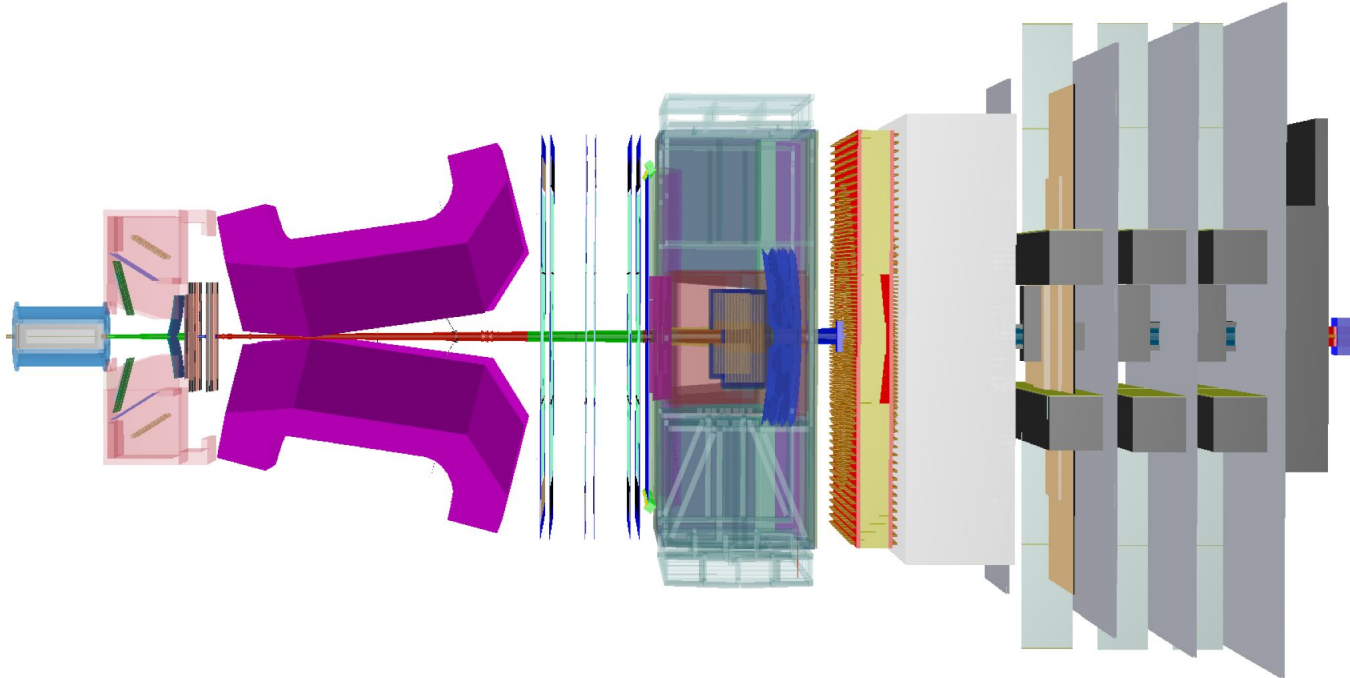


A 3D schematic diagram of the LHCb ECAL detector upgrade. The diagram shows a central beam pipe (green) passing through various detector components. From left to right, there is a blue cylindrical component, a pinkish-red rectangular component, a large purple trapezoidal structure, a series of vertical blue and green bars, a central blue cylindrical structure with internal components, a yellow and red layered structure, a white rectangular block, and a series of grey and light blue rectangular blocks. The text "Radiation-Hard, ML-Based, Low-Latency Compression for the LHCb ECAL Upgrade" is overlaid in the center.

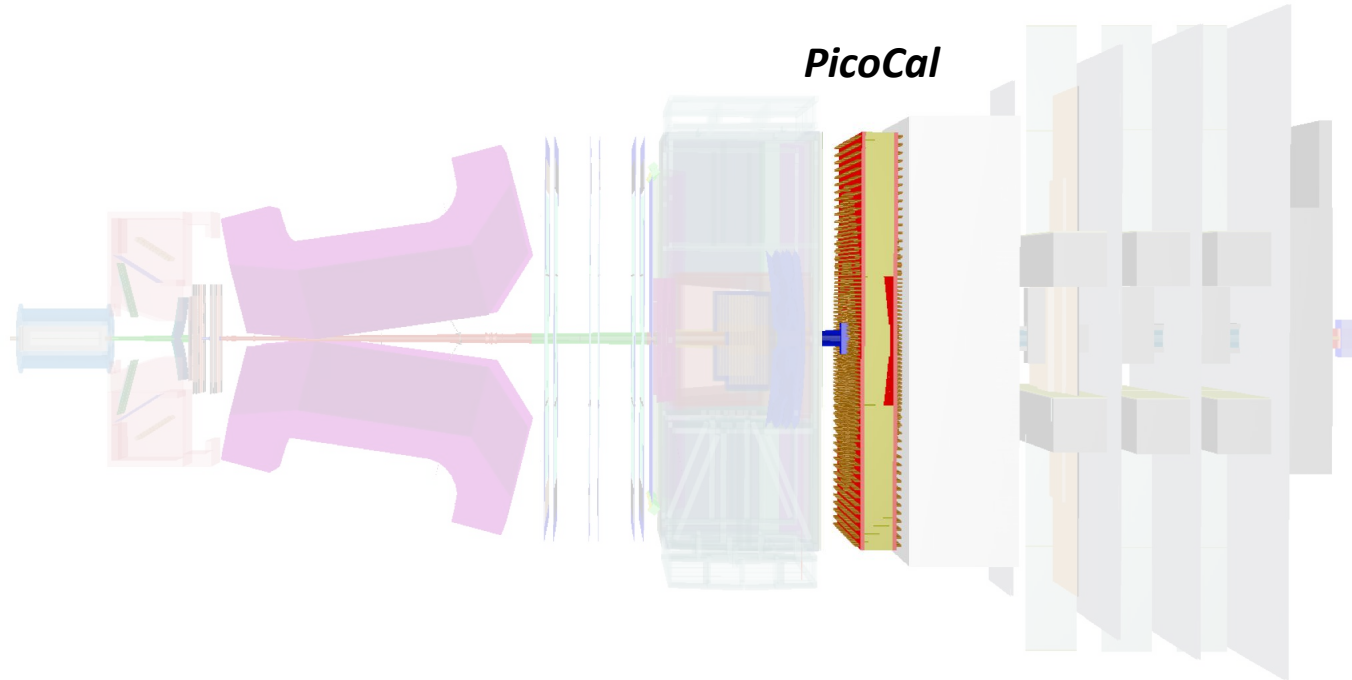
# Radiation-Hard, ML-Based, Low-Latency Compression for the LHCb ECAL Upgrade

Katya Govorkova, Julián García Pardiñas, Victoria Nguyen, Eluned Anne Smith (MIT)  
Vladimir Loncar (CERN)  
4th September 2025

The *aim of the LHCb Upgrade II*, scheduled for installation during LS4, is to operate at more than *five times the current instantaneous luminosity* with *data rate of 200 Tb/s*

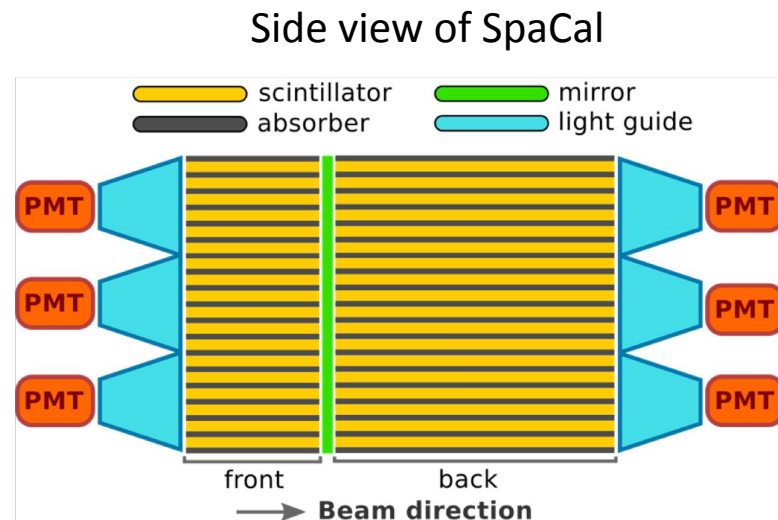
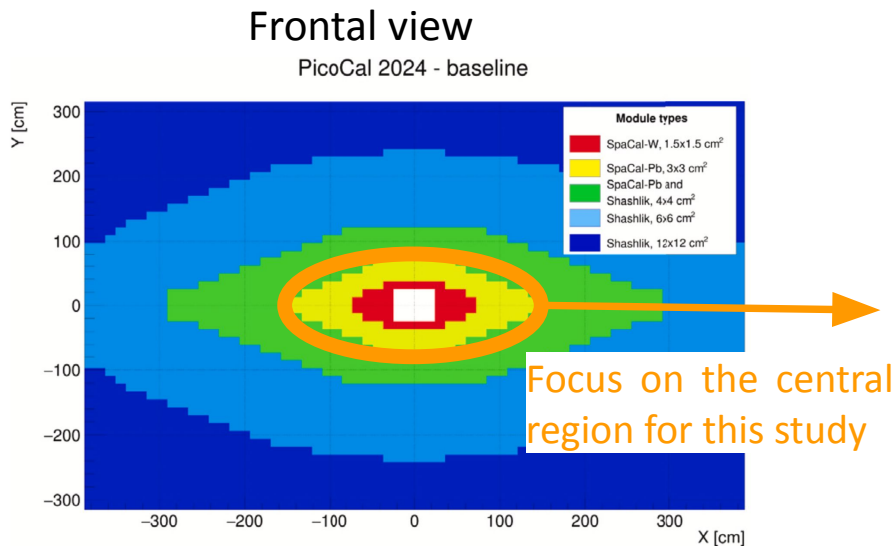


The *aim of the LHCb Upgrade II*, scheduled for installation during LS4, is to operate at more than *five times the current instantaneous luminosity* with *data rate of 200 Tb/s*



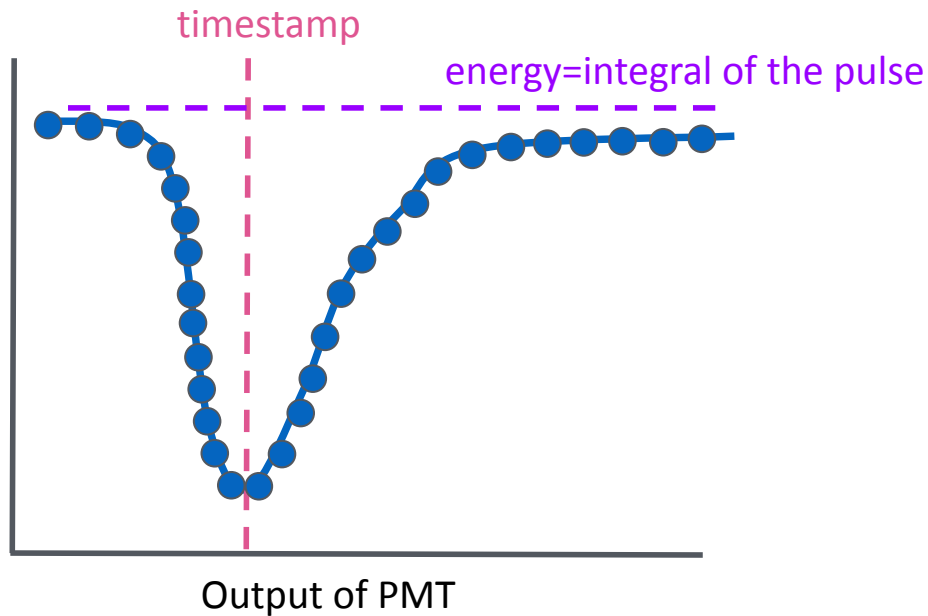
The main tasks of *the LHCb electromagnetic calorimeter* are the *reconstruction* of neutral pions and photons, as well as charged *particle identification* with an emphasis on electrons

Timing capabilities with  $O(10)$  ps precision at high energy are needed in each readout unit over the entire surface. *The Upgrade II calorimeter* is therefore referred to as *PicoCal*:

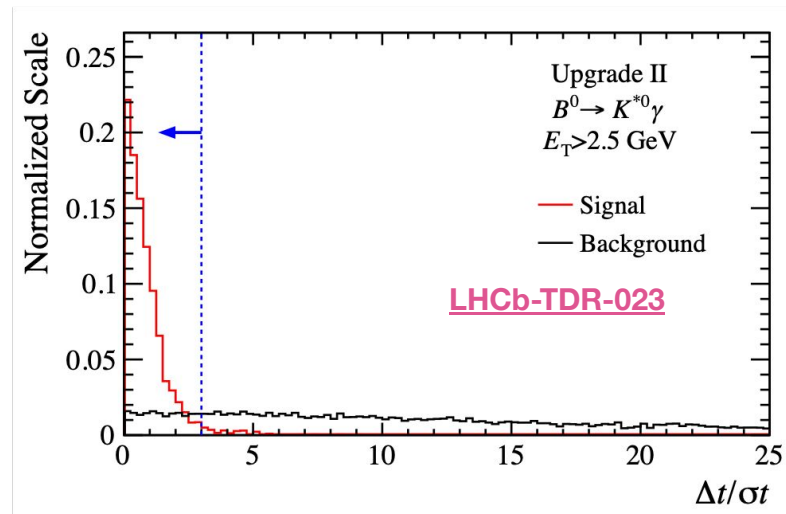


SpaCal consists of alternating **absorber** and **scintillator** layers:

- Light is collected by **wavelength-shifting fibers / light guides** and sent to **PMTs** (photomultiplier tubes)
- Provides precise **energy and timing information** for each particle

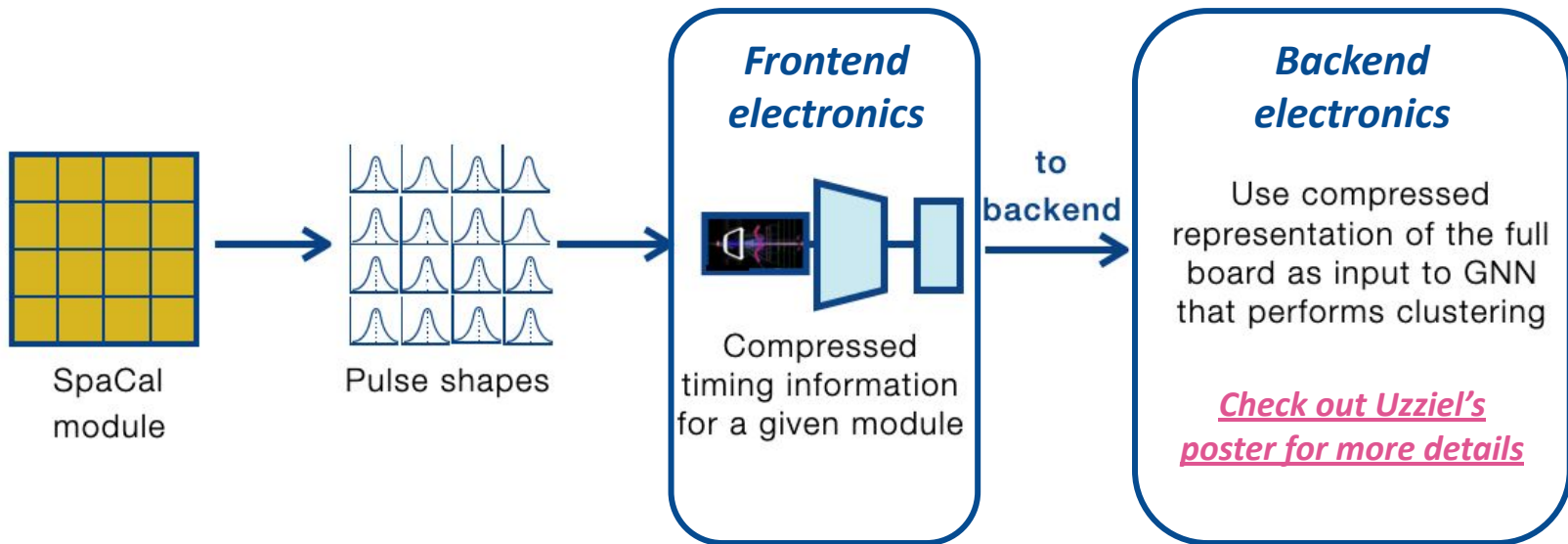


- The **output of PMTs** go through digitizer
- Digitized output of the PMT consists of **32 data points**
- The **timestamp** measures the time of arrival of particles to the calorimeter



- At **30% occupancy we can send a handful of 10-bit numbers per cell** from frontend to backend → **we can not send the whole pulse** (32 data points)

We **could extract only the timestamp (baseline)** and send only it, but then **we lose** all the rest of the information from the pulse shape

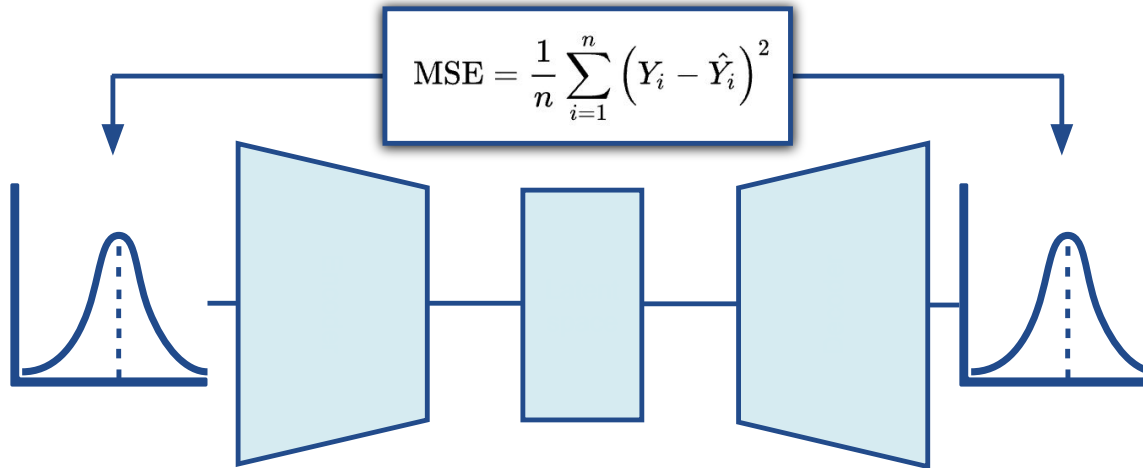


We develop compression algorithm that will have to operate in **radiation high conditions**

- ***Designing ML algorithm to be able to operate in high radiation environment***, for efficient compression, allowing us to pass more timing information (e.g. pulse shape/timing from more cells) and/or save bandwidth/cost
- ***Studying the expected improvements in energy reconstruction*** derived from using ML-enhanced timing information (improved time stamp/pulse shape) as part of the input
- ***Expanding the hls4ml software library with a new SmartHLS/Libero backend***, to enable ML model inference in Polarfire FPGAs

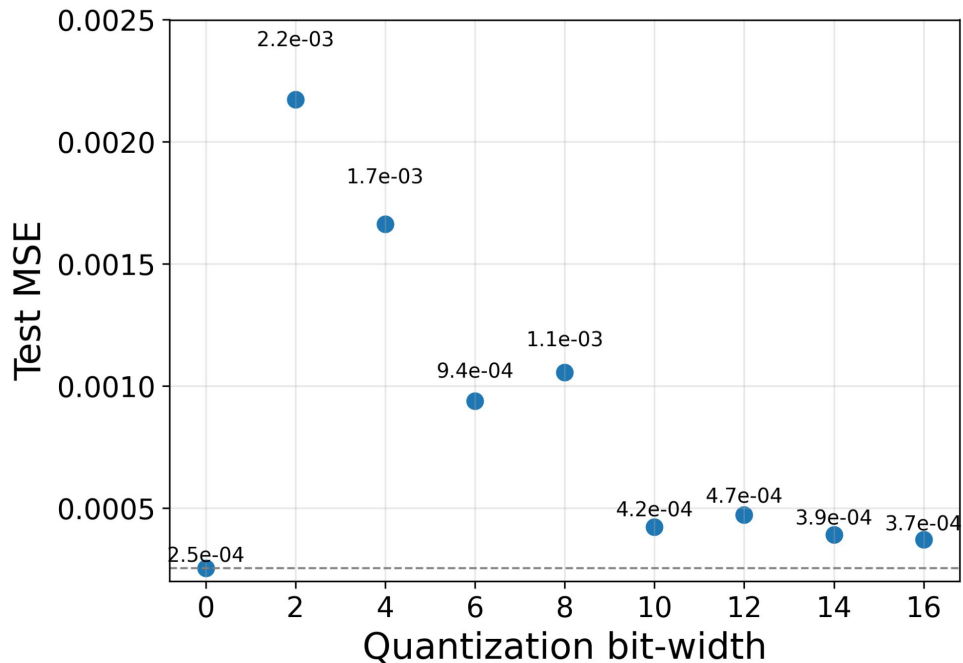
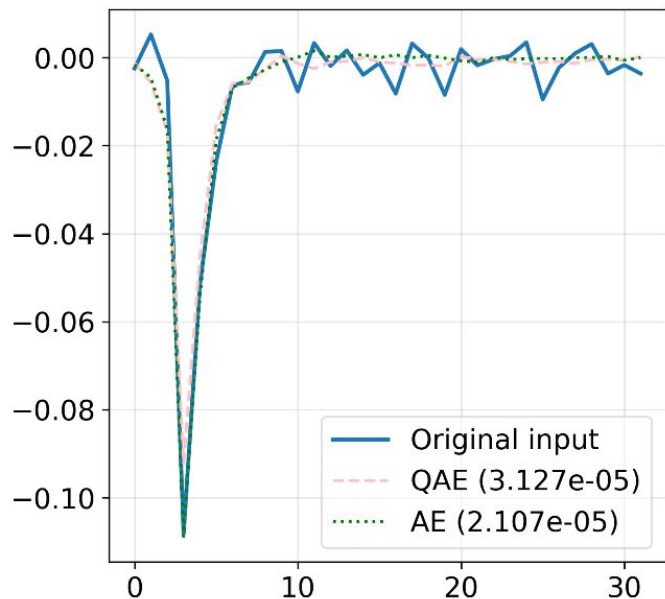
- ***Designing ML algorithm to be able to operate in high radiation environment***, for efficient compression, allowing us to pass more timing information (e.g. pulse shape/timing from more cells) and/or save bandwidth/cost
- *Studying the expected improvements in energy reconstruction* derived from using ML-enhanced timing information (improved time stamp/pulse shape) as part of the input
- *Expanding the hls4ml software library with a new SmartHLS/Libero backend*, to enable ML model inference in Polarfire FPGAs

- We found that the smallest **32–2–32 model** works well enough
- We use **Dense layers** as the most suitable for our type of data
- **Encode** input in a smaller dimensional space
- Minimize **mean squared error** during the autoencoder training to find an optimal configuration of the network weights



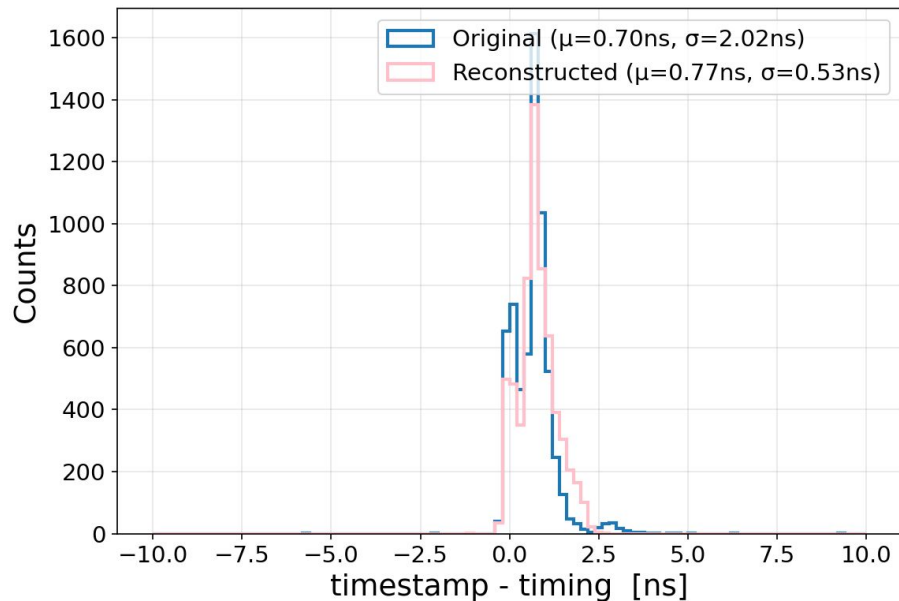
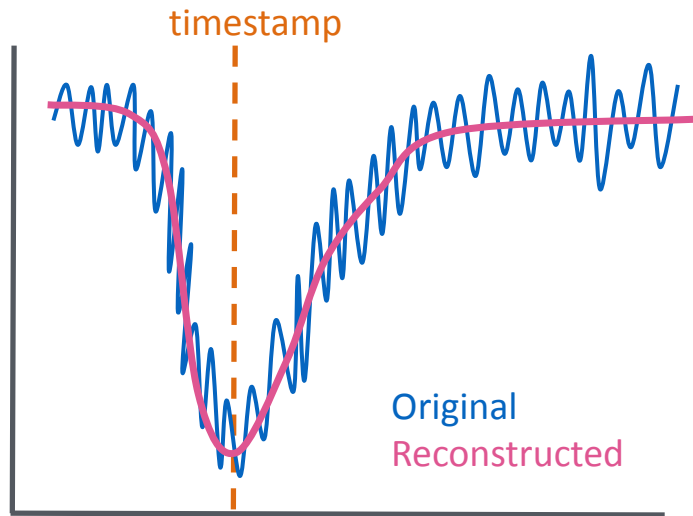
## Example of the reconstructed pulse shape

We use Quantization-aware training to quantize the model before deployment



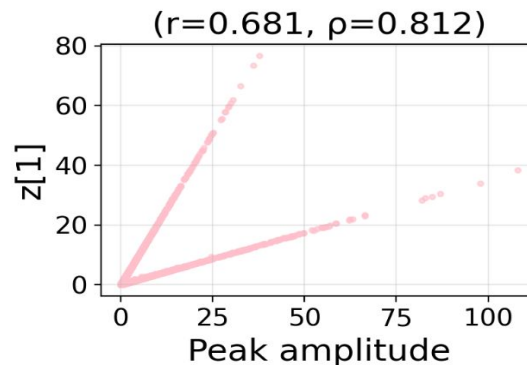
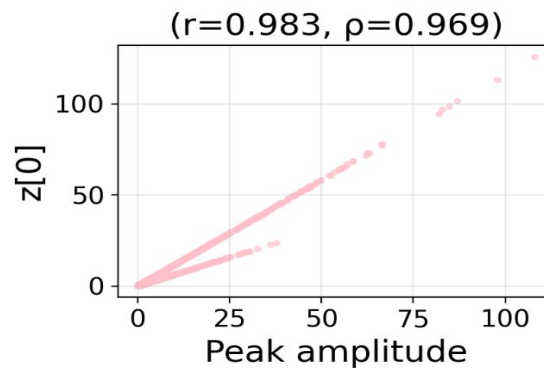
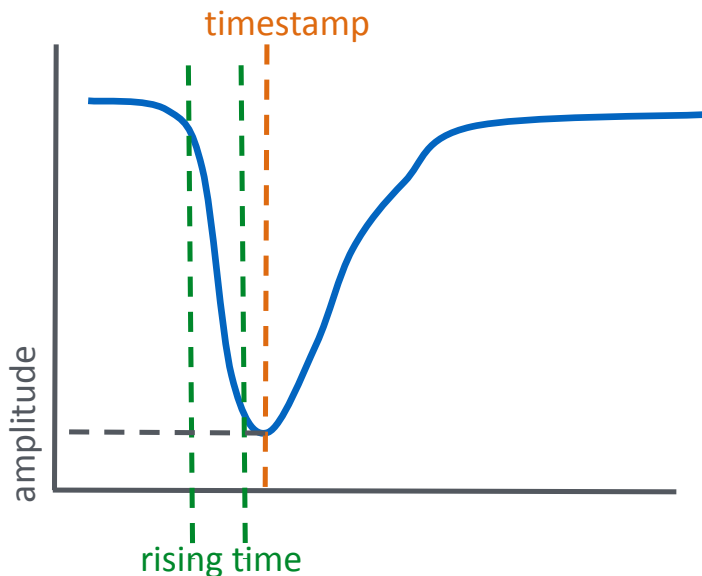
- *Designing ML algorithms* to be able to operate in high radiation environment, for efficient compression, allowing us to pass more timing information (e.g. pulse shape/timing from more cells) and/or save bandwidth/cost
- ***Studying the expected improvements in energy reconstruction*** derived from using ML-enhanced timing information (improved time stamp/pulse shape) as part of the input
- *Expanding the hls4ml software library with a new SmartHLS/Libero backend*, to enable ML model inference in Polarfire FPGAs

- The **autoencoder de-noises pulses**, enabling reliable CFD timestamp extraction
- In raw data, around 50% of timestamps are not defined due to noise; in reconstructed pulses, **all timestamps are correctly extracted**



The *autoencoder* latent space shows correlation with the the pulse:

- *rising time* (time between 10% and 90% of the max amplitude)
- *pulse amplitude*
- *generated time* of arrival of the particle



- *Designing ML algorithms* to be able to operate in high radiation environment, for efficient compression, allowing us to pass more timing information (e.g. pulse shape/timing from more cells) and/or save bandwidth/cost
- *Studying the expected improvements in energy reconstruction* derived from using ML-enhanced timing information (improved time stamp/pulse shape) as part of the input
- ***Expanding the hls4ml software library with a new SmartHLS/Libero backend***, to enable ML model inference in Polarfire FPGAs

1. [✓] Use hls4ml to parse an example keras model targeting a Vivado backend. Understand the structure of the produced C++ code.
2. [✓] Understand how SmartHLS works. Study the project and file structure for a simple example provided in their tutorials.
3. [✓] Transform the relevant hls4ml output files to allow SmartHLS to parse them.
  - A. [✓] Get a basic version of the C++ code to compile within SmartHLS.
  - B. [✓] Check the numerical results after variable quantisation with CSIM.
  - C. [✓] Synthesise and study latency.
  - D. [✓] Play with pragmas to optimise performance as needed.
4. [✓] Abstract from the learnings and implement an expansion of hls4ml to cover this new backend.
5. [✓] Use the new expansion of hls4ml to convert the LHCb ML models.



[https://github.com/vloncar/hls4ml/tree/libero\\_backend](https://github.com/vloncar/hls4ml/tree/libero_backend)

Orientative preliminary target specs:

- **FPGA model: MPF100T** (✓ we are focusing on MPF100T-FCVG484I)
- **Initialisation interval: 4** (✓ requirement met)
- **Clock period: 6.25 ns** (✓ requirement met)
- **Max total latency: 1  $\mu$ s** (✓ ~30 ns)

```
===== 1. Simulation Result =====
```

| Top-Level Name | Number of calls | Simulation time (cycles) | Call Latency (min/max/avg) | Call II (min/max/avg) |
|----------------|-----------------|--------------------------|----------------------------|-----------------------|
| myproject_top  | 10              | 45                       | 5 / 5 / 5.00               | 4 / 4 / 4.00          |

```
SW/HW co-simulation: PASS
```

```
===== 2. Timing Result of HLS-generated IP Core (top-level module: myproject_top) =====
```

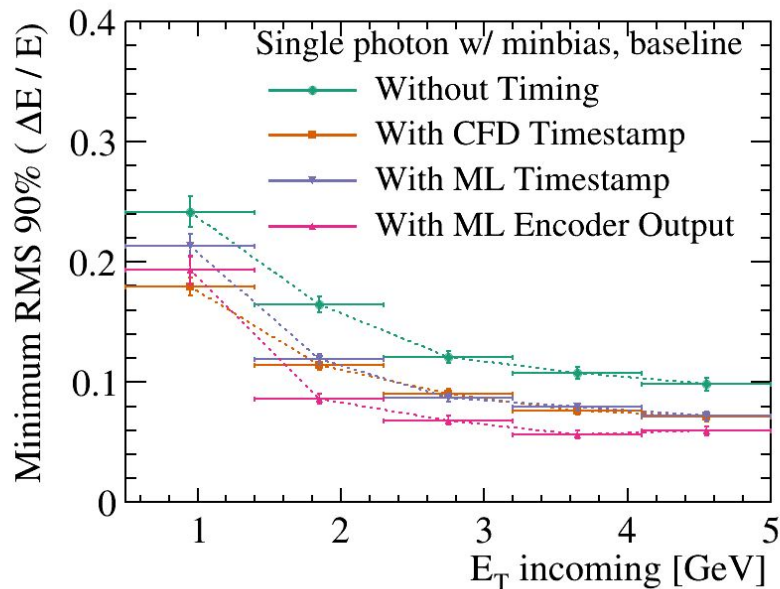
| Clock Domain | Target Period | Target Fmax | Worst Slack | Period   | Fmax        |
|--------------|---------------|-------------|-------------|----------|-------------|
| clk          | 6.250 ns      | 160.000 MHz | 0.600 ns    | 5.650 ns | 176.991 MHz |

- ***Efficient compression with ML*** – We developed algorithms that preserve detailed timing information (pulse shape and timestamp) while reducing bandwidth and cost
- ***Improved physics performance*** – Incorporating ML-enhanced timing information leads to measurable gains in energy reconstruction
- ***Hardware deployment*** – We extended **hls4ml** with a SmartHLS/Libero backend, enabling ML inference on Polarfire FPGAs

***BACKUP***

***LHCb***  
***LHCb***

**First (very preliminary) comparison** of energy resolution from GNN reconstruction:



### Caveats:

- **Small training dataset:** ~3000 data points after prefiltering.
- **GNN model not fully tuned** for each setup.
- In contrast with Syracuse, **we are including all module types** in the dataset, not only SpaCal Pb.

- First studies point to an **improvement in resolution when using the encoder's output**
- **More detailed studies and checks to follow**, with more data and better model tuning.