Bayesian optimization at LCLS using Gaussian processes

Optimization of free electron laser pulse energy

Joseph Duris, Dylan Kennedy, Daniel Ratner

ICFA Advanced Beam Dynamics Workshop on High-Intensity and High- Brightness Hadron Beams (HB2018)

June 21, 2018

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

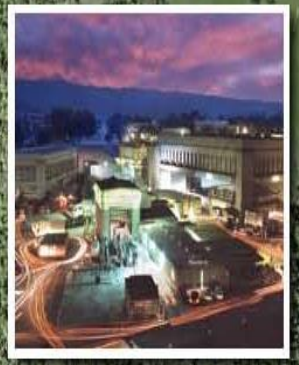
Outline

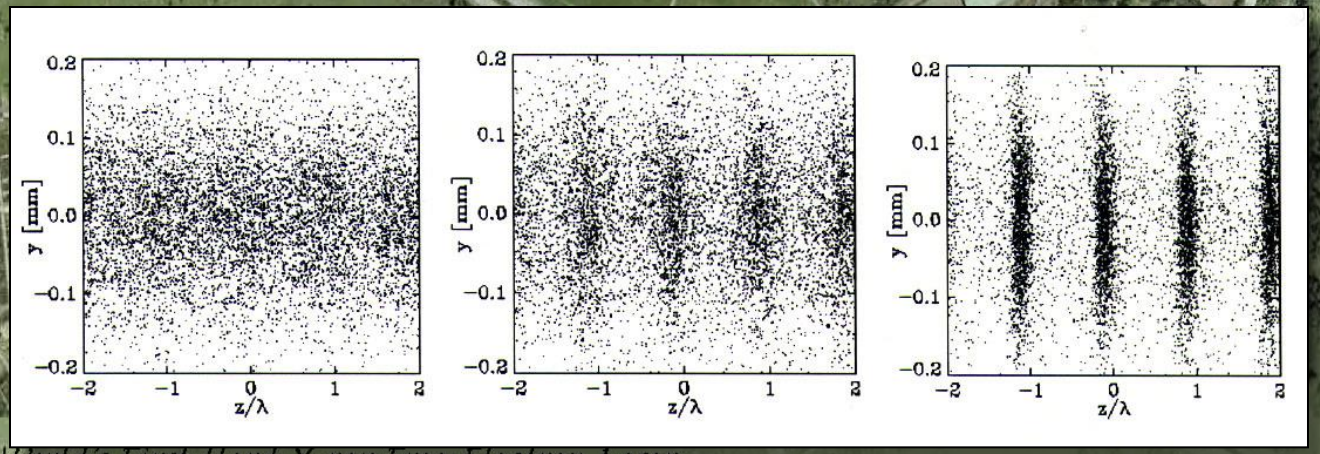
- Problem:
  - Beamline tuning at the Linac Coherent Light Source
- Bayesian optimization:
  - Introduction
  - Gaussian process (GP) for probabilistic modeling
- Our work:
  - Training the GP model on archive data
  - Some results: Bayesian optimization vs simplex for tuning
- Future direction:
  - Limits to archive data
  - Plans to calculate GP model parameters from a physical model

LCLS

The Linac Coherent Light Source (LCLS) is transforming the face of SLAC. For more than 40 years, SLAC's two-mile long linear accelerator has produced cutting edge physics.

Now, scientists will continue this tradition of discovery using the final third of SLAC's linac to create an entirely new kind of laser.



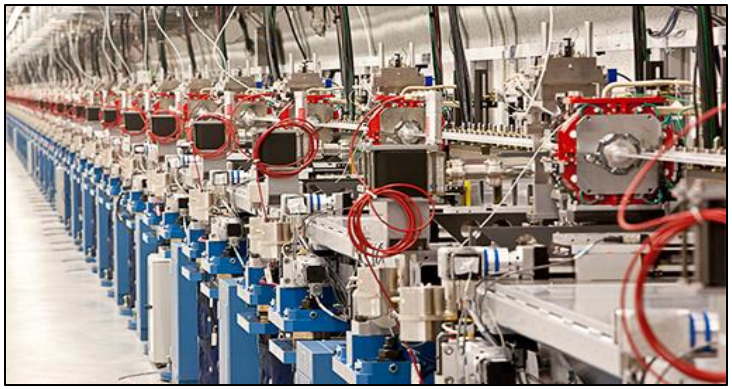


The World's First Hard X-ray Free-Electron Laser

Courtesy D. Ratner

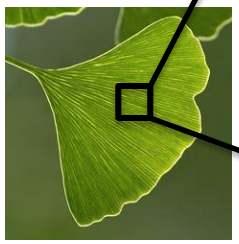
SLAC

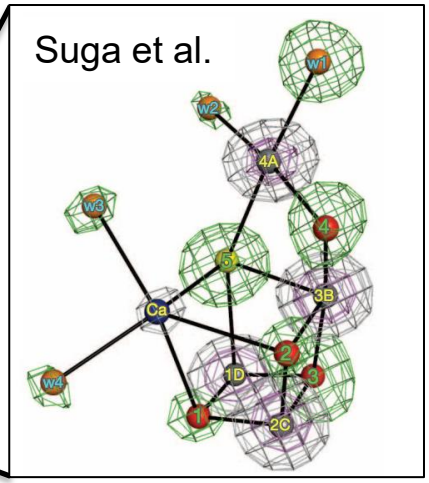
Why build an FEL?



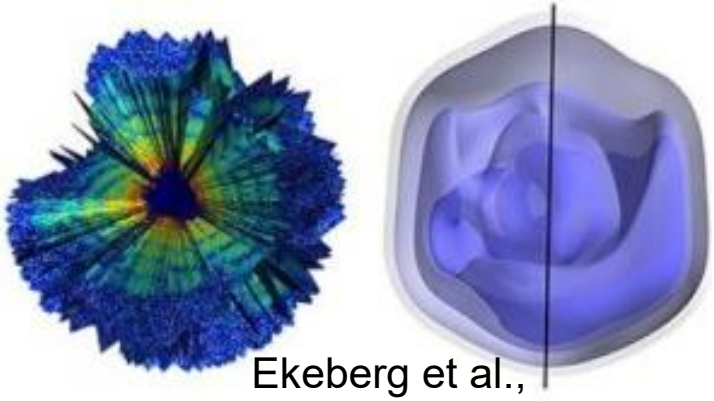
Photosystem II

Structural biology



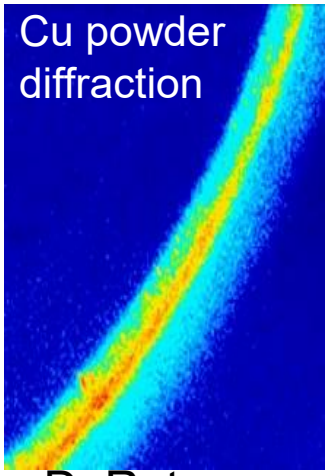


Single particle imaging (Mimivirus)



Ekeberg et al.,

Shock waves in extreme conditions



Courtesy D. Ratner

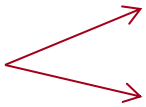
FEL beamline tuning

- LCLS tuning takes ~400 hours of beam time per year
  - Change beamline configurations 2-5 times per day
  - Online optimization of >30 dimensional space
  - Typical setup time ~30 mins

Opportunities for time savings

Action	Time (mins)	Controller	Search space
Config change	10	Operators	small
Tune to find FEL	**5-10**	**Operators**	**large**
Tune quads	**15**	**Simplex**	**24**
Undulator tuning	**5-10**	**Operators**	**30**
Pointing / focusing	5	Operators	small

Most time spent here

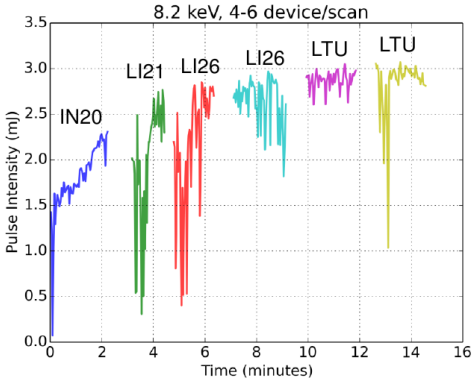


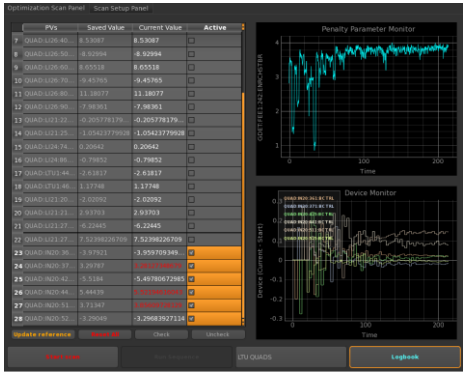
- LCLS-II: more beamlines => more work with same people
- Useful tools needed to help ease increased burden on operators

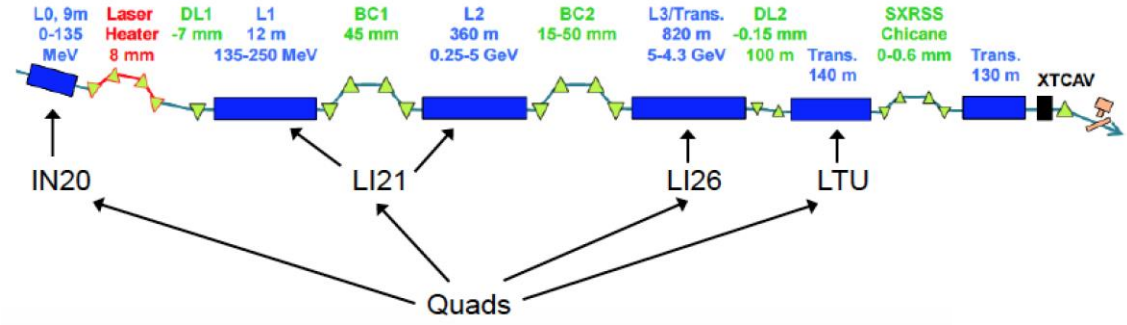
Beamline tuning: FEL vs quads

Current approach to tuning:

- Main objective: FEL pulse energy
- Mostly operator controlled
- Optimization is slow and costly







Ocelot optimizer

- Collaboration with DESY
- Local simplex optimizer
- Small batches of devices

Tuning strategy tradeoffs

SLAC

Human optimization

- **mental models**
- **experience**
- (relatively) slow decisions
- limited working memory

Numerical optimization

- **fast decisions**
- **juggle many things at once**
- blind, local search
  - limited search space

Bayesian optimization with machines combines strengths of both approaches

Bayesian optimization

Challenge: Global optimization of difficult to evaluate, opaque function

Algorithmic solution with Bayesian optimization:

- Build **probabilistic model** of function given data
- **Acquisition function** to choose next point based on model and uncertainties in that model's predictions
- Sample new point and **update probabilities** given new data

Useful for any optimization problem but especially when:

- Function evaluations are noisy
- Function evaluations are costly (time = money)
- Derivatives are difficult to evaluate
- Prior information about the function is available

Comparison with classical machine learning algorithms

SLAC

Classical machine learning optimization often follows this pattern:

- Given a training set of data
- **Fit data to a model** (e.g. linear or neural network regression)
- Use this fit model to make **best guess** predictions for future acquisitions (hope that your model extrapolates well)
- **Refit** model (or back propagate errors to modify weights) & repeat

Bayesian models do not give a best fit; rather, they give posterior distributions over functions.

We exploit knowledge of our model uncertainty to make robust predictions for new test points.

Probabilistic model: Gaussian process

Say we measure a function $f(x_i)$ with additive Gaussian noise

$$y_i = f(\vec{x}_i) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma)$$

Covariance of any two points drawn from f is

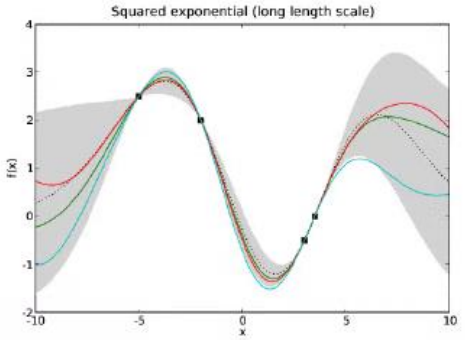
$$\mathrm{cov}(y_i, y_j) = k(\vec{x}_i, \vec{x}_j) + \sigma^2 \delta_{i,j}$$

Covariance or **kernel** function describes a function's structure:

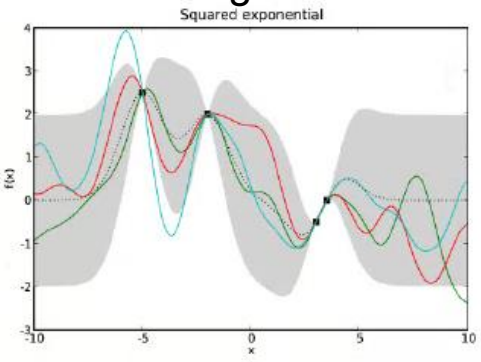
$$k(x_i, x_j) = \theta\, e^{-\frac{1}{2}|x_i - x_j|^2/\ell^2}$$

Radial basis function: neighboring points within a length scale are related

Long length scale



Short length scale



http://pythonhosted.org/infpy/gps.html

Probabilistic model: Gaussian process

Covariance function: $k(\vec{x}_i, \vec{x}_j) = \theta\, e^{-\frac{1}{2}(\vec{x}_i - \vec{x}_j)^T \Sigma (\vec{x}_i - \vec{x}_j)}$

Covariance matrix K for a collection of measured points:

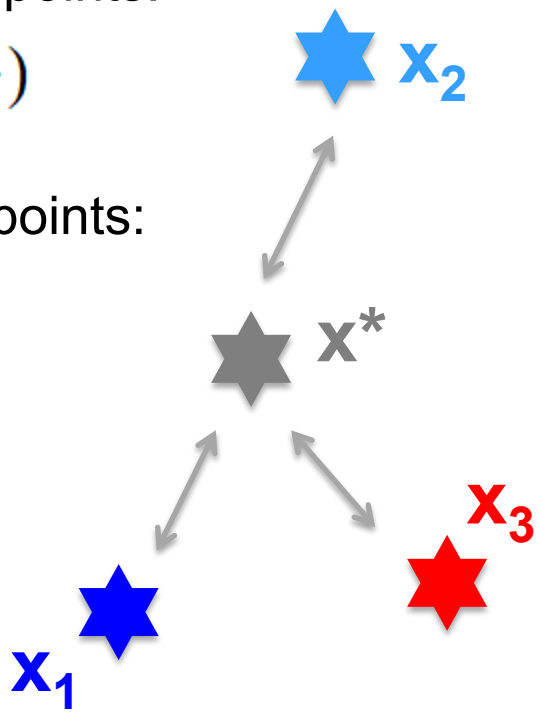
$$\mathrm{cov}(\vec{y}) = K + \sigma^2 I \qquad K_{i,j} = k(\vec{x}_i, \vec{x}_j)$$

Covariance between test point x* and measured points:

$$[K_*]_i = k(\vec{x}_i, \vec{x}_*)$$

Covariance between test point and itself:

$$K_{**} = k(\vec{x}_*, \vec{x}_*)$$



Gaussian process regression

SLAC

Construct a joint prior on the observations and test point

observations

new point to predict

prior mean

new point

$$\begin{bmatrix} \vec{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} K + \sigma^2 I & K_*^T \\ K_* & K_{**} \end{bmatrix}\right)$$

Conditional probability for y* yields **GP prediction**:

Expectation: $\langle y_* \rangle = K_*^T \left(K + \sigma^2 I\right)^{-1} \vec{y} + \vec{y}_p$

Uncertainty: $\sigma_{y*}^2 = K_{**} - K_*^T \left(K + \sigma^2 I\right)^{-1} K_*$

Gaussian process takeaways

GP is a non-parametric model: prediction is a function of measured data rather than a function of fit parameters

What's to love:

- **Posterior PDFs** over functions and hyper-parameters
- Noise is modeled => **immune to overfitting**
- **Automatic model selection**
  - Likelihood maximization guides kernel selection

One issue:

- Matrix inversion => prediction time ~ cubic in number of acquisitions
  - At 2-3 seconds per acquisition, this isn't really a problem
  - Myriad of ways to speed up: parallelize inversion, sparse GP, online GP, kernel interpolation, etc

Acquisition functions

SLAC

- Acquisition functions determine exploration behavior
- Incorporate the prediction's uncertainty in decision

Expected improvement

$$I(\vec{x}) = \max(f(\vec{x}) - y_{\text{best}}, 0)$$

Easy to calculate for a Gaussian PDF!

$$\text{EI}(\vec{x}) = \langle I(\vec{x}) \rangle = \int_{y_{\text{best}}}^{\infty} (f(\vec{x}) - y_{\text{best}})\, p(y \mid \vec{x})\, dy$$

We find using upper confidence bounds yields faster optimization. Free parameters are tuned via Monte Carlo tests.

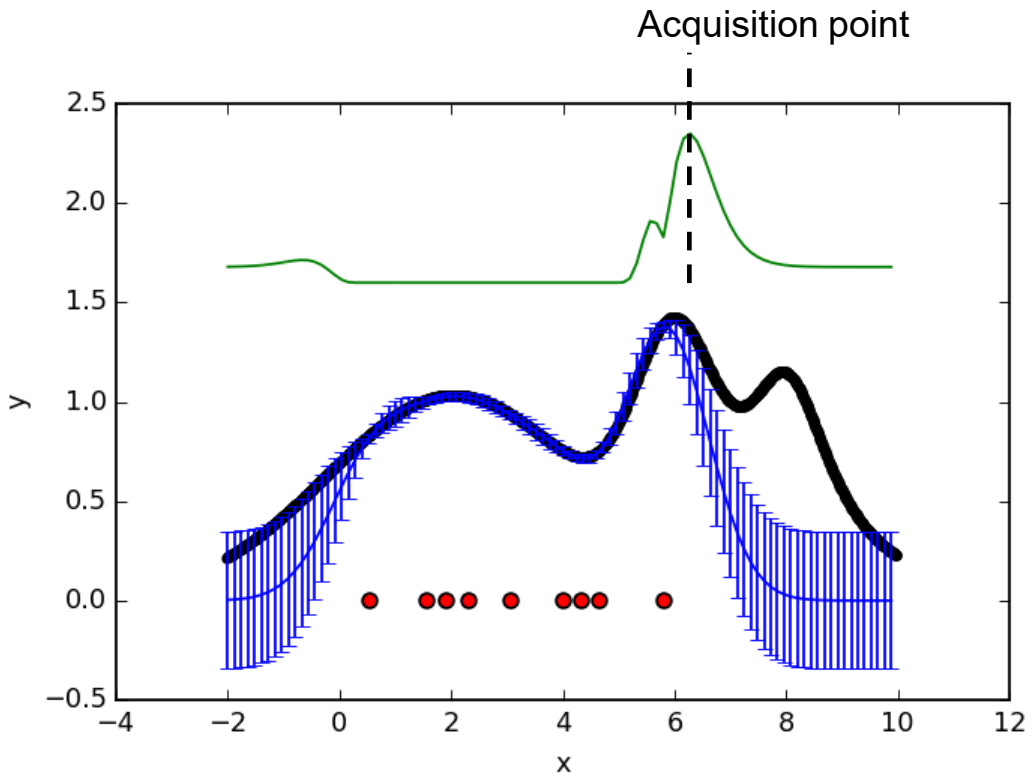
$$UCB(x^*) = \mu(x^*) + \sqrt{(\nu\tau_t)}\sigma(x^*)$$

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

(Srinivas et al., 2010)

Bayesian optimization with Gaussian processes

- Gaussian process => **probabilistic model**
- **Acquisition function** uses resulting probabilities to guide search

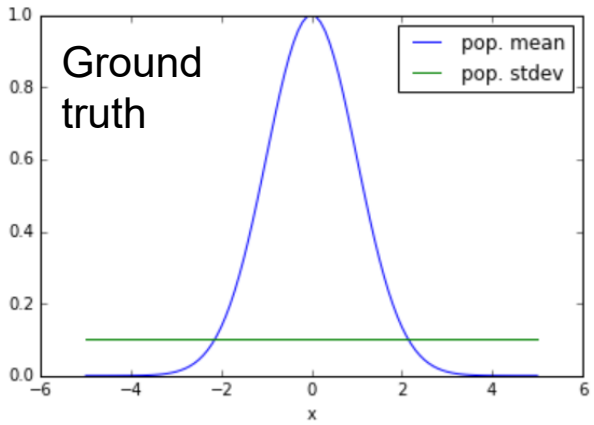


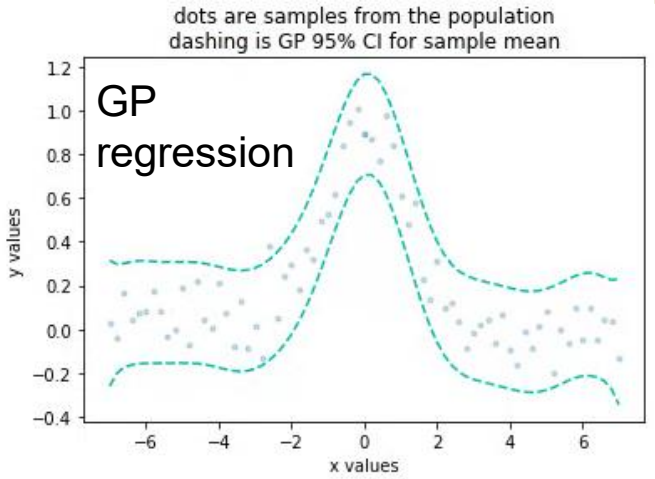
Acquisition Function (EI)

Ground truth

GP posterior

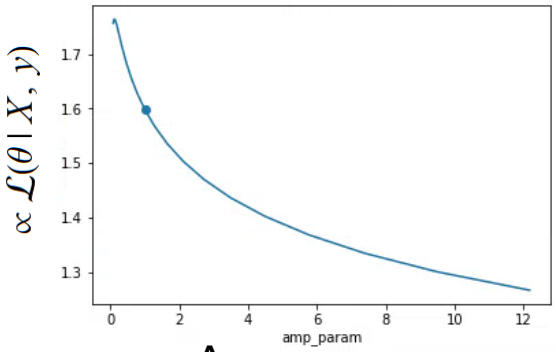
GP fuzzy logic: data is more important than parameters



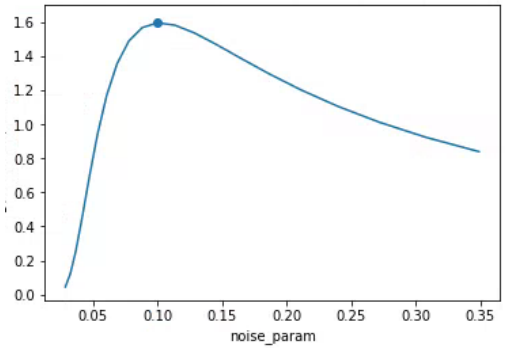


The underlying distribution's moments, are good approximations to the maximum likelihood estimate hyper-parameters.

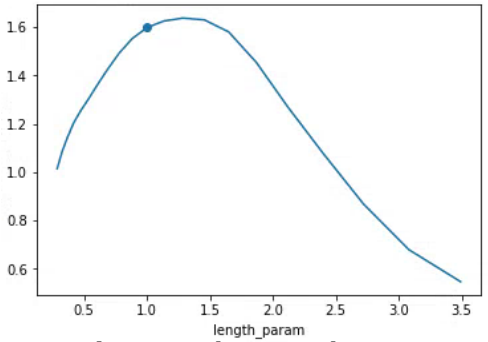
Regression is insensitive to parameter errors.



Amp param



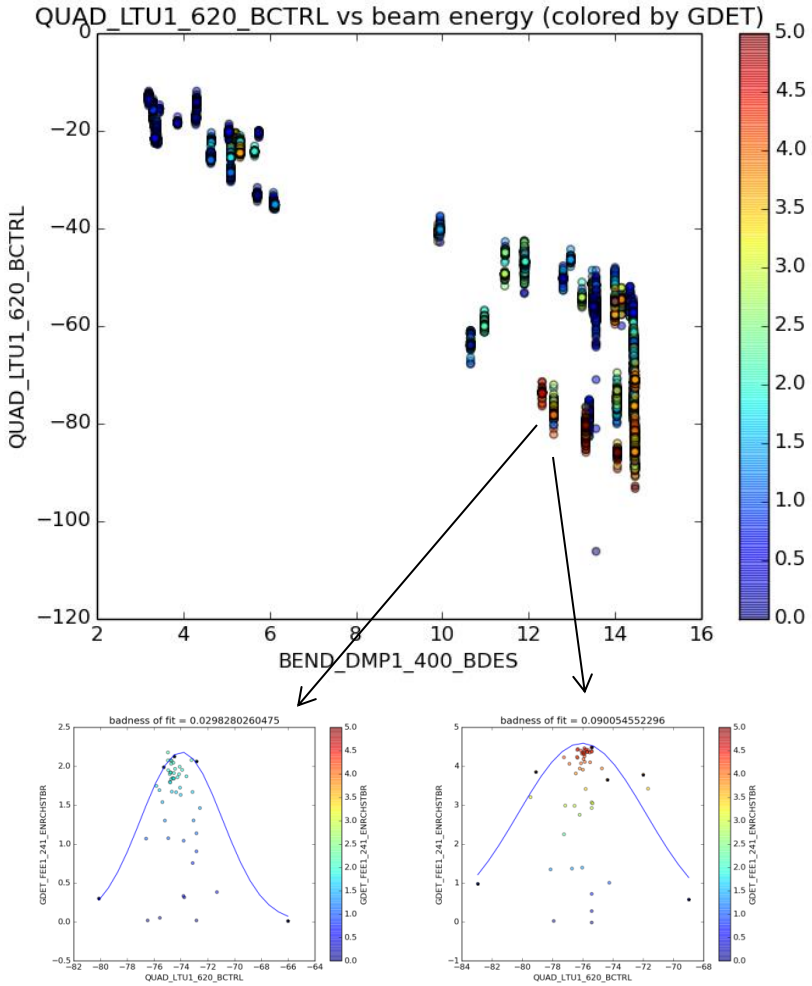
Noise param



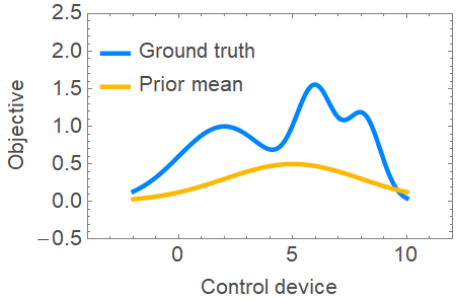
Length scale

Optimize FEL vs quadrupole magnets

GP needs 2 key parameters: **kernel** and **prior mean**. Training data available over a wide range of configs from historical tuning.

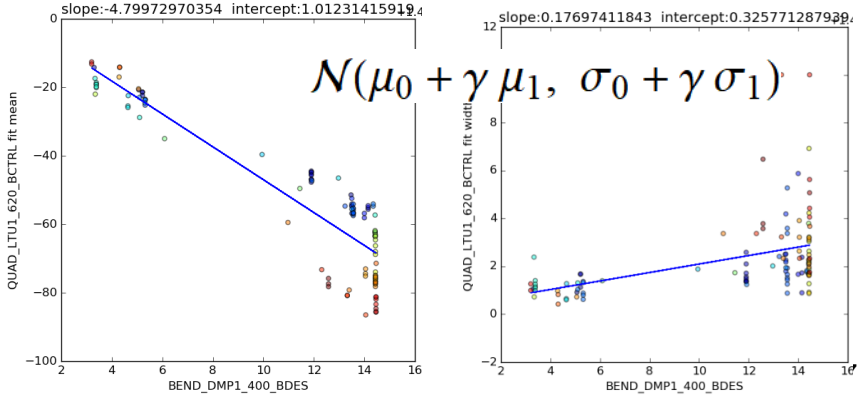


Prior mean biases and constrains search



Trends in peaks of scans=>
GP prior mean

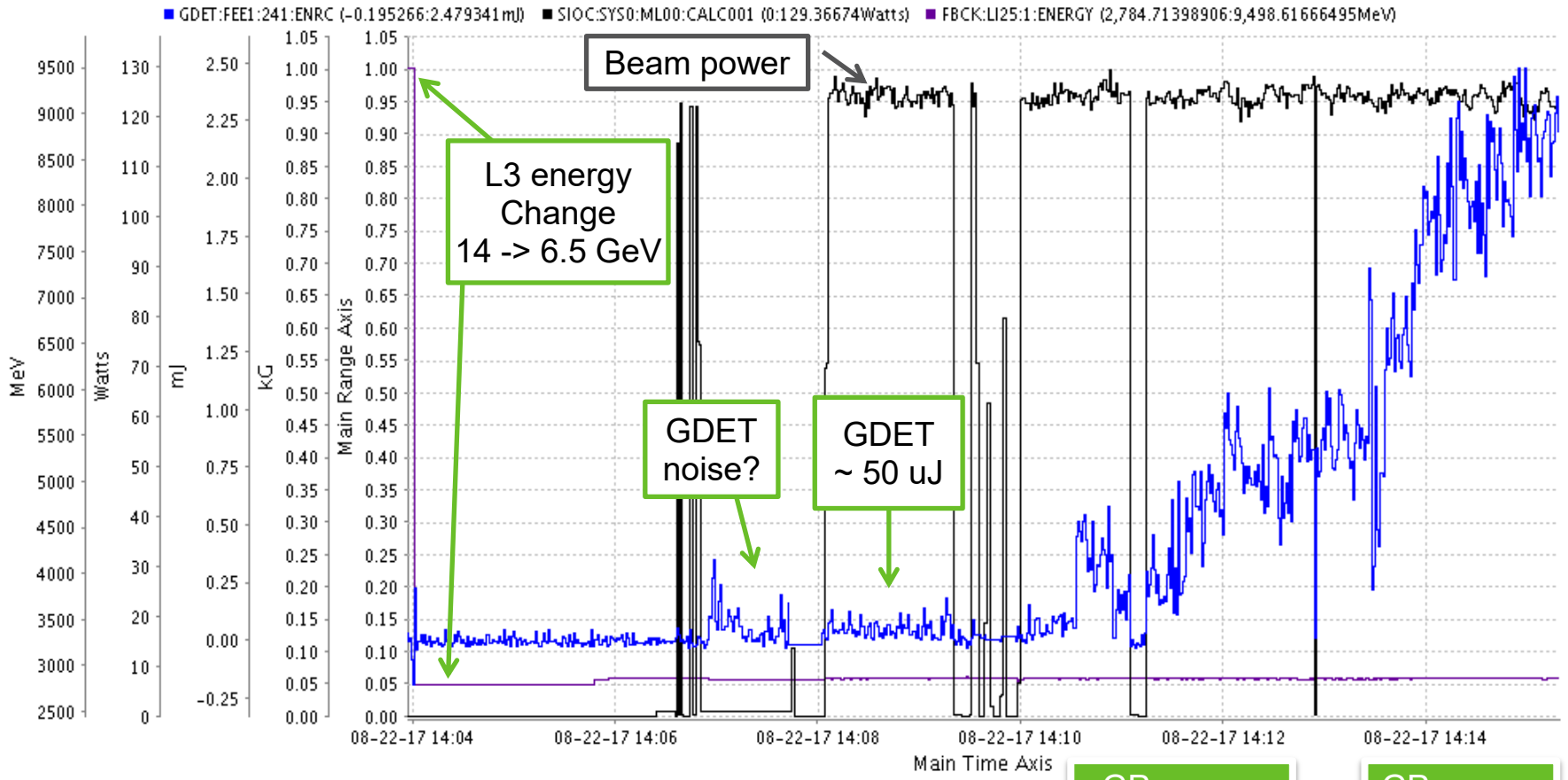
Widths of scans =>
GP kernel parameters



$$\mathcal{N}(\mu_0 + \gamma\,\mu_1,\ \sigma_0 + \gamma\,\sigma_1)$$

Using the prior mean to tune up from noise

We used a Bayes prior from archived data to tune up a brand new config from noise. Simplex could not do this as it needs signal to tune on.

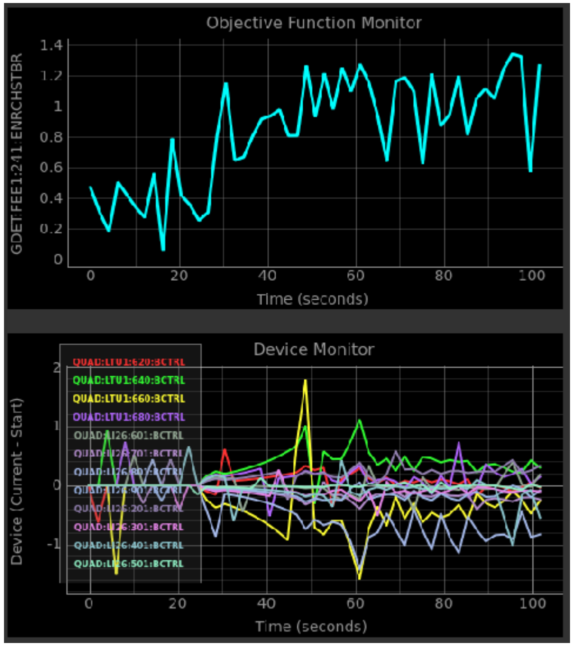


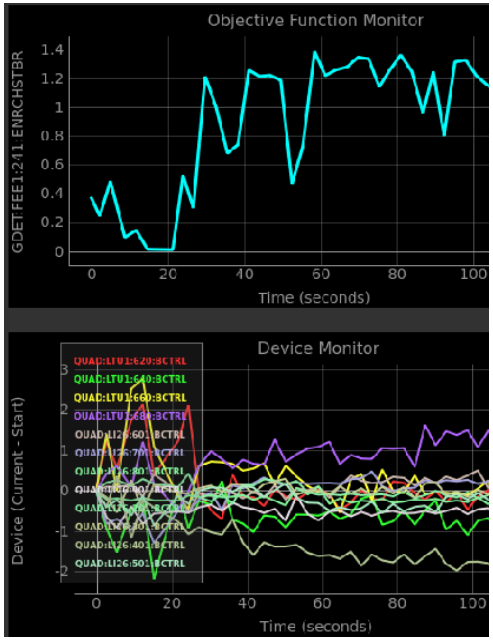
Tuning 12 quads starting with 30% of peak FEL

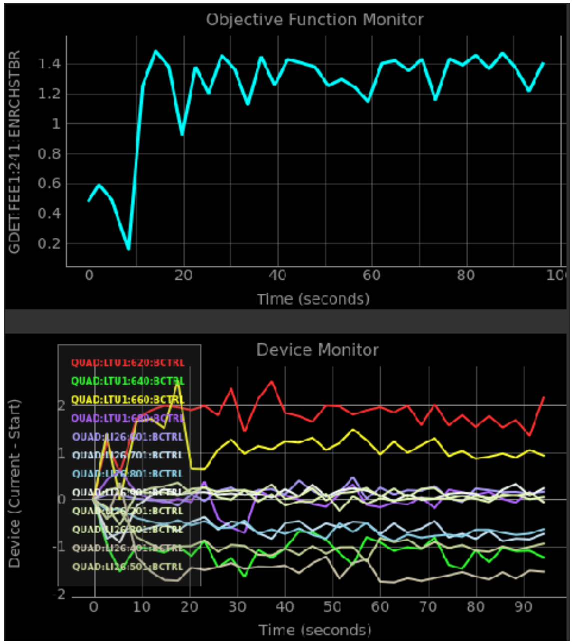
SLAC

Simplex

GP, expected improvement, Jan 2018 prior







Mean of 120 shots

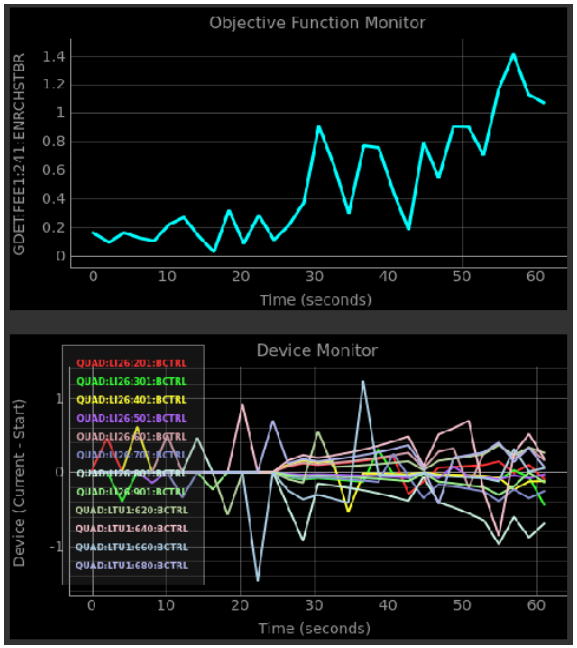
80th percentile of 120 shots

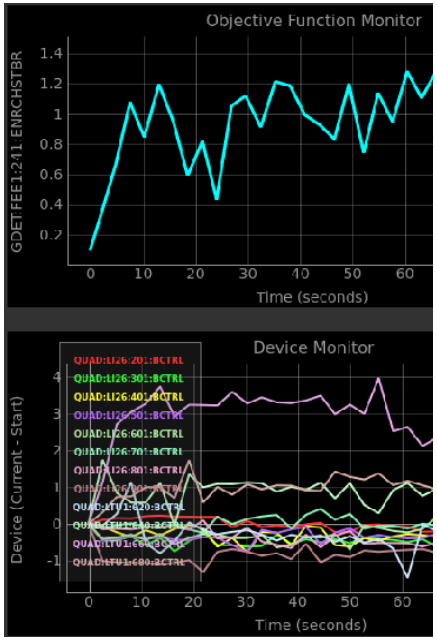
Tuning 12 quads starting with 10% of peak FEL

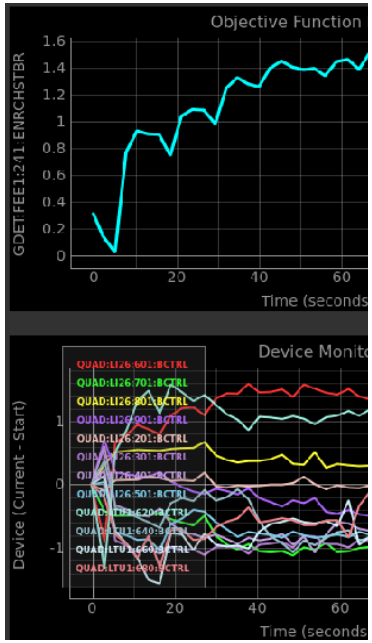
Simplex

GP, expected improvement, Jan 2018 prior

GP, UCB Jan 2018 prior







Mean of 120 shots

80th percentile of 120 shots

Accommodating correlations between devices

SLAC

- FEL vs quads with RBF kernel

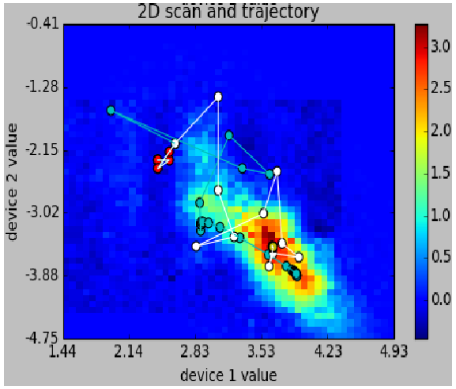
n devices

n x n kernel matrix

$$k_{\mathrm{RBF}}(\vec{x}, \vec{x}') = \exp\left(-\frac{1}{2}(\vec{x} - \vec{x}')^T \Sigma (\vec{x} - \vec{x}')\right)$$

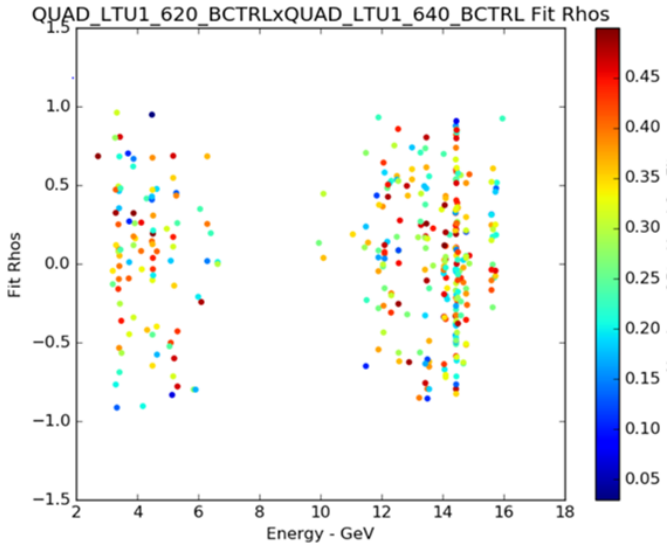
- Diagonal kernel matrix => ignores correlations between quads
- One approach: vary kernel matrix elements to maximize marginal likelihood for a set of prior scans
- Another approach: map x to linearly independent basis y with diagonal kernel matrix

$$\begin{aligned}
&\vec{x}^T \Sigma \vec{x} = \vec{x}^T S^T S \Sigma S^T S \vec{x} \\
&= (S\vec{x})^T \left(S \Sigma S^T\right)(S\vec{x}) \\
&= \vec{y}^T \Sigma' \vec{y} \\
&\text{where } \vec{y} \equiv f(\vec{x}) \equiv S\vec{x} \text{ and } \Sigma' \equiv S \Sigma S^T
\end{aligned}$$

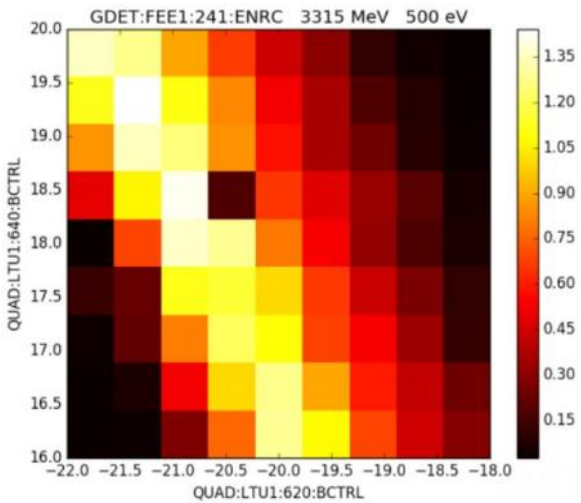


Accommodating correlations between devices

- Correlations are not apparent in the archived data despite obvious relations (adjacent quads are anti-correlated)
- High dimensional search space => sparse data



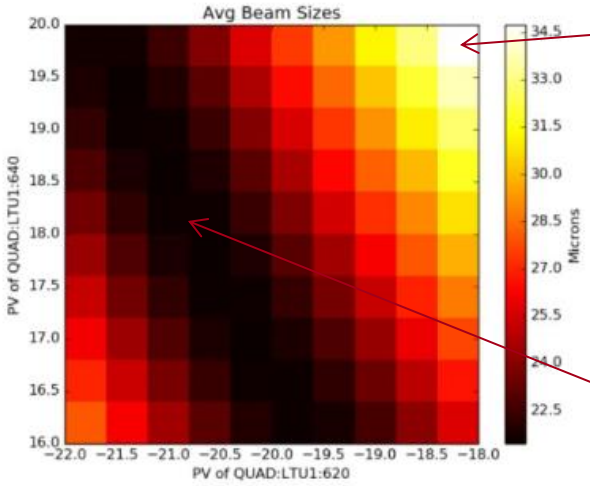
No trend in correlations between quads 620 and 640 in a bunch of scans



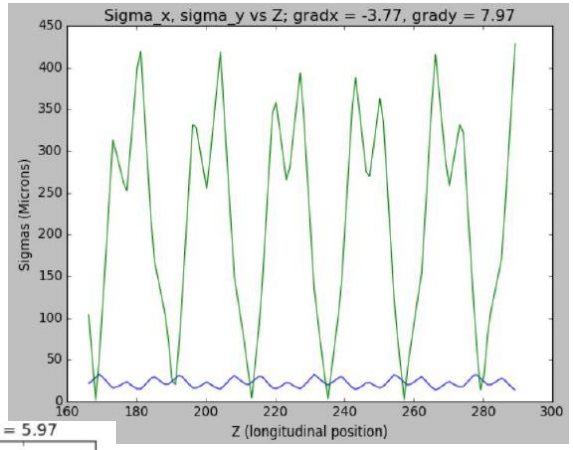
Measured FEL: quads 620 and 640 are adjacent so must be anti-correlated

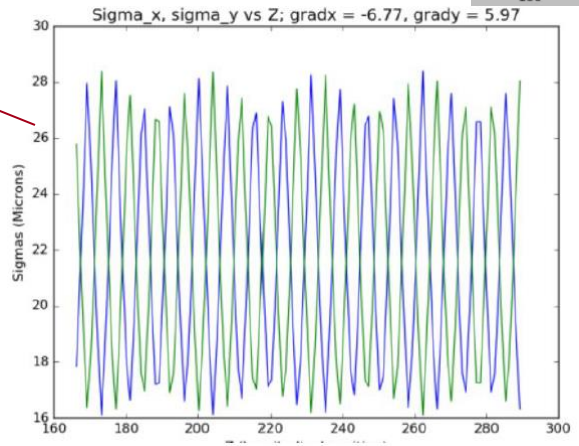
Hyper parameters from physical model

Assuming estimate of Twiss functions somewhere along the beamline, we can use a linear transport model to estimate the beam sizes along the undulators as we change quads.



Modeled beam size vs quads



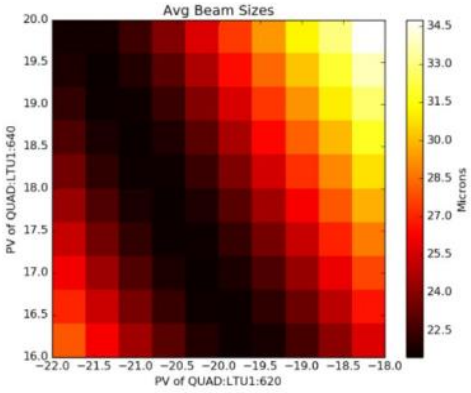


Work by UCSC PhD student D. Kennedy

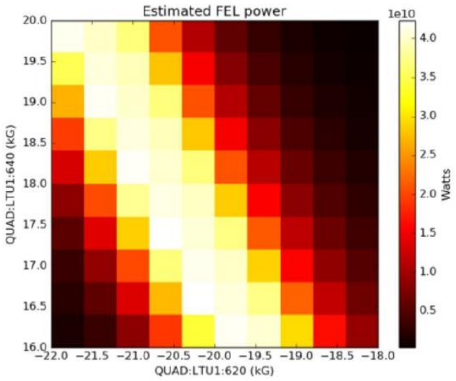
Build GP from physical model

- Estimated beam sizes + Ming Xie FEL model => FEL vs quads
- Reasonable agreement with Genesis FEL code, but faster (~10 ms)

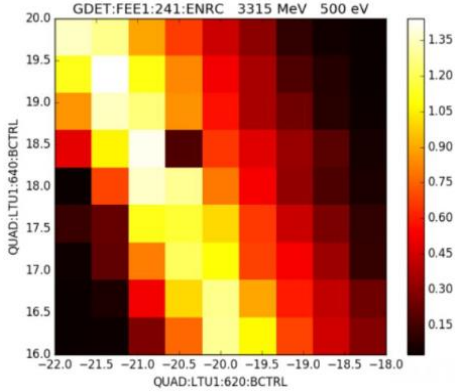
Modeled beam size vs quads



Modeled FEL vs quads



Measured FEL vs quads



- Modeled FEL => prior
- Kernel params: Hessian of -log FEL(quads) estimates RBF kernel with correlations
- After scan completes, GP likelihood yields posterior PDF on latent variables of model (e.g. slice emittance)

LCLS Bayesian optimization progress

We've seen the Bayesian optimizer reduce tuning times from minutes to seconds using

- GP hyper parameters and prior mean from fits to archived scans

Would like to increase speed and consistency:

- FEL model: hyper-parameters and prior
- Correlations: transform coordinates

Expand use-cases

- Tune quads to minimize beam losses
- Self-seeding optics vs. FEL peak brightness
- Tuning quads, undulator taper to maximize FEL pulse energy
- Control x-ray optics to maximize experimental signals

Acknowledgements

Daniel Ratner (SLAC) conceived of the idea of applying Bayesian optimization to LCLS tuning

Mitch McIntire (Stanford CS, Google) implemented the online GP and transforms on the feature space

Dylan Kennedy (UCSC) helped calculate prior and kernel parameters from archive data; now working on FEL model

Stefano Ermon (Stanford CS professor) advised GP deployment

Auralee Edelen (SLAC) new postdoc will help investigate acquisition function strategies.

This work was supported by the Department of Energy, Laboratory Directed Research and Development program at SLAC National Accelerator Laboratory, under contract DE-AC02-76SF00515.

Thank you for your attention!

SLAC

For context: some various optimization methods

Numerical optimizers

Local
- Gradient descent
- Nelder-Mead simplex

Global
- Simulated annealing
- Genetic algorithms

Model based optimization (aka. machine learning)

Deterministic
- Fit model to data
- Policy: optimize cost with respect to model

Bayesian
- Calculate probability over functions given data
- Policy: optimize acquisition function given prediction and uncertainty