High Performance FPGA-Based Embedded System for Decision Making in Scientific Environment

Supported by DOE grant DE-SC0019518
Phase I: 2019-2020, Phase II: 2020 – 2022 (2023)

Yu Sun Principal Investigator and CEO
sunrisetechnology001@gmail.com
SBIR Exchange, August/17/2021-August/19/2021
Mingxiong Liu Co-PI of subcontract science Lead

Outline

- Company Introduction and its capabilities
  - Customers
  - Success Story: sPhenix Prototype AI-Enabled Detector
- Description of the Phase I project: Proof of concepts
- Phase II project: the objectives
- Nuclear Physics Background (Dr. Ming Liu)
- Trigger Algorithm Description
- Trigger Hardware Description
- Performance Report of Accuracy and Throughput
- Accomplishments, Future years' Plan and Milestones
- Highlights of the final products
- Plans

About Sunrise Technology

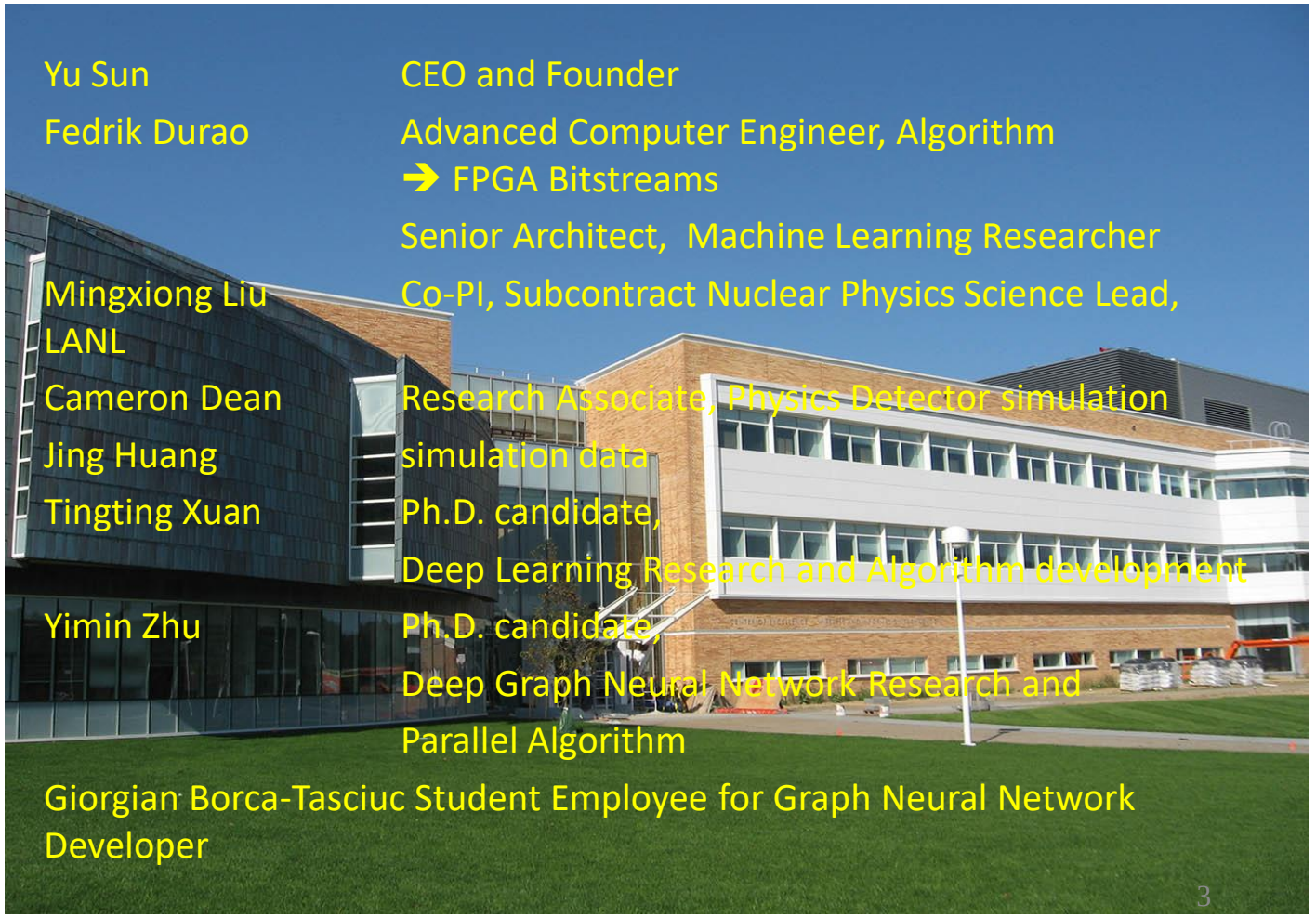
Founded in 2017 and located in Incubator@stonybrook

Providing AI technologies for science experiment control and education

The team: two full-time software and hardware engineer, one full-time post-doctoral associate, a part-time computer science consultant. We have been working with several graduate interns.

Product and Service areas:

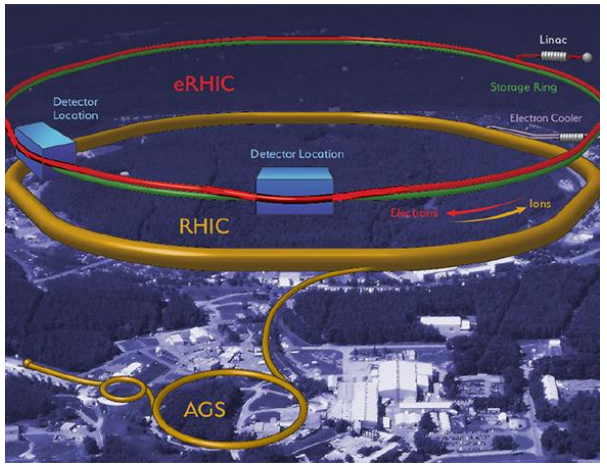
1) GNN machine learning models
2) Science Embedded Systems
3) FPGA-enabled GNN Solutions
4) FPGA High-level-Synthesis (HLS)
5) Embedded system for modeling training
6) Deep Reinforcement Learning for large facility control



Core Competencies of Project Team and Customer

- AI/ML for Science Facilities
- Machine Learning and Deep Learning Algorithm Design
- Deep Reinforcement Learning for Orbit Control
- AI-Enabled Heterogenous Embedded systems with CPU, GPU, and FPGA) for Science Facilities Automation (particularly, accelerators detectors)
- Data Science for Physics Analysis
- Edge Systems Software Stack

sPHENIX



SBIR Project Periods

Project Period April/2020-April/2021: Basic Algorithmic Development

- Identify and Train Machine Learning Models with Inner-most MVTX detector simulation data
- Verify the baseline Performance Accuracy and Throughput
- Preliminary Hardware Development on FPGA

Apr. 2020 – Apr. 2021

May 2021 – Apr. 2022

Project Period May/2021-April/2022: Advanced Algorithmic and FPGA Development

- Advanced ML-Algorithm Development with MVTX + INTT detector simulation data
- Improve model performance
- Reduce model inference latency
- FPGA HLS development and deployment
- Objective: 200K-300K events/second

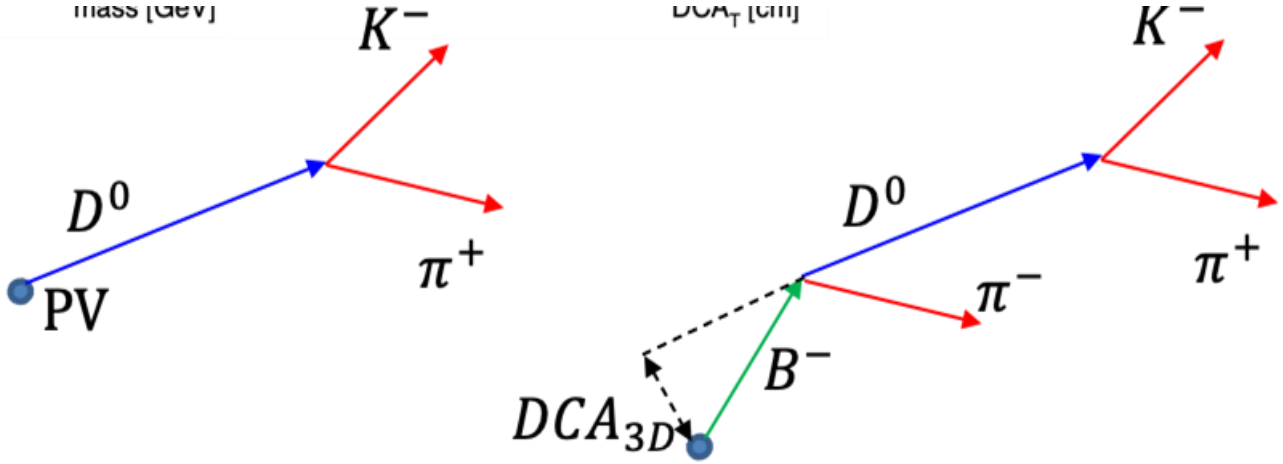
Benefit to DOE NP SBIR Program

- Project Focus:
  - Real-time AI technologies will be applied to the very high-rate data streams from detectors.
  - Accelerate GNN on FPGA, one of the first work that attempts to accelerate GNN prediction.
  - Play the central role in sPhenix and Future EIC detectors running under trigger systems and in-situ streaming analysis for event selections.
- Project Impacts:
  - **ASCR Topic 6, Subtopic b):** EMERGING INFORMATION TECHNOLOGIES FOR SCIENTIFIC FACILITIES AND HPC ENVIRONMENTS
  - **NP Topic 32: Nuclear Physics Software and Data Management and subtopic b. Applications of AI/ML to Nuclear Physics Data Science**
    - US DOE SBIR FY 2022 Topics Document: areas a), b, c), in Pages 90-92.
  - **NP TOPIC 33. NUCLEAR PHYSICS ELECTRONICS DESIGN AND FABRICATION**
    - **Use FPGA as prototype for Front-End Application-Specific Integrated Circuits (subtopic b)**
    - Provide ML-based Data processing capability for Next Generation Pixel Sensors
  - **NP TOPIC 34. NUCLEAR PHYSICS ACCELERATOR TECHNOLOGY**
    - **Subtopics f. Accelerator Control and Diagnostics**

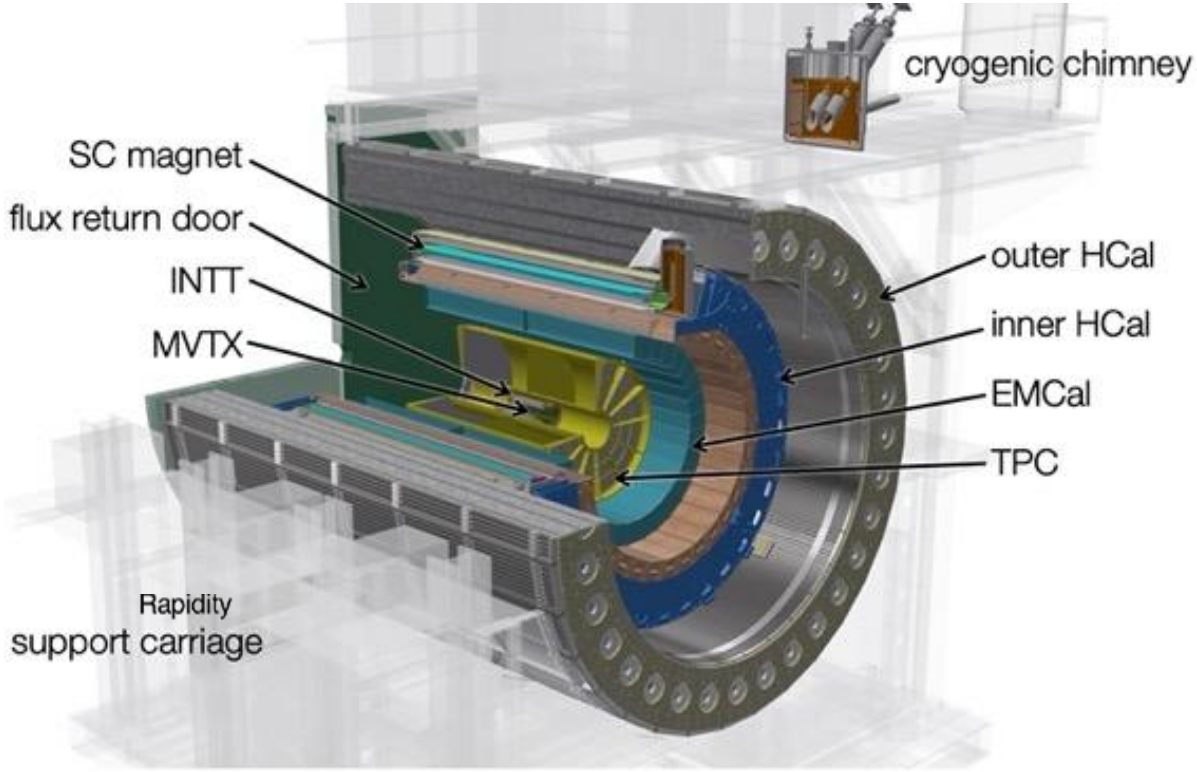
The Readout Challenge for High Luminosity Physics

- **The readout challenge**
  - Raw data volume >> hardware bandwidth/storage
  - Only a small fraction of data will be recorded to tape

- **sPHENIX: DAQ trigger rate, 15kHz**
  - AuAu collisions
    - Max collision rate ~50kHz
    - Can collect all central collisions, OK
  - p+p and p+Au
    - Collisions on each beam crossing, ~9.4MHz
    - Okey for high energy jet program with triggers
    - Lose most of the low pT physics events
- AI-based Triggering: filter events to reduce data rates for data archive and offline processing
- sPhenix Trigger ➔ TPC (Time Projection Chamber)➔ Data Acquisition
- SBIR project focuses on designing, building, simulating, and benchmarking a prototype event readout system with AI-based fast online data processing and autonomous detector control system that meets the physics and engineering requirements.



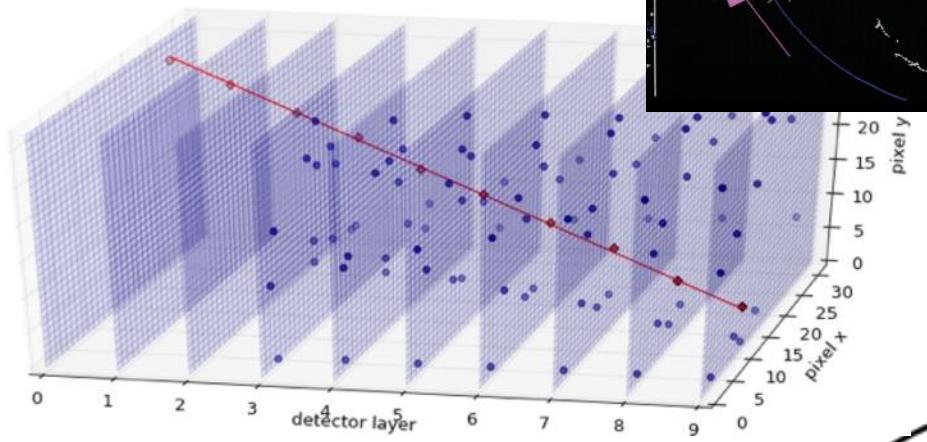
sPHENIX experiment under construction at RHIC:
- Day-1 physics in 2023

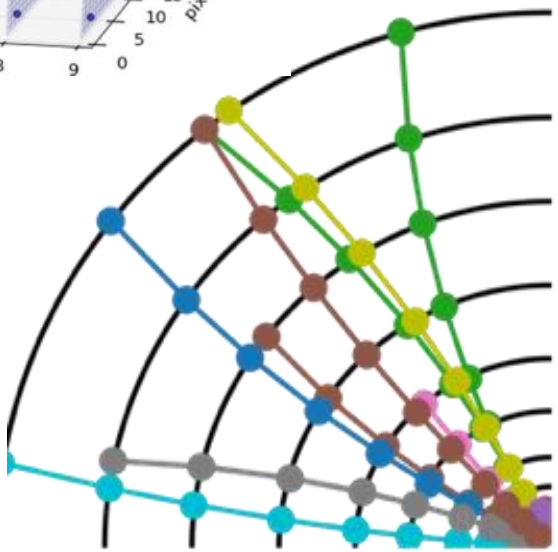


Event Data Descriptions

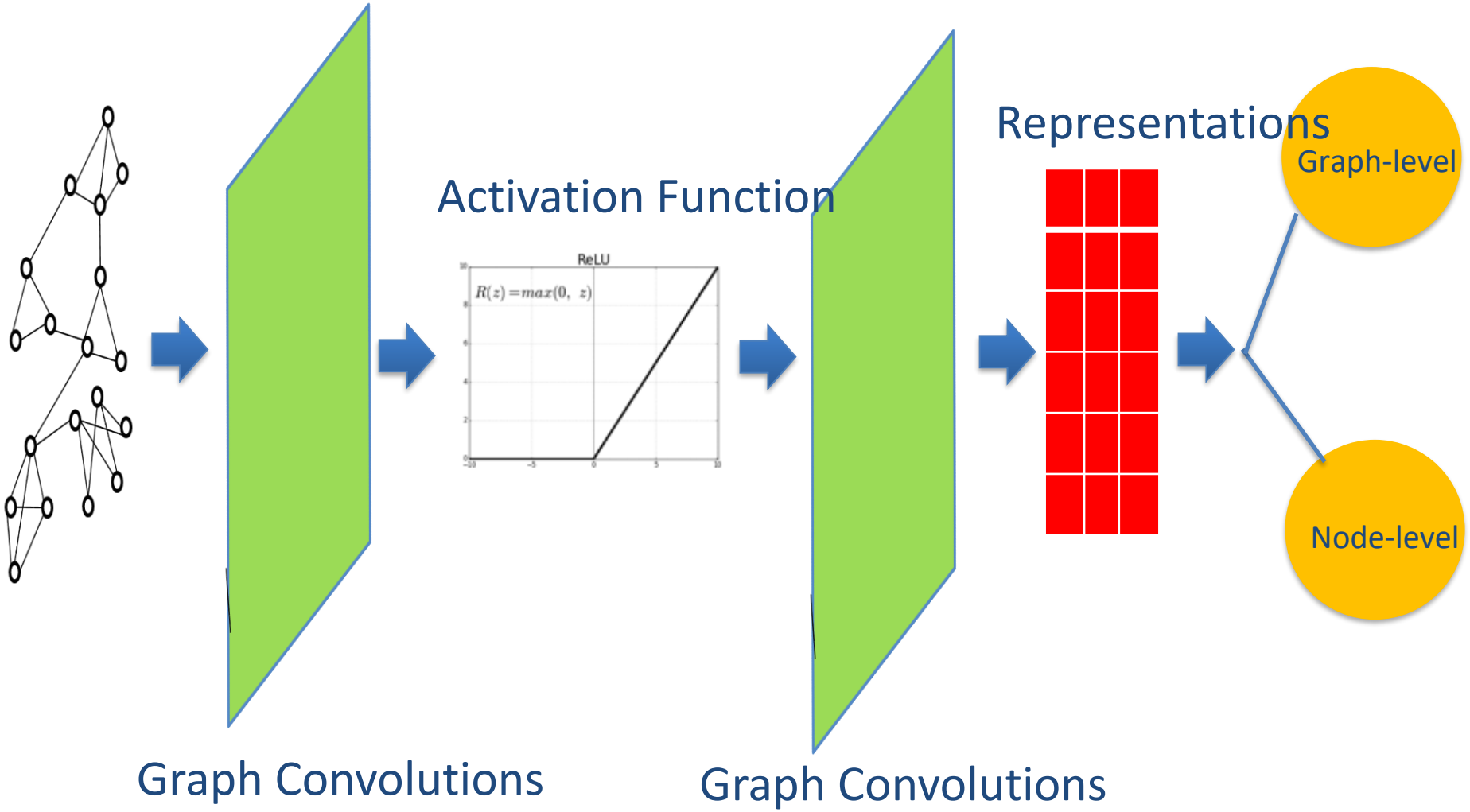
Moving from images to points

- **Image-based methods face challenges scaling up to realistic HL-LHC conditions**
  - High dimensionality ($9K \times 9K \times 3$) and sparsity
  - Irregular detector geometry
- **Instead of forcing the data into an image, use the space point representation**
  - Harder to design models (variable-sized inputs/outputs)
  - But now we can exploit the structure of the data with full precision
- What ML models are appropriate for the event on right
  - Recurrent neural networks and Graph neural networks





Graph Neural Networks



Trigger Software Pipeline

1. Fetch events from event buffer (Work in Progress)

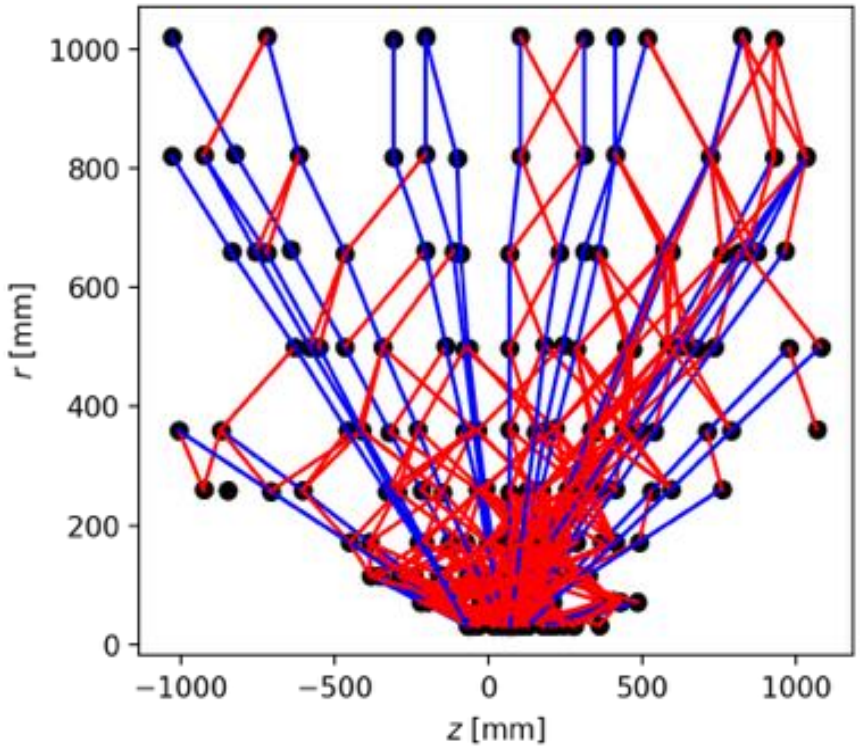
2. Data Pre-processing Clustering (Work in Progress on FPGA implements)

3. Tracking + Outlier hits Removal (Done in FPGA)

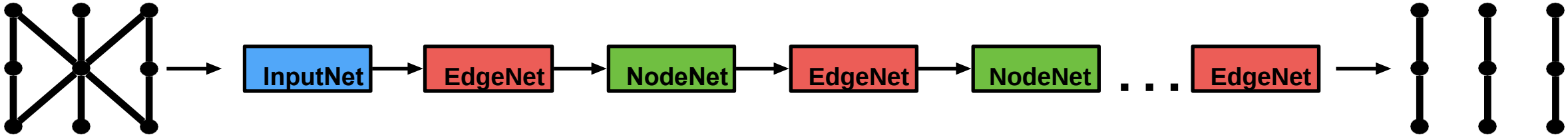
4. Triggering (Done in FPGA, need performance tuning)

5. Triggers on TPC (Interface and integration with sPhenix Detector)

Graph Tracking and Outlier Removal

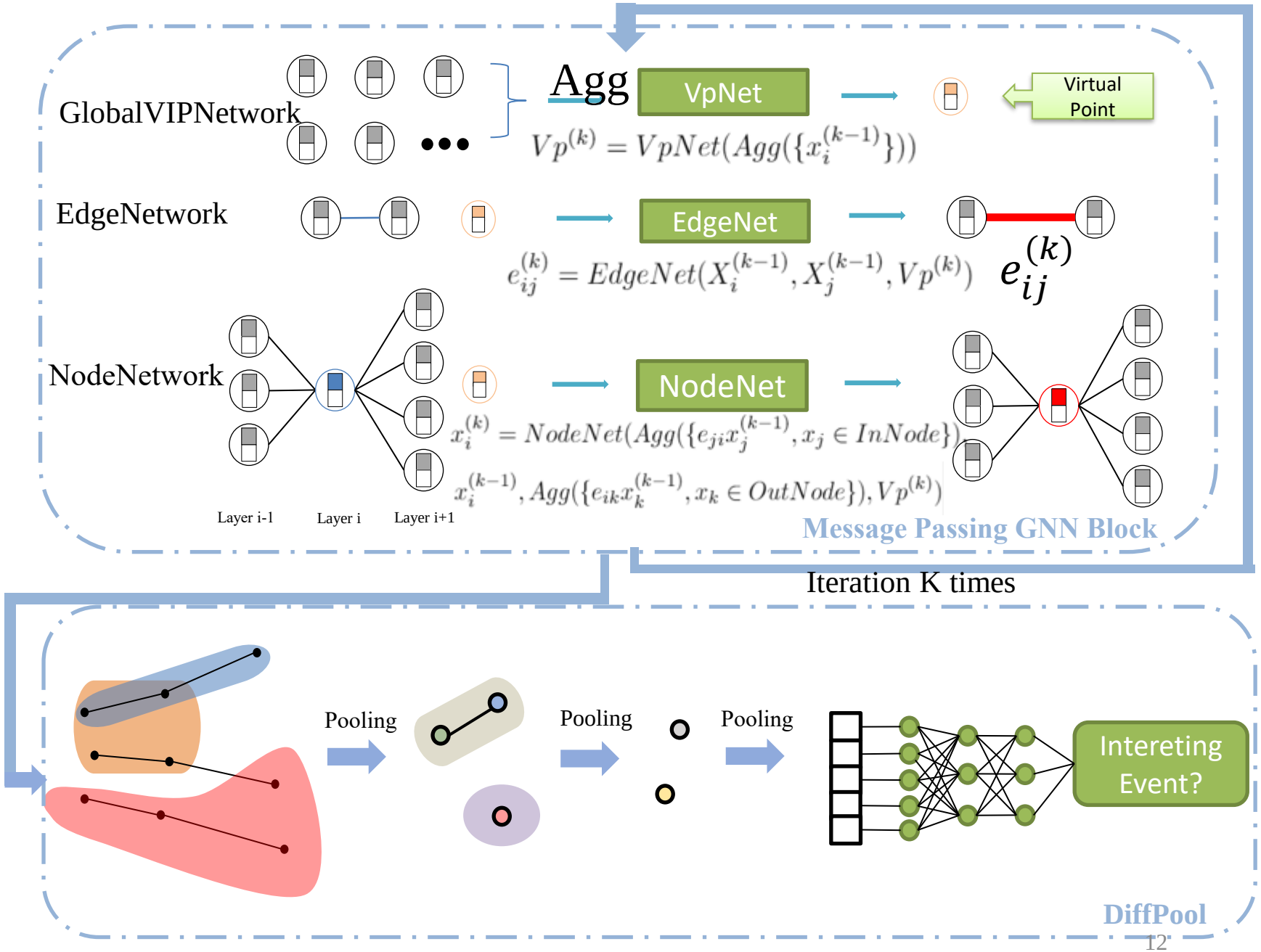


- **What if we structure our data as a *graph* of connected hits?**
  - Connect plausibly-related hits using geometric constraints
- **What kinds of models can we apply to this representation?**
  - Traditional architectures clearly don't work
  - but there's a growing sub-field of ML called *Geometric Deep Learning*
- Connect hits on adjacent layers using crude geometric constraints, i.e., $\delta(\phi) \leq \frac{\pi}{4}$ and $\delta(z) \leq 300mm$



With each iteration, the model propagates information through the graph, strengthens important connections, and weakens useless ones.

Trigger Detection



Deep Learning Training and Inference Product Hardware

Stage 1: Off-line Training

Stage 2: On-line Training /Transfer Learning

Stage 3: Real-Time Inference

Stage 4: Light-Weight Real-Time Deep Learning & Inference Engine

Stage 5: Real-Time Sensing & Decision Making Engine

1

2

3

4

High Performance Computing — Edge Computing — Embedded System — Controller

Electricity Cost High — Electricity Cost Medium — Electricity Cost Low

Four GPU servers

High-end Embedded System

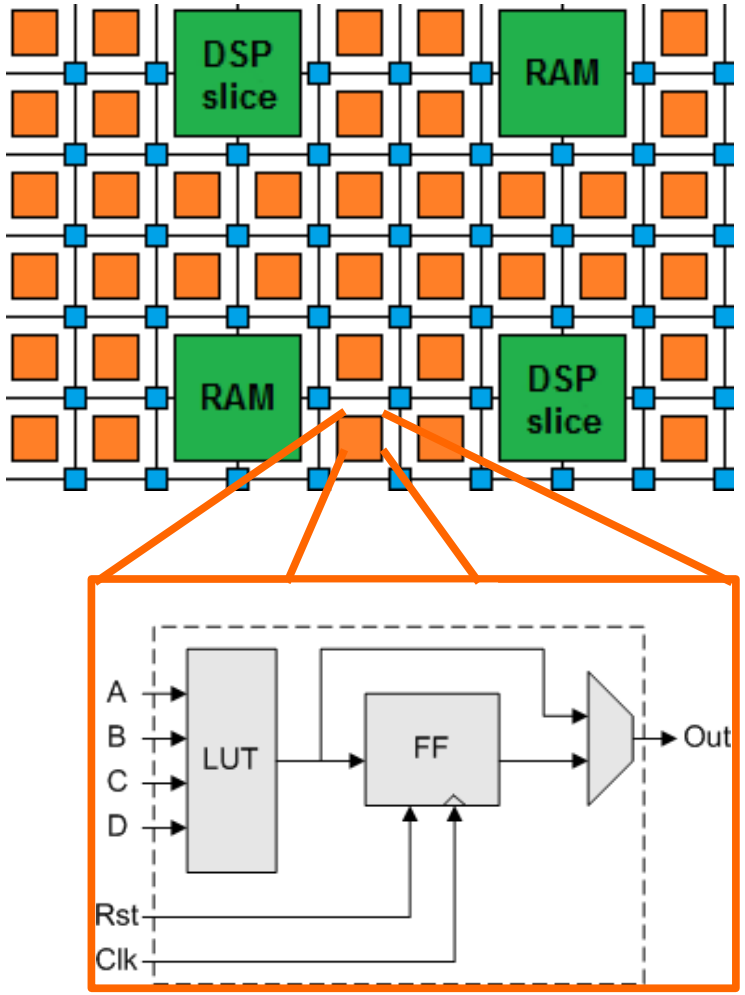
AI-Engine Board and Standalone Embedded System

FPGAs

5us latency to decide whether acquire event data in TPC

ASIC and FPGA

- Field-programmable gate arrays are a common solution for fast-computing
- Building blocks:
  - **Multiplier units (DPSs) [arithmetic]**
  - **Look Up Tables (LUTs) [logic]**
  - **Flip-flops (FFs) [registers]**
  - **Block RAMs (BRAMs) [memory]**
- Algorithms are wired onto the chip
- Programming traditionally done in Verilog/VHDL
  - Low-level hardware languages
- Possible to translate C to Verilog/VHDL using High Level Synthesis (HLS) tools



Virtex 7 XCKU 115
5520 Multipliers
663K LUTs
1.3 FFs
72 MB BRAM

Alveo U280 FPGA
9024 Multipliers
1.4 M LUTs
2.6 M FFs
8G HBM
53MB Block RAM

Programming Environment for FPGA

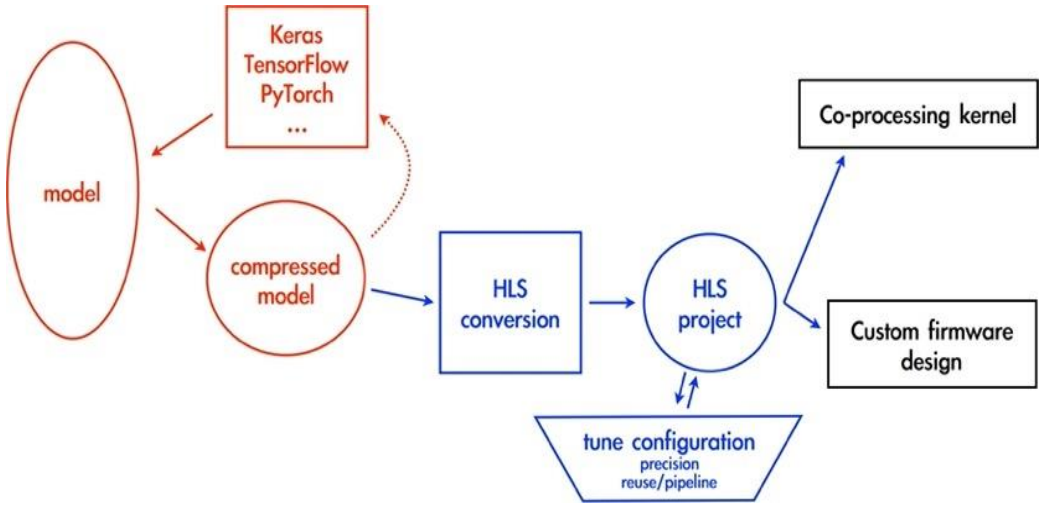
`hls4ml` is a software package for creating HLS implementations of neural networks–

https://hls-fpga-machine-learning.github.io/hls4ml/

- Supports common layer architectures and model software
- Highly customizable output for different latency and size needs
- Simple workflow to allow quick translation to HLS
- Design model with standard software tools (Keras, Tensorflow, PyTorch)
- Pass network architecture and weights/biases along with configuration parameters to **`hls4ml`** (creates HLS project)

XiLinx DPU

- Only handles CNN
- A predefined set of deep learning modules
- Does not support Matrix operations.
- Does not work for GNN
- Inefficient for large pipelines require CPU to handle I/O and coordinate FPGA
- Not appropriate for trigger detection with sparse images

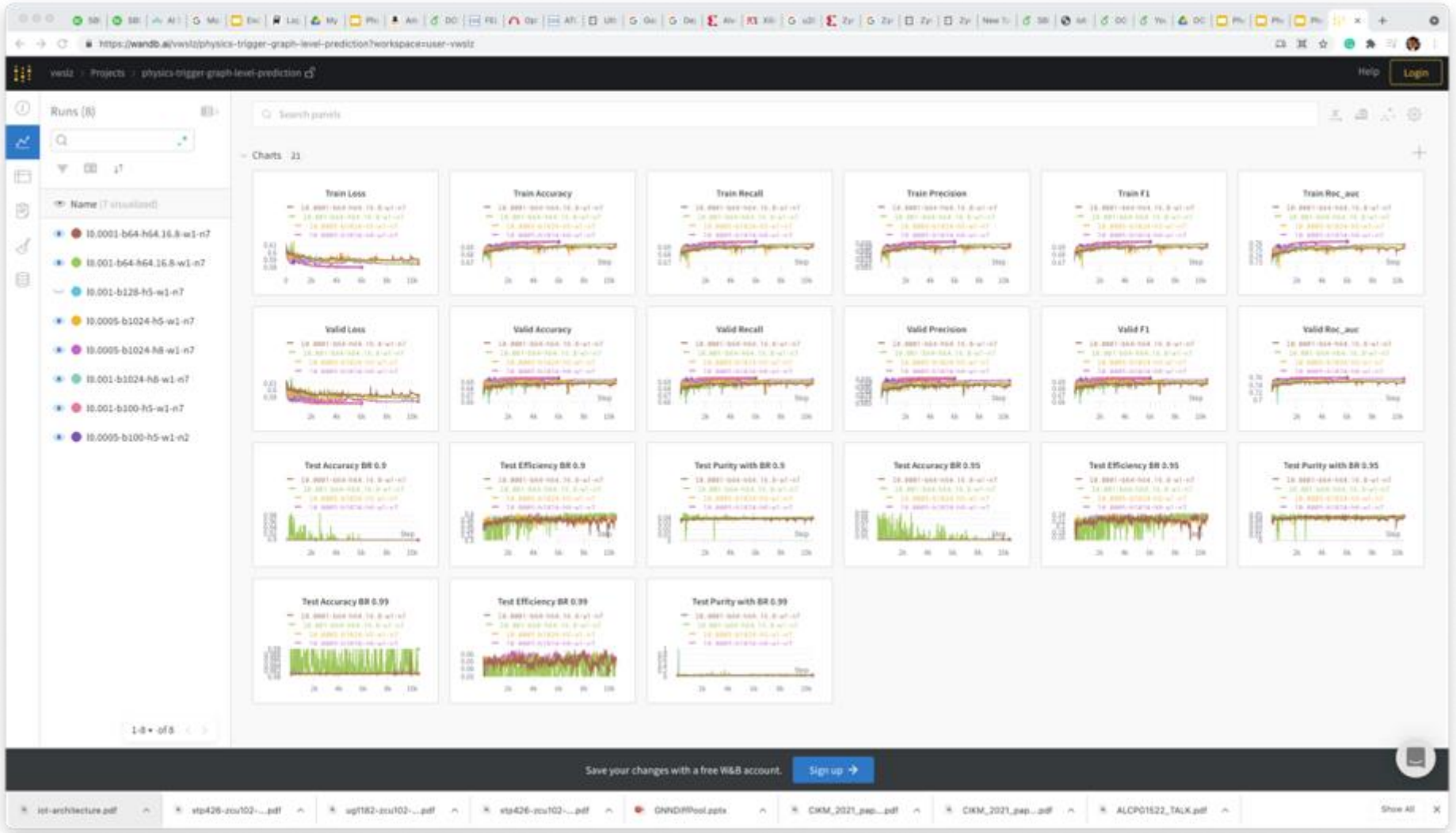


Performance Analysis: Accuracy and Throughput

	Current performance		Goal 1	Goal 2
Efficiency	50%	25%	20-50%	90%
Purity	5%	5%	5%	5%
Background Rejection	90%	95%	99%	95%

- Accuracy for Track Reconstruction
- Accuracy for Trigger Detection
- F1 and AOC for Trigger Detection
- Given the percentage of trigger events in original dataset and background rejection rate, calculate the efficiency and purity of the detected triggers

Training Dashboard



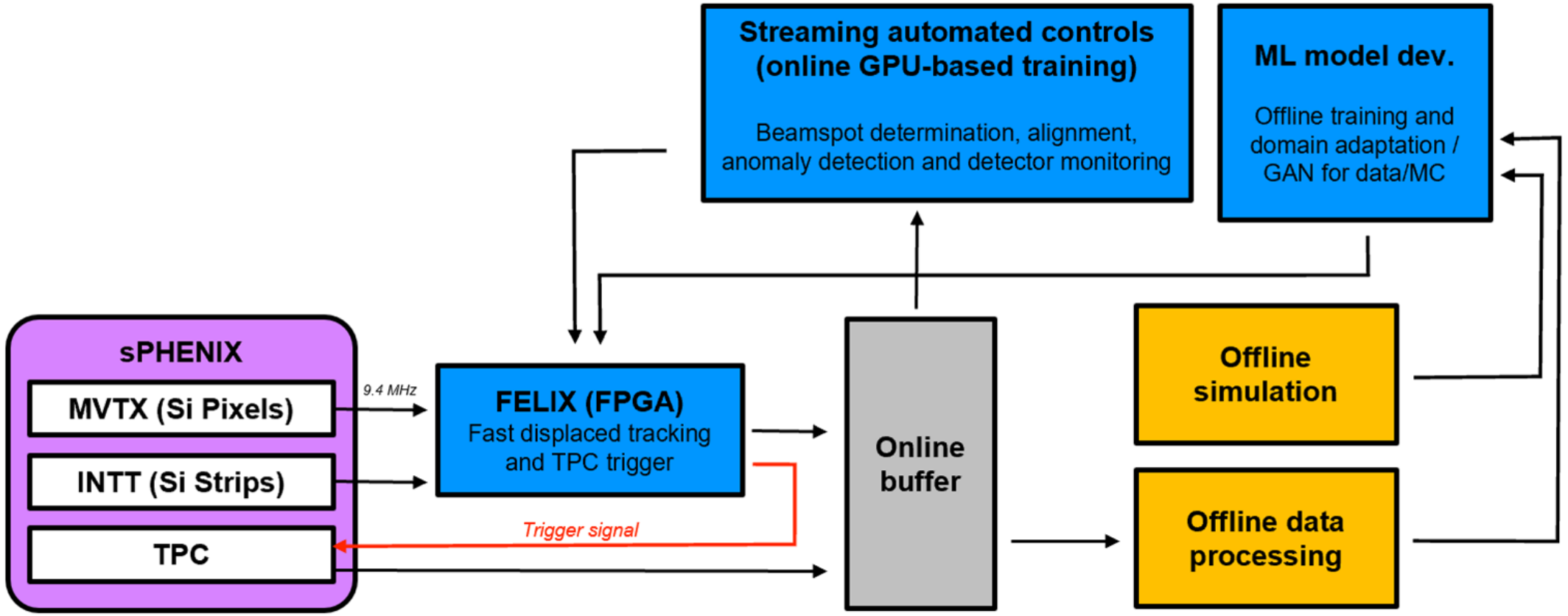
Tracking + Outlier Detection Performance Results

- Graph Neural Network (GNN)
- Without noise, cluster the pixels with ground truth Accuracy: >99%
- With noise, clustering algorithm applied, Accuracy: 93 -96%, depends on the size of the model (hidden dimension: 8, 16, 24, 32, …)
  - Will keep 93%-96% hits. The performance will trade-off with the size of model (complexity).

Future Plan: sPHENIX Readout Upgrade

AI-based real-time system: Fast Data Processing and Autonomous Detector Control

- Identify heavy quark events in p+p and p+Au collision events
- Autonomous detector & beam position misalignment monitoring and correction



Conclusion, Accomplishments and Milestone

1. Implement the Trigger Detection Algorithm based on advanced GNN
2. Implement Physics-aware pipeline for decision making
3. Extremely fast GNN algorithm on FPGA (3KHz/second for end-to-end pipeline), 20 times faster than GPU.
4. With the Support of HLS4ML, the trigger software runs on a server and embedded system with FPGA

Year 2 milestones

- Simulation Dataset with MVTX+INTP (1~5 million events) and retrained models
- FPGA implementation for new models with MVTX and INTP
- Fast prototype design for online triggering hardware
- Design and implement embedded system with both training (on GPU) and inference (on FPGA)

(Possible) Year 3 milestones:

- sPhenix trigger to be deployed for upcoming sPhenix experiment run at 2023.