# Optimisation of embedded neural networks for the energy reconstruction of the LAr cells of ATLAS

-

IN2P3/IRFU ML workshop 2025

**Raphaël Bertrand** (CPPM)

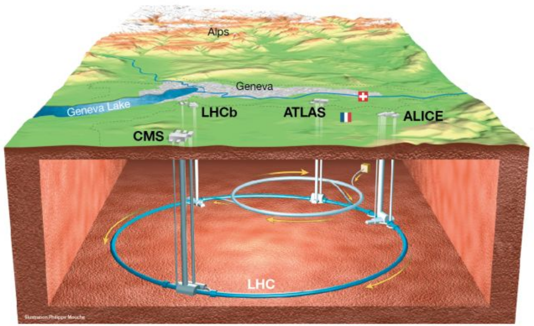
Aix Marseille Université

CENTRE DE PHYSIQUE DES PARTICULES DE MARSEILLE

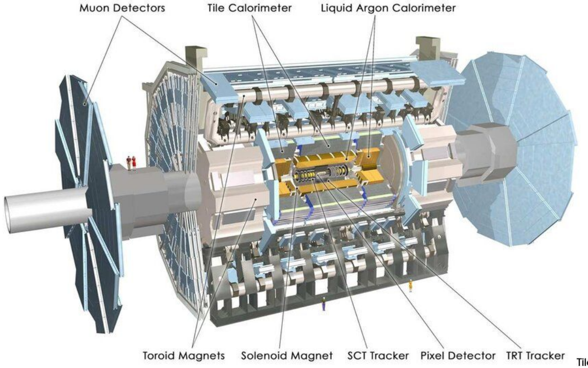
ATLAS EXPERIMENT

# Introduction

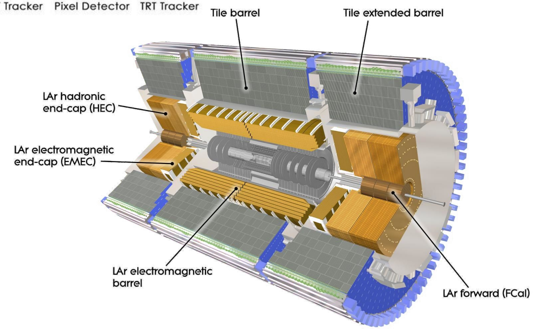
# Experimental Context




- **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



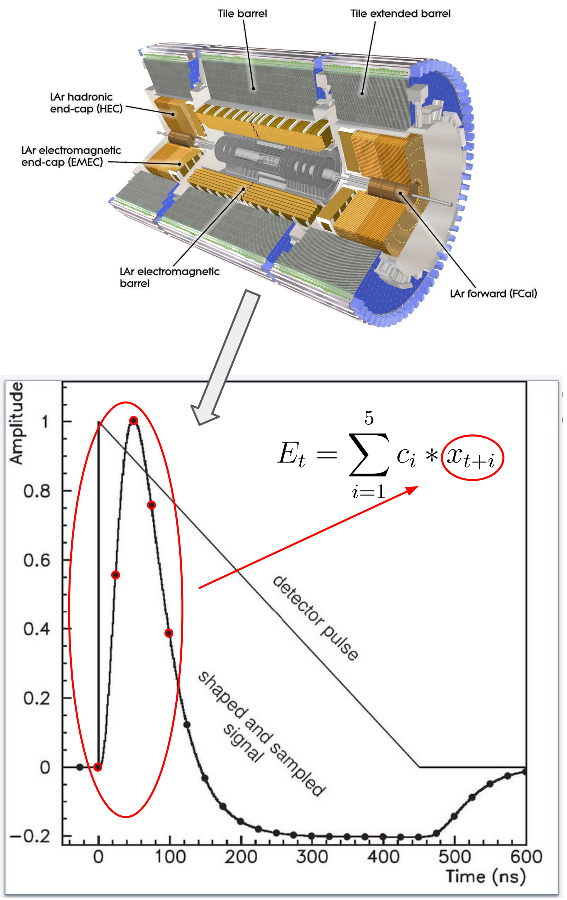

- **Liquid Argon (LAr) Calorimeter**
  - ATLAS sub-detector for energy measurement ($e^{-/+}$, $\gamma$)
    - ➢ ~180,000 LAr calorimeter cells
    - ➢ Ionization signal from particle interactions

# Signal processing and energy reconstruction

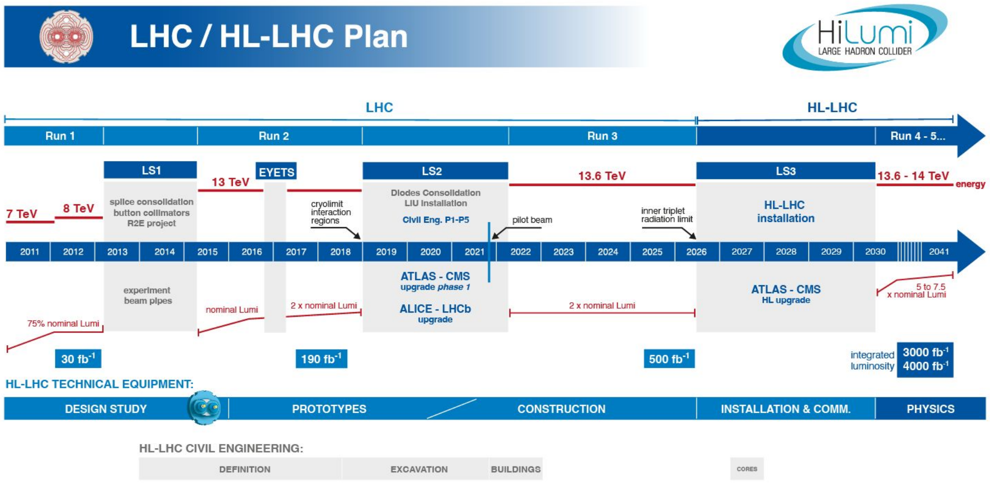
- **Electronic signal** produced
  - **Amplitude ∝ true deposited energy ($E^{true}$)**
  - **Spans ~625 ns** (25 proton-proton **B**unch **C**rossings)
  - **Shaped**, **sampled** and **digitised at 40 MHz**

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

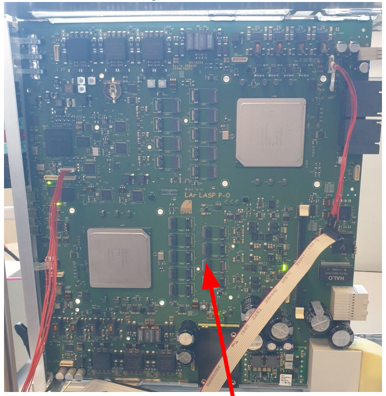
- Reconstruction algorithm requirements :
  - **Online** computation (per BC)
  - **Max latency : ~125 ns** (used in trigger system)
  - **Fit in FPGAs** : **O(500)** Multiply-Accumulate operations (**MAC units**)
    - ➢ 5 MAC units required to implement OF
  - **384 channels per FPGA** (many algorithm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

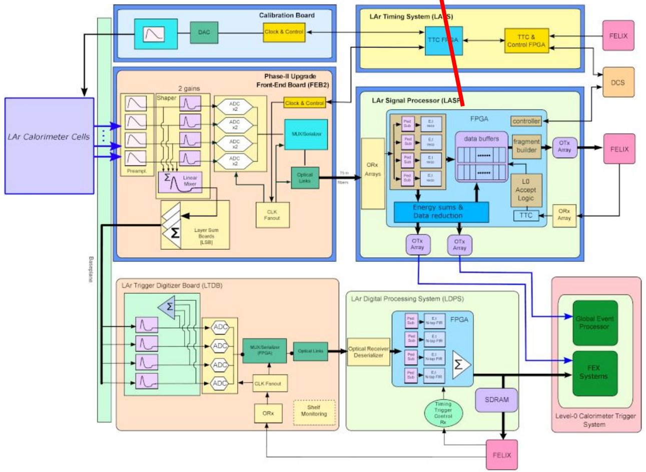


# HL-LHC schedule and ATLAS Phase-II Upgrade




LASP board produced at CPPM

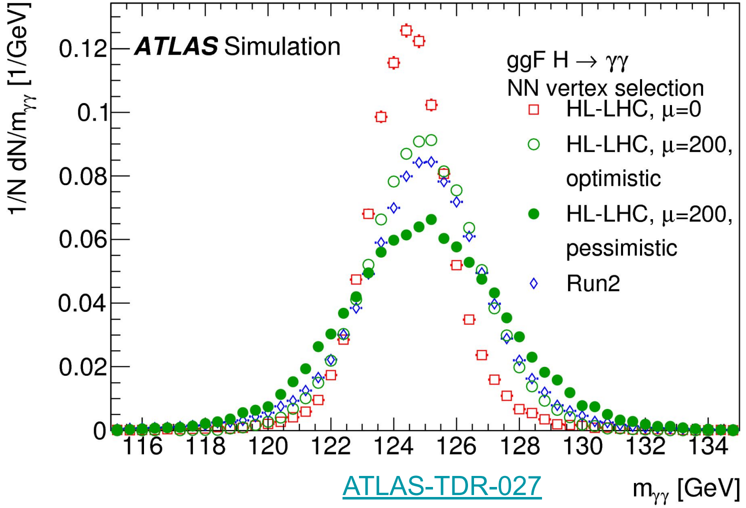




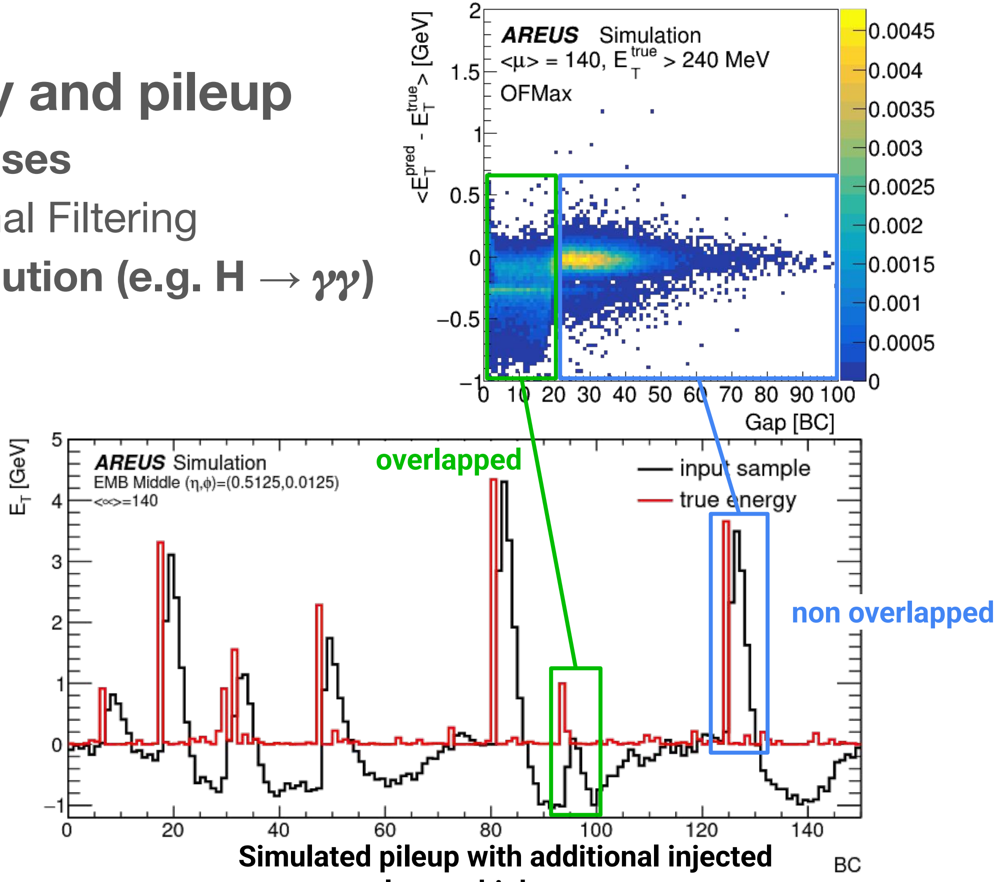

- HL-LHC ⇒ Increased luminosity
- ATLAS LAr Phase-II upgrade needed
  - Exchange of full readout electronics
- Off-detector readout board (LASP) will carry two state-of-the-art **FPGAs for energy computation**
  - **Opportunity to embark more complex algorithms**

# 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



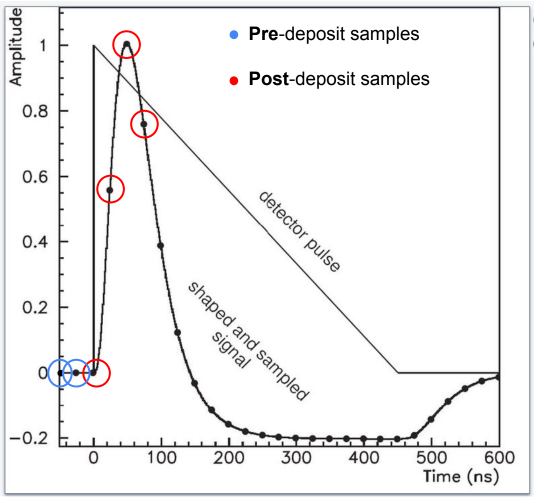

**Simulated pileup with additional injected pulses at higher energy**

# Neural network approaches as energy reconstruction algorithms

# Neural networks

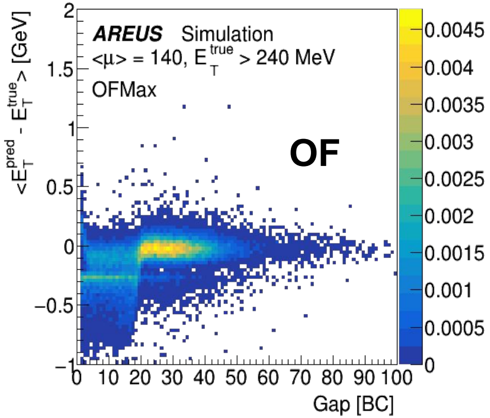
- Exploit samples before the energy deposit to **correct overlapping pulses**
- Several architectures tested : RNN, Dense+RNN, CNN, Dense

- **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

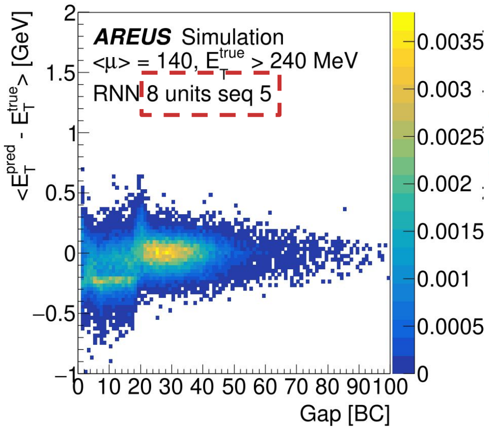


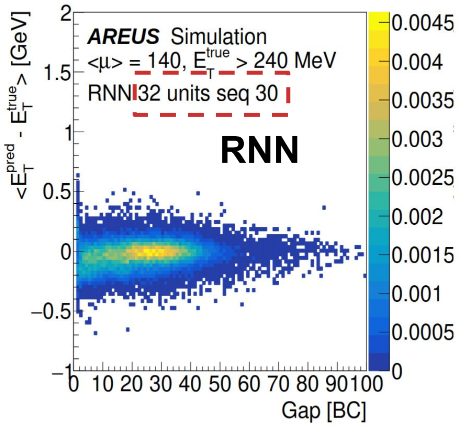

- Preliminary studies done with high rate of pulse overlap
  - **Neural networks can correct for overlapping pulses**
    - The correction is **dependent on the size** of network

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




**Longer sequences and higher number of units are needed**

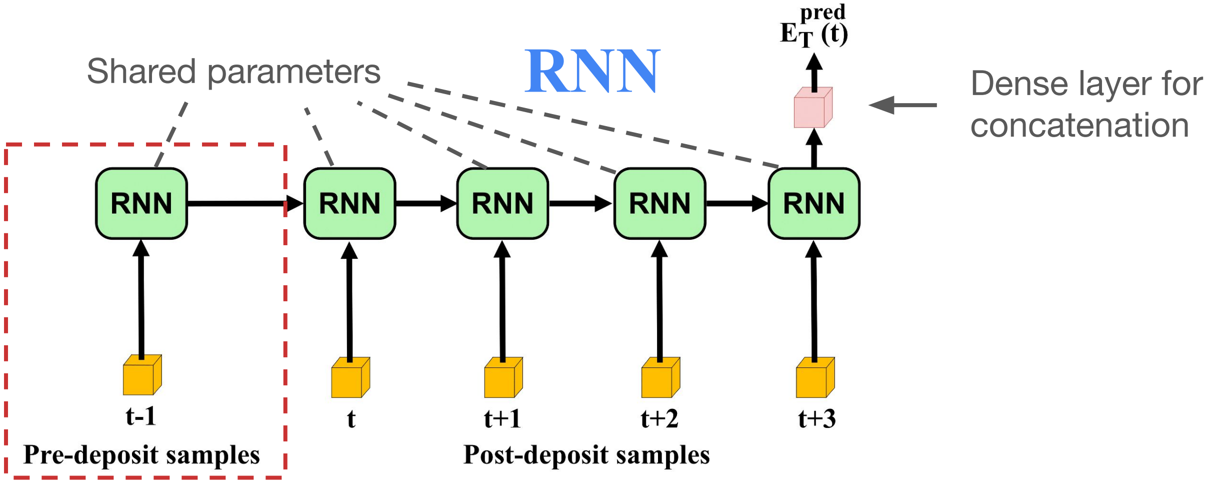


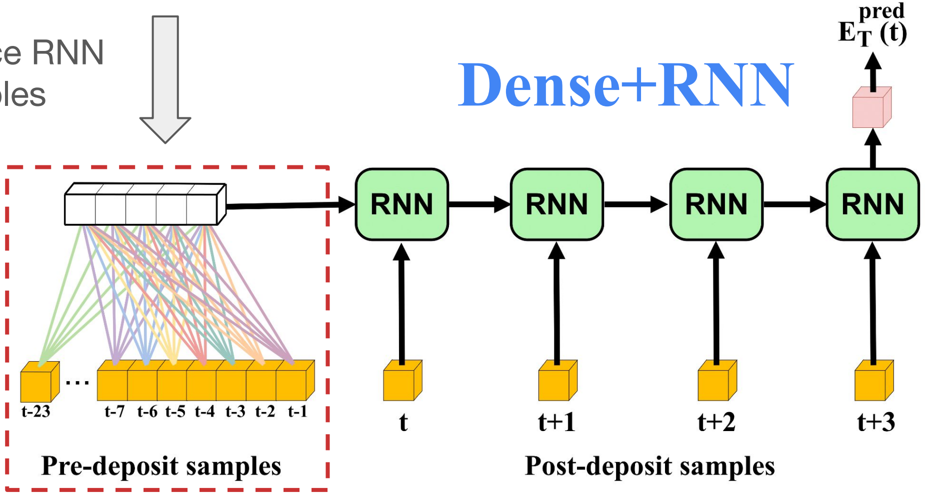
# RNN architectures

Optimised for latency on FPGA
Computations start at :

- **RNN : arrival of 1$^{st}$ sample**
- **Dense+RNN : arrival of all pre-deposit samples**



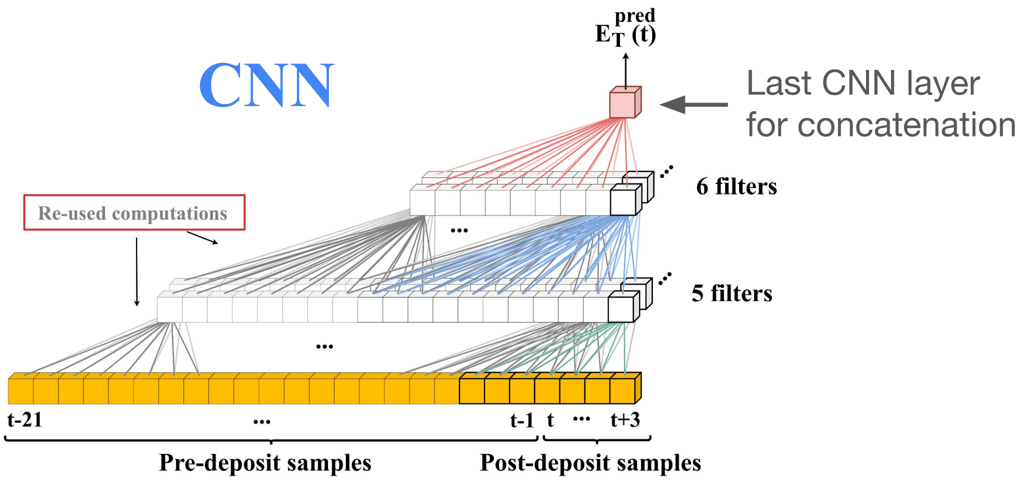

Dense layer to replace RNN for pre-deposit samples

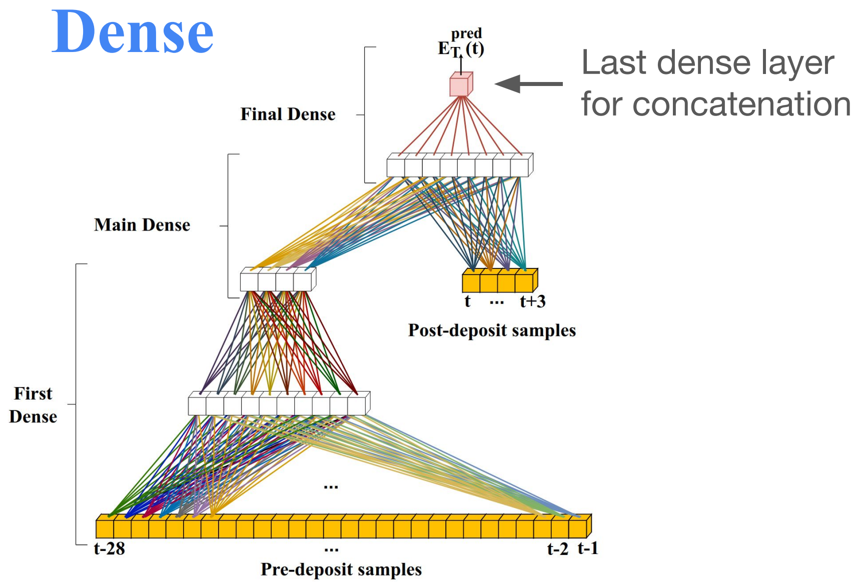



$\Rightarrow$ MAC units $_{\text{RNN layer}}$ $\propto$ $\boxed{\text{units}^2}$ x nb of samples

$\Rightarrow$ MAC units $_{\text{Dense layer}}$ $\propto$ $\boxed{\text{units}}$ x nb of samples

# CNN and Dense architectures




**Computations start** at the arrival of **the last post-deposit sample**

**Computations start** at the arrival of **every pre-deposit samples**

# Neural networks architectures

unstable training for low number of units

still better than OF and RNN

| | Description | Latency | MAC units | Tunability |
|---|---|---|---|---|
| **RNN** | Multiple cells sharing the same parameters | ✓ | × | ∼ |
| **Dense+RNN** | Preceding dense layer before four RNN cells | ✓ | ∼ | ✓ |
| **Dense** | Exclusively based on dense layers | ✓ | ✓ | ∼ |
| **CNN** | Multiple CNN layers | ∼ | ✓ | ∼ |

require firmware optimisation

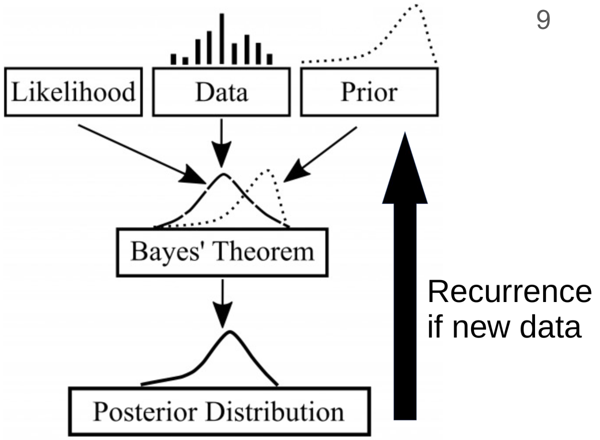
covered using Bayesian optimisation

# Neural networks hyperparameter tuning

using bayesian optimization

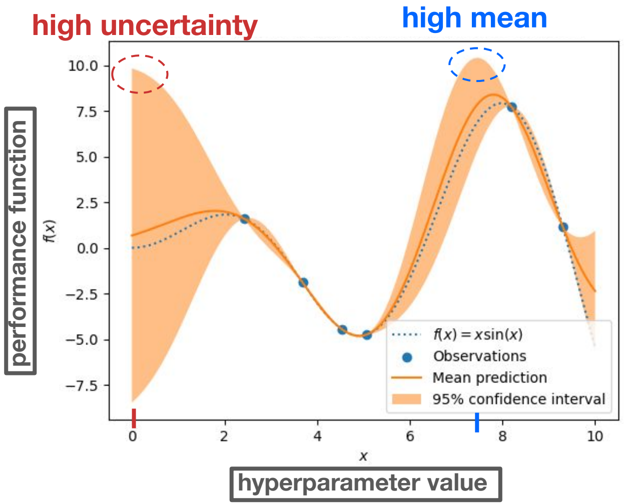
# Bayesian optimisation

- Goal : Identify the **optimal parameters** that maximize a **performance function** while keeping the number of function **evaluations** to a **minimum**
  - Useful on **expensive** (time-consuming) performance functions
  - Provide the **best parameters** (in average) **for a given number of evaluations** of the performance function




- **Initialization** with several random points followed by **iterations** to find the best parameters space
  - Using a **gaussian kernel** ⇒ surrogate of the performance function
  - Using an **acquisition function** ⇒ decision of the next most interesting point

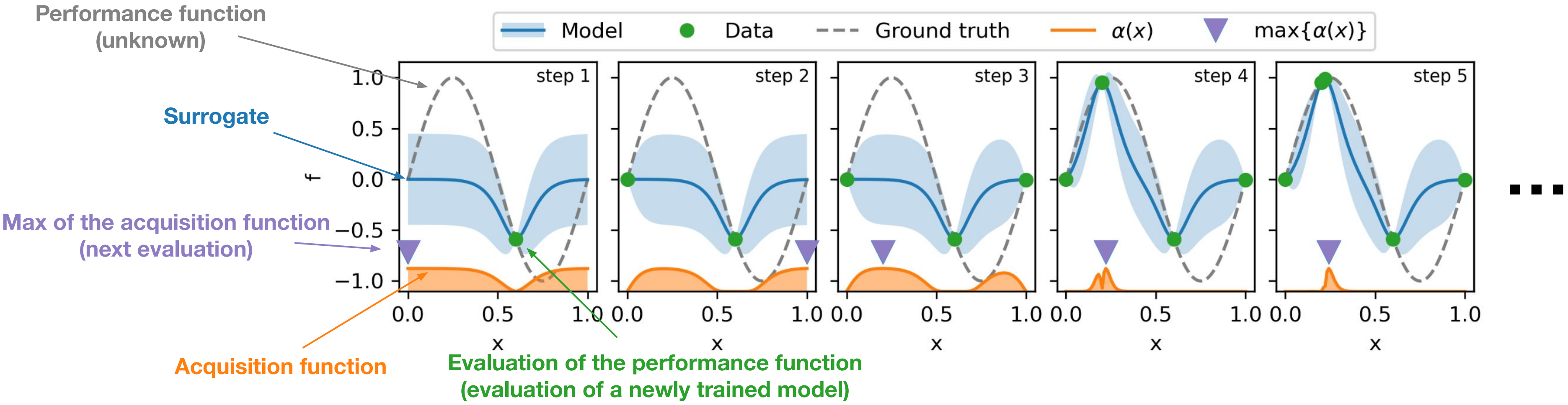
- Balance between **exploration** and **exploitation**
  - **Exploration ⇒ favours high-uncertainty region**
  - **Exploitation ⇒ favours high-mean region**



# Bayesian optimisation process

- **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

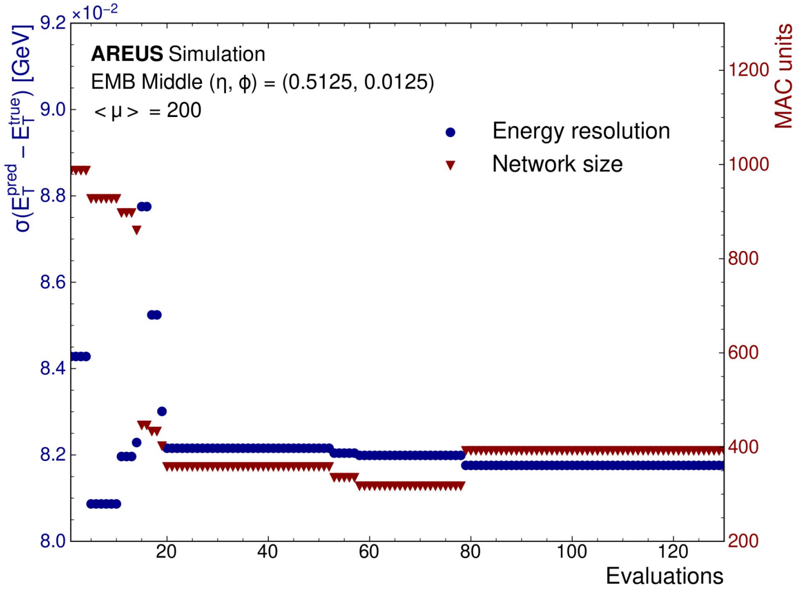
- Hyperparameters to be tuned (e.g. for the Dense architecture) :
  - **Number of samples** (before the energy deposit)
  - **Number of units** for the intermediate layers
- Optimisation on both performance and hardware to fit in FPGAs
  - **Energy resolution** (σ [MeV])
  - **Number of MAC units** (M)

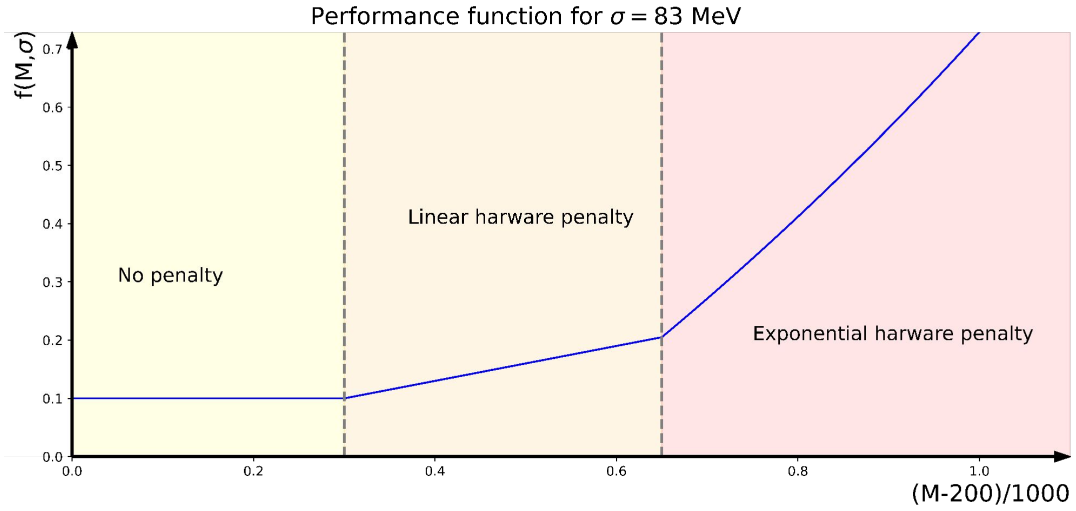
**Performance function** used for the bayesian optimization :

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

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

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

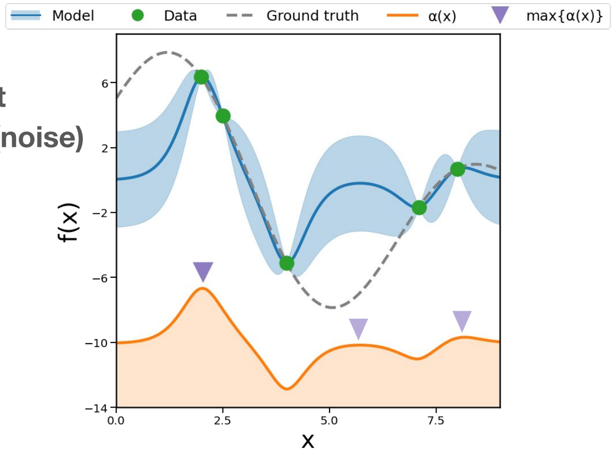


# Bayesian optimisation code

- **Kernel choice** ⇒ 5/2 Matérn
  - Twice differentiable ⇒ **More realistic**
- **Multiple neural networks trained for a given parameters set**
  - **Accounts for fluctuations with different initialisations (noise)**
  - **1st mode** ⇒ Best network with no uncertainty is used
  - **2nd mode** ⇒ Average with uncertainty is used

Example with min_distance = 3 and 3 iterations in parallel



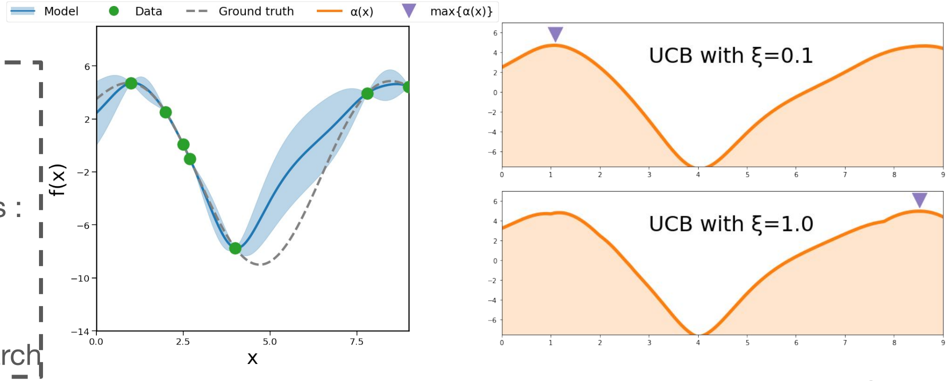

| $X$ | $\alpha$ |
|---|---|
| 2.030 | 6.348 |
| 2.053 | 6.336 |
| 2.008 | 6.344 |
| 1.985 | 6.322 |
| ... | ... |
| 8.165 | 0.760 |
| 8.188 | 0.755 |
| 8.143 | 0.762 |
| 8.211 | 0.748 |
| ... | ... |
| 5.143 | -0.593 |
| 5.120 | -0.631 |
| 5.098 | -0.671 |
| 5.075 | -0.713 |
| ... | ... |

**skipped**

**skipped**

- **Parallel iterations** using multiprocessing
  - Introduce minimal distance between parameters sets
    - ➢ Avoid evaluating similar parameters in parallel
- **Three phases** of iterations changing acquisition function settings :
  - **1st phase** : focus on exploration
  - **2nd phase** : reasonable exploration
  - **3rd phase** : focus on exploitation
- **Integer optimisation** ⇒ Non-integer values are excluded for search

Bayesian optimisation is still **computationally-heavy**, need for **time reduction**

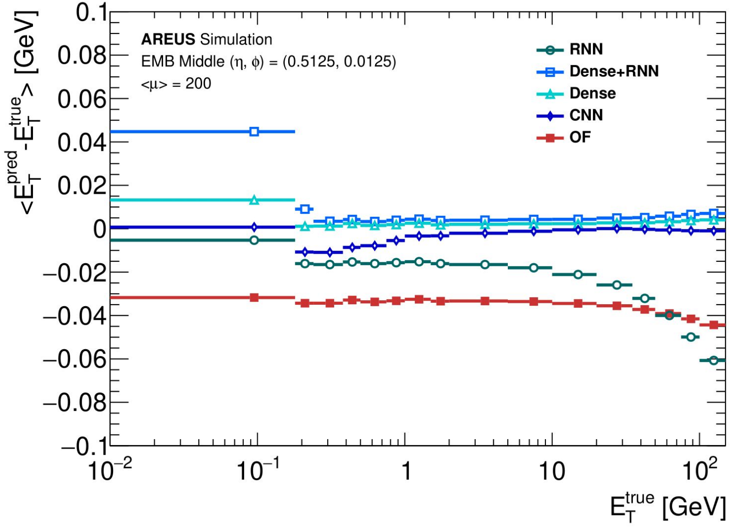



**Example using different values for exploration parameter ξ**

# 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 $E_T^{true}$
- **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

**$<E_T^{pred}$ - $E_T^{true}>$**

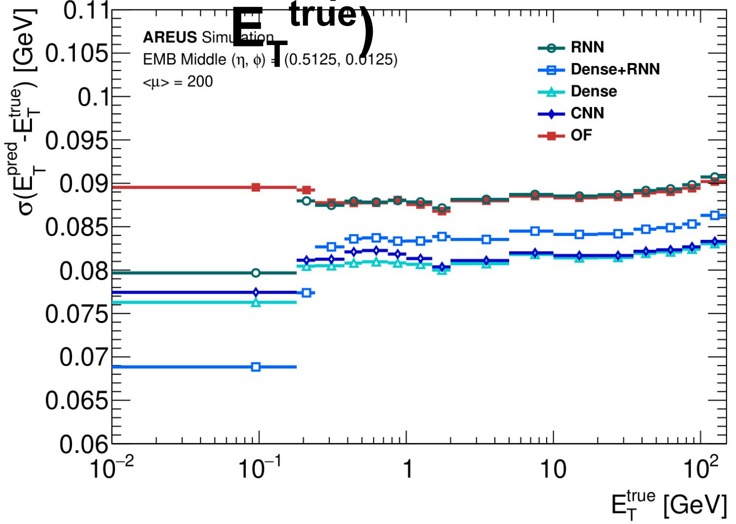



MAC units
**368**
**240**
**392**
**419**
**5**

**Pre-print**
**(submitted to EPJC)**

Public code

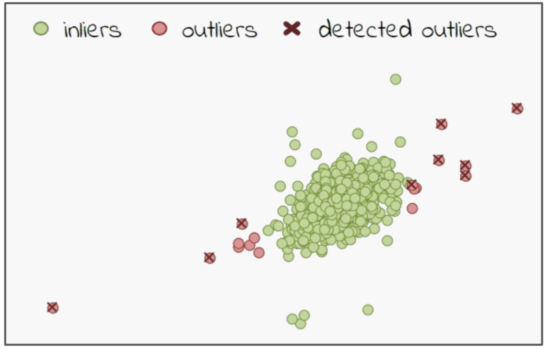
**$\sigma(E_T^{pred}$ - $E_T^{true})$**



# Uncertainty prediction using neural network

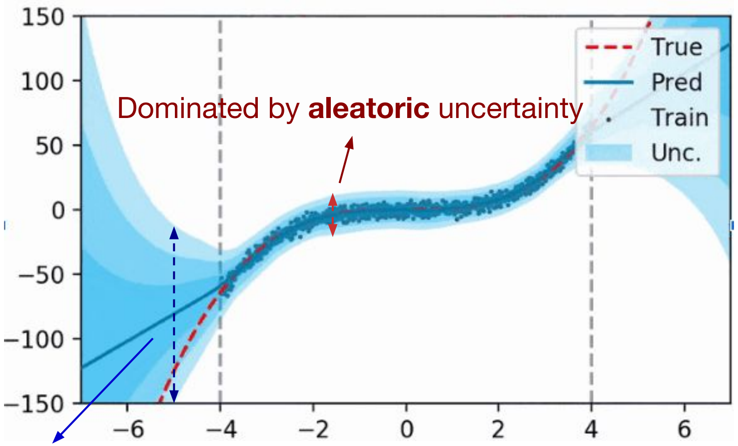
using deep evidential regression

# Deep evidential regression (DER)




- NNs are trained to minimize their prediction errors
  - Unknown accuracy of the model for individual predictions
  - It would be interesting to **know when the model is more likely to fail (or the opposite)**
- **Model the energy prediction as a distribution**
  - Network targeting parameters of this distribution
  - Mean of the distribution → **energy prediction**
  - Standard deviation of the distribution → **uncertainty**

- Differentiate uncertainties :
  - **Epistemic**
    - ➢ Model uncertainty, systematics
    - ➢ Can be reduced
  - **Aleatoric**
    - ➢ Inherent to data, data uncertainty
    - ➢ Cannot be reduced




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

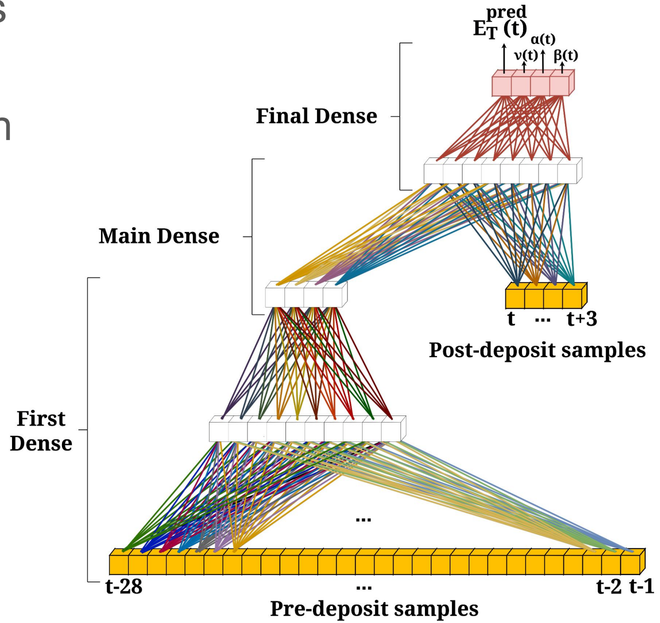
# Deep evidential regression (DER)

- **DER applied to LAr cells energy reconstruction**
  - Detect outliers due to noise bursts, instantaneous luminosity changes, or bunch train structures
- Normale-Inverse Gamma distribution to describe mean and uncertainty
  - **4 parameters** (γ,ν,α,β)
- Adapted to the Dense architecture
  - **Still possible to implement in FPGA**
    - ➢ **416 MAC units**
- Training loss function : ~~MSE~~
  - Likelihood + Regularisation

fit the distribution

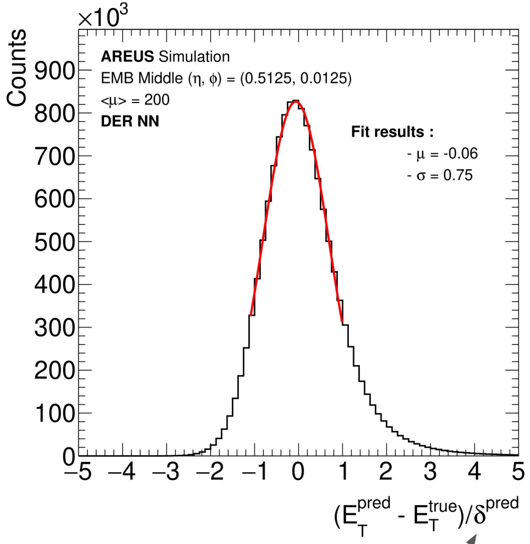
increase uncertainty on outliers

- **Similar resolution** observed to NN without DER

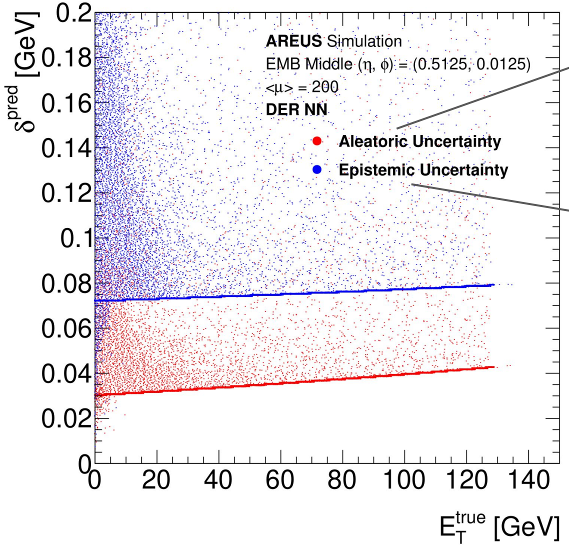


# Uncertainty prediction

- Overall **good pull distribution**
  - Estimated uncertainty comparable to $E_T^{pred}$ - $E_T^{true}$
    - ➢ Uncertainty overestimated by 25%
  - **Slightly biased**
    - ➢ Right tails




predicted uncertainty




Data uncertainty

$$\mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}$$

Model uncertainty

$$\mathrm{Var}[E_{\mathrm{T}}^{\mathrm{pred}}] = \frac{\beta}{\nu(\alpha - 1)}$$

- Epistemic and aleatoric uncertainties are mainly constant
- **Epistemic uncertainty is dominant**

# Conclusion

- Four neural network architectures were tested and optimized
  - **CNN**, **RNN**, **Dense+RNN** and **Dense**
- **Hyperparameter 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
  - **Good** uncertainty prediction
  - **Possible to implement on FPGAs**
- **Paper submitted to EPJC : pre-print on arxiv : Optimised neural networks for online processing of ATLAS calorimeter data on FPGAs**
- **Code available on Zenedo**