Machine Learning Applications at SLAC

Jan 12, 2024

D. Ratner

SLAC National Accelerator Laboratory

U.S. DEPARTMENT OF ENERGY

Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

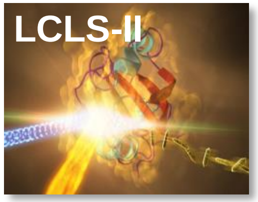
Outline

1. **Online control of x-ray facilities**
2. **Anomaly detection**
3. **ML for inverse problems**

(But there are also many other applications across SLAC)

SLAC’s mission revolves around major scientific facilities

Research projects across biology, chemistry, geology, physics, energy sciences, materials science, and more



FACET-II Facility for Advanced Accelerator Experimental Tests

CMB-S4 Next Generation CMB Experiment

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

LZ

SSRL

LSST Large Synoptic Survey Telescope

ATLAS EXPERIMENT

Particle accelerators for x-ray science

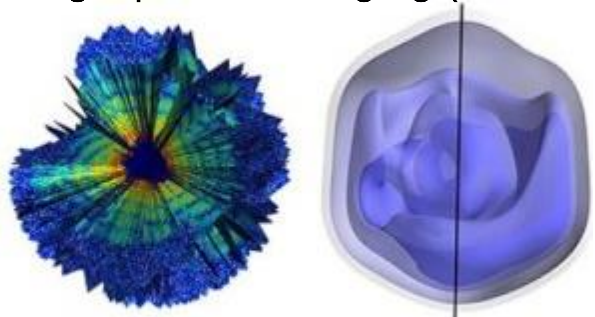
Vera Rubin Observatory

Cryo-EM facilities for COVID research

Introduction to X-ray Free-Electron Lasers (XFELs)

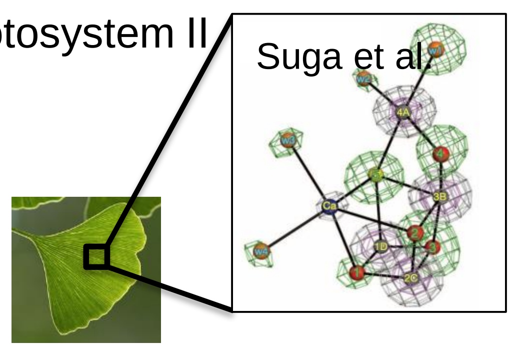
X-ray free-electron lasers: Accelerator driven x-ray sources to study ultrafast, ultrasmall scale phenomena

Single particle imaging (Mimivirus)

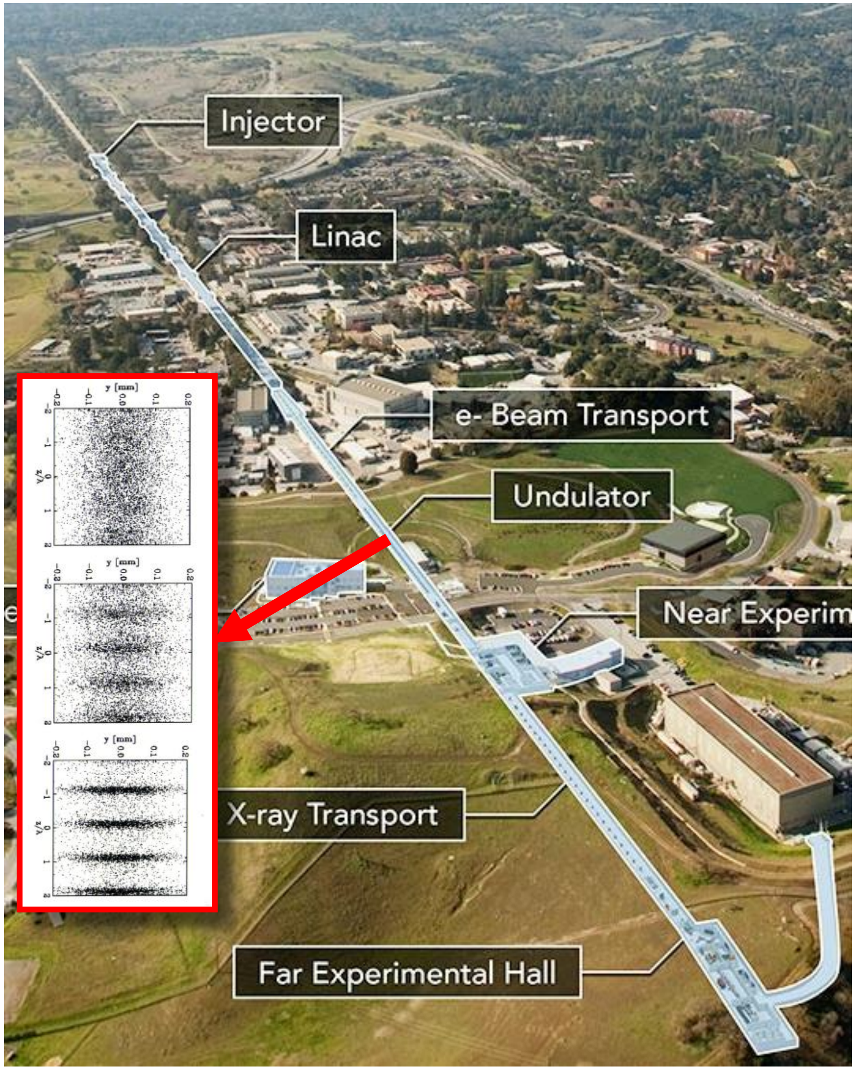


Ekeberg et al.,

Photosystem II



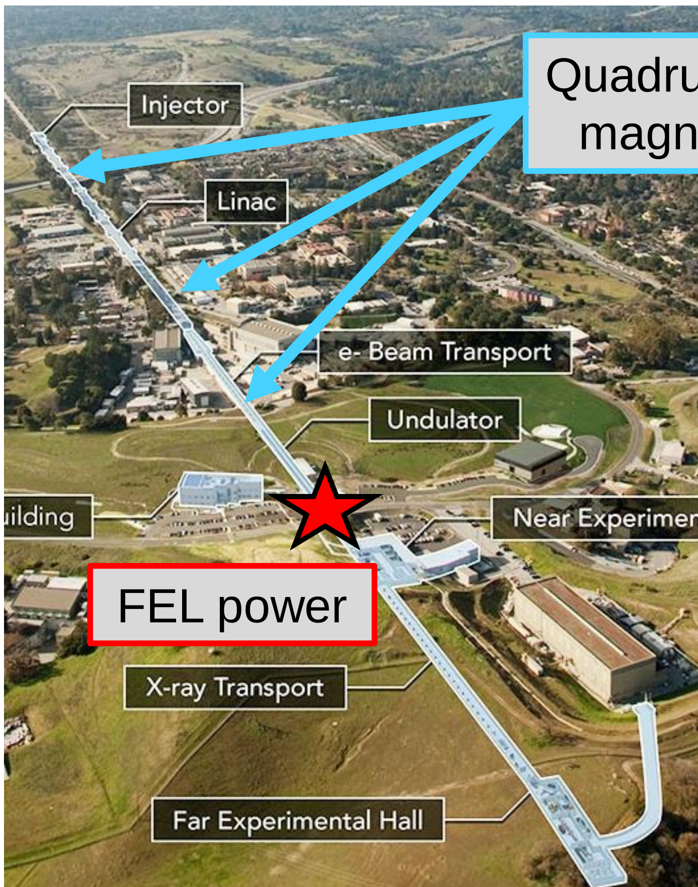
Shock waves in extreme conditions

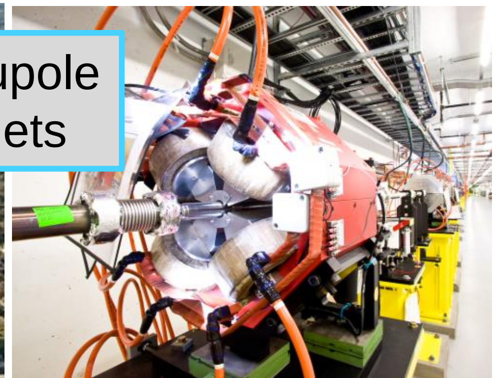


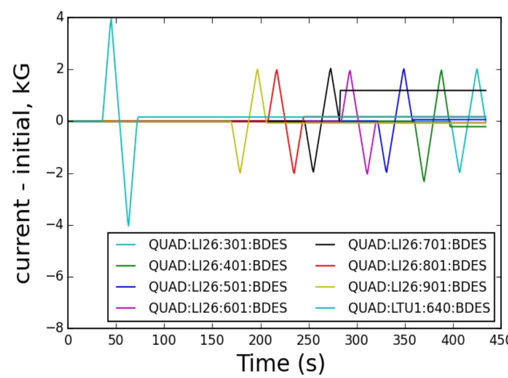
Online optimization

XFEL tuning:

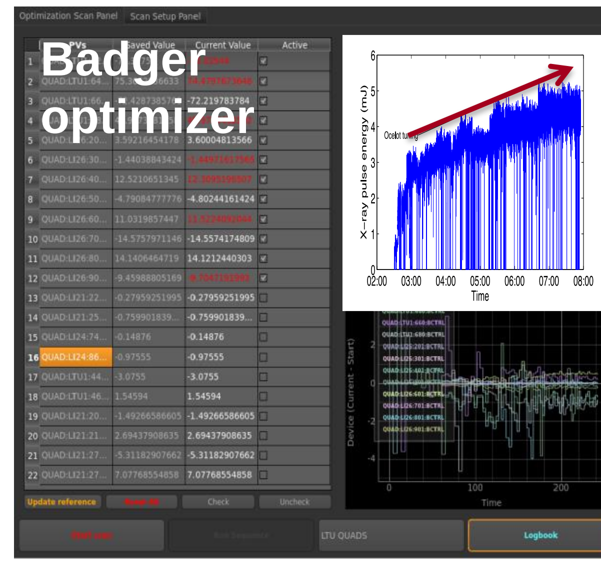
- XFELs are instabilities ➔ very sensitive to initial conditions
- In total 500 hours/year on single task of quad tuning







Automate (e.g. simplex)



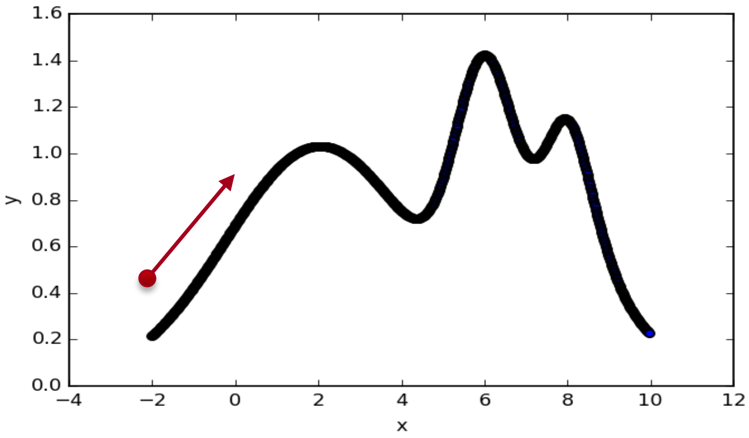
Online optimization: Bayesian optimization

Model-based optimization

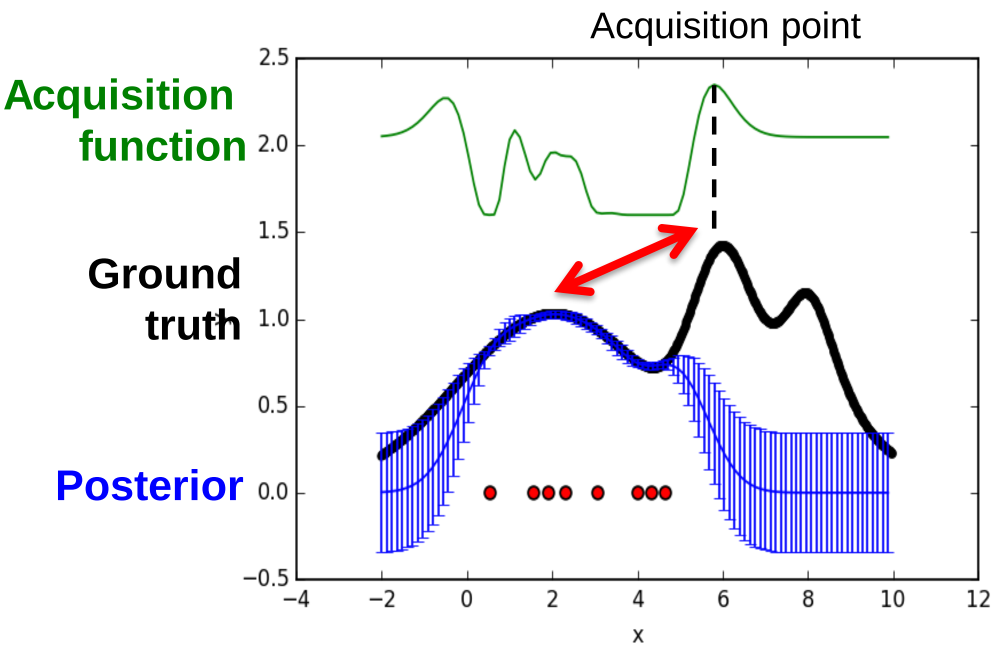
Advantage 1: Balance “exploitation vs. exploration”

➔ Find global maximum

Gradient optimizer



Bayesian optimizer

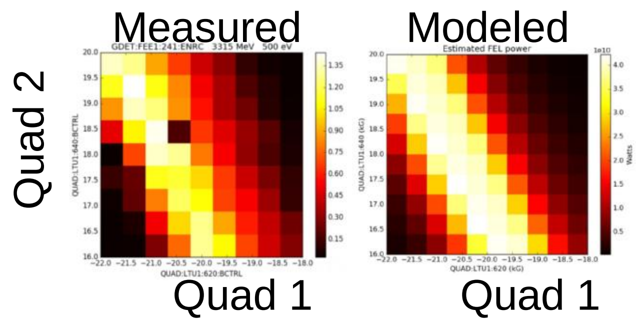


Online optimization: Bayesian optimization

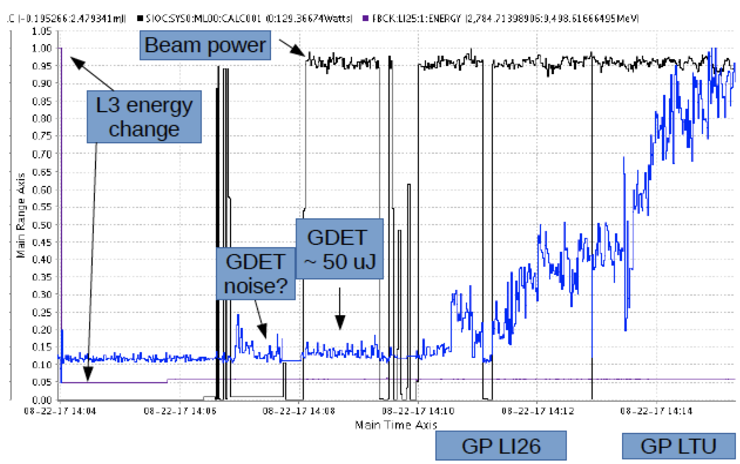
Model-based optimization

Advantage 2: Existence of model enables use of physics, archived data

e.g. learning correlations in data improves modeling



- 5x faster than simplex
- Can tune from pure noise

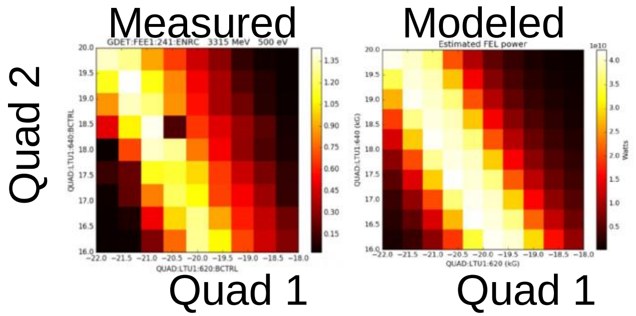


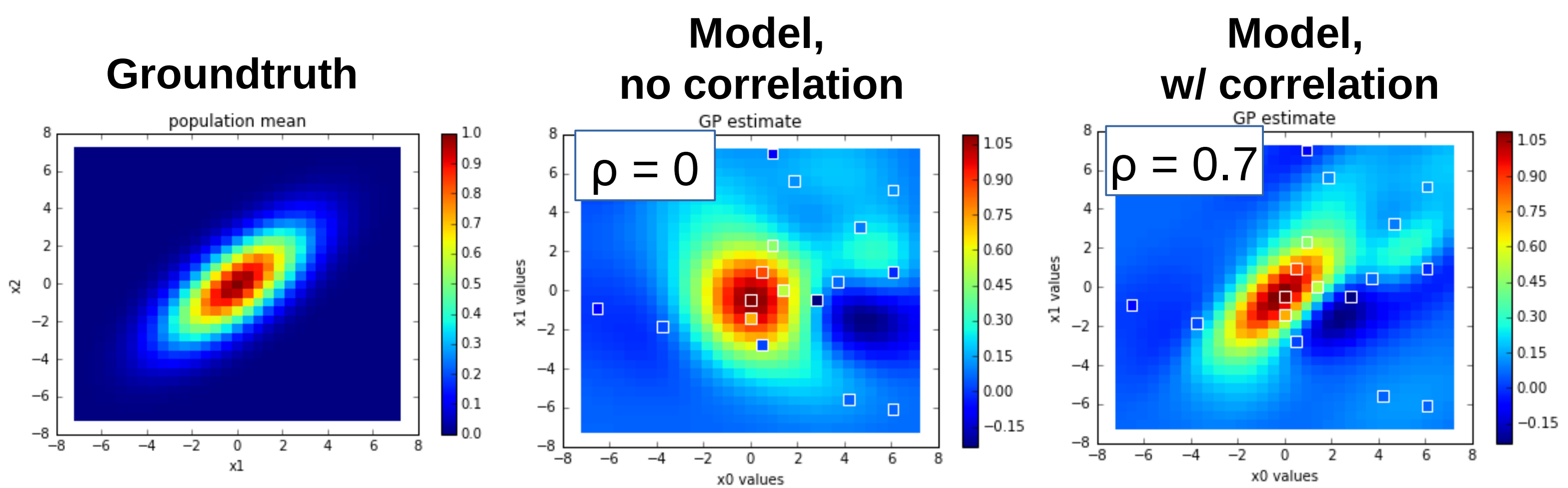
Bayesian Optimization of a FEL, Duris et al., PRL, 2020

Model-based optimization

Advantage 2: Existence of model enables use of physics, archived data

e.g. learning correlations in data improves modeling





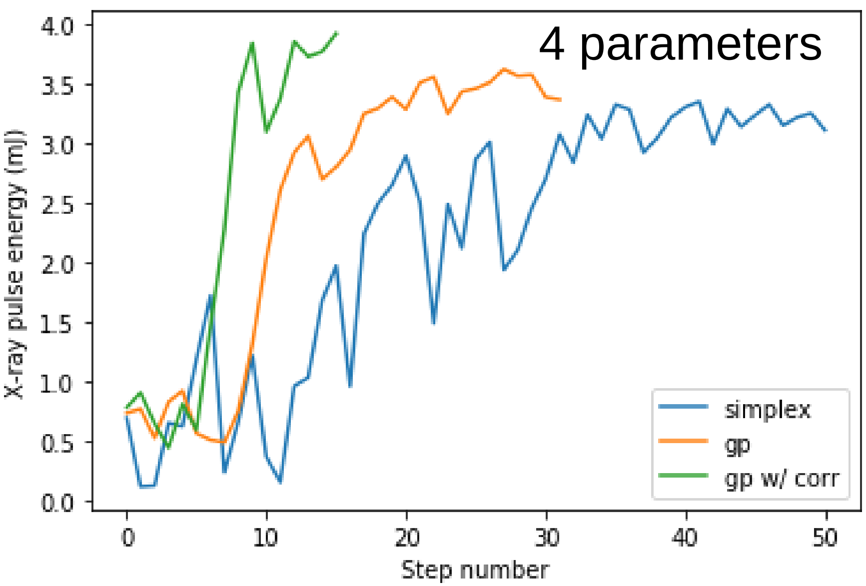
Bayesian Optimization of a FEL, Duris et al., PRL, 2020

Online optimization: Bayesian optimization

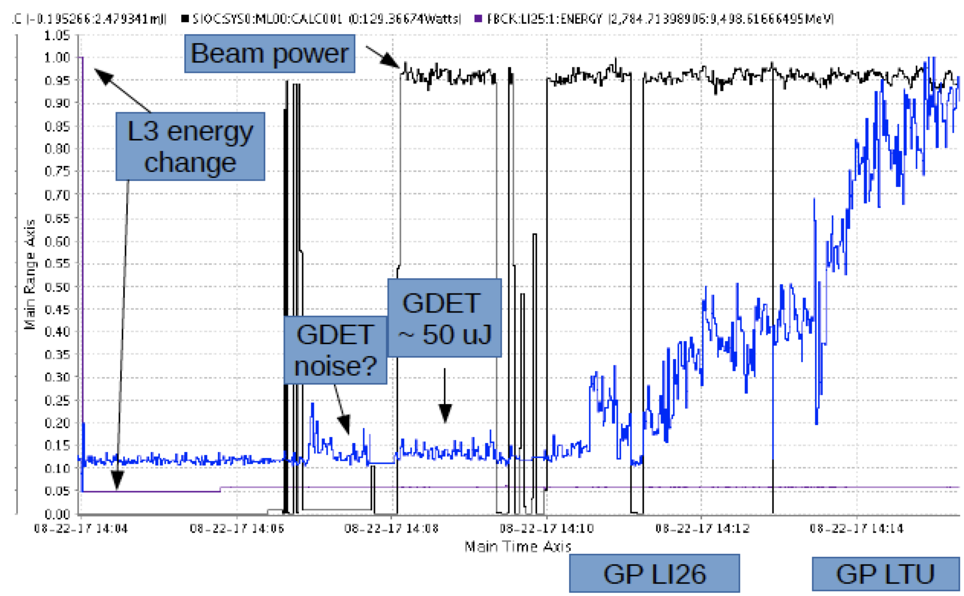
Model-based optimization

Upshot: ➔ Faster tuning (factor of 4 vs. simplex)
➔ More robust tuning (e.g. tune from noise)

Example 1: Fast tuneup in high dimensions



Example 2: tuning quadrupoles from pure noise



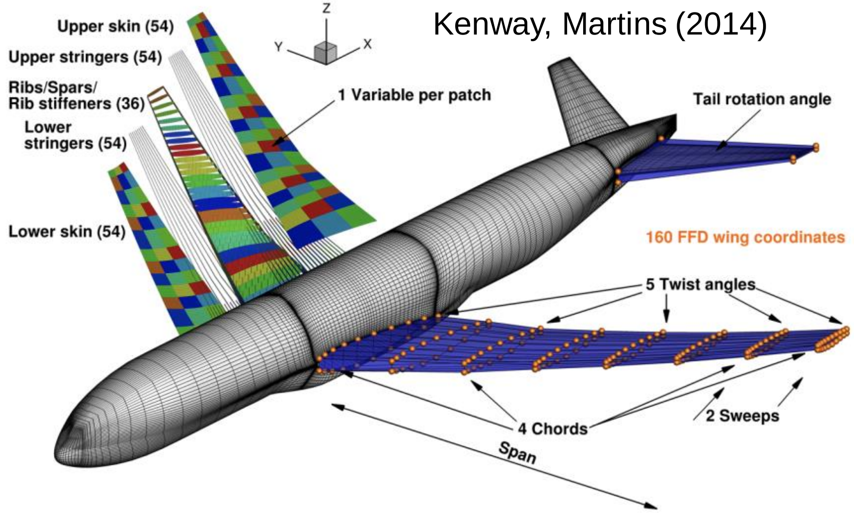
Bayesian Optimization of a FEL, Duris et al., PRL, 2020

Online optimization: multi-point optimization

Classic problem from aeronautics design:
how to optimize an airplane wing?

1. Choose a design
2. Simulate range of conditions
3. Combine into single metric / multiple metrics
4. Loop to step 1

Very slow process!
Very information inefficient!

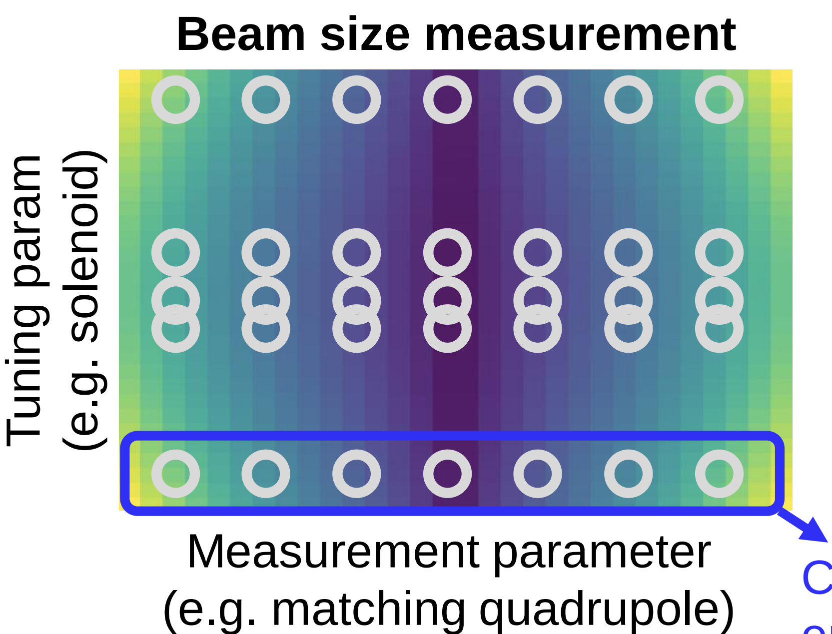


Kenway, Martins (2014)

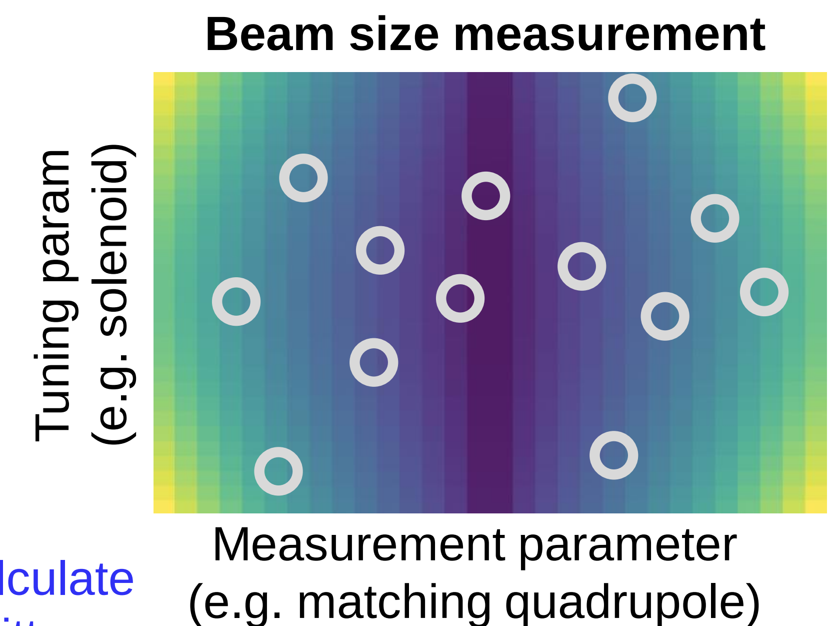
Image courtesy FAA

Online optimization: multi-point optimization

Accelerator example: emittance optimization, i.e. beam quality



Calculate emittance



BO has multiple problems:

- Sampling inefficient
- Information inefficient
- Difficult to model

Models beam size efficiently…

…but we want to minimize emittance, not beam size

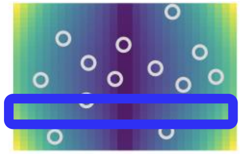
W. Neiswanger, S. Miskovich, A. Edelen

Online optimization: multi-point optimization

Bayesian Algorithmic Execution (BAX)

Standard BO: model and find optimal point in black box function $f(x)$

BAX: model $f(x)$, and find optimal output of $A[f(x)]$, for known algorithm $A[\cdot]$



Emittance example:
- Model beam size behavior, $f(x)$
- Algorithm, $A[\cdot]$, outputs minimum emittance from $f(x)$
- Measure new beam size that has the biggest impact on output of $A[\cdot]$

Note: We never actually measure emittance… just calculate from a virtual model!

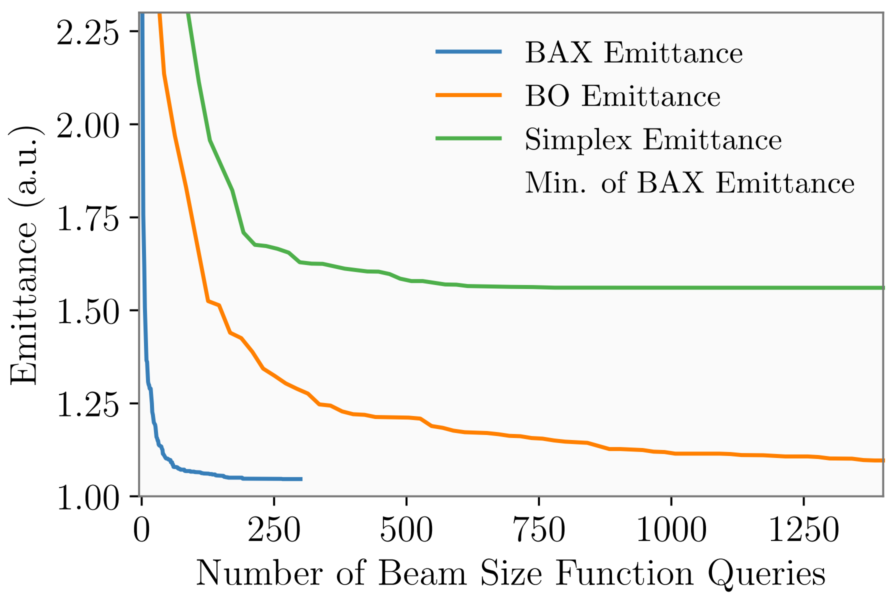
(For computer scientists, acquisition function calculated from mutual information of algorithm output and model posterior)

➔ **BAX:** *https://willieneis.github.io/bax-website/* W. Neiswanger et al., ICML, 2021

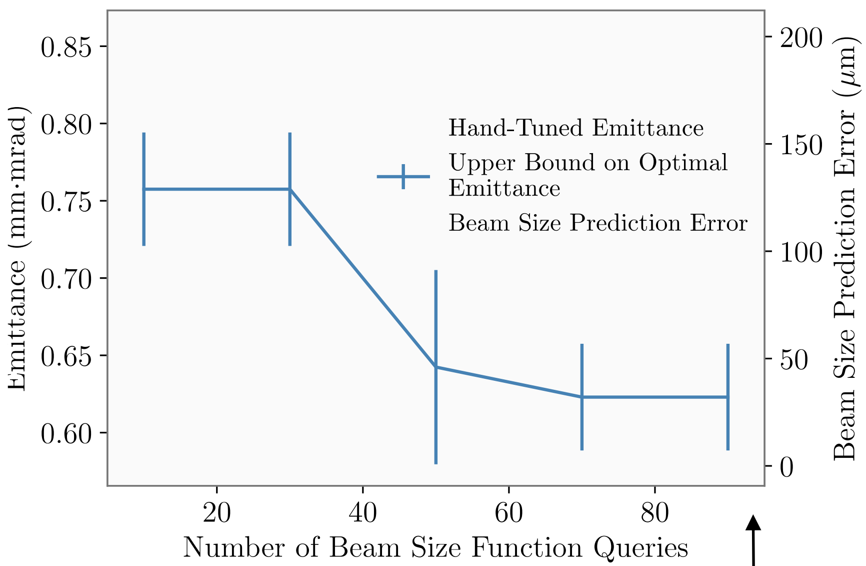
Online optimization: multi-point optimization

Accelerator example: emittance optimization, i.e. beam quality

Simulated optimization, LCLS Injector



Experimental optimization, LCLS Injector



~2 minutes

W. Neiswanger, S. Miskovich, A. Edelen

Design optimization: dynamic/momentum aperture

Dynamic aperture optimization for storage rings

Goal: maximize dynamic aperture (i.e. the blue area):

Larger aperture ➔ longer lifetime ➔ higher luminosity/brightness

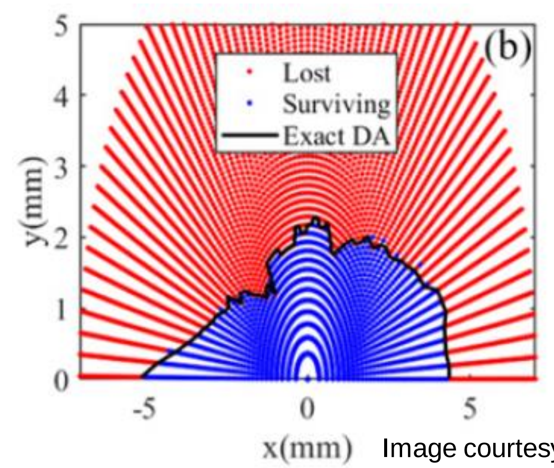


Image courtesy J. Wan, Y. Jiao

- Each calculation of the objective function (blue area) requires 1000s of simulations
- MultipointBAX: choose individual particles to simulate, not full DA scans. Potentially *orders of magnitude faster*
- *…*requires NN modeling! (New topic of research.)

Outline

1. Online control of x-ray facilities
2. **Anomaly detection**
3. ML for inverse problems

Anomaly Detection

Failure prediction at accelerators:

- Failures cause downtime, degrade performance
- 1000s of subsystems → frequent failures
- 200k variables to monitor at LCLS → impossible to do manually
- Big data, but nearly none of it labeled


SLAC Linear Accelerator and 280 Overpass


SLAC RF Stations

Anomaly Detection: RF Station Faults

Case study: RF station faults

- RF stations provide power for acceleration
- Most common fault at LCLS (1000s/year)
- Degrades performance even if no downtime
- Years of data, only dozens labeled

Anomaly Detection: RF Station Faults

SLAC

Existing solution #1: Thresholding on subsystem amplitudes

Readbacks of station power condensed to "status bit" for each of 80 stations

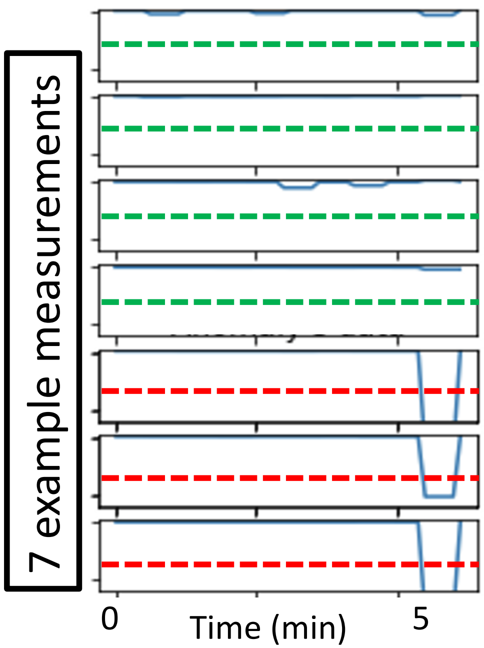
Status bit '0' if healthy, and flips to '1' if RF amplitude changes more than 2%

Status bit has multiple problems:

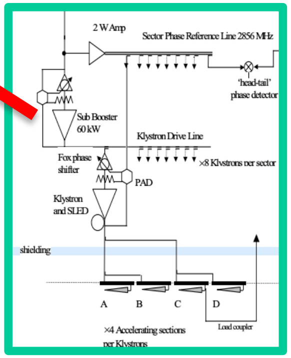
- Too simplistic (only looks for sudden drops)
- Too permissive (misses subtle faults)
- Too noisy (lots of false positives)

70% of alarms are false (i.e. poor precision)

RF station amplitude for 1 station



RF station diagram



Recall = TP/(TP+FN)   Precision = TP/(TP+FP)

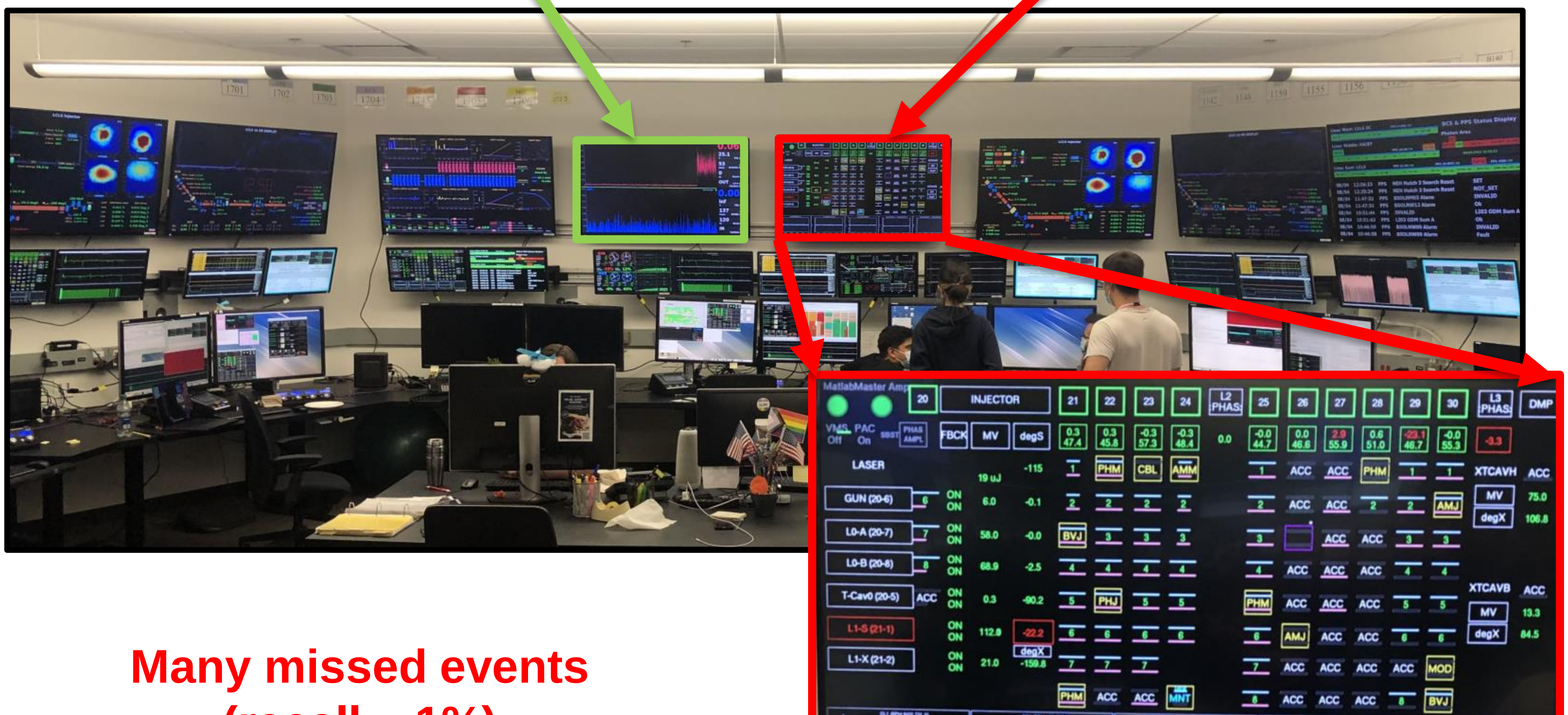
TP = true positive, FP = false positive, FN = false negative

Anomaly Detection: RF Station Faults

Existing solution #2: Manual, beam-based anomaly detection

Operators watch for a drop in X-ray power here

After observing a drop, search for RF stations with warning/fail status

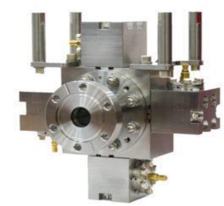


Many missed events (recall ~ 1%)

Anomaly Detection: RF Station Faults

New approach: Beam-based algorithm inspired by operators

RF stations



Beam position monitors

Question: How do we train these algorithms with no labels?

Algorithm #1 looks for subsystem anomalies

Algorithm #2 looks at final beam quality

Key insight: A 'good' pair of algorithms will be self-consistent

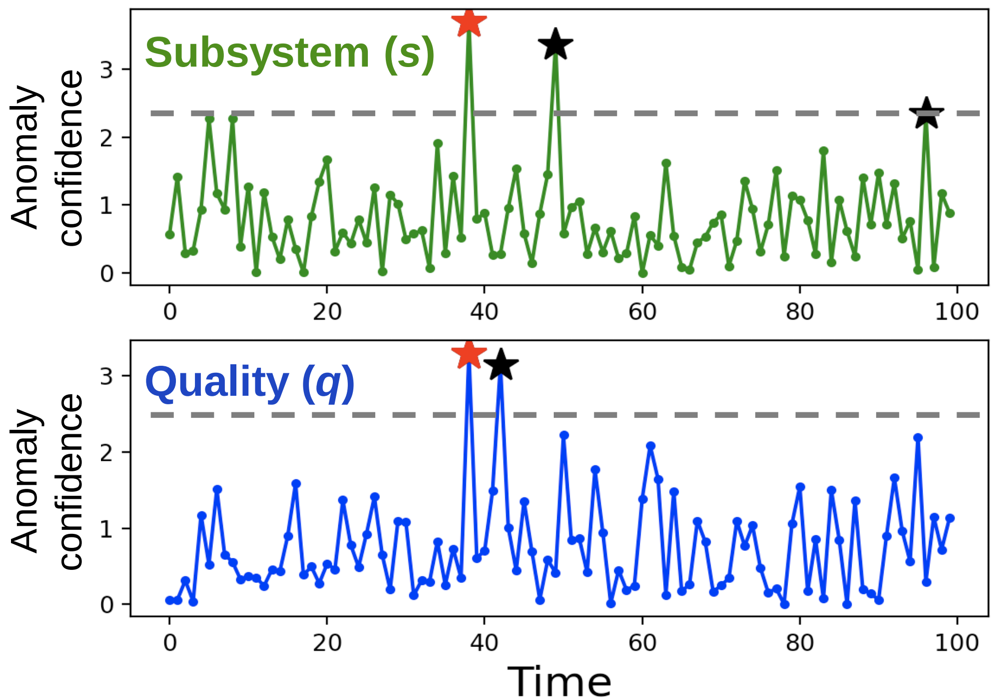
Simultaneous anomalies trigger alarm

Coincident Anomaly Detection (CoAD)

Consider two algorithms making predictions on two streams of data

Unknown machine state → Subsystem → Algorithm 1 → Prediction → Self-Consistency metric

Unknown machine state → Beam quality → Algorithm 2 → Prediction → Self-Consistency metric



- **Red** stars = consistent
  black stars = inconsistent

- More **red** = higher recall;
  less **black** = higher precision

FP rate = product of black star rates

TP rate = red star rate – FP rate

➔ Unsupervised precision/recall

Coincident Anomaly Detection (CoAD)

Consider two algorithms making predictions on two streams of data

We estimate **unlabeled** version of recall/precision from coincidence

Estimates can be **differentiable** ➔ can train neural networks!
i.e. a loss function, not just a metric

Formally the loss function is defined for NN outputs f_1, f_2:

$$\hat{F}_\beta = \frac{(1+\beta^2)(J-D)}{J+\alpha\beta^2} \leq F_\beta = \frac{(1+\beta^2)(PR)}{R+\beta^2 P}$$

$J = \mu_{12} = \mathbb{E}(f_1 f_2)$: red star rate

$D \stackrel{\text{def}}{=} \frac{\mu_1 - \mu_{12}}{1-\mu_2}\frac{\mu_2 - \mu_{12}}{1-\mu_1}$: estimated FP rate

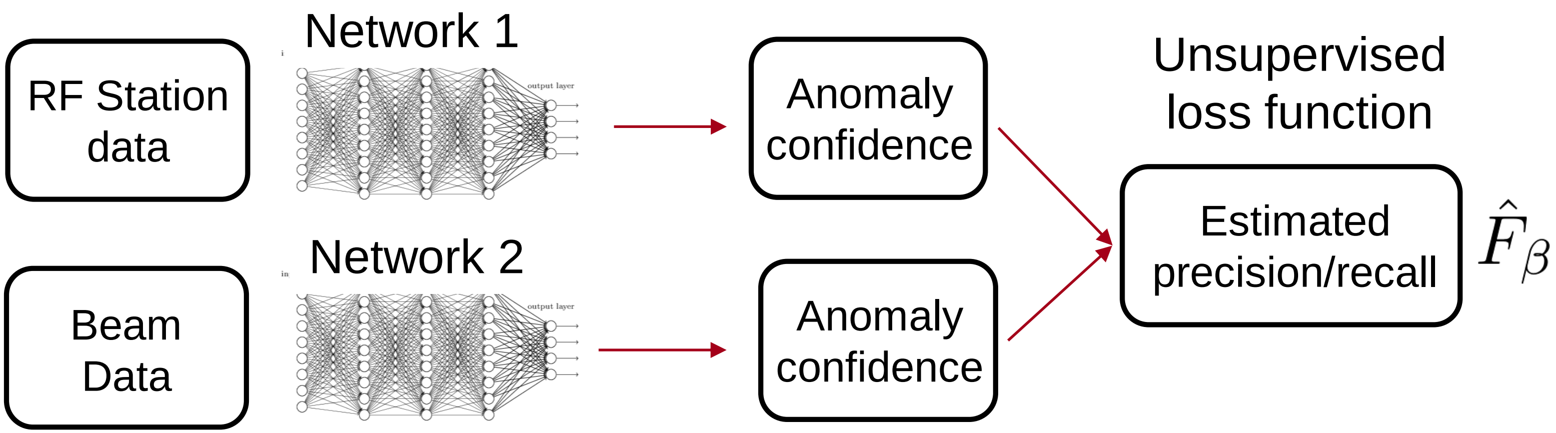
μ_i : mean of f_i

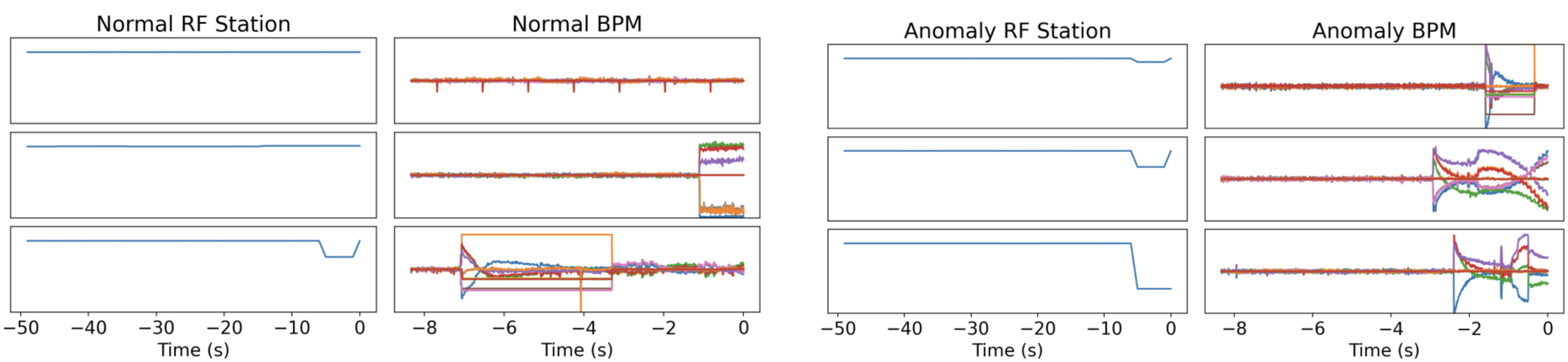
α: Estimated anomaly rate

β: Weighting factor (recall vs. precision)

CoAD for RF Station Faults

Implementation for RF station task:





Case Study 1: Performance in the control room

SLAC

Summary of predictions on 4 months of data

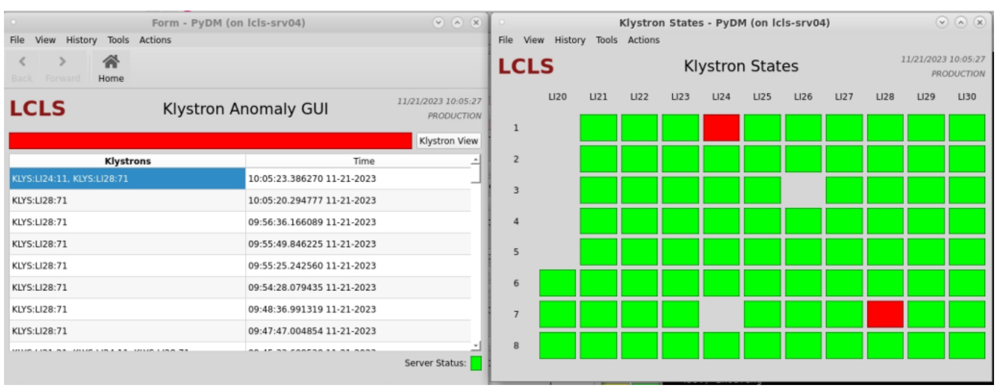
	Precision	Recall (events)
Manual recording by operator	NA	6
Status bit	0.31	385
Coincident Detection	0.88	504

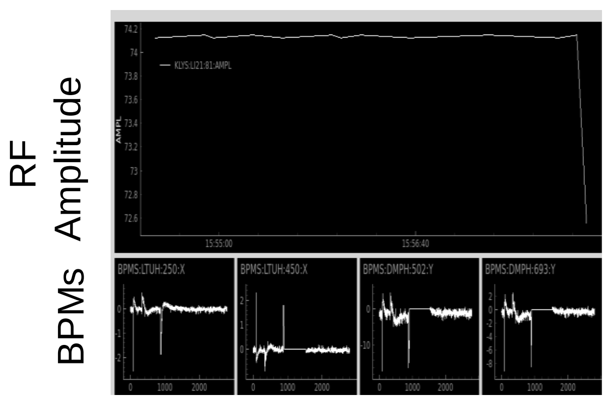
100x more anomalies vs. manual recording

30% more anomalies vs. status bit

6x fewer false positives vs. status bit

Live deployment in LCLS control room!



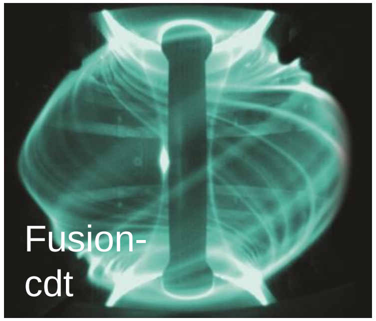


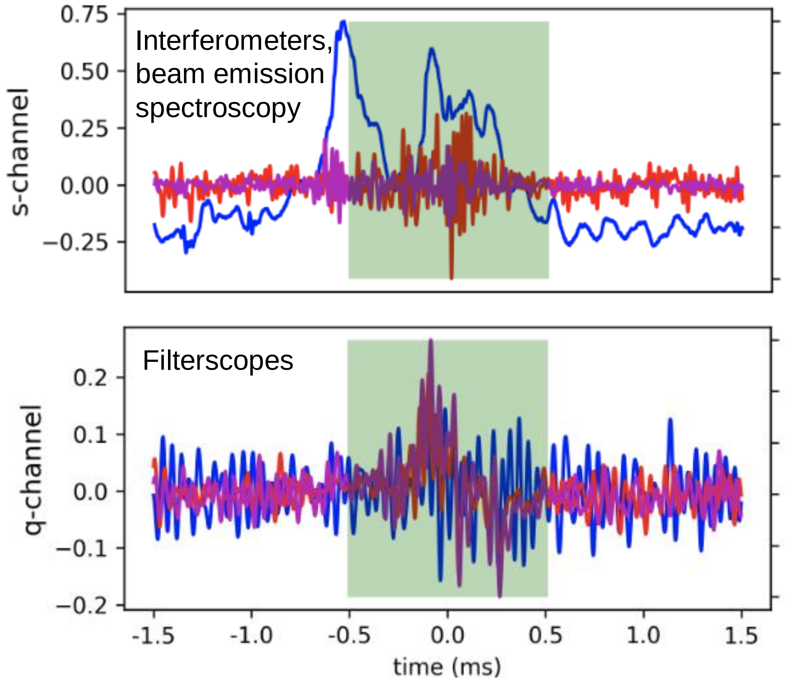
Ryan Humble, Finn O’Shea, Zhe Zhang

Anomaly detection: other applications

Searching for edge-localized modes in tokomaks

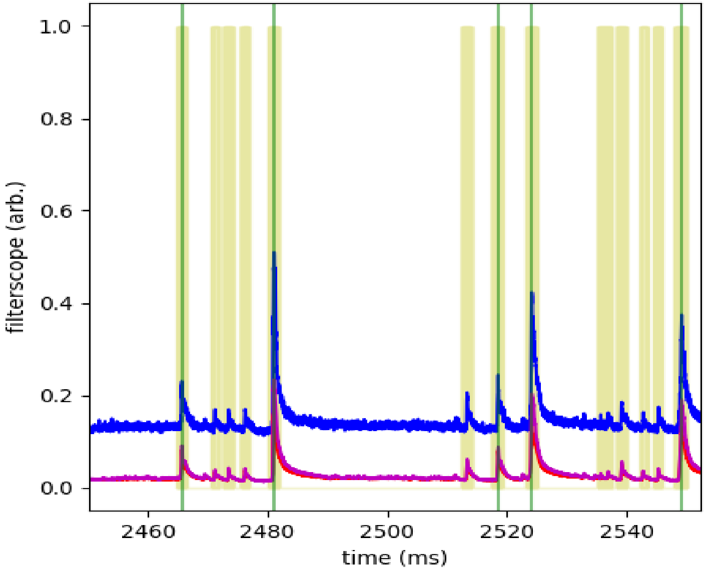
- Lots of data
- Mostly unlabeled
- Multi-modal diagnostics





Green lines: existing State-of-art

Yellow bands: DNN trained with coincidence



F. O'Shea

Outline

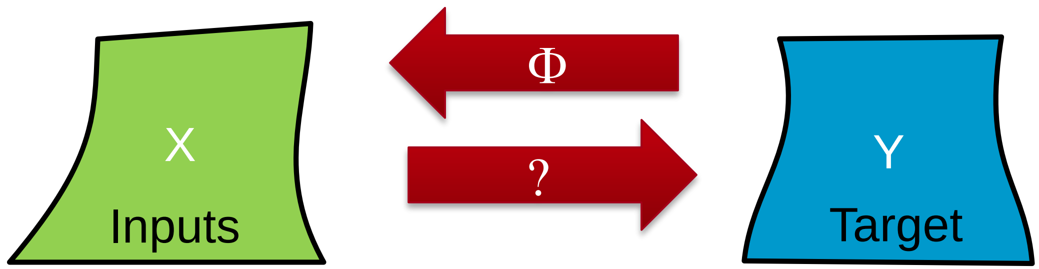
1. Online control of x-ray facilities

2. Anomaly detection

3. **ML for inverse problems**

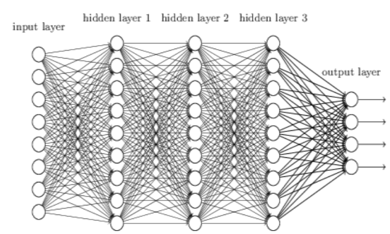
Inverse problems

Generic inverse problem:



1. Solve optimization problem: $$y^* = \underset{y}{\operatorname{argmin}} \left|\Phi(y) - x\right|^2$$

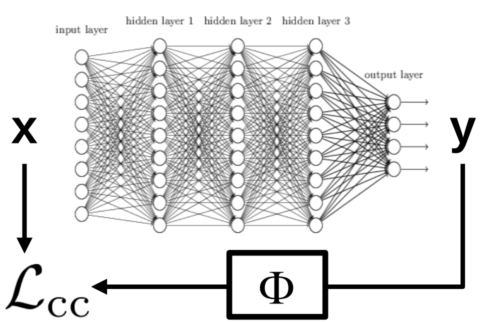
2. Train a neural network:

x  **y**

$$\mathcal{L} = \sum_i^n \left|y_{\mathrm{pred}} - y_{\mathrm{GT}}\right|^2$$

3. Train a cycle-consistent neural network:

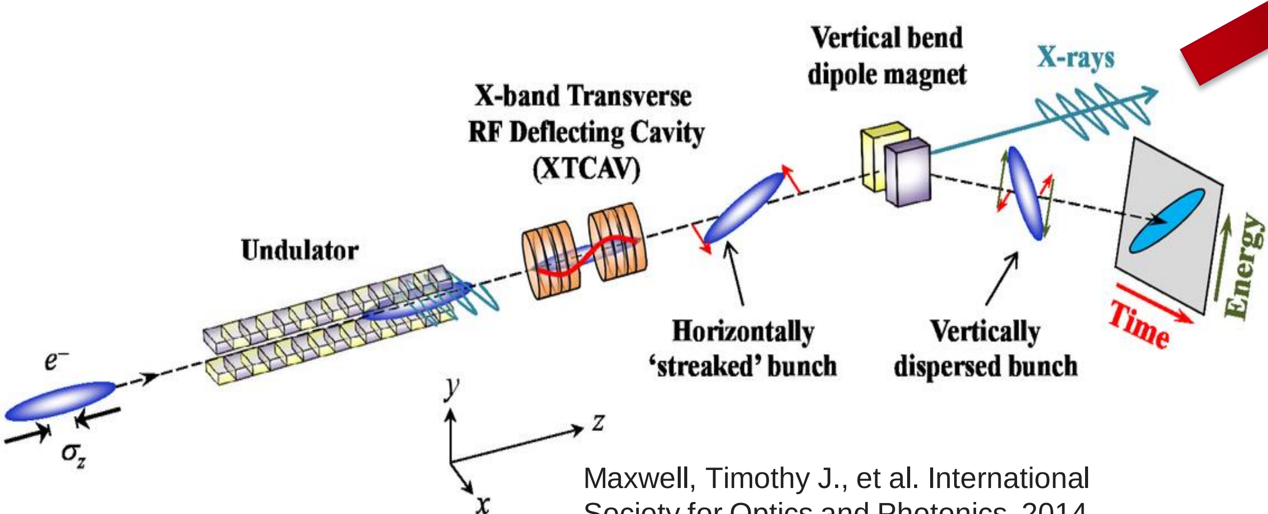
Caveat: Φ must be differentiable!



$$\mathcal{L}_{\mathrm{cc}} = \sum_i^n \left|\Phi(y_{\mathrm{pred}}) - x\right|^2$$

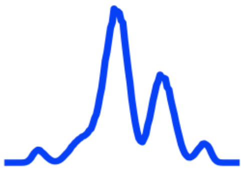
Inverse problems: X-ray Pulse Reconstruction

Measure amplitude of power/spectrum: can I recover phase?



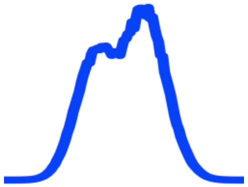
Maxwell, Timothy J., et al. International Society for Optics and Photonics, 2014.

Input spectral power points
(80meV FWHM resolution)



Spectrum

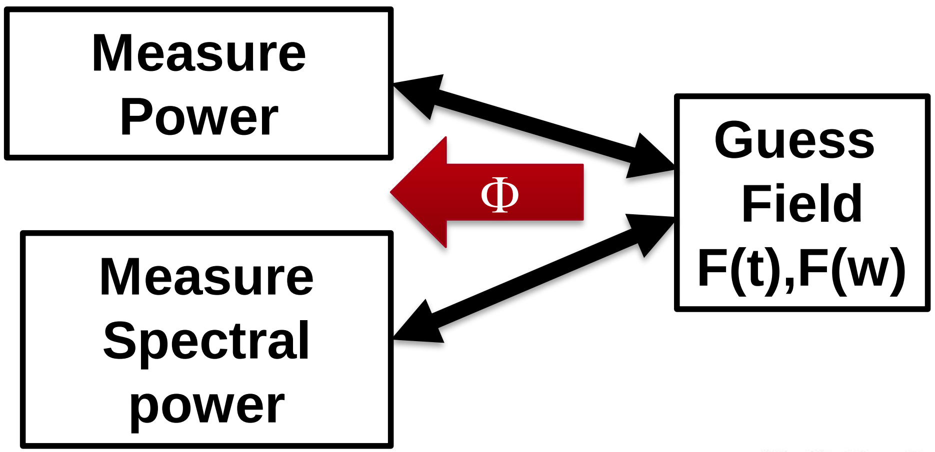
Input power points
(3fs FWHM resolution)

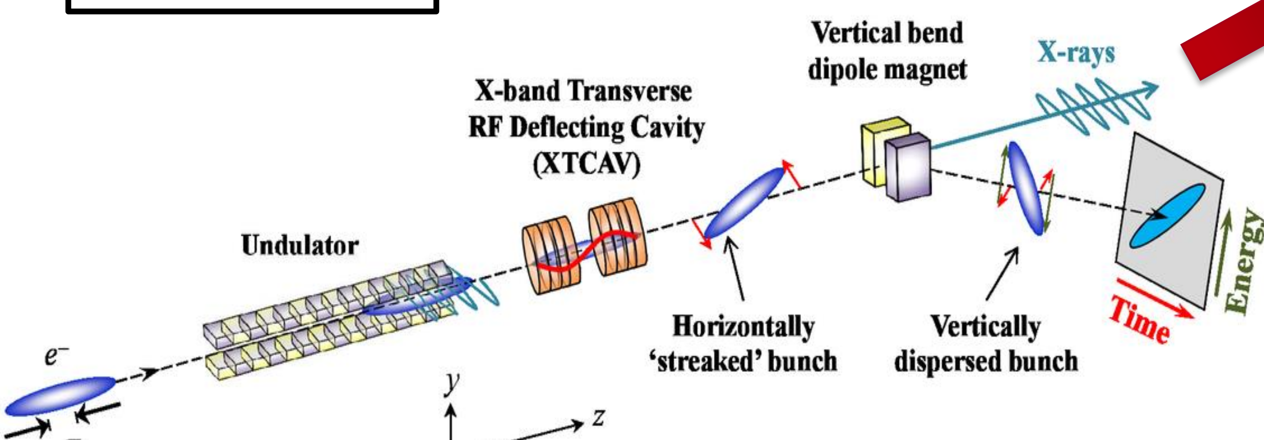


power

Inverse problems: X-ray Pulse Reconstruction

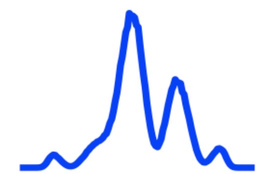
Approach #1: Iterative optimization





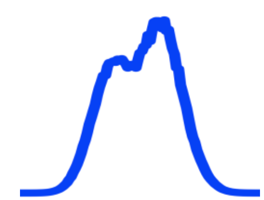
Maxwell, Timothy J., et al. International Society for Optics and Photonics, 2014.

Input spectral power points (80meV FWHM resolution)



Spectrum

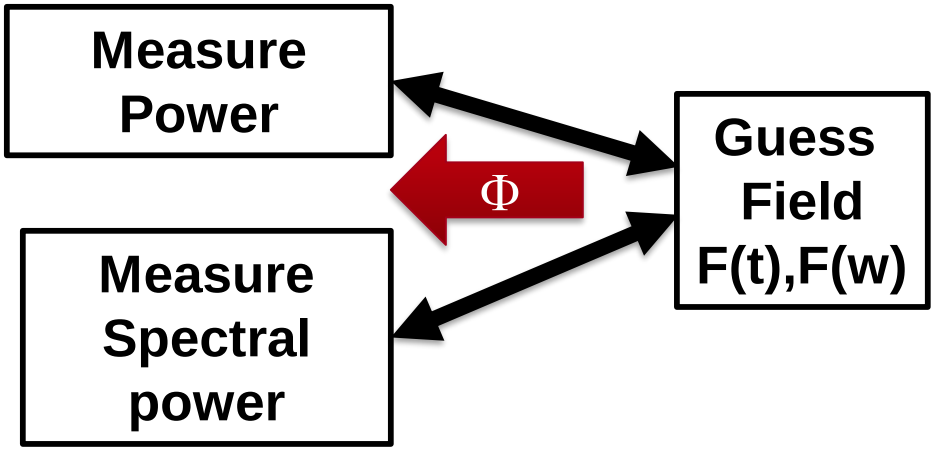
Input power points (3fs FWHM resolution)



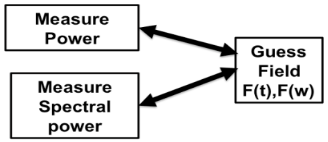
power

Inverse problems: X-ray Pulse Reconstruction

Approach #1: Iterative optimization



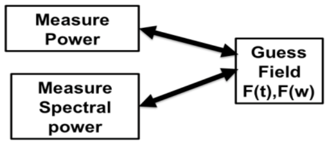
Pulse #1



⋮

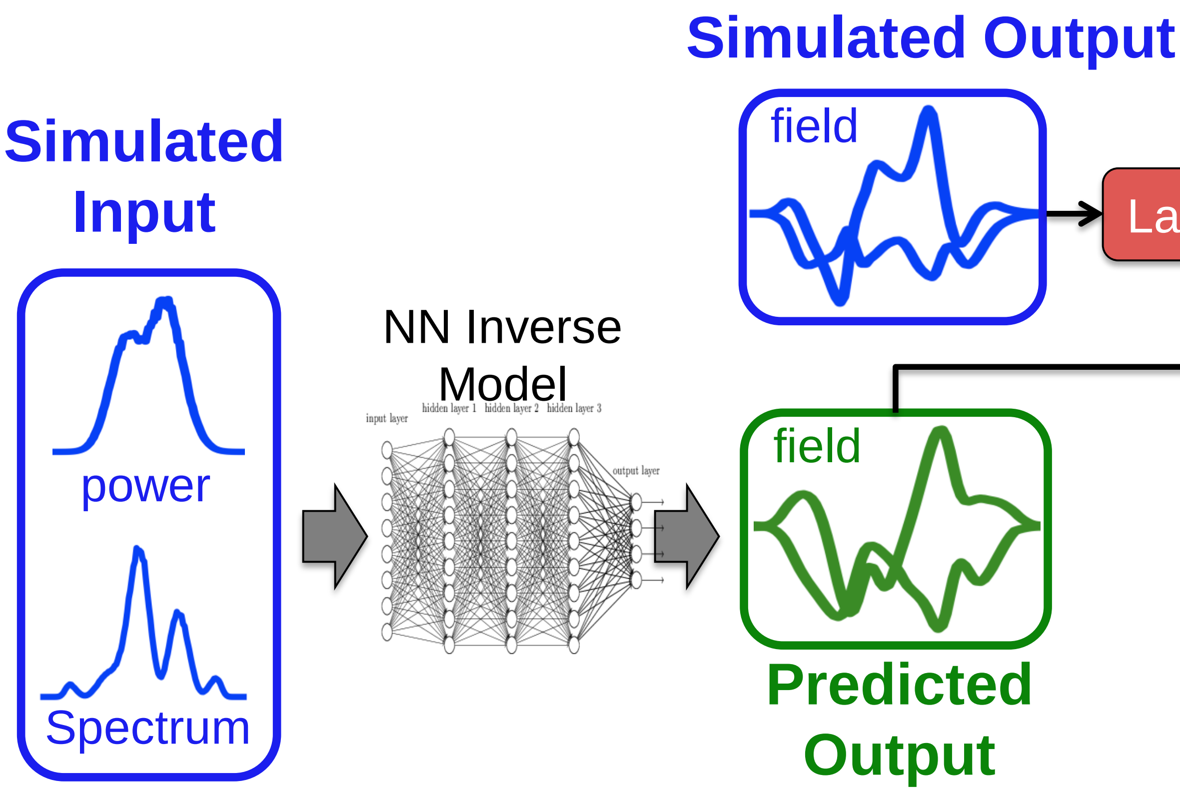
10^6-10^9 times

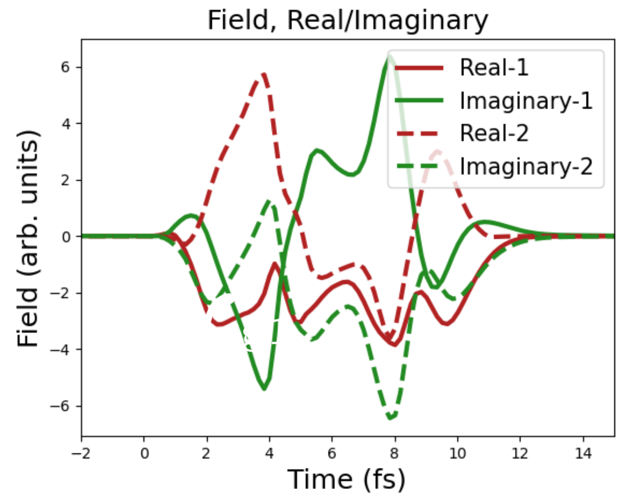
Pulse #n



Inverse problems: X-ray Pulse Reconstruction

Approach #2: Train a labeled neural network





Problem: absolute phase is meaningless!

D. Ratner et al., Opt. Exp., 2021

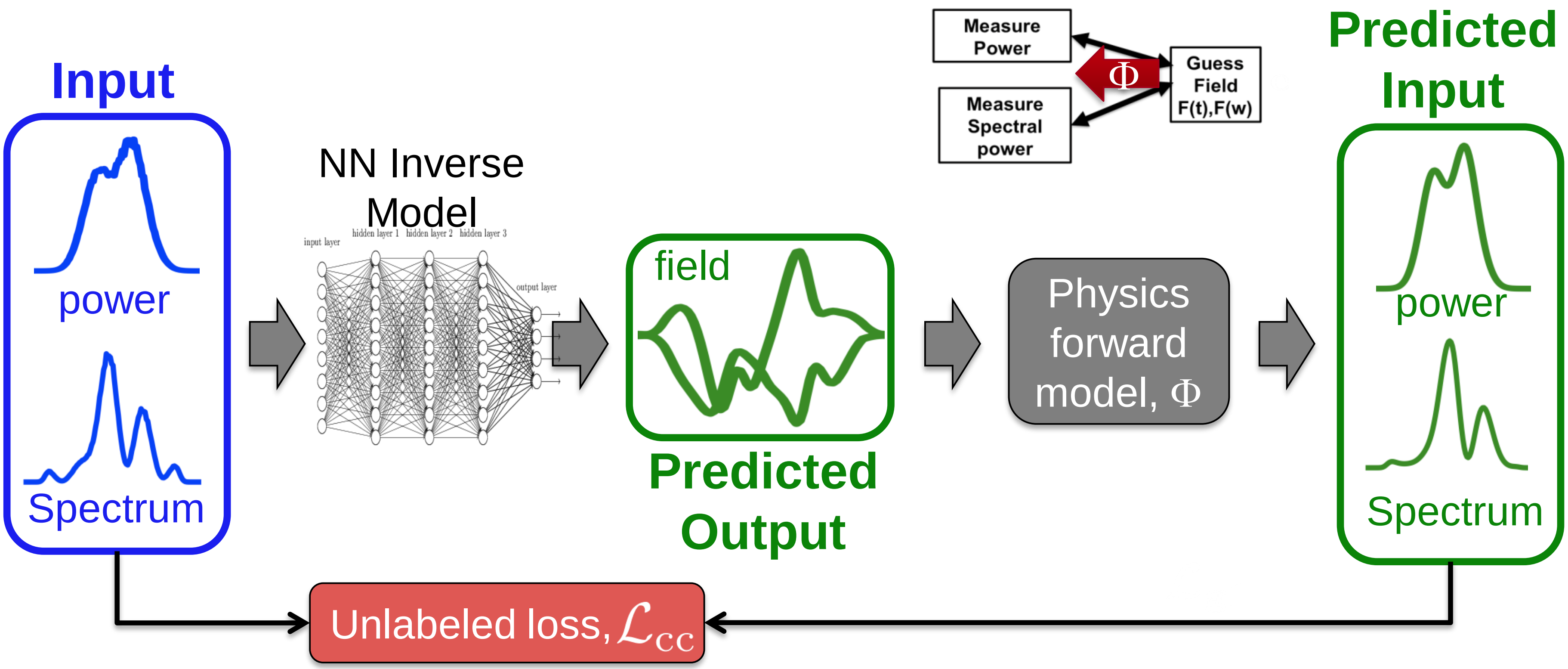
Inverse problems: X-ray Pulse Reconstruction

SLAC

Approach #3: Train a neural network with cycle-consistent loss

"Physics-informed neural network" (PINN)

Advantages: No need for labeled data, train directly on experimental data

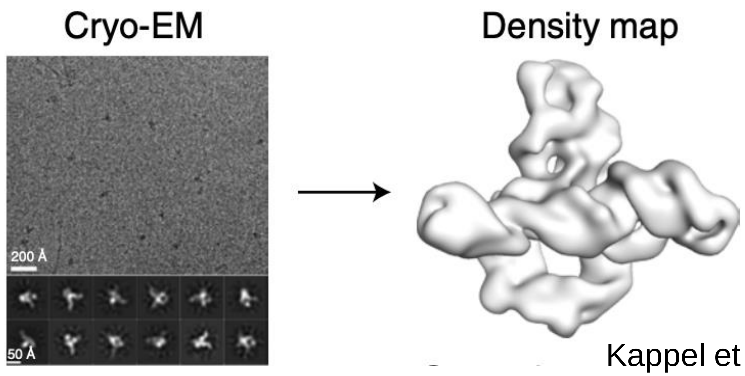


D. Ratner et al., Opt. Exp., 2021

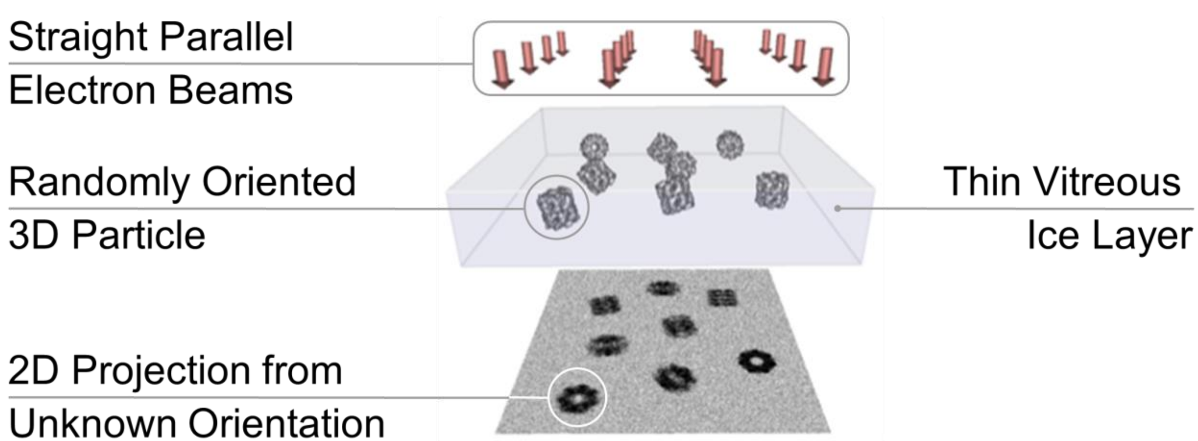
Inverse problems: Single-particle imaging

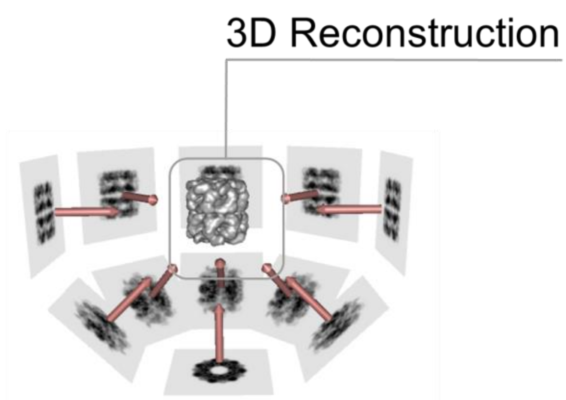
Single-particle imaging of biomolecules with Cryo-EM:

- Structures for un-crystallizable proteins/RNA
- Closer to in-vivo than a crystal
- Access to *conformations* not just average structures



Kappel et al.





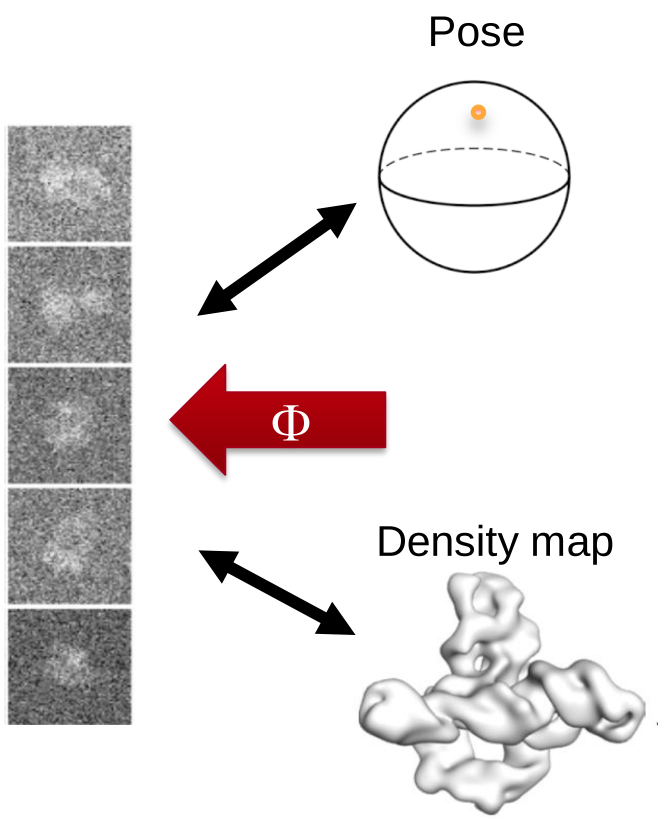
Images from people.csail.mit.edu Greg Pintilie

Inverse problems: Single-particle imaging

Recast as an inverse problem:

Iteratively solve for pose and densities
(e.g. expectation-maximization)

Why use deep learning?

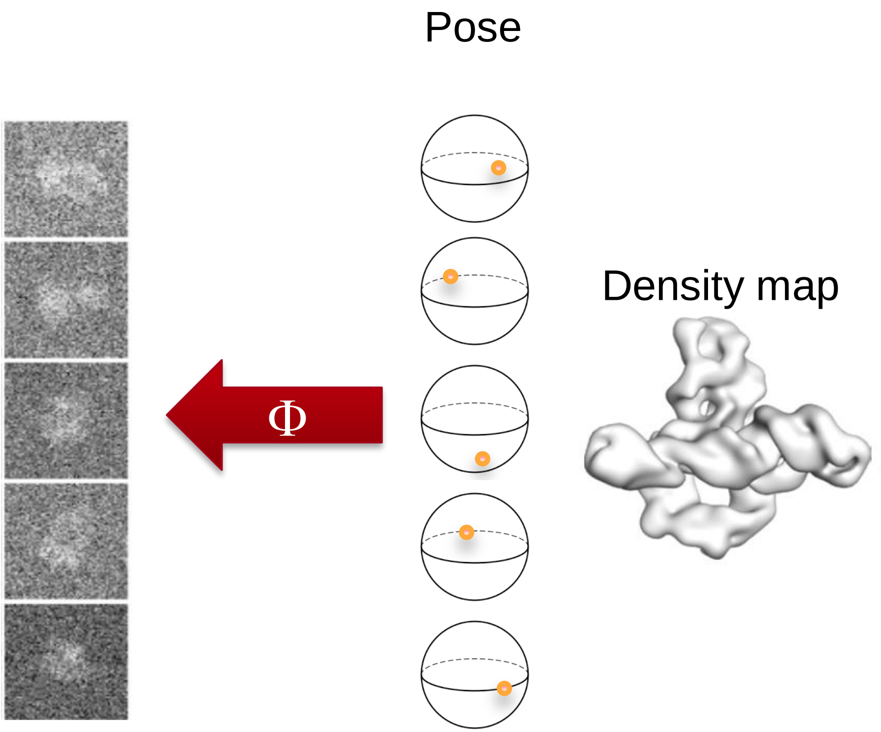


Inverse problems: Single-particle imaging

Recast as an inverse problem:

Iteratively solve for pose and densities
(e.g. expectation-maximization)

Why use deep learning?



1. Scaling with particle number: classical methods require pose for every particle

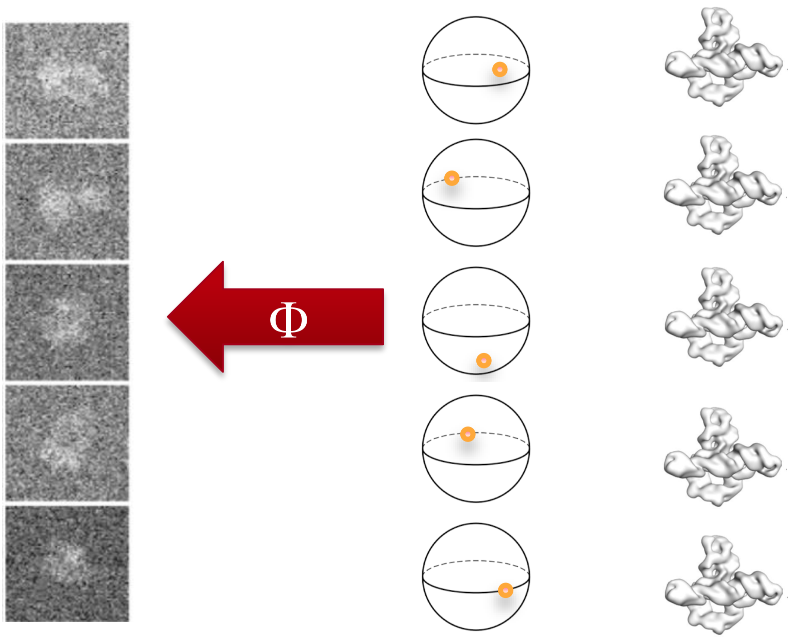
Inverse problems: Single-particle imaging

Recast as an inverse problem:

Iteratively solve for pose and densities (e.g. expectation-maximization)

Why use deep learning?

Pose Density map



1. Scaling with particle number: classical methods require pose for every particle

2. Solving for conformations: classical methods cannot handle complex, continuous variation

Inverse problems: Single-particle imaging

SLAC

Recast as an inverse problem:

Iteratively solve for pose and densities (e.g. expectation-maximization)

Why use deep learning?

ARTICLES
https://doi.org/10.1038/s41592-020-01049-4

nature methods

Check for updates

CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks

Ellen D. Zhong[1,2], Tristan Bepler[1,2], Bonnie Berger[2,3] and Joseph H. Davis[1,4]

IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING, VOL. 7, 2021 759

CryoGAN: A New Reconstruction Paradigm for Single-Particle Cryo-EM Via Deep Adversarial Learning

Harshit Gupta, Michael T. McCann, *Member, IEEE*, Laurène Donati, and Michael Unser, *Fellow, IEEE*

DeepMind 2021-06-25

Inferring a Continuous Distribution of Atom Coordinates from Cryo-EM Images using VAEs

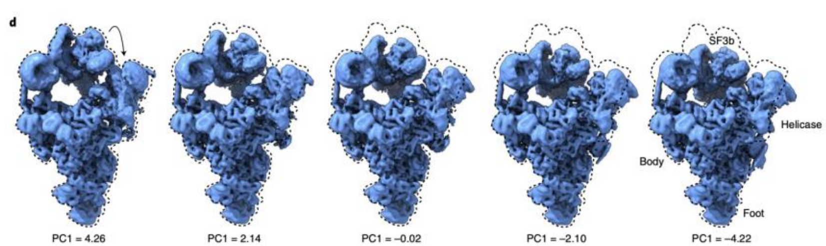
Dan Rosenbaum[*,1], Marta Garnelo[*,1], Michal Zielinski[*,1], Charlie Beattie[1], Ellen Clancy[1], Andrea Huber[1], Pushmeet Kohli[1], Andrew W. Senior[1], John Jumper[1], Carl Doersch[1], S. M. Ali Eslami[*,1], Olaf Ronneberger[*,1] and Jonas Adler[*,1]
[*]Equal contributions, [1]DeepMind

3D Flexible Refinement:
Structure and Motion of Flexible Proteins from Cryo-EM

Ali Punjani[1,2,3] David J. Fleet[1,2]

[1]University of Toronto [2]Vector Institute [3]Structura Biotechnology Inc.

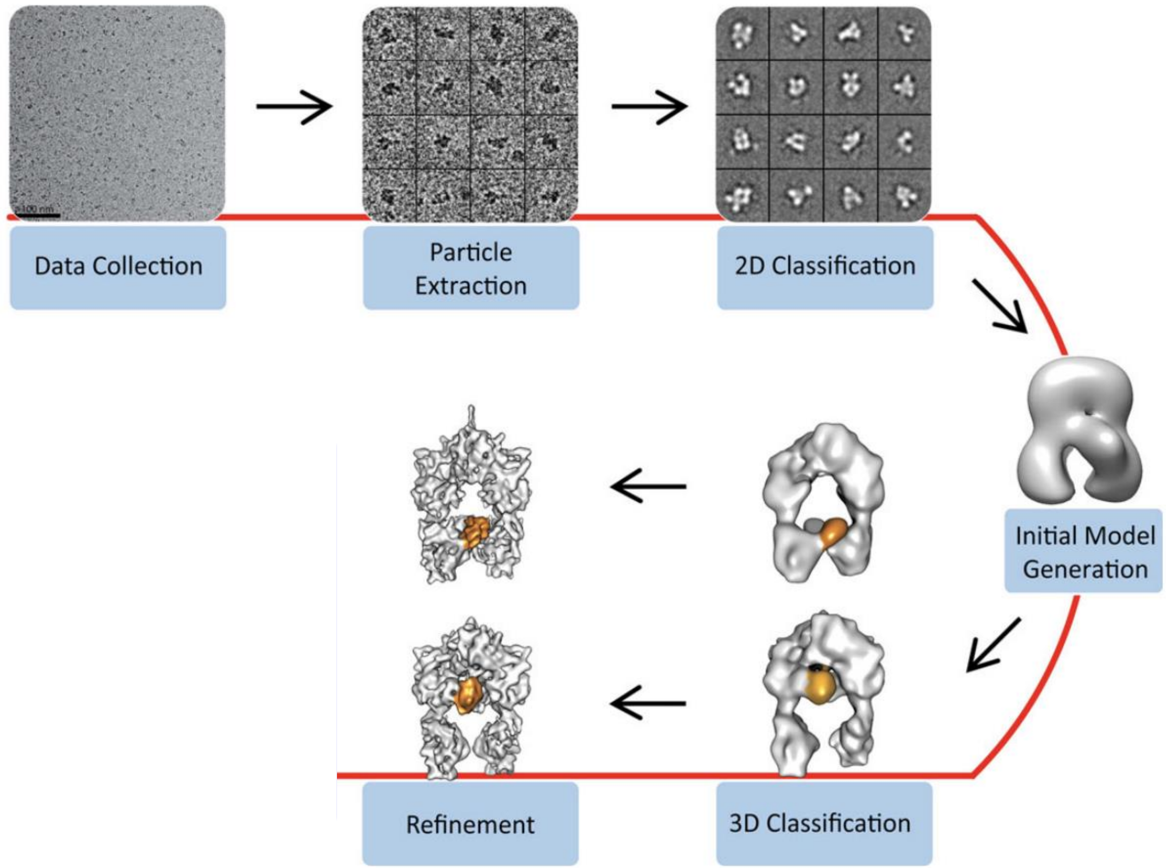
April 2021



CryoDRGN

Inverse problems: Single-particle imaging

Ultimate goal is to produce an atomic model…



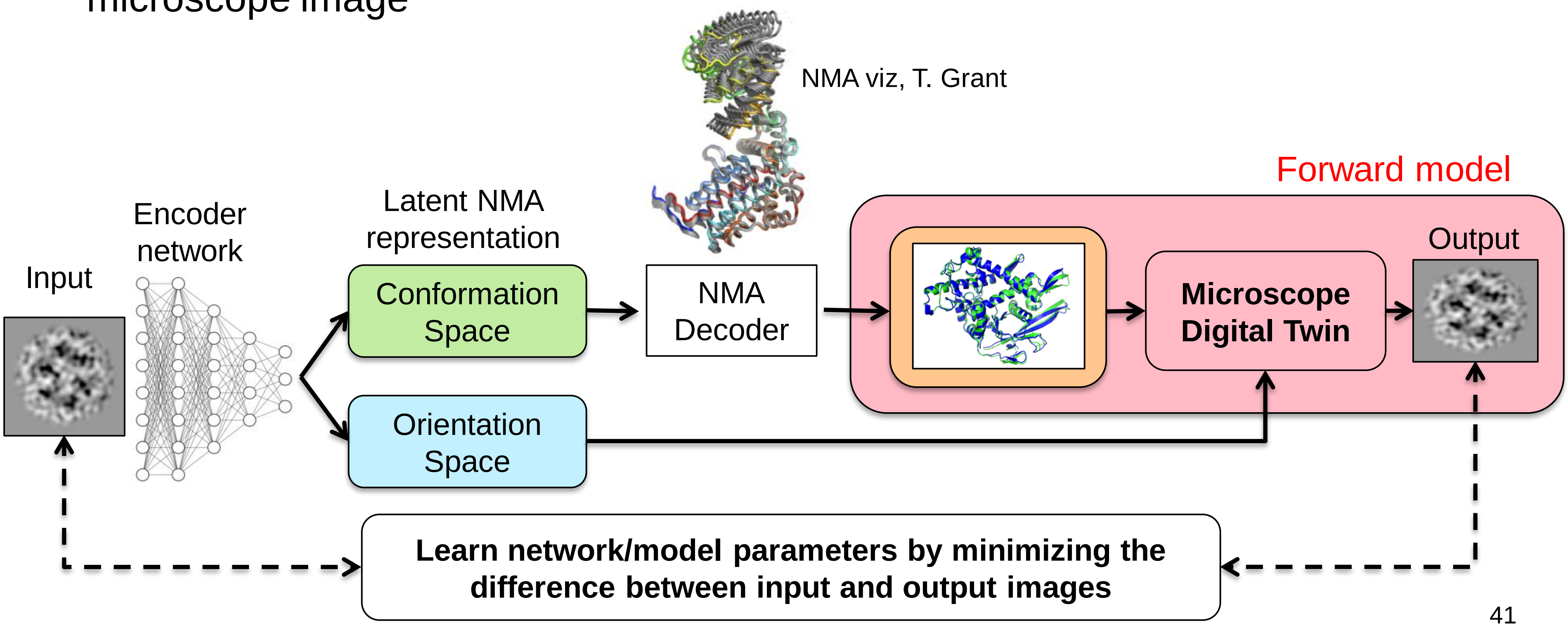
…but fitting an atomic model is still a manual process

G. Skinkiotis, D. Southworth, Microscopy 2016

Inverse problems: Single-particle imaging

Recent work: directly learn atomic model conformations on data

- Parametrize atomic model with normal mode analysis (NMA)
- Train encoder network to map each image to a point in NMA space
- Differentiable simulation maps NMA space to atomic model to microscope image

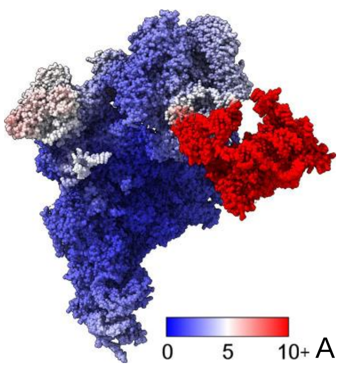


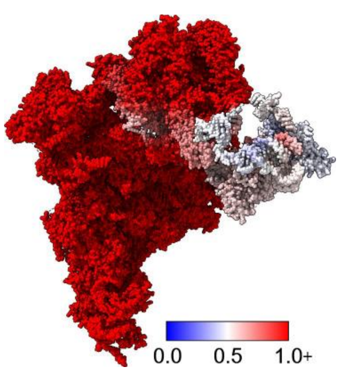
Inverse problems: Single-particle imaging

First experimental results on spliceosome:

- Fitting 16 normal modes, 140k particles, 128x128 images
- Uncertainties from cross-validation (2 partitions)

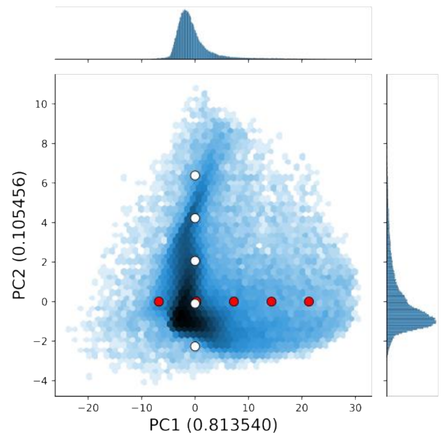
Absolute error

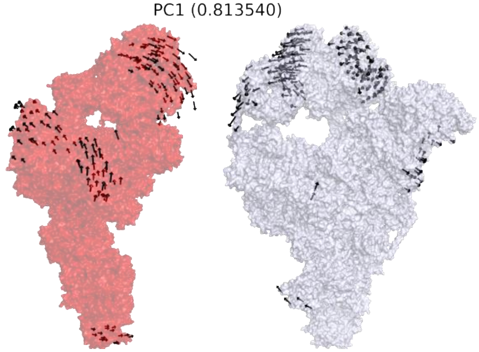




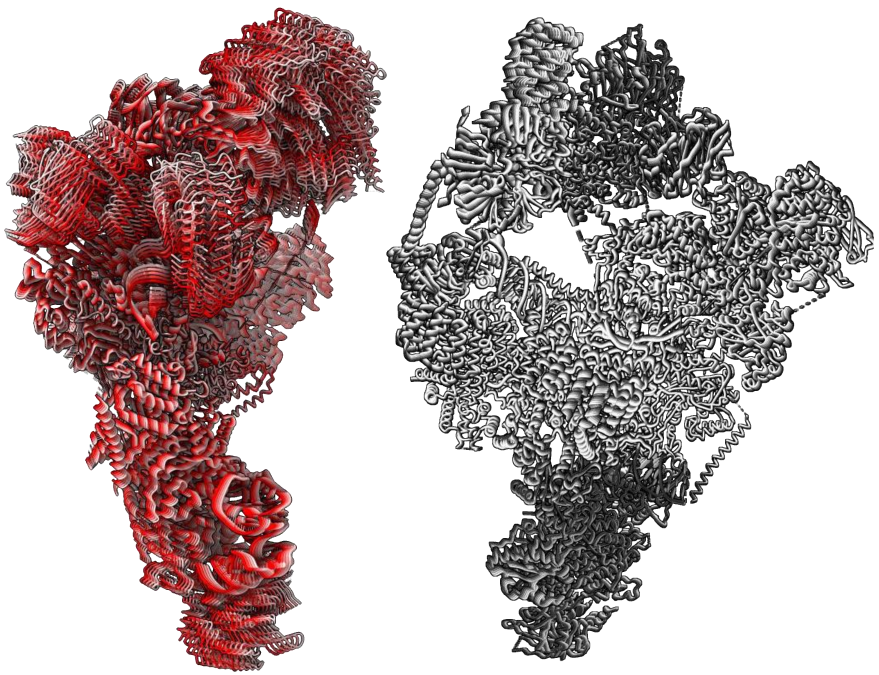
Signal-to-noise

NMA latent space





Observed motion



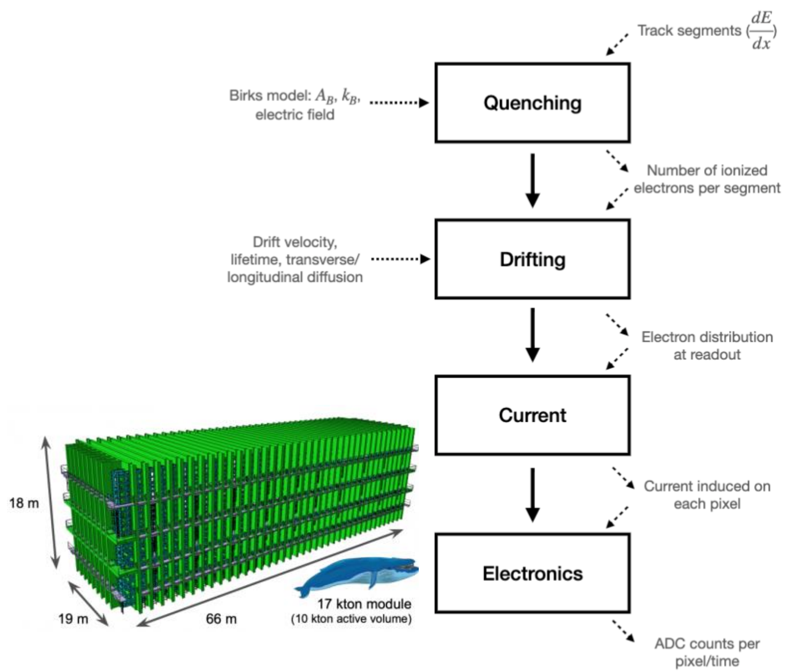
Y. Nashed, F. Poitevin, et al.

Inverse problems: differentiable simulations

Key requirement for PINNs: *differentiable* simulator

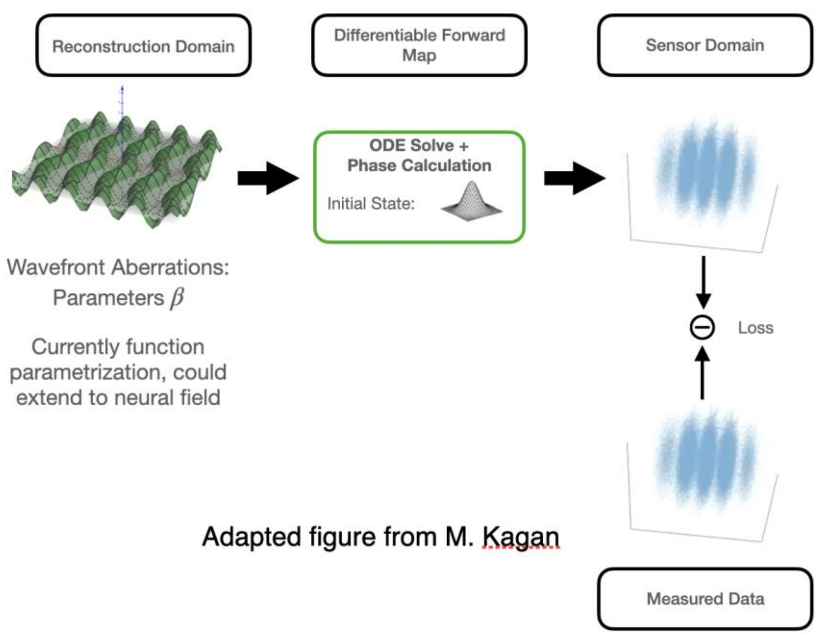
Example of benefit from cross-domain collaboration: same team working on LArTPC, MAGIS, and CryoEM!

LArTPC (e.g. DUNE) simulator:



S. Gasiorowski, Y. Nashed, Y. Chen, K. Terao et al.

MAGIS-100 simulator



Adapted figure from M. Kagan

Cheong, Sanha, et al (2022)

Acknowledgements / Stanford-SLAC Collaboration

Thanks to everyone who worked on these projects!

	Stanford CS	SLAC MLI	SLAC Domains
BAX	Stefano Ermon Willie Neiswanger	Sara Miskovich Sathya Chitturi	Auralee Edelen Chris Tassone
Anomaly detection	Eric Darve Ryan Humble	Finn O'Shea Zhe Zhang	William Colocho Matt Gibbs Ryan Coffee Maria Elena Monzari
CryoEM	Gordon Wetzstein	Youssef Nashed Sean Gasiorowski Leah Reeder	Fred Poitevin Ariana Peck Michael Kagan Axel Levy Kazu Terao Yifan Chen

Thanks for your attention!

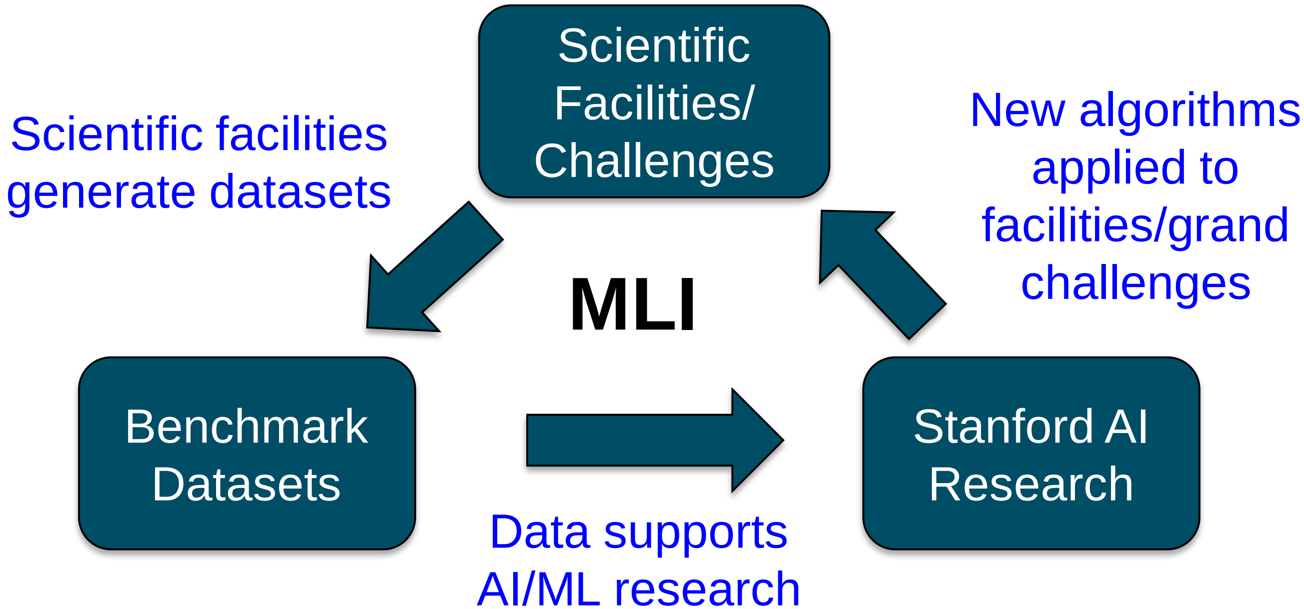
Backup

SLAC’s mission revolves around major scientific facilities

Machine Learning Initiative (MLI) structured around 3 connected pillars:

- Scientific facilities (SLAC)
- Algorithmic development (Stanford)
- Benchmark datasets (SLAC)

→ Tackle grand challenges in Science and AI/ML

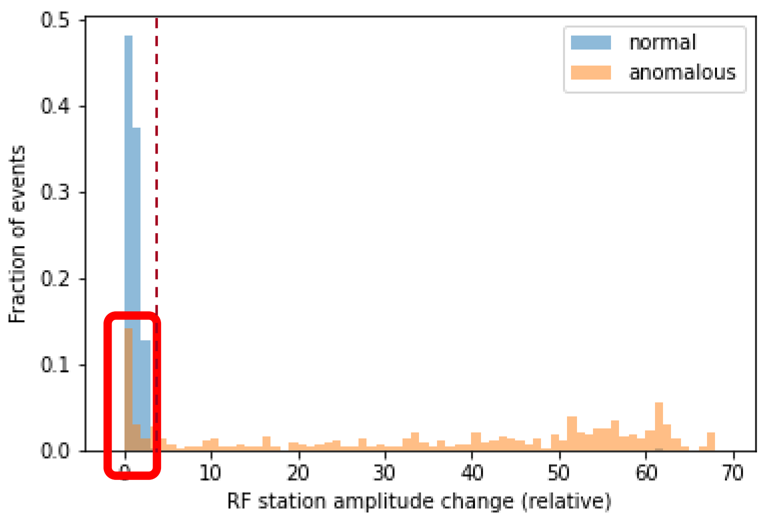


Example: SLAC Particle Accelerator Data Set

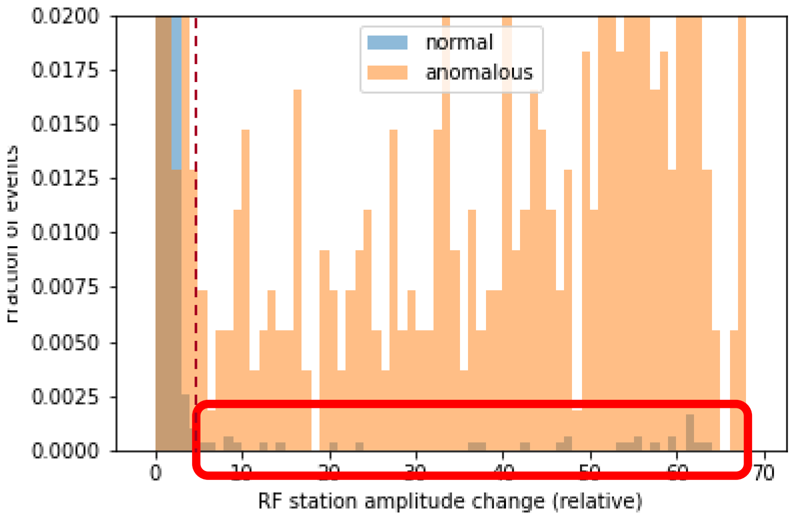
Does simple thresholding on RF station amplitude work?

No! Thresholding on the RF amplitude misses anomalies and triggers false alarms

Nearly 20% false negatives and false positives from a simple threshold



False negatives



False positives