Computing challenges activities from Oviedo

S. Folgueras on behalf of the Oviedo team

Funded by the European Union

European Research Council
Established by the European Commission

INTREPID

Universidad de Oviedo

A word about the INTREPID project

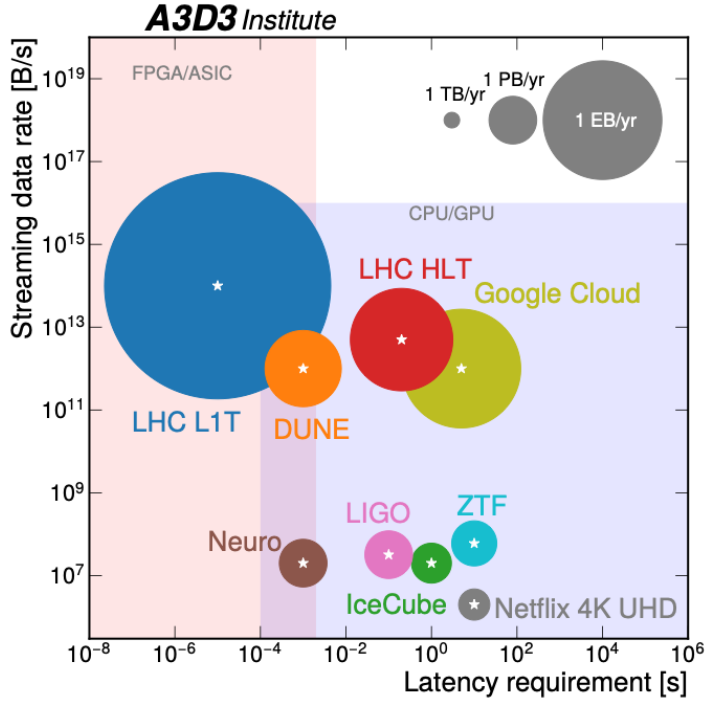
INnovative**TR**igg**E**r techniques for beyond the standard model **P**hys**I**cs**D**iscovery at the LHC

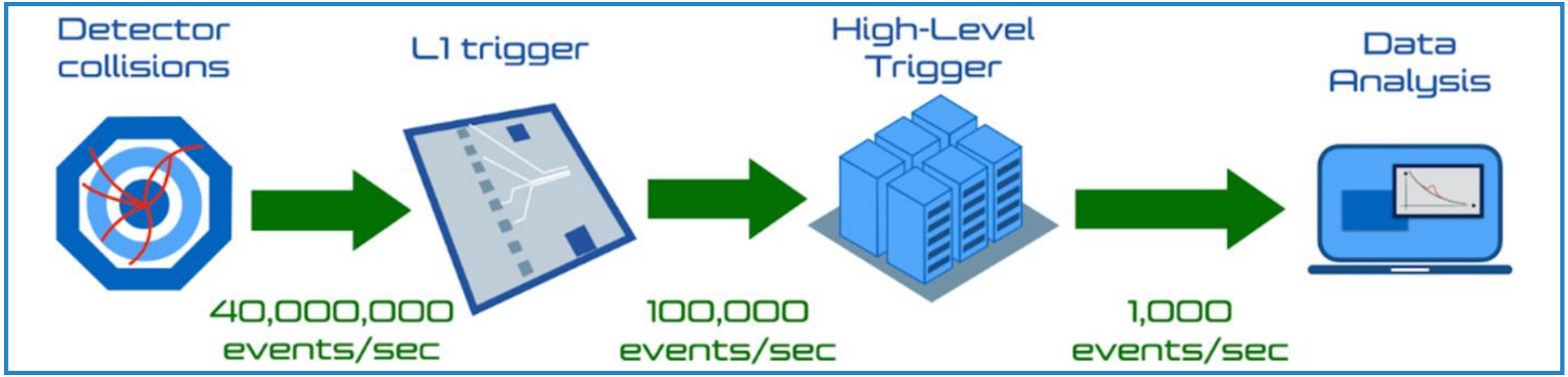
- Got funding from the ERC to explore alternative technologies and ideas which could not be otherwise investigated that could potentially lead to a significant breakthrough.
- **Project focuses on muons signatures**, but ideas can be ported elsewhere and targets to expand the sensitivity to long-lived / displaced signatures.
- For the **HL-LHC and beyond**:
  - Muon trigger for the HL-LHC (in collaboration with CIEMAT): expand existing low-latency algorithms to detect and trigger on non-standard signatures such as hadronic showers or slow charged particles.
  - GNN for real-time (O(μs)) muon reconstruction, using Versal ACAP and the AI engines.
- Hardware/Setup:
  - Setting up a local test stand for benchmarking firmware.
  - Various devices available for testing: Xilinx Kintex, Virtex Ultrascale+, Versal 7nm AI.
- Person power:
  - 3 PhD students (full time), 1 postdoc (full time), 2-3 senior (0.5 FTE each)

Part of our local team, setting up the test bench.

Introduction: how a trigger system works?

- Data explosion and AI applications: our world needs **higher throughput** and **real-time computing** capabilities
- LHC provides ideal benchmark to explore real-time data processing technologies
- Only a handful of the collisions contain **interesting physics**
- **Trigger system** decides, in **real time** if a collision is saved or lost forever

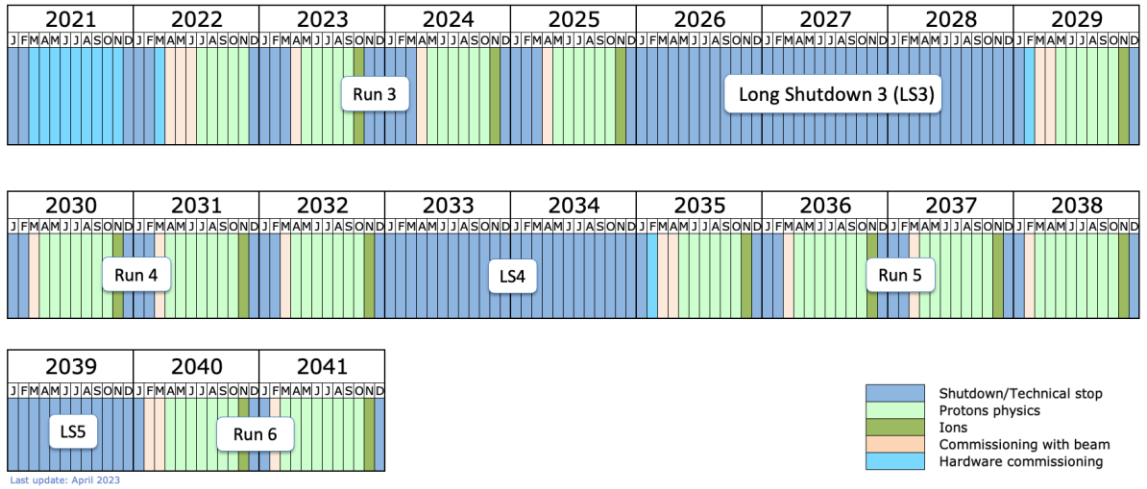


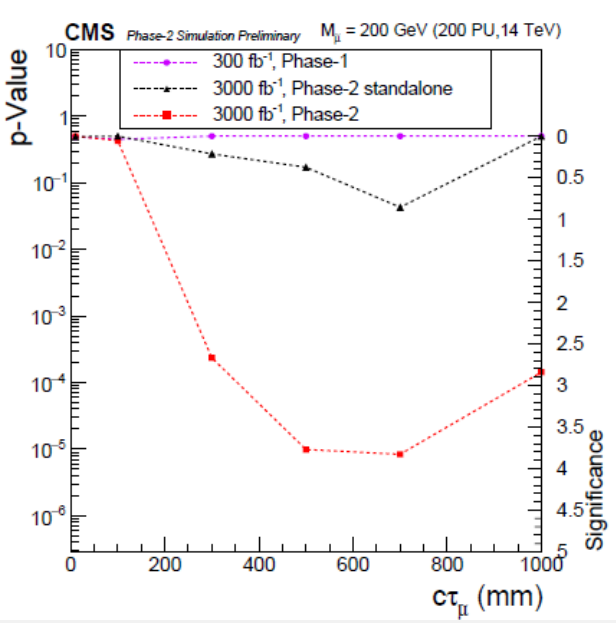


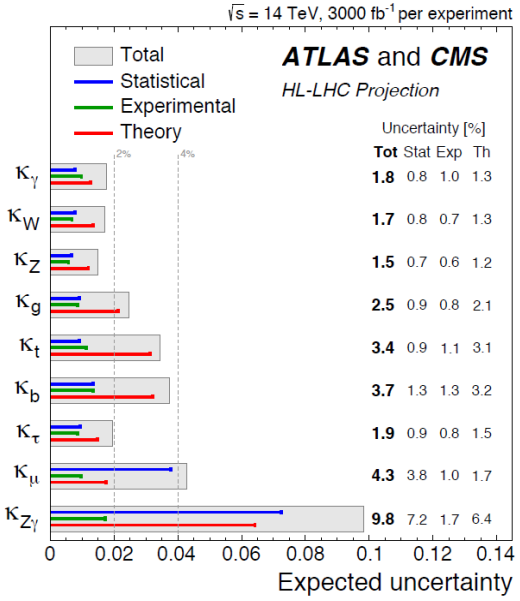
[CERN-LPCC-2019-01]

Towards the HL-LHC

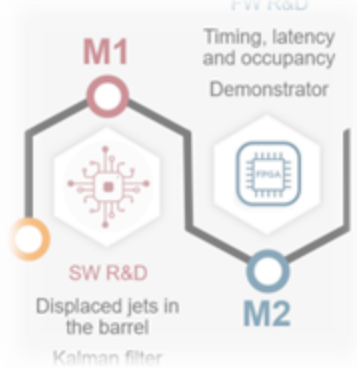
- **Preparing for the big upgrade** of the LHC detectors, starting 2029.
- HL-LHC upgrade offers an **unprecedented opportunity** to explore uncharted lands and achieve scientific progress.
- 10 times more data to what we will have by the end of Run 3 will facilitate a rich physics program.
- **Extend reach of new physics searches**: unexplored signatures (LLPs, HSCPs…) or regions of the phase-space will be within reach.
- **Improve current understanding of the SM and Higgs** sector by improving existing precision measurements and accessing rare decays ($H \to \mu\mu$) or production modes (HH) previously unseen at the LHC.
- However, this physics program will have to overcome **significant challenges** to succeed.

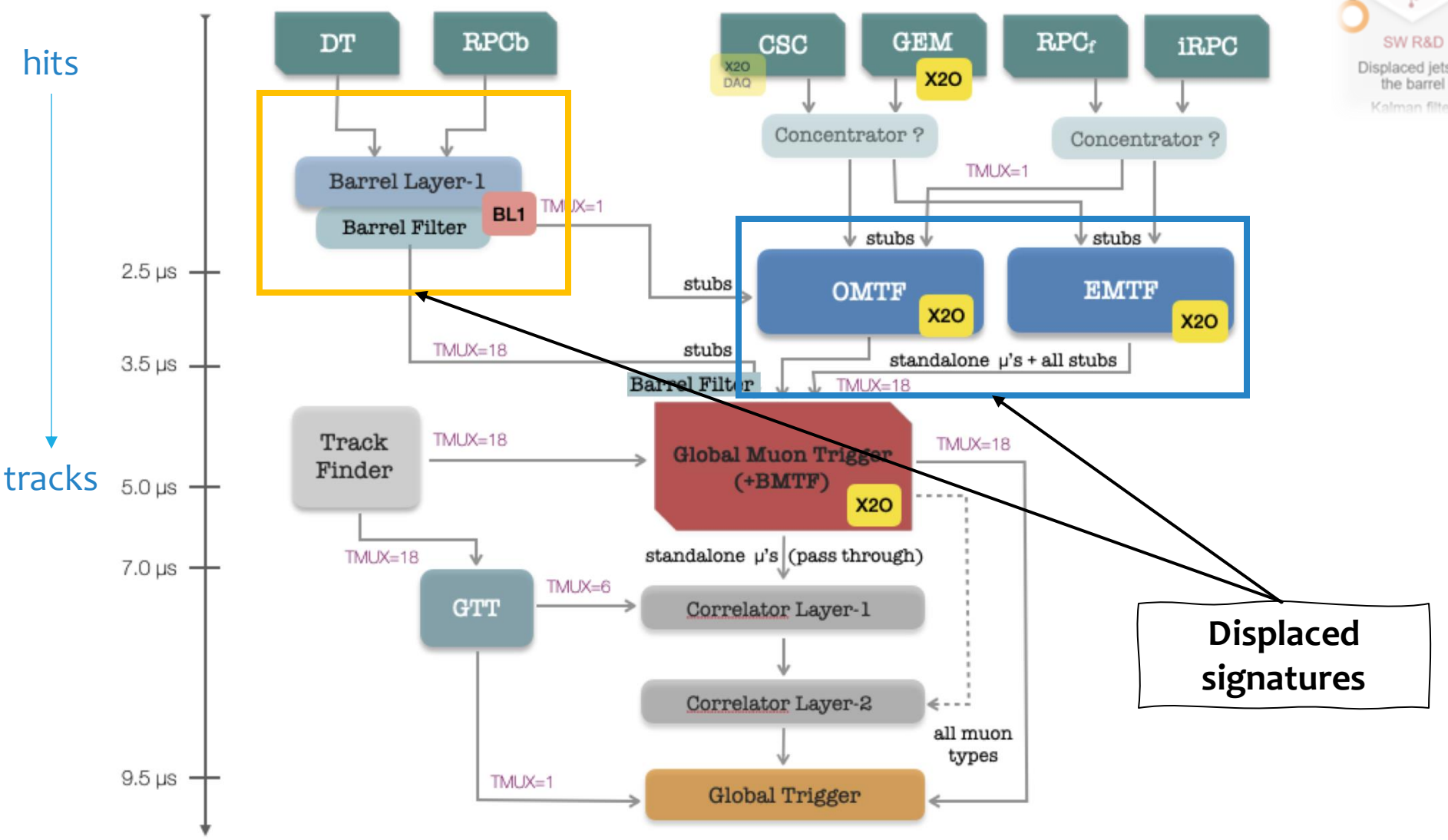






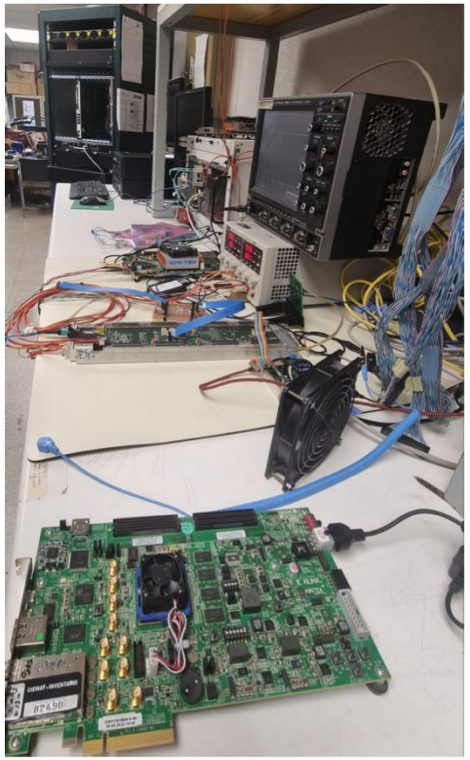
Improve muon triggers using the existing architecture

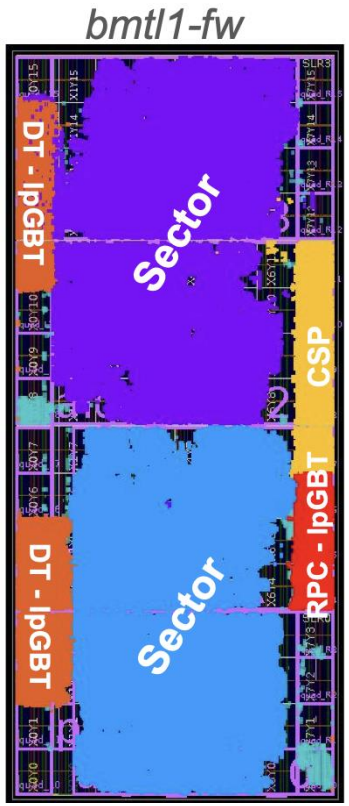


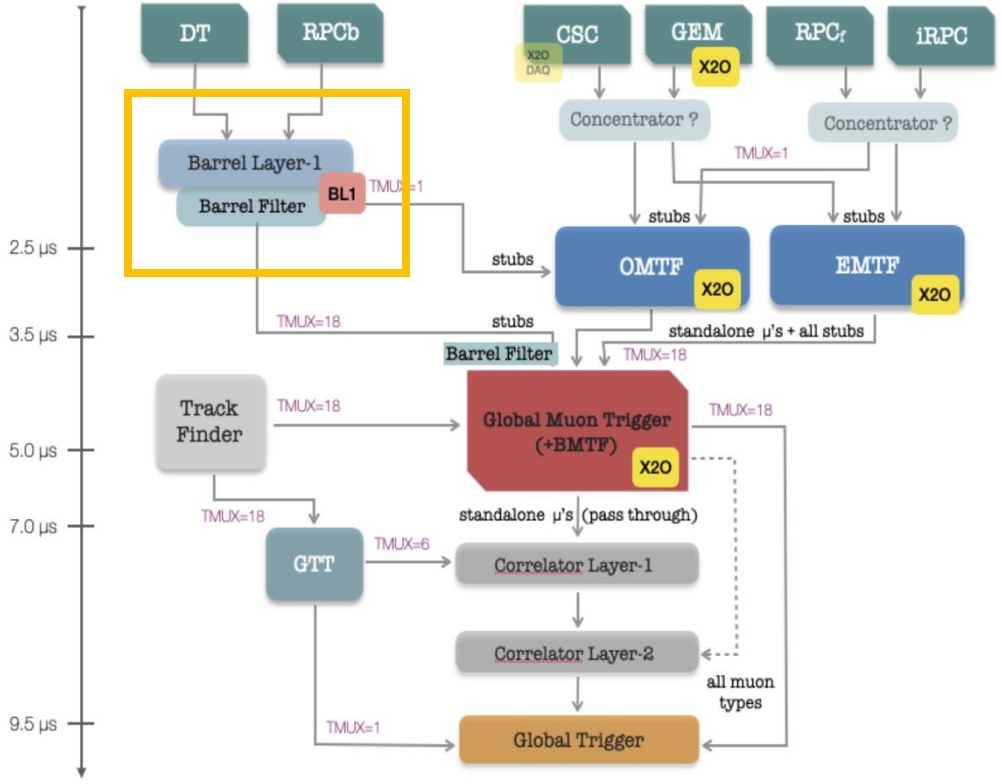


Muon trigger for the HL-LHC (CIEMAT + Oviedo)

- Algorithm and firmware design for a very low-latency trigger system for the barrel muon trigger of the CMS experiment.
  - Highly-optimized FPGA algorithms: High clock speeds (~500 MHz), minimal resource utilization.
  - Enable reconstruction for exotic signatures beyond capabilities of the current system: hadronic showers, slow-particles

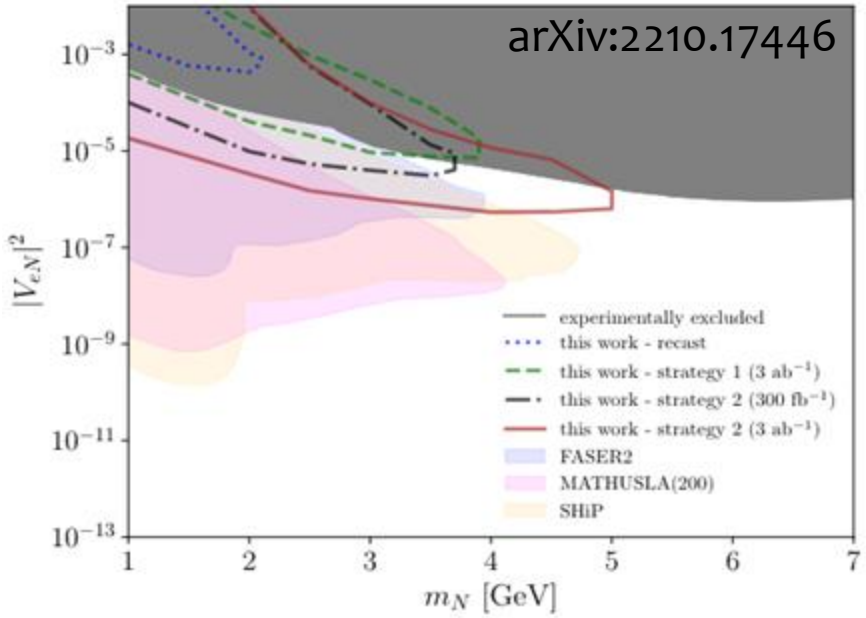


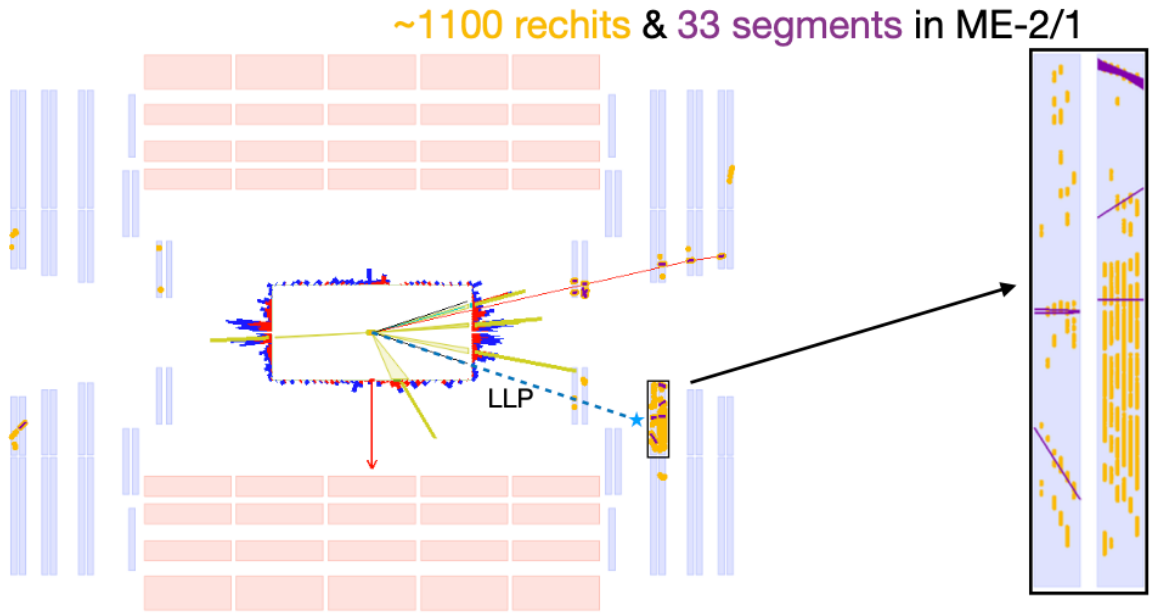




Muon showers: motivation

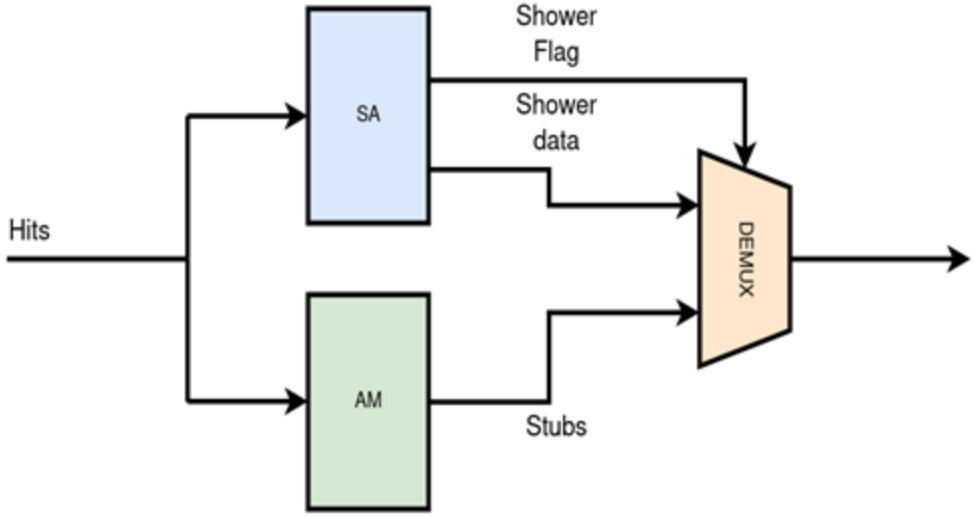
- Muon showers appear naturally in events with high-momentum muons.
  - Detected in the muon system as high-multiplicity events. Affect the reconstruction of muons: inefficiencies and/or momentum mismeasurements
- Long-lived particles such as HNL could decay in the muon system if sufficient long lifetime (or low mass)
  - Hadronic showers in the muon system. Probe lower masses (< 10 GeV) or longer lifetimes O(1m) parameter space
- Nowadays this signatures escape detection due to reconstruction and trigger constrains
- **Unprecedented opportunity for new physics discovery**

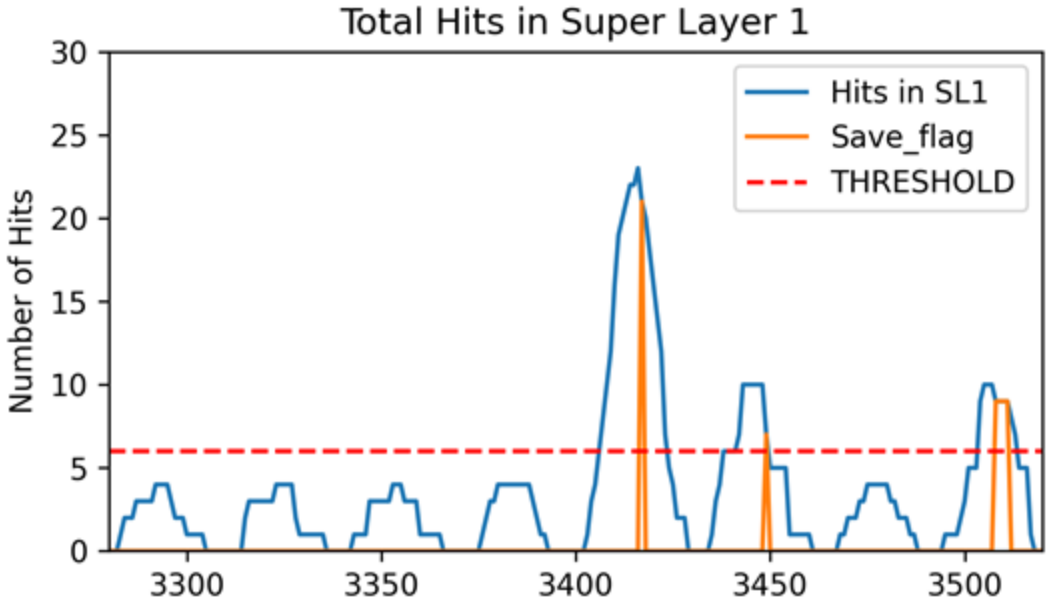




Shower identification algorithm

- Each hit is stored for 16 BXs (400ns) to account for their drift times
  - Hits are received serially
- We can represent the total amount of hits in a given superlayer over time
  - When the accumulated hits exceed a threshold, stop AM
  - When maximum detected, store hits to identify a shower (position, time,…)



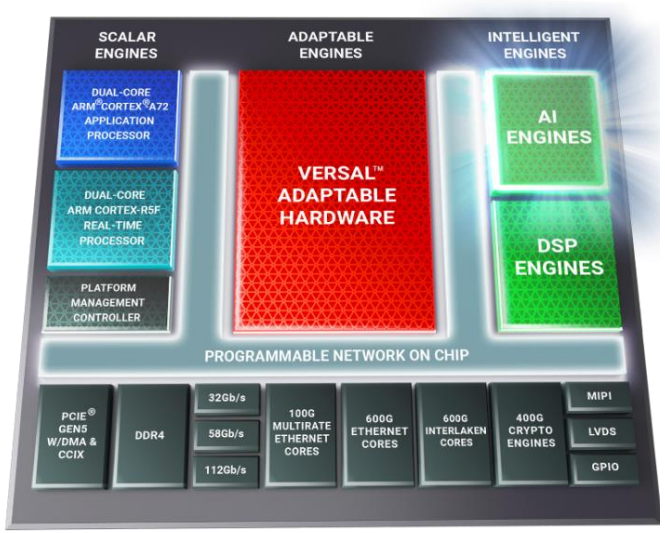


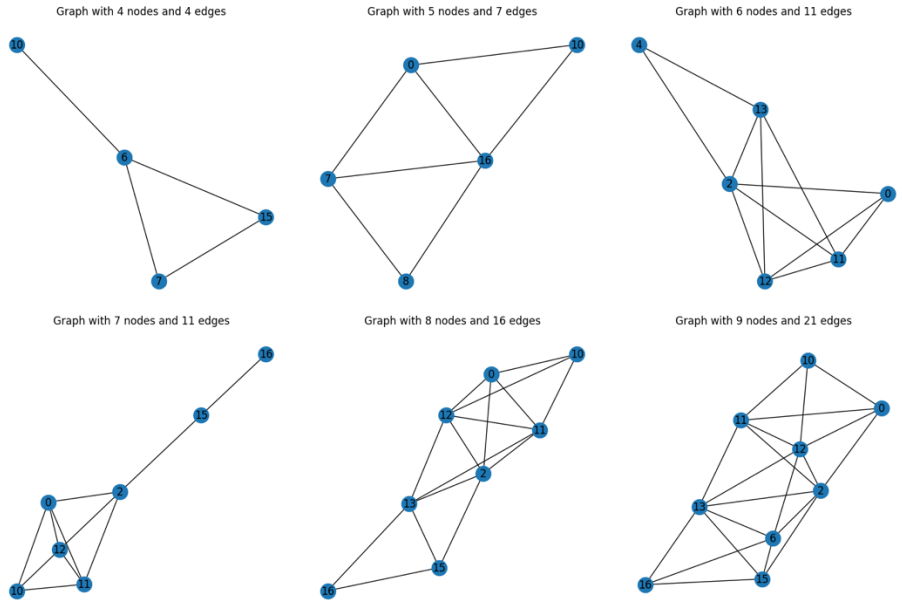
- **Future work:** Spiking networks for Shower identification (INTREPID and RYC2021-033305-I)
  - First stage: classification "shower took place" vs "shower didn't take place"
  - Second stage: build regression model to identify the most likely BX to associate the shower to

Work presented at TWEPP and ICHEP 2024.

AI-assisted particle reconstruction in ACAP COTS

- In current (foreseen) architecture LLP signals might be easily overlooked or misinterpreted in LHC data.
- Exploring advanced ML / AI – based algorithm for particle reconstruction in the CMS muon system for the next-generation trigger systems.
- Demonstrate the capabilities of the Xilinx Versal device in low-latency (μs) applications and explore their potential use in real-time particle reconstruction.





INTREPID

Funded by the European Union

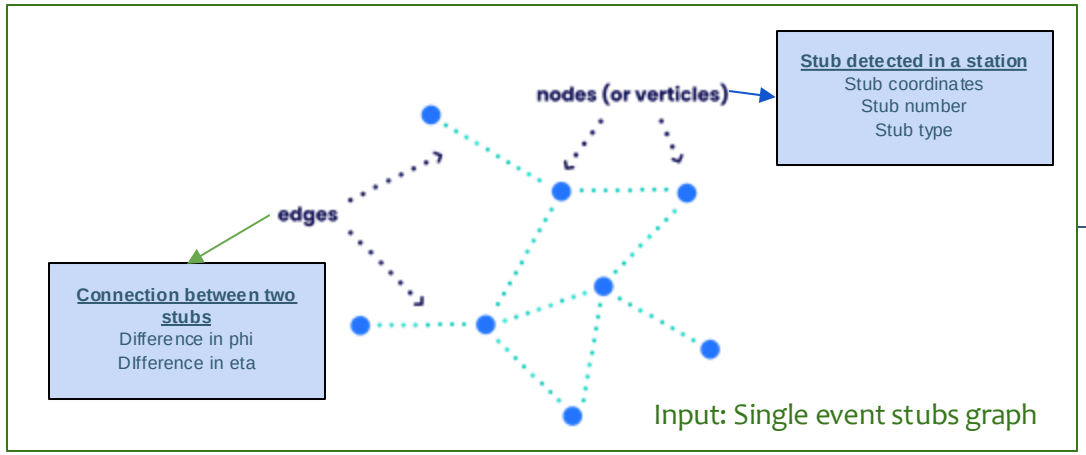
European Research Council
Established by the European Commission

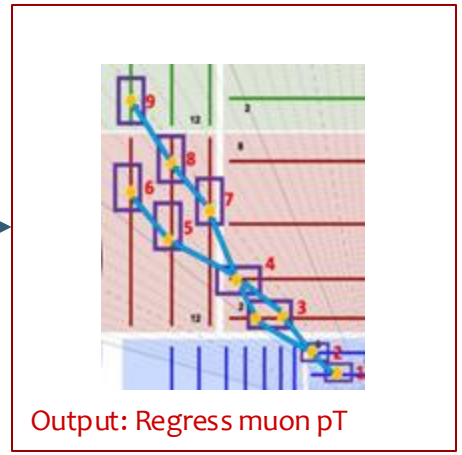
Software implementation

Designing a basic network

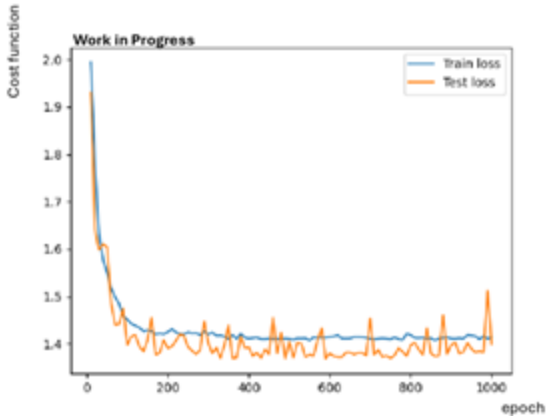
Current training:

- Muon gun phase-II sample, using only negative muons (symmetry)
- Batches of 64 graphs (events)
- Learning rate = 0.0005
- Weight decay = 0.75
- Epochs = 1000





Loss function

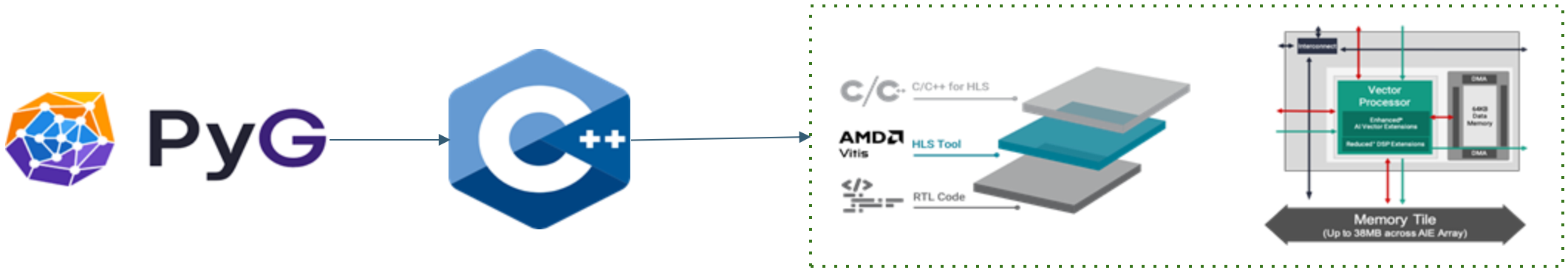


- Using fully connected graphs for the moment.
- Using pyTorch geometric libraries.
- Current architecture based on two Graph Attention Layer (GAT) [arXiv:1710.10903] to process graph data, making use of the edge information. After each GAT layer, ReLU activation is applied. The model combines global mean and average pooling to aggregate node-level features into a graph-level representation.

Hardware implementation

First steps to design with versal architecture (with and without AI cores)

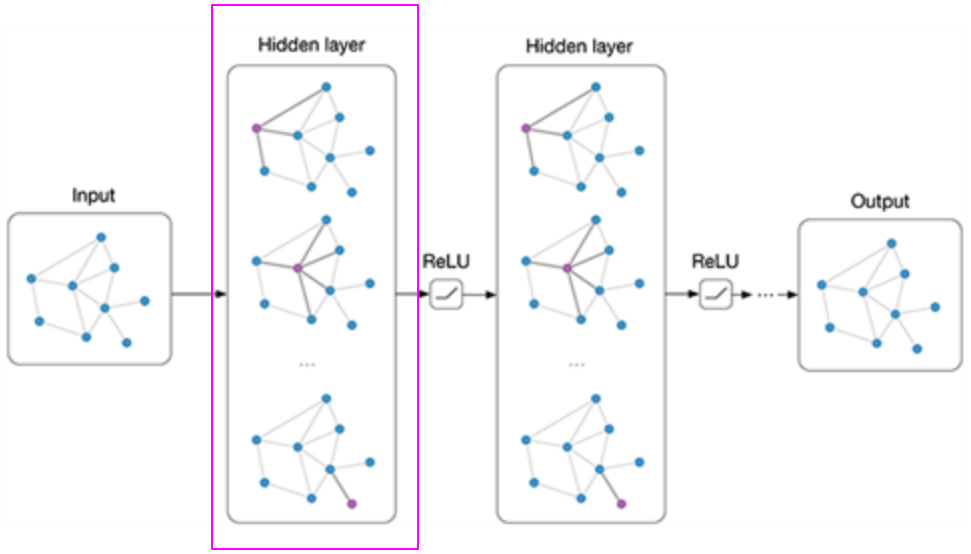
- Build specific kernels, each processing a different layer of the net.
- This would require a direct implementation on C++, to build HLS kernels.
- The high-computational cost operations could be run in the versal AI cores.
  - Implement the model in C++ using vectorized operations that can leverage the parallel computing capabilities of the cores.
  - Develop AI Engine kernels for the most compute-intensive parts of your model. The AI Engine kernels are designed to run on the AI Engines, which are optimized for high-throughput, low-latency vector processing.
  - Integrate your AI Engine kernels with the rest of the system.
  - Profile the application to identify bottlenecks



Test case

Implementing a simple GCNConv layer

- Graph Convolutional Networks extend the concept of convolution from grid-like data (like images) to graph-structured data.

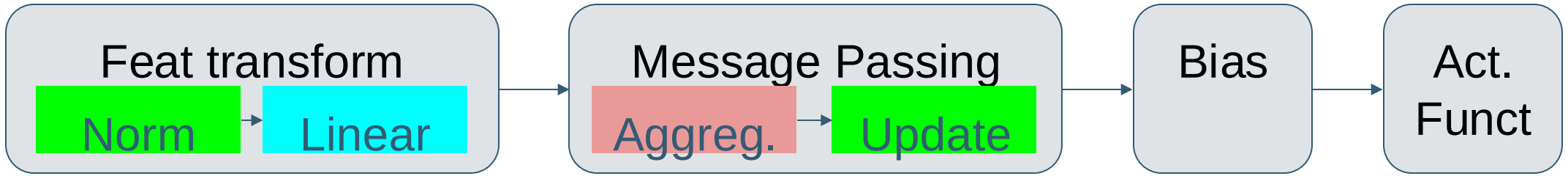


$$\mathbf{X}' = \hat{\mathbf{A}}\mathbf{X}\mathbf{W} + \mathbf{b}$$

In PyG the conv layer is called **GCNConv**

Aggregates feature information from a node's neighbors (including itself) and transforms it using learnable weights.

We dissect the layer into its main subfunctions:



$$\hat{A} = \hat{D}^{-1/2}(A + I)\hat{D}^{-1/2}$$

$$X' = XW$$

$$X''_i = \sum_{j \in \mathcal{N}(i)} m_{j \to i}$$

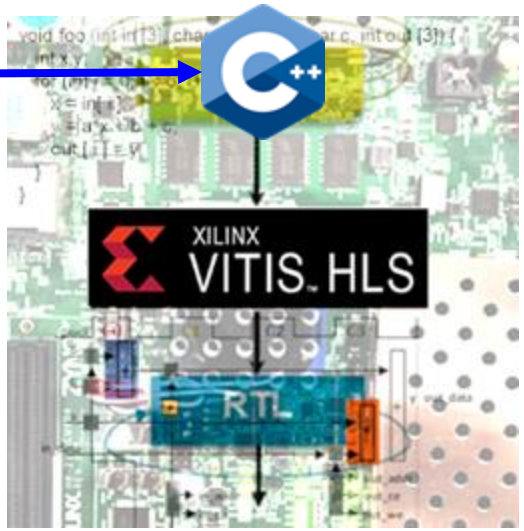
$$m_{j \to i} = \hat{A}_{ij} X'_j$$

$$X'''_i = X''_i + b$$

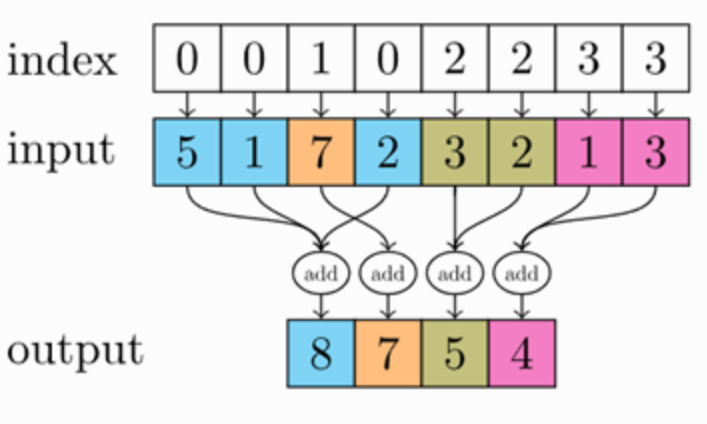
$$H_i = \sigma(X'''_i)$$

Test case

Implementing a simple GCNConv layer



- We need C++ to import the layer into our device w/HLS!!



Aggregate example

$$m_{j \to i} = \hat{A}_{ij} X'_j$$

```python
def aggregate(
    self,
    inputs: Tensor,
    index: Tensor,
    ptr: Optional[Tensor] = None,
    dim_size: Optional[int] = None,
) -> Tensor:
```

```cpp
using Features = std::vector<std::vector<float>>;
```

```cpp
virtual Features aggregate(const Features &messages, const EdgeList &edge_index, int num_nodes)
{
    // Initialize a vector to store the aggregated messages for each node
    Features aggregated_messages(num_nodes, std::vector<float>(messages[0].size(), 0.0f));

    for (size_t i = 0; i < edge_index.size(); ++i)
    {
        int target = edge_index[i].second; // Target node index

        // Aggregate the message to the target node
        for (size_t j = 0; j < messages[i].size(); ++j)
        {
            aggregated_messages[target][j] += messages[i][j];
        }
    }

    // Optionally, implement other aggregation types (mean, max, etc.) based on your model's requirements

    return aggregated_messages;
}
```

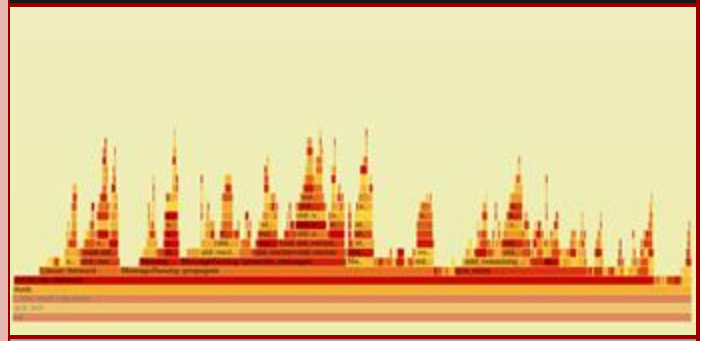
torch_geom_doc

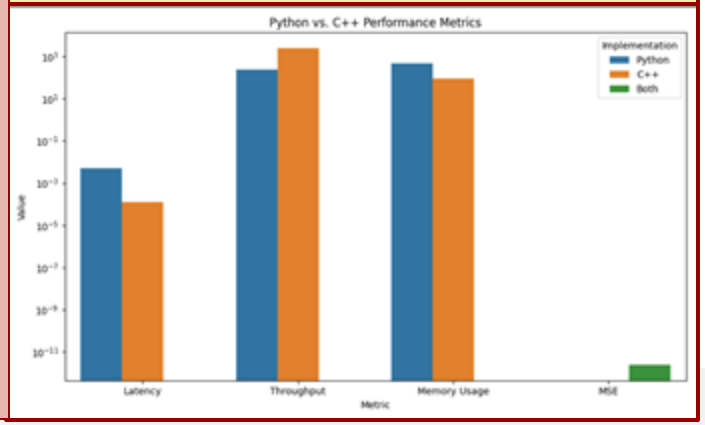
Test case

Implementing a simple GCNConv layer

Profiling

```
=== Profiling Summary ===
gcn_norm Total Time: 0.00556605 seconds
linear.forward Total Time: 0.017838 seconds
generate_messages Total Time: 0.00619323 seconds
aggregate Total Time: 0.00248693 seconds
update Total Time: 0.00160153 seconds
propagate Total Time: 0.143146 seconds over 1001 runs
propagate Average Time: 0.000143003 seconds
forward Total Time: 0.171707 seconds over 1001 runs
forward Average Time: 0.000171536 seconds
========================
```

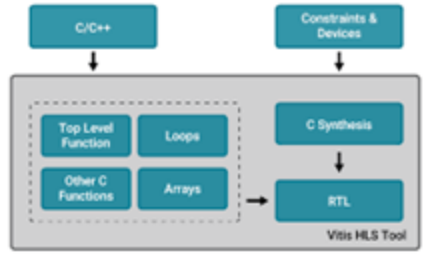




- After converting all the main elements, we test the performance of the layer and do profiling, so we can check the metrics and see what are the high-computational cost zones.

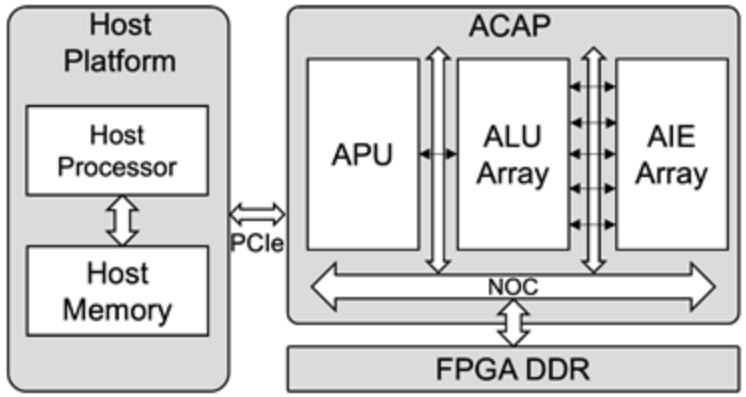
Then we build the HLS-like version

```
class MessagePassingHLS
{
public:
    // Top-level propagate function
    void propagate(
        const EdgeFeature edge_index[NUM_EDGES],
        const feature_t x[NUM_NODES][FEATURE_SIZE],
        const feature_t edge_weight[NUM_EDGES],
        feature_t updated_features[NUM_NODES][FEATURE_SIZE]
    );
```



Test case : next steps

Implementing a simple GCNConv layer

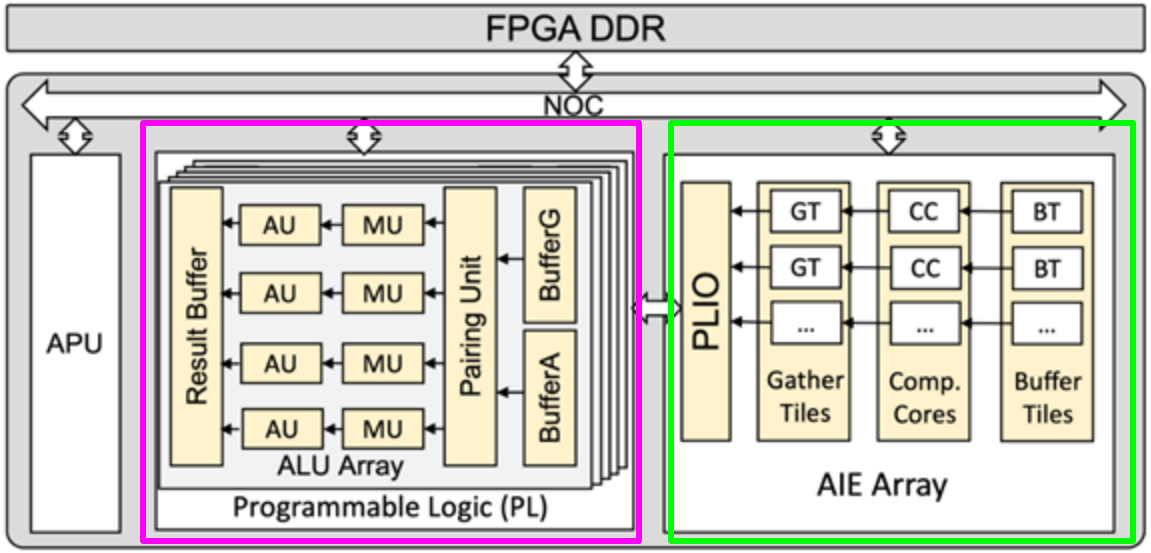


Implement the kernel cores into the versal device.

Experiment with different architectures for the matrix computations.

Matrix computation in the PL.

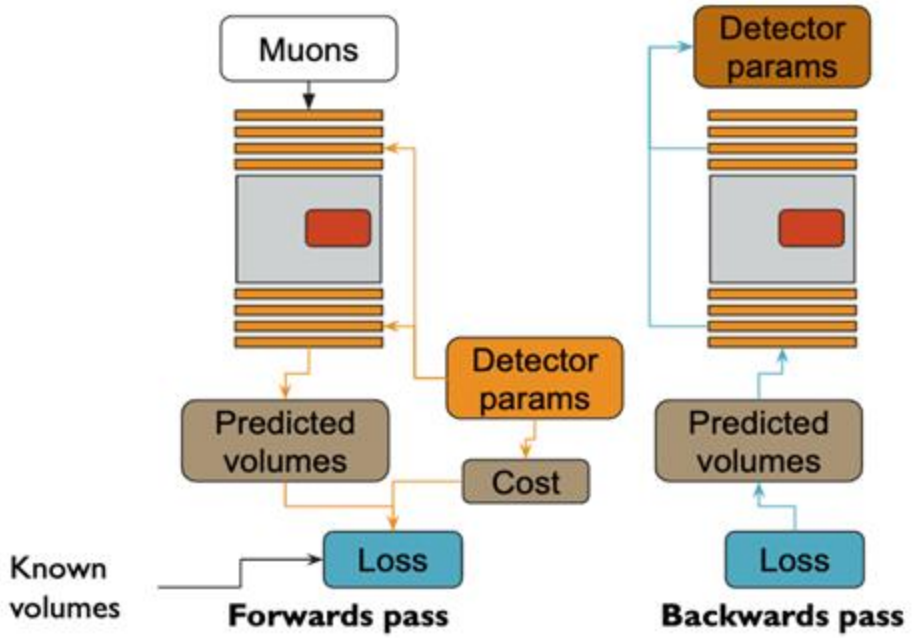
Matrix computation using AIE cores.

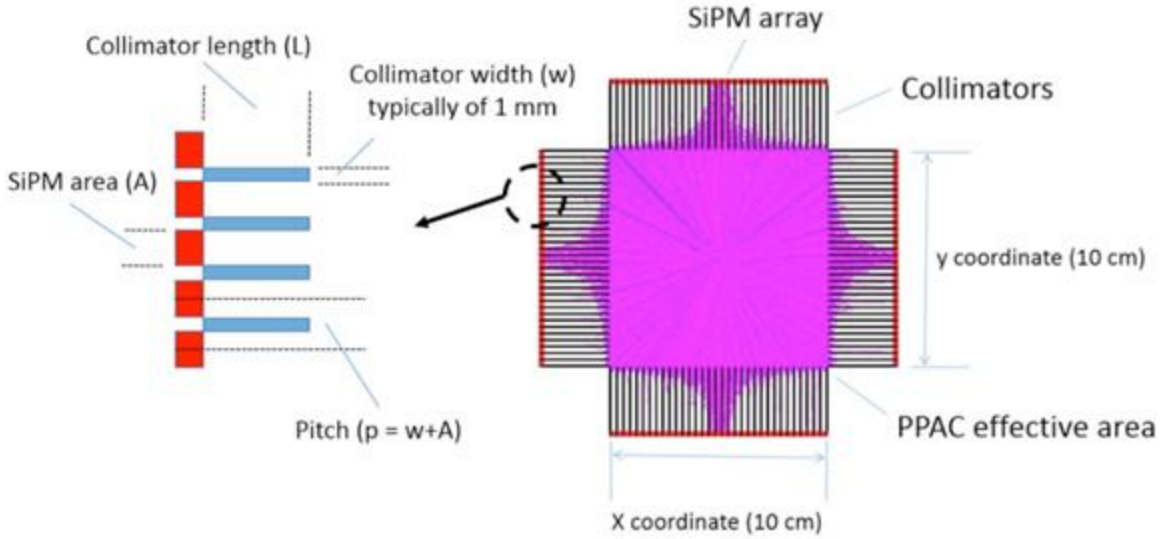


- The main idea is to develop an hybrid system that allows us to achieve maximum performance for the GNN inference.

MODE (ML-optimized design of experiments)

- Strong contribution from Oviedo (RYC2021-033305-I) also at coordination level https://mode-collaboration.github.io/
- Program of MODE: Reviews in Physics 10 (2023) 100085
  - Update (nearing publication) 2310.05673
- Optimize Muon Tomography design Mach. Learn.: Sci. Technol. 5 035002
- (ongoing) Optimize Neutron Tomography design (w/ María Pereira Martínez and Xabier Cid, USC)

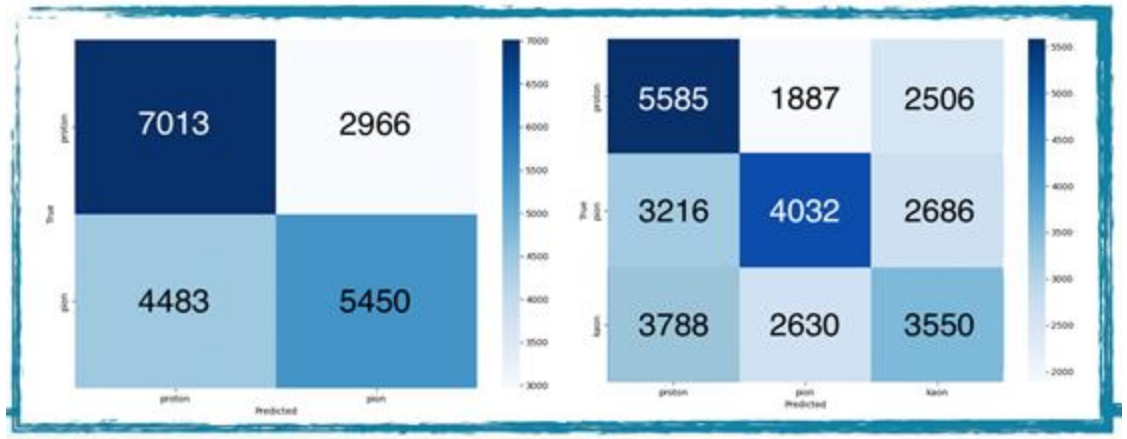




Calorimetry with Neuromorphic Computing

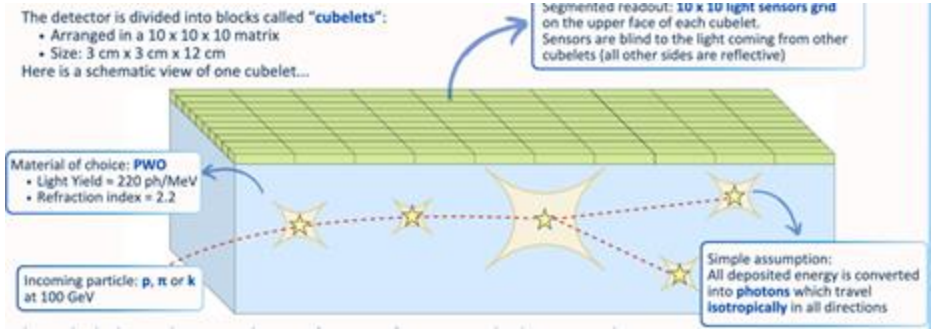
Starting work in Oviedo (MODE, RYC2021-033305-I, also within DRD6)

- Particle ID at high granularity
  - Towards integrated tracking-calorimeter ID
  - Classification based on shower properties

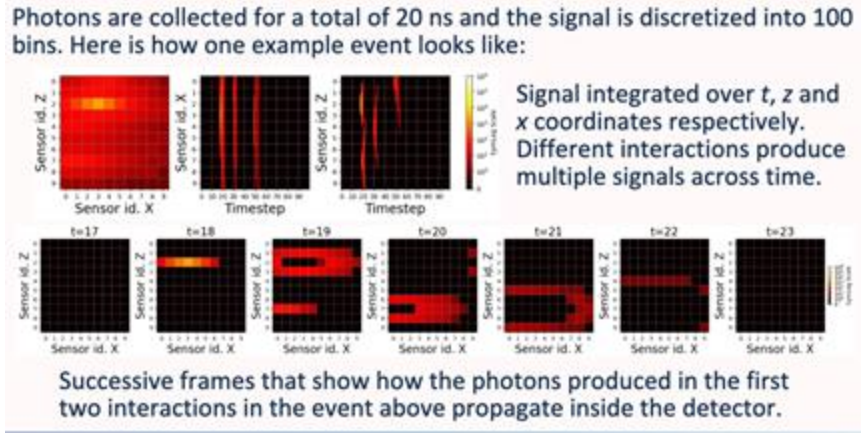


Figures by Andrea de Vita and Enrico Lupi

- Neuromorphic readout via network of nanowires
  - Fast, energy-efficient local computation
  - Generate informative high-level primitives



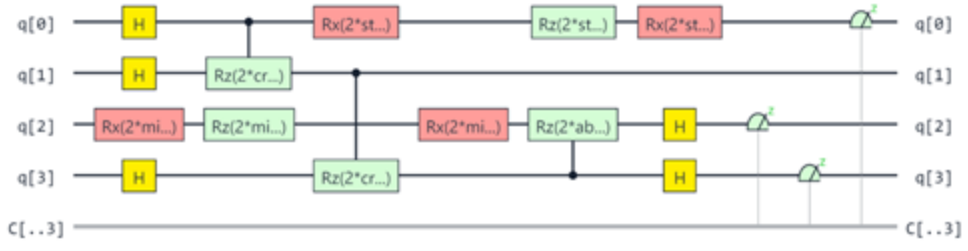
- Time evolution accessible via spiking networks

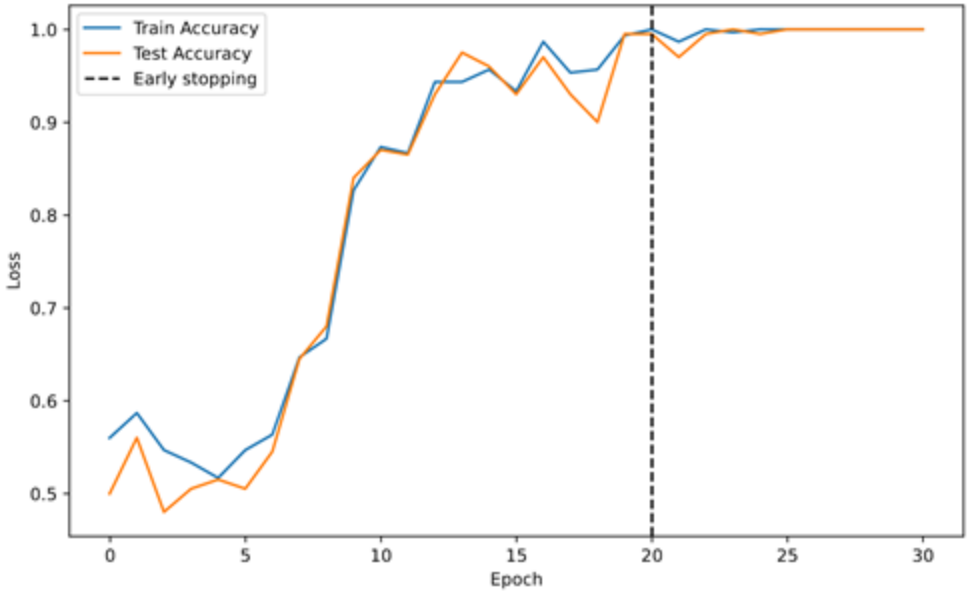


Universidad de Oviedo

Quantum Machine Learning

State-of-the art performance (w.r.t. classical ML) with orders of magnitude less training data





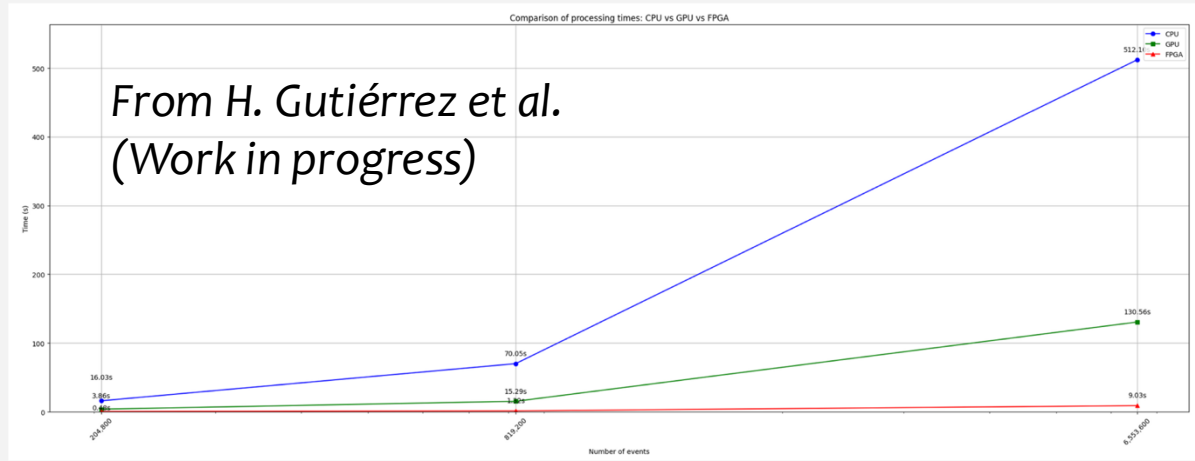
- Ongoing work in Oviedo (RYC2021-033305-I):
  - Application to physics-informed networks
  - Application to experiment design, exploiting differentiability of quantum circuits

$$
\begin{aligned}
\partial_\mu \langle\psi(x,\theta)|\sigma_z|\psi(x,\theta)\rangle &= \langle 0|\ldots\partial_\mu e^{-i\mu\sigma}\ldots\sigma_z\ldots e^{i\mu\sigma}\ldots|0\rangle \\
&\quad + \langle 0|\ldots e^{-i\mu\sigma}\ldots\sigma_z\ldots\partial_\mu e^{i\mu\sigma}\ldots|0\rangle \\
&= \langle 0|\ldots(-i\sigma)e^{-i\mu\sigma}\ldots\sigma_z\ldots e^{i\mu\sigma}\ldots|0\rangle \\
&\quad + \langle 0|\ldots e^{-i\mu\sigma}\ldots\sigma_z\ldots(i\sigma)e^{i\mu\sigma}\ldots|0\rangle \\
&= \langle 0|\ldots(1-i\sigma)e^{-i\mu\sigma}\ldots\sigma_z\ldots(1+i\sigma)e^{i\mu\sigma}\ldots|0\rangle \\
&\quad + \langle 0|\ldots(1+i\sigma)e^{-i\mu\sigma}\ldots\sigma_z\ldots(1-i\sigma)e^{i\mu\sigma}\ldots|0\rangle
\end{aligned}
$$

Figures: M. Uría García, J.M. Uría, P. Vischia

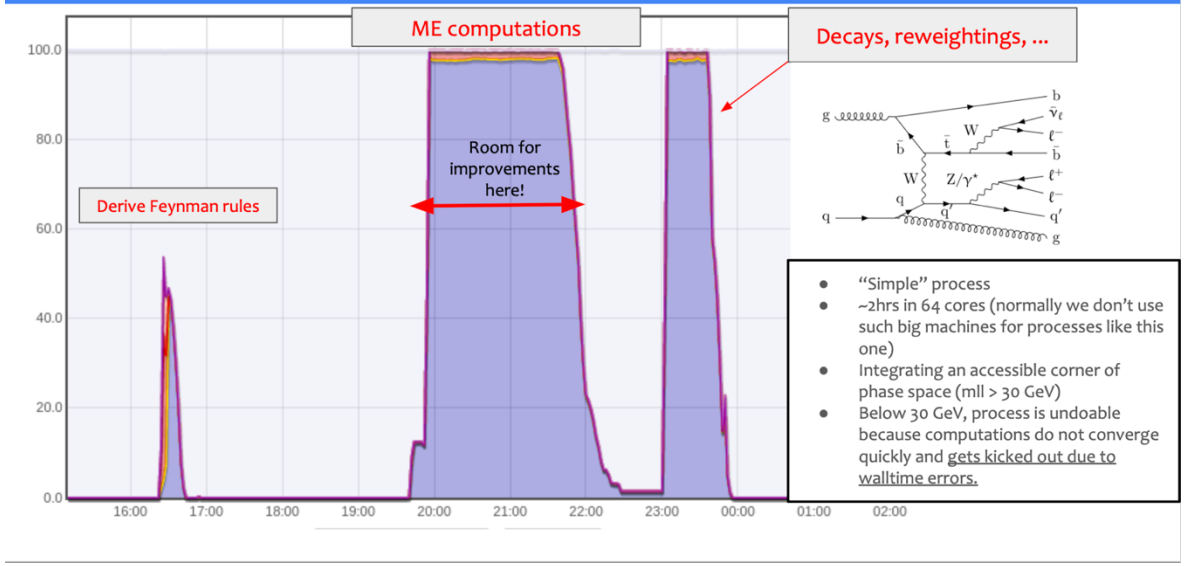
Computing acceleration for MC generation (IFIC + Oviedo)

- Collaborative project with IFIC
- Current MC generation tools take significant amount of time (and CPU resources, and power) to obtain the matrix element calculation.
- Exploring the acceleration of such computation in different COTS (GPUs, FPGAs, ACAPs…) and compare speed, accuracy, and **power consumption**.



- BONUS: Project will be extended to benchmark clustering or tracking algorithms in available COTS.

Showcase of the bottleneck (one common example – tZq@NLO)



Summary

- Many exciting projects ongoing with solid collaboration with other members of the network.
- Main focus is real time data analysis. Activities R&D for the CMS trigger upgrade and beyond.
  - Software and firmware development
  - First exploration of the use of advanced ML algorithms for muon (track) reconstruction
  - **Extend reach of new physics searches**: unexplored signatures (LLPs, HSCPs…) or regions of the phase-space will be within reach
  - Investigate the use of top-of-the-line COTS for future trigger applications
- Differential programming for future experiments design.
- Accelerating CPU-intensive processes with FPGAs.

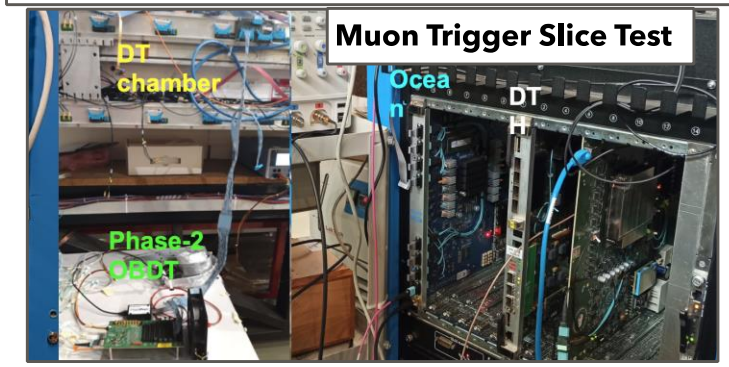
backup

Our demonstrator

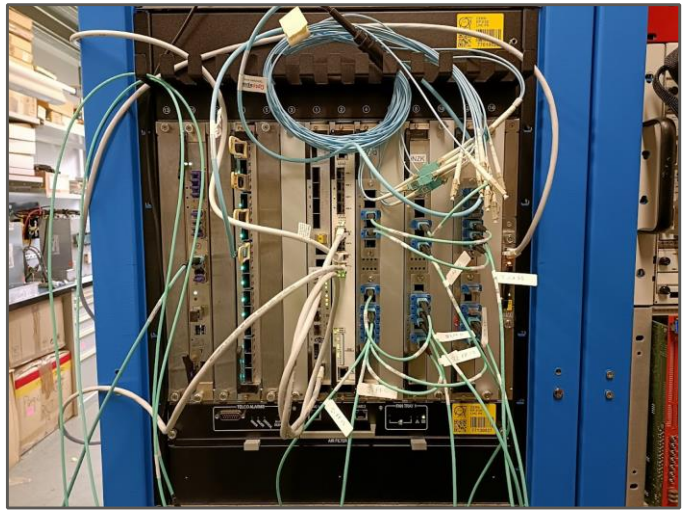
TESTING AND SYSTEM DEMONSTRATION

Phase-2 Level-1 Trigger system demonstration

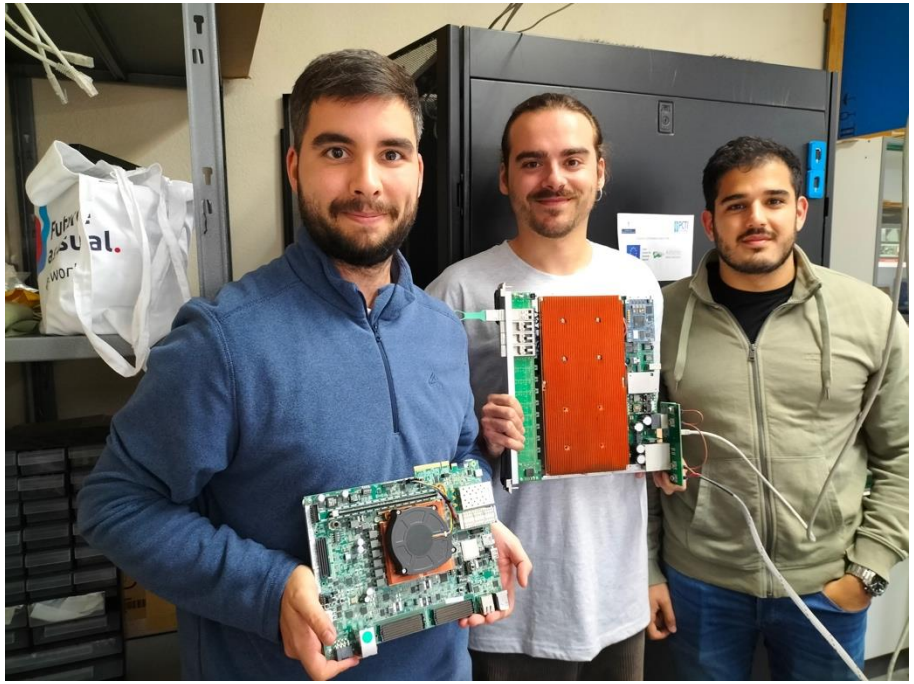
- Single-board and multiple board tests performed
- Integration centers across the globe: larger scale integration @ CERN (904). Multiple flavour board tests.
- Slice test in Muon Barrel Trigger during Run-3. Installation @P5: DT–>BMT–>GMT–>GT
- **Board interconnection: protocol**
  - Links (asynchronous) operation @ 25.78 Gb/s
  - L1 Trigger boards sending packets only once (no retransmission) → error proof
  - Protocols (64/66b or 64/67b) encoding achieved low error rate, validated recovery mechanism etc.



Muon Trigger Slice Test



Building 904 @ CERN



INTREPID

M1

SW R&D

Displaced jets in the barrel

Kalman filter

M2

FW R&D

Timing, latency and occupancy

Demonstrator

M3

GNNs

Hit-based pattern recognition

AI engines with Versal ACAPs

M4

Slice Test

System-level demonstrator

Proof of concept

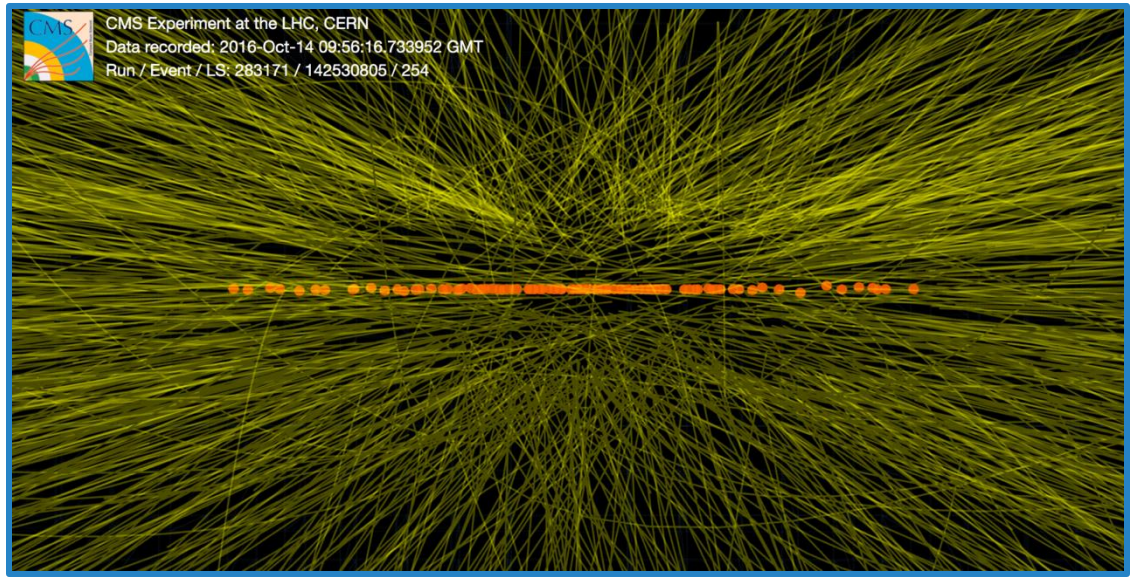
M5

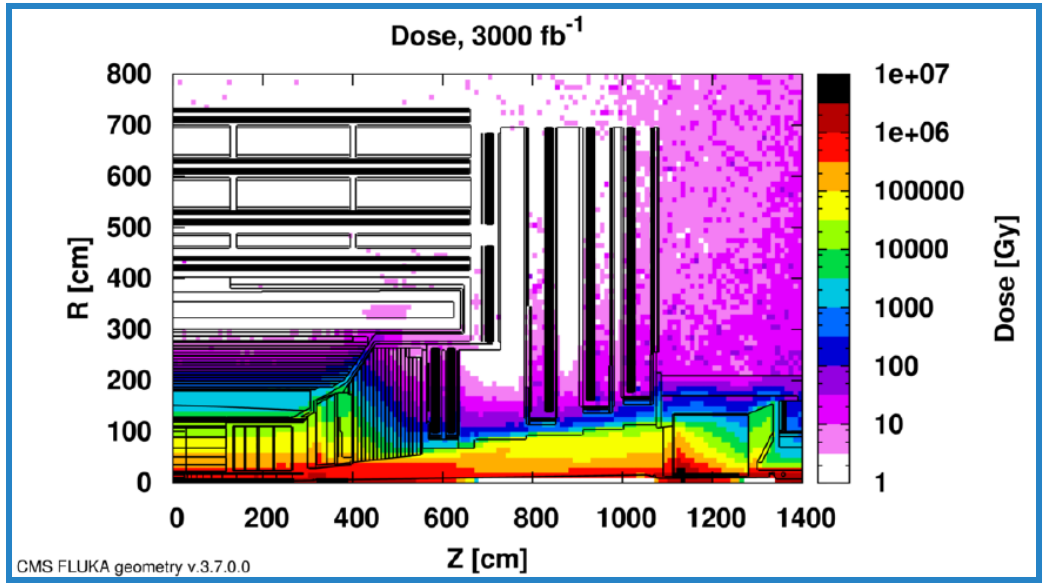
Discovery

Long-lived particles at the HL-LHC or beyond

Dark matter?

HL-LHC: challenges

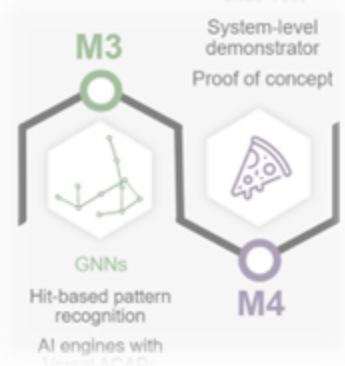




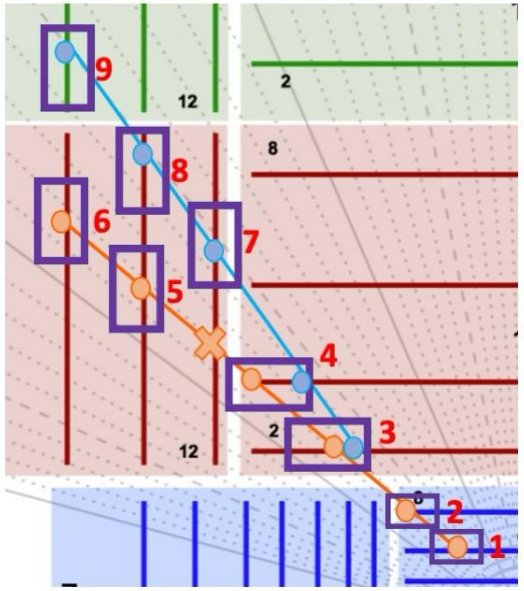
- **Expected pileup** (PU): ~140 (nominal HL-LHC lumi)
- Motivates/requires:
  - Improved granularity wherever possible
  - Novel approaches to in-time Pile Up mitigation: Precision Timing detectors (30ps)
  - A complete renovation of the Trigger and DAQ systems for better selectiveness, despite the high PU.

- **Radiation damage / accumulated dose** in detectors and on-board electronics may result in a progressive degradation of the performance.
- Maintain detector performance in harsh conditions:
  - The complete replacement of the Tracker and Endcap Calorimeter systems.
  - Major electronics overhaul and consolidation of the Barrel Calorimeters and Muon systems

Graph building techniques

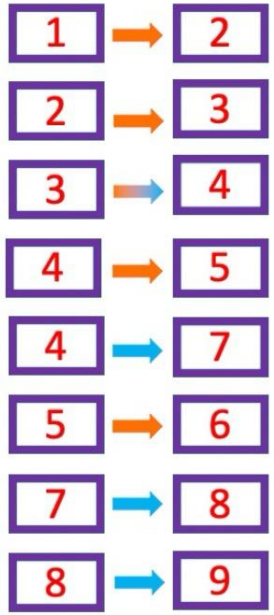


Particles leaving hits



Done once

Module map creation



For event reconstruction

Graph creation