Signal Decomposition AI/ML Progress

Mario Cromaz, LBNL

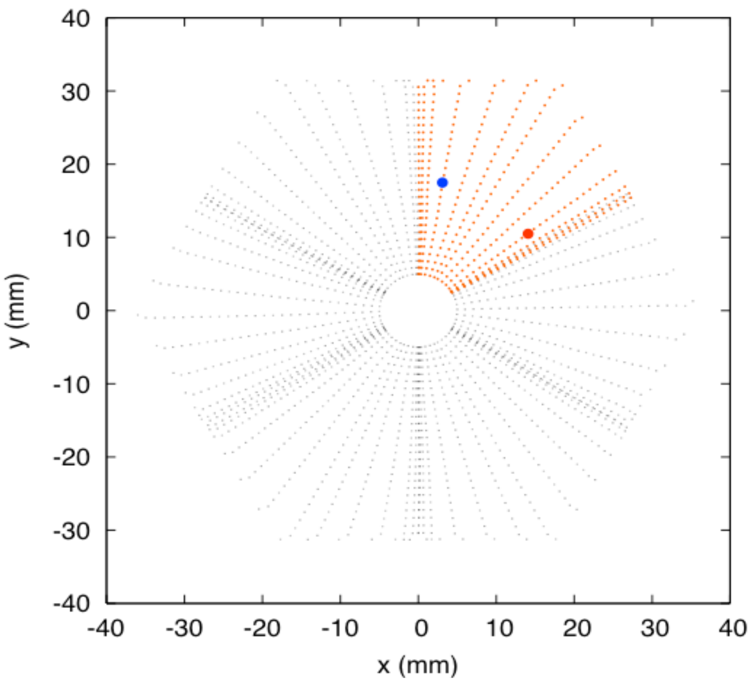
4th AGATA-GRETINA/GRETA Collaboration Meeting, Nov. 20, 2024

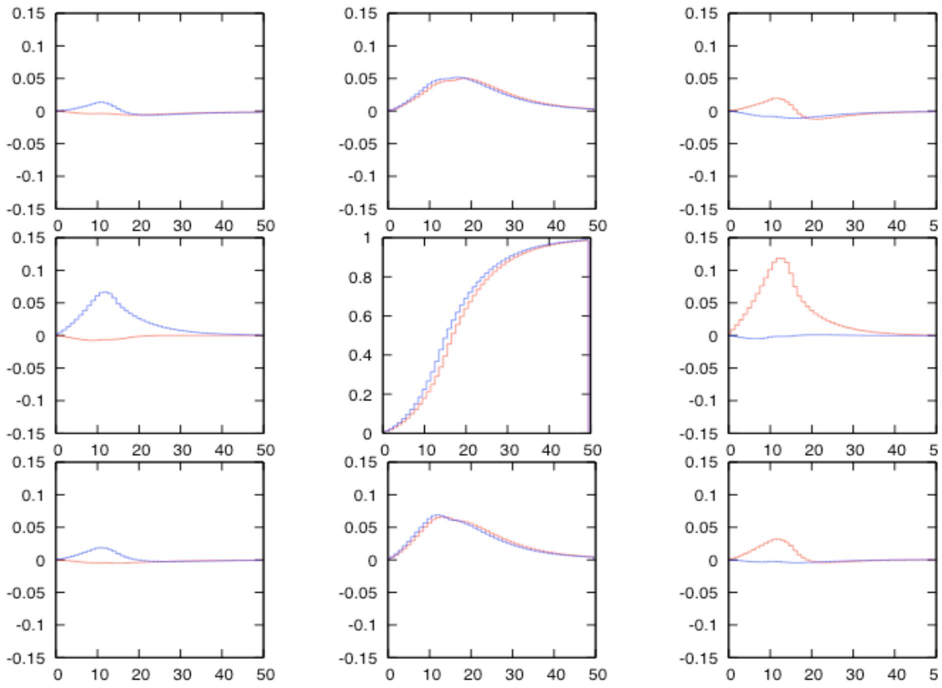
Work supported under contract number DE-AC02-05CH11231.

Outline

- Outline GRETINA/GRETA approach to the signal decomposition problem
- Plan to update software infrastructure to support ML workflows
- Two approaches to applying ML to signal decomposition:
  - Improve parametrization of simulation:
    - Electronic response
    - Crystal modelling
  - Direct application of ML models to locate interaction points
- Labelling/formatting facility at remote HPC using streaming framework

Sub-segment Position Determination

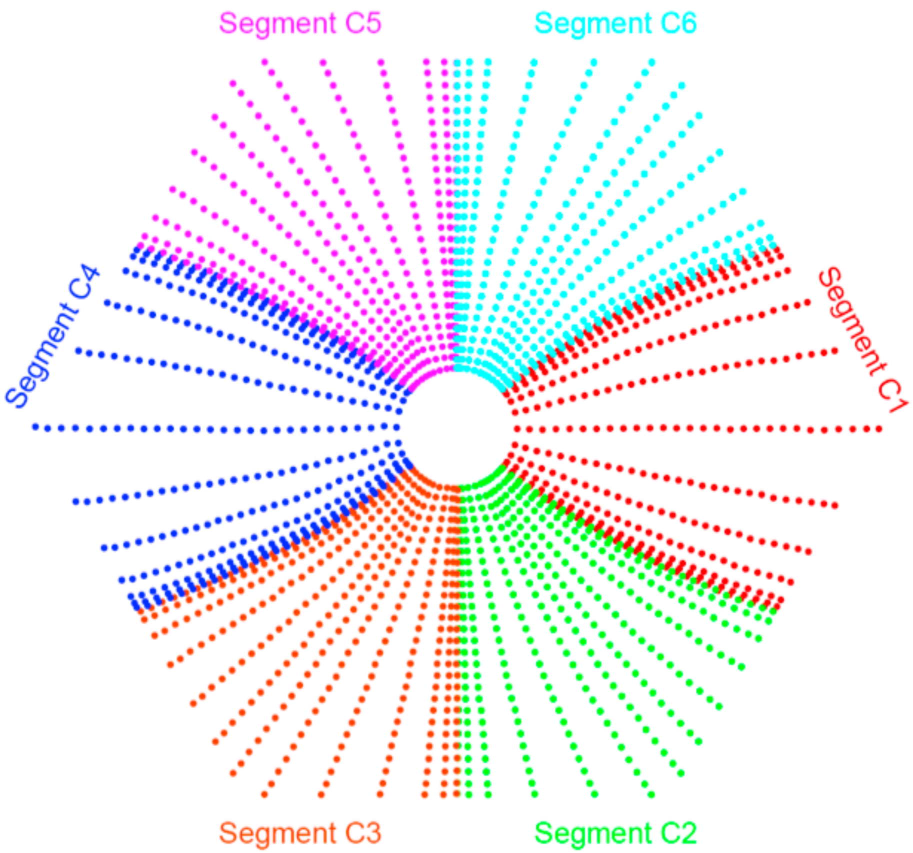




- Net charge on a segment pad identifies the interaction point location to within a given segment / voxel
- Waveforms from a net charge segment and neighbors (with induced signals) locate interaction points with sub-segment position resolution

Simulate Signals on Regular Grid

- Calculate electric field, weighting potentials based on crystal geometry, bias voltage, impurity concentration of each crystal
- Generate a grid (sensitivity weighted) and simulate charge collection on each of the segment pads for unit charge on each grid point
- Compare simulated signals against those measured experimentally - best fit gives grid point location



Grid point on equi-sensitivity grid in x-y plane
(D. Radford, K. Lagergren)

Multiple Interaction Points

- Roughly half the time, 2 or 3 interaction points per segment (dependent on gamma-ray energy)
  - And worse, the number of interactions is not known a priori
- Net charge signals from adjacent segments overlap with image charge signals
- Must fit linear combinations of simulated signals against experiment signal s

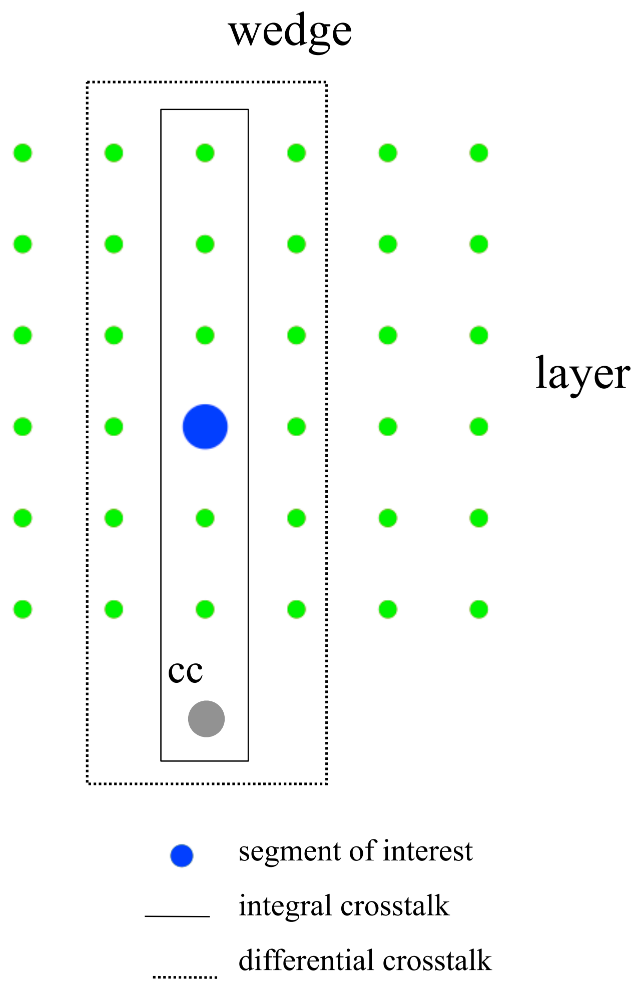
$$\mathbf{s} = e_1\mathbf{s}_{r1} + e_2\mathbf{s}_{r2} + e_3\mathbf{s}_{r3} + ..$$
$$\mathbf{s}_{rn} \equiv \mathbf{tr}_{net} \mid \mathbf{tr}_{n1} \mid \mathbf{tr}_{n2} \mid \mathbf{tr}_{n3} \mid \ldots$$

tr is 50 samples (500 ns)

- Grid spacing ~ 1mm
  - 230000 basis elements $\mathbf{s}_{rn}$ per crystal (6500 / segment)
- ~5 ms / crystal / cpu core
- In GRETA/GRETA, we refer to this as the signal decomposition problem

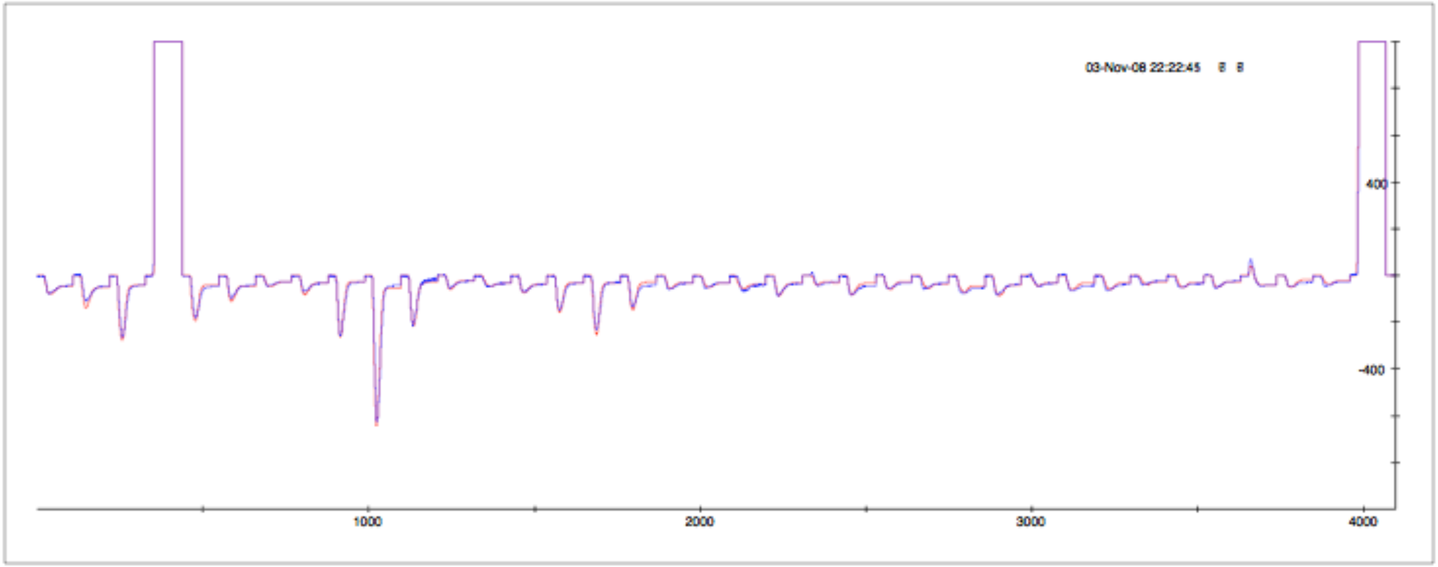
Electronic Response Corrections

- To compare experimental signals to simulation, one needs to correct for (and therefore determine):
  - Integral crosstalk
  - Differential crosstalk 😞
  - Preamplifier shaping
  - Delays between channels
- Only a subset of all possible crosstalk parameters are required
- But still … hundreds of parameters needed / detector - 669 in our current setup .. up from ~3 in non-tracking arrays
- Many of these parameters, specifically differential crosstalk, one cannot directly measure



Superpulse Method

- Use the fact that signals in each segment have the same electronic crosstalk
- Generate averaged traces from (1) experiment and (2) simulation - with proper weighting of interaction points
- Fit simulated averaged traces with response model applied (includes integral and differential crosstalk, relative delays between channels and preamplifier shaping) against experimental averaged traces to determine parameters in response model.

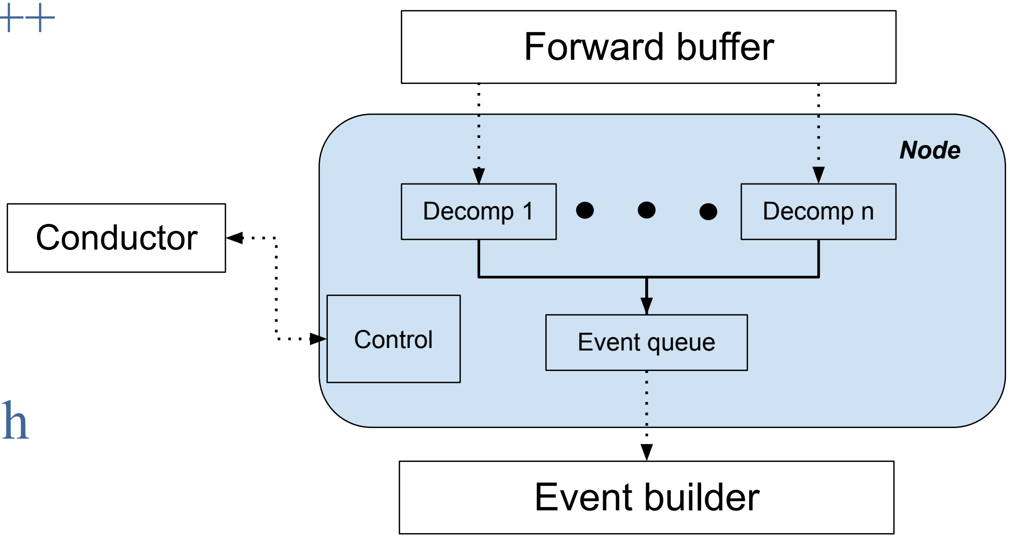


Current Signal Decomposition Algorithm

- Two phase algorithm employed:
  - Grid Search
    - 2-interaction search in a segment
    - Exhaustive course grid search followed by local refinement step
    - Multiple hit segments fit by iteration
  - Gradient Descent
    - Allows fit to go off grid points
    - Fit up to 3 interactions per segment when appropriate
    - Fit time offsets of traces

Refactored Signal Decomposition

- GRETA employs refactored C++ code (OLCF/ORNL)
  - Directly compatible with GRETINA/petcat implementations
  - GPU support for grid search
  - Can run on large scale computing facilities (Deleria project) - valuable for training
- Framework allows other single decomposition algorithms to be explored with minimal coding changes



Block diagram of the node client and associated node runners (labeled Decomp i). All node runners write to a common event queue which sends data to the event builder.

Modernizing Basis Generation Workflow

- Reworking toolchain used to generate cross-talk corrected basis
- Aim is to make algorithm choices less hard-coded - more amenable to experimentation
- Currently implementing Python-based response correction code
- Adopt new field generation code - more modern, flexible
- Take advantage of opportunities for trivial parallelization

```
669 parameter values:

delay0:
 20.61  20.67  19.48  19.67  19.90  20.07
 19.64  20.60  19.32  19.54  19.73  19.31
 19.29  19.71  19.05  19.34  19.80  20.04
 19.03  19.11  18.61  19.35  19.36  19.29
 19.00  18.66  18.04  18.63  19.33  19.63
 18.05  17.44  17.00  17.85  18.78  18.95

delay1:
 -2.87  -2.87  -2.52  -3.95  -2.63  -2.17
 -1.73  -2.83  -1.69  -1.46  -1.79  -0.64
 -1.00  -1.80  -1.69  -2.09  -2.12  -1.21
 -0.85  -0.92  -1.26  -1.59  -1.51  -0.42
 -1.45  -1.96  -2.04  -1.55  -1.23  -0.26
  0.56  -1.26  -1.46  -1.10  -0.47  -0.27

dxt  0:
dxt  1:   0:  0.144
dxt  2:   1:  0.245
dxt  3:   2:  0.458
dxt  4:   3:  0.488
dxt  5:   0:  0.082   4:  0.217
dxt  6:   0:  0.423   1:  0.012   5: -0.017
dxt  7:   0: -0.054   1:  0.260   2:  0.051   6:  0.015
dxt  8:   1:  0.050   2:  0.399   3:  0.286   7:  0.074
dxt  9:   2:  0.040   3:  0.911   4:  0.059   8:  0.079
dxt 10:   3:  0.312   4:  0.463   5:  0.070   9:  0.106
dxt 11:   0: -0.055   4:  0.010   5:  0.469   6:  0.039  10:  0.125
dxt 12:   0: -0.007   1:  0.003   5: -0.026   6:  0.224   7: -0.001  11: -0.033
```

xtalk_pars_in.txt

ML to improve Parametrization of Crystal Simulation

- Currently apply superpulse method to derive electronics response parameters
- When new basis generation toolchain in place we want to:
  - Experiment with improved electronic response corrections
  - Include crystal simulation parameters (charge carrier mobilities, temperature)
- Use standard hyper-parameter optimization methods/packages in ML (Ray Tune)

- Computationally expensive - recalculate basis on each iteration
- Benefits from improved (parallelized!) workflows

Direct Application of ML models

- Use ML model (3D CNN likely starting point) to directly infer location and energy of interaction points
- Use simulation to train base model - experimental data to refine

- ML models could be potentially be used for other (potentially more tractable) observables to aid/constrain standard signal decomposition procedure
  - Number of interaction points (constrains standard signal decomposition procedure)
  - Veto of anomalous events - for example, those that sample a specific part of the crystal that is poorly modeled

- Engage ScienceIT / NERSC ML groups at LBNL

Labelling / Formatting Facility

- General ML methods will require large amounts of training data
- Adapt GRETA/Deleria streaming framework to send data to HPC to be labelled, formatted and stored
- General purpose schema for exposing labels in data and metadata (work with SciData group at LBNL)
- Mechanism for high performance ingest into ML framework (Torch + friends)
- Goal - store all the data … looking towards future facilities like HPDF

Summary

- Much of current effort is aimed towards modernizing GRETINA/GRETA software infrastructure so it is amenable to ML work

- Currently working on applying ML techniques to improving parametrization of crystal and electronics response

- Once GRETA complete we want to pursue direct application of ML models to signal decomposition

Thanks to!

Arin Manohar, Gustav Jansen, Marco Salathe, Eric Pouyoul, Augusto Macchiavelli, David Radford, Paul Fallon, Heather Crawford, Chris Campbell