Machine Learning for Real-Time Processing of ATLAS Liquid Argon Calorimeter Signals with FPGAs

-

EPS-HEP Marseille
9th of July 2025

Raphaël Bertrand (Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France)
on behalf of the ATLAS LAr community

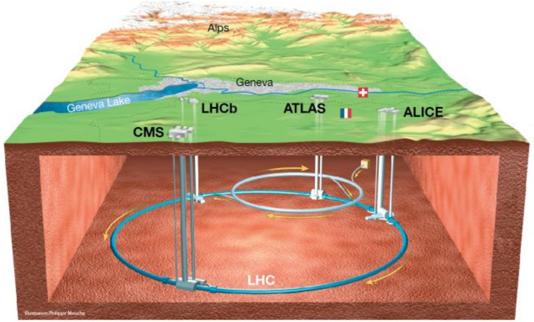
Aix Marseille Université

CENTRE DE PHYSIQUE DES PARTICULES DE MARSEILLE
CPPM

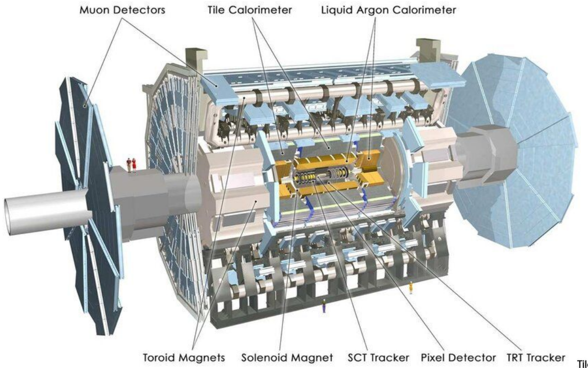
Introduction

Experimental Context

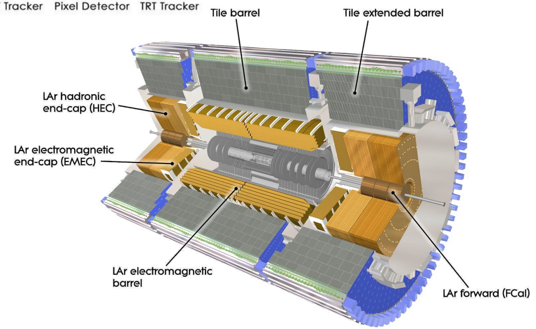
- **Large Hadron Collider (LHC)**
  - Proton-proton collider at 13.6 TeV
  - Protons accelerated via superconducting magnets
  - Collisions at 40 MHz



- **ATLAS detector**
  - General-purpose experiment
  - Very high data rate
    - ➢ On-the-fly event selection required

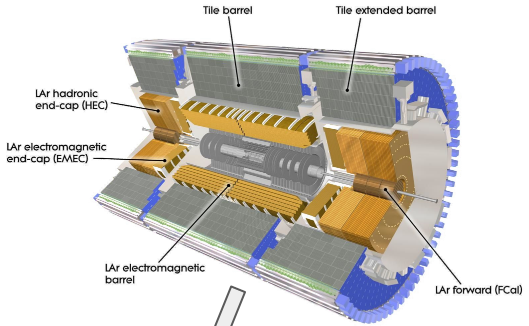


- **Liquid Argon (LAr) Calorimeter**
  - ATLAS sub-detector for energy measurement ($e^{-/+}$, γ, hadrons)
  - Sampling in active LAr alternating with inactive metal (Cu, Pb, W)
    - ➢ Accordion shaper absorbers for EMB and EMEC
    - ➢ Ionization signal from particle interactions

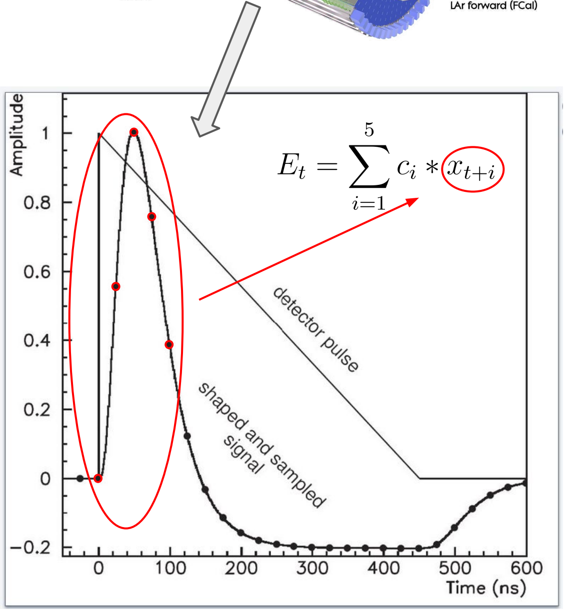


Signal processing and energy reconstruction

- **Electronic signal** produced
  - **Amplitude ∝ true deposited energy (E^{true})**
  - **Spans ~625 ns** (25 proton-proton **B**unch **C**rossings)
  - **Shaped**, **sampled** and **digitized at 40 MHz**

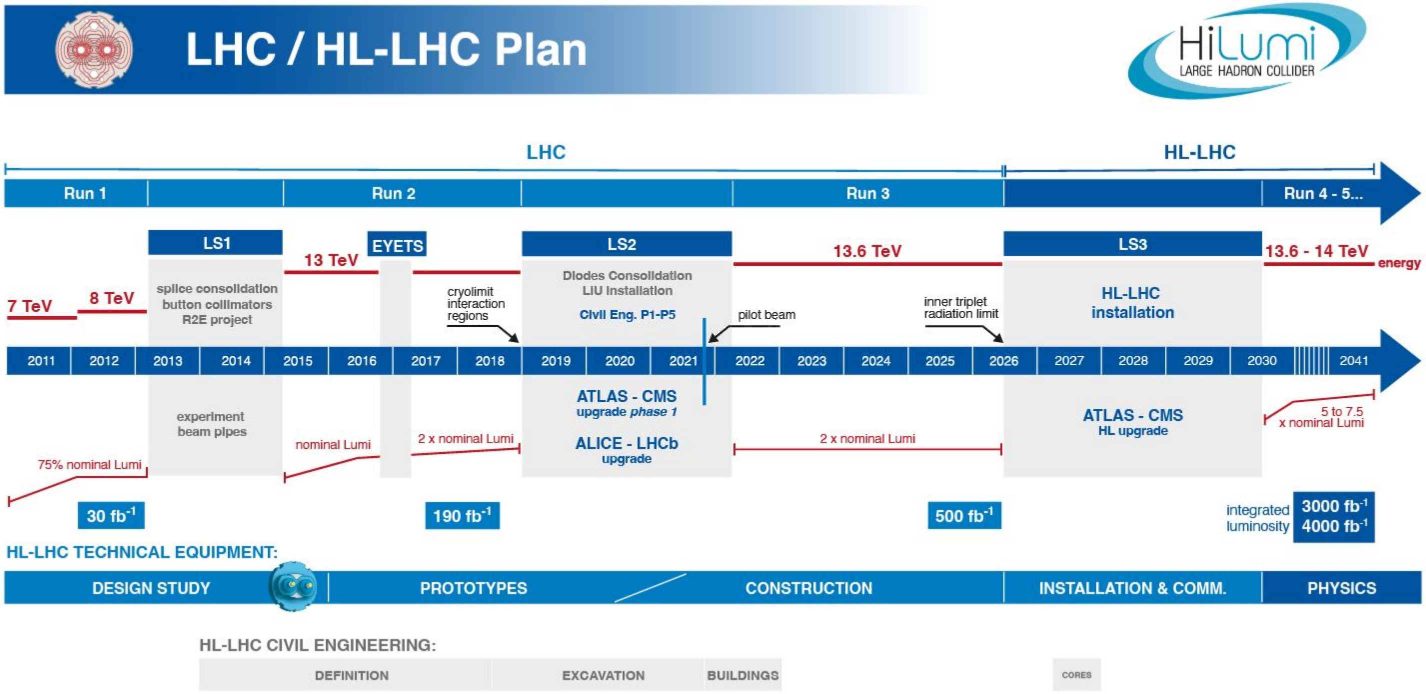


- **Energy reconstruction** with optimal filtering (OF) algorithm
  - **Weighted sum** of samples around the pulse peak
  - **Max finder/Timing cut** to select the correct BC



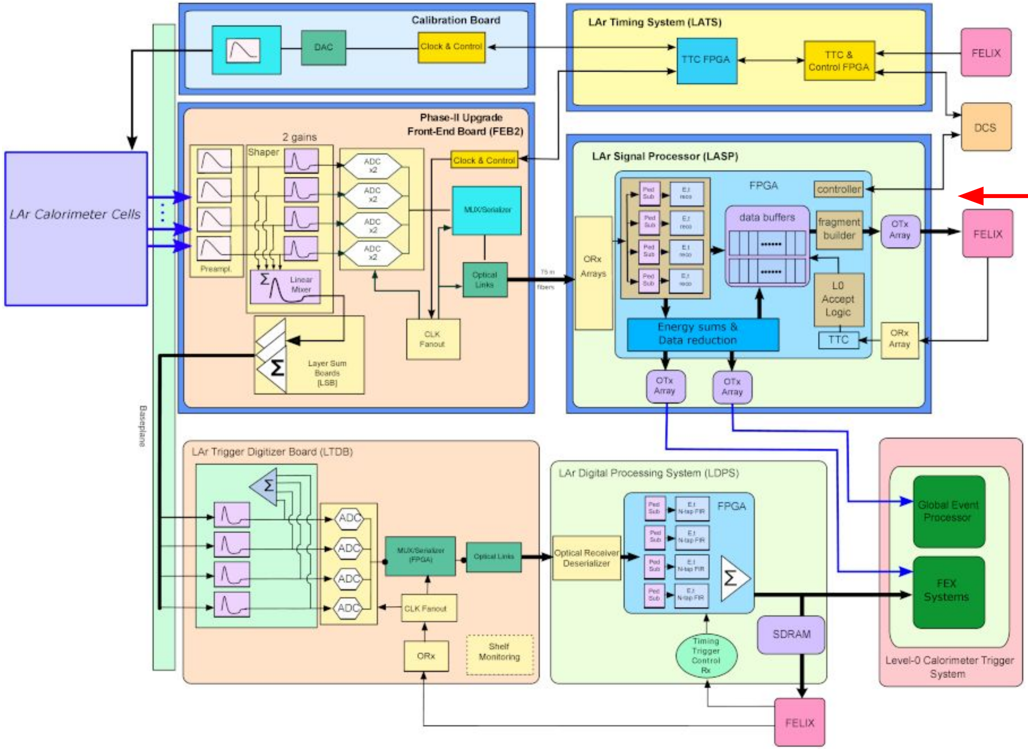
- Reconstruction algorithm requirements :
  - **Online** computation (per BC)
  - **Max latency : ~125 ns** (used in trigger system)
  - **Fit in FPGAs** : **O(500)** Multiply-Accumulate operations (**MAC units**)
    - ➢ 5 MAC units required to implement OF
  - **384 channels per FPGA** (many algorithm instances needed)

HL-LHC schedule

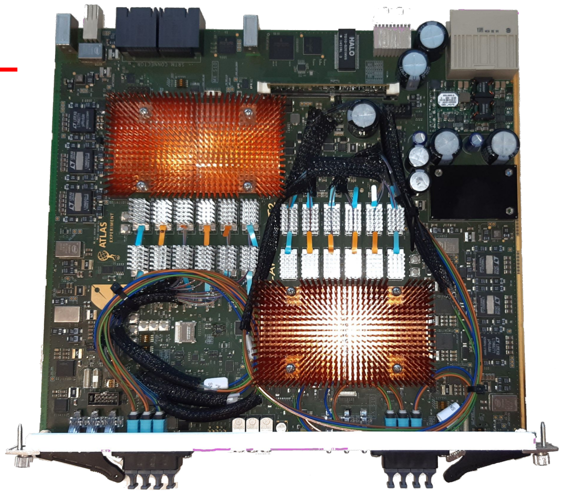


- **Increased luminosity ⇒ Increased pileup**
- HL-LHC is needed to study Higgs properties and detect new rare processes

New LAr readout electronics for energy computation



LASP board Demonstrator

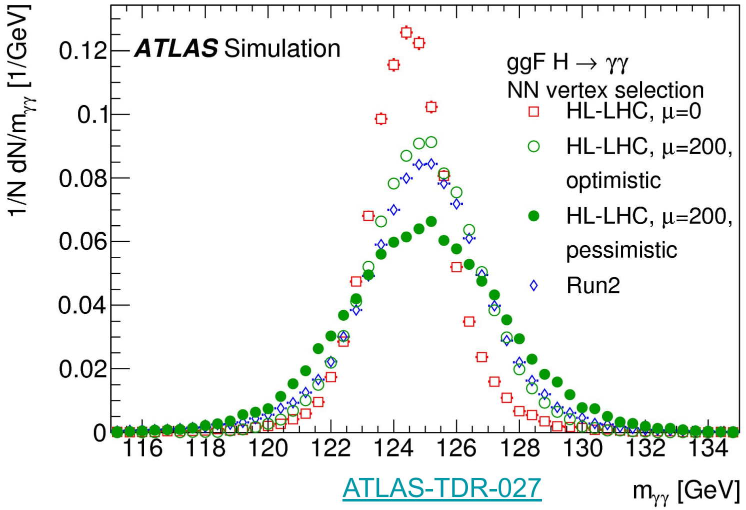


Off-detector readout board (LASP) will carry two state-of-the-art **FPGAs for energy computation**

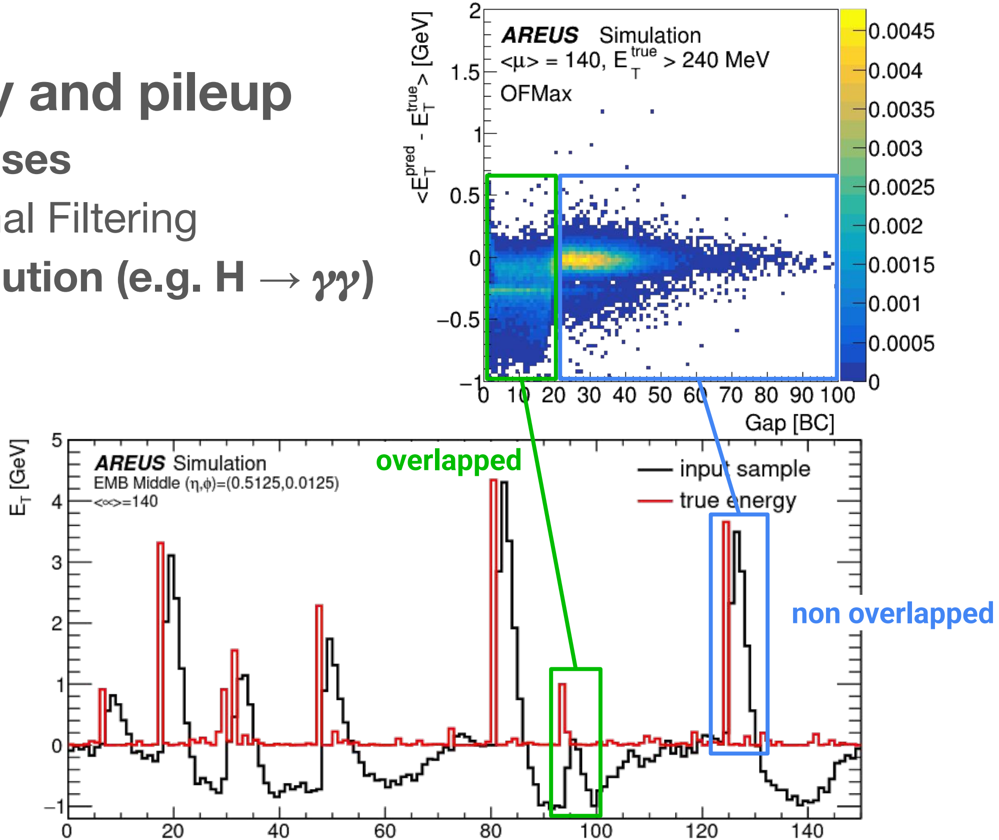
An **opportunity to embark more complex algorithms**

Impact of high luminosity

- HL-LHC ⇒ **Increased luminosity and pileup**
  - **Increased rates of overlapping pulses**
  - ↳ **Degraded performance** of Optimal Filtering
  - **Significant impact on energy resolution (e.g. H → $\gamma\gamma$)**



ATLAS-TDR-027



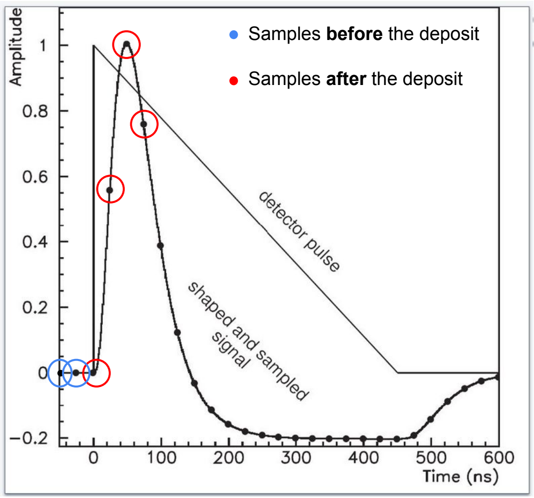
Simulated pileup with additional injected pulses at higher energy

Neural network approaches as energy reconstruction algorithms

Neural network architectures (1/3)

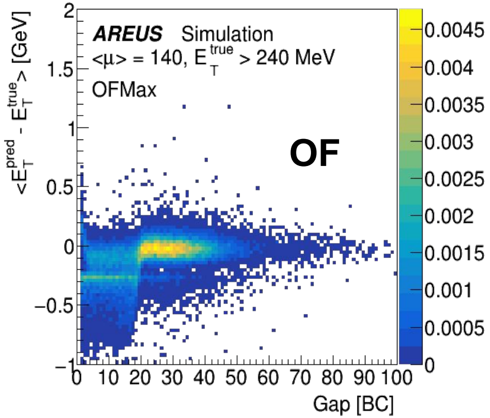
- Exploit samples before the energy deposit to **correct overlapping pulses**
- Several architectures tested : CNN, RNN, Dense layer-based

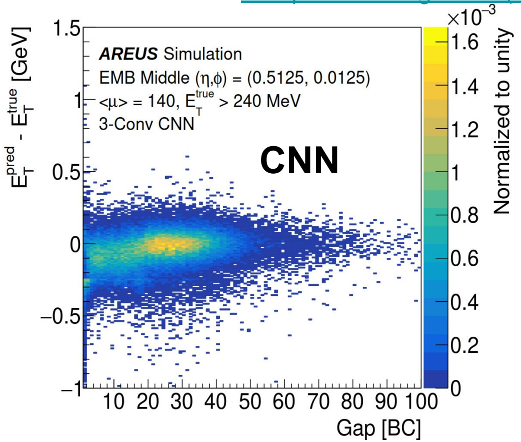
- **Samples** from **before** and **after** the energy deposit are used :
  - **After the energy deposit** (similar to OF inputs)
    - ➢ Capture the pulse amplitude
  - **Before the energy deposit** (additional inputs)
    - ➢ Correct for pulse distortions from previous deposits

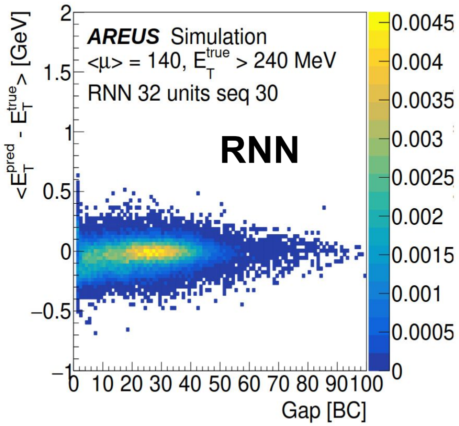


- Preliminary studies done with high rate of pulse overlap
  - **Neural networks can correct for overlapping pulses**
    - The correction is **dependent on the size** of network

Comput.Softw.Big Sci. 5 (2021), s41781-021-00066-y

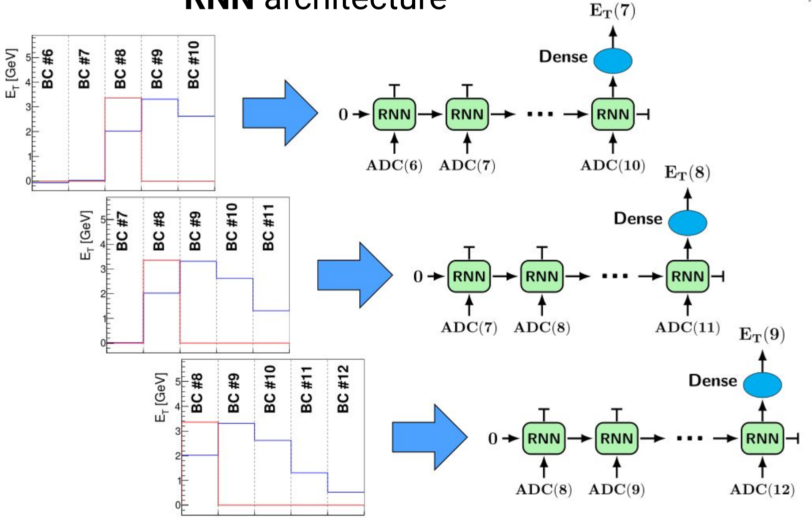




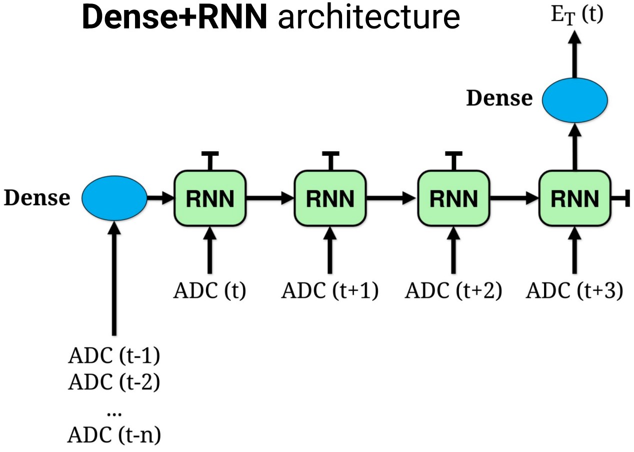


Neural network architectures (2/3)

RNN architecture



Dense+RNN architecture



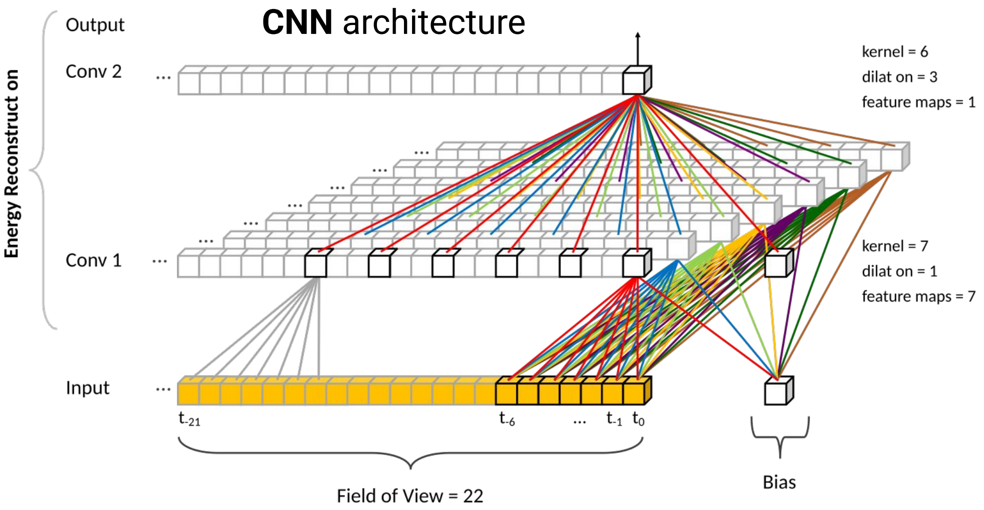
- RNN → One Vanilla RNN cell per sample
  - **Same parameters shared for all the cells**

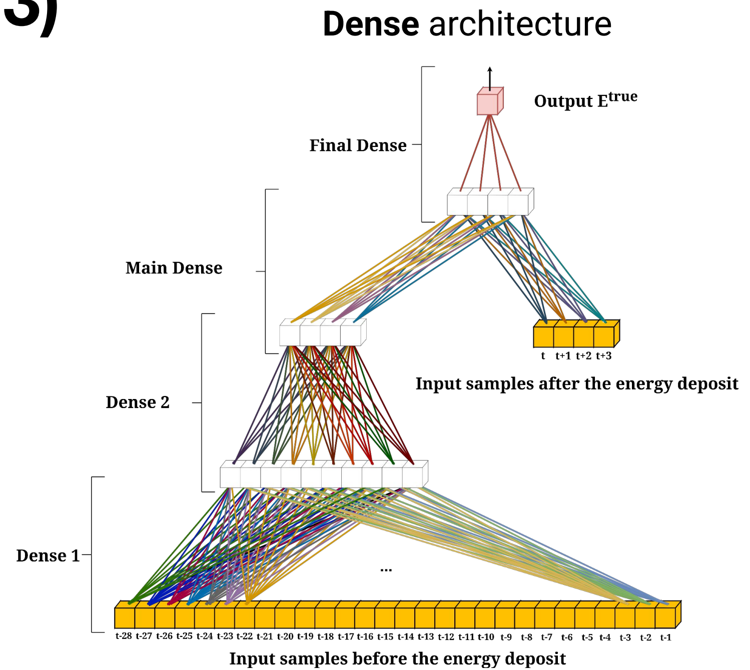
- This architecture has a **low latency** and but requires **high number of MAC units**

- Dense+RNN → RNN architecture optimization
  - **Dense for samples before the deposit**
  - **Vanilla RNN cells for samples after the deposit**

- This architecture has a **low latency** and requires **lower number of MAC units** than RNN.

Neural network architectures (3/3)





- CNN → Convolutional layers
  - **Capture features in the sequence**

- This architecture requires **low number of MAC units** but has a **high latency**.

- Dense → only use Dense layers
  - **Multiple Dense applied on samples before the deposit**
  - **Samples after the deposit are used on the latter layers**

- This architecture requires **low number of MAC units** and has a **moderate** latency

Neural networks hyperparameters tuning

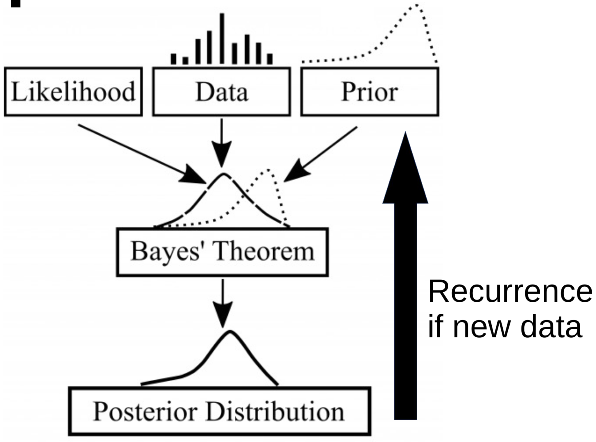
bayesian optimization

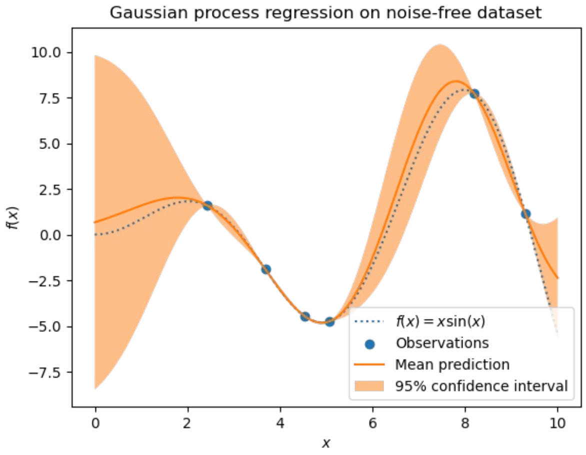
Bayesian optimization

- Goal : **Find the best parameters** to maximize/minimize a **performance function** while **evaluating the function as few times as possible**

- **Initialization** with several random points

- **Iterations** to find the best parameters space
  - **Interpolation** between points
    - ➢ Based on a gaussian kernel with associated uncertainty
  - **Acquisition function** to determine where to evaluate next
    - ➢ Balance between **exploration** and **exploitation**
  - **Evaluation** of the performance function **at the chosen point**





Hyperparameters tuning with bayesian optimization

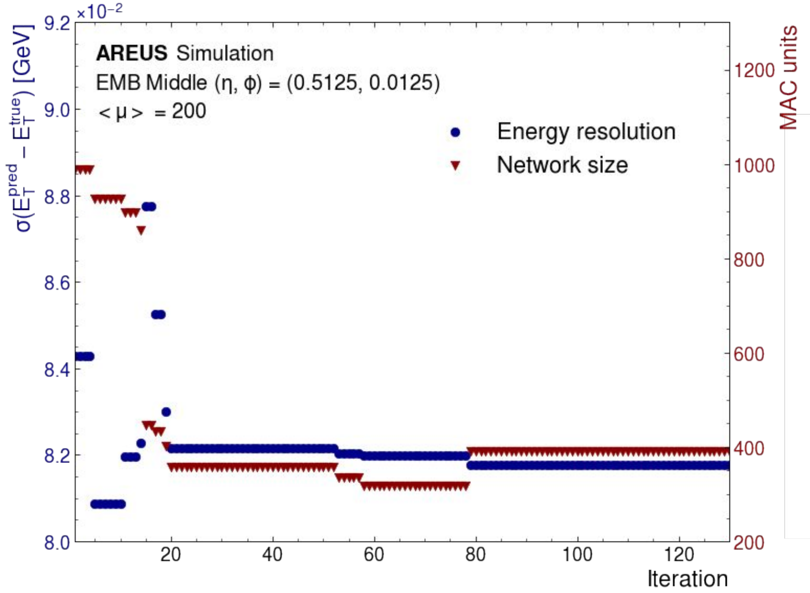
- Optimization on both performance and hardware to fit in FPGAs
  - **Energy resolution** (σ [MeV])
  - **Number of MAC units** (M)
- Hyperparameters to be tuned (e.g. for the Dense architecture) :
  - **Number of samples** (before the energy deposit)
  - **Number of units** for the intermediate layers

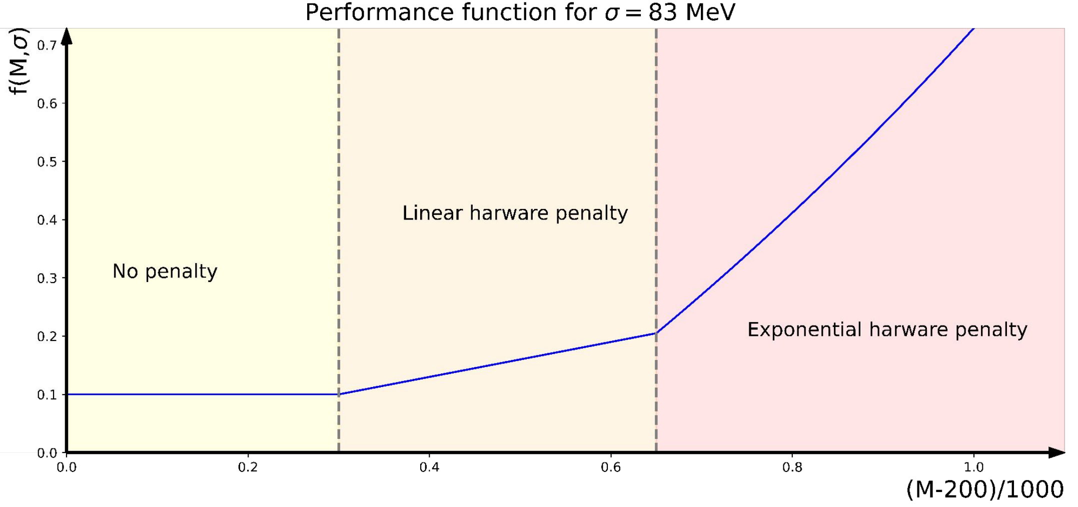
Performance function used for the bayesian optimization :

$$f(M,\sigma) = \frac{\sigma - 70}{130} \; for\ M \leqslant 500$$

$$f(M,\sigma) = f(500,\sigma) + a * \frac{M-500}{1000} \; for\ M \in \,]\, 500\,;850\,]$$

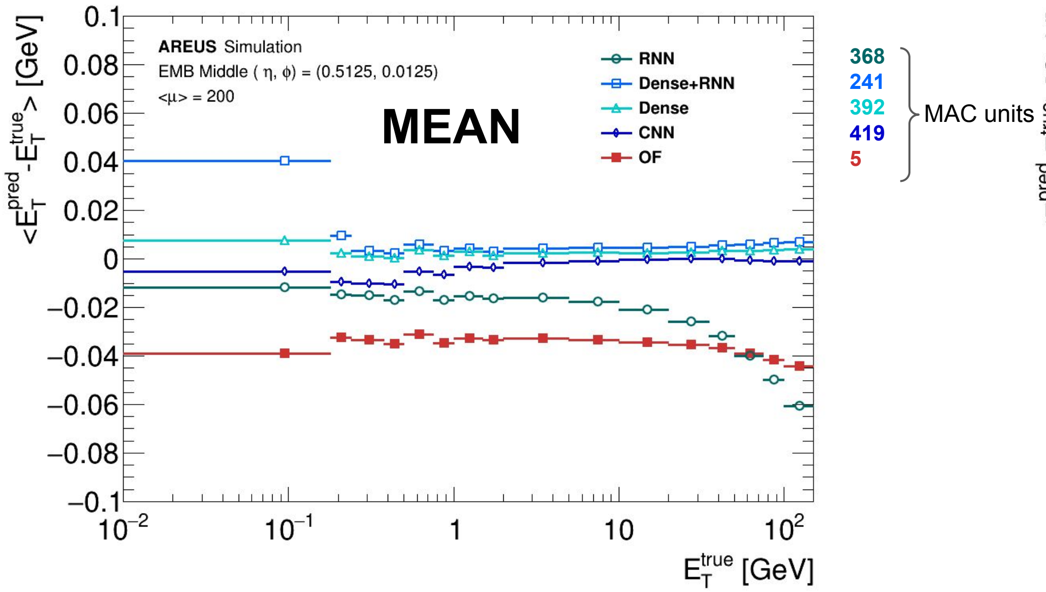
$$f(M,\sigma) = f(850,\sigma) + b * e^{\frac{M-850}{1000}} - 1 \; for\ M > 850$$

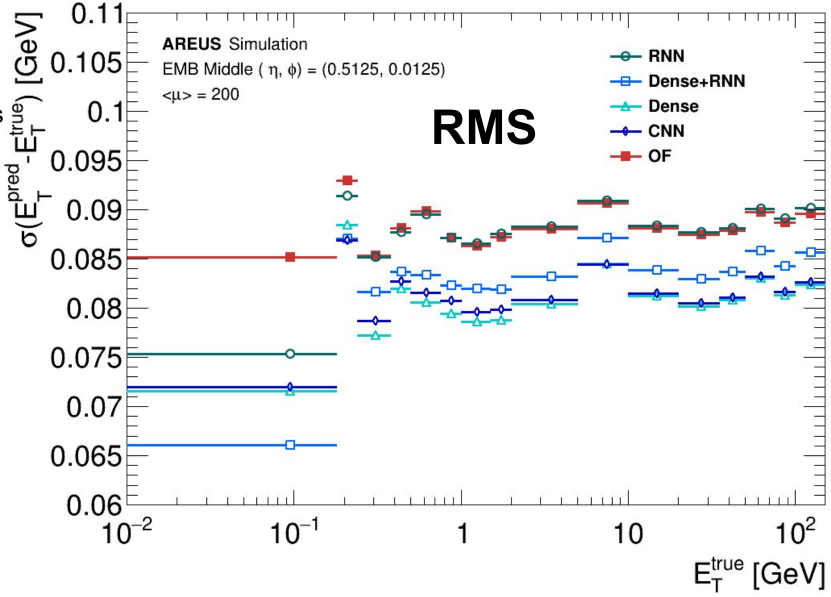




Hyperparameters optimization results

- **Better energy scale** for NNs (E_T^{pred} - E_T^{true} closer to 0)
  - Correction for shift in baseline due to pileup
  - Especially for Dense and CNN
- **Better energy resolution for NNs** compared to OF over the whole energy range
  - Especially for Dense and CNN



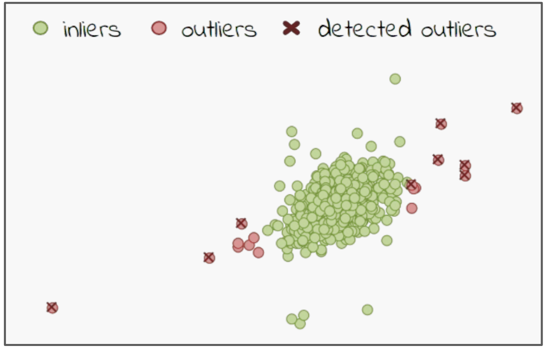


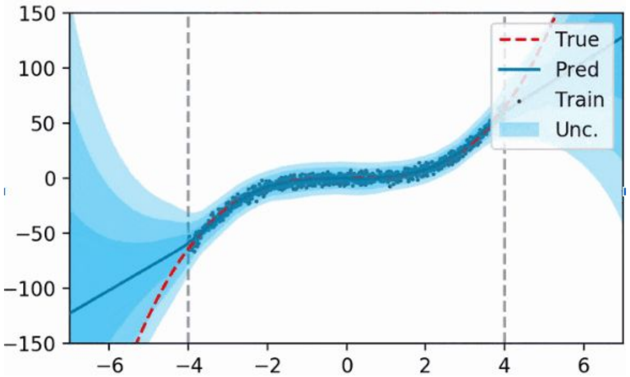
Uncertainty prediction using neural network

with deep evidential regression

Deep evidential regression (DER)

- NNs are trained to minimize their prediction errors
    - Unknown accuracy of the model for individual prediction
    - It would be interesting to **know when the model is more likely to fail (or the opposite)**
- **Model the energy prediction as a distribution**
    - Mean of the distribution → **energy prediction**
    - Standard deviation of the distribution → **uncertainty**
    - Trained to maximize the likelihood
- Differentiate uncertainties :
    - **Epistemic**
        - ➢ Lack of knowledge, model uncertainty
        - ➢ Can be reduced
    - **Aleatoric**
        - ➢ Inherent to data
        - ➢ Cannot be reduced

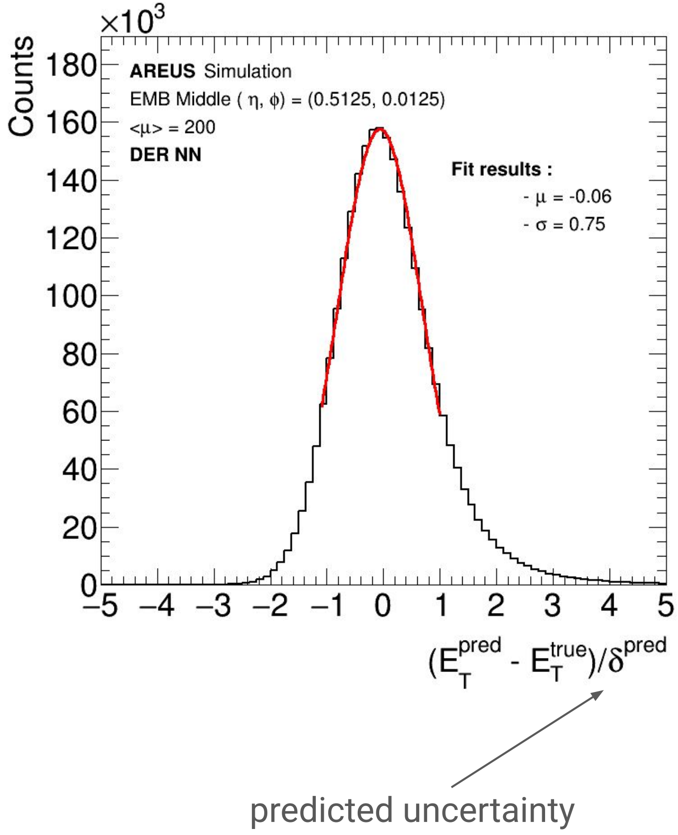




[Deep Evidential Regression, Amini et al.](#)

Deep evidential regression (DER) results

- **DER applied to LAr cells energy reconstruction**
  - Would allow to take into account instantaneous luminosity changes or bunch train structure
- Normale-Inverse Gamma distribution to describe mean and uncertainty
  - **4 parameters** (γ,ν,α,β) rather than one
    - ➢ Uncertainty computation
  - **Still possible to implement in FPGA**
- Overall **good pull distribution**
  - Estimated uncertainty comparable to $E_T^{pred} - E_T^{true}$
  - **Slightly biased**
    - ➢ Right tails
    - ➢ Uncertainty overestimated by 25%

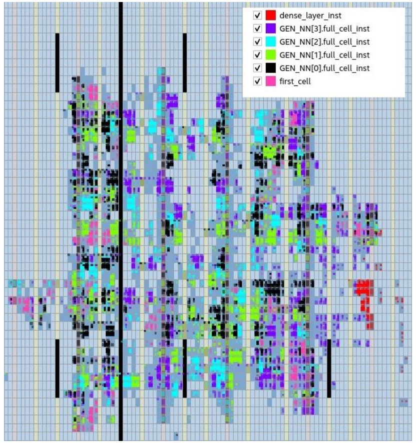


Implementation on an FPGA

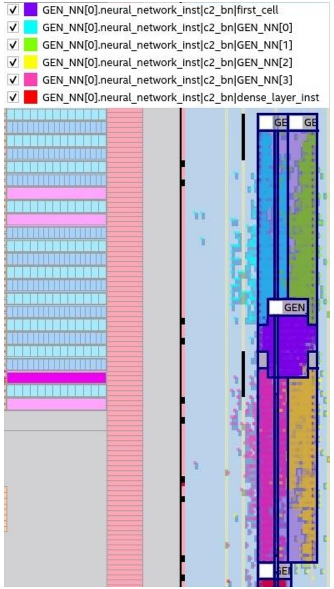
Firmware implementation

- RNN implementation with 304 MAC units
  - Successfully implemented on a Stratix-10 FPGA
    - **HLS** implementation for fast prototyping
      - Supporting HLS4ML
    - **VHDL** implementation to meet all requirements

HLS placement



VHDL forced placement



	N networks x multiplexing	ALM	DSP	FMax	latency
Target	**384 channels**	**30%***	**70%***	**Multiplexing x 40 MHz**	**125 ns**
“Naive” HLS	384x1	226%	529%	-	322 ns
HLS optimized	37x10	90%	100%	393 MHz	277 ns
VHDL optimized	28x14	18%	66%	561 MHz	116 ns

Conclusion

- Online **energy reconstruction for LAr cells** performed **using neural networks**
- Four neural network architectures were tested and optimized
    - **CNN**, **RNN**, **RNN+Dense** and **Dense**
- **Hyperparameters tuning** performed using bayesian optimization
    - **Balance between performance and size of the network** to fit in FPGAs
    - **NNs outperform OF**
- **Uncertainty** on energy prediction using deep evidential regression
    - **Accurate** uncertainty prediction
    - **Possible to implement in FPGAs**
- **Prototype implementation in firmware** performed