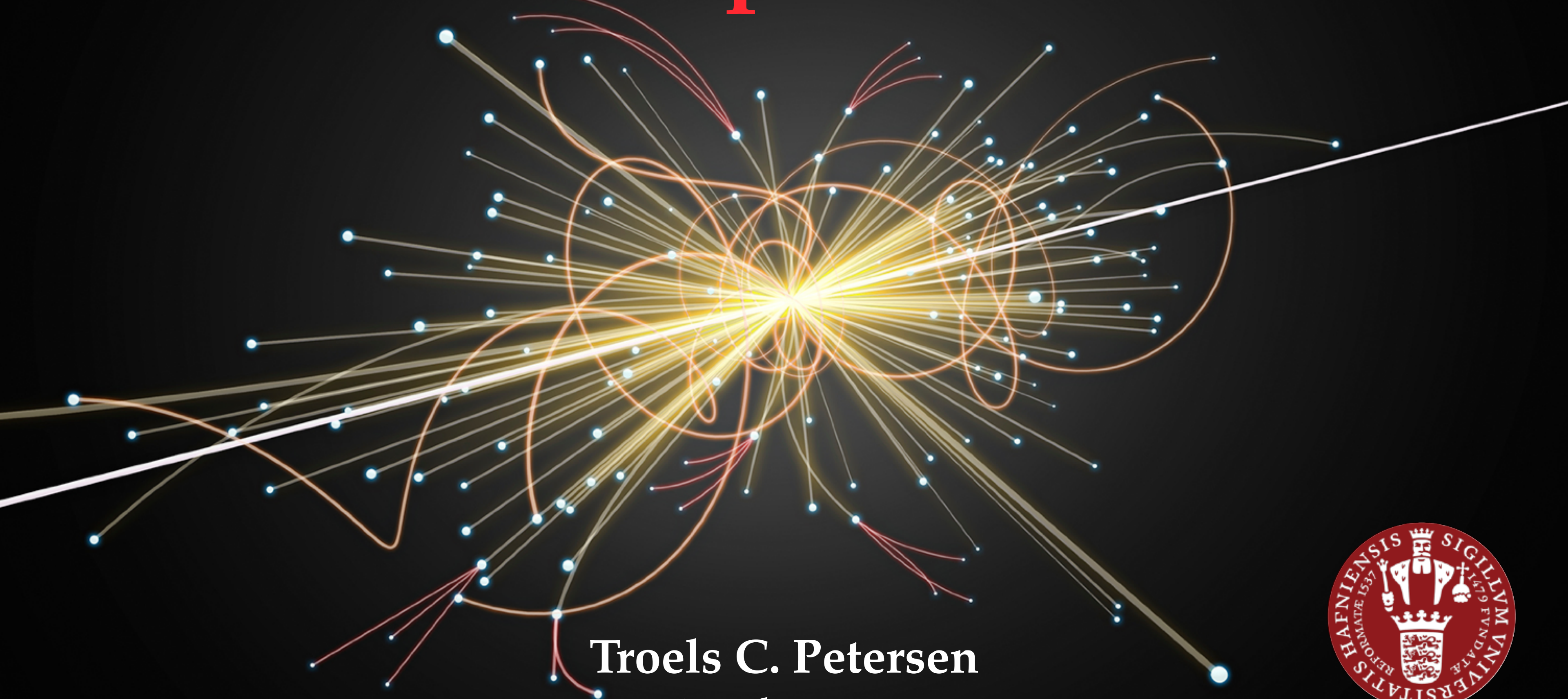
HAMLET
How to Apply Machine Learning to Experimental & Theoretical

ML

Past experiences



Troels C. Petersen
Niels Bohr Institute

On experience

Sentence:

"Experience is simply the name we give our mistakes",

[Oscar Wilde]

Lemma:

"I didn't fail. It was a learning experience",

[Anonymous]

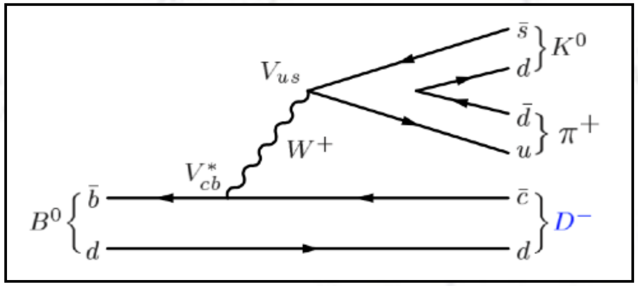
First encounters

On a dark and stormy night in 2001, PostDoc Andreas Hoecker called me into his office: "Troels, come and see this…"

It was a piece of Fortran code, that he had gotten in an Email:

It was a Neural Network!

For context, I was working on the BaBar experiment at SLAC, focusing on B to DKpi decays:




```
 PROGRAM TPK
! The TPK Algorithm
! Fortran 90 style
  IMPLICIT NONE
  INTEGER                     :: I
  REAL                        :: Y
  REAL, DIMENSION(0:10)       :: A
  READ (*,*) A
  DO I = 10, 0, -1          ! Backwards
    Y = FUN(A(I))
    IF ( Y < 400.0 ) THEN
      WRITE(*,*) I, Y
    ELSE
      WRITE(*,*) I, ' Too large'
    END IF
  END DO
  CONTAINS                   ! Local function
    FUNCTION FUN(T)
      REAL  :: FUN
      REAL, INTENT(IN)  :: T
      FUN = SQRT(ABS(T)) + 5.0*T**3
    END FUNCTION FUN
  END PROGRAM TPK
```

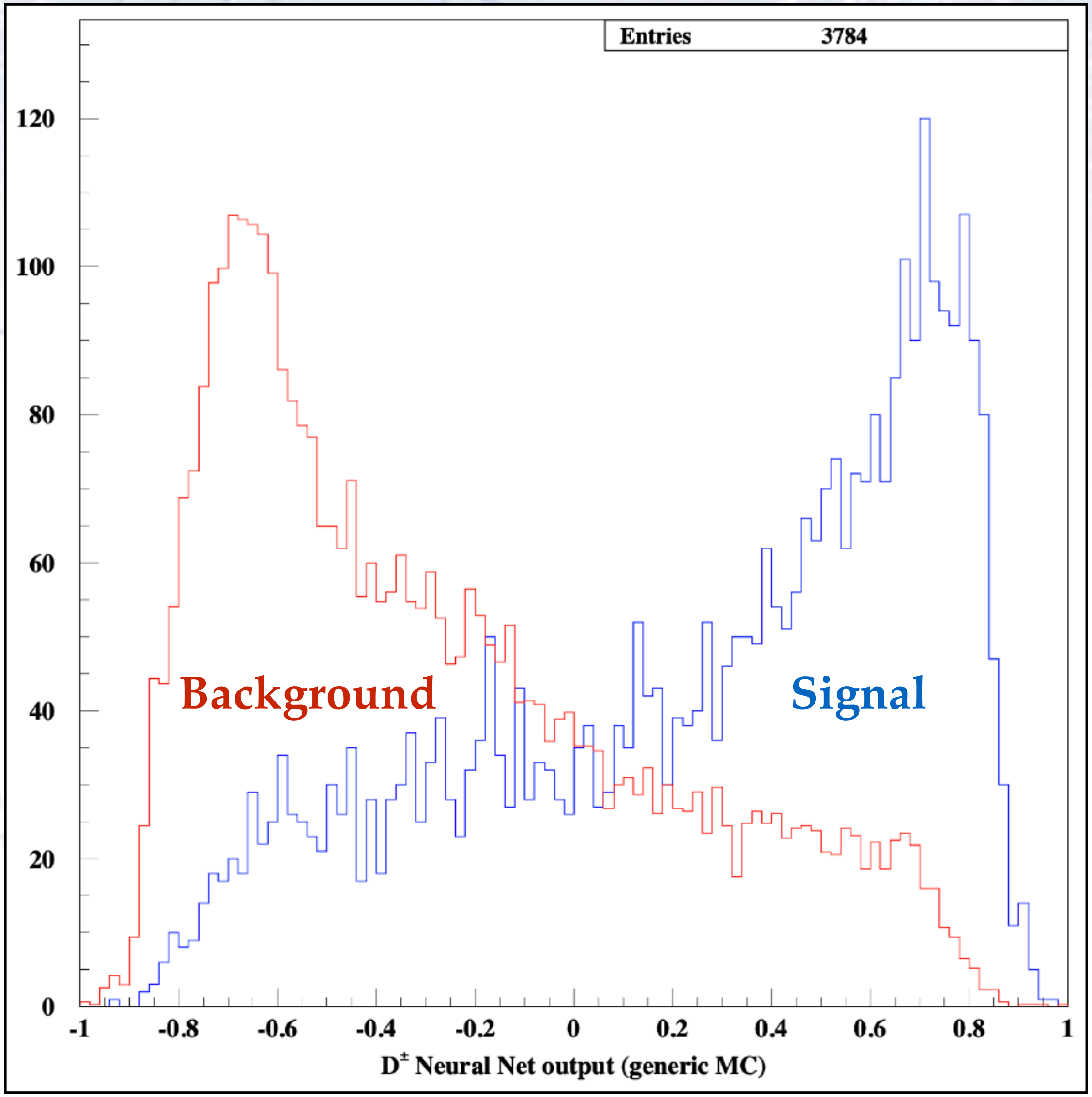
First encounters

Not having any experience with ML, I did a lot of mistakes:

- No description of architecture!
- No HP optimisation.
- No check of data-MC correspondence.
- No loss / epoch plot.

I had not thought of any way to cross check and calibrate the output.

But… simply throwing myself at it was a great experience to build on.

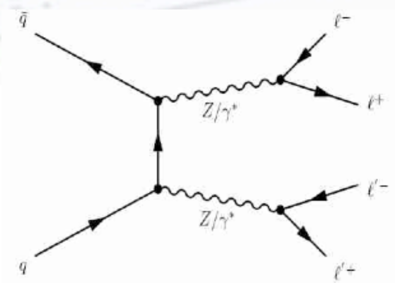


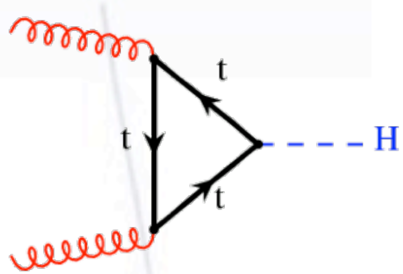
Higgs Search/Discovery

Motivation

Problem:
Given a number of clean ZZ events,
determine if they are Higgs or SM diboson events!

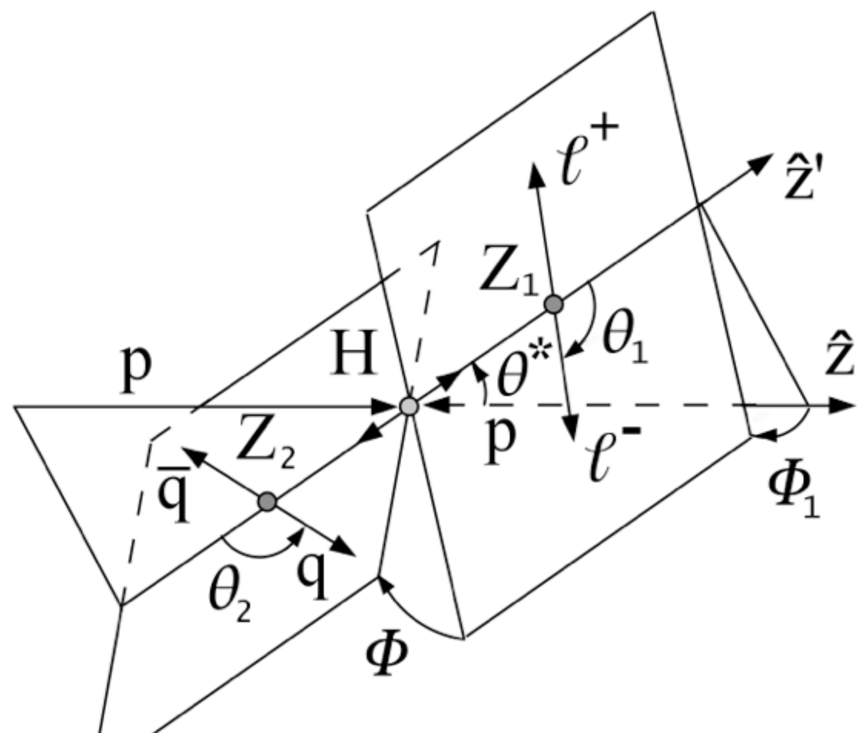
Possible solution:
Since Higgses are produced quite differently then SM diboson ZZ,
their angular distributions differ!





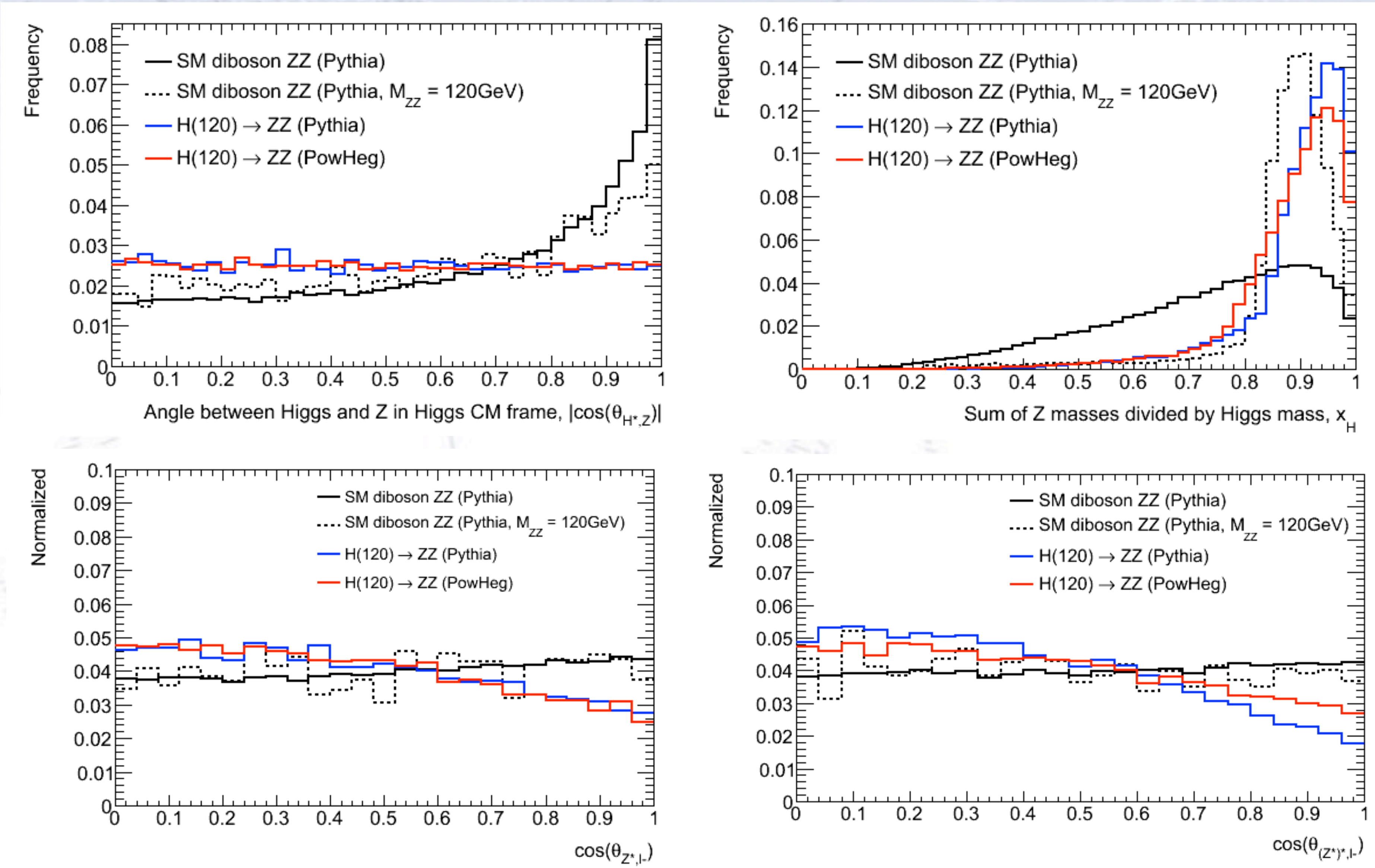
Variables available/used:

- Higgs rapidity
- Angle Z to Higgs in Higgs CM
- Angle lep- to Z in Z CM
- Angle lep- to Z* in Z* CM
- Fraction of mZ+mZ* to mHiggs

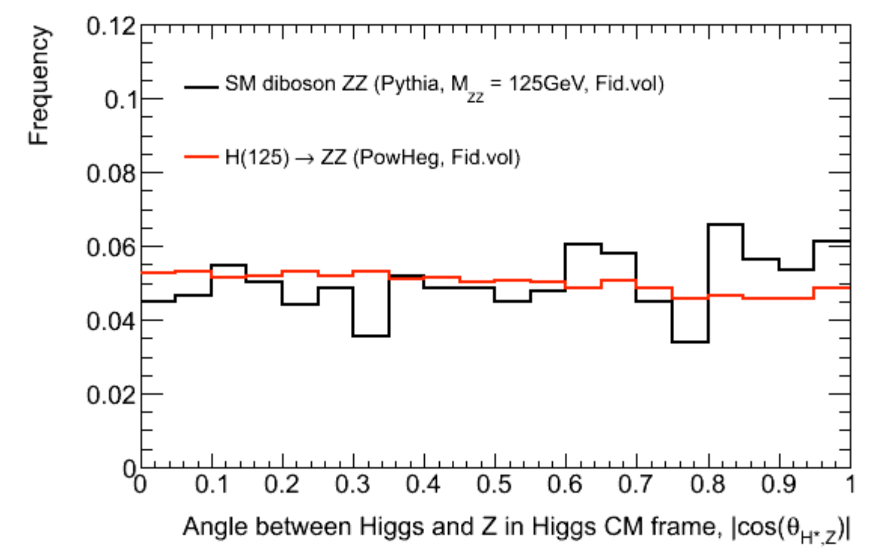


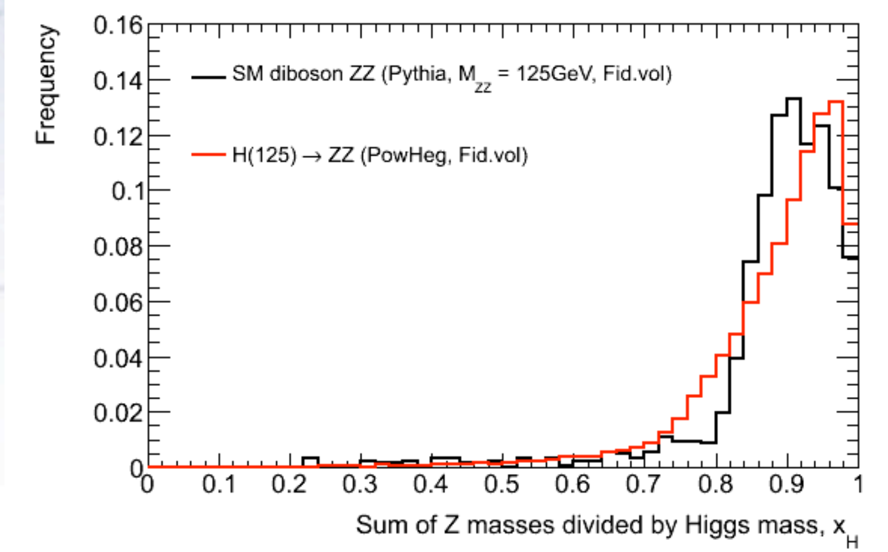
Note: H denotes the ZZ system, Higgs or not!

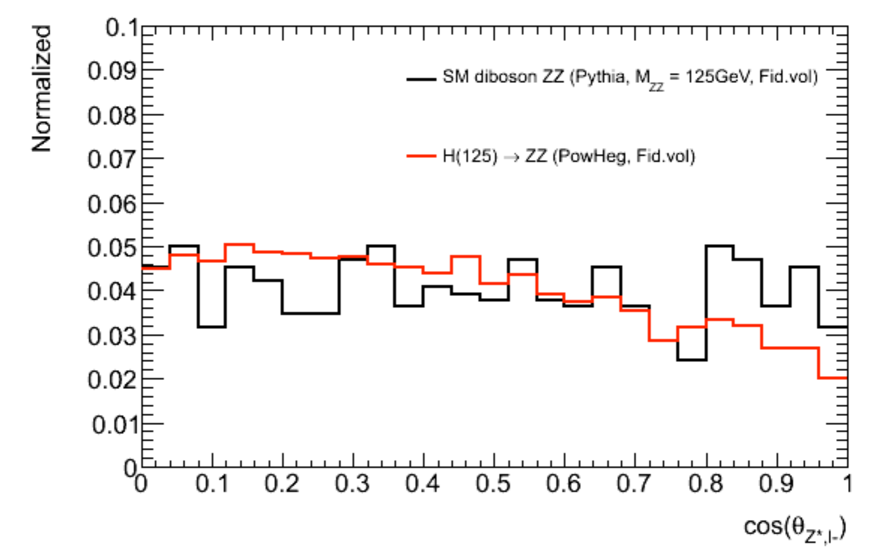
Generator level comparison

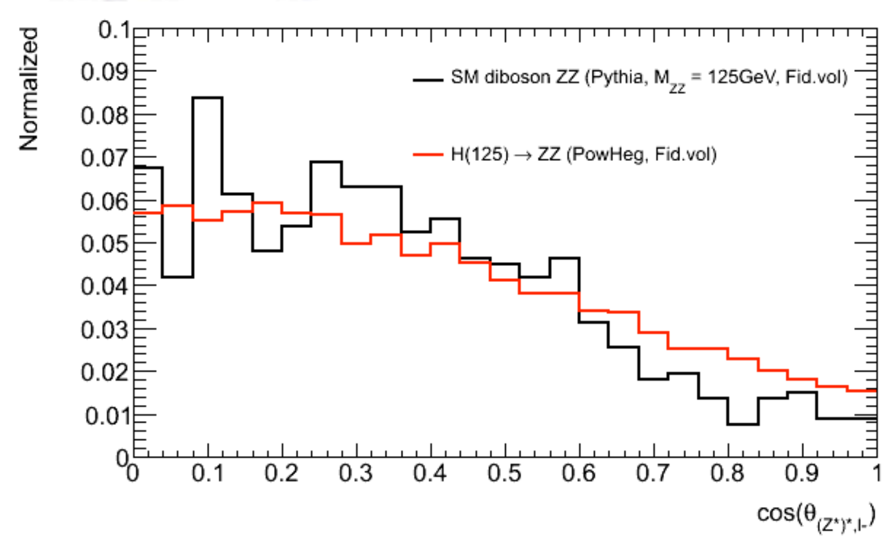


After fiducial requirements



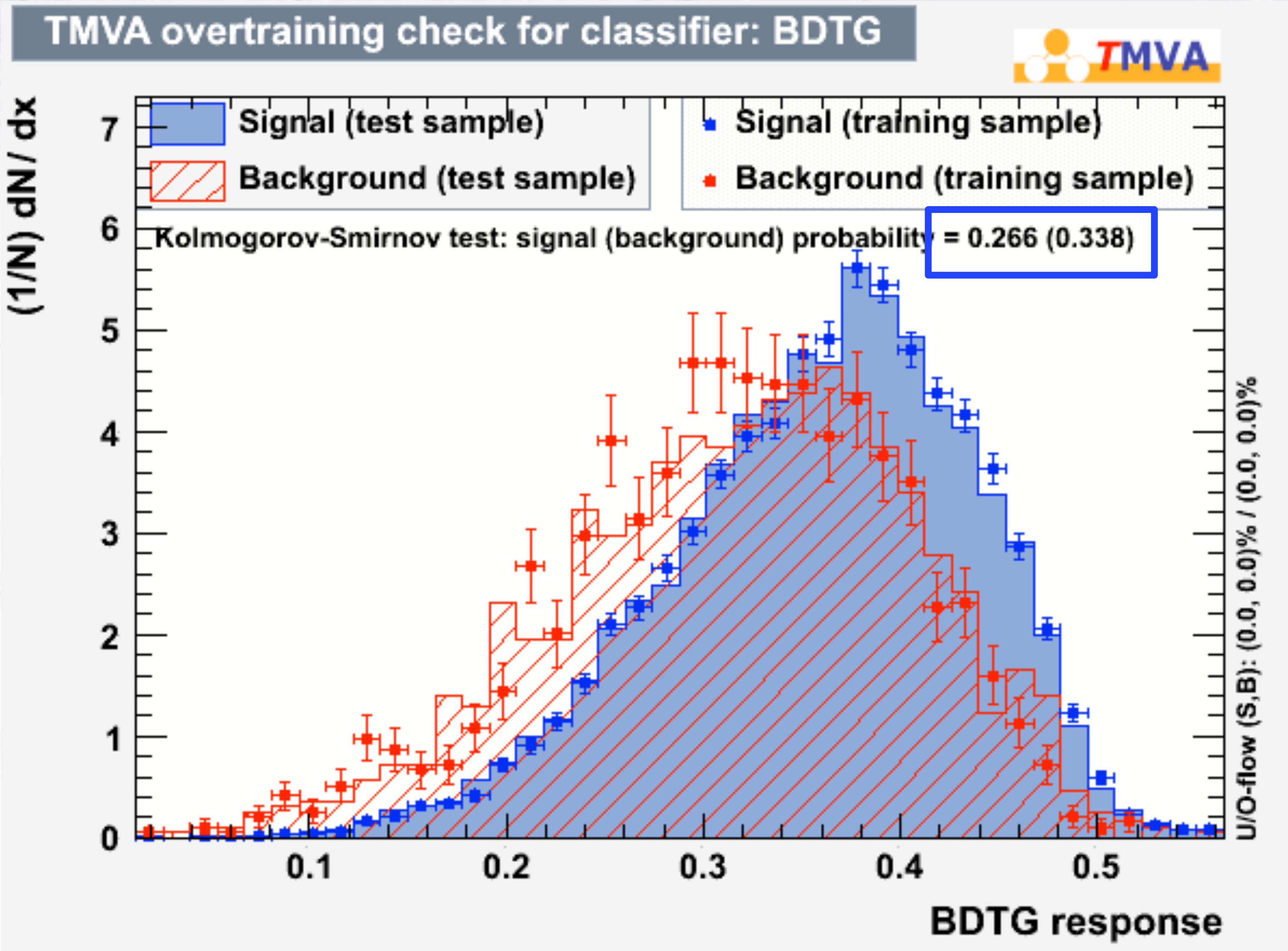




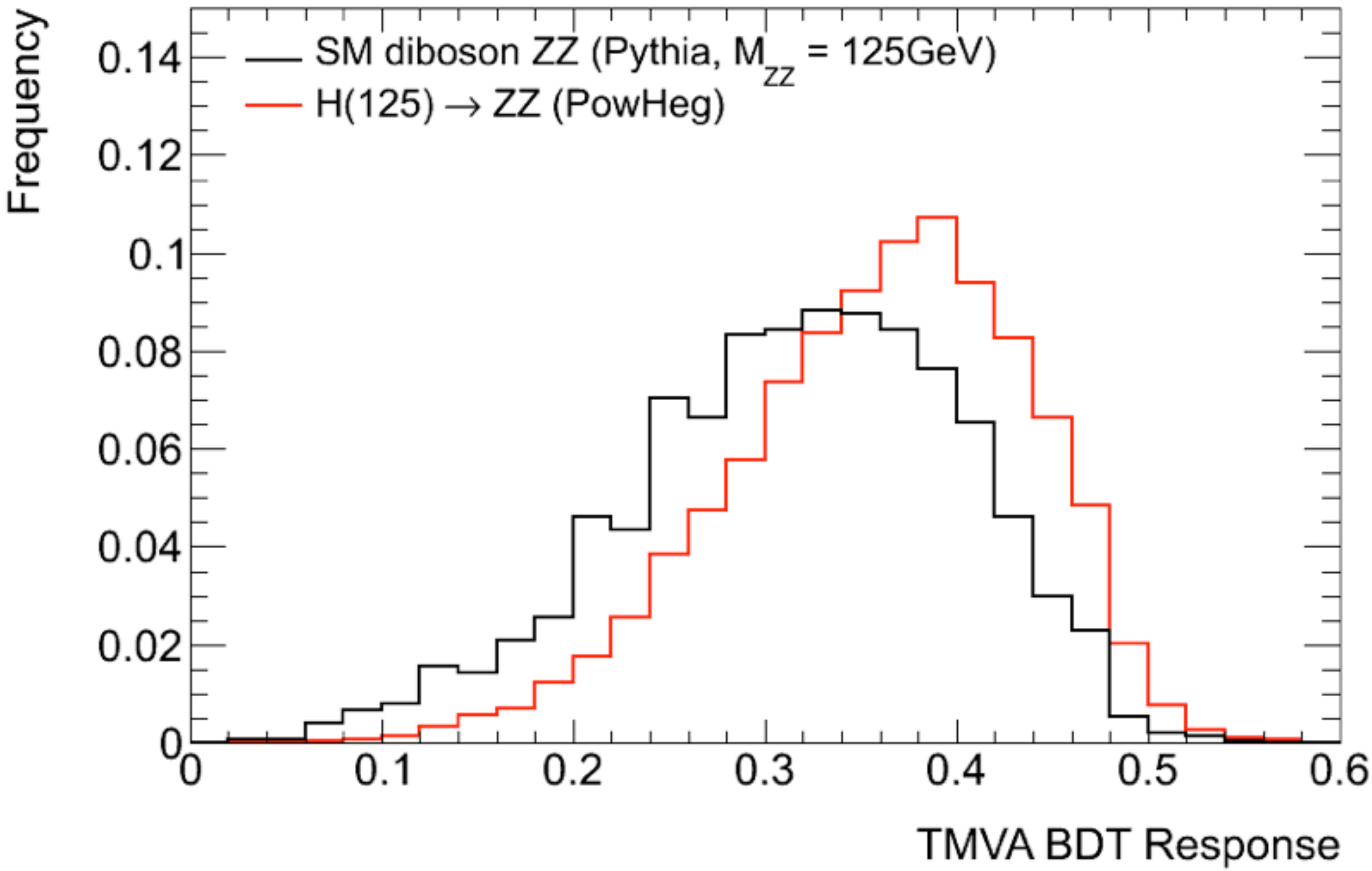


Combining variables

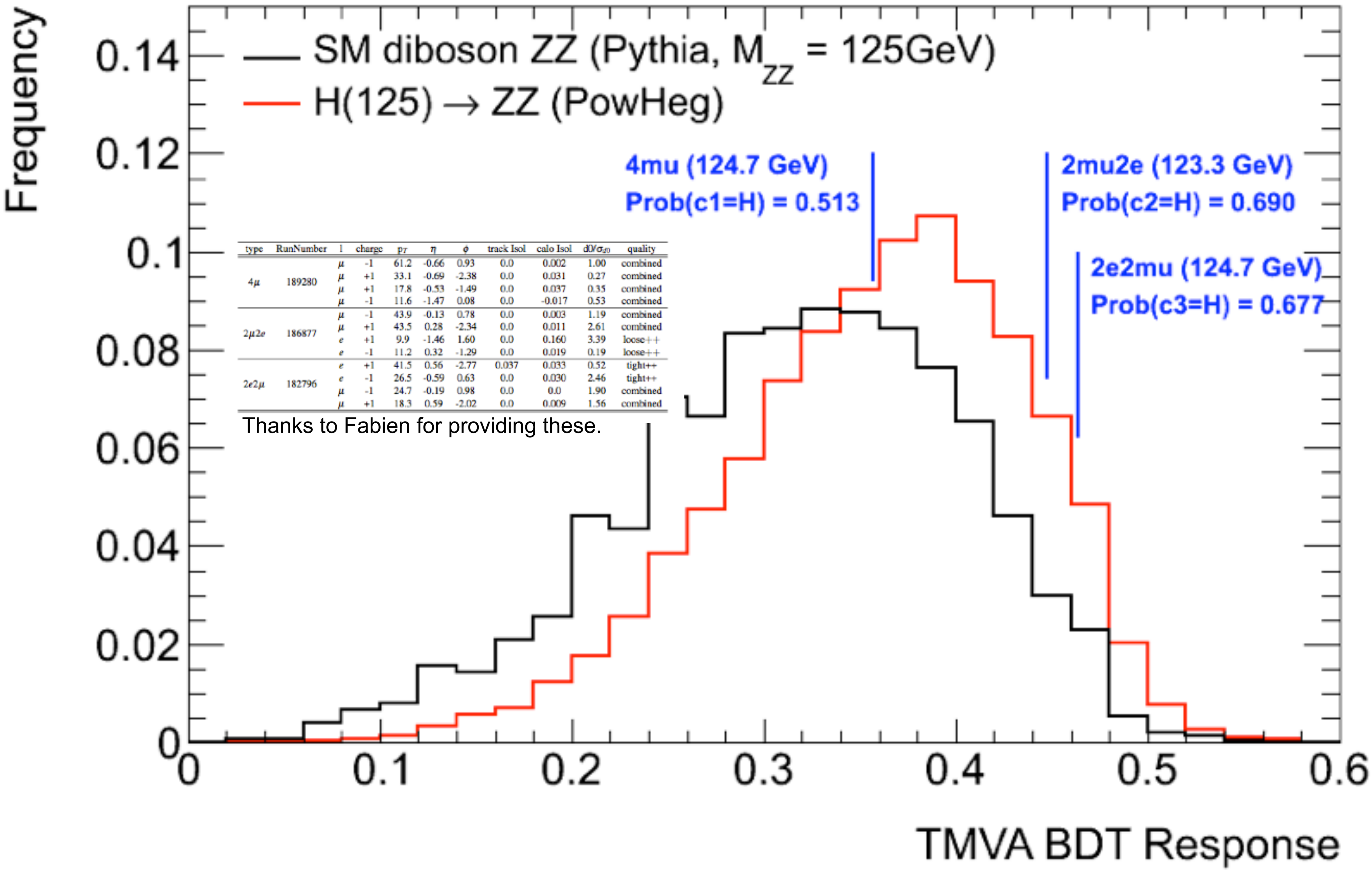
Using the 5 variables (i.e. including rapidity) in a BDT (100 trees, 4 nodes):



Combined angular variable



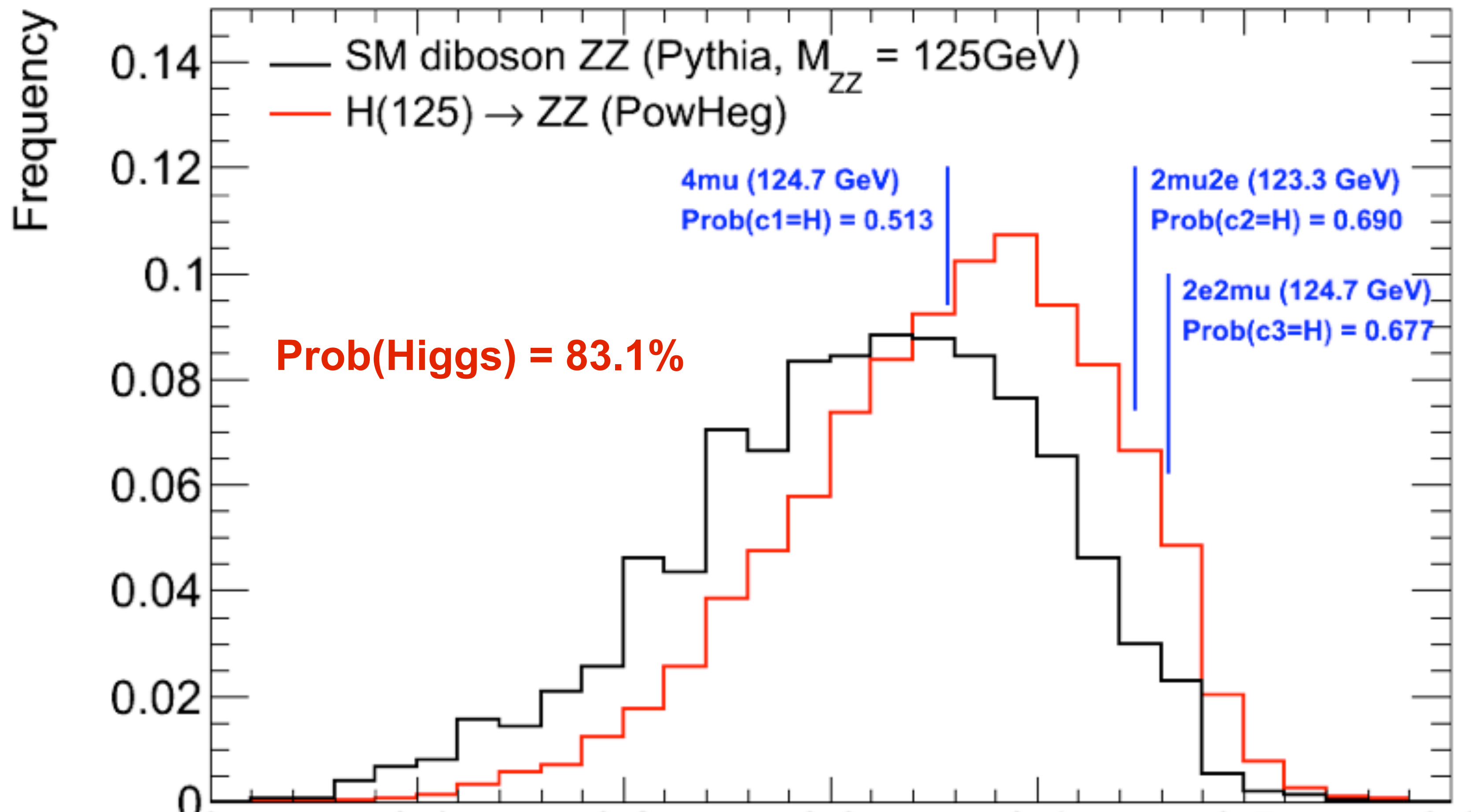
Combined angular variable



type	RunNumber	l	charge	p_T	η	ϕ	track Isol	calo Isol	$d0/\sigma_{d0}$	quality
4μ	189280	μ	-1	61.2	-0.66	0.93	0.0	0.002	1.00	combined
		μ	+1	33.1	-0.69	-2.38	0.0	0.031	0.27	combined
		μ	+1	17.8	-0.53	-1.49	0.0	0.037	0.35	combined
		μ	-1	11.6	-1.47	0.08	0.0	-0.017	0.53	combined
$2\mu 2e$	186877	μ	-1	43.9	-0.13	0.78	0.0	0.003	1.19	combined
		μ	+1	43.5	0.28	-2.34	0.0	0.011	2.61	combined
		e	+1	9.9	-1.46	1.60	0.0	0.160	3.39	loose++
		e	-1	11.2	0.32	-1.29	0.0	0.019	0.19	loose++
$2e2\mu$	182796	e	+1	41.5	0.56	-2.77	0.037	0.033	0.52	tight++
		e	-1	26.5	-0.59	0.63	0.0	0.030	2.46	tight++
		μ	-1	24.7	-0.19	0.98	0.0	0.0	1.90	combined
		μ	+1	18.3	0.59	-2.02	0.0	0.009	1.56	combined

Thanks to Fabien for providing these.

Combined angular variable

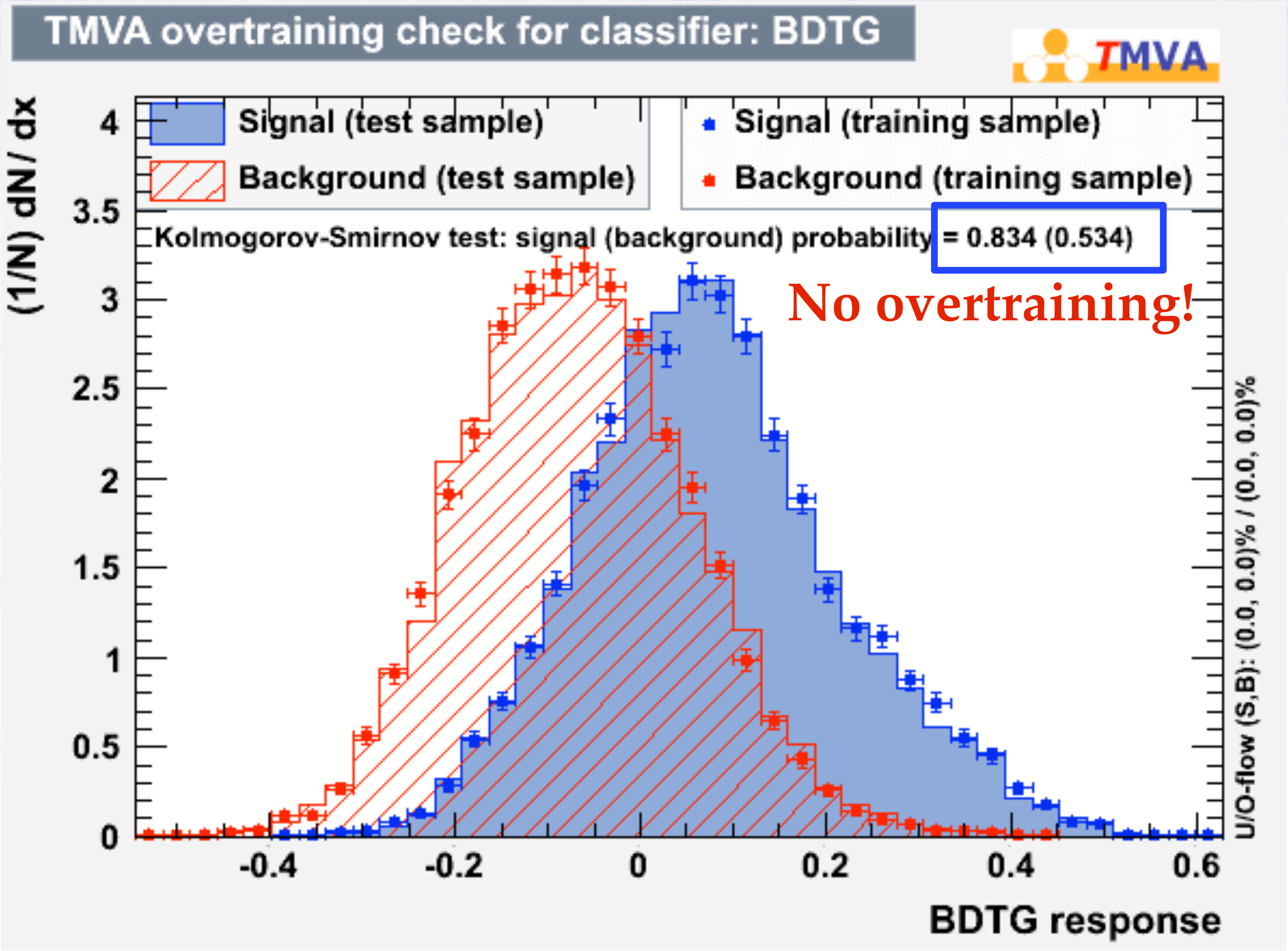


Conclusion:

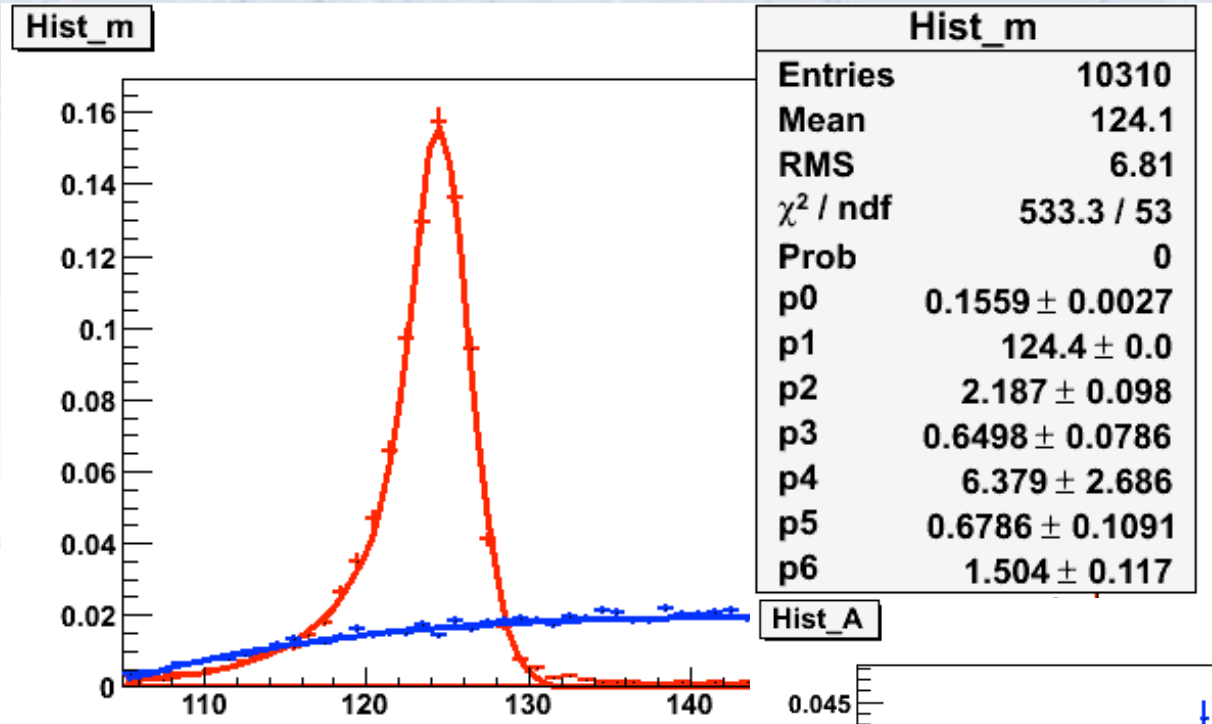
The 3 ZZ candidates at 125 GeV are more Higgs than SM dibosons like!

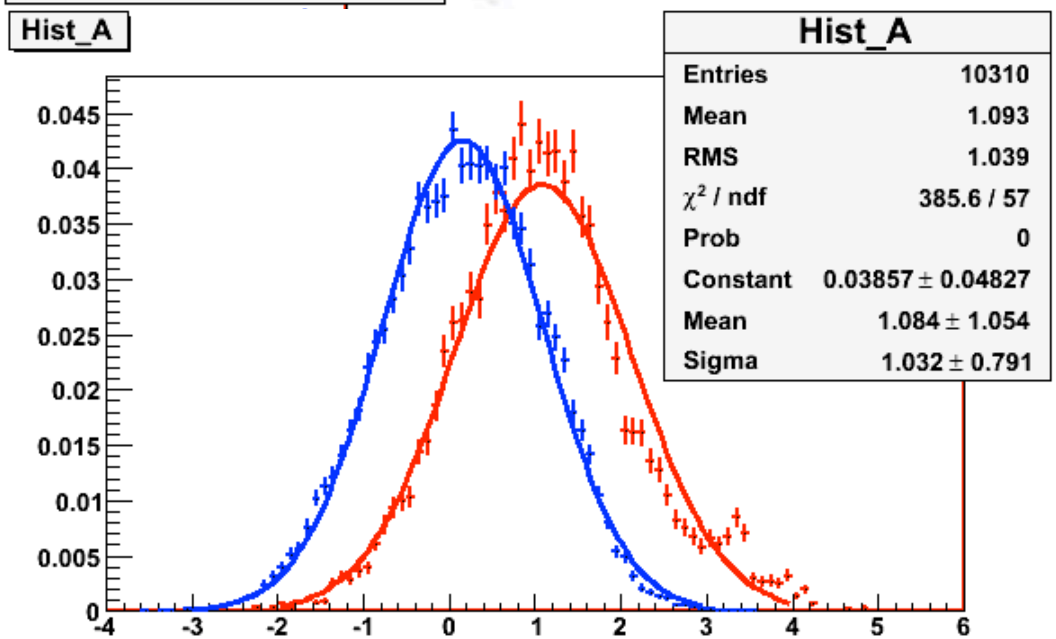
Check for overtraining

Using 9 variables in a BDT (200 trees, 4 nodes) and checking for overtraining:

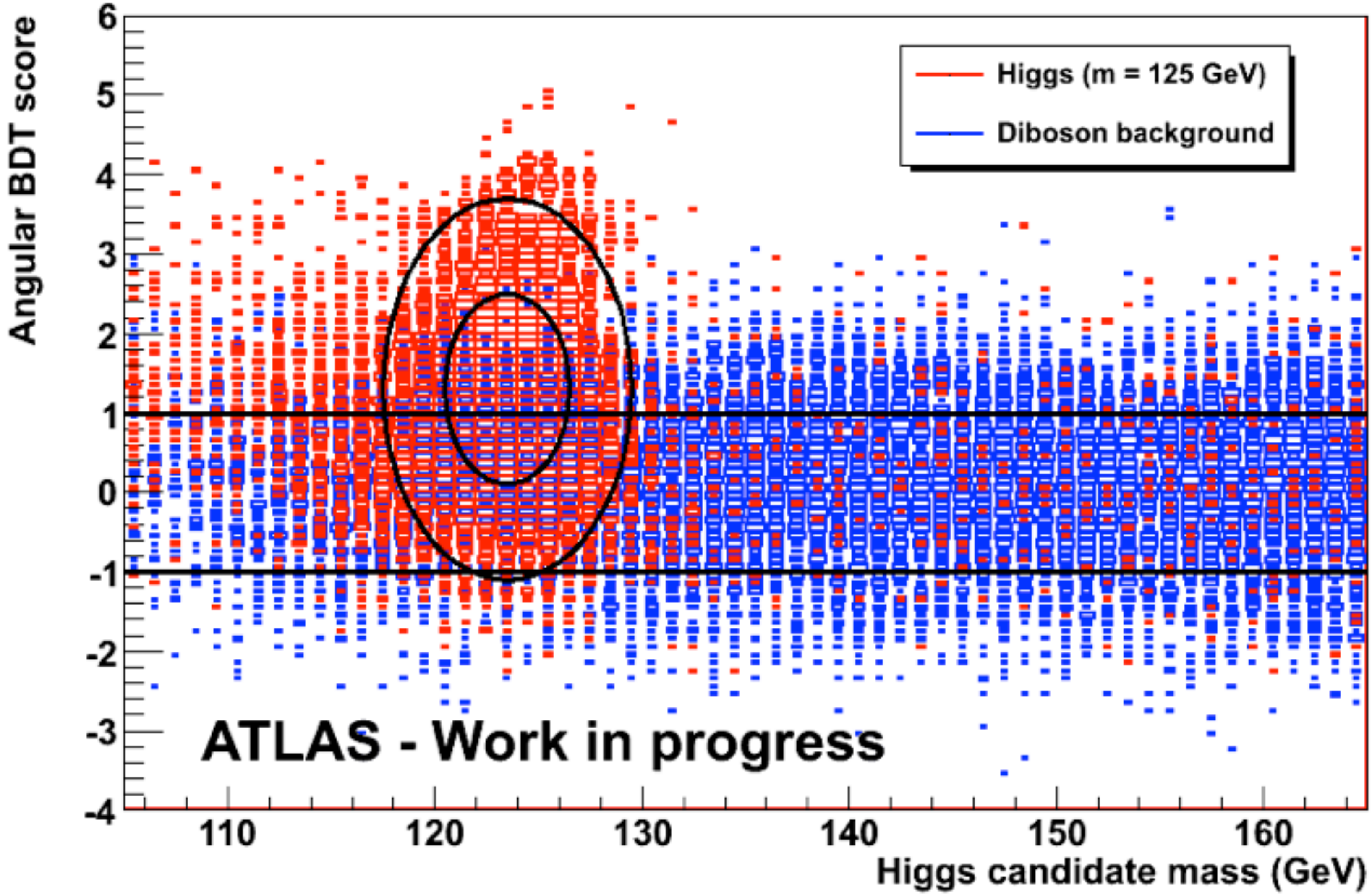


PDFs used in likelihood

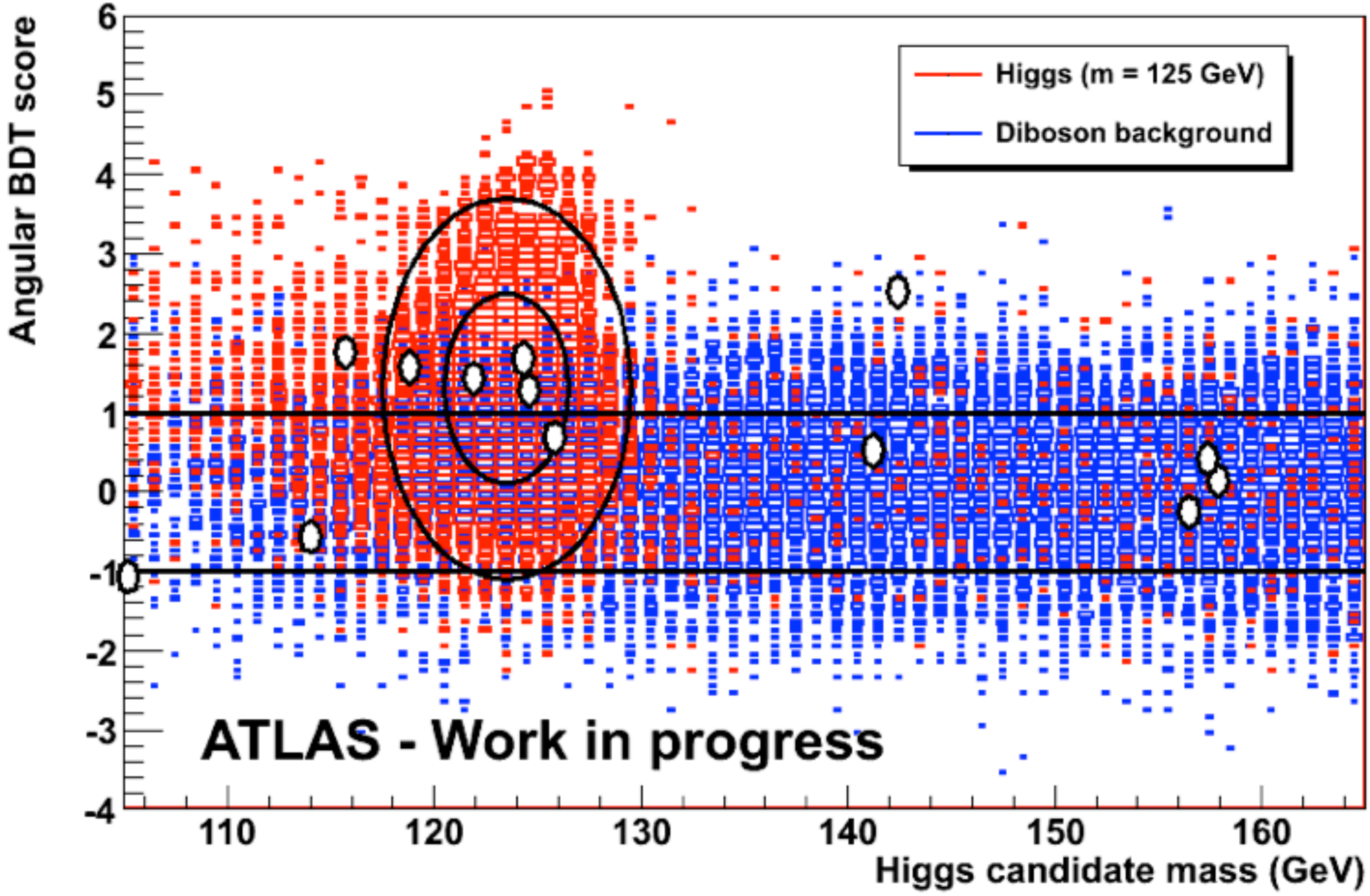




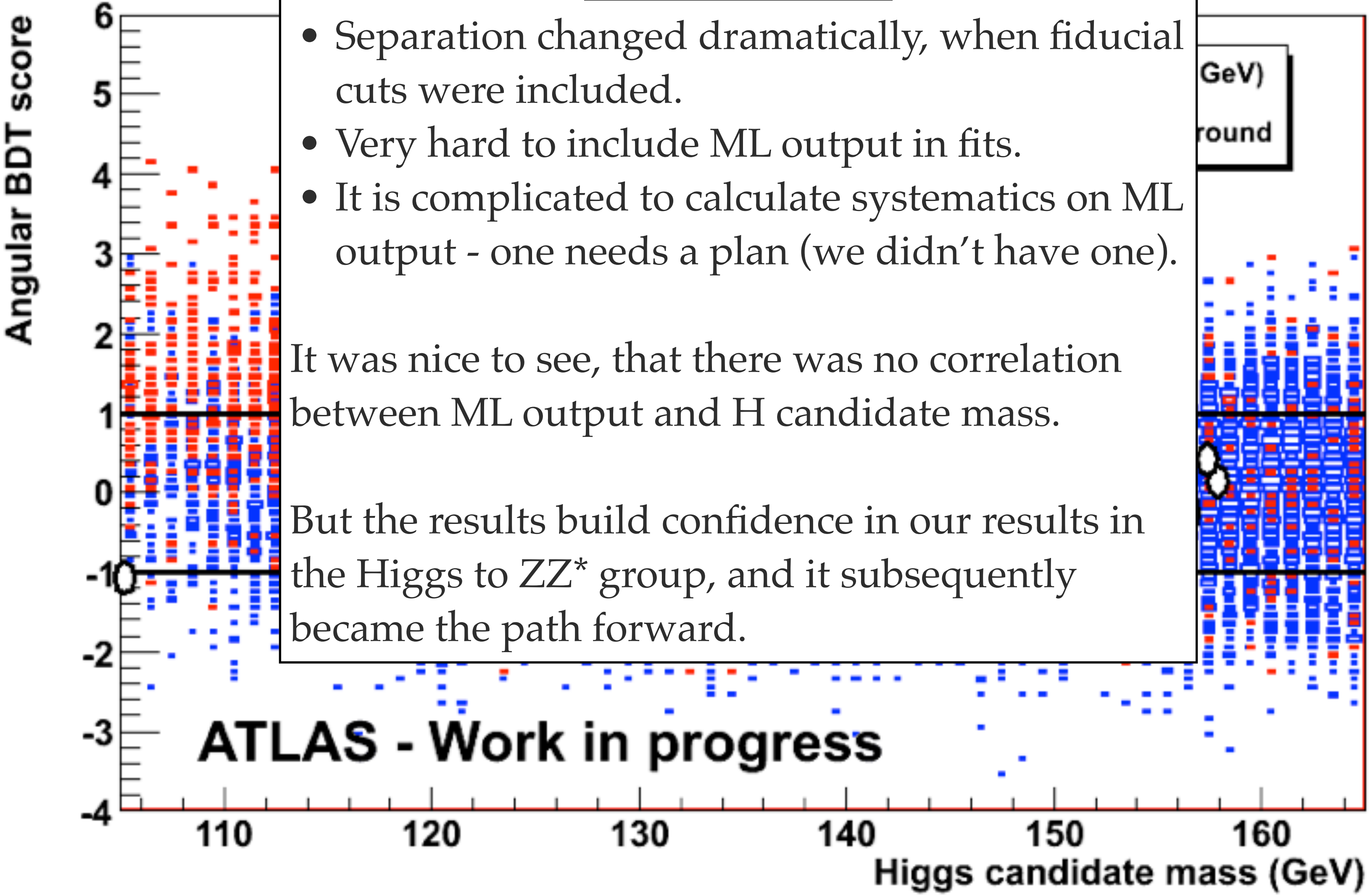
PDFs used in likelihood



PDFs used in likelihood



PDFs used in likelihood



<u>Lessons Learned:</u>

- Separation changed dramatically, when fiducial cuts were included.
- Very hard to include ML output in fits.
- It is complicated to calculate systematics on ML output - one needs a plan (we didn't have one).

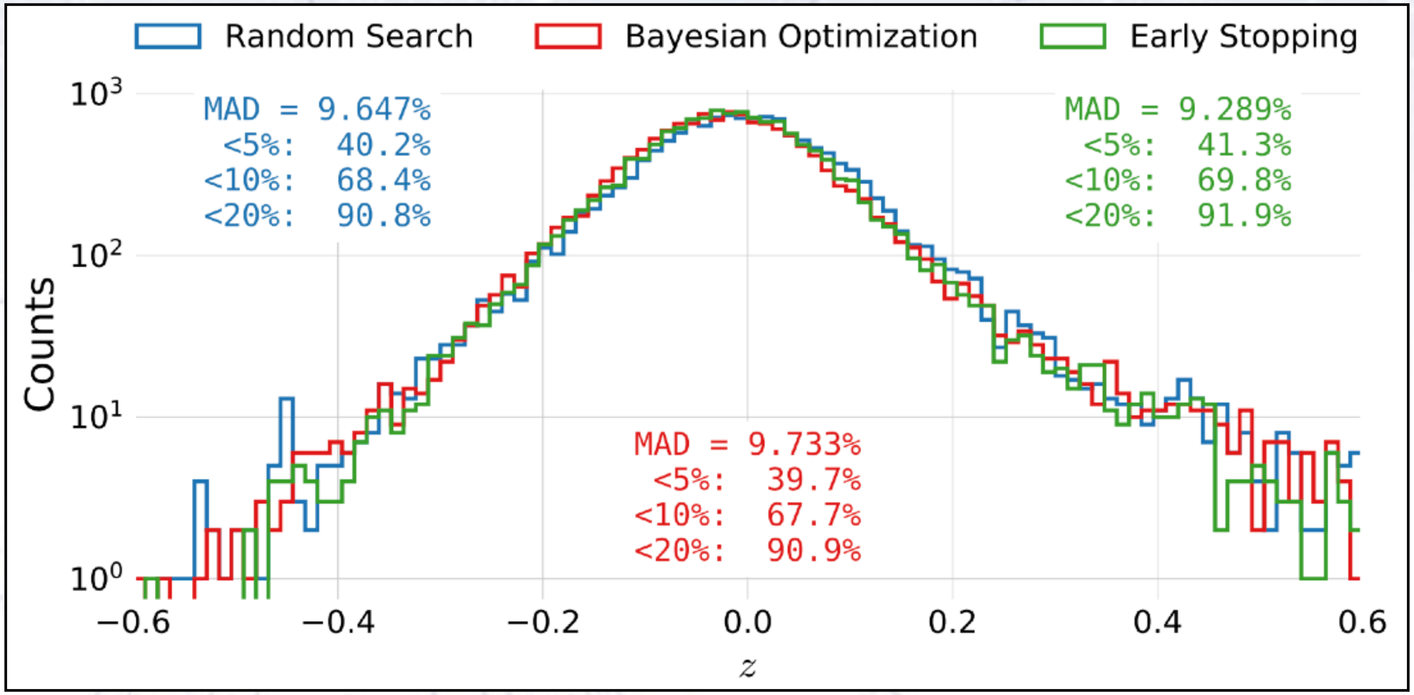
It was nice to see, that there was no correlation between ML output and H candidate mass.

But the results build confidence in our results in the Higgs to ZZ* group, and it subsequently became the path forward.

Housing Prices

Estimating Housing Prices

Slightly by coincidence, we got in contact with BoligSiden and collaborated.
They had data on 0.5M house sales 2008-2019 (90+% of all).

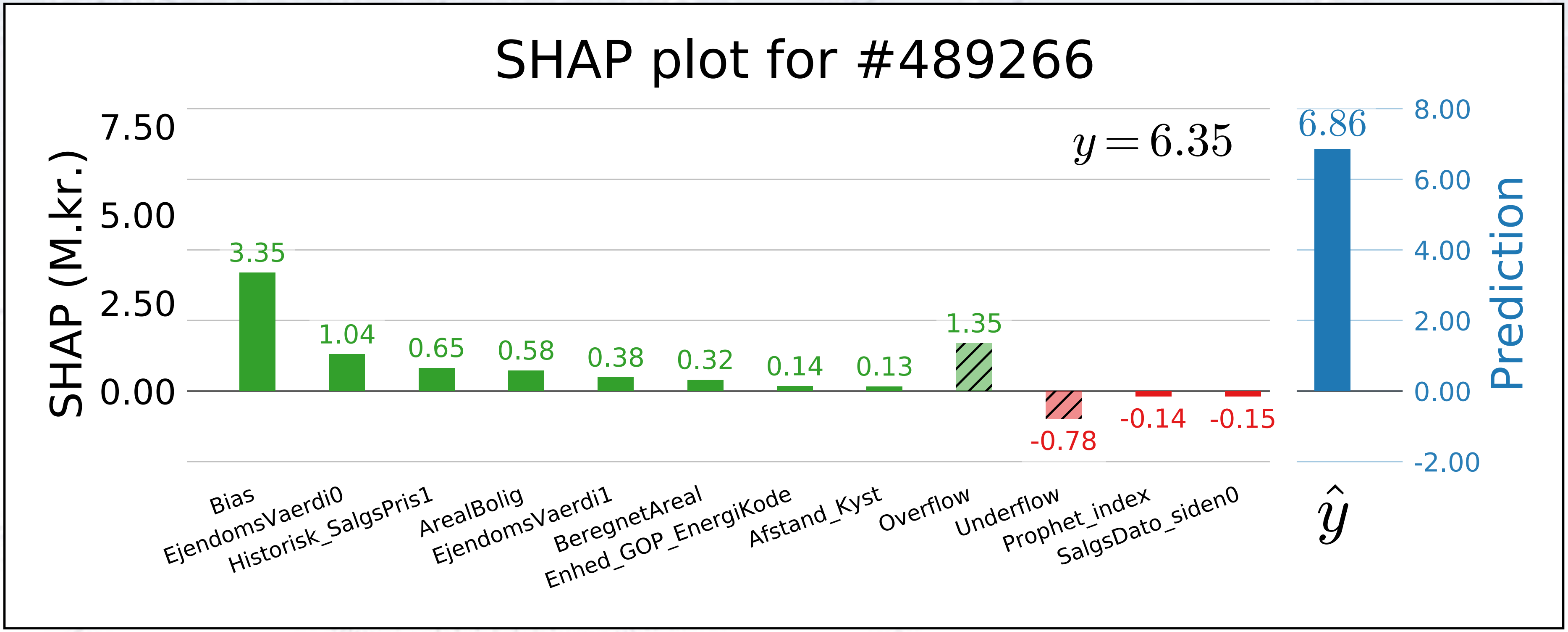


We used XGBoost to build a model: Dealt well with categories and NaNs.

For apartments, we managed to "break" the tough 10% uncertainty limit.

Individuel estimates

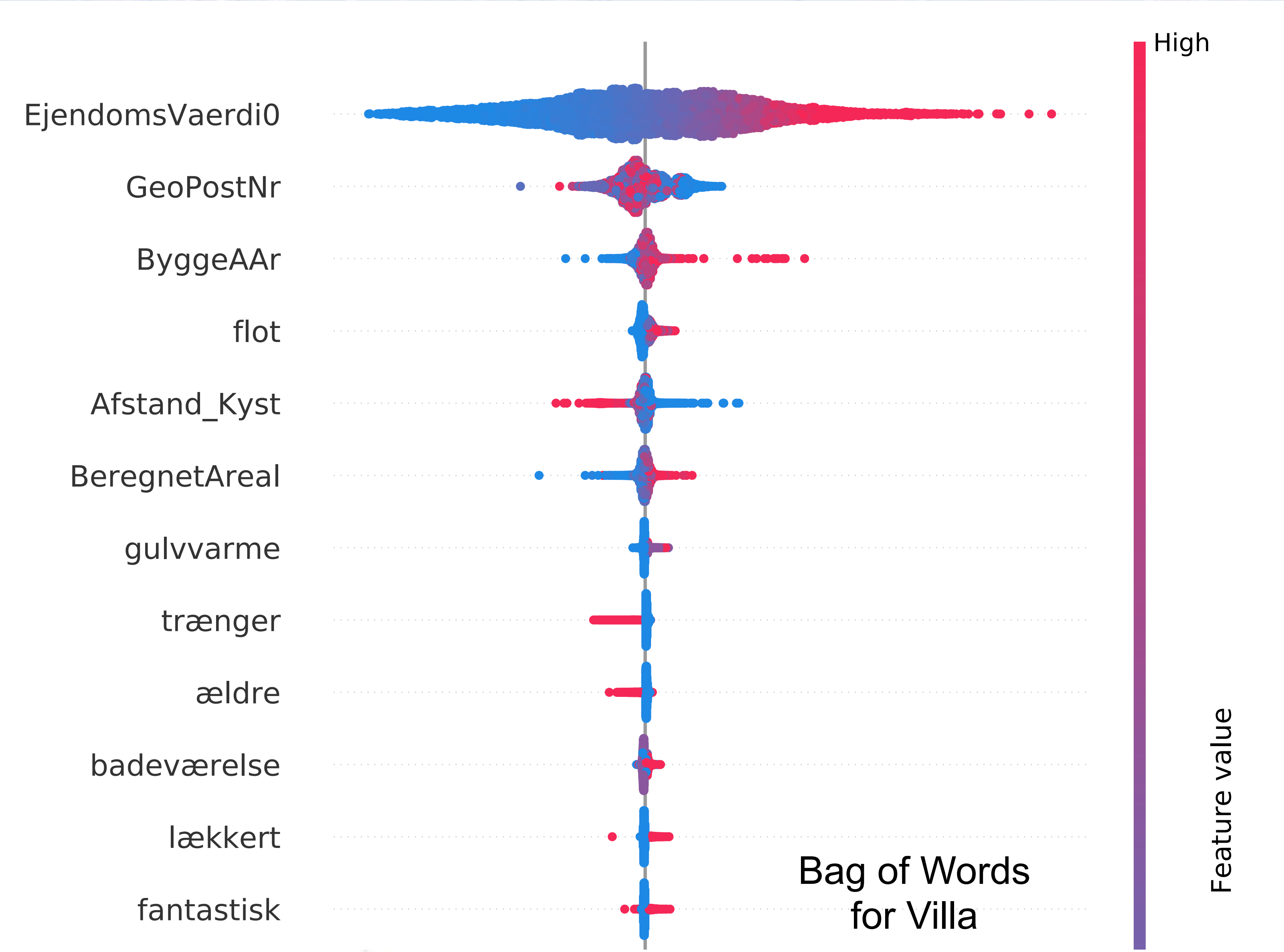
Shapley-values also gives the possibility to see the reason behind **individuel estimates**. Below is an example, illustrating this point.



Above is shown which factors that influences the final estimate of the sales price (and how much). The estimate is the sum of the contributions (here 6.86 MKr.).

This is a fantastic tool to get insight into the ML workings!!!

Word ranking



Result of including text

Natural Language Processing

Term Frequency - Inverse Document Frequency: *TF-IDF*

Natural weighting of words

CountVectorizer, TfidfVectorizer

Assign a weight to each word, according to its frequency of use.
weight_IDF = $\log(N_{all} / N_{appearances})$

MAD(XGB, numerics only) = 0.165

MAD(XGB, text only, BOW) = 0.254

MAD(XGB, combined) = 0.147

(Numerics: GeoPostNr, BeregnetAreal, ByggeAAr, EjendomsVaerdi0, Afstand_Kyst)

Result of including text

Natural Lan

Term F… …-*IDF*

Natu…

Cou…

Lessons Learned:

- The ML part of the project was fun and BDTs worked really well.
- Including text was (at the time) harder, but we had a way to cross check, if it worked.
- We were not at all prepared for the reluctance to use this in the real world.

"Big ships turn very slowly!"

… each word,
…equency of use.
…(N_{all} / $N_{appearances}$)

MAD(XGB, numerics only) = 0.165

MAD(XGB, text only, BOW) = 0.254

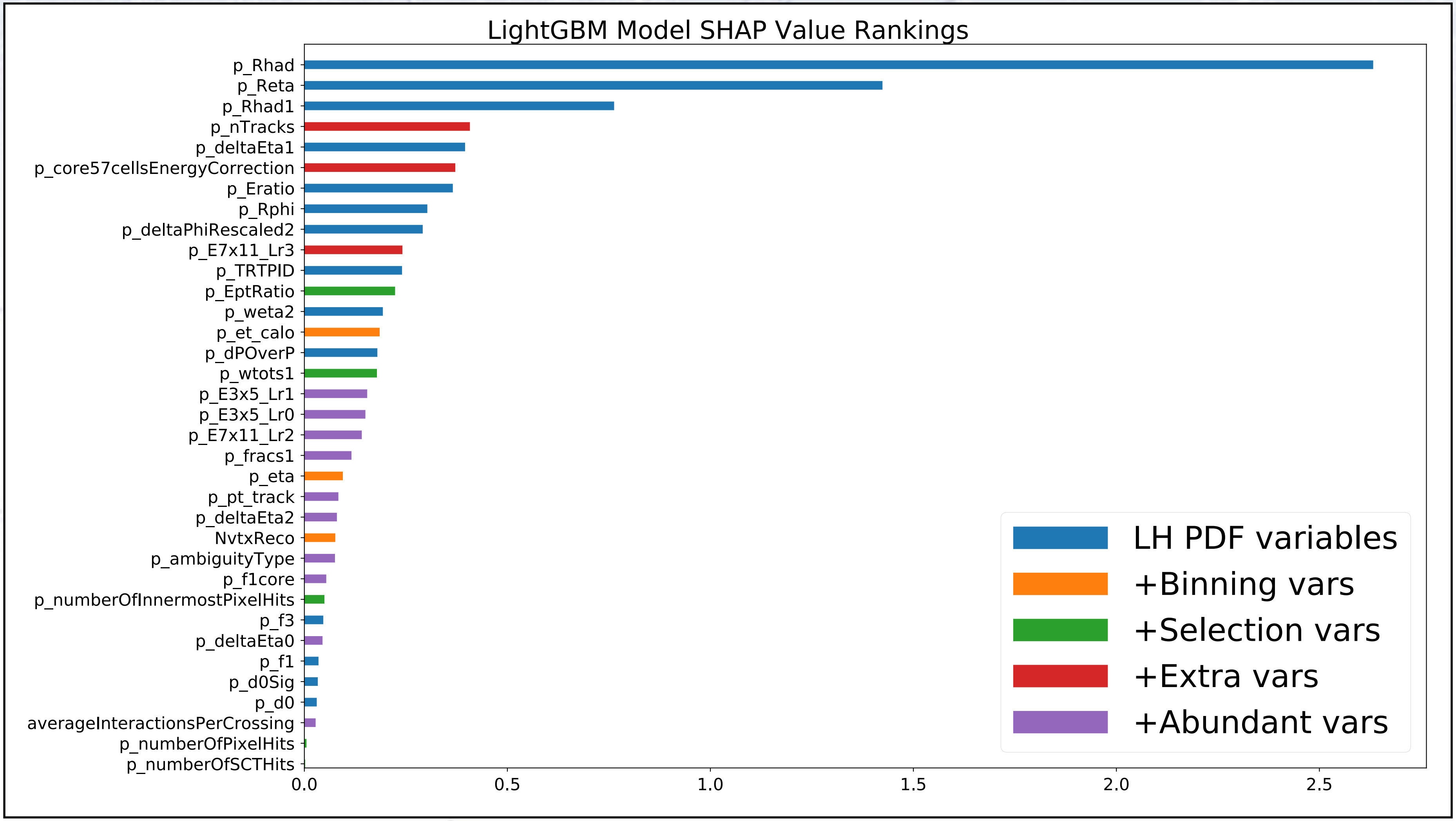
MAD(XGB, combined) = 0.147

(Numerics: GeoPostNr, BeregnetAreal, ByggeAAr, EjendomsVaerdi0, Afstand_Kyst)

Electron Identification

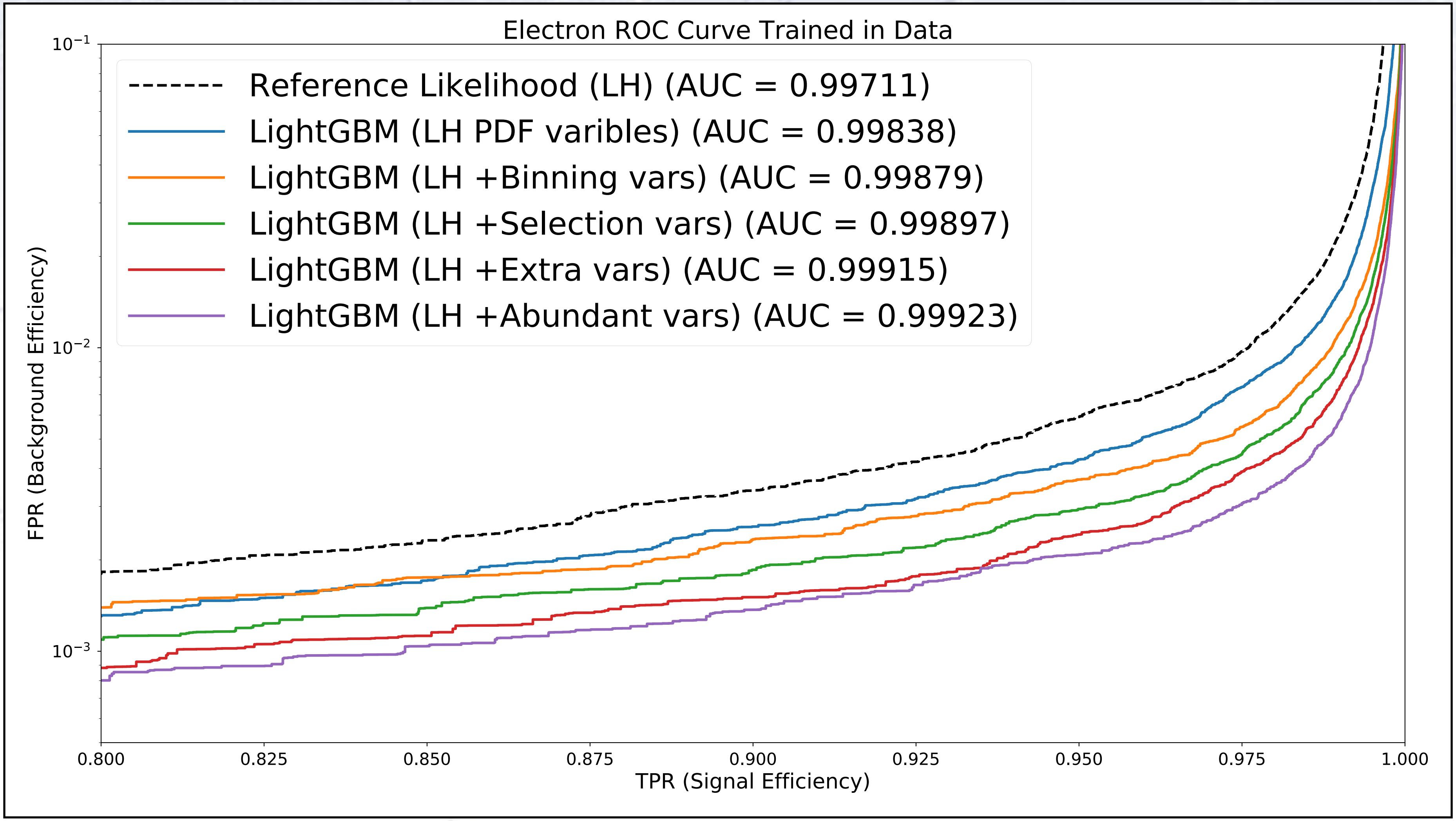
Input Feature Ranking

Here is an example from particle physics. The blue variables were "known", but with SHAP we discovered three new quite good variables in data.



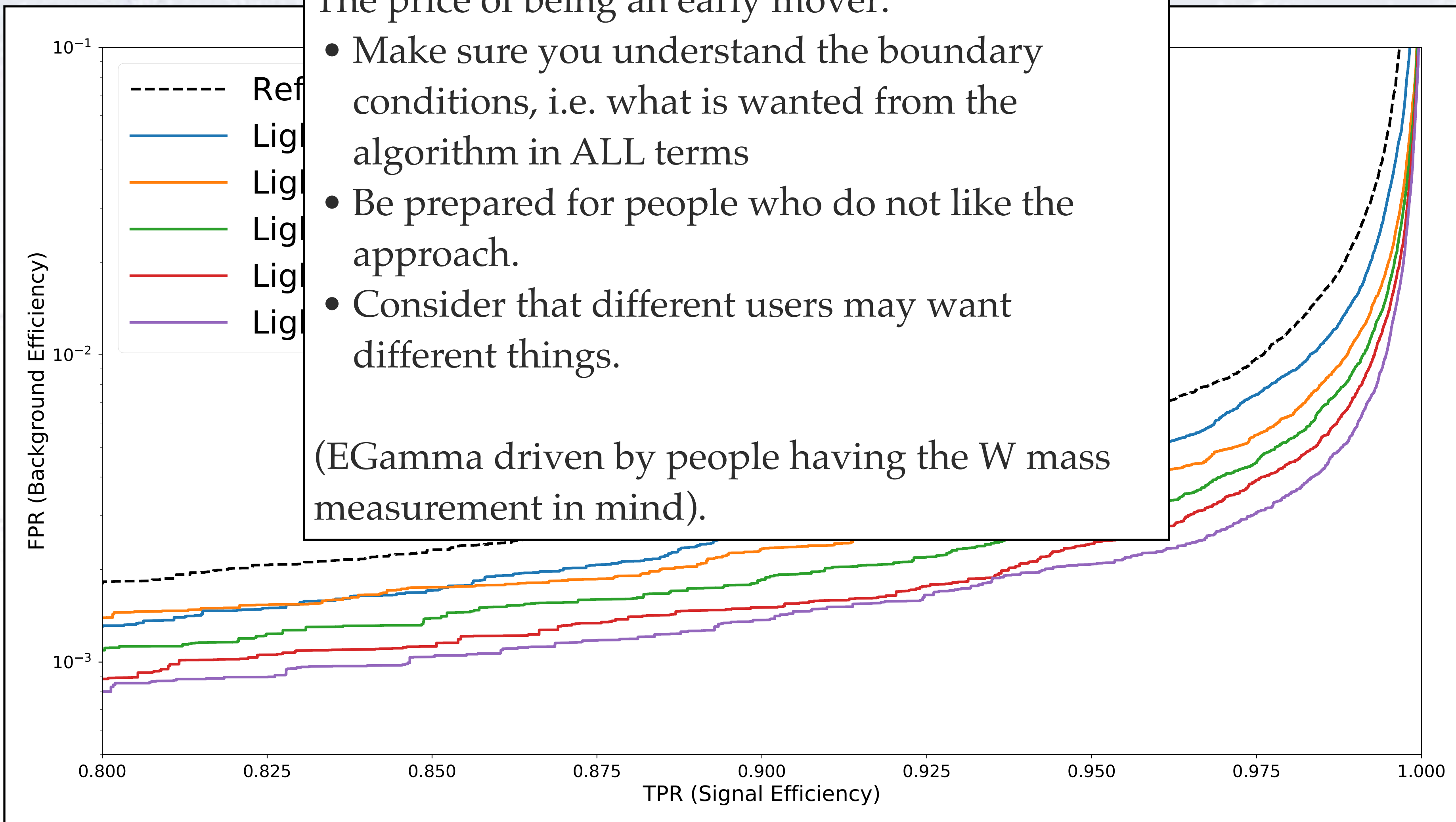
Input Feature Ranking

We could of course just add all variables, but want to stay simple, and training the models, we see that the three extra variables gives most of gain.



Input Feature Ranking

We could of c… , and
training the m… ost of gain.



<u>Lessons Learned:</u>

The price of being an early mover:

- Make sure you understand the boundary conditions, i.e. what is wanted from the algorithm in ALL terms
- Be prepared for people who do not like the approach.
- Consider that different users may want different things.

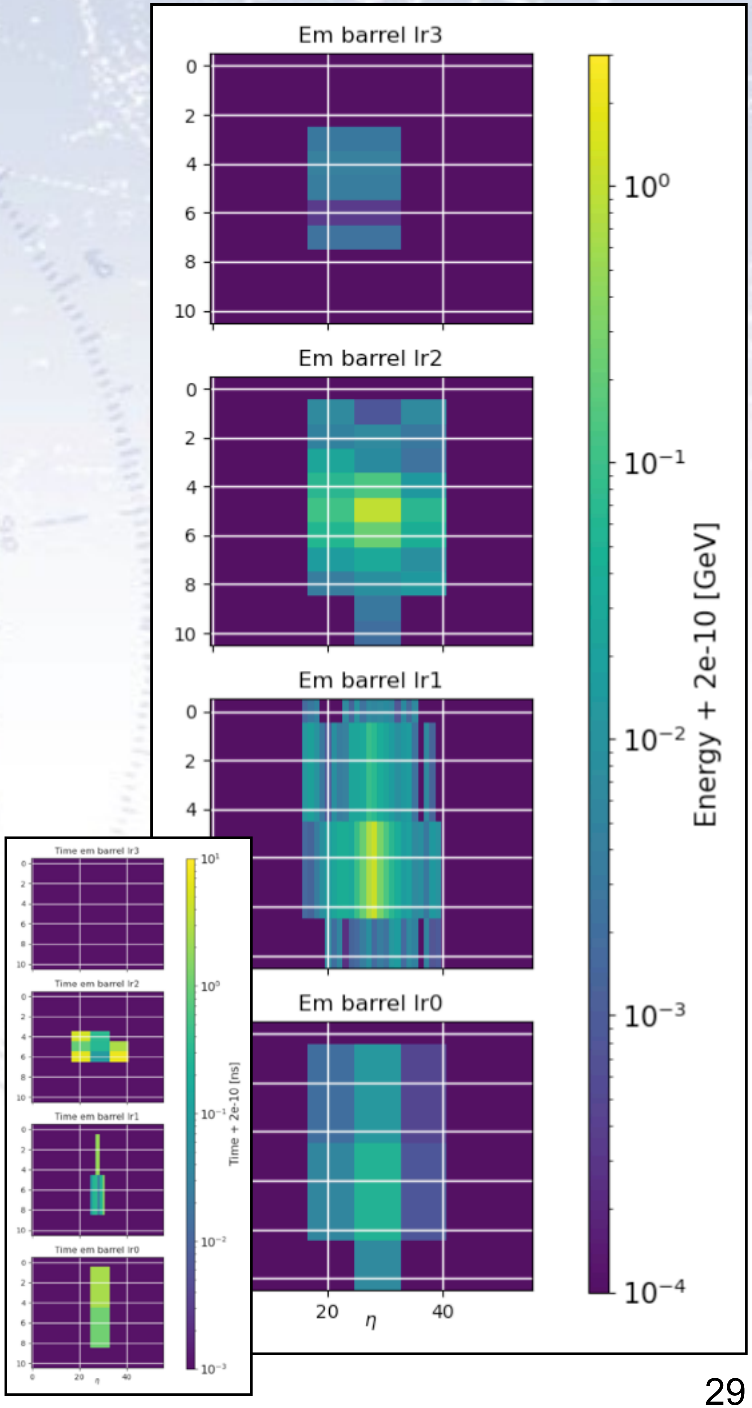
(EGamma driven by people having the W mass measurement in mind).

Electron Regression

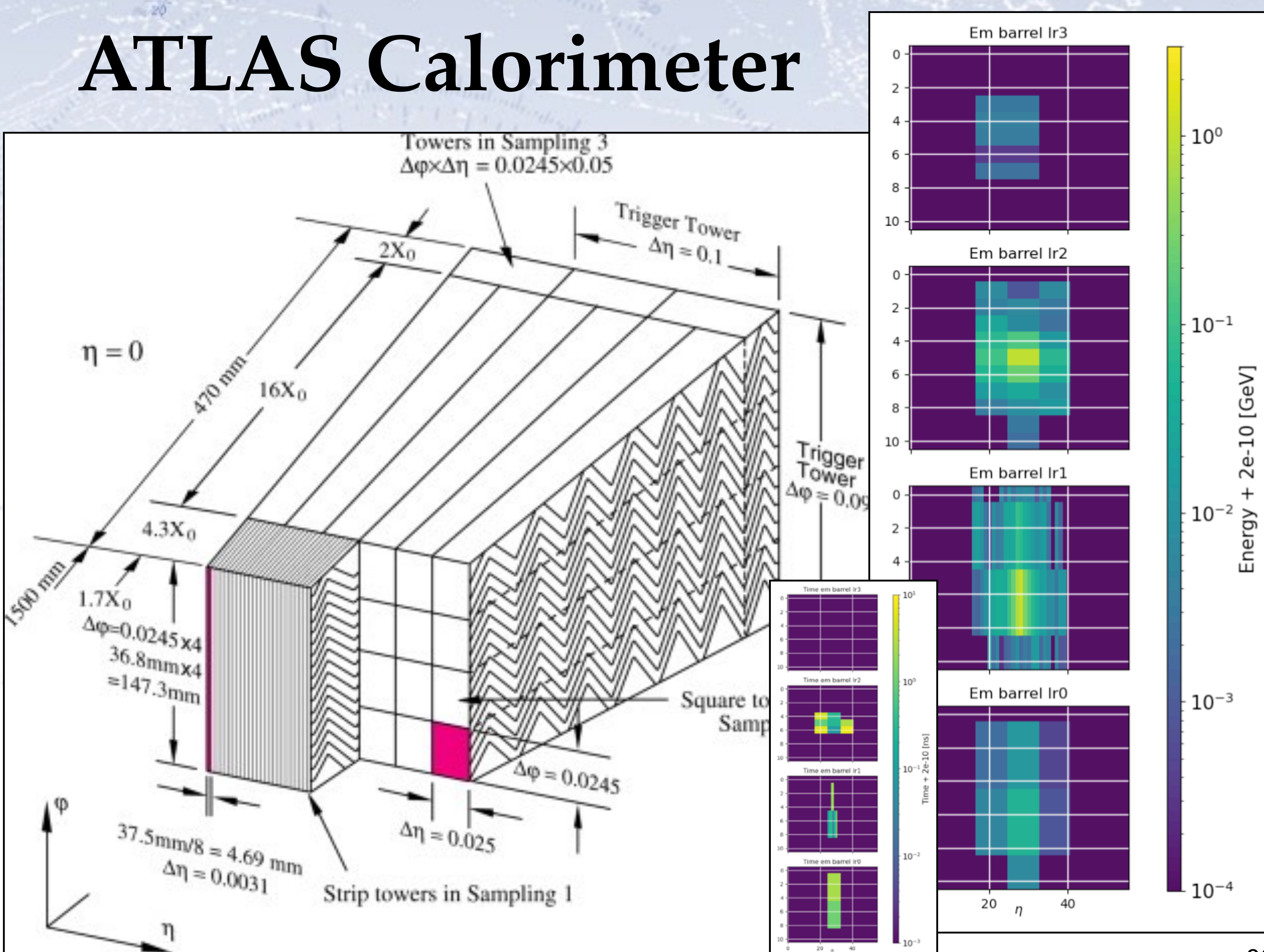
CNNs at work

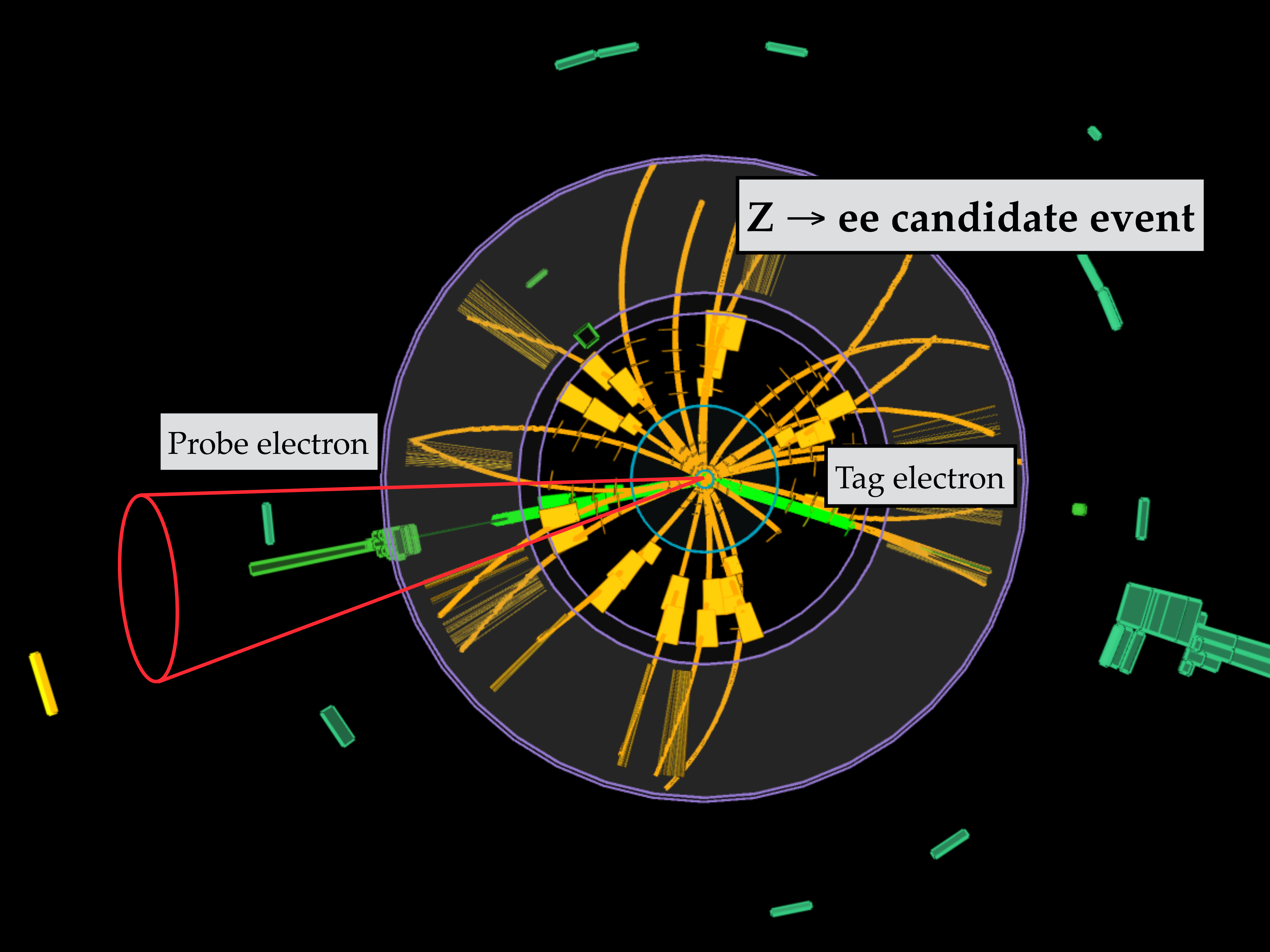
The ATLAS calorimeter data looks like images.

Can we use CNNs to get a better energy measurement?



ATLAS Calorimeter




Z → ee candidate event
Probe electron
Tag electron

The input variables

We consider the cell energies as pixels in four images. The cells contain two (used) types of information:

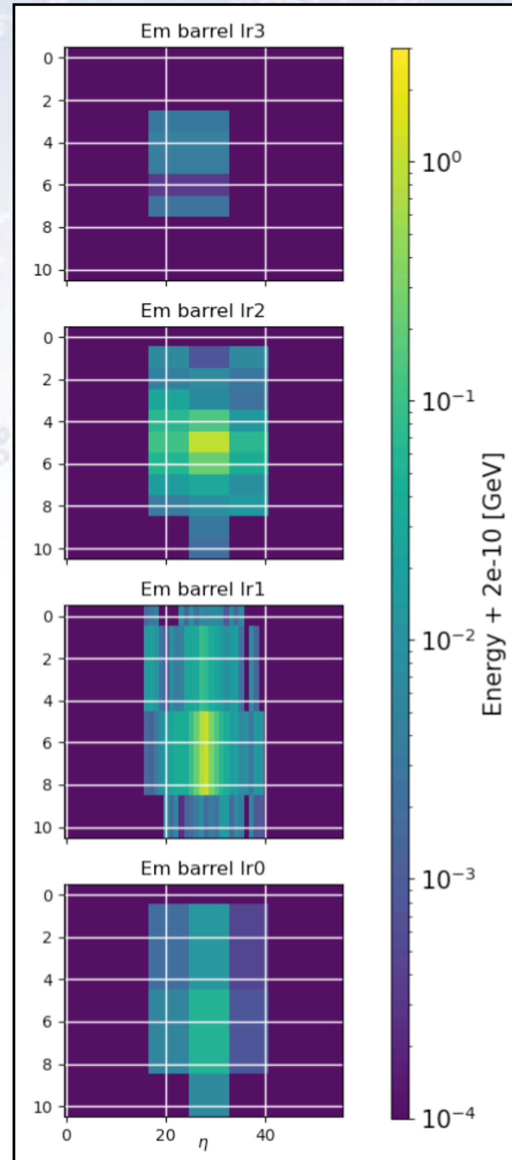
- Energy (primary variable)
- Time of cell energy

The variables are both scalar and cell based. The scalars can be seen in table on the right.

Finally, we consider the (up to) 10 nearest tracks in a "TrackNet" input:

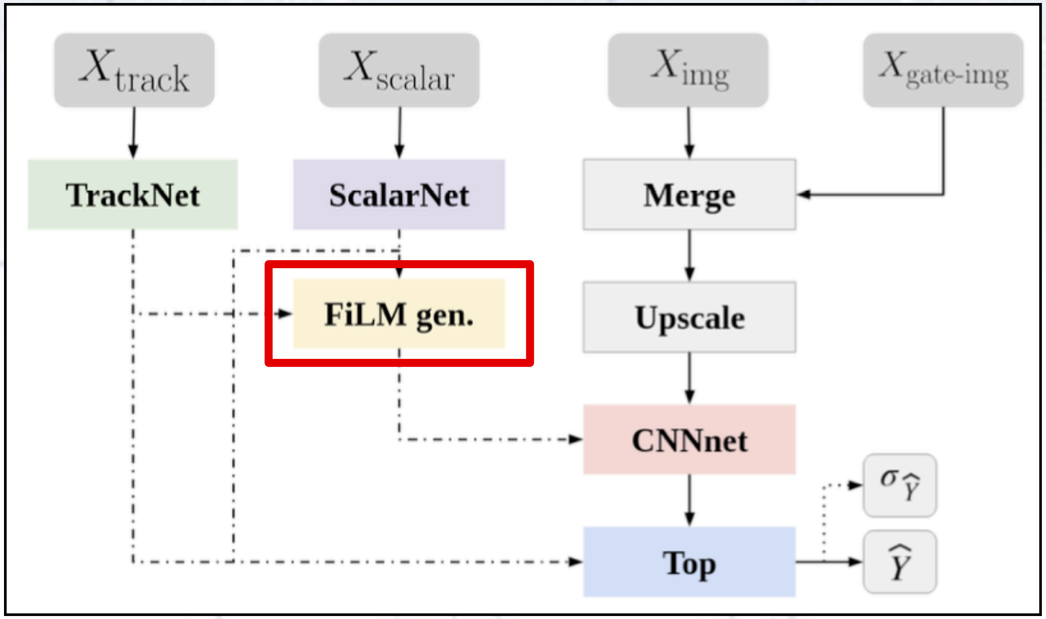
Type	Name	Description
Energy	$p_{t,track}/q_{track}$	Transverse momentum of track divided by its charge q
	d_0/σ_{d0}	$d0$ is the signed transverse distance between the point of closest approach and the z-axis where σ_{d0} is its uncertainty
	ΔR	$\Delta R = \sqrt{(\phi_0-\phi)^2+(\eta_0-\eta)^2}$
Geometric	$\text{vertex}_{\text{track}}$	Reconstructed vertex of the track
	z_0	Longitudinal distance between the point of closest approach and the z-axis.
	η_{track}	Reconstructed $\|\eta\|$ of tracks.
	ϕ_{track}	Reconstructed ϕ of tracks.
	n_{pixel}	Number of hits in the pixel detector
Misc.	n_{SCT}	Number of hits in the SCT
	n_{TRT}	Number of hits in the TRT

Type	Name	Description
	E_{acc}	Energy deposit in layer 1-3 of ECAL.
	η_{index}	η cell index of cluster of layer 2.
	$f0_{cluster}$	Ratio of energy between layer 0 and E_{acc} in $\|\eta\| < 1.8$ (end of layer 0).
Energy	$R12$	Ratio of energy between layer 1 and 2 in the ECAL.
	p_t^{track}	p_T estimated from tracking for the particle (only e).
	E_{TG3}	Ratio between the energy in the crack scintillators and E_{acc} within $1.4 < \|\eta\| < 1.6$.
	$E_{tile-gap}$	Sum of the energy deposited in the tile-gap.
	η	Pseudorapidity of the particle.
	$\Delta\phi_2^{rescaled}$	Difference between ϕ, as extrapolated by tracking, use for ECAL momentum estimation and ϕ of the ECAL cluster.
	η_{ModCalo}	Relative η position w.r.t. the cell edge of layer 2 in the ECAL*.
Geometric	$\Delta\eta_2$	Difference between η, as extrapolated by tracking, use for ECAL momentum estimation and η of the ECAL cluster (only e).
	$poscs_2$	Relative position of η within cell in layer 2 in ECAL. $2(\eta_{cluster}-\eta_{maxEcell})/0.025-1$, $\eta_{cluster}$ is η of the barycenter of the cluster and $\eta_{maxEcell}$ is η of the most energetic cell of the cluster.
	$\Delta\phi_{TH3}$	Relative position in ϕ in a cell. $\mathrm{mod}(2\pi+\phi,\pi/32)-\pi/32$.
	$\langle\mu\rangle$	Average proton-proton interaction per bunch crossing.
Misc.	n_{tracks}	# of tracks assigned (only e).
	$n_{vertexReco}$	Number of reconstructed vertices.

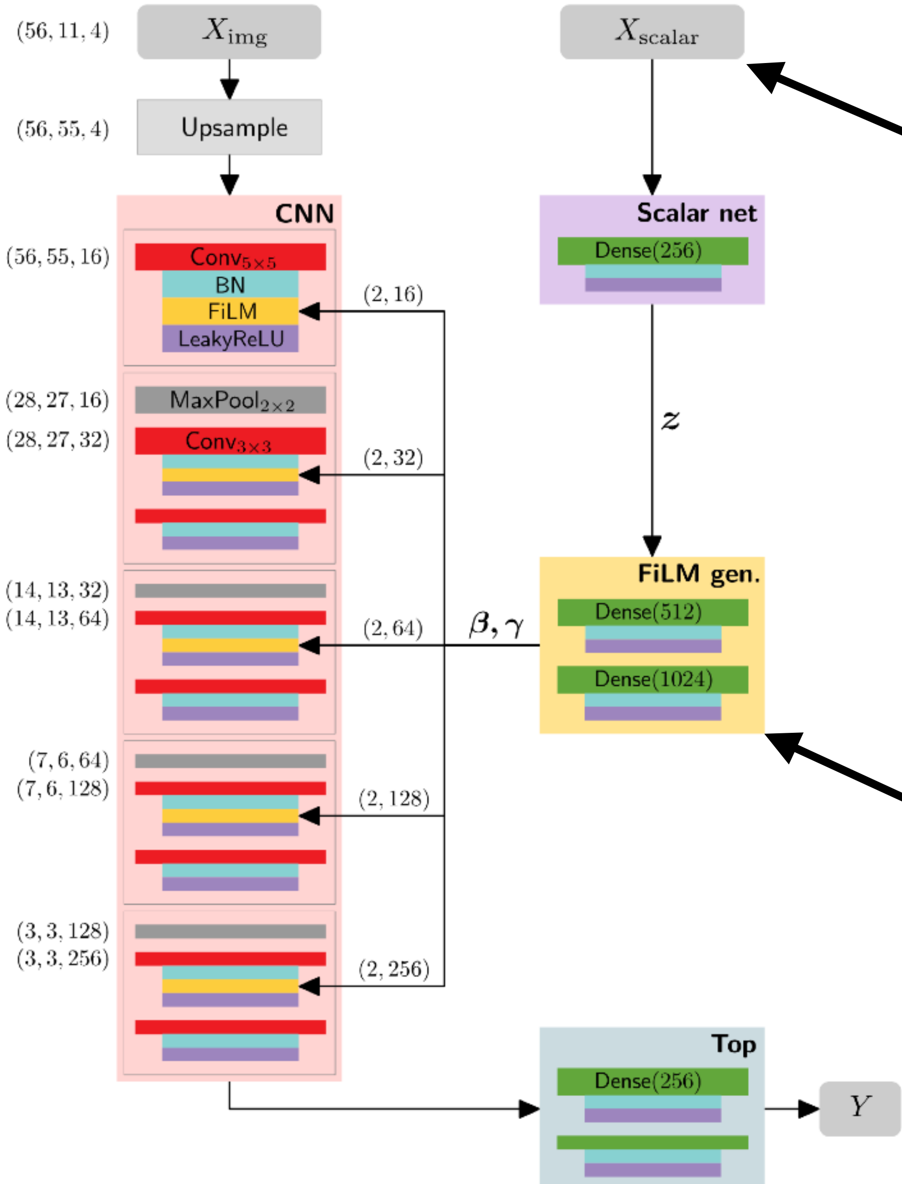


The network architecture

There are many ways to combine the input variables, and we have considered the following architectures, where the dashed lines are the considerations.



Feature wIse Linear Modulation



Type	Name	Description
	E_{acc}	Energy deposit in layer 1-3 of ECAL.
	η_{index}	η cell index of cluster of layer 2.
	$f0_{cluster}$	Ratio of energy between layer 0 and E_{acc} in $\lvert\eta\rvert < 1.8$ (end of layer 0).
Energy	$R12$	Ratio of energy between layer 1 and 2 in the ECAL.
	p_T^{track}	p_T estimated from tracking for the particle (only e).
	E_{TG3}	Ratio between the energy in the crack scintillators and E_{acc} within $1.4 < \lvert\eta\rvert < 1.6$.
	$E_{tile-gap}$	Sum of the energy deposited in the tile-gap.
	η	Pseudorapidity of the particle.
	$\Delta\phi_2^{rescaled}$	Difference between ϕ, as extrapolated by tracking, use for ECAL momentum estimation and ϕ of the ECAL cluster.
	η_{ModCalo}	Relative η position w.r.t. the cell edge of layer 2 in the ECAL*.
Geometric	$\Delta\eta_2$	Difference between η, as extrapolated by tracking, use for ECAL momentum estimation and η of the ECAL cluster (only e).
	$poscs_2$	Relative position of η within cell in layer 2 in ECAL. $2(\eta_{cluster} - \eta_{maxEcell})/0.025 - 1$, $\eta_{cluster}$ is η of the barycenter of the cluster and $\eta_{maxEcell}$ is η of the most energetic cell of the cluster.
	$\Delta\phi_{TH3}$	Relative position in ϕ in a cell. $\mathrm{mod}(2\pi + \phi, \pi/32) - \pi/32$.
	$\langle\mu\rangle$	Average proton-proton interaction per bunch crossing.
Misc.	n_{tracks}	# of tracks assigned (only e).
	$n_{vertexReco}$	Number of reconstructed vertices.

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

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

Before the convolutions are pooled, they are weighted (linearly) by the “context”. In this way, the best filters in the given case are given the most weight.

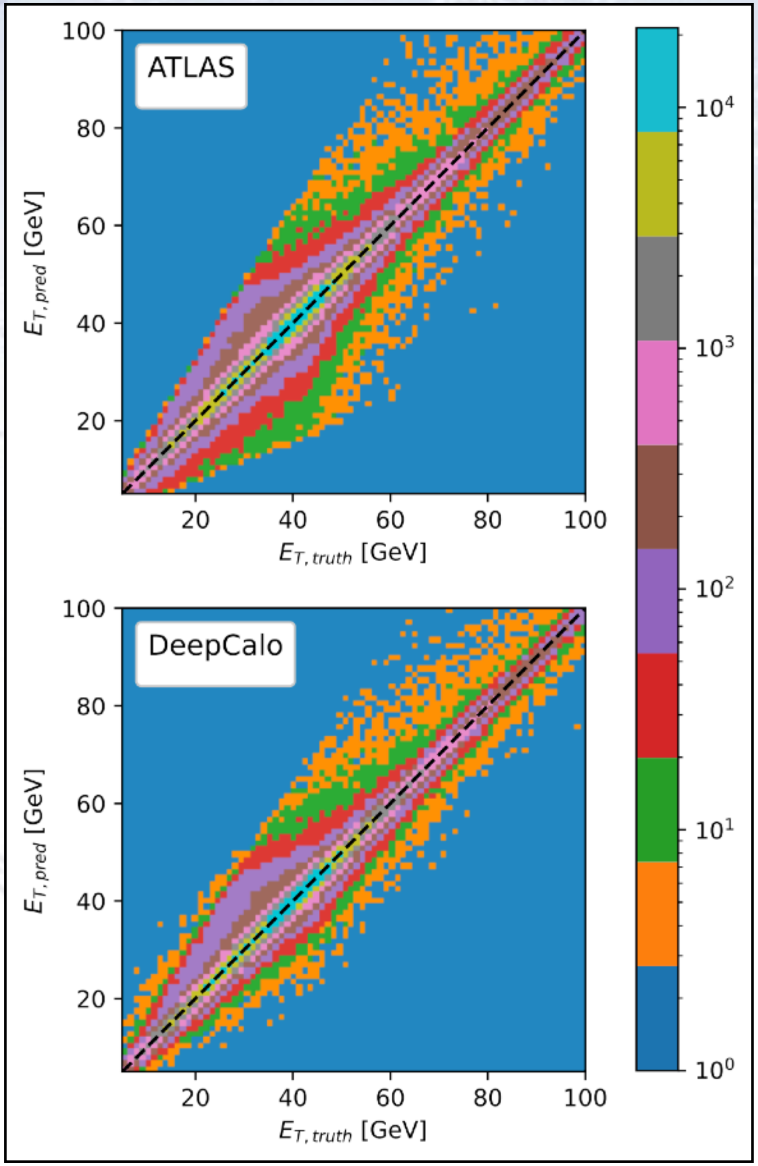
Electron Energy Regression Results (v1)

The results in 2D - MC

The E_T distribution for truth (x-axis) and reconstruction (y-axis) can be compared for the current ATLAS and the DeepCalo algorithms.

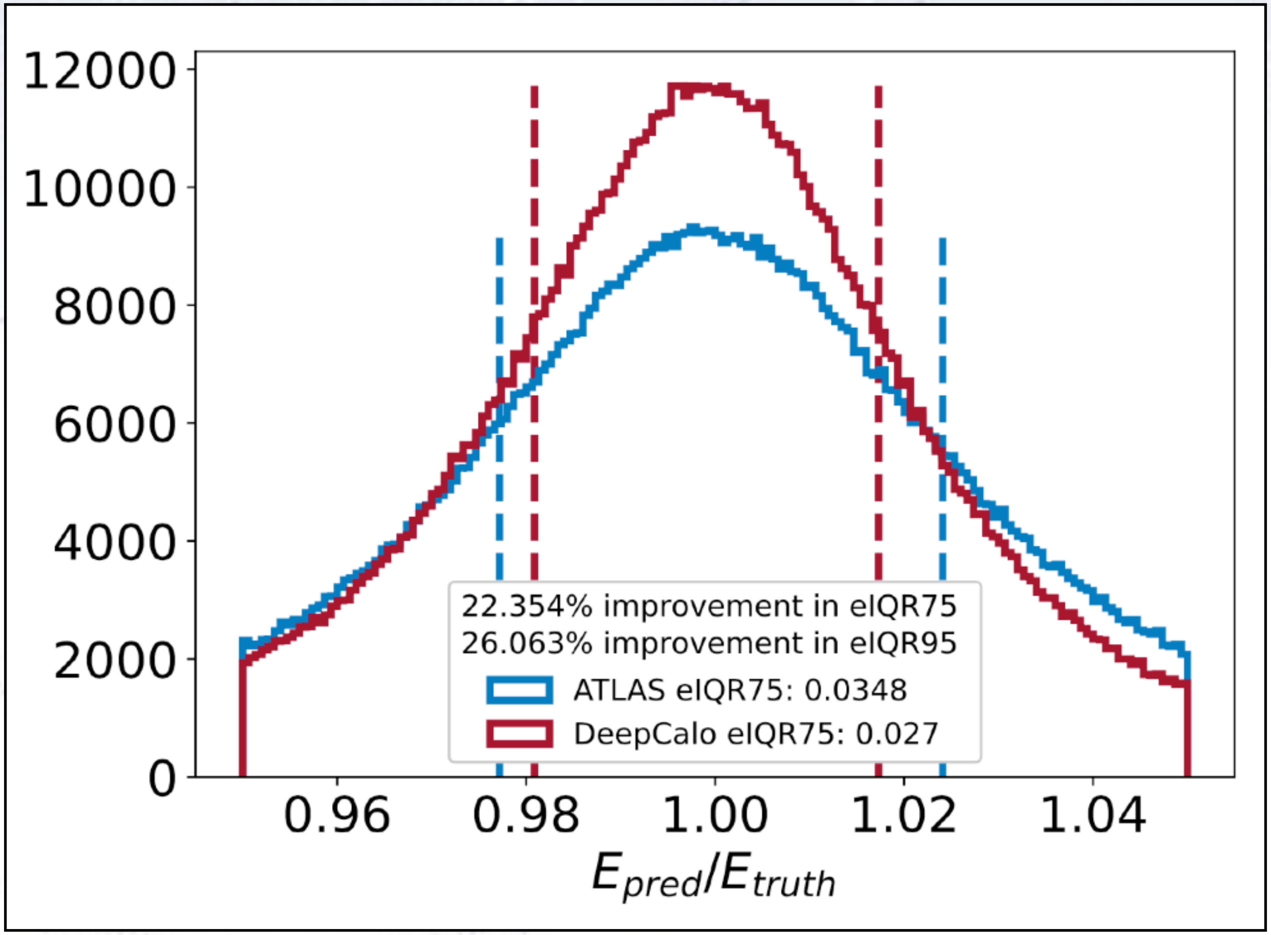
As the figure shows, both algorithms do well, and improve with energy.

As the statistics is largest around 40 GeV, this is where the comparison is most detailed, and here DeepCalo visibly has a significantly reduced lower edge.
Thus, the DeepCalo more rarely undershoots the energy.



The results in 1D - MC

Integrating the previous plot into 1D considering the RE distribution, we see a general sharpening. The improvement in relative eIQR (reIQR) is about 22%.

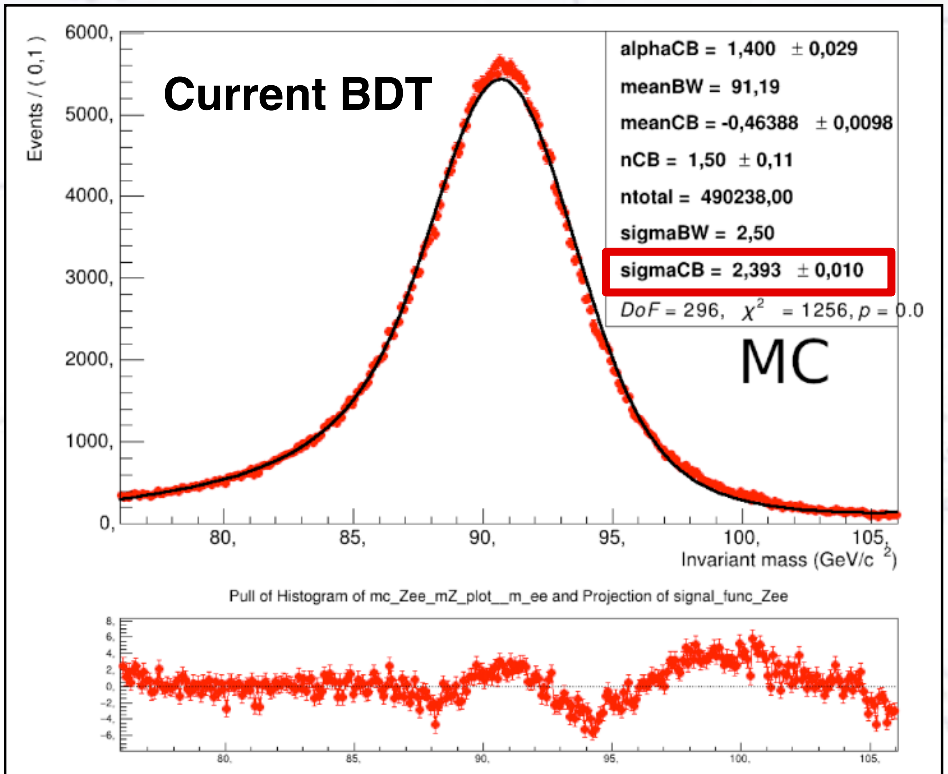


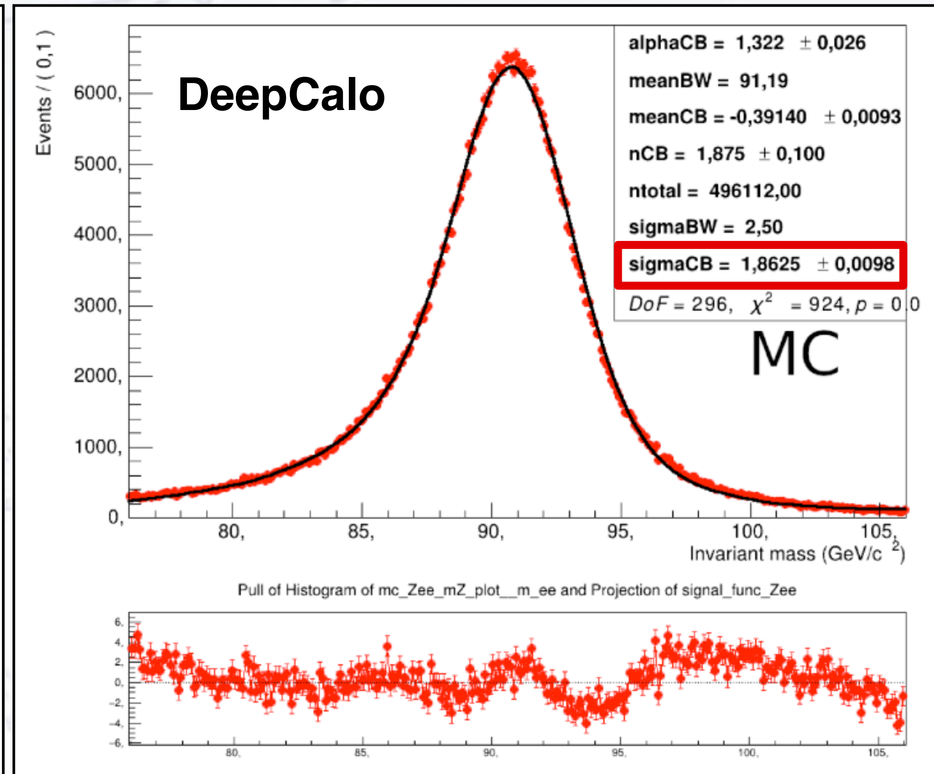
Naively, we would of course love to see a similar number in data!

Result in Zee - MC

On the Zee peak, we evaluate the improvement by fitting with a BW⊗CB fit, considering the CB width (sigmaCB) as the performance parameter. We get:

$$\langle 1 - \frac{\sigma_{CB}^{DeepCalo}}{\sigma_{CB}^{ATLAS}} \rangle = 1 - \frac{1.8310 \pm 0.006}{2.393 \pm 0.01} = 23.5 \pm 0.4\%$$

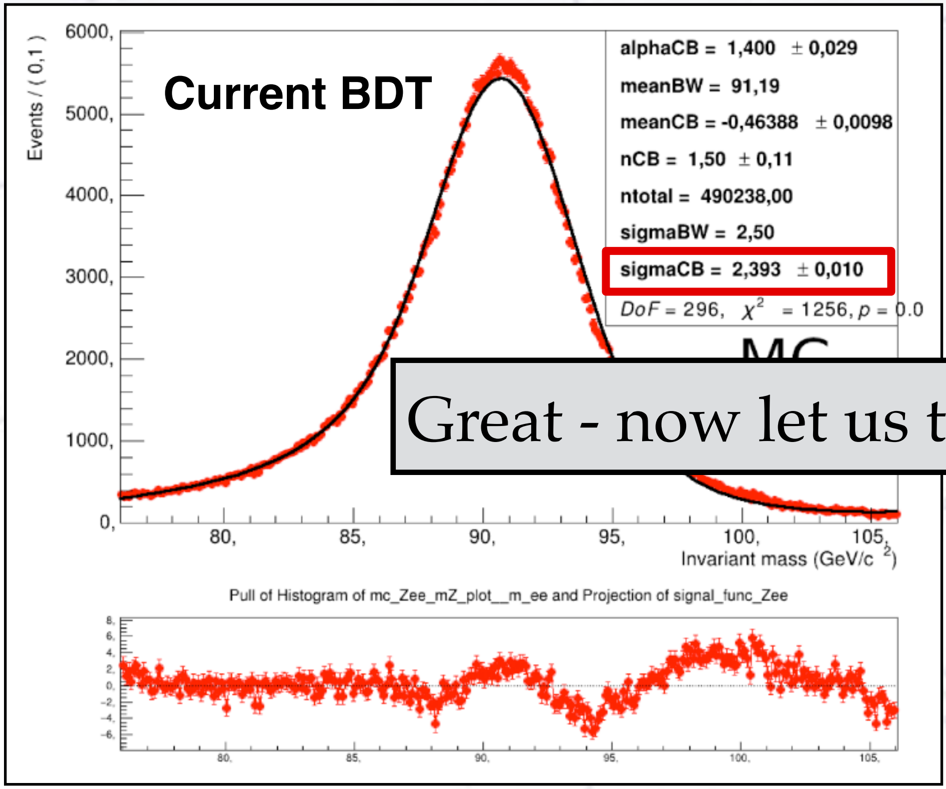


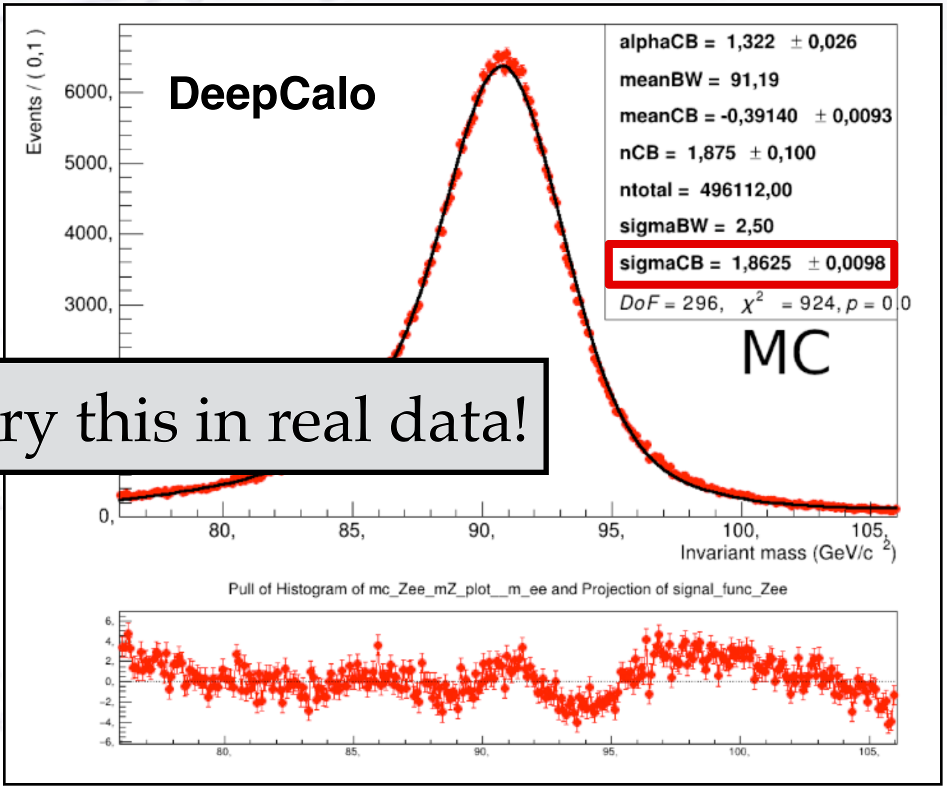


Result in Zee - MC

On the Zee peak, we evaluate the improvement by fitting with a BW⊗CB fit, considering the CB width (sigmaCB) as the performance parameter. We get:

$$\langle 1 - \frac{\sigma_{CB}^{DeepCalo}}{\sigma_{CB}^{ATLAS}} \rangle = 1 - \frac{1.8310 \pm 0.006}{2.393 \pm 0.01} = 23.5 \pm 0.4\%$$





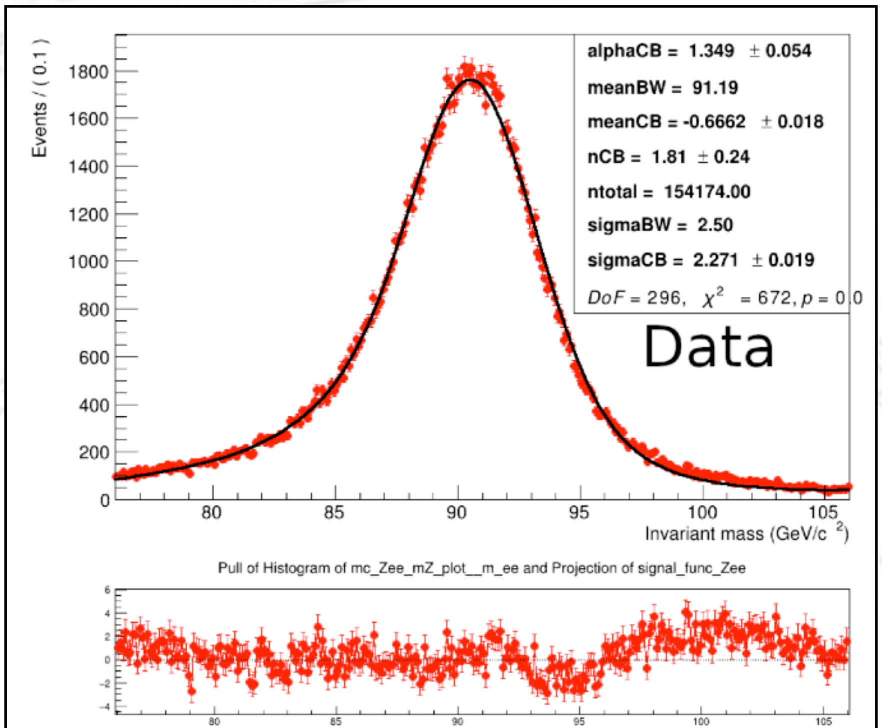
Great - now let us try this in real data!

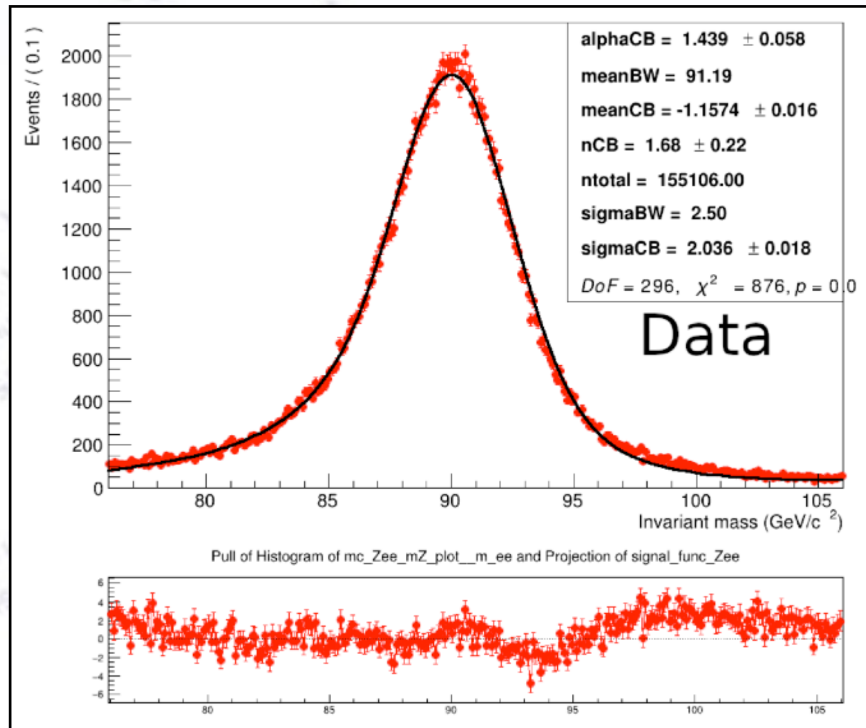
Results on Zee - data (v1)

The result we get is a much more modest improvement:

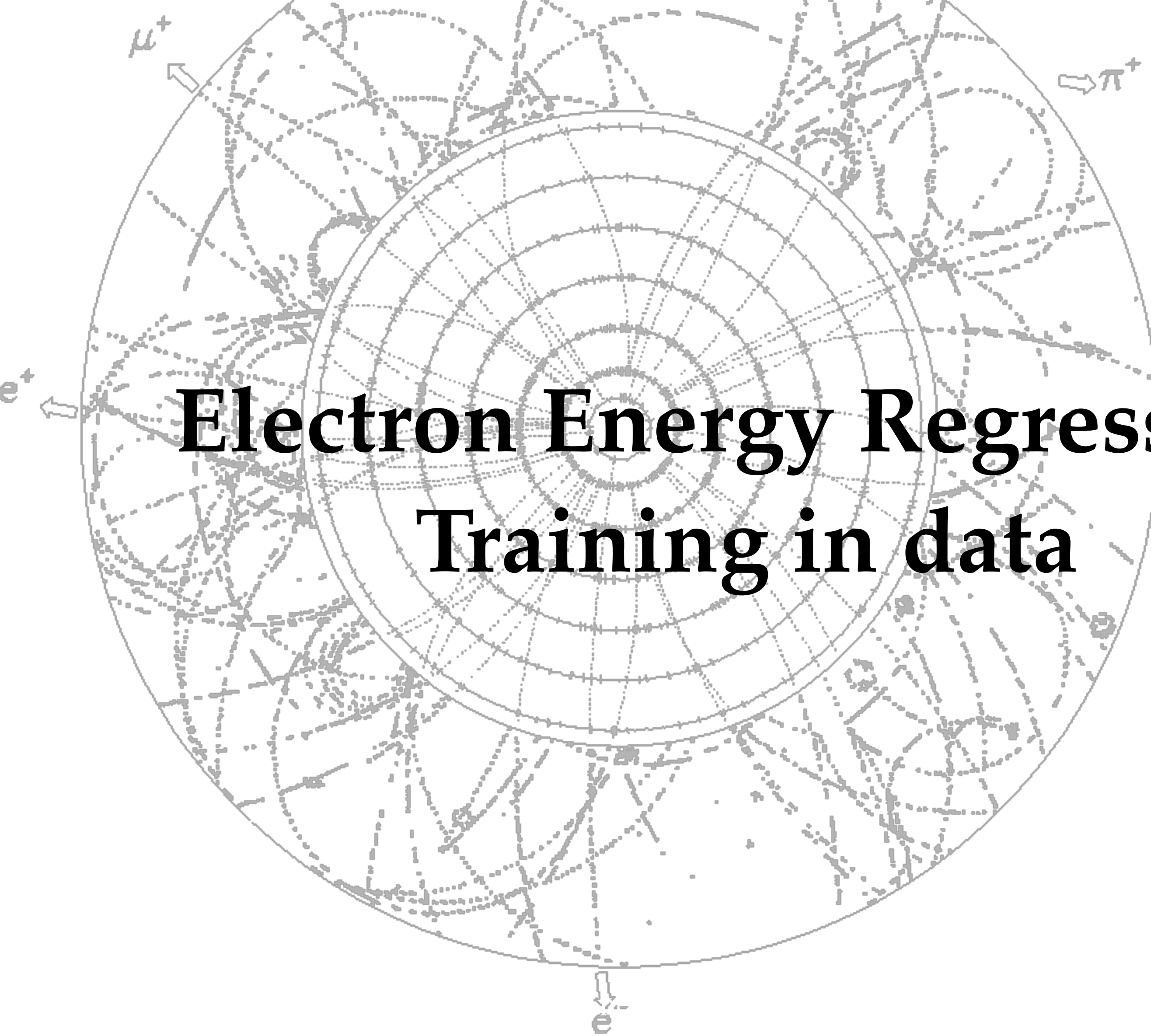
$$\langle 1 - \frac{\sigma_{CB}^{DeepCalo}}{\sigma_{CB}^{ATLAS}} \rangle = 1 - \frac{2.058 \pm 0.010}{2.271 \pm 0.019} = 9.4 \pm 0.9\%.$$

Though perhaps a little disappointing, this is not surprising, as we can not expect the MC to mimic data perfectly in the very large space considered.





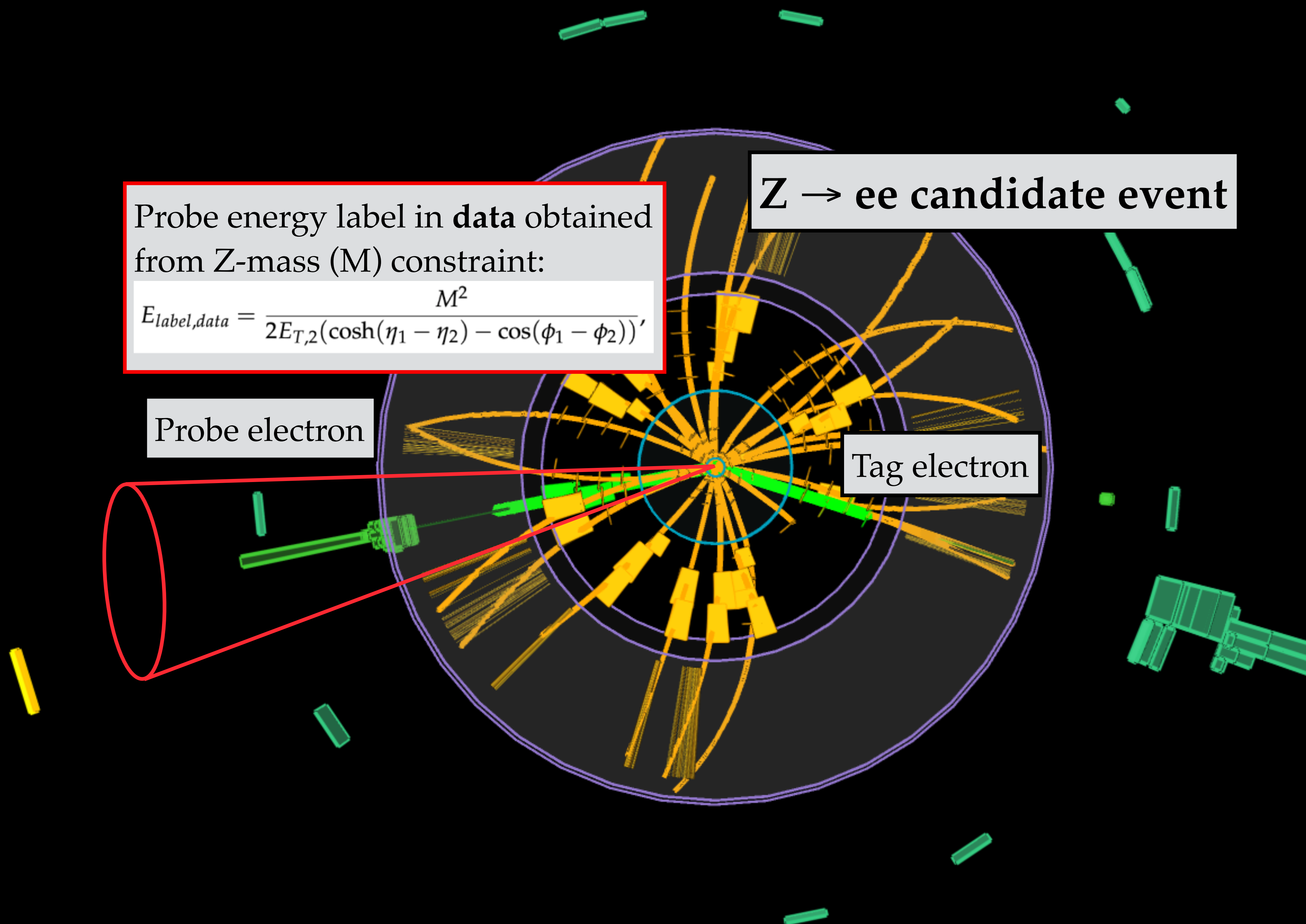
Electron Energy Regression Training in data



$Z \rightarrow ee$ candidate event

Probe energy label in **data** obtained from Z-mass (M) constraint:

$$E_{label,data} = \frac{M^2}{2E_{T,2}(\cosh(\eta_1 - \eta_2) - \cos(\phi_1 - \phi_2))},$$



Training in data

Using Zee events with invariant masses 86-97 GeV, one can get “approximate labels” in data, by assuming the true Z mass:

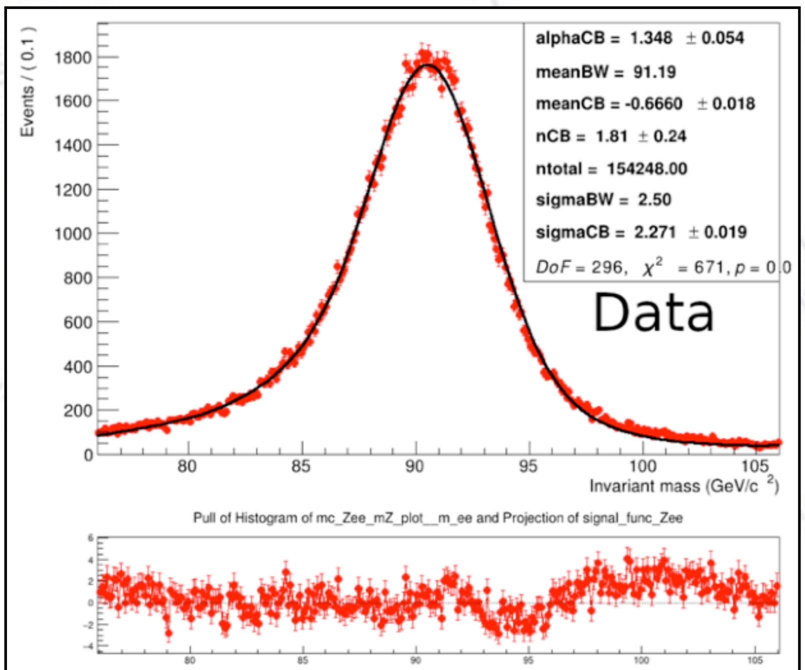
$$M^2 = 2p_{T,1}p_{T,2}(\cosh(\eta_1-\eta_2)-\cos(\phi_1-\phi_2)), \quad p_T = E_T \updownarrow$$

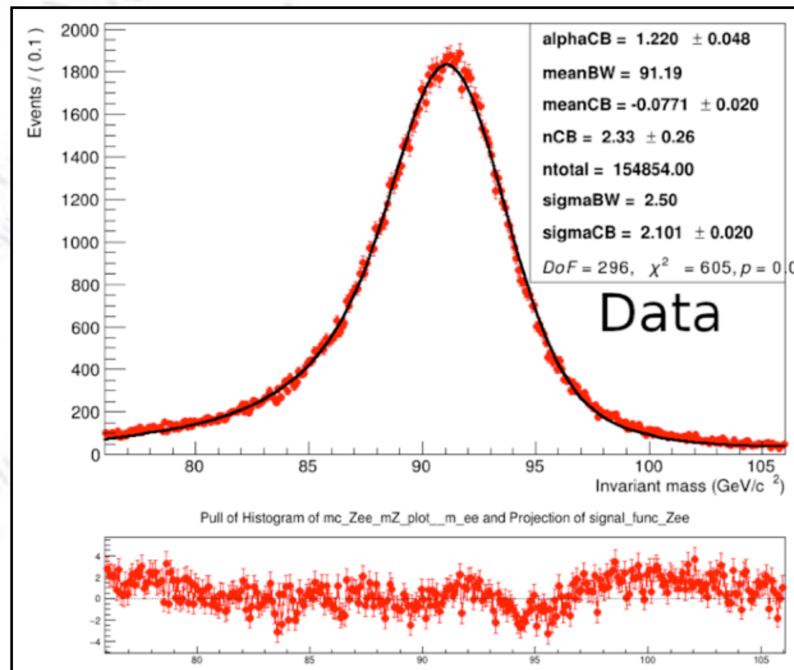
$$E_{label,data} = \frac{M^2}{2E_{T,2}(\cosh(\eta_1-\eta_2)-\cos(\phi_1-\phi_2))},$$

with $E_{T,2} = Ecalib^{(BDT)}$ and $M^2 = 91.19^2$

Using such labels, we train in data and get…

$$\langle 1 - \frac{\sigma_{CB}^{DeepCalo}}{\sigma_{CB}^{ATLAS}} \rangle = 5.9 \pm 0.9\%$$





Training in data

Using Zee events with invariant masses 86-97 GeV, one can get "approximate labels" in data, by assuming the true Z mass:

$$M^2 = 2p_{T,1}p_{T,2}(\cosh(\eta_1 - \eta_2) - \cos(\phi_1 - \phi_2)), \quad p_T = E_T \updownarrow$$

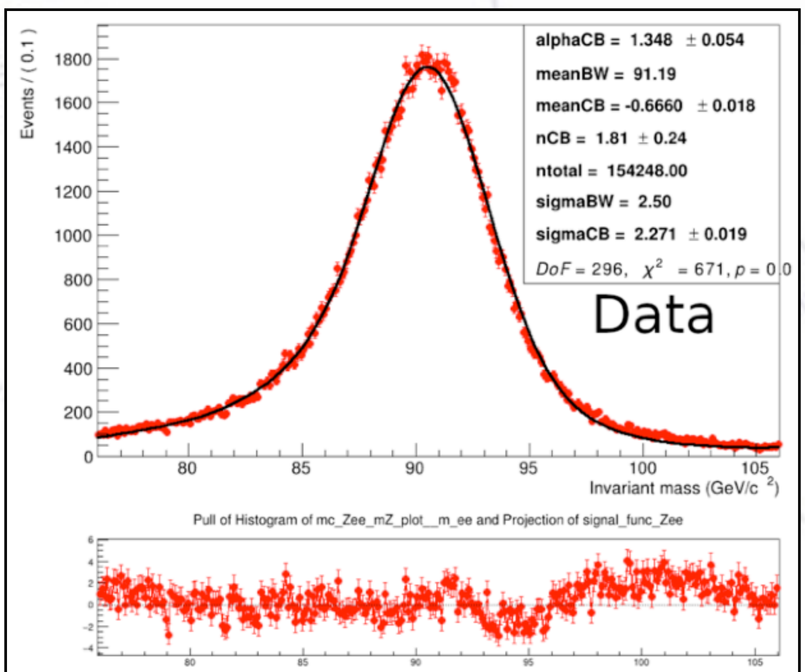
$$E_{label,data} = \frac{M^2}{2E_{T,2}(\cosh(\eta_1 - \eta_2) - \cos(\phi_1 - \phi_2))},$$

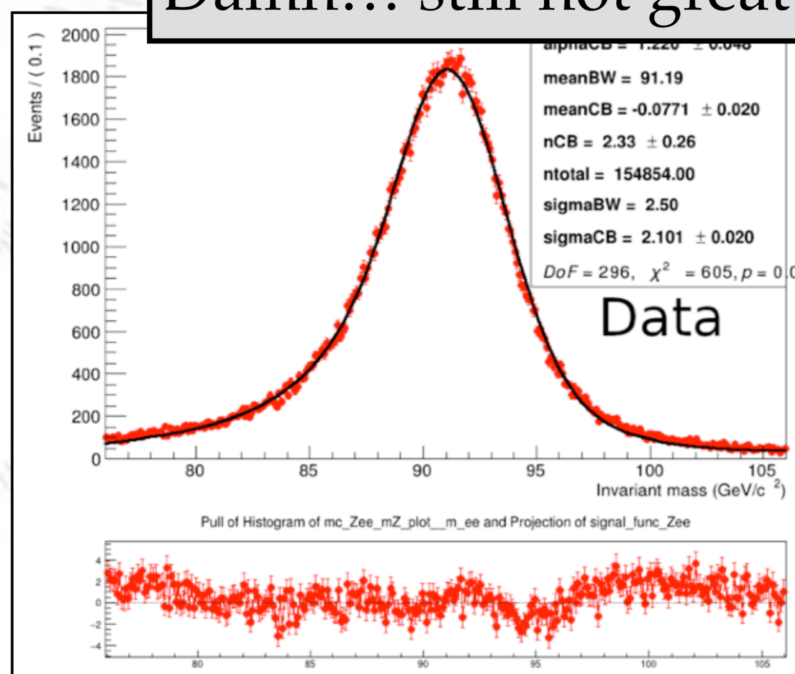
$$\text{with } E_{T,2} = \mathrm{Ecalib}^{(BDT)} \text{ and } M^2 = 91.19^2$$

Using such labels, we train in data and get…

$$\langle 1 - \frac{\sigma_{CB}^{DeepCalo}}{\sigma_{CB}^{ATLAS}} \rangle = 5.9 \pm 0.9\%$$

Damn… still not great!





Electron Energy Regression Training in data and MC

Training in data and MC

Once we have labels in data, there is nothing keeping us from combining the loss functions of MC and data (they even have the same form), and thus training **simultaneously** in data and MC:

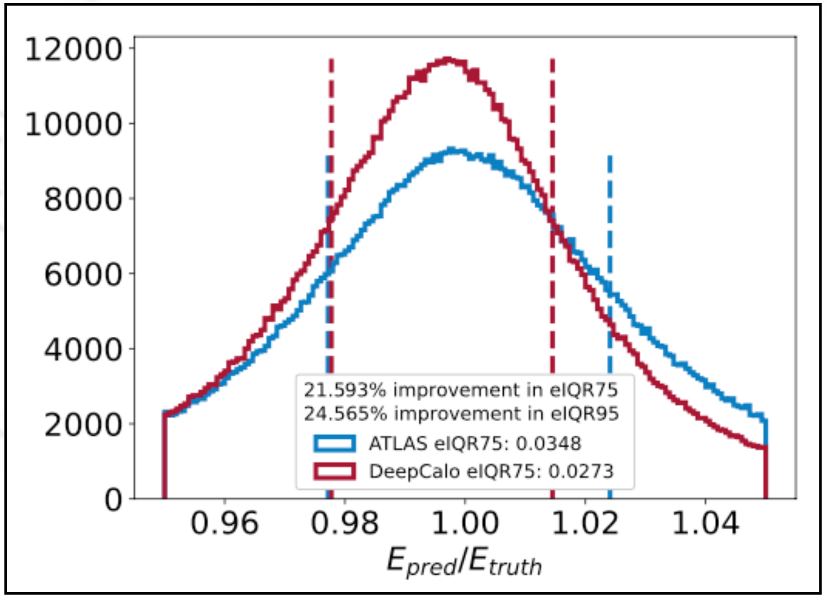
$$\mathcal{L}(y,\hat{y}) = \mathcal{L}(y_{(\text{Zee, MC})}, \hat{y}_{(\text{Zee, MC})}) + \mathcal{L}(y_{(\text{Zee, Data})}, \hat{y}_{(\text{Zee, Data})})$$

This allows the model to both use the "strength" of MC, but also learn the differences between MC and real data.

Doing this and trying out the result in MC first yields:

$$\langle reIQR_{75}^{DeepCalo} \rangle = 22.1 \pm 0.3\%$$

OK, so at least it doesn't ruin the model for MC. Now let us try data…

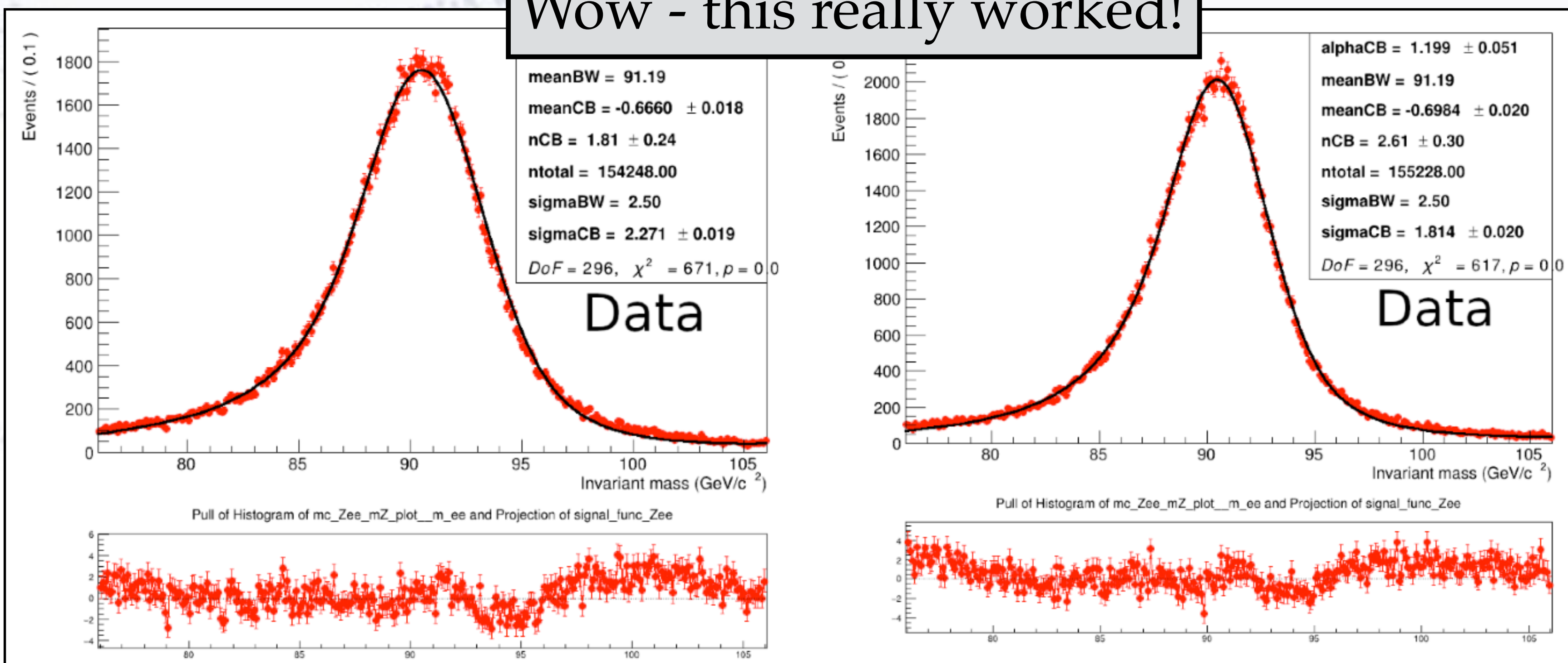


Result in data (v2)

The result in data is rather encouraging, and **greater than the sum of the improvements** from training separately in MC (9.4%) and data (5.9%).

$$\langle 1 - \frac{\sigma_{CB}^{DeepCalo}}{\sigma_{CB}^{ATLAS}} \rangle = 1 - \frac{1.86 \pm 0.010}{2.271 \pm 0.019} = 18.3 \pm 0.8\%,$$

Wow - this really worked!



Outlook

While this is still “only” an improvement in the electron energy regression, and only for lower energies (Zee range), the simultaneous training allows for extending the energy range, by including the Electron Gun MC.

Furthermore, this training might be extended to include photons, as these behave much the same as electrons, and suffer the same sources of uncertainties and smearing.

For improving the H → γγ resolution, one might use the following loss function and related training samples:

$$\boxed{\begin{aligned}\mathcal{L}(y,\hat{y}) = \mathcal{L}(y_{(\text{Zee, MC})},\hat{y}_{(\text{Zee, MC})}) + \mathcal{L}(y_{(\text{Zee, Data})},\hat{y}_{(\text{Zee, Data})}) +\\ \mathcal{L}(y_{(\text{Z}\mu\mu\gamma\text{, MC})},\hat{y}_{(\text{Z}\mu\mu\gamma\text{, MC})}) + \mathcal{L}(y_{(\text{Z}\mu\mu\gamma\text{, Data})},\hat{y}_{(\text{Z}\mu\mu\gamma\text{, Data})}) +\\ \mathcal{L}(y_{(H\gamma\gamma\text{, MC})},\hat{y}_{(H\gamma\gamma\text{, MC})})\end{aligned}}$$

Meanwhile, we are trying to write this up somehow (but Malte is now a Ph.D. in Geneva).

Outlook

While this is st … gression, and only for lower … ws for extending the …

Furthermore, t … as these behave much t … f uncertainties and smearing.

Lessons Learned:

- Remember to think about publishing. Even what may seem "a fun little example" at the time, may turn out to inspire a new line of thinking.
- Remember to think about the longevity of any approach. In this case, the storage of cell information was discontinued shortly after!

For improving the $H \to \gamma\gamma$ resolution, one might use the following loss function and related training samples:

$$\begin{aligned}\mathcal{L}(y,\hat{y}) = \mathcal{L}(y_{(\text{Zee, MC})},\hat{y}_{(\text{Zee, MC})}) + \mathcal{L}(y_{(\text{Zee, Data})},\hat{y}_{(\text{Zee, Data})}) + \\ \mathcal{L}(y_{(\text{Z}\mu\mu\gamma,\text{ MC})},\hat{y}_{(\text{Z}\mu\mu\gamma,\text{ MC})}) + \mathcal{L}(y_{(\text{Z}\mu\mu\gamma,\text{ Data})},\hat{y}_{(\text{Z}\mu\mu\gamma,\text{ Data})}) + \\ \mathcal{L}(y_{(H\gamma\gamma,\text{ MC})},\hat{y}_{(H\gamma\gamma,\text{ MC})})\end{aligned}$$

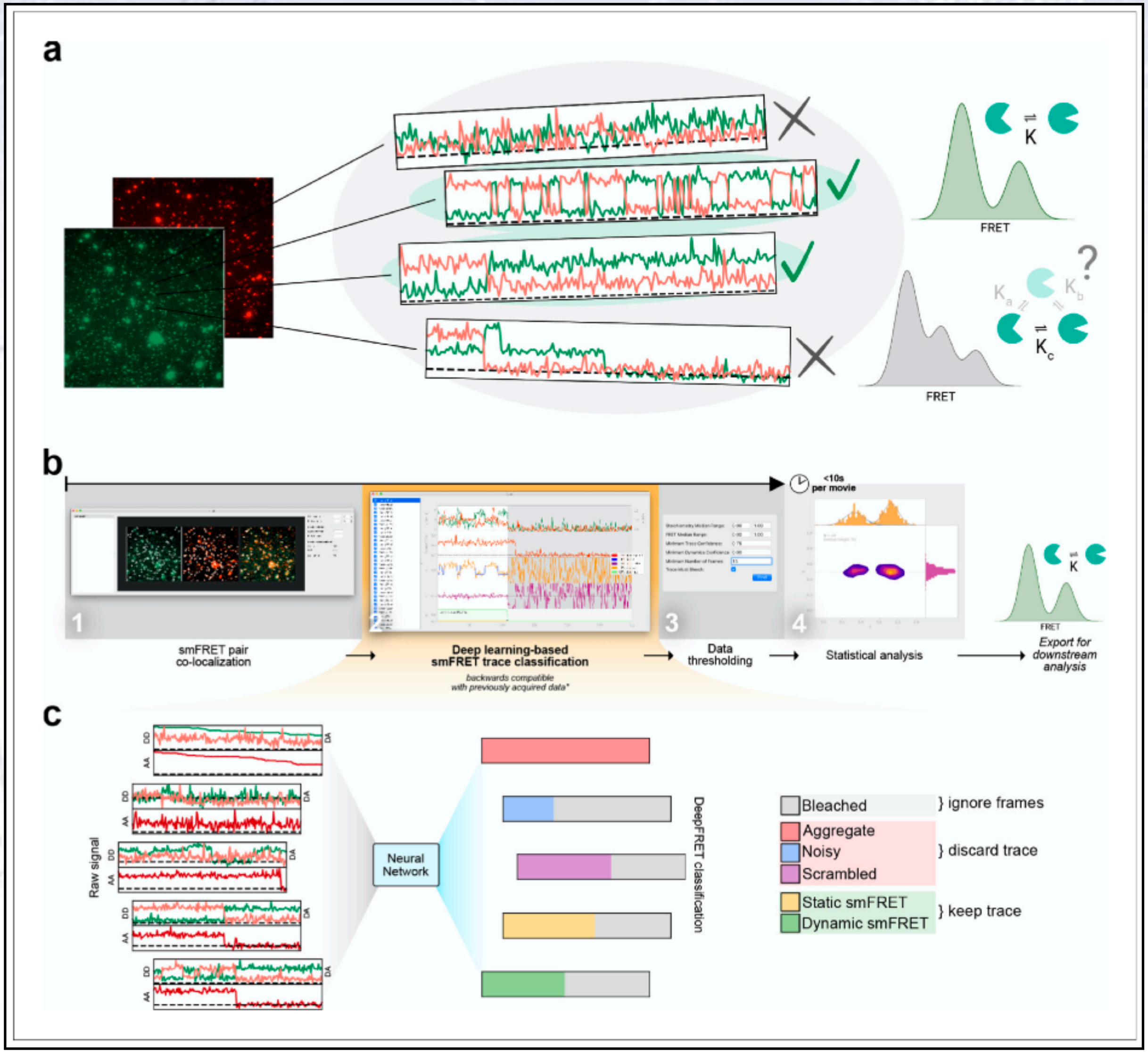
Meanwhile, we are trying to write this up somehow (but Malte is now a Ph.D. in Geneva).

DeepFRET

FRET is a technique used to study and dynamics of biomolecules.
The data is a "trace", which is a time series with possible phase transitions.

The group would go through 10000 traces and select about 250 of these… **by hand**!!!
This took a few people about a week, and was neither reproducible nor optimal.

So we made DeepFRET.

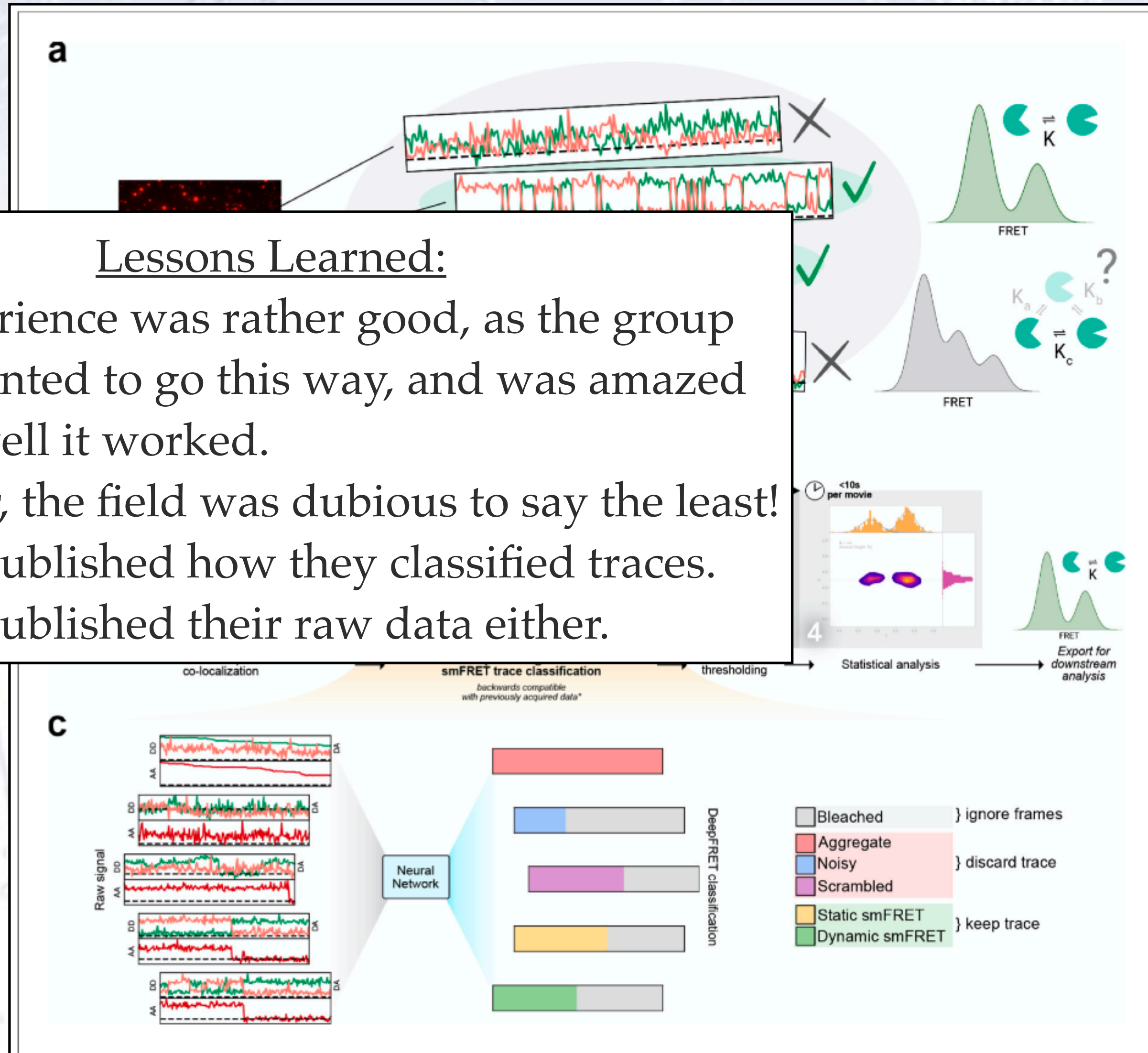


DeepFRET

FRET is a technique used to study and dynamics of biomolecules.
The data is a "trace", which is a time s with possible ph transitions.

The group woul through 10000 t select about 250 these… **by hand**!!!
This took a few people about a week, and was neither reproducible nor optimal.

So we made DeepFRET.



<u>Lessons Learned:</u>

- The experience was rather good, as the group really wanted to go this way, and was amazed at how well it worked.
- However, the field was dubious to say the least! No one published how they classified traces. No one published their raw data either.

Knee- & Hip surgery

The data

The analysis is based on V1.0 of the data:
Dtasæt_NBI_Predict_PrimæreTXA_16_17_MASTER_ WORK.csv

Dtasæt_NBI_Predict_PrimæreTXA_16_17_MASTER_ WORK

Køn	Civilstatus	Height	Weight	Hb	Hb_g_dL	Anæmi	Rygning	Alkohol	Gangredskab	Udhvilet	Snorken	DM_type	Hypertension_ja_ell_recept	Hyperkolesterol	Cardiac_disease	Pulmonary_disease	Psych_D	PsD	recept_PsD	Cerebral_attack	Tidl_VTE	Fam_VTE	AK_beh	PotentAK	Cancer	Nyre	Led	Alder	BMI	Årstal	Hospital	Medical_outcome
0	1	165	56	6,9	11,109	1	0	0	1	0	2	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	81	20,5693296602388	2016	7	0
0	0	168	77	8,2	13,202	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	27,281746031746	2017	5	0
0	1	160	70	8,7	14,007	0	0	0	1	1	2	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	81	27,34375	2017	7	0
1	0	170	85	9,5	15,296	0	0	0	0	1	0	2	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	78	29,4117647058824	2016	4	0
1	0	168	73	8,8	14,168	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	76	25,8645124716553	2016	4	0
1	0	183	95	7,8	12,558	1	0	0	0	1	1	1	1	1	1		0	0	0	0	0	0	0	0	0	0	0	77	28,3675238644868	2017	3	0
1	0	173	96	9	14,49	0	0	0	1	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	1	75	32,075912994086	2016	1	0
0	0	164	75	8	12,88	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	74	27,8851873884593	2016	4	0
1	0	169	77	8,7	14,007	0	0	1	0	1	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	73	26,9598403417247	2016	1	0
0	1	170	53	7,8	12,558	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73	18,3391003460208	2017	7	0
0	0	168	85	8,2	13,202	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	72	30,1162131519274	2017	7	0
0	1	163	63	7,8	12,558	1	0	0	0	1	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	71	23,7118446309609	2016	1	0
1	0	176	83	9,6	15,456	0	0	0	0	1	2	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	71	26,7949380165289	2016	4	0
0	0	158	80	7,9	12,719	1	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	1	0	0	0	0	0	71	32,0461464508893	2016	6	0
0	0	170	68	8,5	13,685	0	0	1		1	2	0	0	1	0	0	1	1	1	0	0		0	0	0	0	0	71	23,5294117647059	2016	7	0
0	0	157	69	7,3	11,753	1	0	0	1		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	72	27,9930220292912	2017	7	0
0	0	157	92	6,8	10,948	1	0	0	1	0	1	2	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1	72	37,3240293723883	2017	9	0

There were 10573 entries with 32 variables in the data, and we tried to give a prediction for the medical outcome (stay more than 4 nights or returning within 30 days). The data is **quite imbalanced**, with only 5.7% in one class.

We have so far used a “simple” setup (algorithm: LightGBM with focal loss), and not done a lot of optimisation… yet!

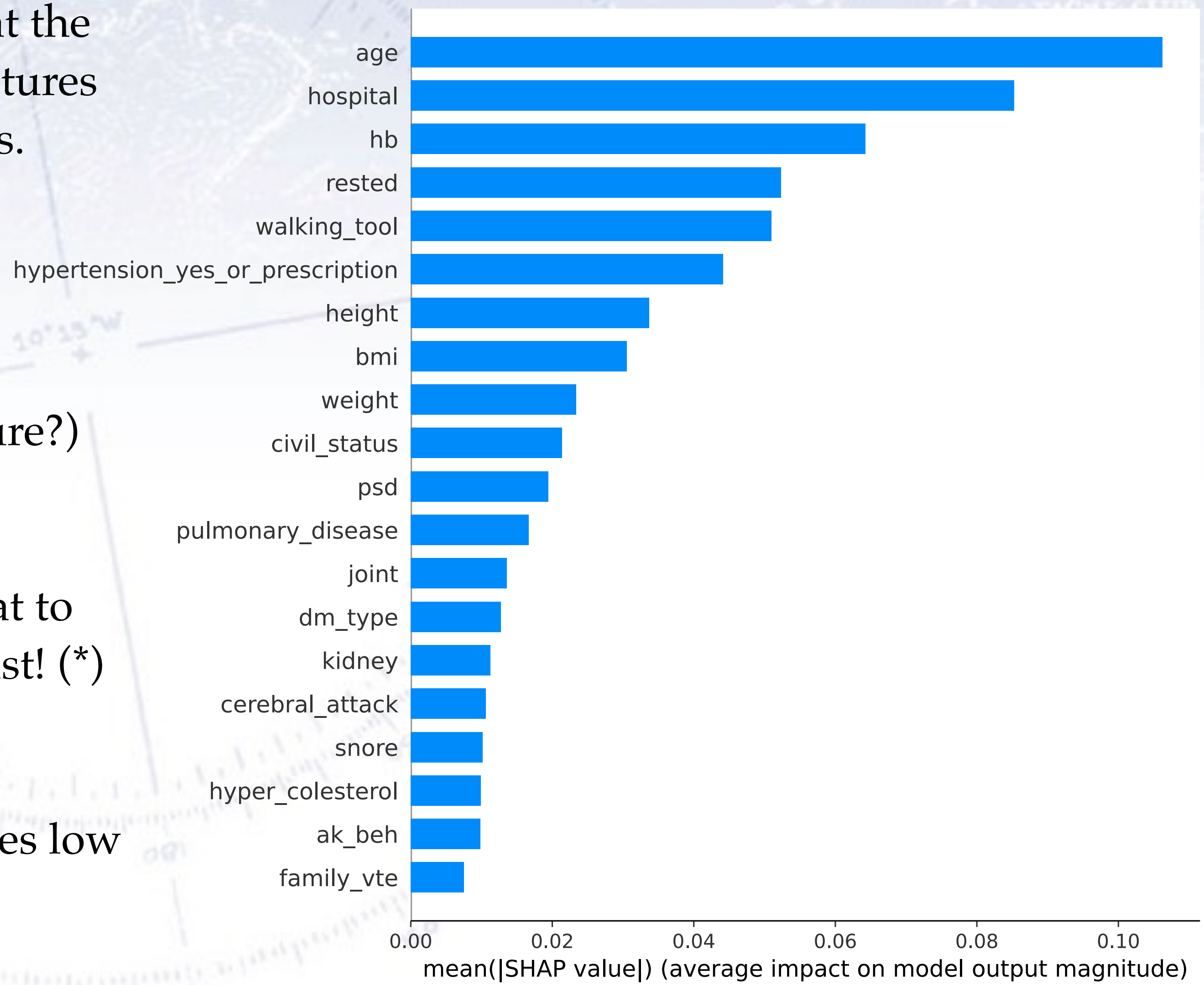
Ranking of features

Here we show what the most important features were in the analysis.

Age is no surprise!
HB (= blood pressure?) also ranks high.

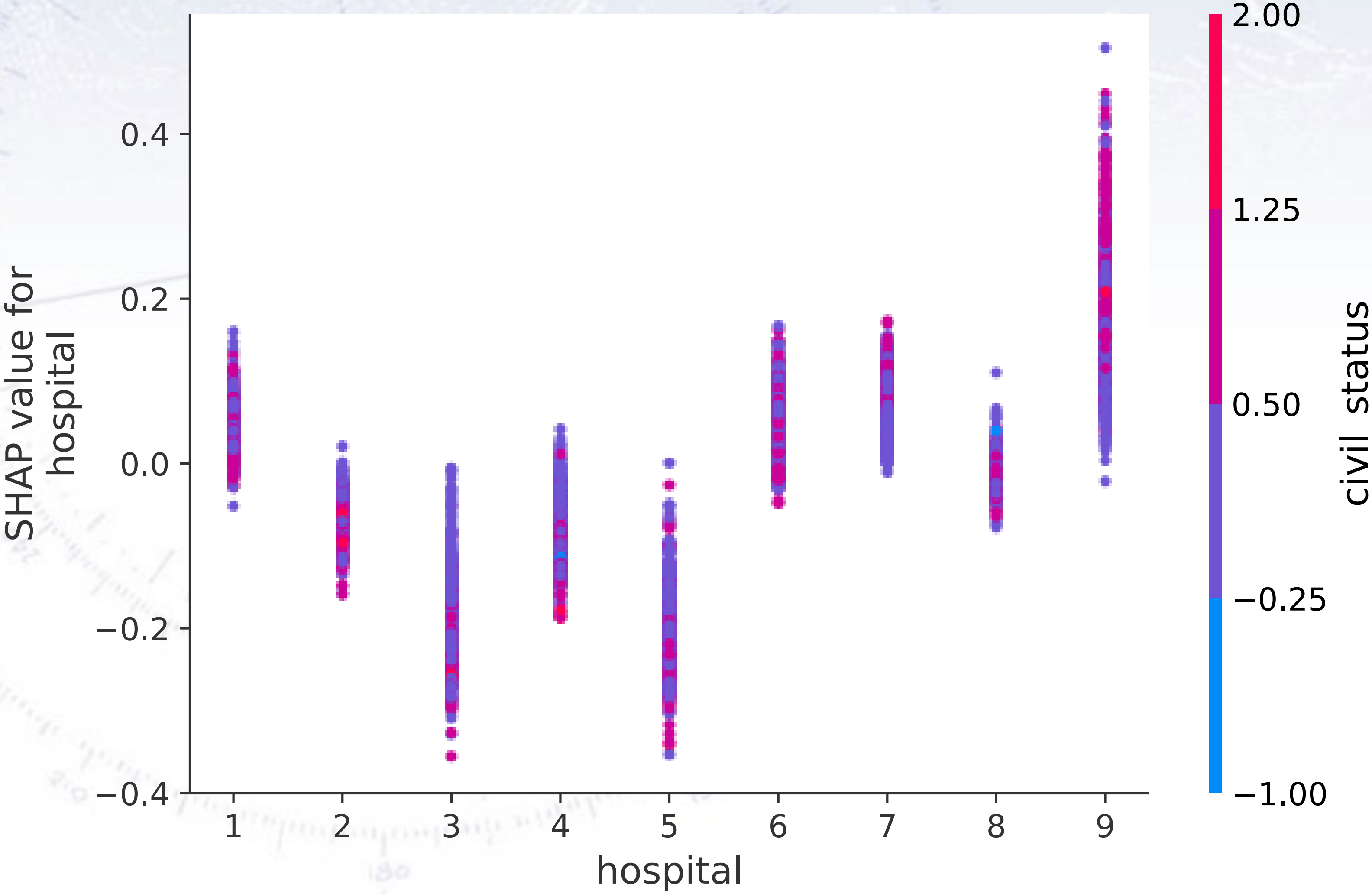
Hospital is not great to see so high in the list! (*)

Also good is to see "snore" and the likes low in the list.



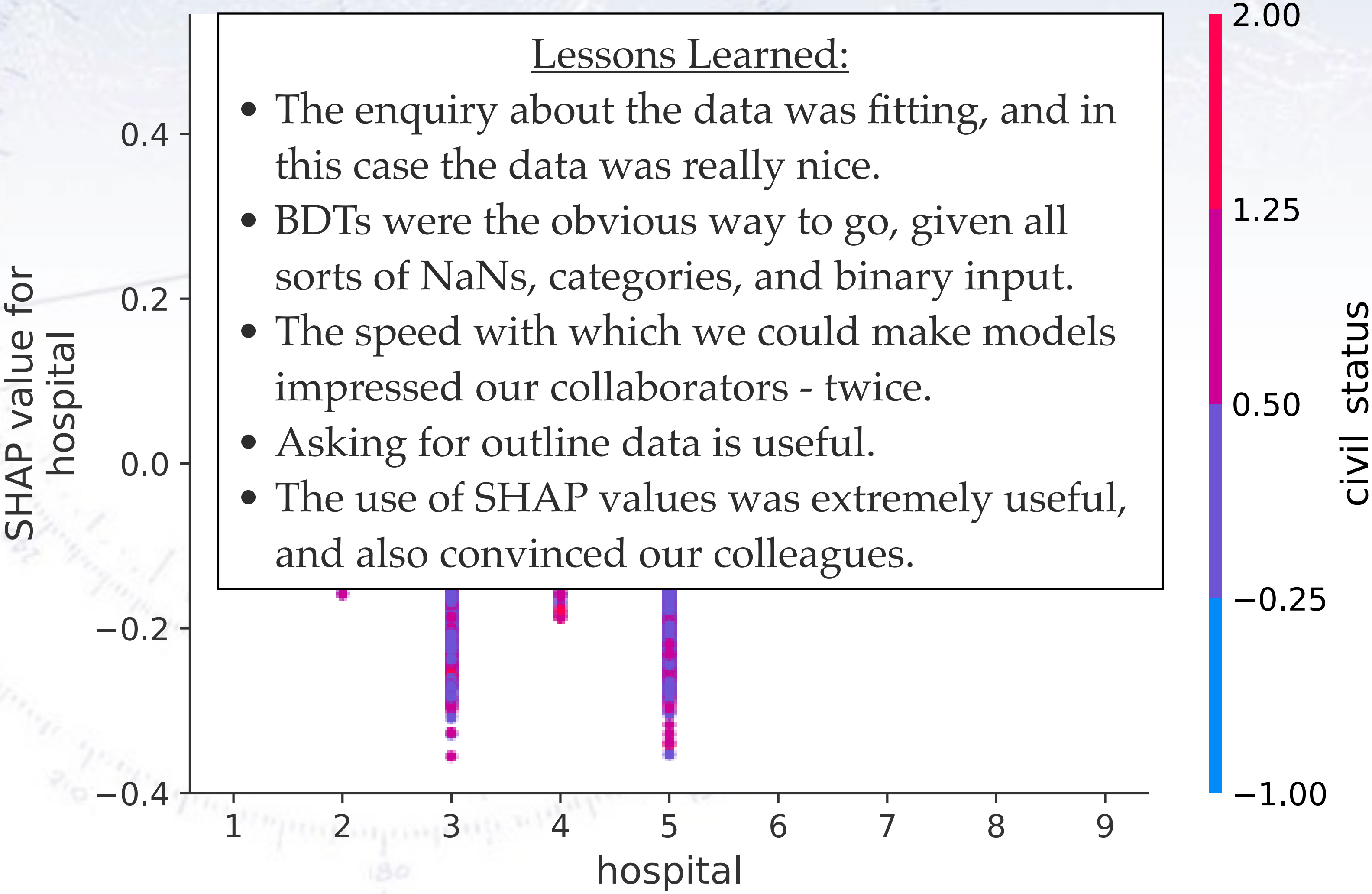
Further improvements

We don’t know which is “Hospital=9”, but we don’t want to send Mathias there!



Further improvements

We don’t know which is “Hospital=9”, but we don’t want to send Mathias there!



Lessons Learned:

- The enquiry about the data was fitting, and in this case the data was really nice.
- BDTs were the obvious way to go, given all sorts of NaNs, categories, and binary input.
- The speed with which we could make models impressed our collaborators - twice.
- Asking for outline data is useful.
- The use of SHAP values was extremely useful, and also convinced our colleagues.

Zooming Out

Conclusions

Machine Learning is a great new tool, but of course comes with caveats:

- Remember the **context** and goal:

  You might do something great… to no avail!

Conclusions

Machine Learning is a great new tool, but of course comes with caveats:

- Remember the **context** and goal:
  You might do something great… to no avail!
- Start with a **simple** model, and then expand.
  Maybe the simple model is best/enough.

Conclusions

Machine Learning is a great new tool, but of course comes with caveats:

- Remember the **context** and goal:

  You might do something great… to no avail!

- Start with a **simple** model, and then expand.

  Maybe the simple model is best/enough.

- Ensure **reproducibility**:

  Save hyperparameters, models, and results.

Conclusions

Machine Learning is a great new tool, but of course comes with caveats:

- Remember the **context** and goal:

  You might do something great… to no avail!

- Start with a **simple** model, and then expand.

  Maybe the simple model is best/enough.

- Ensure **reproducibility**:

  Save hyperparameters, models, and results.

- Beware of domain shifts:

  Simulated and real data are **never** the same.

Conclusions

Machine Learning is a great new tool, but of course comes with caveats:

- Remember the **context** and goal:
  You might do something great… to no avail!
- Start with a **simple** model, and then expand.
  Maybe the simple model is best/enough.
- Ensure **reproducibility**:
  Save hyperparameters, models, and results.
- Beware of domain shifts:
  Simulated and real data are **never** the same.
- All the **old rules apply**:
  Inspect, Check, and Question data and output.

Conclusions

Machine Learning is a sharpening of our scientific senses - not a substitution