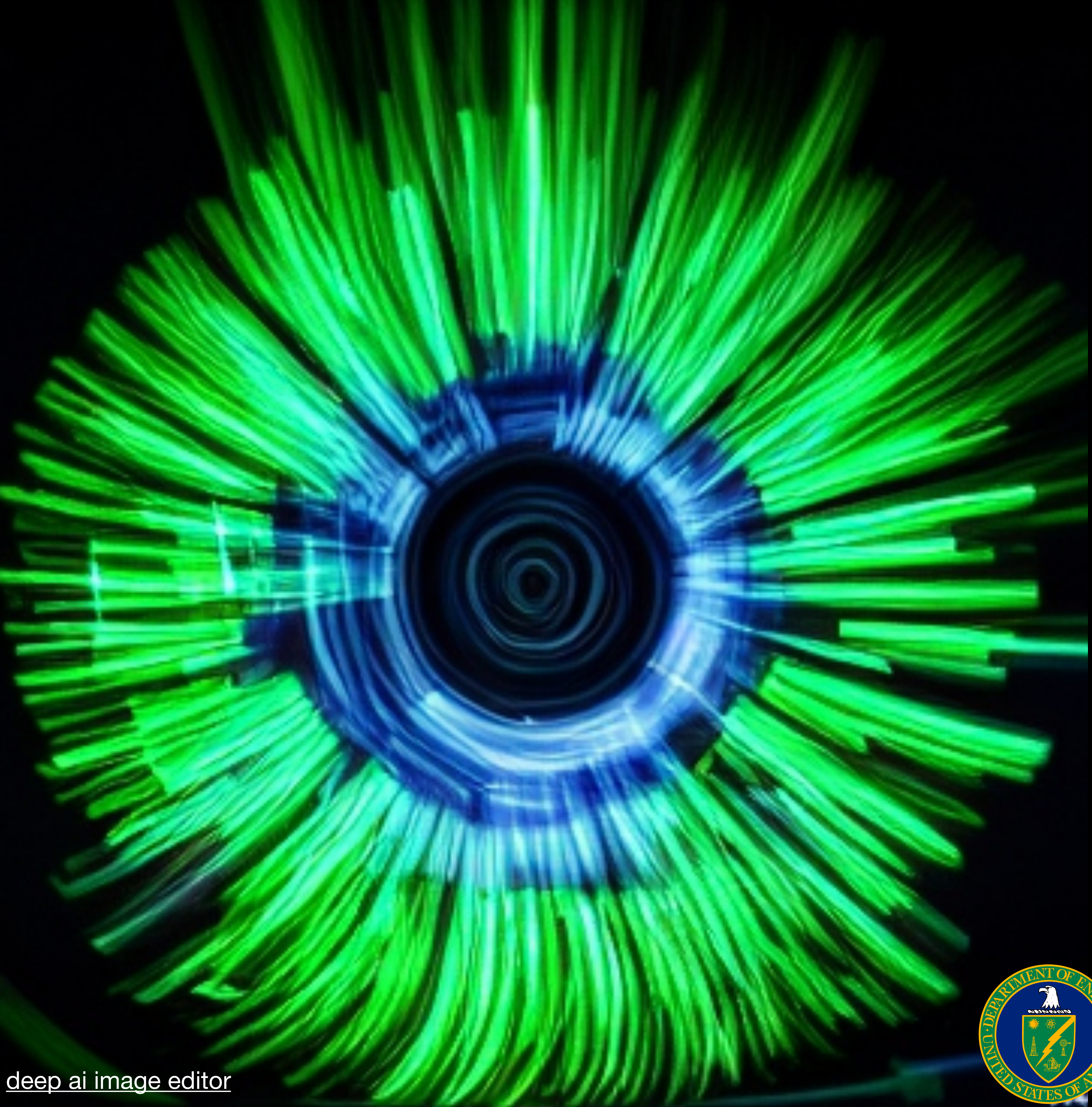


deep ai image editor

Machine learning on FPGAs for event selection

Hannah Bossi (MIT)

RHIC/AGS User's Meeting

May 21, 2025

MIT HIG's work was supported by US DOE-NP

Laboratory for Nuclear Science

ROADMAP

sPHENIX

HOW COULD ML ON FPGAS HELP THE PHYSICS GOALS OF SPHENIX?

WHAT IS THE PROGRESS WE'VE MADE?

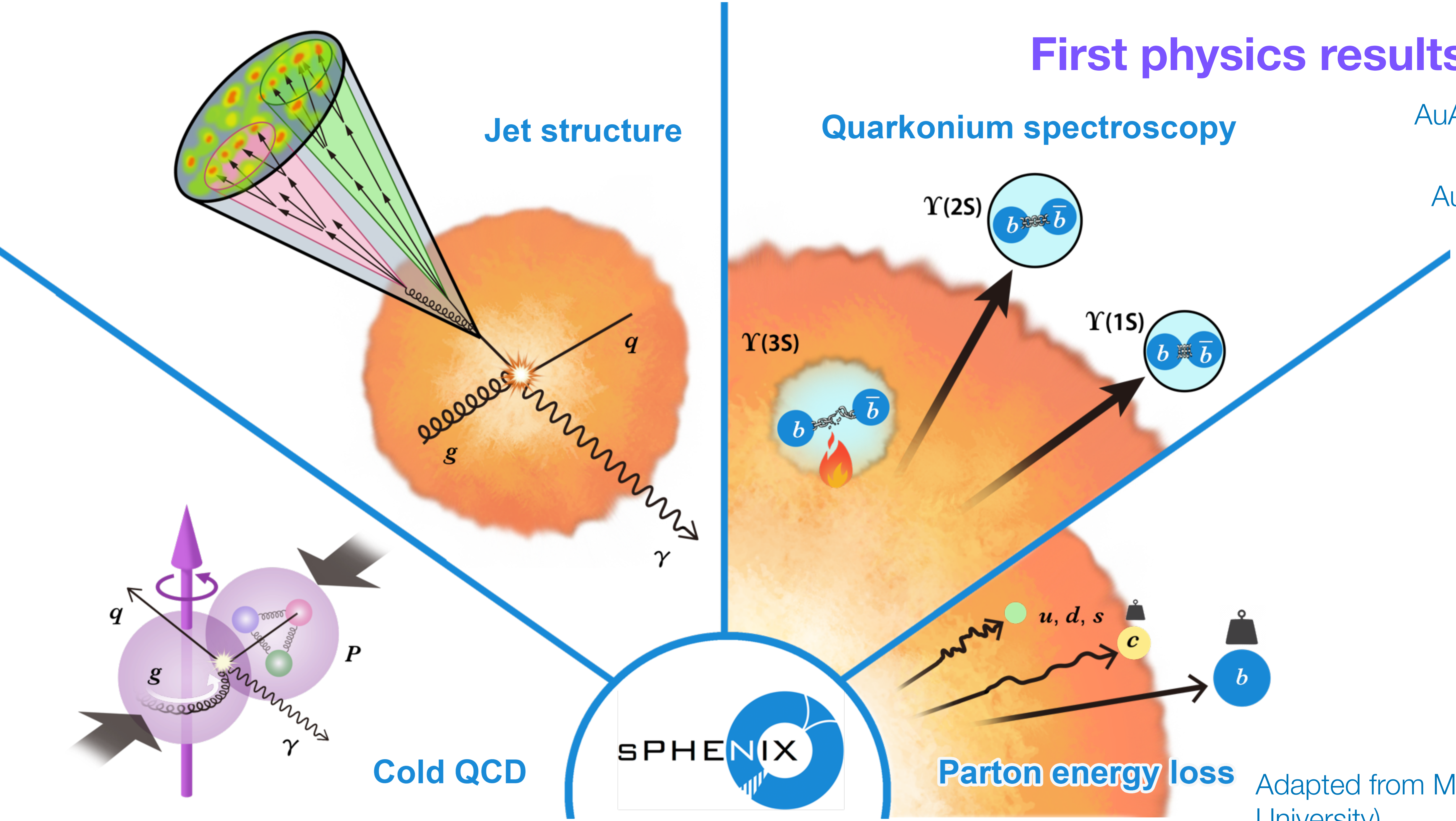
HOW WILL THESE TECHNIQUES BE USED IN THE FUTURE?

sPHENIX Physics Goals

First physics results appearing now!

AuAu charged hadron multiplicity [arXiv:2504.02240]

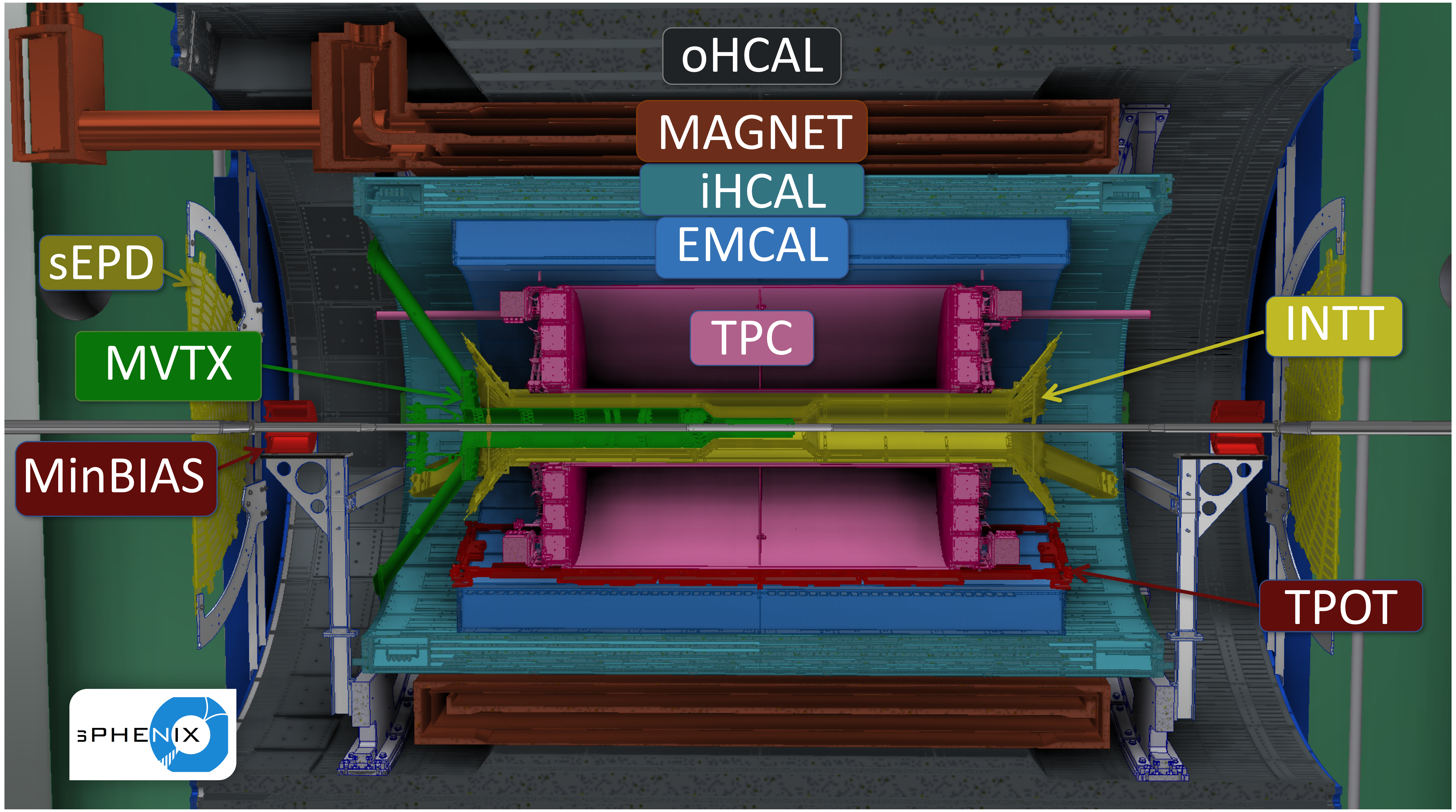
AuAu transverse energy density [arXiv:2504.02242]



Adapted from Misaki Ouchida (Hokkaido University)

sPHENIX Detector

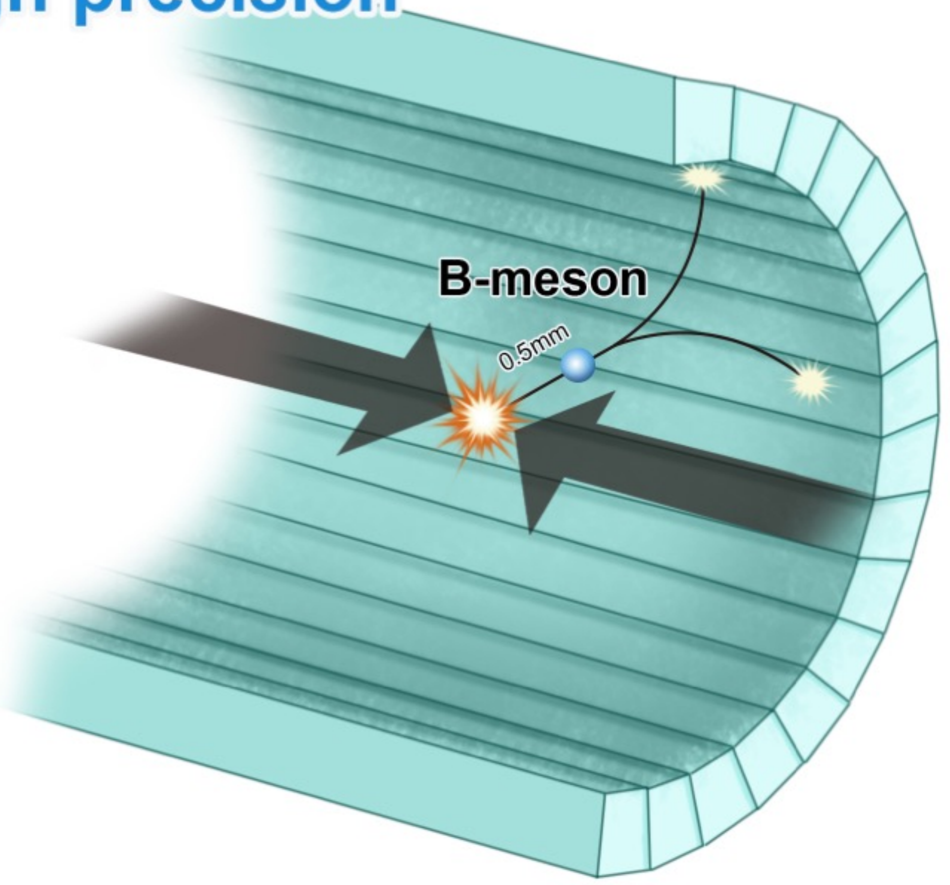
- sPHENIX designed with physics goals in mind!



- High precision tracking and calorimetry key for jet measurements.

- Inner tracking system key for heavy-flavor measurements relying on displaced decay vertex.

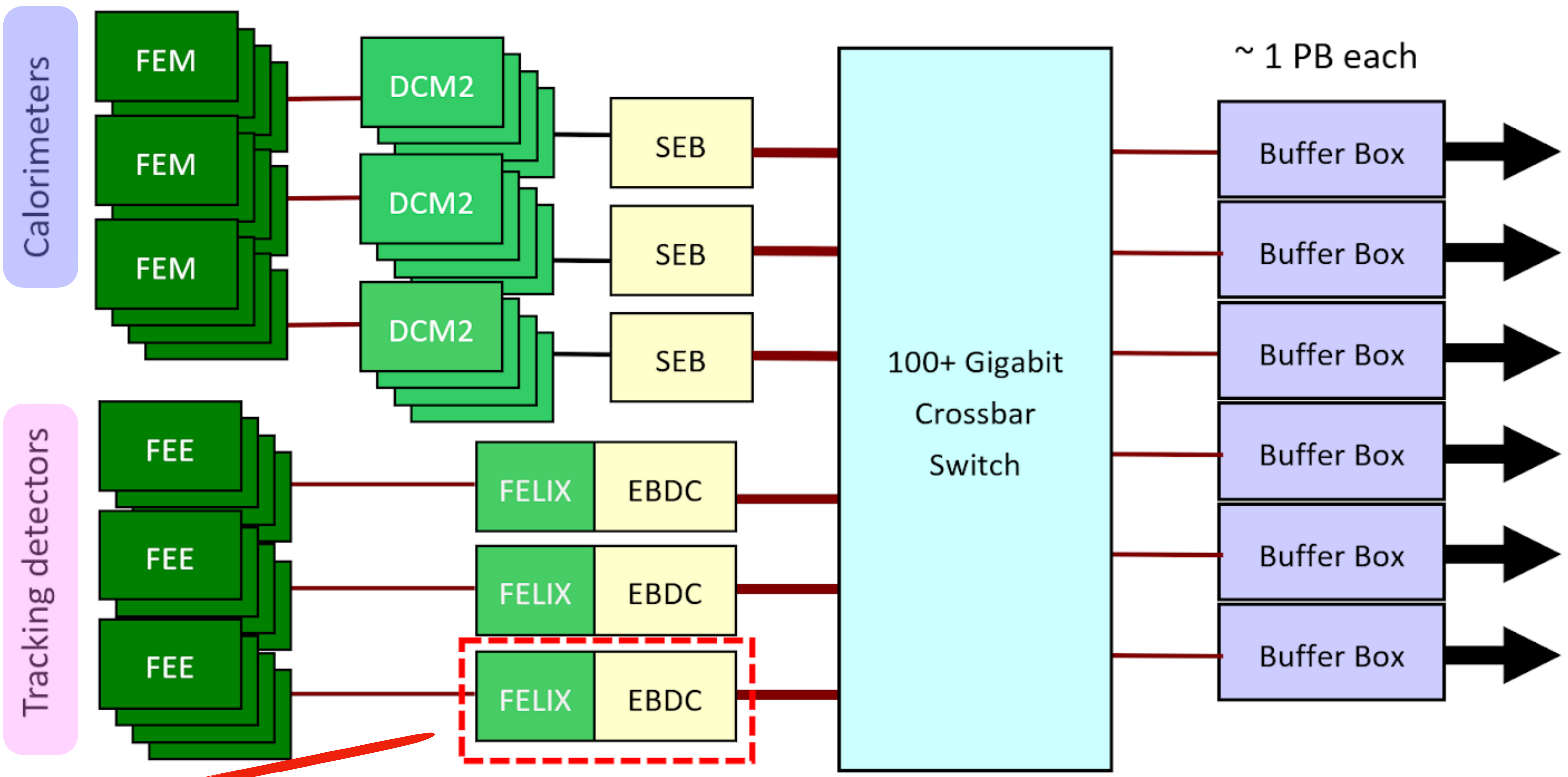
Tracking hadrons and heavy quarks with high precision

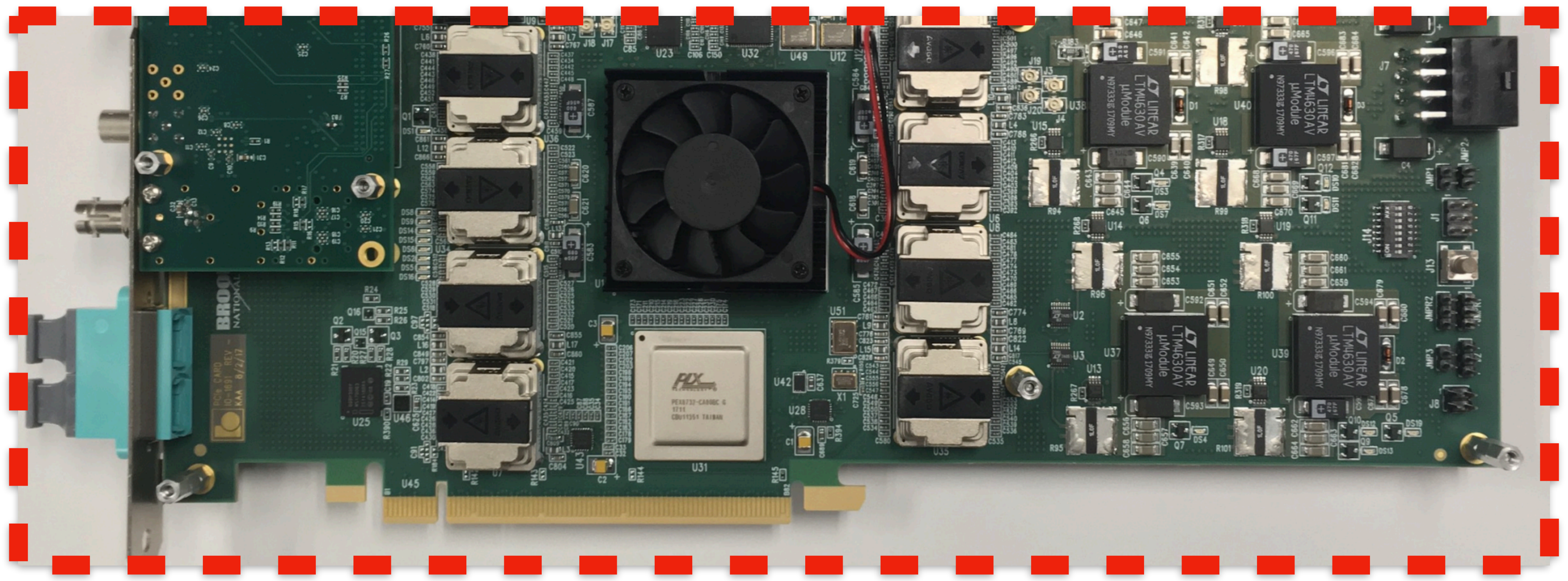


See talk by Alex Patton yesterday! [here]

sPHENIX Data Acquisition

- sPHENIX uses a hybrid data acquisition framework that supports both streamed (trackers) and triggered (calorimeters) readout.
- TPC buffers can hold ~30 μs at a time.





- Readout uses a BNL FELIX 712 board which contains Xilinx Kintex Ultrascale FPGA.
- Useful since it has large amount of firmware and software support.

INCREASING DATA RATES

[BNL Newsroom]

Brookhaven National Laboratory
Newsroom Media & Communications Office

Newsroom | Photos | Videos | Fact Sheets | Lab History | News Categories

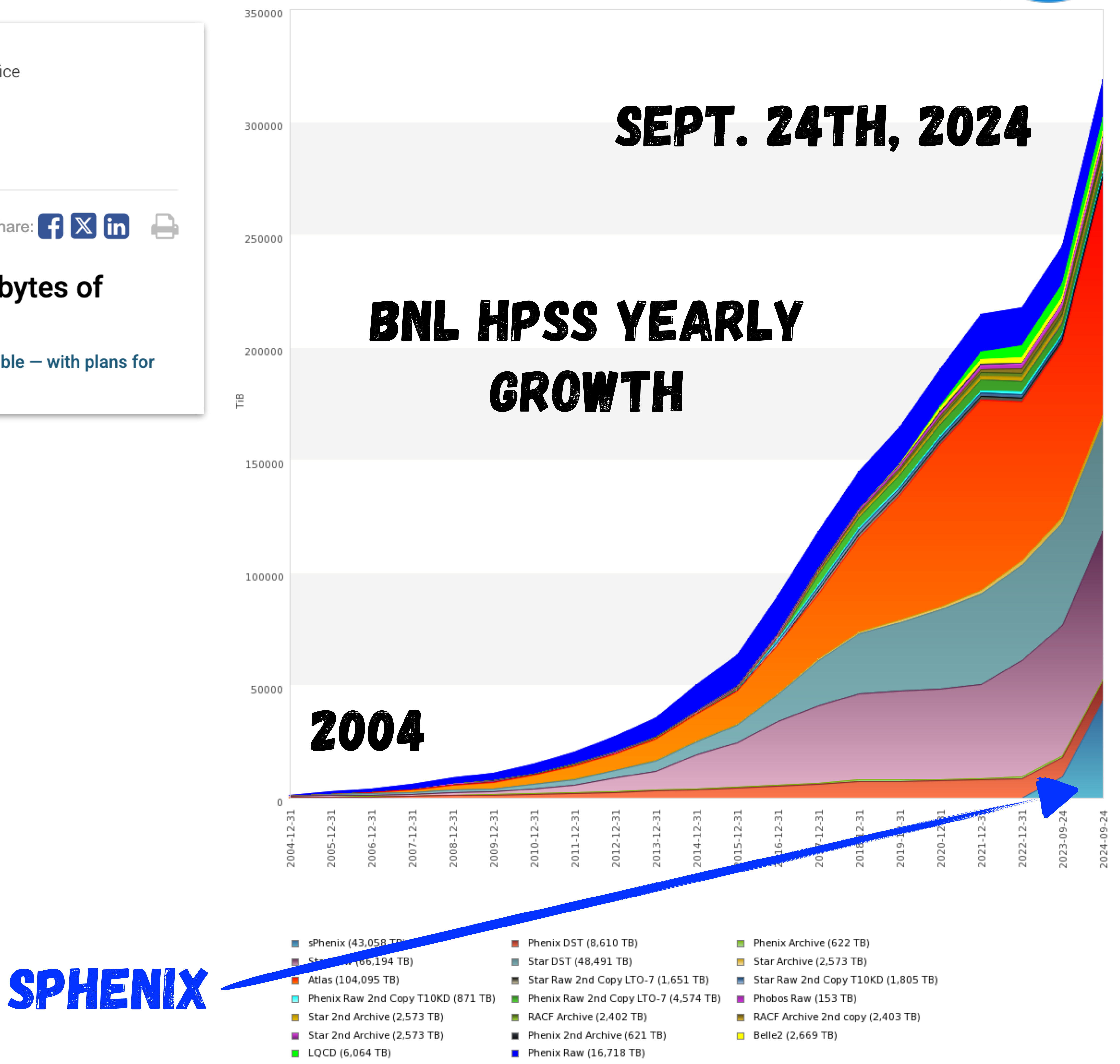
Contact: Karen McNulty Walsh, (631) 344-8350, or Peter Genzer, (631) 344-3174 share:

Brookhaven's Computing Center Reaches 300 Petabytes of Stored Data

Largest compilation of nuclear and particle physics data in the U.S., all easily accessible — with plans for much more

- DAQ rate of sPHENIX (15kHz), shows large improvement in DAQ rate of [PHENIX from 2005 (2.2kHz)].

- Total expected output from sPHENIX ~250 Petabytes
  - Already making up a large fraction of the total storage space.



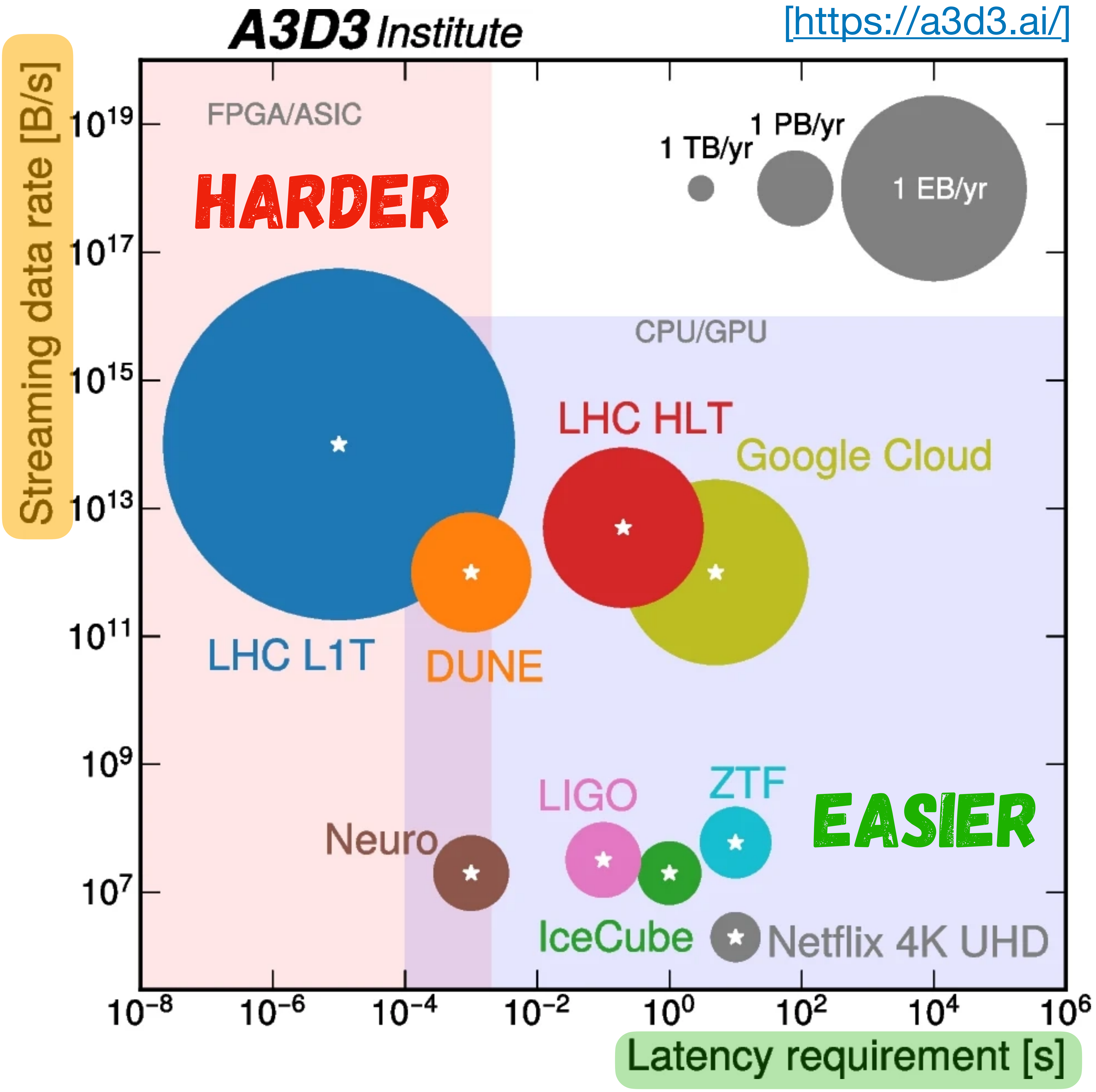
DATA RATE VS. LATENCY

[https://a3d3.ai/]

- Difficulty of a selection problem can be expressed in terms of the rate that data comes in (streaming rate) and how fast the system needs to make a selection decision (latency).

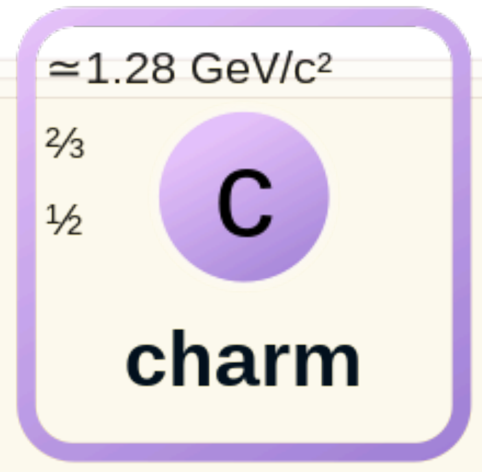
- Having large initial data volumes makes this even more difficult.

- **Physics applications in a more difficult region than most industry applications!**

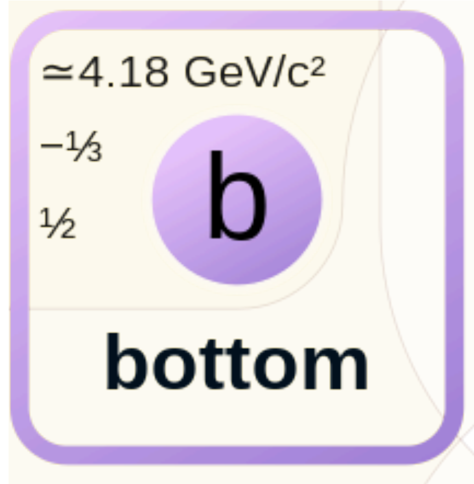


Streaming Readout at sPHENIX

- RHIC collision rate in pp is ~1MHz.
- Calorimeter max DAQ rate is 15kHz (**1.5% of available pp collisions**).
- Avoid this limit by modifying DAQ system to allow triggered readout of tracking + calorimeter plus streaming readout of tracker only.
    - Streaming readout rate is ~30% for TPC (dominates rate), 100% for MVTX/INTT
    - Useful since most heavy-flavor measurements only need tracker information.



Can measure most charm decays (1 out of every 60 collisions)

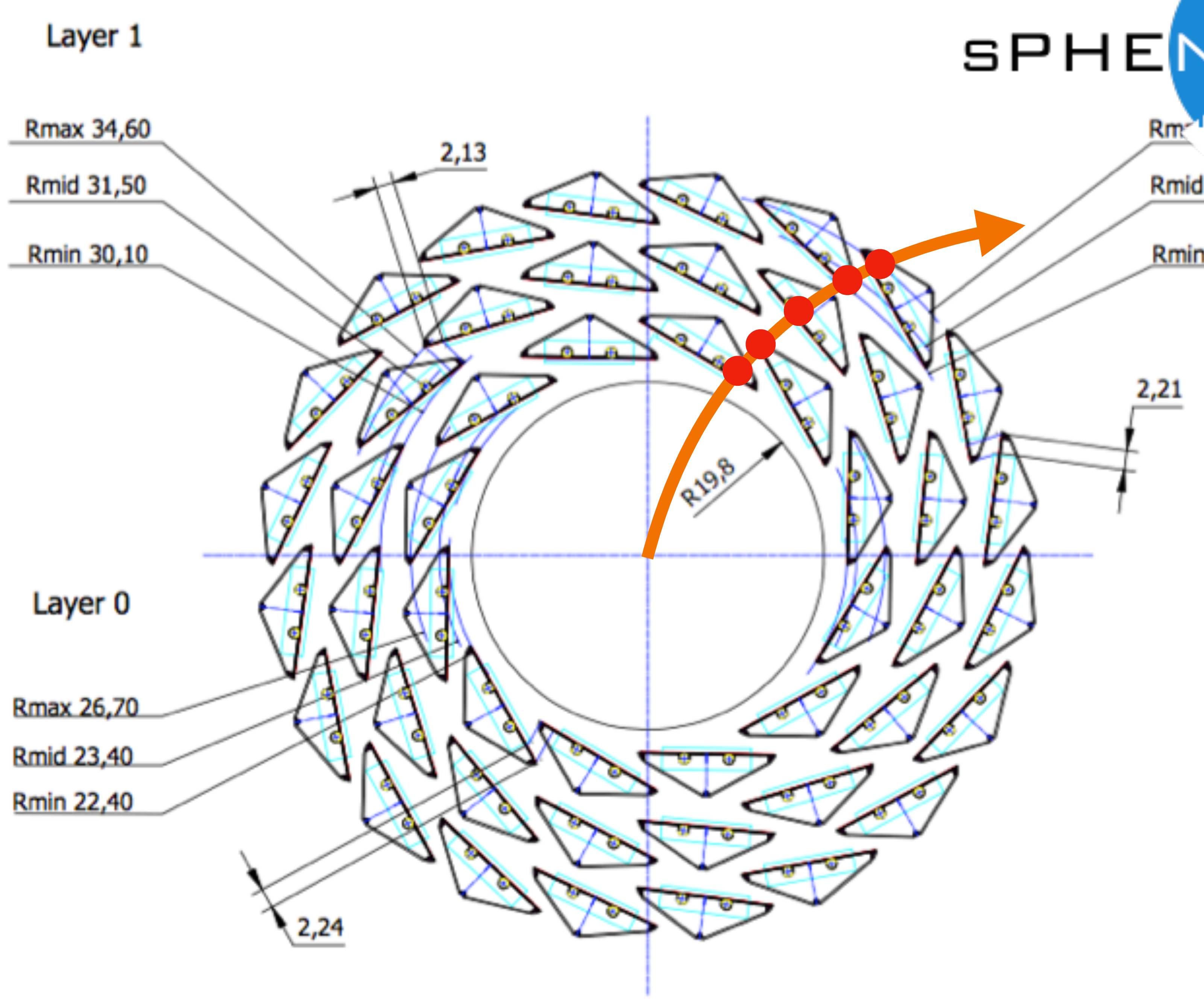


Bottom decays much more rare! (1 out of every 1000 collisions)

https://www.particlezoo.net/

How do we save more heavy flavor decays?

Event Filtering w/ ML



- Perform fast selection/rejection of data with ML integrated into the firmware (FPGAs)

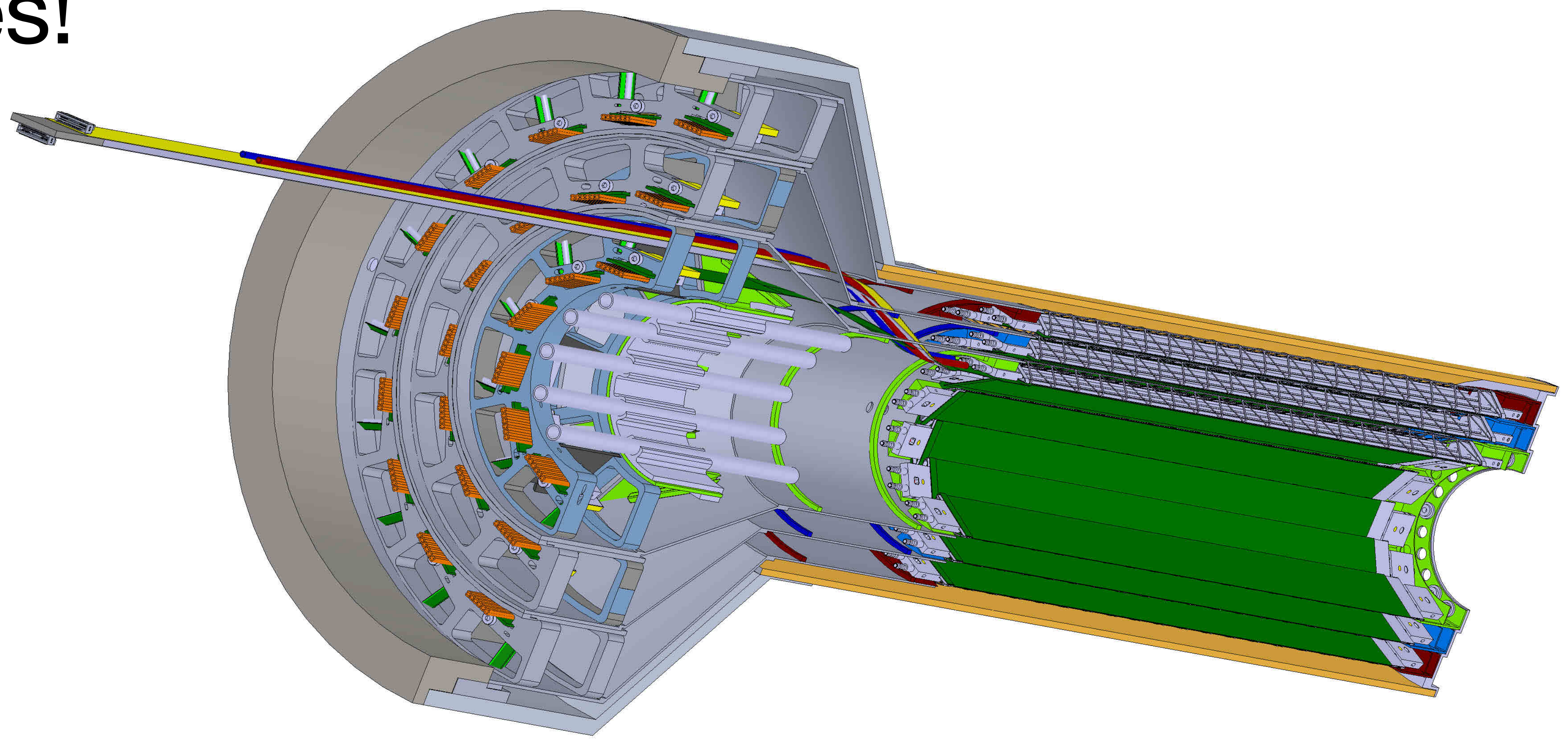
  ATLAS Fake Track Rejection in Event Filter [ATLAS-TDR-029-ADD-1]

  LHCb track reconstruction for HLT system [See website here]

  CMS L1 Trigger [CMS-TDR-021]

  See also: https://fastmachinelearning.org/

- Trackers are great candidates for event filtering studies!
  - Typically many channels, large fraction of data volume.
  - Typically also close to the beam line so are susceptible to beam background and noise.



Can we apply something similar in sPHENIX?

THIS PROJECT AT A GLANCE

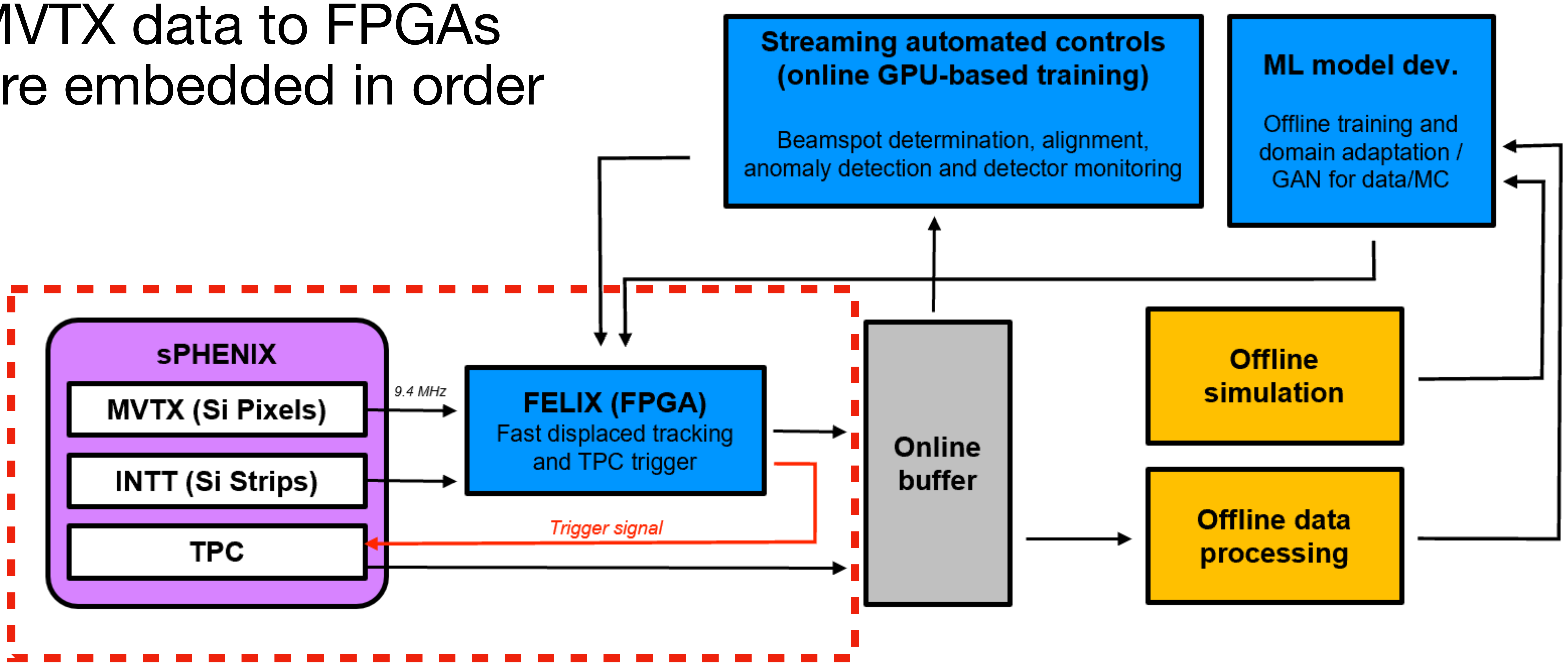
- **Collaboration since 2021 between physicists and computer scientists at LANL, FNAL, MIT, NIJT, ORNL, and GIT!**

- **Goal:** Use ML on FPGAs Increase the statistics available of heavy-flavor decays.

- Will stream INTT and MVTX data to FPGAs where ML algorithms are embedded in order to tag HF topologies.

- Send trigger signal downstream to the readout of the TPC to save data.

- Aim to deliver trigger in 10 μs



DEMONSTRATOR IMPLEMENTATION

sPHENIX

- Implementing trigger decision can be risky, need a demonstrator to show that the output is reasonable! Will use BNL FELIX 712 board.
  - Try two approaches for synthesis
    - hls4ml [arXiv: 1804.06913]
    - Flow GNN [arXiv: 2204.13103]

MLP layer-wise approach (hls4ml)

Data Decoding → Event Building → Hit Clustering → Track Construction → Heavy Flavor Tagging/Triggering

BGN-ST (FlowGNN)

Let's go through this step-by-step for each piece then combine them!

Conventional Approach

Machine Learning

DATA DECODING

Data Decoding

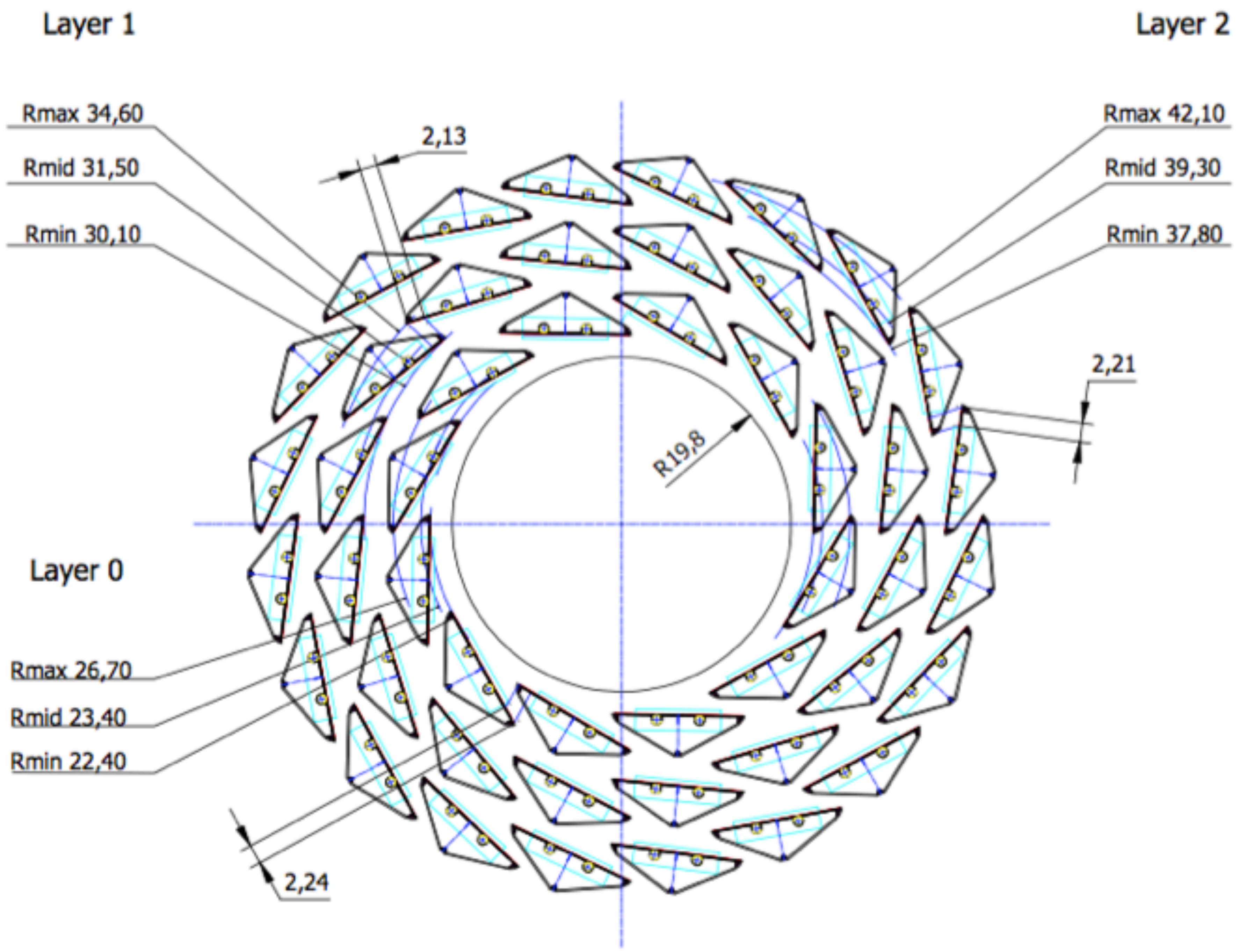
- **For simplicity, start with the MVTX-only case then add in INTT.**

- MVTX consists of 3 layers, 48 staves, with 9 chips per stave with > 500k pixels per chip where each chip's information is sent to its own decoder.

- In p+p collisions, low occupancy (~20 hits per chip per collision).

- Decoding works sequentially where first the layer/stave value, bunch crossing ID (time), chip value and row and column of each active pixel hit is decoded, respectively.

- Implemented directly in VHDL.



Data Decoding

Data Decoding

- Minimal, simplified design with one decoder per detector link (FeeID).

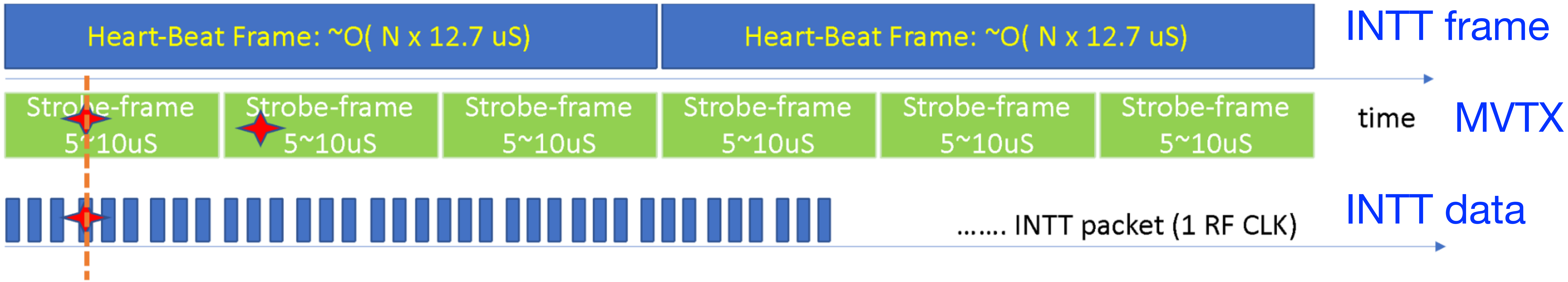
	LUT (663K)	FF (1.3M)	BRAM (2K)
Frame decoder	151	287	0
ALPIDE decoder (x3)	343	256	0
FIFOs (x6)	31	36	1
Total per half-barrel	**98K (14.7%)**	**91K (7%)**	**432 (21%)**

- Efficient utilization of resources!

EVENT BUILDING

Event Building

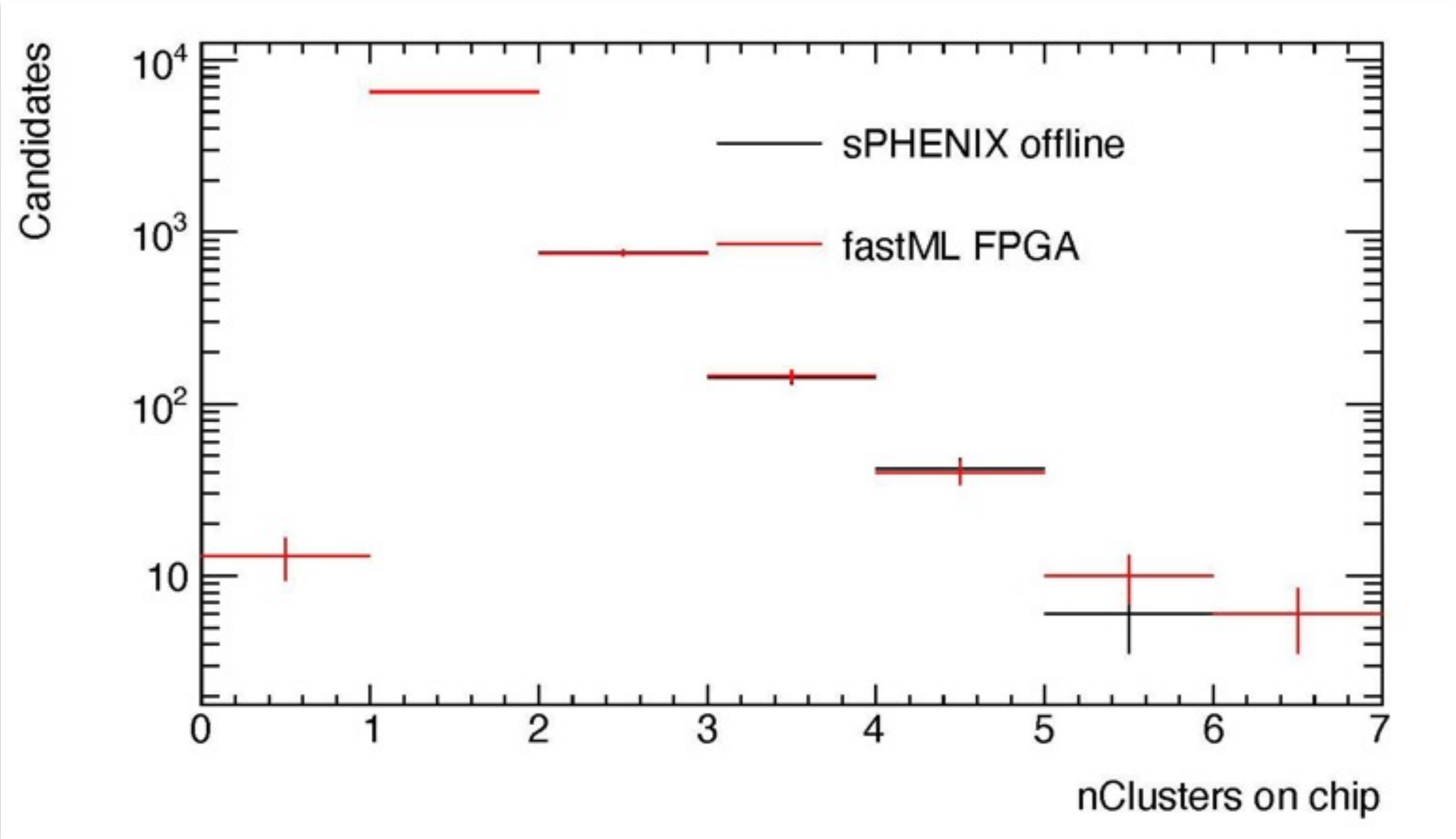
- Primary task of event building is to associate low-level detector information with an event.
  - Eventually need to combine detector information together.
- For MVTX-only approach, bunch crossing ID is included so we can just carry information along the chain.
- When we add in INTT information this becomes much more difficult, due to different streaming configurations.

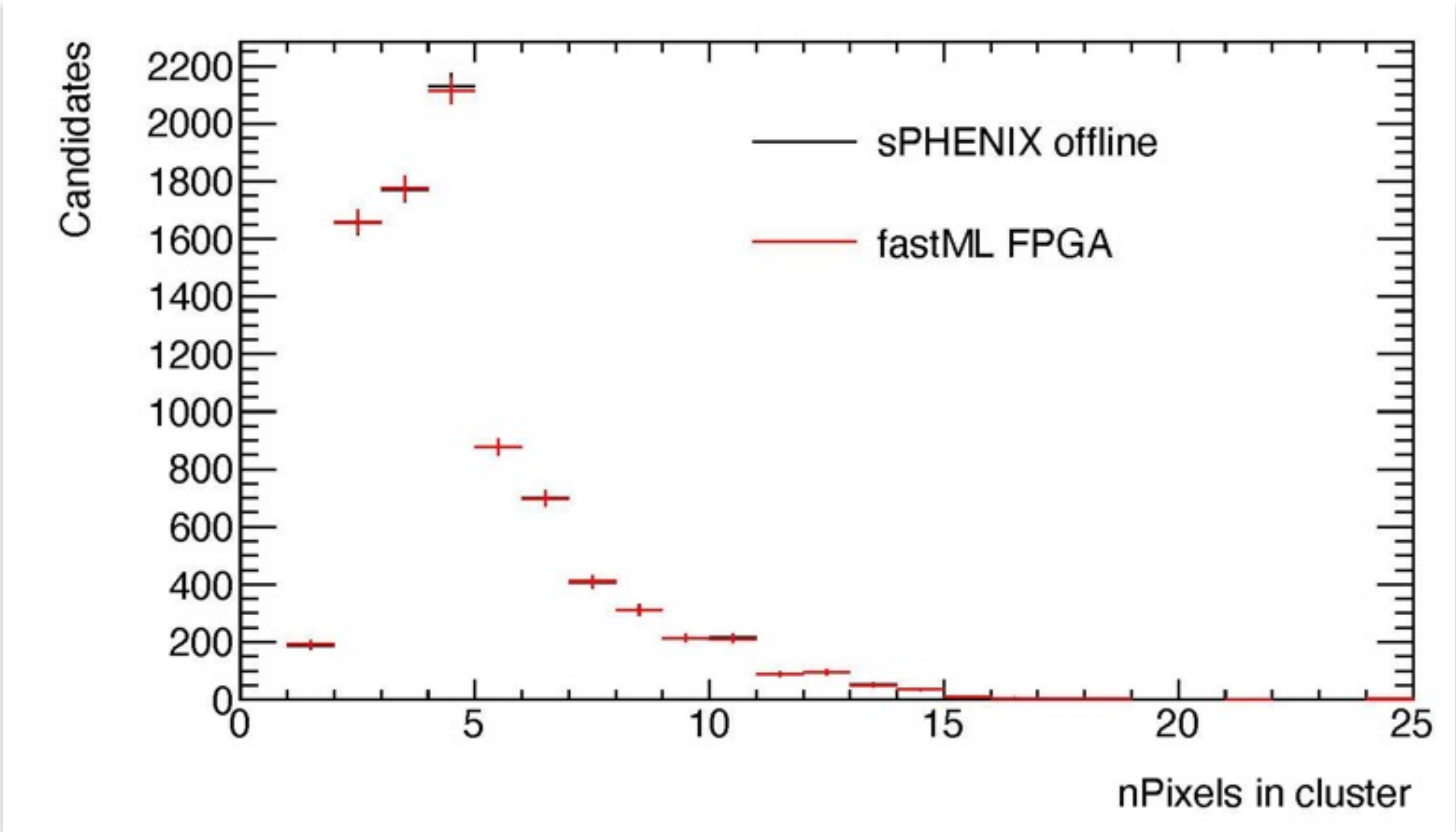


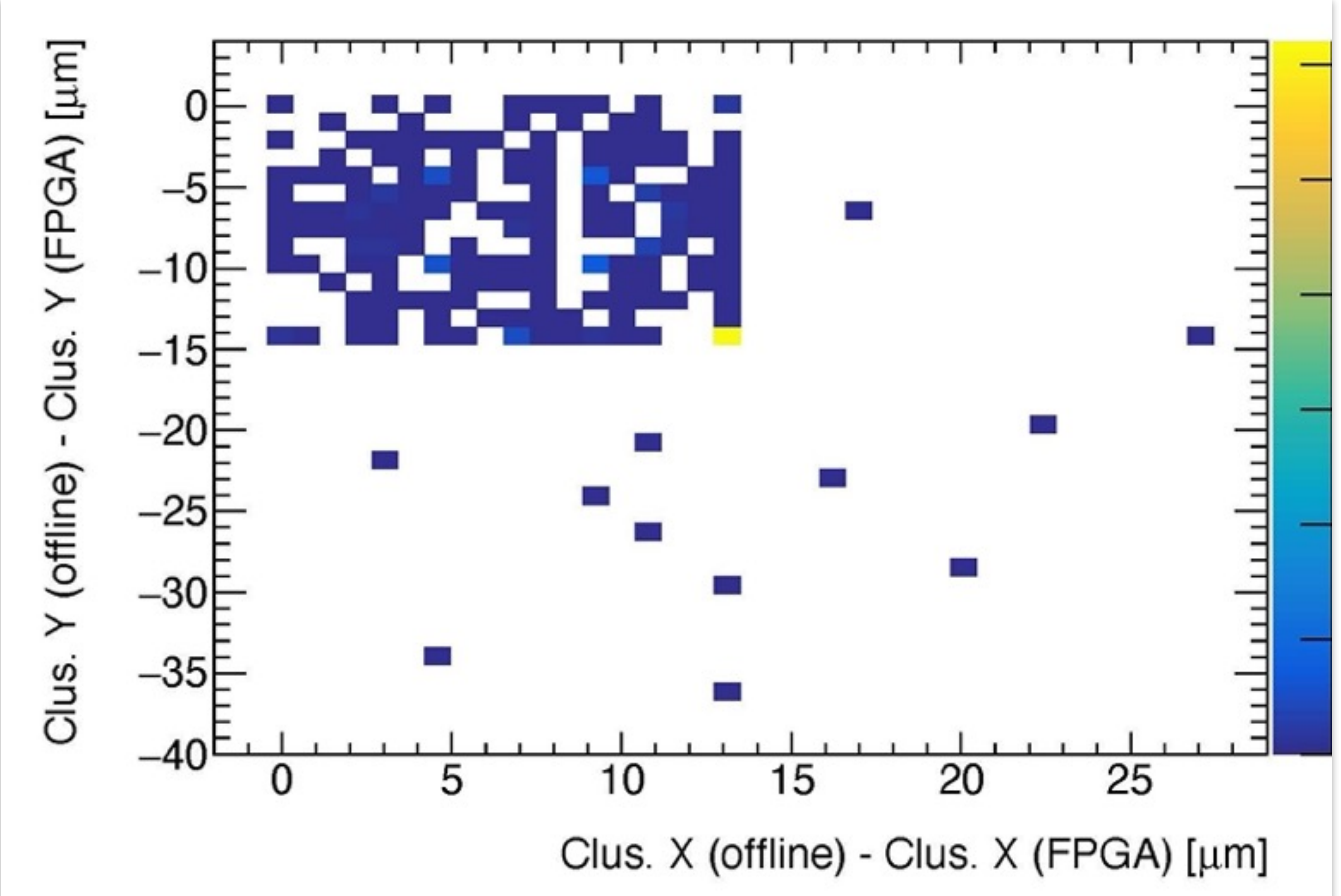
Hit Clustering

Hit Clustering

- ALPIDE chips reads out data in double columns from 0 to 1023.
- Clusters are assembled as they arrive.
- Can achieve 13.5 μm cluster resolution.
- This code was the translated to VHDL using vitis_hls.







# of clusters on a chip **# of pixels in cluster** **Cluster position**

Results show good agreement with sPHENIX offline code!

Hit Clustering

Hit Clustering

sPHENIX

	LUT (663K)	FF (1.3M)
Clustering	3711 + 135 (memory)	2964
per chip (x216)	801K (120%)	640K (49%)
per feeID (x72)	**267K (40%)**	**213K (16.4%)**
per stave (x24)	89K (13.4%)	71K (5.4%)

- Clustering is a very resource-intensive process by nature! Will need to iterate to reduce this!

Coordinate Transformation

- **Output of hit clustering:** layer, stave, chip, row, column
- **Input of the ML algorithm:** layer, r,ϕ, z
  - **Need to do a coordinate transformation!!!**

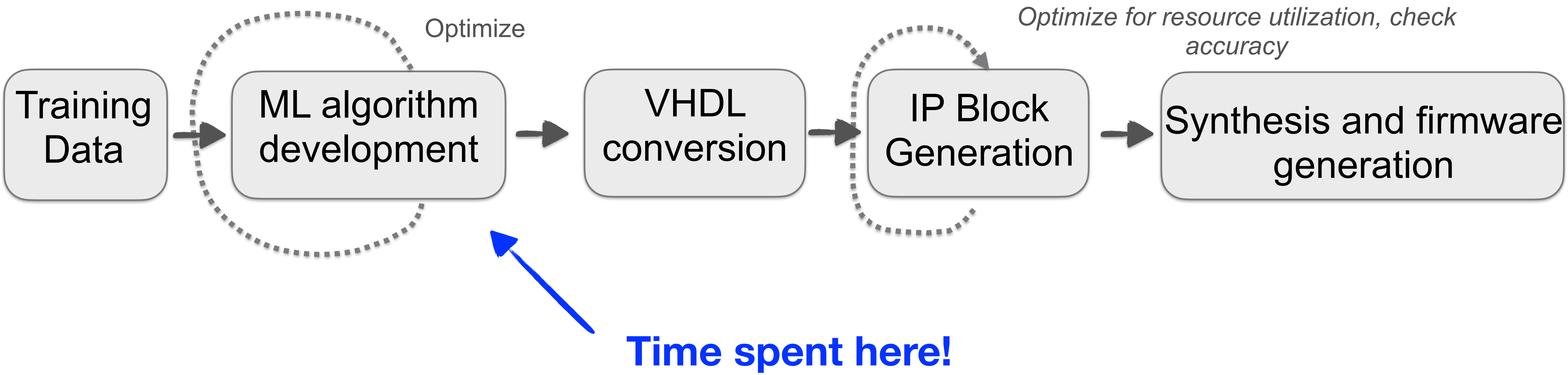
Hit Clustering → Track Construction → Heavy Flavor Tagging/Triggering

	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%)

Coordinate transformation is quite resource intensive!

ML Algorithms on FPGAs

- Previously, all steps done using conventional logic.
- For better heavy-flavor identification, want a more complex algorithm.
- **Still needs to meet the latency and resource utilization requirements!**

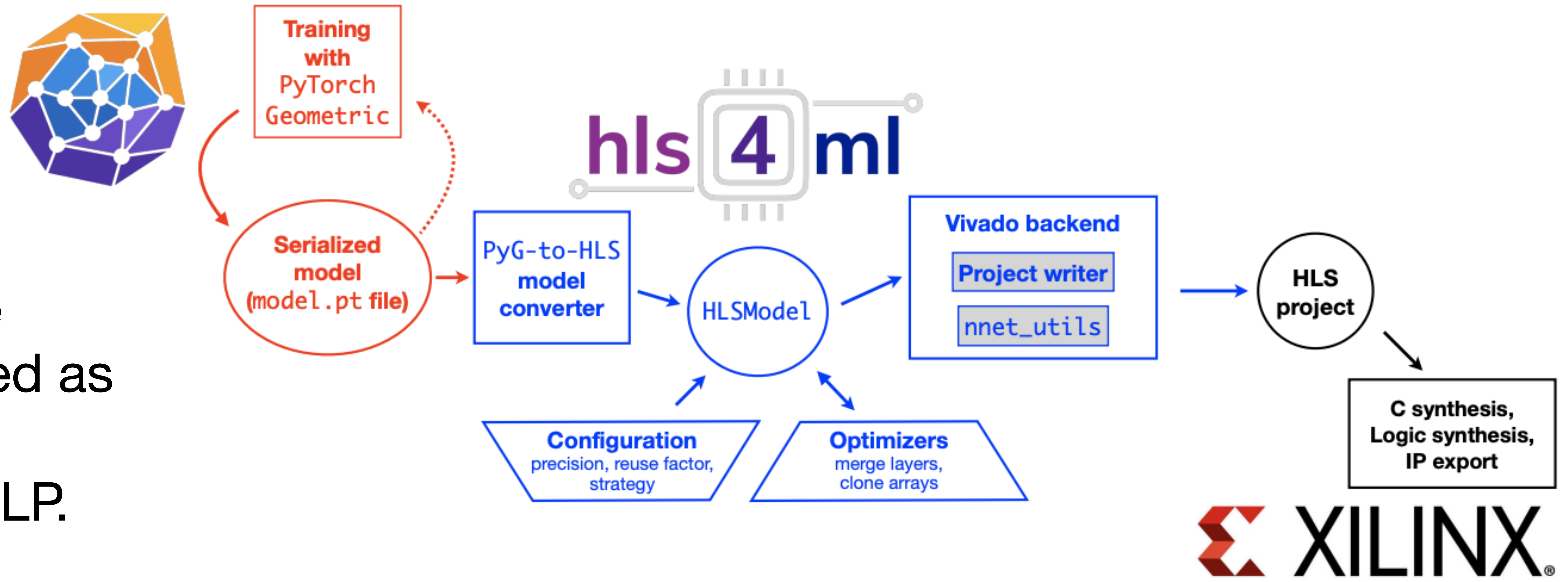


APPROACH #1: HLSML

- hls4ml is a compiler taking Keras, Pytorch, or ONNX input and producing High Level Synthesis (HLS) code implementing the network as spatial dataflow.

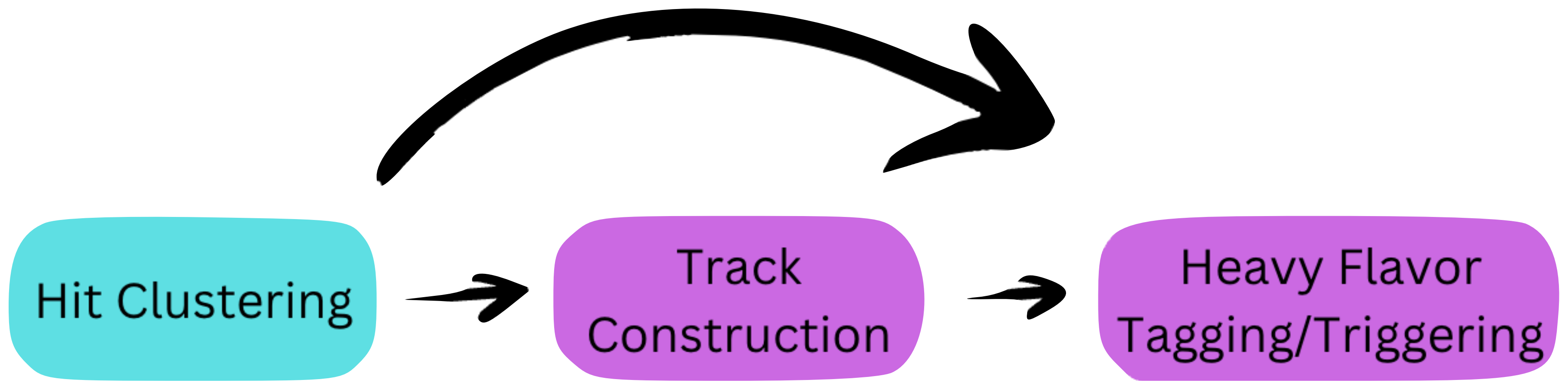
- HLS code is usually C++ or similar with directives to guide the produced hardware.

- 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.
- For HLS approach, use MLP.

APPROACH #1: HLSML



- The approach 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 (II) 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**: II 63 clock cycles, **308 ns latency**
- The two models are synthesized separately, 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%)

- **The MLP-layerwise model has been synthesized for the FPGA!**

APPROACH #2: FLOW GNN

Hit Clustering → Track Construction → Heavy Flavor Tagging/Triggering

sPHENIX

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

- (New this year) Extending to supporting more types of GNNs, e.g., EdgeConv, to facilitate better algorithm support
- (New this year) Perfecting the automation flow from PyTorch → Verilog, based on GNNBuilder [arXiv:2303.16459]

APPROACH #2: FLOW GNN

- Trigger detection using Bipartite Graph Network with Set Transformer (BGN-ST) [DOI: 10.1007/978-3-031-26409-2]
  - Input vectors contain a total of 37 features including: 5 hits (INTT + MVTX), length of each edge, angle between edges, total length of the edges, track radius
  - Not yet supported in hls4ml
  - **97.38% accuracy for b-decays, no pileup**

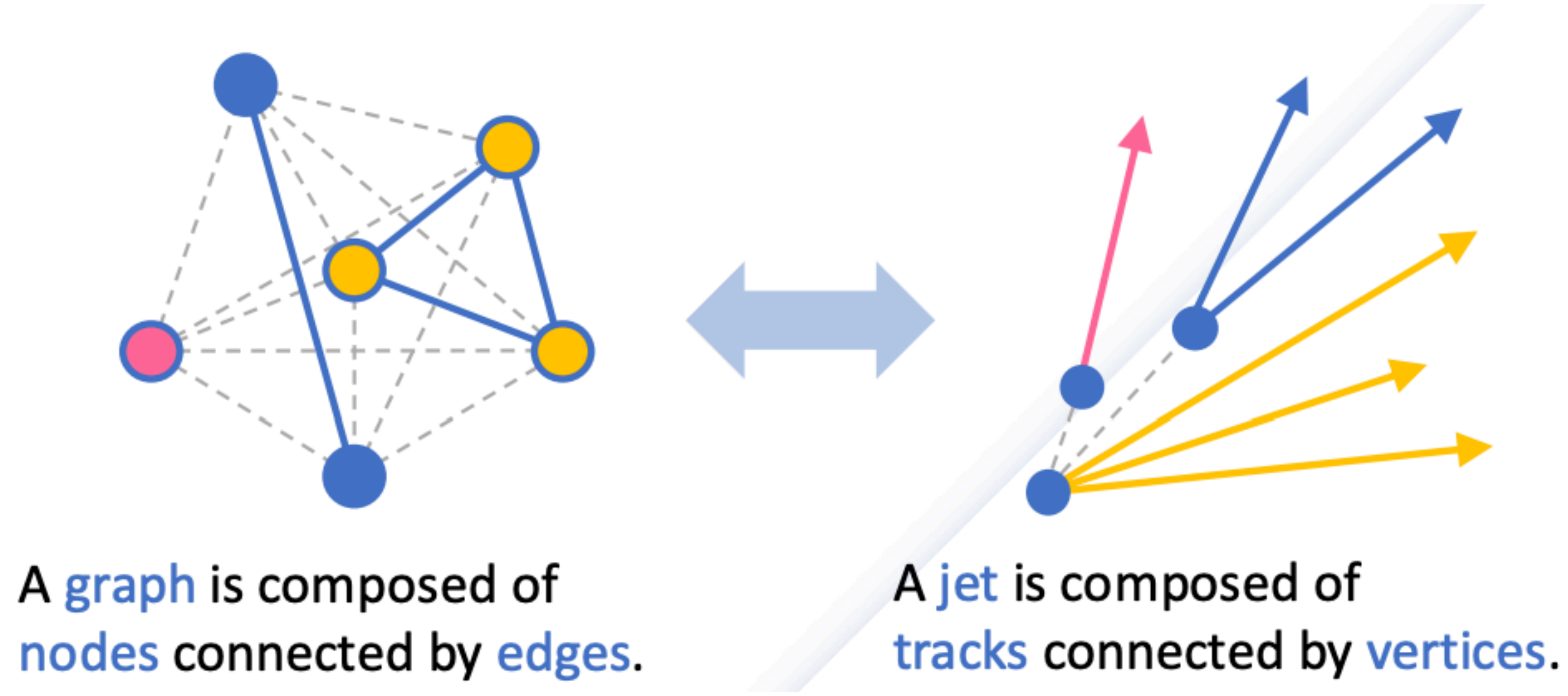
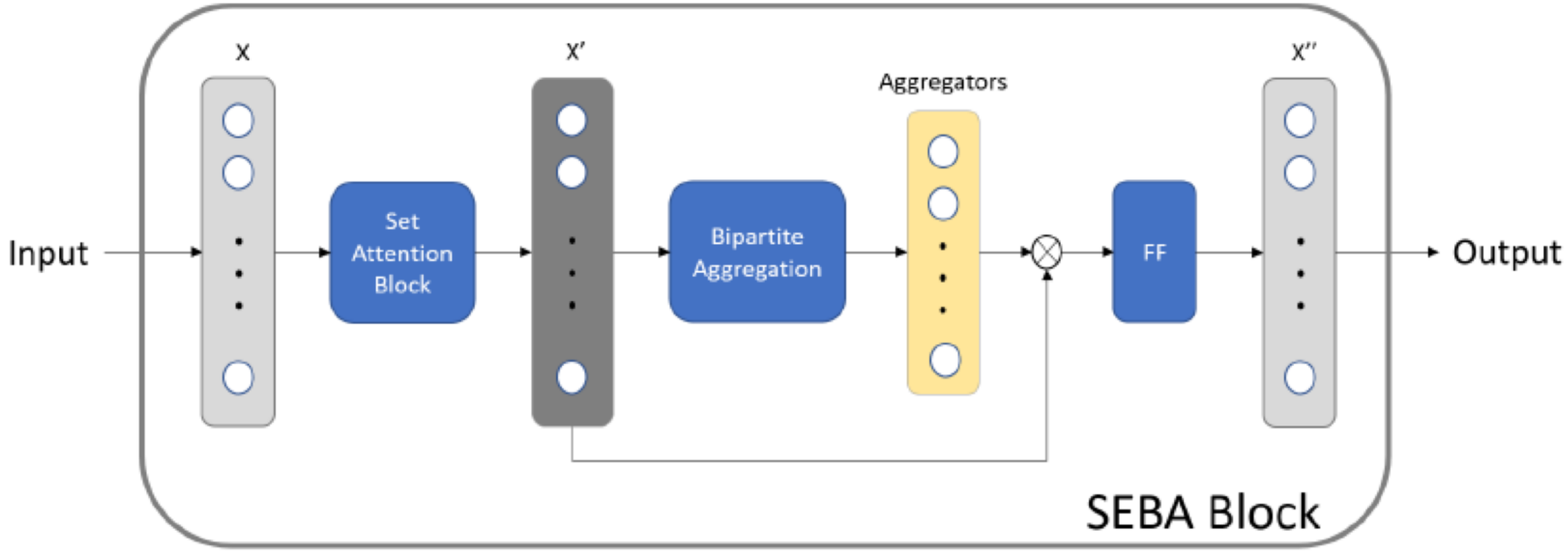


Image from Changwhan Choi, Poster



Putting it all together

- **Currently, single-stave demonstrator implemented in the full chain!**
  - Full-chain established only for MVTX-only version (w/ INTT in progress!)
- Current projections to half-barrel scenario show there are some pieces that require reduction!

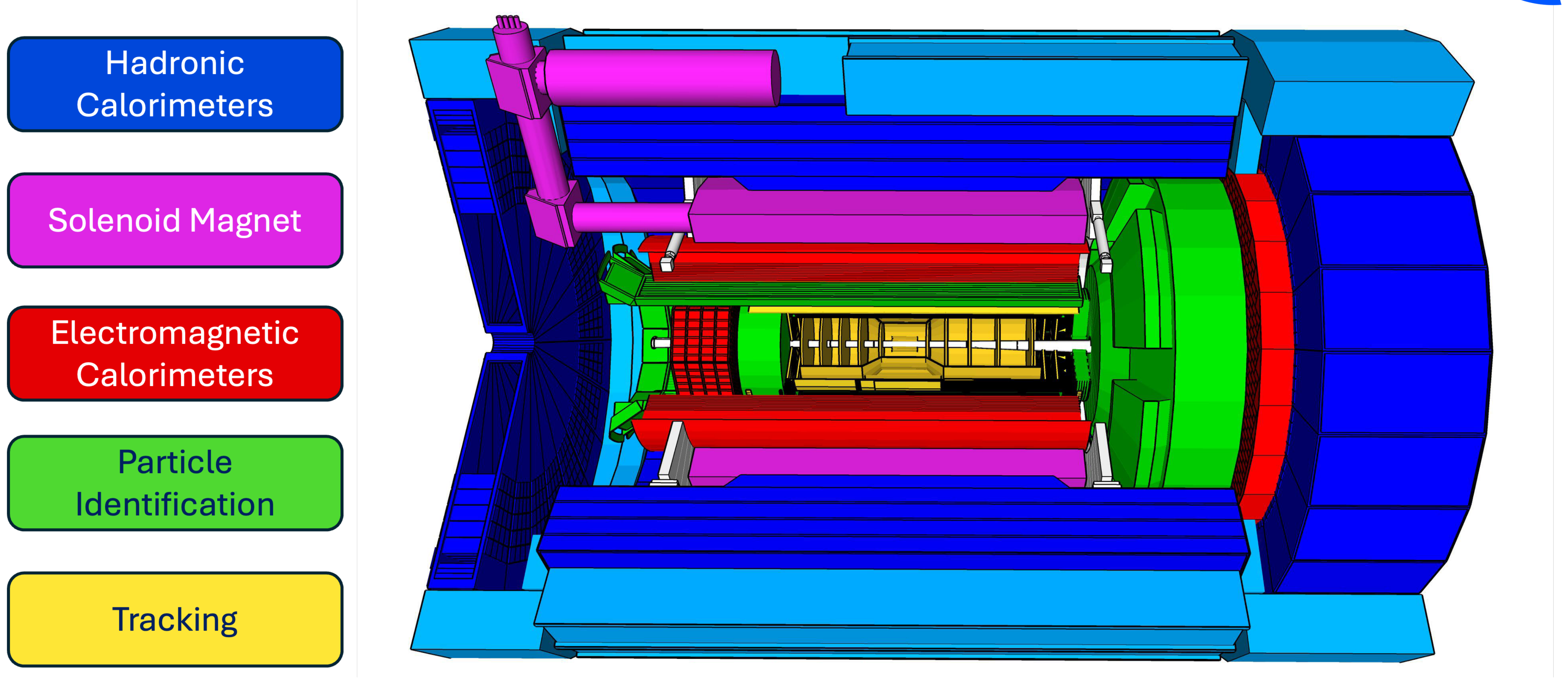
	LUT (663K)	FF (1.3M)	BRAM (2K)	DSP (5.5K)
Infrastructure	87K (13.1%)	196K (14.8%)	879 (40%)	-
Decoder	98K (14.7%)	91K (7%)	432 (21%)	-
Clustering	267K (40%)	213K (16.4%)	-	-
Transformation	25K (3.8%)	22K (1.7%)	540 (27%)	576 (10.4%)
AI module (FlowGNN)	194K (29%)	214K (16.4%)	406 (20%)	488 (8.8%)
AI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

PLAN FOR THE FUTURE

- Main pp physics run of sPHENIX happened in Run 24 last year.
  - Opportunities to test in real-time are limited, can use replicated data-stream.
- Lots of future potential for this work, for example at EIC!

- 100% streaming, but can use ML to tag HF events to reduce computation needed for analysis.

- Other possibilities for use include data reduction, filtering of beam gas events, etc.



- **Need to develop technology and methodology now - these techniques will be crucial in the future!**

Conclusions

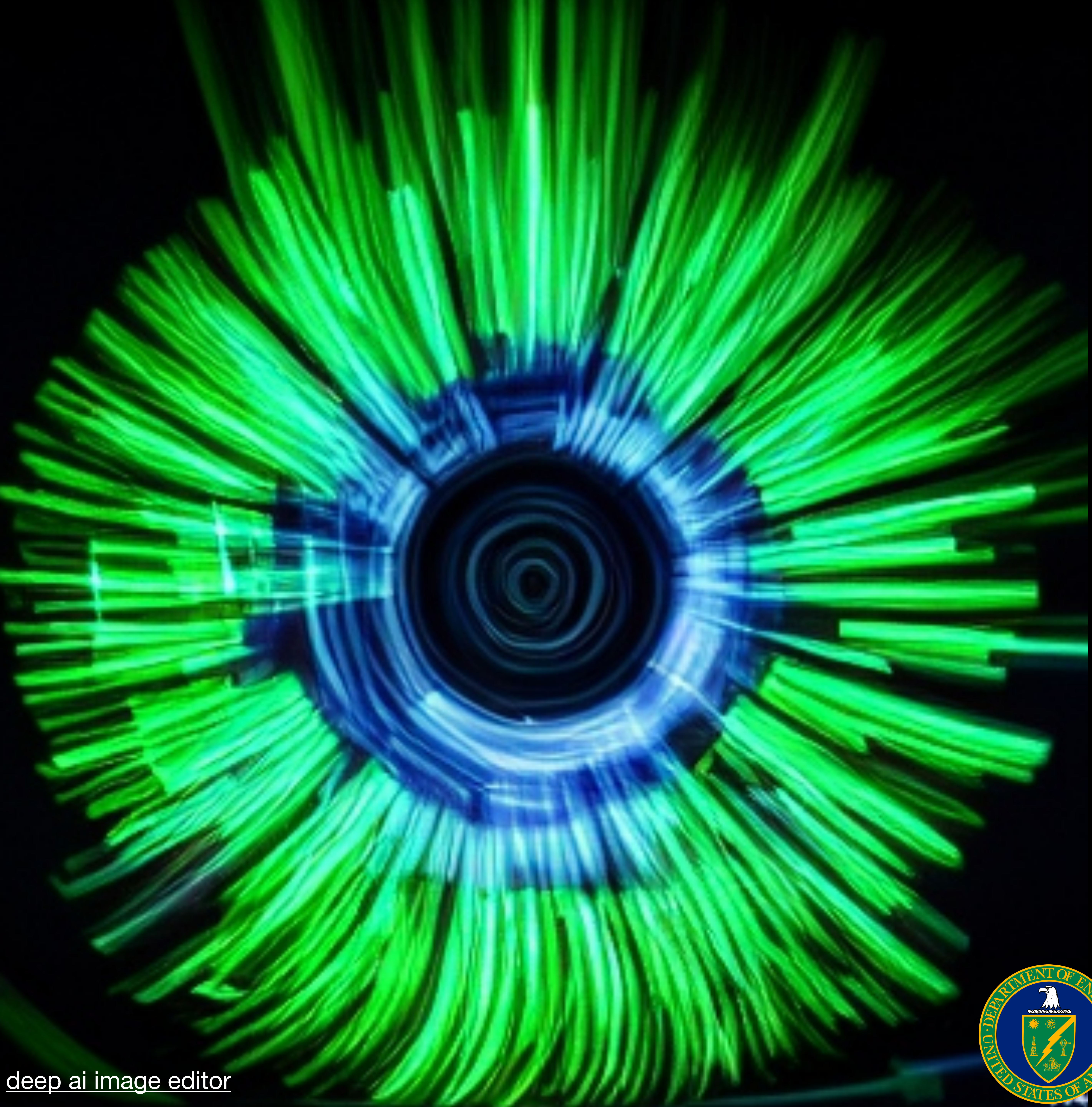
- ML has the potential to revolutionize our approach to collecting, reconstructing and understanding data, and thereby maximizing the discovery potential in the new era of nuclear physics experiments.

- In this project we use ML algorithms that are embedded onto FPGAs in order to to tag heavy-flavor event topologies using streamed data from the inner trackers (INTT + MVTX) of sPHENIX.

- This is beneficial as it promises a dramatic increase for the amount of available data for heavy-flavor analyses, crucial to the physics program of sPHENIX.

- **All pieces of the pipeline implemented and a single-stave chain has been created! Can continue to be tested/improved!**

Thanks! Stay tuned!



deep ai image editor

Backup

MIT HIG's work was supported by US DOE-NP

Laboratory for Nuclear Science