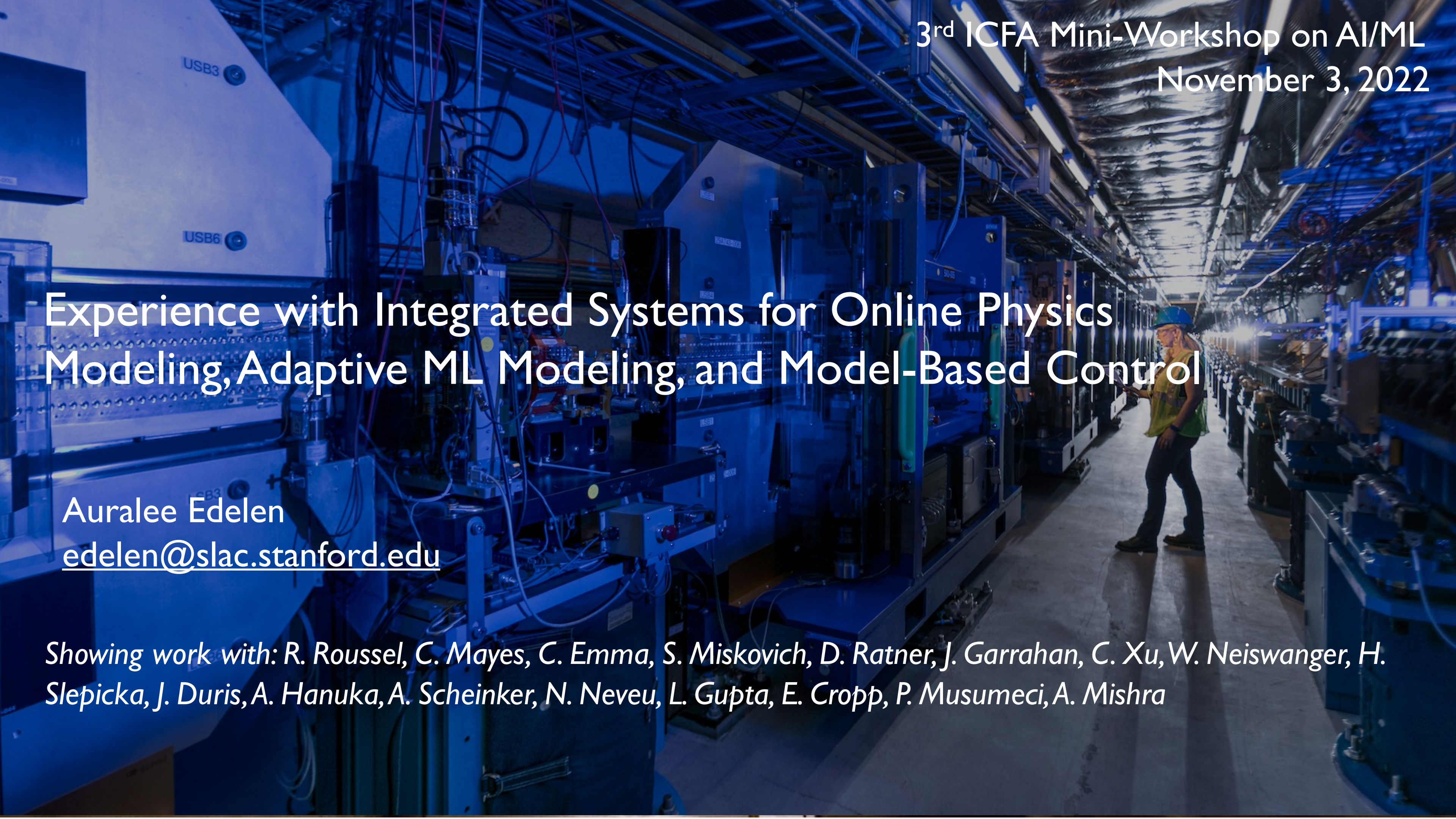


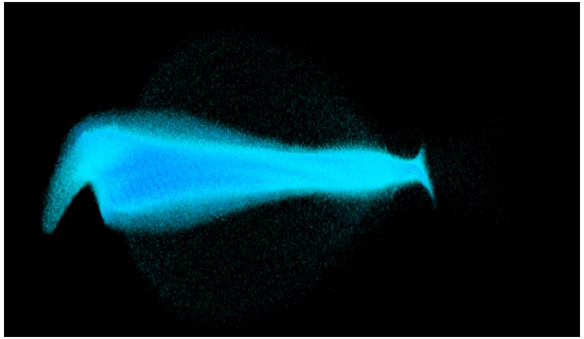
3rd ICFA Mini-Workshop on AI/ML
November 3, 2022

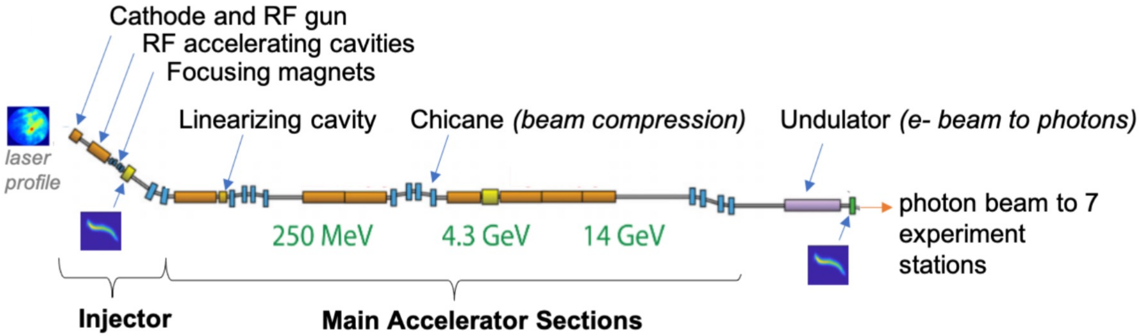
Experience with Integrated Systems for Online Physics Modeling, Adaptive ML Modeling, and Model-Based Control

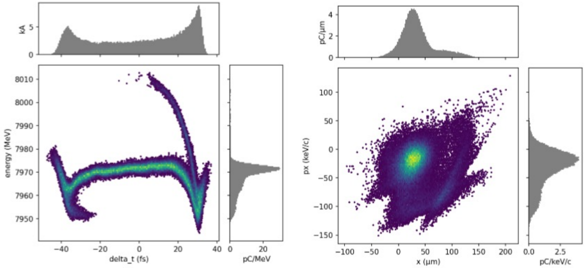
Auralee Edelen
edelen@slac.stanford.edu

Showing work with: R. Roussel, C. Mayes, C. Emma, S. Miskovich, D. Ratner, J. Garrahan, C. Xu, W. Neiswanger, H. Slepicka, J. Duris, A. Hanuka, A. Scheinker, N. Neveu, L. Gupta, E. Cropp, P. Musumeci, A. Mishra

Many tuning problems at LCLS/LCLS-II and FACET-II at SLAC require detailed phase space customization for different experiments





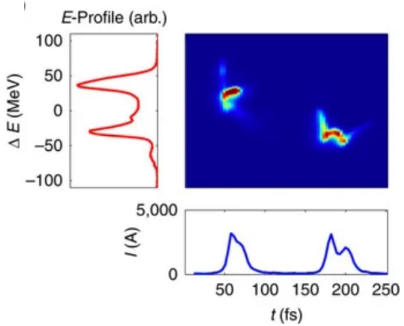


Beam exists in 6-D position-momentum phase space

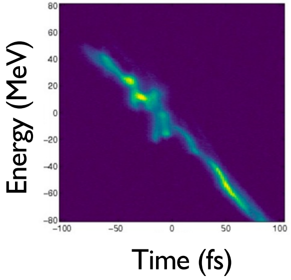
Have incomplete information: *measure 2-D projections or reconstruct based on perturbations of upstream controls (e.g. tomography, quad scans)*

Have dozens-to-hundreds of controllable variables and hundreds-of-thousands (up to millions for LCLS-II) to monitor

Nonlinear, high-dimensional optimization problem

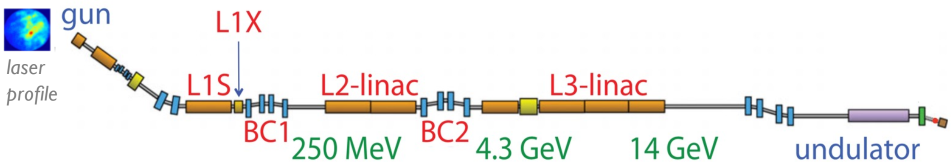


A. Marinelli, et al., Nat. Commun. 6, 6369 (2015)

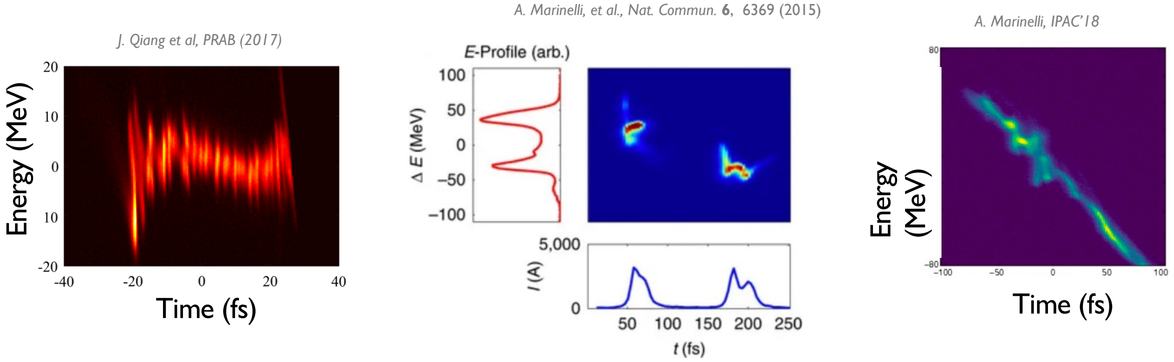


A. Marinelli, IPAC'18

wide spectrum of tuning needs



photon beam to 7 experiment stations



Rapid beam customization

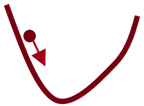
Achieve new configurations + unprecedented beam parameters

Fine control to maintain stability within tolerances

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

less ← assumed knowledge of machine → more

Model-Free Optimization

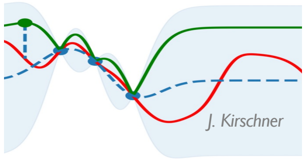


Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

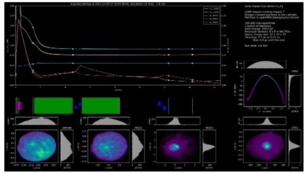


Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



Make fast system model

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

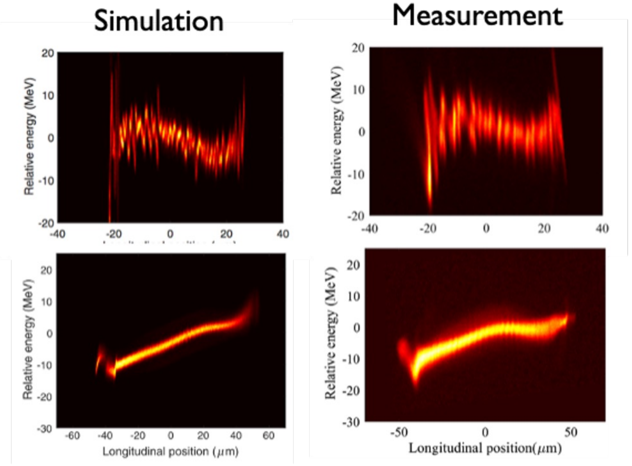
ML system models +
inverse models

Tuning research at SLAC is aimed at combining the strengths of different approaches.

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

Fast-Executing, Accurate System Models

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



10 hours on thousands of cores at NERSC!

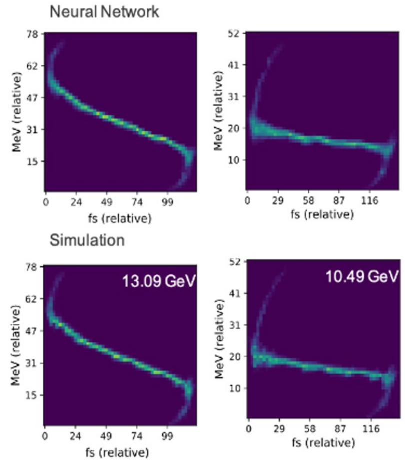
J. Qiang, et al., PRSTAB30, 054402, 2017

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

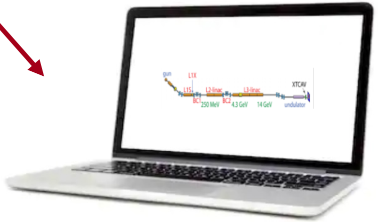
Edelen et al., NeurIPS 2019

ML modeling enables accurate 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

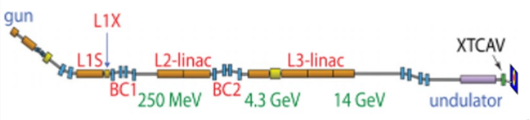
Bringing simulation tools from HPC systems to online/local compute

Online prediction
Model-based control



Control prototyping
Experiment planning

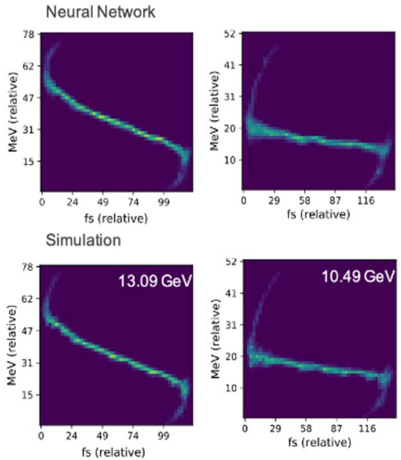
ML models are able to provide fast approximations to simulations ("surrogate models")



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

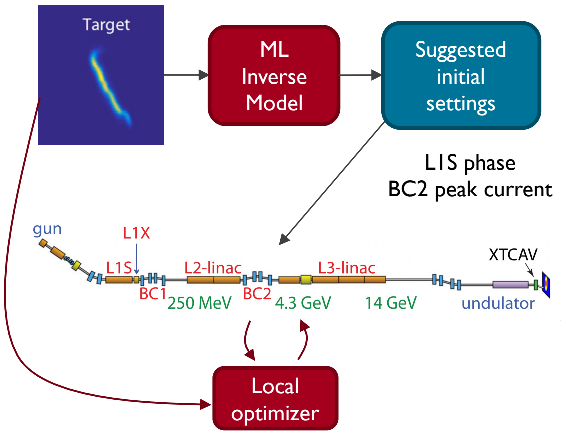
10^6 times speedup

Edelen et al., NeurIPS 2019

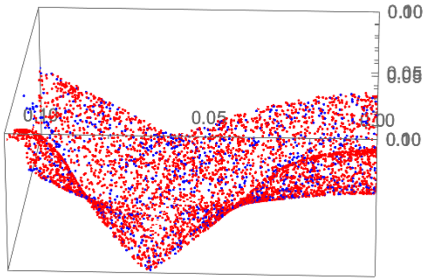
ML modeling enables accurate predictions of system responses with unprecedented speeds, opening up new avenues for high-fidelity online prediction, tracking of machine behavior, and model-based control

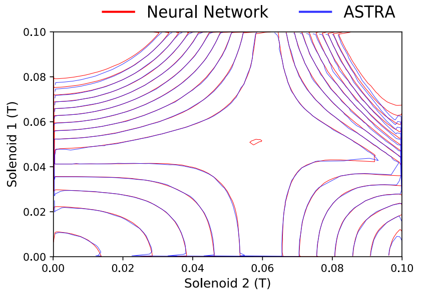
Warm starts for optimization

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

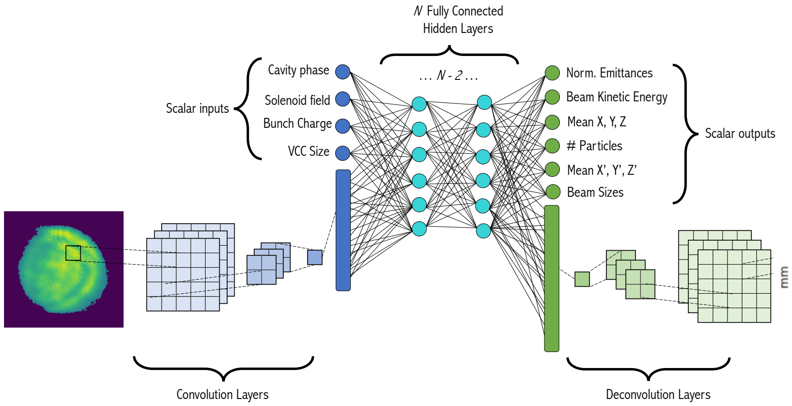


Smooth interpolation
Example σ_x surface from 2D scan, LCLS-II Injector

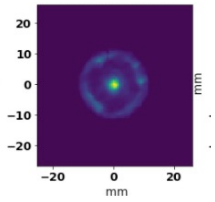




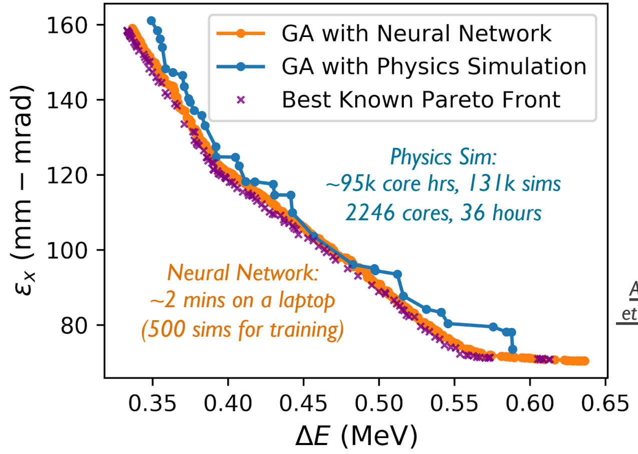
Edelen et al., NeurIPS 2019



L. Gupta, et al., MLST, 2021



Include high-dimensional input information → better output predictions

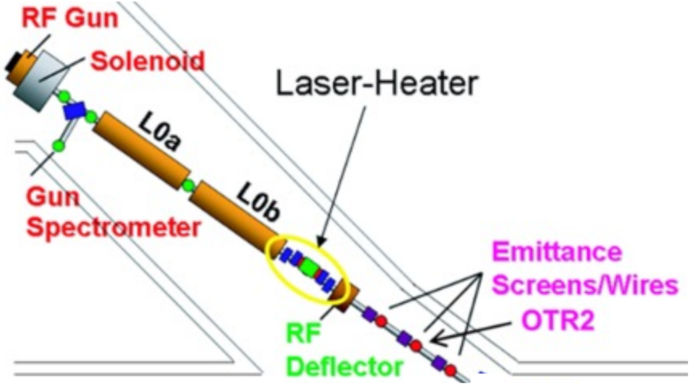


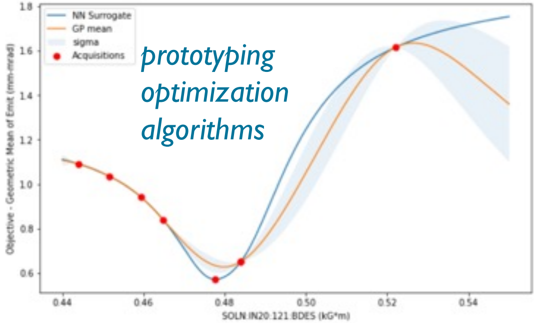
A. Edelen et al., PRAB, 2020

Surrogate-boosted design optimization

In Regular Use: Injector Surrogate Model at LCLS

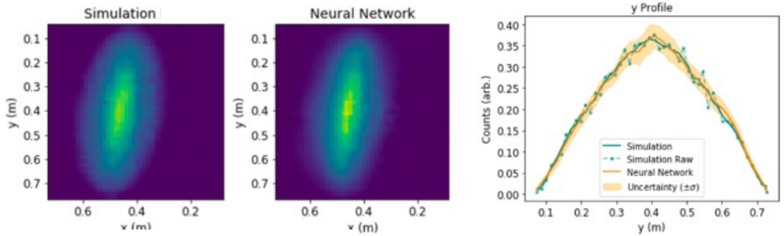
- ML models trained on detailed physics simulations (IMPACT-T)
- Inputs sampled widely across valid ranges of settings
  → *specifically leave out ranges of variables to test generalization*
- **Used to develop/prototype new algorithms before testing online at FACET-II and LCLS** *e.g. new optimization methods such as BAX (see S. Miskovich talk), adaptive emittance measurement*
- Getting set up to provide initial twiss parameters for downstream online model continuously (use regularly in optics matching)



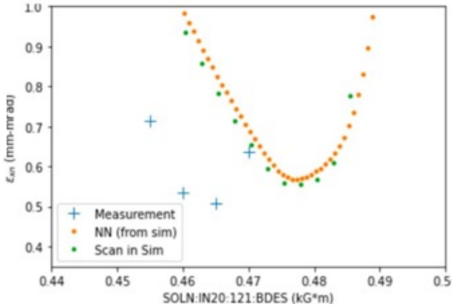


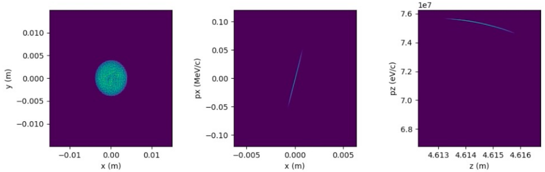
prototyping optimization algorithms

ML model provides accurate replication of simulation



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

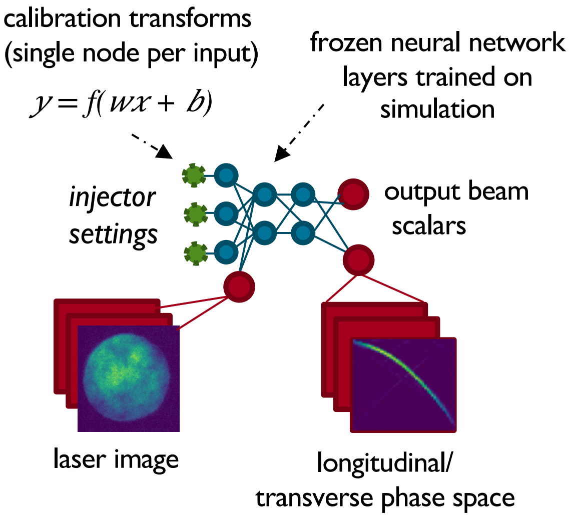
ML models trained on simulations have enabled fast prototyping of new optimization algorithms
→ has greatly reduced algorithm development time

Finding Sources of Error Between Simulations and Measurement

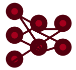
Many non-idealities not included in physics simulations:

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

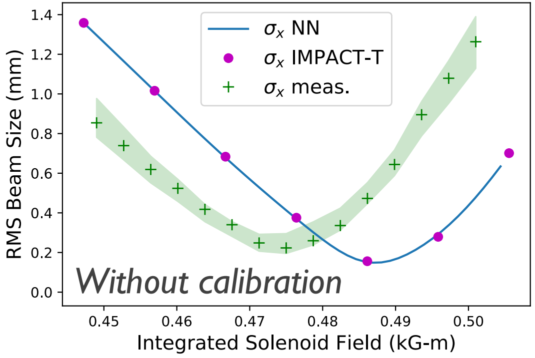
Want to identify these to get ***better understanding of machine*** *→* ***fast-executing ML model allows fast / automatic exploration of possible error sources simultaneously***

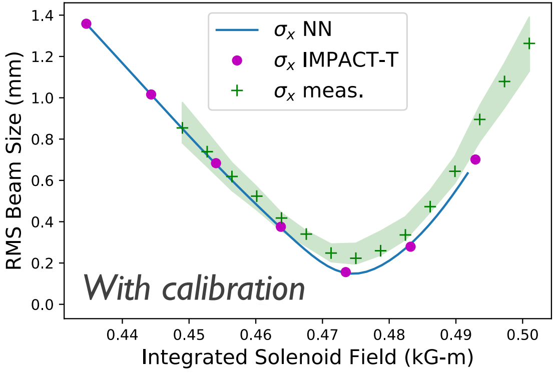


Inputs
Laser radius
Laser spot sizes
Pulse length
Charge
Solenoid
L0A phase
L0B phase
SQ quad
CQ quad
6 matching quads



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





Calibration offset in solenoid strength found automatically with neural network model (trained in simulation, then calibrated to machine)

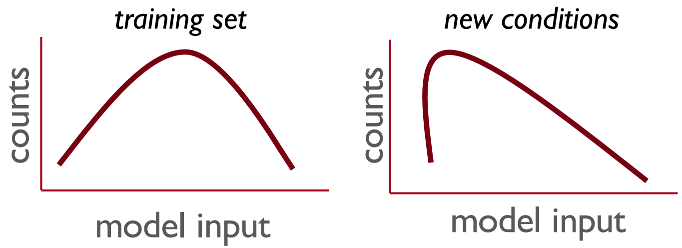
Example above is simulation-to-machine, but can adapt model over time as well

First studies look promising → current work focuses on examining robustness and extending to larger subsystems

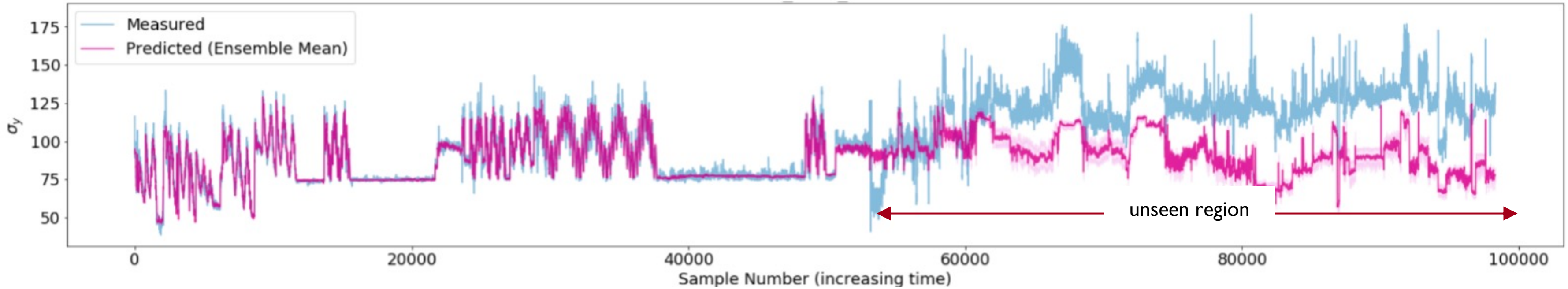
Uncertainty Quantification / Robust Modeling

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

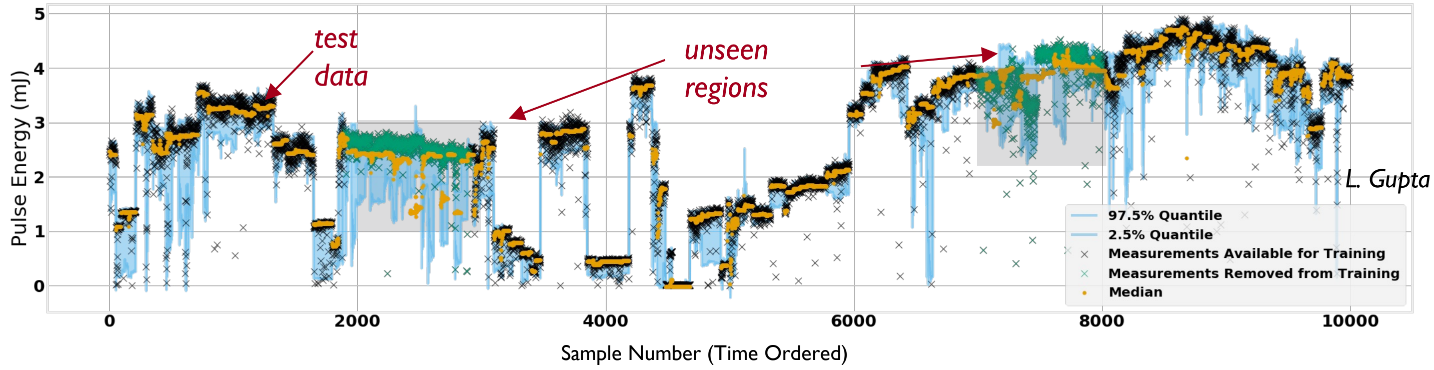


Reliable uncertainty estimates and uncertainty calibration methods are key for putting online models to use operationally

Uncertainty Quantification / Robust Modeling

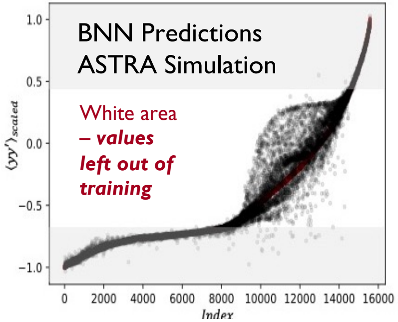
Essential for decision making under uncertainty (e.g. safe opt., intelligent sampling, virtual diagnostics)

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



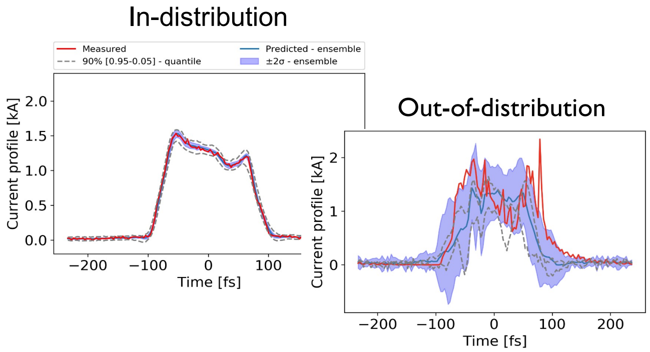
L. Gupta

Neural network with quantile regression predicting FEL pulse energy at LCLS



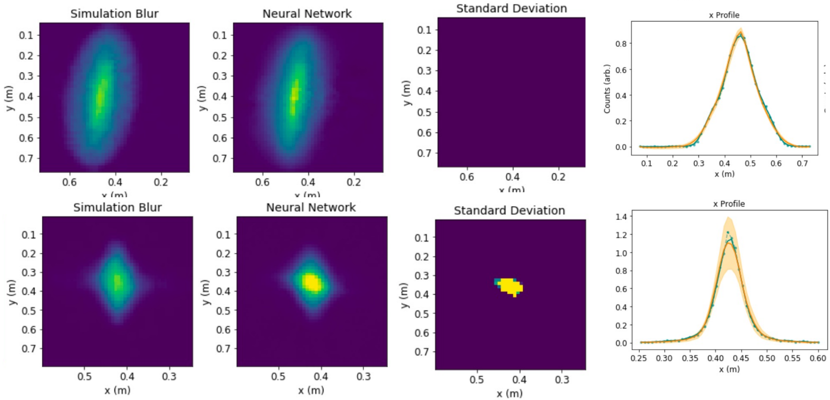
Scalar parameters for the LCLS-II injector (Bayesian neural network)

A. Mishra et. al., PRAB, 2021



longitudinal phase space (quantile regression + ensemble)

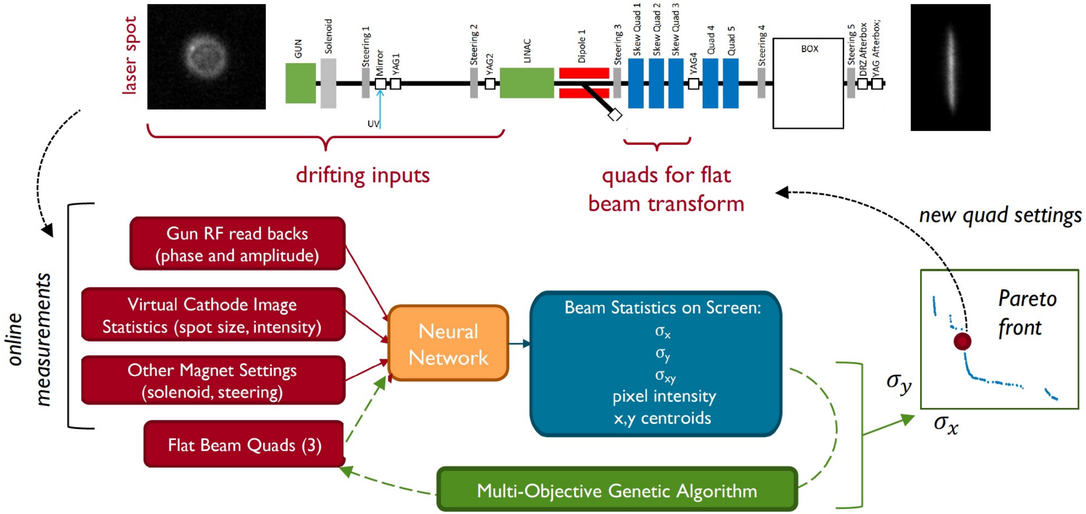
O. Convery, et al., PRAB, 2021



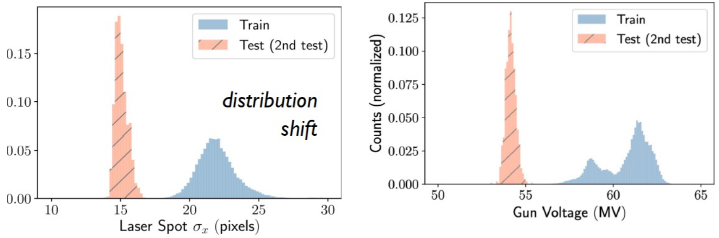
LCLS injector transverse phase space (ensemble)

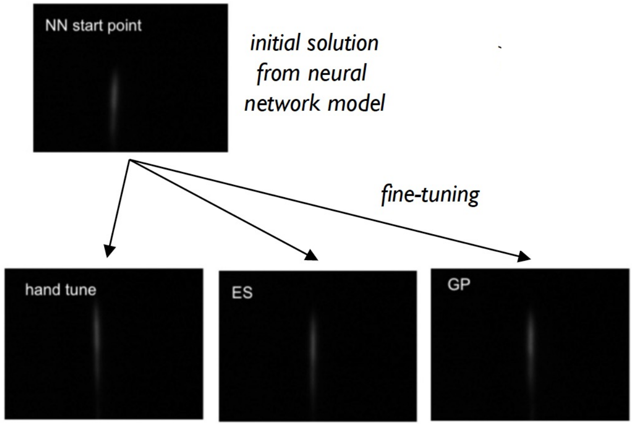
Example: Warm Starts from Online Models

E. Cropp et al., in preparation

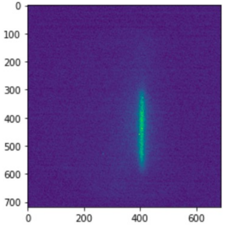


Can work even under distribution shift





- 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

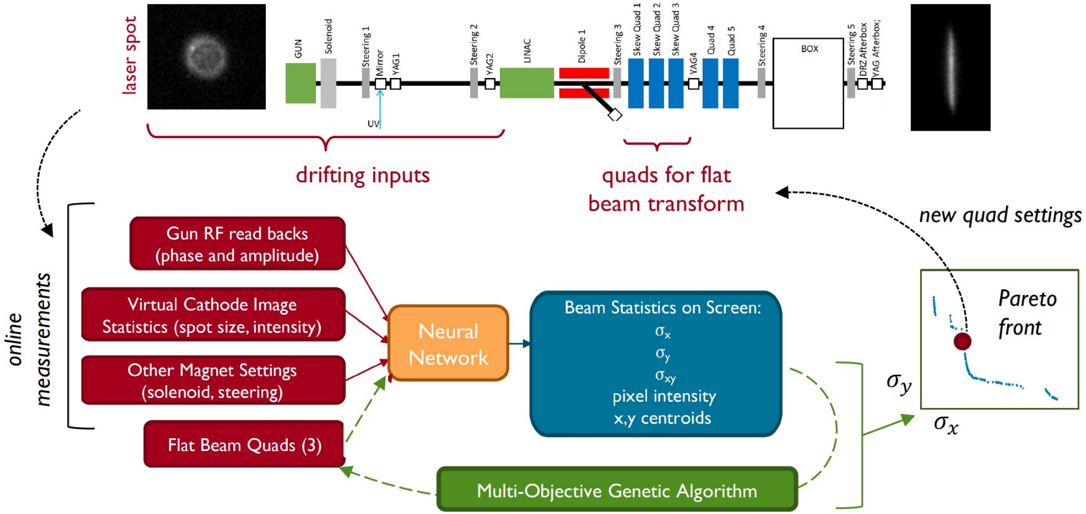


Hand-tuning in seconds vs. tens of minutes

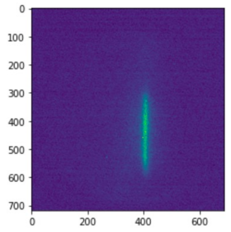
Boost in convergence speed for other algorithms

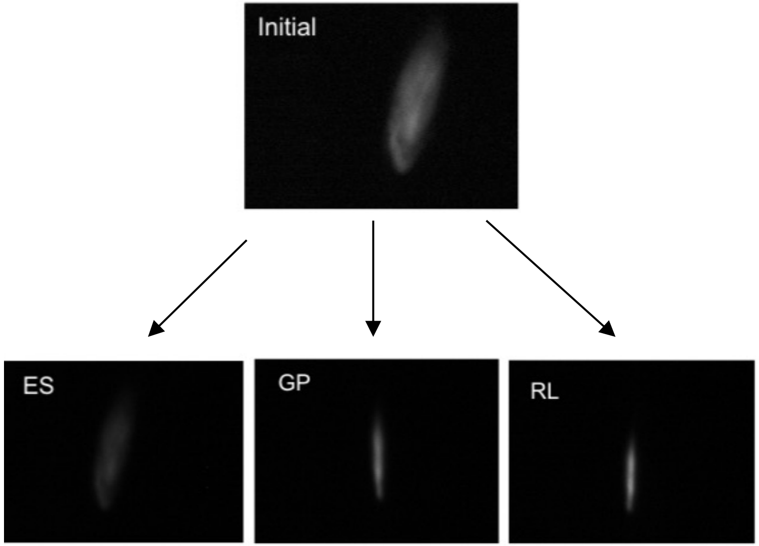
Example: Warm Starts from Online Models

E. Cropp et al., in preparation



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





RL was fastest to converge for cases examined → but did not get a chance to test comprehensively for different initial conditions

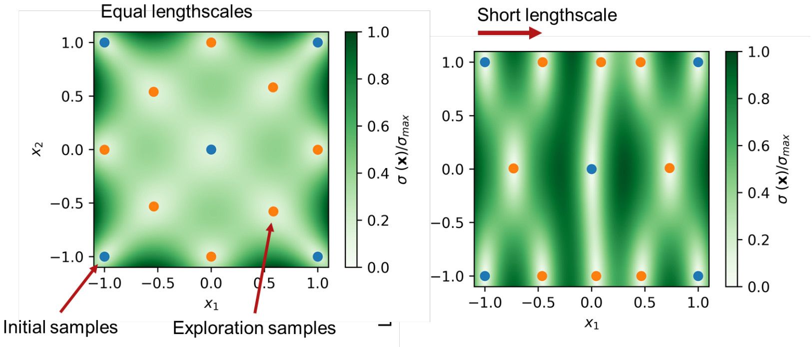
Efficient Characterization with Bayesian Exploration

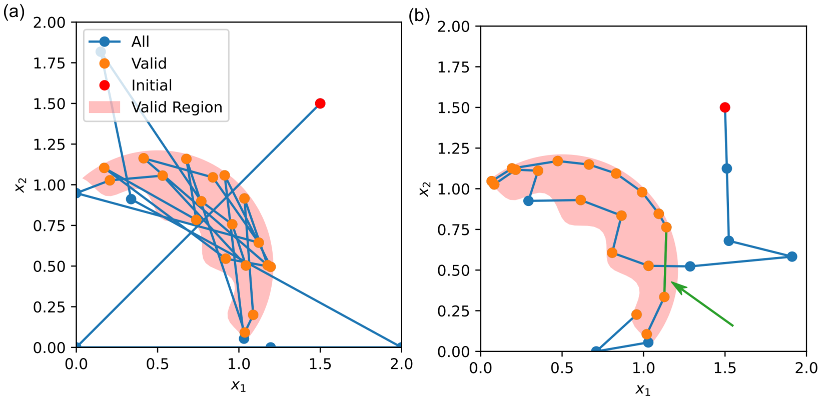
$$\alpha(\boldsymbol{x}) = \sigma(\boldsymbol{x}) \prod_{i=1}^{N} p_i(g_i(\boldsymbol{x}) \geq h_i)\, \Psi(\boldsymbol{x}, \boldsymbol{x_0})$$

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

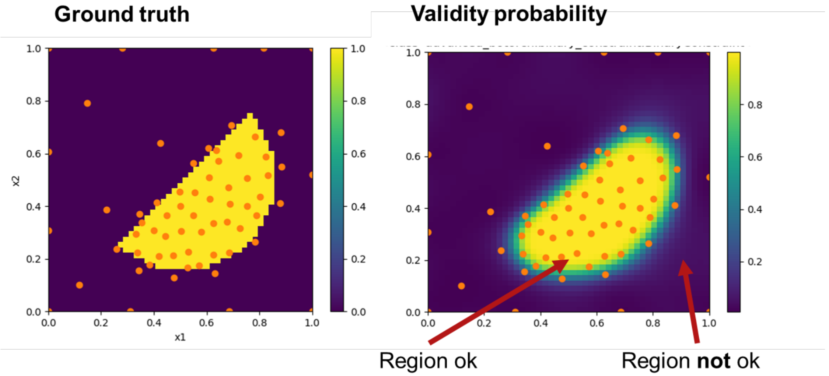
proximal biasing

adaptive sampling





learning constraints



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

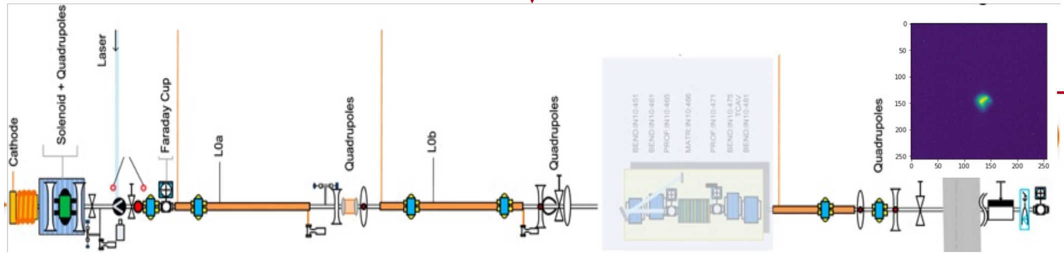
See R. Roussel's Bayesian Optimization tutorial from this workshop

Efficient Characterization of FACET-II Injector

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)

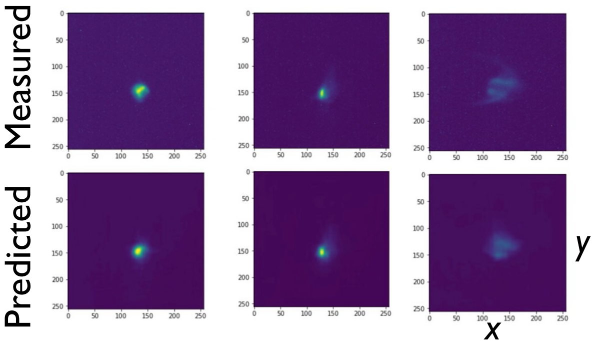


FACET-II Injector

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

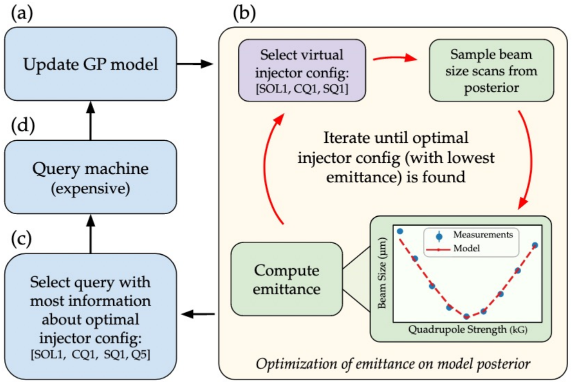


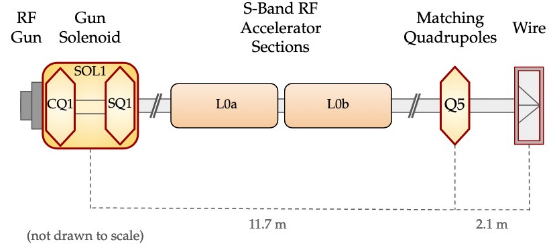
Use of Bayesian exploration to generate training data was sample-efficient, reduced burden of data cleaning, and resulted in a well-balanced distribution for the training data set over the input space. ML models were immediately useful for optimization.

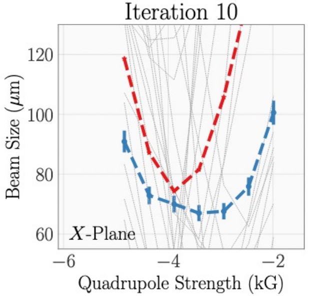
Efficient Emittance Optimization with Partial Measurements

See S. Miskovich's talk from yesterday

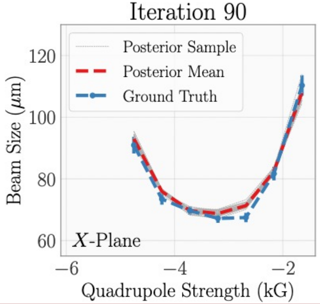
- **Instead of tuning on costly emittance measurements directly: learn a fast-executing model online for beam size while optimizing** → *learn on direct observables (e.g. beam size); do inferred "measurements" (e.g. emittance)*
- **New algorithmic paradigm leveraging "Bayesian Algorithm Execution" (BAX)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

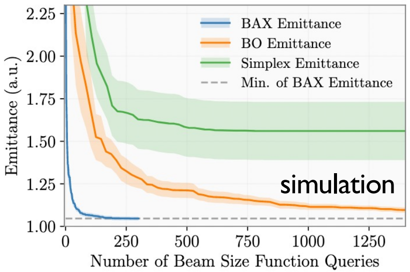


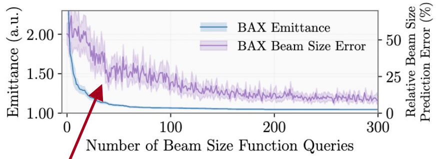




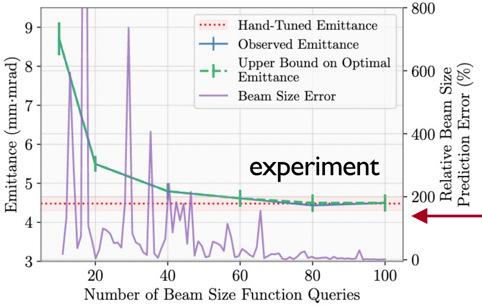
model is learned on-the-fly







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

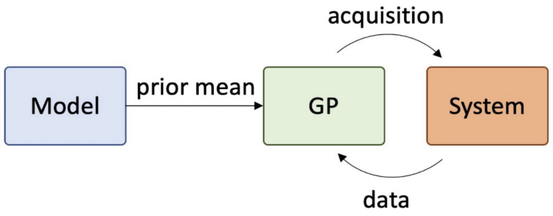
Paradigm shift in how tuning on indirectly computed beam measurements (such as emittance) is done, with 20x improvement over standard method for emittance tuning. → ***Now working to integrate into operations.***

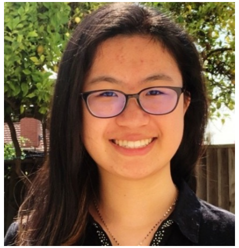
→ ***Also now working to incorporate more informative global models /priors rather than learning the model from scratch each time.***

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





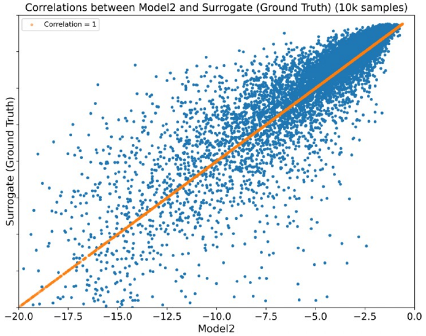
Summer '22 undergrad intern
Connie Xu

Used the LCLS injector surrogate model for prototyping

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

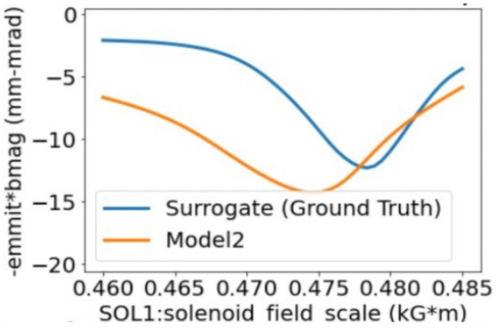
objective: *minimize emittance and matching parameter*

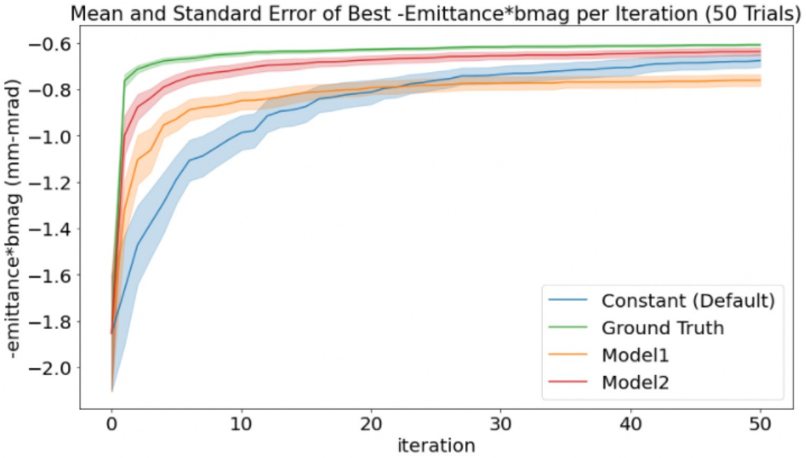
Correlations Between
Predictions and Ground Truth



Ground Truth

Model 2





Beta = 2.0

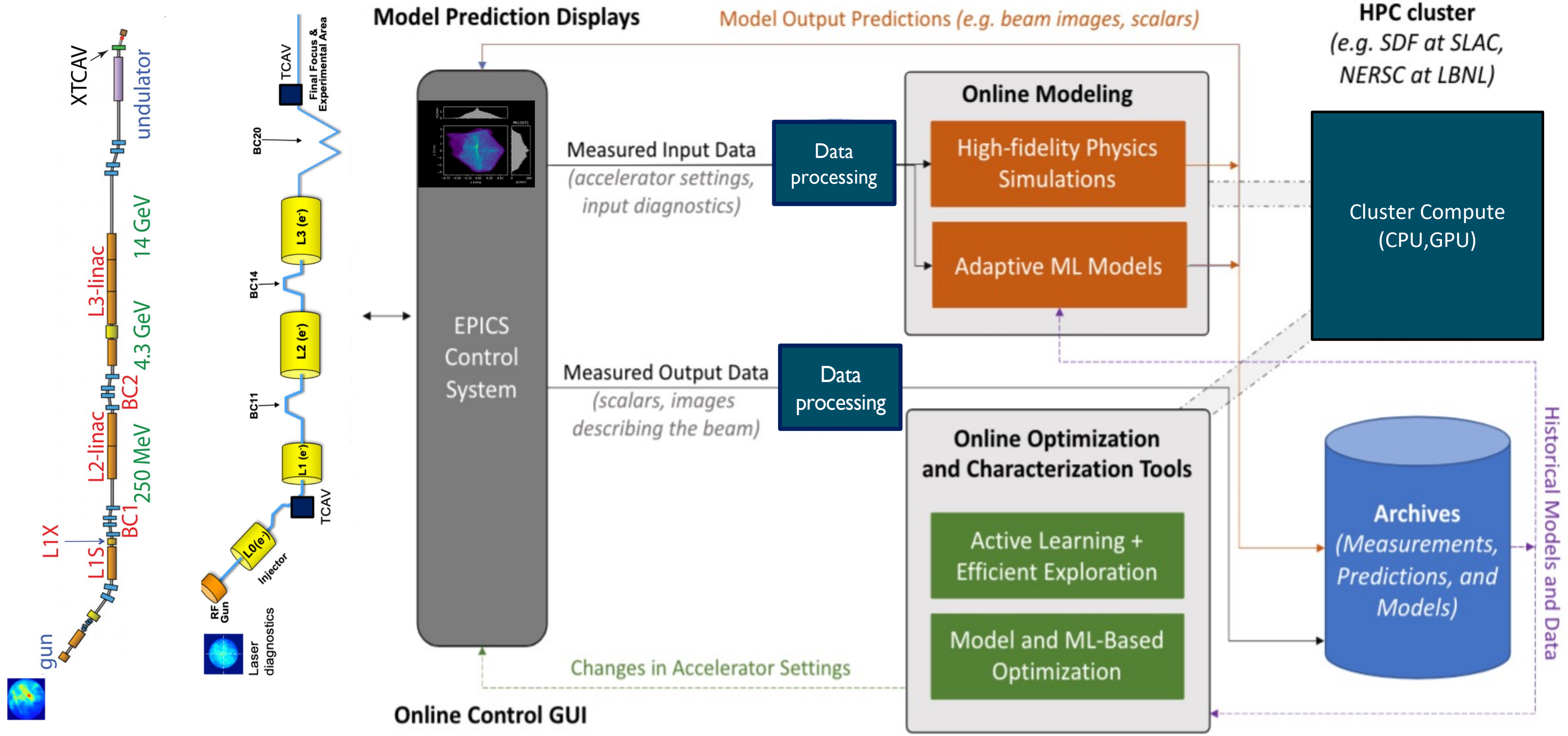
Even prior mean models with substantial inaccuracies provide a boost in initial convergence
→ now testing on machine and refining approach

Forthcoming paper at NeurIPS ML for Physical Sciences workshop

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

Want 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)*

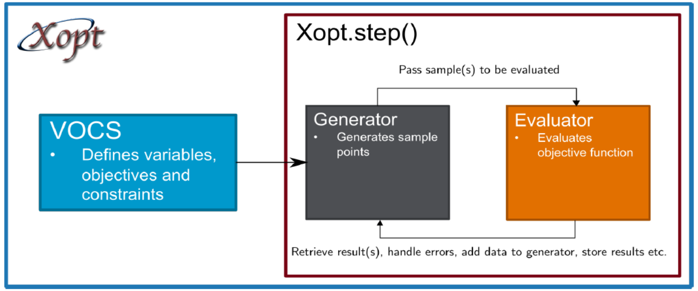


Making good progress toward this vision with open-source, modular software tools

Modular, Open-Source Software Development

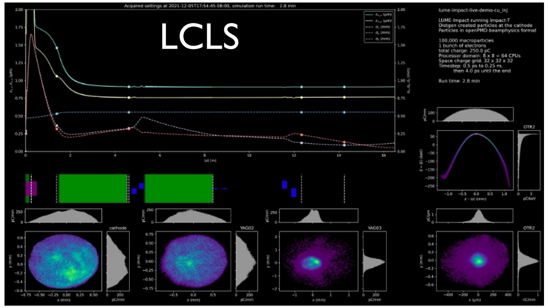
Community development of **re-usable, reliable, flexible software tools** for AI/ML workflows has been essential to maximize return on investment and ensure transferability between systems

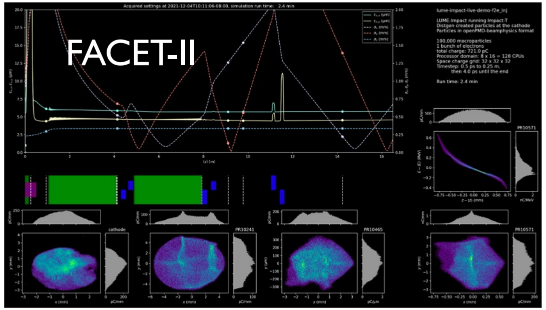
Modularity has been key: separating different parts of the workflow + using shared standards




```
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
```





Different software for different tasks:

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

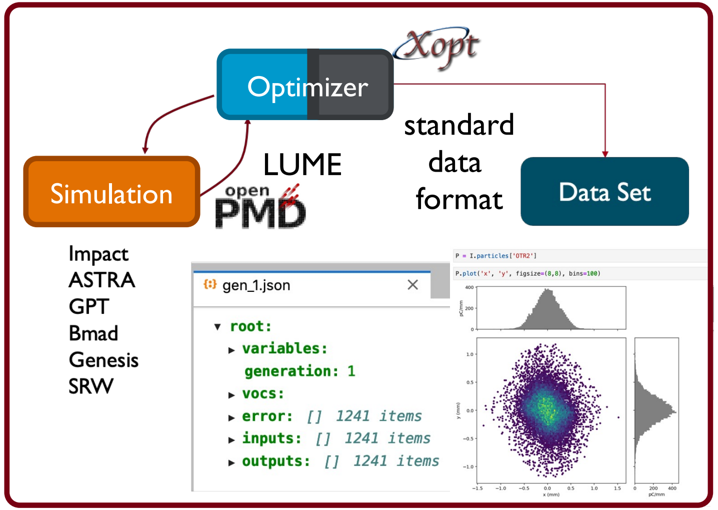
Visual control room interface *(e.g. Badger)*

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/



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. We welcome new users and contributors.

LUME-services: An online modeling service built on microservices

Provide continuously executing online models

- Slow-executing physics simulations
- Fast-executing ML surrogates

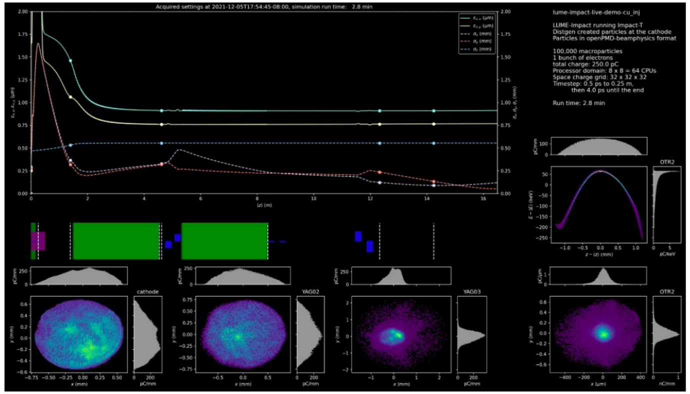
Generality of tooling

- Provide abstracted interfaces for model packaging
- Provide standardized set of services for composing applications

EPICS integration

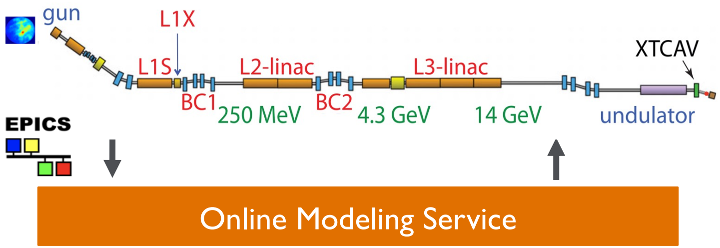
- Collect PV values over EPICS and queue simulations
- Serve model output over EPICS using programmatic IOC

Impact Dashboard:

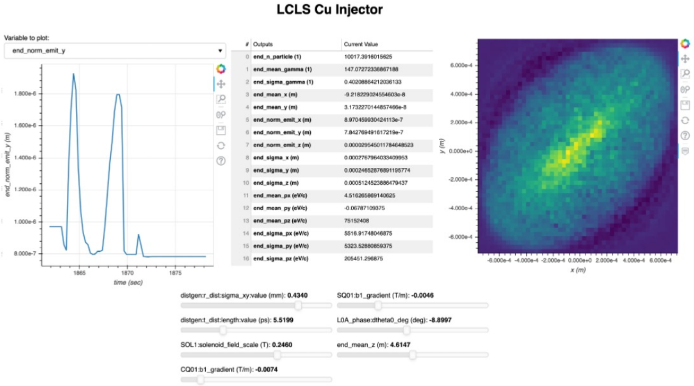


Example applications:

Particle data or screen images (e.g. laser profile) as input (distgen → Impact)
Advanced online visualization
Optimization using online model information (e.g. prior mean for Bayes opt)



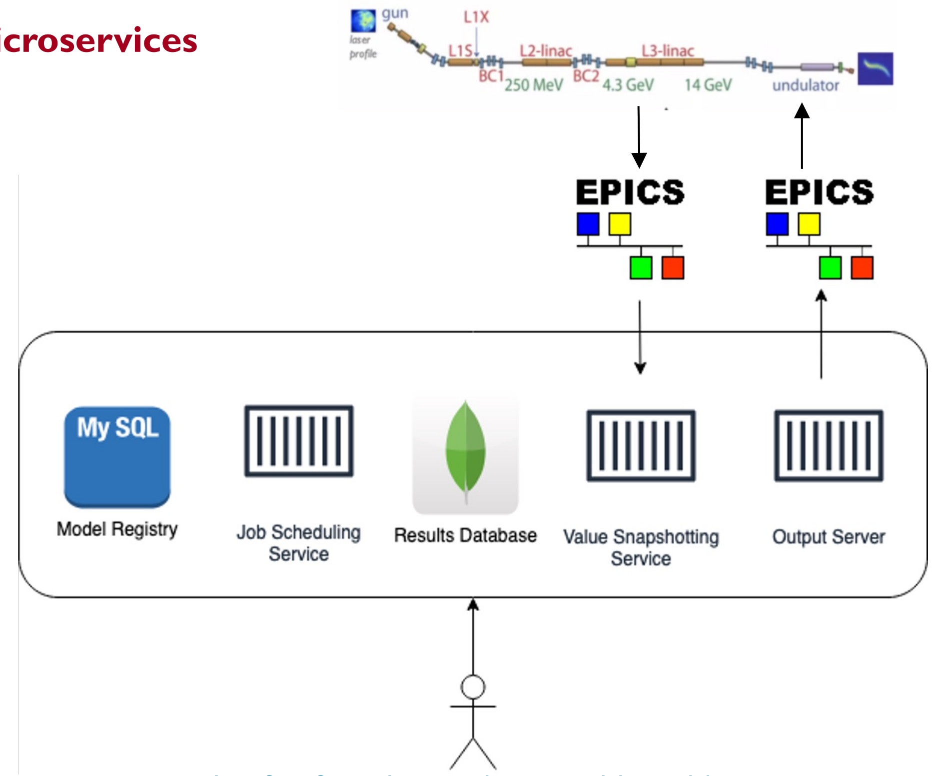
LCLS Injector UI w/ EPICS-based widgets (Using LUME-EPICS tools):



Have used at LCLS for linac/injector, FACET-II injector, LCLS-II injector → now want to interface with tuning (e.g. model info → Xopt)

LUME-services: An online modeling service built on microservices

- LUME-services is a Python package providing data APIs for inter-service interactions and user tooling
- Models are pip-installable Python packages and templates may be auto-generated using the LUME-services tools
- Models run in containers when a user schedules a workflow run
- The template provides Continuous Integration (CI) tools (e.g. GitHub actions) for users to use for testing and deployment
- Have demoed for a variety of physics sims and ML models at SLAC → now testing / improving for new cases (e.g. non-expert use)
- Have not yet integrated MLOps components (e.g. continuous/triggered automated model adaptation)
- Resources:
  - lume-services https://slaclab.github.io/lume-services/demo/
  - lume-model https://slaclab.github.io/lume-model/
  - lume-epics https://slaclab.github.io/lume-epics/
  - distgen https://github.com/ColwynGulliford/distgen



Interface for packaging arbitrary models, model registry

Enforcement of minimal metadata (model descript, owner, model type, PVs)

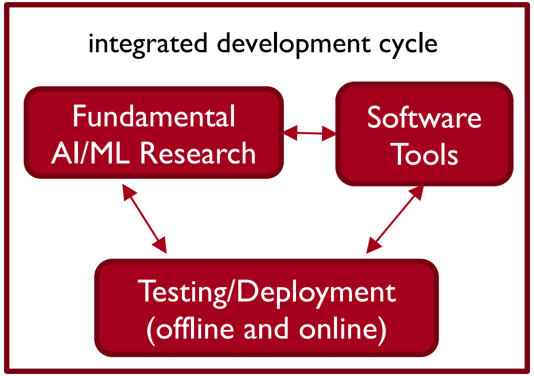
Ability to scale to arbitrary number of models and clients

Result storage + programmatic IOC for model results

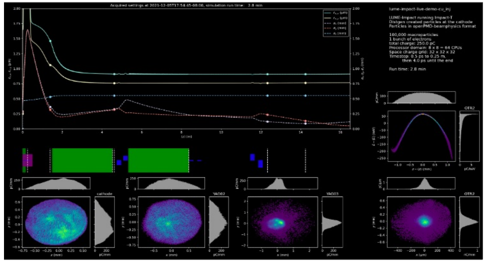
Essential infrastructure for reliable, continuous online model deployment and model version tracking / updating

Aimed for transferrable design between platforms → welcome collaborators and users

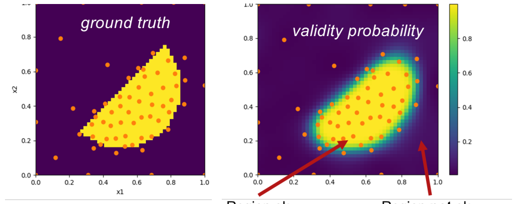
(1) Developing new approaches for accelerator optimization/characterization and faster higher-fidelity system modeling, (2) developing portable software tools to support AI/ML, (3) integrating these into regular use



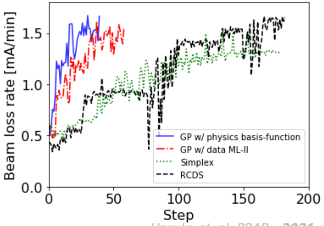
Online prediction with physics sims and **fast/accurate ML models**



Efficient optimization and characterization *(useful also for simulation exploration/design, data generation)*



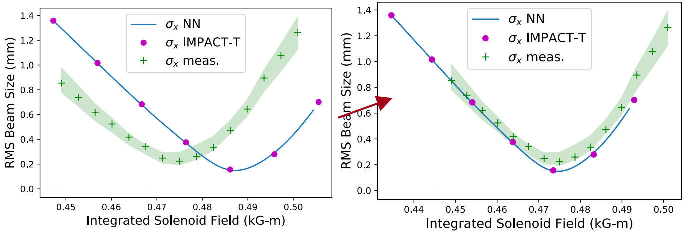
Roussel et. al. *Nat. Comm.* **2021**



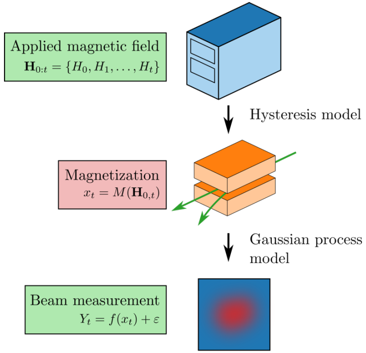
*Hanuka et. al. PRAB , **2021***

Output constraints learned on-the-fly

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

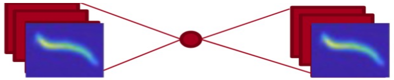


Techniques for **combining physics and ML** *(more reliable/transferrable, require less data, more interpretable)*, including **differentiable simulators**



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

Representation learning *(e.g. better ways of modeling beams)*



Software packages and standards for data generation, modeling, and optimization (*LUME, xopt, Badger*)

Summary

General strategy for comprehensive tuning at SLAC:

- Improve global models (accuracy, expressivity, speed, uncertainty estimates, adaptability)
- Develop algorithms for exploration and optimization of new parameter spaces
- Incorporate physics with ML modeling wherever useful ← See JP's talk and Ryan's talk later today
- Set up algorithms and software tools that link each of the above

Making lots of progress in these individual areas and **increasingly using combinations of approaches**

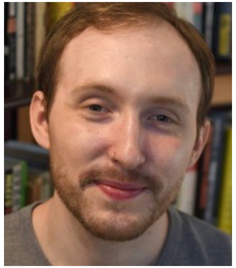
Some tools are **integrated into regular operations or are used regularly offline** (with more on the way)

Have been placing much emphasis on modular, interoperable software tools / standards → *tools have been used now for a variety of tasks at SLAC and AWA*









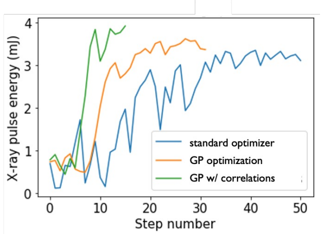


Want to join SLAC or collaborate with us?
We are actively hiring and eager to collaborate

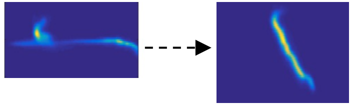
Broad Set of Areas for ML to Impact Operation

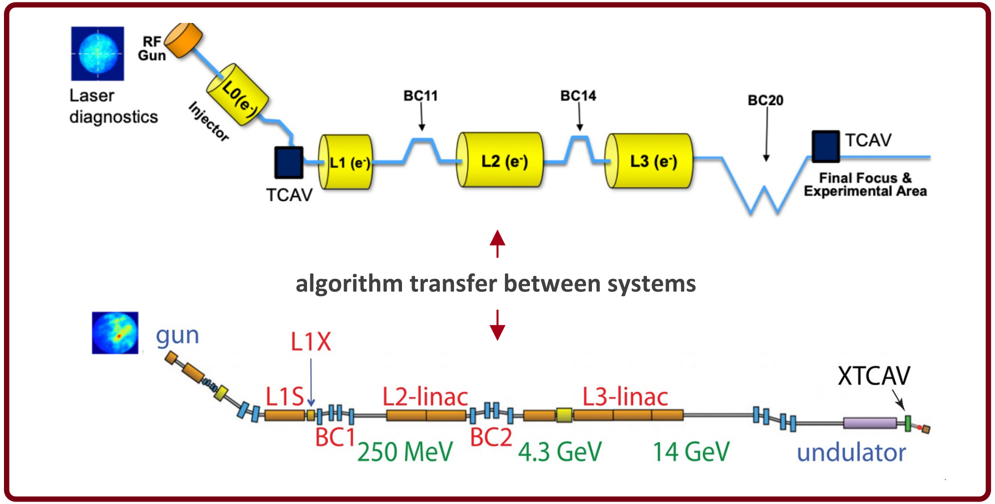
Data reduction/rejection *(kHz/MHz data streams)*
Event triggering

automated control + optimization

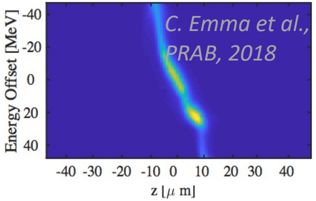


J. Duris et al., PRL, 2020

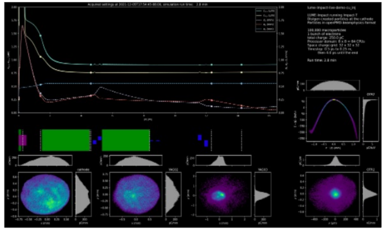


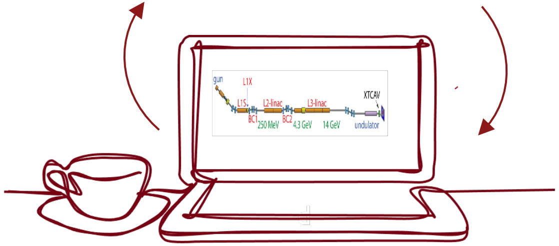


ML-enhanced diagnostics
(provide insight at faster rate, at higher resolution, non-invasively)



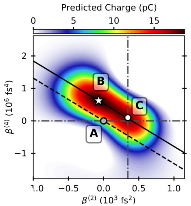
anomaly detection
failure prediction
(plan maintenance; alert to changes in machine; alert to interesting science)





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

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



R. Shaloo et al. [arXiv:2007.14340]

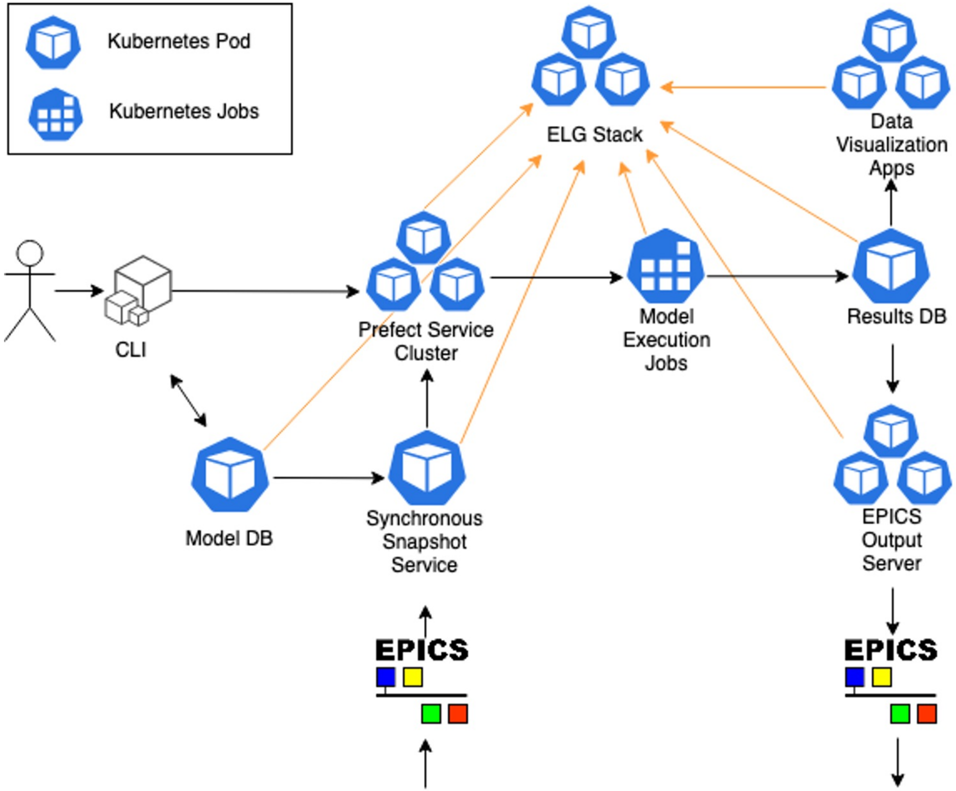
+ need uncertainty quantification for all
+ can incorporate physics information in all

Backup Slides

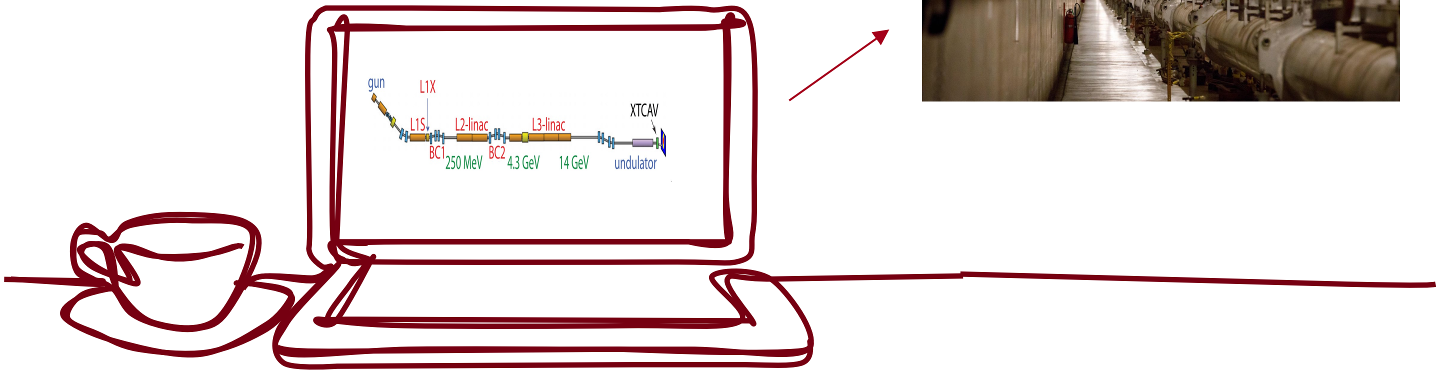
Component Architecture

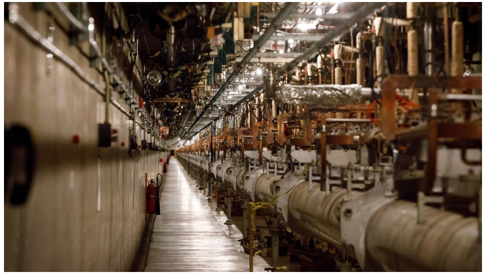
Components

High-level component	Function
Model DB	● Stores model metadata ● Tracks versioned deployments and associated workflows
Synchronous Snapshot Service	● Single pulse EPICS PV collection ● Submission of Prefect workflow runs
Prefect Service	● Orchestration of workflows ● Workflow monitoring ● Result management
Results DB	● Result storage
EPICS Output Server	● Monitors new entries to the results database ● Serves latest model output variables ● Responsible for uniqueness check ● Implement archiver integration
Data Visualization Apps	● Provide data visualization for model inputs/outputs
ELG Logging Stack	● Consolidation of in-cluster logs ● Cluster metrics in Grafana dash



In a perfect world…





Use a fast, accurate model …

find some knobs that give us the beam we want and apply those to the machine

get info about unobserved parts of machine (online model / virtual diagnostic)

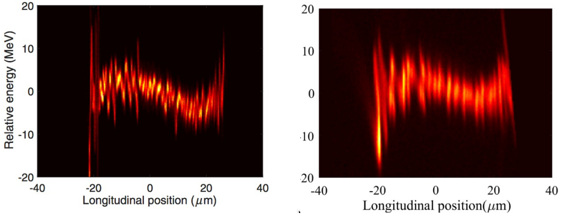
do offline planning and control algorithm prototyping

In reality things are much more difficult…

computationally expensive simulations

Simulation

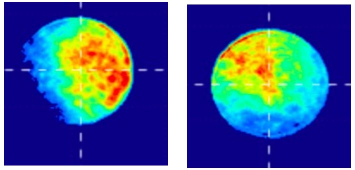
Measurement

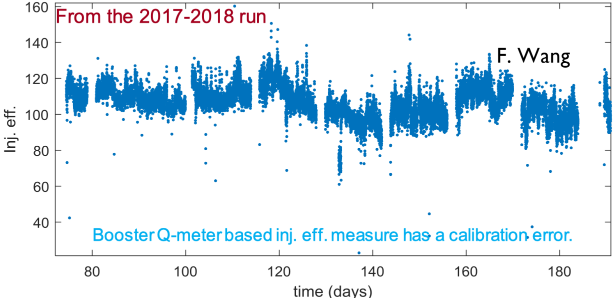


10 hours on thousands of cores at NERSC!

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

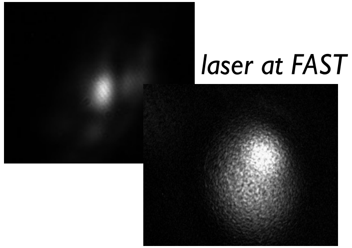
fluctuations/noise (e.g. laser spot)



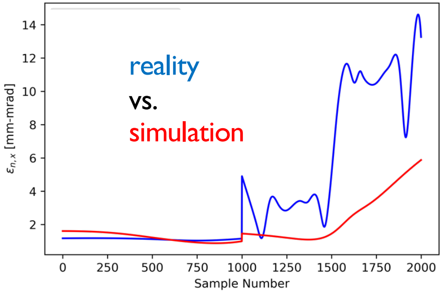


hidden variables / sensitivities

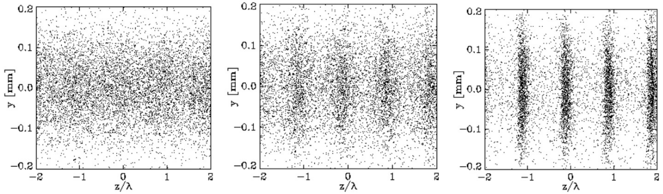
laser at FAST



drift over time



many small, compounding sources of uncertainty

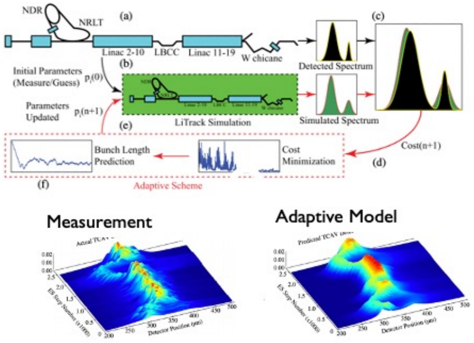


nonlinear effects / instabilities

Virtual Diagnostics

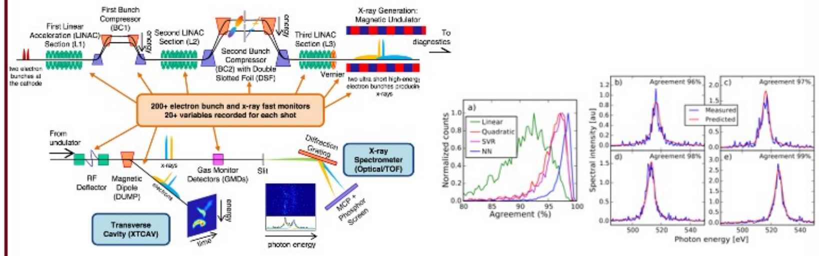
Provide information about parts of the system that are typically inaccessible (destructive, too slow, not directly measurable)

Adaptively tune a simple physics model



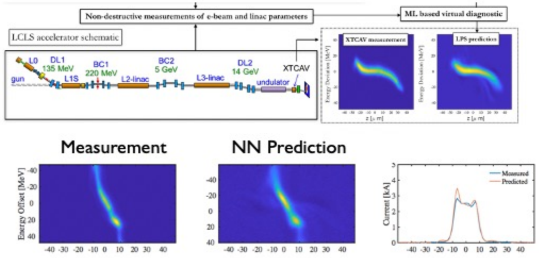
A. Scheinker, S.Gessner, PRAB 18, 102801 (2015)

Fill in shots: use archive data to learn correlation between fast and slow diagnostics



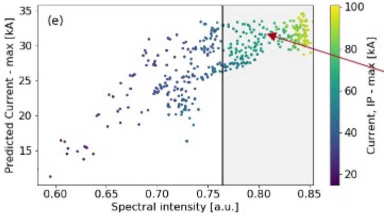
A. Sanchez-Gonzalez, et al., Nature Comms (2017)

Predict with a trained neural network



C. Emma, A. Edelen, et al., PRAB21, 112802 (2018)

Can use spectral information as input to predict beyond typical diagnostic resolution



Shots are beyond the TCAV resolution

A. Hanuka, et al. 2009.12835 [accepted to Nature Scientific Reports]

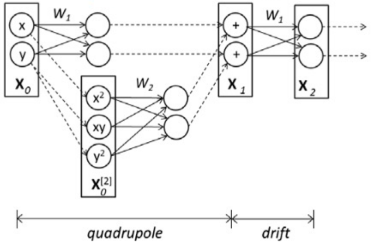
“Physics-informed” modeling → *incorporate physics domain knowledge to reduce need for data, and aid interpretability + generalization*

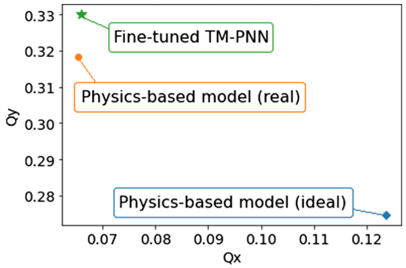
Many approaches:

- Combine physics representations and machine learning models directly *(e.g. differentiable simulations)*
- Add physics constraints to output metrics
- Force to satisfy expected symmetries *(e.g. inductive biases in ML model)*
- Loose form: learn from many physics sims in a way that results in good representation of the physics *(also related to representation learning)*

Differentiable Taylor map physics model + weights → train like ML model needed very little data to calibrate PETRA IV model

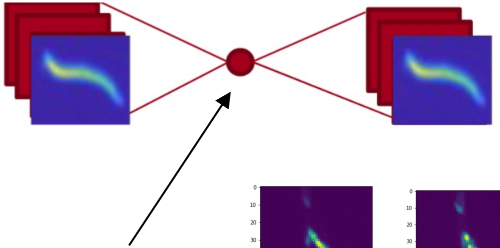
Ivanov et al, PRAB, 2020



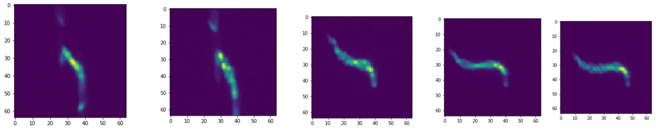


Physics-driven representation learning

(e.g. encoder-decoder neural network models)



useful latent space



Many examples in our field!

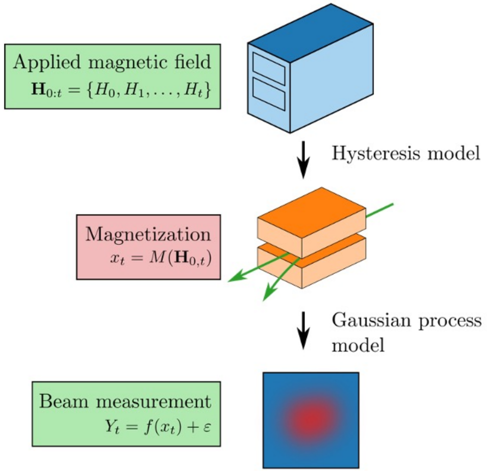
Review paper: Karniadakis et al, *Nat Rev Phys* **3,** 422–440 (2021)

Snowmass accelerator modeling white paper: arXiv:2203.08335

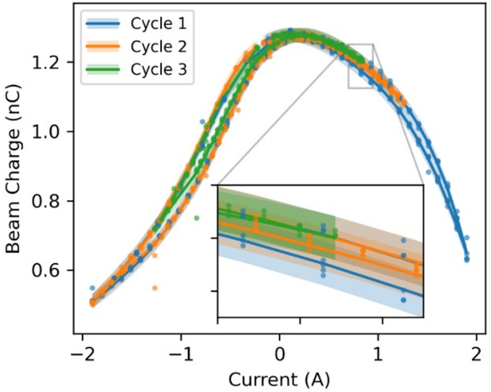
Differentiable Physics Simulations and ML

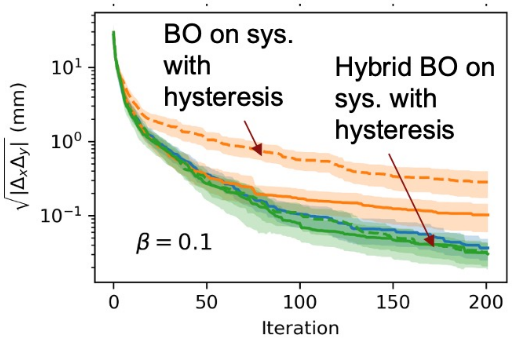
Modern ML uses gradients in learning → *differentiable physics sims enable modular combinations with ML components, analyses, etc.*

Fundamentally new approach in combining physics models, data, and ML



Differentiable physics model of hysteresis combined with ML enables in situ characterization of magnetic hysteresis in accelerator magnets and higher-precision optimization



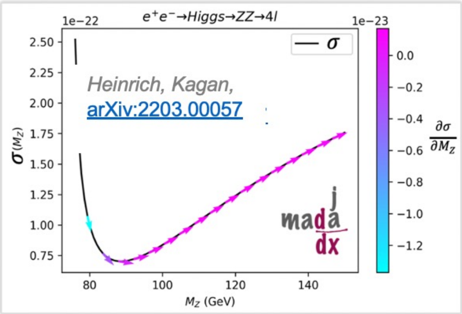


R. Roussel, et al., PRL, 2022, arXiv:2202.07747

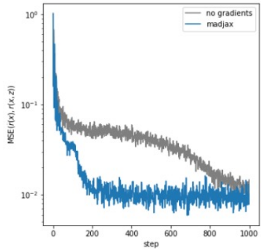
arXiv:2203.13818

Toward the End-to-End Optimization of Particle Physics Instruments with Differentiable Programming: a White Paper

Differentiable physics models can facilitate instrument-wide optimization, from accelerator to detector to physics analysis



Differentiable matrix elements of high energy scattering processes



ML-Assisted Optimization and Characterization

Large, nonlinear, and sometimes noisy search spaces for accelerators and detectors → need to find optima and examine trade-offs with limited budget *(computational resources, machine time)*

ML-assisted optimization **leverages learned representations** to improve sample efficiency. Some methods also include **uncertainty estimation** to inform where to sample next *(avoid undesirable regions, target information-rich areas).*

Similar set of tools for operation and design *(with a few differences: parallel vs. serial acquisition, need for uncertainty-aware/safe optimization)*

Bayesian optimization / active learning / reinforcement learning

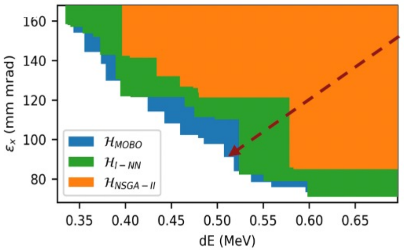
→ All learn iteratively via online interaction with the system

next point to search

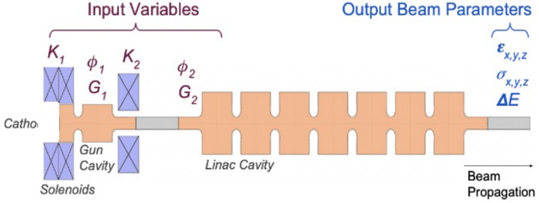
Learning Algorithm (GP model + Bayes. opt., iterative surrogate, RL agent)

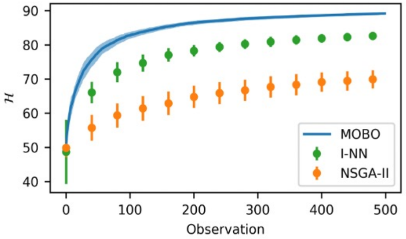
System (Simulation or Online Interaction)

data



Pareto front: optimal tradeoff between parameters of interest

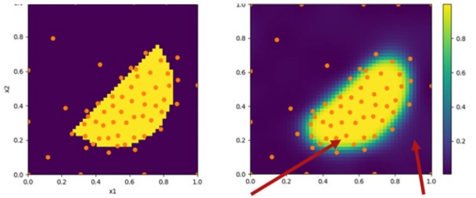




Faster multi-objective optimization with Bayesian optimization and iterated surrogate models

R. Roussel et al., [arXiv:2010.09824]

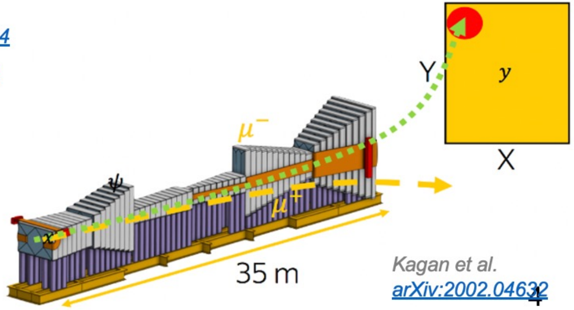
A. Edelen et al., [arXiv:1903.07759]



Output constraints learned on-the-fly

R. Roussel et al., [arXiv:2106.09202]

Local generative surrogates and gradient descent for the SHiP magnetic shield design



Kagan et al. [arXiv:2002.04632]