ML-enabled End-to-End Tracking Reconstruction and Trigger Detection

Supported by DOE grant DE-SC0019518

Yu Sun, PI
Sunrise Technology Inc.
yu.sun@sunriseaitech.com
2024 DOE SBIR NP Exchange Meeting
August 15th, 2024

About Sunrise Technology Inc.

- Founded in 2017
- Located in an incubator at Stony Brook University, Long Island, NY
- Team: three full-time employees, a part-time consulting scientist, and several graduate interns.
- Developing advanced AI/machine learning technology for autonomous systems, such as scientific experiments decision-making engines and education platforms.
- Projects

1) ML-based slow orbit feedback control, deployed at BNL NSLS-II in July 2023

2) Autonomous driving toolkit for AI education

3) ML-based particle collision triggering system

4) Terabits data transfer toolset for distributed data analysis

SBIR Phase II Objectives

- SBIR Phase II award
  - Title “High Performance FPGA-based Embedded System for Decision Making in Scientific Environments”
  - Co-funded by NP and ASCR
  - End Year 4

- Ultimate Goal
  - Design real-time AI-enabled DAQ trigger algorithms applied to the very high-rate data streams from detectors.
  - Play a central role in sPhenix and future EIC detectors running under trigger systems and in-situ streaming analysis for event selections.

- Phase II Technical Objectives
  - Designing Graph Neural Networks for High-Speed Physics Event Triggers.
  - Collaborate with sPhenix team to integrate the algorithms to sPhenix experiment and reaches the target of 15Khz data acquisition rate.

- Phase II Commercialization Objective
  - Manufacture smart embedded system to facilitate real-time data collection for experiment and facility control

Team on this project


Yu Sun, PI


Giorgian Borca-Tasciuc


Kevin Mahon


Tingting Xuan


Yimin Zhu

Collaborators

- Dr. Ming Xiong Liu, Dr. Cameron Dean, LANL
- Dr. Jin Huang, Dr. Zhaozhong Shi, BNL

Motivation

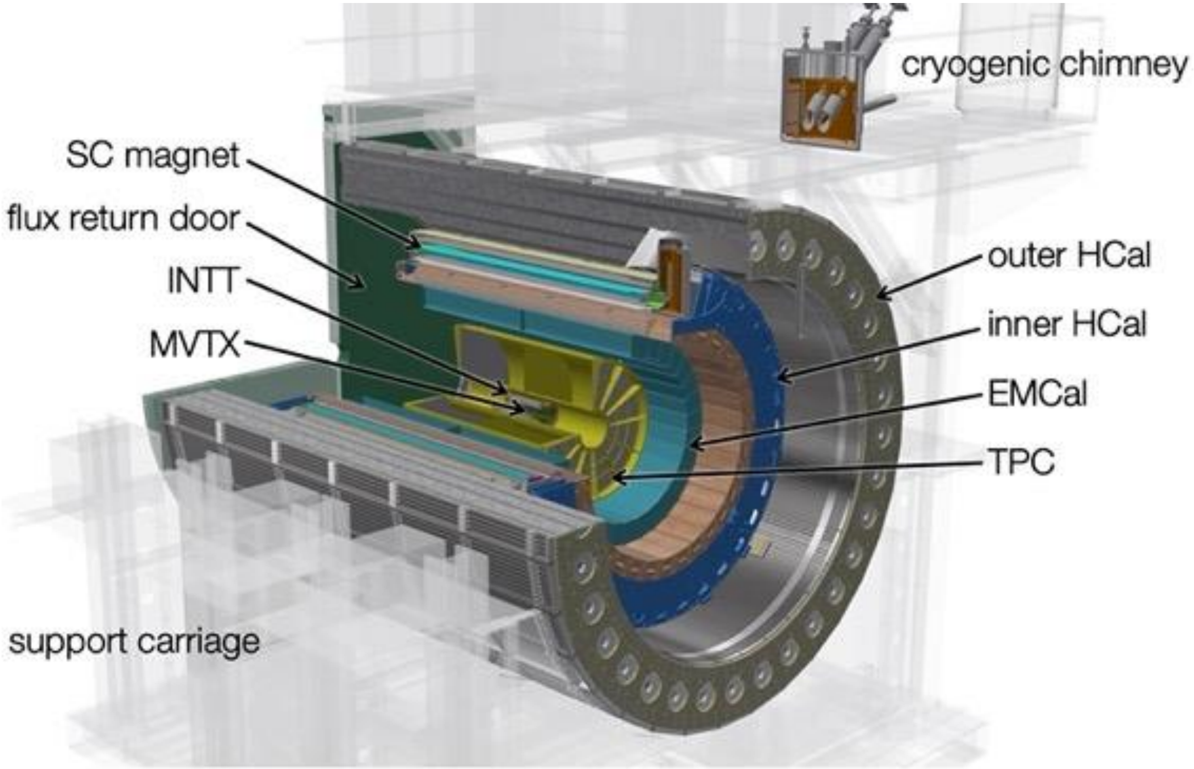
- The readout challenge
  - Raw data Speed and Volume >> Hardware bandwidth/Storage Capacity

    Only a small fraction of data will be recorded to tape
- Trigger events are very rare, ~0.1% probability at RHIC
  - RHIC collision rate is several MHz, sPHENIX readout 15 kHz

  -Without an effective trigger algorithm, experiments must use random event taking.

  - With the same level of recall, AI-based trigger will significantly improve the detector efficiency.
- Integrate the AI-based trigger system into the sPHENIX experiment for p+p run in 2024
- Potential future deployment on Electron-Ion Collider (EIC)

sPHENIX experiment



sPHENIX experiment under construction at RHIC:

- Running period 2023-2025
- ~4m long, ~3m high, 1000 tons
- 15kHz trigger rate
- 3 MVTX layers and 2 INTT layers - detectors capable of streamed readout

Approaches

- There is a trade-off between latency (prediction speed) and accuracy (prediction performance)
- Longer pipelines enable more sophisticated data processing and higher accuracy, but at the expense of inference speed
- As the details of the hardware implementation remains a moving target, we develop several pipelines to cover various points on the latency-accuracy trade-off frontier

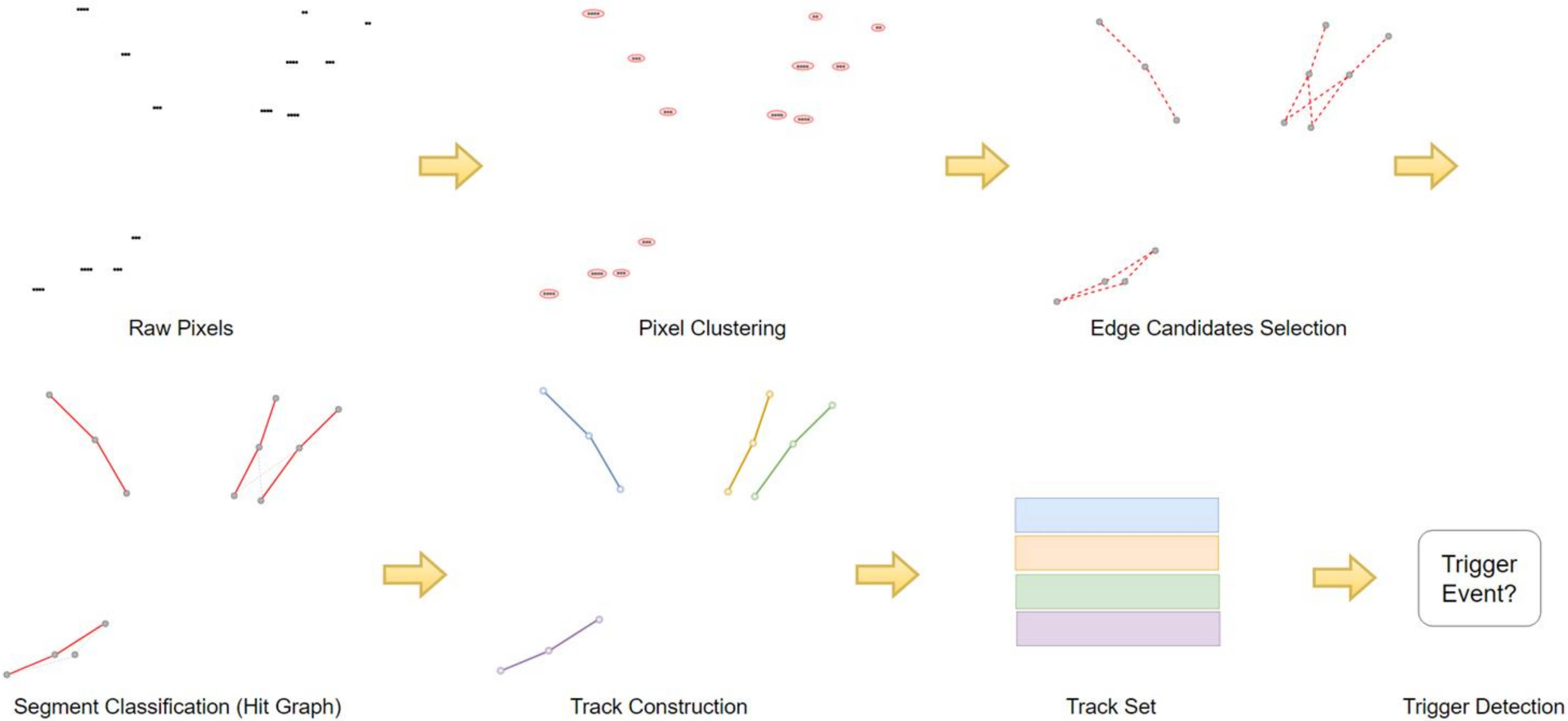
Pipeline Stages

- Each of the pipelines developed is composed of one or more of the following stages:
  - *Pixel Clustering*: Contiguous clusters of activated pixels are found and collapsed to a single point, called a *hit*.
  - *Edge Candidate Selection*: A graph is constructed on the set of hits by using geometric constraints to select pairs of hits that are likely to come from the same particle.
  - *Segment Classification:* Edge candidates are classified using a Graph Neural Network (GNN) to only keep the edges connecting hits that really do come from the same particle.
  - *Track Construction*: Connected hits are grouped together to form the trajectory (track) of the particle as it flies outwards from the detector center. This leads to a set of tracks.
  - *Trigger Detection*: The data at this point of the pipeline is processed by a GNN to predict whether the event is a trigger event.

Pipelines

- We have developed the following pipelines:
  - *Track-Set Pipeline:* Pixel Clustering ➔ Edge Candidate Selection ➔ Segment Classification ➔ Track Construction ➔ Trigger Detection
  - *Hit-Graph Pipeline:* Pixel Clustering ➔ Edge Candidate Selection ➔ Trigger Detection
  - *Hit-Set Pipeline:* Pixel Clustering ➔ Trigger Detection
- Each of these pipelines realize a different point on the latency-accuracy curve.

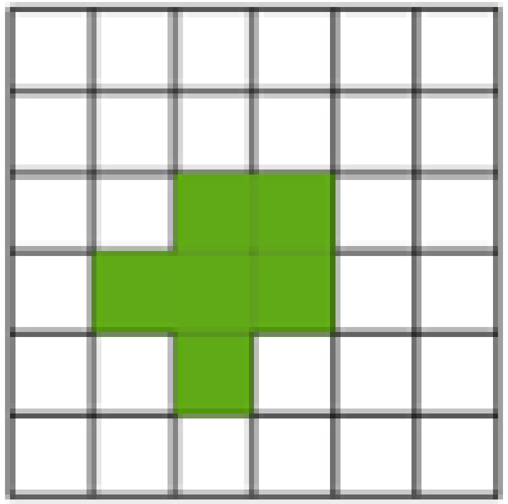
Track-Set Pipeline Overview



Pixels $\mapsto$ Hits

From Pixels to Hits - Clustering

- Clustering is done by solving a spanning forest problem
- There is an edge between pixels that are adjacent to each other
- Mean of all pixels in a cluster is taken as the hit location

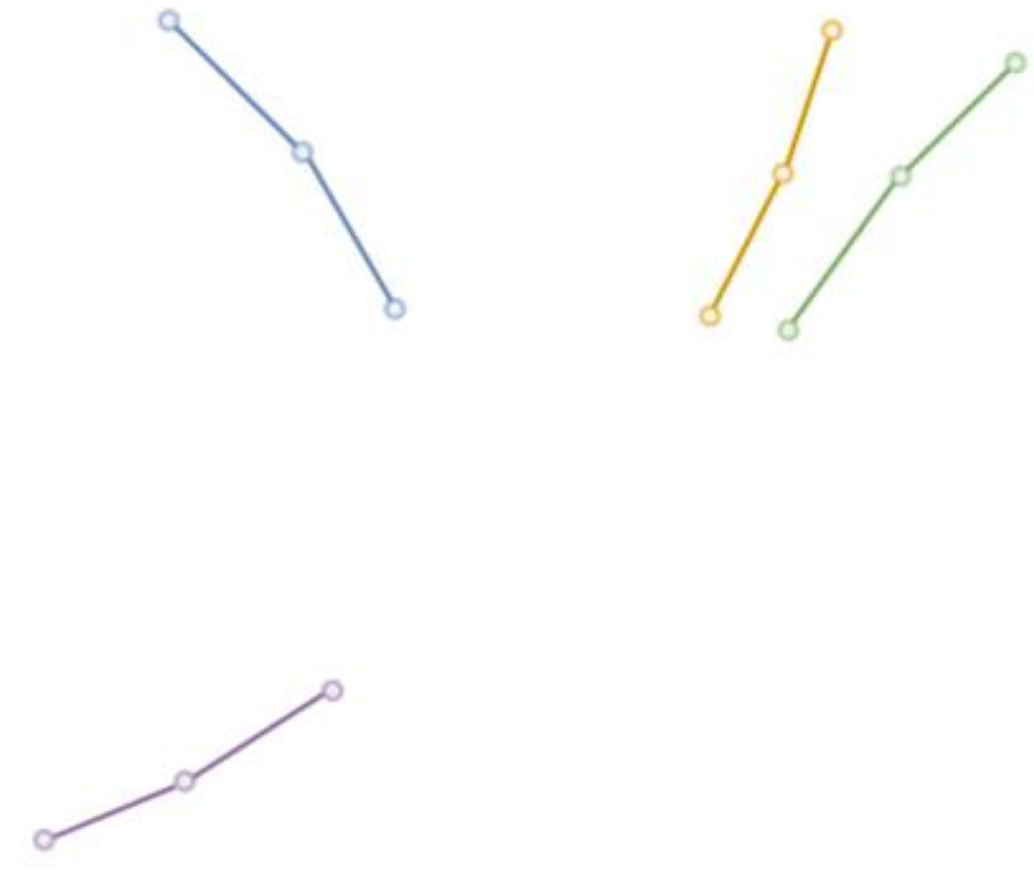


Pixels on Detector

Hits $\mapsto$ Tracks

From Hits to Tracks

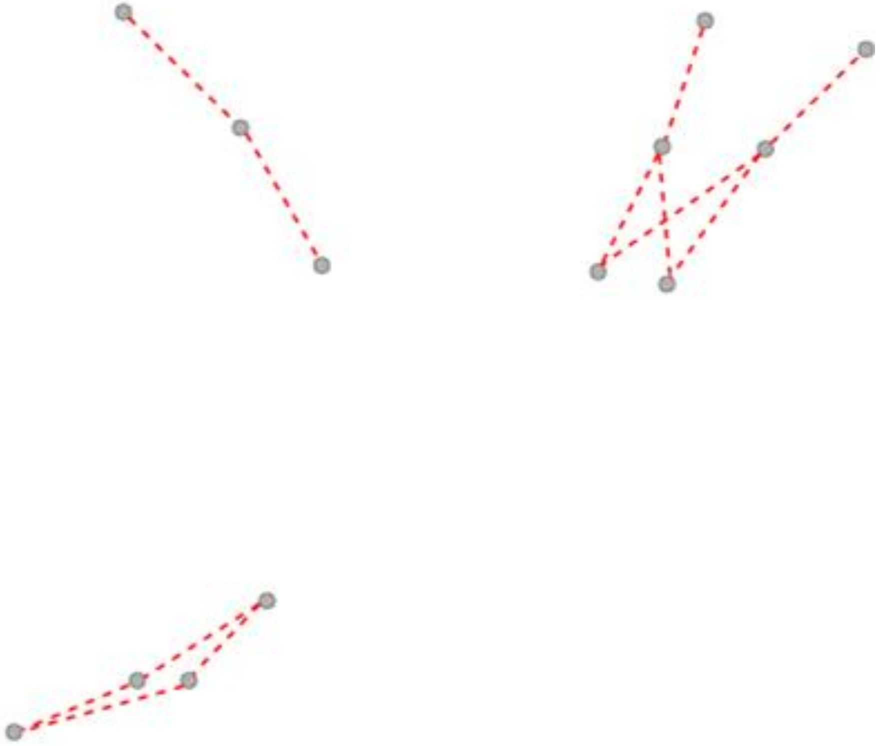
- Once we have hits, we want to group hits that came from the same particle into a track
- This will be solved by treating the problem as an edge classification problem
- Out of the N^2 possible edges between the hits, we want to know the true edges.



Track Construction

Edge Candidate Selection

- Not all of the N^2 possible edges are plausible - we can eliminate a lot of edges from the get-go
- We can use some basic geometric constraints on the cylindrical coordinates of the hits
  - $|\Delta\varphi/\Delta r|$ <= PHI_SLOPE_MAX
  - $|z_0|$ <= Z_ORIGIN_MAX
  - $z_0 = z_1 - r\cdot(\Delta z/\Delta r)$
- The geometric constraints determine the number of candidate edges and affects the latency and will play a vital role in further reducing the FPGA's latency.



Edge Candidates Selection

Track Construction

- Once edge classification is performed, a track is constructed by finding the connected components

Track Construction Performance

	2022	2023
Accuracy	**96.30%**	92.07%
Precision	84.55%	**92.54%**
Recall (efficiency)	83.25%	**97.97%**
F1	83.89%	**95.18%**
Latency	**17.92µs**	**3.1725µs**

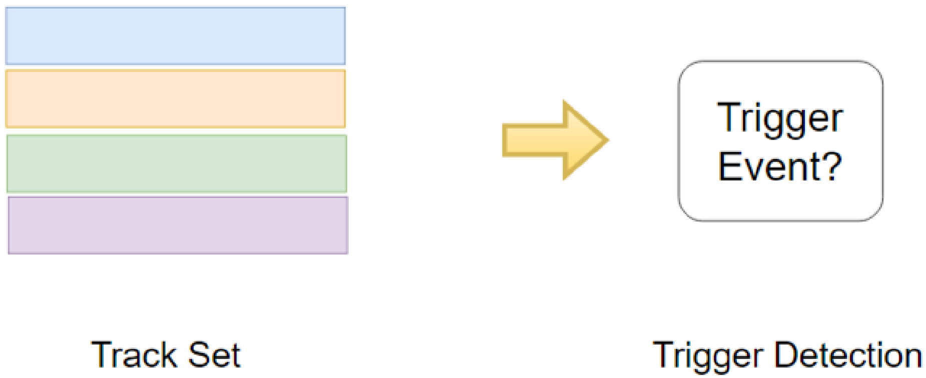
Software of Year 3 is much more hardware aware than that of Year 2!

- 1 iteration on hits generation instead of 4 iterations
- Hidden layer of MLP is reduced from 1024 to 8
- Much more constraints on geometry to select edge candidate

Tracks $\mapsto$ Trigger Label

From Tracks to Trigger

- After creating the tracks, we have a set of tracks
- We want to know whether the event that created these tracks was a trigger event
- A *trigger event* is an event in which we had a beauty decay event, (10^{-5})



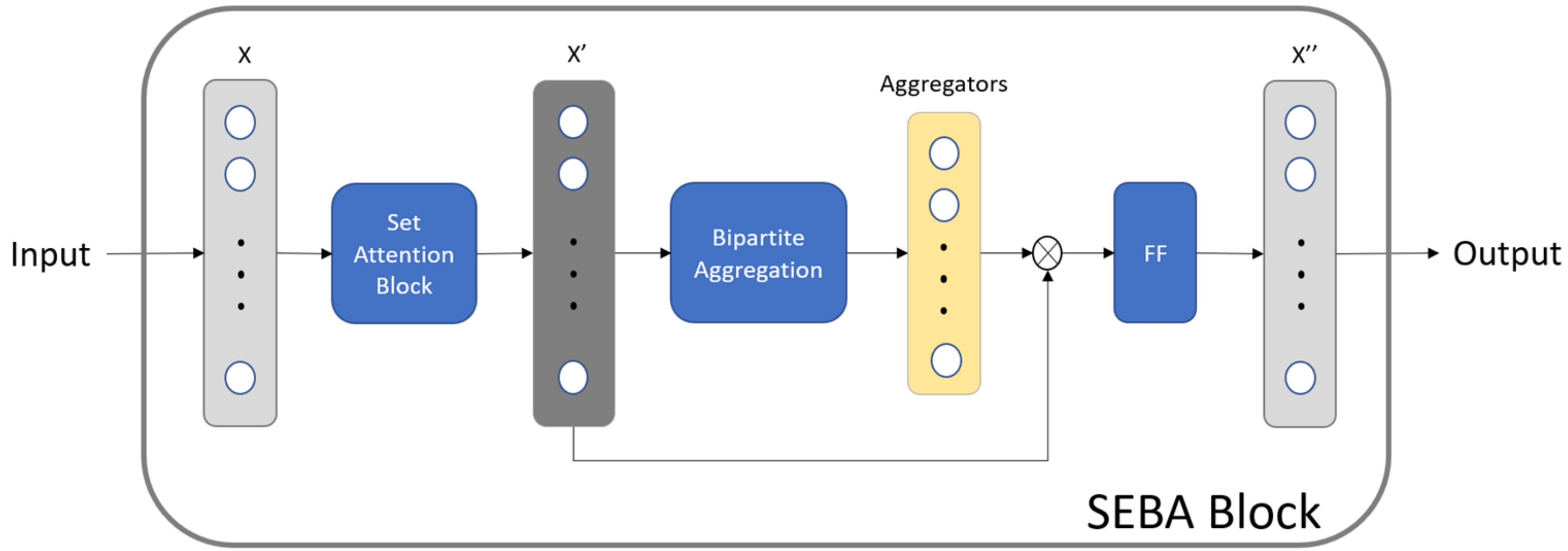
What needs to be modeled?

- We consider the beauty decay event. (produce a $b\bar{b}$) (beauty-antibeauty) quark pair).
- Considering the problem from a high level perspective, we need to consider:
  - Track-to-track Interactions: Do these pair of tracks form a beauty decay product pair?
  - Track-to-global Interactions: Where is the origin of this track?
  - Global-to-Track Interactions: Incorporate information about the origin of this track into the track embeddings

Architecture

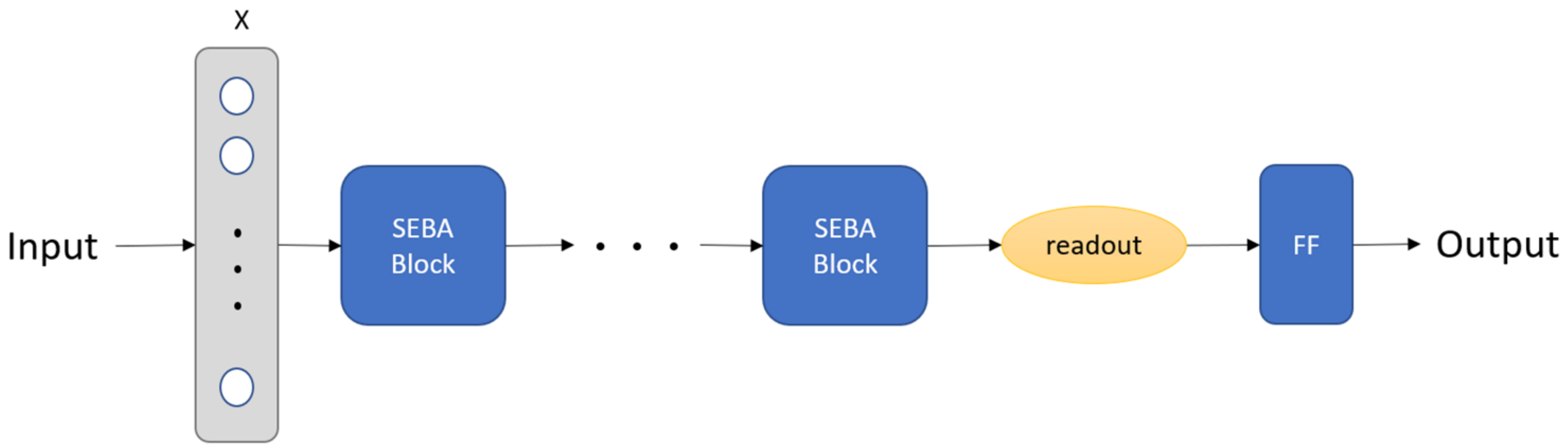
- Previous considerations motivate the following block.
  - Set Encoder: Track-to-Track interactions
  - Bipartite Aggregation: Track-to-Global and Global-to-Track interactions

SEBA (Set Encoder with Bipartite Graph Affinity)



Architecture

- Stack multiple SEBA Blocks
- Use Bipartite Aggregation with single aggregator to generate event embedding
- MLP on event embedding to predict Trigger Event



Physics Knowledge Added

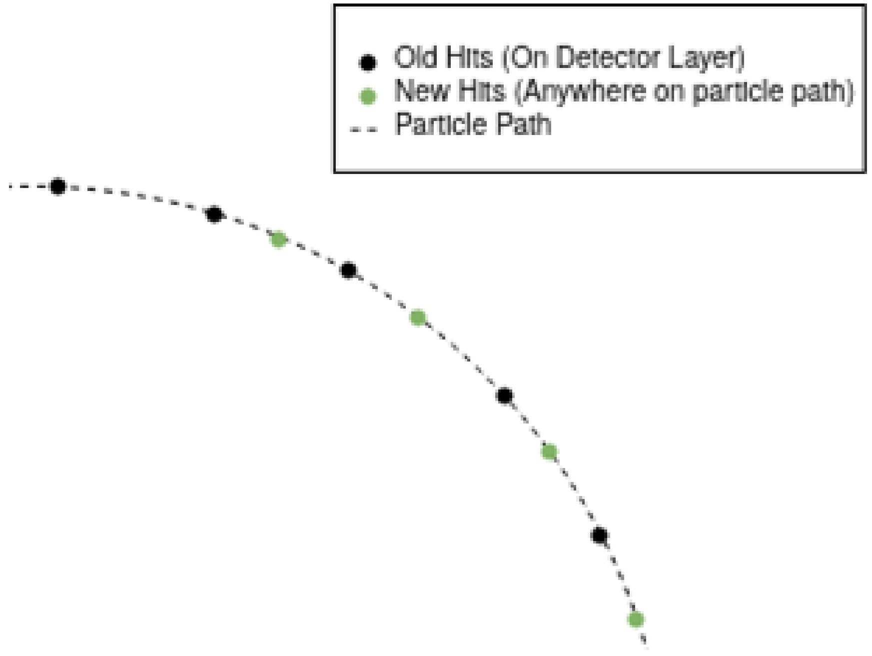
- Track given to trigger classifier has the following features:
  - (x, y, z) location of hit on each layer
  - Length segment between each layer
  - Angle formed by segments
  - Estimated radius of circle fit to hits
  - Estimated center of circle fit to hits
  - Estimated transverse momentum of track
- **Estimated radius and center** provided ~10% increase in accuracy in the 2022 triggering problem (charm to pion and kaon) ($D^0 \rightarrow \pi K$ decay)*.

*In 2024 we were provided with the data for a new triggering target (beauty decay). Re-evaluation of performance improvement on new data was not done.

Multi-Task Learning to Improve model performance

- Several modifications to standard training process in order to improve the performance and robustness of our trigger algorithm
  - Data augmentation: We perturb hits off the detector layers while keeping it on the particle path
  - Track embeddings used predict whether two tracks come from the same parent

$$\mathscr{L} = L_{CE}(\mathrm{trigger}_{pred}, \mathrm{trigger}_{true}) + L_{CE}(A_{pred}, A_{true})$$



Pile-Up

- MVTX and INTT readout speed differ (INTT is much faster and have high-res timestamps)
- Event data “piles-up” in the MVTX detectors
- Thus, when we read out the data, the MVTX data is the activated pixels from the last *10* events instead of just from the last event
- No pile-up in INTT data
- We are adapting our algorithms to handle event pile-up robustly

Pile-Up Strategies

- Pile-up introduces major latency problems by increasing the data by ~10X that our ML solutions need to process
- Prediction is also harder as 90% of the data is noise irrelevant to the current problem. Model needs to distinguish between signal and noise.
- Need strategies to reduce the amount of data ML algorithms need to process
- Three-pronged approach:
  - Use Hit-Set Pileline: This is our fastest pipeline, with the least number of processing steps.
  - Drop Inner 2 MVTX layers from data: The first 2 MVTX layers improve the triggering performance only marginally while comprising the majority of the data.
  - INTT-based filtering: The hits in the third (outermost) MVTX layer are filtered using geometric constraints with the INTT hits.
  - **75% reduction in data quantity with only marginal reduction in trigger performance.**
- We refer to this pipeline as fast hits-set.

Trigger Prediction Performance (Beauty Decay)

Pipeline	Pileup	Accuracy
Track-Set	No	95.4%*
Hits-Graph	No	91.5%
Hits-Set	No	90.6%
Hits-Set	Yes	88.5%
Fast Hits-Set	Yes	86.5%

*Trained on ground-truth tracks, not predicted tracks. Performance on predicted tracks should be similar due to high performance of tracking stage.

Trigger Prediction Performance (Old Trigger Definition on Charm)

Data	Year	Metric	Result
Predicted Tracks	**2023**	**Accuracy**	**85.6%**
GT Tracks	**2023**	**Accuracy**	**90.22%**
GT Tracks	2023	Precision	86.35%
GT Tracks	2023	Recall	95.41%
Predicted Tracks	**2022**	**Accuracy**	**84.01%**
GT Tracks	**2022**	**Accuracy**	**87.5%**

Set Transformer: arXiv:1810.00825
GarNet: arXiv:1902.07987
PN: arXiv:1902.08570
SAGPool: arXiv:1904.08082
GCN: arXiv:1609.02907

Model accuracy studies

1. Triggering on $D^0 \rightarrow \pi K$ (0.1% events)

Note: all Accuracies are calculated on 50% signal/background samples

Model	With p_T prediction with LS-radius			Without p_T prediction without radius		
	#Parameters	Accuracy	AUC	#Parameters	Accuracy	AUC
Set Transformer	299,266	86.40%	91.92%	298,882	72.04%	78.92%
GarNet	284,210	86.22%	91.81%	284,066	72.59%	79.61%
PN+SAGPool	780,934	86.25%	92.91%	780,678	69.22%	77.18%
BGN-ST	363,426	**87.56%**	**93.22%**	363,170	**74.13%**	**81.81%**

Estimating p_T from vertex detectors resulted in 14% accuracy increase!

0.1% signal/background ratio		
BG Rejection	Efficiency	Purity
90%	76%	0.75%
99%	23.2%	2.3%

- 3 MHz collision rate
- 10% HF efficiency (ext. readout)
- 1 kHz available for additional triggers
- 3000 MB rejection needed

23x rate increase comparing to random selection!

- Trigger detection on tracks vs hits
  - Accuracy: 90.22% (BGN-ST, track construction, model v2) vs 85% (GCN, hit-based)

3. Triggering on Beauty decays, (0.05% events)

- No pileup
  - Accuracy: 97.38% (BGN-ST, track construction, model v2)
  - Clusters -> Edge Candidate Generation -> Trigger prediction: Accuracy 91.53% (Graph Attention Network, hit-based)
  - Clusters -> Trigger prediction: Accuracy 90.57% (GarNet, hit-based)
- Pileup (~350 hits + 65 noise)
  - Clusters -> Trigger Prediction: Accuracy 88.52% (GarNet, hit-based)

Large accuracy increase reconstructing tracks!

Attention provides slight improvement for clusters

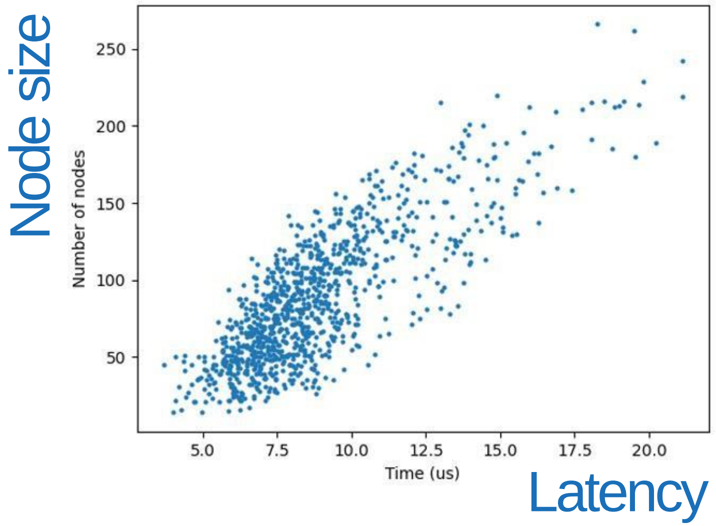
Pileup has a small effect!

Generation of the FPGA IP core – two parallel efforts

1. Team lead by the Georgia Institute of Technology (GIT)
   - Direct translation using FlowGNN (ArXiv:2204.13103)
   - Goal: 100-200 nodes, 200-500 edges
   - Implementation of edge classification
     - 92 nodes, 142 edges
     - Measured Start-to-end latency
       - 150 us @ 130 MHz, edge classification v1
       - 8.82 us @ 285 MHz, edge classification v2
   - Implementation of hit-based model
     - Measured Start-to-end latency
       - 9.2 us @ 180 MHz

Utilization (Alveo U280)	
LUT	194K (14.9%)
FF	214K (8.2%)
BRAM	406 (20.2%)
DSP	488 (5.4%)

1k graphs



- Detailed latency breakdown and parallelism exploration ongoing
  - The effects of FlowGNN parameters

Close discussion between model developers and FPGA engineers

Experiment Integration

- The hits-based algorithm was validated end-to-end (data readout, clustering, trigger prediction) in FPGA.
- sPhenix experiment experiences delays in streaming INTT hits (INTT readout needs to be commissioned in the current year.
- We can not use INTT to down-select pile-up event data.
- We have to train new models for MVTX only events.
- Hits based model works with FPGA.
- The model will be deployed into sPhenix Felix FPGA in August 2024.

Conclusion, Accomplishments and Milestone

- ML models have shown steady increases in performance on the triggering problem
- Incorporating physics knowledge has contributed to large performance improvement in trigger prediction
- New strategies developed to effectively handle event pile-up while maintaing latency and accuracy targets.
- Challenges remain in adapting the ML algorithm to the real-world latency and data availability constraints

Future Work

- Futher Work on simplifying algorithms and reducing data quantity to meet latency challenges
  - Improve Fast Hit-Set pipeline to bring performance closer to Hit-Set pipeline while further reducing the data quantity
- Ensure trigger algorithm works in explainable and robust way
  - Initial study has shown model prefers to drop non-trigger tracks without affecting event label and prefers to perturb hits as to not affect the track radius

Test model with real sPhenix experimental data!!!
(end of 2024 expected)

Acknowledgement

- Thank DOE Office of Science, Dr. Michelle Shinn for funding this project, and every contributor for working on this project!