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

Lauri Laatu on behalf of the ATLAS Liquid Argon Calorimeter group

CPPM, Aix-Marseille Université, CNRS/IN2P3 (FR)

Introduction

- The High-luminosity phase of LHC (HL-LHC) is expected to produce 5-7 times the nominal instantaneous luminosities
- Up to 140-200 simultaneous proton-proton interactions every 25 ns
- Requires major upgrades to the ATLAS detector to meet the physics goals

LHC			HL-LHC
Run 1	Run 2	Run 3	**Run 4**
2011-2013	2015-2018	2022-2024	**2027-2040**
30 fb^{-1}	190 fb^{-1}	350 fb^{-1}	3000 fb^{-1}

Energy Reconstruction in LAr Calorimeter

- The Liquid-Argon (LAr) Calorimeters of ATLAS measure the energy of electromagnetic showers of photons/electrons using their ionisation signals
- Bipolar pulse shape (total length of up to 600 ns, 25 BCs)
  - Sampled and digitized at 40MHz (figure 1)
- Energy reconstruction in real-time using FPGAs
  - Latency of about a few hundred nano seconds required by the trigger data path
- Current energy reconstruction uses the optimal filtering algorithm with a maximum finder (OFMax)
  - Using five samples around the pulse shape peak
  - Decreased performance at the HL-LHC
- Full electronics readout chain will be upgraded for Run 4
  - Increased computing capacity with Stratix-10 FPGAs
  - In the current design options, each FPGA processes 384 or 512 LAr calorimeter cells

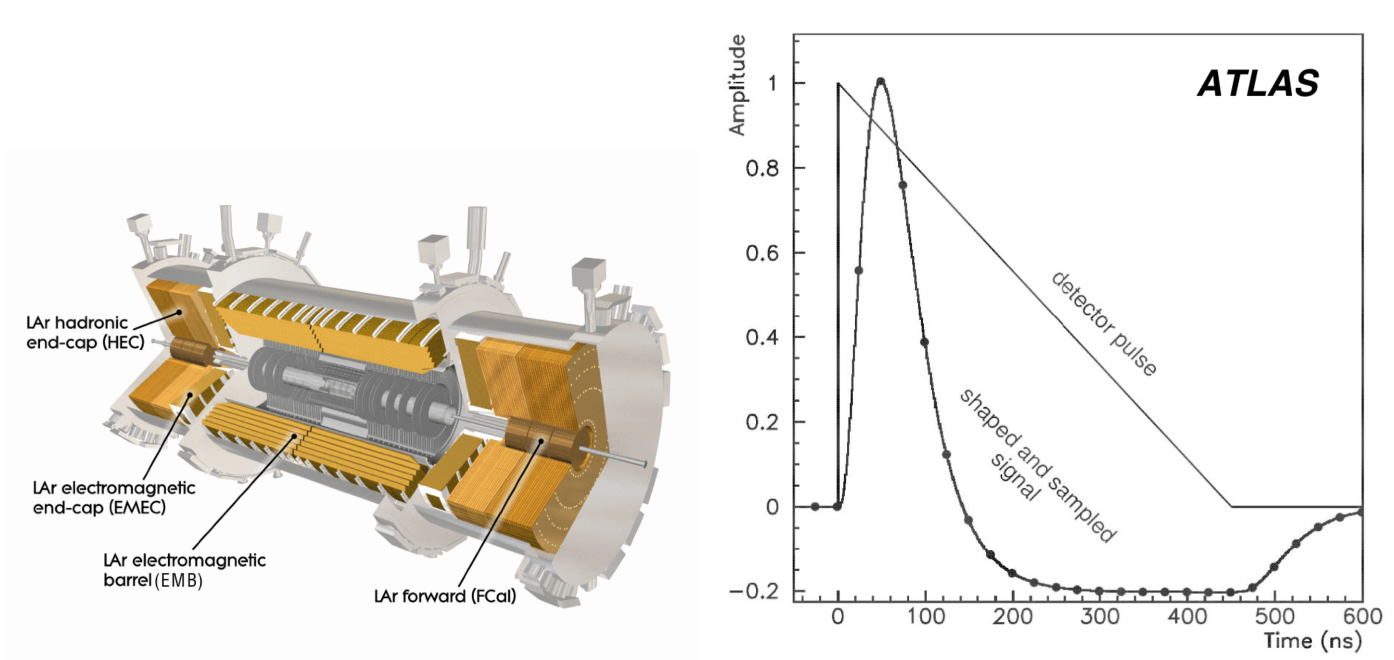

Figure 1:Left - cutout of the LAr calorimeter, right - shaped and digitized LAr calorimeter pulse

Convolutional Networks

- **1-D convolutional network (1-D CNN)** for sequence processing consisting of a two-staged architecture: tagging and energy reconstruction

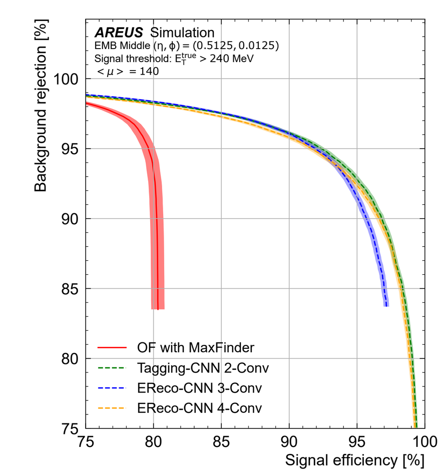

Figure 2:Signal efficiency and background rejection ROC curves of the CNNs and their tagging part, compared to the OFMax

- Pulse tagging sub-network (2 layers)
  - First trained to detect energy deposits above noise threshold (signal)
  - Sigmoid activation function
- Energy reconstruction sub-network (1-2 layers)
  - Uses the results of the tagging sub-network and raw ADC samples
  - One or two reconstruction layers resulting in 3-Conv and 4-Conv networks
  - ReLU activation function

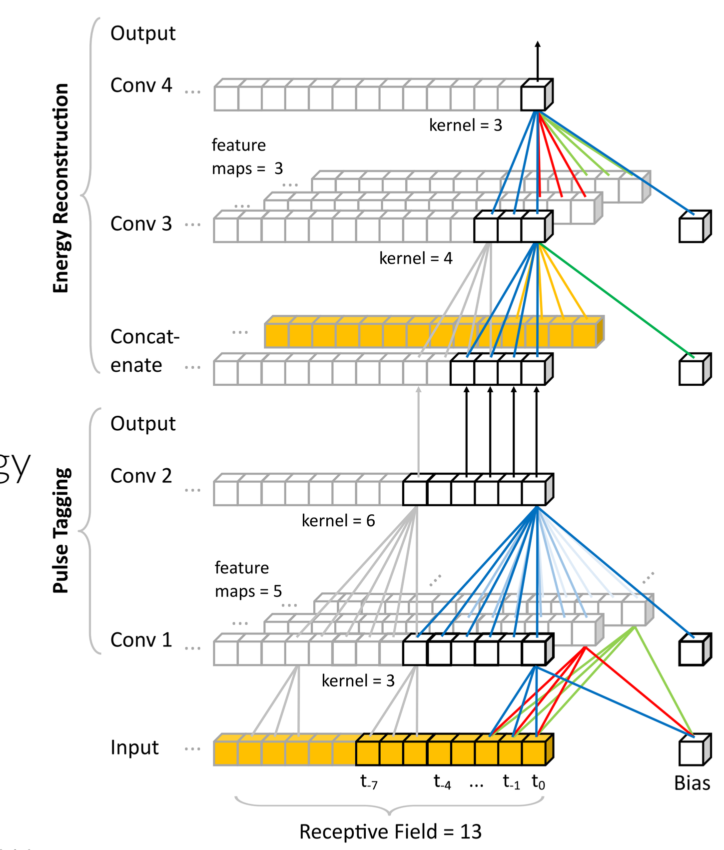

Figure 3:Architecture of a CNN with four convolutional layers. The input sequence is first processed by the tagging layers and then by the energy reconstruction layers.

Recurrent Neural Networks

- **Recurrent neural networks (RNNs)** are a family of neural networks for processing sequential data
  - Vanilla RNN with ReLU activation is the simplest recurrent structure
  - Long short-term memory (LSTM) with a gated cell design and with sigmoid and tanh activations can handle long term effects better
- Two ways to feed data to RNNs as shown in figure 4
  - **Sliding window** with a window size of 5 including one sample before the pulse
  - Continuous stream of digitized samples for **single cell** LSTM with unlimited information of past events

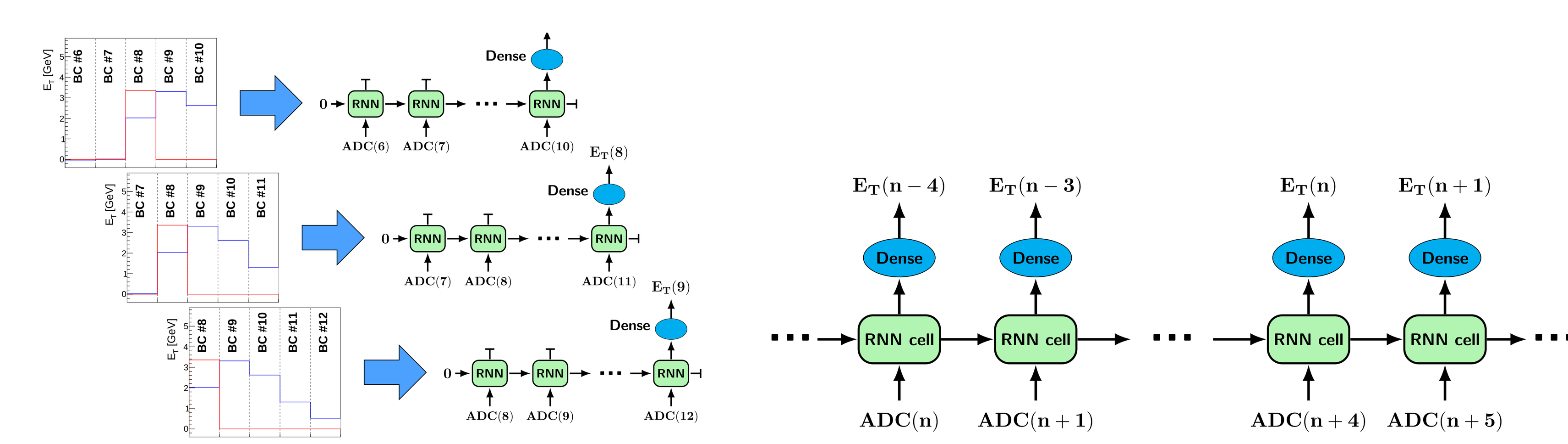

Figure 4:Left - RNN processing of calorimeter samples for RNNs with sliding window, right - stream for single cell LSTM

Network Performance

- The NNs outperform the OF algorithm as shown in figure 6
- The NNs better reconstruct pulses distorted by previous events (figure 5)
  - Proportionally to the usage of past information

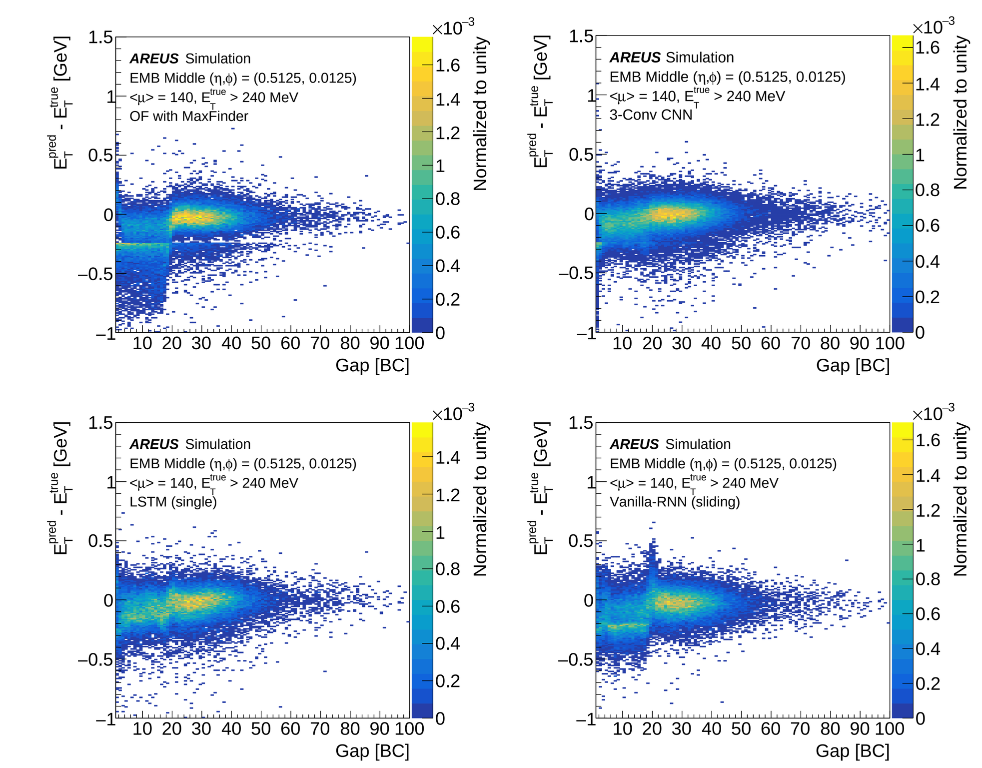

Figure 5:Resolution as a function of the distance to previous high energy deposit (gap)

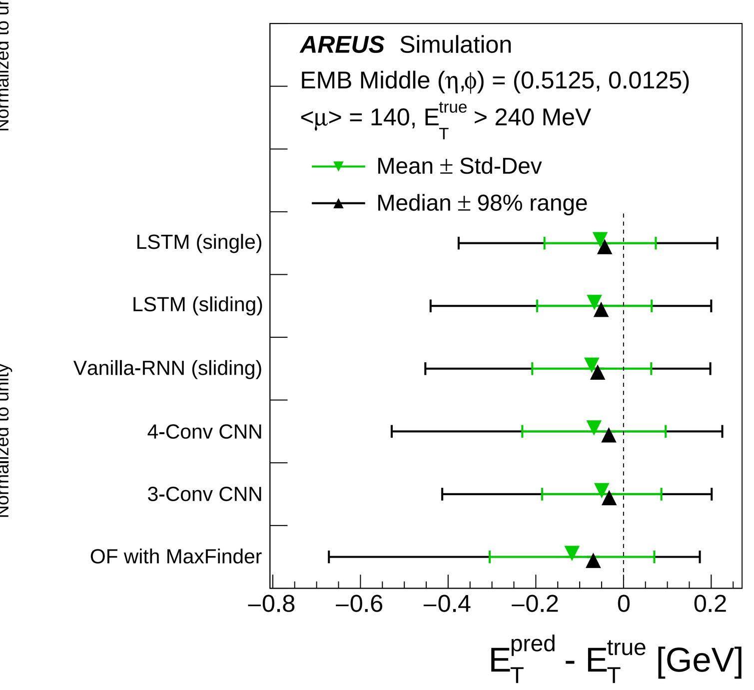

Figure 6:Energy resolution for different algorithms

FPGA Performance

- Good agreement between firmware and software solutions is observed as shown on figure 7
- Table 1 shows the resource usage for different methods
  - Multiplexing is used to process multiple calorimeter cells with one network instance
- Reasonable resource usage for implementation on the real hardware for Phase-II

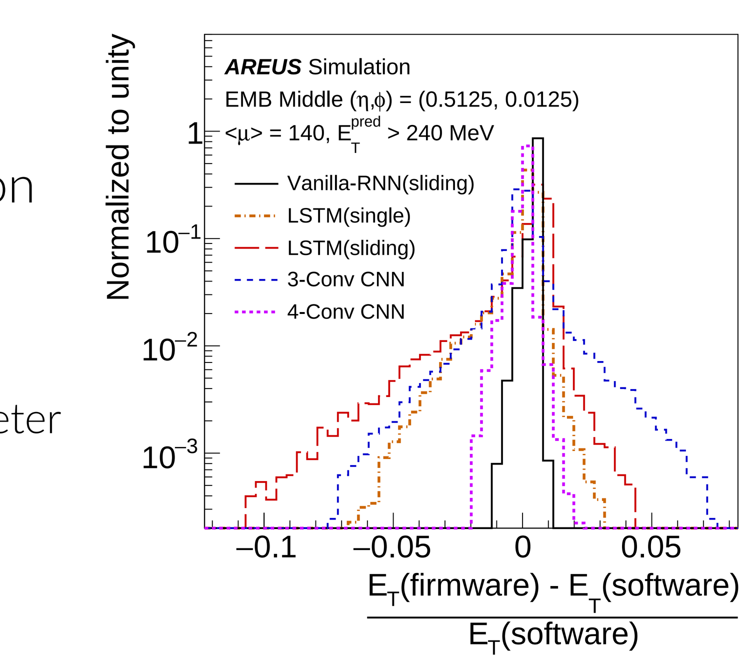

Figure 7:Relative deviation of the firmware and software results

	Multiplexing	Freq F_{max} [MHz]	Latency clk_{core} cycles	LAr Channels	Resource Usage DSP/ALM
3-Conv	6	344	81	390	0.8% / 1.5%
Vanilla	15	640	120	576	2.6% / 0.6%

Table 1:Occupancy of the NN implementations on a Stratix-10 FPGA