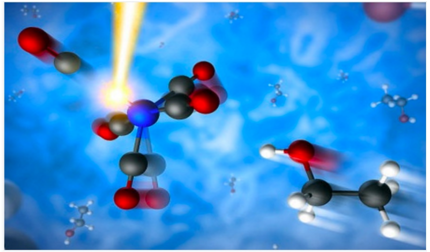
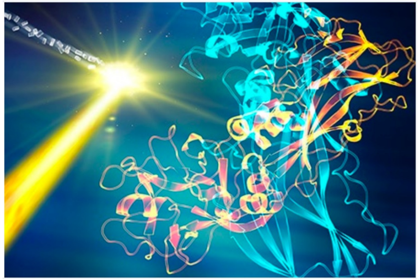
RL4AA '24
Feb. 5, 2024

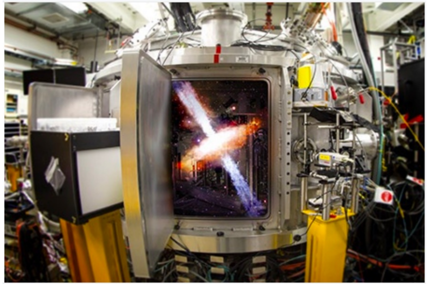
ML-based Optimization/Control for SLAC's Accelerators

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

SLAC team: Daniel Ratner, Ryan Roussel, Zhe Zhang, Tobias Boltz, Zihan Zhu, Chris Mayes, Dylan Kennedy, Willie Neiswanger, Claudio Emma
Collaborators: Jan Kaiser, Annika Eichler, Malachi Schram, Kishan Rajput, J.P. Gonzalez-Aguillera







1,062 experiments in 2016

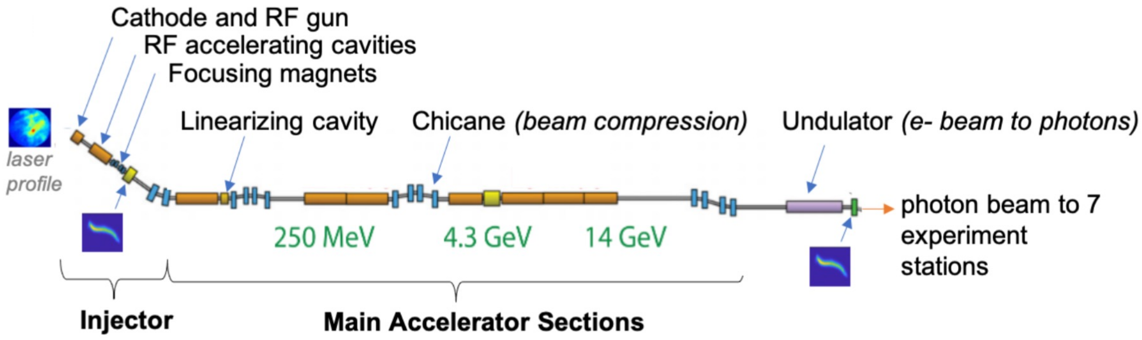
~1023 papers since 2009*

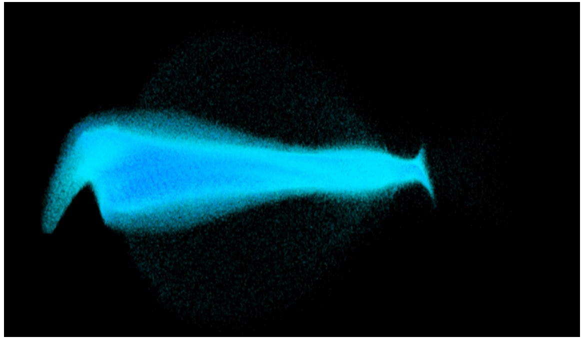
Experimenters come for a few days – a week

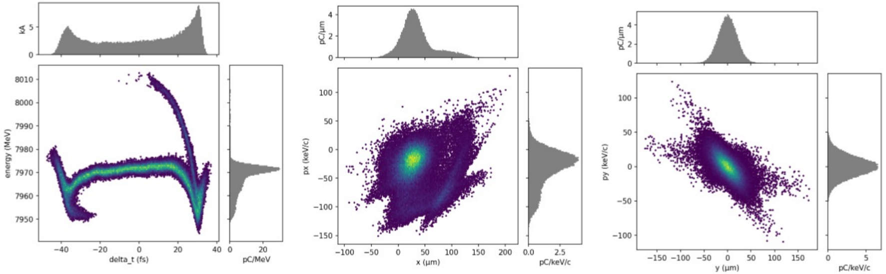
beam duration, x-ray wavelength etc. adjusted for each experiment

** Even more now; these numbers are a few years old*

https://lcls.slac.stanford.edu







Beam exists in 6-D position-momentum phase space

Have incomplete information from many different diagnostics (2D beam images, spectra, scalars)

Hundreds of controllable variables and millions to monitor

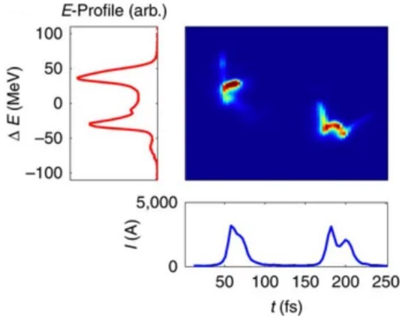
Many different user setups requested, machine in high-demand

Dynamic control during experiment (e.g. scan two bunch positions) increasingly requested

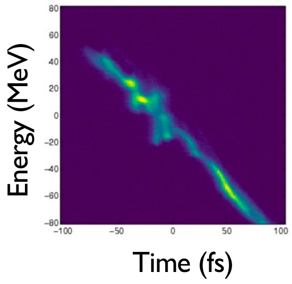
Time-dependent behaviors + slow drift over time

Nonlinear, high-dimensional optimization/control problem

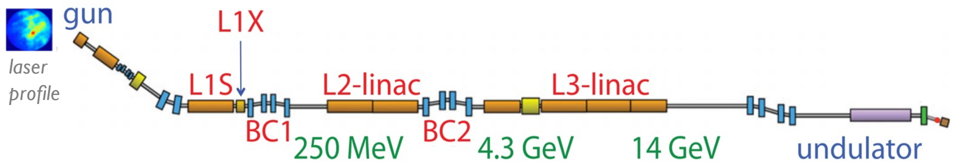
A. Marinelli, et al., Nat. Commun. **6**, 6369 (2015)



A. Marinelli, IPAC'18

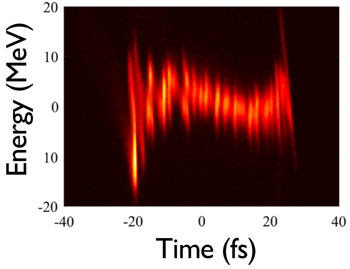


Variety of optimization/control needs

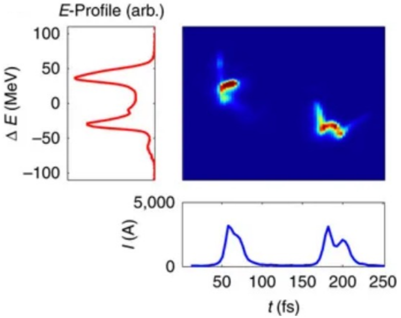


photon beam to
7 experiment
stations

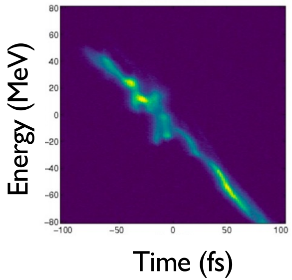
J. Qiang et al, PRAB (2017)



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



A. Marinelli, IPAC'18



Achieve unprecedented beam parameters *(new science)*	Rapid beam customization	Maintain stability for experiments *(tight tolerances)*	Actively scan bunch parameters

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

less ← assumed knowledge of machine → more

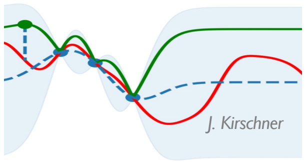
Model-Free Optimization

Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

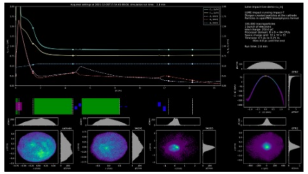


Update a model at each step

→ use model to help select the next point

Bayesian Optimization
Reinforcement Learning

Global Modeling + Feed-forward Corrections



Make fast system model

→ provide initial guess (i.e. warm start) for settings or fast compensation

ML system models +
inverse models

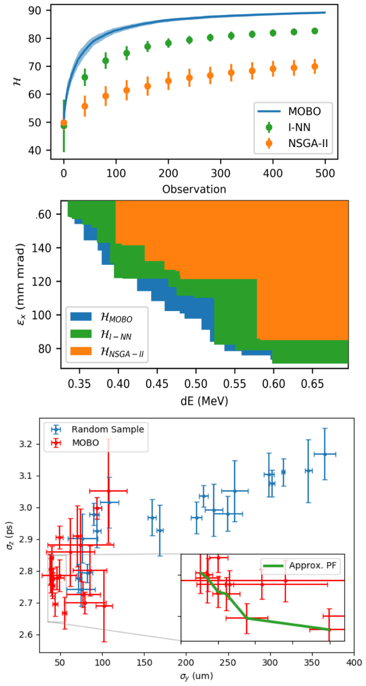
SLAC strategy: first use algos with minimal data overhead, then build up to more data-intensive / model-informed approaches

In practice: a lot of initial focus on BO, and now incorporating more system model information + moving toward deep RL

Many successes with Bayesian Optimization

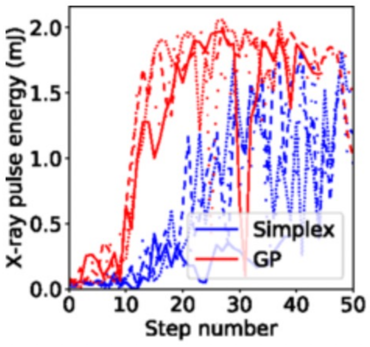
(+ improvements)

Multi-objective BO (+ exp. Pareto front)



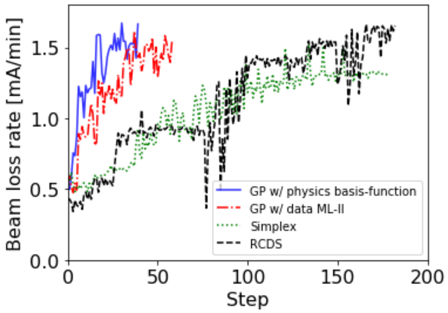
*Roussel et. al. PRAB , **2021***

FEL pulse energy tuning at LCLS + physics in kernel design



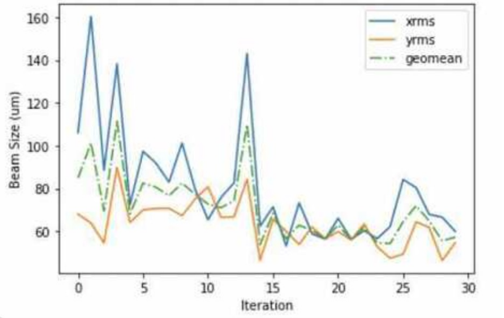
*Duris et. al. PRL , **2020***

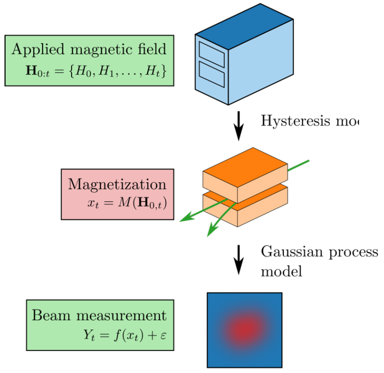
Loss rate tuning at SPEAR3 + physics kernel improvements



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

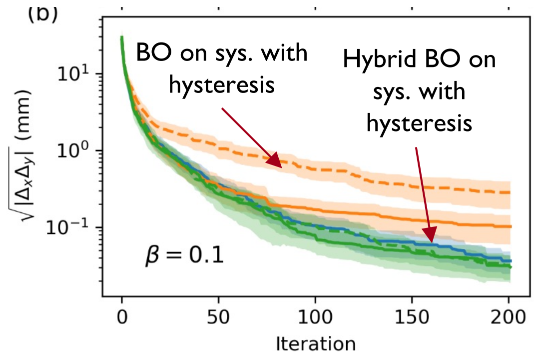
Sextupole tuning for IP at FACET-II (notorious to tune by hand)



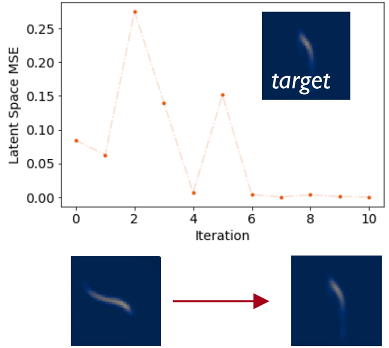


*Roussel et. al. PRL , **2022***

Higher-precision optimization possible when including magnetic hysteresis effects



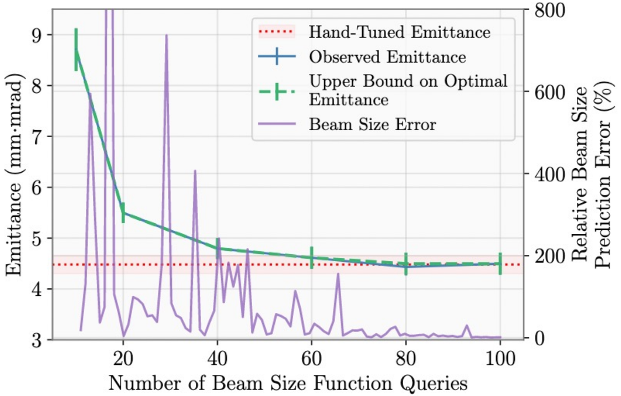
Longitudinal phase space tuning on LCLS



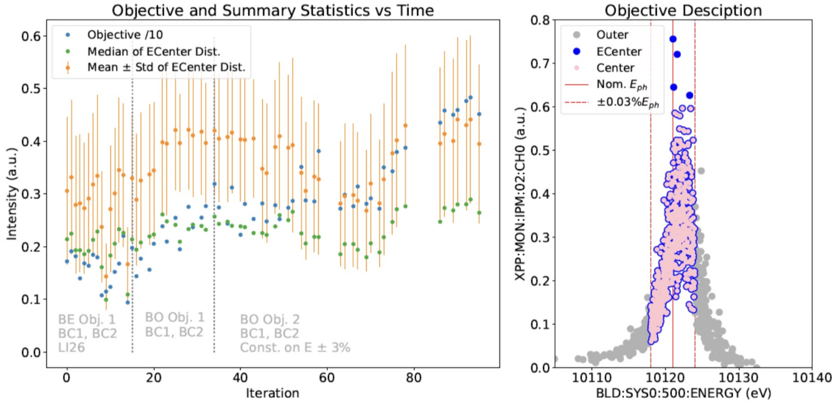
Many successes with Bayesian Optimization

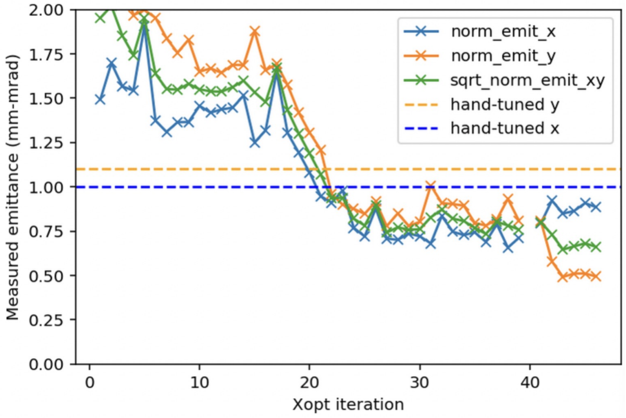
(+ improvements)

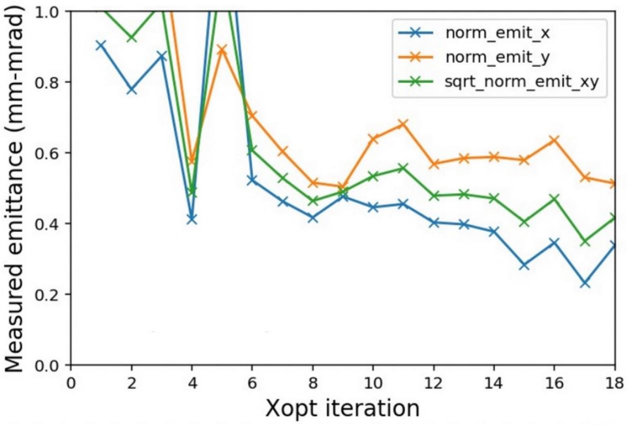
20X faster emittance tuning with BAX "virtual objective"



Monochrometer tuning (work in progress)





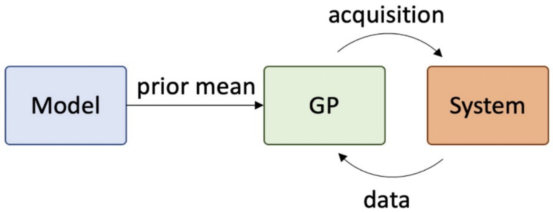


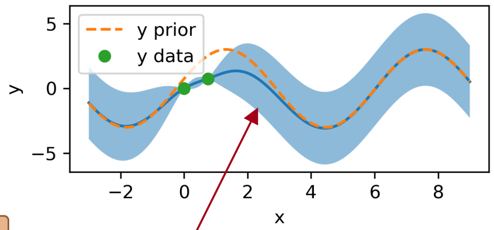
Optimize LCLS-II injector emittance while learning to keep dark current low

Neural Network System Models + Bayesian Optimization

Combining more expressive models with BO → **important for scaling up to higher-dimensional tuning problems (more variables)**

Good first step from previous work: use neural network system model to provide a prior mean for a GP





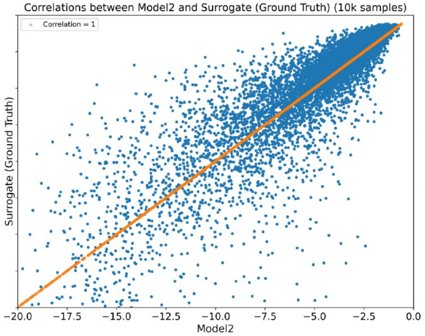
model prediction returns to prior

Used the LCLS injector surrogate model for prototyping

variables: *solenoid, 2 corrector quads, 6 matching quads*

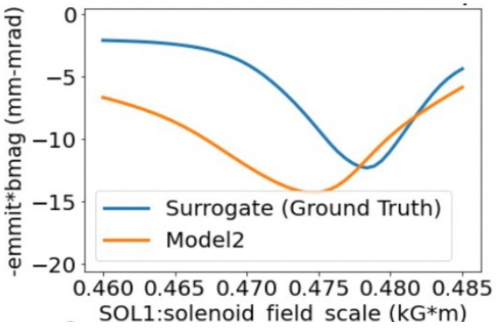
objective: *minimize emittance and matching parameter*

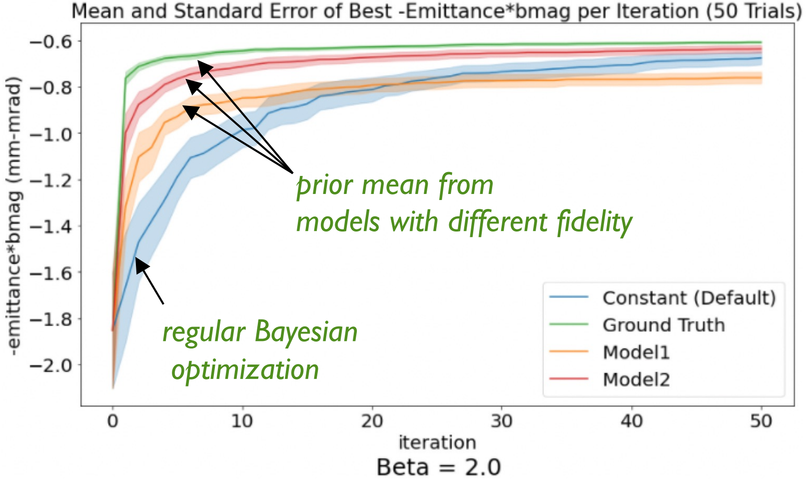
Correlations Between Predictions and Ground Truth



Ground Truth

Model 2





Even prior mean models with substantial inaccuracies provide a boost in initial convergence

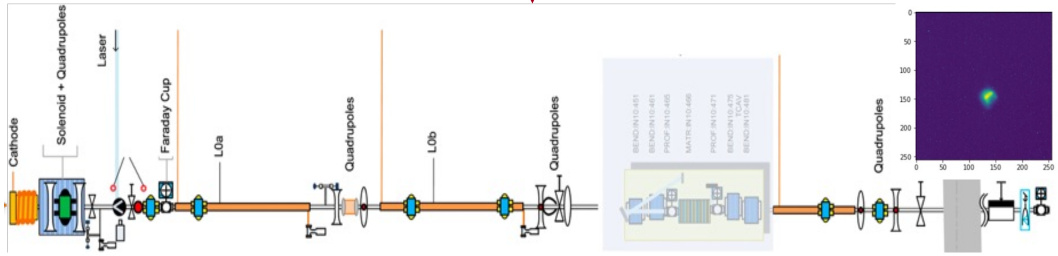
NeurIPS proceeding: *https://arxiv.org/abs/2211.09028*

“Bayesian Exploration” for Efficient Characterization

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

Comprehensive ML Models of Injector

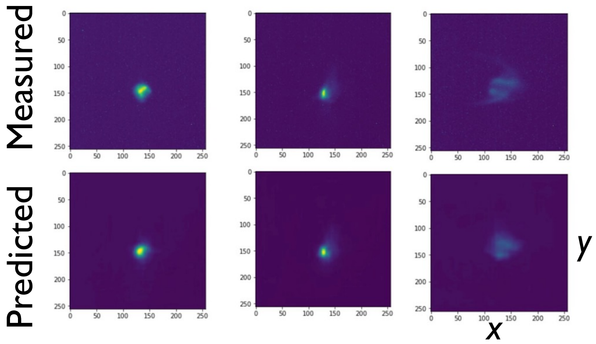
Setting changes on 10 variables (solenoid, bucking coil, corrector quads and matching quads)



FACET-II Injector

x-y emit, match, and beam images

transverse phase space



- 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**

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.

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 interface

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

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

X.data
```

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

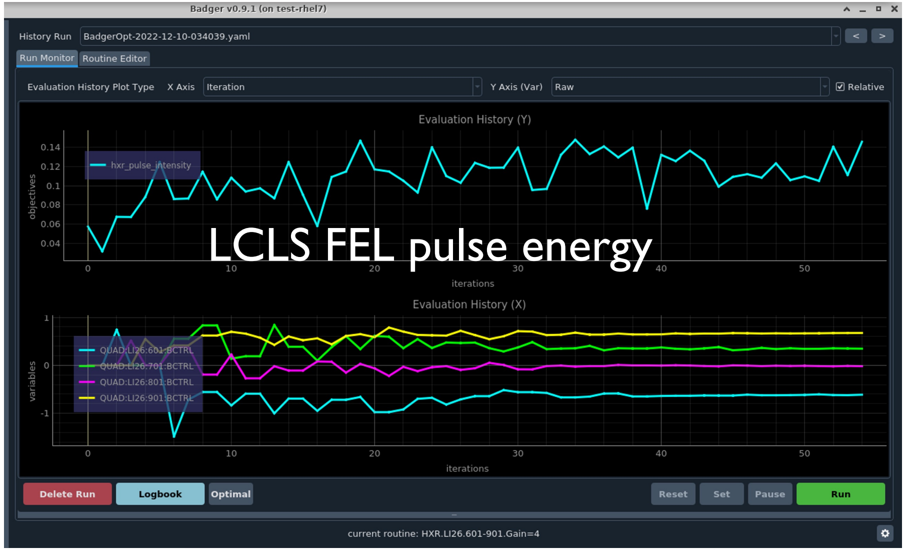
https://christophermayes.github.io/Xopt/

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%

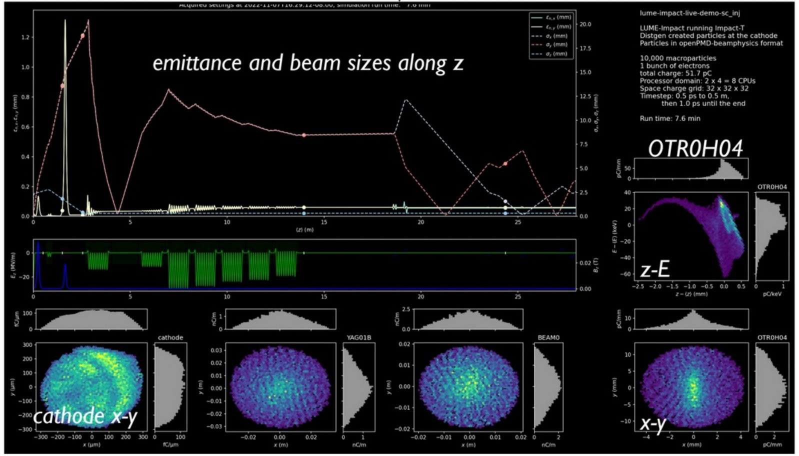
- Can specify constraints on settings and outputs (e.g. avoid dark current, beam losses, etc)
- Trust-region method allows conservative high-dimensional tuning (e.g. used >100 sextupoles at ESRF)
- Working on integrating global model priors → not learning from scratch each time
- Working to make compatible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

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

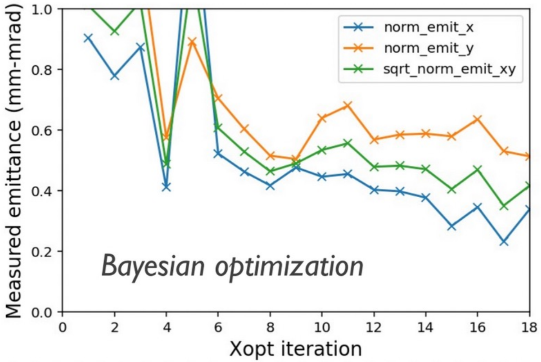


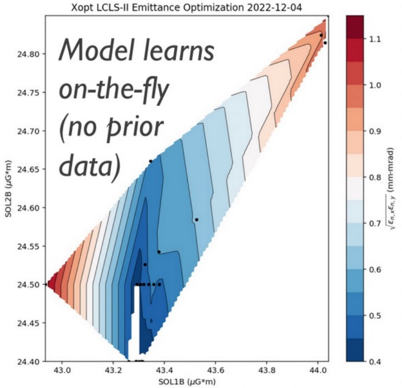
LCLS-II live sim: run on HPC and display in control room

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

That's a lot of success with BO … but we want RL too

BO is very useful in some contexts:

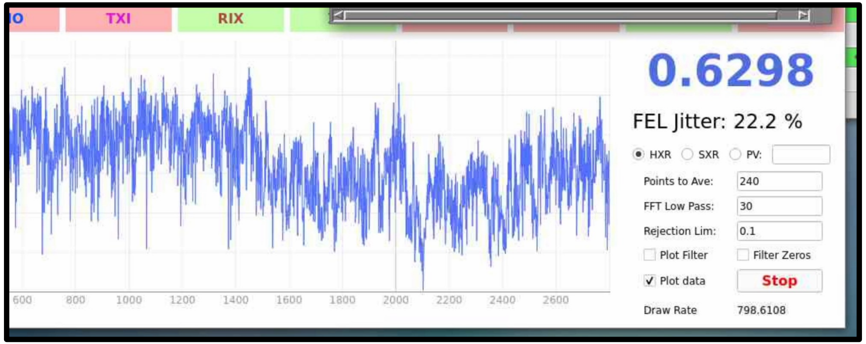
- Tune/characterize new systems/problems from scratch
- Output constraints/safety constraints (simple with BO)
- Tricks can aid convergence speed/dimensionality

RL can help address a different set of needs:

- Use global machine information / changes over time for rapid setup + fine-tuning *(interpolate in high dim. space)*
- Treat as a dynamical system to aid fine control/setup *(many time-dependent processes/feedbacks + drift)*
- Address demands for fast dynamic control from users

Suitability of accelerator tuning problems for RL:

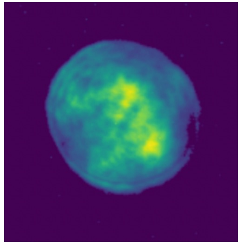
- Many variables, multi-modal signals (images, scalars, time series)
- Continuous state/action spaces
- Have physics models/simulators for many problems



120 Hz FEL pulse intensity

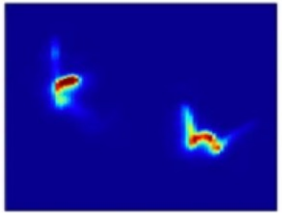
Nonlinear instability → sensitive to dynamic processes

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

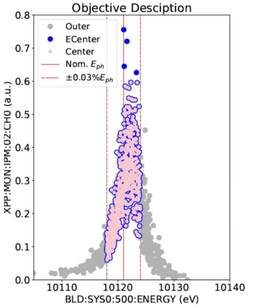


x-y laser

energy

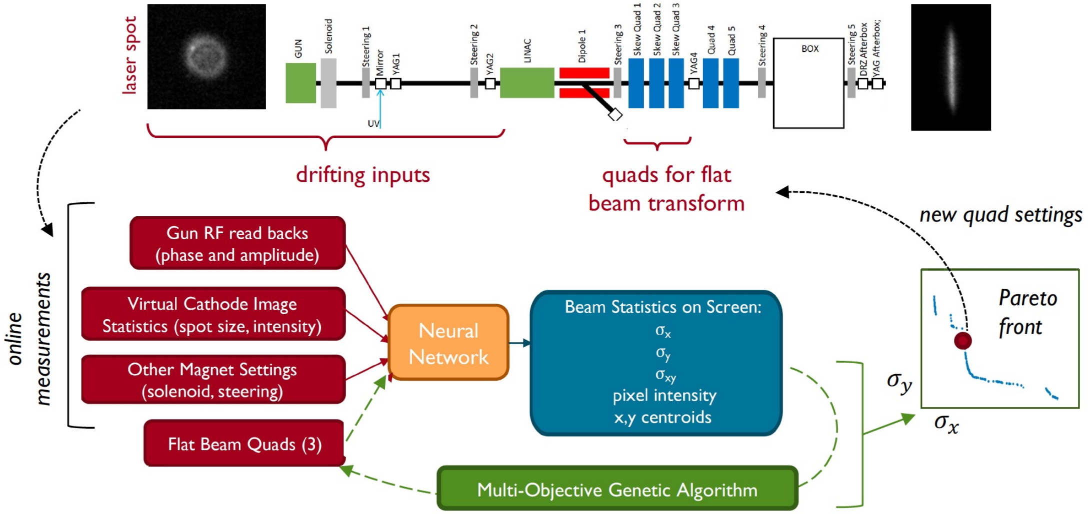


time



Variety of high dimensional signals for states, objectives

Example Problem: Compensate for Upstream Drift in Fast Setup



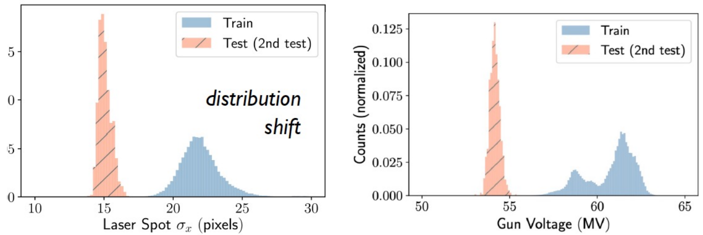
- Round-to-flat beam (RTFB) transforms are challenging to optimize; sensitive to upstream drift (e.g. in laser, rf systems)
  → *want to be able to set up RTFB quickly despite drift*

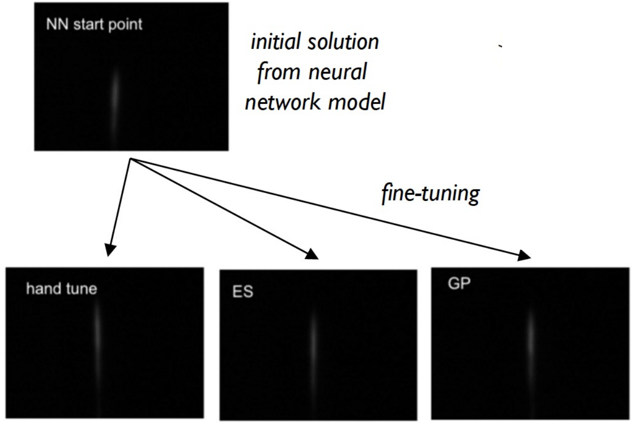
- 2019 study explored ability of a learned model and tuning algorithms to help

- NN model used as warm start for BO, extremum seeking, hand-tuning

- Trained DDPG Reinforcement Learning agent on NN model and tested on machine under different conditions

Can work even under distribution shift



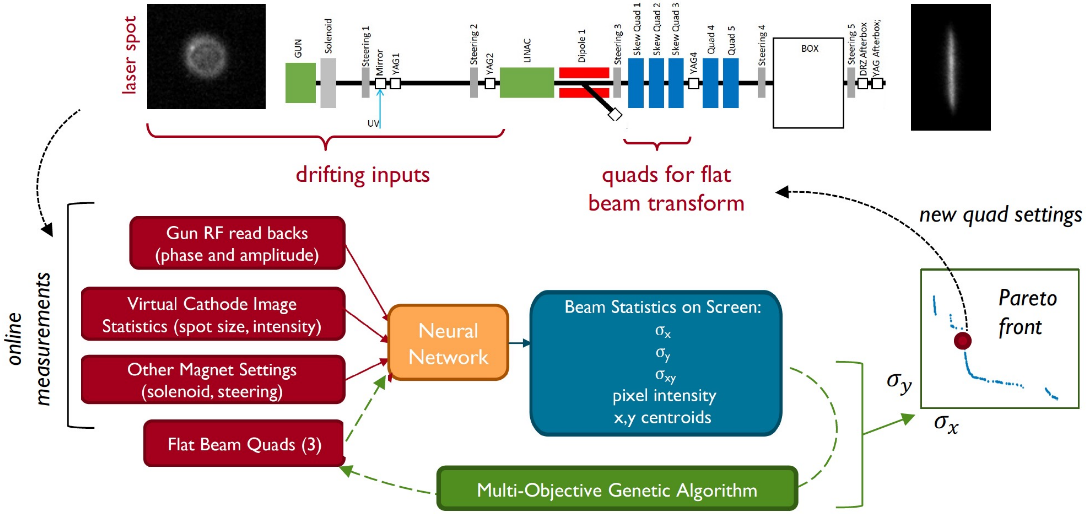


Hand-tuning in seconds vs. tens of minutes

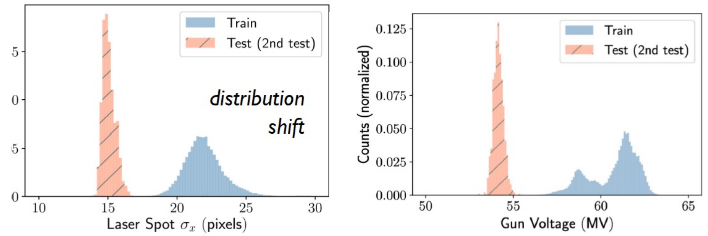
Boost in convergence speed for other algorithms

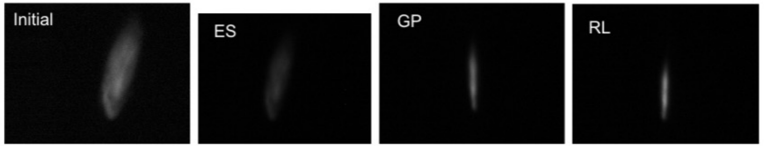
→ Broadly similar problem (at different scale) for LCLS/FACET-II switching between setups

Example Problem: Compensate for Upstream Drift in Fast Setup



Can work even under distribution shift





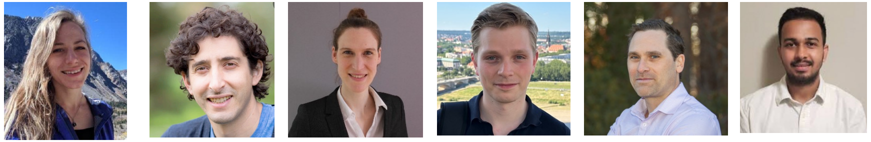
- Round-to-flat beam (RTFB) transforms are challenging to optimize; sensitive to upstream drift (e.g. in laser, rf systems)
  → *want to be able to set up RTFB quickly despite drift*
- 2019 study explored ability of a learned model and tuning algorithms to help
- NN model used as warm start for BO, extremum seeking, hand-tuning
- Trained DDPG Reinforcement Learning agent on NN model and tested on machine under different conditions

RL agent converged faster/more smoothly than BO

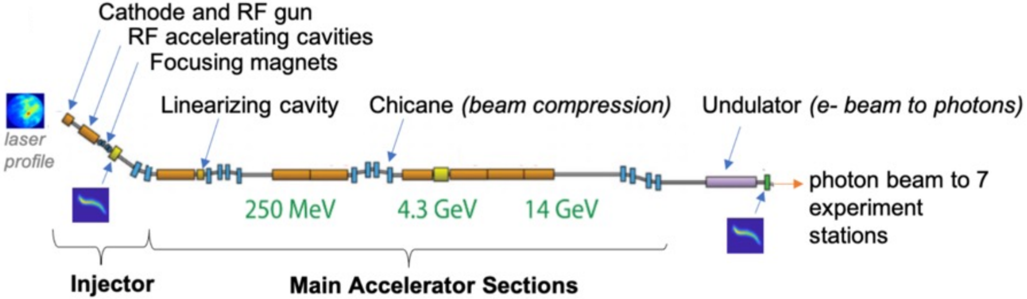
But was much larger overhead in prep +
we had concerns about the effort needed to generalize it

→ *Broadly similar problem (at different scale) for LCLS/FACET-II switching between setups*

RL for LCLS Accelerator

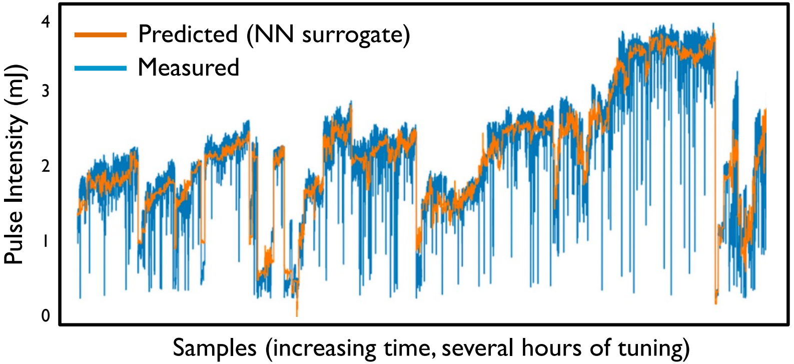


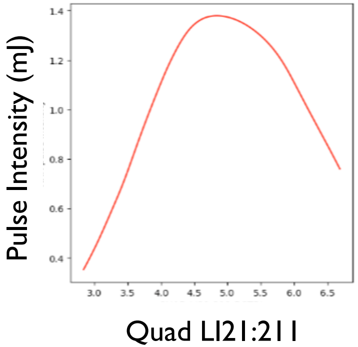
- Focusing on FEL pulse intensity tuning and quadrupole magnets first
- FEL is sensitive to focusing, trajectory; perturbing beam/feedbacks too much results in beam losses
- Using data-driven surrogates and differentiable sims (Cheetah and Bmad) to train agents (TD3, PPO)
- Iteratively add more data and variables:
    - Longitudinal phase space, spectra
    - RF phases and amp., undulator taper
    - Combine with photon beamline, trajectory control
- Expect first beam times very soon (weeks)



~28 focusing magnets tuned regularly for FEL pulse intensity

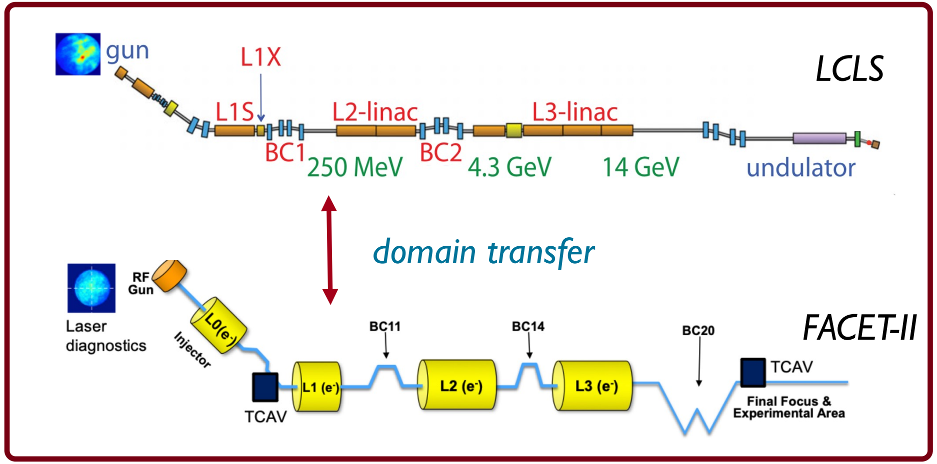
(many more variables to include: steering, rf, taper, drive laser)



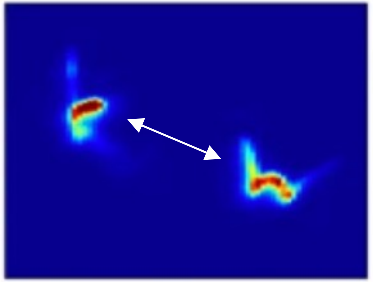


Needs/Opportunities at SLAC

- Uncertainty-aware / trust region RL needed
- Sample-efficient adaptation across setups needed
  (different charges, beam phase space, multi-bunch)
- Transfer learning between LCLS/LCLS-II/FACET-II
  → Similar layouts, component design, beam diagnostics, user needs (e.g. scan two bunches)
- Enabling fundamentally new capabilities
  - *FACET-II "extreme beams"; highly sensitive*
  - *Photon science requiring precise dynamic control*
- Comprehensive online system modeling + RL
  - *Physics sims + ML surrogates being deployed on local HPC connected to control system*
- RL with human feedback → *human-AI interaction in the control room is a current area of study*
- Fast feedback: LCLS-II kHz to MHz beam rate



fast dynamic beam customization

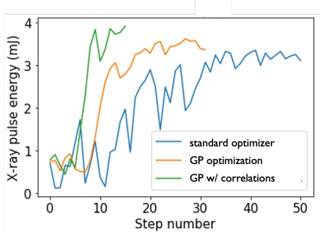


RL + human feedback

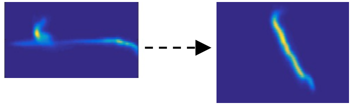
SLAC Pursuing ML for Accelerators Very Broadly

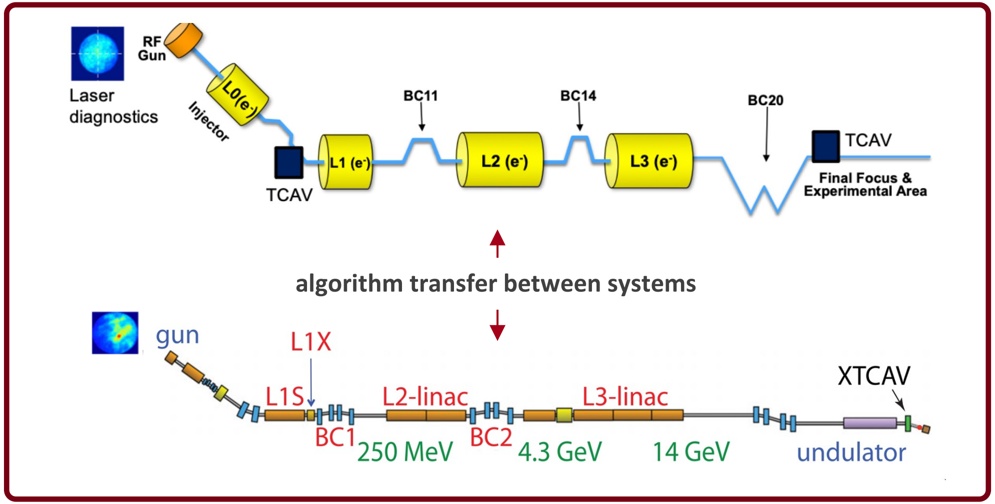
Data reduction/rejection *(kHz/MHz data streams)*
Event triggering

automated control
+ optimization

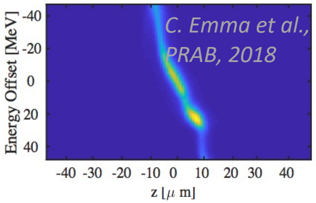


J. Duris et al., PRL, 2020

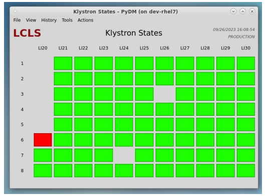


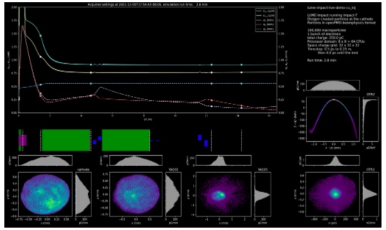


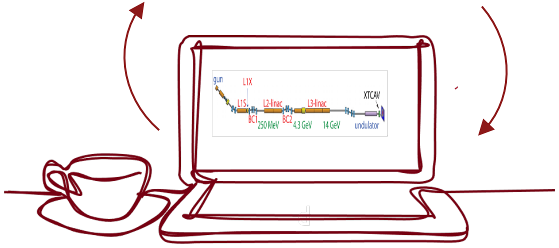
ML-enhanced diagnostics
(provide insight at faster rate, at higher resolution, non-invasively)



anomaly detection
failure prediction
(plan maintenance; alert to changes in machine; alert to interesting science)







extract unknown relationships + correlations
(feed into future control / design)

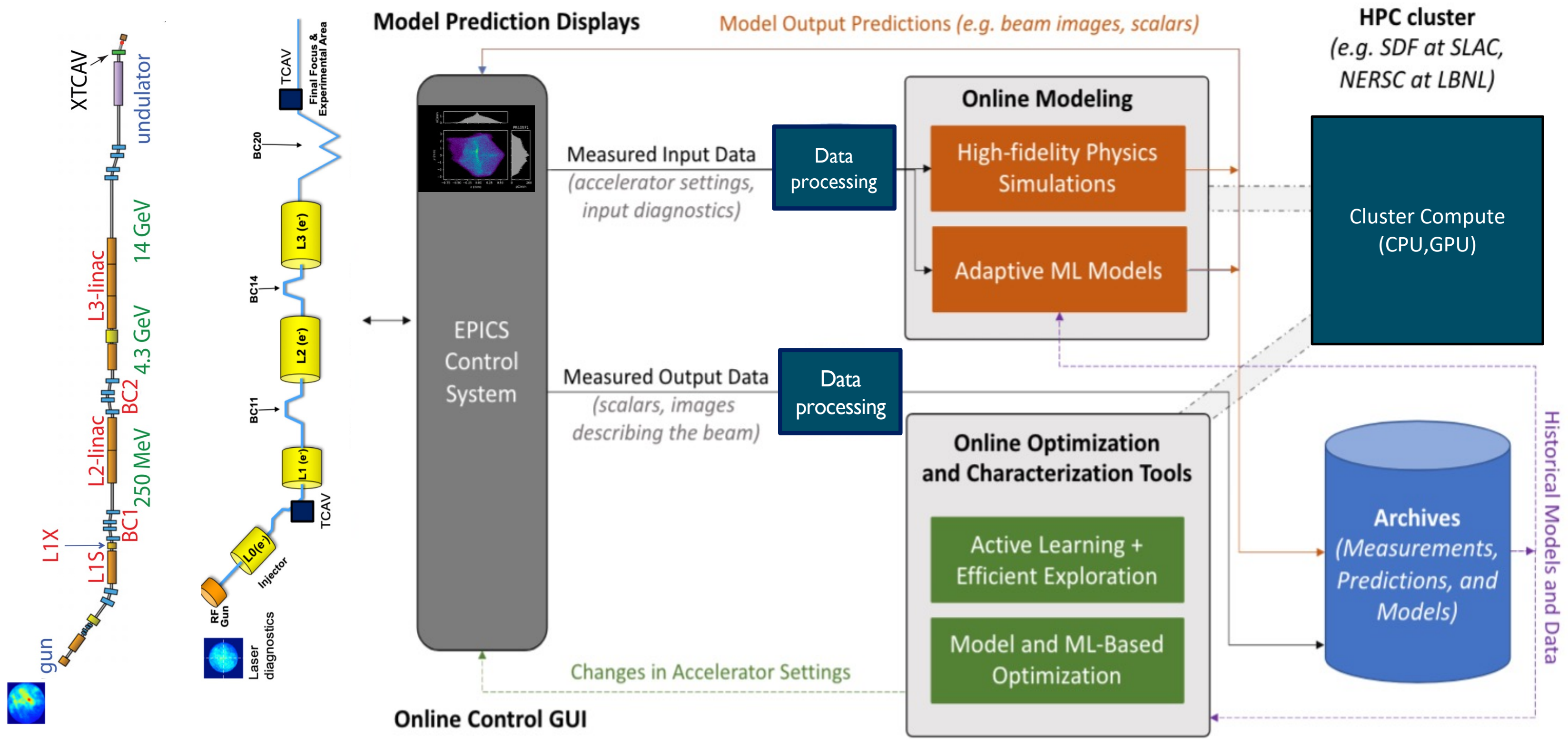
digital twins + online modeling
(fast sims, differentiable sims, model calibration, model adaptation)

+ need uncertainty quantification for all
+ can incorporate physics information in all

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)*

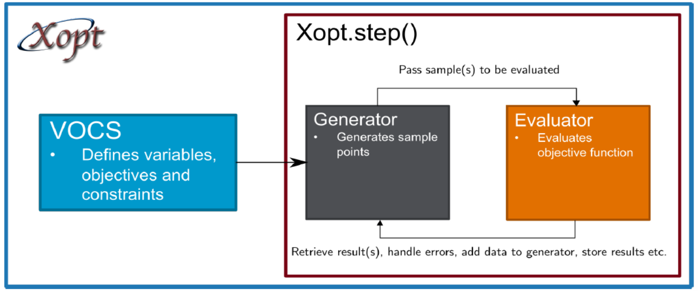


Making good progress toward this vision with open-source, modular software tools

Modular, Open-Source Software Development

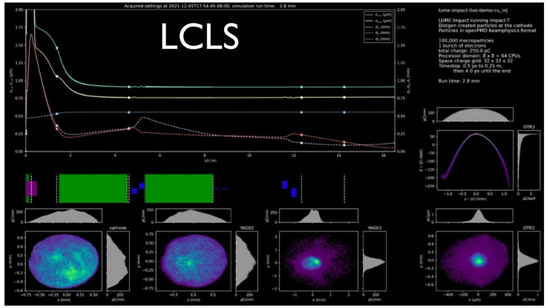
Community development of **re-usable, reliable, flexible software tools** for AI/ML workflows has been essential to maximize return on investment and ensure transferability between systems

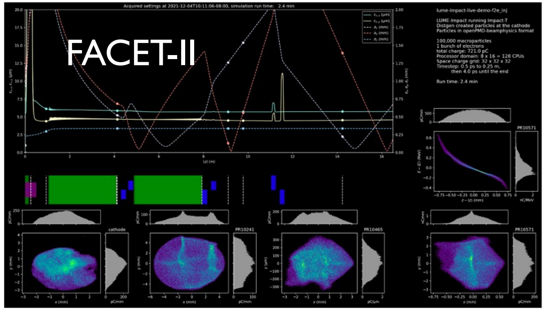
Modularity has been key: separating different parts of the workflow + using shared standards




```
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)*

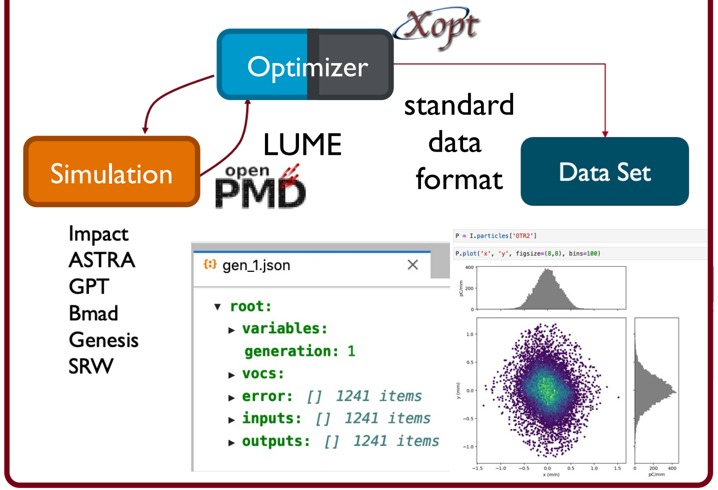
Visual control room interface *(e.g. Badger)*

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. We welcome new users and contributors.

Conclusion

- Initially focused on BO and its improvements → *have used very broadly at SLAC and beyond (see review https://arxiv.org/abs/2312.05667)*

- Well-established, portable, open-source tools for optimization *(Xopt, Badger)*

- Also focused on infrastructure for online modeling: *physics models and surrogate modeling approaches for faster, high-fidelity execution (training RL agents, online inference, etc)*

- Returning now to investigating RL → *deal with time-dependent behavior, larger parameter spaces, fast switching between setups + fine-tuning*

Thanks for your attention!
Any questions?











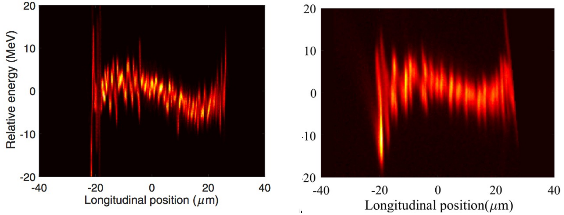


Backups

In reality things are much more difficult…

computationally expensive simulations

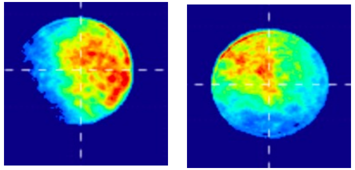
Simulation | Measurement

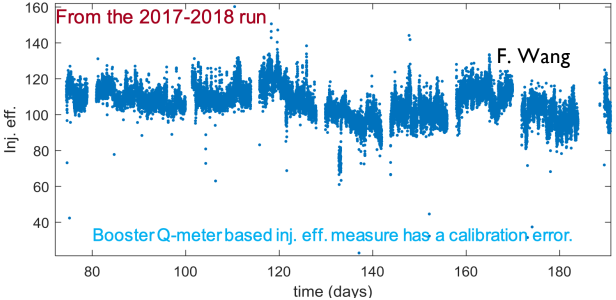


10 hours on thousands of cores at NERSC!

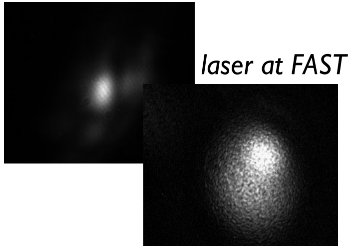
J. Qiang, et al., PRSTAB30, 054402, 2017

fluctuations/noise (e.g. laser spot)



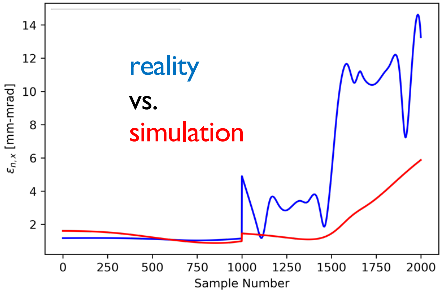


hidden variables / sensitivities

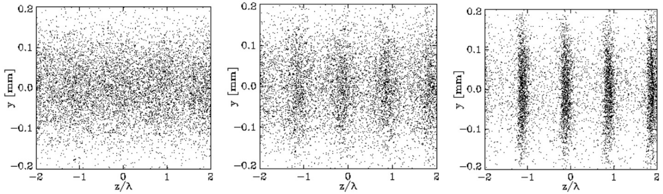


laser at FAST

drift over time



many small, compounding sources of uncertainty



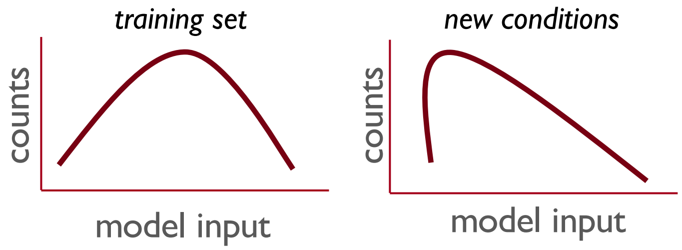
nonlinear effects / instabilities

AI/ML is well-positioned to help address these challenges

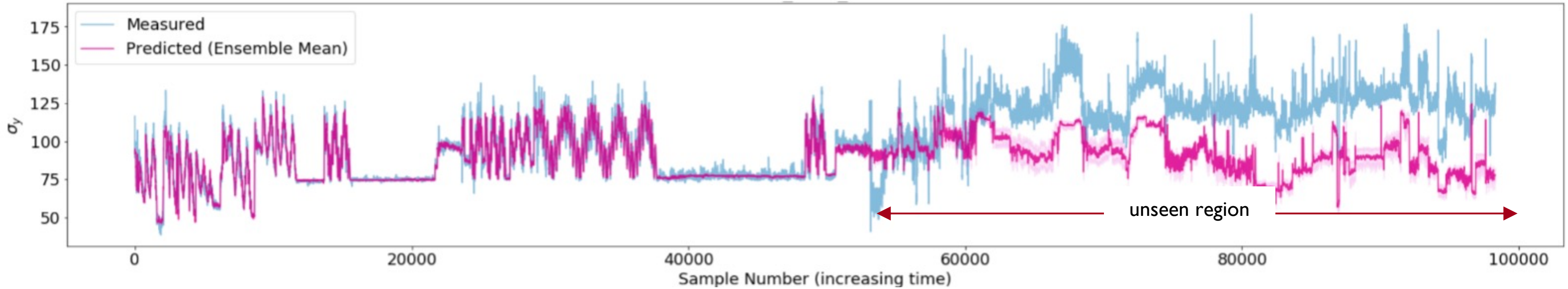
Uncertainty Quantification / Robust Modeling / Model Adaptation

Major area of AI/ML research: statistical distribution shift between training and test data degrades prediction

Distribution shift is extremely common in accelerators, due to both deliberate changes in beam configuration and uncontrolled or hidden variables



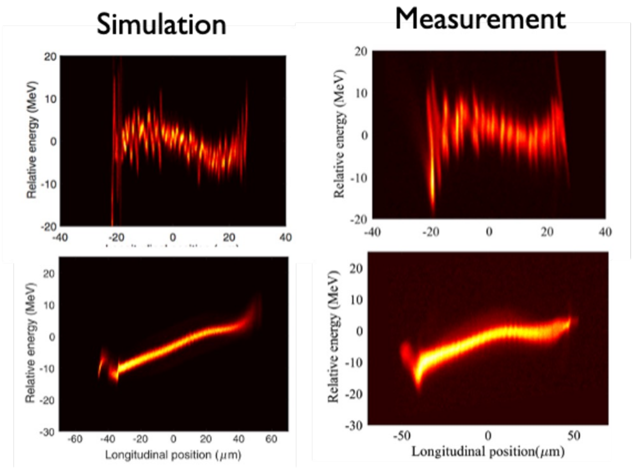
Example: beam size prediction and uncertainty estimates under drift from a neural network
Uncertainty estimate from neural network ensemble does not cover prediction error, but does give a qualitative metric for uncertainty



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

Fast-Executing, Accurate System Models

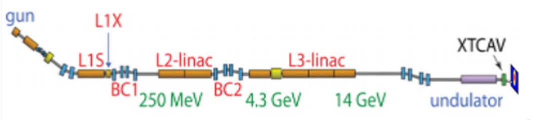
Accelerator simulations that include nonlinear and collective effects are powerful tools, but they can be computationally expensive



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

10 hours on thousands of cores at NERSC!

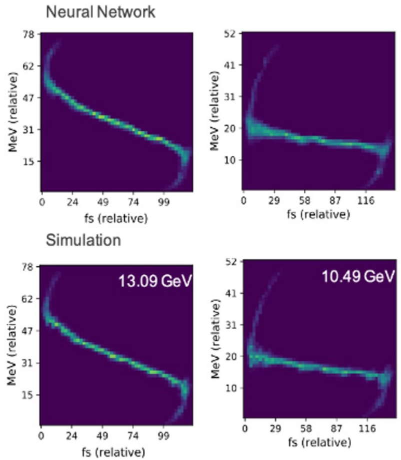
ML models can provide fast approximations to simulations ("surrogate models")



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

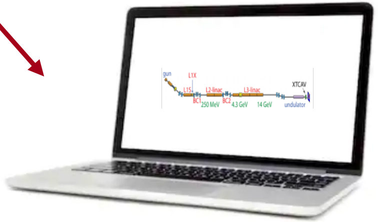
10^6 times speedup

ML modeling enables high-fidelity predictions of system responses with unprecedented speeds, opening up new avenues for high-fidelity online prediction, tracking of machine behavior, and model-based control

Fast-Executing, Accurate System Models

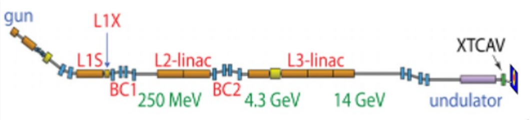
Bringing simulation tools from HPC systems to online/local compute

Online prediction
Model-based control



Control prototyping
Experiment planning

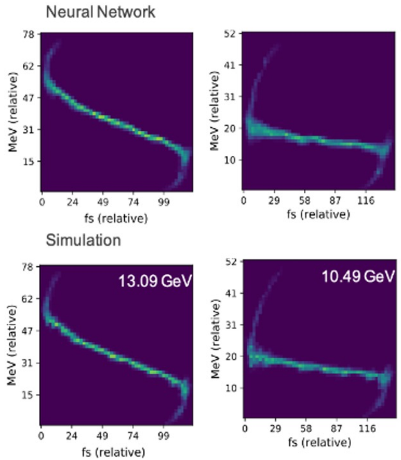
ML models can provide fast approximations to simulations ("surrogate models")



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



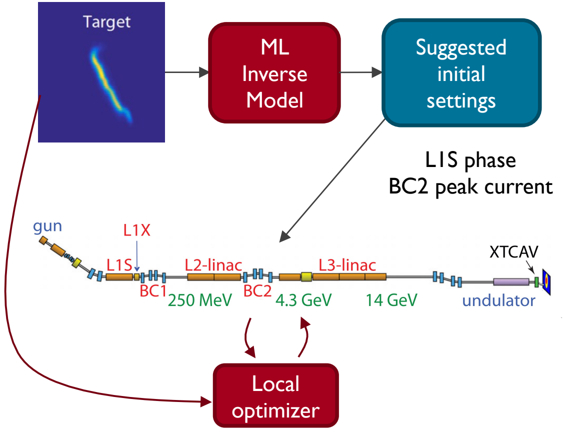
< ms execution speed

10^6 times speedup

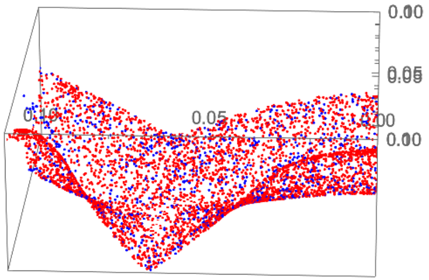
ML modeling enables high-fidelity predictions of system responses with unprecedented speeds, opening up new avenues for high-fidelity online prediction, tracking of machine behavior, and model-based control

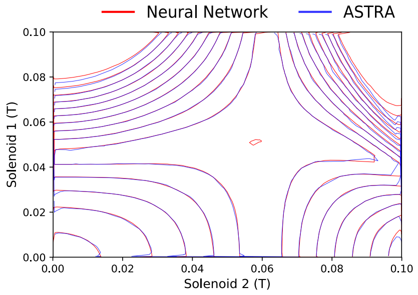
Warm starts for optimization

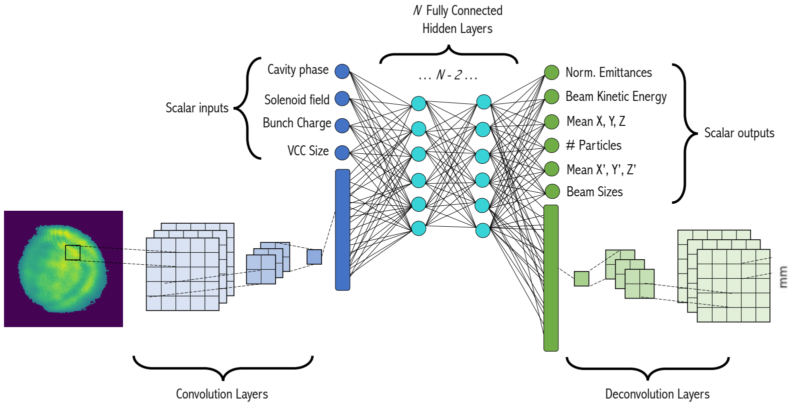
A. Scheinker, A. Edelen, et al, PRL, 2018



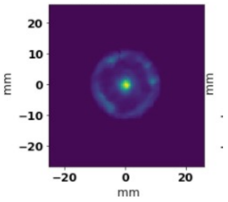
Smooth interpolation
Example σ_x surface from 2D scan, LCLS-II Injector

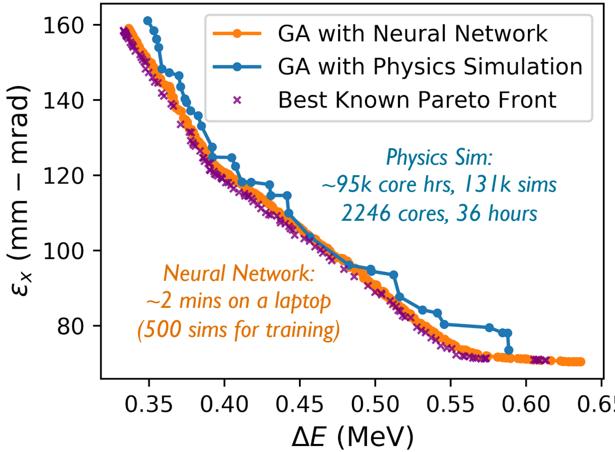






L. Gupta, et al, MLST, 2021



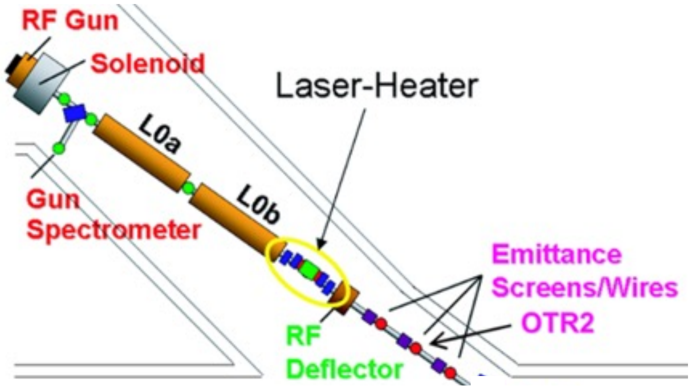


A. Edelen et al., PRAB, 2020

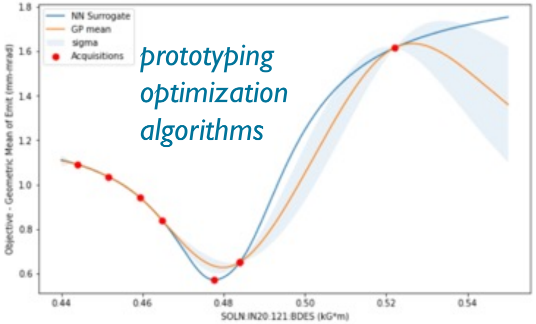
Include high-dimensional input information → better output predictions

Surrogate-boosted design optimization (example on AWA)

Example: Injector Surrogate Model at LCLS

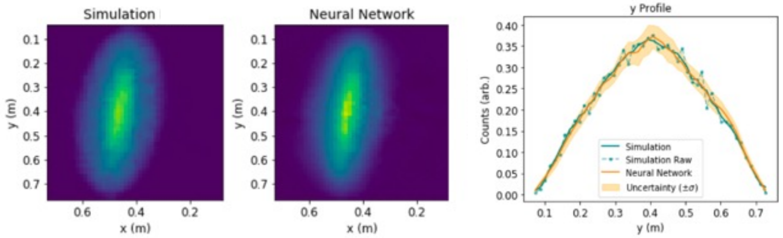


- ML models trained on physics simulations
- Inputs sampled widely across valid ranges
- **Used to develop/prototype new algorithms before testing online at FACET-II and LCLS** *e.g. new Bayesian optimization methods, adaptive emittance measurement*

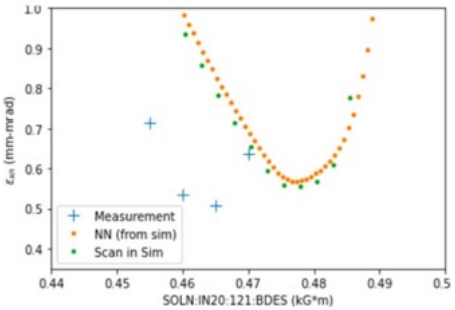


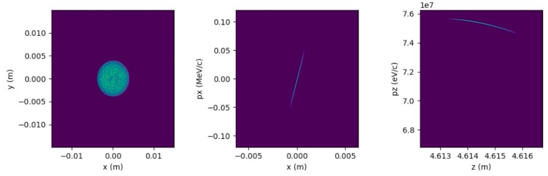
prototyping optimization algorithms

ML model provides accurate replication of simulation



Simulation and ML model trained on it are qualitatively similar to measurements





interactive model widget and visualization tools

ML models trained on simulations enable fast prototyping of new optimization algorithms → greatly reduces development time

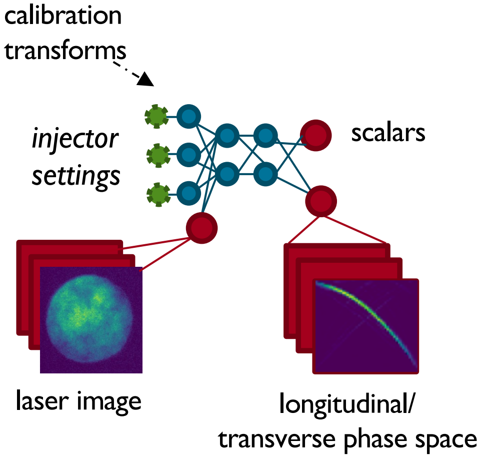
Finding Sources of Error Between Simulations and Measurement

Many non-idealities not included in physics simulations:

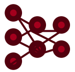
- **static error sources** (e.g. magnetic field nonlinearities, physical offsets)
- **time-varying changes** (e.g. temperature-induced phase calibrations)

*Want to identify these to get **better understanding of machine***

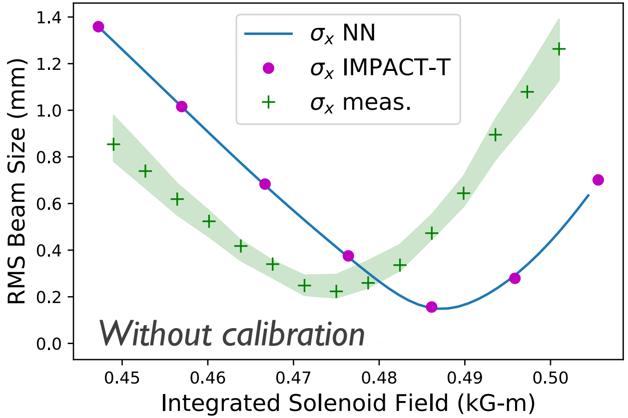
→ fast-executing ML model allows fast / automatic exploration of possible error sources

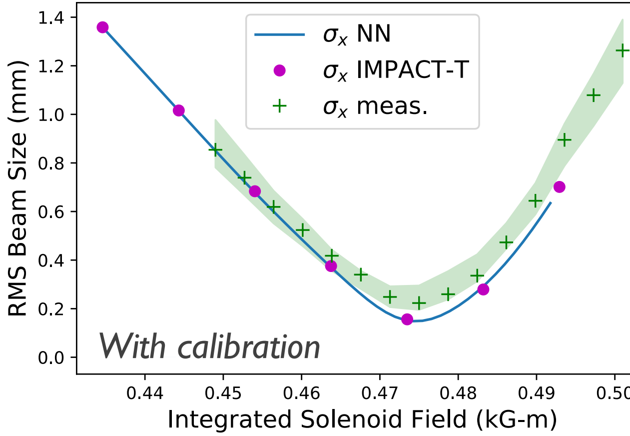


Inputs
Laser radius
Laser spot sizes
Pulse length
Charge
Solenoid
L0A phase
L0B phase
SQ quad
CQ quad
6 matching quads



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

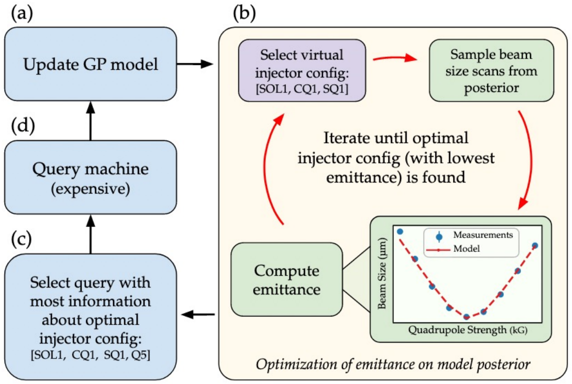


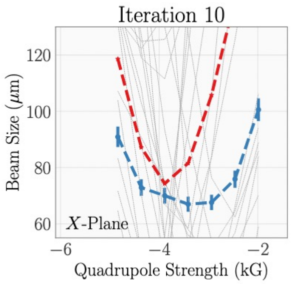


Here: calibration offset in solenoid strength found automatically with neural network model (trained first in simulation, then calibrated to machine)

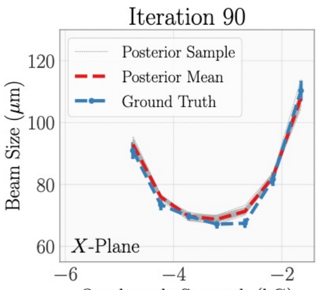
Efficient Emittance Optimization with Partial Measurements

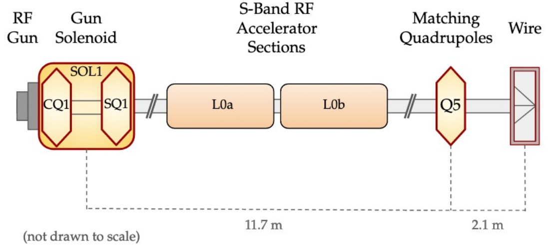
- **Instead of tuning on costly emittance measurements directly: learn a fast-executing model online for beam size while optimizing** → *learn on direct observables (e.g. beam size); do inferred "measurements" (e.g. emittance)*
- **New algorithmic paradigm leveraging "Bayesian Algorithm Execution" (BAX)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

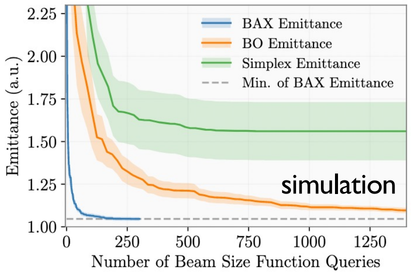


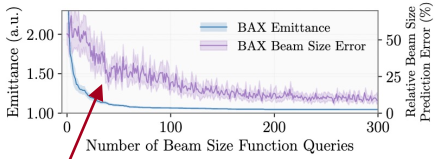


model is learned on-the-fly

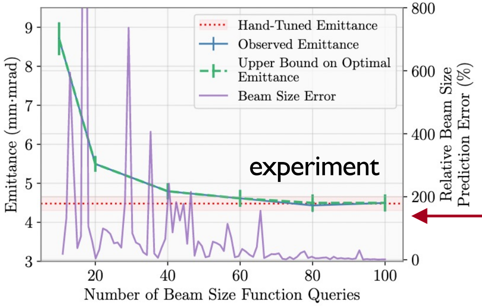








Convergence of beam size prediction error gives practical indicator of optimization convergence *(no need to do direct emittance measurement until the end)*



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.***