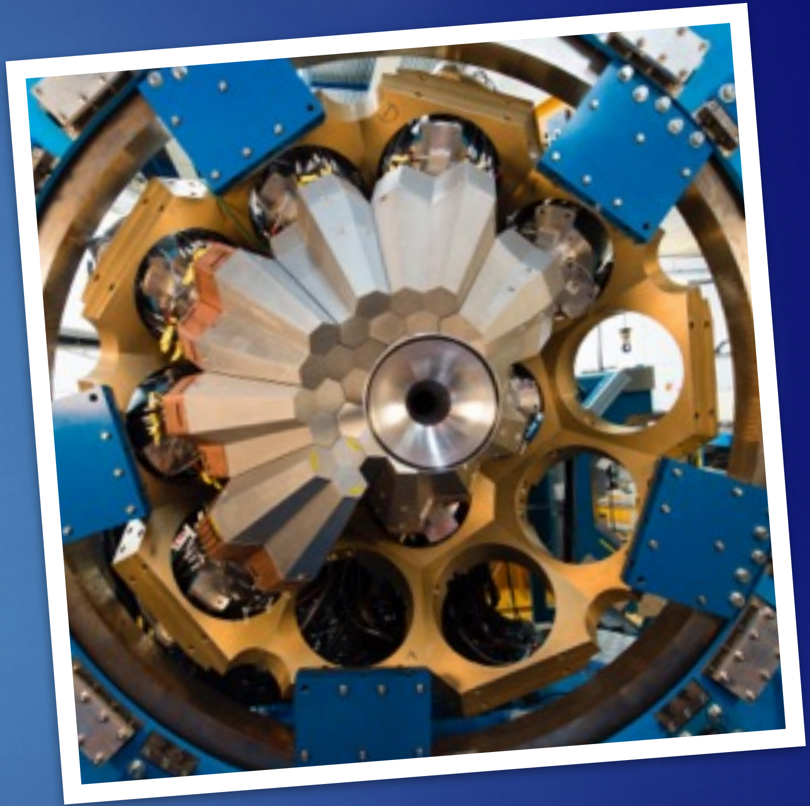
AGATA-GRETINA Week 2019

# Machine Learning and Topological Data Analysis for Pulse Shape Analysis

Fraser Holloway



UNIVERSITY OF LIVERPOOL

LIV.DAT

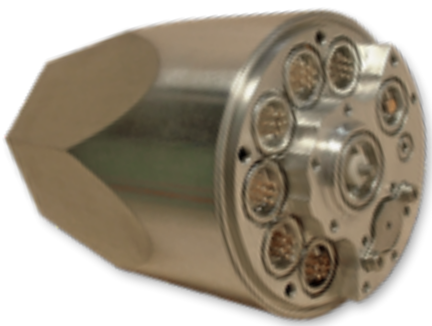
Science & Technology Facilities Council

AGATA ADVANCED GAMMA TRACKING ARRAY

# Pulse Shape Analysis (PSA)

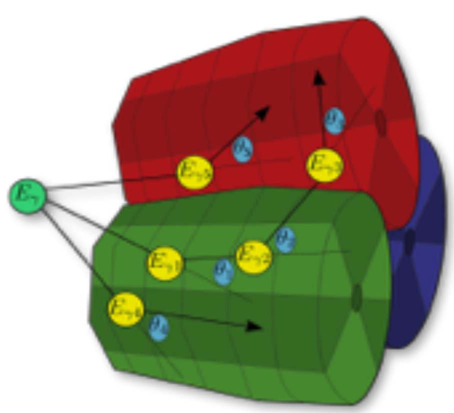


- $\gamma$-ray tracking requires positions at resolution ~5mm FWHM at ~5kHz/CPU.
- Positions must be inferred from electrical response (PSA).
- Complex detector response makes parametric methods insufficient.
- Instead we simulate the detector response in ADL.
- Interaction locations are then determined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



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

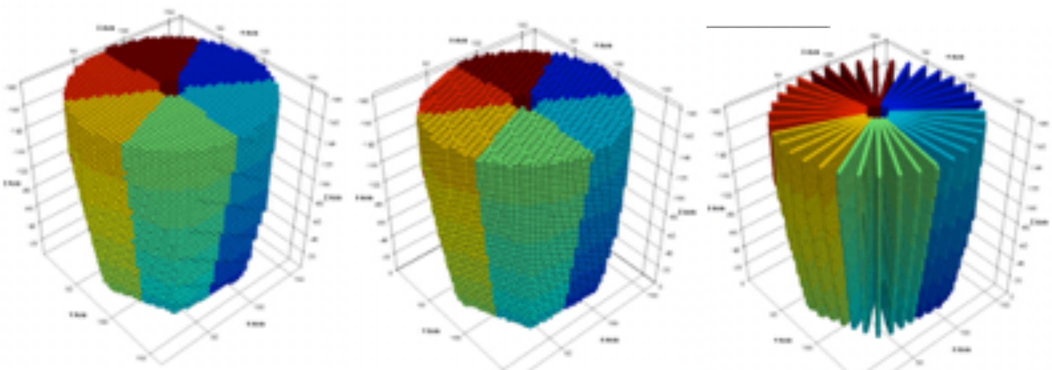
*For signals of segment $j$ at time step $t_i$ with $p$ typically $=2$*

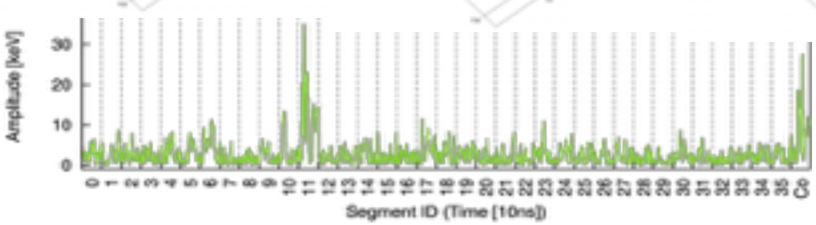


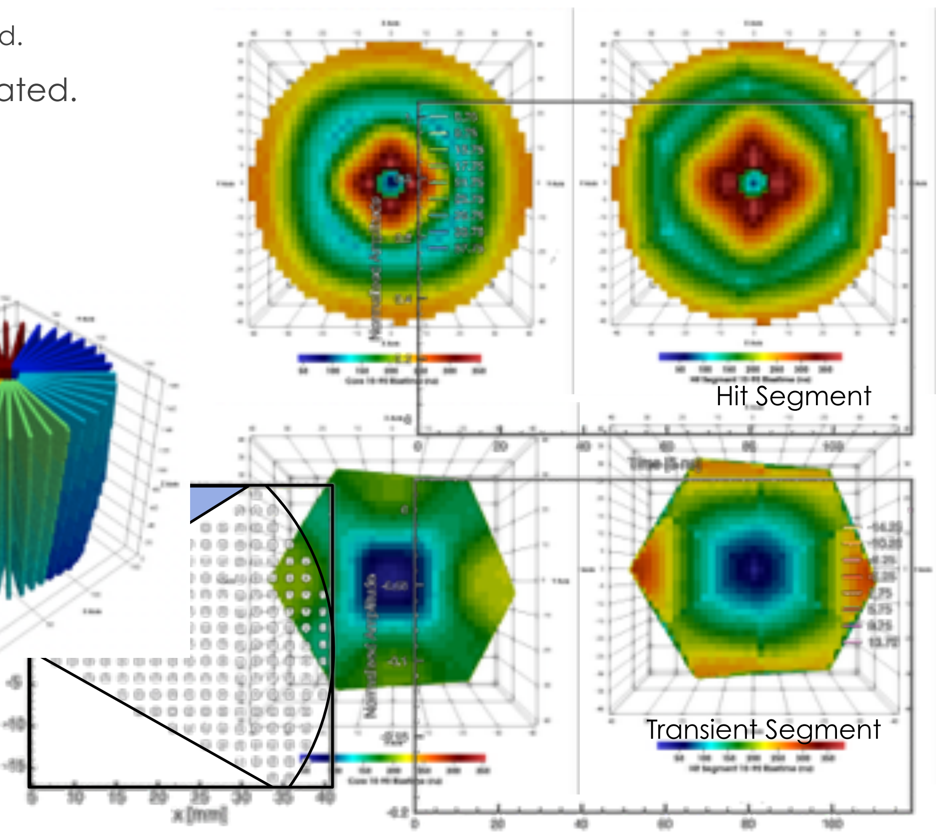
- Other metrics can be used to highlight different sensitivities.
  - Different exponents, weighting for segments.
  - Time shifting via Dynamic Time-Warping.
- My work is on developing Novel PSA techniques for AGATA.

# Detector Simulation

- Simulated data looks reasonable as expected.
- Parametric trends are seen in the data, useful for clustering
  - $T_{10\text{-}90}$, 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.





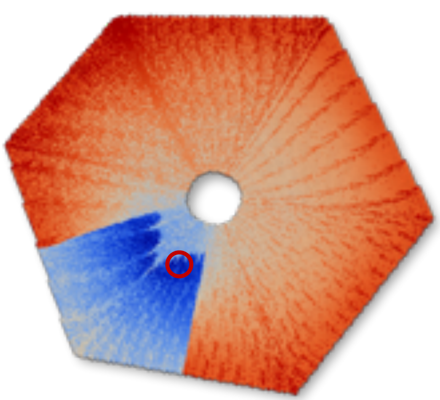

B.Bruyneel – Eur. Phys. J. A (2016)

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

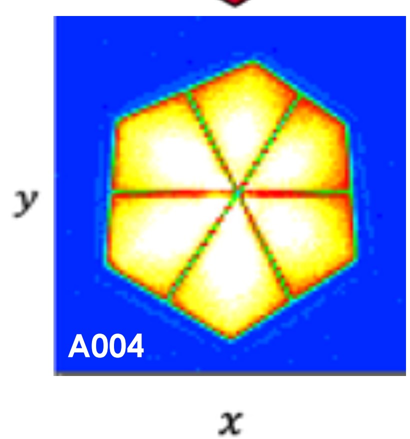
# Simulation Limitations – (Blame SIMION)

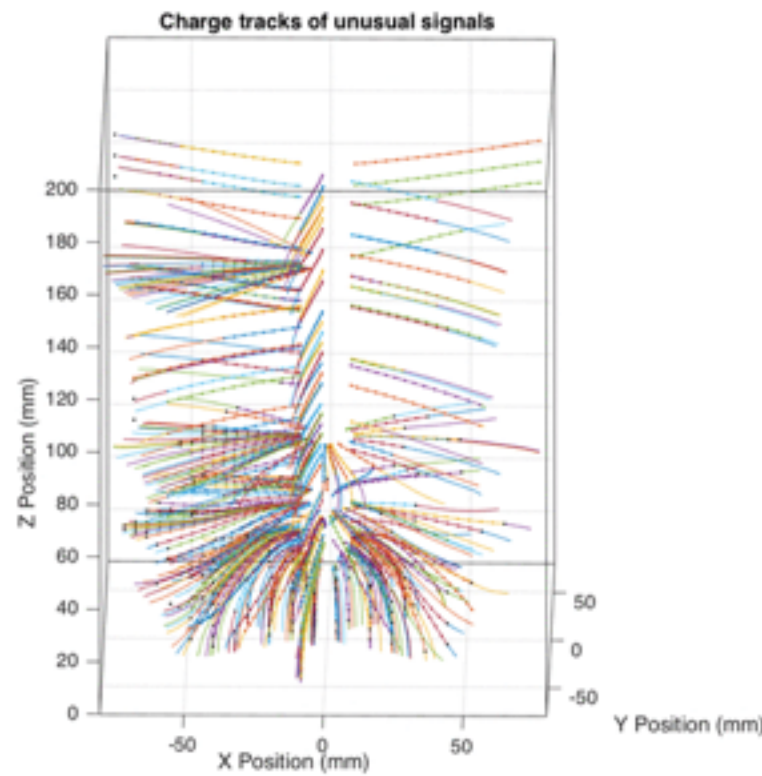


- Field simulation limited to 1mm spacing, ADL is done at 2mm for a reason.
- SIMION segmentation is wrong on face of crystals.
- Odd effects seen at segment boundaries & high resolution:
  - Unexplained ‘charge sharing’ between segments.
  - Sharp discontinuities at edge changes.
  - Overlap of SIMION definitions?



0.5mm FoM Plot showing odd effects
Optimum Circled




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

# Novel Algorithm Development

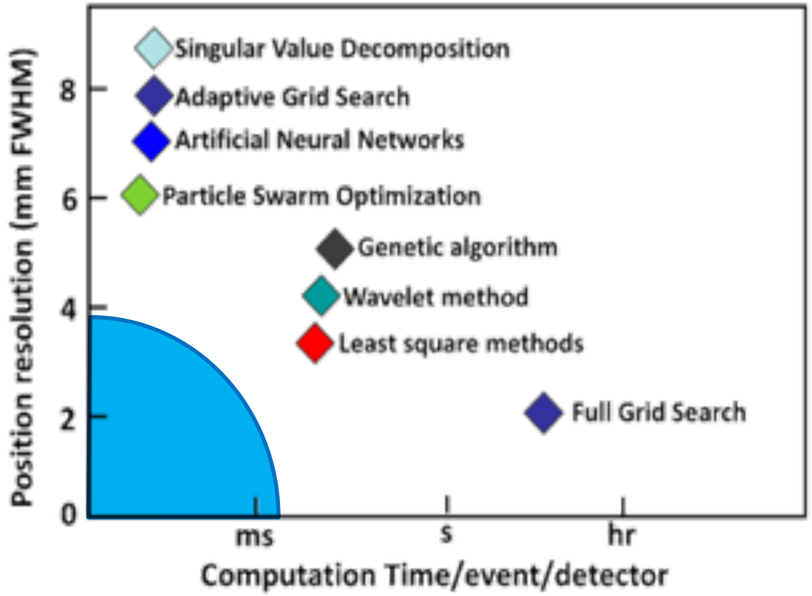


Several PSA algorithms have been tried for AGATA.

Time limits for online PSA mean 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 (**ML**).




Moving beyond the basis simulation this becomes a computer science problem, existing techniques can be applied.

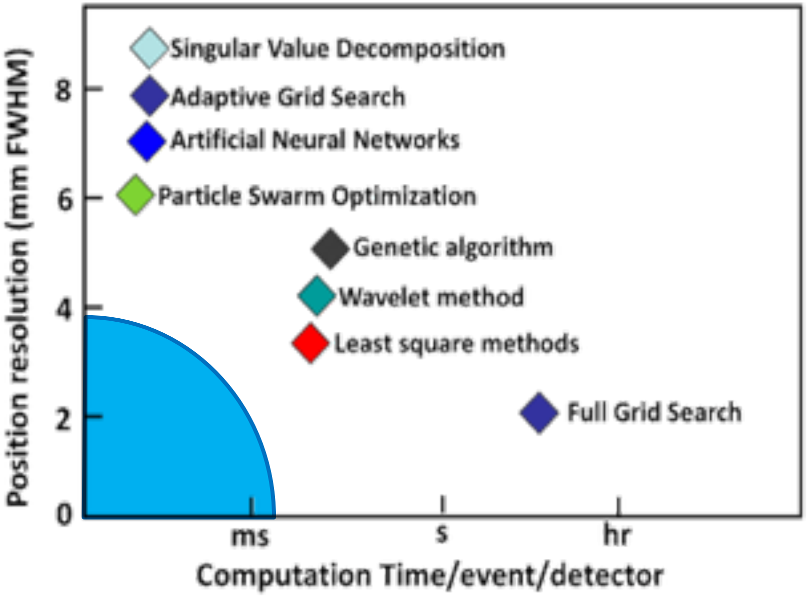
- ∴ Plenty of established fields to learn from.

# 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 $O\log(n)$.




**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.

# Novel Algorithm Development

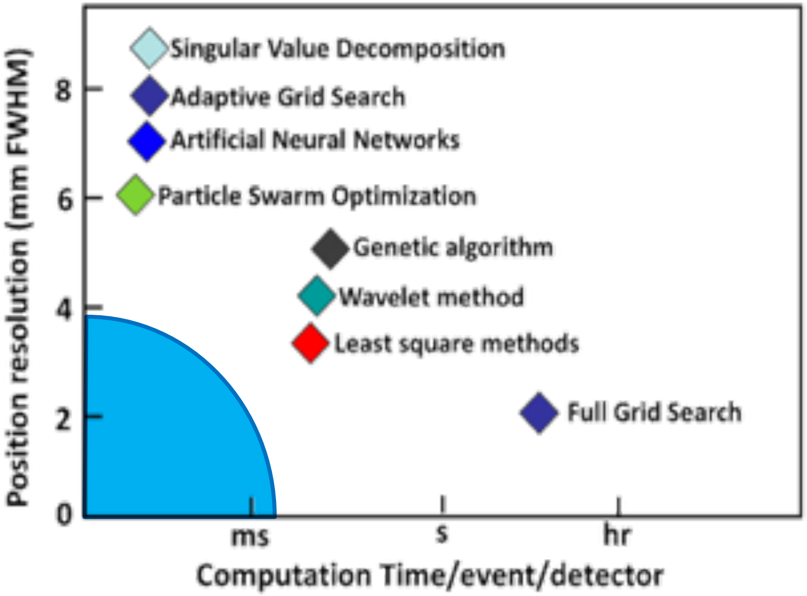


**Tree-based search approaches:**

- *k*NN - *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 Acceleration.

- **All Algorithms have been tested with Gaussian Noise, experimental noise to be determined.**
  - Performance is likely to decrease.
  - Will know more when scanning table is operational.

# 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 use GPU accelerated code:
  - C, C++ (NVCC).
  - Python (with Numba).
- Programs can be compiled to use NVBLAS:
  - MLPACK (Armadillo).

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


Nvidia P5000
(277 GFLOPS)

| Routine | Types | Operation |
|---|---|---|
| GEMM | | Multiplication of 2 matrices. |
| SYRK | | |
| TRSM | | Triangular solve (right angled) |
| TRMM | | Triangular matrix-matrix multiply |
| SYMM | | Symmetric matrix-matrix multiply |

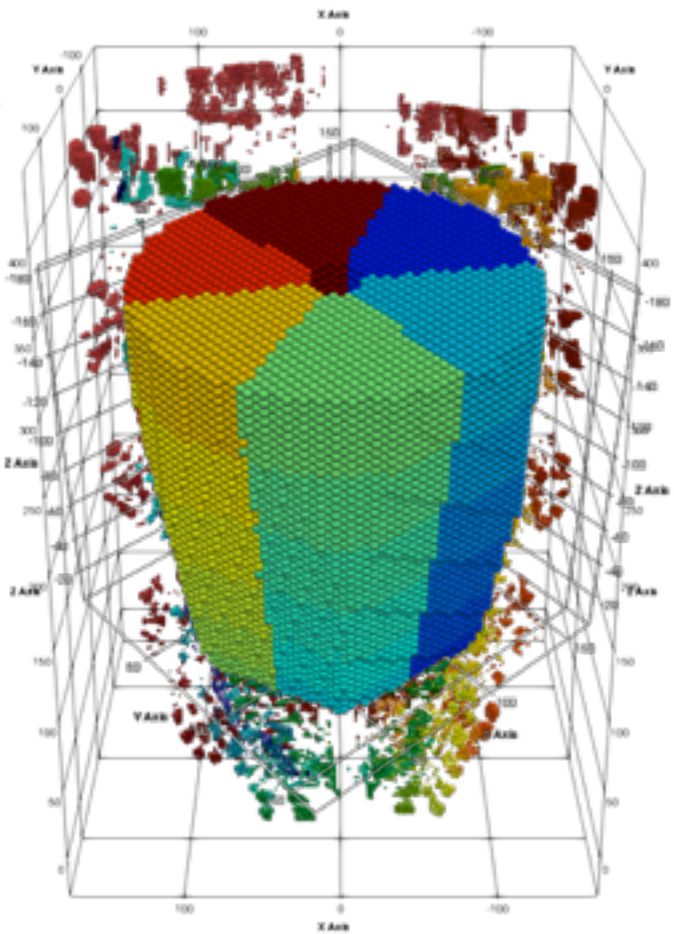
Fraser Holloway - F.Holloway@liverpool.ac.uk

# 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.

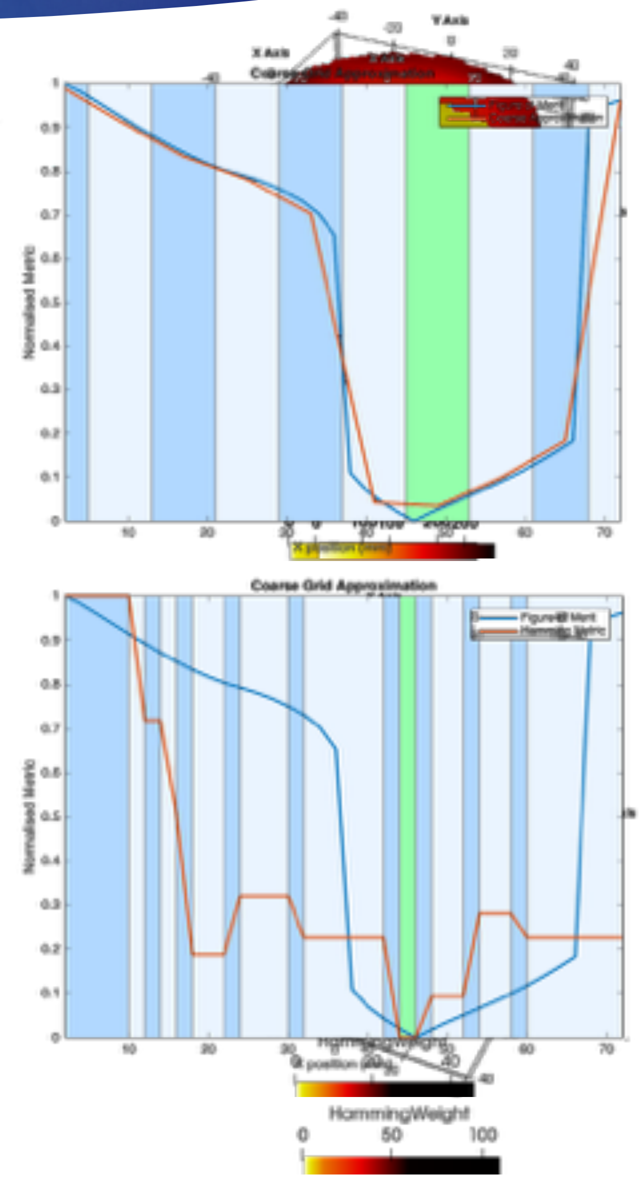


# Cluster Optimisation & Tree Building



- 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.



# 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.

- Currently 3 techniques look applicable to Fold-1 searches:
  - *k*NN
  - LSH
  - **MKS**

# Fast-MKS Searching

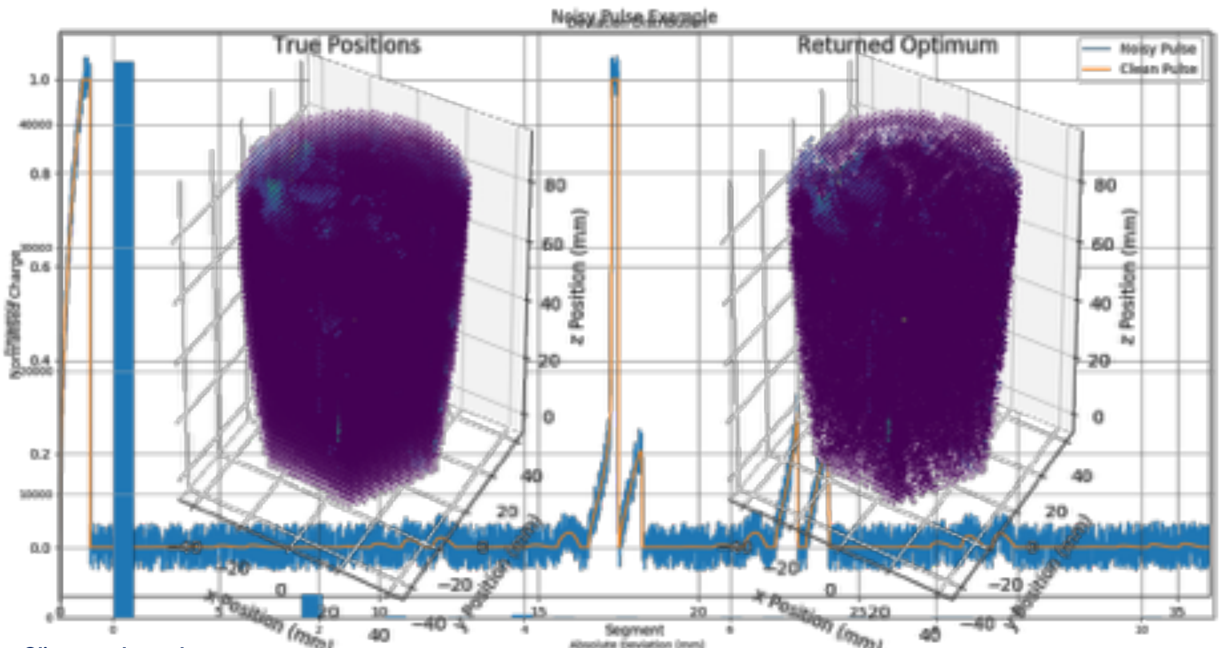
- Fast Maximum Kernel Search uses two trees to search an ordered data structure.
- First tree is used to convert reference set into structured data.
- Second tree is then dynamically built using query set.

- Efficient comparisons mean that the space can be searched quickly.
- Mercer Kernels allow for modifications of phase space, improve separations.
  - More complex kernels have execution penalty.

# Fast-MKS Preliminary Results



- 10% Gaussian noise added to simulated database for preliminary validation.
- MKS with Gaussian kernel used to return top 5 solutions of kernel search with confidences.
- 95% of fold-1 events identified at input location.
- 99% of fold-1 events within 2mm.
- Discrete distances due to finite grid size.
- Currently clustering of deviations are not well understood, needs further analysis.

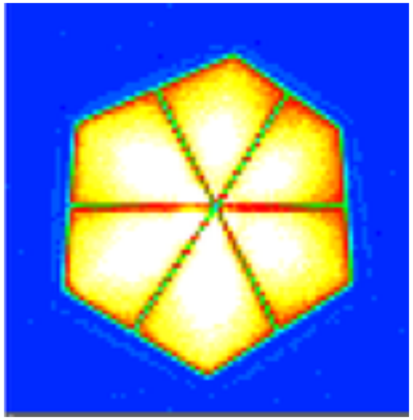


# Signal Discrimination with ML



- 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 | |
|---|---|---|
| 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.

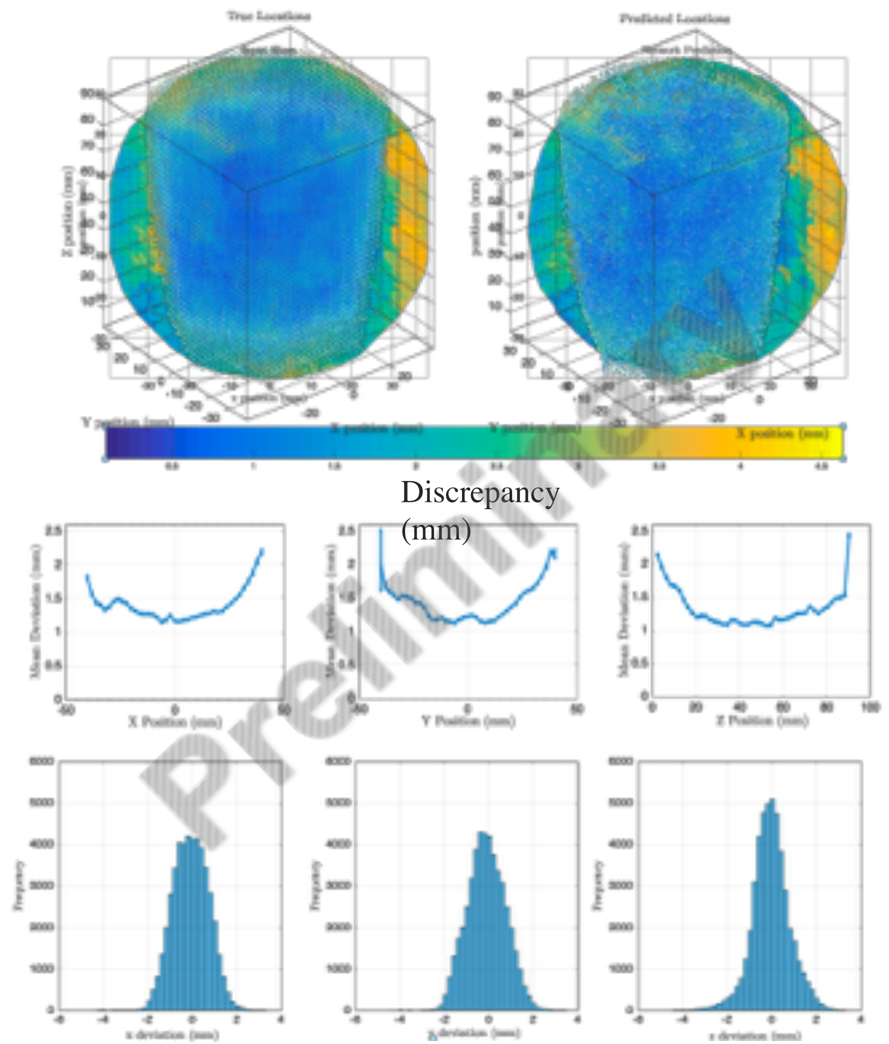
**Issues with this method:**

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

# 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.

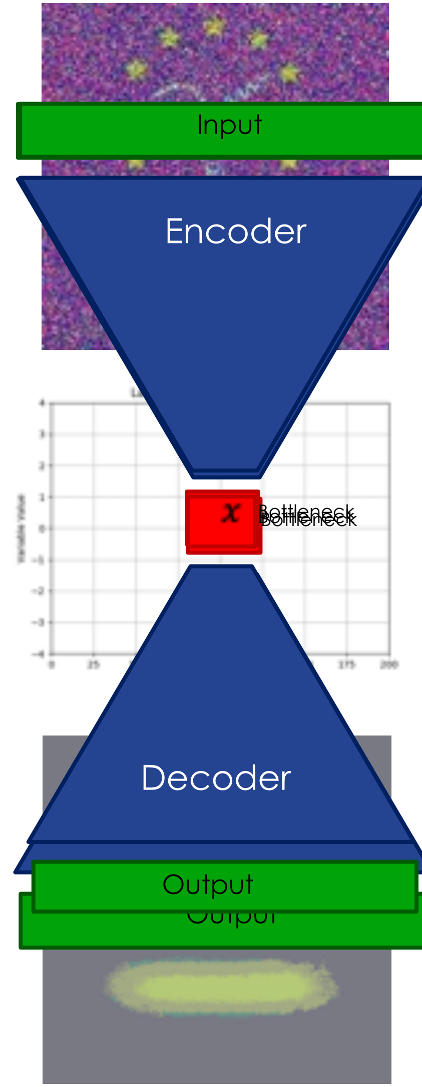



Fraser Holloway - F.Holloway@liverpool.ac.uk

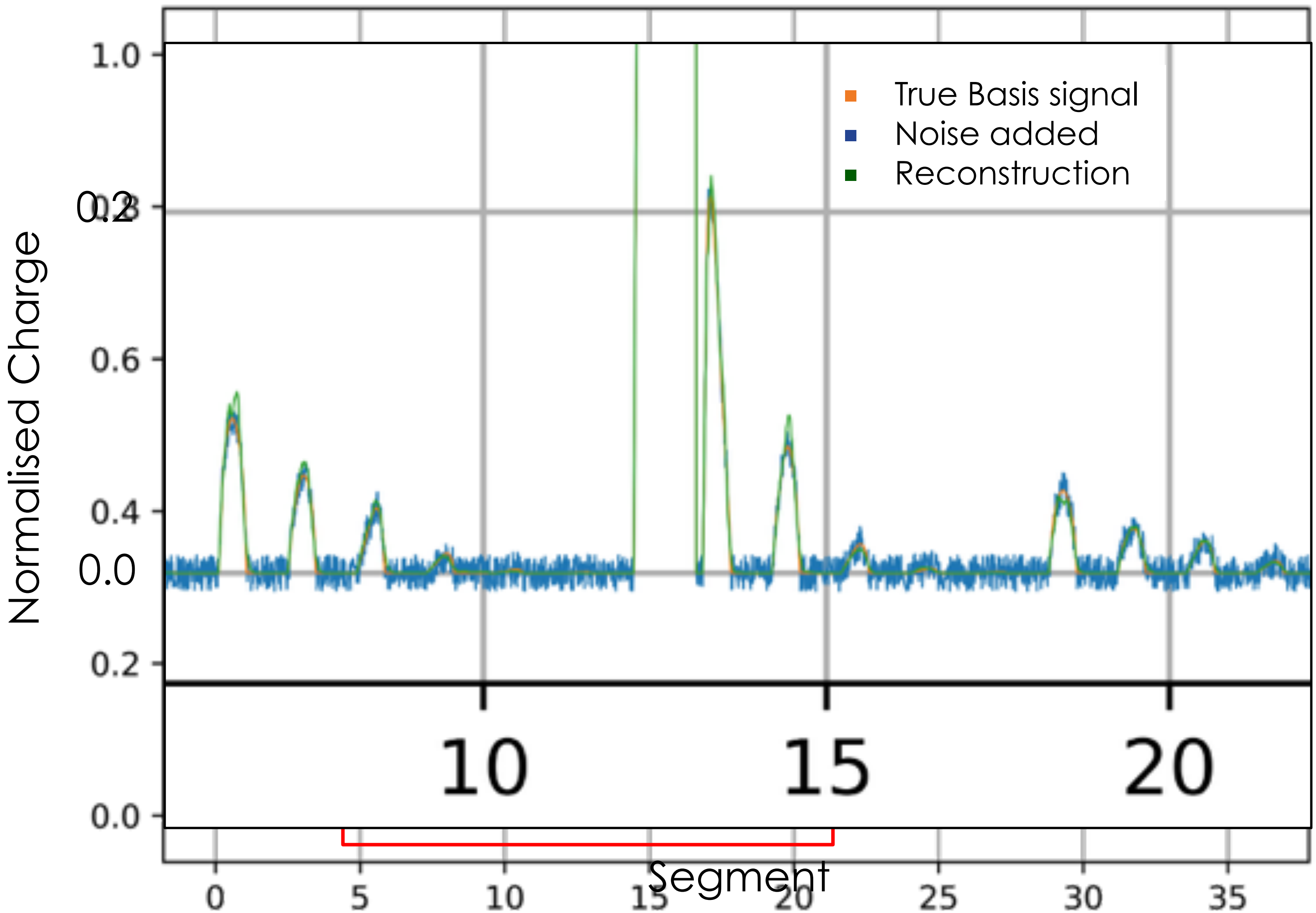
# Autoencoders for Tagging & Compression



- 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.
- As a whole the network replicates a denoised input.
  - Signal is intelligently denoised, small transients are unaffected.
  - Network doesn't see noise as useful information.
- Current Execution time ~56$\mu s$ 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.



# Example Reconstructions, ~44x Compression

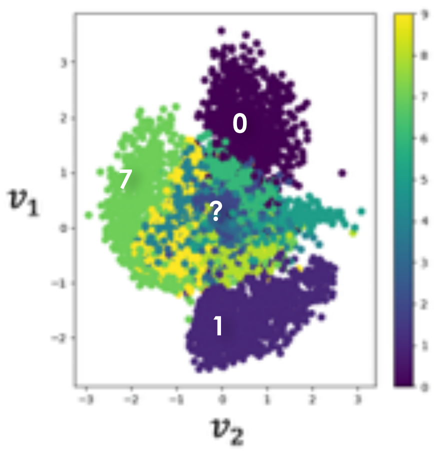


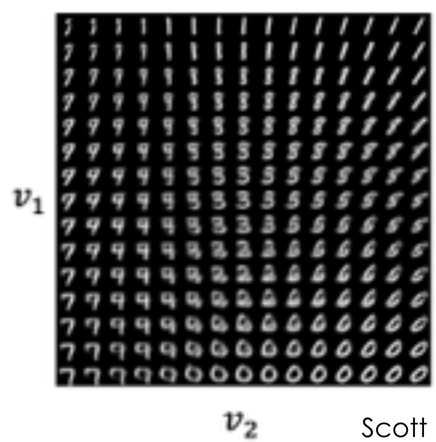

**Impressive reconstruction for 2% size of the original, still room for improvement.**

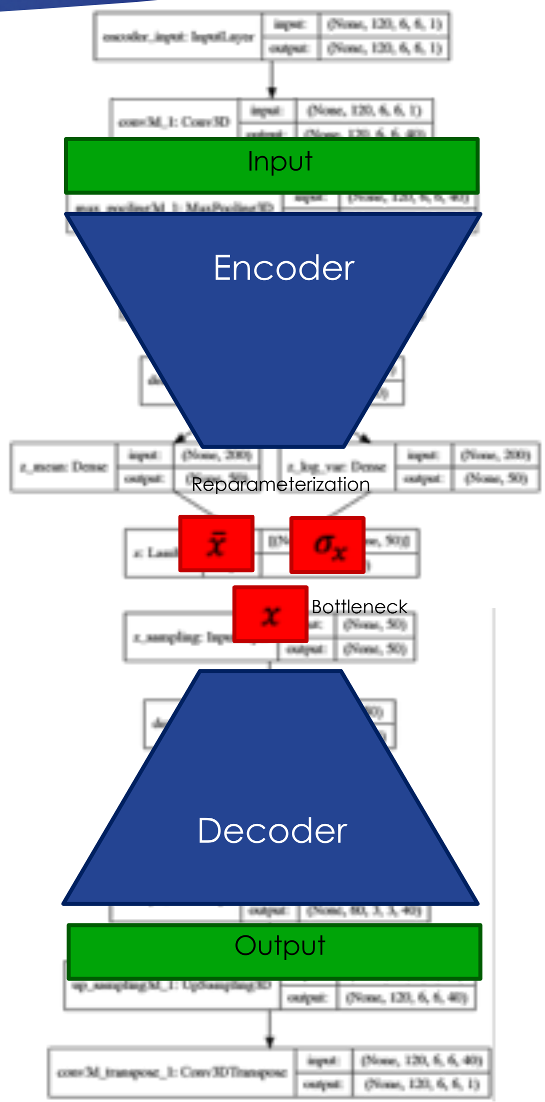
# Disentangled Autoencoders

- Typical AE bottlenecks are impossible to interpret.
- 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:**



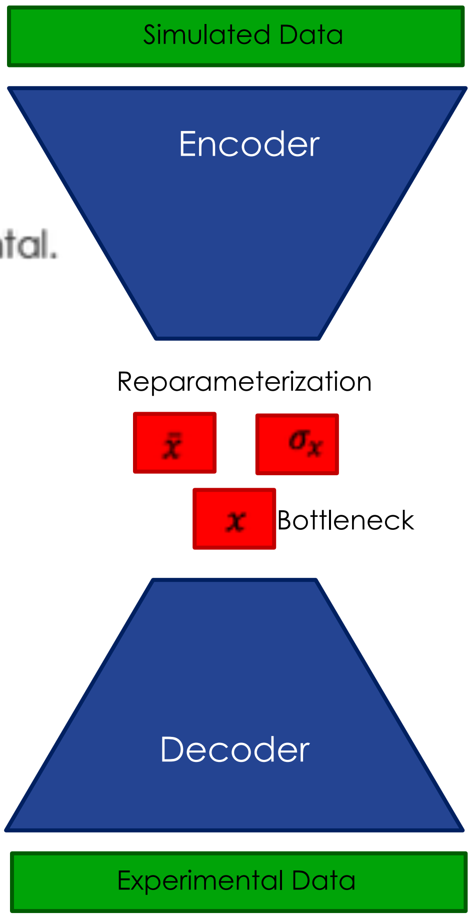

Scott Freitas (CSE 591, 2018)

# 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.

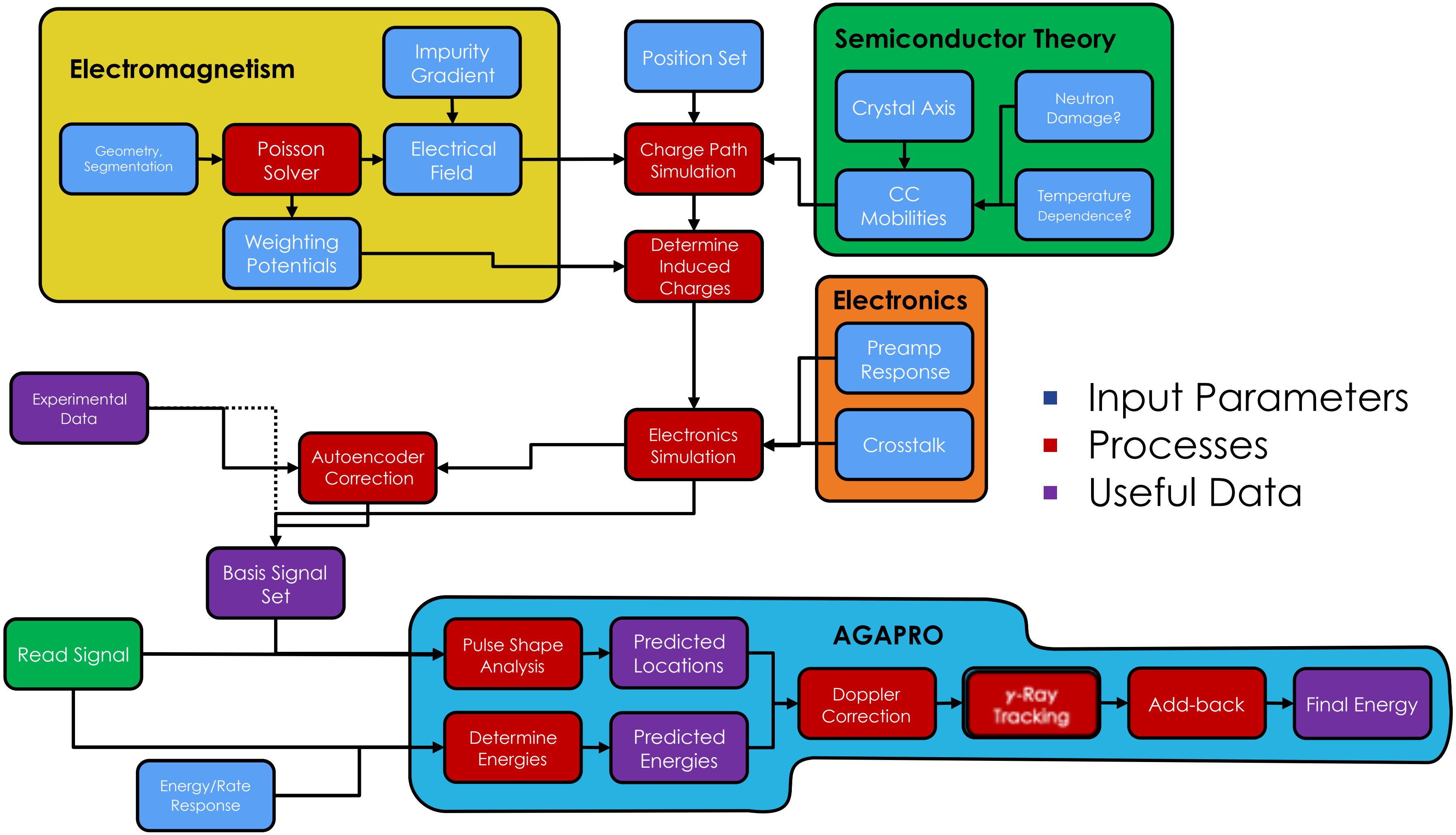
- This approach requires very good experimental data:
  - Full $x, y, z$ characterisation of the crystal.
  - No guarantee that trained model can be adapted to different crystals.

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




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

# AGATA Pipeline

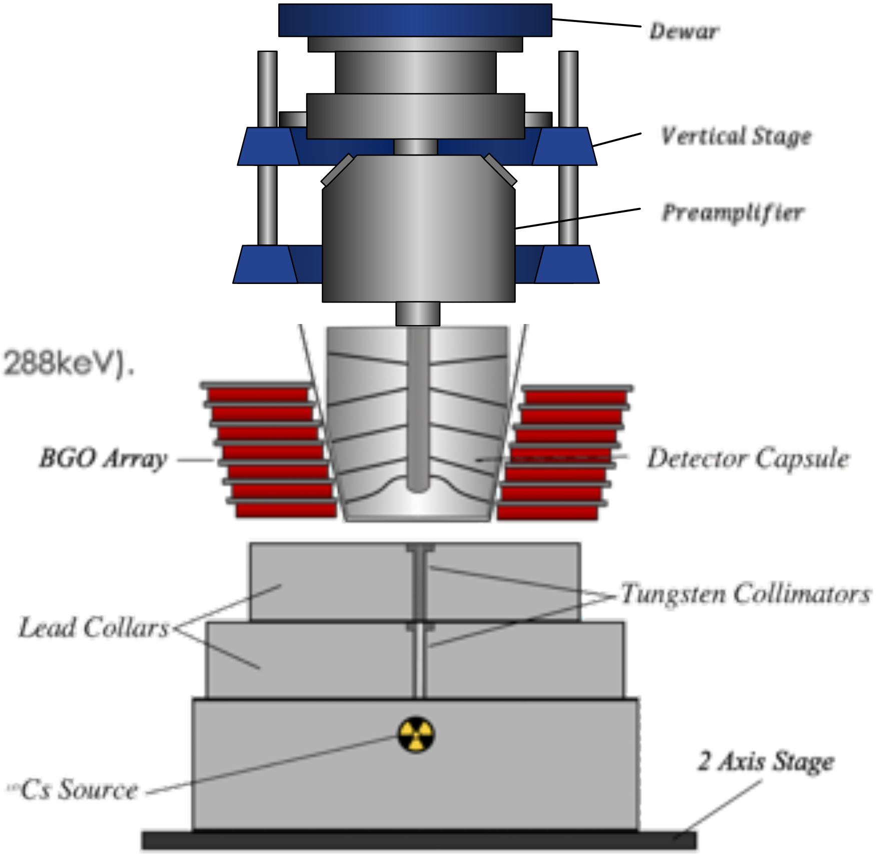



- Input Parameters
- Processes
- Useful Data

# Experimental Validation



- Coincidence scanning will be used to validate simulations, ML efforts and PSCS method (IPHC, Strasbourg).
- Will provide a definitive & time aligned basis for Geant4.
  - Allows for proper simulations of high-fold events.
  - Currently using Caen 1724s, may switch to AGATA digitizers.
- 1GBq $^{137}$Cs source collimated to 1mm on $x, y$ stage.
- Vertical stage added to apparatus for quick $z$ movements.
- 90º scatter gating using BGO array & energy gating (374 & 288keV).
- I'm currently writing the MTSort code for acquisition.
- Typical validation measurements will be taken:
  - $^{241}$Am surface scan for alignment.
  - Gated cross & circle measurements for CAO.
  - Gated coarse cubic grid using vertical stage.
  - High-resolution pencil beam of front segmentation.
  - (Time permitting) Automated High-resolution scan.



Coincidence scanning apparatus found at Liverpool

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

# Conclusion



- GPUs have advanced significantly over the last decade, likely to continue in the future.
  - Definitely 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.
- We have a good standing for more ambitious ML techniques.
  - Discrimination
  - Regression
  - Auto-tagging / Fingerprinting
  - Compression
  - Basis Correction
- Variational Autoencoders may simplify pulse storage whilst helping with PSA.
- I can't take all these methods to completion, future work will involve whittling down algorithms.

# Thanks for Listening

Any Questions?

AGATA
ADVANCED GAMMA TRACKING ARRAY

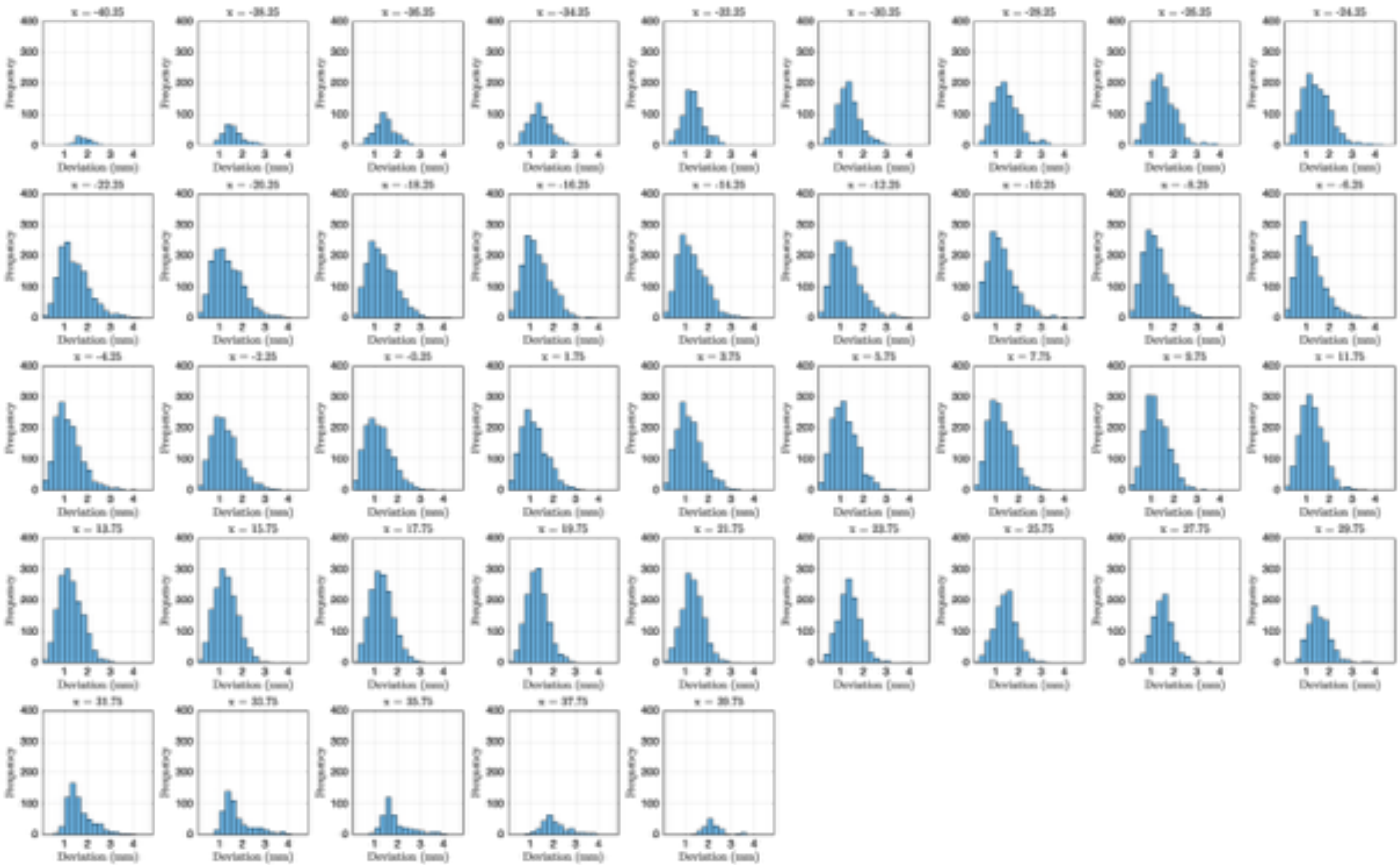
UNIVERSITY OF LIVERPOOL

Science & Technology Facilities Council

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

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

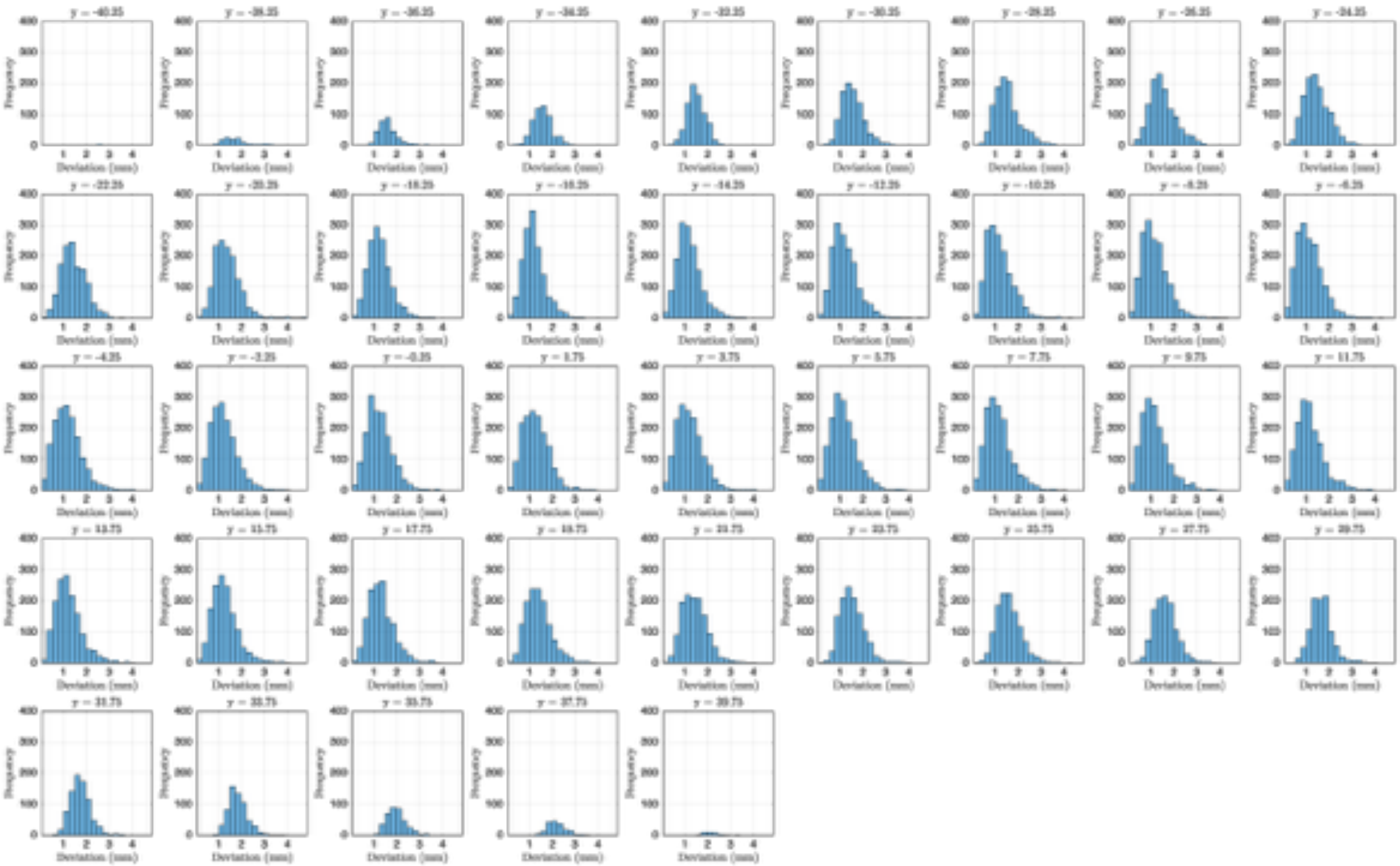
# CNN $x$ Deviations



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

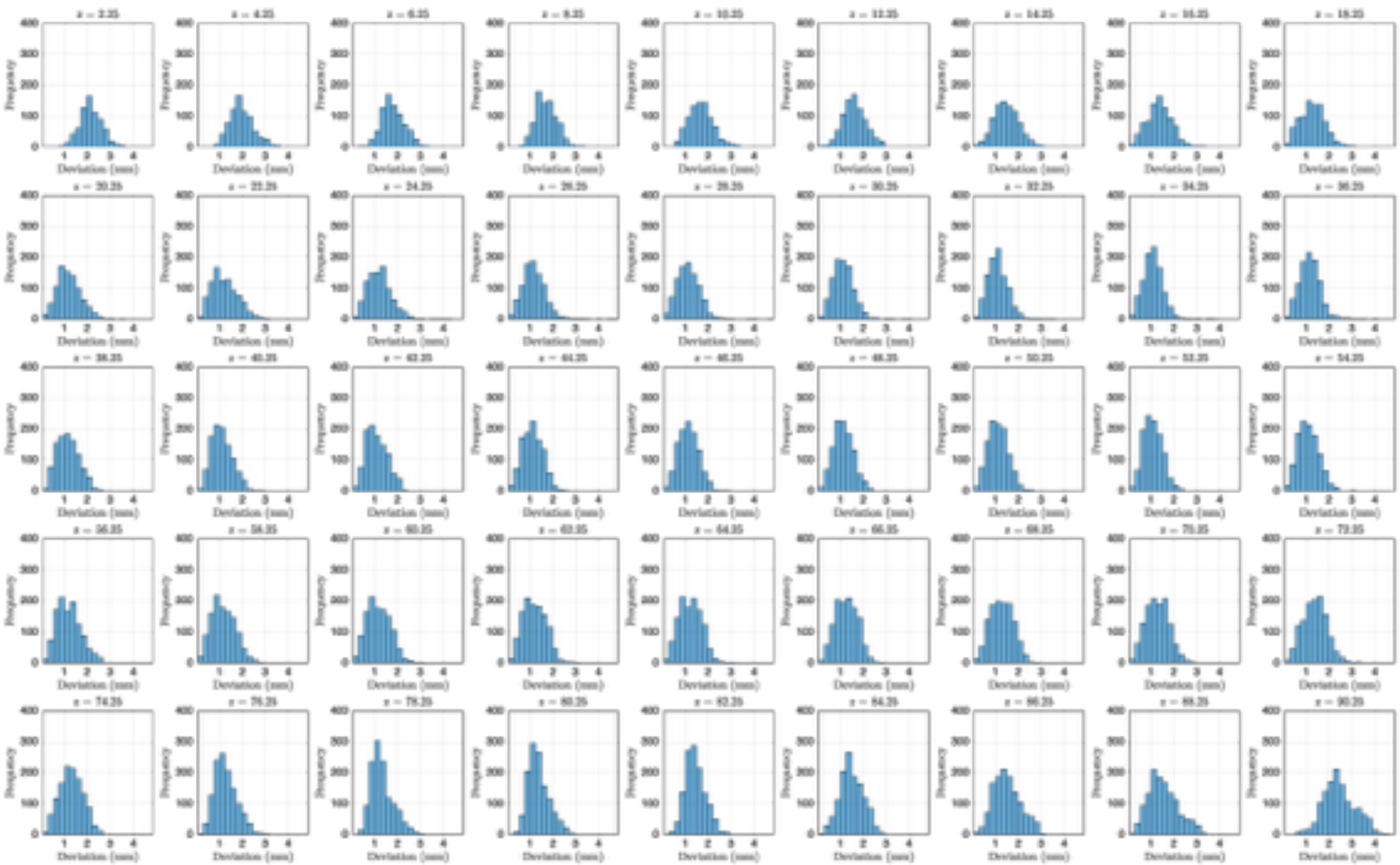
# CNN $y$ Deviations





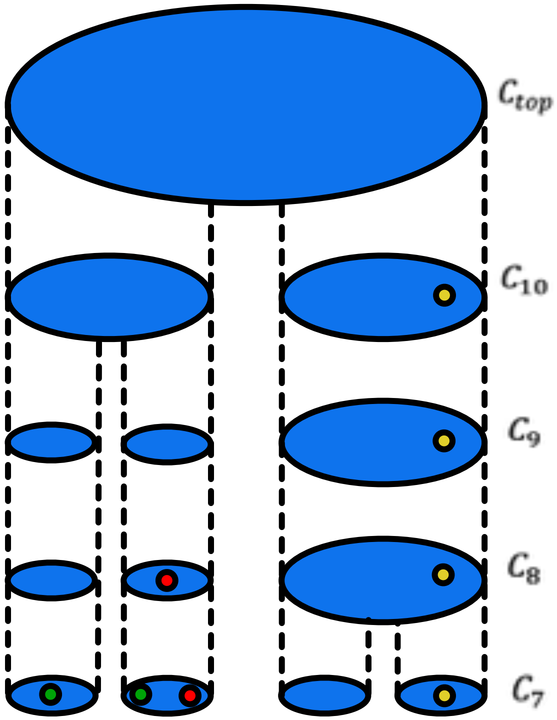
Fraser Holloway - F.Holloway@liverpool.ac.uk

# CNN $z$ Deviations



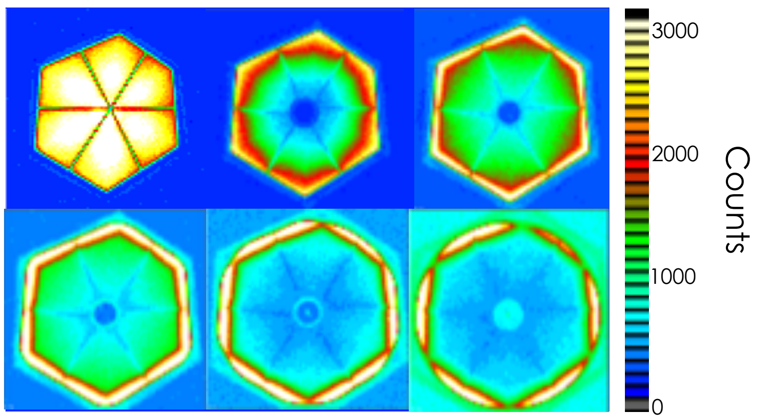
# Cover Tree Rules

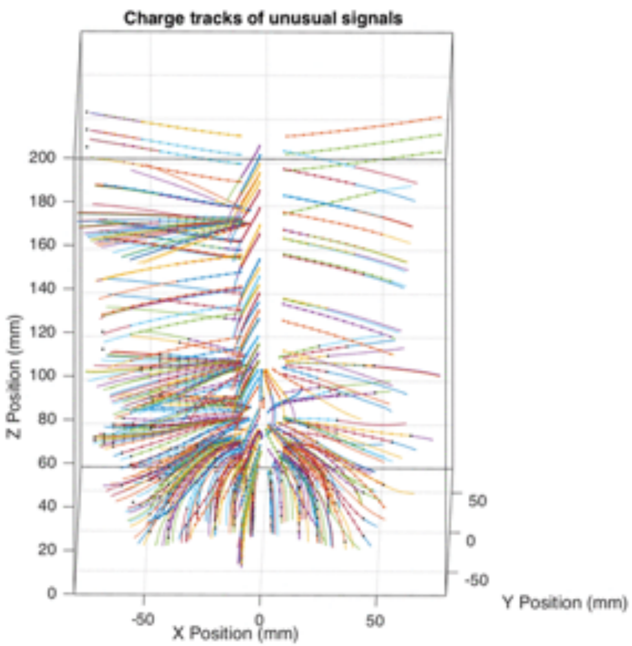
- For a collection of points $C_i$ on level $i$ of $T$ which represent a subset of points in $S$ the following rules must be enforced:
  - $C_i \subset C_{i-1}$ - Nesting: any point $p \in S$ that exists in $C_i$ must have an associated node in all lower levels.
  - $\forall_p \in C_{i-1}$ - Covering: for every $p \in C_{i-1}$ there exists one $q \in C_i$ such that $d(p,q) \leq 2^i$ where the node for $q$ is the sole parent of the node for $p$.
  - $\forall_p, r \in C_i, d(p,r) > 2^i$ – Separation: For all $p, r \in C_i$ then $d(p,r) > 2^i$



# In Summary

- Several algorithms have been developed for fold-1
- Adaptions for multiplicity are hard
- Database needs to be validated experimentally
- Odd effects in basis set need to be investigated



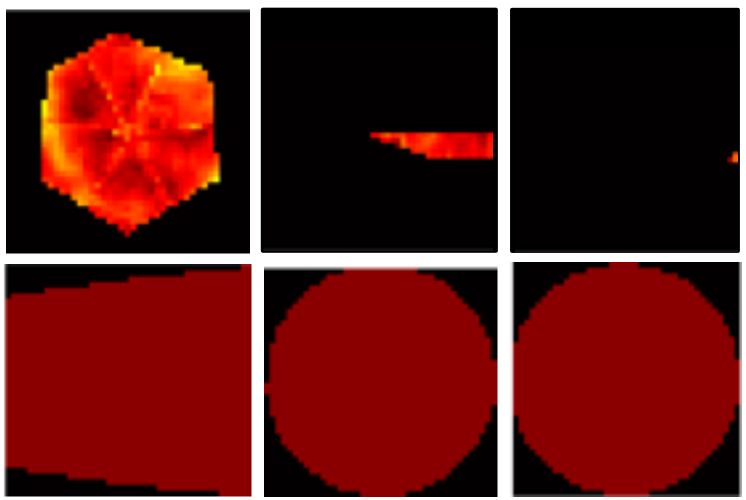




Fraser Holloway - F.Holloway@liverpool.ac.uk

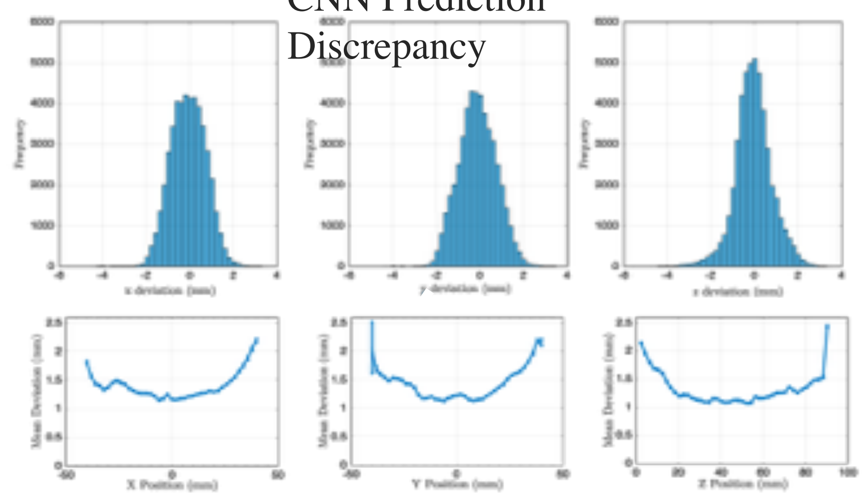
# Position Regression with Machine Learning



- 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



CNN Prediction Discrepancy



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