Convolutional Neural Networks for real-time processing of ATLAS Liquid-Argon Calorimeter signals with FPGAs

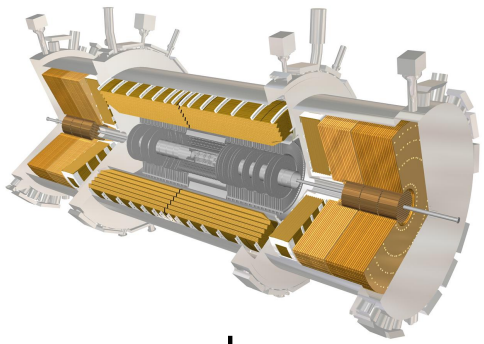
Nick Fritzsche,
Anne-Sophie Berthold, Rainer Hentges, Arno Straessner, Johann Christoph Voigt

Institute of nuclear and particle physics,
TU Dresden

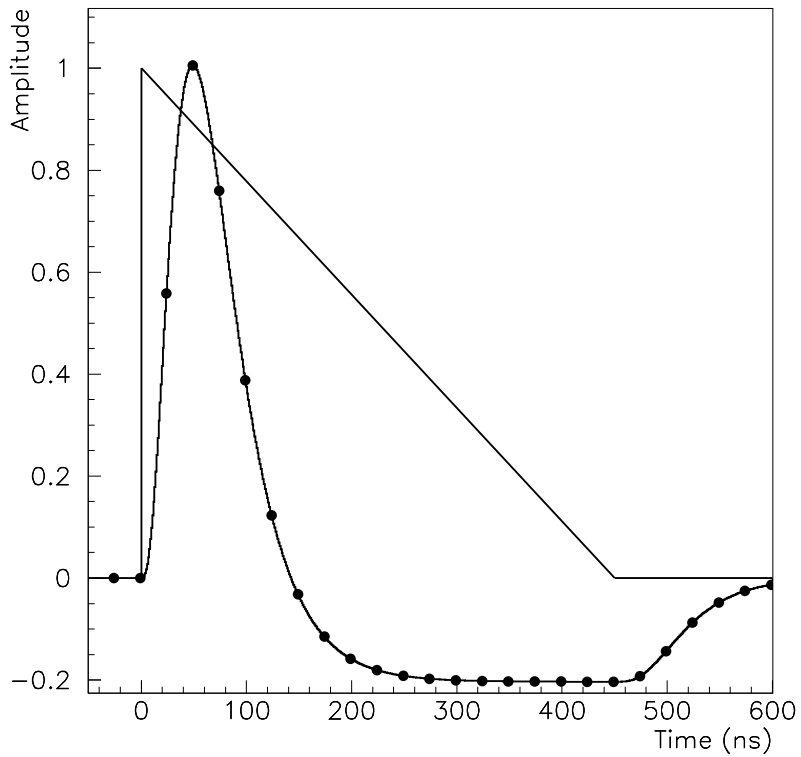
November 30, 2020

Signal Readout of the **ATLAS LAr Calorimeter**

LAr calorimeter system is part of ATLAS particle detector at Large Hadron Collider (LHC) at CERN



readout per cell →



- energy deposits in the LAr calorimeter raise a triangular pulse
- shaped by CR(RC)^2 analog filter into bipolar pulse and digitized
- amplitude is proportional to deposited energy

Energy Reconstruction by **Convolutional Neural Networks**

Our Approach

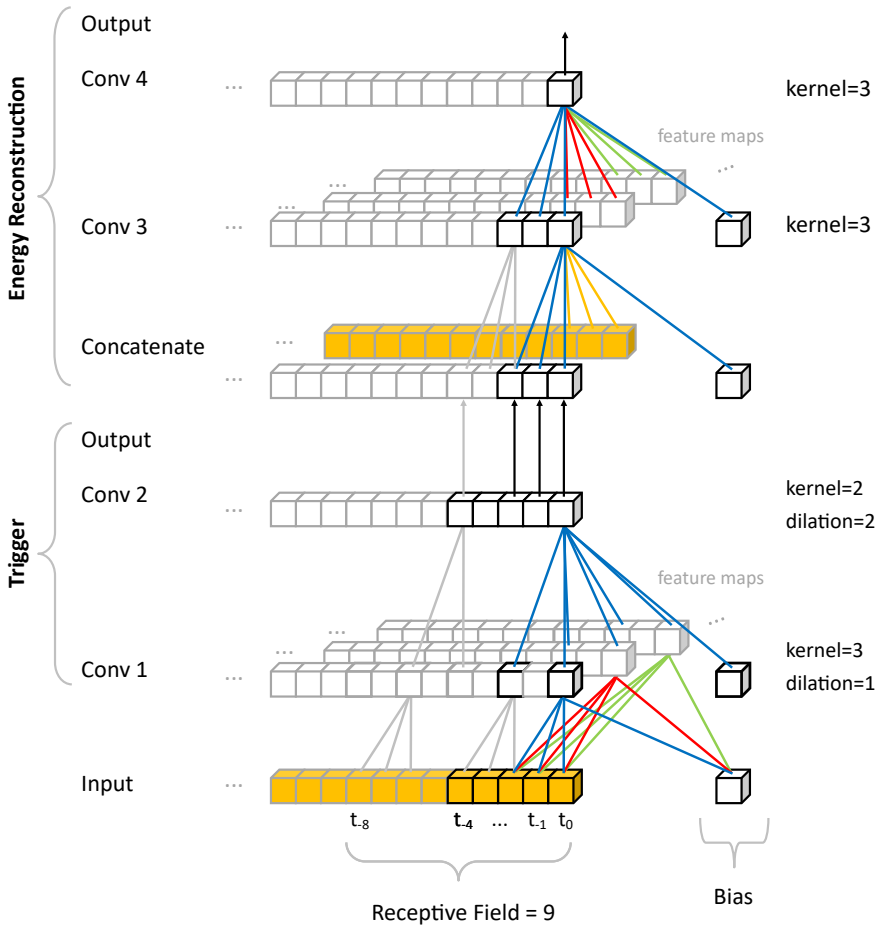
Combine trigger CNN with energy reconstruction CNN

Trigger CNN

- output: probability of detection
- sigmoid activation function
- dilation enlarges receptive field without parameter increase

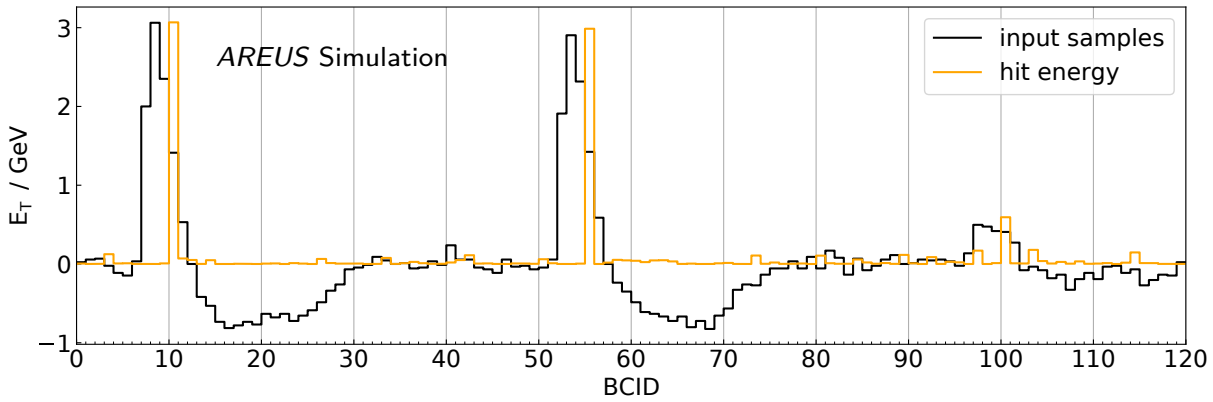
Energy reconstruction CNN

- output: energy
- gets trigger probability and ADC sequence for reconstruction

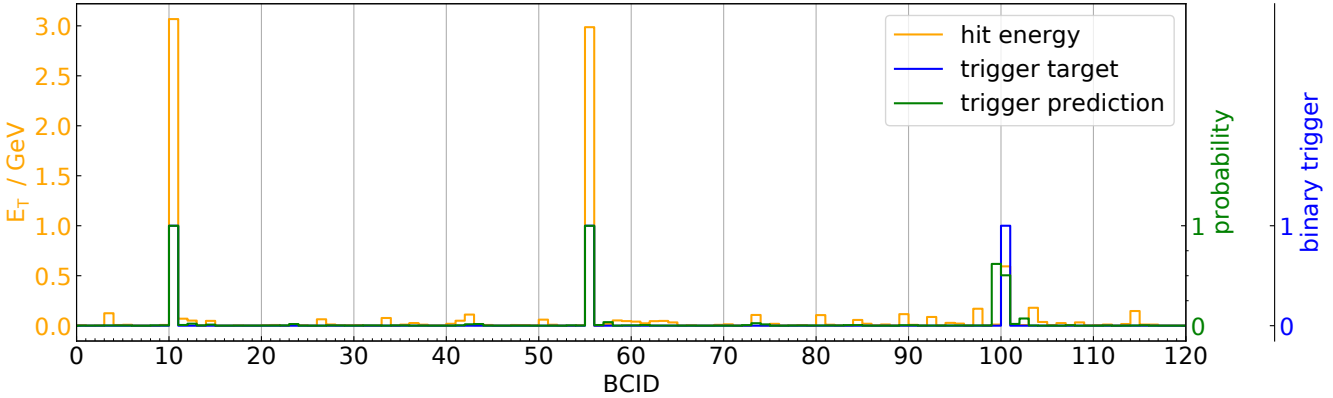


Training of Convolutional Neural Networks

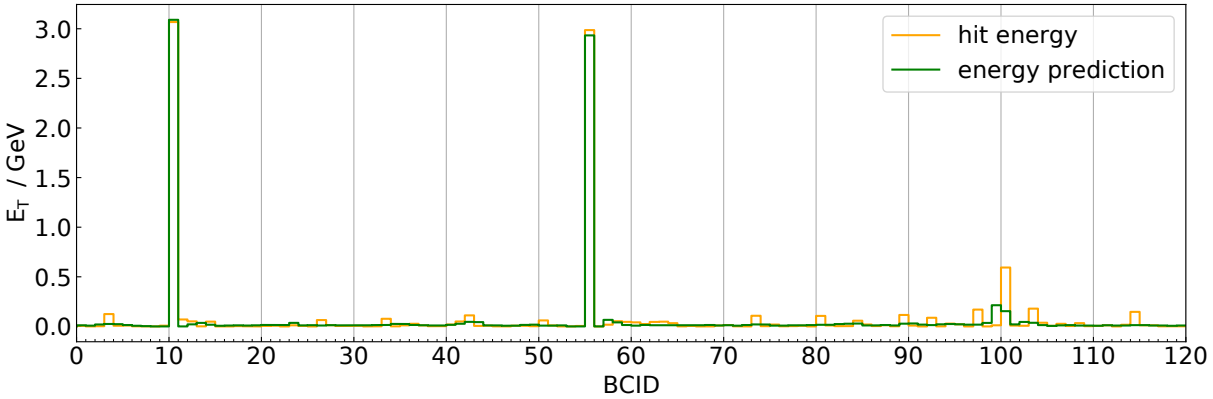
Input: digitized sequence
Overall target: hit energy



Trigger part:
output is detection probability

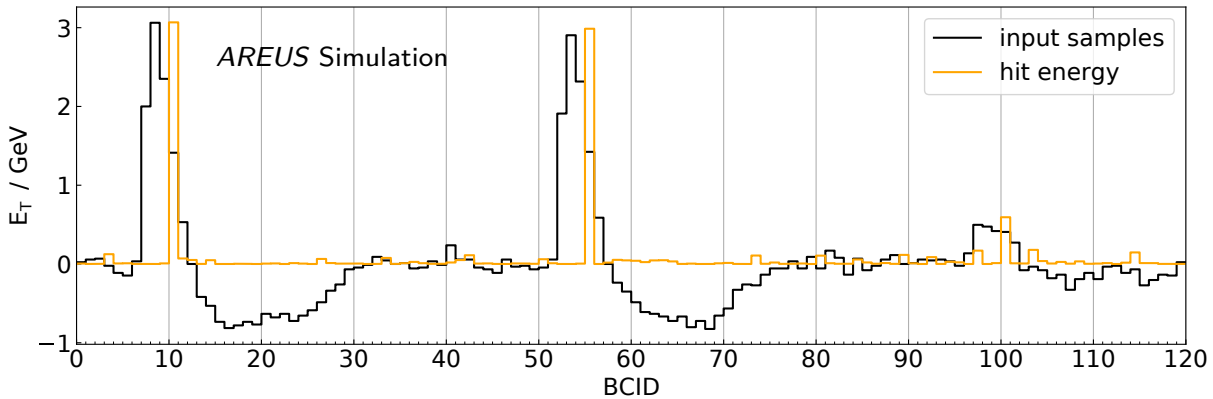


Energy reconstruction part

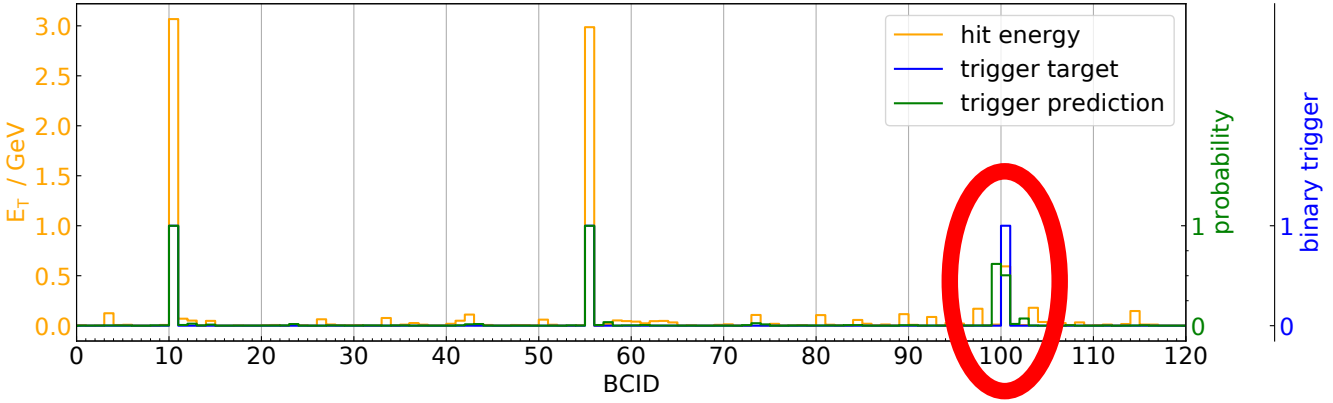


Training of Convolutional Neural Networks

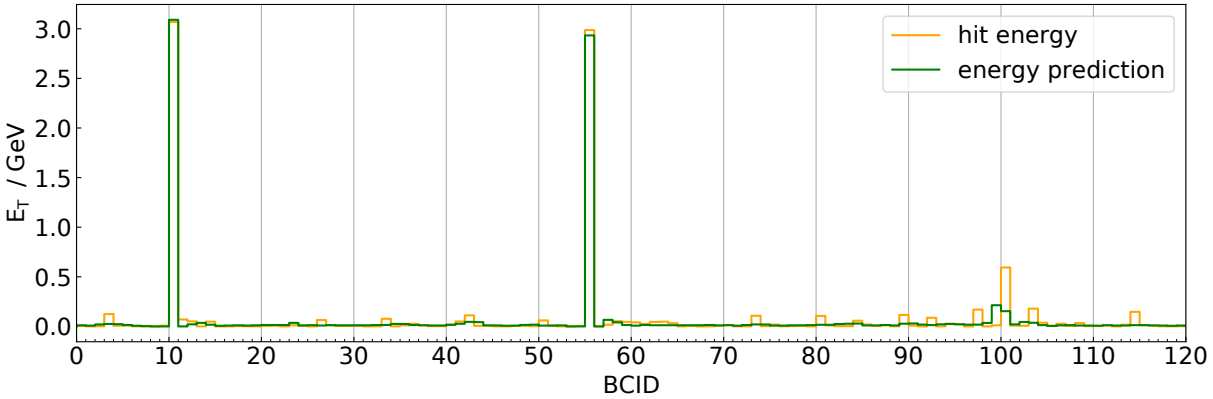
Input: digitized sequence
Overall target: hit energy



Trigger part:
output is detection probability

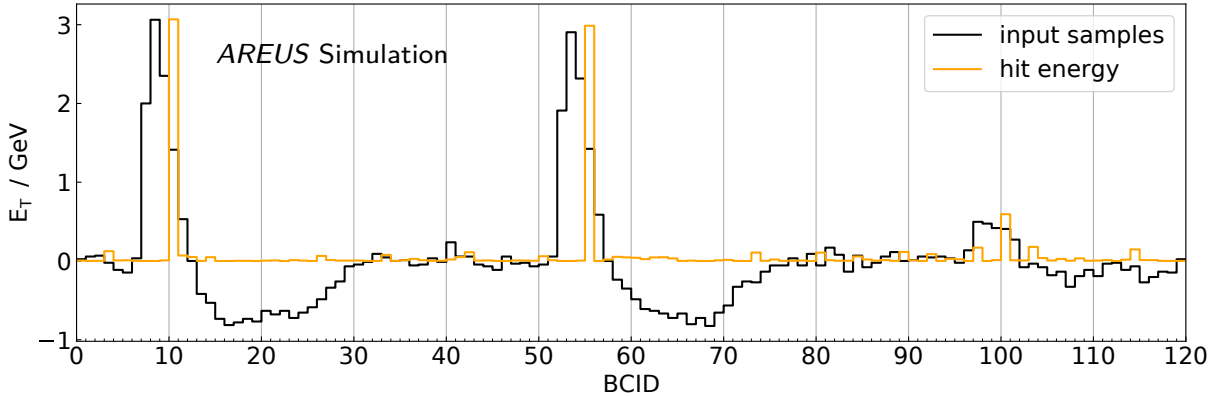


Energy reconstruction part

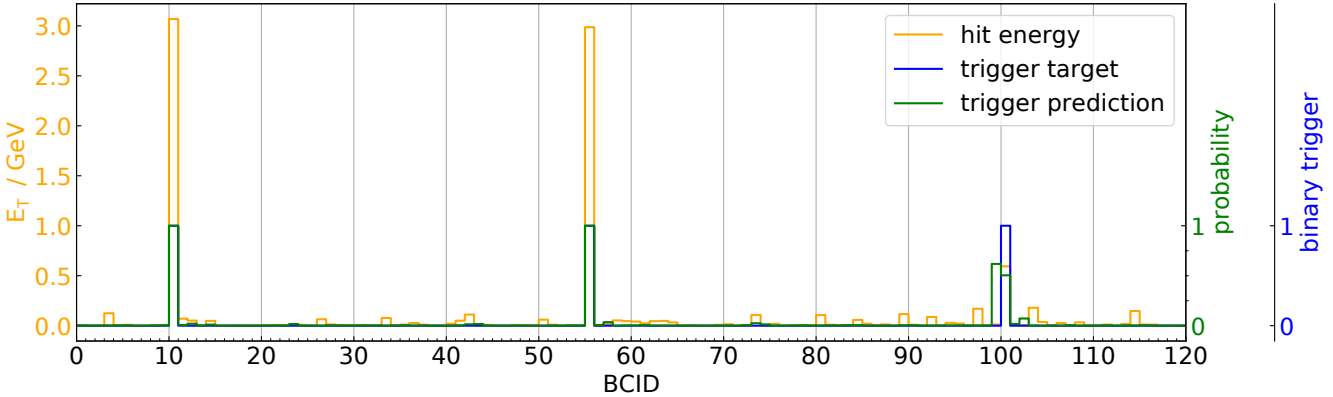


Training of Convolutional Neural Networks

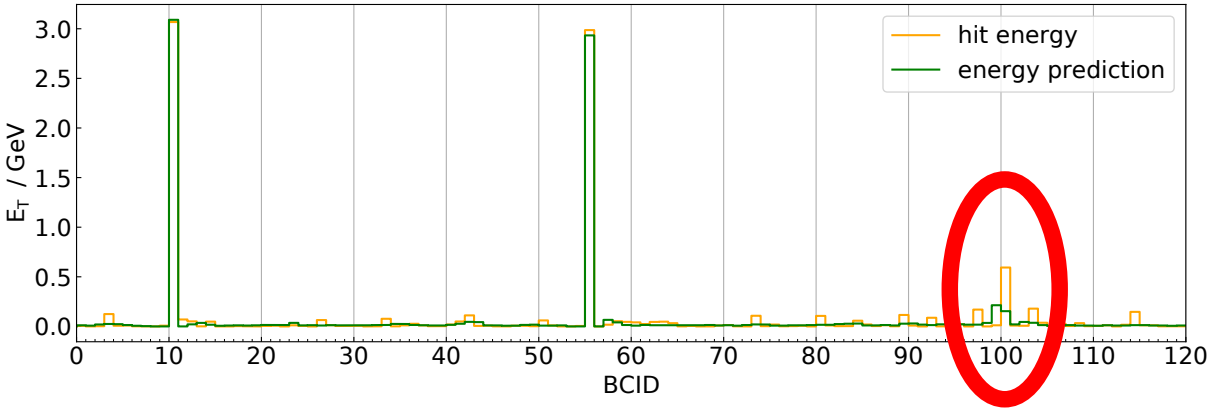
Input: digitized sequence
Overall target: hit energy



Trigger part:
output is detection probability

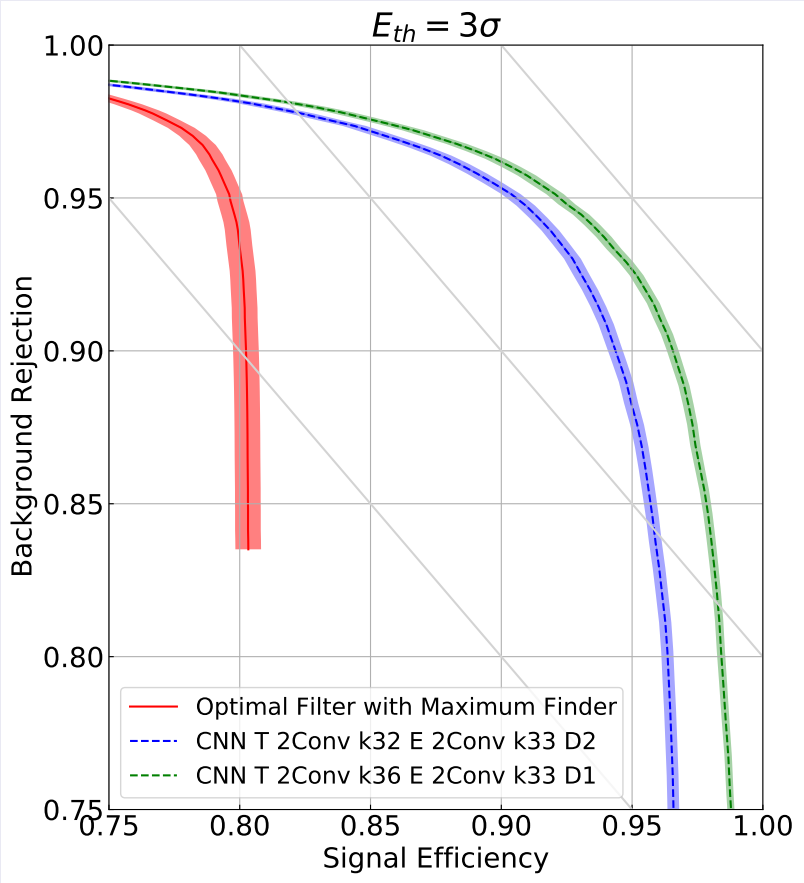


Energy reconstruction part



Performance Evaluation: **Trigger Efficiency** and **Energy Resolution**

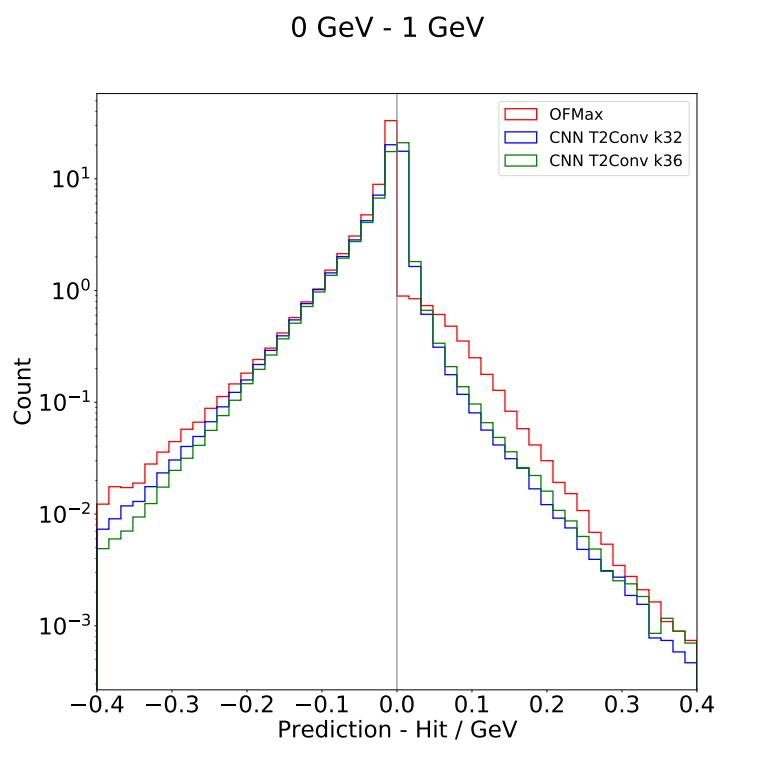
Trigger Efficiency: ROC-Curve



Blue: Trigger Net: 2 conv. layers, kernel sizes 3 and 2
E-Reco Net: 2 conv. layers, kernel sizes 3 and dilation rate of 2

Energy Resolution

Signal Energy Region



Green: Trigger Net: 2 conv. layers, kernel sizes 3 and 6
E-Reco Net: 2 conv. layers, kernel sizes 3 and dilation rate of 1

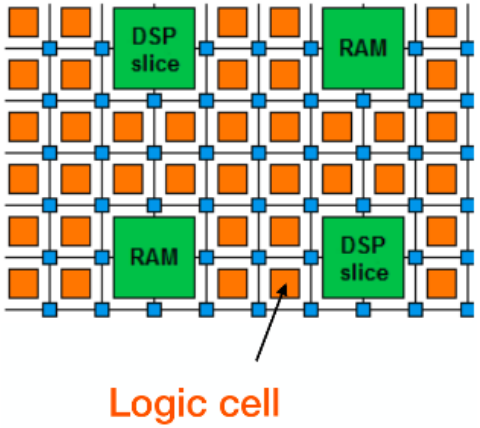
$\longrightarrow$ CNNs outperform Optimal Filter

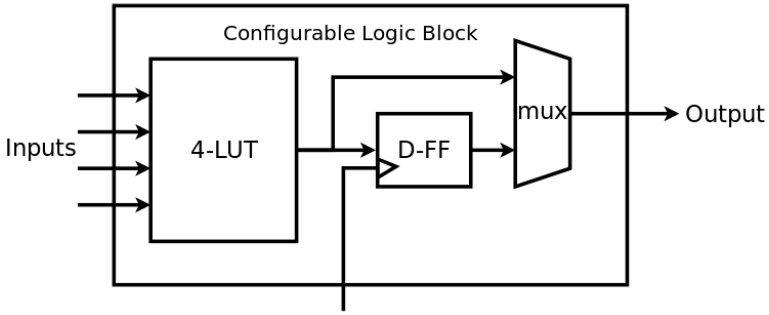
CNNs on **Field Programmable Gate Arrays**

Field Programmable Gate Arrays (FPGAs)

- integrated circuit configurable by the designer after manufacturing
- parallel data processing with high bandwidth ($> 1\,\mathrm{TB/s}$) in real time
- hardware reconfigurable for different CNN structures

Pictures taken from Intel and hls4ml





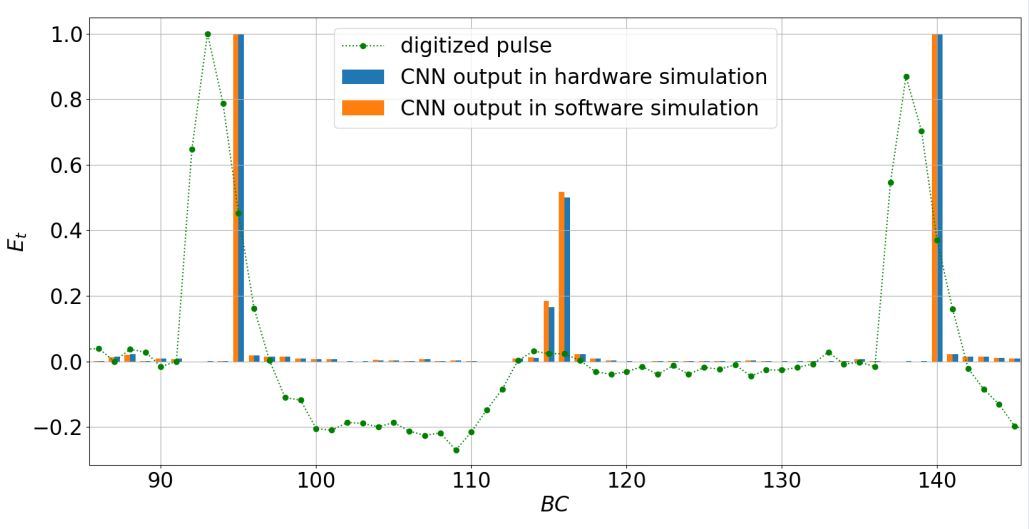
FPGA Resources

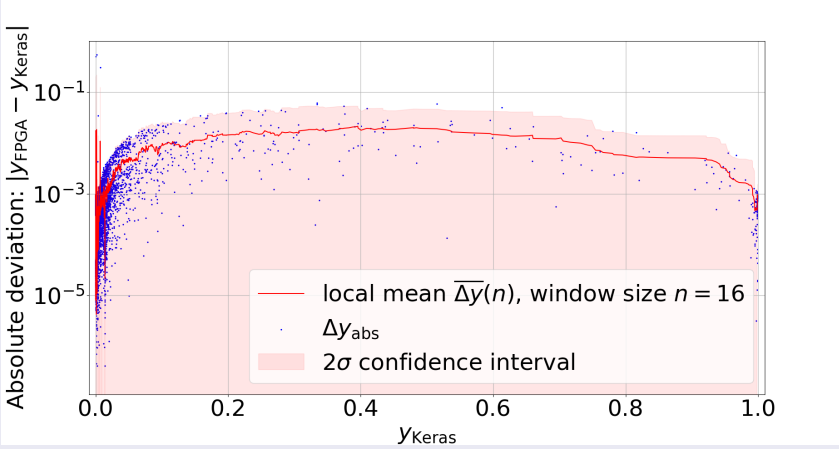
- **Digital Signal Processors (DSPs):** highspeed multiplications of layer inputs and weights
- **Lookup Tables (LUTs):** complex logic functions like sigmoid activation function
- **Dual port RAM:** connect to slow control system for loading weights

General **VHDL Implementation** for Different CNN Architectures

Component configurable with file produced after training

Accuracy: Software Simulation vs Hardware Implementation





Performance of different CNNs

Network				Frequency	Latency	Resource Usage	
Conv. Layer	Kernel Size	Dilation Rate	Feature Maps	F_{max}/ MHz	clk_{core} cycles	#DSPs	#ALMs
4	2,2,2,2	1	3,3,3	511.25	46	24	1828
2	8,8	3	4	527.98	27	32	1615
2	3,6	1	5	508.39	32	25	1377

Summary

- various CNN architectures developed and optimized
- show good performance in terms of trigger efficiency and energy resolution

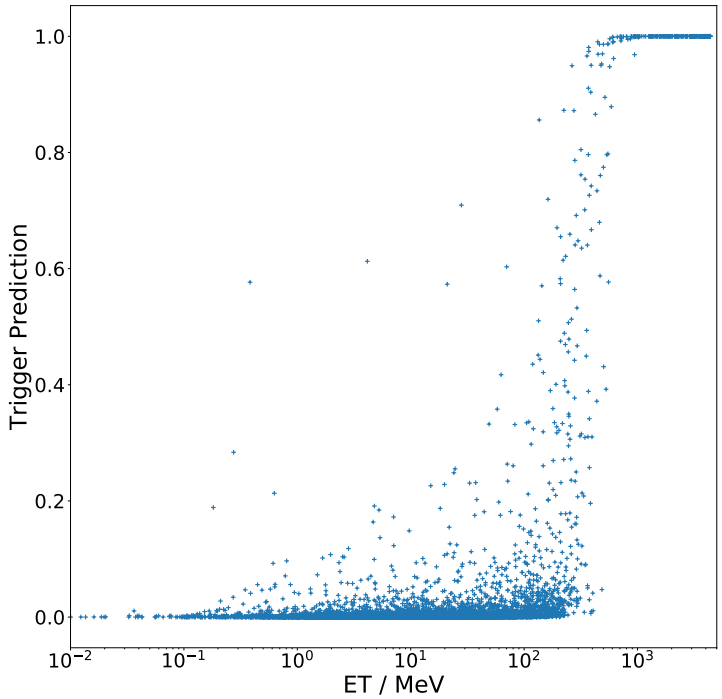
- configurable and automated firmware implementation for CNNs developed
- optimized regarding possible frequency, latency and resource consumption

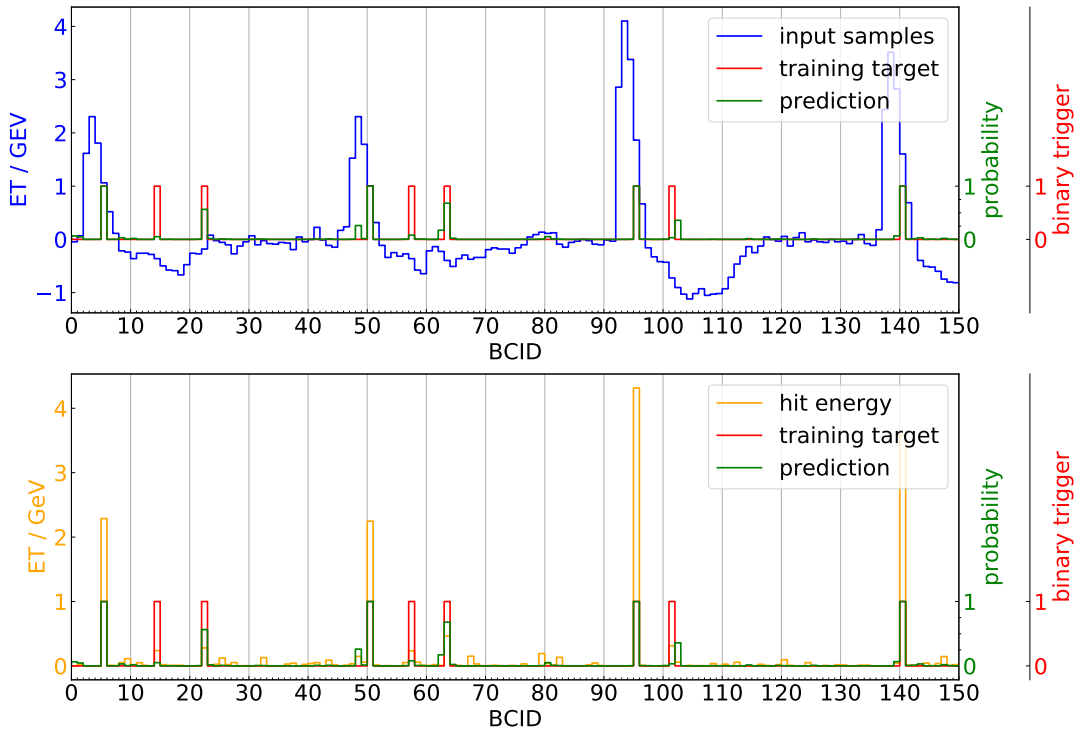
This work has been performed together with the LAr group of the ATLAS experiment.

Backup

Model: Neural Net Trigger

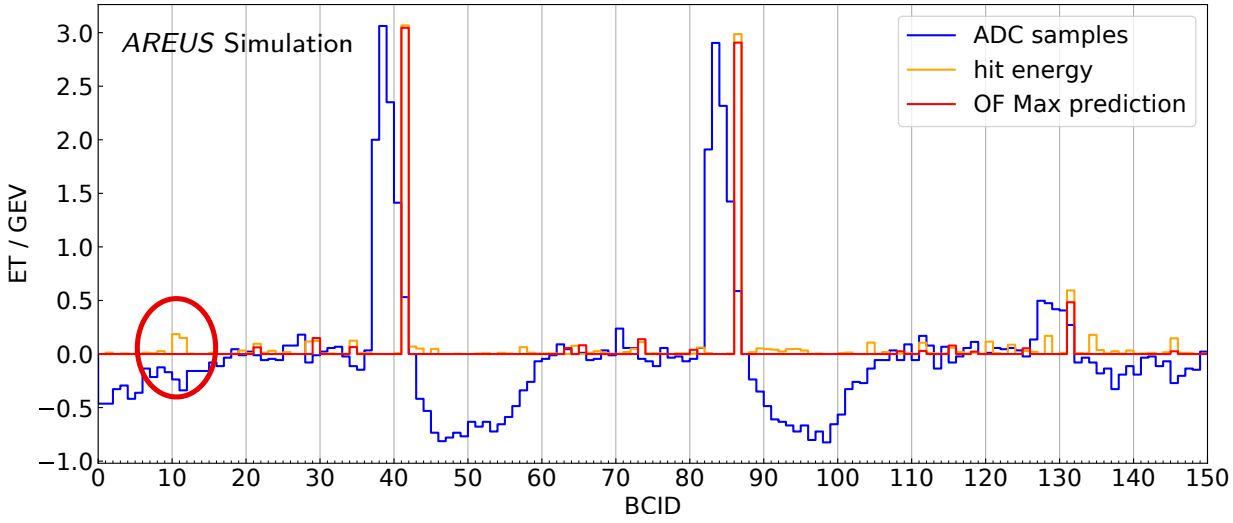
- architecture: Conv1Ds, causal, dilation, act.fun.=sigmoid, loss=binary crossentropy
- input: *AREUS* simulated digitized samples
- training on trigger T:
  - take *AREUS* hit sequence (truth E), shift for causality and convert into binary sequence:
  - $T = 1 \iff E_{hit} > 3\sigma$
  - $T = 0 \iff E_{hit} \leq 3\sigma$
  - σ: RMS of electronic noise ($\approx$ 80 MeV in EMB Middle layer, $\eta = 0.5$)





Energy Reconstruction by the **Optimal Filter**

- Optimal Filtering algorithm used to calculate deposited energy
- is optimized to suppress noise and reconstruct peak timing
- is insensitive during undershoot of a pulse



Phase-2 Upgrade

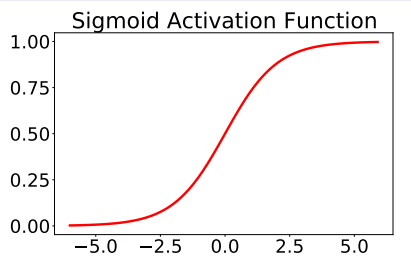
- with larger luminosity, pile-up will grow and affect subsequent signals increasingly
- trigger may occur in every bunch crossing

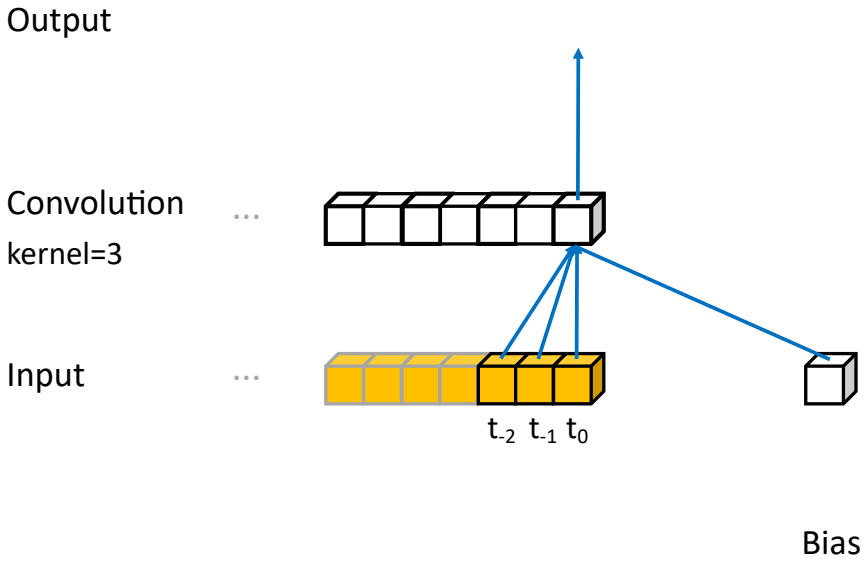
$\longrightarrow$ OF performance will decrease

Energy Reconstruction by **Convolutional Neural Networks**

Convolutional Neural Networks (CNNs)

- convolutional operation with certain **filter/kernel** size
- **activation function** gives opportunity to classify, weight, cut
- **feature maps** with different kernels can focus on different properties
- **training** minimizes difference between output and target

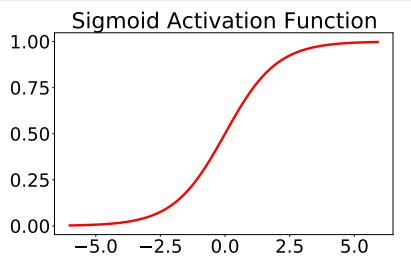


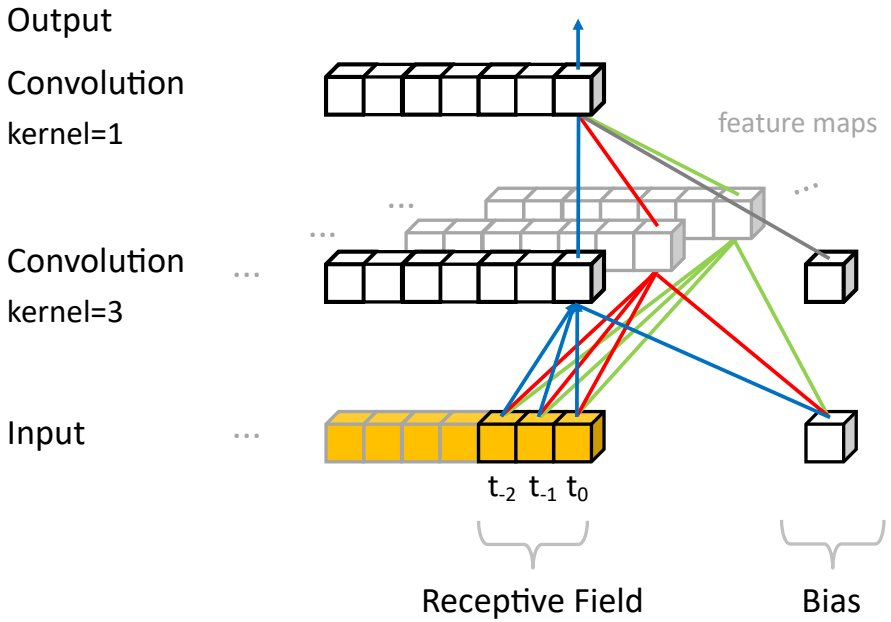


Energy Reconstruction by **Convolutional Neural Networks**

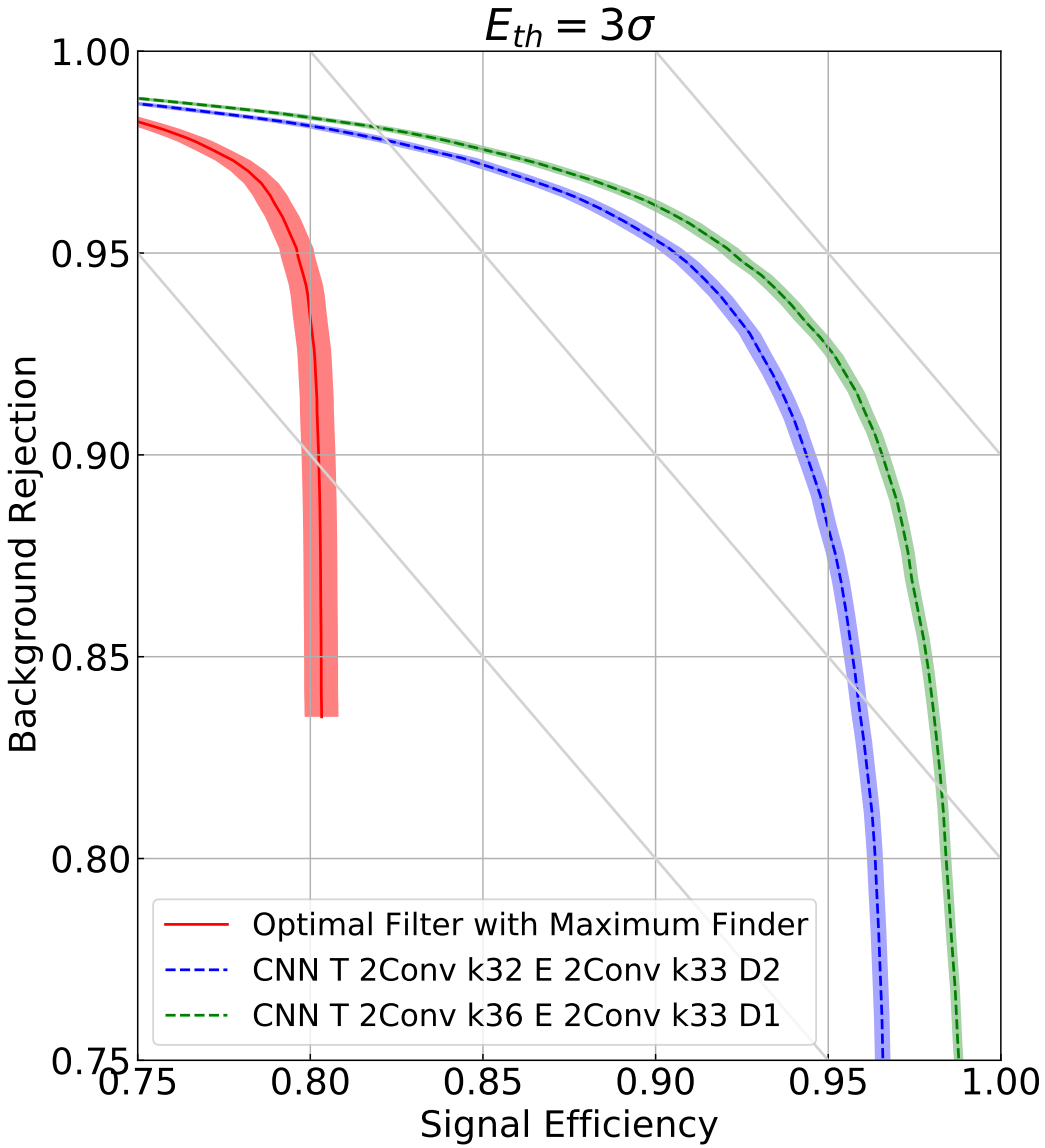
Convolutional Neural Networks (CNNs)

- convolutional operation with certain **filter/kernel** size
- **activation function** gives opportunity to classify, weight, cut
- **feature maps** with different kernels can focus on different properties
- **training** minimizes difference between output and target





Performance Evaluation: OF vs CNN - Trigger Efficiency



ROC curves

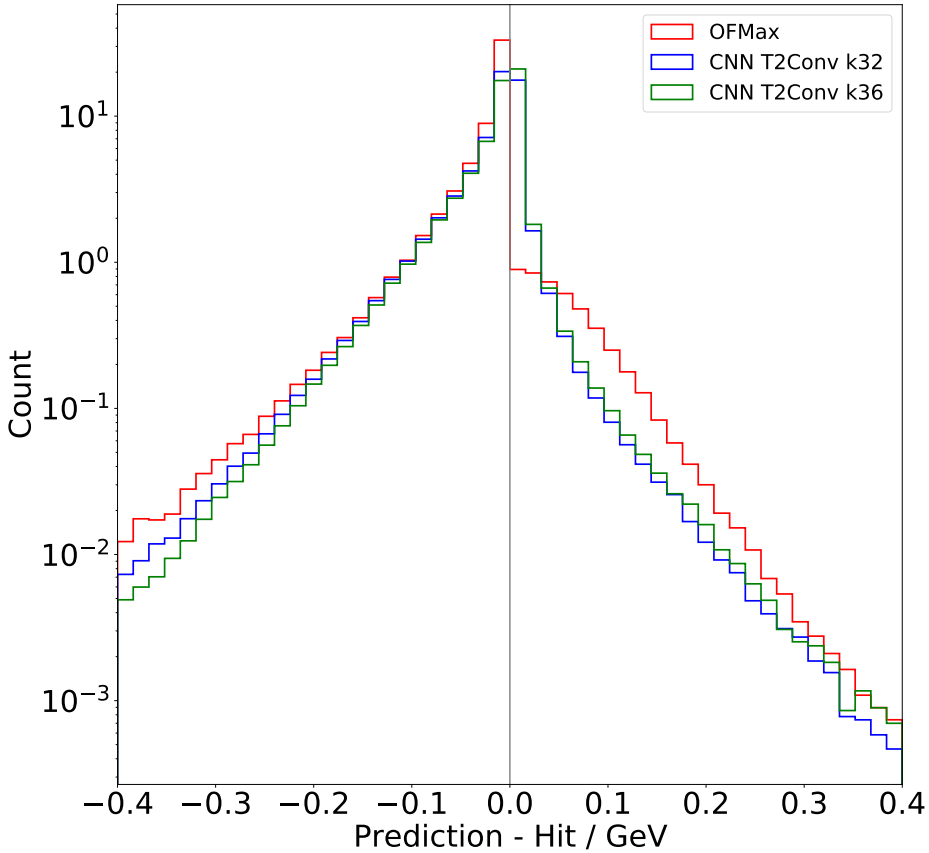
- indicate detection performance
- signal efficiency $= \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$
- background rejection $= \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}$
- dependent on threshold

$\longrightarrow$ CNNs outperform OF

Performance Evaluation: OF vs CNN - Energy Resolution

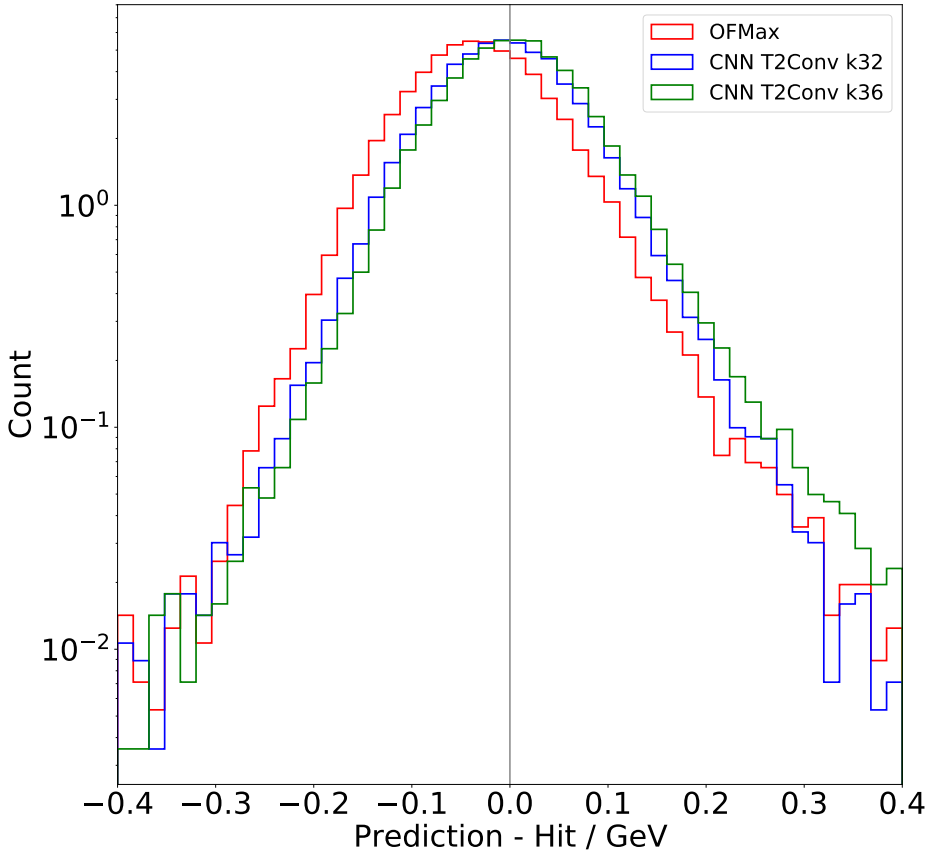
Pile-up Energy Region

0 GeV - 1 GeV



Signal Energy Region

1 GeV - 5 GeV

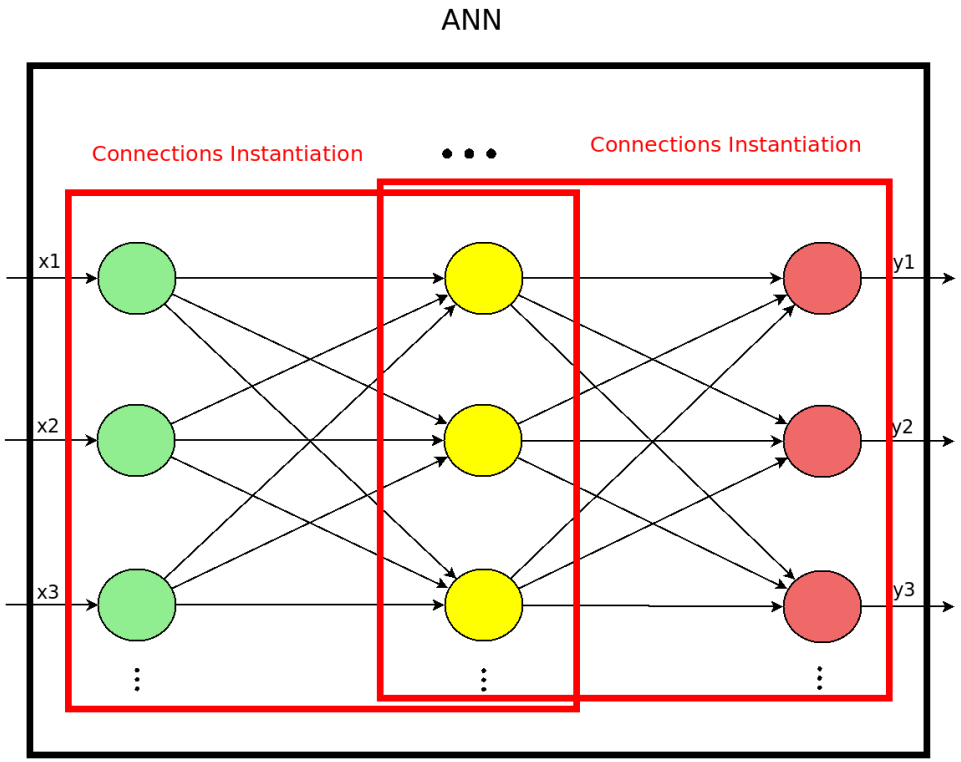


$\longrightarrow$ CNNs outperform OF

CNN Firmware Implementation

Connections subcomponent

- general *connection* component for all operations between neighboring layers of ANN
  - configurable #inputs, #outputs, activation functions
- multi-layer network component chains *connection* instances
  - capable to implement different kernel sizes, #feature maps and dilated CNNs
  - configurable with file produced after training (json)



Pipelining

Firmware Optimization

- aim core frequency of LASP FPGA used for Phase-2 data processing of $480\,\text{MHz} = 12 \times 40\,\text{MHz}$
- minimize latency as trigger accept must come in time
- meet resource limitations of FPGA

$\longrightarrow$ set pipeline registers as a compromise of the factors above

Pipelining over Input Samples

Start calculating layer output as soon as first required sample is available

BC	Input			
n	x_0	$y_0 = w_0 \cdot x_0 +$	$w_1 \cdot x_0 +$	$w_2 \cdot x_0 + b$
$n-1$	x_1	$y_1 = w_0 \cdot x_1 +$	$w_1 \cdot x_1 +$	$w_2 \cdot x_1 + b$
$n-2$	x_2	$y_2 = w_0 \cdot x_2 +$	$w_1 \cdot x_2 +$	$w_2 \cdot x_2 + b$