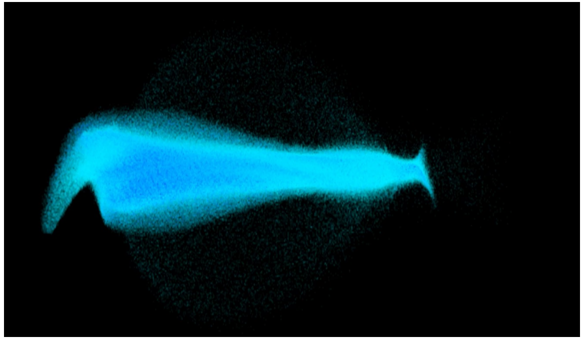
EPICS Workshop
April 26, 2023

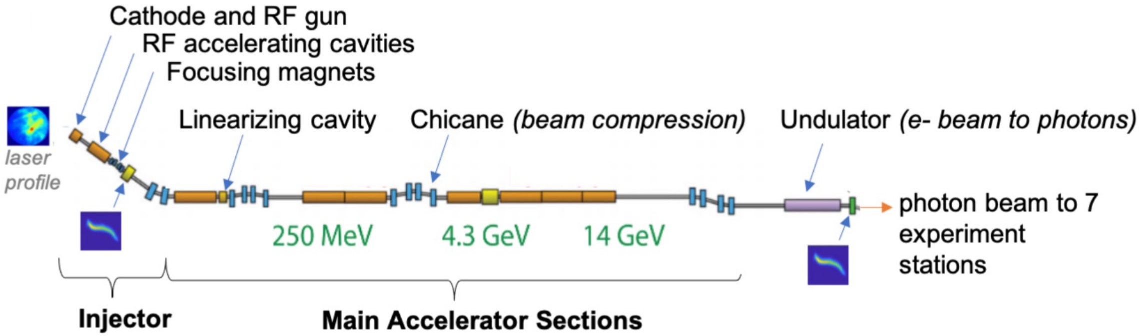
Experience with Machine Learning Enhanced Modeling and Optimization at SLAC

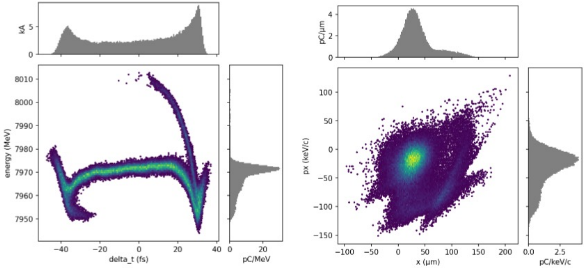
Auralee Edelen
edelen@slac.stanford.edu

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

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





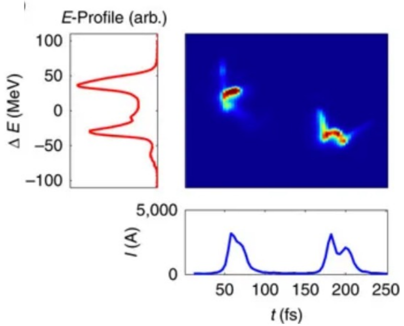


Beam exists in 6-D position-momentum phase space

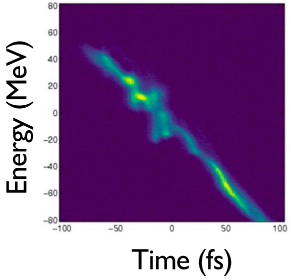
Have incomplete information: measure 2-D projections or reconstruct based on perturbations of upstream controls (e.g. tomography, quad scans)

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

Nonlinear, high-dimensional optimization problem

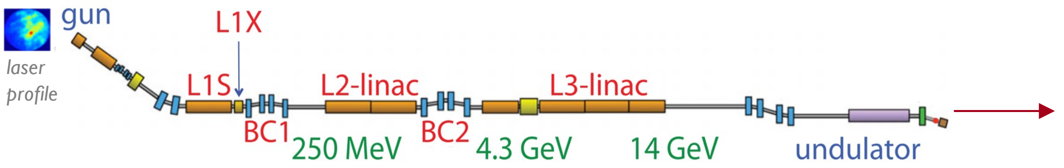


A. Marinelli, et al., Nat. Commun. 6, 6369 (2015)



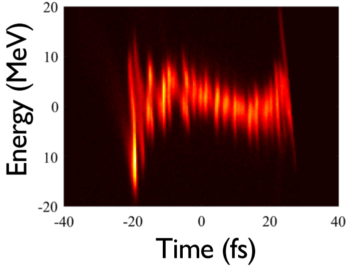
A. Marinelli, IPAC'18

wide spectrum of tuning needs

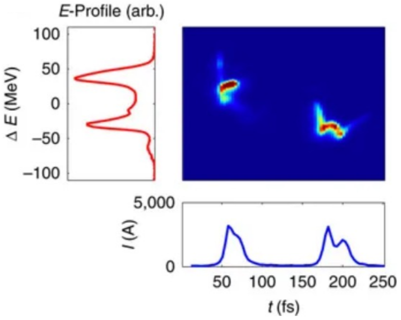


photon beam to 7 experiment stations

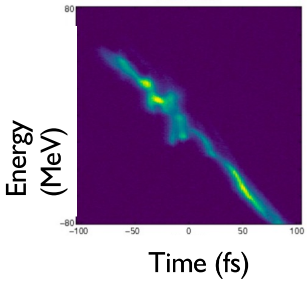
J. Qiang et al, PRAB (2017)



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



A. Marinelli, IPAC'18

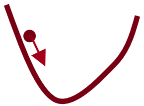


Rapid beam customization	Achieve new configurations + unprecedented beam parameters	Fine control to maintain stability within tolerances

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

less ← assumed knowledge of machine → more

Model-Free Optimization

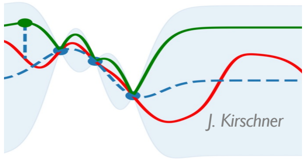


Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

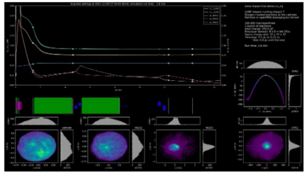


Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



Make fast system model

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

ML system models +
inverse models

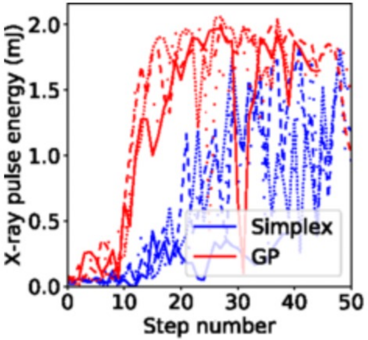
Tuning research at SLAC is aimed at combining the strengths of different approaches.

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.

Many successes with Bayesian Optimization

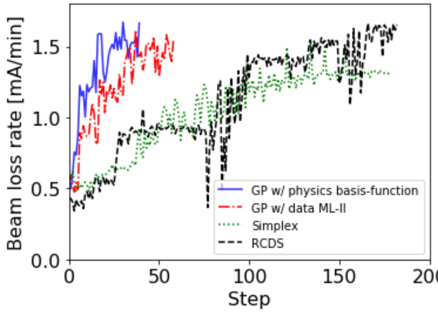
(+ improvements)

FEL pulse energy tuning at LCLS



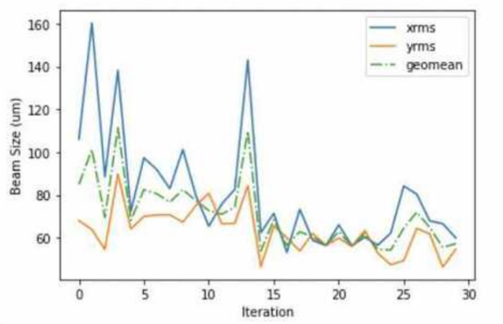
Duris et. al. PRL , **2020**

Loss rate tuning at SPEAR3

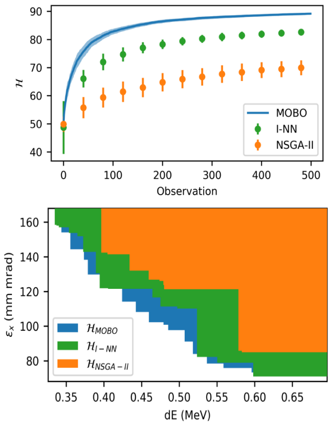


Hanuka et. al. PRAB , **2021**

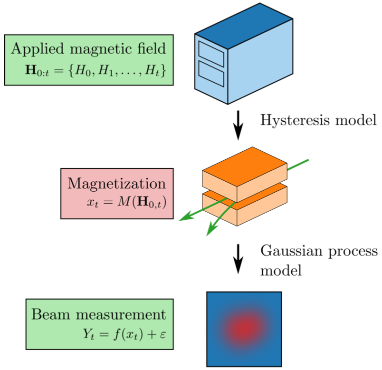
Sextupole tuning for IP at FACET-II



Multi-objective Bayesian Optimization

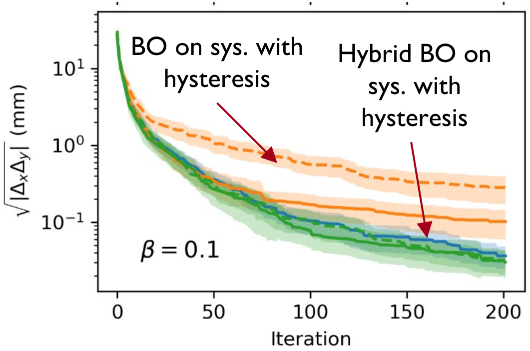


Roussel et. al. PRAB , **2021**

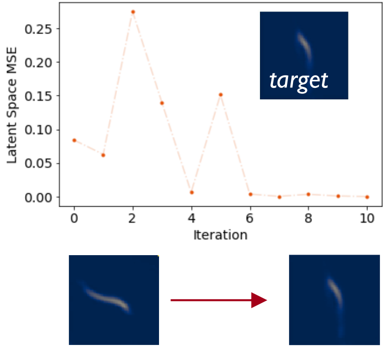


Roussel et. al. PRL , **2022**

Higher-precision optimization possible when including hysteresis effects in model



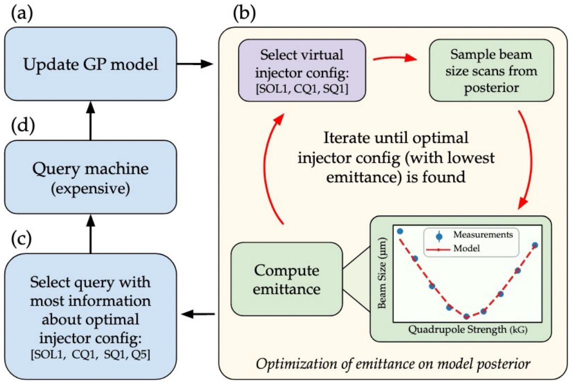
Longitudinal phase space tuning on LCLS

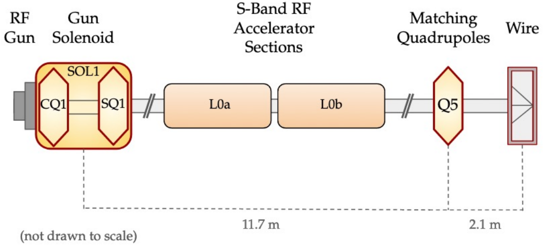


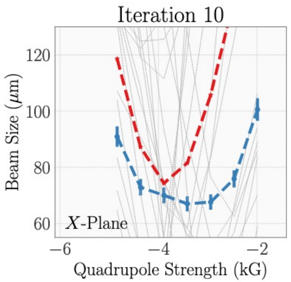
Algorithms being implemented/distributed in Xopt: https://github.com/ChristopherMayes/Xopt

Efficient Emittance Optimization with Partial Measurements

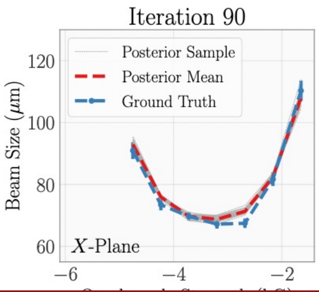
- **Instead of tuning on costly emittance measurements directly: learn a fast-executing model online for beam size while optimizing** → *learn on direct observables (e.g. beam size); do inferred "measurements" (e.g. emittance)*
- **New algorithmic paradigm leveraging "Bayesian Algorithm Execution" (BAX)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

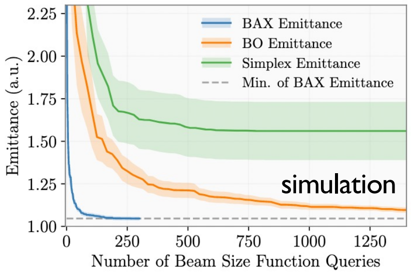


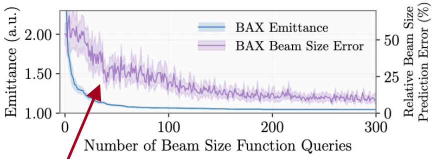




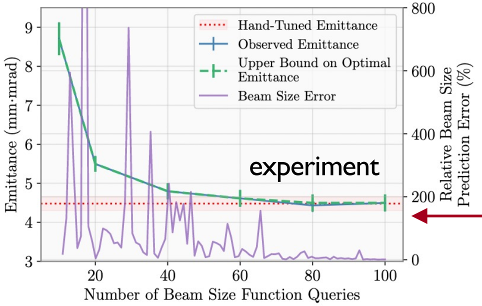
model is learned on-the-fly







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

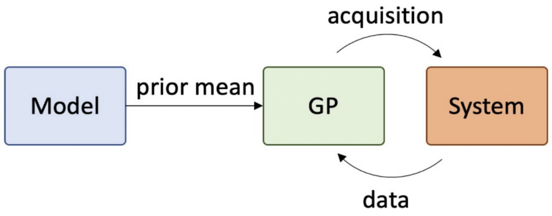
Paradigm shift in how tuning on indirectly computed beam measurements (such as emittance) is done, with 20x improvement over standard method for emittance tuning. → ***Now working to integrate into operations.***

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

Neural Network System Models + Bayesian Optimization

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

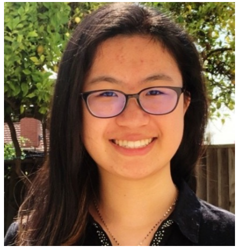
Good first step from previous work: use neural network system model to provide a prior mean for a GP



Used the LCLS injector surrogate model for prototyping

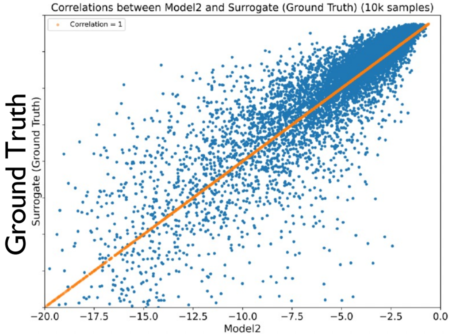
variables: *solenoid, 2 corrector quads, 6 matching quads*

objective: *minimize emittance and matching parameter*

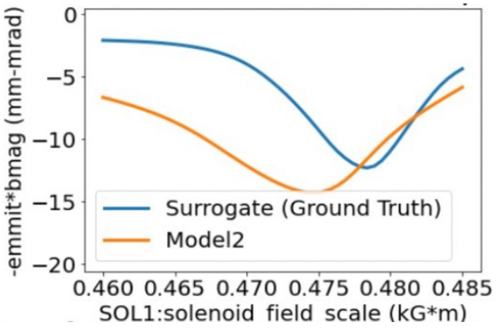


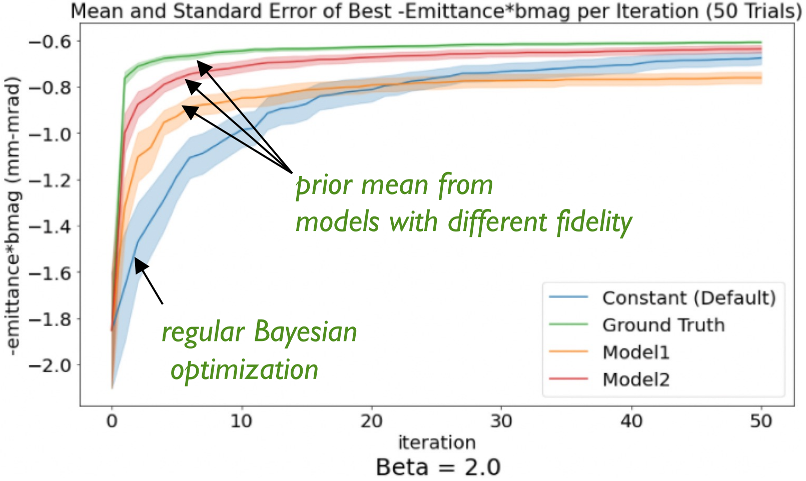
Summer '22 undergrad intern
Connie Xu

Correlations Between Predictions and Ground Truth



Model 2





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

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

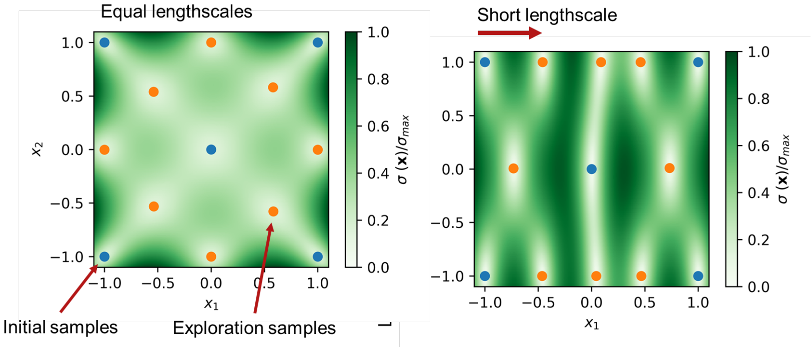
Efficient Characterization with Bayesian Exploration

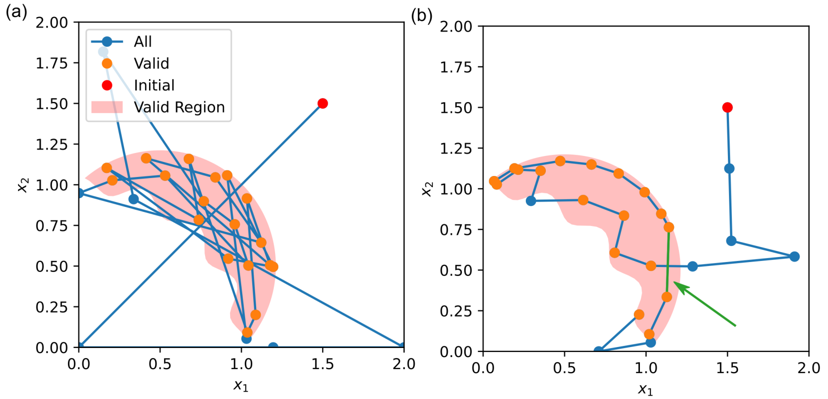
R. Roussel et. al.
Nat. Comm. **2021**

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

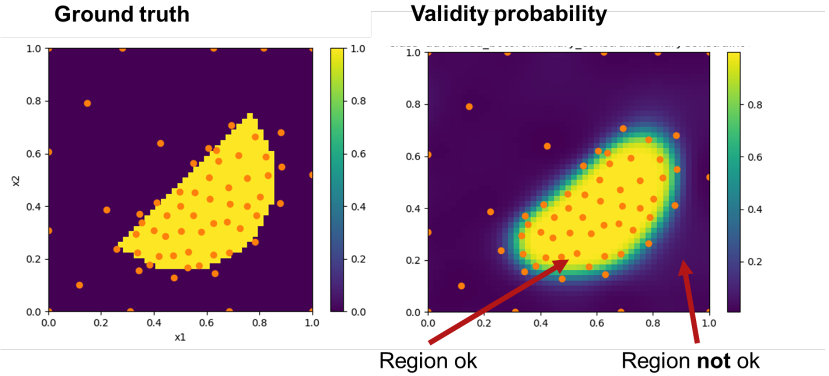
proximal biasing

adaptive sampling



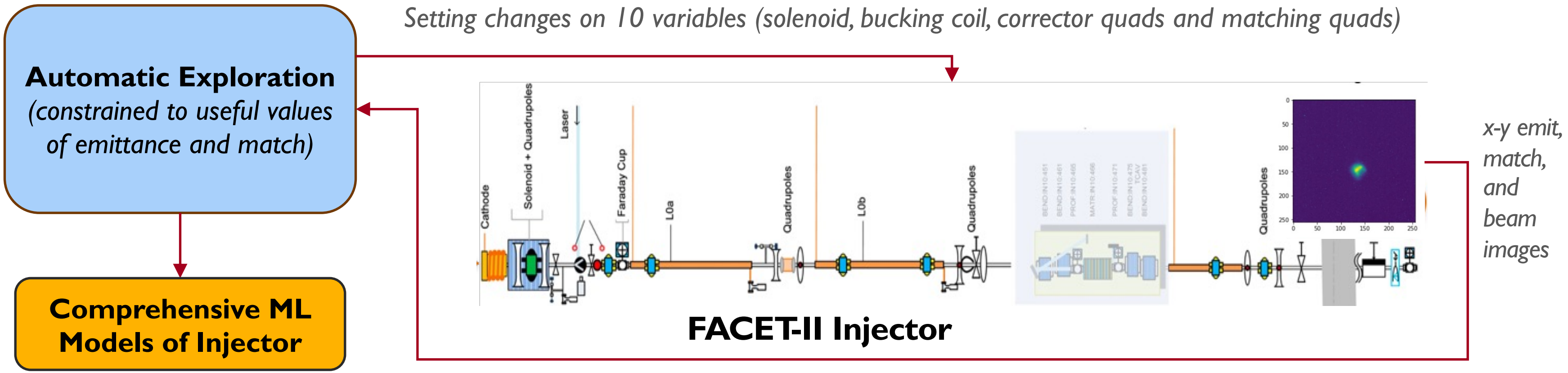


learning constraints



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

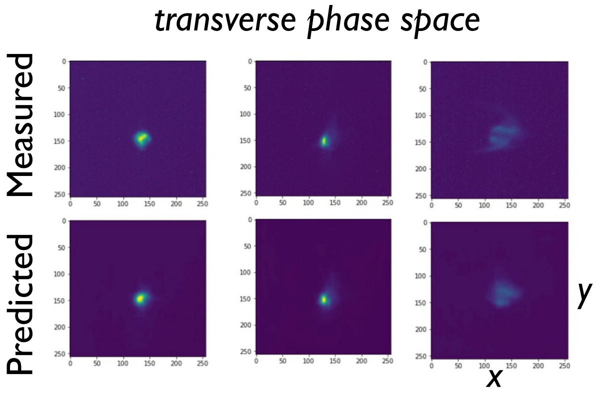
Efficient Characterization of FACET-II Injector



- 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.

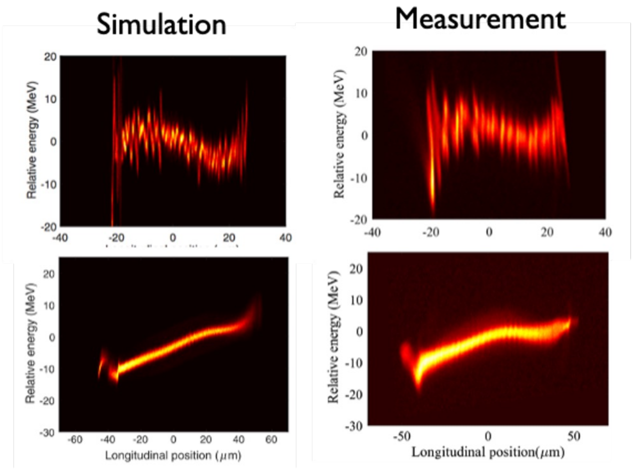
- **Example of integrated cycle between characterization, modeling, and optimization → now want to extend to larger system sections and new setups**



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.

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

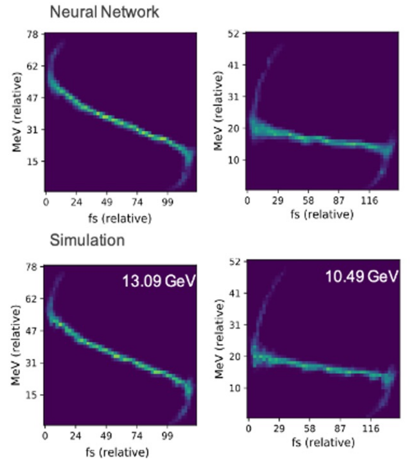
10 hours on thousands of cores at NERSC!

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

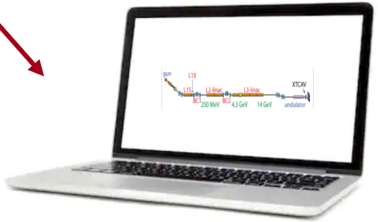
Edelen et al., NeurIPS 2019

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

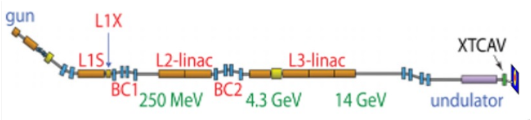
Bringing simulation tools from HPC systems to online/local compute

Online prediction
Model-based control



Control prototyping
Experiment planning

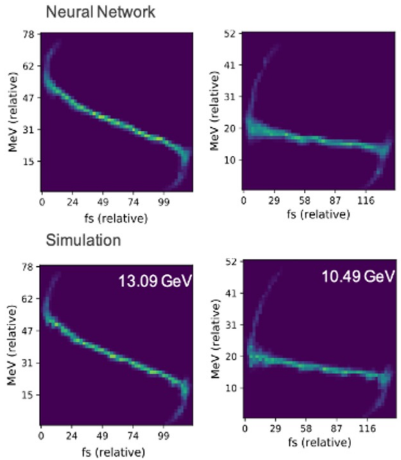
ML models are able to provide fast approximations to simulations (“surrogate models”)



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



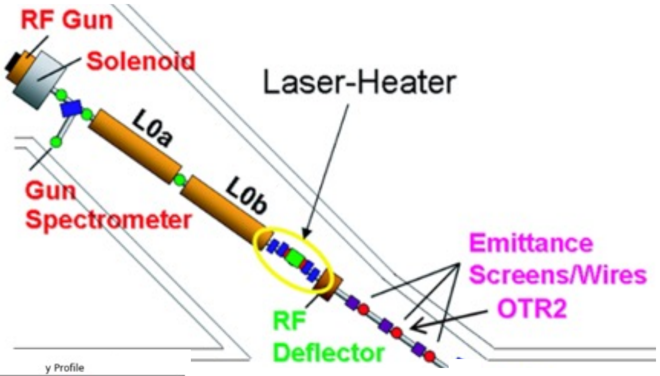
< ms execution speed

10^6 times speedup

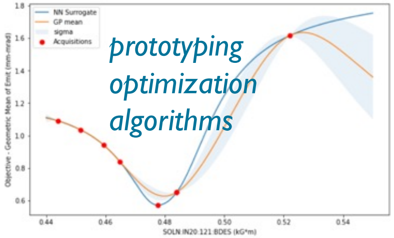
Edelen et al., NeurIPS 2019

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

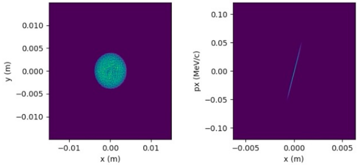
In Regular Use: Injector Surrogate Model at LCLS



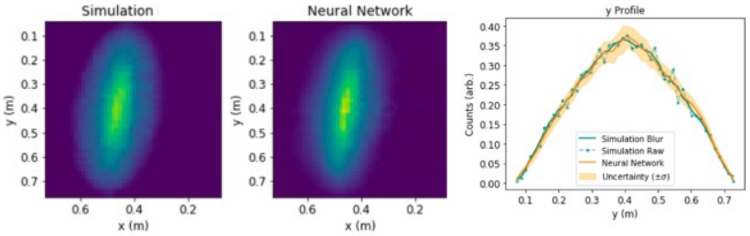
- ML models trained on detailed physics simulations with nonlinear collective effects
- Accurate over a wide range of settings → calibrate to match machine measurements
- **Used to develop/prototype new algorithms before testing online**
  (e.g. BAX w/ 20x speedup in emittance tuning https://arxiv.org/abs/2209.04587)
- Will provide initial Twiss parameters for downstream online model for optics matching
- Working on integrating model information to further speed up optimization algorithms



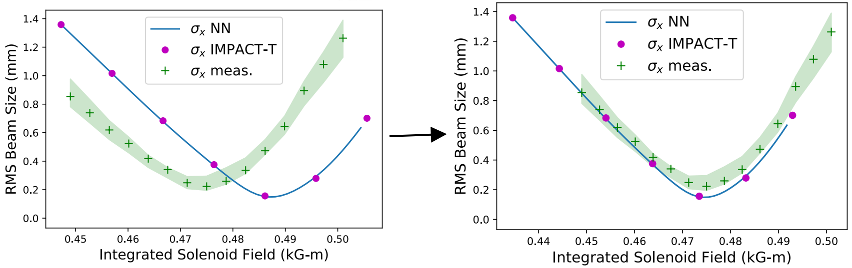
prototyping optimization algorithms



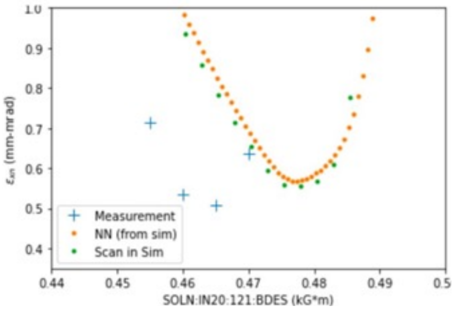
interactive model widget and visualization tools



ML model matches simulation under interpolation



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

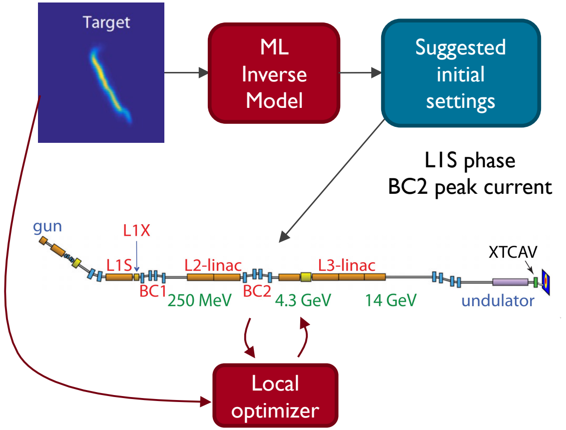


Simulation and ML model trained on it are qualitatively similar to measurements under interpolation (setting combinations reasonable distance from training set)

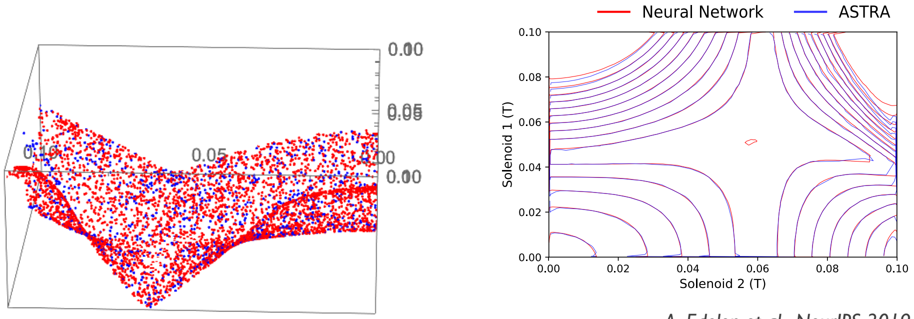
ML models trained on simulations and measurements have enabled fast prototyping of new optimization algorithms, facilitated rapid model adaptation under new conditions, and can directly aid online tuning and operator decision making

Warm starts for optimization

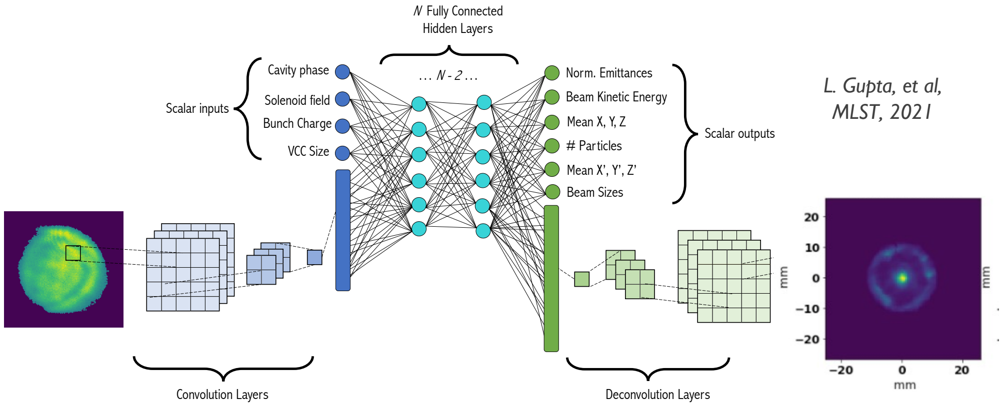
A. Scheinker, A. Edelen, et al, PRL, 2018



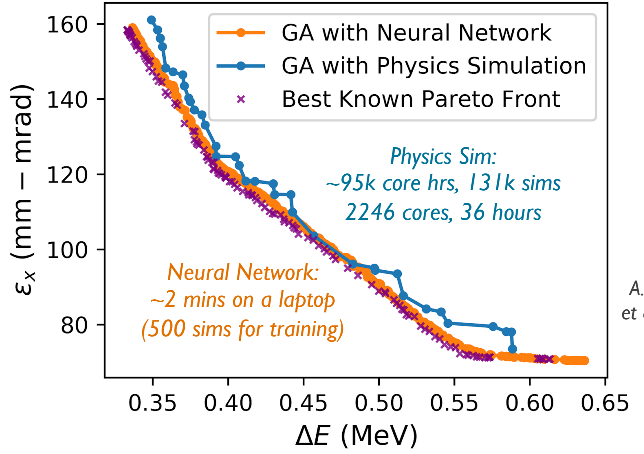
Smooth interpolation
Example σ_x surface from 2D scan, LCLS-II Injector



A. Edelen et al., NeurIPS 2019



L. Gupta, et al, MLST, 2021



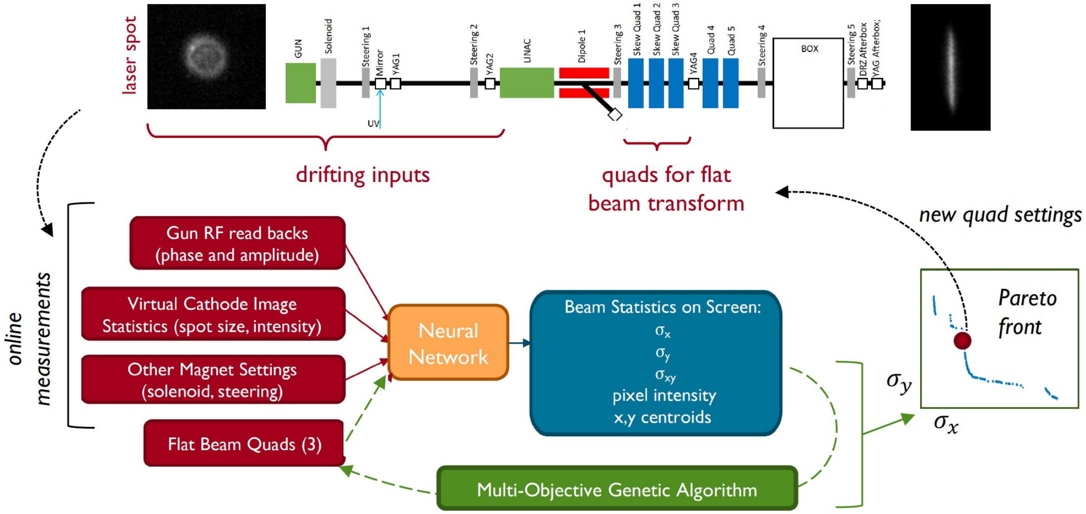
A. Edelen et al., PRAB, 2020

Include high-dimensional input information → better output predictions

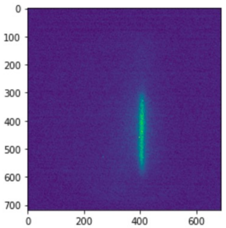
Surrogate-boosted design optimization

Example: Warm Starts from Online Models

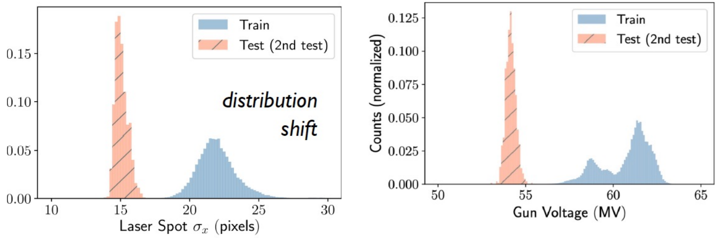
E. Cropp et al., in preparation

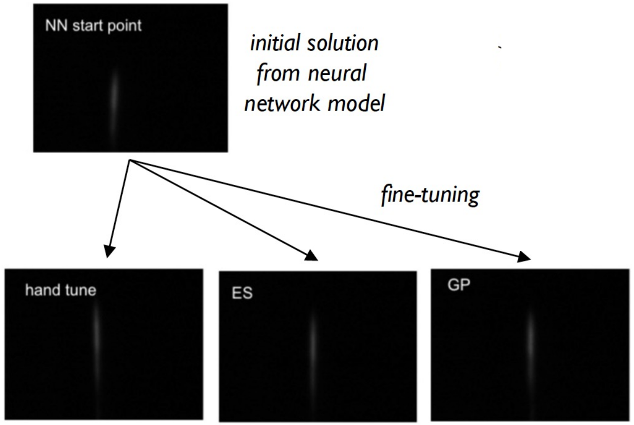


- Round-to-flat beam transforms are challenging to optimize → 2019 study explored ability of a learned model to help
- Trained neural network model to predict fits to beam image, based on archived data
- Tested online multi-objective optimization over model (3 quad settings) given present readings of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



Can work even under distribution shift





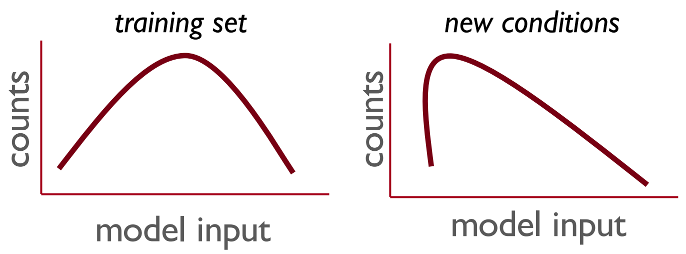
Hand-tuning in seconds vs. tens of minutes

Boost in convergence speed for other algorithms

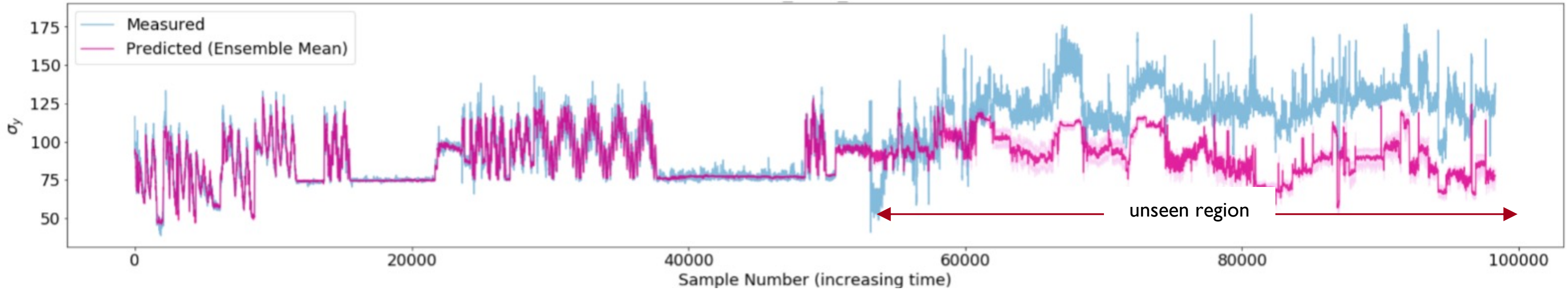
Uncertainty Quantification / Robust Modeling / Model Adaptation

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

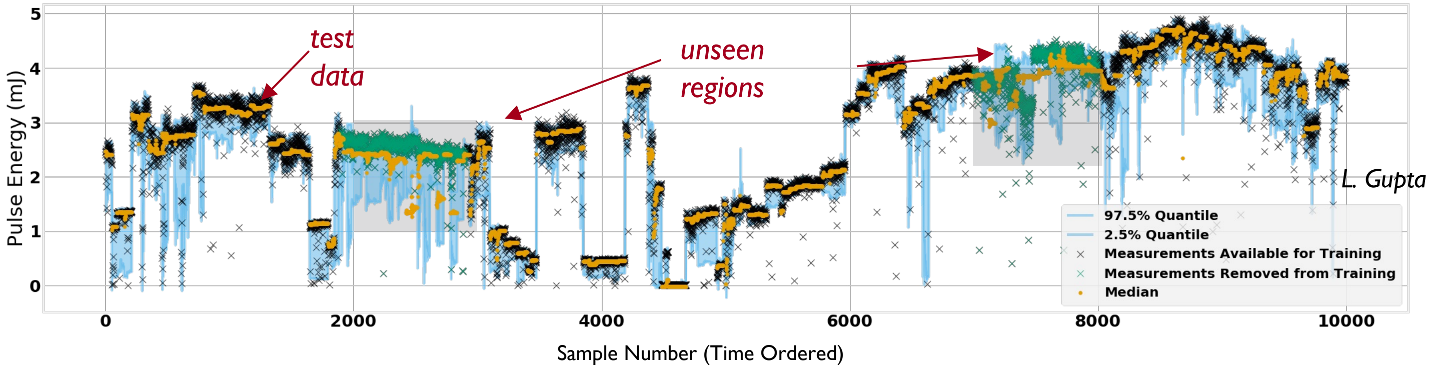


Reliable uncertainty estimates and model adaptation methods are key for putting online models to use operationally

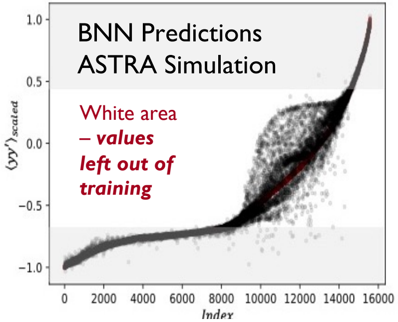
Uncertainty Quantification / Robust Modeling

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

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

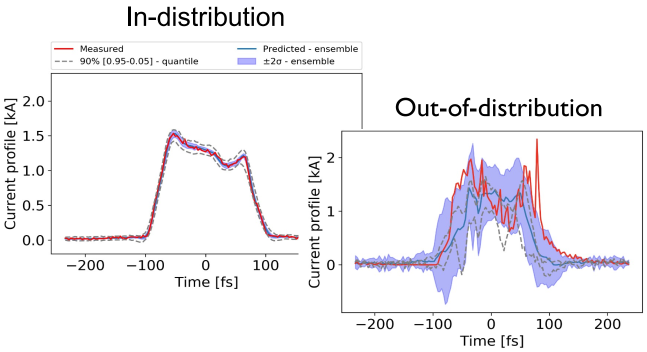


Neural network with quantile regression predicting FEL pulse energy at LCLS



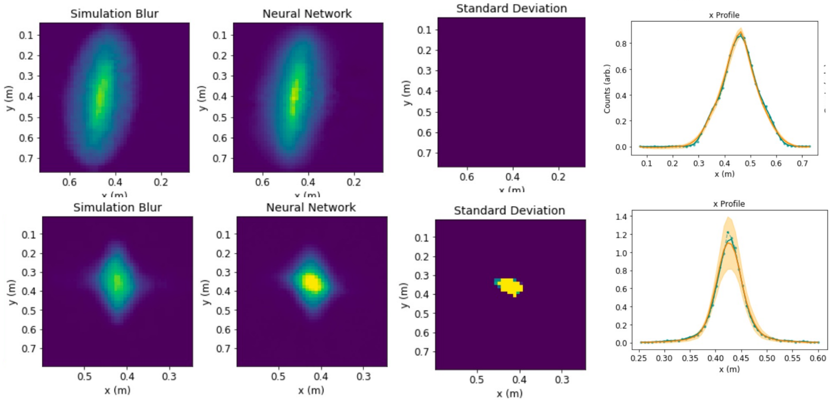
Scalar parameters for the LCLS-II injector (Bayesian neural network)

A. Mishra et. al., PRAB, 2021



longitudinal phase space (quantile regression + ensemble)

O. Convery, et al., PRAB, 2021

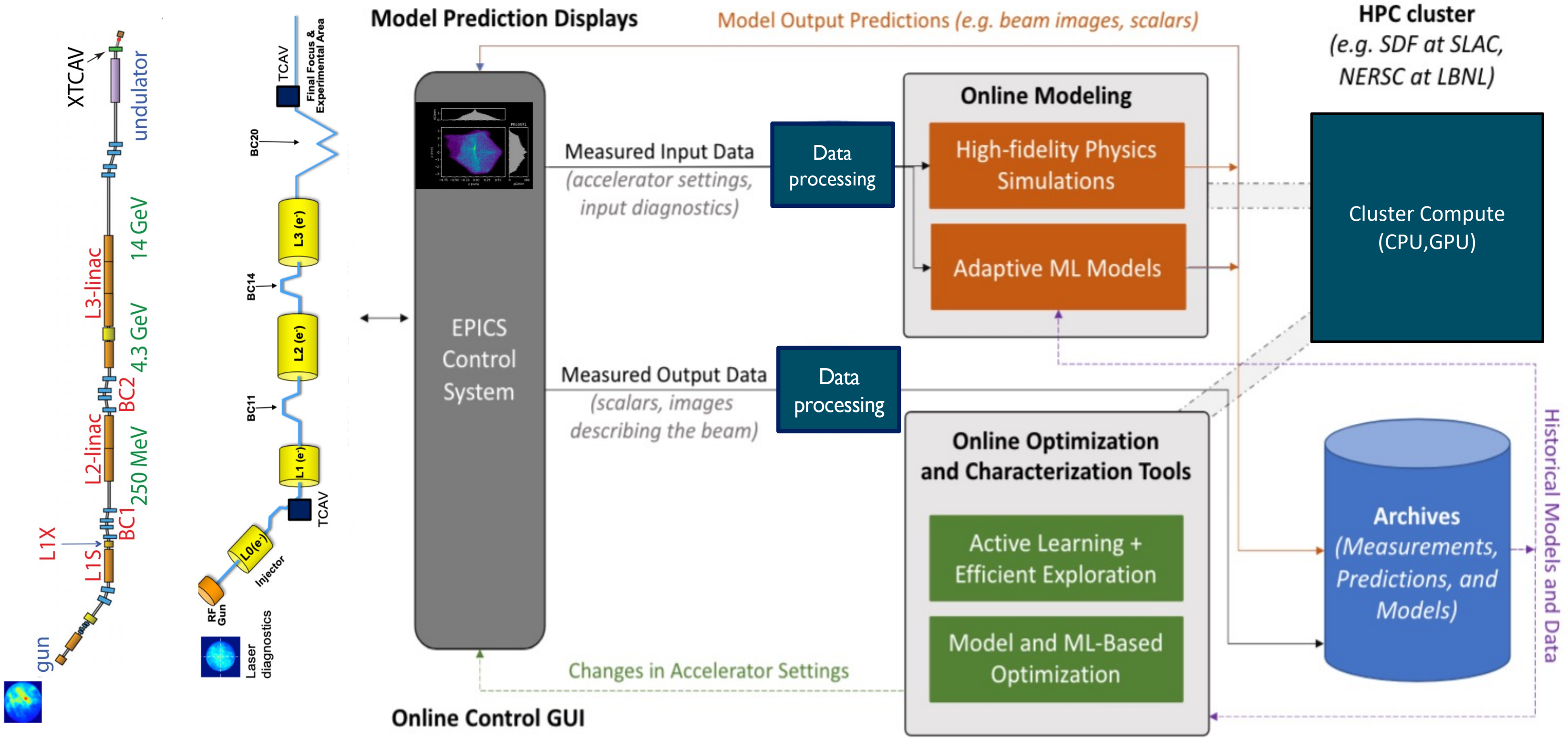


LCLS injector transverse phase space (ensemble)

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)*

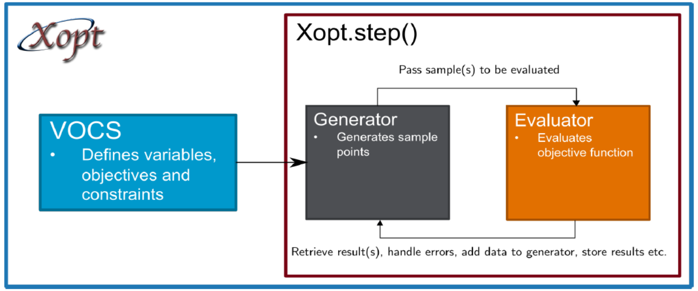


Making good progress toward this vision with open-source, modular software tools

Modular, Open-Source Software Development

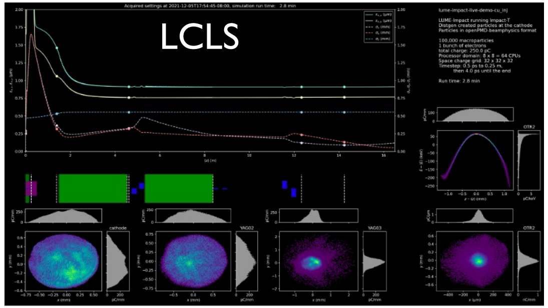
Community development of **re-usable, reliable, flexible software tools** for AI/ML workflows has been essential to maximize return on investment and ensure transferability between systems

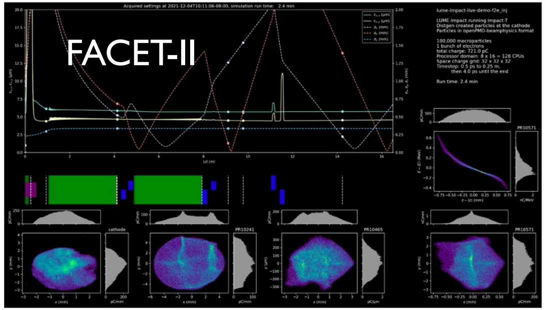
Modularity has been key: separating different parts of the workflow + using shared standards




```
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
```





Different software for different tasks:

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

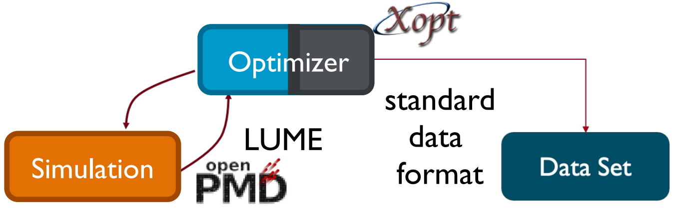
Visual control room interface *(e.g. Badger)*

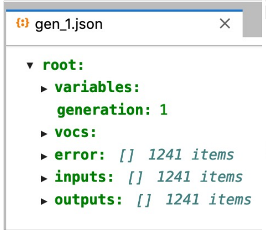
Simulation drivers *(e.g. LUME)*

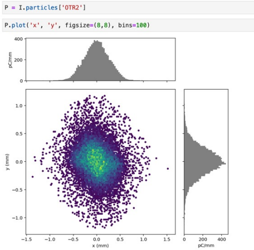
Standards model descriptions, data formats, and software interfaces *(e.g. openPMD)*

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

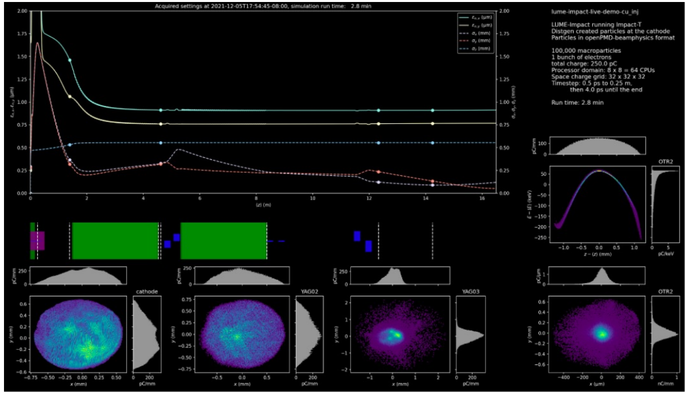
Generality of tooling

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

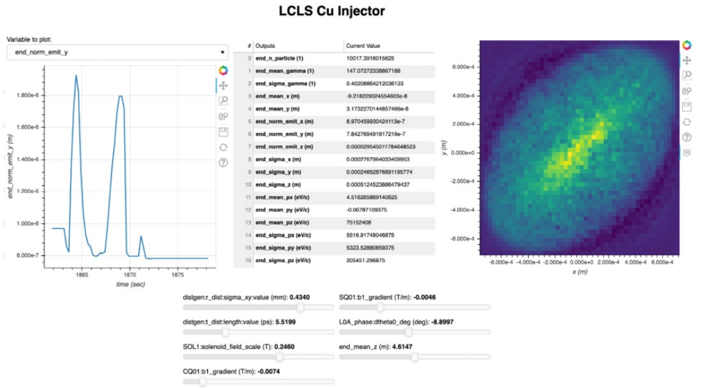
EPICS integration

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

Impact Dashboard:

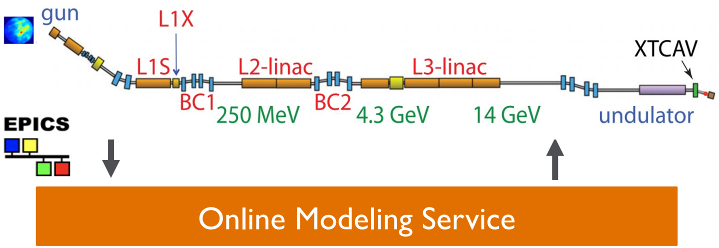


LCLS Injector UI w/ EPICS-based widgets (Using LUME-EPICS tools):



Example applications:

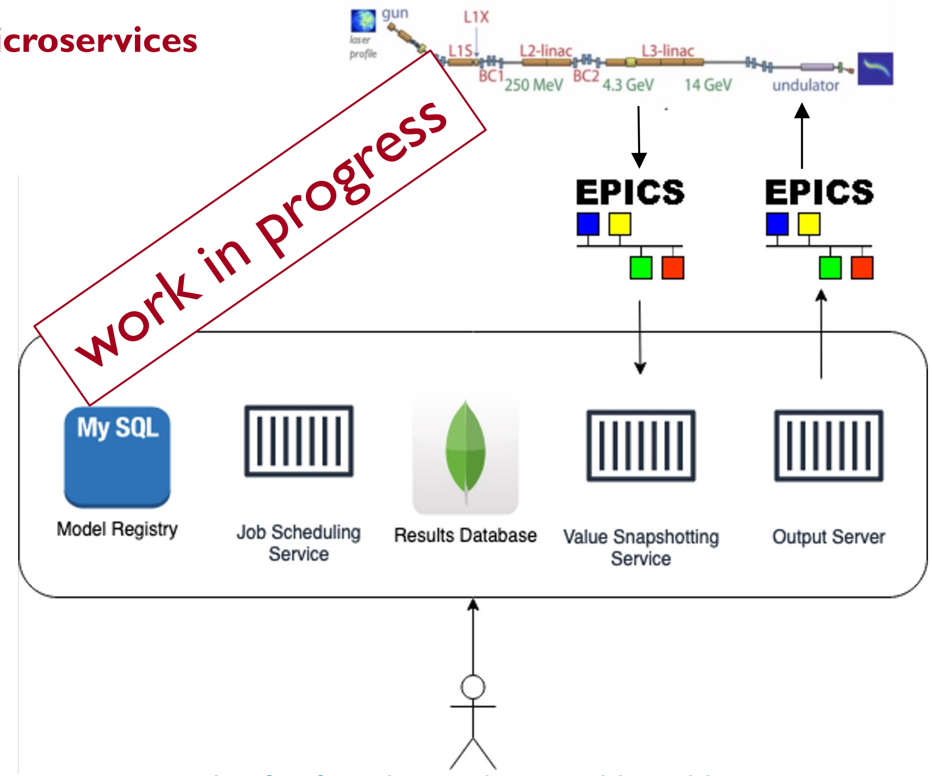
Particle data or screen images (e.g. laser profile) as input (distgen → Impact)
Advanced online visualization
Optimization using online model information (e.g. prior mean for Bayes opt)



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
- 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

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

Aimed for transferrable design between platforms → we welcome collaborators!

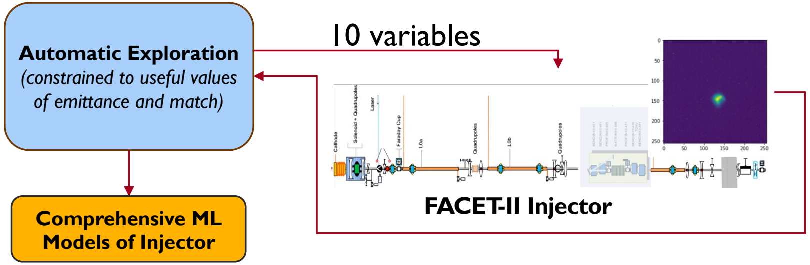
Variety of Successes with Online Modeling and Optimization Tools So Far

- **Digital twins:** Online simulation and modeling with LUME infrastructure + adaptive ML models have been used with LCLS, LCLS-II, and FACET-II

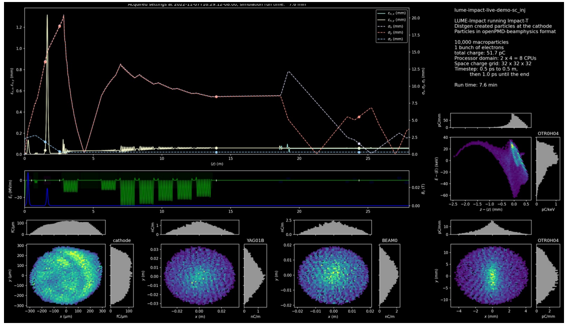
- **Data collection/characterization:** Smart sampling for efficient characterization successfully/robustly used online
  → *data used directly to create ML model*

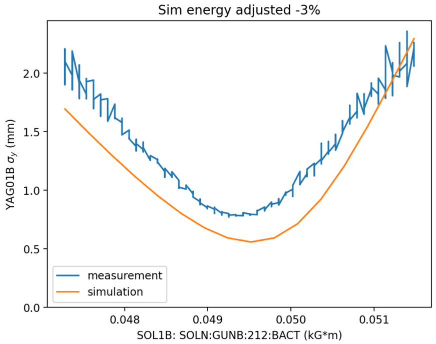
- **ML-enhanced optimization:** numerous successes with new algorithms for safe, efficient online tuning (e.g. injector emittance tuning, FEL pulse energy tuning, longitudinal phase space tuning, sextupole tuning for beam size)

- **Software transferability:** have shown easy transfer of modeling and tuning software between LCLS/LCLS-II/FACET-II accelerators and to other labs (e.g. AWA at Argonne); now working integrate between accelerator/photon side at LCLS/LCLS-II



8x faster characterization than by hand





LCLS-II live simulation of injector *(with nonlinear collective effects included)* and online re-calibration to match measurements

Used combination of **online physics simulation and custom Bayesian tuning algorithms** in LCLS-II injector commissioning
→ ***achieved best emittance to date***

Critical steps toward having an adaptive digital twin system and model-guided, ML-enhanced optimization have been demonstrated
→ *now need infrastructure investment + more incorporation into regular operations*

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
- 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 (e.g. Badger, Xopt), others are used regularly offline (e.g. Xopt, LUME), others need substantial investment / work (e.g. LUME-services)

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*

Next slide: pain points

Pain Points → *Where have we encountered challenges?*

- **Data coordination**
  - Consistent BSA (120 Hz) accelerator and photon side data streams (plus tools for combining)
  - 1hz archive, 120 Hz archive, Matlab files, etc

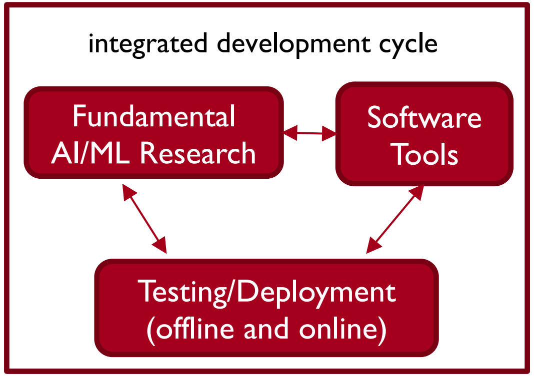
- **Cameras** (saving images + archiving them, accurate timestamps to correlate with BSA data) → many upgrades but some remain TBD after years

- **Data cleaning**
  - Many variables, much unknown → have preferred to use data from known shifts
  - How to flag/filter for different machine states from the archive
  - Sensitivity/feature importance → would be nice to filter variables easily for different problems *(archive data doesn't represent all variables well; can use smart sampling to supplement)*

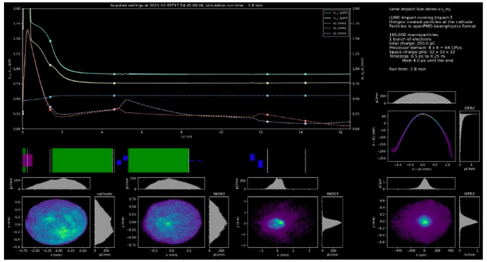
- **Continuous deployment/integration of simulation models**
  - Need to do I/O between control system and HPC
  - Managing "virtual" accelerators (PV naming, etc)
  - Biggest problem: people power + software engineering support

- **Logistical/social**: beam time for testing, socialization of tools into control room, buy-in from operations
  → need cooperative development cycle with operations and time to test in order to make truly robust tools

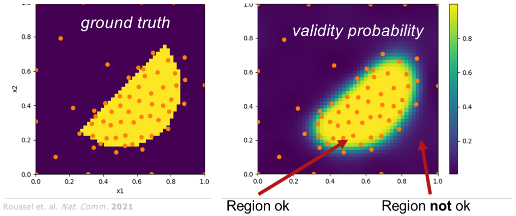
(1) Developing new approaches for accelerator optimization/characterization and faster higher-fidelity system modeling, (2) developing portable software tools to support AI/ML, (3) integrating these into regular use



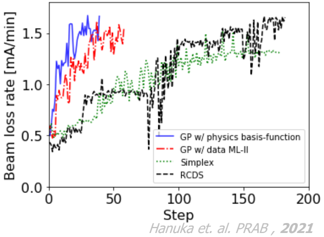
Online prediction with physics sims and **fast/accurate ML models**



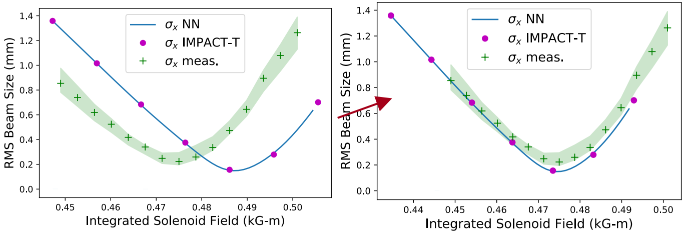
Efficient optimization and characterization *(useful also for simulation exploration/design, data generation)*



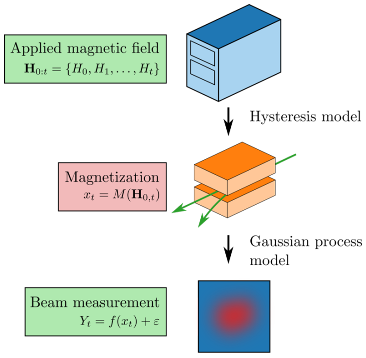
Output constraints learned on-the-fly



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



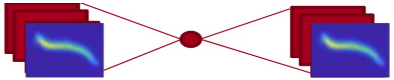
Techniques for **combining physics and ML** *(more reliable/transferrable, require less data, more interpretable)*, including **differentiable simulators**



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

Representation learning

(e.g. better ways of modeling beams)



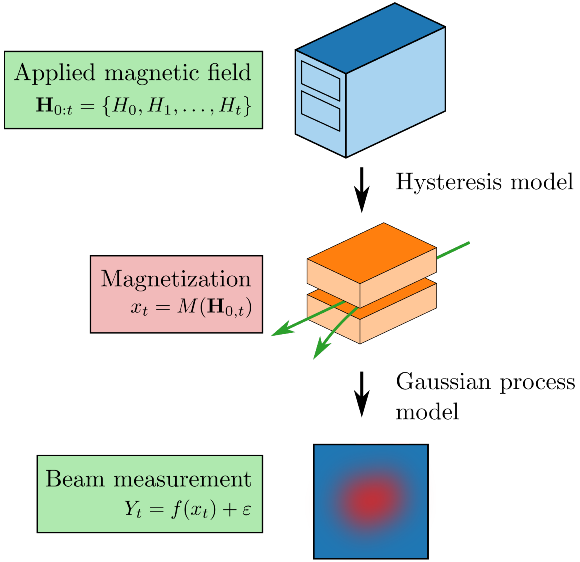
Software packages and standards for data generation, modeling, and optimization (*LUME, xopt, Badger*)

Backups

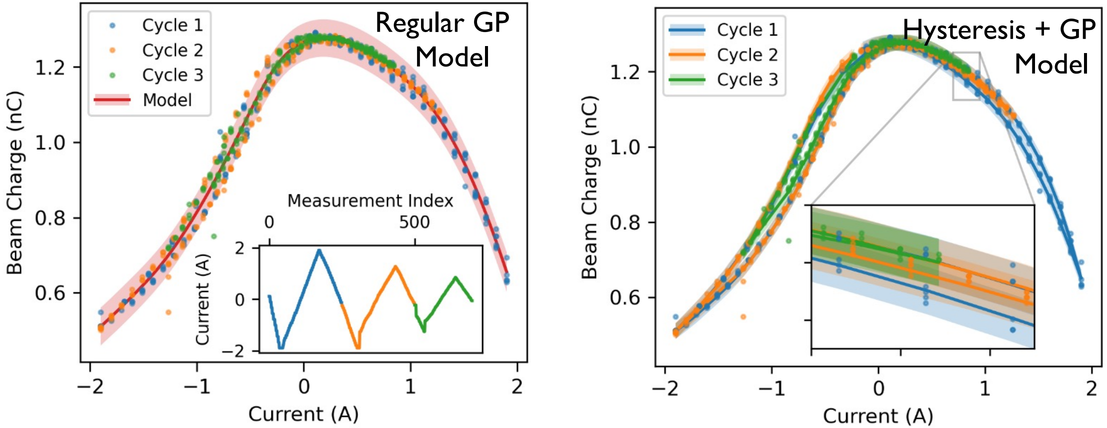
Example: Differentiable Physics + ML Modeling of Hysteresis

Magnetic hysteresis has been a major impediment to high-precision tuning → *historically required standardization of magnets*

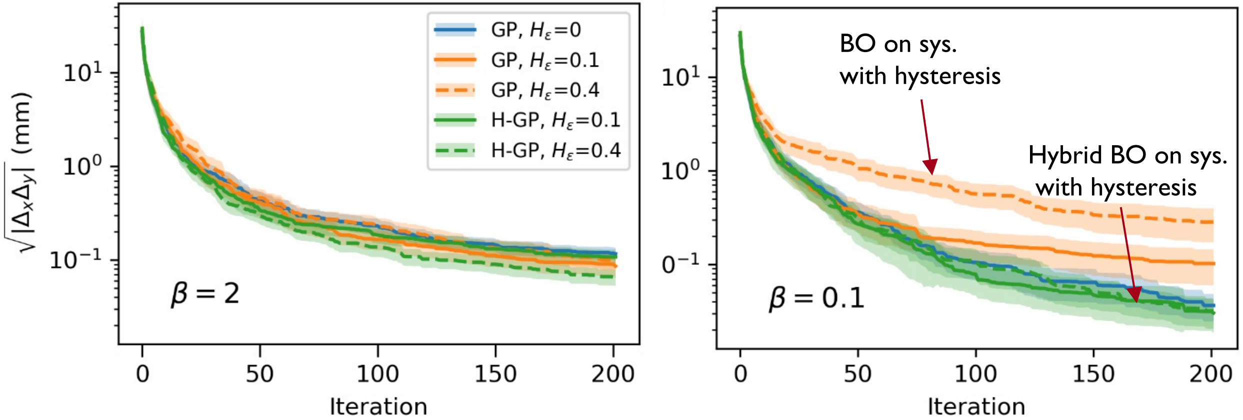
New modeling approach combining classical Preisach model and a Gaussian Process



R. Roussel, et al., PRL, 2022



Joint modeling of hysteresis and beam propagation is more accurate and enables in-situ hysteresis characterization



Higher-precision optimization possible when including hysteresis effects in model

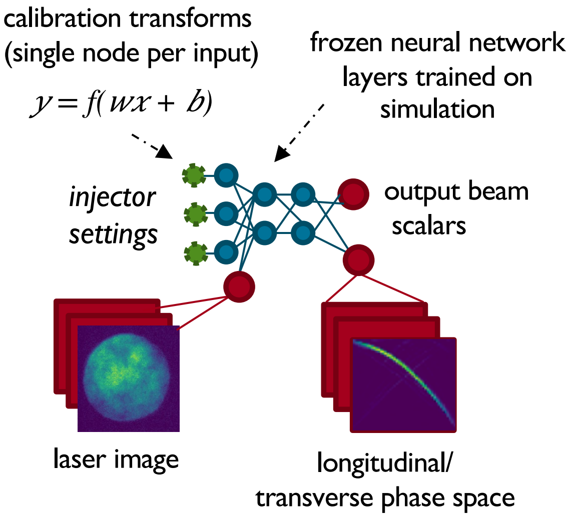
Promising example showing the power of differentiable physics and ML models to enable high-precision characterization and control with minimal data.

Finding Sources of Error Between Simulations and Measurement

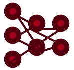
Many non-idealities not included in physics simulations:

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

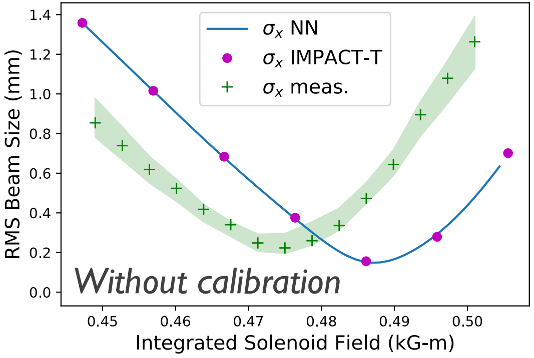
Want to identify these to get ***better understanding of machine*** *→* ***fast-executing ML model allows fast / automatic exploration of possible error sources simultaneously***

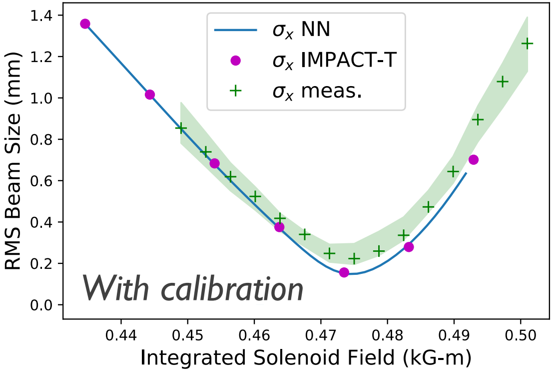


Inputs
Laser radius
Laser spot sizes
Pulse length
Charge
Solenoid
L0A phase
L0B phase
SQ quad
CQ quad
6 matching quads



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





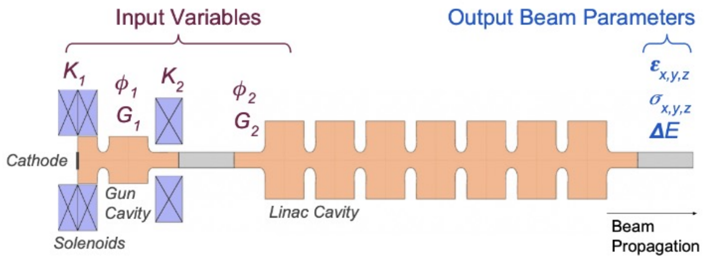
Calibration offset in solenoid strength found automatically with neural network model (trained in simulation, then calibrated to machine)
Example above is simulation-to-machine, but can adapt model over time as well

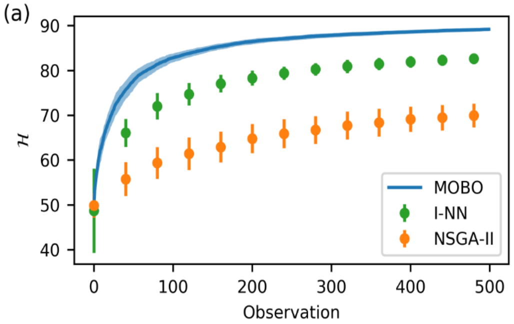
First studies look promising → current work focuses on examining robustness and extending to larger subsystems

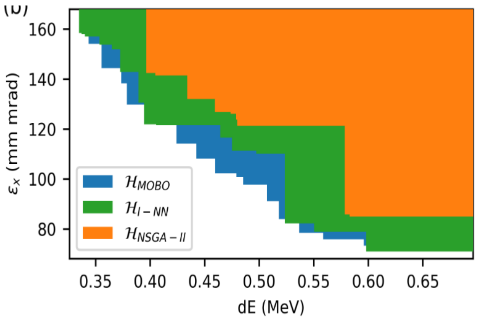
Example: Multi-Objective Bayesian Optimization (MOBO)

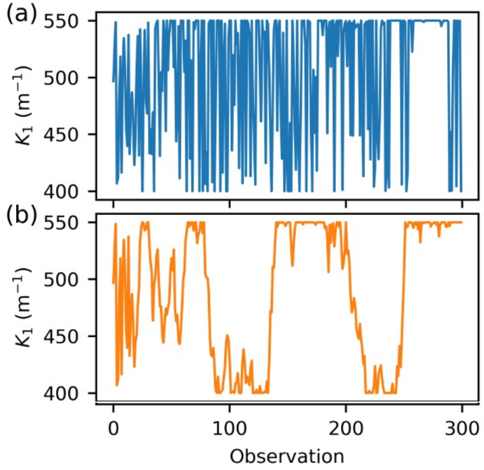
Multi-objective optimization (MOO) in accelerators is traditionally done offline with high performance computing and simulations, or online at individual working points only

- MOBO enables full characterization of optimal beam parameter tradeoffs (i.e. the Pareto front) online with high sample-efficiency
- Has now been used experimentally at AWA, FACET-II, LCLS and SLAC UED









Can enforce smooth exploration

(no wild changes in input settings)

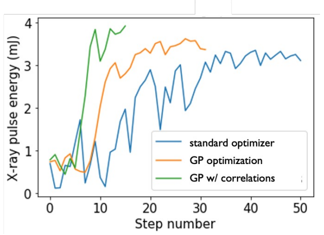
R. Roussel, et al., PRAB (2021)

Unprecedented ability to fully characterize tradeoffs between beam parameters in real accelerator systems.

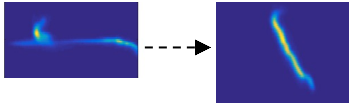
Broad Set of Areas for ML to Impact Operation

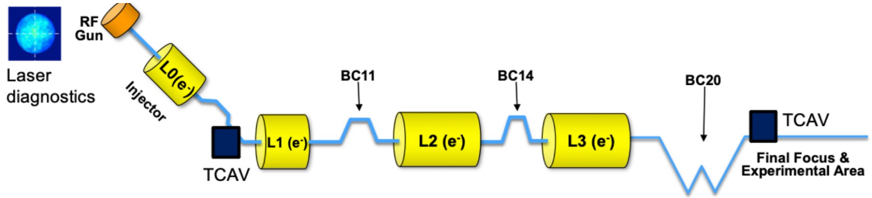
Data reduction/rejection *(kHz/MHz data streams)*
Event triggering

automated control
+ optimization

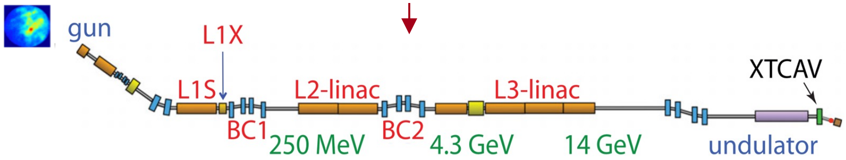


J. Duris et al., PRL, 2020

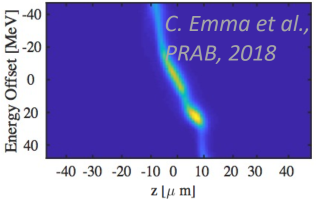




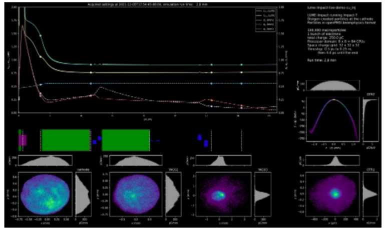
algorithm transfer between systems

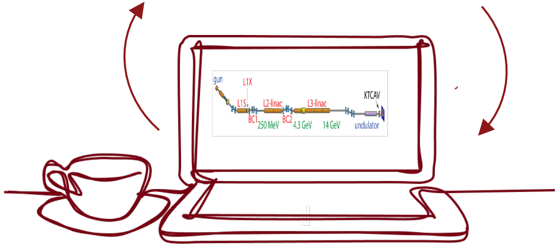


ML-enhanced diagnostics
(provide insight at faster rate, at higher resolution, non-invasively)



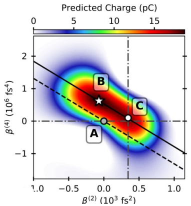
anomaly detection
failure prediction
(plan maintenance; alert to changes in machine; alert to interesting science)





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

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



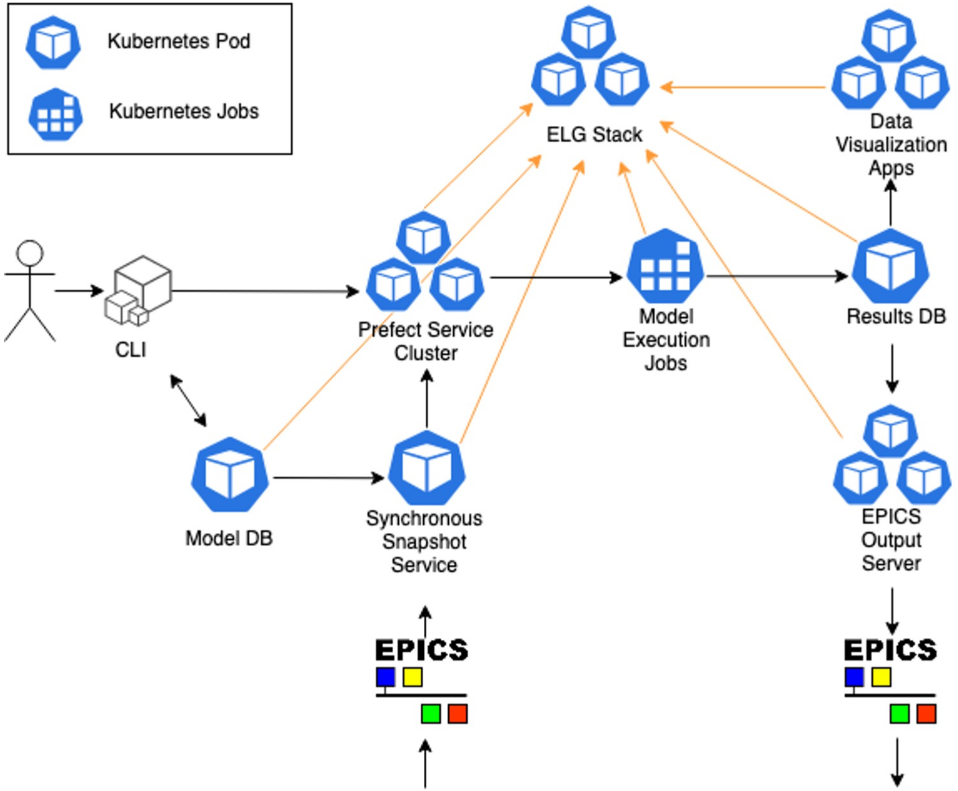
R. Shaloo et al. arXiv:2007.14340

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

Component Architecture

Components

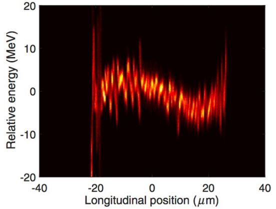
High-level component	Function
Model DB	● Stores model metadata ● Tracks versioned deployments and associated workflows
Synchronous Snapshot Service	● Single pulse EPICS PV collection ● Submission of Prefect workflow runs
Prefect Service	● Orchestration of workflows ● Workflow monitoring ● Result management
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



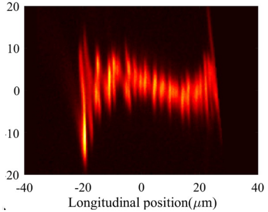
In reality things are much more difficult…

computationally expensive simulations

Simulation



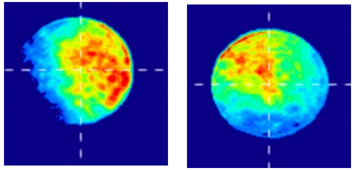
Measurement

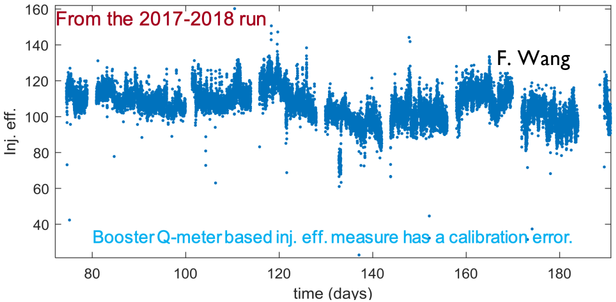


10 hours on thousands of cores at NERSC!

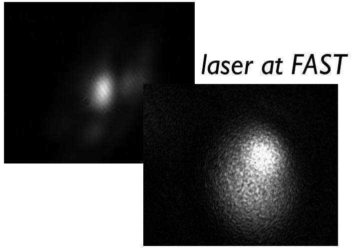
J. Qiang, et al., PRSTAB30, 054402, 2017

fluctuations/noise (e.g. laser spot)



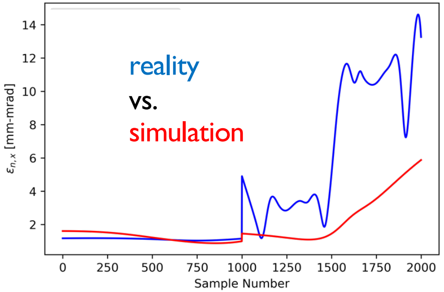


hidden variables / sensitivities

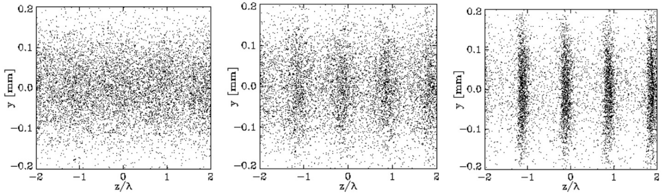


laser at FAST

drift over time



many small, compounding sources of uncertainty

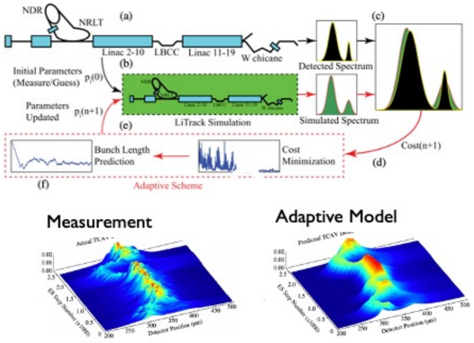


nonlinear effects / instabilities

Virtual Diagnostics

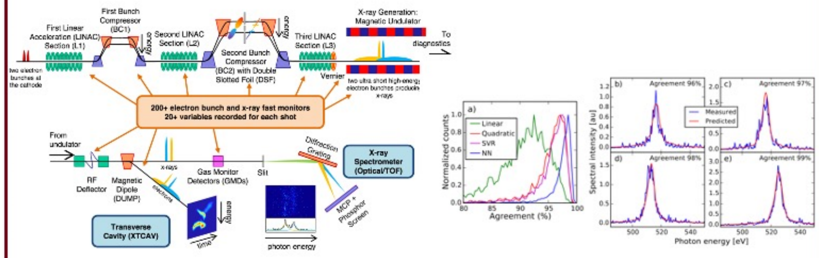
Provide information about parts of the system that are typically inaccessible (destructive, too slow, not directly measurable)

Adaptively tune a simple physics model



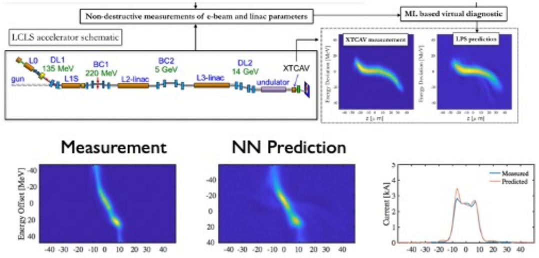
A. Scheinker, S.Gessner, PRAB 18, 102801 (2015)

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



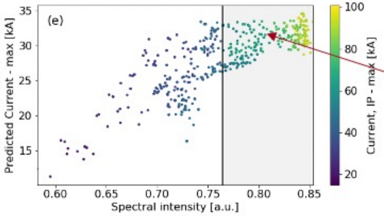
A. Sanchez-Gonzalez, et al., Nature Comms (2017)

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:

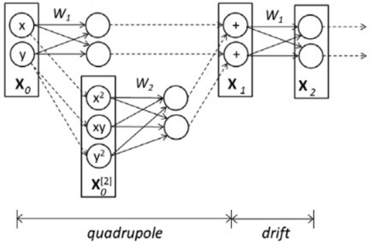
- Combine physics representations and machine learning models directly *(e.g. differentiable simulations)*

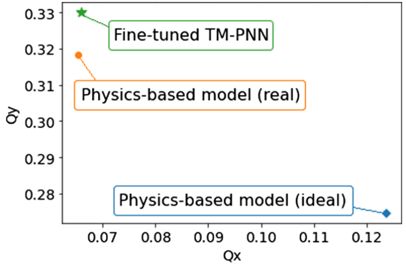
- Add physics constraints to output metrics

- Force to satisfy expected symmetries
  (e.g. inductive biases in ML model)

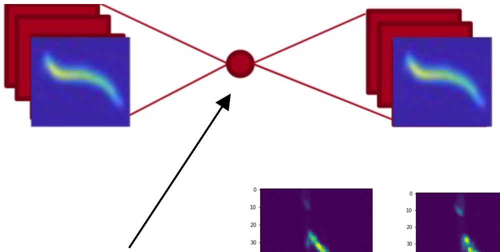
- Loose form: learn from many physics sims in a way that results in good representation of the physics *(also related to representation learning)*

Differentiable Taylor map physics model + weights → train like ML model
needed very little data to calibrate PETRA IV model
Ivanov et al, PRAB, 2020



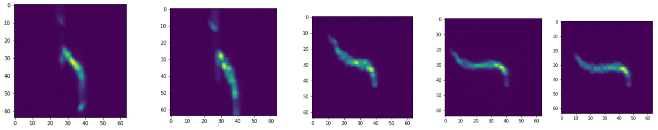


Physics-driven representation learning
(e.g. encoder-decoder neural network models)



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

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

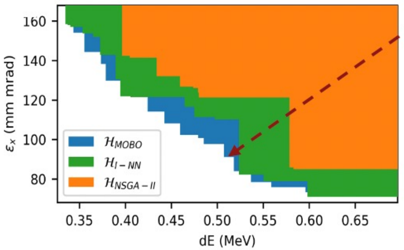
next point to search

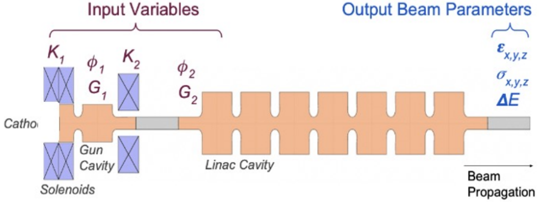
Learning Algorithm
(GP model + Bayes. opt., iterative surrogate, RL agent)

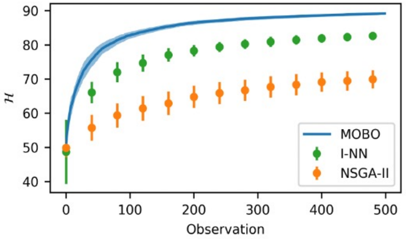
System (Simulation or Online Interaction)

data

Pareto front: optimal tradeoff between parameters of interest



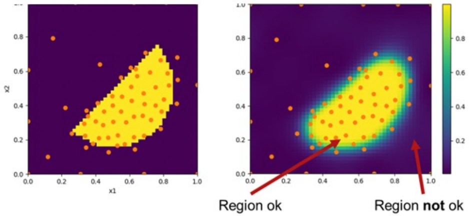




Faster multi-objective optimization with Bayesian optimization and iterated surrogate models

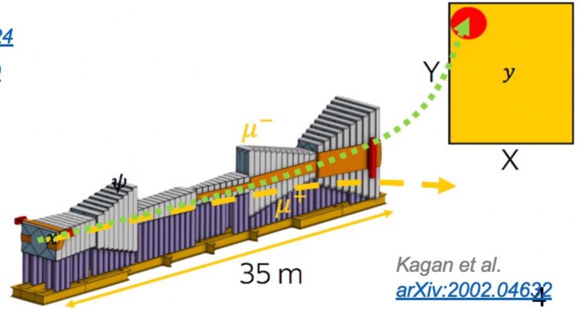
R. Roussel et al., arXiv:2010.09824

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