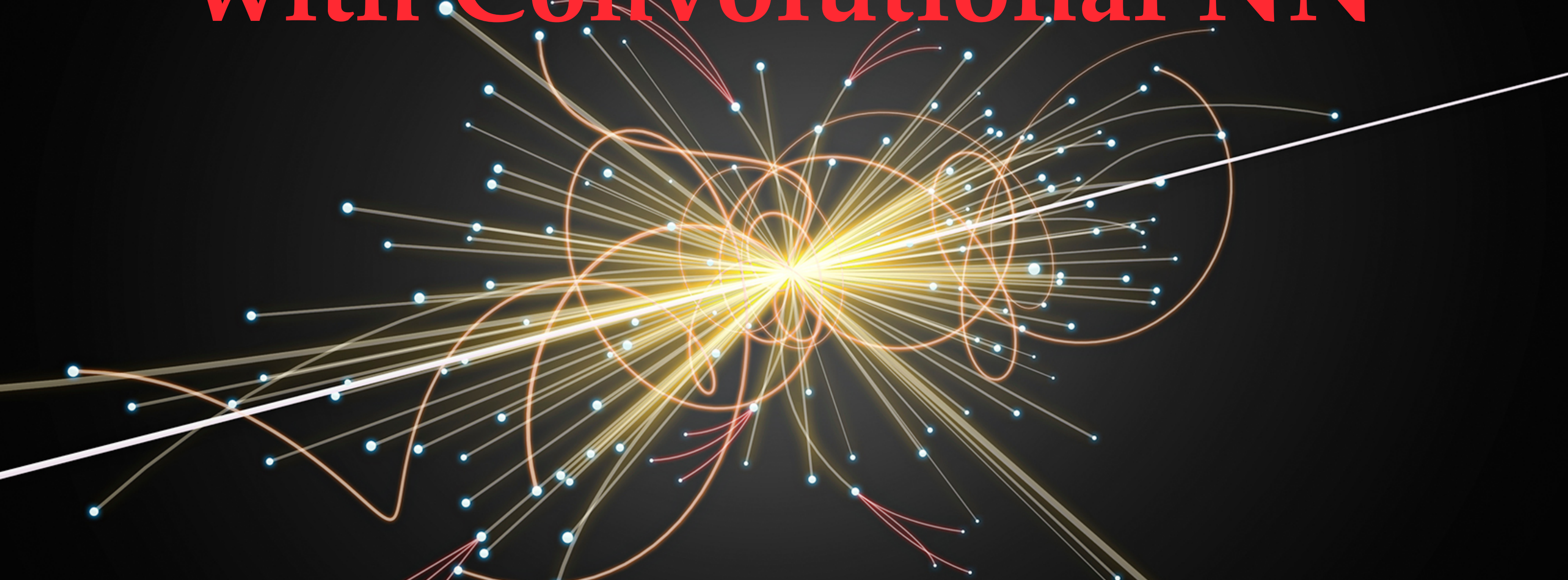
Electron/Photon Energy regression with Convolutional NN

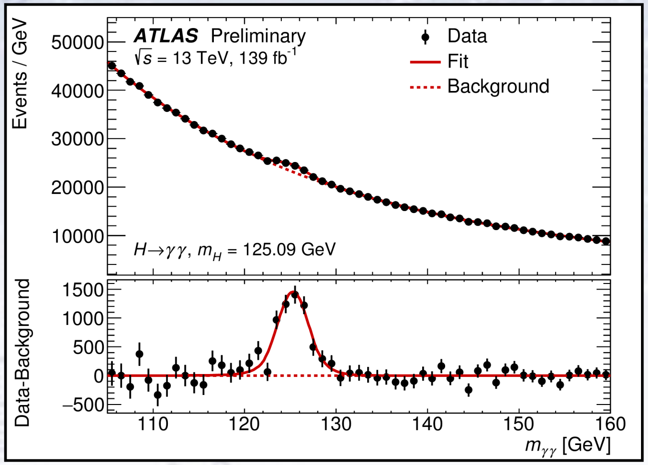


Lukas Ehrke, Frederik Faye, Benjamin Henckel, Loui Wentzel,
Daniel Nielsen, and **Troels C. Petersen**
Niels Bohr Institute, Copenhagen

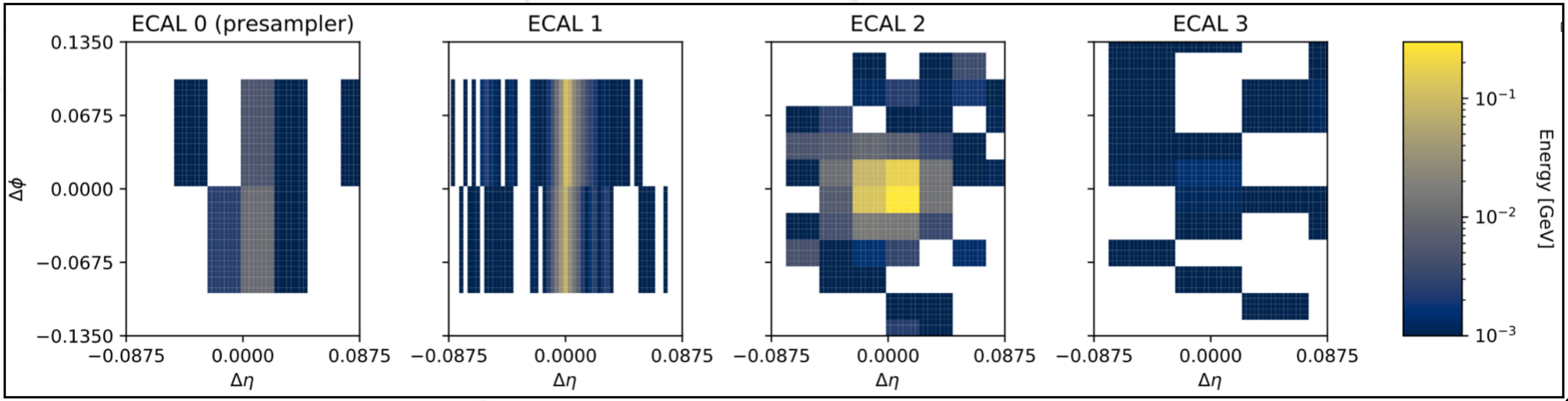
Sharpening the H ➝ γγ peak

The photon energy resolution $\sigma(E_\gamma)$ is the **dominant performance parameter** for the H ➝ γγ analysis, and has influenced the calorimeter design of ATLAS and CMS.

Goal: Improve this resolution in ATLAS.



To do so, we have considered the cell energies in the LAr calorimeter as pixels in four images (one for each layer), and trained a Convolutional Neural Network (CNN) to estimate the energy of electron/photon candidates.



The data

We used several samples to try to span the relevant energy range, but only partially succeeded. We insisted on using physics channels to get the most realistic settings. Unfortunately, cell re-weighting was not used (not available).

The variables are both scalar and cell based. The scalars can be seen below.
The total data sample size was 1-5M events, in barrel, crack, and endcap regions.

The cells contain four types of information:

- Energy (primary variable)
- Noise estimate (average)
- Gain in readout
- Time of cell energy

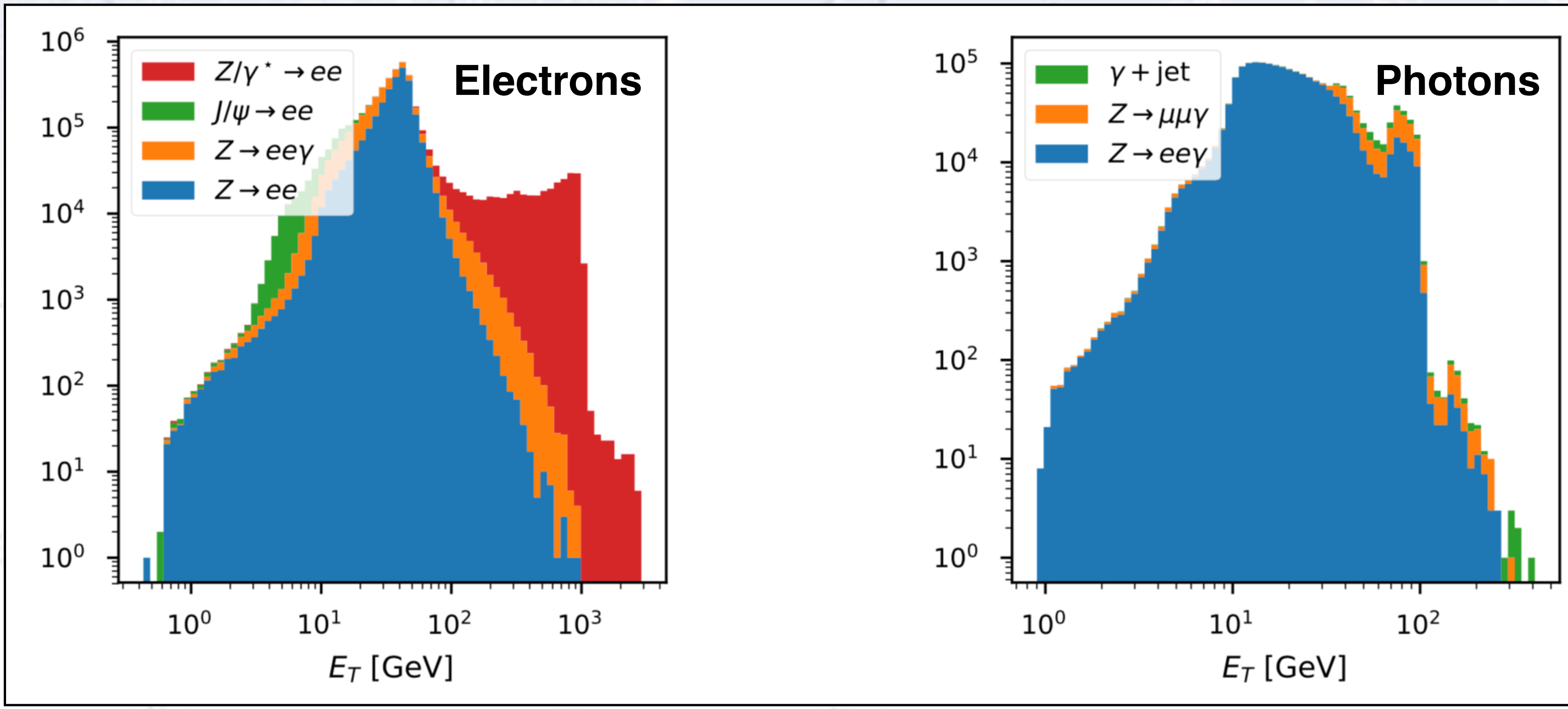
We investigated these somewhat, and found smaller improvements possible.

However, only energy is considered in the following, as time consumption and complexity increased significantly.

ECAL	E_{acc}	
	f_0	Fraction of energy in different ECAL layers
	R_{12}	
	E_{TG3}	
Context	η	
	η_{index}	
	$\langle\mu\rangle$	
	$n_{vertexReco}$	
	$\eta_{modECAL}$	Relative position within cell
	$poscs_2$	
	$\Delta\phi_{TH3}$	
Tracking	$\Delta\phi_2^{rescaled}$	Difference in tracking-ECAL position estimate
	$\Delta\eta_2$	
	p_T^{track}	
	n_{tracks}	

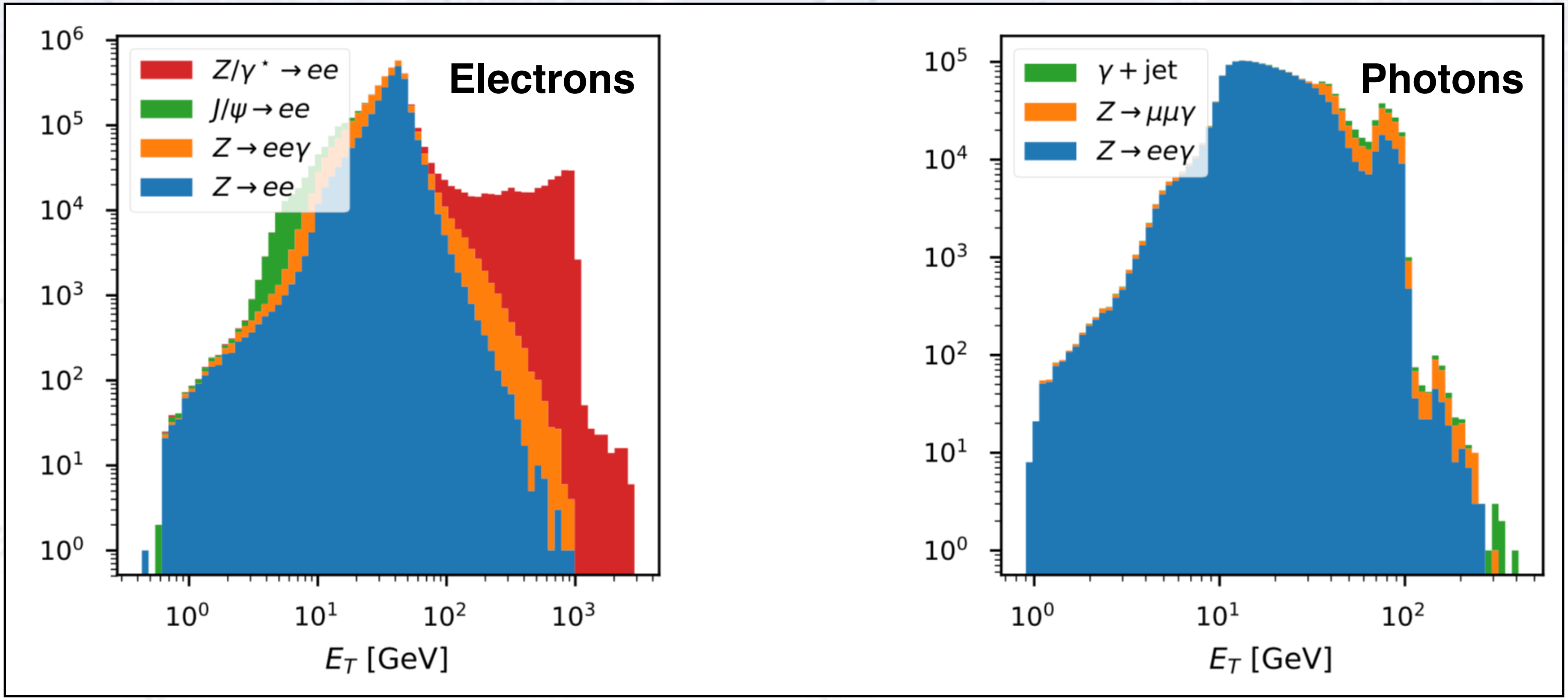
The data

The E_T distribution for electrons (left) and photons (right) are dominated by the range around the Z-peak range, in particular for the electrons. For photons, no high energy (> 100 GeV) data was available.



The data

The E_T distribution for electrons (left) and photons (right) are dominated by the range around the Z-peak range, in particular for the electrons. For photons, no high energy (> 100 GeV) data was available.



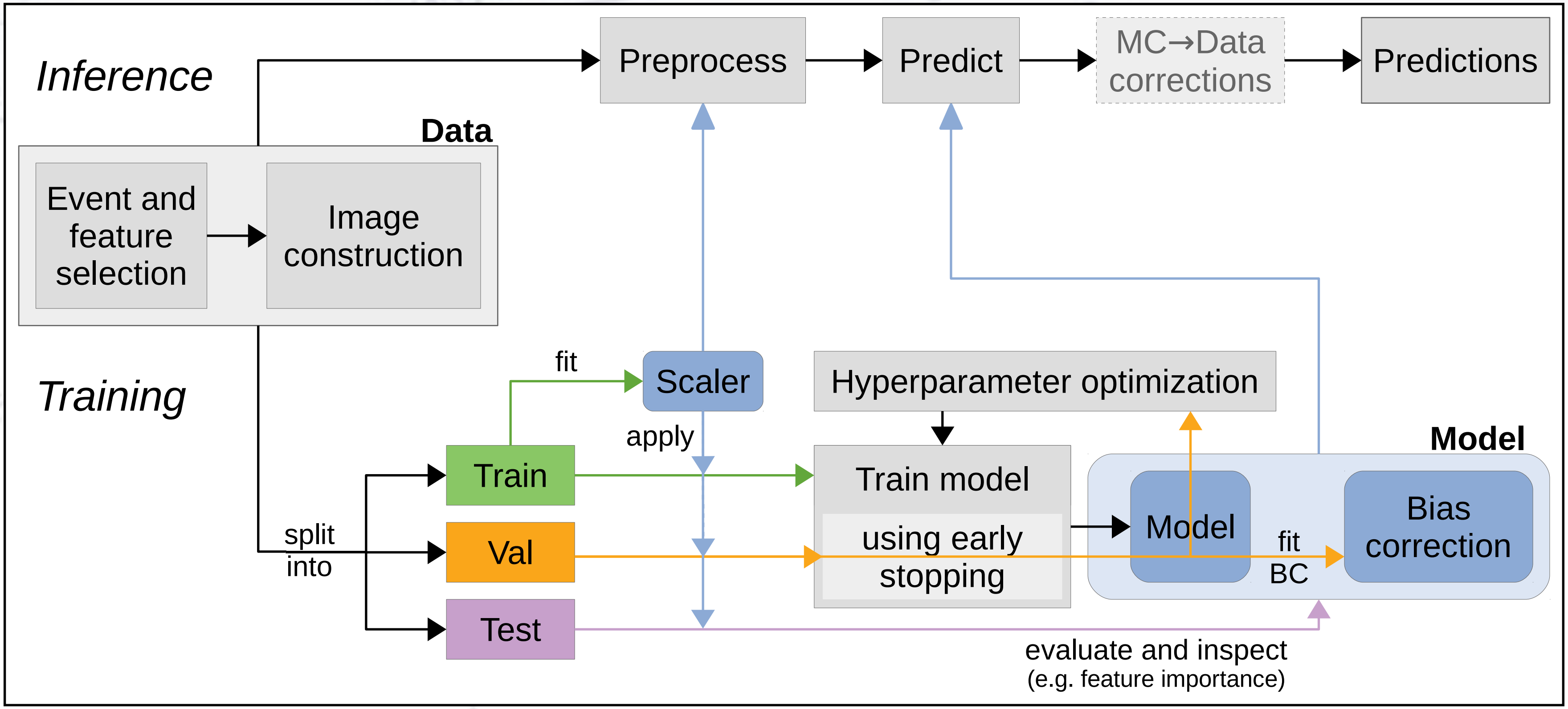
Note: The data production was done by Lukas Ehrke (as a master student, now Ph.D. student at Univ. of Geneva). It took **considerable effort and time**, and should in the future be done centrally by the EGamma (or H$\gamma\gamma$?) group.

Pipeline of analysis

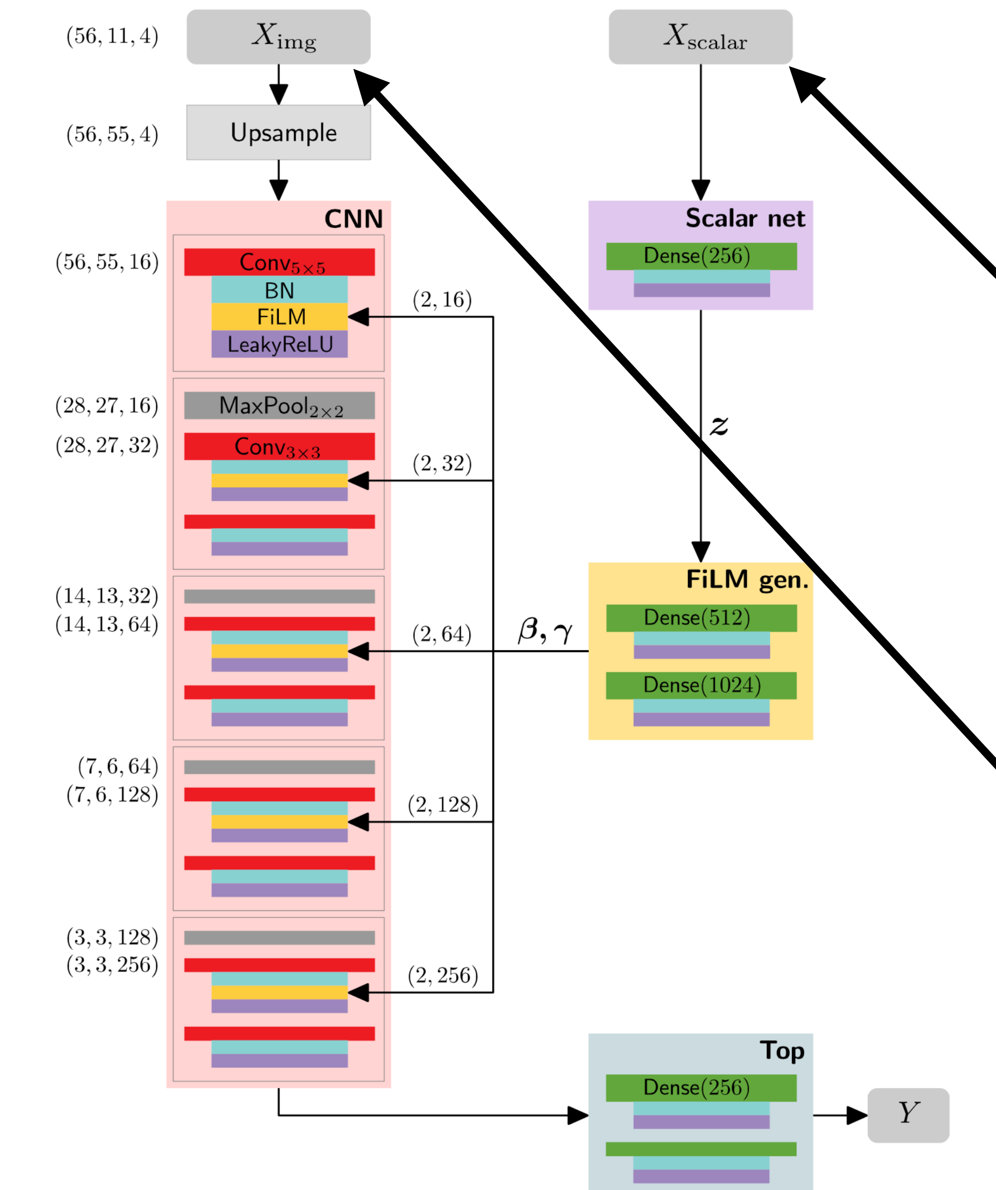
In order to stream line the analysis, Frederik set up the following pipeline:

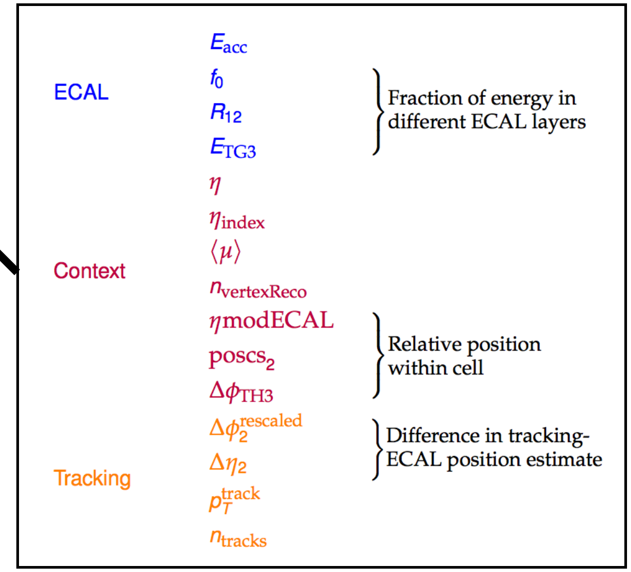
1. First, the scalars are fitted with a NN.
2. Next, a hyperparameter (HP) optimisation is done.
3. Next, the actual model is trained with the optimal HP (using early stopping).
4. Finally, a bias correction is applied (due to the use of medians).

This was done on 1-8 GPUs, depending on availability (limiting factor!).



Setup of CNN



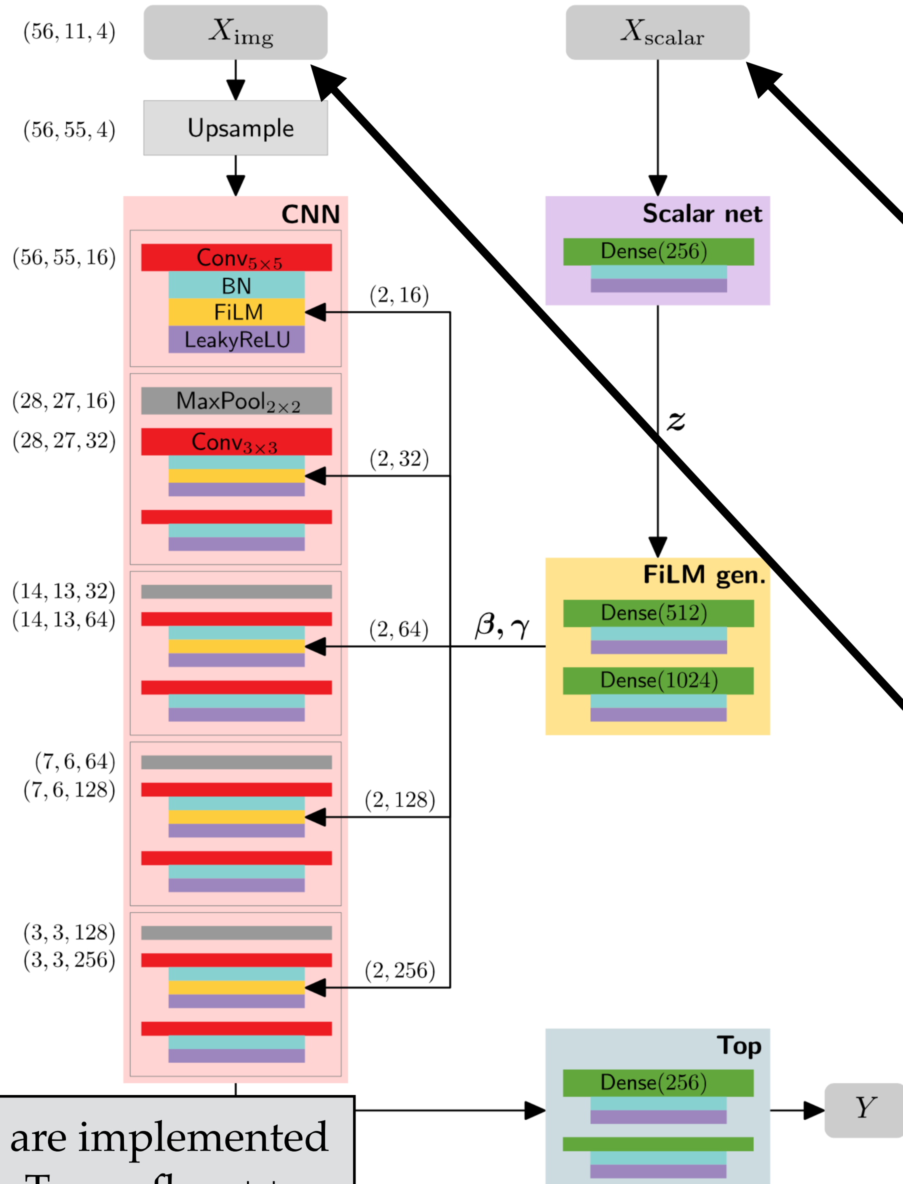


$$\mathrm{FiLM}(x_i) = \gamma_i(\boldsymbol{z})x_i + \beta_i(\boldsymbol{z})$$

(FiLM: Feature wIse Linear Modulation of the CNN output layers based on the scalar input variables.)

The cell images are upscaled to the common format **56x55**. For each candidate we have four images and the 14 scalar variable above (incl. crack scintillators).

Setup of CNN



ECAL	E_{acc}, f_0, R_{12}, E_{TG3}	Fraction of energy in different ECAL layers
Context	η, η_{index}, $\langle\mu\rangle$, $n_{\text{vertexReco}}$	
	ηmodECAL, poscs_2, $\Delta\phi_{\text{TH3}}$	Relative position within cell
Tracking	$\Delta\phi_2^{\text{rescaled}}$, $\Delta\eta_2$	Difference in tracking-ECAL position estimate
	p_T^{track}, n_{tracks}	

$$\text{FiLM}(x_i) = \gamma_i(\boldsymbol{z})x_i + \beta_i(\boldsymbol{z})$$

(FiLM: Feature wIse Linear Modulation of the CNN output layers based on the scalar input variables.)

The cell images are upscaled to the common format **56x55**. For each candidate we have four images and the 14 scalar variable above (incl. crack scintillators).

Models are implemented in Keras Tensorflow++

Metrics used

The metrics used were all based on **Relative Error (RE)**:

$$RE = (E_{estimated} - E_{true}) \;/\; E_{true}$$

Given a distribution in RE, we considered the **InterQuartile Range**:

$$IQR(RE) = Q3\,(RE) - Q1\,(RE)$$

Finally, for comparison we use the **Relative Improvement**:

$$rIQR = 1 - IQR_{model} / IQR_{ATLAS\ (BDT)}$$

Metrics used

The metrics used were all based on **Relative Error (RE)**:

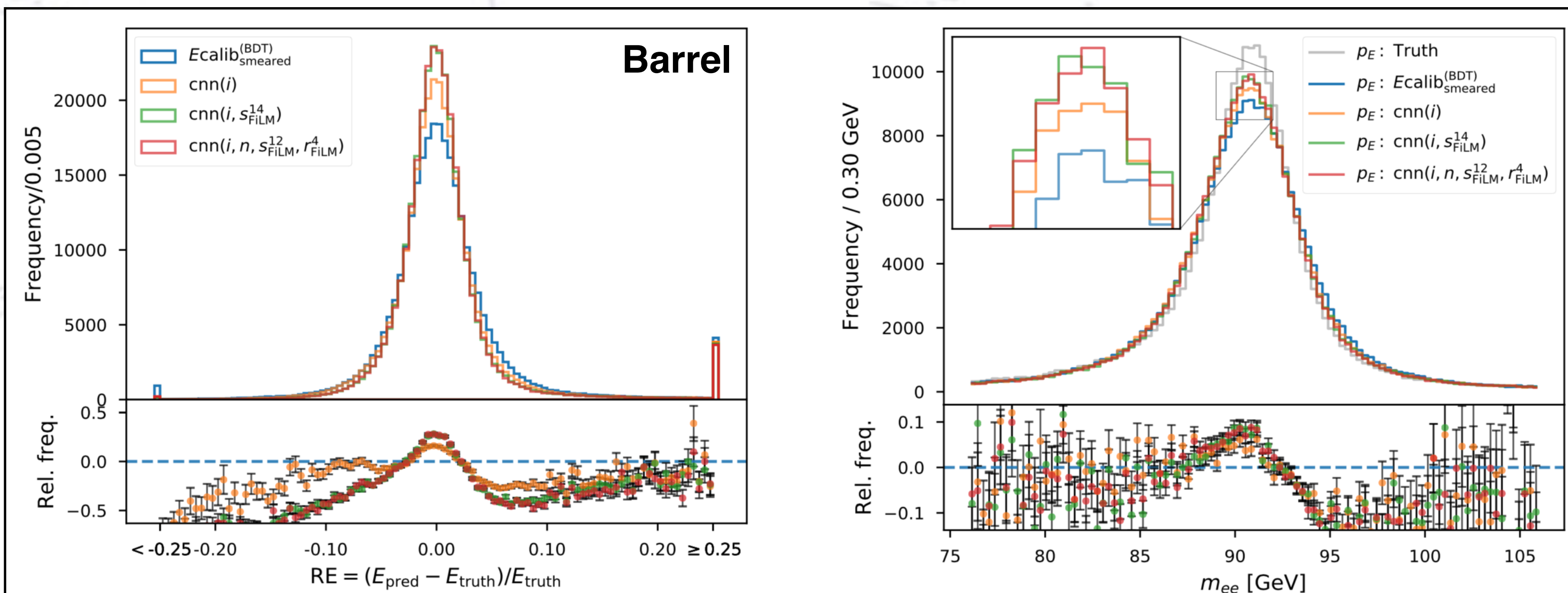
$$RE = (E_{estimated} - E_{true}) \;/\; E_{true}$$

Given a distribution in RE, we considered the **InterQuartile Range**:

$$IQR(RE) = Q3\,(RE) - Q1\,(RE)$$

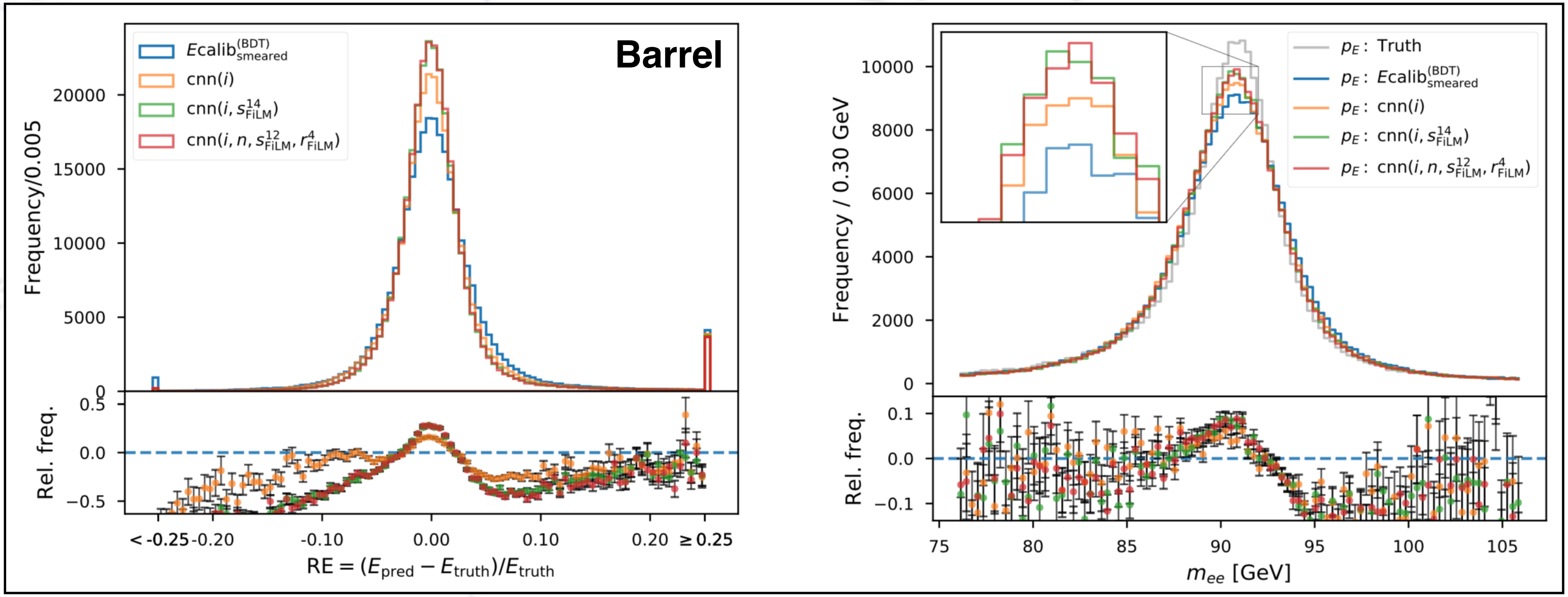
Finally, for comparison we use the **Relative Improvement**:

$$rIQR = 1 - IQR_{model} / IQR_{ATLAS\ (BDT)}$$



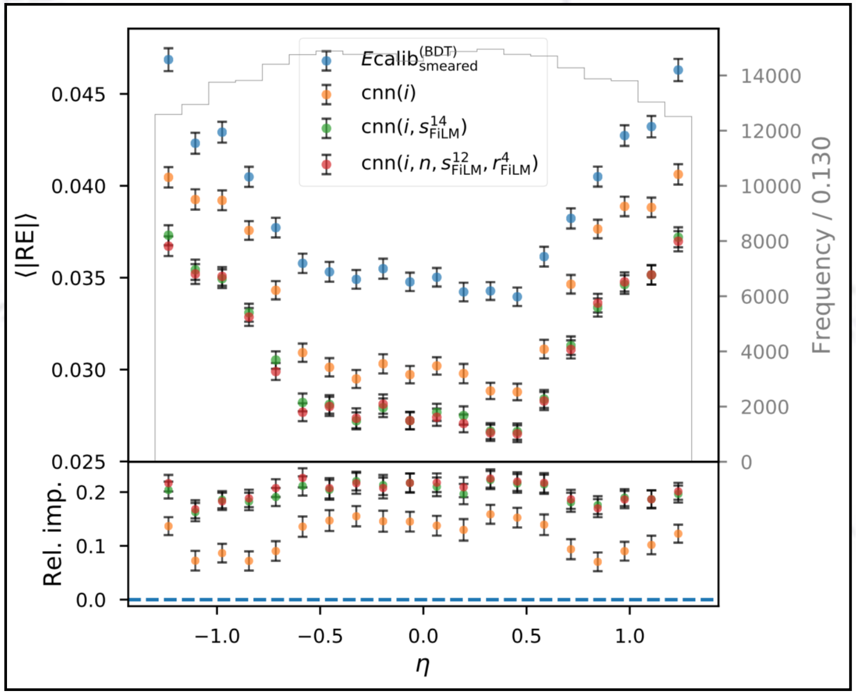
Results on electrons

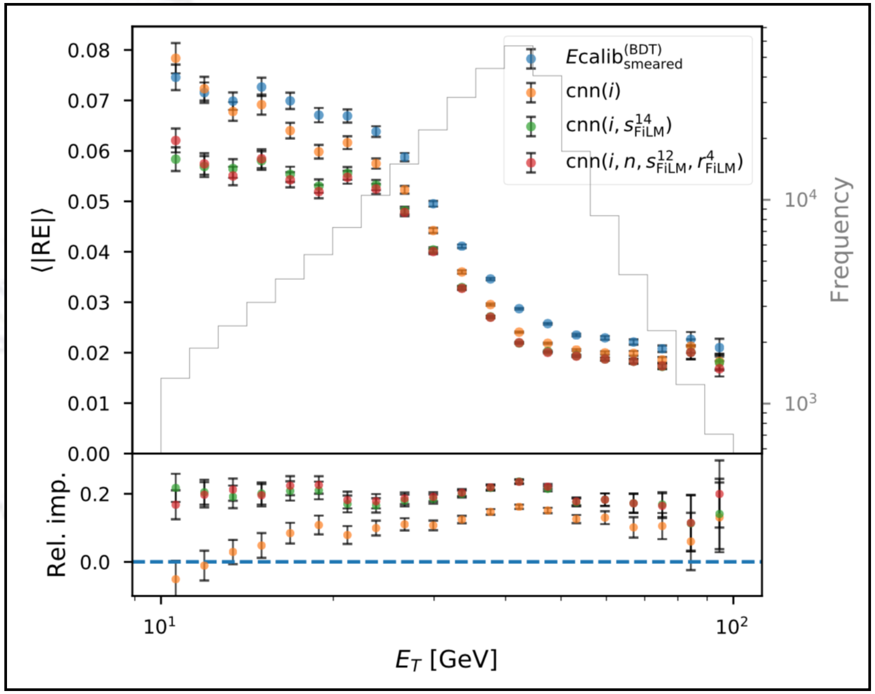
Model	$r\mathrm{IQR}^{\mathrm{corr.}}$	$\#p$ [M]
$E\mathrm{calib}^{(\mathrm{BDT})}_{\mathrm{smeared}}$	-2.0%	
$\mathrm{cnn}(i)$	11.9%	1.8
$\mathrm{cnn}(i, s^{14}_{\mathrm{FiLM}})$	20.5%	3.5
$\mathrm{cnn}(i, n, s^{12}_{\mathrm{FiLM}}, r^{4}_{\mathrm{FiLM}})$	21.7%	3.6



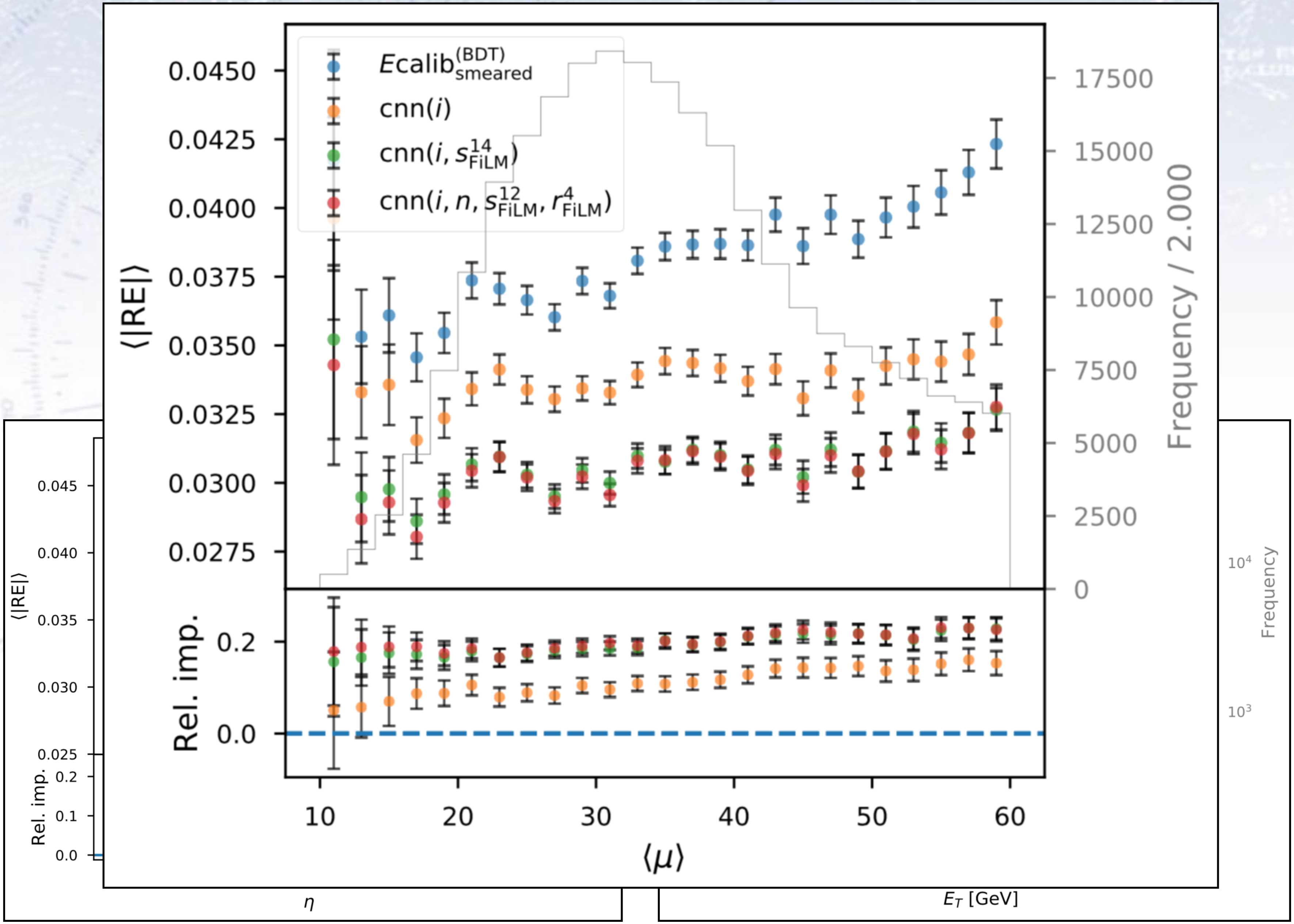
Results on electrons

Model	$r\mathrm{IQR}^{\mathrm{corr.}}$	$\#p$ [M]
$E\mathrm{calib}^{(\mathrm{BDT})}_{\mathrm{smeared}}$	-2.0%	
$\mathrm{cnn}(i)$	11.9%	1.8
$\mathrm{cnn}(i, s^{14}_{\mathrm{FiLM}})$	20.5%	3.5
$\mathrm{cnn}(i, n, s^{12}_{\mathrm{FiLM}}, r^{4}_{\mathrm{FiLM}})$	21.7%	3.6





Results on electrons



Results on photons

The corresponding results in photons can be seen below.

However, also notice the number of parameters in the models! And here we were still using three separate models for each part of the detector.

CNN(i) (19.9 M $\#p$)

Trained on:	Barrel	Crack	Endcap
$r\mathrm{IQR}^{(\mathrm{all})}$	23.6%	17.7%	27.5%
$r\mathrm{IQR}^{(\mathrm{unconv})}$	25.4%	20.7%	31.0%
$r\mathrm{IQR}^{(\mathrm{conv})}$	9.7%	11.1%	18.4%

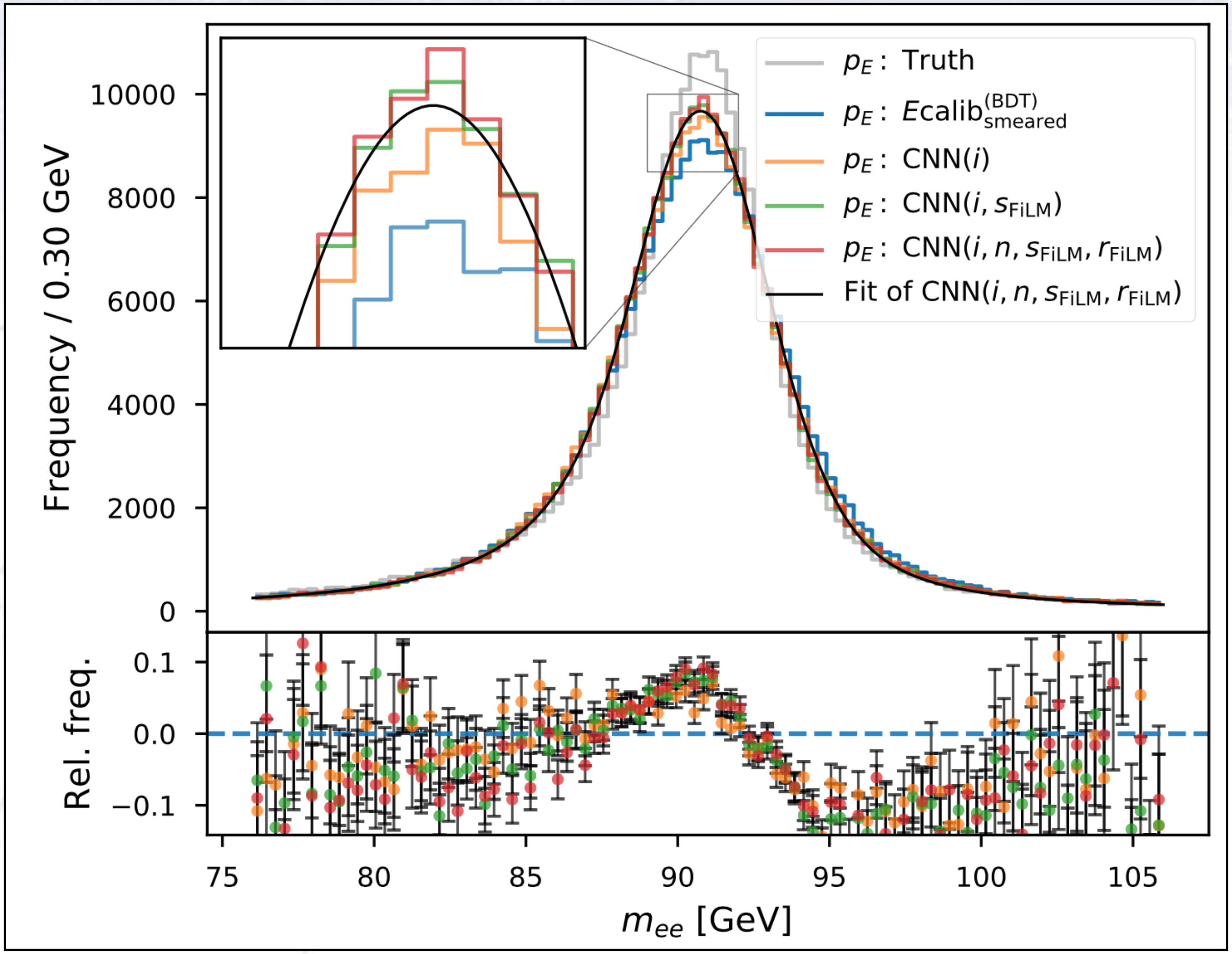
But what if we could build "one model to rule them all"?

Issue: How to deal with the crack?

Redoing it all…
…not dividing in eta

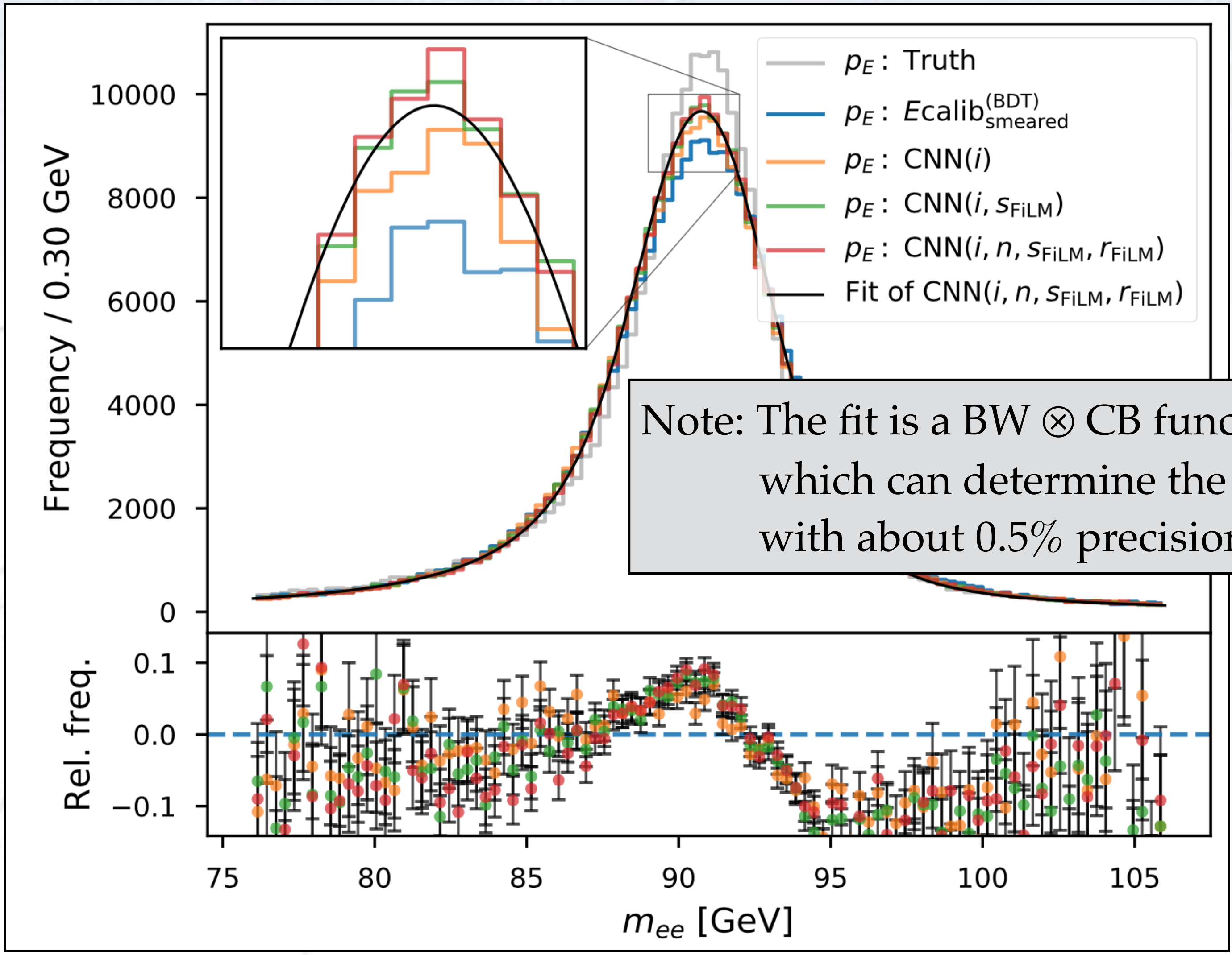
Results on electron - one model

Using a single model, Frederik managed to improve performance and reduce the model complexity (i.e. number of parameters).



Results on electron - one model

Using a single model, Frederik managed to improve performance and reduce the model complexity (i.e. number of parameters).

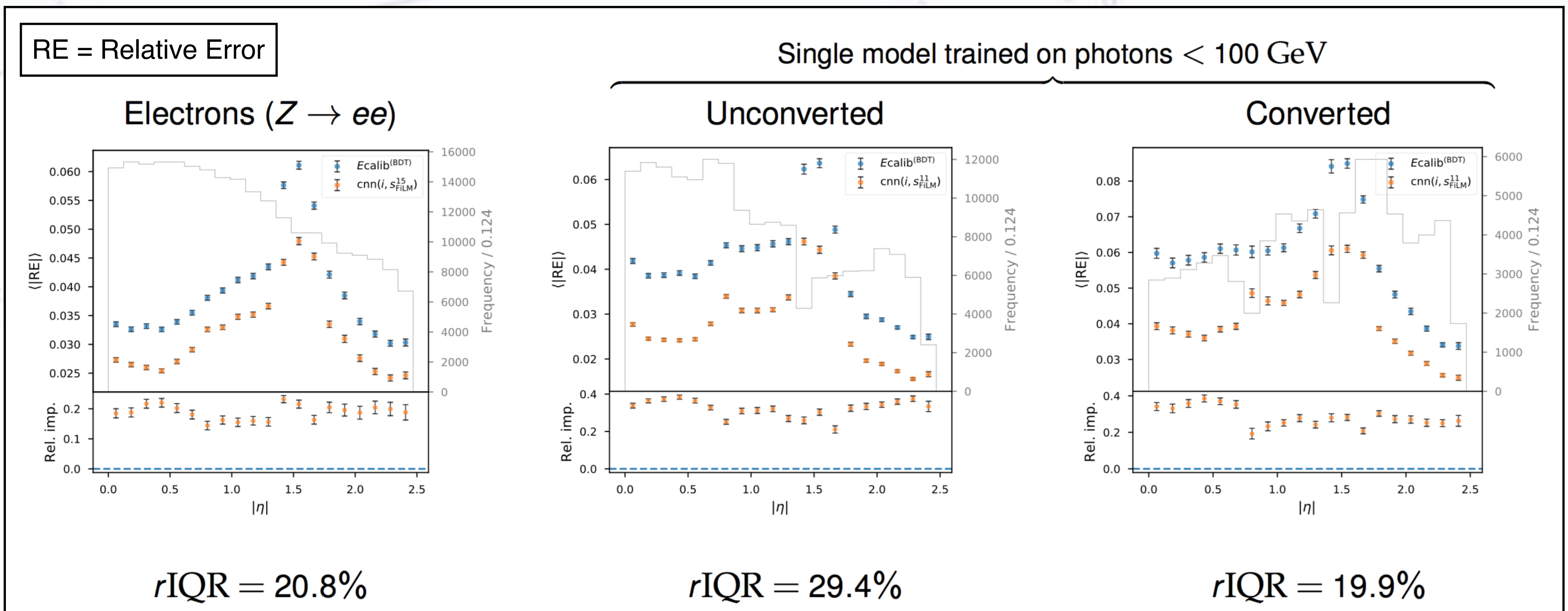


Note: The fit is a BW ⊗ CB function fit, which can determine the resolution with about 0.5% precision.

Results in general

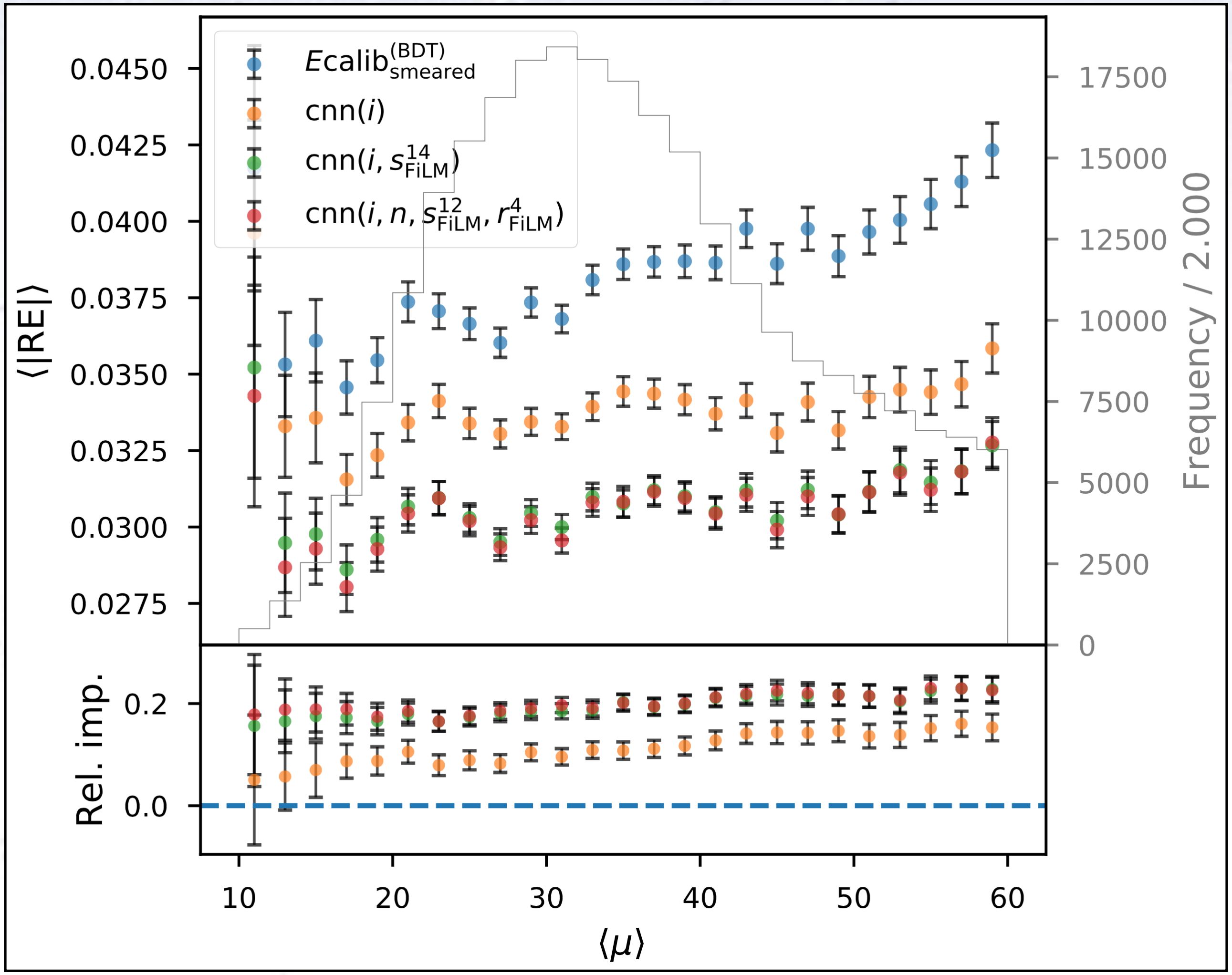
In simulation we find that we are able to improve the energy resolution (defined as ratio of Inter Quantile Range (rIQR)) by about **20% for electrons and 30%/20% for unconverted/converted photons**.

The improvement is reasonably uniform across η (see below) and also so in energy. It should be noted, that the model was trained on photons with energies below 100 GeV (i.e. not spanning the whole physics analysis range).



Performance vs. pile-up

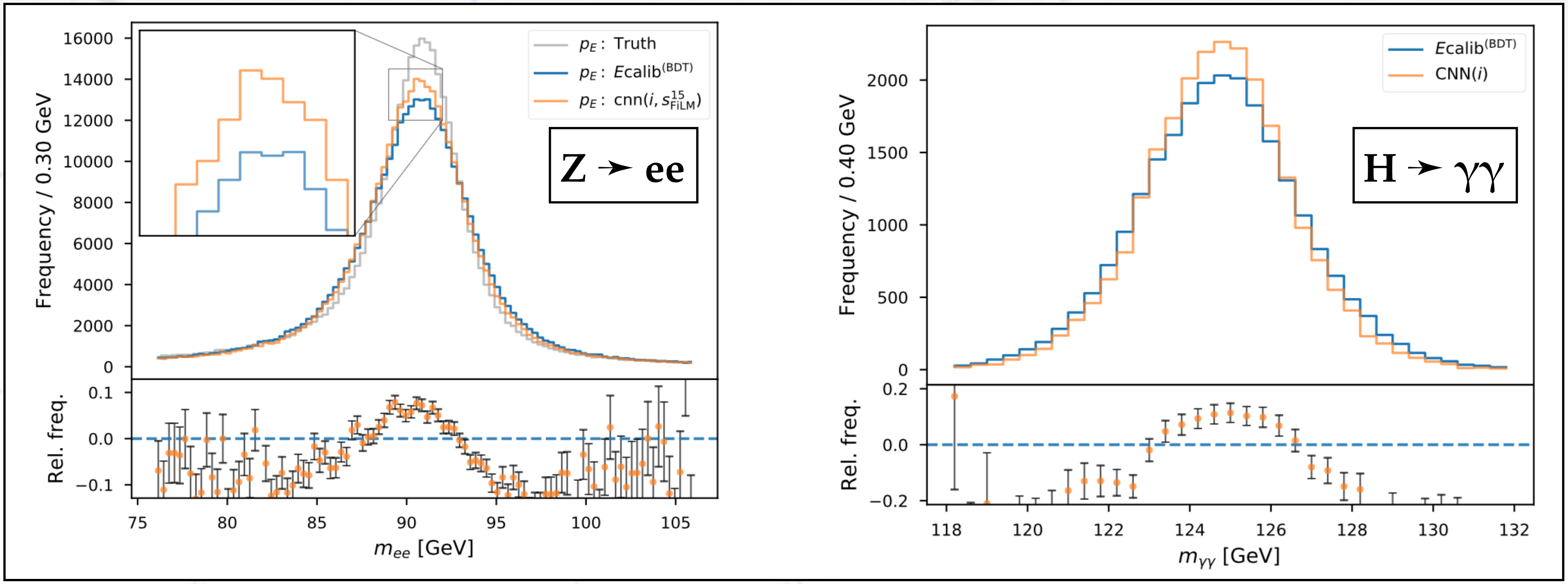
Shown in the figure is the resolution as a function of ⟨μ⟩. Relative improvement increases with pile-up, i.e. the CNN is more robust to pule-up.



Impact on physics channels

Using the CNN energy estimates on the Z ➛ ee and H ➛ γγ channels yields more narrow peaks, as one would expect.

In the first case, the CNN energy is only applied to the probe electron, and it can here be compared to what is obtained using the truth.



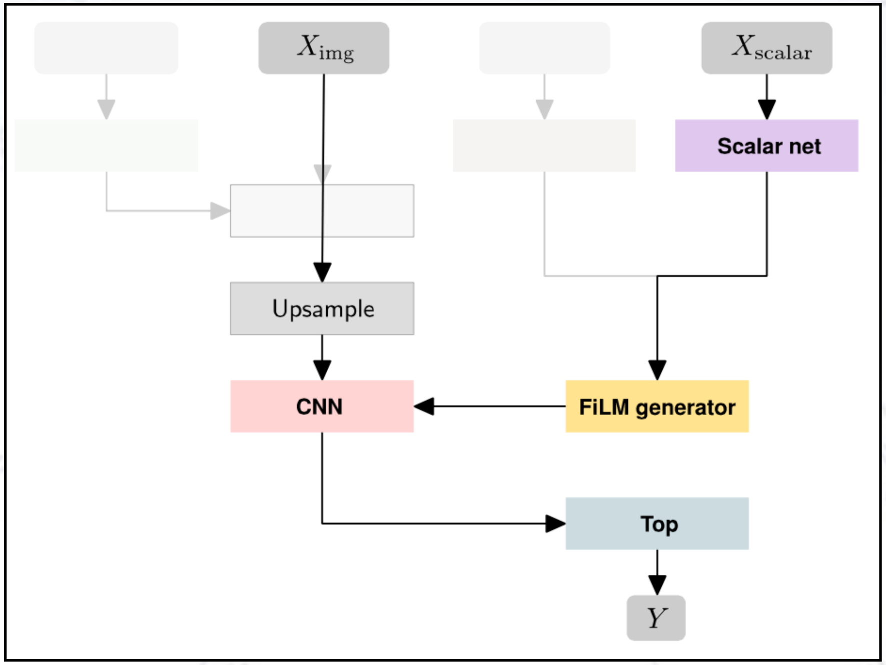
This performance improvement is (so far) only seen in MC, but the performance in data can be measured well in the Z ➛ ee channel (checked in MC).

More advanced model?

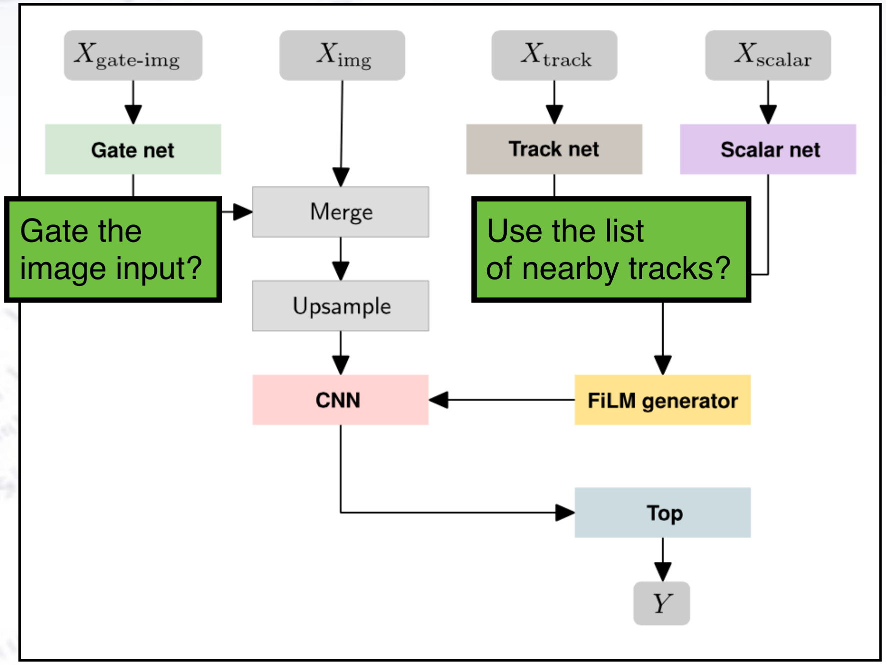
The results shown are obtained with the variables and network presented.

More input: Cells also holds information on time, noise, and gain. Tracks useful.
Larger architecture: More versatility allows network to use information better.

Current model:



Potential model:



However, implementation into Athena is a technical challenge, which should be factored in, when choosing final model.

Where to go from here?

The first thing to do is to:

- Produce and train on **cell re-weighted** MC samples.
- Apply to Z ➛ ee sample in data, and **extract data performance from fit**.

If this checks out, then getting these models into Athena is the next challenge.

This has turned out to be a challenge, which requires significant dedicated work. Several people (in particular Debo Gupta) are already working on it and progressing, but so far with no really good solution.

Maybe the solution would have to be to apply it “by hand” in the analysis.

But imagine… if we got a H ➛ $\gamma\gamma$ peak that was sharper than that of CMS :-)

Bonus Slides

Performance vs. pile-up

Shown in the figure is the resolution as a function of ⟨μ⟩. Relative improvement increases with pile-up, i.e. the CNN is more robust to pule-up.