# AI/ML for Design and Operation of Particle Accelerators and Detectors

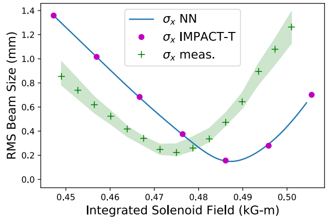
Snowmass Summer Study 2022
Computational Frontier

Auralee Edelen (on behalf of the whitepaper authors)
edelen@slac.stanford.edu

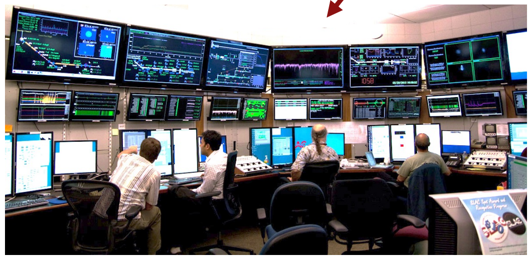
# Fast Simulations for Accelerators and Detectors

- Physics simulations for accelerators and detectors are powerful tools but are **computationally expensive**
- Numerous ML approaches being taken to create fast-executing representations of detailed simulations
  → **"surrogate models"** *(GANs, Bayesian NNs, VAEs, Simple NNs)*
- Faster predictions enable enhanced capabilities:
  - More comprehensive design + experiment planning
  - Discovery and online monitoring of differences between simulations and the real instrument *("calibration")*
  - Online predictions to provide more information for control and analysis
  - Model-informed online control
  - Future: "end-to-end" optimization from accelerator to detector/experiment

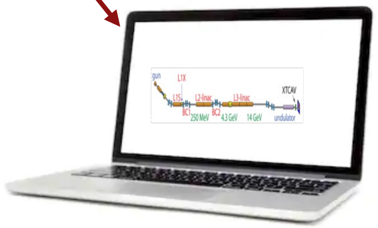
***ML brings prediction tools from HPC systems to online/local compute***



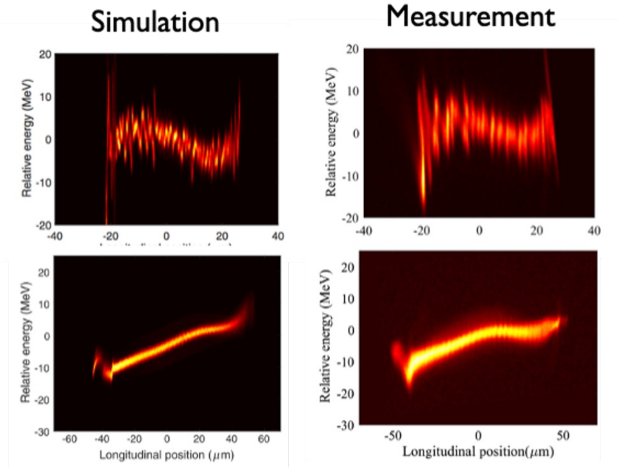

*Automatic Model Calibration*



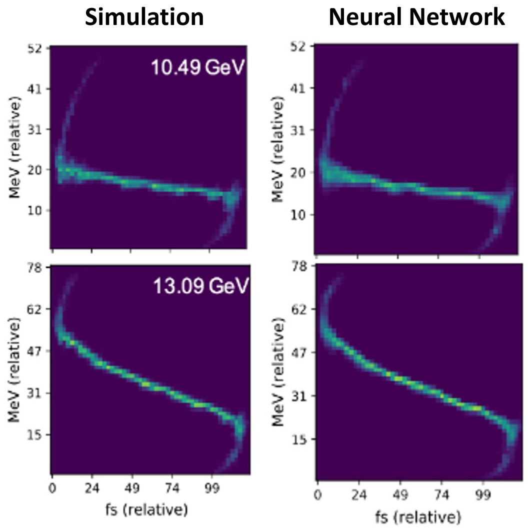
*Online prediction*
*Model-based control*



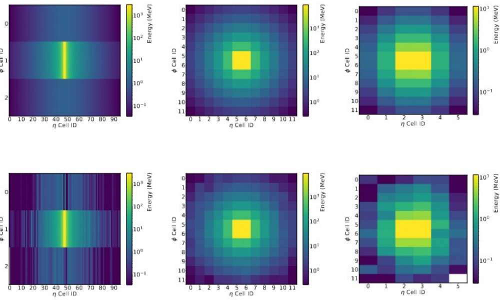
*Control prototyping*
*Experiment planning*

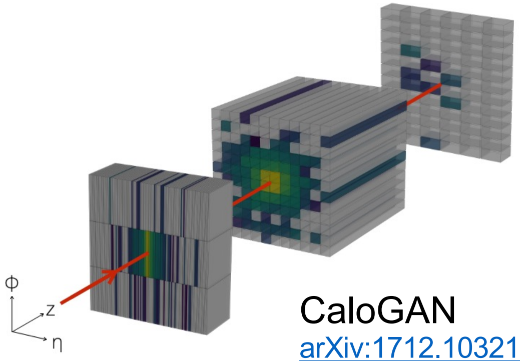



*Accelerator simulation: 10 hrs on thousands of cores at NERSC!*




*Neural network surrogate: < ms ($10^6$ speedup)*





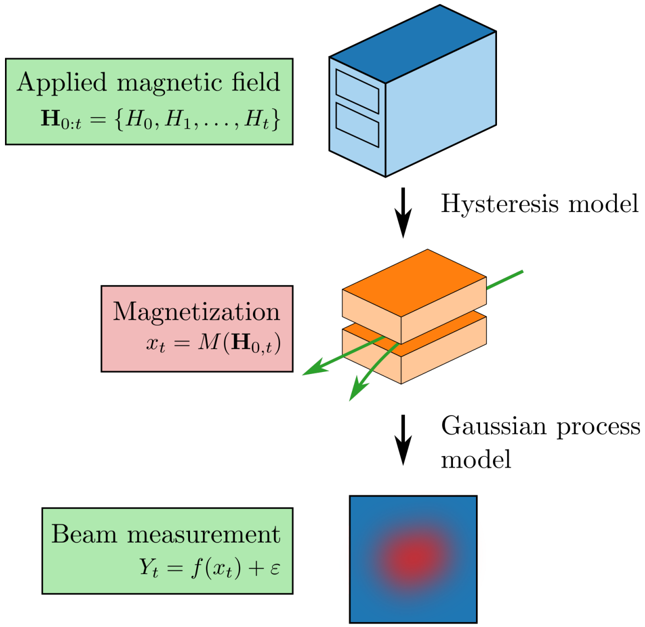

CaloGAN
[arXiv:1712.10321](arXiv:1712.10321)

*See accelerator community modeling WP: arXiv:2203.08335*

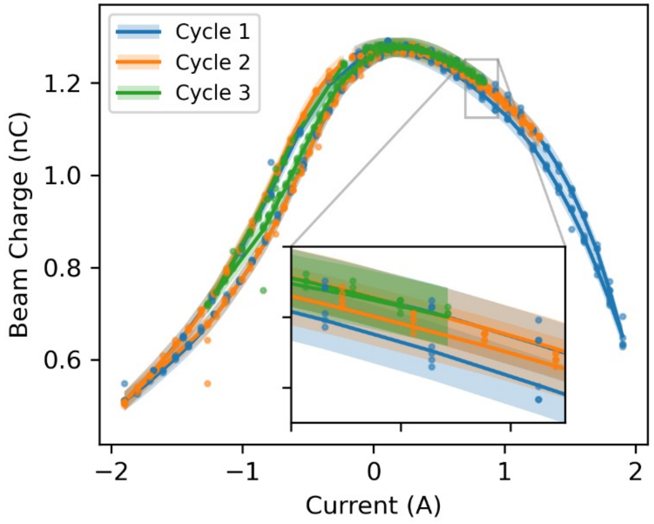
# Differentiable Physics Simulations and ML

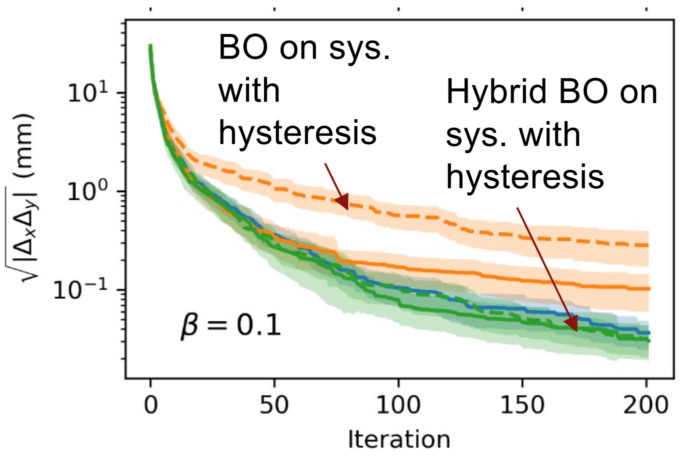
Modern ML uses gradients in learning → *differentiable physics sims enable modular combinations with ML components, analyses, etc.*

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




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



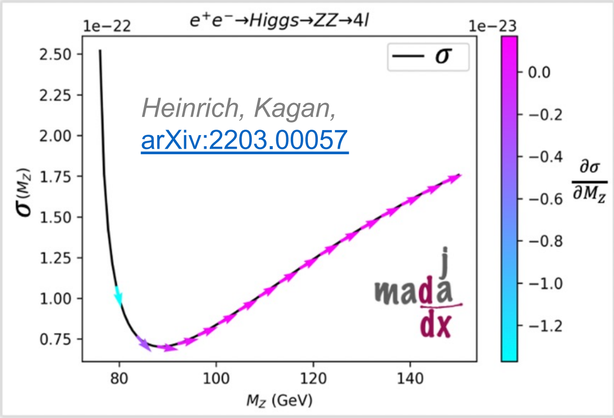




R. Roussel, et al., PRL, 2022, arXiv:2202.07747

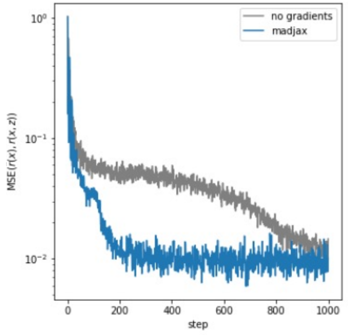
arXiv:2203.13818

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

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




*Differentiable matrix elements of high energy scattering processes*



# ML-Assisted Optimization and Characterization

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

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

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

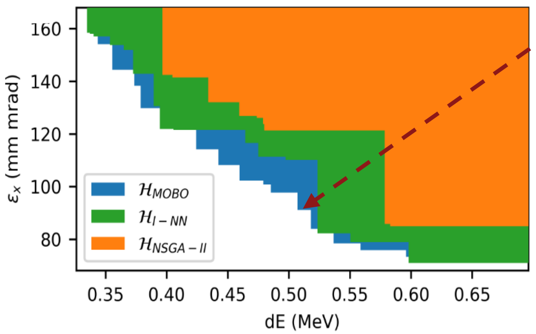
**Bayesian optimization / active learning / reinforcement learning**
→ All learn iteratively via online interaction with the system

next point to search

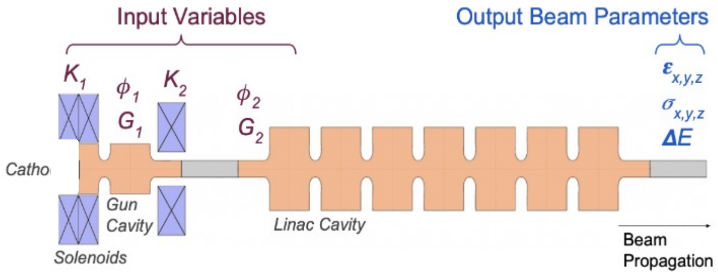
**Learning Algorithm** (GP model + Bayes. opt., iterative surrogate, RL agent)

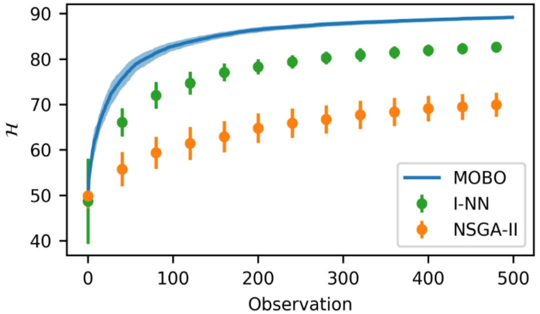
**System** (Simulation or Online Interaction)

data




Pareto front: optimal tradeoff between parameters of interest

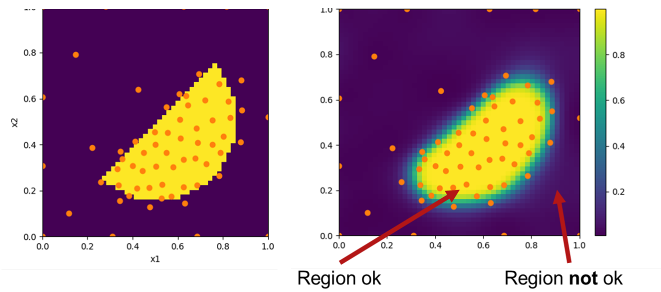



***Faster multi-objective optimization with Bayesian optimization and iterated surrogate models***

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

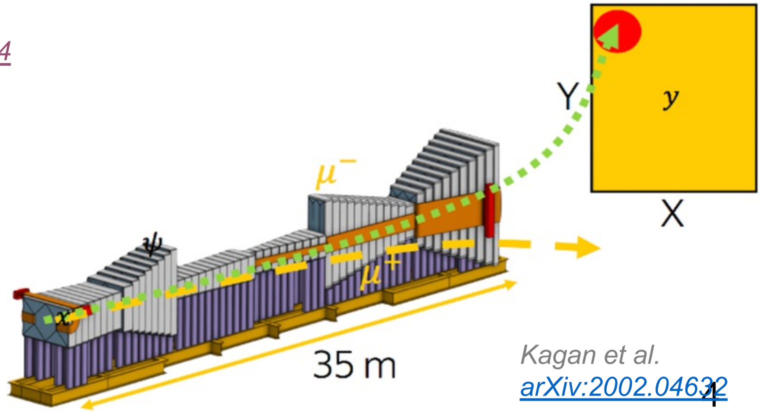
*A. Edelen et al., [arXiv:1903.07759]*




***Output constraints learned on-the-fly***

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

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

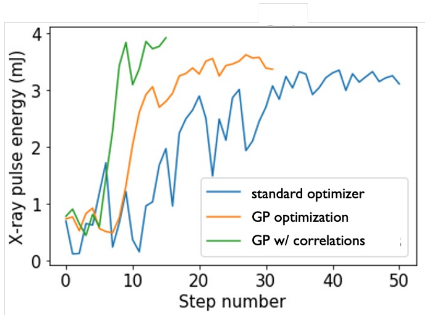



*Kagan et al. [arXiv:2002.04632]*

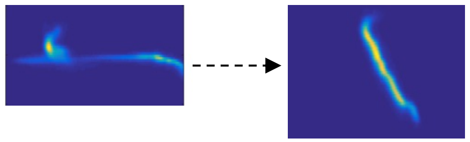
# Broad Set of Areas for ML to Impact Operation

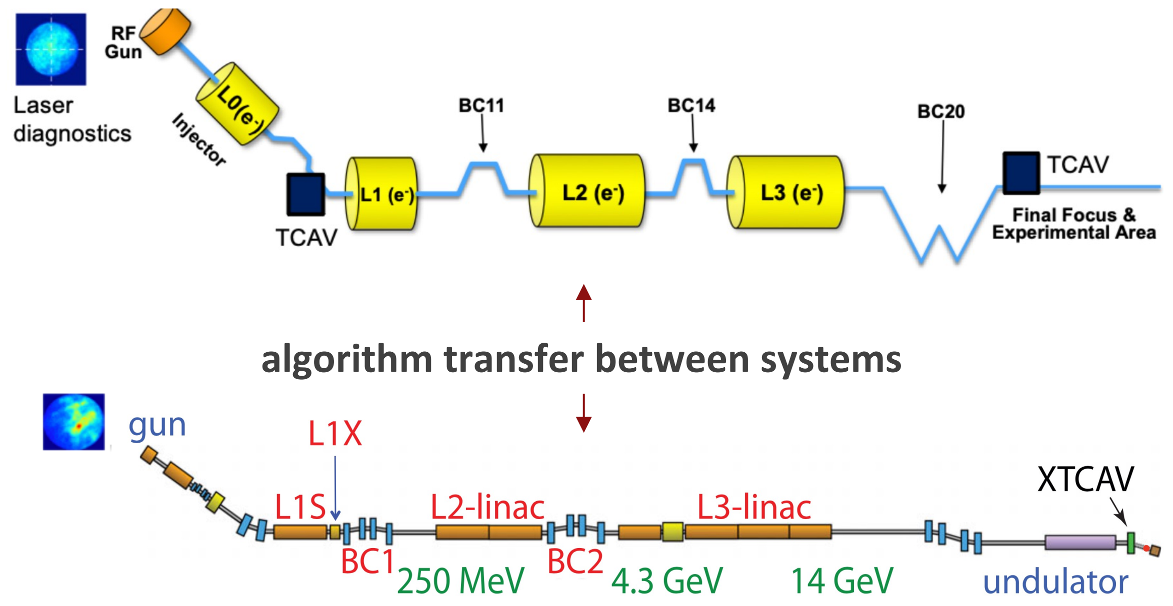
**Data reduction/rejection** *(kHz/MHz data streams)*
**Event triggering**

**automated control + optimization**

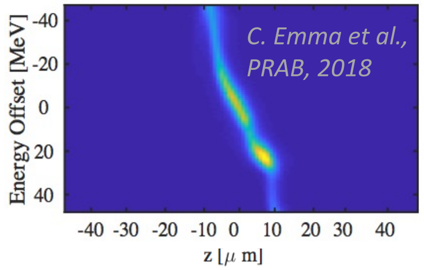



*J. Duris et al., PRL, 2020*

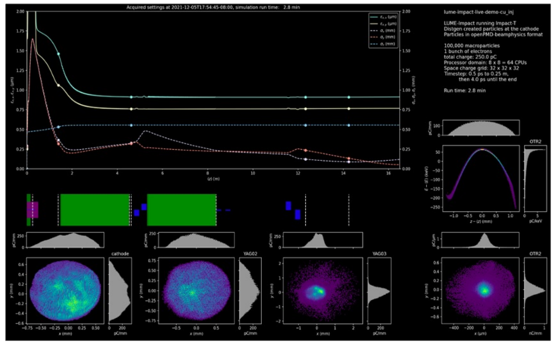


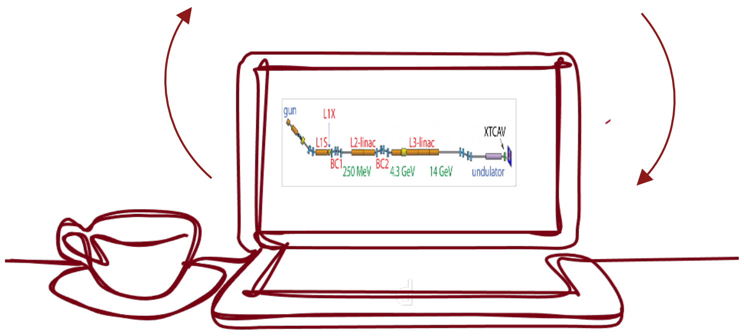



**ML-enhanced diagnostics**
*(provide insight at faster rate, at higher resolution, non-invasively)*



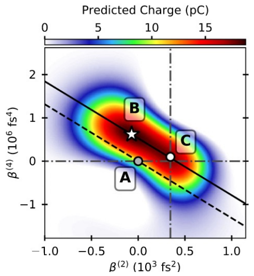

**anomaly detection**
**failure prediction**
*(plan maintenance; alert to changes in machine; alert to interesting science)*





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

R. Shaloo et al.
arXiv:2007.14340



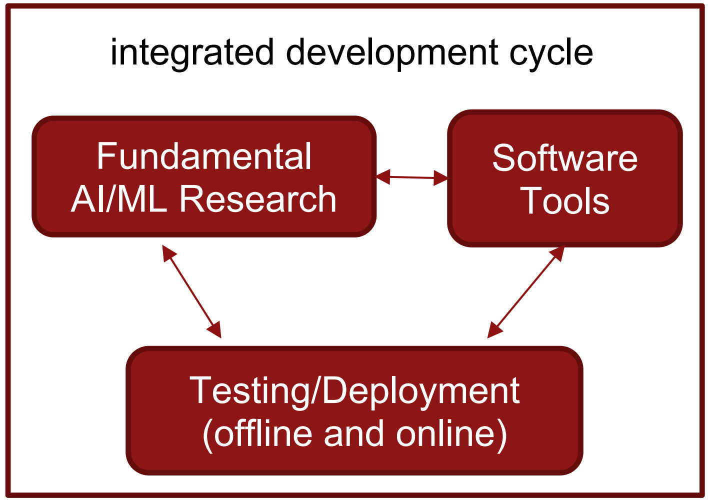

**digital twins + online modeling**
*(fast sims, differentiable sims, model calibration, model adaptation)*

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

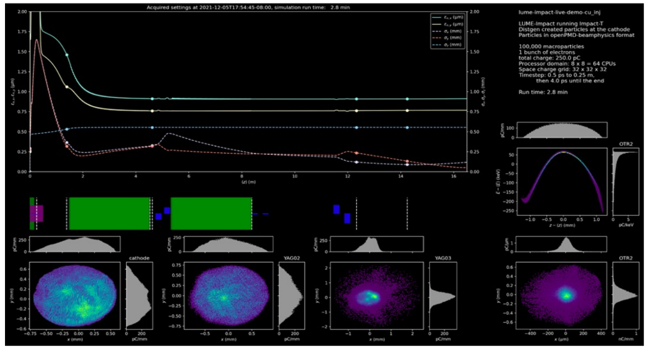
# Conclusions/Recommendations

- AI/ML are enabling broad-sweeping shifts in how accelerators and detectors are designed and operated
- Spans search for new experimental setups/designs, optimization of online systems, "seeing into" instrument operation in real time, opportunities for co-design with ML-based analysis and control techniques (e.g. placement and type of diagnostics given ML-based reconstruction capabilities)
- Modularly combining physics modeling and ML is promising area of research to improve generalizability, reduce reliance on large data sets, improve interpretability
- Broad need also for edge compute + ML (fast prediction, feedback, event triggering and data rejection/reduction)
- Other areas that need investment include improving uncertainty quantification and domain adaptation, combining physics and ML (differentiable simulations, inductive biases), online adaptation and calibration of models/controllers (both predictions and uncertainties), and combining/scaling up methods into a full integrated ecosystem
- Challenges/needs for design optimization and operation are broadly similar across systems: need investment in open-source, community software tools and ecosystems → software/firmware infrastructure in addition to underlying ML methods

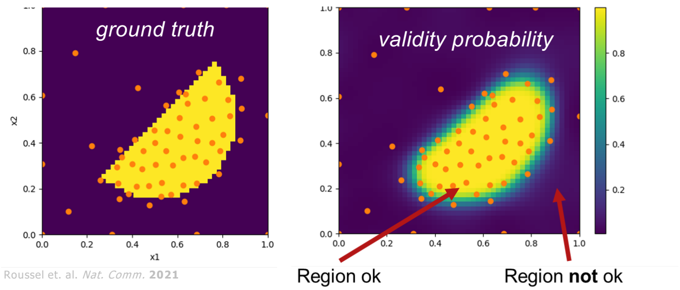
# Backups

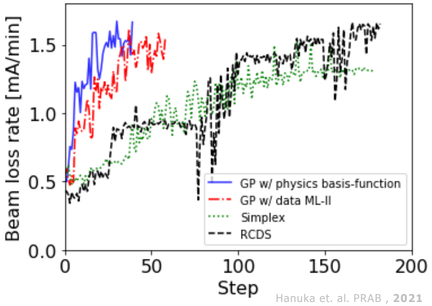



**Online prediction** with physics sims and **fast/accurate ML models**



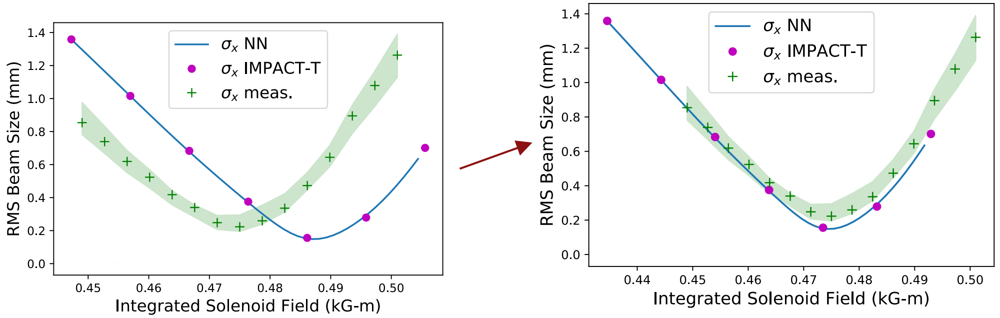
**Efficient optimization and characterization** *(useful also for simulation exploration/design, data generation)*



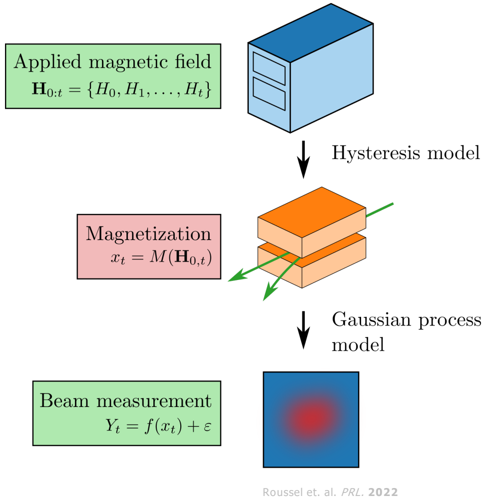




***Output constraints learned on-the-fly***

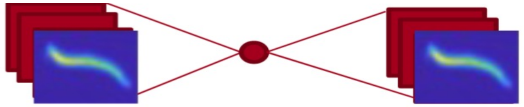
**Adaptation of models** and **identification of sources of deviation** between simulations and as-built machine




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




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

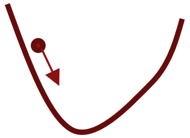


**Software packages and standards** for data generation, modeling, and optimization (*LUME, xopt*)

## Approaches can leverage different amounts of data/previous knowledge: suitable in different circumstances

less ← assumed knowledge of machine → more

### Model-Free Optimization

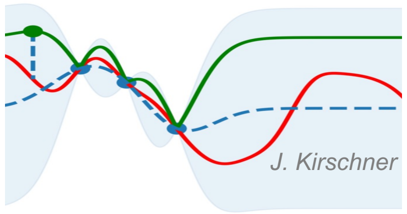


*Observe performance change after a setting adjustment*

*→ estimate direction or apply heuristics toward improvement*

gradient descent, genetic algorithms, simplex

### Model-guided Optimization

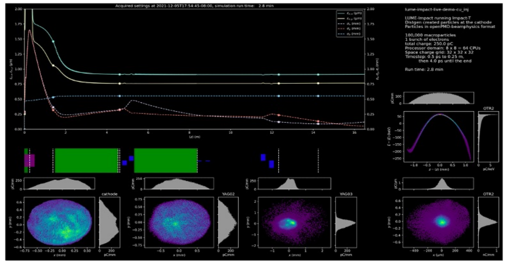



*Update a model at each step*

*→ use model to help select the next point*

Bayesian optimization
reinforcement learning

### Global Modeling



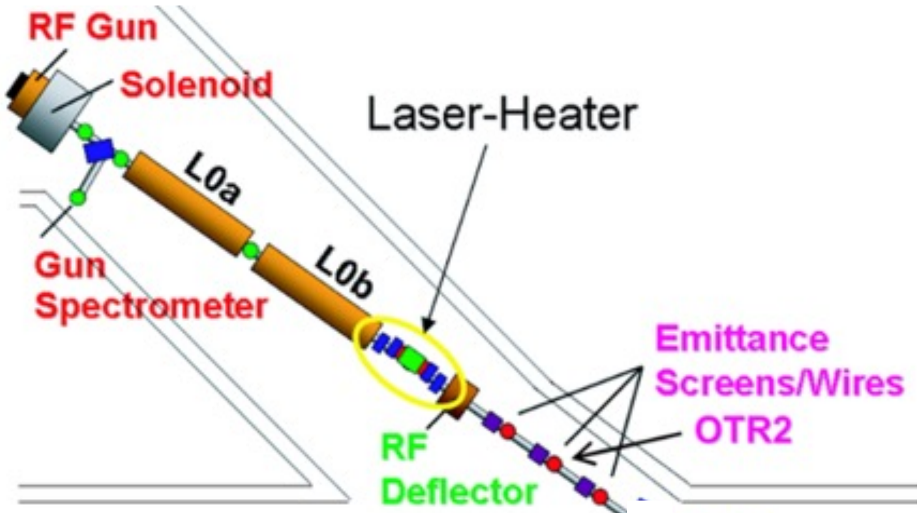
*Make fast system model*

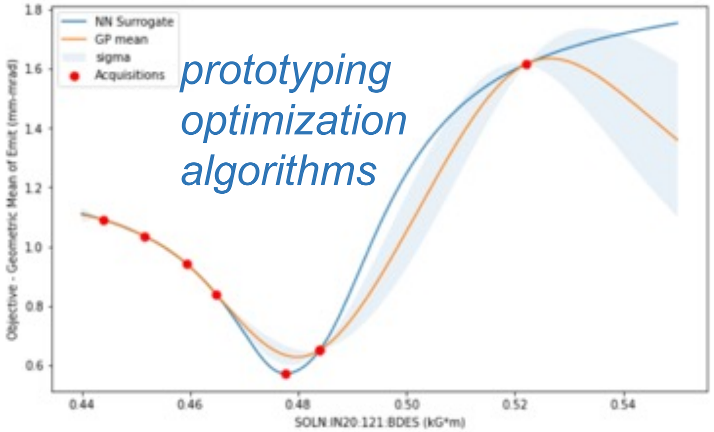
*→ provide initial guess (i.e. warm start) for settings*

ML system models +
inverse models

# Example: Injector Surrogate Models

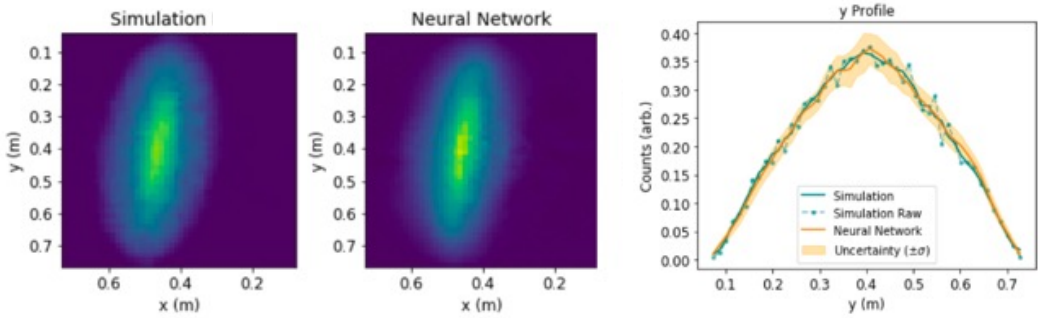
- ML models trained on physics simulations
- Inputs sampled widely across valid ranges
- **Used to develop/prototype new algorithms before testing online at FACET-II and LCLS** *e.g. new Bayesian optimization methods, adaptive emittance measurement*



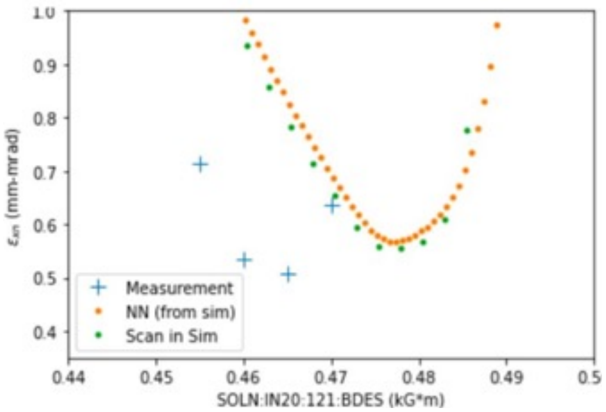



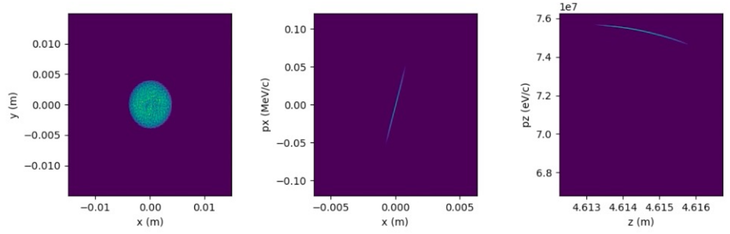

*prototyping optimization algorithms*

*ML model provides accurate replication of simulation*




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





*interactive model widget and visualization tools*

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