Development of real-time online tracking system

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

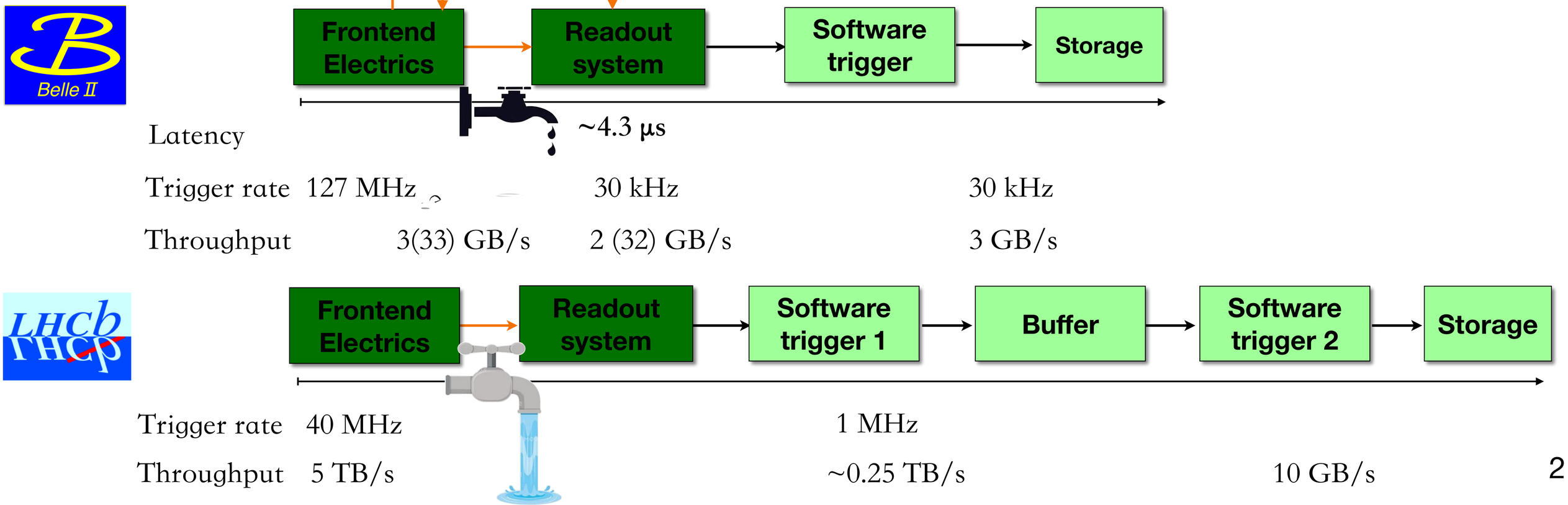
21-23 Jul. 2027, Huizhou
2nd Workshop on Tracking in Particle Physics Experiments

Readout system (Belle II vs. LHCb)

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

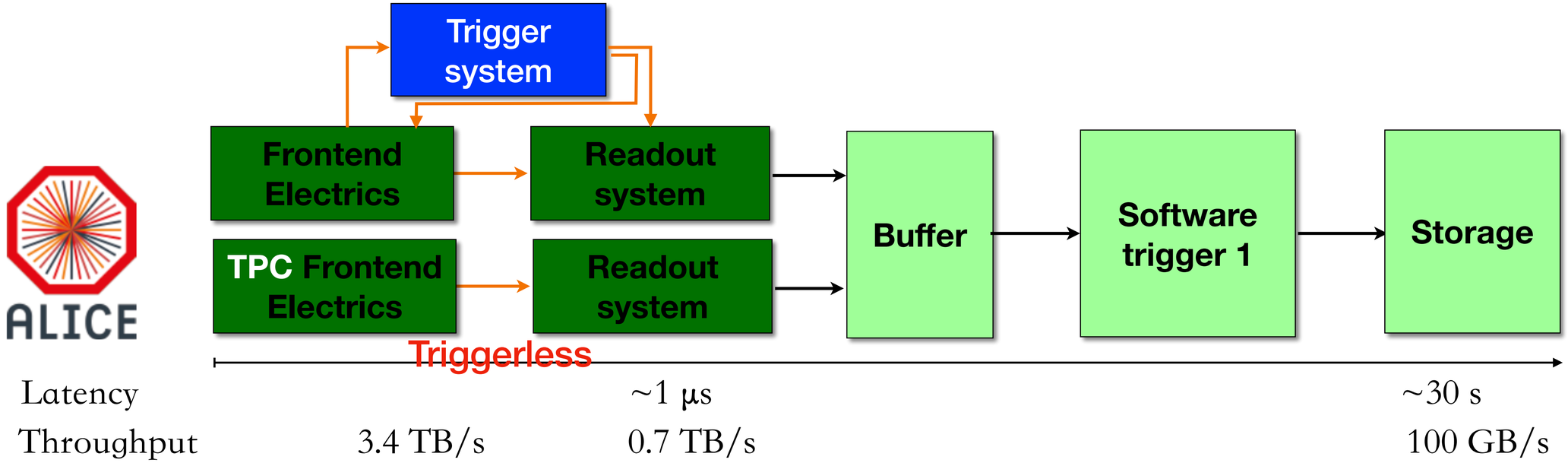
      70~95%

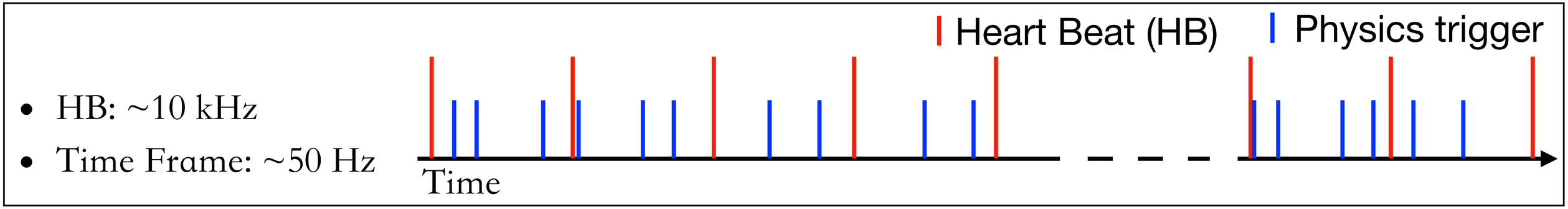
- LHCb: "triggerless" readout & DAQ
  - CPU+GPU based software trigger
  - Rate of physical process: ~MHz
    - No hardware trigger available



Readout and DAQ system(ALICE)

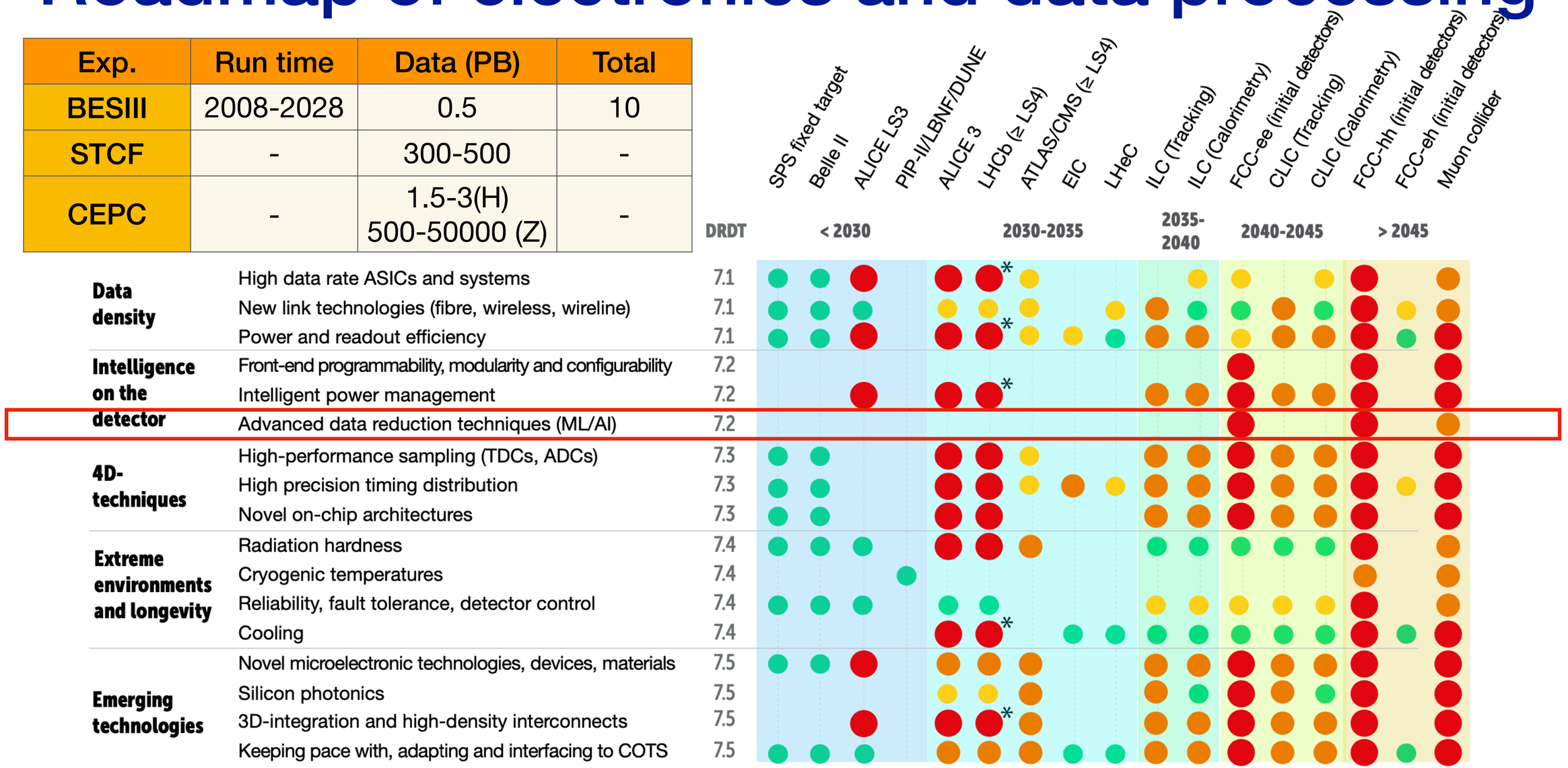
- ALICE: continus readout
  - TPC w/ triggerless readout + others w/ hardware trigger
    - TPC signal: ~100 μs, physical event rate 50 kHz, TPC signal overlap
  - Very basic hardware+ more effective software trigger





Roadmap of electronics and data processing

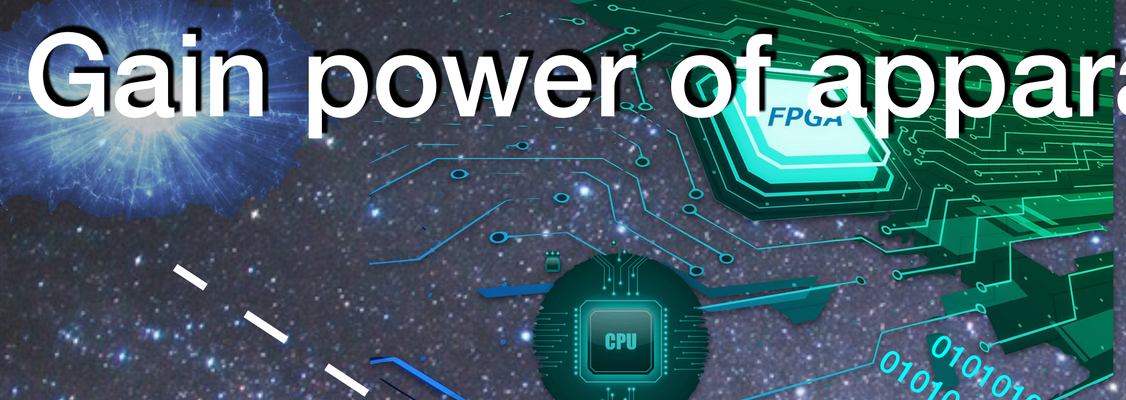
Exp.	Run time	Data (PB)	Total
BESIII	2008-2028	0.5	10
STCF	-	300-500	-
CEPC	-	1.5-3(H) 500-50000 (Z)	-

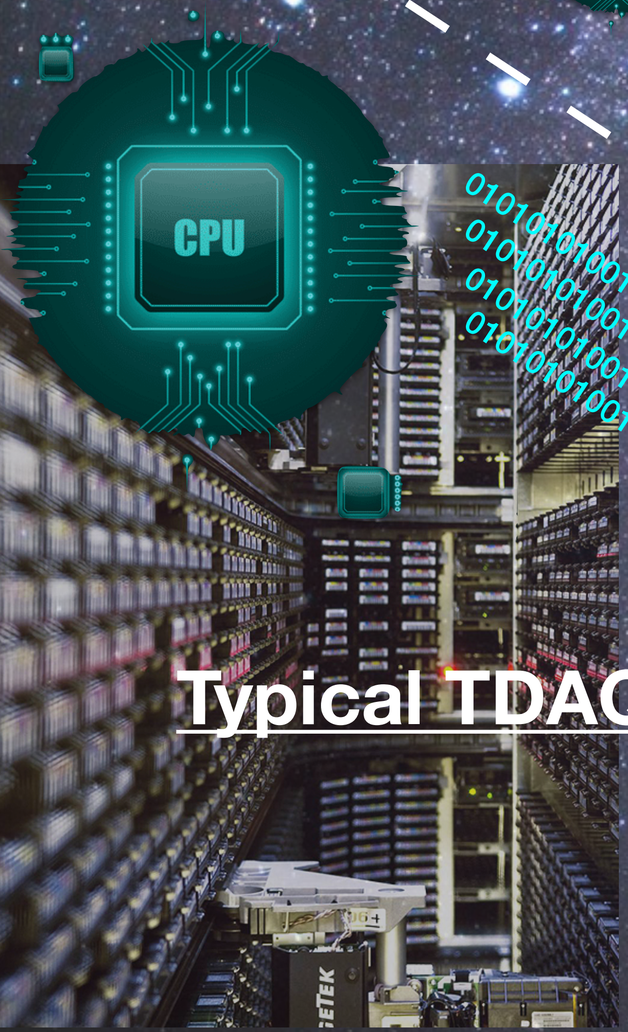


ECFA detector R&D

Gain power of apparatus with data acceleration

- Continues readout (less-hardware filtering)
- Powered by hardware acceleration
- Heterogeneous computing

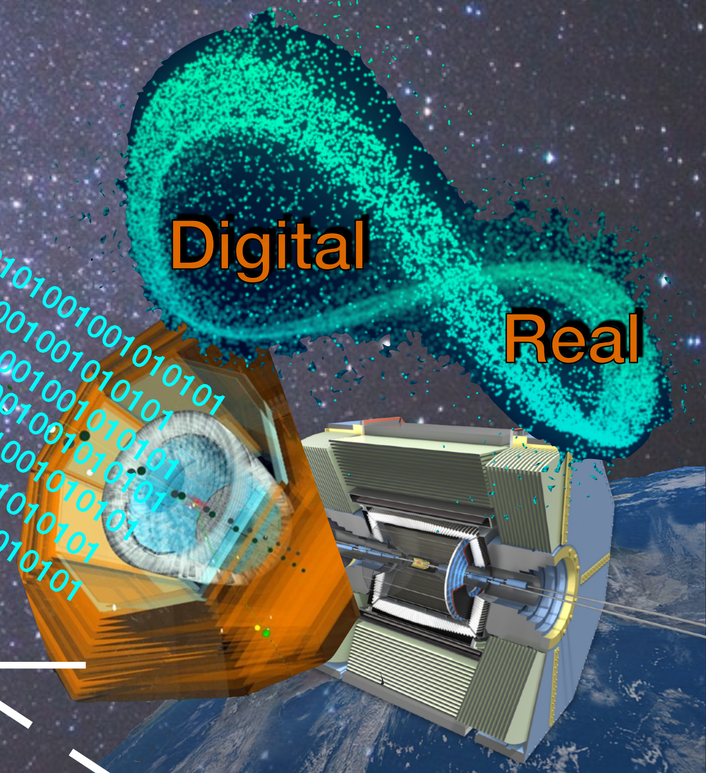




Typical TDAQ system

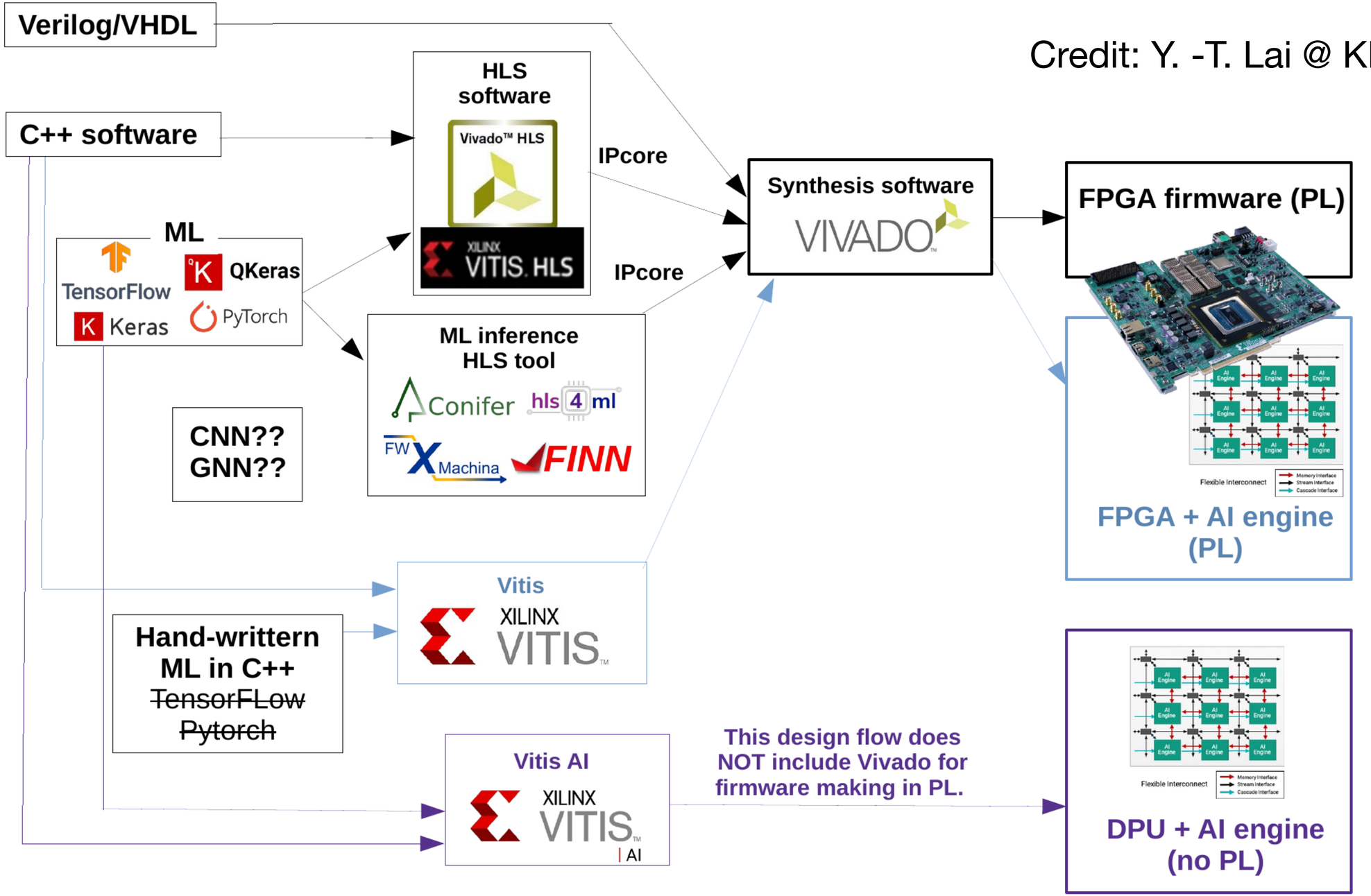
Trigger-less data readout system

Decisions



Roadmap of machine learning on “FPGA”

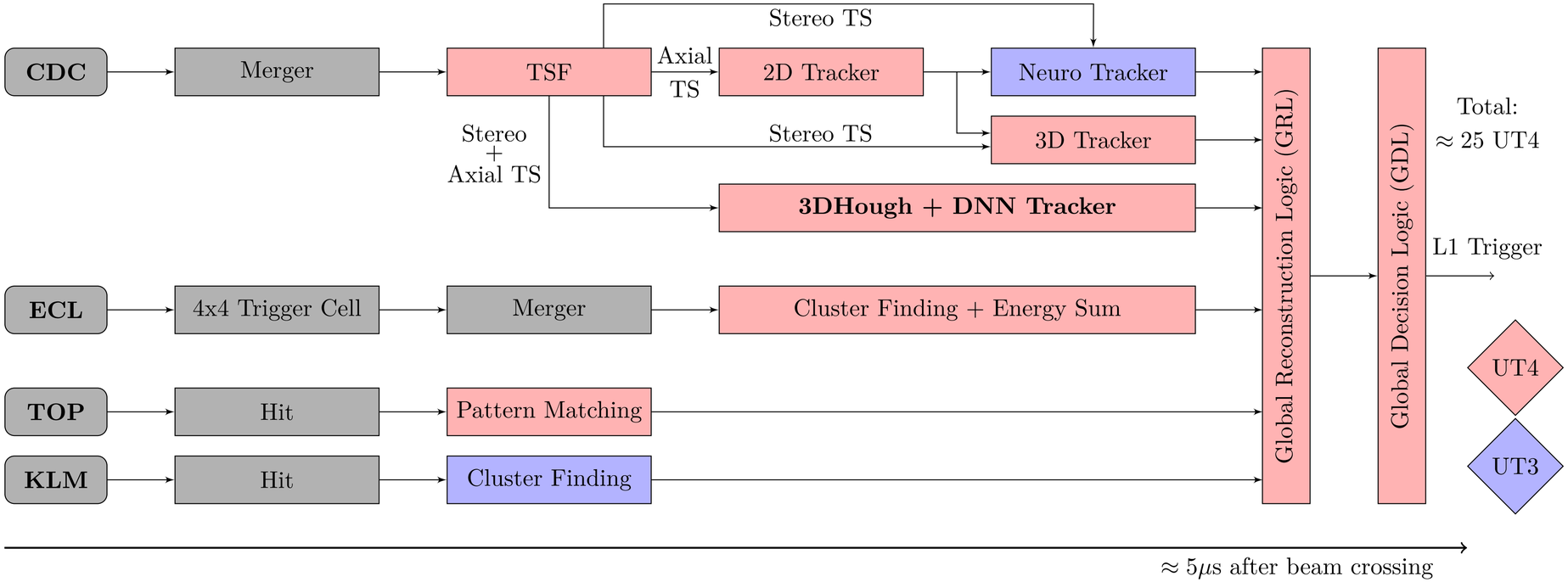
Credit: Y. -T. Lai @ KEK



Belle II trigger system

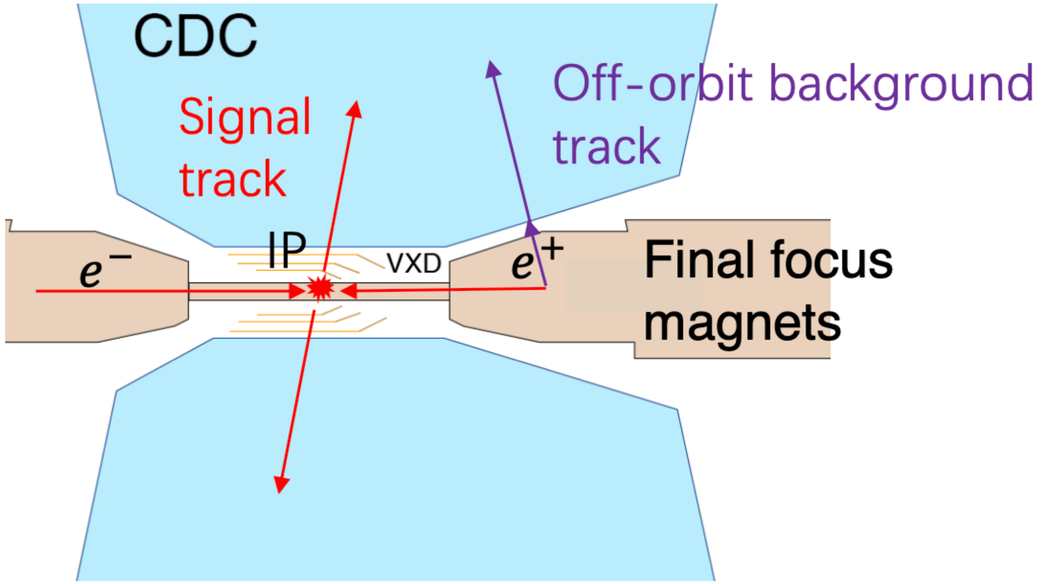
- Max. trigger rate: 30 kHz @ 6 x 10^{35} cm^{-2} s^{-1}
  - Physics trigger ~15 kHz
- Latency limit: ~5 usec (SVD APV25 buffer structure)
  - A fixed latency of about 4.4 usec
- Event timing resolution: 10 nsec
- CDC, ECL: main triggers for tracks and clusters, KLM: trigger muon, TOP: event timing

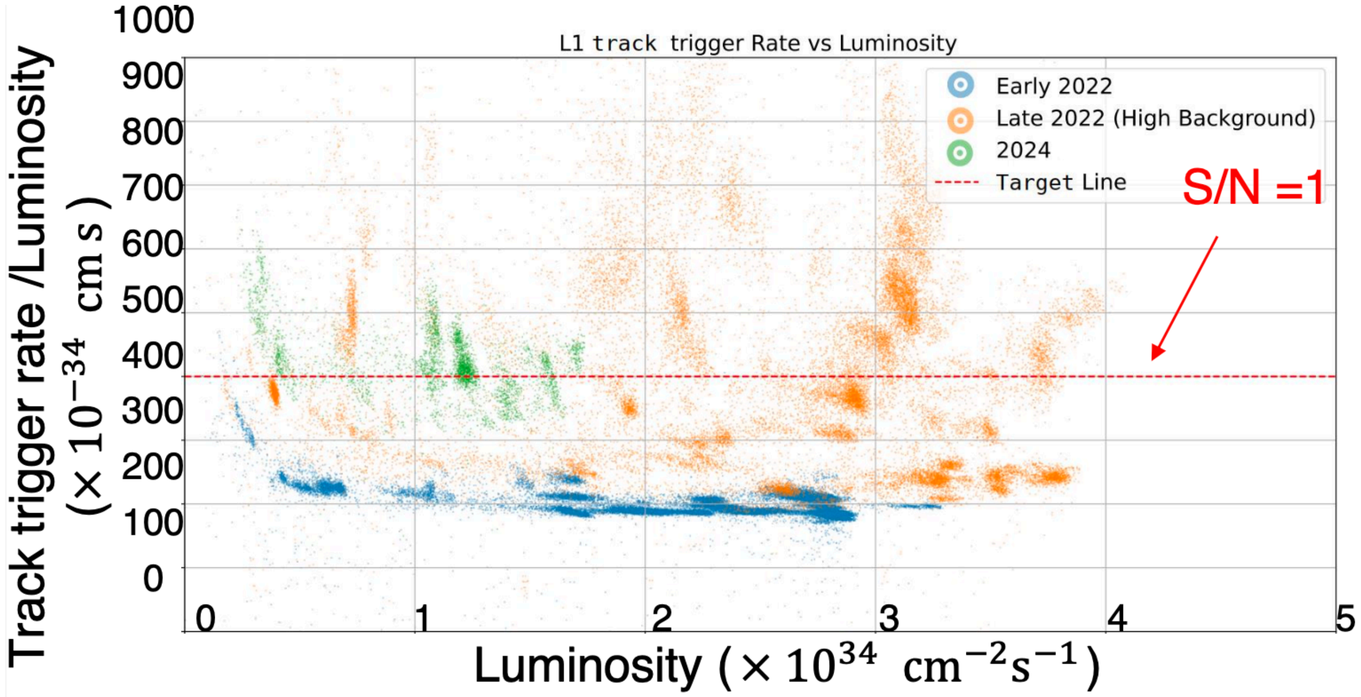
- Challenges:
  - low multiplicity trigger vs. background
  - High track trigger vs. crosstalk
  - Drawback of track trigger at endcap
  - Latency budget vs. transmission and logics
  - …



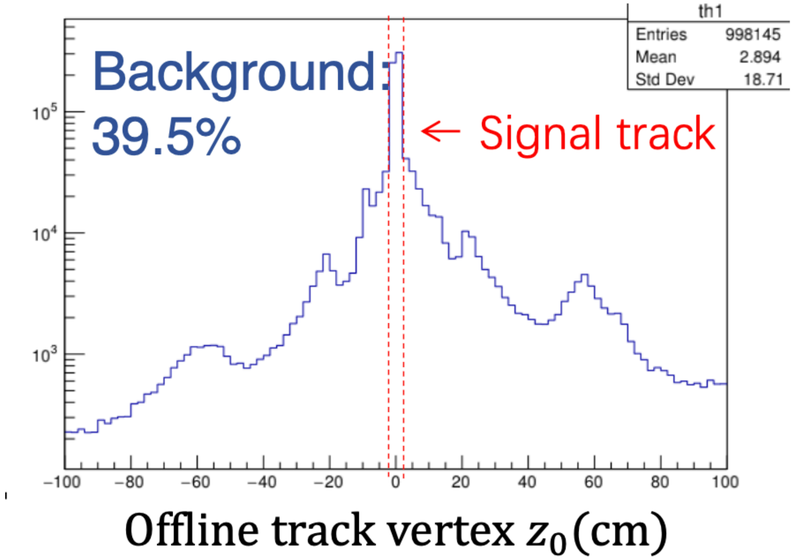
Motivation of Neural Network for L1 Track trigger

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

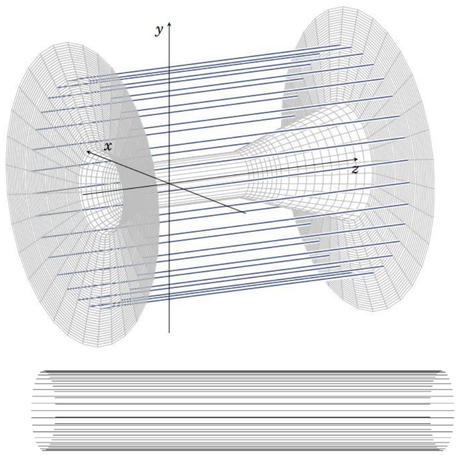




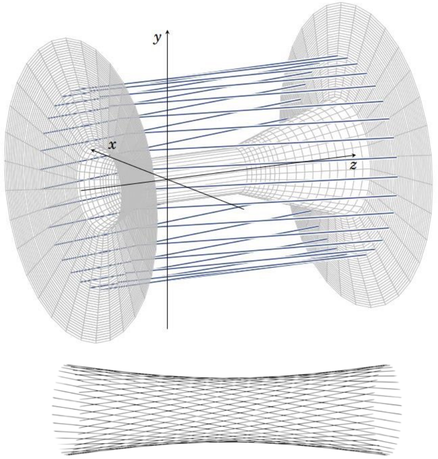
Tracks z_0 distribution after trigger



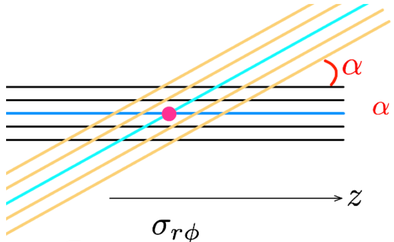
Basics of L1 CDC trigger



Axial wire



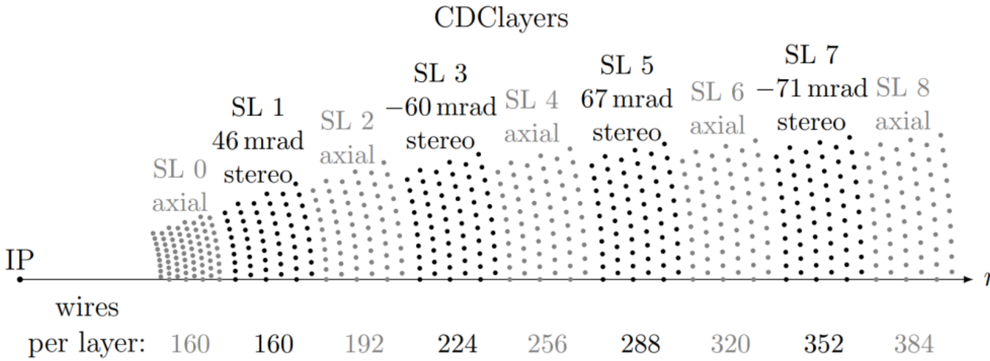
Stereo wire

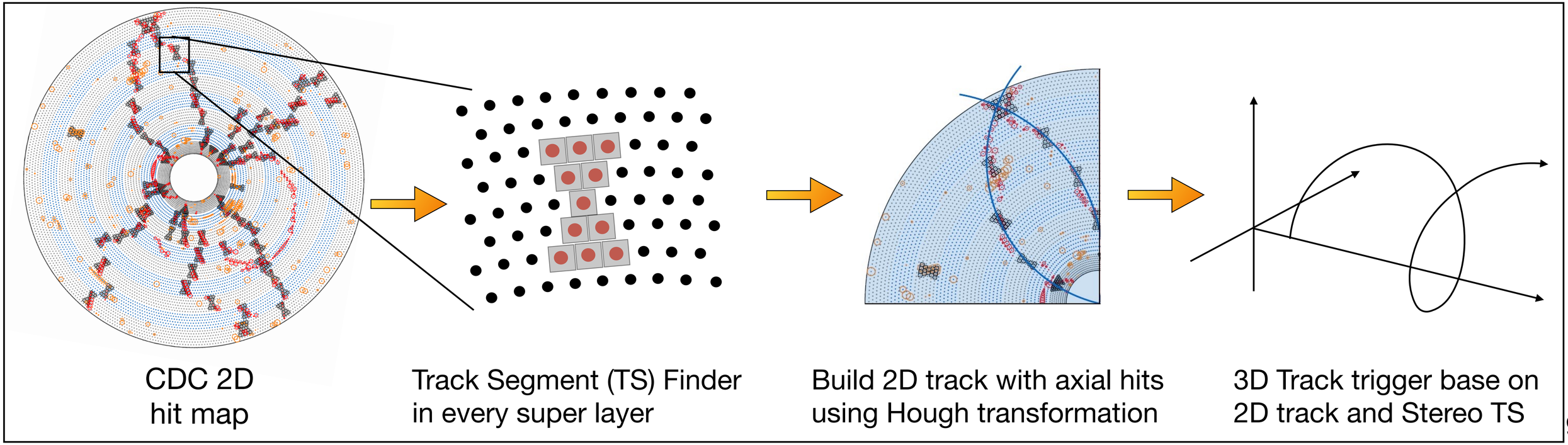


$\alpha = 60 \sim 80\text{mrad}$ (BelleII)

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

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





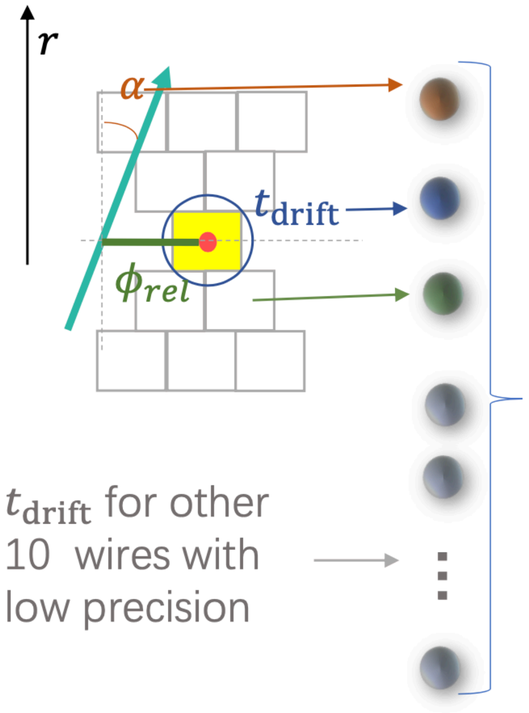
CDC 2D hit map

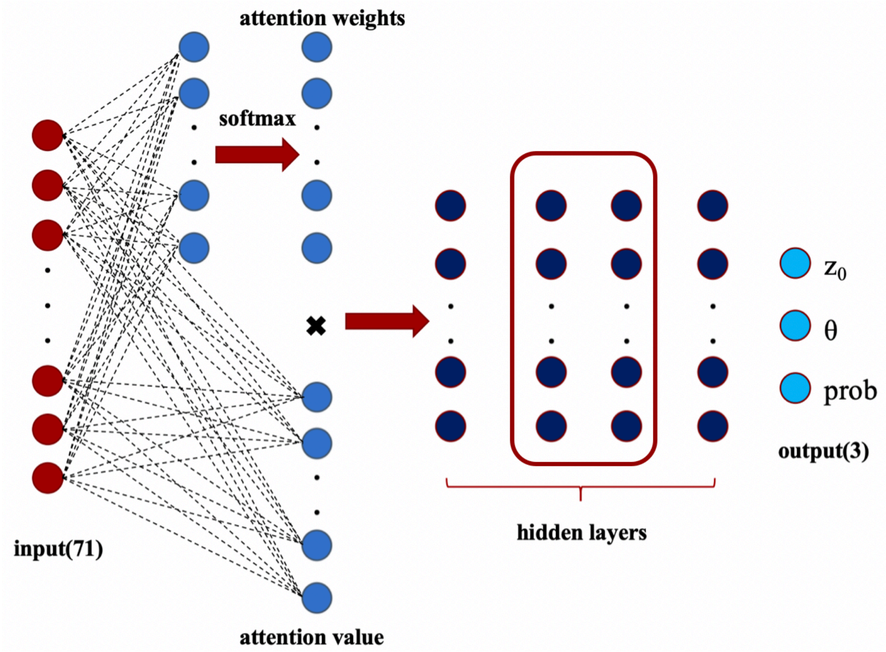
Track Segment (TS) Finder in every super layer

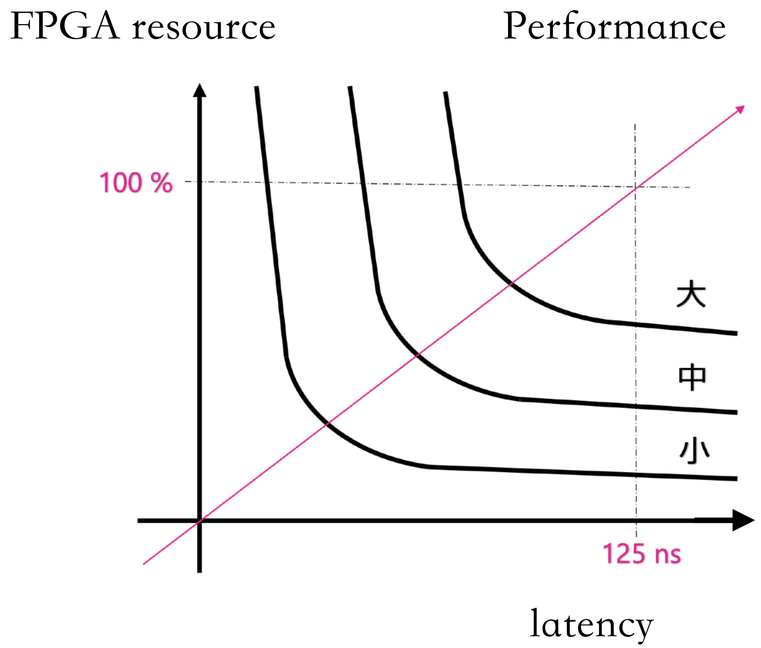
Build 2D track with axial hits using Hough transformation

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

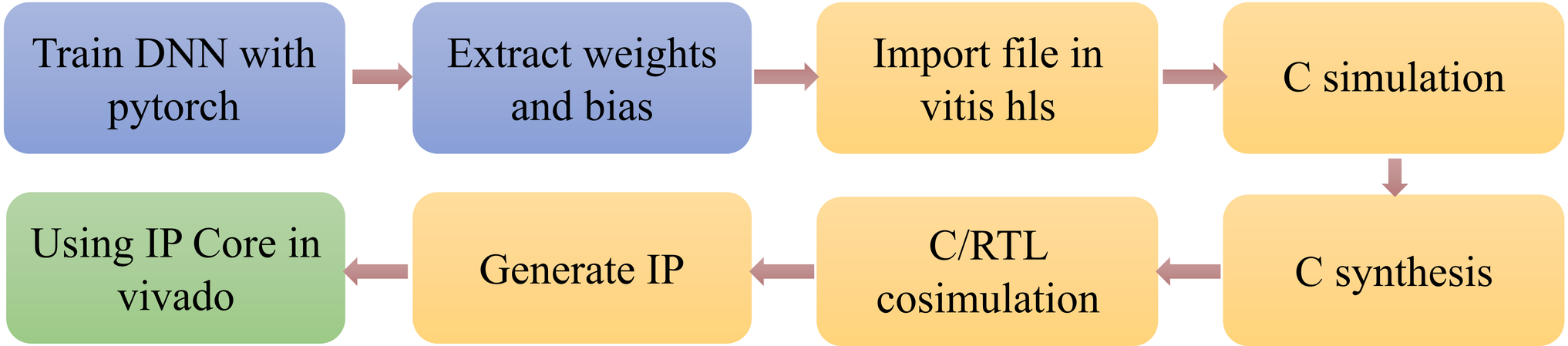
VIVADO

- Start fitter

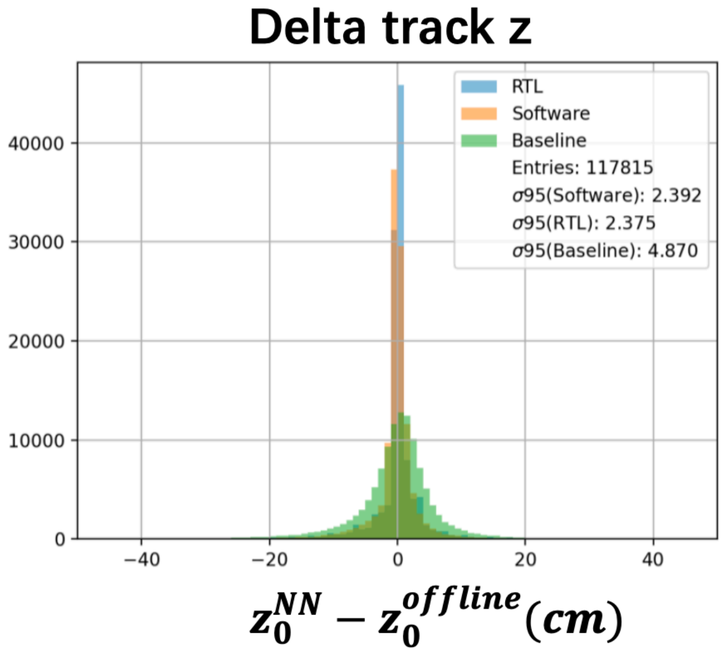
- Evaluation

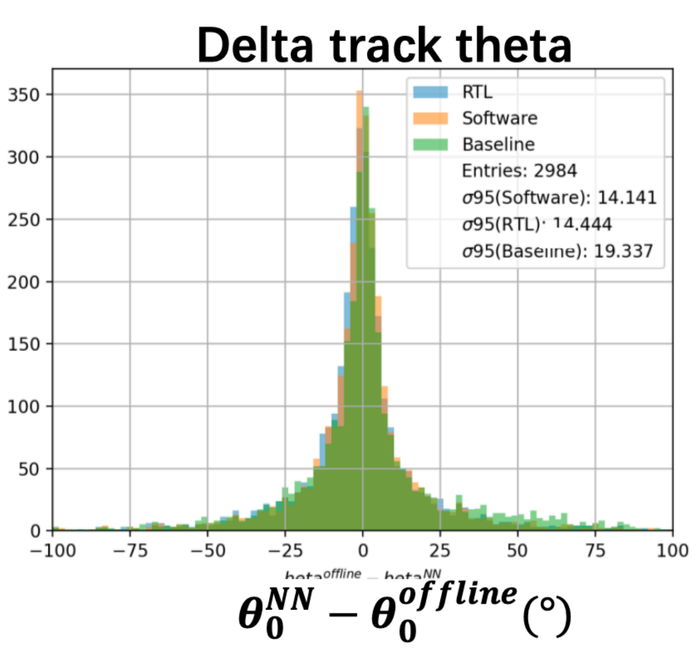
Belle II UT4

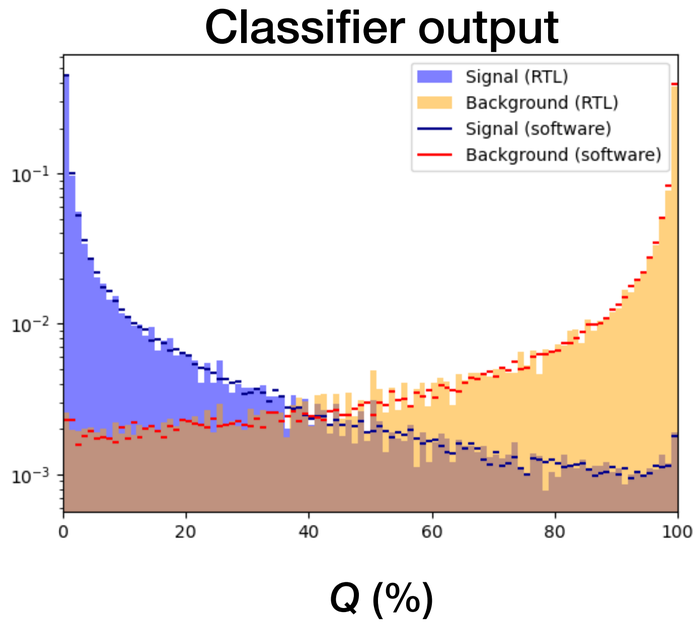
Xilinx UltraScale
XCVU080, XCVU160
25 Gbps with 64B/66B



Performance of DNN algorithm

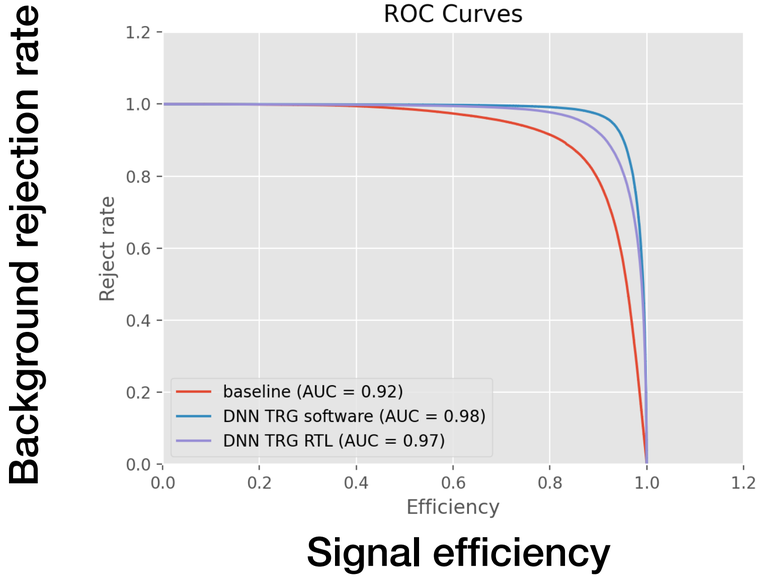






- Latency : 76 clock = **592.8 ns** ;require: < 600ns
- FPGA resource (UT4: Virtex UltraScale XCVU160) usage:
  - DSP: ~70%, LUT: ~50%, others <30%
- AUC do not get large drop comparing RTL and software simulation
- At signal efficiency ~95%
  - Background rejection rate ~85%

- **DNN trigger with HARDWARE under commissioning, close to operate**

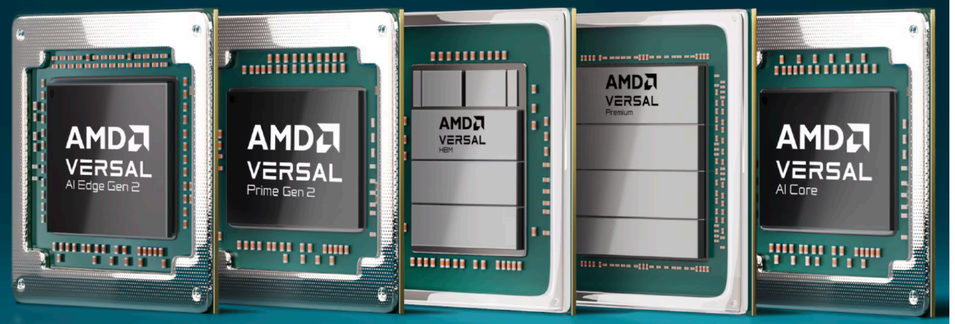


AMD Versal projects

AMD Versal™ Adaptive SoCs

For any application from cloud to edge

New 2nd Gen Versal Portfolio



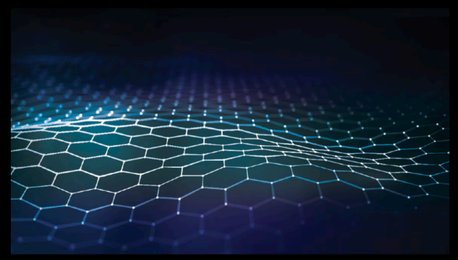
Overview Portfolio Developers Resources

Overview

curved gradient divider

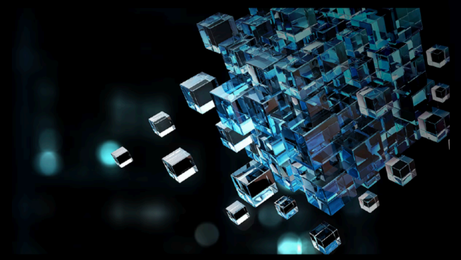
Heterogeneous Acceleration

Highly integrated, multicore compute platform that can adapt with evolving and diverse algorithms.



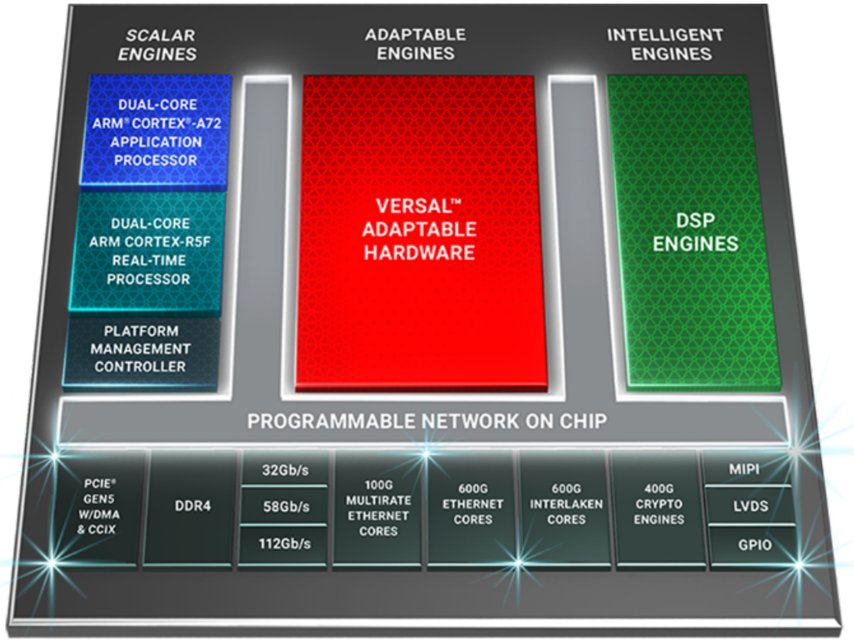
Any Application

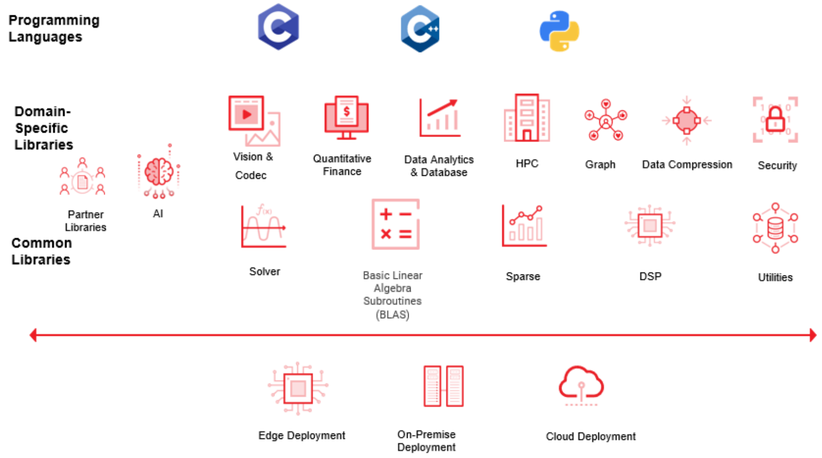
Dynamically customizable at hardware and software levels to fit a wide range of applications and workloads.



Any Developer

Architected around a programmable network on chip (NoC), Versal adaptive SoCs are programmable by software developers and hardware programmers alike.





DNN implementation on Versal ACAP

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

UG1079

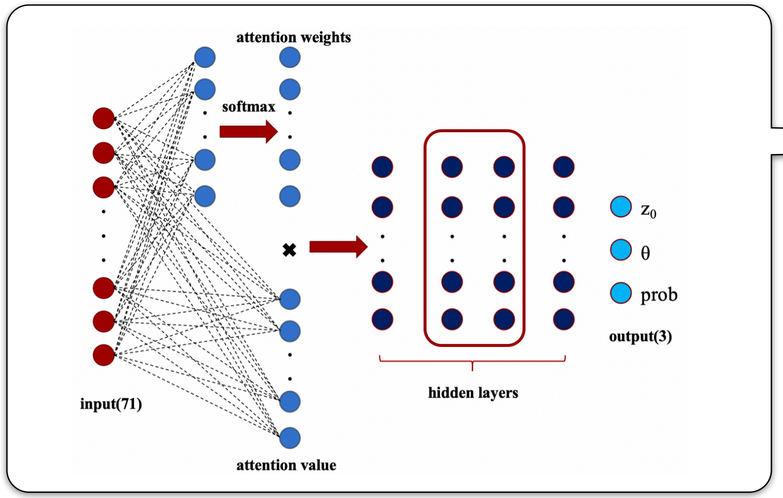


Figure 2: **AI Engine Array**

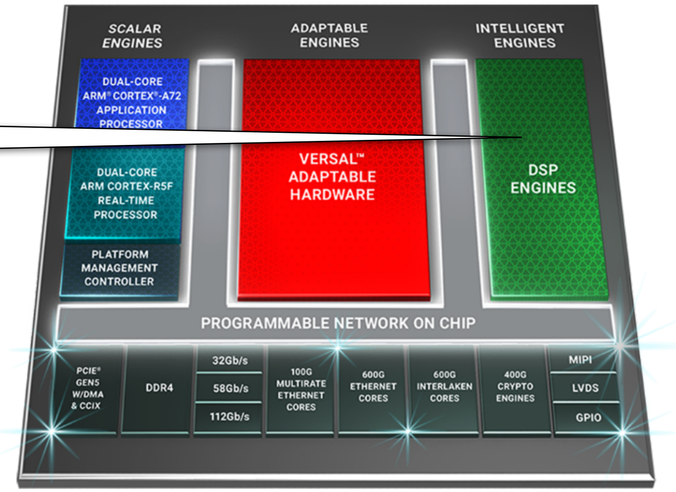
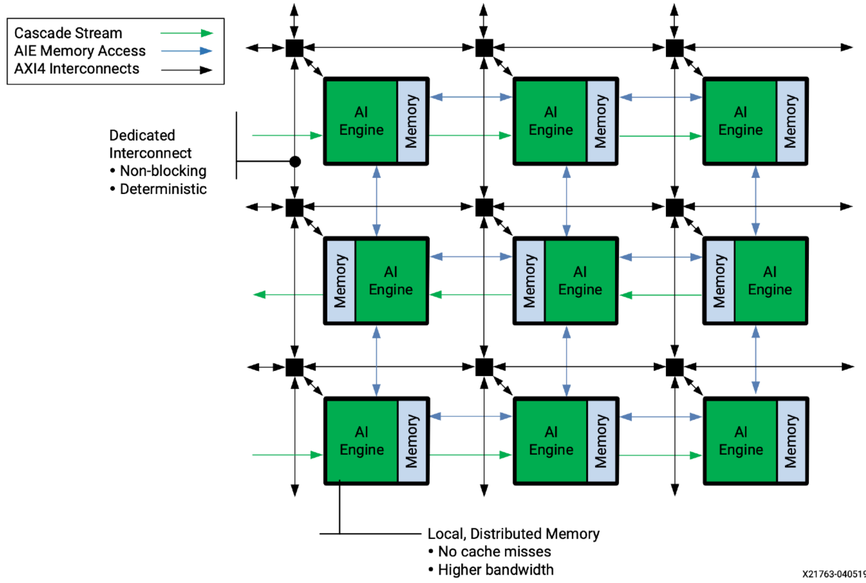
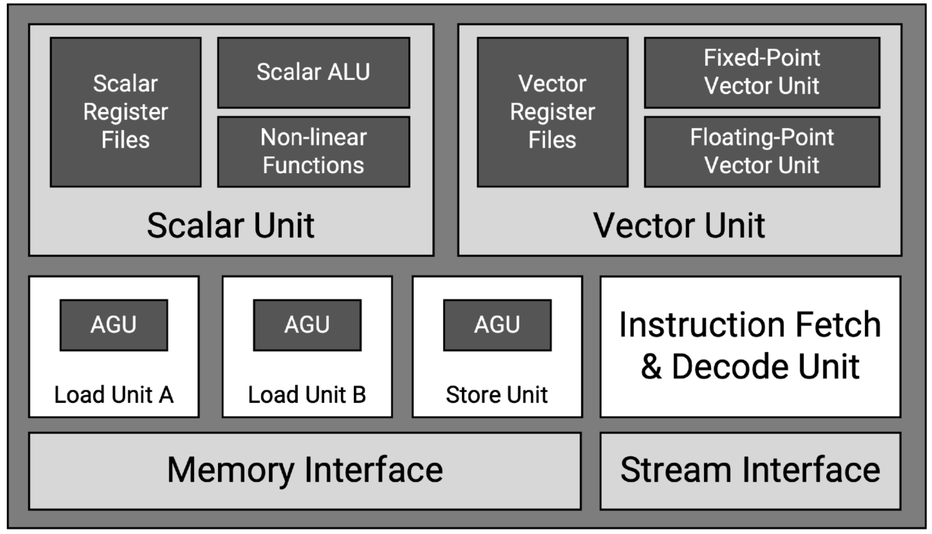
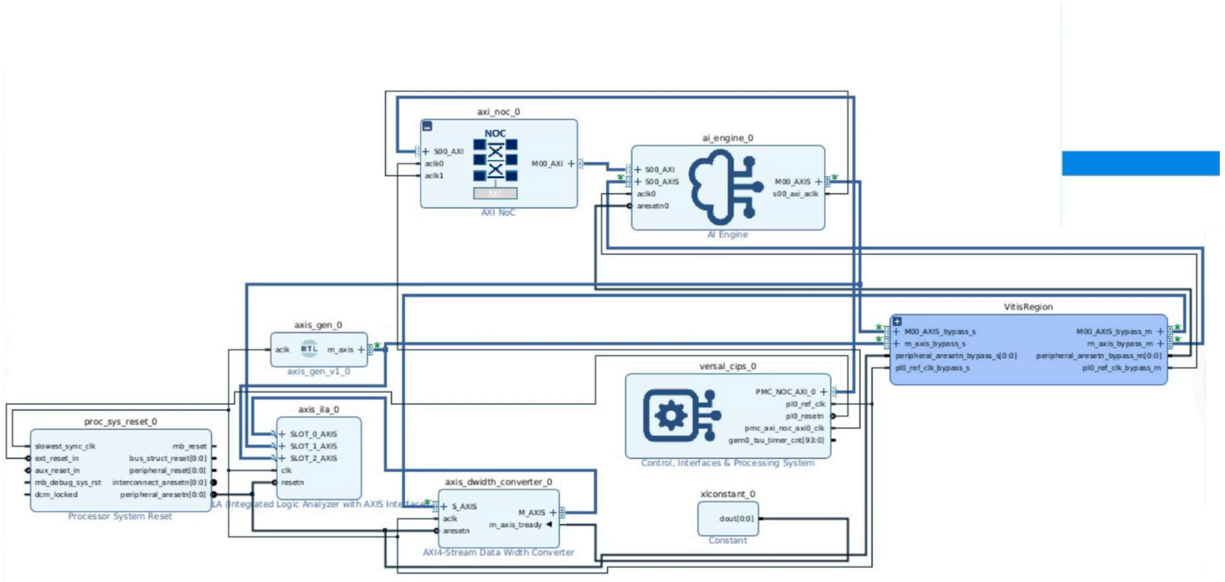


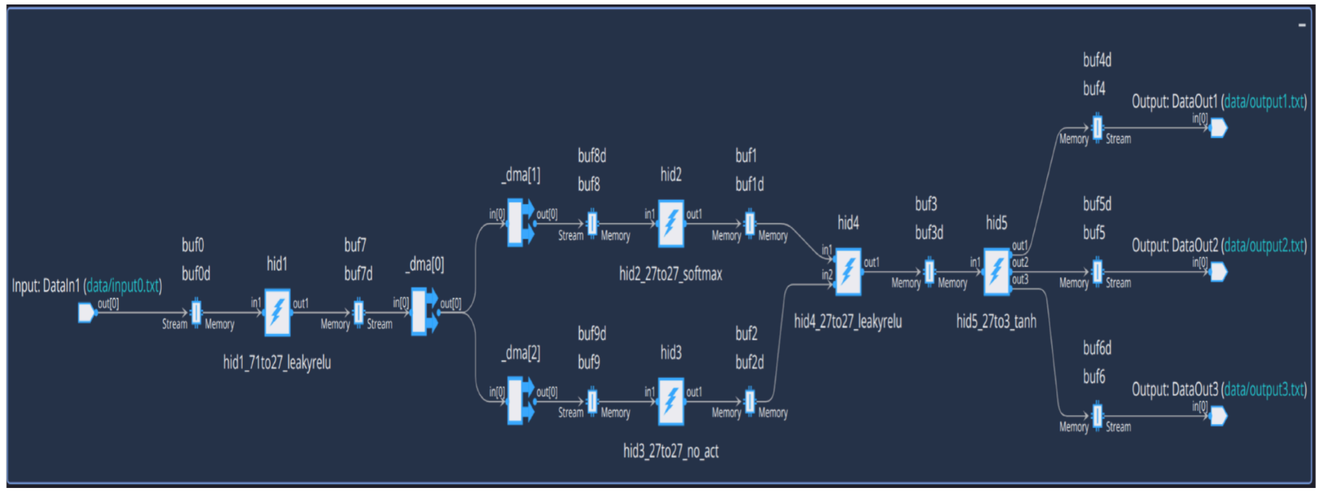
Figure 4: **AI Engine**





DNN acceleration on Versal ACAP





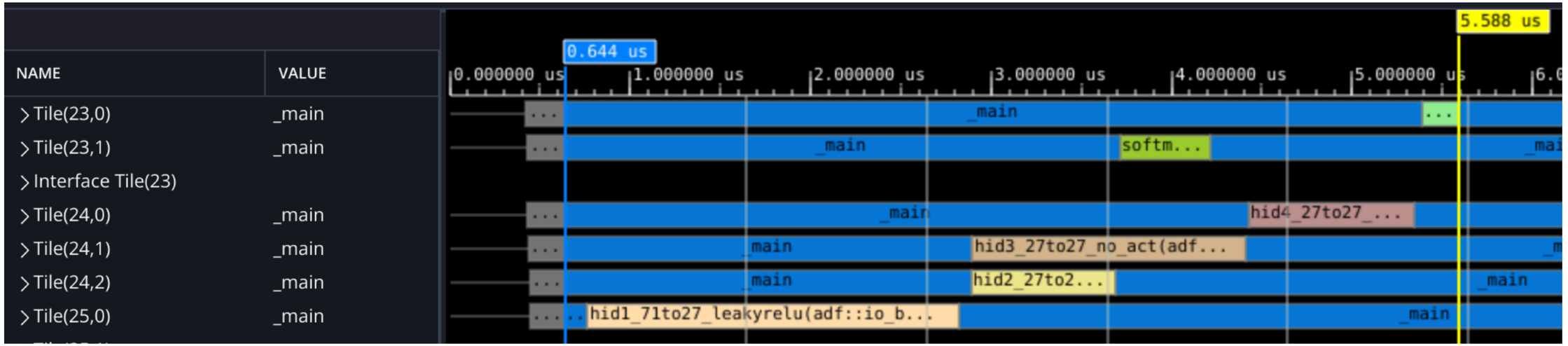
- DNN implementation:
  - Model on a “graph”
  - Dense layer on a “kernel”
- AI engine: C++ based coding on Vitis
  - AI engine libraries
  - AI engine specific functions
  - Scaler, Vector engines, pipelining, etc.

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Input nodes	71	27	27	27	27
Output nodes	27	27	27	27	3
Active Func.	LeakyReLU	Softmax	—	LeakyReRU	Tanh

AI Engine Resource Utilization

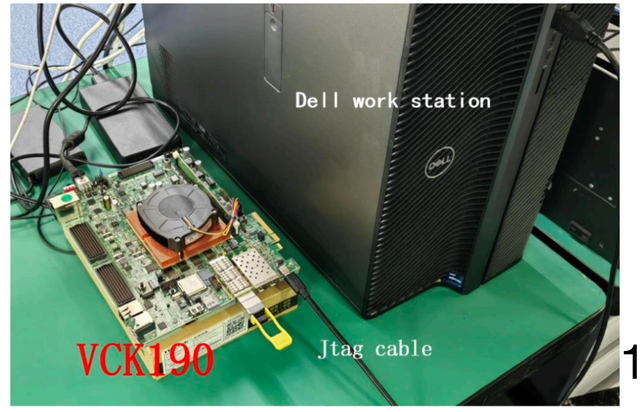
Tiles used for AI Engine Kernels:	5 of 400 (1.25 %)
Tiles used for Buffers:	7 of 400 (1.75 %)
Tiles used for Stream Interconnect:	8 of 450 (1.78 %)
DMA FIFO Buffers:	0
Interface Channels used for ADF Input/Output:	4 (PLIO: 4)
Interface Channels used for Trace data:	0

Latency optimization on Versal ACAP

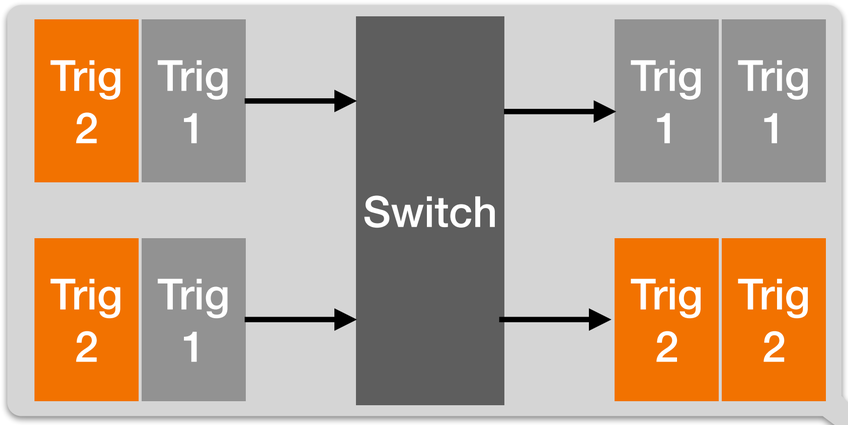


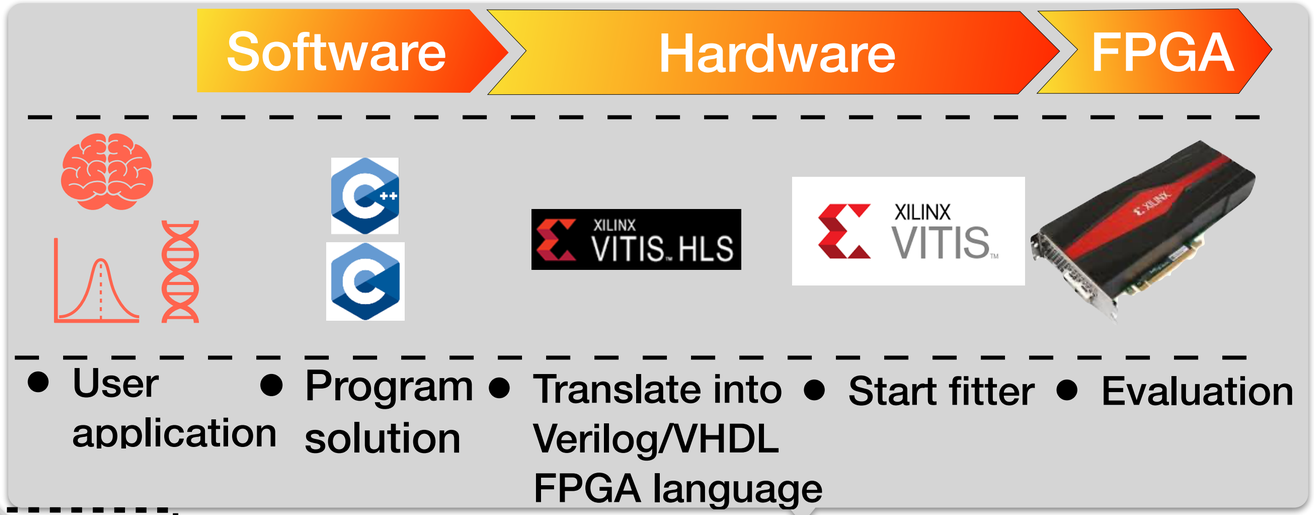
	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Total
Input nodes	71	27	27	27	27	—
Output nodes	27	27	27	27	3	—
Active Func.	LeakyReLU	Softmax	—	LeakyReRU	Tanh	—
Ver.0 latency	~12us	~66us	~1.5us	~5.5us	~9.9us	~86us
Ver.1 latency	~2.1us	~1.3us	~1.5us	0.9us	~0.2us	~5us
Ver.2 latency	~0.48 us	~0.93us	~0.33us	~0.40us	~0.10us	~2.1us

Total **305 clk cycles** one instance. Clock period **10ns**.
Latency running on Versal ACAP is **3.05us**



Real-time and AI integrated High Level Trigger





Sorting

Optional

Frontend Electrics

Detector1

Frontend Electrics

Detector2

Data merger

Data merger

Cross-point switch

Switch

Tracking Readout system

PID Readout system

Real-time buffer

Real-time buffer

Fast Reco (Level1) CPU + FPGA acceleration

Detector Readout system

Real-time buffer

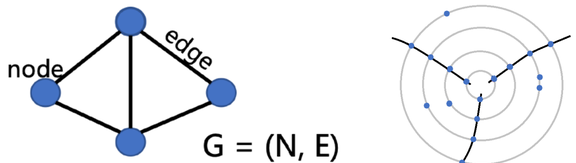
HLT

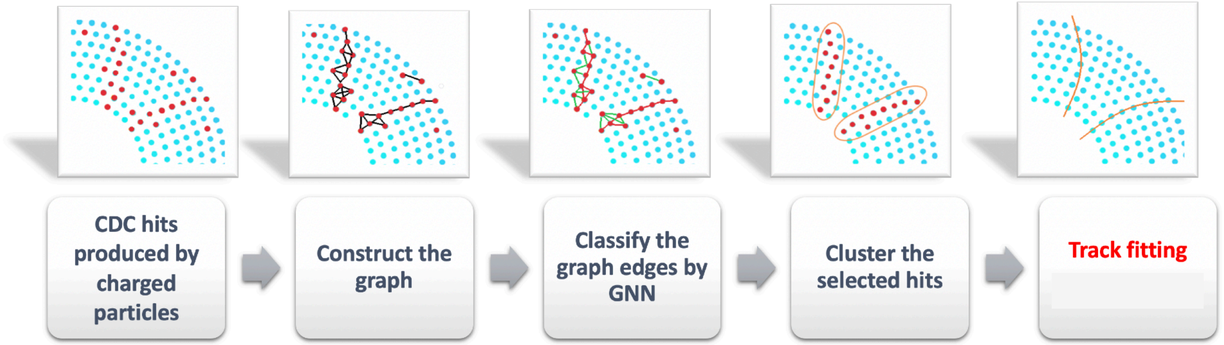
Storage

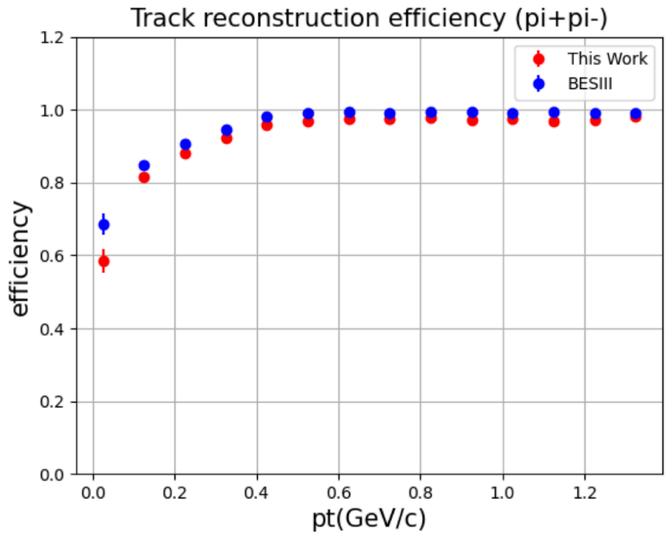
GNN for CDC track background filtering

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

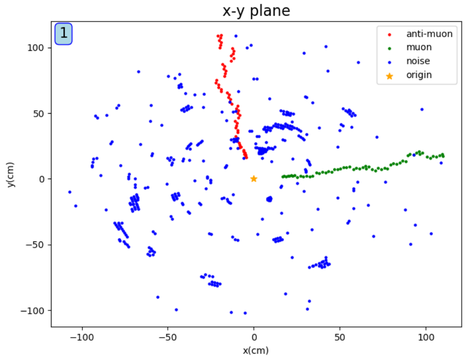
Noise filtering → Clustering → Fitting

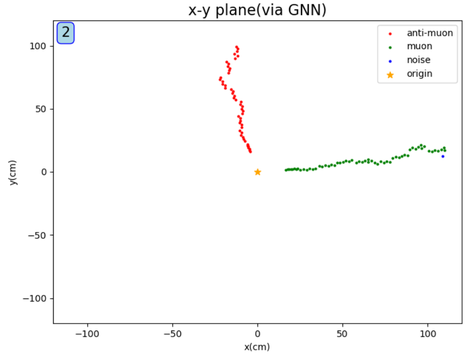






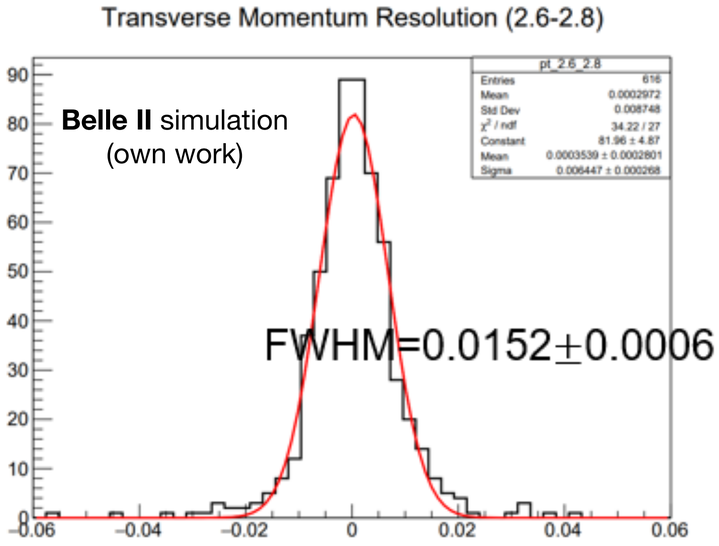
Belle II simulation (own work)





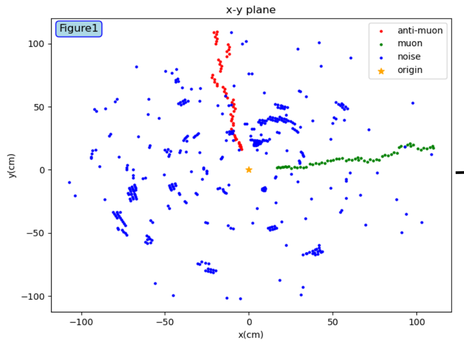
μ+ μ- (particle gun)

GNN noise filtering

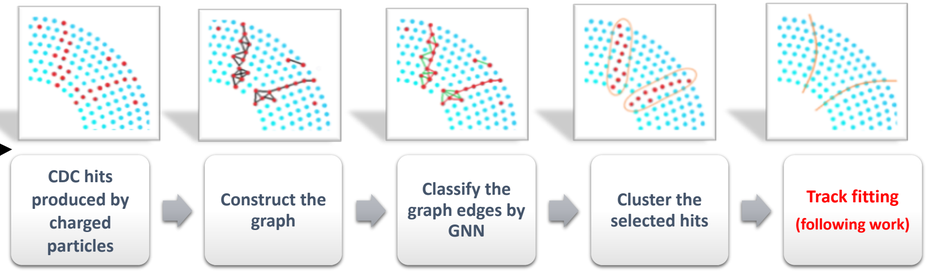


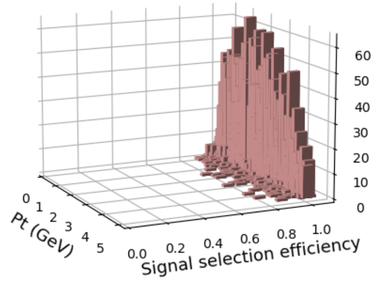
Acceleration on Versal ACAP platform

μ+ μ- (particle gun)



Belle II simulation (own work)

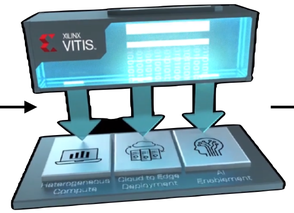




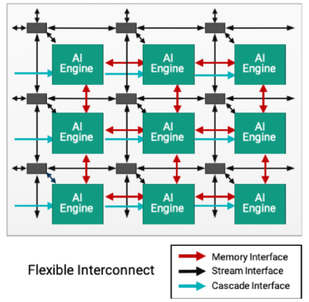
Hit selection efficiency: 98.4%
Hit selection purity : 97.9%

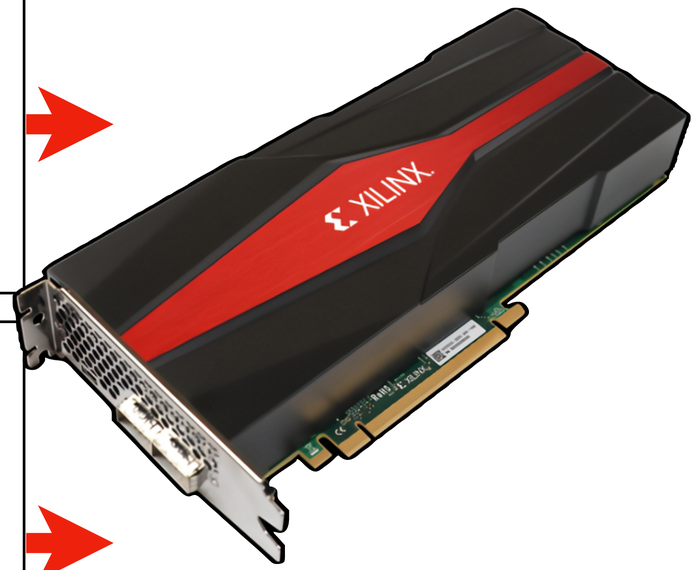
AI engine

C++ software



VIVADO™





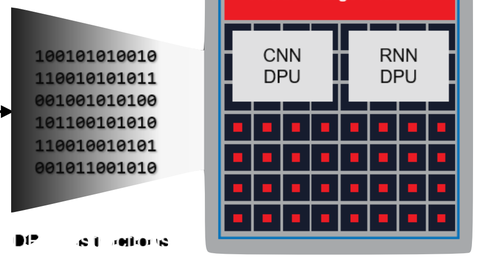
DPU

ML

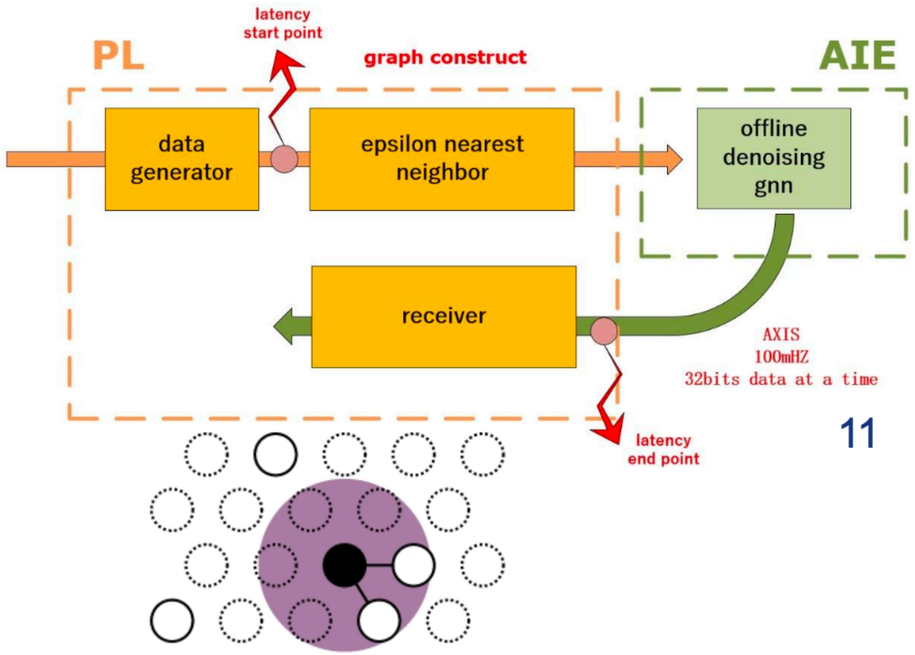
TensorFlow
QKeras
Keras
PyTorch

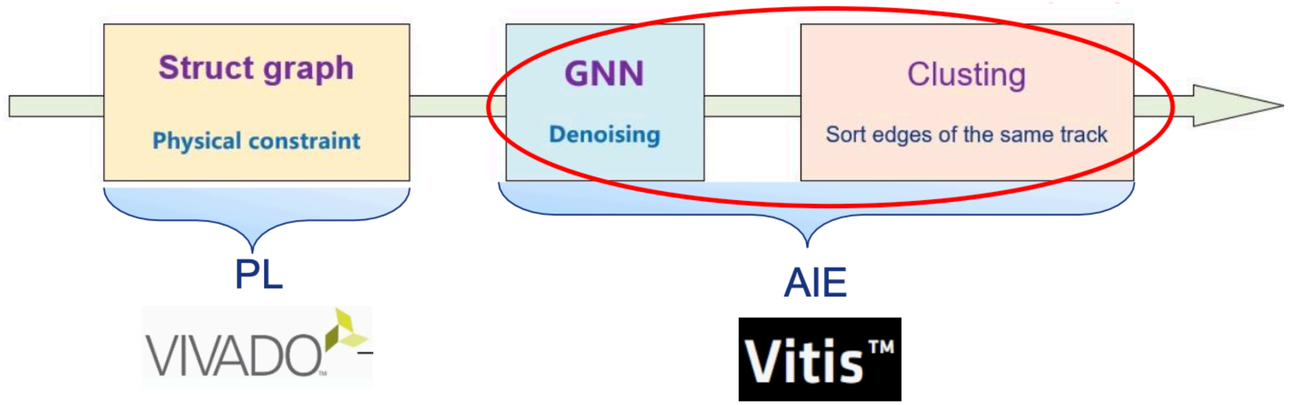
C++ software

Vitis AI
XILINX VITIS AI



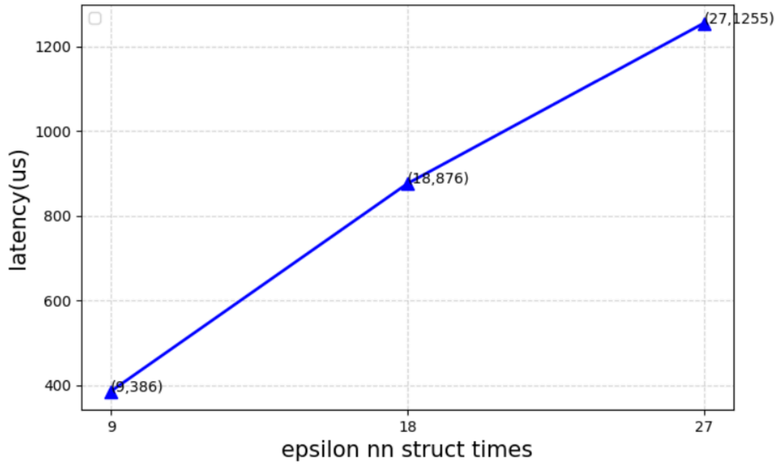
Demo of GNN implementation on Versal ACAP

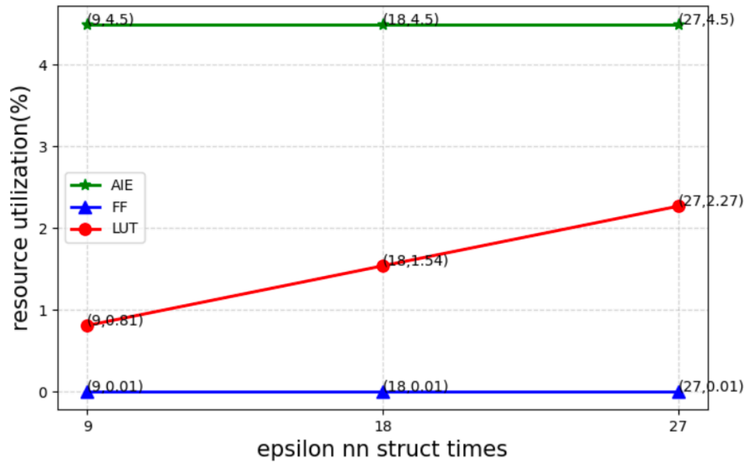




εNN times	Feature matrix [εNN, feature]	Adjacent matrix (*2) [node,edge]
9	[9, 12]	[9, 5]
18	[18, 12]	[18, 5]
27	[27, 12]	[27, 5]

Latency of simple implementation with only 1 iteration of GNN algorithm on Versal is in order of **~ms**

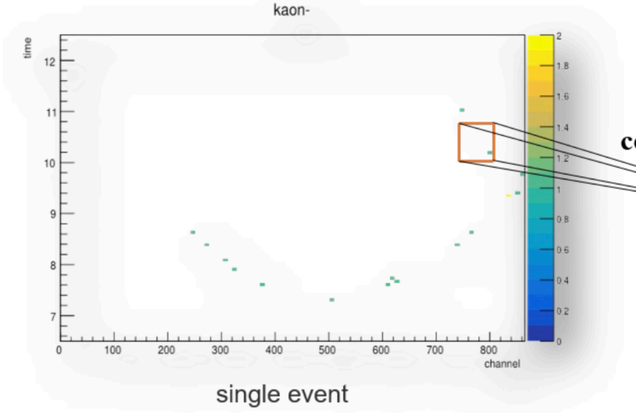


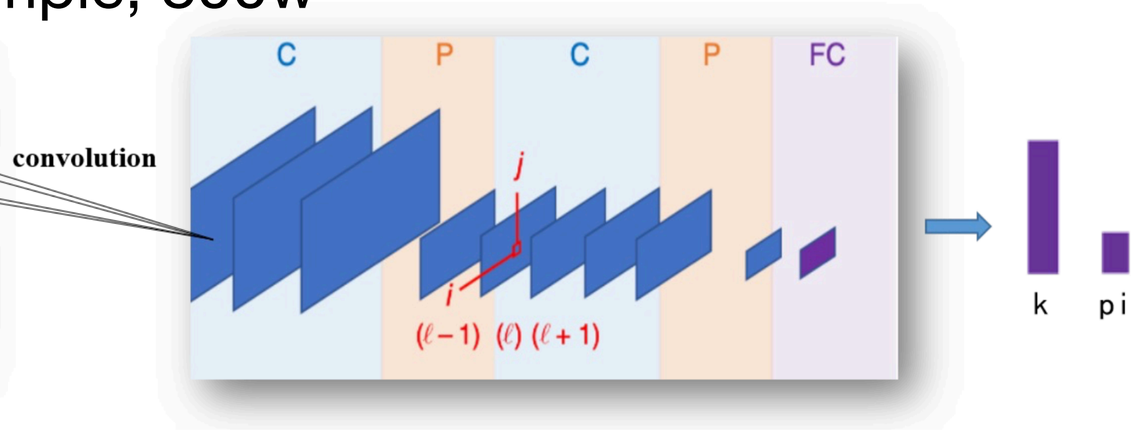


CNN algorithm for PID

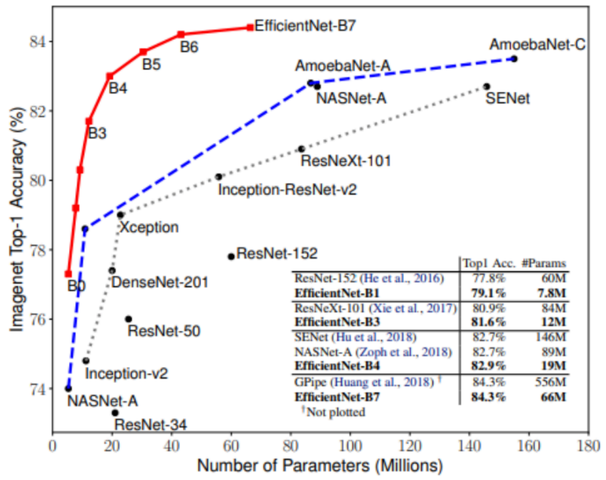
- DTOF as a PID subdetector of STCF
- CNN algorithm developed for Kaon/pion identification
- Kaon/Pion MC simple, 800w

Z. Yao et al.@SDU





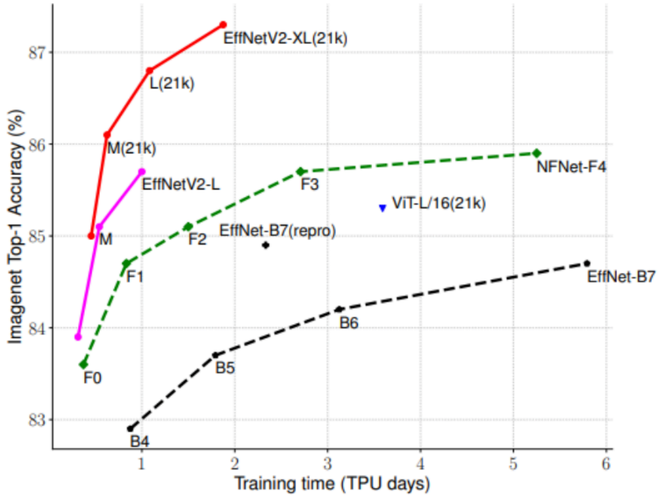
EfficientNetV1

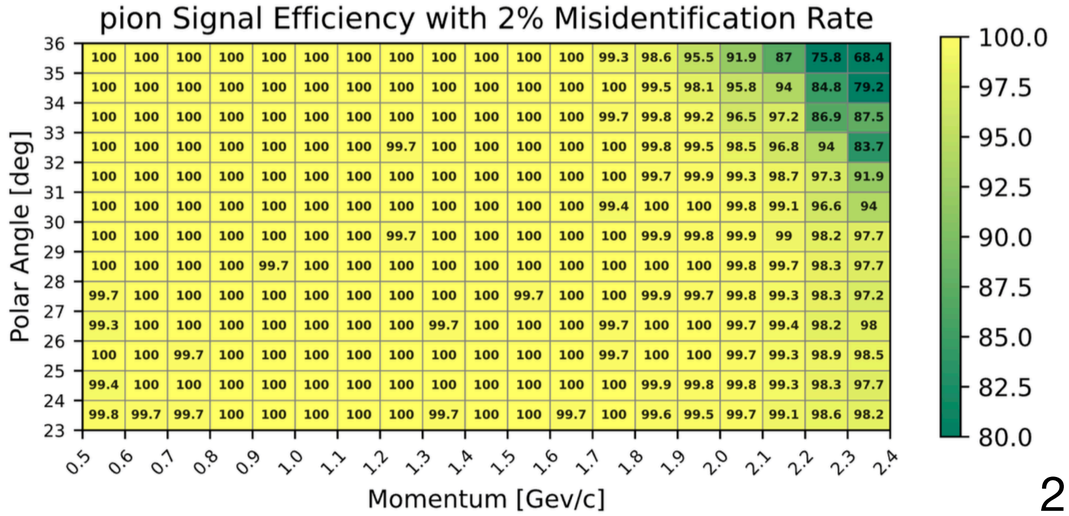


	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

EfficientNetV2





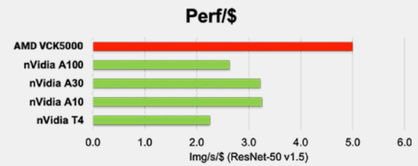
Implementation on DPU with Vitis AI

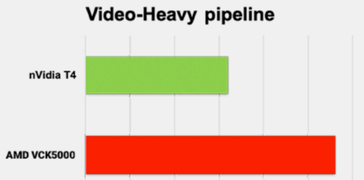
AI Inference Development

If you are an AI developer, bring your TensorFlow and PyTorch trained models to directly infer on Versal using Mipsology Zebra and build, configure, and deploy computer vision applications on FPGA platforms with Aupera Video Machine Learning Streaming Server solution.

Key Features

Explore partner solutions and articles, and learn about the key features for AI Inference Development with the VCK5000





2x TCO Reduction vs Mainstream nVidia GPUs

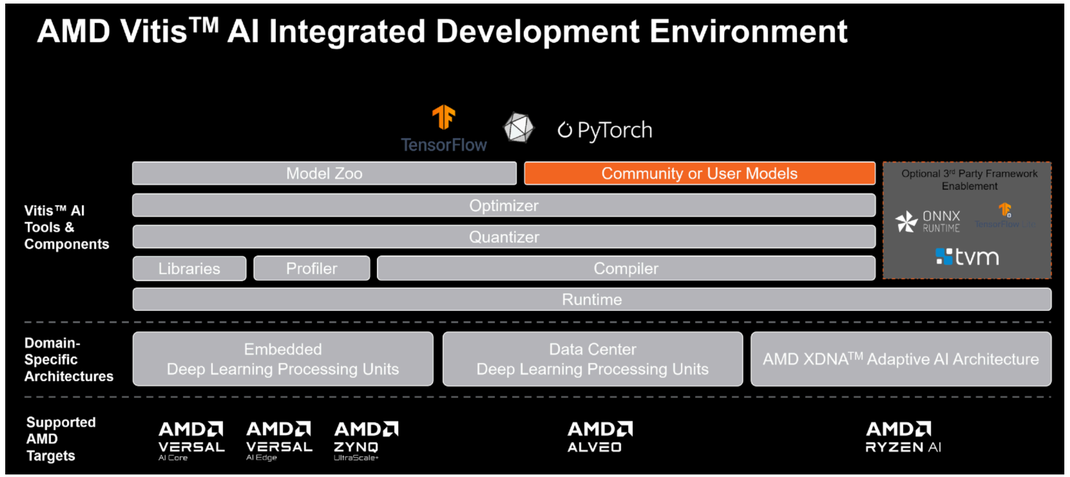
- 2x perf/w and perf/$ compared to Nvidia Ampere with standard MLPerf Models
- Achieves 90% compute efficiency
- Consume less than 100W at card level

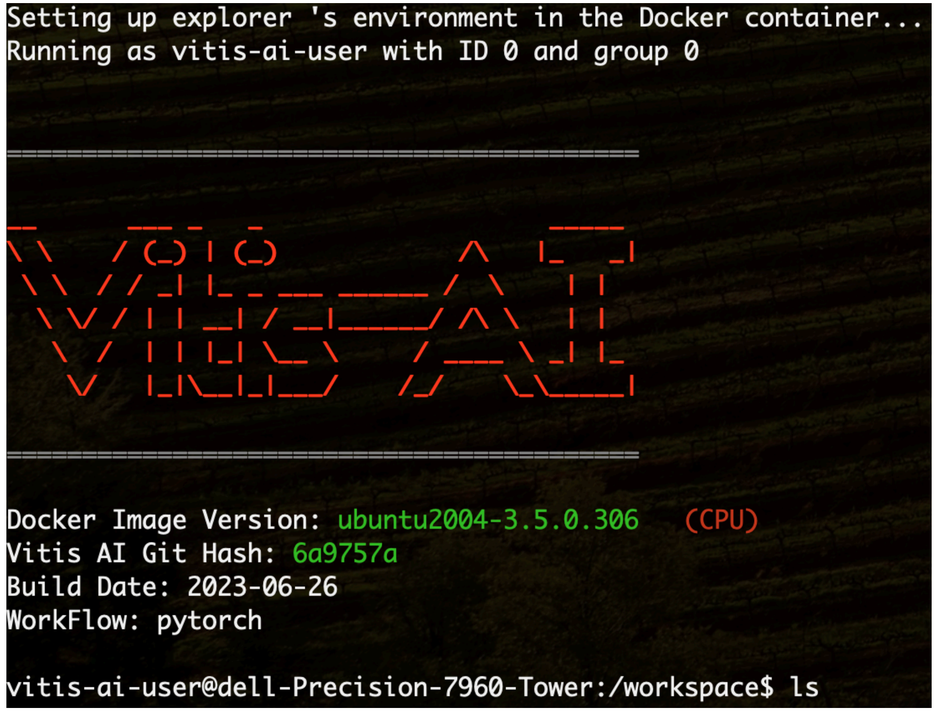
2x End-to-End Video Analytics Throughput vs nVidia GPUs

- Full pipeline from H.264 decode to computer vision to up to 10 AI models
- Video decode and CV run on x86 CPU or discrete U30 Alveo card
- Plug-in based pipeline composition from FFmpeg / Gstreamer

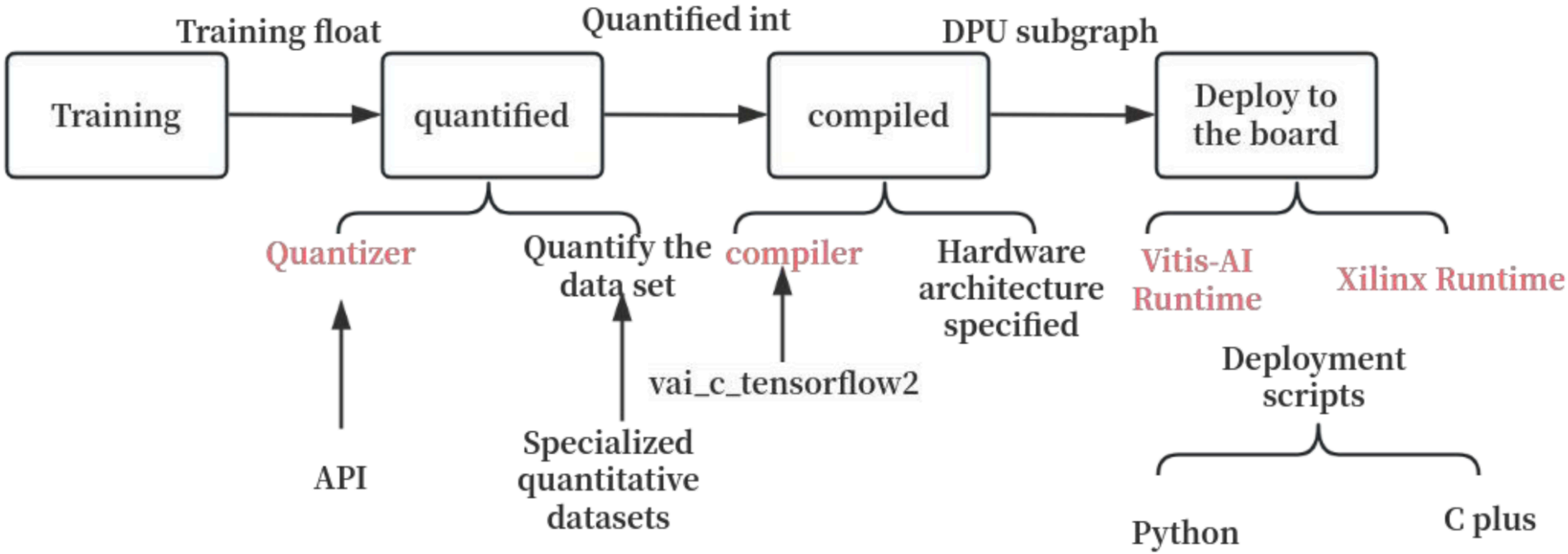
Easy to Use with Familiar Frameworks

- Easy-to-use software flow for any CPU & GPU users, no hardware programming required
- Run inference from Tensorflow framework directly on board
- State-of-the-art model supported with mainstream frameworks PyTorch, TensorFlow, TensoFlow 2 and Caffe





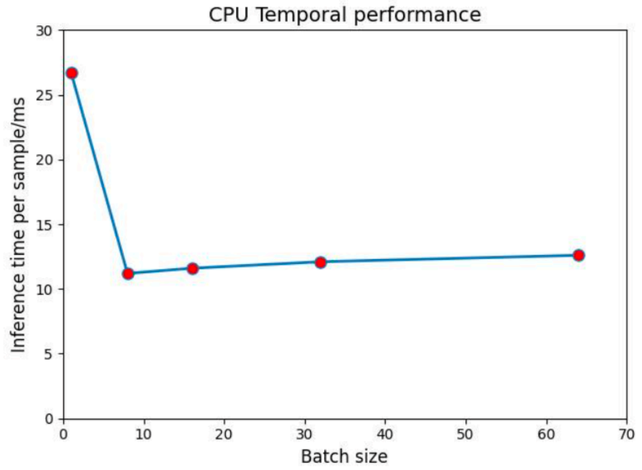
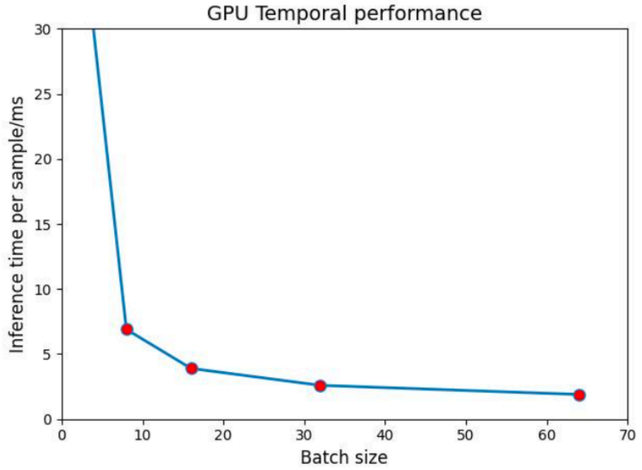
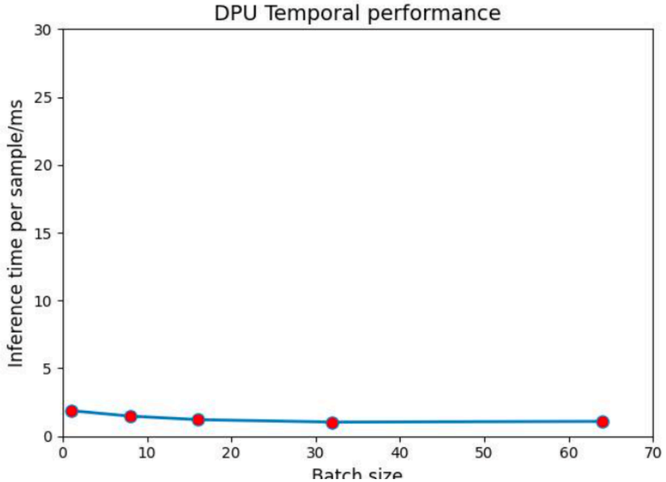
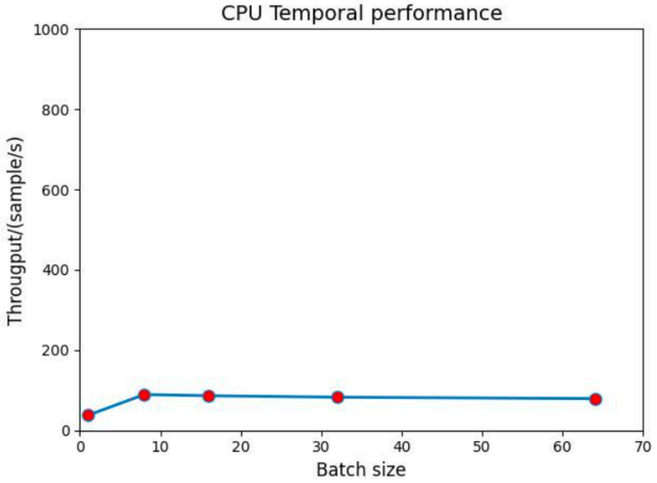
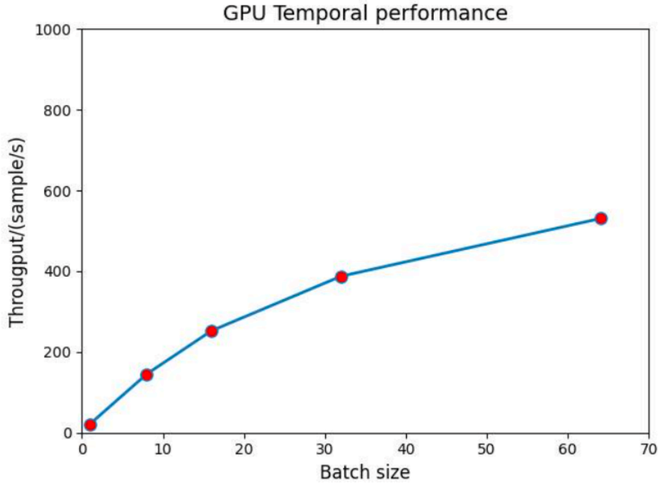
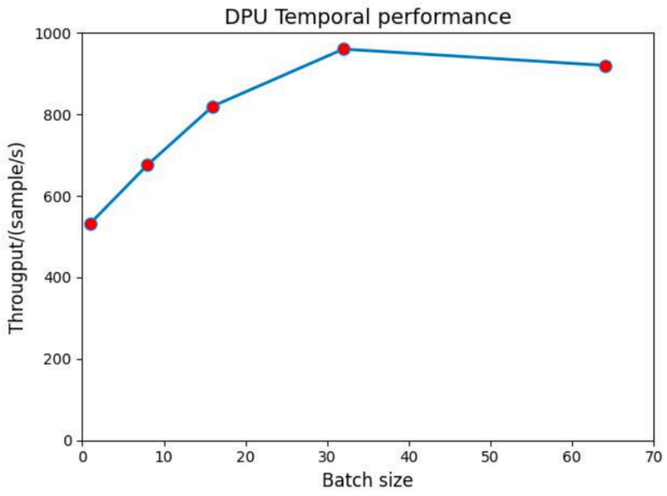
Implementation of CNN on DPU



- Training performed same as typical software model development
- Quantization and compilation using the Vitis-AI tool (AMD)
- Vitis-AI Runtim and Xilinx Runtime tools used for deployment on Versal ACAP

Performance comparison

Inference result based on 10000 samples

	CPU	GPU	DPU
Inference time			
Throughput			

- DPU based on Versal ACAP shows 8-15x(CPU)/2-4x(GPU) inference time

Summary and prospects

- Advance data reduction technique (real-time, ML/AI) is essential for new/upgraded collider experiments
- System design, algorithm and heterogeneous platform developments
- DNN algorithm was designed for L1 trigger, implemented to FPGA
- DNN/GNN algorithm for L1 trigger/HLT was approved able to be implemented to Versal ACAP
  - A possible solution for latency-limit algorithm of HLT (e.g. track seed finding)
- DPU based on Versal ACAP shows 8-15x(CPU)/2-4x(GPU) inference time performance for the STCF PID CNN model
  - A possible solution for HLT (CNN algorithm)