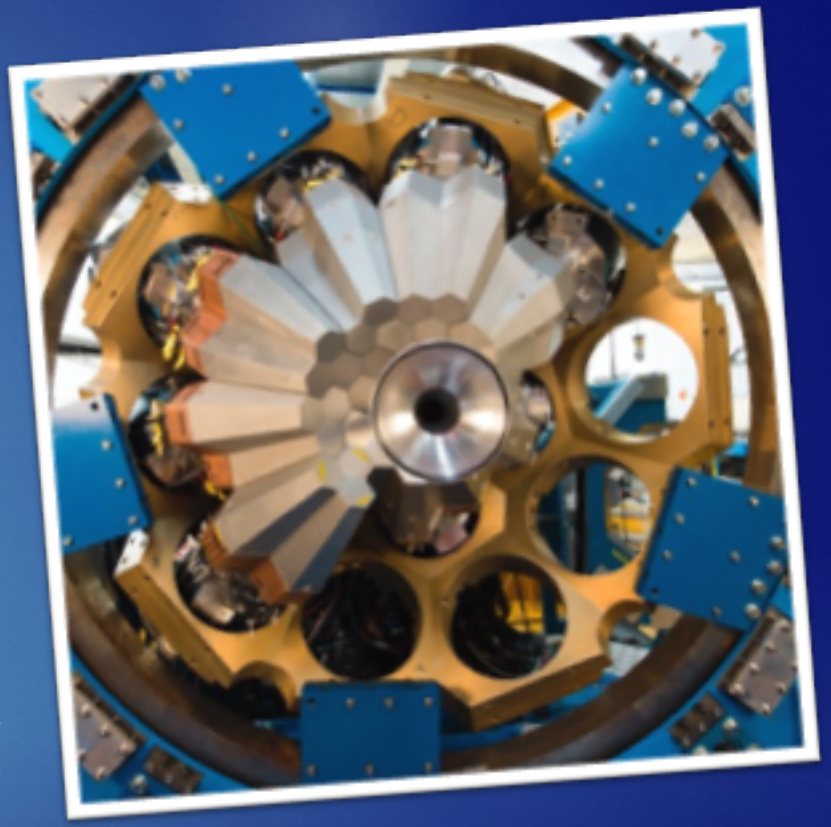
The Development of Novel Pulse Shape Analysis Techniques for AGATA

Fraser Holloway



UNIVERSITY OF LIVERPOOL

LIV.DAT

Science & Technology Facilities Council

AGATA
ADVANCED GAMMA TRACKING ARRAY

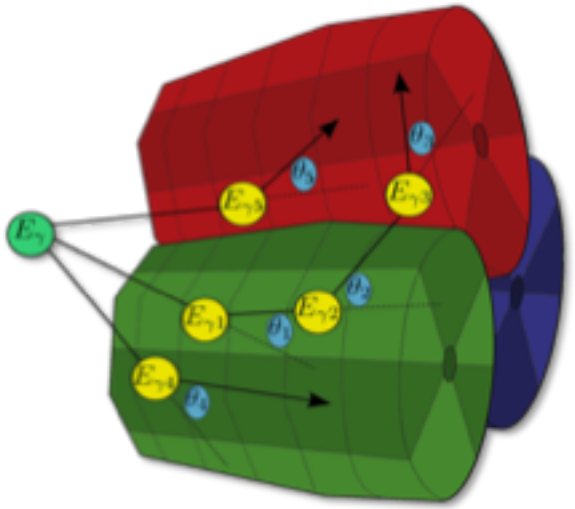
Pulse Shape Analysis (PSA)

- 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.
- 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

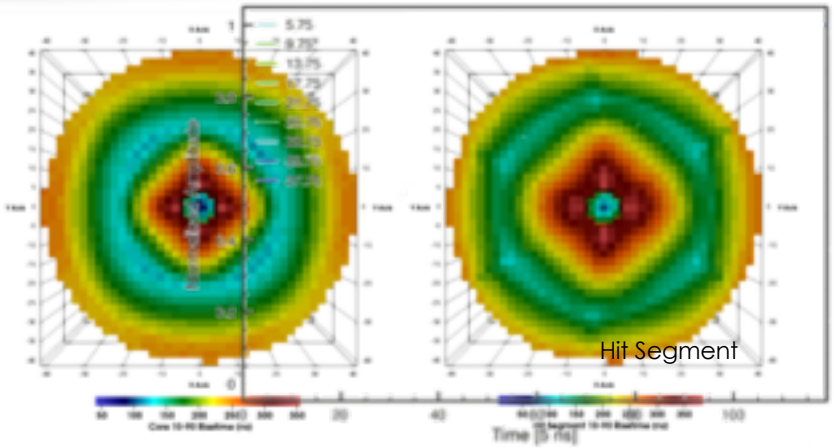
- Other metrics can be used to highlight different sensitivities.
  - Different exponents, weighting for segments (L. Lewandowski, P. Reiter (2019))
  - Time shifting via Dynamic Time-Warping.
- My work is on developing Novel PSA techniques for AGATA.

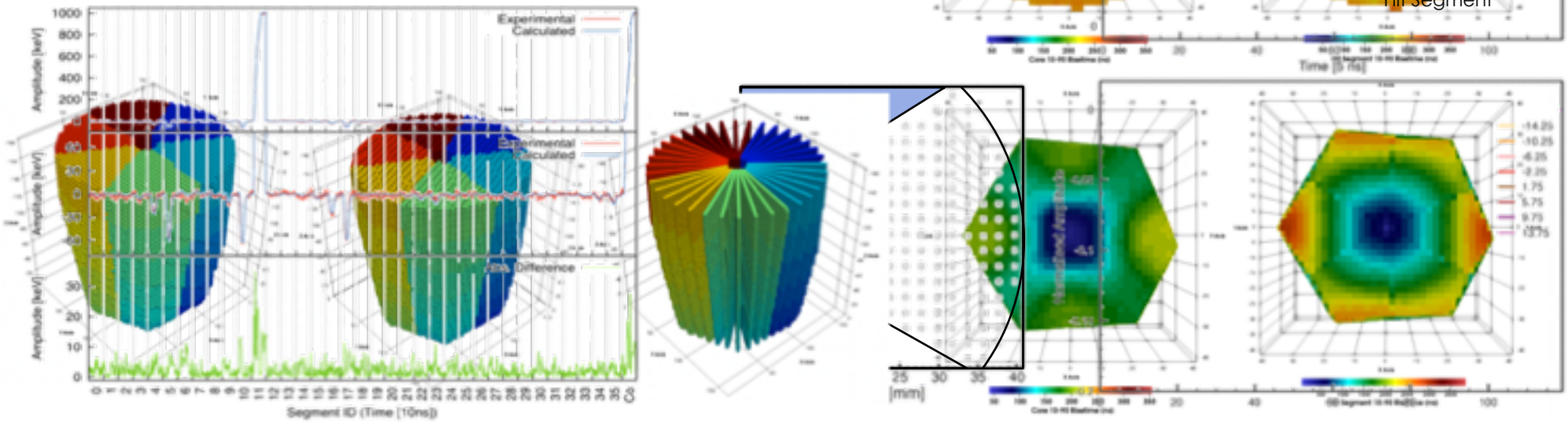


Fraser Holloway - F.Holloway@liverpool.ac.uk

Detector Simulation

- Basis sets are typically precomputed at 2mm cubic grid spacing
- Parametric trends are seen in the data, useful for clustering fold-1 data
  - $T_{10\text{-}90}$, charge asymmetry, knee-point, skewness etc.
  - These parameters are continuous but break down at high fold.
- 6-fold symmetric, polar and tetrahedral basis sets simulated.
- High resolution (0.5mm) basis set generated too.
- Option for dynamic resolution basis sets.



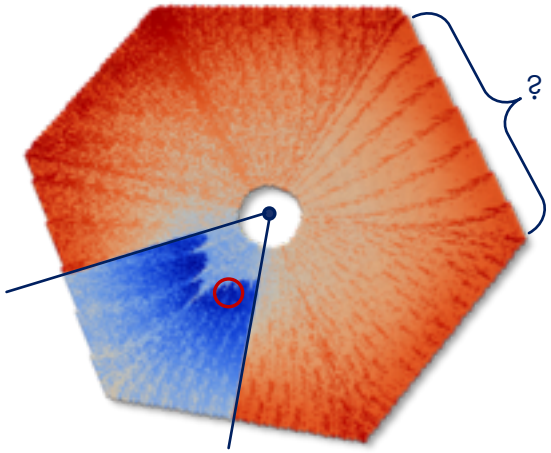


B.Bruyneel – Eur. Phys. J. A (2016)

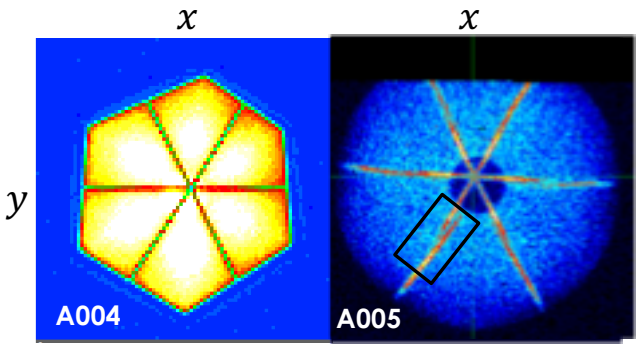
Fraser Holloway - F.Holloway@liverpool.ac.uk

Simulation Limitations

- SIMION Field simulation limited to 0.5mm spacing, ADL3 2mm basis has been hiding issues.
- SIMION front segmentation is wrong, has been since the beginning.
- Odd effects seen at segment boundaries & high resolution:
- Unexplained ‘charge sharing’ between segments.
  - Overlap of SIMION definitions? (Confirmed by Marco AW:2019)
- Sharp discontinuities at edge changes (over-relaxation flaw?)

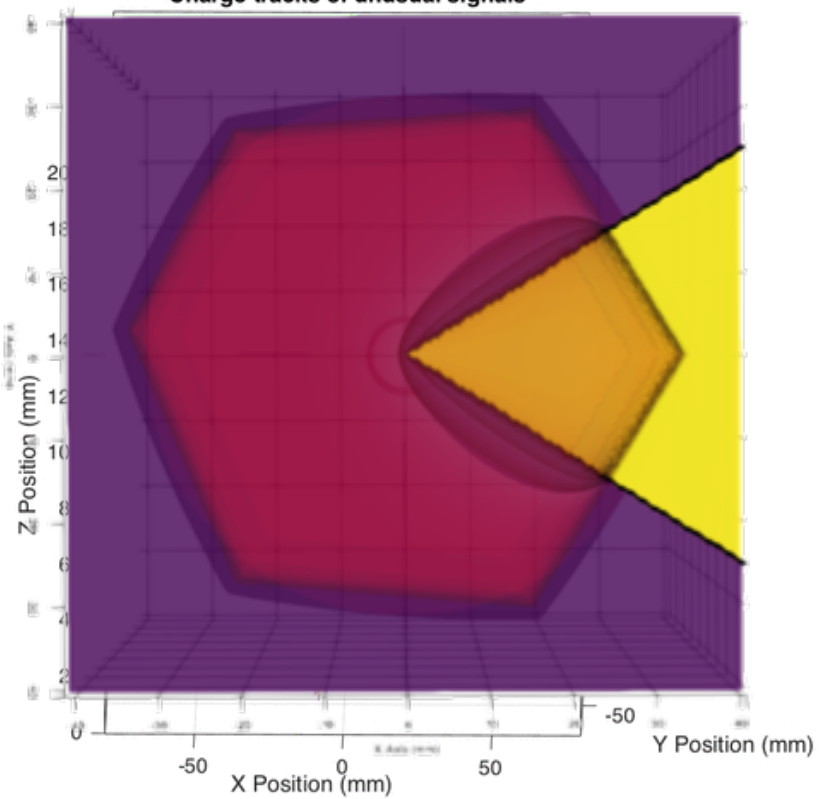


0.5mm FoM Plot showing odd effects
Optimum Circled



ADL3 segment labelling

SIMION Field Segment A1 Definition

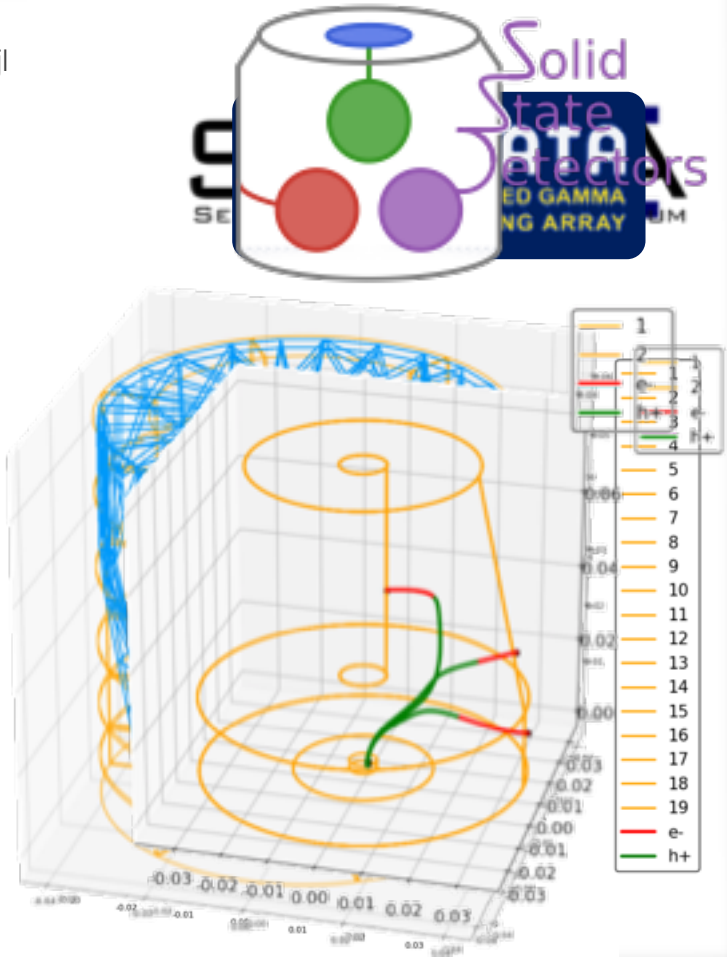


Fraser Holloway – F.Holloway@Liverpool.ac.uk

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:
  - Geometry-on-demand philosophy.
  - ∴ Dynamic resolution possible.
- 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.

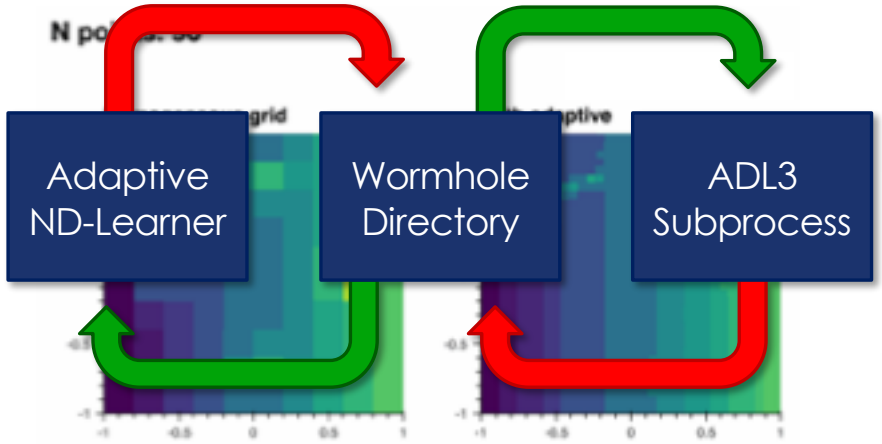
- AGATA crystals are far from simplistic:
  - Difficult to properly define using existing primitives.
  - Instead I added Tri-mesh support into the geometry constructor.
  - Computation is significantly more intensive ∴ I multithreaded it.
- All fields & potentials are generated, CAO needs to be checked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



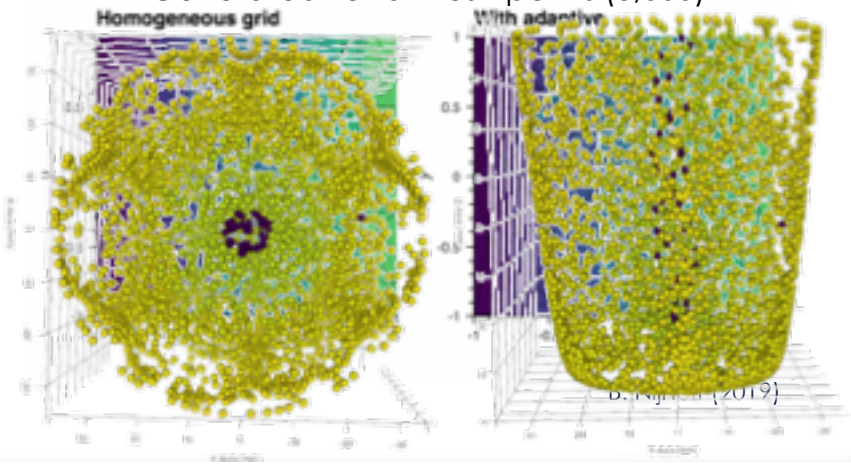
Fraser Holloway - F.Holloway@liverpool.ac.uk

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.
- I modified ADL3 to generate these using a Wormhole Directory
  - ADL is controlled by an Adaptive ND-Learner function.
  - Points generated iteratively, initial coarse Tetra-mesh → fine detail.
  - Resultant structure is stochastic, dynamic resolution.
- Work ongoing adding support for SolidStateDetectors control.



Generated Tetra-mesh points (5,000)



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

Novel Algorithm Development

Several PSA algorithms have been tried for AGATA.

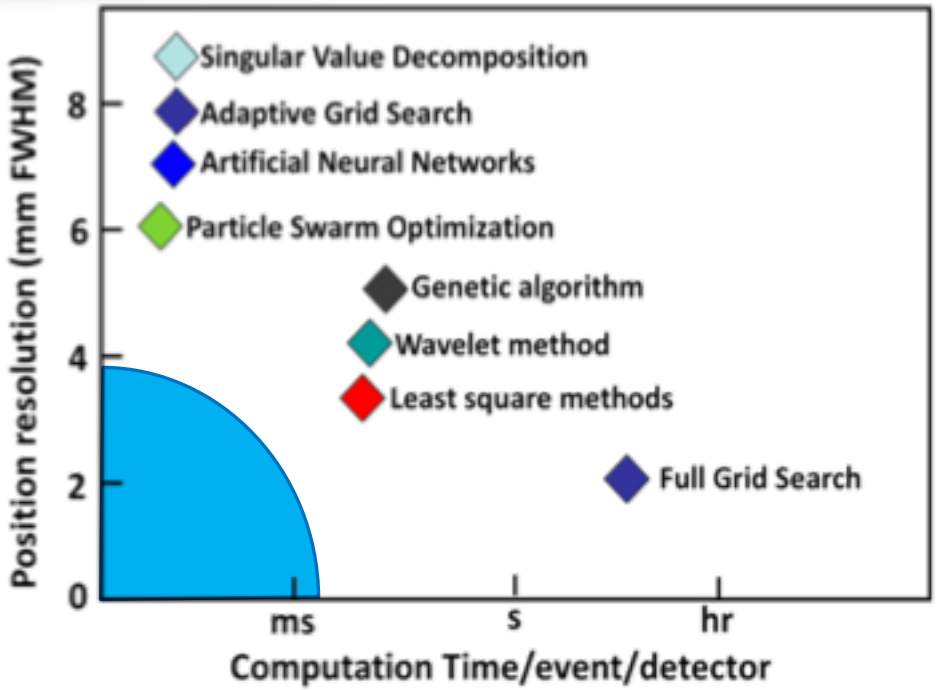
Only ~5% of the basis can be searched using current CPU methods.

There are three different ways to solve this issue:

- Hyper-parallelize the search (GPU acceleration).
- Use more efficient search methods (**TDA**).
- Don't search at all, instead infer locations via training (**ML**).

This becomes a computer science problem

- ∴ Plenty of established fields to learn from.



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

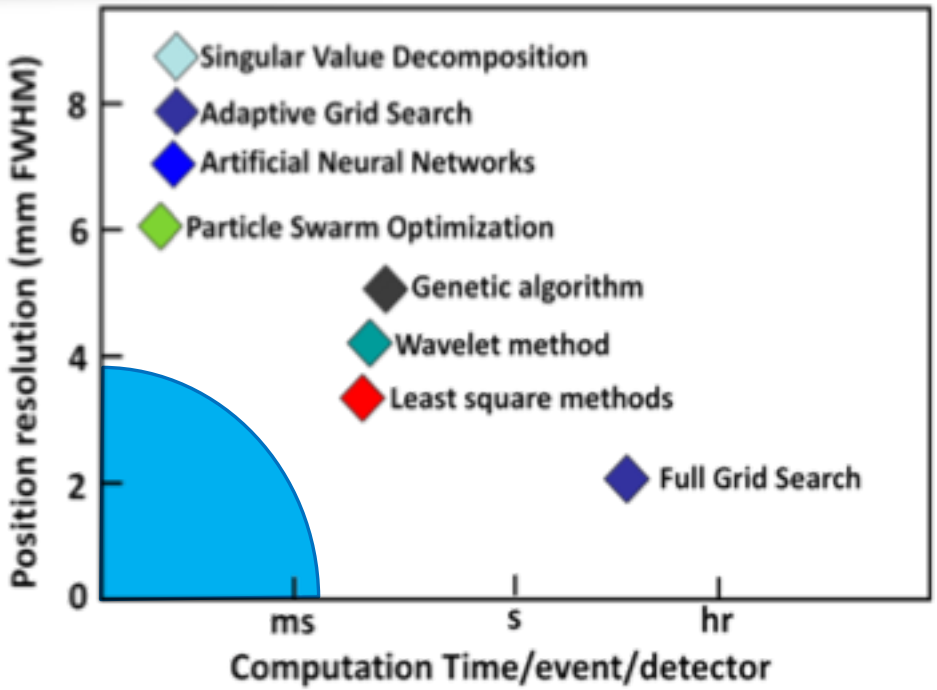
Novel Algorithm Development

Topological Data Analysis (TDA) techniques try to organize data and form efficient search spaces.

- Search spaces are Non-Euclidean
- Generally kd-ball or cover trees used.
- Less prone to local minima.
- Search algorithms aren't naïve.
- Each step made moves search closer to optimum.
- Searching n points can be $\mathcal{O}\log(n)$.

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.



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

Novel Algorithm Development

Tree-based search approaches:

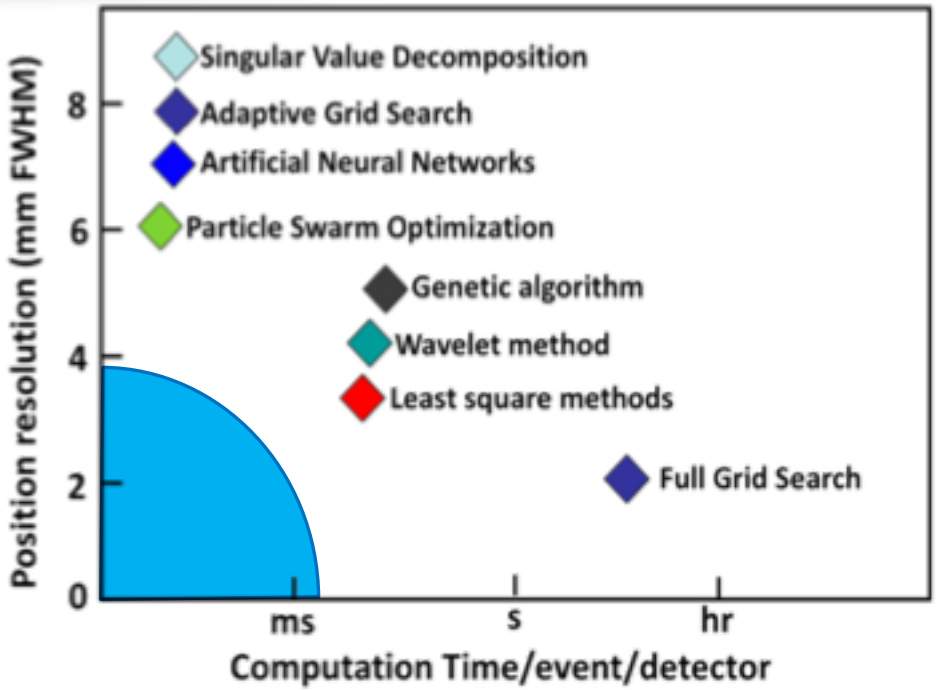
- kNN - k-dimensional Nearest Neighbors.
- LSH – Locality-based Sensitivity Hashing.
- ST/DT MKS – Maximum Kernel Search.

Machine Learning options:

- Signal Classification.
- Regression (CNN).
- Autoencoding/Fingerprinting (β-VAE).

Other options:

- GPU Accelerated parallel search.



- **All Algorithms have been tested with Gaussian Noise, experimental noise to be determined.**
  - Performance is likely to decrease.
  - Will know more once signals are properly analysed.

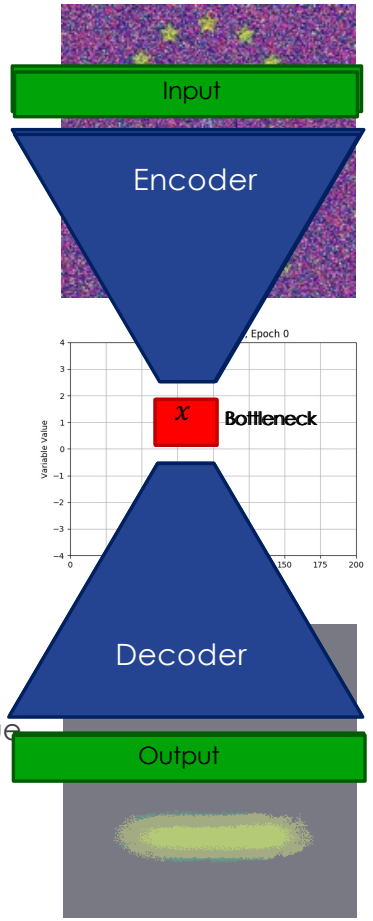
Fraser Holloway - F.Holloway@liverpool.ac.uk

Machine Learning in Pulse Shape Analysis

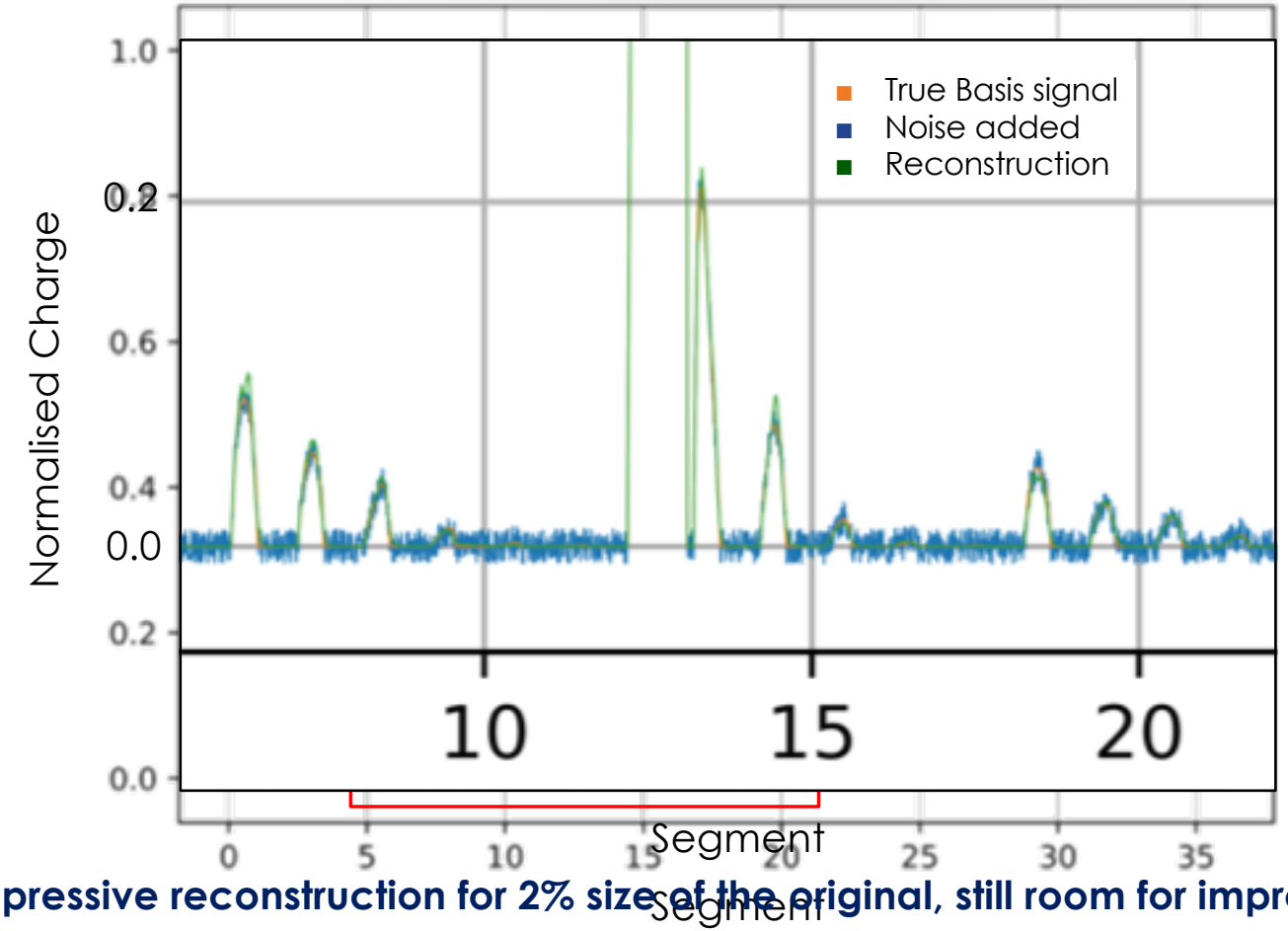
- Most of my Machine learning work hasn't progressed much since AGATA Week 2019
- It'll be revisited once experimental analysis is completed.
- As such I've moved most of the slides to the appendix.
- The work can be discussed if we have time.

Autoencoders for Tagging & Compression

- Autoencoders combine two separate networks to function:
  - Encoder: converts input to a learned latent space via feature extraction.
  - Decoder: converts latent space into a reconstructed output.
- Autoencoders are **incredibly** efficient however can be lossy.
- The network effectively replicates a denoised input.
  - Signal is intelligently denoised, small transients are unaffected.
  - Network doesn't see noise as useful information.
- Current Execution time ~56μs however will likely change.
- Autoencoders become more useful when split into parts:
  - The Encoder and Decoder compress data far better than traditional methods.
  - The latent representation can be used to express parametric trends.
    - This requires disentangling the latent space (difficult)
    - Can this be used for tagging?
- Compression isn't necessarily bad, oddly the reconstructed pulses could end up being better than the inputs due to denoising.
- If the Autoencoded signal is significantly different to the real signal this suggests that the signal is weird
  - Multiple hits in segment?



Example Reconstructions, ~44x Compression Ratio

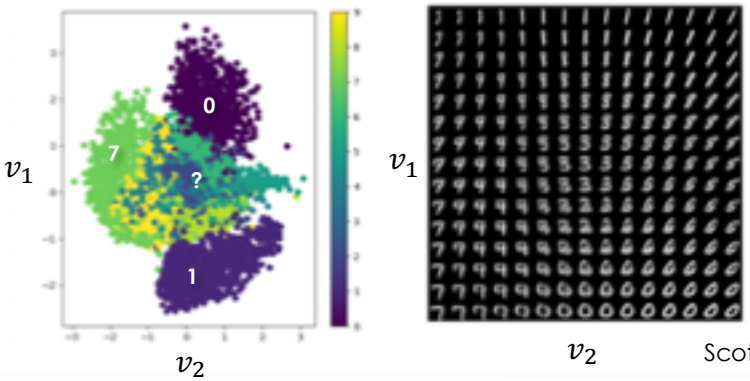


Impressive reconstruction for 2% size of the original, still room for improvement.

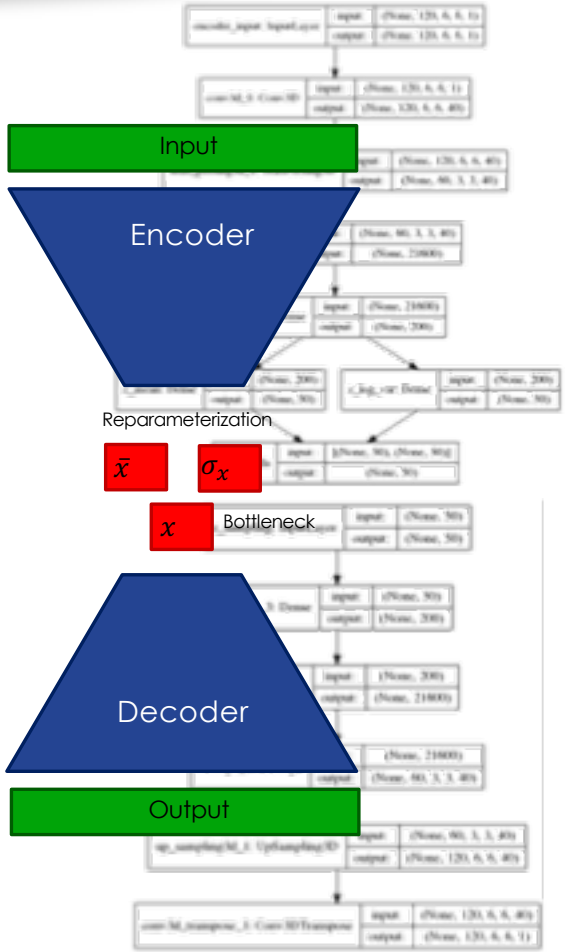
Disentangled Autoencoders

- Typical AE bottlenecks are difficult to interpret manually.
- Optimum bottleneck size is unknown, how many variables contribute?
- DAE attempt to maximize the usefulness of the latent representation.
- This is done by making each latent variable strongly independent.
- Each latent variable should represent a different parametric trend.
  - Latent space should be separable.
- Latent representation ***should*** be fold-invariant.
- Perform MKS on latent representation.

MNIST set example



Scott Freitas (CSE 591, 2018)

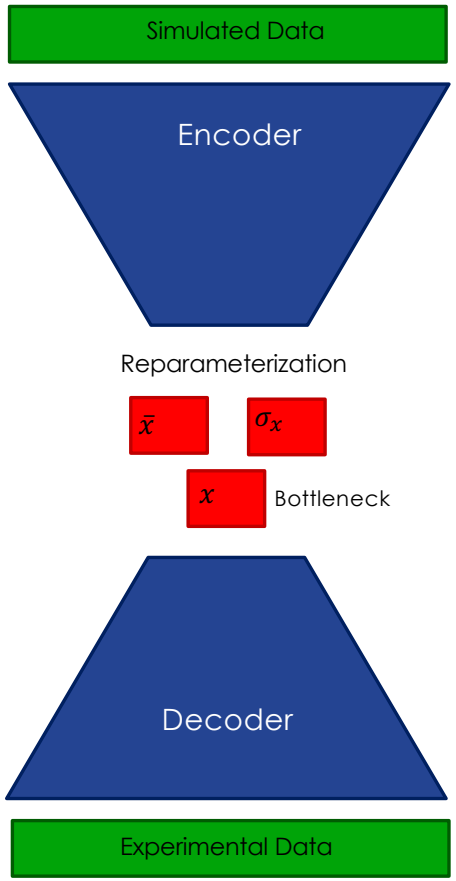


Autoencoders for Basis Correction

- PSA and GRT perform differently when given real & simulated data.
- Therefore there's likely some form of discrepancy between the two.
- How about using ML to transform simulated into real data?
- Simulation reduced to latent space & then reconstructed to experimental.

- This approach requires very good experimental data:
  - Full x, y, z characterisation of the crystal.
  - No guarantee that trained model can be adapted to different crystals.

- Validation data for A005 will be taken anyways.
  - May as well test the feasibility of this method.
- Transform of preamplifier response also possible.



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

GPU Acceleration

- GPUs have advanced significantly (10x) since the last AGATA investigation.
- GPU acceleration can be used on embarrassingly parallel problems:
  - Exhaustive search.
  - Adaptive Grid search (two step).
  - Matrix manipulations.
    - Figure of merit (although matrix sum $\mathcal{O}\log_2(n)$)
- Shared memory makes things complicated.
- Multiple languages can utilise GPU accelerated code:
  - C, C++ (NVCC).
  - Julia (CuArrays).
  - Python (with Numba, Scikit-CUDA).
- Programs can be compiled to use NVBLAS:
  - MLPACK (Armadillo).

- GPUs are **very** powerful for ML approaches.

Nvidia Quadro RTX5000

10.63 TFLOPS

Nvidia Quadro P5000

8.9 TFLOPS

Routine	Types	Operation
GEMM	S, D, C, Z	Multiplication of 2 matrices.
SYRK	S, D, C, Z	Symmetric rank-k update
HERk	C, Z	Hermitian rank-k update
SYR2k	S, D, C, Z	Symmetric rank-2k update
HER2k	C, Z	Hermitian rank-2k update
TRSM	S, D, C, Z	Triangular solve (right angled)
TRMM	S, D, C, Z	Triangular matrix-matrix multiply
SYMM	S, D, C, Z	Symmetric matrix-matrix multiply
HEMM	C, Z	Hermitian matrix-matrix multiply

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

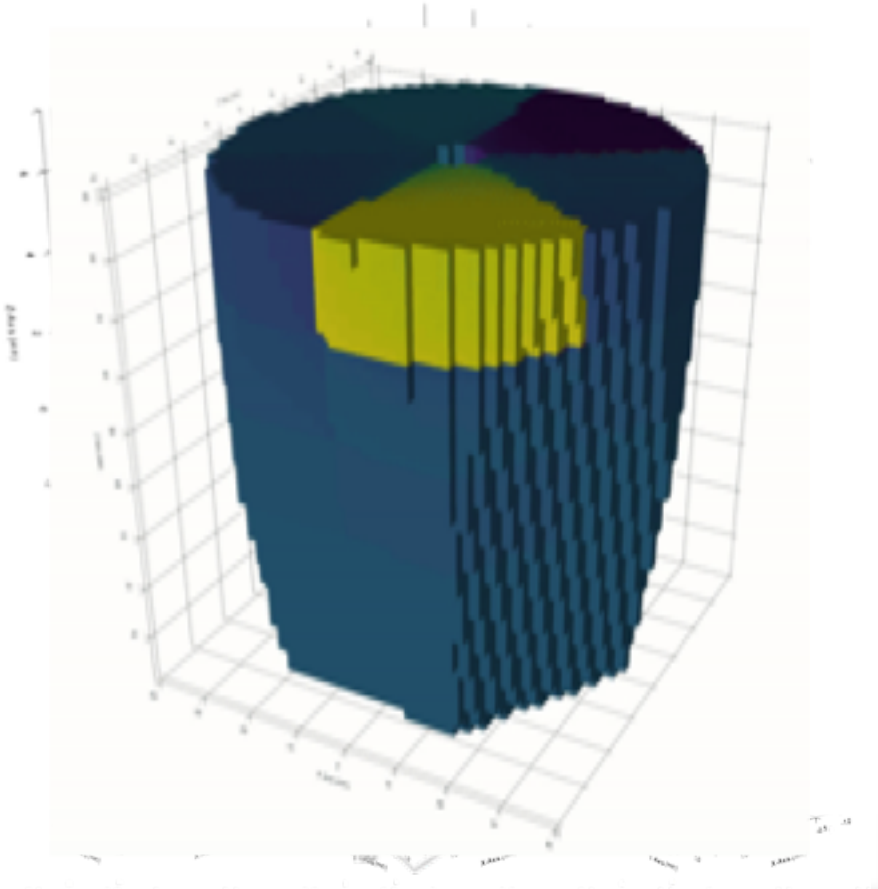
Automated TDA Searching

- Established C++ Library MLPACK used for KNN & MKS operations.
- GPU acceleration possible using NVBLAS.
- Additional Python API & Command line interfaces available.
- Modular design allows for custom Figures of Merit, segment handling.
- Prefers smooth & convex search spaces.
  - Doesn't like searching multiple segments.
    - Metric penalizes segments far from interaction.
- *Should* work for multiple interactions within the same segment.
  - Combinations need to be precomputed.
    - Outrageous memory costs if implemented.
      - PCA transform mitigates this
- Currently 3 techniques look applicable to Fold-1 searches:
  - kDT
  - LSH
  - **MKS**

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

Fast-MKS Searching

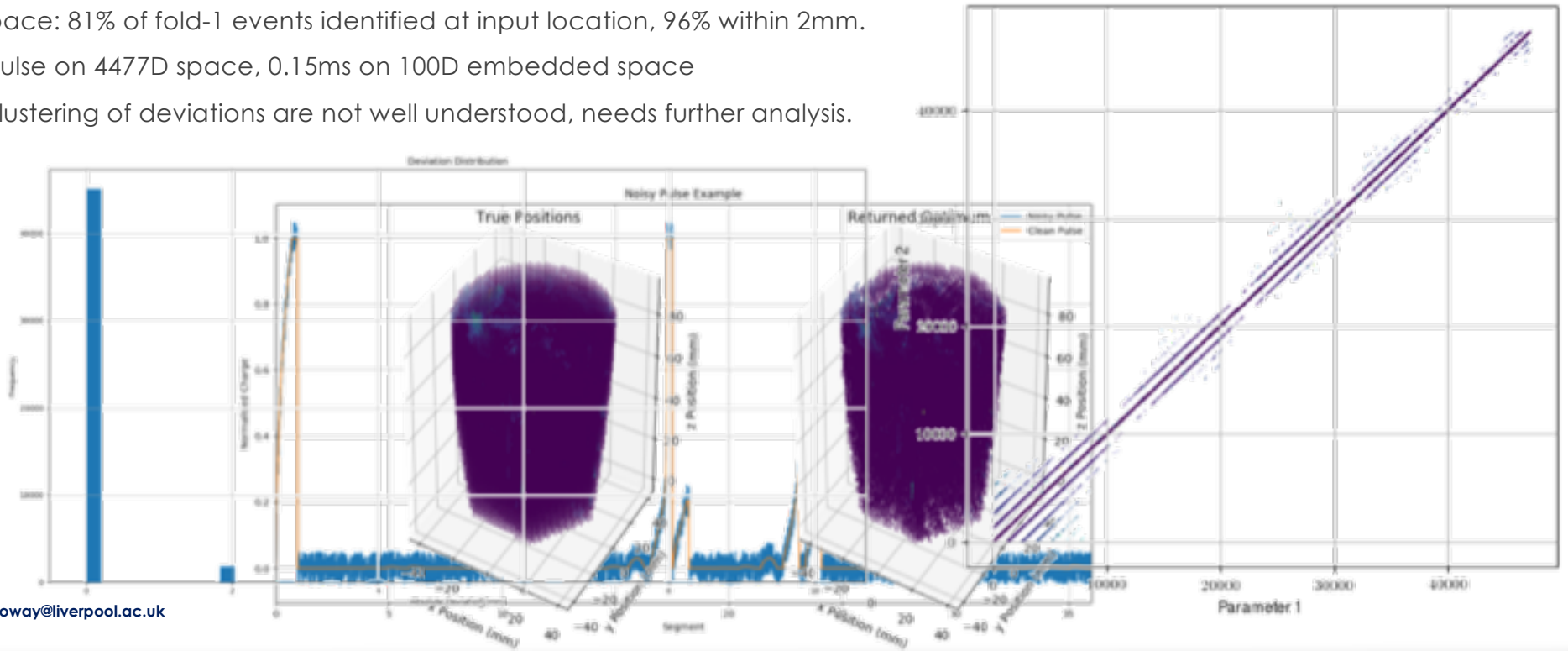
- Fast Maximum Kernel Search uses two trees to search an ordered data structure.
- First tree is used to convert reference set into structured data.
- Second tree is then dynamically built using query set.
- Efficient comparisons mean that the space can be searched quickly.
- Self-navigation allows for non-Euclidean search space ∴ PCA, ICA & βVAEs
- Lower-dimensional space requires preserved homology:
  - Check MST, connectivity, point density.
- Mercer Kernels allow for modifications of phase space, improve separations.
  - More complex kernels have execution penalty.
  - Cosine kernel offers best tradeoff.



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

Fast-MKS Preliminary Results

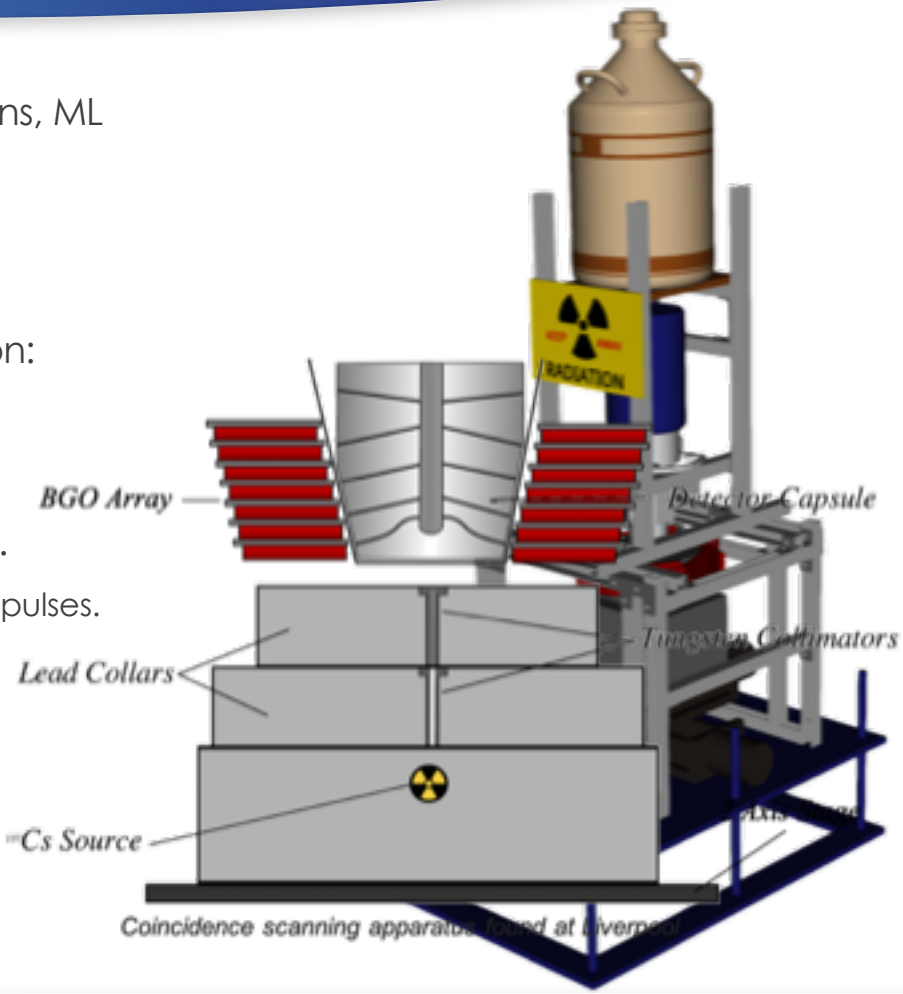
- 10% Gaussian noise added to simulated database for preliminary validation.
- MKS with Cosine kernel used to return top 5 solutions of kernel search with confidences.
- On 4477D space: 95% of fold-1 events identified at input location, 99% within 2mm.
- For 100D space: 81% of fold-1 events identified at input location, 96% within 2mm.
- ~1ms per pulse on 4477D space, 0.15ms on 100D embedded space
- Currently clustering of deviations are not well understood, needs further analysis.



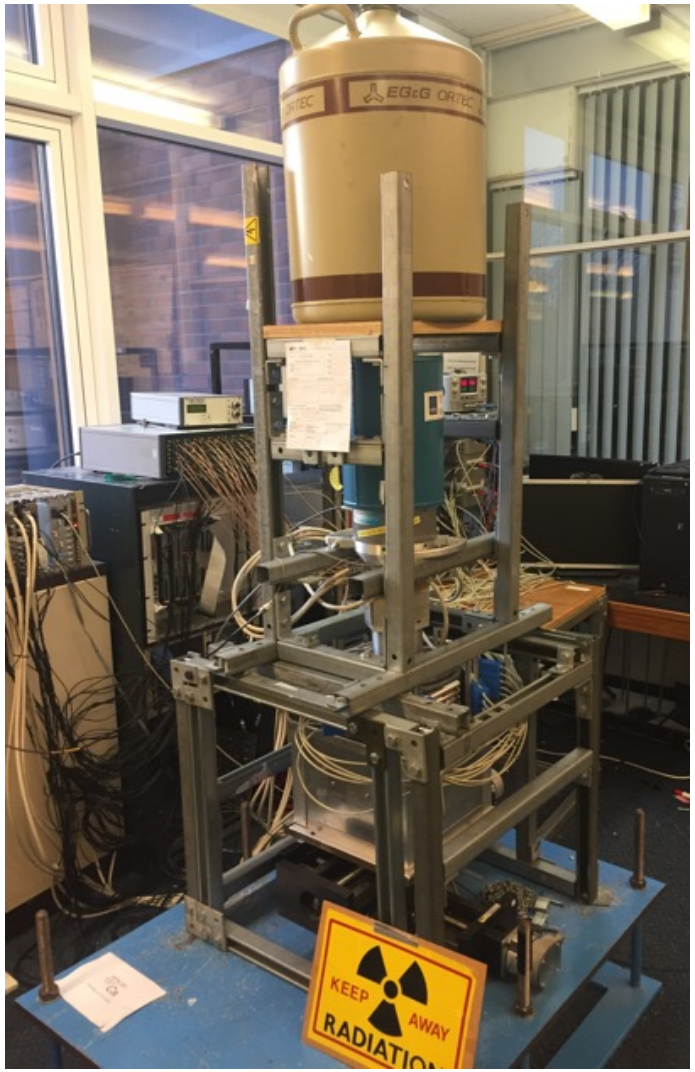
Fraser Holloway - F.Holloway@liverpool.ac.uk

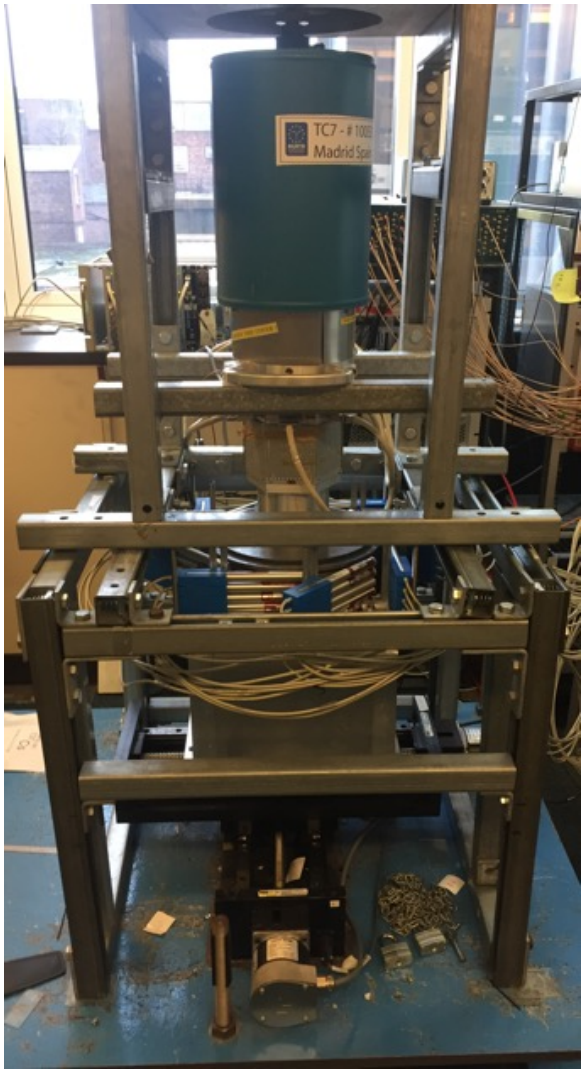
Experimental Validation

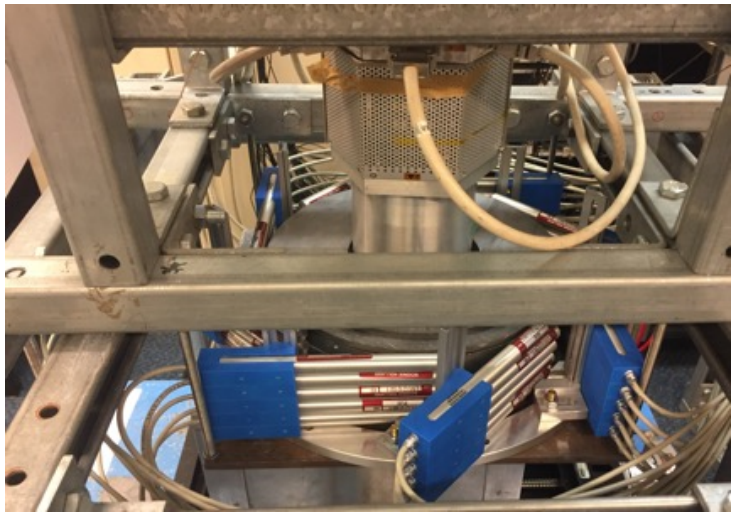
- Coincidence scanning of A005 will be used to validate simulations, ML efforts and PSCS method (IPHC, Strasbourg & Salamanca).
- Will provide a definitive & time-aligned basis for GEANT4.
  - Allows for proper simulations of high-fold events.
- Currently using Caen 1724s, requires GO box & a **lot** of conversion:
  - AGATA → BNC → LIMO → SMA → MCX
- Analysis using MTSort 5.2 (with improved Transpiler)
- 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).
- Most validation measurements have now been taken.
- ^{241}Am surface scan remains to be done.
- After A005 is completed we'll start with A009

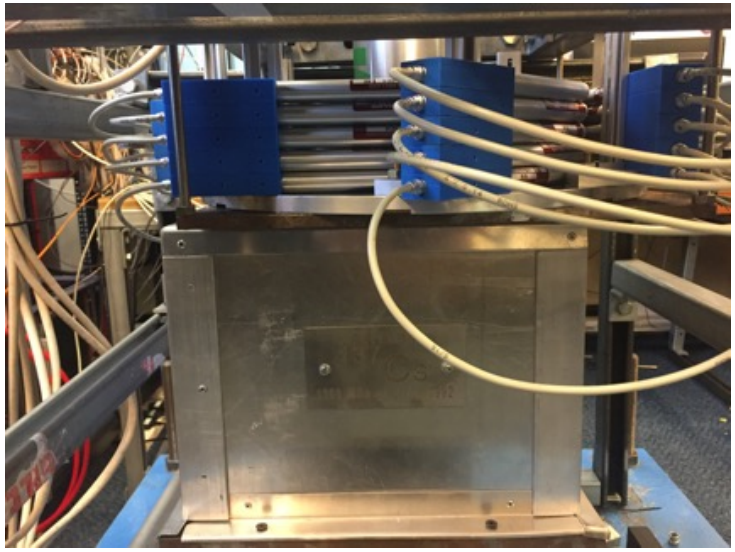


Fraser Holloway - F.Holloway@liverpool.ac.uk

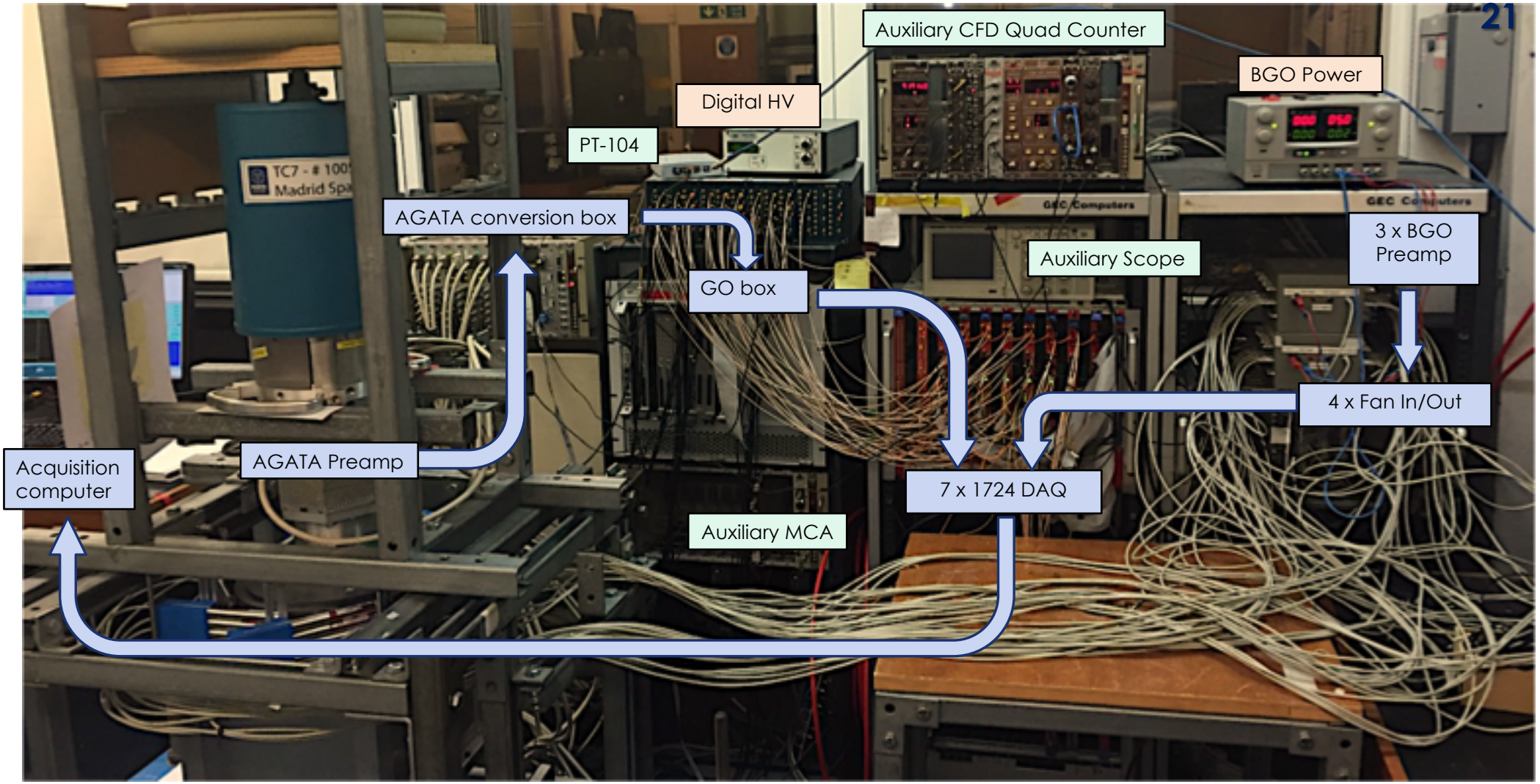






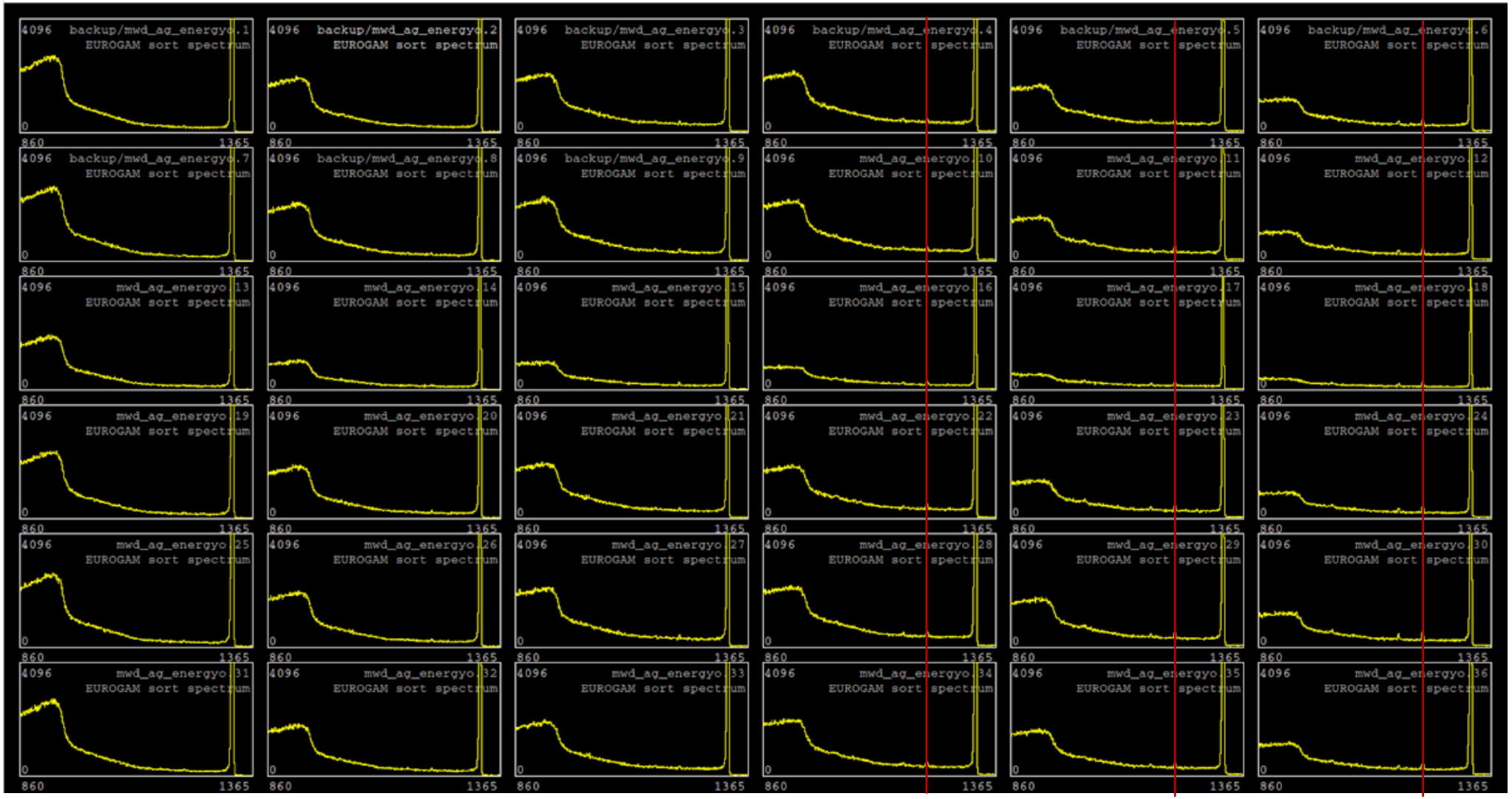


Fraser Holloway - F.Holloway@liverpool.ac.uk



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

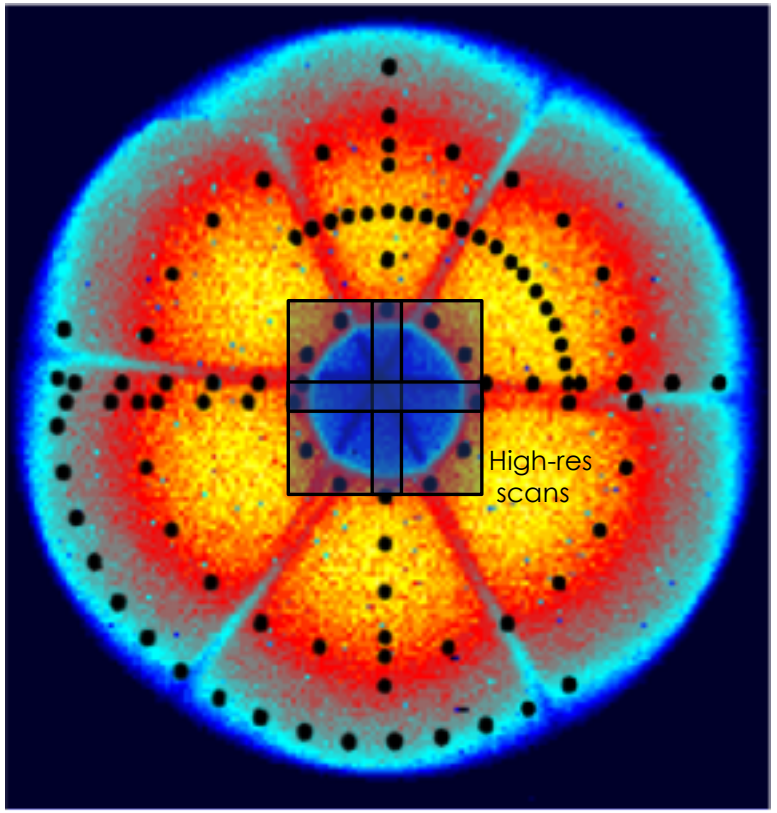
Spectra looks pretty standard, fairly large 511keV peak due to lead collimators, occasional contaminant from external sources



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 Transpiler)
- 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).
- Most validation measurements have now been taken.
- ^{241}Am surface scan remains to be done.
- After A005 is completed we'll start with A009

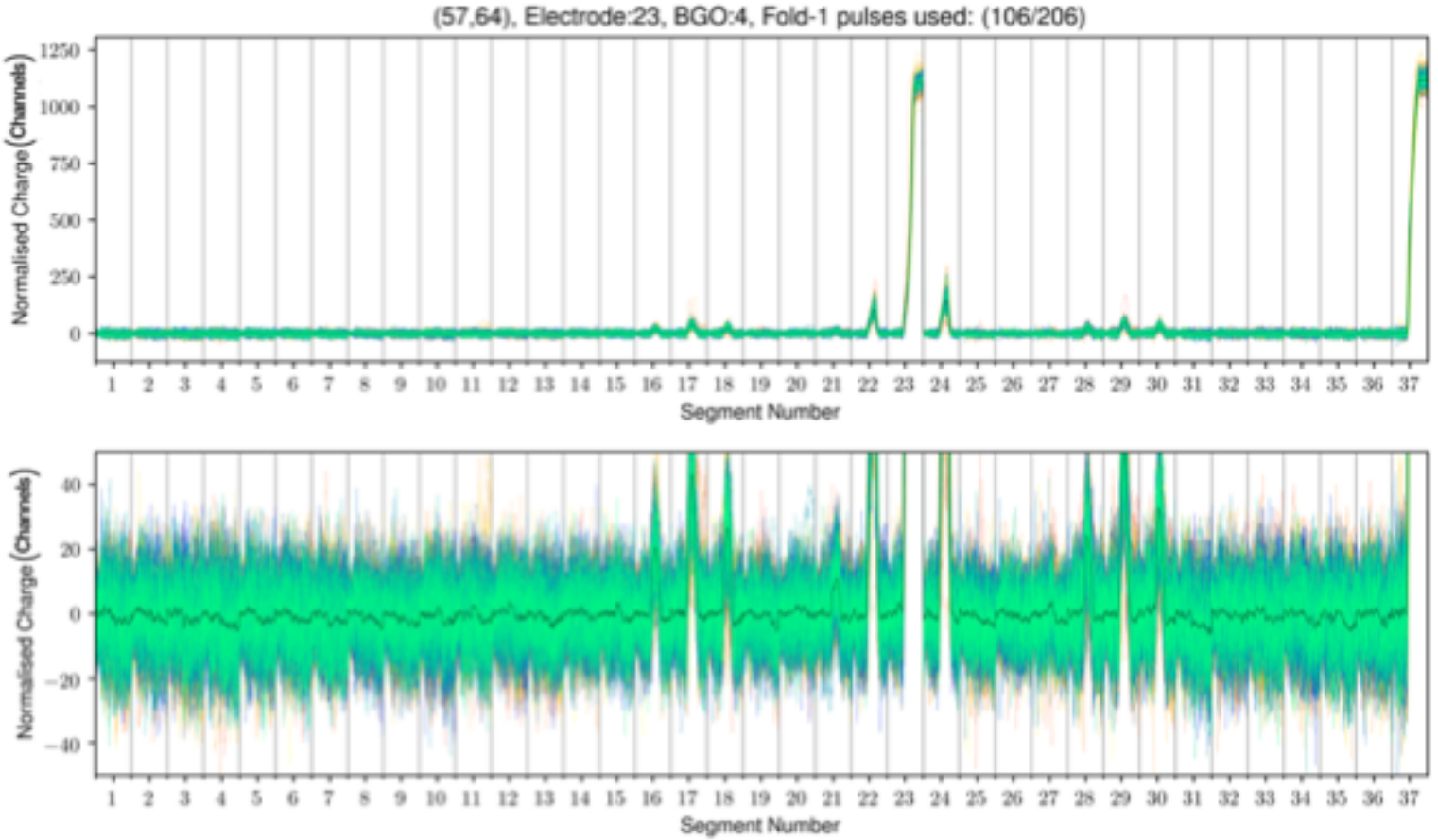
Completed Scan positions



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

Example signal

- Average pulses created via mean exclusion filtering
- Outliers iteratively excluded till ~50% of pulses remain.
- Periodic noise present in mean?
  - ~1.5keV, thermal fluctuation?
- Needs conversion before PSA profiling (10ns sampling)



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

Conclusion

- SIMION fields should really be redone, probably all basis sets too.
- GPUs have advanced significantly over the last decade, likely to continue in the future.
  - Should be revisited considering future projections.
- Tree-based search methods are incredibly efficient but difficult to adapt to high fold.
  - Use fold-invariant search space instead?
  - Very applicable for Fold-1 regardless.
- ML approaches offer good learned relationships but need adaptions to high fold.
  - Realistic high fold dataset necessary.
- Embedded space searching offers a speedup at the cost of accuracy.
- Variational Autoencoders may simplify pulse storage whilst helping with PSA.
- Experimental work is continuing well despite quarantine.

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

Thanks for Listening

Any Questions?

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

This project has received funding from STFC under grant reference ST/P006752/1

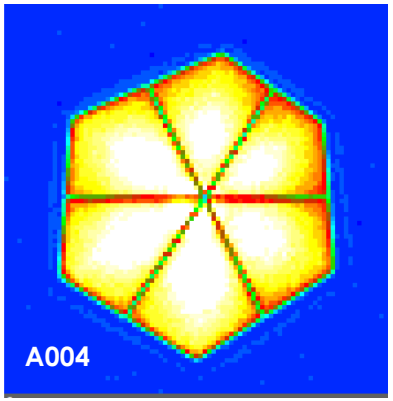
UNIVERSITY OF LIVERPOOL

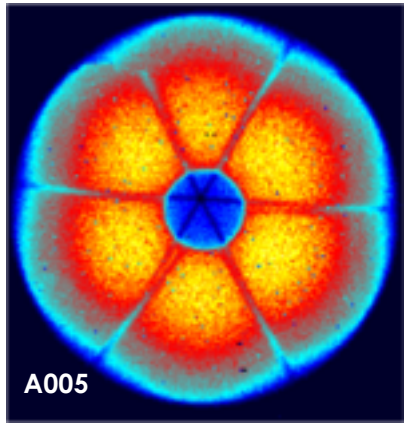
Science & Technology Facilities Council

Signal Discrimination with ML

- Main motivation of this method was to identify interesting sections of the interaction.
  - Possible groundwork for software-based trigger.
  - Because of this these networks need to be fast (and likely simple).
- Position gated pulses used to generate database of hit, transient & noise samples.
- Various networks trained to predict category.
- Ultimately the cut is arbitrary, open to interpretation.
- Doesn't offer much above traditional methods.
  - However if we want to look for something specific it's pretty useful.

Method	Agreement with Midas Label	Execution Time (μs)
Multi-Level Perceptron	~68%	9
Binary Perceptron	~87%	9
Neural Network	~94%	22
Convolution Neural Network	~97.6%	26





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

Determining Multiplicity with CNNs

- Similar setup as before, input data is either core electrode or superpulse.
- Multiplicity to simulate taken from expected distribution.
- Two scenarios simulated:
  - Multiple hits in the same segment.
  - Multiple hits in the same crystal.
- Output of network still treated as categorical
  - Likelihood of fold reported, pick the most likely

Initial results look promising however simulation was heavily idealized.

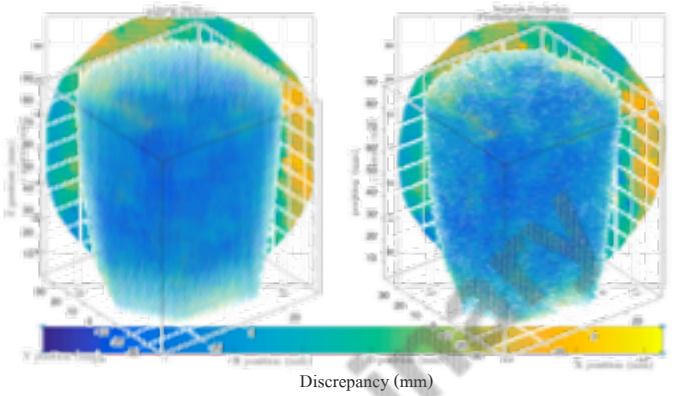
Issues with this method:

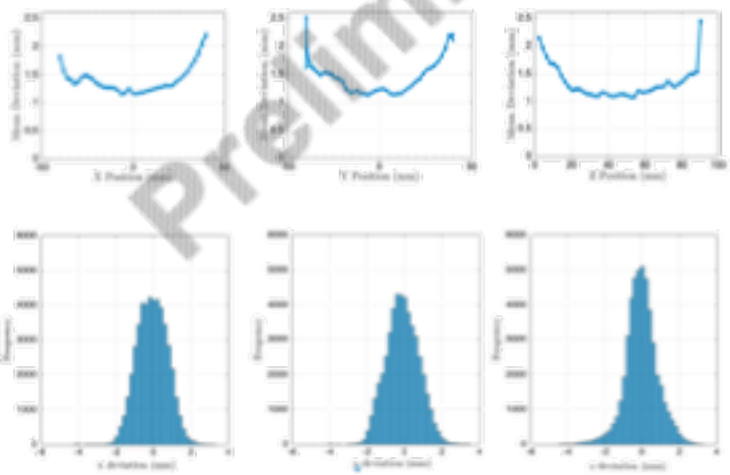
- Interaction locations & energies picked at random, should use GEANT4 instead.
- Realistic noise floor needed, will use experimental data.

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

CNNs for Regression

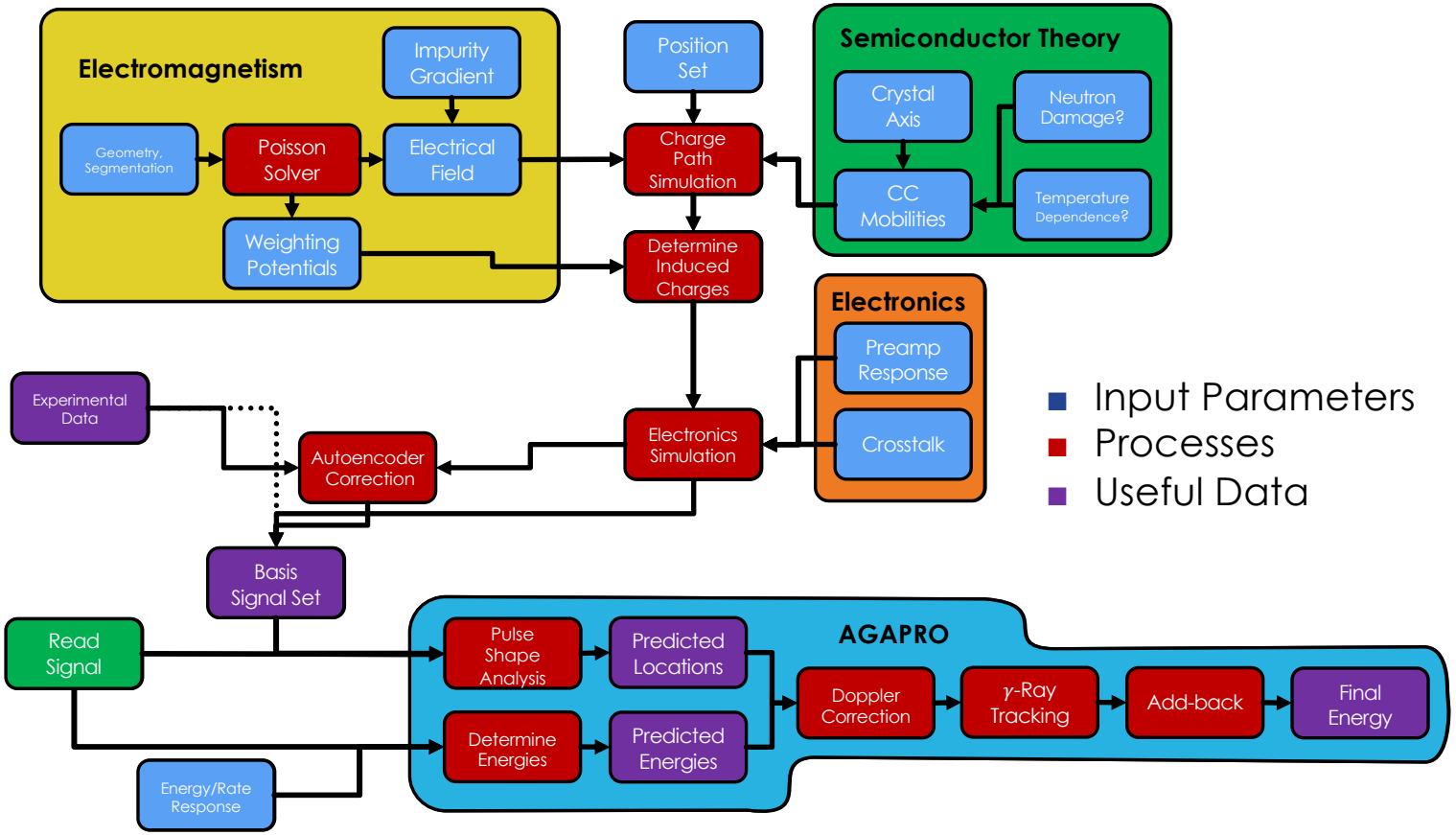
- CNN used to return continuous outputs.
- Trained on 6x8x120 tensor (core contact excluded).
  - Column repeats used for CNN windows.
- ResNet architecture used for robustness.
- Gaussian noise & Dropouts used for reliability.
  - **Should use experimental noise instead.**
- Works well on detectors with high connectivity.
- Currently only implemented for fold-1 events.
  - Training on multi-fold requires separate networks.
  - This isn't difficult, I'm just waiting for an accurate simulation of multiple fold events.
- Reasonable execution time ~300μs.
- Variable FWHM, performs worse at boundaries.
  - Will likely decrease with realistic data.



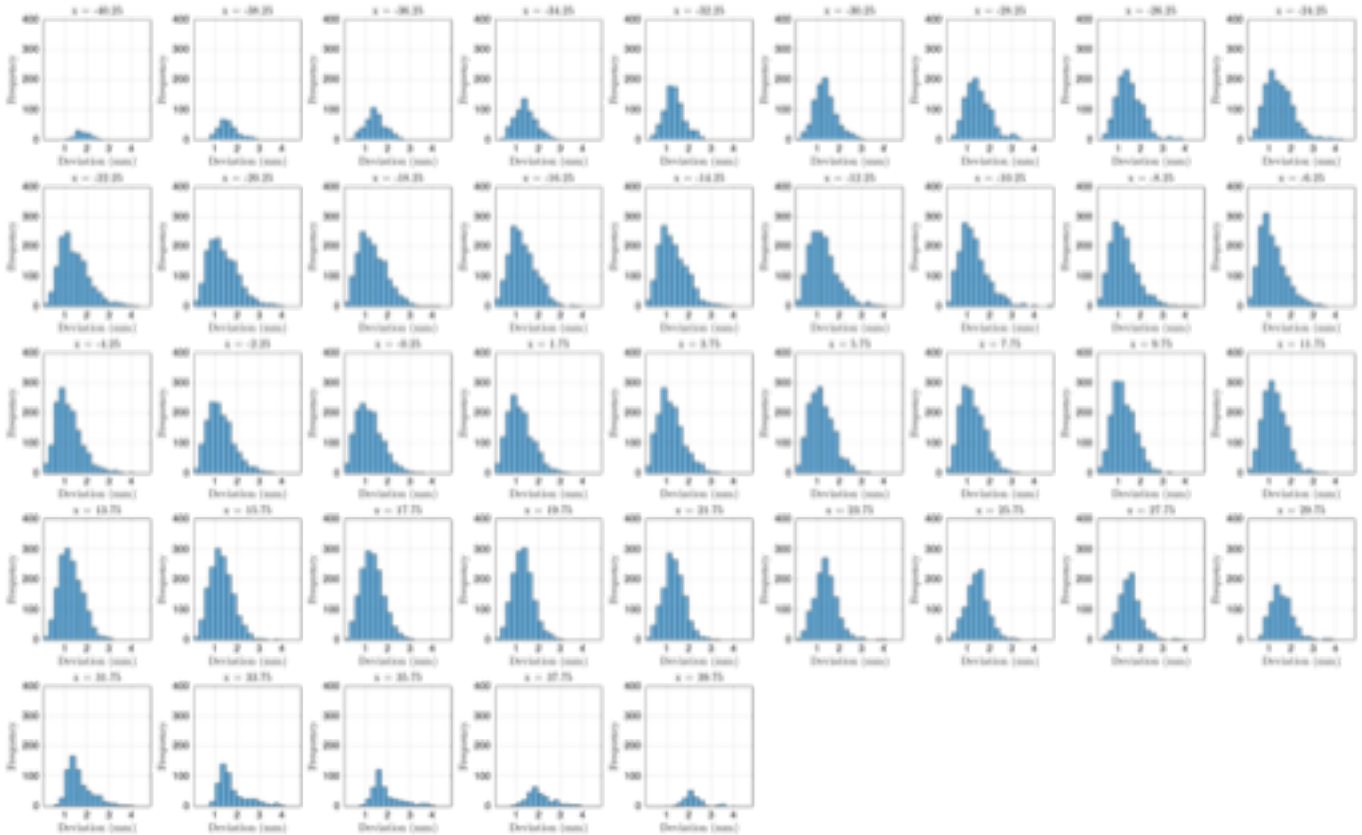


Fraser Holloway - F.Holloway@liverpool.ac.uk

AGATA Pipeline

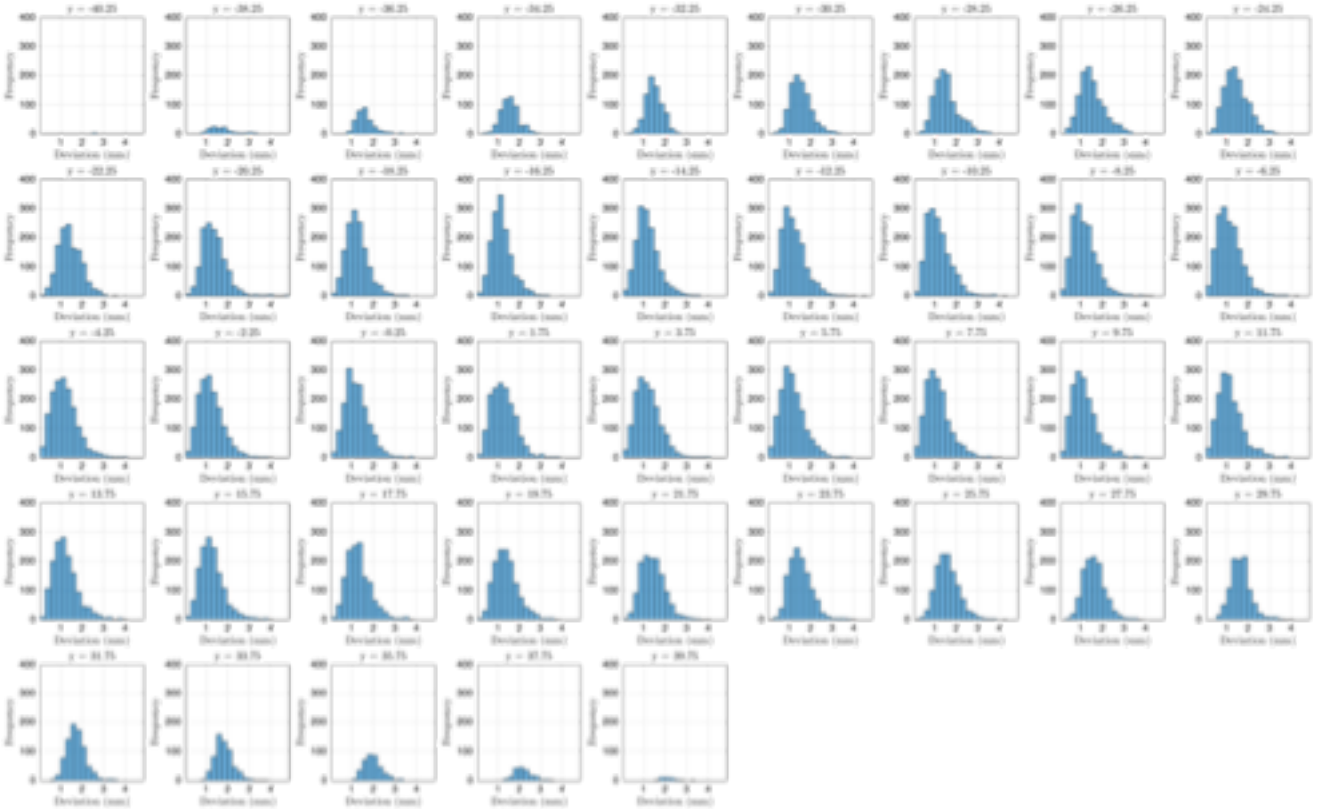


CNN x Deviations



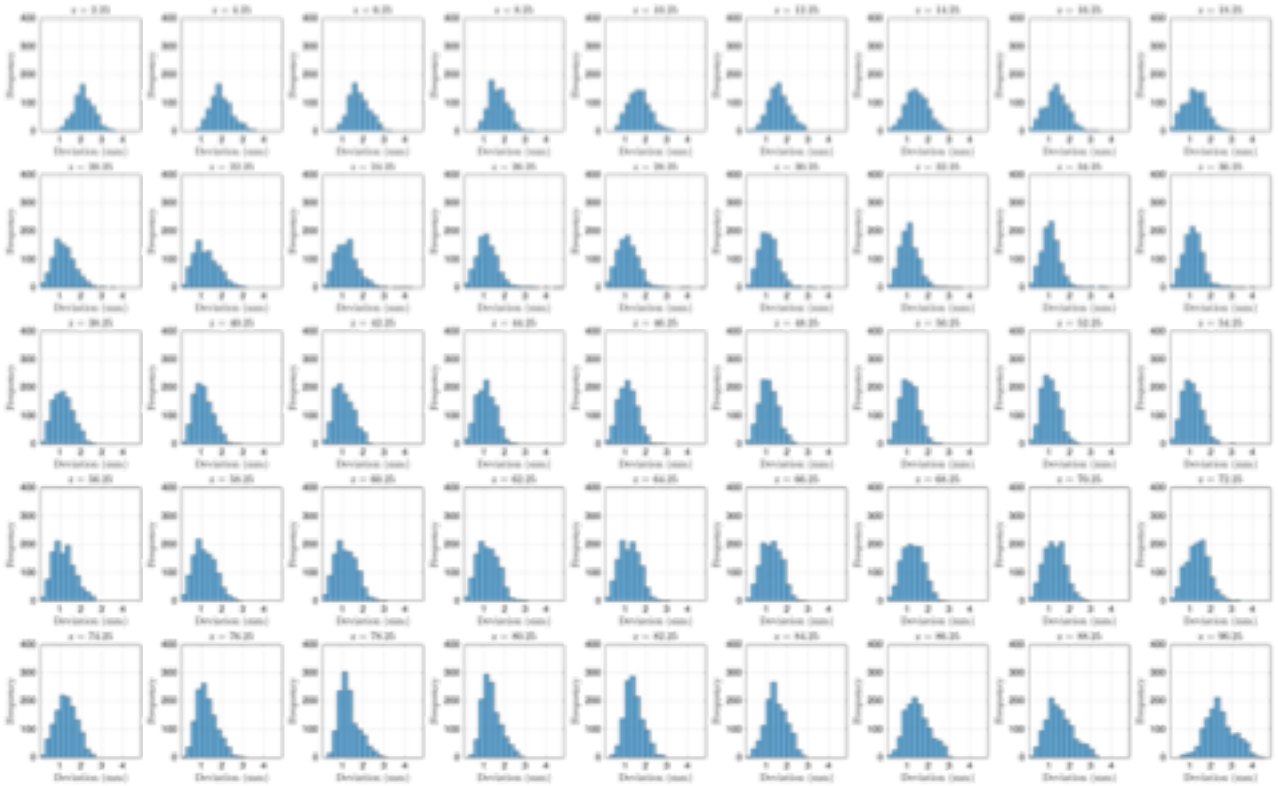
Fraser Holloway - F.Holloway@liverpool.ac.uk

CNN y Deviations



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

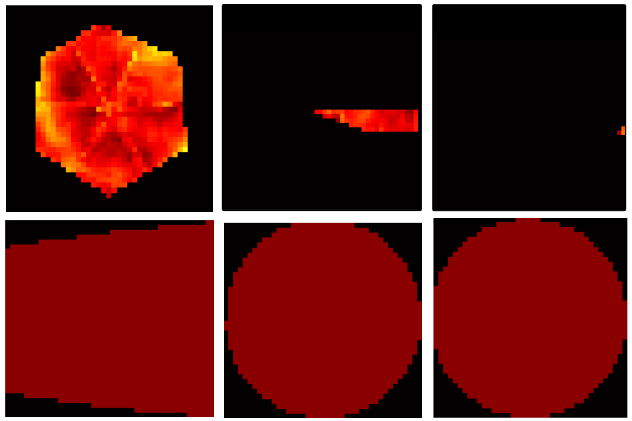
CNN z Deviations

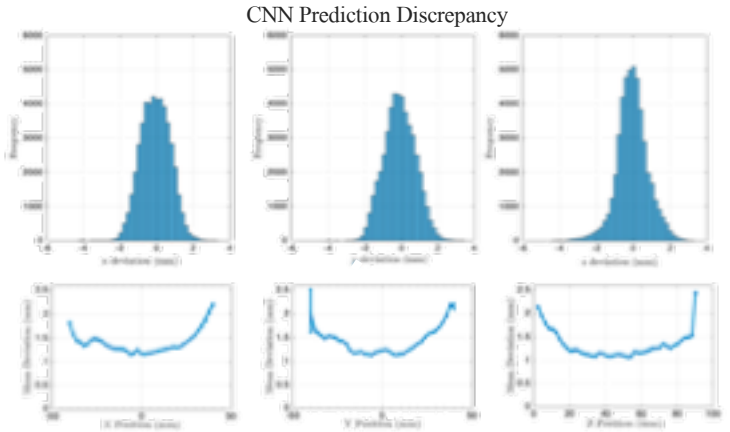


Fraser Holloway - F.Holloway@liverpool.ac.uk

Position Regression with Machine Learning

- Training set taken from ADL simulated pulses, Gaussian noise added
- CNN attempts to predict interaction location from superpulse
- Currently limited to fold-1 events, may be mitigated by using windows



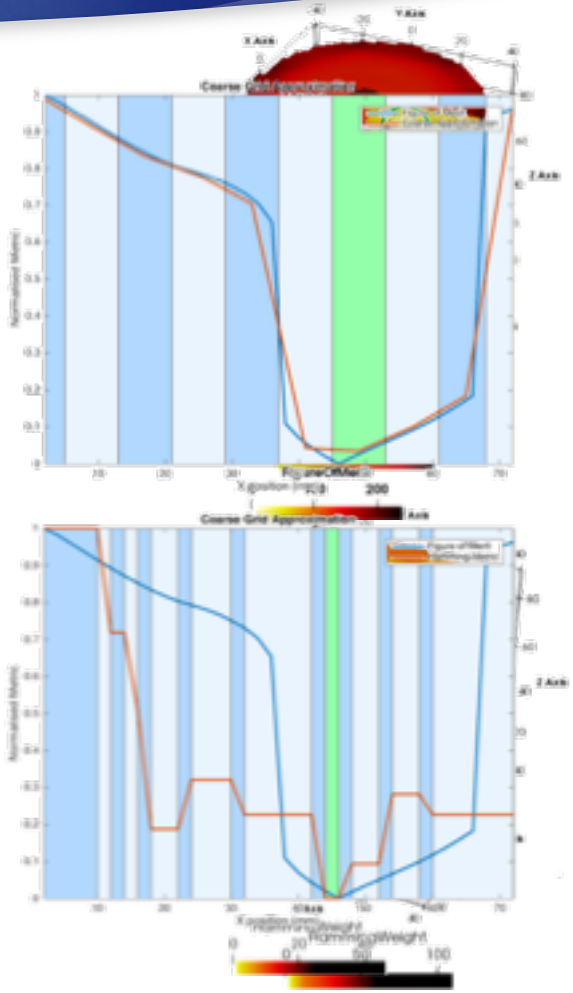


Fraser Holloway - F.Holloway@liverpool.ac.uk

Cluster Optimisation & Tree Building

- Initial investigations were made into optimizing the clustering used in AGS.
- Instead of using Euclidean splitting the basis was split parametrically:
  - Segment # → $T_{10\text{-}90}$ → Charge asymmetry → Transient Signal Fingerprint → FoM
- This allows for hierarchical ordering of basis & bespoke optimizations.
- Resolution of metrics inversely related to execution time.
  - Faster metrics narrow down solution → FoM test applied on final cluster.
- Low resolution metrics mitigate overfitting.
- Sensitivity of the detector is accounted for.

- Ultimately parametric clustering difficult (impossible) at high fold.
  - Accurate fold-invariant metrics difficult to make (*might* be possible with ML).
- Method will likely be revisited in the future.
  - Framework written in C ∴ can be compiled into MTSORT.
- Made somewhat obsolete by LSH.



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