Introduction to Machine Learning 2

Nov., 2018

D. Ratner,

SLAC National Accelerator Laboratory

Unsupervised learning

What can be accomplished without labels?

Supervised learning: **X, y**

Unsupervised learning: **X**

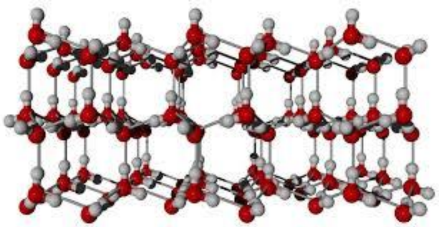
Cat

Dog

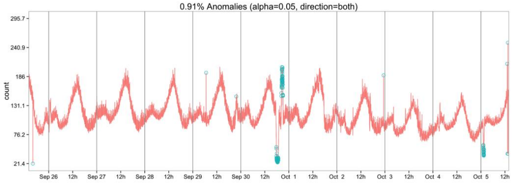
What can we hope to accomplish?

1. Clustering (classification)

2. Decomposition (e.g. “cocktail party problem”, species identification)



3. Anomaly/breakout detection (e.g. fault detection/prediction)



4. Generation (e.g. creating new examples within a class)

Unsupervised learning

What can be accomplished without labels?

Clustering: Divide **x** into **k** categories

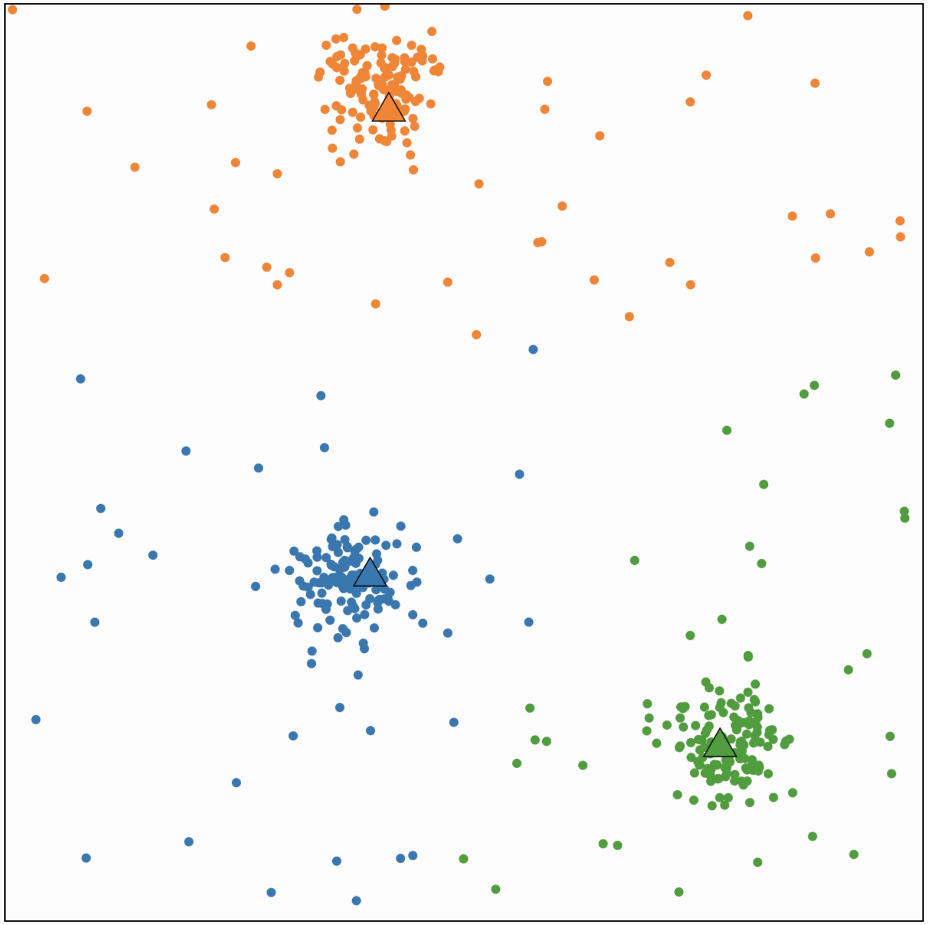
K-means algorithm:

a. Pick 'k' random centroids
b. Loop until convergence {
    1. Assign examples to nearest centroid
    2. Update centroids to mean of clusters

    }

See also: **Hierarchical clustering, DBSCAN, etc…**

K-means

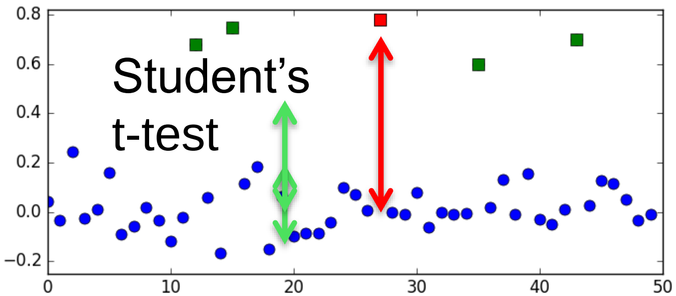


http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

Unsupervised learning

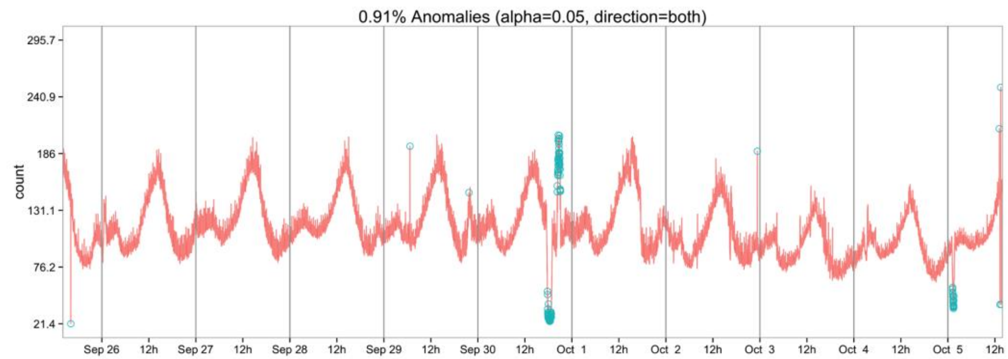
Time series data: Anomaly/Breakout/Changepoint Detection

Anomaly detection:
identify points that are statistical outliers from a distribution



PyAstronomy: Generalized ESD (GESD)
(Available from pip install)

twitter / AnomalyDetection

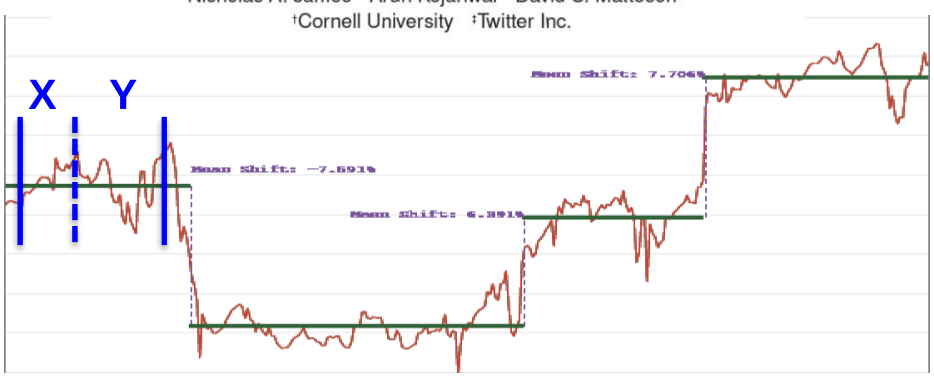


Breakout/Changepoint detection:
Find point in time at which distribution changed

$$\mathcal{E}(X,Y) = 2E|X-Y| - E|X-X'| - E|Y-Y'|$$

Leveraging Cloud Data to Mitigate User Experience from 'Breaking Bad'

Nicholas A. James† Arun Kejariwal‡ David S. Matteson†
†Cornell University ‡Twitter Inc.



Unsupervised learning

Generating new data

Unsupervised learning with neural networks:
train a model to generate new examples based on training set

Deep dreaming of dogs

If you train a network to recognize dogs…

…it will hallucinate dogs

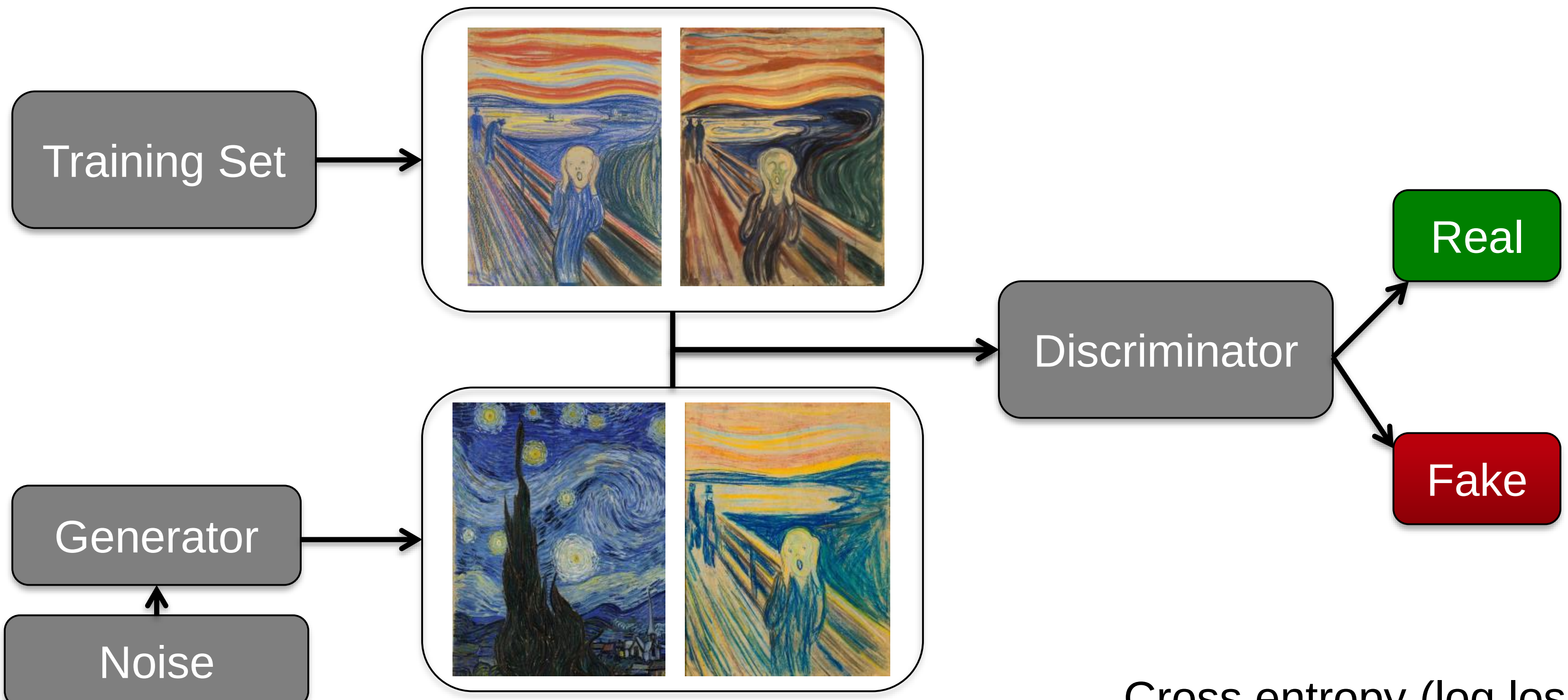
Style transfer

Gatys, et al.

Unsupervised learning

Generating new data

Generative Adversarial Network (GAN)

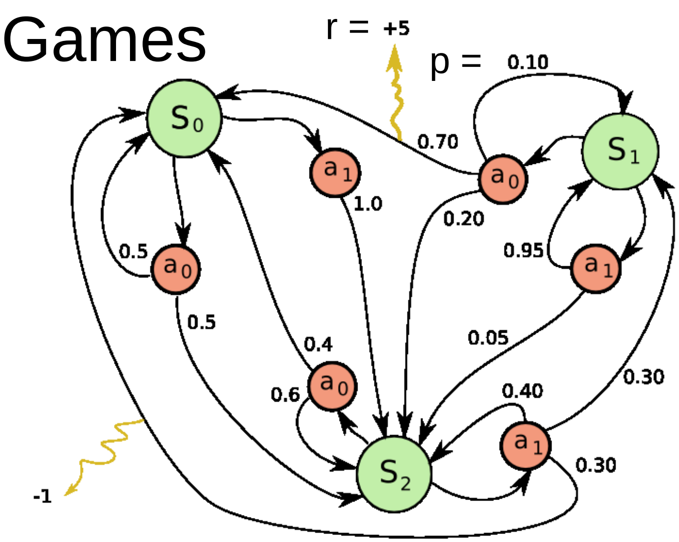


Cross entropy (log loss)

$$J^{(G)} = -J^{(D)}$$

$$J^{(D)} = -\sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

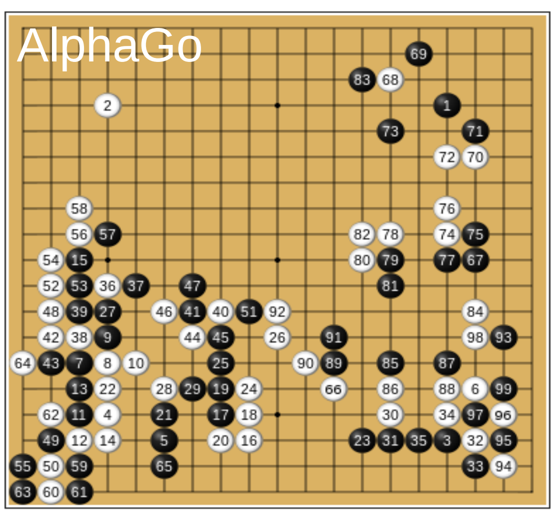
Machine Learning for Games

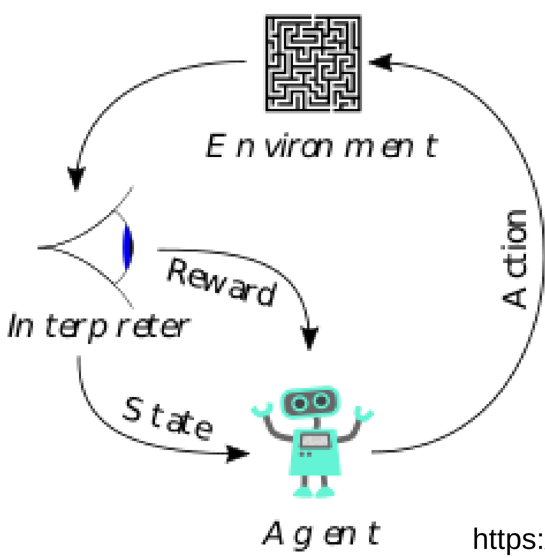


Third category: partial supervision

e.g. when playing a game, will not have a known label for every position, but will know who wins at the end

Goal is to solve for a"policy":
i.e. optimal action a_s, given state s





States: s
Actions: a
Transition probability: p
Rewards: r

https://en.wikipedia.org/wiki/Reinforcement_learning

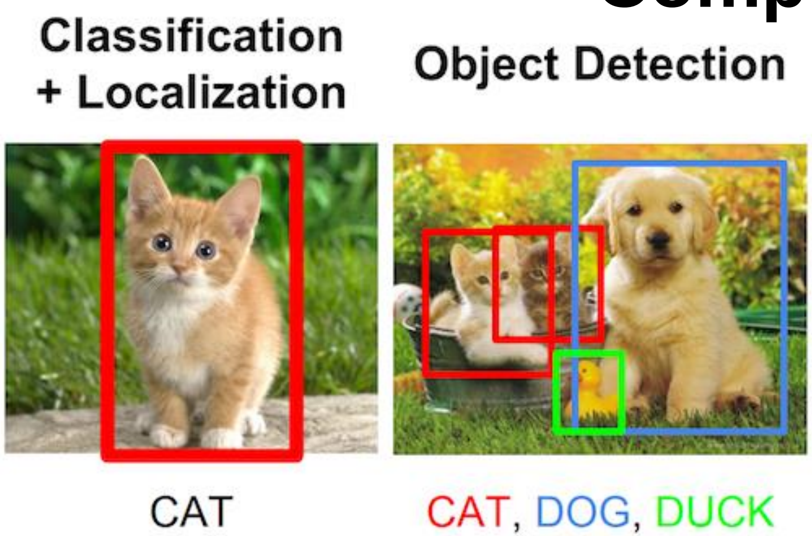
But is it useful??

Look at examples for Free Electron Lasers:

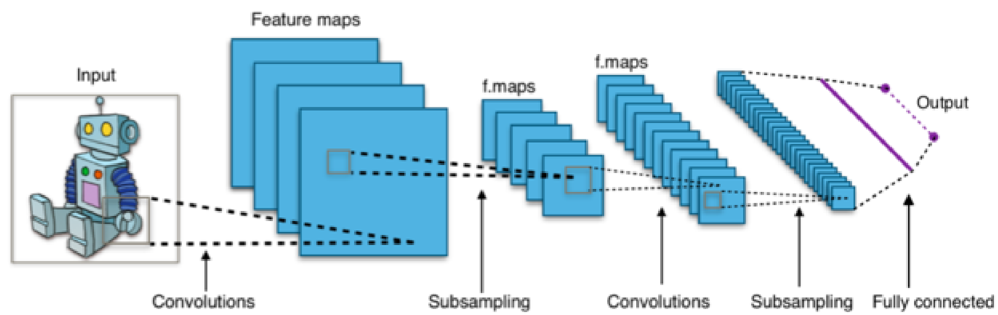
1. Computer vision to process screens
2. Neural networks to solve inverse problems
3. GANs to augment data sets
4. Breakout detection for fault recovery
5. Bayesian optimization and RL for online tuning
6. Regularization and convex optimization to analyze data

Supervised Learning:
Data Analysis – Pulse reconstruction

Computer vision

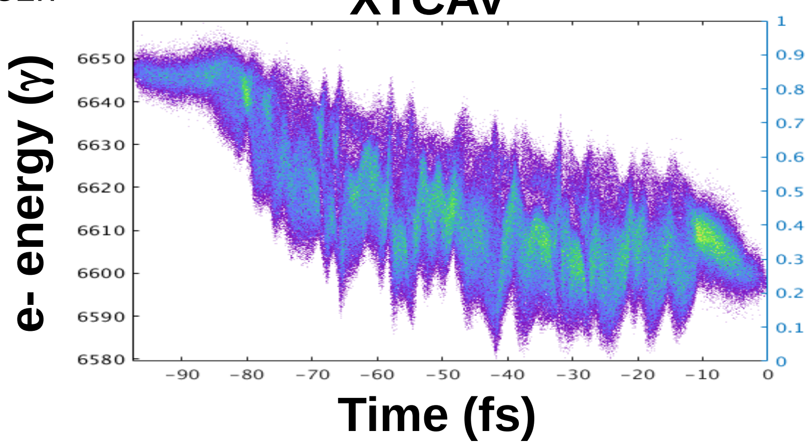


Stanford CS231n



Aphex34 https://commons.wikimedia.org/w/index.php?curid=45679374

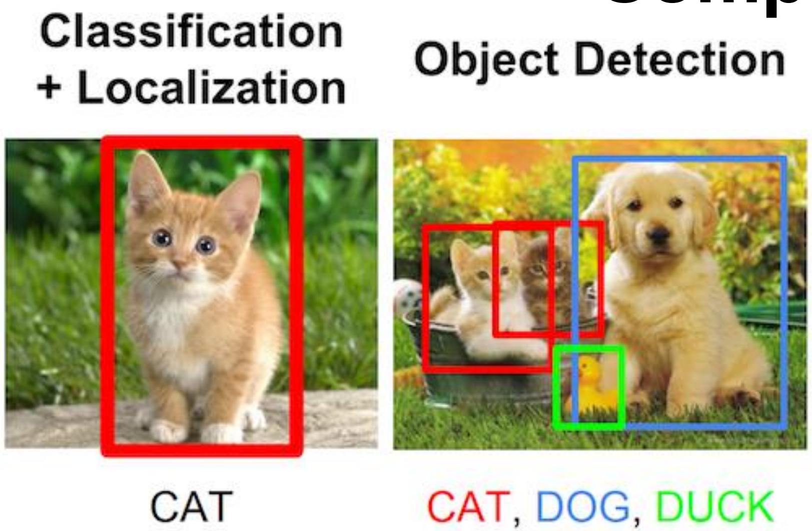
XTCAV



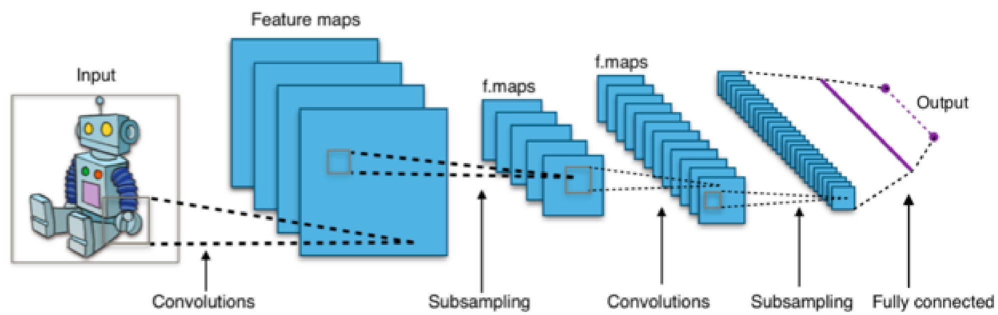
Longitudinal phase space of an electron beam for an XFEL

Supervised Learning:
Data Analysis – Pulse reconstruction

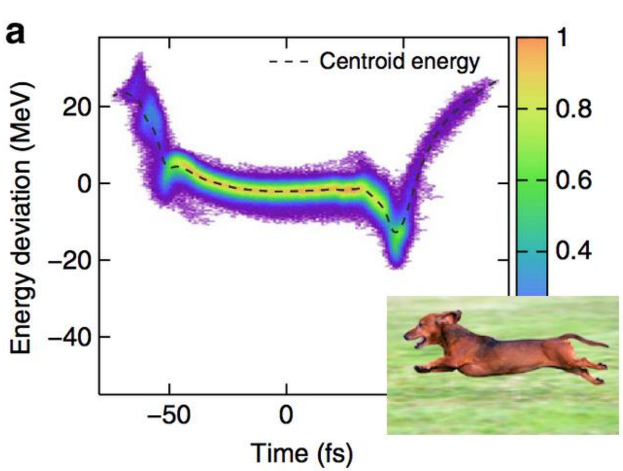
Computer vision

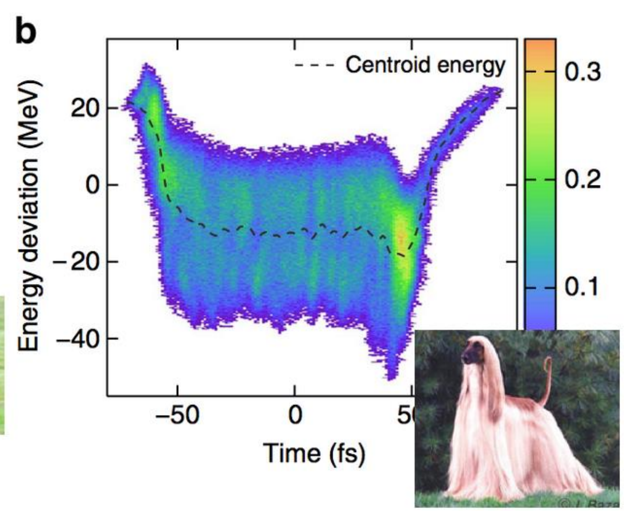


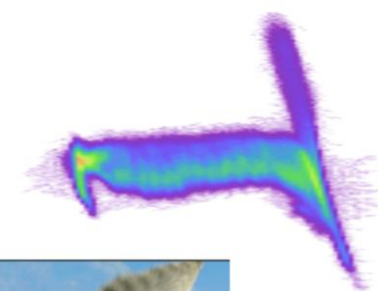
Stanford CS231n



Aphex34 https://commons.wikimedia.org/w/index.php?curid=45679374



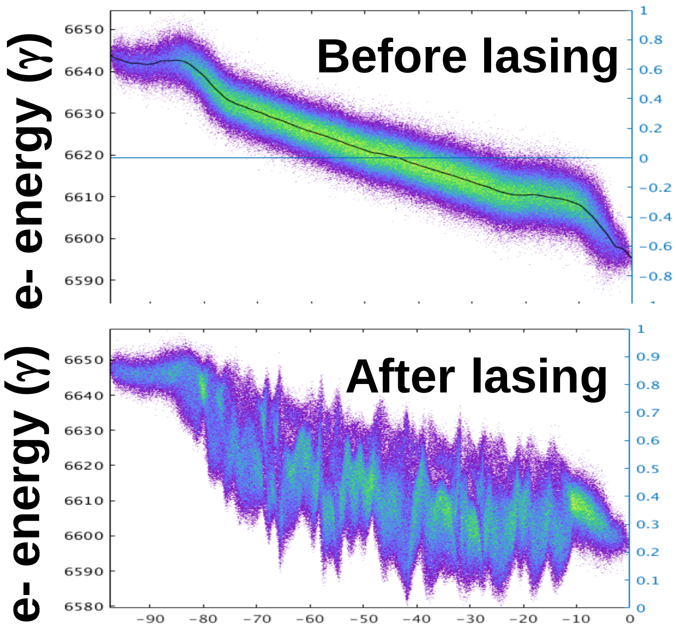


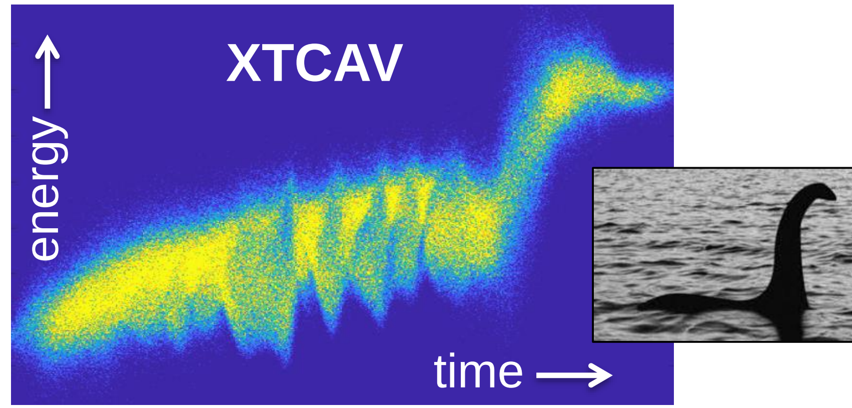


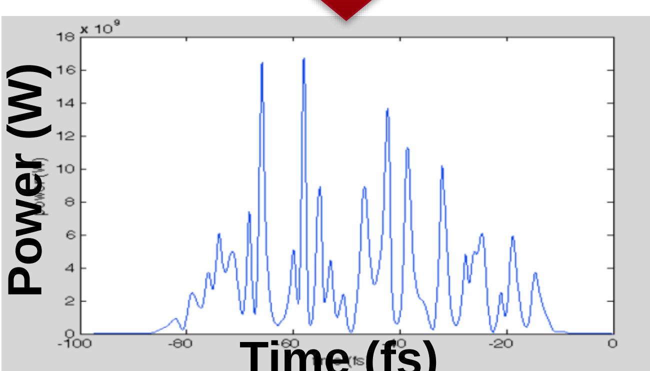
D. Mendez

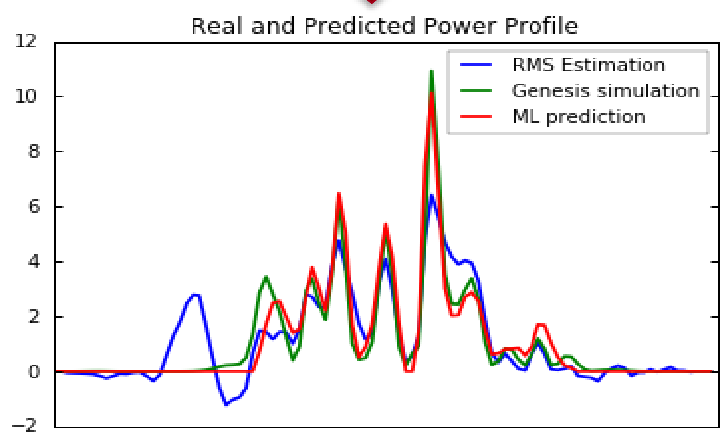
Supervised Learning:
Data Analysis – Pulse reconstruction

XTCAV Analysis









CNN prediction

Xinyu Ren

Full beam reconstruction

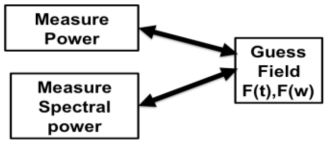
Measure amplitude of power/spectrum: can I recover phase?

Measure Power

Measure Spectral power

Guess Field F(t),F(w)

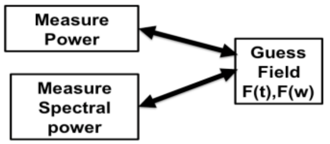
Pulse #1



⋮

10^6-10^9 times

Pulse #n

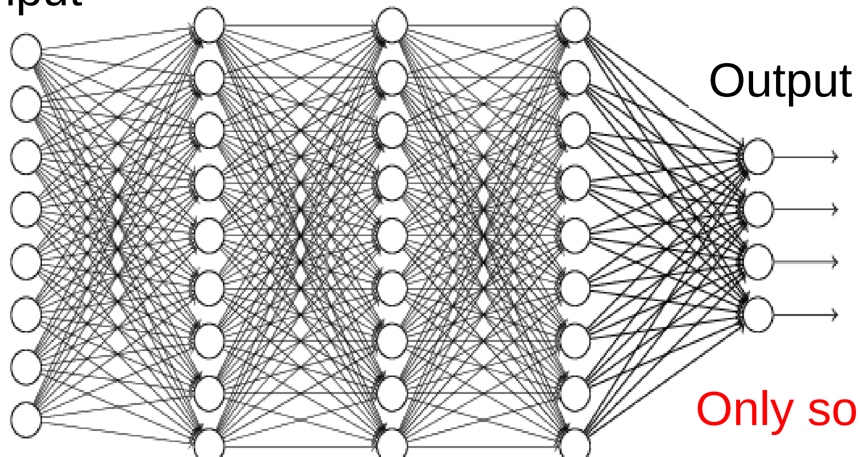


Measure Power

Measure Spectral power

Input

Hidden layers



Output

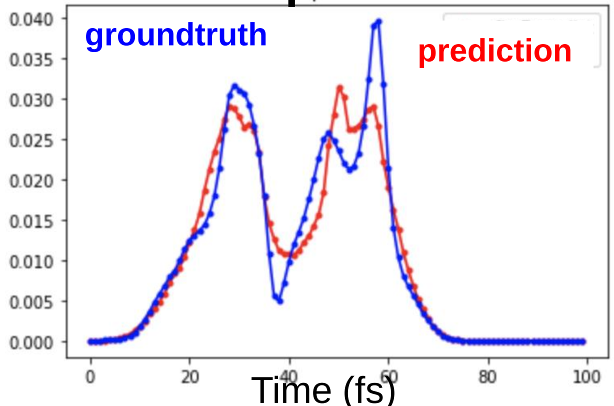
Guess Field F(t),F(w)

Only solve **once!**

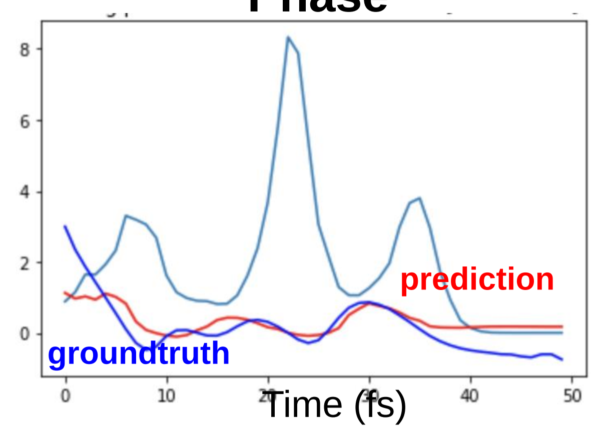
Full beam reconstruction

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

Amplitude

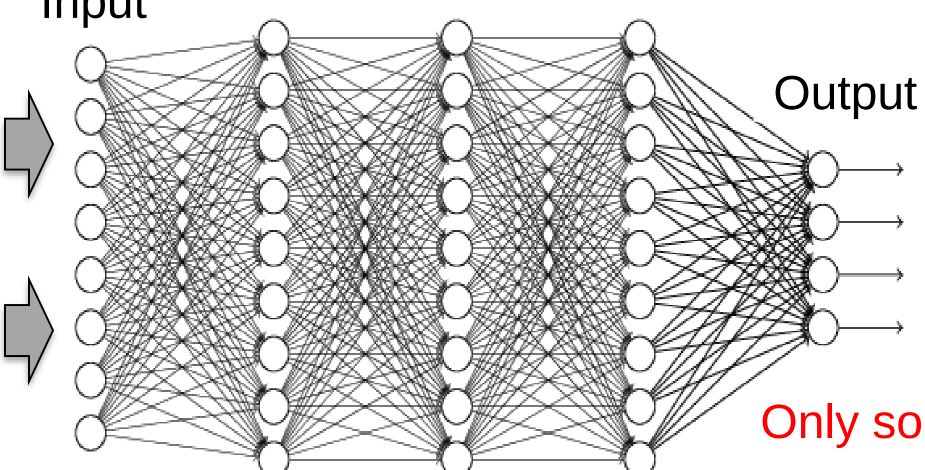


Phase



Measure Power

Measure Spectral power



Guess Field F(t),F(w)

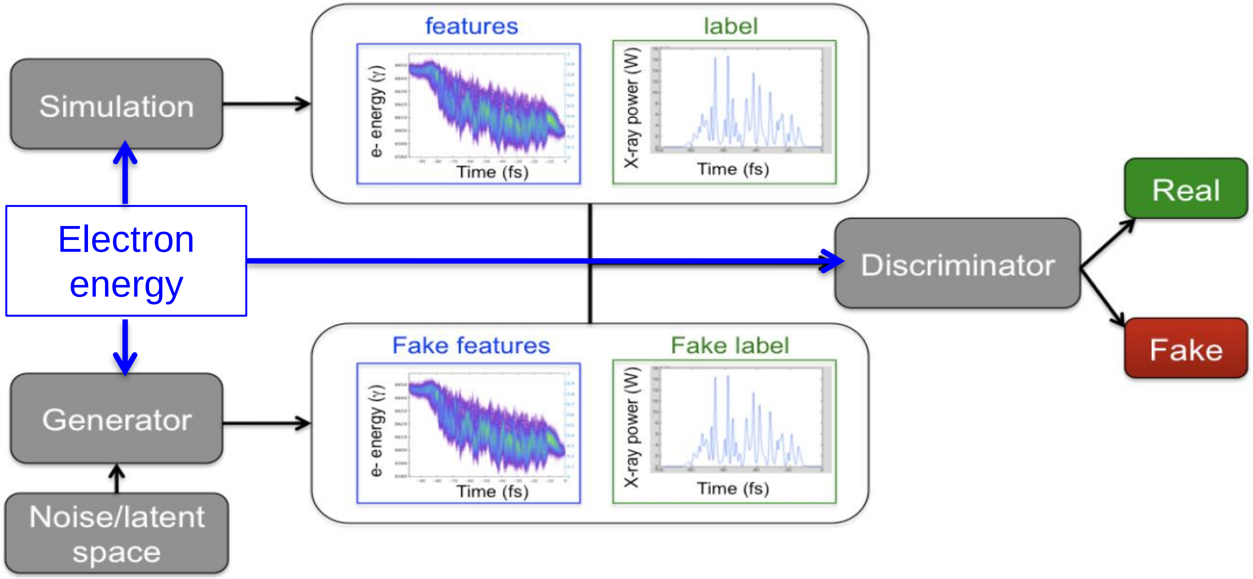
Only solve **once!**

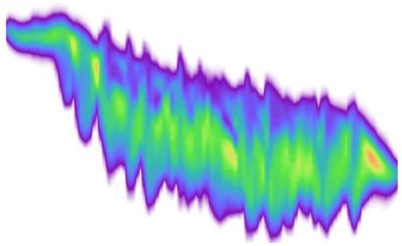
Xiao Zhang

Unsupervised learning
Surrogate Models – FEL simulations

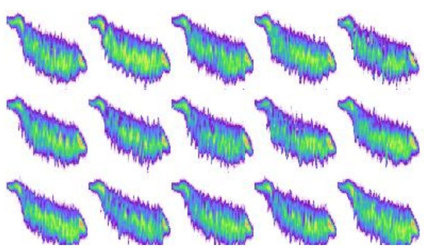
Generative adversarial network (GAN)

LCLS users need 100k to 1M pulses to prepare for beamtime
➔ 1 billion cpu-hours!





Genesis:
~1000 cpu-sec



Conditional GAN (CGAN):
provide knob to control parameters

GAN (neural net):
~0.001 gpu-sec

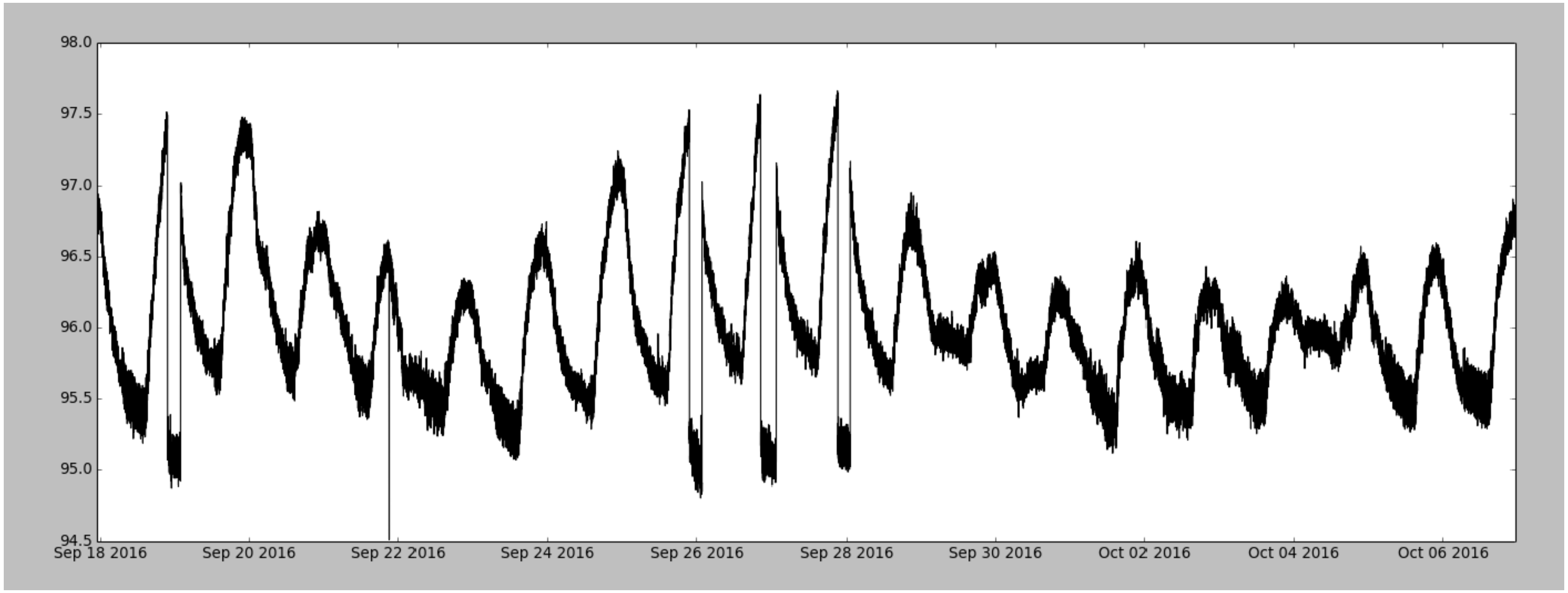
X. Ren

Unsupervised Learning: Anomaly/Breakout/Changepoint detection

Breakout Detection

1. Alarm handling (e.g. drifting temperatures)
2. Identification of anomalous conditions (e.g. shorted quadrupole magnet)
3. Machine configuration setup (e.g. global optimization)

LCLS Temperature monitor (timing system)



N. Norvell

Unsupervised Learning: Anomaly/Breakout/Changepoint detection

SLAC

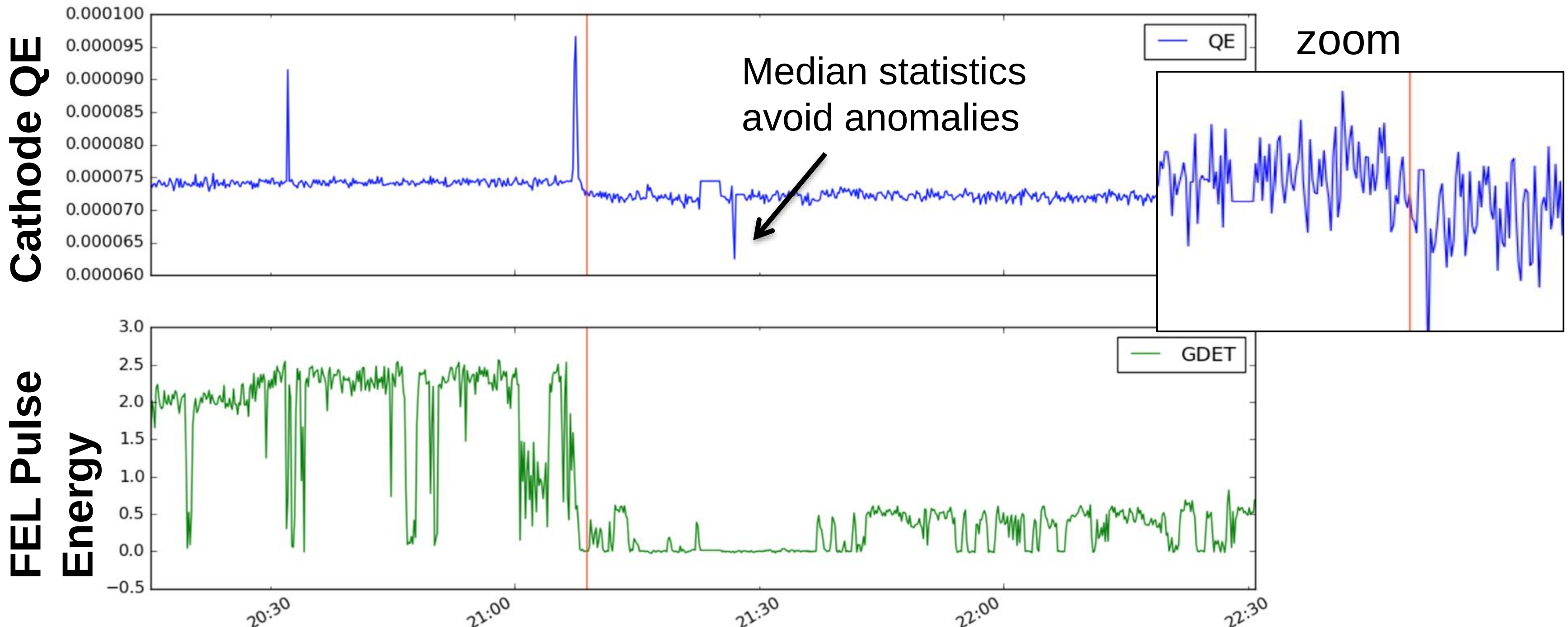
Breakout Detection

Leveraging Cloud Data to Mitigate User Experience from 'Breaking Bad'

Nicholas A. James[†] Arun Kejariwal[‡] David S. Matteson[†]
[†]Cornell University [‡]Twitter Inc.

$$\mathcal{E}(X,Y) = 2E|X-Y| - E|X-X'| - E|Y-Y'|$$

Cathode quantum efficiency drop caused hours of downtime.

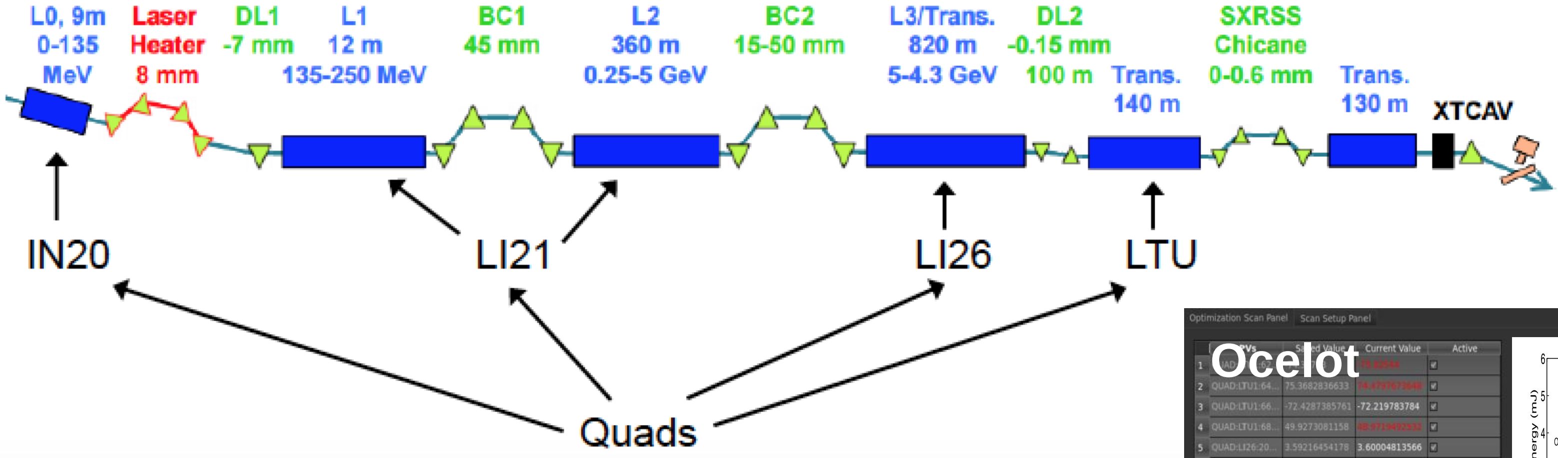


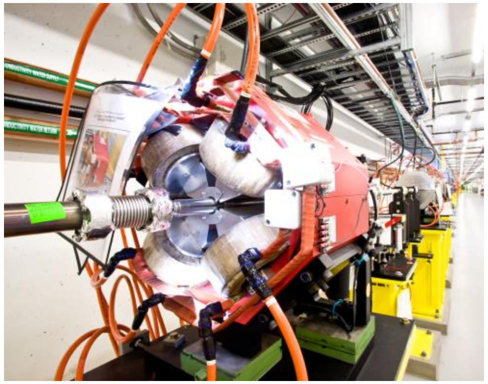
D. Sanzone

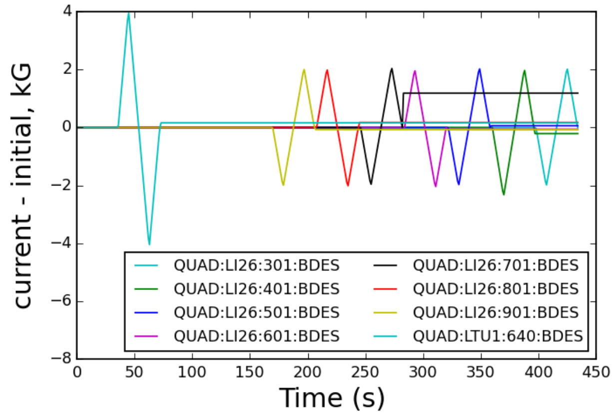
Supervised Learning:
Optimization – Online tuning

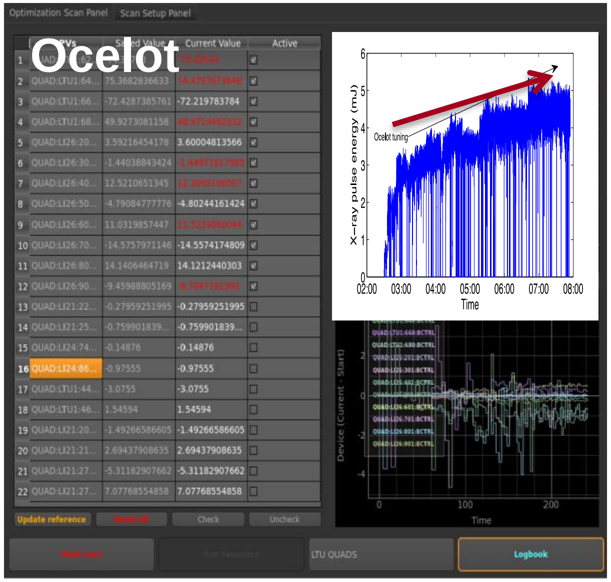
Online tuning:

- Twice daily, ~500 of hours/year
- A single task, quadrupole tuning, required 1 hour/day









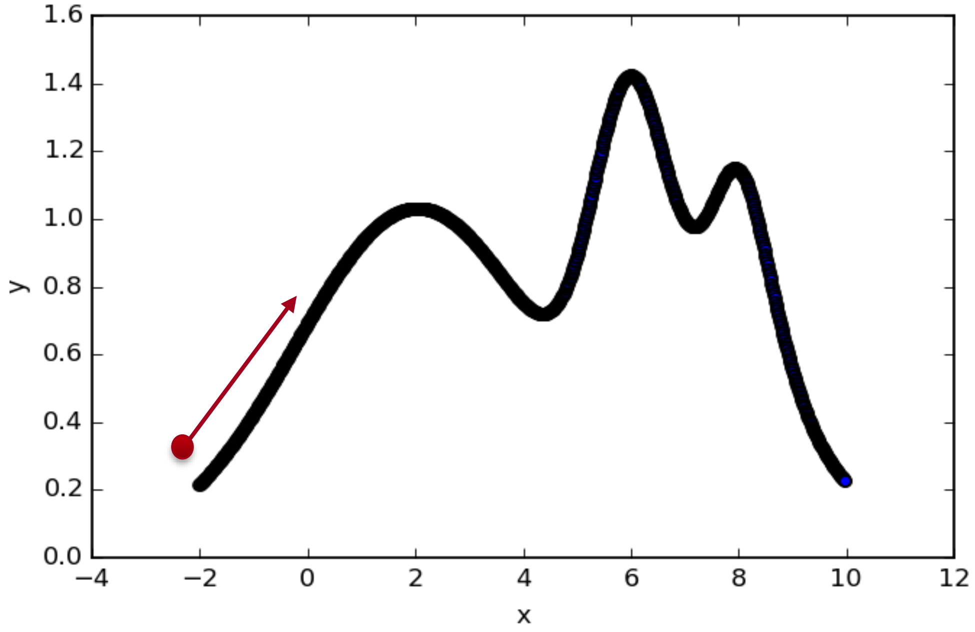
Supervised Learning: Optimization – Bayesian optimization

Model-based optimization

Advantage 1: Balance “exploitation vs. exploration”

➔ Find global maximum

Gradient optimizer



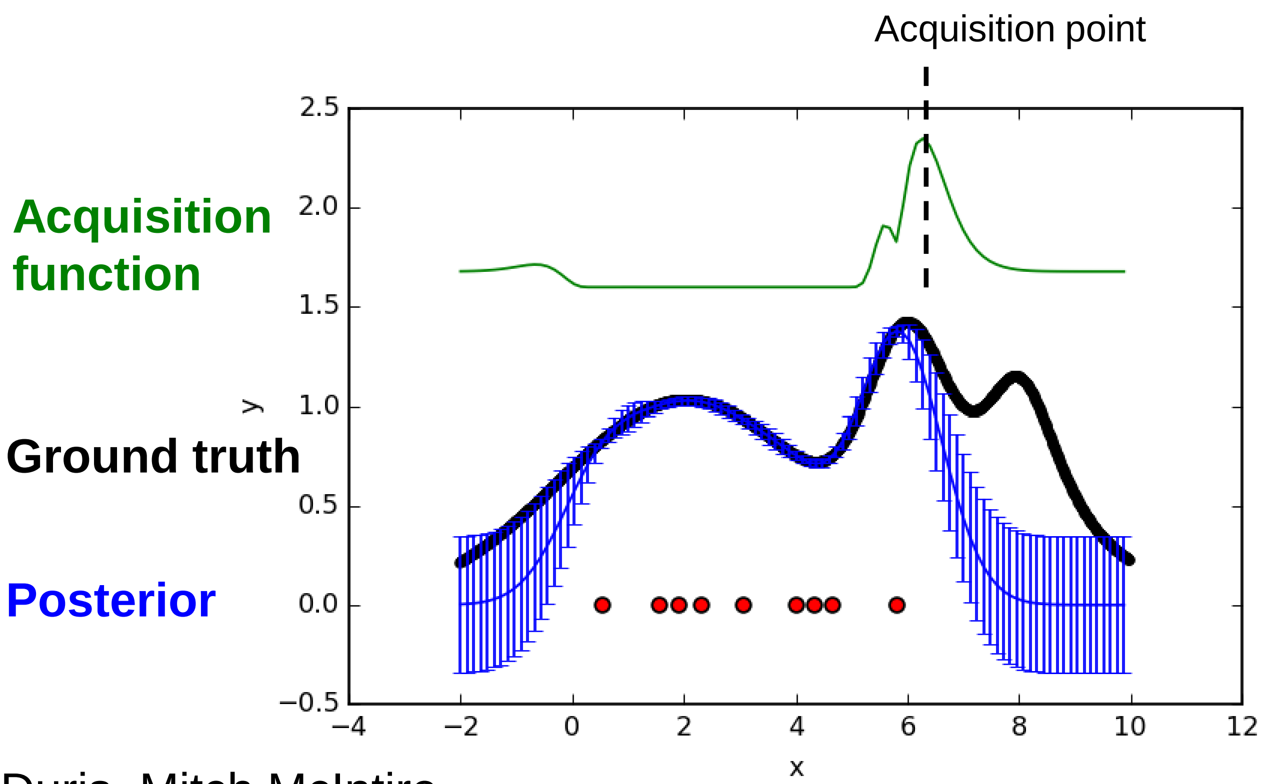
Joe Duris, Mitch McIntire

Supervised Learning: Optimization – Bayesian optimization

Model-based optimization

Advantage 1: Balance "exploitation vs. exploration"

➔ Find global maximum



Joe Duris, Mitch McIntire

Supervised Learning: Optimization – Bayesian optimization

Model-based optimization

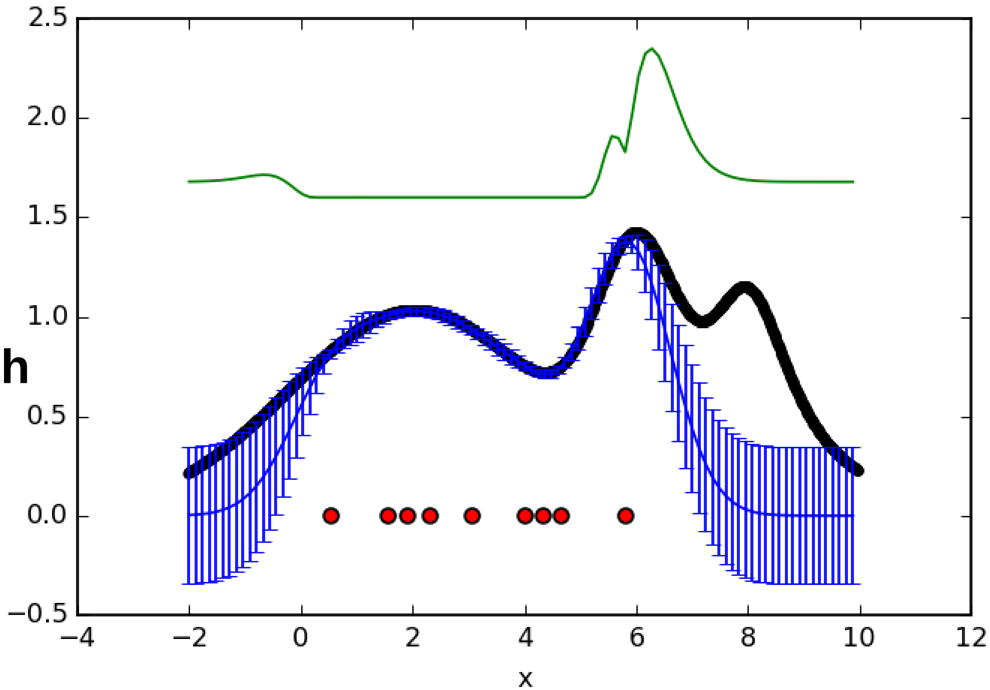
Advantage 2: Model can incorporate physics, experience

➔ Learn from data, simulations

Acquisition function

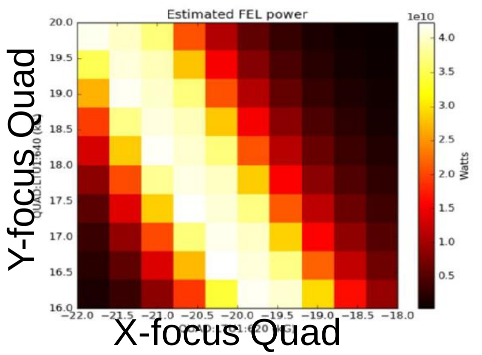
Ground truth

Posterior

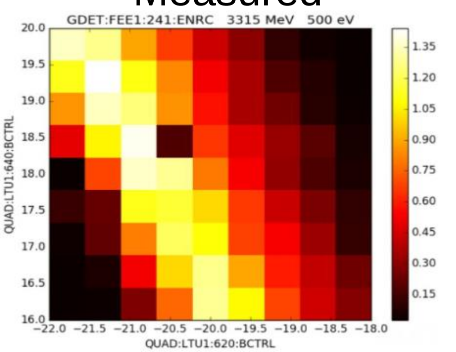


FEL vs quads

Modeled



Measured



Supervised Learning:
Optimization – Bayesian optimization

Gaussian process: instance based learning method

Kernel (covariance): $$k(x_1, x_2) = \theta e^{-(x_1 - x_2)^T \Lambda (x_1 - x_2)}$$

observations

new point to predict

prior mean

new point

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & K_*^{\mathrm{T}} \\ K_* & K_{**} \end{bmatrix}\right)$$

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & \cdots & k(x_*, x_n) \end{bmatrix}$$
$$K_{**} = k(x_*, x_*)$$

taken from M. Ebner, GP for Regression

Supervised Learning:
Optimization – Bayesian optimization

SLAC

Gaussian process: instance based learning method

Kernel (covariance): $k(x_1, x_2) = \theta e^{-(x_1 - x_2)^T \Lambda (x_1 - x_2)}$

prior mean

observations

new point to predict

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & K_*^{\mathrm{T}} \\ K_* & K_{**} \end{bmatrix}\right)$$

new point

Prediction of new point: $\overline{y}_* = K_* K^{-1} \mathbf{y}$

Variance of new point: $\mathrm{var}(y_*) = K_{**} - K_* K^{-1} K_*^{\mathrm{T}}$

taken from M. Ebner, GP for Regression

Supervised Learning: Optimization – Bayesian optimization

Gaussian process: instance based learning method

Kernel (covariance): $k(x_1, x_2) = \theta e^{-(x_1 - x_2)^T \Lambda (x_1 - x_2)}$

observations

new point to predict

prior mean

new point

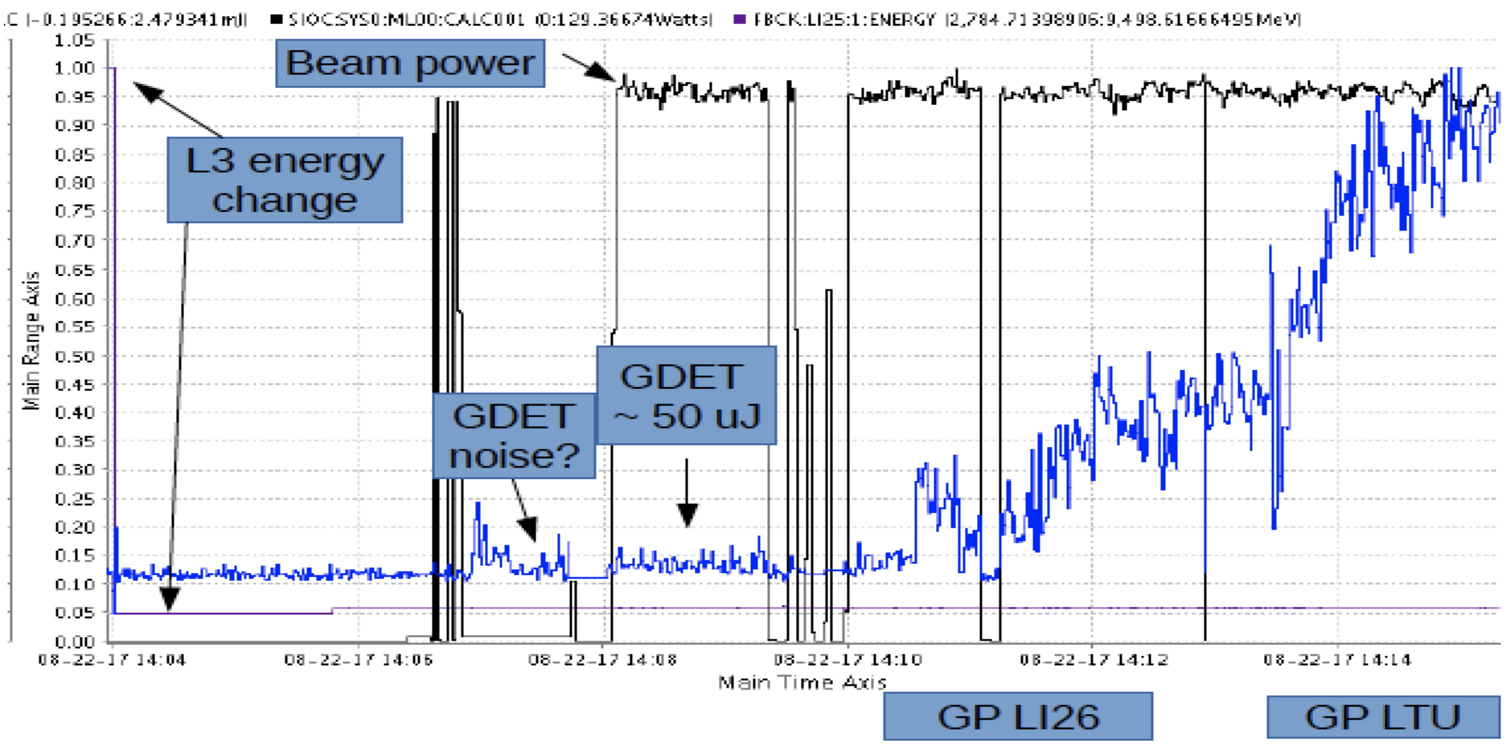
$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & K_*^{\mathrm{T}} \\ K_* & K_{**} \end{bmatrix}\right)$$

Acquisition function: $UCB(x^*) = \mu(x^*) + \sqrt{(\nu \tau_t)}\sigma(x^*)$

$$\tau(t) = 2\log(t^{d/2+2}\pi^2/3\delta),\ 0 < \delta < 1,\ 0 < \nu$$

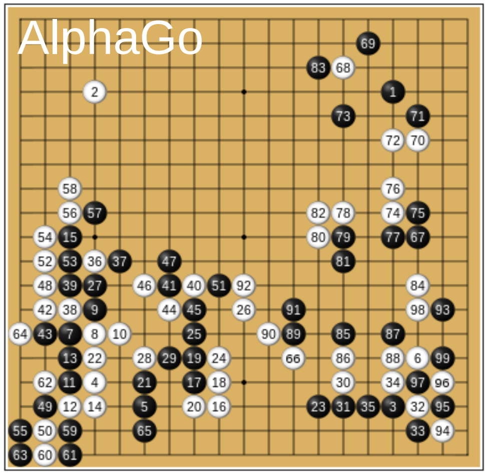
Example: tuning quadrupoles from noise

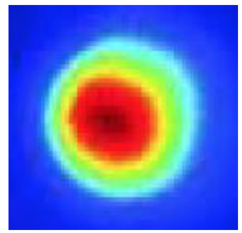
Model knows history ➔ makes educated guesses where to explore

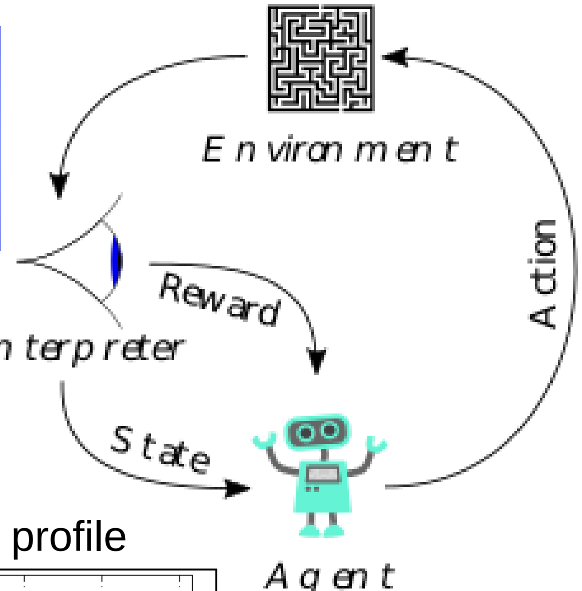


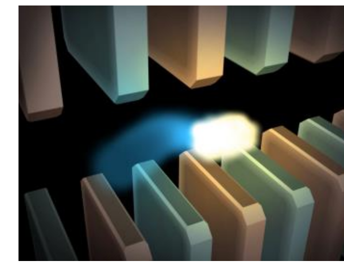
Reinforcement Learning: Optimization – Online tuning

Treat optimization like a game: FEL power is the score

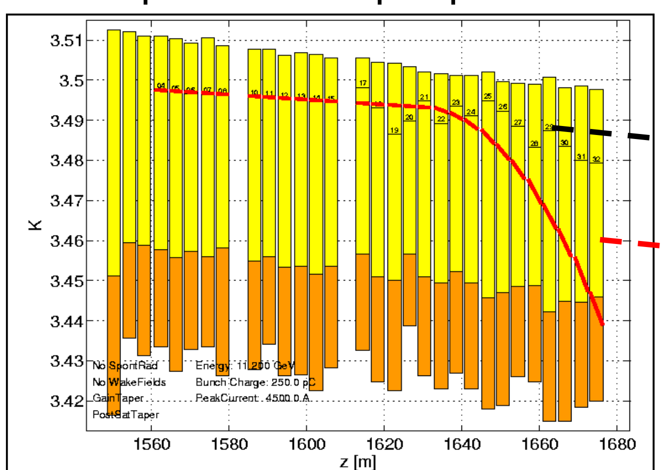




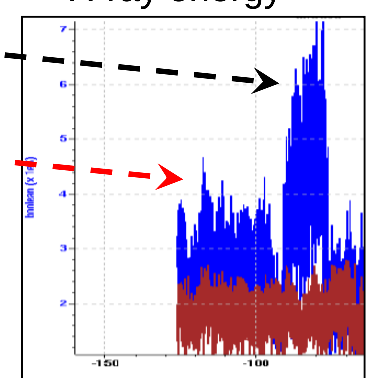




Experiment: Taper profile



X-ray energy



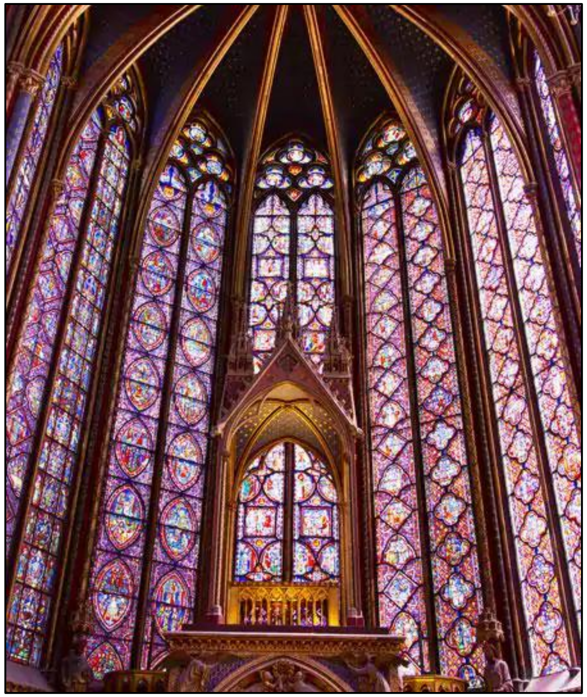
LCLS Experiment: 5.5 KeV Self-seeding FEL, **Zig-zag** doubles power from **continuous profile**

Juhao Wu

Ghost Imaging / Single Pixel Camera

Riddle: How can I take a picture with a spectrometer?

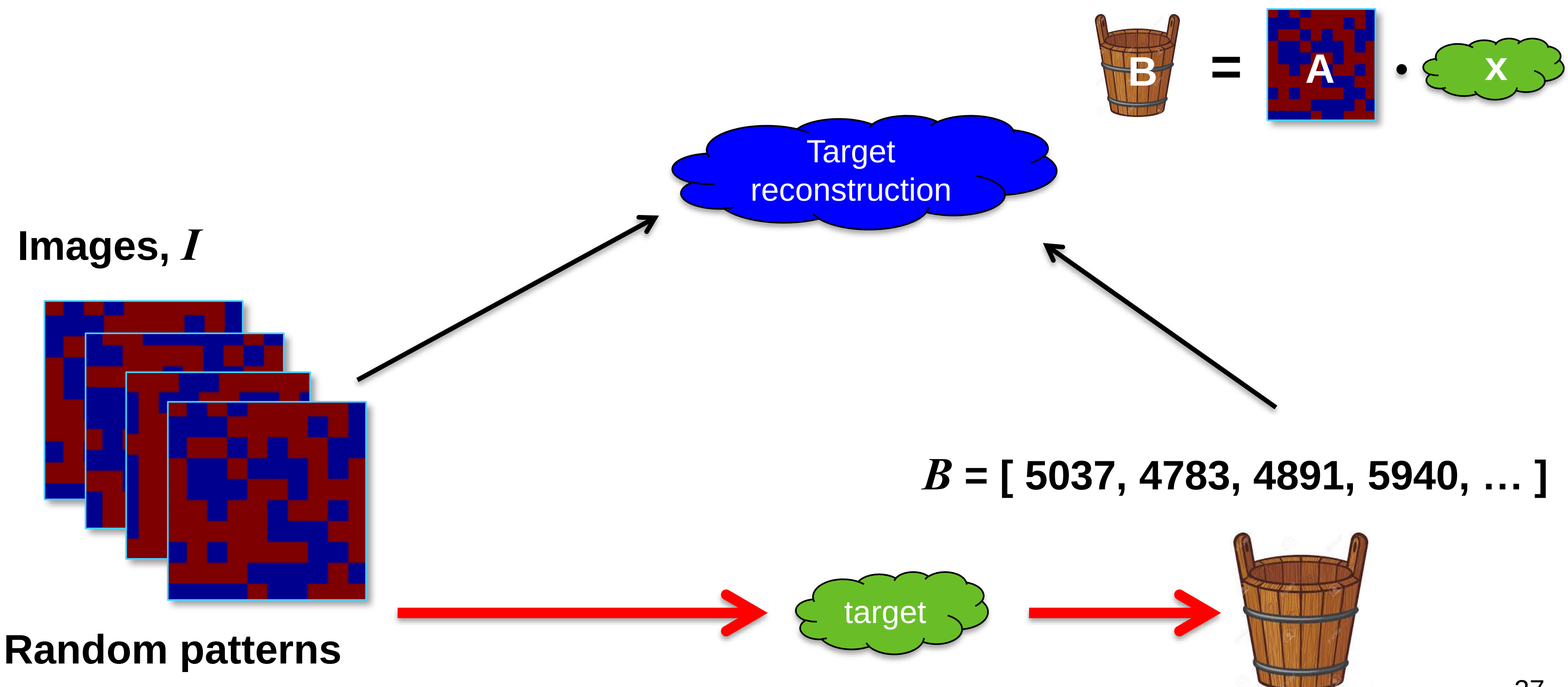
Answer: Have a friend with a flashlight



Supervised Learning:
Statistical methods for data analysis

Ghost Imaging / Single Pixel Camera

$$\mathbf{x}^\star = \mathrm{argmin}_{\mathbf{x}} \left(||\mathbf{Ax}\text{-}\mathbf{B}||^2 + \lambda_2 ||\mathbf{x}||^2 + \lambda_1 \sum_j |x_j| \right) \text{ subject to } x_j \geq 0$$

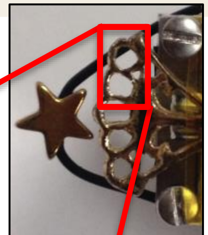


Compressive sensing + ghost imaging
➔ Compressive ghost imaging

Compressive sensing concept:
Why record 660k points if only 39k needed?

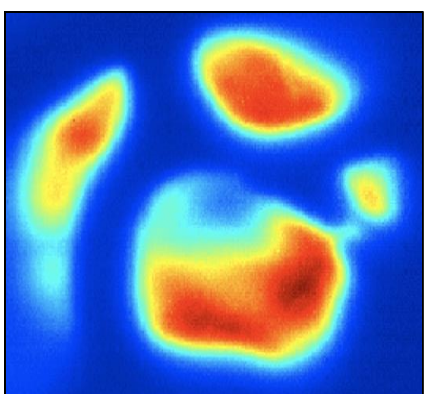
Supervised Learning:
Statistical methods for data analysis

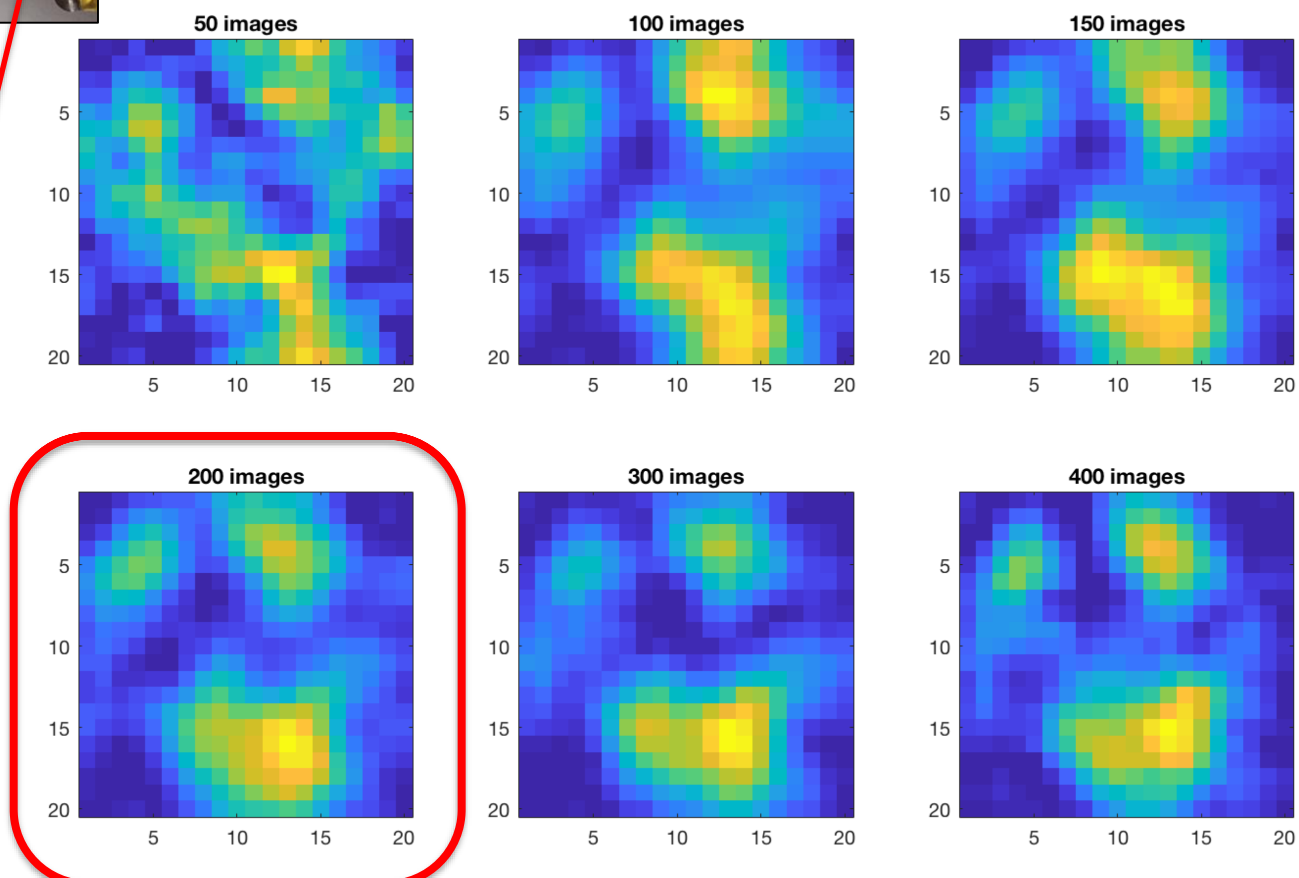
Experimental Results



Target

Transmission at camera

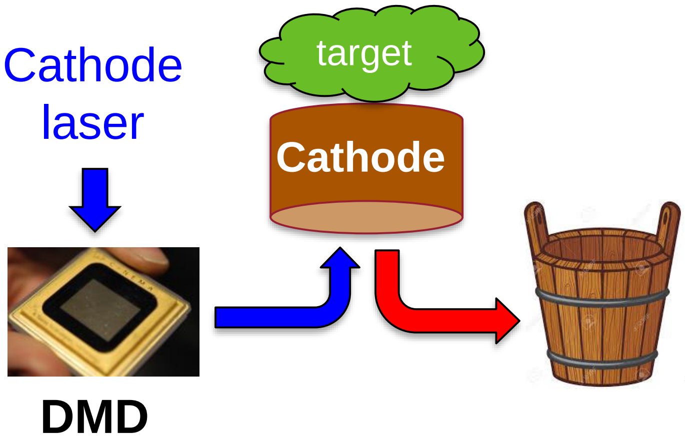


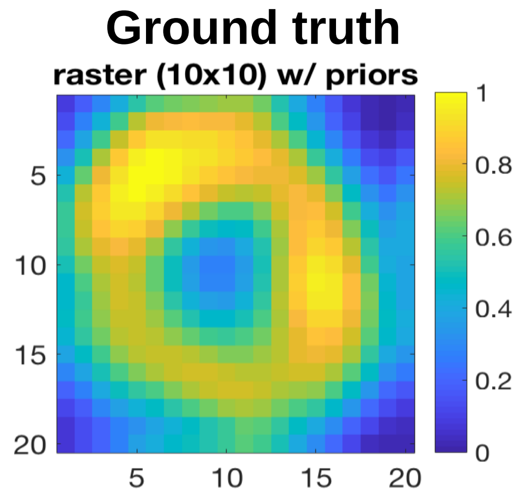


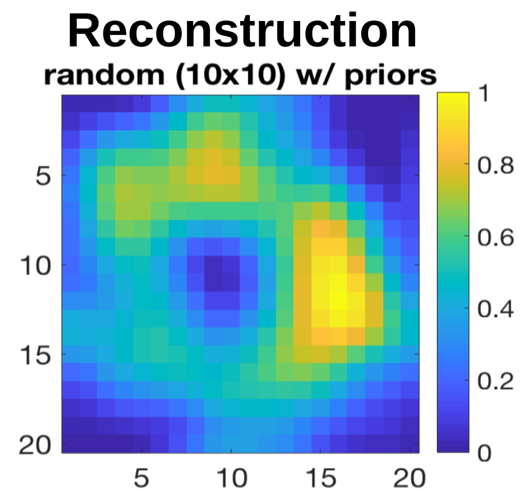
200 images pretty good!

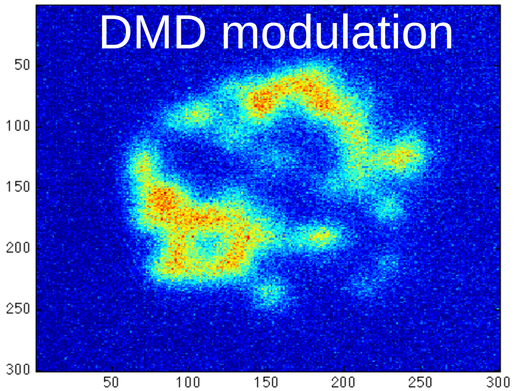
S. Li

Example application: photocathode quantum efficiency

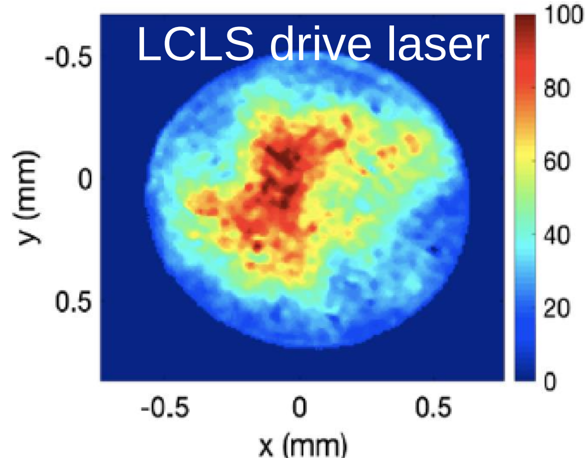




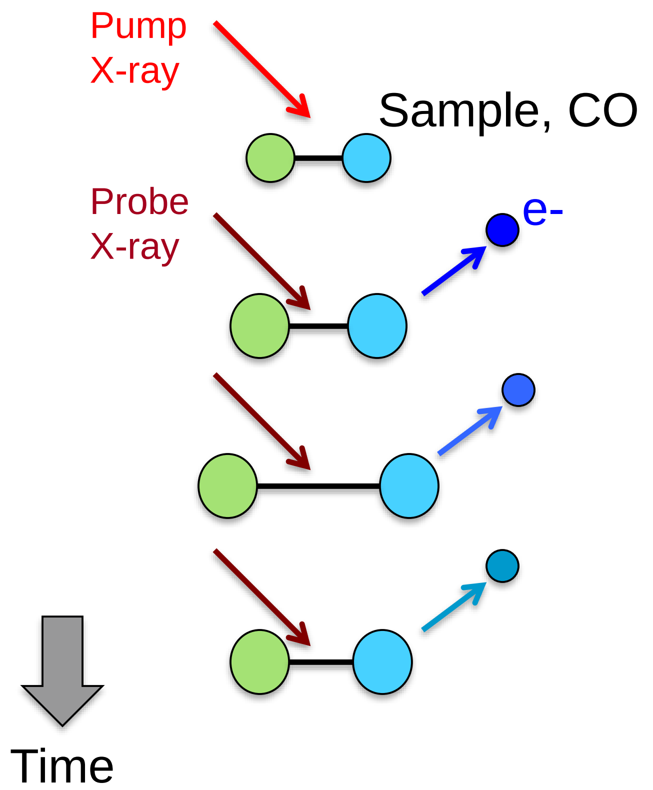


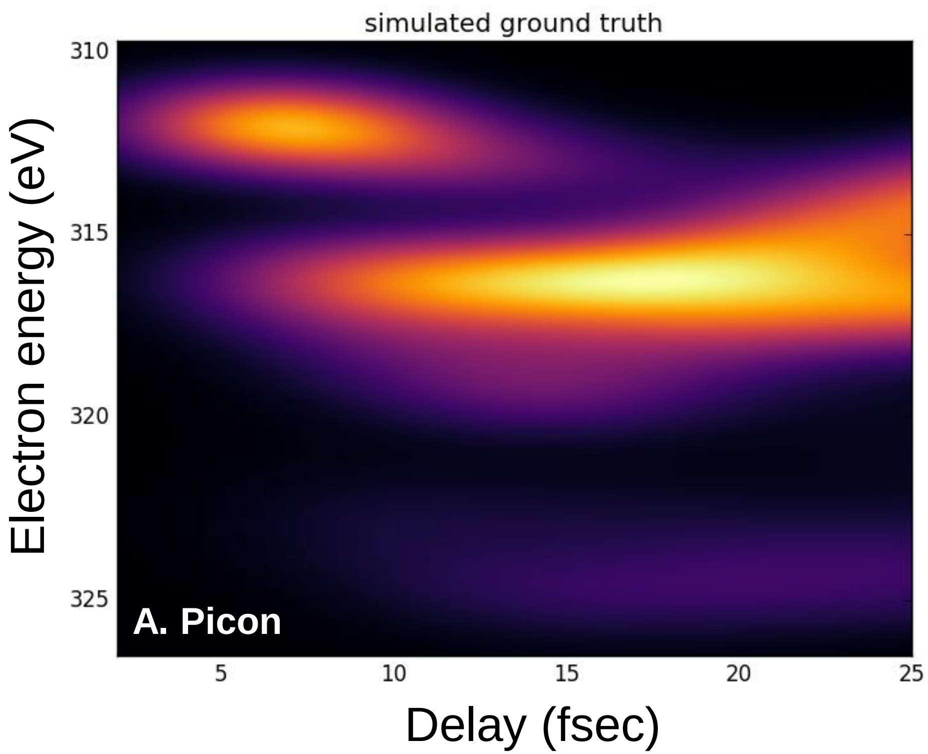


Don’t need DMD:
exploit natural variation,
jitter of drive laser



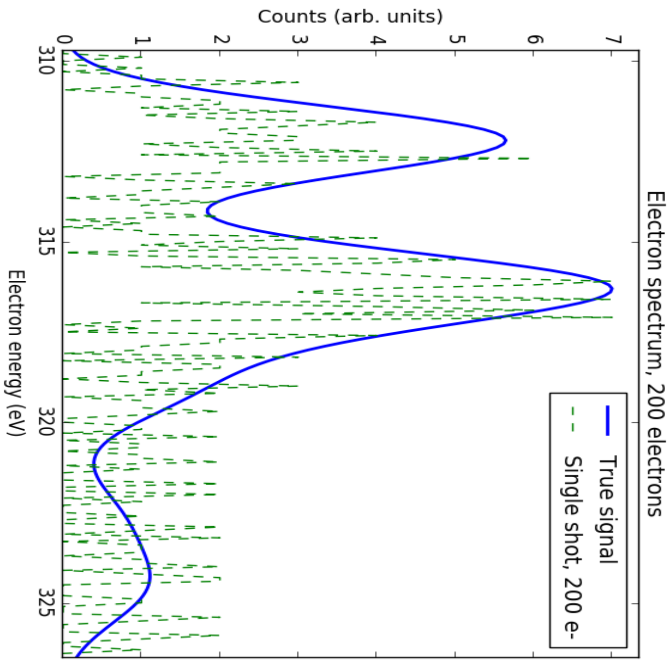
Supervised Learning:
Statistical methods for data analysis

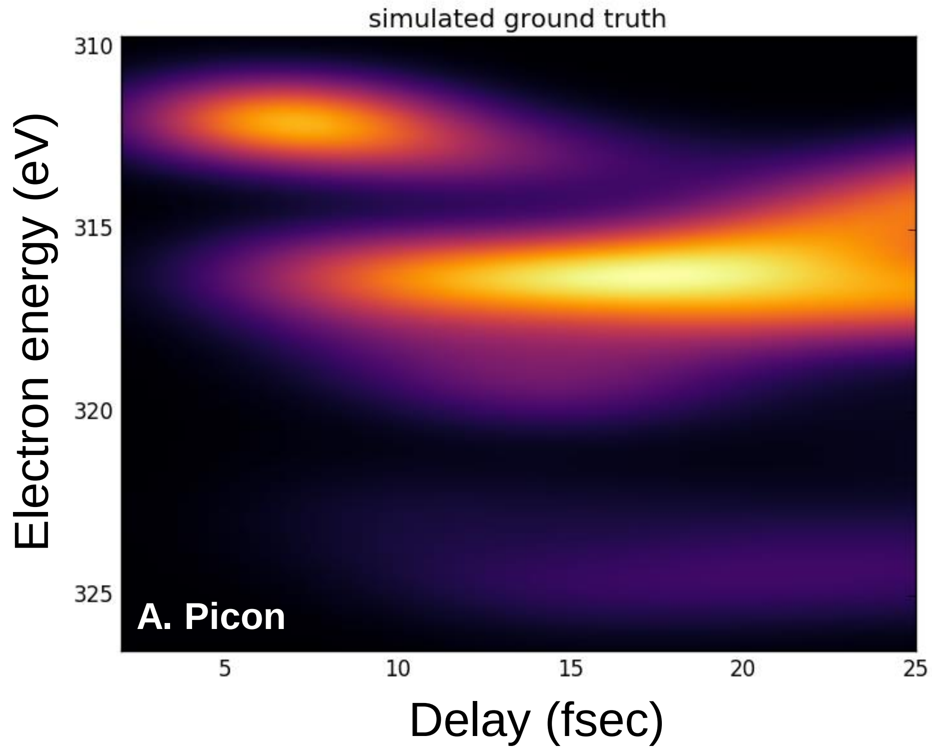




As system (CO) evolves after absorbing X-ray, electron energies change

Supervised Learning:
Statistical methods for data analysis





$$M_i(E)$$

$$R(\delta, E)$$

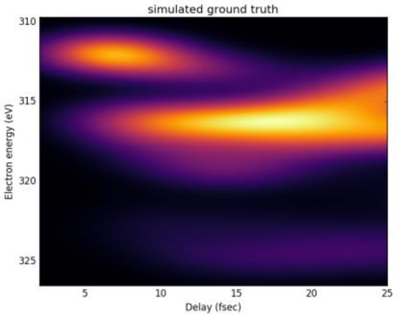
As system (CO) evolves after absorbing X-ray, electron energies change

Goal: can we recover R from M?

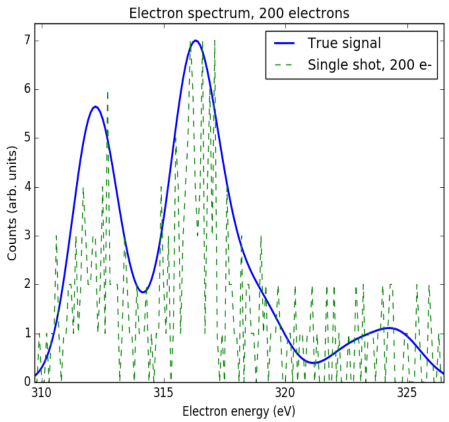
Supervised Learning:
Statistical methods for data analysis

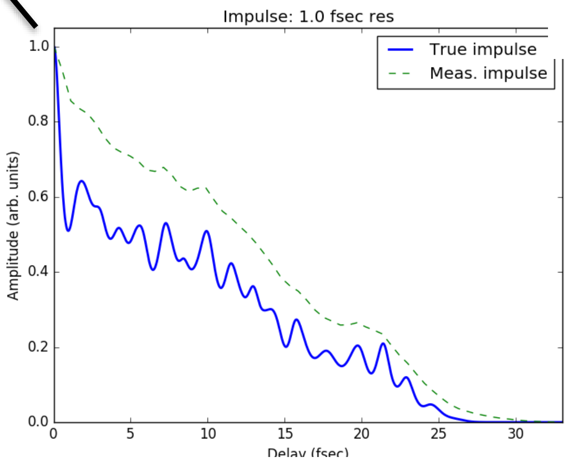
SLAC

Solving for the response, R(d,E):



$$\boxed{M = AR}$$





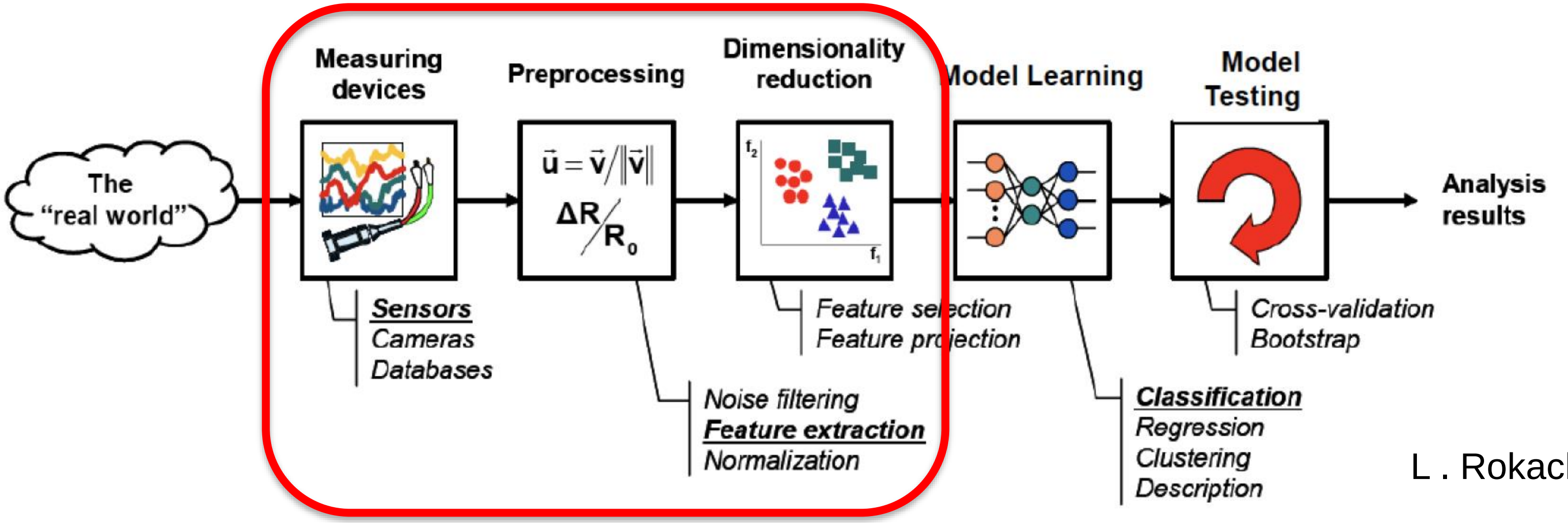
$$A_i(\delta) = \int_{-\infty}^{\infty} P_i(\tau)\, P_i(\tau - \delta)\, d\tau$$

Optimization formalism:

$$\mathbf{R}^\star = \mathrm{argmin}_{\mathbf{R}} \left(||\mathbf{M\text{-}AR}||^2 + \lambda_2 ||\mathbf{R}||^2 + \lambda_1 \sum_j |R_j| \right) \text{ subject to } R_j \geq 0$$

Conclusion

Some personal opinions on ML for accelerators



L . Rokach

1. No data ➔ no learning: #1 task is building data
2. Noise, outliers, dropped data will dominate performance: #2 task is cleaning
3. Deep learning is the dream… but time spent thinking about physics is well-rewarded.

Applications for XFELs (and other accelerators)

1. **Online tuning:** transverse matching, longitudinal phase space, X-ray spectrum, emittance minimization, etc.)
2. **Surrogate modeling:** efficient machine design, user support, predictive control
3. **Data analysis:** X-ray pulse reconstructions, electron parameters, user experiments
4. **Prognostics:** Fault prediction, fault recovery, identification of anomalous conditions

When is ML useful?

- Tasks that humans can do, but hard to describe…
- When data is abundant and well labeled
- When simple algorithms fails
- When the goal is worth the effort

ML should be your last resort!

Conclusion

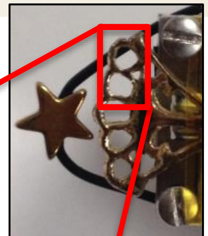
Thanks for your attention!

And thanks to the people who did the work shown here:

E. Cropp, J. Duris, A. Edelen, K. Kabra, D. Kennedy, T. J. Lane, S. Li, T. Maxwell, P. Musumeci, X. Ren, J. Wu, X. Zhang

Data Analysis:
Statistical methods for data analysis

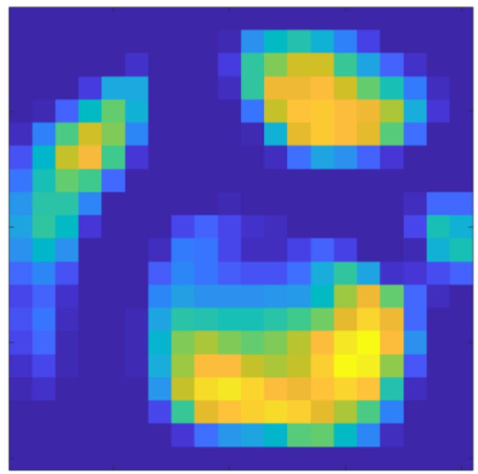
Experimental Results



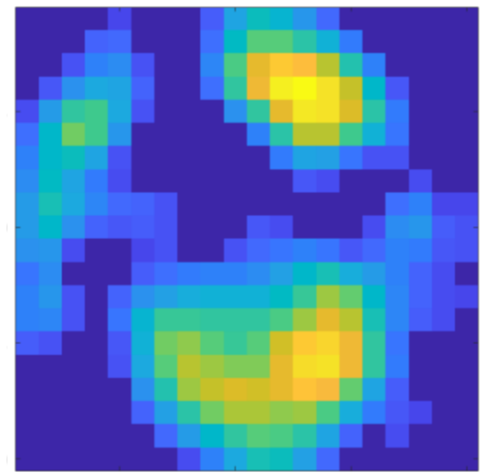
Target

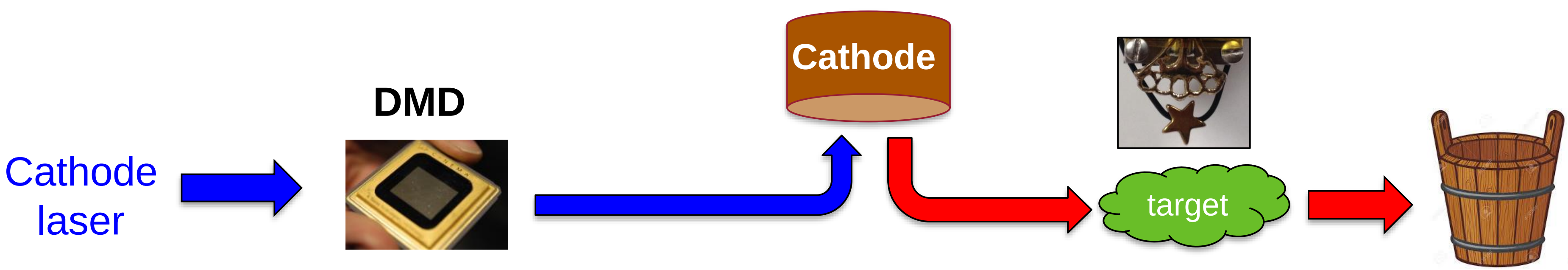
Transmission at camera

Ground truth



ADMM reconstruction

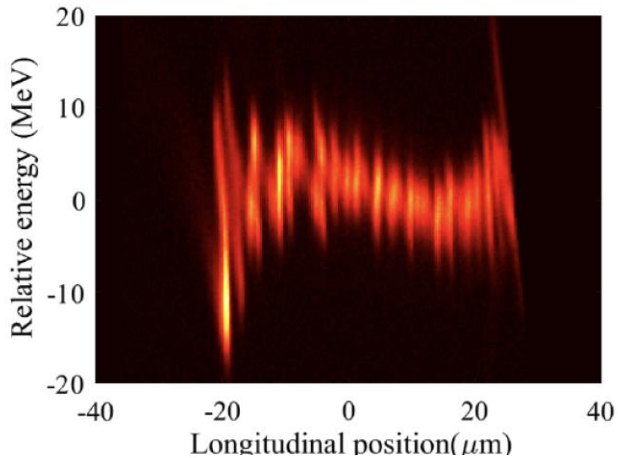




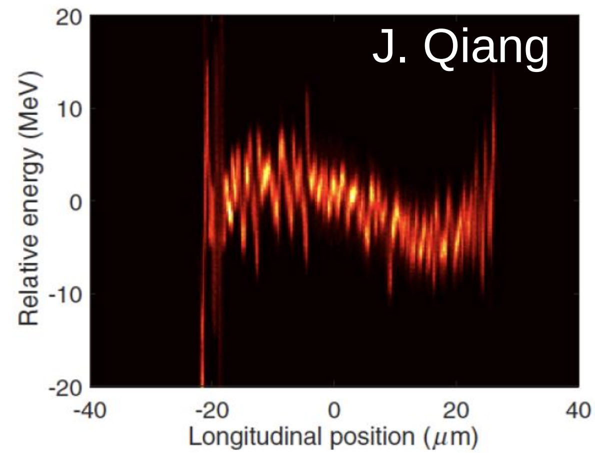
Surrogate Models: Accelerator models

High fidelity physics simulations are remarkable:

LCLS microbunching instability



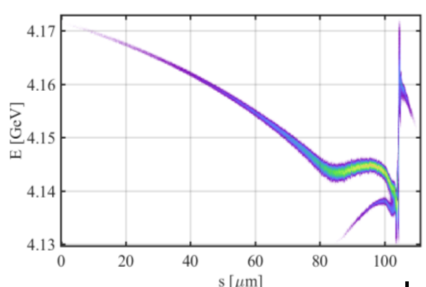
measurement

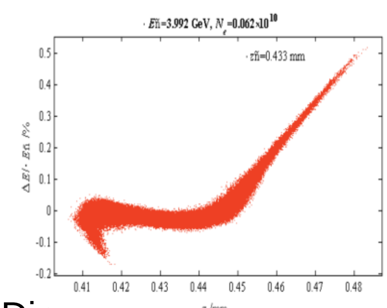


simulation

…but also slow. (e.g. hours on NERSC)

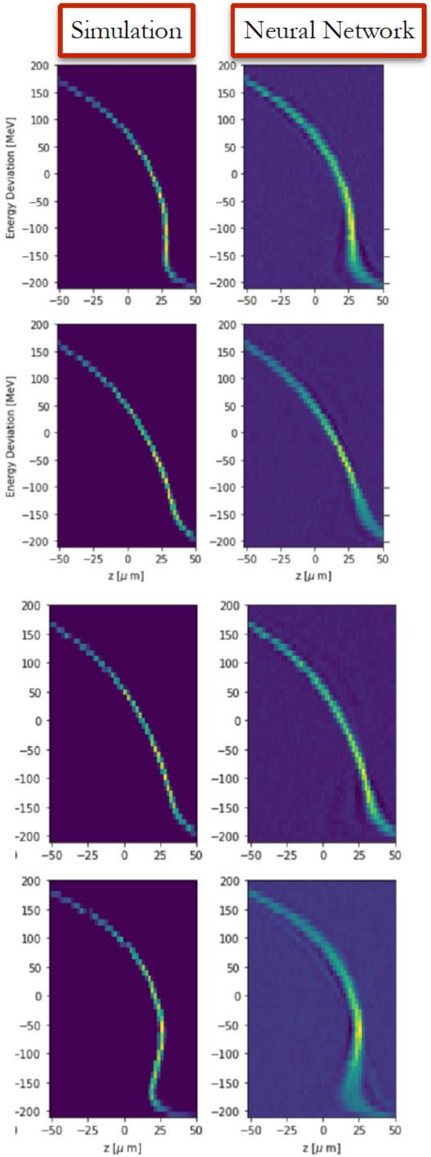
How can we best support design of a new machine?



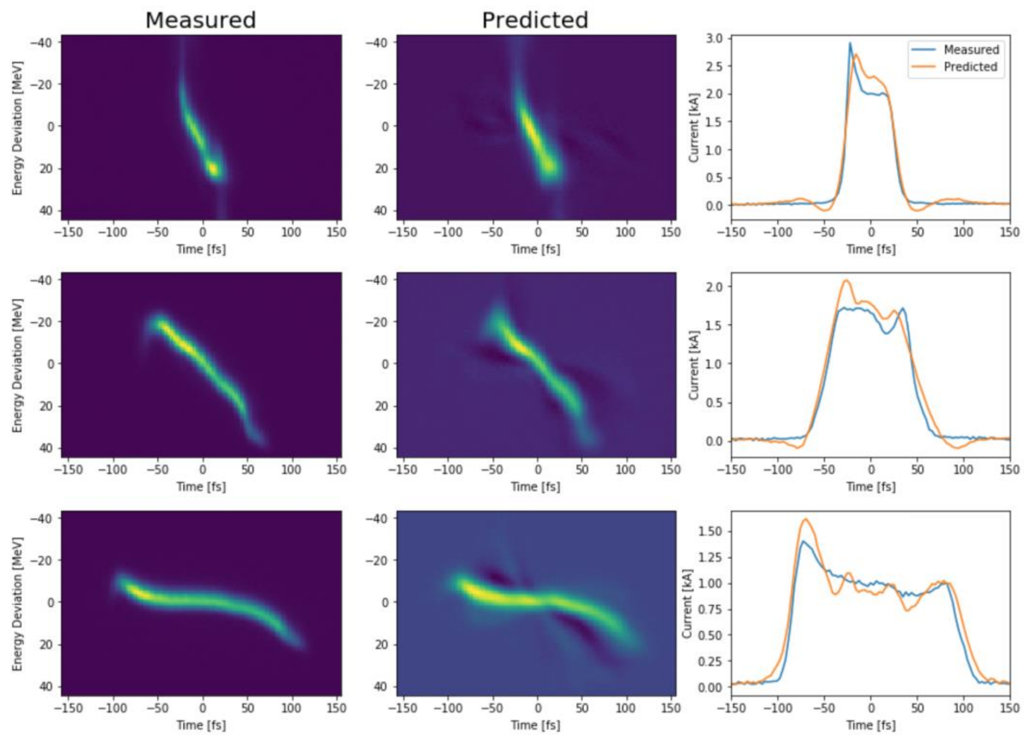


LCLS-II simulations, Y. Ding

Surrogate Models: Accelerator models

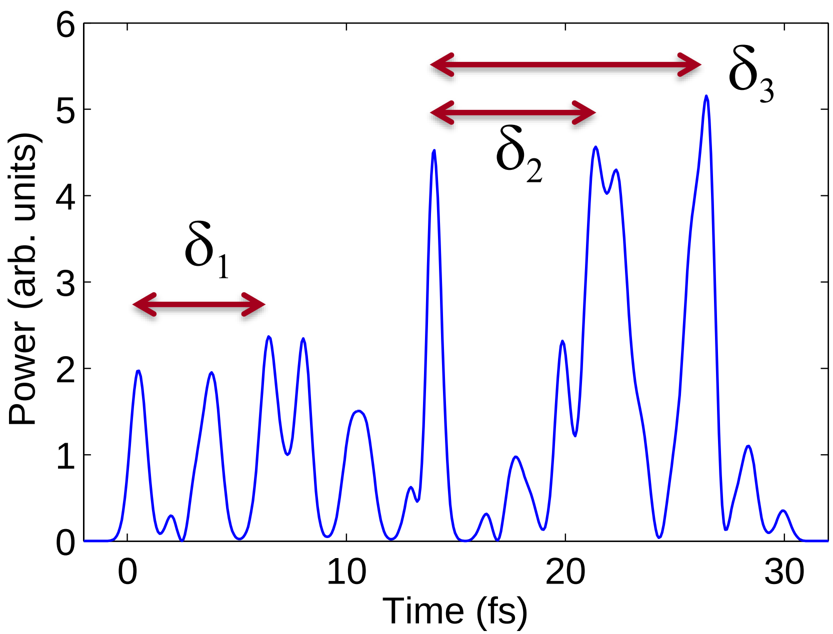


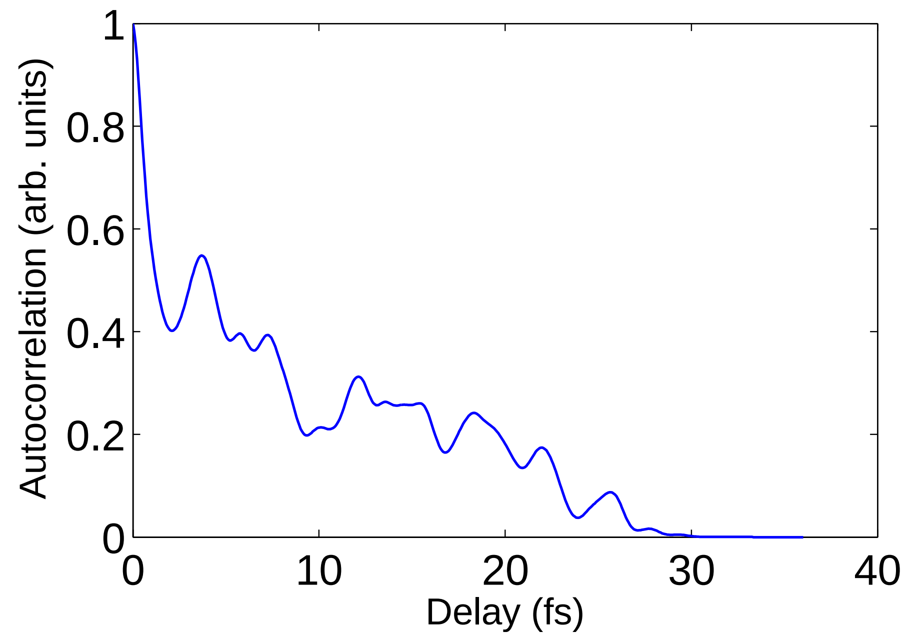
- Predict XTCAV image from other diagnostic output or upstream machine settings to create a non-destructive **virtual diagnostic**
- Simulation + neural network results match well for FACET-II *(see left)*
- Small study with LCLS machine data and XTCAV images *(scan of L1S phase and BC2 peak current at 13.4 GeV)*



Emma, Edelen, et al. in preparation

TDGI Example

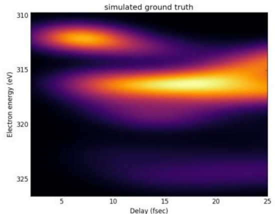


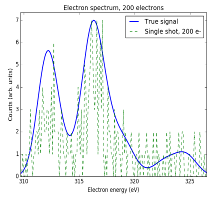


$$P_i(t)$$

$$A_i(\delta) = \int_{-\infty}^{\infty} P_i(\tau)\, P_i(\tau - \delta)\, d\tau$$

Signal determined by probability of two photons separated by time δ





$$\boxed{M = AR}$$