Hyperparameters optimization for LAr energy reconstruction neural networks

-

THINKII Workshop 2025

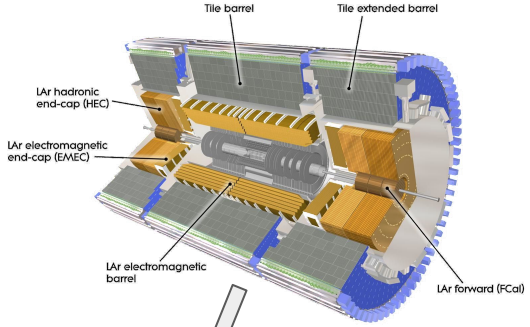
Raphaël Bertrand (CPPM)

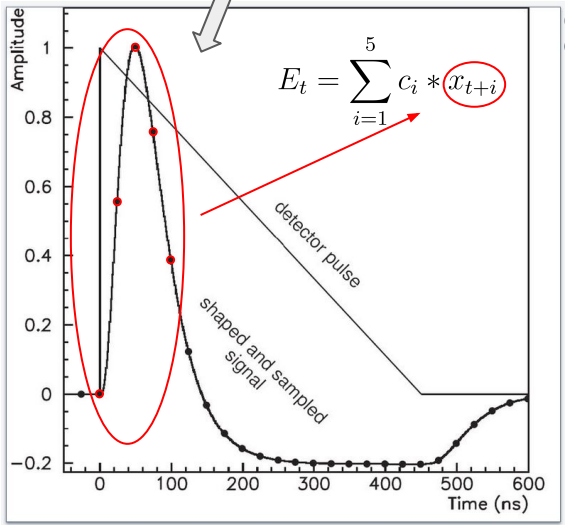
Signal processing and energy reconstruction

- **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**

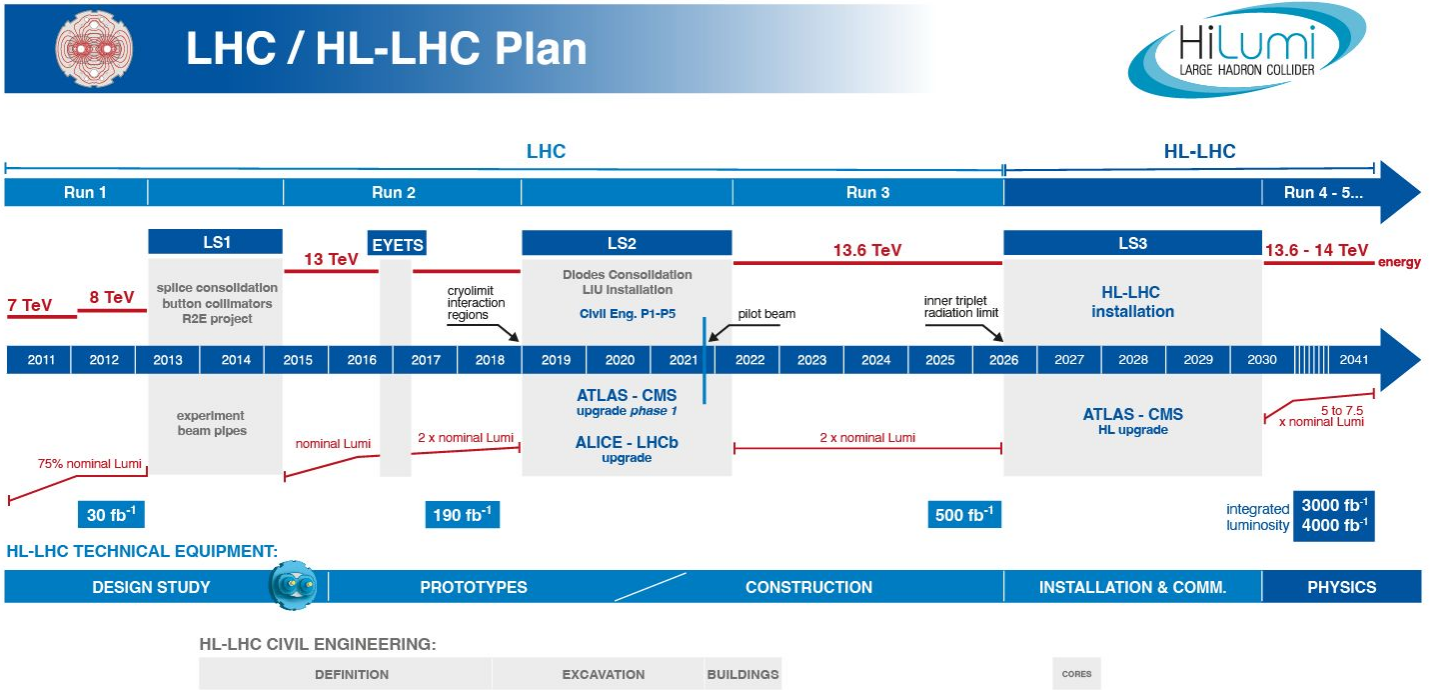
- **Energy reconstruction** with optimal filtering (OF) algorithm
  - **Weighted sum** of samples around the pulse peak
  - **Max finder/Timing cut** to select the correct BC

- 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)



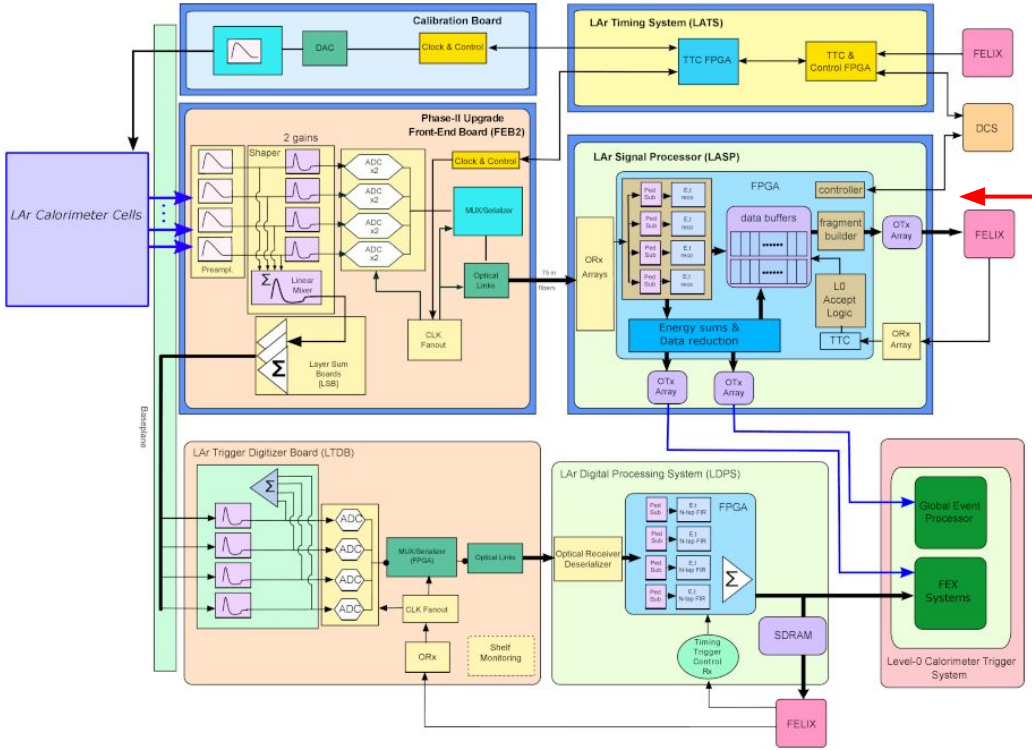


HL-LHC schedule

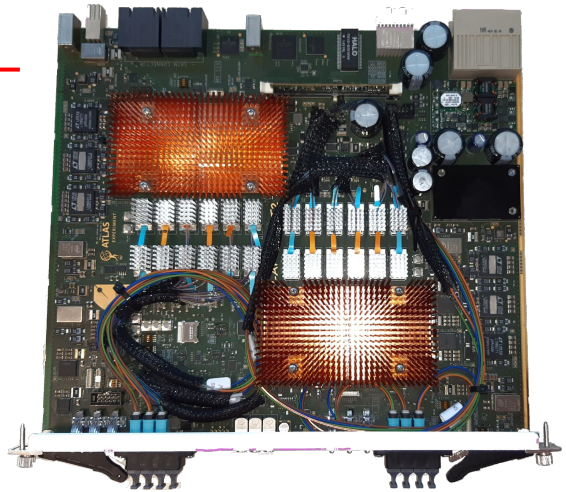


- **Increased luminosity ⇒ Increased pileup**
- HL-LHC is needed to study Higgs properties and detect new rare processes

New LAr readout electronics for energy computation



LASP board Demonstrator

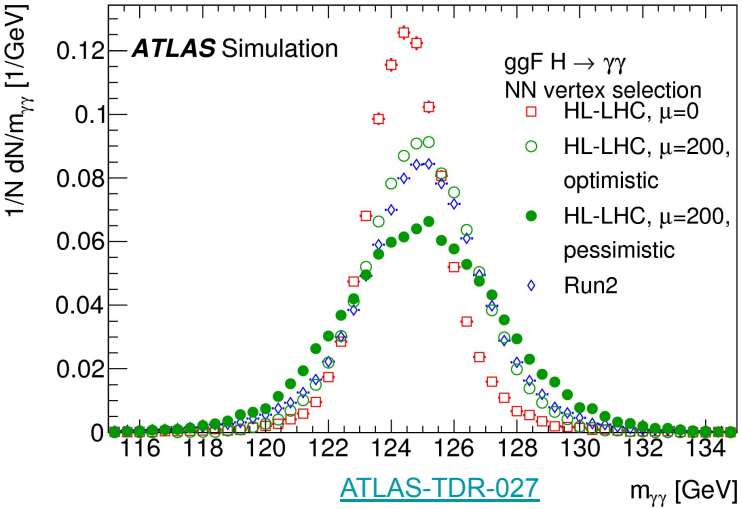


Off-detector readout board (LASP) will carry two state-of-the-art **FPGAs for energy computation**

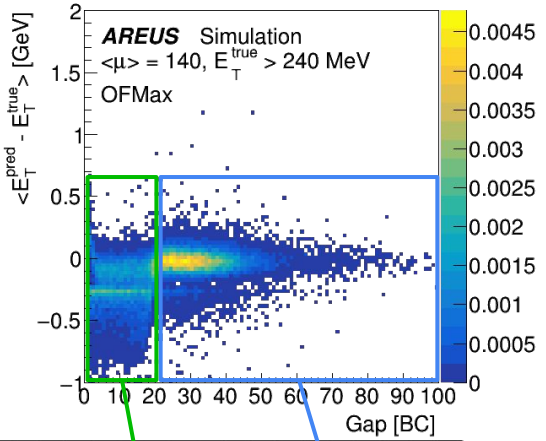
An **opportunity to embark more complex algorithms**

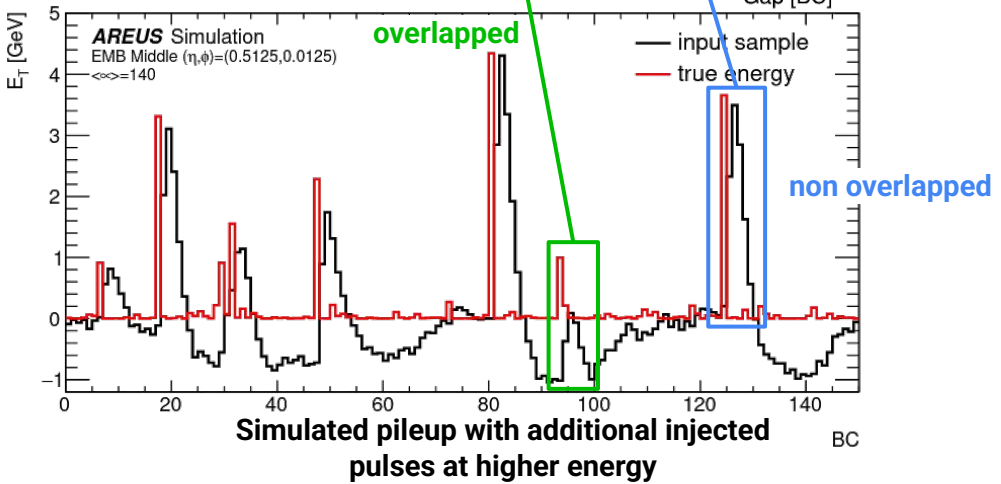
Impact of high luminosity

- 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



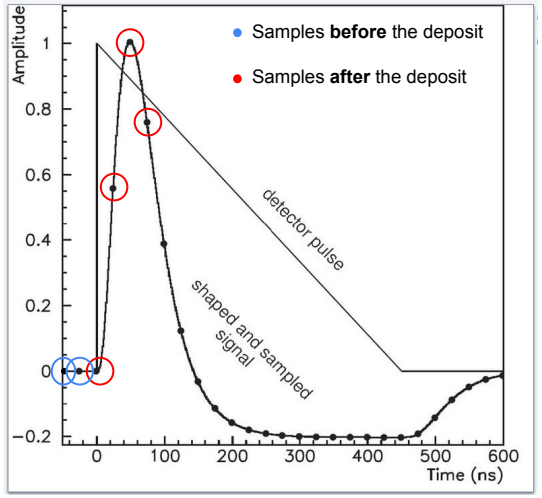


Simulated pileup with additional injected pulses at higher energy

Neural networks

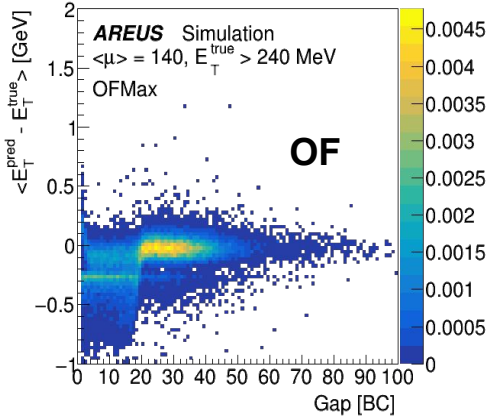
- Exploit samples before the energy deposit to **correct overlapping pulses**
- Several architectures tested : CNN, RNN, Dense layer-based

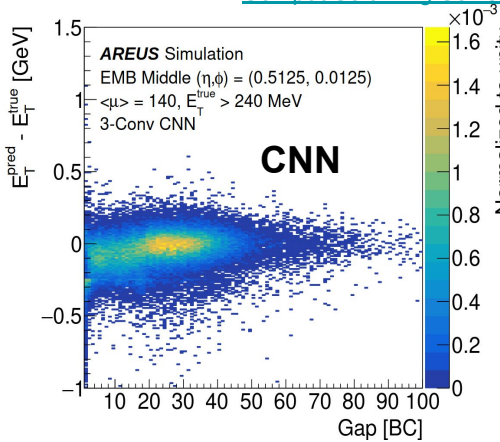
- **Samples** from **before** and **after** the energy deposit are used :
  - **After the energy deposit** (similar to OF inputs)
    - ➢ Capture the pulse amplitude
  - **Before the energy deposit** (additional inputs)
    - ➢ Correct for pulse distortions from previous deposits

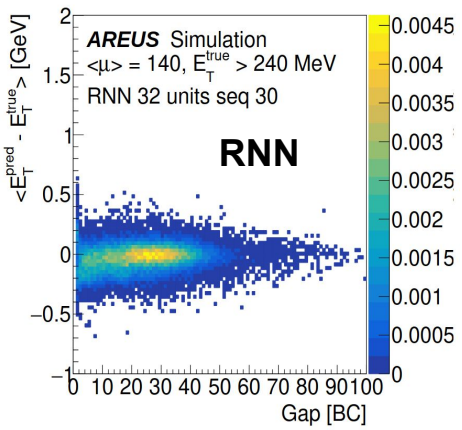


- Preliminary studies done with high rate of pulse overlap
  - **Neural networks can correct for overlapping pulses**
    - The correction is **dependent on the size** of network

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

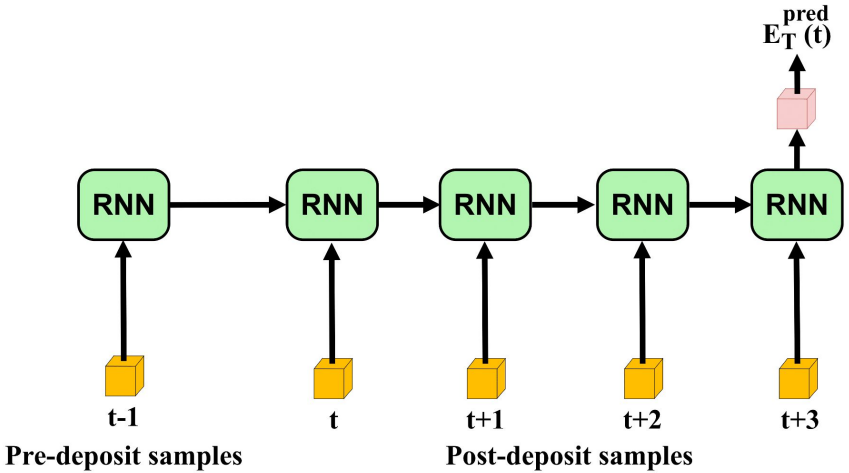






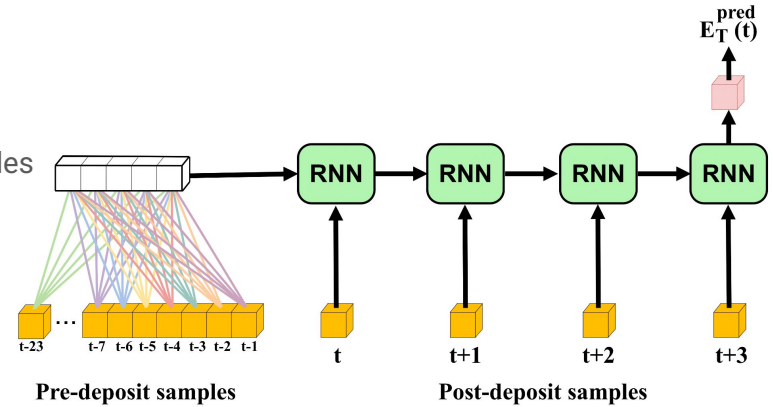
Neural network architecture - RNN

- Architecture
  - Multiple RNN cells **sharing the same parameters**
    - ➢ **One cell per sample**
  - One dense layer to concatenate output from last cell
    - ➢ **Return predicted energy**
- The RNN architecture assigns **equal importance to all samples**
- Start computations at first sample
  - **Very good latency**
- FPGA implementability :
  - **RNN layer**
    - ➢ MAC units $_{RNN}$= $output_{dim}(ouput_{dim}+1)$ * $input_{dim}$
  - **Dense layer**
    - ➢ MAC units $_{dense}$ = $output_{dim}$ * $input_{dim}$
  - **Total**
    - ➢ **MAC units $_{\text{RNN architecture}}$ = 368**
      - Firmware optimisation ⇒ **304** MAC units



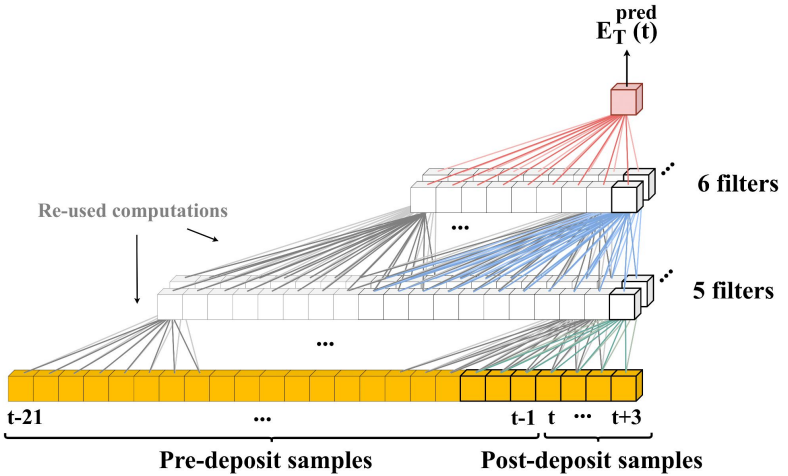
Neural network architecture - Dense+RNN

- Architecture
  - One dense layer for **aggregation of the pre-deposit samples**
    - ➢ Connected to the initialisation of the first RNN cell
  - Multiple RNN cells **sharing the same parameters**
    - ➢ **One cell per sample**
  - One dense layer to concatenate output from last cell
    - ➢ **Return predicted energy**
- The RNN architecture assigns **more importance to post-deposit samples**
  - Reduced computations on pre-deposit samples to enable more samples
- Computation from the pre-deposit samples start before latter samples arrive
  - **Good latency**
- FPGA implementability :
  - **RNN layer**
    - ➢ MAC units $_{RNN}$= $output_{dim}(ouput_{dim}+1)$ * $input_{dim}$
  - **Dense layer**
    - ➢ MAC units $_{dense}$ = $output_{dim}$ * $input_{dim}$
  - **Total**
    - ➢ **MAC units $_{Dense+RNN\ architecture}$ = 240**



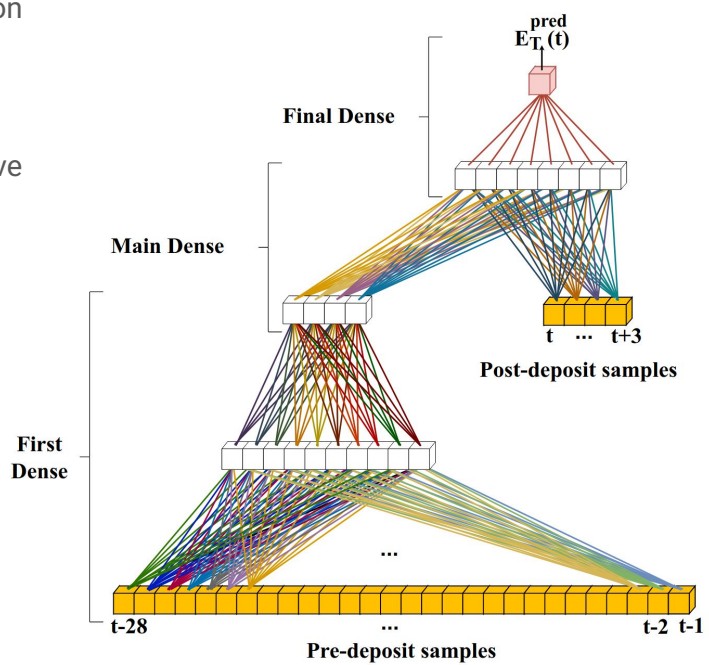
Neural network architecture - CNN

- Architecture
  - Two large CNN layers to add layers of computation
  - One last CNN layer to concatenate output from last cell
    - ➢ **Return predicted energy**
- The CNN architecture assigns **equal importance to all samples**
- Uncertain latency compliance
- FPGA implementability :
  - **CNN layer**
    - ➢ MAC units $_{CNN}$= filter$_{number}$ * kernel$_{size}$ * (input$_{dim}$ - kernel$_{size}$ + 1)
  - **Total**
    - ➢ **MAC units $_{CNN\ architecture}$ = 3689**
  - **Firmware**
    - ➢ **MAC units $_{CNN\ architecture}$ = 419**
    - ➢ **Re-used computations**



Neural network architecture - Dense

- Architecture
  - One block of two dense for **aggregation of pre-deposit samples**
  - One dense layer to add a layer of computations before concatenation
  - One dense layer to concatenate output from last cell
    - ➢ **Return predicted energy**
- Computation from the pre-deposit samples start before latter samples arrive
  - **Good latency**
- FPGA implementability :
  - **Dense layer**
    - ➢ MAC units $_{dense}$ = output$_{dim}$ * input$_{dim}$
  - **Total**
    - ➢ **MAC units $_{Dense+RNN\ architecture}$ = 392**

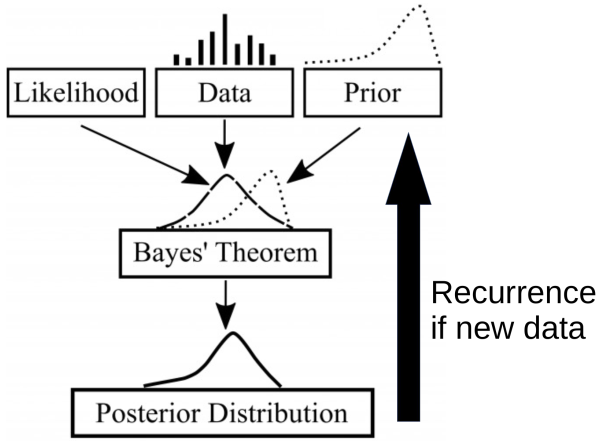


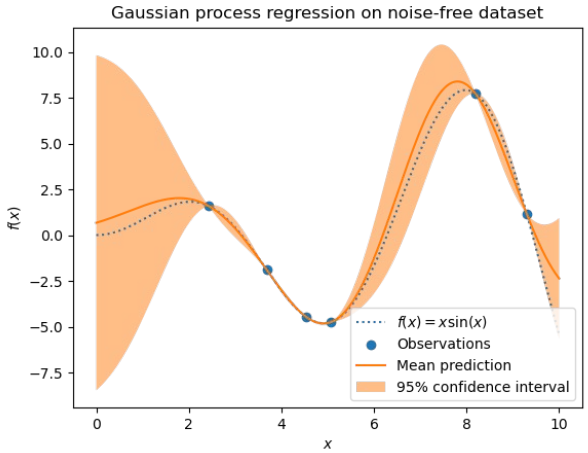
Bayesian optimisation

- 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**





Bayesian optimisation applied on energy reconstruction

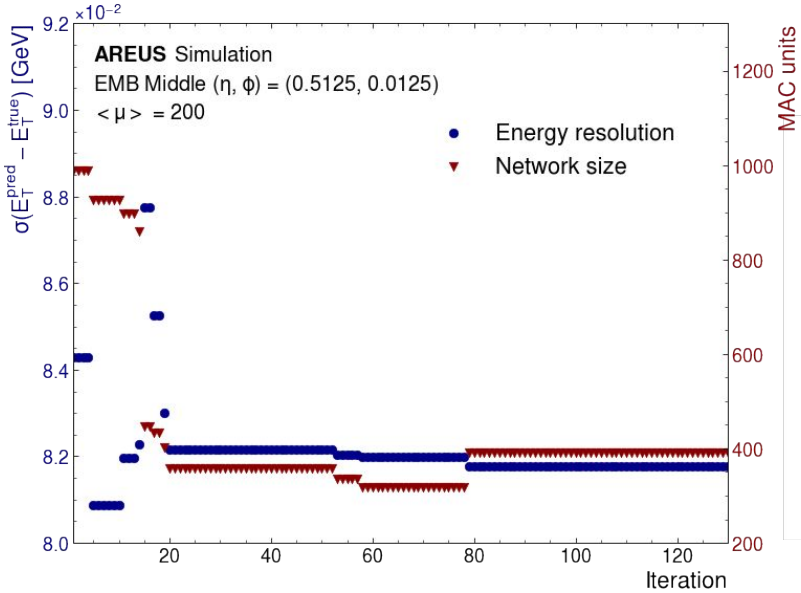
- Optimization on both performance and hardware to fit in FPGAs
  - **Energy resolution** (σ [MeV])
  - **Number of MAC units** (M)
- Hyperparameters to be tuned (e.g. for the Dense architecture) :
  - **Number of samples** (before the energy deposit)
  - **Number of units** for the intermediate layers

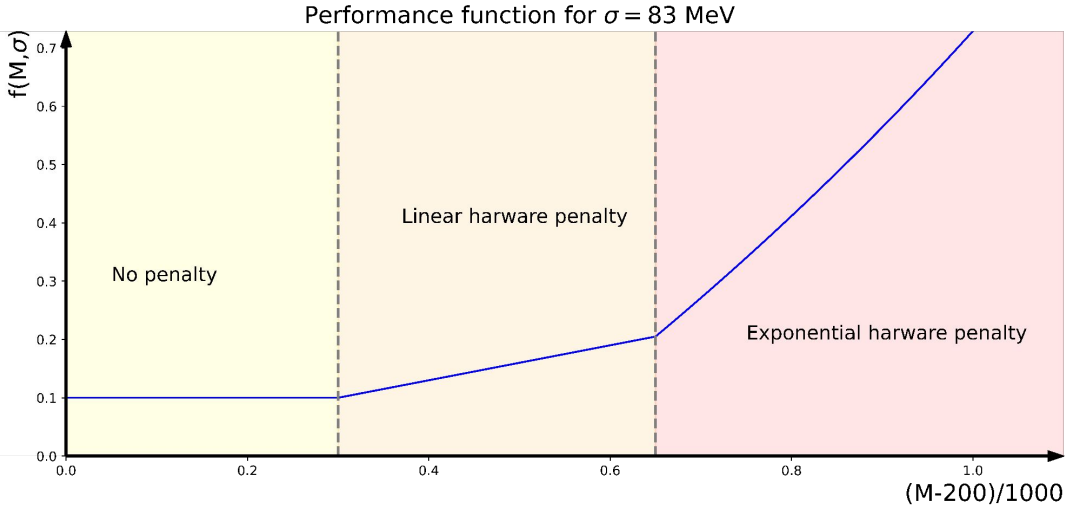
Performance function used for the bayesian optimization :

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

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

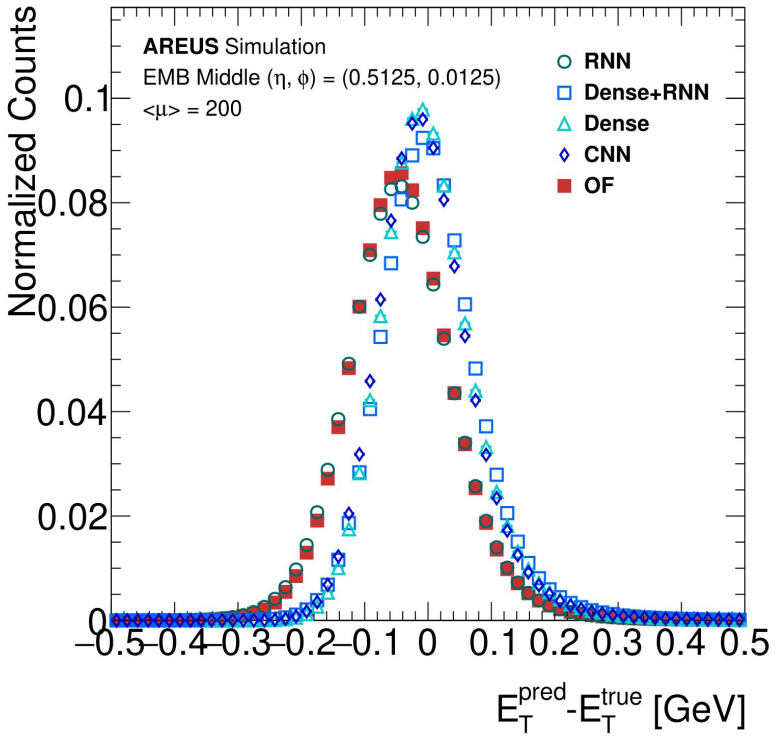
$$f(M,\sigma) = f(850,\sigma) + b*e^{\frac{M-850}{1000}} - 1 \; for \; M > 850$$





Energy resolution

- **OF and RNN**
  - RNN optimisation didn’t converge
    - ➢ Use of legacy RNN : 8 units 5 samples
  - Energy resolution : **~90 MeV**
- **Dense+RNN**
  - Energy resolution : **~85 MeV**
- **Dense and CNN**
  - Energy resolution : **~80 MeV**

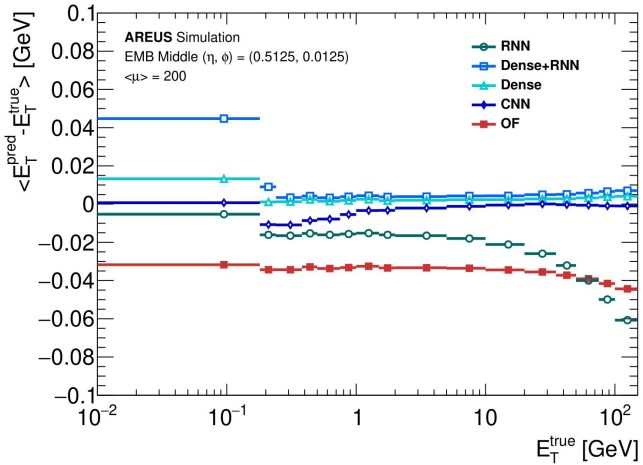


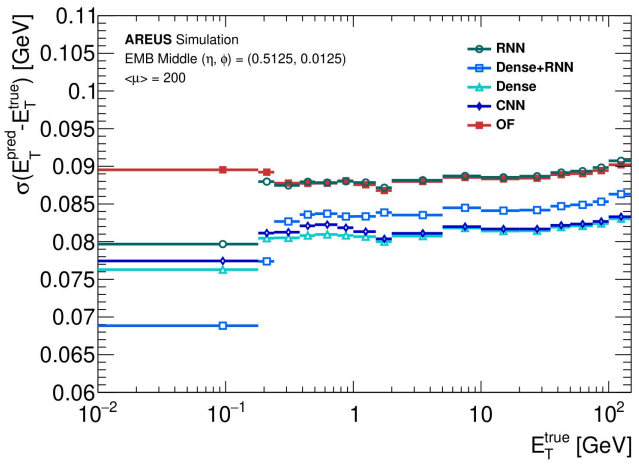
Optimised neural networks for online processing of ATLAS calorimeter data on FPGAs

Python code

Energy scale and resolution as function of true energy

- **Better energy scale** of E_T^{pred}- E_T^{true} **for Dense+RNN, Dense and CNN** architectures compared to OF
  - RNN energy scale falling with higher E_T^{true}
  - OF energy scale not centered around 0
- **Better energy resolution** of E_T^{pred}- E_T^{true} **for Dense+RNN, Dense and CNN** architectures compared to OF
  - Visible for the whole energy range





Conclusion

- **LAr energy reconstruction performed using neural networks**
  - Four architectures presented : **RNN, RNN+Dense, Dense and CNN**
  - **Limited size** : ~500 MAC units
- **Optimisation performed** on the architectures **using Bayesian optimisation**
  - Balance between performance and size of the network
    - ➢ Fit in FPGA
    - ➢ Better performance than OF