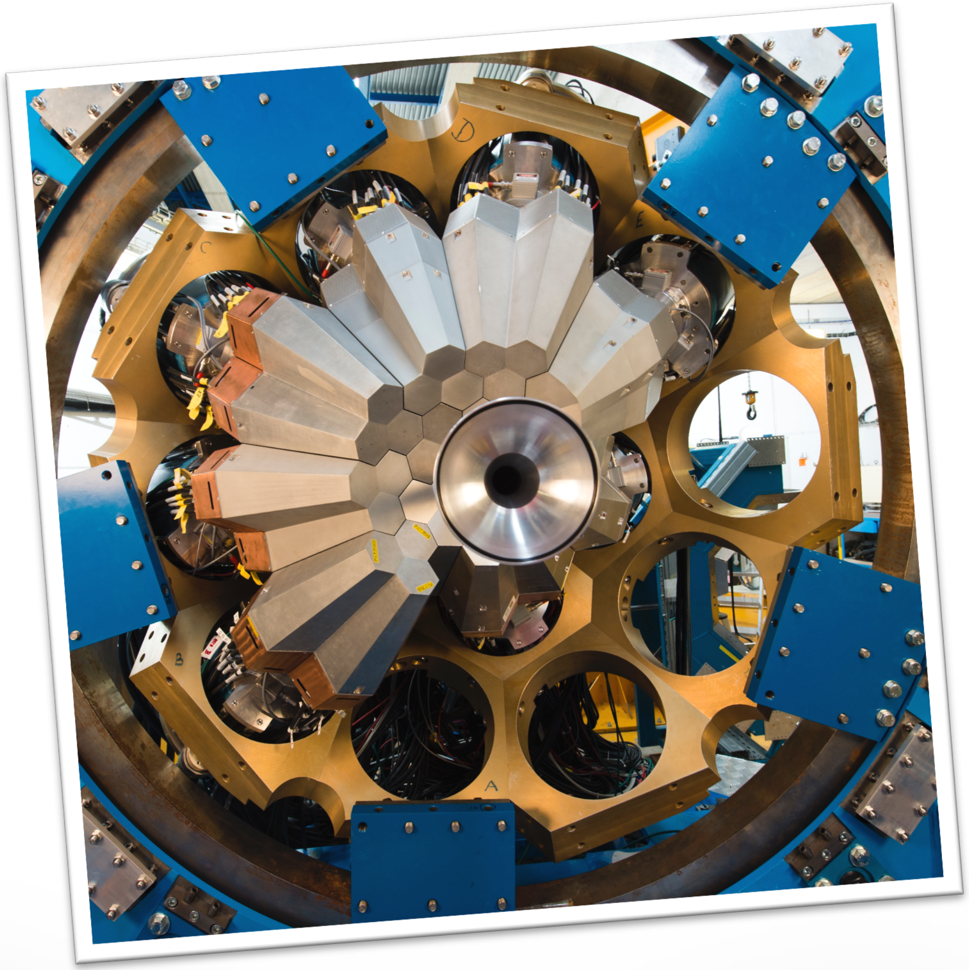
The Development of Novel Pulse Shape Analysis Techniques for AGATA

Fraser Holloway



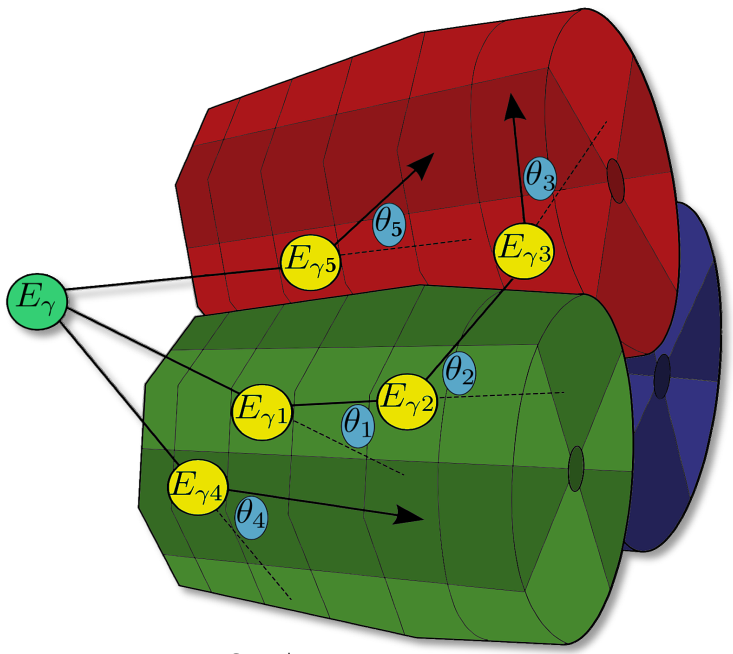
UNIVERSITY OF LIVERPOOL
LIV.DAT
Science & Technology Facilities Council
AGATA ADVANCED GAMMA TRACKING ARRAY

Pulse Shape Analysis (PSA)

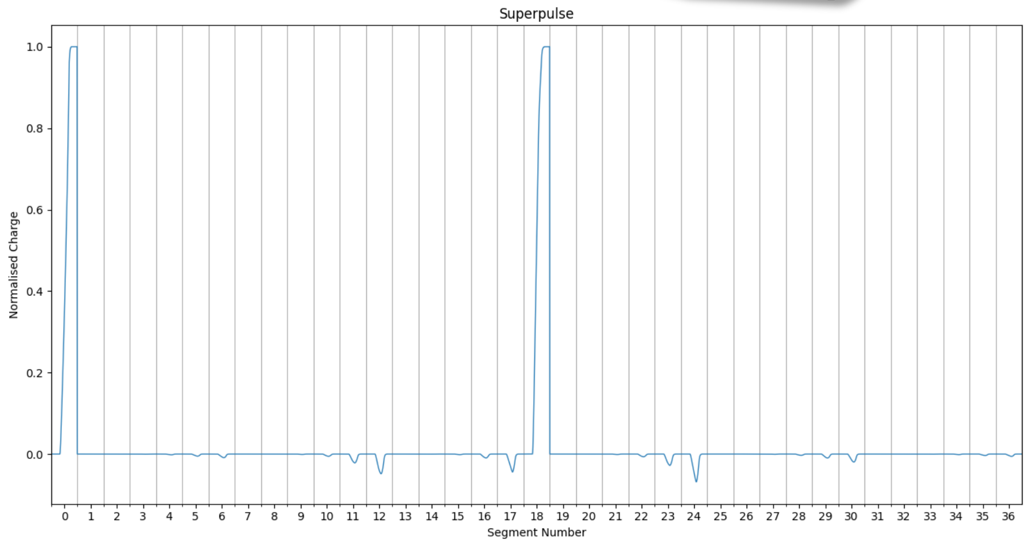
- γ-ray tracking requires positions at resolution ~5mm $M3D$ at ~5kHz/CPU.
- Positions must be inferred from electrical response (PSA).
- Complex detector response makes parametric methods insufficient.
- Instead we simulate the detector response in ADL 3.0
- Interaction locations are then determined by optimisation metrics:

$$Figure\ of\ Merit = \sum_{j}\sum_{t_i}\left|A_m^j\,[t_i] - A_s^j[t_i]\right|^p$$

For signals of segment j at time step t_i with p typically =2



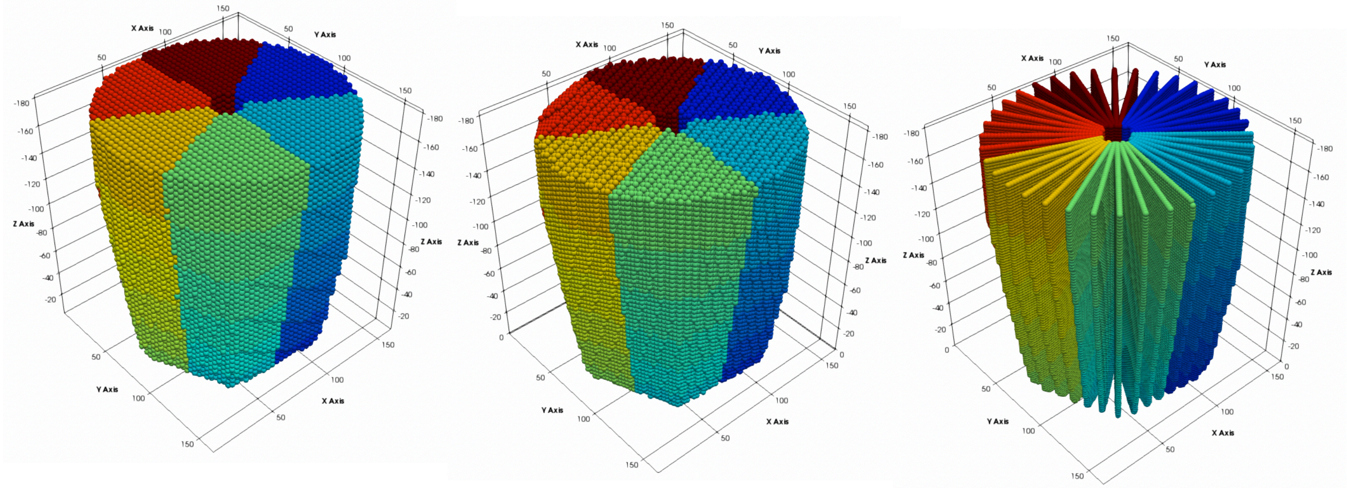
- Dataset is effectively 50,000 points, each with 37x121 values.
- My work is on developing Novel PSA techniques for AGATA.
  - Faster, more accurate.

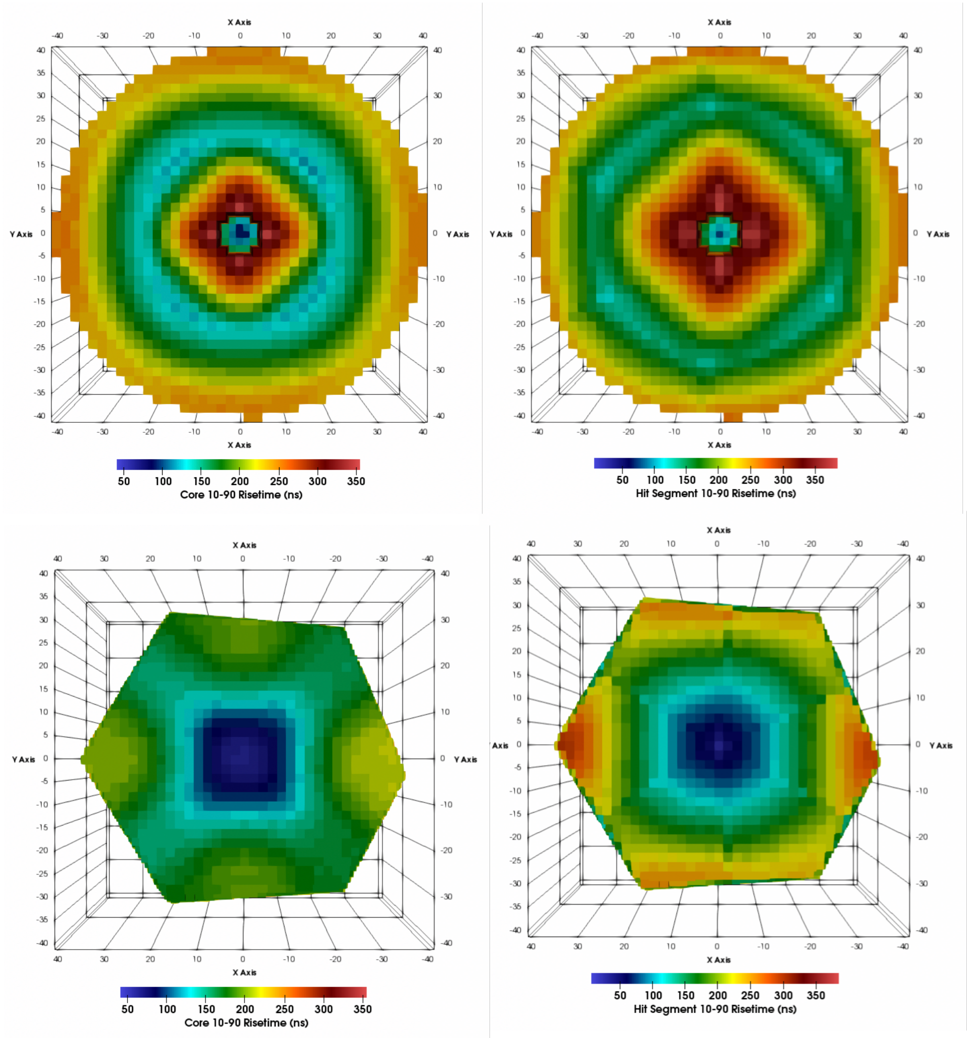


Fraser Holloway - F.Holloway@liverpool.ac.uk

Detector Simulation

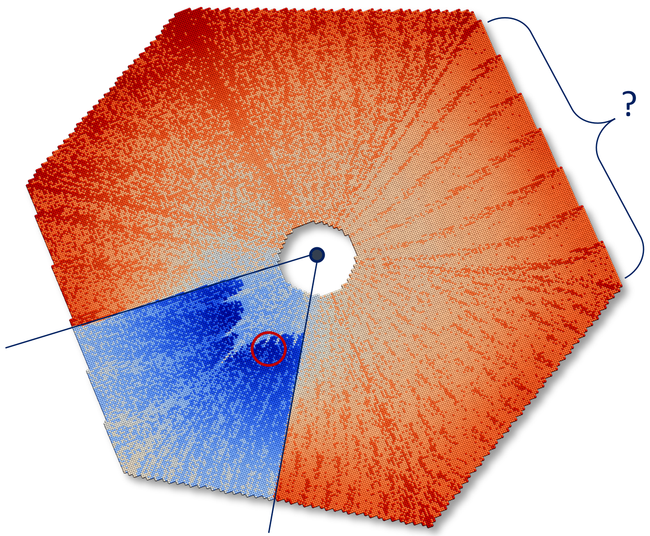
- Basis sets are typically precomputed at 2mm cubic grid spacing.
- Reliant on external SIMION potentials.
- 6-fold symmetric, polar and tetrahedral basis sets simulated.
- High resolution (0.5mm) basis set generated too.
- Option for dynamic resolution basis sets.



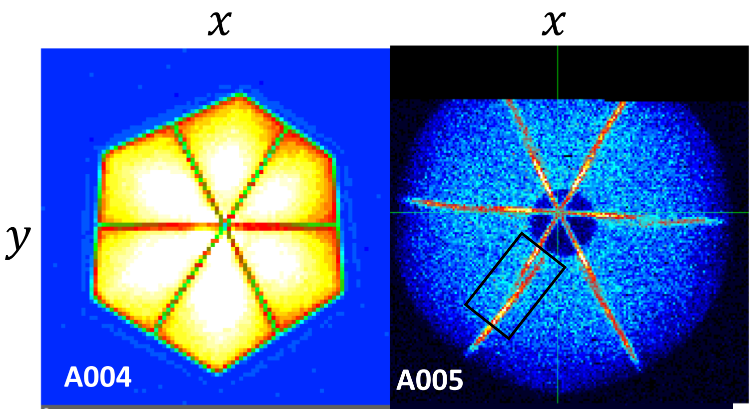


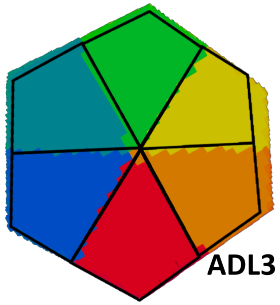
Simulation Limitations & Irregularities

- SIMION Field simulation limited to 0.5mm spacing.
- SIMION front segmentation is wrong.
- Unexplained ‘charge sharing’ between segments.
    - Overlap of SIMION definitions? (Confirmed by Marco AW:2019)
- Sharp discontinuities at edge changes (over-relaxation flaw?)



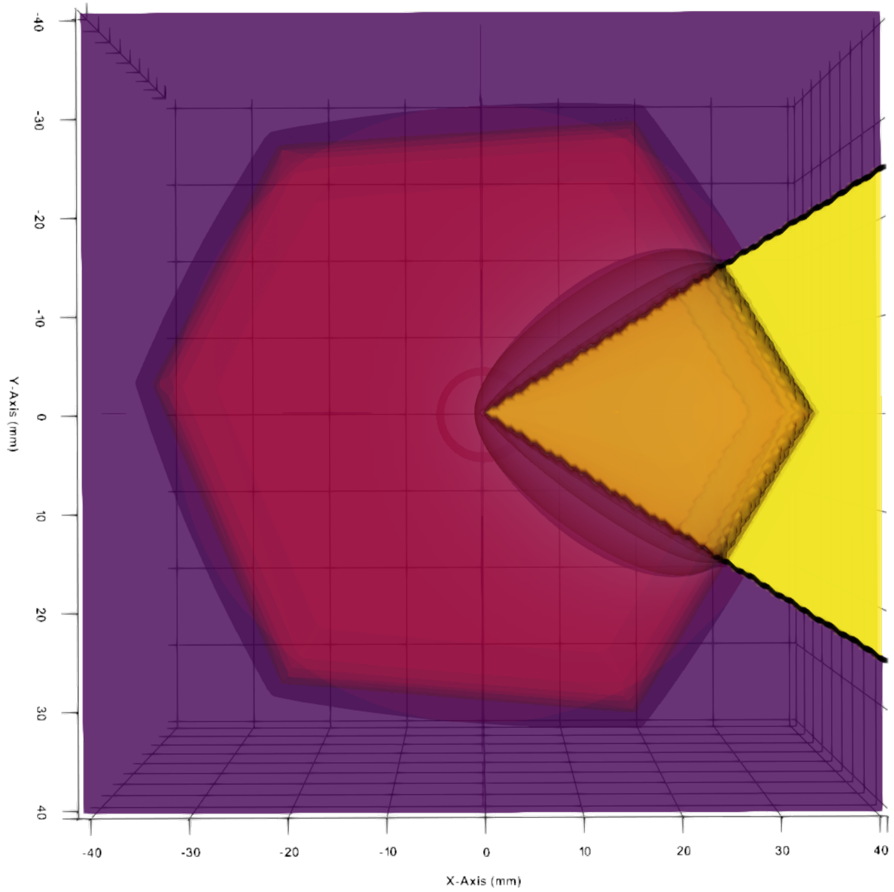
0.5mm FoM Plot showing odd effects
Optimum Circled





ADL3 segment labelling

SIMION Field Segment A1 Definition

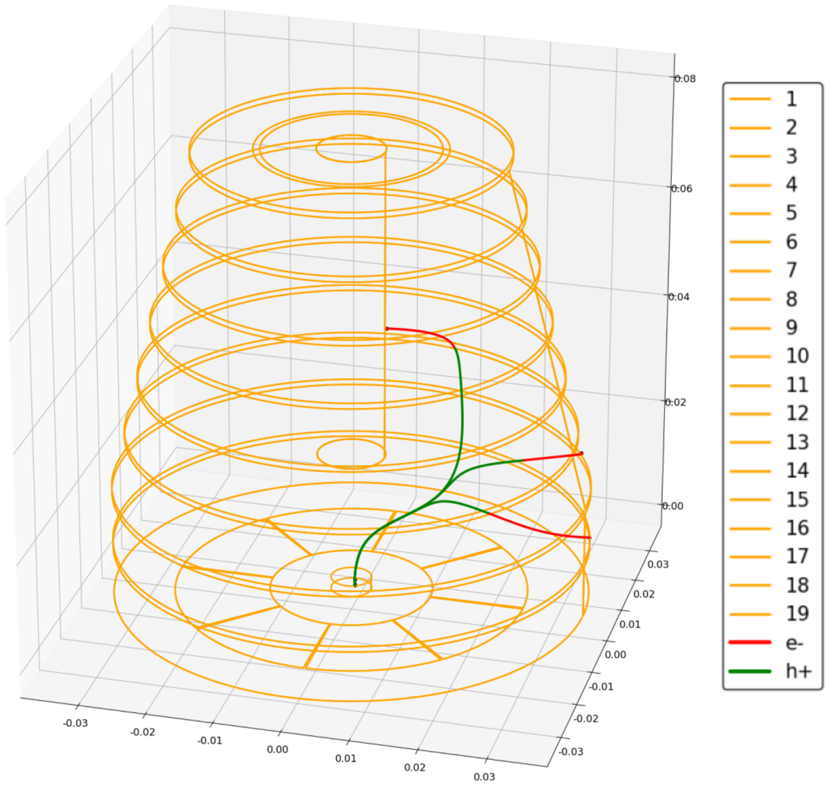


Fraser Holloway – F.Holloway@Liverpool.ac.uk

Simulations Moving Forward

- New detector simulation package has been developed by LEGEND: SolidStateDetectors.jl
- Written in Julia, multithreaded implementation with GPU (CUDA) support.
- Utilises ADL mobility models for simulation.
- Uses Cartesian & cylindrical geometry systems.
- Geometry defined off primitives, implicitly defined:
- Produces rectilinear grids of ρ, ϵ, weighting potentials, could be converted to .pa files.
- Produces charge trajectories, pulses, ∴ full simulation possible.
- Calculates depletion volumes, voltages.

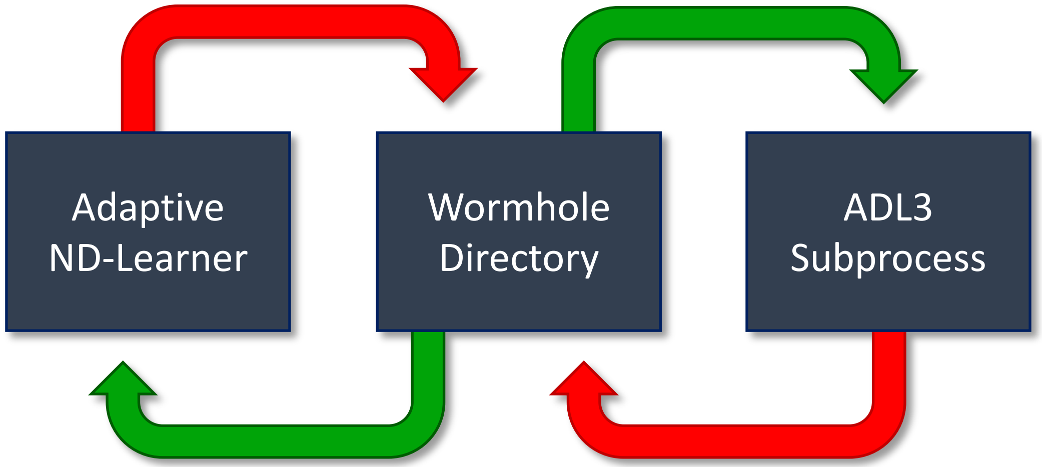
- AGATA crystals are far from simplistic:
    - Difficult to properly define using existing primitives.
    - Instead I added Tri-mesh support into the geometry constructor.
    - Computation is significantly more intensive ∴ I multithreaded it.
- All fields & potentials are generated, CAO needs to be checked before simulatior



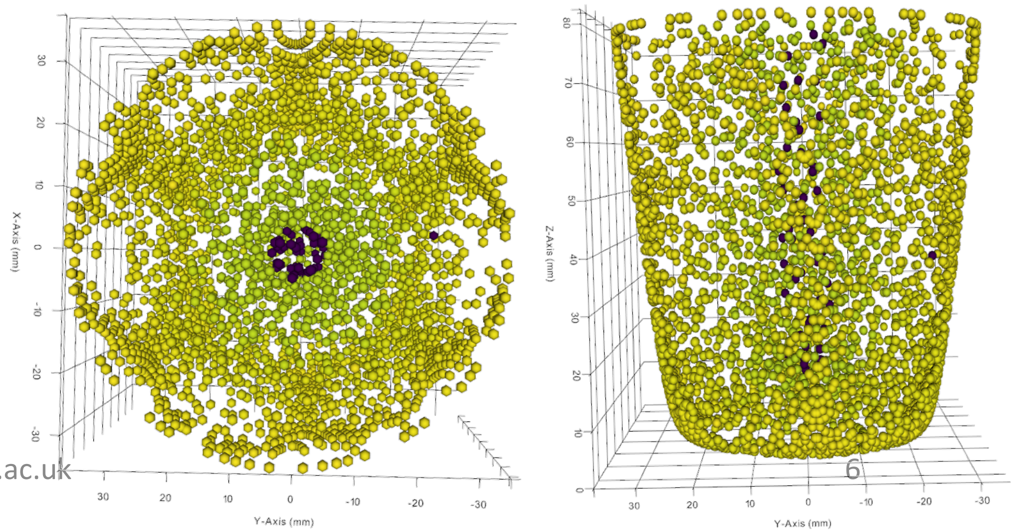
Fraser Holloway - F.Holloway@liverpool.ac.uk

Simulations Moving Forward

- My PSA methods don't rely on Euclidean information for navigation.
  - Maps & relationships are self-organized off pulse shape.
  - Most mapping occurs in non-Euclidean space anyways.
  - ∴ we don't need to keep a cubic or polar grid system.
- Other basis organisations could be used:
  - Adaptive Tetra-mesh.
  - Rectilinear grid.
- Work ongoing adding support for SolidStateDetectors control.



Generated Tetra-mesh points (5,000)



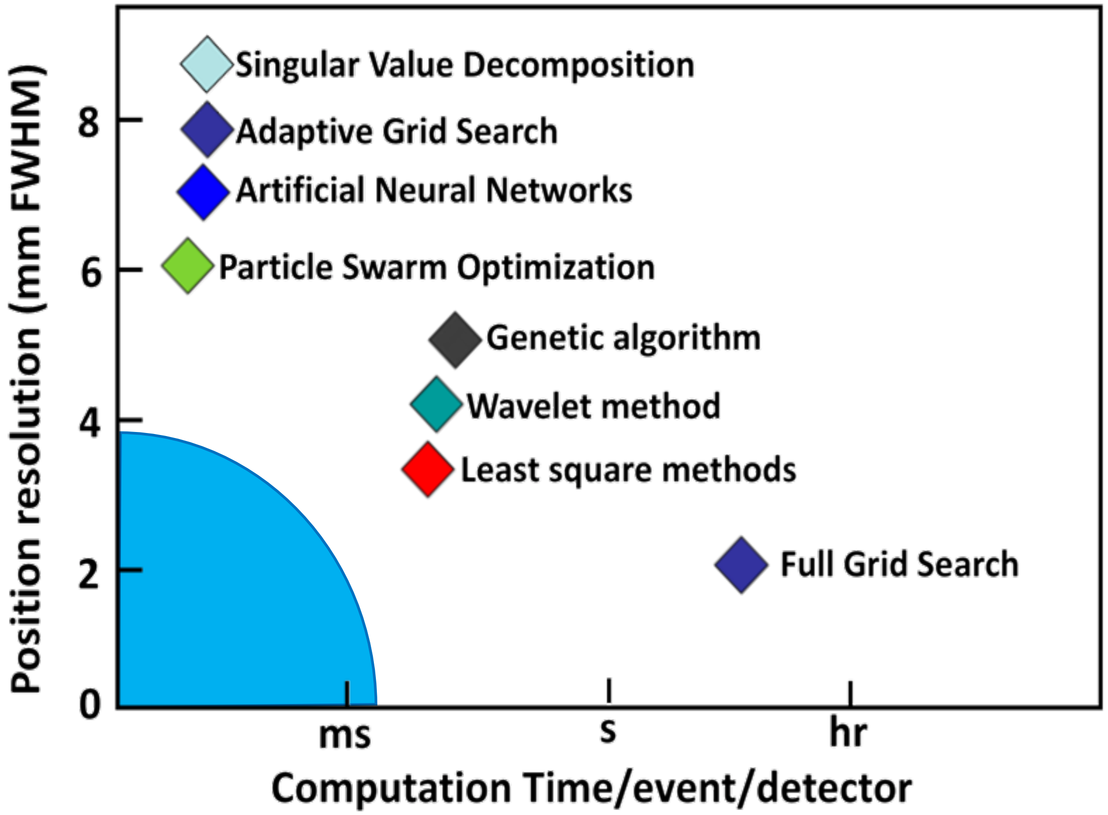
Novel Algorithm Development

Graph-accelerated techniques try to organize data and form efficient search spaces.

- Search spaces can be Non-Euclidean
- Searching n points can be $\mathcal{O}\log(n)$,
  - AGS is $\mathcal{O}\sqrt{n}$ at best

Machine Learning (ML) uses the simulated basis to learn trends via feature extraction.

- No searching is performed whatsoever.
- Simulated basis only needed for training.
- Needs an appropriate model & good data.
- No method for higher fold.



Novel Algorithm Development

Graph-accelerated approaches:

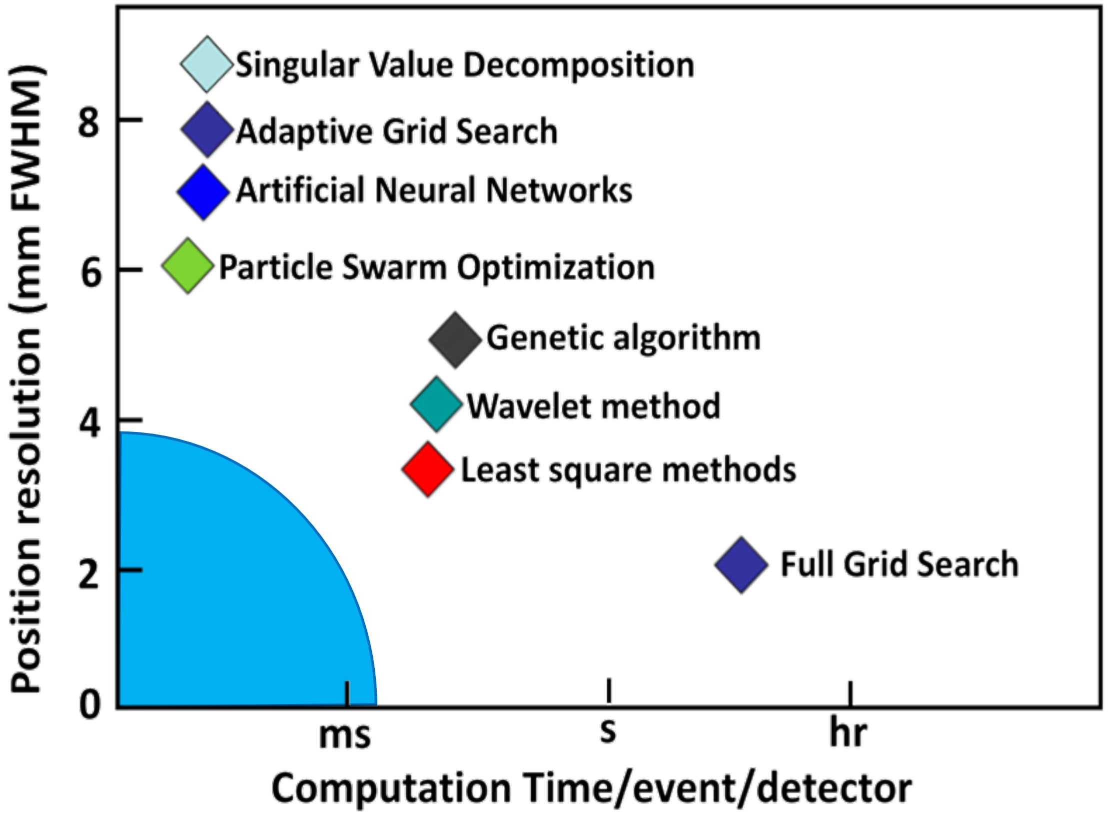
- kNN - k-dimensional Nearest Neighbors.
- LSH – Locality-based Sensitivity Hashing.
- ST/DT MKS – Maximum Kernel Search.
- ANN methods (FAISS, HNSW, NGT, NanoFLANN)

ML approaches:

- Fold Determination.
- Position Regression (CNN).
- Autoencoding/Fingerprinting (β-VAE).

Other options:

- GPU Accelerated parallel search.



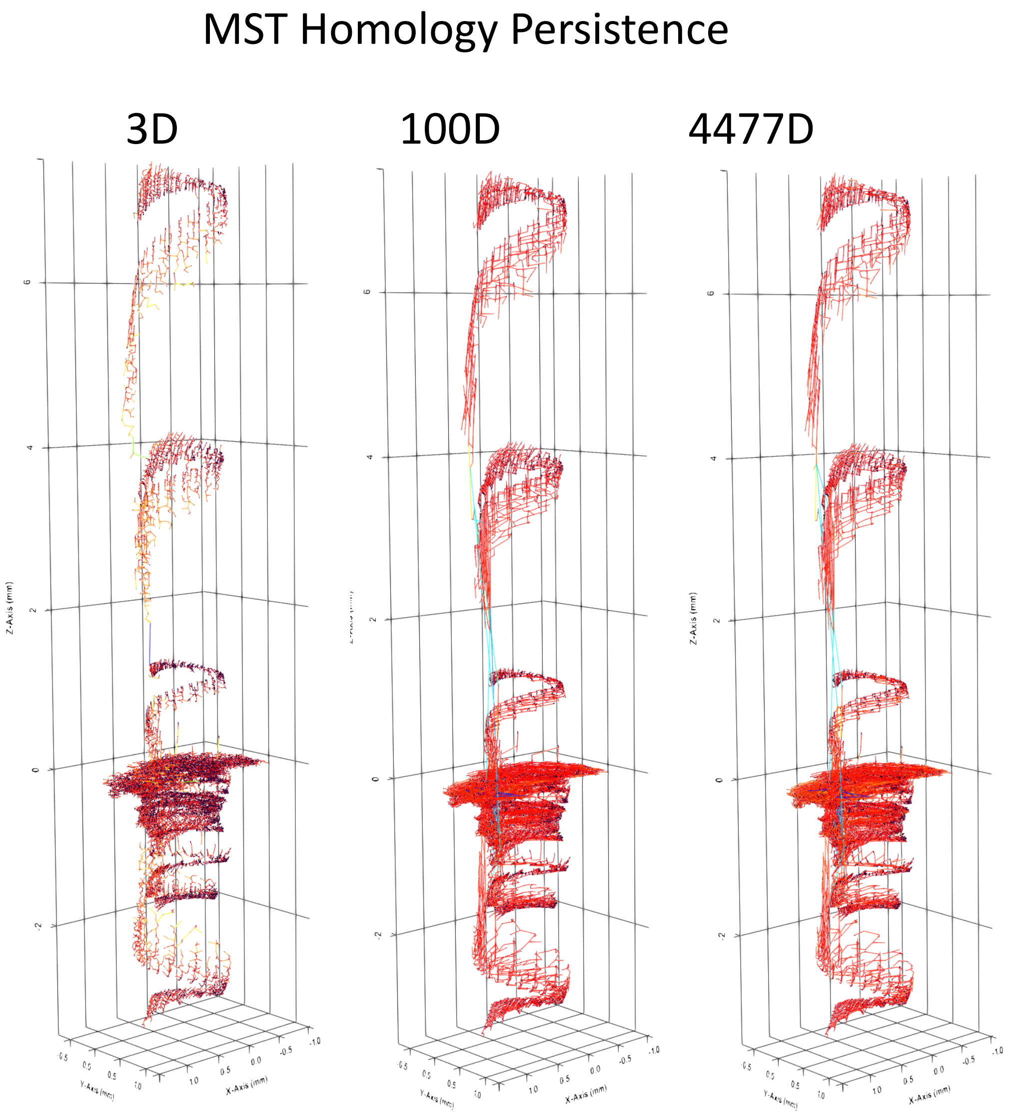
Dimensionality Reduction

- Current signal response is over-descriptive.
    - Massive covariance in signal response.
- Can we embed the response in fewer dimensions?
- Lower-dimensional space requires preserved homology:
    - Check MST, connectivity, point density, reconstruction loss.
- Linear transforms *should* still allow for graph algorithms to work:
    - Won’t work for all distance metrics.

- Nonlinear transforms form better reconstructions but embedded space is less useful.
    - Signal compression for data storage?

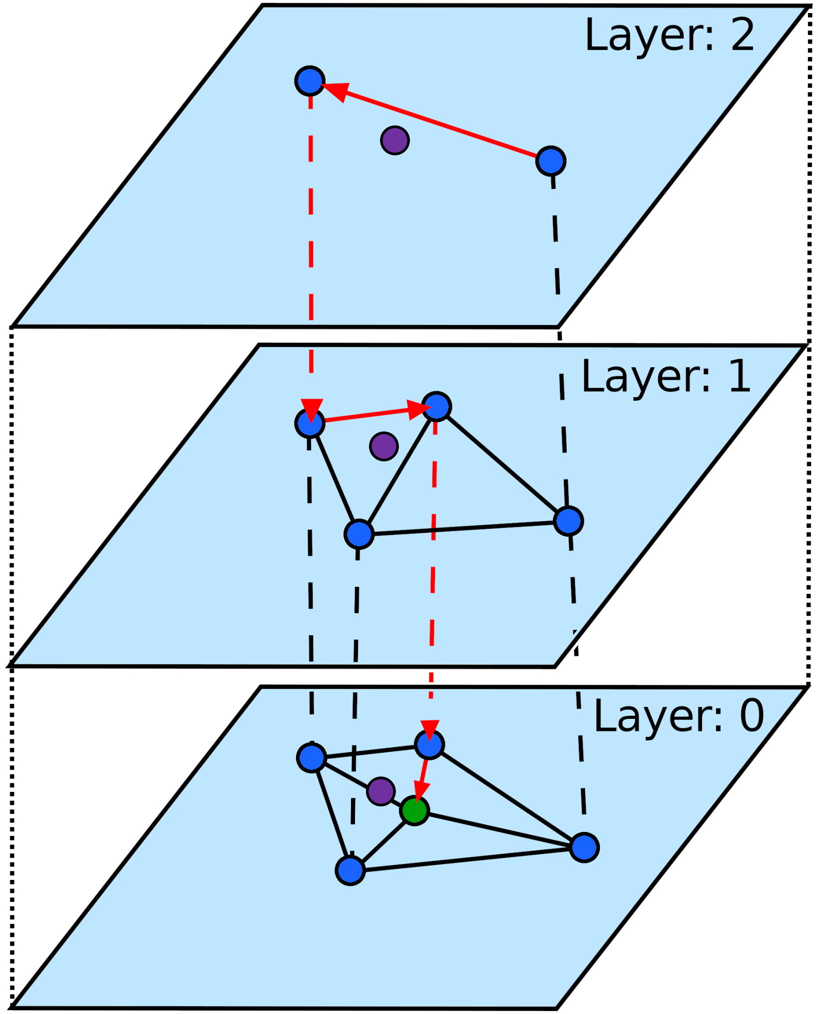
Alternate option is Autoencoders

- Can give a better reconstruction
- Generally slower
- Embedded space ranges in usefulness



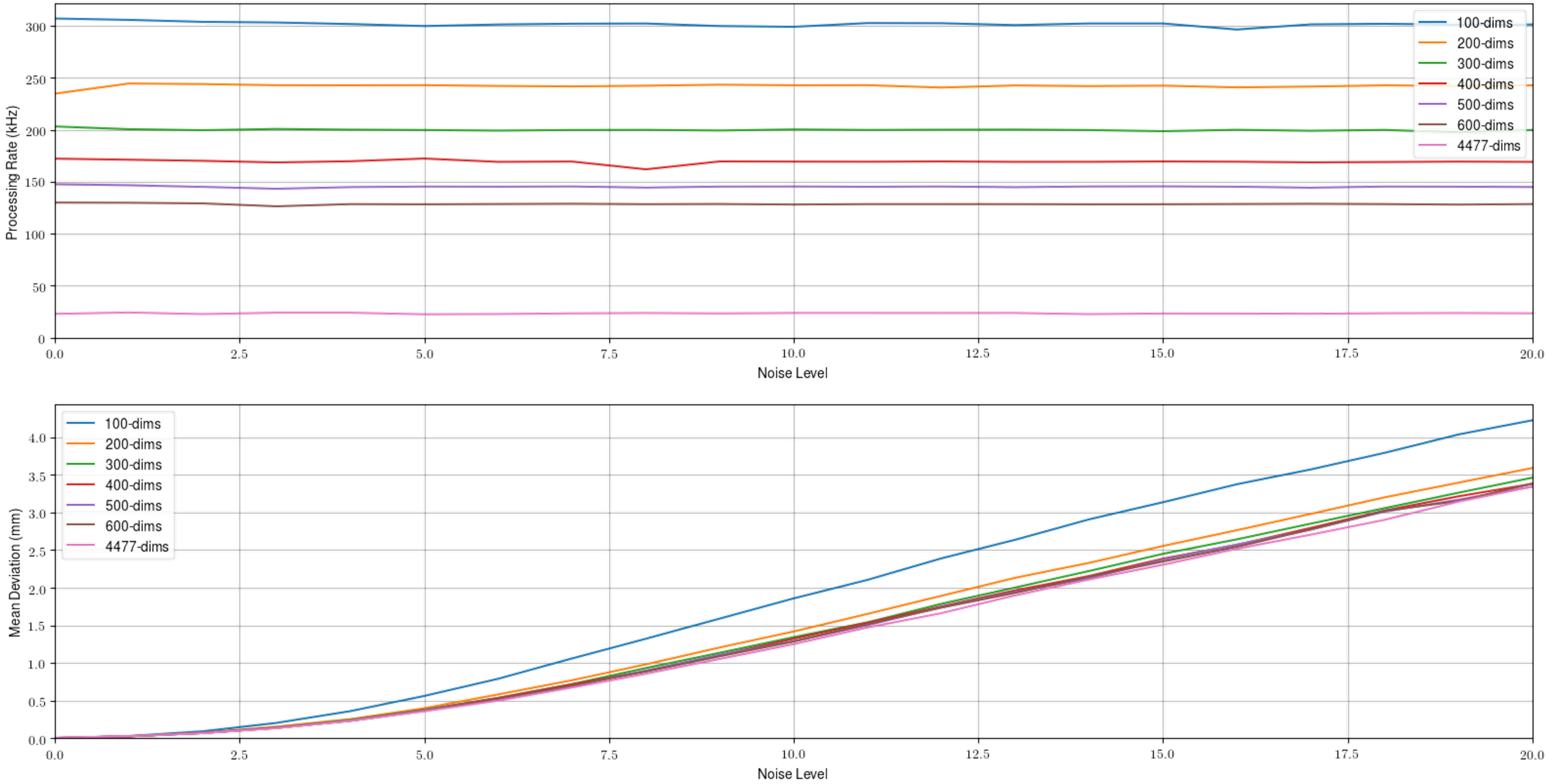
HNSW & Facebook AI Similarity Search (FAISS)

- FAISS is a k-NN search algorithm developed by Facebook AI labs (2018)
- Utilises A coarse HNSW Quantizer:
  - Hierarchical greedy graph traversal.
  - Allows for Billion-scale indexing.
- Optimised for mass execution:
  - Multi CPU computation using Heap Parallelisation.
  - Multi GPU computation using Sharding & Replication.
  - Perfect for multi-node PSA servers.
- Slightly faster performance than HNSWLIB, better prediction accuracy.
- GPU accelerated using Quadro RTX 5000



Simplified Search Structure

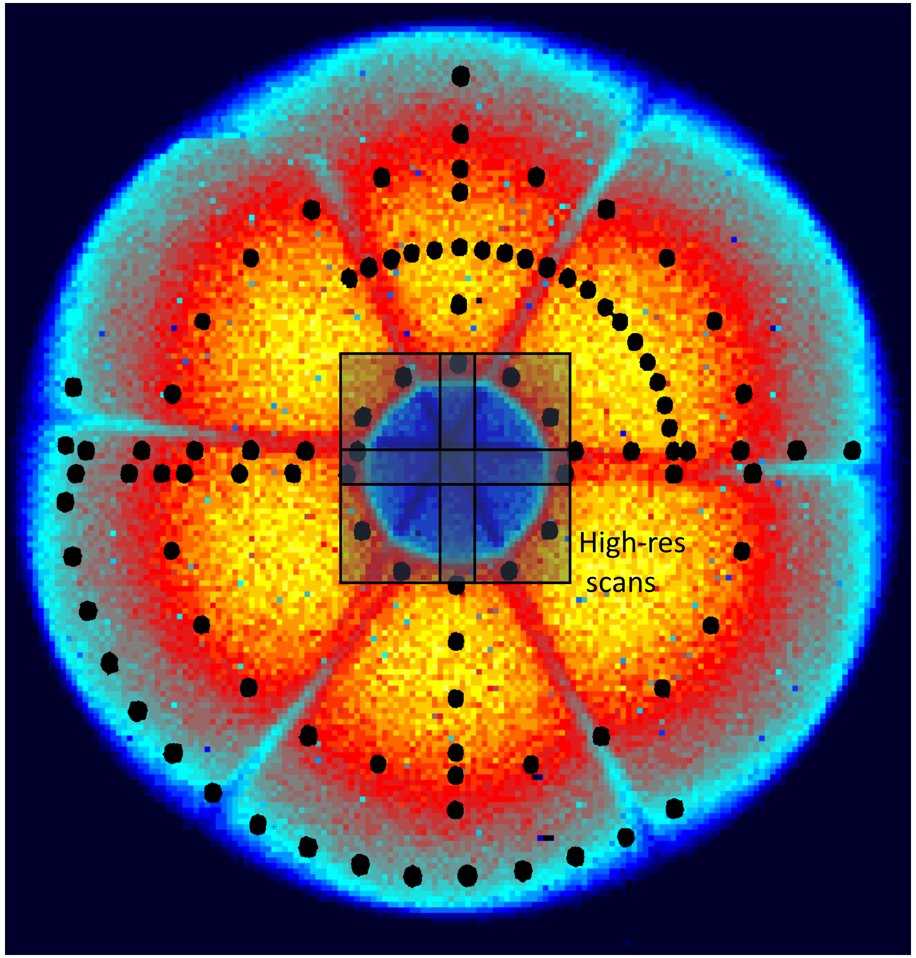
FAISS Performance Scaling



Experimental Validation

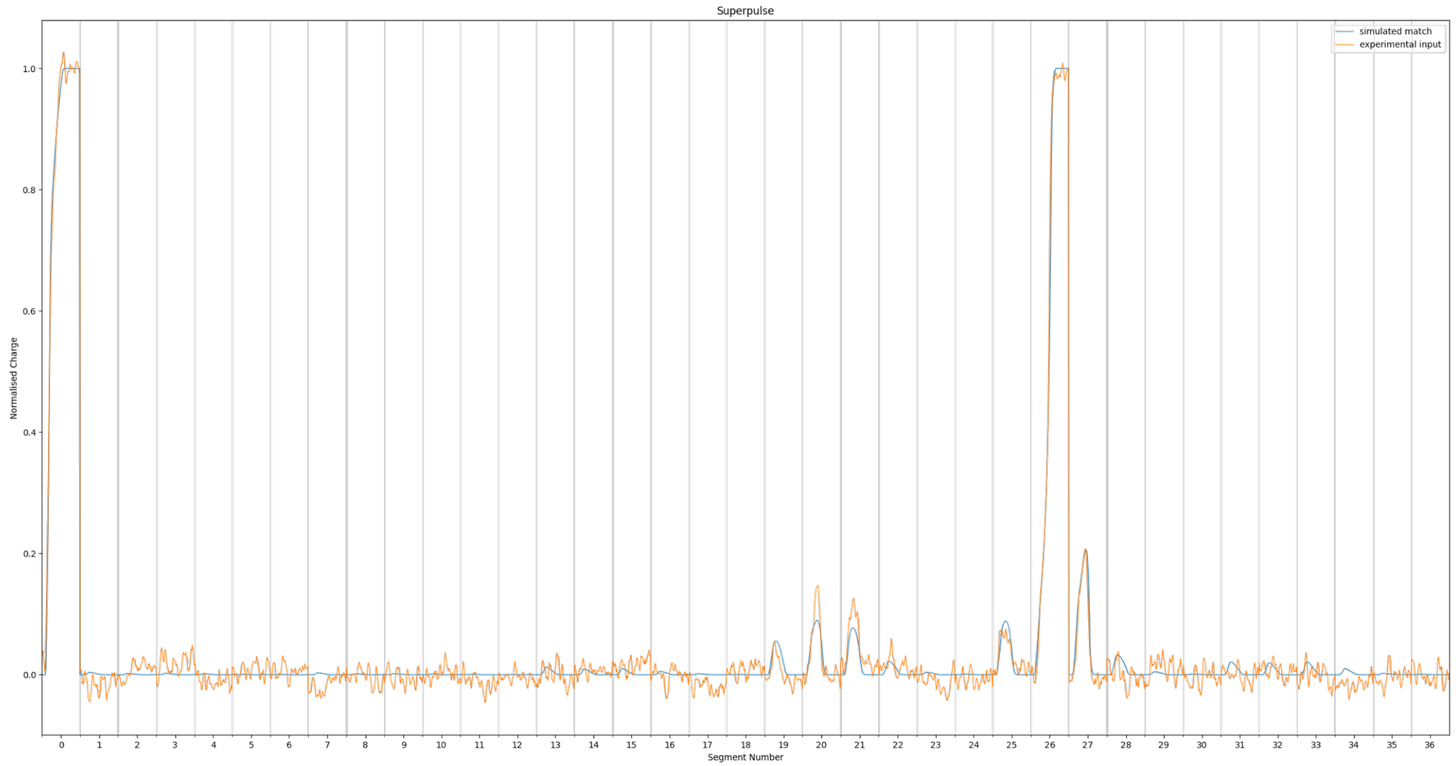
- Coincidence scanning of A005 will be used to validate simulations, ML efforts and PSCS method (IPHC, Strasbourg & Salamanca).
- Will provide a definitive & time-aligned basis for GEANT4.
  - Allows for proper simulations of high-fold events.
- Currently using Caen 1724s, requires GO box & a **lot** of conversion:
  - AGATA → BNC → LIMO → SMA → MCX
- Analysis using MTSort 5.2 (with improved CUDA Trans-piler)
- 1GBq ^{137}Cs source collimated to 1mm on x, y stage, 0.5mm steps.
  - Currently ~ 180 (x, y) scan positions at 2 crystal depths, ~ 1500 usable pulses.
- 90° scatter using BGO array & energy gating (374 & 288keV).
- Several ^{241}Am scans performed at varying resolutions.
- In total ~20 million pulses, 2TB of compressed data.
- Must be converted from EUROGAM → CSV → NPY for PSA.
  - Process will take weeks, ~ 10TB of storage.

Completed Scan positions

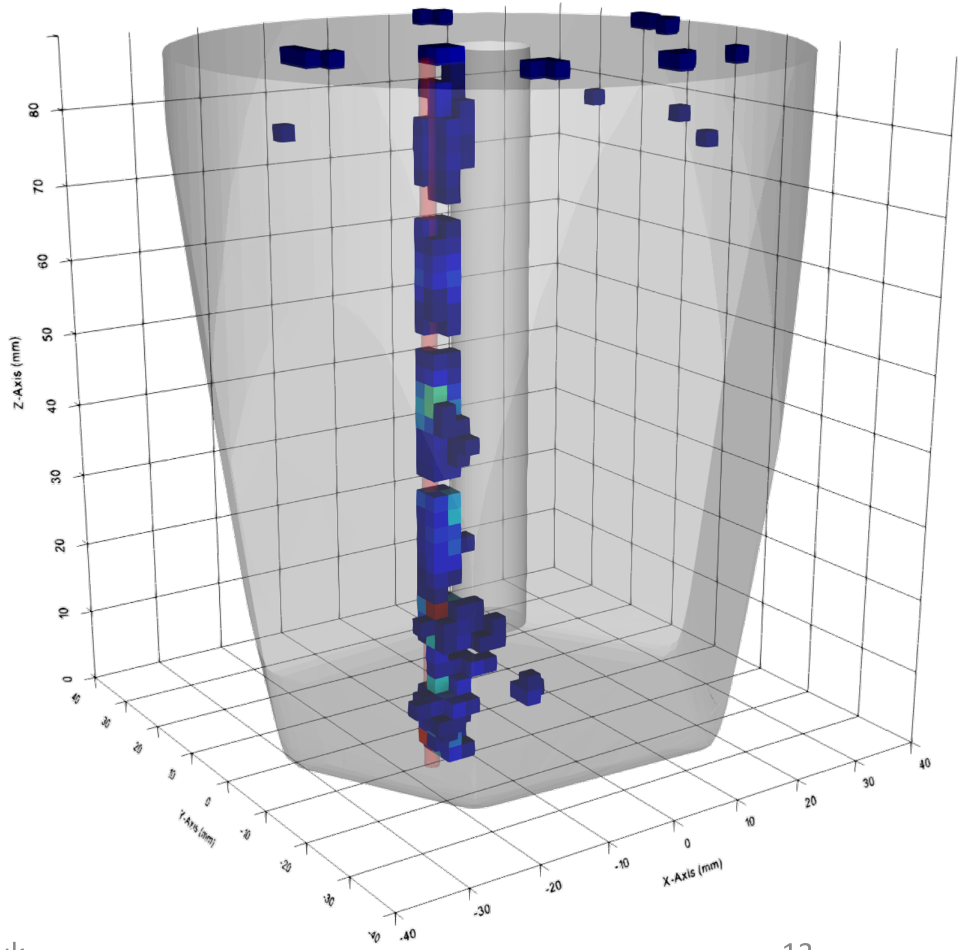


Experimental Results

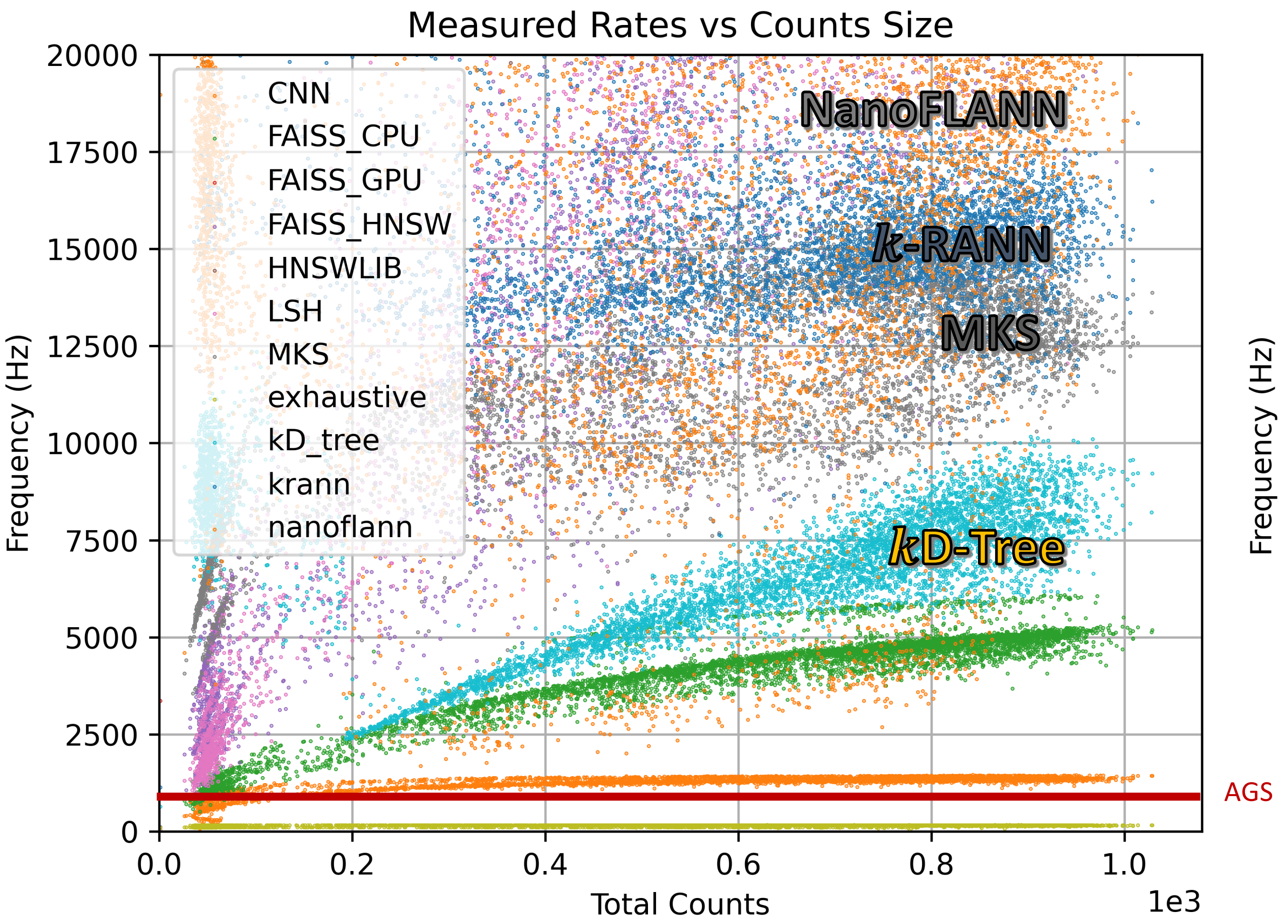
Example Optimum Solution

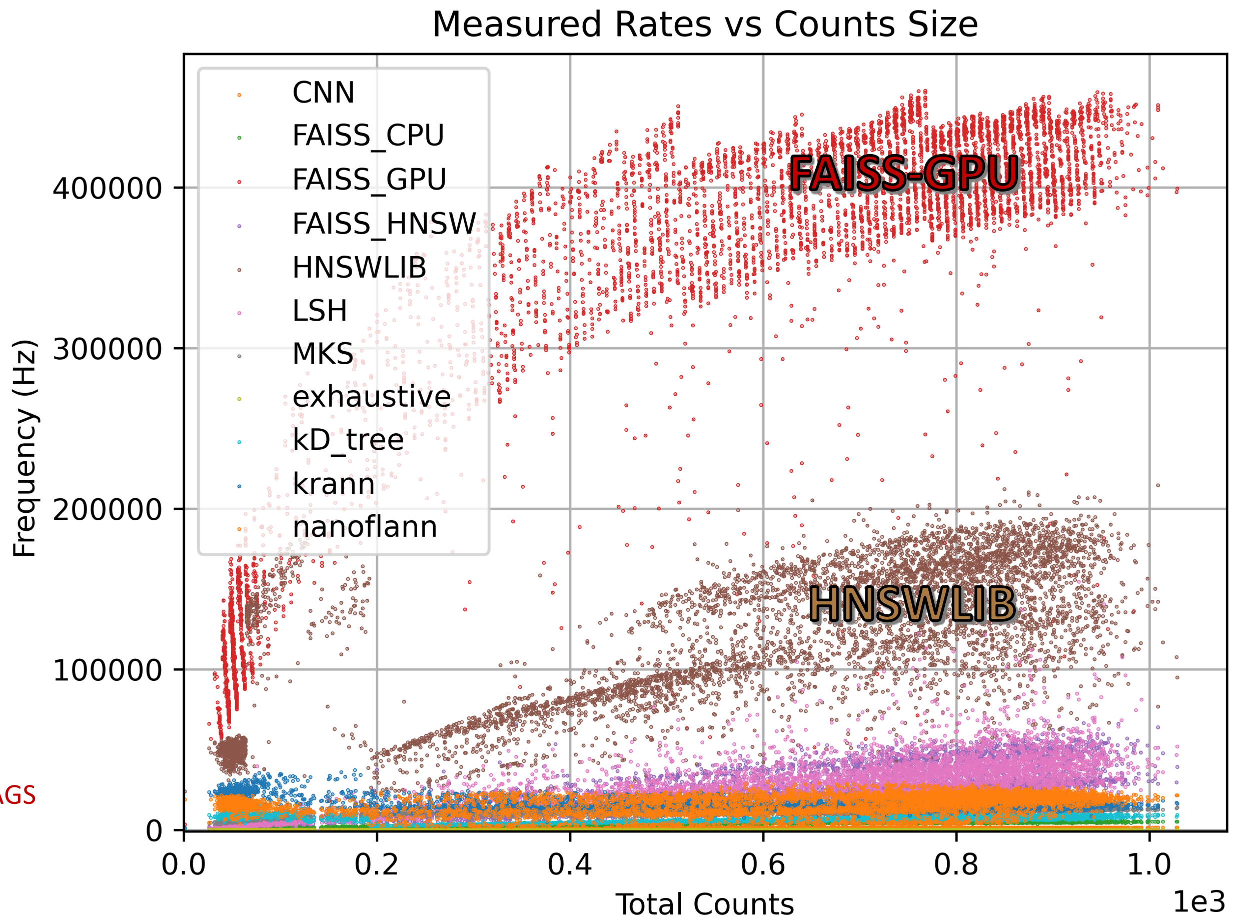


Pencil Beam Reconstruction



Observed Execution Rates



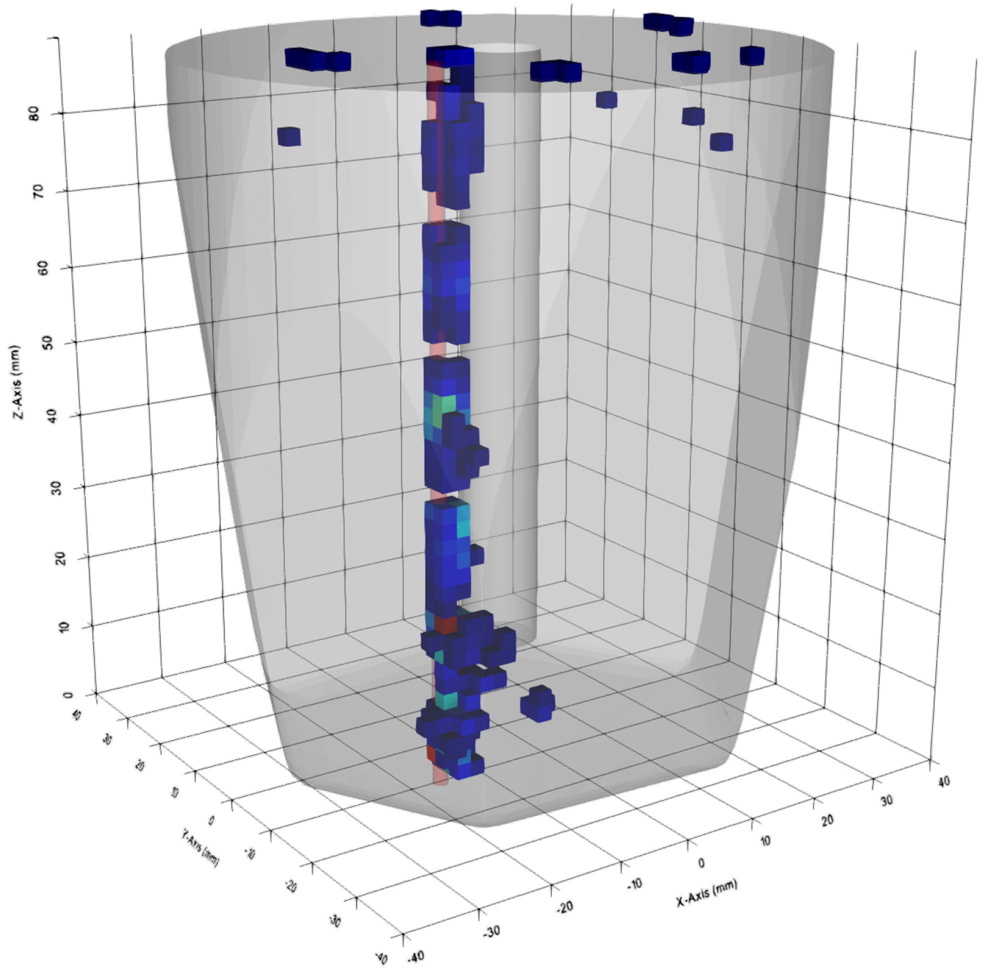


Experimental Results: Pencil Beam Projection

- Some predictions far from collimation beam.
  - Matched pulses suggest Compton Scattering.
    - Cold finger Backscatters?
  - No energy gating used, low energy events?
    - Fold gating imperfect?

- Only first PSA prediction used, TDA methods are k-NN.
  - ∴ No penalty returning top 5 results.
  - λ-Ray Tracking will need adapting.

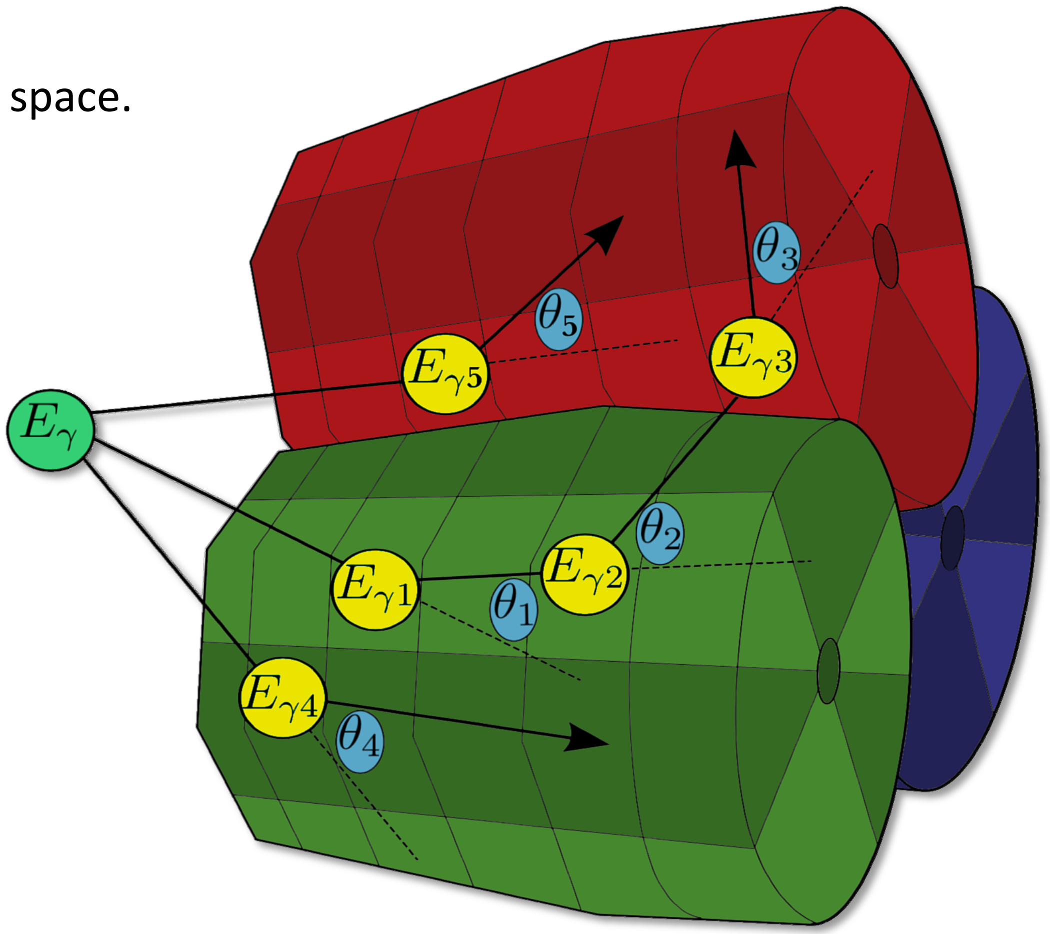
- Current processing rate of ~400kHz, likely larger for more events.
  - Will improve with better hardware (Dual RTX 3090's ?)



The Problem of Higher Fold

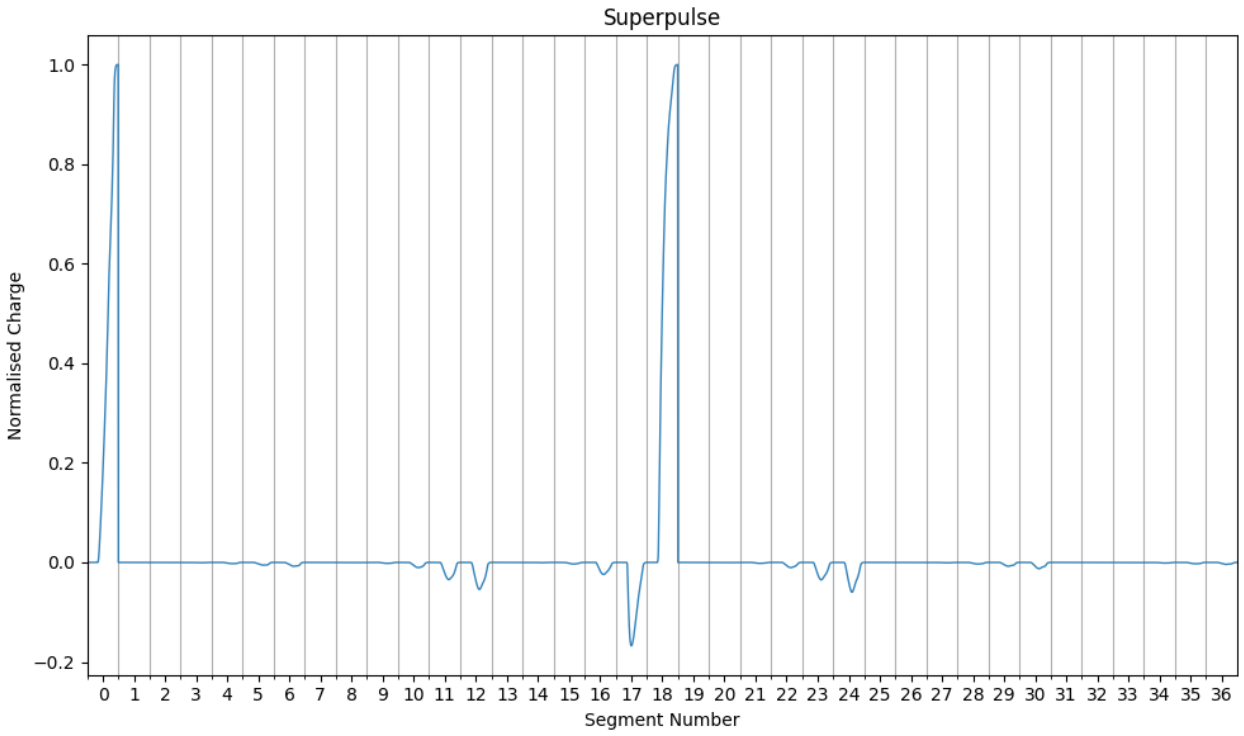
- Within AGATA multiple interactions may occur simultaneously.
- Mixing fraction is unknown for Fold-2_1
- PSA uses normalised signal responses ∴ they exist in the same search space.
- kNN is not designed to search for 2 mixed vectors.
- Precomputing for kNN is '*somewhat*' feasible:
  - 50,000 $\longrightarrow$ 1,250,000,000 minimum points
    - Splitting via hit segments into 1,300 trees would require bunching.
  - Realistically far more due to granularity (n~10).
  - Execution times might not be awful ($\ln(500{,}000) \approx 13$)
  - Exorbitant memory costs but DR reduces this (~8GB for Fold-2_2, $n = 1$)
    - Nvidia A6000 has 48GB of GDDR6X ∴ $n \sim 6$
- Machine Learning models also struggle:
  - No easy way to deconvolve signals (ICA will not work)
  - ∴ No way to form inference

- ***If Fold-2 sounds difficult imagine what Fold-3 is like***

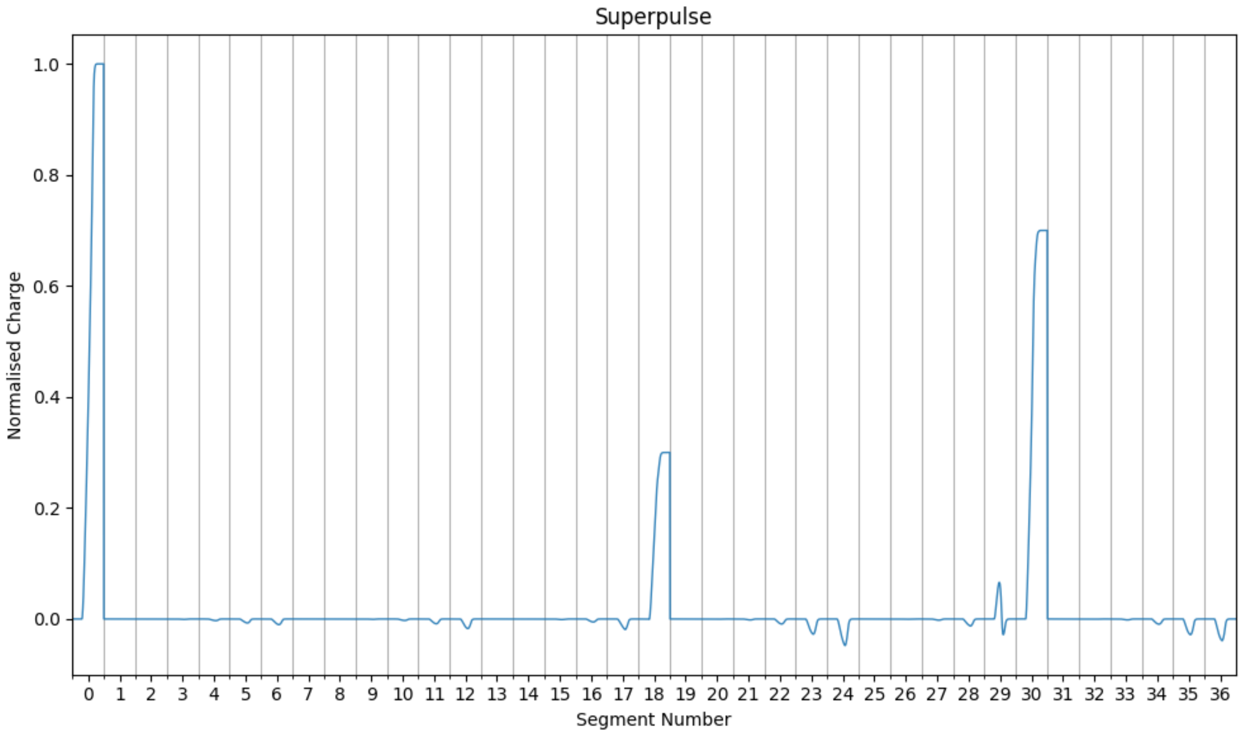


Fold 2 – Coincident Responses

Fold-2: Single Segment, AKA: $Fold-2_1$



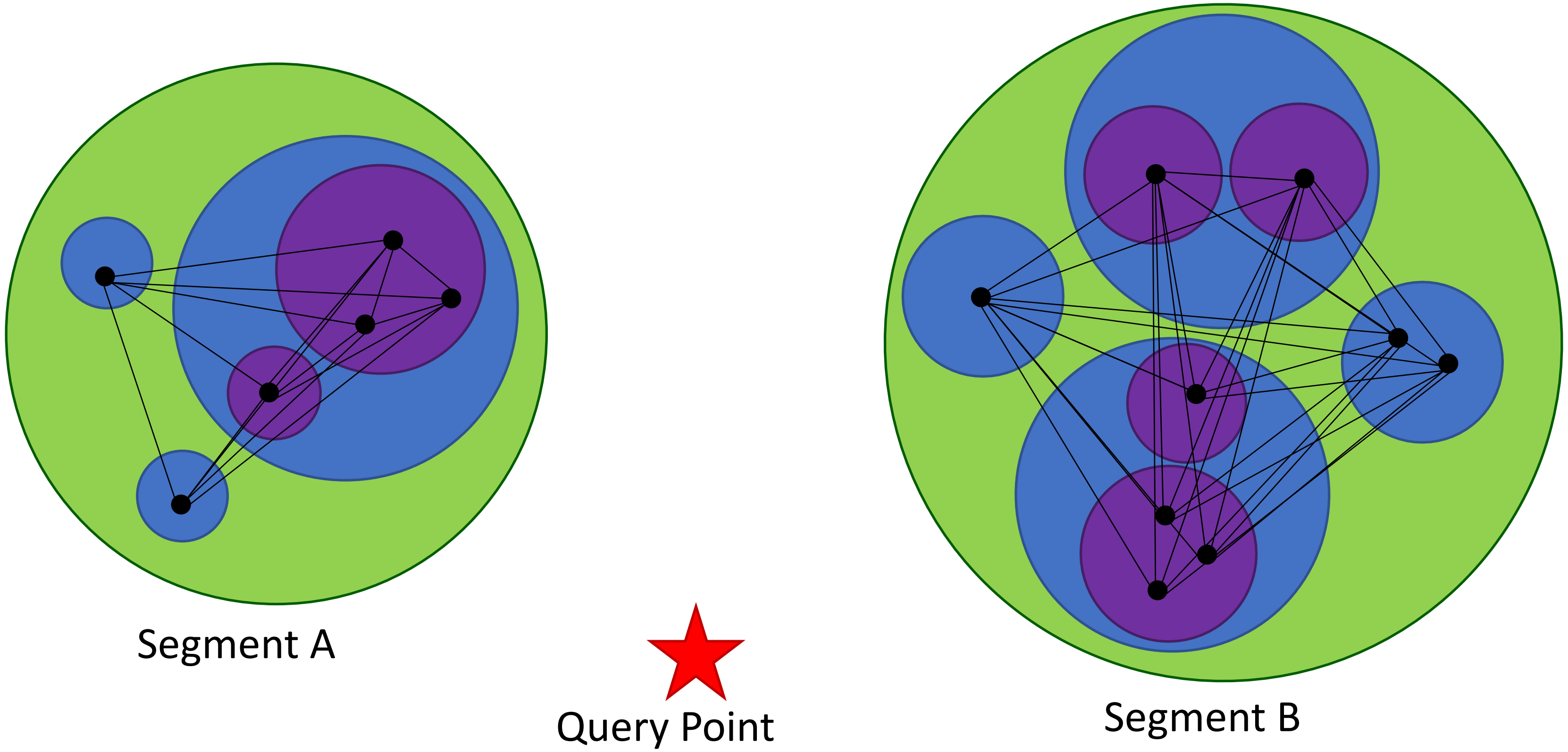
Fold-2: Dual Segment , AKA: $Fold-2_2$



Fold-2 Single segment looks pretty much identical to a Fold-1 response

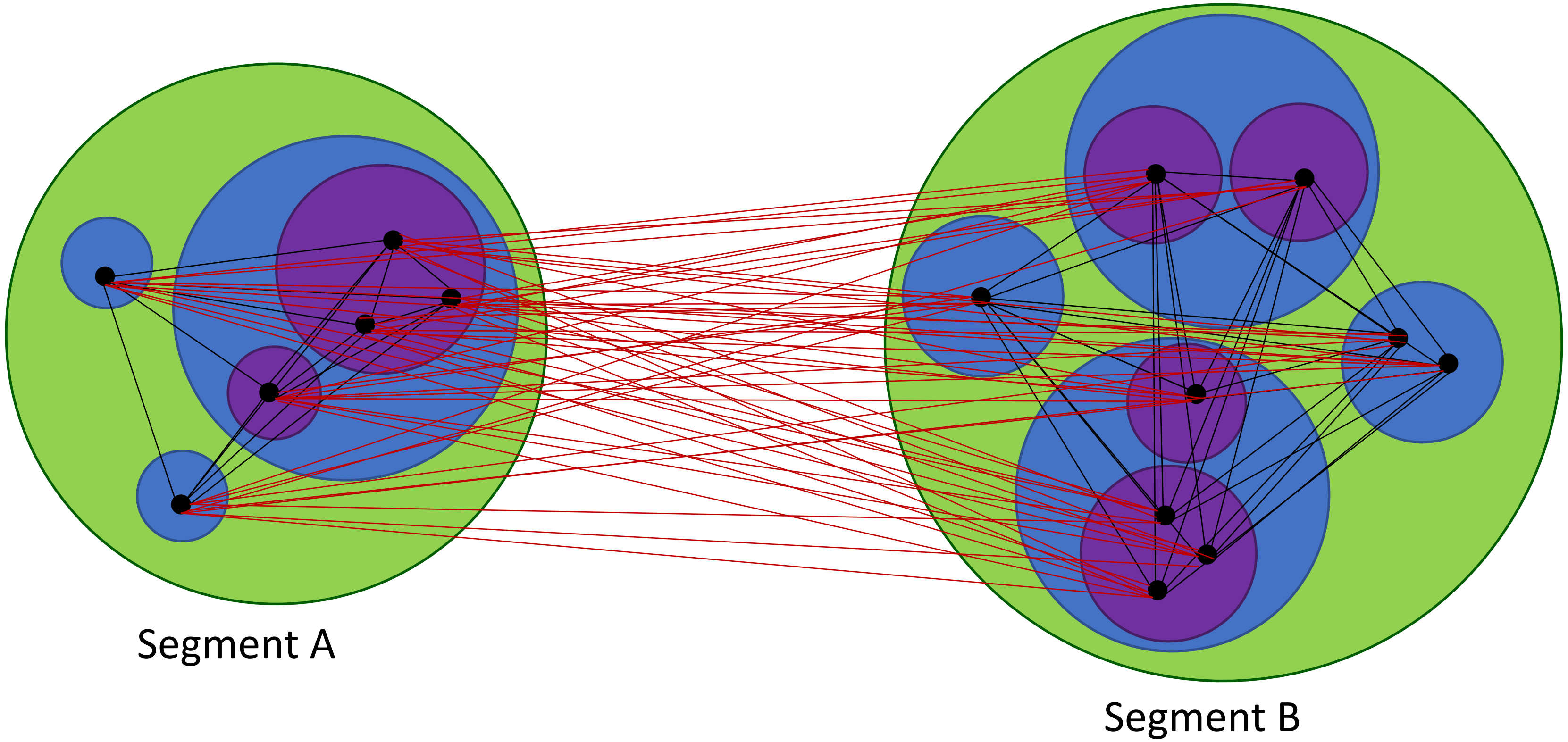
Fold-2 in Response Space: $\boldsymbol{Fold - 2_1}$

- All lines are constrained to within a segment
- ∴ Easy to confuse with Fold-1
- Mixing Ratio (α) unknown

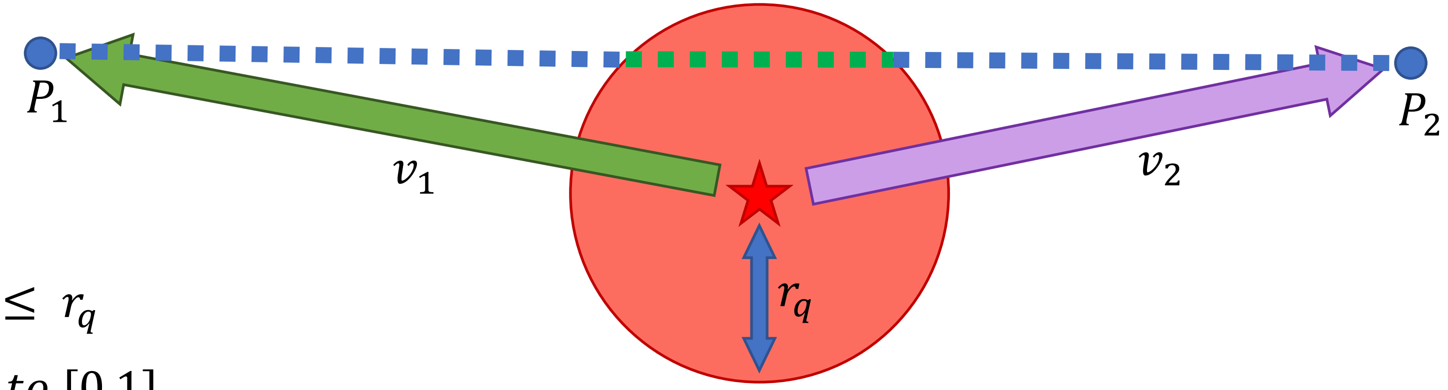


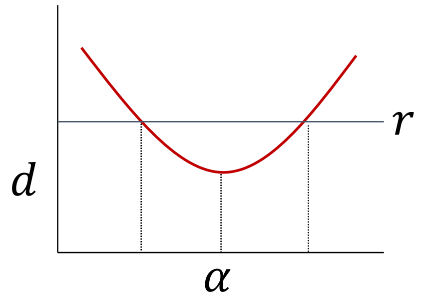
Fold-2 in Response Space: $Fold-2_2$

- Now lines can exist outside the segments
- 97% of Fold-2 combinations are Fold-2_2
- Mixing ratio (α) well known



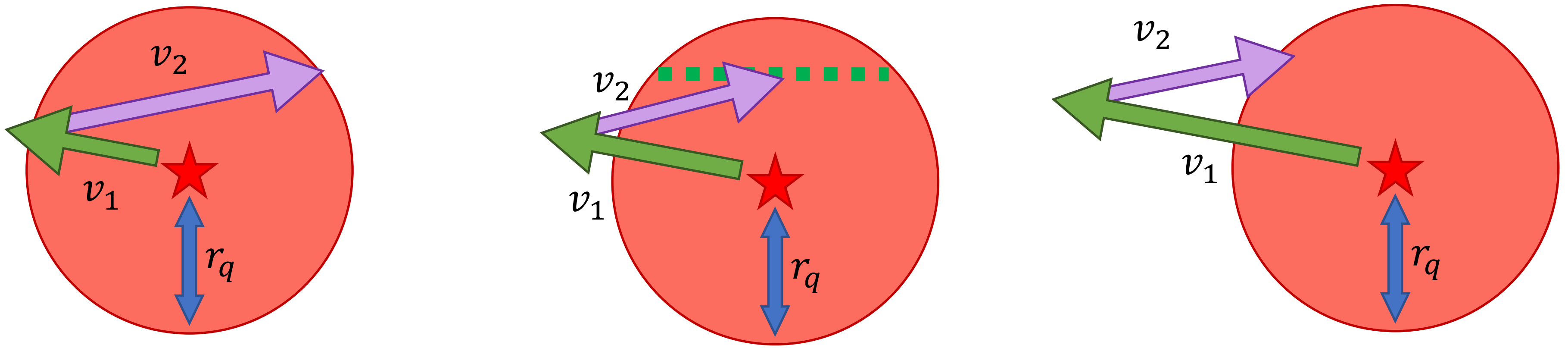
1st Adaption: Solution Cells are now Balls of radius r





$$\|\alpha \cdot v_1 + (1-\alpha) \cdot v_2\| \leq r_q$$

where α *is constrained to* $[0,1]$

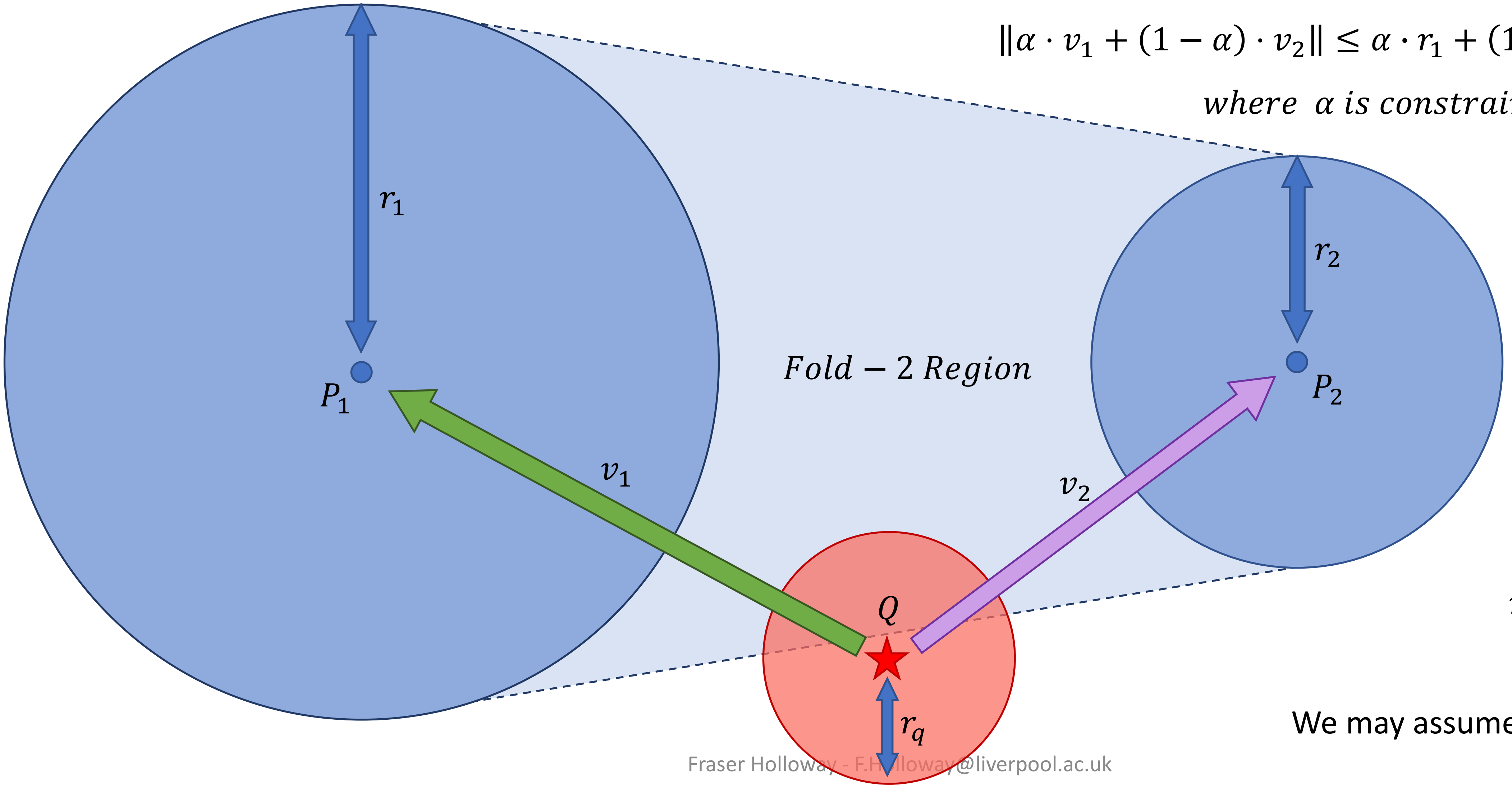


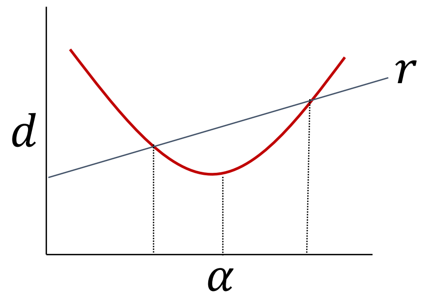
Small α | Optimum α | Large α

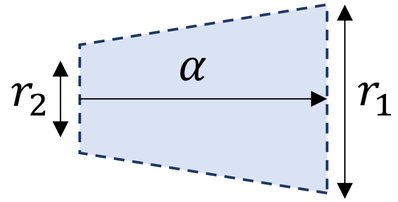
2nd Adaption: Pair cells are also Balls

$$\|\alpha \cdot v_1 + (1-\alpha)\cdot v_2\| \leq \alpha \cdot r_1 + (1-\alpha)\cdot r_2 + r_q$$

$where\ \ \alpha\ is\ constrained\ to\ [0,1]$



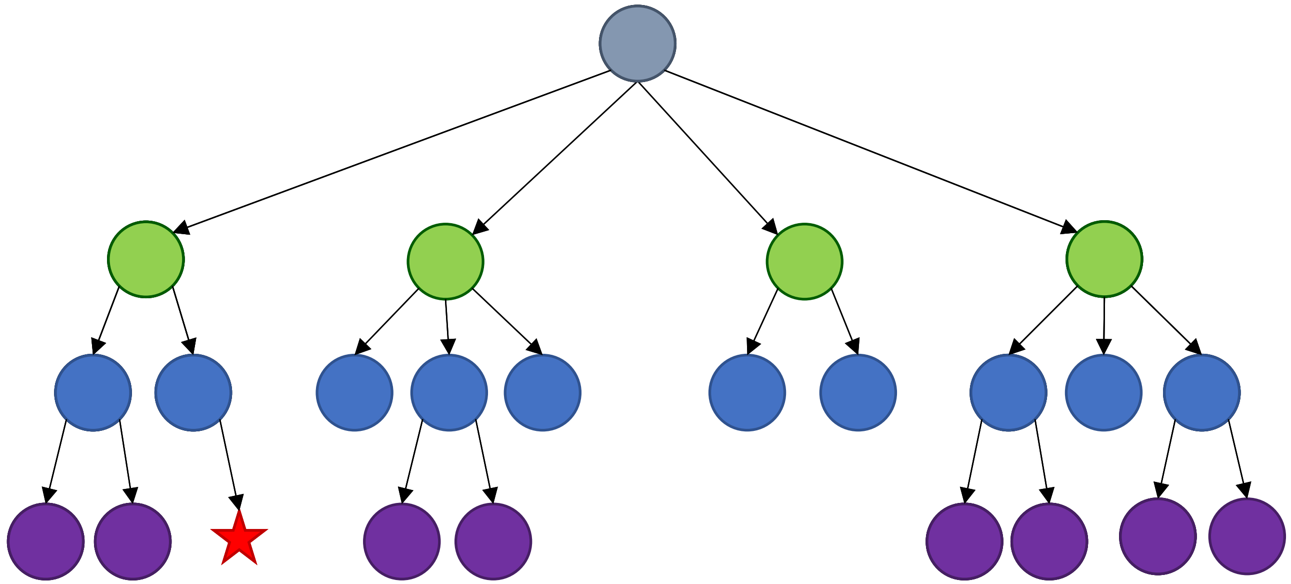


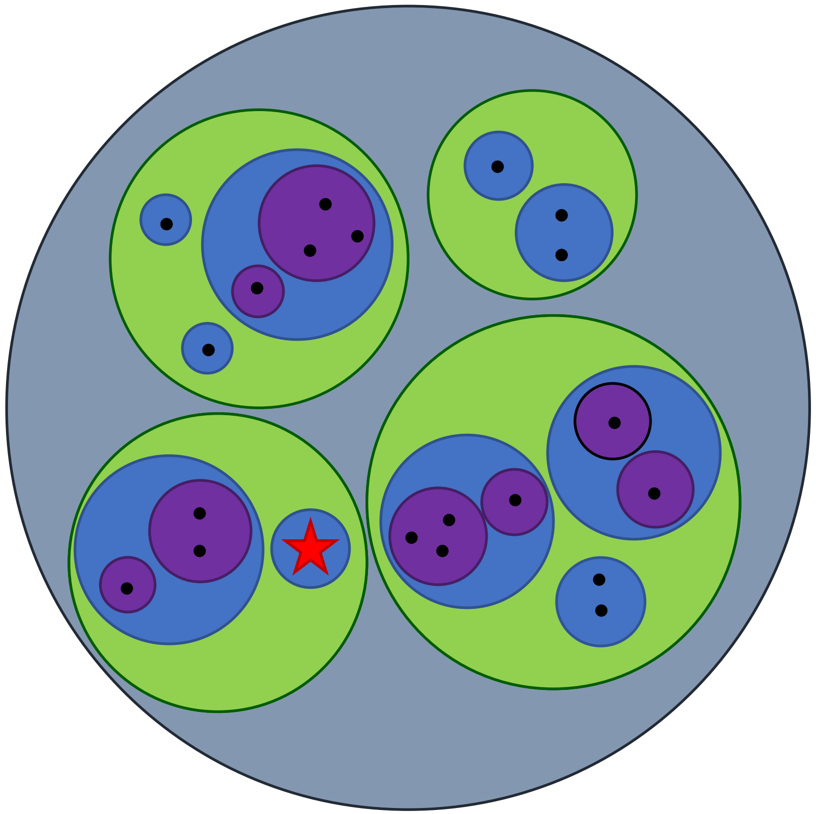


We may assume $r_q \ll r_1\ , r_2$

Recursive Pair Culling on M-Trees

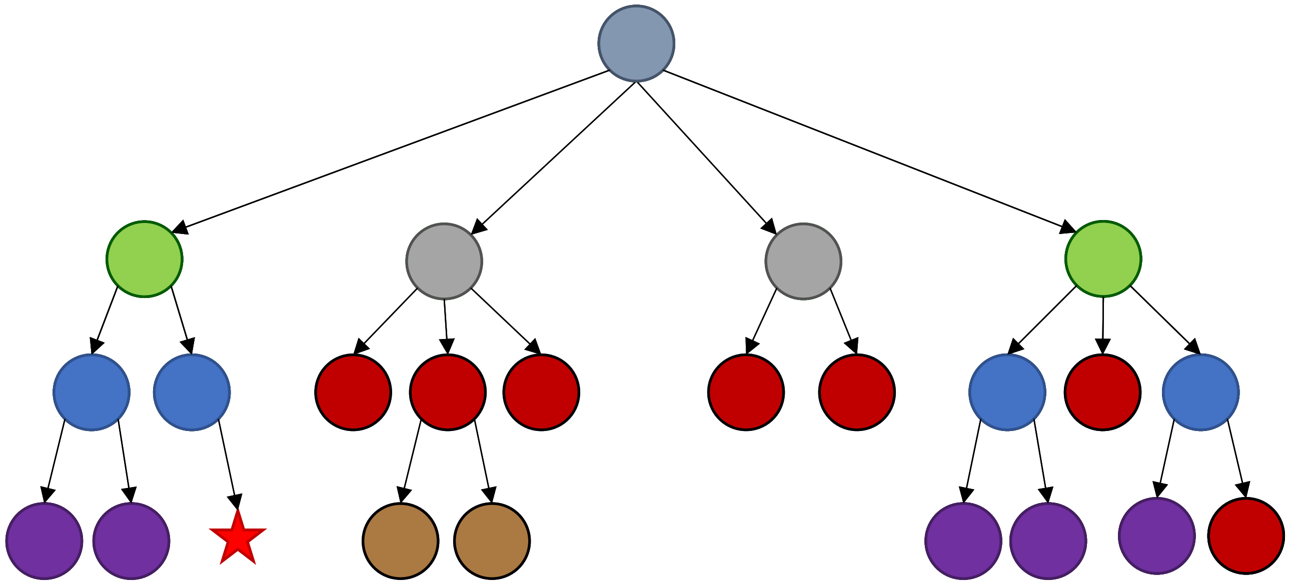
- Metric is applied to a hierarchical graph structure.
- This allows for recursive culling of solutions.



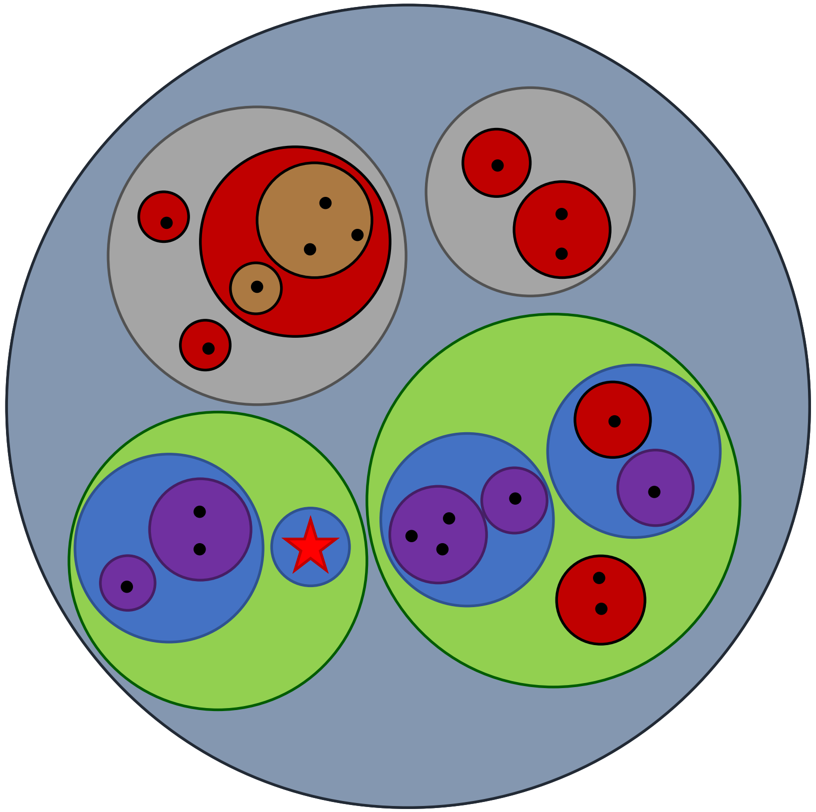


Recursive Pair Culling on M-Trees

- Only a fraction of pairs are calculated & stored.
- Much faster to compute & search.



Culled, **Parent-Culled**, **Child-Culled**



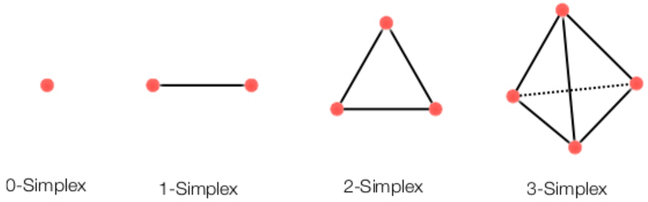
Pros & Cons

Pros:

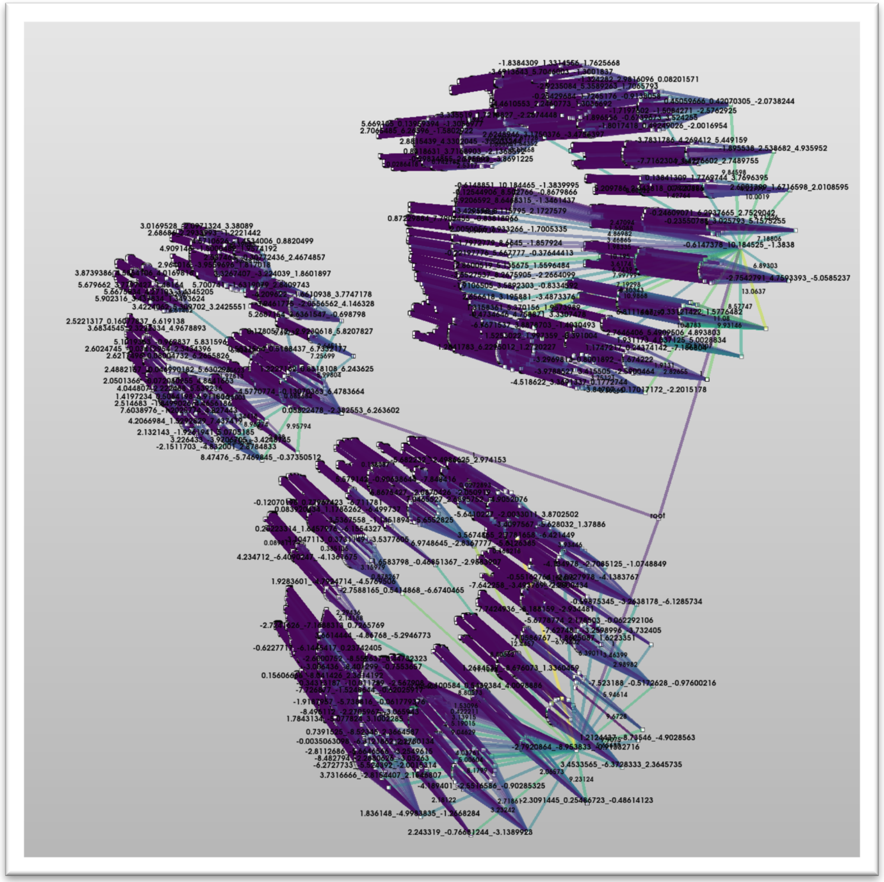
- Selection metric scales well with higher fold:
  - Just add free parameters for other energy fractions
- No complicated high-D geometry to solve (e.g. union of Voronoi Cells)
- All radii & vectors generated during cover tree formation
  - Culling is entirely offline
- α determination at higher levels forms bounds for lower levels.
  - Trivial for Fold-2_2

Cons:

- Metric will only work in linear subspaces.
  - Can't form fast matrix inverse
- Fold-1 solution may be far from query point
  - Add in some Fold-2 solutions
- Pure Python implementation
  - Slow



Realistic M-Tree



Conclusion

- Ongoing work for novel detector simulation using advanced adaptive techniques.
  - Emphasis on cross-detector compatibility & flexibility.
- Development of several methods for k-NN search:
  - MKS, k-NN, KRANN, LSH, NanoFLANN, HNSW, FAISS.
  - Multithreaded, Multi-CPU & Multi-GPU supported.
  - Several times faster than existing AGS
- Initial Experimental Validation looks promising.
- Optimisation of PSA ongoing.
- Fold-n Metric proposed to determine combinations.
- Realistic test on PSA server hardware possible?

Thanks for Listening

Any Questions?

Fraser Holloway – F.Holloway@Liverpool.ac.uk