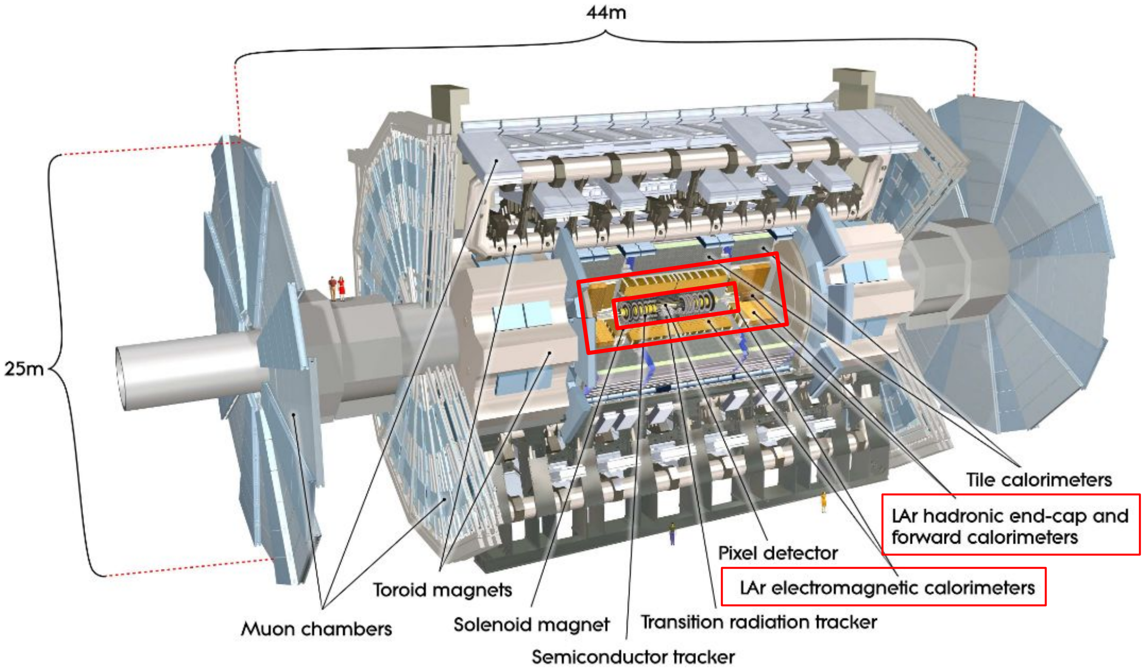
Convolutional Neural Networks for the Energy Reconstruction of ATLAS Liquid-Argon Calorimeter Signals

Christian Gutsche, Anne-Sophie Berthold, Nick Fritzsche, Max Märker, Johann Christoph Voigt and Arno Straessner

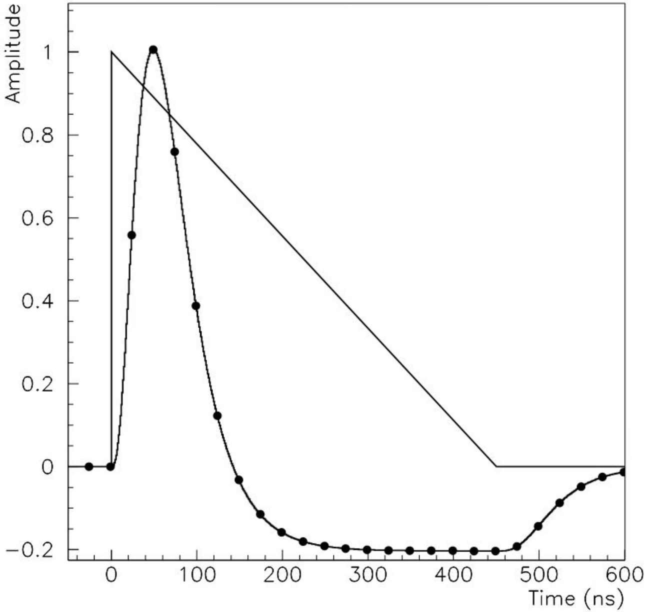
IKTP, TU Dresden

DPG Heidelberg 2022, March 22nd

The ATLAS Experiment and its LAr Calorimeter



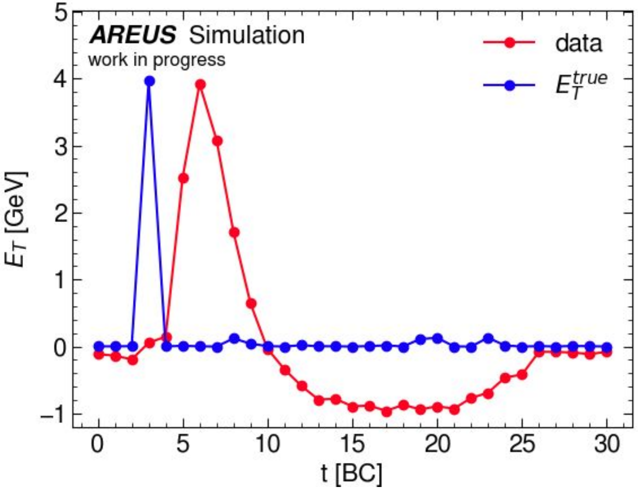
source: *The ATLAS Experiment at the CERN Large Hadron Collider, The ATLAS Collaboration et al, 2008 JINST 3 S08003*



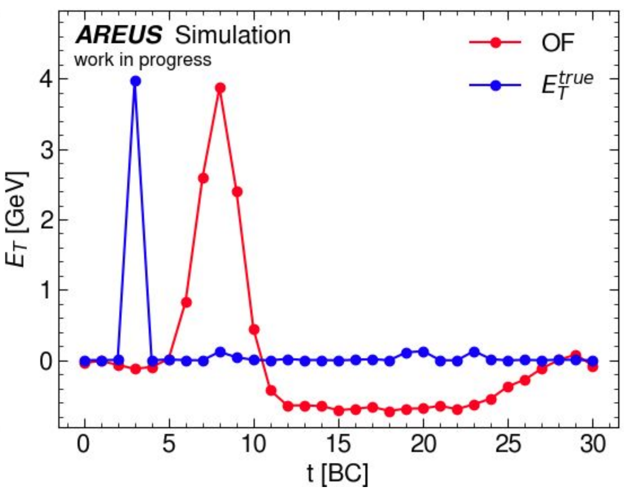
Free charge from ionization process creates triangular pulse
→ Reshaped to bipolar pulse

source: ATLAS *Liquid Argon Calorimeter Technical Design Report, 1996*

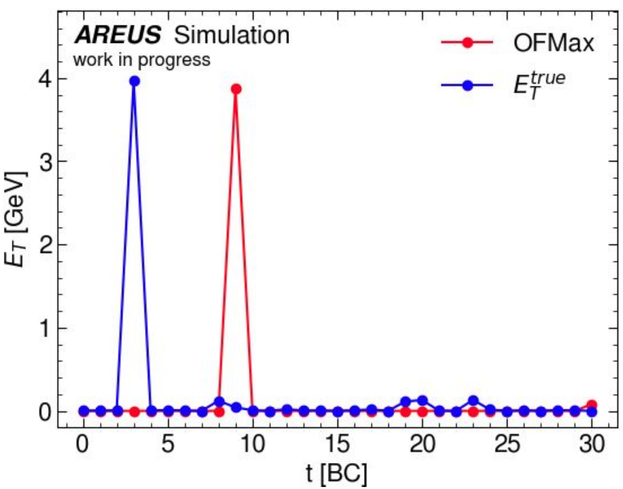
Signal processing at the LAr Calorimeter



Shaped data with noise (electronic noise, pile-up, ...)



Optimal Filter (OF) maximizes Signal-to-noise-ratio



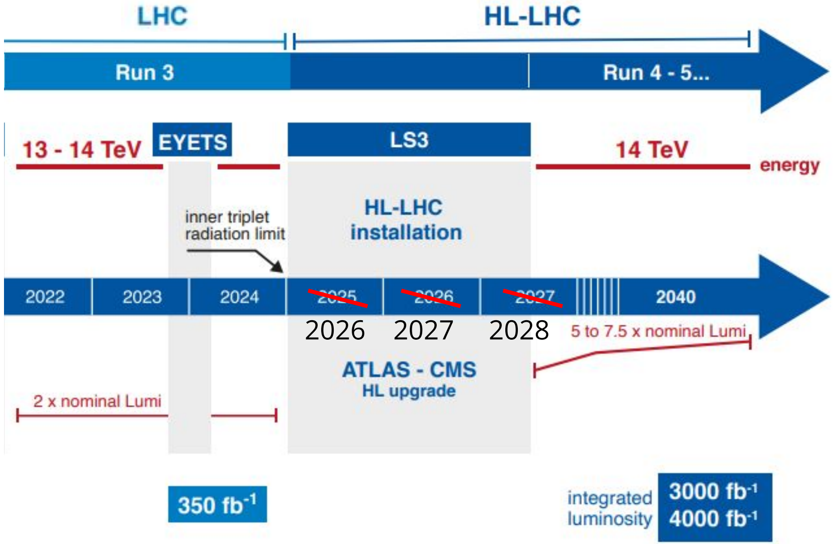
Maximum finder gives reconstructed energies

OF:

$$\mathrm{output}(t_j) = \sum_{i=0}^{n} w_i \cdot \mathrm{input}(t_{j-i})$$

with calculated weights w_i
here: n = 4

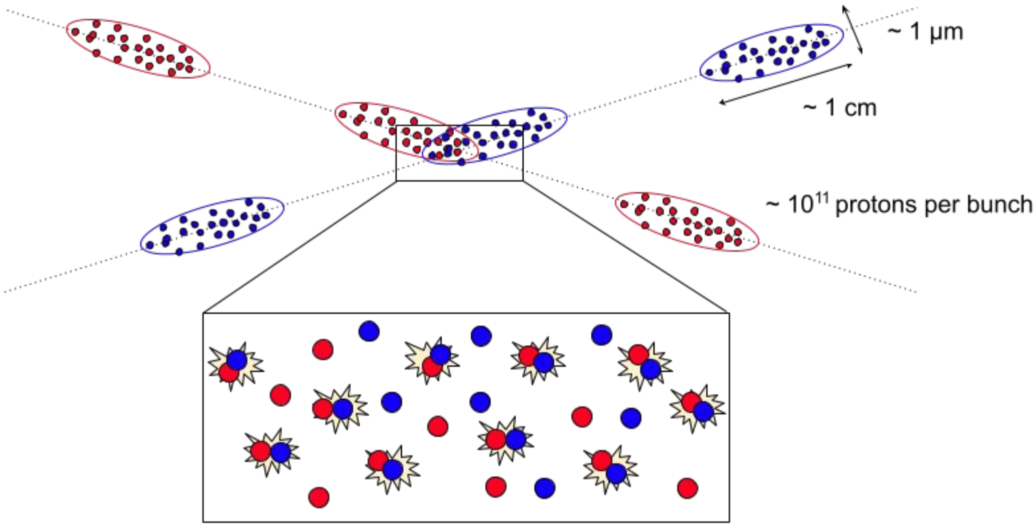
The HL-LHC



Upgrade schedule for the LHC.

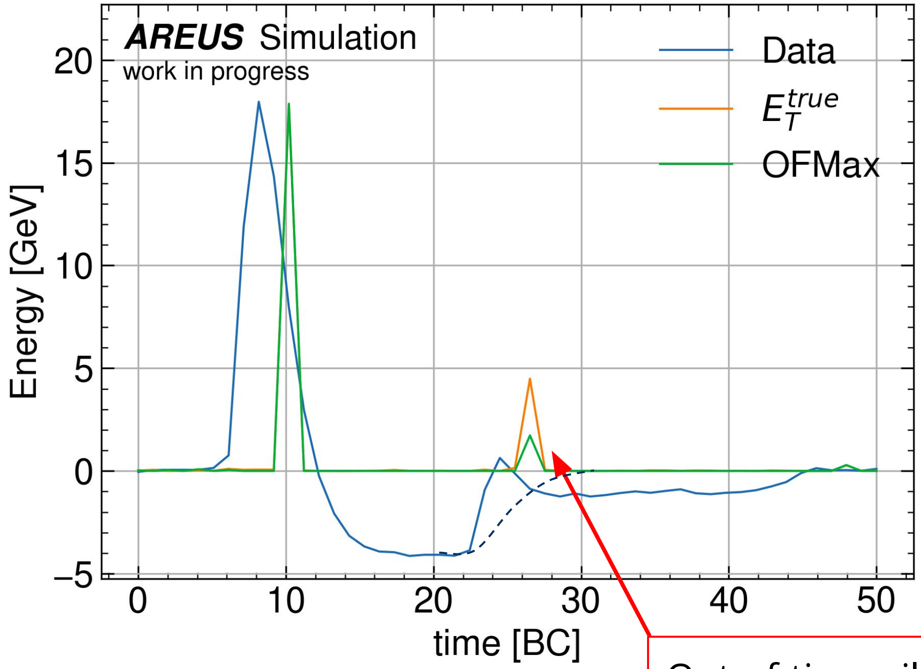
HL-LHC Upgrade Shutdown: 2026-2028
Start of Run 4: 2029

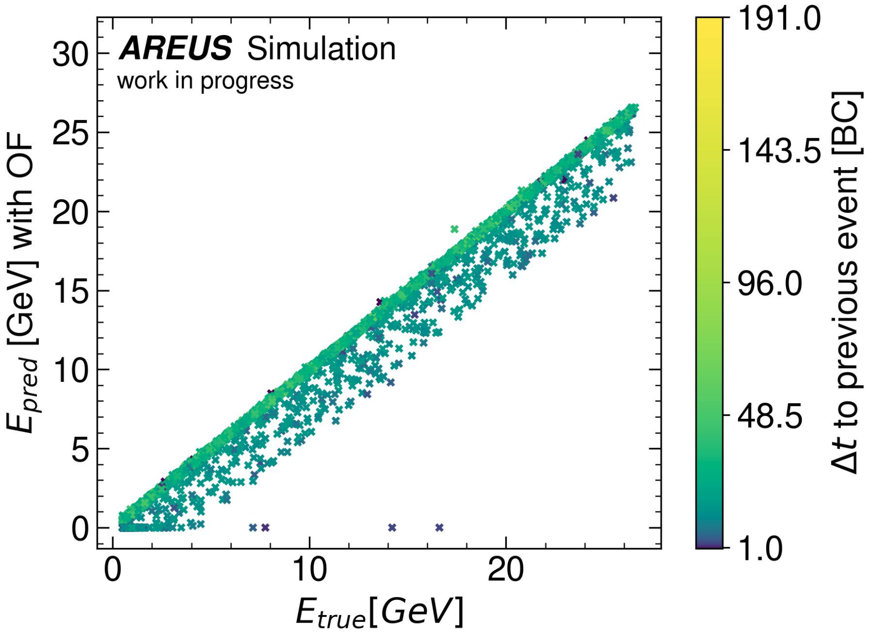
source: *hilumilhc.web.cern.ch*



40.000.000 bunch crossings (BC) per second
(1 BC every 25ns)
LHC: ~ 20 proton-proton-collisions per BC
HL-LHC: ~ 140 *pp*-collisions per BC
→ Optimal Filter will struggle with pile-up

Optimal Filter at the HL-LHC





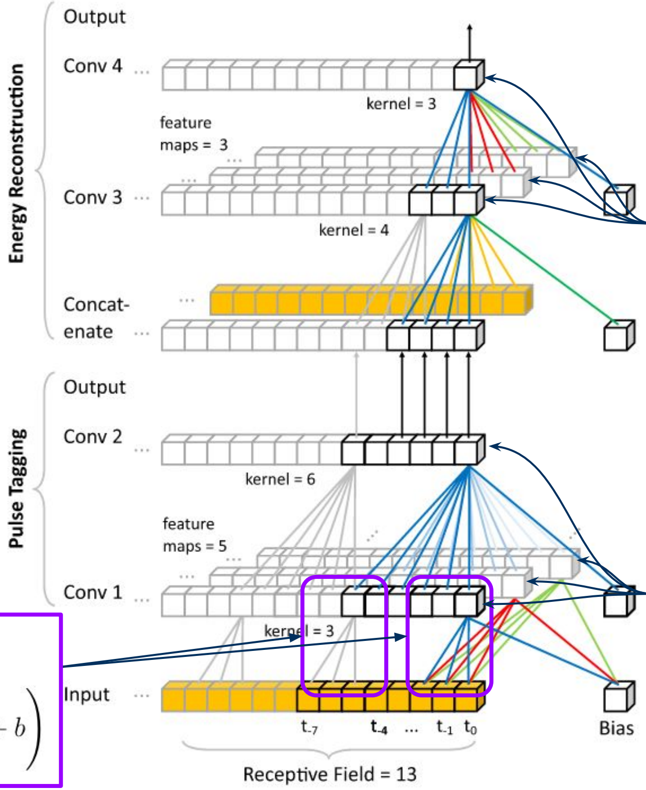
Out-of-time pile-up leads to underestimation of energy = Biggest source of errors

Solution: Neural Networks?

source: Artificial Neural Networks on FPGAs for Real-Time Energy Reconstruction of the ATLAS LAr Calorimeters, Georges Aad et al, Computing and Software for Big Science 5, 19 (2021)

Idea: Artificial Neural Networks (ANNs) could learn to correct pile-up and noise

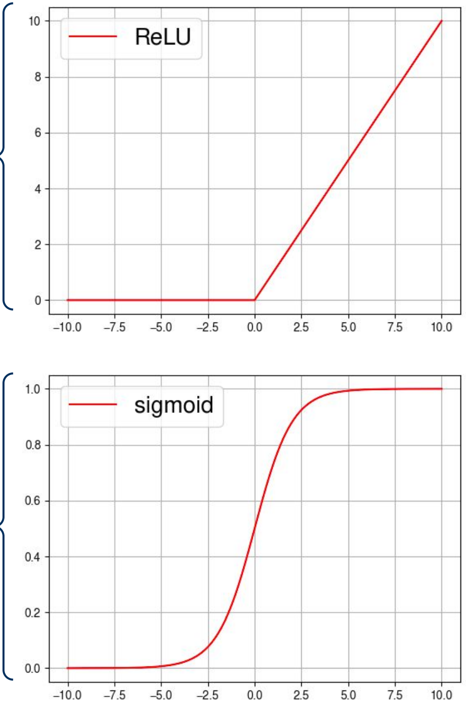
Currently tested ANNs are trained in 2 stages:
1. Tagging part with binary cross-entropy
2. All layers with mean absolute error



ANNs work similar to OF, but more features:

$$\text{feature map output}(t_j) = \text{activation}\left(\sum_{i=0}^{n} w_i \cdot \text{input}(t_{j-i}) + b\right)$$

activation functions



TECHNISCHE UNIVERSITÄT DRESDEN

ATLAS EXPERIMENT

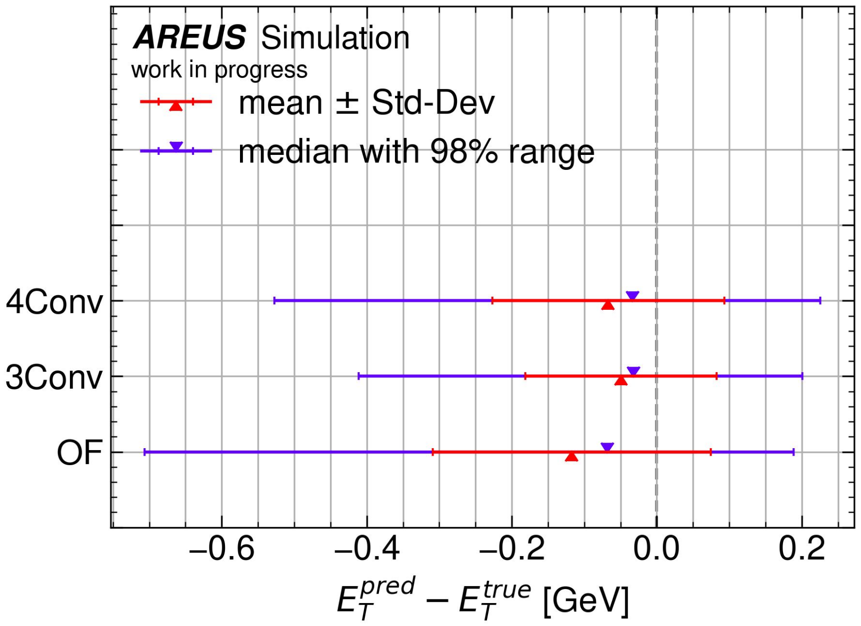
DRESDEN concept

Performance of current ANNs

- Good results compared to OFMax in simulation
- Difference of predicted and true Energy smaller on average and varies less
- ANNs handle pile-up better
- My task: optimize these ANNs

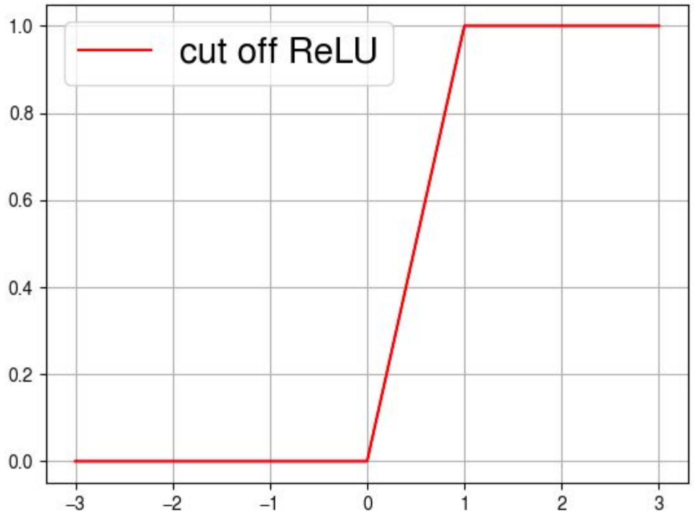
Plot only includes data with:

$E^T_{\text{true}} > 3\sigma$ of total noise $= 240 MeV$



Hardware-friendlier activation functions

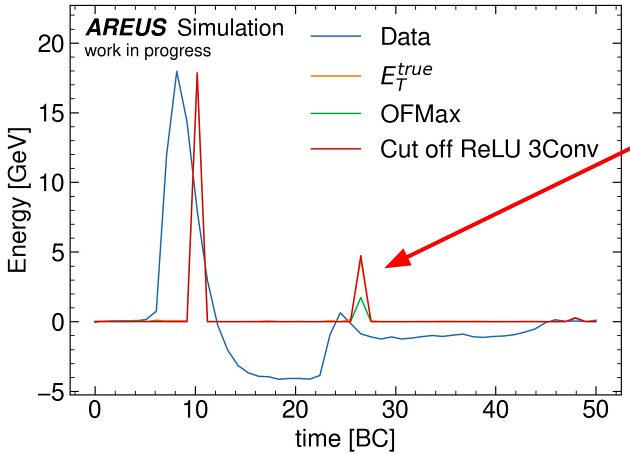
Simplest replacement for sigmoid:
cut off ReLU



→ Activation function changed architecture, training structure kept

For 3Conv and 4Conv ANNs:

- Performance got less reproducible for multiple trainings
- Good trainings produce similar results

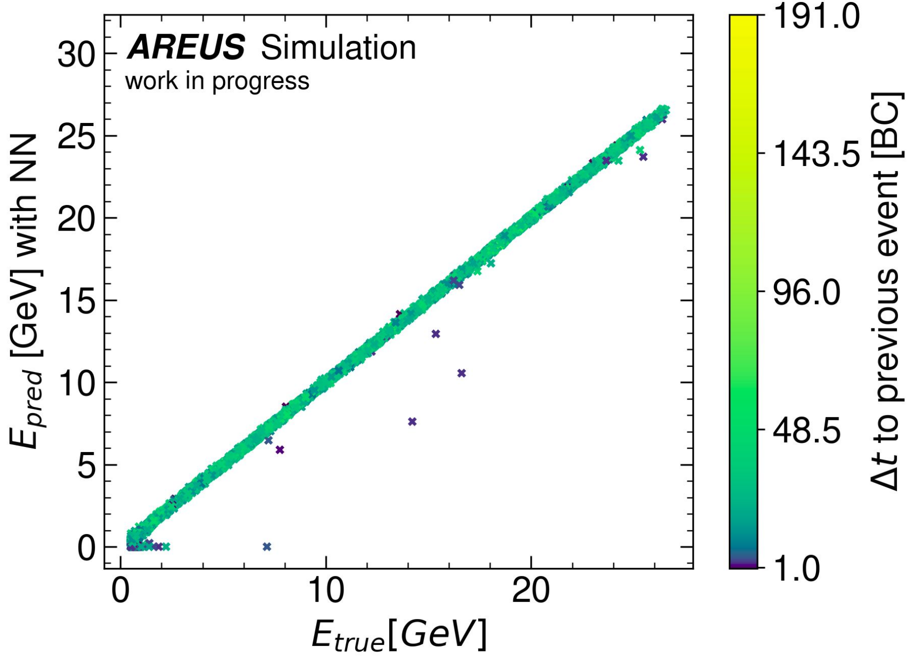


NN handled pile-up better than OFMax

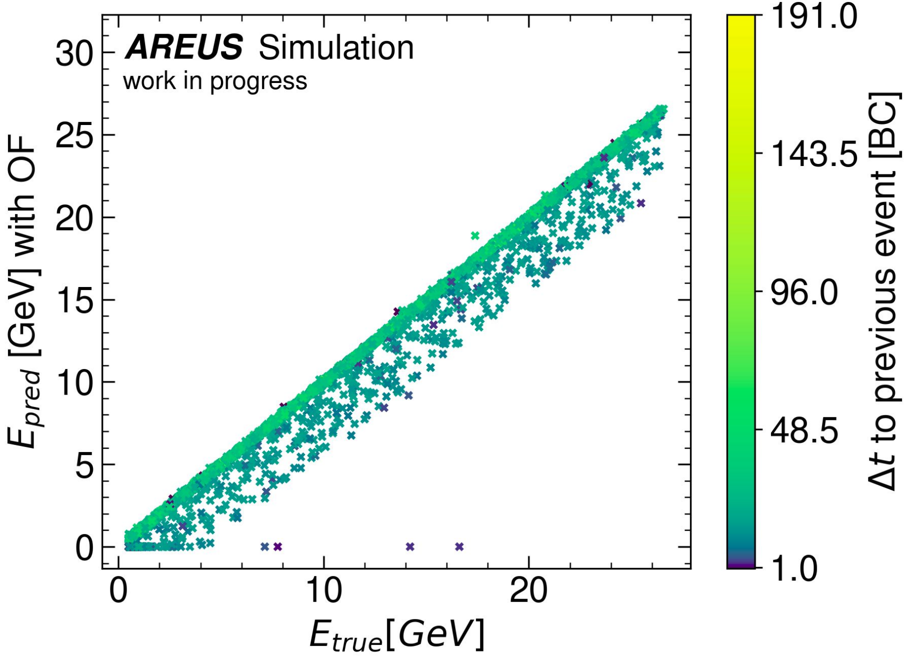
ANNs with cut off ReLU as activation

Energy reconstruction and dependency on temporal distance to previous event:

Neural Network



OFMax

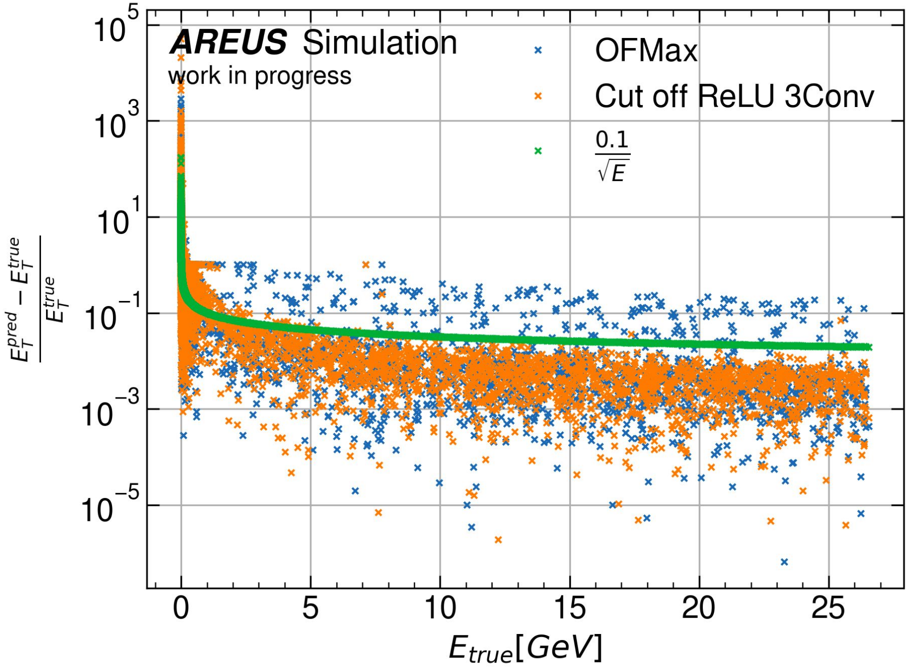


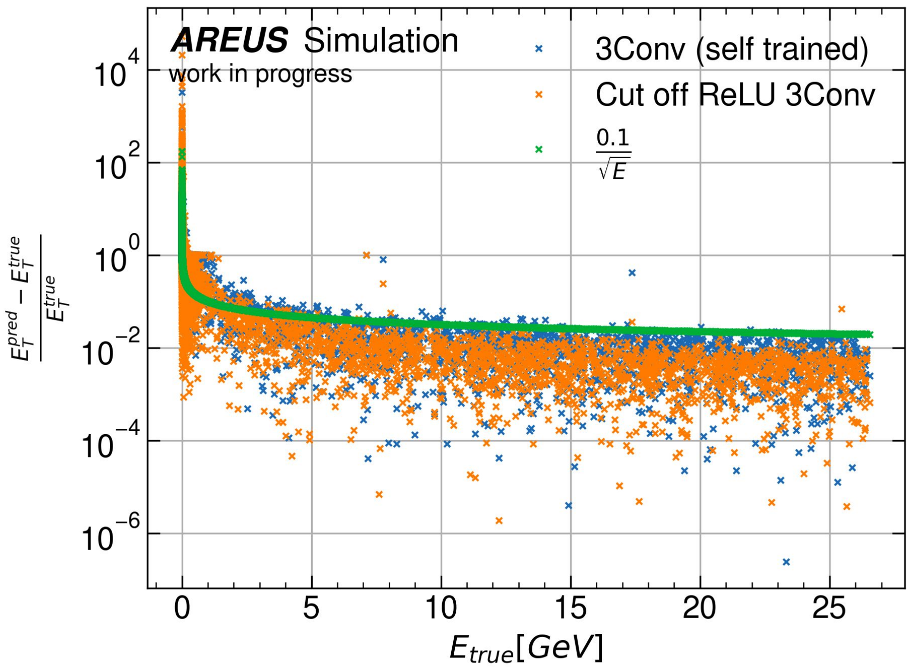
Summary and Outlook

- Alternatives for Optimal Filter are needed for high event rate at HL-LHC
- ANNs seem to be promising candidates for energy reconstruction with high pile-up noise
- Reliability with varying pulse shapes needs to be tested
- Reproducibility is an important criterion since ANNs need to be retrained/recalibrated
- To improve results:
  - New activation function was tested
  - Different loss functions are currently tested to improve results

Backup

ANNs with cut off ReLU as activation





Optimization using loss functions

= Loss defines which quantity will be minimized
→ Possibility to let ANN focus on specific details of the data without changing number of parameters.

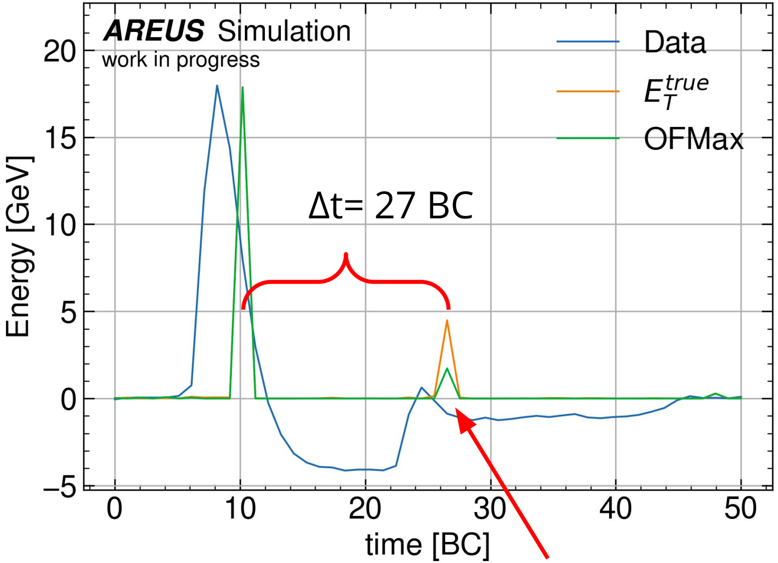
Weigh specific event in the loss function using a weight matrix Λ

$$\text{weighted mean absolute error} = \frac{\sum_{i=0}^{N} \Lambda_i \cdot |E^T_{\text{pred},i} - E^T_{\text{true},i}|}{N}$$

Idea:
Weight events with short temporal distance to previous pulse

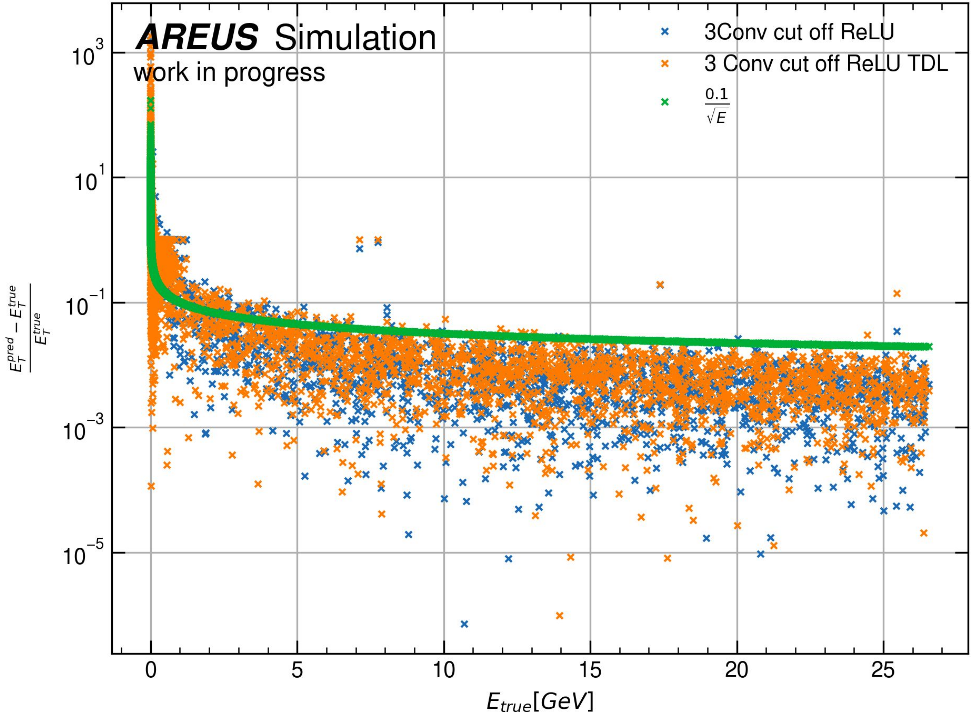
for example:

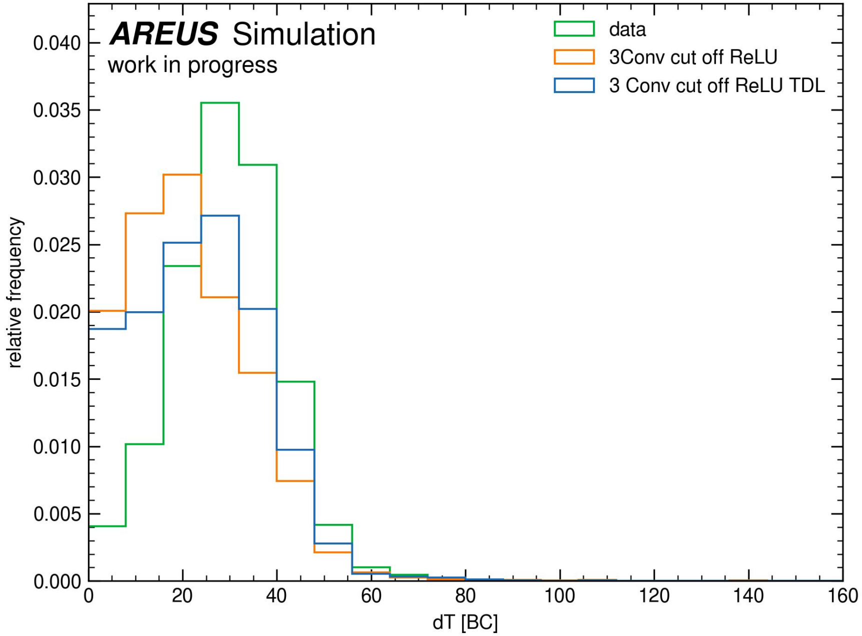
$$\Lambda_i = \begin{cases} 5 & \text{if } \Delta t < 5 \\ 10 & \text{if } \Delta t < 30 \\ 1 & \text{otherwise} \end{cases}$$



second peak gets weight 10

Optimization using loss functions





→ Seems to improve results for small temporal distance events but does not improve overall performance