

# A demonstrator for a real-time AI-FPGA-based triggering system for sPHENIX at RHIC

Jakub Kvapil  
for the FastML team and sPHENIX collaboration

**TWEPP 2023 Topical Workshop on Electronics for Particle Physics**  
**Geremeas, Sardinia, Italy**

Los Alamos National Laboratory (LANL)  
Fermi National Laboratory (FNAL)  
Massachusetts Institute of Technology (MIT)  
New Jersey Institute of Technology (NJIT)  
Oak Ridge National Laboratory (ORNL)  
Georgia Institute of Technology (GIT)

# Motivation – Heavy Flavour

- The aim is to deploy **future system on Electron-Ion Collider (EIC)**
  - AI-based **electron tagging with streaming readout** to identify the (non)interesting Deep-Inelastic-Scattering (DIS) processes in the e+p/A collisions.
    - based on the measured scattering electron energy and direction
- Integrate the AI-based heavy flavour trigger system **demonstrator** into the **sPHENIX** experiment **for p+p run in 2024** to R&D its feasibility, requirements, and constrains
  - **Heavy-flavour (HF) events are very rare** ~1% probability recording of triggered events at RHIC energy
  - RHIC collision rate is around 3 MHz, sPHENIX readout 15 kHz (DAQ - 300 Gb/s)
  - Trackers are Streamed-Readout (SRO) capable, but can't save all TPC data
  - 10% trigger-enhanced SRO increases HF MB rate ~ 300 kHz
  - **ML HW tagging aims to sample remaining 90% of the luminosity**

# Who are we?

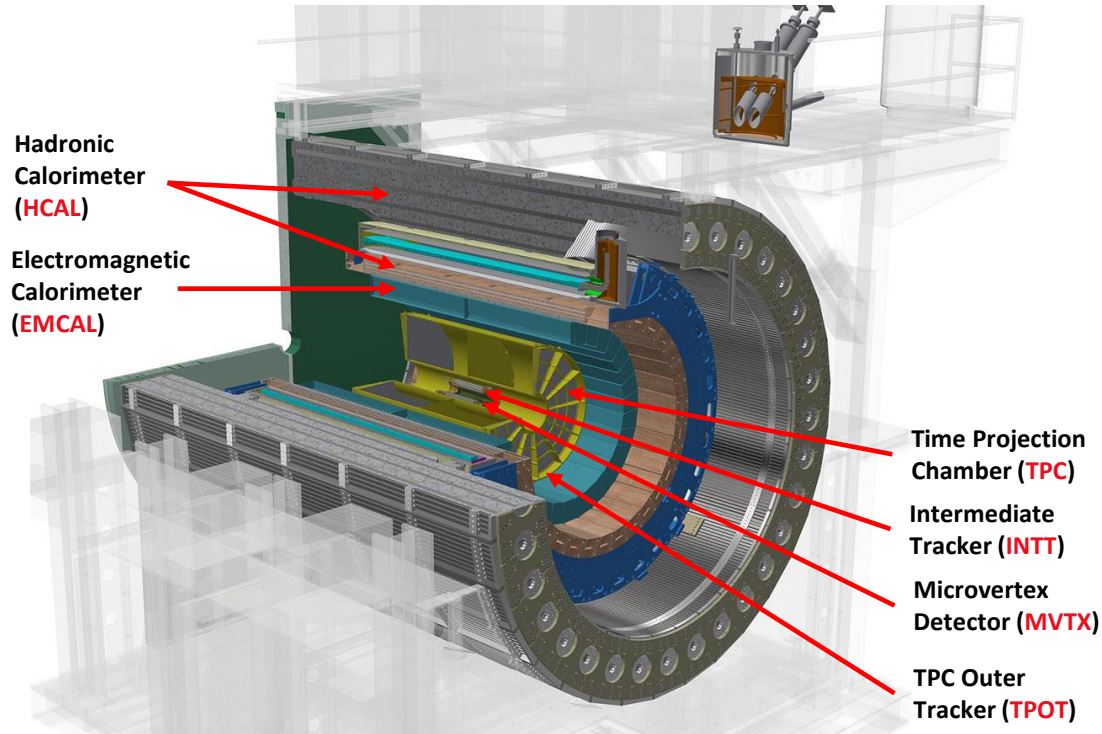
- Cross-discipline group of sPHENIX and LHC physicist, engineers, and computer scientists working on firmware-based ML applications data selections
  - sPHENIX is benefiting from a 2020 Department of Energy (DOE) funding call
- **The mission**
  - Efficiently extract critical and strategic information from large complex data sets
  - Address the challenges of autonomous control and experimentation
  - Artificial Intelligence for data reduction of large experimental data

## **Intelligent experiments through real-time AI: Fast Data Processing and Autonomous Detector Control for sPHENIX and future EIC detectors**

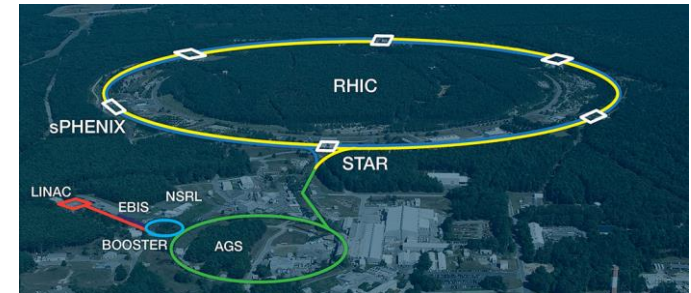
A proposal submitted to the DOE Office of Science

April 30, 2021 @ renewed for 2 more years in 2023

# sPHENIX experiment



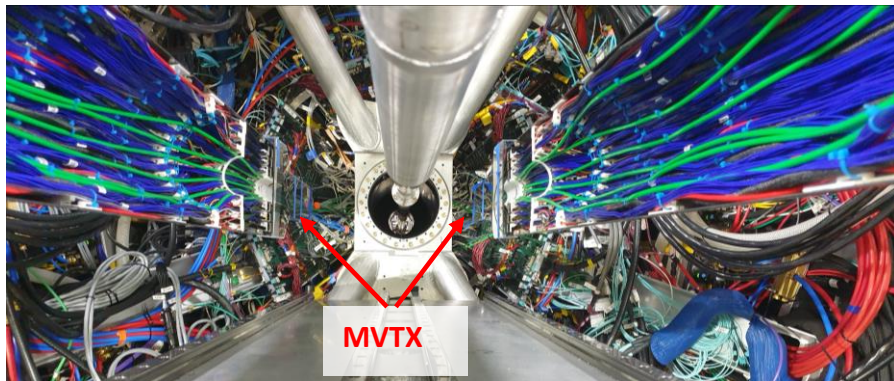
- Located at RHIC accelerator at BNL (USA)
- Running period 2023-2025
- ~4m long, ~3m high, 1000 tons
- Tracking detectors (MVTX, INTT, TPC, TPOT) and calorimeters (EMCAL, HCAL)
- 1.4 T Magnetic Field,  $|\eta| \leq 1.1$
- 160 MHz accelerator clock with 10 MHz BC
- 15 kHz Trigger Rate (due to calorimeter limit)
- MVTX and INTT
  - Tracking detectors capable of **streamed readout**



# MVTX and INTT

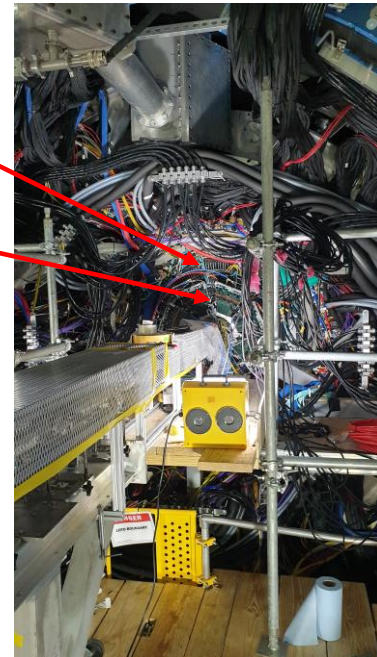
**MVTX** - Active area  $\sim 1685 \text{ cm}^2$

- Based on ALICE ITS2 **ALPIDE** chips, with ATLAS **FELIX** backend
  - Monolithic Active Pixel Sensors
  - Very fine pitch ( $27 \mu\text{m} \times 29 \mu\text{m}$ )
  - Good Time resolution  $\sim 2\text{-}5 \mu\text{s}$
  - 3 layers, 48 staves total, 9 chips per stave  $\sim 230\text{M}$  total channels



TPC

INTT

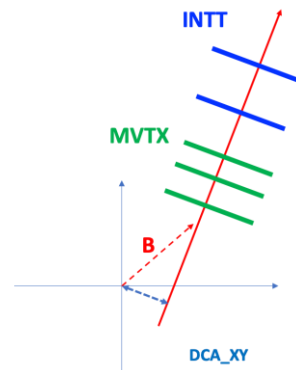


## INTT

- Silicon Strip Detector
  - Hamamatsu silicon modules
  - Pitch  $27 \mu\text{m} \times 16$  (or  $20$ ) mm
  - Excellent Time resolution  $\sim 100 \text{ ns}$   
( $100 \text{ ns}$  is the RHIC BC time)
  - 2 layers, 56 ladders total,  $360\text{k}$  channels

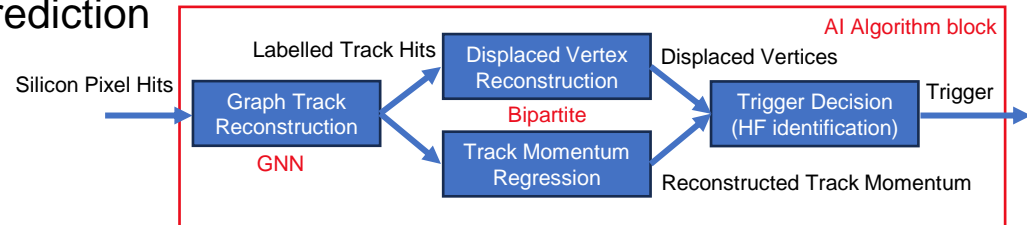
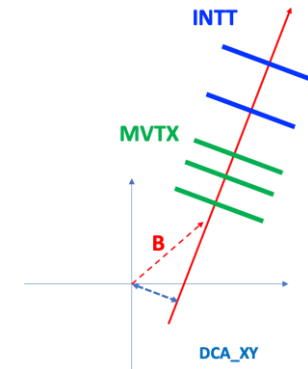
# The ML algorithm – TrackGNN

- **Based on Graph Neural Network (GNN)**
  - Detector and physics knowledge improves prediction
  - Based on **PyTorch** and **PyTorch Geometric**
- **Topological selection** of HF signals on FPGA
  - **Tracking and clustering** must be done on FPGA
- **Beam-spot and anomaly detection** on GPU based feed-back system
- Initial training on simulated data from MVTX and INTT
  - On GPU - NVIDIA Titan RTX, A500, and A6000
- We propose a **novel method to treat the events as track graphs** instead of hit graphs. This method is **driven by the physics** (transverse momentum)
  - Estimate momentum based on silicon hits -> 15% improvement on trigger decision



# The ML algorithm – TrackGNN

- **Three stages of event processing**
  1. Hits clustering
  2. Track reconstruction + outlier hits removal
    - by connecting the hits across different detector layers into hit pairs.
    - apply geometric constraints and down select the hit
  3. Trigger detection
- **Graph Neural Network** to solve
  - the track reconstruction problem
  - the inter-track adjacency matrix prediction
  - the graph level trigger detection



# The ML algorithm – TrackGNN

- **Challenges**
  - To provide an end-to-end solution that uses raw detector readout hit information to make trigger decisions for data collection.
  - To design a neural network compatible with the given detector readout and capable of learning a broad spectrum of physics properties
- **using low-level hits to build the high-level trigger decision.**
- Growing sub-field of geometric deep learning

$D^0 \rightarrow K^- \pi^+$  sample

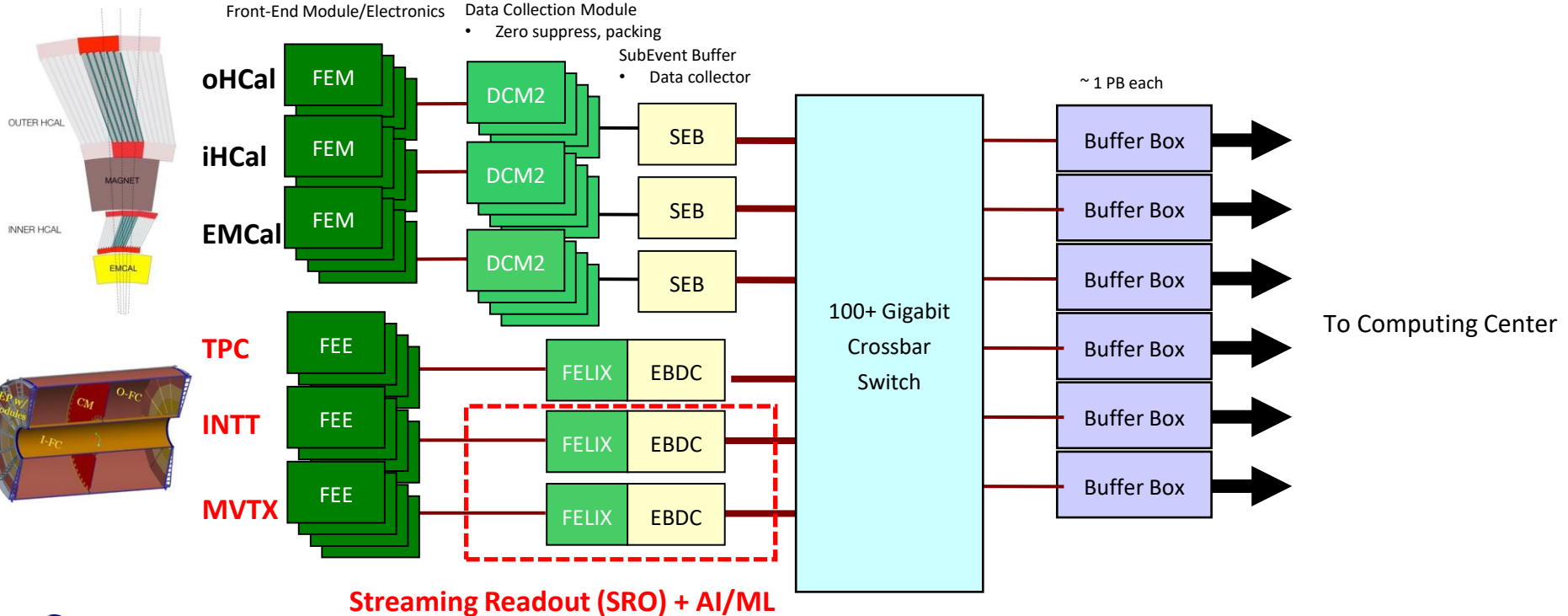
1% signal/background ratio			0.1% signal/background ratio		
Background Rejection	Efficiency	Purity	Background Rejection	Efficiency	Purity
90%	72.5%	7.25%	90%	78%	0.78%
95%	48.9%	9.78%	95%	50%	1.0%
99%	15.0%	15.0%	99%	17%	1.7%
99.33%	10.5%	15.74%	99.33%	11.0%	1.65%



# sPHENIX Readout and AI-ML HF Trigger Integration

On Detector

Rack Room



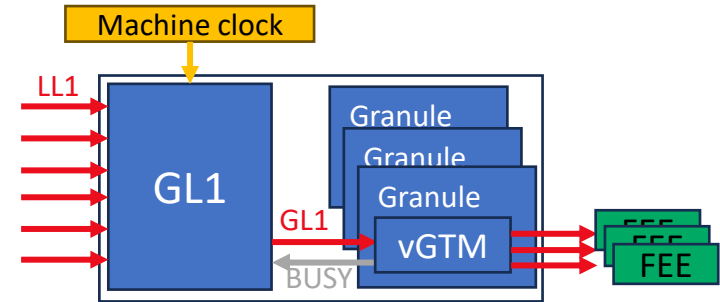
# The timing and trigger distribution

- The Global Level 1 Trigger (GL1) and the machine clock is distributed via Granule Timing Module (GTM)
  - GL1 transmits clock and trigger to the vGTM, which then transmits it to the FEE
    - Shared busy within a given Granule
    - vGTM can run in local mode decoupled from the global trigger (good for debugging)



## Trigger Inputs

- 8 LEMO, 56 fiber
- EMC, iHCAL, oHCAL, MBD, SEPD, ZDC

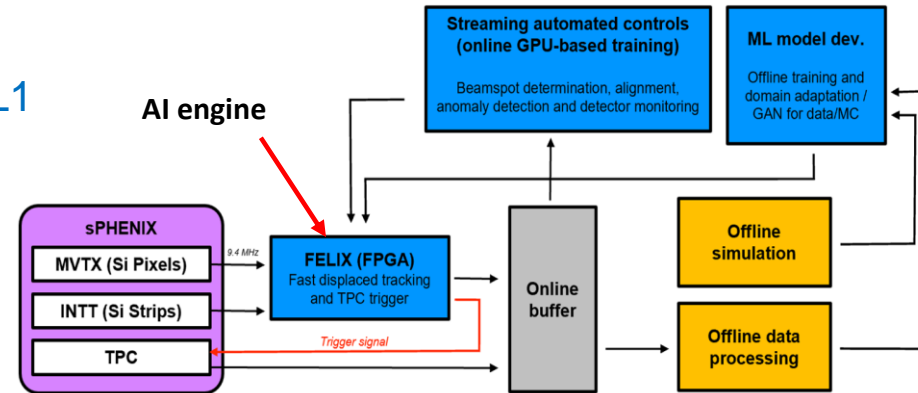


# The latency constrains for the TrackGNN

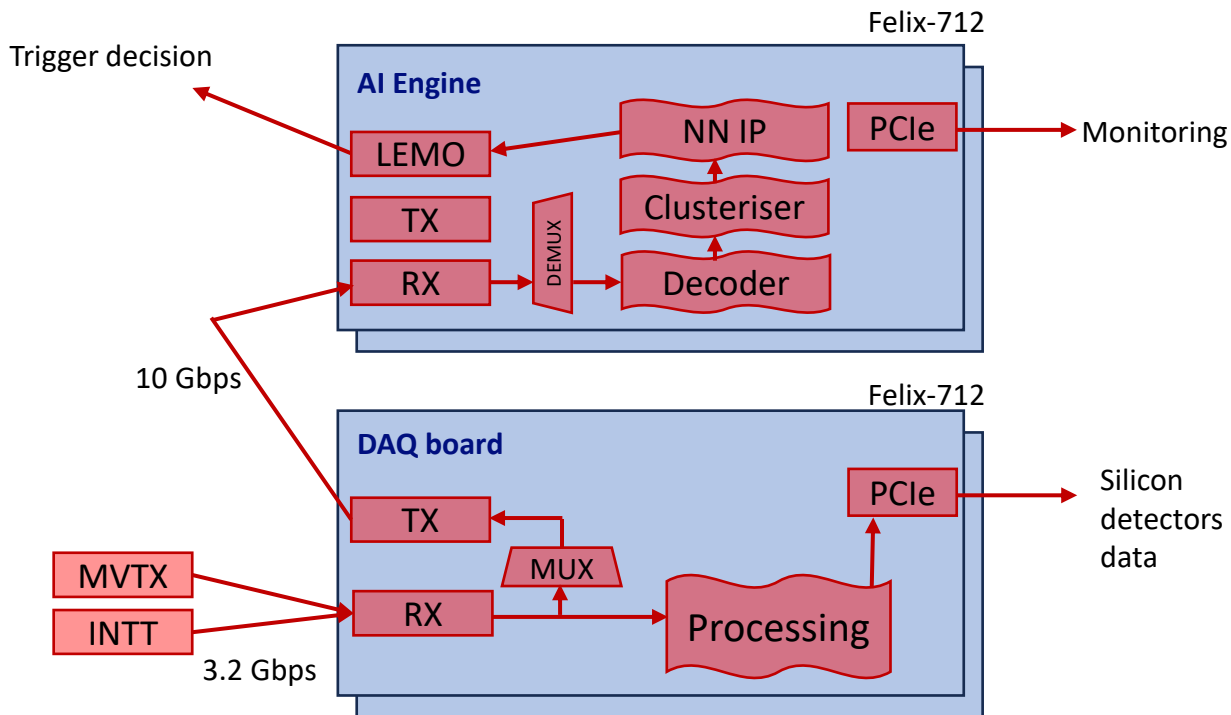
- The TPC buffers can hold up to 30  $\mu\text{s}$  of data
  - The goal of this project is to aim for 10  $\mu\text{s}$  collision-trigger latency to capture the TPC stream
- The Calorimeter buffers can hold up to 6.4  $\mu\text{s}$  of data
  - Can we stretch the latency down to 5  $\mu\text{s}$  to also capture the calorimeter stream?
- The latency breakdown
  1. MVTX readout window 5  $\mu\text{s}$  (aim to use 2  $\mu\text{s}$ ) – not fixed interaction-readout latency!
  2. IR -> Counting house  $\sim 0.3$   $\mu\text{s}$  (100 m cables)
  3. FELIX-AI data forward, decoder buffers  $\sim 0.6$   $\mu\text{s}$  (@240 MHz)
  4. Clusterizer + tracking + Trigger decision (currently 130  $\mu\text{s}$  for model)
  5. GL1 decision latency + Counting house -> IR  $\sim 0.3$   $\mu\text{s}$

# The DAQ–AI Data Flow

- **MVTX** 144 links @ 3.2 Gbps and **INTT raw data** stream will feed **two AI engines** (one for each hemisphere)
  - 8b10b protocol with links driven @ 10Gbps
    - tested up to 14 Gbps, with external loopback measurement at FELIX with BER <  $10^{-16}$
  - 24 links for MVTX and 24 links for INTT per AI engine
- The **decision signal** of heavy flavor event from the **AI-Engine** will be sent out via the LEMO connectors to the **sPHENIX GTM/GL1 system** to initiate the TPC readout in the triggered mode
- GPU based feed-back system for the beamspot monitoring



# The firmware design - data flow

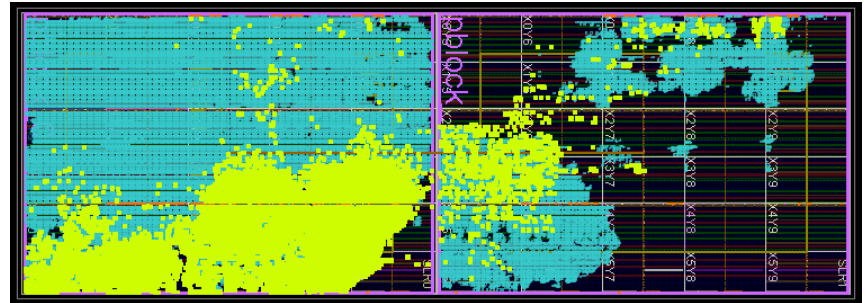


- Motivation to use FELIX board:
  - To reuse the PCIe implementation (16-lane Gen-3) and software tools provided by the FELIX developers
  - on-board FPGA is a Kintex Ultrascale XCKU115FLVF1924-2E
- Data needs to be decoded, clustered and feed the neural network
- Raw **MVTX** and **INTT** data packets:
  - 1 MVTX packet @2 us strobe
    - ~20 pp collisions (MB events) @4MHz pp collisions
  - 50 INTT packets (RF) @ 100 ns strobe
- **Very challenging** project to fit in the FPGA resources!

# The PCIe utilization

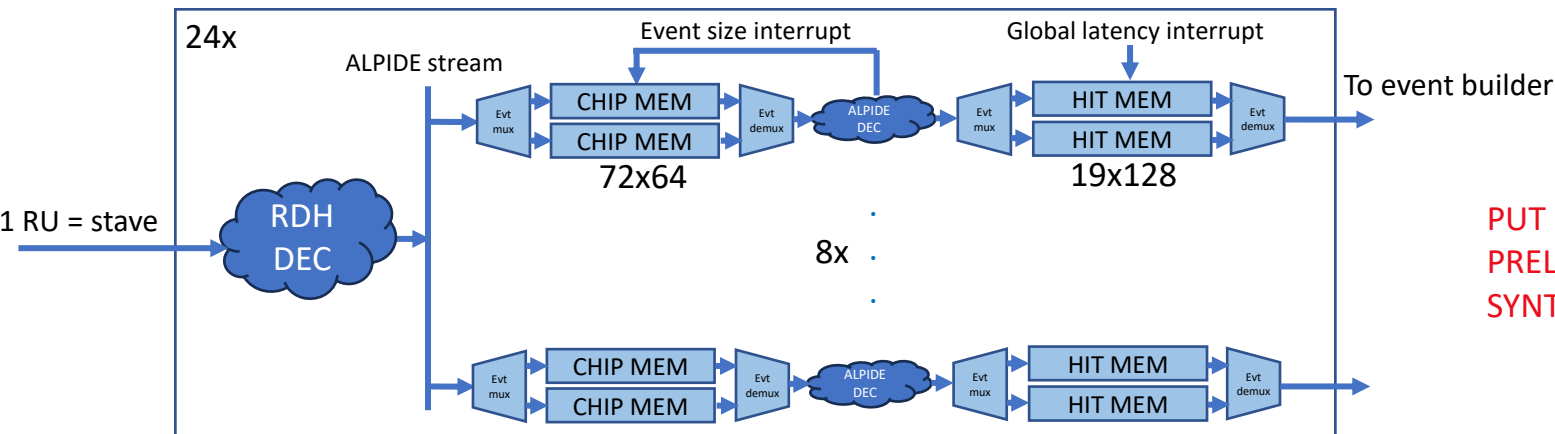
- Initial implementation at FELIX-711 by FNAL group
- Aim to use FELIX FW implementation of PCIe and its Software tools
- Current efforts to extract the Wupper module to lighten to logic and keep more resources for the AI IP code
- We use this standard well-understood **benchmark model “Jet Tagger”** (arXiv:1804.06913) to test the workflow
  - QKeras and converted to hls4ml to create an IP
  - 16 inputs (expert variables) and 3 dense hidden layers with 64, then 32, then 32 neurons;

Utilization (FELIX-711)			
	FELIX-711	PCI (Wupper)	JetTagger
LUT	241K (36.3%)	28K (4.26%)	83K (12.5%)
FF	310K (23.41%)	76K (5.75%)	50K (3.76%)
BRAM	635 (29.4%)	91 (4.22%)	195 (0.09%)
DSP	72 (1.3%)	0 (0%)	72 (1.3%)



# MVTX decoder

- Initial implementation of the FPGA-based MVTX decoder
- Max 128 hits per chip stored (expected physics ~50, issues with beam background?)
  - Maximum latency 532 ns @ 240 MHz
- The MVTX data latency depends on the actual collision time and hit occupancy
  - In order to fit into latency requirement an interrupts to event size/processing time are in place
  - Separate memory per MVTX event
    - For fast clear of the memory



PUT HERE SOME  
PRELIM POST-  
SYNTHESIS UTILIZATION

# Generation of the GNN IP core – two parallel efforts

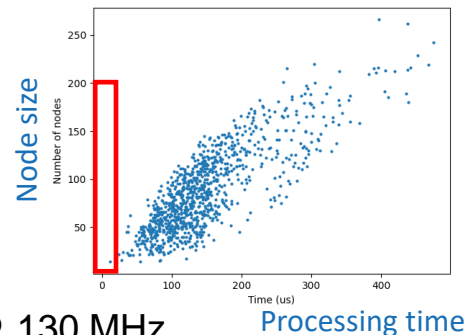
## 1. Team lead by the Georgia Institute of Technology (GIT)

- Direct translation of the sPHENIX TrackGNN model to IP using HLS
- Model
  - 5 layers, each layer: 64 dim 4 layers for node and 64 dim 4 layers for edge embedding

### - Implementation

- 100 nodes, 140 edges
- Measured Start-to-end latency
  - 150 us @ 130 MHz, 130 us @ 180 MHz
- Still needs 10-20x speedup!
- Fast-paced development 380 us (25<sup>th</sup> August) -> 150 us (4<sup>th</sup> September) @ 130 MHz
  - Attempts to increase clock to 300 MHz failed on timing constrains
  - Detailed latency breakdown and parallelism exploration ongoing
  - Might require model changes

Utilization (Alveo U280)	
LUT	308K (23.7%)
FF	378K (14.5%)
BRAM	1025 (50.8%)
DSP	1426 (15.8%)



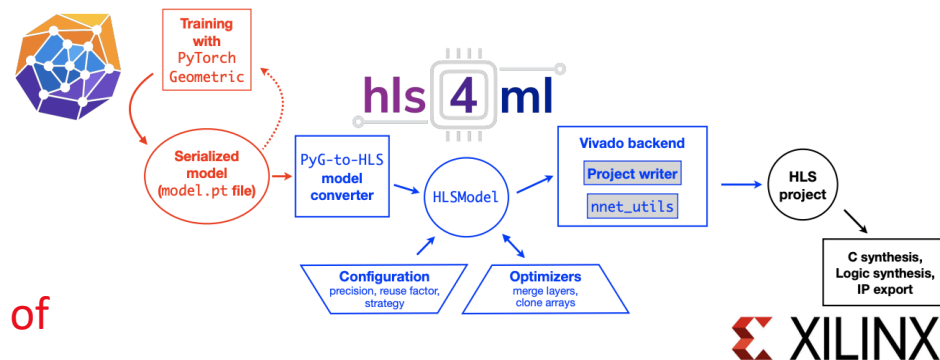
Close discussion between model developers and FPGA engineers



# Generation of the GNN IP core – two parallel efforts

## 2. Team lead by the Massachusetts Institute of Technology (MIT) and Fermilab (FNAL)

- Based on **High Level Synthesis for Machine Learning (hls4ml)**, a generalized python framework for machine learning inference in FPGAs
- **Third main upgrade underway**, focusing on 3 examples
  - Example 1: Tri-muon reconstruction with the LHC (muon endcaps)
  - Example 2: **Heavy flavor tracking at sPHENIX**
  - Example 3: Silicon strip tracking at LHC



Initial translation just started, expected first version of the TrackGNN model on FPGA end of October 2023

# The timeline

- July 2023
  - FELIX-712 and FELIX-182 setups installed at sPHENIX Counting house
- October 2023
  - TrackGNN IP core should be optimized and Implemented
  - Discussing between physicist, model developer, and FPGA engineers to meet the physics goals and constrains of the triggering system
- November 2024
  - Cosmic stream from the MVTX sent to the AI engine – tuning of the decoder parameters
- December 2024
  - Cosmic stream from the INTT sent to the AI engine – tuning the alignment and event builder
- January 2025
  - First pp beam at RHIC, final adjustment of the AI engine, performance studies

# Summary

- The TrackGNN model has been developed and tested on HF event simulation for sPHENIX
  - provides good precision while analysing two hemispheres independently
- IP core generation by two teams
  - Huge progress and improvement of the utilization and latency
  - Might need to reassess the model used to fit within FPGA resources and latency
- FLX-712 boards to serve as AI engine installed in sPHENIX counting house
  - Final push to finalize development of each FPGA component and placing them together
- A new FLX-182 board arrived to BNL which will be the base for EIC development
  - The backup plan to use it for the sPHENIX TrackGNN model (the FPGA is 3x bigger)

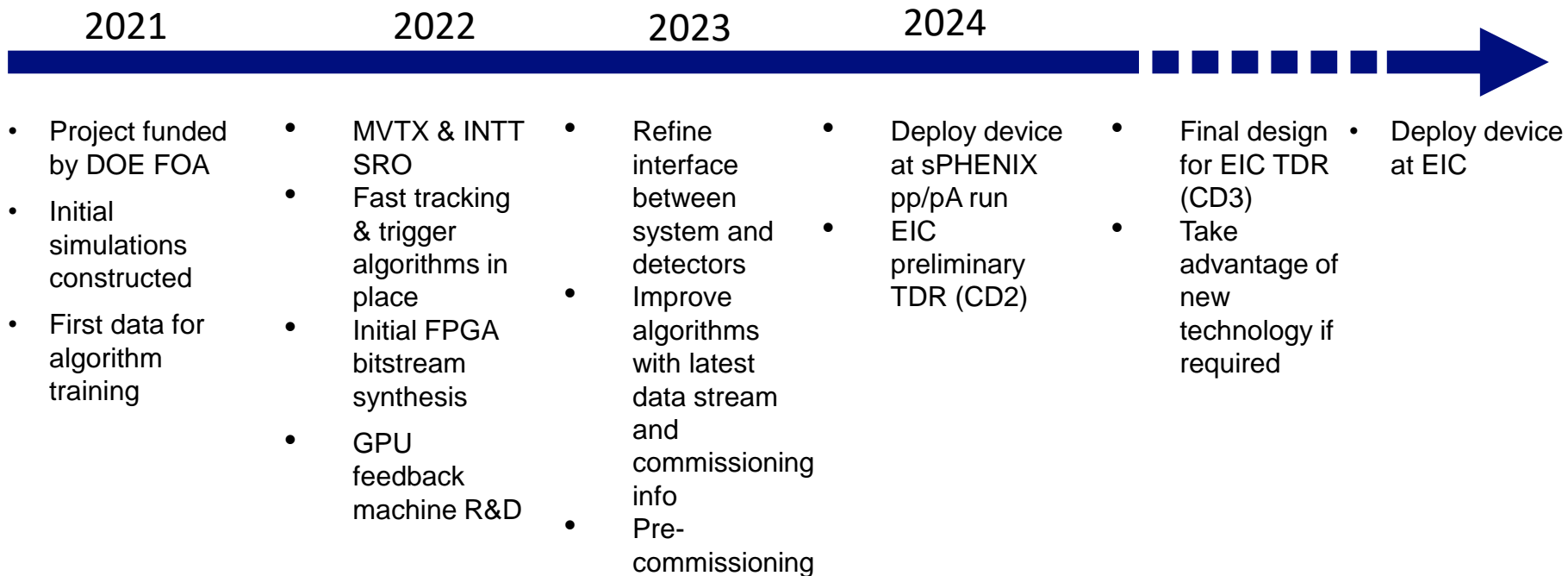
Thank you for your attention

Artificial intelligence and machine learning have the potential to revolutionize our approach collecting, reconstructing and understanding data, and thereby maximizing the discovery potential in the new era of nuclear physics experiments.

# Motivation – The challenges

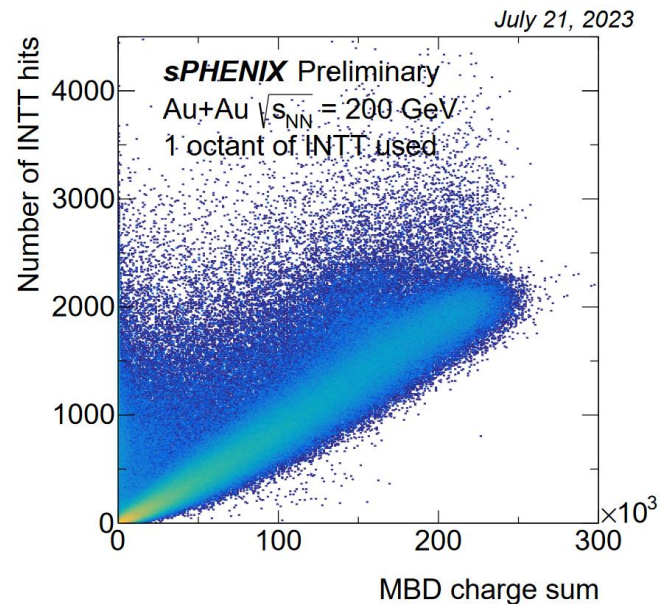
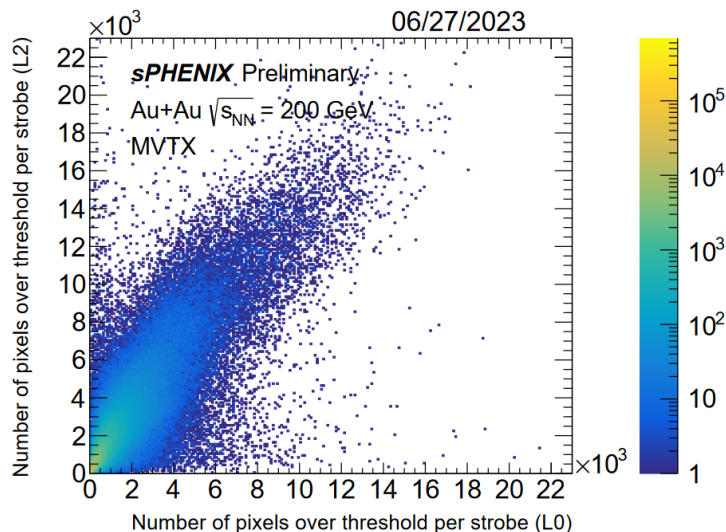
- **Real-time selection** of rare decays of HF particles
  - requires **continuous monitoring** and adjustment of the
    - beam trajectory (“beam spot”) – in time periods of seconds to hours, the position and shape can change (this will affect the HF the topology)
    - detector alignment, conditions and anomalies
- Adapt AI to **continuous learning** and changing conditions -> adaptive learning
  - Development of real-time autonomous closed loop adaptive learning system

# Predicted timeline



We are here!

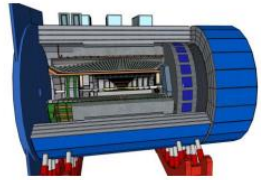
# MVTX and INTT commissioning performance



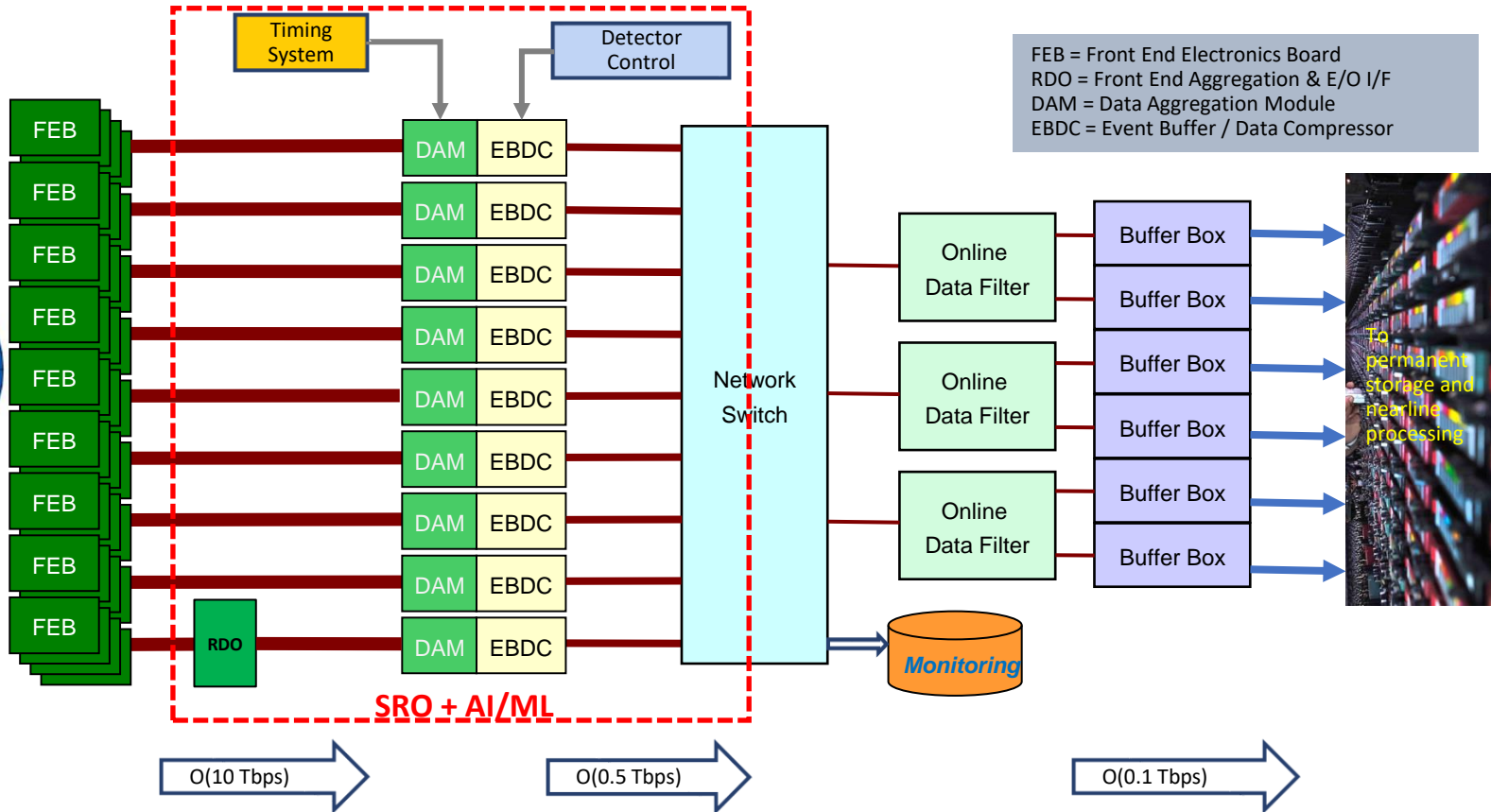
- Timing in detectors on good track

# From sPHENIX to ePIC: Streaming + AI/ML DAQ

from Jo's talk at  
ePIC collab. mtg



(s)PHENIX  
ePIC detector





# hls4ml – Planned Upgrade

2022, NP-Accel-RD-PI-Meeting

- **High Level Synthesis for Machine Learning (hls4ml)**
  - Python package for machine learning inference in FPGAs
- Hls4ml translates **NN algorithm into high level synthesis** and generates IP (Intellectual Property) core
  - translate it to the FPGA synthesizable high-level synthesis code.
- **Third main upgrade underway**, focusing on 3 examples
  - Example 1: Tri-muon reconstruction with the LHC (muon endcaps)
  - Example 2: **Heavy flavor tracking at sPHENIX**
  - Example 3: Silicon strip tracking at LHC

Design	$(n_{\text{nodes}}, n_{\text{edges}})$	Reuse factor	Precision	Latency	multiplier	DSP [%]	LUT [%]	FF [%]	BRAM [%]
		RF		[cycles]	$\Pi$ [cycles]				
Throughput-opt.	(28, 56)	1	ap_fixed<14, 7>	59	1	99.9	66.0	11.7	0.7
Throughput-opt.	(28, 56)	8	ap_fixed<14, 7>	75	8	21.9	23.8	4.7	0.7
Resource-opt.	(28, 56)	1	ap_fixed<14, 7>	79	28	56.6	17.6	3.9	13.1
Resource-opt.	(448, 896)	1	ap_fixed<14, 7>	470	174	56.6	25.0	7.4	16.5
Resource-opt.	(448, 896)	8	ap_fixed<14, 7>	<b>1590</b>	520	5.6	25.0	7.4	16.3

total width integer

@200 MHz, 1590 Cycles → 7.5μs