Jefferson Lab
Thomas Jefferson National Accelerator Facility

Fast ML on FPGA for Particle Identification and Tracking

Sergey Furletov
(Jefferson Lab)

Streaming Readout Workshop SRO-XII

University of Tokyo ,1-4 Dec 2024

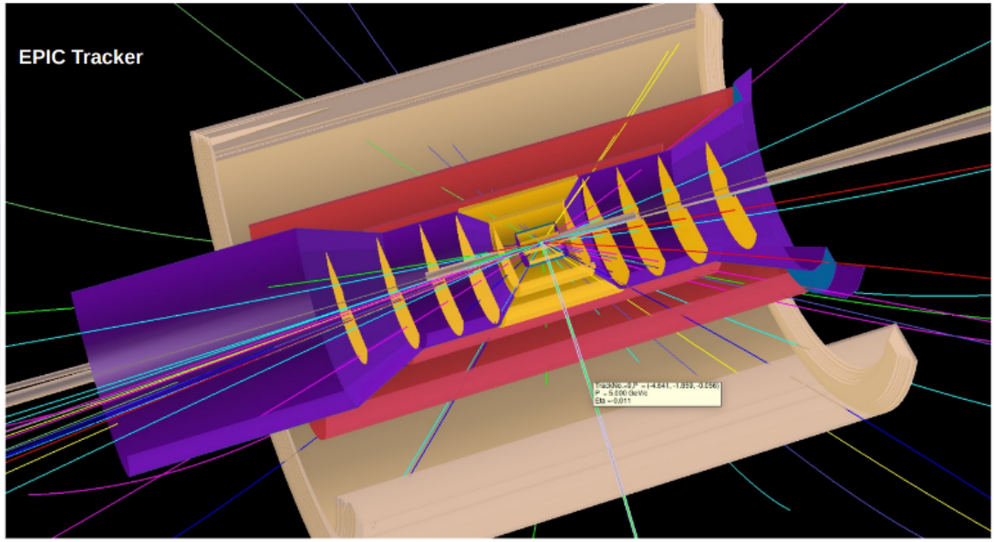
Outline

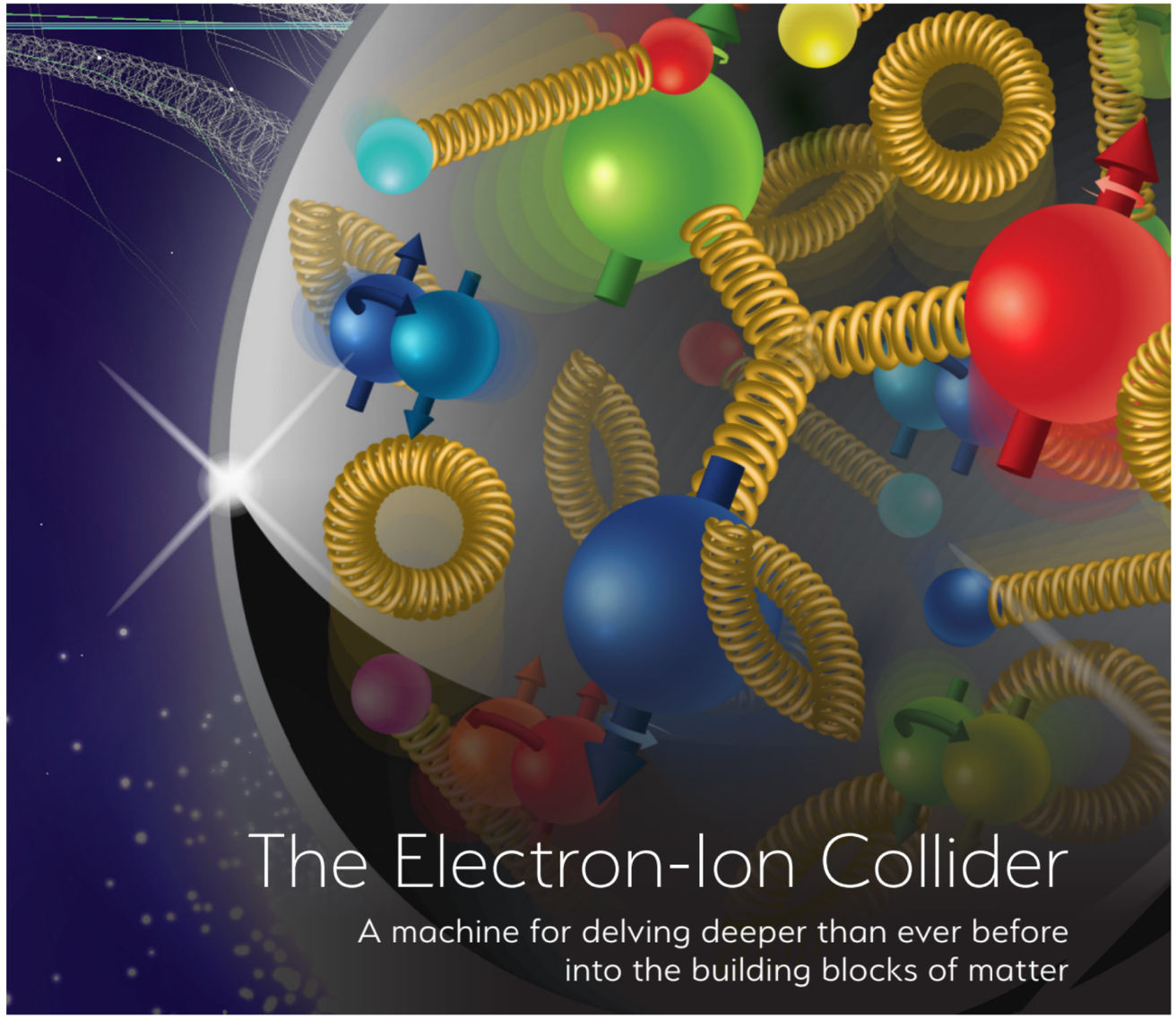
- *Report on **ML-FPGA** developments for 2 nuclear physics experiments.*
  - **EIC** - new Electron-Ion Collider under construction at BNL.
  - **GlueX** - experiment located at the Thomas Jefferson National Accelerator Facility (JLab)

Electron Ion Collider (EIC)

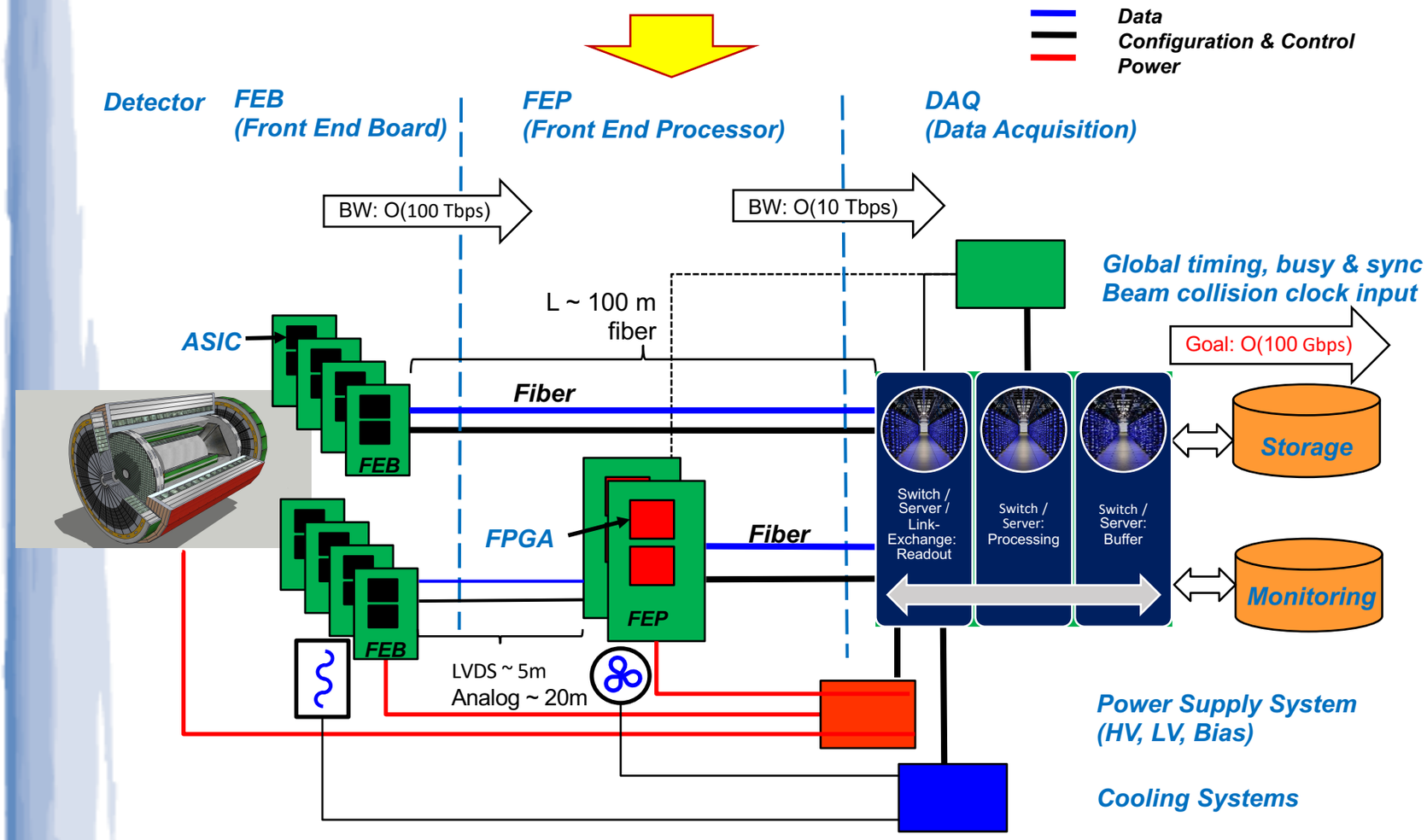
Jefferson Lab
Thomas Jefferson National Accelerator Facility

- The Electron-Ion Collider, a new facility for nuclear physics research to be located at Brookhaven Lab, will allow scientists from across the nation and around the globe to peer inside protons and atomic nuclei to reveal secrets of the strongest force in nature.
- Research at the EIC will take our understanding of matter to the next level—beyond the interactions of atomic nuclei with their orbiting electrons, which power the electronic and information technologies we now use every day, to the forces acting inside the nucleus.





EIC streaming readout as motivation for ML-FPGA



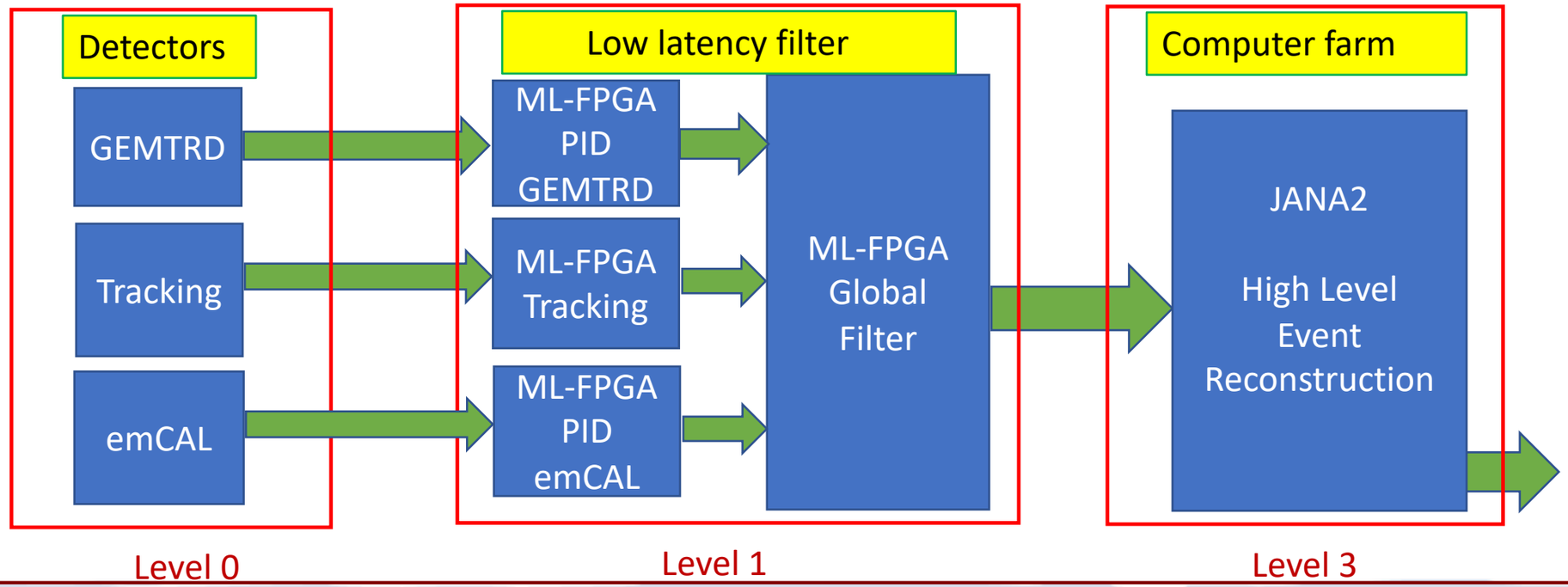
- *The correct location for the ML on the FPGA filter is called "FEP" in this figure.*
- *This gives us a chance to reduce traffic earlier.*
- *Allows us to touch physics: ML brings intelligence to L1.*
- *However, it is now unclear how far we can go with physics at the FPGA.*
- *Initially, we can start in pass-through mode.*
- *Then we can add background rejection.*
- *Later we can add filtering processes with the largest cross section.*
- *In case of problems with output traffic, we can add a selector for low cross section processes.*
- *The ML-on-FPGA solution complements the purely computer-based solution and mitigates DAQ performance risks.*

Generic EIC R&D project RD15, ML-(on)-FPGA

- *The goal is to build a demonstrator that can operate under beam test conditions in real-time.*
- *The setup consists of several PID and tracking detectors: emCAL, GEMTRD, GEM tracker.*
- *Preprocessed data from detectors including decision on the particle type will be transferred to another ML-FPGA board with neural network for global PID decision.*
- *The global filter transfers data to off-line computer farm, running JANA2 software.*

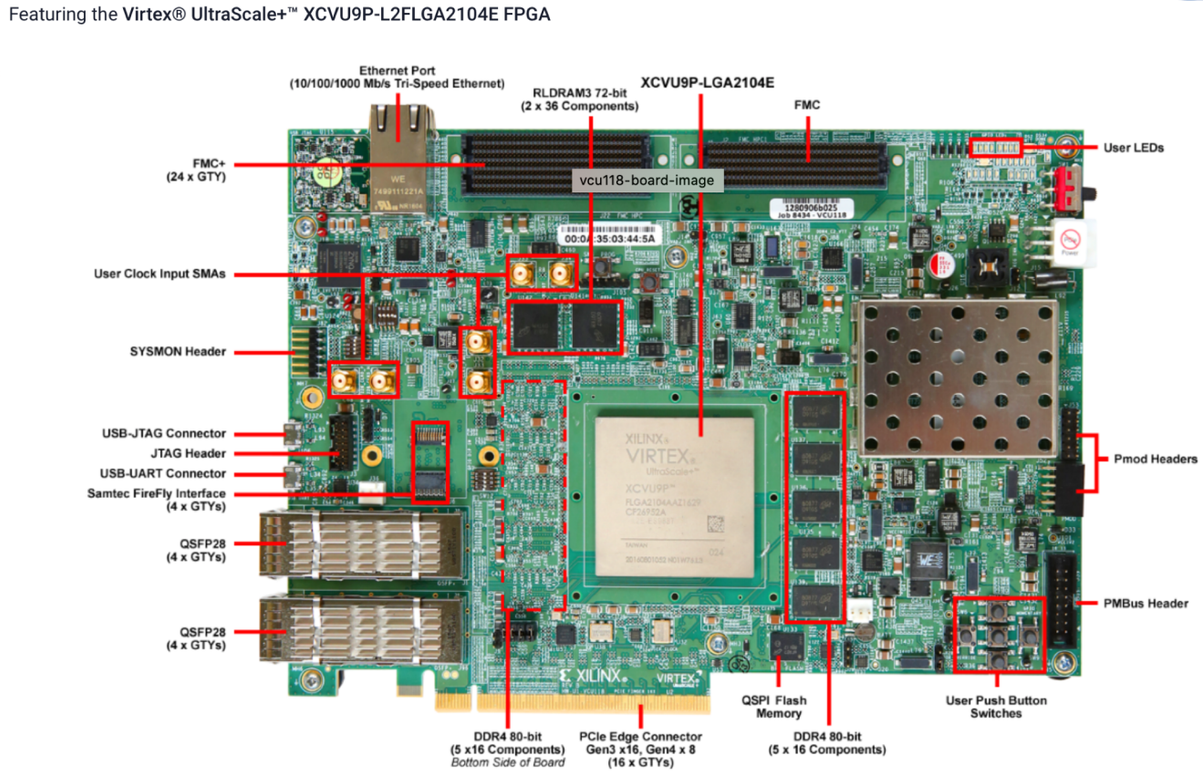
Team :

F. Barbosa, L. Belfore, N. Branson, N. Brei, C. Dickover, C. Fanelli, D. Furletov, L. Jokhovets, D. Lawrence, C. Mei, D. Romanov, K. Shivu



FPGA test board for ML

- At an early stage in this project, as hardware to test ML algorithms on FPGA , we use a standard Xilinx evaluation boards rather than developing a customized FPGA board. These boards have functions and interfaces sufficient for proof of principle of ML-FPGA.
- The Xilinx evaluation board includes the Xilinx XCVU9P and 6,840 DSP slices. Each includes a hardwired optimized multiply unit and collectively offers a peak theoretical performance in excess of 1 Tera multiplications per second.
- Second, the internal organization can be optimized to the specific computational problem. The internal data processing architecture can support deep computational pipelines offering high throughputs.
- Third, the FPGA supports high speed I/O interfaces including Ethernet and 180 high speed transceivers that can operate in excess of 30 Gbps.

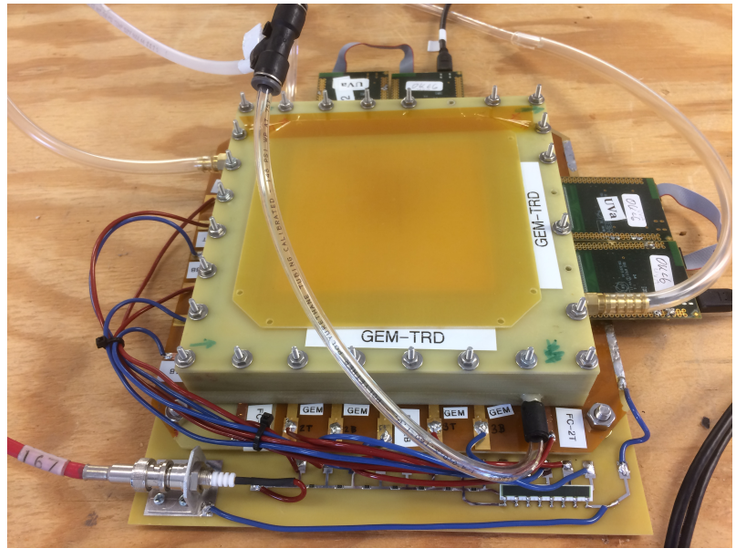


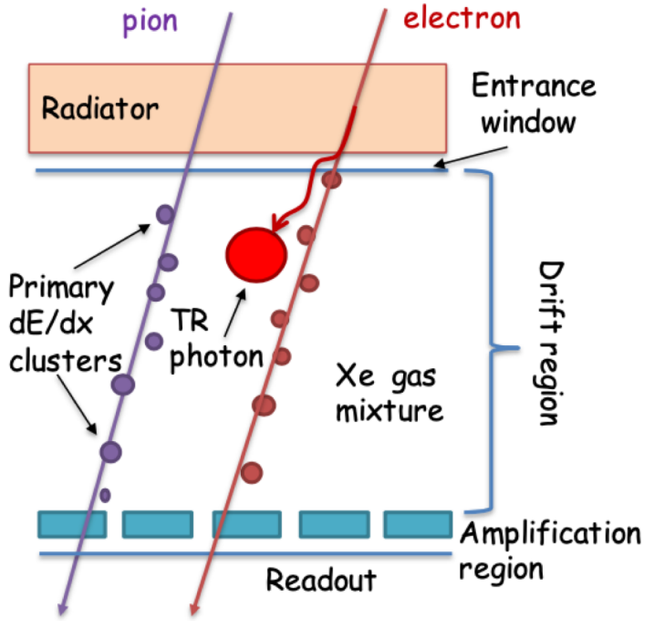
Xilinx Virtex® UltraScale+™

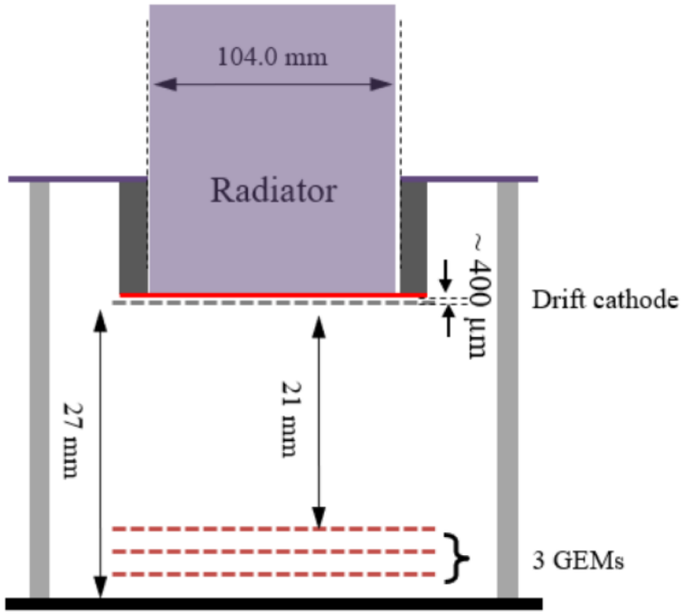
GEM-TRD prototype for EIC R&D

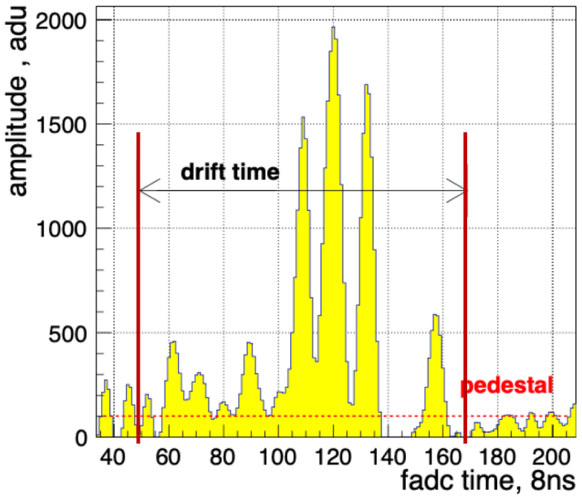
- To demonstrate the operating principle of the ML FPGA, we use the existing setup
- from the EIC detector R&D project
- A test module was built at the University of Virginia
- The prototype of GEMTRD/T module has a size of 10 cm × 10 cm with a corresponding to a total of 512 channels for X/Y coordinates.
- The readout is based on flash ADC system developed at JLAB (fADC125) @125 MHz sampling.

- GEM-TRD provides e/hadron separation and tracking





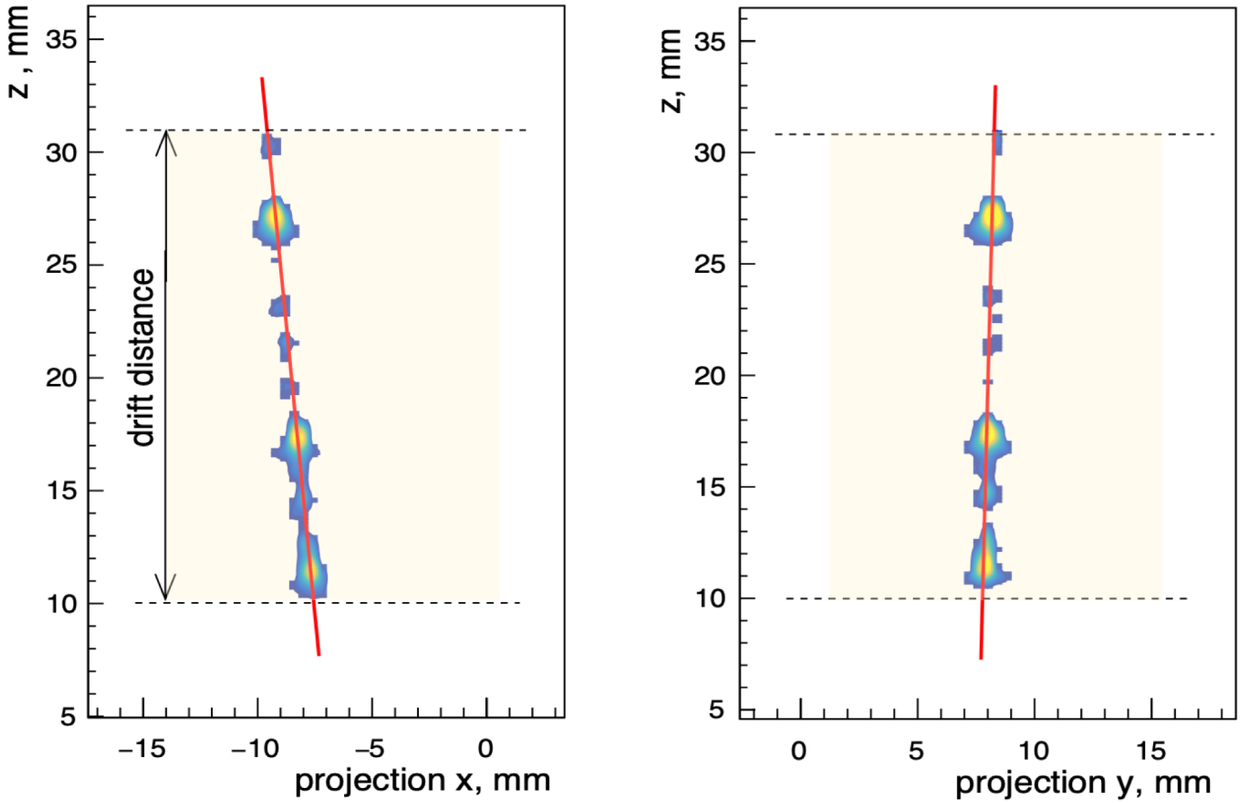




GEM-TRD principle

- *The e/pion separation in the GEM-TRD detector is based on counting the ionization along the particle track.*
- *For electrons, the ionization is higher due to the absorption of transition radiation photons*
- *So, particle identification with TRD consists of several steps:*
  - The first step is to cluster the incoming signals and create "hits".
  - The next is "pattern recognition" - sorting hits by track.
  - Finding a track
  - Ionization measurement along a track
  - As a bonus, TRD will provide a track segment for the global tracking system.

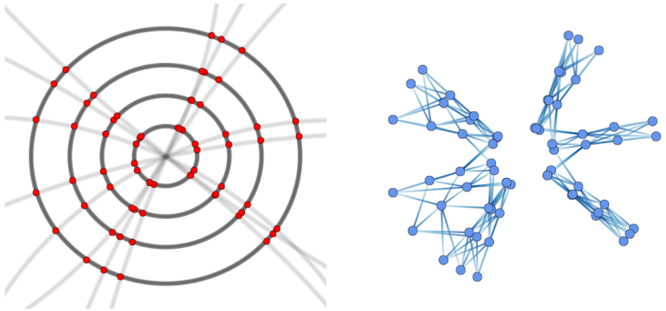
GEM-TRD can work as micro TPC, providing 3D track segments



GEMTRD tracks

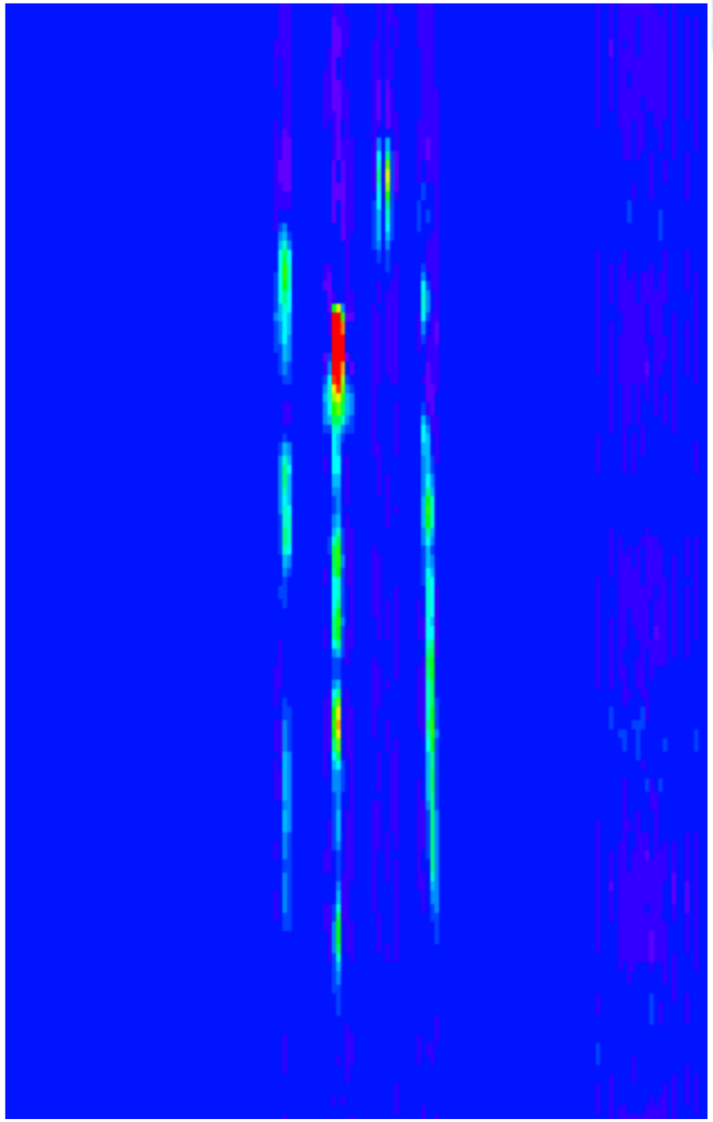
Jefferson Lab
Thomas Jefferson National Accelerator Facility

- *In a real experiment, GEMTRD will have multiple tracks.*
- *So we also need a fast algorithm for pattern recognition*
- *As well as for track fitting.*
- *The decision was made to try the Graph Neural Network (GNN) for pattern recognition.*
- *And a recurrent neural network – LSTM, for track fitting.*



Javier Duarte
arXiv:2012.01249v2 [hep-ph] 7 Dec 2020

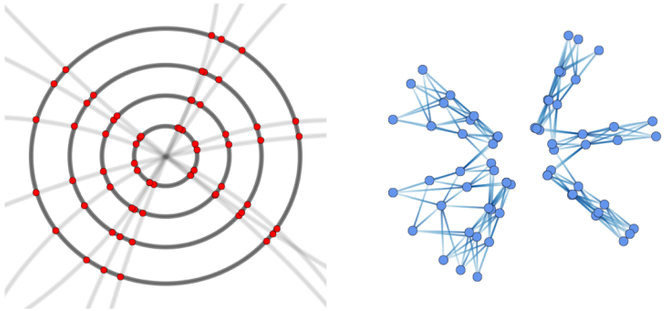
- HEP advanced tracking algorithms at the exascale **(Project Exa.TrkX)**
- https://exatrkx.github.io/



GEMTRD tracks

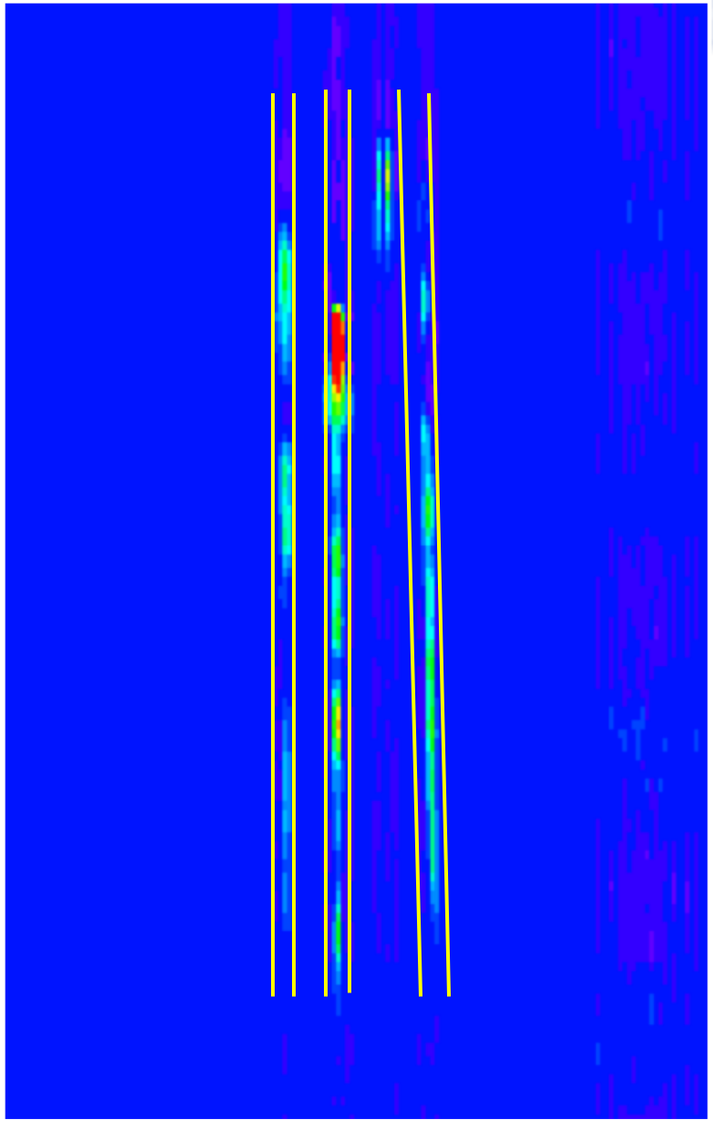
Jefferson Lab
Thomas Jefferson National Accelerator Facility

- *In a real experiment, GEMTRD will have multiple tracks.*
- *So we also need a fast algorithm for pattern recognition*
- *As well as for track fitting.*
- *The decision was made to try the Graph Neural Network (GNN) for pattern recognition.*
- *And a recurrent neural network – LSTM, for track fitting.*
- *PID is based on measuring ionization along the track.*



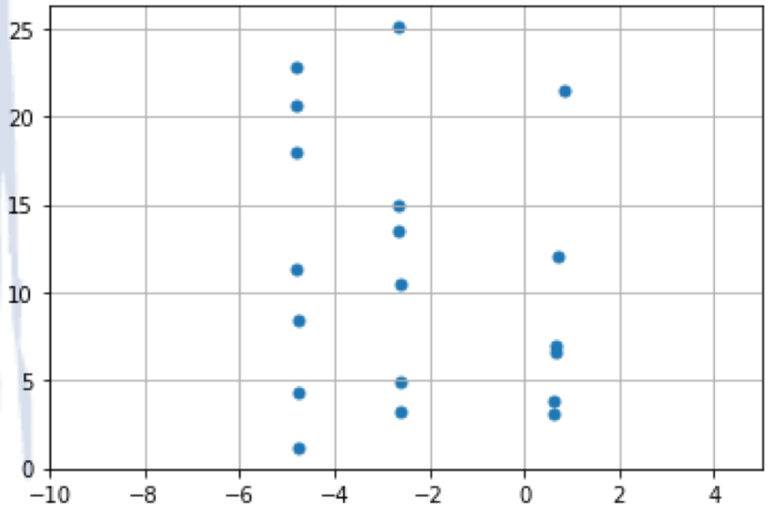
Javier Duarte
arXiv:2012.01249v2 [hep-ph] 7 Dec 2020

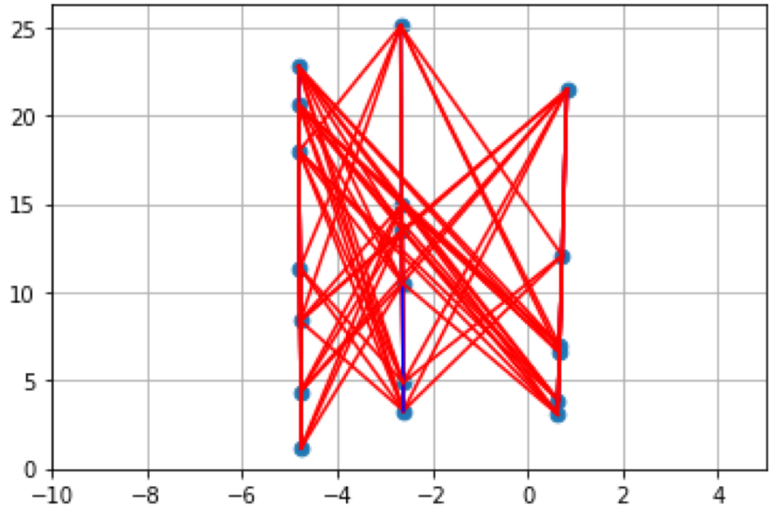
- HEP advanced tracking algorithms at the exascale **(Project Exa.TrkX)**
- https://exatrkx.github.io/

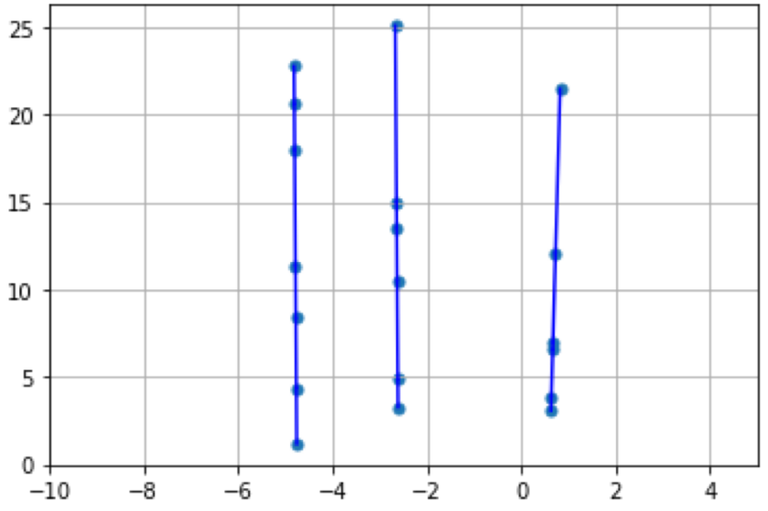


GNN for pattern recognition

- *Graph Neural Networks (GNNs) designed for the tasks of hit classification and segment classification.*
  - These models read a graph of connected hits and compute features on the nodes and edges.
- *The input and output of GNN is a graph with a number of features for nodes and edges.*
  - In our case we use the edge classification
- *A complete graph on N vertices contains N(N - 1)/2 edges.*
  - This will require a lot of resources which are limited in FPGA.
- *To keep resources under control, we can construct the graph for a specific geometry and limit the minimum particle momentum.*
- *In our case we have a straight track segments, with a quite narrow angular distribution ~15 degree.*
- *Thus, for the input hits (left), we connect only those edges that satisfy our geometry and the momentum of most tracks (middle)*
- *The trained GNN processes the input graph and sets the probability for each edge as output.*
- *The right plot shows edges with a probability greater than 0.7*

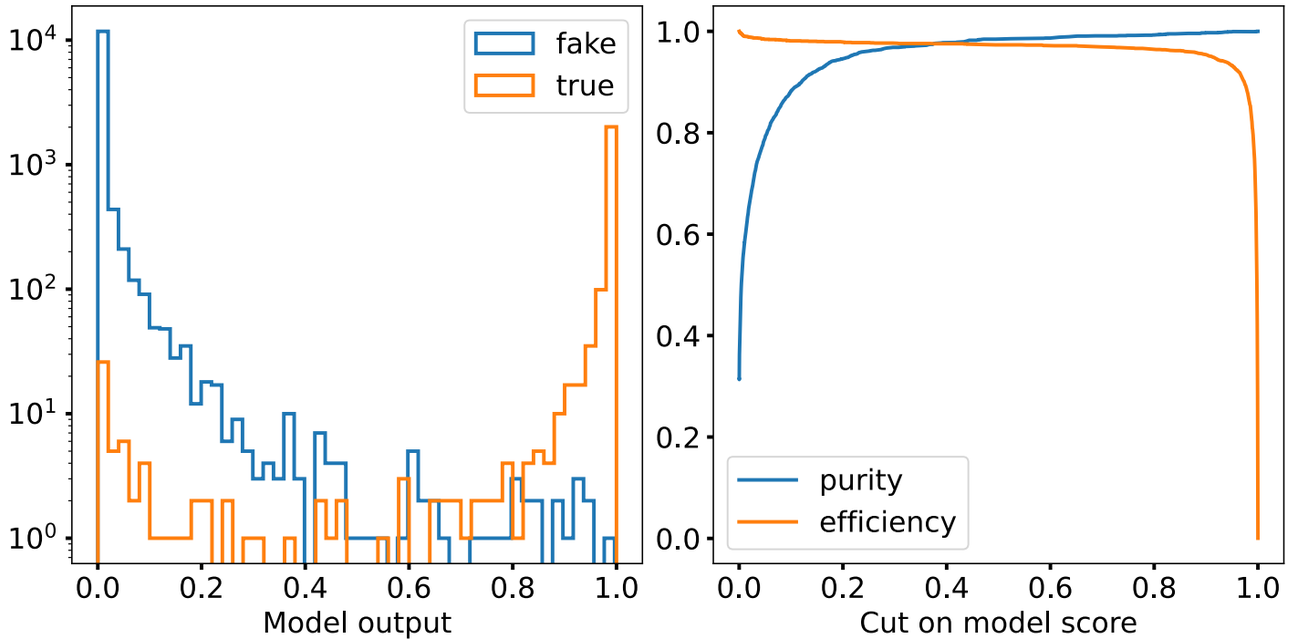






GNN performance

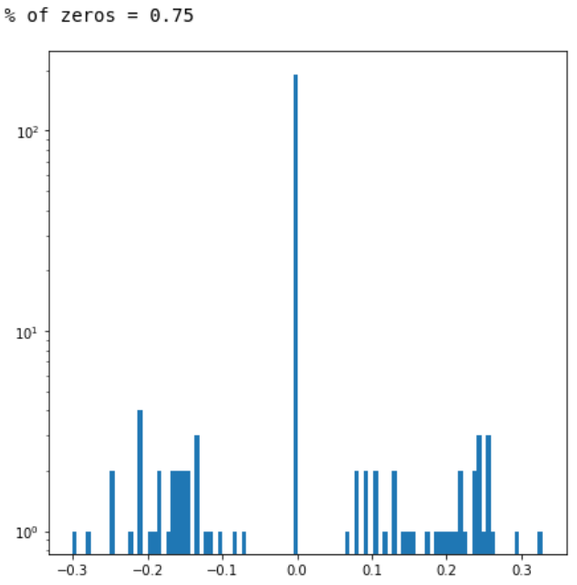
- *This type of graph neural network is not yet supported in HLS4ML.*
- *So we did a manual conversion first to C++ and then to Verilog using Vitis_HLS.*
- *This neural network has not been optimized/pruned, so it consumes a lot of resources - 70% of DSPs, (4651 of 6840).*
  - Network use precision ap fixed < 16,9 >
  - At the moment it can serve up to 21 hits and 42 edges, or , in our case (GEM-TRD), it will be 3-5 tracks.
- *However, it performs all calculations in ~3 µs (left plot) (thanks to Ben Raydo), providing good purity and efficiency (right plot).*

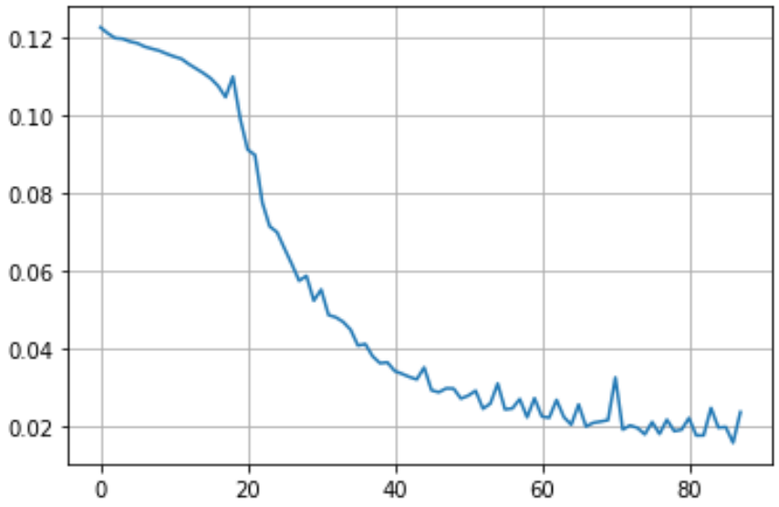


Modules & Loops	Issue Type	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
gnn2dfs2		-	589	2.945E3	-	590	-	no	42	4424	394036	2519454	0
toGraph		-	499	2.495E3	-	497	-	dataflow	42	4424	391308	2515320	0
fromGraph		-	331	1.655E3	-	1	-	yes	0	0	197686	1673583	0
gnn2dfs_loc_1		-	496	2.480E3	-	496	-	no	42	4422	172620	785082	0
toGraph_Block_split100_proc205		-	480	2.400E3	-	480	-	no	0	2	7226	49627	0
VITIS_LOOP_1365_1		-	63	315.000	3	-	21	no	-	-	-	-	-
VITIS_LOOP_1400_3		-	22	110.000	3	1	21	yes	-	-	-	-	-

RNN/LSTM for track fit

- *The hits sorted by tracks from the pattern recognition GNN are fed into another neural network trained to fit the tracks.*
- *We use RNN/LSTM neural networks. (thanks to Dylan Rankin for help)*
  - Network use precision ap fixed < 24,11 >
  - The input layer is designed for 26 hits.
  - The work on optimization of NN is ongoing.
- *The LSTM network after pruning consumes 19% of the DSP resources and has a latency of 1 µs.*





+ Latency (clock cycles):
    * Summary:

Latency		Interval		Pipeline
min	max	min	max	Type
213	213	208	208	function

== Utilization Estimates

* Summary:

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6	-
FIFO	-	-	-	-	-
Instance	64	4271	23258	163672	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	955	-
Register	-	-	2323	-	-
Total	64	4271	25581	164633	0
Available SLR	1440	2280	788160	394080	320
Utilization SLR (%)	4	187	3	41	0
Available	4320	6840	2364480	1182240	960
Utilization (%)	1	62	1	13	0

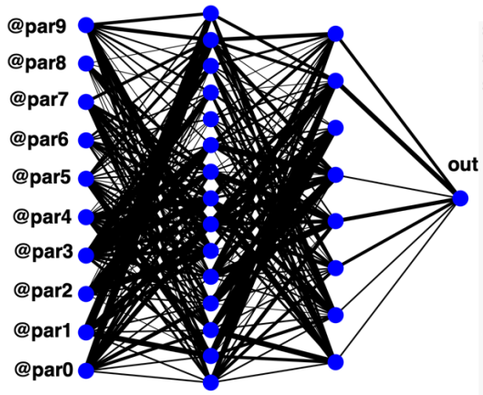
== Utilization Estimates

* Summary:

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6	-
FIFO	-	-	-	-	-
Instance	64	1308	12199	53194	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	955	-
Register	-	-	2147	-	-
Total	64	1308	14346	54155	0
Available SLR	1440	2280	788160	394080	320
Utilization SLR (%)	4	57	1	13	0
Available	4320	6840	2364480	1182240	960
Utilization (%)	1	19	~0	4	0

MLP neural network for PID

- *After the track is fit, the ionization along the track can be counted.*
- *The distance along the track is divided into 10-20 bins, and the ionization energy in these bins is fed to the input of the MLP neural network.*
- *Typically neural network weights often have many zeros, thus, it is possible to reduce the size of the network by removing weights close to zero (~50%)*
- *The network performance near the working value of 90% efficiency.*

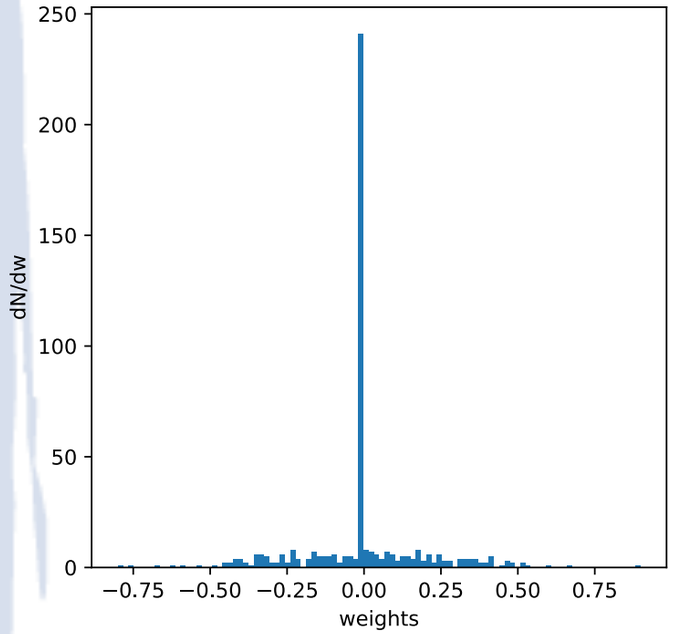


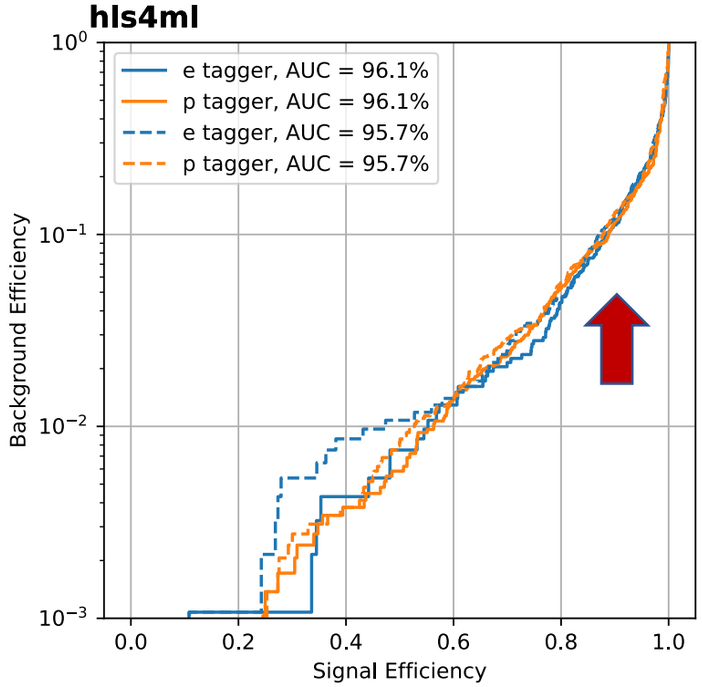

```
================================================================
== Performance Estimates
================================================================
+ Timing (ns):
    * Summary:
    +--------+-------+----------+------------+
    |  Clock | Target| Estimated| Uncertainty|
    +--------+-------+----------+------------+
    |ap_clk  |   5.00|     3.968|        0.62|
    +--------+-------+----------+------------+

+ Latency (clock cycles):
    * Summary:
    +-----+-----+-----+-----+----------+
    |  Latency  |  Interval | Pipeline |
    | min | max | min | max |   Type   |
    +-----+-----+-----+-----+----------+
    |   13|   13|    1|    1| function |
    +-----+-----+-----+-----+----------+

================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+--------+-----+
|       Name      | BRAM_18K| DSP48E|   FF   |  LUT   |URAM |
+-----------------+---------+-------+--------+--------+-----+
|DSP              |        -|      -|       -|       -|    -|
|Expression       |        -|      -|       0|       6|    -|
|FIFO             |        -|      -|       -|       -|    -|
|Instance         |       16|    233|    1241|   11742|    -|
|Memory           |        -|      -|       -|       -|    -|
|Multiplexer      |        -|      -|       -|      36|    -|
|Register         |        -|      -|    1235|       -|    -|
+-----------------+---------+-------+--------+--------+-----+
|Total            |       16|    233|    2476|   11784|    0|
+-----------------+---------+-------+--------+--------+-----+
|Utilization (%)  |      ~0 |      3|      ~0|      ~0|    0|
+-----------------+---------+-------+--------+--------+-----+
```

Latency = 65ns

II = 5ns

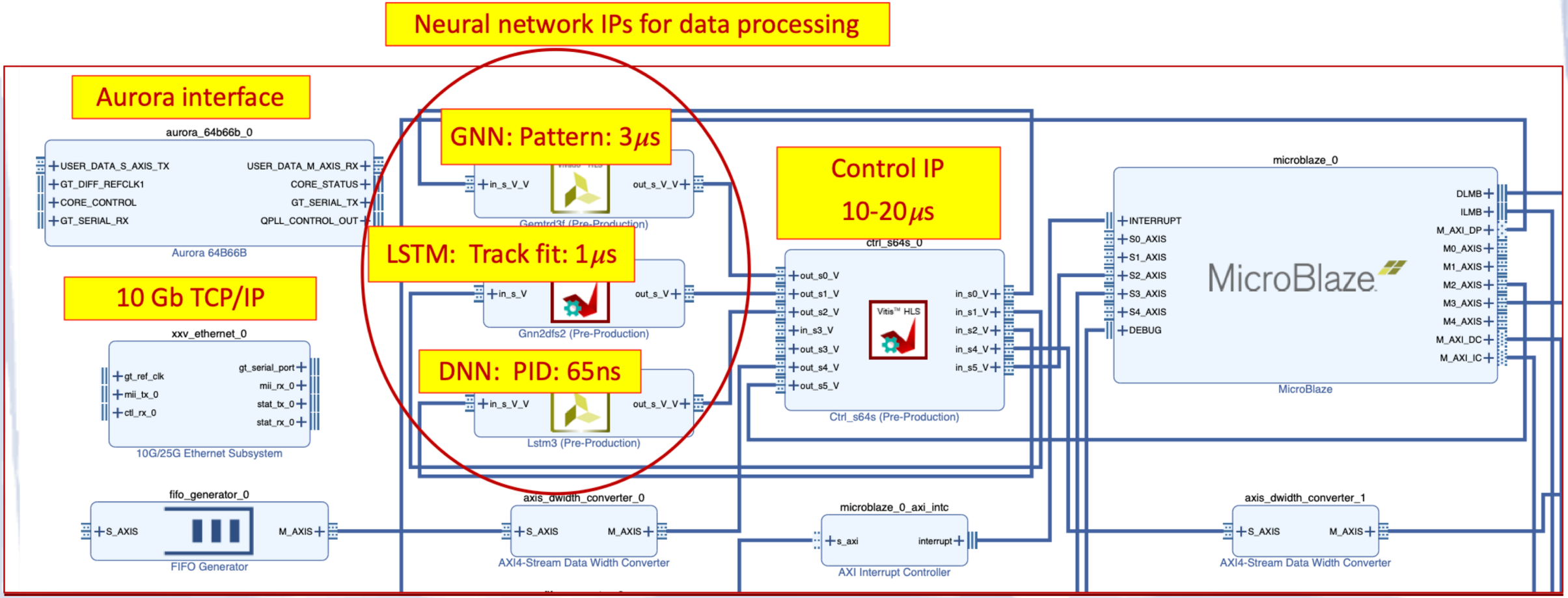
DSP utilization 3%





Board design

- *All data I/O operations are performed by Control IP*
- *MicroBlaze is only used to configure the board and monitor data processing.*
- *Aurora interface provides communication with a second FPGA board that processes the calorimeter data (CNN).*
- *10 Gigabit Ethernet uses TCP/IP, receives data from detectors (DAQ) and sends pre-processed data to the computer (farm).*



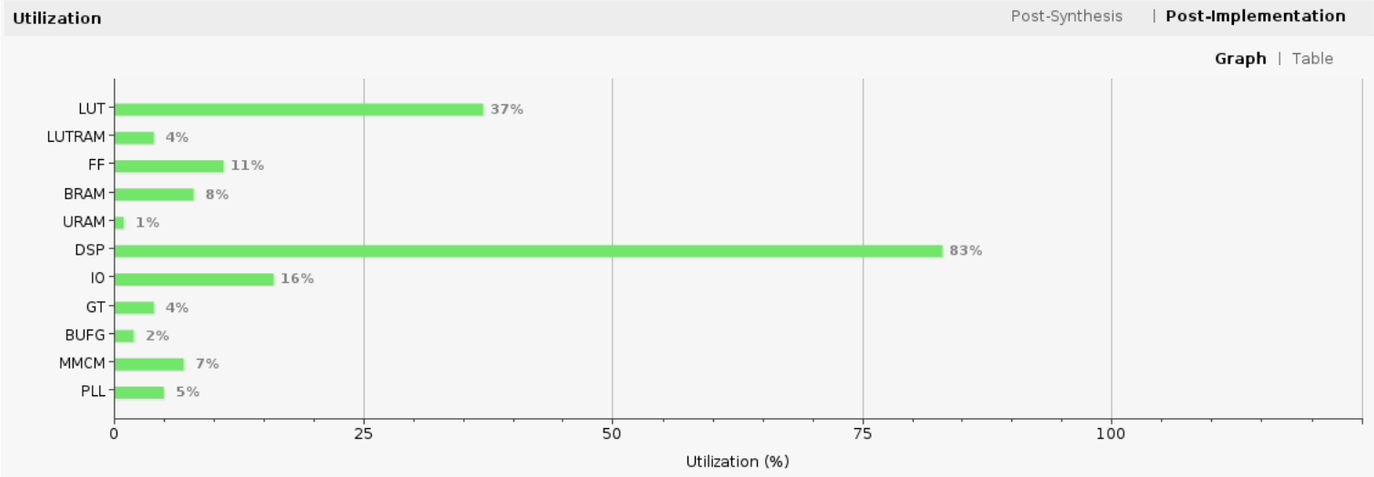
Latency and rates (very preliminary)

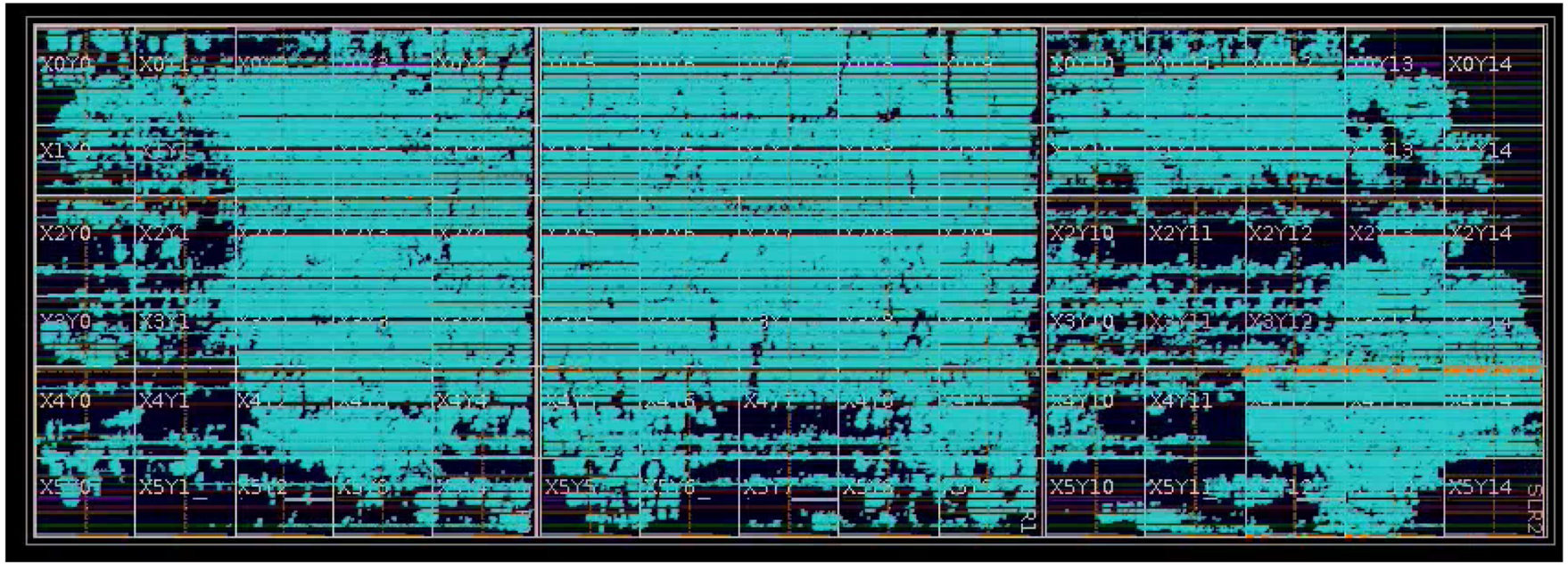
- *Control IP manages data traffic between NN-IP and the Ethernet interface.*
- *The IP block was synthesized directly using Vitis_HLS, the total latency is about ~20 µs (~50 kHz).*
- *Control IP block primarily performs serial I/O*
  - Therefore, it consists of long loops designed to accommodate the maximum data size.
- *In reality, the average data size is much smaller, so the actual speed should be higher.*
- *This was confirmed in measurements - peak performance reached 80 kHz.*
- *This is the first version, not yet optimized and II violations have not been fixed.*

Modules & Loops	Issue Type	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
ctrl_s64s	II Violation	-	4178	2.089E4	-	4179	-	no	8	5	4184	22984	0
VITIS_LOOP_399_2		-	4	20.000	1	1	4	yes	-	-	-	-	-
VITIS_LOOP_443_3		-	1024	5.120E3	1	1	1024	yes	-	-	-	-	-
VITIS_LOOP_464_4		-	1025	5.125E3	3	1	1024	yes	-	-	-	-	-
VITIS_LOOP_475_5	II Violation	-	45	225.000	6	2	21	yes	-	-	-	-	-
VITIS_LOOP_479_7	II Violation	-	43	215.000	4	2	21	yes	-	-	-	-	-
VITIS_LOOP_484_9_VITIS_LOOP_484_10		-	45	225.000	5	1	42	yes	-	-	-	-	-
VITIS_LOOP_503_11		-	7	35.000	5	1	4	yes	-	-	-	-	-
VITIS_LOOP_508_12		-	21	105.000	1	1	21	yes	-	-	-	-	-
VITIS_LOOP_523_13		-	27	135.000	3	1	26	yes	-	-	-	-	-
VITIS_LOOP_540_14		-	21	105.000	1	1	21	yes	-	-	-	-	-
VITIS_LOOP_542_15		-	22	110.000	3	1	21	yes	-	-	-	-	-
VITIS_LOOP_562_16	II Violation	-	804	4.020E3	45	40	20	yes	-	-	-	-	-
VITIS_LOOP_626_20		-	44	220.000	3	2	21	yes	-	-	-	-	-
VITIS_LOOP_642_21		-	1025	5.125E3	3	1	1024	yes	-	-	-	-	-

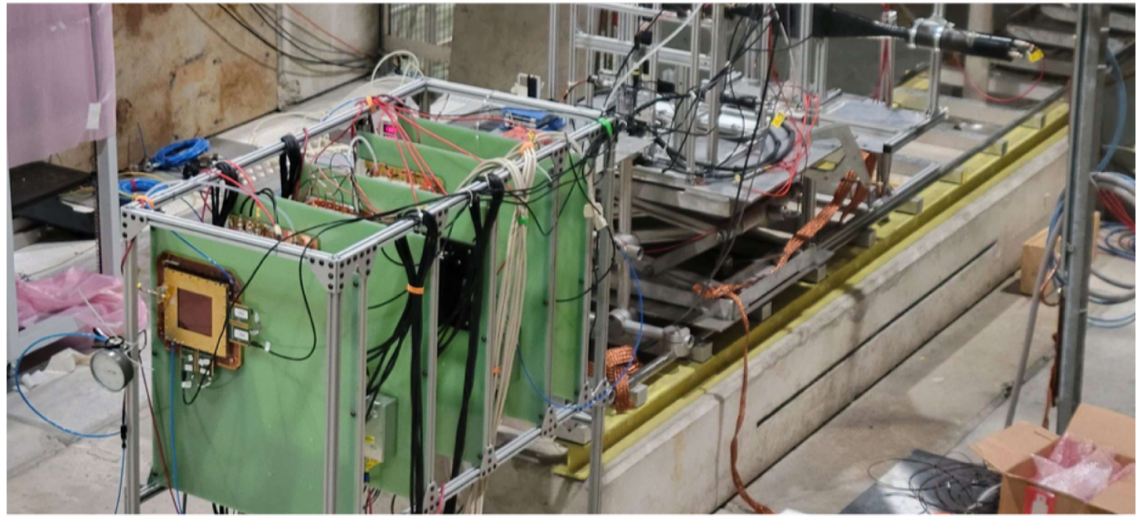
FPGA board resources for GEMTRD

- *Neural networks use a lot of FPGA resources.*
- *Therefore, one VCU118 board can only process data from GEMTRD.*

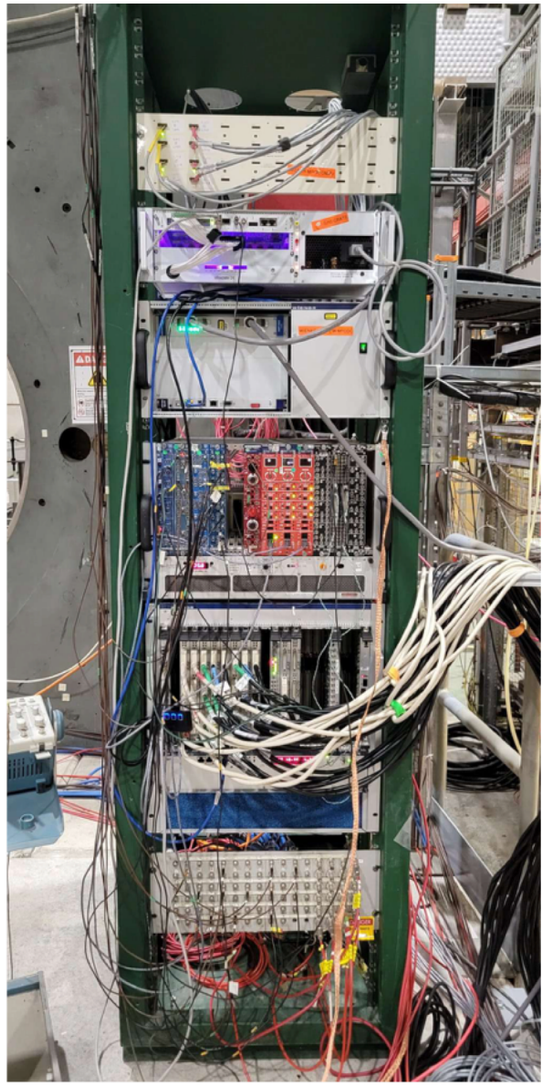




Test setup at CERN SPS/H8 beam line

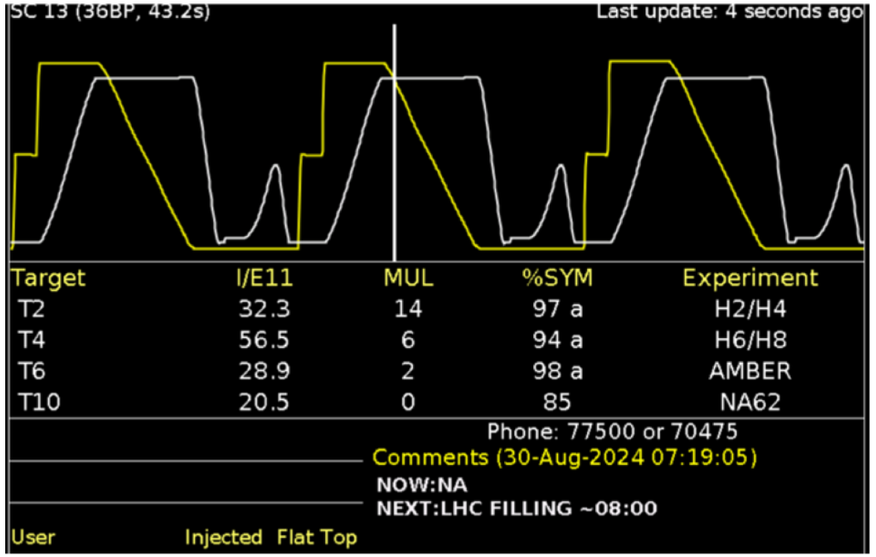


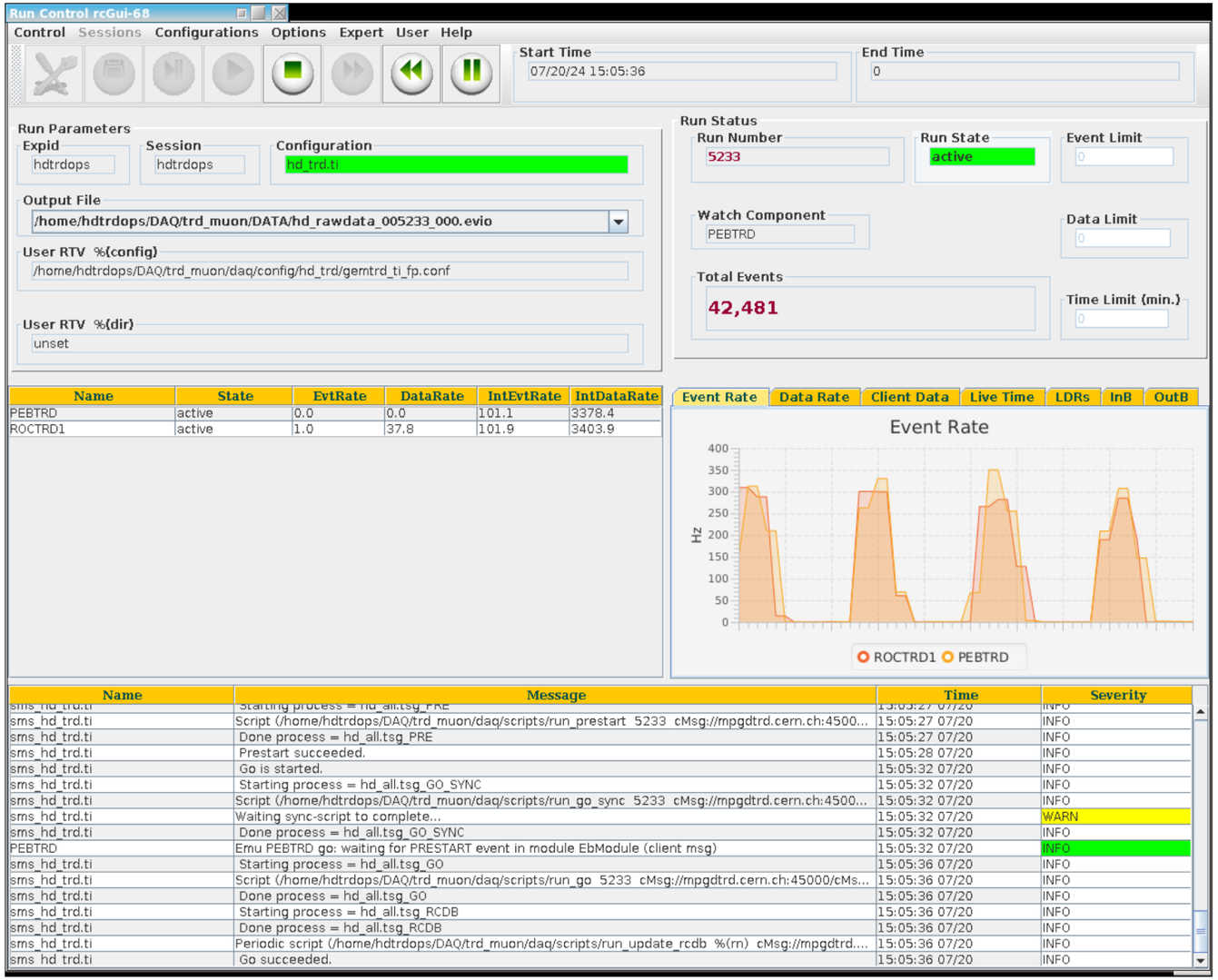
Detectors



Electronics rack

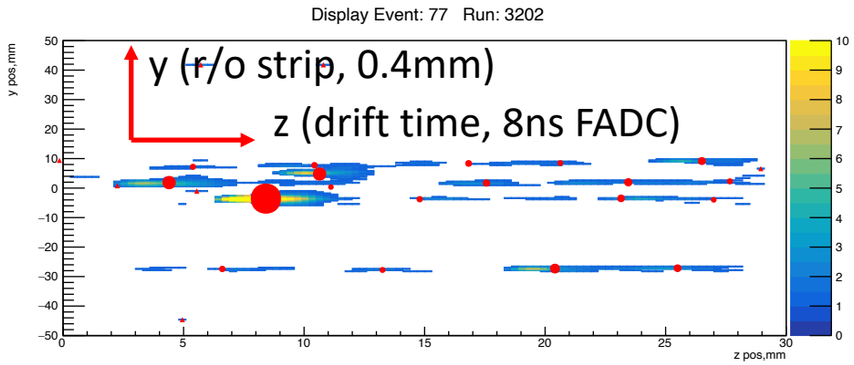
Beam structure and rate

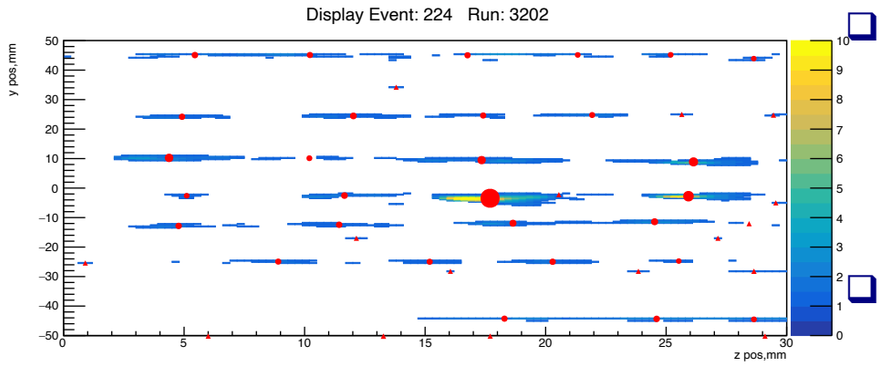


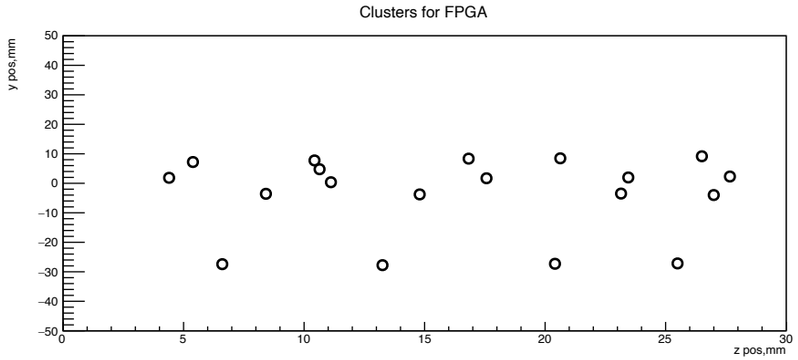


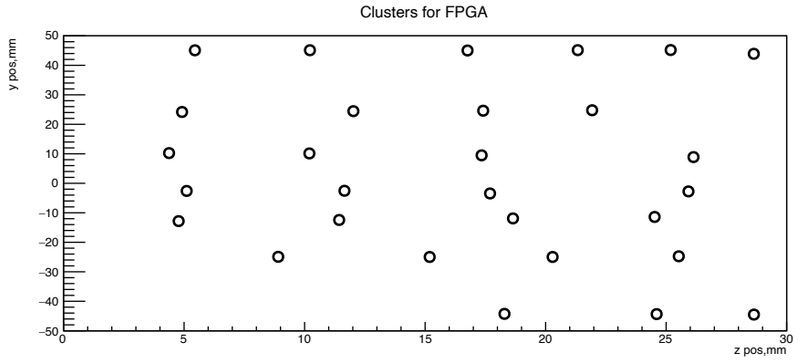
- *Spill Duration: 4.8 s.*
- *Repetition rate: 10 - 40 s.*
- *Energy: 20 GeV*
- *Trigger rate during spill: 300-400 Hz*

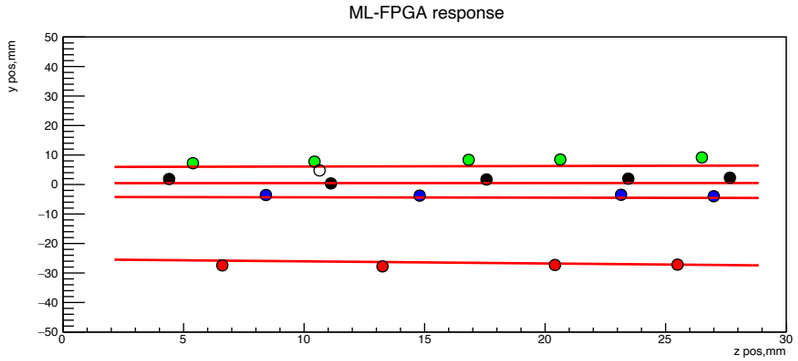
Tracking performance

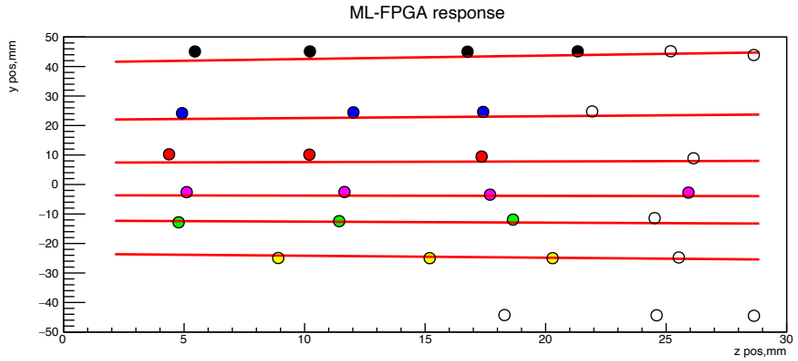












- *Top rows: show ionization along the track in GEMTRD detector.*
  - Red circles are reconstructed clusters using some dE/dx threshold. The size is proportional to energy.
- *Middle rows: after filtering out the noisy clusters, the coordinates of the clusters are sent to the FPGA/GNN for pattern recognition.*
- *Bottom rows: GNN provides labeling of clusters (by color in the figure), the same colors belong to the same track.*
- *Then clusters of the same color (tag) are sent to the track fitting module: LSTM.*
- *The results of track fitting are represented by lines in the figures.*
- *The next step is to count all the ionization in the corridor around the track and send it to the PID module (DNN).*
- *As a bonus, GEMTRD provides a track segment for the global tracking system.*

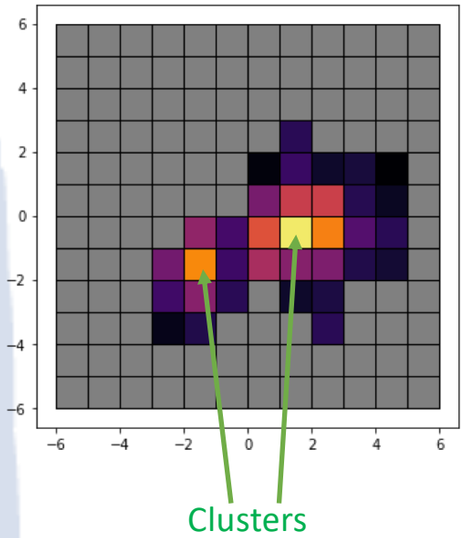
ML for Calorimeter

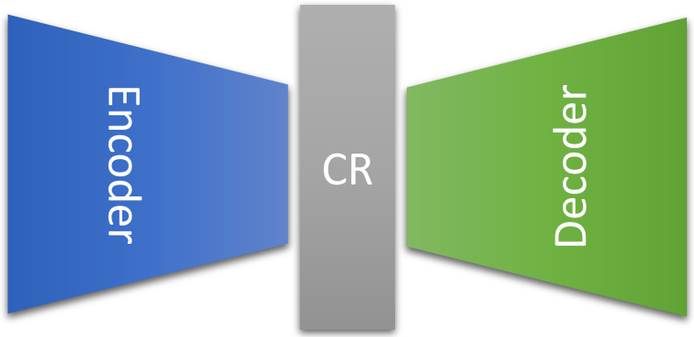
Calorimeter parameters reconstruction

Jefferson Lab
Thomas Jefferson National Accelerator Facility

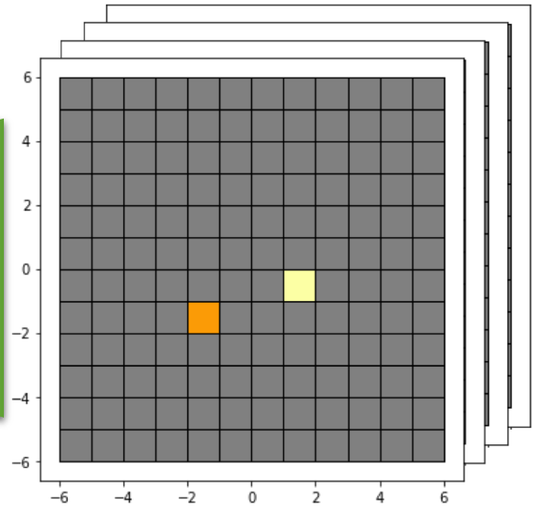
By Dmitry Romanov

Modules deposits





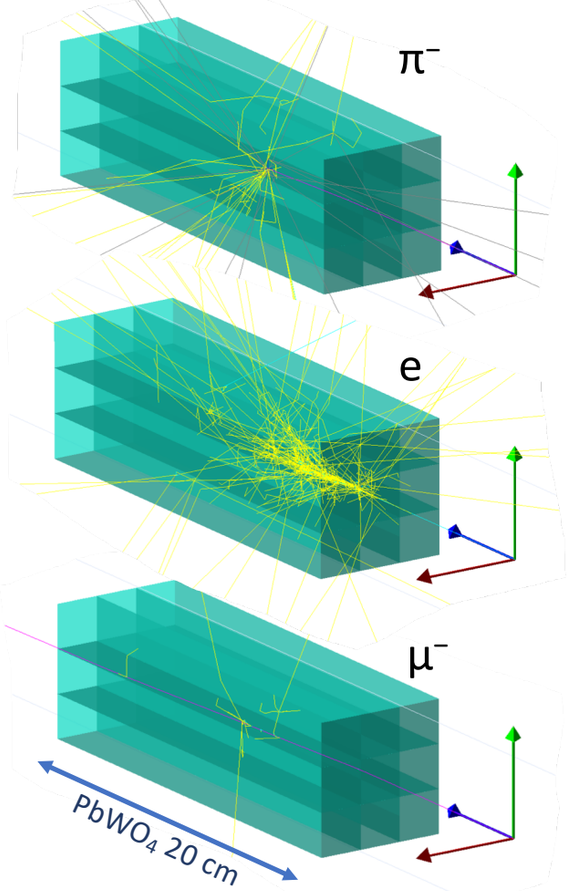
Convolutional variational autoencoder



Per cluster values: energy, pid, x, y, features

- Convolutional VAE as a backbone
- Modules deposits as inputs
- Per cluster output of multiple values:
- Energy, e/ π, coordinates, features

Geant 4 simulation



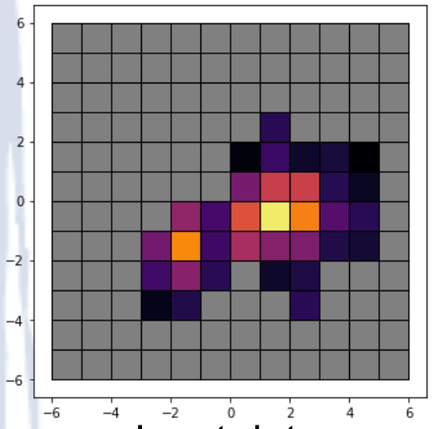
Examples of events with e and π^- showers and μ^- passing through.

CNN for calorimeter reconstruction

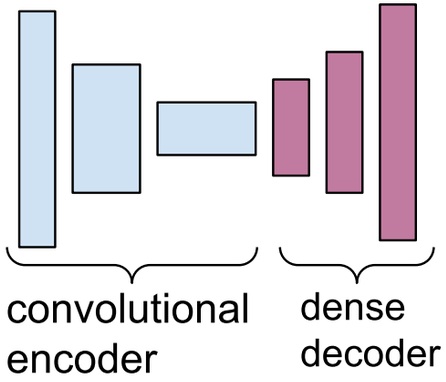
- *In this work we used a convolutional encoder with a decoder consisting of dense layers, which provide e-π separation scores as the output.*
- *Synthesized with HLS4ML, for calorimeter 11x11 cells.*
- *This was done to minimize a network size in FPGA and due to current limitation of HSL4ML of supported network layer types.*
- *FPGA synthesis with reuse factor of 1 has a latency of 0.7μs and an interval of 125 clocks. It uses 74% of DPS resources*
- *Network use precision ap fixed < 20,10 >*

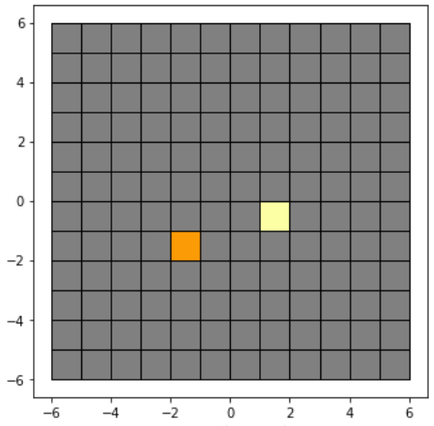
```
+--------+-------+----------+------------+
|  Clock | Target| Estimated| Uncertainty|
+--------+-------+----------+------------+
|ap_clk  |   5.00|     4.303|        0.62|
+--------+-------+----------+------------+

Latency (clock cycles):
  * Summary:
+-----+-----+-----+-----+----------+
|  Latency  |  Interval | Pipeline |
| min | max | min | max |   Type   |
+-----+-----+-----+-----+----------+
|  139|  139|  125|  125| dataflow |
+-----+-----+-----+-----+----------+
```

Actual values	Predicted results	
	e	π
e	98.8 %	1.2 %
π	2.9 %	97.1 %



Input data





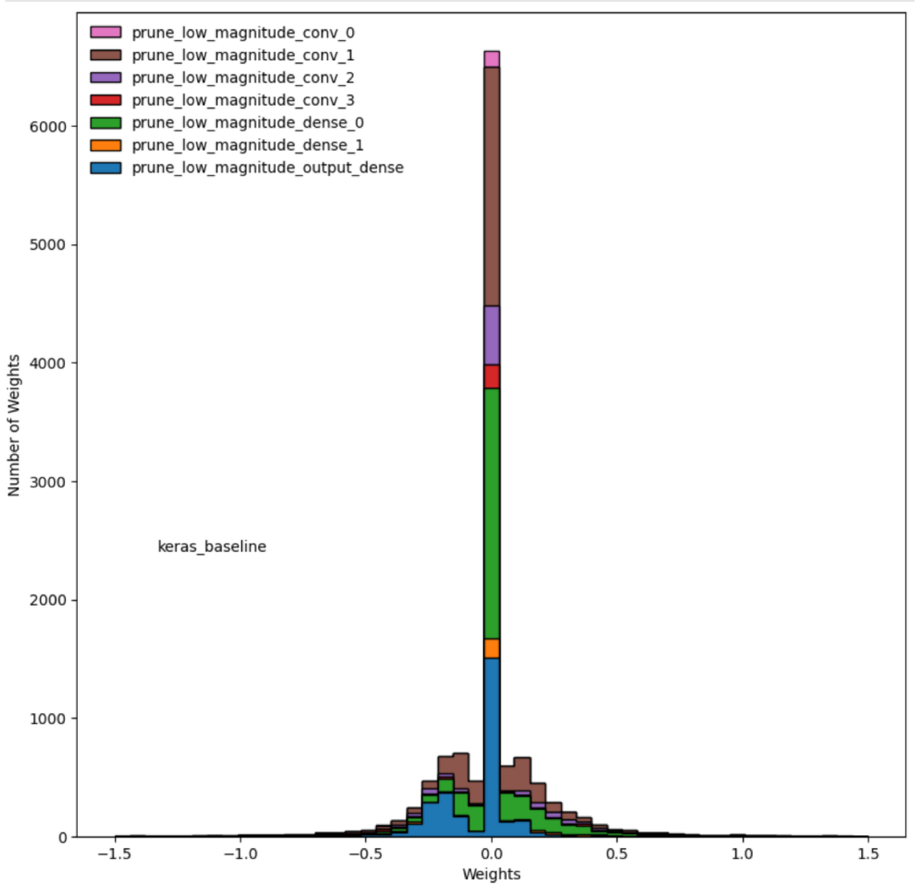
output

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	2	-
FIFO	404	-	8999	15698	-
Instance	61	5124	55854	243846	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	-	-
Register	-	-	-	-	-
Total	465	5124	64853	259546	0
Available SLR	1440	2280	788160	394080	320
Utilization SLR (%)	32	224	8	65	0
Available	4320	6840	2364480	1182240	960
Utilization (%)	10	74	2	21	0

Calorimeter CNN optimization with HLS4ML

```
hls_config['Model']['Precision'] = 'ap_fixed<20,10>'
```

```
Layer prune_low_magnitude_conv_0: % of zeros = 0.5
Layer prune_low_magnitude_conv_1: % of zeros = 0.5
Layer prune_low_magnitude_conv_2: % of zeros = 0.5
Layer prune_low_magnitude_conv_3: % of zeros = 0.5
Layer prune_low_magnitude_dense_0: % of zeros = 0.5
Layer prune_low_magnitude_dense_1: % of zeros = 0.5
Layer prune_low_magnitude_output_dense: % of zeros = 0.5
Layer prune_low_magnitude_fused_convbn_0: % of zeros = 0.0
Layer prune_low_magnitude_fused_convbn_1: % of zeros = 0.0
Layer prune_low_magnitude_fused_convbn_2: % of zeros = 0.0
Layer prune_low_magnitude_fused_convbn_3: % of zeros = 0.0
Layer prune_low_magnitude_dense_0: % of zeros = 0.0
Layer prune_low_magnitude_dense_1: % of zeros = 0.0
Layer output_dense: % of zeros = 0.0
```



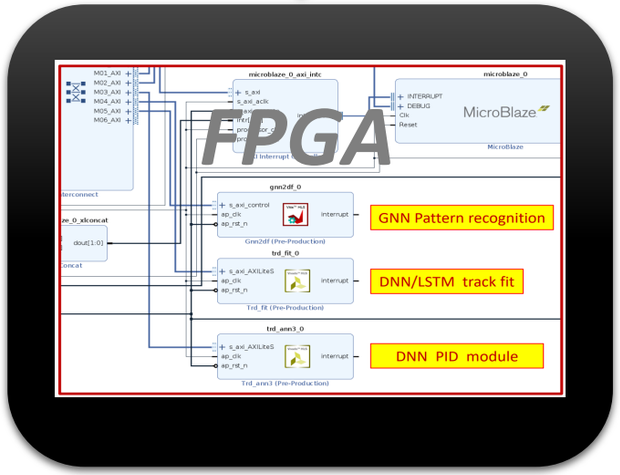
JANA2 for ML on FPGA

Pre-processed data from the FPGA is transferred over the network (TCP/IP) to a computer running JANA2 software.

JANA4ML4FPGA

Jefferson Lab
Thomas Jefferson National Accelerator Facility

Detector



Validation software

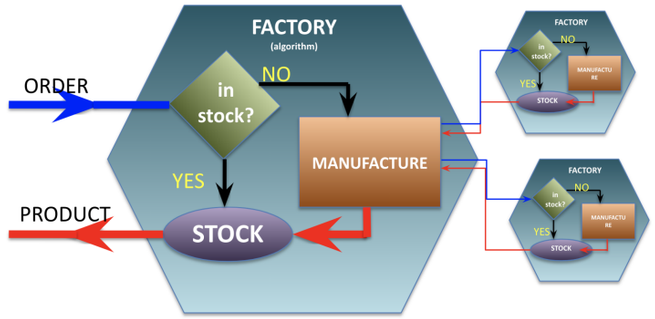
JANA2

(**J**Lab **ANA**lysis framework)

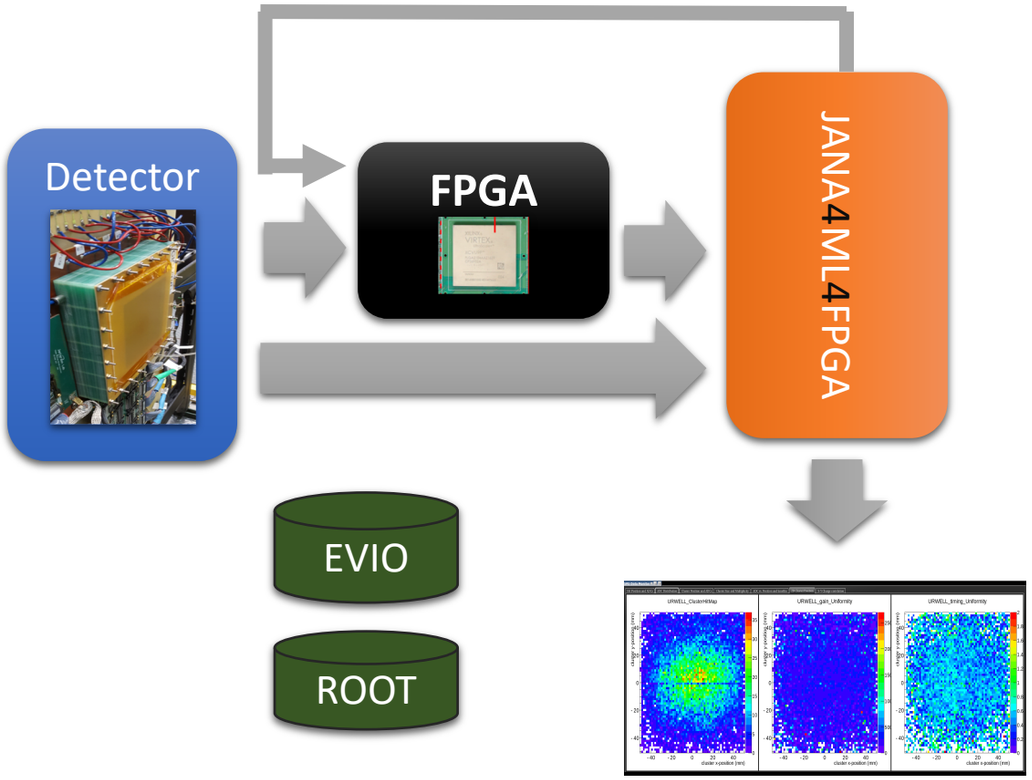
- JANA2 is a multi-threaded modular event reconstruction framework being developed at Jlab for online and offline processing

- JANA2 is a rewrite based on modern coding and CS practices. Developed for modern NP experiments with streaming readout, heterogeneous computing and AI

- JANA2 is the main framework chosen for EIC. Used for ePIC collaboration reconstruction and further Detector 2. Used in multiple Jlab experiments and prototypes



JANA4ML4FPGA



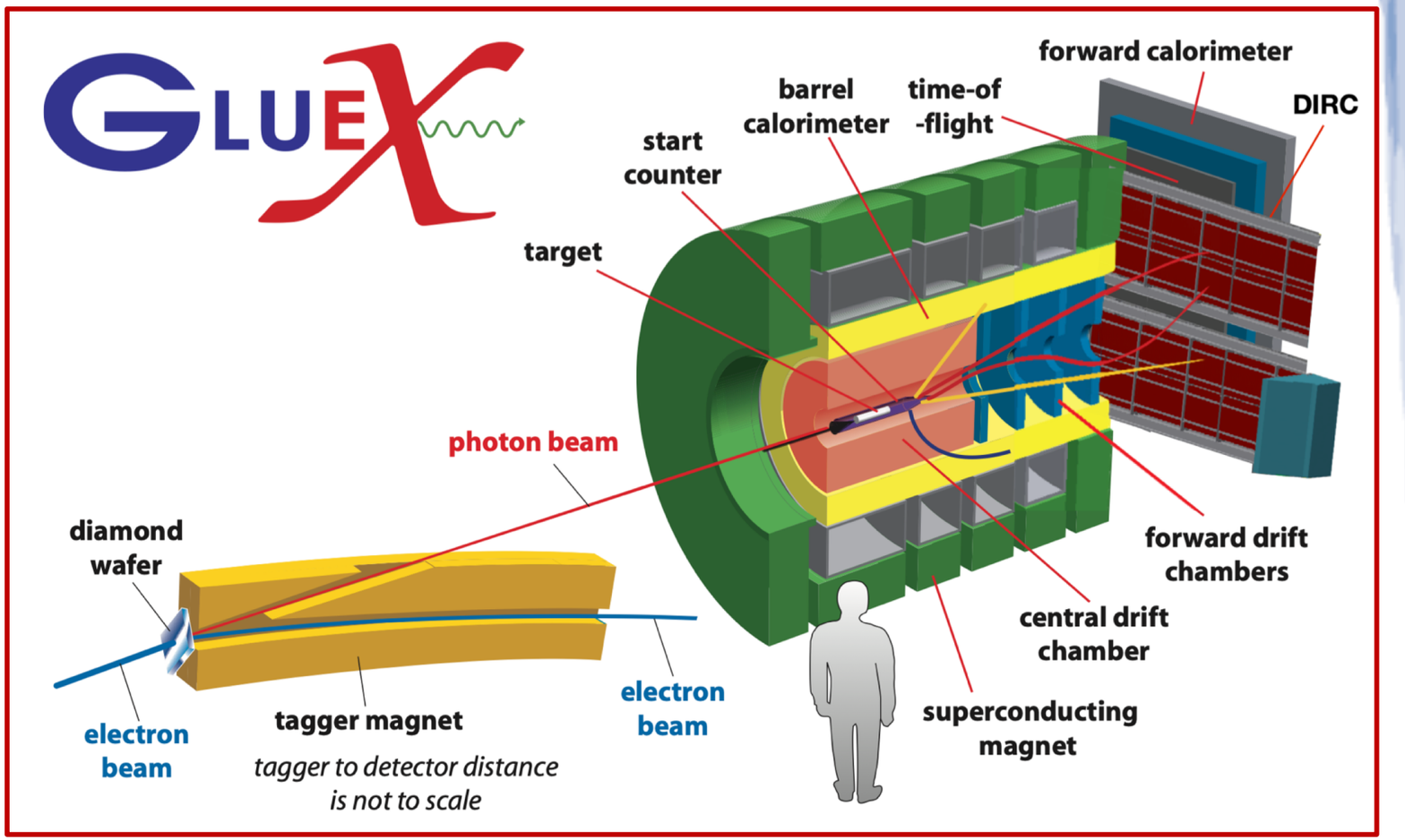
Goals:

- Read and write EVIO
- Write flat ROOT files
- Receive EVIO by TCP (and save)
- Receive network streams
- Receive FPGA data
- Simulate sending detector data
- Data Quality Monitor
- AI streaming preprocessing
- Conventional preprocessing

Tracking for GlueX experiment

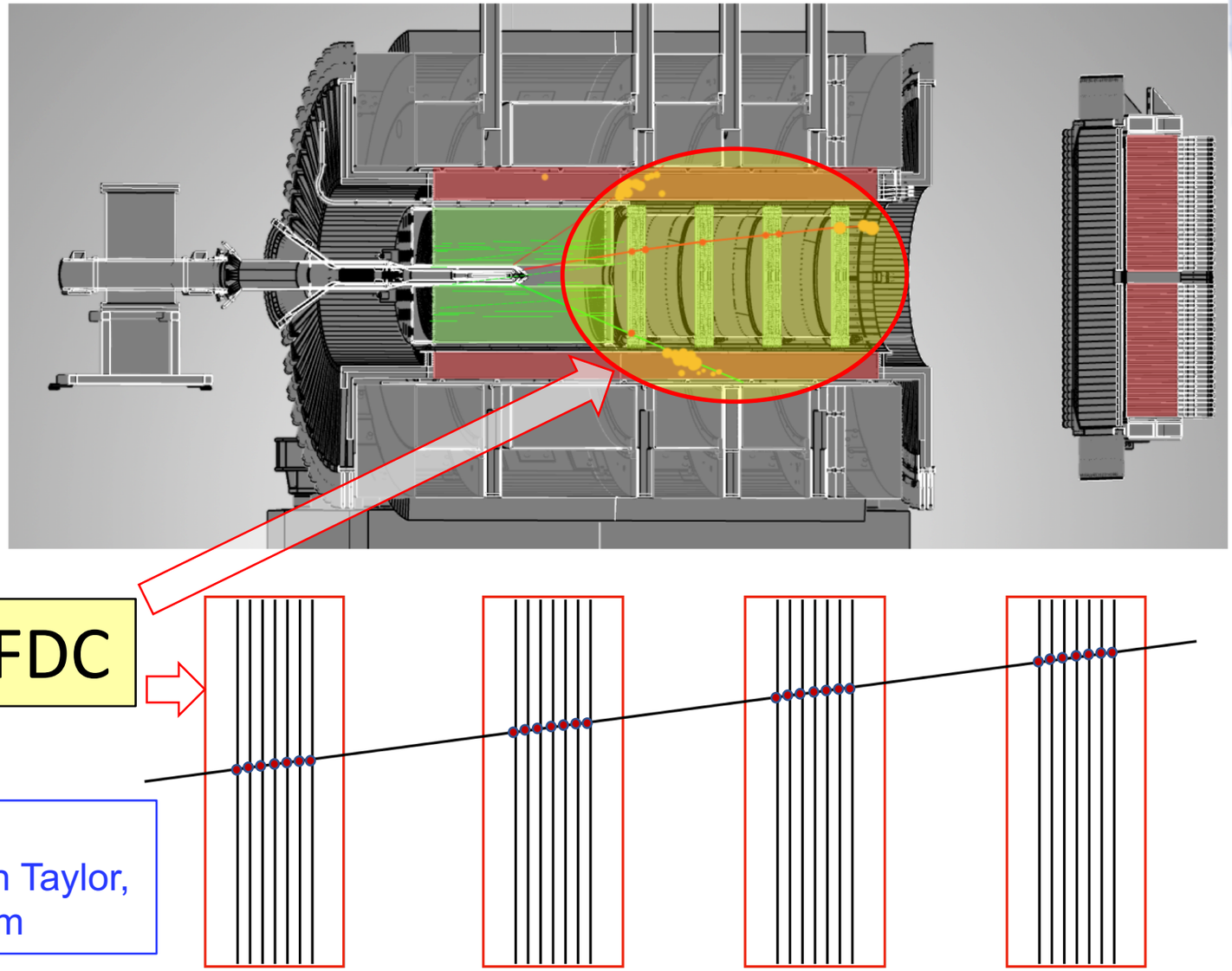
GlueX experiment

- *GlueX is a particle physics experiment located at the Thomas Jefferson National Accelerator Facility (JLab) accelerator in Newport News, Virginia.*
- *Its primary purpose is to better understand the nature of confinement in quantum chromodynamics (QCD) by identifying a spectrum of hybrid and exotic mesons generated by the excitation of the gluonic field binding the quarks.*
- *Hall D is dedicated to the operation with a linearly-polarized photon beam produced by ~12 GeV electrons from CEBAF at Jefferson Lab.*
- *Typical L1 trigger rate 40-70 kHz*
- *Data rate 0.7 – 1.2 GB/s*
- *L1 Trigger latency 3.5 us.*



Tracking for GlueX experiment

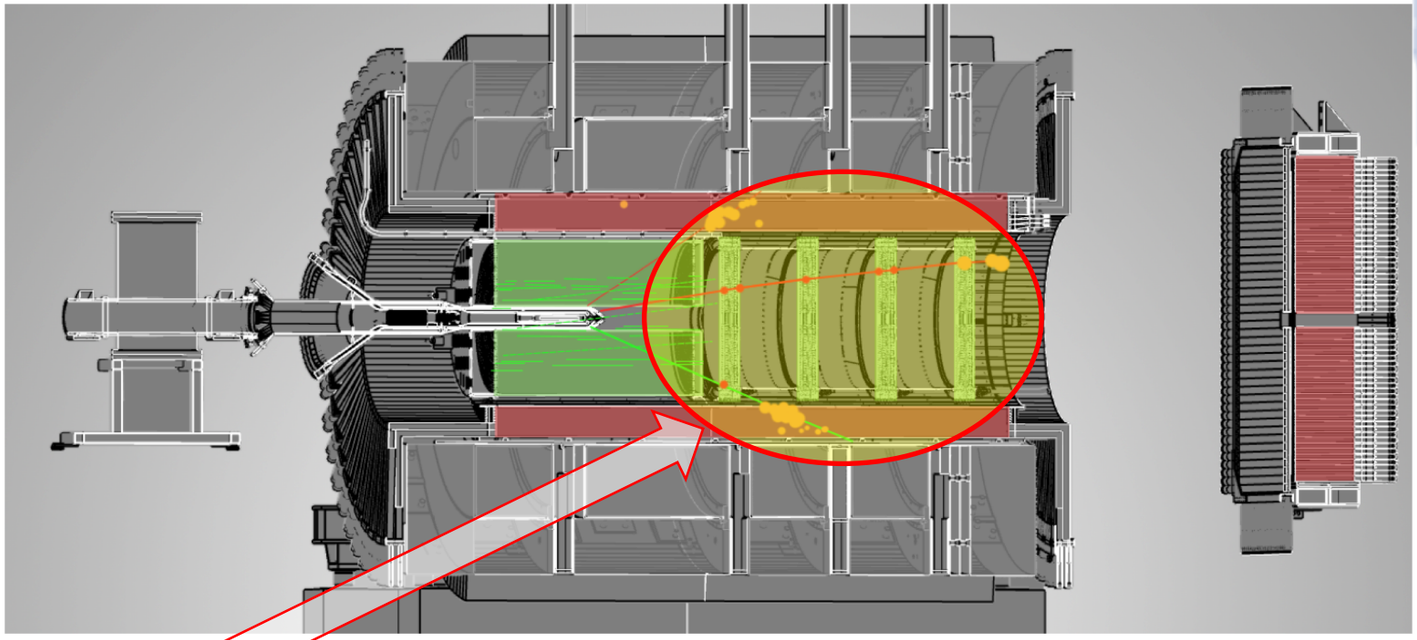
- The first target for implementing neural network-based tracking is the Forward Drift Chamber (FDC).
- The GlueX experiment has relatively low occupancy:
- Number of hits/event:
  - (Q25, Q75, Max) = (50, 70, 558)
- Number of tracks/event
  - (Q25, Q75, Max) = (4, 6, 11)
- *This, in principle, makes it possible to fit a neural network in existing FPGAs.*
- *The FDC consists of 4 modules, each consisting of 6 planes, providing up to 24 points per track.*
- *The FDC is placed in a magnetic field, so the particles move in a helical trajectory.*



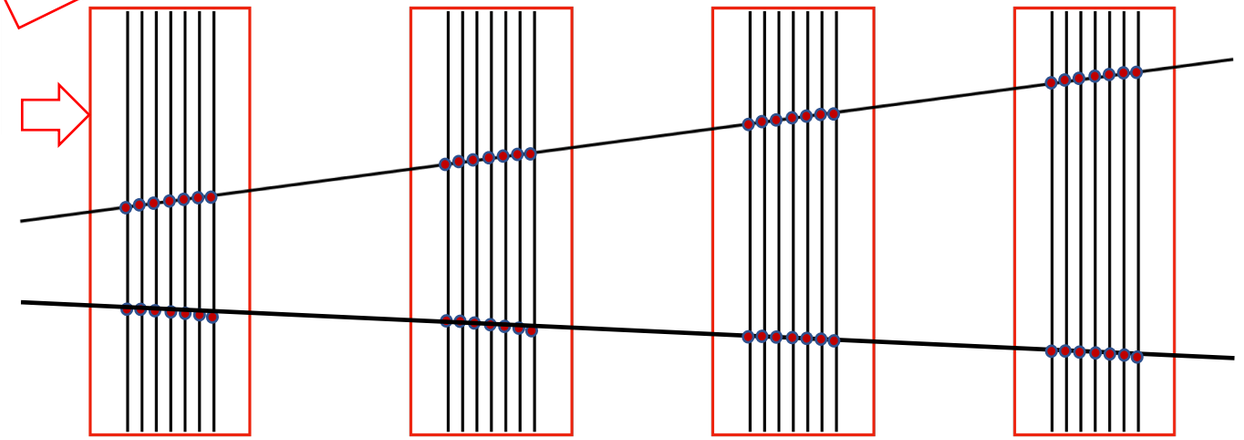
Team:
Ahmed Mohammed, Kishansingh Rajput, Simon Taylor, Sergey Furletov, Denis Furletov, Malachi Schram

Tracking for GlueX experiment

- The first target for implementing neural network-based tracking is the Forward Drift Chamber (FDC).
- The GlueX experiment has relatively low occupancy:
- Number of hits/event:
  - (Q25, Q75, Max) = (50, 70, 558)
- Number of tracks/event
  - (Q25, Q75, Max) = (4, 6, 11)
- *This, in principle, makes it possible to fit a neural network in existing FPGAs.*
- *The FDC consists of 4 modules, each consisting of 6 planes, providing up to 24 points per track.*
  - 6 tracks x 24 hits/trk = 144 hits
- *The FDC is placed in a magnetic field, so the particles move in a helical trajectory.*



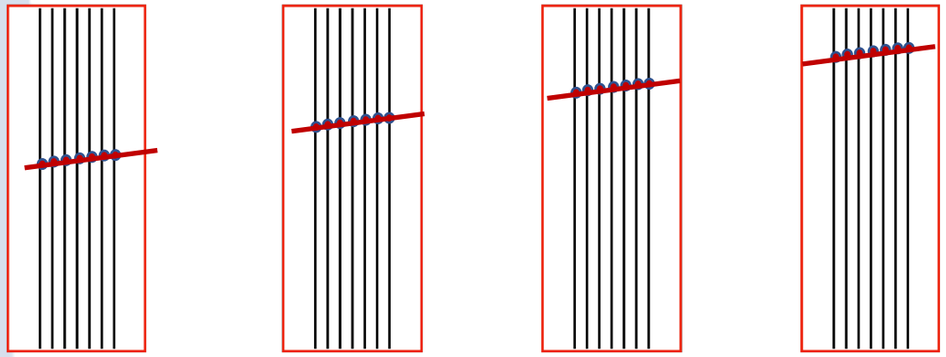
FDC

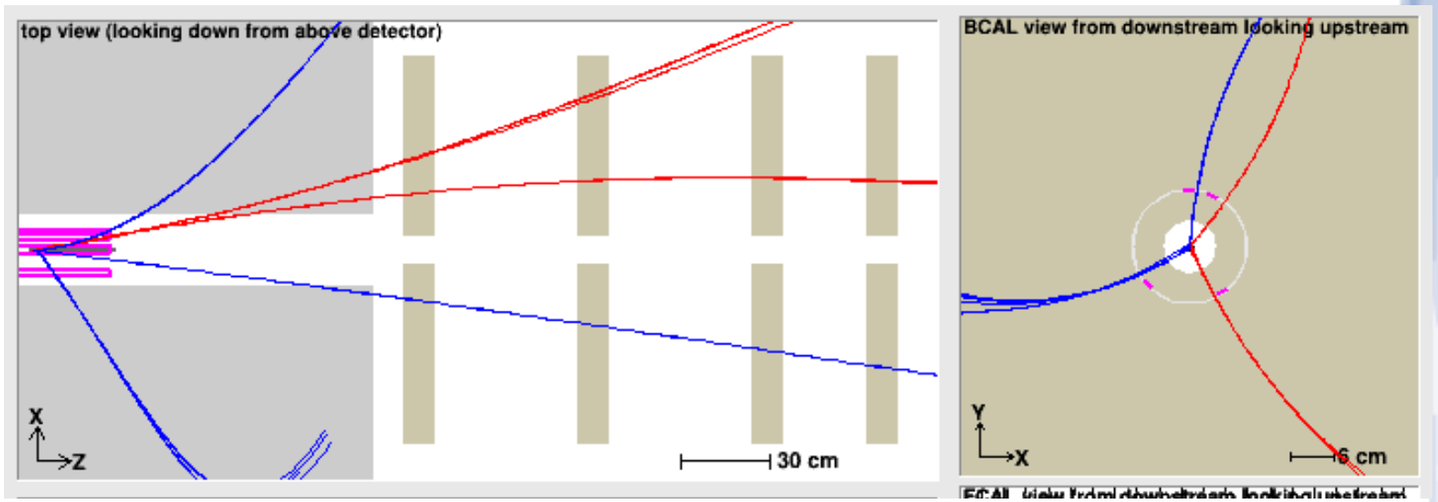


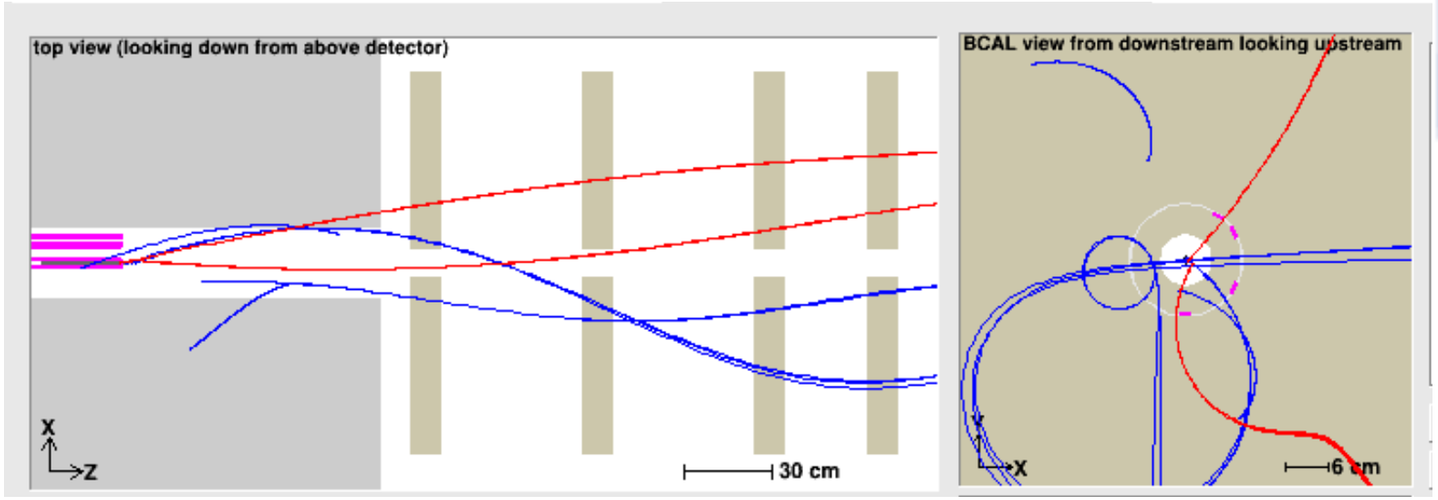
Team:
Ahmed Mohammed, Kishansingh Rajput, Simon Taylor, Sergey Furletov, Denis Furletov, Malachi Schram

Event Display

- *The FDC geometry with 6 closely spaced planes and large distances between modules makes it difficult to directly use GNN for pattern recognition in a magnetic field, see event display on the right.*
- *Moreover, a large graph uses too many FPGA resources – need to process > 150 hits.*
- *Better results are achieved by using a two-stage reconstruction:*
  - in first GNN, the track segments in each module are reconstructed and fitted with a straight line,
  - and then the resulting vectors are fed into a second GNN to reconstruct the full track.

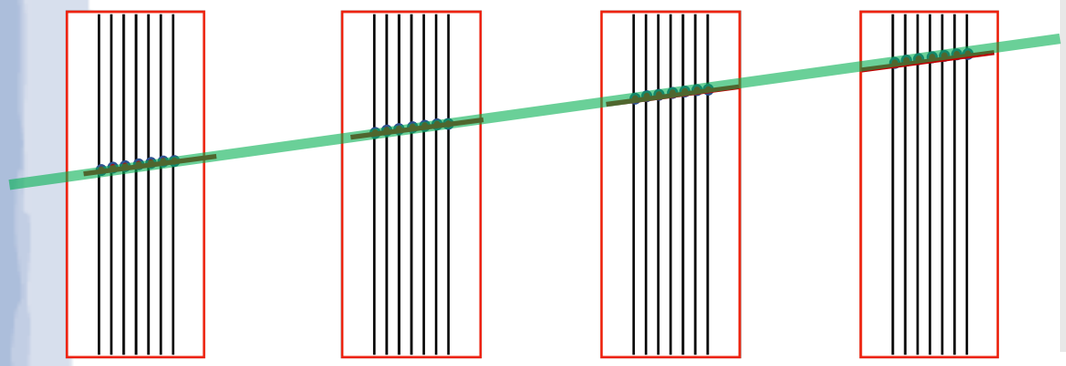


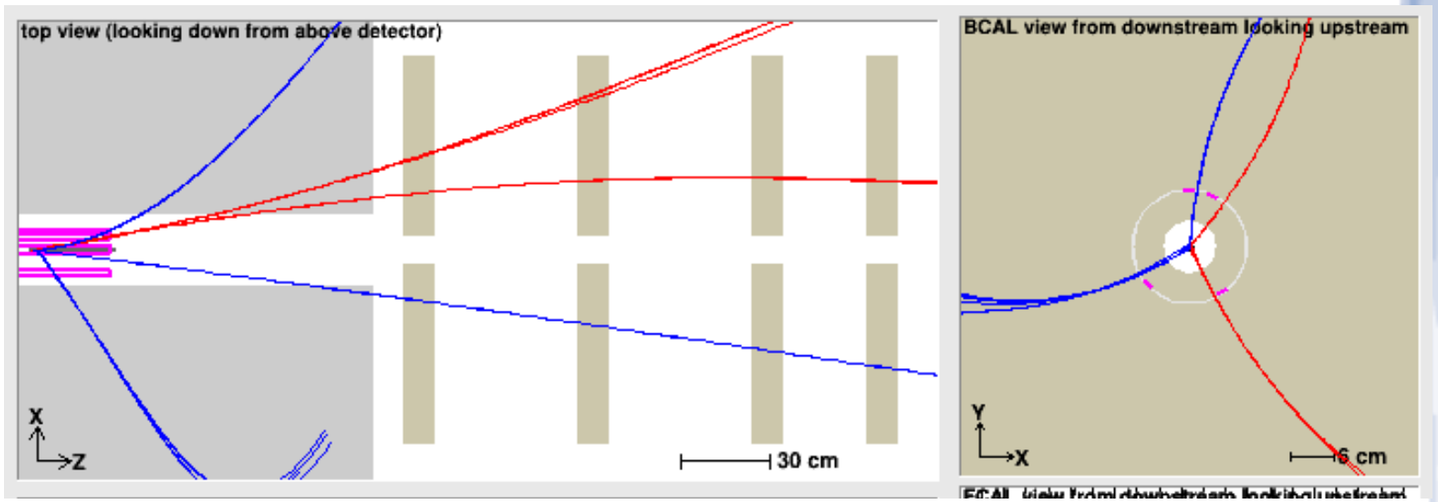


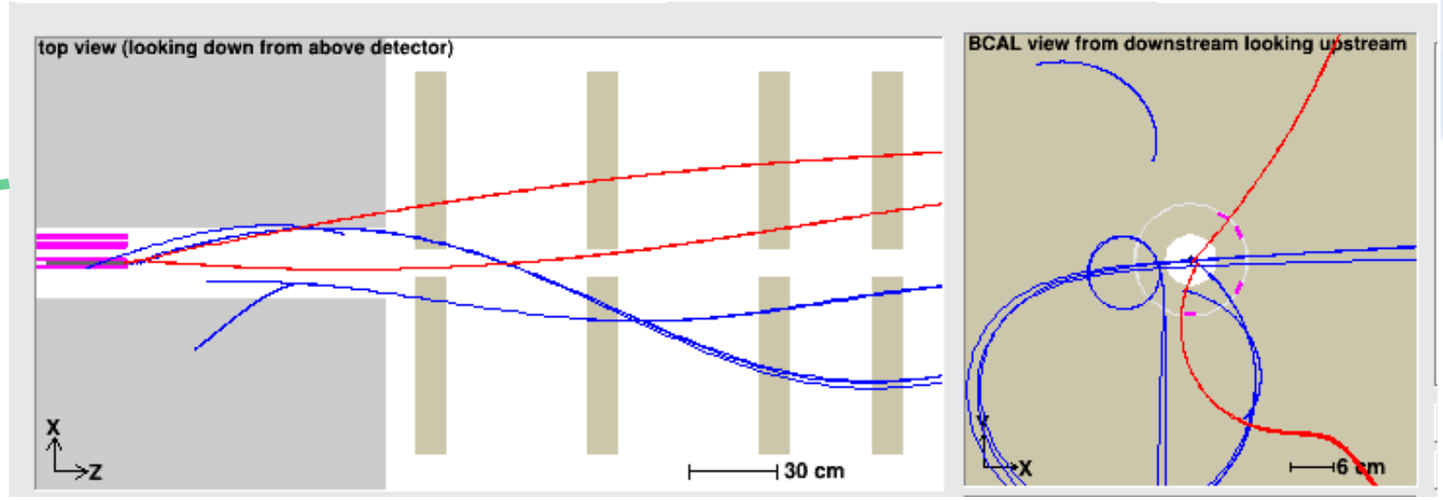


Event Display

- *The FDC geometry with 6 closely spaced planes and large distances between modules makes it difficult to directly use GNN for pattern recognition in a magnetic field, see event display on the right.*
- *Moreover, a large graph uses too many FPGA resources – need to process > 150 hits.*
- *Better results are achieved by using a two-stage reconstruction:*
  - in first GNN, the track segments in each module are reconstructed and fitted with a straight line,
  - and then the resulting vectors are fed into a second GNN to reconstruct the full track.

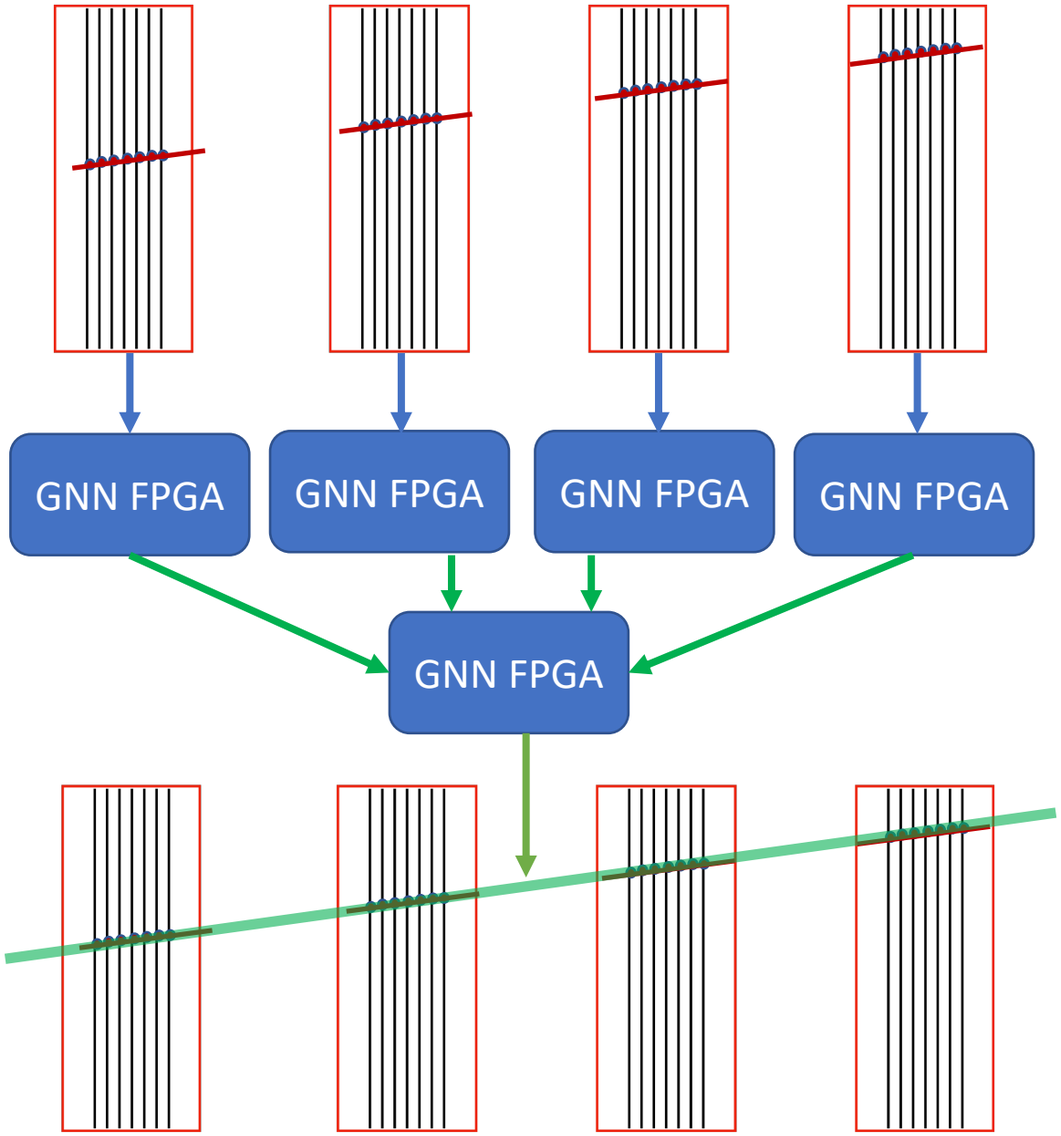






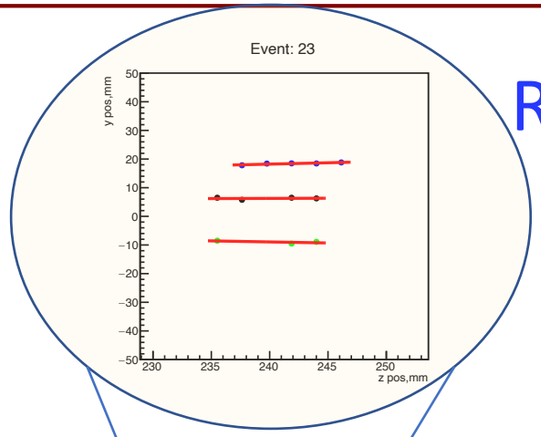
Processing with FPGA

- *The FDC geometry with 6 closely spaced planes and large distances between modules makes it difficult to directly use GNN for pattern recognition in a magnetic field, see event display on the right.*
- *Better results are achieved by using a two-stage reconstruction:*
  - in first GNN, the track segments in each module are reconstructed and fitted with a straight line,
  - and then the resulting vectors are fed into a second GNN to reconstruct the full track.
- *In this way, FDC modules are processed in parallel and the FPGA resource usage is significantly reduced.*



Reconstruction of track segments in FDC

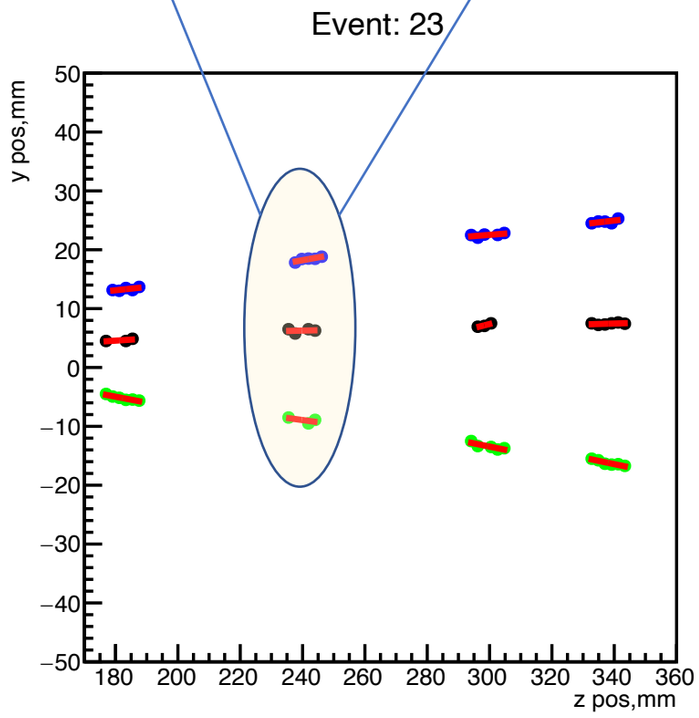
Jefferson Lab
Thomas Jefferson National Accelerator Facility

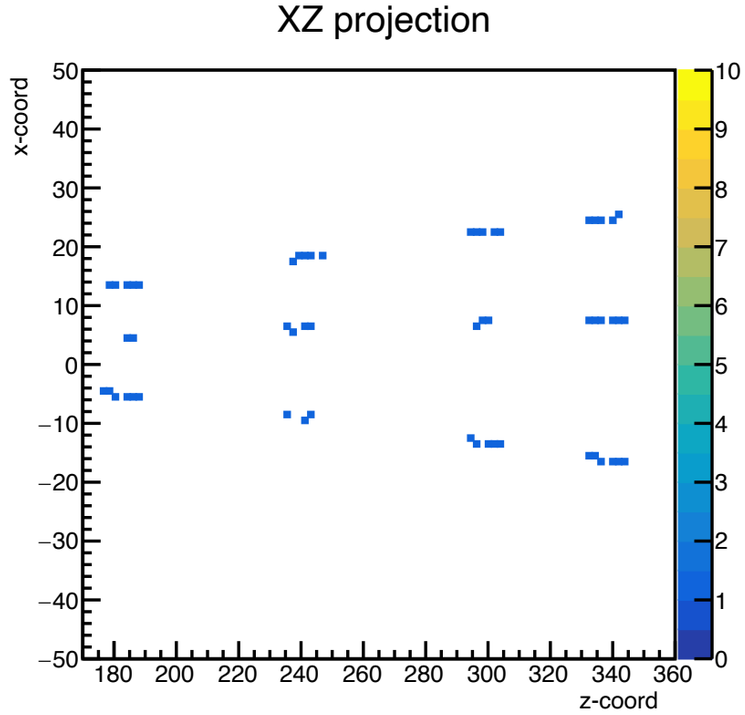


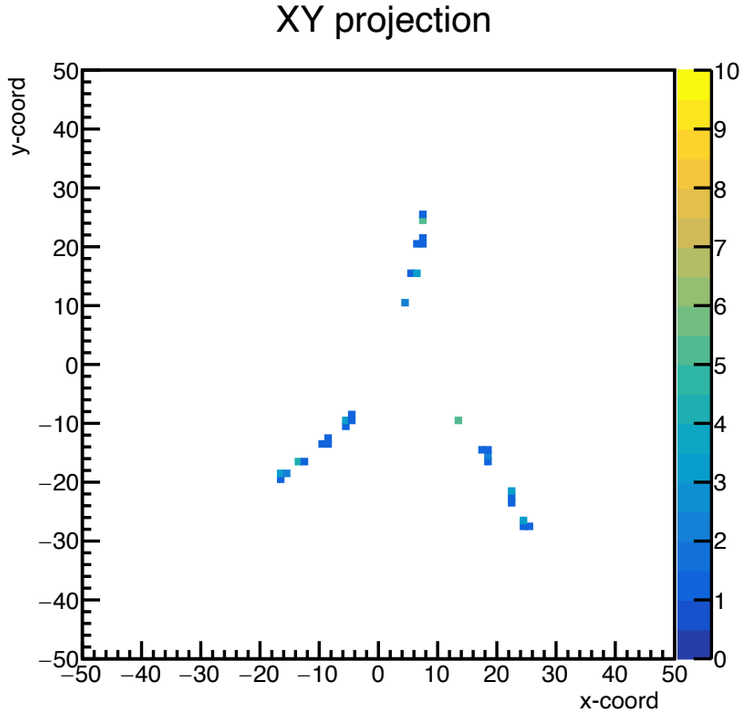
Reconstructed track segments.

Currently we work in 2D, with only one projection: x-z.

Original hits projections in FDC: x-z and x-y

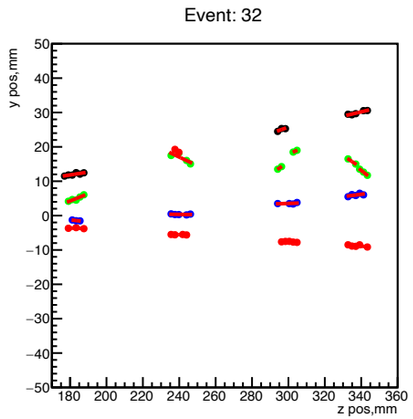


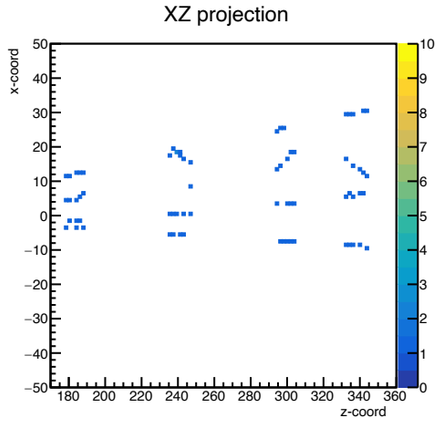


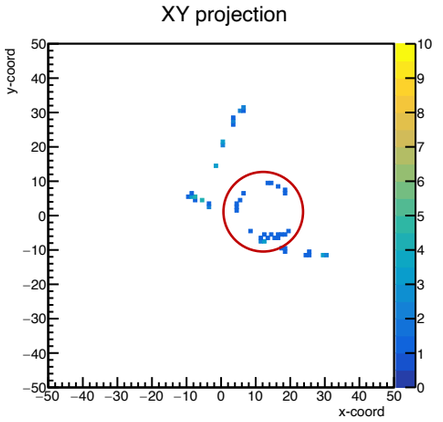


GNN tracking performance

- *The bottom left figure shows the efficiency of segment reconstruction.*
- *The bottom right figure shows the efficiency of full track reconstruction.*
- *The relatively low efficiency for the full track is explained by the presence of low momentum tracks, and hence high curvature, for which single projection is not efficient. (top right)*
- *In the future we plan to work in 3D.*
- *For now we will move forward with the implementation of the current 2D model on FPGA.*

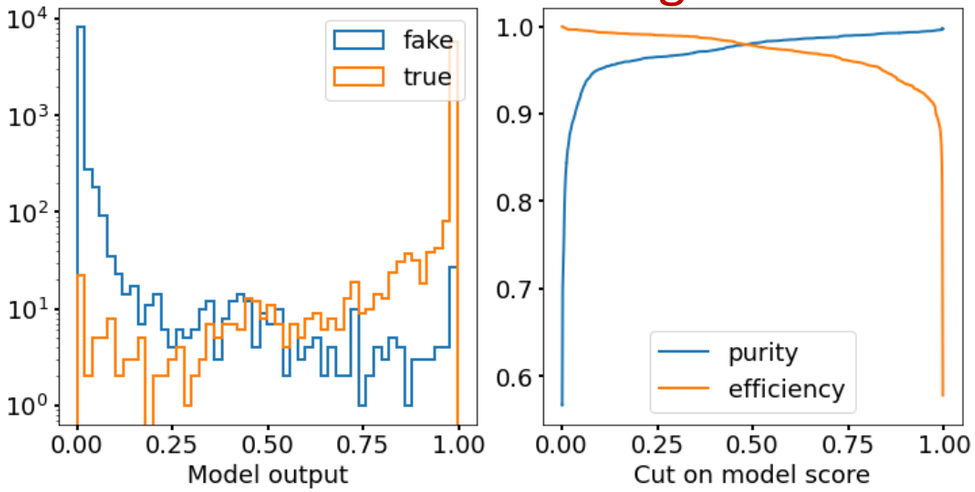






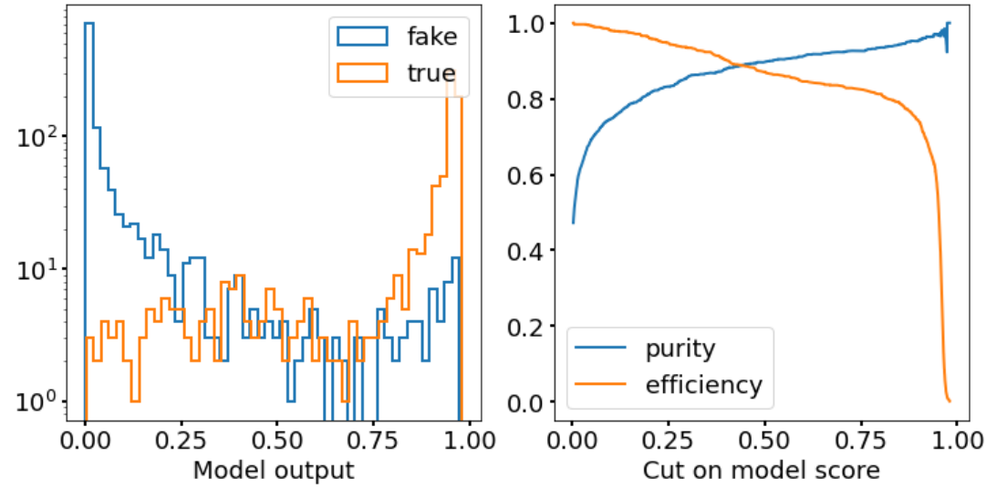
Segments

```
Accuracy:            0.983474
Precision (purity):  0.980984
Recall (efficiency): 0.978171
```



Tracks

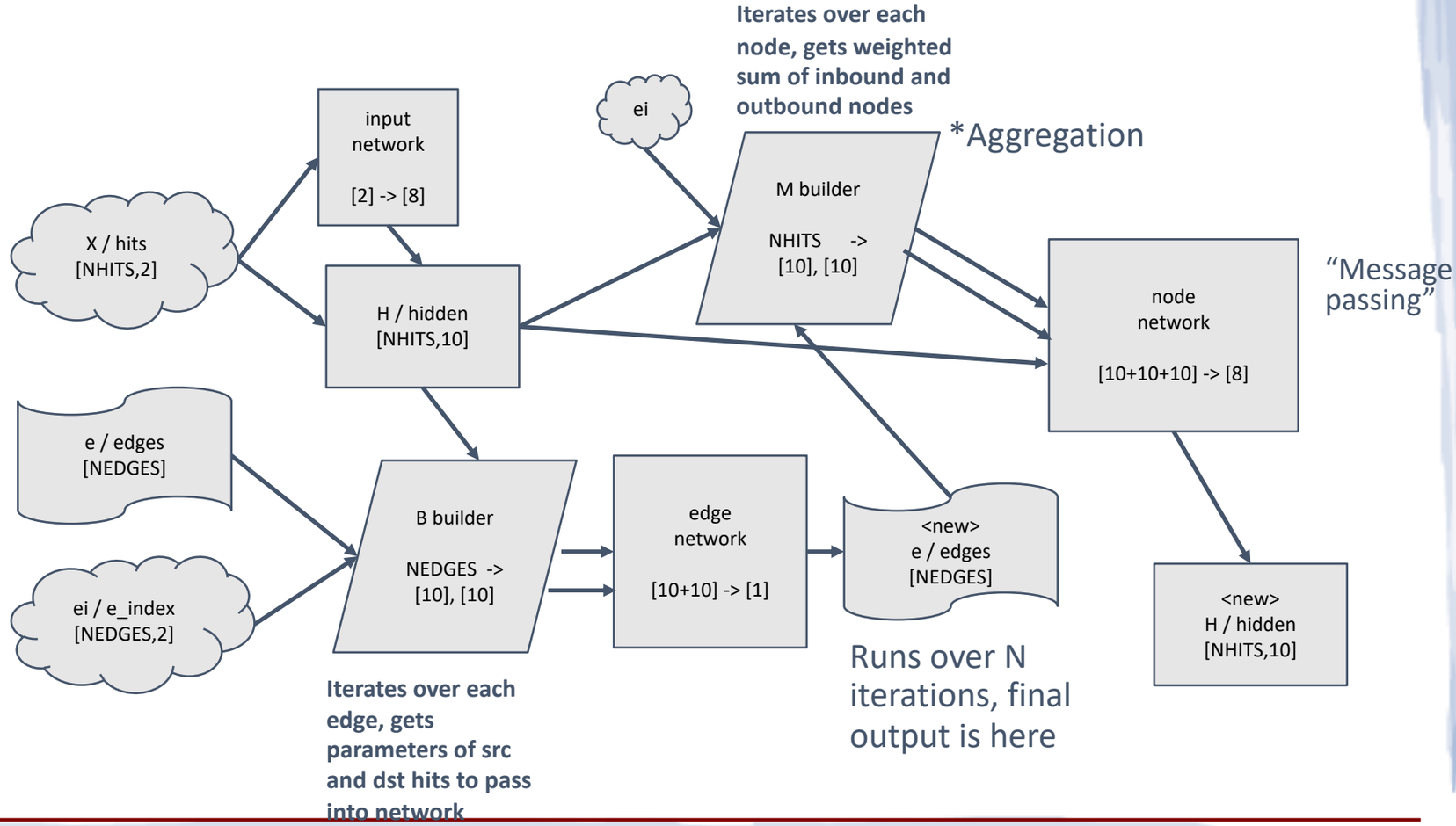
```
Accuracy:            0.908382
Precision (purity):  0.897114
Recall (efficiency): 0.870889
```



New GNN for FDC tracking

- *The results shown look good, but we are still limited to 30 hits/nodes in the network, while FDC requires at least 100 nodes.*
- *We started designing a new GNN network capable of handling 150 nodes and 256 edges.*
- *The new GNN design uses the layer library from HLS4ML with a custom wrapper and aggregation functions.*
- *Also removed all dependency to external libraries – Hep.TrkX and sonnet from DeepMind.*

- *Trained three (3) keras neural networks*
- *Using `hls4ml` library, convert each of the three (3) networks into separate C++ projects*
- *Manually/Script rename project files to append "myproject<_type>.*" where <_type> is any of [_i, _n, _e]*
- *Wrapper project to retrieve data from stream and custom functions to build B and M matrix values*
- *Call each network within this `runner` top function*



Optimized GNN IP

- *The GlueX trigger rate is up to 70 kHz, so on average we have ~14 µs to process events.*
- *We optimized the GNN to have a latency of ~10 µs, which allows it to operate at 70 kHz.*
- *On the other hand, the neural network fits in an FPGA and supports 150 nodes and 256 edges.*
- *Next we plan to test it on hardware.*

MODULES & LOOPS	ISSUE TY	SLACK	LATENCY(CYCLES)	LATENCY(NS)	ITERATION LATENCY	INTERVAL	TRIP COUNT	PIPELINED	BRAM(%	DSP(%)	FF(%)	LUT(%)
runGraphNetwork (6)		-0.24	1991	9.955E3	-	1992	-	no	~0	11	10	25
edge_network (1)		-	271	1.355E3	-	271	-	no	~0	3	~0	3
node_runner (1)	⚠	-0.24	363	1.815E3	-	363	-	no	0	5	7	21
runGraphNetwork_Pipeline_INPUT_HIT_LOOP (1)		-	159	795.000	-	159	-	no	0	2	1	~0
runGraphNetwork_Pipeline_VITIS_LOOP_42_1 (1)		-	302	1.510E3	-	302	-	no	0	0	~0	~0
runGraphNetwork_Pipeline_VITIS_LOOP_72_2 (1)		-	514	2.570E3	-	514	-	no	0	0	~0	~0
runGraphNetwork_Pipeline_VITIS_LOOP_136_3 (1)		-	258	1.290E3	-	258	-	no	0	0	~0	~0

Outlook

- *An FPGA-based Neural Network application would offer online event preprocessing and allow for data reduction based on physics at the early stage of data processing.*
- *The ML-on-FPGA solution complements the purely computer-based solution and mitigates DAQ performance risks.*
- *FPGA provides extremely low-latency neural-network inference.*
- *Open-source HLS4ML software tool with Xilinx® Vivado® High Level Synthesis (HLS) accelerates machine learning neural network algorithm development.*

- *The ultimate goal is to build a real-time event filter based on physics signatures.*

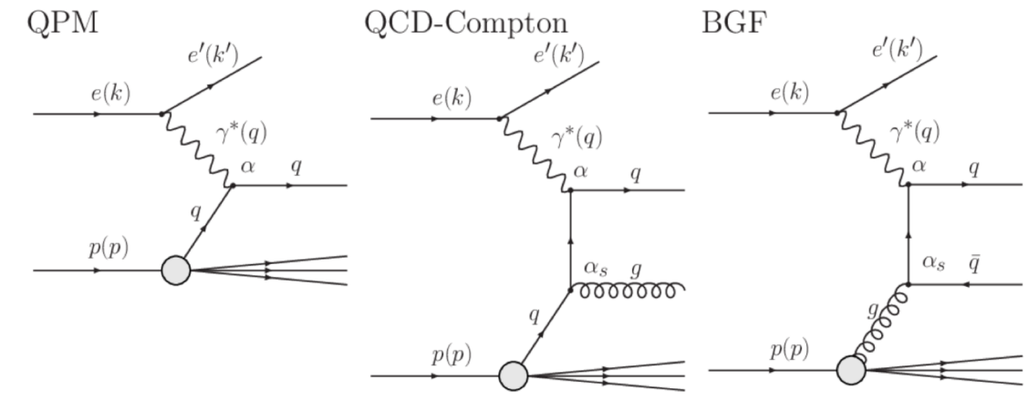


Figure 2.1: Feynman diagrams of the Quark Parton Model, QCD-Compton and Boson Gluon Fusion processes in NC DIS.

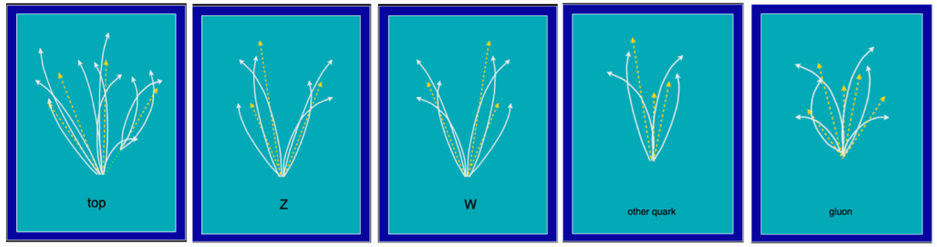
Published in 2007

Measurement of multijet events at low x_{Bj} and low Q^2 with the ZEUS detector at HERA

T. Gosau

Case study: jet tagging

Study a multi-classification task: discrimination between highly energetic (boosted) ***q, g, W, Z, t*** initiated jets



t→bW→bqq	Z→qq	W→qq	q/g background
3-prong jet	2-prong jet	2-prong jet	no substructure and/or mass ~ 0

Signal: reconstructed as one massive jet with substructure

Jet substructure observables used to distinguish signal vs background [*]

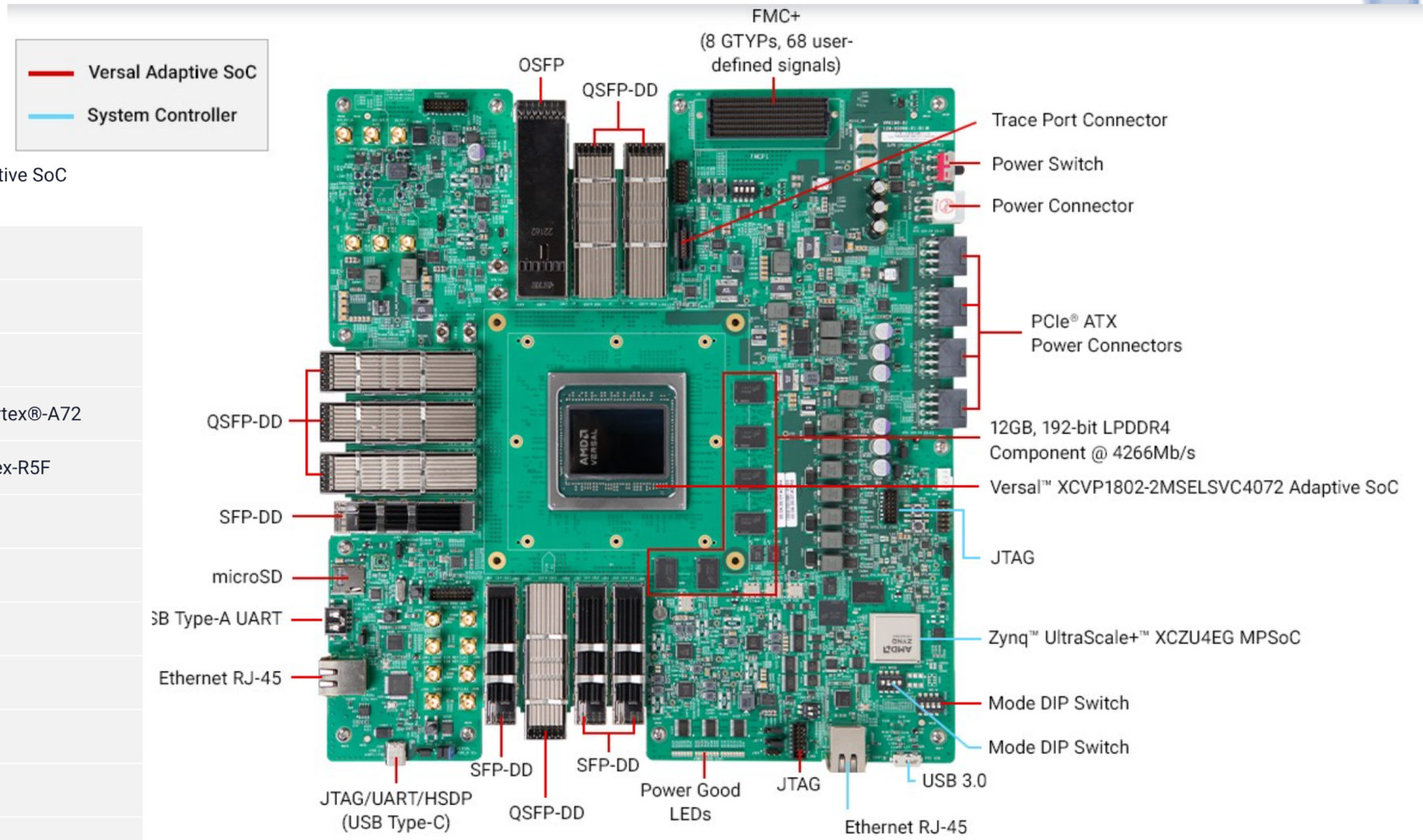
[*] D. Guest at al. PhysRevD.94.112002, G. Kasieczka et al. JHEP05(2017)006, J. M. Butterworth et al. PhysRevLett.100.242001, etc..

11.01.2019 Jennifer Ngadiuba - hls4ml: deep neural networks in FPGAs 25

Backup

Xilinx VPK180 board

Jefferson Lab
Thomas Jefferson National Accelerator Facility



Featuring the Versal Premium **XCVP1802-2MSELSVC4072** Adaptive SoC

System Logic Cells (K)	7,352
LUTs	3,360,896
DSP Engines	14,352
Application Processing Unit	Dual-core Arm® Cortex®-A72
Real-Time Processing Unit	Dual-core Arm Cortex-R5F
GTYP Transceivers (32.75Gb/s)	28[1,2]
GTM Transceivers (58G (112G))	140 (70)[2]
PCIe w/ DMA & CCIX (CPM5)	2 x Gen5x8[3]
PCI Express	2 x Gen5x4[3]
100G Multirate Ethernet MAC	8
600G Ethernet MAC	7
400G High-Speed Crypto Engine	4

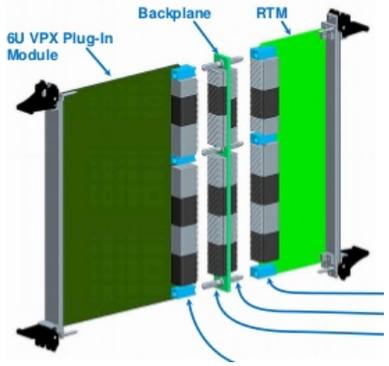
ADC based DAQ for PANDA STT

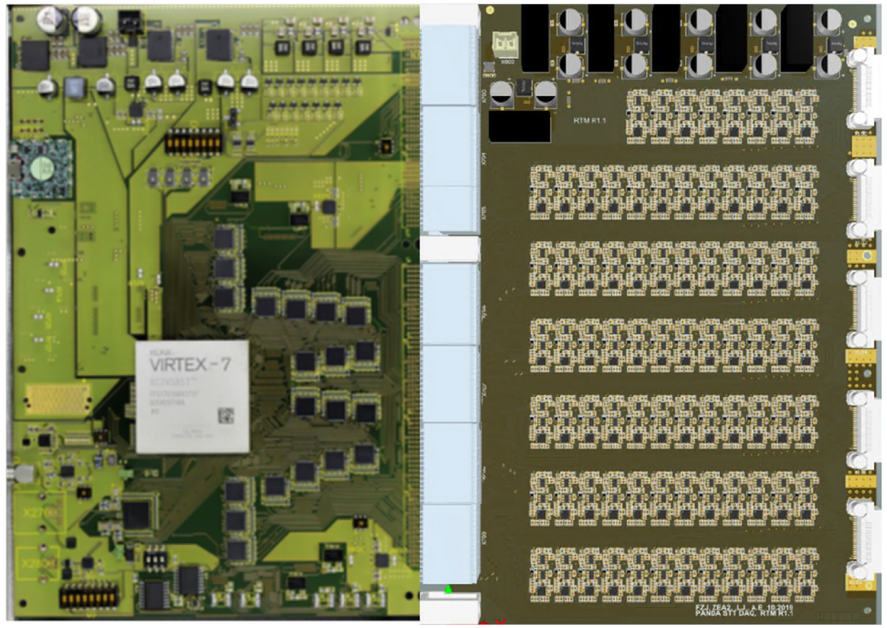
Jefferson Lab
Thomas Jefferson National Accelerator Facility

Level 0 Open VPX Crate

ADC based DAQ for PANDA STT (one of approaches):

- 160 channels (shaping, sampling and processing) per payload slot, 14 payload slots+2 controllers;
- **totally 2200 channels per crate**;
- time sorted output data stream (arrival time, energy,...)
- noise rejection, pile up resolution, base line correction, ..





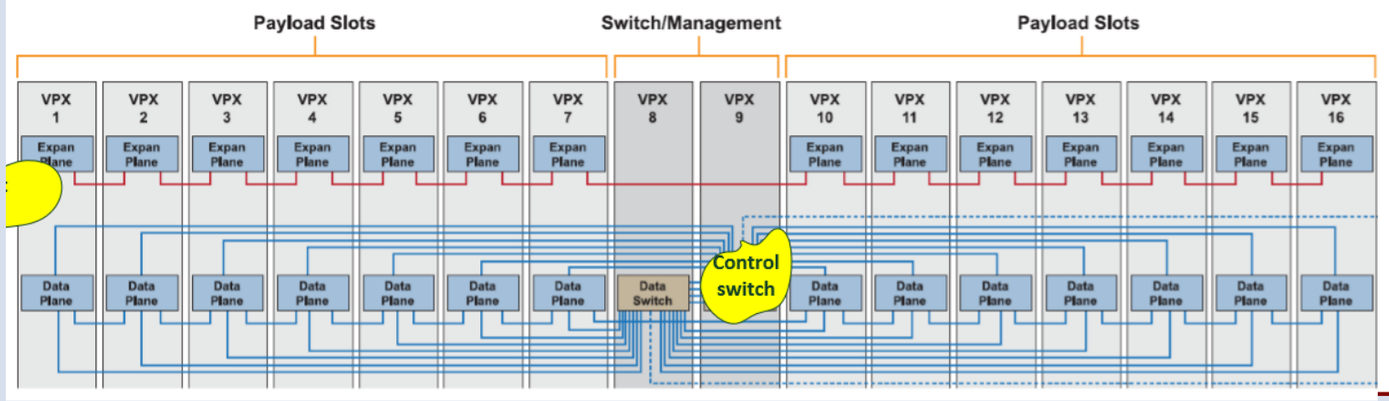
- 40 4-channel ADCs (configurable up to 1 GSPS);
- Single Virtex7 FPGA

- 160 Amplifiers;
- 5 connectors for 32-pins samtec cables

✦ *All information from the straw tube tracker is processed in one unit.*

✦ *Allows to build a complete STT event.*

✦ *This unit can also be used for calorimeters readout and processing.*



https://doi.org/10.1088/1748-0221/17/04/C04022
2022_JINST_17_C04022

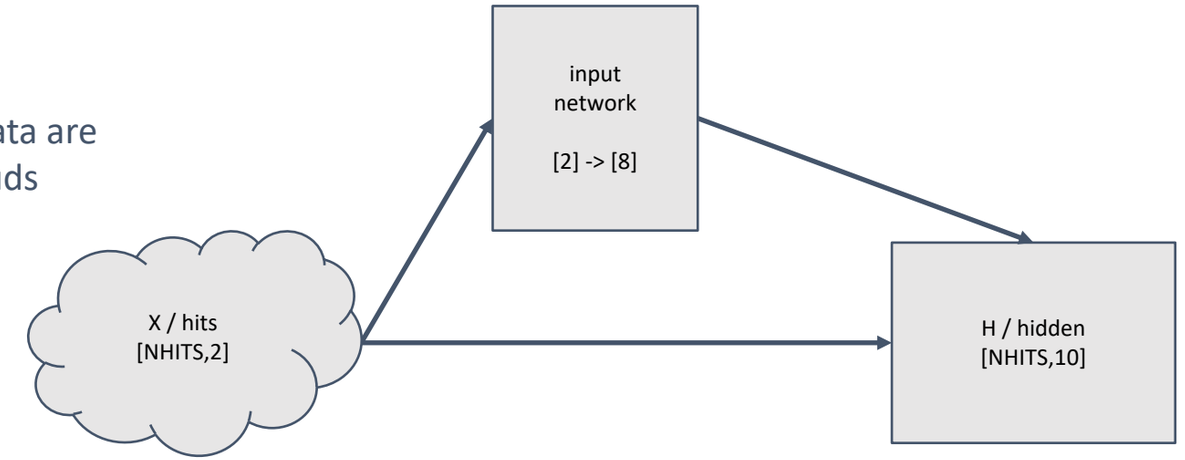
Powerful Backplane up to 670 GBs

L. Jokhovets, P Kulessa ..

JÜLICH Forschungszentrum

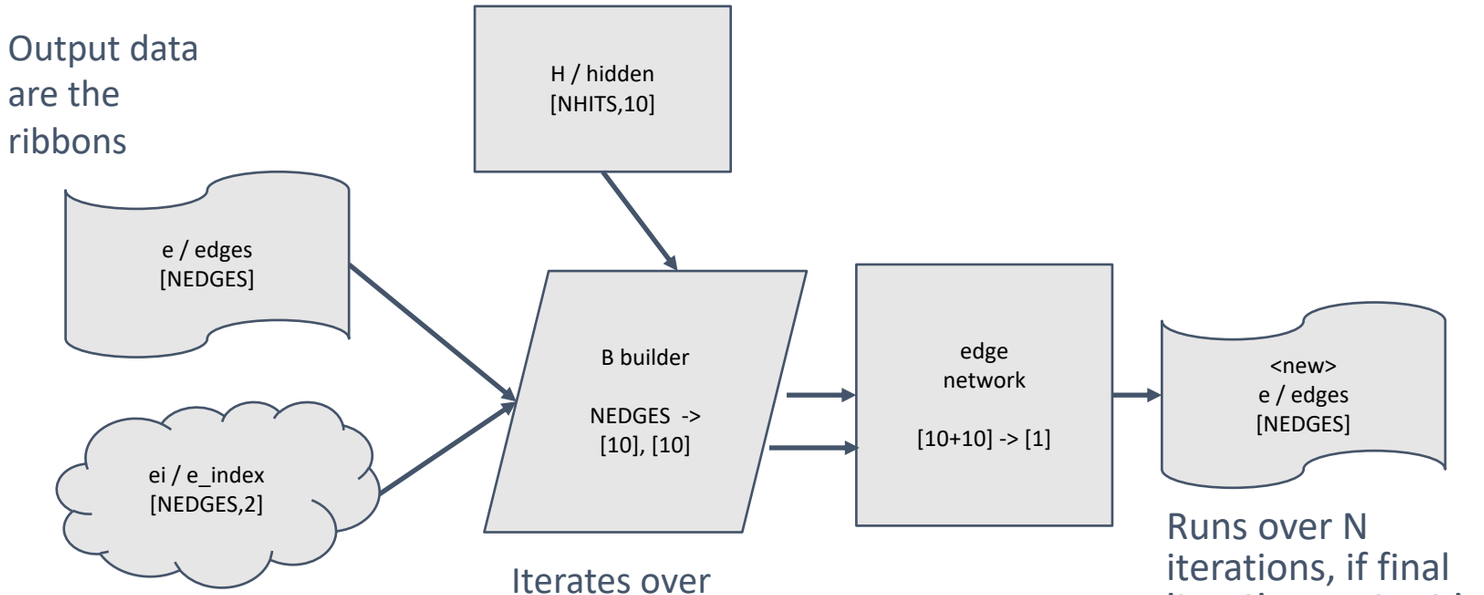
Step 1: Input Network

Input data are the clouds

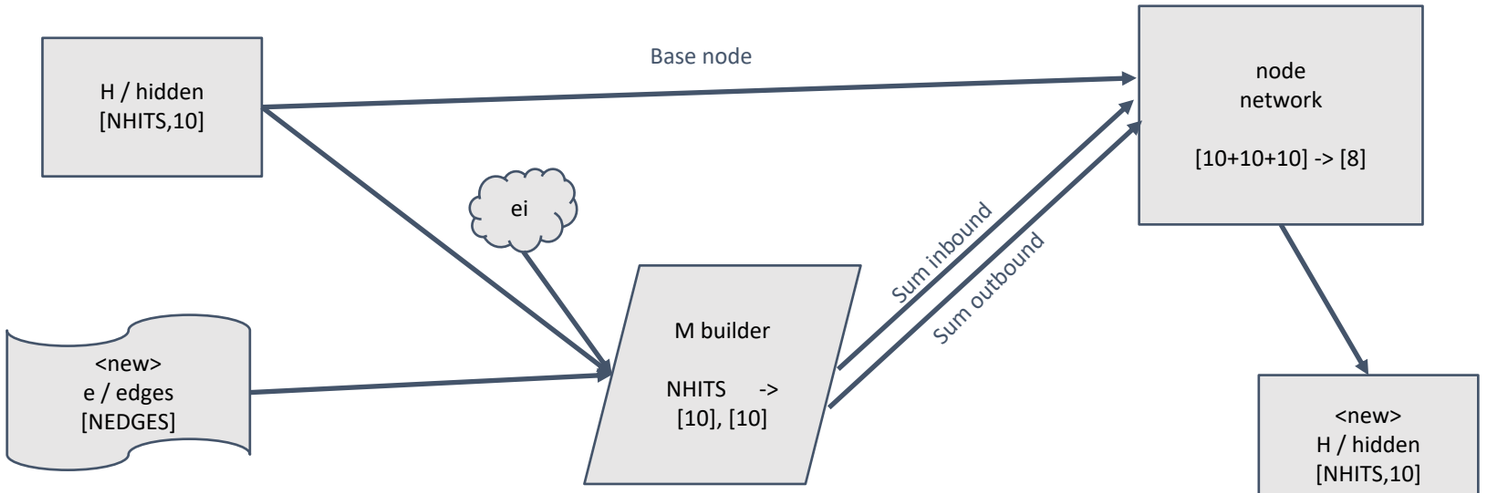


Stores the node information, 10 parameters, 8 hidden and 2 original.

Step 2a: Iterations (edge)



Step 2b: Iterations (node)

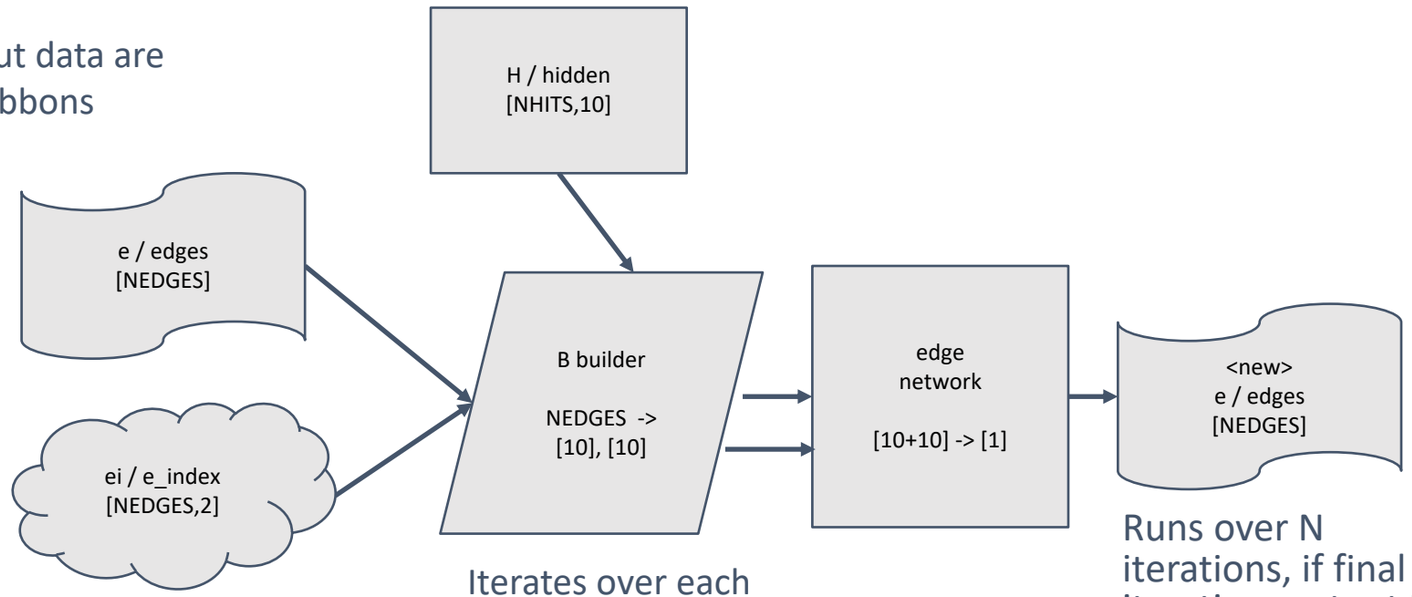


Iterates over each node, gets weighted sum of inbound and outbound nodes

*Aggregation

Step 3: Final edge output

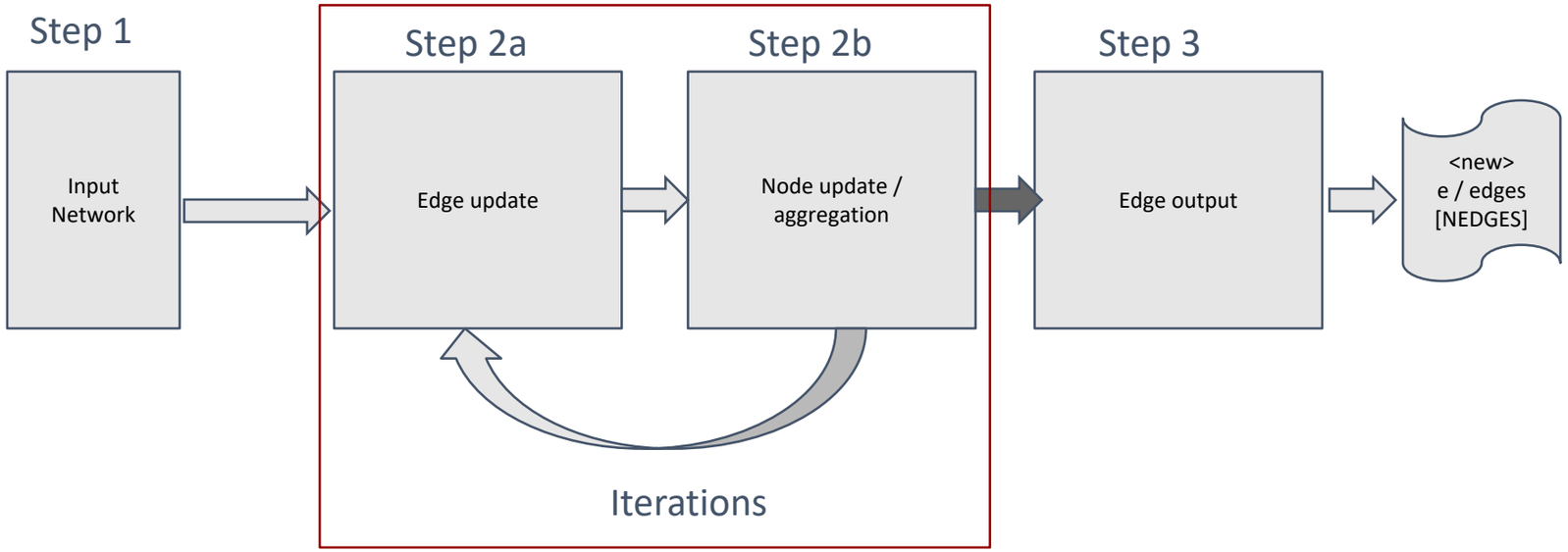
Output data are the ribbons



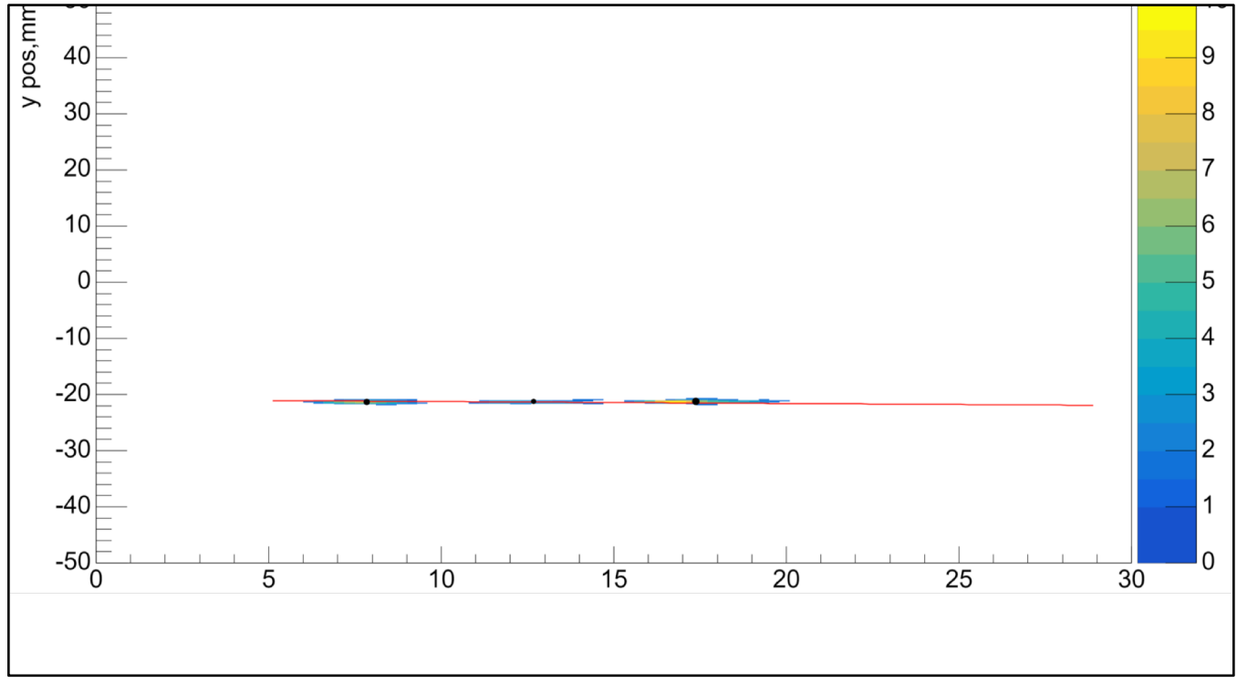
Iterates over each edge, gets parameters of src and dst hits to pass into network

Runs over N iterations, if final iteration, output is here

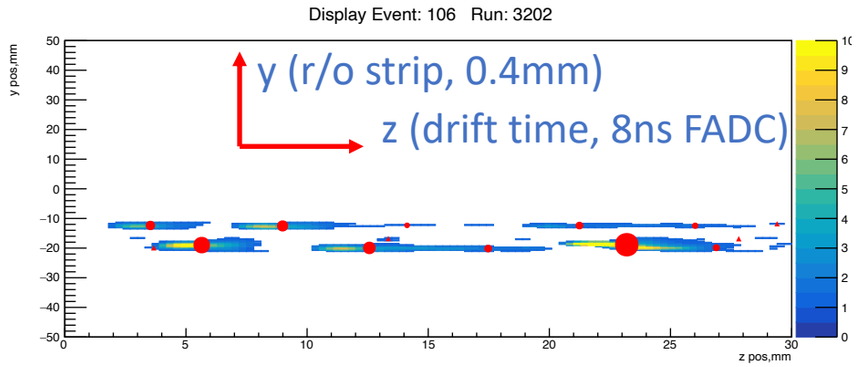
Simple Overview

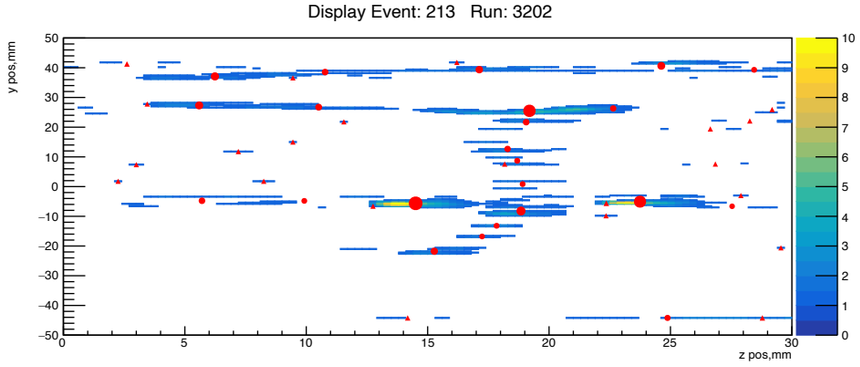


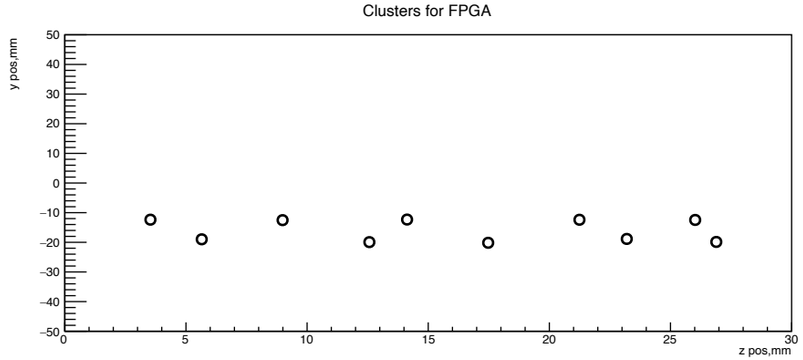
Event display, single track

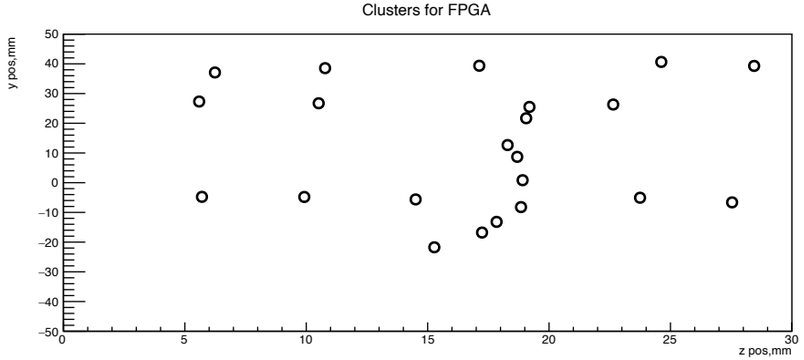


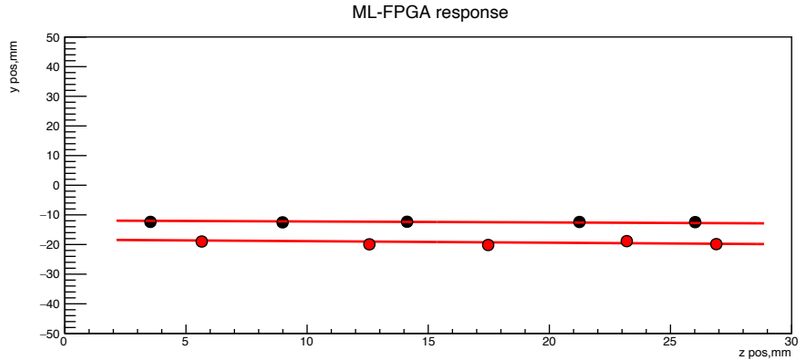
Tracking performance

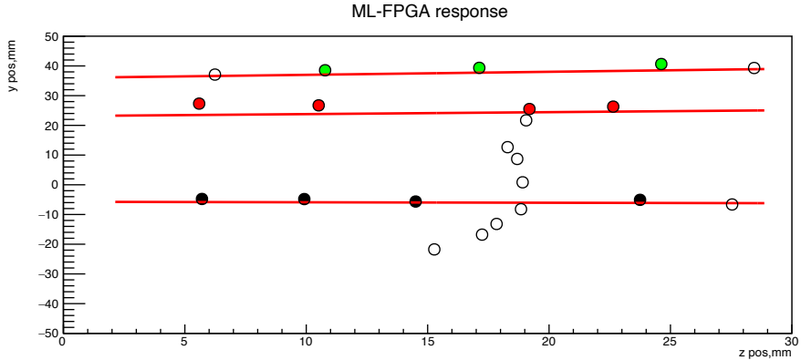




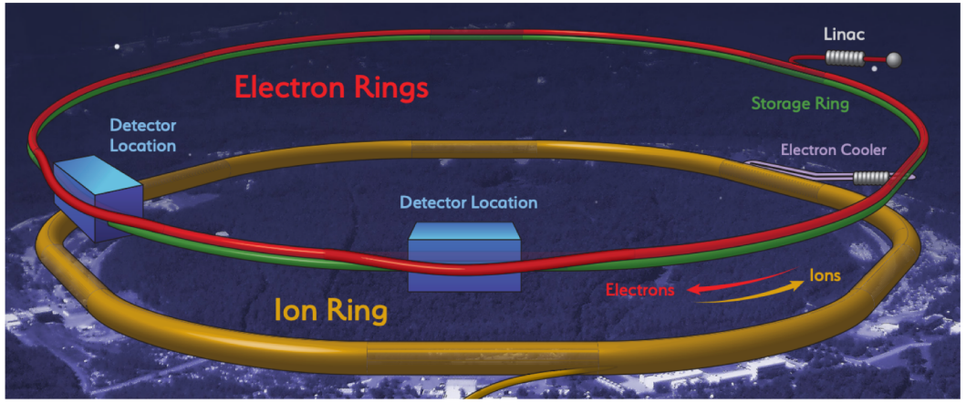








- *Top rows: show ionization along the track in GEMTRD detector.*
  - Red circles are reconstructed clusters using some dE/dx threshold. The size is proportional to energy.
- *Middle rows: after filtering out the noisy clusters, the coordinates of the clusters are sent to the FPGA/GNN for pattern recognition.*
- *Bottom rows: GNN provides labeling of clusters (by color in the figure), the same colors belong to the same track.*
- *Then clusters of the same color (tag) are sent to the track fitting module: LSTM.*
- *The results of track fitting are represented by lines in the figures.*
- *The next step is to count all the ionization in the corridor around the track and send it to the PID module (DNN).*
- *As a bonus, GEMTRD provides a track segment for the global tracking system.*


Linac
Electron Rings
Storage Ring
Detector Location
Electron Cooler
Detector Location
Electrons
Ions
Ion Ring