FlashSim: End-to-End simulation with Flow Matching

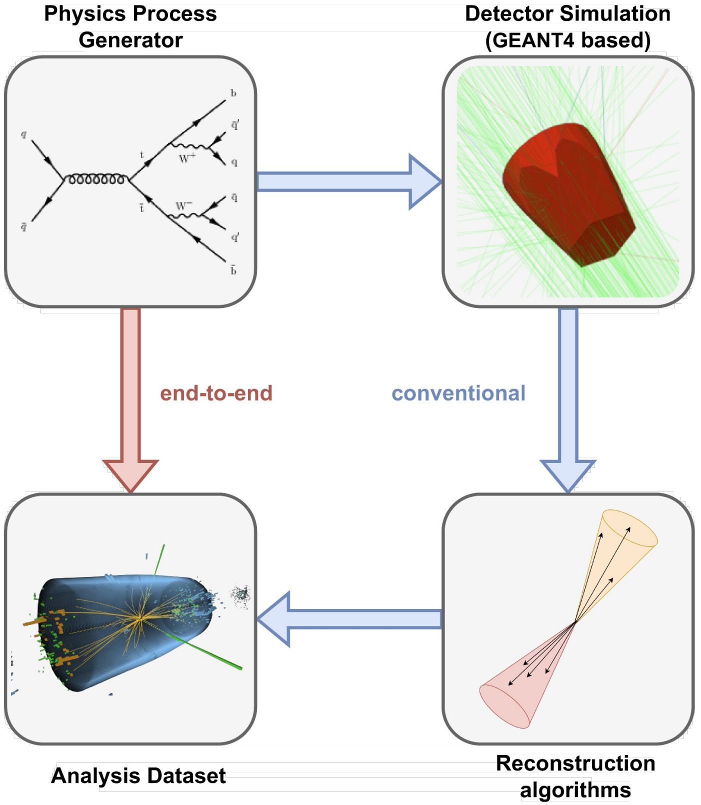
Francesco Vaselli on behalf of the CMS Collaboration
francesco.vaselli@cern.ch

We propose an *end-to-end* approach for faster simulations

Main idea: going directly from the generator output objects to the high level analysis objects (jets, muons ...)!

We want something:

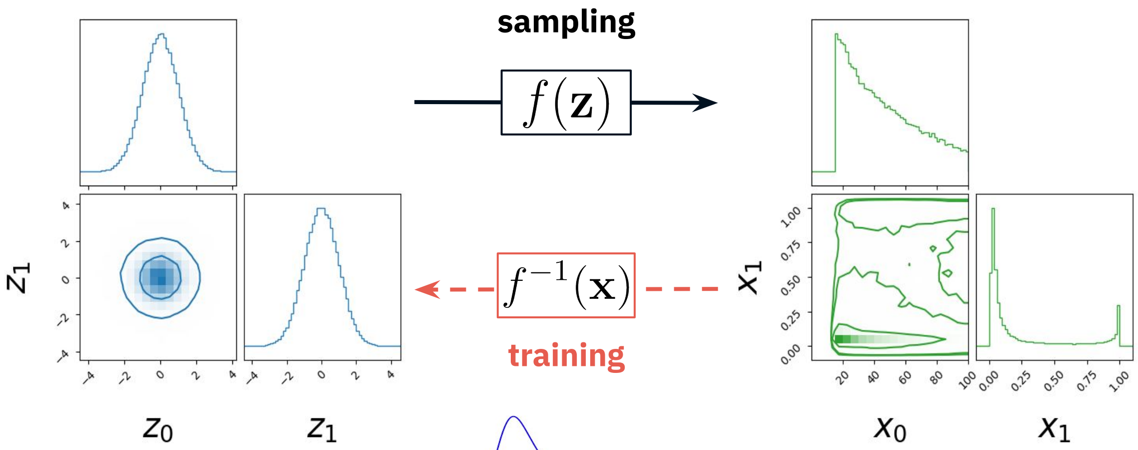
- Fast(er): reached ~kHz!
- Not analysis specific
- Depending on Gen (not just a generic event but the event)

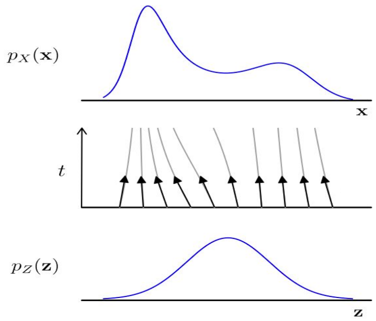


Continuous Normalizing Flows are the backbone of our approach!

We learn an invertible transformation, taking us from data *x* to noise *z*

Once *f* has been found we can invert it, start from noise and sample new data from the unknown PDF!





see https://arxiv.org/abs/2210.02747, and https://arxiv.org/abs/2302.00482, figure from https://ehoogeboom.github.io/post/en_flows/

Results are convincing

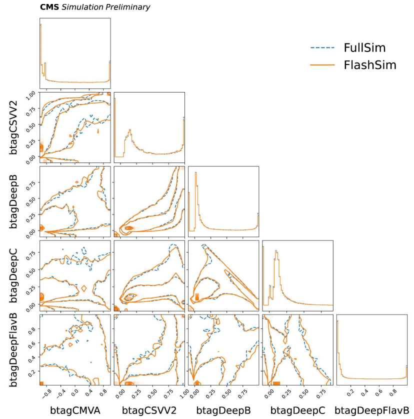
Simulation speed per object is around 10 kHz.

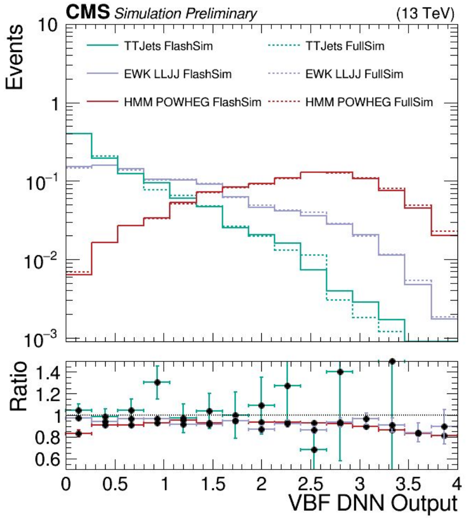
Our results accurately reproduce the Full Simulation data of the CMS Experiment, on both training and unseen processes, for:

- 1-d distributions;
- correlations between the variables;
- different physical processes;
- analysis-level plots.

For more:
francesco.vaselli@cern.ch

and
https://cds.cern.ch/record/2858890, https://arxiv.org/abs/2402.13684



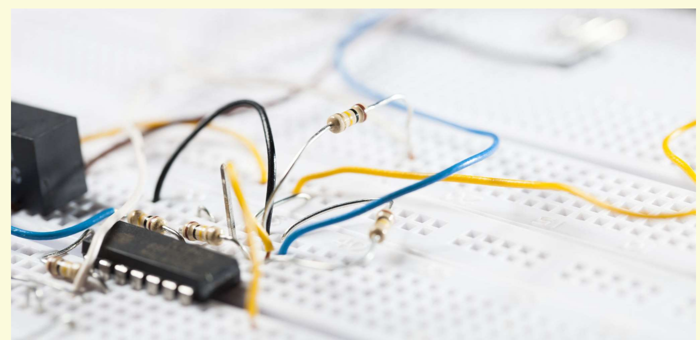


Improving the Inference of the Graph Neural Networks for Track Reconstruction

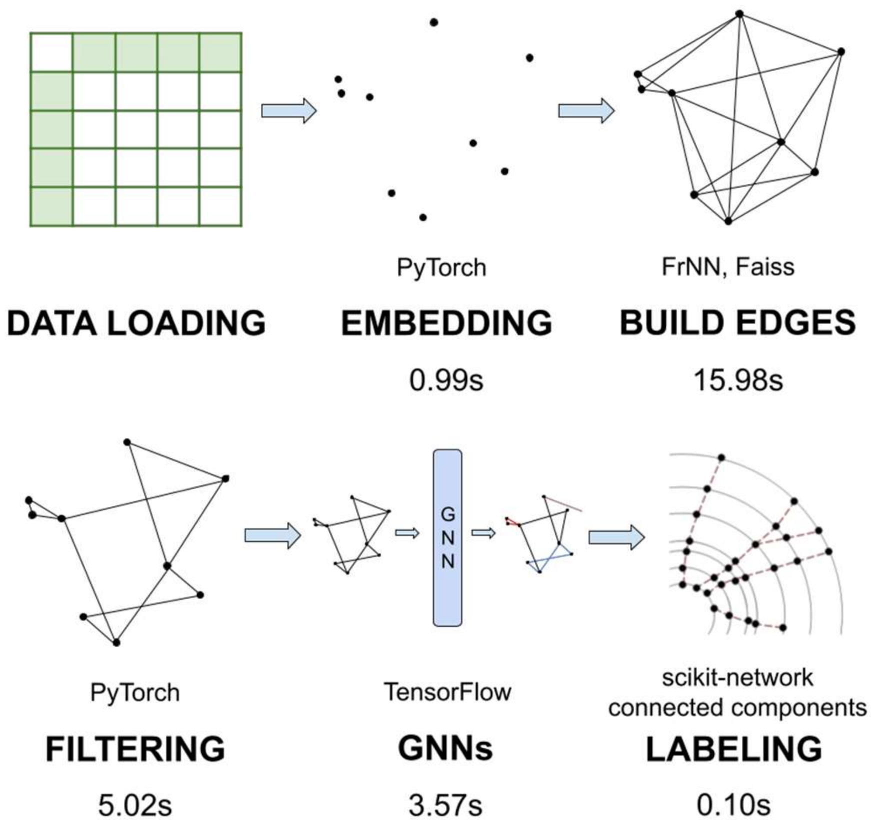
James S Gaboriault-Whitcomb, Henry H Paschke and Alina Lazar

on behalf of the **Exa.TrkX** collaboration

Youngstown State University,



The Exa.TrkX GNN Inference Pipeline



	GPU (ms)	CPU (s)
Data Loading	2.2	
Metric Learning	6.7	0.99
Graph building	40 ± 10	15.98
Filtering	370 ± 80	5.02
GNN	170 ± 30	3.57
Track Building (CC)	90 ± 8	0.1
Total	700 ± 100	25.66

MPI was used to run events in parallel, using multiple cores.

The most time-consuming steps of the pipeline are Build Edges and Filtering. To speed-up Build Edges we used Faiss with 2 threads and multiprocessing for the Filtering for-loop.

The results indicate that it is best to use between 10 and 15 cores per event, however running it on the GPU is still 27 times faster.

EPFL

X-HEEP

3

EMBEDDED SYSTEMS LABORATORY

September 23, 2024

An Open-Source RISC-V-based GPGPU Accelerator for Machine Learning-based Edge Computing Applications

EPFL - Embedded Systems Laboratory (ESL)

simone.machetti@epfl.ch

Motivation

Analyzing the state-of-the art, we realized the need for a GPU…

Motivation

Analyzing the state-of-the art, we realized the need for a GPU…

Open-source

Motivation

Analyzing the state-of-the art, we realized the need for a GPU…

Open-source

Natively Configurable

Motivation

Analyzing the state-of-the art, we realized the need for a GPU…

Open-source

Natively Configurable

RISC-V-based

Motivation

Analyzing the state-of-the art, we realized the need for a GPU…

- Open-source
- Natively Configurable
- RISC-V-based
- Fully Synthesizable

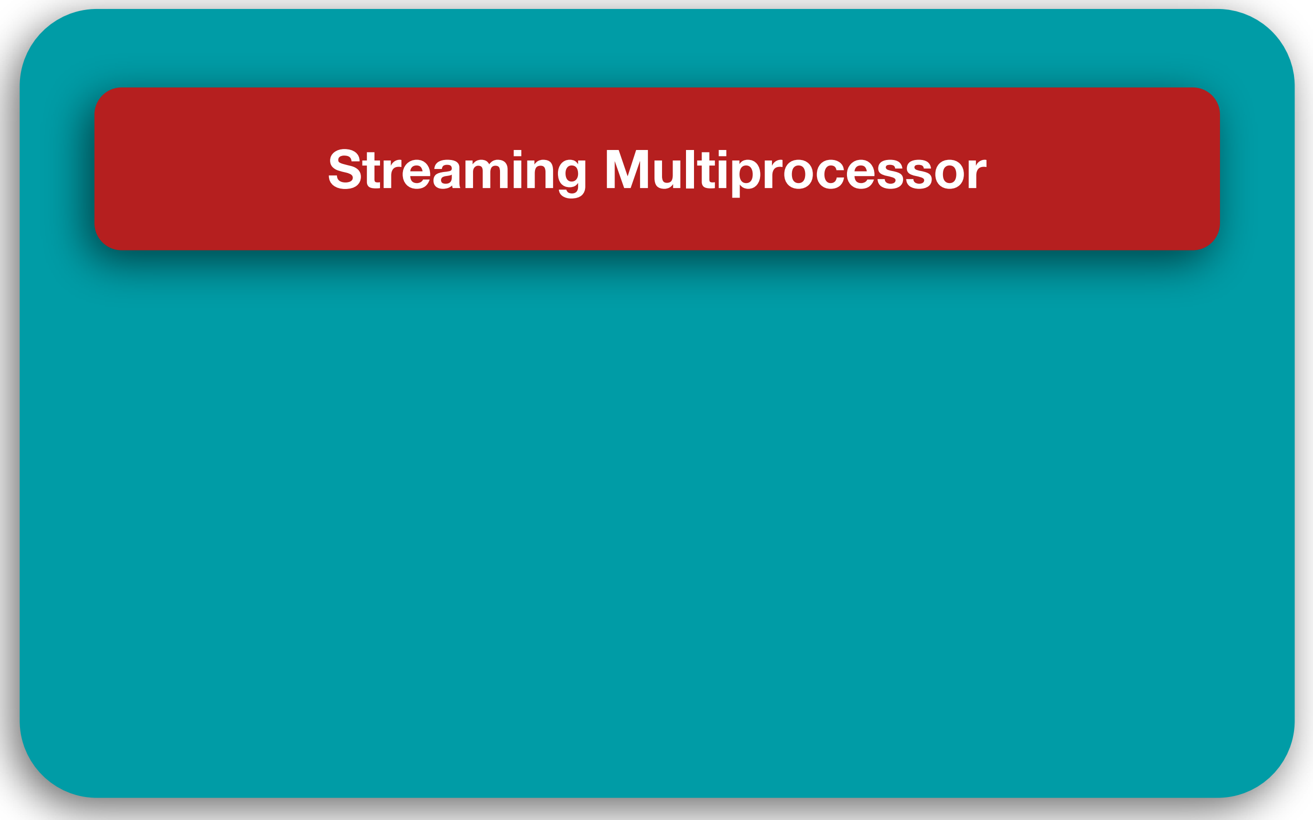
Motivation

Analyzing the state-of-the art, we realized the need for a GPU…

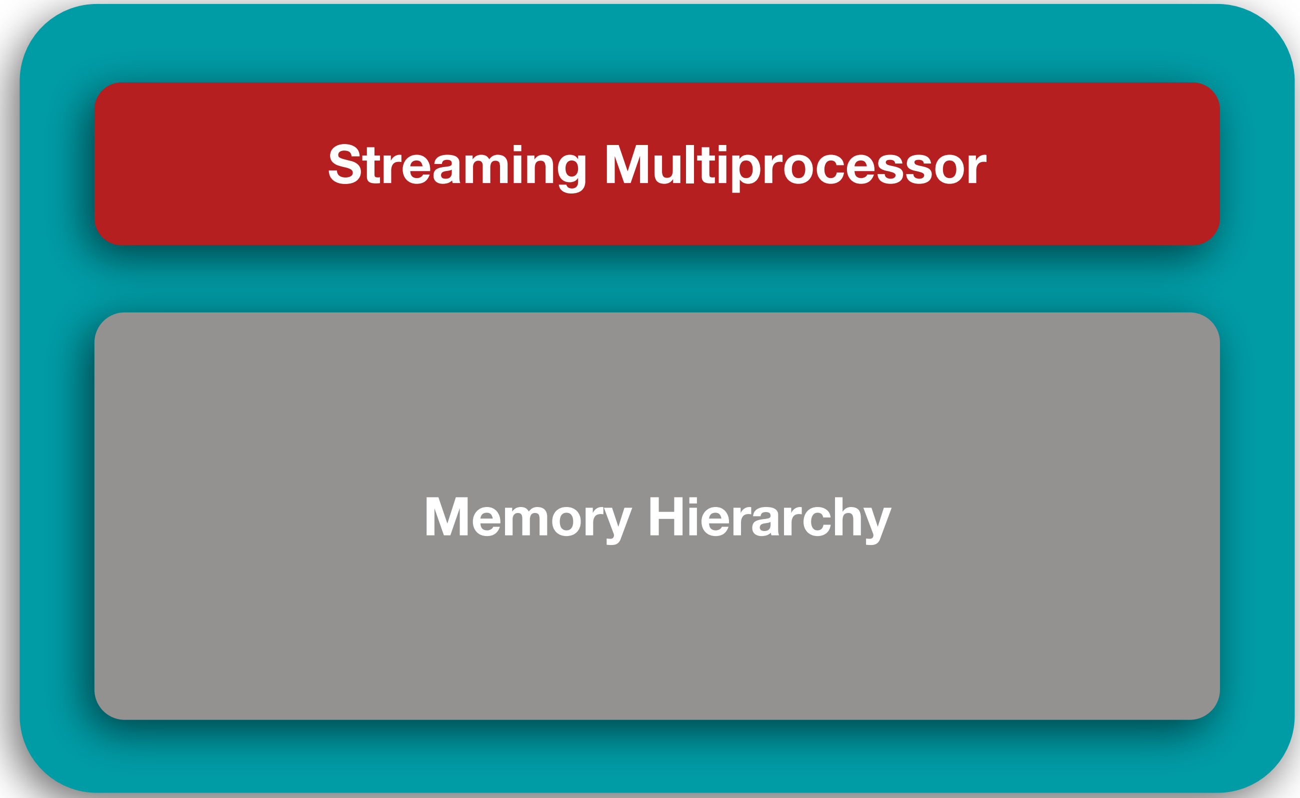
- Open-source
- Natively Configurable
- RISC-V-based
- Fully Synthesizable
- OpenCL Support

GPGPU Accelerator

GPGPU Accelerator

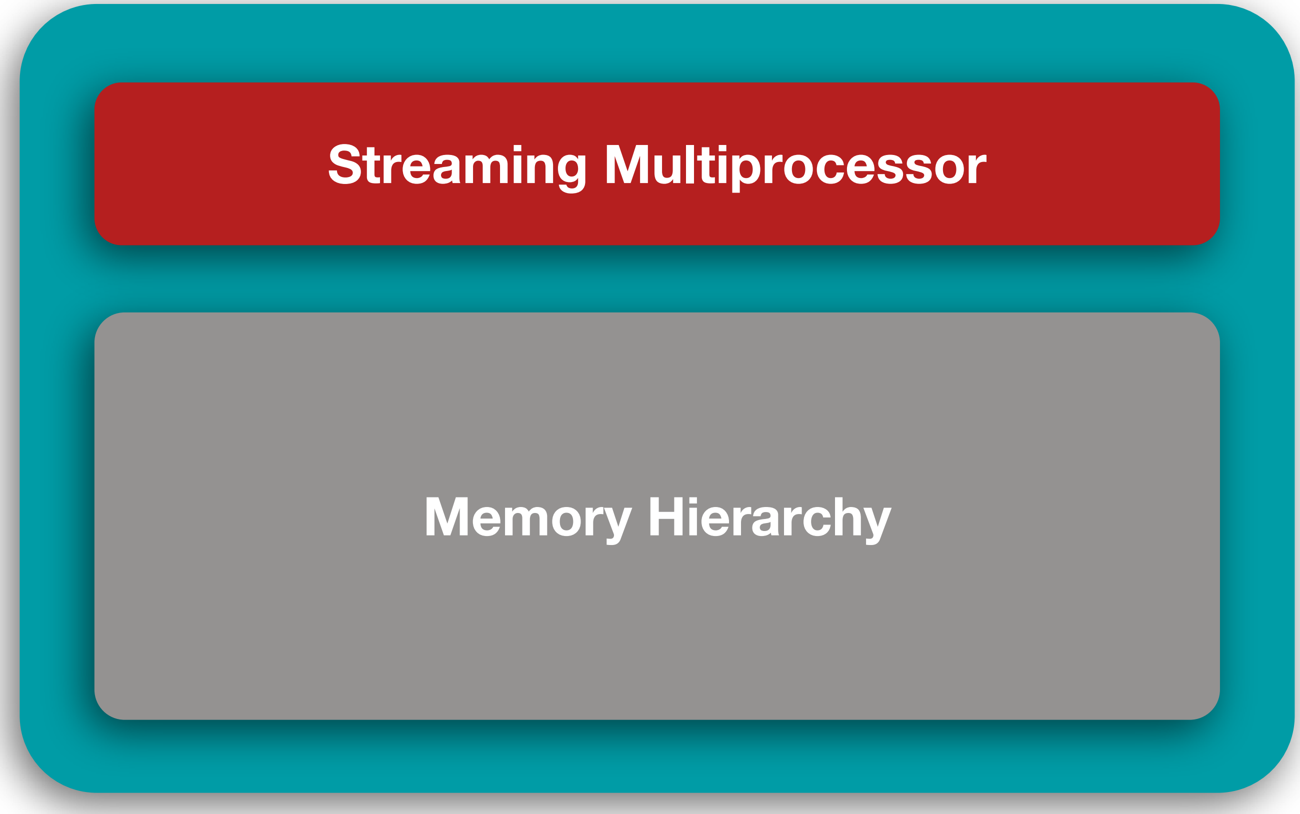


GPGPU Accelerator



GPGPU Accelerator

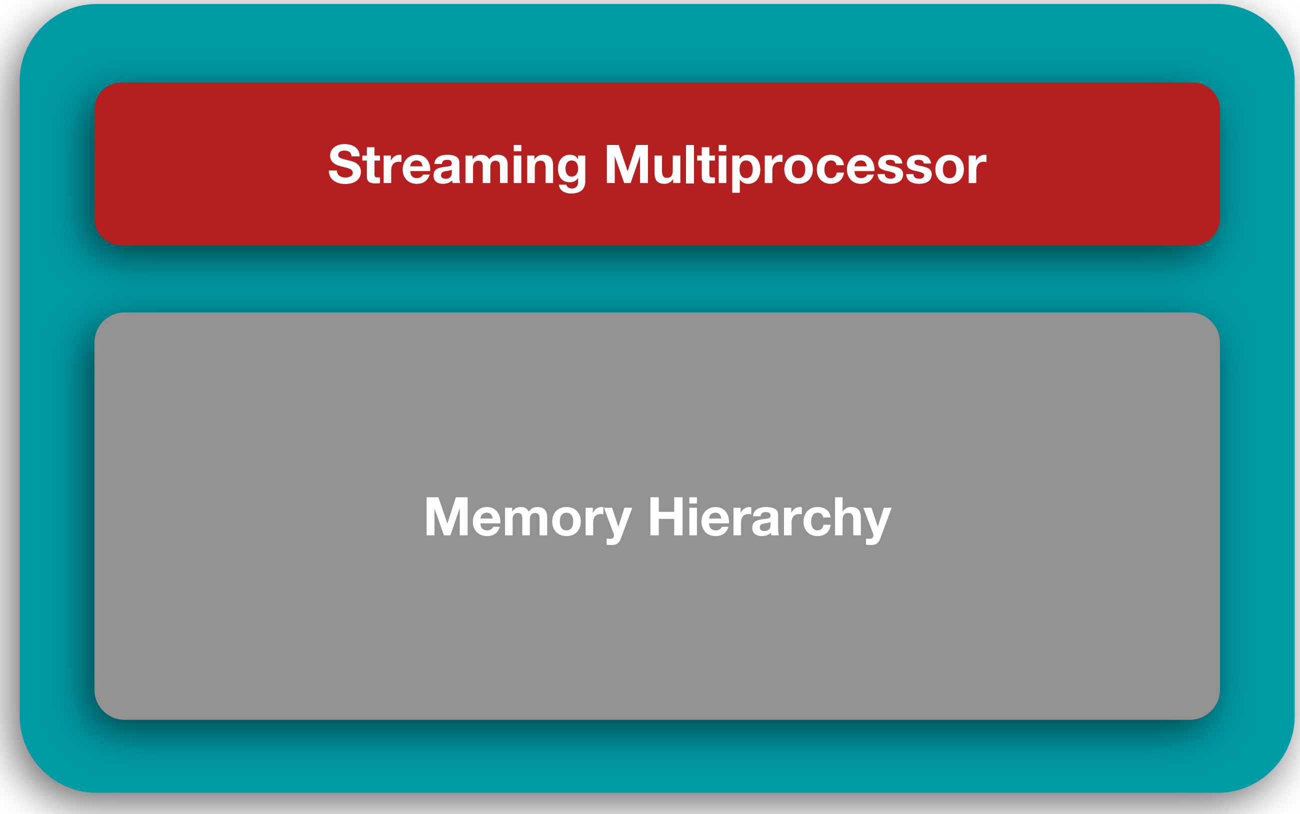
Configurable



GPGPU Accelerator

Configurable

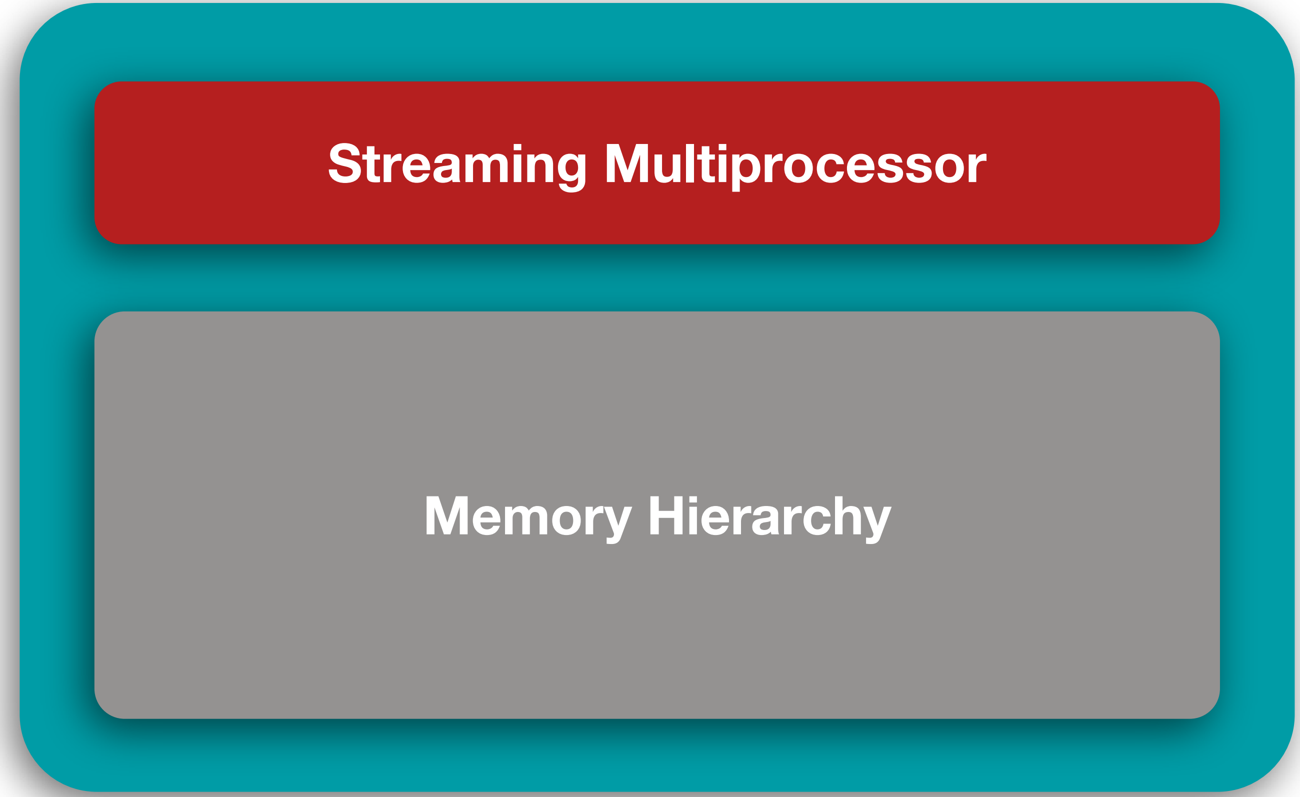
- Number of threads



GPGPU Accelerator

Configurable

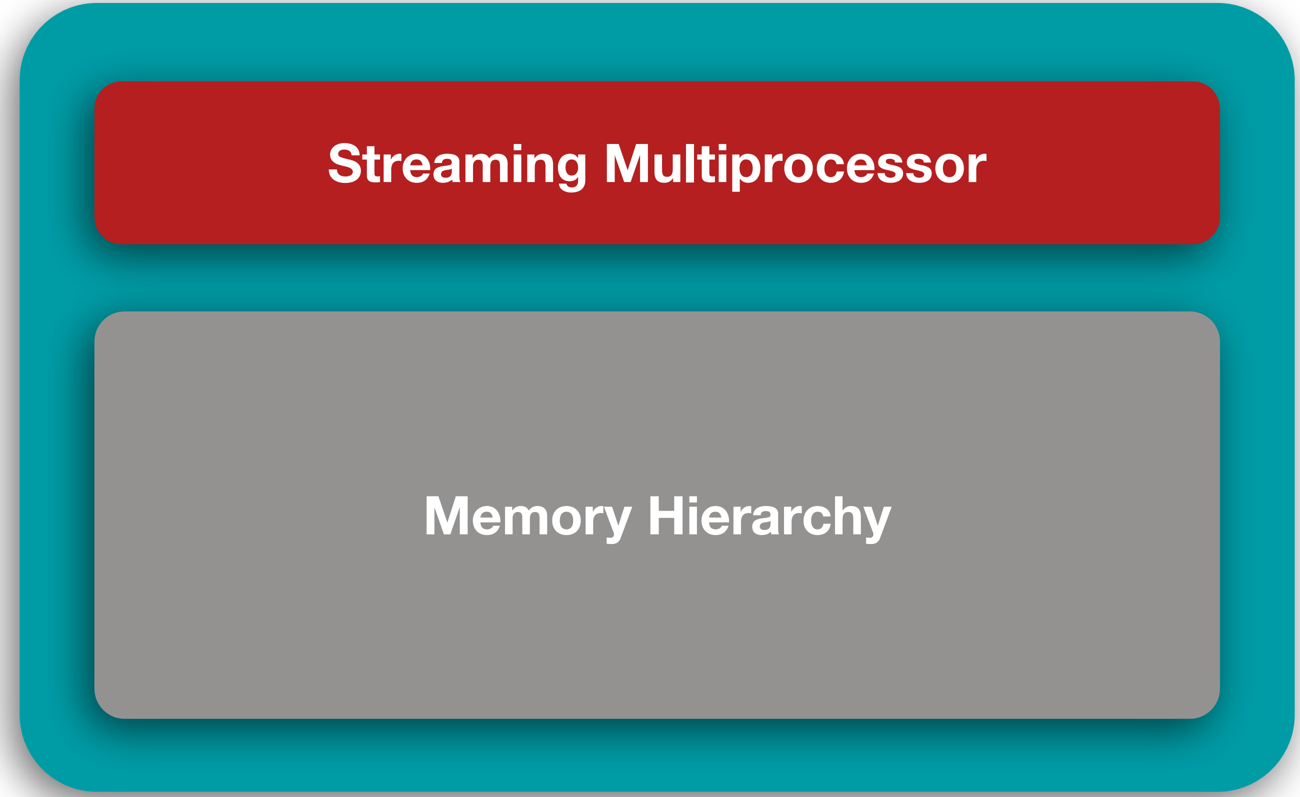
- Number of threads
- Number of warps



GPGPU Accelerator

Configurable

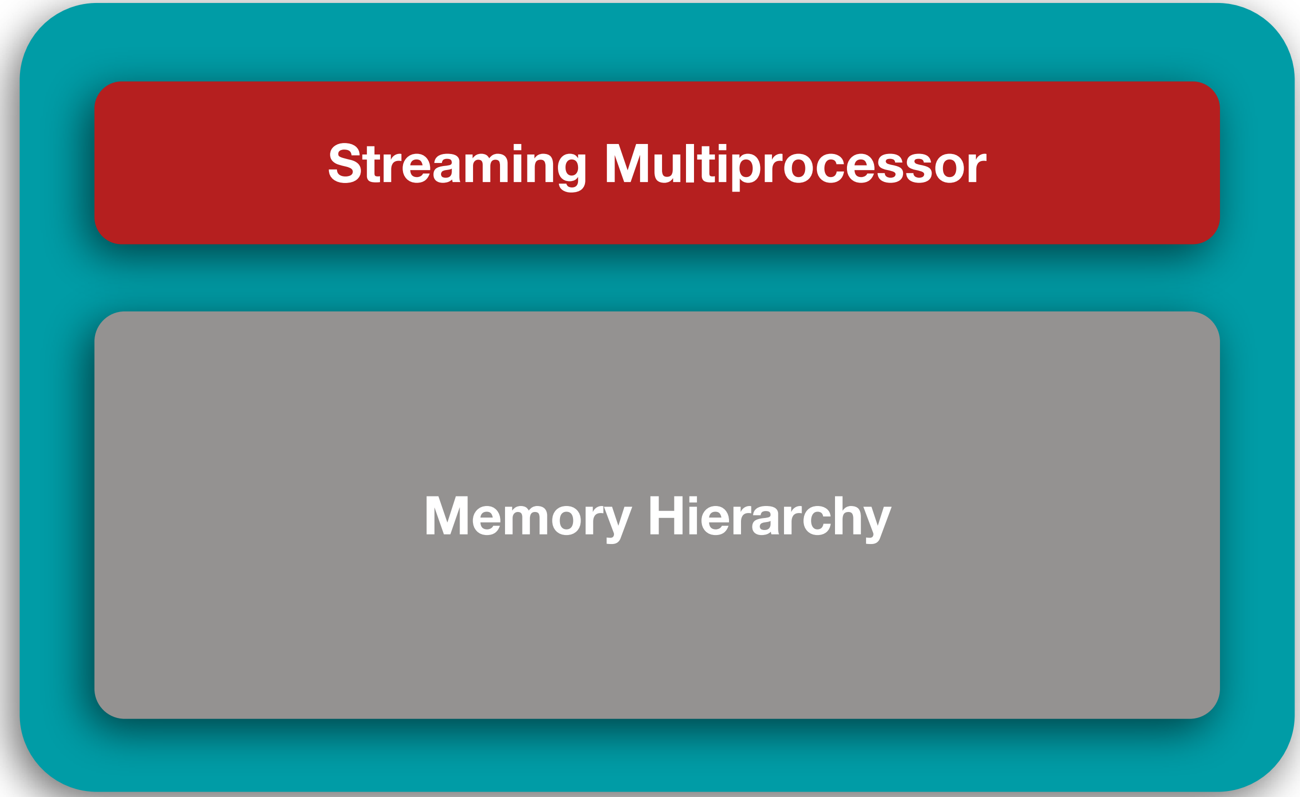
- Number of threads
- Number of warps
- Floating-point unit



GPGPU Accelerator

Configurable

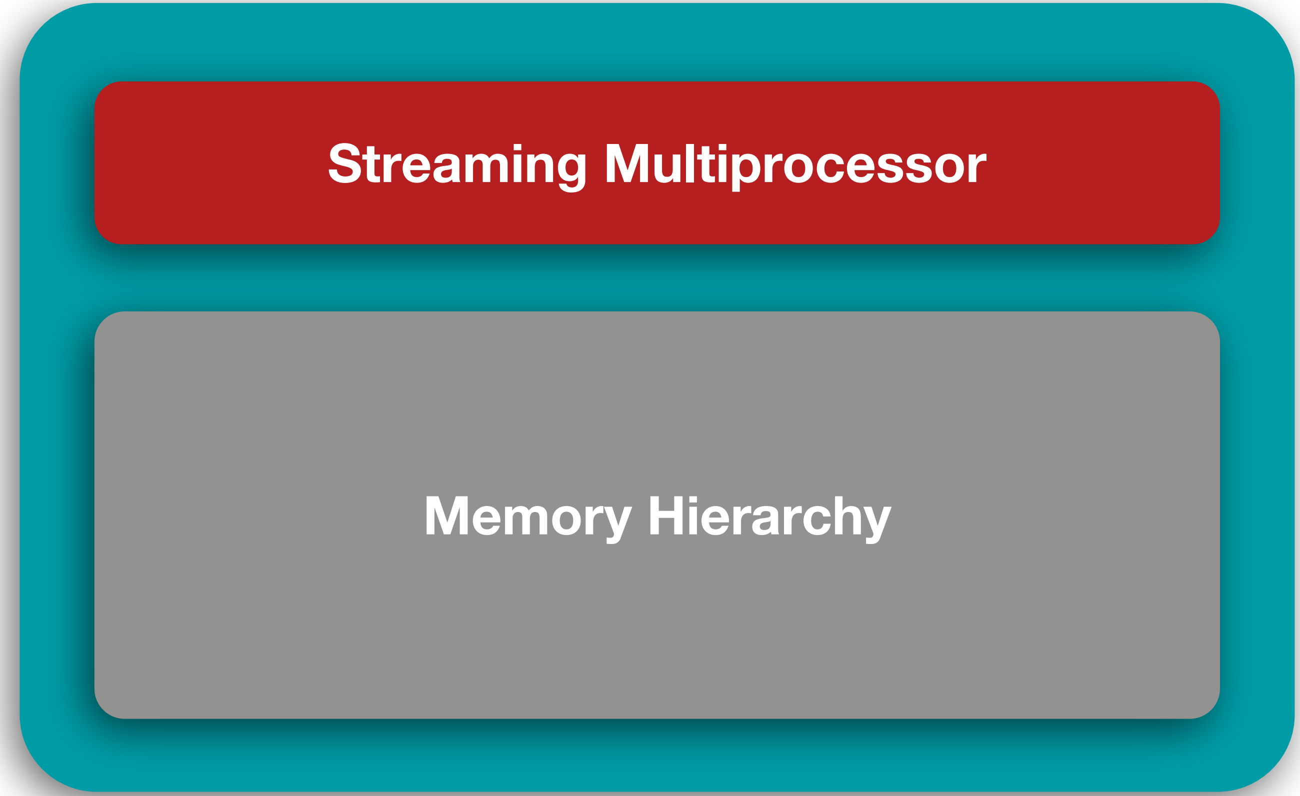
- Number of threads
- Number of warps
- Floating-point unit
- Memory hierarchy



GPGPU Accelerator

Configurable

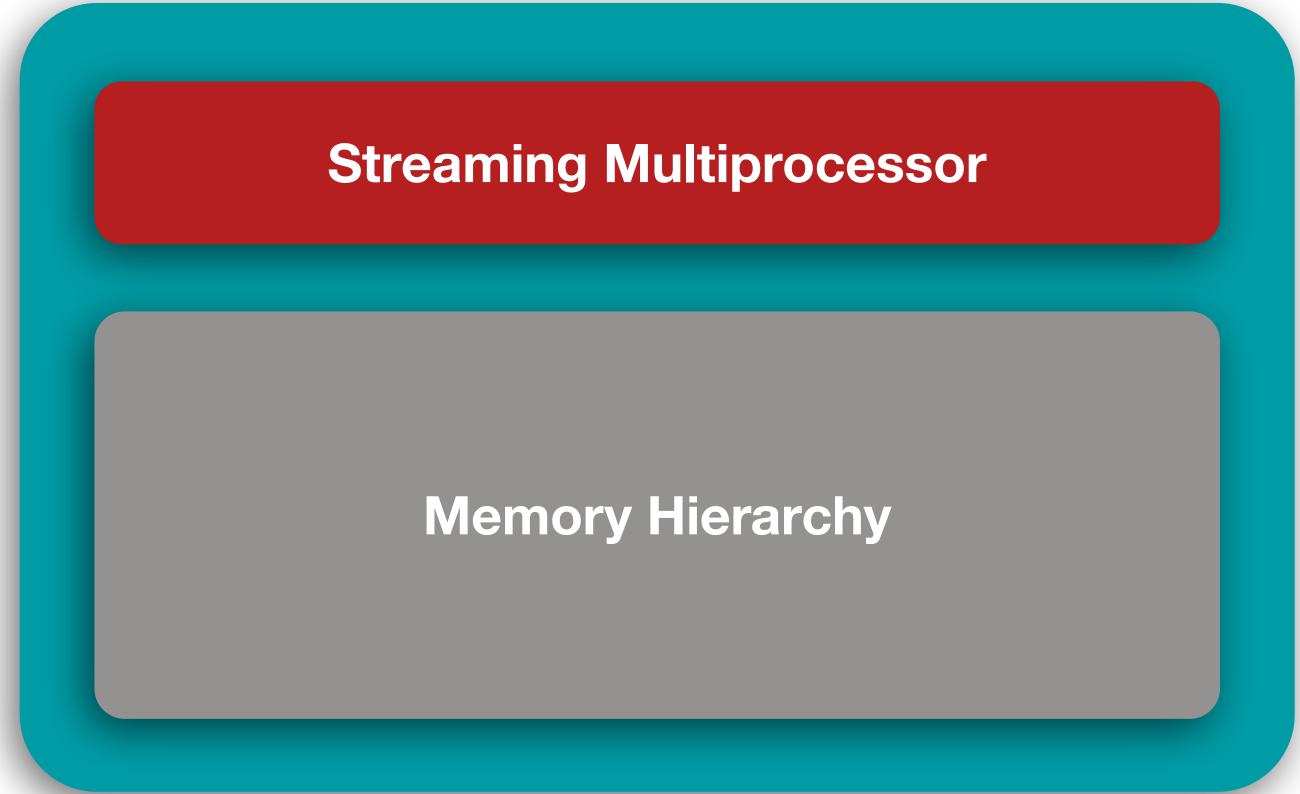
- Number of threads
- Number of warps
- Floating-point unit
- Memory hierarchy
    - Scratchpad-based



GPGPU Accelerator

Configurable

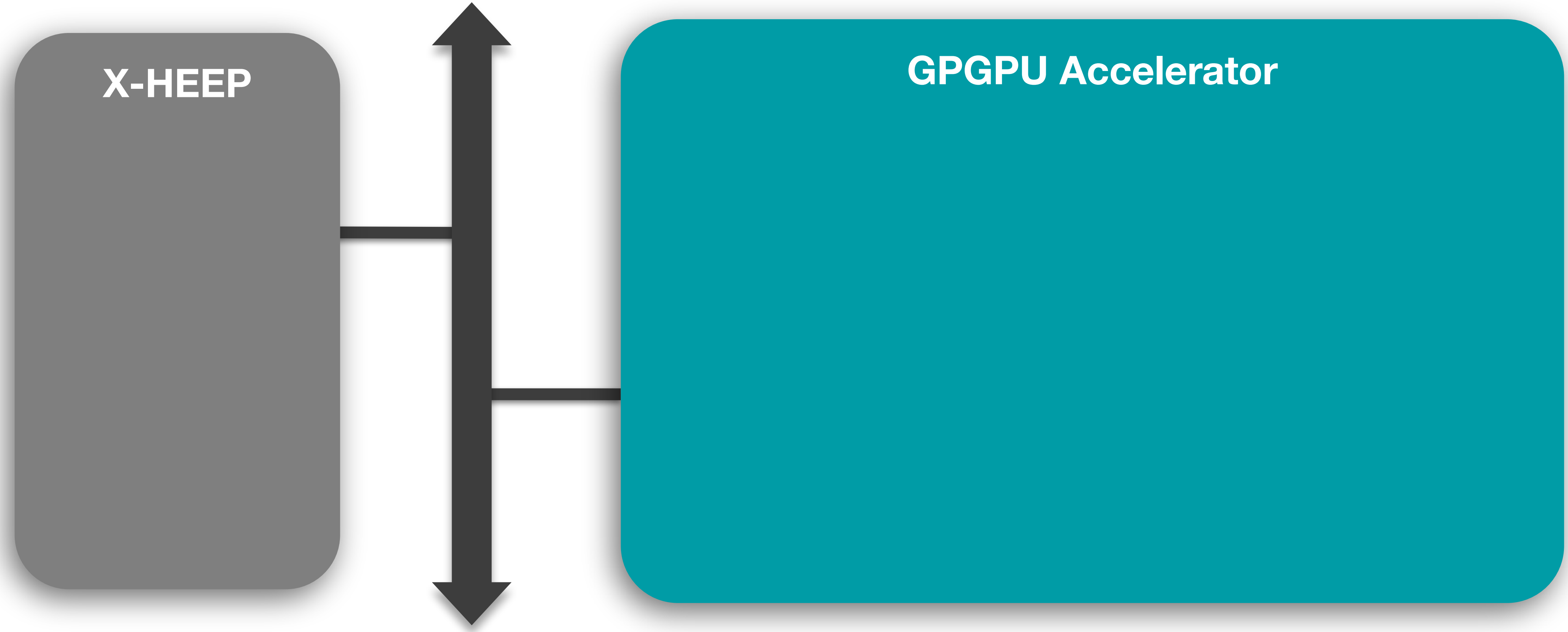
- Number of threads
- Number of warps
- Floating-point unit
- Memory hierarchy
  - Scratchpad-based
  - Cache-based



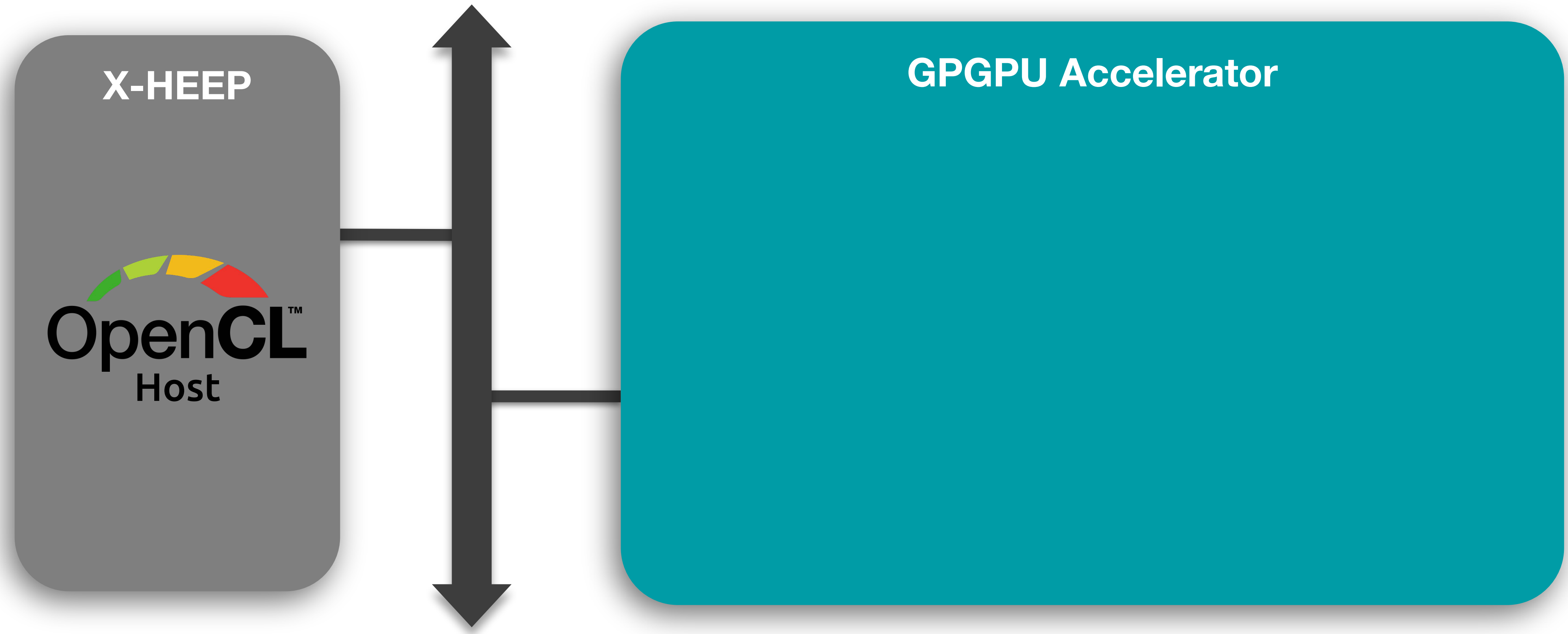
Accelerated Processing Unit (APU)

GPGPU Accelerator

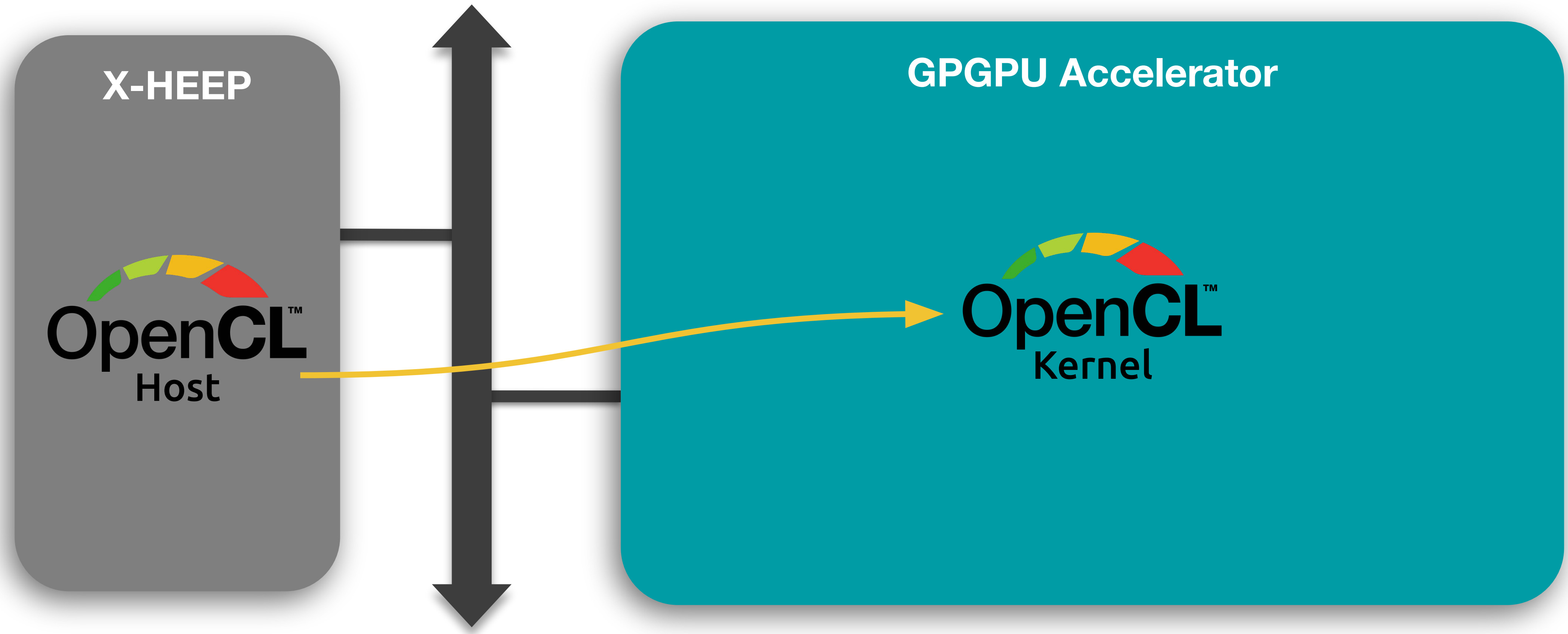
Accelerated Processing Unit (APU)



Accelerated Processing Unit (APU)



Accelerated Processing Unit (APU)



Conclusion

The APU code and documentation will be 100% open-source and the first version will be released very soon…

Conclusion

The APU code and documentation will be 100% open-source and the first version will be released very soon…

Conclusion

The APU code and documentation will be 100% open-source and the first version will be released very soon…

GitHub

Stay Tuned!

Thank you for your attention!

EPFL - Embedded Systems Laboratory (ESL)

simone.machetti@epfl.ch

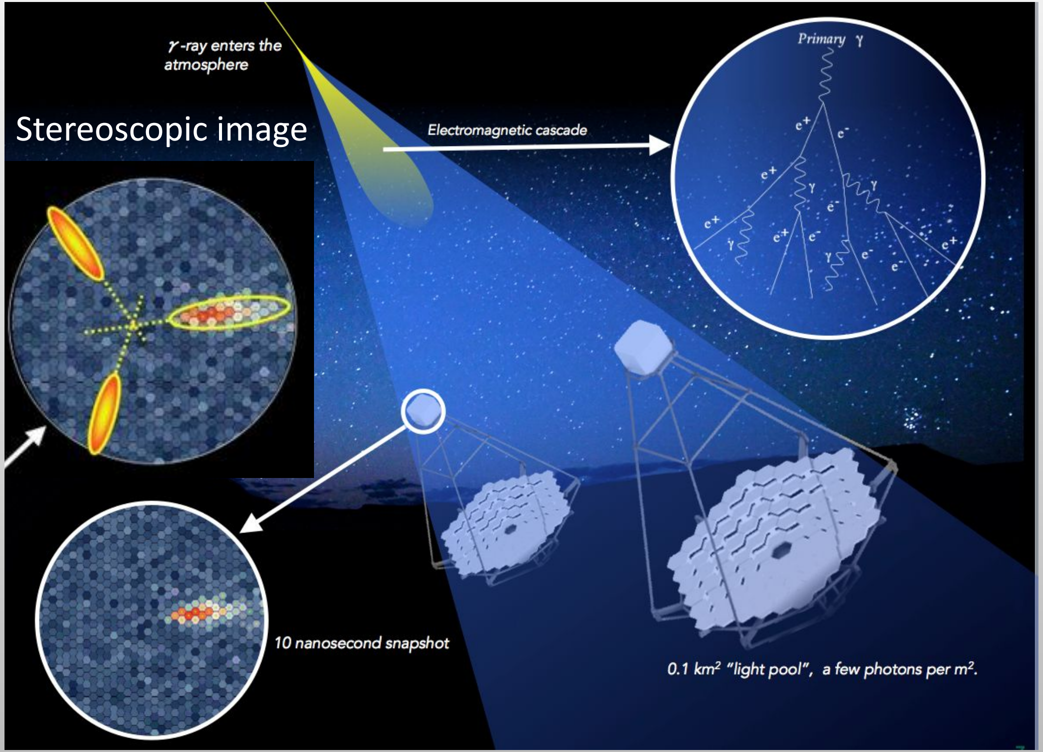
4

Accelerating Machine Learning algorithms in FPGAs for the trigger system of a SiPM-based upgraded camera of the CTA Large-Sized Telescopes

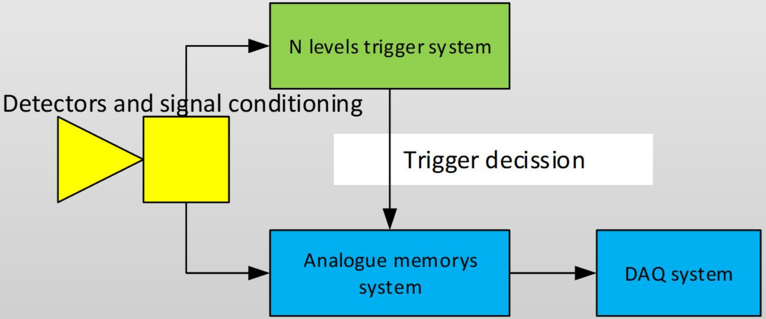
A. Pérez Aguilera[1], L. Á. Tejedor[1], J. A. Barrio[1], T. Miener[2], D. Martín[1]

(1) Grupo de Altas Energías (GAE), Instituto de Física de Partículas y del Cosmos, and EMFTEL Department, Universidad Complutense de Madrid (IPARCOS-UCM), E-28040 Madrid, Spain
(2) University of Geneva - Département de physique nucléaire et corpusculaire, 24 Quai Ernest Ansernet, 1211 Genève 4, Switzerland

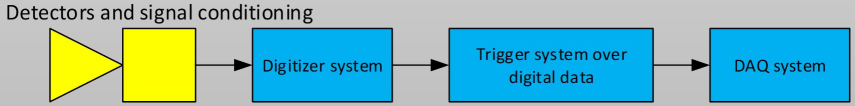
IACTs introduction



Combined analogue and digital trigger system approach with a separated branch for event data:



Fully digital trigger system approach:



Implementing ML algorithms for the trigger system

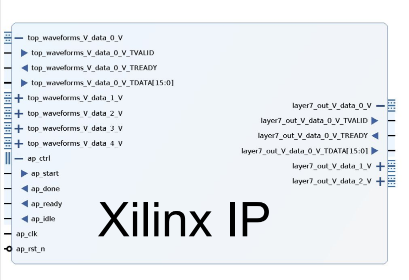
Fully digital trigger ⇒ More complex algorithms to tag/eliminate NSB events ⇒ Possibility of Machine Learning

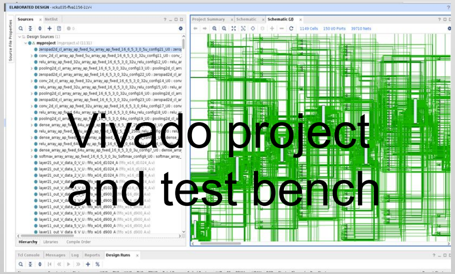
Hundreds of kHz ⇒ Processing time few μs ⇒ FPGAs

```
Model: "CTLearn_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 waveforms (InputLayer)      [(None, 30, 30, 5)]       0

 SingleCNN_block (Functiona  (None, 16)                1536
 l)

 fc_particletype_1 (Dense)   (None, 32)                544

 particletype (Dense)        (None, 3)                 99

 type (Softmax)              (None, 3)                 0

=================================================================
Total params: 2179 (8.51 KB)
Trainable params: 2179 (8.51 KB)
```

Reduced TensorFlow model used for IACT offline event analysis.

hls 4 ml





Preliminary results when simulating with RoIs composed of 5 samples of 30x30 pixels

R. Factor	Latency (us)	DSP
1	5.2	122
8	12.9	66
16	15.3	52
32	15	29
64	20.4	17
128	33	9
256	41	6

Discussion and near future activities

-**Several RoIs** need to be processed **in parallel** to cover all the area of a camera event.

-Further optimizations of the CNN models, such as **quantization aware training**, are yet to be explored.

-Density-Based Scan models also to be explored.

-Works to check the **tagging performance** are ongoing.

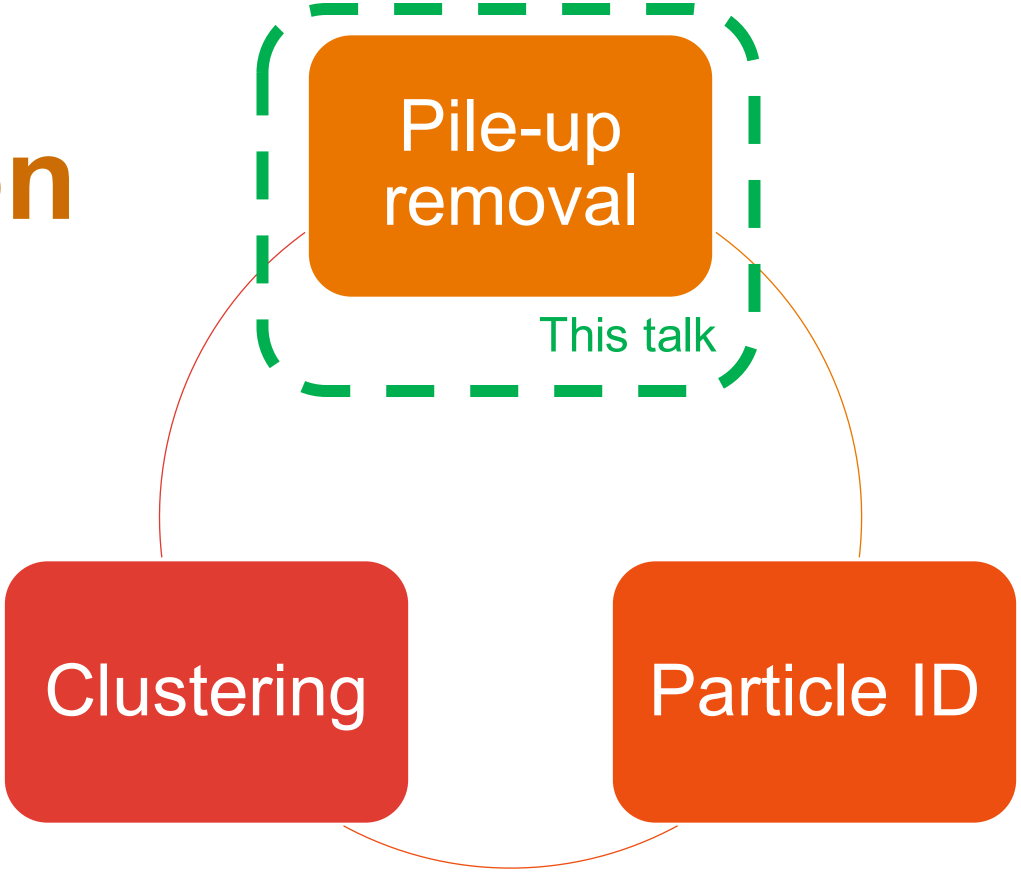
-Recently joined DRD7.5 WP to share expertise.

-Short-term: test-bench/algorithms characterized by 2026.

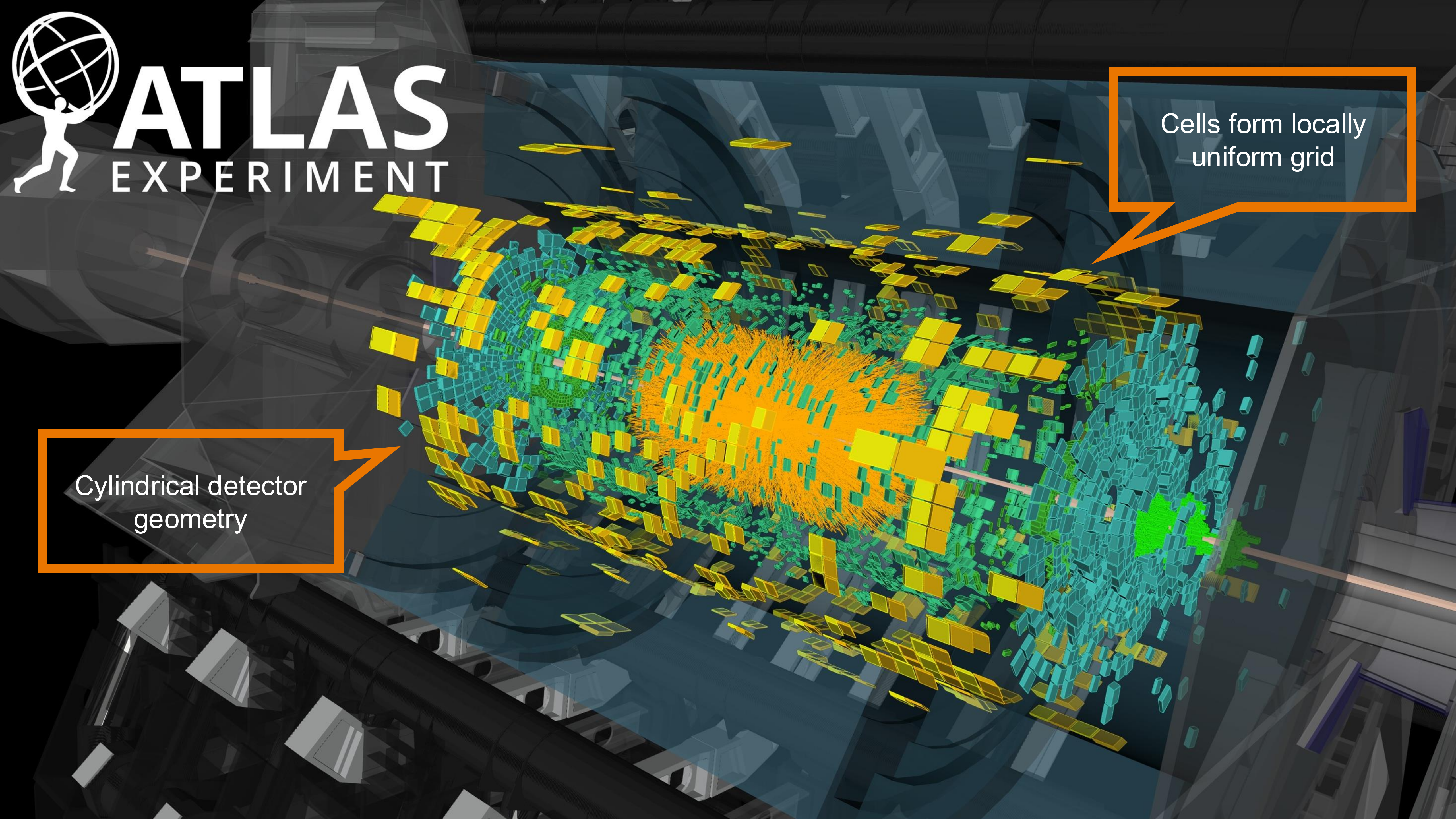
Mid-term: full prototype produced by 2028.

Nanosecond ML for calorimeter segmentation

Noah Clarke Hall, Nikos Konstantinidis, Alex Martynwood, Naoki Kimura

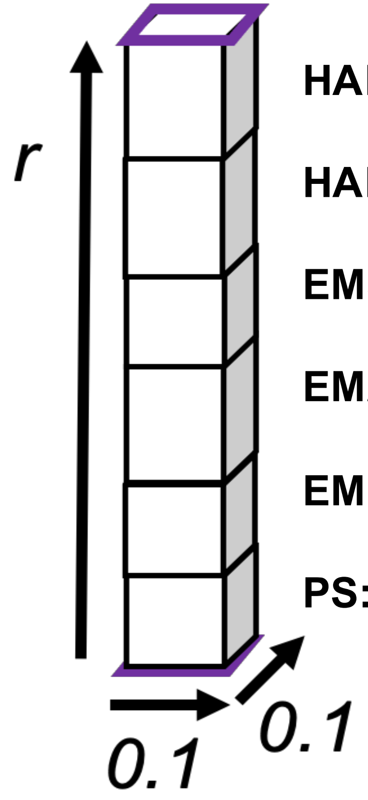


5


ATLAS
EXPERIMENT
Cells form locally uniform grid
Cylindrical detector geometry

Towers & topoclusters

Tower (|η|<2.5)



HAD2: Sum of 1 >2σ cells

HAD1: Sum of 1 >2σ cells

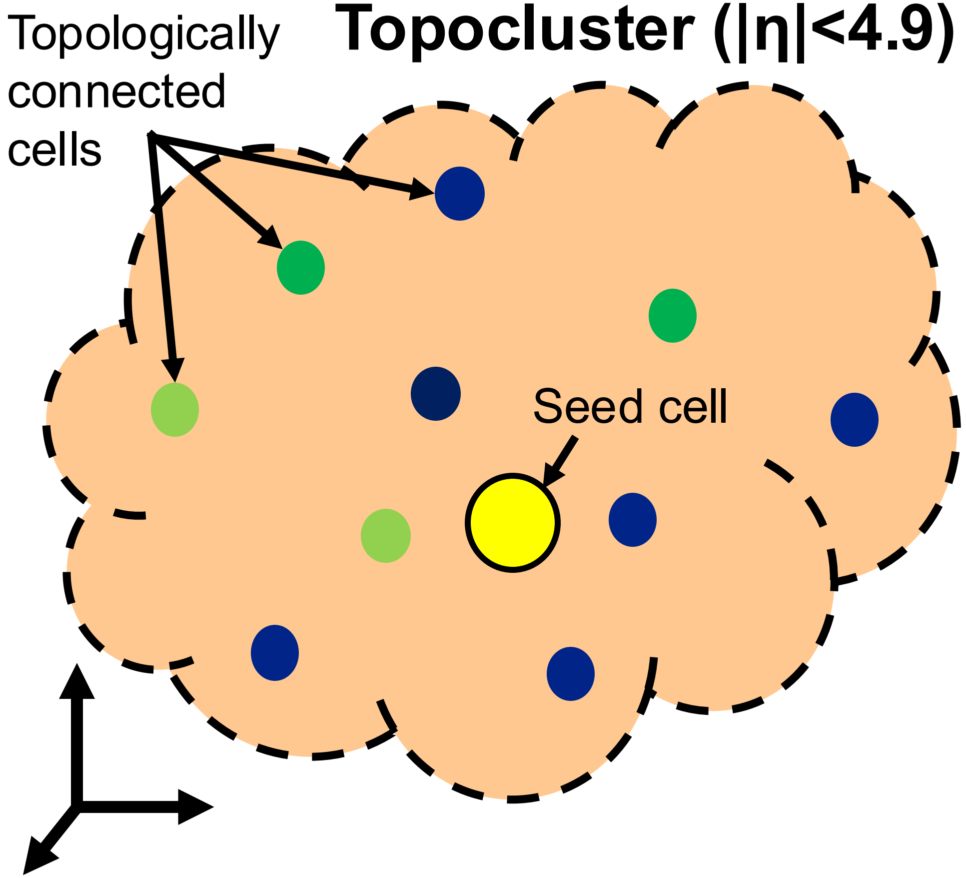
EM3: Sum of 8 >2σ cells

EM2: Sum of 16 >2σ cells

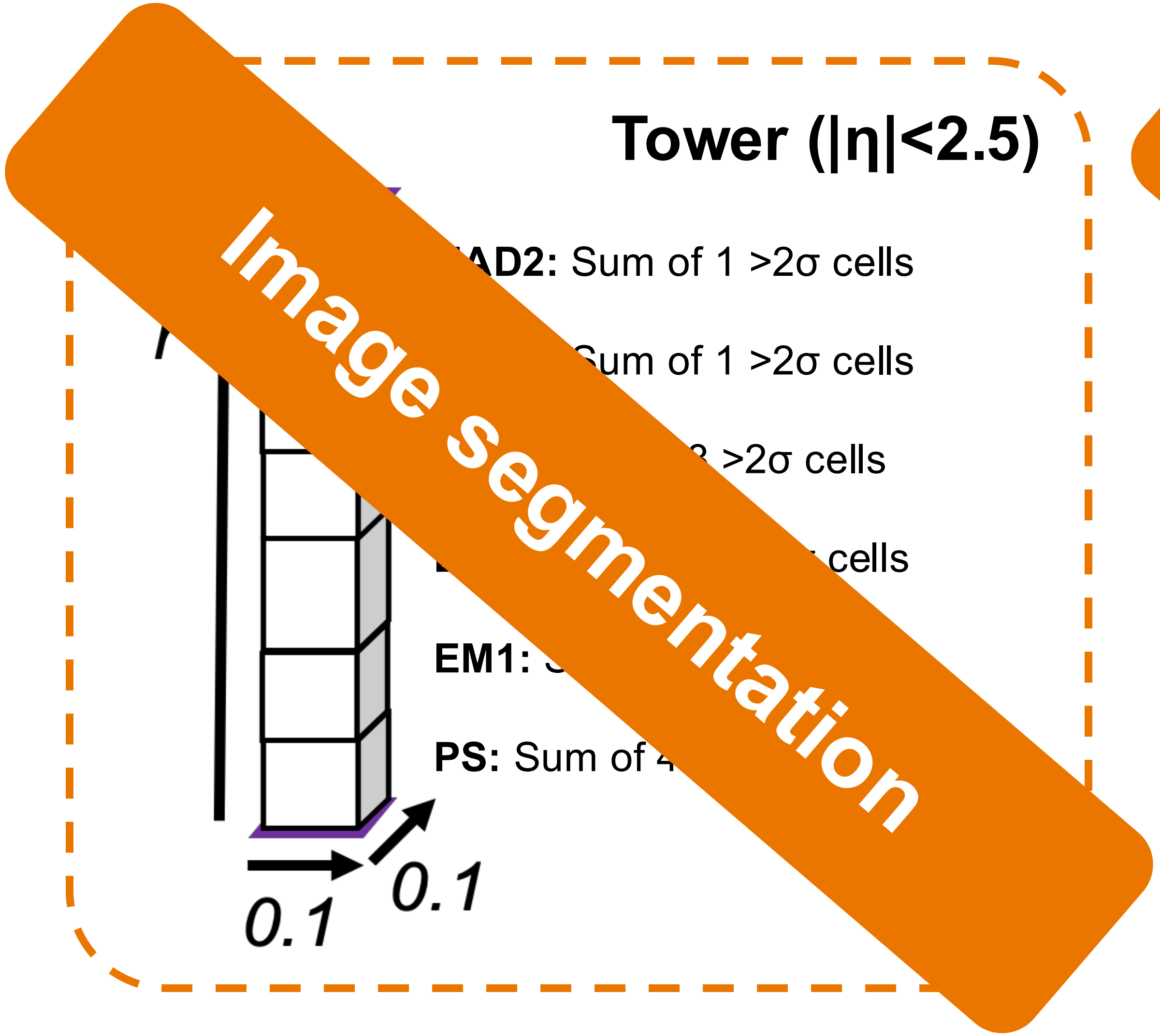
EM1: Sum of 32* >2σ cells

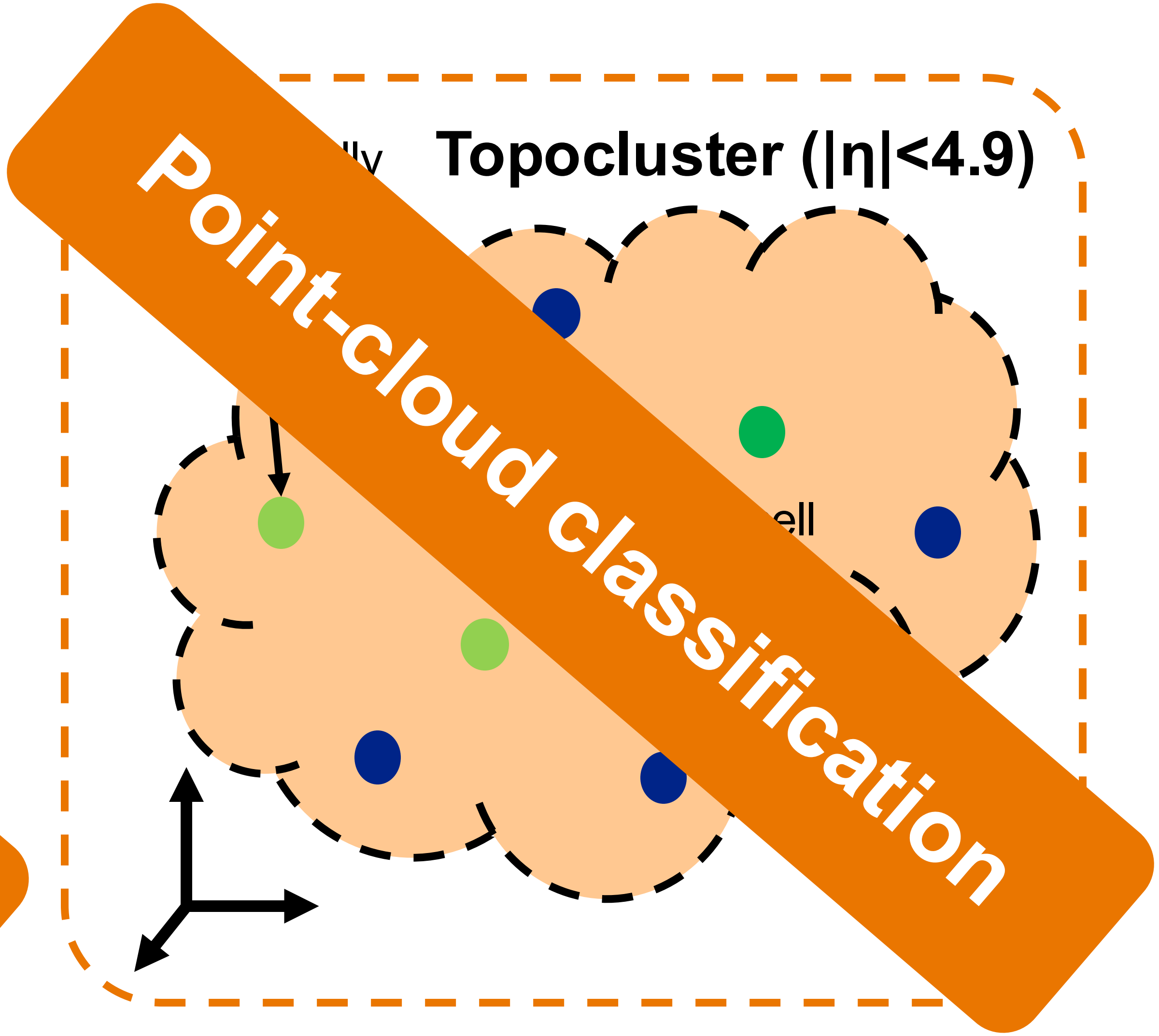
PS: Sum of 4 >2σ cells

Topocluster (|η|<4.9)



Towers & topoclusters

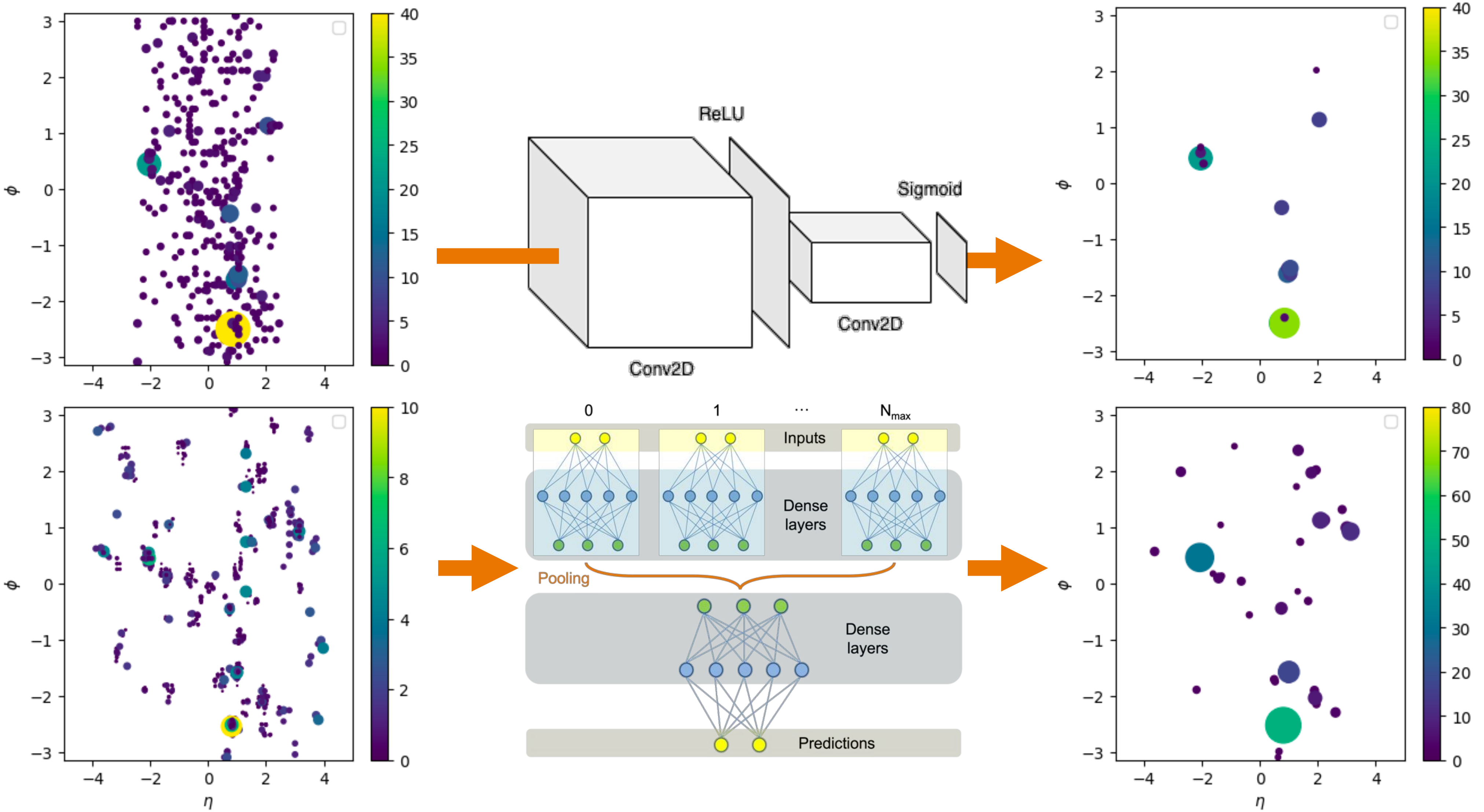




Two ML approaches

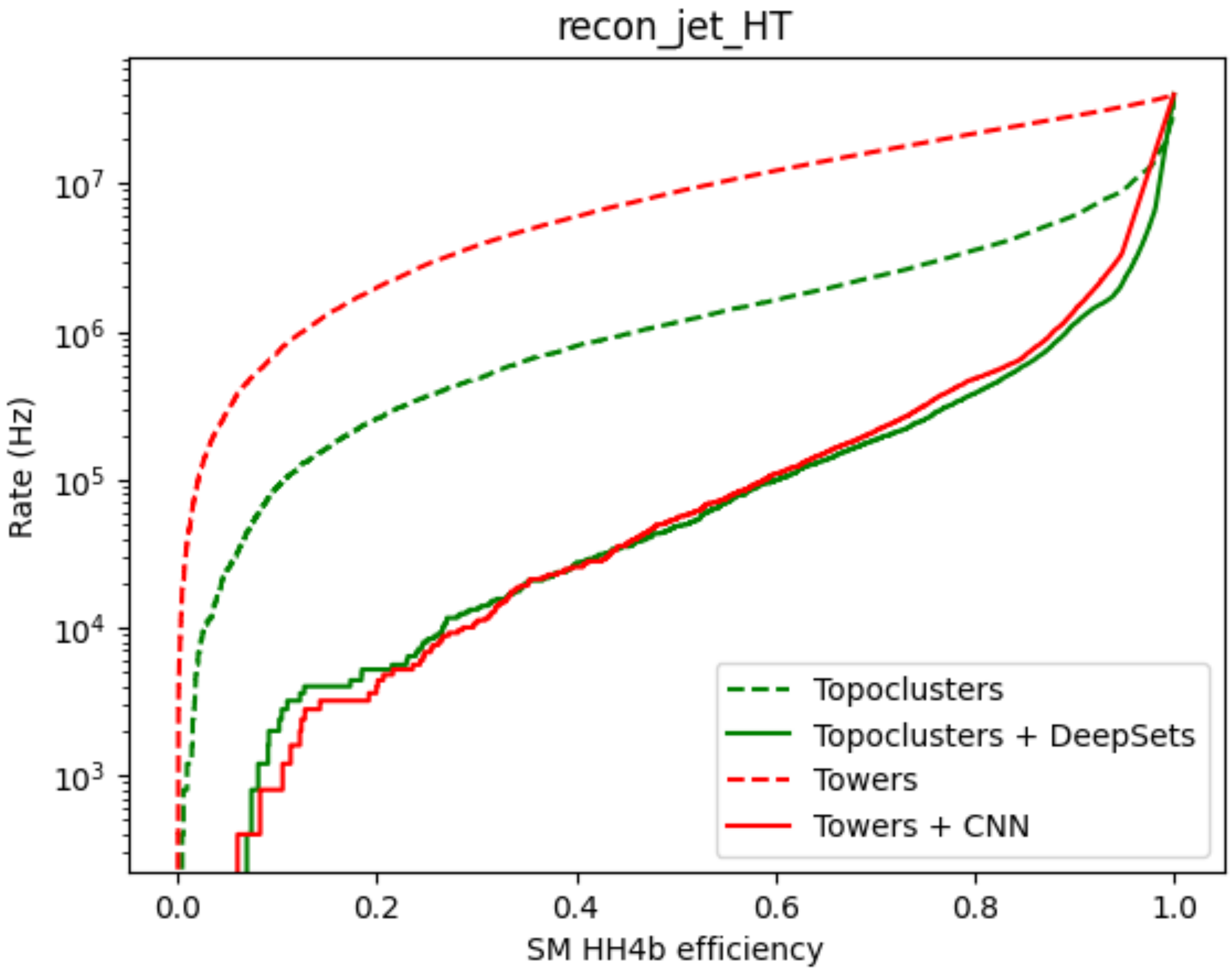
Image segmentation

Point cloud classification



Physics performance

- Form anti-k_t central ($|\eta|<2.5$) jets
- Both approaches give similar physics performance
- Large improvement over baselines!



Resources

google/**qkeras**

QKeras: a quantization deep learning library for Tensorflow Keras

hls4ml

- Xilinx UltraScale+ XCU250
- 250 MHz clock
- CNN looks fast & light enough to be viable
- More optimisation needed

Resource/timing	CNN	DeepSets
Precision	Fixed <10,5>	Fixed <10,5>
# parameters	494	913
Latency (clk)	5	73
Interval (clk)	2	25
BRAM_18K	0	0
DSP	0	16
FF	1883	54478
LUT	33529	270742
URAM	0	0

6

Enhancing the L0 Muon Trigger: project goals and needs

SMARTHEP Edge Machine Learning school (23-27 Sept 2024, CERN)

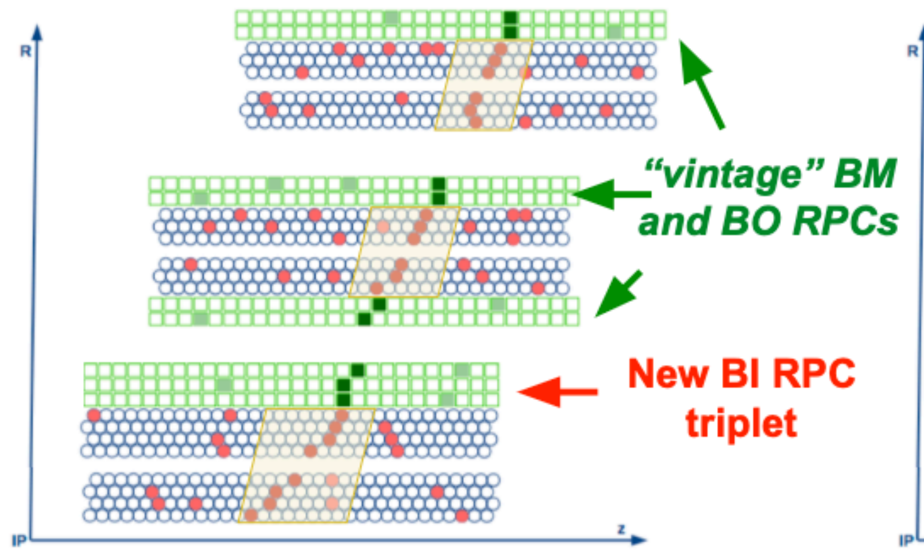
Oliver Kortner (MPI), Verena Martinez Outschoorn (UMass Amherst)
Maria Carnesale (CERN), Rimsky Rojas (CERN)

Enhancing the L0 Muon Trigger

L0 MDT trigger: improve the robustness of L0 muon trigger system against the potential loss of performance due to aging RPC detectors and to improve acceptance coverage

Hit Extraction

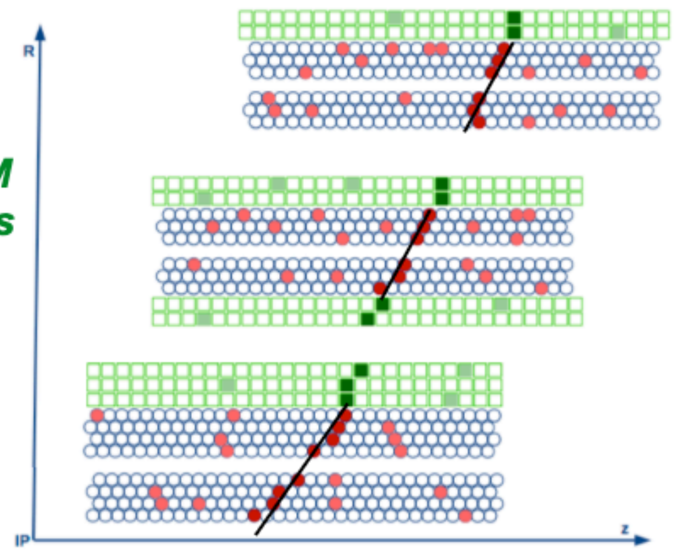
RPCs provide seeds to identify MDT hits from a muon & set up segment fitting step



Pattern recognition algorithms to identify regions of interest with only MDT hits

Segment Fitting

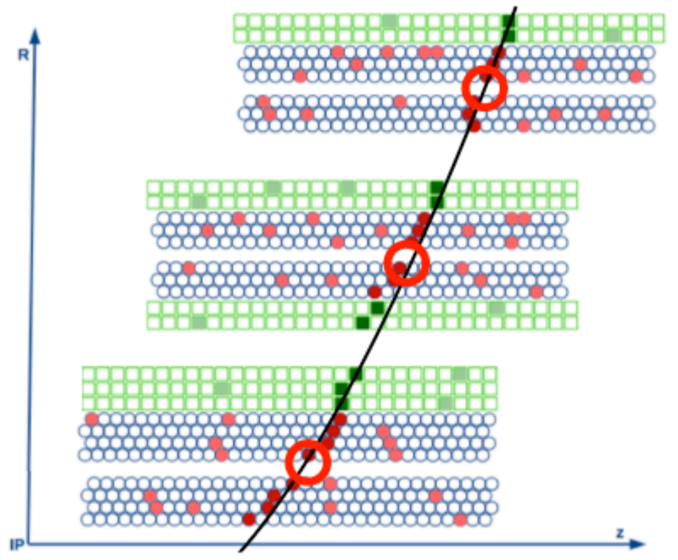
RPCs provide timing to calibrate hits and derive segments



Timing of the muons to determine bunch crossing with Tile or only MDTs

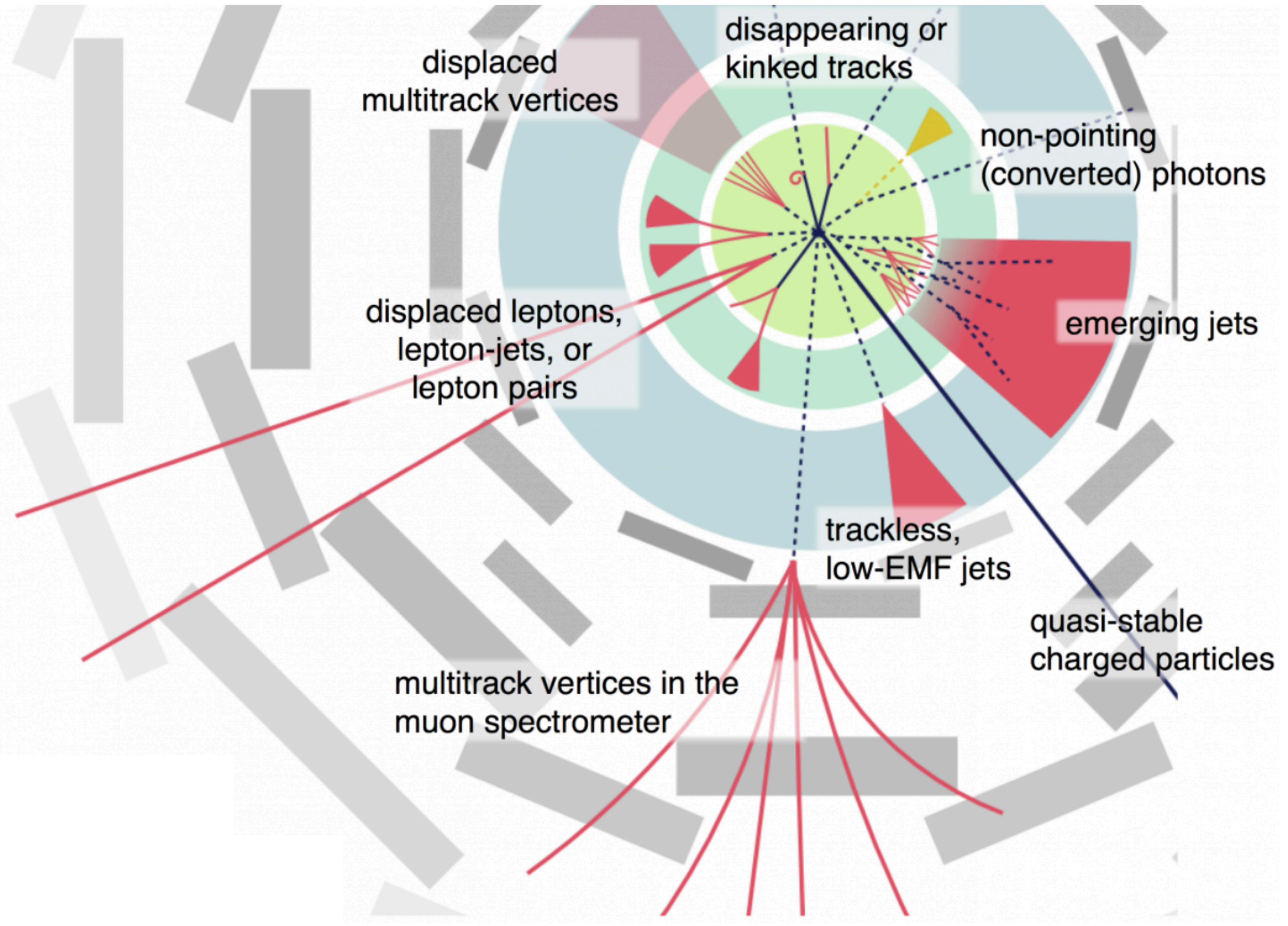
Momentum Estimation

RPCs provide second coordinate for the p_T estimate since the B-field is non-uniform



Momentum estimation without a second coordinate from RPCs

Exotic signatures: additional trigger strategies for non-pointing signatures from decay of long-lived exotic particles



Implement novel trigger strategies in firmware

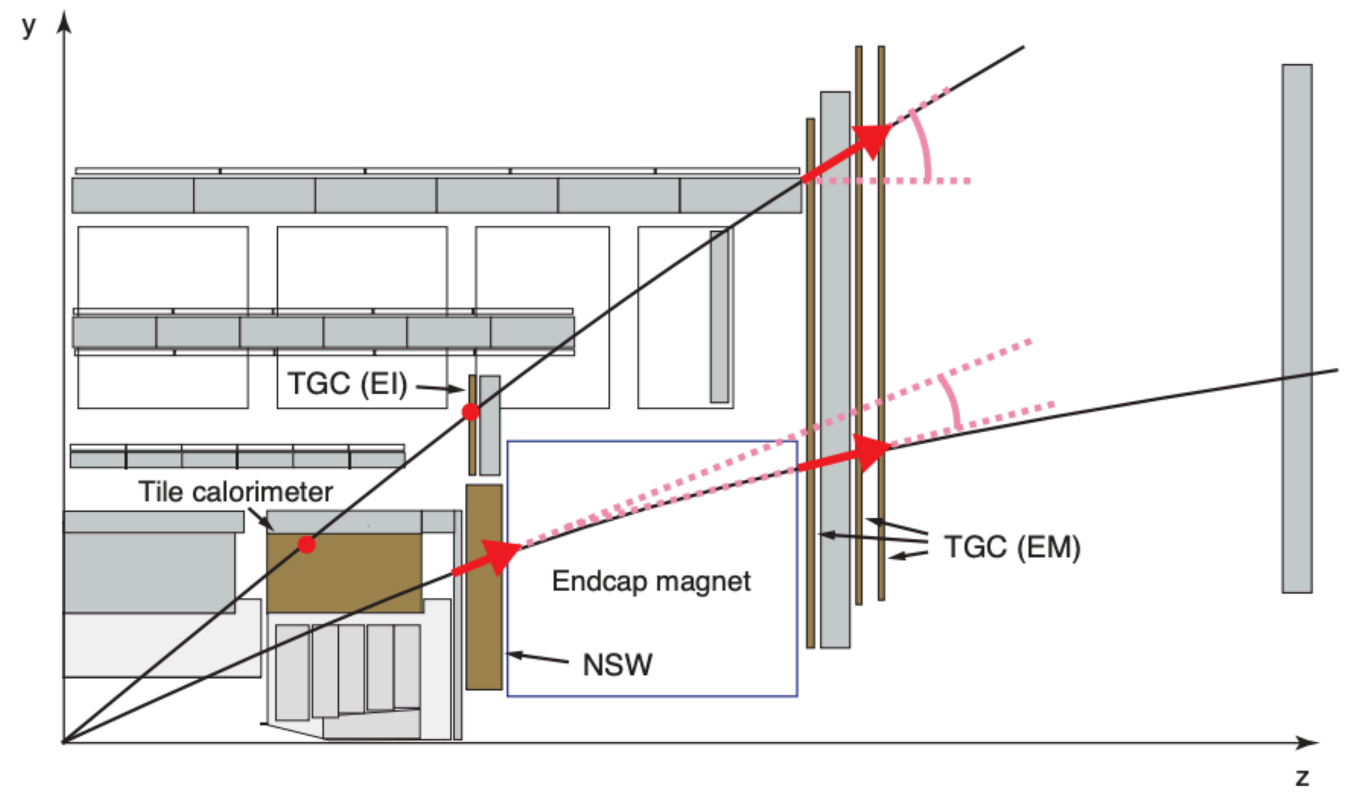
Starting from displaced muons, but also interested in closeby muons, high multiplicity signatures, slow moving or highly ionizing particles

Plenty of room for innovative ML algorithms!

Enhancing the L0 Muon Trigger

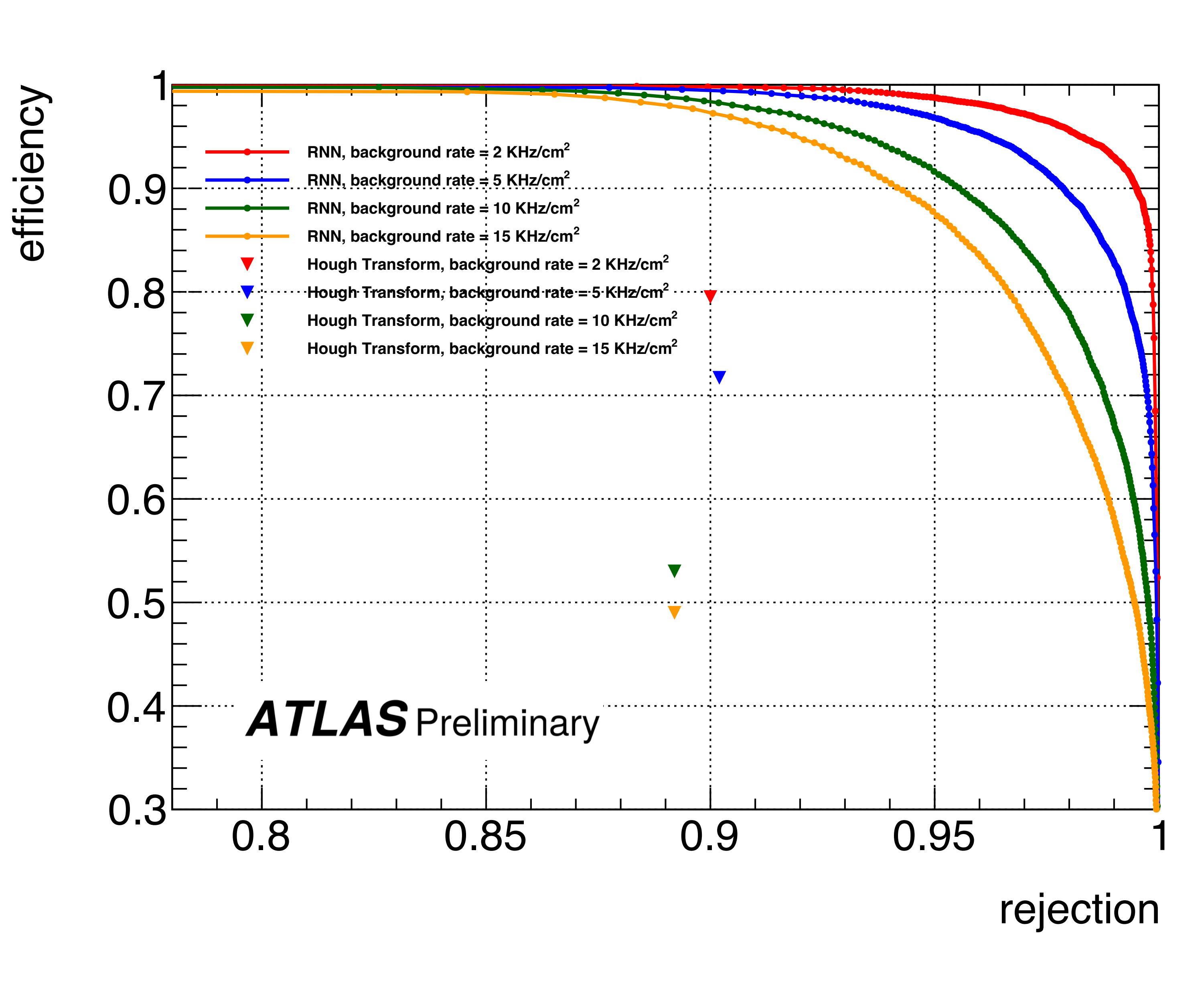
Goal is to be forward-thinking and use ML in FPGAs

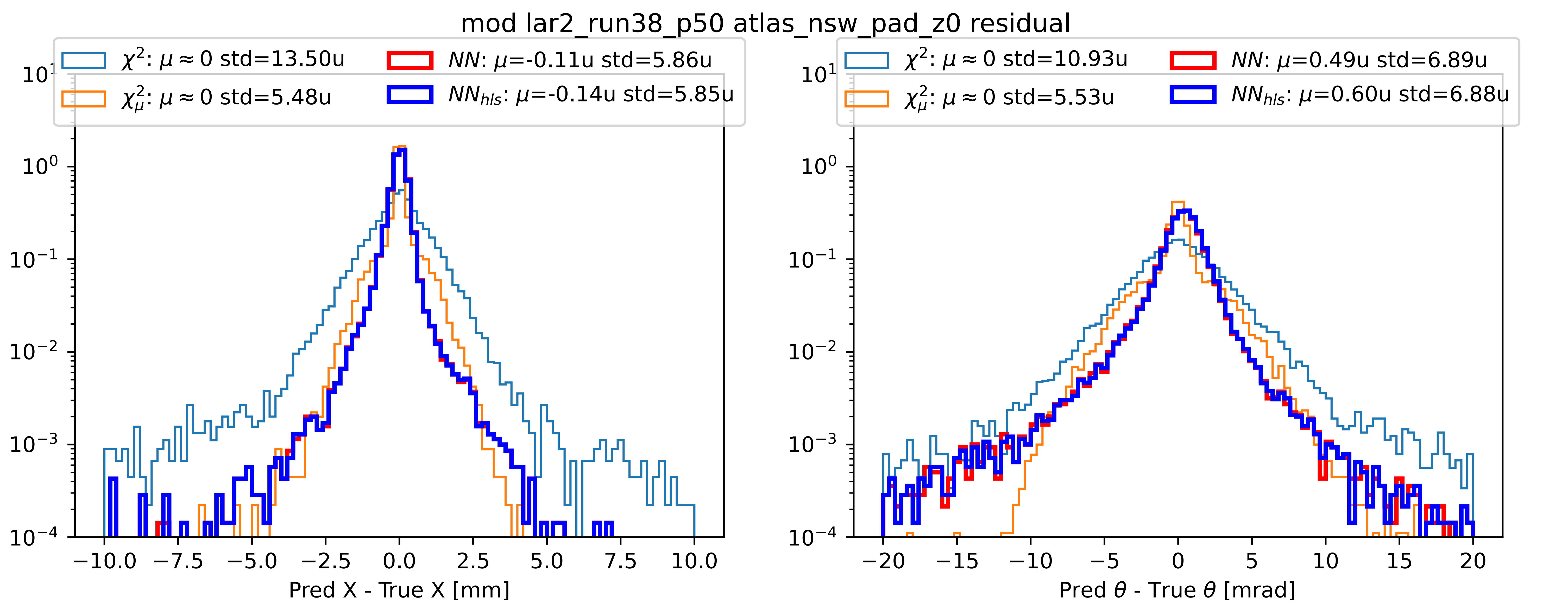
- Study different algorithms/approaches for L0 Muon triggers in case of loss of RPC performance or coverage



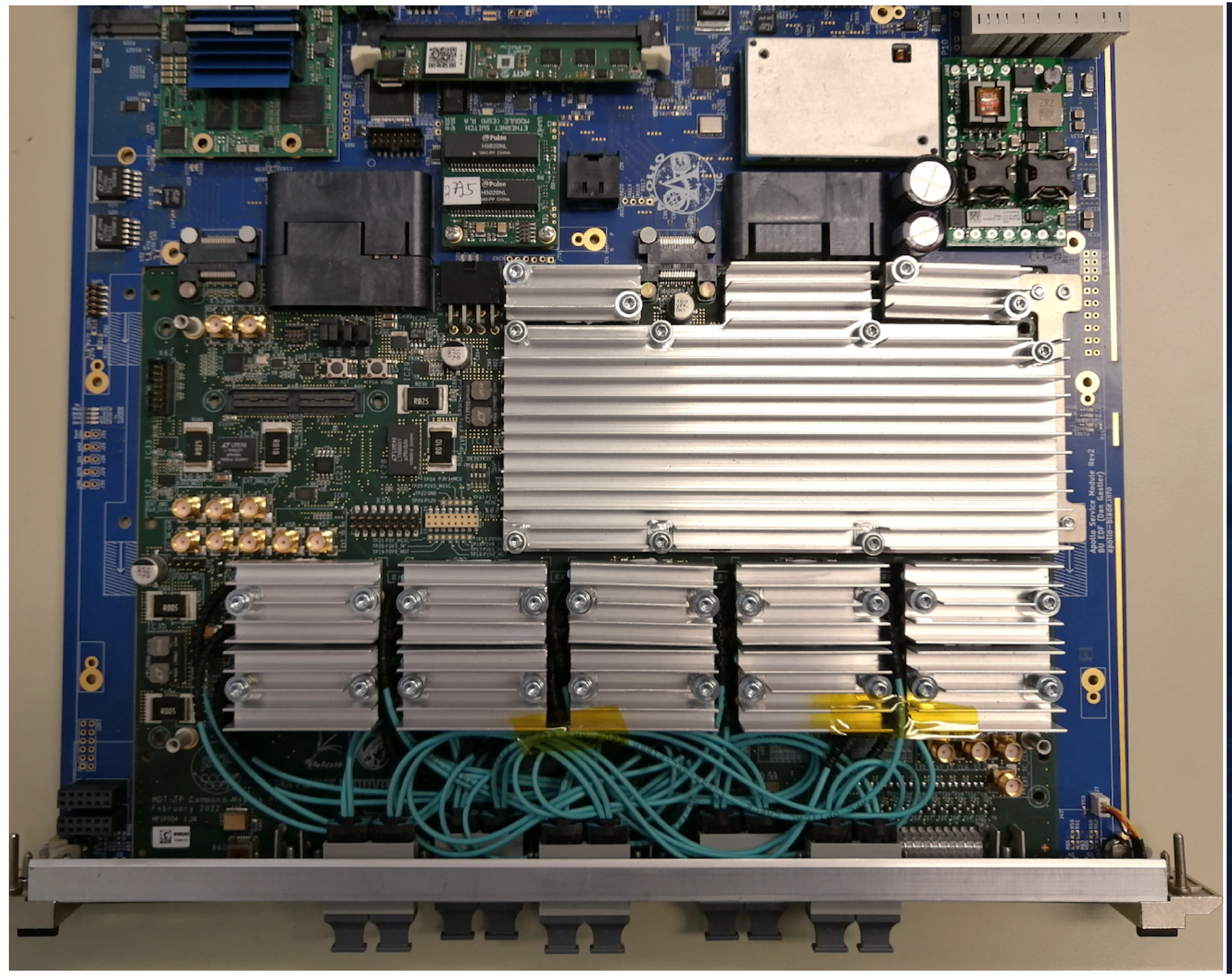
Studies on muon detectors toy model simulation for segment reconstruction show promising results

- Starting from toy model simulations based on ATLAS muon subsystem →layers of detectors identifying the crossing position of a passing muon
- For muon p_T we need to measure the particle bending ⟶ must determine both segment position and angle





Enhancing the L0 Muon Trigger

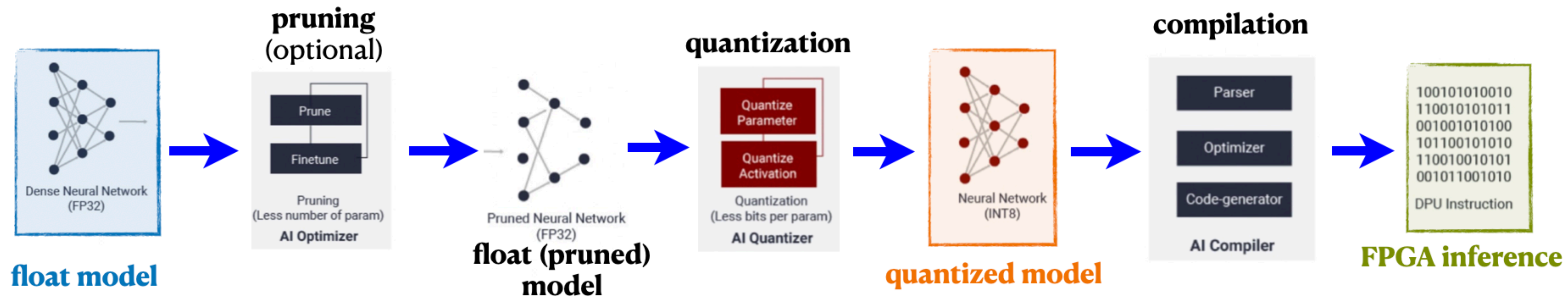


FPGA implementation

- Can target the current L0 Muon trigger hardware (Xilinx VU13P FPGA) using HLS4ML
- Explore potential improvements using different hardware

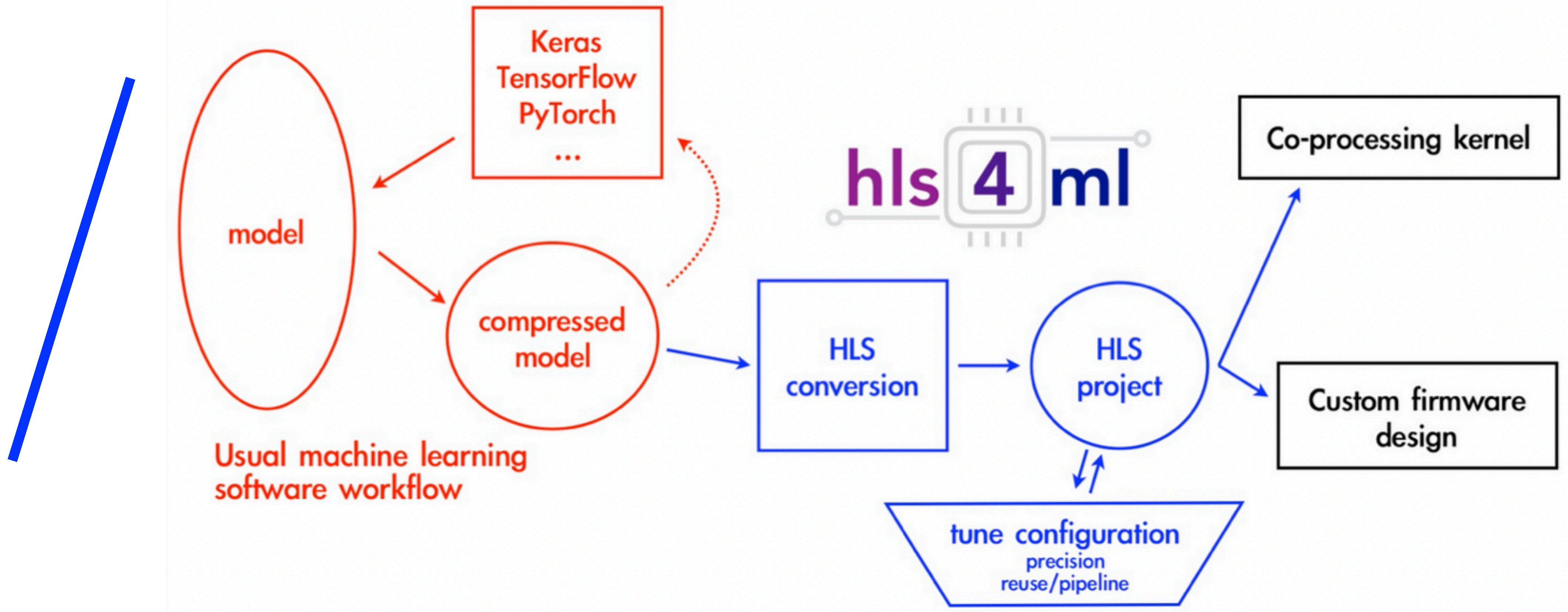
Use already existing frameworks developed for ML inference on FPGA such as:

VITIS-AI



AMD development platform for optimized deployment of deep learning models on FPGA

HLS4ML



7

Nanosecond AI for anomaly detection with decision trees on FPGA using fwXmachina

ATLAS EXPERIMENT

SMARTHEP Edge Machine Learning School

Mon, 23 Sep 2024

Ben Carlson, Isabelle Taylor, Joerg Stelzer, Kemal Ercikti, Kyle Mo, Pavel Serhiayenka, Rajat Gupta, Santiago Cane, Stephen Roche, Tae Min Hong, Yuvaraj Elangovan

WESTMONT

University of Pittsburgh

SAINT LOUIS UNIVERSITY
SCHOOL OF MEDICINE

fwX – an efficient BDT implementation on FPGAs

Framework for generating nanosecond-scale inference BDTs for use in FPGAs

Anticipated areas of use: event analysis in hardware triggers in HEP experiments

Work on

- Fast event classification with BDT (Hong et al., JINST 16, P08016 (2021))
- Fast regression with deep BDT's (*) (Carlson et al., JINST 17, P09039 (2022))
- Fast anomaly detection with BDT-based auto-encoders (*) (Roche et al., accepted for publication)

* Currently being implemented in ATLAS L1 trigger

BDTs for auto-encoders

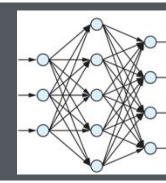
See: Govorkova et al., *Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider*, Nature Mach. Intell. **4** (2022) 154–161 https://doi.org/10.1038/s42256-022-00441-3

Typically constructed using neural networks

- Challenge to implement in pure digital logic on FPGA

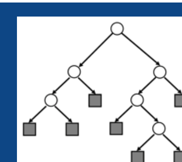
Neural Network

Popular	Been around HEP since the 80s[1]
Depth	Challenging, so ~3 on FPGA[2]
Score	$y = \Theta(\underbrace{}_{\text{Activation}}\mathbb{M} \underbrace{\cdot}_{\text{Multiplication}} x + b)$



Decision Tree

Popular	Discovered the Higgs![3]
Depth	Challenging, so 4 to 8 on FPGA[4,5,6]
Score	$y = \Theta(\underbrace{}_{\text{Step fn}}x \underbrace{<}_{\text{Comparison}} \text{threshold})$



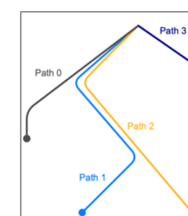
Classification performance of BDTs is often comparable

Advantages of BDT

- Technical (no multiplication)
- Philosophical (interpretable)

FWX approach:

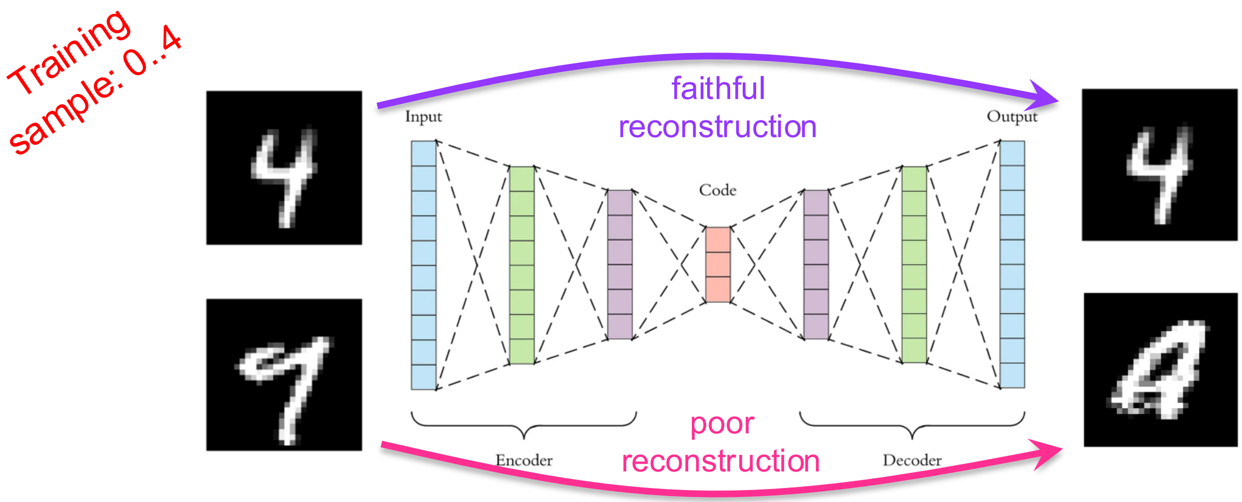
- Goal: make evaluation of the BDT in FPGA faster while using less resources
- Achieved by parallelizing node evaluation



Auto-encoders for anomaly detection

Auto-encoders rely on data-compression algorithm (usually NN, fwX: BDT), trained on known, expected data (background)

Encoding input into latent ("code"-) space and decoding back into input space preserves objects which are similar to training sample (known data), but fails to faithfully re-construct anomalies (unknown data)



Poor reconstruction

large discrepancy between input and output => high anomaly score

Our approach to using BDTs for auto-encoders

Novel algorithm for using decision trees in auto-encoders for anomaly detection

- Anomaly score from comparison of input with latent space, no decoding step

Method: (a glimpse)

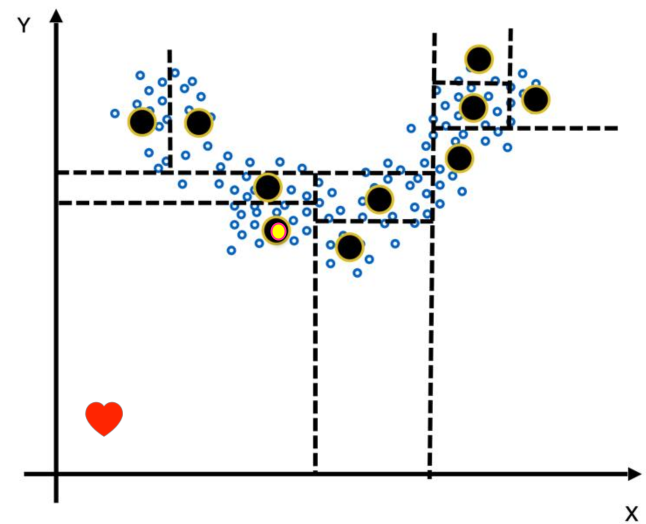
Place small boxes around locations of high event density

Encoding an event ♥:

- Return *index b* of the box the event ♥ falls into

Decoding a box index *b*:

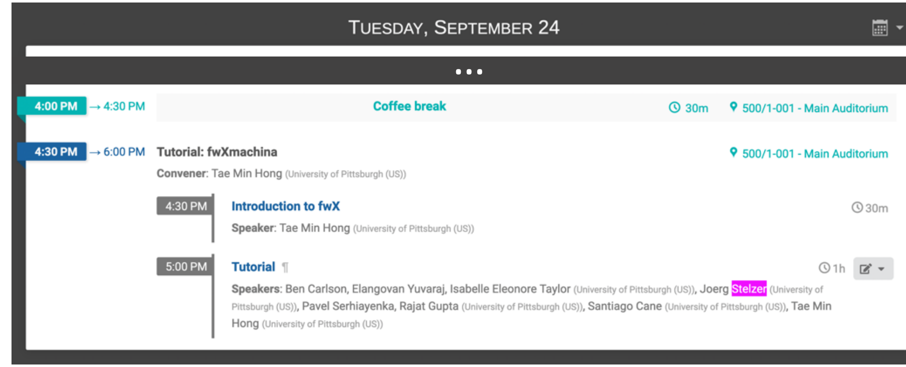
- Return the median ● of the training data in box *b*



Want to learn more – join us tomorrow

In-depth introduction to anomaly detection with FWX by Tae tomorrow afternoon @16:30.

Followed by a hands-on tutorial



Tutorial with three parts

- Training and fwX-BDT code generation (with TMVA and FWX)
- Synthesis (with Vivado)
- FPGA evaluation (simulation with Vivado)

Each part has a 10' video (where you can work along), followed by a Q&A session

- If you like to follow the tutorial on your laptop, please make sure you have root, fwX (part 1) and vivado (parts 2+3) installed

8

Neural Architectures and Data Processing Pipelines for Irradiation Experiments:

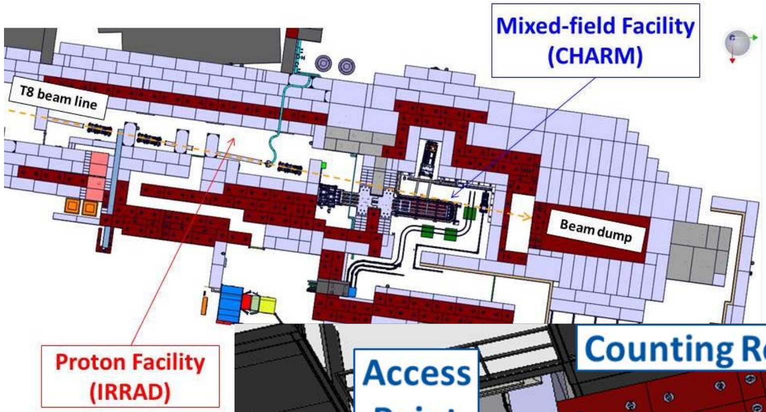
from the Automatic Assessment of Proposals to the Monitoring of the Beam Quality

Jarosław Szumega
CERN EP-DT-DD,
Mines Paris – PSL

on behalf of the team:
Jaroslaw Szumega, Lamine Bougueroua, Blerina Gkotse, Pierre Jouvelot, Federico Ravotti

CERN | IRRAD Proton Facility

1. Introduction to IRRAD facility



IRRAD Proton Facility

- Located on T8 beamline – a 24 GeV/c proton beam extracted from PS ring.

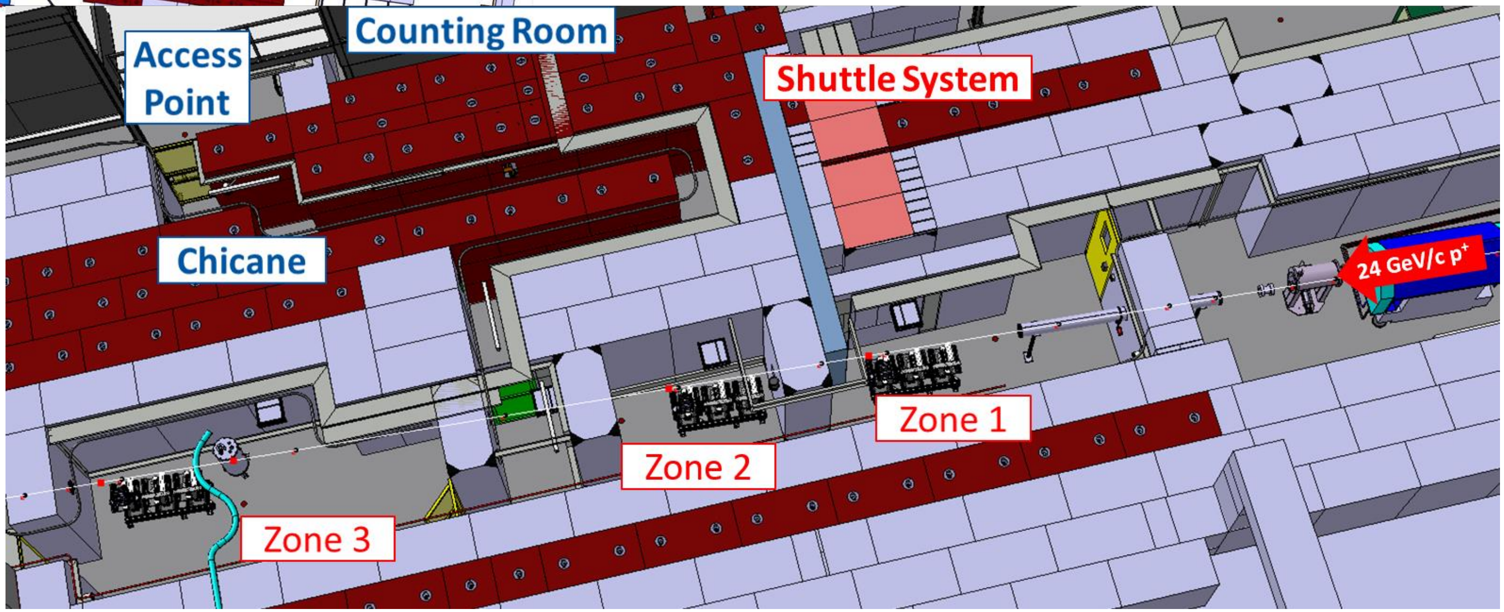


Fig. 1. The location and layout of the IRRAD facility. Divided into three zones and equipped with a shuttle system, it is a place for electronic qualification and radiation hardness assessment.

2. Automatic Assessment of Experimental Proposals

Goal

- Support to facility users – to prepare better experiments
- Support to User Selection Panels – to prepare better reviews

Simple goal – yet lots of challenges

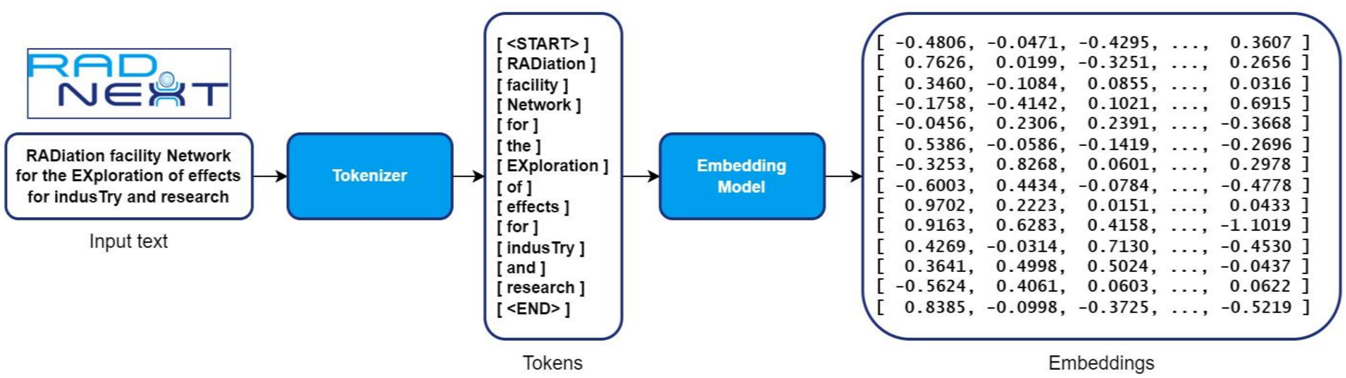


Fig. 2. An illustration of embeddings creation of a short text. The result is a real vector obtained with the transformer architecture.

2. Automatic Assessment of Experimental Proposals

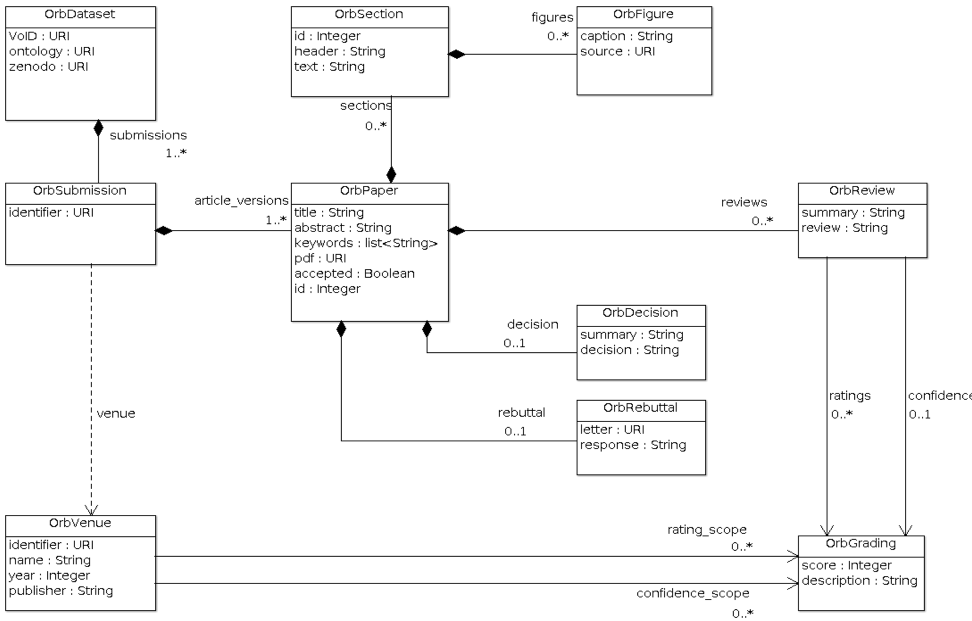


Fig. 3. The UML diagram presents the architecture of the ORB dataset. It is the third iteration involving resources like OpenReview, Sci-Post and PeerJ.

Values of MAE (Mean Absolute Error) for the final and confidence scores and their variances

Score error	Score variance error	Confidence error	Conf. variance error
0.87	0.78	0.40	0.30

3. Transverse Beam Profile Monitoring

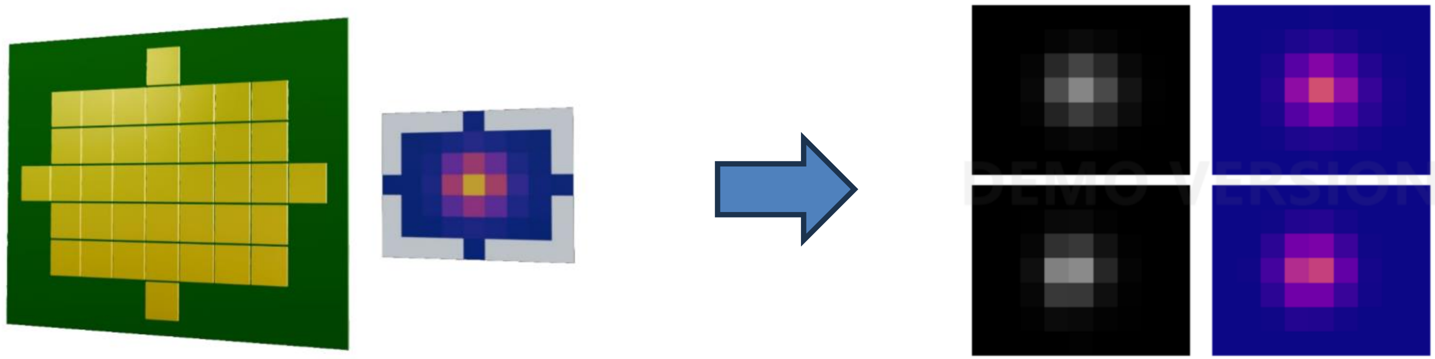
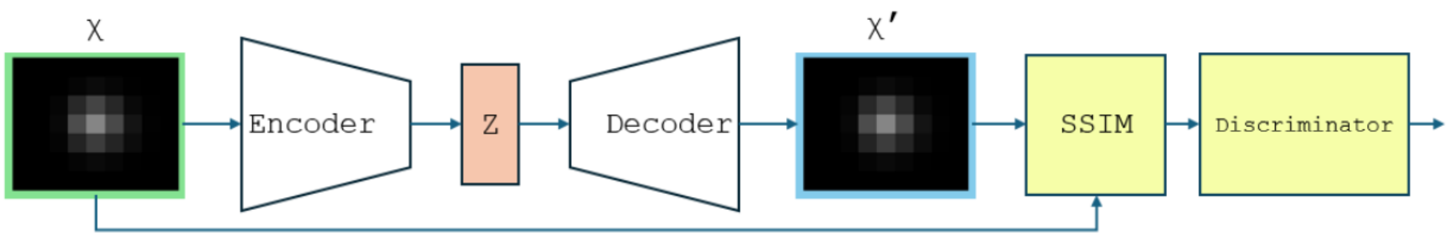


Fig. 4. New BPM DAQ (Data Acquisition) electronics is used to monitor the beam profile. The existing data was used to create custom dataset for anomaly detection.



94.5%

+offline with 16x speed-up

Fig. 5. A Convolutional Autoencoder with SSIM (Structural Similarity Index Measure) metric provides the foundation for real-time anomaly detection - an off-centred beam.
One problem is that a „good" profile is sometimes mistaken for an off-centred.

Acknowledgements

www.radnext.web.cern.ch

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101008126.

www.web.infn.it/EURO-LABS

This project has received funding from the European Union's Horizon Europe Research and Innovation programme under grant agreement No 101057511.

This project received support in the form of hardware resources and computation time in the framework of the NVIDIA Corporation Cloud GPU Grant program. The access to GPU computation cluster was provided by the Saturn Cloud platform.