University of Glasgow

AI/ML at CLAS12

R. Tyson
NP-Twins
12th of May 2026

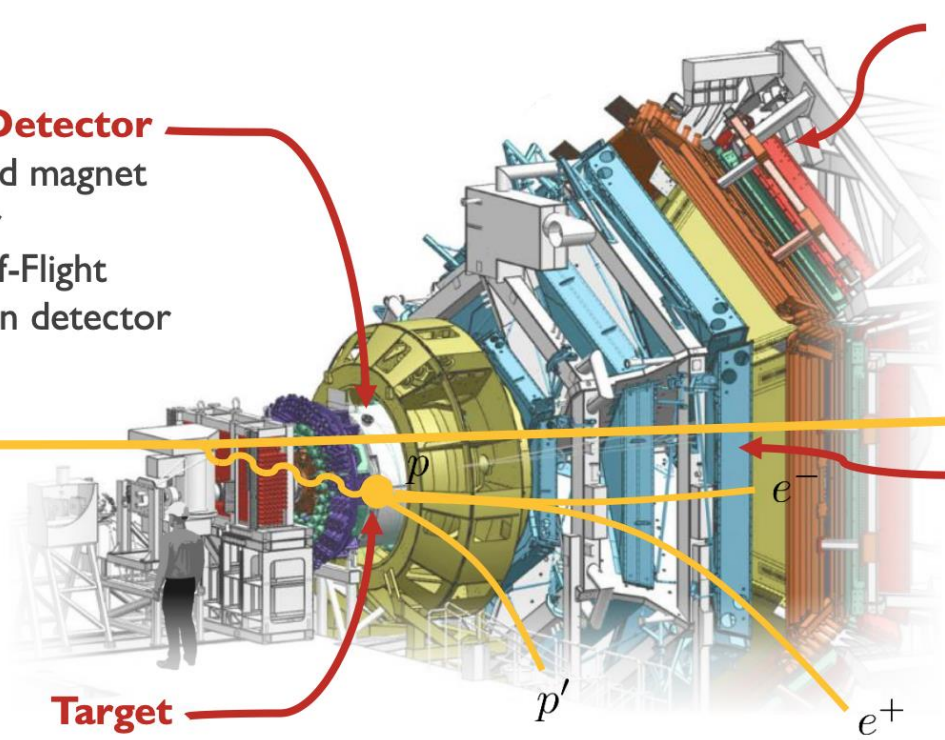
CEBAF Large Acceptance Spectrometer

CLAS12 @ Thomas Jefferson National Accelerator Facility (JLab)



Central Detector
- Solenoid magnet
- Tracker
- Time-of-Flight
- Neutron detector

Forward Detector
- Torus magnet
- Drift Chambers
- Time-of-Flight
- Calorimeters
- Cherenkov counters

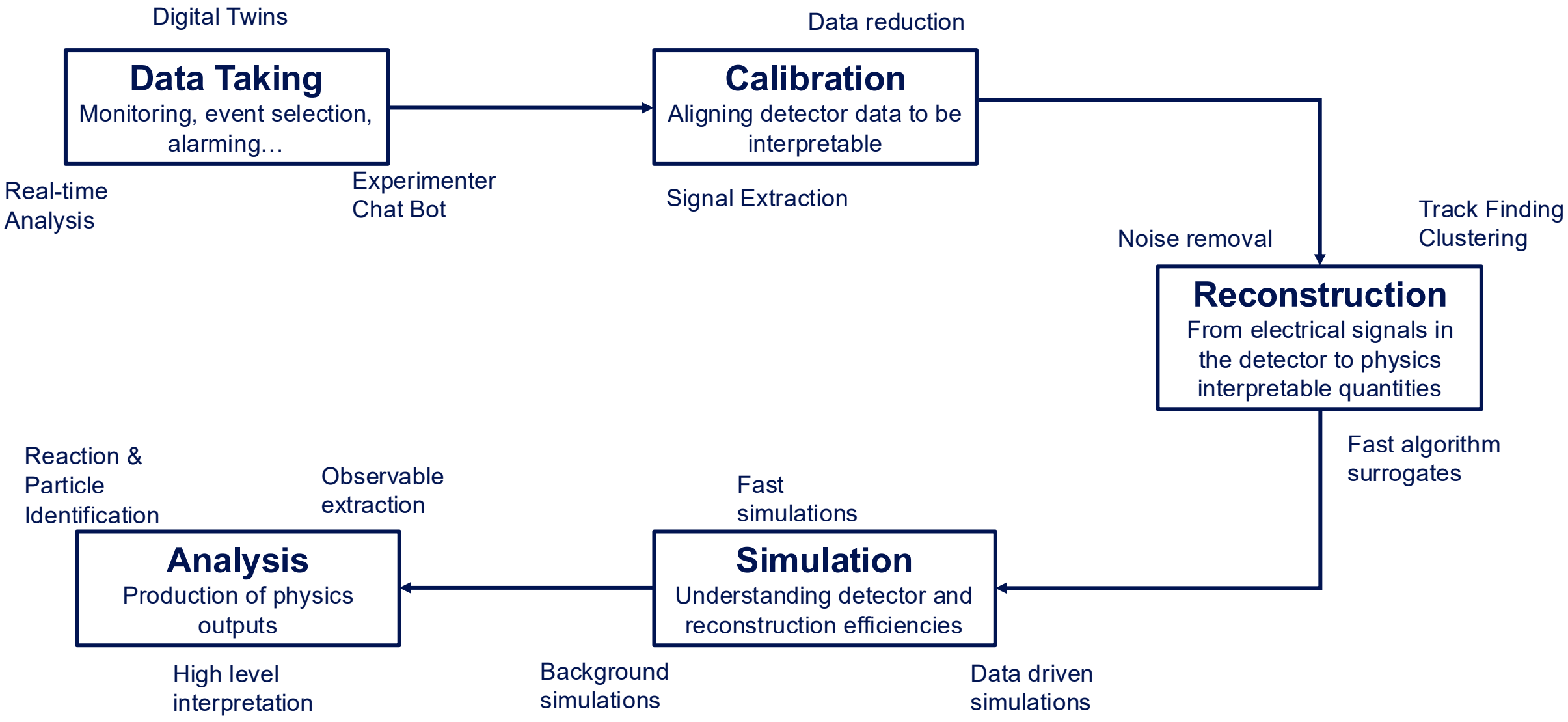


Beam
- 85% longitudinally polarized e^-
- Max. luminosity: 10^{35} $s^{-1}cm^{-2}$
- Energy up to ~10.6 GeV

Forward Tagger
- Calorimeter
- Time-of-Flight
- Tracker

Target
- **Proton**
- **Deuterium**
- Longitudinally pol. H/D
- Nuclear targets

Experimental Lifecycle and AI/ML



Denoising

P. Thomadakis, A. Angelopoulos, G. Gavalian, and N. Chrisochoides, Comp. Phys. Comm. 271, 108201 (2022).

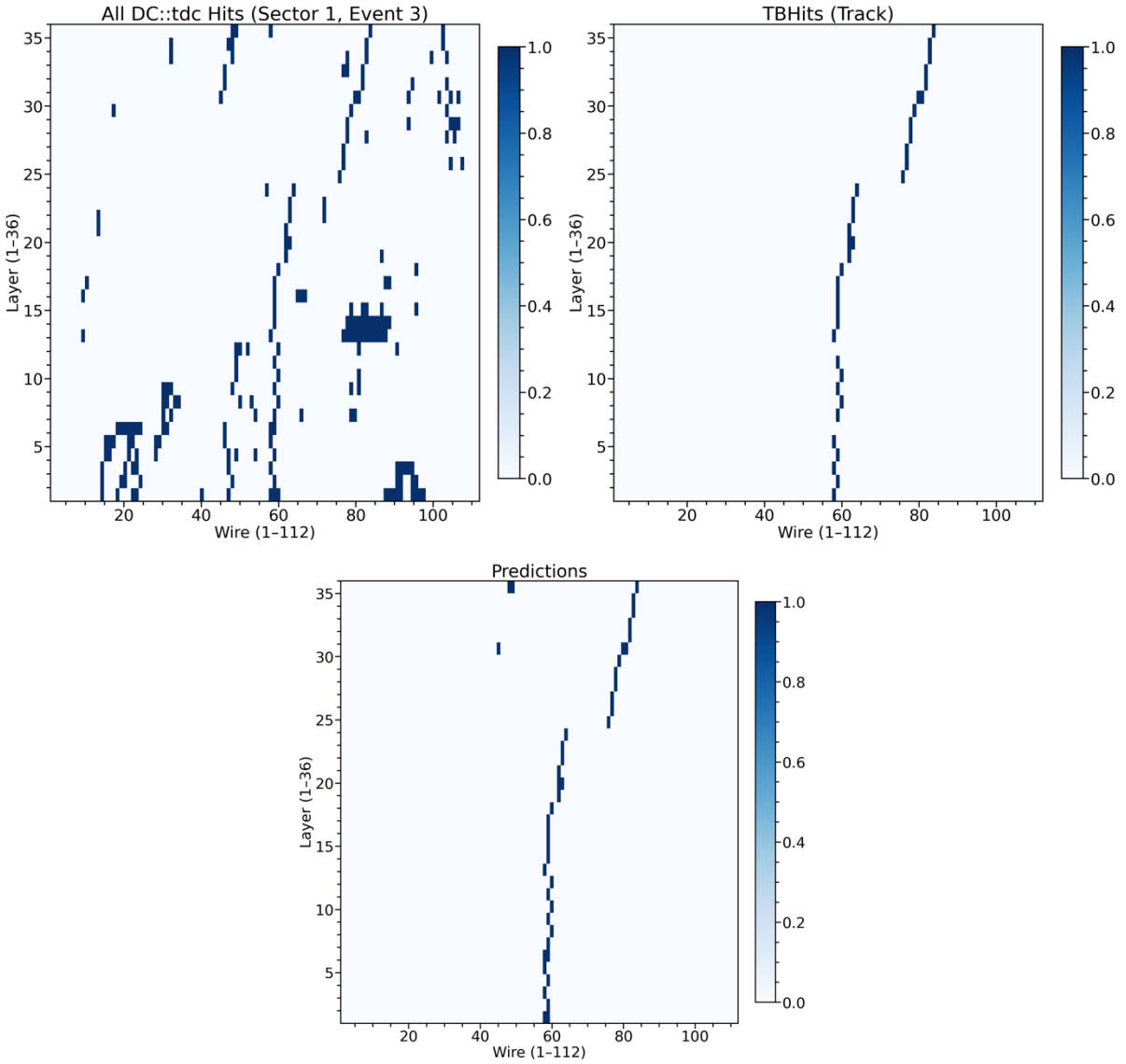
The CLAS12 tracking detectors can suffer from high noise levels, which makes tracking harder.

We therefore want to remove as much noise as possible

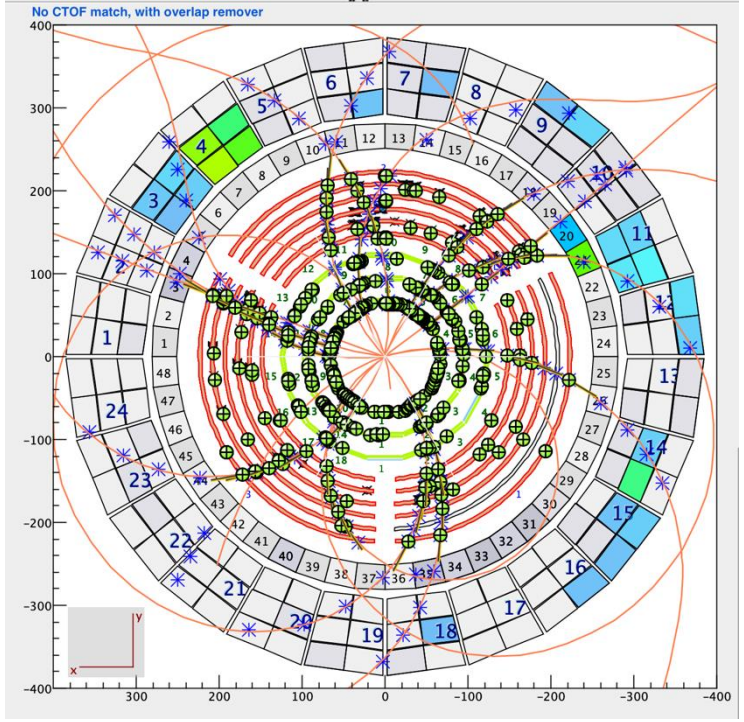
Define training sample with tracks found using conventional reconstruction, add noise identified using random trigger data.

Training data could also use simulation, we are testing this.

Train a convolutional autoencoder to predict whether each hit belong to signal or noise.



More Complicated Detectors



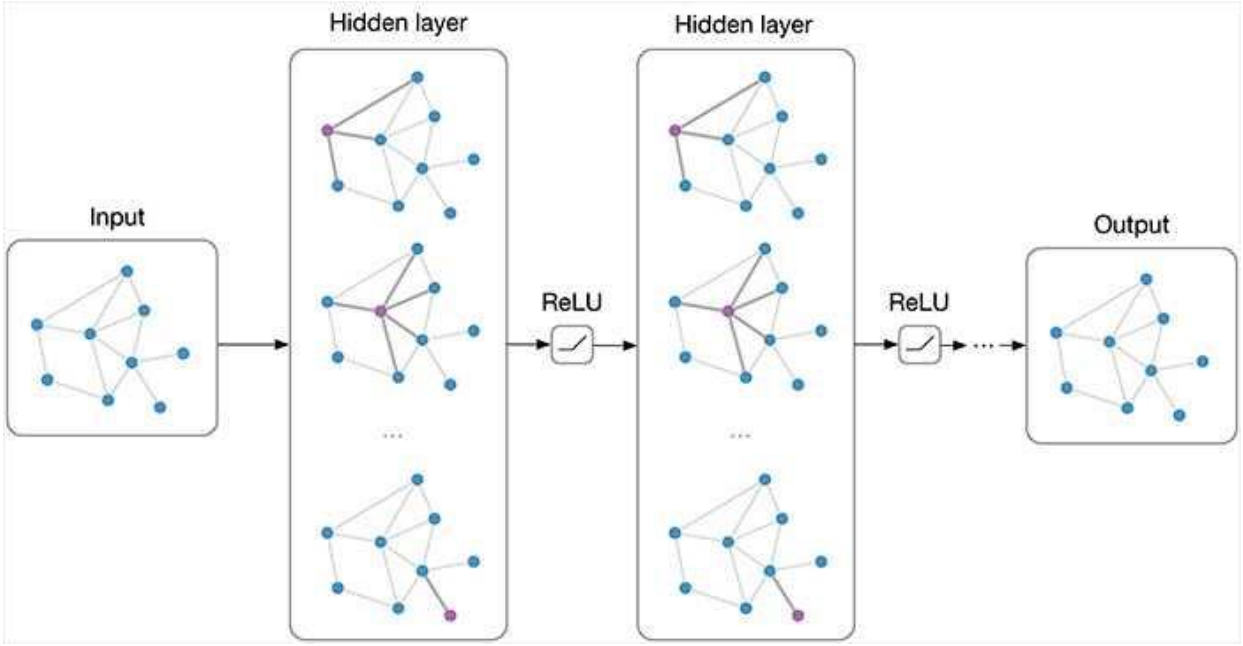
The CLAS12 drift chambers have small segmentation (112 wires in 6 layers), so we can use CNNs

This is not possible for Central Detector tracker:

- Several times more strips per layers
- A lot more noise as closer to target

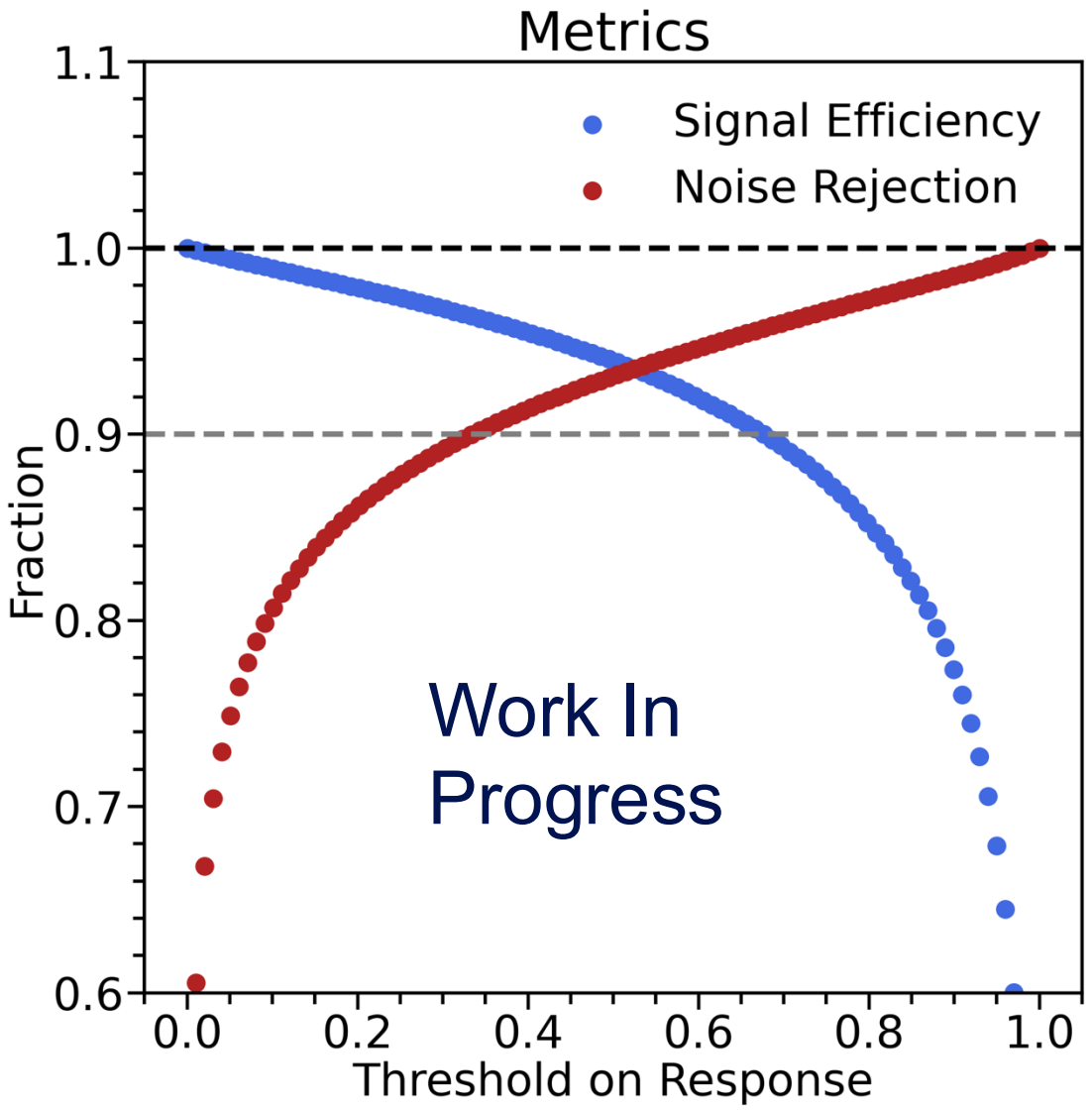
Use Graph Neural Networks (GravNet):

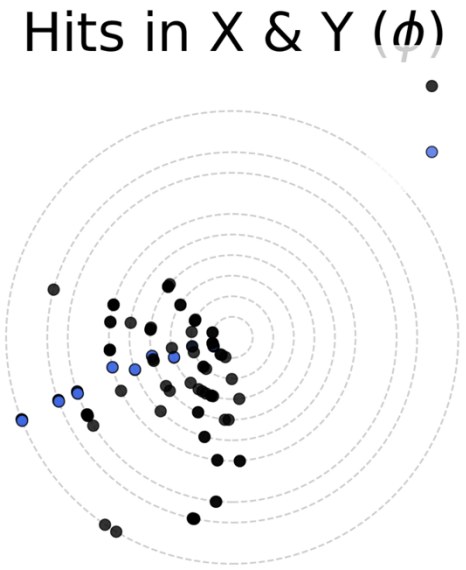
- Nodes ⇔ hits
- Edges ⇔ relationships (e.g. same track)
- Message passing (share information between hits)
- Output properties of hits (e.g. is signal/noise)

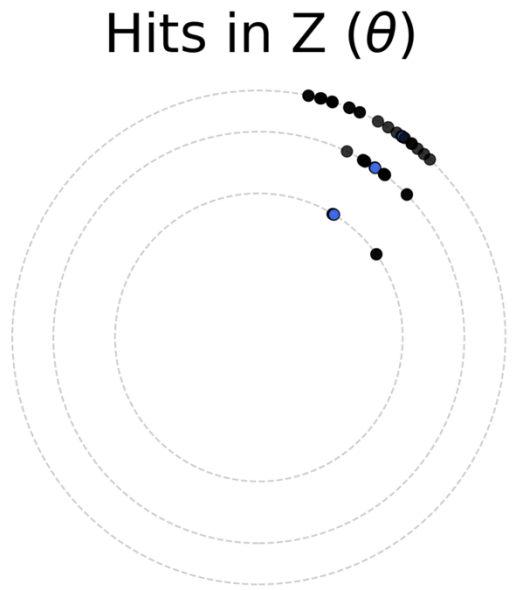


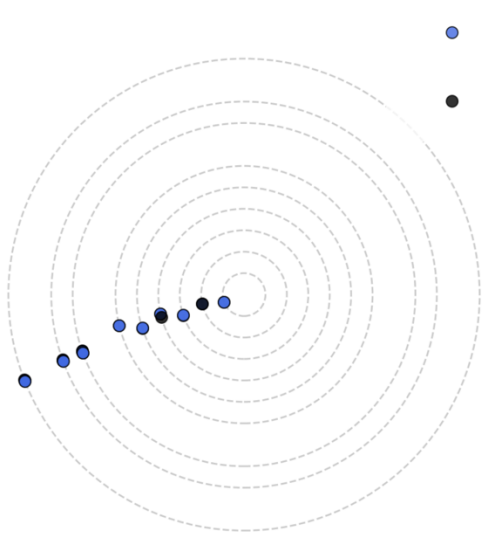
RT, V Ziegler,
P Chatagnon,
M Royanette,
T Cao

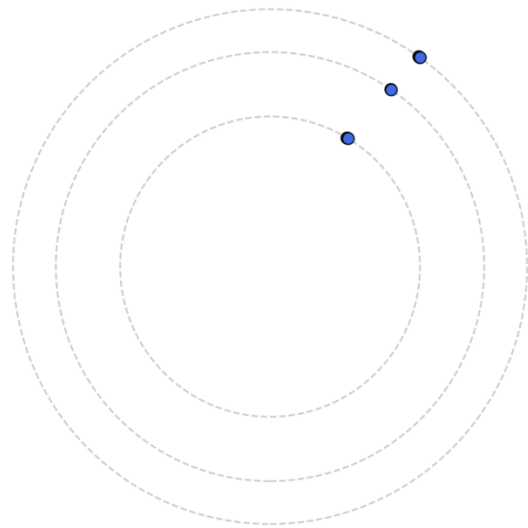
CD Denoising



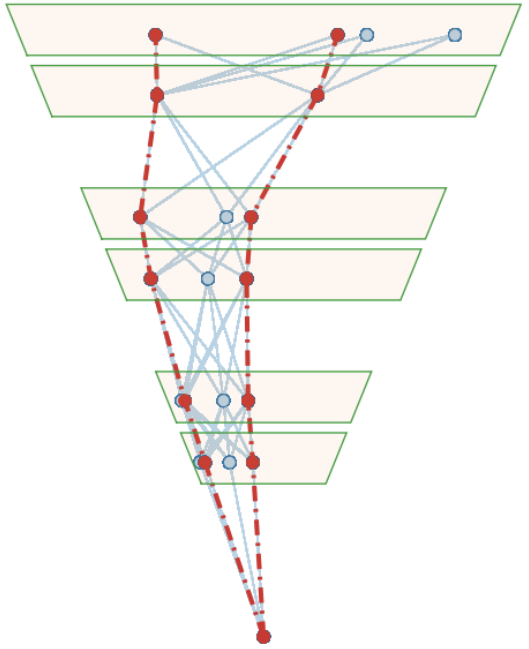








Track Finding

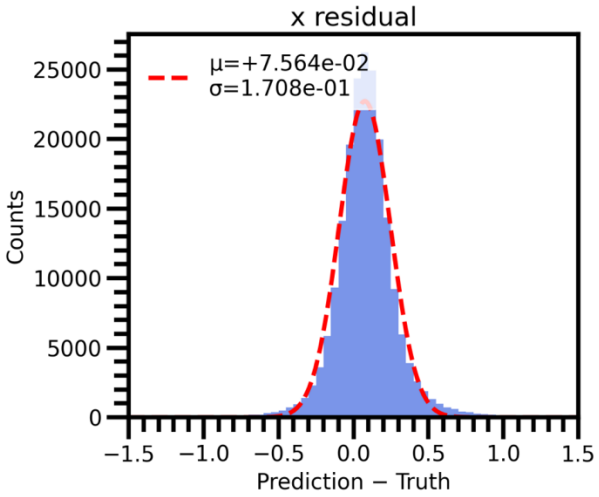


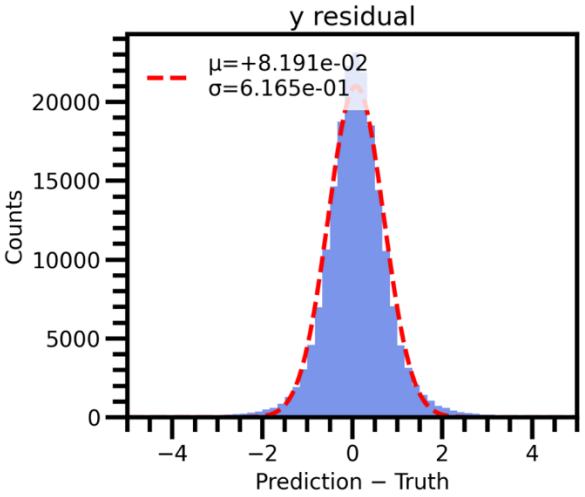
After the denoising stage, we aim to form tracks given the surviving hits

Using geometrical arguments, can define small enough number of permutations per event.

Several track finding algorithms:

- Cluster combo predictor
- Missing hit estimator
- Kalmann Filter initial state estimator for tracks (can also predict four momentum)





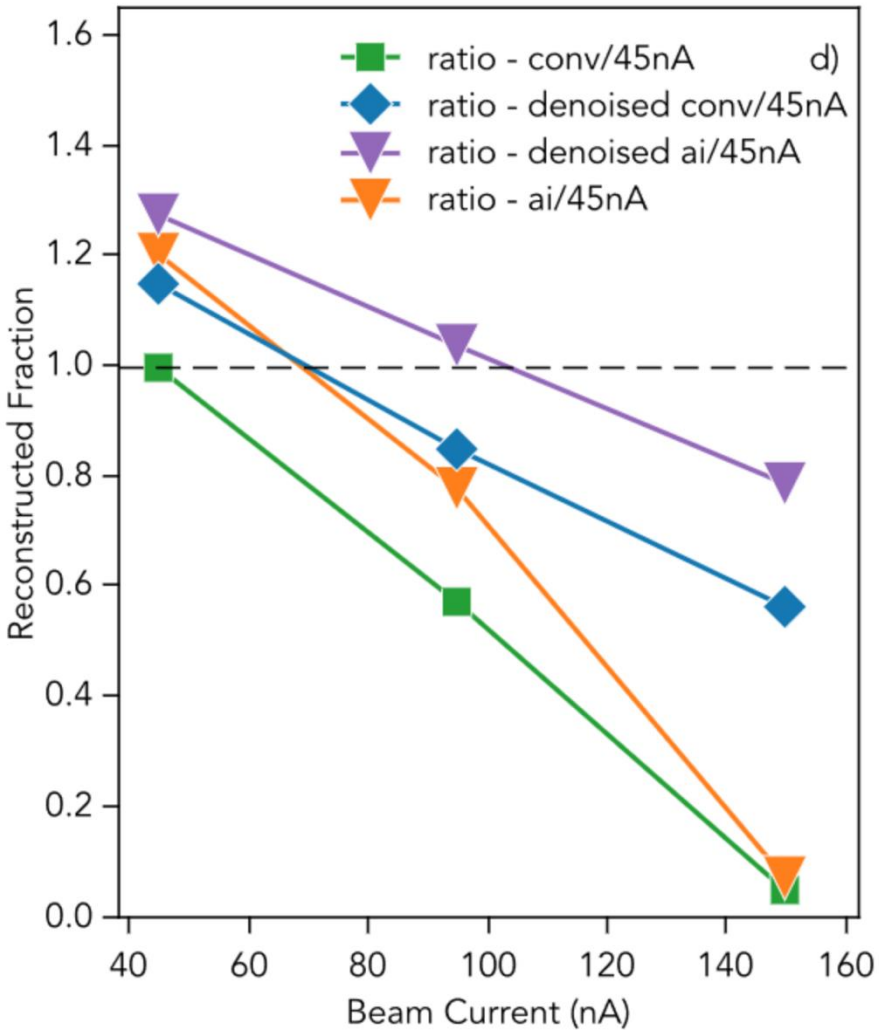
AI/ML Assisted Tracking

Putting together all tracking algorithms, CLAS12 gets an impressive increase in efficiency.

FAIR implementation in CLAS12 post-processing software.

Software experts are trained and doing own improvements – situation is actually better than shown here!

P. Thomadakis, A. Angelopoulos, G. Gavalian, and N. Chrisochoides, Comp. Phys. Comm. 271, 108201 (2022).
G. Gavalian, P. Thomadakis, A. Angelopoulos, N. Chrisochoides, R. De Vita, V. Ziegler, arXiv e-prints arXiv:2202.06869v3 (2022).
P. Thomadakis, A. Angelopoulos, G. Gavalian, and N. Chrisochoides, Comp. Phys. Comm. 276, 108360 (2022).
P. Thomadakis, K. Garner, G. Gavalian, and N. Chrisochoides, Comp. Phys. Comm. 287, 108694 (2023).



University of Glasgow

CLAS12 Calorimeter

G. Matousek, A. Vossen, NIM A 1082 2 170990 (2026)

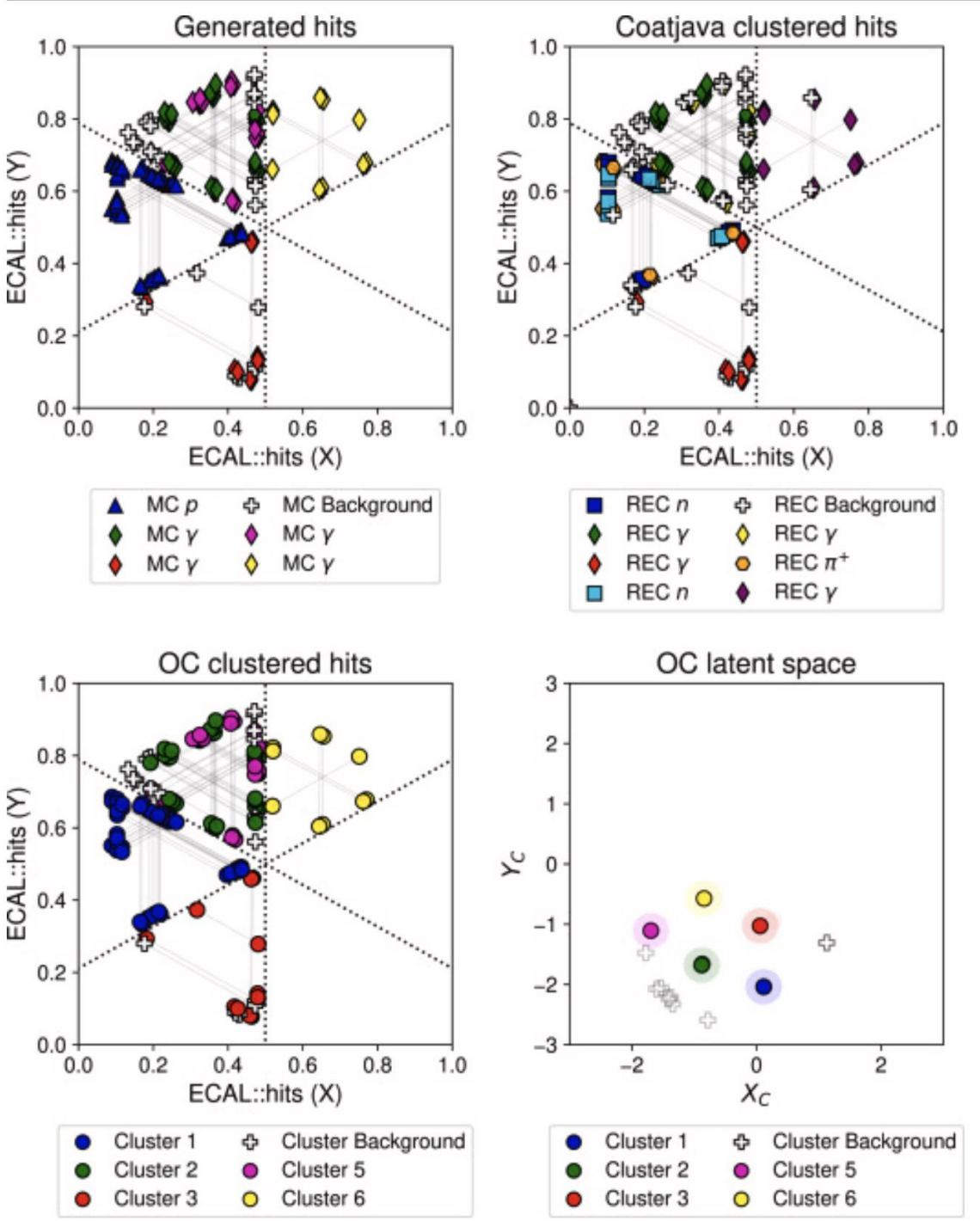
Reconstruction creates many fakes neutral clusters:

- Mostly split offs due to requirement for clusters to have decreasing energy from centroid and no gap
- Cluster with no found track
- Splash-back or secondaries

Aim is to decrease this neutron background (not as bad for gammas that have high β).

Utilise object condensation loss which uses attractive and repulsive charges to group hits belonging to same objects together.

Good performance in MC, ongoing efforts to implement these in data processing.



Cluster Finding

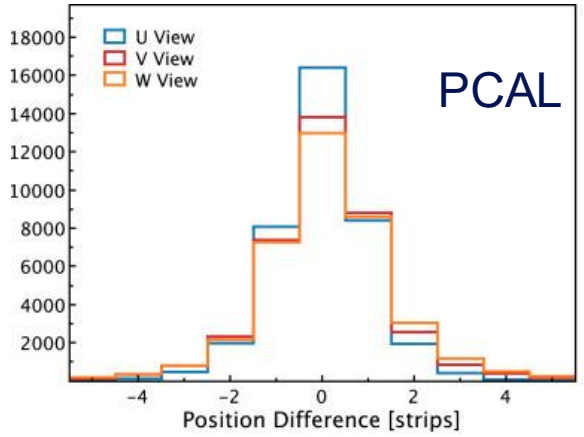
The swimming algorithm which associates tracks to hits in other detector subsystems can be slow.

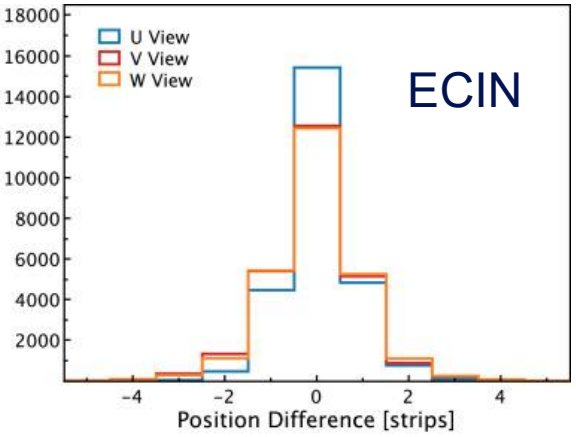
Instead, we can aim to train a neural network to predict the hit positions given a track (defined by its hit positions or four momenta)

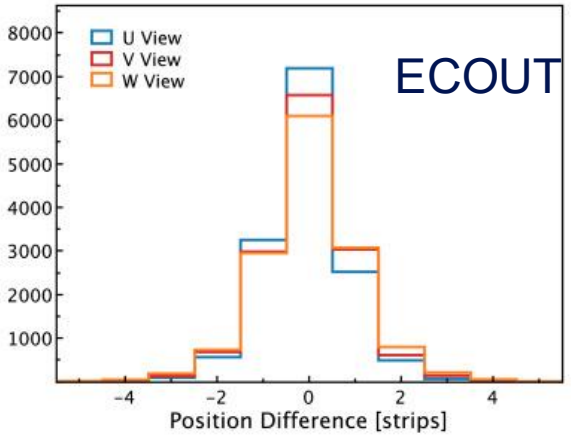
Define training sample using conventional swimmer used to associate hits to tracks.

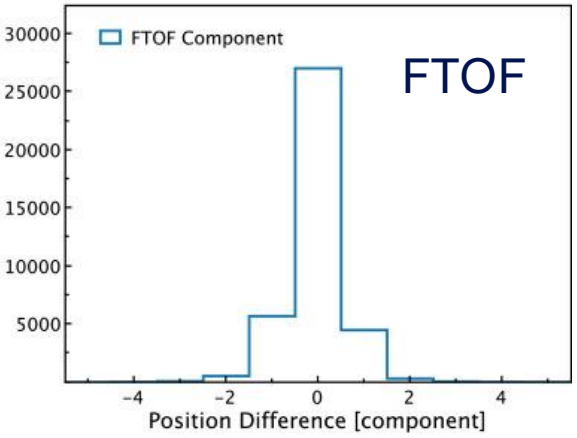
Example given is predicting hit positions in the CLAS12 calorimeters and time of flight detectors given the drift chamber hits.

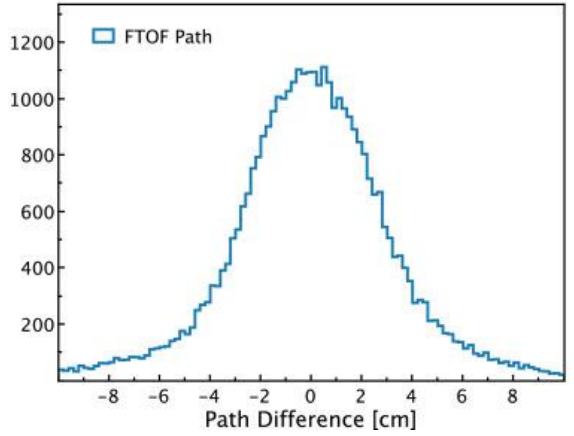
Can also predict path to TOF detectors for fast hadron PID.











RT, G. Gavalian Comp. Phys. Comm. 321 110032 (2026).

Online Electron Reconstruction

RT, G. Gavalian, D.G. Ireland, B. McKinnon, Comp. Phys. Comm. 290, 108783 (2023).

G. Gavalian, JINST 19 C05050 (2024).

RT, G. Gavalian Comp. Phys. Comm. 321 110032 (2026).

Efforts over the last couple of years to develop online electron reconstruction.

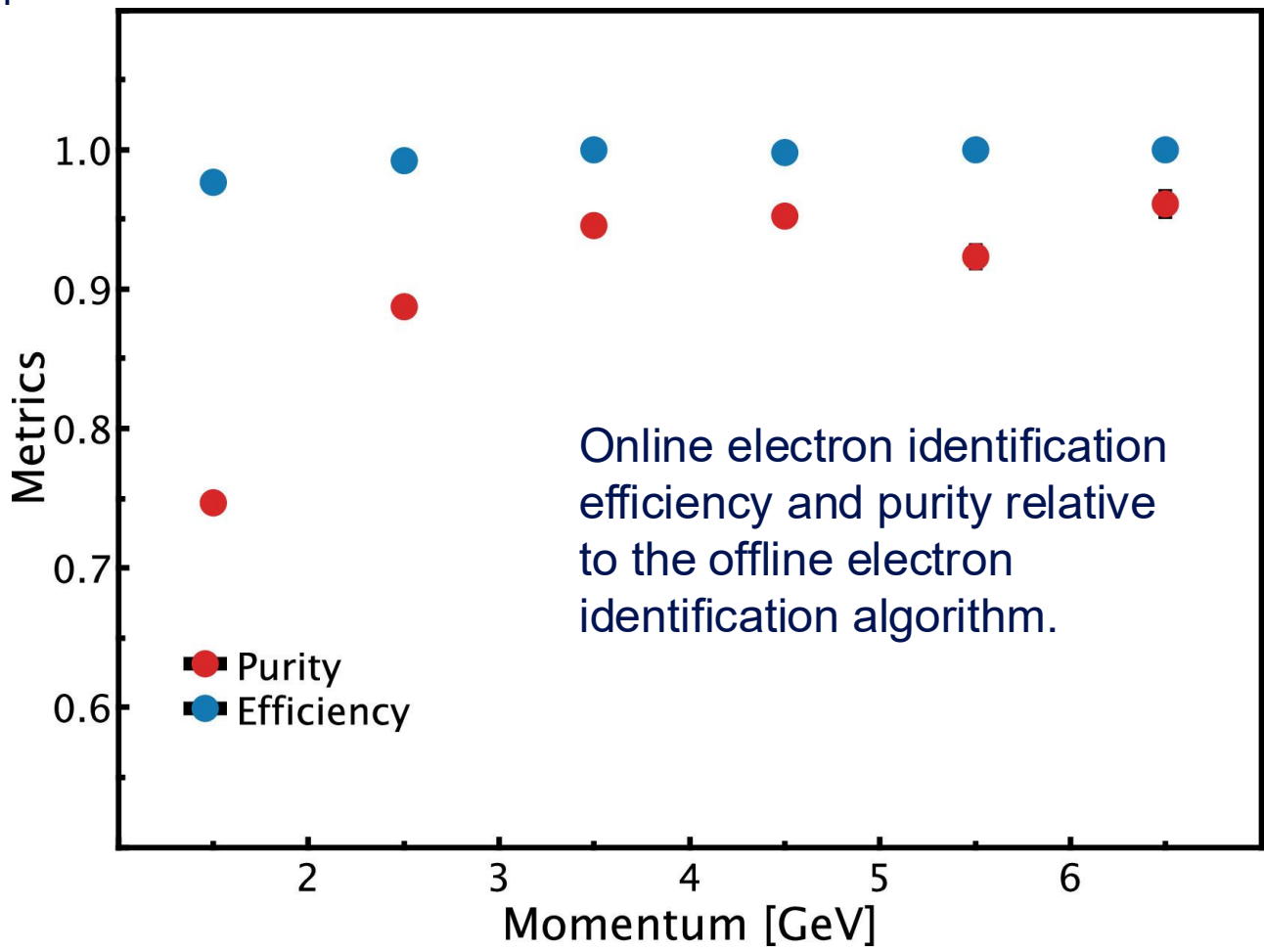
Online reconstruction chain would be implemented on CPU connected to DAQ stream.

Use several simple neural networks:

- Track finding
- Momentum and angle estimator
- Cluster finding
- Electron PID

Benefits:

- Improved online monitoring via real time analysis
- Improved trigger purity (40% in example here)



Online electron identification efficiency and purity relative to the offline electron identification algorithm.

Online Electron Reconstruction

RT, G. Gavalian, D.G. Ireland, B. McKinnon, Comp. Phys. Comm. 290, 108783 (2023).

G. Gavalian, JINST 19 C05050 (2024).

RT, G. Gavalian Comp. Phys. Comm. 321 110032 (2026).

Efforts over the last couple of years to develop online electron reconstruction.

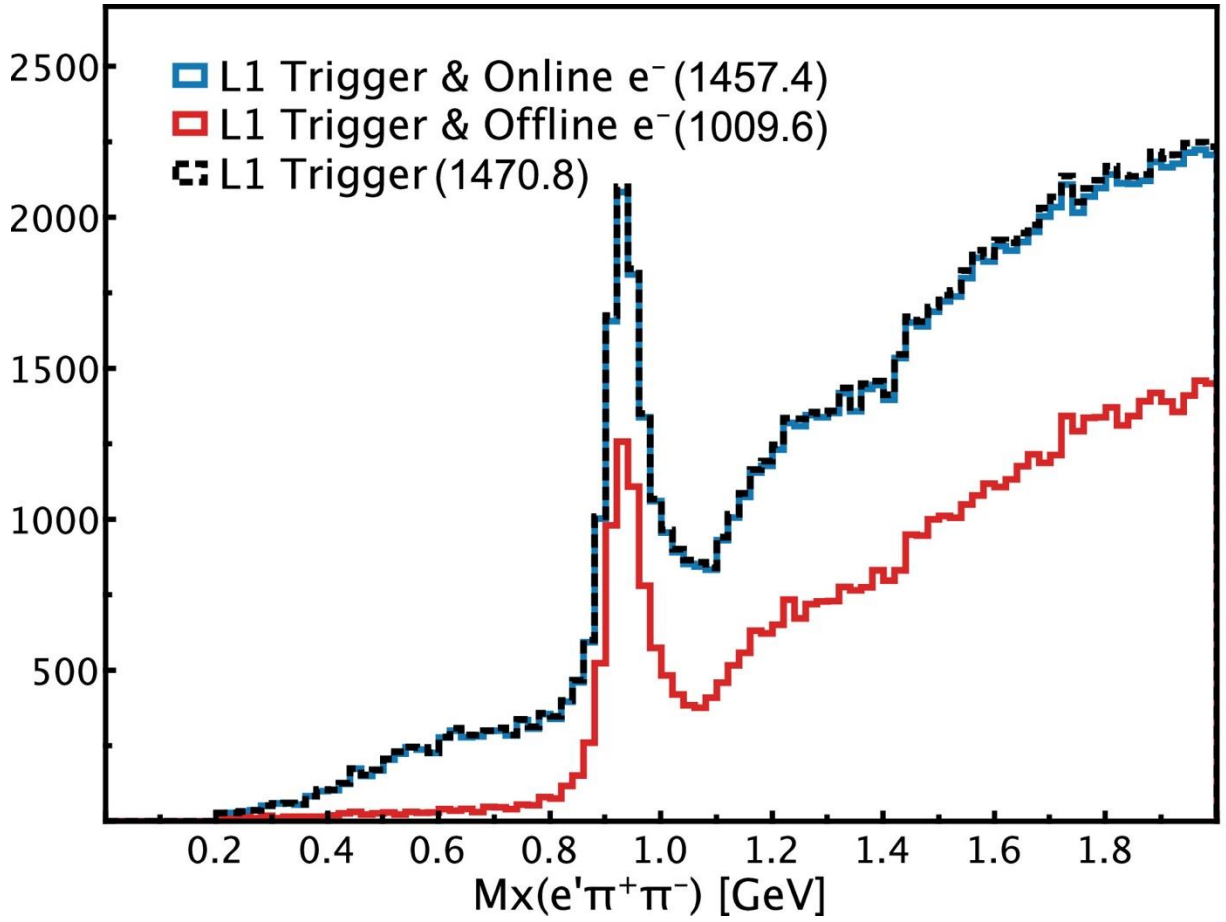
Online reconstruction chain would be implemented on CPU connected to DAQ stream.

Use several simple neural networks:

- Track finding
- Momentum and angle estimator
- Cluster finding
- Electron PID

Benefits:

- Improved online monitoring via real time analysis
- Improved trigger purity (40% in example here)



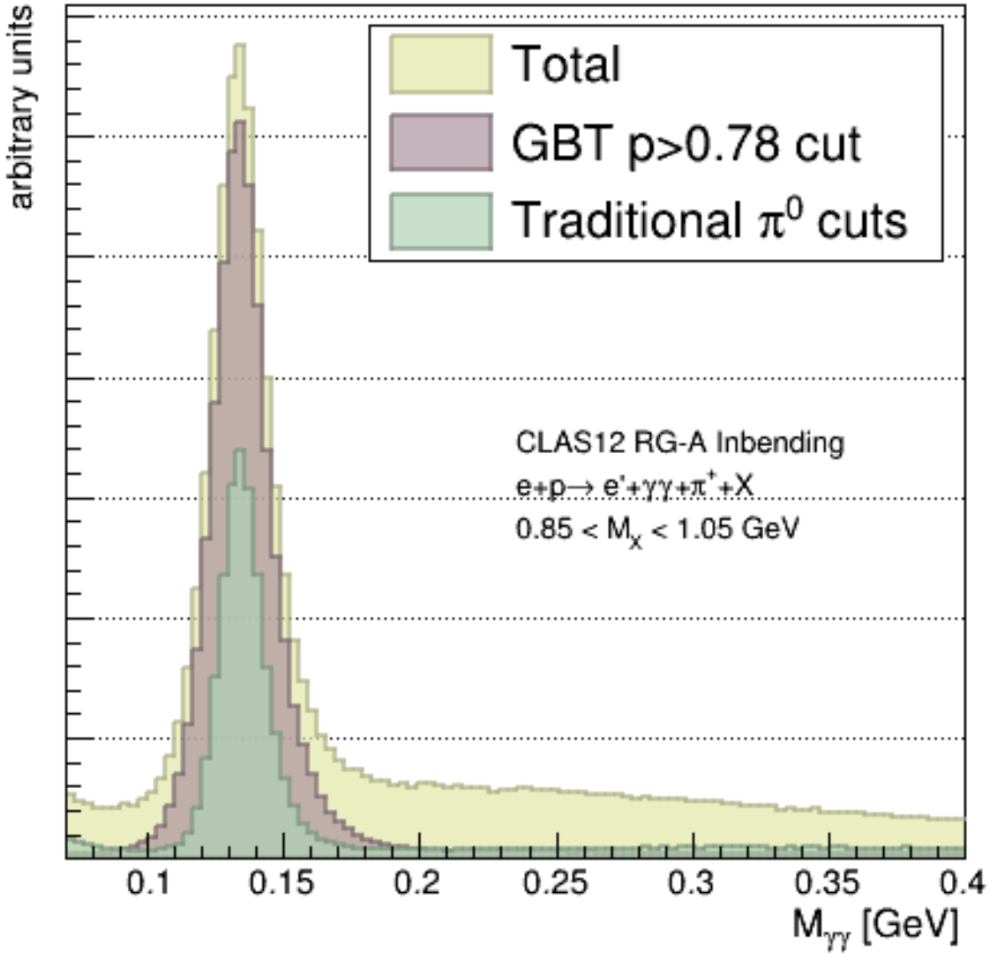
Offline PID

G.Matousek and A.Vossen,
JINST 19, 06, C06006 (2024).

Many different algorithms:

- **CD Neutrons**: Li Xu / Noemie Pilleux / Natalie Wright
- **FD Kaon ID**: Aron Kripko /Connor Pecar
- **Muon ID** : RT
- **Electron ID**: Mariana Tenorio Pita
- **Photon ID**: Gregory Matousek

In most cases, these utilize simple algorithms at the analysis stage (i.e. well calibrated, easy to use by non software experts)



PID Good Practice

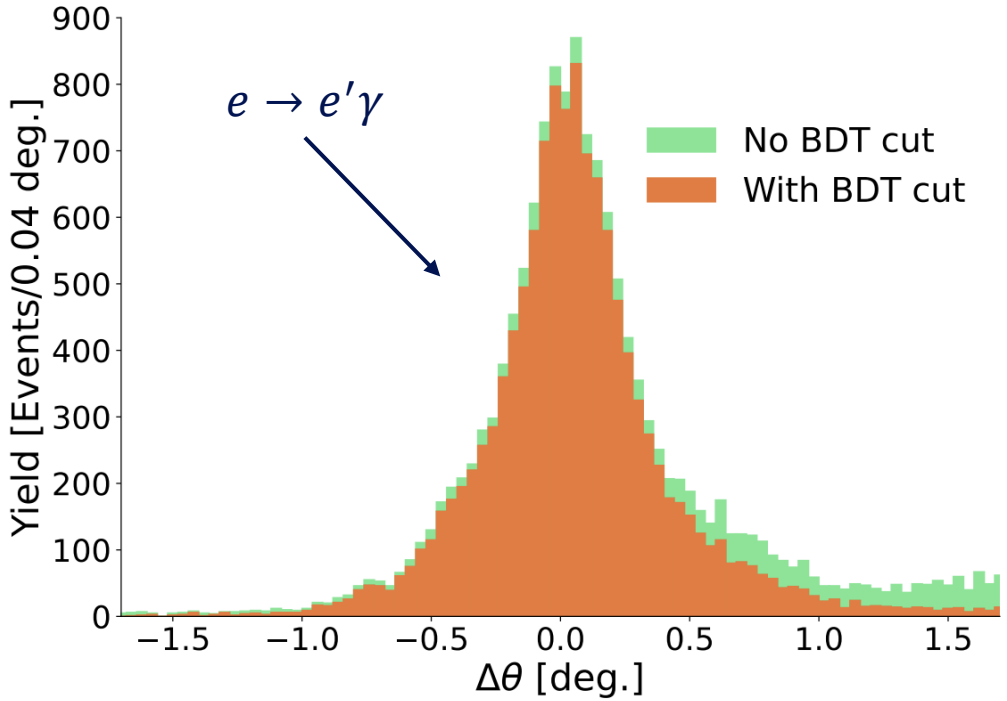
Algorithms trained on MC should be validated in CLAS12 data using exclusive reactions.

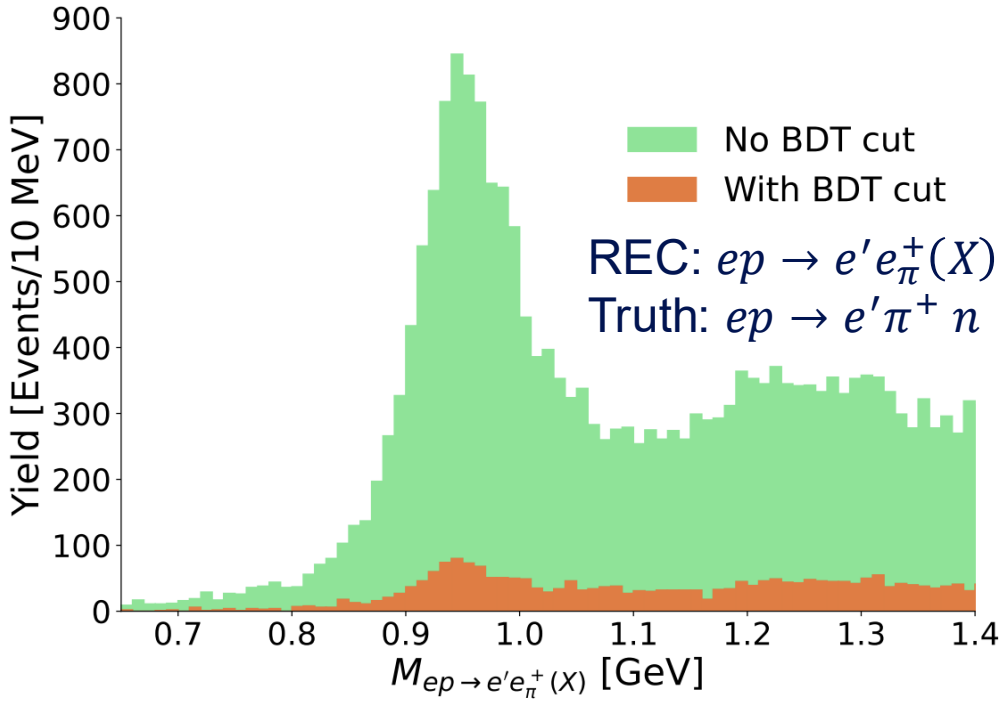
Good example by M. Tenorio Pita for electron PID.

Cross sections need data to MC efficiency corrections if algorithm trained on MC.

In general, aiming to define guidelines on AI/ML use in physics outputs:

- Prevent against algorithmic or training data driven biases
- Ensure non-ML experts can review analyses
- Improve AI/ML algorithms, make them FAIR





AI/ML Analysis Guidelines

RT
P Chatagnon

Spurious Correlations Model reliance on input features statistically correlated with target classes in the training data but not causally related to underlying physics.	Training data has particle types with different momentum ranges, and the trained model is biased towards momentum.	Training data creation, feature selection, model output correlation to variables, binning, decorrelation introduced in loss…
Representation Bias Range, diversity of the training samples is insufficient or different to data to which the algorithm will be applied	Training data is created from simulated data that is inaccurate and therefore not representative of the “real” data to which the model is applied.	Synthetic validation datasets with known conditions, non-simulated validation sets, comparison of model outputs on different datasets…
Sampling Bias Process of selecting training data systematically favours certain values or ranges of a physics observable, leading to an unbalanced or skewed dataset	Training dataset contains simulated physics signal; trained model sculpts observable to reproduce simulated signal, distorting true signal in data.	Input/output tests, synthetic validation datasets with known conditions, binning, impact on observable extraction technique (fits)…

Conclusion

Many interesting projects ongoing at CLAS12.

One indicator of this is the need to define guidelines for use of AI/ML in analyses.

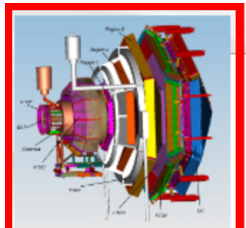
Several projects that I didn't mention:

- DVCS interpretation
- Fast MC and Fast Calorimeter Simulation
- Generalised PID
- Other detector specific tracking & PID (ALERT, RICH)

Several other interesting opportunities:

- RAGs for shift taking
- Developing good practices for AI/ML in Nuclear Physics analyses

CLAS12 AI/ML Group



page | discussion | view source | history

129.57.64.203 talk for this ip address log in

CLAS12 AI Group

navigation
- Main Page
- Community portal
- Current events
- Recent changes
- Random page
- Help

search
Search
Go Search

tools
- What links here
- Related changes
- Special pages
- Printable version
- Permanent link
- Page information

Contents [hide]

1 CLAS12 AI/ML Group
2 Meetings
3 Publications
3.1 Hall B Related
3.2 External
4 Software & Tools
4.1 Hall B Related Software
4.2 External Libraries
5 Projects
6 Guidelines
6.1 Generative AI

https://clasweb.jlab.org/wiki/index.php/CLAS12_AI_Group

[edit]

CLAS12 AI/ML Group | Meetings | Publications | Software & Tools | Projects | Guidelines

The general aims of the CLAS12 AI/ML Group are as follows:

- Establish standard procedures for AI/ML analyses within CLAS (eg reviews of models, training samples, validation, systematics...). The aim here is to facilitate the review process for both analysers and reviewers.
- Provide training opportunities and a forum for feedback from experienced AI/ML practitioners.
- Advertise existing analyses and software to create a common knowledge base so that we can avoid re-inventing the wheel.
- Identify opportunities for AI/ML at CLAS12 and create a point of contact for communication with detector experts, working groups and run groups.
- Enhance exchange and collaboration with the JLab Data Science Department, Scientific Computing and the Experimental Physics Software and Computing Infrastructure Group (EPSCI).

Please email tyson@jlab.org if you are interested in presenting at the AI Group meetings. In particular we are very interested in hearing about ongoing projects including work in progress, and about any software or tools developed for CLAS12.

We would like to keep a list of ongoing AI/ML projects in Hall B (see Ongoing Projects tab). We also would like to keep a list of AI/ML software and publications relevant to CLAS12 (see Software & Tools tab and Publications tab). Please consider adding your work to these lists.

Subscribe to our mailing list Clas12_ai for updates and meeting reminders. You can subscribe to the mailing list here.

Email the mailing list or tyson@jlab.org for any comments or questions.

University of Glasgow

Back-up Slides

Graph Neural Networks

Graph are made of **nodes** (eg **hits**) & **edges**. They have variable size based on the number of hits per event.

Graph Neural Networks utilise graphs. During learning, "**message passing**" updates each hit with information from other hits.

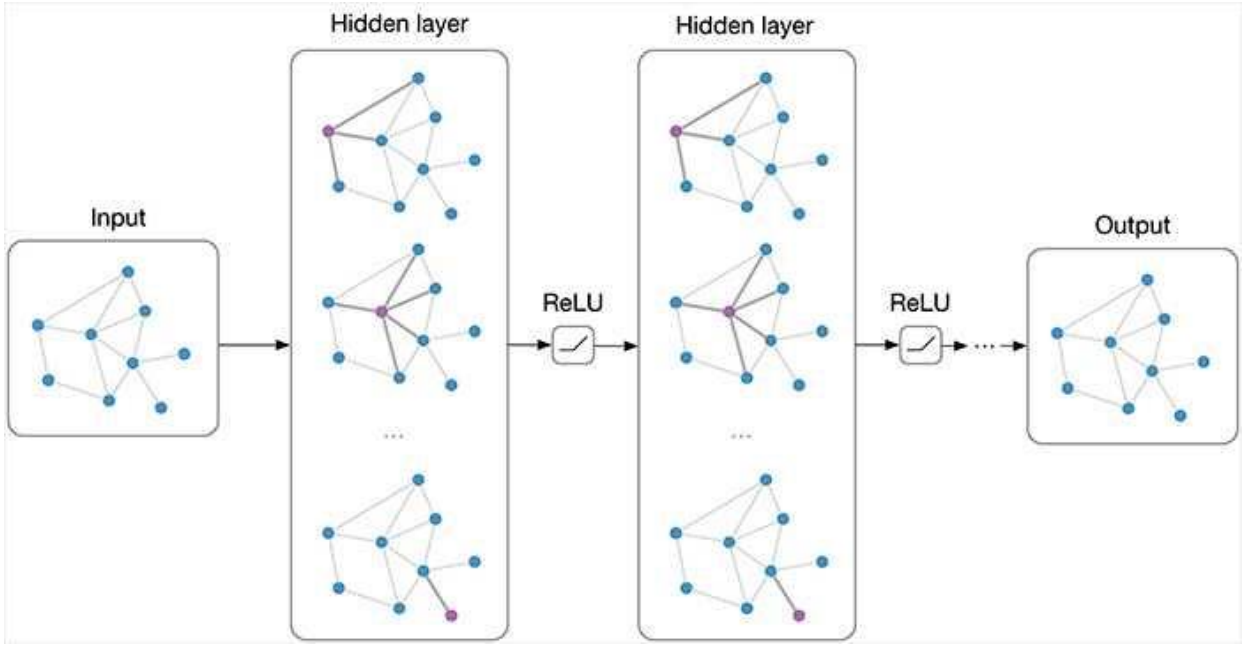
This is learned (eg what information is passed is the output of a small neural network).

GNNs can be used in different ways:

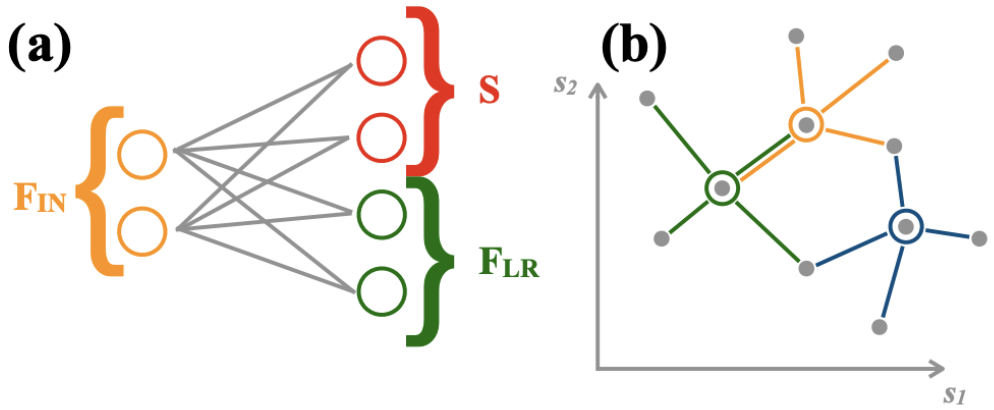
- **Node classification (denoising)**
- Edge classification (track finding)

We are going to use an approximation of GNNs with **fixed-size** event representation.

$\Longrightarrow$ Avoids dynamic memory allocation and makes the architecture **implementable on FPGA** via conversion to HLS (see arXiv:2008.03601).



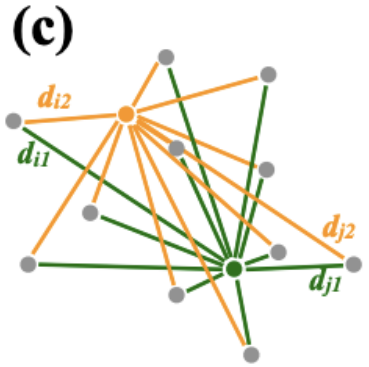
GarNet Layers (arxiv:902.07987)



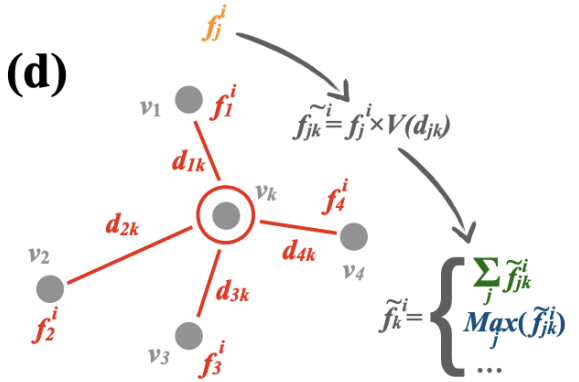
(a & b) The model first passes each hit through a small neural network to produce:

- **Learned features:** extra information extracted from the hit
- **Spatial information:** ~ coordinates in an abstract space

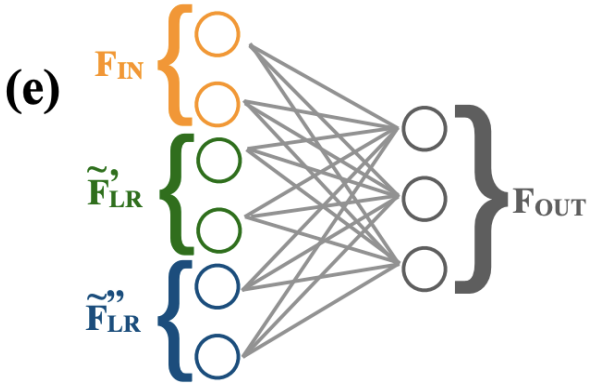
(c) This spatial information represents distances from each hit to a set of helper nodes called **aggregators**.
Every hit is connected to all aggregators using these distances.



(d) Aggregators collect info from all connected hits. The contribution from each hit is **weighted by distance**.
Each hit then receives the aggregated information back (~info of whole event).



(e) The original hit features and the aggregated information are combined and passed through a final neural network to produce the model output.
⟹ A number between **0 (photon)** and **1 (muon)**.



Aggregators guarantee a **fixed-size** event representation
⟹ Avoids dynamic memory allocation and makes the architecture **implementable on FPGA** via conversion to HLS (see arXiv:2008.03601).

University of Glasgow

J. Kieseler, Eur. Phys. J. C 80 886, (2020)

Object Condensation

Ideally we want a one shot algorithm:

- Hits in
- Tracks/Clusters Out
- Track/Cluster Properties (momentum, PID…)

Object condensation loss:

$L_{OC} = \omega_{att} L_{att} + \omega_{rep} L_{rep} + \omega_{\beta} L_{\beta} + \omega_{noise} L_{noise} + \omega_{p} L_{p}$

Model outputs β (convert to charge q), point (**x**) in new space and track parameters.

Create condensation points (CP) from hits with high β:

L_{att} pushes hits to CP from same track

L_{rep} pushes hits away from CP of different track

L_{β} encourage 1 CP per true track (eg 1 track per true track)

L_{noise} encourages no CP in noise to avoid forming track from background

L_{p} predicts track parameters

Let:

- $i \in \{1, \ldots, N\}$ index hits,
- $k \in \{1, \ldots, K\}$ index signal objects,
- $\mathbf{x}_i \in \mathbb{R}^D$ be the embedding of hit i,
- $\beta_i \in (0, 1)$ be the condensation score of hit i,
- $q_i = q(\beta_i)$ be the charge derived from β_i,
- $\mathcal{S}_k$ be the set of hits belonging to object k,
- $c(k)$ be the condensation hit for object k, defined as

$$c(k) = \arg\max_{i \in \mathcal{S}_k} \beta_i,$$

- $\mathbf{x}_{c_k}, \beta_{c_k}, q_{c_k}$ denote the embedding, beta, and charge of the condensation point,
- $d_{ki} = \|\mathbf{x}_{c_k} - \mathbf{x}_i\|_2$.

Hyperparameters:

- attractive radius δ_v,
- repulsive radius δ_d.

$N_{\text{att}} = \sum_k |\mathcal{S}_k|$ is the number of contributing hit–object pairs.

$$L_{\text{att}} = \frac{1}{N_{\text{att}}} \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_k} q_{c_k}\, q_i\, \left(d_{ki} - \delta_v\right)^2,$$

$$L_{\text{rep}} = \frac{1}{N_{\text{rep}}} \sum_{k=1}^{K} \sum_{i \notin \mathcal{S}_k} q_{c_k}\, q_i\, \max(0,\ \delta_d - d_{ki}),$$

$$L_{\beta} = \frac{1}{K} \sum_{k=1}^{K} \left(1 - \beta_{c_k}\right).$$

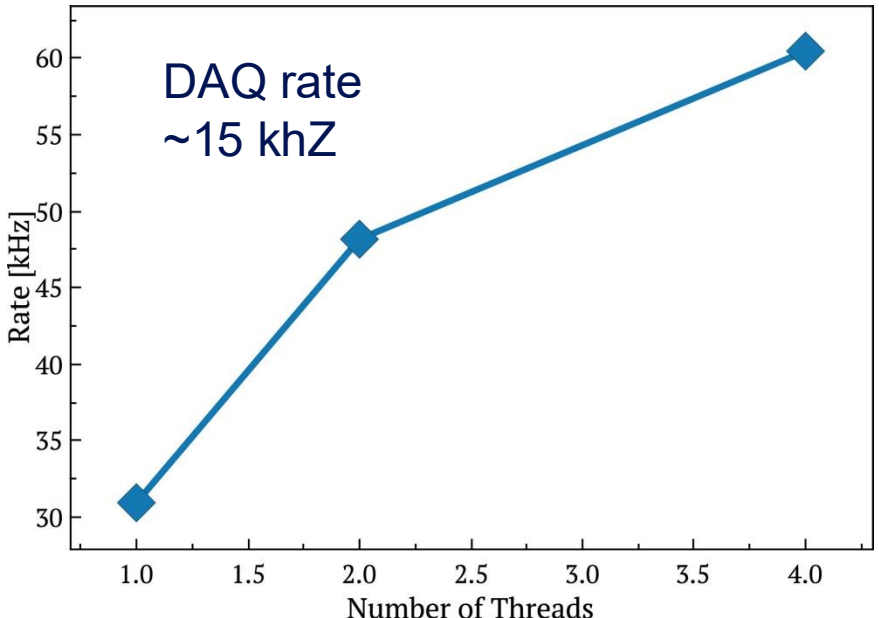
$$L_{\text{noise}} = \frac{1}{N_{\text{bg}}} \sum_{i \in \text{background}} \beta_i,$$

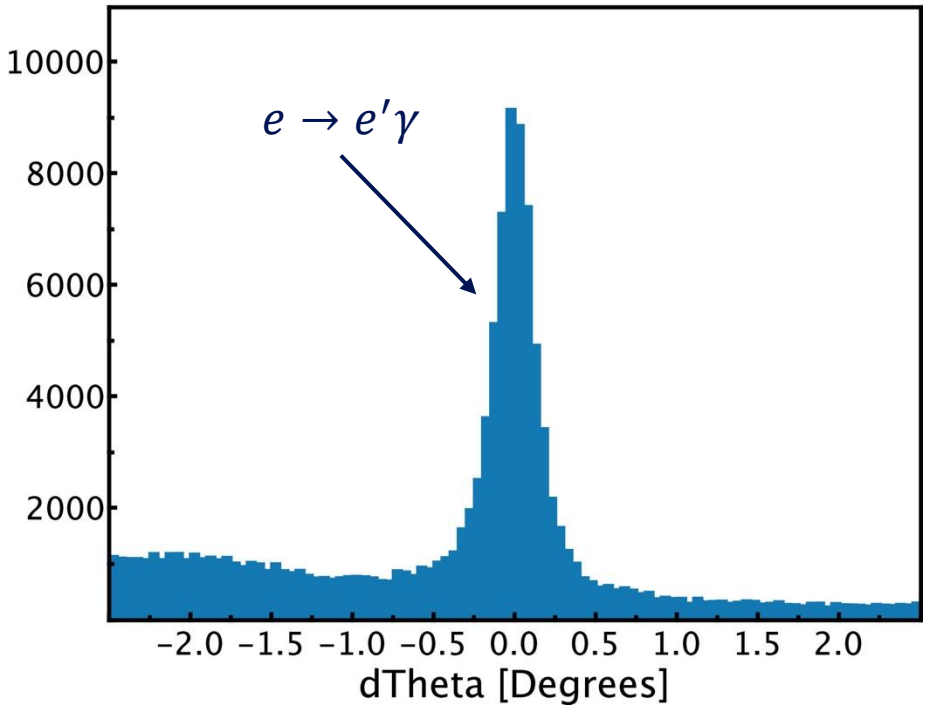
Online Electron Reconstruction

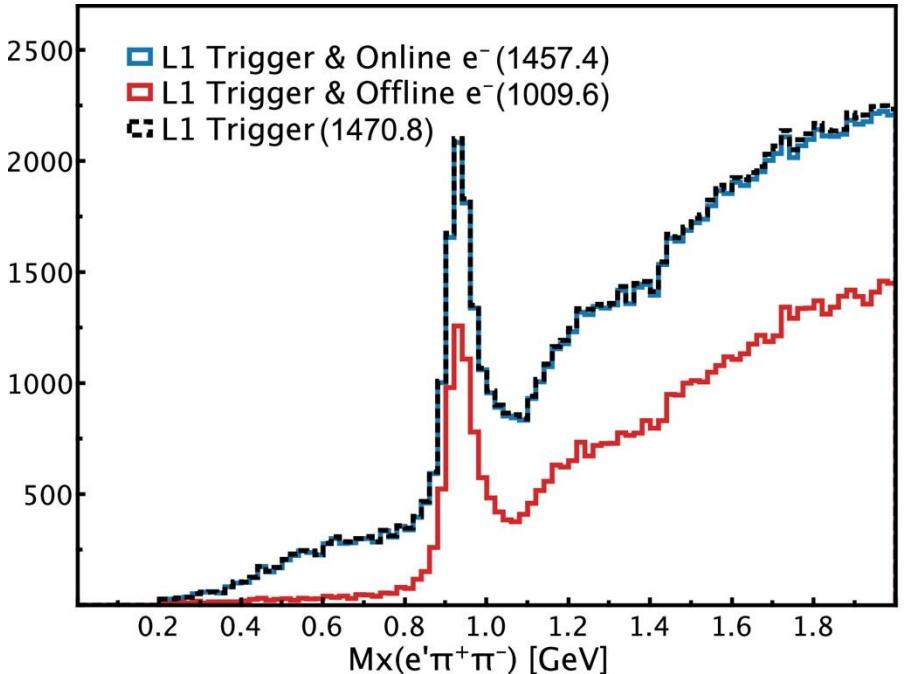
Offline electron PID is itself inefficient

Use $e \rightarrow e'\gamma$ to ID true electrons for training sample, $ep \rightarrow e'\pi^+\pi^-(p)$ for validation

Would lead to a data reduction of 40%

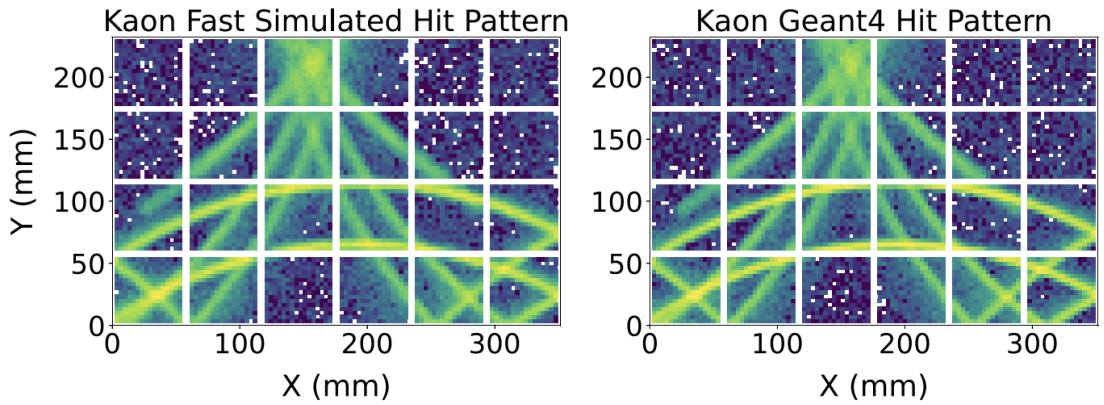






Fast Detector Simulations

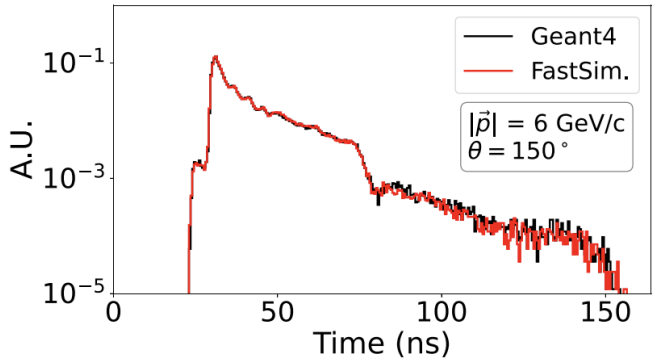
J Giroux, M Martinez, C Fanelli, Mach. Learn.: Sci. Technol. 6 040501 (2025)



Some detectors take up a large fraction of Geant4 simulation time:

- Calorimeters, need to simulate expensive photon/electron showers
- RICH/DiRC detectors have to simulate photon reflections and trajectories

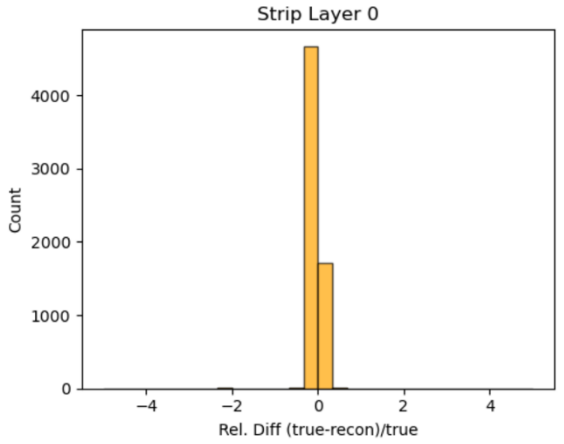
→ @ EIC

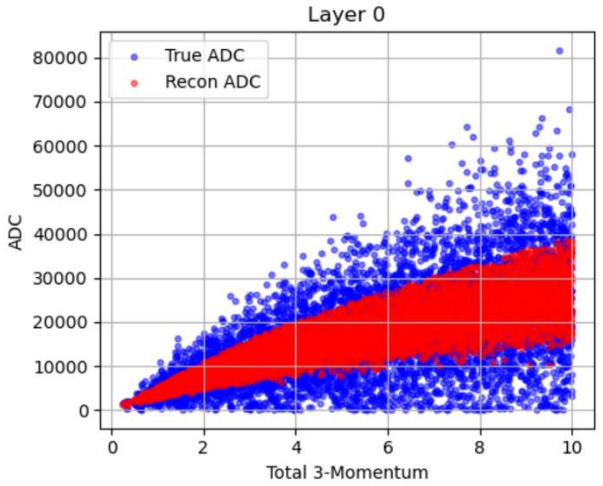


Instead we can use ML:

- Generative Adversarial Networks – if data isn't too sparse
- Normalising Flows
- Graph Neural Networks?

Work ongoing for CLAS12 calorimeters →





Fast Simulations

D. Darulis, RT, D. G. Ireland, D. I. Glazier, B. McKinnon, P. Pauli, arXiv e-prints arXiv:2207.11254 (2022).

Simulations are key to High Energy and Nuclear Physics experiment.

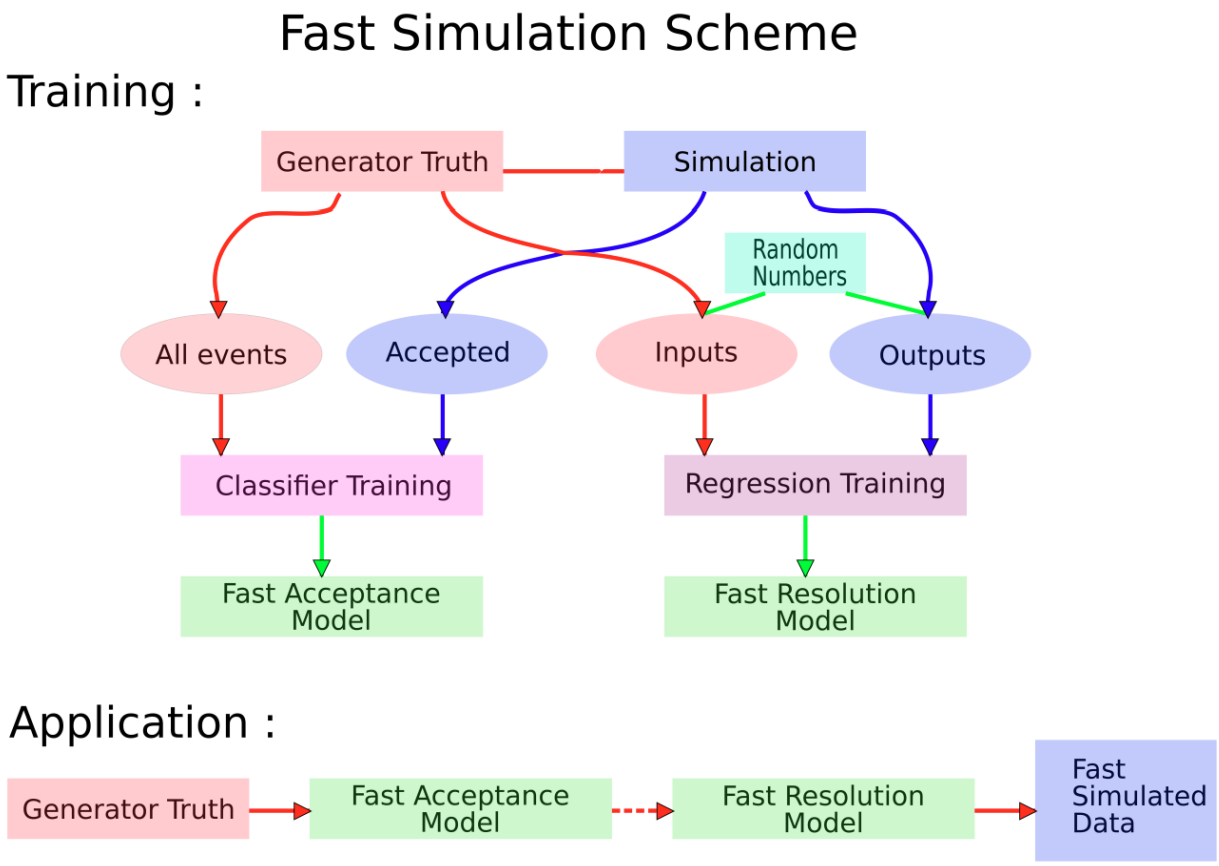
Simulations can be computationally expensive. Increased luminosities and detector complexity leads to increased computational overhead.

Instead, we want to create ML-based fast MC that can accurately reproduce simulated data.

Simulate + train once, apply many times

Two step process:

- Simulate Efficiency with density ratio estimation
- Simulate Resolution



Efficiency

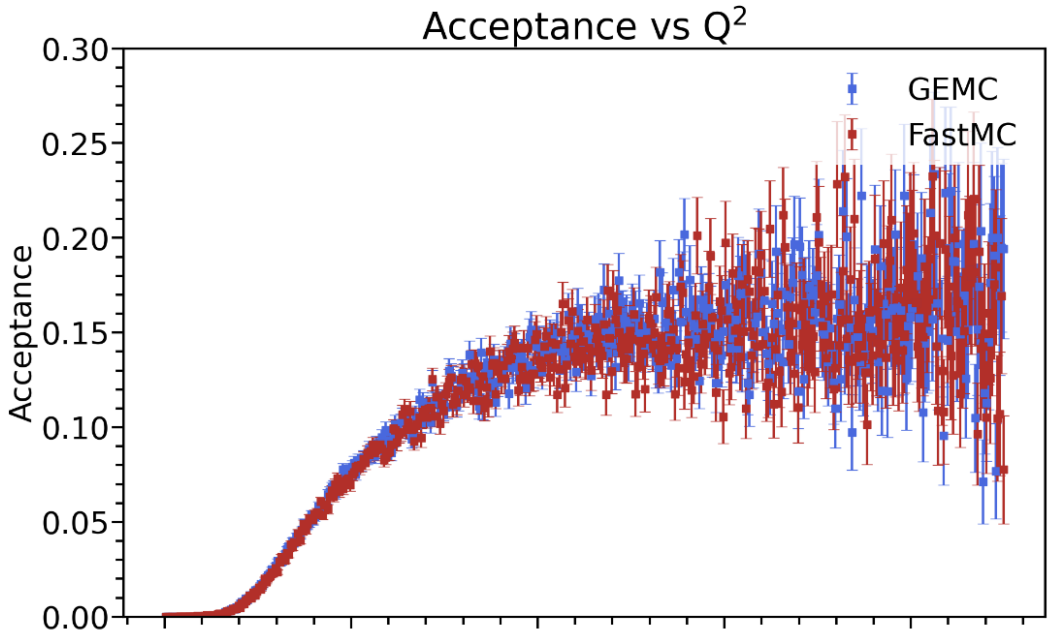
The efficiency of a detector can be estimated from the ratio of the probability densities for all generated and accepted events:

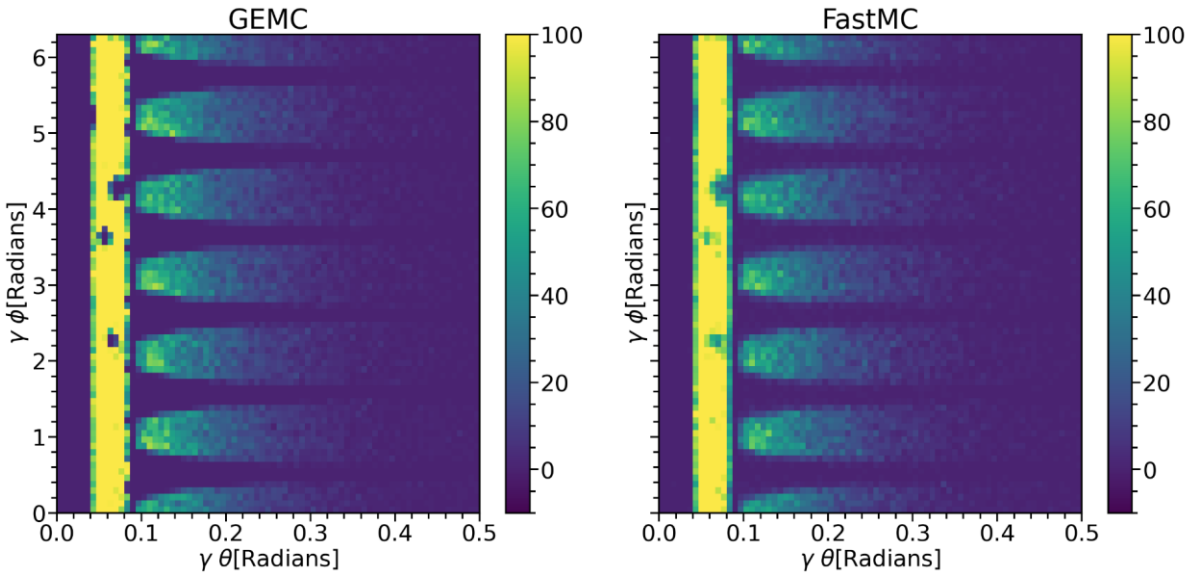
$$\varepsilon(x) = \frac{D_a(\boldsymbol{x})}{D_{all}(\boldsymbol{x})}$$

Create a training sample with all accepted events as class 1 and all events as class 0. Output for class 1:

$$f(x_i) = \frac{D_a(\mathbf{x})}{D_a(\mathbf{x}) + D_{all}(\mathbf{x})}$$

$$\Rightarrow \varepsilon(x) = \frac{f(x_i)}{1 - f(x_i)}$$





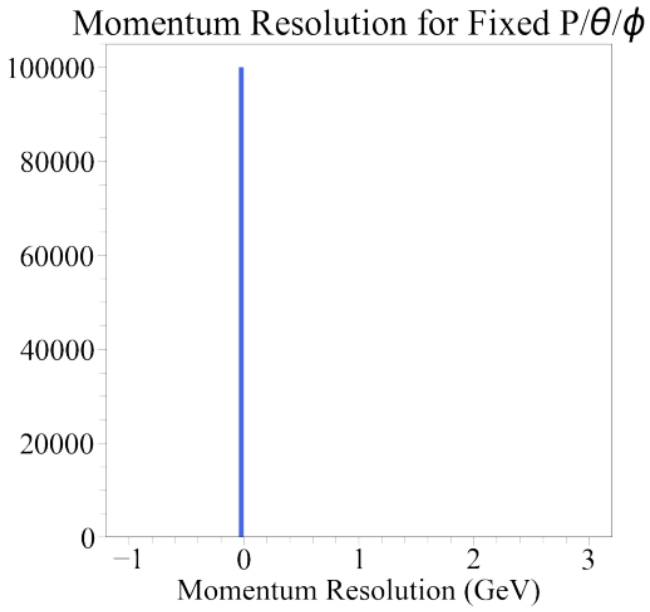
Resolution

If the detector measures $p_r = p_{true} + \delta p$, aim to learn the δp distribution (and on angles or other quantities).

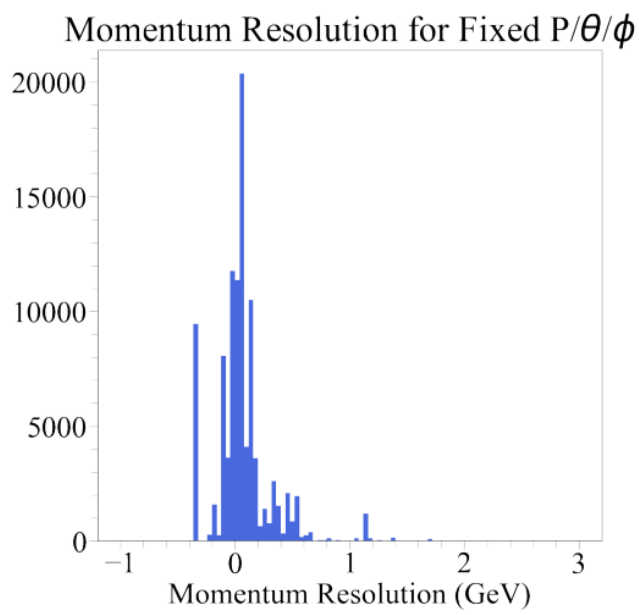
Overfit decision trees, accurately produces δp but they produce one value of δp for any one point in the training feature space.

Add random input variables, complexifies input feature space. Now predict different values of δp.

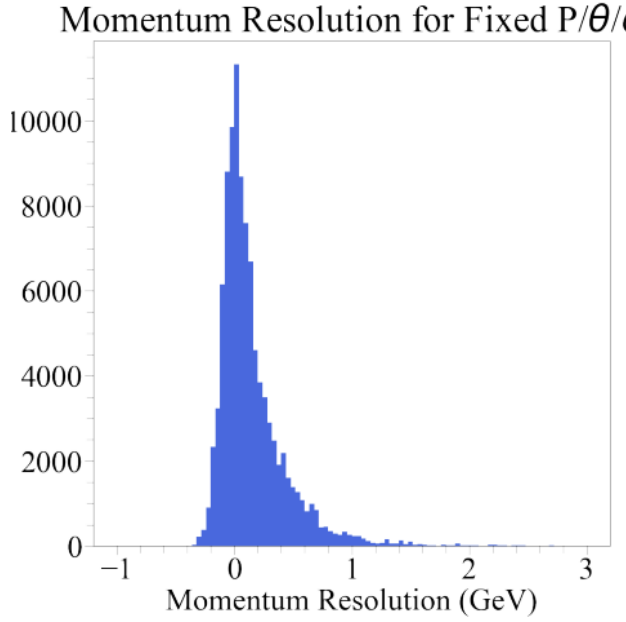
Train N decision trees with different random inputs, ie N different predictions. Pick DT at random during prediction time.



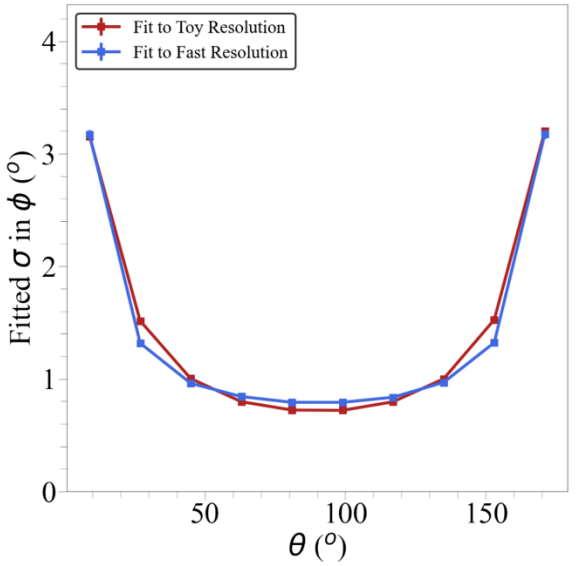
No Random Input, 1 DT



5 Random Input, 1 DT



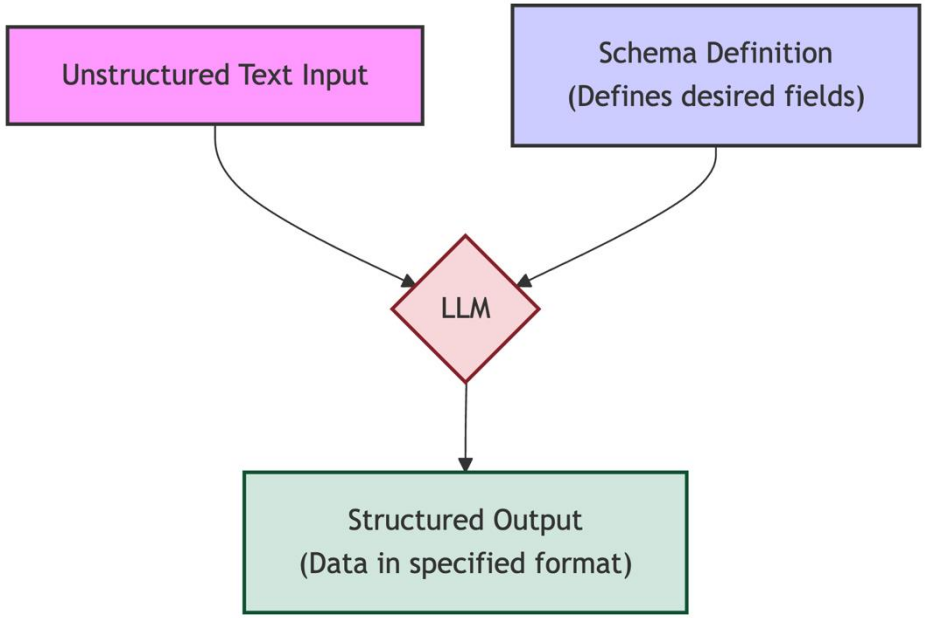
5 Random Input, 50 DTs



5 Random Input, 50 DTs

R. Brunk,
W. Phelps

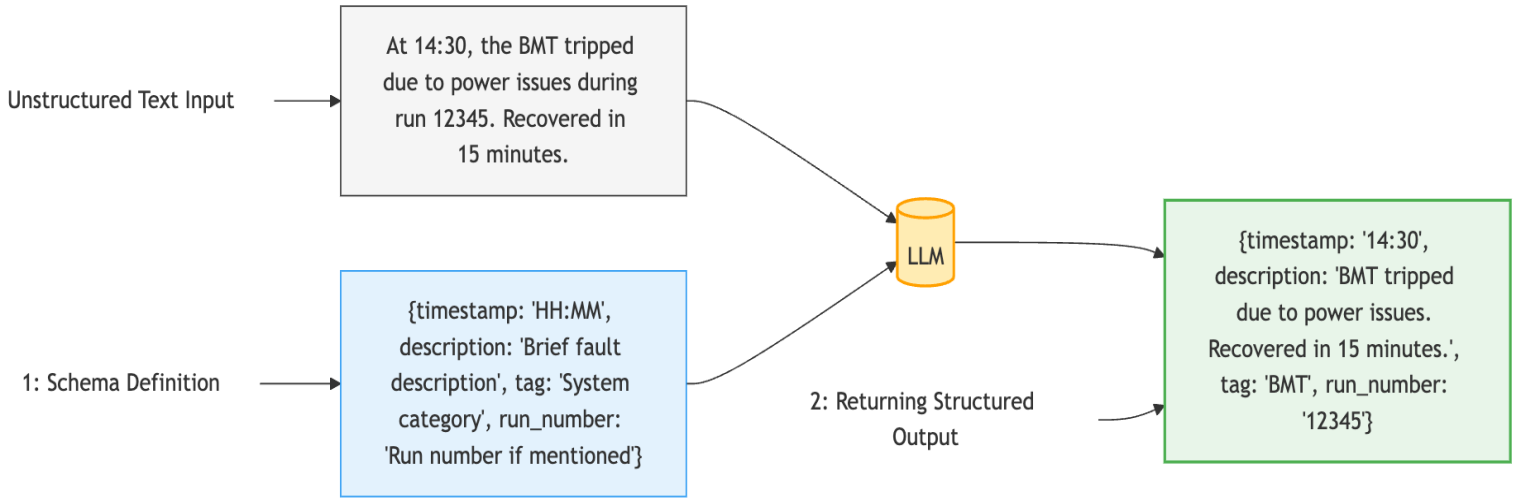
Structured Output



Aim is to parse unordered logbooks into an easy to parse dataset.

This can then be used to study eg recurring faults, why this happens.

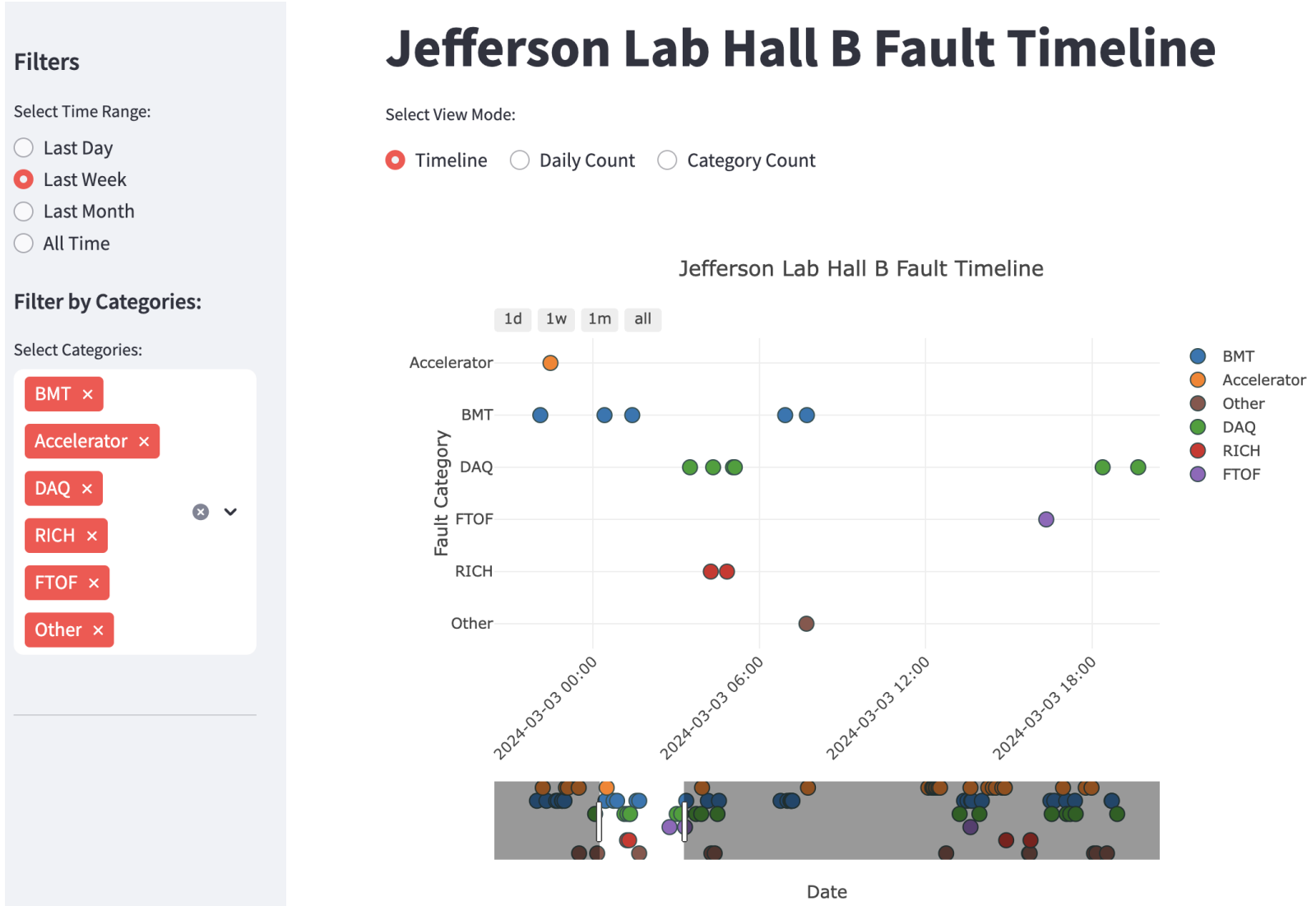
Helpful to create timelines for data quality assurance, train eg HYDRA etc…



R. Brunk,
W. Phelps

Structured Output

Work ongoing in JLab, lost track of it slightly…



RAGs

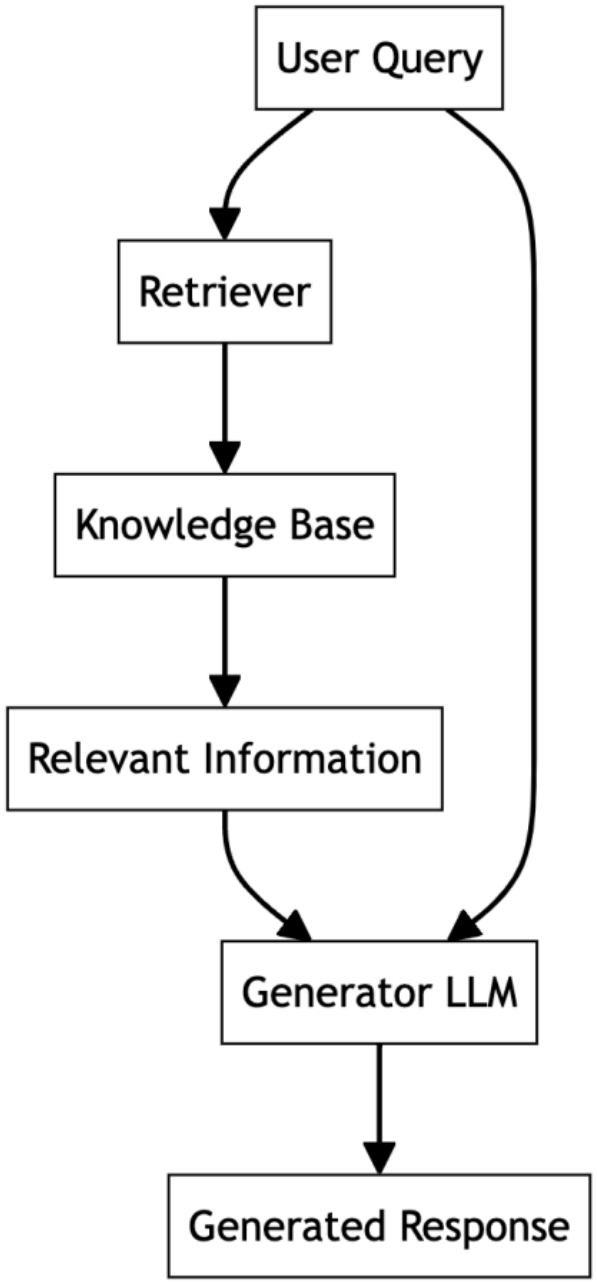
Retrieval Augmented Generation aims to take in a user query and:

- Parse this user query to LLM
- LLM searches a knowledge base for matching text
- LLM creates answer, with references

Several possible use cases at CLAS12:

- Shift taking – parse logbook and manuals to assist shift takers. Connect to alarm system
- Documentation, pass previous code bases, examples, articles, proposals etc…

RAG interface will be available, most of the work will be based on validation and understanding limitations (eg related to knowledge base size, diversity…)



A. Panta,
R. Brunk,
W. Phelps

M. Ouillon

General PID

Use as much info as possible (96 features):

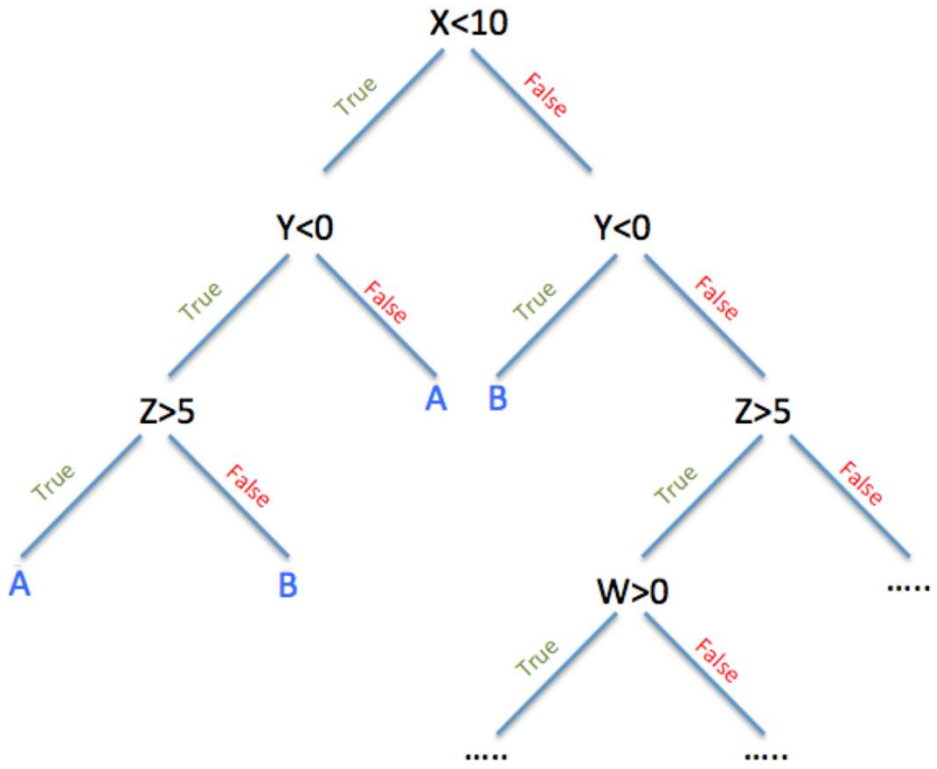
- Calorimeters
- Cherenkov counters,
- TOF
- Higher-level variables (px, py, pz, charge, …)

Neural networks tend to not deal well with variables that don't carry relevant or correlated info

Decision trees work well on these (just not used in decision making)

Boosting avoids overtraining:

- Train once, get prediction error on events
- Train again, error with wrong prediction have higher weight
- …



General PID

Use as much info as possible (96 features):

- Calorimeters
- Cherenkov counters,
- TOF
- Higher-level variables (px, py, pz, charge, …)

Neural networks tend to not deal well with variable that don't carry relevant or correlated info

Decision trees work well on these (just not used in decision making)

Boosting avoids overtraining:

- Train once, get prediction error on events
- Train again, error with wrong prediction have higher weight
- …

M. Ouillon

