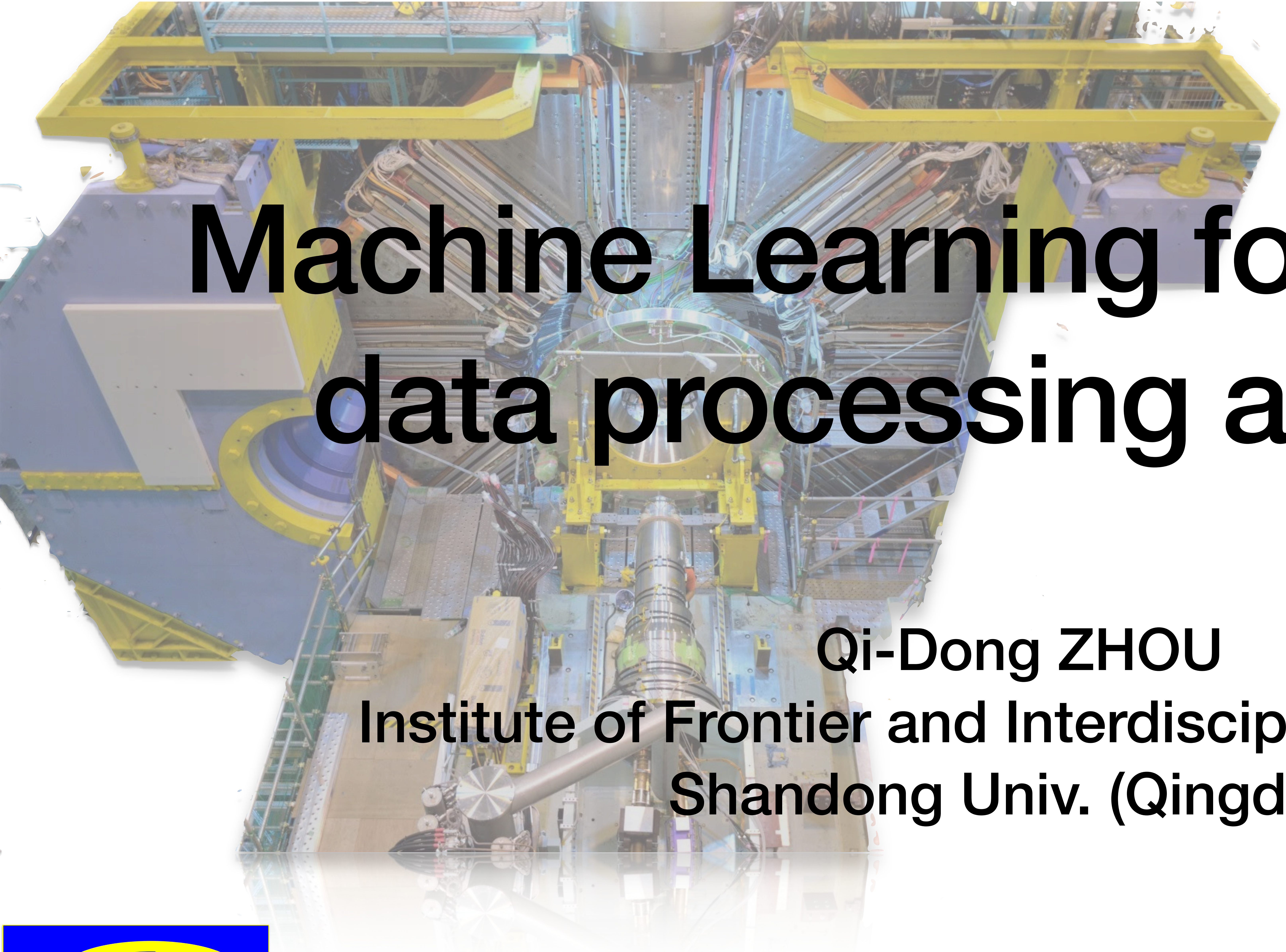


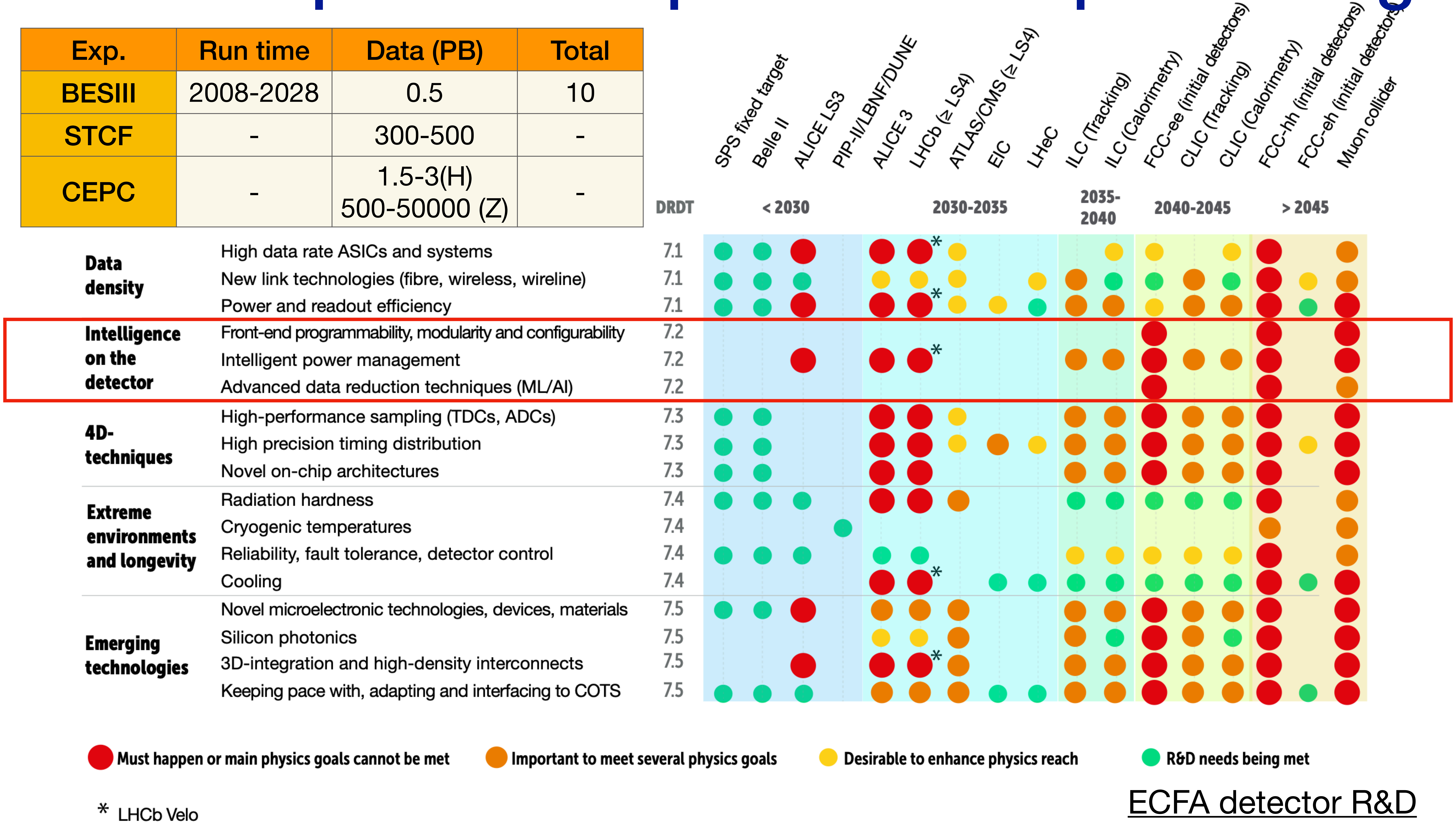
Machine Learning for real-time data processing at Belle II

Qi-Dong ZHOU
Institute of Frontier and Interdisciplinary Science,
Shandong Univ. (Qingdao)

20-23 Aug. 2025, Qingdao
Quantum computing and machine learning workshop 2025

Roadmap of techniques for data processing

Exp.	Run time	Data (PB)	Total
BESIII	2008-2028	0.5	10
STCF	-	300-500	-
CEPC	-	1.5-3(H) 500-50000 (Z)	-



ECFA detector R&D

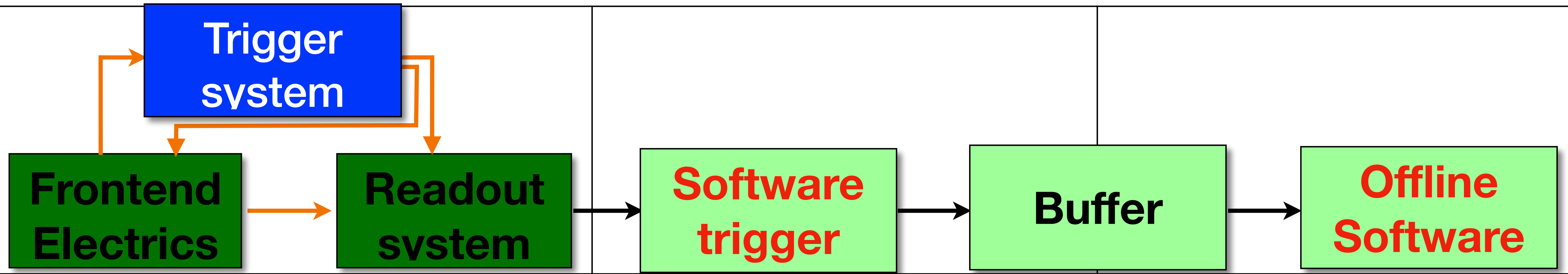
Data processing system (Belle II vs. LHCb)

- Belle II: L1 trigger + HLT
  - Trigger efficiency:
    - Had. B physics～100% τ physics 70~95%

- LHCb: "triggerless" readout & DAQ
  - CPU+GPU based software trigger
  - ~350 GPU RTX A5000
  - Part of online data processing with FPGA for upgrade

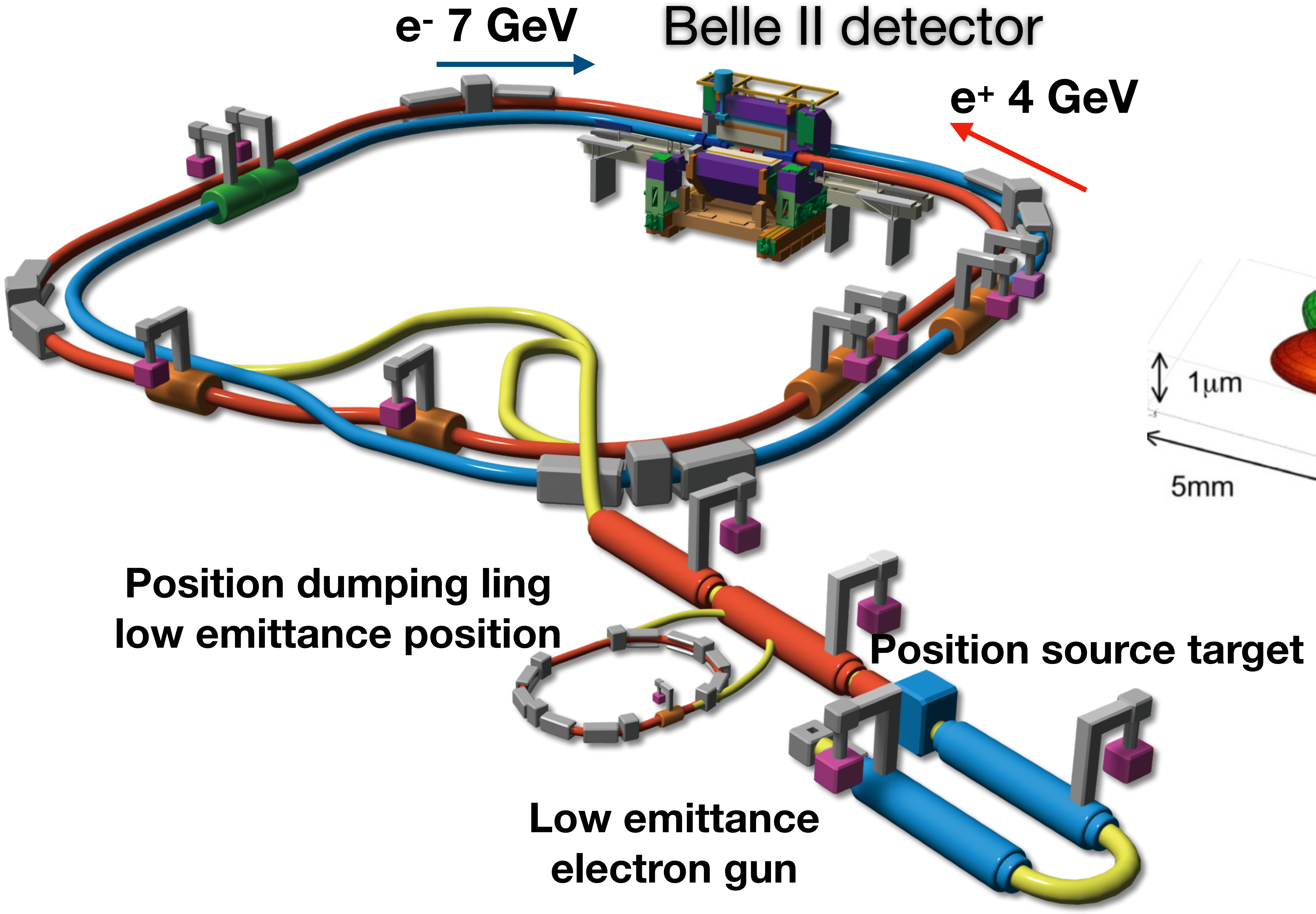
Heterogenous computing system

- System integrated with different devices
  - System level, chip level
- FPGA widely used in frontend and trigger electrics
- FPGA, AI engine, DPU, System On Chip, Network On Chip for computing acceleration

Devices	Specifics	
		
FPGA	• Level 1 trigger: low latency w/ **Ous** • Simple ML algorithm available	
FPGA/ AI engine/ DPU	• Software trigger w/ **Oms** • CPU/FPGA/GPU system on chip • CNN easily run on DPU	 
GPU	• Software trigger & offline software • User friendly and widely supported	 

Luminosity frontier: SuperKEKB

- Asymmetric e^+e^- collider
  - $e^+e^- \to \Upsilon(4S) \to B\bar{B}$
  - very clean and well-known initial state

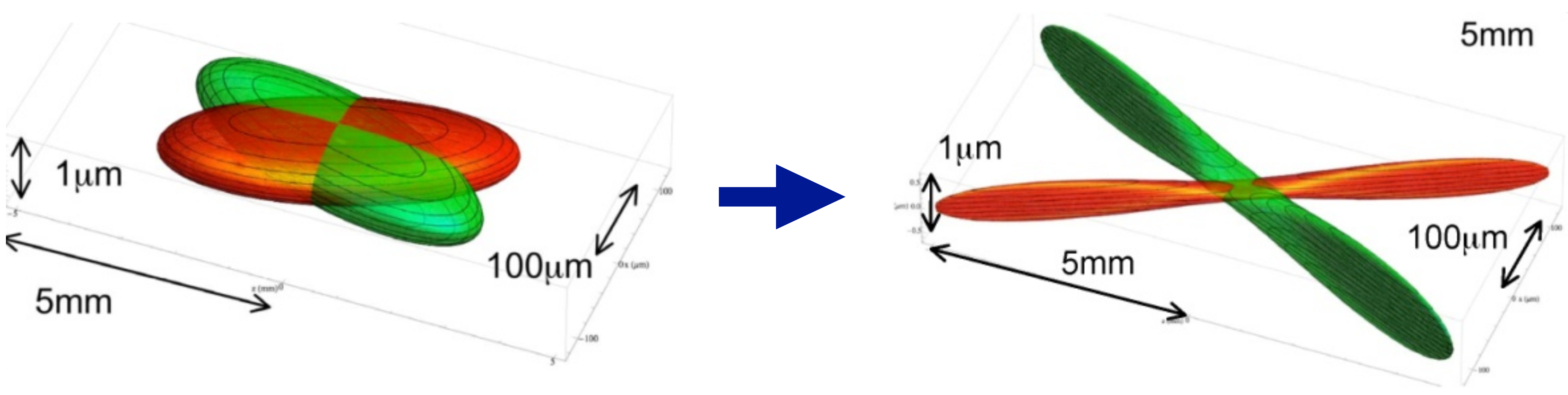


Beam current: KEKB x ~1.5

$$L = \frac{\gamma_\pm}{2er_e}\left(1+\frac{\sigma_y^*}{\sigma_x^*}\right)\frac{I_\pm \xi_{\pm y}}{\beta_y^*}\left(\frac{R_L}{R_y}\right)$$

Beam squeeze: KEKB / ~20

Nano beam scheme



Target: L = 60 x 10^{34} cm^{-2} s^{-1}

Achieved : 5.1 x 10^{34} cm^{-2} s^{-1} (Record)

- Data:
  - **575** fb^{-1} (Belle II) <-> 980 fb^{-1} (Belle)

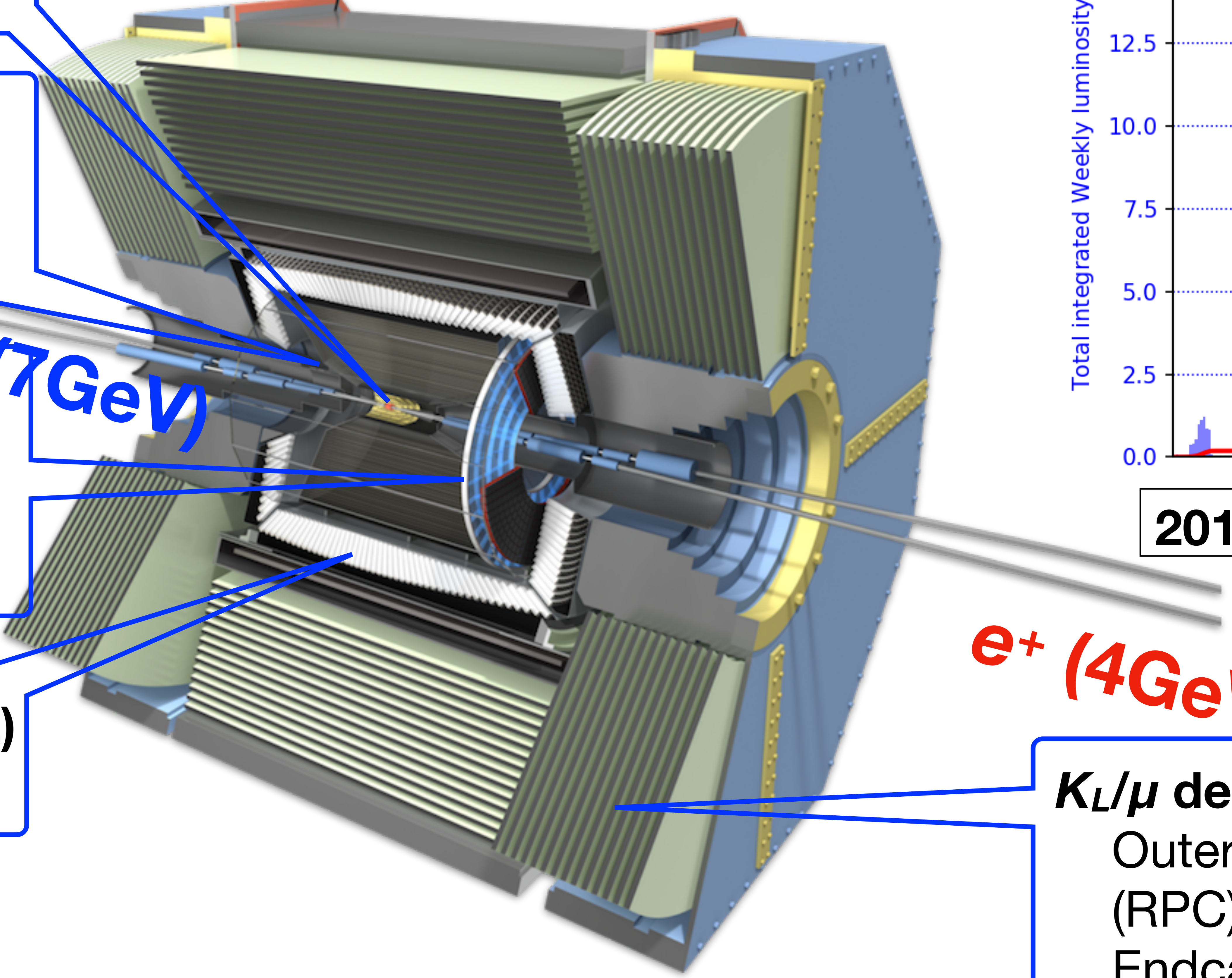
Belle II detector and dataset

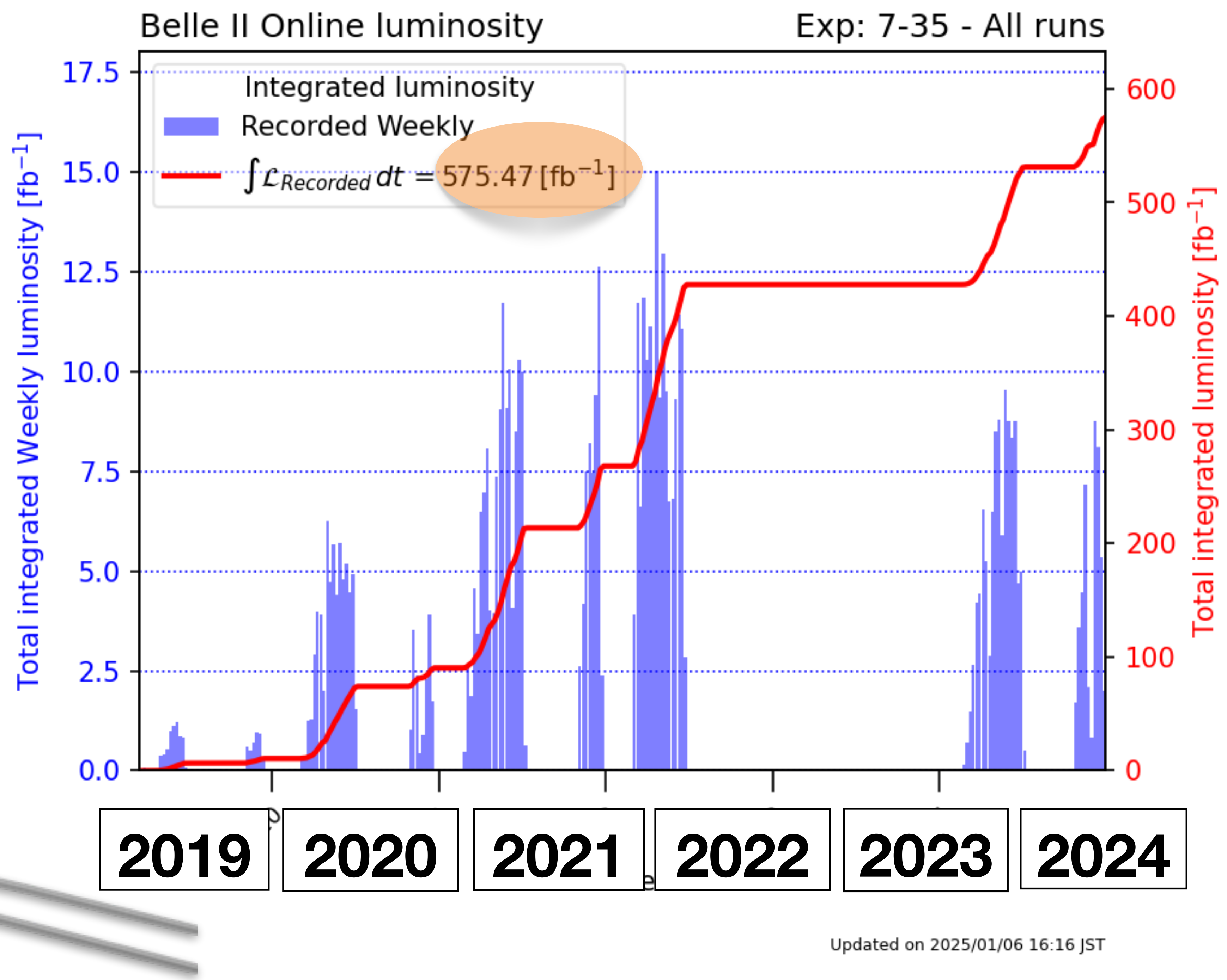
Vertex detector (VXD)
Inner 2 layers: pixel detector (PXD)
Outer 4 layers: strip sensor (SVD)

Central Drift Chamber (CDC)
He (50%), C_2H_6 (50%), small cells, long lever arm

Particle Identification
Barrel: Time-Of-Propagation counters (TOP)
Forward: Aerogel RICH (ARICH)

ElectroMagnetic Calorimeter (ECL)
CsI(Tl) + waveform sampling



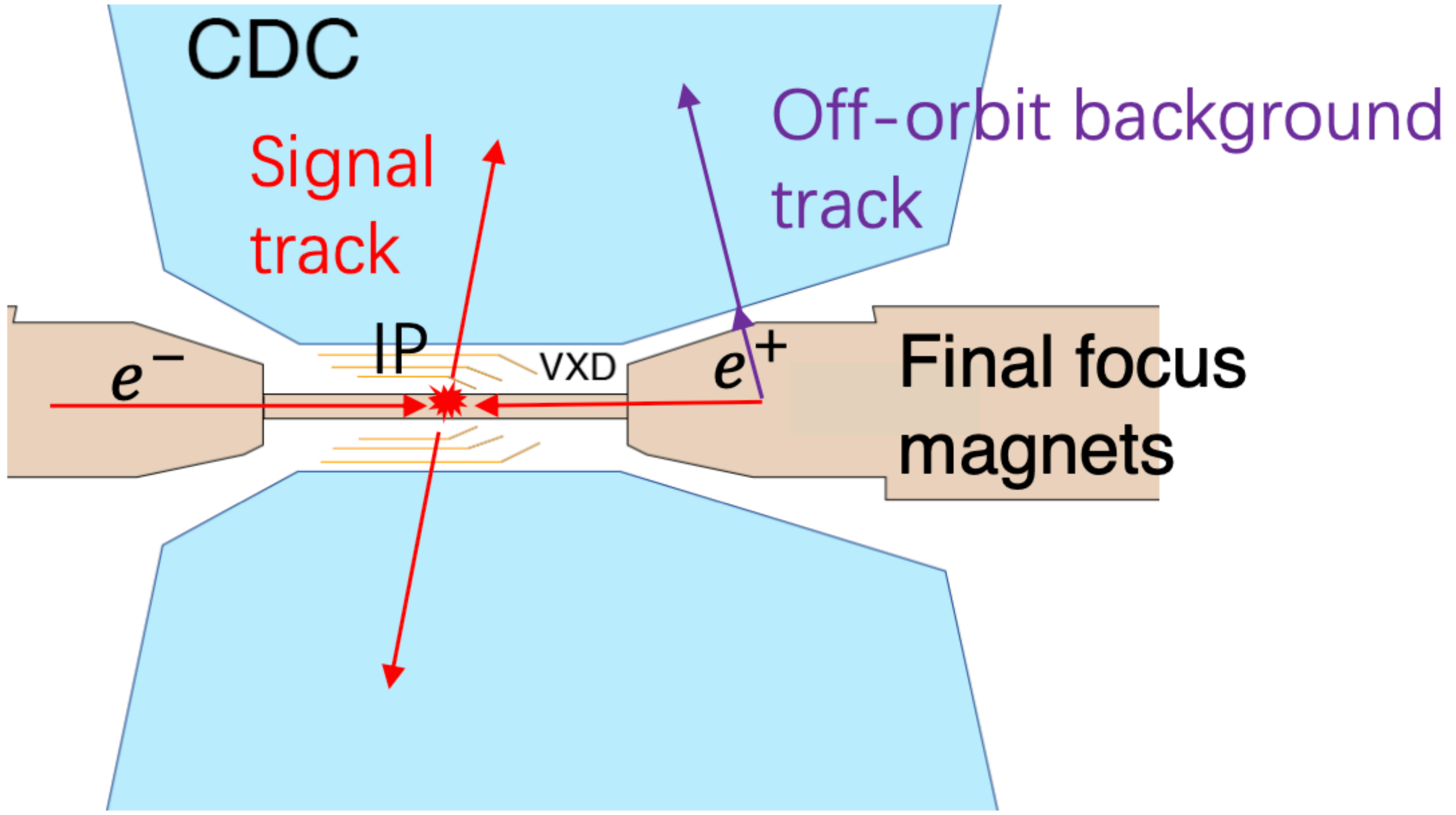


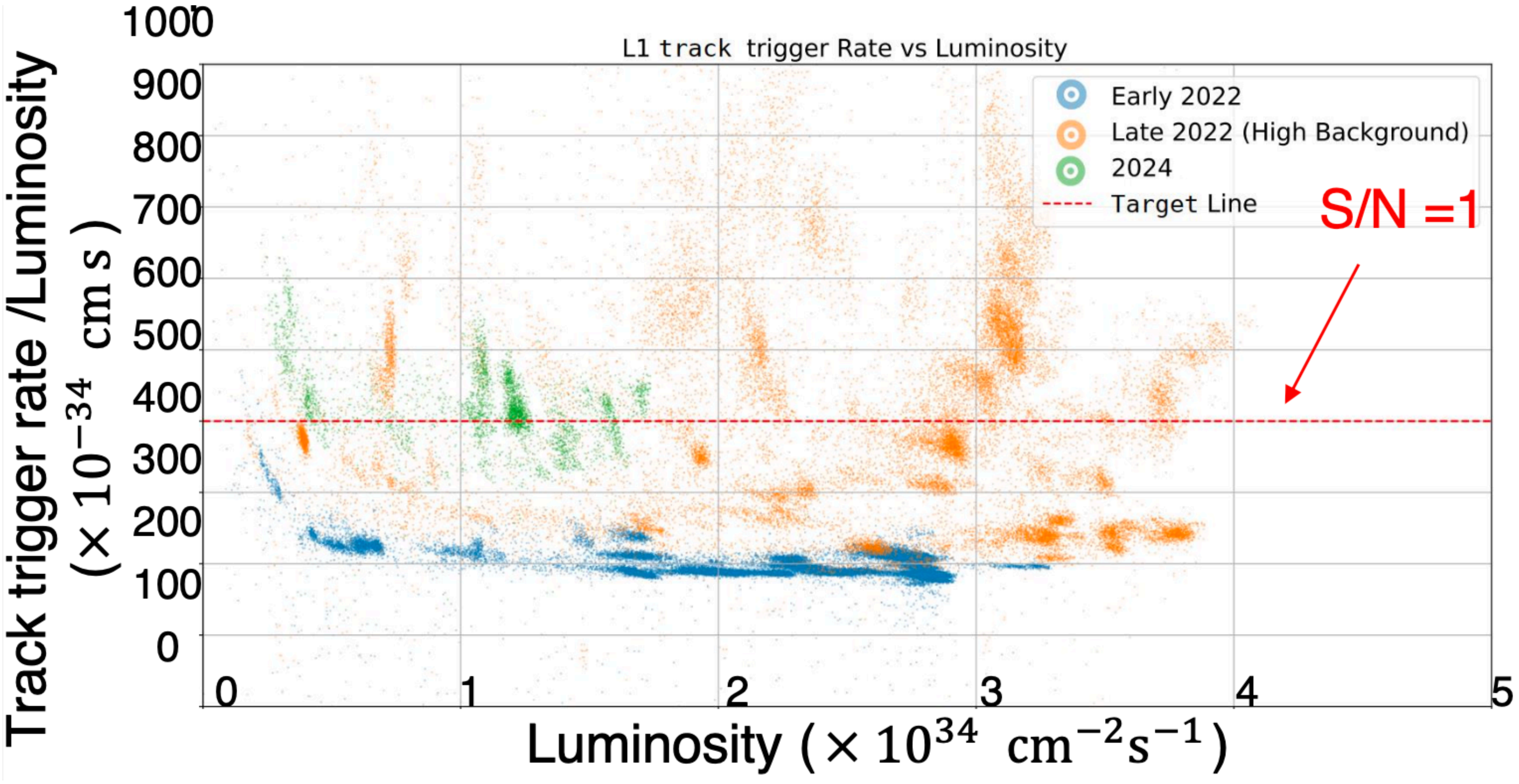
***K_L/μ* detector (KLM)**
Outer barrel: Resistive Plate Counter (RPC)
Endcap/inner barrel: Scintillator

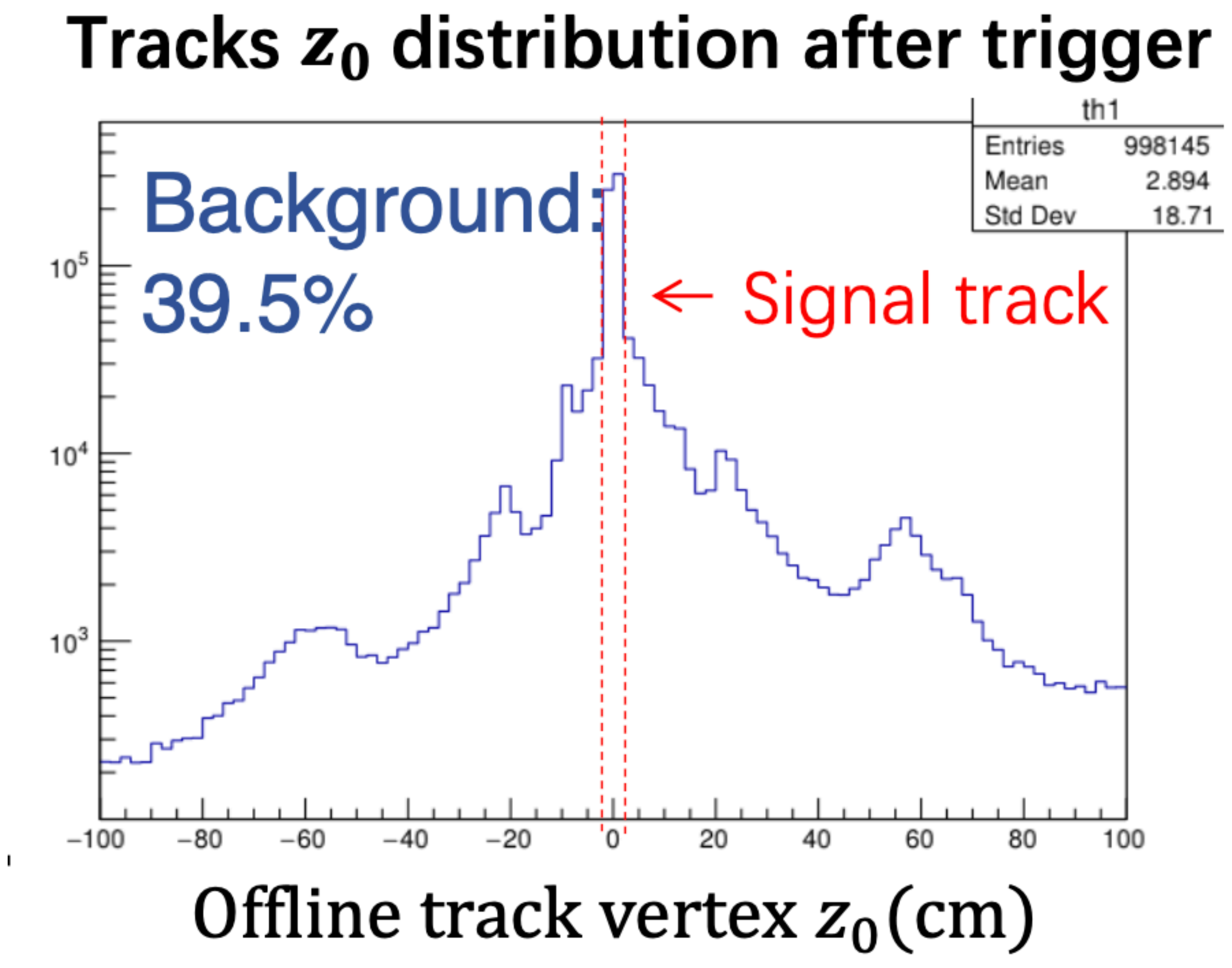
- Features:
  - Near-hermetic detector
  - Vertexing and tracking: σ vertex ~ 15μm, CDC spatial res. 100μm $\sigma(P_T)/P_T$ ~ 0.4%
  - Good at measuring neutrals, π^0, γ, K_L… $\sigma(E)/E$ ~ 2-4%

Motivation of Neural Network for L1 Track trigger

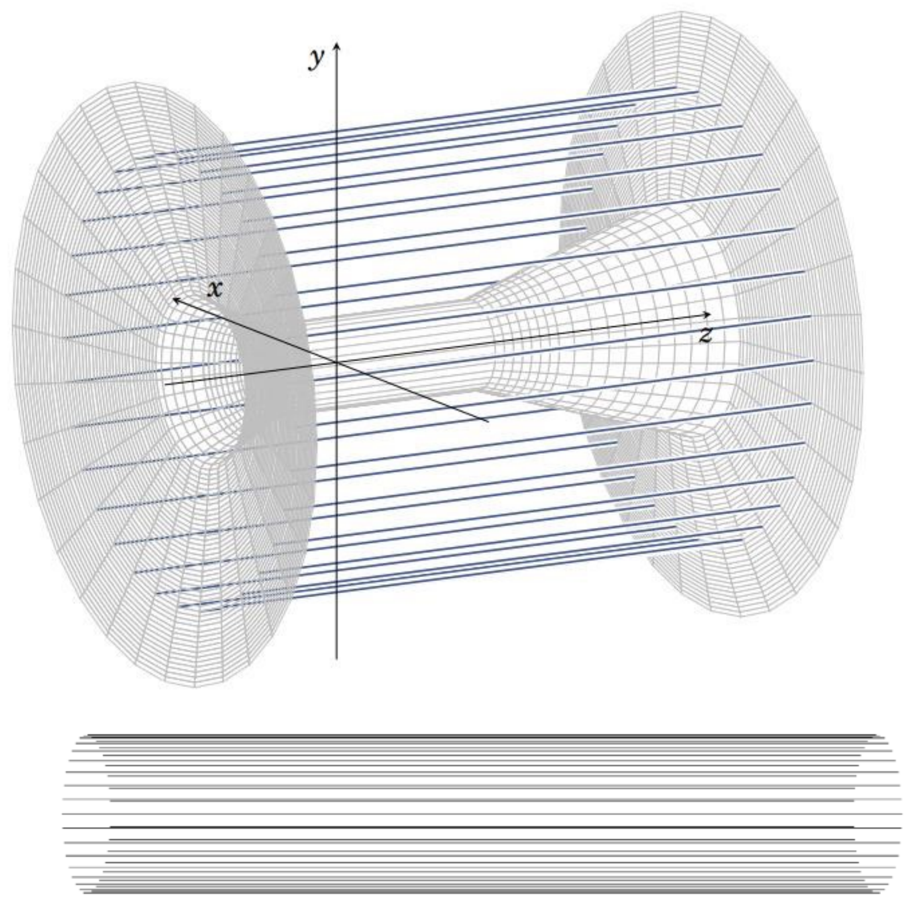
- DAQ system is designed to handle 30 kHz
  - Physical trigger ~15 kHz, require S/N = 1
  - 200350 kHz (total rate) -> ~15 kHz (physics rate)
- L1 trigger rate depends significant on background condition
- Advanced CDC algorithm to further suppress background
- A fixed latency of about **4.4 usec**



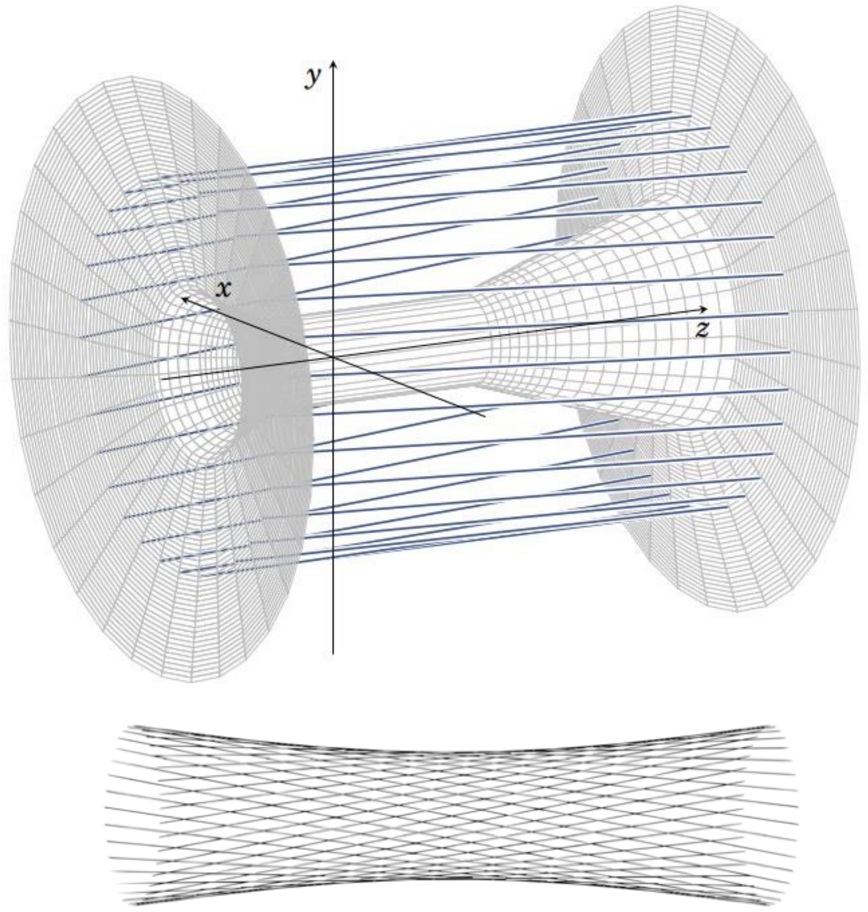




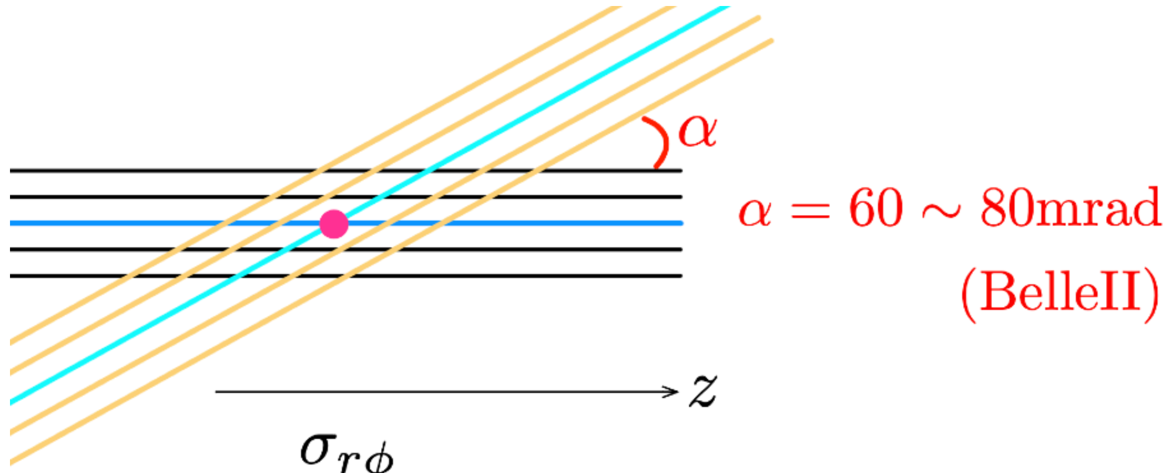
Basics of L1 CDC trigger



Axial wire

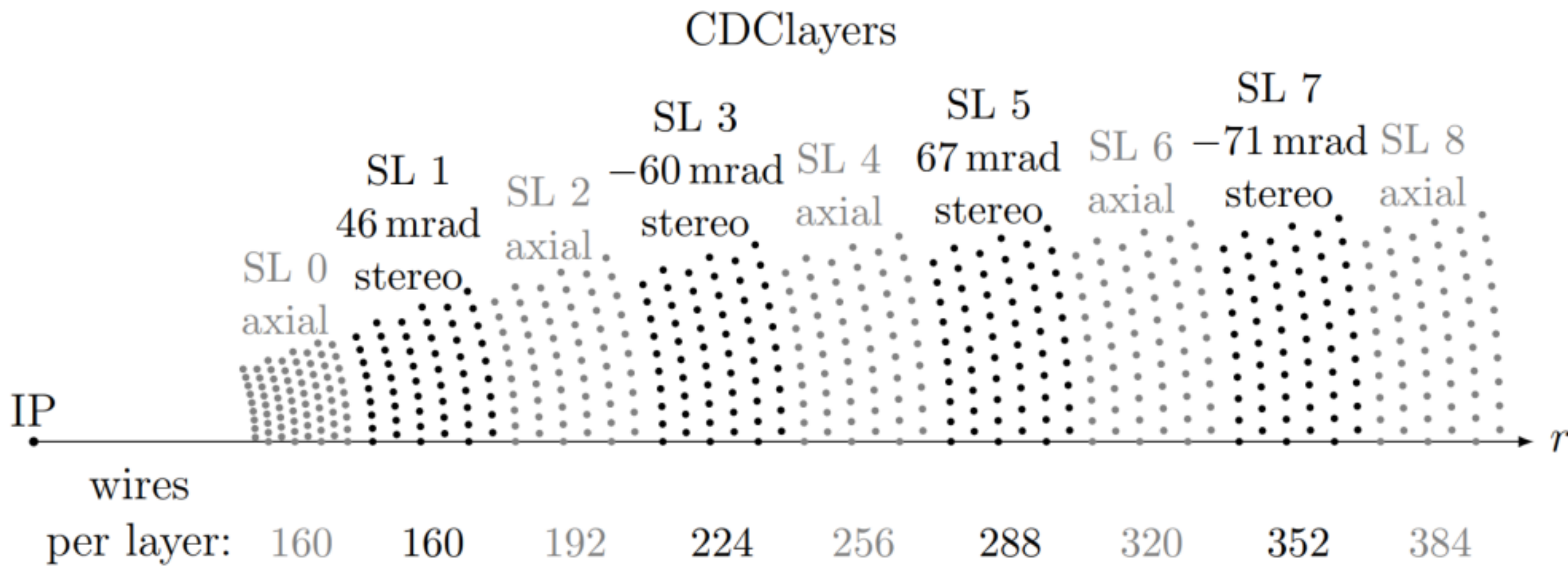


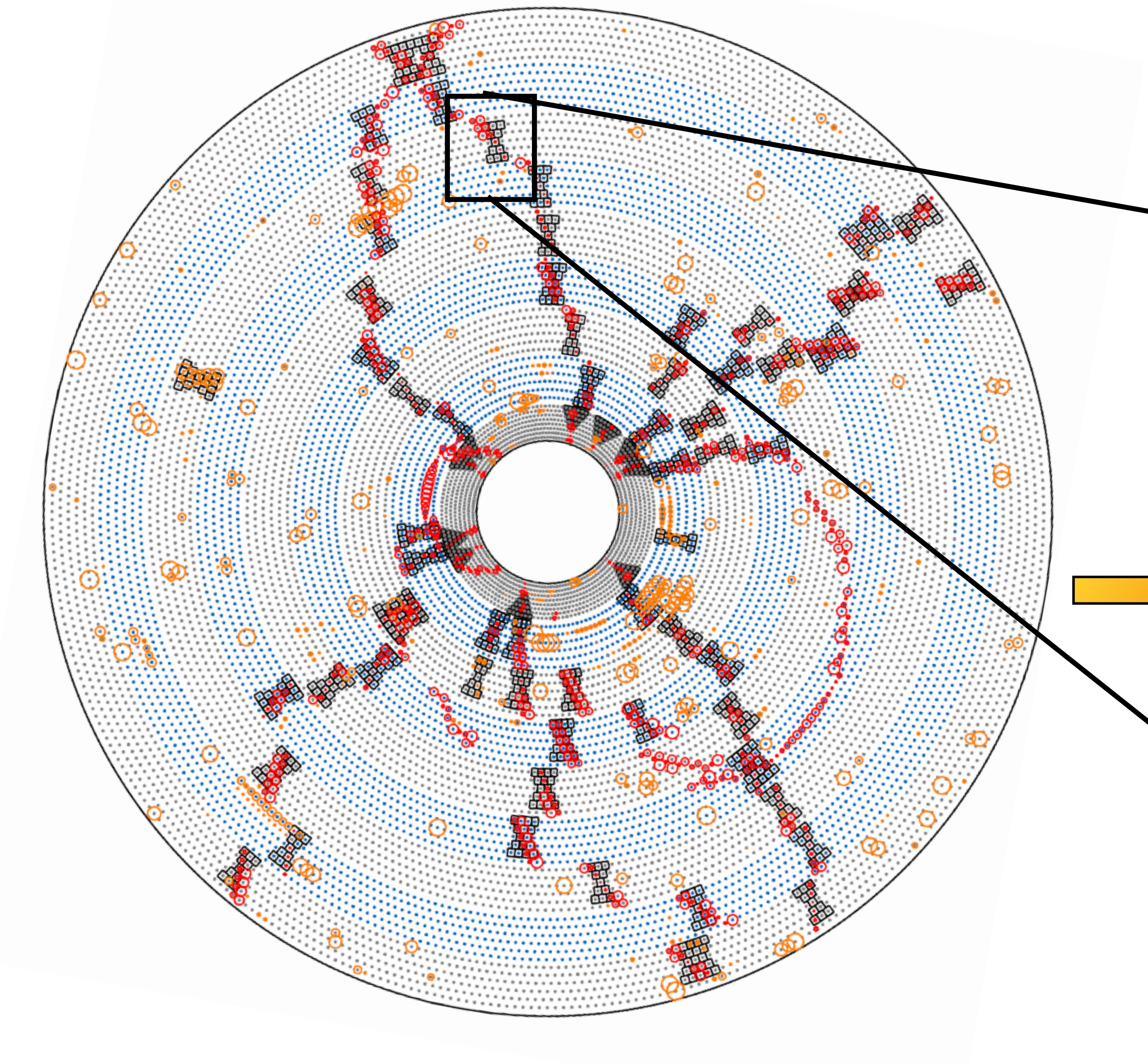
Stereo wire



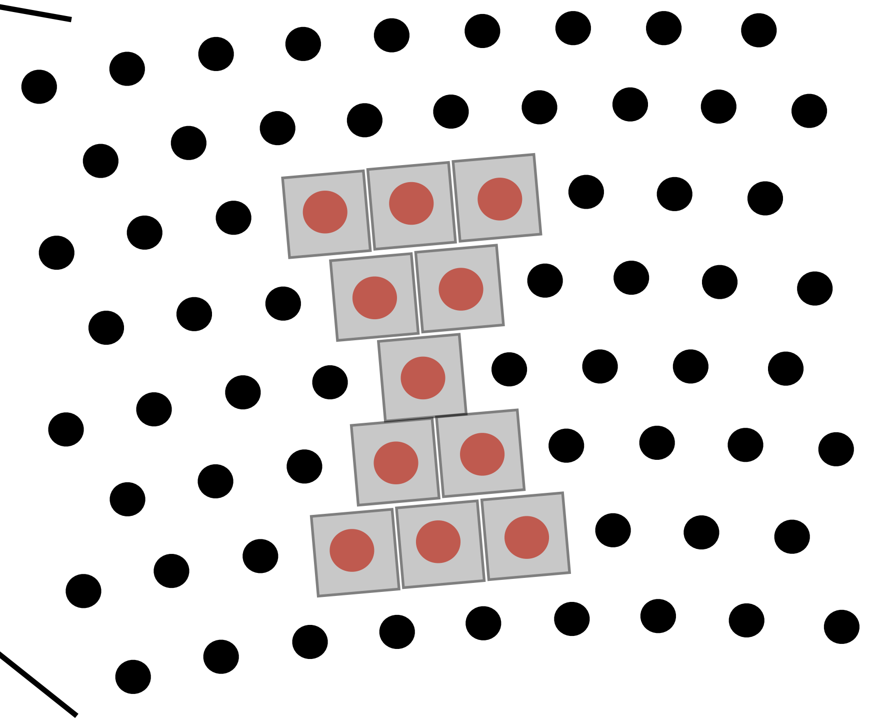
$$\sigma_z = \frac{\sigma_{r\phi}}{\sin\alpha}$$

$$\sigma_z = 1.9 \sim 2.5\text{mm}(\sigma_{r\phi} = 0.15\text{mm})$$

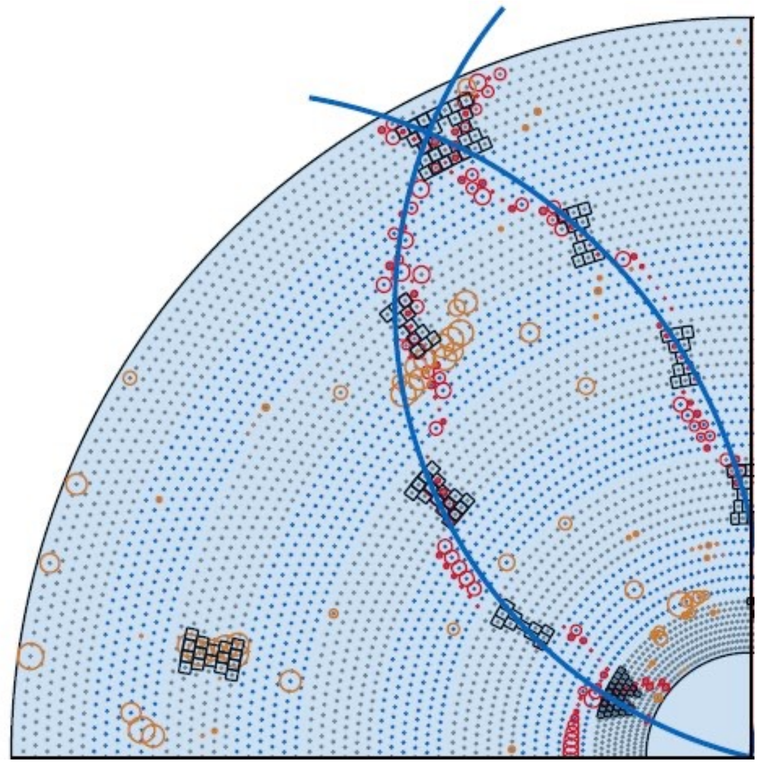




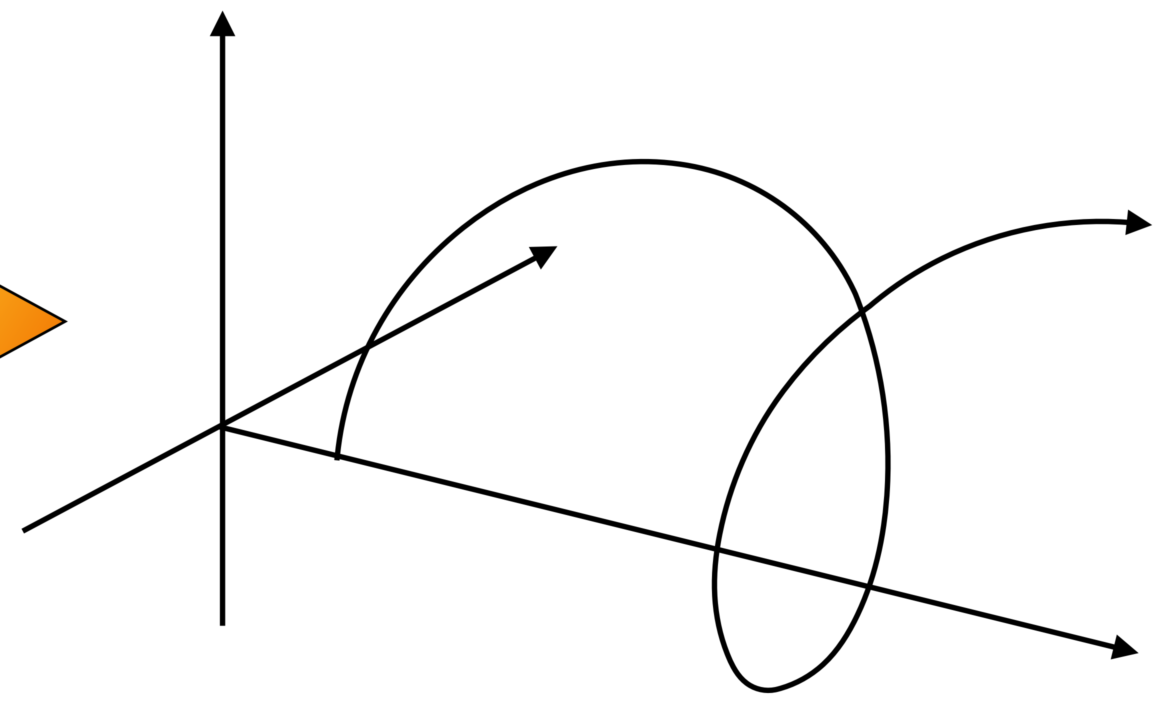
CDC 2D hit map



Track Segment (TS) Finder in every super layer

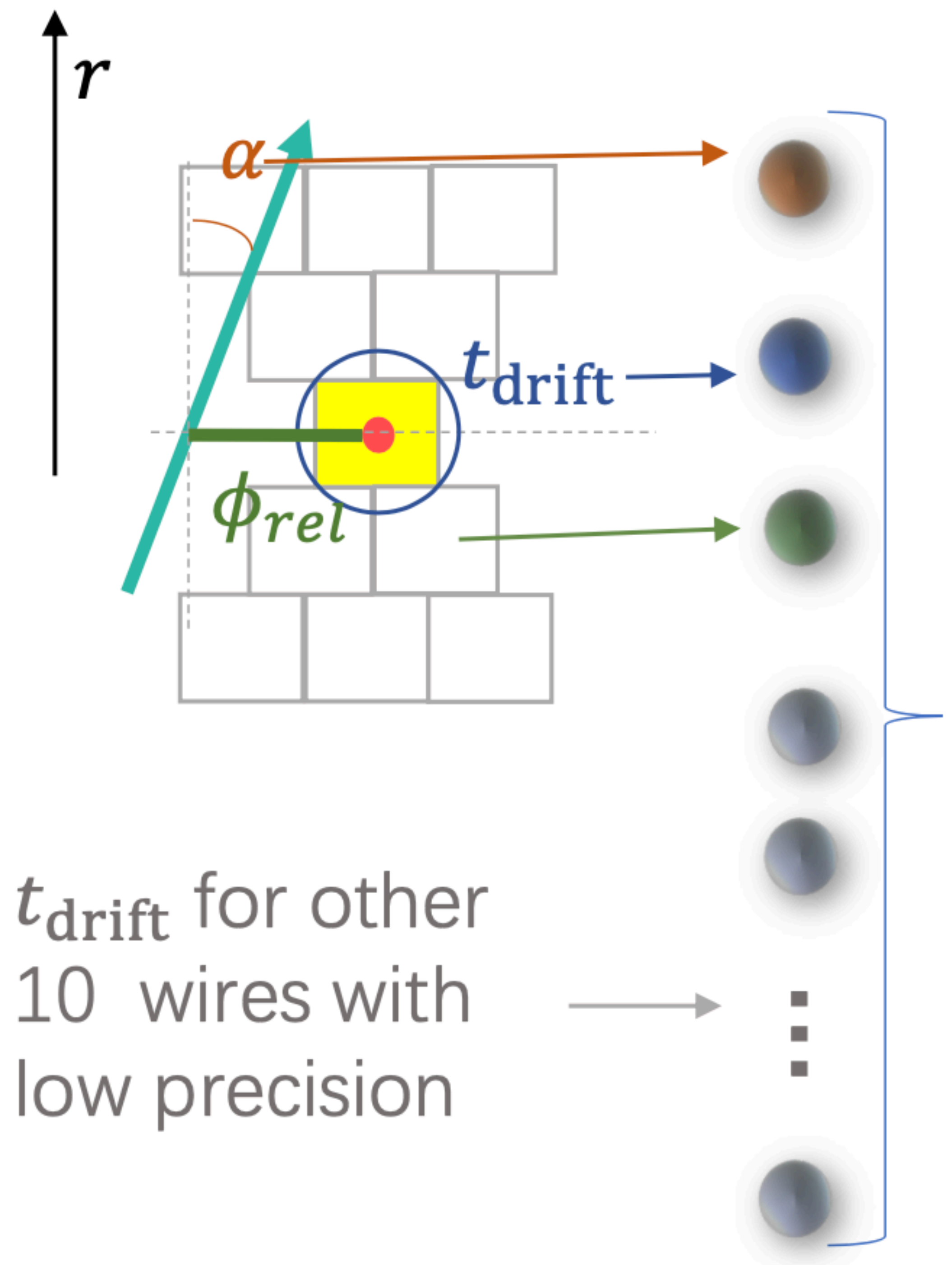


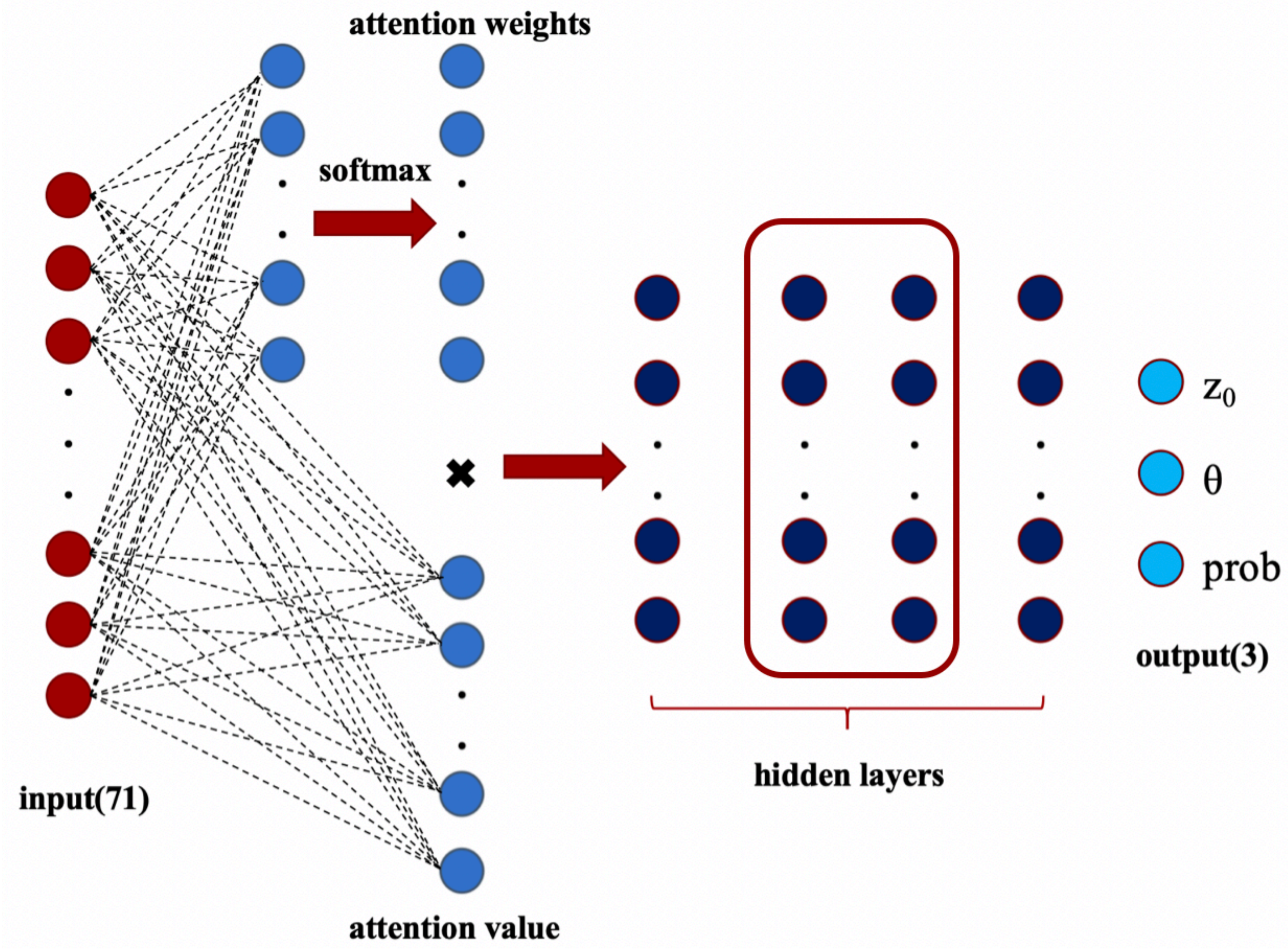
Build 2D track with axial hits using Hough transformation

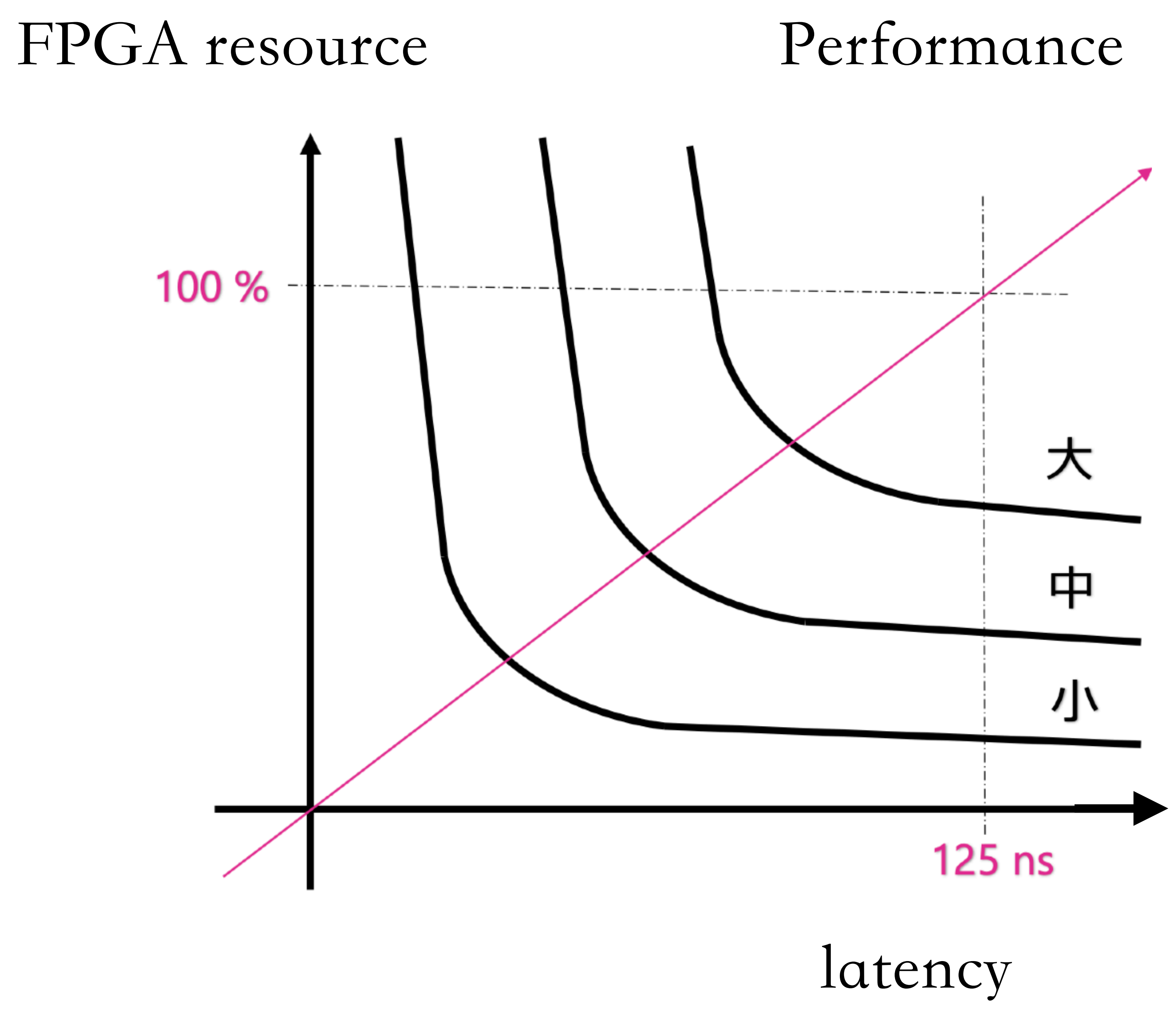


3D Track trigger base on 2D track and Stereo TS

Deep Neural Network for Z trigger







- Inputs: **Drift time t_{drift}**, **wires relative location ϕ_{rel}**, **Crossing angle α** for priority wires + **Drift time for all other wires**
- Introduce the **self-attention architecture** to "focus" on certain inputs
- Output track vertex z_0, track θ and **signal/ background classifier output** (Q)

Parameter	#Attention value	#hidden nodes	#hidden layer	activate	precision	Total multiplier
Values	**27**	**27**	**2**	**Leaky Relu**	**Float 16**	**4,185**

Development flow of DNN on FPGA

Software → Hardware → FPGA

Keras + TensorFlow
PyTorch

- Machine Learning model
- Parameter

hls4ml

- C/C++ transition

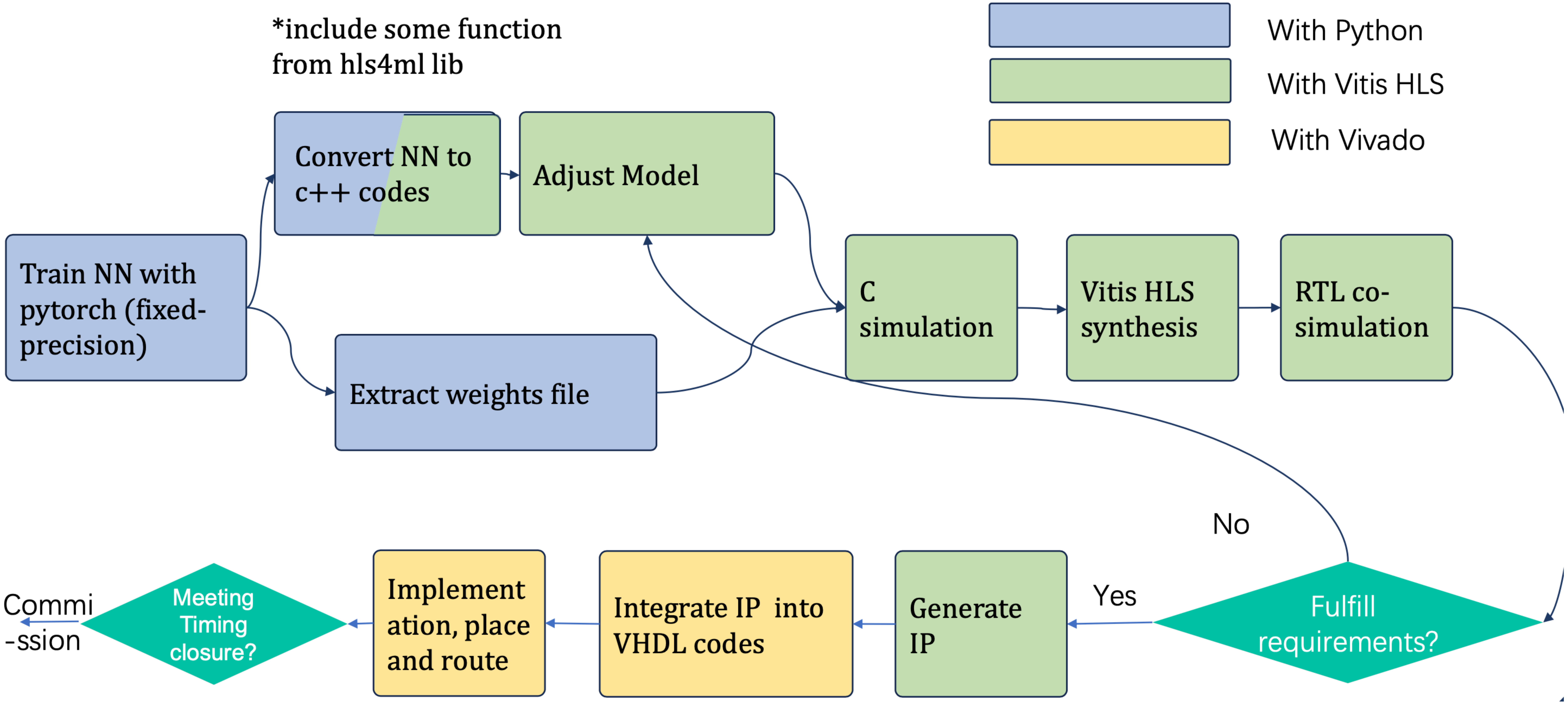
Vivado™ HLS

- Translate into Verilog/VHDL FPGA language

VIVADO

- Start fitter

- Evaluation



Belle II UT4

Xilinx UltraScale
XCVU080, XCVU160
25 Gbps with 64B/66B

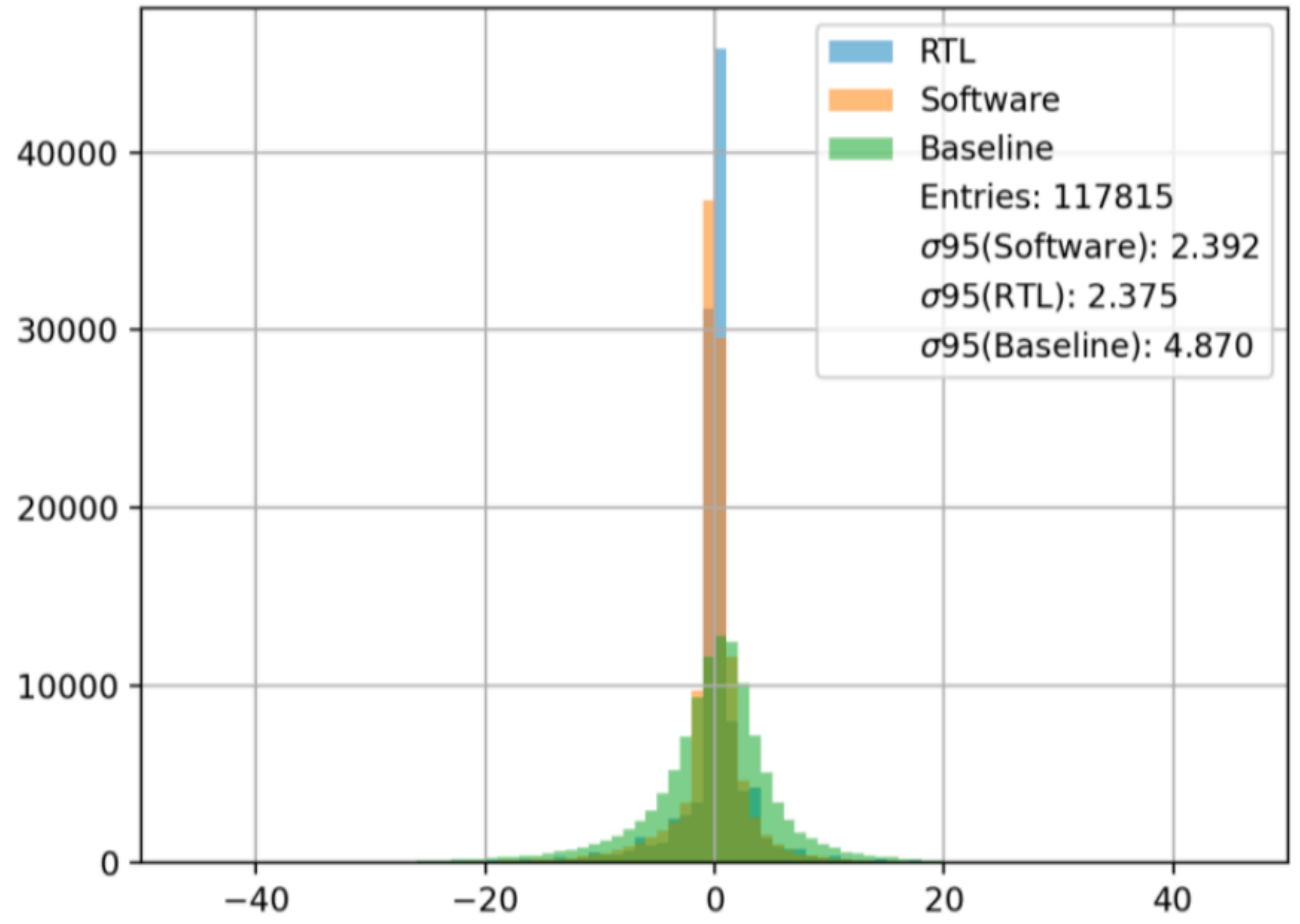
Quantization-Aware Training (QAT) for FPGA implementation

- Quantization is essential technique to speed up inference, reduce resource usage
  - Embedded system, edge device
  - Fake quantization during training
  - For example, convert 32-bit floating to 8-bit integer
    - Reduction in the model size, memory bandwidth 4x
  - Optimization item: performance vs. latency vs. resource usage

	No QAT	QAT
LUT	~46%	~27%
DSP	~64%	~56%
Latency	551 ns (70 clock)	488 ns (62 clock)

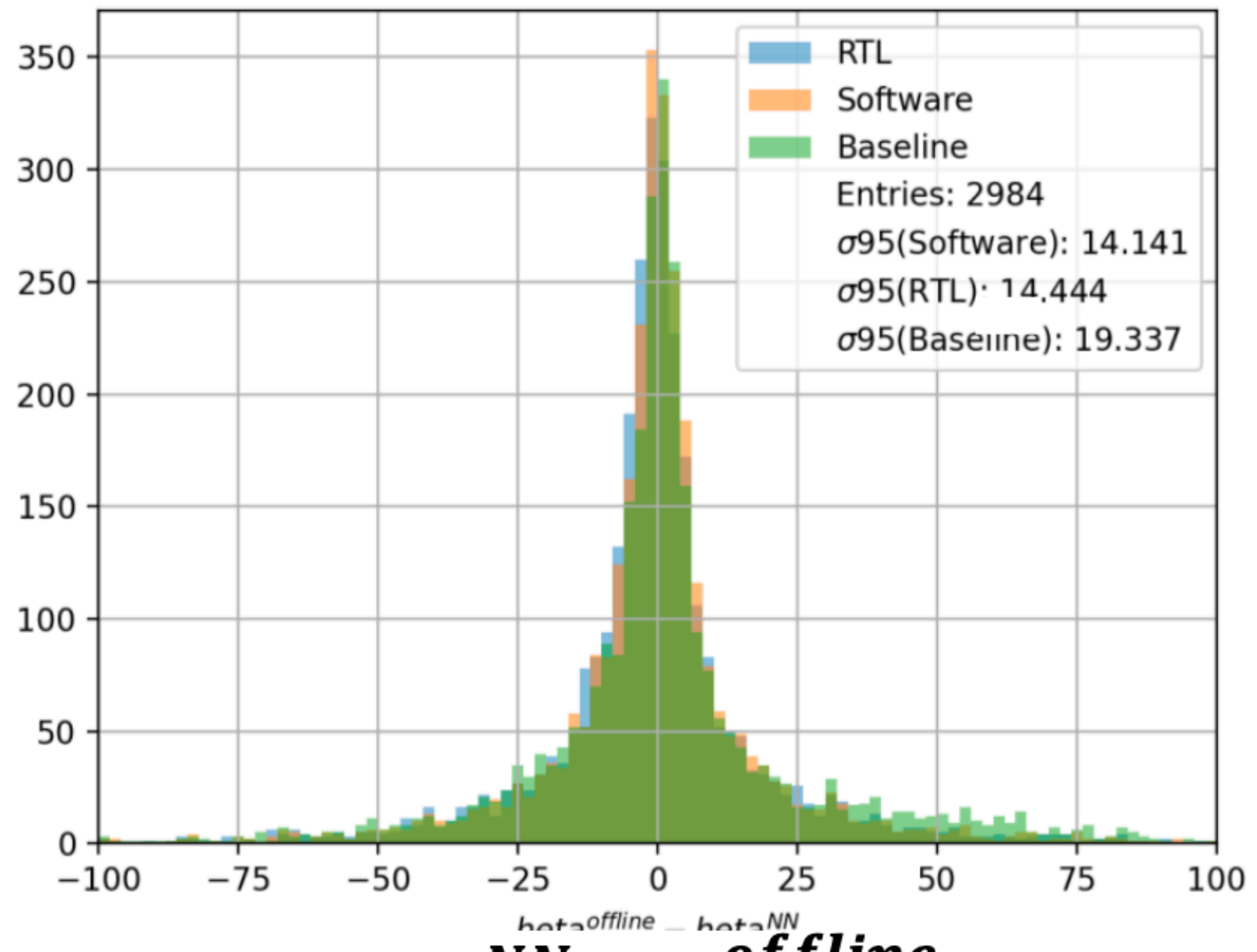
Performance of DNN algorithm

Delta track z



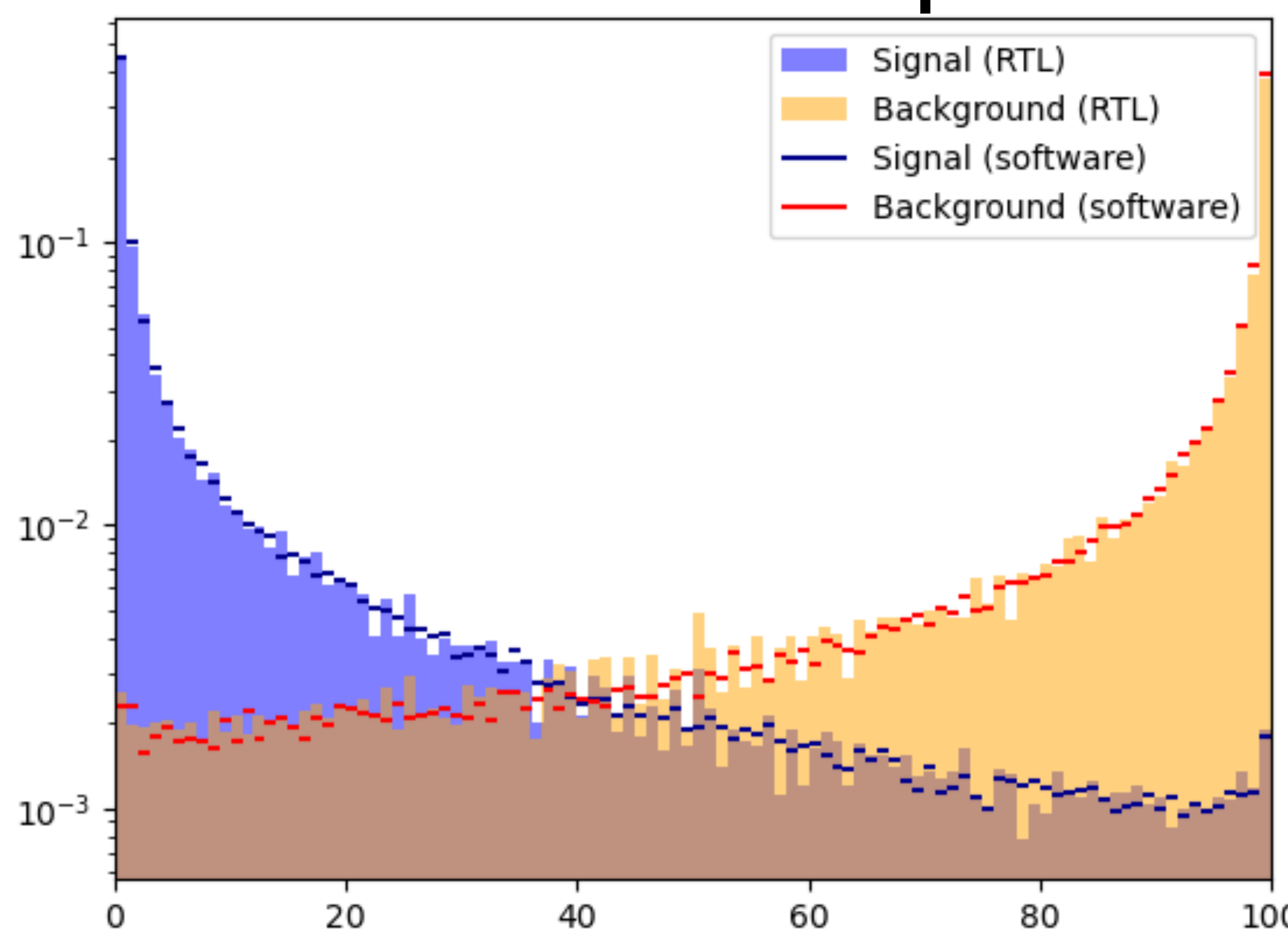
$z_0^{NN} - z_0^{offline} (cm)$

Delta track theta



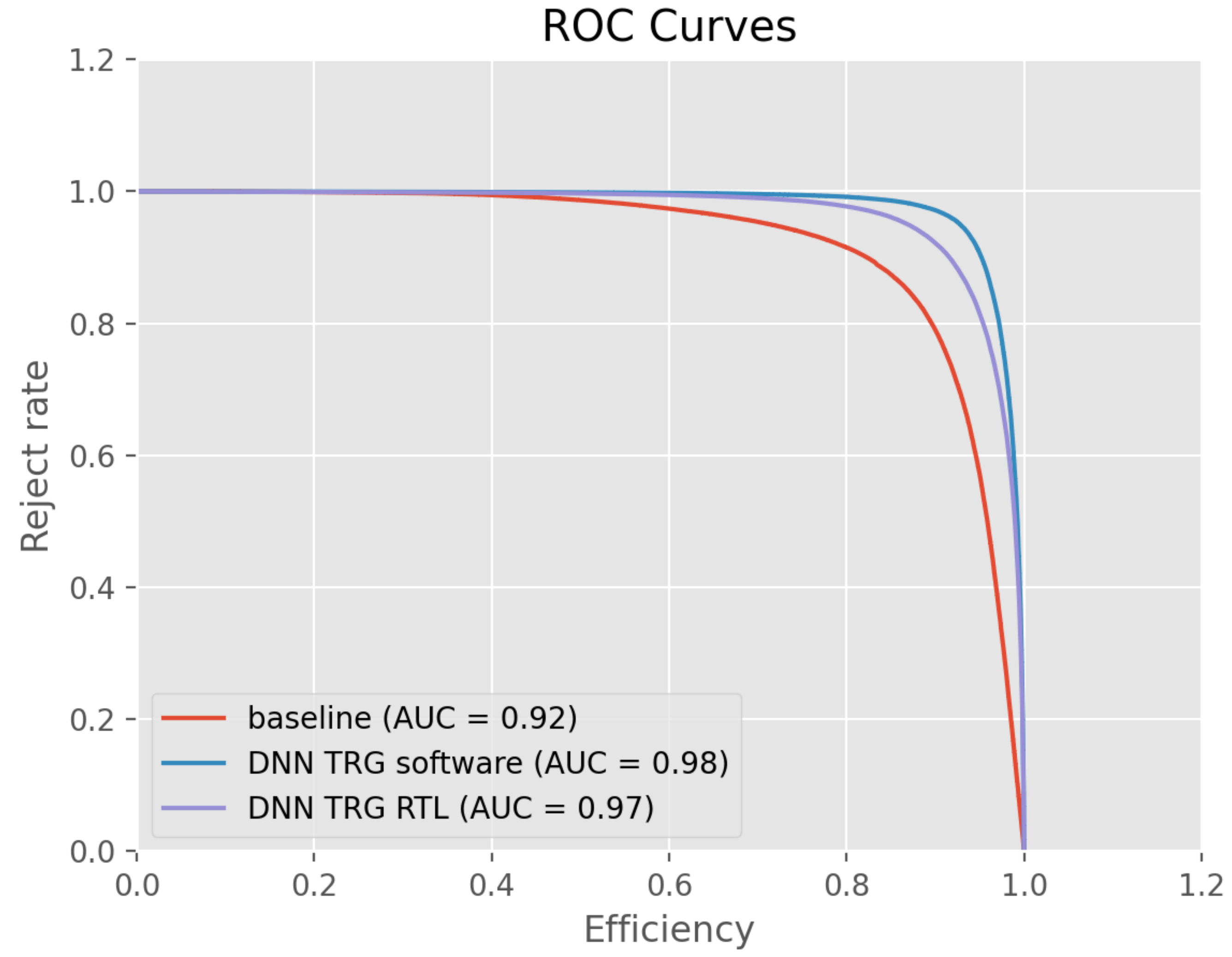
$\theta_0^{NN} - \theta_0^{offline} (°)$

Classifier output



Q (%)

Background rejection rate



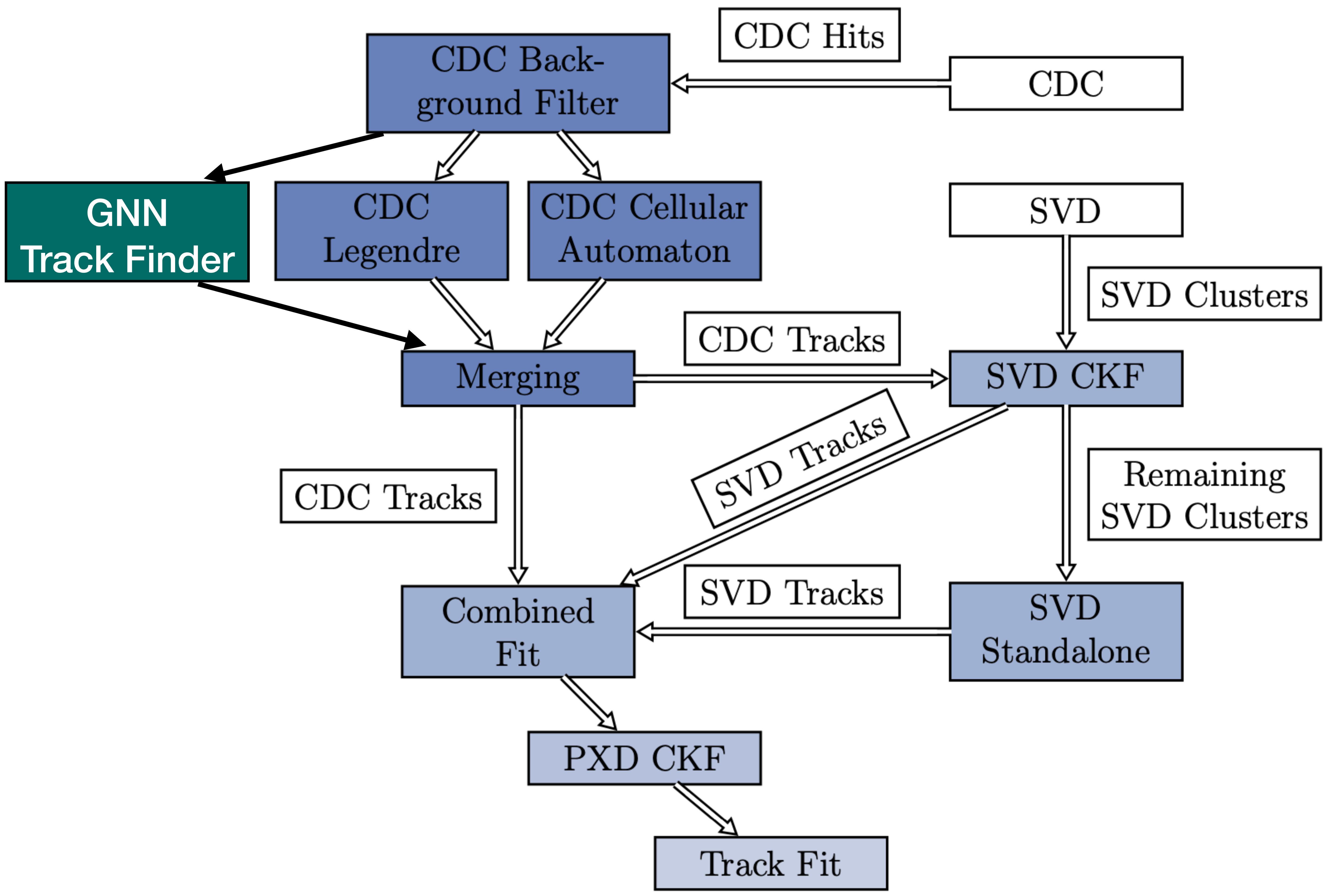
Signal efficiency

- Latency : 76 clock = **551.2 ns** ;require: < 600ns
- FPGA resource (UT4: Virtex UltraScale XCVU160) usage:
  - DSP: ~75%, LUT: ~45%, others <30%
- AUC do not get large drop comparing RTL and software simulation
- At signal efficiency ~95%
  - Background rejection rate ~85%

GNN based CDC track finder

- <u>Motivations of introducing a GNN track finder (**SOFTWARE**)</u>
- Low efficiency for displaced vertices
  - Efficiency decrease as displacement increase
  - Important signature for new physics search
- Higher background
- CDC wire inefficiencies
  - Bad wires or electrics
  - Decreased efficiency

Comput.Phys.Commun. 259 (2021) 107610

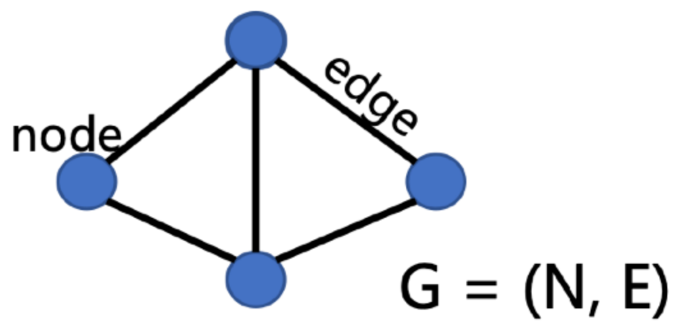


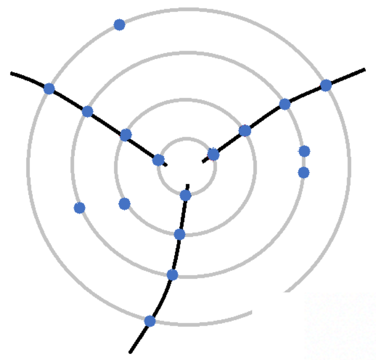
- Modular structure for track finding, with flexible of reconstruction sequence

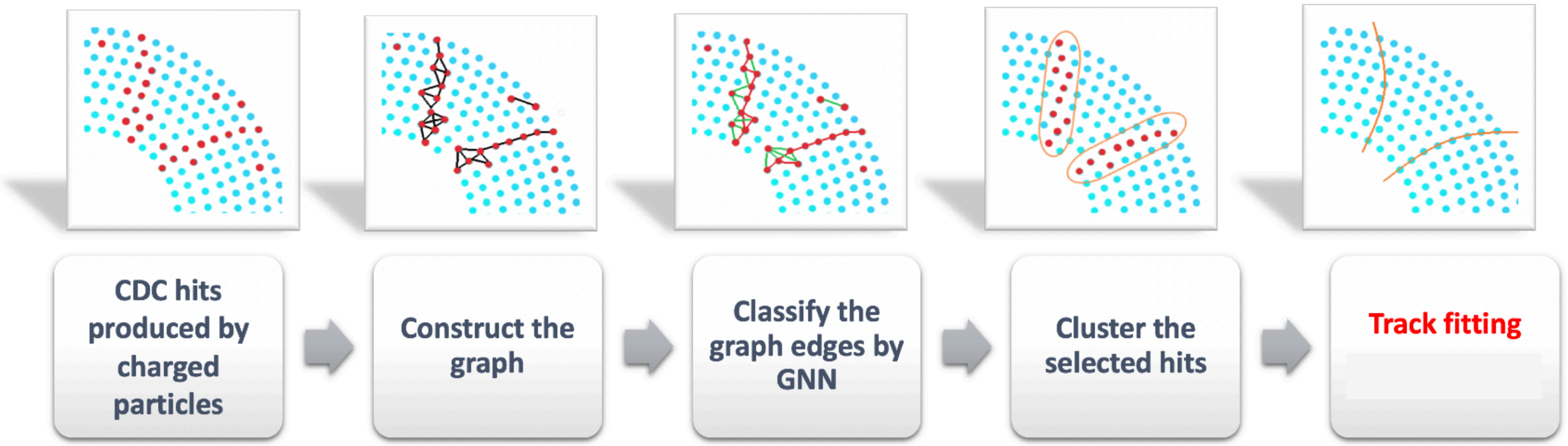
Model I: GNN for CDC track background filtering

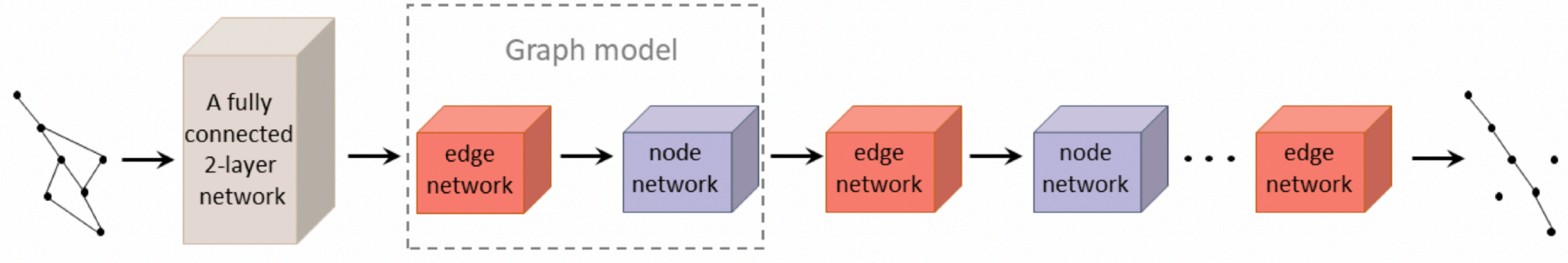
- Developed a GNN algorithm (based on X. Q. Jia (SDU) et al. BESIII's algorithm) for Belle II CDC hits clean up
  - Inputs: TDC, position coordinates r, ϕ

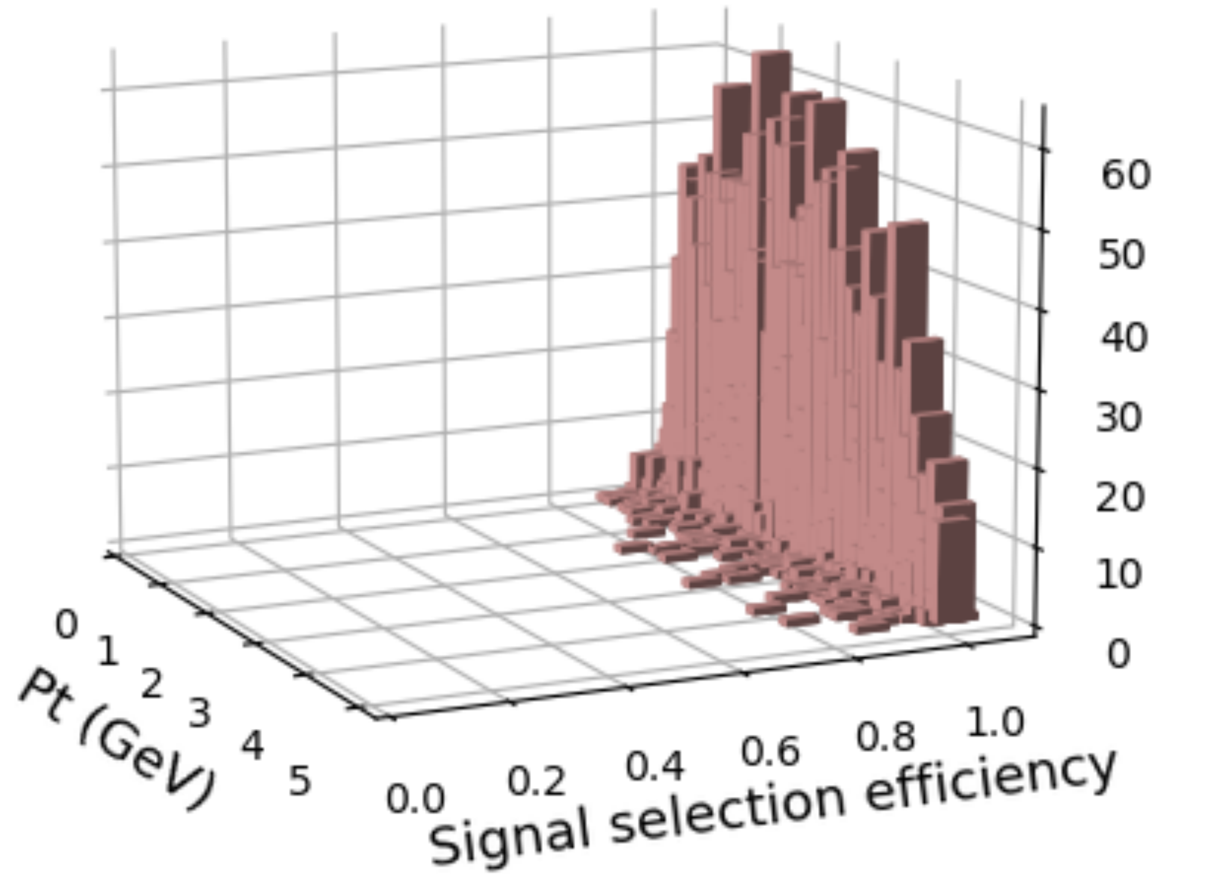
Xiaoqian Hu (SDU)





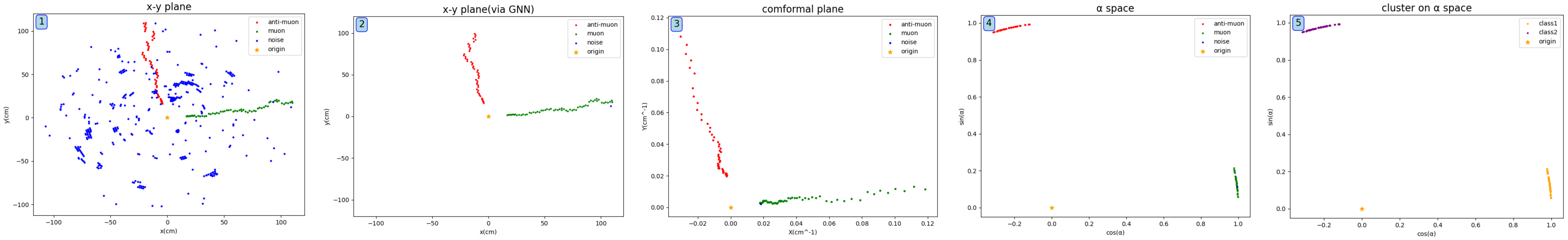






Hit selection efficiency: 98.4%
Hit selection purity : 97.9%

Belle II simulation (own work)



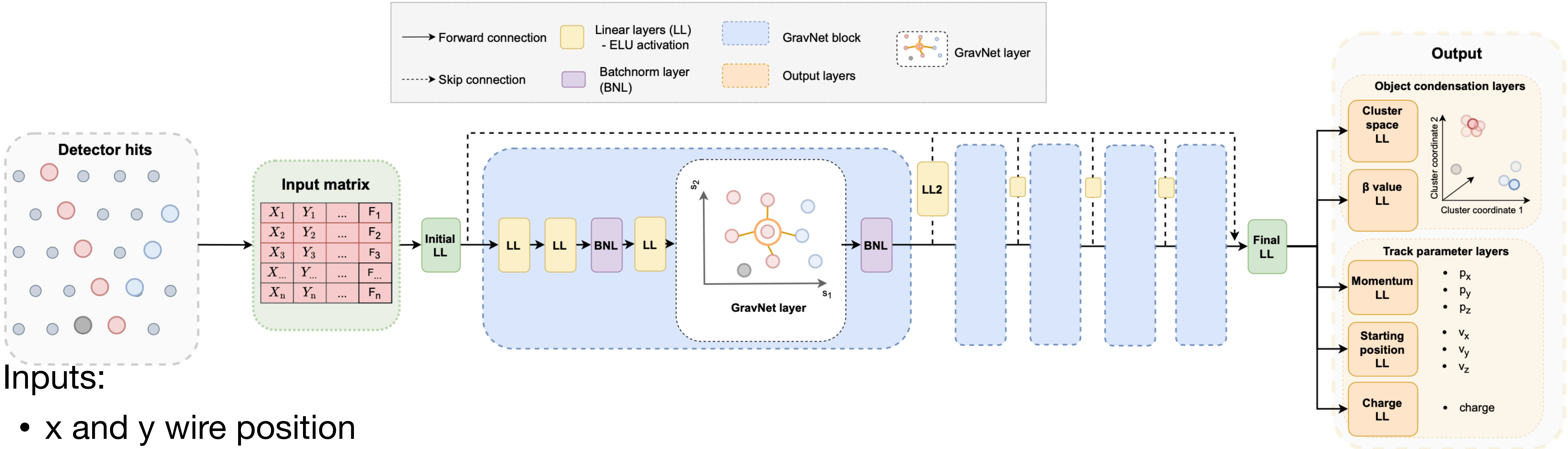
μ+ μ- (particle gun) | GNN noise filtering | Transform space | Transform α space | DBSCAN clustering

Model II: GNN for offline track finding

- Find track parameters: momentum, starting position and charge
- Find unknown number of tracks → **Object Condensation** (arXiv:2002.03605)
- Computing resource and time constraint may reducible

Noise filtering → Clustering → Fitting

L. Reuter et. al. (KIT) arXiv: 2411.13596

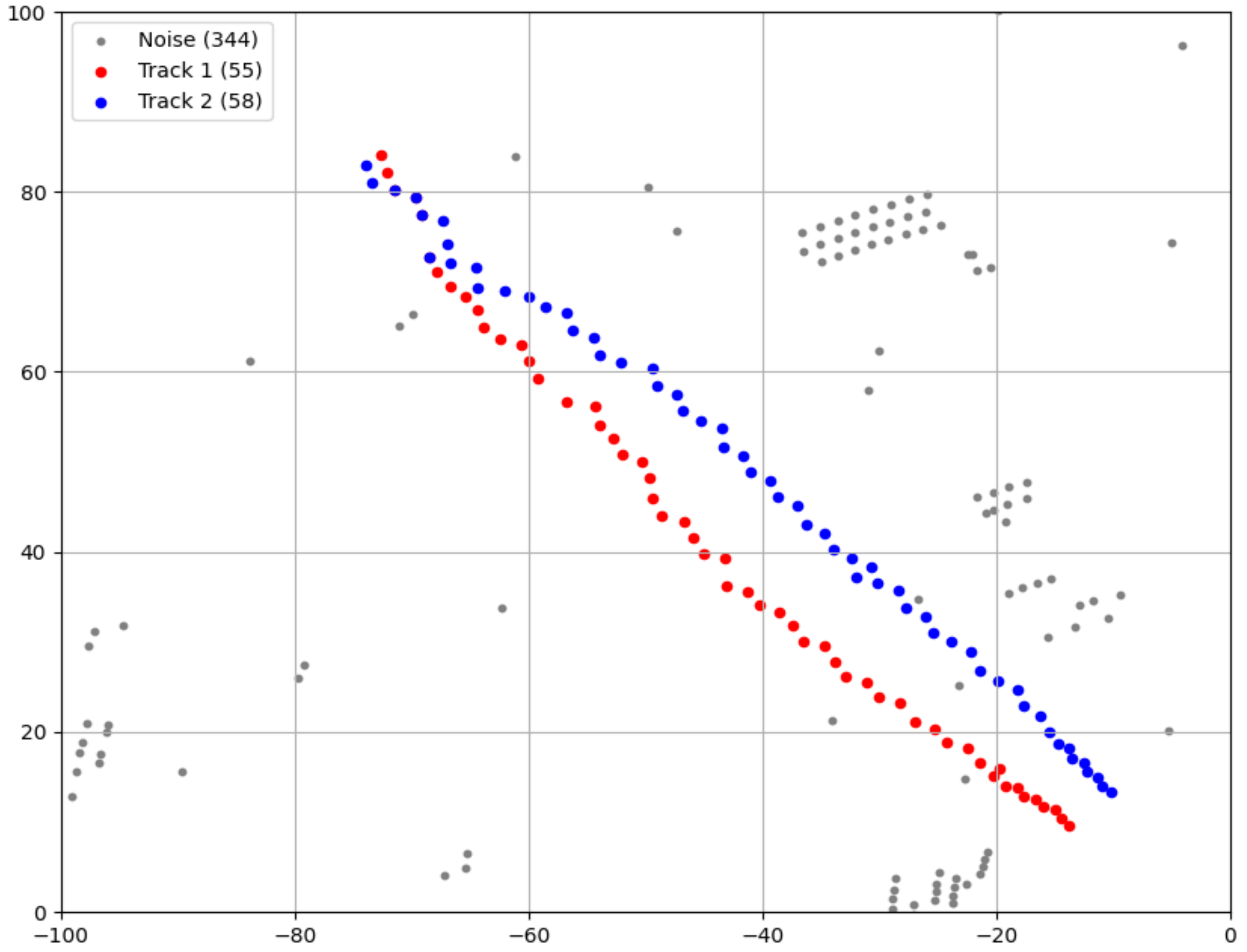


- Inputs:
  - x and y wire position
  - TDC and ADC of signal information
  - layer, superlayer, and layer info. with suprlayer
- Adjustable Parameters
  - 797,812 trainable parameters (3MB weight files)

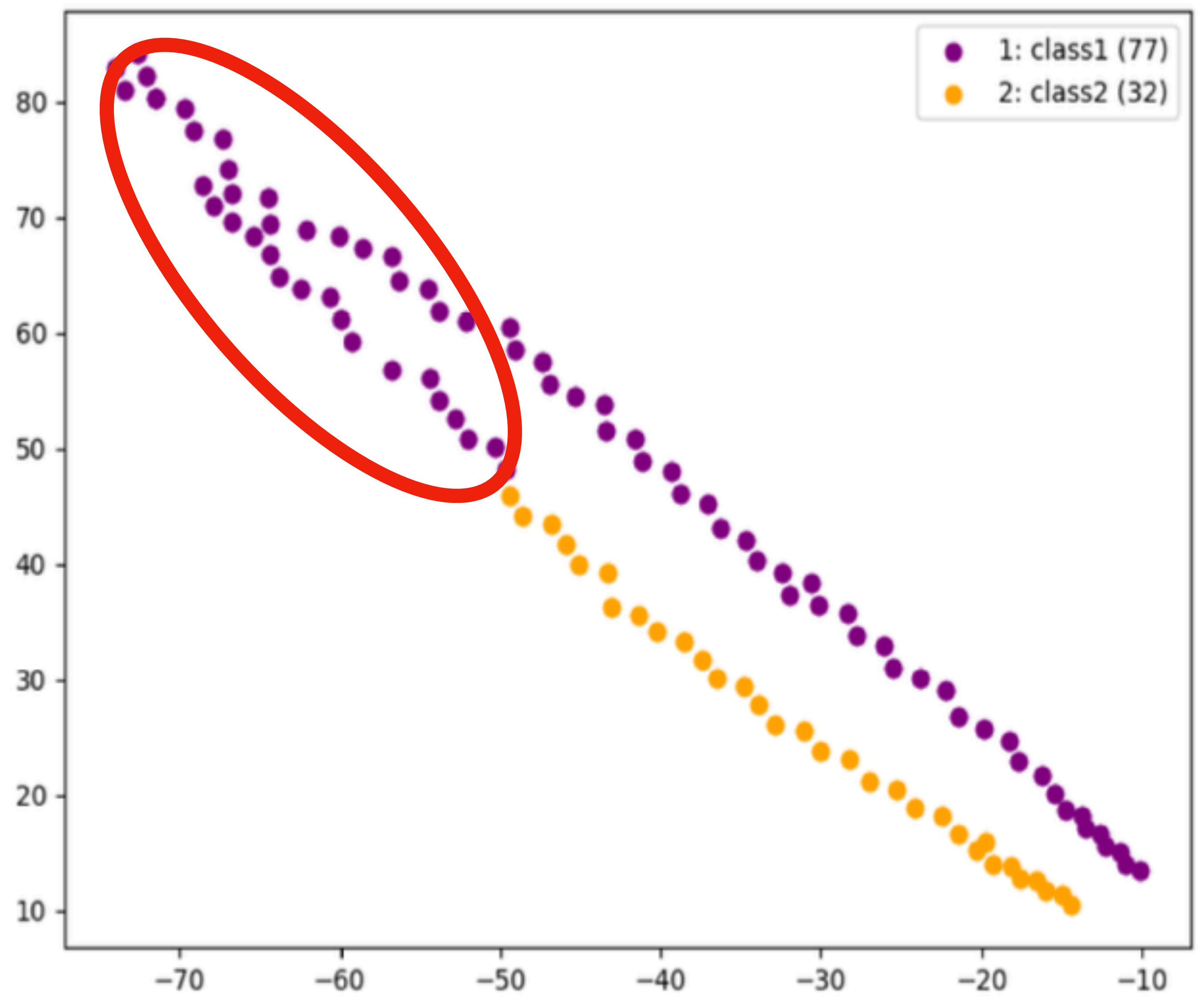
Performance of GNN

- Model II (Object condensation) shows better clustering performance than model I
- Track finding efficiency do not increase that much
  - Failed in the track finding, even clustering of hits shows better performance
  - Further improvement is needed

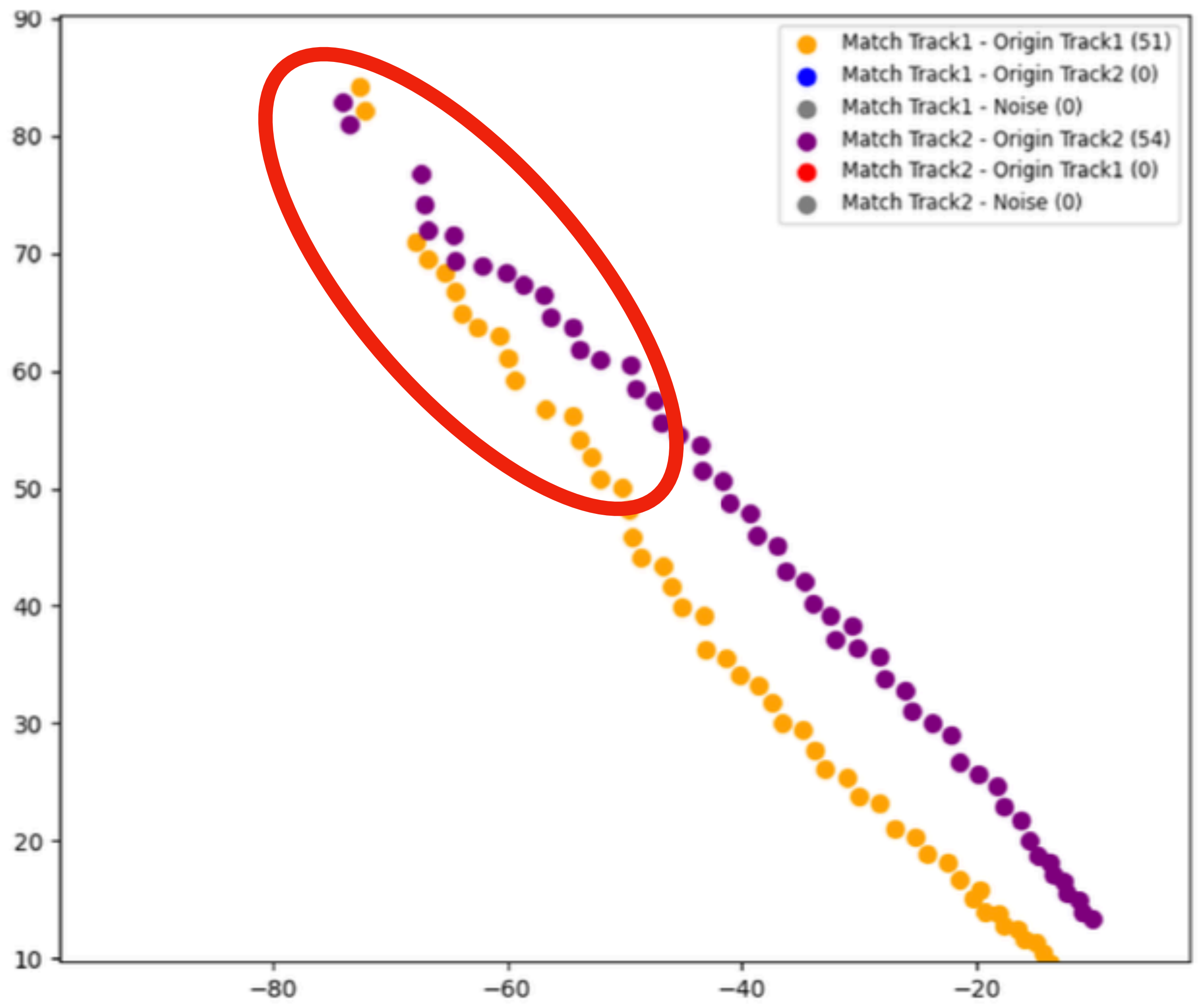
Belle II simulation (own work)



Model I GNN

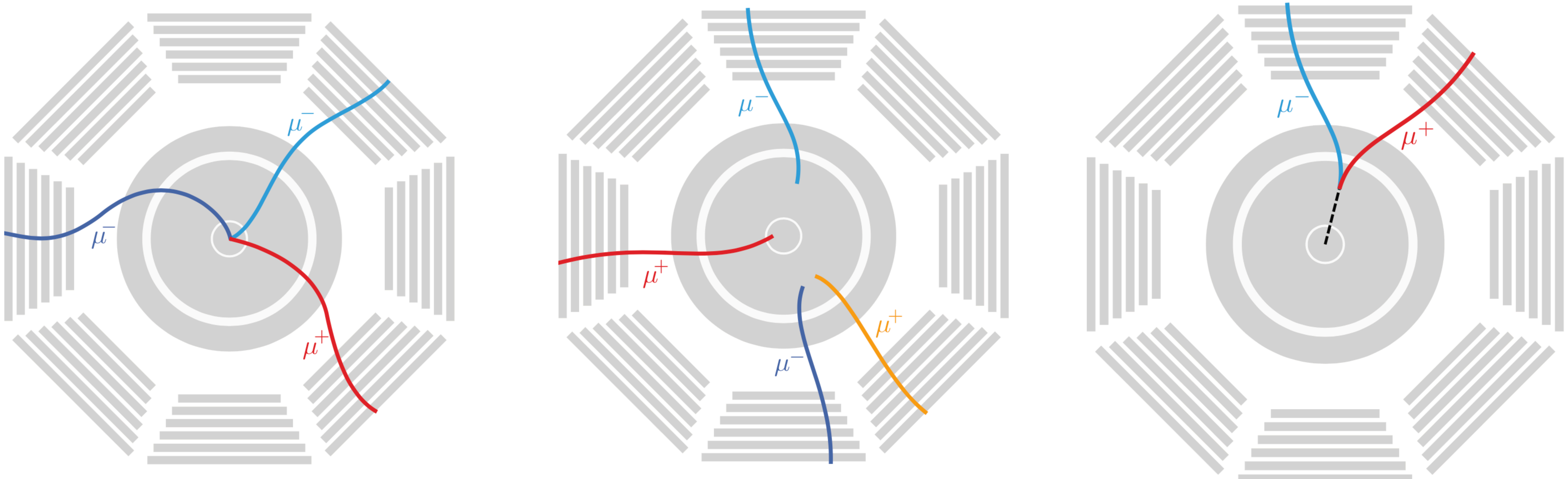


Model II GNN



Training samples for GNN

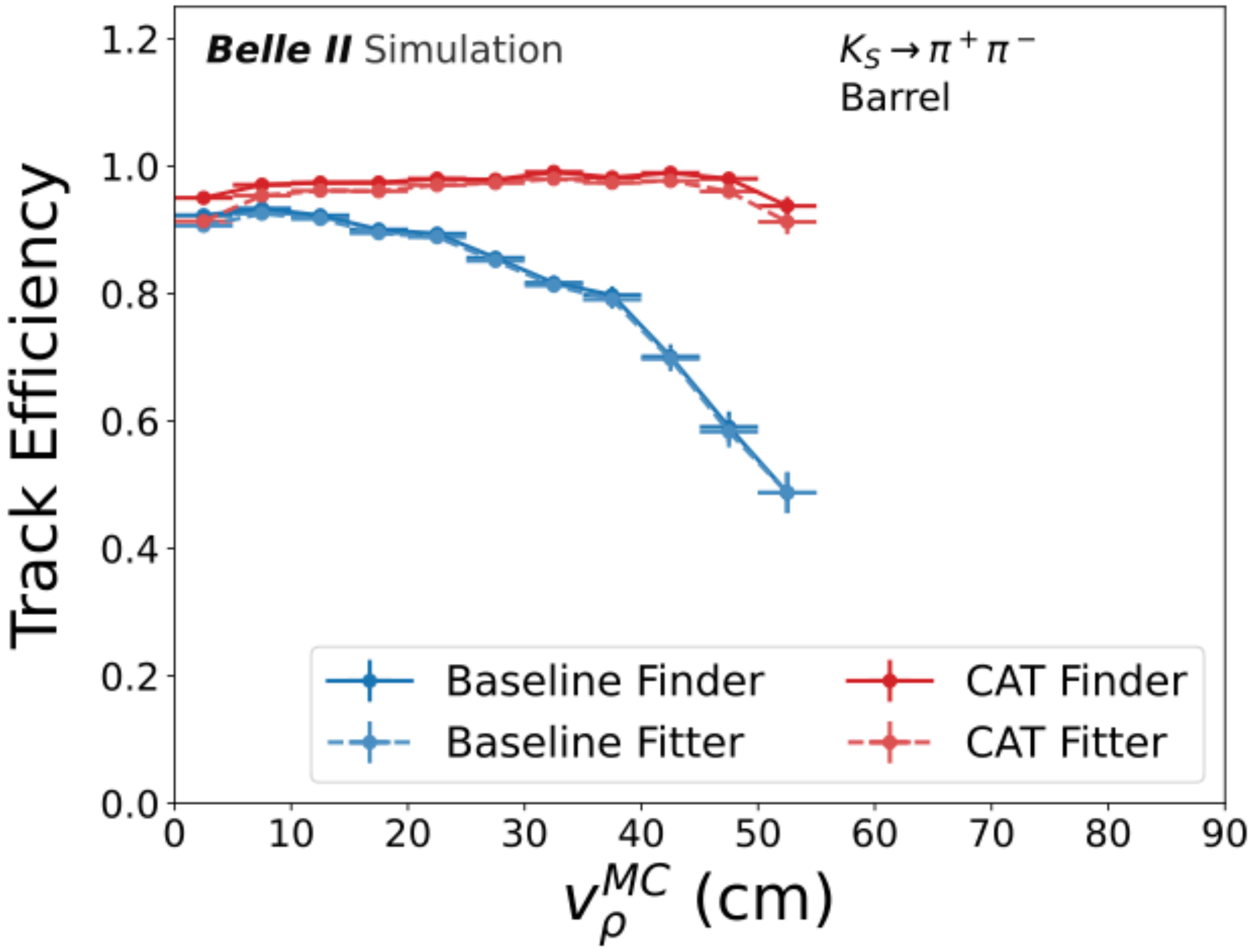
- Simulate 1 million events with over 4 million tracks
  - Train: Validation = 4 :1
- Training samples contain different topologies that cover all interested event features, to not bias the model, **no conservation laws involved here!**
  → crucial step to be agnostic about the physics processes
- Sample features
  - Low momentum tracks forming circles in the CDC ($P_t < 0.4$ GeV) <-> High momentum tracks
  - Short tracks <-> tracks penetrate all CDC layers
  - Small opening angle <-> well isolated two tracks
  - …



Performance of GNN

- Efficiency of displaced vertex tracks improved from 85.4% with a fake rate of 2.5%, compared to 52.2% and 4.1%
  - The other performance similar as original algorithm
- Momentum p_x , p_y , p_z starting position v_x , v_y , v_z ,charge
  - Provide initial inputs for GENFIT
- GNN prediction is drawn according to the track parameters predicted by the GNN

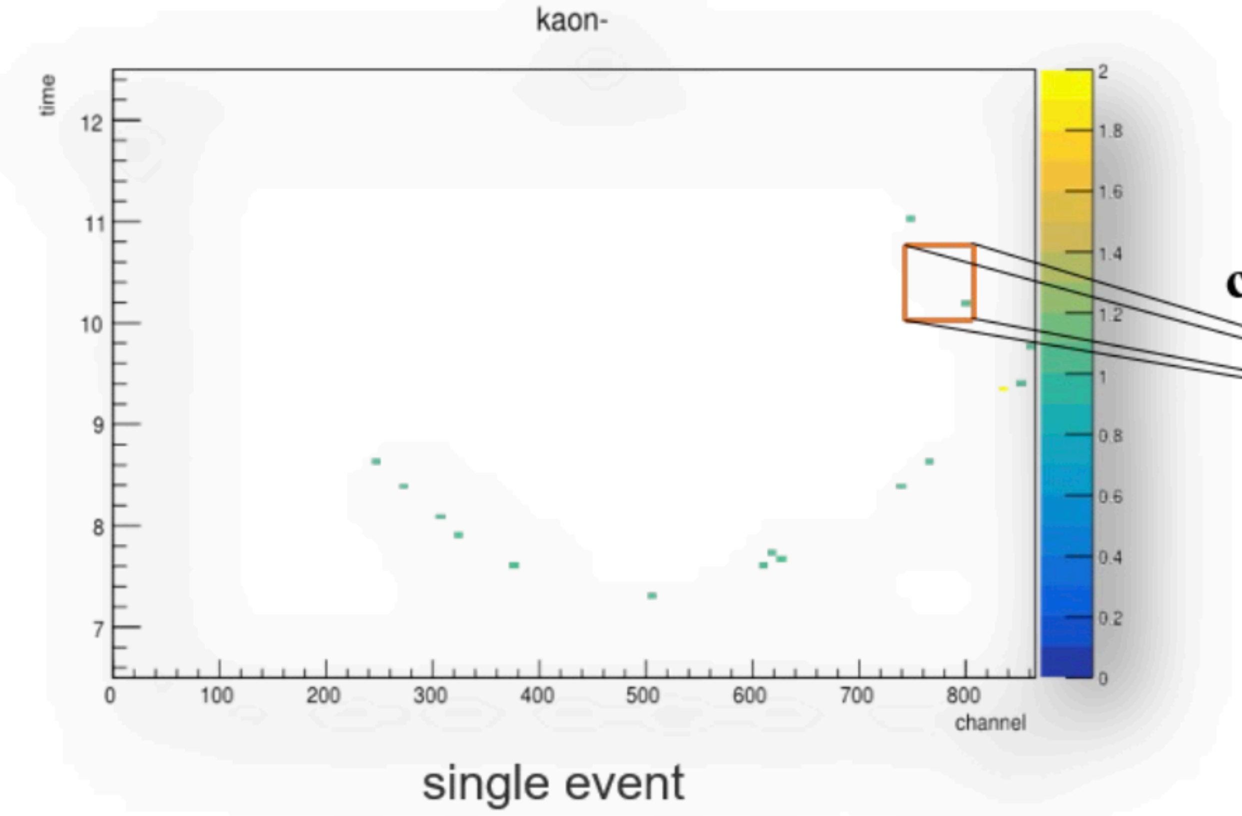
L. Reuter et. al. (KIT) arXiv: 2411.13596



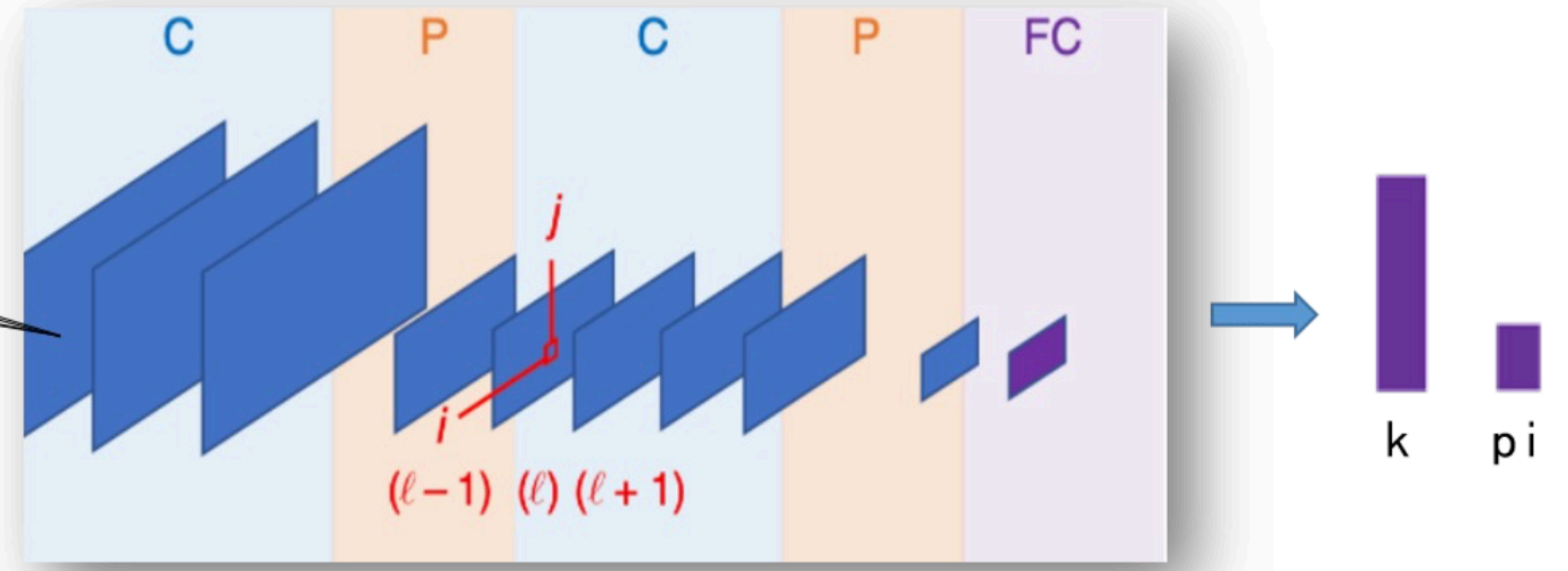
CNN algorithm for STCF PID

- DTOF as a PID subdetector of STCF
- CNN algorithm developed for Kaon/pion identification
- Kaon/Pion MC simple, 800w

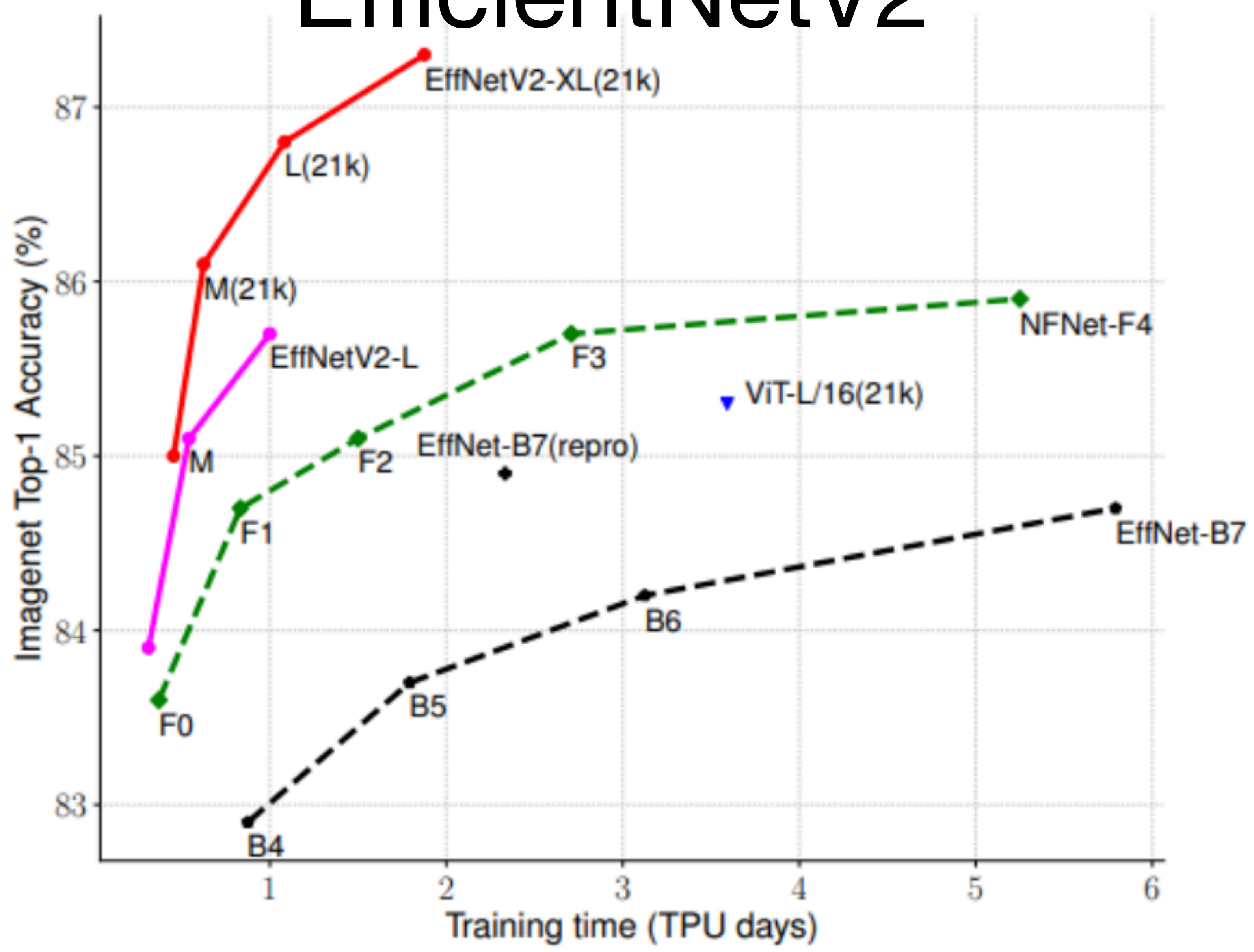
Z. Yao et al.@SDU

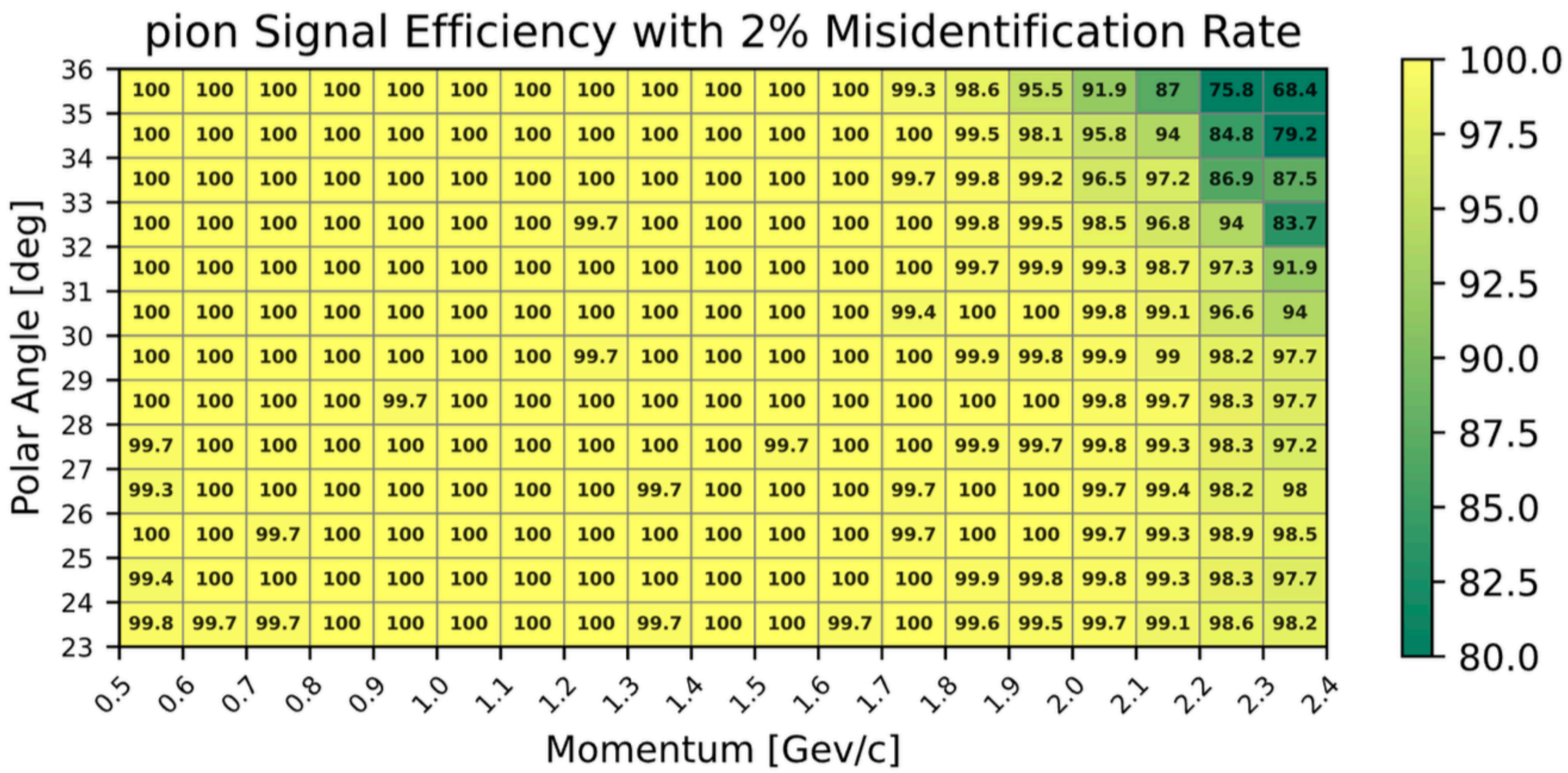


single event



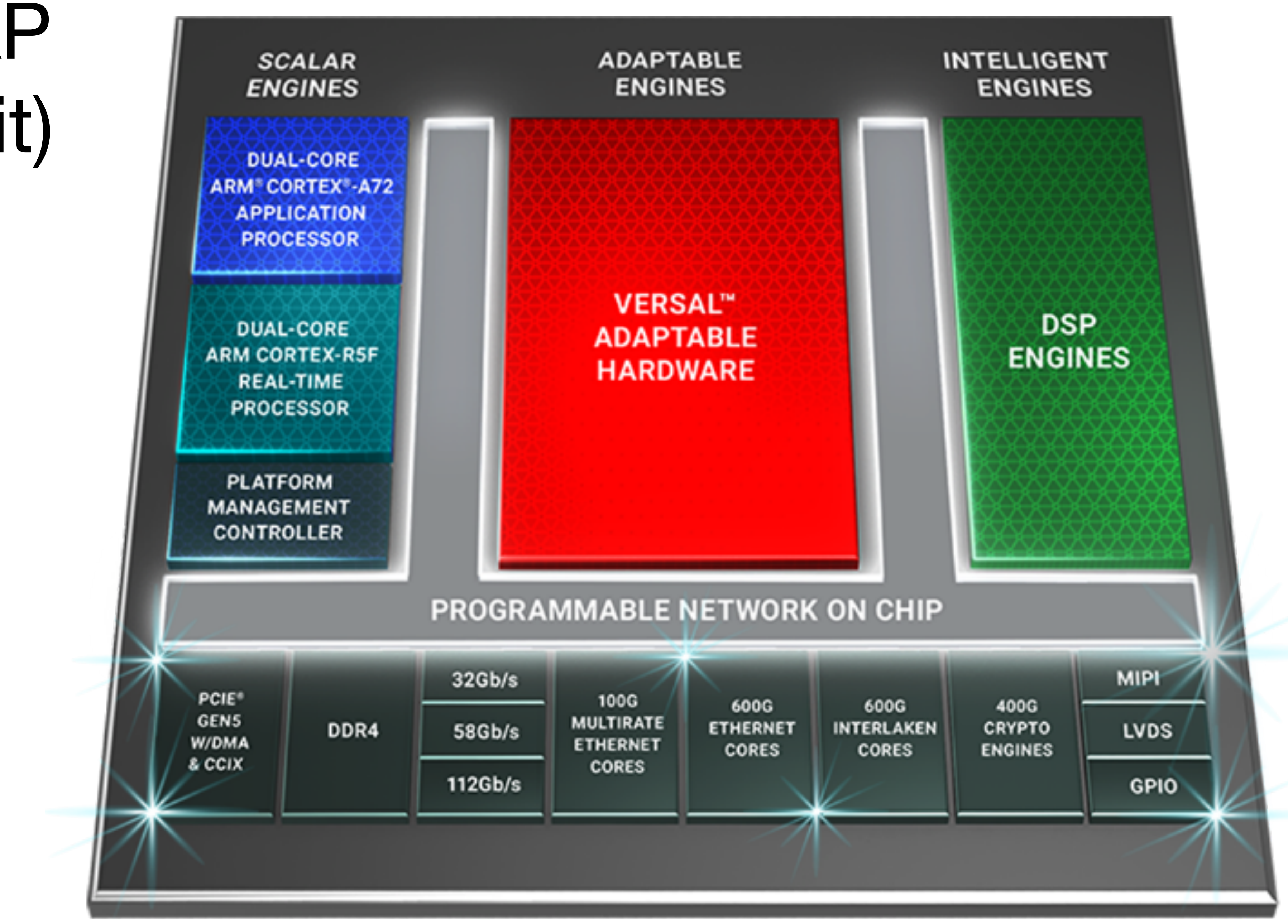
EfficientNetV2





Heterogenous computing platform

- R&D of a new general FPGA device using the AMD Versal ACAP
    - Heterogenous acceleration (VCK190, VCK5000 evaluation kit)
        - AI engine (AIE)



UG1079

Figure 2: **AI Engine Array**

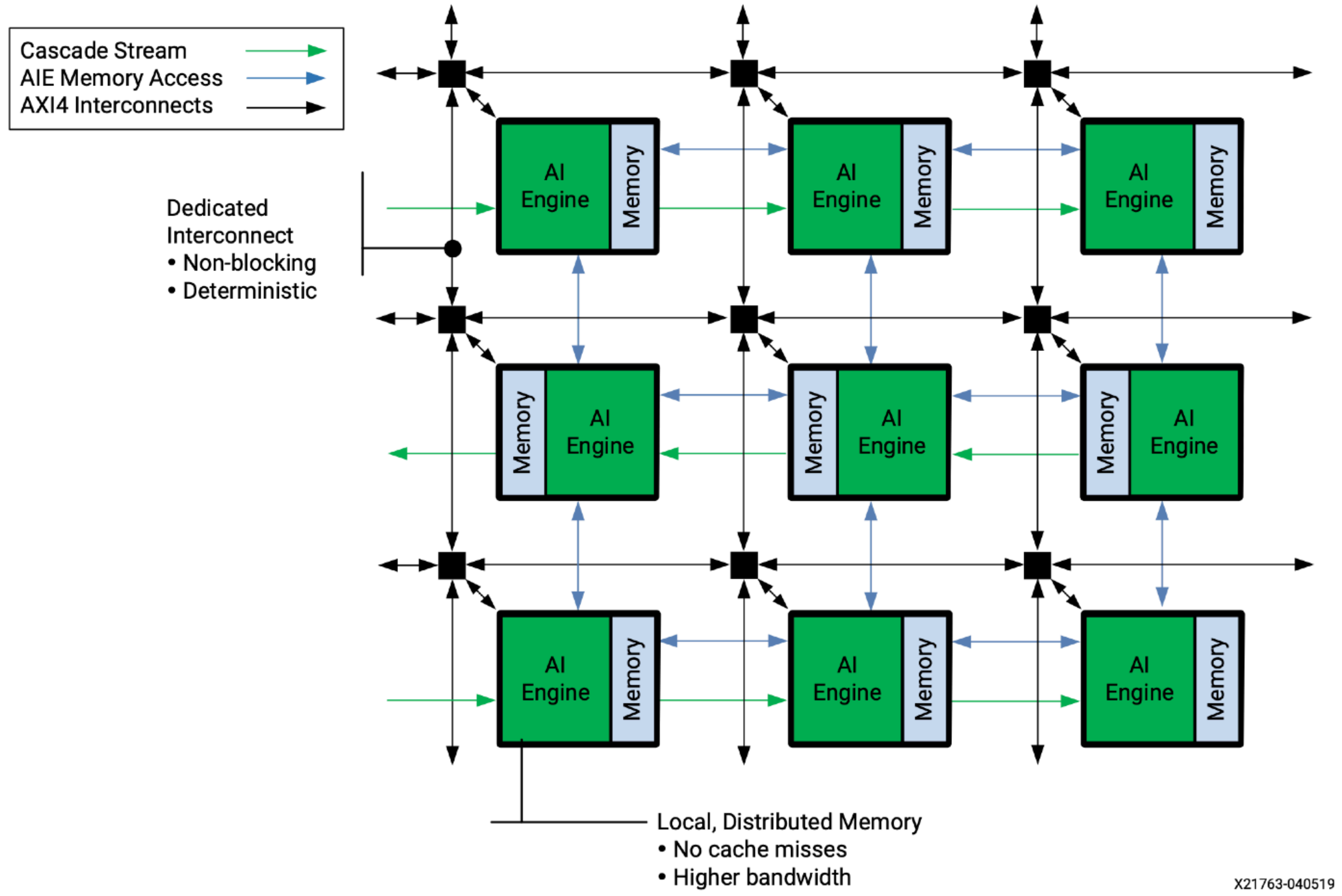
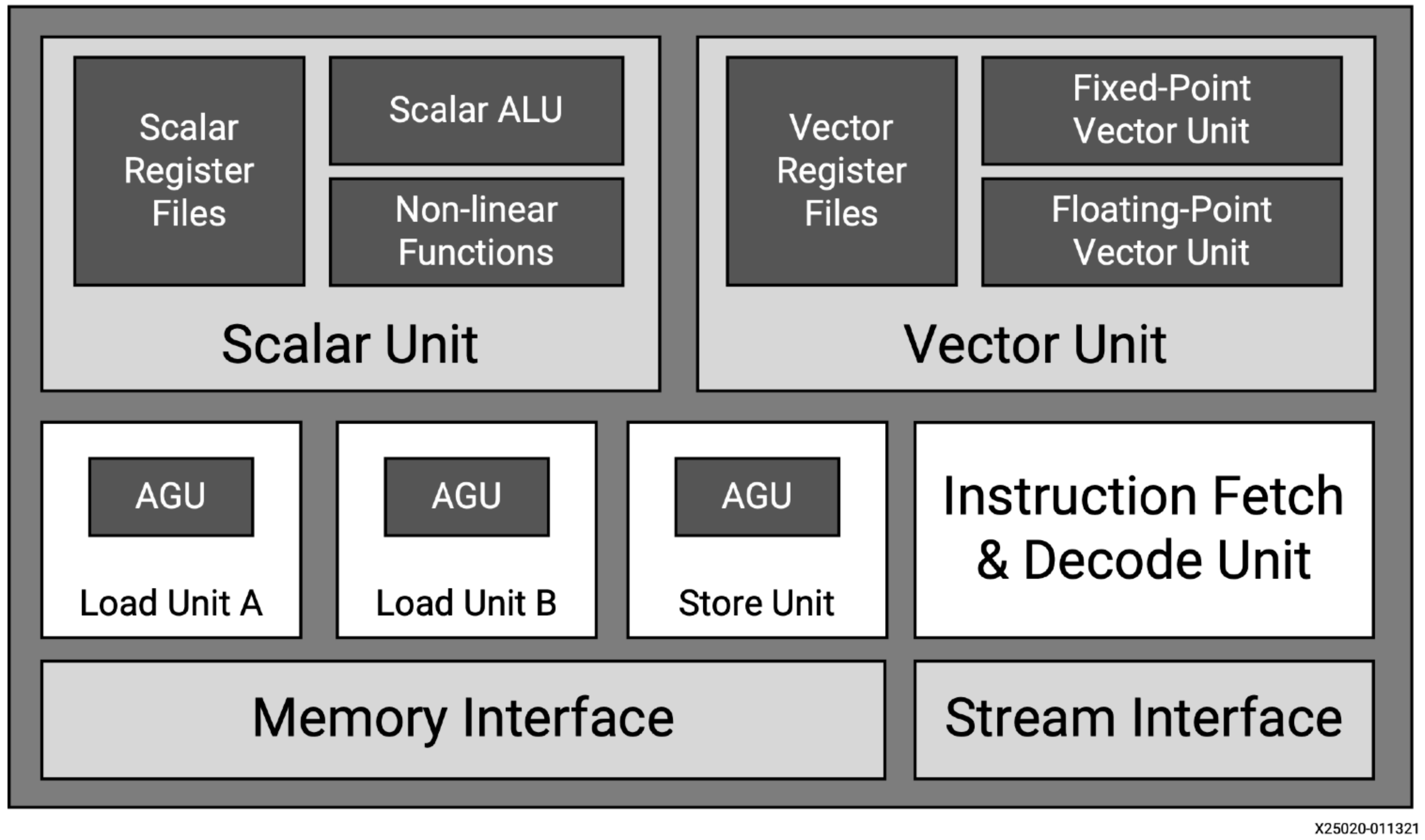
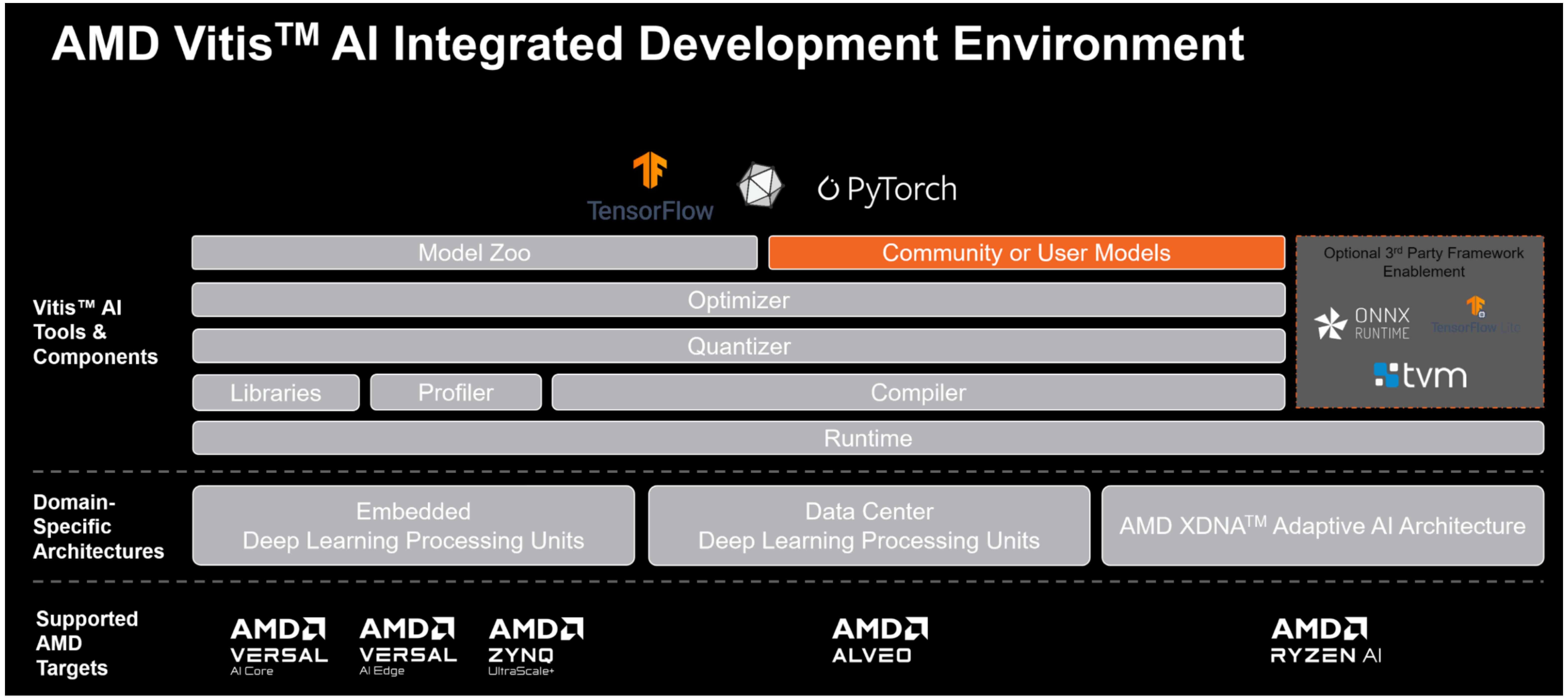
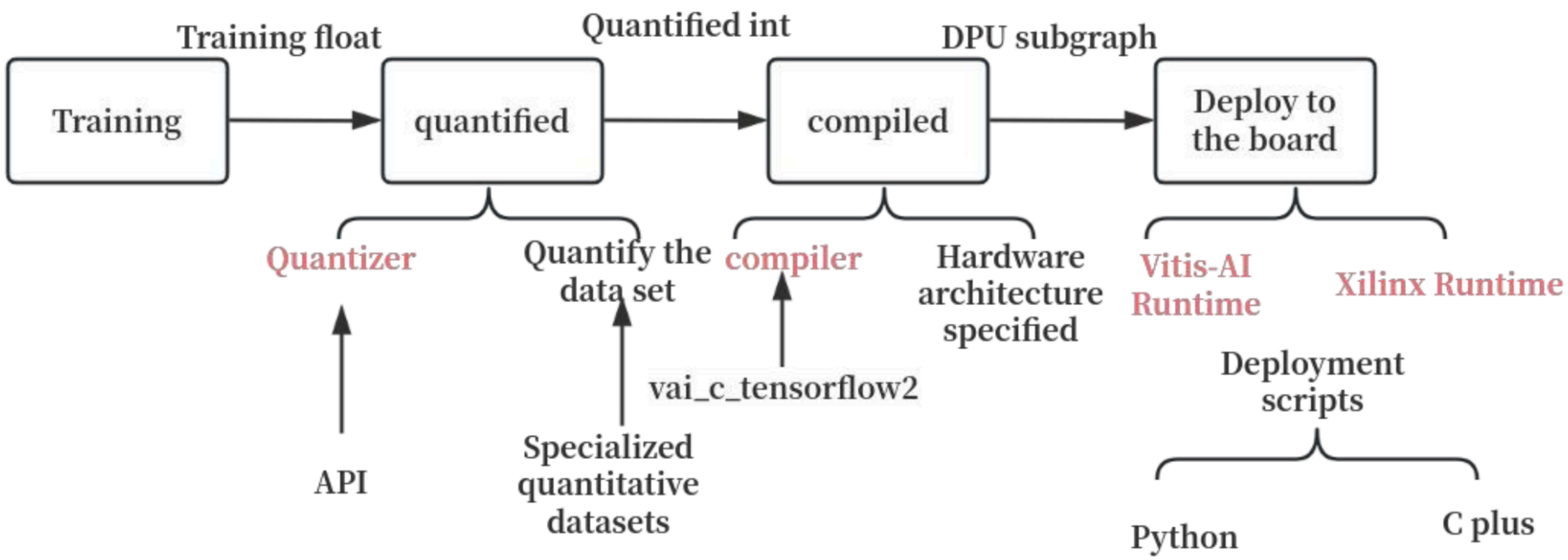


Figure 4: **AI Engine**



CNN algorithm implementation



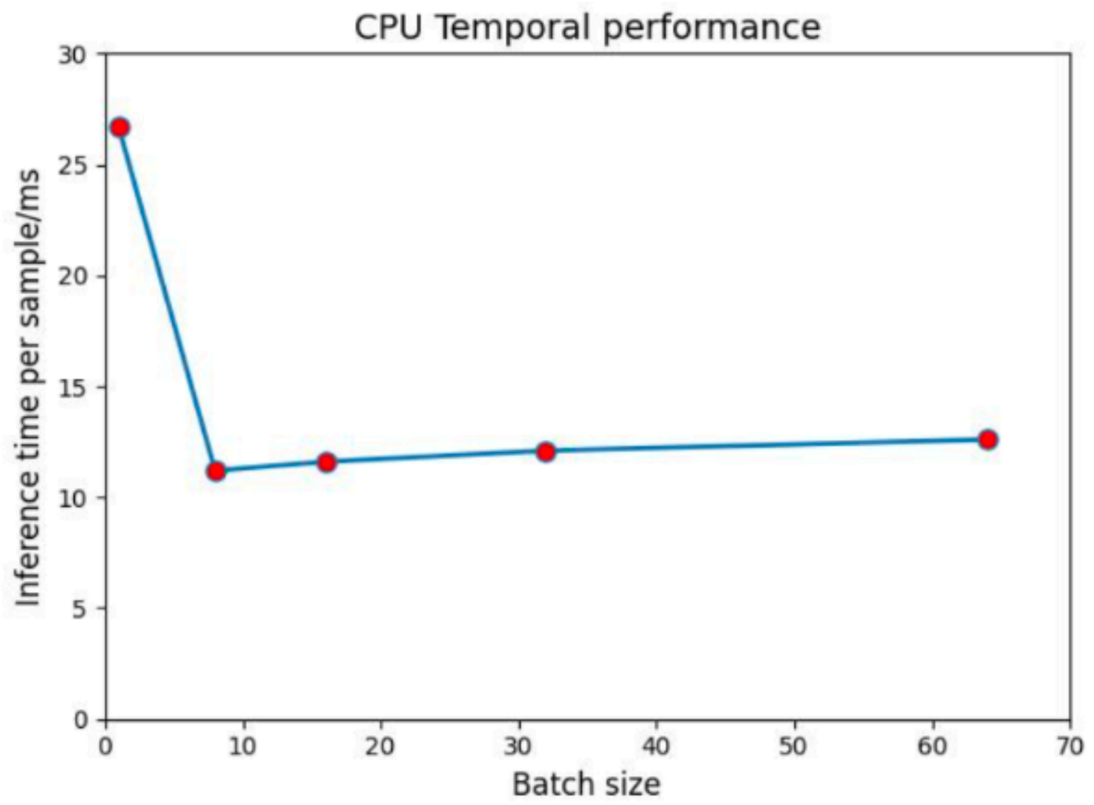


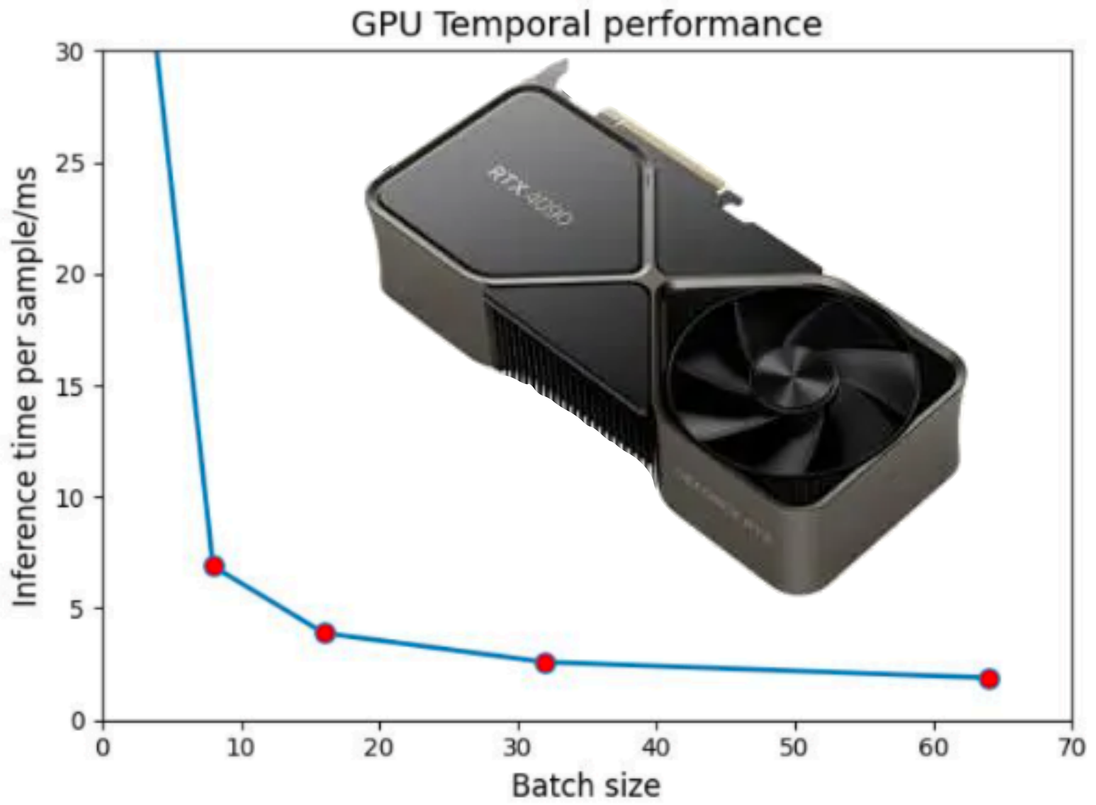
CNN algorithm implementation

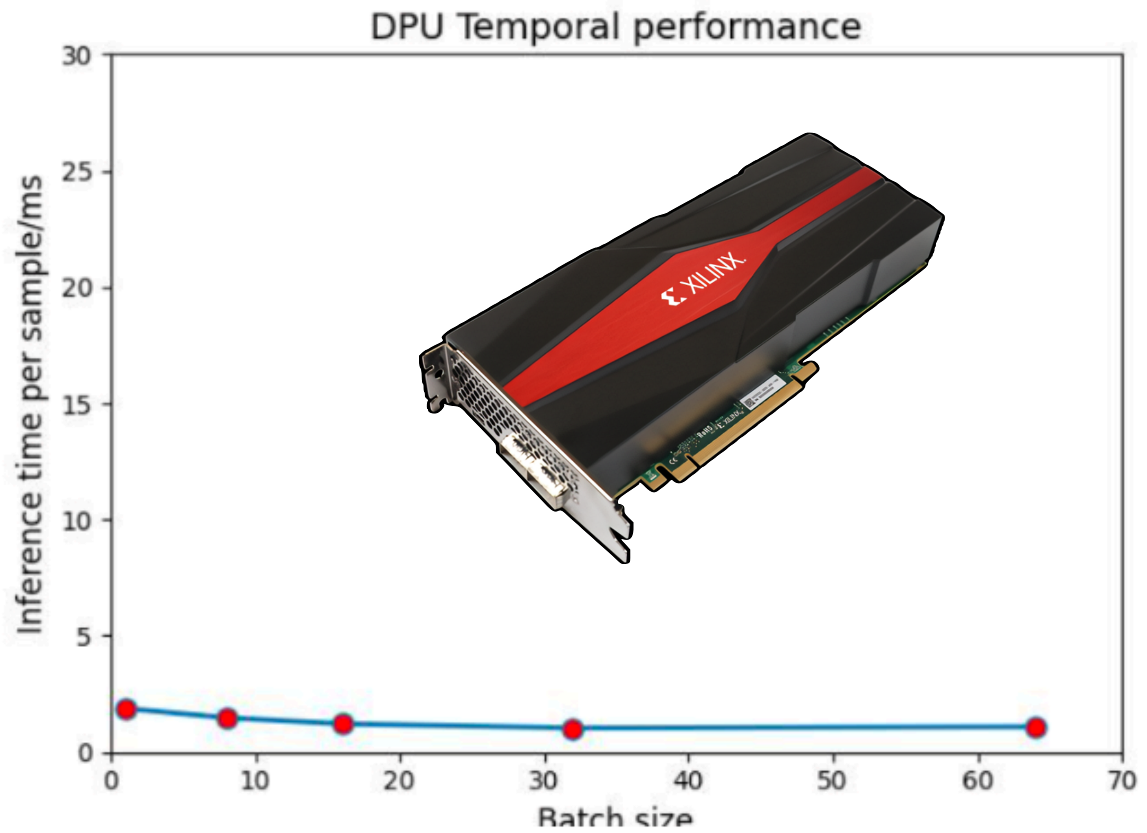
Inference result based on 10000 samples

CPU | GPU (RTX4090) | DPU (VCK5000)

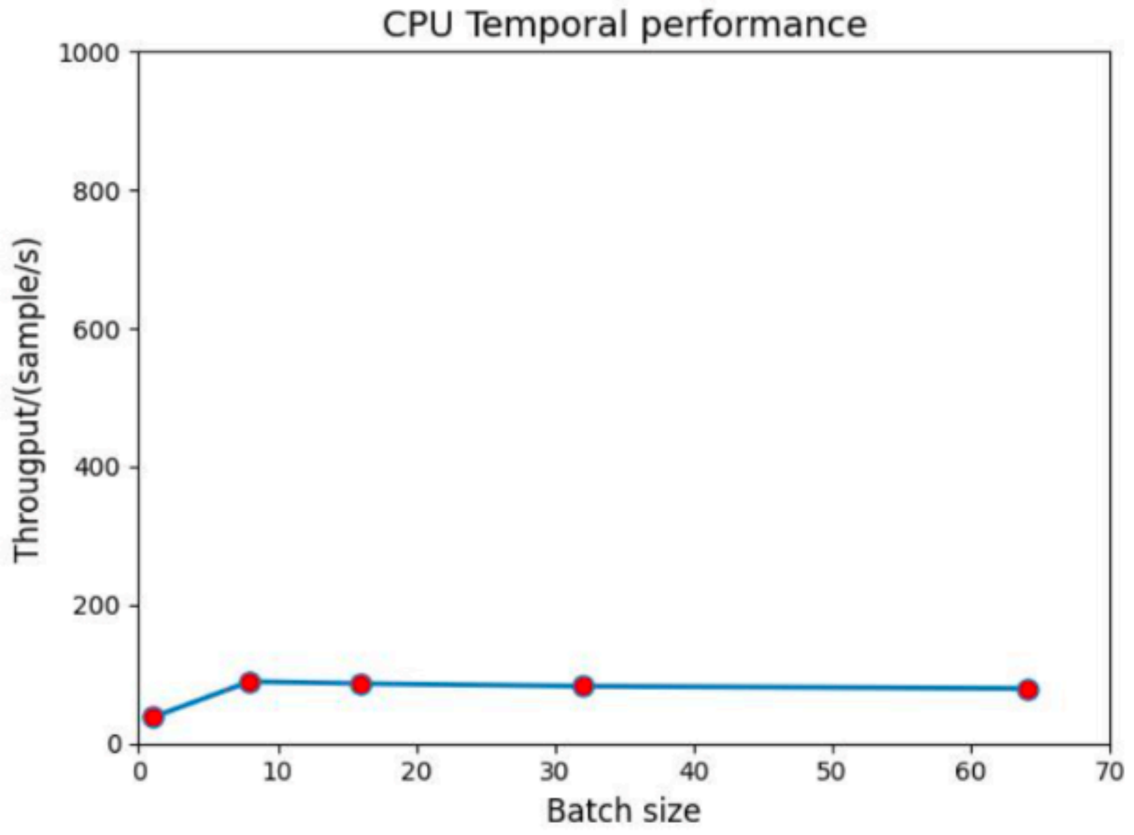
Inference time

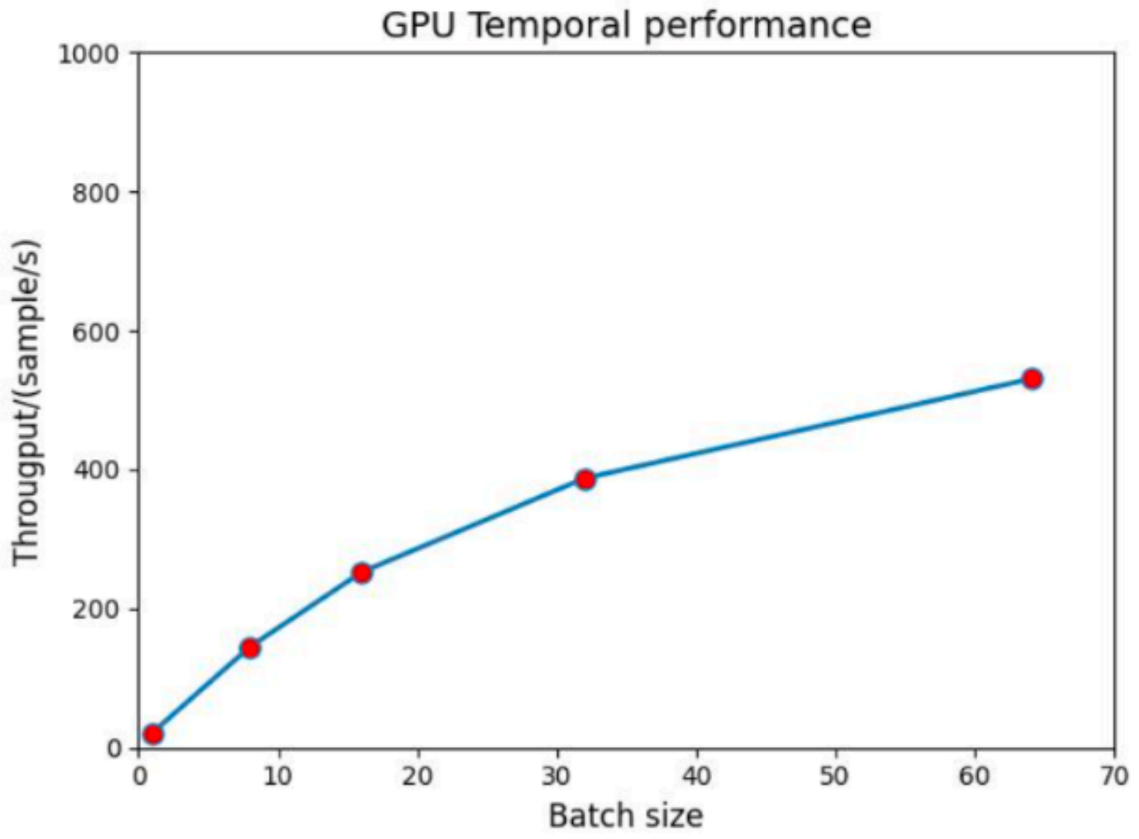


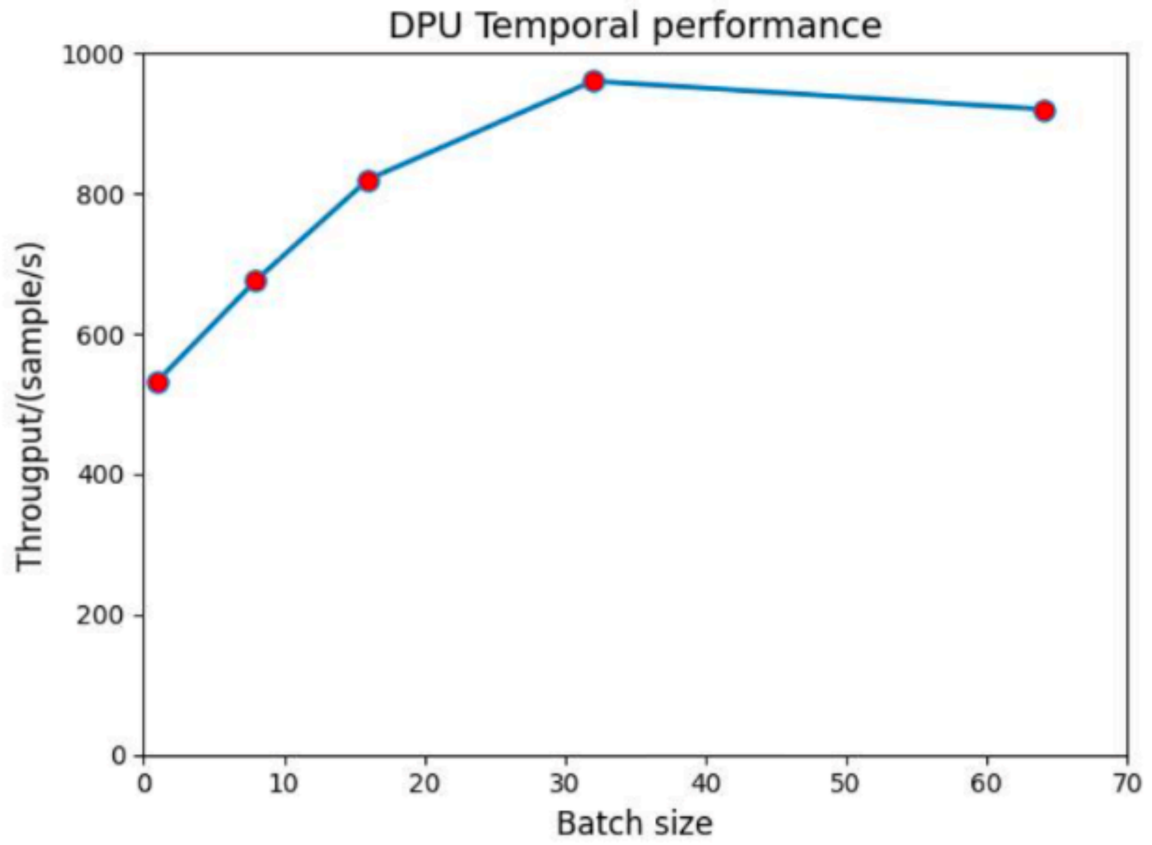




Throughput







DPU based on AMD Versal ACAP shows ~13 times(CPU)/~3(GPU) faster inference time

Summary and prospects

- Advanced data reduction technique is essential for next-generation HEP experiment.
  - AI/ML integrated with heterogenous computing acceleration
- DNN with hardware based L1 track trigger for improving background rejection
- GNN based hit filter, and DNN were implemented on AMD Versal ACAP
- CNN based PID algorithm for STCF running on DPU has advanced performance than CPU/GPU
  - Both GPU and DPU can be a solution to accelerate data processing for online or offline data processing system