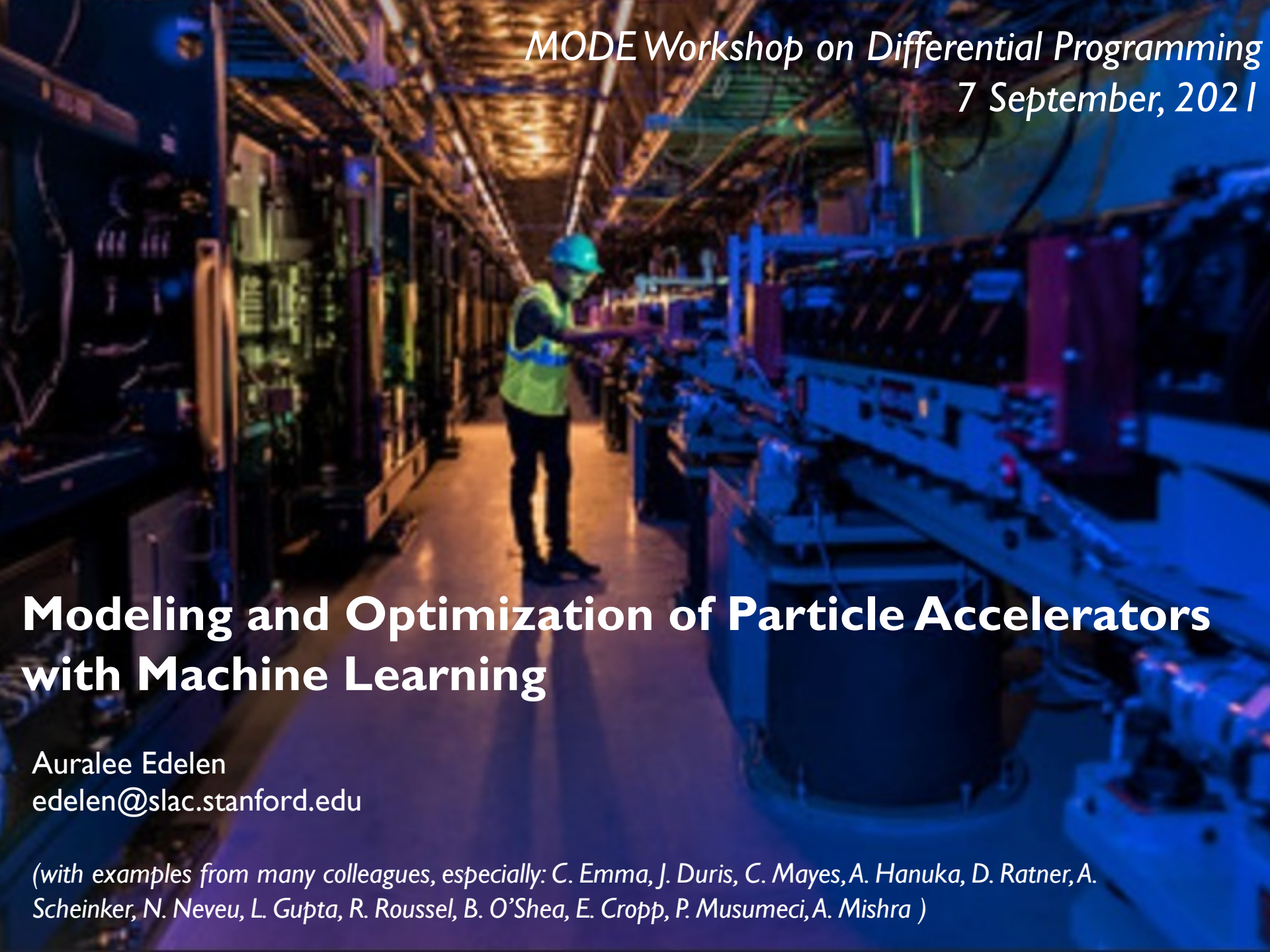


MODE Workshop on Differential Programming
7 September, 2021

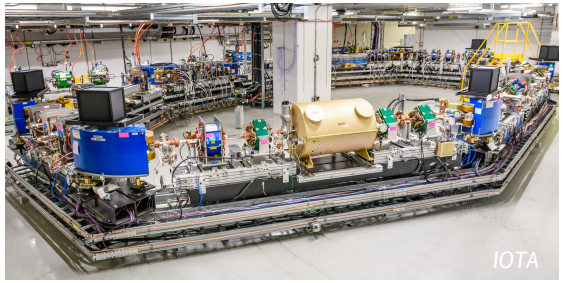
Modeling and Optimization of Particle Accelerators with Machine Learning

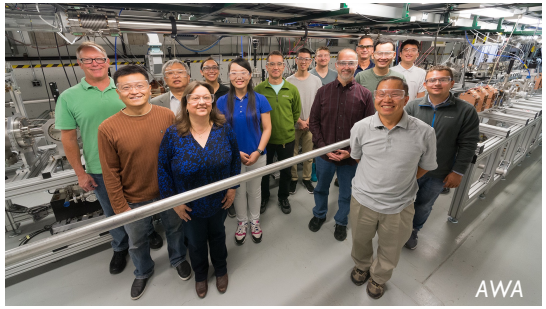
Auralee Edelen
edelen@slac.stanford.edu

(with examples from many colleagues, especially: C. Emma, J. Duris, C. Mayes, A. Hanuka, D. Ratner, A. Scheinker, N. Neveu, L. Gupta, R. Roussel, B. O'Shea, E. Cropp, P. Musumeci, A. Mishra)

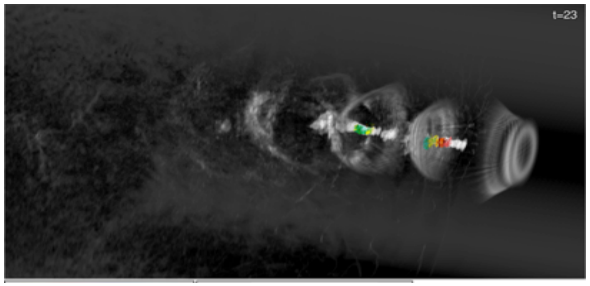
Large Scientific Facilities

Small Test Facilities



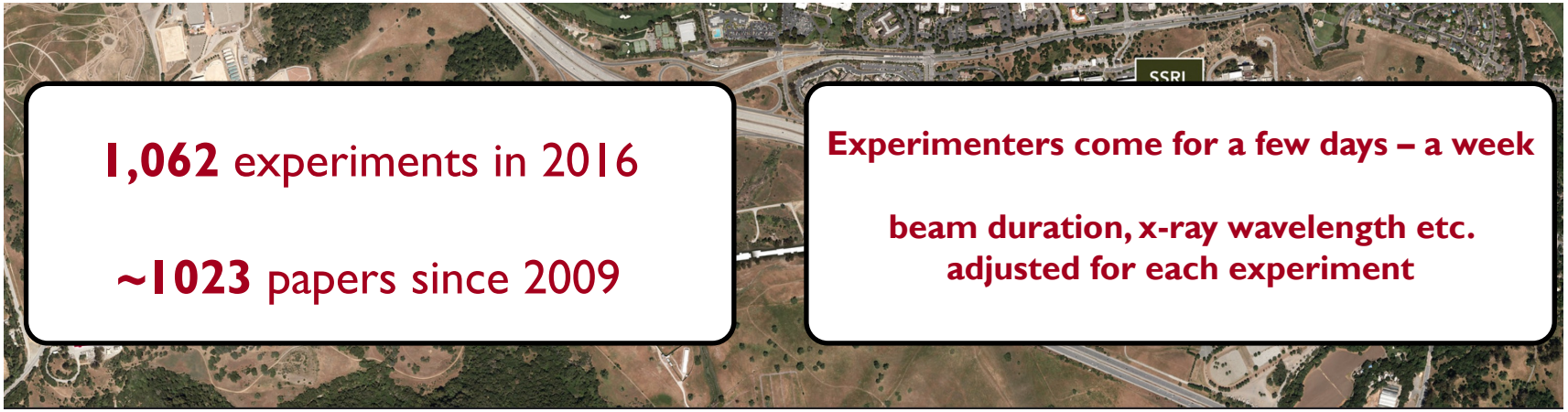


Advanced Acceleration



Industrial / Medical

Emory Proton Therapy Center



1,062 experiments in 2016

~1023 papers since 2009

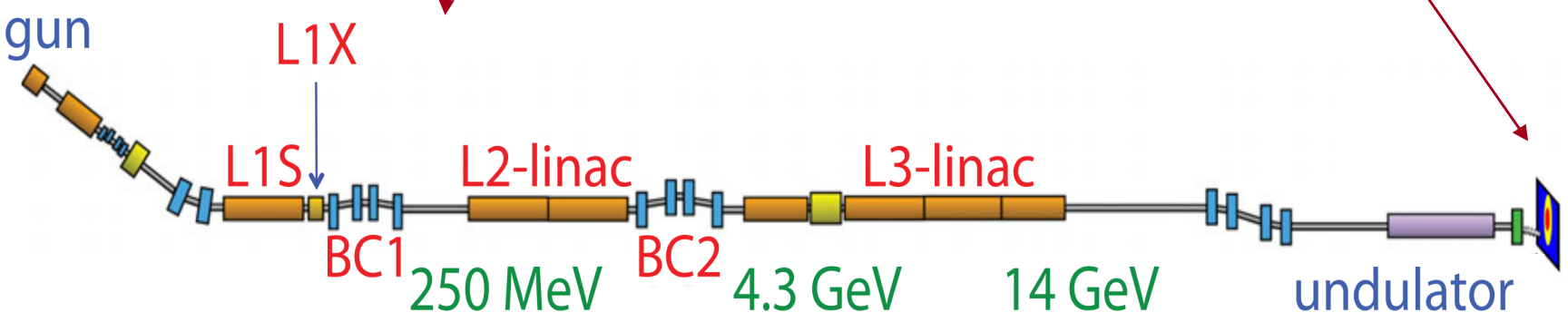
Experimenters come for a few days – a week

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

https://lcls.slac.stanford.edu

machine settings

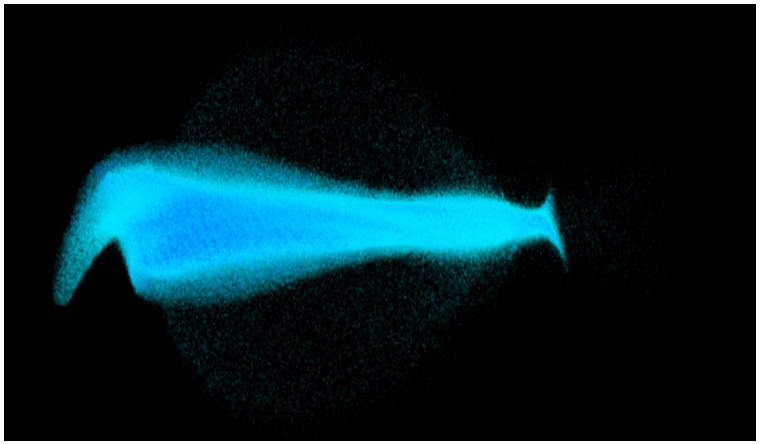
specific beam characteristics

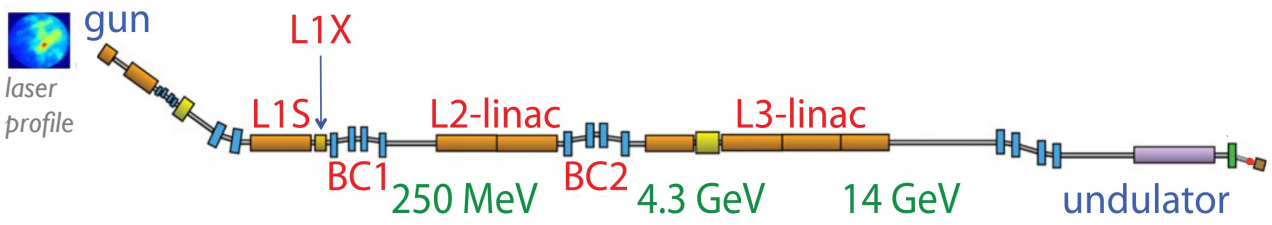


Beam exists in 6-D position-momentum phase space

Measure 2-D projections or reconstruct based on perturbations of upstream controls

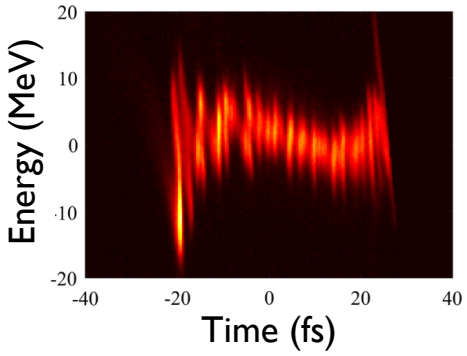
Can have dozens-to-hundreds of controllable variables and hundreds-of-thousands to measure



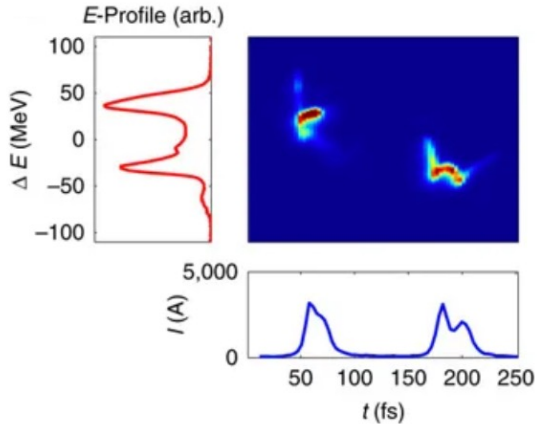


photon beam to 7 experiment stations

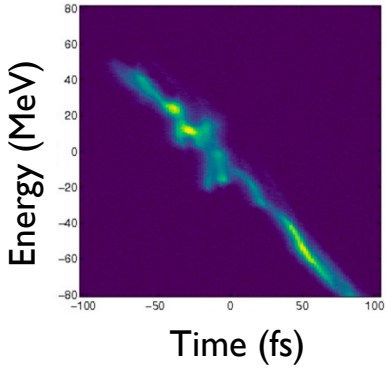
J. Qiang et al, PRAB (2017)



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



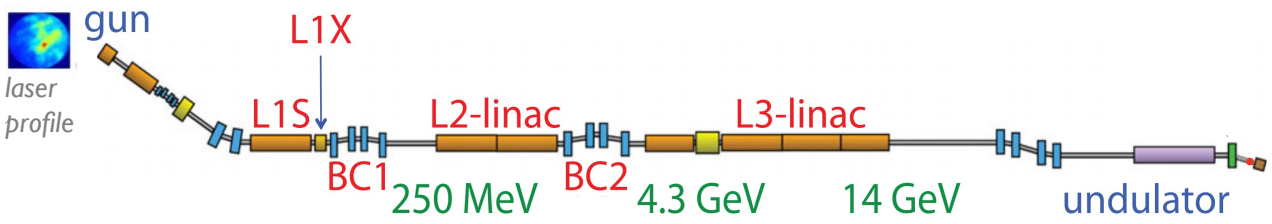
A. Marinelli, IPAC'18



~400 hours spent tuning per year

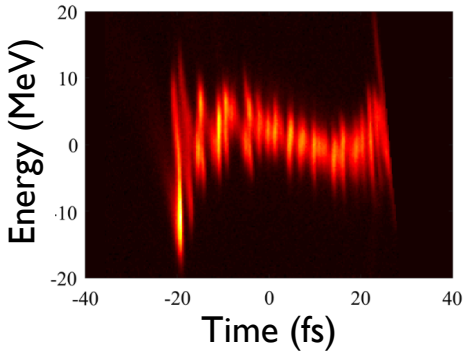
Changing configurations roughly 2-5 times per day

Average setup time is ~30 minutes

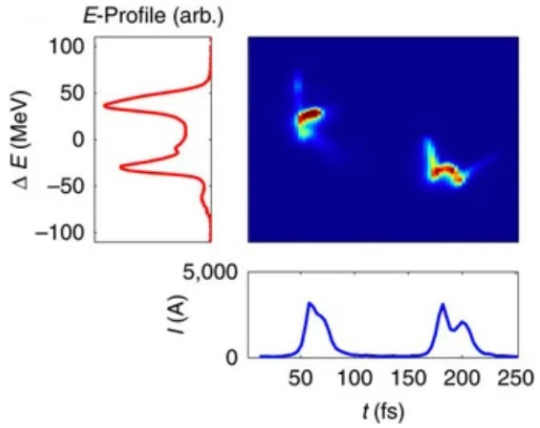


photon beam to 7 experiment stations

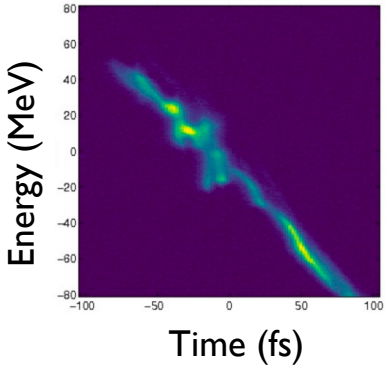
J. Qiang et al, PRAB (2017)



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

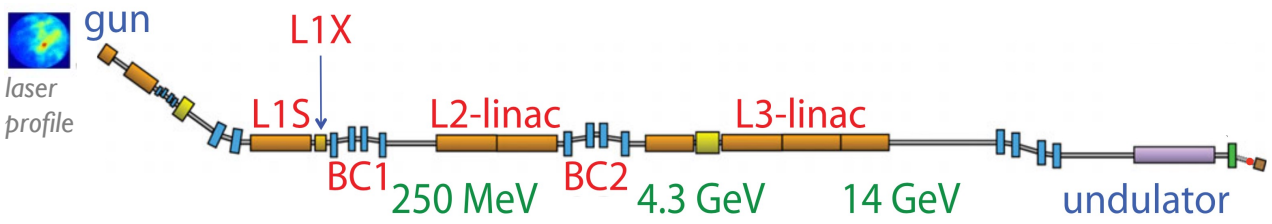


A. Marinelli, IPAC'18



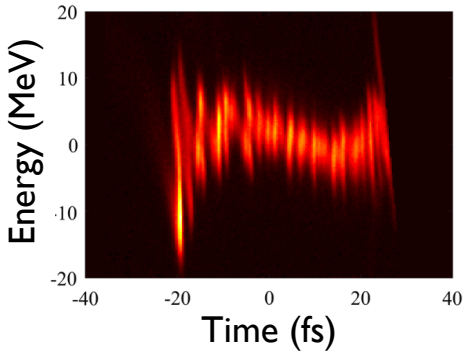
Approximate Annual Budget: $145 million
Approximate hours of experiment delivery per year: 5000
About $30k per experiment hour to run!

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

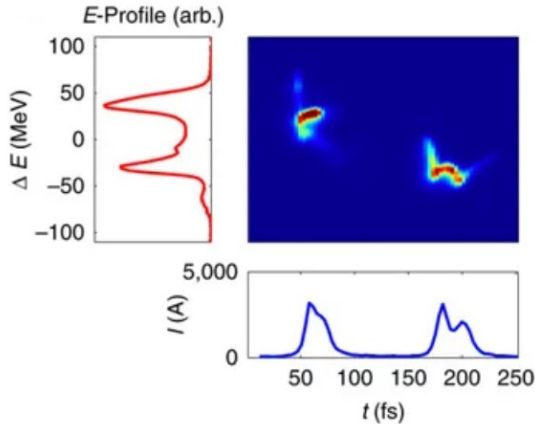


photon beam to 7 experiment stations

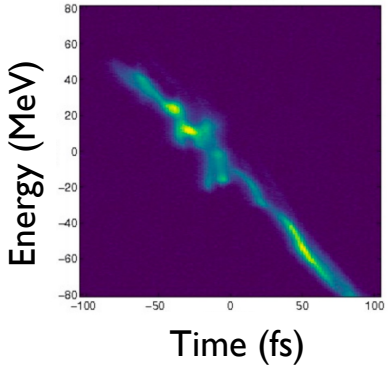
J. Qiang et al, PRAB (2017)



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

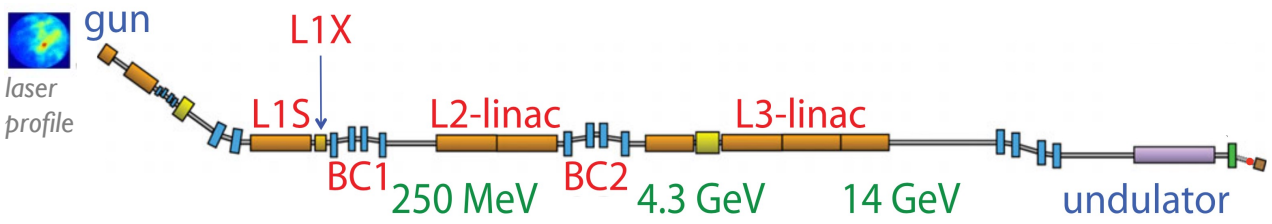


A. Marinelli, IPAC'18



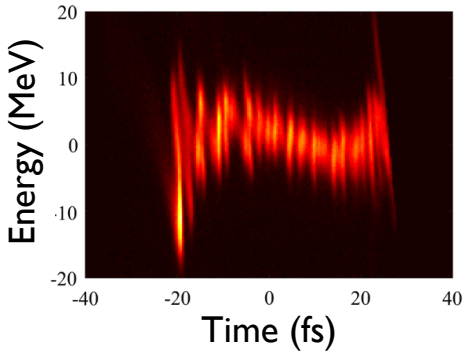
Efficient tuning matters to maximize scientific output

Achieving fundamentally higher beam quality or new beam parameters can enable new science

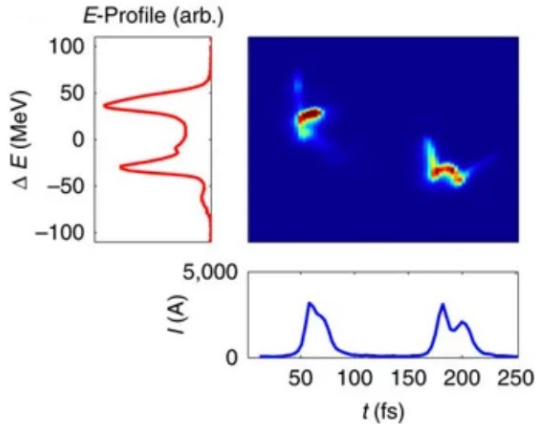


photon beam to 7 experiment stations

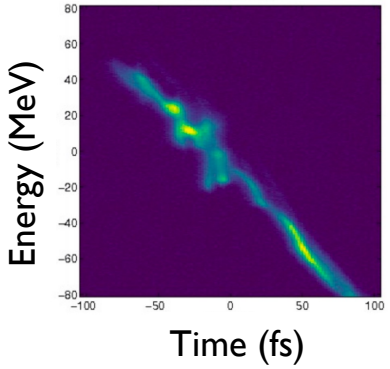
J. Qiang et al, PRAB (2017)



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



A. Marinelli, IPAC'18

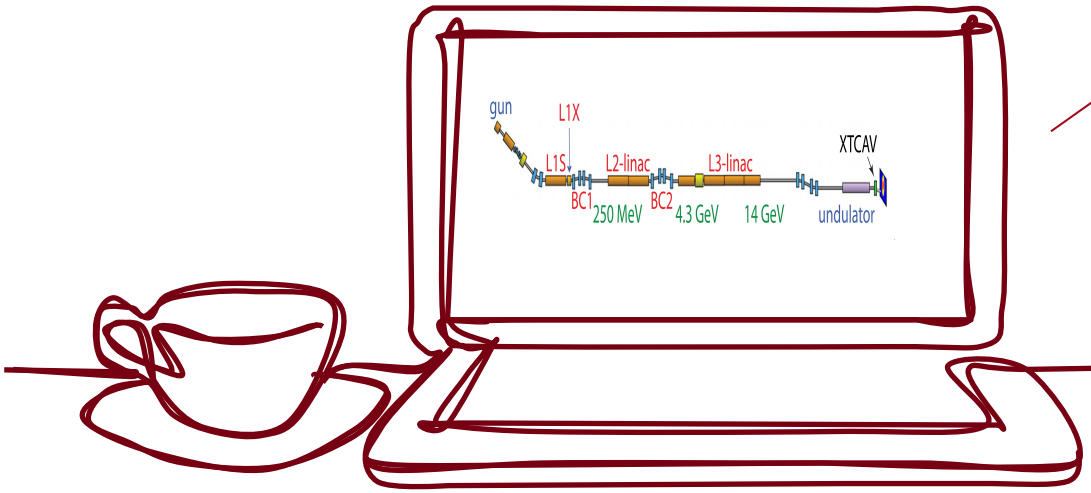


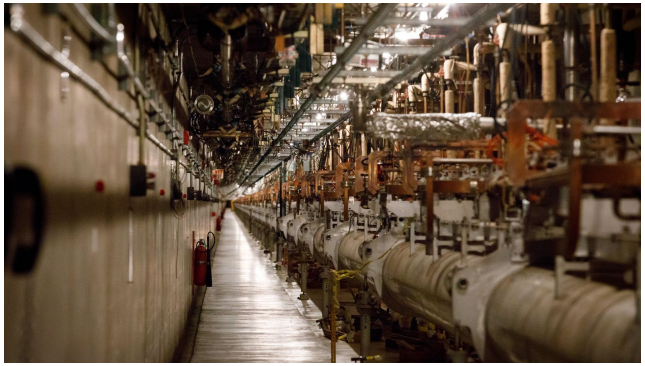
Rapid beam customization

Achieve new configurations + unprecedented beam parameters

Fine control to maintain stability within tolerances

In a perfect world…





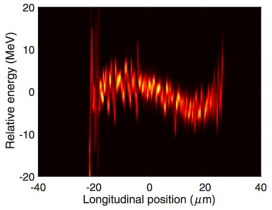
Use a fast, accurate model …

- 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

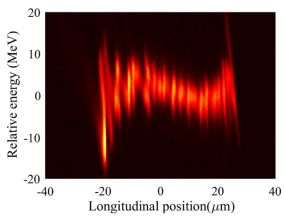
In reality things are much more difficult…

computationally expensive simulations

Simulation



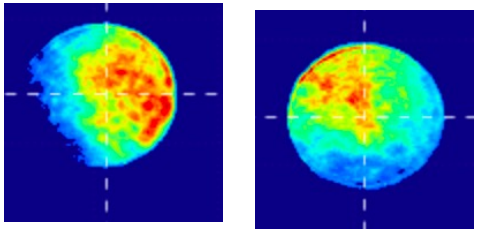
Measurement

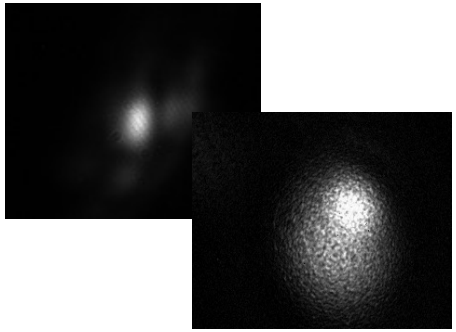


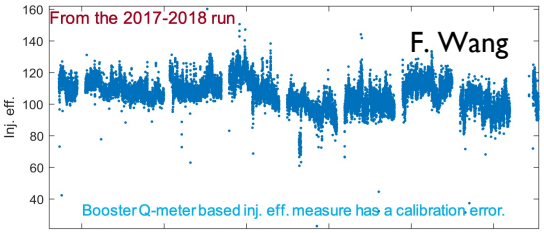
"10 hours on thousands of cores at the NERSC"

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

fluctuations/noise (e.g. laser spot)

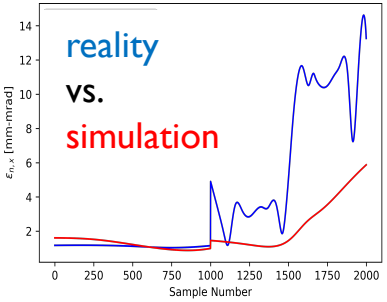




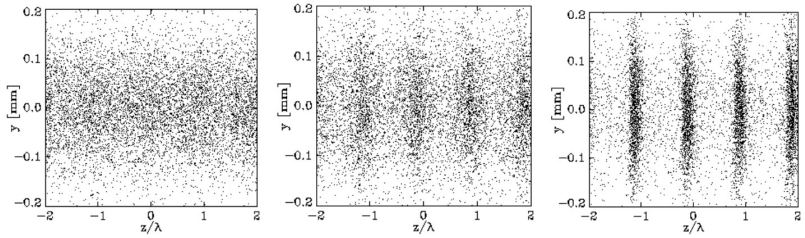


hidden variables / sensitivities

drift over time



many small, compounding sources of uncertainty

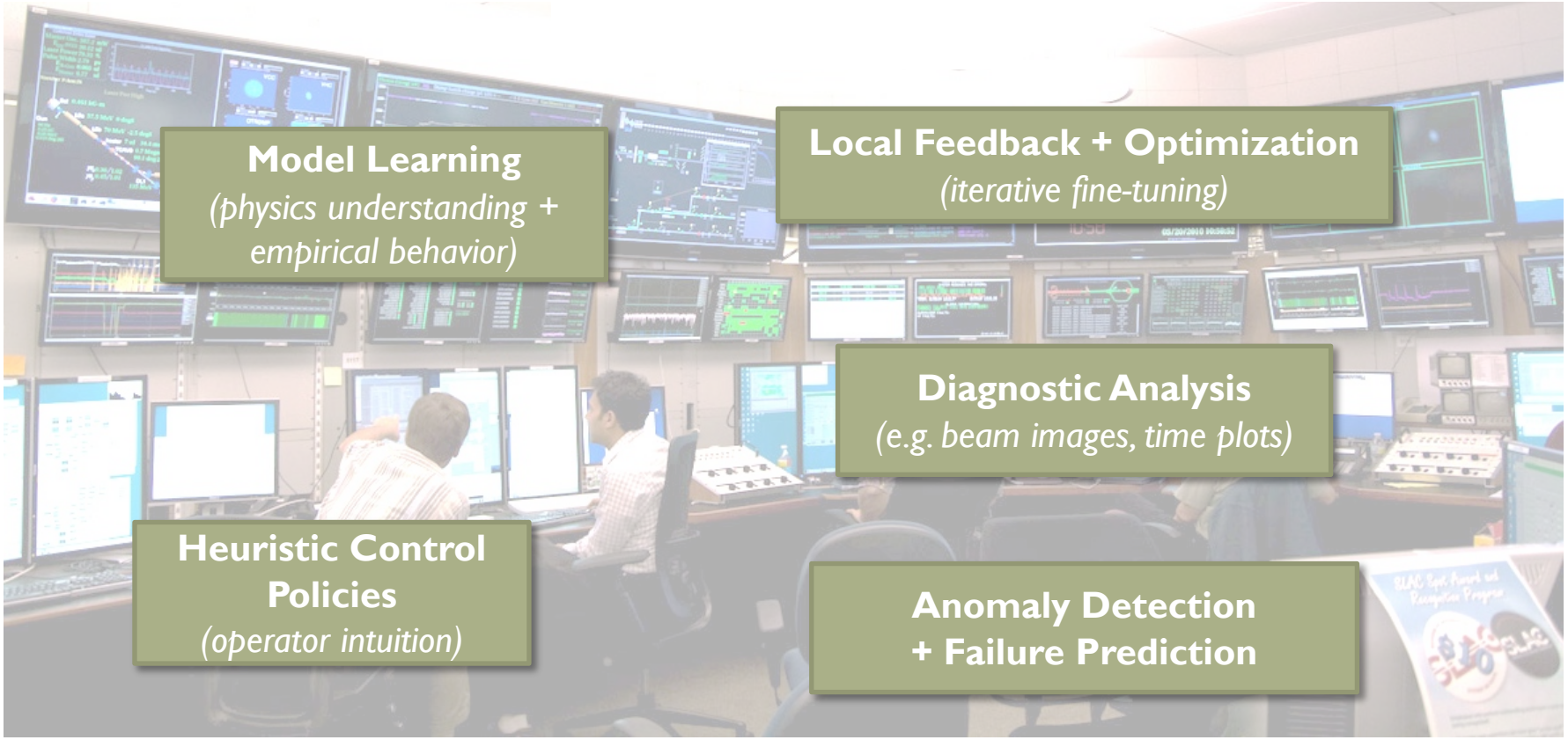


nonlinear effects / instabilities

ML can help with speed, accuracy, and uncertainty estimates for models

We rely heavily on operators for day-to-day control tasks . . .

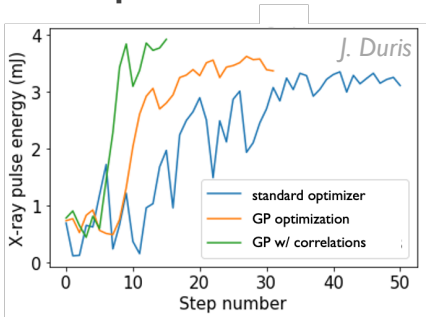
We rely heavily on operators for day-to-day control tasks . . .



…many analogous techniques in optimization, machine learning, computer vision, etc.

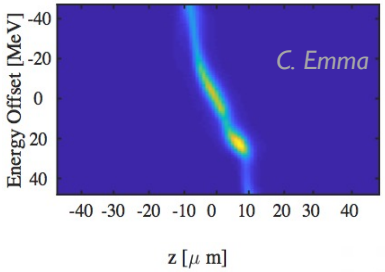
Several major areas for ML to play a role

automated control
+ optimization

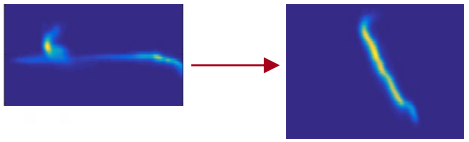


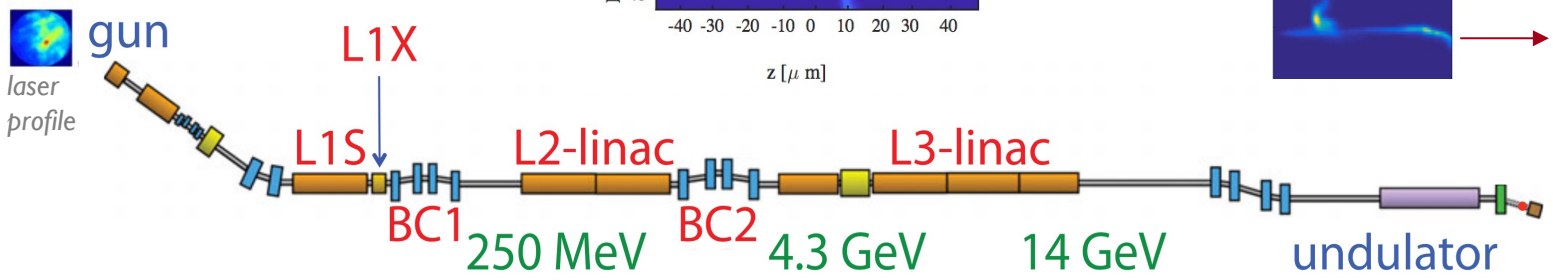
diagnostics

(reconstruct / analyze beam)

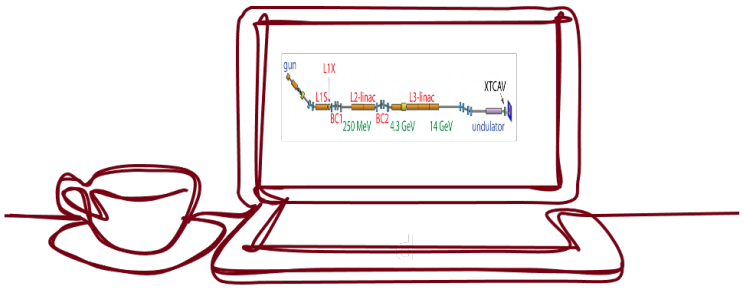


anomaly detection
failure prediction





incorporate
physics
information



extract unexpected
relationships

(feed into control / design)

digital twins + online modeling

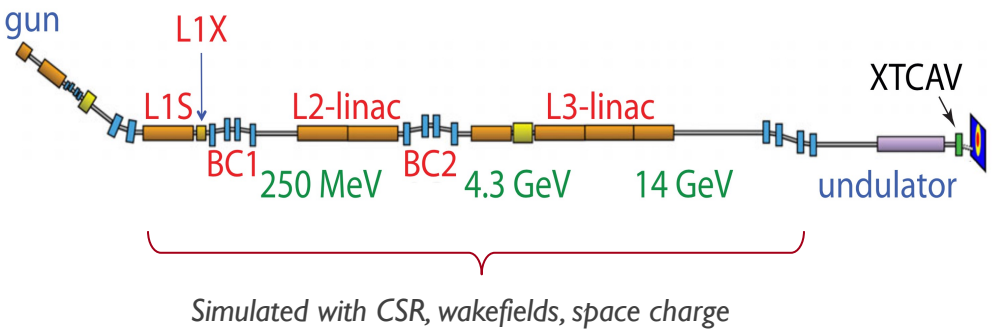
(planning, model-based control, finding differences between sim/machine)

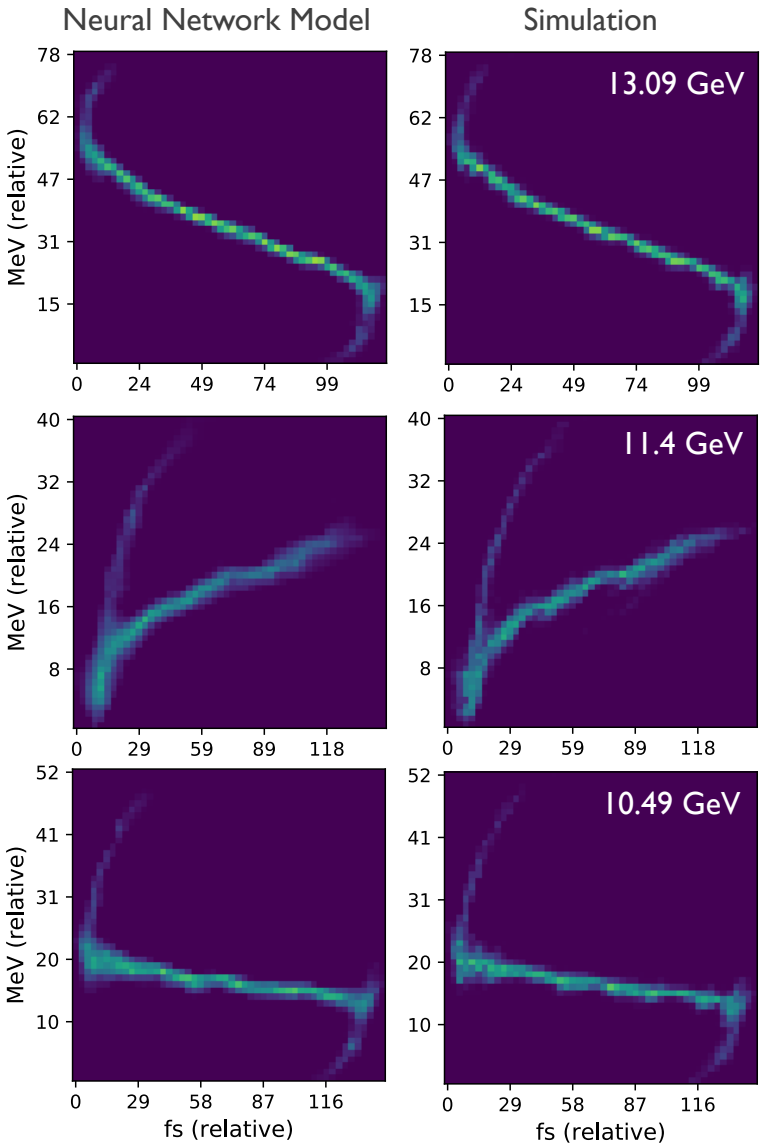
+ need uncertainty
quantification for all

Accelerator Modeling

Trained neural network on simulation data

→ ~ million times faster execution





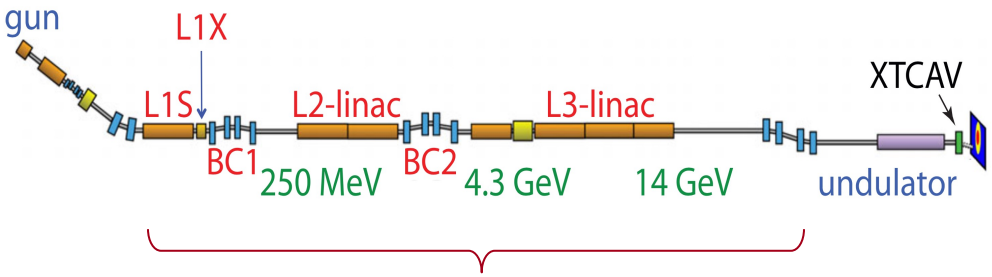
Wide scan of 6 settings in Bmad

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

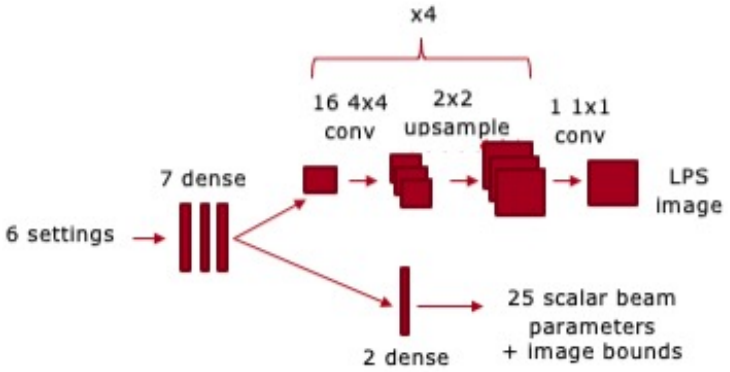
NN predicts 25 scalar outputs ($\sigma_{x,y,z}$ $\varepsilon_{x,y}$ $\sigma_{x',y'}$ σ_E *etc…*) and phase space at the undulator entrance

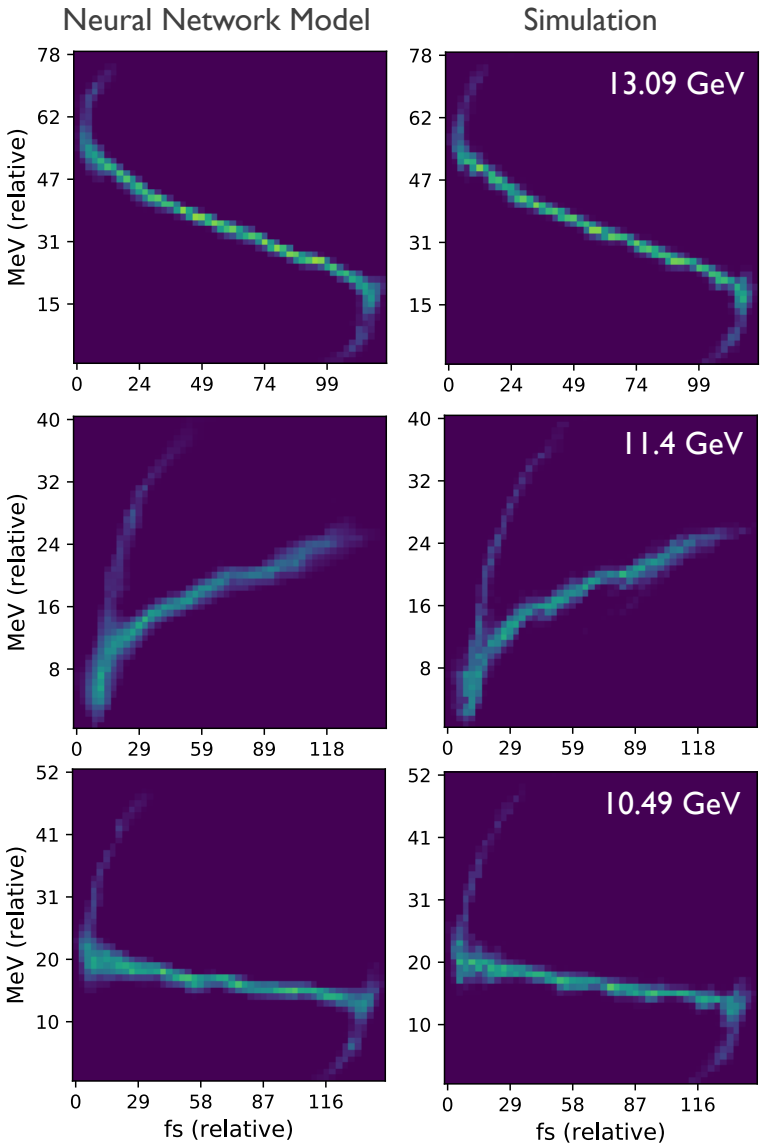
Trained neural network on simulation data

→ ~ million times faster execution



Simulated with CSR, wakefields, space charge

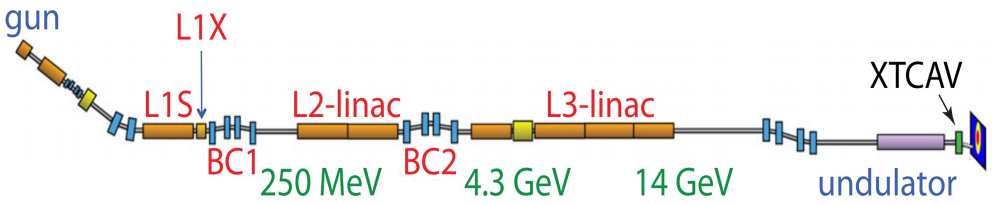


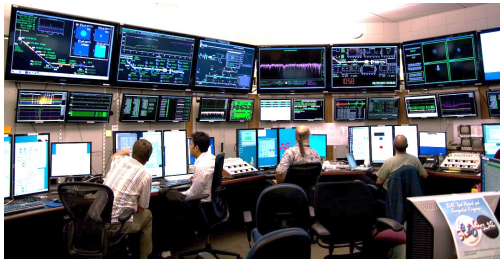


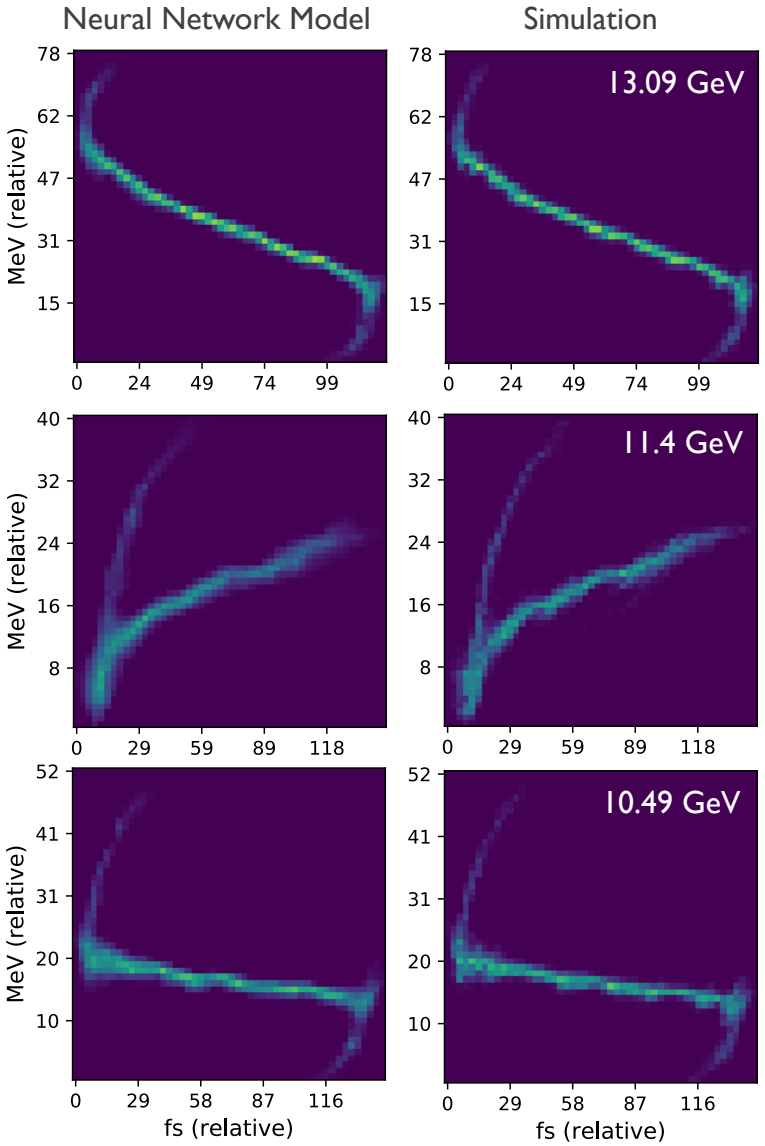
NN predicts 25 scalar outputs ($\sigma_{x,y,z}$ $\varepsilon_{x,y}$ $\sigma_{x',y'}$ σ_E *etc…*) and phase space at the undulator entrance

Trained neural network on simulation data

→ ~ million times faster execution



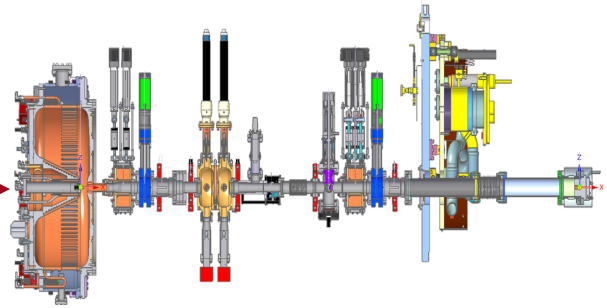


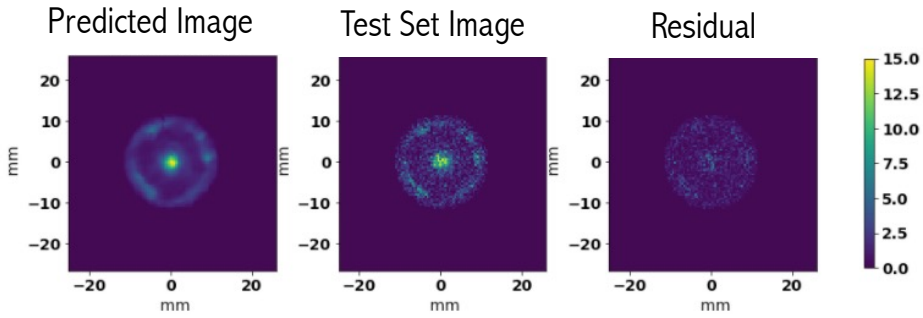


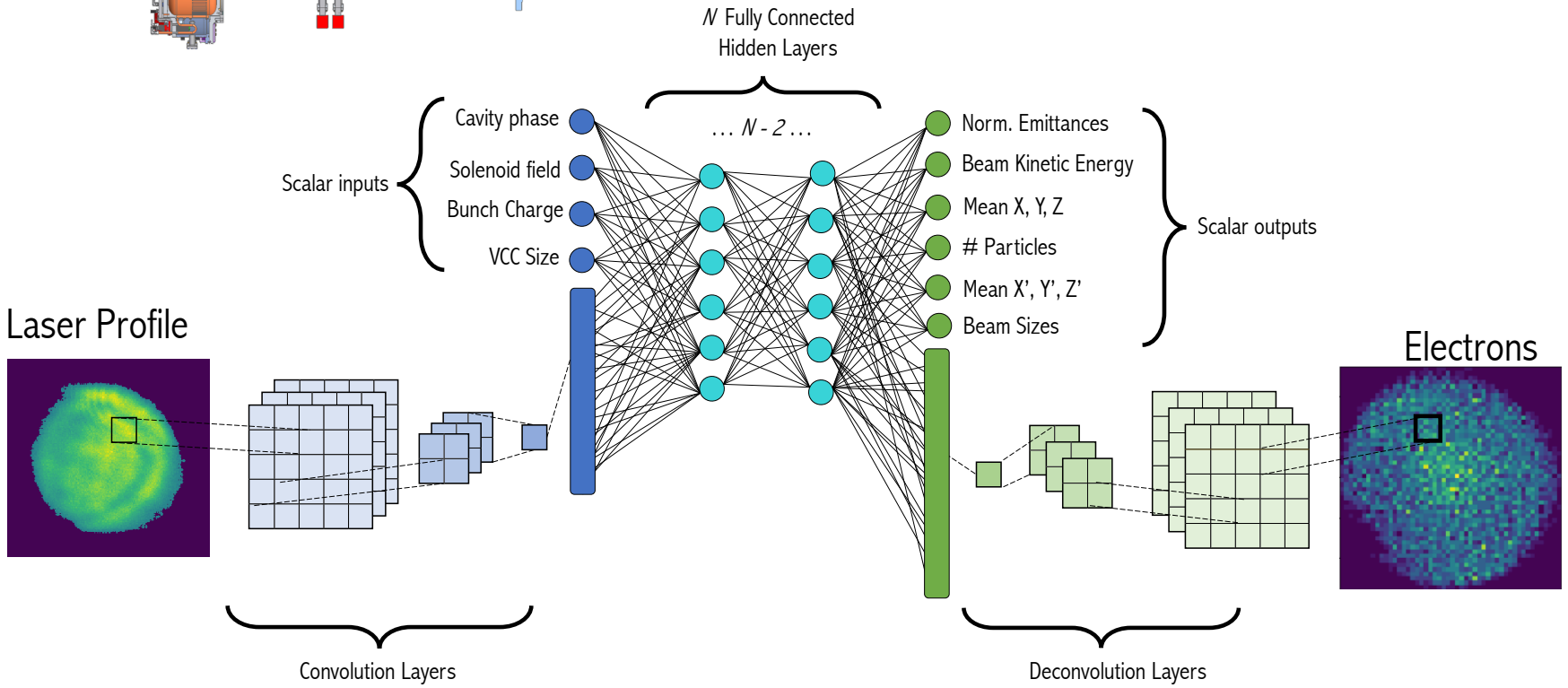
NN predicts 25 scalar outputs ($\sigma_{x,y,z}$ $\varepsilon_{x,y}$ $\sigma_{x',y'}$ σ_E *etc…*) and phase space at the undulator entrance

Using image-based diagnostic input directly

Photocathode drive laser





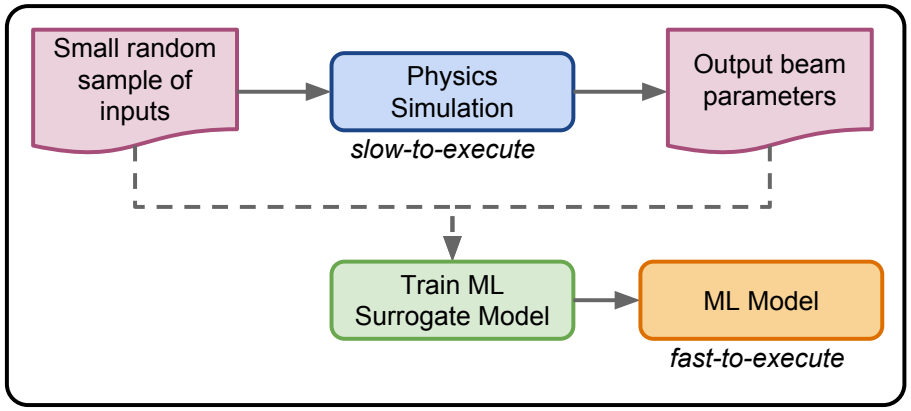


L. Gupta

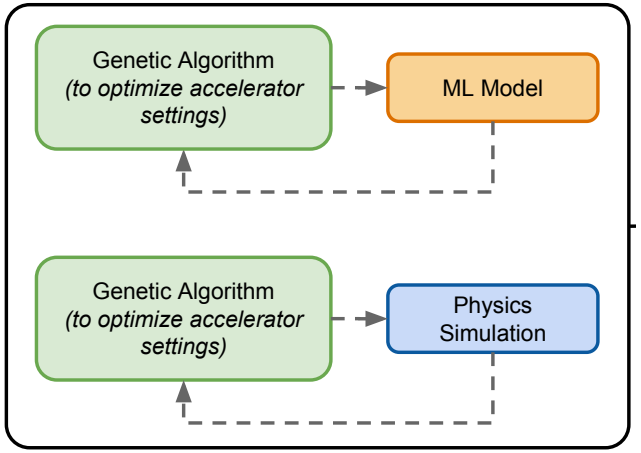
A. Edelen, N. Neveu, et al., https://arxiv.org/abs/1903.07759

Can we trust these models under optimization?

Generate ML Model using Sparse Random Sample



Run GA on ML Model and Physics Simulation

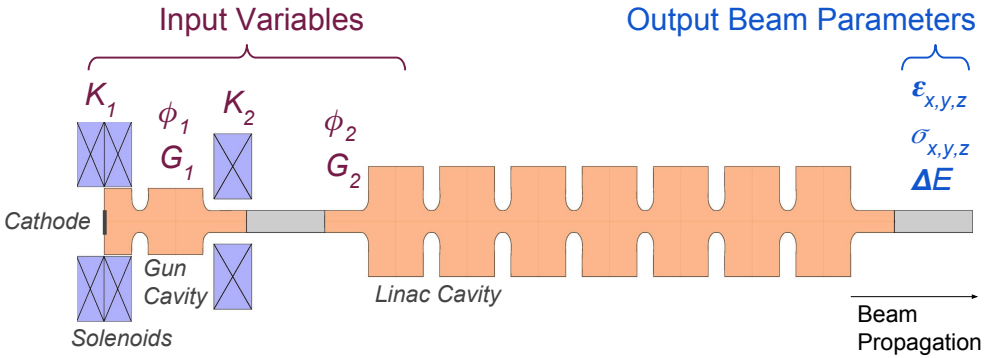


Test Case with Existing Data: Argonne Wakefield Accelerator Injector

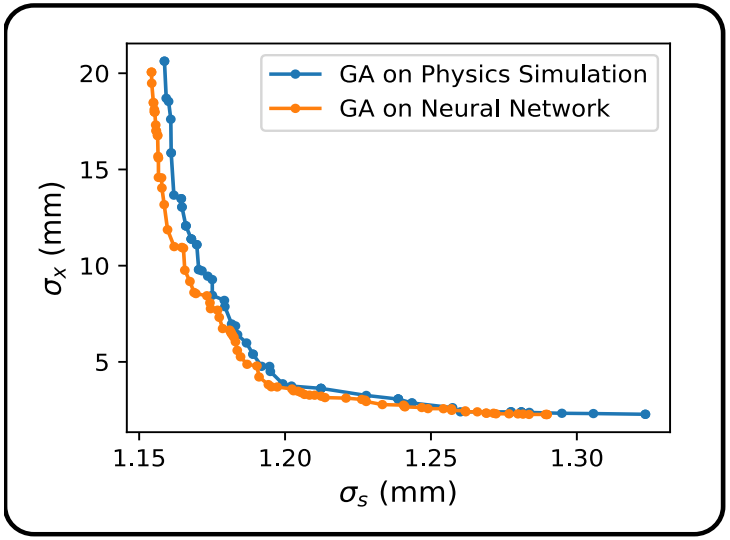
OPAL simulation (PIC) : 3D space charge 3D field maps

NSGA-II for optimization: 200 generations ~600 individuals

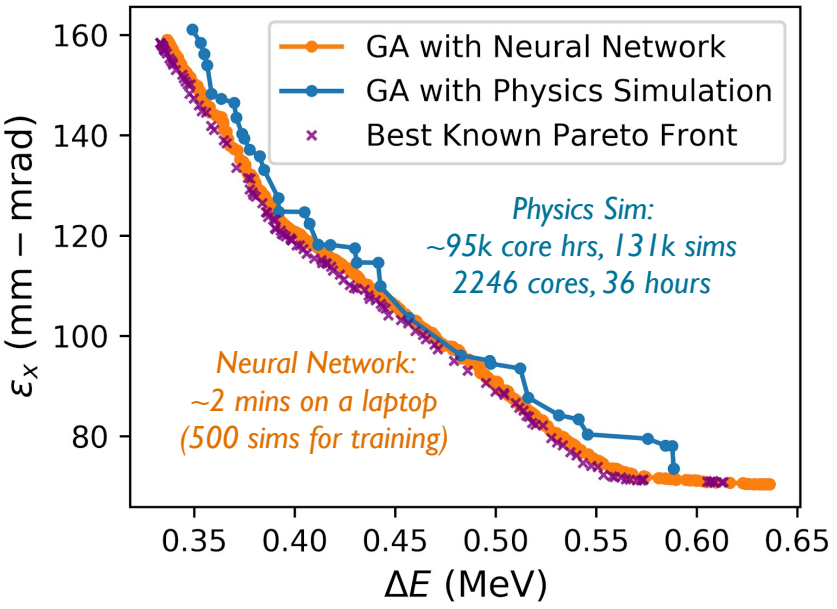
500 random points for training

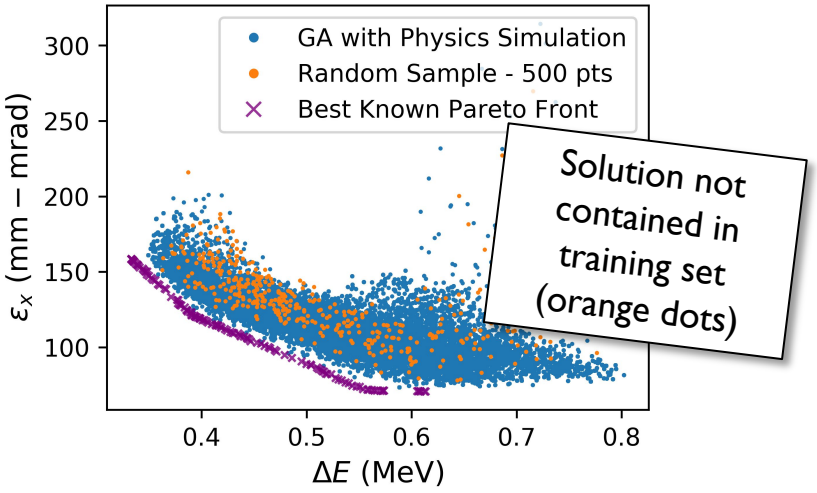


Compare Resulting Pareto Fronts



Required **~260x fewer simulation evaluations** overall and had **10^6 x faster execution** in equivalent optimization task





In terms of time-to-solution:

~6.4 mins on 8 cores to make 500-point training data

~10 minutes to train on a laptop

~2 minutes to do optimization on a laptop

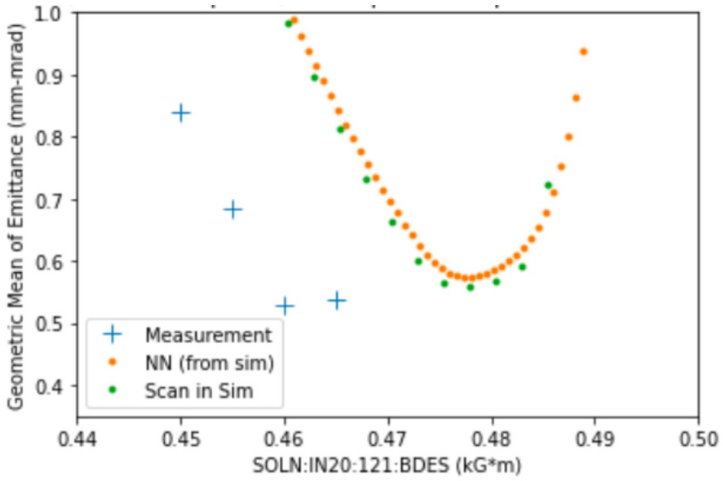
Also useful for initial optimization with greater sample-efficiency

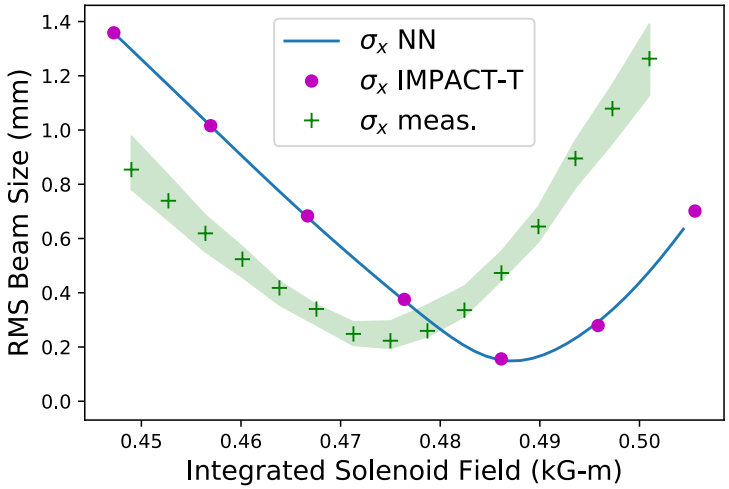
Can do iteratively for further improvement, or use bayes opt (later slide)

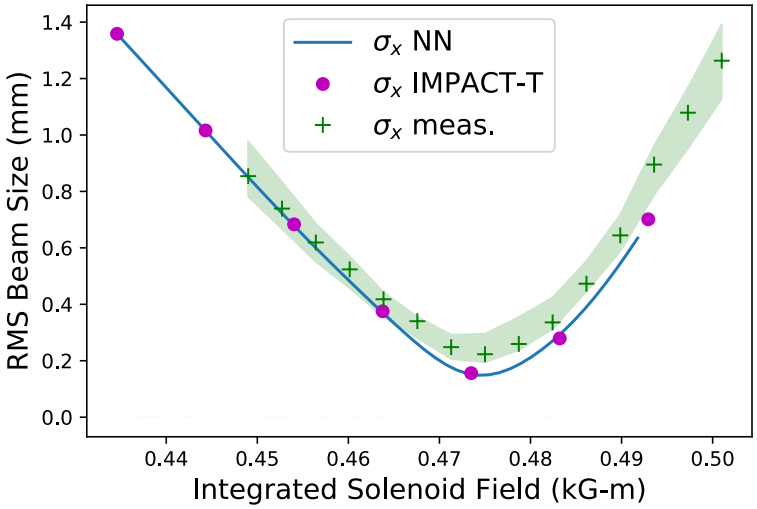
Finding Sources of Error Between Simulations and Measurement

Real accelerator can have many non-idealities and miscalibrations not included in physics simulations

→ *Neural network model allows fast / automatic exploration of possible error sources*





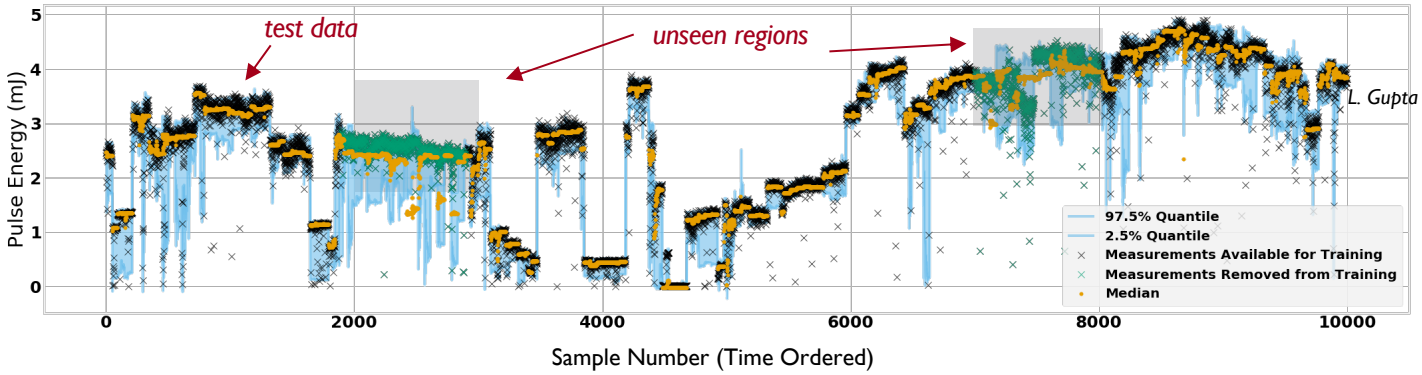


Here: calibration offset in solenoid strength found automatically with neural network model (trained first in simulation, then calibrated to machine)

Uncertainty Quantification

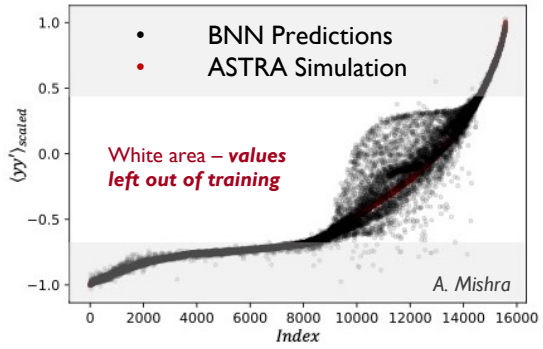
Need prediction uncertainties → want to trust predictions, have safe exploration of parameter space

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

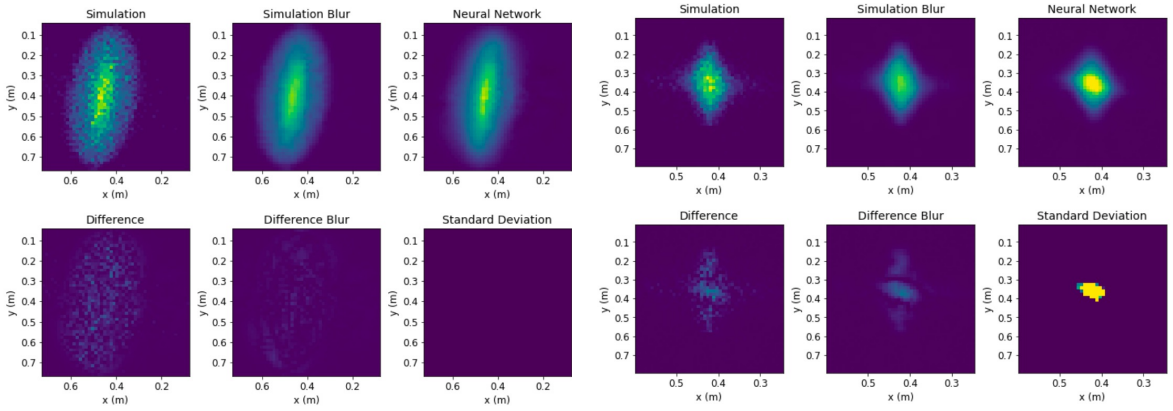


Neural network with quantile regression predicting FEL pulse energy at LCLS

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

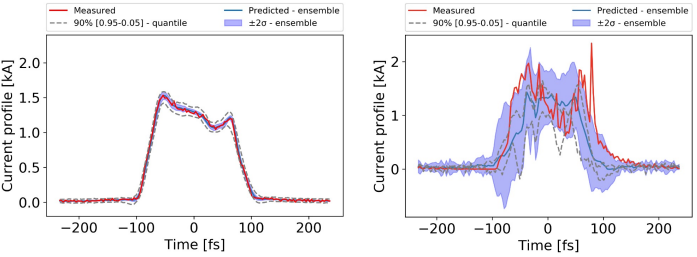


Bayesian neural network predicting scalar parameters for the LCLS-II injector



Test shot within trained distribution

Out-of-distribution

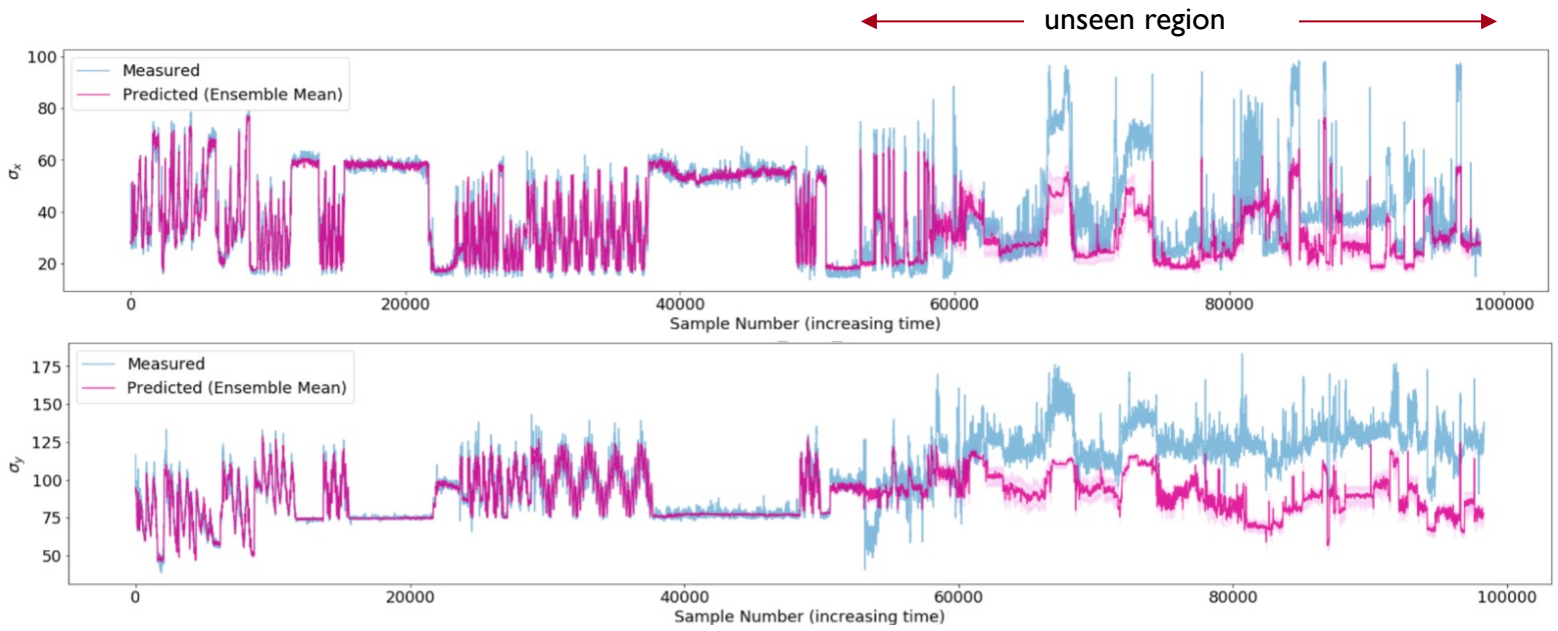


Longitudinal phase space beam profiles

O. Convery, PRAB, 2021

LCLS injector transverse distributions on out-of-training distribution shots, neural network ensemble

Example of prediction under large drift in inputs (and possibly hidden variables):



Uncertainty estimate from neural network ensemble does not accurately cover the OOD prediction error, but it is relatively higher than for in-distribution data

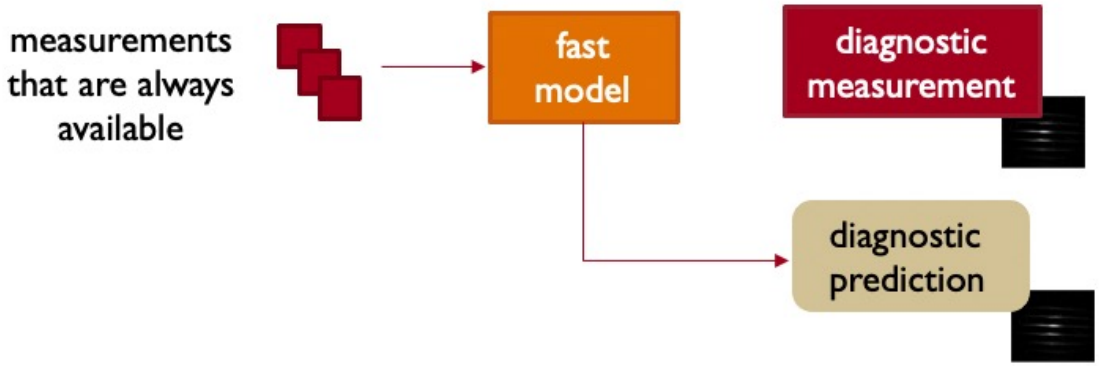
ML-enhanced diagnostics

faster measurements and more information → better control

Virtual Diagnostics

Real diagnostic not always available:

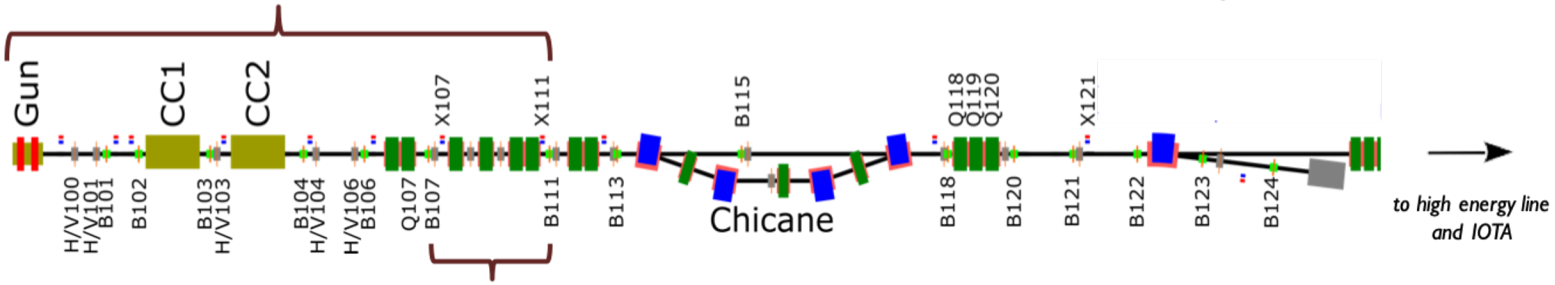
- *destructive, cannot use during user operations*
- *not sensitive in entire operating range*
- *slower update rate than desired*
- *moved to another location*

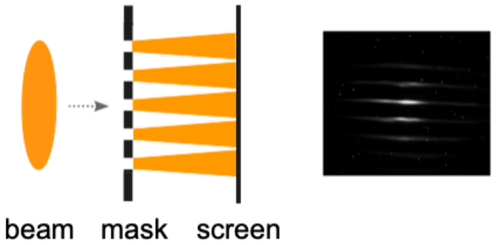


Can use a physics simulation if fast / accurate enough
→ without this, can use a learned model

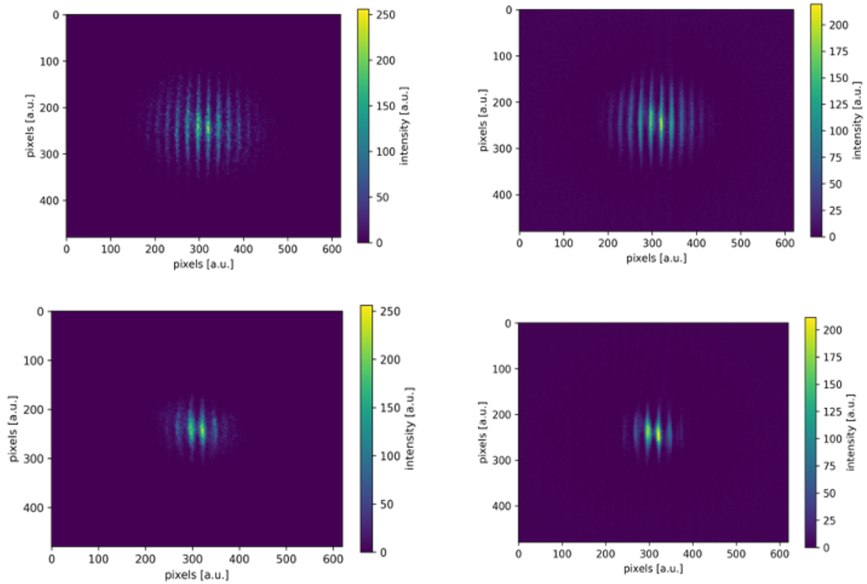
section for virtual diagnostic

A. L. Edelen, J.P. Edelen, D. Edstrom, et al. NAPAC16, TUPOA51
A. L. Edelen, J.P. Edelen, D. Edstrom, et al. 1st ICFA ML Workshop, Feb. 2018



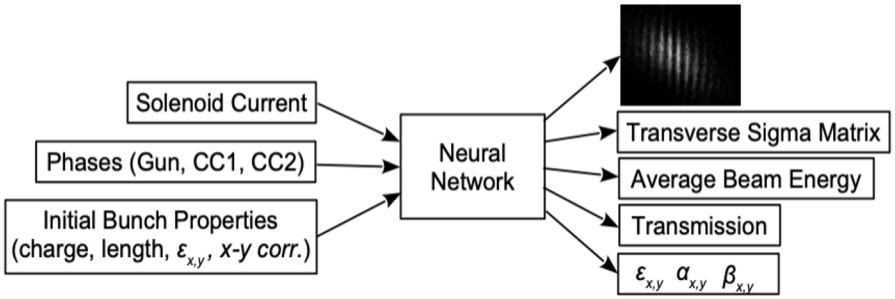


fit to obtain $\langle y^2 \rangle$
subset of phase $\langle yy' \rangle$
space parameters $\langle y'^2 \rangle$



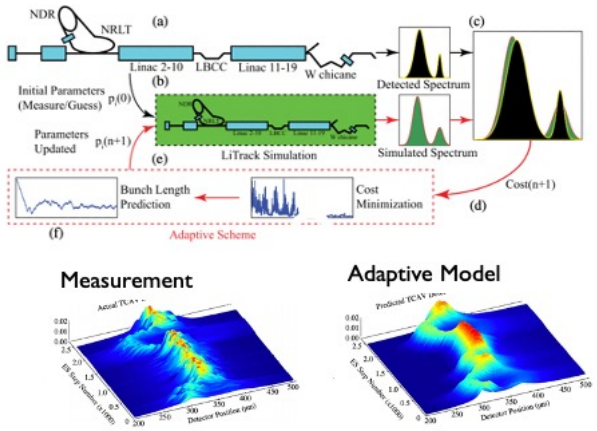
Simulated

NN Predictions



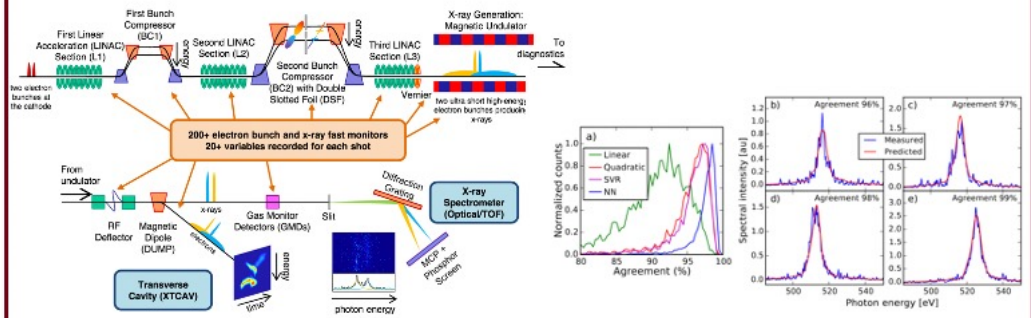
Examples for longitudinal phase space: mix of adaptively calibrated physics models and ML-based prediction…

Adaptively tune a simple physics model



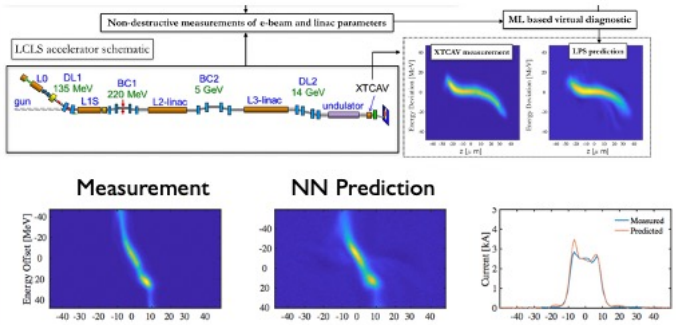
A. Scheinker, S.Gessner, PRAB 18, 102801 (2015)

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



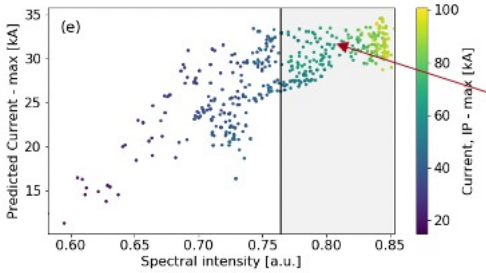
A. Sanchez-Gonzalez, et al., Nature Comms (2017)

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



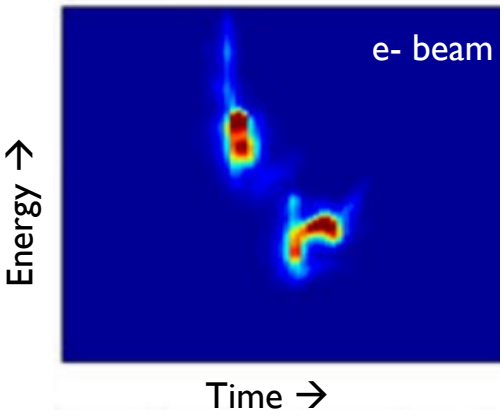
Shots are beyond the TCAV resolution

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

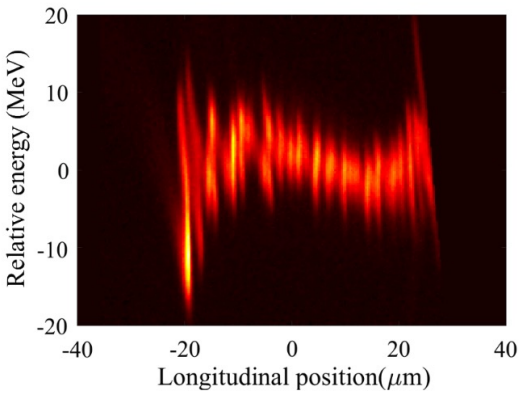
Prediction → Analysis

Signals used in feedback control and experimental analysis are complicated *(e.g. beam images, time series)*

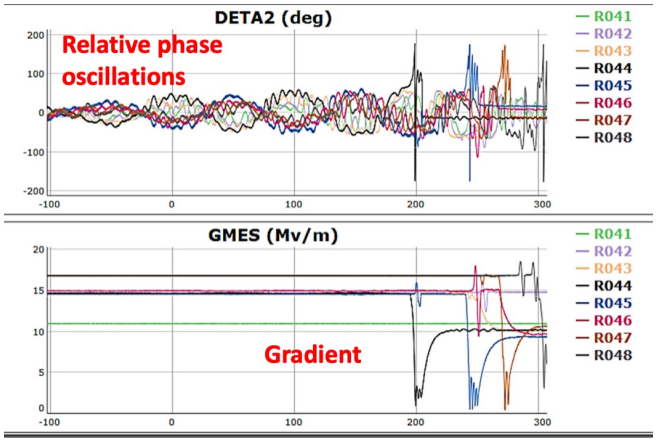
Can use ML to extract more useful information from these high-dimensional signals



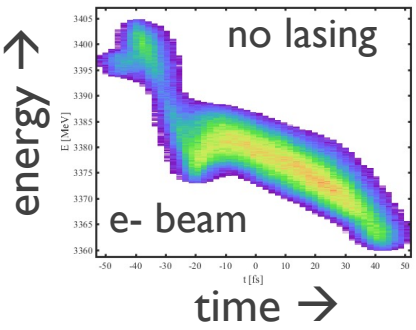
A. Marinelli, et al., Nat. Commun. **6**, 6369 (2015)

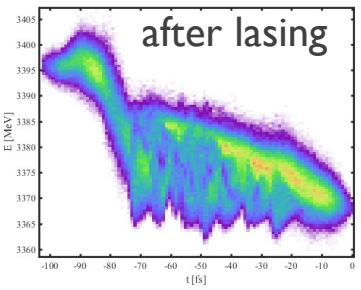


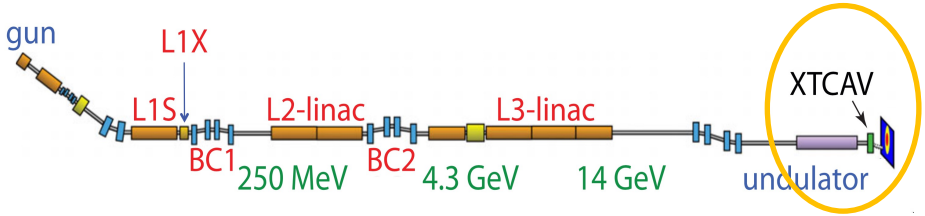
J. Qiang, et al., PRSTAB30, 054402, 2017



A. Solopova, IPAC'19

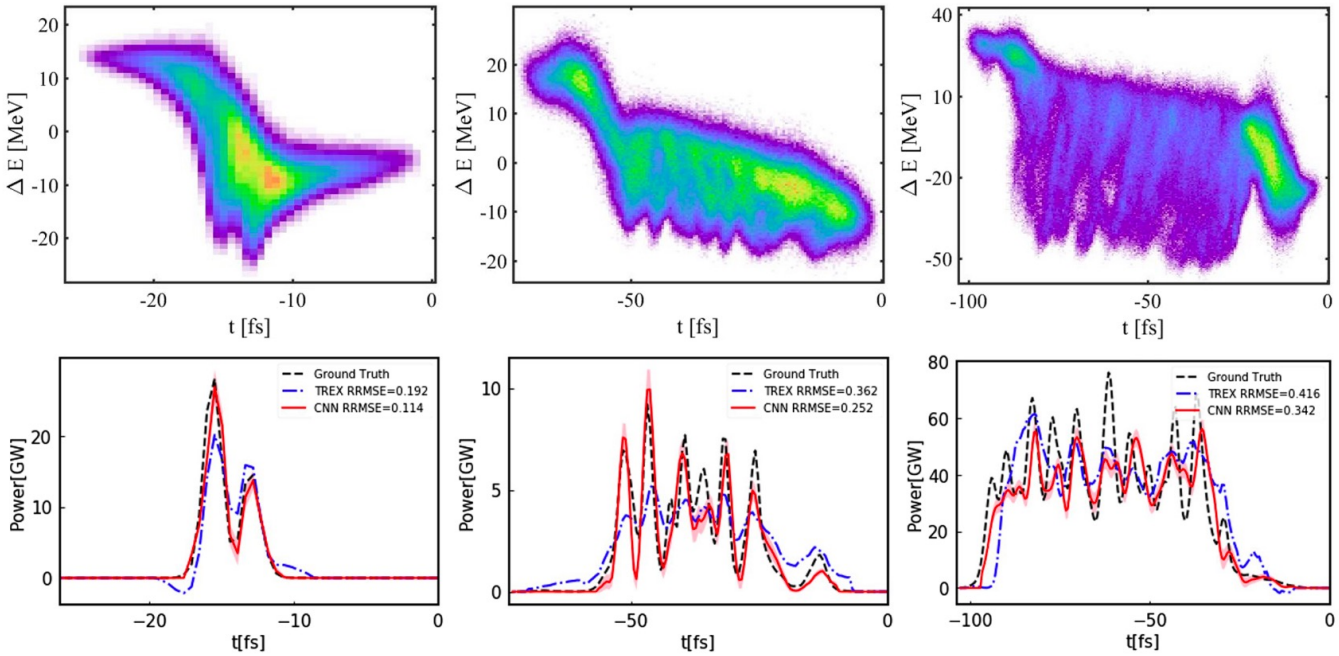






e- beam loses energy to photon beam

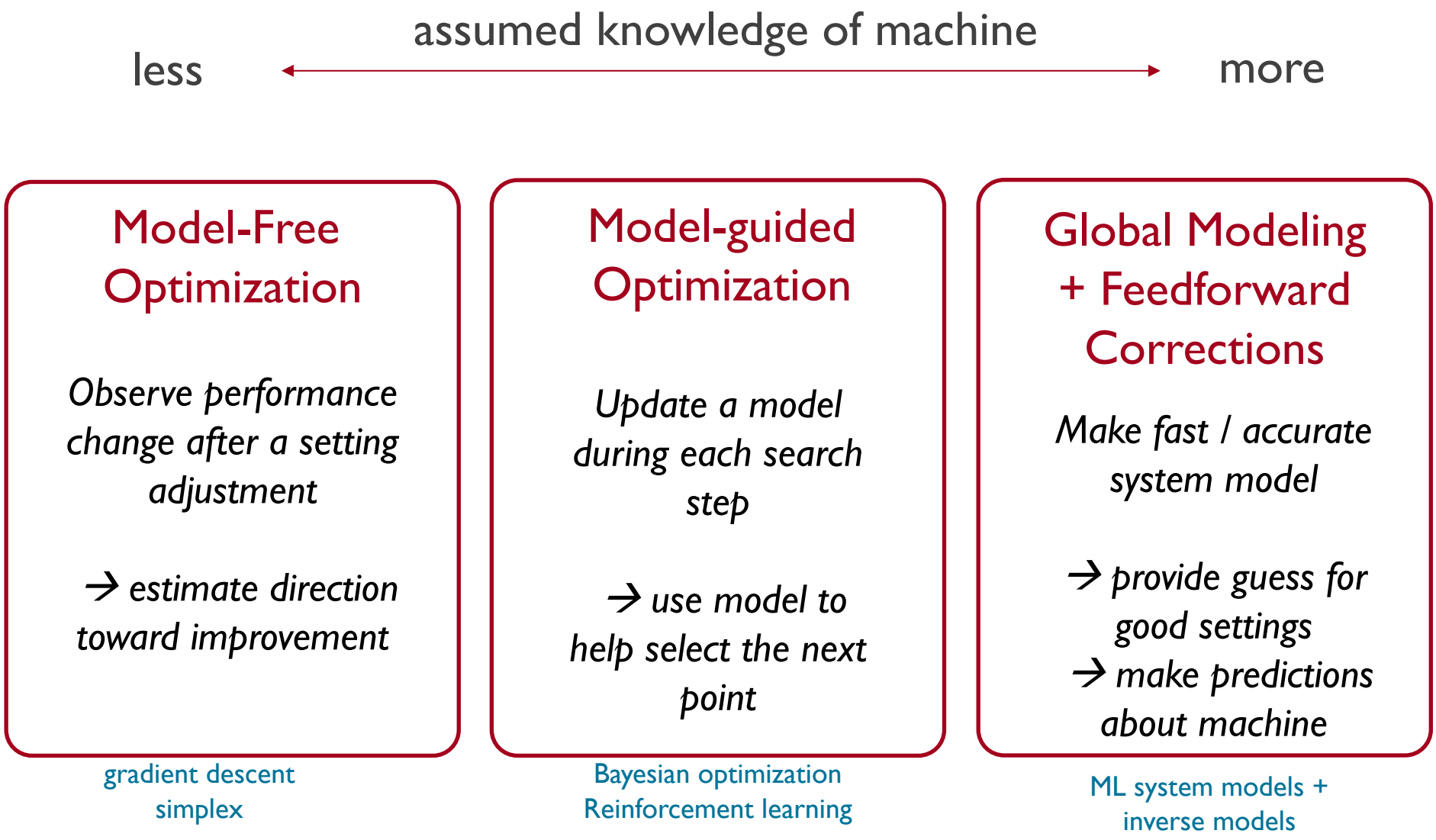
Can use e-beam images to predict unmeasured photon beam power profile
-Standard method is slow/iterative and doesn't work well into saturation



CNN is faster / more accurate than standard reconstruction technique

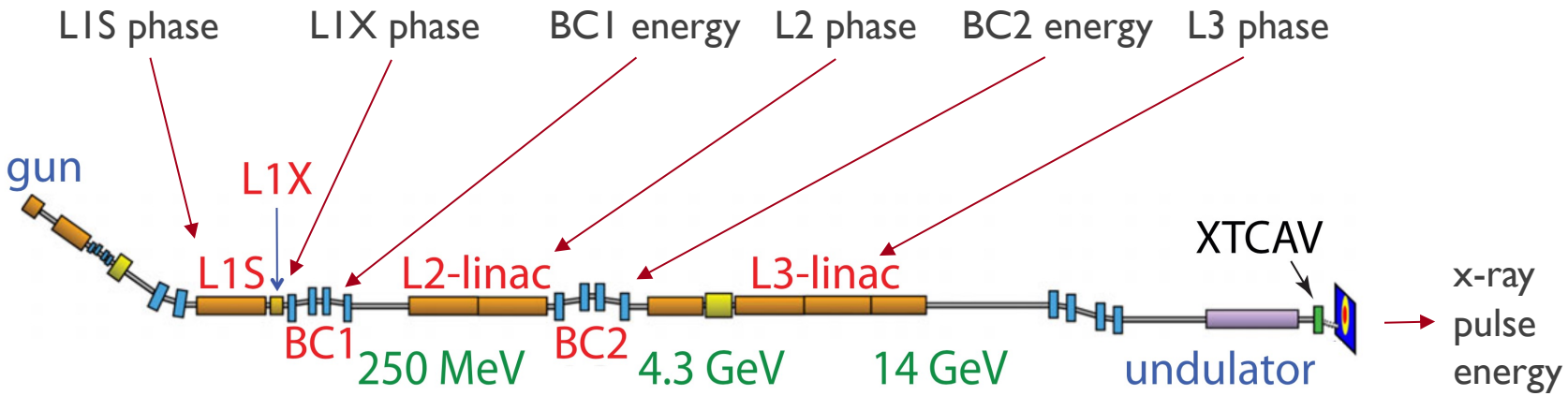
X. Ren, A. Edelen, D. Ratner, et al., PRAB 2020

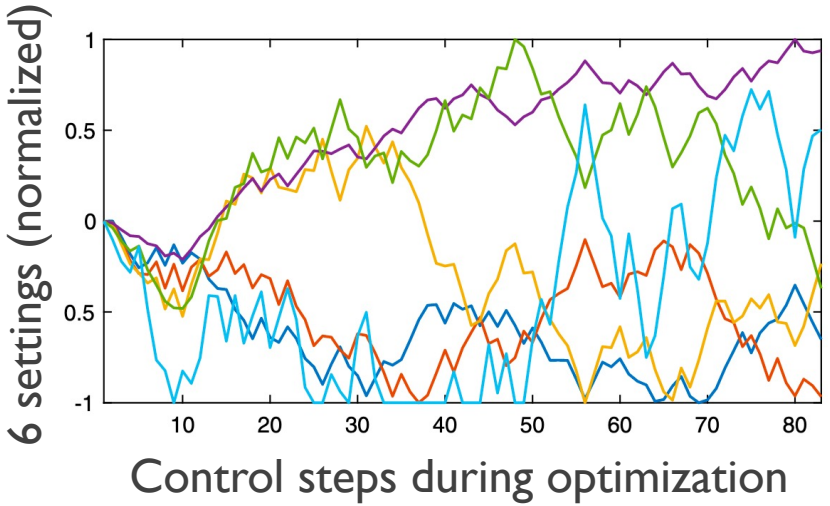
Tuning/Optimization

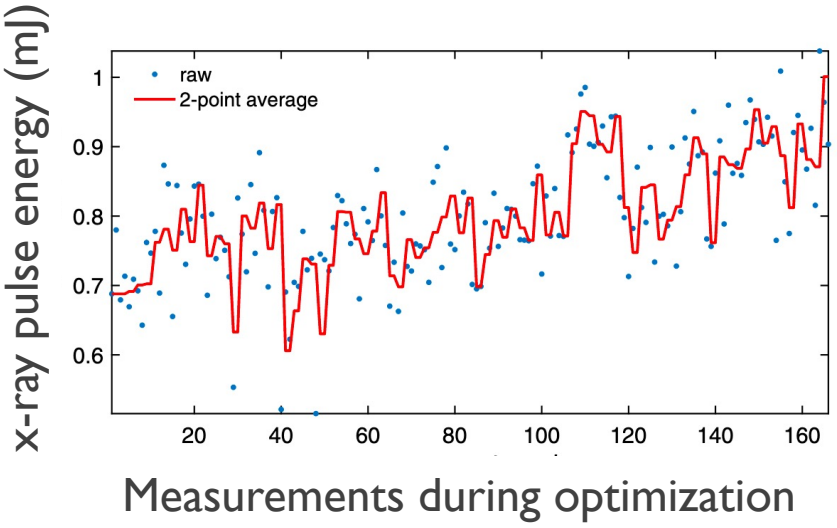

assumed knowledge of machine
less
more
Model-Free Optimization
Observe performance change after a setting adjustment
→ estimate direction toward improvement
gradient descent
simplex
Model-guided Optimization
Update a model during each search step
→ use model to help select the next point
Bayesian optimization
Reinforcement learning
Global Modeling + Feedforward Corrections
Make fast / accurate system model
→ provide guess for good settings
→ make predictions about machine
ML system models + inverse models

Model-free optimizers can help…

A. Scheinker, PRAB, 2019





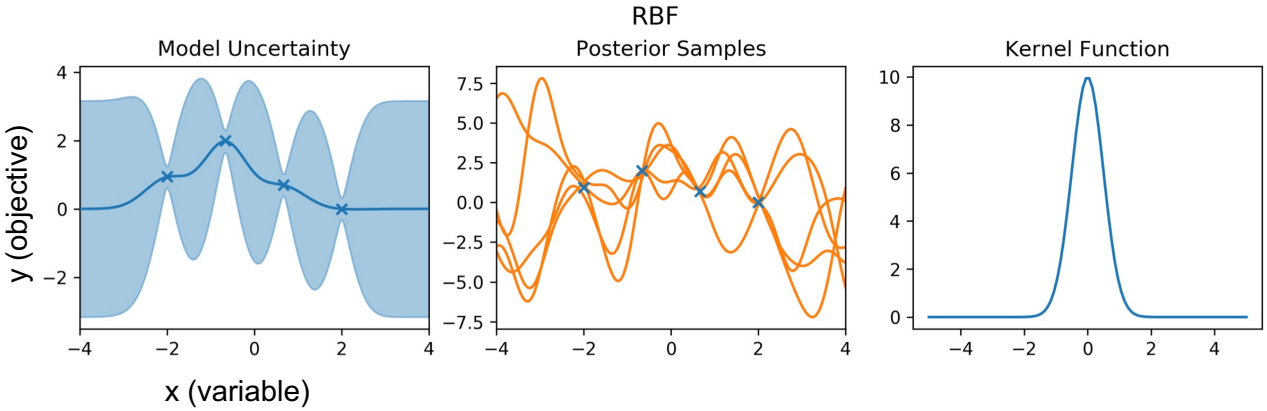


...but convergence can be very slow + can get stuck in local minima

Bayesian Optimization

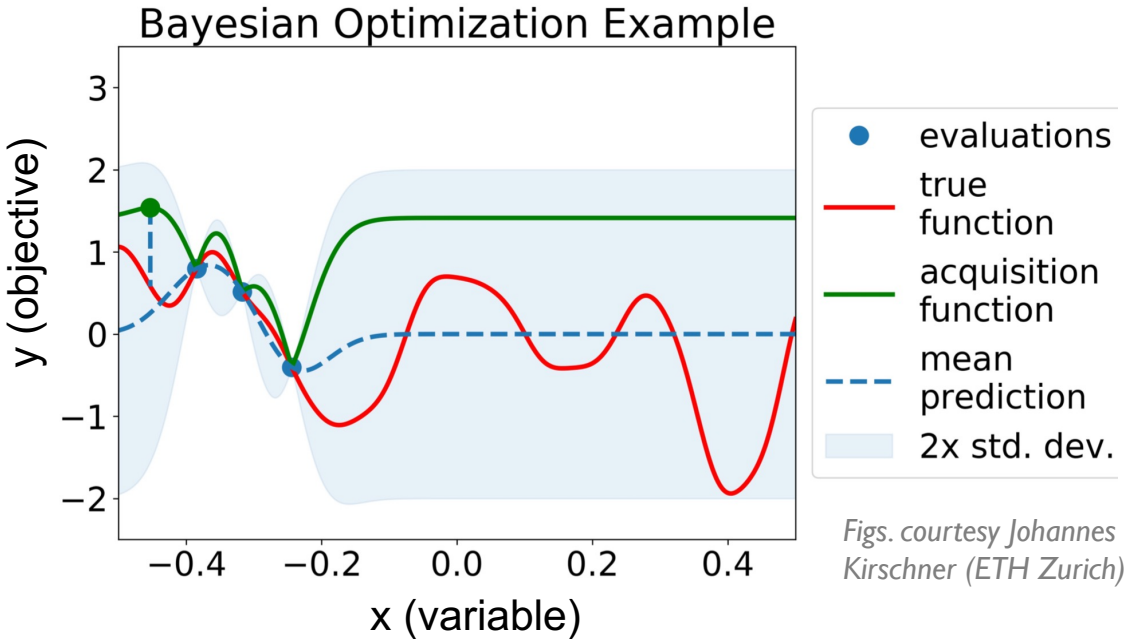
Set up probabilistic model

→ e.g. Gaussian Process



Iteratively refit model while sampling new points

Use model predictions and uncertainty to guide search for optimum while sampling

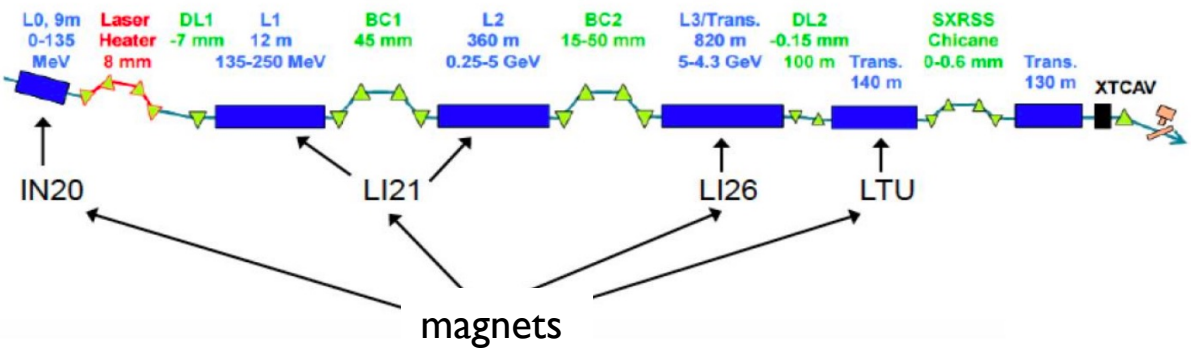


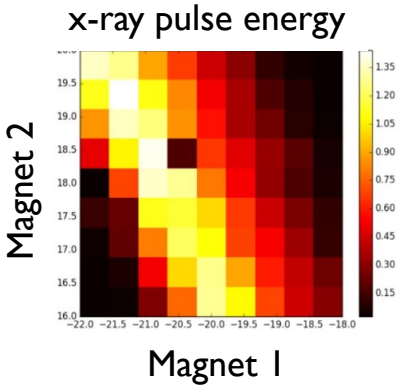
Figs. courtesy Johannes Kirschner (ETH Zurich)

Model-informed Bayesian optimization

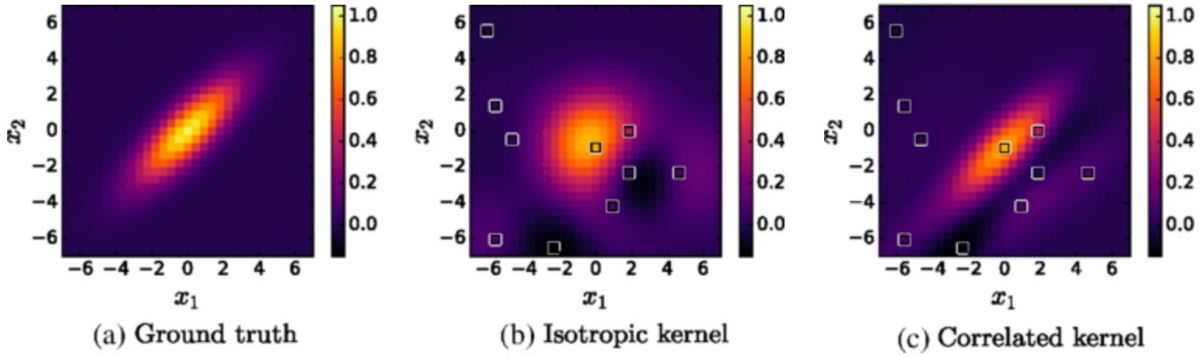
J. Duris et al., PRL, 2020

→ can design GP kernel based on expected physics





Goal: adjust focusing magnets to maximize x-ray pulse energy

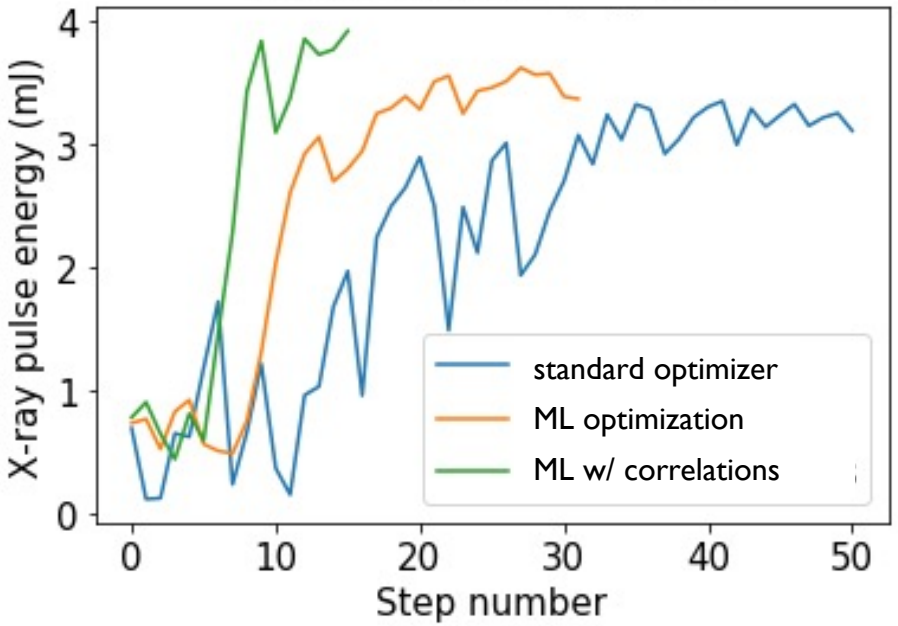


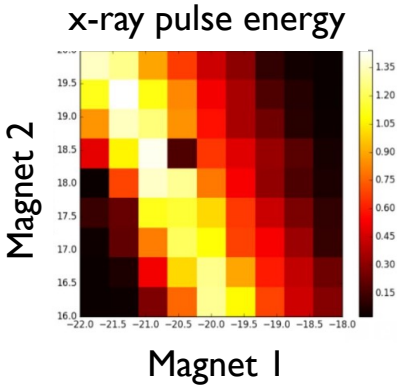
Including expected correlation improves ability to model the data with fewer samples
→ faster optimization

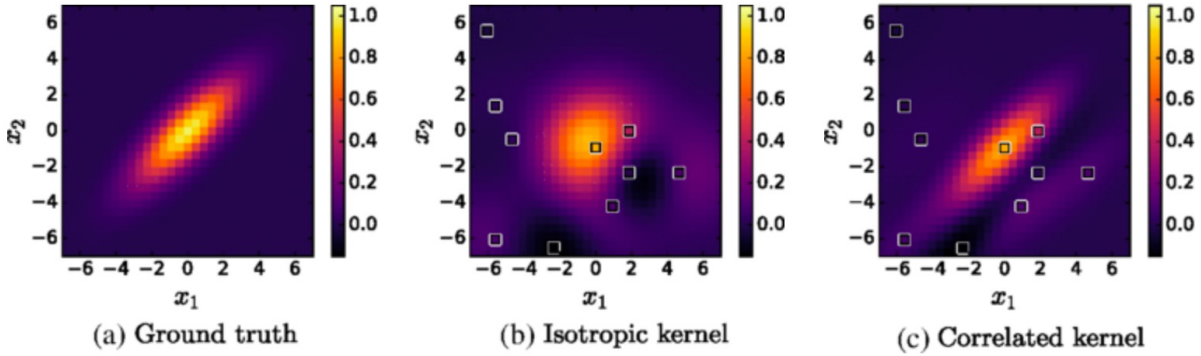
Model-informed Bayesian optimization

J. Duris et al., PRL, 2020

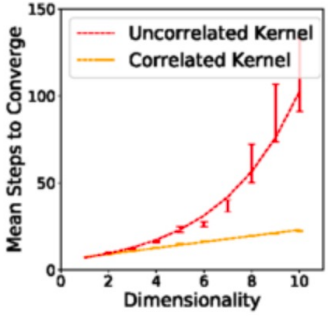
→ can design GP kernel based on expected physics







(a) Ground truth (b) Isotropic kernel (c) Correlated kernel



(d) Convergence tests

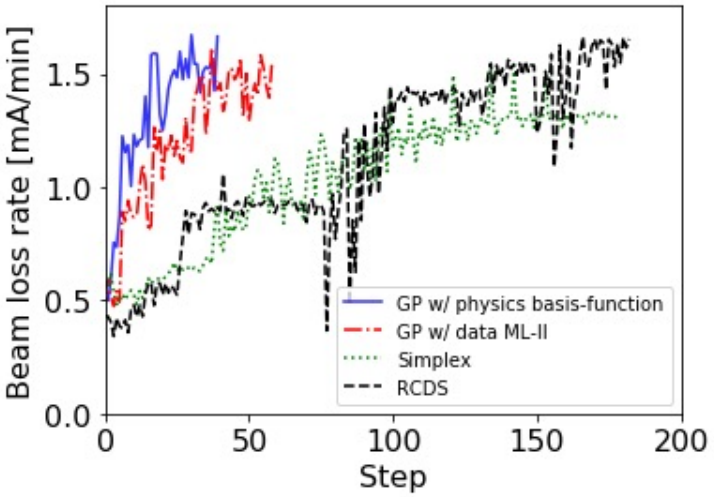
Including expected correlation improves ability to model the data with fewer samples
→ faster optimization

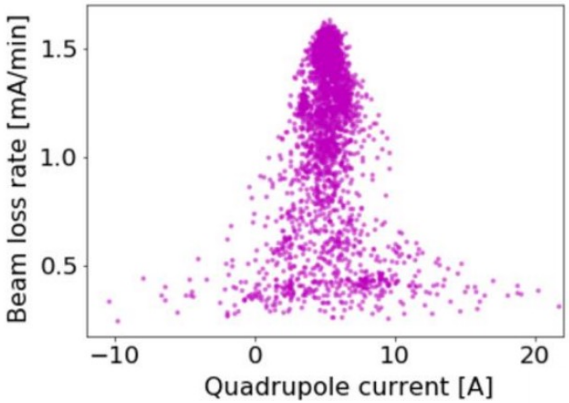
Model-informed Bayesian optimization

A way to get the correlations:

Take the Hessian of a model at the expected optimum → *use those correlations in the GP kernel*

As long as qualitative behavior is correct in model, should result in faster convergence



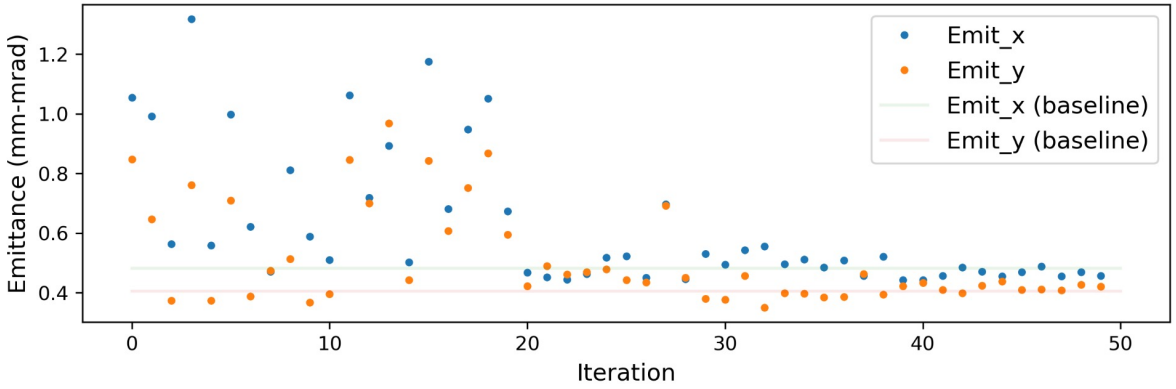


A. Hanuka et al., NeurIPS 2019
A. Hanuka et al., arXiv:2009.03566 (accepted to PRAB)

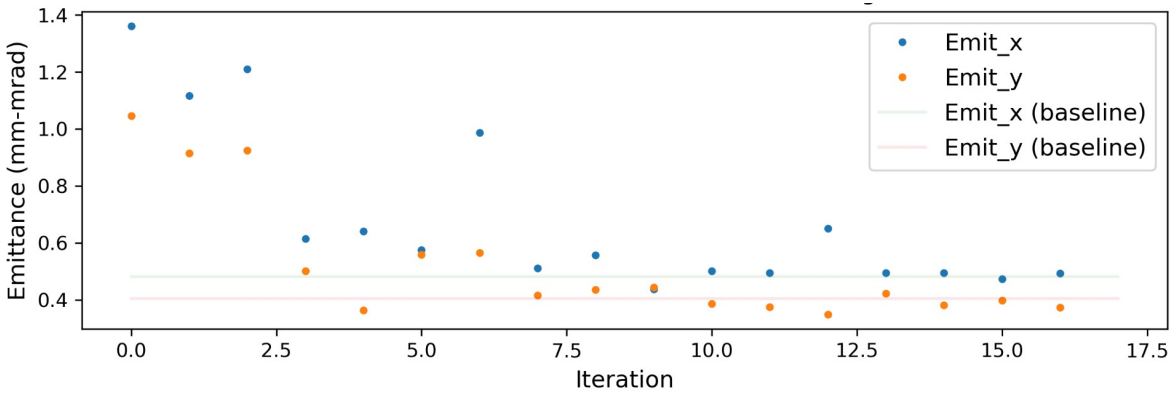
Was demonstrated at SPEAR3 for minimizing the vertical emittance (beam loss rate)

→ **No measured data needed, just a simulation**

Example for faster optimization of LCLS injector leveraging simulation-based surrogate model (no previous data)



Standard RBF Kernel



Kernel from Hessian of Surrogate Model (trained on IMPACT-T sims)

Both start from randomly sampling within the bounds
"Baseline" is tuning solution that ops was using that day

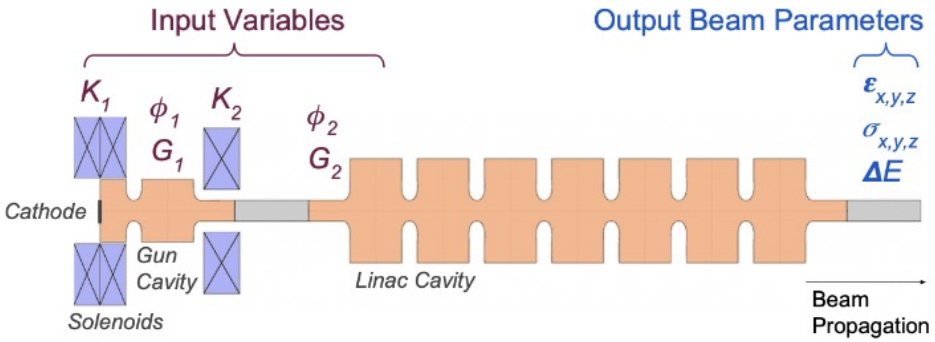
→ ***Using simulation surrogate model to inform optimization allows rapid tuning to human-level quality without any previous data***

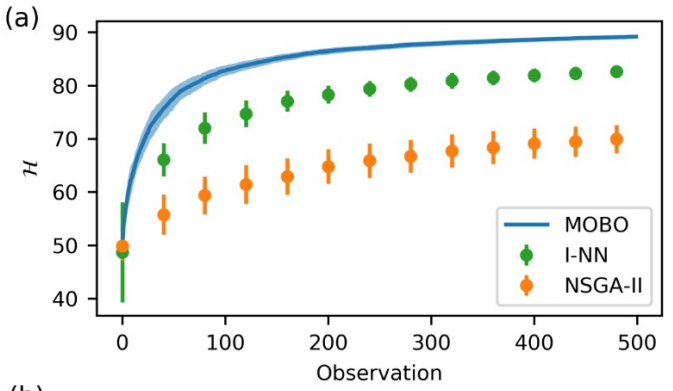
Multi-objective Bayesian optimization

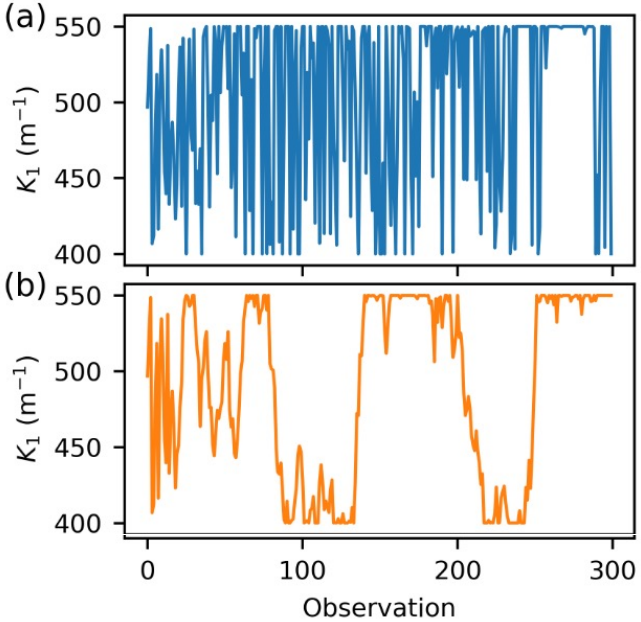
Use Bayesian optimization for **serial online multi-objective optimization**

More sample-efficient and fills out front efficiently than other methods

→ Could be extremely useful for characterization







Can enforce smooth exploration

(no wild changes in input settings)

R. Roussel, et al., PRAB (2021)
https://journals.aps.org/prab/pdf/10.1103/PhysRevAccelBeams.24.062801

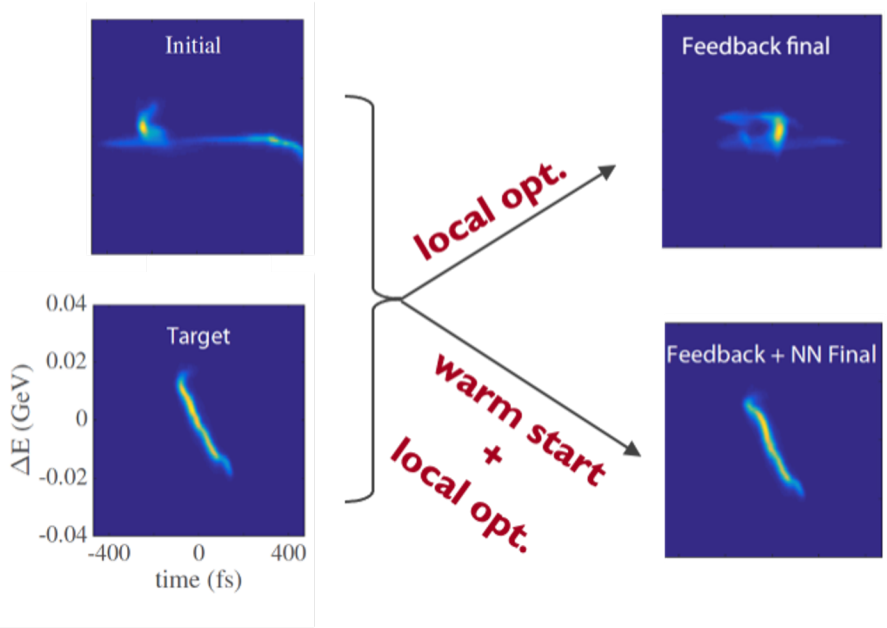
Faster optimization with warm starts from global models

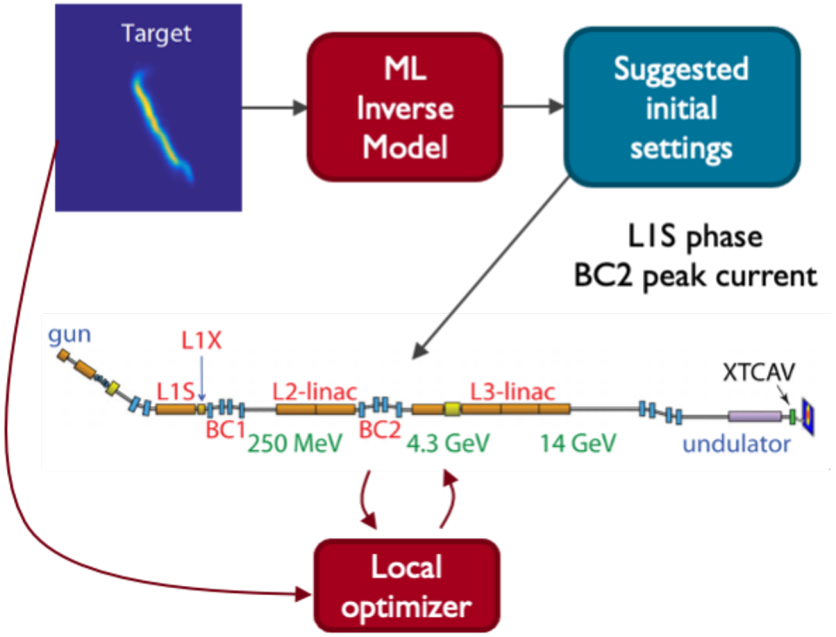
What if we are far away from some target beam parameters and want to switch between configurations quickly?

→ *Use global model to give an initial guess at settings, then refine with local optimization ("warm start")*

Example at LCLS:

- Two settings scanned *(L1S phase, BC2 peak current)*; trained neural network model to map longitudinal phase space to settings
- Compared optimization algorithm with/without warm start

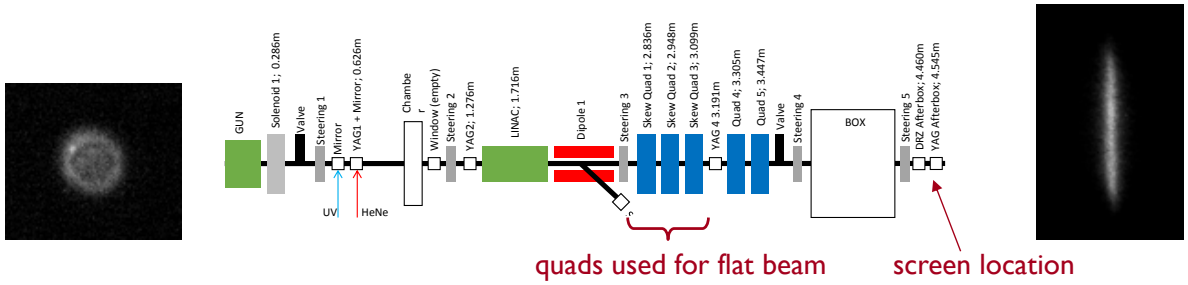




Local optimizer alone was unable to converge → **able to converge after initial settings from neural network**

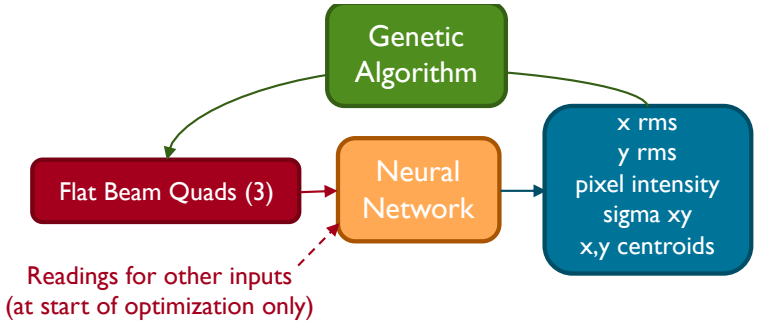
A. Scheinker, A. Edelen, et al., PRL 121, 044801 (2018)
sim study w/ a THz FEL: A. Edelen, et al., FEL'17

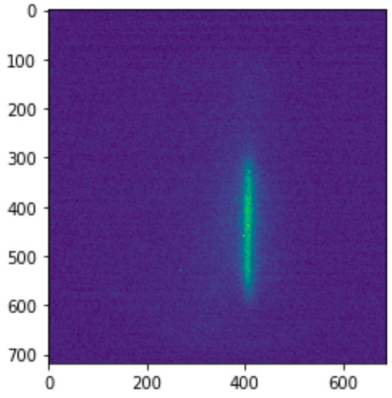
Another way: run optimizer on learned online model



Expert hand-tuning:
10 – 20 minutes

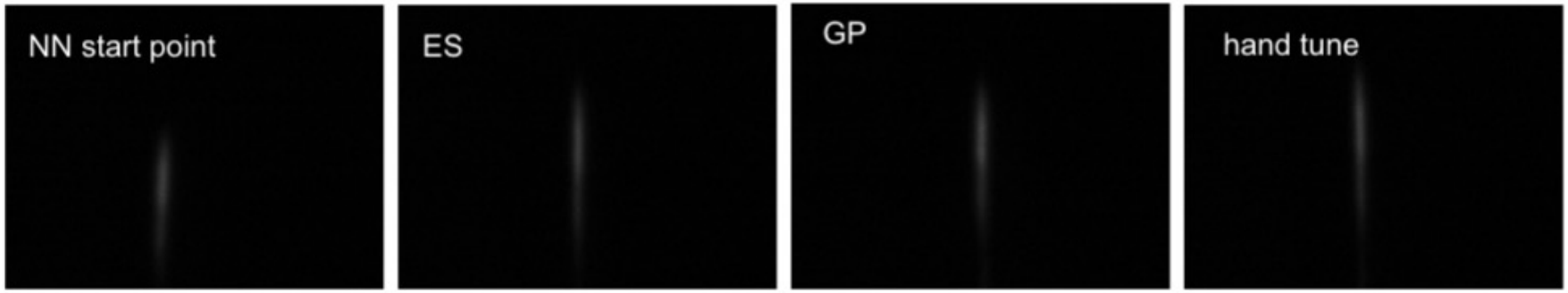
- Round to flat beam transforms are challenging to optimize
- Took measured scan data at Pegasus (UCLA)
- Trained neural network model to predict fits to beam image
- Tested online multi-objective optimization over model (3 quad settings) given present readings of other inputs





Results are for one full day after last training data

Can use neural network to provide first guess at solution, then fine tune with other methods…



Hand-tuning in seconds vs. tens of minutes

Significant boost in convergence speed for other algorithms

E. Cropp et al., in preparation

Differentiable Simulators

Several tiers of simulations used in accelerators:

- Simple transport matrices
- Particle-in-cell simulations
- Challenging cases (e.g. coherent synchrotron radiation, FEL process)

Use cases:

- Back-track the beam from end to starting conditions
- Gradient-based optimization of design / setups
- Model calibration to machine
- Support optimization methods that use gradients from simulation (e.g. Hessian to inform GP kernel in Bayesian optimization)

Still a relatively unexplored area in accelerators

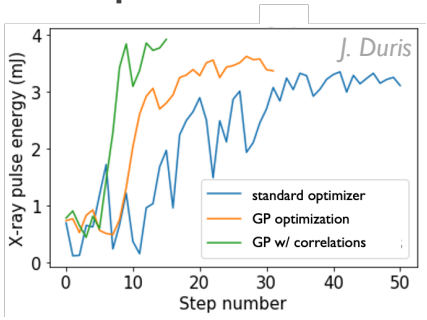
Simple example: code lie algebra used for particle tracking in rings into autodiff code
https://journals.aps.org/prab/abstract/10.1103/PhysRevAccelBeams.23.074601

ML Future Directions / Needs for Accelerator R&D

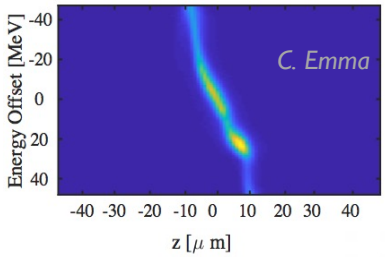
- Uncertainty quantification
  - Detect when model may not be accurate (e.g. outside training range)
  - Leverage for safe exploration of parameter space
- Active learning
  - Retraining to account for drift or adapt during search
  - Sampling strategies to efficiently explore large parameter space + generate training data (maximize information with the least samples)
- Efficient ways to handle high dimensional data:
  - Images, 6D phase space
  - More variables (full accelerator vs. small test cases)
- Physics-informed / constrained ML
  - Improve robustness / generalization to unseen regions of parameter space
  - Reduce need for additional data
  - Extract physics from measured data
- Differentiable Simulators
  - Wide range of types of simulation codes for accelerators (analytic matrix transport codes, particle-in-cell) → relatively unexplored area
- Interpretability
  - Important for ML-based tuning, identifying physic underpinning a prediction
- *Many shared challenges with other SciML domains → accelerators are unique test beds for these kinds of problems*

Future: tying together and scaling these to higher dimension, more extreme beams

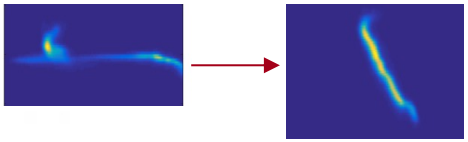
automated control + optimization

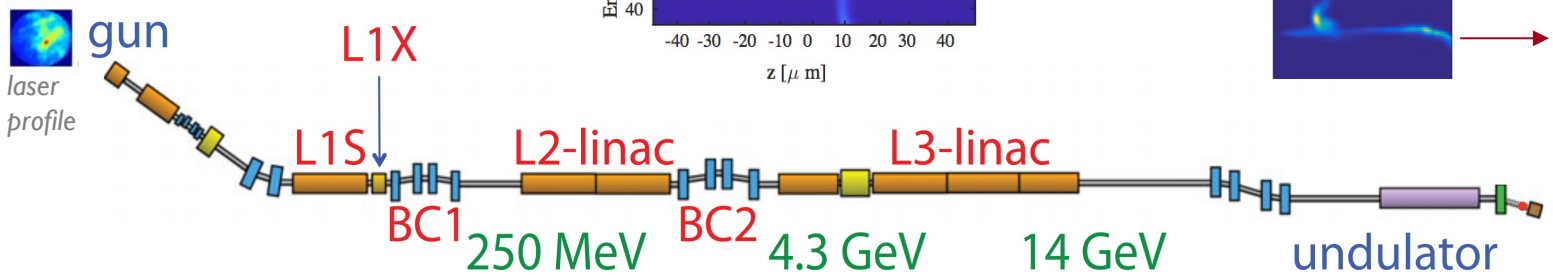


advanced diagnostics (reconstruct / analyze beam)

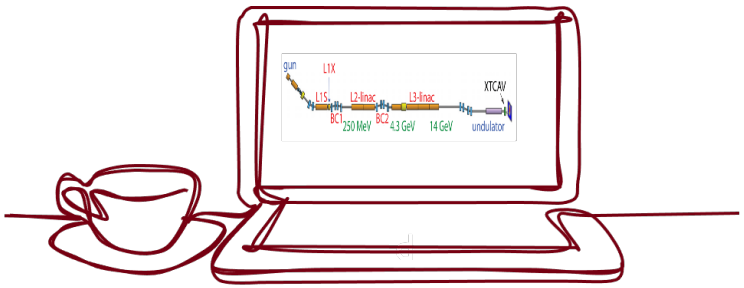


anomaly detection
failure prediction





incorporate physics information



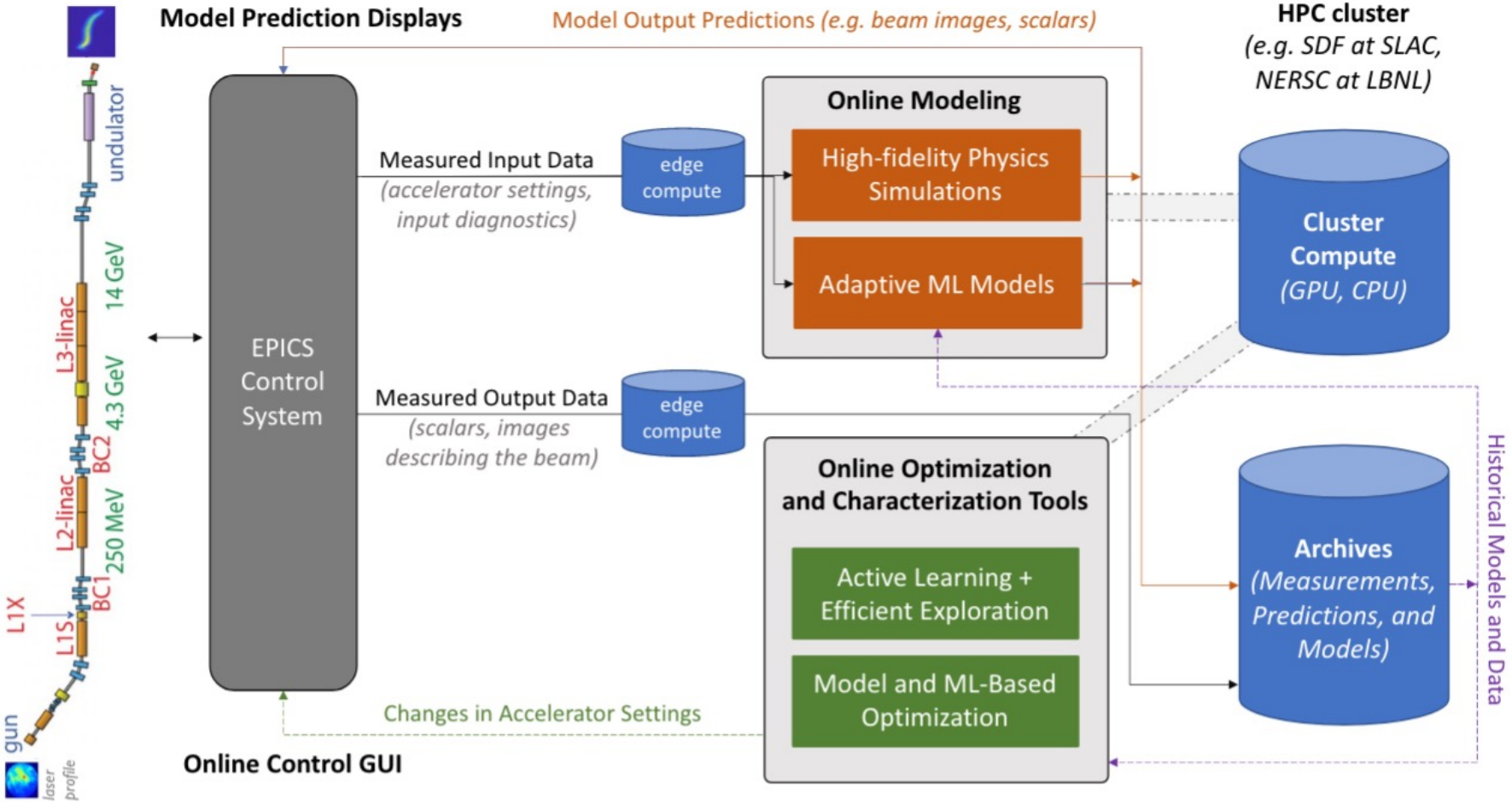
extract unexpected relationships
(feed into control / design)

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

+ need UQ for all

Thanks for your attention!

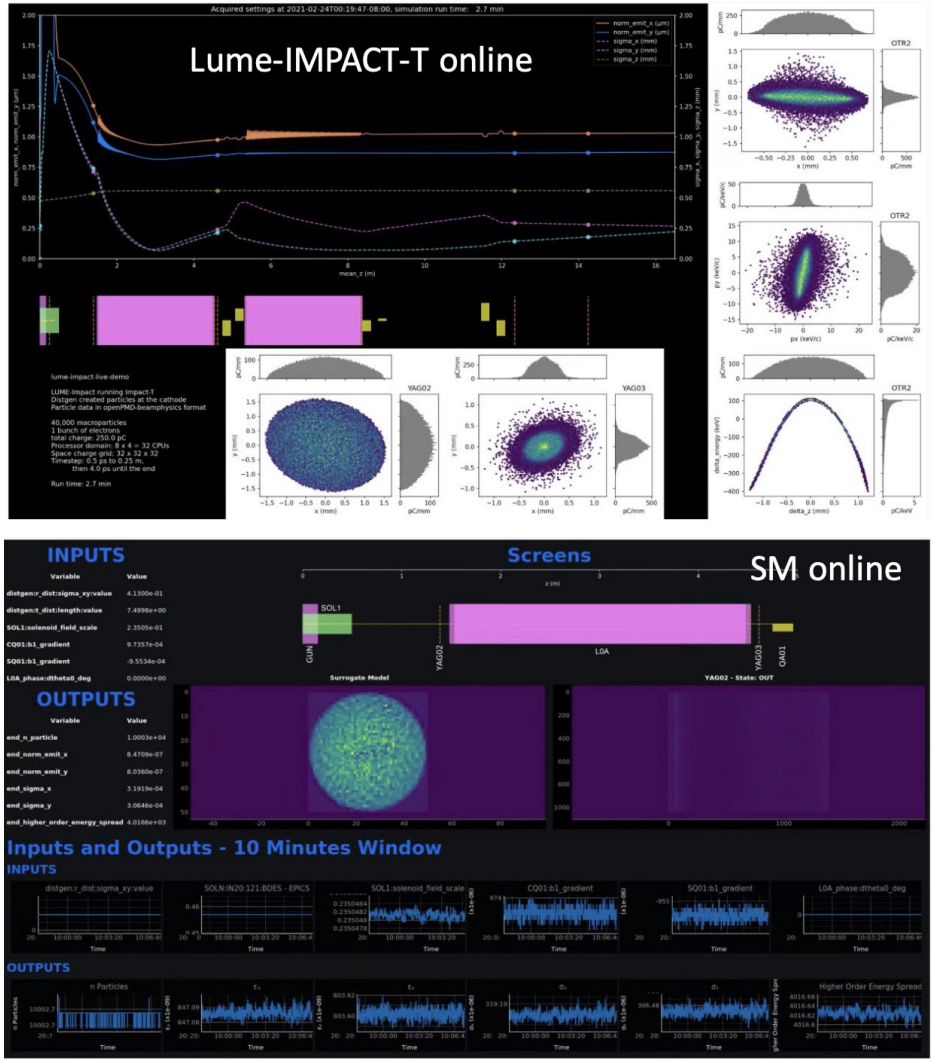
How to use all this together? Need dedicated investment in online compute and ML infrastructure



Example Prototype: Running Lume-IMPACT-T and Neural Network Model of LCLS Injector Online

- Lume-IMPACT-T online
  - Read EPICS PVs as input
  - Displays phase space predictions at OTR2 + line plots
  - Updates every 2 minutes (length of time for one IMPACT-T run)
  - https://www.youtube.com/watch?v=P6HYfpV6xXM

- SM at YAG02
  - Continuously updates
  - Serves output PVs
  - Will update to include OTR2, line plots soon
  - https://www.youtube.com/watch?v=FZny98PGcmU&feature=youtu.be



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