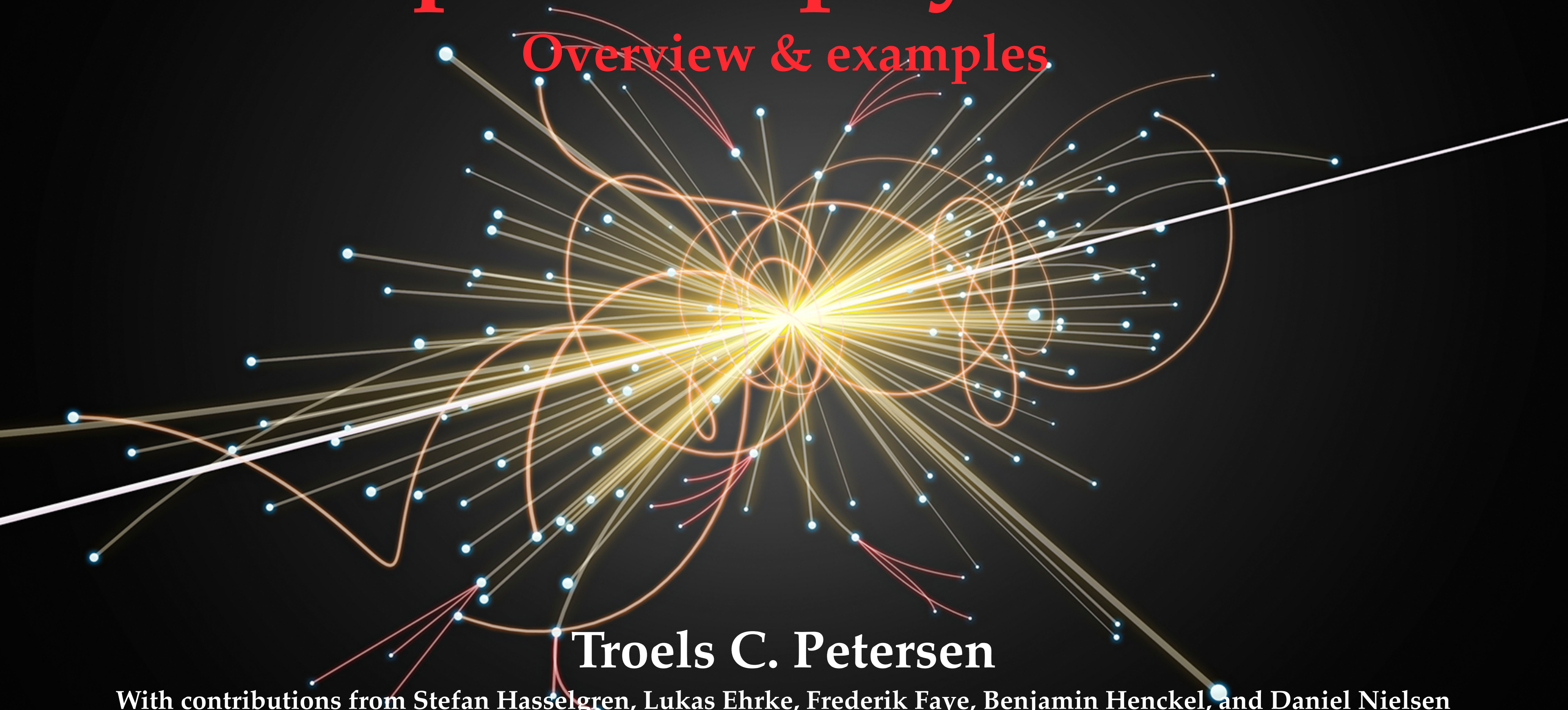
Machine Learning

meets

particle physics

Overview & examples



Troels C. Petersen

With contributions from Stefan Hasselgren, Lukas Ehrke, Frederik Faye, Benjamin Henckel, and Daniel Nielsen

Niels Bohr Institute

AI center seminar, 3rd of May 2019

Machine Learning
meets
particle physics
Overview & examples

Troels C. Petersen

With contributions from Stefan Hasselgren, Lukas Ehrke, Frederik Faye, Benjamin Henckel, and Daniel Nielsen

Niels Bohr Institute

AI center seminar, 3rd of May 2019

Outline

Outline of talk:

- In the beginning: **b-jet tagging in ALEPH**
- HEP data and why it is exciting for Machine Learning (ML)
- Going large scale: **Electrons and photons in ATLAS**
  - Data samples, variables, and selections
  - Electron PID and Energy Reconstruction (ER)
  - Discussion of performance measures (loss functions)
- Looking at the future: **ν-reconstruction in IceCube**

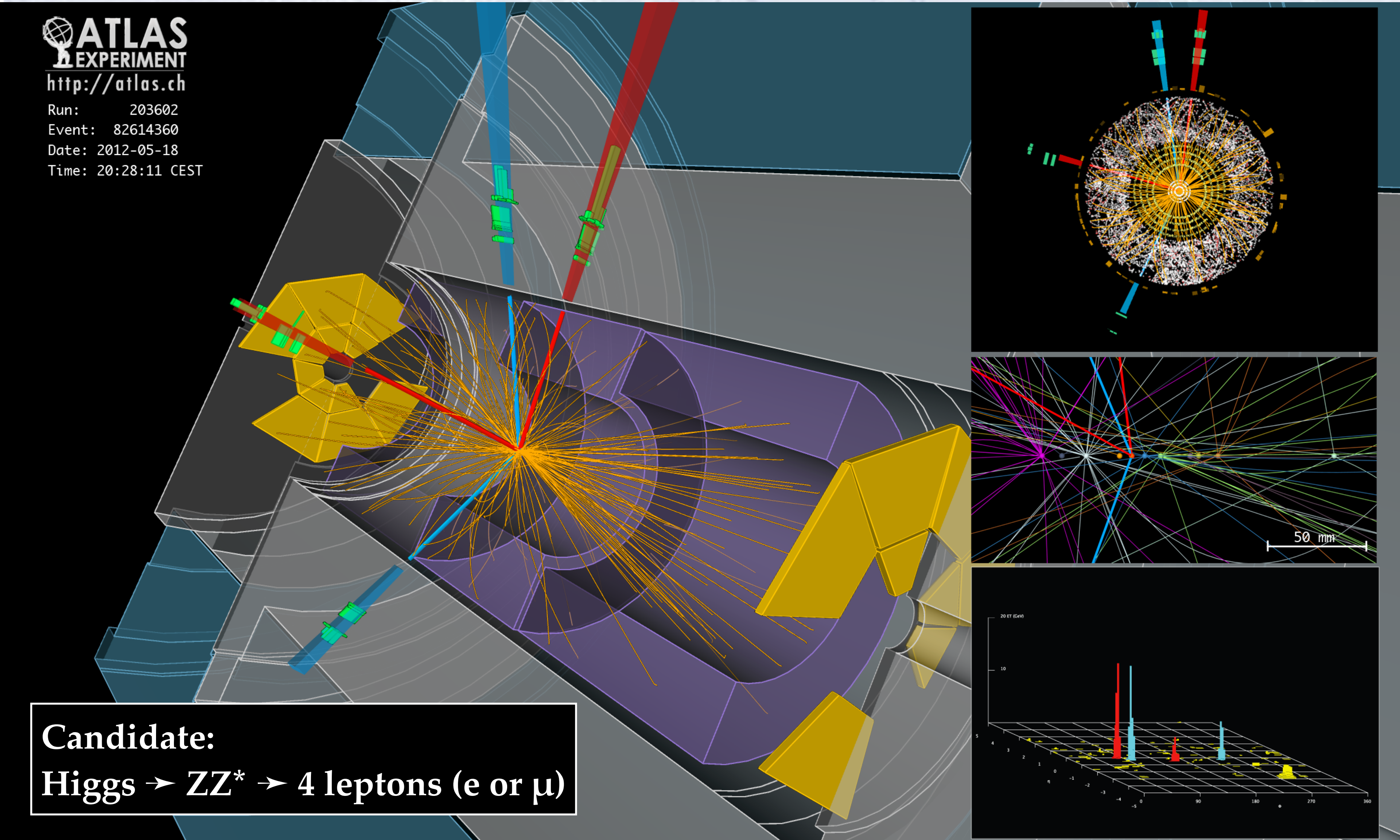
Purpose of talk:

- Show Machine Learning cases in science.
- Present HEP data, and why it is great (but also cumbersome!)
- Open up for possible inspiration/collaboration

In the following, all numbers and plots are "Not Even Preliminary", and should in not be used elsewhere.

A word about particle physics

We search for MANY different things, typically rare (1:10^9) with complex decays.

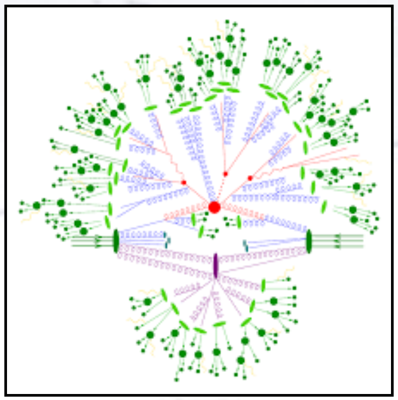


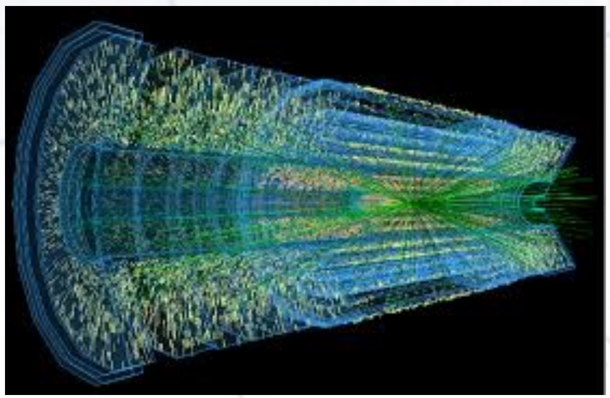
Candidate:
Higgs ➛ ZZ* ➛ 4 leptons (e or μ)

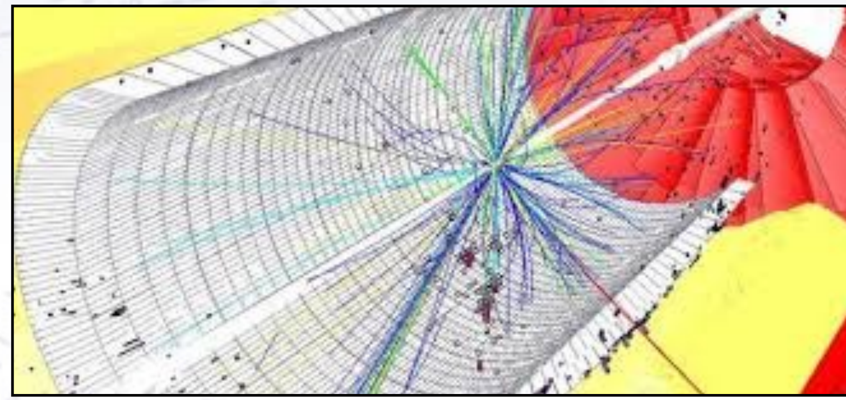
Particle physics data and simulation

To make sure that we understand our experiment we use simulation extensively:

- Detector optimisation (before experiment)
- Reconstruction design/optimisation (before/during experiment)
- Selection optimisation (during experiment)
- Signal efficiency estimates (during experiment)







The simulation is done in three steps: Event generation, Simulation, and Digitisation.

The total CPU time needed for one event is about 20-30 minutes, and we have now simulated about a billion events (using 0.5M cores).

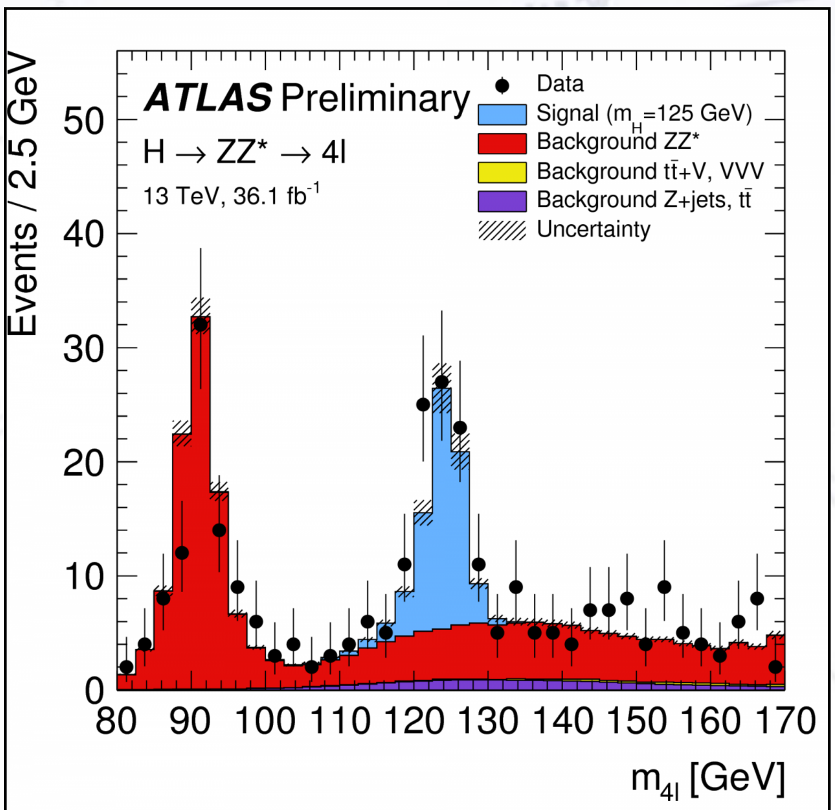
The simulation is done from **first principles**, and there are therefore (smaller) **differences between simulation and data** (maybe a point of interest to fix?).

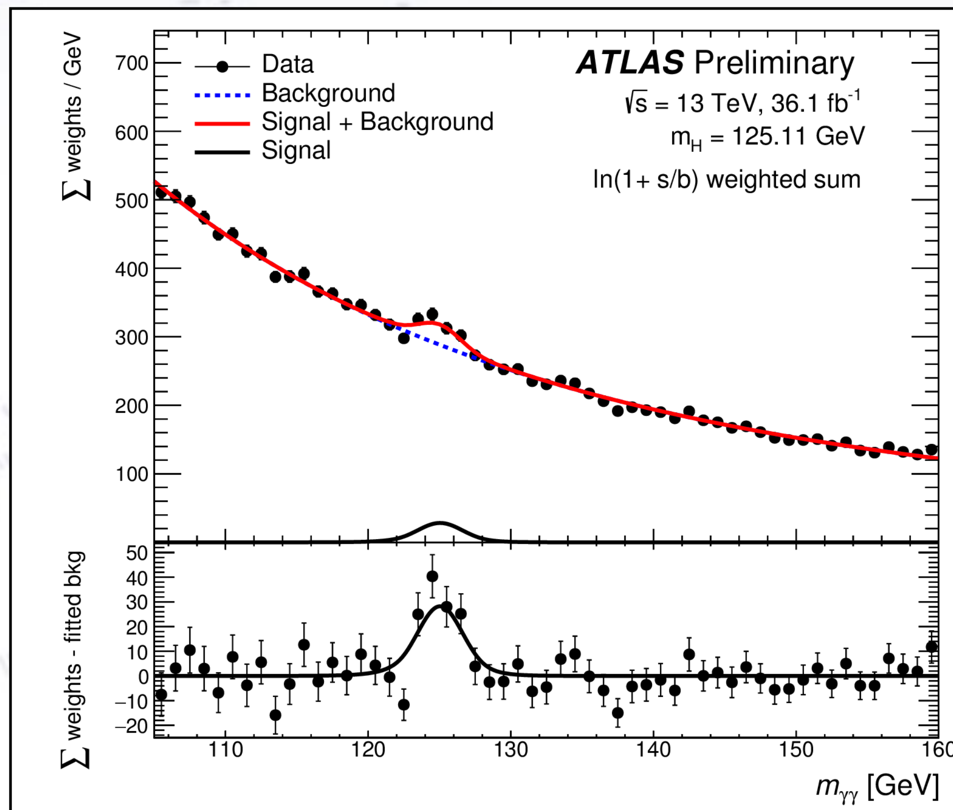
Aim of this project

Electrons and photons play a central role in the ATLAS physics programme, in particular in **Higgs physics**, where they dominate the two golden channels.

Current methods use likelihood approach (PID) and simple ML (Energy).

Given the cost of running, we would be satisfied, if we could add 10% to each of these in terms of statistics, knowing this would also benefit many other analysis.





In the beginning: b-jet tagging in ALEPH

ALEPH b-jet tagging

25 years ago, particle physics was actually at the forefront of using Machine Learning. We had large computers and much data fit for ML usage.

At the time, LEP was searching for the Higgs boson at lower masses, where its decay was almost always to b-quarks.

For this reason, many resources were used to get the best possible b-jet tagging in place.

ALEPH b-jet tagging

25 years ago, particle physics was actually at the forefront of using Machine Learning. We had large computers and much data fit for ML usage.

At the time, LEP was searching for the Higgs boson at lower masses, where its decay was almost always to b-quarks.

For this reason, many resources were used to get the best possible b-jet tagging in place.

Both lifetime (displaced vertices), jet shape, and lepton pT was used, but non of these by themselves provide a good way to select b-jets.

Impact parameter significance:

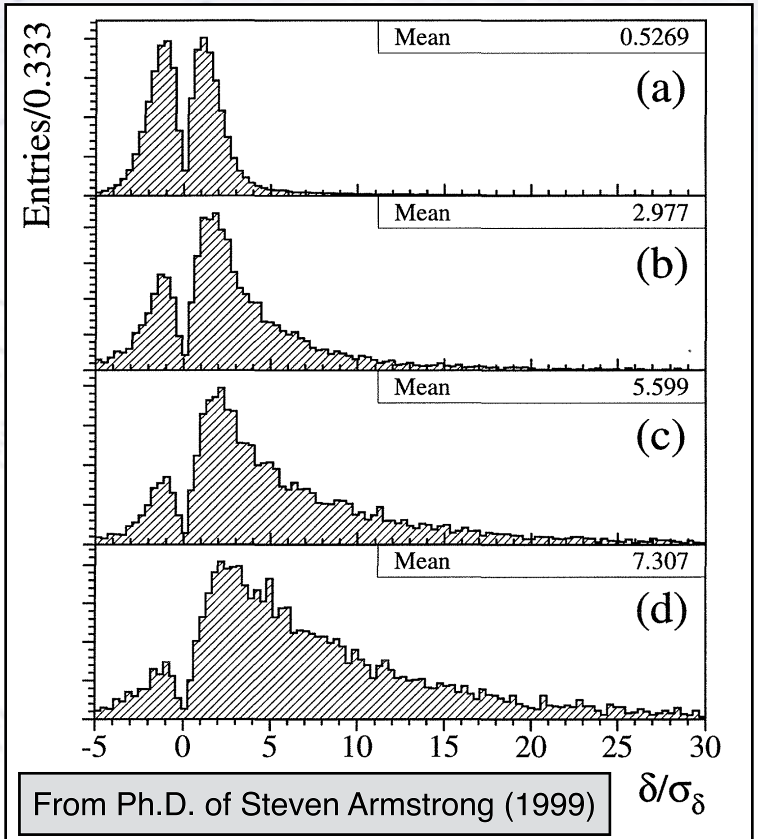


Figure 3.5: Impact parameter significance (δ/σ_δ) distributions for (a) fragmentation or tracks from uds events, (b) tracks from weakly decaying c hadrons, (c) tracks directly from b hadron decay, (d) tracks from the cascade decay of a c hadron from the decay of a b hadron.

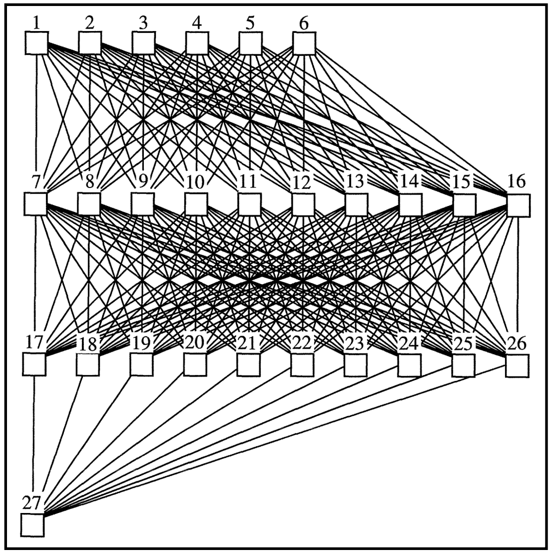
ALEPH b-jet tagging

However, using one of the very first ML algorithms (JetNet 3.4), six variables were put together in a neural network with two hidden layers each with 10 neurons:

- Light quark (uds) jet probability from track impact parameter significance.
- Difference in Chi2 from search for secondary vertices in jet.
- Transverse momentum of (possible) electron/muon in jet.
- Boosted sphericity of jet.
- Energy flow multiplicity (scaled by jet energy).
- Sum of transverse momenta (with respect to the jet axis) squared.

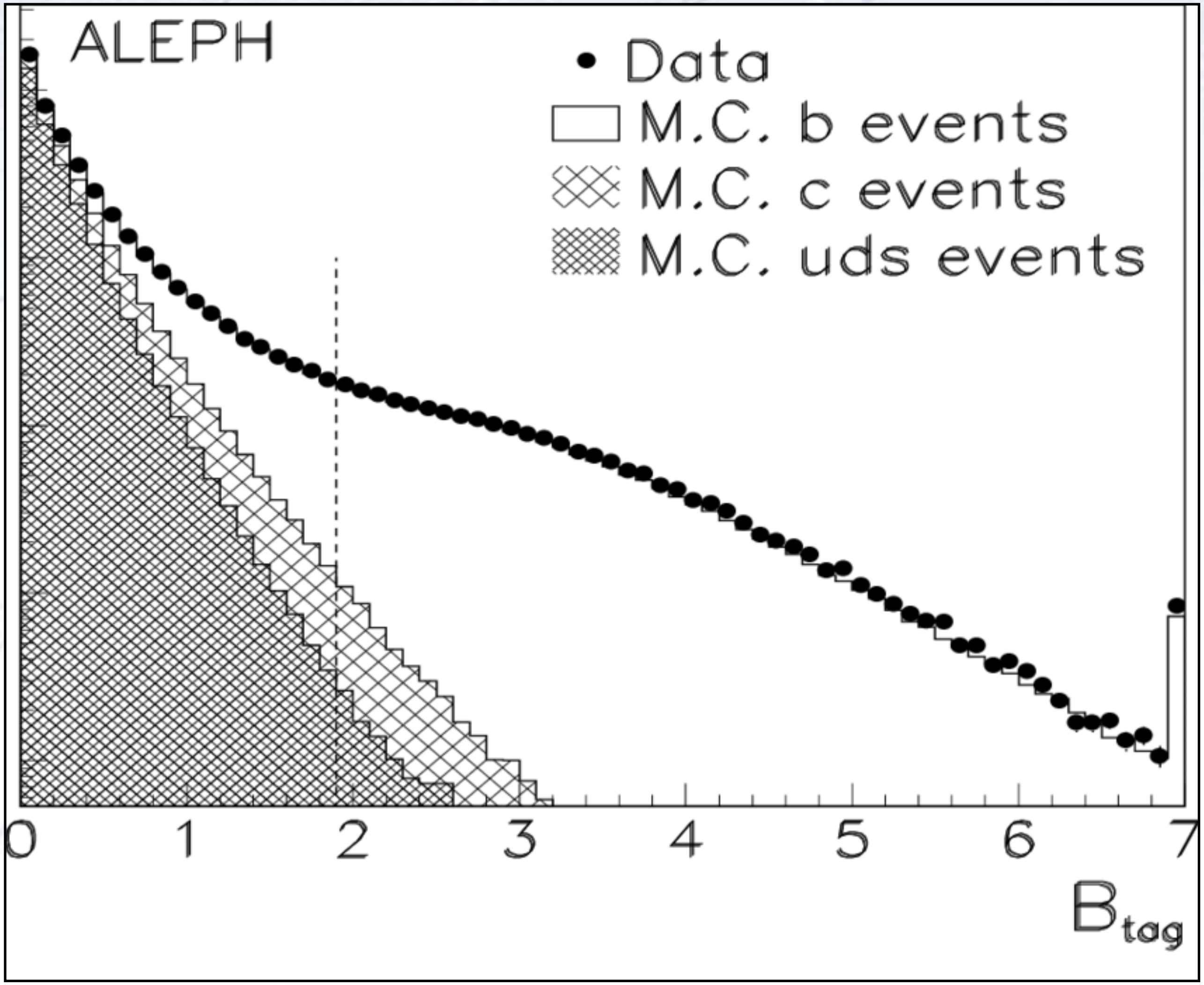
The neural network was trained on 400.000 **simulated** events, and though I haven't been able to find the exact time used for this training, colleagues have told me "many hours, sometime days".

Interestingly, my students now code the setup in about an hour, and get results in minutes.



ALEPH b-jet tagging

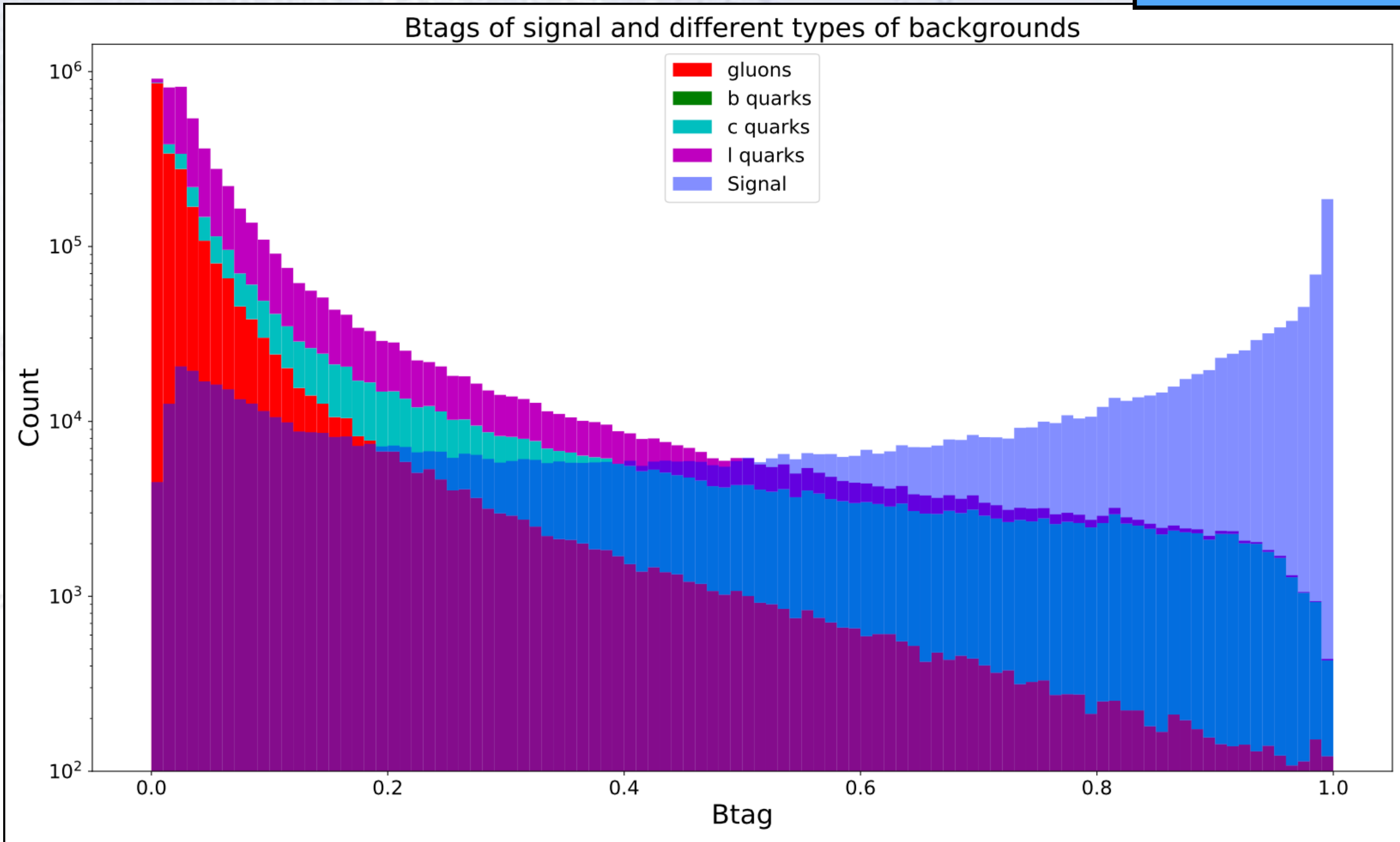
The result of these labours was a very nice b-jet tagging variable, which allowed ALEPH to get the most out of their data.



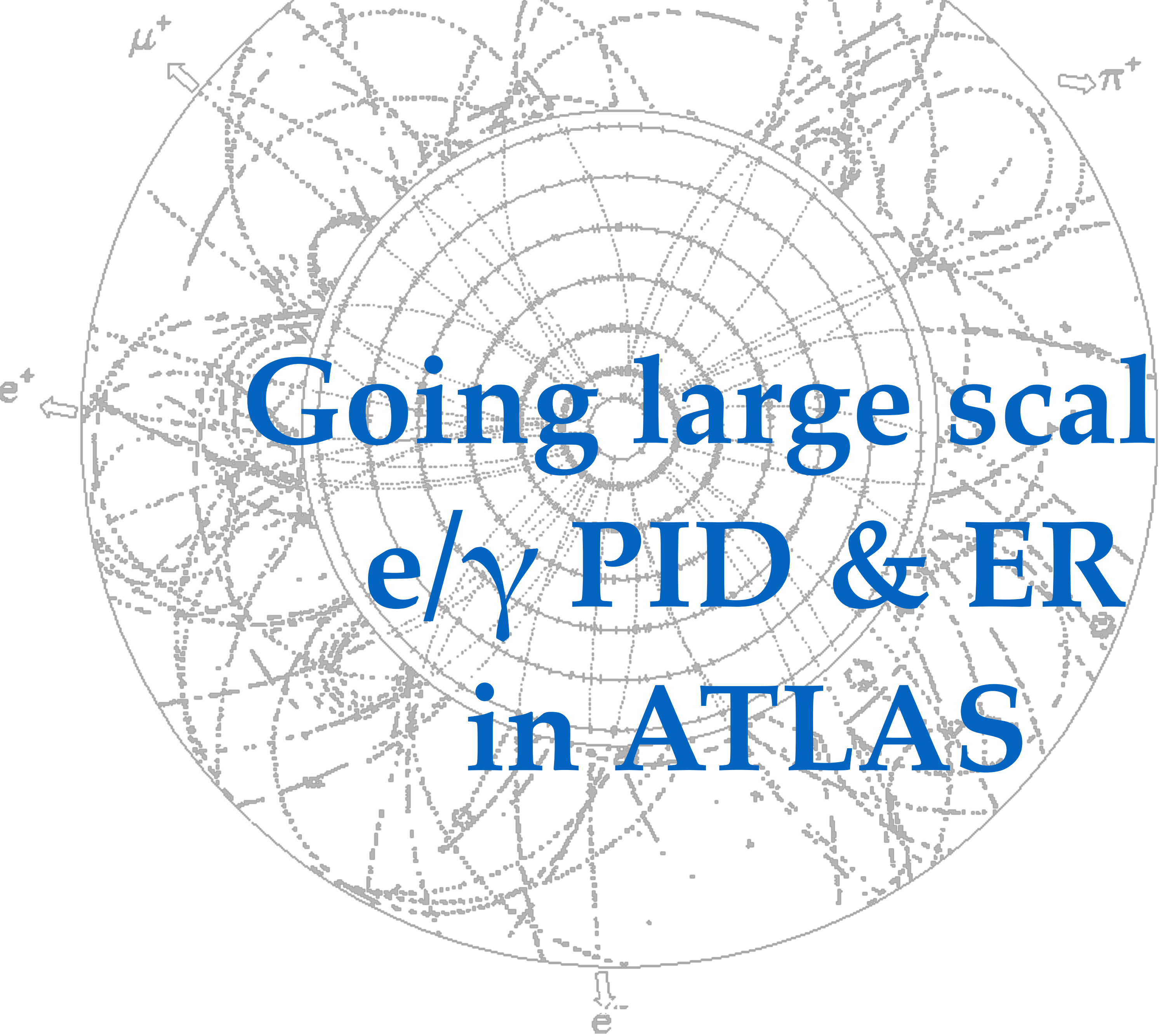
ALEPH b-jet tagging

A more “modern” plot could look like this:

Benjamin Henckel



Going large scale: e/γ PID & ER in ATLAS



Overview

Currently in ATLAS, electrons and photons are identified using a likelihood approach:

$$d_{\mathcal{L}} = \frac{\mathcal{L}_S}{\mathcal{L}_S + \mathcal{L}_B}, \qquad \mathcal{L}_S(\vec{x}) = \prod_{i=1}^{n} P_{s,i}(x_i)$$

The likelihood is composed of 22 variables, for which 1D histograms are used to compute the likelihood value.

To minimise correlations, the likelihood is divided into regions of E_T and η.

This makes for a very transparent approach, which at the same time performs well.

The question is, if there is more information to be gained, and thus a more powerful PID to be gotten.

Enter Machine Learning (ML)…

Type	Description	Name
Hadronic leakage	Ratio of E_T in the first layer of the hadronic calorimeter to E_T of the EM cluster (used over the range $\|\eta\| < 0.8$ or $\|\eta\| > 1.37$)	R_{Had1}
	Ratio of E_T in the hadronic calorimeter to E_T of the EM cluster (used over the range $0.8 < \|\eta\| < 1.37$)	R_{Had}
Back layer of EM calorimeter	Ratio of the energy in the back layer to the total energy in the EM accordion calorimeter	f_3
Middle layer of EM calorimeter	Lateral shower width, $\sqrt{(\Sigma E_i \eta_i^2)/(\Sigma E_i) - ((\Sigma E_i \eta_i)/(\Sigma E_i))^2}$, where E_i is the energy and η_i is the pseudorapidity of cell i and the sum is calculated within a window of 3×5 cells	$W_{\eta 2}$
	Ratio of the energy in 3×3 cells over the energy in 3×7 cells centered at the electron cluster position	R_ϕ
	Ratio of the energy in 3×7 cells over the energy in 7×7 cells centered at the electron cluster position	R_η
Strip layer of EM calorimeter	Shower width, $\sqrt{(\Sigma E_i (i - i_{\max})^2)/(\Sigma E_i)}$, where i runs over all strips in a window of $\Delta\eta \times \Delta\phi \approx 0.0625 \times 0.2$, corresponding typically to 20 strips in η, and $i_{\max}$ is the index of the highest-energy strip	w_{stot}
	Ratio of the energy difference between the largest and second largest energy deposits in the cluster over the sum of these energies	E_{ratio}
	Ratio of the energy in the strip layer to the total energy in the EM accordion calorimeter	f_1
Track quality	Number of hits in the B-layer (discriminates against photon conversions)	n_{Blayer}
	Number of hits in the pixel detector	n_{Pixel}
	Number of total hits in the pixel and SCT detectors	n_{Si}
	Transverse impact parameter	d_0
	Significance of transverse impact parameter defined as the ratio of d_0 and its uncertainty	σ_{d_0}
	Momentum lost by the track between the perigee and the last measurement point divided by the original momentum	$\Delta p/p$
TRT	Total number of hits in the TRT	n_{TRT}
	Ratio of the number of high-threshold hits to the total number of hits in the TRT	F_{HT}
Track-cluster matching	$\Delta\eta$ between the cluster position in the strip layer and the extrapolated track	$\Delta\eta_1$
	$\Delta\phi$ between the cluster position in the middle layer and the extrapolated track	$\Delta\phi_2$
	Defined as $\Delta\phi_2$, but the track momentum is rescaled to the cluster energy before extrapolating the track to the middle layer of the calorimeter	$\Delta\phi_{\mathrm{res}}$
	Ratio of the cluster energy to the track momentum	E/p
Conversions	Veto electron candidates matched to reconstructed photon conversions	isConv

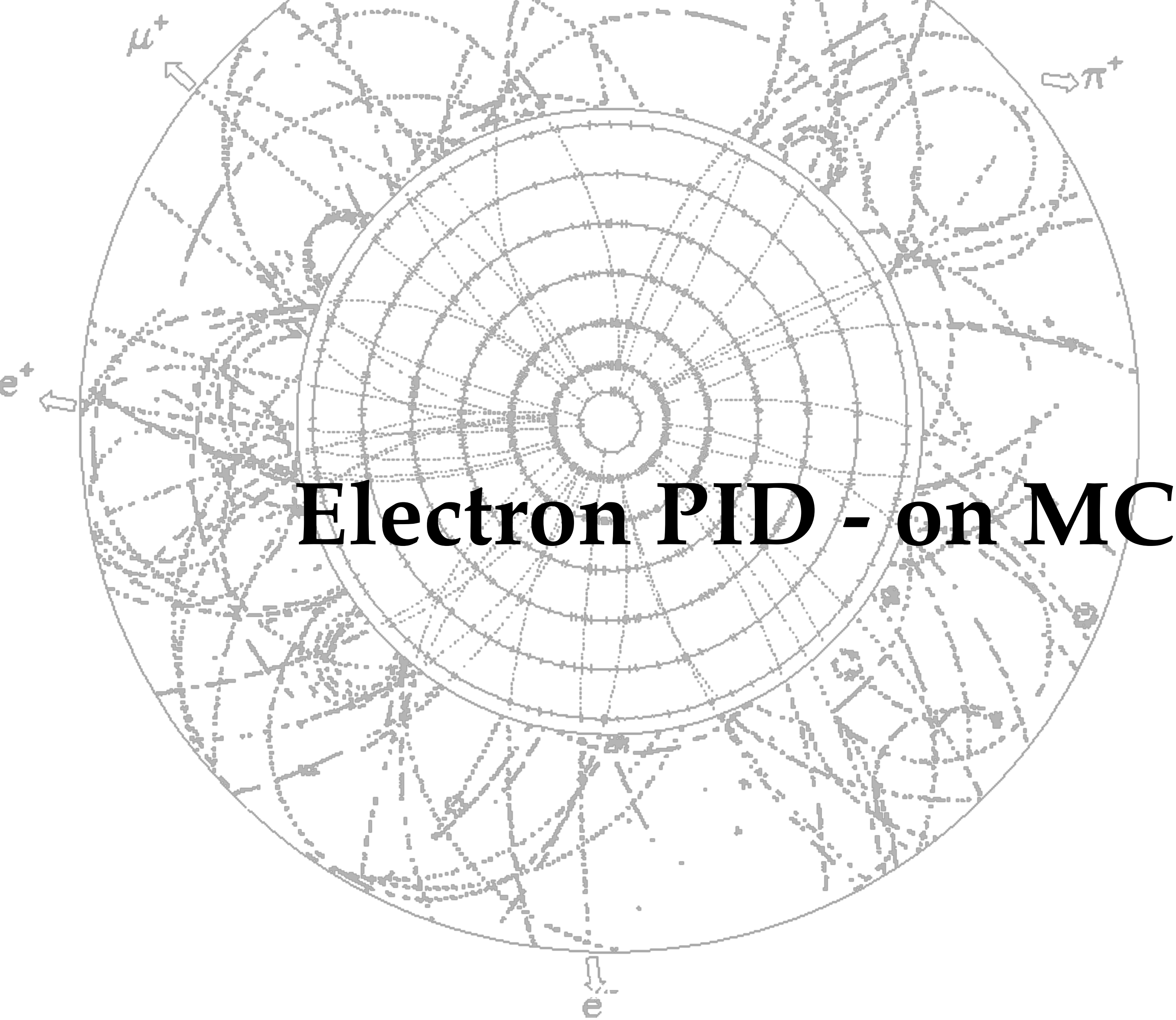
9 calorimeter variables

8 tracking variables

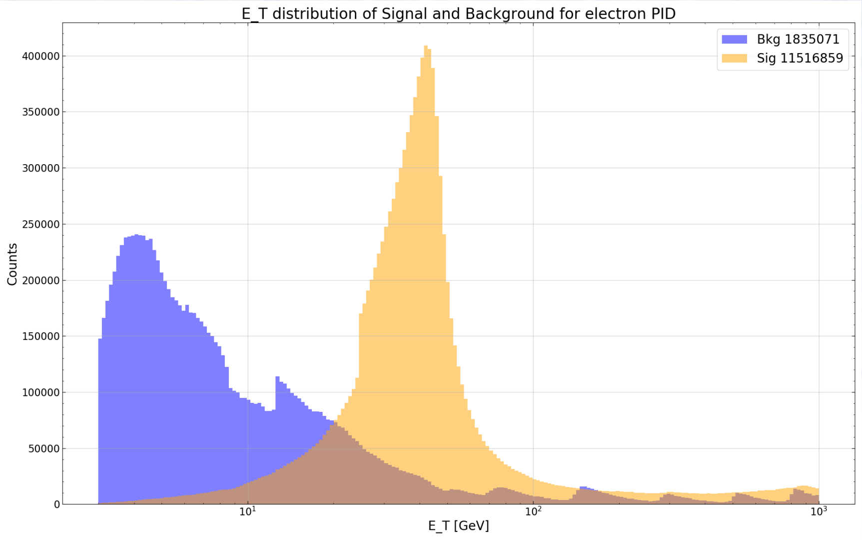
4 matching variables

1 conversion variable (for photons)

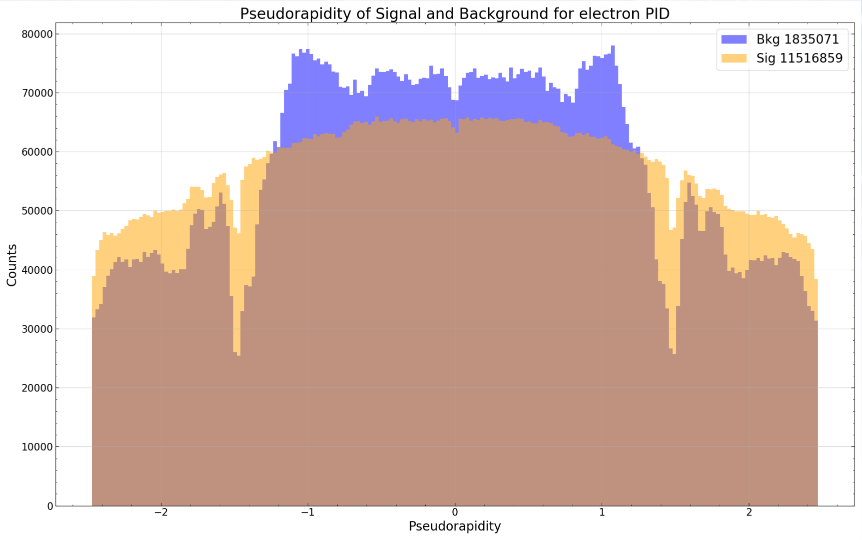
Electron PID - on MC



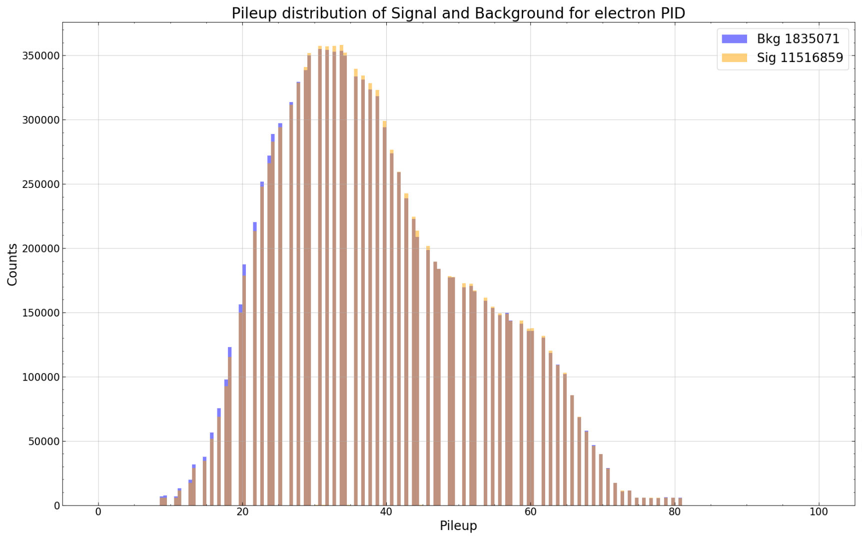
Reweighing



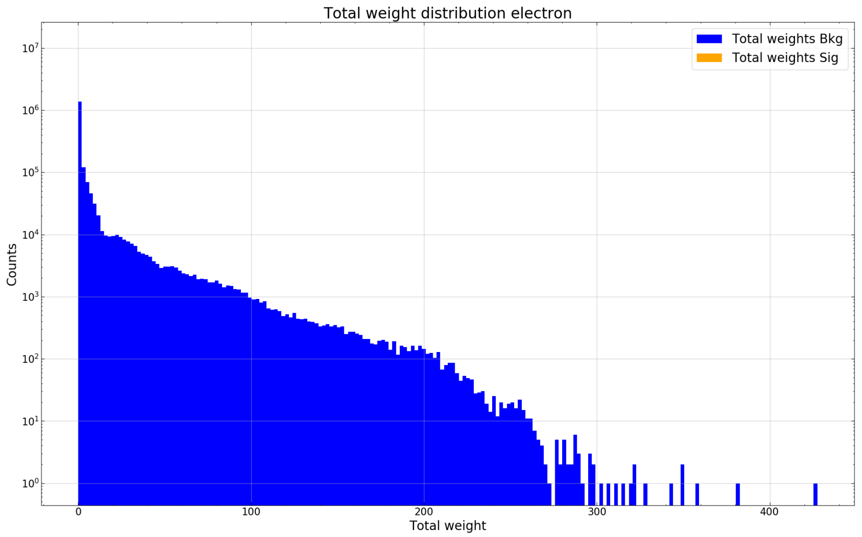
E_T distribution of Signal and Background for electron PID
Bkg 1835071
Sig 11516859
Counts
E_T [GeV]



Pseudorapidity of Signal and Background for electron PID
Bkg 1835071
Sig 11516859
Counts
Pseudorapidity

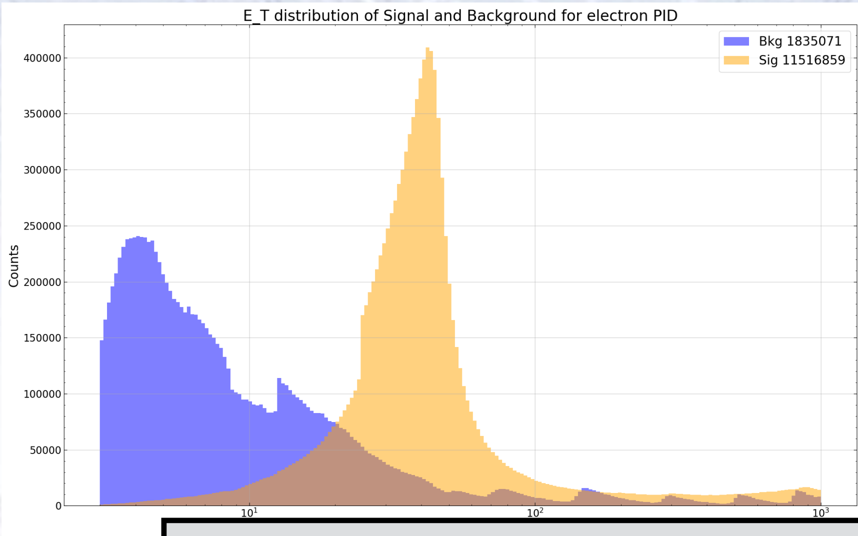


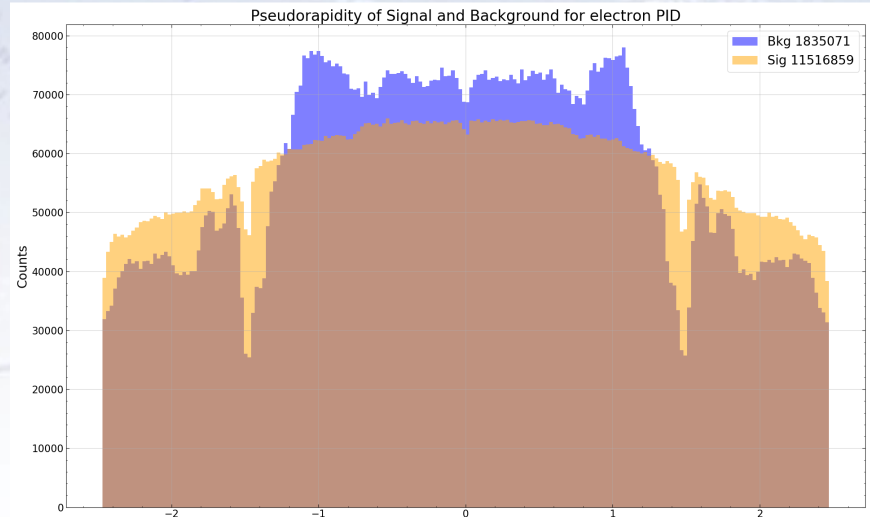
Pileup distribution of Signal and Background for electron PID
Bkg 1835071
Sig 11516859
Counts
Pileup



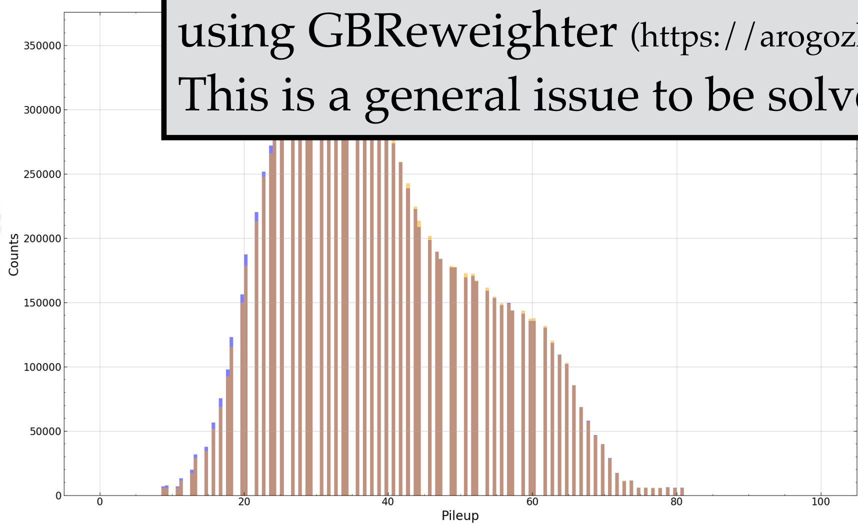
Total weight distribution electron
Total weights Bkg
Total weights Sig
Counts
Total weight

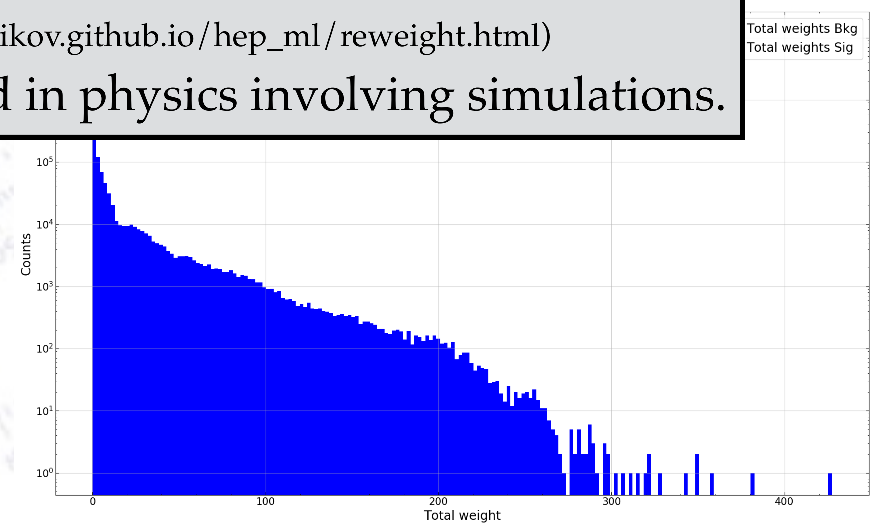
Reweighing





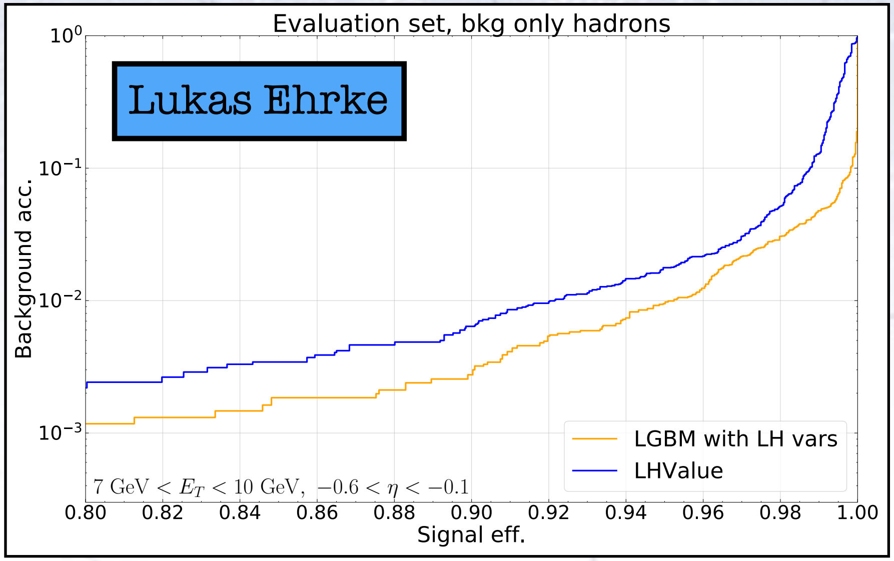
The background is reweighed to look like signal in E_T, η and $\langle\mu\rangle$
using GBReweighter (https://arogozhnikov.github.io/hep_ml/reweight.html)
This is a general issue to be solved in physics involving simulations.



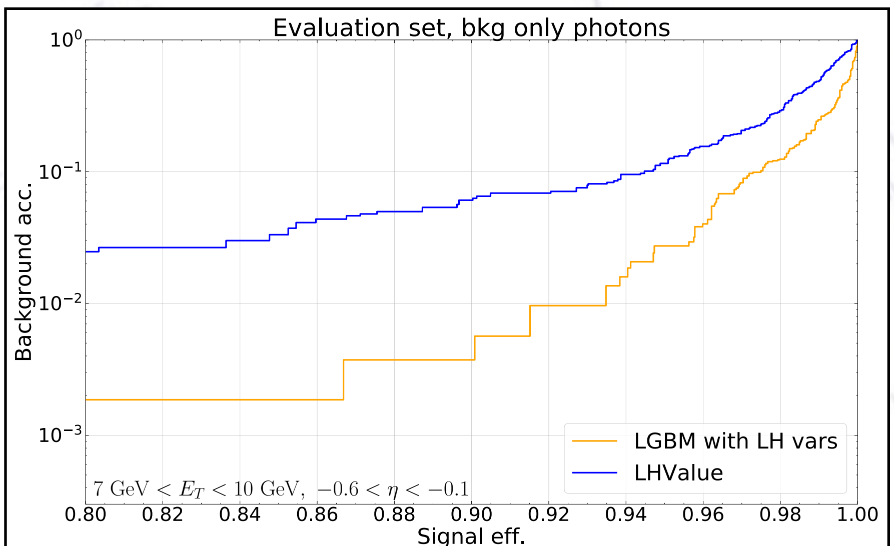


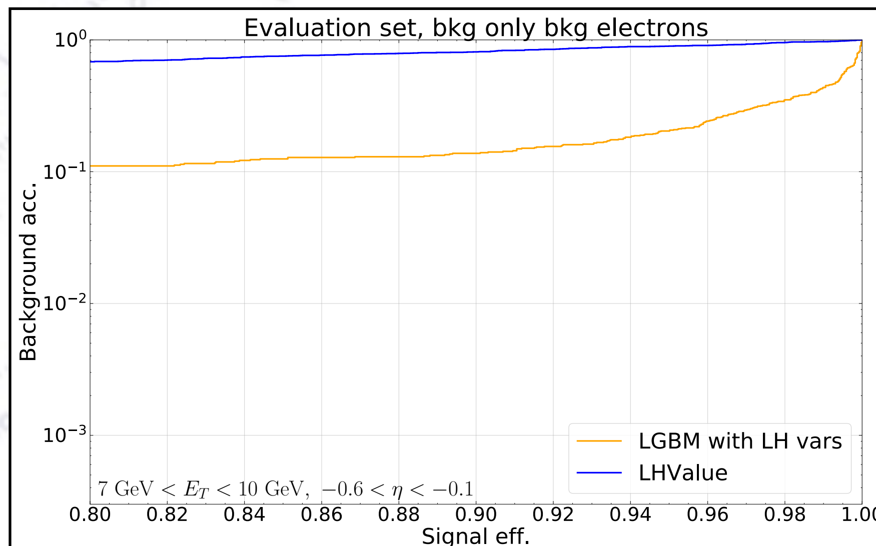
Electron PID performance

The electron PID performance is generally much improved with ML:



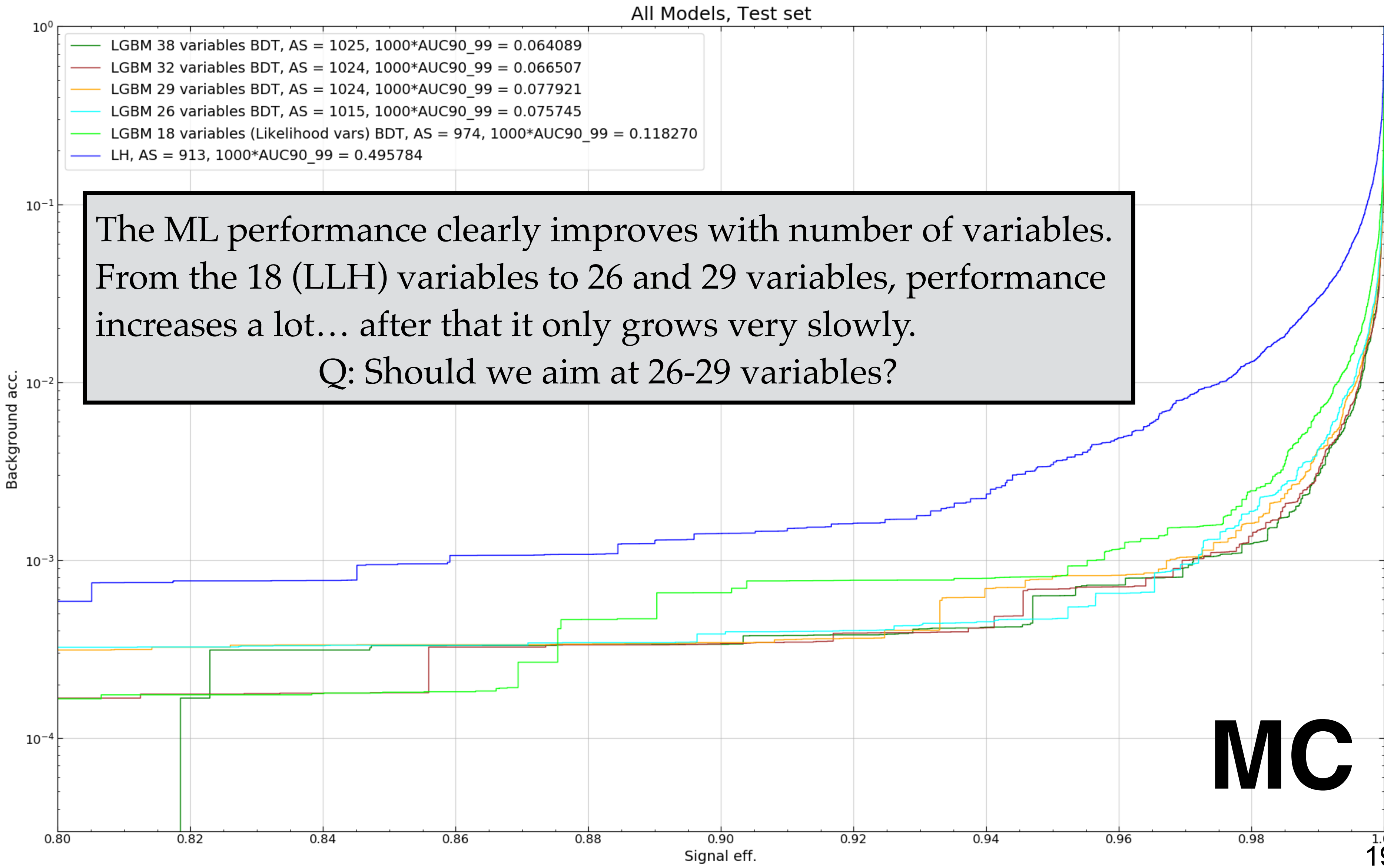
We train the Machine Learning (ML) algorithm (LightGBM) with a mix of backgrounds, and then see how well it performs on each.
We compare to the current ATLAS LH, not to boast our results, but as a solid reference, which helps us getting the most performant & generel results.





Electron PID performance

The electron PID performance is generally much improved with ML:

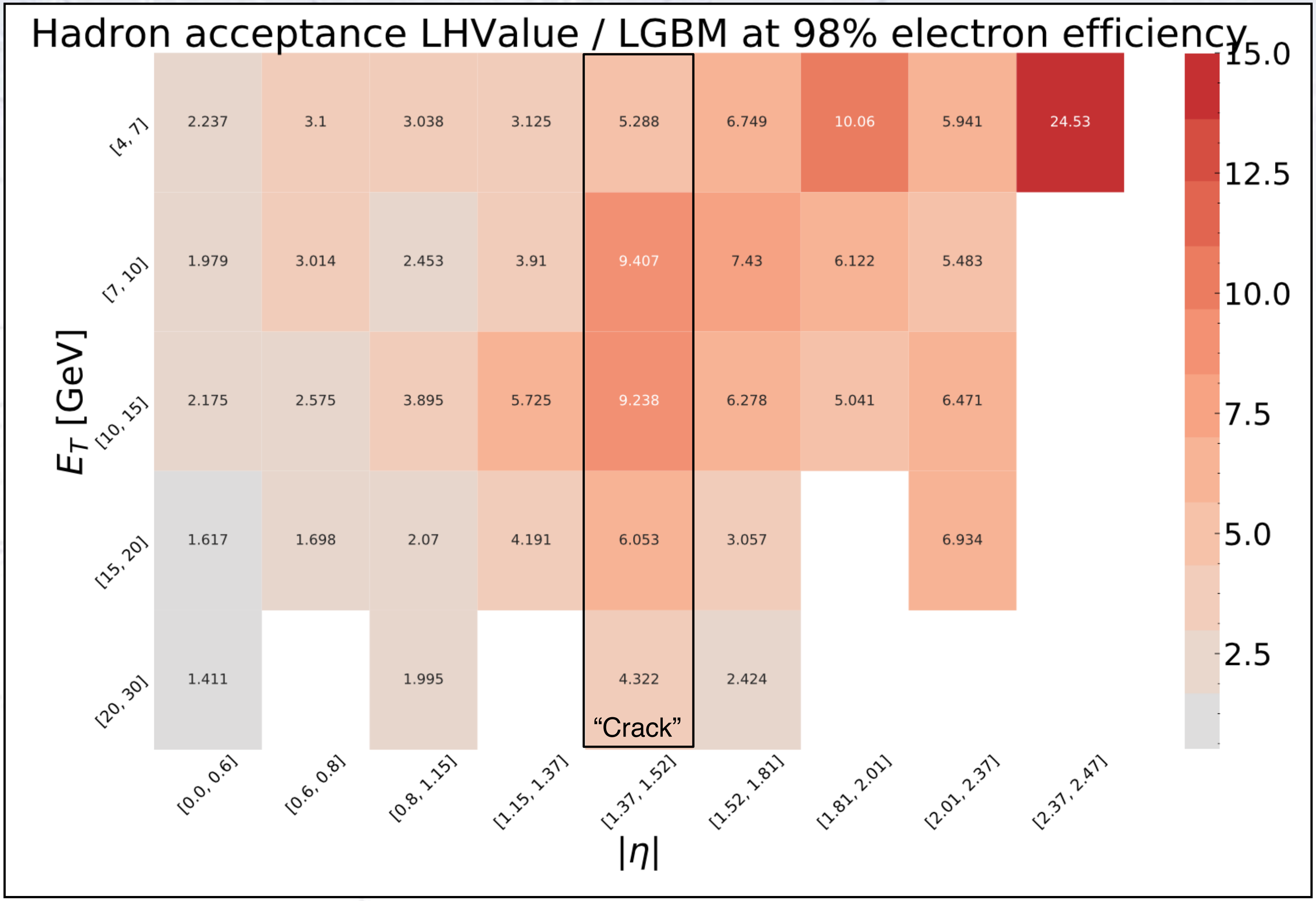


The ML performance clearly improves with number of variables. From the 18 (LLH) variables to 26 and 29 variables, performance increases a lot… after that it only grows very slowly.

Q: Should we aim at 26-29 variables?

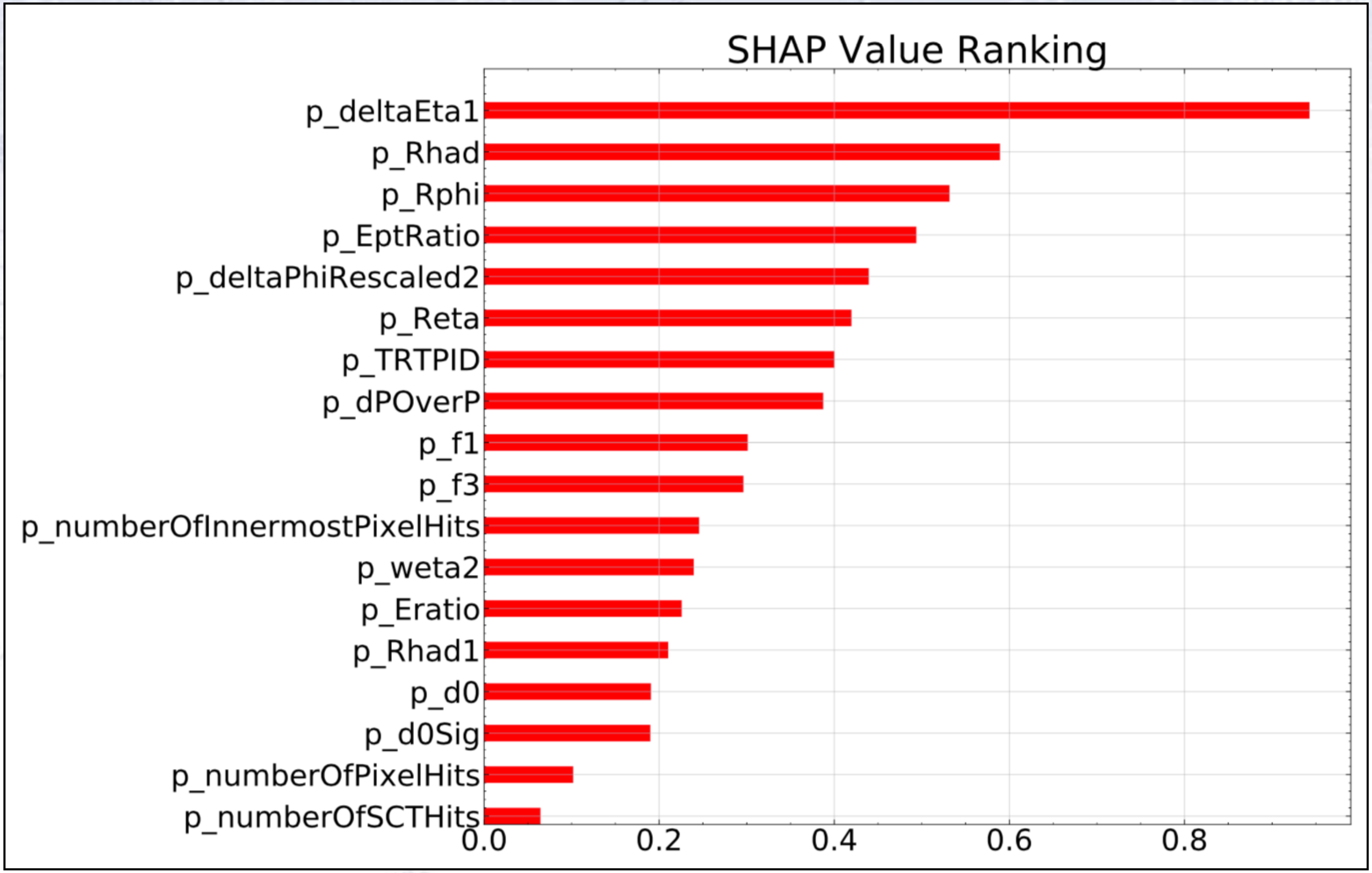
Where do we improve (most)?

The improvements are NOT uniform in energy and angle. We gain most in the "crack" and forward direction.



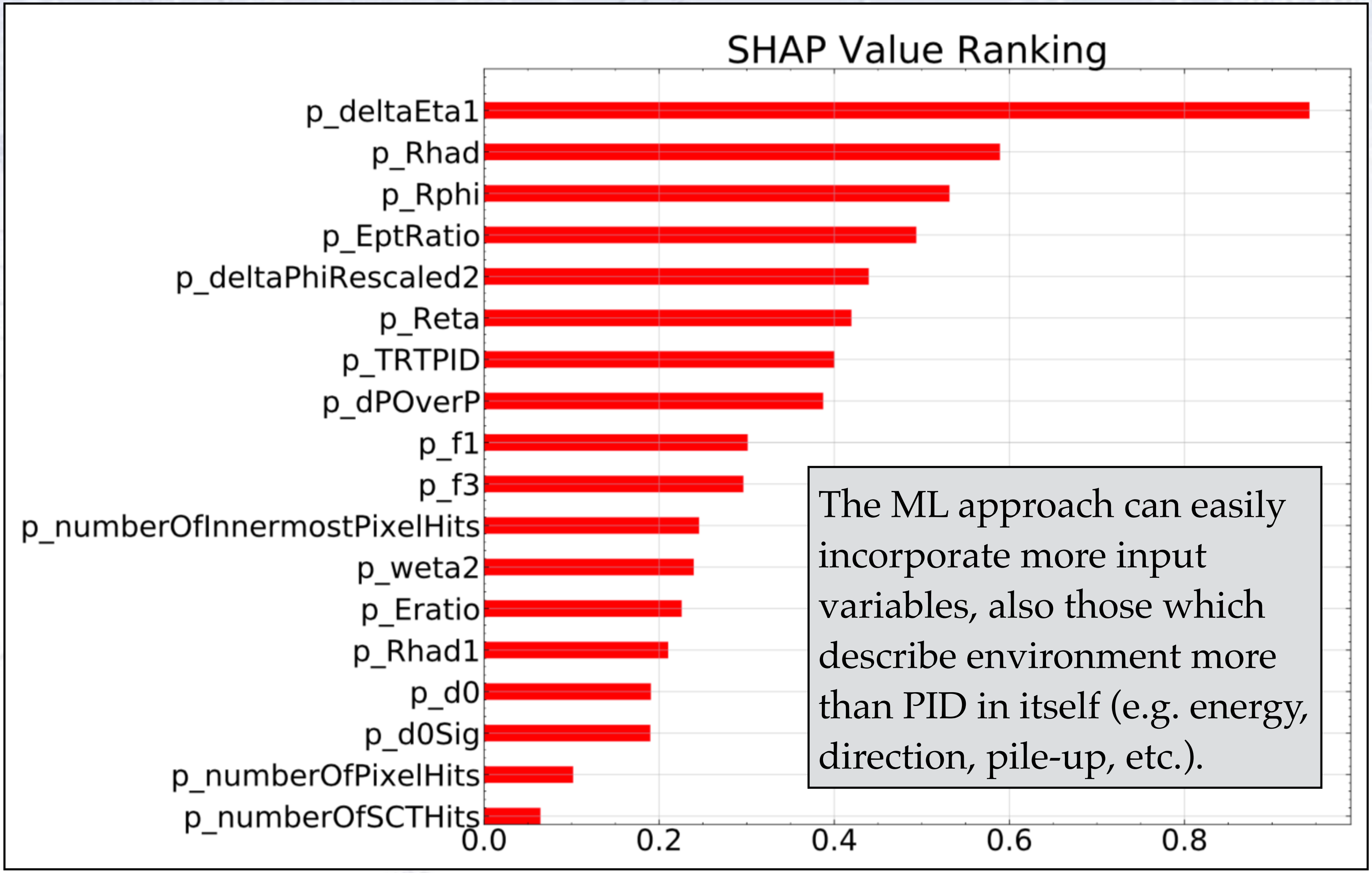
Electron PID feature importance

The importance of each PID input variable is show below (https://github.com/slundberg/shap).



Electron PID feature importance

The importance of each PID input variable is show below (https://github.com/slundberg/shap).



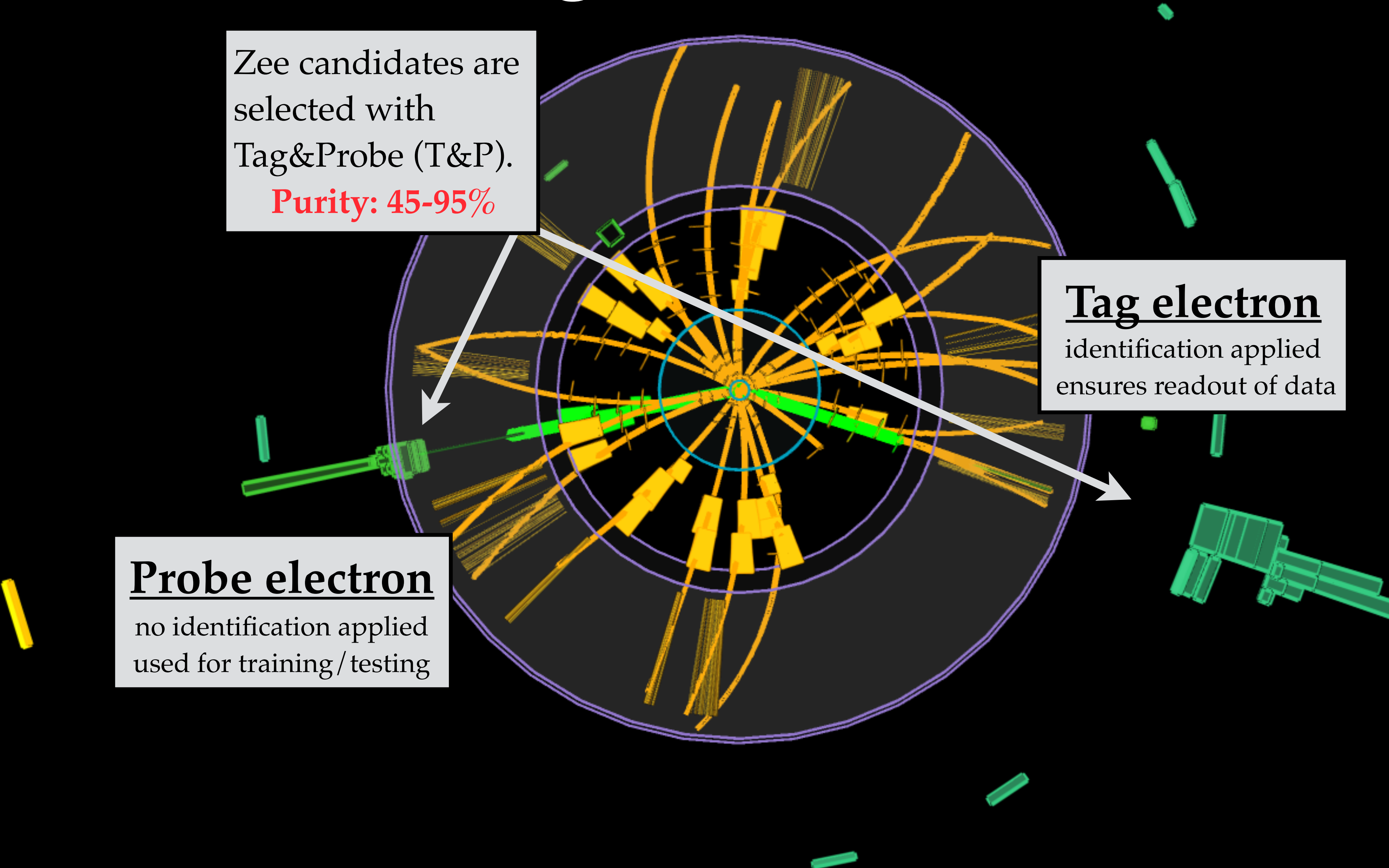
The ML approach can easily incorporate more input variables, also those which describe environment more than PID in itself (e.g. energy, direction, pile-up, etc.).

Status of efforts - DATA

Tag & Probe

Zee candidates are selected with Tag&Probe (T&P).

Purity: 45-95%



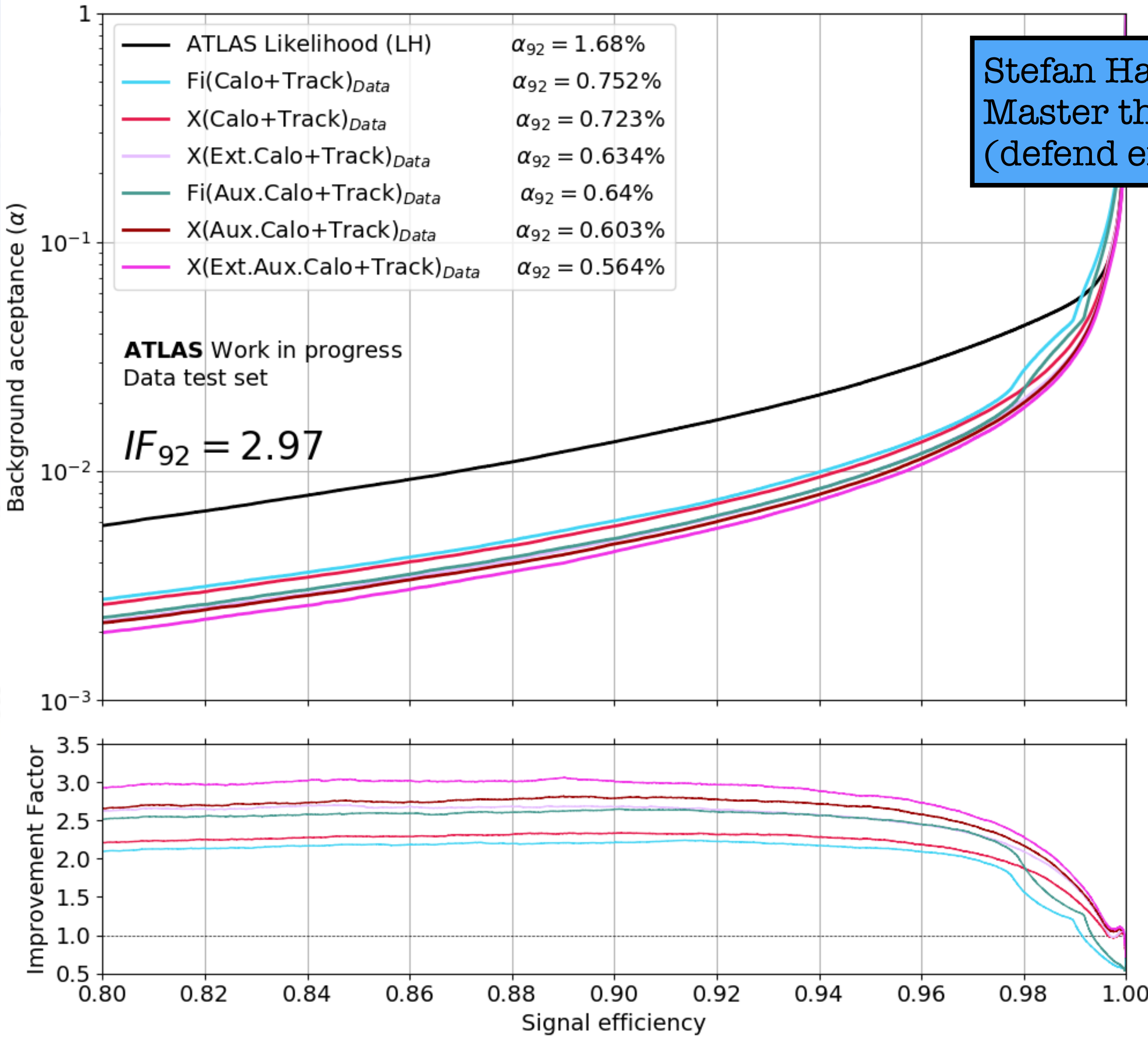
Tag electron

identification applied ensures readout of data

Probe electron

no identification applied used for training/testing

Electron PID on data



Stefan Hasselgren
Master thesis finished (link below)
(defend end of December 2018)

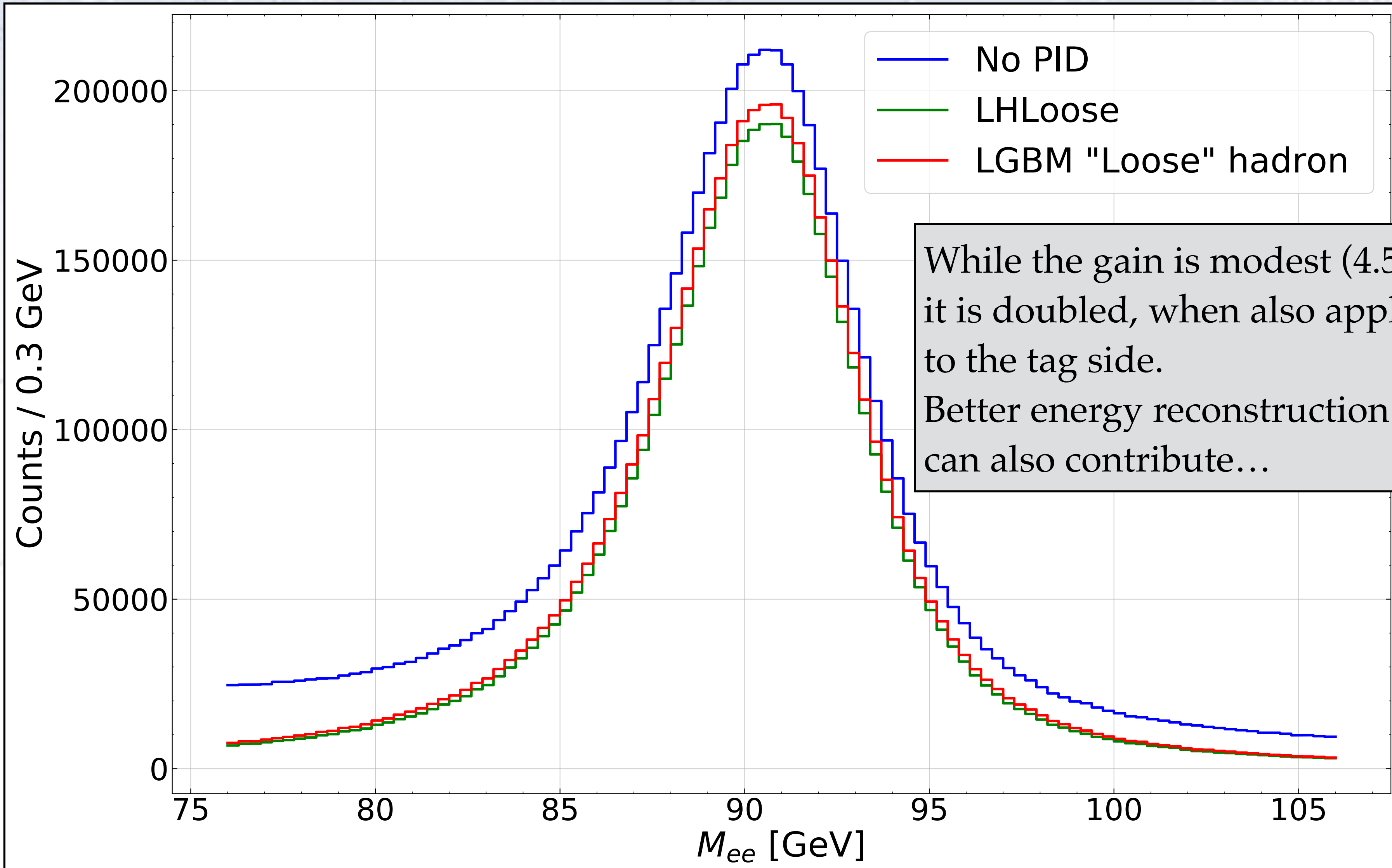
Applying ML PID trained on MC to data naturally gives lesser results.

Also, the shown improvement is a lower bound, as signal in the background lowers (apparent) performance.

http://www.nbi.dk/~petersen/MastersThesis_StefanHasselgren_6dec2018.pdf

Impact in data

ML electron PID on probe side yields **in data** more Zee events (same background):



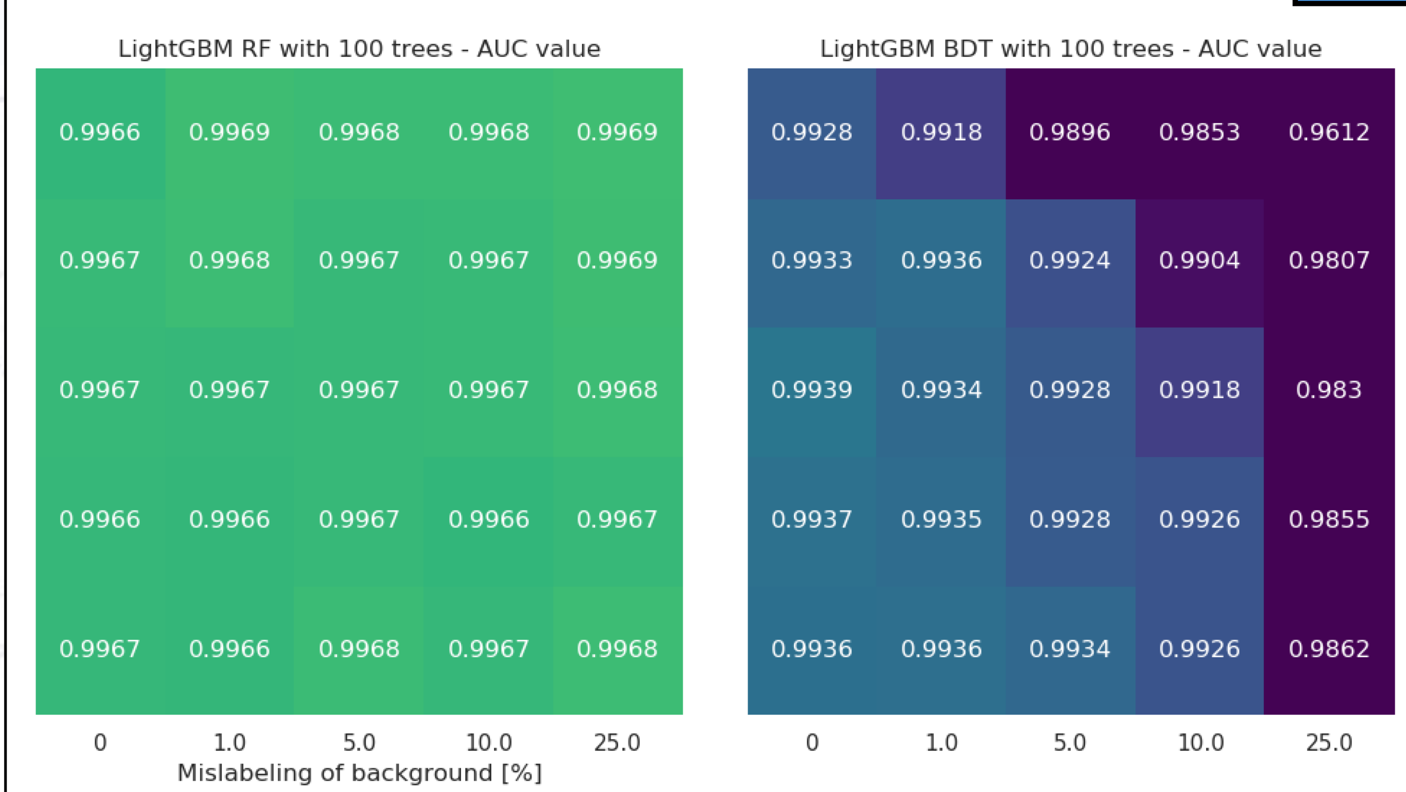
While the gain is modest (4.5%), it is doubled, when also applied to the tag side.
Better energy reconstruction can also contribute…

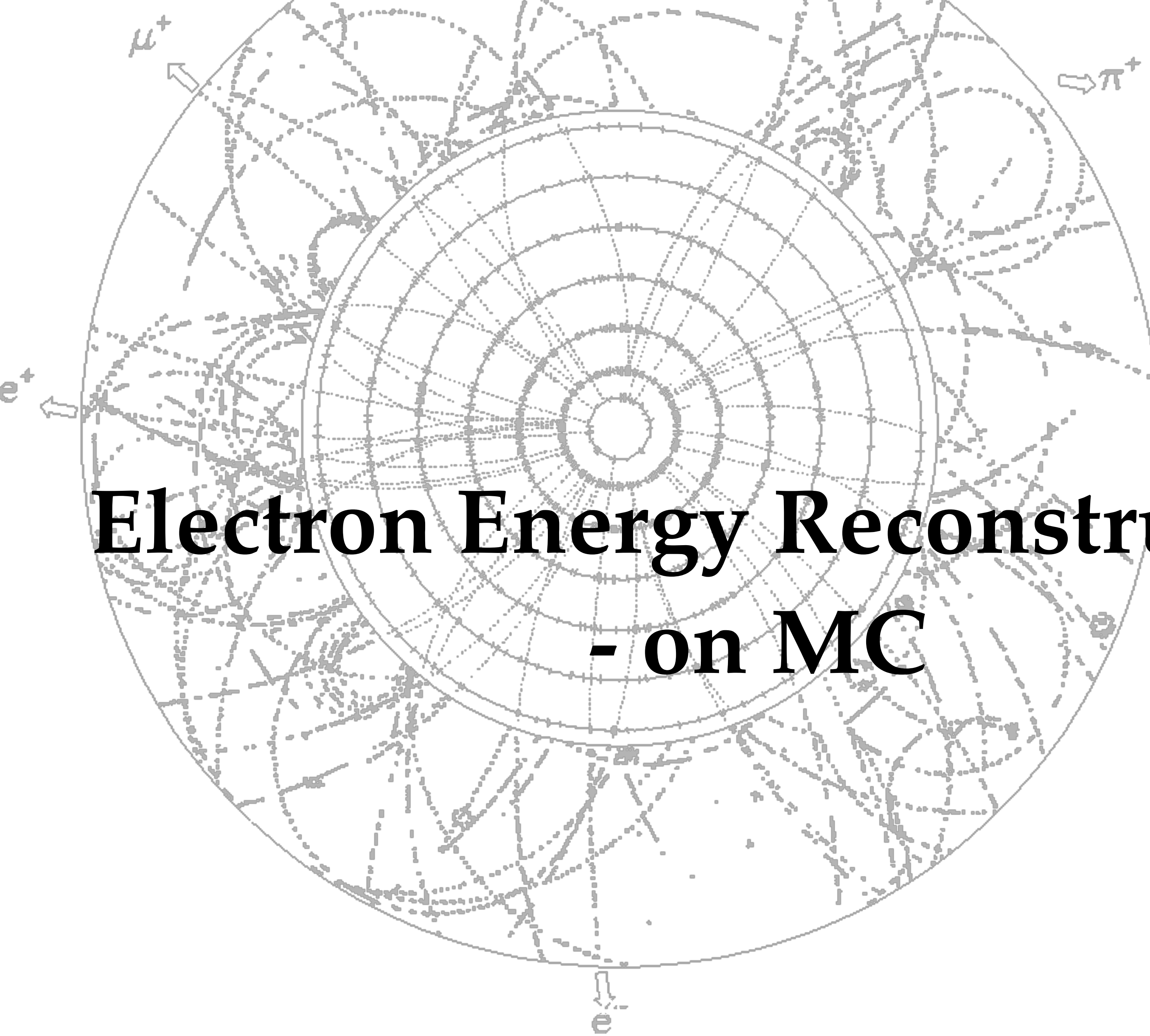
Data: Label confusion

In real data we don't have perfect labelling. The selections for signal and background have a few percent of label confusion each, depending on energy and detector part hit.

It turns out, that Random Forests (RF) perform better than Boosted Decision Trees (BDT) given this label confusion, as one might expect.

Malte Algren





Electron Energy Reconstruction - on MC

Electron ER - BDT vs. CNN

We started to work on **electron energy reconstruction (ER)** using scalar variables combined with a BDT approach, just like ATLAS does.
However, we are now exploring to use a Convoluted Neural Network (CNN) for the task, as this "naturally" fits the problem, when considering the calorimeter cells as images.
Naturally, there are still scalar variables to add to the regression:

BDT scalar variables

- p_eAccCluster
- p_f0Cluster
- p_R12
- p_etaCluster
- p_cellIndexCluster
- p_etaModCalo
- p_phiModCalo
- p_fTG3
- p_dPhiTH3
- p_pt_track
- averageInteractionsPerCrossing
- NvtxReco

  – Blue are used by the current *E* calib

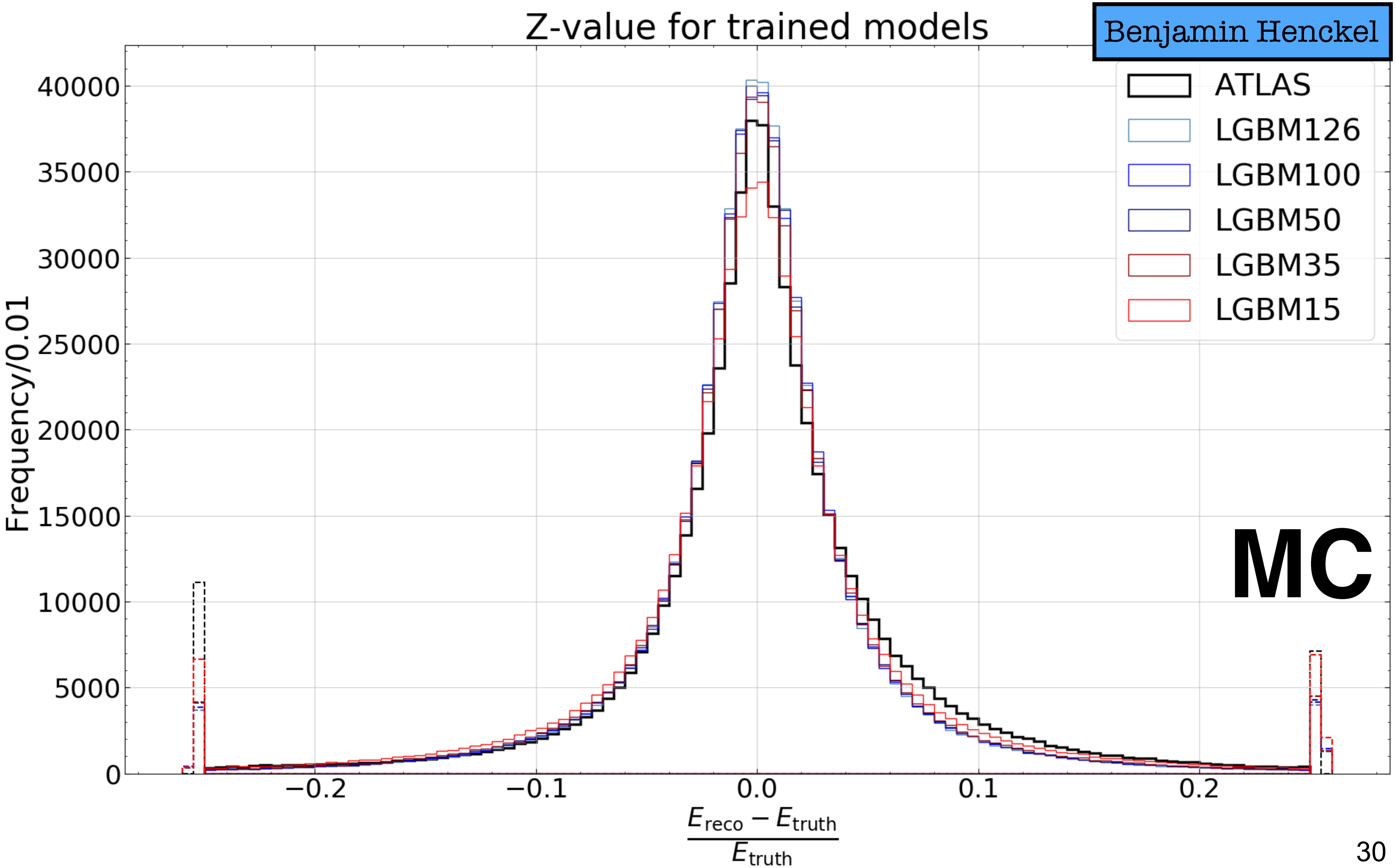
CNN scalar variables

- p_eta
- p_deltaPhiRescaled2
- pX_deltaPhiFromLastMeasurement
- pX_deltaPhiRescaled0
- pX_deltaEta2
- pX_deltaEta3
- p_charge
- BC_distanceFromFront
- BC_filledBunches
- p_pt_track
- averageInteractionsPerCrossing
- NvtxReco

  – No ECAL variables

Photon ER performance

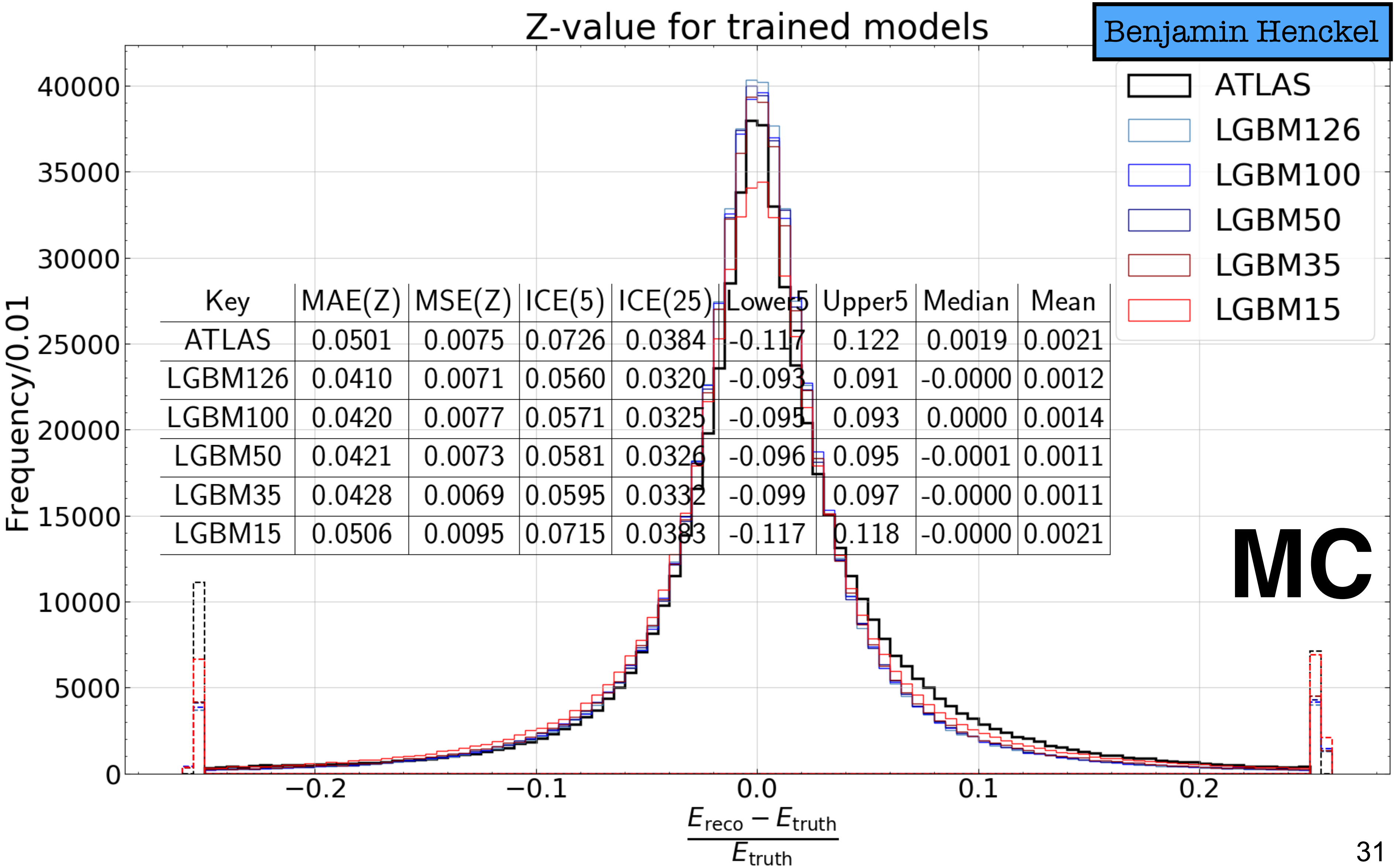
The photon energy reconstruction performance is shown here (for Z➛eeγ sample):



Photon ER performance

The photon energy reconstruction performance is shown here (for Z➛eeγ sample):

Benjamin Henckel



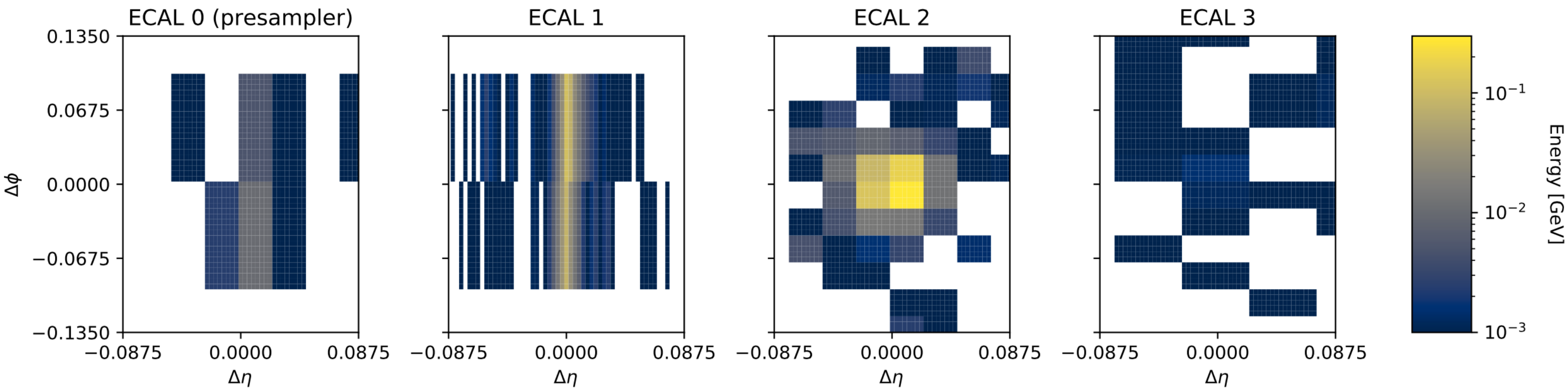
Key	MAE(Z)	MSE(Z)	ICE(5)	ICE(25)	Lower5	Upper5	Median	Mean
ATLAS	0.0501	0.0075	0.0726	0.0384	-0.117	0.122	0.0019	0.0021
LGBM126	0.0410	0.0071	0.0560	0.0320	-0.093	0.091	-0.0000	0.0012
LGBM100	0.0420	0.0077	0.0571	0.0325	-0.095	0.093	0.0000	0.0014
LGBM50	0.0421	0.0073	0.0581	0.0326	-0.096	0.095	-0.0001	0.0011
LGBM35	0.0428	0.0069	0.0595	0.0332	-0.099	0.097	-0.0000	0.0011
LGBM15	0.0506	0.0095	0.0715	0.0383	-0.117	0.118	-0.0000	0.0021

CNN setup

Frederik Faye

- ECAL layers 0 (presampler), 1, 2 and 3 (cell values and times)
- All images are produced to be 56×55 (better than 7×11 and 56×11) and are stacked to give a $56 \times 55 \times 4$ array
- Cell values are scaled to GeV
- Time values are divided by 50 ns

NOTE: For now, we only consider barrel electrons ($|\eta| < 1.3$)



The great thing is that for each cell we don't just have the energy, but also the time (rejecting out-of-time pile-up), gain, and cell noise level (gauging the energy precision).
However, these are not same units, so combined with gate (not concatenation).

CNN architecture

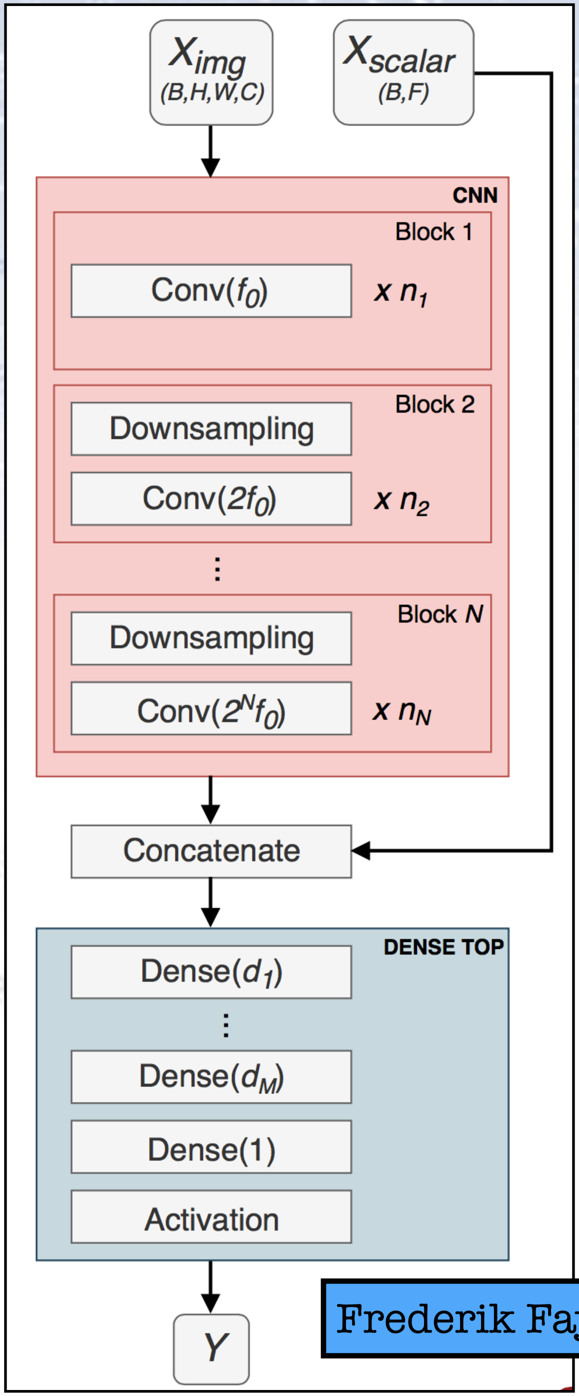
We use a 3 × 3 convolution matrices for all layers.

Each convolution layer is followed by a batch normalisation and activation.

For all i > 1, block begins with downsampling and the number of feature maps is doubled.

A worry is, that the scalar variables "drown" in the many feature map outputs. To be investigated further.
However, we know that scalar variables improve performance as it is!

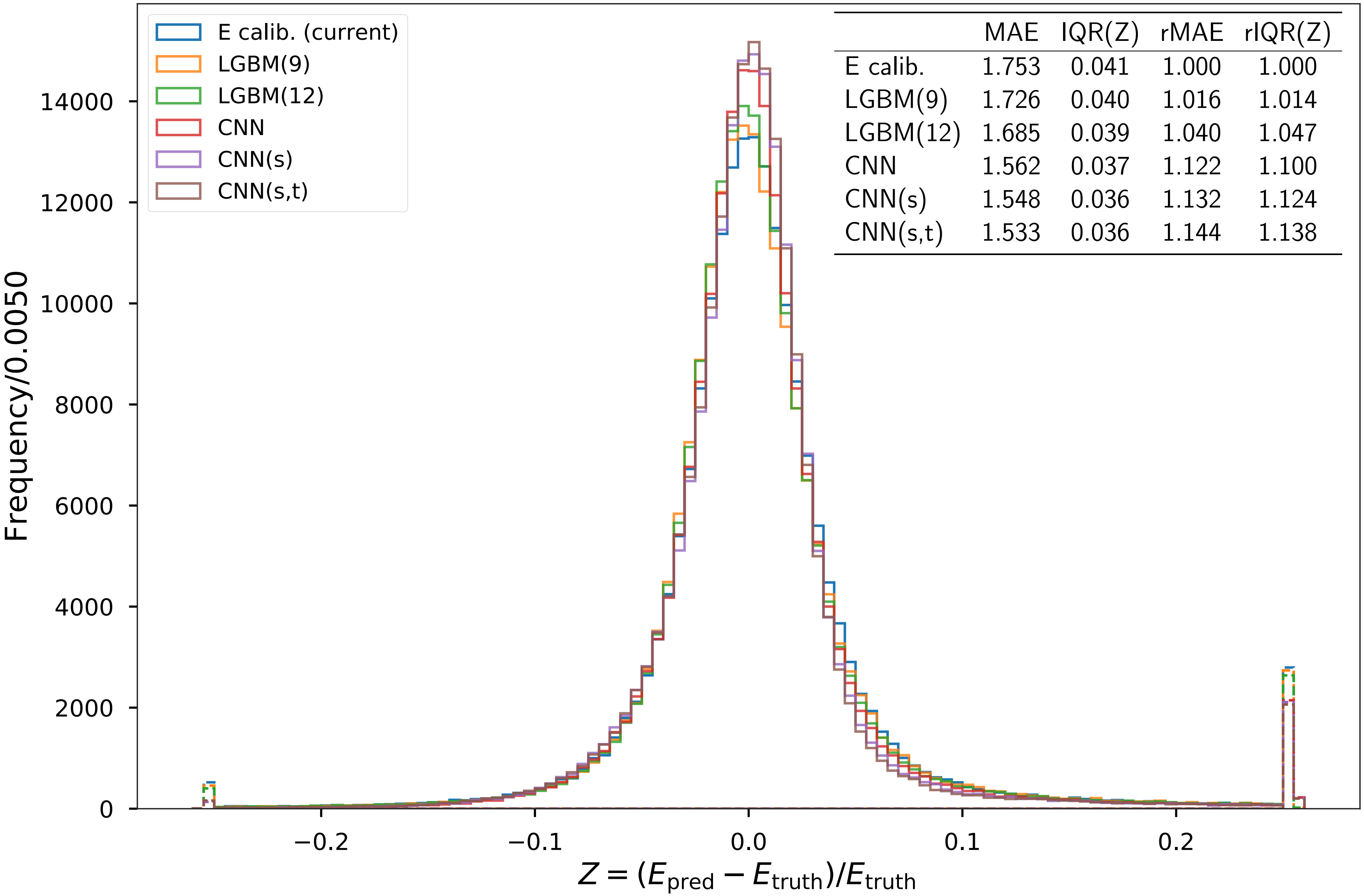
Images containing time are treated differently…



Frederik Faye

CNN results

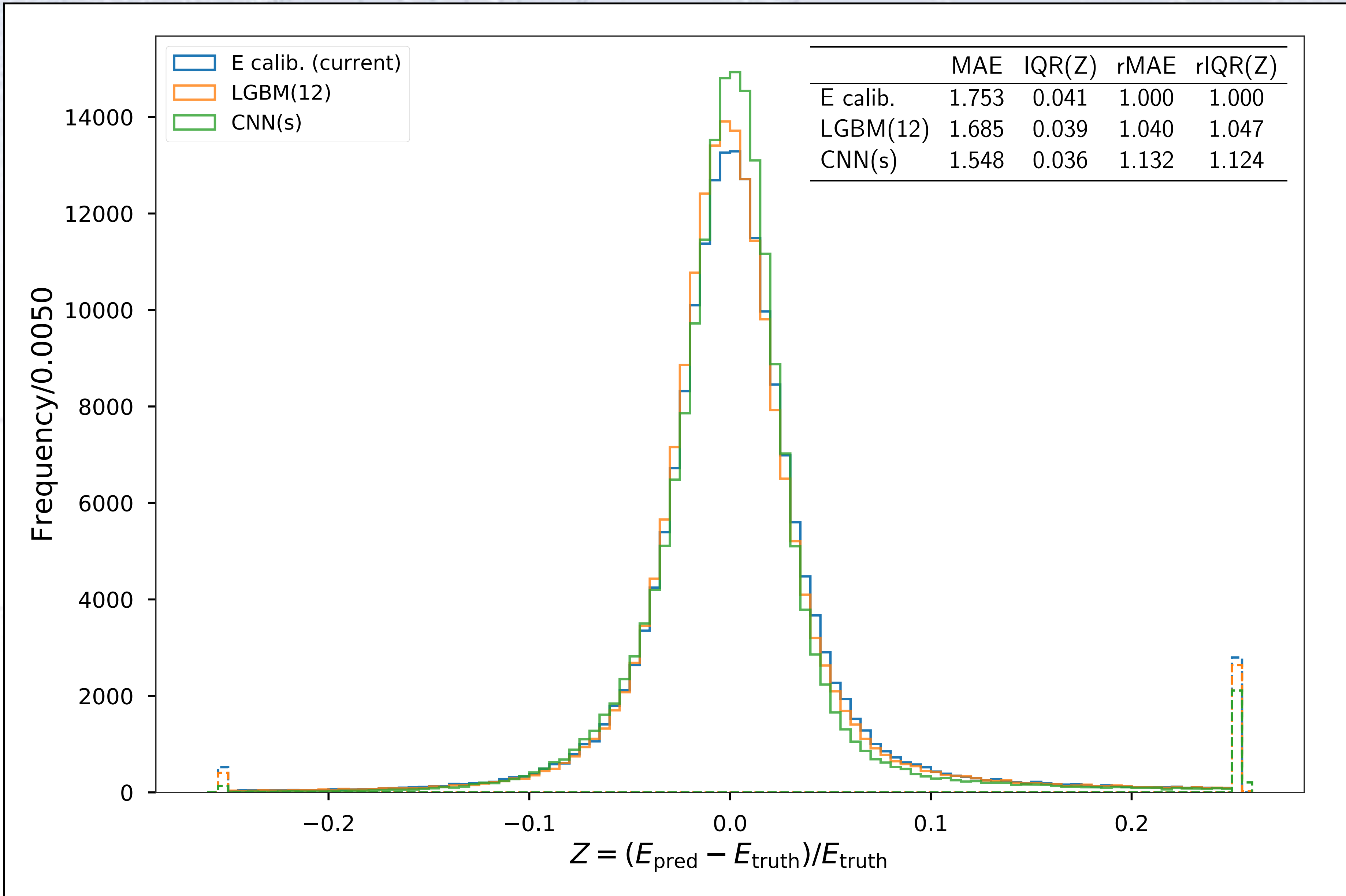
Frederik Faye



	MAE	IQR(Z)	rMAE	rIQR(Z)
E calib.	1.753	0.041	1.000	1.000
LGBM(9)	1.726	0.040	1.016	1.014
LGBM(12)	1.685	0.039	1.040	1.047
CNN	1.562	0.037	1.122	1.100
CNN(s)	1.548	0.036	1.132	1.124
CNN(s,t)	1.533	0.036	1.144	1.138

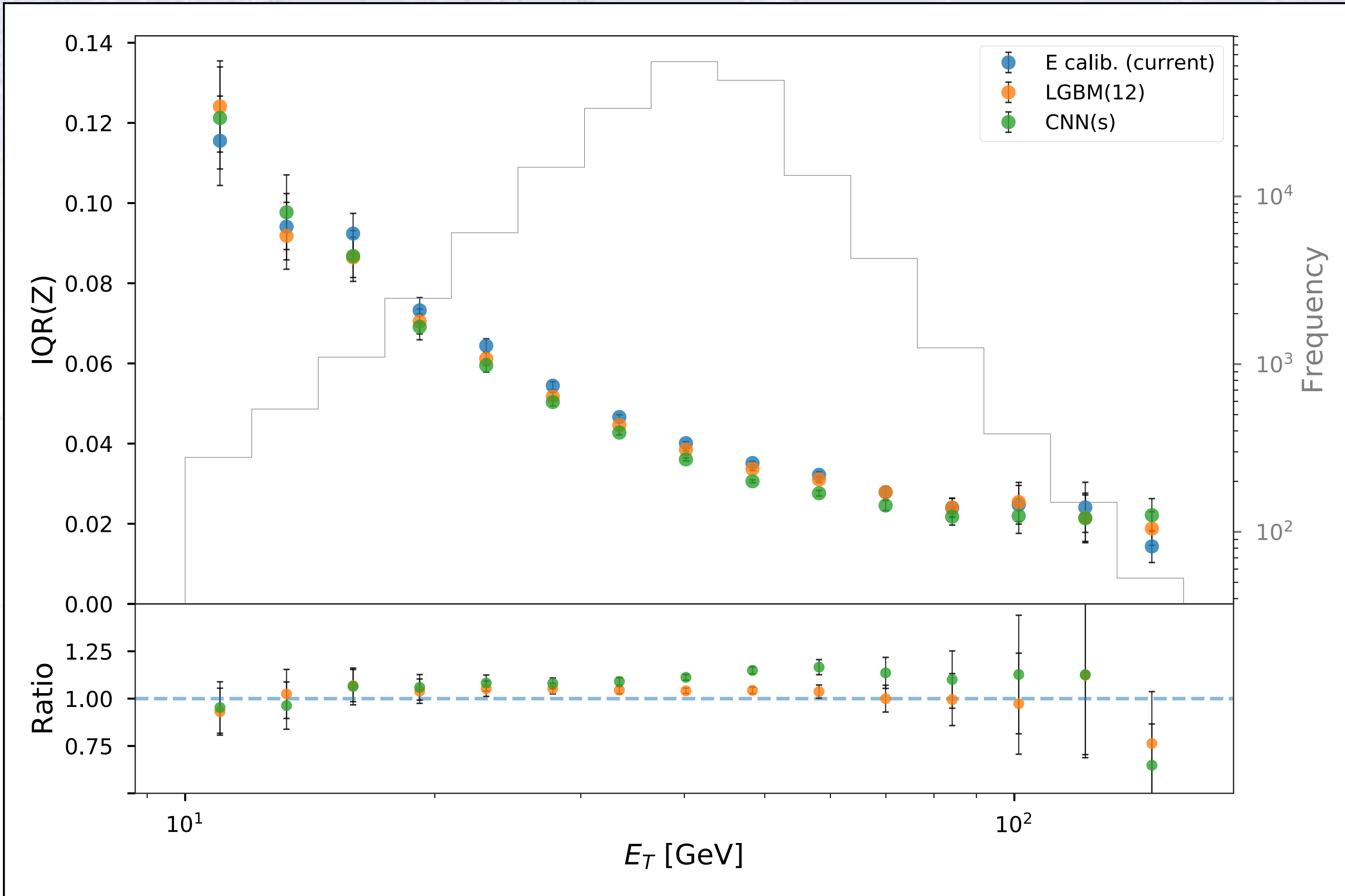
CNN results

Frederik Faye



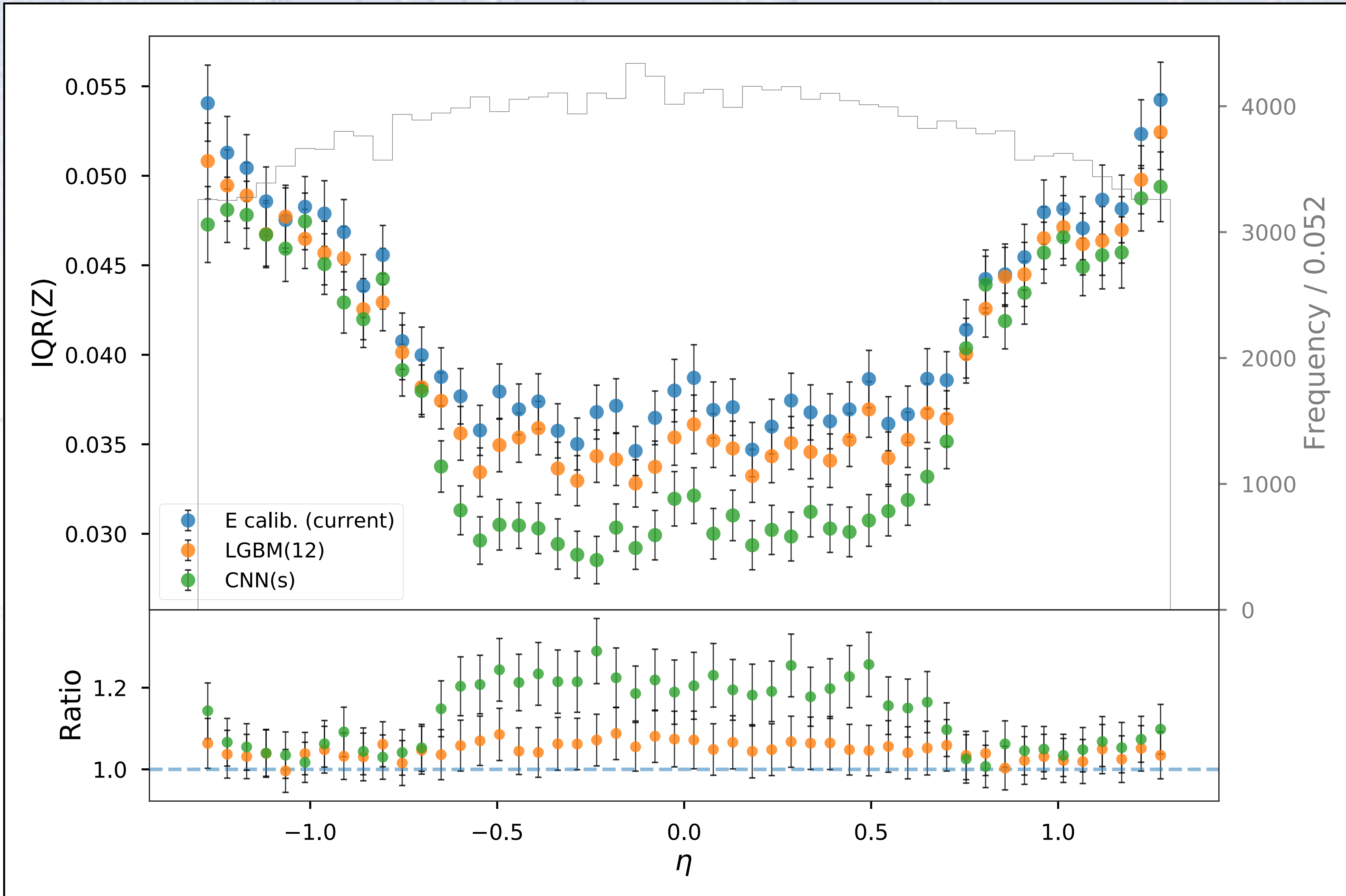
CNN results

Frederik Faye



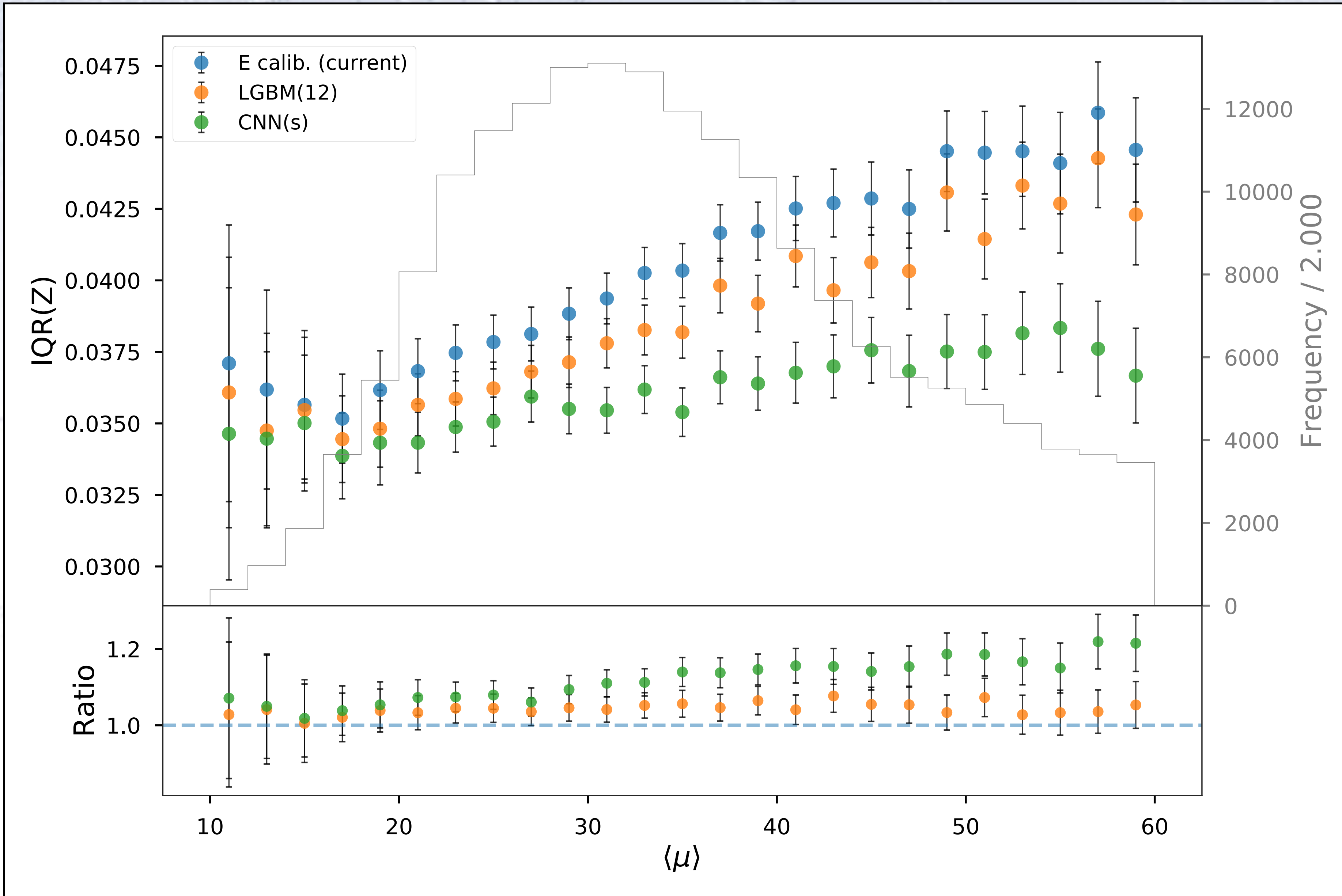
CNN results

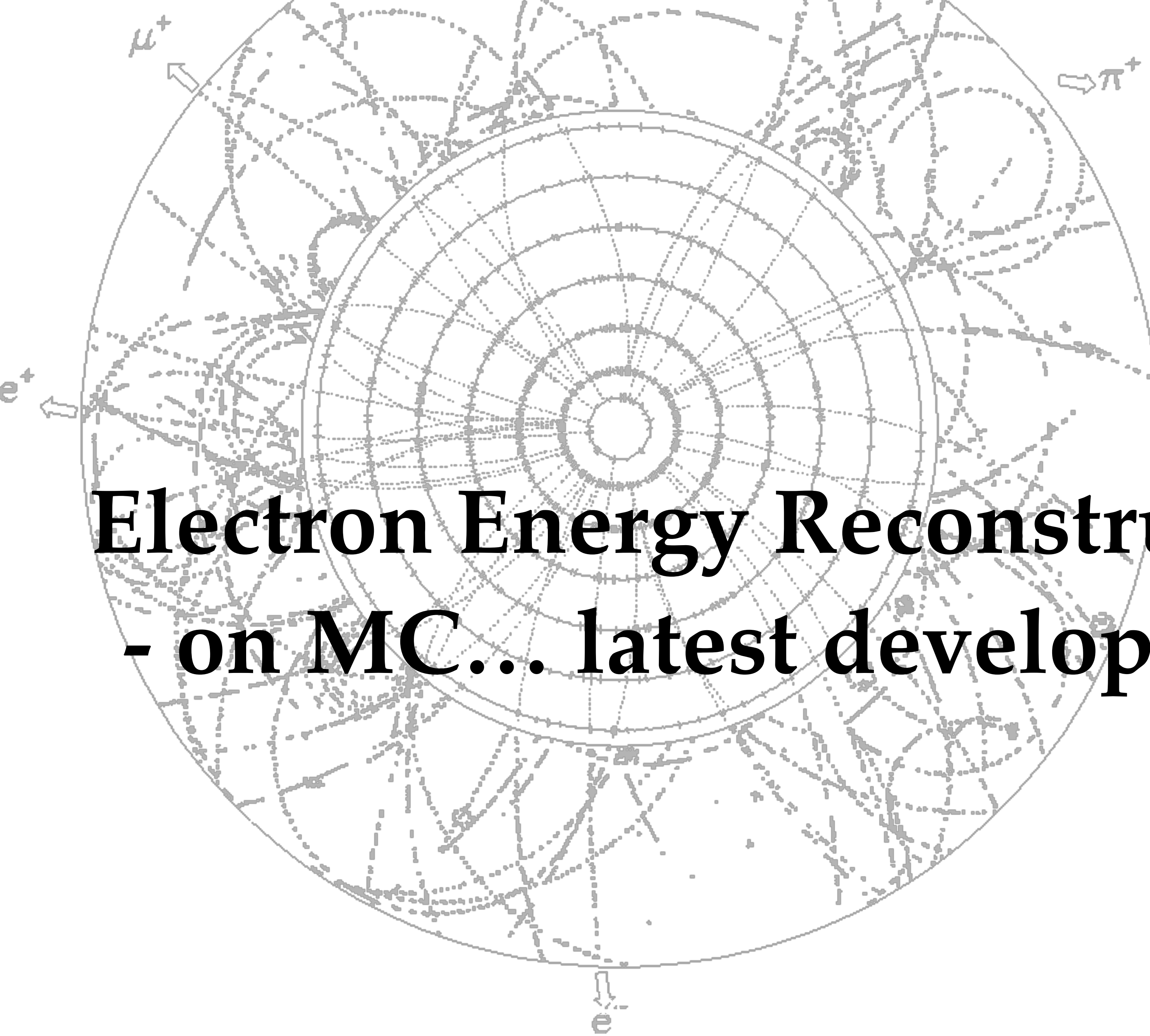
Frederik Faye



CNN results

Frederik Faye



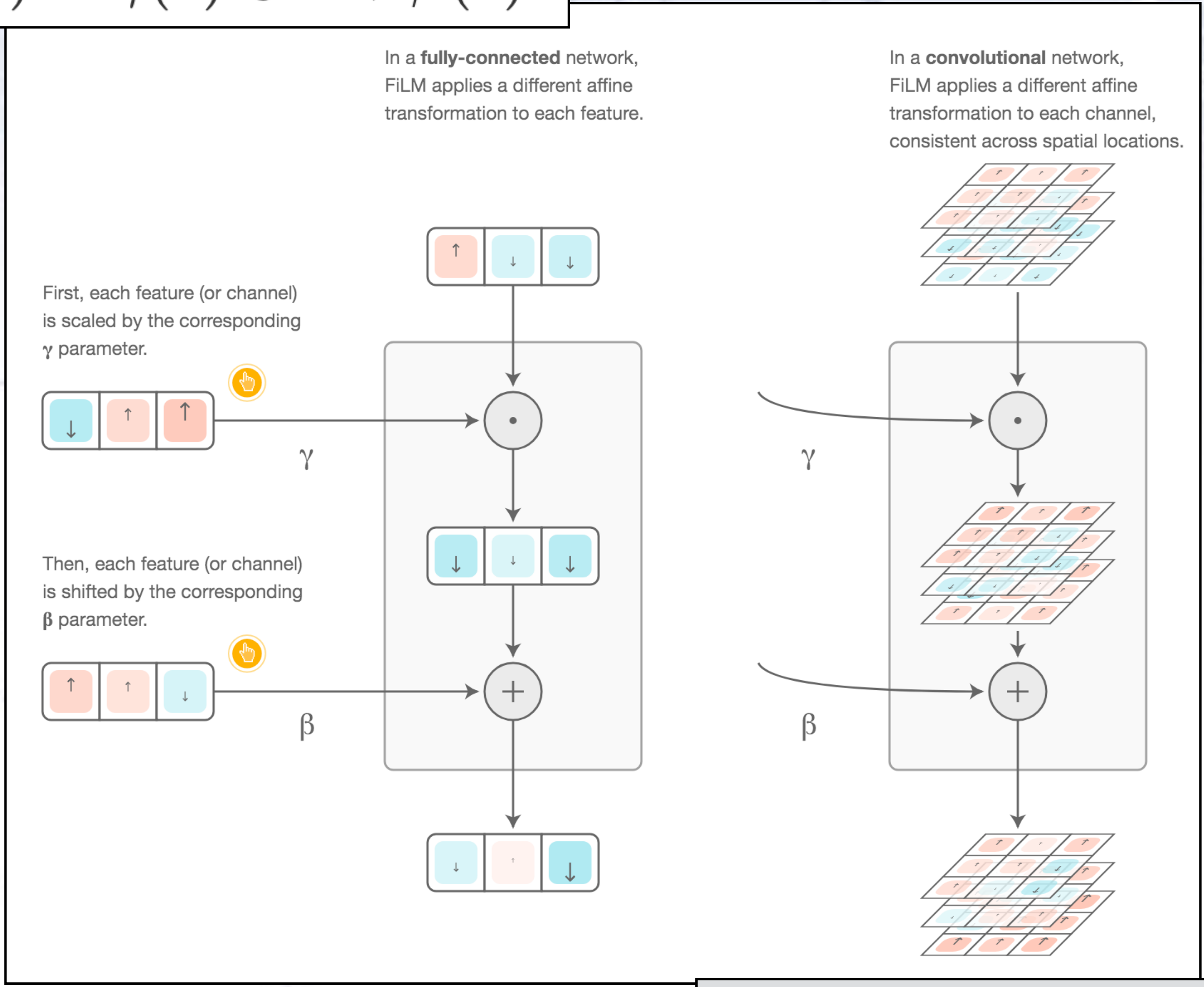


Electron Energy Reconstruction - on MC… latest development!

FiLM = Feature wIse Linear Modulation

Frederik Faye

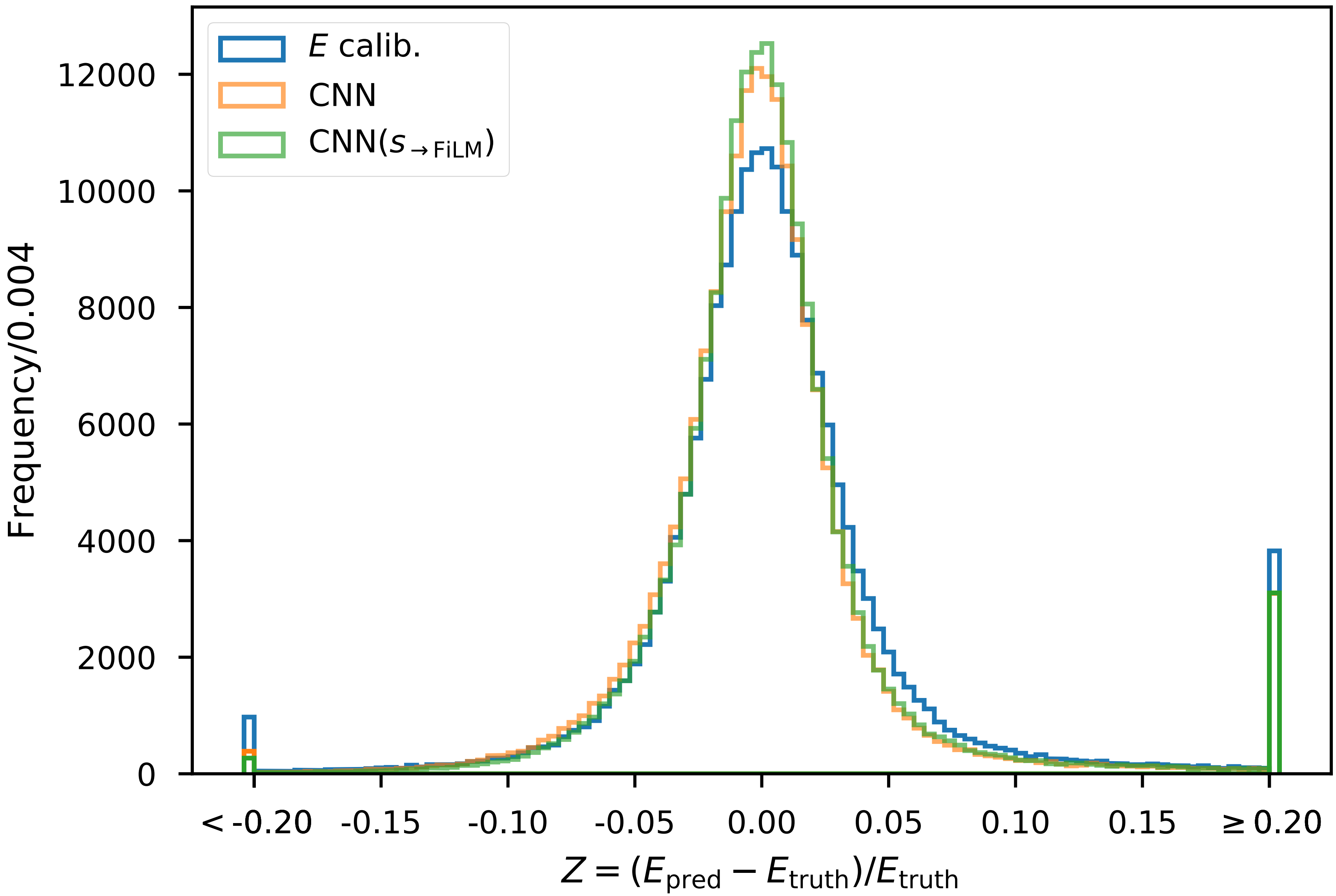
$$\mathrm{FiLM}(\mathbf{x}) = \gamma(\mathbf{z}) \odot \mathbf{x} + \beta(\mathbf{z}).$$



https://distill.pub/2018/feature-wise-transformations/

Results from FiLM

Frederik Faye



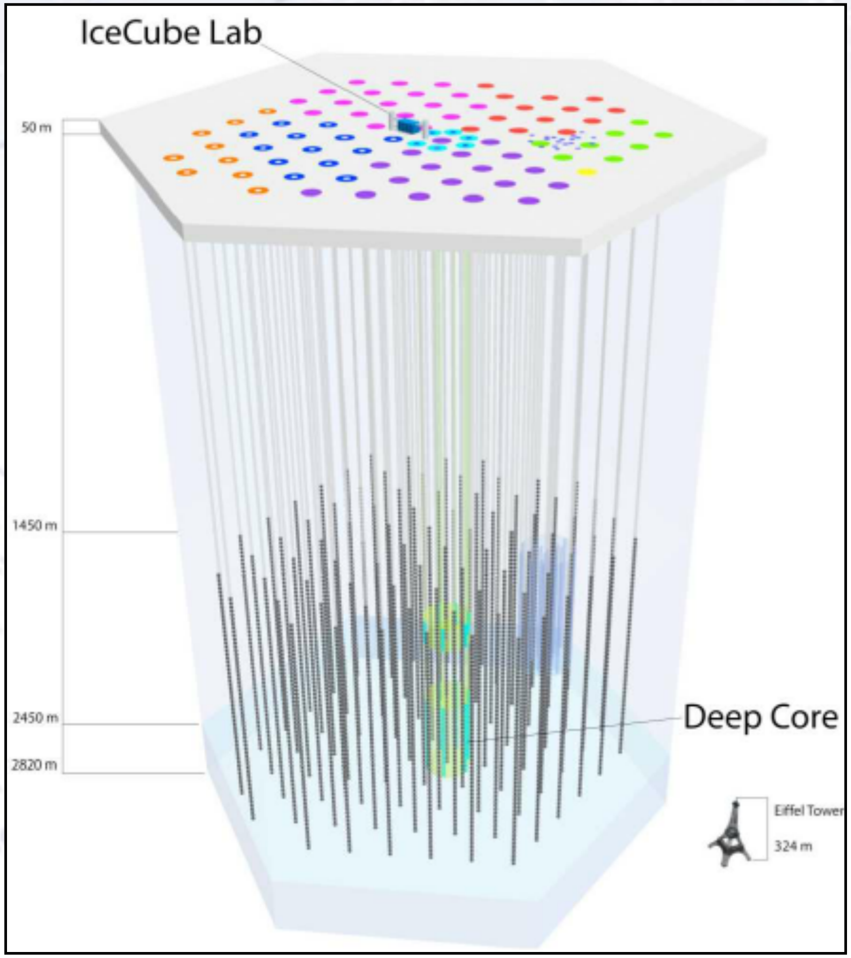
Looking at the future: ν-reconstruction in IceCube

Ideas for the future

The IceCube detector is a less "classic" particle physics detector. Here, 86 strings with about 5000 Digital Optical Modules (DOMs) in total are put in the ice at the South Pole, and used to detector neutrinos (and involuntarily cosmic muons) interact in the ice.

The detector is triggered by coincidences of several adjacent DOMs, and then read out.

Each DOM provides a measurement in time and size of signal. However, there is a significant amount of noise and also effects such as after-pulses, which makes the data less clean.



Ideas for the future

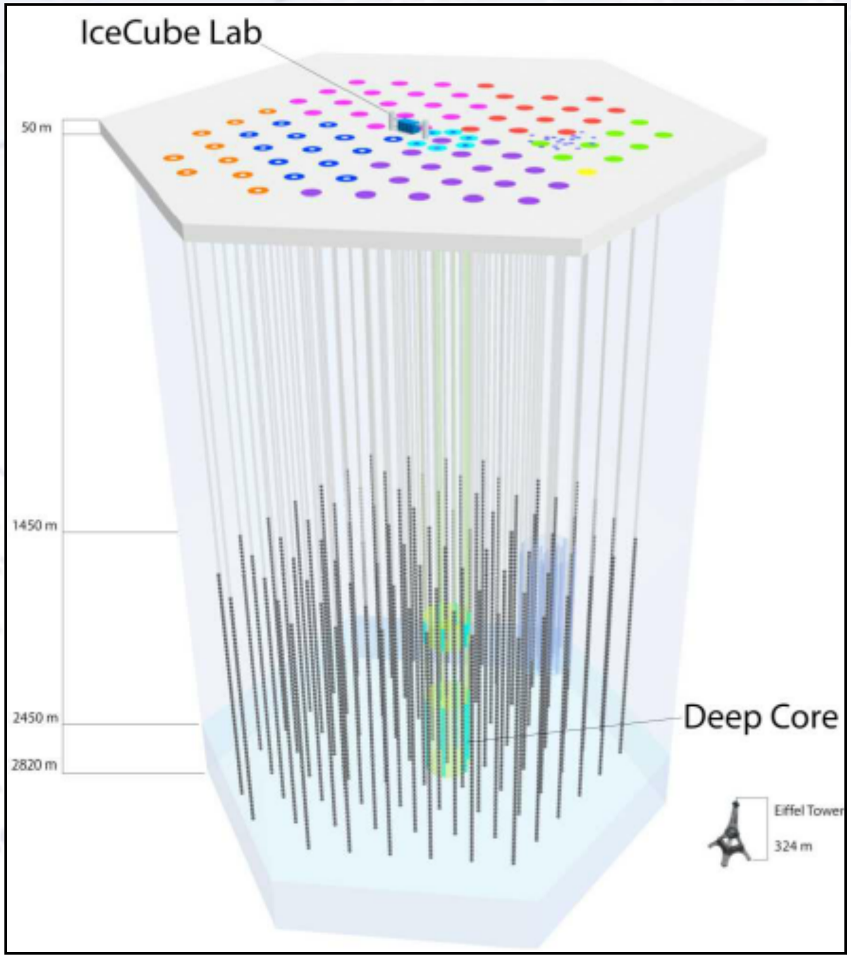
The IceCube detector is a less "classic" particle physics detector. Here, 86 strings with about 5000 Digital Optical Modules (DOMs) in total are put in the ice at the South Pole, and used to detector neutrinos (and involuntarily cosmic muons) interact in the ice.

The detector is triggered by coincidences of several adjacent DOMs, and then read out.

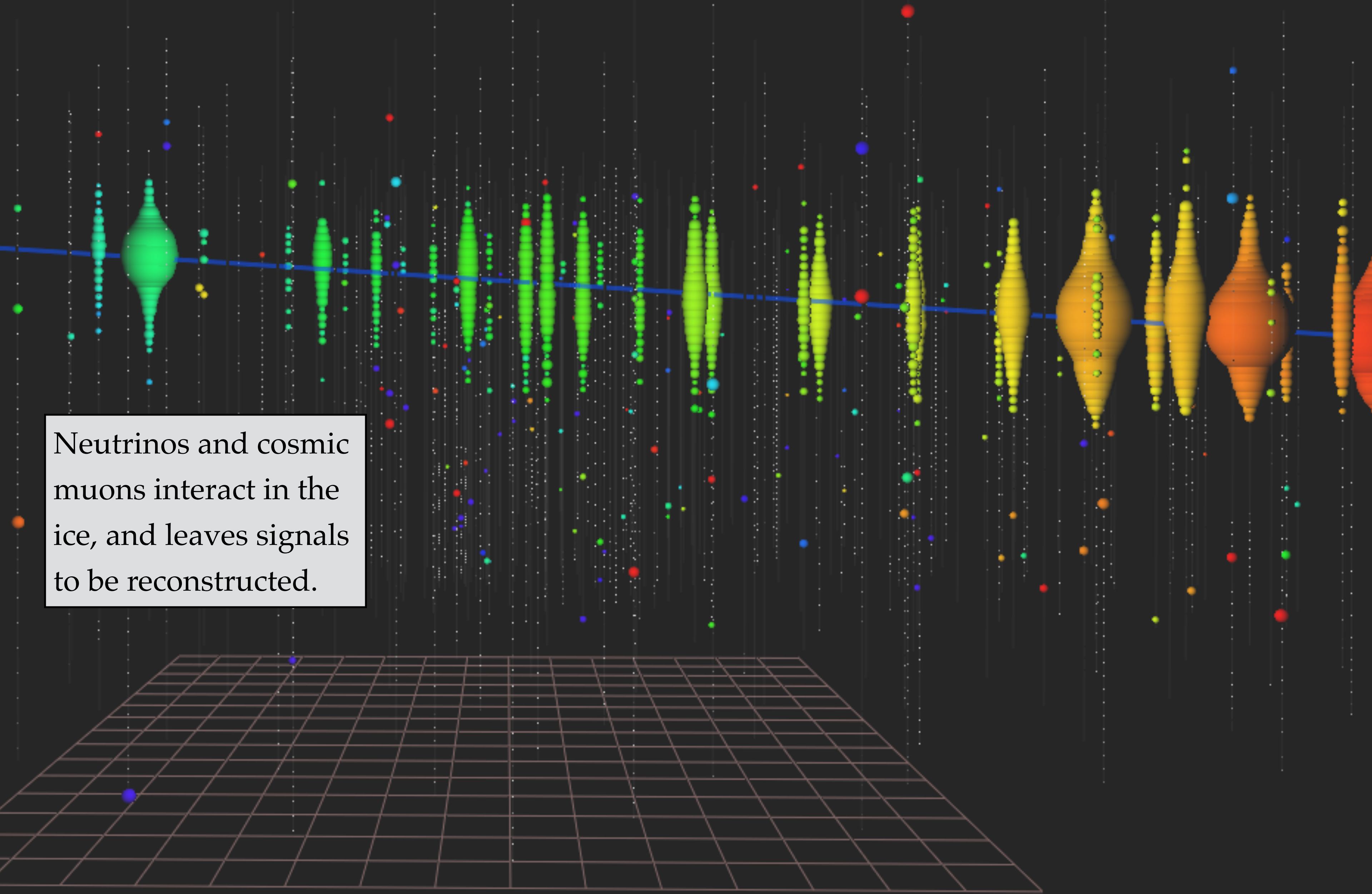
Each DOM provides a measurement in time and size of signal. However, there is a significant amount of noise and also effects such as after-pulses, which makes the data less clean.

The **bottleneck** is the event reconstruction!

This is based on the minimisation of a likelihood including ray tracing and ice properties.

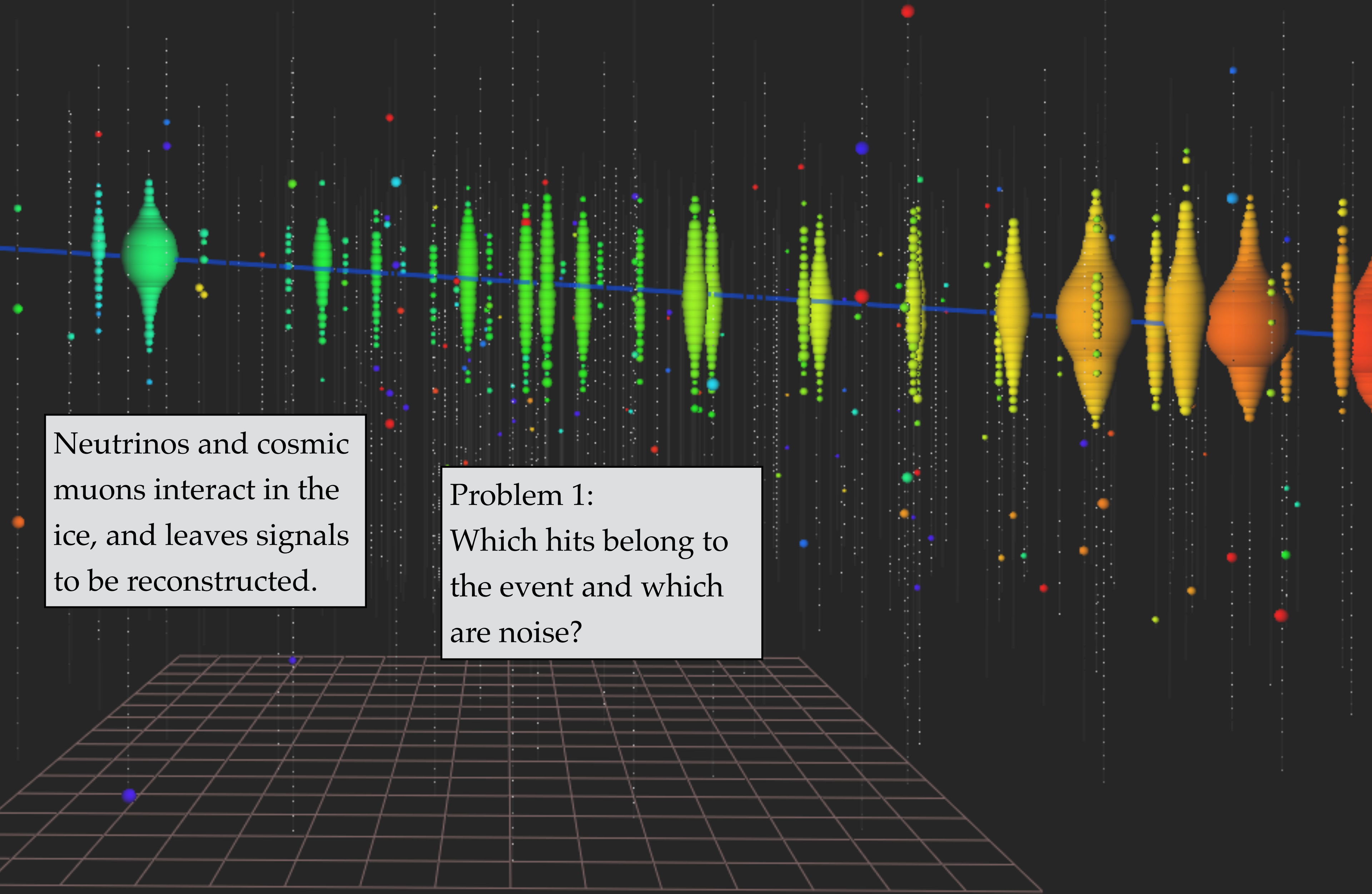


Ideas for the future



Neutrinos and cosmic muons interact in the ice, and leaves signals to be reconstructed.

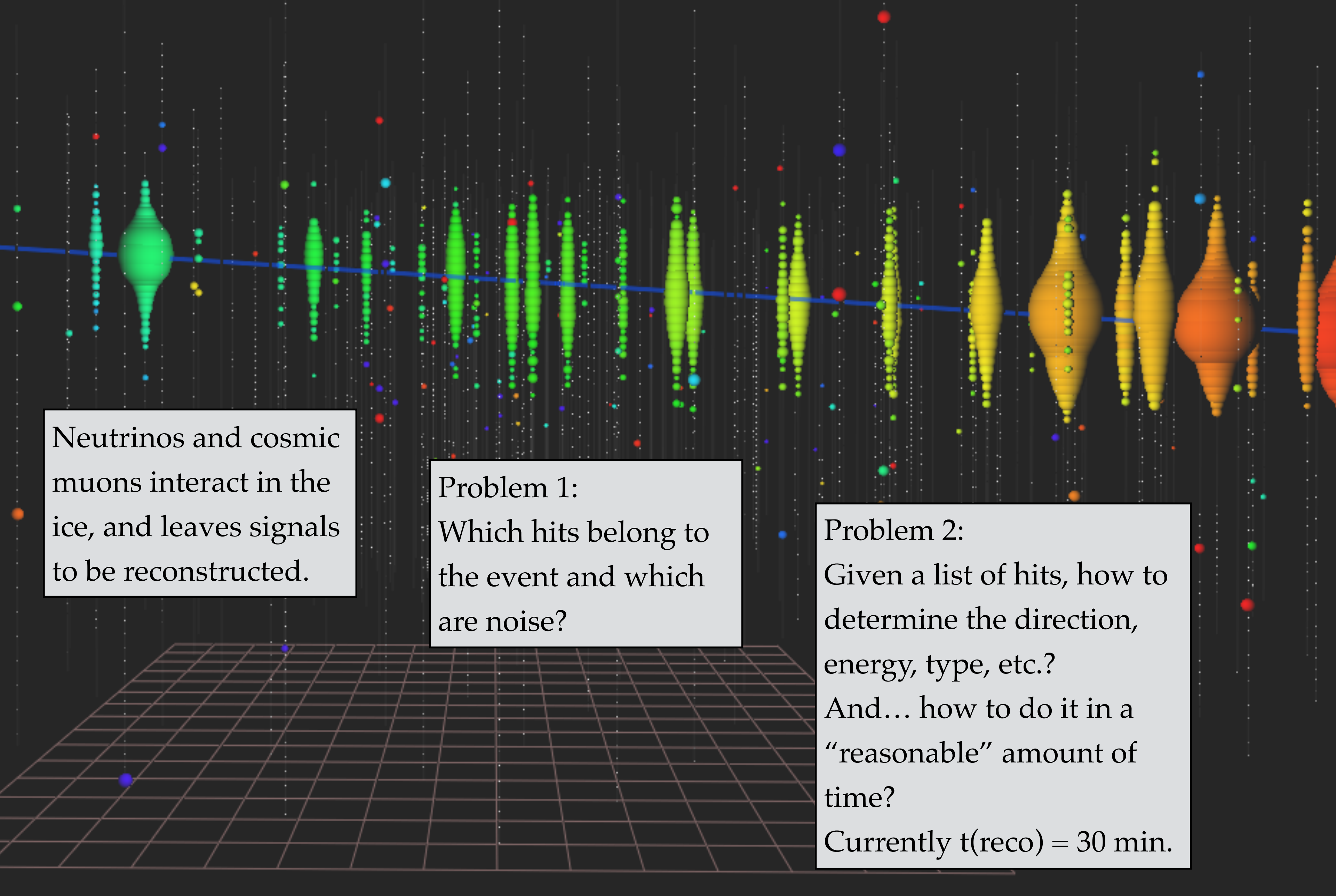
Ideas for the future



Neutrinos and cosmic muons interact in the ice, and leaves signals to be reconstructed.

Problem 1:
Which hits belong to the event and which are noise?

Ideas for the future

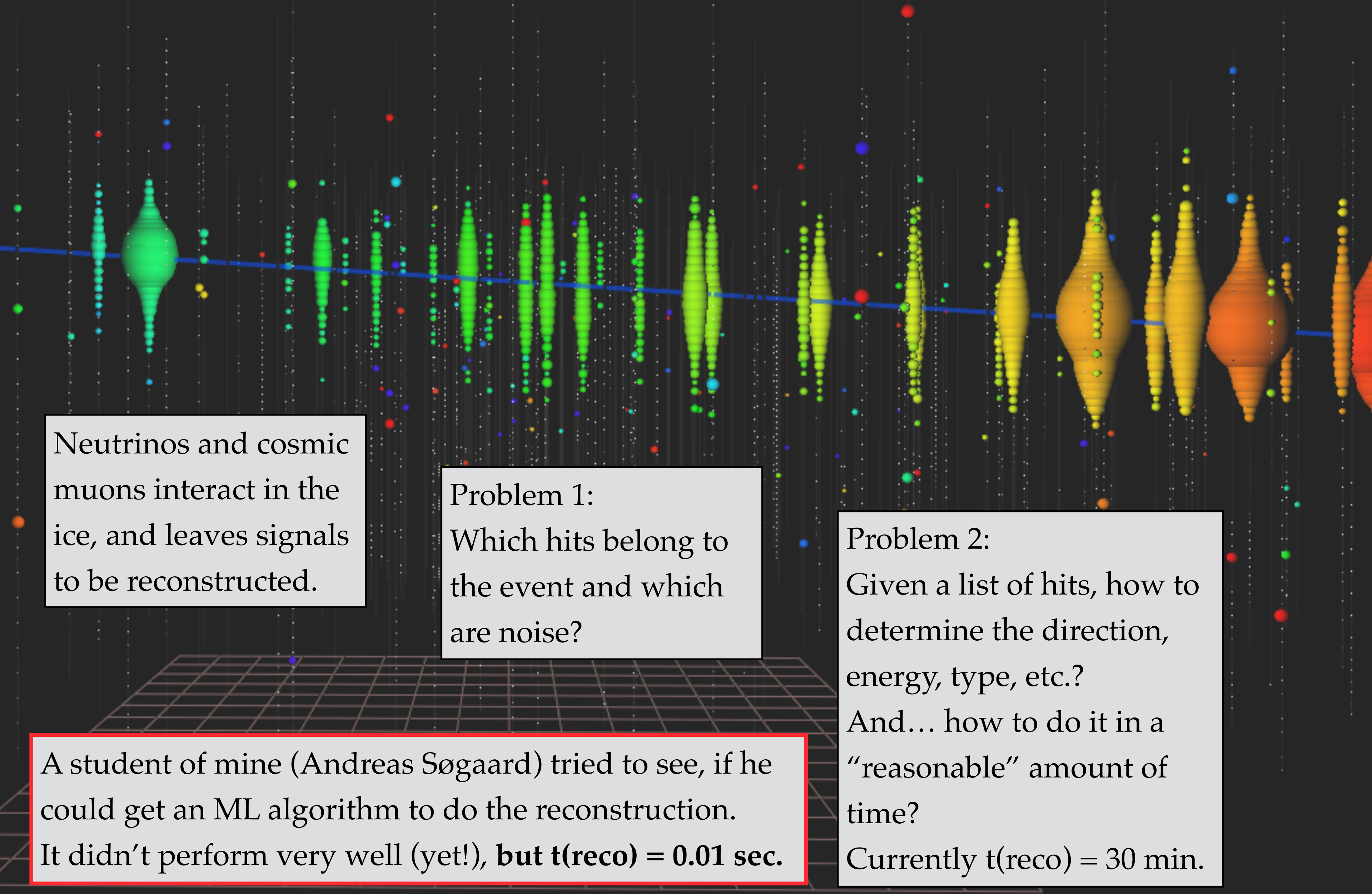


Neutrinos and cosmic muons interact in the ice, and leaves signals to be reconstructed.

Problem 1:
Which hits belong to the event and which are noise?

Problem 2:
Given a list of hits, how to determine the direction, energy, type, etc.?
And… how to do it in a "reasonable" amount of time?
Currently t(reco) = 30 min.

Ideas for the future



Neutrinos and cosmic muons interact in the ice, and leaves signals to be reconstructed.

Problem 1:
Which hits belong to the event and which are noise?

Problem 2:
Given a list of hits, how to determine the direction, energy, type, etc.?
And… how to do it in a "reasonable" amount of time?
Currently t(reco) = 30 min.

A student of mine (Andreas Søgaard) tried to see, if he could get an ML algorithm to do the reconstruction.
It didn't perform very well (yet!), **but t(reco) = 0.01 sec.**

Conclusions

I think that there is a lot of prospect in Machine Learning for physics.

- New algorithms see the light of day almost daily.
- In some cases, it may simply give a more performant data analysis.
- However, in some cases, it makes all the difference.
- Particle physics data is well suited for ML as we have "accurate" simulation.

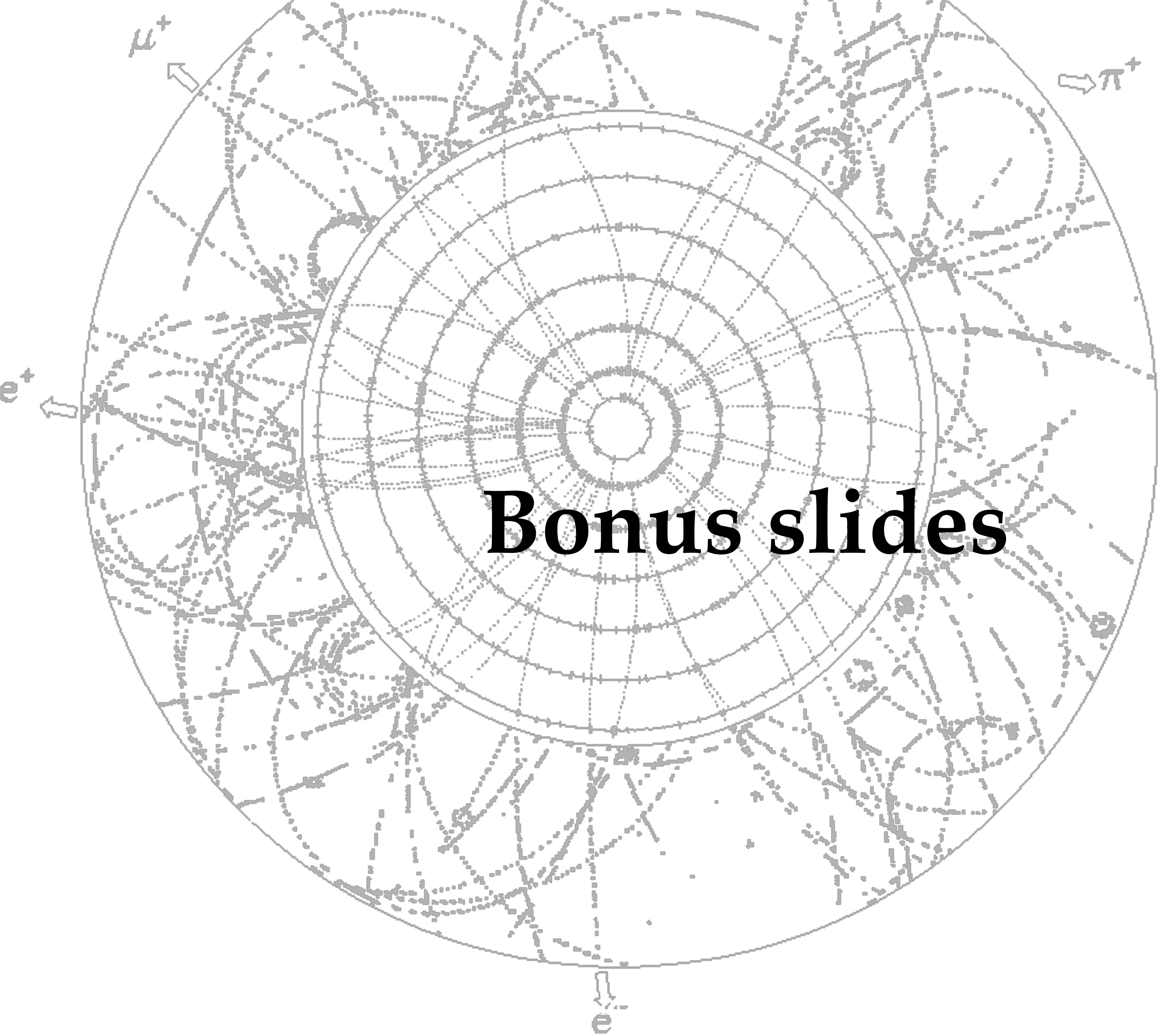
The data requires collaboration, as there are several "particle physics tricks" needed to evaluate performance in real data.

There are many areas to try ML on:

- Transformation of simulation to match data better (challenge: extrapolation).
- Simulation using GANs and VAEs (already started in ATLAS).
- Reconstruction in the IceCube experiment

I've been surprised by the speed with which students "pick up" ML, once you give them an introduction to it. The challenge is often to find data "suitable" for the algorithms given. We - in HEP - tend to actually have that!

Bonus slides

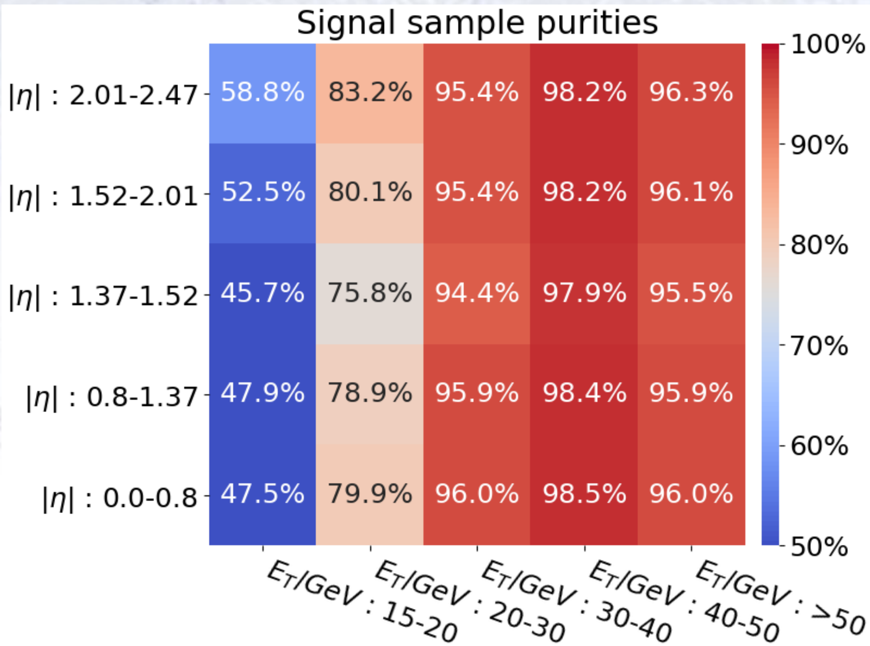


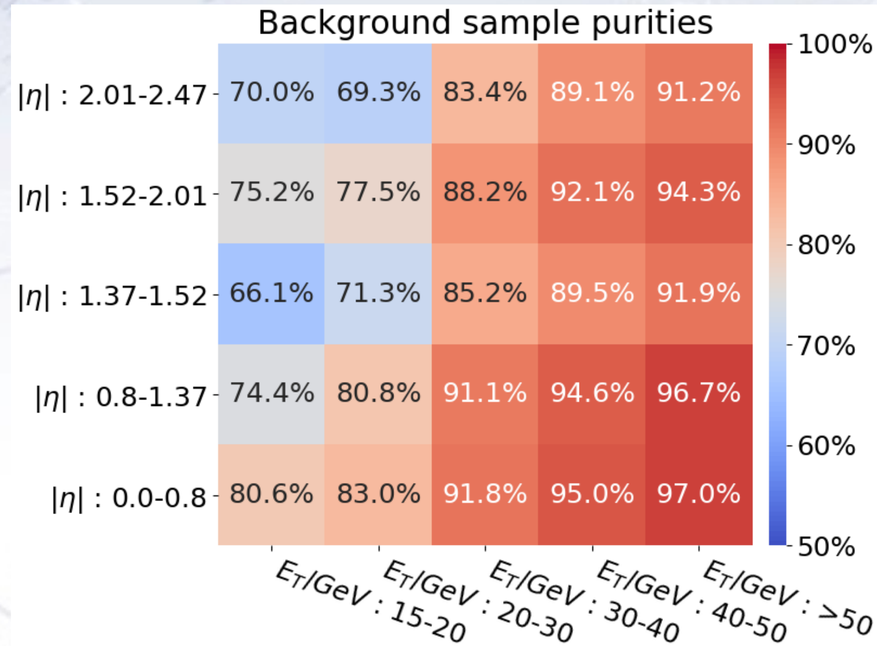
Signal/background selection

- DAOD production: mixture of EGAM1, EGAM3, EGAM7, EGAM8 and EGAM9 including cells and bunch crossing information
- $Z \to ee$ Tag and Probe
  - Tag
    - $p_T > 24.5$ GeV
    - Tight ID
    - Loose Isolation
    - Crack veto and Central
    - Pass HLT_e26_lhtight_nod0_ivarloose
    - Has track particle and vertex
    - Pass object quality cut
  - Probe
    - $p_T > 4.5$ GeV
    - Pass object quality cut
    - $|\eta| < 4.9$
  - Combined
    - $dR > 0.4$ for tag and probe
    - $M_{ee} > 50$ GeV
- Dijet, $W^{\pm} \to \ell^{\pm}\nu$ $(\ell = e, \mu)$, $Z \to \mu\mu$ samples background selection
  - Missing transverse energy (MET) < 25 GeV
  - $p_T > 4.5$ GeV
  - pass Object Quality
  - Z veto: Match with any other medium electron, $|M_{ee} - m_{Z^0}| < 20$ GeV
  - W veto: Match with MET, transverse mass < 40 GeV
- $Z(\gamma) \to ee$ (Drell-Yan), $Z \to \ell\ell\gamma$, $\gamma +$jet
  - EGamma truth particles
- More complete documentation `https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/EgammaMachineLearning`

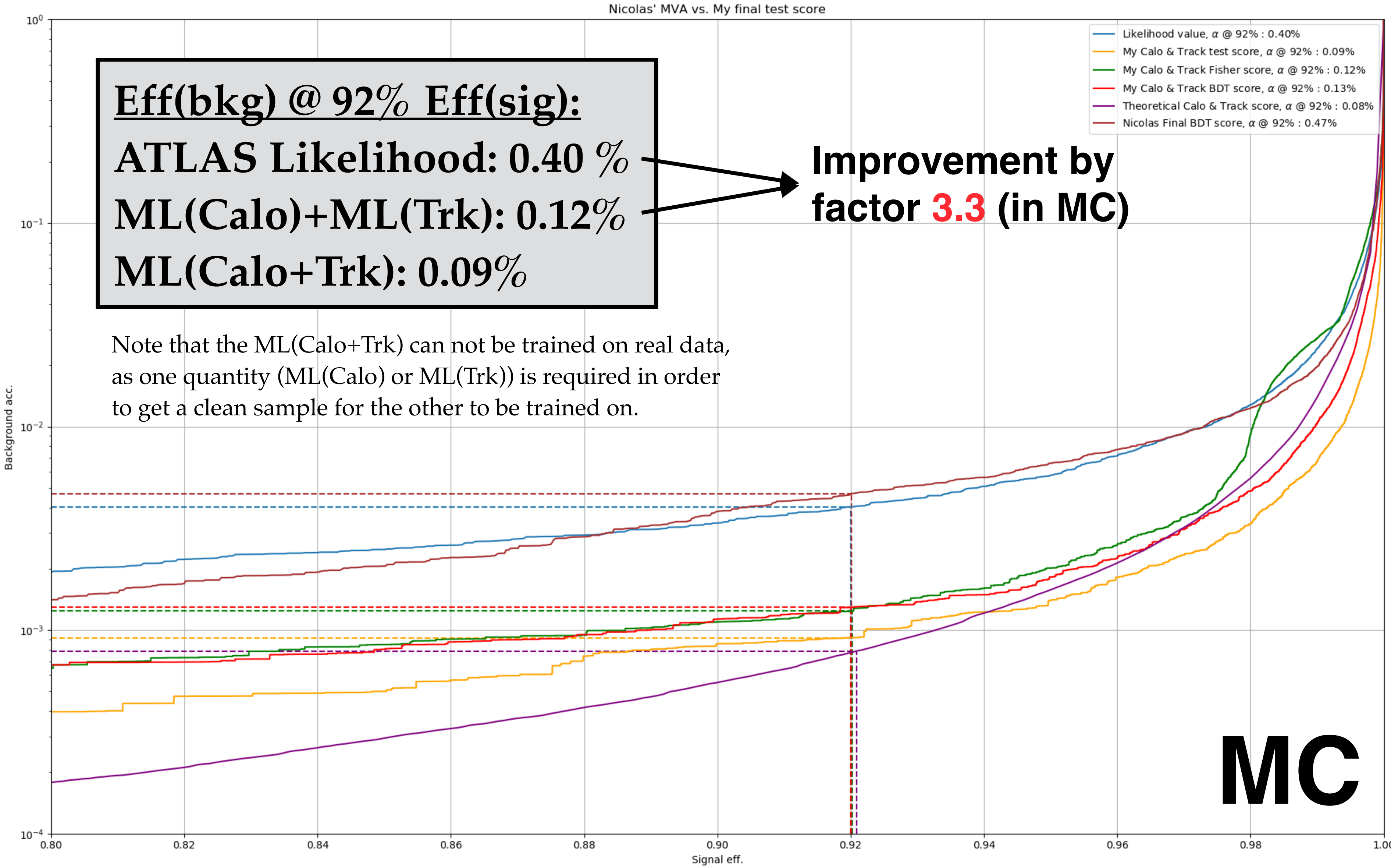
Purities after T&P

The electron probe purities for both signal and background are far from ideal!





Comparison of Combination

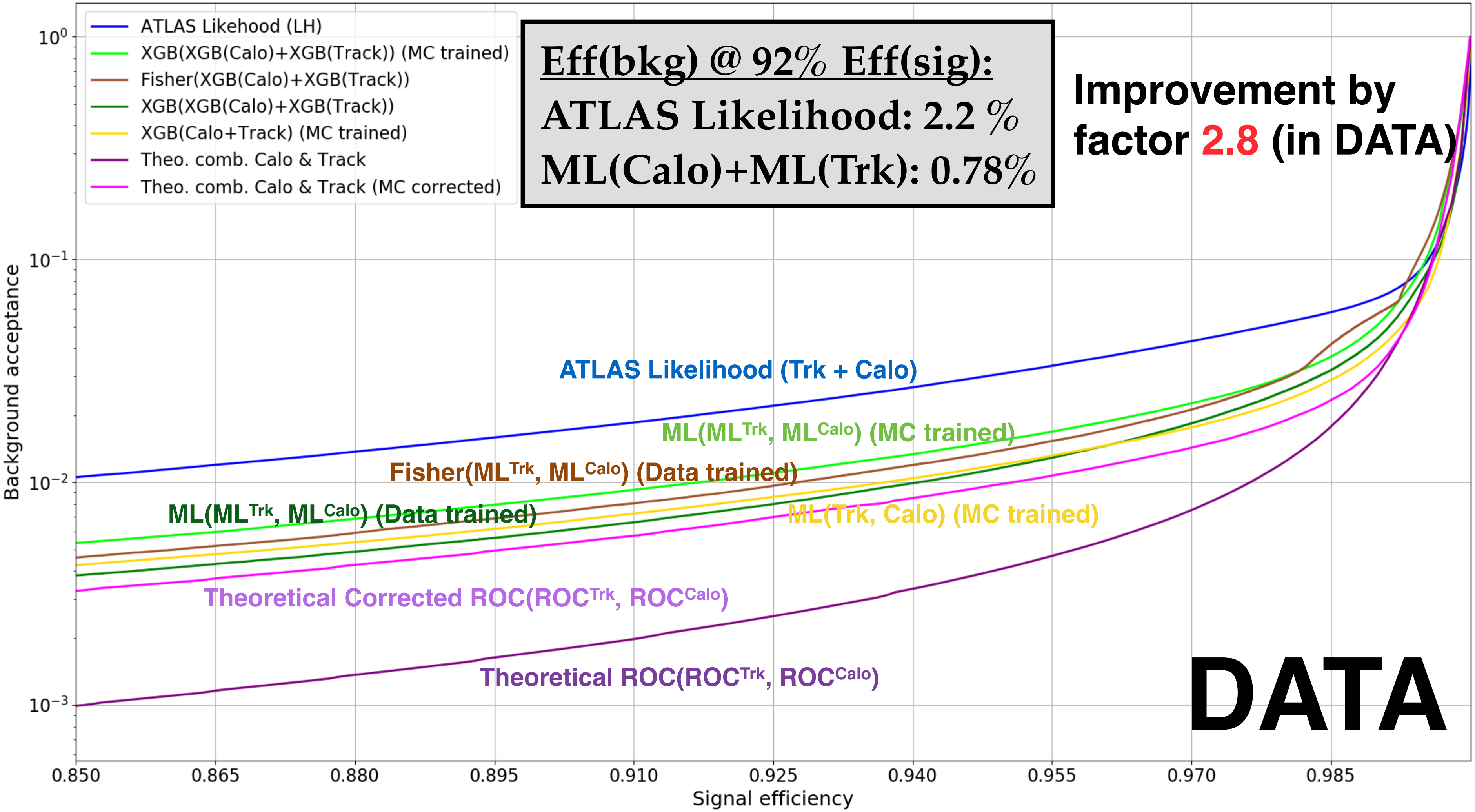


Eff(bkg) @ 92% Eff(sig):
ATLAS Likelihood: 0.40 %
ML(Calo)+ML(Trk): 0.12%
ML(Calo+Trk): 0.09%

Improvement by factor 3.3 (in MC)

Note that the ML(Calo+Trk) can not be trained on real data, as one quantity (ML(Calo) or ML(Trk)) is required in order to get a clean sample for the other to be trained on.

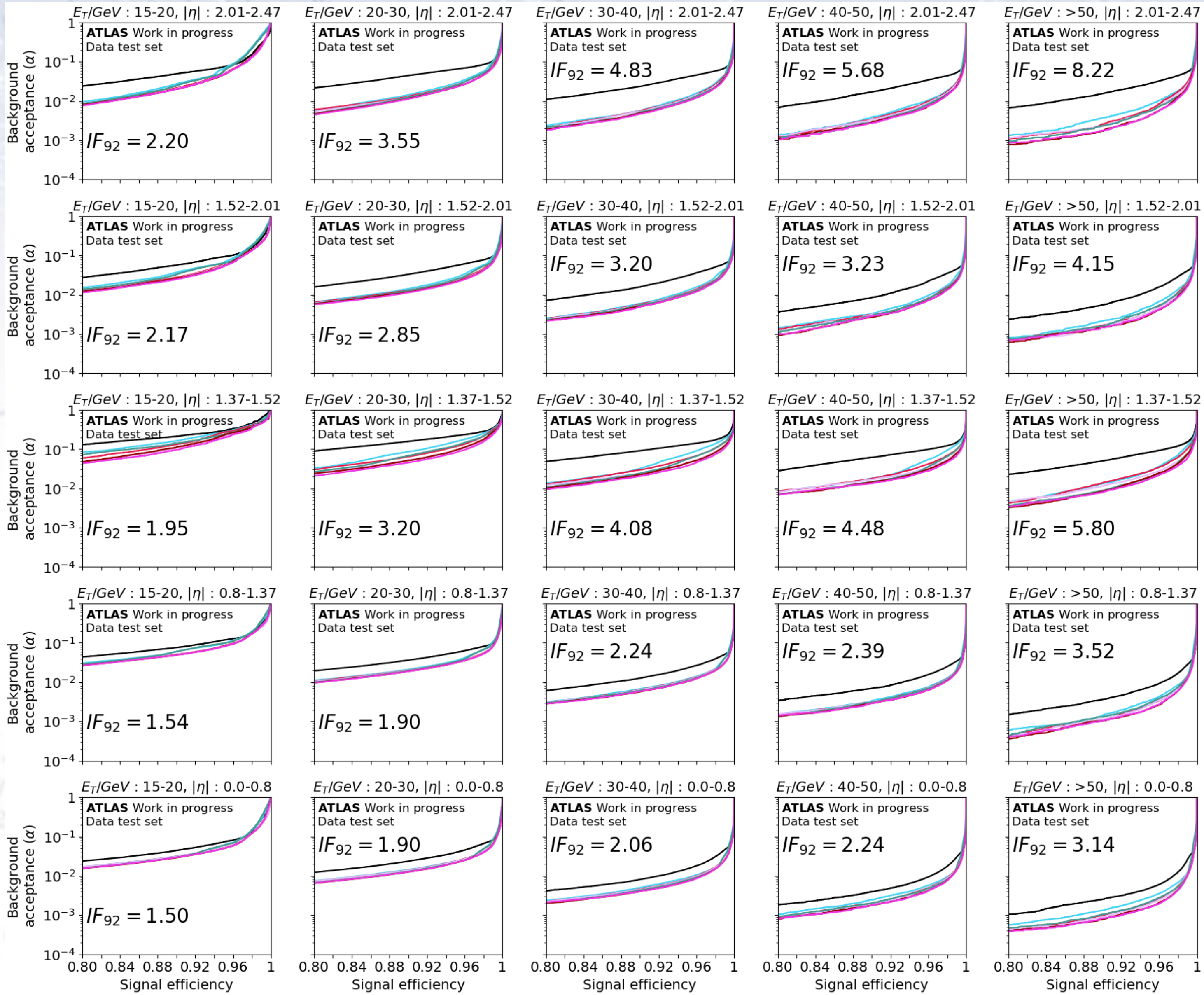
Performance of all methods



Electron PID on data

Performance is best in the **forward region** at **high energies**.

However, the later statement might be a result of determining performance with impure data (more so at lower energies).

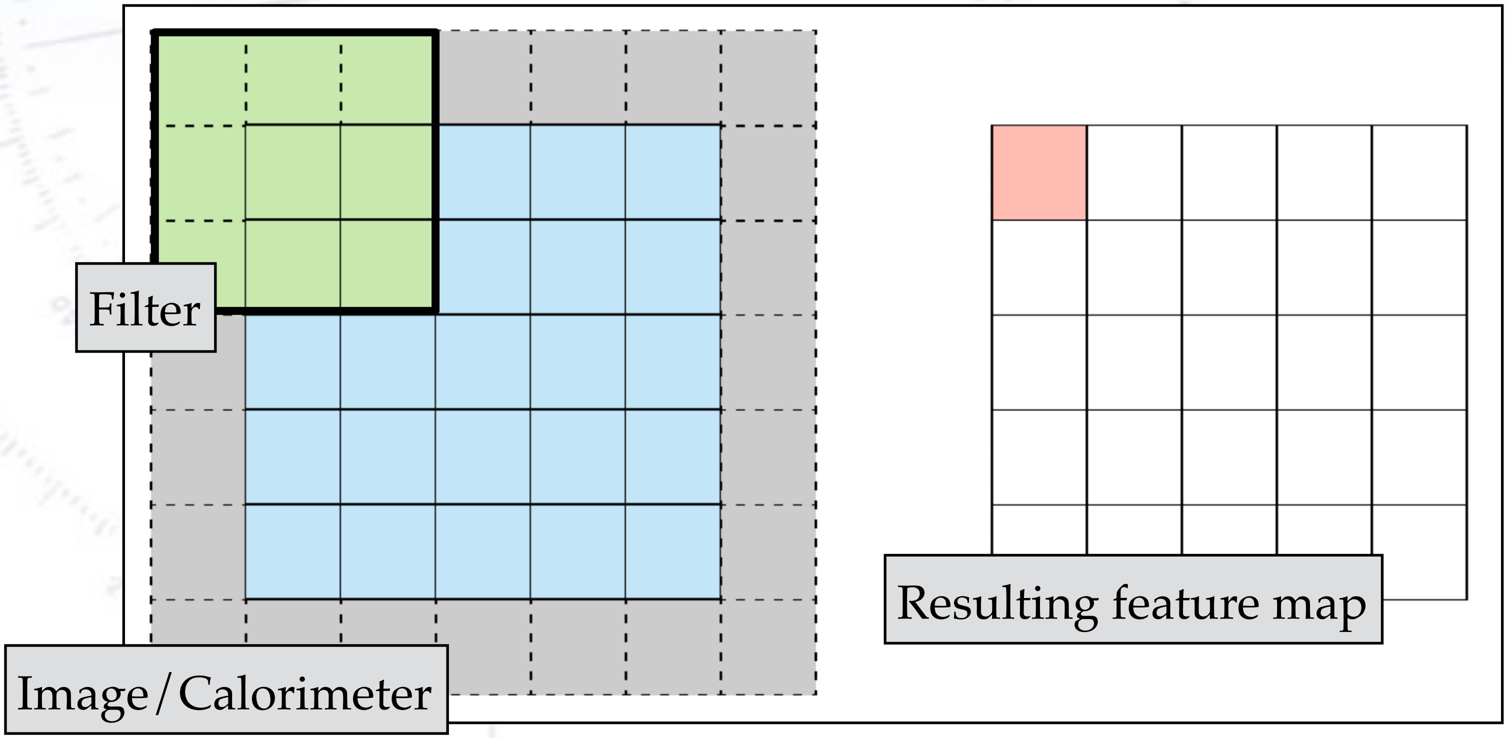


What is a CNN?

CNNs are a type of neural network, which works well with spatially dependent data (typically images). CNNs use parameter/weight sharing.

Multi-layered images (e.g. RGB or ATLAS calorimeter) are handled naturally.

A CNN works by sliding (small) filter across the image, outputting the convolution (inner product) of the filter and pixels covered.

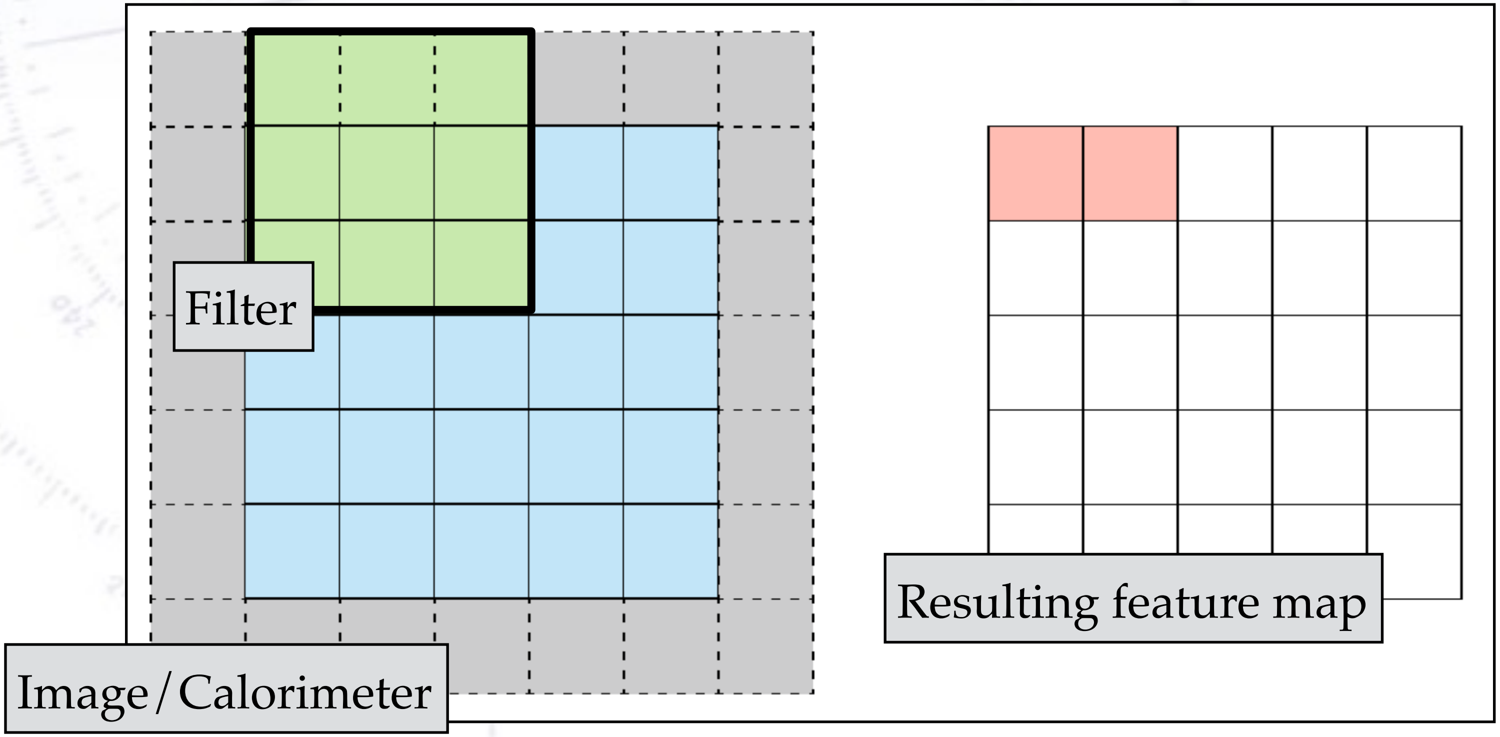


What is a CNN?

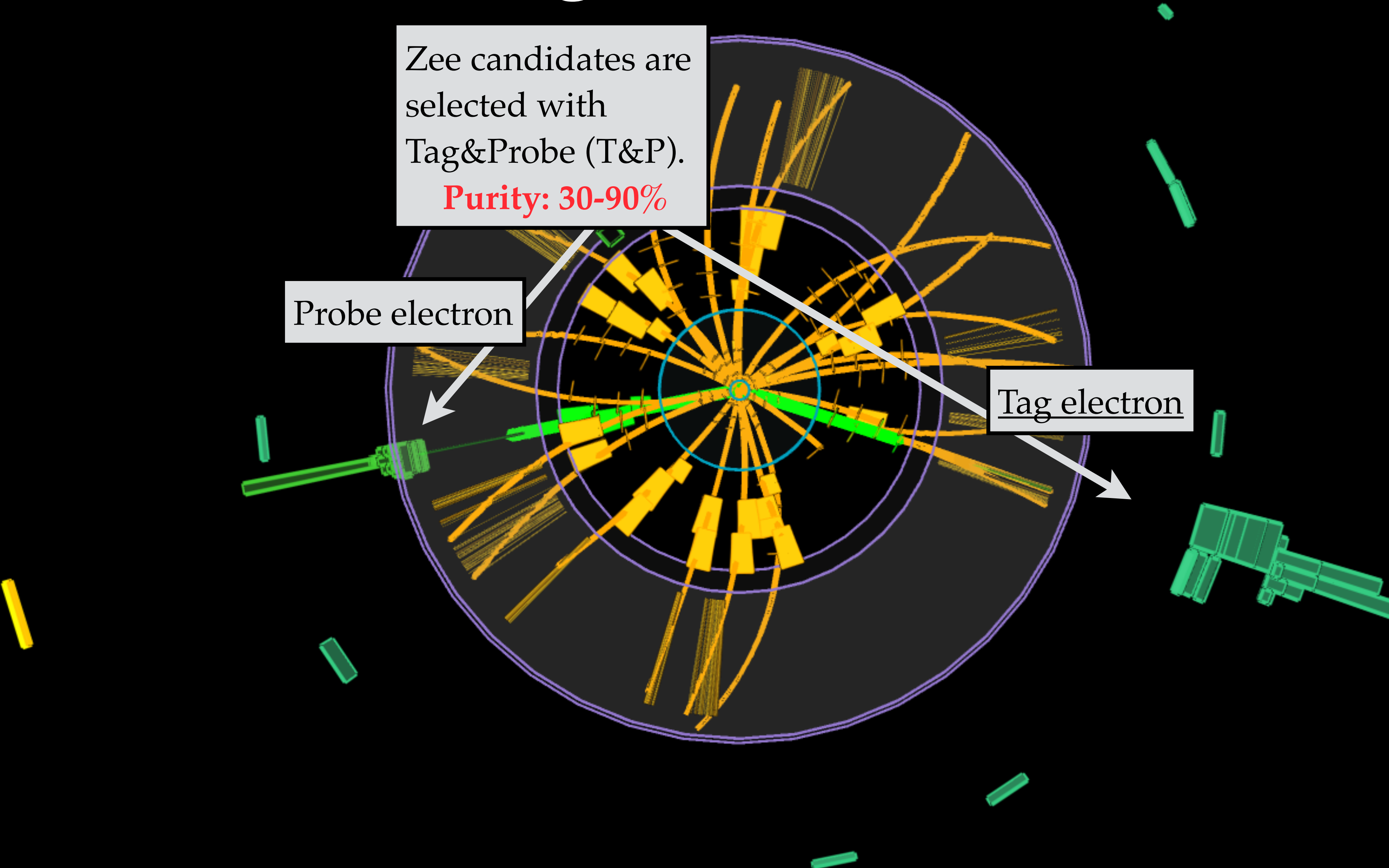
CNNs are a type of neural network, which works well with spatially dependent data (typically images). CNNs use parameter/weight sharing.

Multi-layered images (e.g. RGB or ATLAS calorimeter) are handled naturally.

A CNN works by sliding (small) filter across the image, outputting the convolution (inner product) of the filter and pixels covered.

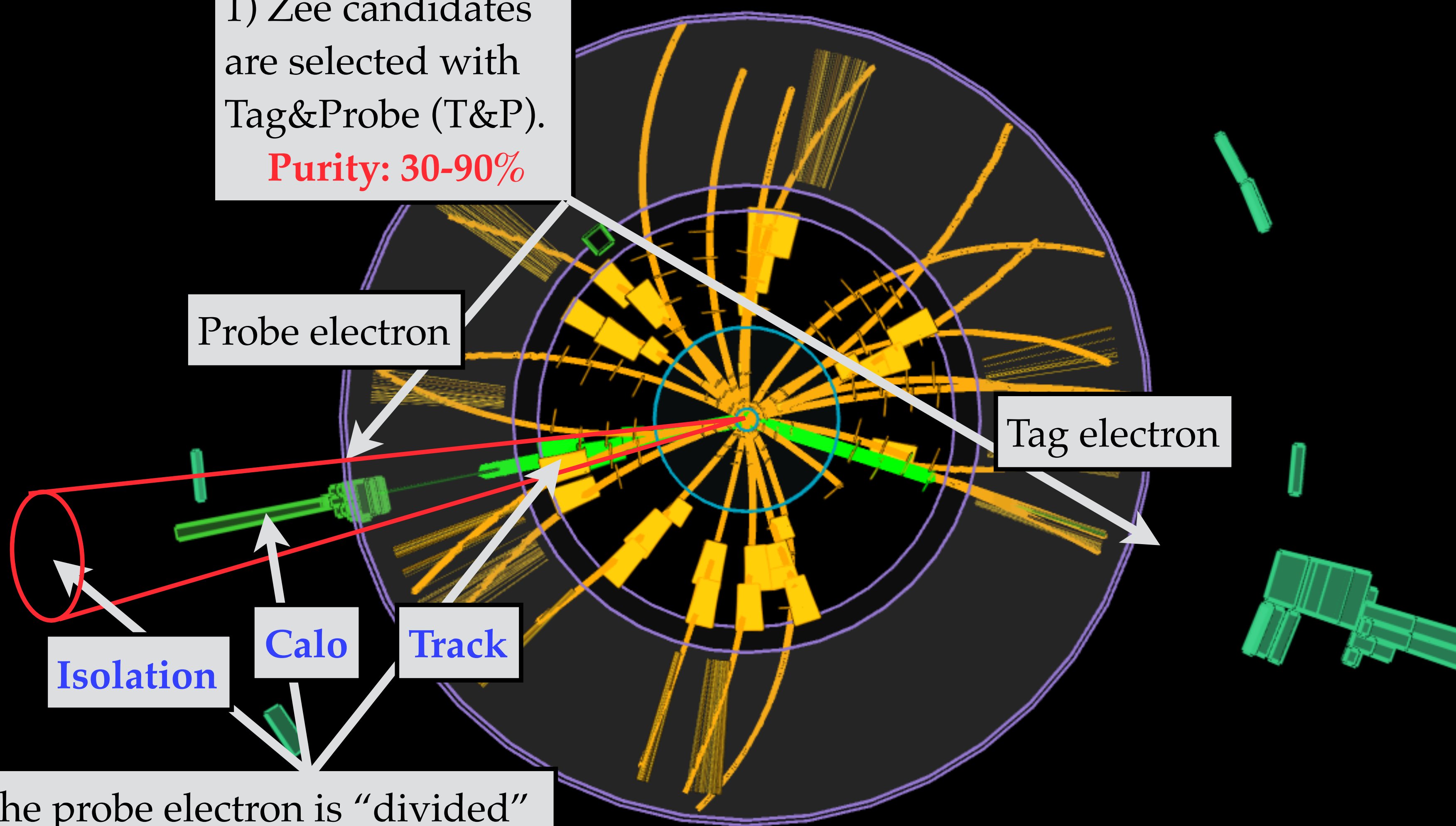


Tag & Probe


Zee candidates are selected with Tag&Probe (T&P).
Purity: 30-90%
Probe electron
Tag electron

The Idea: Extended Tag&Probe

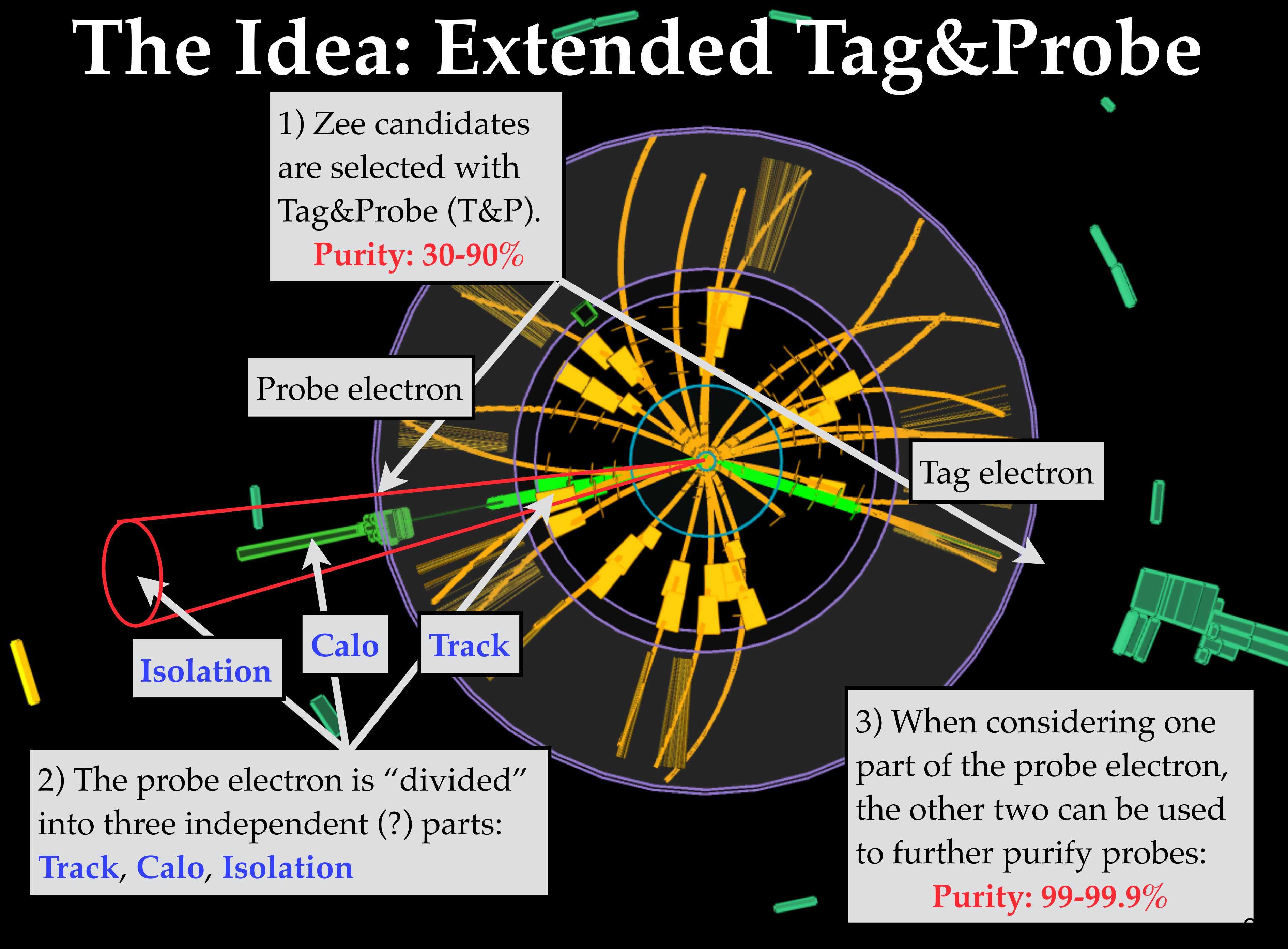
1) Zee candidates are selected with Tag&Probe (T&P).
Purity: 30-90%



2) The probe electron is “divided” into three independent (?) parts:
Track, **Calo**, **Isolation**

The Idea: Extended Tag&Probe

1) Zee candidates are selected with Tag&Probe (T&P).
Purity: 30-90%



2) The probe electron is "divided" into three independent (?) parts:
Track, **Calo**, **Isolation**

3) When considering one part of the probe electron, the other two can be used to further purify probes:
Purity: 99-99.9%