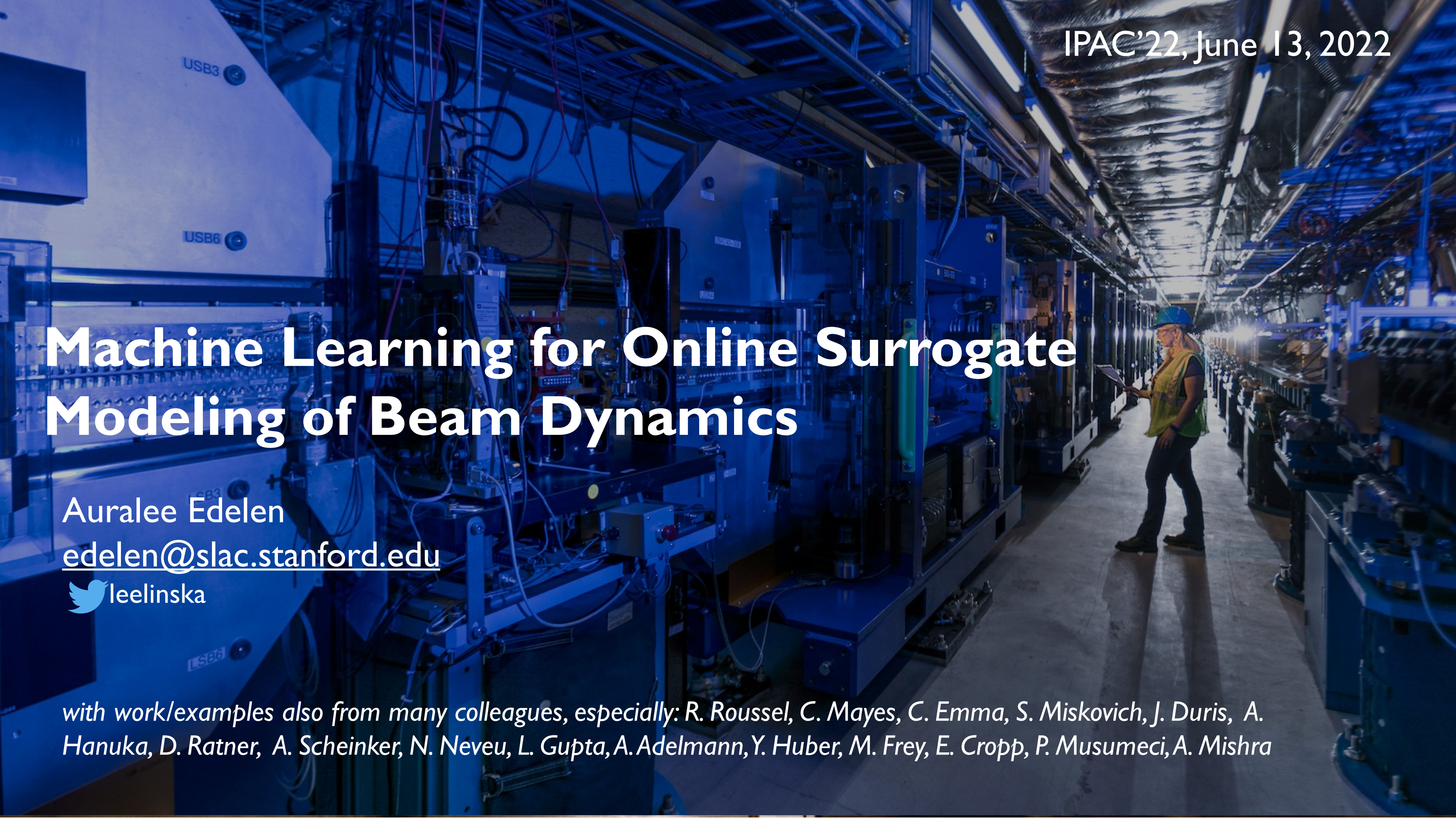
IPAC'22, June 13, 2022

Machine Learning for Online Surrogate Modeling of Beam Dynamics

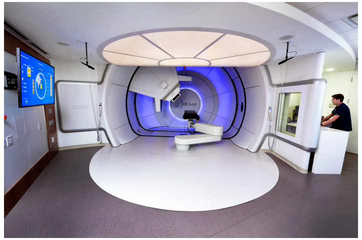
Auralee Edelen
edelen@slac.stanford.edu
leelinska

with work/examples also from many colleagues, especially: R. Roussel, C. Mayes, C. Emma, S. Miskovich, J. Duris, A. Hanuka, D. Ratner, A. Scheinker, N. Neveu, L. Gupta, A. Adelmann, Y. Huber, M. Frey, E. Cropp, P. Musumeci, A. Mishra

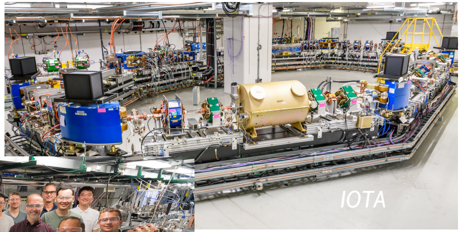


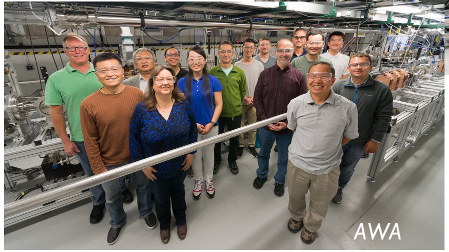
Large User Facilities

Industrial / Medical

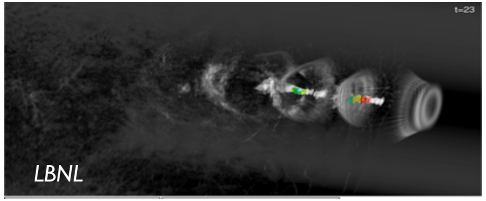


Small Test Facilities

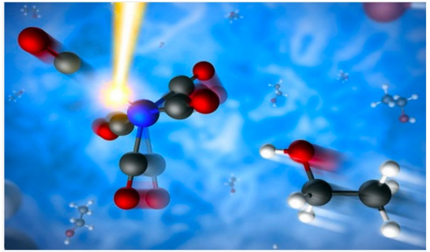


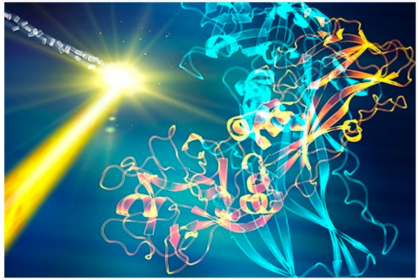


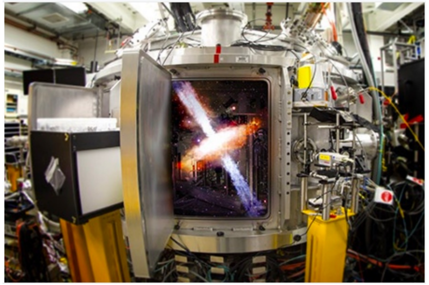
Novel Acceleration Schemes



Lots of different specific needs, but many broadly similar challenges in online modeling, machine understanding, and control





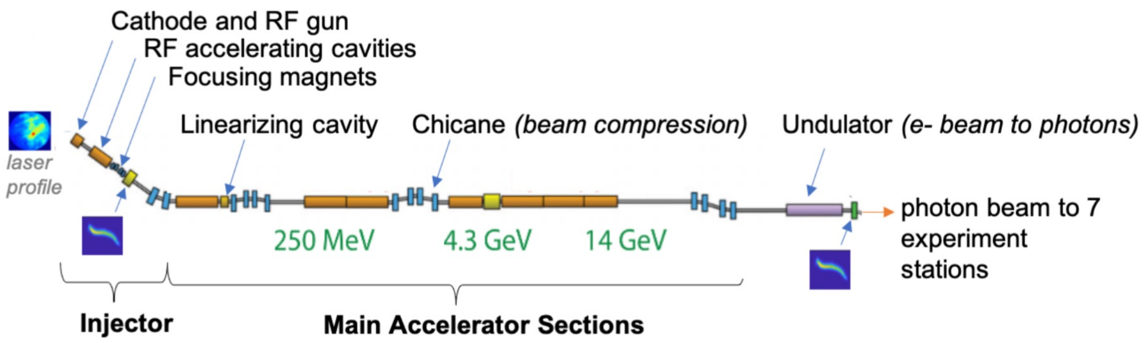


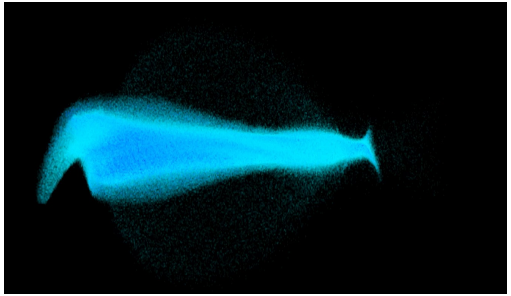
1,062 experiments in 2016

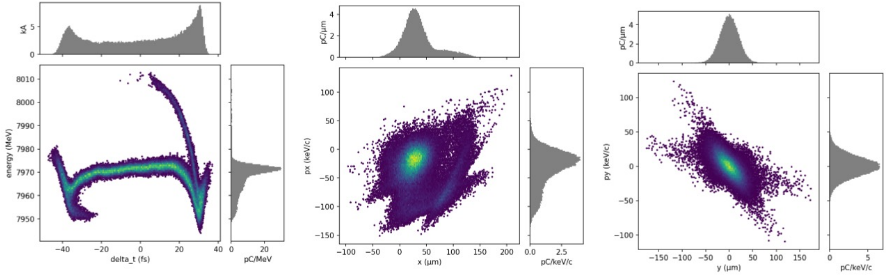
~1023 papers since 2009

Experimenters come for a few days – a week

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





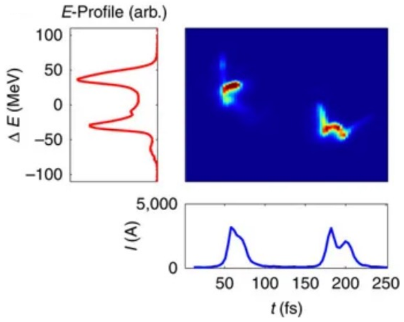


A. Marinelli, et al., Nat. Commun. **6**, 6369 (2015)

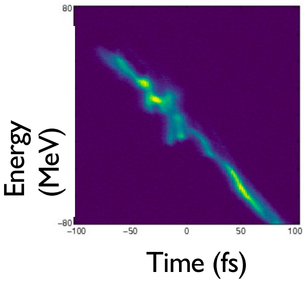
Beam exists in 6-D position-momentum phase space

Have incomplete information: measure 2-D projections or reconstruct based on perturbations of upstream controls

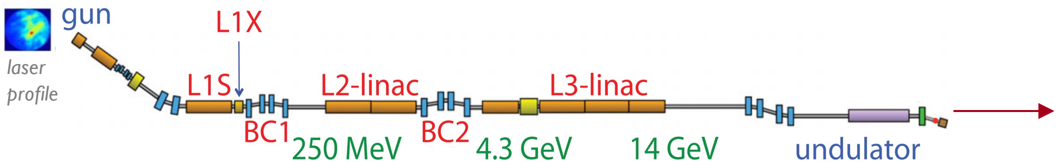
Can have dozens-to-hundreds of controllable variables and hundreds-of-thousands to millions to monitor



A. Marinelli, IPAC'18

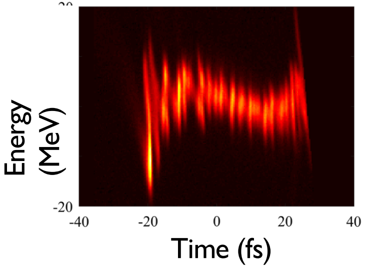


Nonlinear, high-dimensional optimization problem

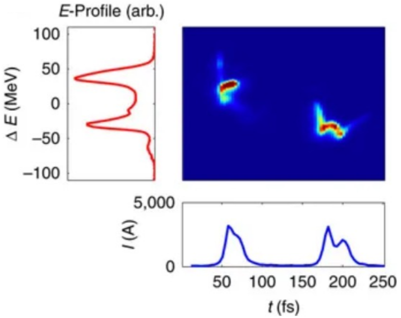


photon beam to
7 experiment
stations

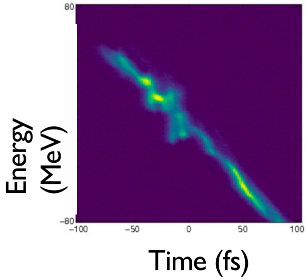
J. Qiang et al, PRAB (2017)



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

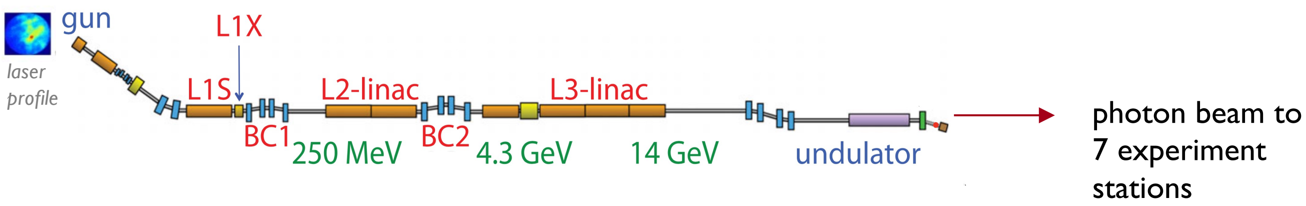


A. Marinelli, IPAC'18

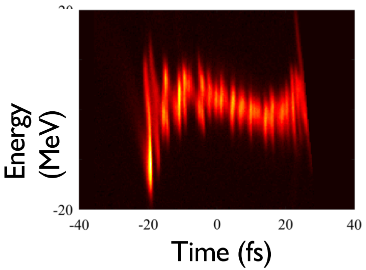


Approximate Annual Budget: $145 million
Approximate hours of experiment delivery per year: 5000
About $30k per experiment hour to run

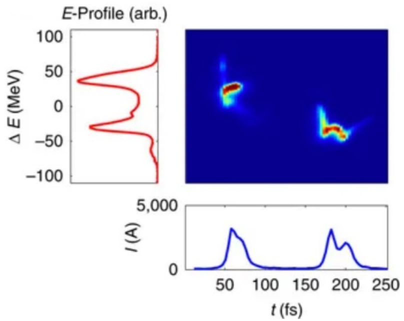
400 hours hand-tuning in a year → $12 million value
~10 additional experiments



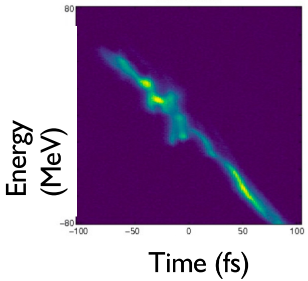
J. Qiang et al, PRAB (2017)



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



A. Marinelli, IPAC'18



Rapid beam customization

Achieve new configurations + unprecedented beam parameters

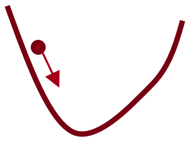
Fine control to maintain stability within tolerances

Tuning approaches can leverage different amounts of data/previous knowledge

assumed knowledge of machine

less ⟷ more

Model-Free Optimization

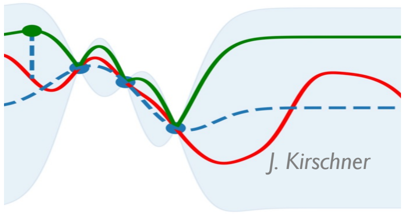


Observe performance change after a setting adjustment

→ estimate direction toward improvement

gradient descent
simplex

Model-guided Optimization

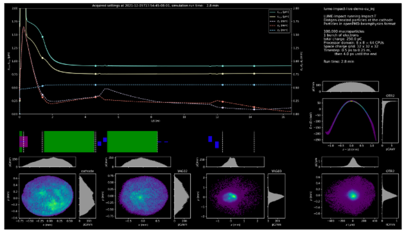


Update a model at each step

→ use model to help select the next point

Bayesian optimization
Reinforcement learning

Global Modeling

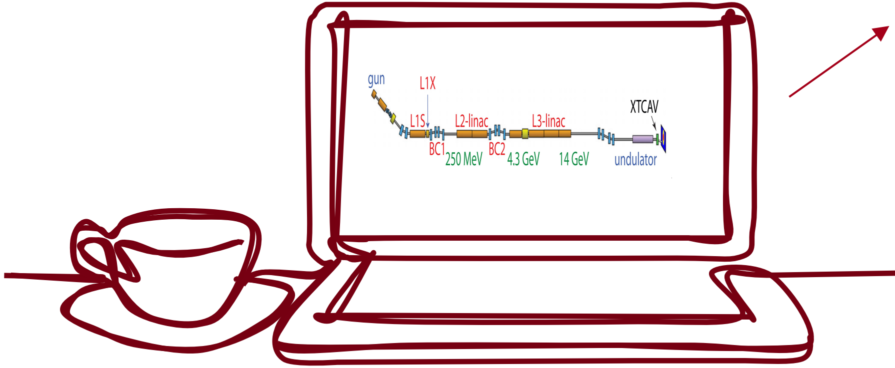


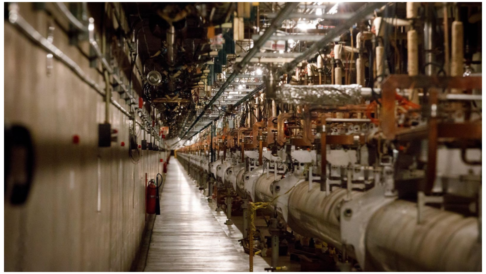
Make fast system model

→ provide guess for settings
→ machine insight from predictions

ML system models +
inverse models

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

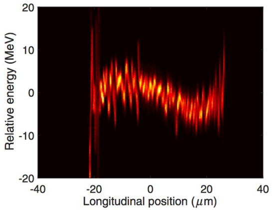
get info about unobserved parts of machine (online model / virtual diagnostic)

do offline planning and control algorithm prototyping

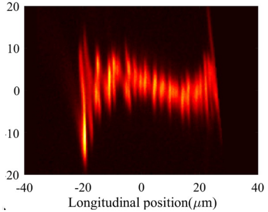
In reality things are much more difficult…

computationally expensive simulations

Simulation



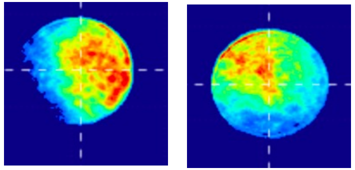
Measurement

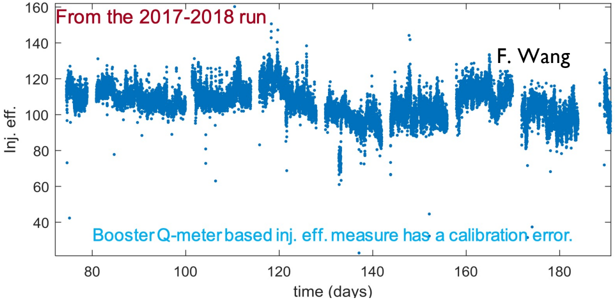


"10 hours on thousands of cores at the NERSC"

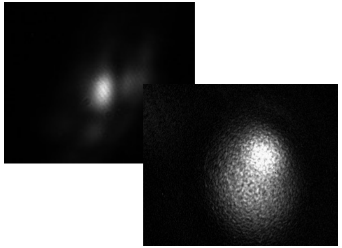
J. Qiang, et al., PRSTAB30, 054402, 2017

fluctuations/noise (e.g. laser spot)

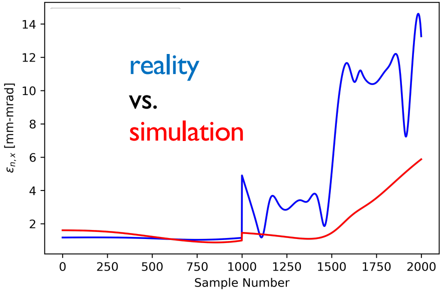




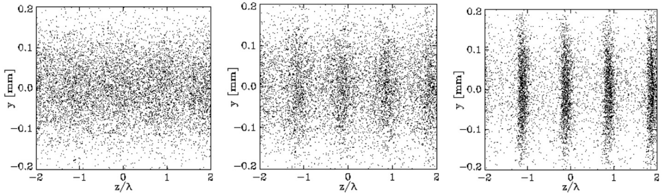
hidden variables / sensitivities



drift over time



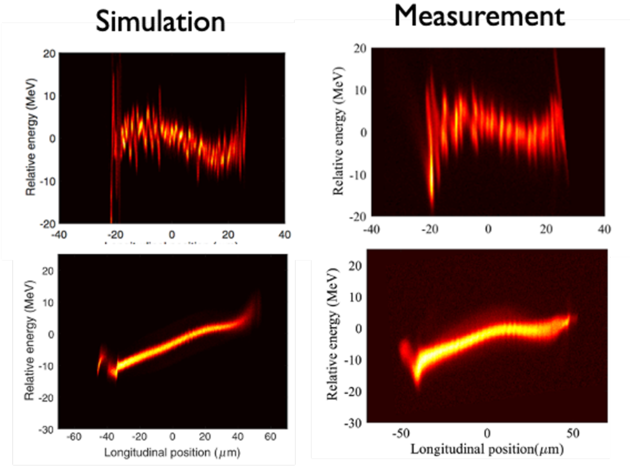
many small, compounding sources of uncertainty



nonlinear effects / instabilities

AI/ML is poised to help with speed, accuracy, and adaptability of models

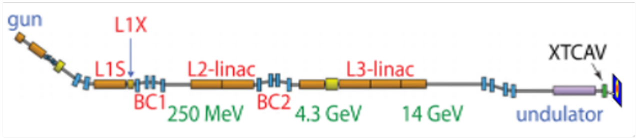
Accelerator simulations that include nonlinear and collective effects are powerful tools, but they can be computationally expensive



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

"10 hours on thousands of cores at the NERSC"

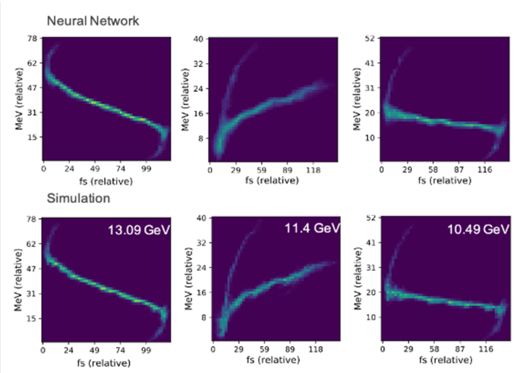
ML models can provide fast approximations to simulations



Linac sim in Bmad with collective beam effects

Scan of 6 settings in simulation

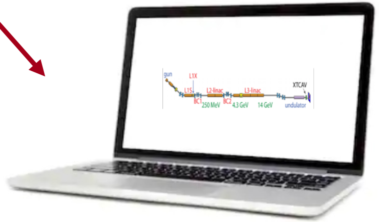
Variable	Min	Max	Nominal	Unit
L1 Phase	-40	-20	-25.1	deg
L2 Phase	-50	0	-41.4	deg
L3 Phase	-10	10	0	deg
L1 Voltage	50	110	100	percent
L2 Voltage	50	110	100	percent
L3 Voltage	50	110	100	percent



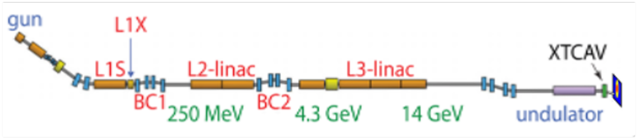
< ms execution speed

10^6 speedup

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



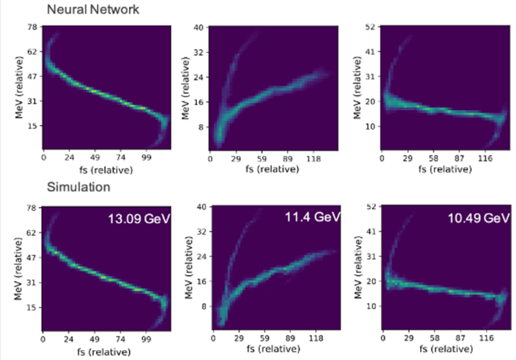
ML models can provide fast approximations to simulations



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

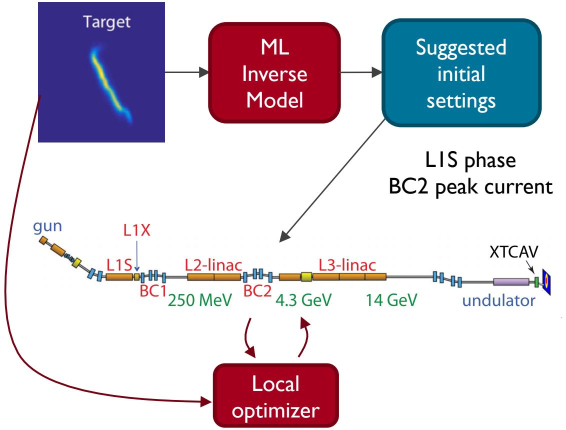


< ms execution speed

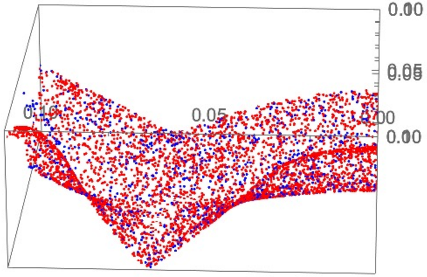
10^6 speedup

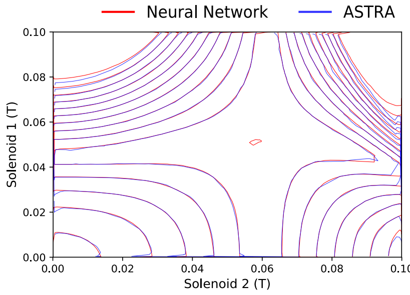
Warm starts for optimization

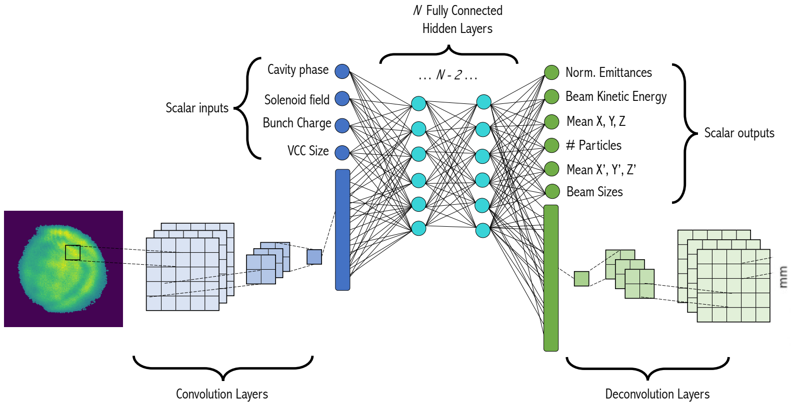
A. Scheinker, A. Edelen, et al, PRL, 2018



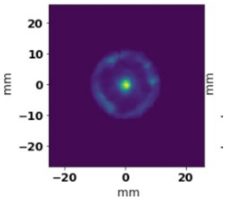
Smooth interpolation
Example σ_x surface from 2D scan, LCLS-II Injector



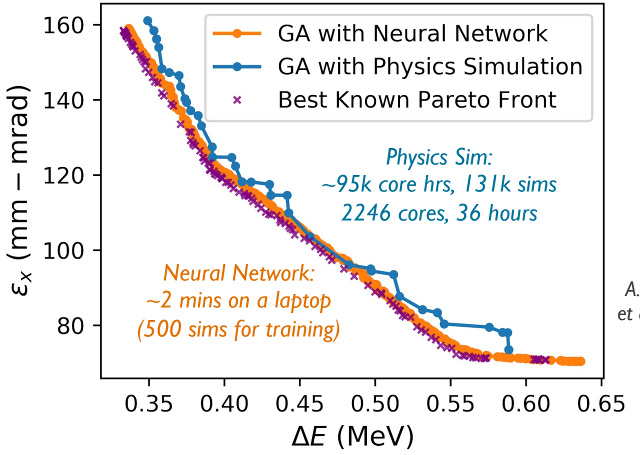




L. Gupta, et al, MLST, 2021



Include high-dimensional input information → better output predictions

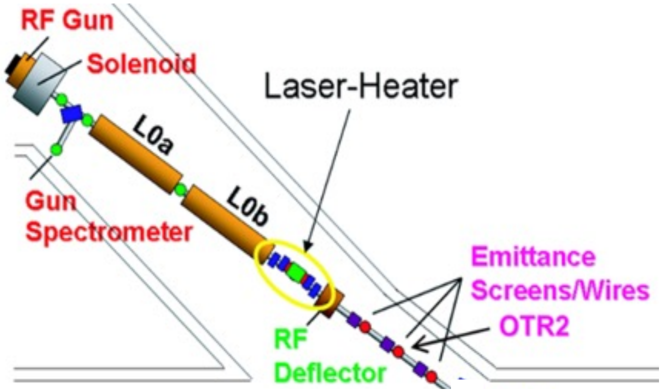


A. Edelen et al., PRAB, 2020

Surrogate-boosted design optimization (example on AWA)

Example Use Case: LCLS Injector Surrogate Models

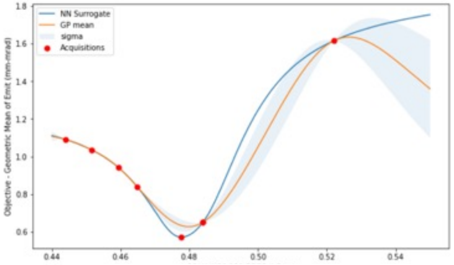
- Neural networks trained on IMPACT-T sims
- Several versions aimed at different outputs and goals *(e.g. 6D phase space projections, scalars along z, interpolation vs. accuracy on known configurations)*
- Inputs sampled widely across valid ranges
  - *Inputs: laser length + spot size, L0 phases, solenoid strength, SQ/CQ quads, 6 matching quads*
  - *Outputs: emittances, bunch length, spot sizes, covariances, energy*

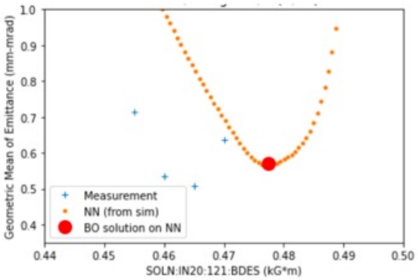


Have been using extensively for algorithm development

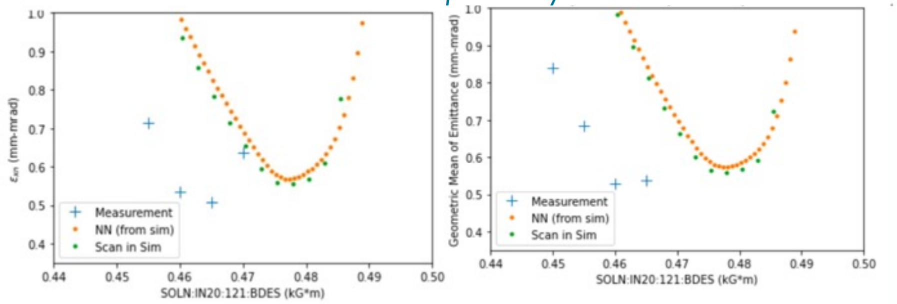
e.g. new Bayesian optimization methods, adaptive emittance measurement →TUPOST059

Example prototyping Bayesian optimization

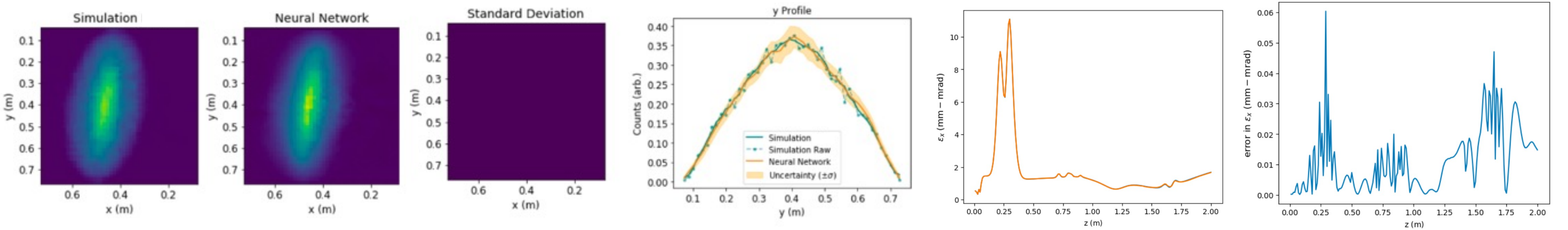




IMPACT-T and SM trained on it are qualitatively similar to measurements



Example outputs



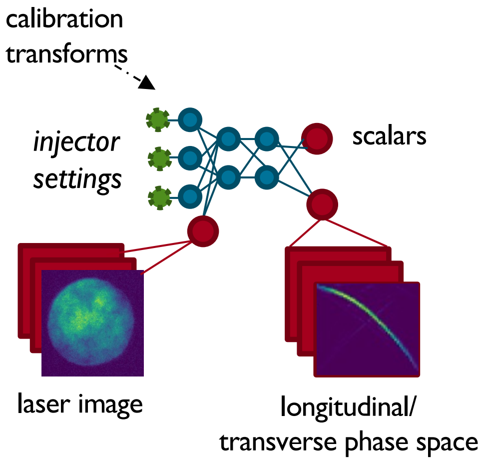
Finding Sources of Error Between Simulations and Measurement

Many non-idealities not included in physics simulations:

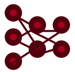
- **static error sources** (e.g. magnetic field nonlinearities, physical offsets)
- **time-varying changes** (e.g. temperature-induced phase calibrations)

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

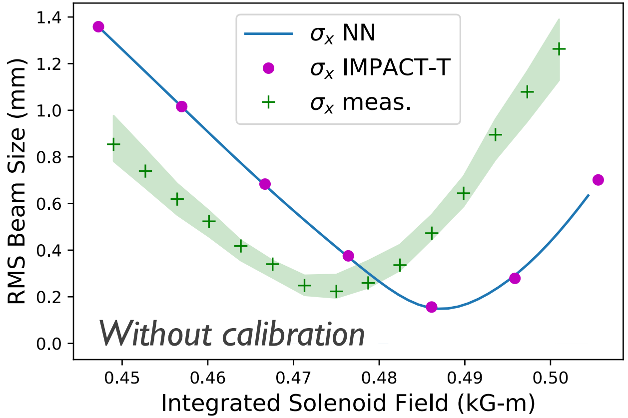
→ fast-executing ML model allows fast / automatic exploration of possible error sources

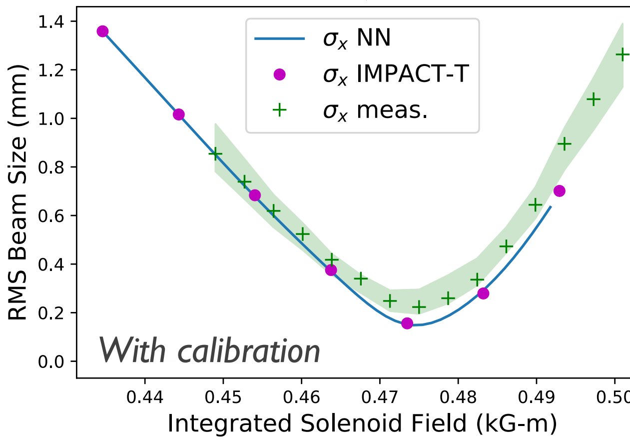


Inputs
Laser radius
Laser spot sizes
Pulse length
Charge
Solenoid
L0A phase
L0B phase
SQ quad
CQ quad
6 matching quads



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





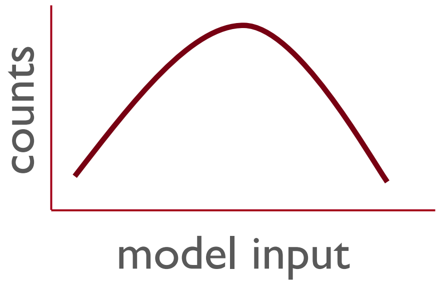
Here: calibration offset in solenoid strength found automatically with neural network model (trained first in simulation, then calibrated to machine)

Fundamental problem for using models online and for tuning: **distribution shift**

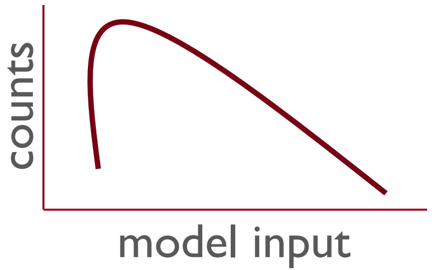
→ *accuracy is degraded on data outside of the statistical distribution of the training data*

→ ***many ML approaches don't consider uncertainty estimates***

training set



new conditions

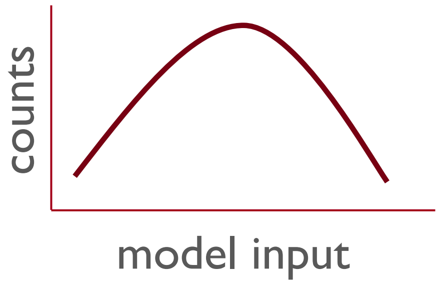


Fundamental problem for using models online and for tuning: **distribution shift**

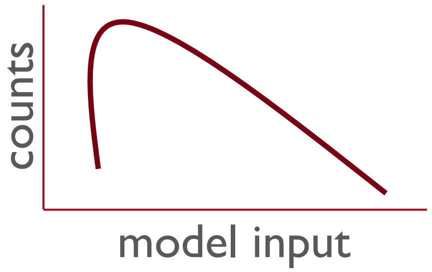
→ *accuracy is degraded on data outside of the statistical distribution of the training data*

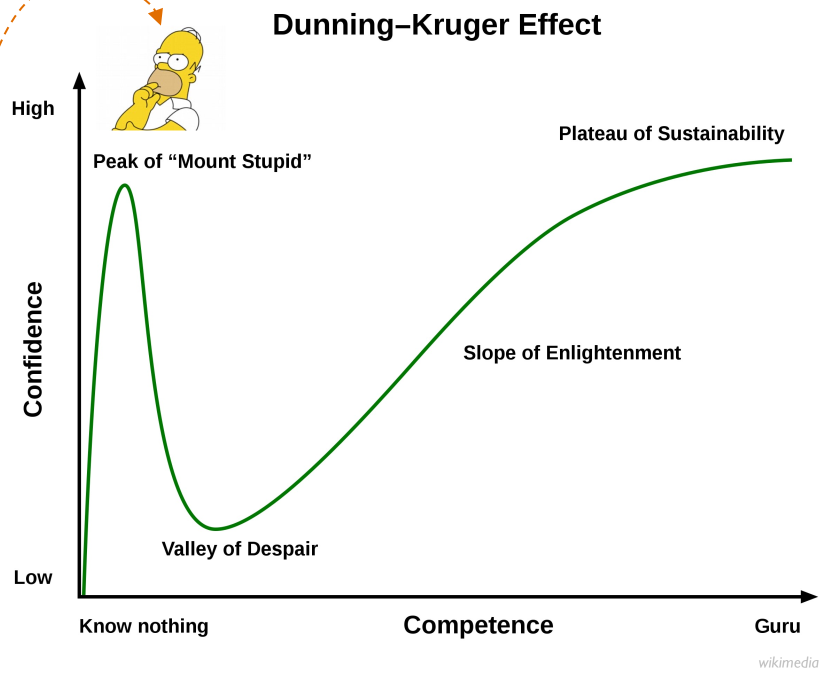
→ ***many ML approaches don't consider uncertainty estimates***

training set



new conditions



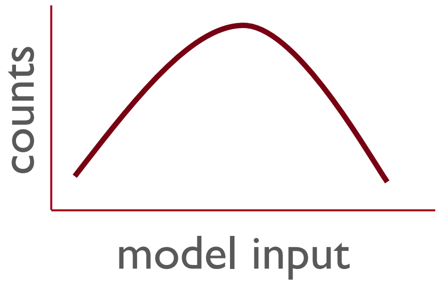


Fundamental problem for using models online and for tuning: **distribution shift**

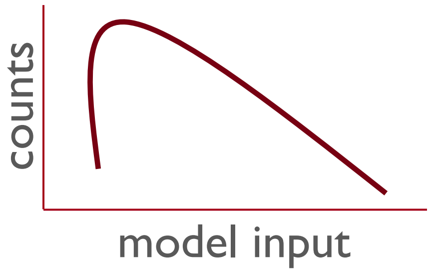
→ *accuracy is degraded on data outside of the statistical distribution of the training data*

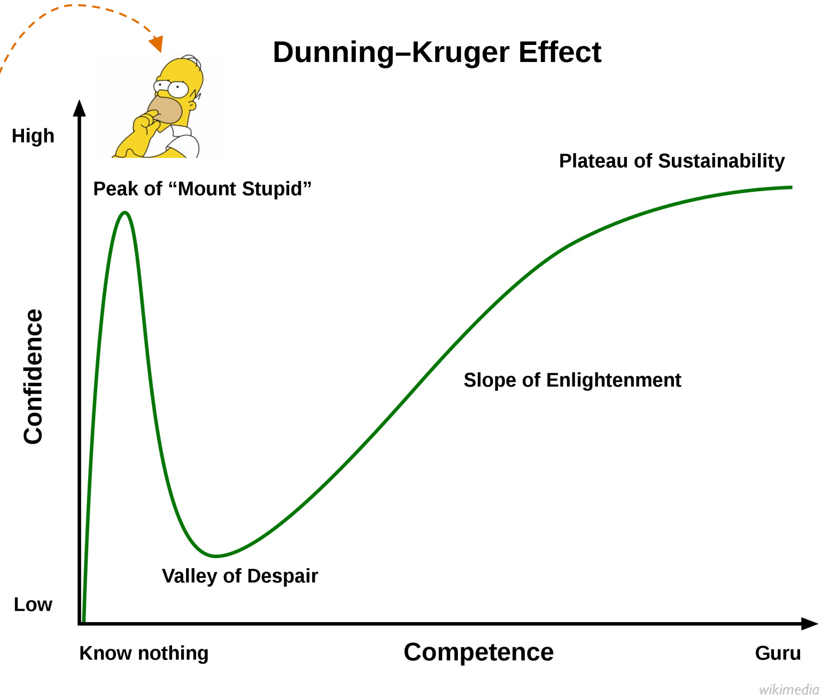
→ ***many ML approaches don't consider uncertainty estimates***

training set



new conditions



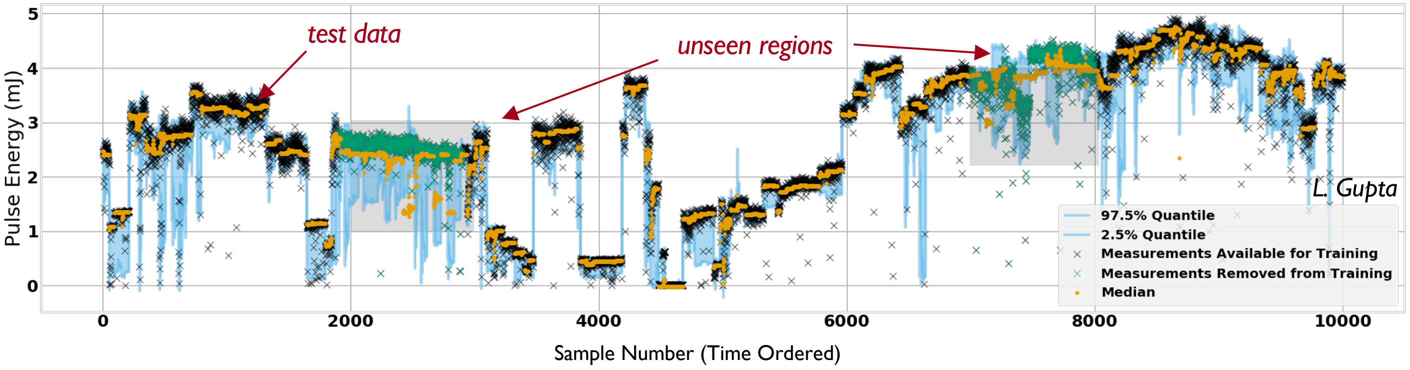


Want to have a reliable model confidence metric before using predictions in control/analysis; can also guide model updating

→ need uncertainty quantification / robust modeling

Uncertainty Quantification / Robust Modeling

Need for decision making under uncertainty (e.g. safe optimization)
Prediction uncertainties can be leveraged for online model updating, intelligent sampling



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

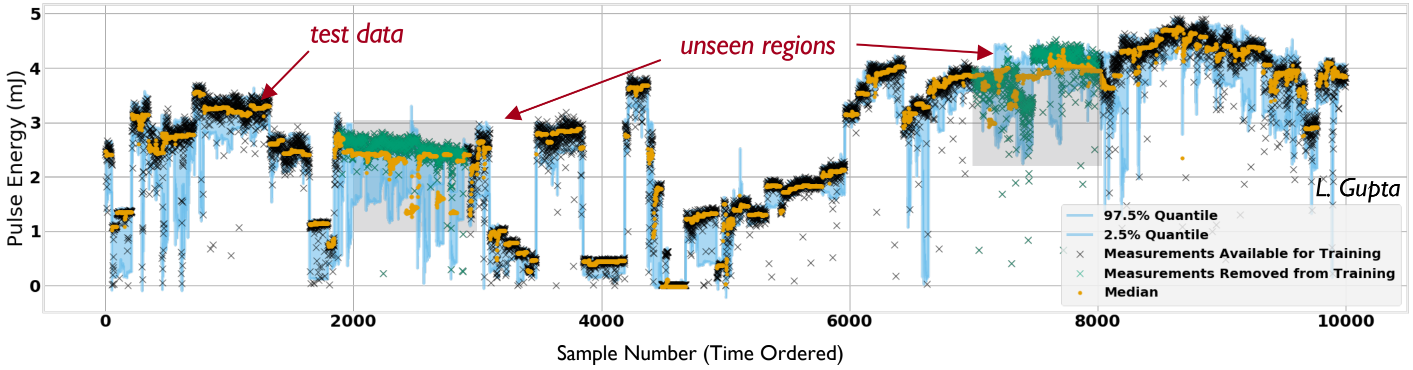
Neural network with quantile regression predicting FEL pulse energy at LCLS

https://github.com/lipigupta/FEL-UQ/blob/main/notebooks/QR--Interp-2.ipynb

Uncertainty Quantification / Robust Modeling

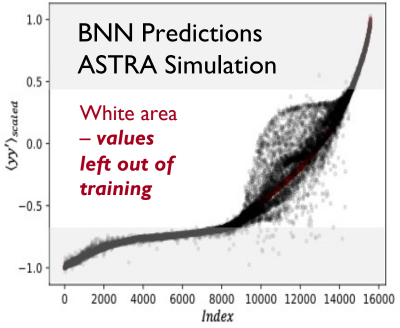
Need for decision making under uncertainty (e.g. safe optimization)
Prediction uncertainties can be leveraged for online model updating, intelligent sampling

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



Neural network with quantile regression predicting FEL pulse energy at LCLS

https://github.com/lipigupta/FEL-UQ/blob/main/notebooks/QR--Interp-2.ipynb



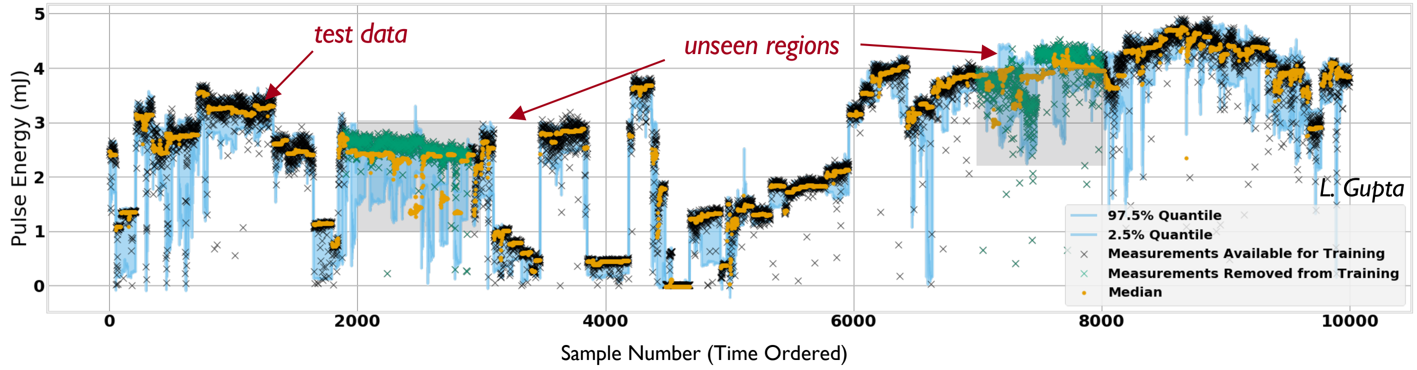
Scalar parameters for the LCLS-II injector (Bayesian neural network)

A. Mishra et. al., PRAB, 2021

Uncertainty Quantification / Robust Modeling

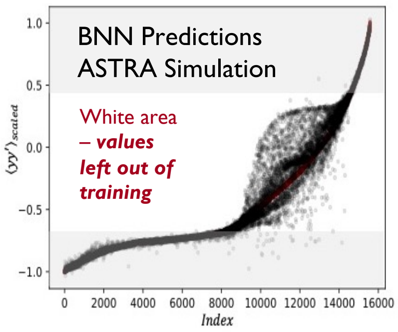
Need for decision making under uncertainty (e.g. safe optimization)
Prediction uncertainties can be leveraged for online model updating, intelligent sampling

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



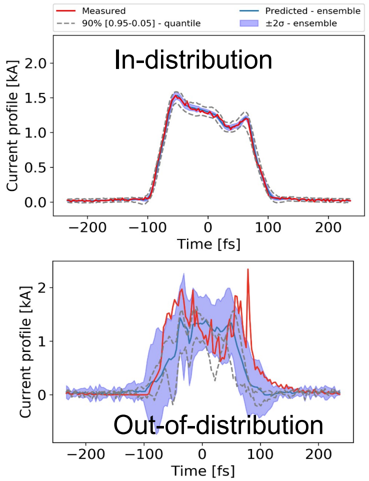
Neural network with quantile regression predicting FEL pulse energy at LCLS

https://github.com/lipigupta/FEL-UQ/blob/main/notebooks/QR--Interp-2.ipynb



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

A. Mishra et. al., PRAB, 2021



LCLS longitudinal phase space

(quantile regression + ensemble)

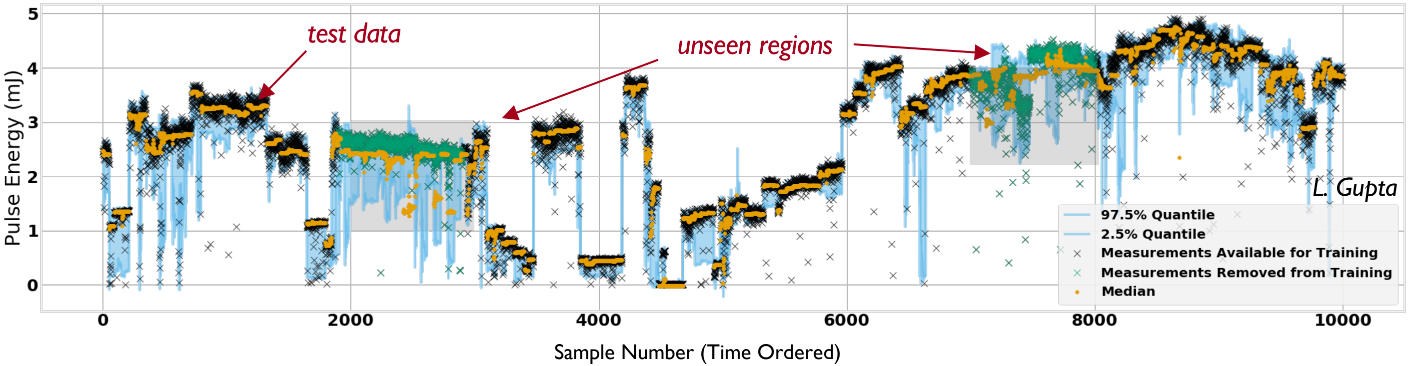
see A. Hanuka talk tomorrow morning, TUIXGD1

O. Convery, et al., PRAB, 2021

Uncertainty Quantification / Robust Modeling

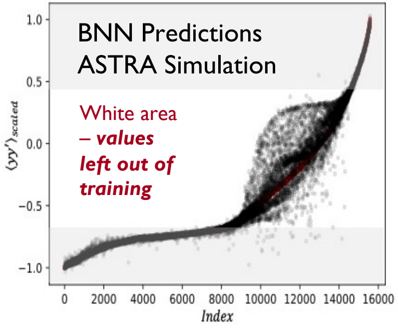
Need for decision making under uncertainty (e.g. safe optimization)
Prediction uncertainties can be leveraged for online model updating, intelligent sampling

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



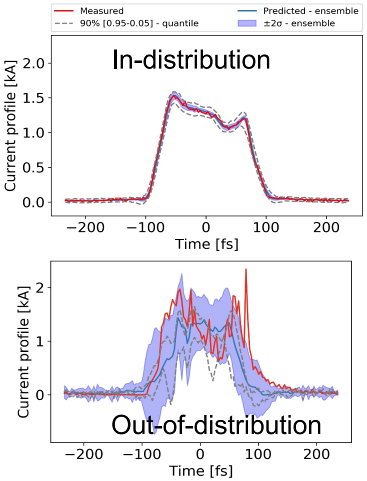
Neural network with quantile regression predicting FEL pulse energy at LCLS

https://github.com/lipigupta/FEL-UQ/blob/main/notebooks/QR--Interp-2.ipynb



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

A. Mishra et. al., PRAB, 2021

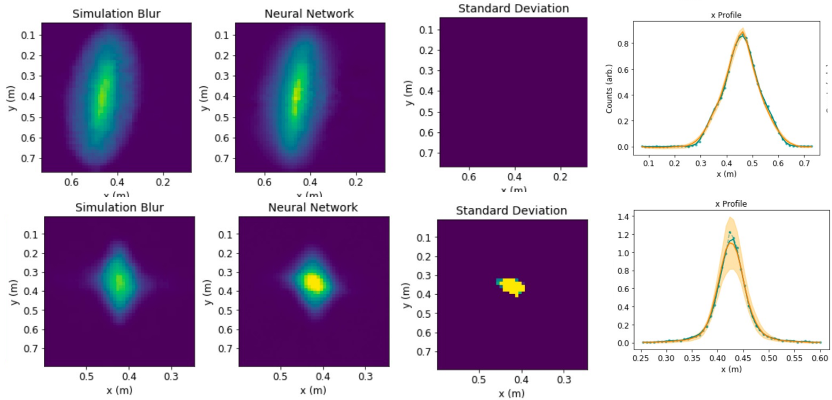


LCLS longitudinal phase space

(quantile regression + ensemble)

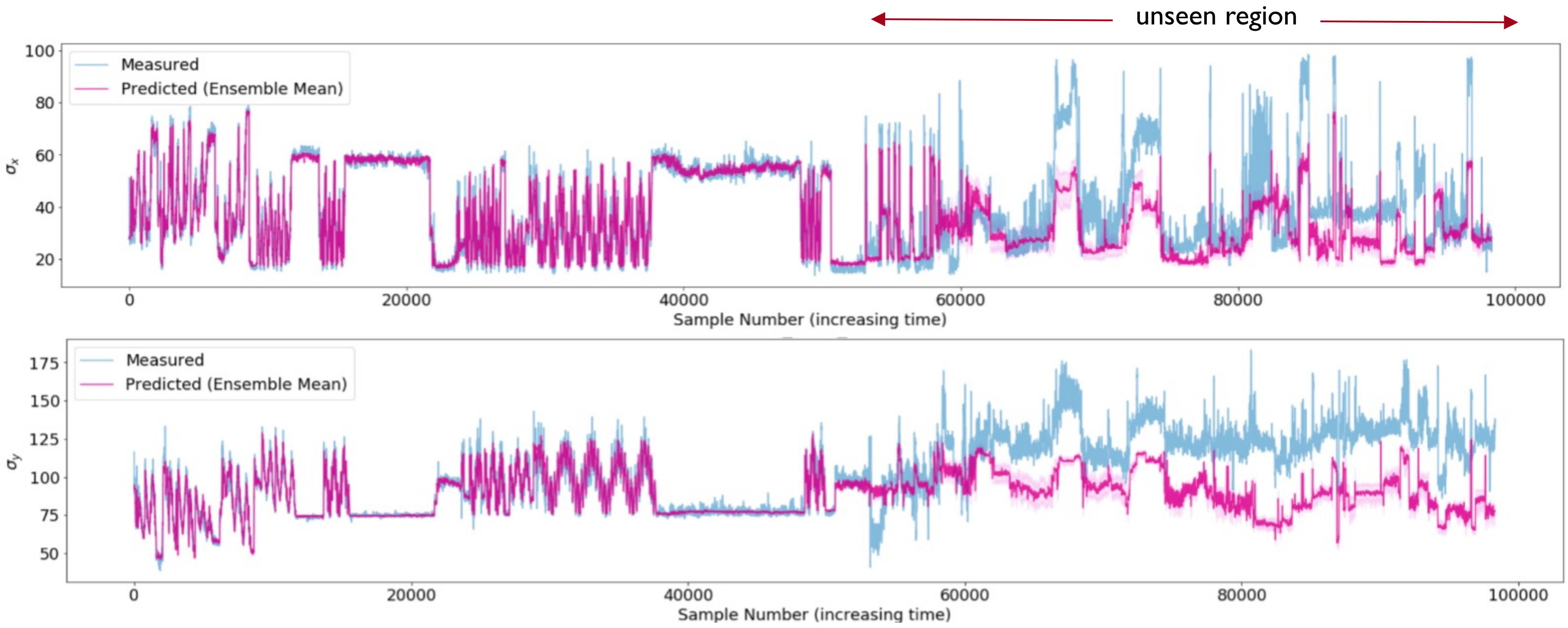
see A. Hanuka talk tomorrow morning, TUIXGD1

O. Convery, et al., PRAB, 2021



LCLS injector transverse phase space (NN ensemble)

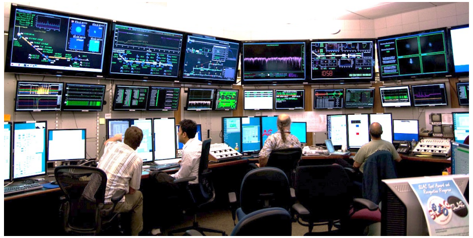
Example of beam size prediction and uncertainty estimates under drift from a neural network (@ UCLA Pegasus)



Uncertainty estimate from neural network ensemble does not cover the OOD prediction error, but it does give a qualitative metric for relative uncertainty

Data sets also present a challenge:

- Most examples above used thousands to tens-of-thousands of examples
- Not feasible to gather new data in every configuration *(from simulation or measurements)*
- Not everyone has access to large compute resources or ample beam time



→ how can we increase model generalization to new conditions and decrease data set sizes (i.e. improve sample-efficiency)?

→ inherent question: how to make ML models more readily adaptable?

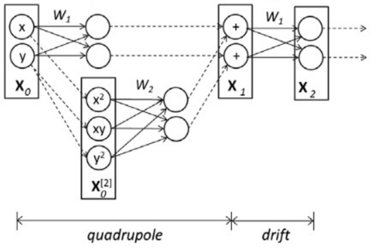
"Physics-informed" modeling → *incorporate physics domain knowledge to reduce need for data, and aid interpretability + generalization*

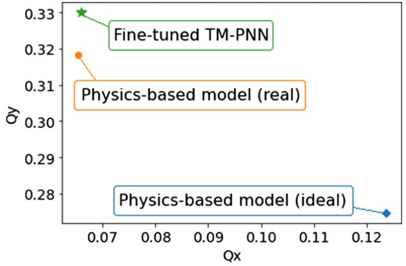
Many approaches:

- 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)*

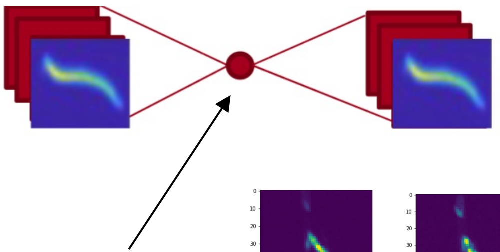
Review paper: Karniadakis et al, *Nat Rev Phys* **3,** 422–440 (2021)
Snowmass accelerator modeling white paper: arXiv:2203.08335

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



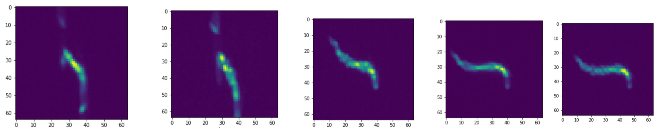


Physics-driven representation learning
(e.g. encoder-decoder neural network models)

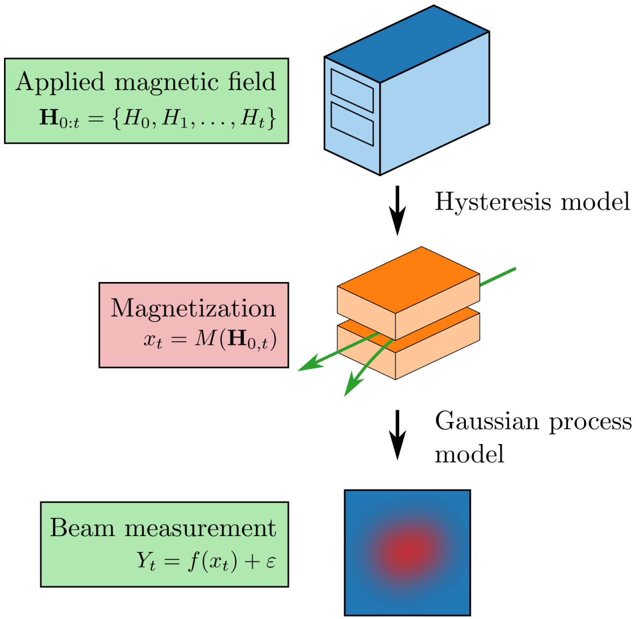


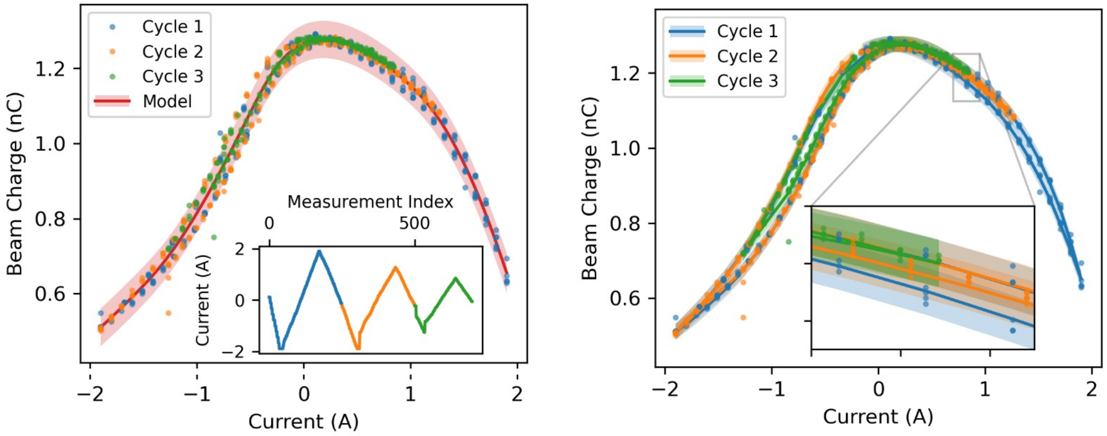
Many examples in this conference, including talks tomorrow: TUIXGD1, TUOXGD3

useful latent space

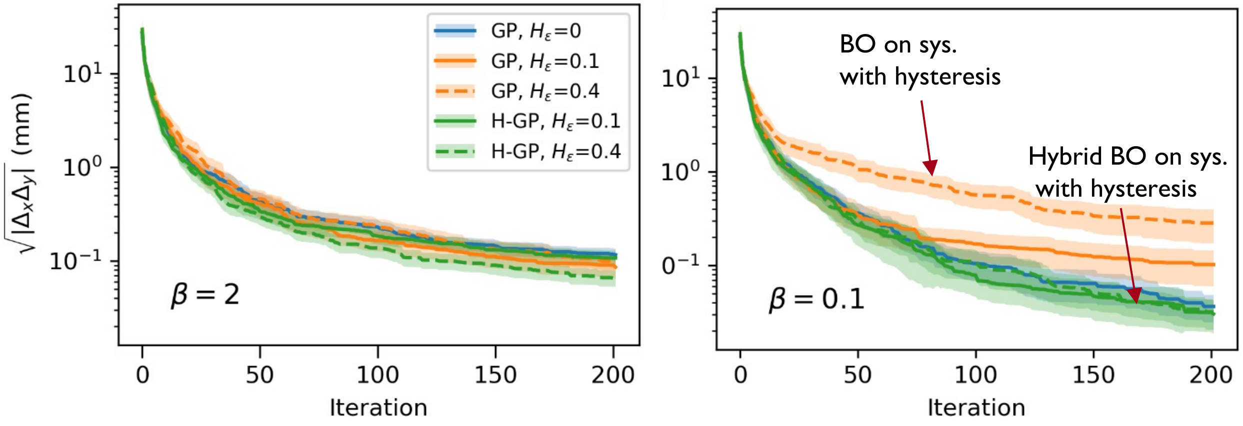


Example: Differentiable Hysteresis Modeling + ML





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



Higher-precision optimization possible when including hysteresis

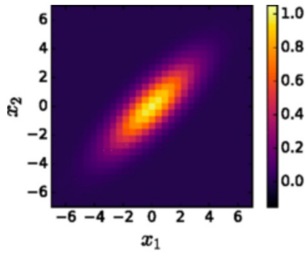
Poster: TUPOST061

R. Roussel, et al., PRL, 2022

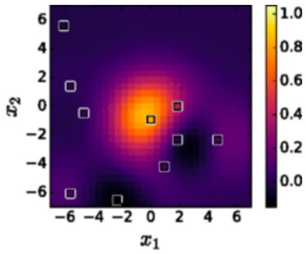
Example: Physics-informed Gaussian Processes

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

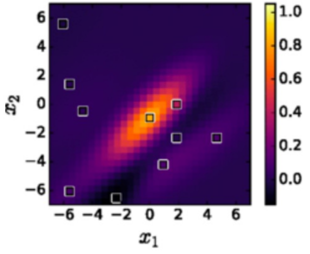
→ design GP kernel from expected correlations between inputs (e.g. quads)



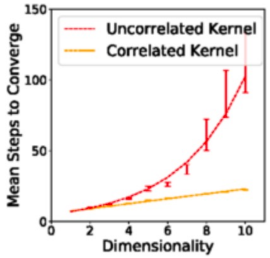
(a) Ground truth

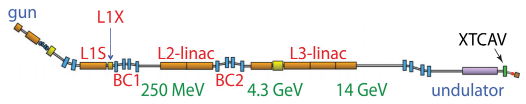


(b) Isotropic kernel



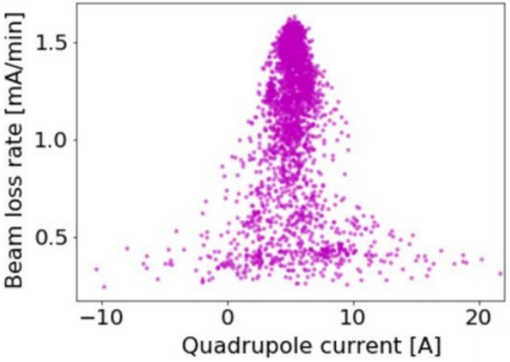
(c) Correlated kernel

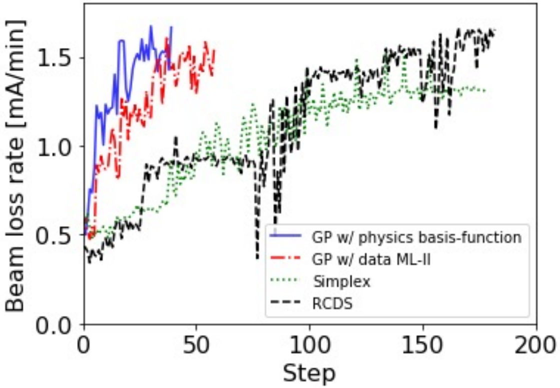


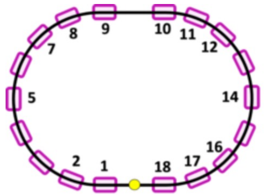


FEL tuning @LCLS

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







vertical emittance tuning @SPEAR3

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

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

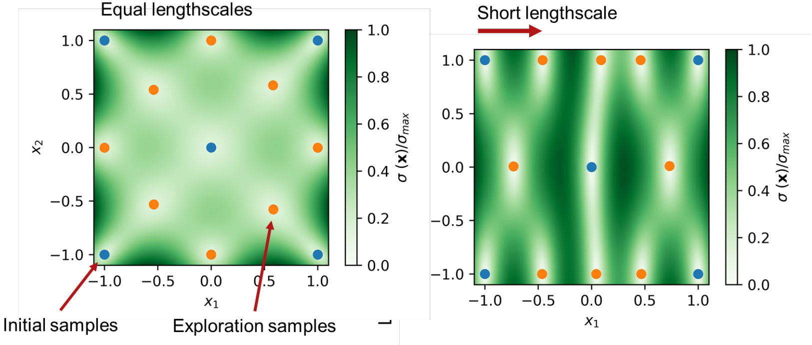
Better Data Sampling: Bayesian Exploration

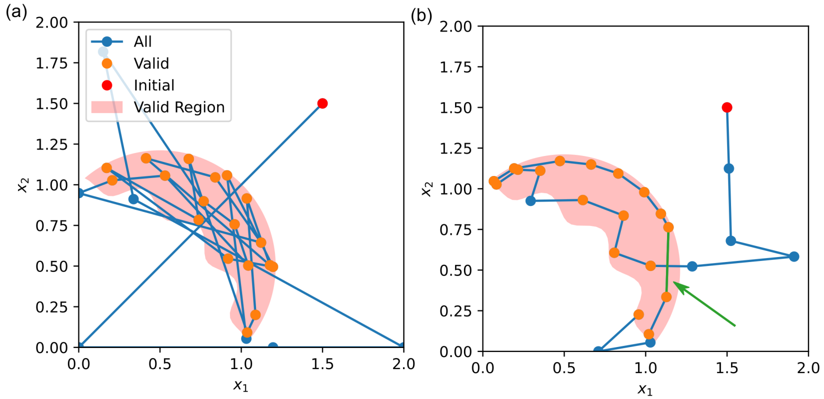
R. Roussel et. al.
Nat. Comm. **2021**

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

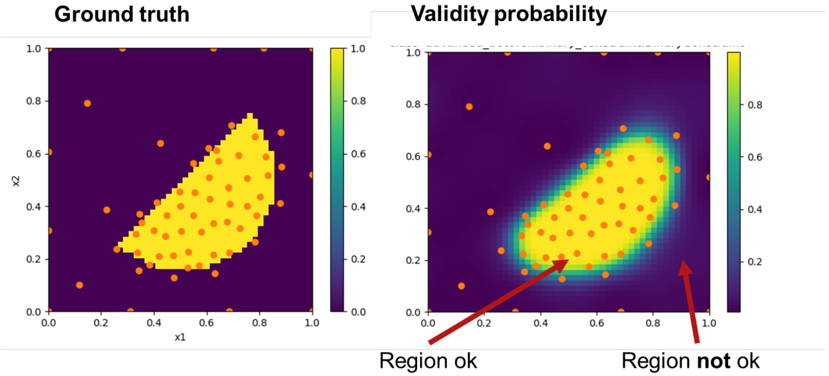
proximal biasing

adaptive sampling

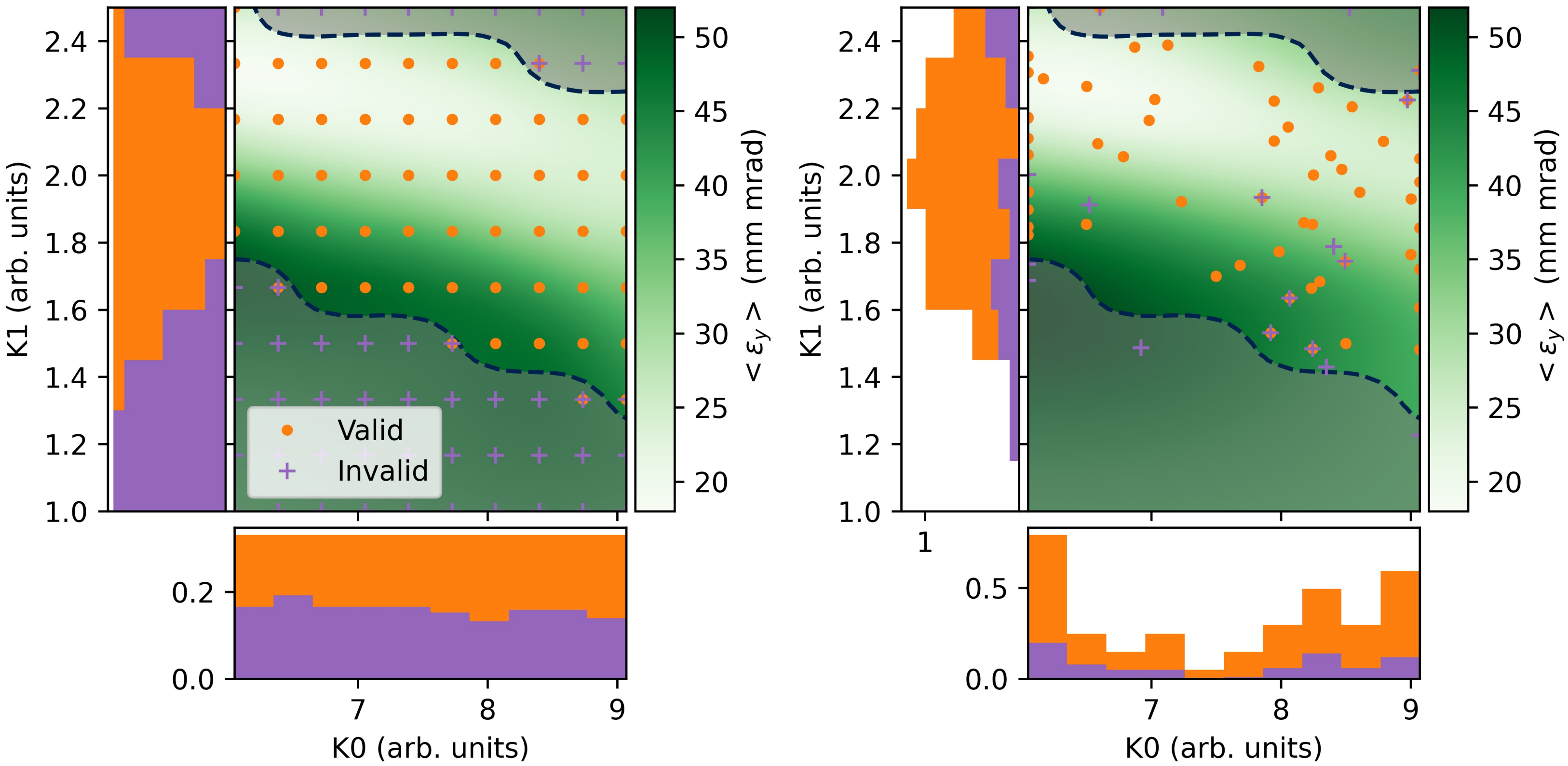




learning constraints



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



Example for photoinjector emittance at AWA
→ much more efficient sampling than N-D scans

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

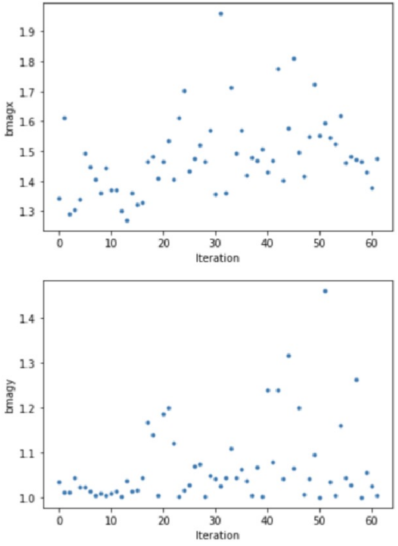
Explored 10-D input space on FACET-II injector at 700pC bunch charge

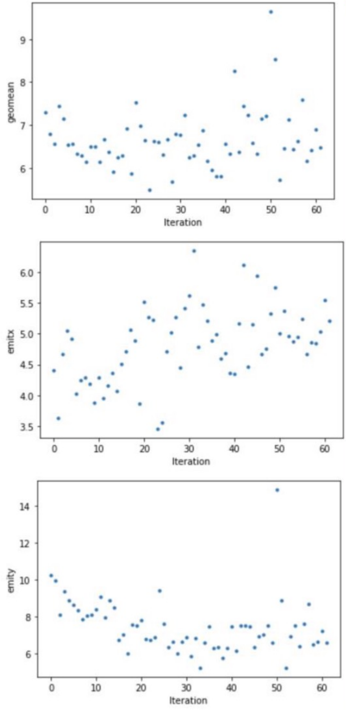
- Inputs: solenoid, bucking coil, corrector quads, matching quads
- Constrained on match and emittance
- Data sampling enabled easy model learning

~2 hours for thorough exploration in 10-D contrast with 8-12 hours for 3-D scan

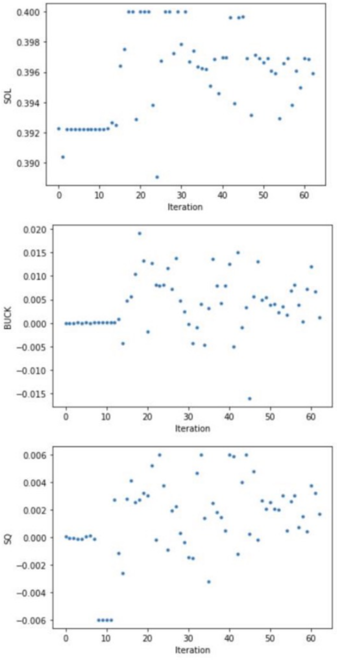
Constrained Outputs

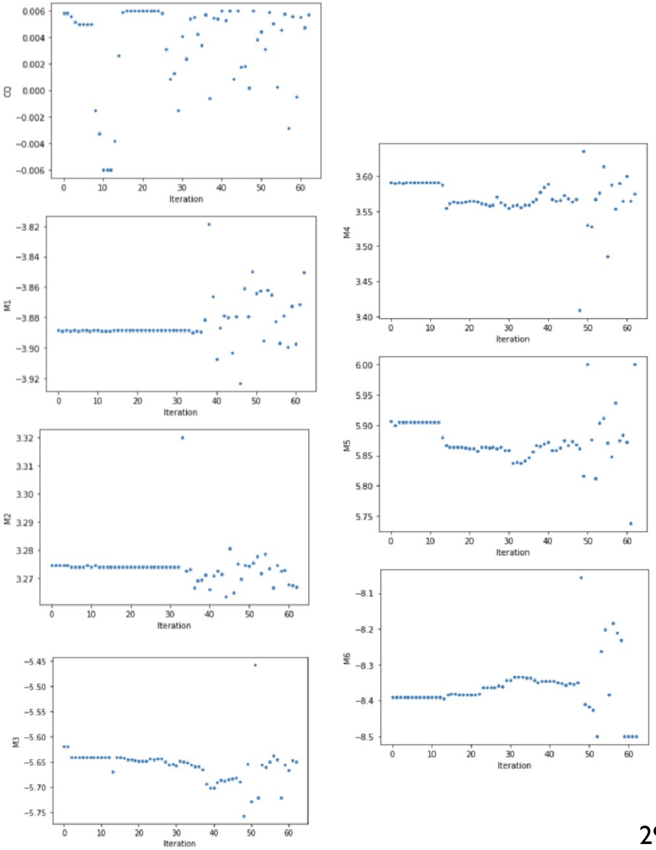
Emit < 10
Bmag < 2





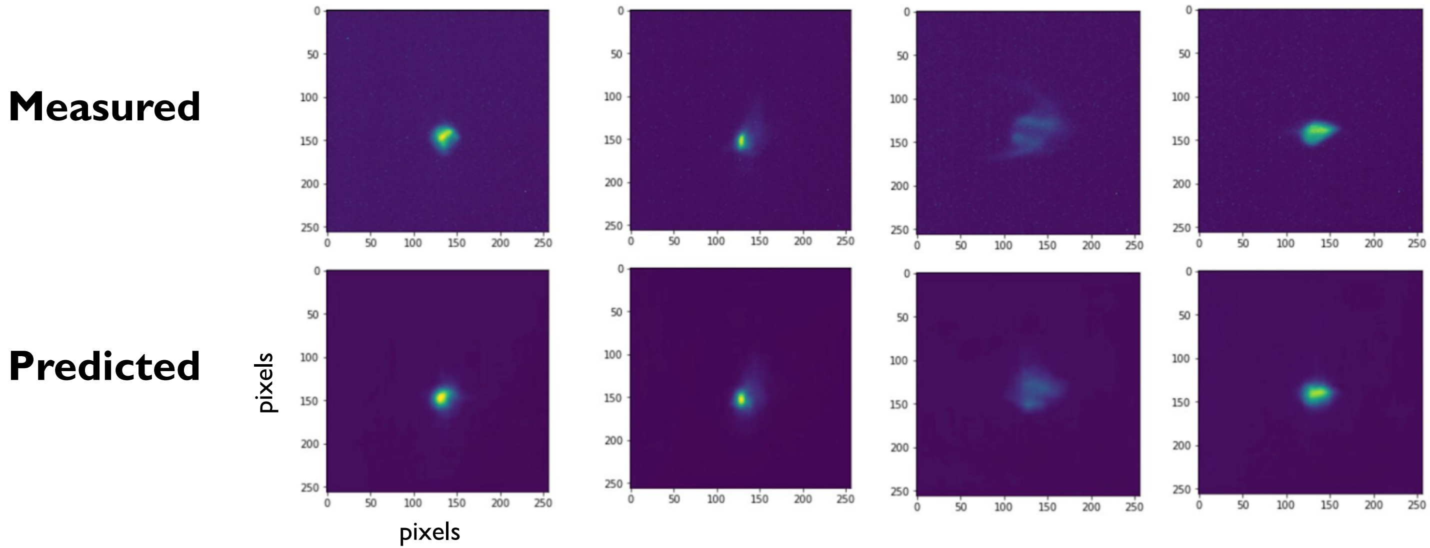
Variables





Explored 10-D input space on FACET-II injector at 700pC bunch charge

- Inputs: solenoid, bucking coil, corrector quads, matching quads
- Constrained on match and emittance
- **Data sampling enabled easy model learning**



Examples from test set of held-out input ranges

Use of Bayesian exploration to generate training data was **sample-efficient**, reduces some of the burden of data cleaning, and results in a **well-balanced distribution for the training data** set over relevant space

→ Each area aids creation of generalizable, adaptable accelerator models

Better Model Representations

Physics-informed Modeling

Generalizable Learned Representations

Model Uncertainty Assessment

Robust Modeling / Uncertainty Quantification

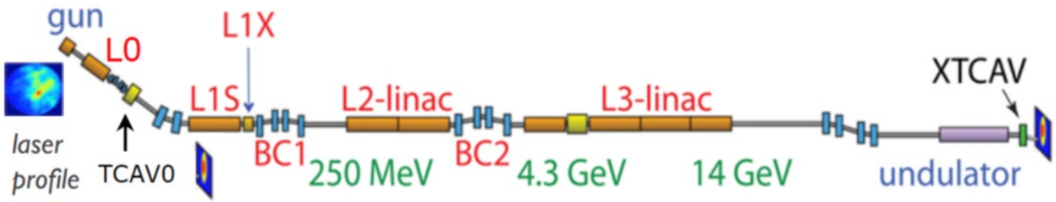
Online Model Updating

Efficient Sampling Methods *(active learning)*

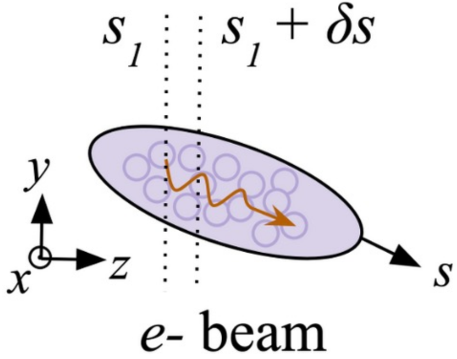
Continual Learning

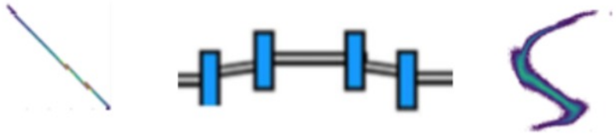
Adaptive Feedback

Surrogate Models of Different Granularities



sub-section models (e.g. injector)
machine-wide models





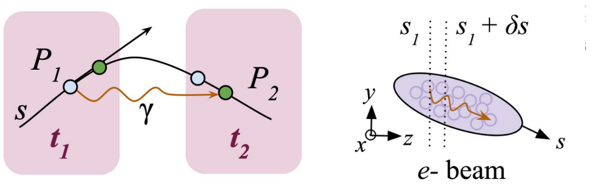
general modular components

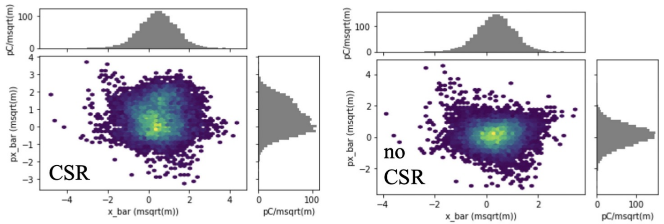
multi-particle tracking steps

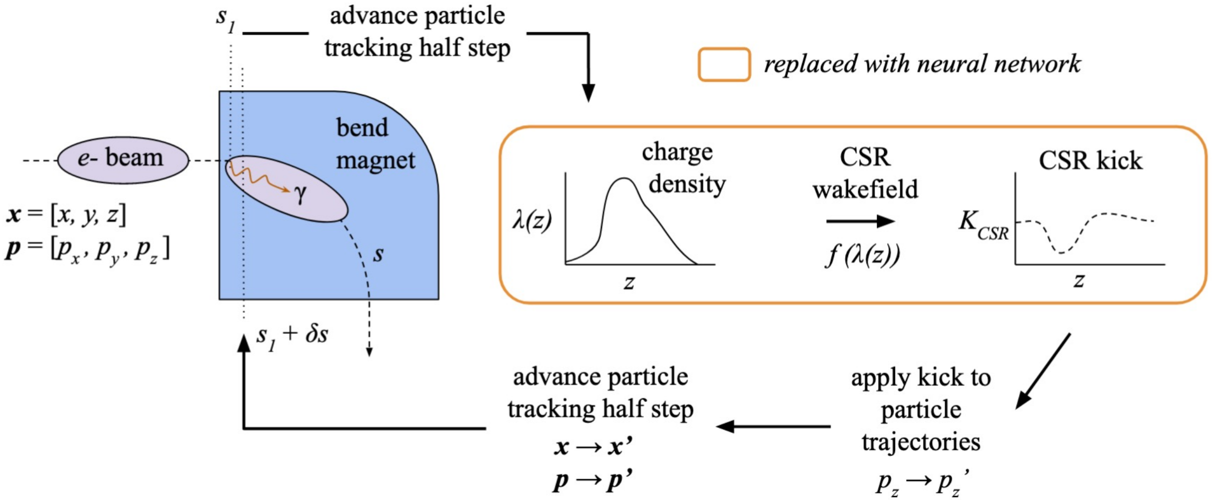
Embedding surrogates in tracking calculations

Impact of Coherent Synchrotron Radiation (CSR) is computationally intensive to simulate, even for 1D

Replace wakefield calculation in tracking step with a neural network







Trained fully-connected, feed-forward network

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

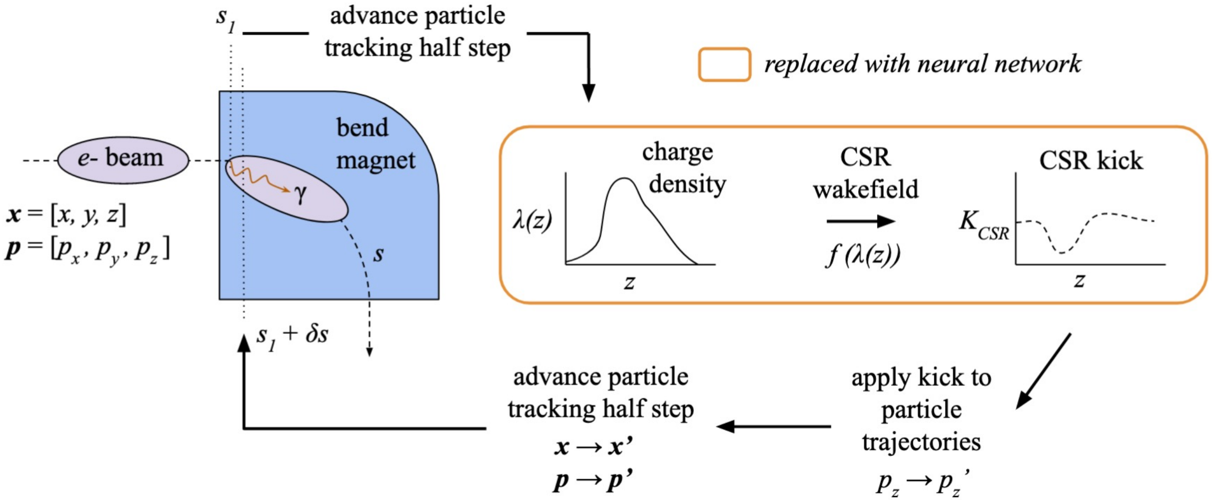
Embedding surrogates in tracking calculations

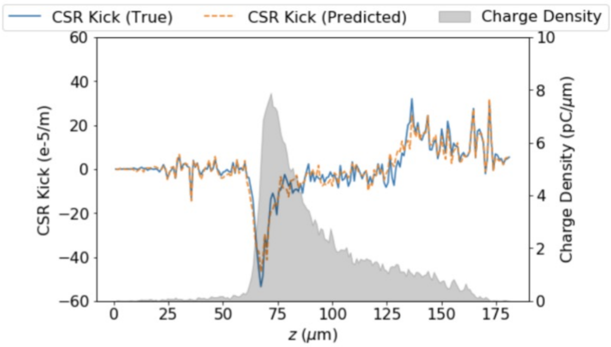
Impact of Coherent Synchrotron Radiation (CSR) is computationally intensive to simulate, even for 1D

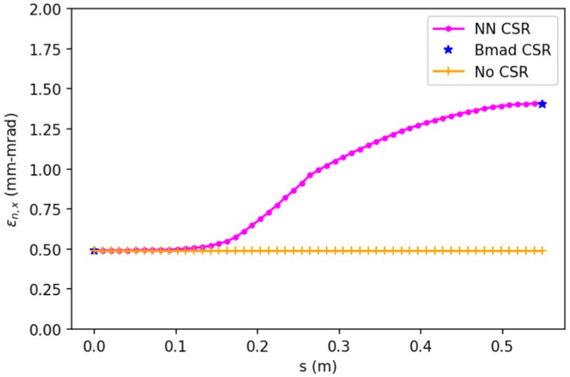
Replace wakefield calculation in tracking step with a neural network

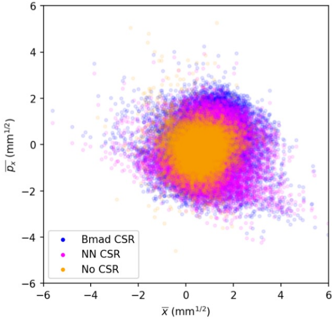
→ *not perfect, but gets the bulk effect (better than excluding CSR)*

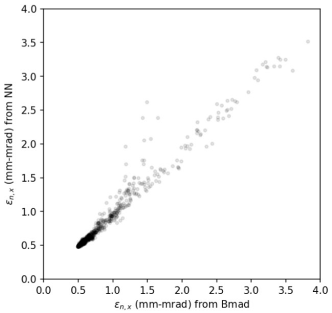
→ *is 10X faster than running with 1D CSR routine*









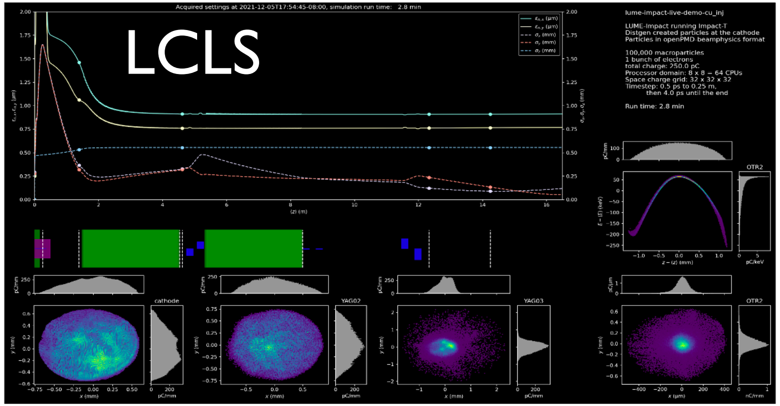


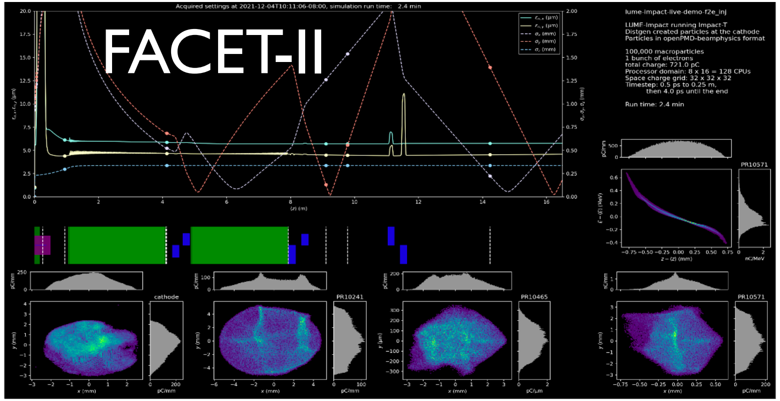
ML and Online Multi-Particle Physics Simulations

Getting easier to run physics sims that include nonlinear collective effects in online / semi-online execution when coupled with HPC

→ *opens up new opportunities for physics-constrained learning*

Standard interfaces and software (e.g. LUME, openPMD) make this more readily extensible to new systems





Impact-T simulations running online at SLAC

Standards for easy interfacing of simulations and optimizers

CNSGA, Bayesian algorithms, sampler
https://christophermayes.github.io/Xopt/index.html

Optimizer

Simulation

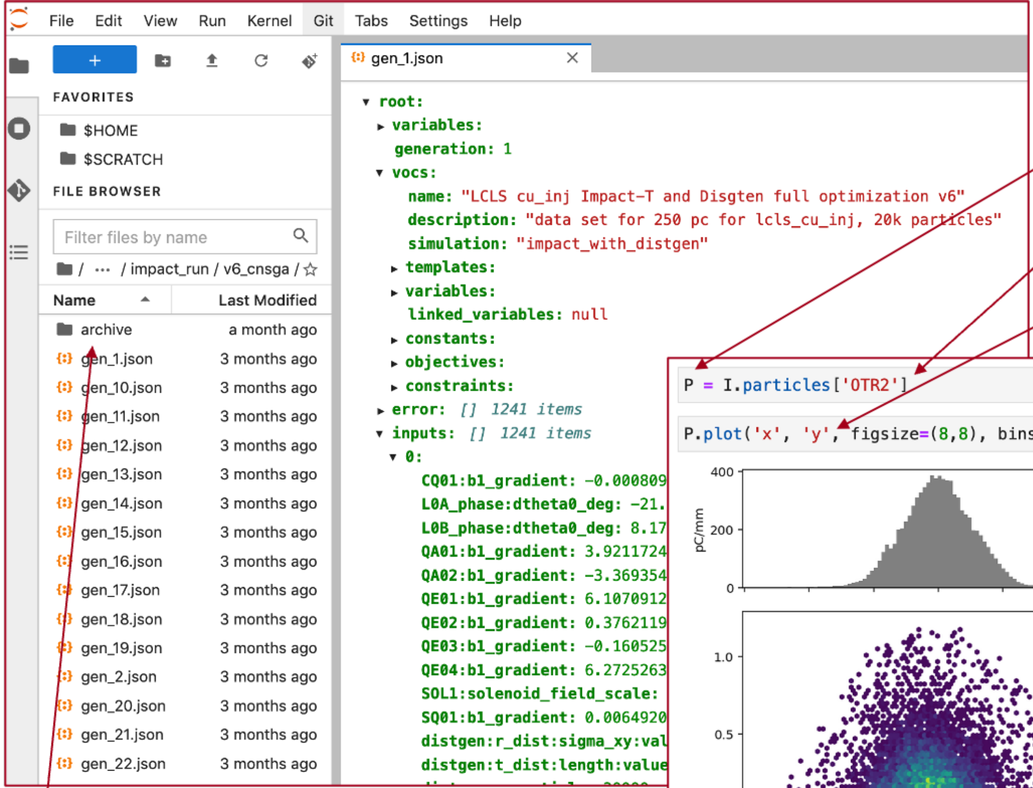
LUME – light source unified modeling environment
https://www.lume.science/

Data Set

Impact
ASTRA
GPT
Bmad
Genesis
SRW
work in progress:
elegant

gen_1.json

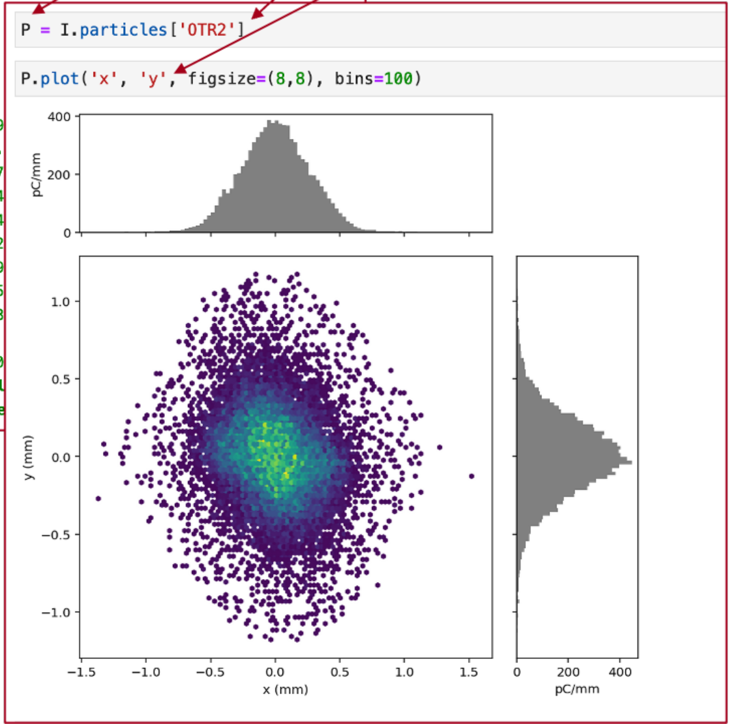
```
▼ root:
  ▸ variables:
    generation: 1
  ▸ vocs:
  ▸ error:  []  1241 items
  ▸ inputs:  []  1241 items
  ▸ outputs:  []  1241 items
```



particle group

location

select projection to plot



h5 files with beam distributions

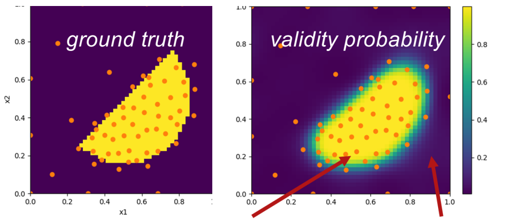
→ easy to use with open-pmd-beamphysics

https://github.com/ChristopherMayes/openPMD-beamphysics

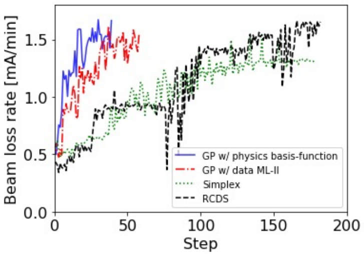
Future directions for ML-based modeling, physics modeling, and optimization/characterization are tightly-linked

Algorithms for efficient optimization and characterization *(useful for simulation exploration/design, data generation, machine characterization)*

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

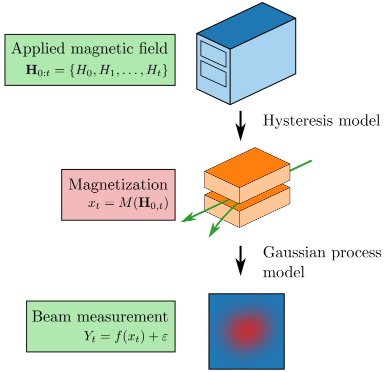


Sampling constraints learned on-the-fly

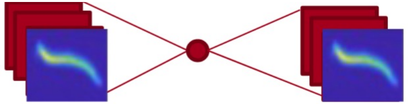


Hanuka et. al. PRAB , **2021**

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

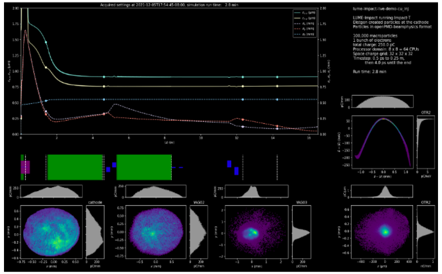


Roussel et. al. *PRL*. **2022**

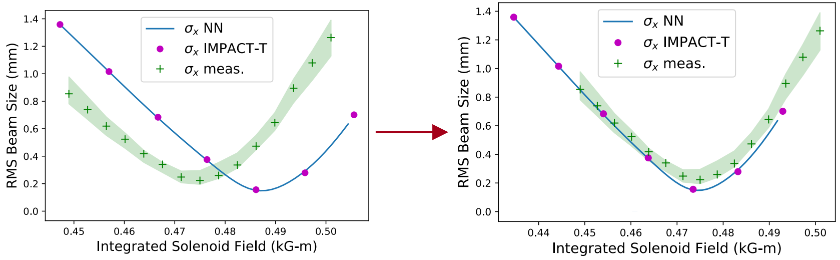


Representation learning

Online physics simulations



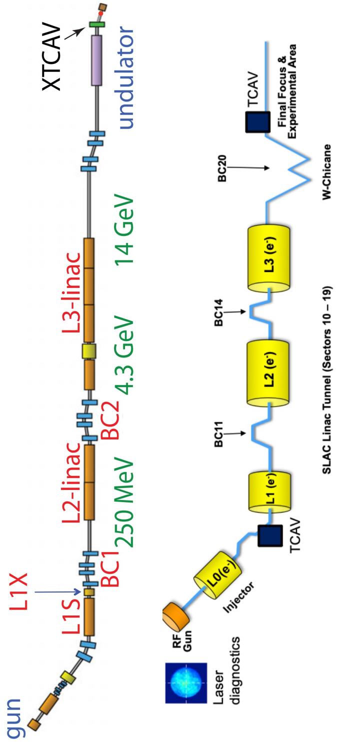
Adaptation on top of core models

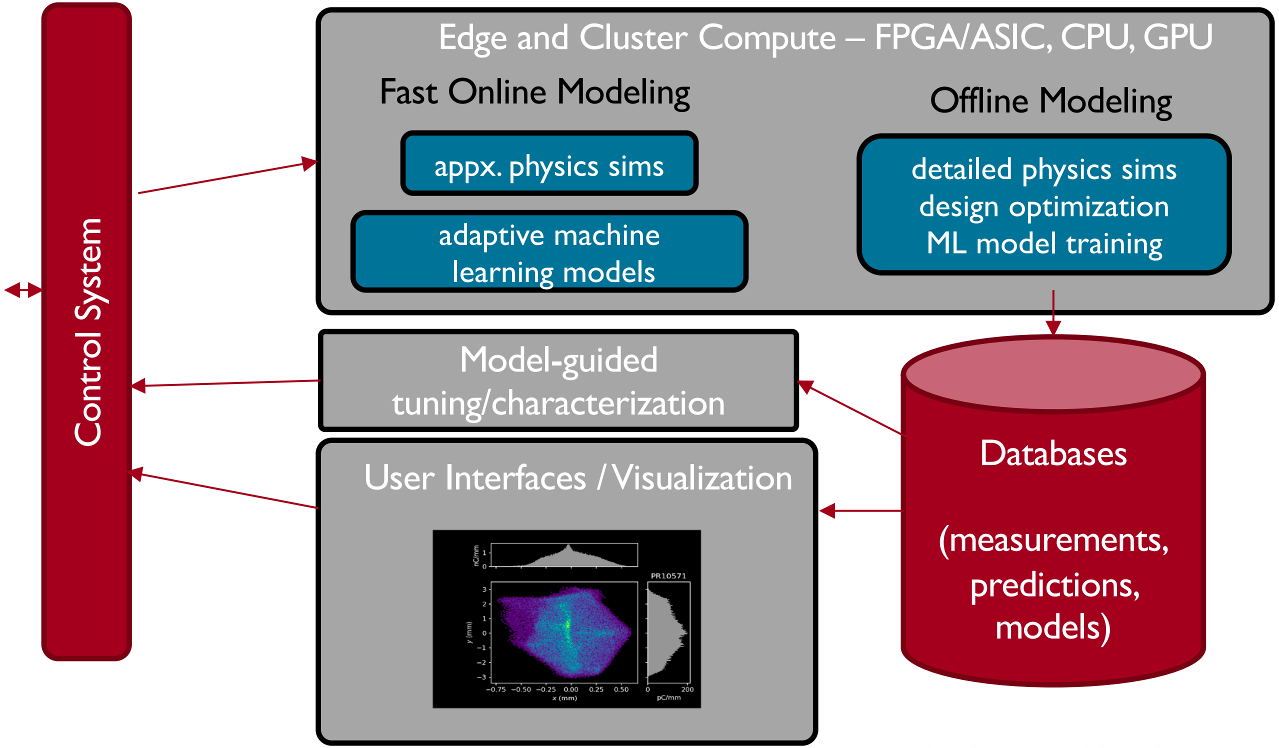


Software packages and standards for data generation, online deployment of models, and optimization (*LUME, xopt, Badger*)

Xopt

A common dream: fully-integrated virtual accelerator





Encourage checking out the Snowmass accelerator modeling whitepaper: [arXiv:2203.08335]

Snowmass21 Accelerator Modeling Community White Paper

by the Beam and Accelerator Modeling Interest Group (BAMIG)*

Authors (alphabetical): S. Biedron[13], L. Brouwer[1], D.L. Bruhwiler[7], N. M. Cook[7], A. L. Edelen[6], D. Filippetto[1], C.-K. Huang[9], A. Huebl[1], N. Kuklev[4], R. Lehe[1], S. Lund[12], C. Messe[1], W. Mori[10], C.-K. Ng[6], D. Perez[9], P. Piot[4,5], J. Qiang[1], R. Roussel[6], D. Sagan[2], A. Sahai[11], A. Scheinker[9], E. Stern[14], F. Tsung[10], J.-L. Vay[1], D. Winklehner[8], and H. Zhang[3]

Thank you for your attention!