Machine Learning for Energy Reconstruction at ATLAS

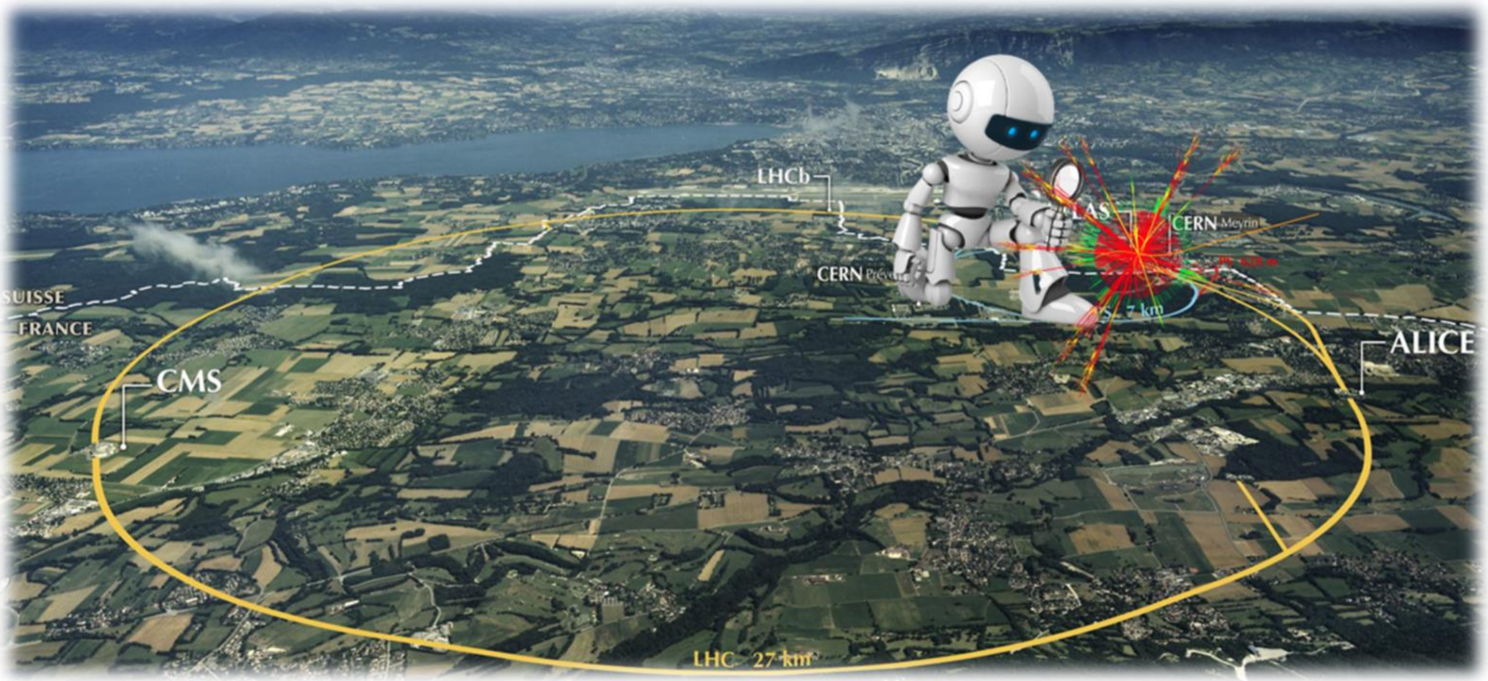
2021 CAP Virtual Congress - June 6-11

Presented by Luke Polson

Supervisor: Dr. Michel Lefebvre

University of Victoria

ATLAS EXPERIMENT



Introduction

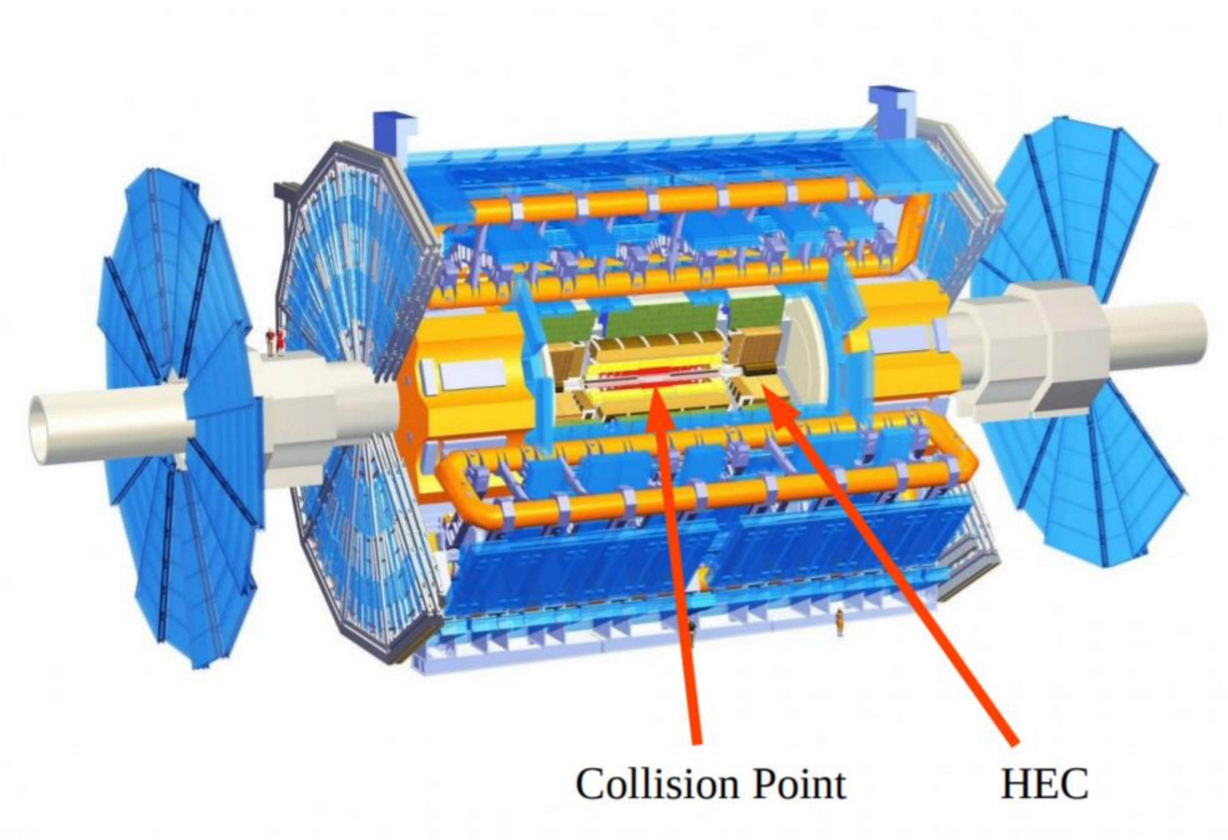
- A given event in ATLAS produces many particles
  - Energy of particles measured in different subsystems of the detector
  - Each energy has an associated uncertainty
  - Energies added together to get net energy for a reconstructed object (such as a jet)

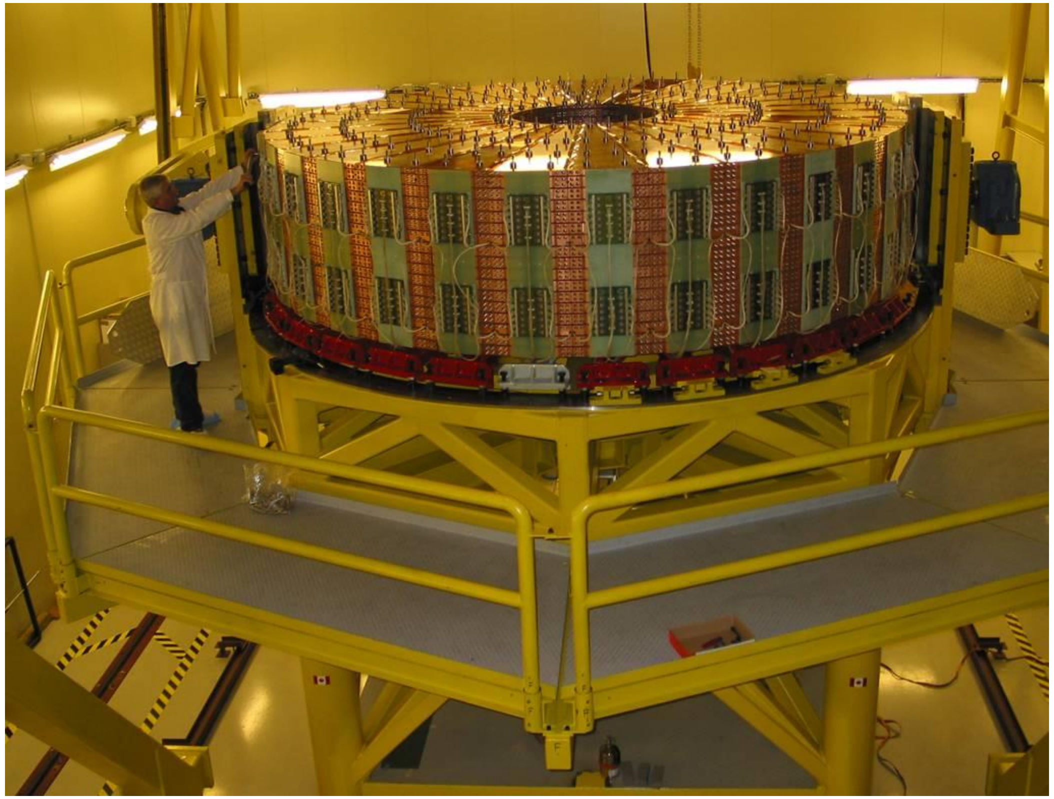
$$\underset{\text{Subsystem 1}}{E_1 \pm \delta_{E_1}} \quad \underset{\text{Subsystem 2}}{E_2 \pm \delta_{E_2}} \quad \underset{\text{Subsystem 3}}{E_3 \pm \delta_{E_3}} \implies \begin{aligned} E_{\text{net}} &= E_1 + E_2 + E_3 \\ \delta_{E_{\text{net}}} &= \sqrt{\delta_{E_1}^2 + \delta_{E_2}^2 + \delta_{E_3}^2} \end{aligned}$$

- Less uncertainty in energy enables
  - more precise measurements on fundamental particles like the **Higgs Boson**
  - Greater reach in searches, such as the search for **dark matter**

ATLAS Hadronic End Cap

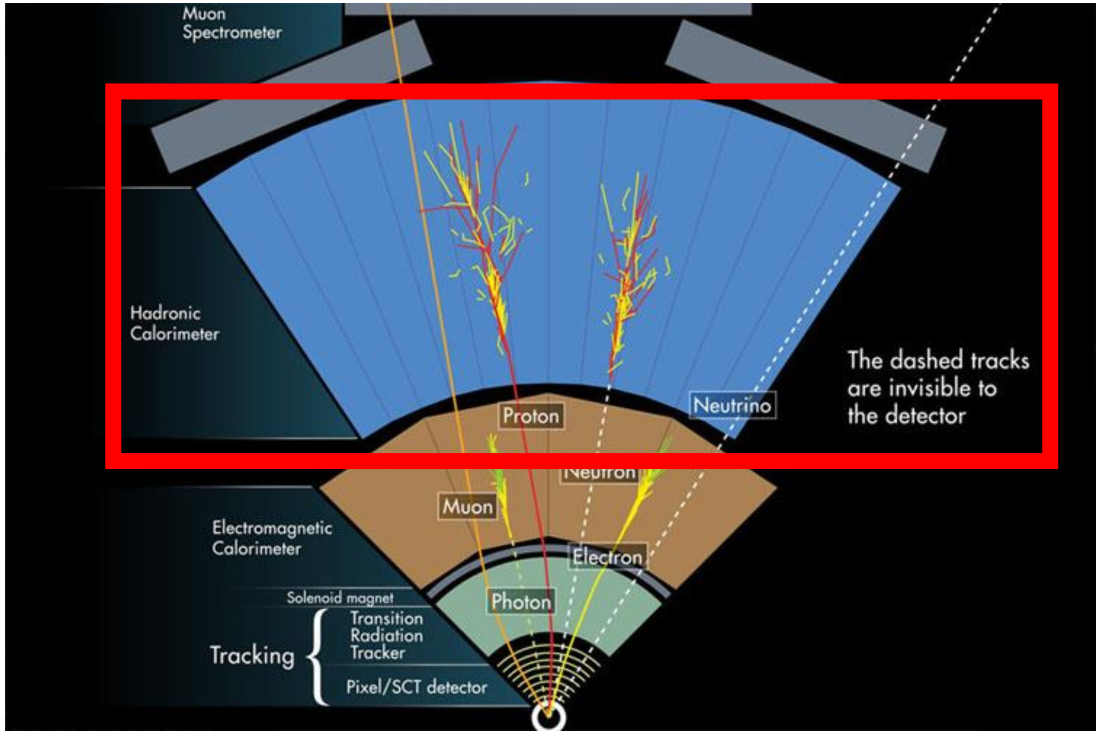
- Measures energy of hadrons by sampling induced showers

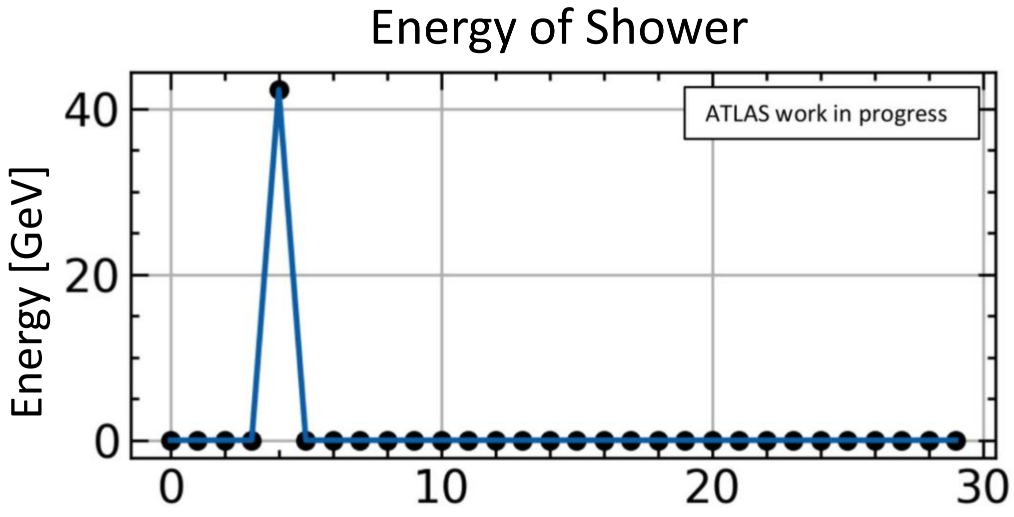


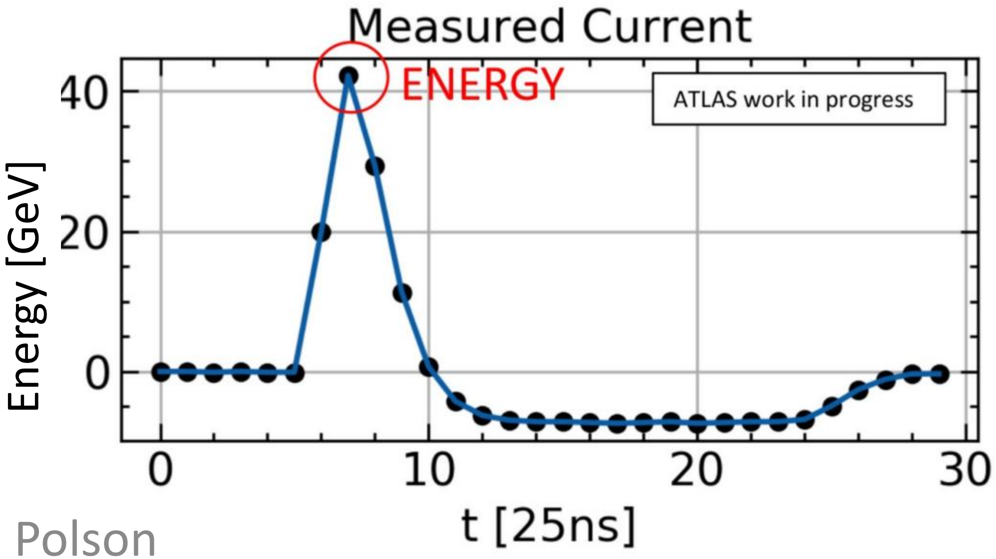


Detection Mechanism

1. Charged particles from induced shower ionize liquid argon
2. Electric field causes electrons to drift
3. Drifting electrons = measurable current





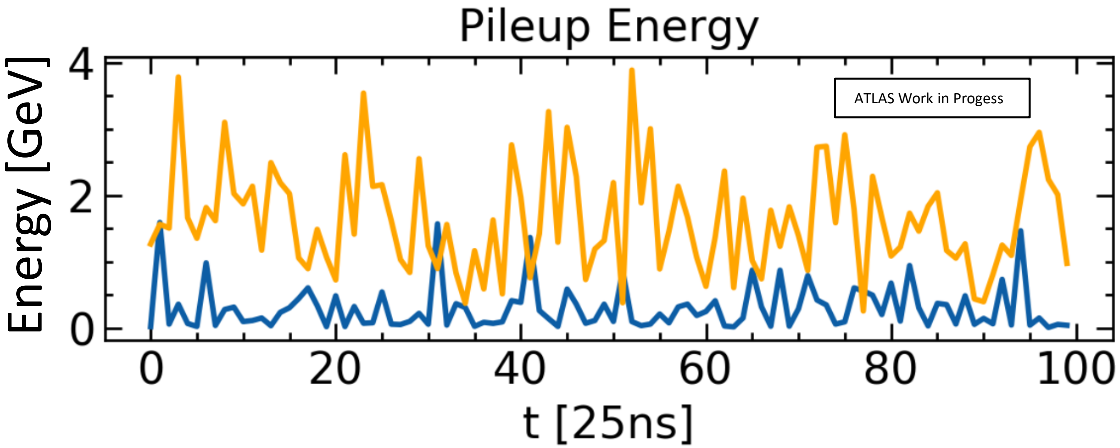


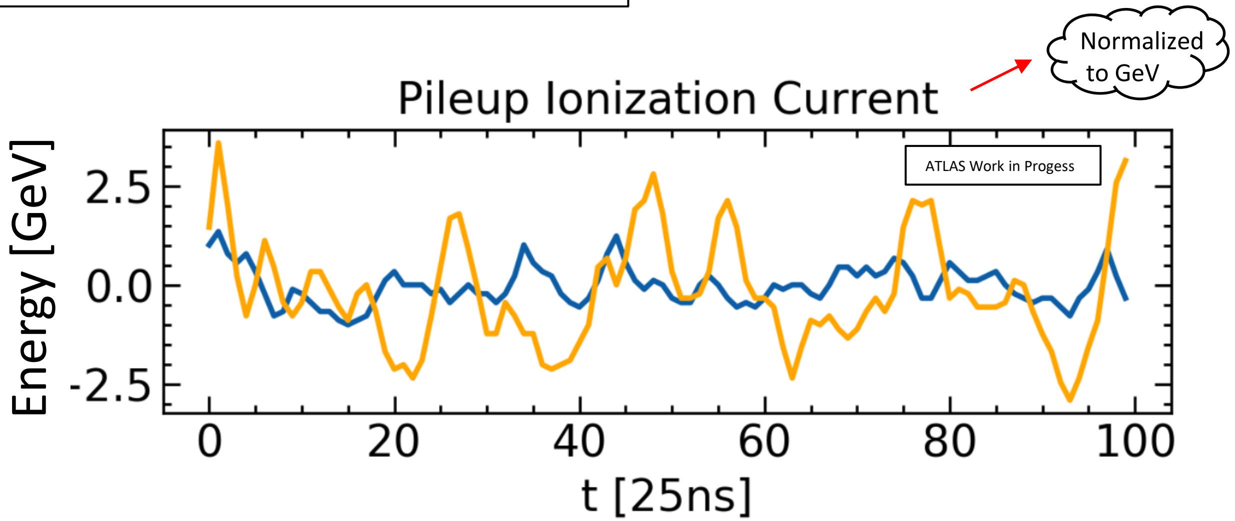
Pileup Noise

- LHC Plans to upgrade beam intensity in the future [1]
- Increased beam intensity => more detected events
- Energy of individual processes becomes difficult to measure

That's one thing I hate! All the noise, noise, noise, noise!
- The Grinch

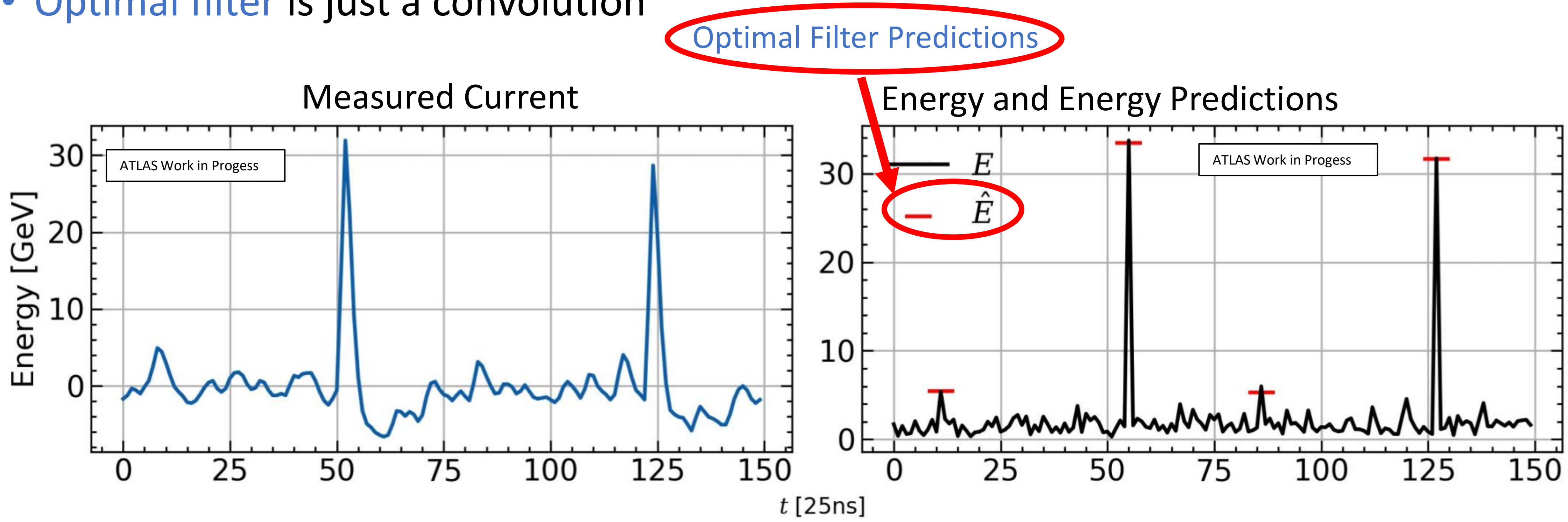
Blue: Present day ATLAS
Orange: Expected future ATLAS



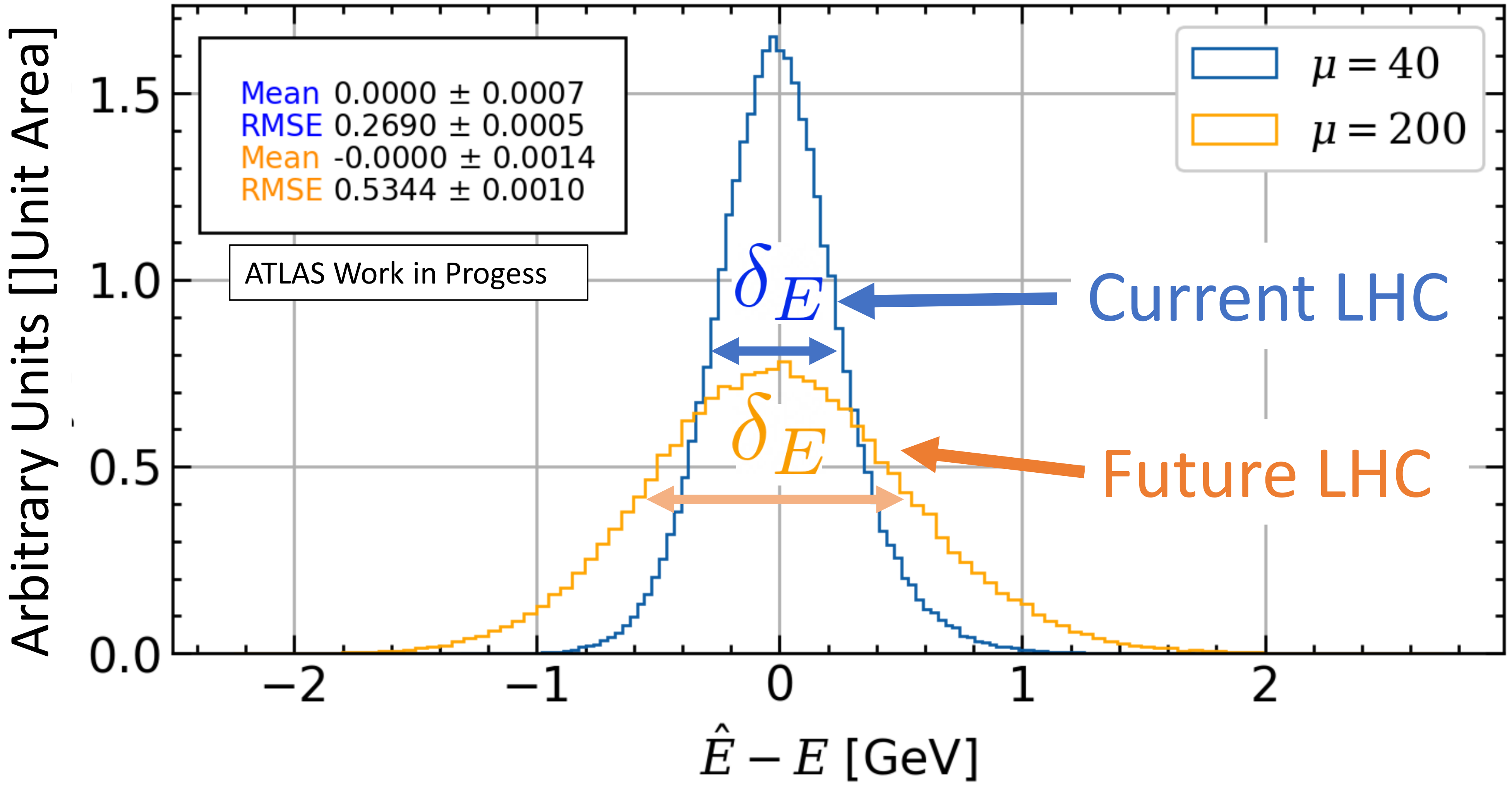


Pileup Noise

- Noise makes it difficult to predict energy of interesting events
- Current technique used for energy prediction is the **Optimal Filter technique** [2]
  - Optimal filter is just a convolution



Pileup Noise



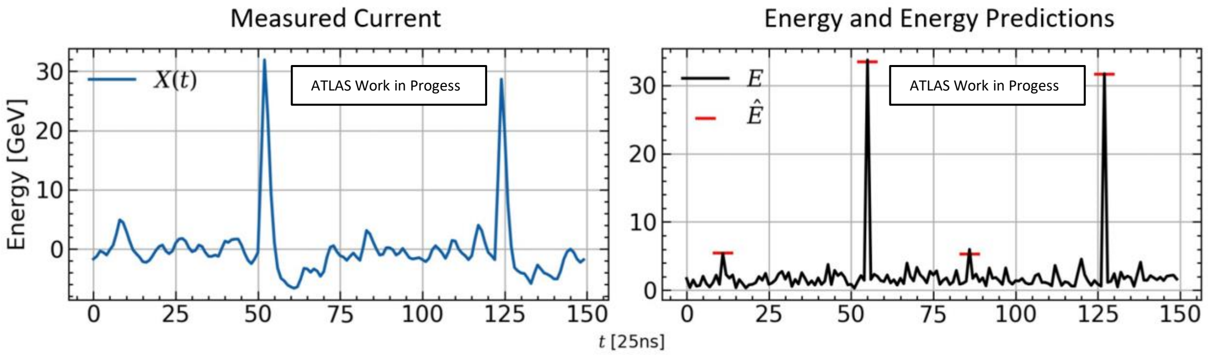
Convolutional Neural Networks

- Discrete Convolution:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m],$$

Time series

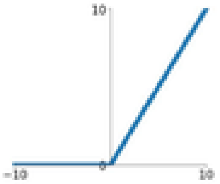
Filter



- The Basic Idea behind CNN
  1. Apply convolution with $g_i[n]$
  2. Add a constant (bias)
  3. Apply non-linear function

repeat

ReLU
$\max(0, x)$

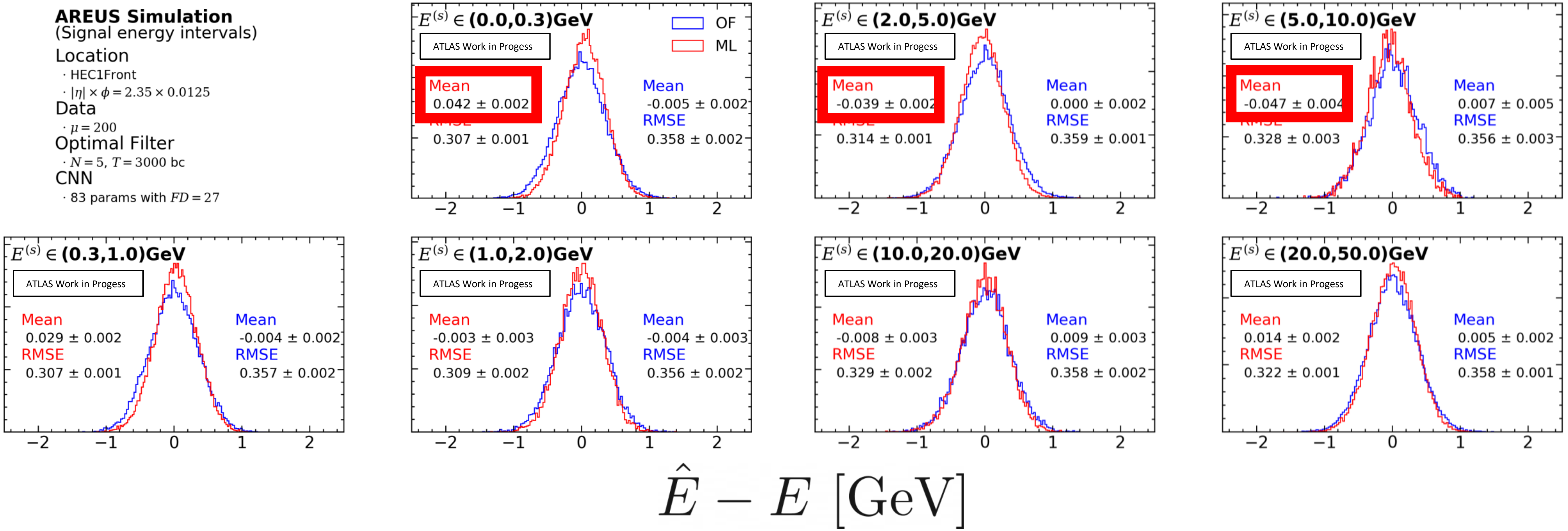


- The Basic Idea behind obtaining g
  1. Define "error" L (such as RMSE)
  2. Evaluate $\nabla L(g_1, g_2, \ldots)$
  3. Move in direction of decreasing L

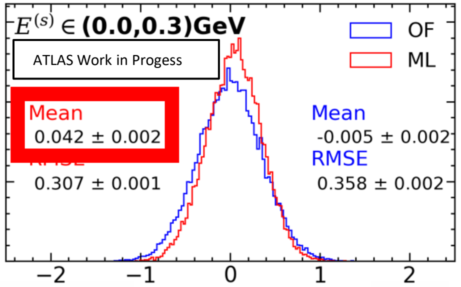
repeat

Issues with Machine Learning

- Optimal Filter has no bias
- RMSE loss function results in the CNN making biased predictions



New Loss Function



$$L = \sqrt{\left\langle \left(E - \hat{E}\right)^2 \right\rangle} + \alpha \sum_j \left| \left\langle E - \hat{E} \right\rangle_j \right|$$

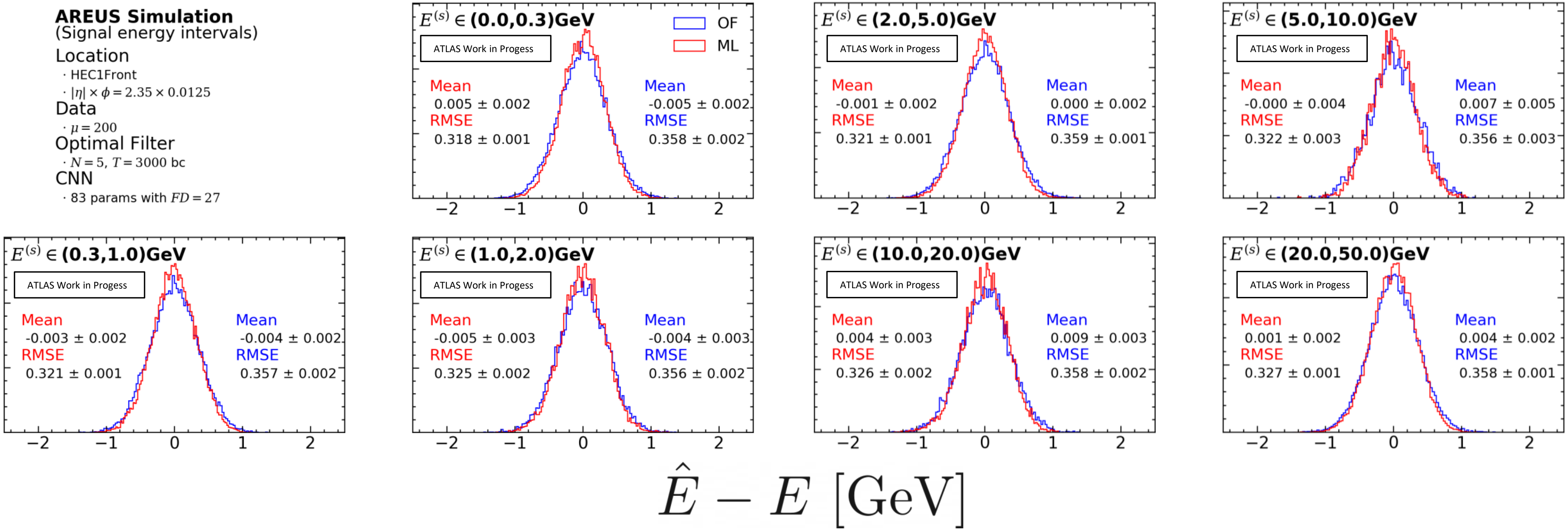
Regular RMSE loss function

Parameter I had to optimize for

Bias of energy predictions in energy region j

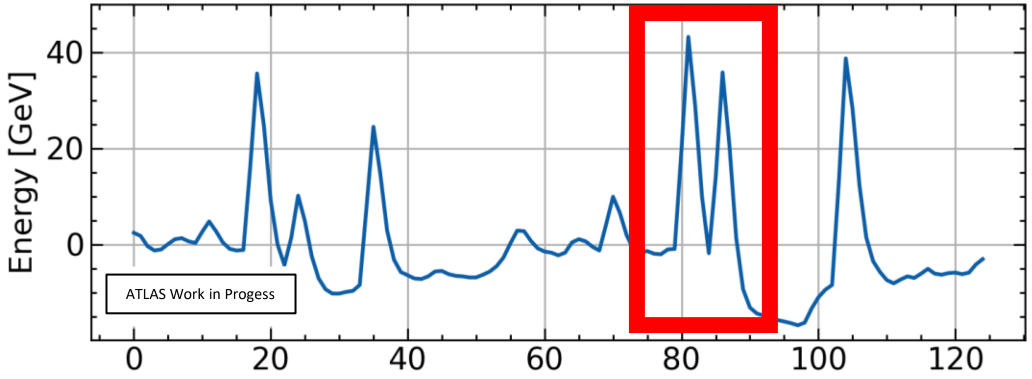
Algorithm Comparison

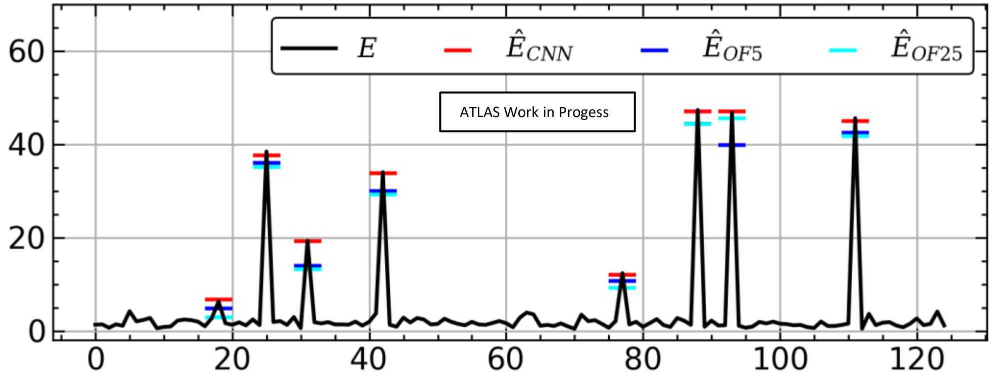
- The Convolutional Neural Network outperforms the optimal filter
  - For all different energy intervals!
  - Training requires special loss function



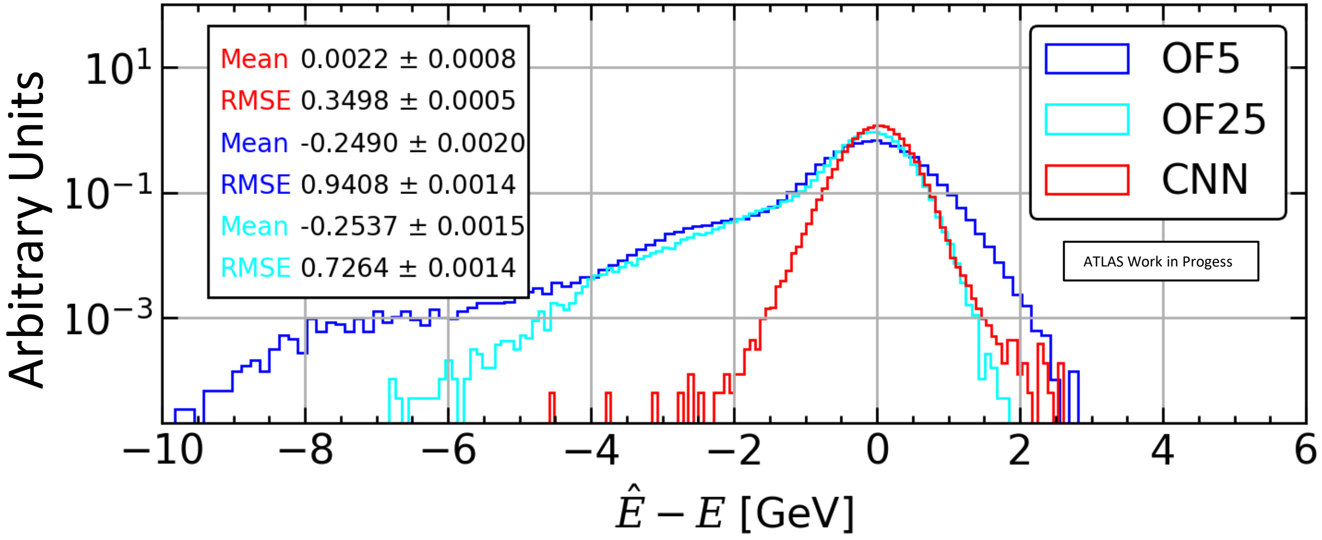
Algorithm Comparison

- Convolutional Neural Network excels when pulses are close together





t [25ns]



Conclusion

- Optimal Filter is the presently used technique for reconstructing energy in the ATLAS Hadronic Endcap Subsystem of ATLAS
  - Perform a convolution on the measured current
- Optimal Filter performs worse in expected future LHC conditions
  - Energy measurements have a greater associated uncertainty

$$\delta_E^{(\text{Future LHC})} > \delta_E^{(\text{Current LHC})}$$

- The convolutional neural network outperforms the optimal filter in future LHC conditions
  - Requires a special loss function during training

$$\delta_E^{(\text{CNN})} < \delta_E^{(\text{OF})}$$

References

[1] G Apollinari, I Bejar Alonso, O Bruning, M Lamont, and L Rossi. High-Luminosity Large Hadron Collider (HL-LHC): Preliminary Design Report. CERN Yellow Reports: Monographs. CERN, Geneva, 2015

[2] W.E. Cleland and E.G. Stern. Signal processing considerations for liquid ionization calorimeters in a high rate environment. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 338(2):467 – 497, 1994

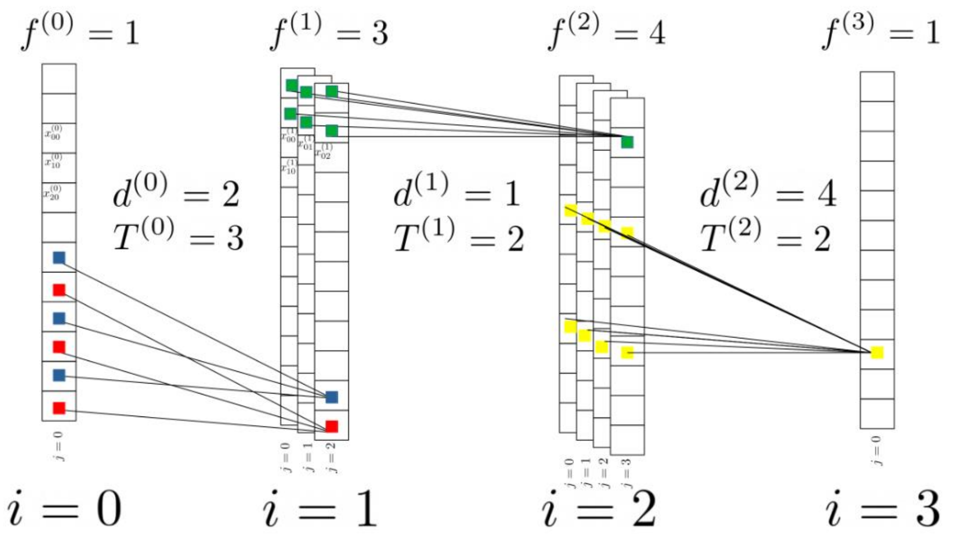
[3] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio (arxiv:1609.03499)

Backup: CNN Mathematics

INPUT   OUTPUT

$$(X_t = x^{(0)}_{t0}) \rightarrow x^{(1)}_{tj} \rightarrow \ldots \rightarrow x^{(L-1)}_{tj} \rightarrow (x^{(L)}_{t0} = \hat{y})$$

$$x^{(i+1)}_{tj} = \sum_{m=0}^{f^{(i)}-1} \sum_{n=0}^{T^{(i)}-1} R^{(i)}\left(A^{(i)}_{jmn} x^{(i)}_{n'm} + b^{(i)}_j\right)$$



1. i represents the layer index, and $f^{(i)}$ is the number of feature maps (or dimensionality) of the time series in layer i. The first and last layers have $f^{(i)} = 1$, for example, corresponding to a univariate time series. L is the number of layers in the network.
2. j, which can take on the indices 0 to $f^{(i)} - 1$ in layer i, represents the feature map (or filter) index. t represents the time index for the time series.
3. $A^{(i)}_{jmn}$ is the weight matrix and $b^{(i)}_j$ is the bias term for feature map j in layer i. These parameters are modified during the training procedure.
4. $n' = t - d^{(i)}n$ where $d^{(i)}$ is the dilation rate of layer i.
5. $T^{(i)}$ is the size of the filter in layer i. It is also typically referred to as the kernel size.
6. $R^{(i)}$ is the activation function for layer i of the neural network. This thesis uses the activation function $R^{(i)}(x) = 0$ if $x < 0$ and $R^{(i)}(x) = x$ if $x \geq 0$. This is known as the *relu* loss function.

Backup: CNN Configuration Used

- Structure of Convolutional Neural Network (Based on WaveNet [3])
  - 3 Layers (Dilation rates of 1, 3, 1 respectively)
  - Filters per layer is 3, 3, 1 respectively
  - Kernel size is 7, 7, 3 respectively
  - Allowed to break causality by 9 bunch crossings (225 ns)
  - 100 parameters total

Layer (type)	Output Shape	Param #
input_10 (InputLayer)	[(None, None, 1)]	0
conv1d_18 (Conv1D)	(None, None, 3)	24
conv1d_19 (Conv1D)	(None, None, 3)	66
concat (Conv1D)	(None, None, 1)	10

Total params: 100
Trainable params: 100
Non-trainable params: 0