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

Nairit Sur

CPPM - CNRS/IN2P3

on behalf of the ATLAS Liquid Argon Calorimeter Group

LEARNING TO DISCOVER

Institut Pascal, Université Paris-Saclay

April 19 - April 29, 2022

Orsay

The Liquid Argon Calorimeter:

A crucial component of the ATLAS detector

- ~160 fb^{-1} p-p collision data reconstructed with high quality and precision
- Designed to measure the **time**, **position**, and **energy** deposited by **electrons** and **photons**, and in addition, **hadrons** in the end-cap region
- ~180K readout channels - Lead, copper, and tungsten as absorbers, cryogenically cooled liquid argon as active material

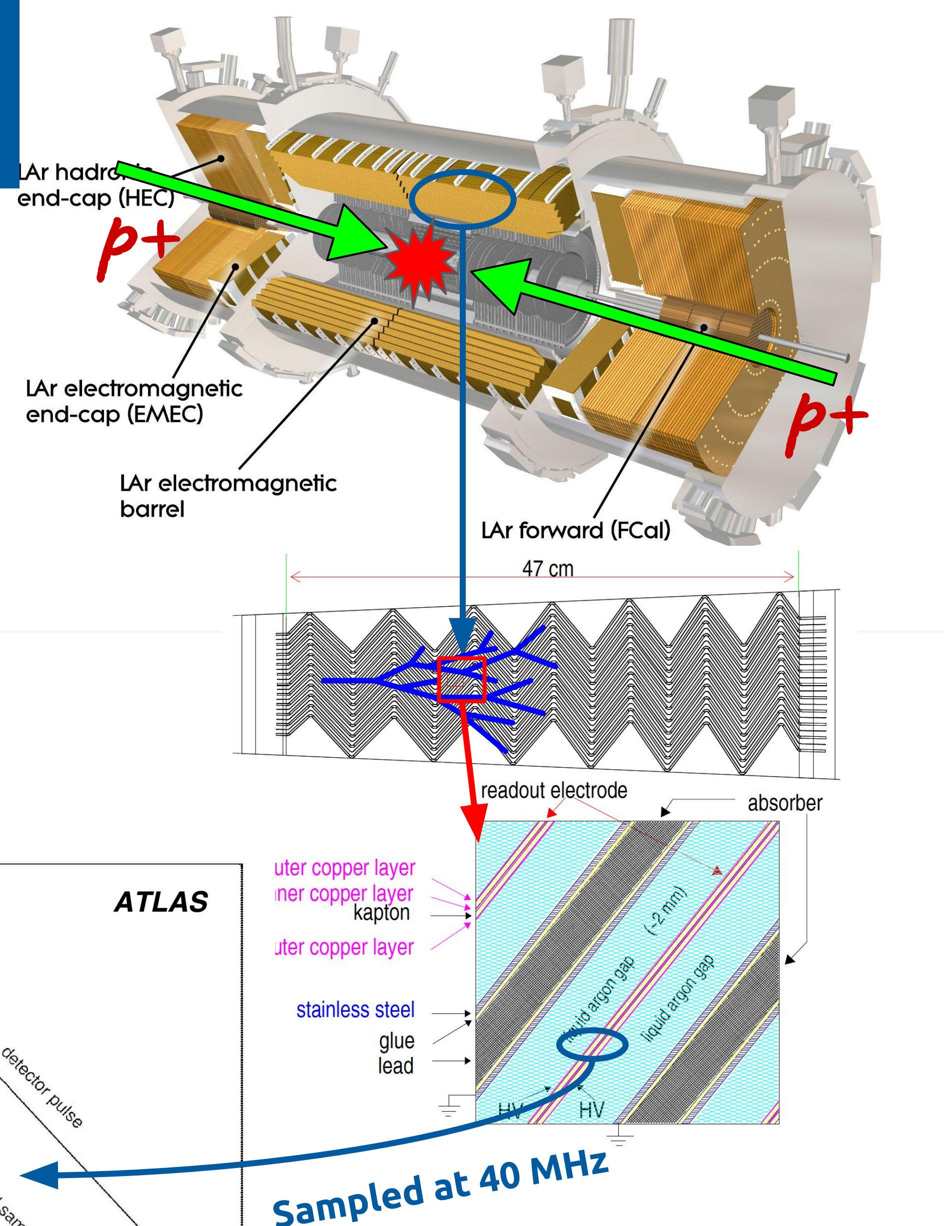
Energy from Optimal-Filter (OF)

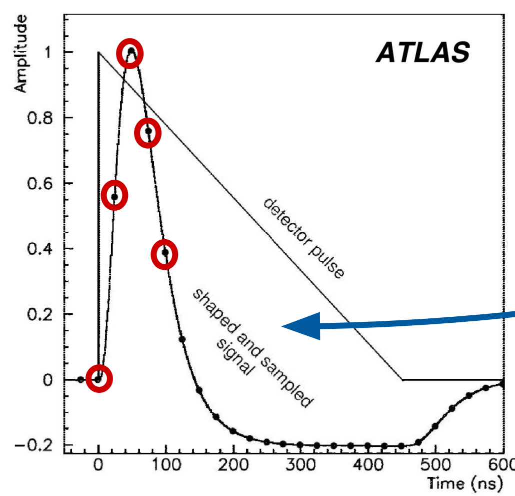
n = 5 in this talk

$$E(t) = \sum_{i=t}^{t+n} a_i \cdot s_i$$

Pulse Samples

Pre-set coefficients (fit of the peak)





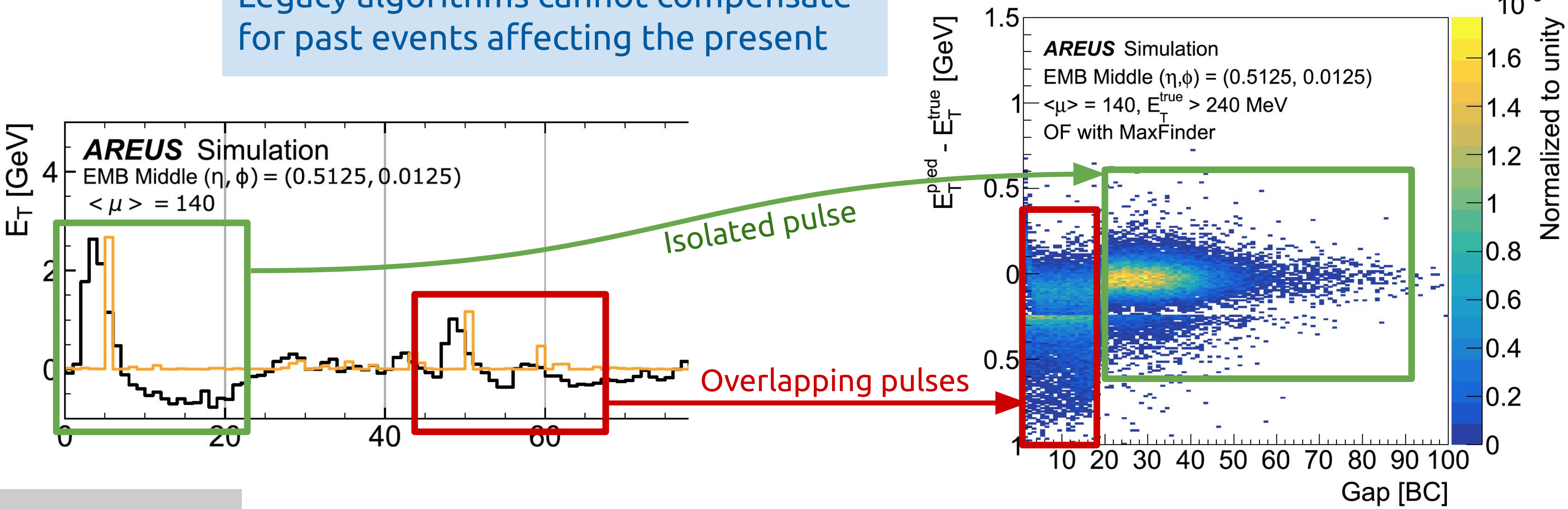
Towards HL-LHC

The high luminosity phase of the LHC (**HL-LHC**) will produce **140-200** simultaneous p-p interactions (pile-up), compared to the current value **~40**

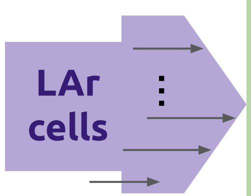
Energy deposits **continuously** sampled and digitized at 40 MHz :
⇒ requires peak finder/trigger (to select the correct BCIDs)
Real-time energies for triggers :
⇒ requires compact algorithms on high-end FPGAs

Legacy algorithms cannot compensate for past events affecting the present

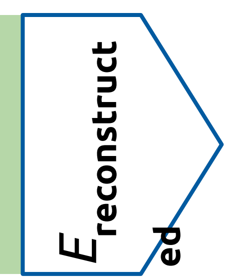
Resolution of deposited energy



Upgrade of readout electronic chain for AI algorithms

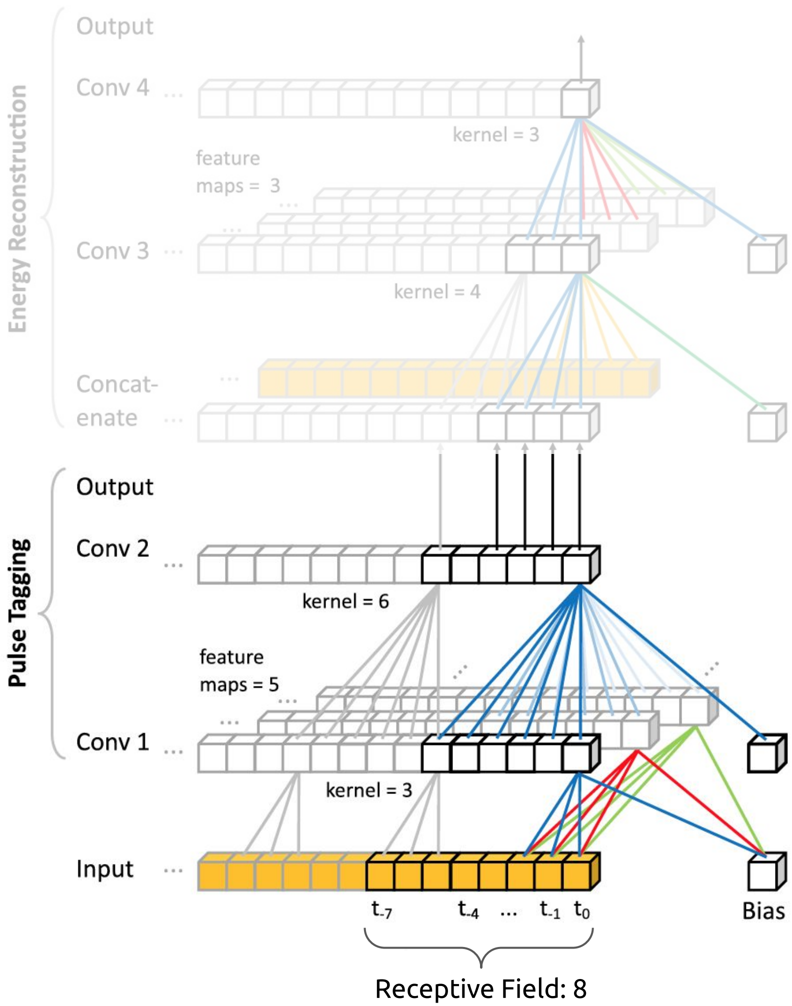


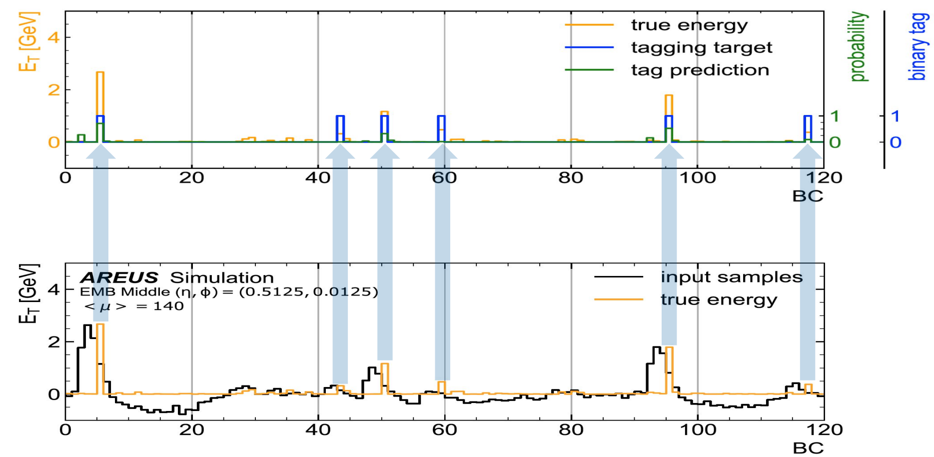
New off-detector electronics on the backend board:
LAr Signal Processor (LASP)
- Two Intel Stratix 10 FPGAs
- ~Tb/s(~500 channels)
- ~200 boards

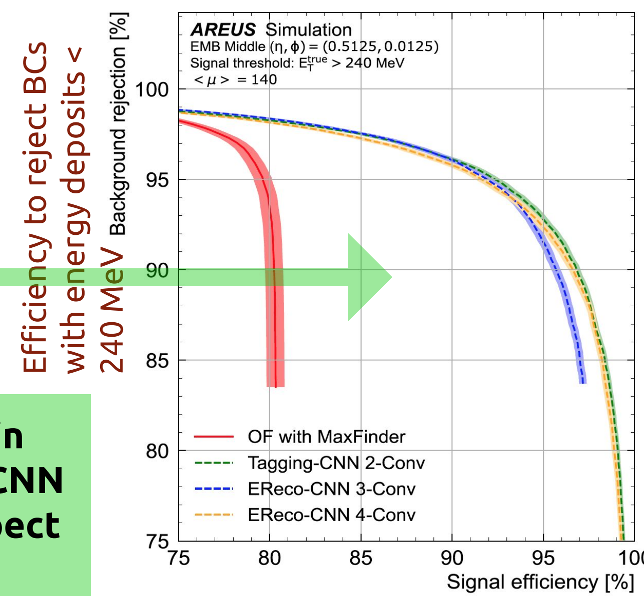


CNN: pulse tagging

CNN for pulse tagging:
Trained to detect energy deposits 3σ above noise (240 MeV) using pulse samples for 8 bunch crossings



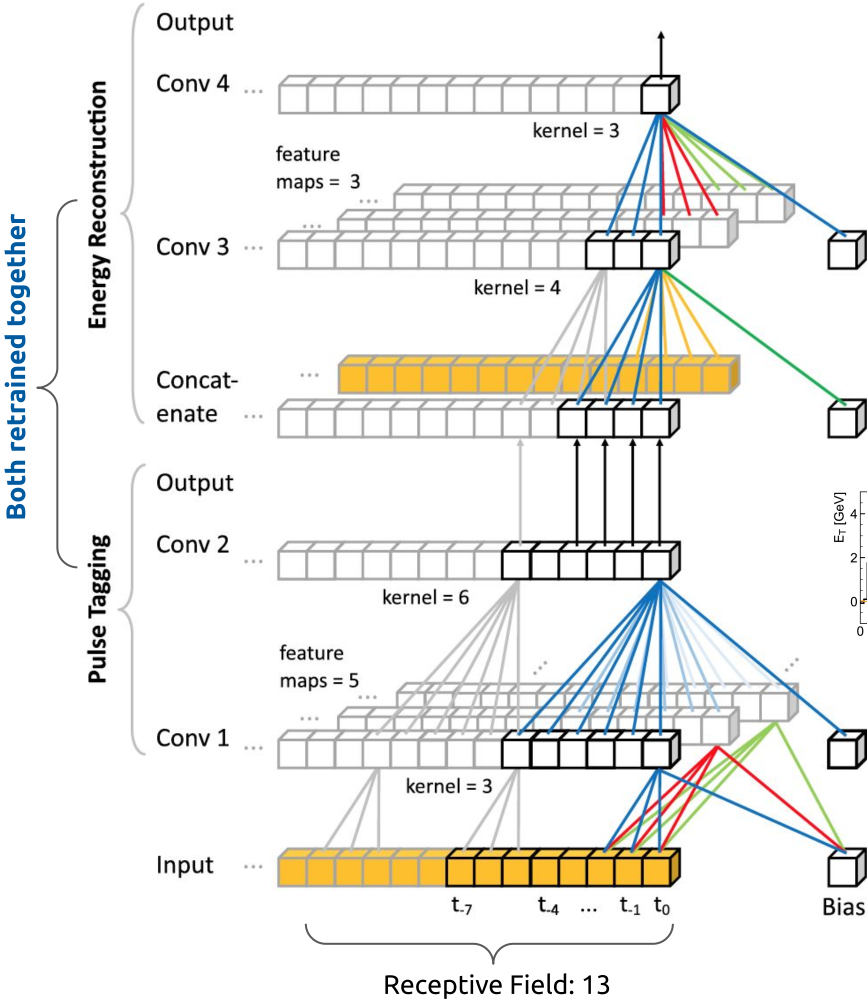




significant gain in efficiency with CNN tagger with respect to OF with MaxFinder

Efficiency to select BCs with energy deposits > 240 MeV

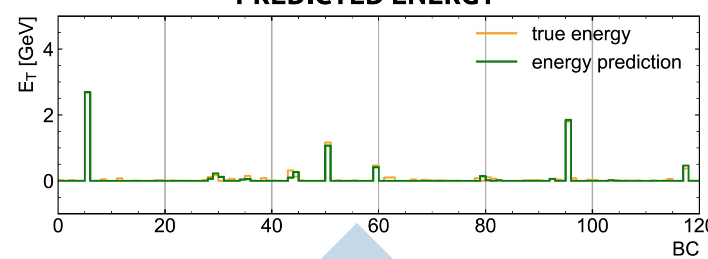
CNN: Energy inference



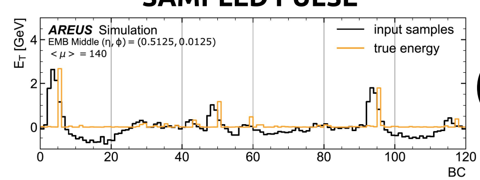
CNN for energy reconstruction:

Energy reconstruction layers are added to the tagging layers and retrained together

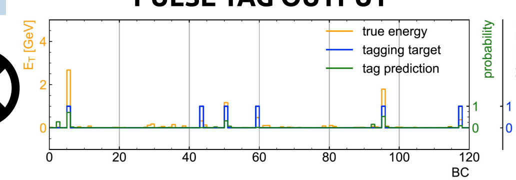
PREDICTED ENERGY



SAMPLED PULSE



PULSE TAG OUTPUT



4-Conv CNN

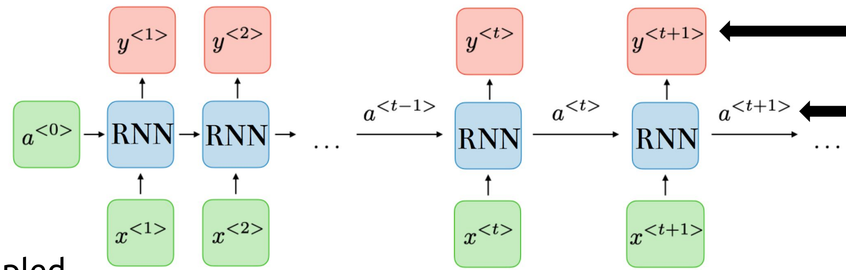
- **2** layers for energy reconstruction
- Receptive field: **13** BC
- **88** parameters in total

3-Conv CNN

- **1** layer for energy reconstruction
- Receptive field: **28** BC
- **94** parameters in total

Recurrent Neural Networks

Designed for handling sequential data, RNNs consist of internal neural networks that process new input combined with the past processed state



Extracts energy at all times (in-built pulse tagging)

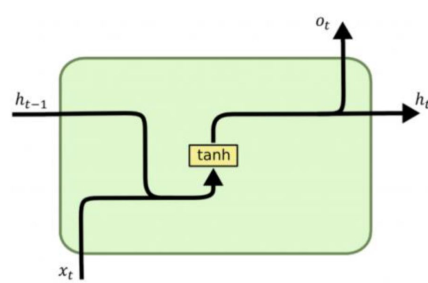
Decides what information to forward in time
⇒ Pulse span 20-30 BC, overlap …

Pulse sampled at every BC

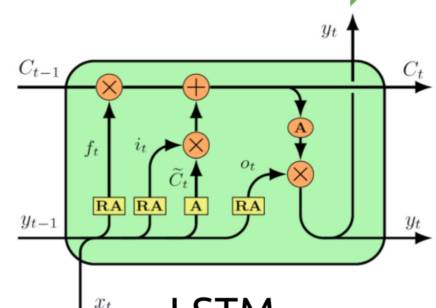
Better performance, more stability in time

Two RNN internal architectures explored:

- Optimised for smaller number of parameters
- Long Short-Term Memory (LSTM) - 10 internal dimensions
- Vanilla-RNN - 8 internal dimensions



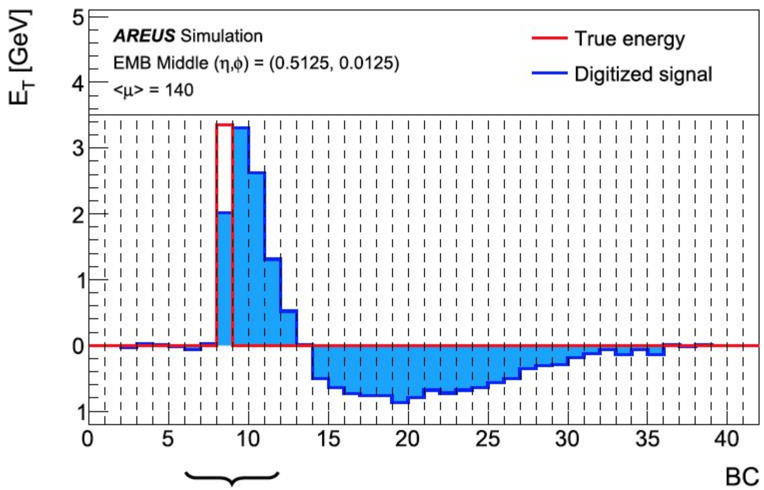
Vanilla-RNN
(89 parameters)

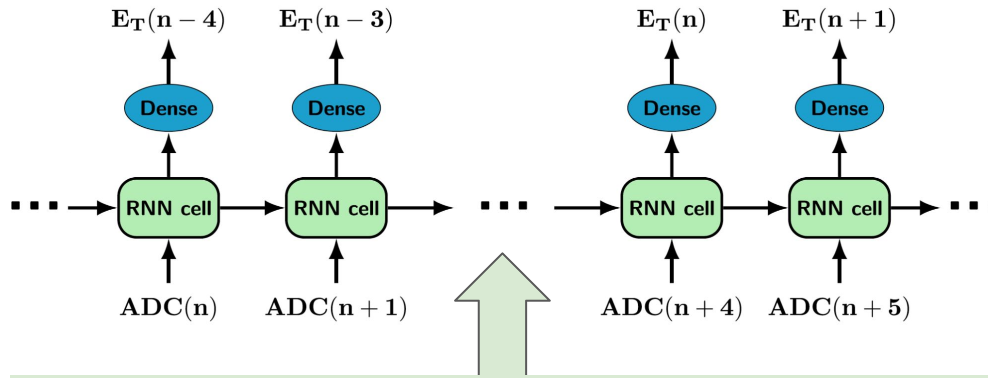


LSTM
(491 parameters)

Higher complexity, bigger size on hardware

RNN applications: two methods

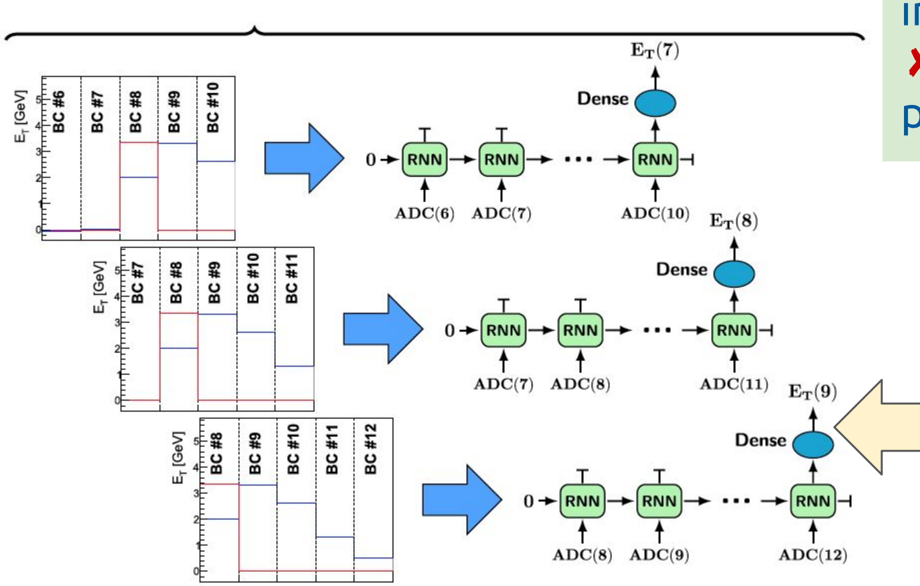




Single Cell Method:

✔ Long range correction, full signal is processed in a stream

✘ Significant amount of complexity needed to process data in time (LSTM only)



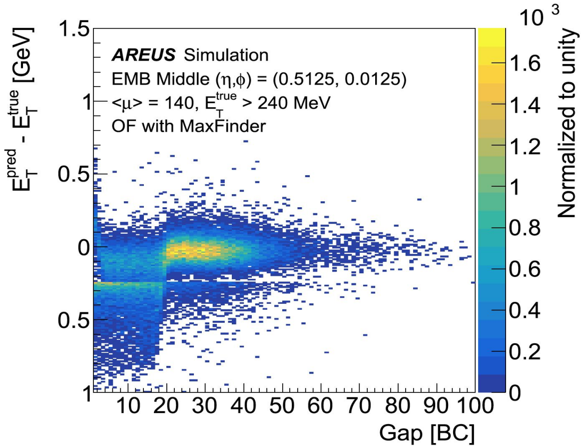
Sliding window Method (5 BC):

✔ Robust against long-lived effects due to unforeseen behaviour of the detector, simpler training

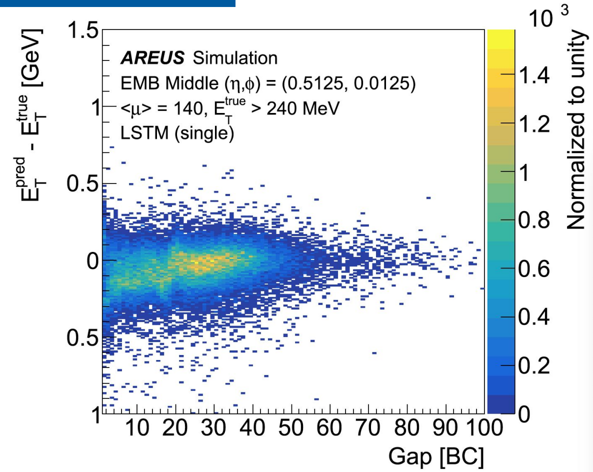
✘ Short range correction only (1 BC in the past)

Performance :
HL-LHC condition with pileup of 140

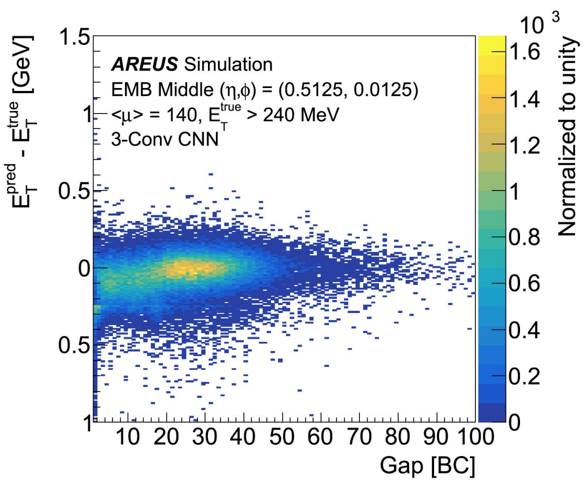
Comparisons on single LAr cell simulations (***AREUS*** software)



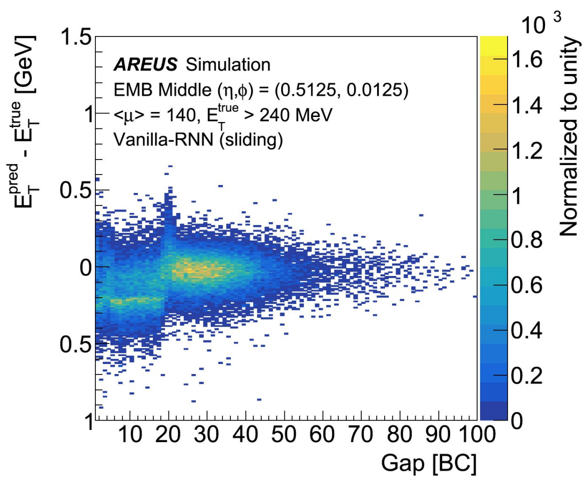
Legacy algorithm:
5 BC in the peak



LSTM (single cell):
5 BC in the peak, ∞ in the past



3-conv CNN:
5 BC in the peak, 8 in the past

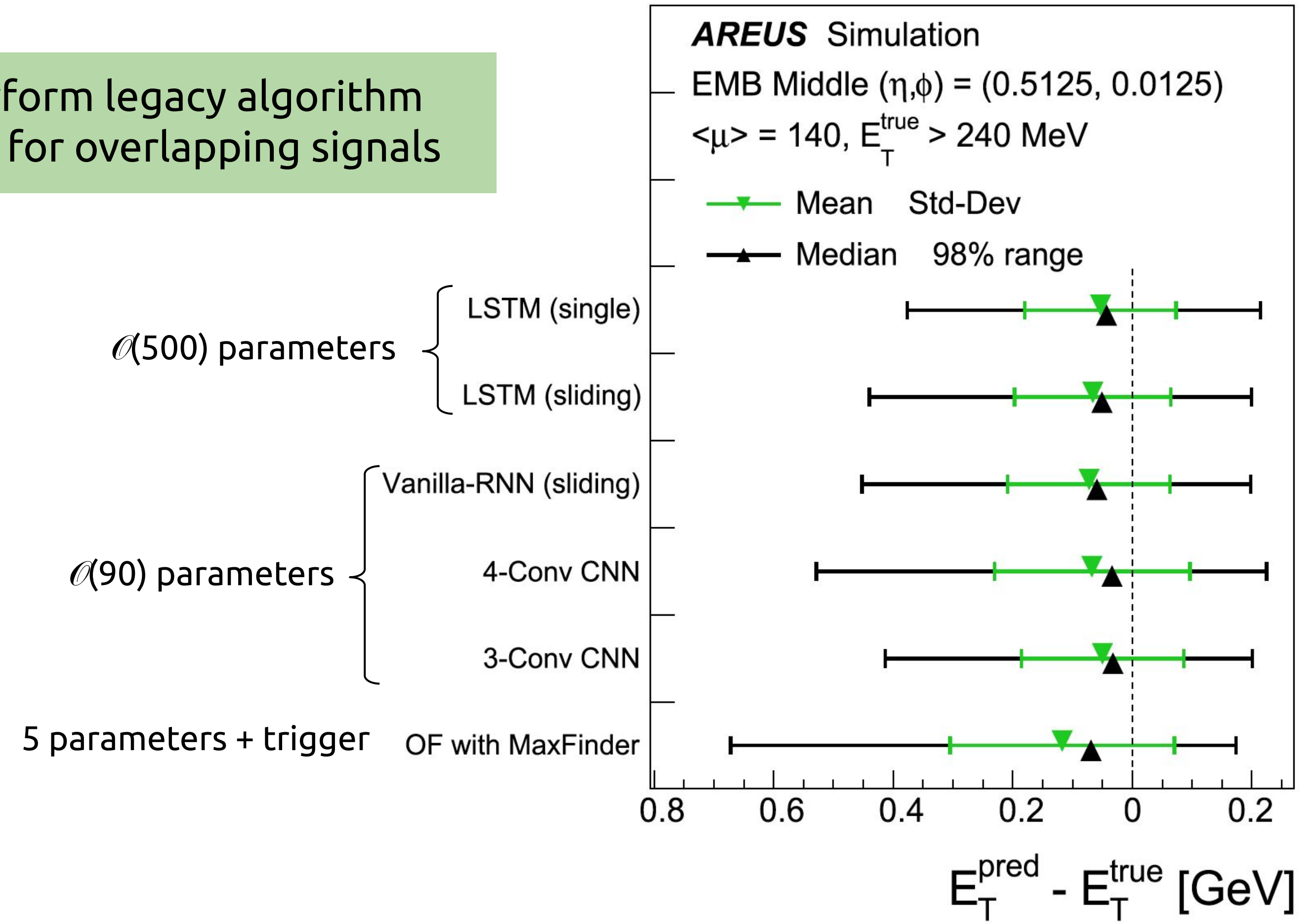


Vanilla (sliding window):
4 BC in the peak, 1 in the past

- Legacy algorithm exhibits big distribution tails especially at low gap
- The tails are reduced significantly with all of the new NN methods

Performance :
HL-LHC condition with pileup of 140

All methods outperform legacy algorithm
Clear improvement for overlapping signals



FPGA Implementations

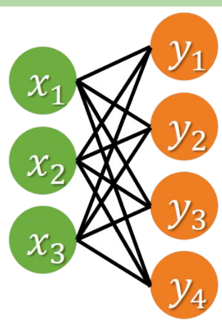
ANN model in Keras

- Set of weights optimised by training
- architecture(layers, dimensions, …)
- Mathematical operations

converter

FPGA firmware

- ALM: adaptive logic modules
- DSP: digital signal processors
- Fixed-point arithmetic, LUT for non-linear functions



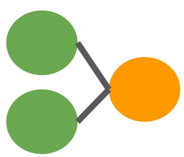
$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = A\left(\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \right)$$

Activation function for
non-linear element operations

FPGA Implementations: CNNs

The CNNs are transformed into VHDL code with the help of a custom-made VHDL converter:

- Configured directly by Keras model
- Optimised for low latency:
  - CNN architecture mapped to DSP chains
  - Pipelined inputs

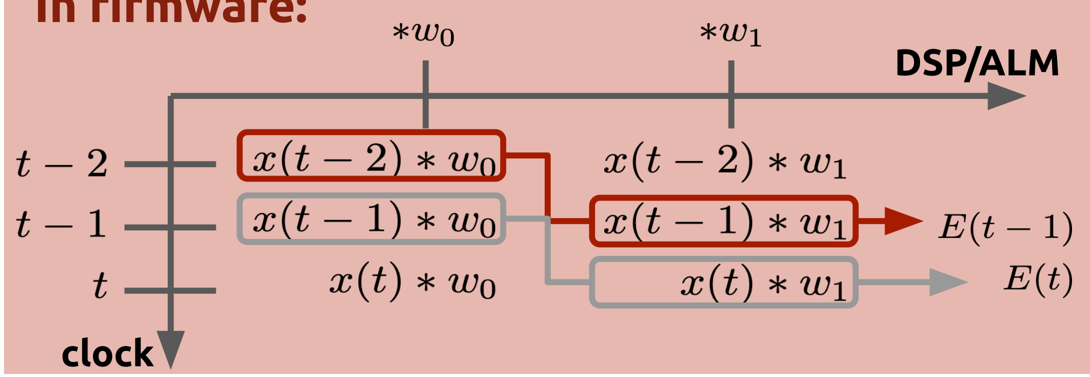


In software:

$$E(t-1) = x(t-1) * w_1 + x(t-2) * w_0$$
$$E(t) = x(t) * w_1 + x(t-1) * w_0$$

Input pipeline : reuse hardware as soon as available to deal with continuous data flow

In firmware:

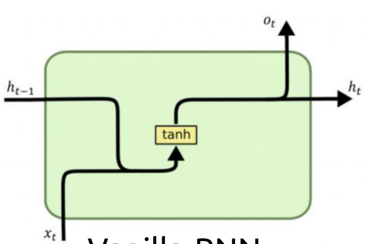


FPGA Implementations: RNNs

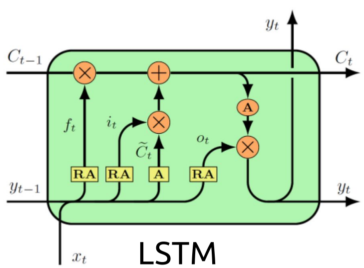
RNNs implemented in Intel HLS:

- automated generation of hardware description language from a C++-like algorithmic description of the network
- flexible design automatically optimised to a given hardware target

RNN cells coded as template functions

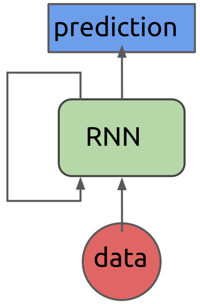


Vanilla-RNN



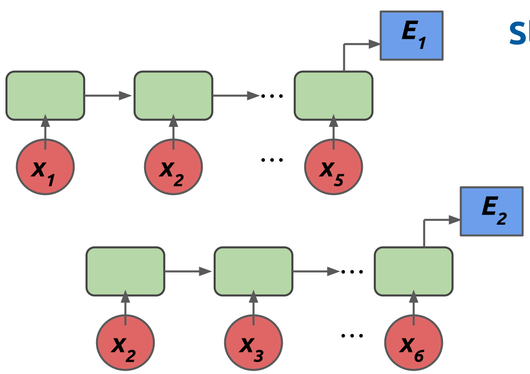
LSTM

Every loop is "fully unrolled"
⇒ each of the loop iterations has its own logic resources



Single-cell HLS:

- Single RNN instance on hardware
- Waits for output of previous cell arriving at frequency/latency



Sliding window HLS:

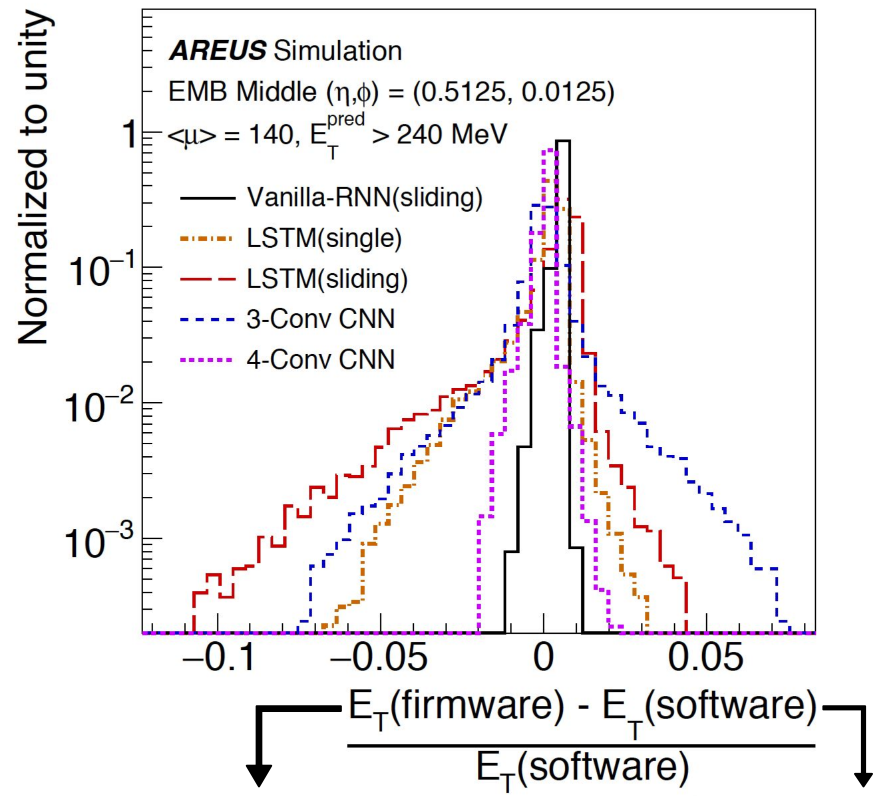
- 5 RNN instances
- Independent sequences ⇒ pipelined

FPGA Implementations: Results

Compare Intel Stratix 10 simulation (Quartus 20.4 and Questa Sim 10.7c) to Keras Tensorflow :

- Pulse samples from AREUS LArcell data

Good compatibility firmware/software
(RMS 0.6% to 2.2%)



FPGA Simulation
(FP + LUT)

Keras/Tensorflow
(float operations)

Optimized fixed point and LUT representations:

- minimize resources VS compatibility software/firmware
- 18 bits total (Stratix 10 ⇒ 18x19 DSP) :
    - 10 decimal for CNNs
    - 13 decimal for RNNs

⇒ Acceptable quantisation noise when using 18 bits (lower than the expected input noise).

FPGA Implementations: Resource usage

Single LAr cell resource usage estimated from Intel Stratix 10 simulation (Quartus 21.1 and Questa Sim 10.7c)

Network	Frequency	Latency	Resource usage	
	F_{max} [MHz]	clock(core) cycles	#ALMs	#DSPs
VanillaRNN (sliding)	640	120	5782 (0.6%)	152(2.6%)
3-Conv CNN	344	81	14235(1.5%)	46(0.8%)
4-Conv CNN	334	62	15627(1.7%)	42(0.7%)

- Many readout channels treated by one FPGA ⇒ time-domain multiplexing
- Maximum achievable frequency : 480-600 MHz ⇒ upto 15x multiplexing of 40 MHz input data
- Assuming all available FPGA resources being dedicated to ANN algorithms, 3-Conv CNN and VanillaRNN can reach a value above 384 channels ⇒ can receive data from three FEBs
- Further VHDL and HLS optimisations ongoing to reach even smaller resource usage, shorter latency, and higher clocking frequency

Conclusion

- HL-LHC will require improving ATLAS LAr energy measurements
  - Two novel methods - CNN and RNN based

- For both CNN/RNN several algorithms are developed:
  - Focused on recovering energy resolution in high pileup environments by using information from past events
    - All methods outperform legacy algorithms in HL-LHC conditions

- FPGA implementation for fast processing:
  - CNN : dedicated VHDL
  - RNN : flexible HLS
    - Good reproduction of Keras results with firmware simulation
    - Optimizations ongoing to reduce resource usage and latency to stay within ATLAS limitations

- CNN/RNN implementation in LAr readout for phase II is challenging, but the preliminary results indicate that it has great potential to improve the energy reconstruction

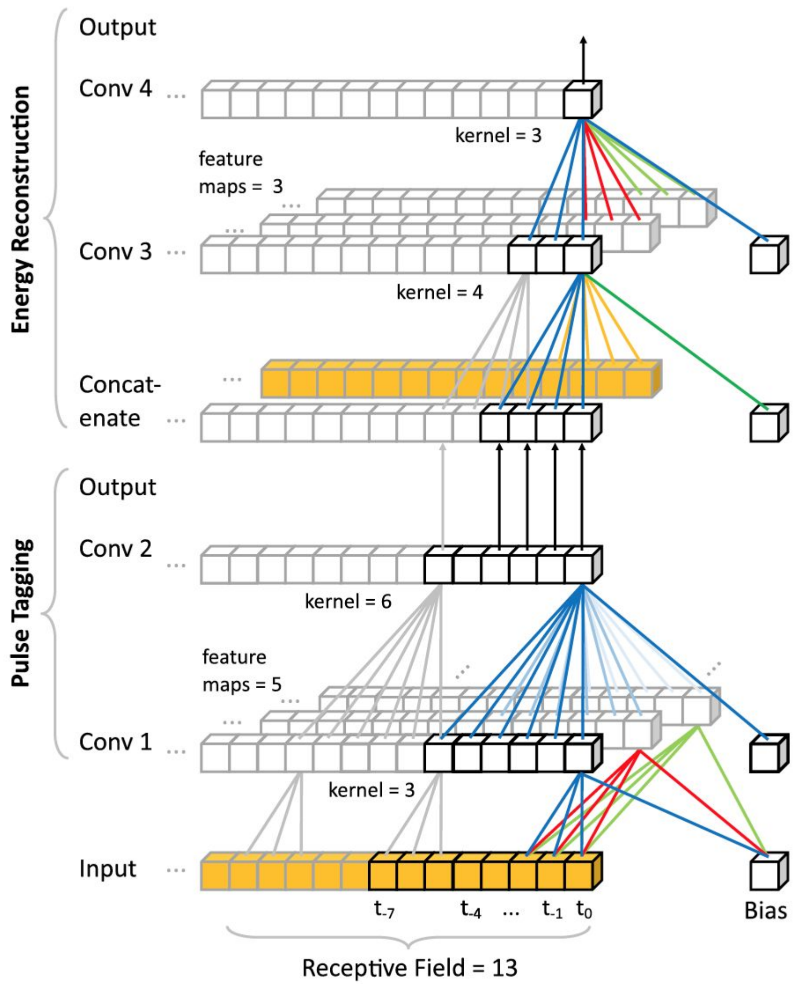
Ref. "Artificial Neural Networks on FPGAs for Real-Time Energy Reconstruction of the ATLAS LAr Calorimeters"
Aad, G., Berthold, AS., Calvet, T. et al., ***Comput Softw Big Sci 5, 19 (2021).***

Backup

Energy inference with Convoluted Neural Networks

1-Dimensional CNN designed with a succession of filters to perform two tasks :

- pulse tagging
- energy reconstruction



CNN example : shape classification