LA-UR-25-29820

# Intelligent Experiments through Real-Time AI:

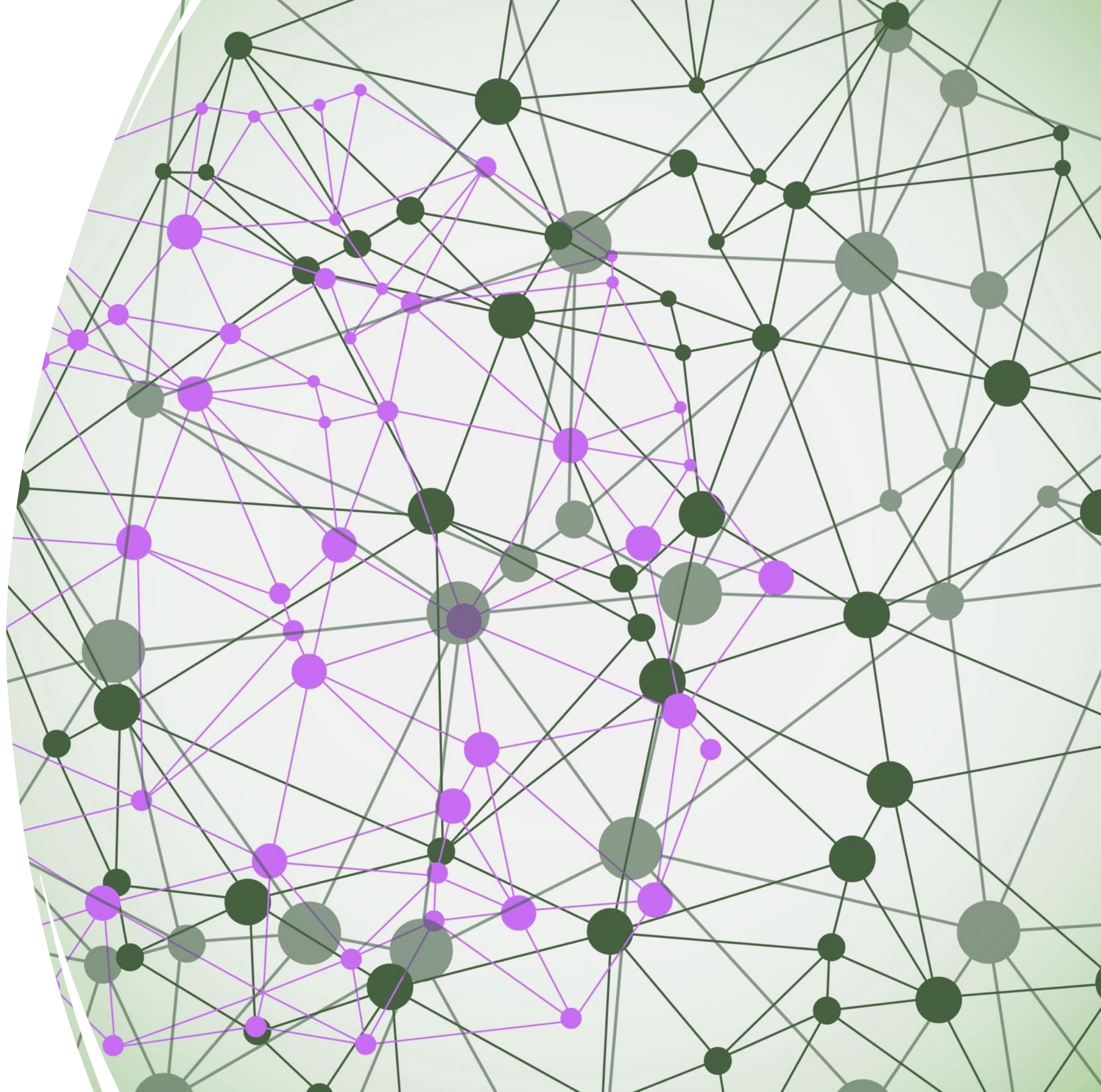
***Fast Data Processing and Autonomous Detector Control for High-Energy Nuclear Experiments*** *(Fast-ML)*

Ming Liu

Los Alamos National Lab

AI4EIC Workshop@MIT

10/27/2025



# The Team – NP, HEP, CS and EE

❑A joint effort of NP, HEP, CS and EE

- LANL, MIT, FNAL, NJIT, GIT, ORNL et al

❑Physics simulation and AI-ML algorithms

❑Firmware implementation

- *hls4ml*, FlowGNN etc.

❑Demonstrator deployment

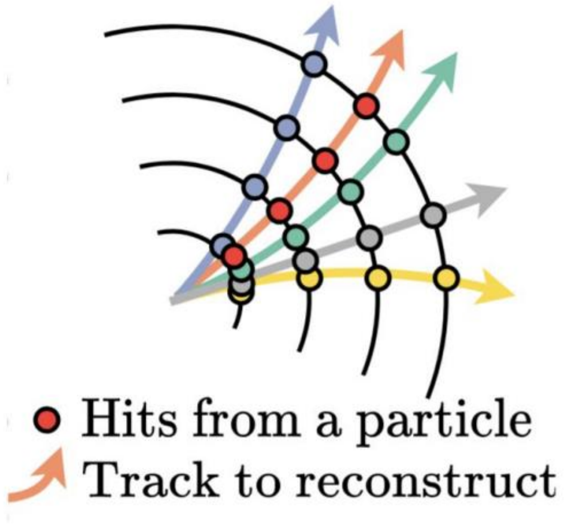
- *FPGA, GPU, CPU etc.*

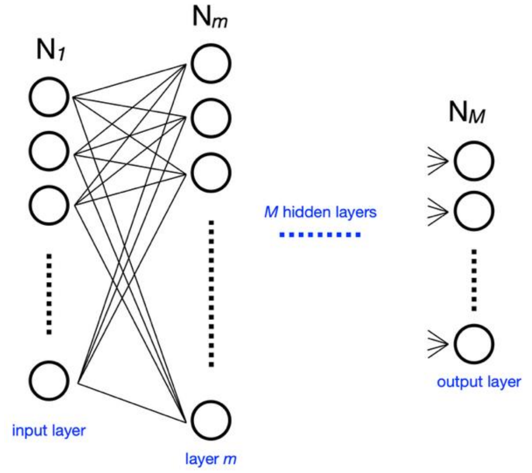
# Why Fast-ML?

- High data throughput from modern detectors in high-energy experiments
  - O(1~10)TB/s @detectors, CMS, ATLAS, ALICE, **sPHENIX, EIC** ...
  - Very large data volume (~100PB/year), it also takes a long time to process the data offline for physics analysis

- **Our goal** - use AI/ML based algorithms to tag important (rare) events in real-time with high efficiency in p+p and e+p/A collisions, for fast data filtering/reduction

**sPHENIX as the first test ground, ultimately for EIC in 2030s**

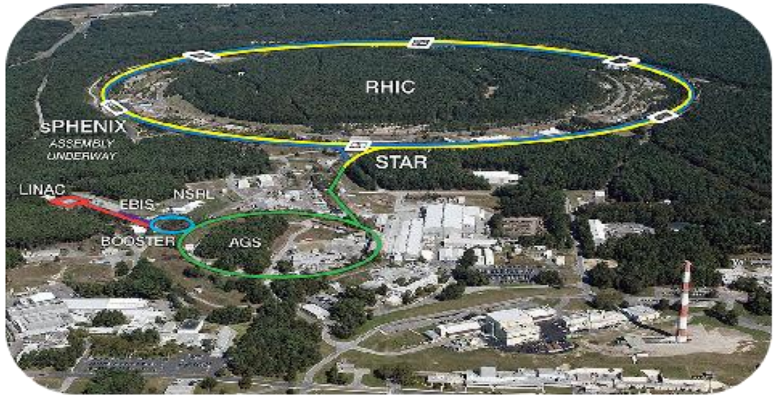


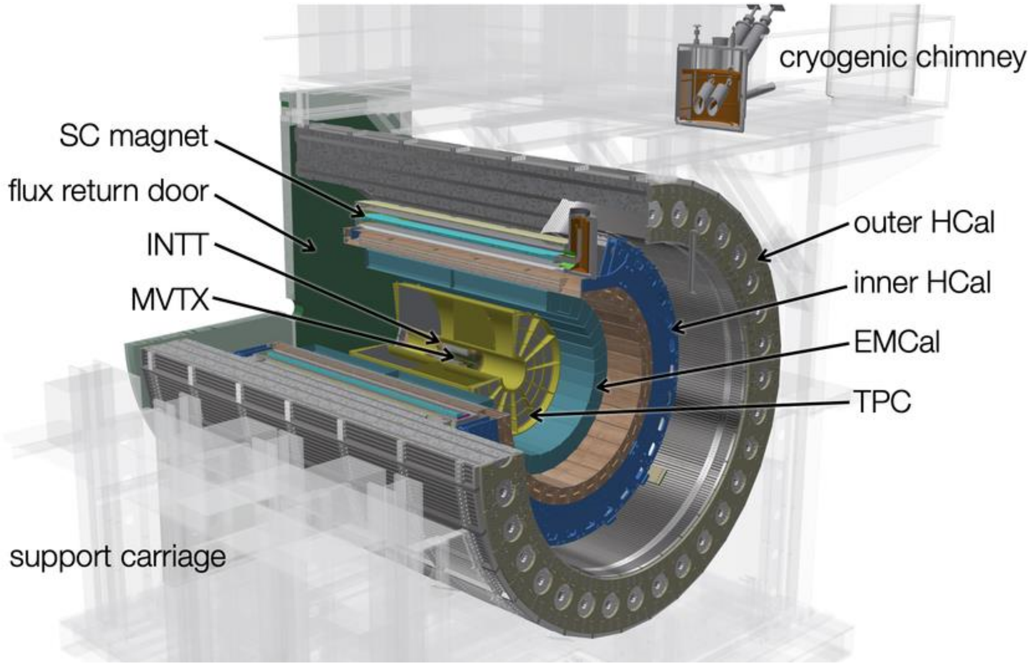


**Real-time AI**

# sPHENIX Experiment at Relativistic Heavy Ion Collider

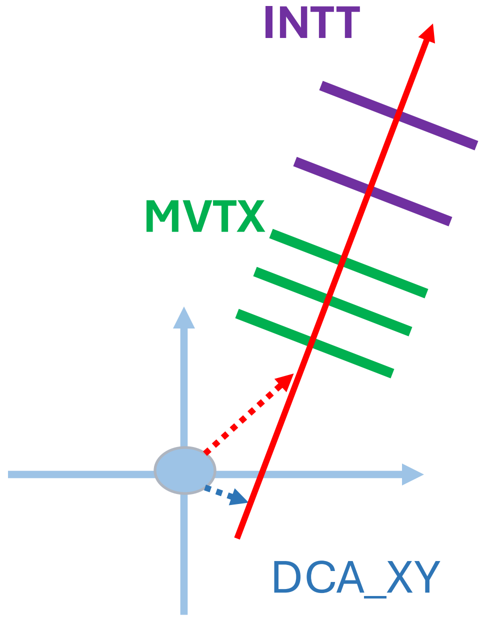
- Located at RHIC (BNL)
- Running period 2023-2025+
- Main detectors: tracking detectors (MVTX, INTT, TPC), calorimeters (EMCal, HCal)
- Hybrid trigger scheme
  - Tracking detectors support streaming readout
    - DAQ limited to ~300Gb/sec
  - Calorimeters readout is trigger-based: 15kHz event rate





# A Test Case: Tag Rare Heavy Quark Events in Real-Time

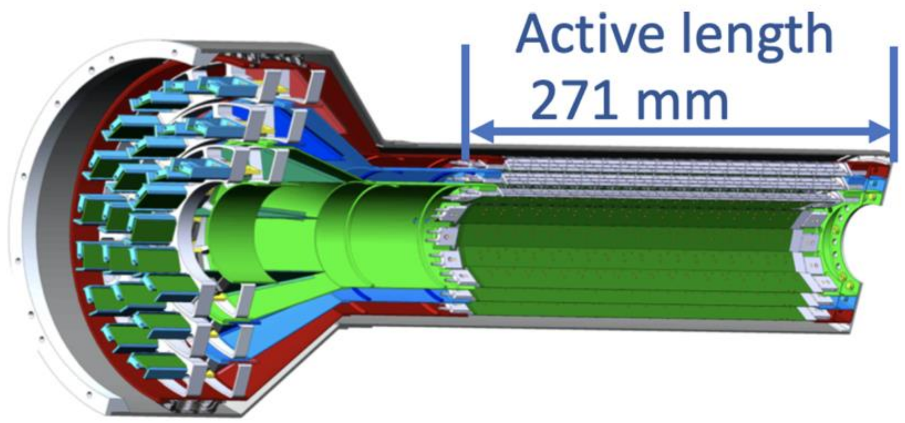
- *High p+p* collision rate ~2MHz, a lot of data!
  - Charm quark production: ~ 30 kHz
    - 500 $\mu$b/42mb ~1%
  - Beauty quarks: ~ 150 Hz
    - 2 $\mu$b/42mb ~0.005%
  - sPHENIX DAQ trigger rate: <15 kHz
    - Tracking detectors are Streamed Readout (SRO) capable
    - Limited DAQ bandwidth prohibits taking all TPC raw data in full streaming mode
      - TPC working in trigger + extended readout mode(~20us), ~O(10%) of MB collisions
    - MVTX and INTT, full SRO in p+p run



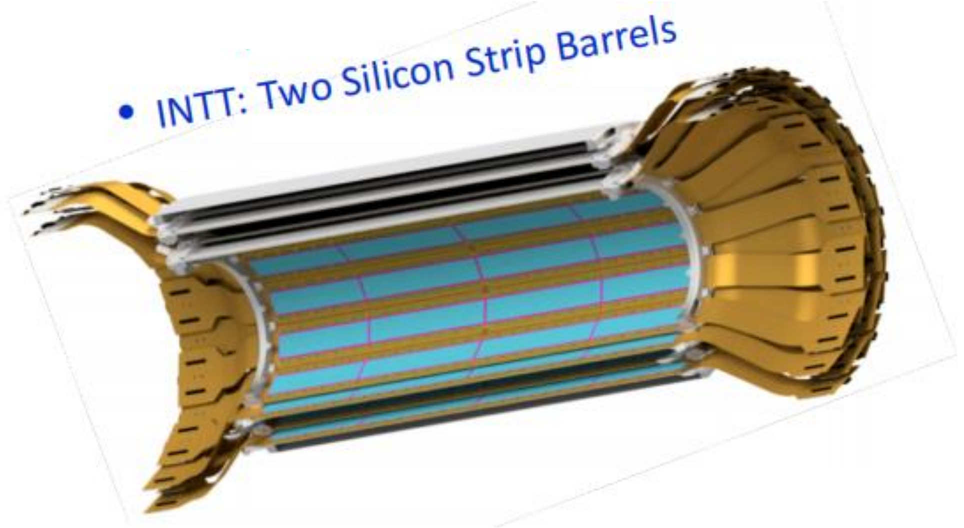
- A real-time ML trigger system aiming to tag HF events with minimal impacts on overall data throughput, with high purity and efficiency
  - MB trigger highly pre-scaled, <0.5% total events (~10kHz/2MHz)

# MVTX and INTT: full streaming readout

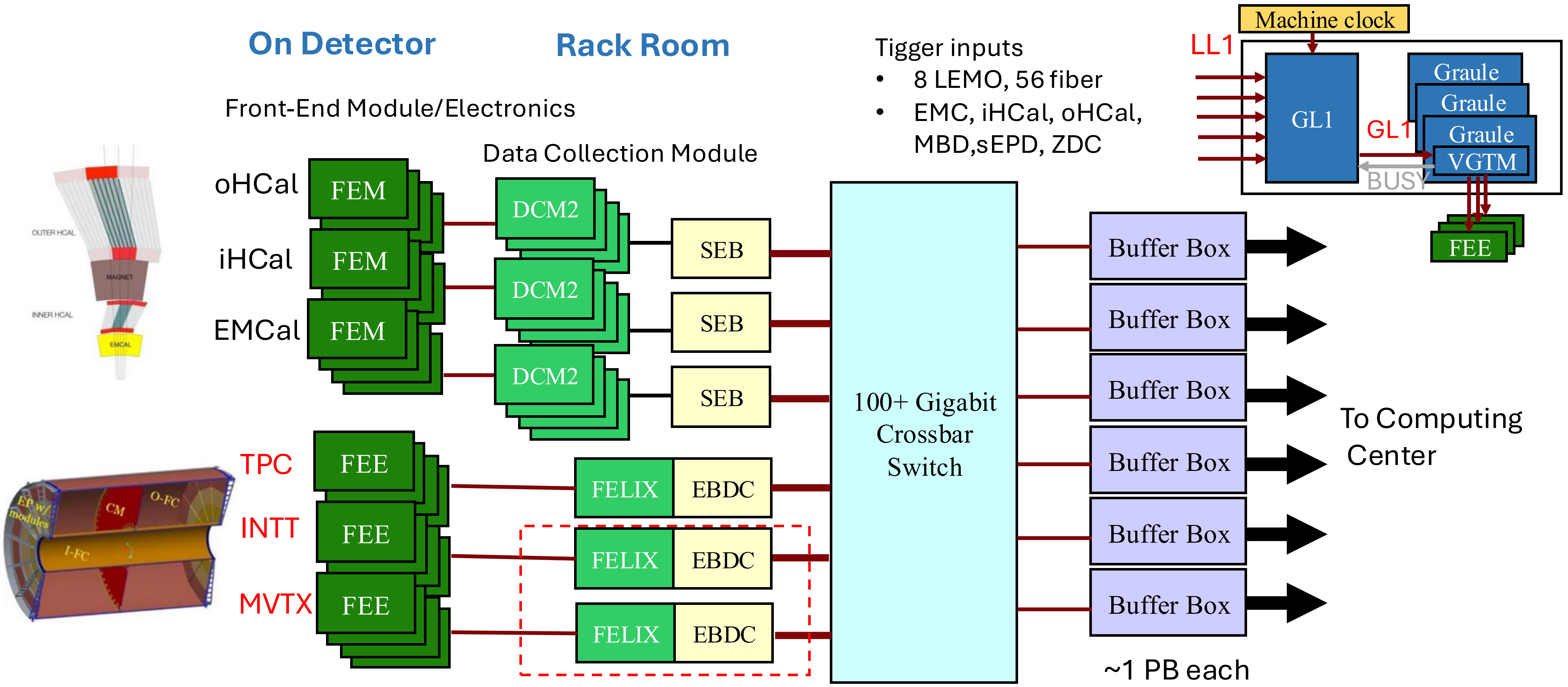
- ❑ MVTX – Monolithic-active-pixel-sensor based vertex detector
  - ➢ Pitch: 27 μm × 29 μm
  - ➢ Time resolution:  5 μs
  - ➢ 3 layers, 48 staves: ~230M pixels channels



- ❑ INTT – micro-strip tracking detector
  - ➢ Pitch: 27 μm × 16 (or 20) mm
  - ➢ Time resolution:  ~ 50 ns (< BCO 106ns)
  - ➢ 2 layers, 56 ladders



# sPHENIX Readout and Trigger Distribution



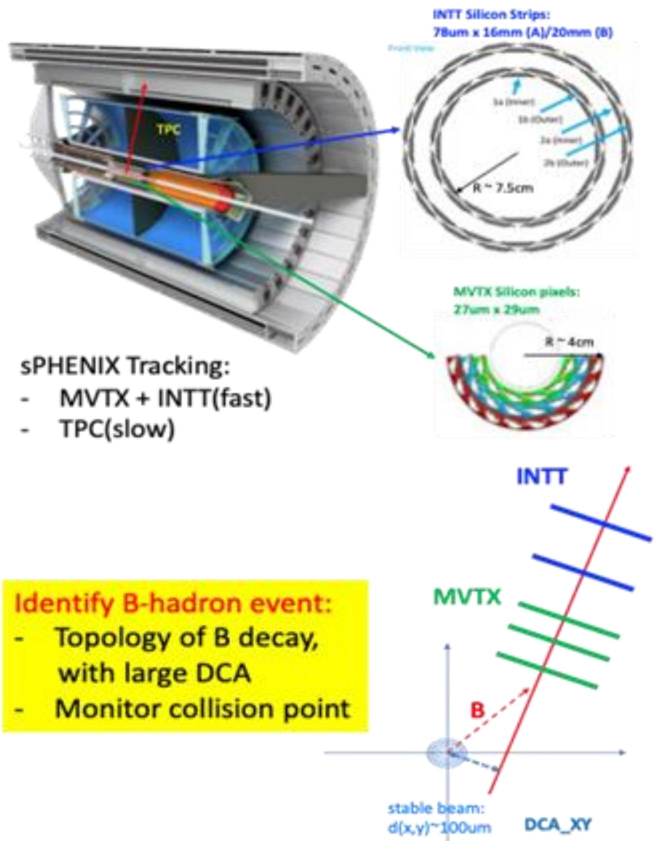
SRO(MVTX+INTT) -> ML -> HF Trigger

# Our Playground

*- Heavy flavor event AI-trigger demonstrator in sPHENIX*

***Two half-barrels for trigger decisions***

**Selective streaming real-time AI and autonomous detector control:**
Deliver a demonstrator for p+p and p+A running for sPHENIX - generalizable for applications in experiments at the EIC

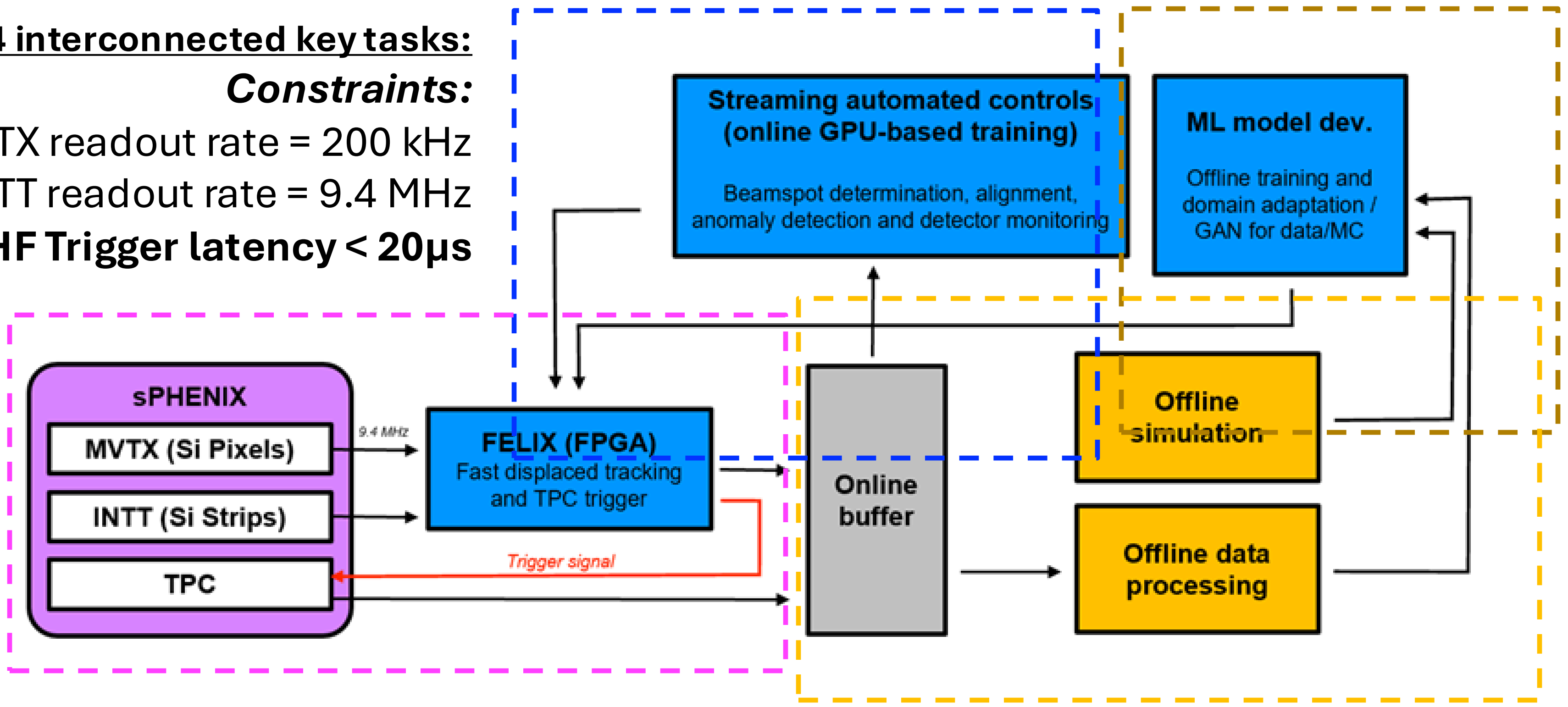


sPHENIX Tracking:
- MVTX + INTT(fast)
- TPC(slow)

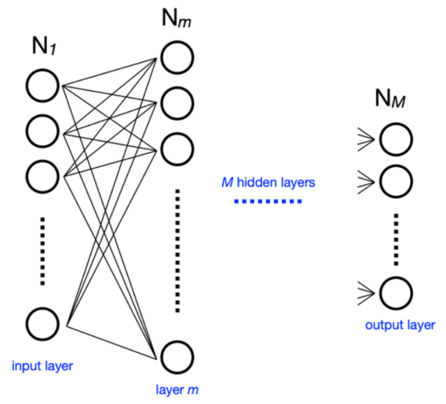
Identify B-hadron event:
- Topology of B decay, with large DCA
- Monitor collision point

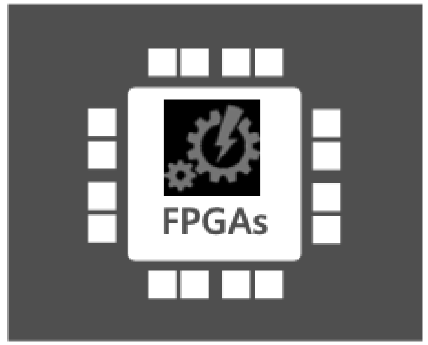
<u>**4 interconnected key tasks:**</u>

***Constraints:***
MVTX readout rate = 200 kHz
INTT readout rate = 9.4 MHz
**HF Trigger latency < 20μs**


sPHENIX
MVTX (Si Pixels)
INTT (Si Strips)
TPC
9.4 MHz
FELIX (FPGA)
Fast displaced tracking and TPC trigger
Trigger signal
Online buffer
Streaming automated controls (online GPU-based training)
Beamspot determination, alignment, anomaly detection and detector monitoring
ML model dev.
Offline training and domain adaptation / GAN for data/MC
Offline simulation
Offline data processing

# 3 Major Areas of R&D





- ❑ Physics/detector simulations and AI/ML algorithm development
- ❑ Translate AI/ML algorithms into hardware language FPGA code with (1) data processing latency and (2)hardware resource constraints
- ❑ Deploy FPGA algorithms in a demonstrator system in sPHENIX

* Good lessons learned from sPHENIX operation with real beam, p+p, Au+Au in 2023-2025

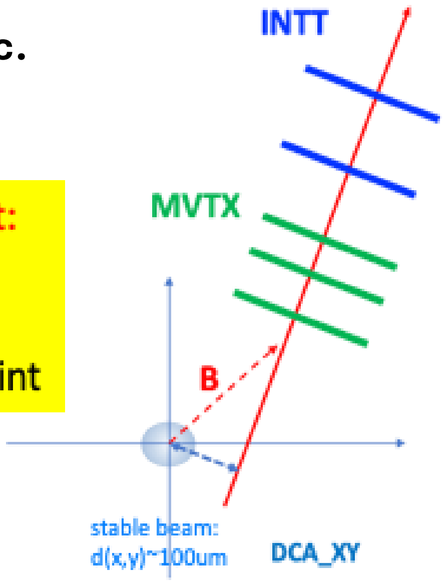
Next – EIC, early 2030s

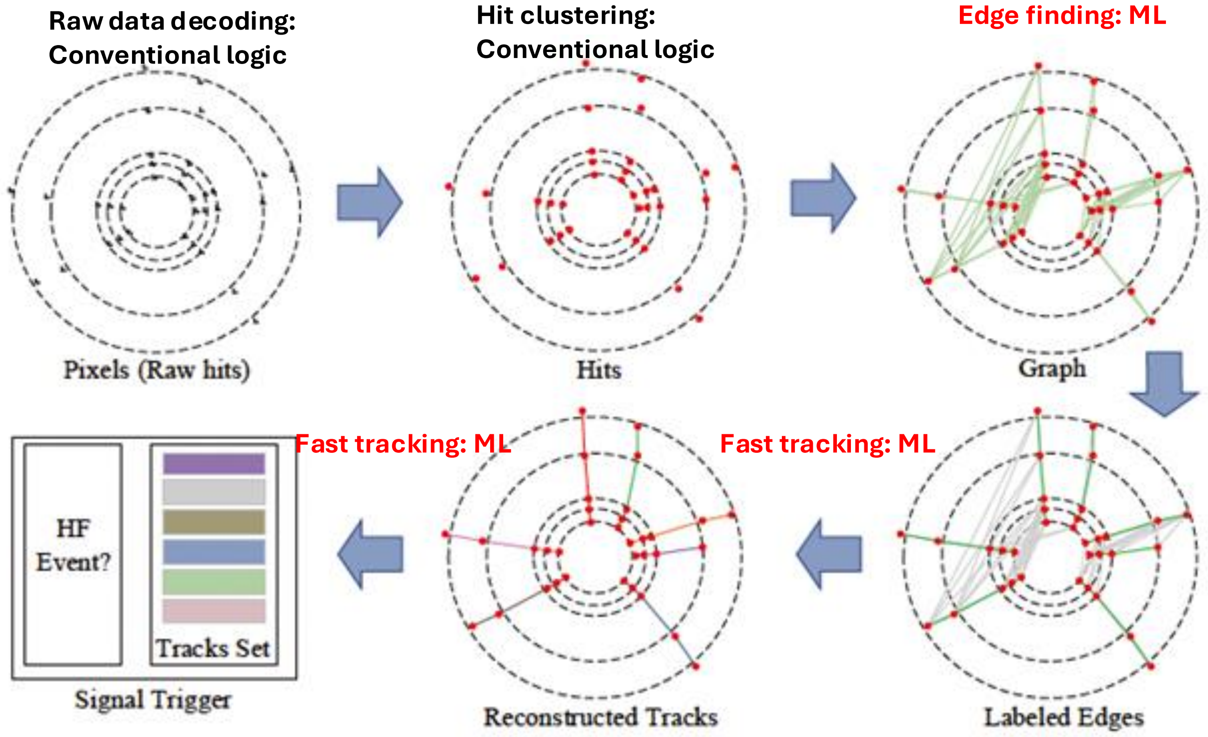
# (I): GNN based Real-Time HF Trigger on FPGA

- AI HF-Trigger algorithms not sensitive to small changes in IP

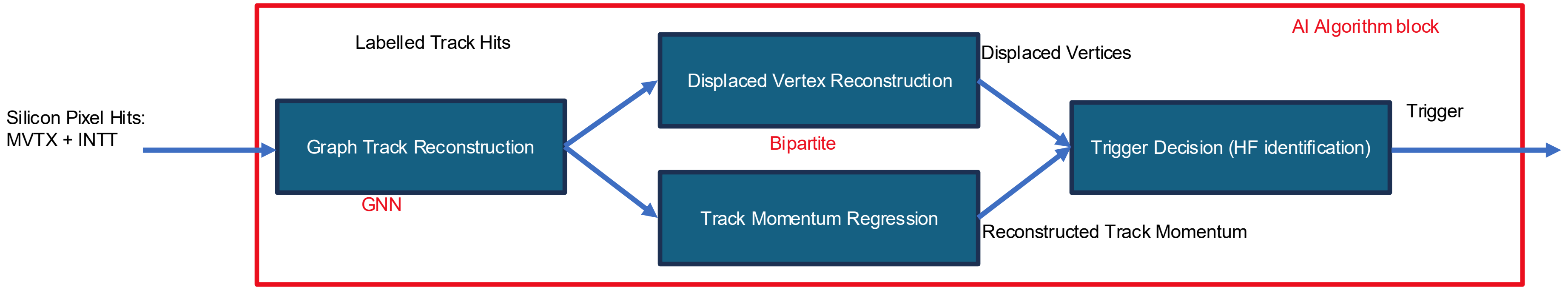
**Features: DCA, pT etc.**

Identify B-hadron event:
- Topology of B decay, with large DCA
- Monitor collision point





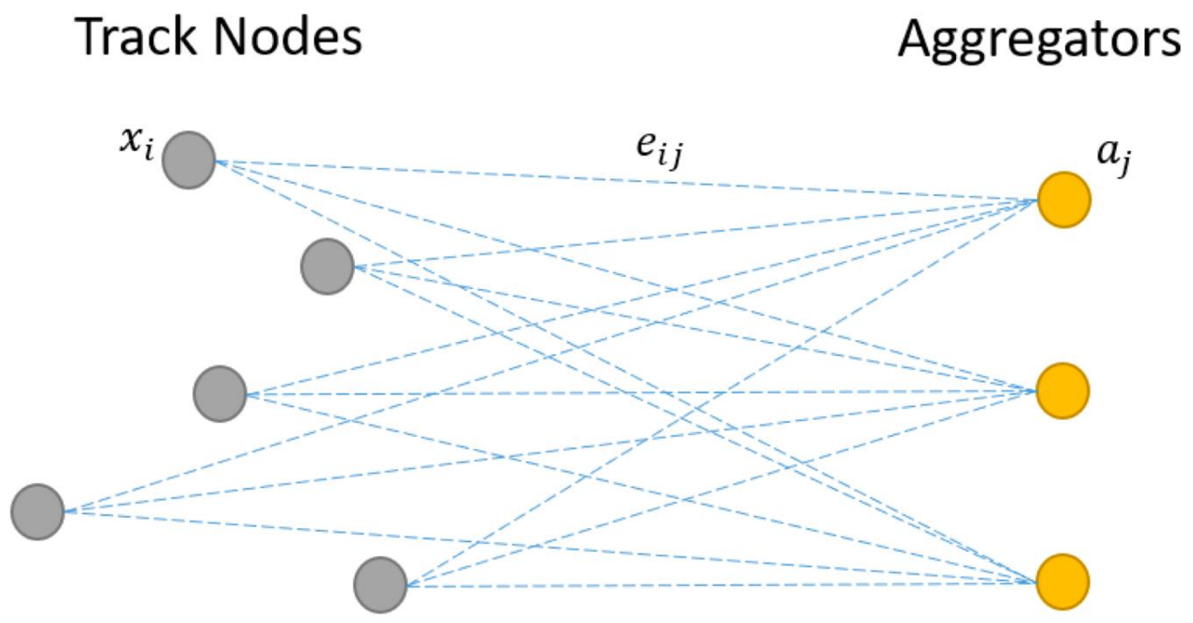
- Two stages of pipeline:
  - Stage 1: Tracking
    - Connect hits left by the same particle to create tracks
    - Reduce data size by eliminating hits left by pileup events
  - Stage 2: Trigger decision
    - Given tracks, predict whether the event is a HF event


Silicon Pixel Hits: MVTX + INTT
Labelled Track Hits
AI Algorithm block
Graph Track Reconstruction
GNN
Displaced Vertex Reconstruction
Bipartite
Track Momentum Regression
Displaced Vertices
Reconstructed Track Momentum
Trigger Decision (HF identification)
Trigger

# HF Tagging with Machine Learning

## Graph Neural Network design

- Track node input vectors
  - 5 hits (MVTX + INTT)
  - Length of each segment: $L = |\overrightarrow{x_{i+1}} - \overrightarrow{x_i}|$
  - Angle between segments
  - Total length of segments
- Aggregators
  - Primary vertex
  - Secondary vertex
- Current ML tracklet algorithm
  - Accuracy > 91% for building tracks
  - Area Under receiver-operating characteristic Curve (AUC) > 97%
  - Excellent signal purity and background rejection



$e_{ij} = s_{ij}x_i$ is track-aggregator messages
$s_{ij}$ is the weight

ECML PKDD 2022, Sub 1256

## Trigger Performance Metric

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

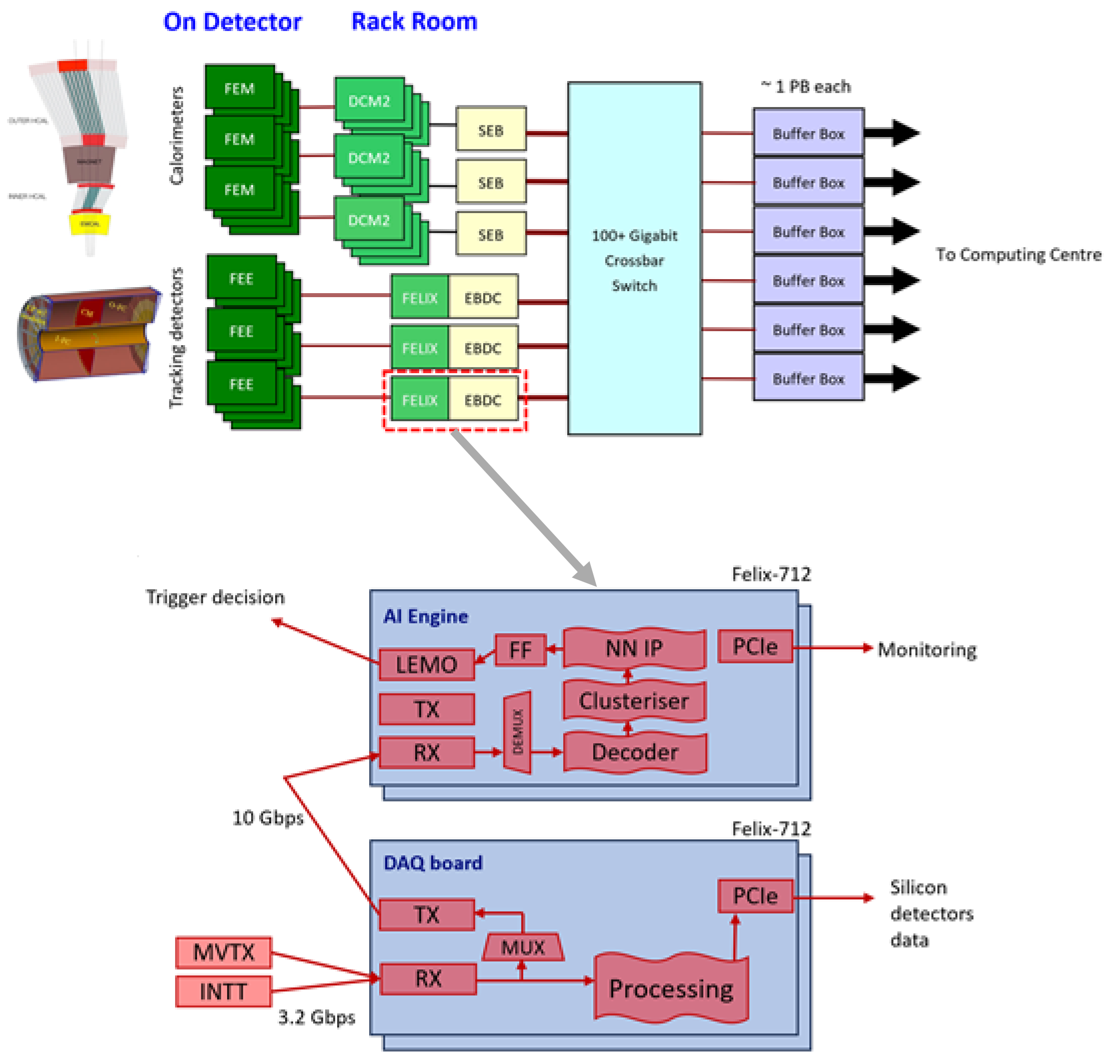
- ❑ Edge candidates are created from hits using geometric criteria
  - ➢ Geometric criteria produces roughly O(n) hits, even with pileup date (usually ~2x as many edge candidates as there are hits)
- ❑ GNN classifies edge candidates based on hits information
- ❑ GNN also performs tracking de-pileup using fast INTT hits
- ❑ GNN trained to prioritize preserving edge candidates arising from trigger particles

An efficient, low-parameter counts, FPGA-ready effective tracking algorithm

| Model Configuration | Precision | Recall (Efficiency) | F1-Score (~purity*Eff) |
|---|---|---|---|
| No Pileup | 92% | 90% | 91% |
| No Pileup, FPGA-ready | 79% | 87% | 83% |
| Pileup (20) | 80% | 73% | 76% |

# (II) Readout and HF AI-Trigger Implementation in FPGA

- The sPHENIX tracking detectors use FELIX-712 PCIe-based boards
  - Contain an AMD/Xilinx Kintex UltraScale FPGA (xcku115-flvf1924-2-e)
- To the readout DAQ boards, add AI Engine boards to perform the B-tagging using AI (FELIX-712)
- Exploring implement graph neural networks (GNNs) with two approaches:
  - FlowGNN (arXiv: 2204.13103)
  - hls4ml (arXiv: 1804.06913)



## The Latency Constrains for ML-base Algorithm

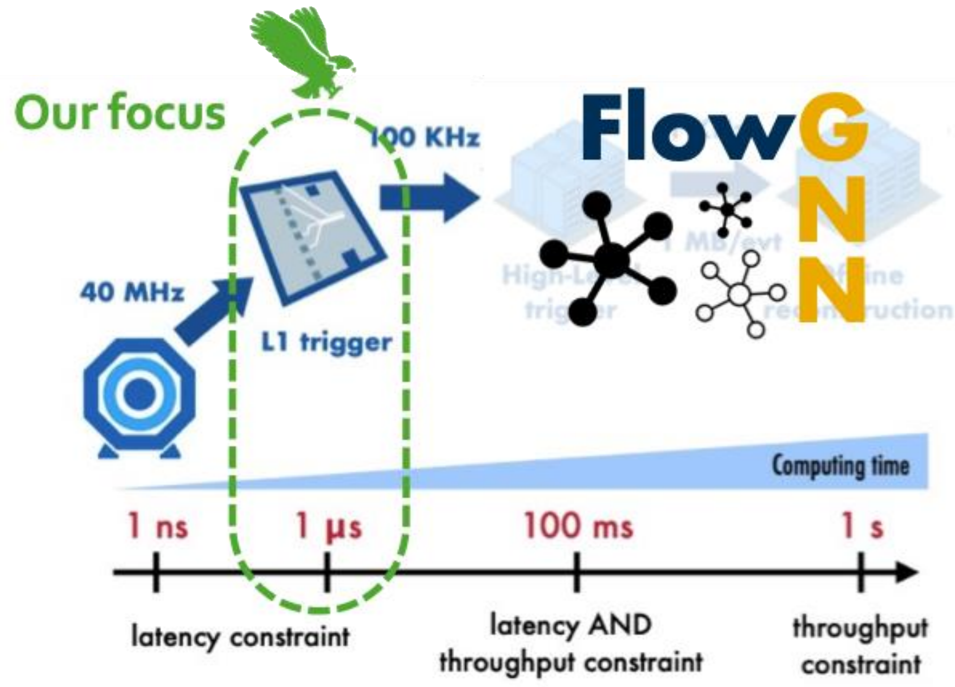
- The TPC buffer can hold up ~30 us of data before receiving a readout trigger
- Detector readout delay, fiber transmission delay, data encoding/decoding
    - MVTX readout window ~8us
    - Interaction Region (IR) ->Counting house ~0.3 us (100 m cables)
    - FELIX data forward, decoder buffers ~0.6 us (@240 MHz)
    - Global level 1 Trigger decision latency + counting house -> IR ~0.3 us
    - … **~10 us**
- The goal is to achieve **~10 us** latency for the trigger algorithm

# Approach 1: Flow-GNN

- FlowGNN is a flexible architecture for GNN acceleration on FPGAs, https://arxiv.org/abs/2204.13103
- Two manual implementations, from PyTorch → C++ → Verilog, using High Level Synthesis (HLS)
  - Version 1: Track construction only:
    - 8.82 us per graph (Freq. 285 MHz), tested with: 92 nodes, 142 edges
  - Version 2: from Hits -> Clustering → Triggering:
    - 9.2 us per graph (Freq. 180 MHz), Tested with: 92 nodes, 142 edges

- In progress:
  - Extending to support more types of GNNs, e.g., EdgeConv, to facilitate better algorithm support
  - Perfecting the automation flow from PyTorch → Verilog, based on GNNBuilder, https://arxiv.org/abs/2303.16459

**Co-design:**
- Algorithms
- FPGA

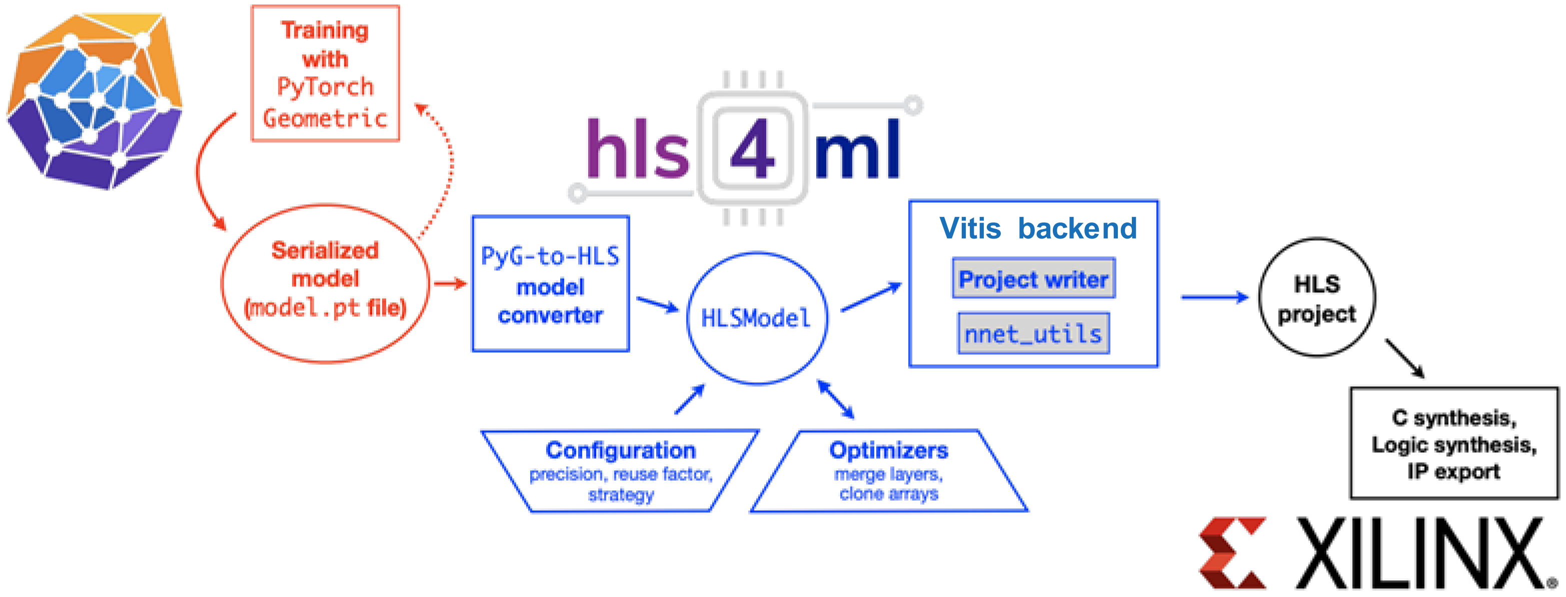


# Approach 2: *hls4ml*

- hls4ml (arXiv: 1804.06913)

❑ ***hls4ml*** is a HEP community developed compiler taking Keras, Pytorch, or ONNX input and producing High Level Synthesis (HLS) code implementing the network as spatial dataflow.
  - ➢ HLS code is usually C++ or similar with directives to guide the produced hardware.
  - ➢ hls4ml has different "backends" for the different flavors of HLS desired by tools.

❑ GNN support is under development: currently the process is not as automated as for other network types, manually implemented a simpler model, hits -> trigger

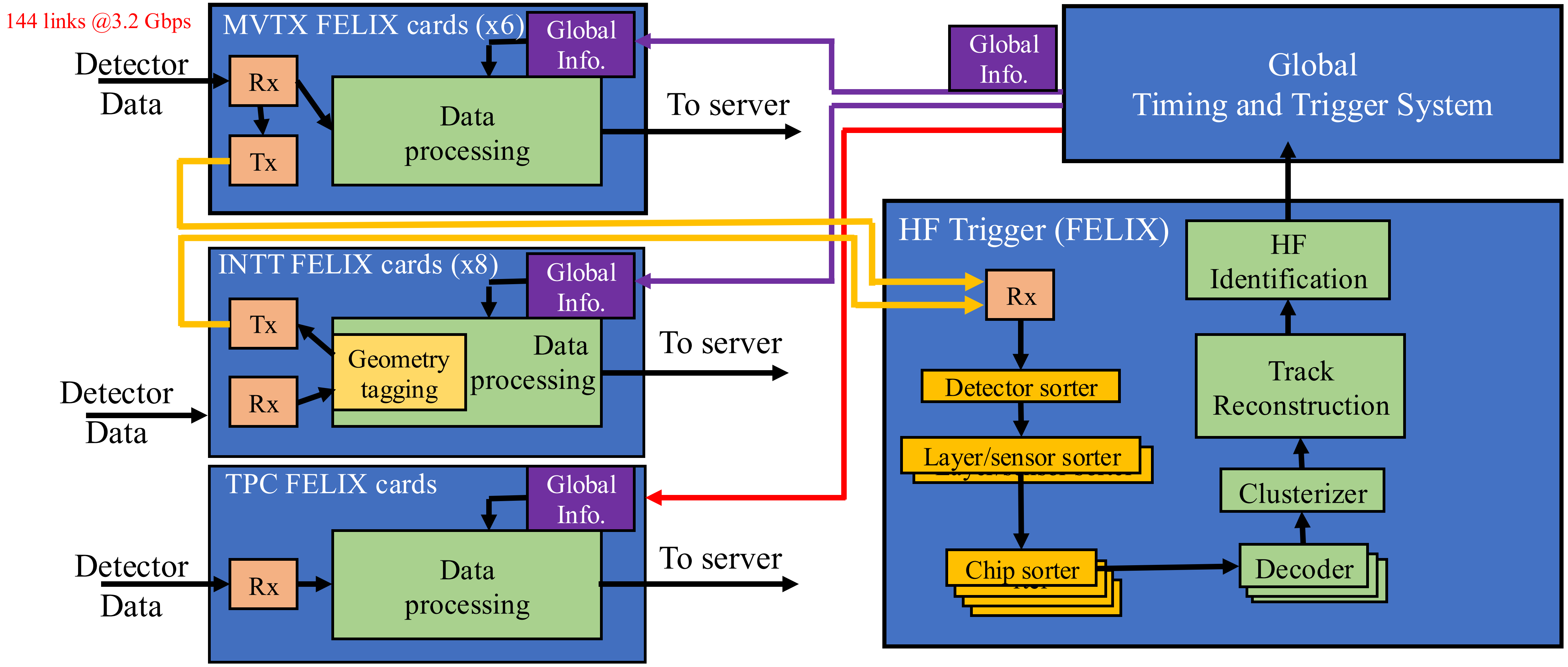

Training with PyTorch Geometric
Serialized model (model.pt file)
hls 4 ml
PyG-to-HLS model converter
HLSModel
Vitis backend
Project writer
nnet_utils
HLS project
Configuration
precision, reuse factor, strategy
Optimizers
merge layers, clone arrays
C synthesis, Logic synthesis, IP export
XILINX

# *hls4ml* Initial Implementation (MVTX-only MLP)

- **The MLP-layerwise model has been synthesized for the FPGA**
- The model consists of two parts
  - The first part, called the **aggregation step**, collects all the clusters. It is called for each cluster in a bunch crossing. This needs a high throughput: initiation interval every 1 clock cycle, 117 ns latency
  - The second part, called the **prediction step**, is called once per bunch crossing, to make a prediction based on the ingested clusters: 63 clock cycles, 308 ns latency
- The two models are synthesized separately, with the FPGA utilization for the FELIX 712 given below, using Vitis HLS and Vivado 2024.1.

| | Aggregation step | Prediction step |
|---|---|---|
| LUT | 23 587 (3.56%) | 16 582 (2.50%) |
| FF | 15 129 (1.14%) | 31 226 (2.35%) |
| DSP | 19 (0.34%) | 498 (9.02%) |
| BRAM | 0 (0%) | 30.5 (1.41%) |

# (III) HF Trigger System Diagram

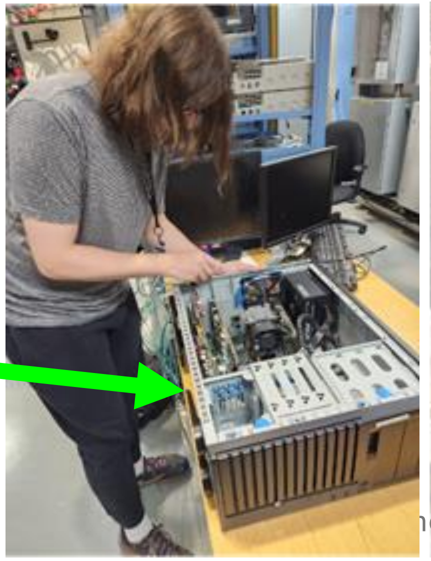


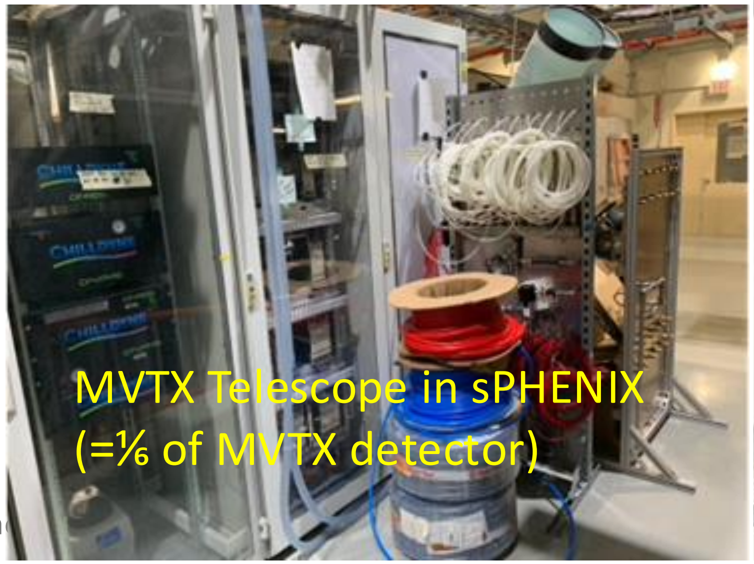
# Smaller Scale Demonstrator:

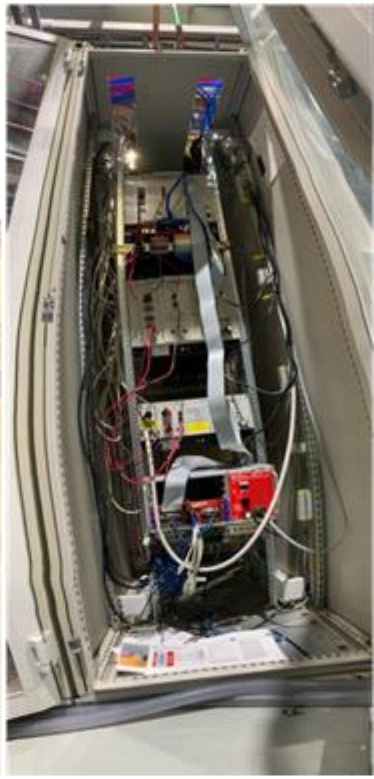
- with MVTX Telescope Communication

*Due to very tight sPHENIX operation schedule and certain detector challenges, we didn't get the opportunity to integrate AI/ML system into the sPHENIX DAQ, instead, used MVTX telescope in the sPHENIX counting house for the system test*

- ❑ FELIX-712 was designed as sPHENIX readout board, the PCIe is used to receive data from the optics
  - ➢ Save the timing (Bunch Crossing ID) and trigger decision from the AI
  - ➢ Configured the PCIe uplink (normally used just for configuration) to load real detector data to the board, for a controlled validation environment
- ❑ Successfully received and decoded data from single stave of the MVTX 8-stave telescope (MVTX = 6 x Telescope)
- ❑ Added ILA via Xilinx virtual cable for additional debugging and monitoring

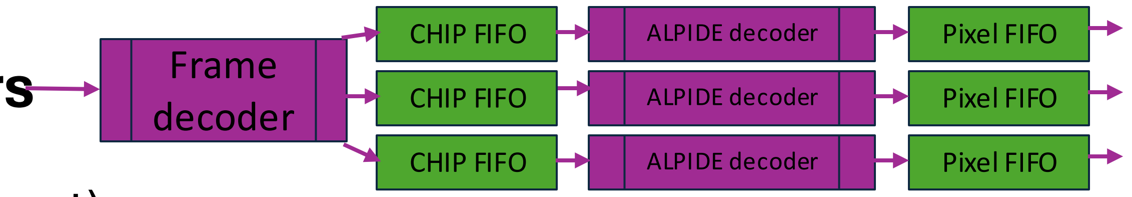






# MVTX Decoder Development (Conventional)

- **First FPGA-based decoder for ALPIDE sensors**
  - The design has been simplified
    - There is only one set of buffers (instead of per event)
  - The design was validated on simulation, PCIe and Telescope data
    - This also helped to validate the PCIe and Telescope comms
  - Due to MVTX data compression we need 1 decoder module per detector (FeeID) link (144 total)



| | LUT (663K) | FF (1.3M) | BRAM (2K) |
|---|---|---|---|
| Frame decoder | 151 | 287 | 0 |
| ALPIDE decoder (x3) | 343 | 256 | 0 |
| FIFOs (x6) | 31 | 36 | 1 |
| Total per FeeID | 1366 | 1271 | 6 |
| **Total per half- barrel** | **98K (14.7%)** | **91K (7%)** | **432 (21%)** |

# FPGA Resource Utilization (FLX-712)

green: PCIe
purple: decoder
yellow: clusterizer
turquoise: local to global
brown: hsl4ml aggregate
pink: hsl4ml predict

- Currently we have single stave implementation to validate modules
  - 3 decoders, 1 clusteriser, 1 transformation

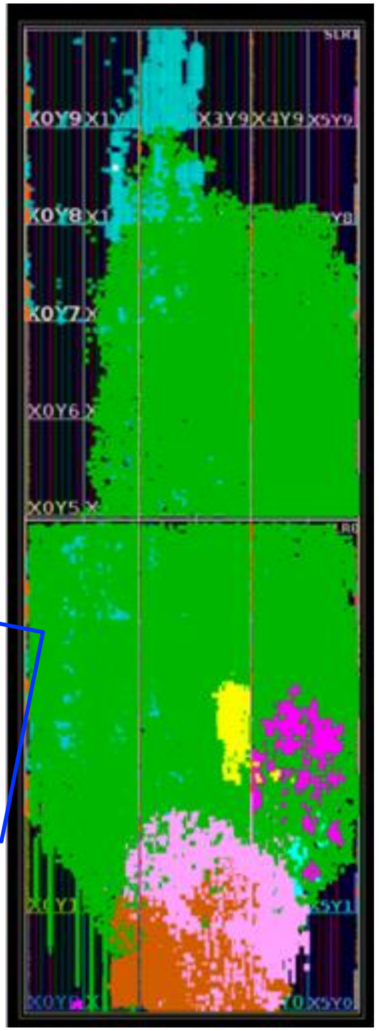
| | LUT (663K) | FF (1.3M) | BRAM (2K) | DSP (5.5K) |
|---|---|---|---|---|
| 1-stave | 163K (24.5%) | 359K (27.6%) | 1K (50%) | 525 (9.5%) |
| 8-staves | 232K (35%) | 412K (31.6%) | 1.2K (60%) | 581 (10.5%) |

- Target is 72 decoders, clusterisers, and transformations
  - Current projection:

**New FLX-155 ~ 3x FLX712**

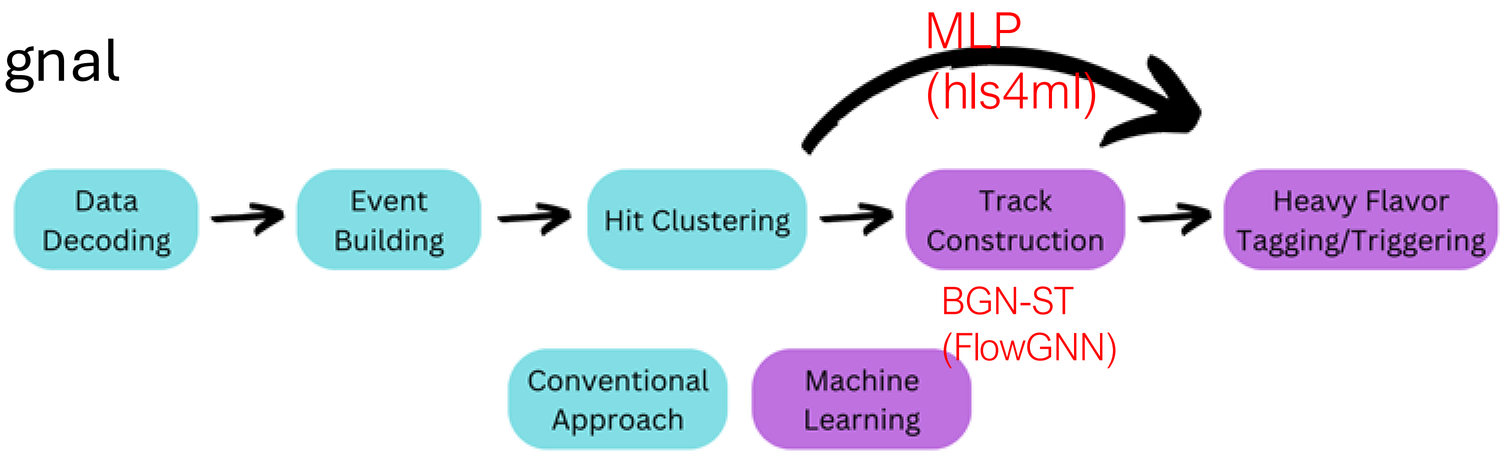
| | LUT (663K) | FF (1.3M) | BRAM (2K) | DSP (5.5K) |
|---|---|---|---|---|
| Infrastructure | 87K (13.1%) | 196K (14.8%) | 879 (40%) | - |
| Decoder | 98K (14.7%) | 91K (7%) | 432 (21%) | - |
| Clustering | 267K (40%) | 213K (16.4%) | - | - |
| Transformation | 25K (3.8%) | 22K (1.7%) | 540 (27%) | 576 (10.4%) |
| AI module (FlowGNN) | 194K (29%) | 214K (16.4%) | 406 (20%) | 488 (8.8%) |
| AI module (hls4ml) | 40K (6.1%) | 45K (3.5%) | 31 (1.5%) | 517 (9.4%) |

**Need to reduce for FLX-712;**
**OK for FLX-155**



## FPGA Ready Algorithm Summary: It is doable!

- ❑ Hit decoding and clustering - conventional algorithms
- ❑ *Event building, collect hits from the same collisions – MVTX(slow) + INTT(fast)*
- ❑ Track reconstruction - using GNNs in two parts
  - ➢ Edge candidate generation - connect clusters (nodes) with edges, with geometric constraints
  - ➢ Edge candidate classification - using graph convolutional network (GCN) (arXiv: 1609.02907)
  - ➢ Construct final tracks
- ❑ Use a least squares method to perform $p_T$ prediction from track curvature
- ❑ Tagging of the heavy flavor signal



***Also an alternate implementation, taking the clusters directly without explicit track reconstruction.***

# EIC – be prepared for unexpected

- lessons learned from sPHENIX data taking and implications for future EIC and other experiments

EIC
A NEW ERA OF DISCOVERY
THE 2023 LONG RANGE PLAN FOR NUCLEAR SCIENCE
A NEW ERA OF DISCOVERY
THE 2023 LONG RANGE PLAN FOR NUCLEAR SCIENCE

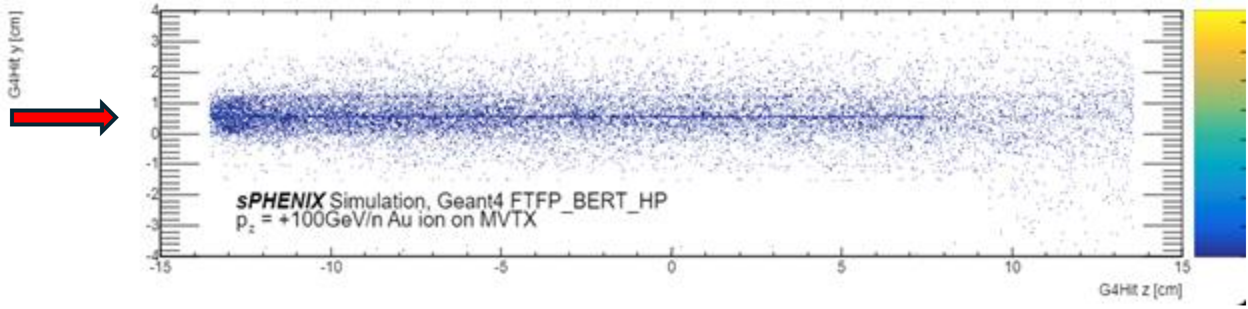
***New ideas being developed to address new challenges …***

# Unexpected Challenges First Observed in sPHENIX 2023 Au+Au Runs

- Full streaming readout in high beam backgrounds!

Expected hits: O(10s) out of ~1M pixels/stave

Beam halo background hits: >> O(10k)



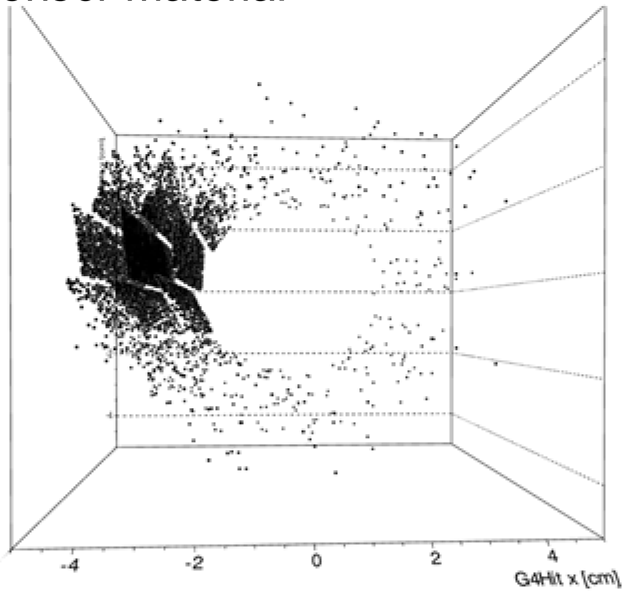
- ❑ **Major beam-related background with Au beam**
  - ➢ Related to beam halo induced particles hitting large number of sensor pixels in the MVTX detector sensors

  **NO problem in p+p collisions**

**Data >> DAQ bandwidth! (>10^3)**

GEANT Simulation:

Single 100 GeV Au ion striking the end of the 50um thick MVTX silicon sensor material



EIC: day-1, e+A program

Could face similar high backgrounds with ion beam

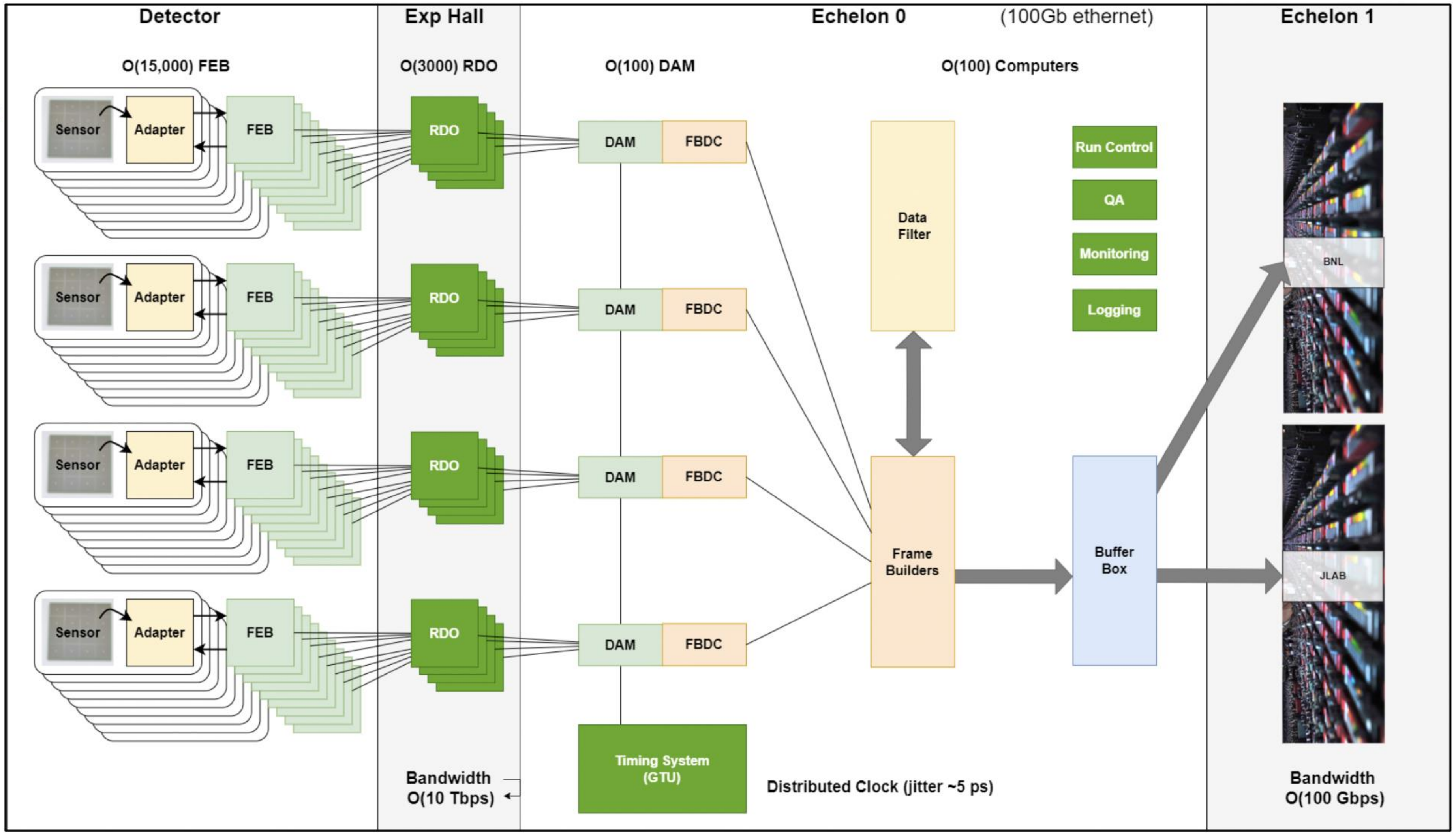
***Smart data management highly desired on/near the detectors for full streaming readout in high background environment***

# EIC SRO … the data throughput challenge

- Bunch Crossing
  ~10.2 ns/98.5 MHz
- Interaction Rate
  ~ 2 us/500 kHz
- Low occupancy

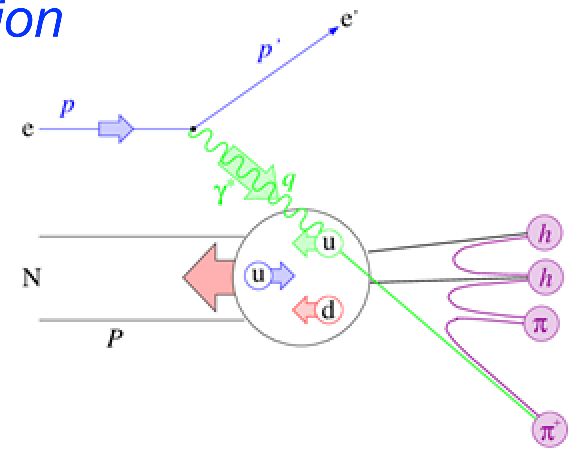
**A big unknown: beam backgrounds, could easily overwhelm the DAQ system!**

**Better be prepared~**



# Fast-ML for EIC – work in progress…

*- DIS-electron identification in real-time with beam background suppression*



**Selective streaming readout for AI-Engine:**

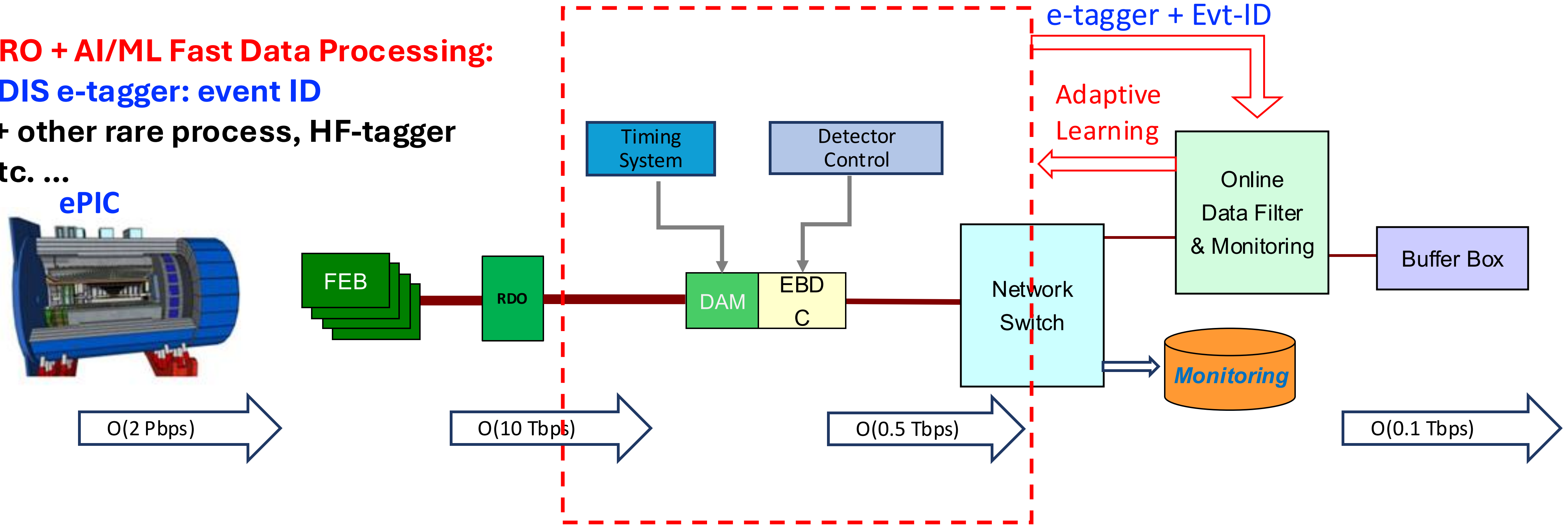
- ❑ tag DIS-electron to define DIS event ID
  - ➢ EMCal + Trker + ePID
  - ➢ DCA~0
- ❑ With AI noise suppression on chip (AI-on-Sensor)!

**Add AI-based active beam halo background rejection: AI-on-Detector!**

**SRO + AI/ML Fast Data Processing:**

**- DIS e-tagger: event ID**

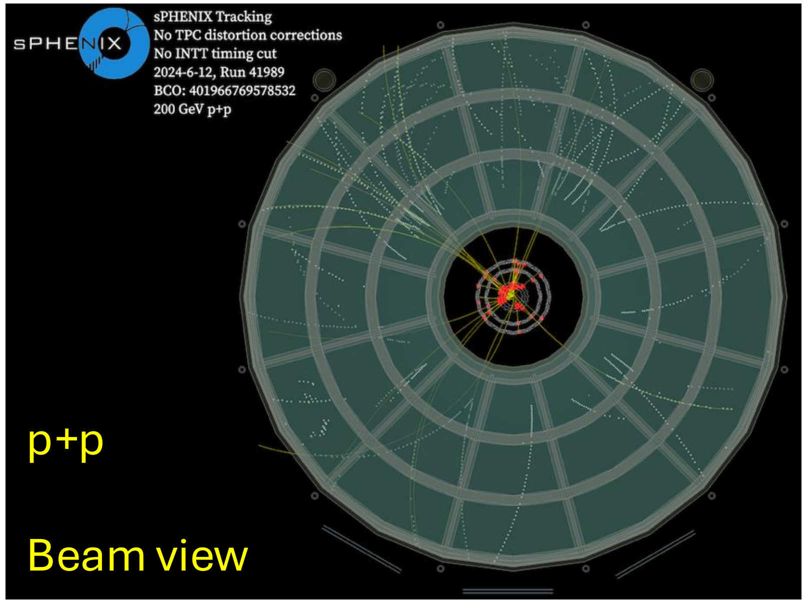
**+ other rare process, HF-tagger etc. …**

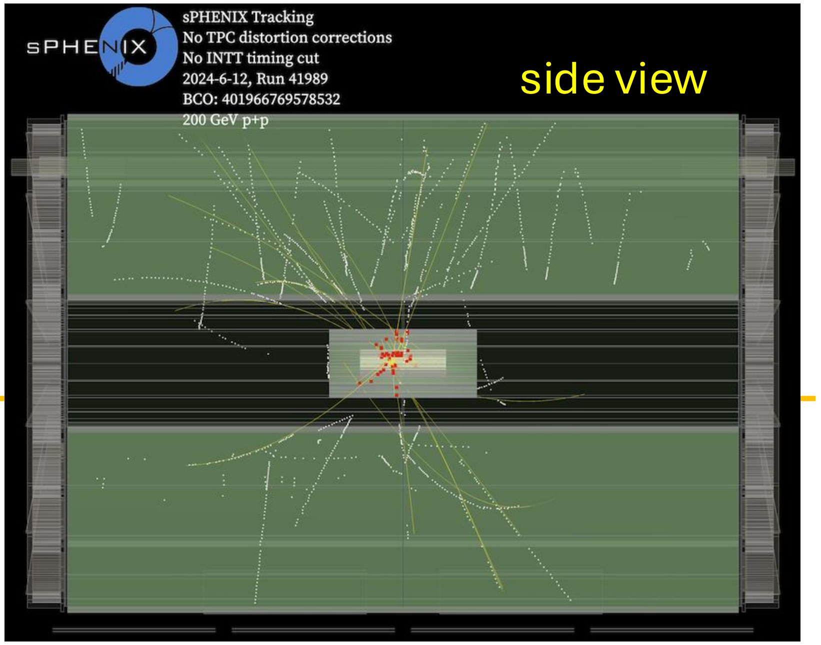


## Backup slides

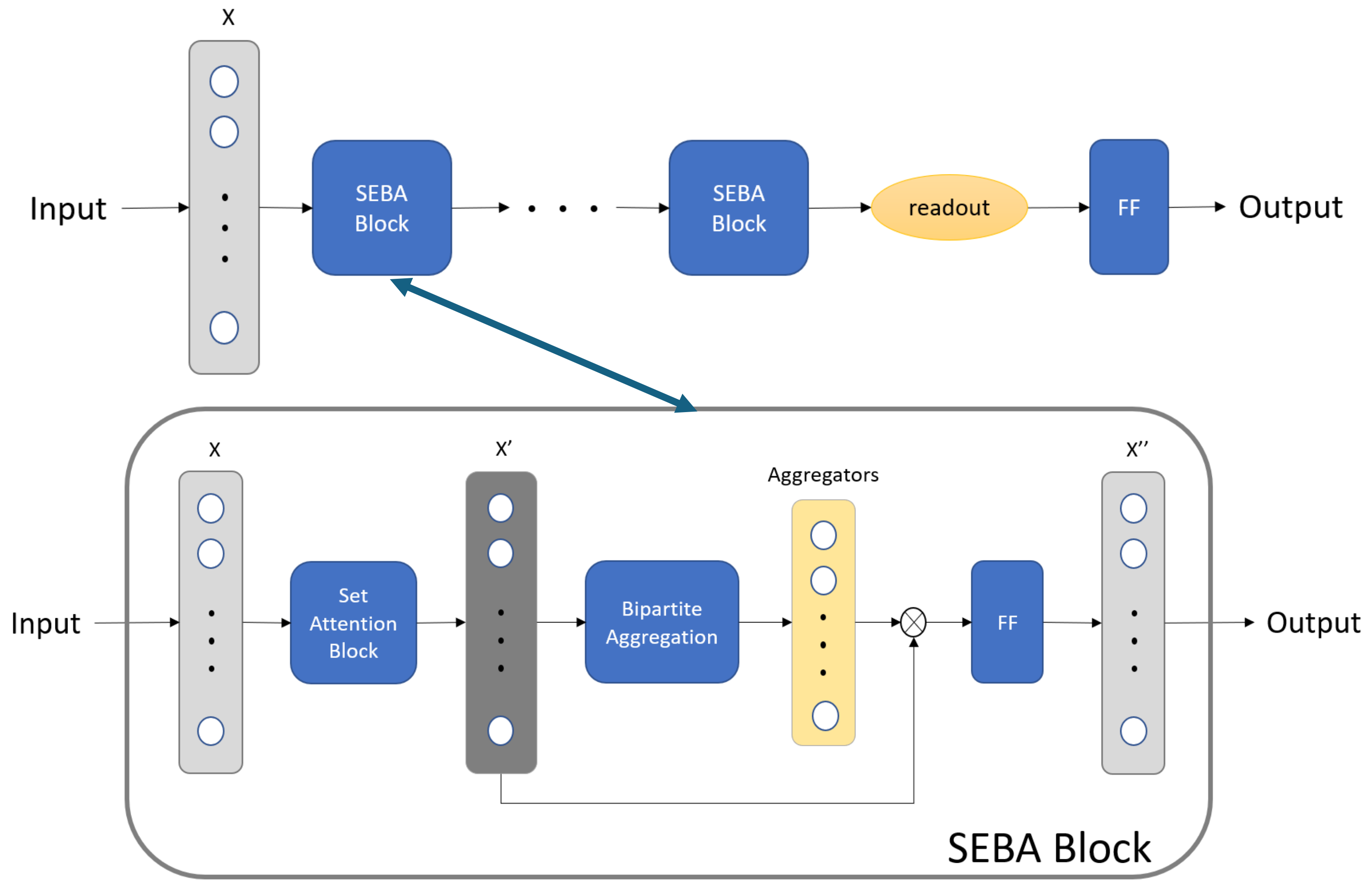
# AI/ML Algorithm Development

- ❑ An efficient, end-to-end, robust trigger pipeline capable of handling multi-collision pileup
  - ➢ pileup of p+p collisions: hits from ~20 events
- ❑ Two stages of pipeline:
  - ➢ Stage 1: Tracking
    - Connect hits left by the same particle to create tracks
    - Reduce data size by eliminating hits left by pileup events
  - ➢ Stage 2: Trigger decision
    - Given tracks, predict whether the event is a HF event
- ❑ Developed algorithm NOT sensitive to the IP variations
- ❑ Improve performance by reinforcing physics laws in the models





# GNNs with Set Transformers



Set Encoder with Bipartite Aggregator (SEBA)

**The cycle**

1. *Track information* is initially defined
2. This is relayed to all primary and secondary vertex information
3. Weights are assigned to each link
4. The PV and SV information go through a FeedForward(FF) NN
5. This updates the track information

# Coordinate Transformation (Conventional)

- The clusterizer provides - layer, stave, chip, row, column (hardware)
- The AI requires - layer, r, phi, z (physics)
- A new transformation module has been created to transform coordinates
- The BRAM usage is quite large
  - Optimize parametrization of the transformation

| | LUT (663K) | FF (1.3M) | BRAM (2K) | DSP (5.5K) |
|---|---|---|---|---|
| Clustering | 347 + 44 (memory) | 310 | 7.5 | 8 |
| per chip (x216) | 75K (11.2%) | 67K (5.1%) | 1620 (81%) | 1728 (31%) |
| **per feeID (x72)** | **25K (3.8%)** | **22K (1.7%)** | **540 (27%)** | **576 (10%)** |
| per stave (x24) | 8.3K (1.2%) | 7.4K (0.5%) | 180 (9%) | 192 (3.5%) |

# Demonstrator Implementation Status

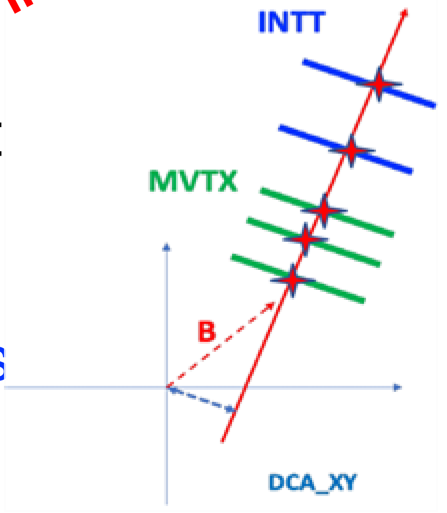
*Two half-barrels*

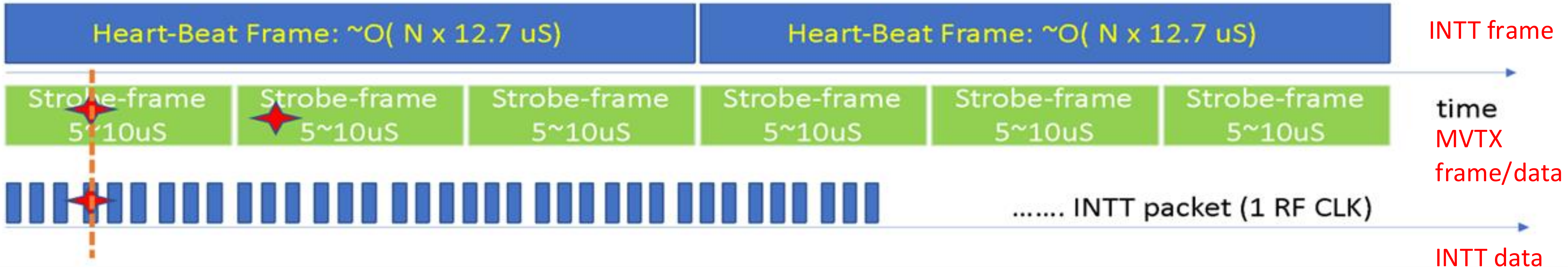
| Module | written | Validated - sim | Validated - test file | Validated - detector |
|---|---|---|---|---|
| PCIe comms | ✅ | ✅ | ✅ | ✅ |
| Optics | ✅ | - | - | ✅ |
| Decoder | ✅ | ✅ | ✅ | ✅ |
| Clusteriser | ✅ | ✅ (C++)<br>Ongoing (VHDL) | ✅ | ✅ |
| Event build and coordinates transform | ✅ | ✅ | Ongoing | |
| AI module | ✅ FlowGNN<br>✅ hls4ml | ✅<br>✅ | Ongoing | |

# Raw Data Pre-processing: Event Building

A big challenge of data integration!

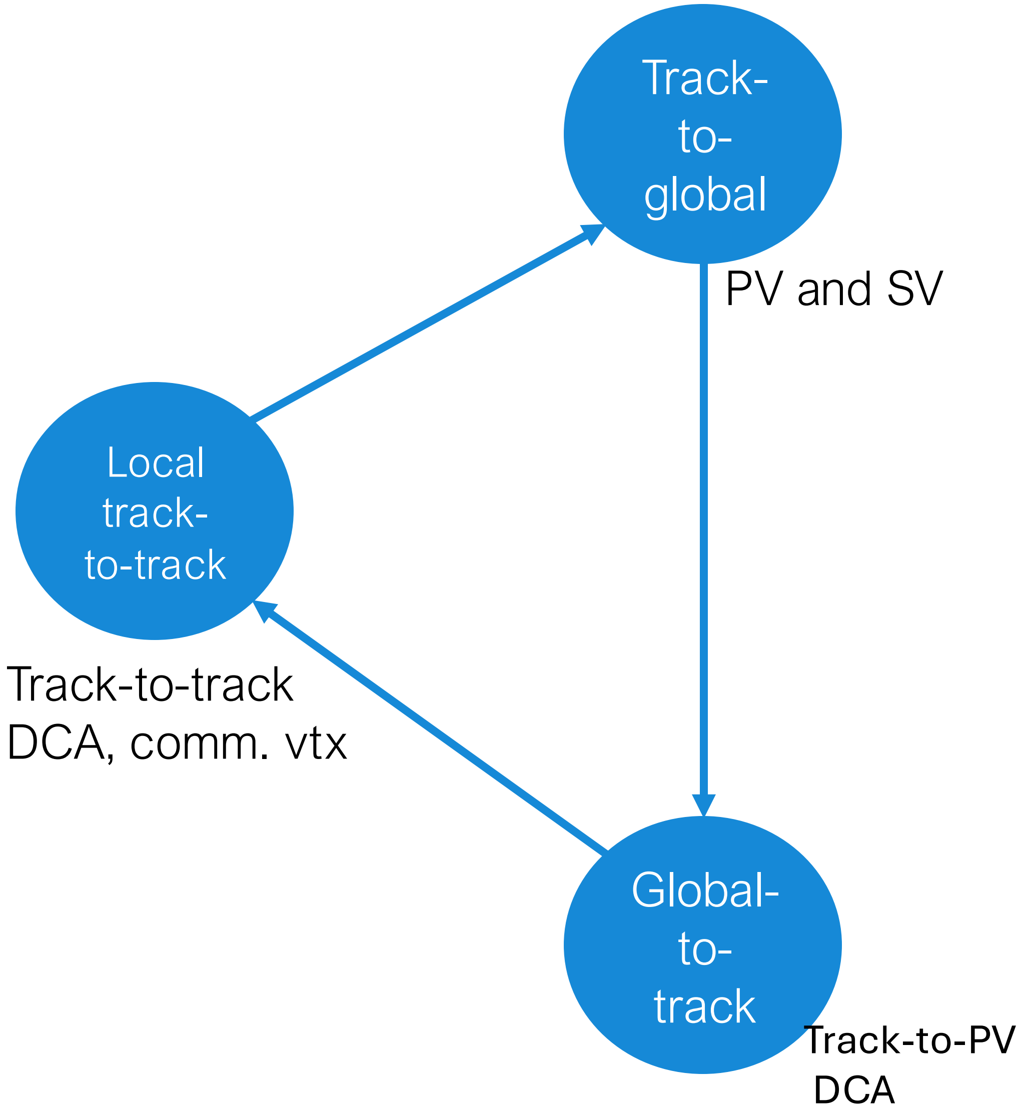
- ❑ With the current MVTX-only setup the event building is easy
  - ➢ Since the detector links contain Bunch Crossing ID we can just read event by event link by link
- ❑ Challenge: once we add INTT stream this will be much more complicated due to different reading stream lengths and latencies
- ❑ Important is to first have the simpler MVTX-only implementation working!





# Feedback Algorithms

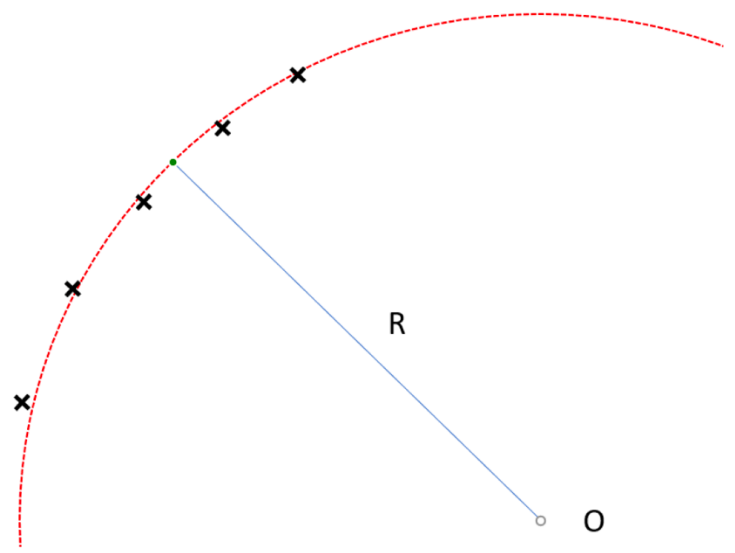
- ❑Tracking algorithms developed using simulated signal and background events in the MVTX and INTT
- ❑Used these models to feed into the models to select interesting events
  - ➢Models are bi-directional, local information is passed to global and global information is passed back to local to refine
- ❑Initial trainings and models are developed on GPU
  - ➢NVIDIA Titan RTX, A5000, and A6000
  - ➢Developed with PyTorch and PyTorch Geometric



## Transverse Momentum pT Estimation

- ❑A feed-forward neural net is used to predict the pT
- ❑Uses least-squares method to estimate track radius
- ❑~15% improvement in tracking with pT estimation

Heavy quark decay ➔ higher pT daughter particles



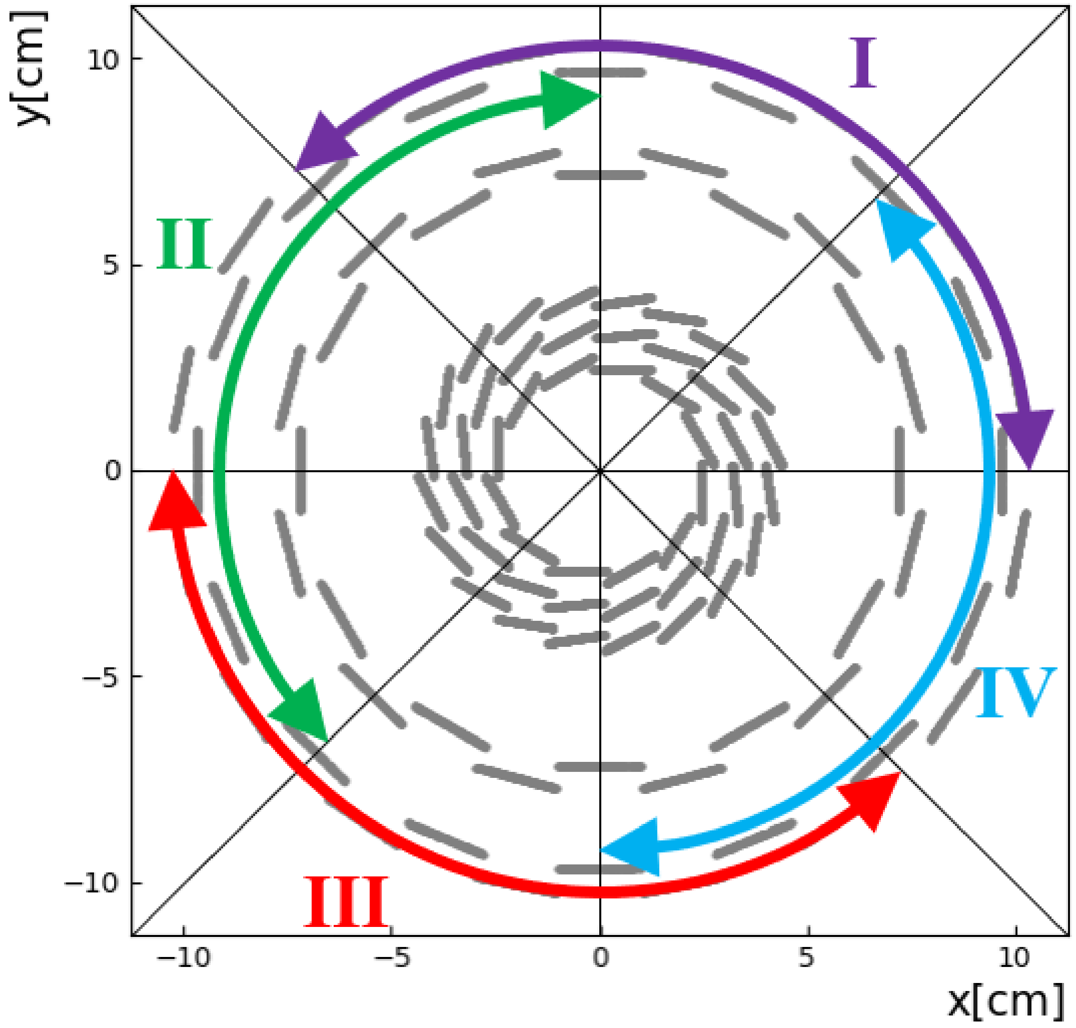
| | with LS-radius | | | without radius | | |
|---|---|---|---|---|---|---|
| Model | #Parameters | Accuracy | AUC | #Parameters | Accuracy | AUC |
| Set Transformer | 300,802 | 84.17% | 90.61% | 300,418 | 69.80% | 76.25% |
| GarNet | 284,210 | 90.14% | 96.56% | 284,066 | 75.06% | 82.03% |
| PN+SAGPool | 780,934 | 86.25% | 92.91% | 780,678 | 69.22% | 77.18% |
| BGN-ST | 355,042 | **92.18%** | **97.68%** | 354,786 | **76.45%** | **83.61%** |

| | LS | | MLP | |
|---|---|---|---|---|
| Hidden dim | Accuracy | AUC | Accuracy | AUC |
| 32 | 91.52% | 97.33% | 91.48% | 97.31% |
| 64 | 92.18% | 97.68% | 92.23% | 97.73% |
| 128 | **92.44%** | **97.82%** | **92.49%** | **97.86%** |

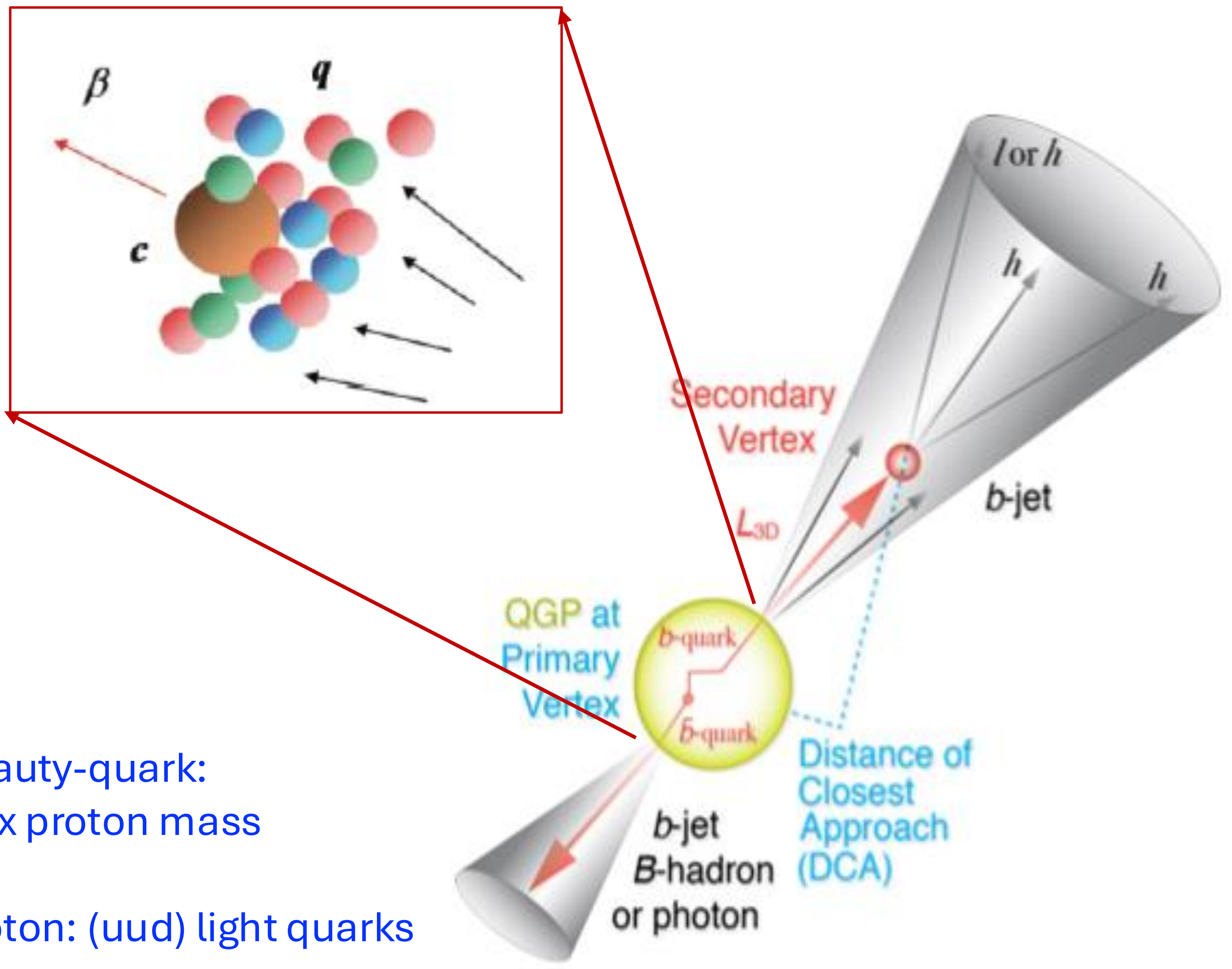
Performance: LS ~ MLP

# Alternative – more Partitions for Parallel Processing

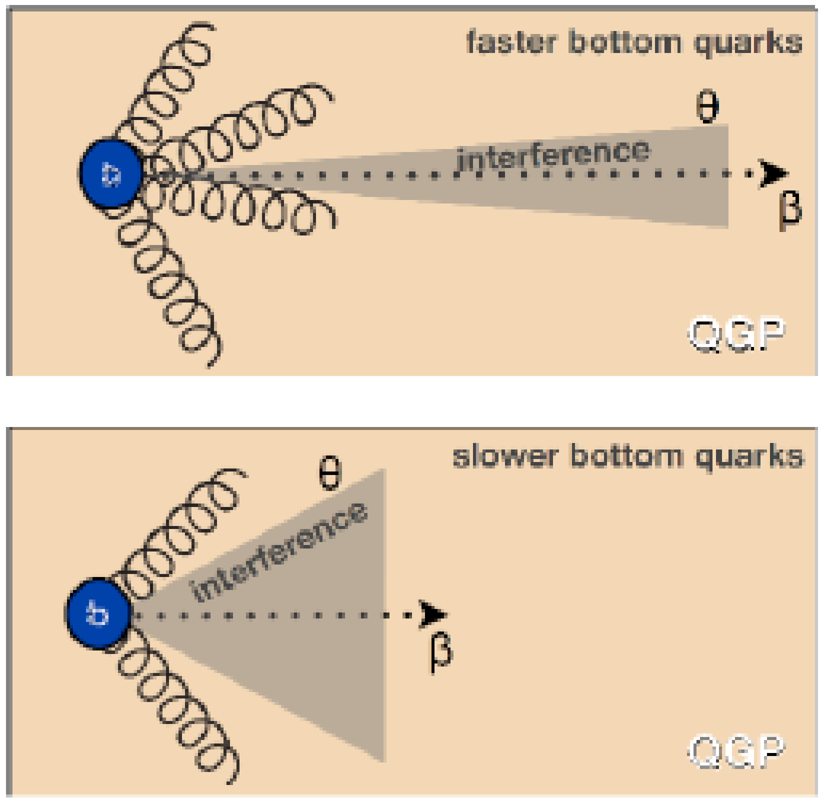
- ❑ **8 sectors** evenly divided along the azimuth angle $\varphi$
- ❑ **3** consecutive sectors form a **Zone**
- ❑ Adjacent zones share one overlapping sector
- ❑ Data streams within each zone are processed in parallel



# Heavy Quark Physics: a Pilar of RHIC Science



B-quark radiative energy loss in QGP
- Less dE/dx due to heavy mass



Beauty-quark:
~ 5x proton mass

Proton: (uud) light quarks