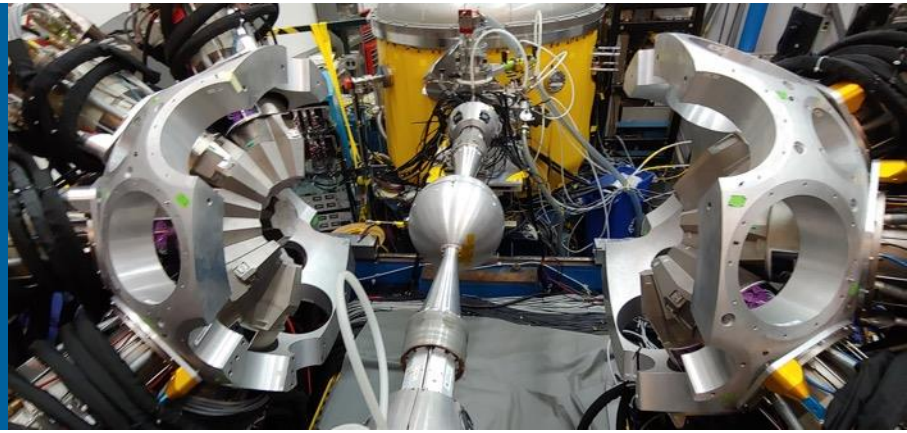
PRESENTATION TO NUCLEAR PHYSICS AI AND DATA SCIENCE, PI EXCHANGE MEETING

Argonne NATIONAL LABORATORY

DEVELOPING MACHINE-LEARNING TOOLS FOR GAMMA-RAY ANALYSIS



MICHAEL CARPENTER, SCOTT CARMICHAEL TORBEN LAURITSEN, AMEL KORICHI, FILIP KONDEV, MARCO SICILIANO
Physics Division
Argonne National Laboratory

THOMAS LYNN, DAVID LENZ, SVEN LEYFER, ROB ROSS, ROB LATHAM
Mathematics and Computer Science Division
Argonne National Laboratory

U.S. DEPARTMENT OF ENERGY Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

November 19, 2025

Advancing Nuclear Structure Science Through Machine Learning and Modern Data Architectures

Modern γ-ray arrays (Gammasphere, GRETA/GRETINA, AGATA) produce increasingly complex, high-fold coincidence data. Extracting physics from this data is labor-intensive and often slow.

We are developing a unified ML-driven analysis ecosystem that:

- Enhances γ-ray tracking with learning-to-rank models
- Automates level-scheme construction using inverse optimization & ML
- Enables fast, scalable data access with Arrow/Parquet + Roaring Bitmaps
- Integrates into the GRayMAN platform for calibration, fitting, and analysis

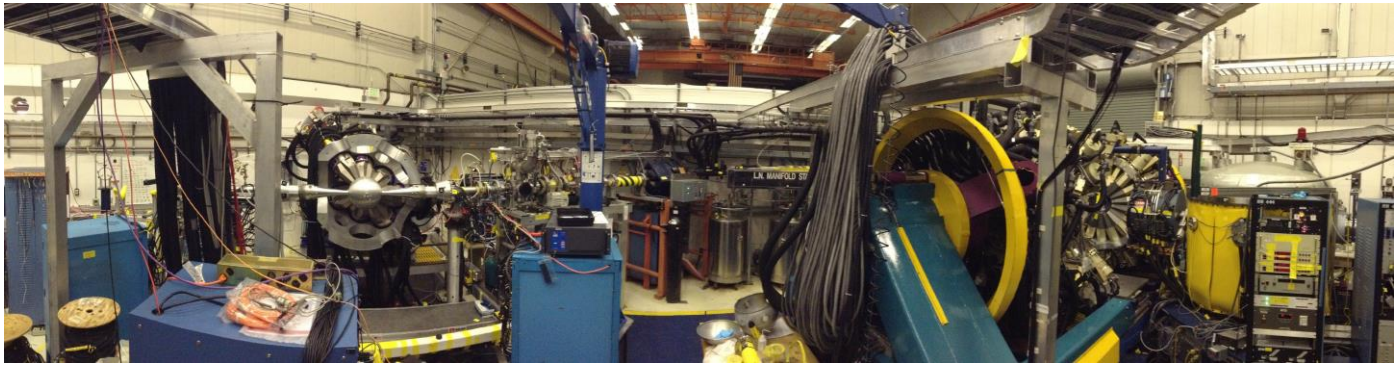
Joint effort between ANL Physics and MCS Divisions and IJCLab (Orsay)

PHASE I: PROJECT PURPOSE AND GOALS

The **purpose** of phase I is to develop automated decision-support tools to assist physicists in the analysis of complex experimental data taken with the large gamma-ray spectrometers (Gammasphere, GRETINA and AGATA).

Goals:

1. Develop machine-learning tools to improve γ-ray tracking (GRETINA/GRETA).
2. Develop machine-learning tools to assist in the construction of complicated level schemes using γ-γ and γ-γ-γ coincidence data.



PHASE I/II - OUTLINE

Machine-Learning (ML) tools for Gamma-Ray Analysis

Gamma-ray Tracking

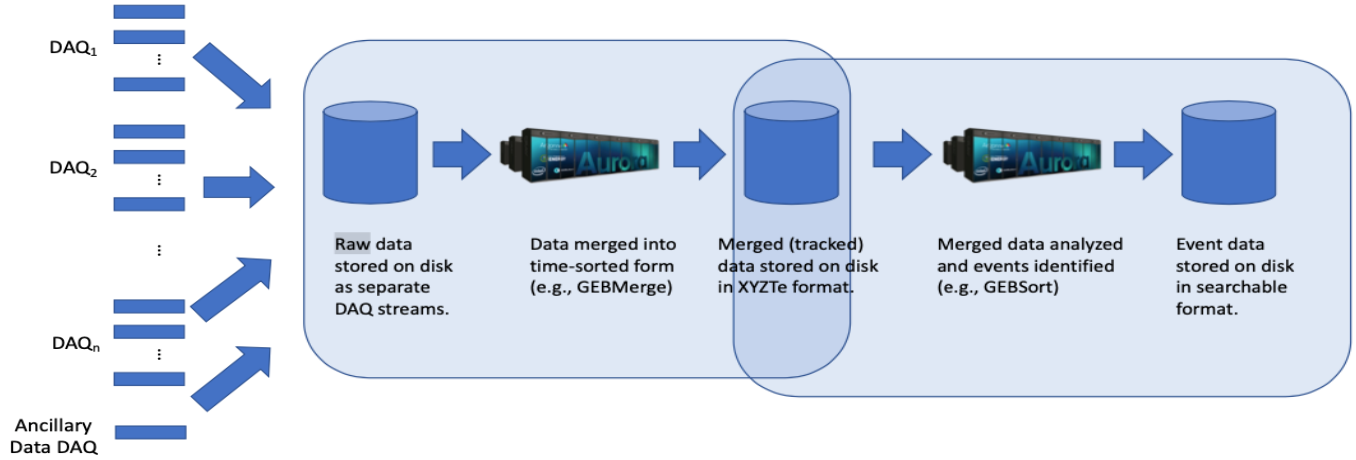
- Develop new methods to improve on current gamma-ray tracking algorithms to increase both photopeak efficiency and background rejection.
- Utilize machine learning tools to improve on these methods.
- Extend these methods to include pair production events.
- Incorporate these tools into tracking codes used by the community.

Level Scheme Construction

- Develop a mathematical toolkit to build levels schemes using both 2-fold and 3-fold coincidence information bench marking with known level schemes.
- Develop tools to automatically extract intensity information from gamma-ray coincidence data (2D, 3D).
- Apply toolkit to both simulated data and experimental data taken with Gammasphere and GRETINA.

PHASE II EXPANSION: HIGH-PERFORMANCE DATA ARCHITECTURE FOR γ-RAY ANALYSIS

HPC Tools for Gamma-Ray Analysis



Implement a modern, scalable data backend using Apache Arrow + Parquet:

- Zero-copy in-memory event processing (Arrow)
- Efficient compressed on-disk storage with schema evolution (Parquet)
- Storage-side filtering for fast scans of high-fold coincidence data

PHASE II - ADDITIONS

Optimization and ML tools for Coulomb excitation

$$\min_M \chi^2(M) := \frac{1}{N} \underbrace{\sum_I w_I \sum_{k\in I} \left(C_I Y_k^c(M) - Y_k^e \right)^2 / \sigma_k^2}_{\text{Coulomb } \gamma-\text{yields}, S_y} + \underbrace{\sum_j \left(\frac{Y_j^c(M)}{Y_j^n} - u_j \right)_+^2 / u_j^2}_{\text{observation limits}, S_1} + \underbrace{\sum_i \frac{d_i(M) - d_i^e}{\sigma_i^2}}_{\text{auxiliary terms}, S_a}$$

We are investigating the use of modern machine-learning and optimization techniques to accelerate the least-squares optimization in GOSIA. Our developments will enable other outer loop analysis, such as the automatic selection of weights and the use of reinforcement learning techniques for the determination of matrix signs. (Leyfer and Siciliano)

This part of project has been abandoned due to the fact that a substantial effort is being undertaken in Europe.

PROJECT PARTICIPANTS

Joint project between two ANL divisions: Physics (PHY) and Math and Computer Science (MCS) and IJCLab Orsay

PHY

- Mike Carpenter (ANL Staff)
- Scott Carmichael (FOA Funded Pdoc)
- Filip Kondev (ANL Staff)
- Amel Korichi (IJCLab Orsay Staff)
- Torben Lauritsen (ANL Staff)
- Marco Siciliano (ANL Staff)
- Emma Weiller (IJCLab, Grad. Stud.)

MCS

- David Lenz (ANL Staff)
- Sven Leyffer (ANL Staff)
- Robert Ross (ANL Staff)
- Rob Latham (ANL Staff)

T. Budner and T. Lynn have completed their post-doc appointments and have moved on to Staff positions elsewhere.

BUDGET TABLE

Summary of expenditures by fiscal year (FY):

	FY21 ($k)	FY22 ($k)	FY23 ($k)	FY24 ($k)	FY25 ($k)	Total ($k)
a) Funds allocated	500	500	0	820	0	1,820
b) Costs to date	0	179	392	435	215	1,221

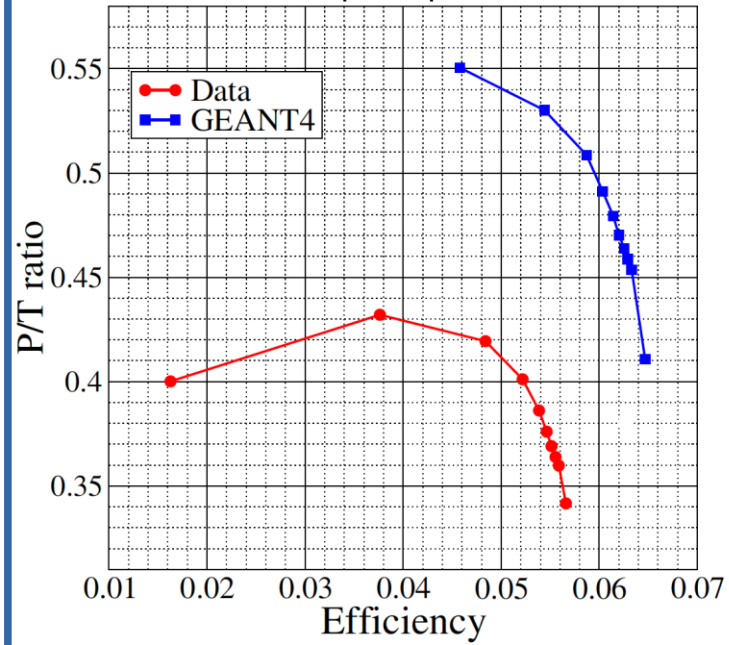
We had ~$428k remaining at the end of FY23. The remaining funds were due to delay in finding and hiring post-doctoral appointees until later in FY22. Both Post-Docs ended their appointments in FY24-Q4. This took care of funding from Phase I. We received funding for Phase 2 in FY24 and are working on the proposed deliverables. As of the start of FY26, we are funding ½ salary of PDOC Scott Carmichael on the gratn.

ML TOOLS FOR GAMMA-RAY TRACKING

PROJECT GOALS

Machine-Learning (ML) tools for Gamma-Ray Tracking

Curent tracking arrays (AGATA & GRETINA) do not meet the required performance



A. Korichi and T. Lauritsen, Eur. Phys. J. A (2019) 55: 121
AGATA-GRETINA Review paper

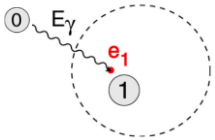
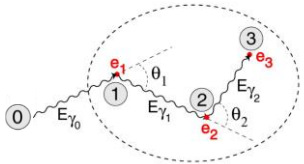
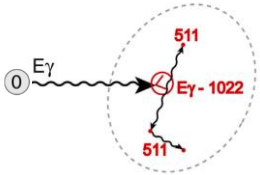
- Develop new techniques to enhance existing γ-ray tracking algorithms, boosting photopeak efficiency and improving the signal-to-background ratio (P/T).
- Adapt these techniques to accurately perform Doppler correction with the first interaction point (ordering!)
- Expand these methods to handle pair production events.
- Incorporate these tools into tracking codes used by the community.

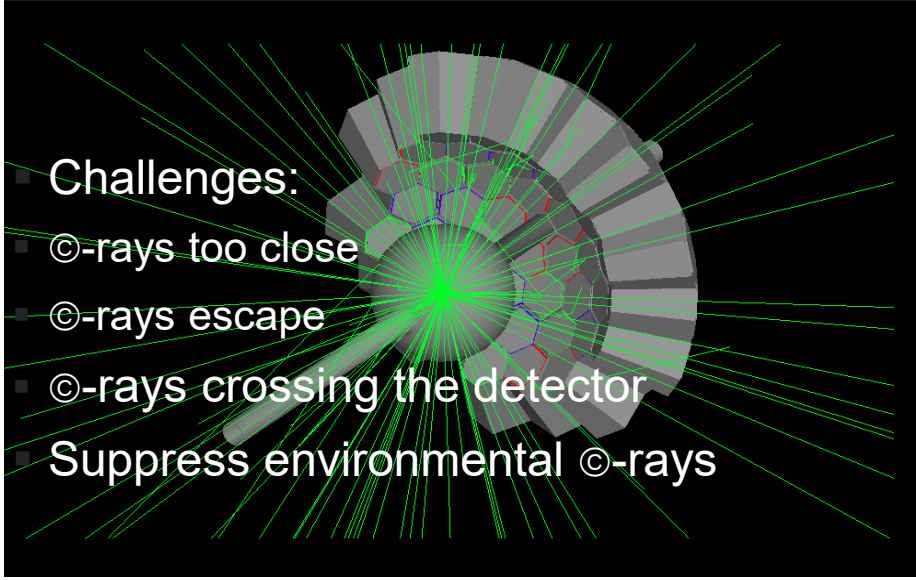
©-RAY TRACKING

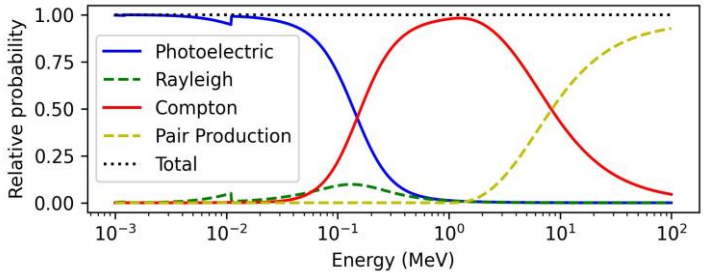
Overview of the principle

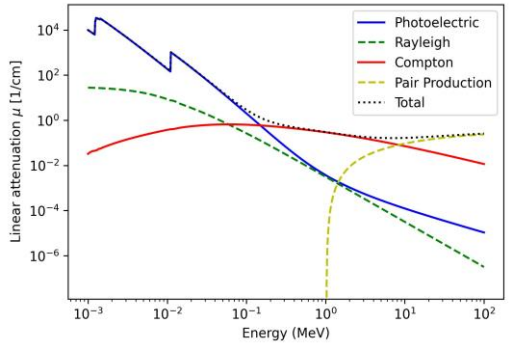
Three known interaction types of interest

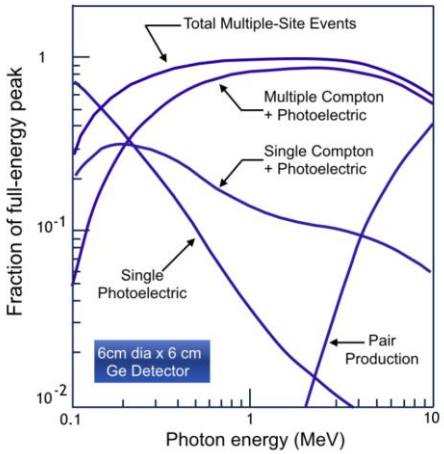
~ 100 keV ~1 MeV ~10 MeV γ-ray energy

Photoelectric	Compton Scattering	Pair Production
		
Isolated hits	Angle/Energy	Pattern of hits
Probability of Interaction depth	$E'_\gamma = \frac{E_\gamma}{1 + \frac{E_\gamma}{m_0 c^2}(1 - cos\theta)}$	$E_{1st} = E_\gamma - 2m_0c^2$









GRETO APPROACH

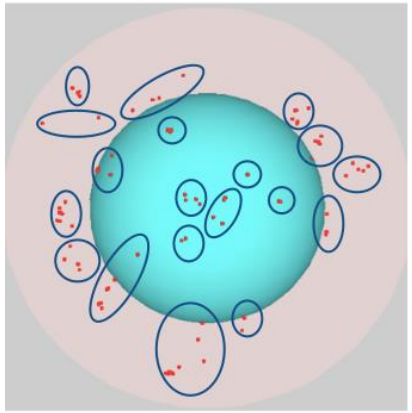
Cluster interactions into separate γ-rays

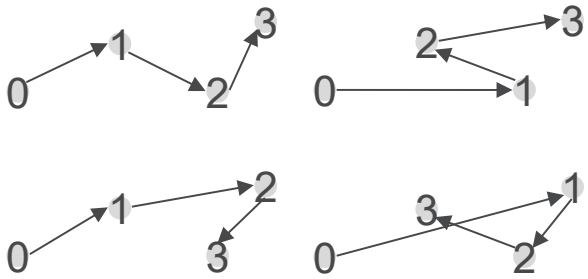
Order interactions for individual γ-rays

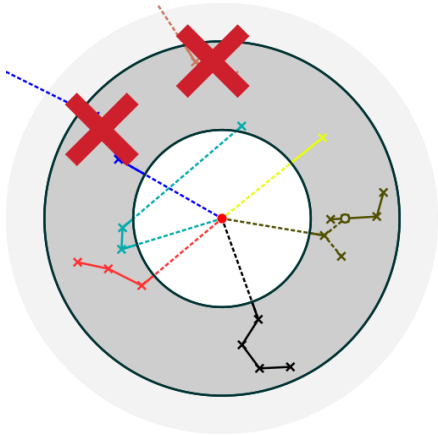
Suppress γ-rays scattering out of the detector

AFT (Argonne Forward Tracking)
AI/GRETO

AI/GRETO







GRETO approach

1. **Clustering**: AFT approach – each cluster is assumed to be from an individual γ-ray (often violated when many γ rays are present).

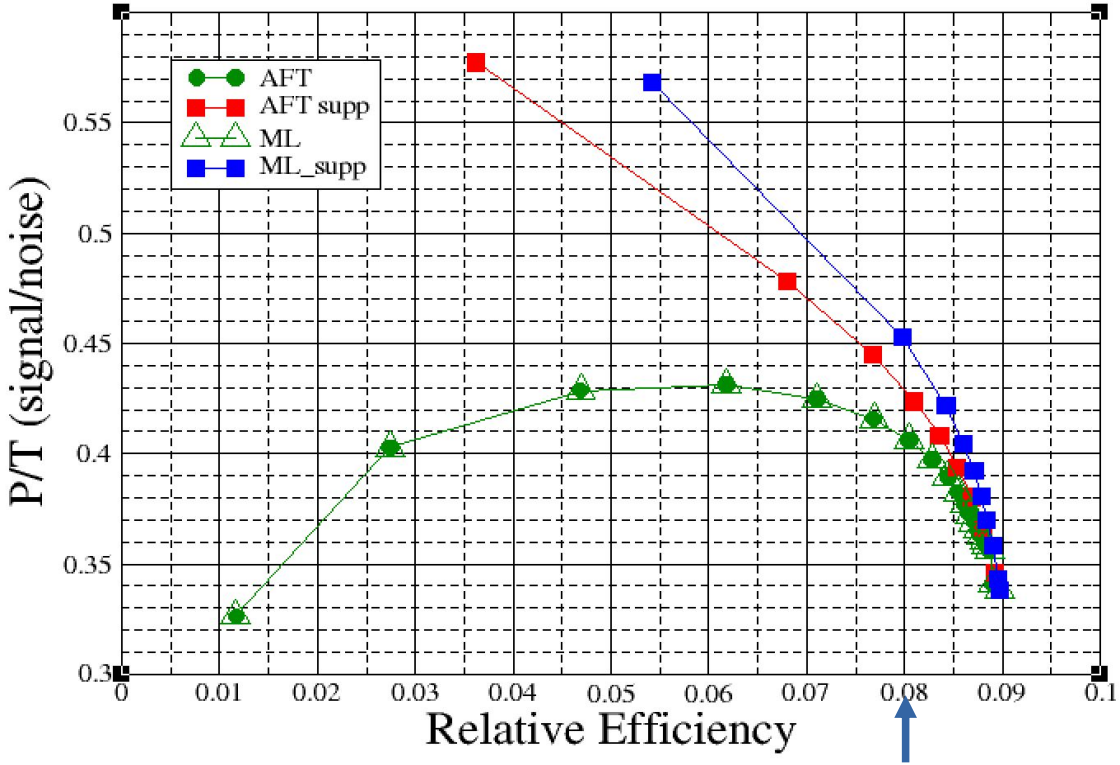
2. **Ordering**: A figure-of-merit (FOM) is used to select the "best" order by evaluating all (or most) possible orders of interactions within each cluster. γ–rays are assumed to completely deposit their energy in the detector and emitted energy is assumed to be equal to the energy sum of interactions. (LETOR – Learning to Rank)

3. **Suppression**: Any clusters with FOM values from ordering worse than some user defined threshold are filtered out from the final spectrum. The FOM measures disagreement with the CSF indicating non-physical scattering (bad cluster or missing energy).

Training performed with simulated data :G4 simulated data: 300k γ-rays from an M=30 rotational band (80 to 2600 keV) AGATA 4π geometry (with packing and smearing)

Several models for the ordering and suppression/Evaluation can be used

Results for ^{60}Co source data GT with only 28 crystals

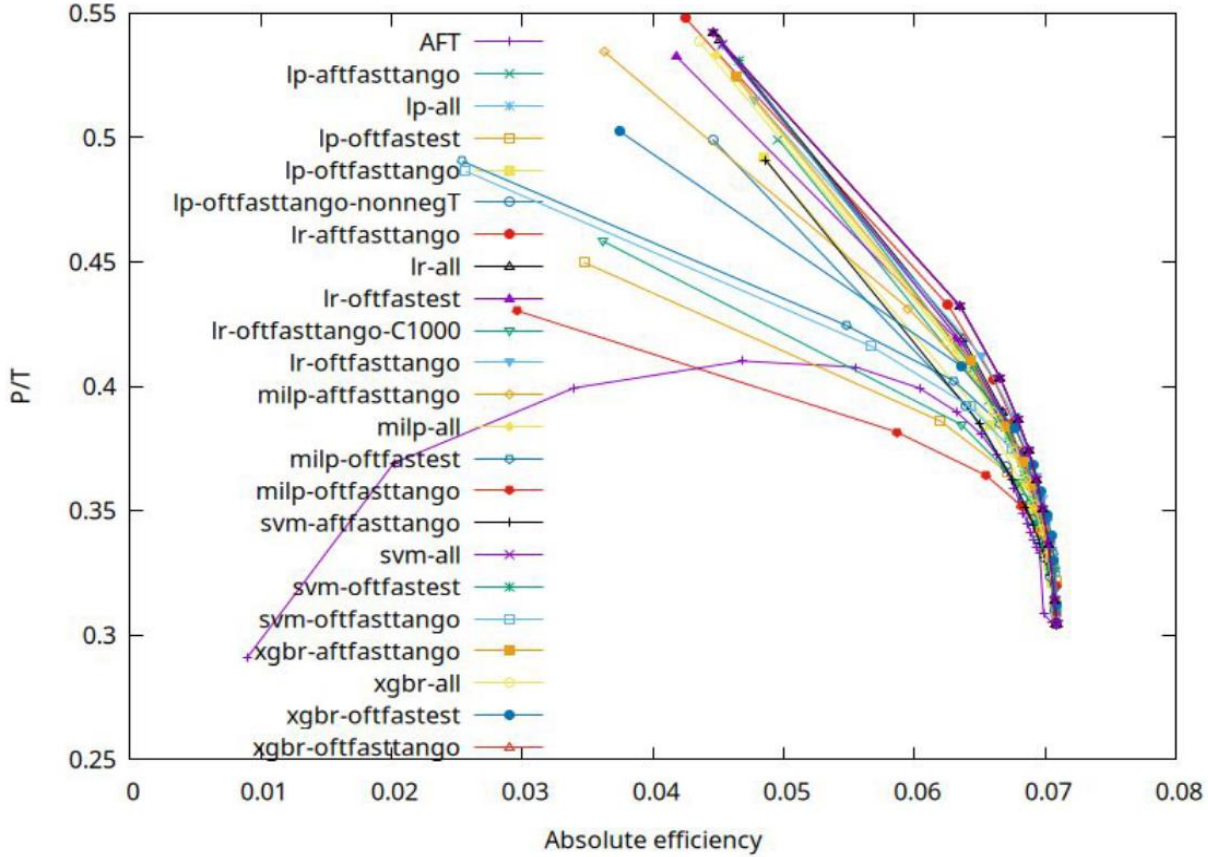


ML ordering + ML suppression

AFT +ML suppression model

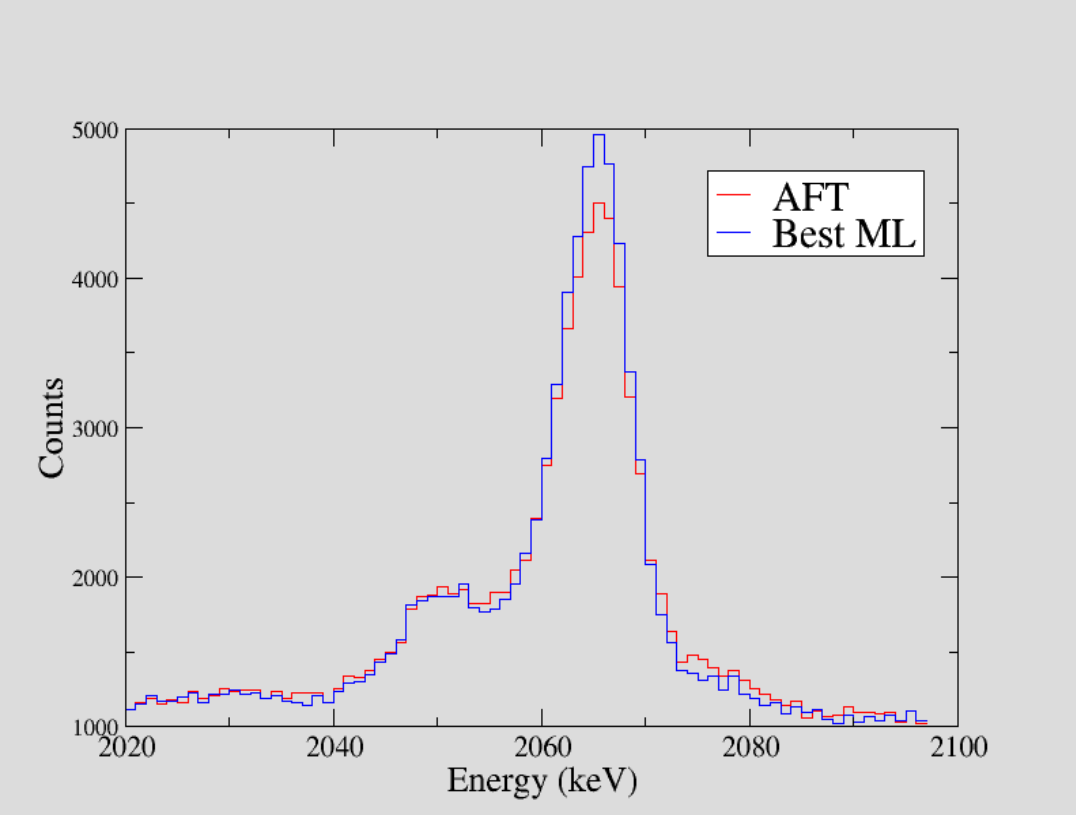
P/T for conventional FOM cut : AFT 40 % GRETO 45 %

More than 300 models currently under evaluation

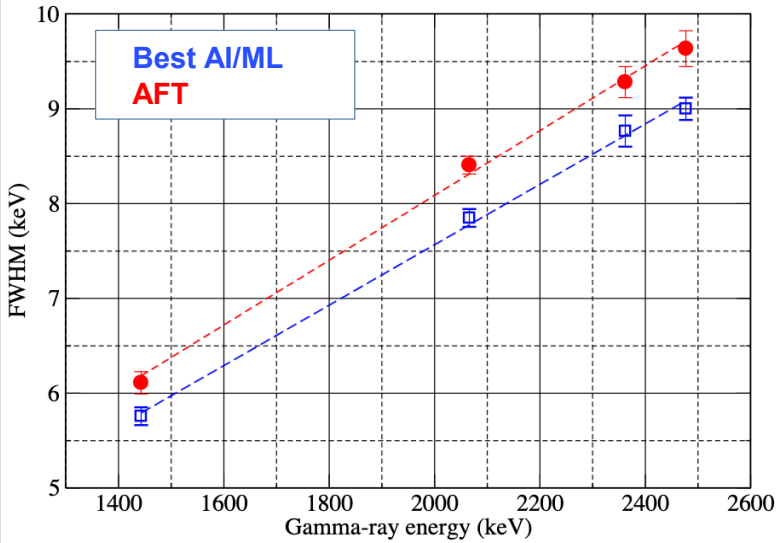


Emma Weiller's PhD student

Results for ^{92}Mo in-beam data



FWHM	Peak Area	Energy
8.02(6)	31763(266)	2065.63(4)
8.75(7)	30169(277)	2065.65(5)



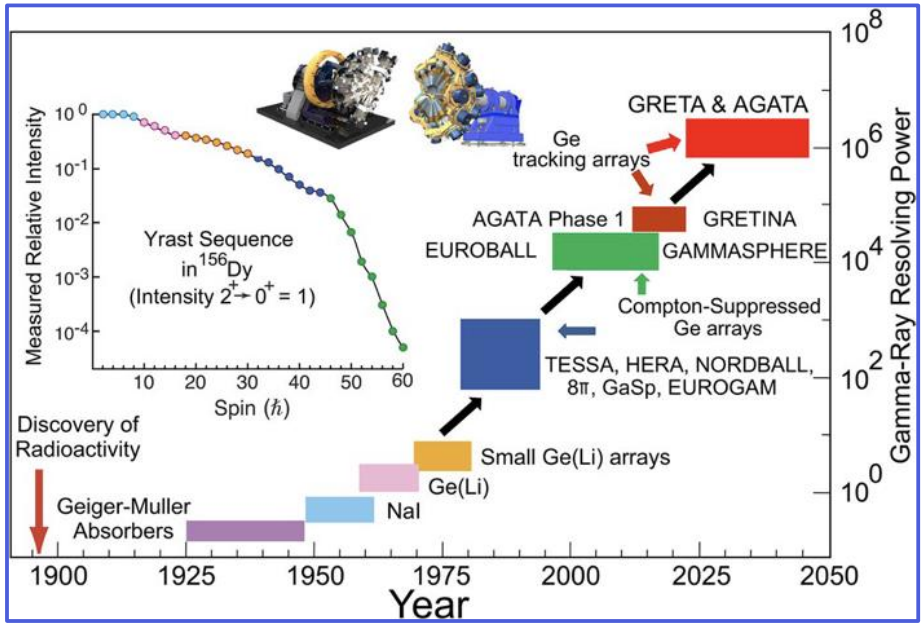
Clear improvement in the energy resolution & efficiency

FIGURE OF MERIT FOR THE EVALUATION OF A SPECTROMETER PERFORMANCE COMPOSITE PARAMETER WITH:

Total photopeak efficiency ε
Energy resolution FWHM
photopeak-to-total ratio P/T

$$R \sim \frac{\varepsilon\ {}^{TM}E}{FWHM}\ P/T$$

^{TM}E Average spacing between consecutive transitions in a typical cascade



Resolving Power(RP) ~ R^{Fold}

For a 5-fold ©-ray event
(typical for high-spin Gammasphere exp.)

10 %P/T better → increase RP by 60%

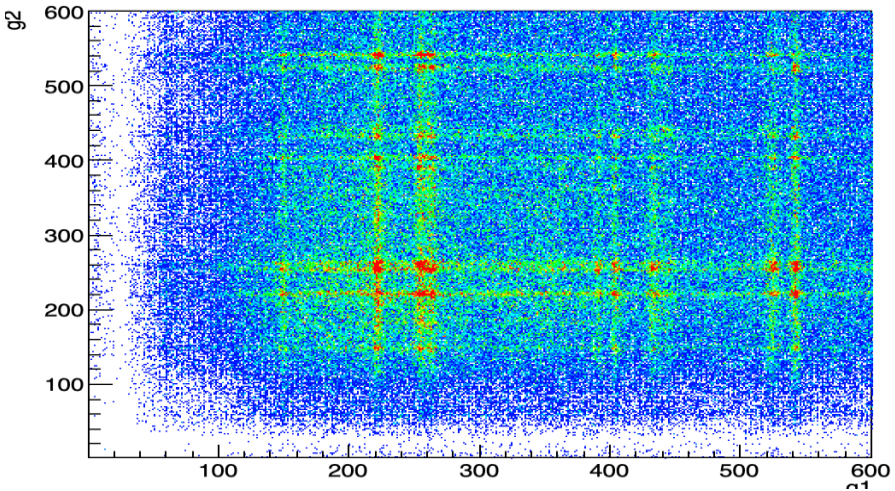
8 % fwhm better → increase RP by 52%

This results in more than a factor 2.5 gain in the Resolving Power

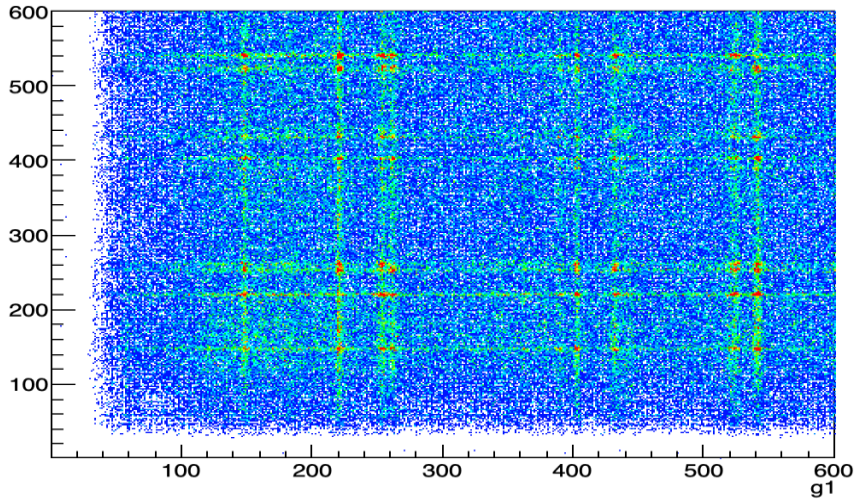
AFT vs GRETO

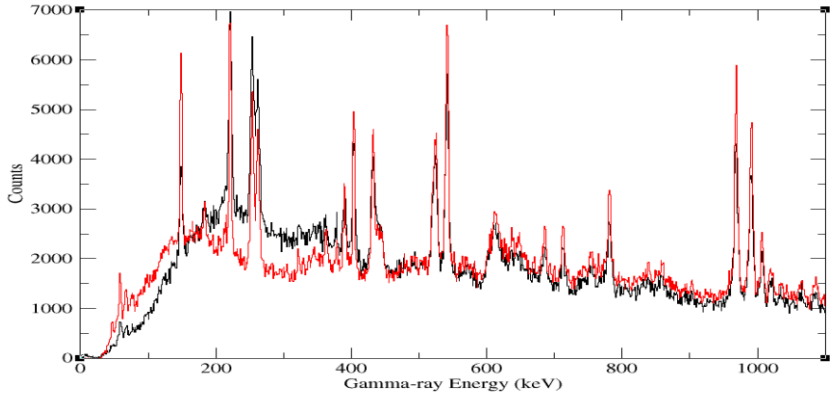
$^{108}Pd(^{48}Ca,4n)^{152}Dy$ @ 196 MeV - GRETINA 48 crystals

AFT



GRETO





These improvements of the performance results in a factor of 2.5 gain in the resolving power (fold 5).

AI/ML + new tools offers new physics opportunities with GRETA

GRETO is a very friendly user but slow

```
# Example settings for gamma-ray tracking
# There are more options available, but not necessarily useful to tracking.
# See greto/track_default.yaml for full options file

DETECTOR: AGATA                                  # Available detectors are "GRETINA", "GRETA" (identical to "GRETINA"), and "AGATA"
COINCIDENCE_TIME  40                             # Units of 10 ns; Used for grouping recorded crystal events in time
NUM_PROCESSES:   30                              # Number of processors to use for multi-processing (should be changed depending on the machine)
VERBOSITY: 2                          # Level of console output;
MAX_HIT_POINTS:   300                                 # Maximum number of hit points for constructed events (experiment/data dependent)
MONSTER_SIZE: &monster  8          # Maximum size of a cluster; clusters larger than the "monster" size are not tracked

REEVALUATE_FOM: false               # Reevaluate the FOM values according to the eval_FOM_kwargs; Used in conversion, not used for tracking
order_FOM_kwargs:                                # Arguments for ordering interactions (recorded in mode1)
  # fom_method: selected                         # FOM method for ordering; Available methods are "oft"/"agata", "aft"/"angle", "selected" (ML optimized for ordering), "model"
  # model_filename: null                         # Filename of the model
  fom_method: model
  model_filename: models/ordering/N2000_lr_nonnegFalse_C10000_cols-aft_fast_tango_width5.json
  width: 5                                                  # How many interactions to consider in the forward direction in the enumeration of possible tracks
  stride: 2                                                 # How many interactions to accept after each enumeration step
  max_cluster_size: *monster             # Maximum cluster size for tracking (should be the same as the monster size, but can be different)
eval_FOM_kwargs:                          # What FOM should be recorded in mode1
  # fom_method: angle                              # FOM method for recording in mode1
  # model_filename: null                           # Filename of the model
  # singles_method: depth                          # Singles method for recording in mode1;
                                                                  Available methods are "depth" (identical to AFT treatment), "range" (similar to AFT treatment
using a continuous function to determine ranges),                                                                  probability" (range probability is
returned), "continuous" (linear attenuation * distance)
  fom_method: model
  model_filename: models/suppression/N10000_sns-logistic_pca-0.95_order-model.pkl
  max_cluster_size: *monster                     # Maximum cluster size for tracking, determines if the gamma ray is considered tracked or not (should match the tracking options)
cluster_kwargs:                                  # How should interactions be clustered?
  alpha_degrees: 13                              # Cone clustering alpha in degrees
  time_gap: *tg                                  # Time gap for clustering; interactions with timestamps differing by more than gap p have the distance between them set to infinity
                                                                  #(can still be clustered together if there is an interaction bridging the time gap)
```

GAMMA-RAY TRACKING SUMMARY

The synergy/collaboration between the Physics Division, MCS Division and IJCLab(Orsay) has been crucial to the project's success.

Thomas Lynn

- **Promising results with GRETO**
  - Python Code has been published on GitHub
  - New ordering approaches enhance existing techniques, improving the resolving power by up to 2.5
  - Learning To Rank (LTR) methods enable expanded tracking optimizations
  - New suppression approaches further enhance the resolving power and are nearly ready for experiments
  - Based on simulations, metrics will improve as array is filled (GRETA)
  - paper manuscript is in preparation

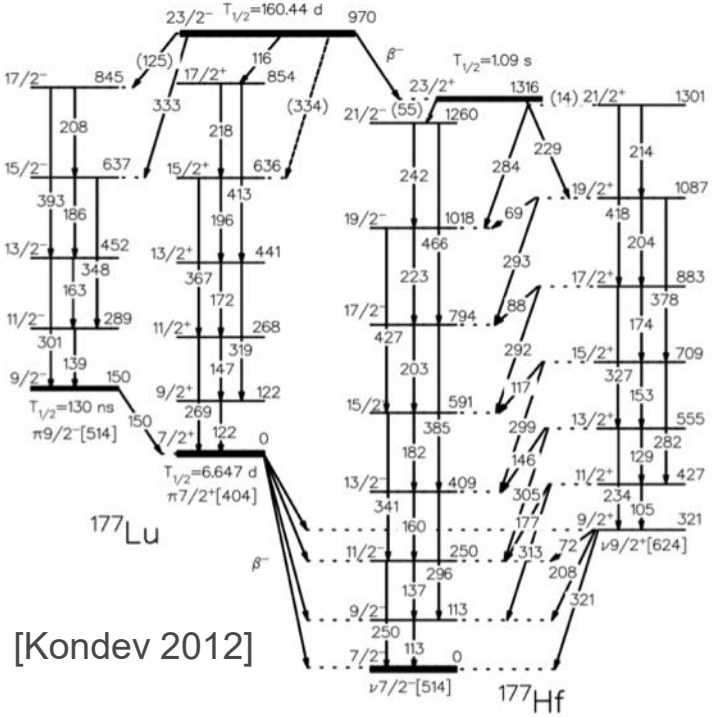
- **Things to do**
  - Incorporate pair production algorithm into model – early results look promising
  - SPEED up code - 10-20x slower than AFT tracking

ML TOOLS FOR LEVEL-SCHEME DESIGN

MAPPING OF EXCITED STATES IN NUCLEI

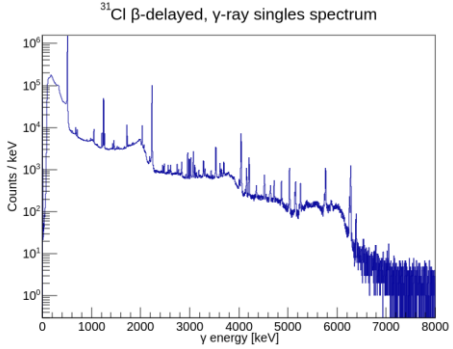
Building level schemes from data collected from the large gamma-ray arrays

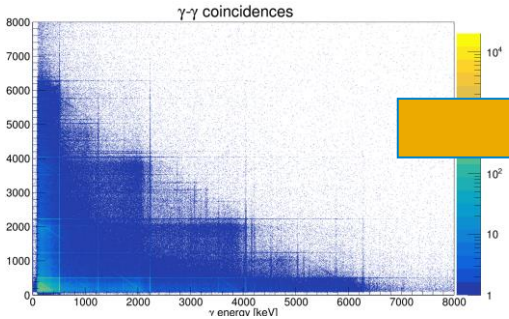
- A major deliverable from large γ-ray arrays is the mapping of excited nuclear states to energy levels which have spin, parity quantum numbers.
- Accomplished by analysis of γ-ray coincidence data *e.g.* 2-fold, 3-fold, … and performing angular distribution/correlation analysis.
- Level schemes can be complicated, and analysis times can take many months.
- Goal: Develop automated mathematical & ML tools for faster, reproducible decay-scheme extraction.



LEVEL-SCHEME BUILDER WORKFLOW

Extract Singles (S) and Coincidence Matrix (C)

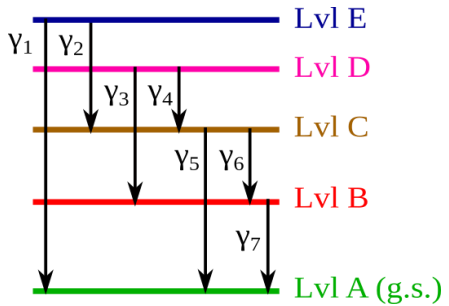




$$S = \begin{pmatrix} S_1 \\ S_2 \\ \cdots \\ S_n \end{pmatrix} \quad C = \begin{pmatrix} 0 & C_{i,j} & \cdots & C_{i,j} \\ C_{i,j} & 0 & & \vdots \\ & & 0 & \\ \vdots & & & \ddots \\ C_{i,j} & \cdots & & 0 \end{pmatrix}$$

Need to take the measured information and map to a level scheme.

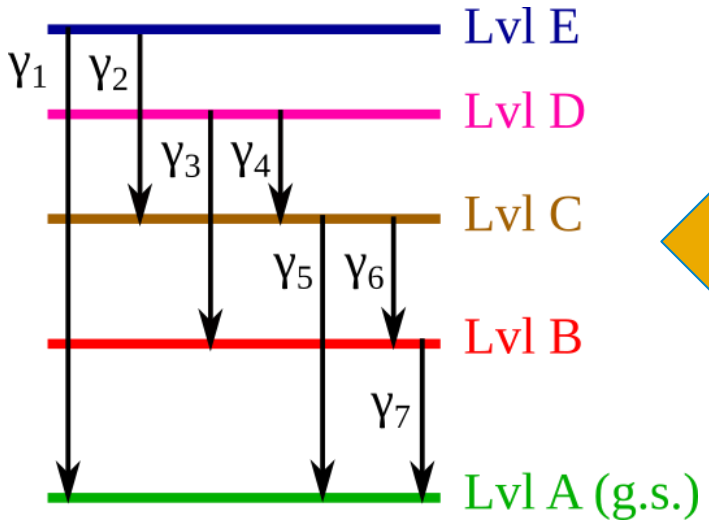
$$S = \begin{pmatrix} S_1 \\ S_2 \\ \cdots \\ S_n \end{pmatrix} \quad C = \begin{pmatrix} 0 & C_{i,j} & \cdots & C_{i,j} \\ C_{i,j} & 0 & & \vdots \\ & & 0 & \\ \vdots & & & \ddots \\ C_{i,j} & \cdots & & 0 \end{pmatrix}$$

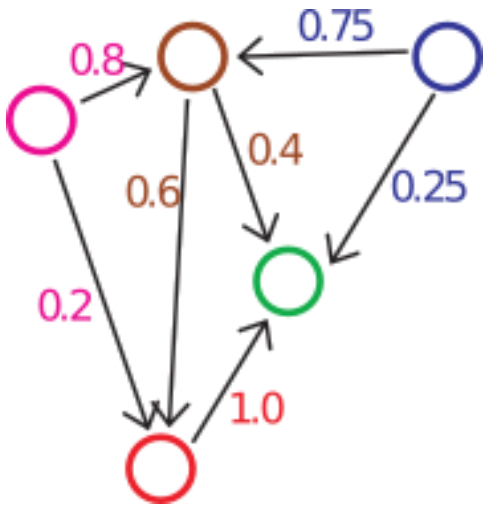


“Level-Centric” Decay Scheme

Decay schemes can be represented as graphs.

- Each level within the decay scheme corresponds to a vertex (or node), and the edges connecting these vertices correspond to γ-ray transitions between levels.
- Gamma-ray branching ratios correspond to edge weights.



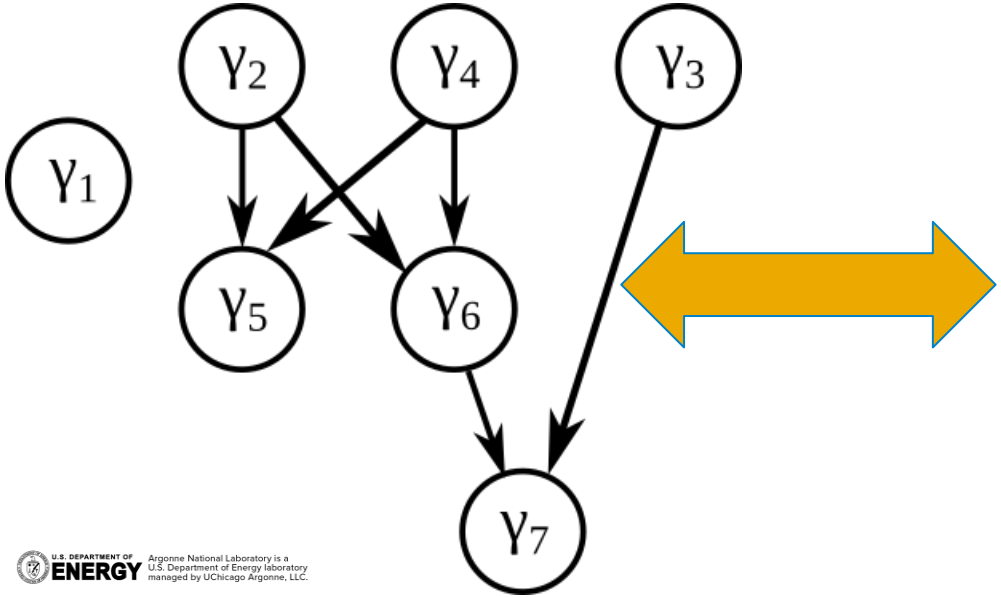


G. Demand, *Development of a Novel Algorithm for Nuclear Level Scheme Determination*, Master’s thesis, University of Guelph, 2009.

Adjacency Matrix

Every weighted, directed graph has a unique adjacency matrix *A*.

- Given a start position of vertex *i*, element $A_{i,j}$ is the probability of transitioning directly to vertex *j* (non-zero numbers=branching ratios)
- Transition energy information not needed for network connectivity but is useful for level-centric scheme construction.



	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7
γ_1	0	0	0	0	0	0	0
γ_2	0	0	0	0	0.4	0.6	0
γ_3	0	0	0	0	0	0	1.0
γ_4	0	0	0	0	0.4	0.6	0
γ_5	0	0	0	0	0	0	0
γ_6	0	0	0	0	0	0	1.0
γ_7	0	0	0	0	0	0	0

MATHEMATICAL FORMULATION

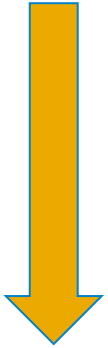
Writing Level Scheme Construction as Matrix Equations

- Start with data from Gamma-Sphere experiment:
  - *S*: γ-ray transitions & intensities (as diagonal matrix)
  - *C*: γ-γ coincidence data
- Determine the outputs:
  - *A:* the matrix of branching ratios (Adjaceny Matrix)
  - *D*: the directed coincidence data (Direction of Gamma-ray flow)
- Following Demand (2013) using graph theory, we try to satisfy two equations simultaneously:

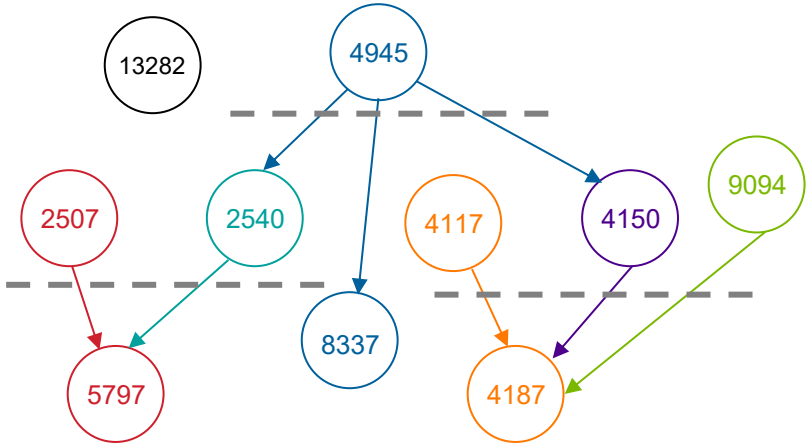
$$D = S((I - A)^{-1} - I) \quad \text{and} \quad C = D + D^T$$

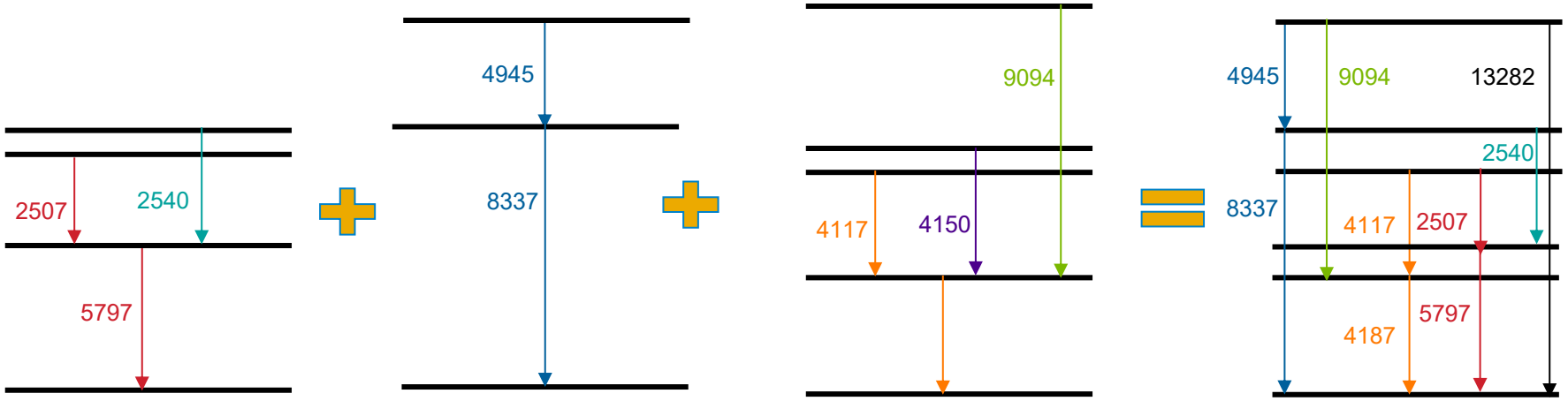
G. Demand, *Development of a Novel Algorithm for Nuclear Level Scheme Determination*, Master's thesis, University of Guelph, 2009.

Transition-centric graph



Level-centric decay scheme

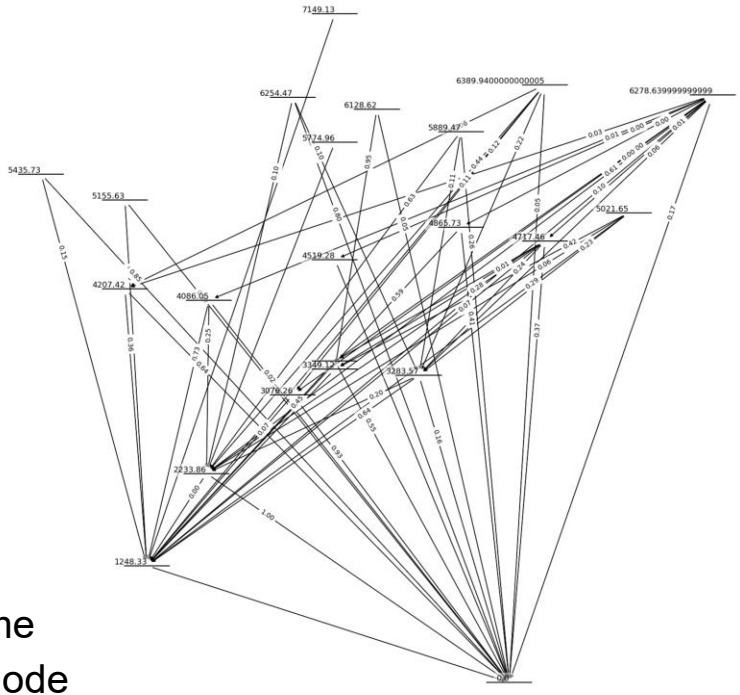




Benchmarking our Work

Successful Outcomes

- Ipopt used to solve large-scale, nonlinear optimization problem
- Successful cases:
  - ^{20}O
  - ^{43}K
  - ^{182}Ta
  - ^{200}Pb
- Maxing out at about 30 - 40 transitions per decay scheme
- Time to converge <1 minute on a serial CPU compute node
- Example case: ^{200}Pb (20 transitions)
  - Original problem size: 800 variables, 390 constraints
  - Reduced problem size: 627 variables, 216 constraints



GRayMAN – Gamma-Ray Multipeak Analysis Network

GRayMAN – Gamma-Ray Multipeak Analysis Network

A unified platform for automated γ-ray calibration and multipeak analysis, designed to improve accuracy, efficiency, and reproducibility in nuclear-structure research.

GrayMAN Viewer • GrayCAL • Autofitter • Arrow/Parquet Data Layer • Analytics Layer (Level Scheme Builder, ENSDF Reader, Other User Defined Tools)

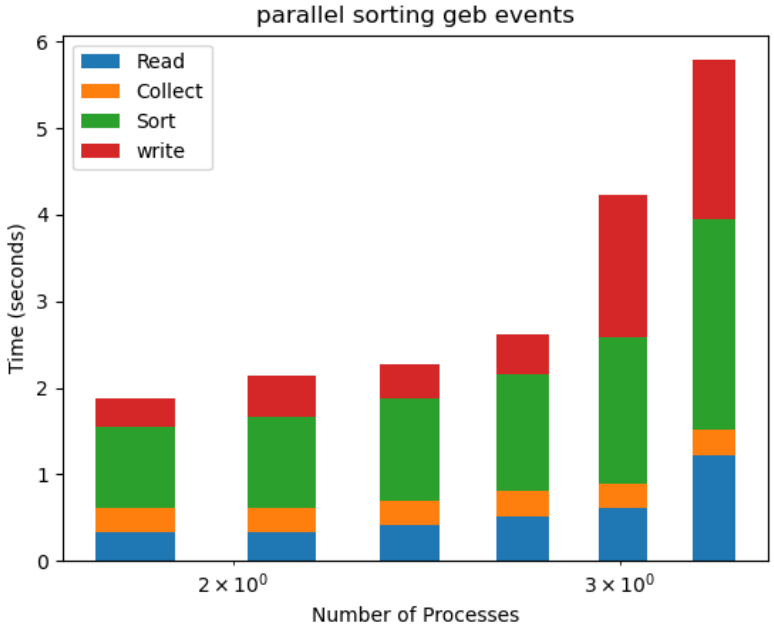
Provides End-to-end workflow: raw events → nuclear structure

Develop a modular, extensible Python suite that:

1. Loads and organizes large sets of γ-ray data
2. Provides interactive visualization tools
3. Performs calibrations and corrections
4. Fits spectra automatically using statistical guardrails
5. Integrates with advanced analysis techniques to extract level-schemes, band identification, and data labeling (the other FOA).

IMPROVING SCALABILITY OF GEBSORT/GEMERGE

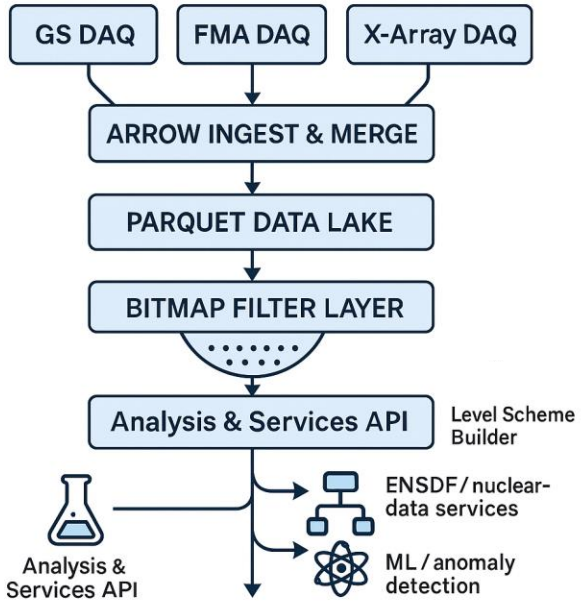
- Current workflow for managing detector studies of gamma-gamma interactions requires processing multi-gigabyte files on workstations overnight.
- We implemented a **parallel sample sort** able to process a 12 GiB dataset in under 7 seconds.
- This graph shows our scalability and performance on the ALCF Polaris machine, broken out by component of the new sorting program: On 32 nodes (2048 total processes) we have improved sorting time enough that file I/O begins to be more of a contributing factor, but 7 seconds represents a very small portion of overall workflow runtime.



MULTI-DAQ DATA PROCESSING ARCHITECTURE

Problem: How to store event data in compressed format and for real-time data analysis?

- Adobe Arrow for in-memory processing, time and energy calibration, event time sorting
- Columnar storage (Adobe Parquet) for fast scanning, ML, analytics
- Implementation of roaring bitmaps for ultra-fast coincidence queries
- Results of query passed to analysis framework *e.g.* GrayMAN spectrum analysis for coincidence intensities used in Level Scheme building.



Cromaz, M., Symons, T. J. M., Lane, G. J., Lee, I. Y., & MacLeod, R. W. (2001). *Blue: A database for high-fold γ-ray coincidence data.* **Nuclear Instruments and Methods in Physics Research Section A**, **462**(3), 519-529.

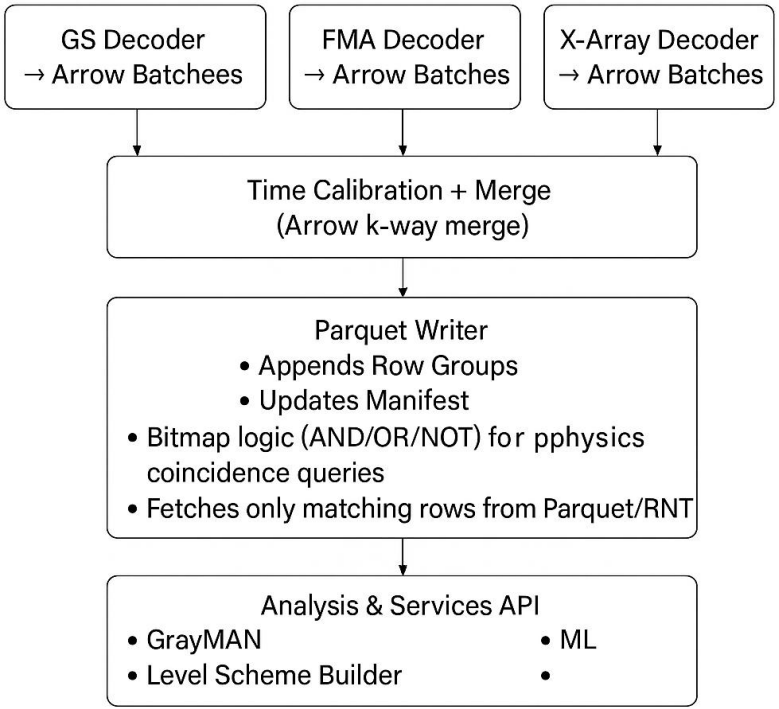
BLUE → ARROW/PARQUET MAPPING HIGHLIGHTS:

BLUE Component	Arrow/Parquet Equivalent	Improvement
Bitsets for detector coincidences	Roaring Bitmaps	Faster, compressed, standard format
Fixed binary layout	Parquet schema	Evolves, self-describing
C++ pointer-walking	Arrow compute kernels	SIMD + GPU acceleration
Fold patterns	Fold bitmaps + column filters	Multi-attribute queries
Gating tables	Combined bitmap + predicate pushdown	More flexible and faster
No ML support	Arrow → NumPy → PyTorch/TensorFlow	AI-ready

Arrow/Parquet gives us a **modern, ML-ready framework** that BLUE cannot provide.

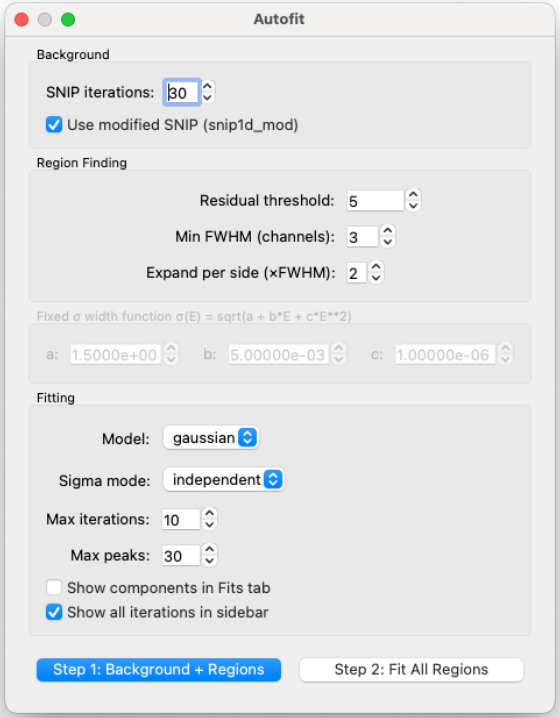
WHAT THIS FRAMWOK ENABLES

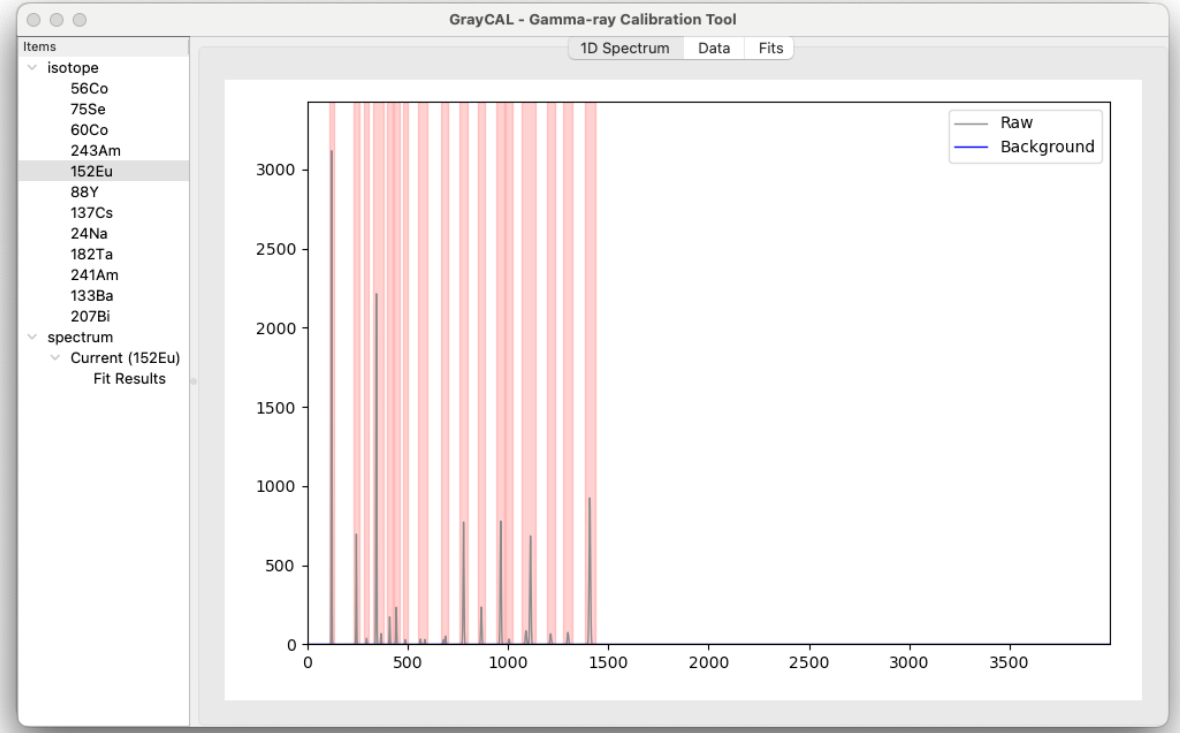
- Handles independent DAQ streams (GS, FMA, X-Array) decoded to Arrow Record Batches
- Parallel Arrow-based decoding enables zero-copy (pointers), high-throughput processing
- Arrow-based k-way merge creates unified, time-ordered event batches
- Merged data written as Parquet row groups (compressed, columnar, scalable)
- Bitmap Builder generates Roaring Bitmaps for fast coincidence selection
- Indexing and querying run concurrently with ingestion and writing
- Unified Analysis & Services API supports allows integration with analysis packages (ROOT, GRayMan, etc)
- End-to-end parallelism enables near-real-time access to multi-DAQ experimental data
- Fast access – queries (gates) ~ 1sec, full scan 1-2 minutes for billion events..



The Analysis Front End (GrayMAN/GRayCAL)

Graphical User Interface to Connect with Experiment Database (PyQT, Mathlab)





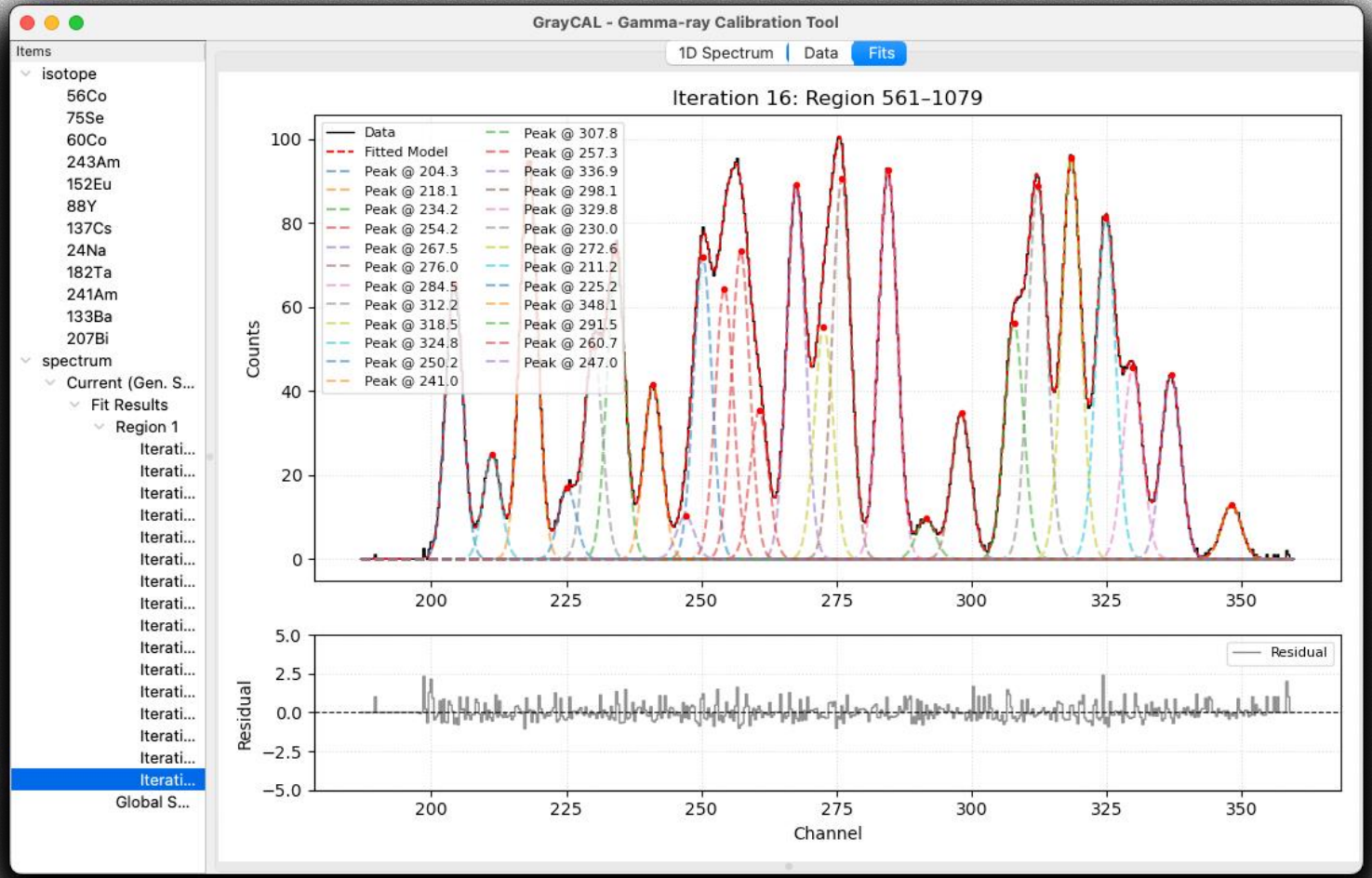

GrayCAL - Gamma-ray Calibration Tool
Items
isotope
56Co
75Se
60Co
243Am
152Eu
88Y
137Cs
24Na
182Ta
241Am
133Ba
207Bi
spectrum
Current (Gen. S...
Fit Results
Region 1
Iterati...
Global S...
1D Spectrum
Data
Fits
Iteration 16: Region 561–1079
Data
Fitted Model
Peak @ 204.3
Peak @ 218.1
Peak @ 234.2
Peak @ 254.2
Peak @ 267.5
Peak @ 276.0
Peak @ 284.5
Peak @ 312.2
Peak @ 318.5
Peak @ 324.8
Peak @ 250.2
Peak @ 241.0
Peak @ 307.8
Peak @ 257.3
Peak @ 336.9
Peak @ 298.1
Peak @ 329.8
Peak @ 230.0
Peak @ 272.6
Peak @ 211.2
Peak @ 225.2
Peak @ 348.1
Peak @ 291.5
Peak @ 260.7
Peak @ 247.0
Counts
Residual
Channel

TABLE OF DELIVERABLES AND SCHEDULE

REMAINING MILESTONE SCHEDULE

Year	Milestone	Personal
FY25	Improve peak-to-total of γ -ray spectra	AK, TL
FY25	Accel. merging of DAQ data	RL, RR, TL
FY25	Algorithm to extract intensities	MC, SC
FY25	~~Optimization and ML tools for Coulex~~	MS, SL, DL
FY26	Improve tracking eff. at high energy	AK, TL
FY26	Storage of event in indexed form	RL, RR, TL
FY26	Level-scheme design from N-fold data	SL, MC, TL
FY25	~~Reinforced learning of Coulom excit.~~	MS, SL, DL

SUMMARY

- Advances in ML γ-ray tracking, level-scheme construction, and the GrayMAN analysis system.
- Designed a high-performance Arrow/Parquet + Bitmap backend for scalable coincidence analysis – could be implemented for all ATLAS data.
- Established an integrated workflow from raw data → event reconstruction → decay scheme.
- Positioned to support GRETA-era experiments and ML-driven discovery.
- Next year: expand GrayMAN/GrayCAL, apply to real data, add 3D coincidences to Level Builder, refine ML models, implement Arrow/Parquet roadmap.

ACKNOWLEDGEMENTS

Collaborators on Project

T. Budner,[1] S. Carmichael,[1] M. Carpenter,[1] F. Kondev,[1] A. Korichi,[3] R. Latham,[2] T. Lauritsen,[1]
D. Lenz,[2] T. Lynn,[2] R. Ross[2], M. Siciliano[1] and E. Weiller[3]

[1] *Physics Division, Argonne National Laboratory, Lemont, IL 60439, USA*
[2] *Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA*
[3] *IJCLab Orsay, IN2P3-CNRS, Université Paris-Saclay and Université Paris-Sud, 91405 Orsay, France*

And thank you for your attention!

PULL PEN SLIDES

Example of parameters, FOMs and models that have been used in this work

		all_accuracy_correlation	all_accuracy_R	complete_accuracy_correlation	complete_accuracy_R	incomplete_accuracy_correlation	incomplete_accuracy_R	validation_accuracy_	validation_accuracy_R
C	C_1000	-0.058193674	0.058193674	-0.053454752	0.053454752	-0.052224147	0.052224147	0.20516106	0.00045849
	C_10000	0.058193674	0.058193674	0.053454752	0.053454752	0.052224147	0.052224147	-0.20516106	0.00045849
Columns	**cols_aft**	-0.076325647	0.076325647	-0.005300437	0.005300437	-0.204519661	0.204519661	-0.01204583	0.8387107
	cols_aft-fast	0.0888634	0.0888634	0.107414741	0.107414741	0.025623966	0.025623966	0.0385706	0.5144265
	cols_aft-fast-tango	0.128330901	0.128330901	0.109293607	0.109293607	0.133188852	0.133188852	0.07734063	0.19061326
	cols_aft-fastest	0.021426865	0.021426865	0.052850234	0.052850234	-0.050385041	0.050385041	-0.14379215	0.01459295
	cols_aft-fastest-tango	0.069065148	0.069065148	0.063813769	0.063813769	0.061197885	0.061197885	-0.07738052	0.19038397
	cols_aft-tango	-0.006229761	0.006229761	-0.003607784	0.003607784	-0.010028997	0.010028997	-0.07953441	0.1783003
	cols_aft-true	-0.432470377	0.432470377	-0.203709027	0.203709027	-0.794319516	0.794319516	0.08811009	0.13578374
	cols_all	0.157322643	0.157322643	0.126755755	0.126755755	0.178449978	0.178449978	0.36398759	0
	cols_fast	0.089000176	0.089000176	0.107563293	0.107563293	0.025698618	0.025698618	-0.06284176	0.28783962
	cols_fast-tango	0.128222102	0.128222102	0.109299883	0.109299883	0.132868287	0.132868287	0.05580173	0.3453722
	cols_fastest	-0.520524263	0.520524263	-0.620266848	0.620266848	-0.168822785	0.168822785	-0.13266372	0.02435088
	cols_oft	0.113075771	0.113075771	0.104667581	0.104667581	0.099797409	0.099797409	0.09618718	0.10330652
	cols_oft-fast	0.153309525	0.153309525	0.137876211	0.137876211	0.143771884	0.143771884	0.00680071	0.90851543
	cols_oft-fast-tango	0.16630786	0.16630786	0.13366864	0.13366864	0.189327228	0.189327228	-0.038431	0.51595111
	cols_oft-fastest	0.130196021	0.130196021	0.11340284	0.11340284	0.129833899	0.129833899	-0.06679056	0.25855802
	cols_oft-fastest-tango	0.140003446	0.140003446	0.10277407	0.10277407	0.179850803	0.179850803	-0.05422619	0.35918367
	cols_oft-tango	0.129167097	0.129167097	0.093509465	0.093509465	0.168679336	0.168679336	-0.05350823	0.36559044
	cols_oft-true	-0.478740905	0.478740905	-0.530005993	0.530005993	-0.240212145	0.240212145	-0.00558416	0.92482719
Model type	model_type_lp	0.043636361	0.043636361	0.038356837	0.038356837	0.042782798	0.042782798	0.06523047	0.26987133
	model_type_lr	-0.077810651	0.077810651	-0.06480834	0.06480834	-0.0838193	0.0838193	0.12590334	0.03269045
	model_type_milp	0.099322254	0.099322254	0.08493778	0.08493778	0.102348504	0.102348504	-0.23708948	0.00004821
	model_type_svm	-0.065147964	0.065147964	-0.058486277	0.058486277	-0.061312002	0.061312002	0.04595567	0.43721044
Non-negative	nonneg_False	0.00252598	0.00252598	-0.032721003	0.032721003	0.075811971	0.075811971	-0.30128662	0.00000019
	nonneg_True	-0.00252598	0.00252598	0.032721003	0.032721003	-0.075811971	0.075811971	0.30128662	0.00000019

C: Controls the sparsity of the model; a smaller C means a simpler model

Columns: Groups of FOM features

Model type: The approach for training the ML model

LP: Linear program (more precise than SVM), LR: Logistic regression (simplest, but least accurate)

MILP: Mixed integer linear program (most accurate), SVM: Support-vector machine (basic linear model)

Non-negative: If "noneg = True," all weights in the FOM are non-negative, focusing on minimizing values.

If "noneg = False," some weights can be negative, allowing for maximization.

USES OF ARROW/PARQUET AND WHY IT FITS

Arrow/Parquet widely adopted across science and industry:

- CERN HEP experiments (CMS, ATLAS, LHCb)
- Astronomy pipelines (LSST, Gaia, NASA time-domain surveys)
- Climate & geoscience: NOAA, Pangeo, ESA cloud-native data
- Genomics & bioinformatics: TileDB, Spark-based genomics
- Modern data engines: DuckDB, Polars, BigQuery, Snowflake

Why Arrow/Parquet suits this application:

- Columnar format → fast filtering on energy, detector, timestamp
- Zero-copy Arrow buffers → efficient merging & time alignment
- Vectorized compute → rapid calibration + event building
- Parquet row groups → scalable multi-DAQ data lake
- Interoperable with ML tools: PyTorch, scikit-learn, Polars
- Integrates naturally with Roaring Bitmaps for msec-level queries

ADOPTED METHODOLOGY

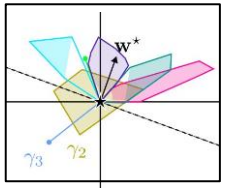
ML Approach for Learning-to-rank

- When ordering, we want

  FOM(best incorrect order) > FOM(true order)

- We don't care about the FOM value, only the difference between desired and undesired orders
- The **best incorrect order** requires ordering with the FOM
- Let FOM be weighted sum of physics derived objectives (e.g. existing FOMs), a simple, interpretable model, that prevents overfitting (*maximizes likelihood that the model can survive the translation from simulated to experimental data*

  FOM(order) = $\mathbf{w}^T f$(order)



- Allows simplification

  $\mathbf{w}^T$(f(incorrect) - f(true)) > 0

- If all features/FOMs are quantities that we want to minimize, constrain **w** positive, protect against overfitting
- Use linear classification (introduce mirrored data as second class → off the shelf solvers)

WHY ARROW, PARQUET, AND ROARING BITMAPS?

- Apache Arrow – High-performance in-memory processing
  - Zero-copy columnar arrays for fast filtering and merging
  - Ideal for multi-DAQ time alignment and calibration
  - Vectorized operations accelerate event building and gating
- Apache Parquet – Efficient on-disk columnar storage
  - Compression + predicate pushdown for fast event scans
  - Splittable datasets scale across multiple DAQs and runs
  - Cross-tool compatible with ML and analytics tools *e.g.* PyTorch
- Roaring Bitmaps – High-speed physics indexing layer
  - Encodes energy/detector/Δt predicates as compressed bitmaps
  - AND/OR/NOT operations enable instant coincidence resolution
  - Reduces billion-event scans to tiny targeted fetches (<1sec/query)