Adaptive LiTrack at LCLS

Dorian Bohler, SLAC National Laboratory

Alex Scheinker, Los Alamos National Laboratory

Jovan Nelson, Science Undergraduate Laboratory Internship Program

Outline

Motivation

LiTrack Upgrades & Description

Extremum Seeking

MD Plan & Future Work

Neural Networks

Scheme for a Virtual Diagnostic

Motivation

This project has grown out of conversations that started with TTO for Two Bunch Two Color Experiment

Develop a online simulator with a useful set of tools that might provide the following:

- Optimization
- Virtual Diagnostics: e.g. feedback

Explore using a Machine Learning and/or ES algorithms at LCLS

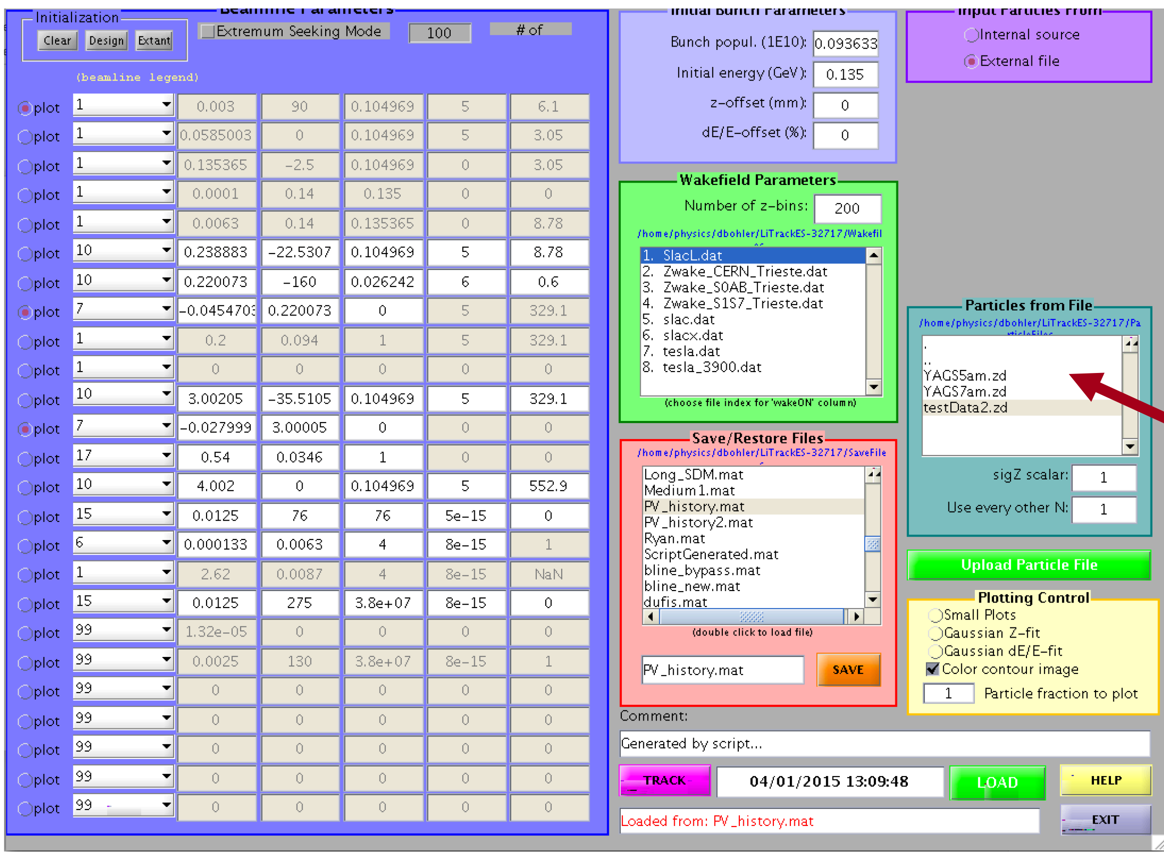
- Using a Neural Network to provide a virtual diagnostic

LiTrack & Recent Changes

<u>Longitudinal Phase Space Tracking Code</u>

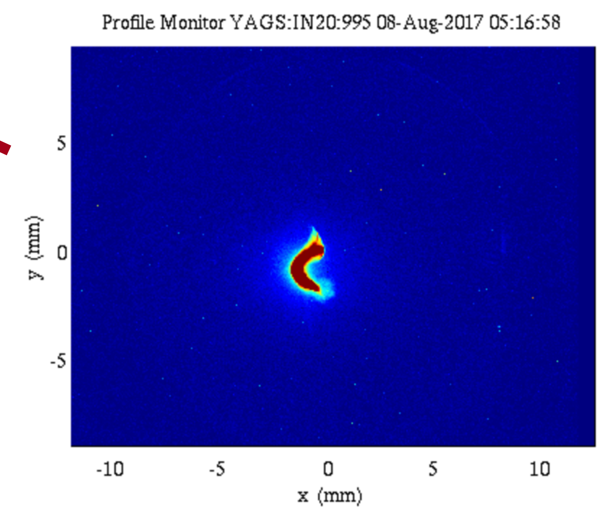
- Faster / simpler than 6D codes
- A lot of physics dominated by longitudinal phase space
- Easier integration

LiTrack & Recent Changes

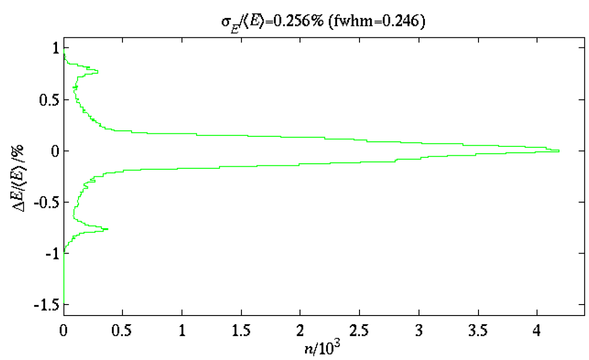


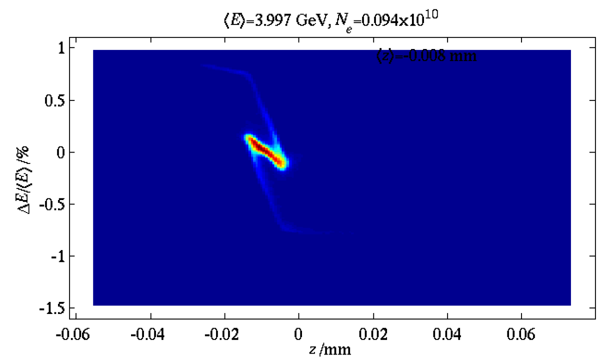
New Features:

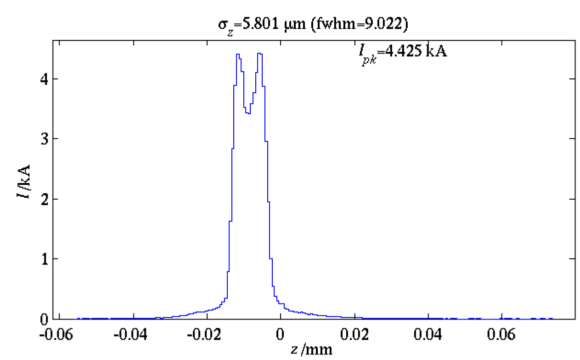
- Input dist. from YAGS2
- Load Machine Parameters
-
- Extremum Seeking Mode



LiTrack Output







SLAC

	opt1	opt2	opt3	opt4	opt5	opt6	opt7	opt8	opt9	opt10
Do Nothing	0.0030	☐	[0,0]	90	☐	[0,0]	0.1050	☐	[0,0]	5
Do Nothing	0.0575	☐	[0,0]	0	☐	[-6:6]	0.1050	☐	[0,0]	5
Do Nothing	1.9794e-04	☐	[0,0]	-2.5000	☐	[-6:6]	0.1050	☐	[0,0]	0
Do Nothing	1.0000e-04	☐	[0,0]	0.1400	☐	[0,0]	0.1350	☐	[0,0]	0
Do Nothing	0.0063	☐	[0,0]	0.1400	☐	[0,0]	1.9794e-04	☐	[124,145]	0
L1S Linac	0.2354	☐	[0,0]	-28.5743	☑	[-45.5:5.5]	0.1050	☐	[0,0]	5
L1X Linac	0.2182	☐	[0,0]	-160	☑	[-170:-150]	0.0262	☐	[0,0]	6
BC1 Chicane	-0.0463	☐	[0,0]	0.2182	☑	[0.200:0.01:0...	0	☐	[0,0]	5
Do Nothing	0.2000	☐	[0,0]	0.0940	☐	[0,0]	1	☐	[0,0]	5
Do Nothing	0	☐	[0,0]	0	☐	[0,0]	0	☐	[0,0]	0
L2 Linac	5.0021	☐	[0,0]	-34.4345	☑	[-45:-23]	0.1050	☐	[0,0]	5
BC2 Chicane	-0.0280	☐	[0,0]	5.0001	☑	[3:0.3:5.5]	0	☐	[0,0]	0
CSR	0.5400	☐	[0,0]	0.0346	☐	[0,0]	1	☐	[0,0]	0
L3 Linac	13.9120	☐	[0,0]	-13	☑	[-30:0]	0.1050	☐	[0,0]	5
Resistive-wall wakefield	0.0125	☐	[0,0]	76	☐	[0,0]	76	☐	[0,0]	5.0000e-15
DL2 Compressor	1.3300e-04	☐	[0,0]	0.0063	☐	[0,0]	13.9100	☐	[0,0]	8.0000e-15
Do Nothing	2.6200	☐	[0,0]	0.0087	☐	[0,0]	4	☐	[0,0]	8.0000e-15
Resistive-wall wakefield	0.0125	☐	[0,0]	275	☐	[0,0]	38000000	☐	[0,0]	8.0000e-15
End of Beamline	1.3200e-05	☐	[0,0]	0	☐	[0,0]	0	☐	[0,0]	0
End of Beamline	0.0025	☐	[0,0]	130	☐	[0,0]	38000000	☐	[0,0]	8.0000e-15
End of Beamline	0	☐	[0,0]	0	☐	[0,0]	0	☐	[0,0]	0
End of Beamline	0	☐	[0,0]	0	☐	[0,0]	0	☐	[0,0]	0
End of Beamline	0	☐	[0,0]	0	☐	[0,0]	0	☐	[0,0]	0
End of Beamline	0	☐	[0,0]	0	☐	[0,0]	0	☐	[0,0]	0
End of Beamline	0	☐	[0,0]	0	☐	[0,0]	0	☐	[0,0]	0

Extremum Seeking Example

SLAC

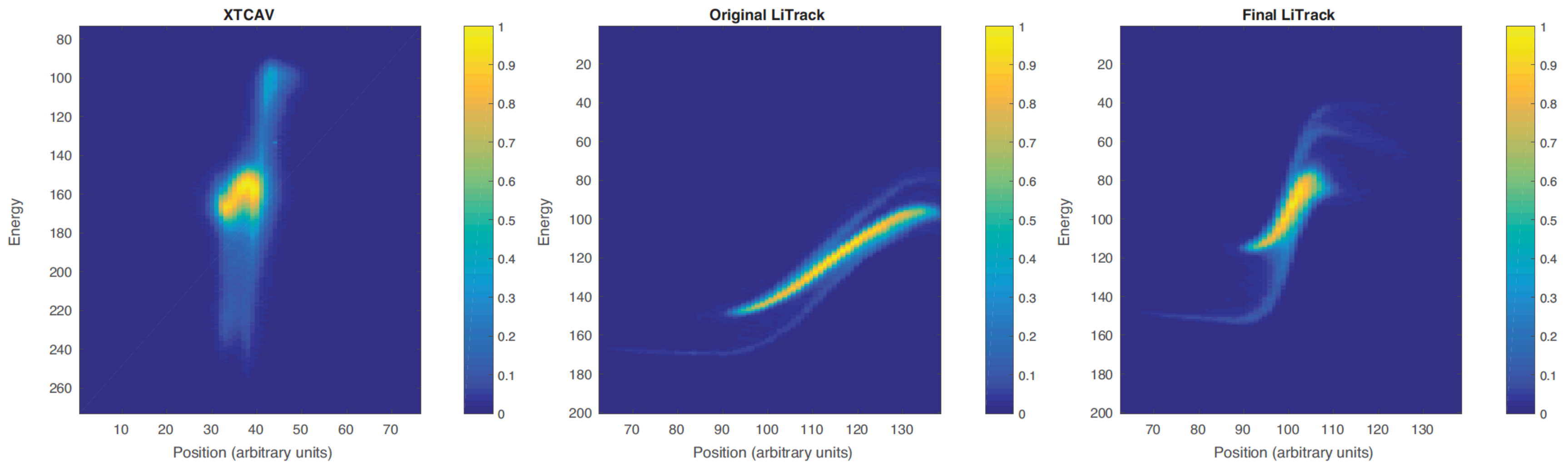


Figure 2: Measured XTCAV, original LiTrack and final, converged LiTrack energy vs position phases space of the electron bunch shown.

<u>ES advantages</u>

1) Multiple parameters tuned simultaneously by dithering at independent frequencies, thereby creating orthogonality in Hilbert space, and extracting the influence of each parameter in real time.

2) Robustness to system and parameter time variation and measurement noise – safe.

Iterative Extremum Seeking Algorithm (Obs.)

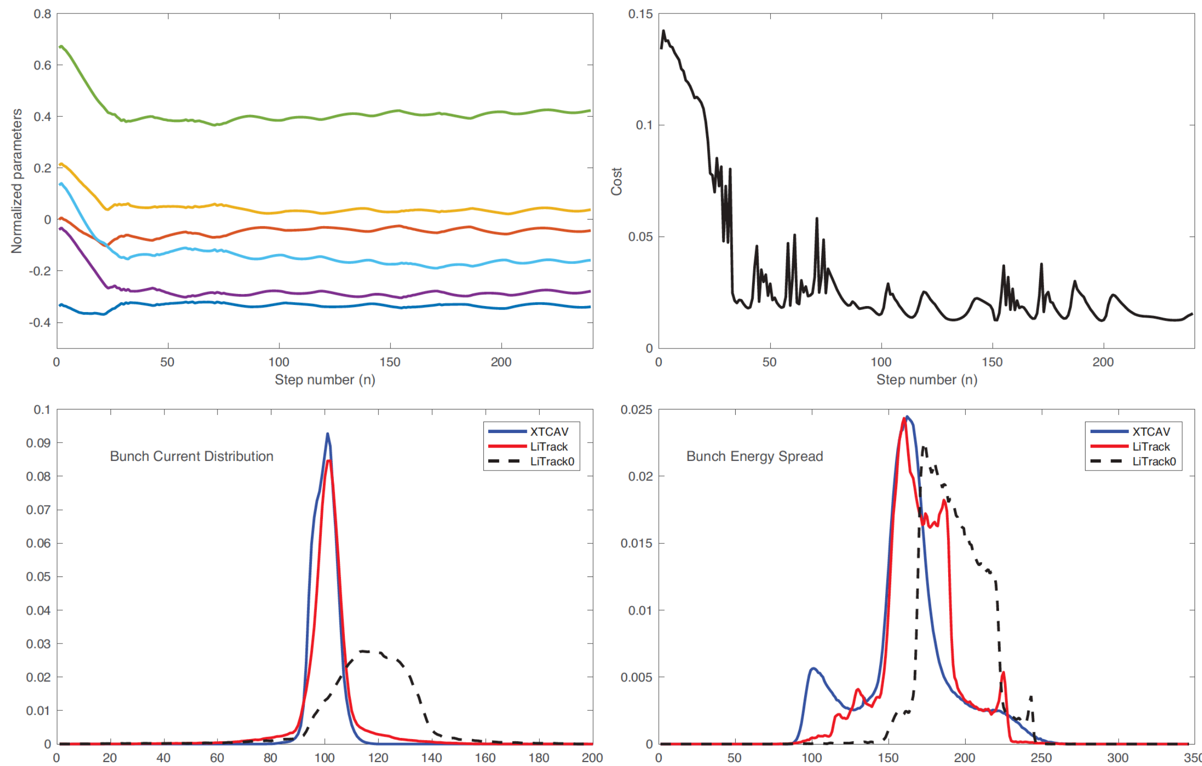


Figure 1: Parameter convergence and cost minimization for matching desired bunch length and energy spread profiles.

XTCAV measured : $\rho_{\mathrm{TCAV}}(\Delta E, \Delta z)$,
LiTrack simulated : $\rho_{\mathrm{LiTrack}}(\Delta E, \Delta z)$.

integrated along projections:
1D energy and charge distrs.

$$\rho_{E,\mathrm{TCAV}}(\Delta E), \quad \rho_{z,\mathrm{TCAV}}(\Delta z),$$
$$\rho_{E,\mathrm{LiTrack}}(\Delta E), \quad \rho_{z,\mathrm{LiTrack}}(\Delta z).$$

$$C_E = \int \left[\rho_{E,\mathrm{TCAV}}(\Delta E) - \rho_{E,\mathrm{LiTrack}}(\Delta E)\right]^2 d\Delta E, \quad (1)$$
$$C_z = \int \left[\rho_{z,\mathrm{TCAV}}(\Delta z) - \rho_{z,\mathrm{LiTrack}}(\Delta z)\right]^2 d\Delta z, \quad (2)$$

E and z spread distributions were compared to create cost values:

$$C = w_E C_E + w_z C_z.$$

Iterative Extremum Seeking Algorithm (Updated)

$$Cost = \sum [\rho_{XTCAV} - \rho_{LiTrack}]^2$$

ES performed via finite difference approximation

$$\frac{\mathbf{p}(t+dt)-\mathbf{p}(t)}{dt} \approx \frac{d\mathbf{p}}{dt} = \sqrt{\alpha\omega}\cos(\omega t + kC(\mathbf{p},t)),$$

by updating LiTrack model parameters

$$p_j(n+1) = p_j(n) + \Delta\sqrt{\alpha\omega_j}\cos\left(\omega_j n\Delta + kC(n)\right),$$

where the previous step's cost is based on the previous simulation's parameter settings

$$C(n) = C(\mathbf{p}(n)).$$

Parameters must be tuned to optimize performance of ES

α - dithering amplitudes

ω −dithering frequencies

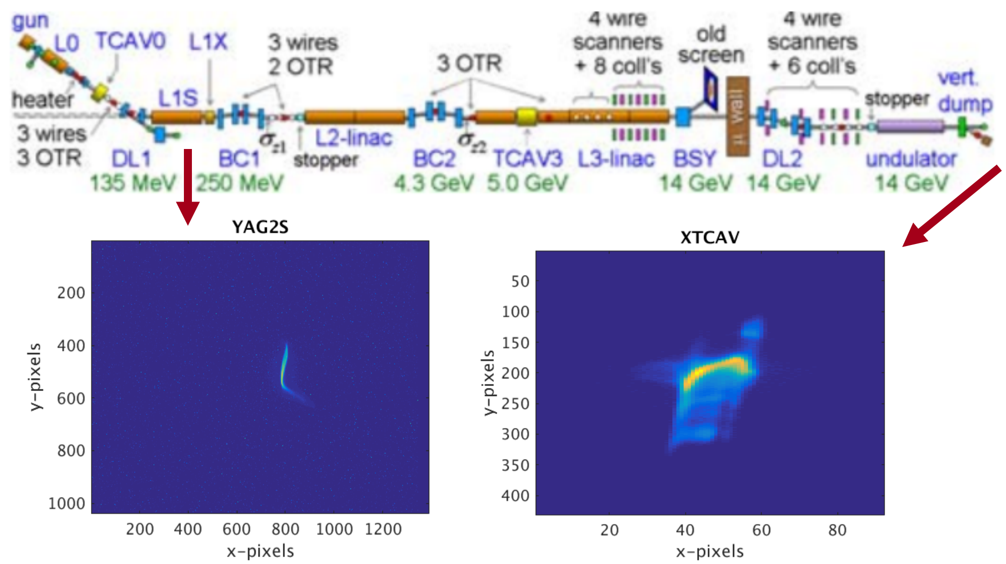
k - feedback gains

$$dt = \frac{(2\pi)}{\max(\omega)}$$

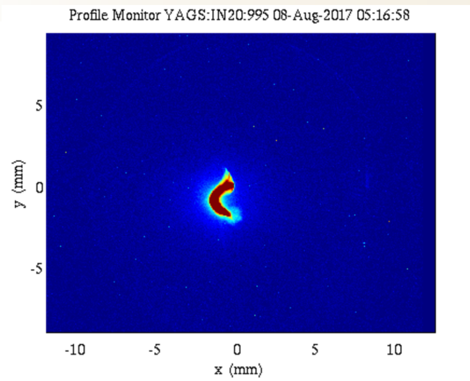
Benefits of ES:
(1) Robust for time varying system

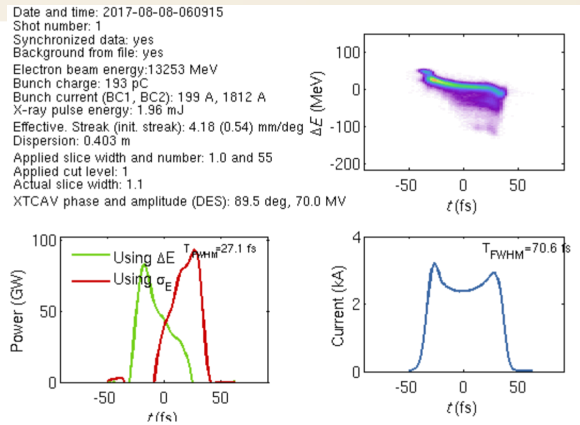
Previous Shift Plan/Results of Shift – 8/8/17

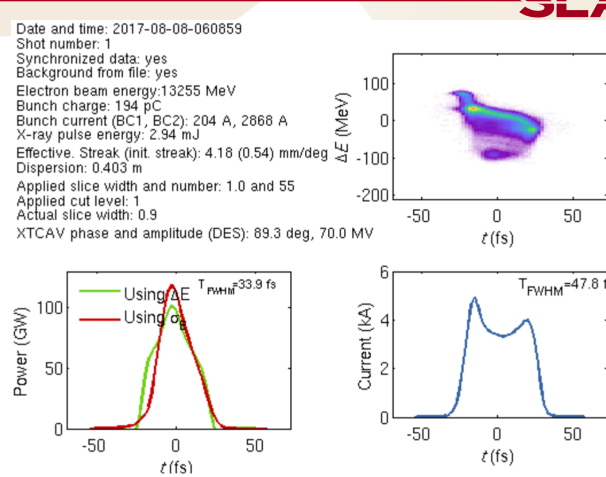
- Not much corresponding YAGS2/XTCAV Data
- Goal to obtain more complete data sets
- Take YAGS2 data w/ bunch length calibrations
- Take XTCAV data at various energies (w/ short, med, longer bunches)
    - 9.5 KeV, 6 KeV, 4 KeV
    - 550 eV, 700 eV, 900 eV

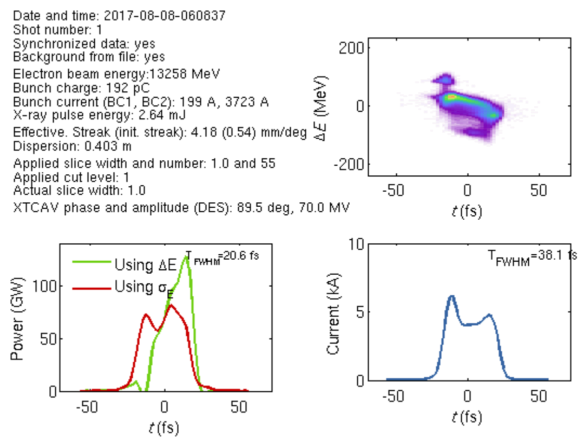


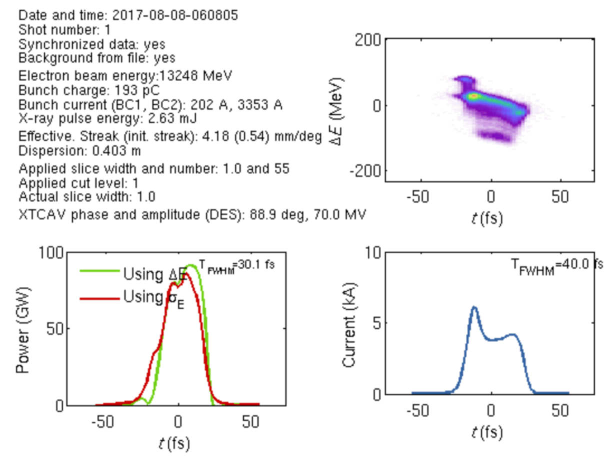
Some Shift Data











Future Plan

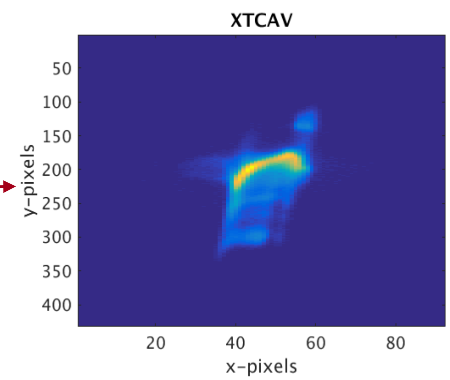
Future Work

- Verify / study how well the simulator is working and identify the limits of the simulation
- Look at how each input changes wrt small changes and make sure model is consistent
- Build catalog for standard injector setups for SAB Data?
- Compare LiTrack output to more diagnostics (e.g. Blen monitors, OTR)
- Compare Optimization Algorithms ES, GD, GA for convergence, speed, robustness…

Exploring Neural Networks

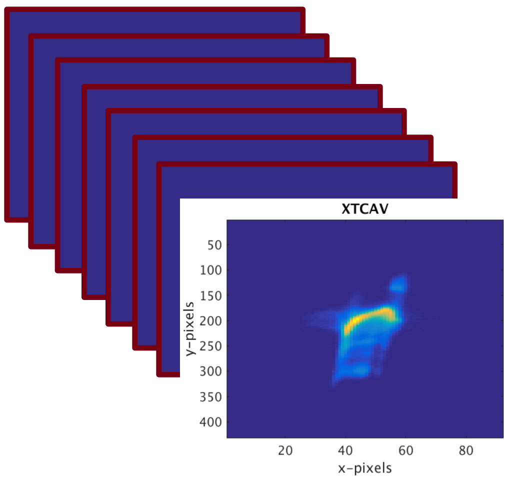
Machine Parameters

Simple Neural Model



Principle Component Analysis

1048 XTCAV Images 2014-16



Size of images: 294 x 299

1048

87906

Reduce Dimensionality Using PCA

20

50

Represent data using 1000 components

PCA Inverse Transform

Description of PCA

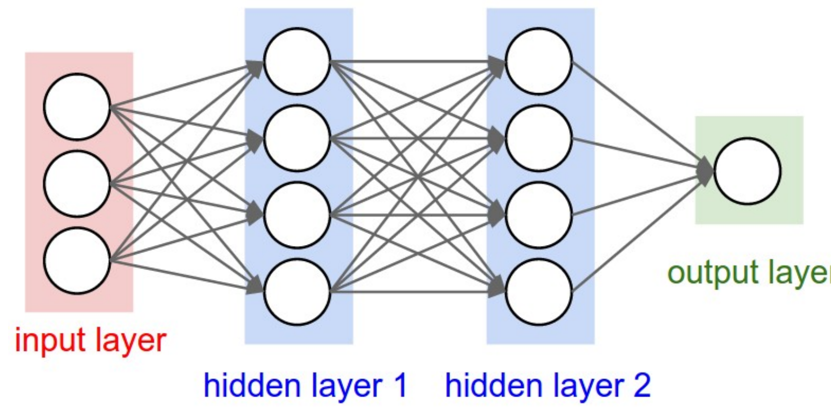
Three main steps:

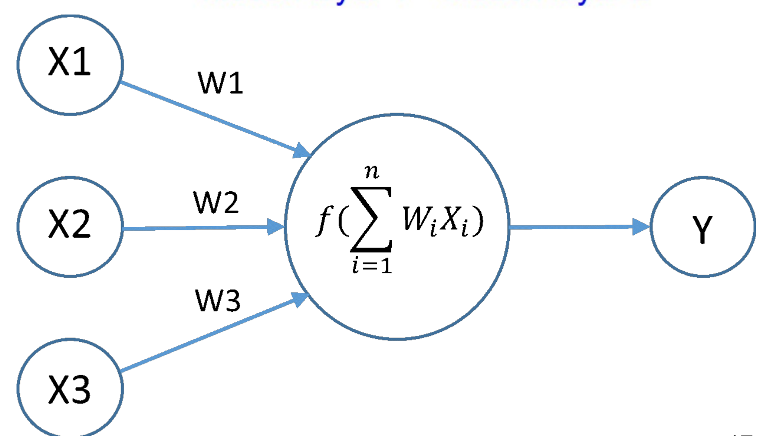
- Eigen decomposition
- Selecting Principal Components by sorting the eigenpairs
- Projection onto a new feature space

Modeling Accelerators: Artificial Neural Networks

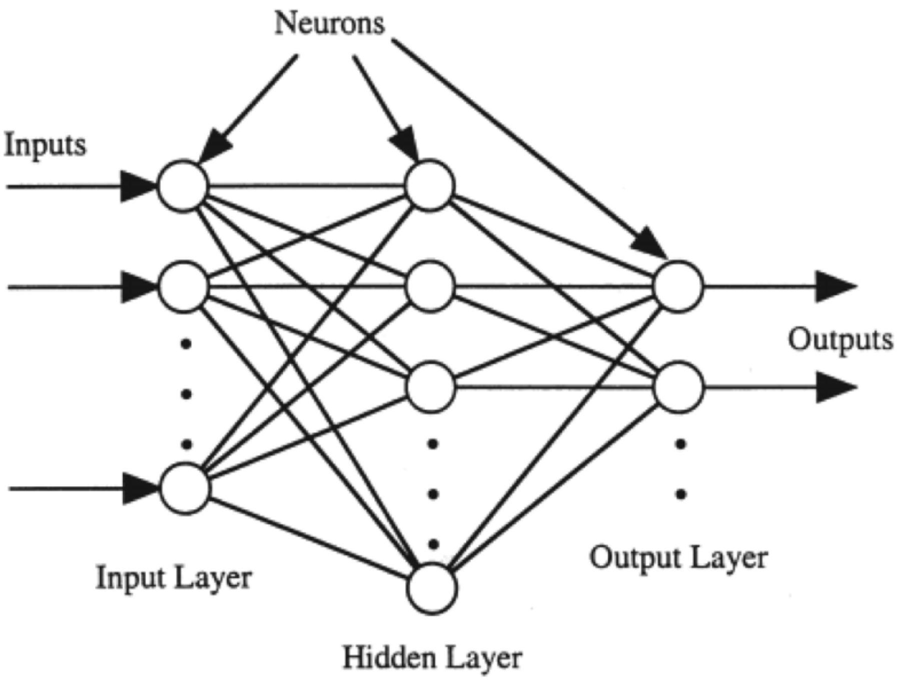
Has artificial neurons that mimic the human brain to learn about data

- A type of machine learning:
  - Supervised
- Use training data to generate weights that are then used to predict the data you're interested in.
- Activation functions can be used at each layer of neurons to learn nonlinear trends in the data



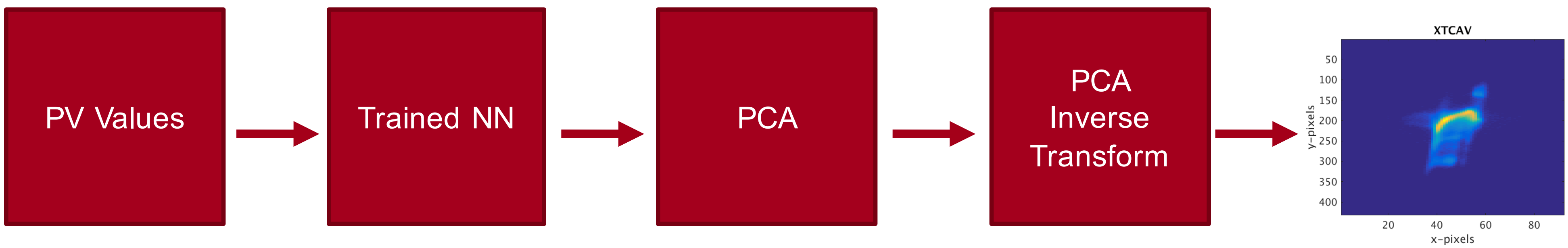


Graphic of Network

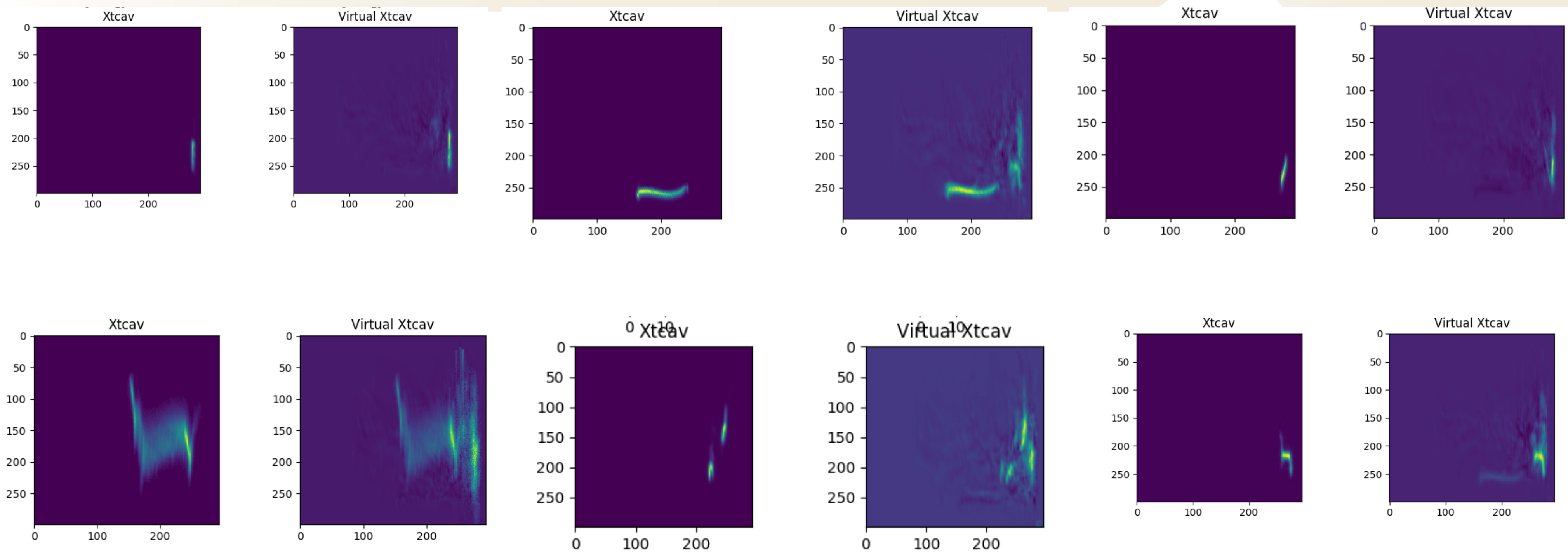


11 PVs 1500 nodes 1000 PCA features

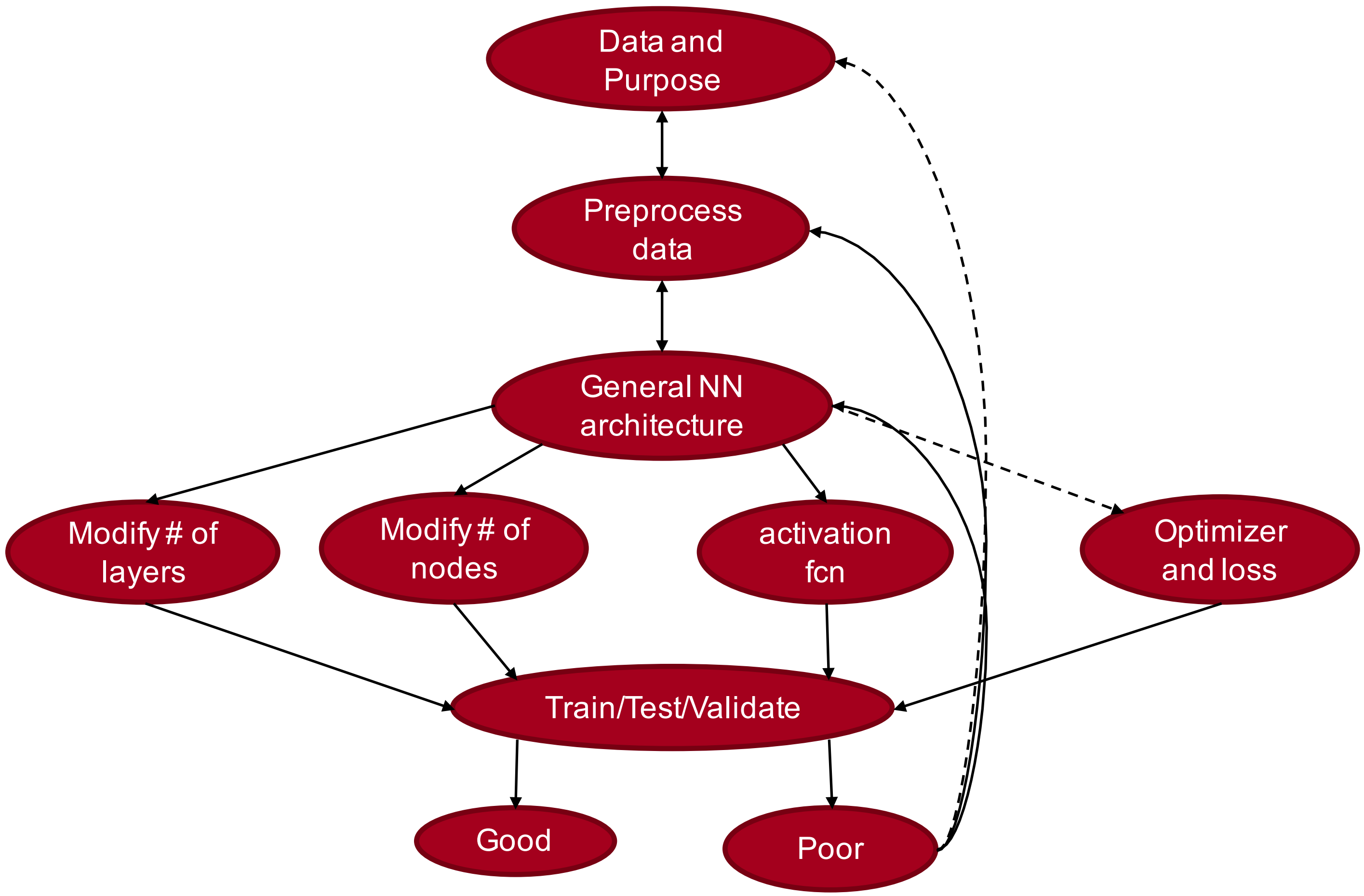
L1SE (GeV)	L1Sphi (deg)	L1XE (GeV)	BC1 R56	BC1E (GeV)	L2E (GeV)	L2phi (deg)	BC2E (GeV)	L3E (GeV)	L3phi (deg)	DL2E (GeV)


PV Values
Trained NN
PCA
PCA Inverse Transform
XTCAV
y-pixels
x-pixels
50
100
150
200
250
300
350
400
20
40
60
80

Preliminary Data



Next Steps …



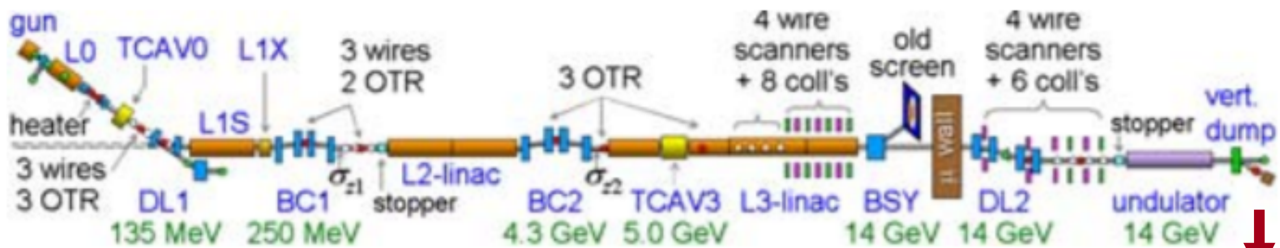
Data and Purpose
Preprocess data
General NN architecture
Modify # of layers
Modify # of nodes
activation fcn
Optimizer and loss
Train/Test/Validate
Good
Poor

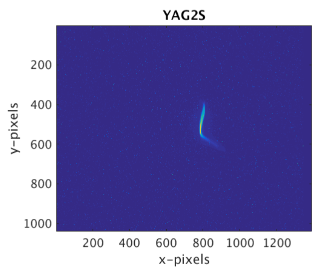
Scheme for a virtual diagnostic?

Trained NN

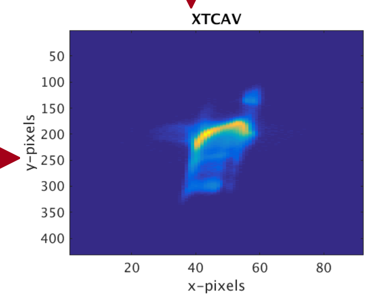
Inj long profile / pulse stacker

Injector PV Values





LiTrack



Virtual Diagnostic = Neural Networks may be use to predict longitudinal distribution before L1S which serves as an input into LiTrack or other physics models. Using the Extremum Seeking (or other optimization) routine we can then feedback back on the XTCAV output. Using a back calculation feature would allow us to compute the Longitudinal distribution anywhere in the machine. Could add many tuning automation tools to this framework.

Feedback with XTCAV