Machine Learning at LCLS

Oct 24, 2016

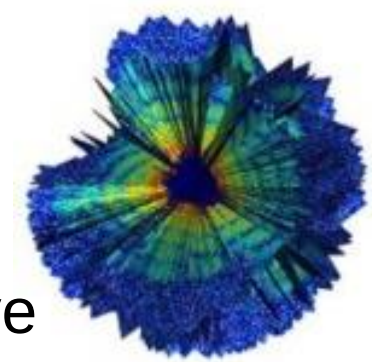
D. Ratner, A. Ahmed, T. Cope, J. Duris, S. Ermon, M. Gibbs, T. J. Lane, S. Li, T. Maxwell, M. McIntire, M. Mongia, N. Norvell, D. Sanzone, D. Schneider, C. Yoon

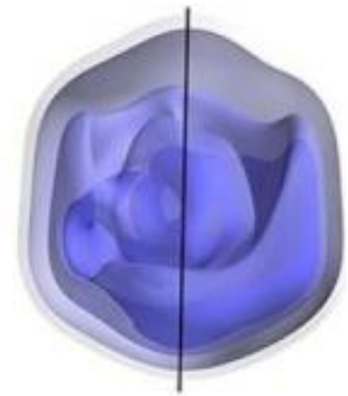
SLAC National Accelerator Laboratory

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Big Data comes to Photon Science



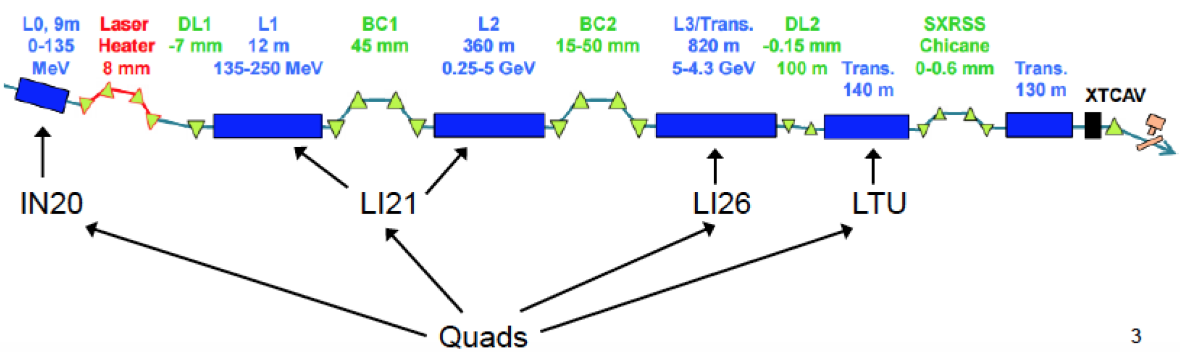


User side:

LCLS: 120 Hz images → 15 TB/hour
LCLS-II: 100 kHz → 1 PB/hour!
➔ exascale computing initiative

Machine side:

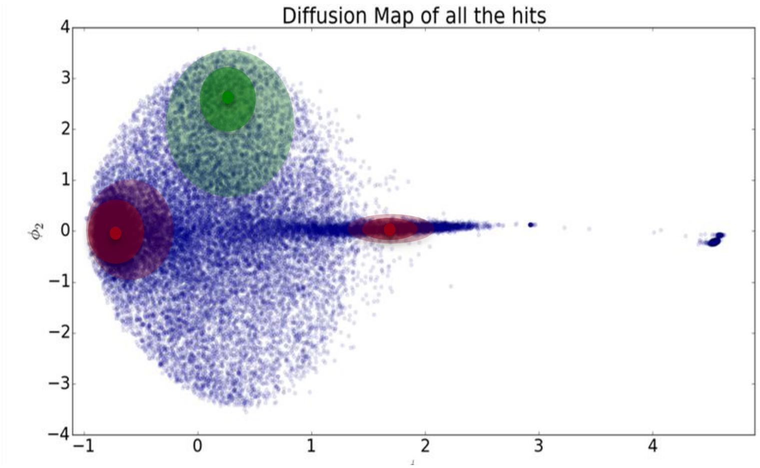
Archive 200k variables at 1Hz → 10^{12} data points so far
Online optimization of ~30 dimensional space
Alarm/anomaly/breakout handling



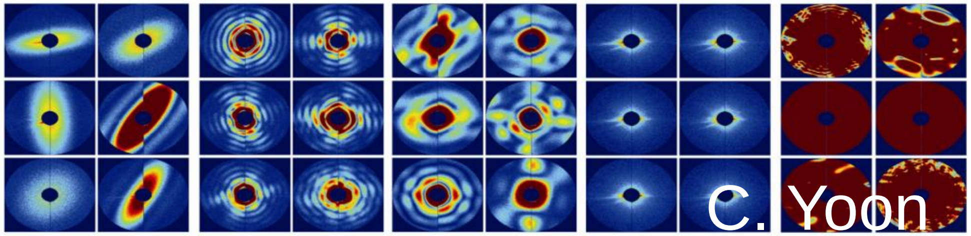
Computer vision: biological imaging (C. Yoon)

SLAC

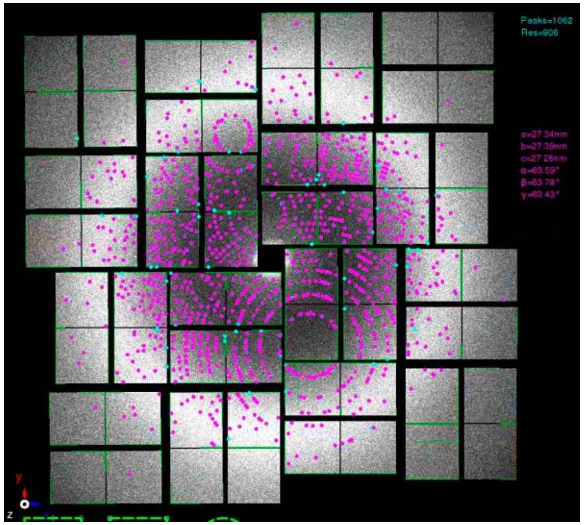
Classification of single particle images

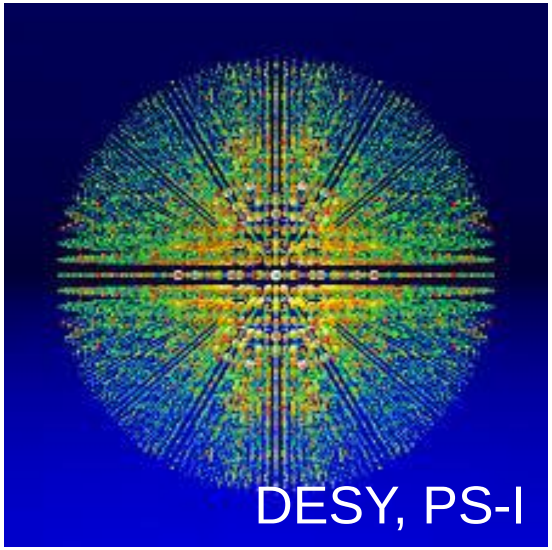


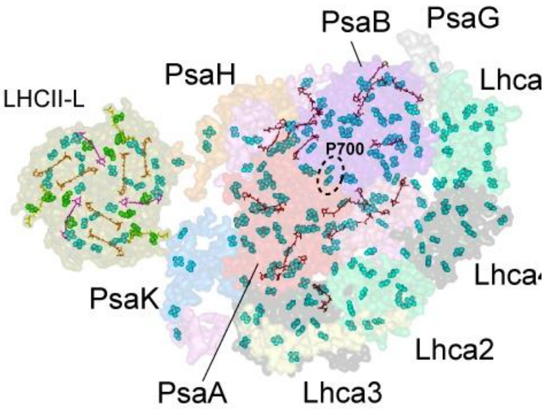
C. Yoon, A. AbuHashem



Indexing and classification of nano-xtal images (Google Accelerated Science)

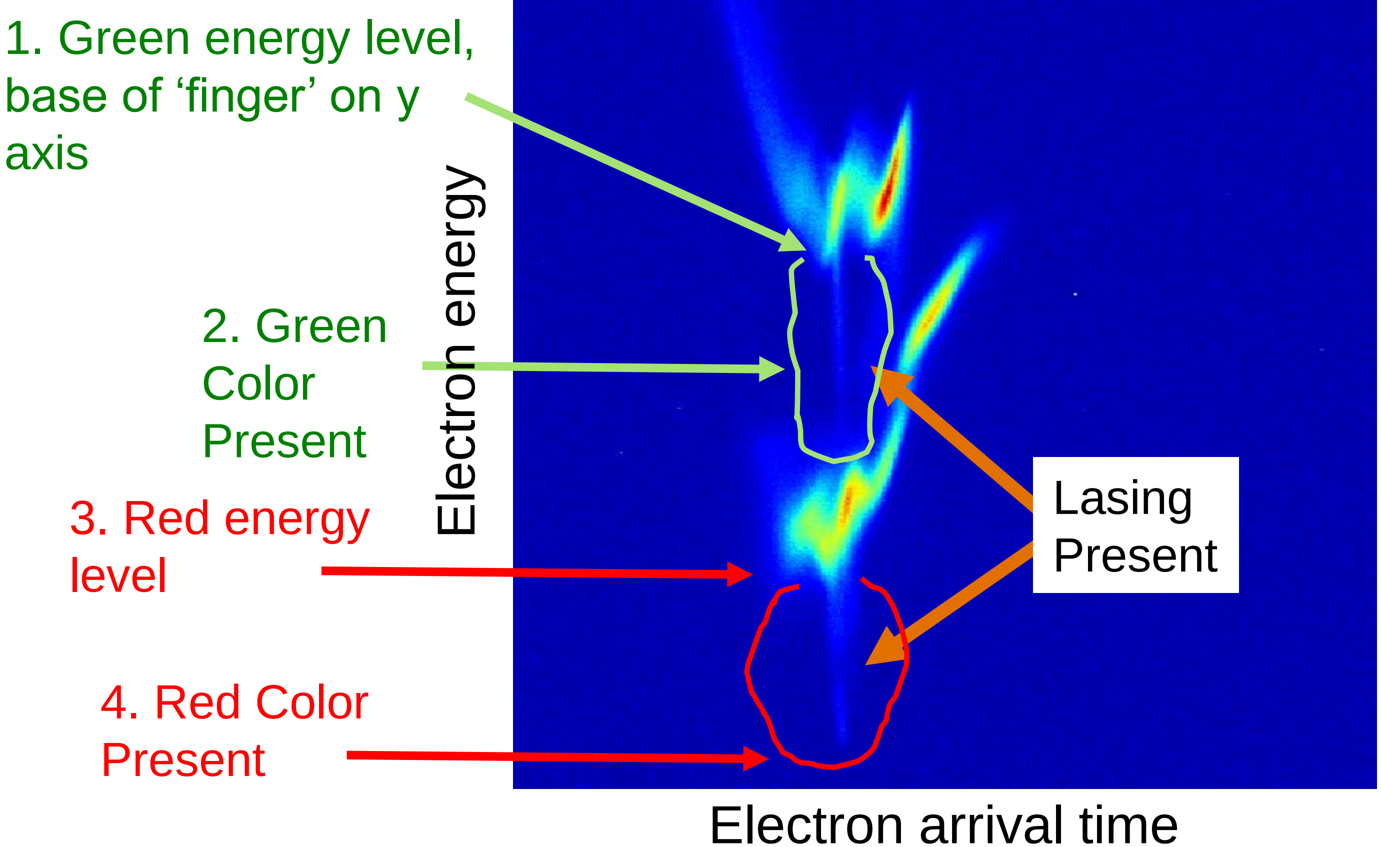






Computer vision: X-ray/electron beams (D. Schneider)

XTCAV electron diagnostic: best source of X-ray temporal info!



Computer vision: X-ray/electron beams (D. Schneider)

1) start with fully trained ***ImageNet*** based convnet

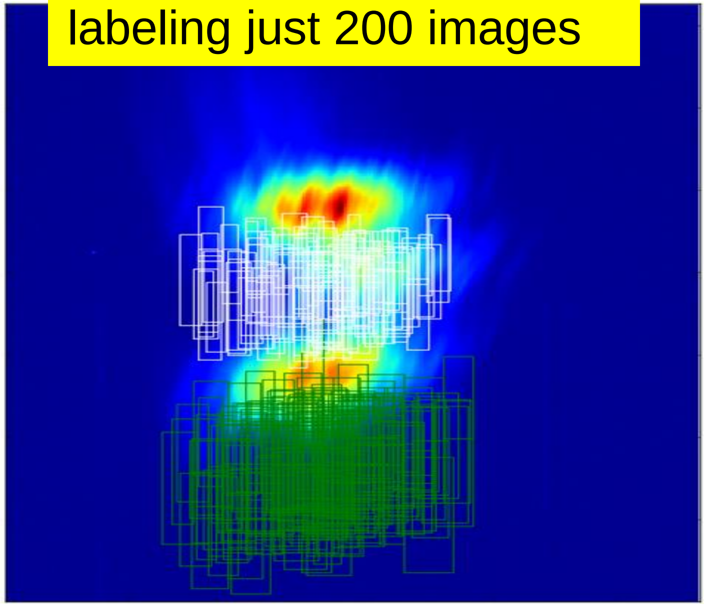
Millions of images

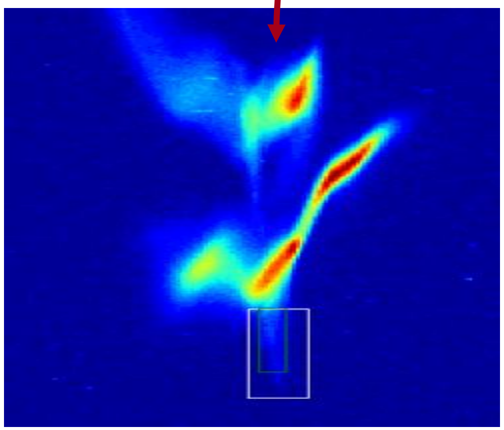
Classifies a 1000 categories - cat, dog, bus, etc

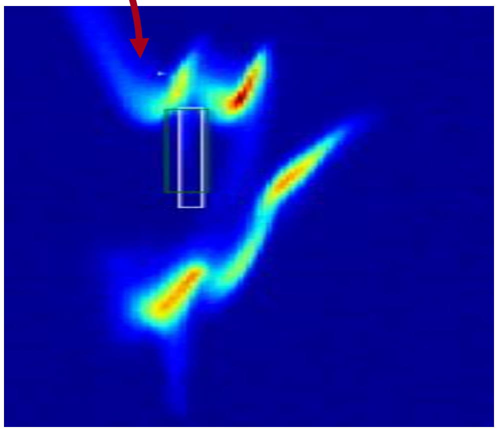
3) train new ***regression heads*** on 200 images

4) Good results

2) spend ***only 3 hours*** labeling just 200 images

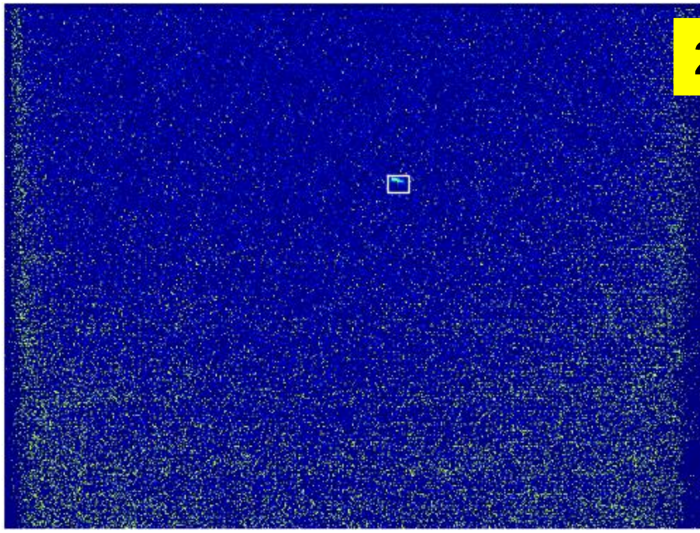






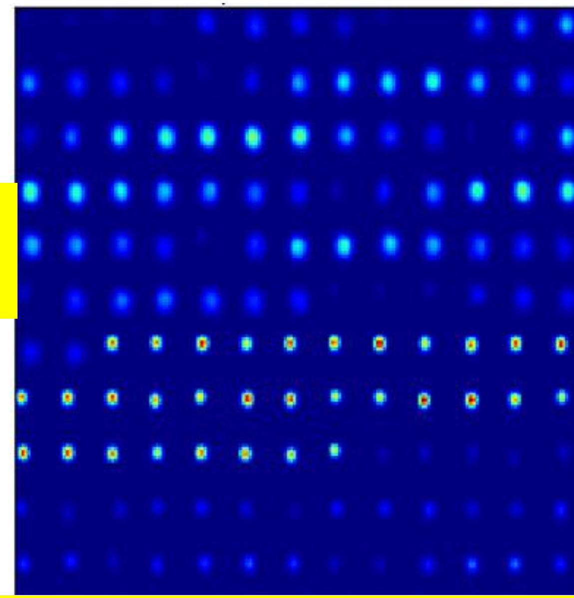
Convnet figure from
http://www.cs.toronto.edu/~frossard/post/vgg16/

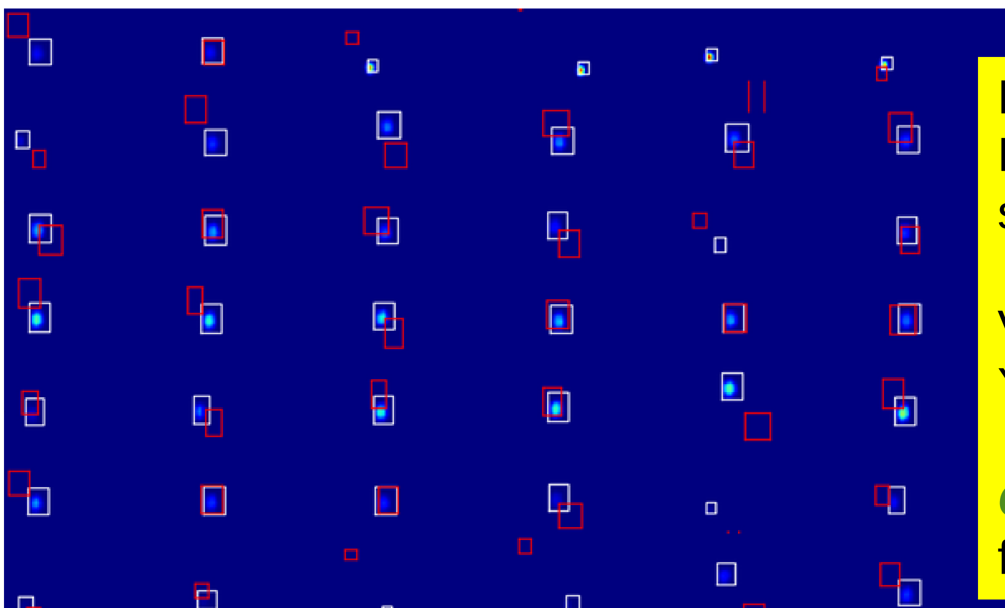
Computer vision: X-ray/electron beams (D. Schneider)



250 labeled boxes

For VCC – high variation in beam





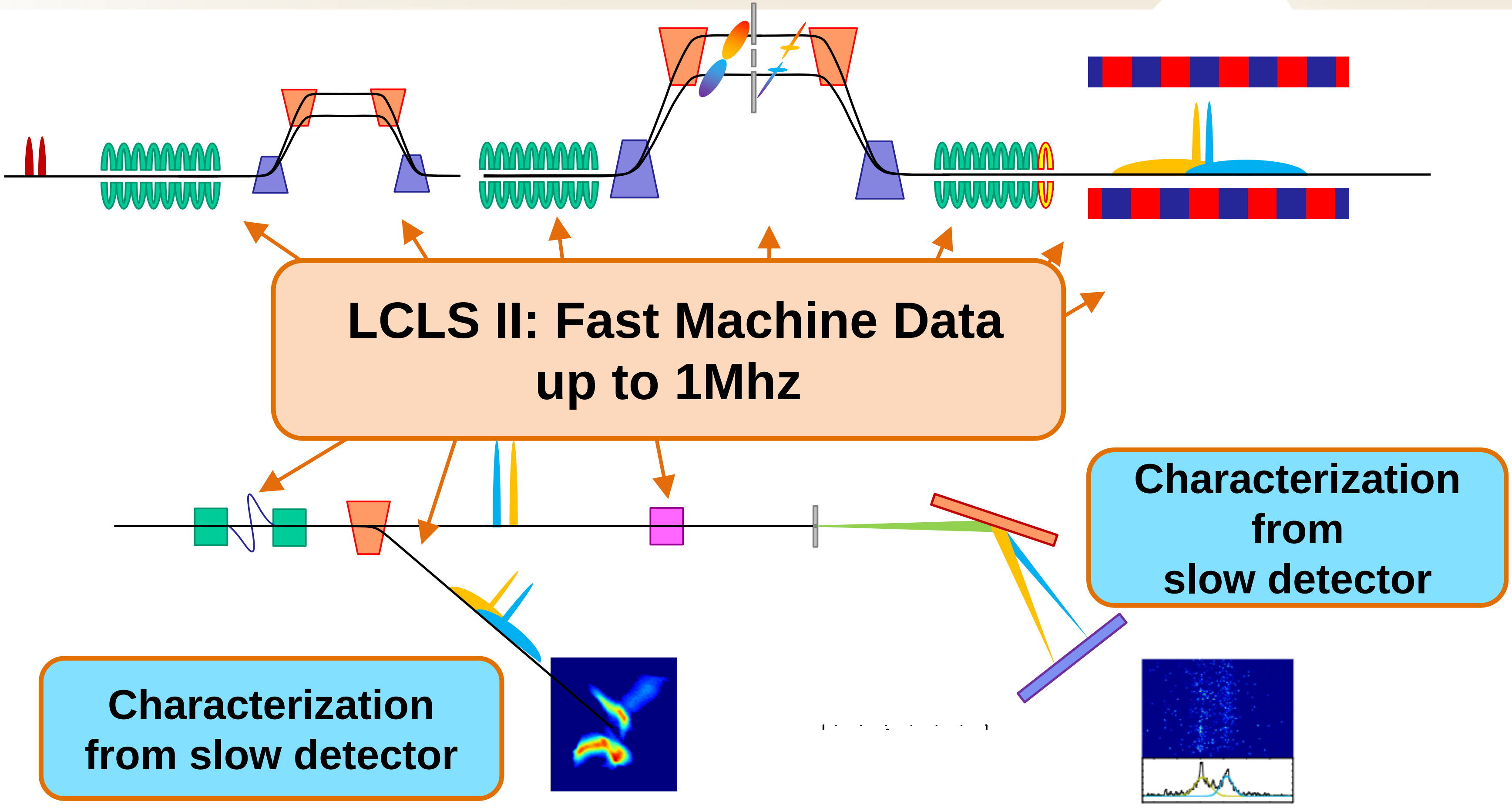
Regression Results:
Measurement of accuracy: for how many shots is area of intersection/union > 0.5?

VCC Screen – only ***0.09*** (left)
YAG Screen – ***0.89*** (not shown)

Critical step: de-noising preprocessing from Abdullah Rashed Ahmed

How to reconcile MHz beam and 120 Hz diagnostic?

SLAC

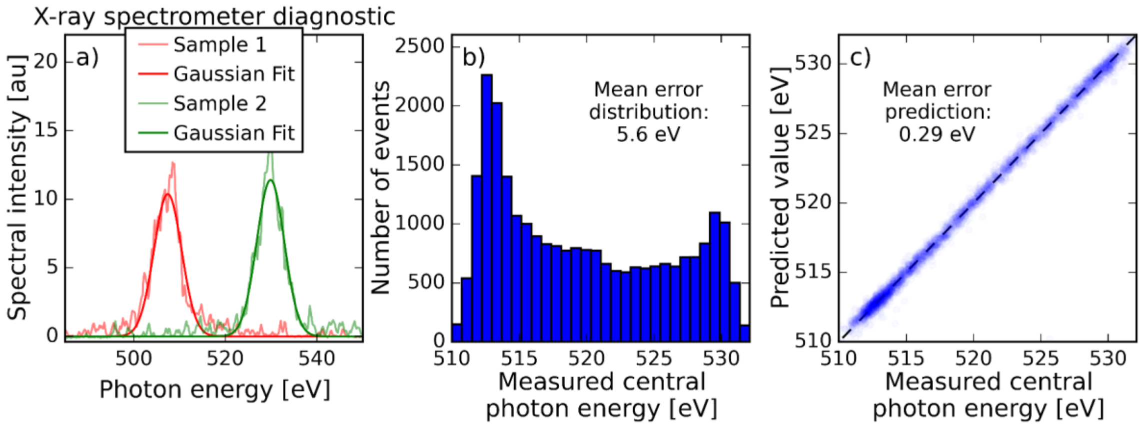


Predict/Interpolate slow characterization with Machine Learning

Imperial College London

A. Sanchez-Gonzalez, P. Micaelli

How to reconcile MHz beam and 120 Hz diagnostic?



Photon Energy

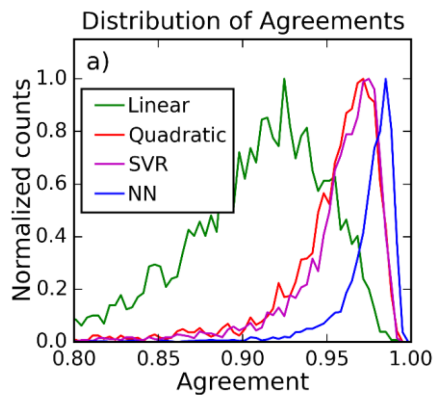
Prediction error

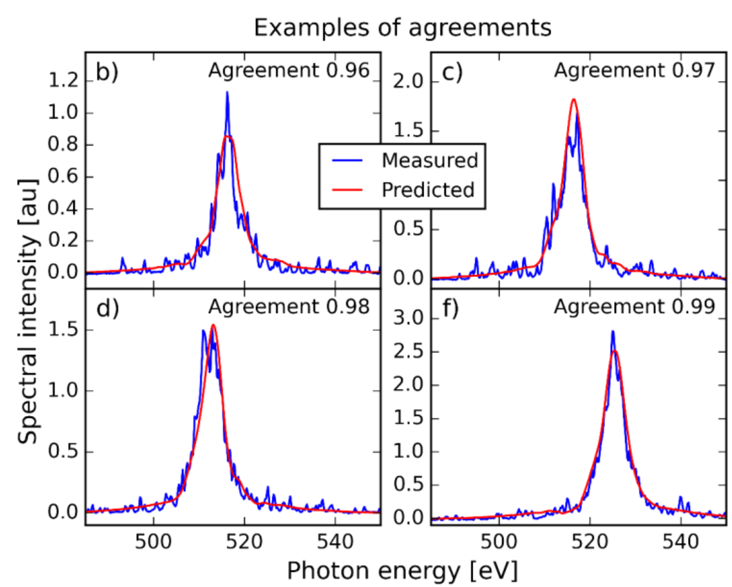
smaller than 0.3 eV

Spectral Shape

Agreements better

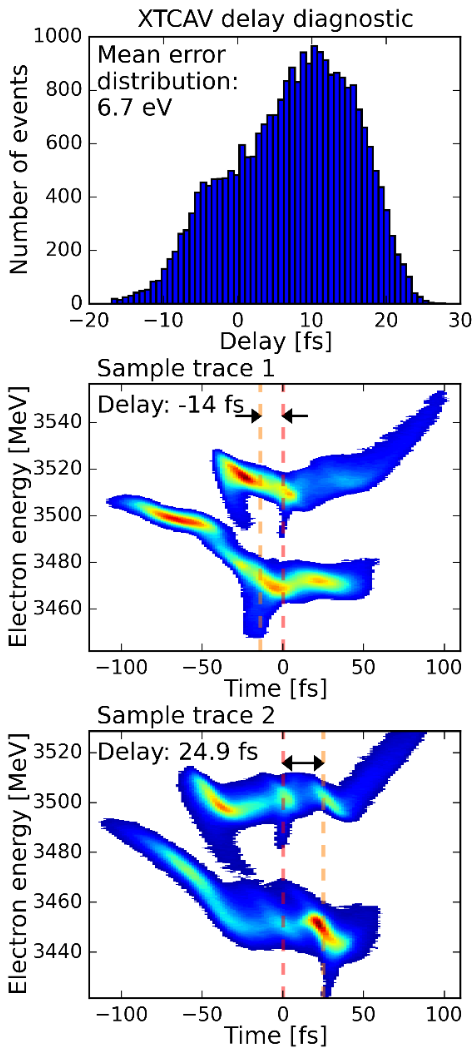
than 0.97%





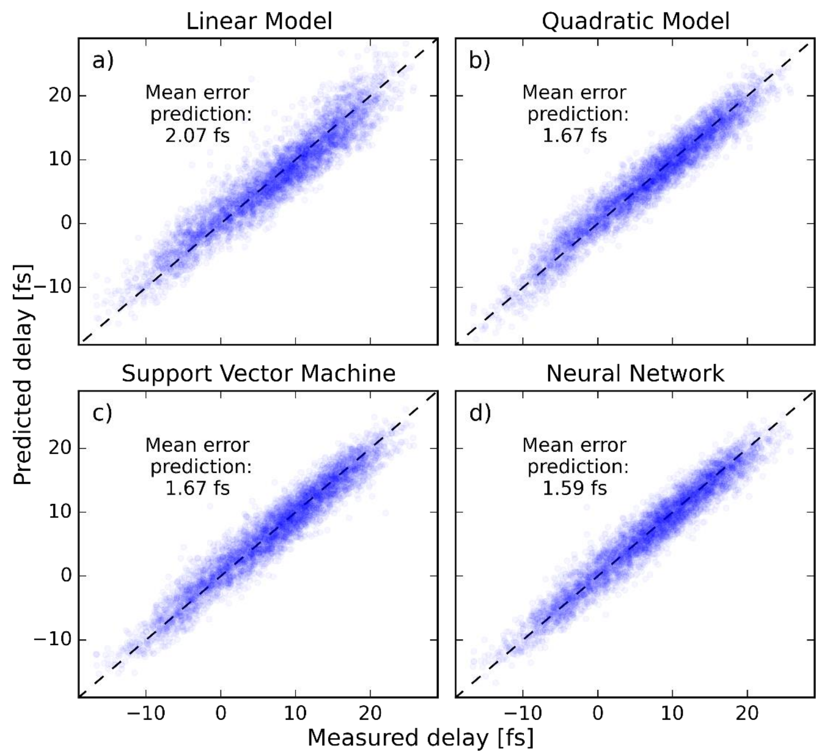
A. Sanchez-Gonzalez, P. Micaelli

How to reconcile MHz beam and 120 Hz diagnostic?



2-pulse delays

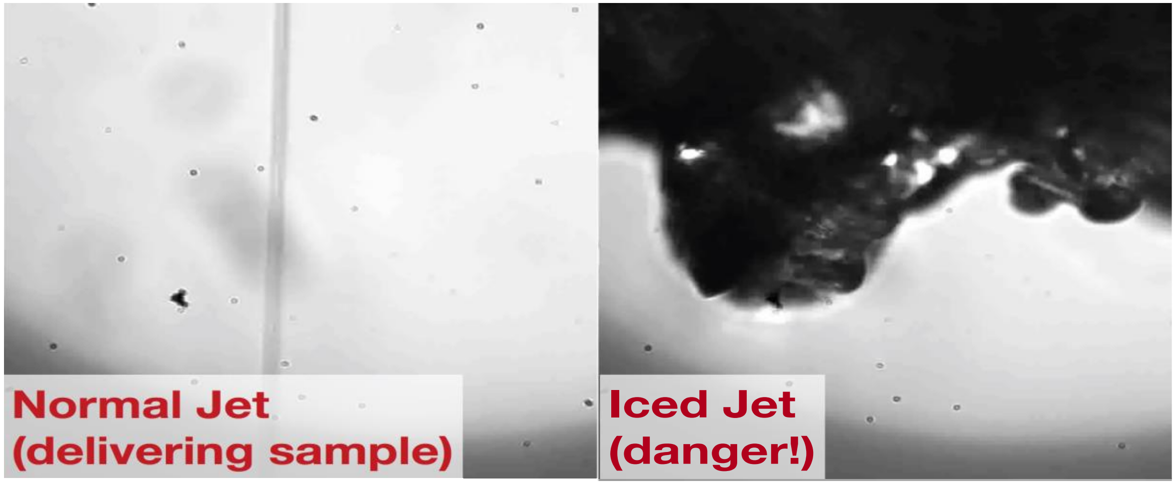
< 1.57 fs mean error



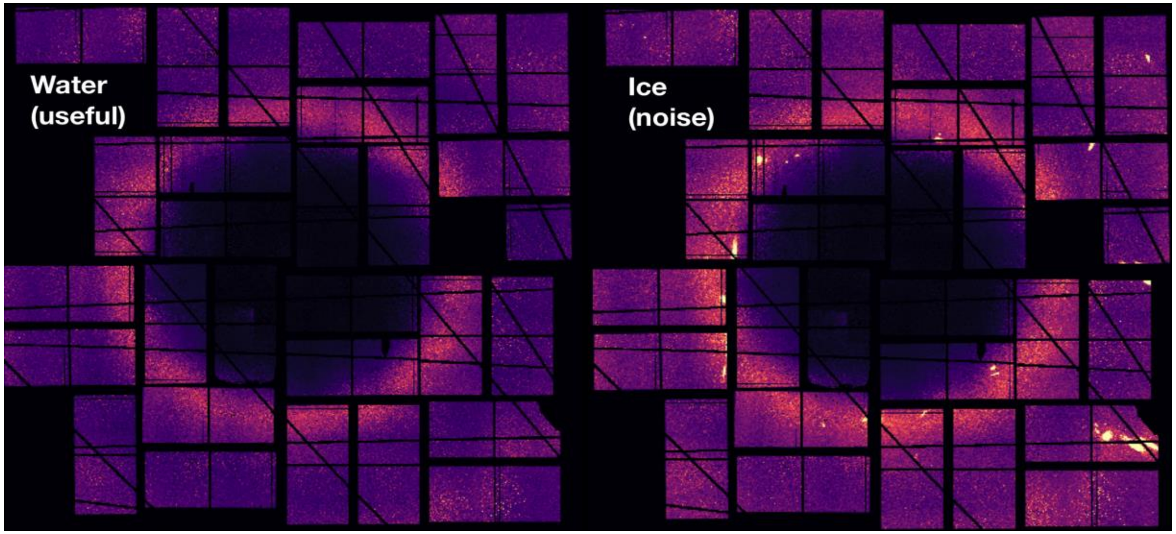
A. Sanchez-Gonzalez, P. Micaelli

Anomaly/Breakout detection (T.J. Lane)

Machine protection:
e.g. detecting ice to protect the detector



Data analysis:
e.g. sorting shots

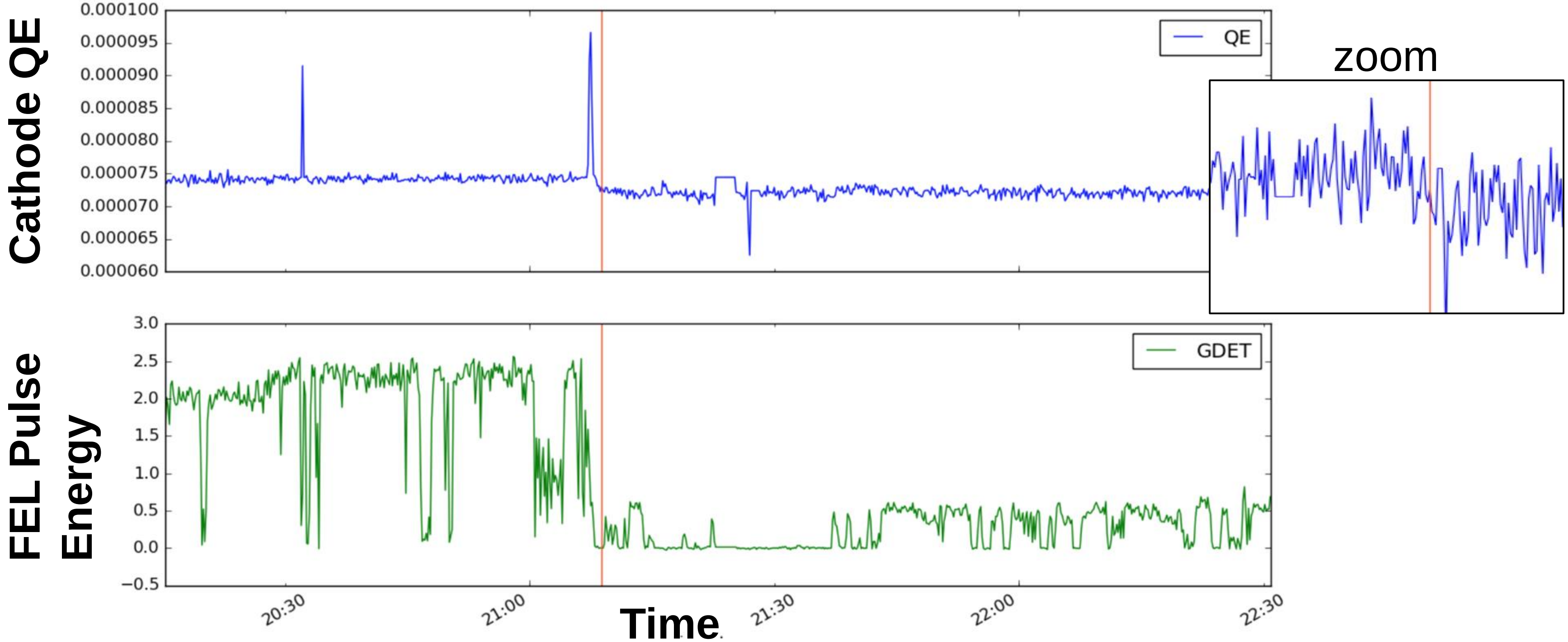


Anomaly/Breakout Detection

Can we detect if something is broken or about to break?

- 200,000 PVs: no human can keep an eye on all of them
- Signals are complex: simple thresholds cannot work

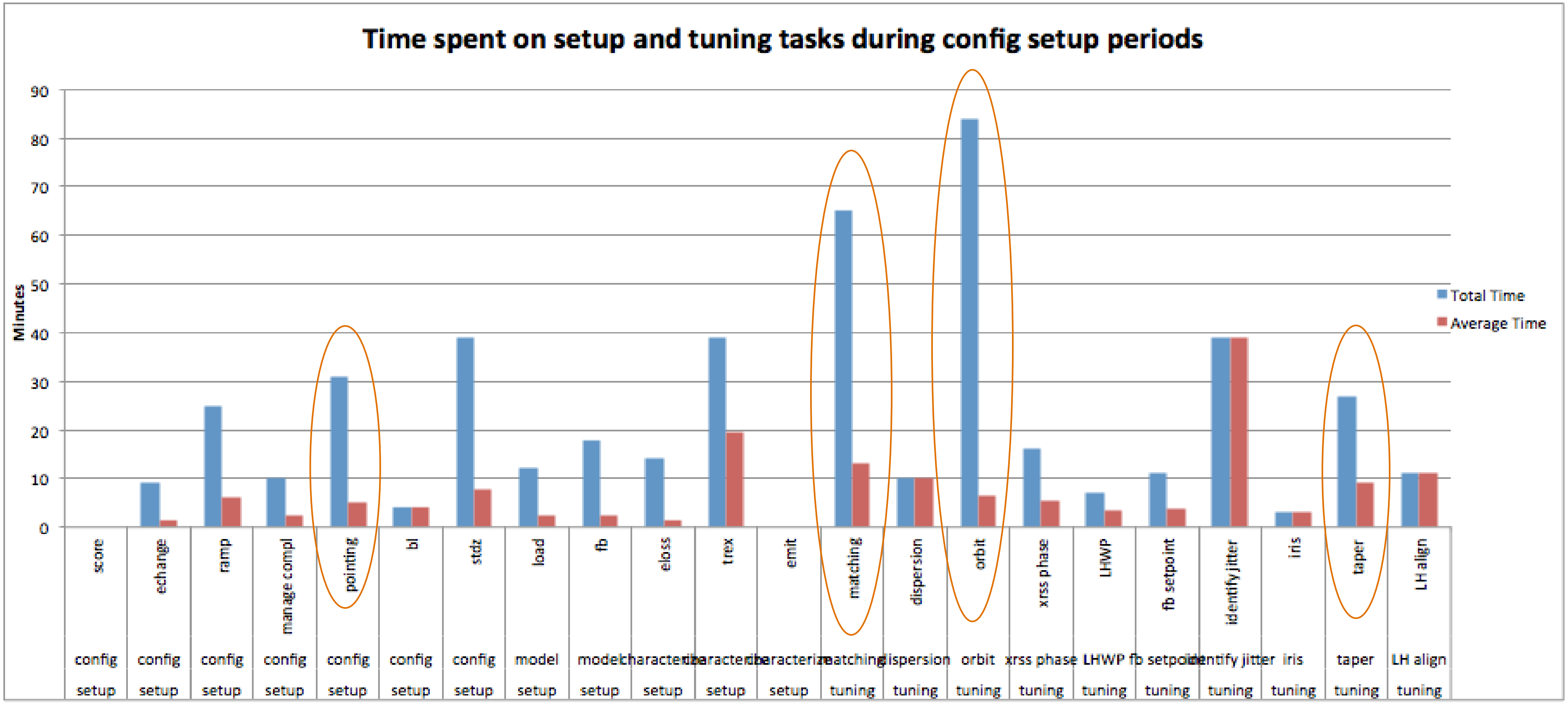
Cathode QE drop caused hours of downtime. Breakout detection would have found change immediately!



M. Gibbs, N. Norvell, D. Sanzone

Machine Tuning Automation

How to optimize 2km long machine?

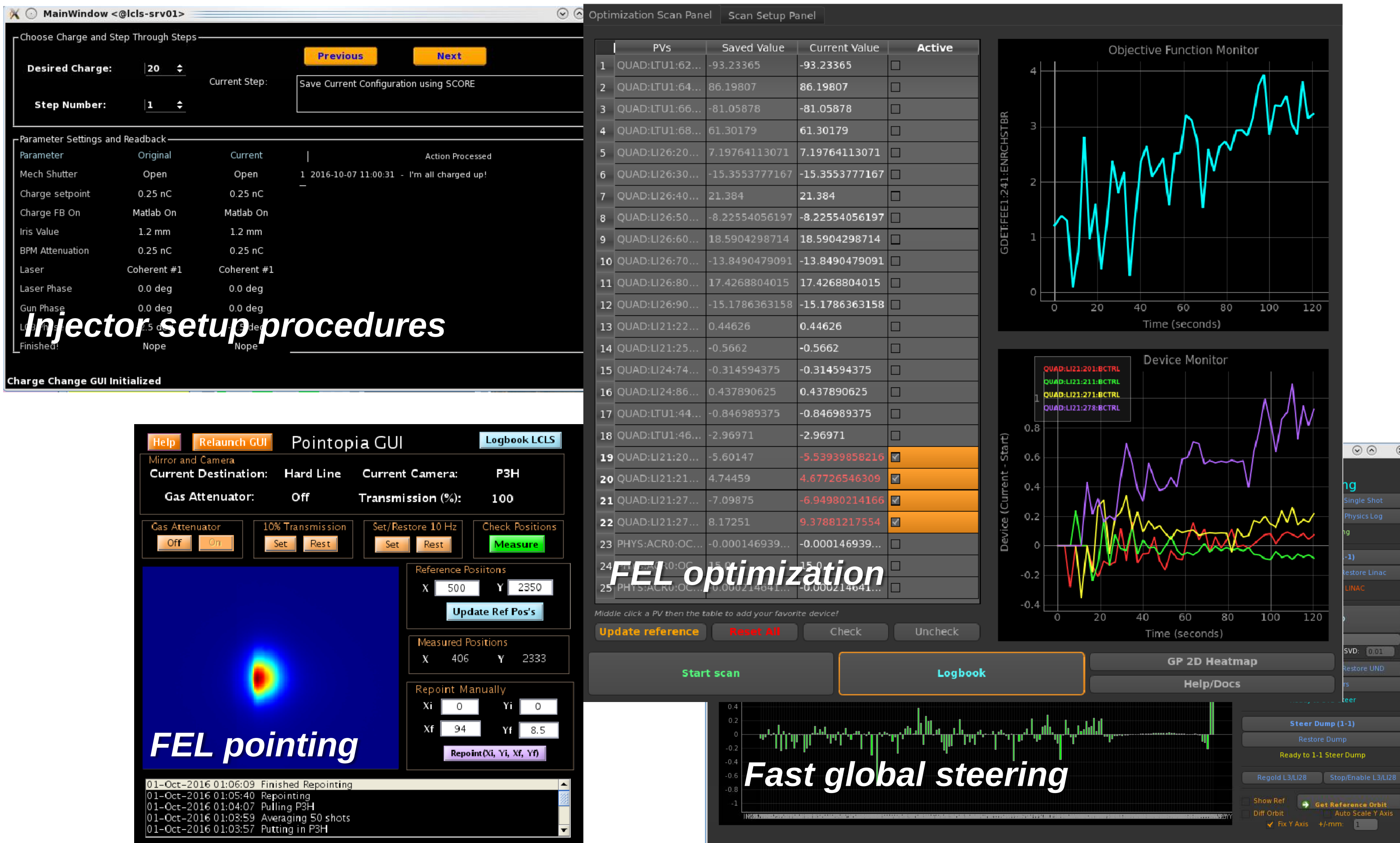


2015: 450 hand tuning hours, 250 dedicated!

⇒ Lots of opportunity to speed operations and relieve operator load

Machine Tuning Automation

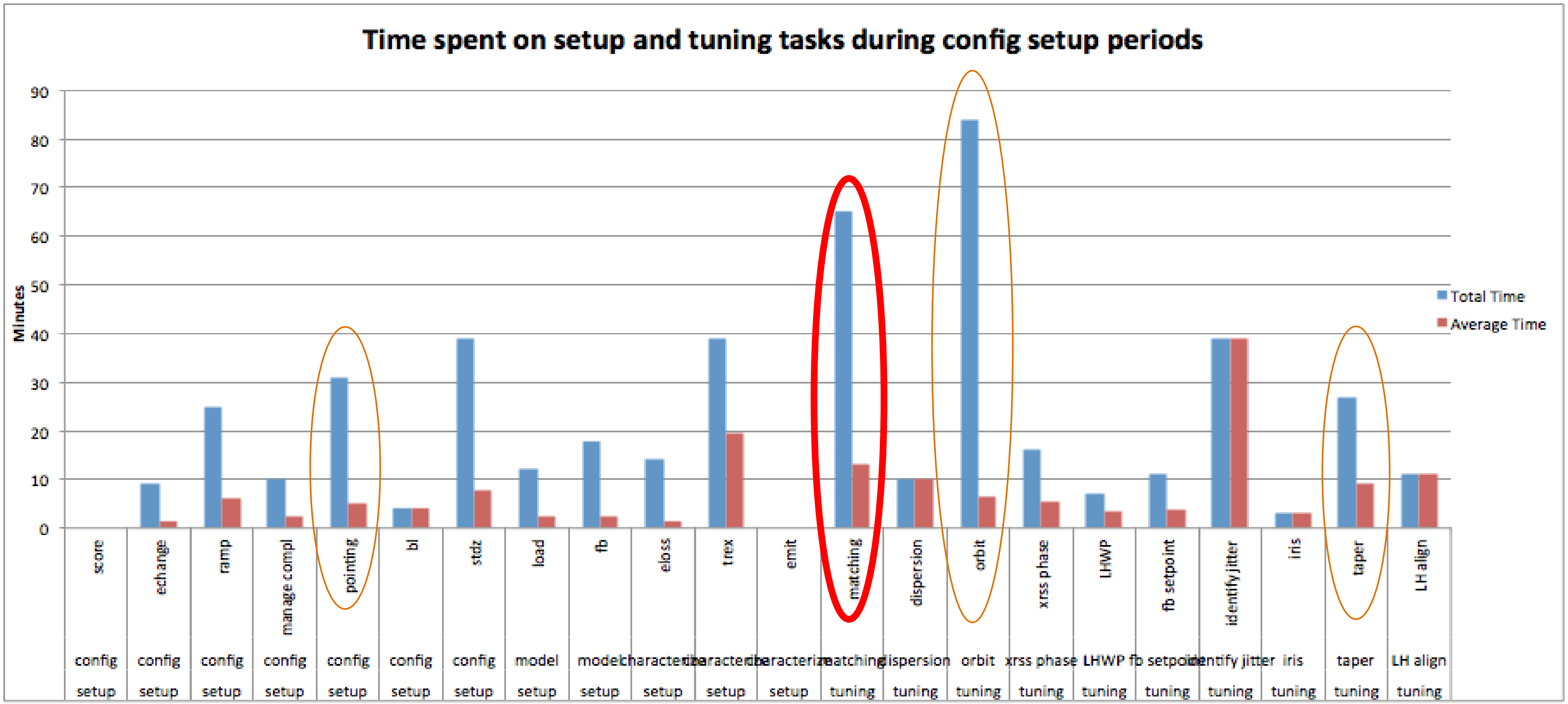
Working with AOSD – Faster tuning, fewer errors



Machine Tuning Automation

SLAC

How to optimize 2km long machine?

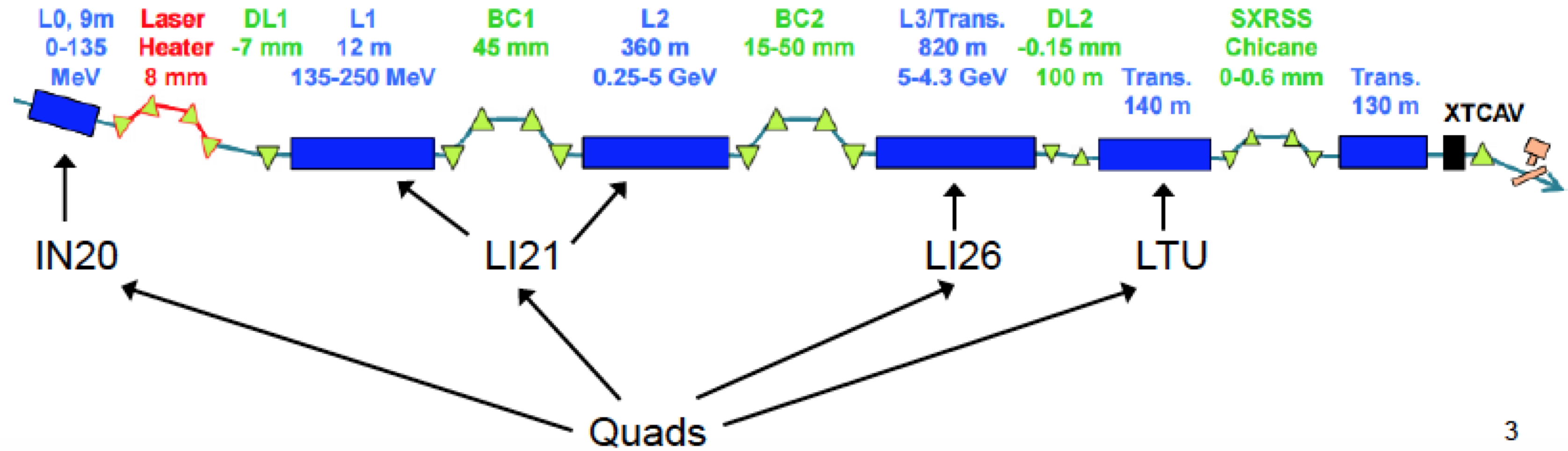


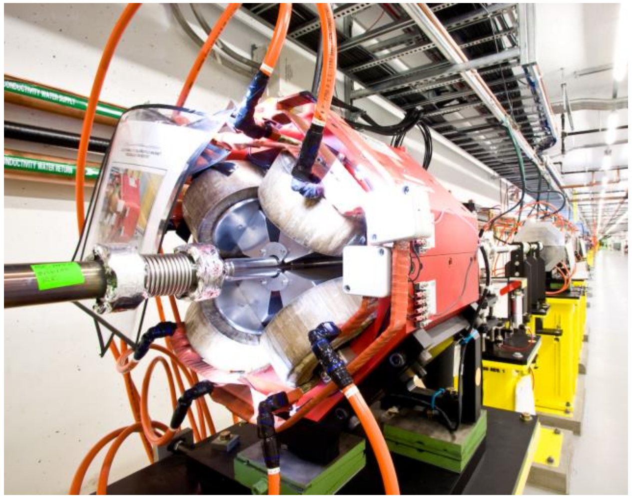
2015: 450 hand tuning hours, 250 dedicated!

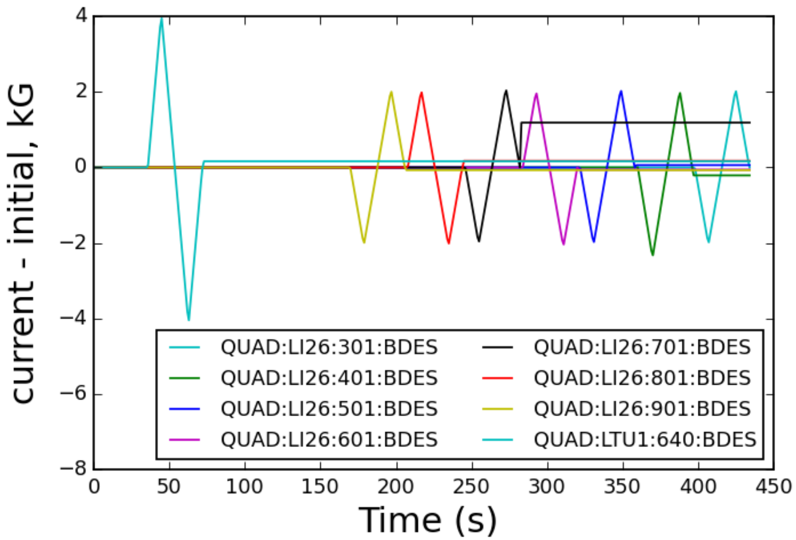
⇒ Lots of opportunity to speed operations and relieve operator load

SLAC

Online optimization of quadrupole magnets





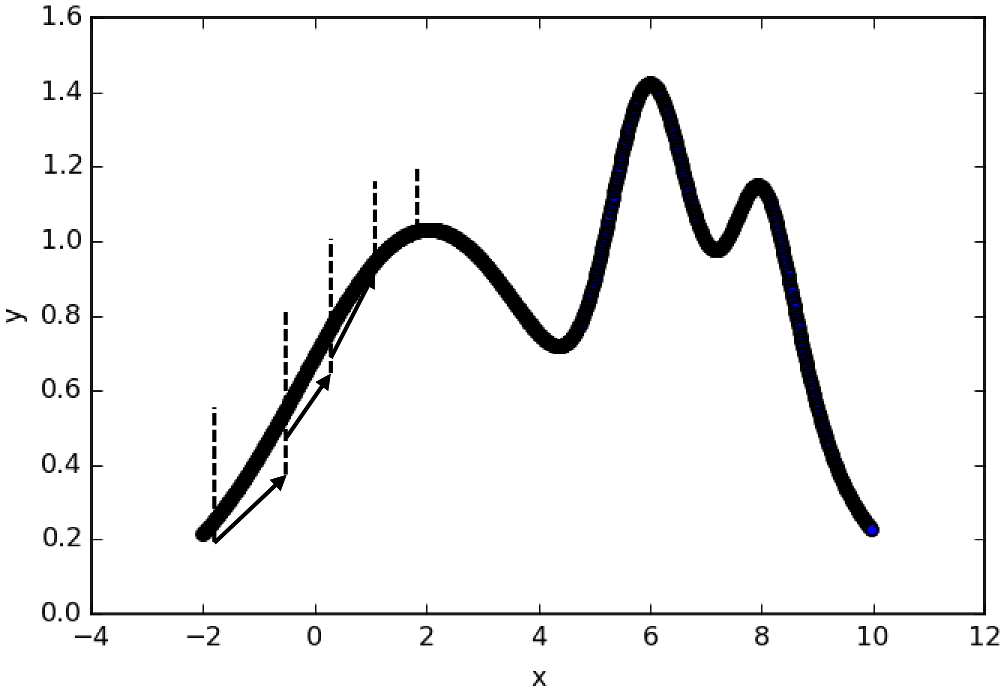


Online optimization

Tried several optimization approaches:

→ Gradient/simplex methods

(Nelder-Mead in general use)

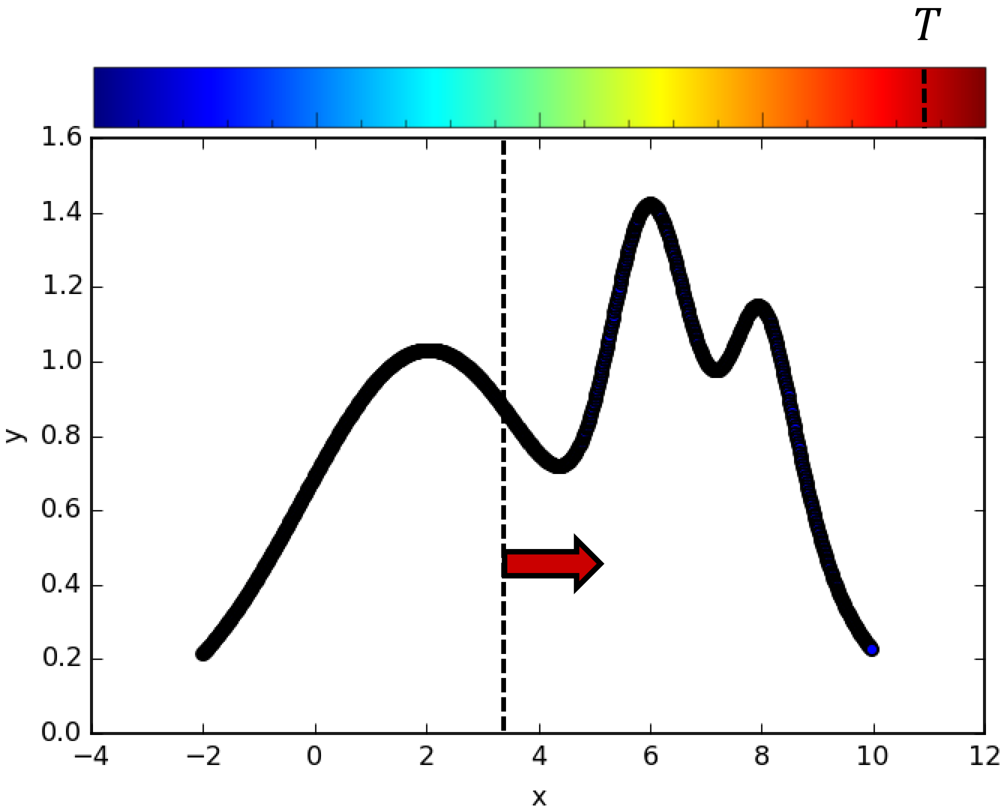


Slide from M. McIntire

Online optimization

Still many optimizers to try:

→ Simulated annealing, genetic algorithms, etc.



Bayesian optimization

Bayesian approach: introduce probabilistic model

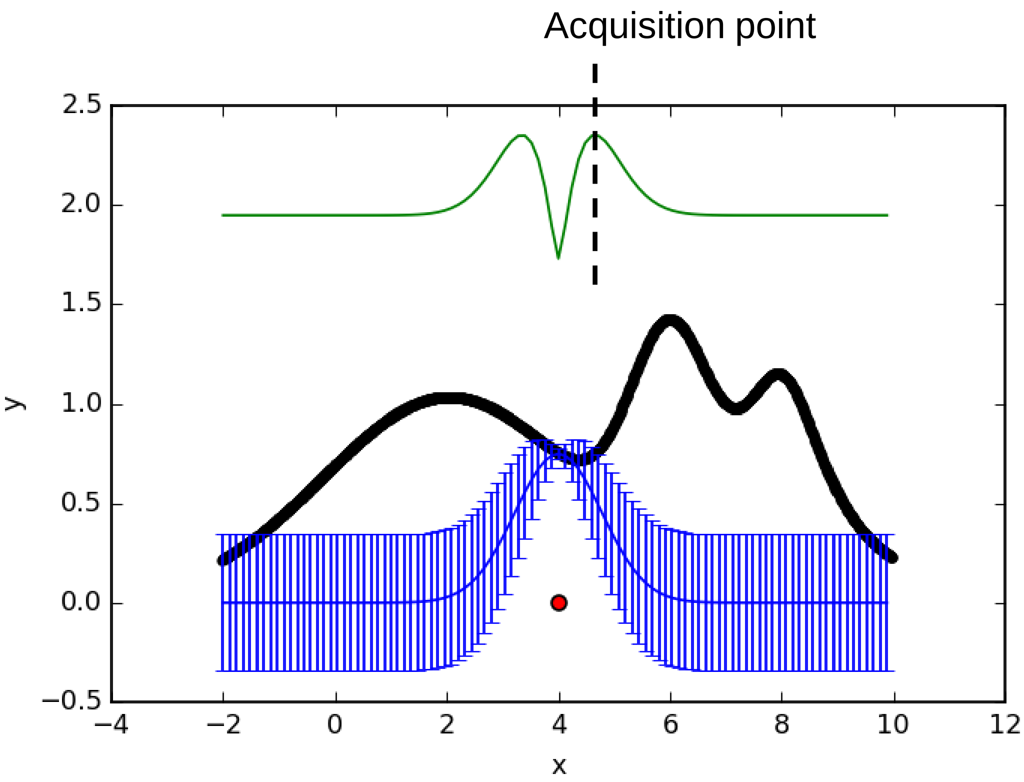
→ create acquisition function

→ more efficient search of high dimensional space

Acquisition function

Ground truth

Posterior



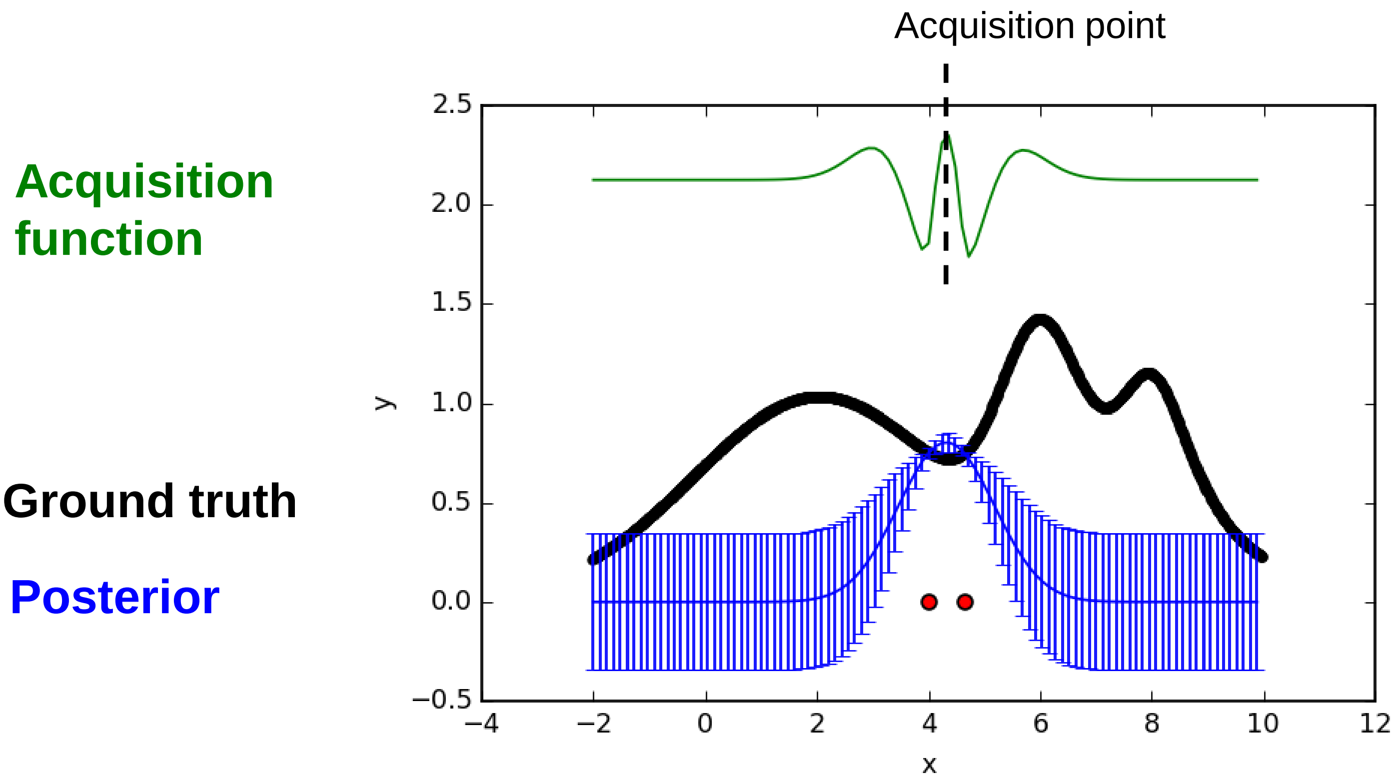
Slide from M. McIntire

Bayesian optimization

Add probabilistic model

→ create acquisition function

→ more efficient search of high dimensional space



Slide from M. McIntire

Bayesian optimization

Add probabilistic model

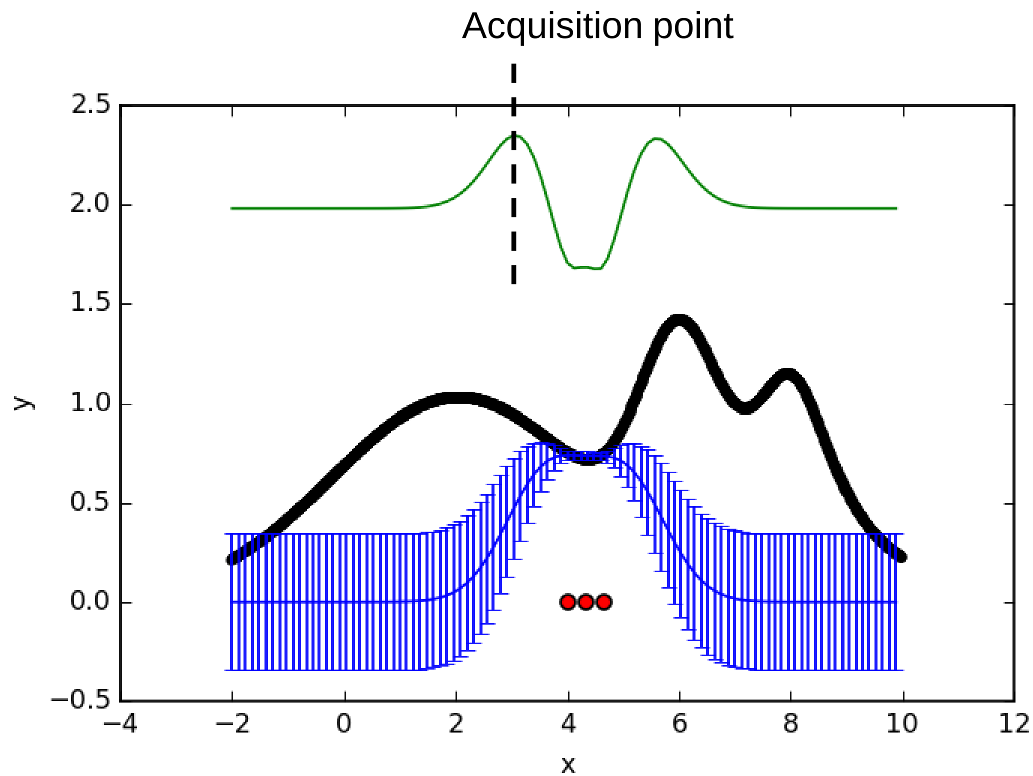
→ create acquisition function

→ more efficient search of high dimensional space

Acquisition function

Ground truth

Posterior



Slide from M. McIntire

Bayesian optimization

Add probabilistic model

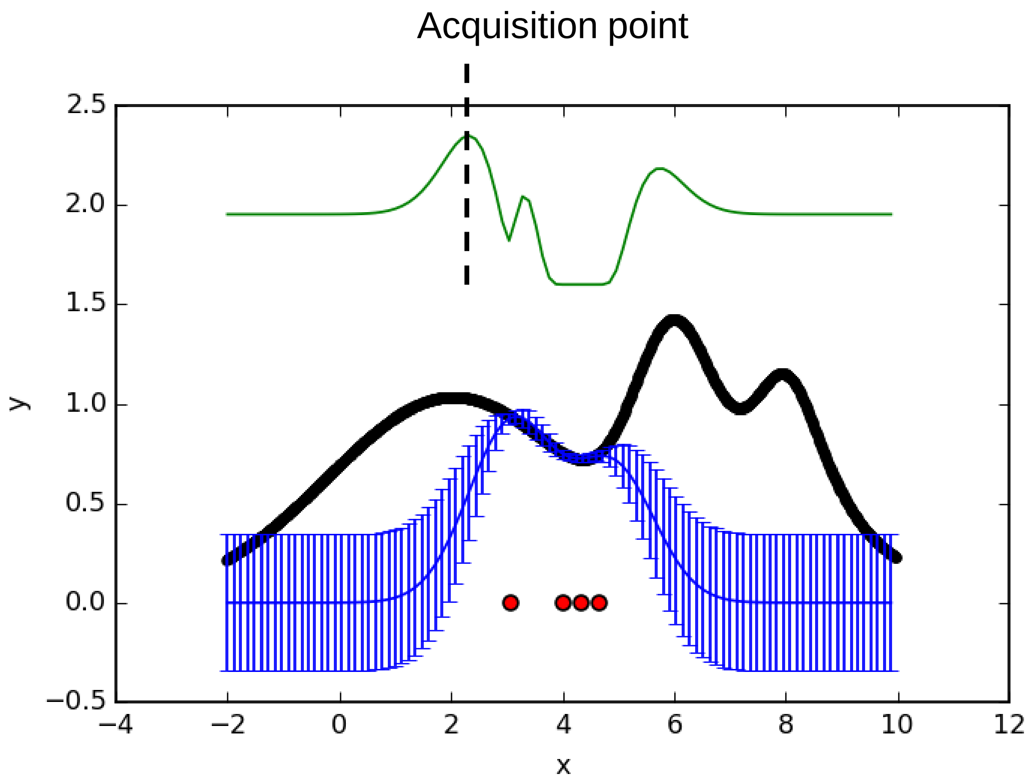
→ create acquisition function

→ more efficient search of high dimensional space

Acquisition function

Ground truth

Posterior



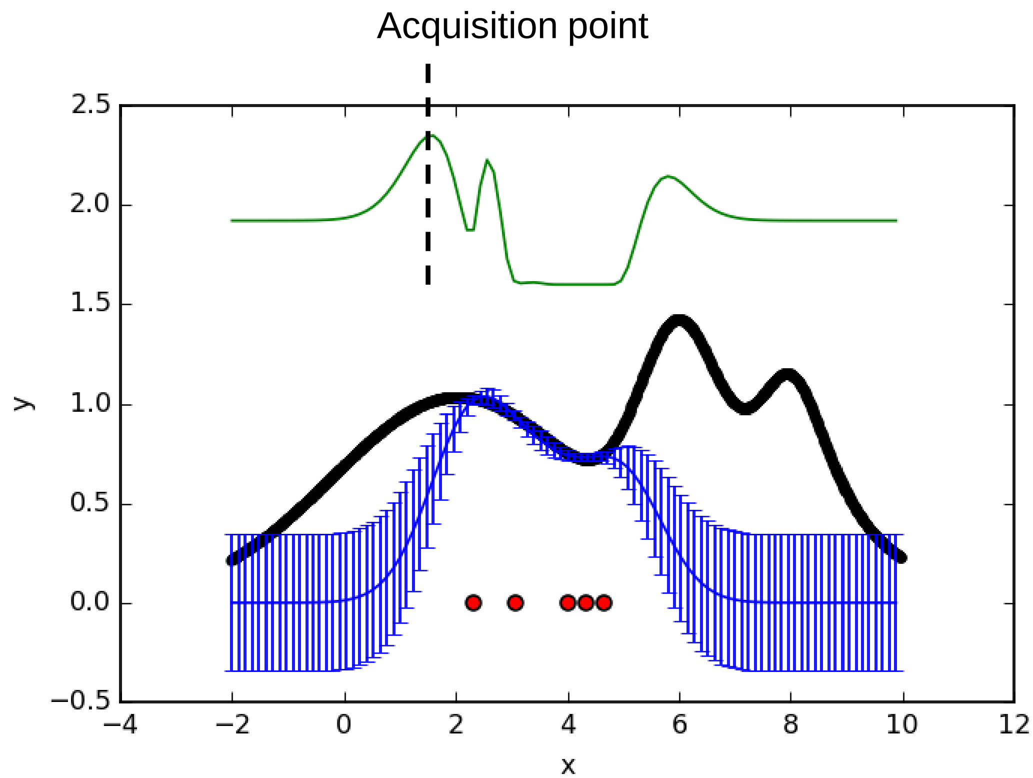
Slide from M. McIntire

Bayesian optimization

Add probabilistic model

→ create acquisition function

→ more efficient search of high dimensional space



Slide from M. McIntire

Bayesian optimization

Add probabilistic model

→ create acquisition function

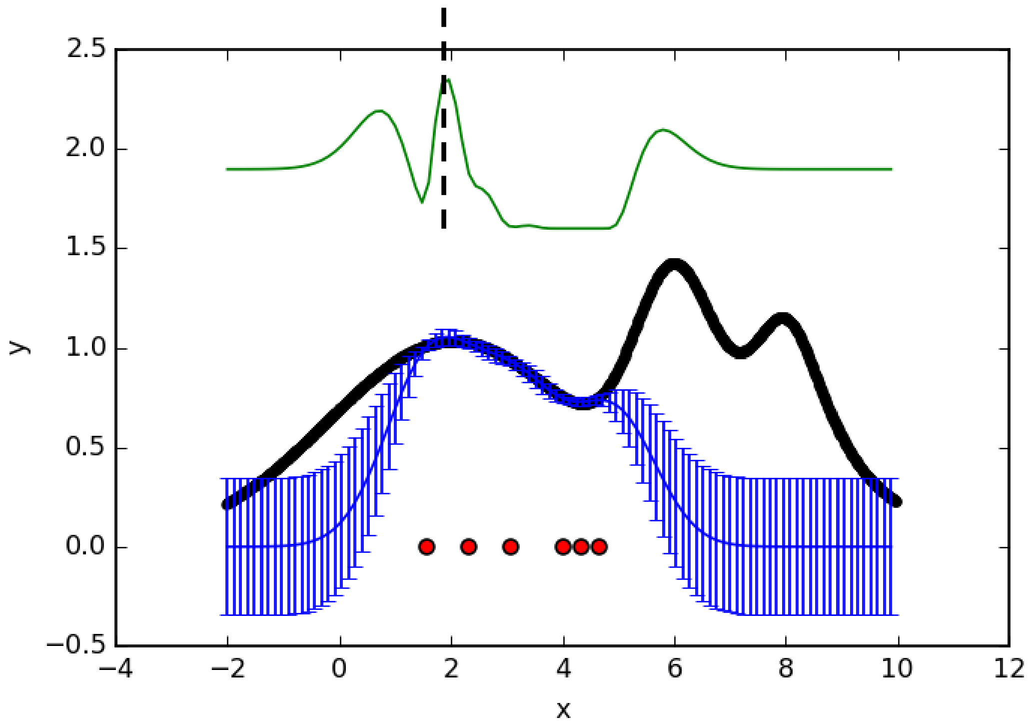
→ more efficient search of high dimensional space

Acquisition function

Ground truth

Posterior

Acquisition point



Slide from M. McIntire

Bayesian optimization

Add probabilistic model

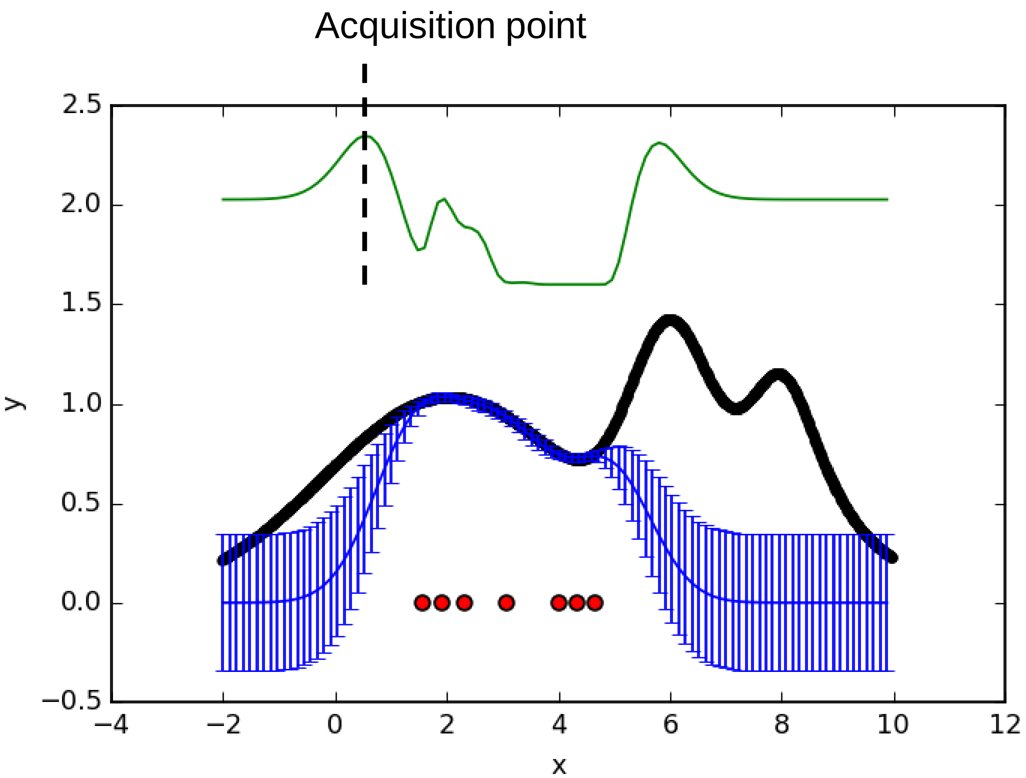
→ create acquisition function

→ more efficient search of high dimensional space

Acquisition function

Ground truth

Posterior



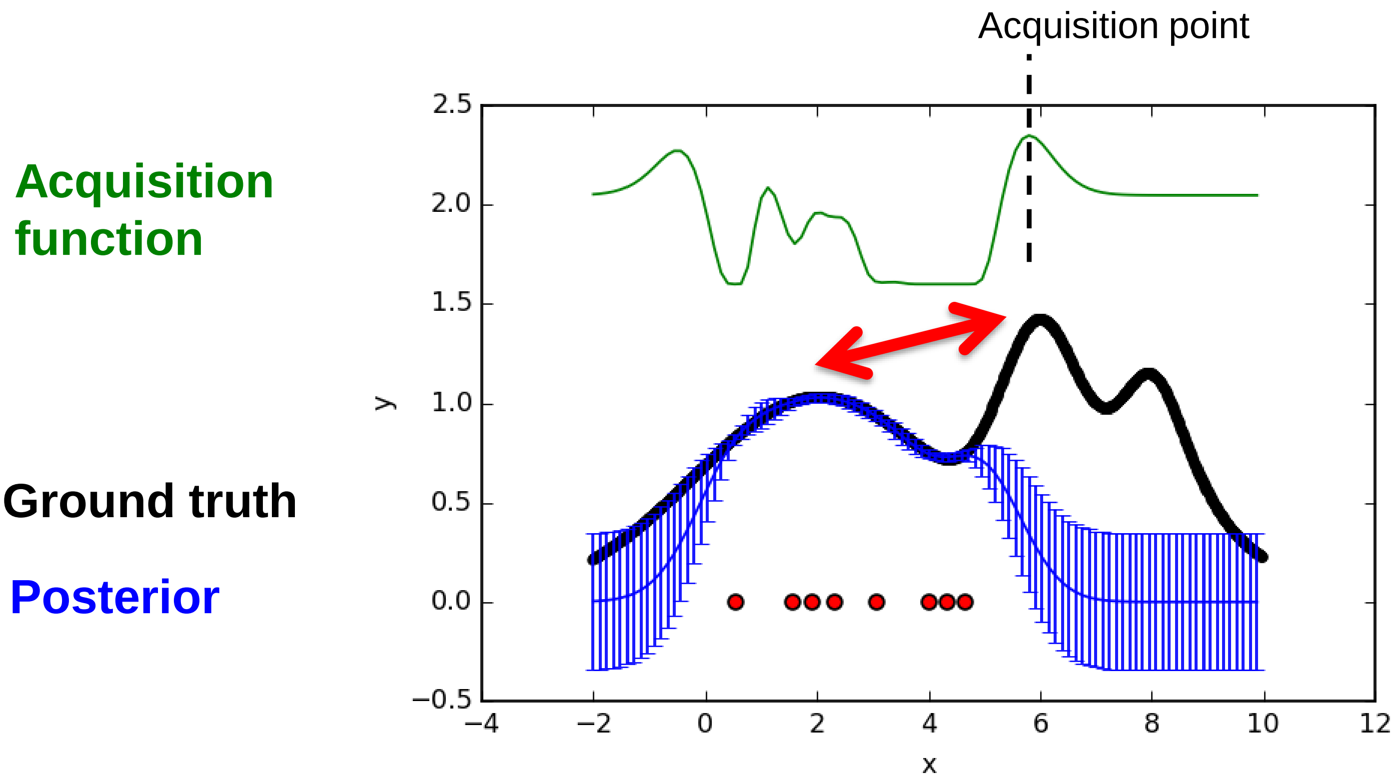
Slide from M. McIntire

Bayesian optimization

Add probabilistic model

→ create acquisition function

→ more efficient search of high dimensional space



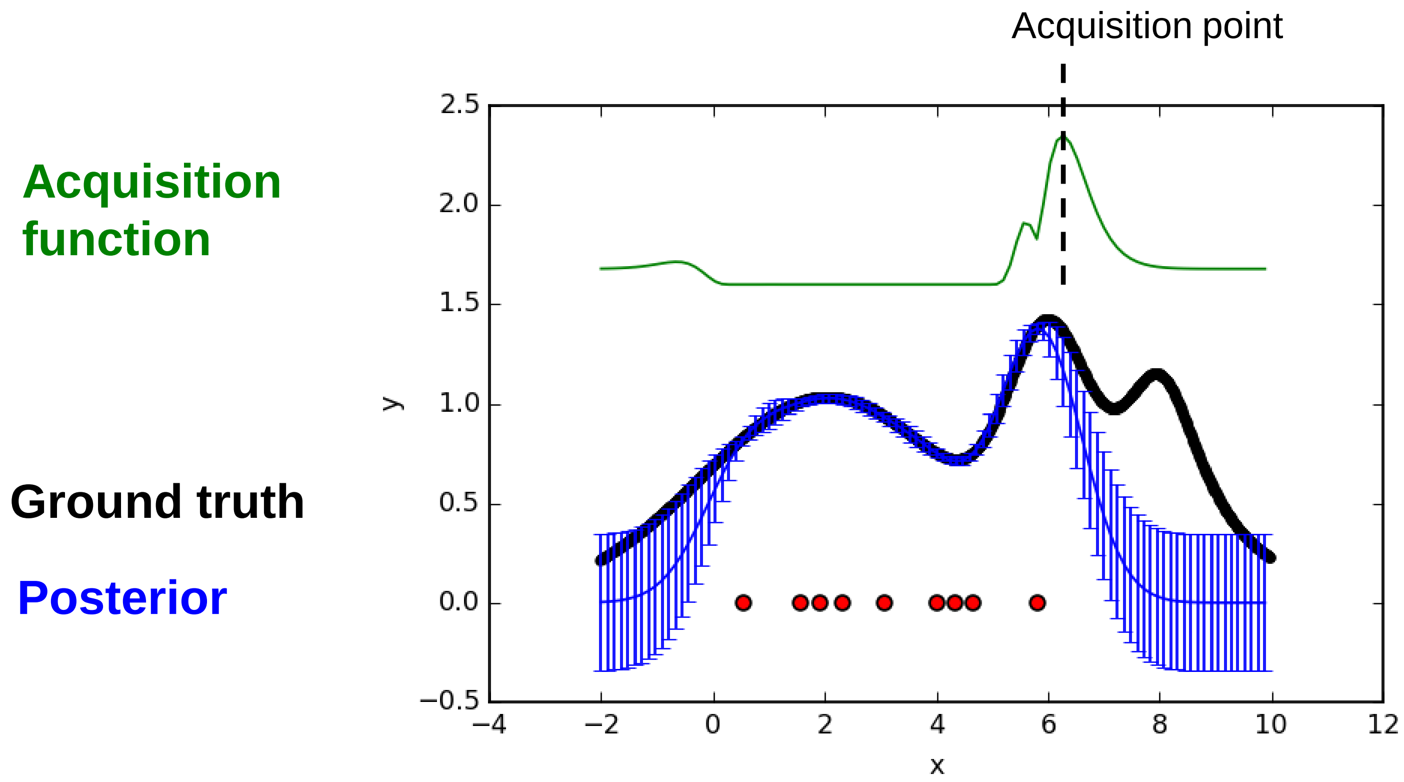
Slide from M. McIntire

Bayesian optimization

Add probabilistic model

→ create acquisition function

→ more efficient search of high dimensional space



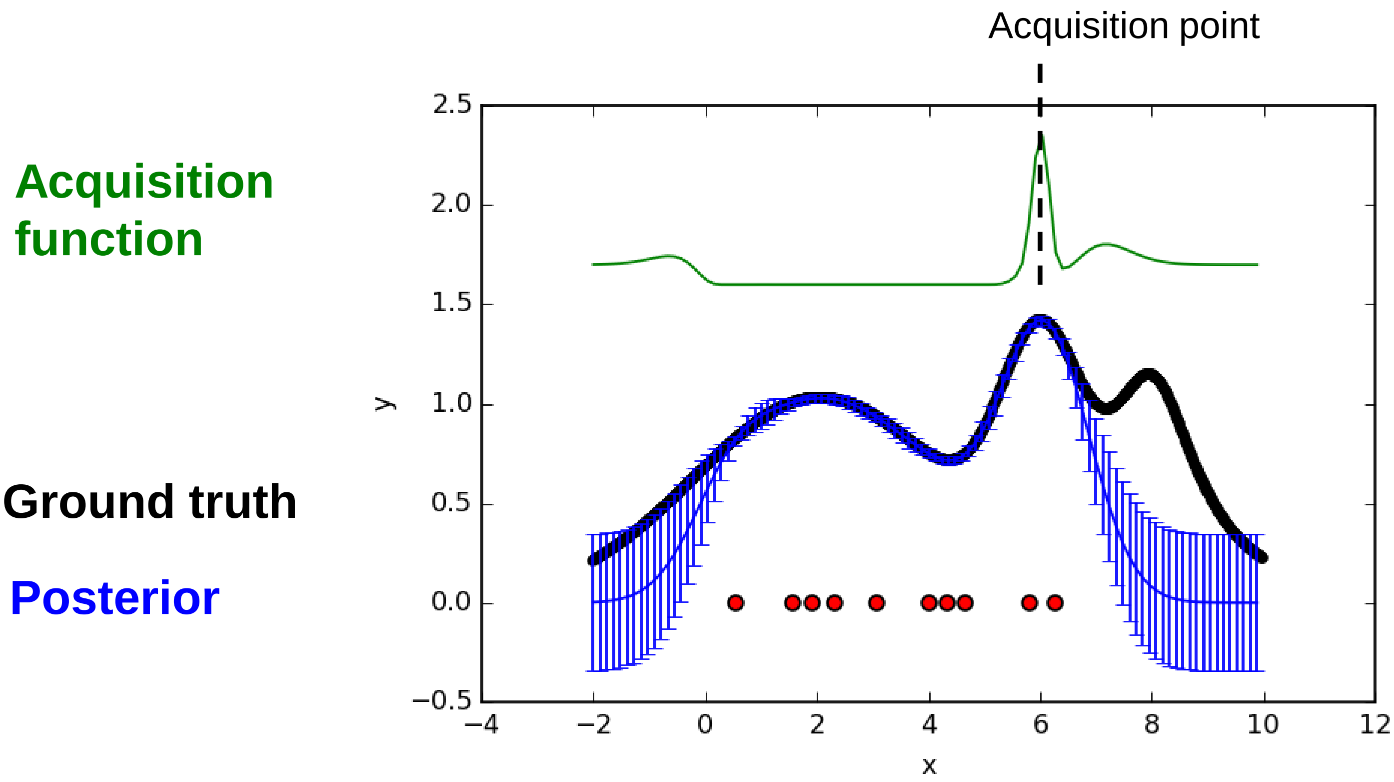
Slide from M. McIntire

Bayesian optimization

Add probabilistic model

→ create acquisition function

→ more efficient search of high dimensional space

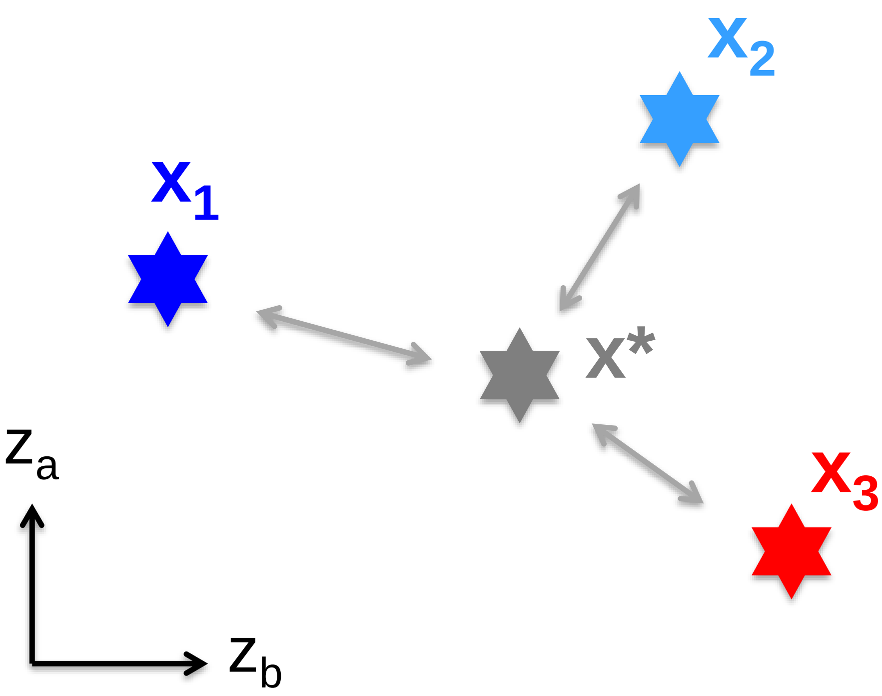


Slide from M. McIntire

Gaussian Process Optimizer

Gaussian process: instance based learning method

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



$$y^* = f(x^*) \\ = f(z_a, z_b, z_a{*}z_b, \ldots)$$

Gaussian Process Optimizer

Gaussian process: instance based learning method

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

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

Gaussian Process Optimizer

Gaussian process: instance based learning method

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

observations

new point to predict

prior mean

new point

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

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

Gaussian Process Optimizer

Gaussian process: instance based learning method

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

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: $EI(x^*) = \int_{\tilde{y}}^{\infty} (y^* - \tilde{y}) P(y^* | x^*) dy^*$

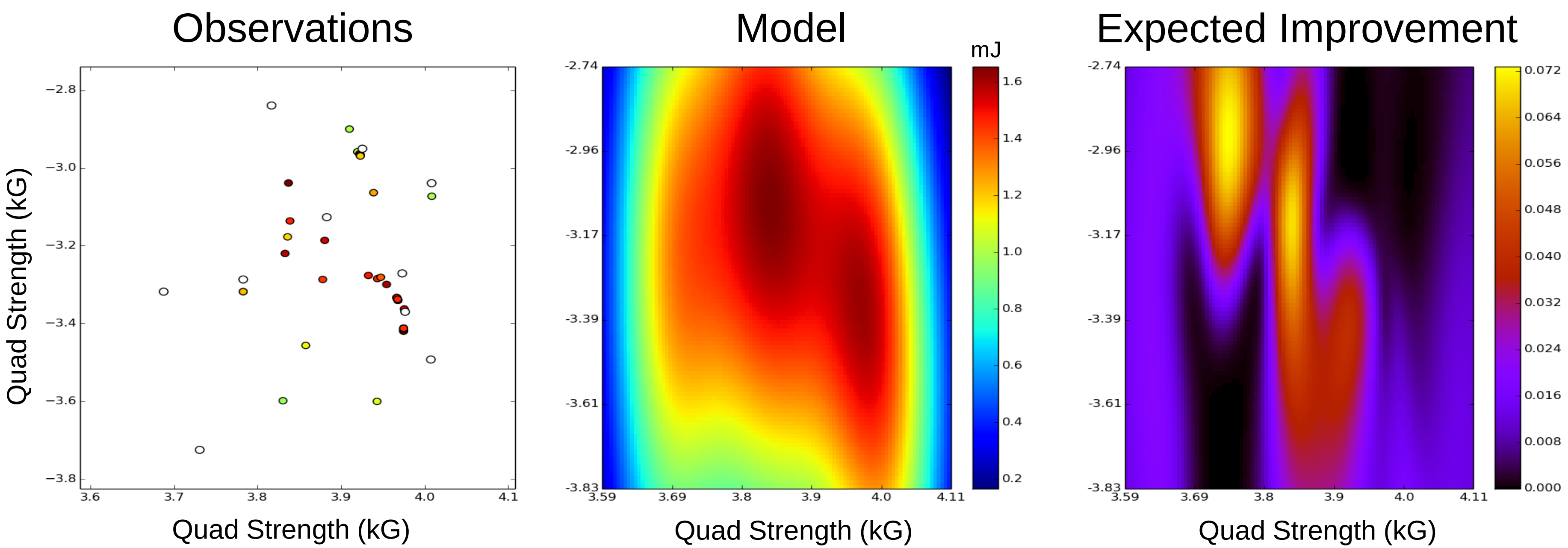
best observed point

Gaussian Process Optimizer

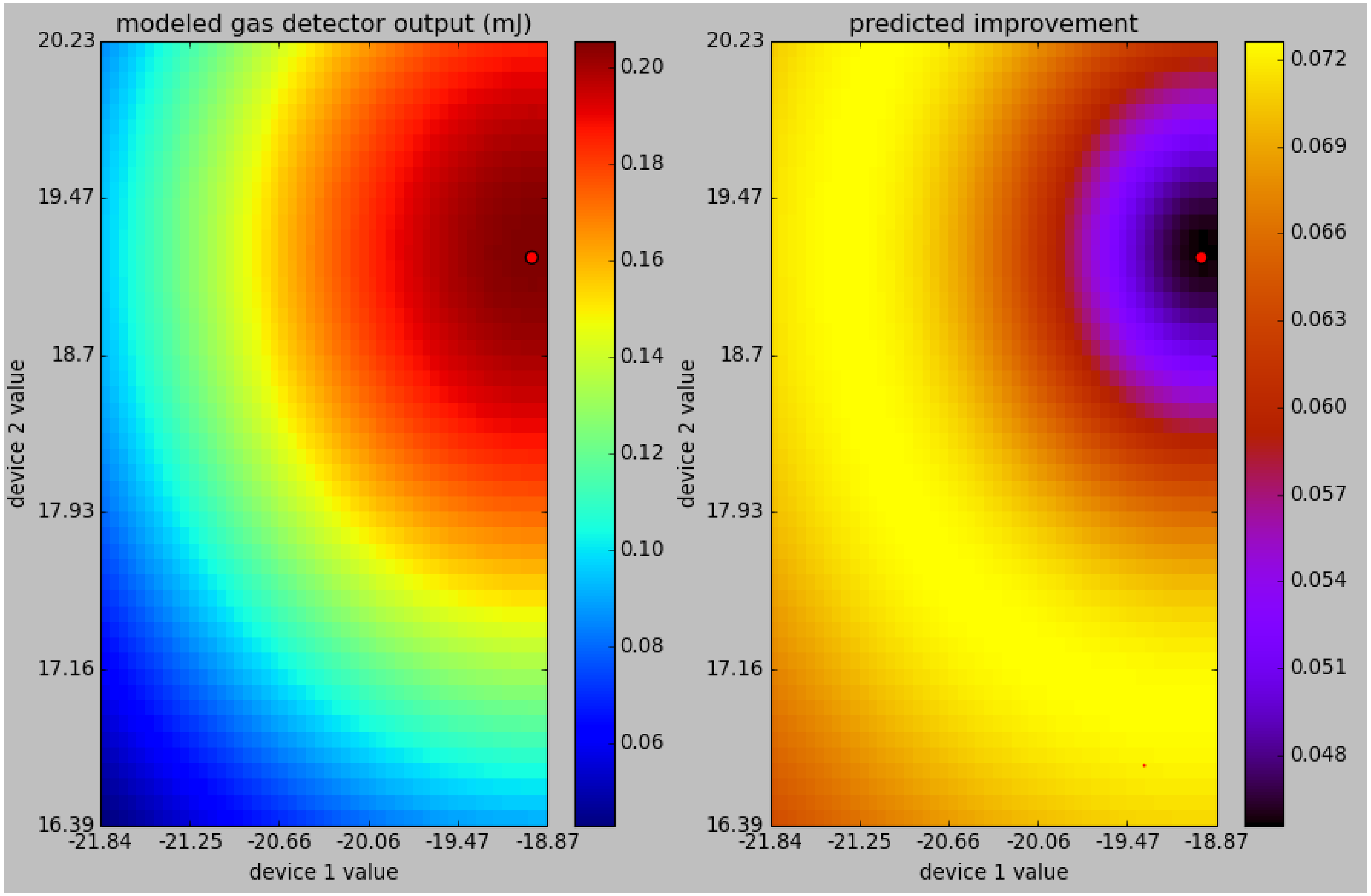
Gaussian process: instance based learning method

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

Acquisition function: $EI(x^*) = \int_{\tilde{y}}^{\infty} (y^* - \tilde{y}) P(y^* | x^*) dy^*$



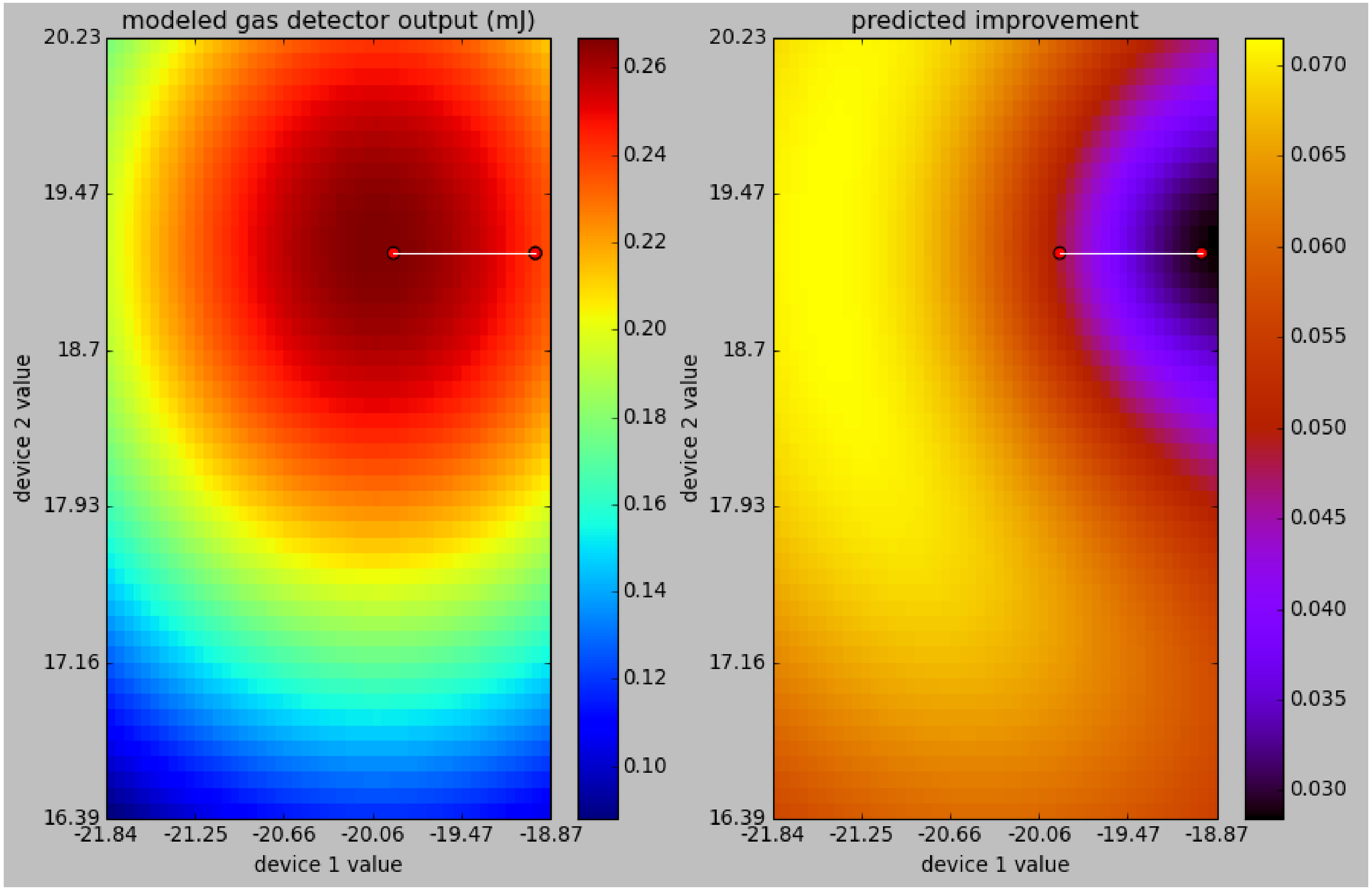
Point 2



OcelotScan-2016-09-21-185122.mat

J. Duris

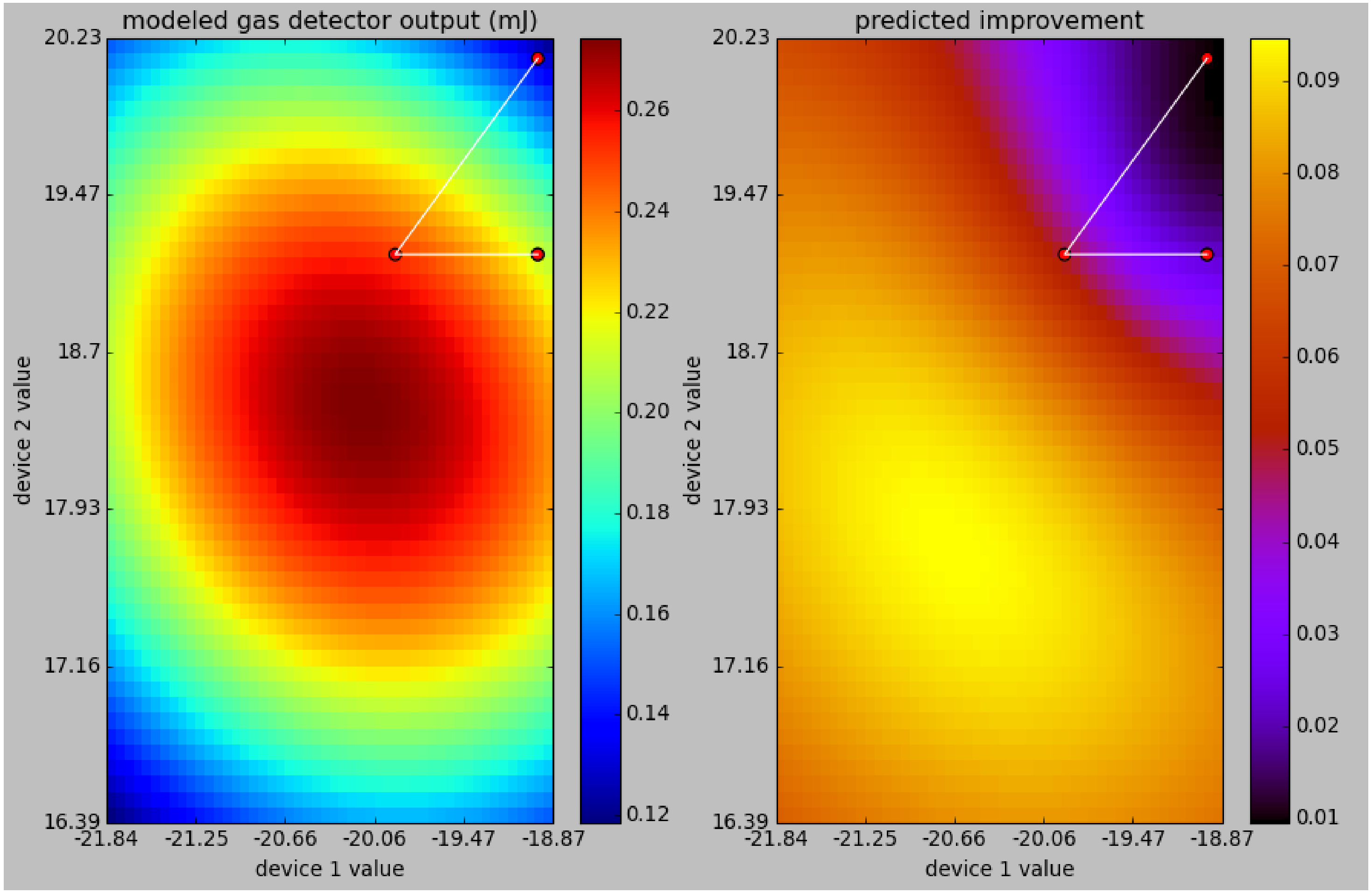
Point 3



OcelotScan-2016-09-21-185122.mat

J. Duris

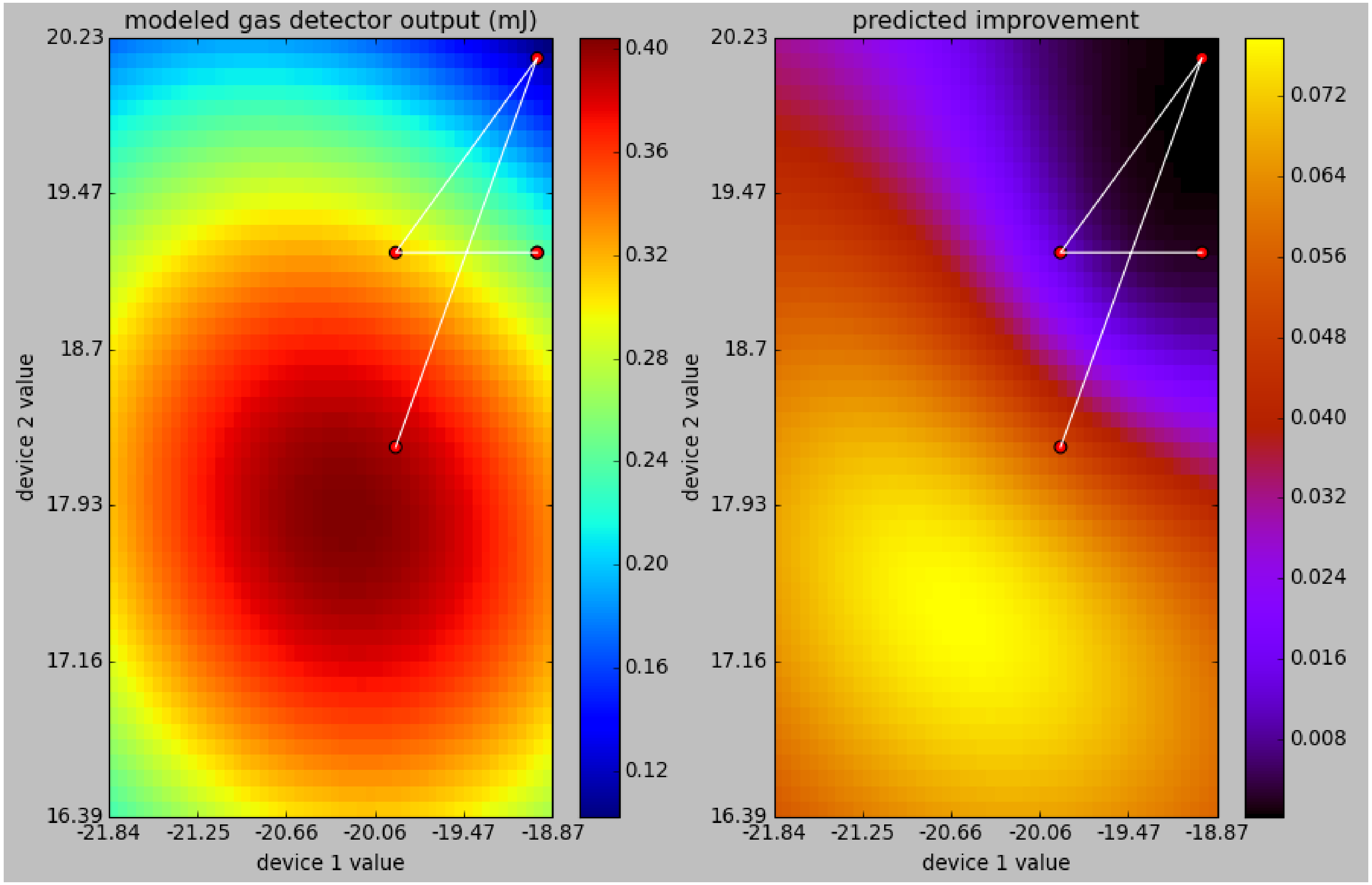
Point 4



OcelotScan-2016-09-21-185122.mat

J. Duris

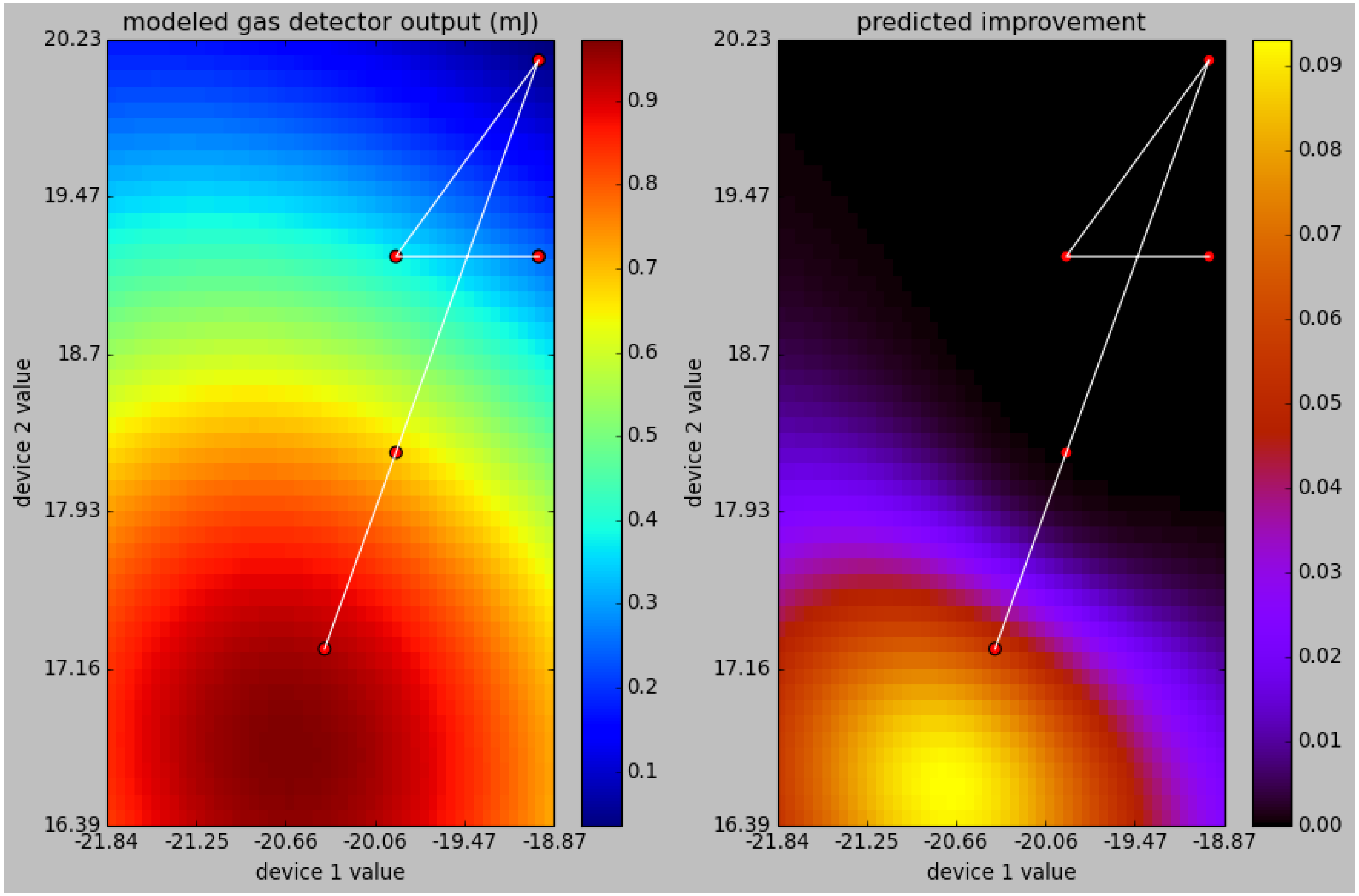
Point 5



OcelotScan-2016-09-21-185122.mat

J. Duris

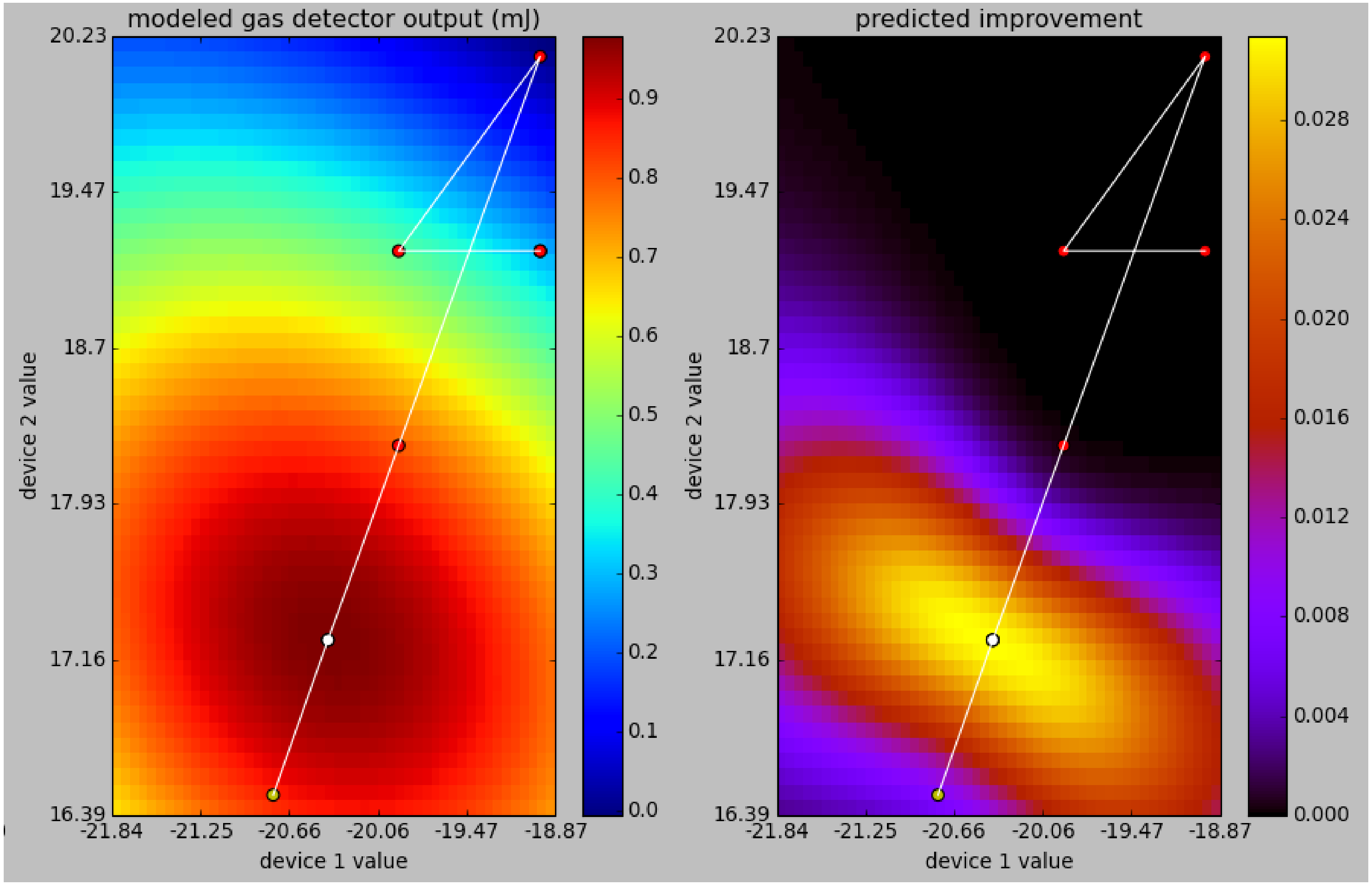
Point 6



OcelotScan-2016-09-21-185122.mat

J. Duris

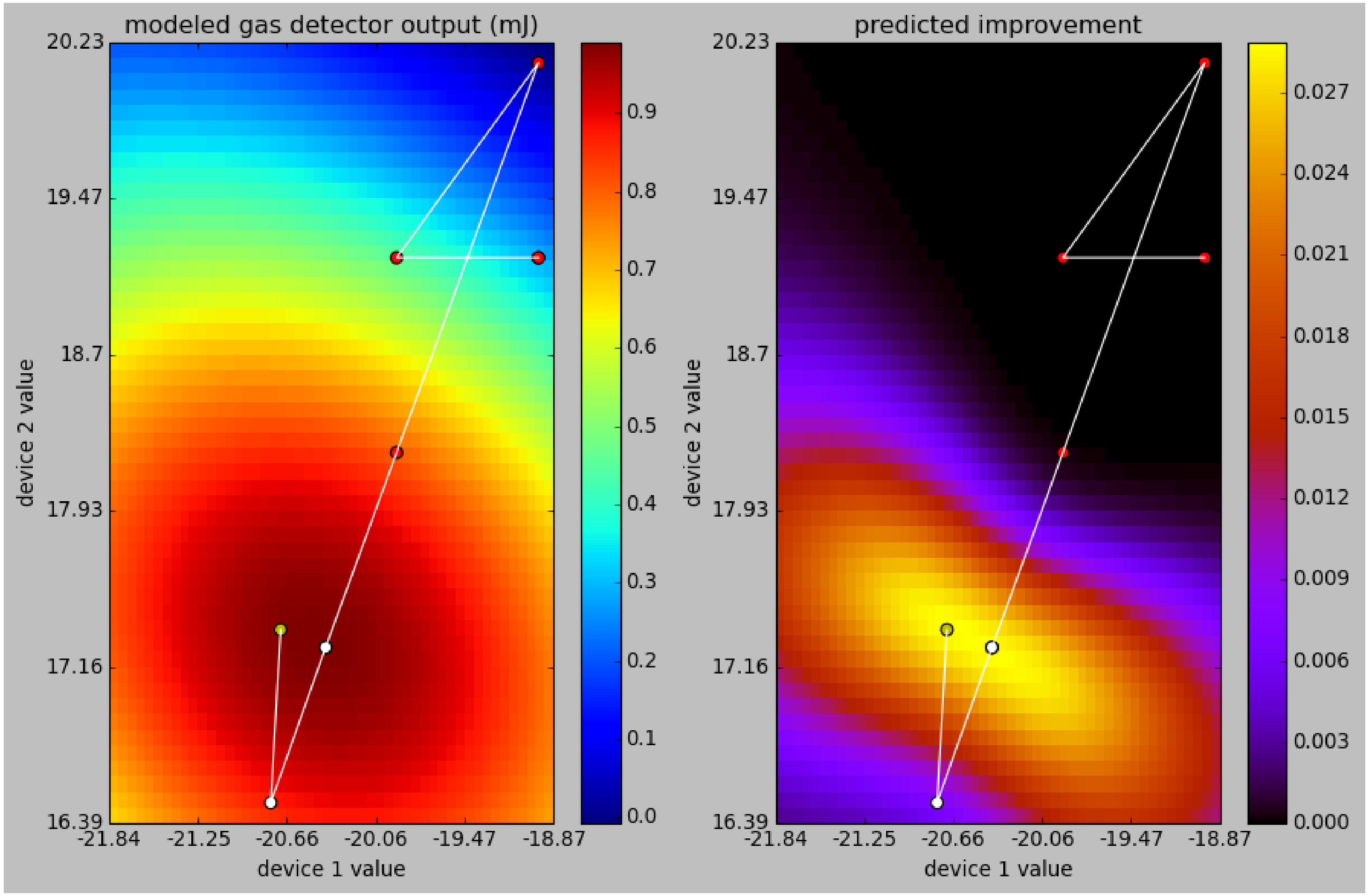
Point 9



OcelotScan-2016-09-21-185122.mat

J. Duris

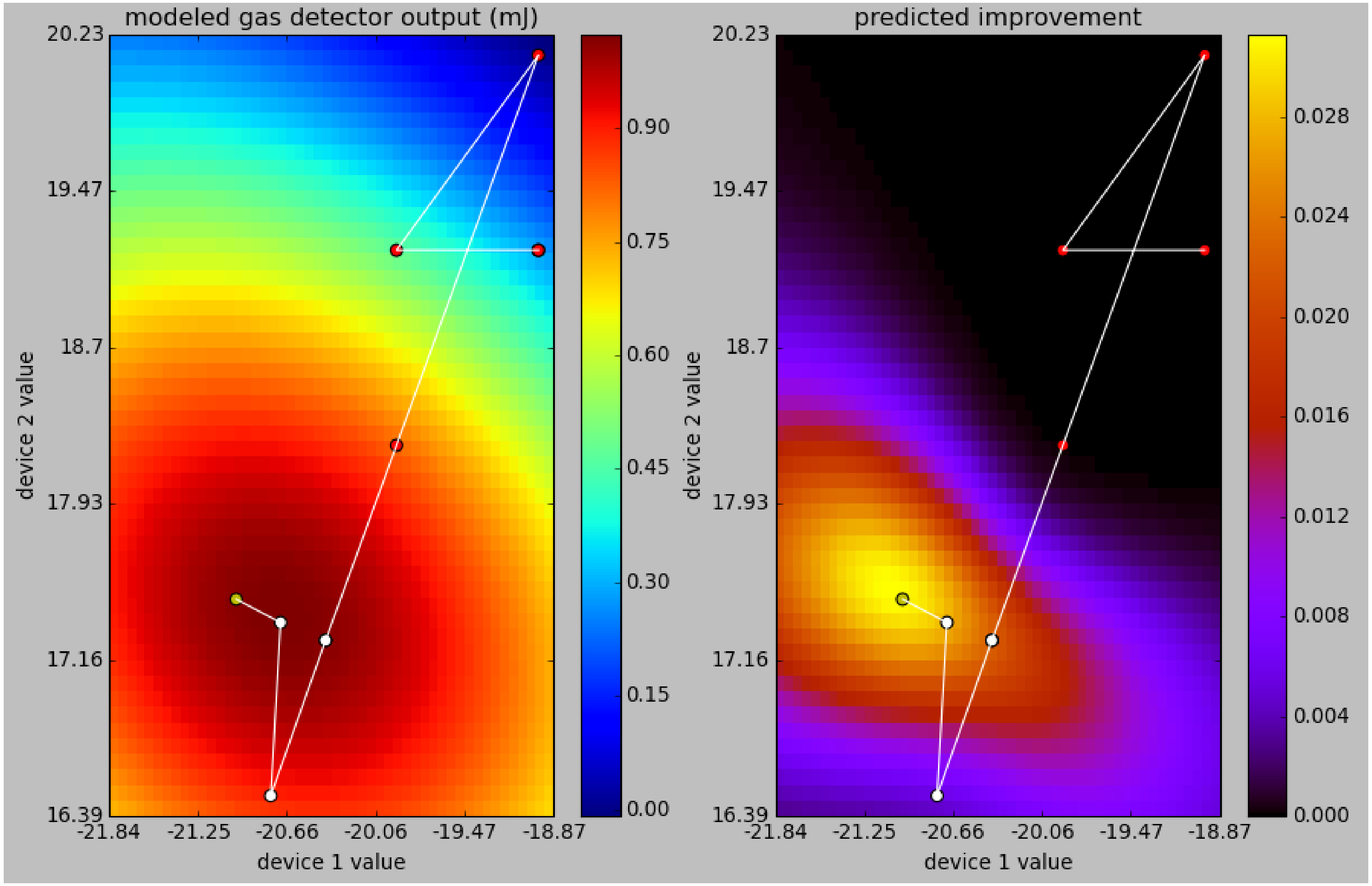
Point 10



OcelotScan-2016-09-21-185122.mat

J. Duris

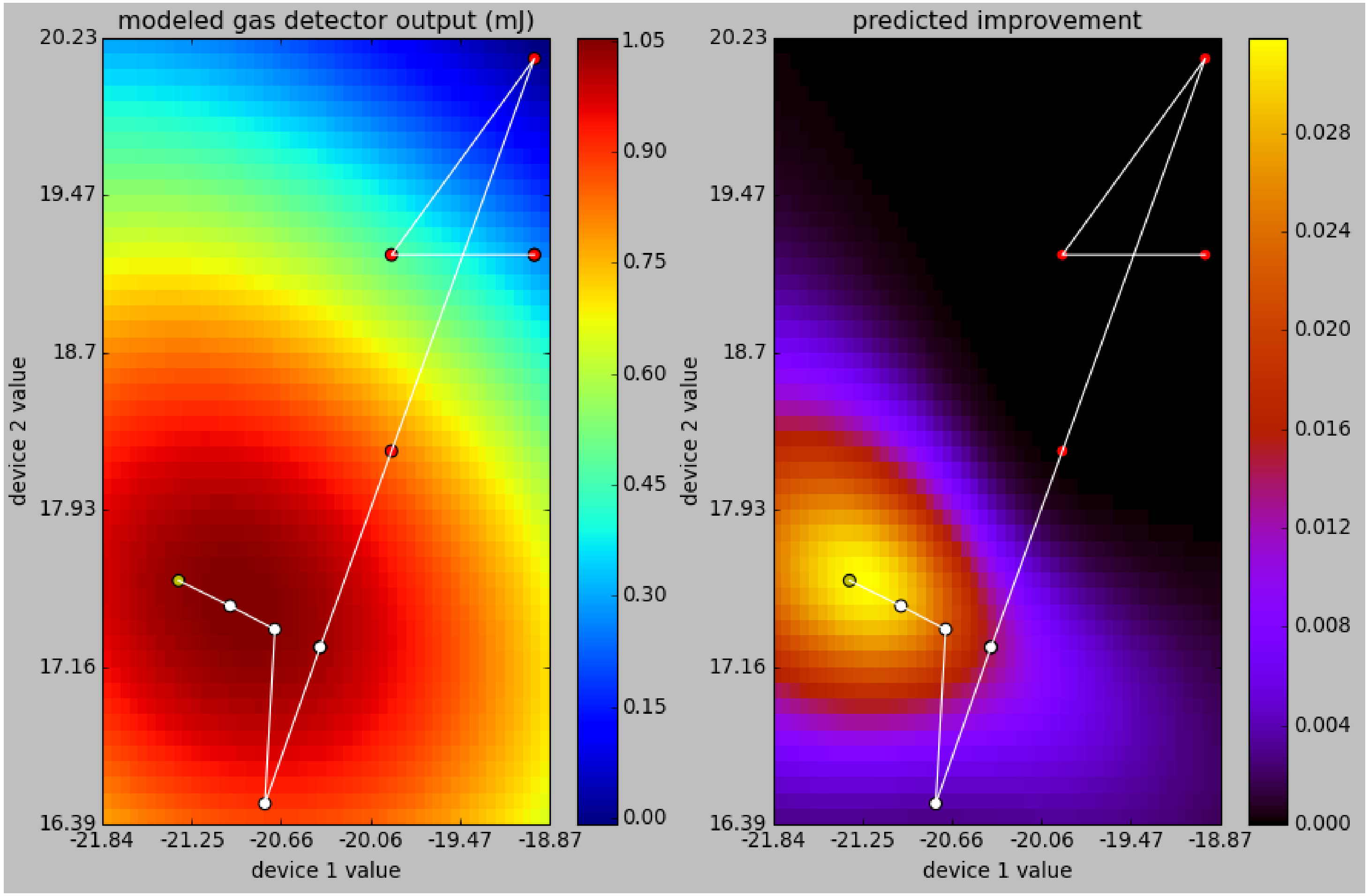
Point 11



OcelotScan-2016-09-21-185122.mat

J. Duris

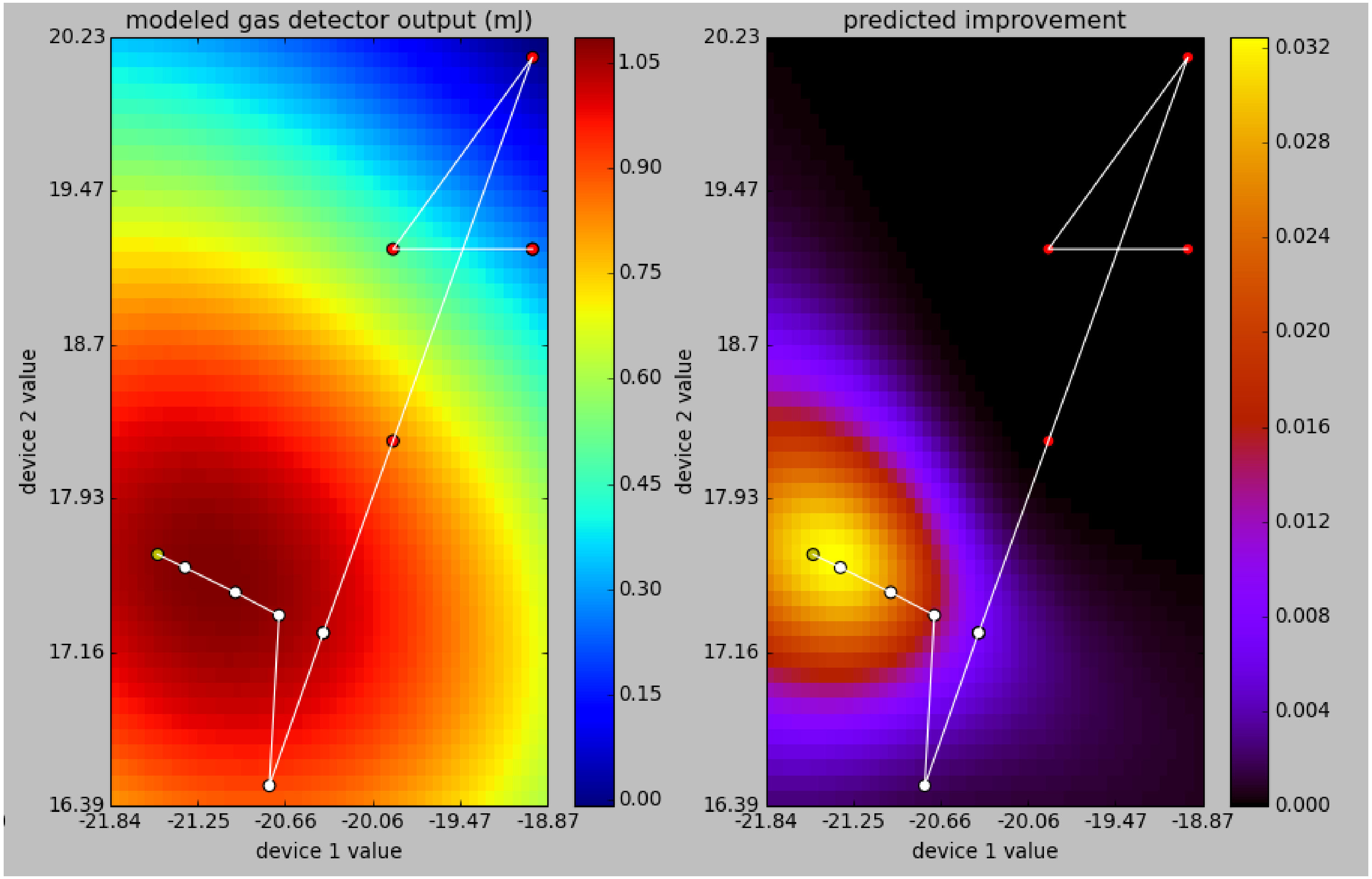
Point 12



OcelotScan-2016-09-21-185122.mat

J. Duris

Point 13

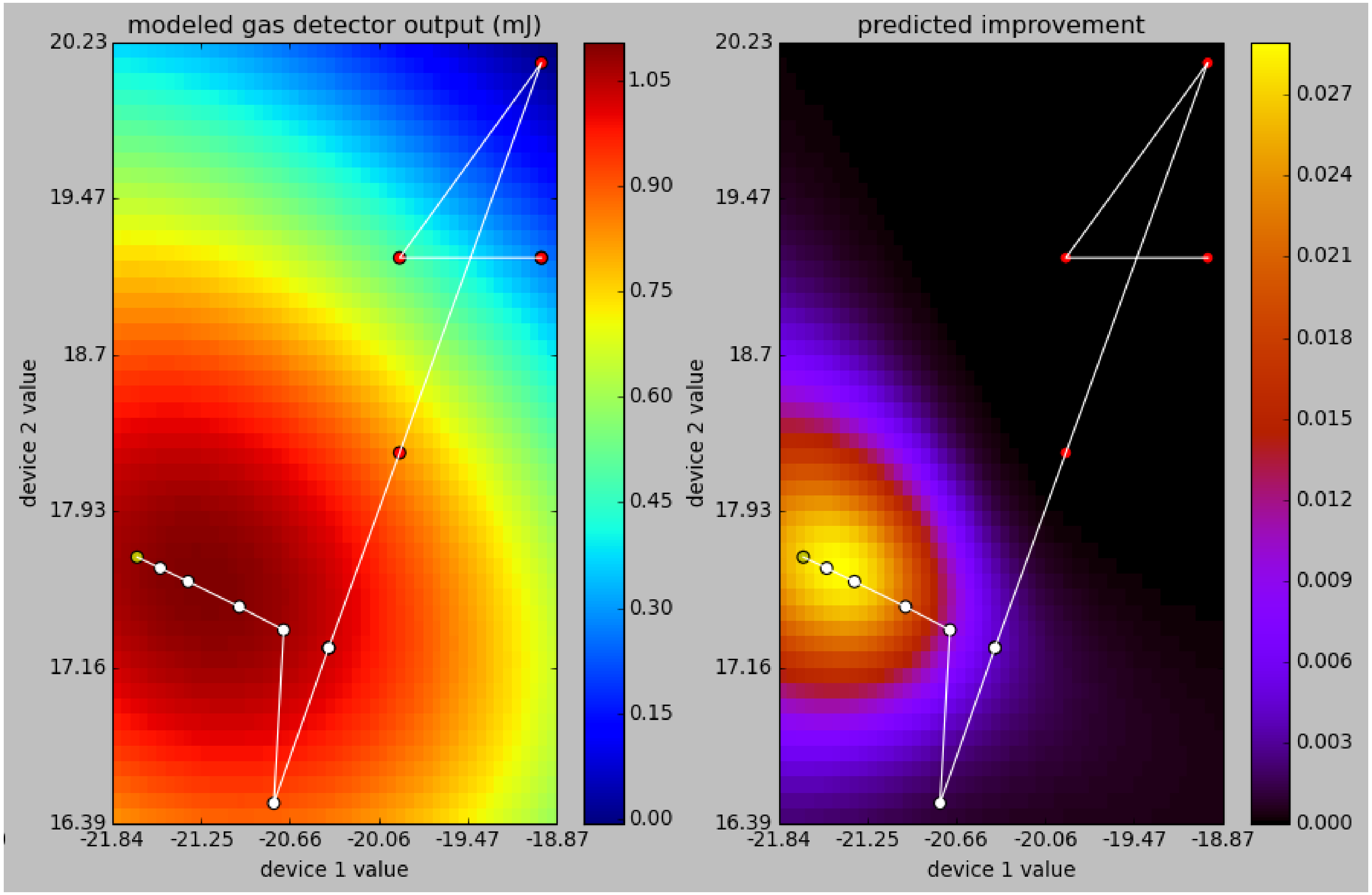


OcelotScan-2016-09-21-185122.mat

J. Duris

Point 14

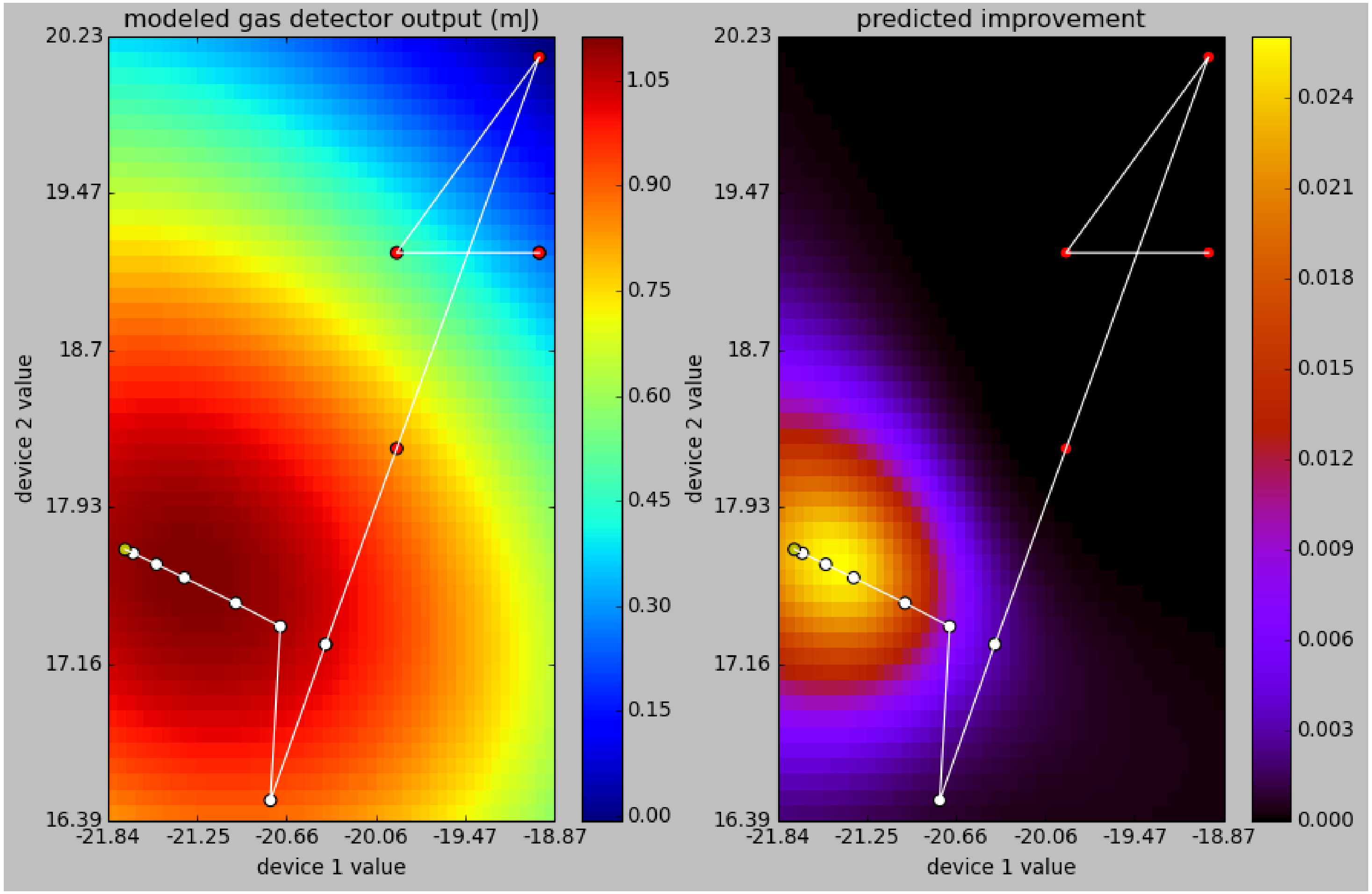
SLAC



OcelotScan-2016-09-21-185122.mat

J. Duris

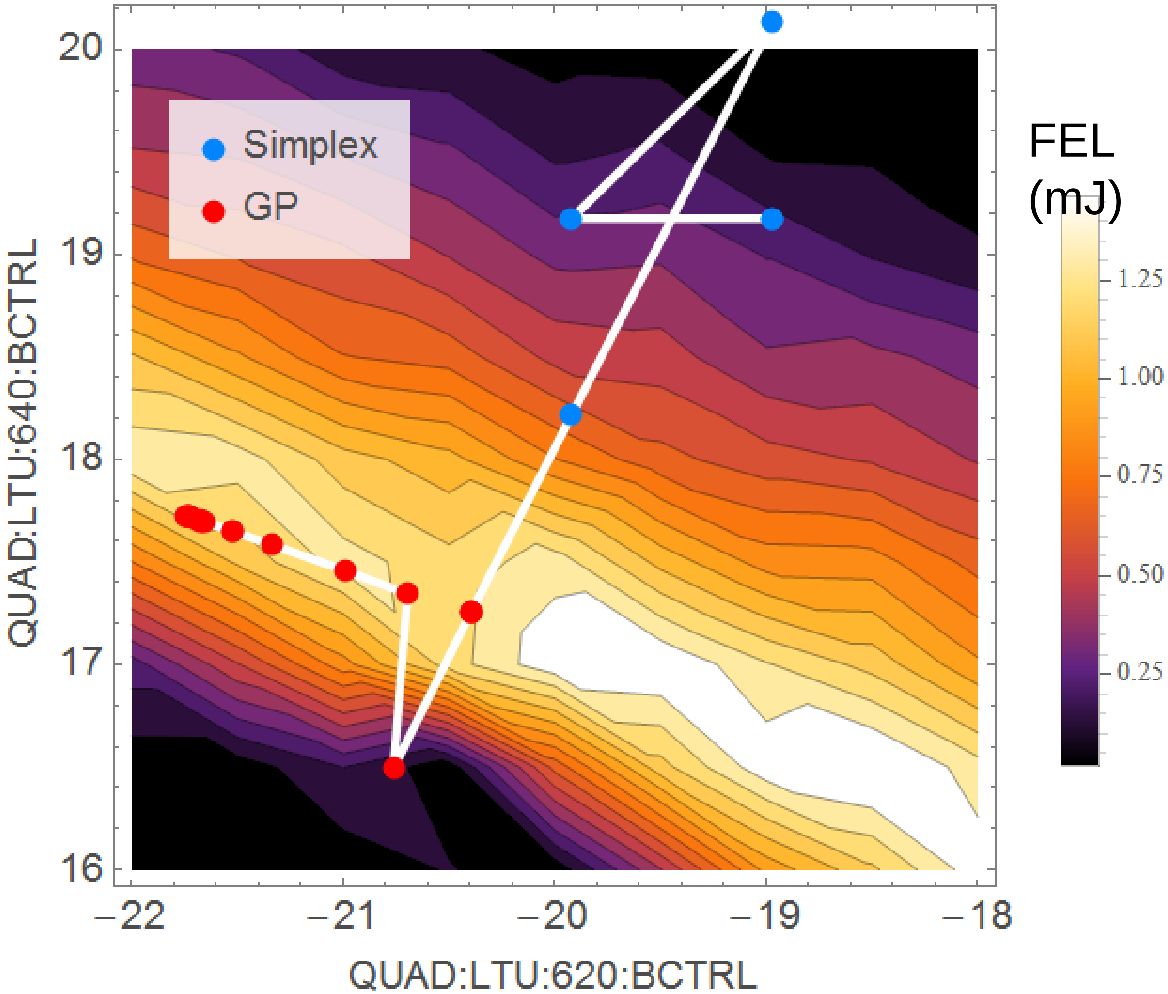
Point 15



OcelotScan-2016-09-21-185122.mat

J. Duris

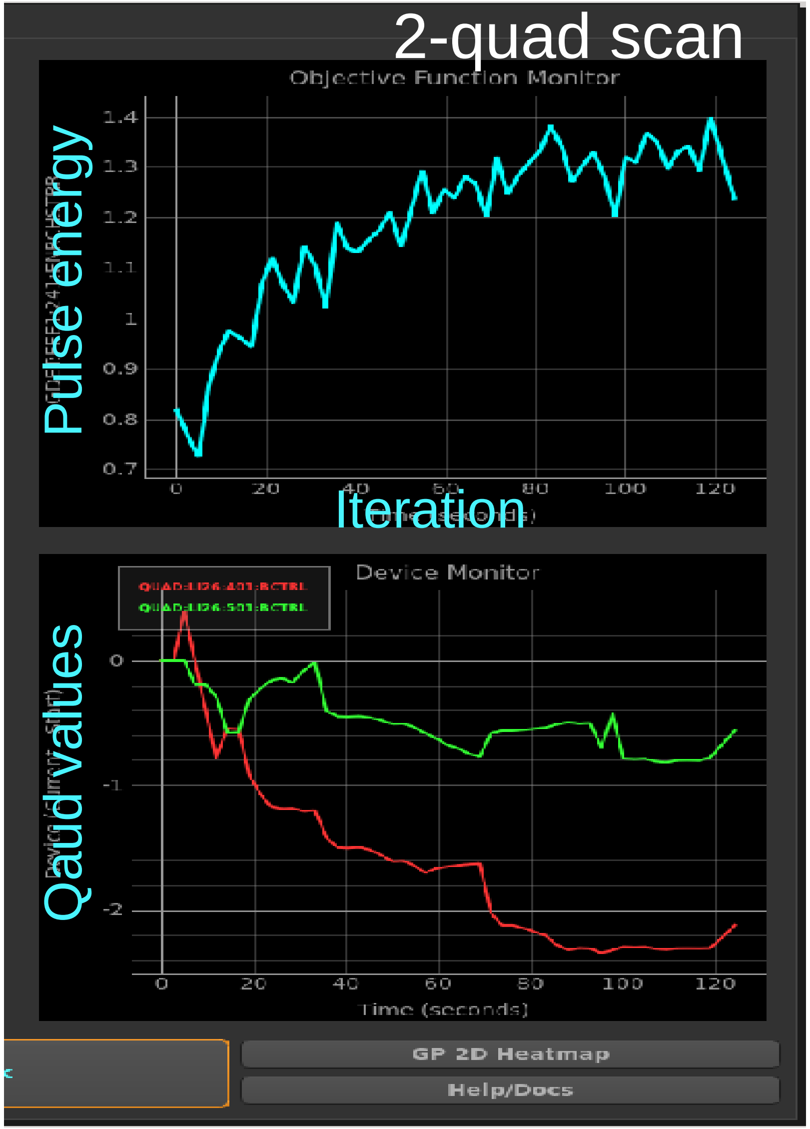
2-quad raster scan + Ocelot path

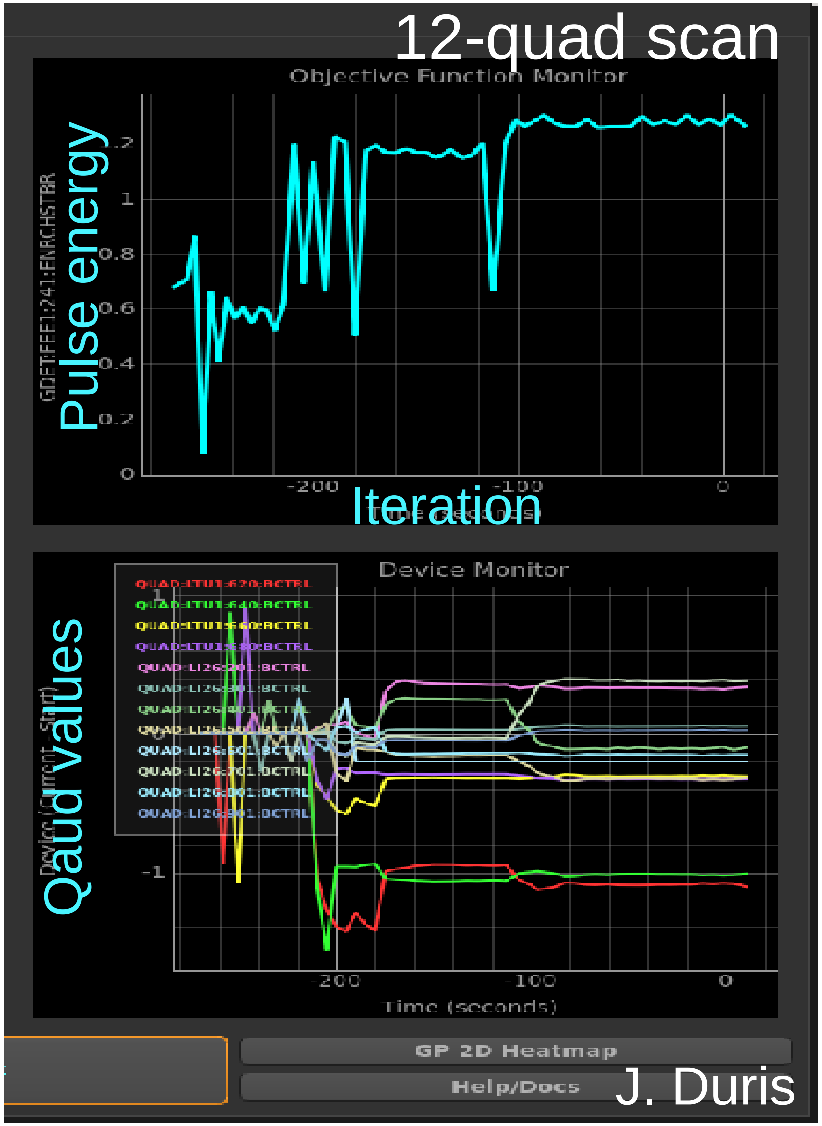


OcelotScan-2016-09-21-185122.mat
CorrelationPlot-QUAD_LTU1_620_BCTRL-2016-09-21-185628.mat

J. Duris

Recent results





J. Duris

Machine Tuning Automation

Performance summary:
Already as good as best human operators!
Still many improvements to come…

	Hand tuning	Simplex Ocelot	GP Ocelot
All runs	25 (30) uJ	72 (50) uJ	95 (51) uJ
Infrequent tuning (>20 mins since last tune)	39 (38) uJ	128 (90) uJ	38 (62) uJ

~4x better than hand tuning

~3x better than hand tuning

Means (and medians in parentheses) FEL change during tune

J. Duris

Machine Tuning Automation

FY16 end – Focused on completing most frequent tasks w/ fast ROI:

Time savings = Est. 103 min / wk

Goal = 210 min / wk

To date 49% of goal

(Evaluating actual integrated savings thru Dec.)

	Tune time (m)	
Procedure	**Past**	**Now**
Injector Tune	180	< 120
Global Steering	6	< 1.5
Und. Pointing	7	3
Global Quad Optimization	20	7

Man hours

Algorithms

FY17 plan – (LFD, AOSD, EED)

- Further code standardization
- Completion of more involved A.I.'s: *XTCAV, true emittance measure/model, E change management, still-faster inj. tuning*
- Plans for LCLS device integration
- Extend machine-agnostic code to add'l SLAC accelerators

J. Duris, D. Ratner, T. Maxwell

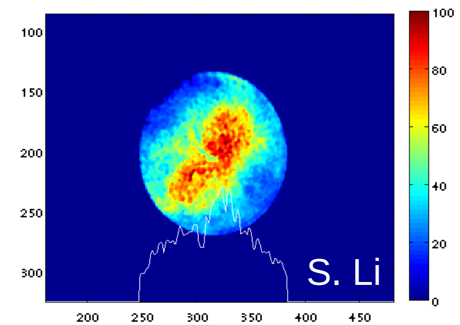
Comparison of FEL changes for different tunes

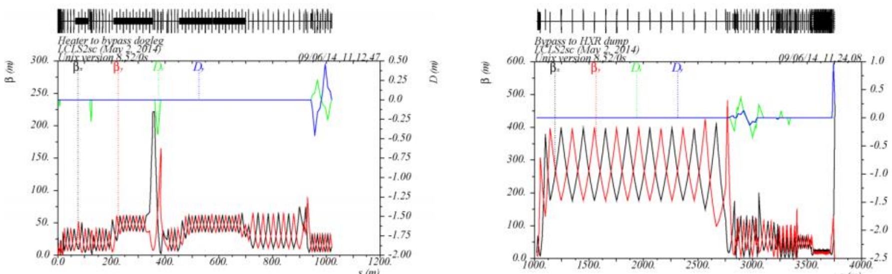
Future directions:

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

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

1. Use ground truth to fit hyper-parameters
2. Use archive/ground truth to introduce prior-mean
3. Expand to more complicated optimization problems (laser profiles, multi-objective functions, etc.)
4. Incorporate physical parameters into the model (i.e fit physical models, not blind tuning parameters.)





Y. Nosochkov, LCLS-II

Comparison of FEL changes for different tunes

Future directions:

Hoping to develop international collaborations on shared online tuning algorithm for accelerators!

Thanks for your attention!

Big thanks to people who did this work: A. Ahmed, T. Cope, J. Duris, S. Ermon, M. Gibbs, T. J. Lane, S. Li, T. Maxwell, M. McIntire, M. Mongia, N. Norvell, D. Sanzone, D. Schneider, C. Yoon