator (e- beam to photons)
250 MeV
4.3 GeV
14 GeV
photon beam to 7 experiment stations
Injector
Main Accelerator Sections

~28 focusing magnets for FEL pulse intensity

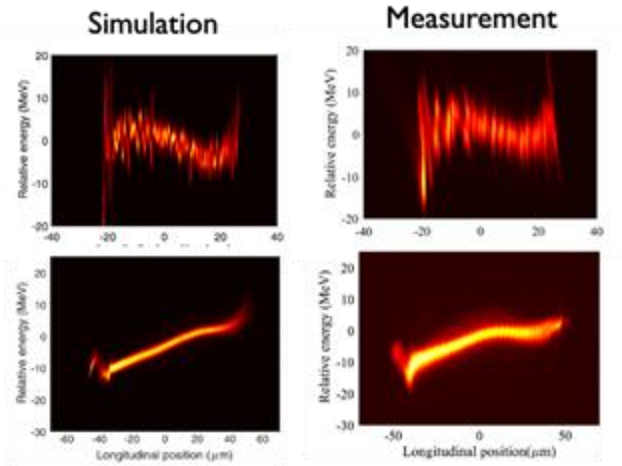
(many more variables to include: steering, rf cavities, undulator, drive laser)





Fast-Executing, Accurate System Models

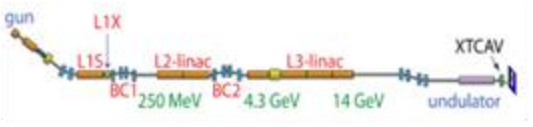
Accelerator simulations that include nonlinear and collective effects are powerful tools, but they can be computationally expensive



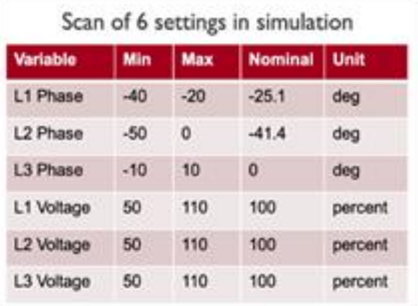
J. Qiang, et al., PRSTAB30, 054402, 2017

10 hours on thousands of cores at NERSC!

ML models are able to provide fast approximations to simulations (“surrogate models”)

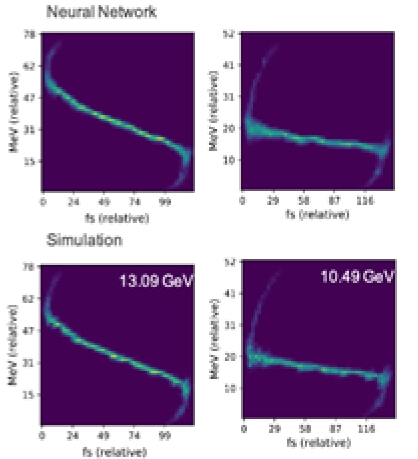


Linac sim in Bmad with collective beam effects



Scan of 6 settings in simulation

Variable	Min	Max	Nominal	Unit
L1 Phase	-40	-20	-25.1	deg
L2 Phase	-50	0	-41.4	deg
L3 Phase	-10	10	0	deg
L1 Voltage	50	110	100	percent
L2 Voltage	50	110	100	percent
L3 Voltage	50	110	100	percent



< ms execution speed

10^6 times speedup

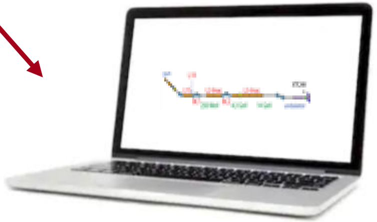
Edelen et al., NeurIPS 2019

Long history now of using ML modeling to enable accurate predictions of accelerator system responses with unprecedented speeds

Fast-Executing, Accurate System Models

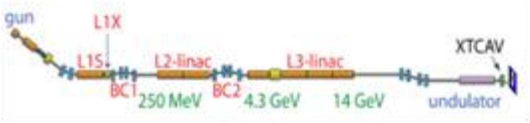
Bringing simulation tools from HPC systems to online/local compute

Online prediction
Model-based control

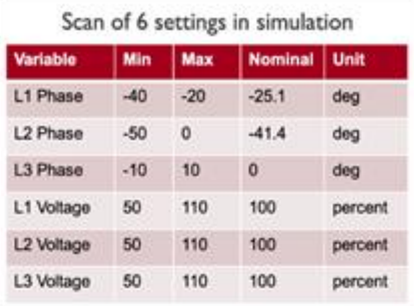


Control prototyping
Experiment planning

ML models are able to provide fast approximations to simulations ("surrogate models")

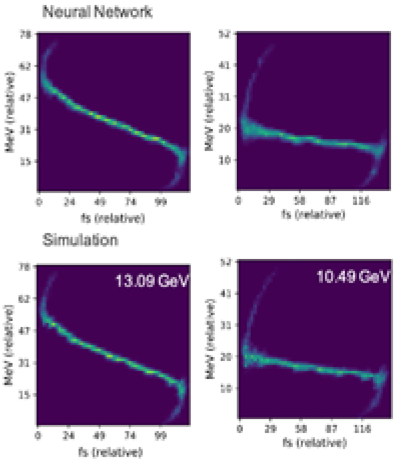


Linac sim in Bmad with collective beam effects



Scan of 6 settings in simulation

Variable	Min	Max	Nominal	Unit
L1 Phase	-40	-20	-25.1	deg
L2 Phase	-50	0	-41.4	deg
L3 Phase	-10	10	0	deg
L1 Voltage	50	110	100	percent
L2 Voltage	50	110	100	percent
L3 Voltage	50	110	100	percent



< ms execution speed

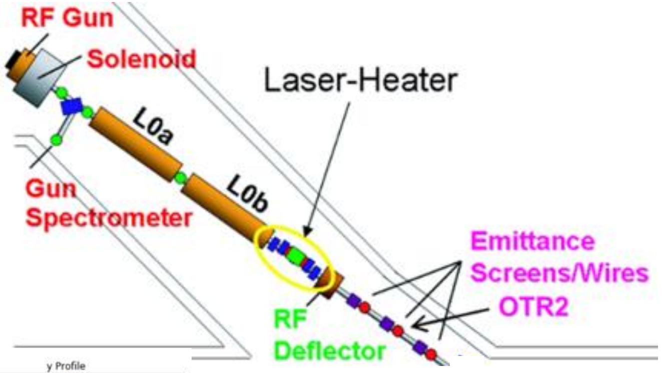
10^6 times speedup

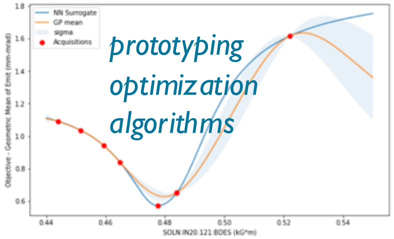
Edelen et al., NeurIPS 2019

Long history now of using ML modeling to enable accurate predictions of accelerator system responses with unprecedented speeds

In Regular Use: Injector Surrogate Model at LCLS

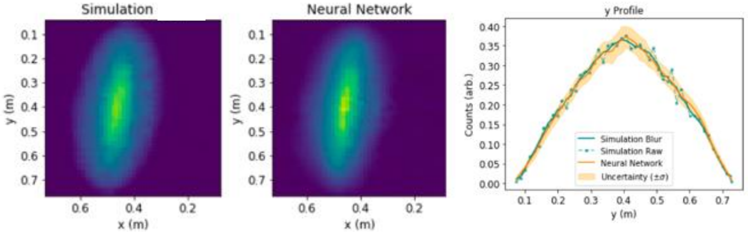
- ML models trained on detailed physics simulations with nonlinear collective effects
- Accurate over a wide range of settings → calibrate to match machine measurements
- Provide initial parameters for downstream model



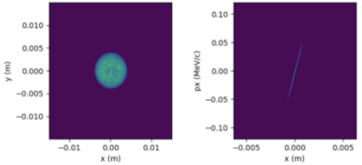


prototyping optimization algorithms

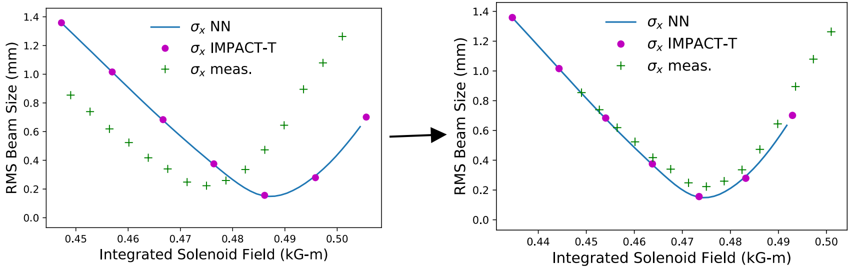
ML model matches simulation under interpolation



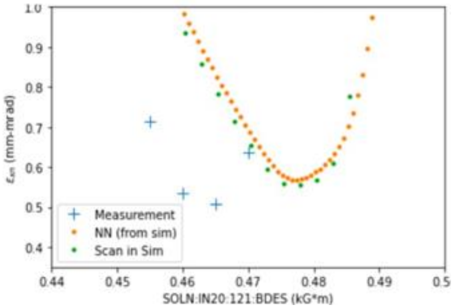
Simulation and ML model trained on it are qualitatively similar to measurements under interpolation (setting combinations reasonable distance from training set)



interactive model widget and visualization tools



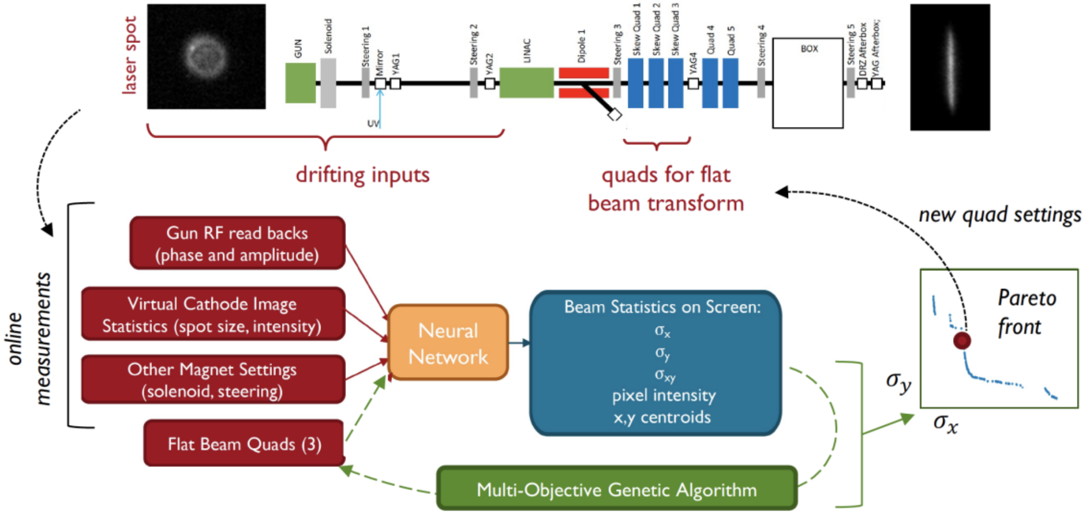
Automatic adaptation of models and identification of sources of deviation between simulations and as-built machine



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

Warm Starts from Fast Online ML Models

E. Cropp et al., in preparation



drifting inputs

quads for flat beam transform

new quad settings

Gun RF read backs (phase and amplitude)

Virtual Cathode Image Statistics (spot size, intensity)

Other Magnet Settings (solenoid, steering)

Flat Beam Quads (3)

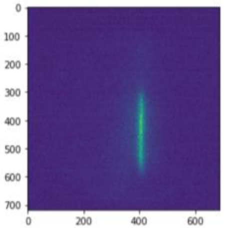
Neural Network

Beam Statistics on Screen: σ_x σ_y σ_{xy} pixel intensity x,y centroids

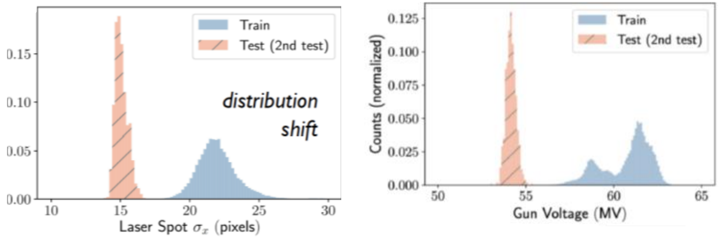
Multi-Objective Genetic Algorithm

Pareto front

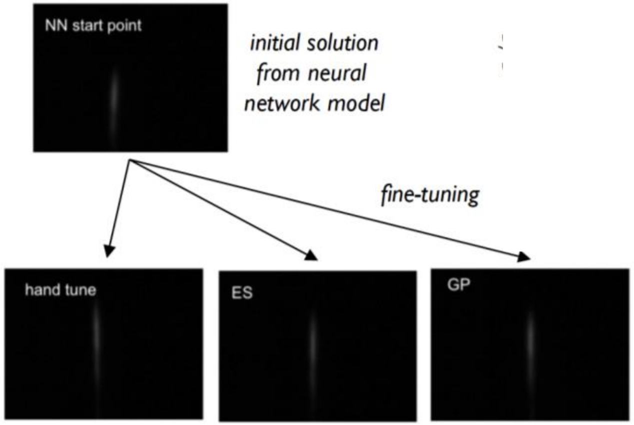
- Round-to-flat beam transforms are challenging to optimize → 2019 study explored ability of a learned model to help
- Trained neural network model to predict fits to beam image, based on archived data
- Tested online multi-objective optimization over model (3 quad settings) given present readings of 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



distribution shift



initial solution from neural network model

fine-tuning

Hand-tuning in seconds vs. tens of minutes

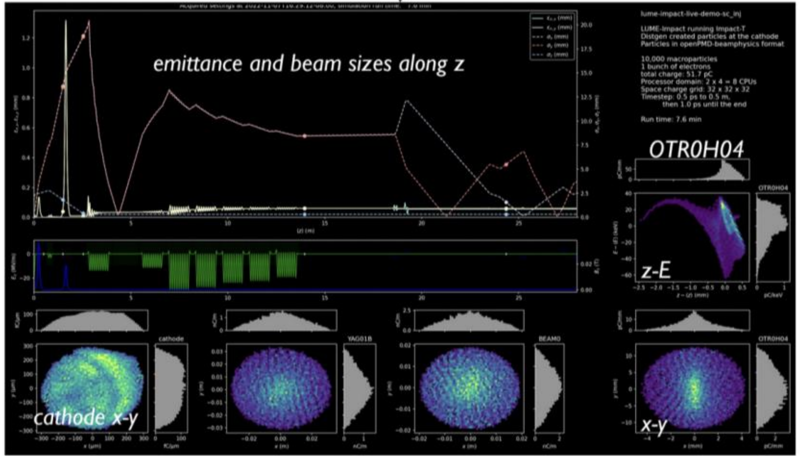
Boost in convergence speed for other algorithms

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



lume-impact-live-demo-sc_inj

LUME-Impact running Impact-T
Distgen created particles at the cathode
Particles in openPMD-beamphysics format

10,000 macroparticles
1 bunch of electrons
total charge: 51.7 pC
Processor domain: 2 x 4 = 8 CPUs
Space charge grid: 32 x 32 x 32
Timestep: 0.5 ps to 0.5 m,
then 1.0 ps until the end

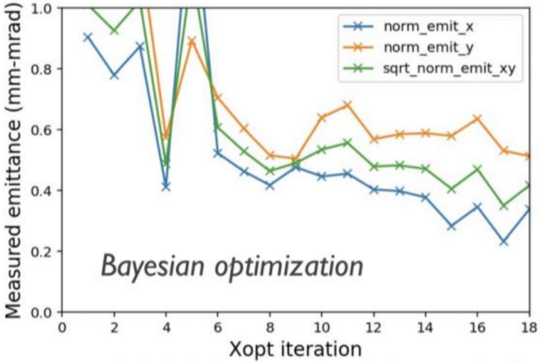
Run time: 7.6 min

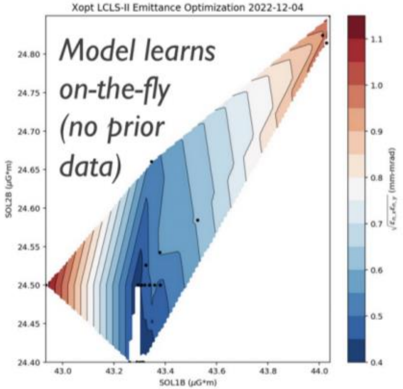
LCLS-II live sim: run on HPC and display in control room

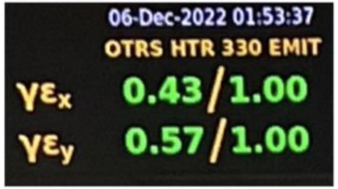
Updates every 3-8 mins, space charge included, uses LUME-IMPACT

Adjust settings / ranges with insight from predictions

Hand over to ML-based optimization for fine tuning







06-Dec-2022 01:53:37

OTRS HTR 330 EMIT

$\gamma\varepsilon_x$ 0.43/1.00

$\gamma\varepsilon_y$ 0.57/1.00

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

Finding Sources of Error Between Simulations and Measurements

Many non-idealities not included in physics simulations:

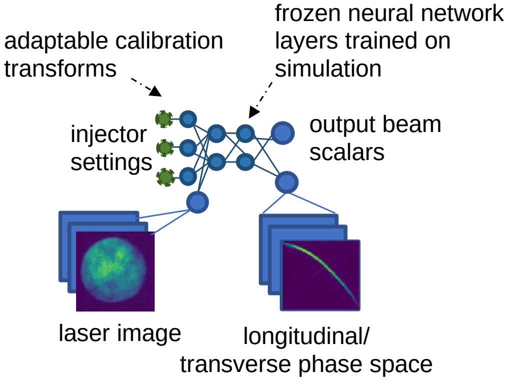
static error sources (e.g. magnetic field nonlinearities, physical offsets)

time-varying changes (e.g. temperature-induced phase calibrations)

Want to identify these to get better understanding of machine performance

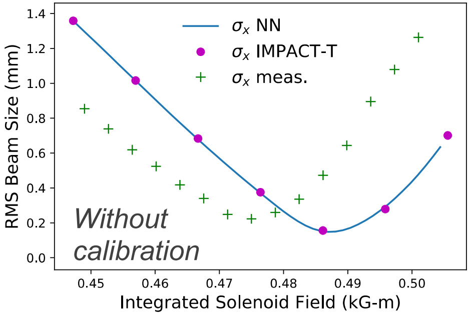
à ***ML model allows fast / automatic exploration of error sources in high dimension***

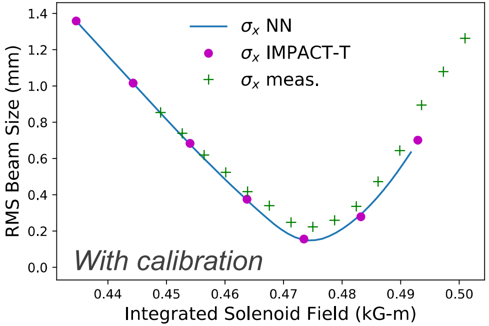
Example: calibration offset in injector solenoid strength found automatically with neural network model (trained first in simulation, then calibrated to machine)


adaptable calibration transforms
frozen neural network layers trained on simulation
injector settings
output beam scalars
laser image
longitudinal/ transverse phase space

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

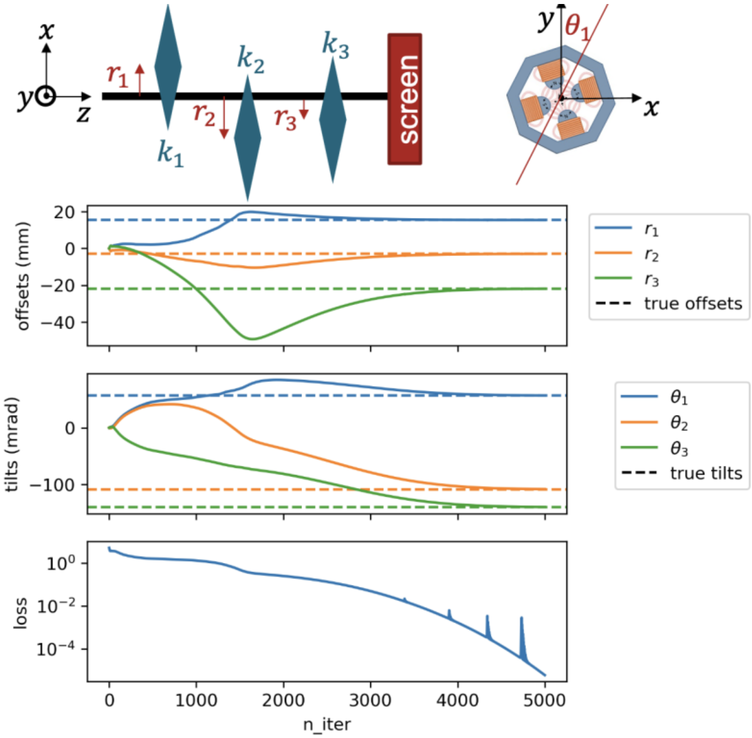


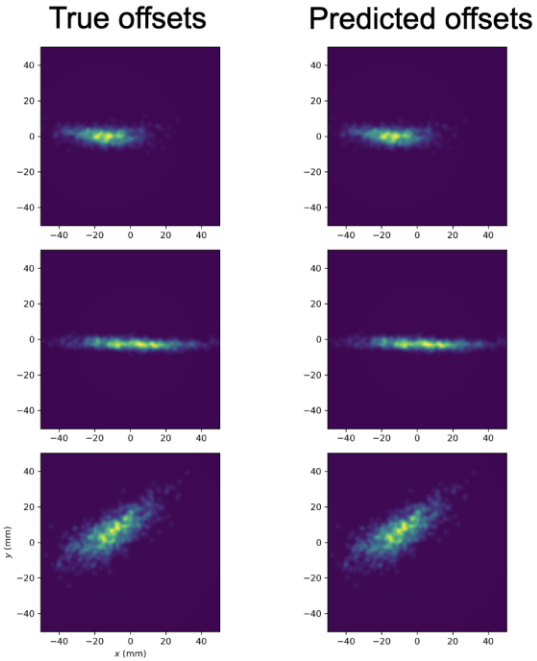


Speed and differentiability of ML models enables rapid identification of error sources between idealized physics simulations and real machine

Finding Sources of Error Between Simulations and Measurements

Same approach can be used with differentiable physics simulations





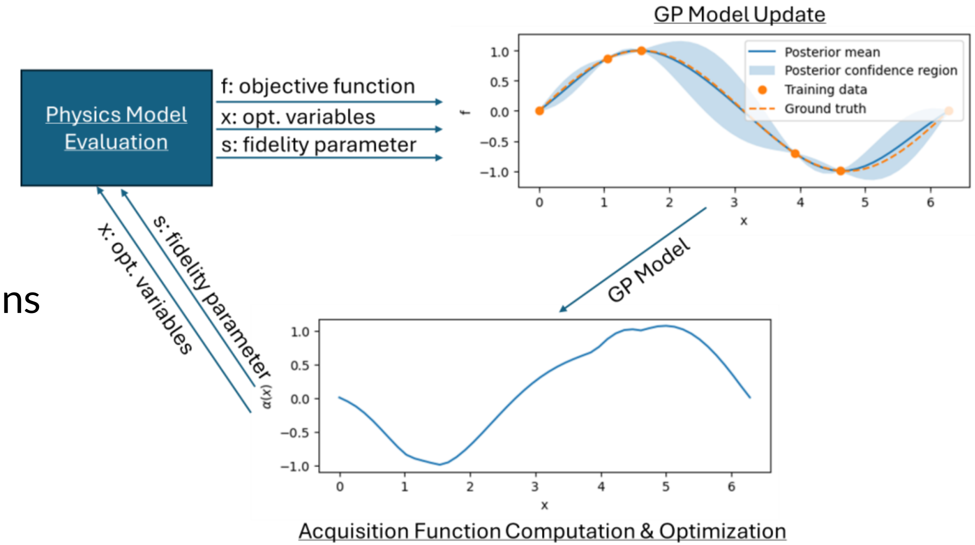
J.P. Gonzalez-Aguilera

https://accelconf.web.cern.ch/ipac2023/pdf/WEPA065.pdf

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

Multifidelity Optimization

- Information theoretic approach to simulations
- Learn correlations between different model fidelities
- Use multi-fidelity Bayesian optimization to select model fidelity and next optimization variables



Physics Model Evaluation

f: objective function

x: opt. variables

s: fidelity parameter

GP Model Update

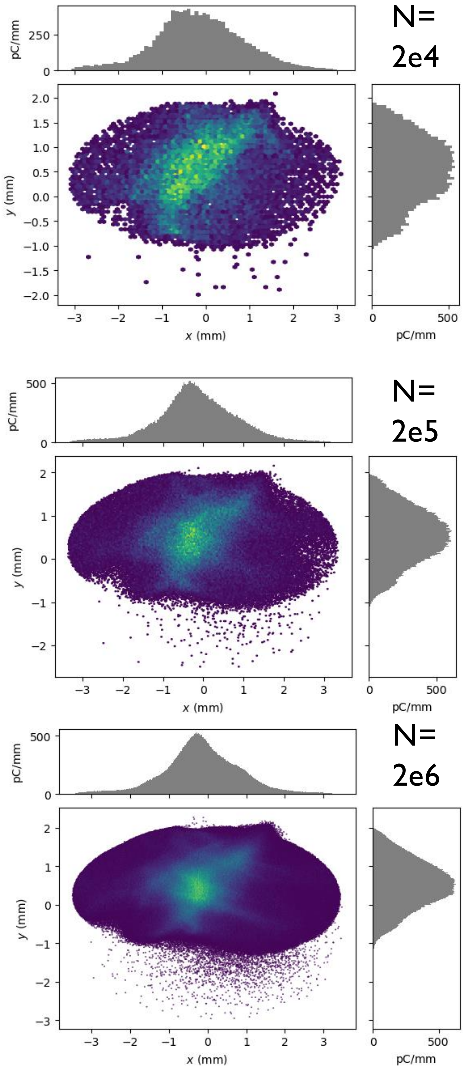
GP Model

s: fidelity parameter

x: opt. variables

Acquisition Function Computation & Optimization

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

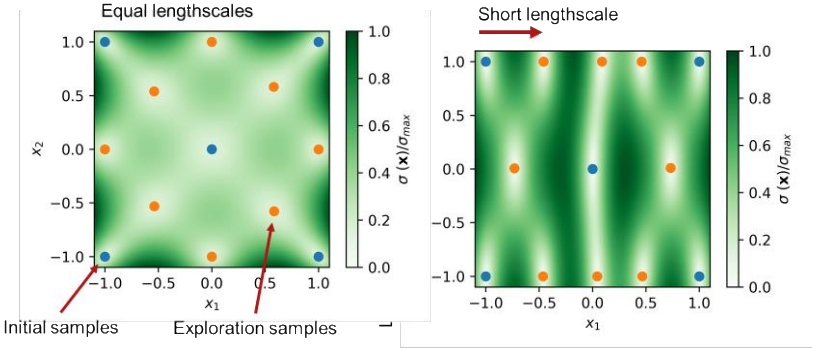
Efficient Characterization with Bayesian Exploration

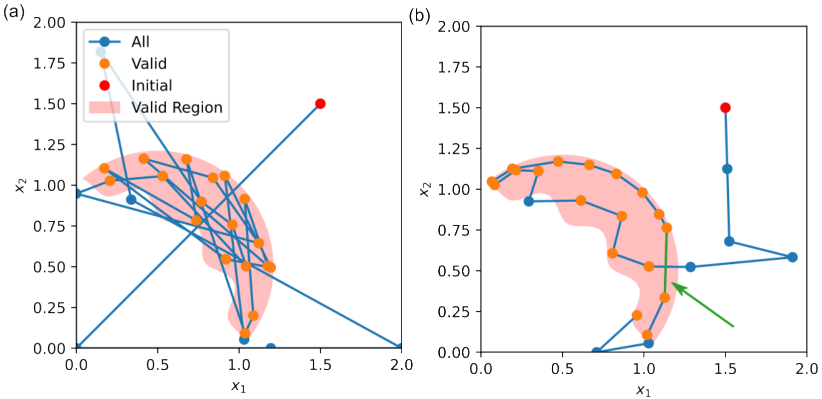
R. Roussel et. al.
Nat. Comm. **2021**

$$!(") = \underset{!\#\$}{\overset{"}{\$(")\%}}\ \&('_!(") \geq h_!)\,\Psi(","_\%)$$

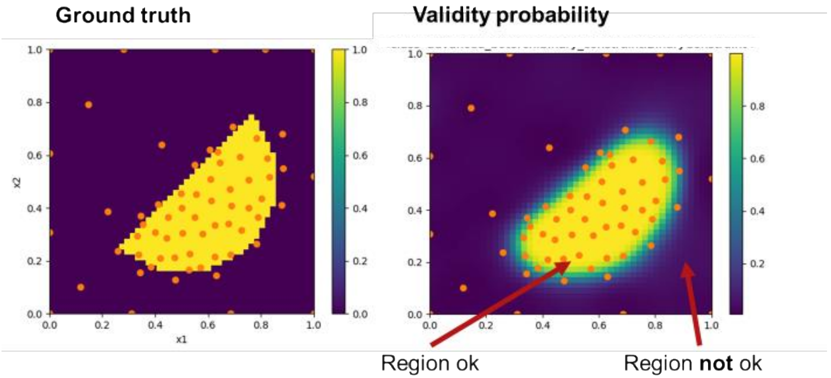
proximal biasing

adaptive sampling



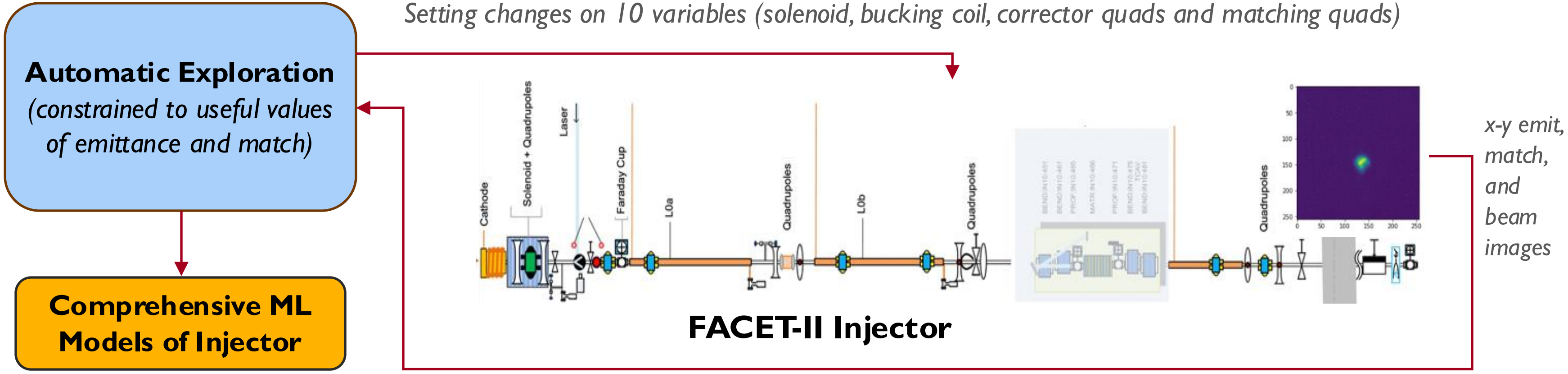


learning constraints



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

Efficient Characterization with Bayesian Exploration



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

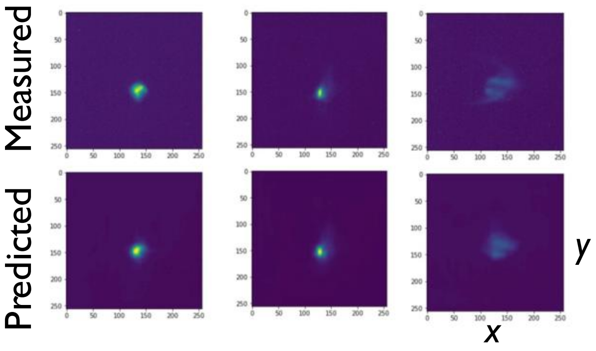
x-y emit, match, and beam images

- 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**

- 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

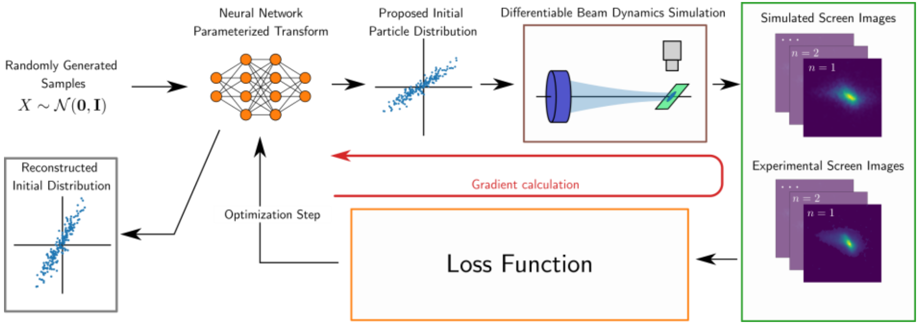
Use of Bayesian exploration to generate training data was sample-efficient, reduced burden of data cleaning, and resulted in a well-balanced distribution for the training data set over the input space. ML models were immediately useful for optimization.

Phase Space Reconstruction with Differentiable Tracking Simulations

https://arxiv.org/abs/2404.10853

https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.130.145001

Differentiable pipeline for reconstructing 6D phase space distribution using neural network parameterization


Randomly Generated Samples $X \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Neural Network Parameterized Transform

Proposed Initial Particle Distribution

Differentiable Beam Dynamics Simulation

Simulated Screen Images

Experimental Screen Images

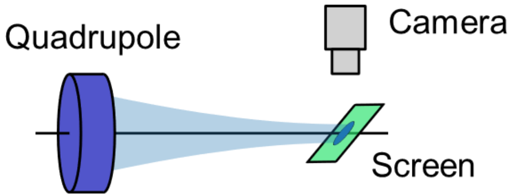
Gradient calculation

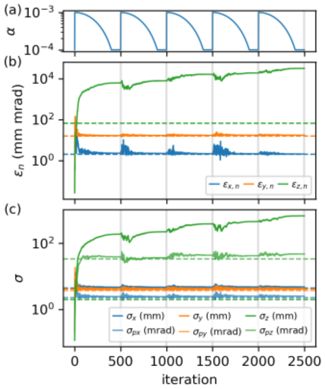
Optimization Step

Loss Function

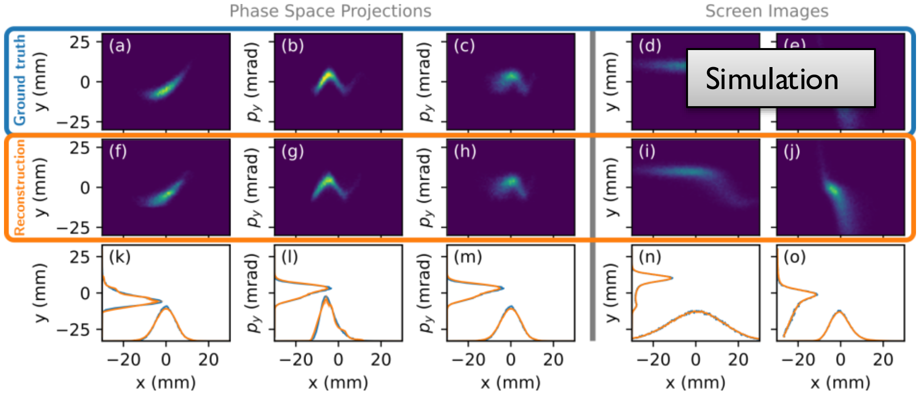
Reconstructed Initial Distribution

Reconstruct 4D phase space distribution + approx. energy spread from simple beamline diagnostic and 10 measurements

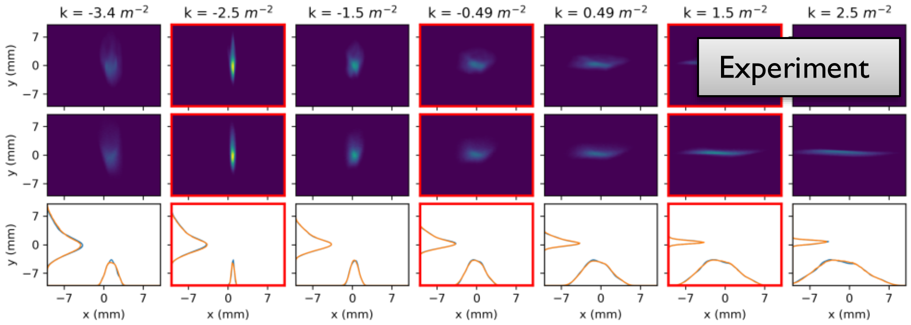




Confidence estimates



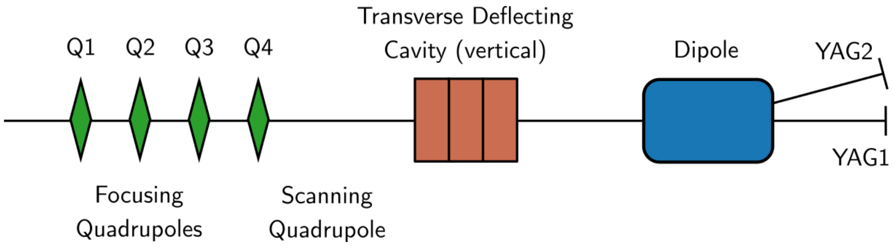
Simulation



Experiment

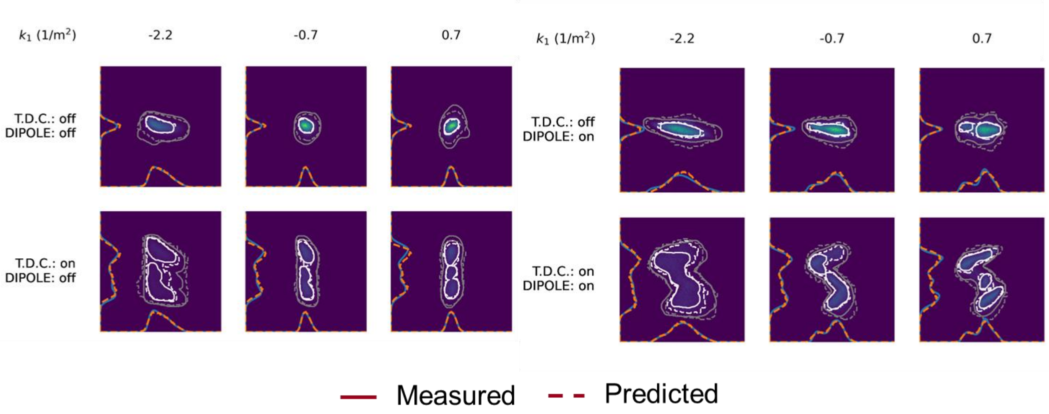
ML combined with differentiable simulations opens up a new paradigm for constructing detailed phase space diagnostics in a way that is computationally-efficient and sample-efficient

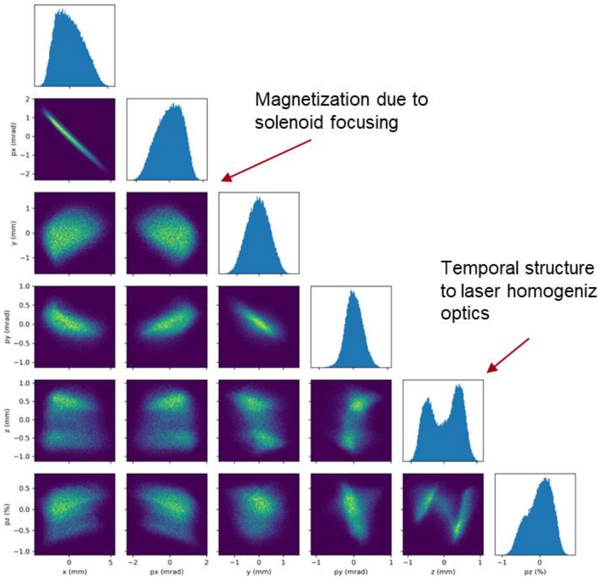
Phase Space Reconstruction with Differentiable Tracking Simulations



Have now extended to 6D phase space

- *20 measurements / ~15 minutes analysis time*
- *~75x faster than conventional approach*





Magnetization due to solenoid focusing

Temporal structure likely to laser homogenization optics

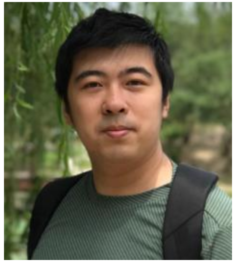
Roussel, R, et al. https://arxiv.org/abs/2404.10853

ML combined with differentiable simulations opens up a new paradigm for constructing detailed phase space diagnostics in a way that is computationally-efficient and sample-efficient

Summary

- Many activities in digital twin infrastructure to learn from / build on
- Continual deployment of simple models for feed-forward corrections has had success, including online updating (e.g. ALS)
- Badger/Xopt being used widely in community
- Suites of algorithms for faster characterization / model calibration are reaching maturity and being expanded upon
    - Bayesian exploration, differentiable simulations, multi-fidelity calibration

Thanks for your attention!
Any questions?



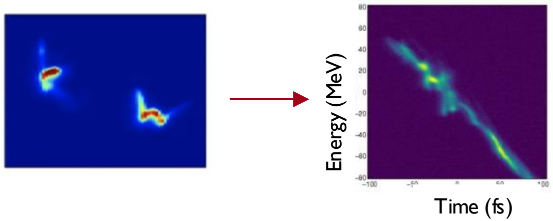
Thanks to the core team at SLAC working on various digital twin and AIML technologies and infrastructure, and many other collaborators!

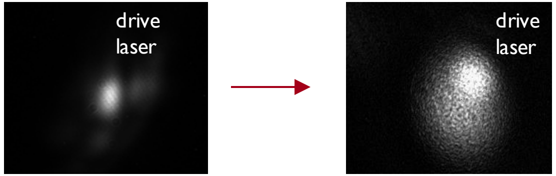
Backups

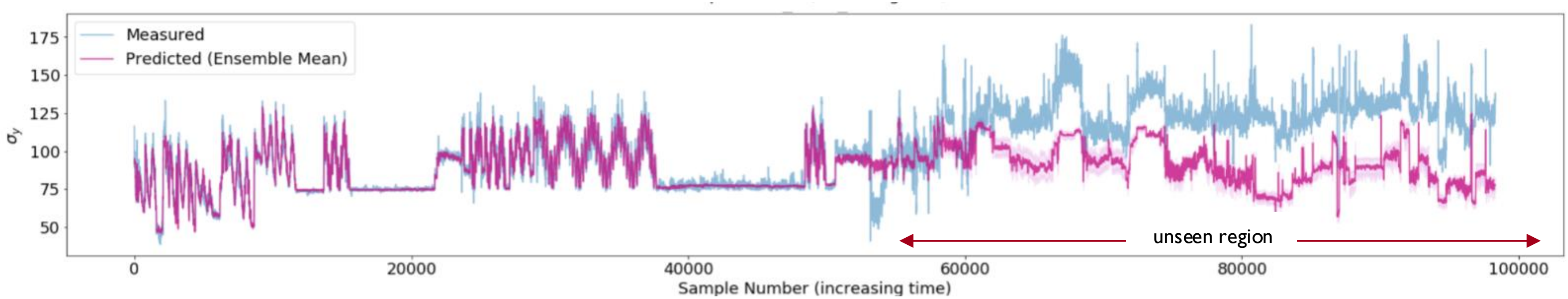
Distribution Shift is a Major Challenge in Particle Accelerators

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
Need fast ways of obtaining characterization data from accelerator

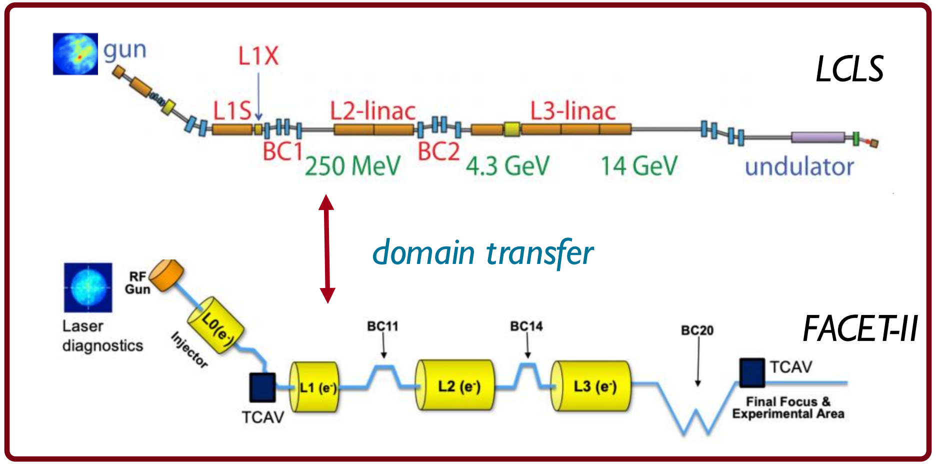
Summary/Conclusions

- Particle accelerators stand to benefit substantially from the development and deployment of AIML for modeling and control
  - Faster optimization, new capabilities in beam customization, human-AI interaction
  - High impact for science that is supported by particle accelerators (and translations to industry/medicine)

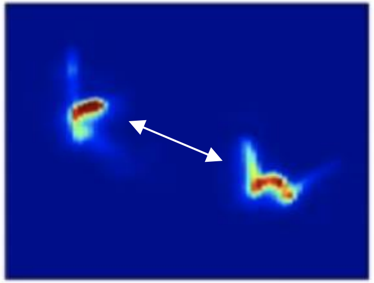
- Now scaling up small-scale demos to tackle larger problems, making algorithms more robust, developing deployment infrastructure, and bringing into routine operation

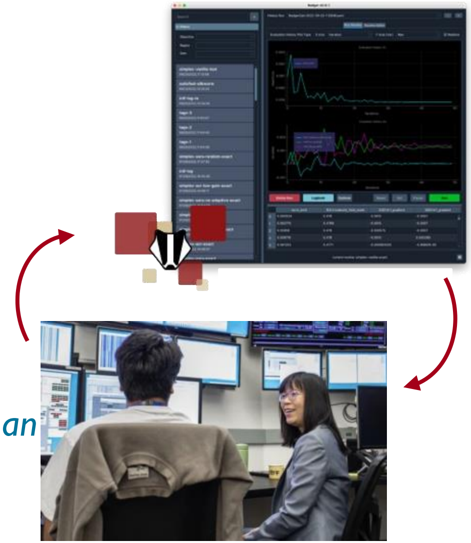
→ **Many interesting problems to tackle, and we welcome collaborations!**

→ **Accelerators are also interesting platforms for AIML research!**



fast dynamic beam customization



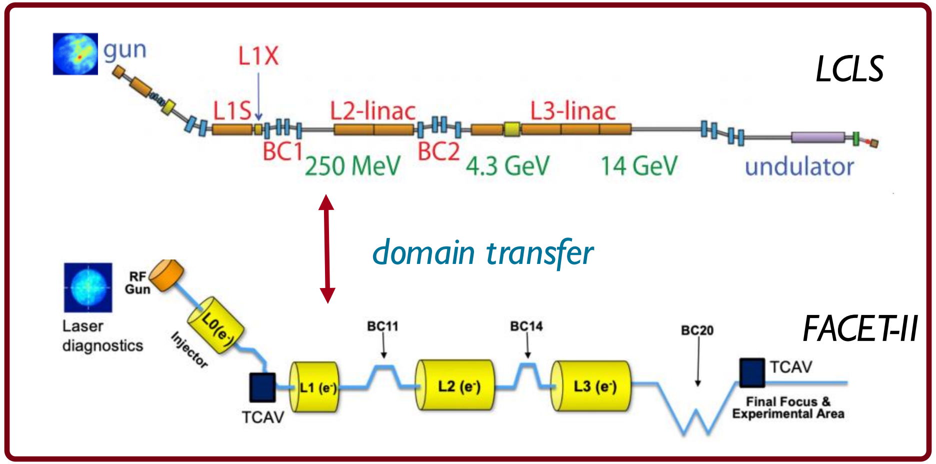


AIML + human feedback

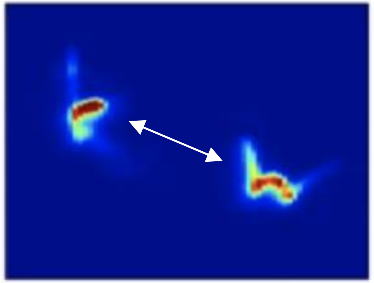
Opportunities for AIML Accelerator Research in Accelerators

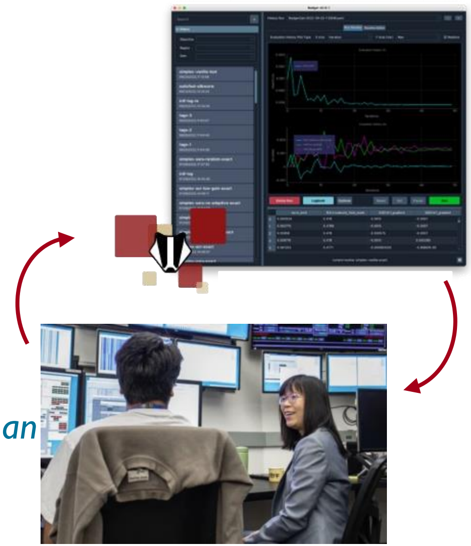
(mix of needs from science side + compelling areas in AIML)

- Pushing to higher-dimensional algorithms (more comprehensive, precise tuning); incorporation of multiple, multi-modal output beam targets
- Sample-efficient adaptation across setups and over time needed *(different charges, beam phase space, multi-bunch)*
- Enabling fundamentally new capabilities in beam physics / photon science
  - *FACET-II "extreme beams" → highly sensitive*
  - *Precise dynamic control over beam*
- Comprehensive online system modeling + ML-based optimization
  - *Physics sims + ML surrogates being deployed on local HPC connected to control system (digital twins)*
- AI and human feedback → *human-AI interaction in the control room is a current area of study*
- Transfer learning between accelerators *→ Similar layouts, component design, beam diagnostics, user needs (e.g. scan two bunches)*



fast dynamic beam customization



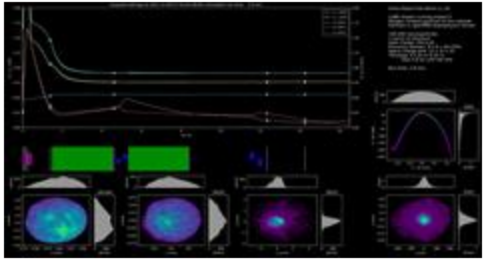


AIML + human feedback

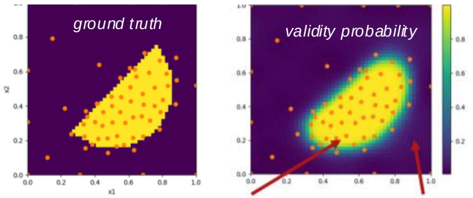
Broad Research Program at SLAC in AI/ML for Accelerators

(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



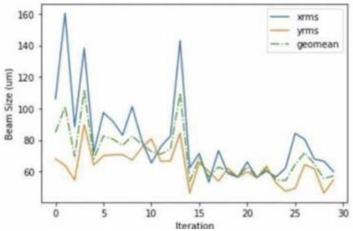
Roussel et. al. *Nat. Comm.* 2021

Region ok

Region **not** ok

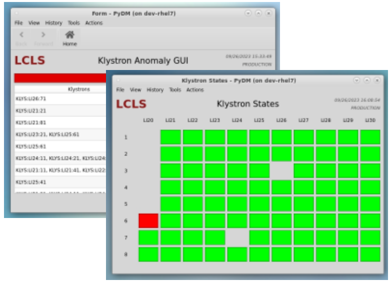
Output constraints learned on-the-fly

Adhere to constraints and balance multiple targets



Challenging problems: e.g. sextupole tuning

Anomaly detection



Form - PyDM (on dev-rhel7)

File View History Tools Actions
Back Forward Home

LCLS Klystron Anomaly GUI

09/26/2023 15:33:49
PRODUCTION

Klystrons

KLYS:LI26:71
KLYS:LI21:21
KLYS:LI21:81
KLYS:LI23:21, KLYS:LI25:61
KLYS:LI25:61
KLYS:LI24:11, KLYS:LI24:21, KLYS:LI24
KLYS:LI21:11, KLYS:LI21:41, KLYS:LI22
KLYS:LI25:41

Klystron States - PyDM (on dev-rhel7)

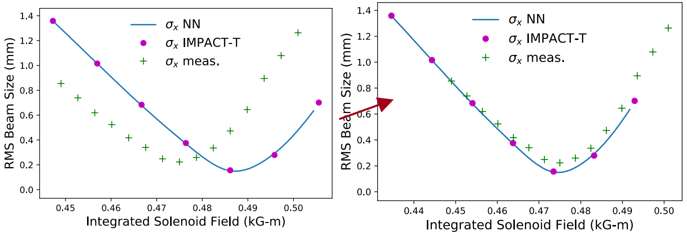
File View History Tools Actions

LCLS Klystron States

09/26/2023 16:08:54
PRODUCTION

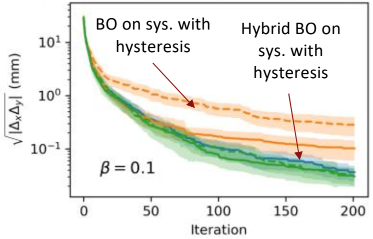
	LI20	LI21	LI22	LI23	LI24	LI25	LI26	LI27	LI28	LI29	LI30
1		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
2		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
3		🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩
4		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
5		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
6	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
7	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩	🟩	🟩
8	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

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



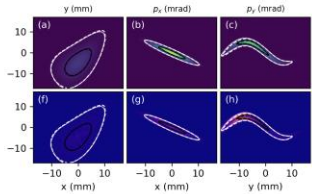
Combining physics and ML for better performance

Hysteresis-aware optimization



Roussel et. al. PRL. 2022

Differentiable simulations + ML for 6D phase space reconstruction

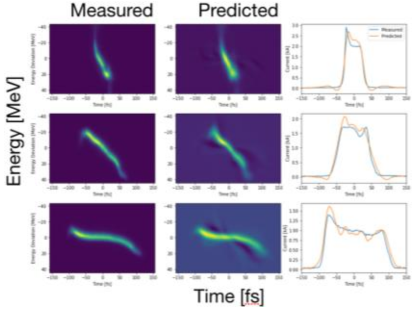


Roussel et. al. PRL. 2023

ML-enhanced diagnostics

Rapid analysis/virtual diagnostics

Shot-to-shot predictions at beam rate



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

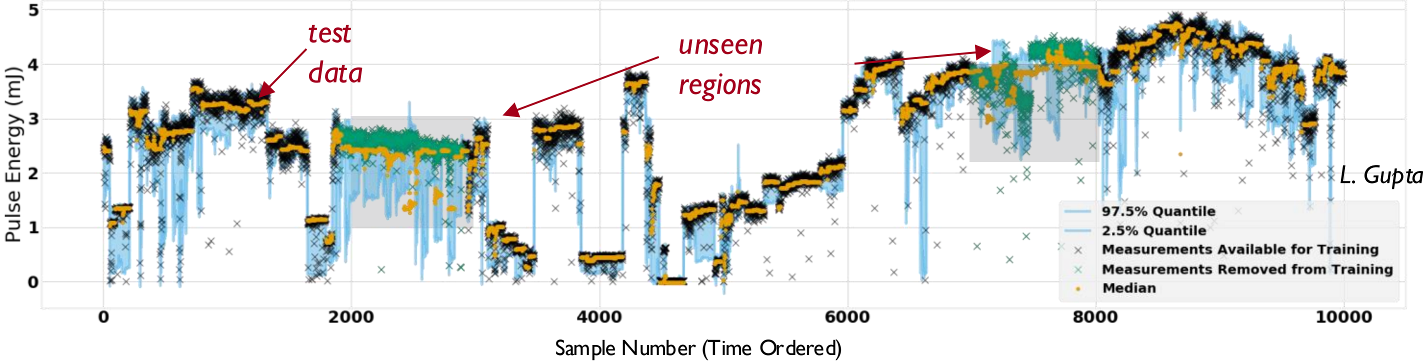
Many solutions put into reusable open-source software (e.g. Xopt/Badger) demoed at many facilities

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

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

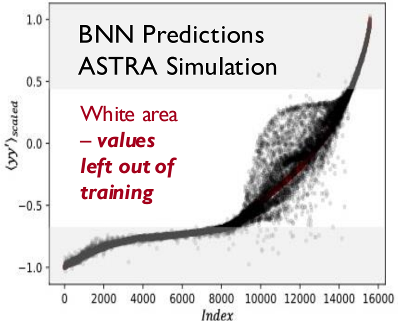


test data

unseen regions

L. Gupta

Neural network with quantile regression predicting FEL pulse energy at LCLS

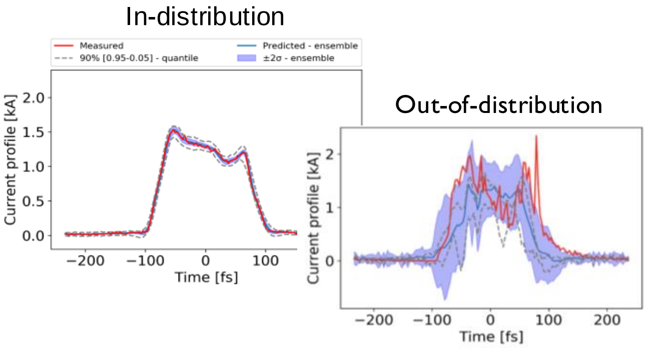


BNN Predictions
ASTRA Simulation

White area – ***values left out of training***

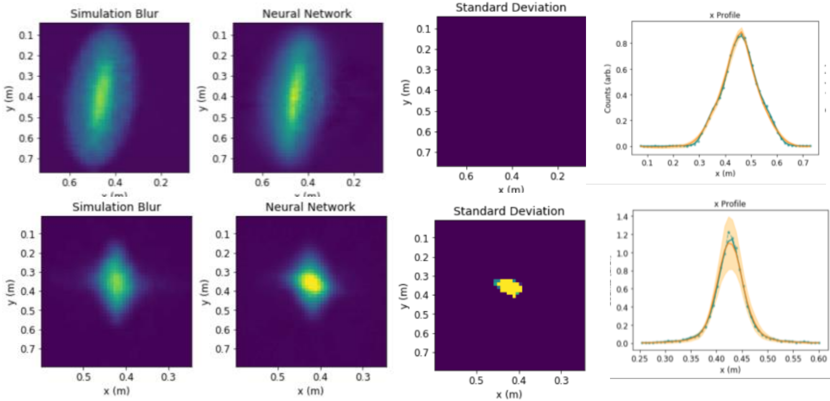
Scalar parameters for the LCLS-II injector (Bayesian neural network)

A. Mishra et. al., PRAB, 2021



longitudinal phase space (quantile regression + ensemble)

O. Convery, et al., PRAB, 2021



LCLS injector transverse phase space (ensemble)

Motivation / Need for AIML for Particle Accelerators

Two major categories of need for AI/ML in accelerators:

- **New fundamental beam dynamics capabilities:** unprecedented beam parameters, finely-detailed customization and characterization for experiments
- **Facility operations:** efficiency of tuning and quality of beam delivery → increase science output, reduce time-to-discovery

Accelerator and Beam Physics Grand Challenges

Intensity – *"How do we increase beam intensity by orders of magnitude?"*

Quality – *"How do we increase the beam phase space density by orders of magnitude?"*

Control – *"How do we measure and control the beam distribution down to the individual particle level?"*

Prediction – *"How do we develop predictive 'virtual particle accelerators'"*

AI/ML features prominently in the ABP Roadmap to address these challenges: https://science.osti.gov/hep/-/media/hep/pdf/2022/ABP_Roadmap_2023_final.pdf

Operational/Facility Challenges

Increasingly complex facilities, challenging setups, tighter tolerances on beam for experiments
(e.g. exotic FEL setups, PWFA)

Need for on-demand dynamic control during experiments
(e.g. scanning beams in XPCS, compensation for drift)

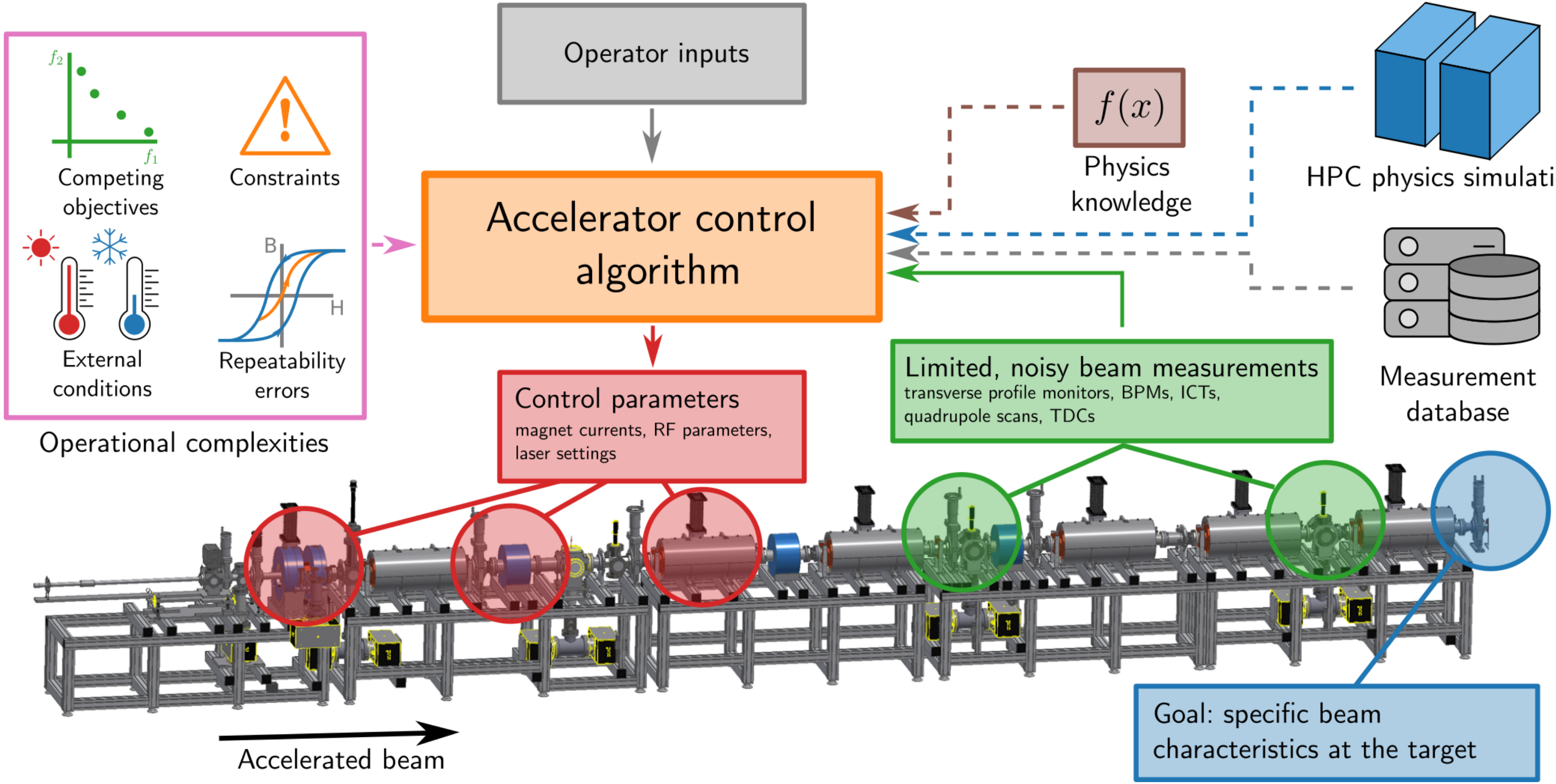
Currently rely on extensive hand-tuning
(e.g. 400 hours per year at LCLS → 10 experiments, $12M)

Limited diagnostics, high dimensional parameter spaces, few accurate models → *challenge to understand machine, do data analysis, do experiment planning*

Mix of operational needs for tuning/control: *stable delivery, fast switching between setups, commissioning new capabilities*

New approaches for beam prediction, measurement, and control are needed to meet the demands for current and future accelerator applications and scientific user facilities.

Machine Learning Based Accelerator Control



Operator inputs

Accelerator control algorithm

Physics knowledge

HPC physics simulation

Measurement database

Operational complexities

Competing objectives

Constraints

External conditions

Repeatability errors

Control parameters

magnet currents, RF parameters, laser settings

Limited, noisy beam measurements

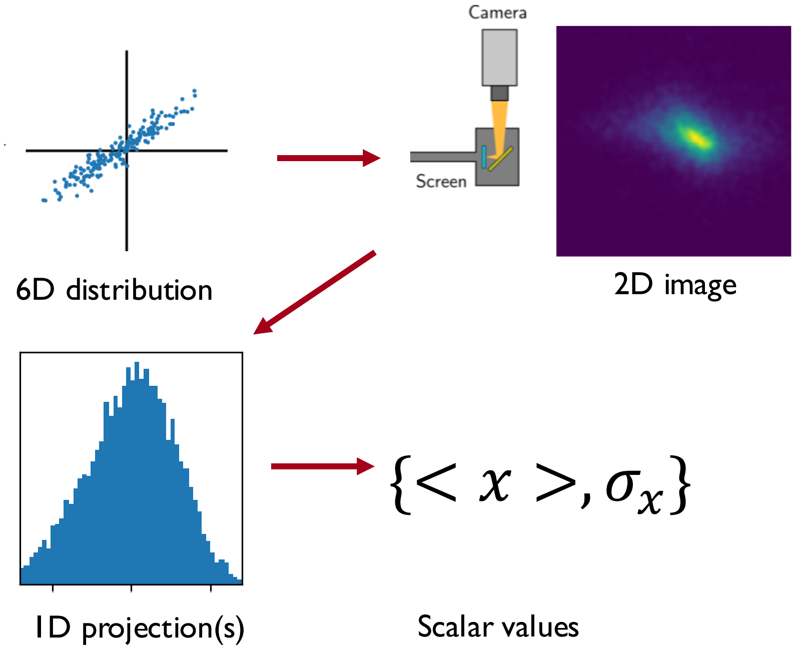
transverse profile monitors, BPMs, ICTs, quadrupole scans, TDCs

Goal: specific beam characteristics at the target

Accelerated beam

Why Is This So Difficult?

Information compression



6D distribution

2D image

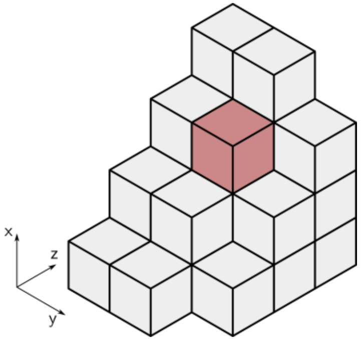
1D projection(s)

$\{< x >, \sigma_x\}$

Scalar values

Often required by analysis constraints (analytical tractability, optimization simplicity, etc.)

Costs of detailed beam representations



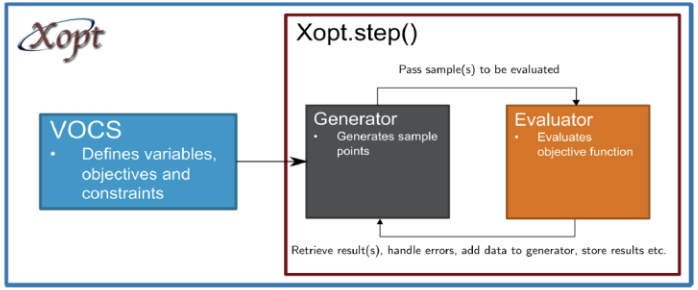
Histogramming scales poorly with number of dimensions, $N_{bins} \propto n^D$

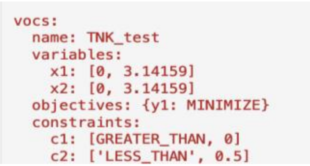
Reasonable resolution, $n = 100$
For a 6D distribution, $N_{bins} = 10^{12}$!

Modular, Open-Source Software Development

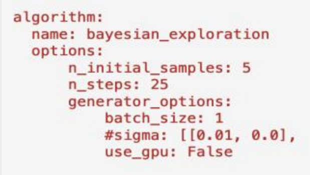
Community development of **re-usable, reliable, flexible software tools** for AI/ML workflows has been essential to maximize return on investment and ensure transferability between systems

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


Xopt
Xopt.step()
Pass sample(s) to be evaluated
VOCS
• Defines variables, objectives and constraints
Generator
• Generates sample points
Evaluator
• Evaluates objective function
Retrieve result(s), handle errors, add data to generator, store results etc.



```
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
```

Different software for different tasks:

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

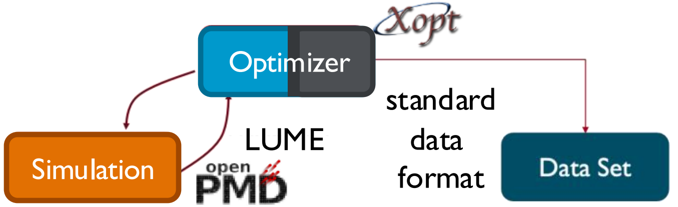
Visual control room interface *(e.g. Badger)*

Simulation drivers *(e.g. LUME)*

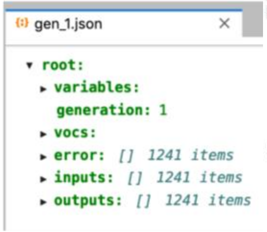
Standards model descriptions, data formats, and software interfaces *(e.g. openPMD)*

Online model deployment *(LUME-services)*

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

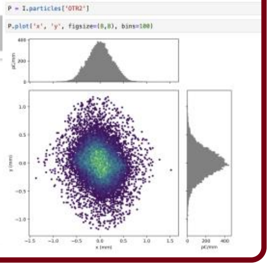


Impact
ASTRA
GPT
Bmad
Genesis
SRW



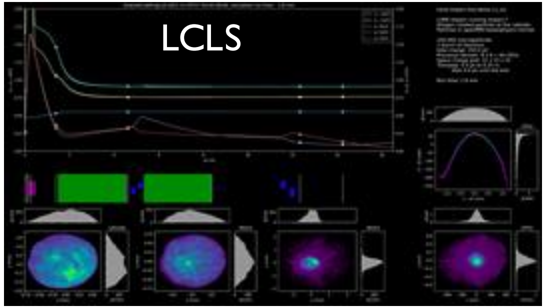
gen_1.json

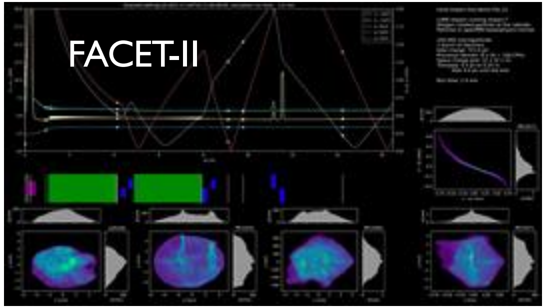
```
▾ root:
  ▸ variables:
    generation: 1
  ▸ vocs:
  ▸ error: [] 1241 items
  ▸ inputs: [] 1241 items
  ▸ outputs: [] 1241 items
```



```
P = I.particles['OTR2']
```

```
P.plot('x', 'y', figsize=(8,8), bins=100)
```





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

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