IDAO 2021: Baobab solution

Oleg Filatov[1], Andrey Filatov[2,3], Andrey Znobishchev[4]

[1] Deutsches Elektronen-Synchrotron
[2] Moscow Institute of Physics and Technology
[3] École polytechnique fédérale de Lausanne
[4] Skolkovo Institute of Science and Technology

Outline

- About IDAO
- Physical motivation
- CYGNO experiment
- Data format
- Competition problem
- Our solution
- Results & summary

About

- **International Data Analysis Olympiad**
  - 4th time this year
  - Organised by Higher School of Economics & Yandex
- **Broad & diverse community**
  - 2187 participants/78 countries (2019)
  - 2756 participants/83 countries (2020)
  - ~1500 participants/66 countries (this year)
- **2 stages:**
  - 1st: qualification round (all teams, 1 month)
  - 2nd: final (30 best teams, 2 days)
- **Traditionally, 1st stage problem comes from physics**
  - Muon ID @LHCb (2019)
  - Space object location (2020)
  - Dark matter searches (this year)

IDAO
INTERNATIONAL DATA ANALYSIS OLYMPIAD

HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

Yandex

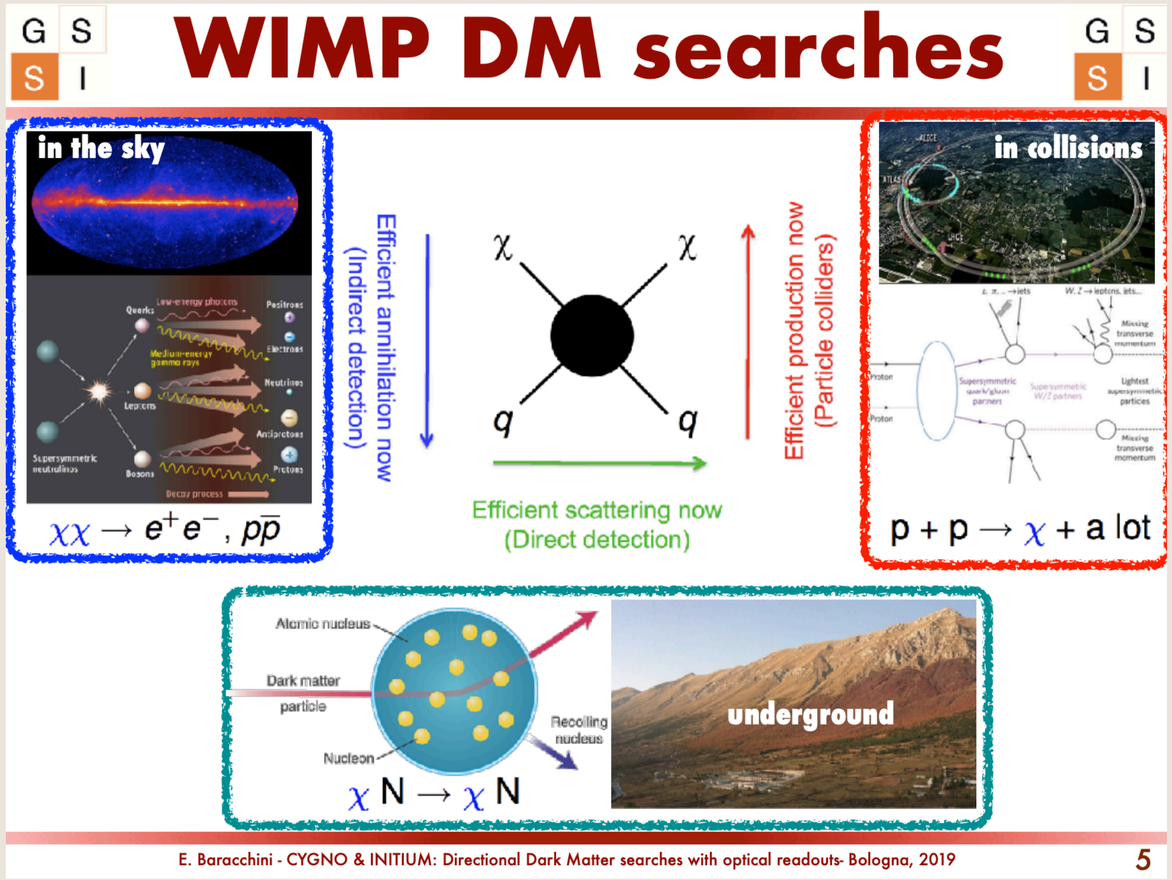
Abdalaziz Rashid Al-Maeeni
Task creator, Research Assistant at the Laboratory of Methods for Big Data Analysis (LAMBDA), HSE University

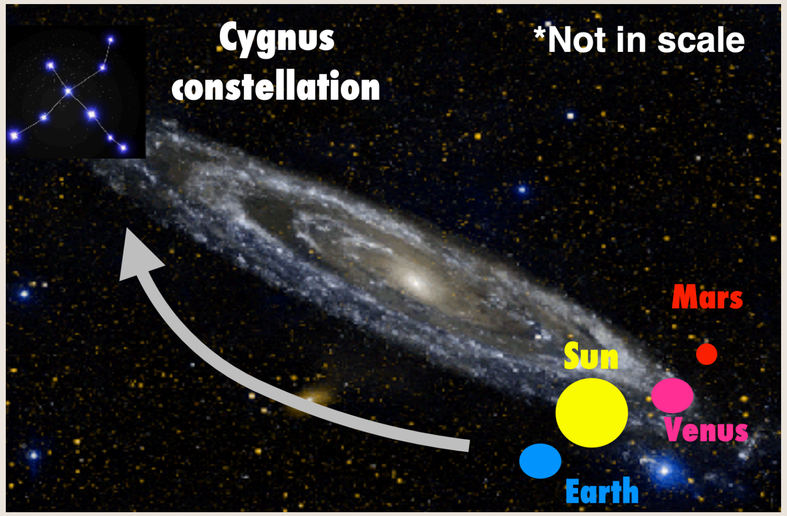
Giulia D'Imperio
Task creator, Researcher at INFN Sezione di Roma

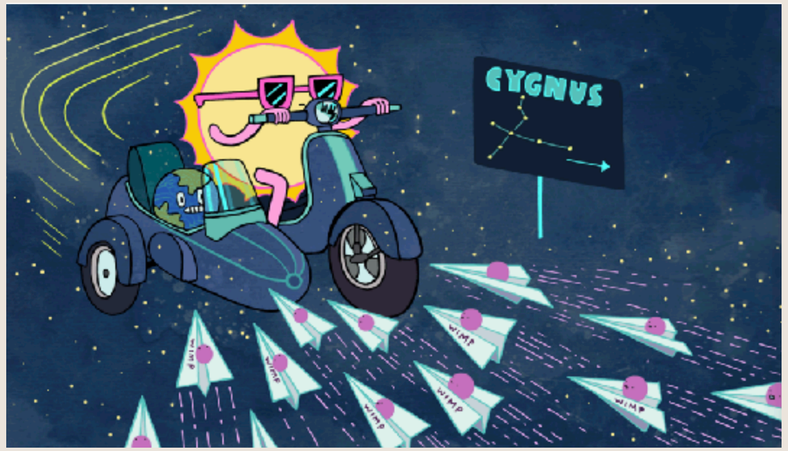
https://idao.world/

Physics

- Dark Matter nature still puzzling
- Can try to search for particle candidate
- Multiple ways, here focus on direct search
- Idea: detect DM by observing particles' recoil







slides from this talk

Non-relativistic collision

WIMP
from galactic halo
v~220 km/s

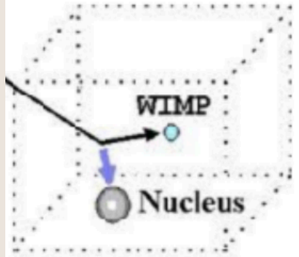
Target Nucleus
in laboratory
v~0 km/s

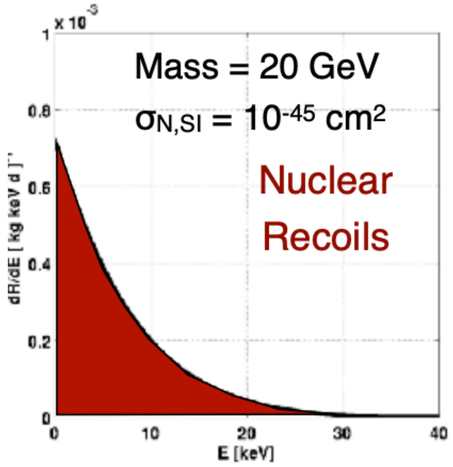
Elastic collision

WIMP
θ_R
E_R~30 keVr

Recoiling nuclei (partially) retain WIMP direction

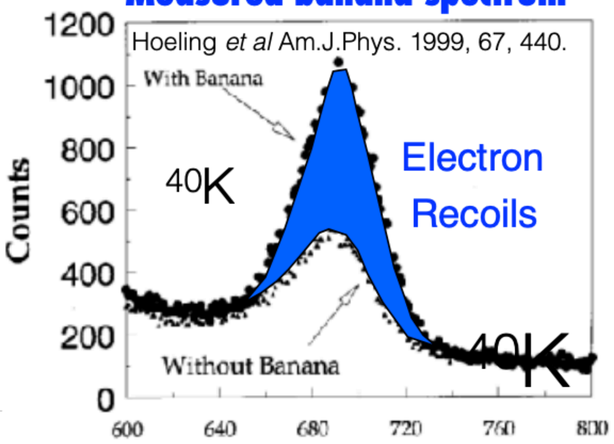
$$R \sim 0.13 \frac{\text{events}}{\text{kg year}} \left[\frac{A}{100} \times \frac{\sigma_{WN}}{10^{-38}\ \text{cm}^2} \times \frac{\langle v \rangle}{220\ \text{km s}^{-1}} \times \frac{\rho_0}{0.3\ \text{GeV cm}^{-3}} \right]$$





~1 events/kg/year

Measured banana spectrum



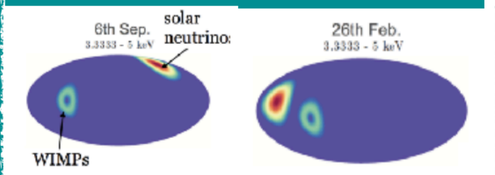
~100 events/kg/s

Interaction rate is extremely low & backgrounds extremely high

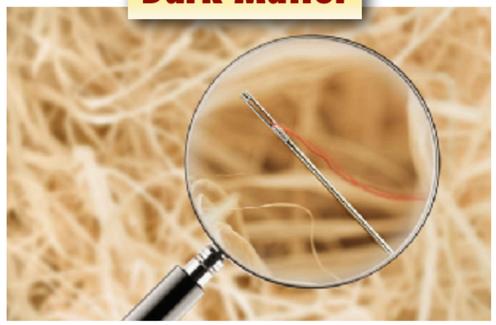
Main backgrounds

Neutrinos from Sun, atmospheric and diffuse supernovae

Only if you can measure track direction (i.e. correlate with the source), only for Solar Neutrinos

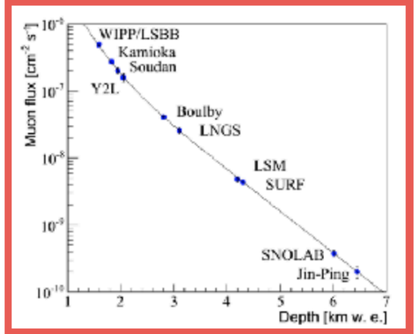


Dark Matter



Cosmic rays & cosmogenic activation of detector materials

Go underground + active/passive shielding + active recoil/e^- discrimination (not available to all exp.)

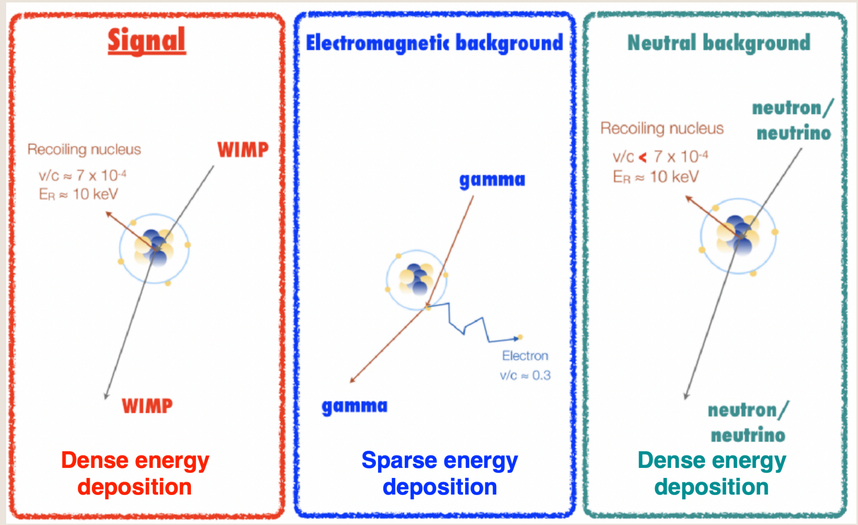


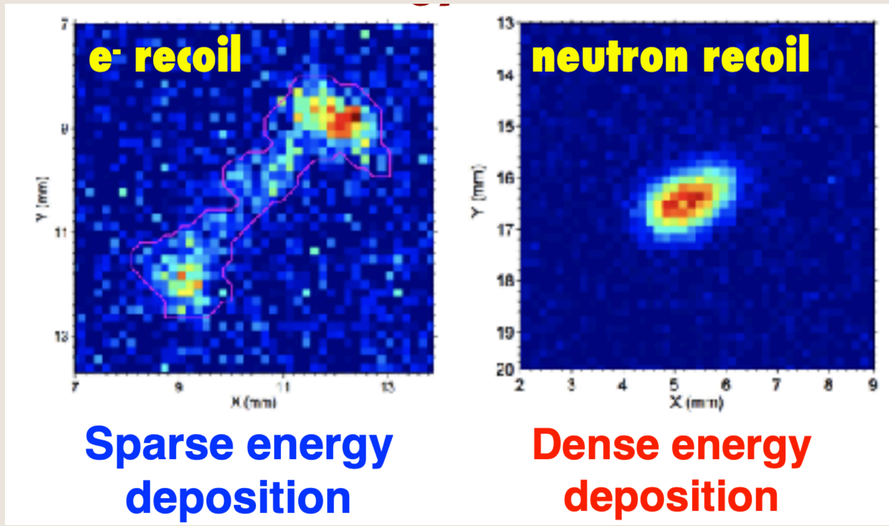
Natural radioactivity from ^{238}U, ^{232}Th, ^{222}Rn, ^{40}K...

Material selection & screening + active recoil/e^- discrimination (not available to all exp.)

Rn progeny recoils on detectors surfaces

Measure 3D position of your event: "fiducialization" (not available to all exp.)





This year's task

CYGNO experiment

PHASE_1: detector concept

Gas Electron Multipliers (GEMs) amplification

Electron Microscopy of a GEM Foil

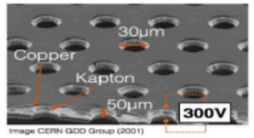
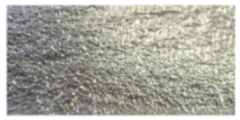
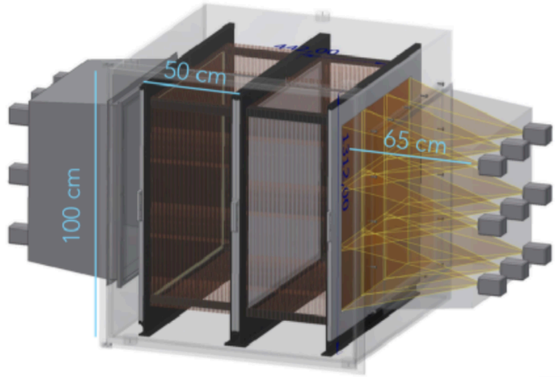


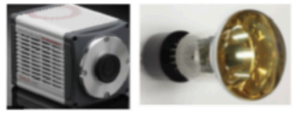
Image CERN GDD Group (2001)

Transparent texturised mylar cathode a'la DRIFT



*gammas & neutrons shielding not shown but present





PMT + sCMOS optical readout decoupled from target volume

Atmospheric pressure & room temperature: minimal infrastructure

18 cameras monitoring 330*330 mm each with **160 mμ** resolution

A total of 72 10^6 readout 165 x 165 μm^2 pixels

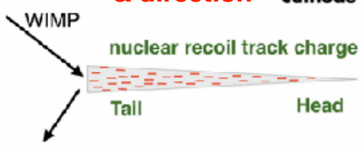
Year	Rich. (k€)
(CSN5) 2018	29
(TDR) 2019	89
2020	237
2021	284
2022	83
Tot (20-22)	604

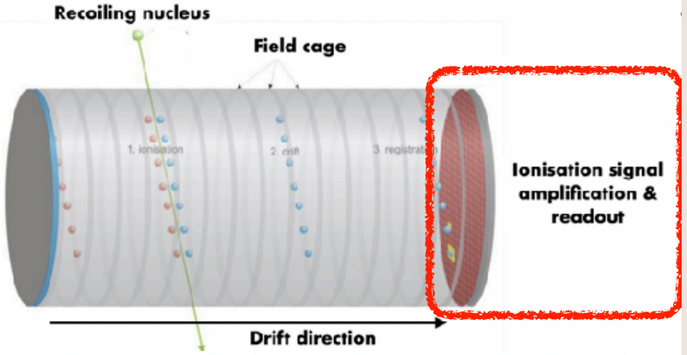
Active contribution from several CYGNUS-TPC members

E. Baracchini - CYGNO & INITIUM: Directional Dark Matter searches with optical readouts- Bologna, 2019 43

Gaseous Time Projection Chamber, inherently a 3D detector

Sensitive to track sense & direction

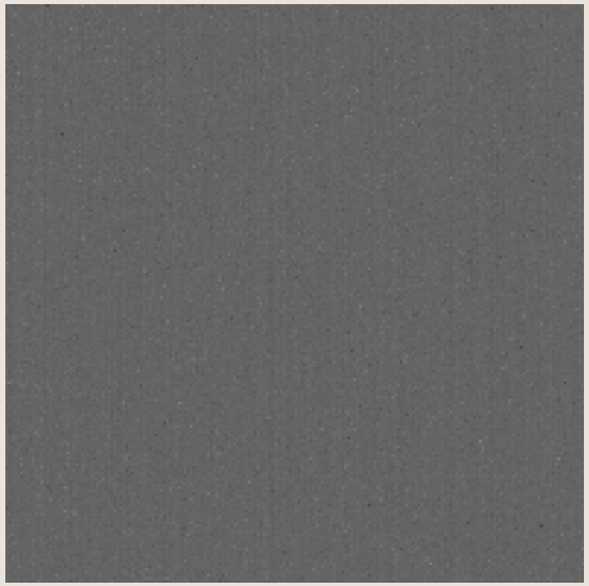


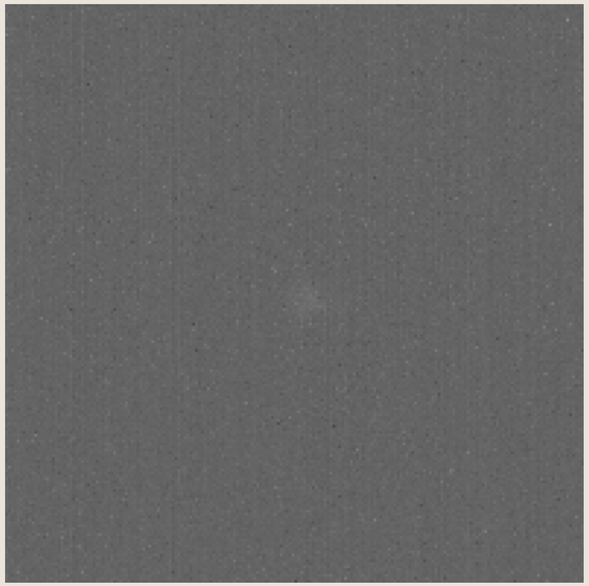


Data

RAW data just simple images!

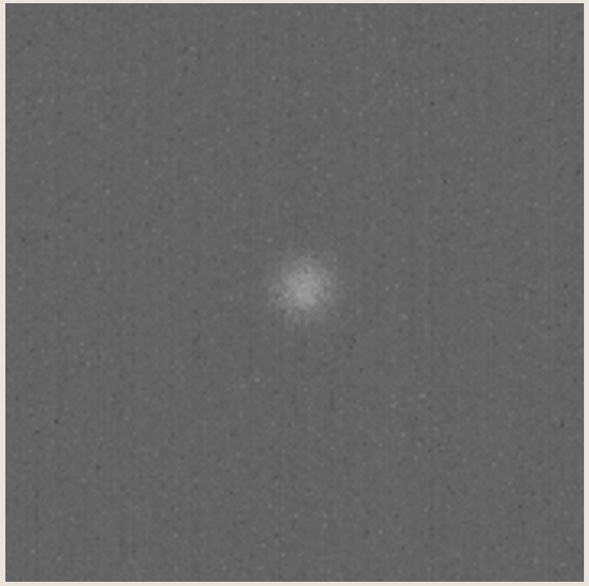
Nuclear recoil (NR), 1 keV

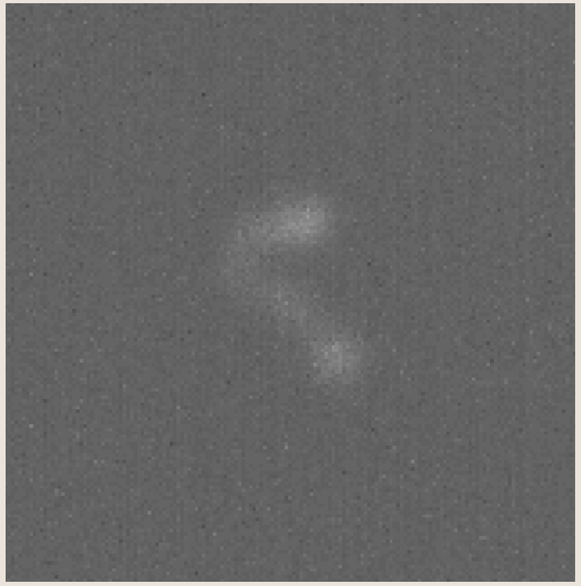




Electron recoil (ER), 3 keV

Nuclear recoil (NR), 20 keV





Electron recoil (ER), 30 keV

Competition

- **Goal:** given image, predict its event class & event energy
  - Event class: ER/NR
  - Event energy: see table
  - Note significant difference btw. training and test domains!
- **Samples:**
  - **Train:** 2200 images/training class, 2 samples/test class ← labelled
  - **Public test:** 250 images/training class ← unlabelled, "black box" evaluated
  - **Private test:** 2500 images/test class ← unlabelled, evaluated after deadline
- **Metric:** (AUC - MAE)*1000 on private test sample
  - For public test displayed on leaderboard
  - For private test used in the final teams' evaluation
- **Two tracks:**
  - Track 1: get the highest score
  - Track 2: get the highest score + limits on memory (1Gb) & runtime (15 min)

#	Participant	A 1092/2430	B 138/1039	Score
126	Baobab	-1537.93 30d. 10h.	-2295.33 30d. 10h.	-3833.26
1	random team	998.00 30d. 9h.	1000.00 30d. 10h.	1998.00
2	White Material	997.34 26d. 3h.	1000.00 26d. 6h.	1997.34
3	fit_predict	996.00 30d. 10h.	1000.00 30d. 10h.	1996.00
4	DataCrackers	996.67 30d. 5h.	996.67 27d. 7h.	1993.34
5	CooperFactory	994.67 26d. 11h.	998.00 28d. 12h.	1992.67

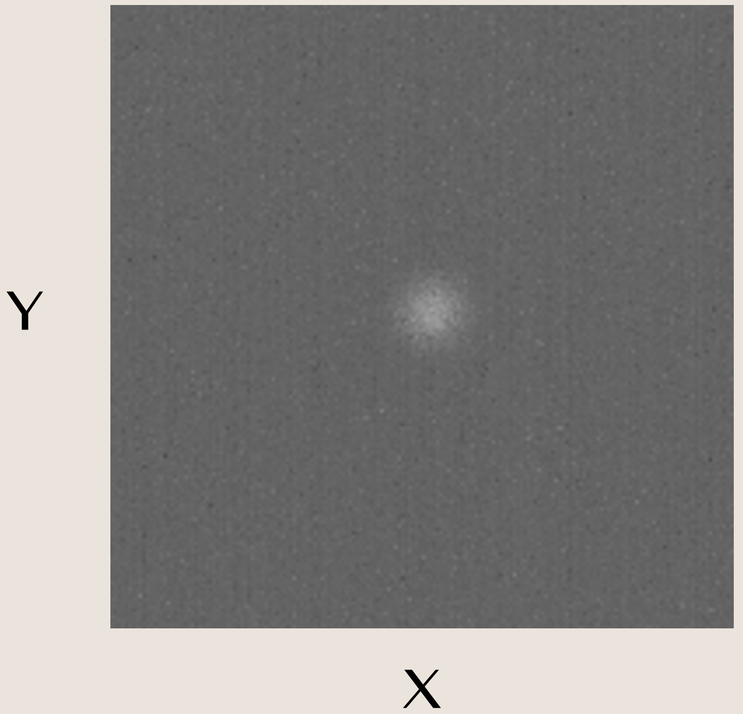
Energy, keV	He	e
1	*	-
3	-	*
6	*	-
10	-	*
20	*	-
30	-	*

* is training; - is testing

Our solution

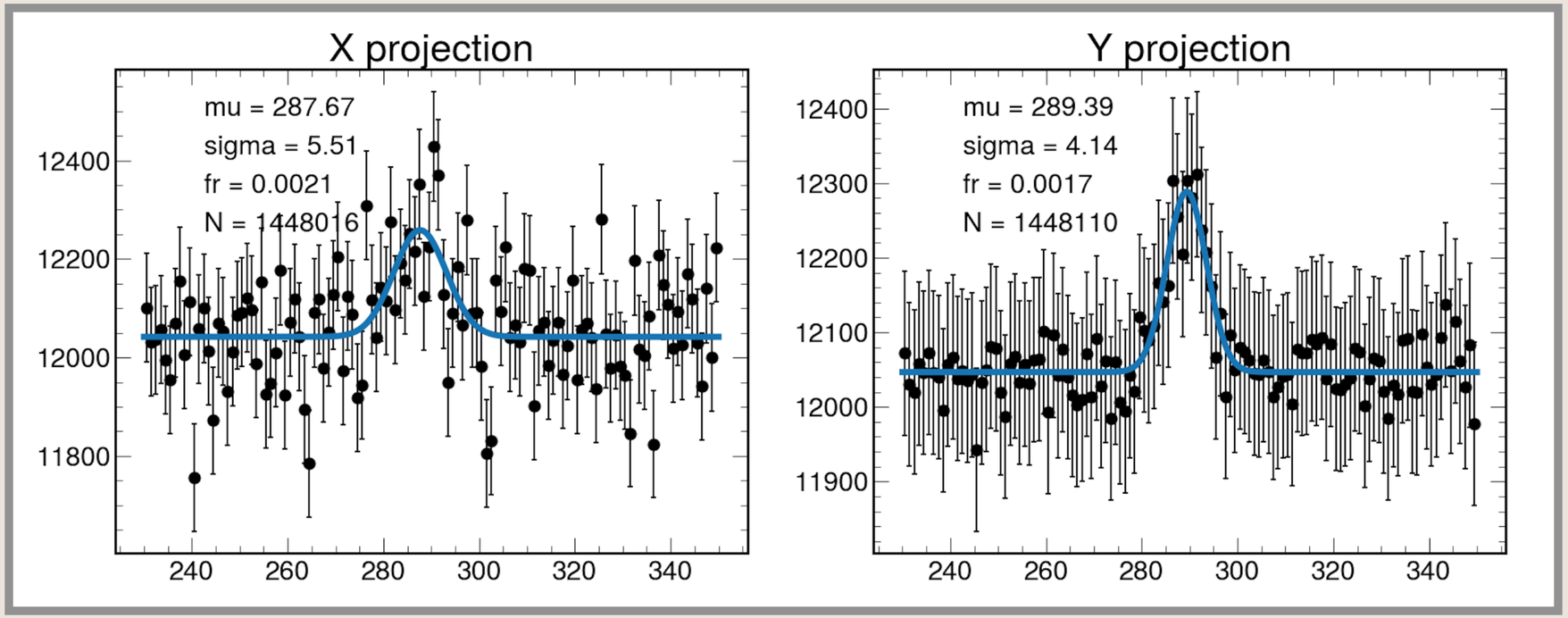
Feature engineering

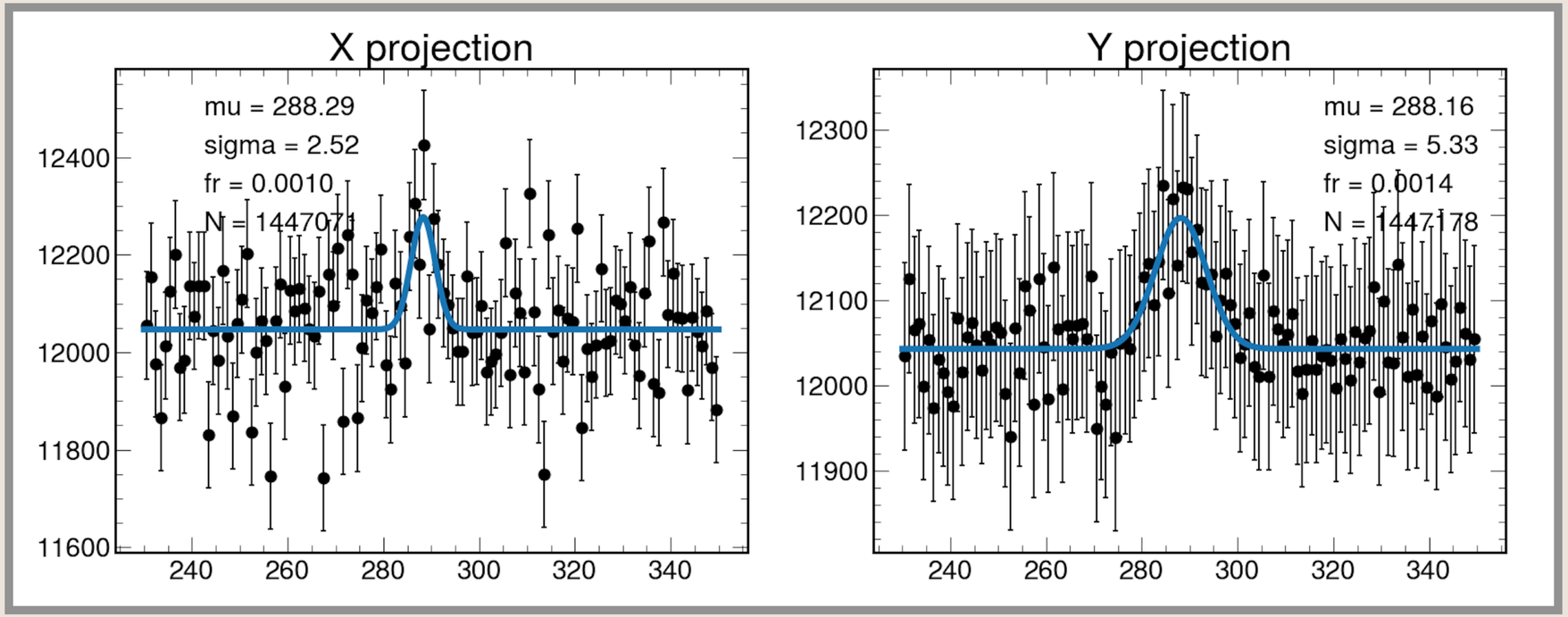
- **Idea:** extract knowledge from images by constructing informative features
- **Observation:** images look like 2D Gaussian function sitting on top of flat background
- **Approach:** fit separately X/Y projections with a mixture of flat and normal distributions*
- **Final features:** parameters of the fit
  - μ, σ, fr, counts (sig,bkgr) + their errors : X,Y projections
  - dsigma, dfr, dmu: parameter difference between X and Y fits
  - abs_dmu = abs(mu - 288): X,Y projections
  - χ^2, p-value under bkgr & sig+bkgr hypotheses: X,Y projections
  - fit convergence info (e.g. HESSE_valid)
- **Note:** in the end we didn't use all of them



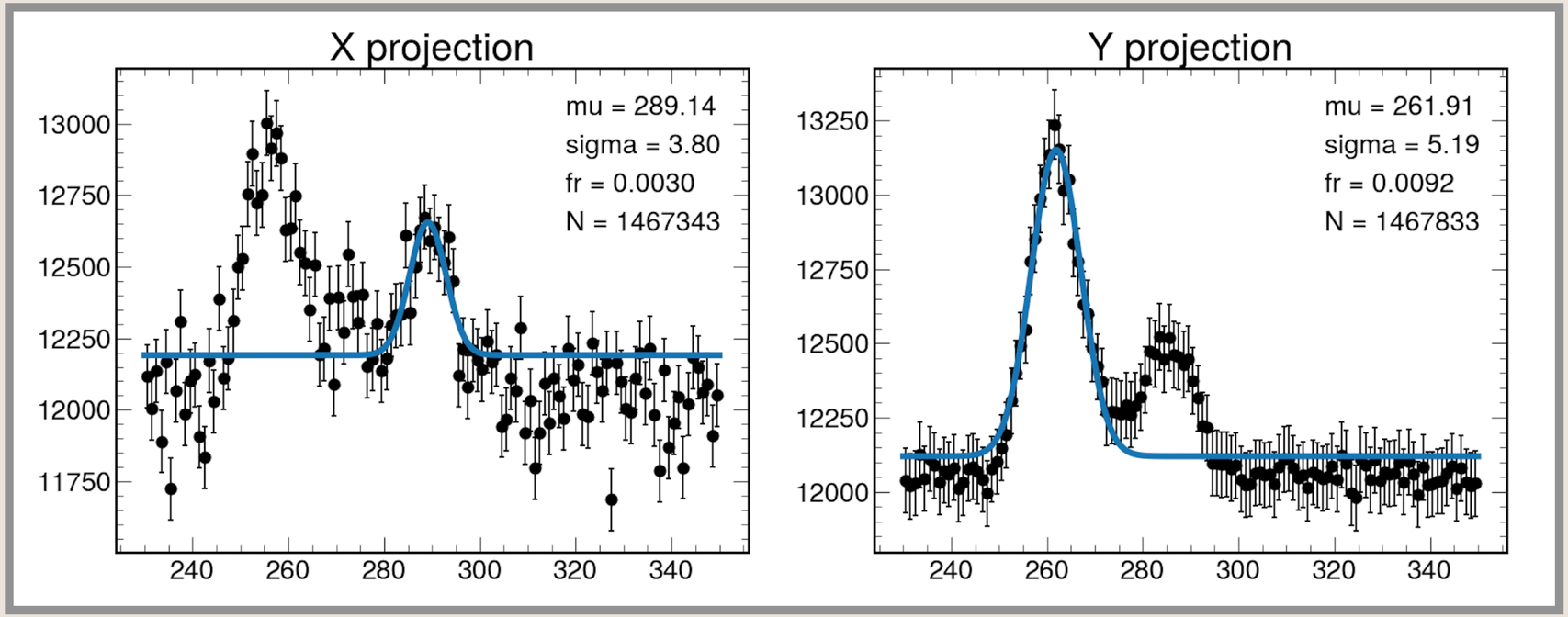
*using custom implementation of χ^2 binned fit in iMinuit speeded up by numba

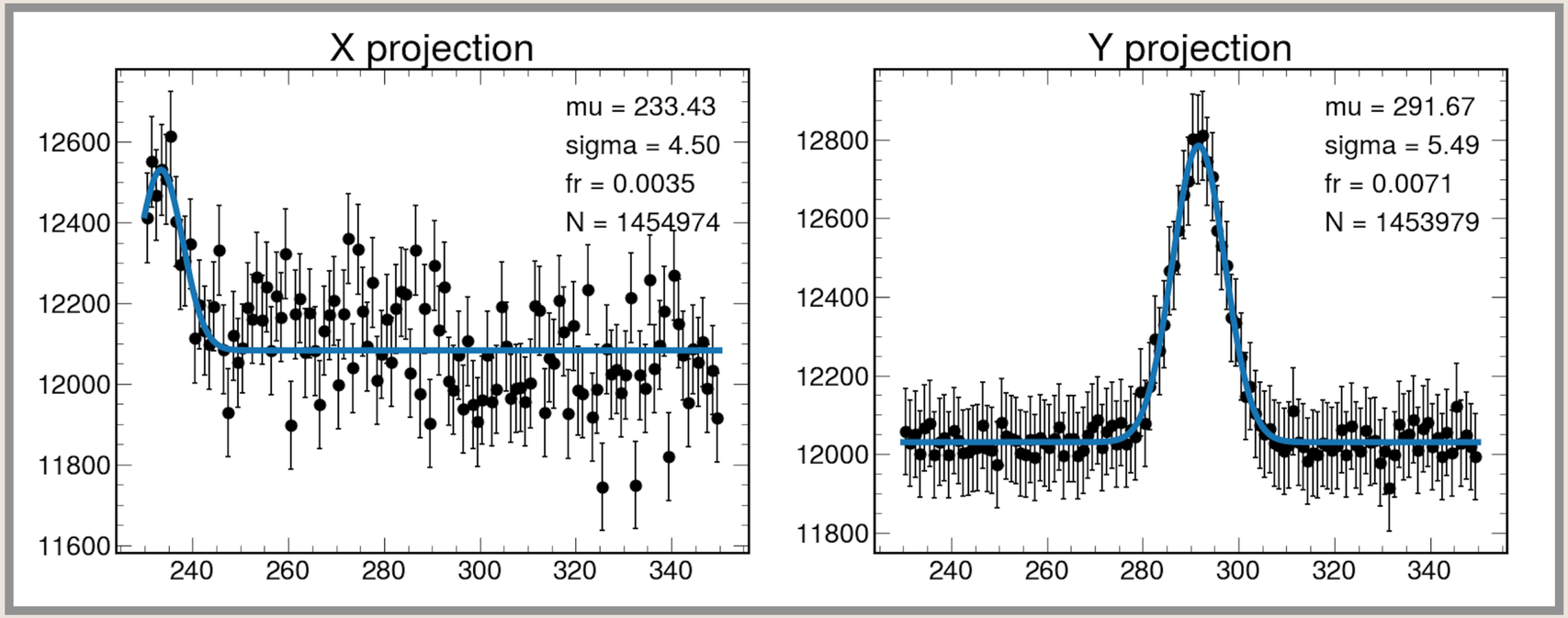
ER, 3 keV


X projection
mu = 287.67
sigma = 5.51
fr = 0.0021
N = 1448016
Y projection
mu = 289.39
sigma = 4.14
fr = 0.0017
N = 1448110

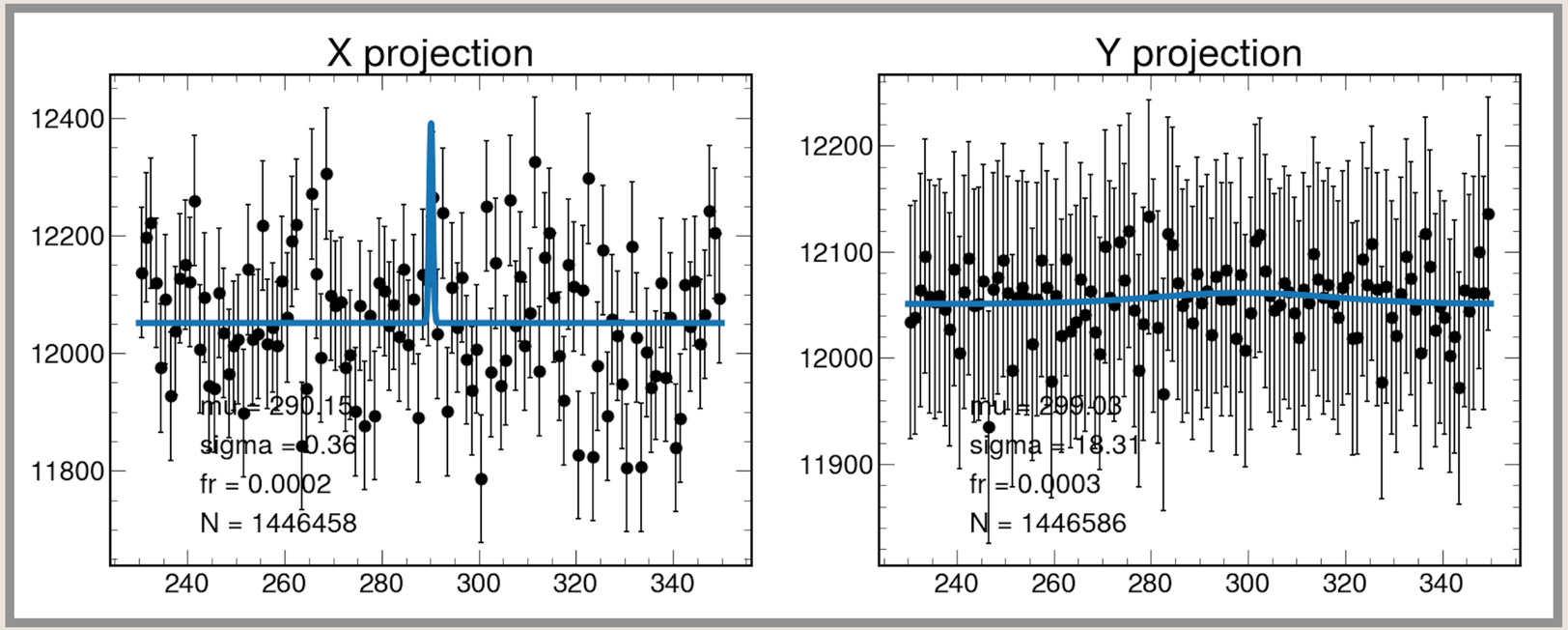

X projection
mu = 288.29
sigma = 2.52
fr = 0.0010
N = 1447071
Y projection
mu = 288.16
sigma = 5.33
fr = 0.0014
N = 1447178

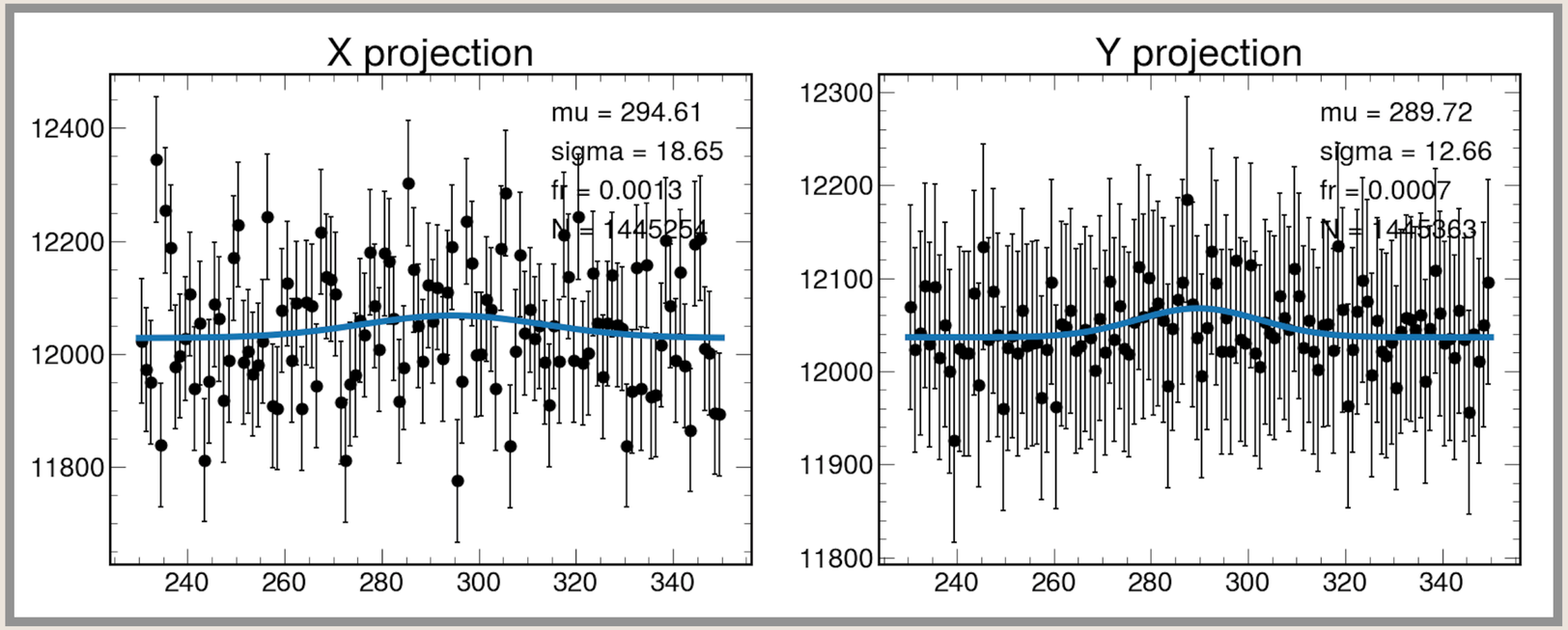
ER, 30 keV


X projection
mu = 289.14
sigma = 3.80
fr = 0.0030
N = 1467343
Y projection
mu = 261.91
sigma = 5.19
fr = 0.0092
N = 1467833

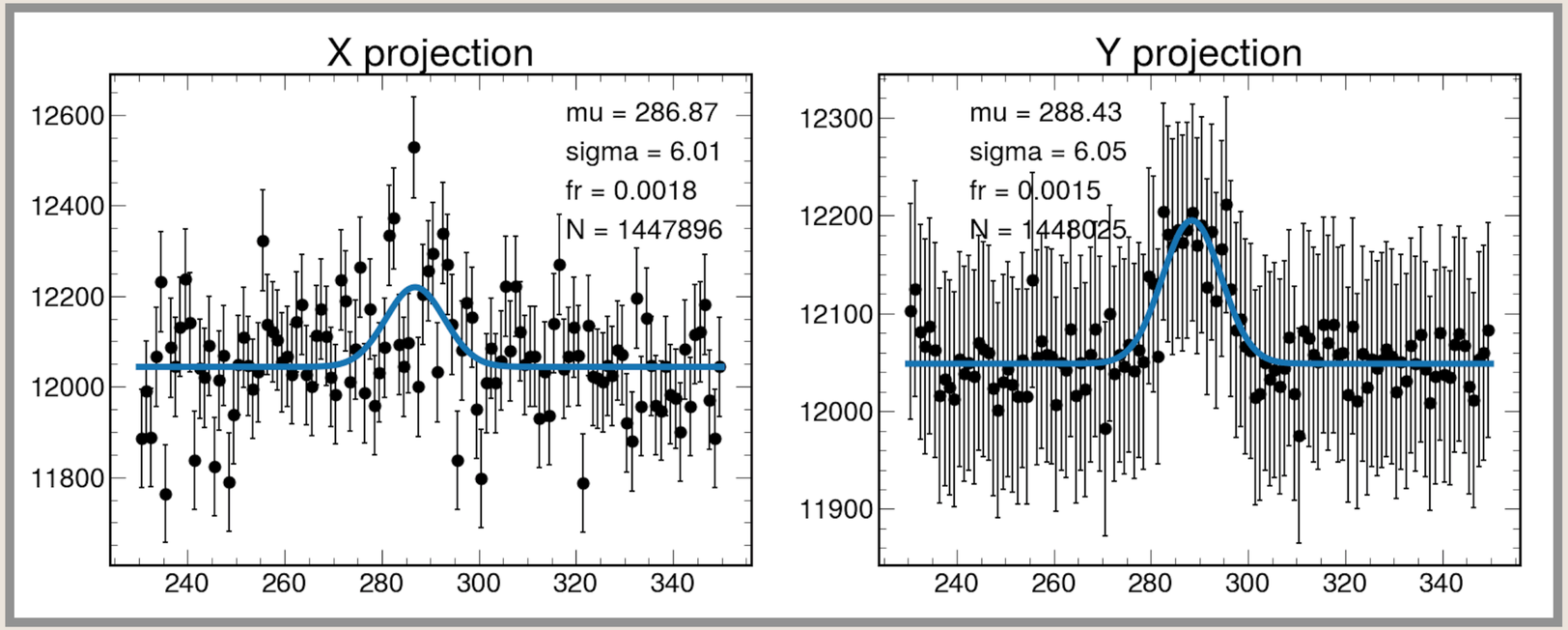

X projection
mu = 233.43
sigma = 4.50
fr = 0.0035
N = 1454974
Y projection
mu = 291.67
sigma = 5.49
fr = 0.0071
N = 1453979

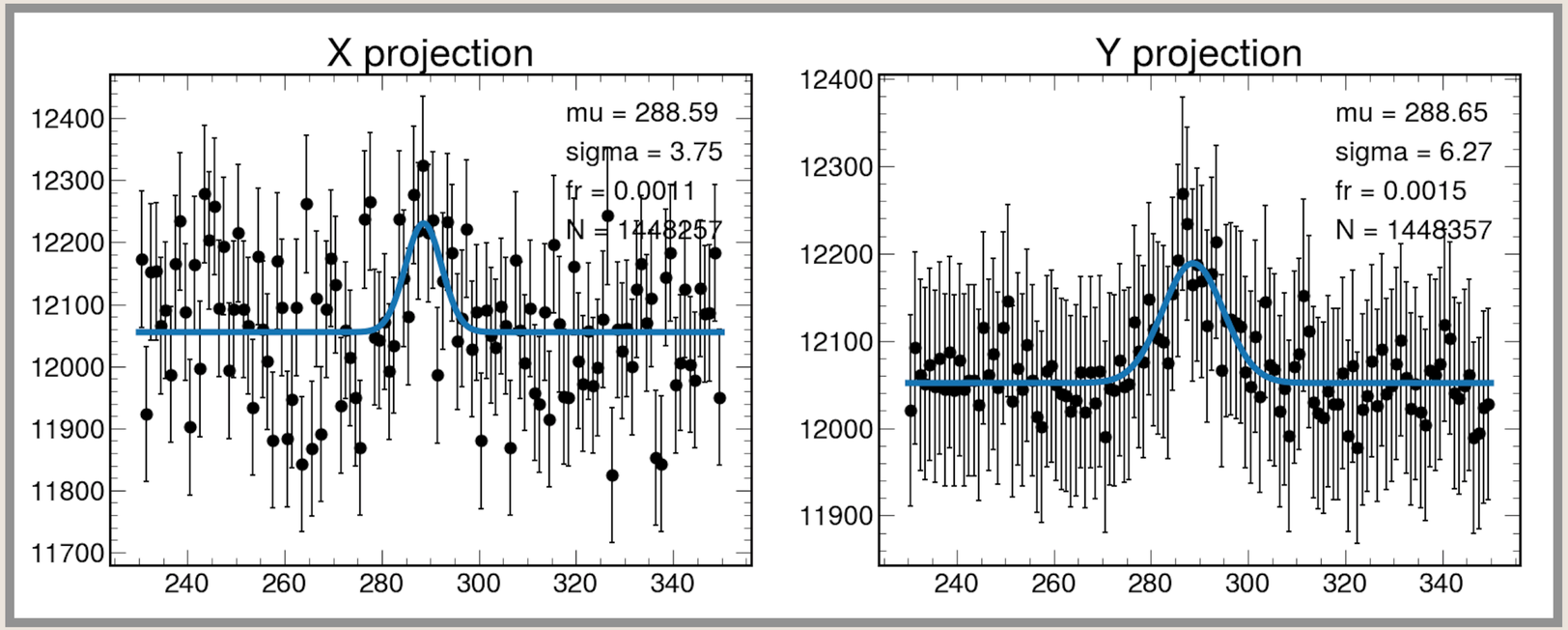
NR, 1 keV


X projection
mu = 290.15
sigma = 0.36
fr = 0.0002
N = 1446458
Y projection
mu = 299.03
sigma = 18.31
fr = 0.0003
N = 1446586


X projection
mu = 294.61
sigma = 18.65
fr = 0.0013
N = 1445254
Y projection
mu = 289.72
sigma = 12.66
fr = 0.0007
N = 1445363

NR, 6 keV


X projection
mu = 286.87
sigma = 6.01
fr = 0.0018
N = 1447896
Y projection
mu = 288.43
sigma = 6.05
fr = 0.0015
N = 1448025


X projection
mu = 288.59
sigma = 3.75
fr = 0.0011
N = 1448257
Y projection
mu = 288.65
sigma = 6.27
fr = 0.0015
N = 1448357

Our solution

First steps

- **Observation:** easy to get high score on public, but that's pointless
  - e.g. by building good model on training sample - they have similar domains
  - Teams are evaluated on private →no sense to fight for leaderboard positions
- Instead, **crucial to do good job on private sample**
  - However, sizeable differences in two domains
  - Meaning model trained on training sample expected to perform worse on private
  - And as I'll show later, the difference is indeed problematic

#	Participant	A 1092/2430	B 138/1039	Score
126	Baobab	-1537.93 30d. 10h.	-2295.33 30d. 10h.	-3833.26
1	random team	998.00 30d. 9h.	1000.00 30d. 10h.	1998.00
2	White Material	997.34 26d. 3h.	1000.00 26d. 6h.	1997.34
3	fit_predict	996.00 30d. 10h.	1000.00 30d. 10h.	1996.00
4	DataCrackers	996.67 30d. 5h.	996.67 27d. 7h.	1993.34
5	CooperFactory	994.67 26d. 11h.	998.00 28d. 12h.	1992.67

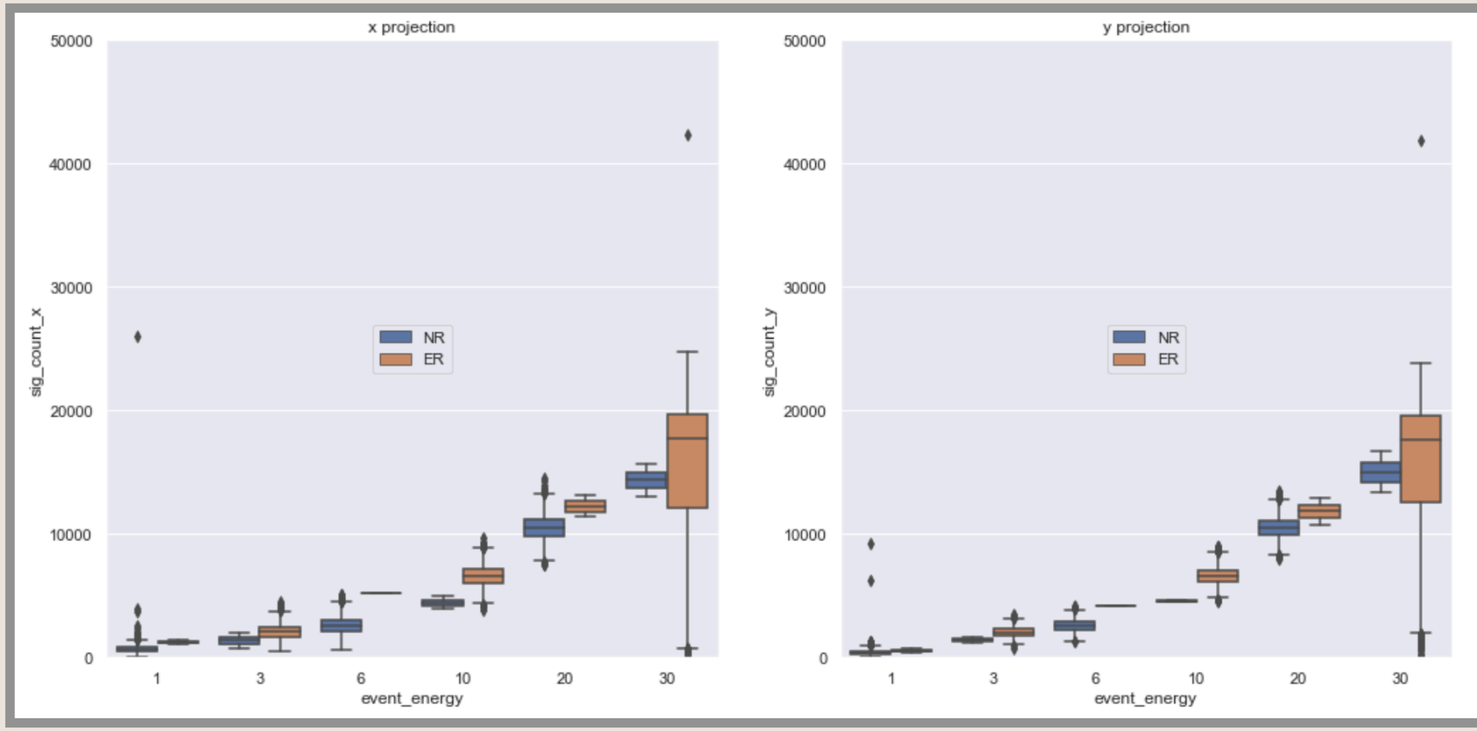
Energy, keV	He	e
1	*	-
3	-	*
6	*	-
10	-	*
20	*	-
30	-	*

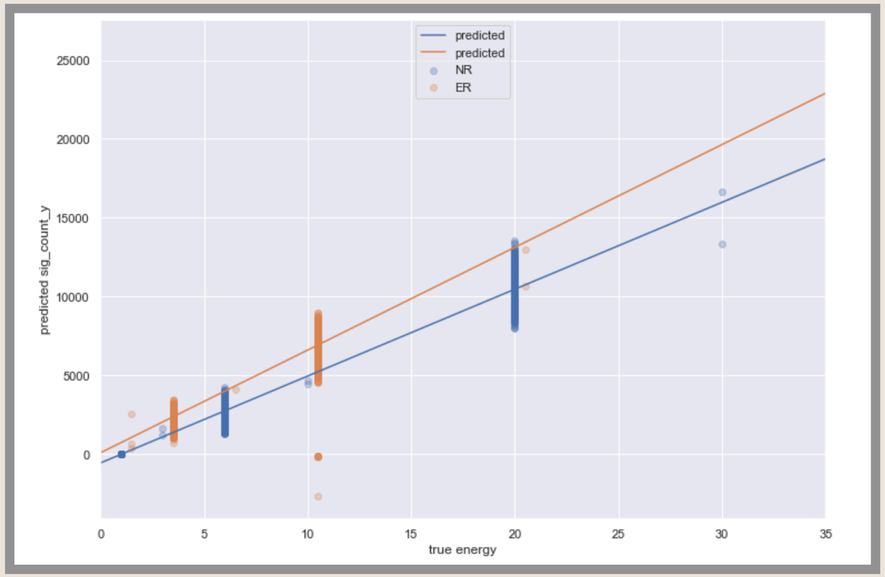
* is training; - is testing

Our solution

Calibration curve

- **Initially, thought that domain difference isn't that big**
- **Focused firstly on the following idea:**
  - It is clear that `sig_count` should be highly correlated with `true_energy`
  - Can try to build a simple regression on that → calibration curve (`E_regr`)
  - Using this curve we can label the data
    - Class prediction can be inferred from predicted energy
- **NB:** we decided to use Y projection for that, since it turned out to provide better resolution at low energies
- **NB:** hereafter use NR curve coefficients for calibration
  - Basically, mapping: `sig_count→E_regr`

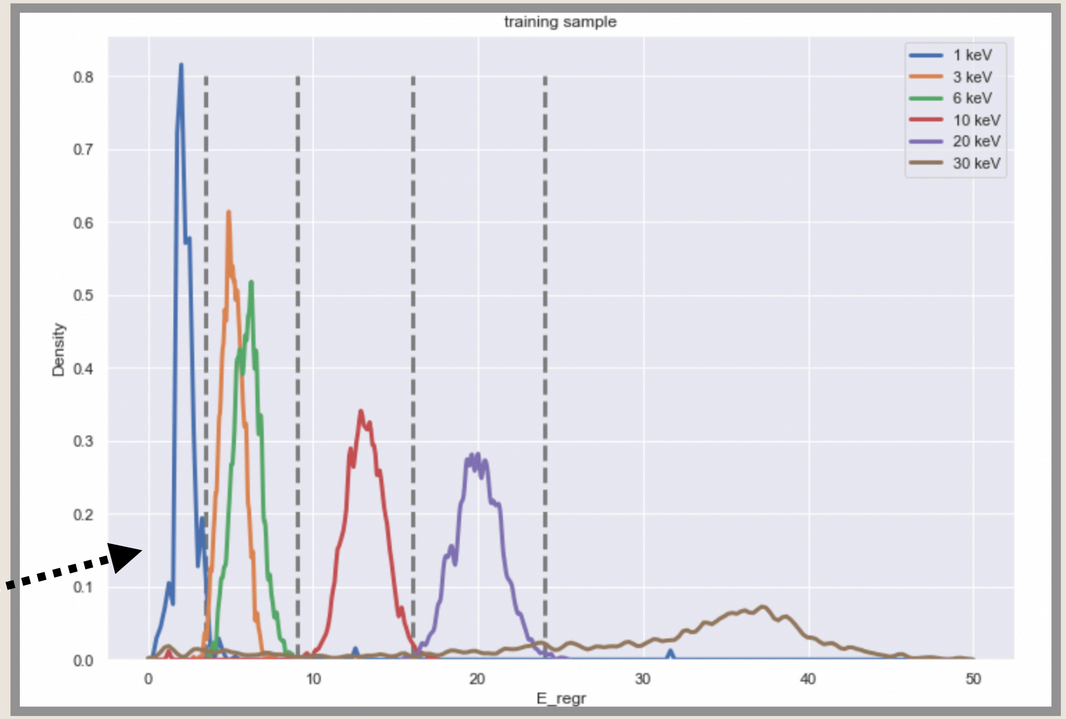


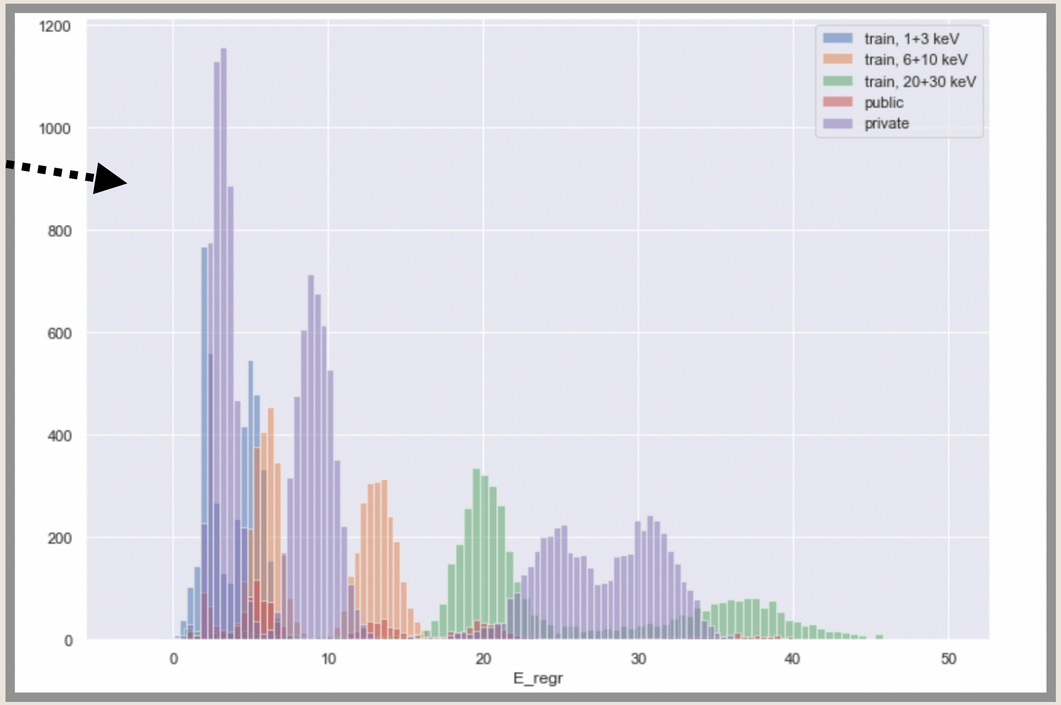


Our solution

From `sig_count` to `E_regr`

- **We didn't want to predict "soft" energies**
  - But rather round them to those which appear in data sample
  - Makes sense, reduces MAE if one is confident in prediction
- **This would require applying a cut on `E_regr`**
  - Kinda easily separated on training sample
  - Basically, no need for ML (unless wanna be super-accurate)
- **Decided to plot `E_regr` distribution for private sample**
- ***horror***
  - Not only the peaks are shifted (expected)
  - They also heavily overlap: e.g. ER (3) vs NR (6), ER (10) vs NR (20)
  - Any classifier trained on training sample would be literally useless on private
- **This made us rethink the whole strategy**

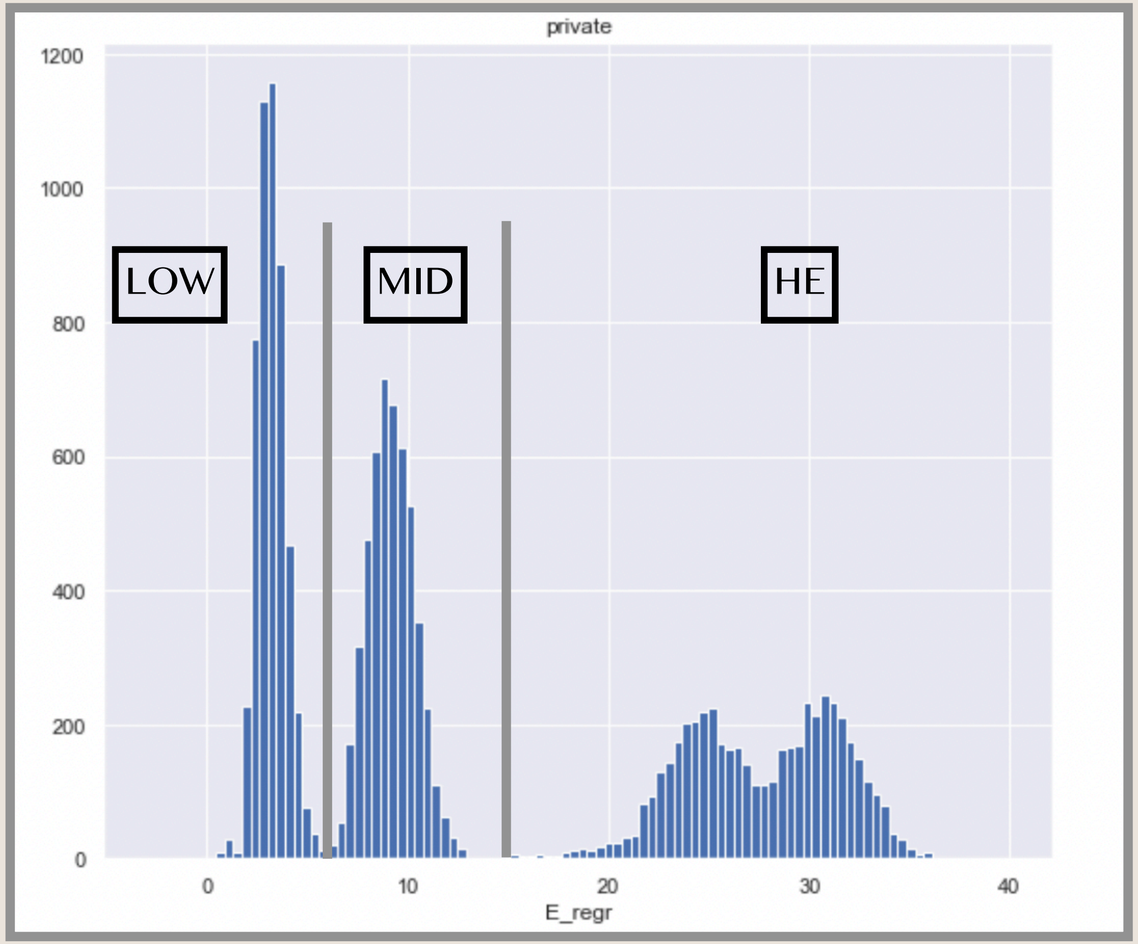




Our solution

Final strategy

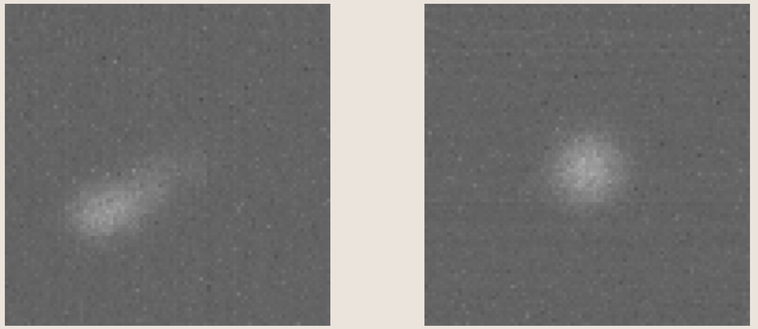
- **Refrain from nicely predicting public and focus entirely on labelling private**
  - Public labelled with the same (not optimal) approach as private
- **Based on `E_regr` define 3 regions of interest:**
  - HE (high energy): `E_regr>=14`
  - MID (middle): `6<=E_regr<14`
  - LOW: `E_regr<6`
- **We expect a priori to have in each of these regions only 2 classes:**
  - HE: ER (20) & NR (30)
  - MID: ER (6) & NR (10)
  - LOW: ER (1) & NR (3)
- **Therefore, problem is split into 3 mutually exclusive binary classification problems**



Our solution

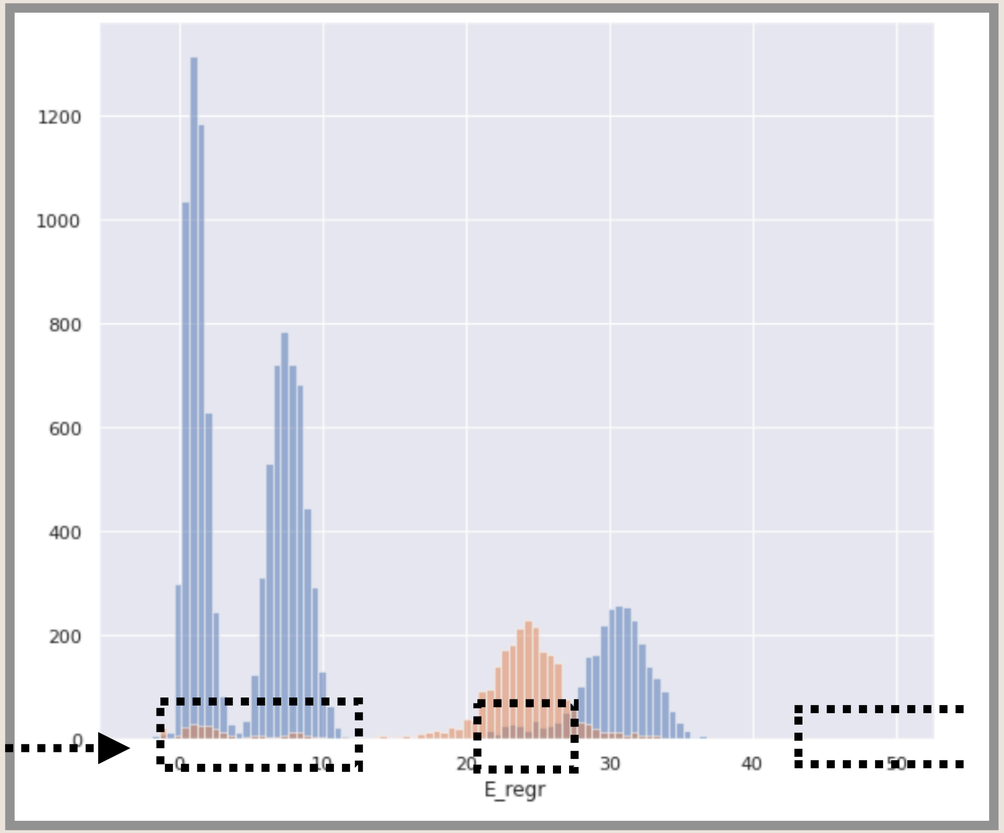
HE region

- **BDT on training dataset**
  - "wiggly classifier" ER(30) vs all: separate "wiggly" electrons from "round" nuclea
  - Features: chi2_pvalue_x, chi2_pvalue_y, abs_dmu_x, abs_dmu_y
  - ROC AUC ≈1 on training sample
- **Can afford using training dataset here**
  - "wiggliness" is ~invariant under ER(30) vs NR(20) <—> ER(20) vs NR(30)
- **Validate on 4 "test domain" samples from training set**
  - Got very confident and correct predictions → looks promising
- **Cut > 0.5 on probas to get deterministic predictions**
- **Remaining misID observed → check & force these events into ER(20) category**
- **Overall approach results in expected ~2500 vs ~2500 split**



ER(20)

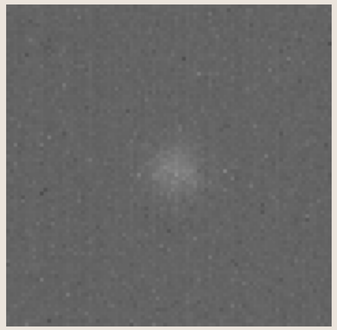
NR(30)



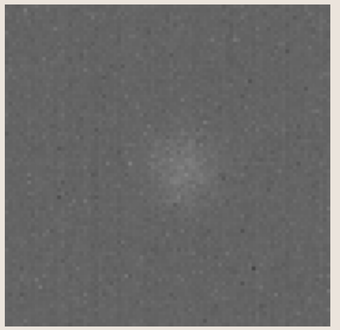
Our solution

MID region

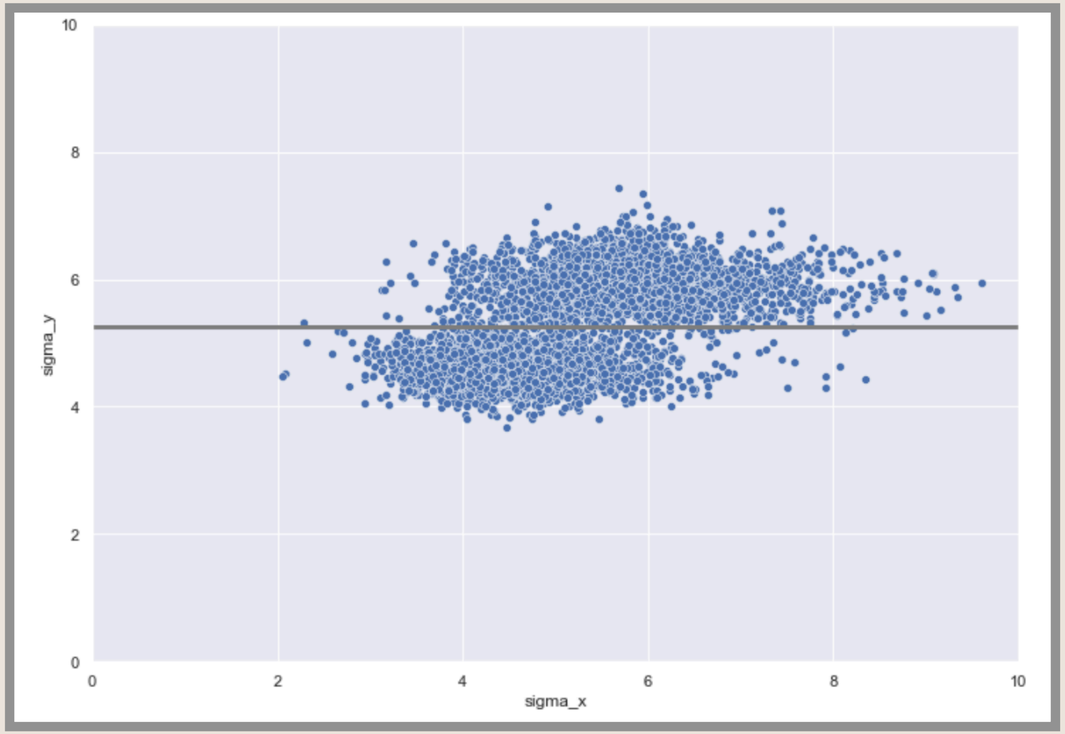
- **Can't use training data as in HE region**
  - Domain shift is significant, ER <—> NR changes distributions dramatically
- **However, turned out that there's a feature where two clearly separated clusters seen**
  - It is σ_y
  - In fact, it makes sense because expect electrons to produce more compact energy deposits
- **Simply went ahead with applying cut $\sigma_y > 5.25$**
- **Validate that it predicts correct labels on 4 private domain samples**
- **Approach also results in expected ~2500 vs ~2500 split**



ER(6)

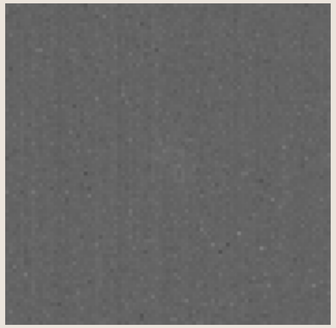


NR(10)

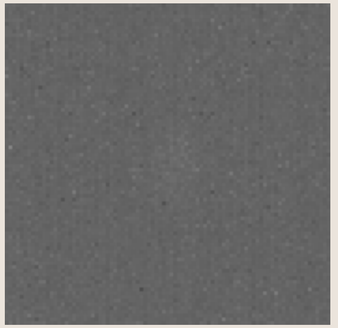


Our solution

LOW region

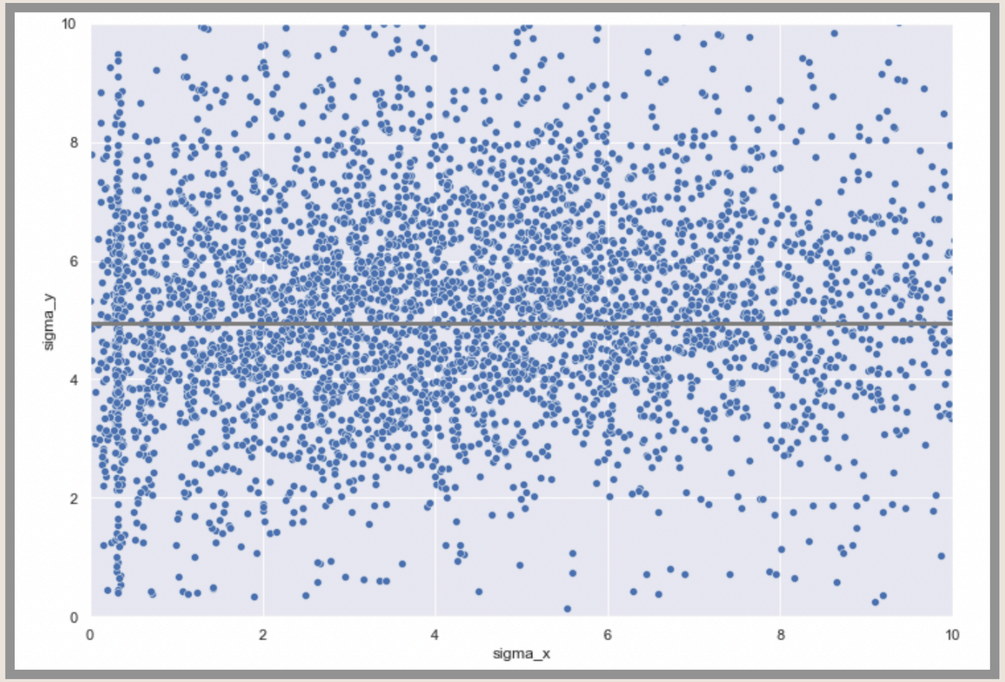


ER(1)



NR(3)

- **The most tough one**
  - Significant overlap of ER(1) and NR(3)
  - Can’t use supervised ML because of domain shift
  - Failed to find "killer feature" as in MID region
- **Decided to pick the most discriminative (by eye) feature and applied cut there**
  - σ_y once again with $\sigma_y > 4.95$
  - 1/5 mispredicted on private domain samples from train
- **Resulted in asymmetric split ~2800 vs ~2200 but that’s OK**
  - Expected since σ_y (NR) > σ_y(ER)
  - However, should be more optimal wrt. balanced option



Our solution

Track 2

- **Approach described above submitted to track 1**
- **But can’t use it for track 2**
  - fitting of images takes O(1) hours → can’t pass 15 min time limit
- **Well, actually we can if we use surrogate features**
  - `sig_count:` subtract average background count from signal region (per projection)
  - `abs_dmu:` finding max count position (per projection)
  - `sigma:` count number of excess bins (per projection)
- **Apply cut on these feature in each region (borders on `sig_count_y` were recalculated) to discriminate btw. corresponding classes**
  - HE: `abs_dmu_y > 2`
  - MID: `sigma_y > 13.5`
  - LOW: `sigma_y > 1.99`
- **Were quite in a rush but managed to make last minute submit before deadline**

Side note

sPlot story

- **Elegant statistical technique to disentangle components from mixture dataset [paper]**
  - Given a priori known (unknown) pdf for each component in discriminating (control) variables*
  - Can derive sWeights for each event (e) w.r.t. for a given species (n):

$$ {}_s\mathcal{P}_n(y_e) = \frac{\sum_j V_{nj} f_j(y_e)}{\sum_j N_j f_j(y_e)} $$

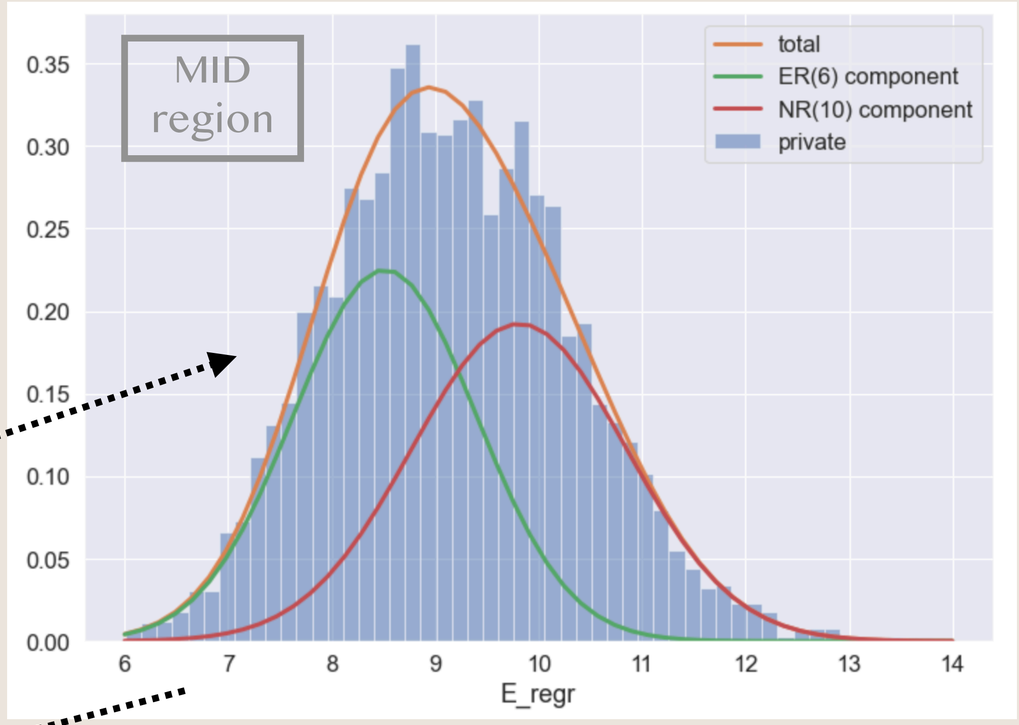
  - By reweighing events can obtain control variable's distributions for each component (as histograms)
  - Basically, "unfold away" (or "background-subtract") all the components in mixture but a given one**
- **Widely used in B-physics community, rarely elsewhere (to my knowledge)**
  - e.g. to study intermediate systems in B-meson decay chains
- **However controversial**
  - Several statistical issues, see more in this dedicated ATLAS+CMS+LHCb workshop
- **Implemented in RooFit and hep_ml library**

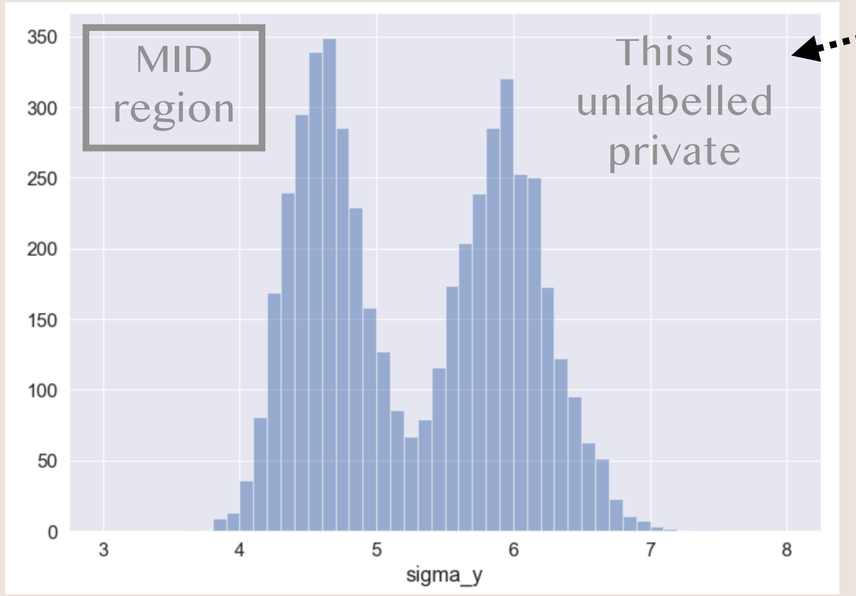
* NB: discriminating and control variables should be uncorrelated!
** See more illustrative example in the backup

Side note

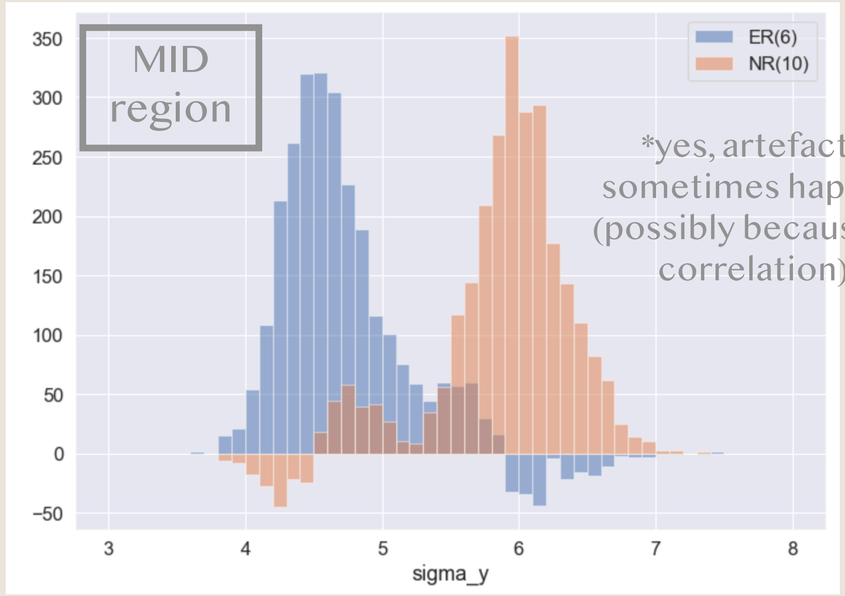
sPlot story

- **Firstly, can apply it to xcheck our assumptions**
  - Also study, how ER/NR is distributed in private
  - e.g. take MID region, 2 components: ER(6) vs NR(10)
  - Discriminative variable: `E_regr`, pdfs: Gaussian
  - Control variable: whichever you like, say σ_y
  - Can get a LOT of insights into private dataset





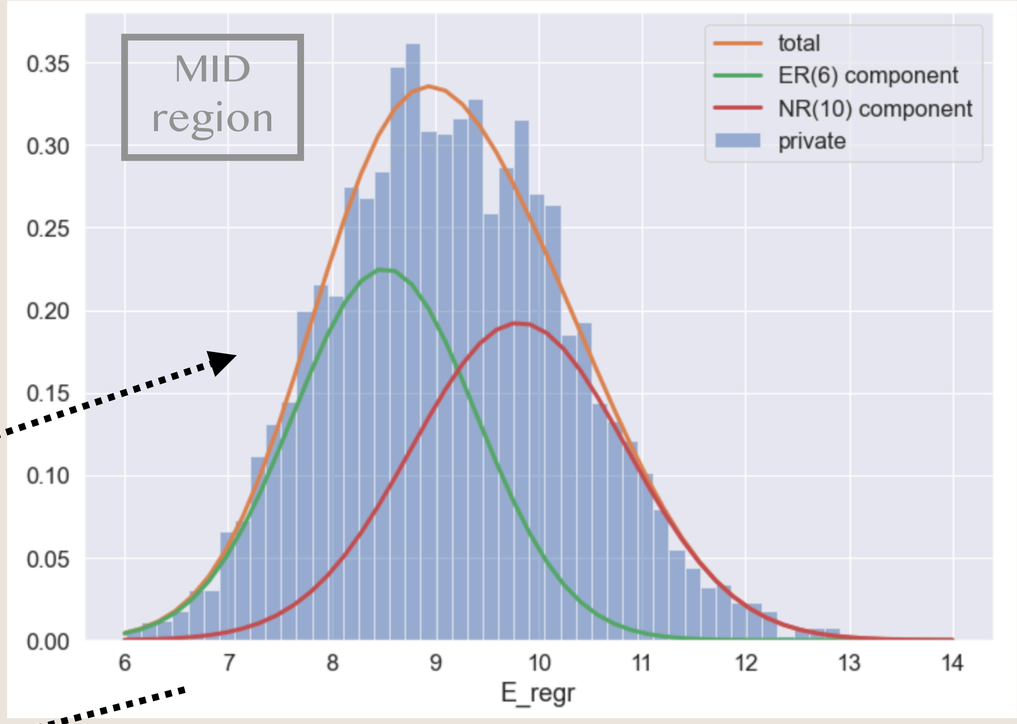
And this is **sPlot-"labelled"** private

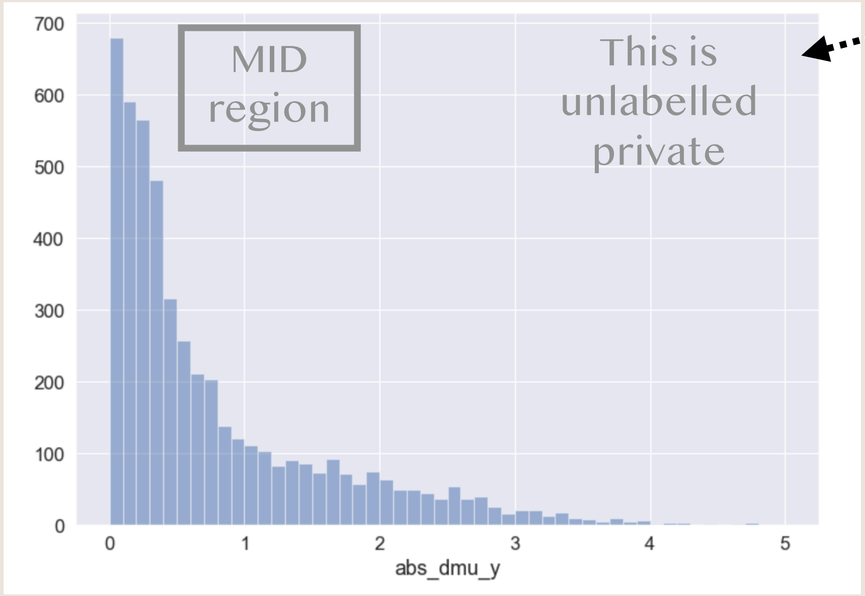


Side note

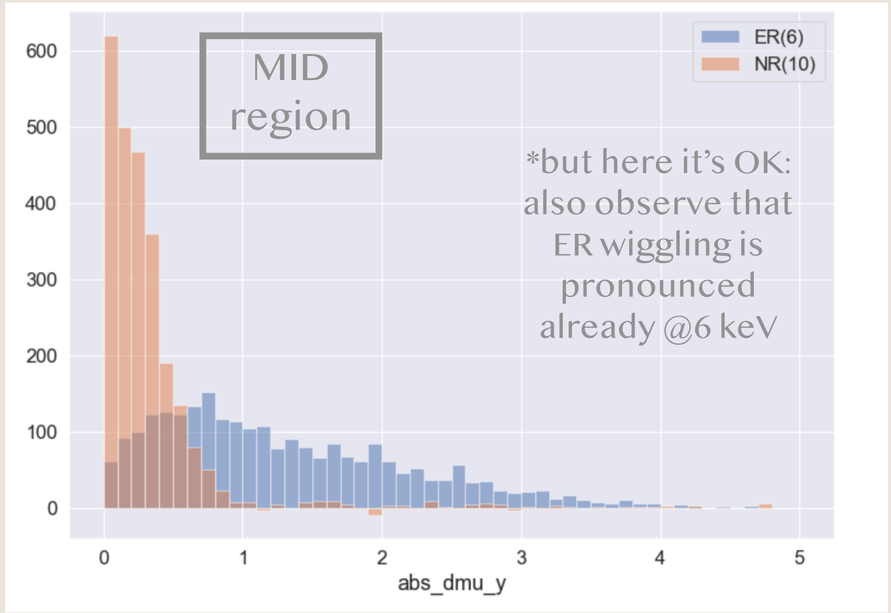
sPlot story

- **Firstly, can apply it to xcheck our assumptions**
  - Also study, how ER/NR is distributed in private
  - e.g. take MID region, 2 components: ER(6) vs NR(10)
  - Discriminative variable: `E_regr`, pdfs: Gaussian
  - Control variable: whichever you like, say σ_y
  - Can get a LOT of insights into private dataset





And this is **sPlot-"labelled"** private



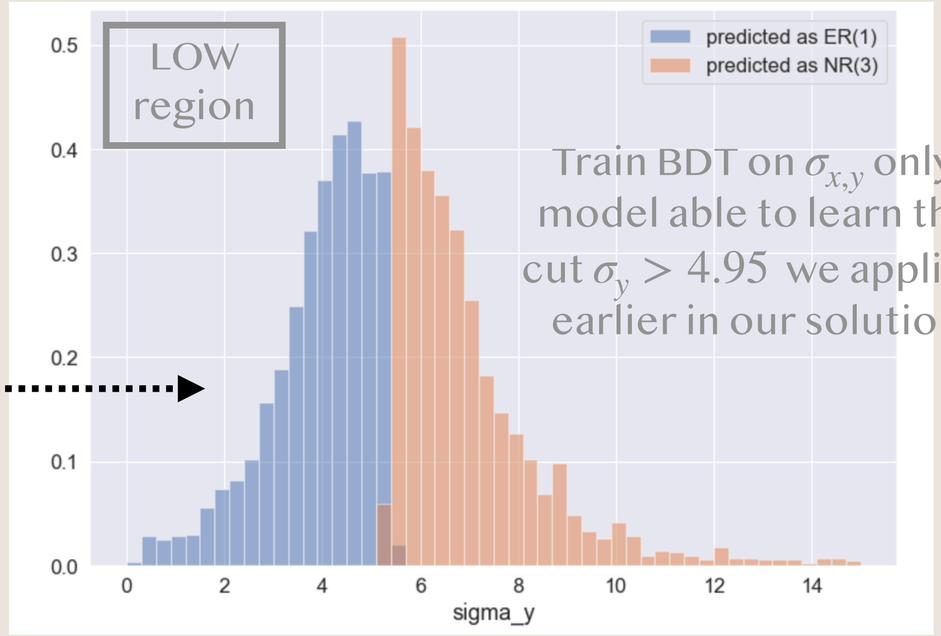
Side note

sPlot story

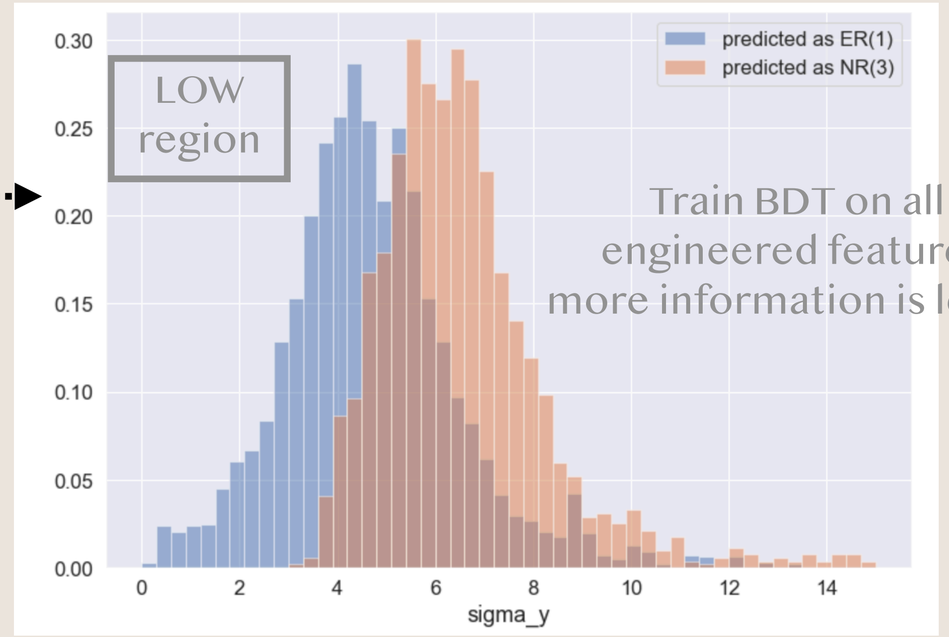
$$ {}_s\mathscr{P}_n(y_e) = \frac{\sum_j V_{nj} f_j(y_e)}{\sum_j N_j f_j(y_e)} $$

$${}_s\mathscr{P}_{sig}(y_e), \hat{y} = 1 \qquad {}_s\mathscr{P}_{bkgr}(y_e), \hat{y} = 0$$

- **But what if we can use sWeighted distributions to train model?**
  - Yes, we can! [paper]
  - Effectively, sPlot disentangles components and therefore "labels" them*
  - Can train model on previously unlabelled data by introducing sWeights into loss function
  - Although in our problem hardly possible to validate the method**, checks hint that it works
- **As further prospects for curious listener, with sPlot one can in principle train models directly on data (not simulation)**



Train BDT on $\sigma_{x,y}$ only: model able to learn the cut $\sigma_y > 4.95$ we applied earlier in our solution



Train BDT on all engineered features: more information is learnt

* happens once we sum up weights of a given component and plot them as a histogram
** that is why to avoid risks we didn't submit this solution

Results & summary

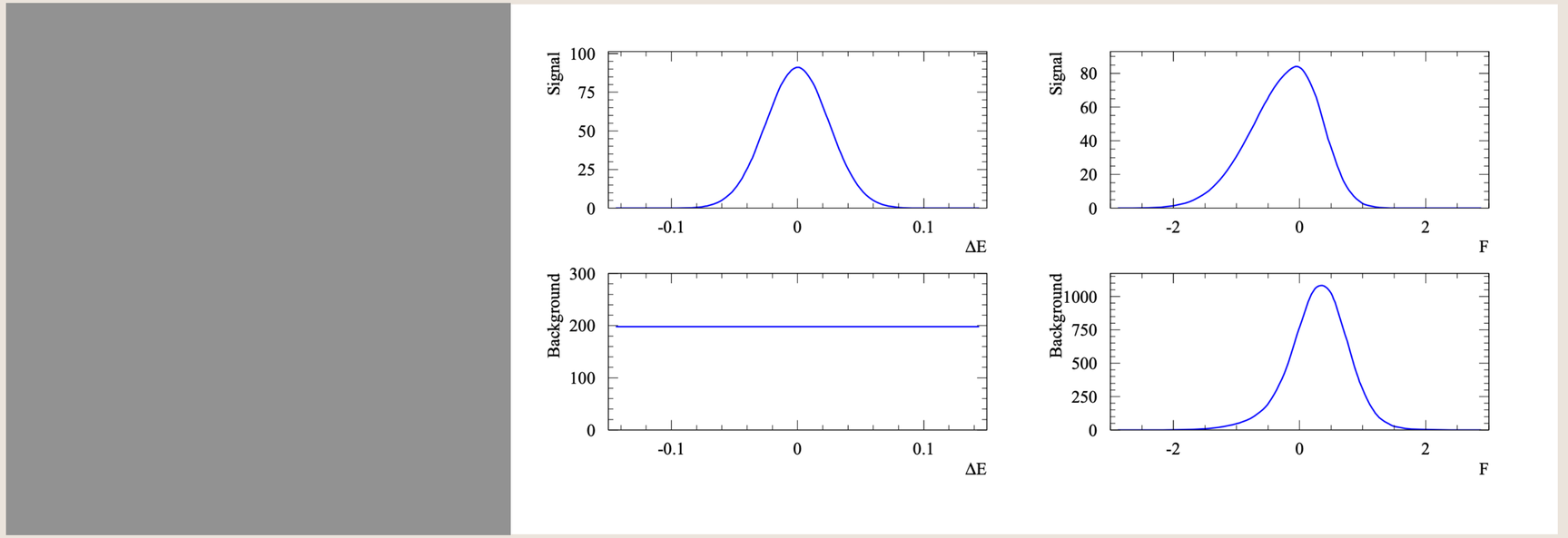
- **Track 1** ($-\infty < \text{score} <= 1000$)
  - Public: AUC= 0.830, MAE = 2.368, total = -1537.9
  - Private: AUC = 0.878, MAE = 0.283, **total = 595.49**
- **Track 2** ($-\infty < \text{score} <= 1000$)
  - Public: AUC= 0.687, MAE = 2.982, total = -2295.3
  - Private: AUC = 0.596, MAE = 3.055, total = -2459.04
- **5th place (track 1) & 11th place (overall) / ~200 teams**
- **The code:** https://github.com/depot-hep/idao-2021

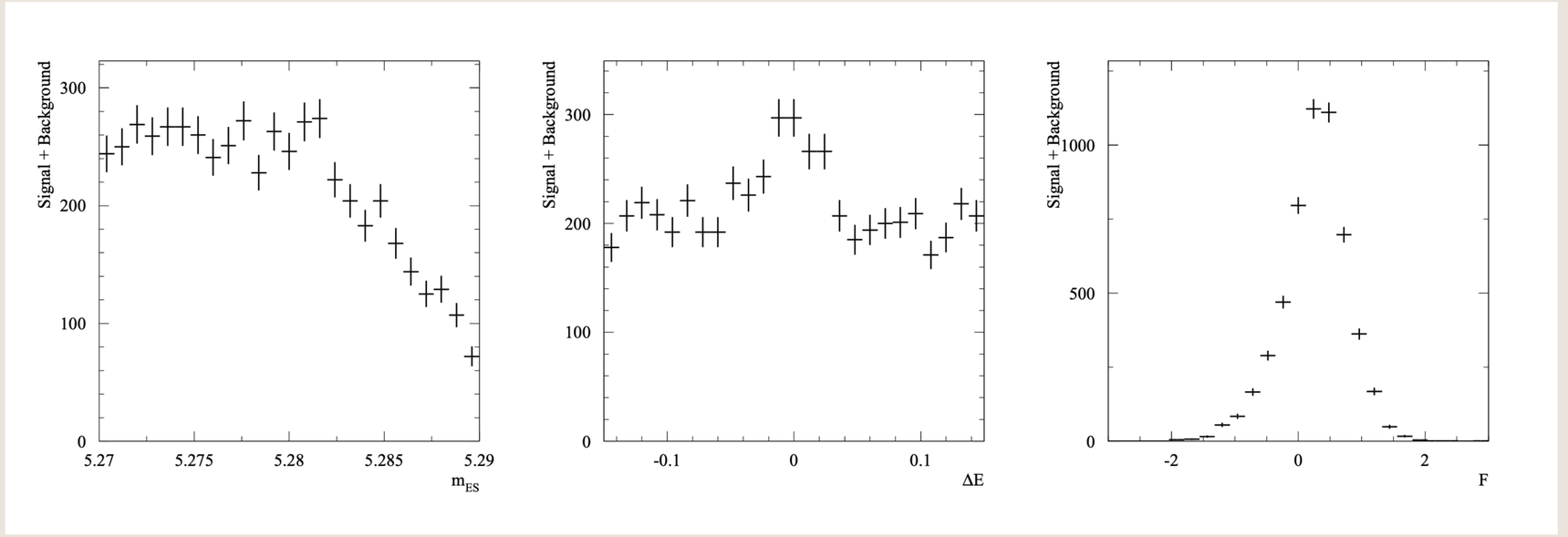
	Track 1	Country
1	It doesn't matter	France
2	Veni Vidi Vici	Belarus, Russia
3	SquattingSlavs	Germany, United States
4	DS19	Russia
5	Baobab	Germany, Russia
6	tsubasa	Azerbaijan
7	Shizika	Russia
8	Super Mario Bros	Bangladesh
9	BegInnors	Russia
10	Data Pro	Australia, Hong Kong
11	Sabhya log	India
12	oski	Russia
13	ai.max.roar	Russia
14	Анти материалисты	Russia
15	data o plomo	France

	Track 2	Country
1	Optimization 4 KO	Russia
2	random team	Switzerland
3	!	Russia
4	mn team 2	Korea, Mongolia
5	Alpha Analysts	Malaysia, Russia, Egypt
6	Made As Described Earlier	Russia
7	data siens	Russia
8	Magic City	Russia
9	CHAD DATA SCIENTISTS	Russia
10	QuantumCurious	India
11	Tonatiuh	Russia
12	misclass_classifier	India
13	NotExperts	Russia
14	Getsuga Tenshou	Russia
15	White Material	Russia

BACKUP

sPlot

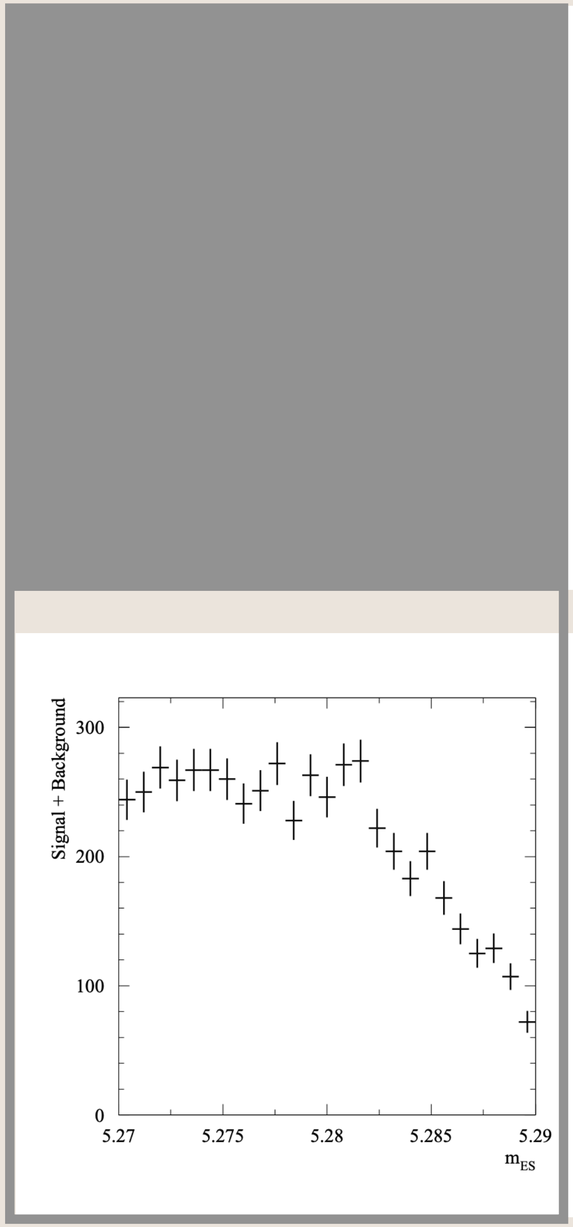

Signal
100
75
50
25
0
-0.1
0
0.1
ΔE
Signal
80
60
40
20
0
-2
0
2
F
Background
300
200
100
0
-0.1
0
0.1
ΔE
Background
1000
750
500
250
0
-2
0
2
F

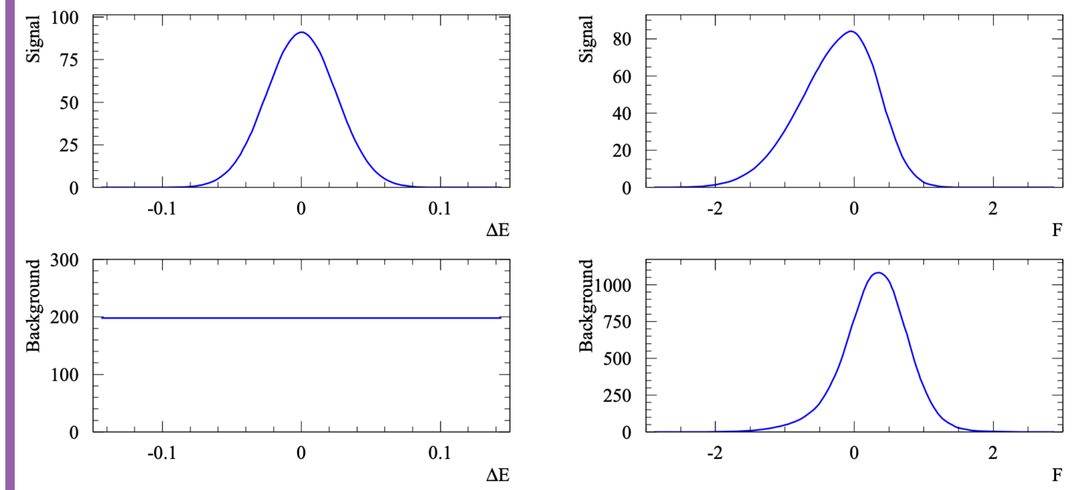

Signal + Background
300
200
100
0
5.27
5.275
5.28
5.285
5.29
m_ES
Signal + Background
300
200
100
0
-0.1
0
0.1
ΔE
Signal + Background
1000
500
0
-2
0
2
F

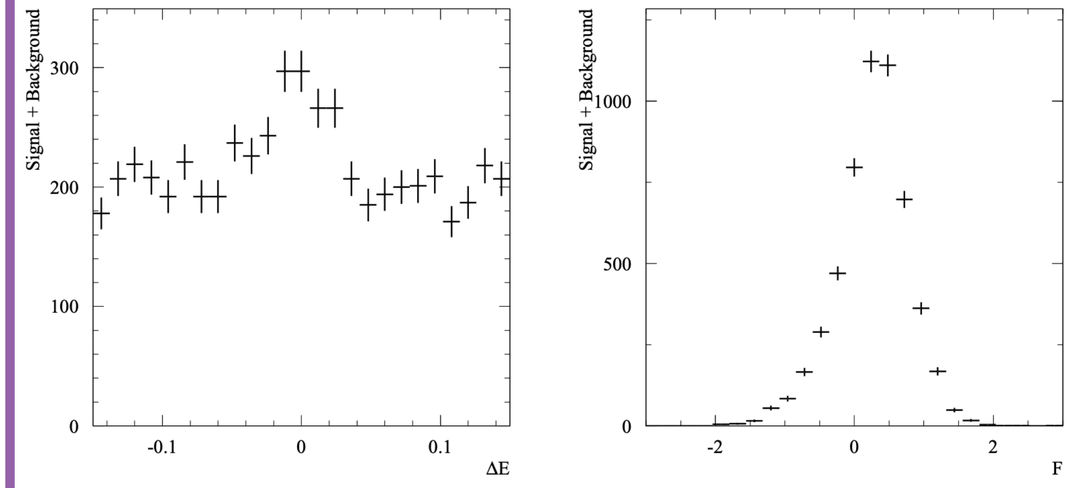
sPlot

Control variable

Discriminating variables







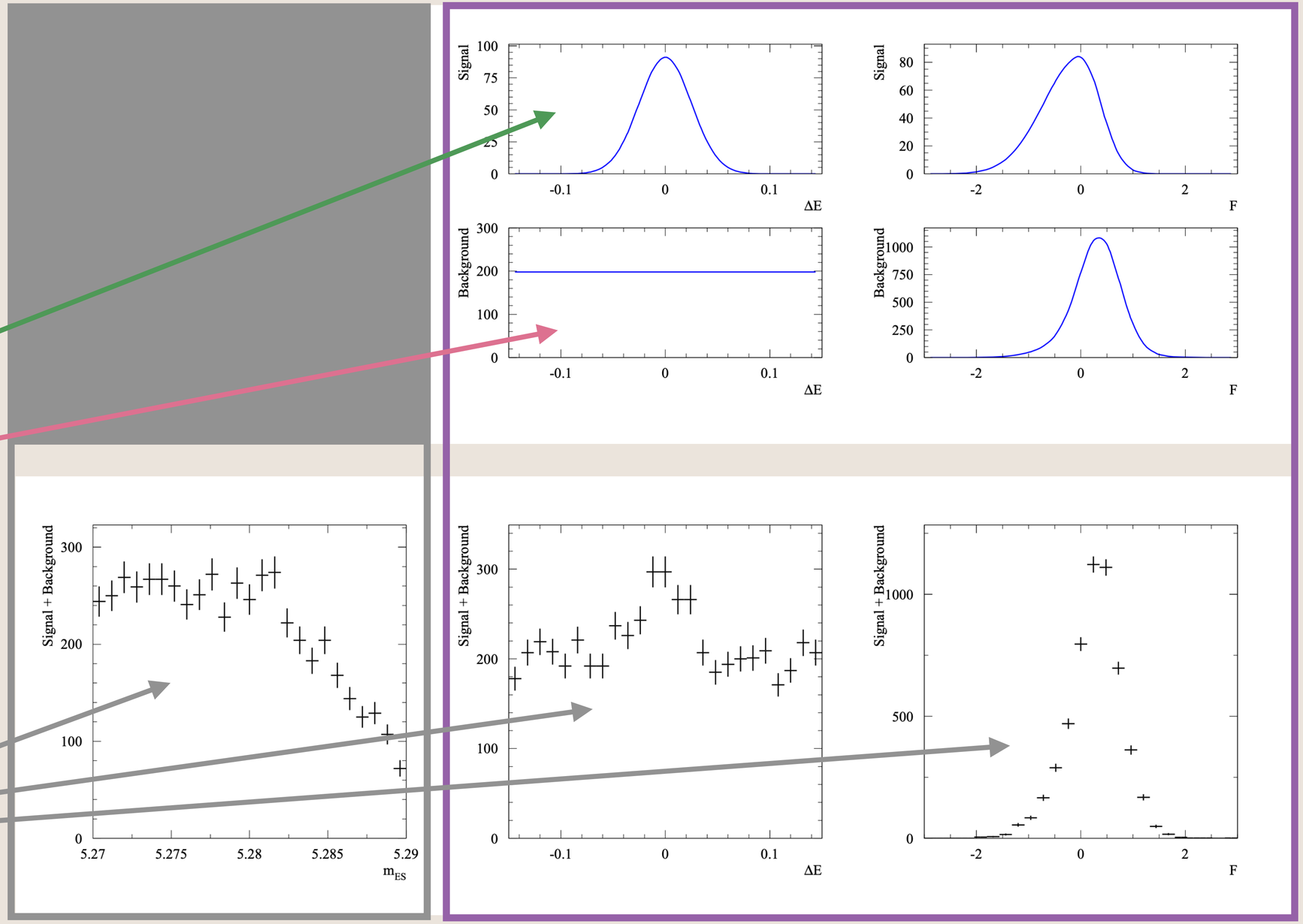
sPlot

Control variable

Discriminating variables

2 components
(signal, background)

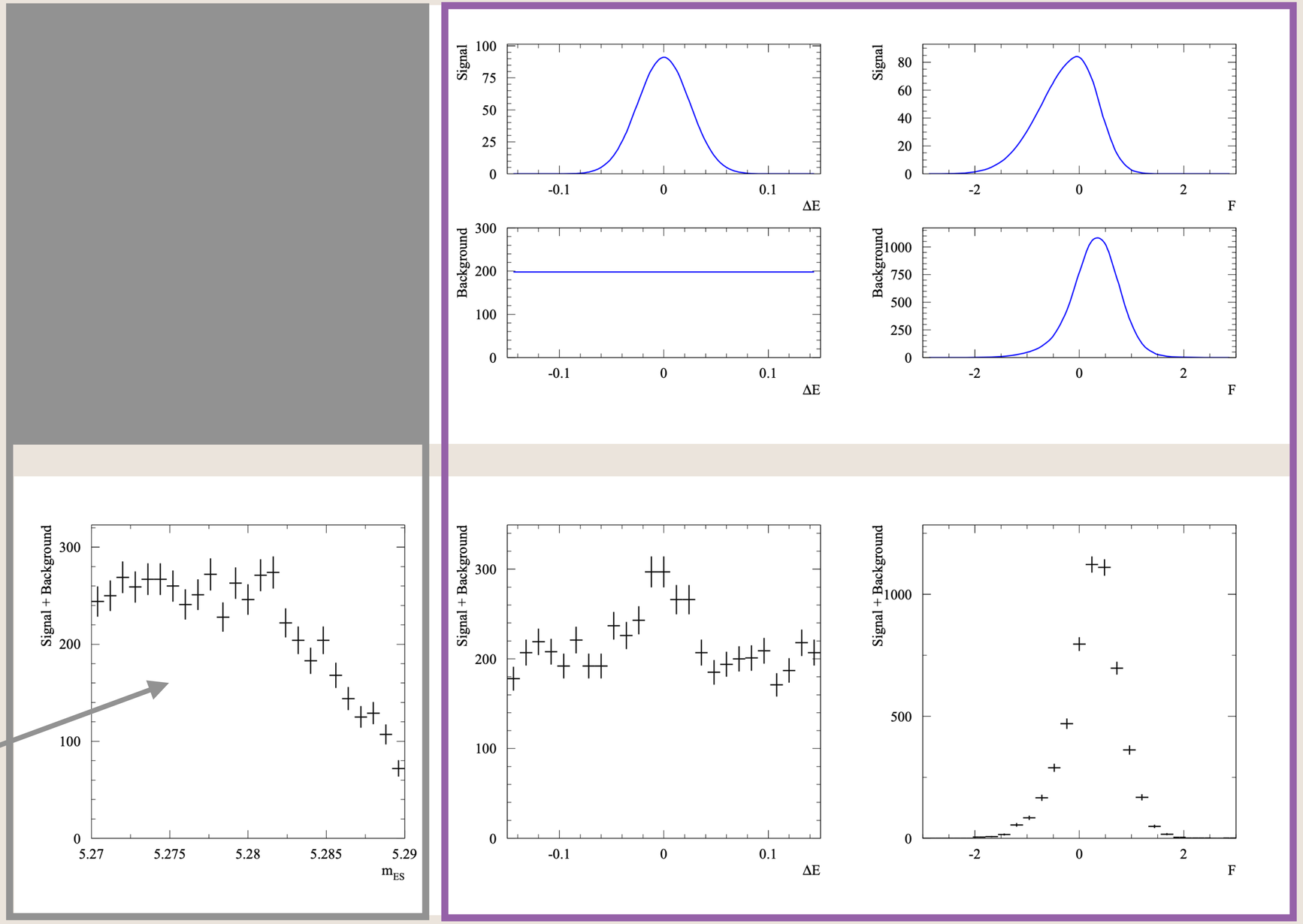
Observe their mixture



sPlot

Control variable

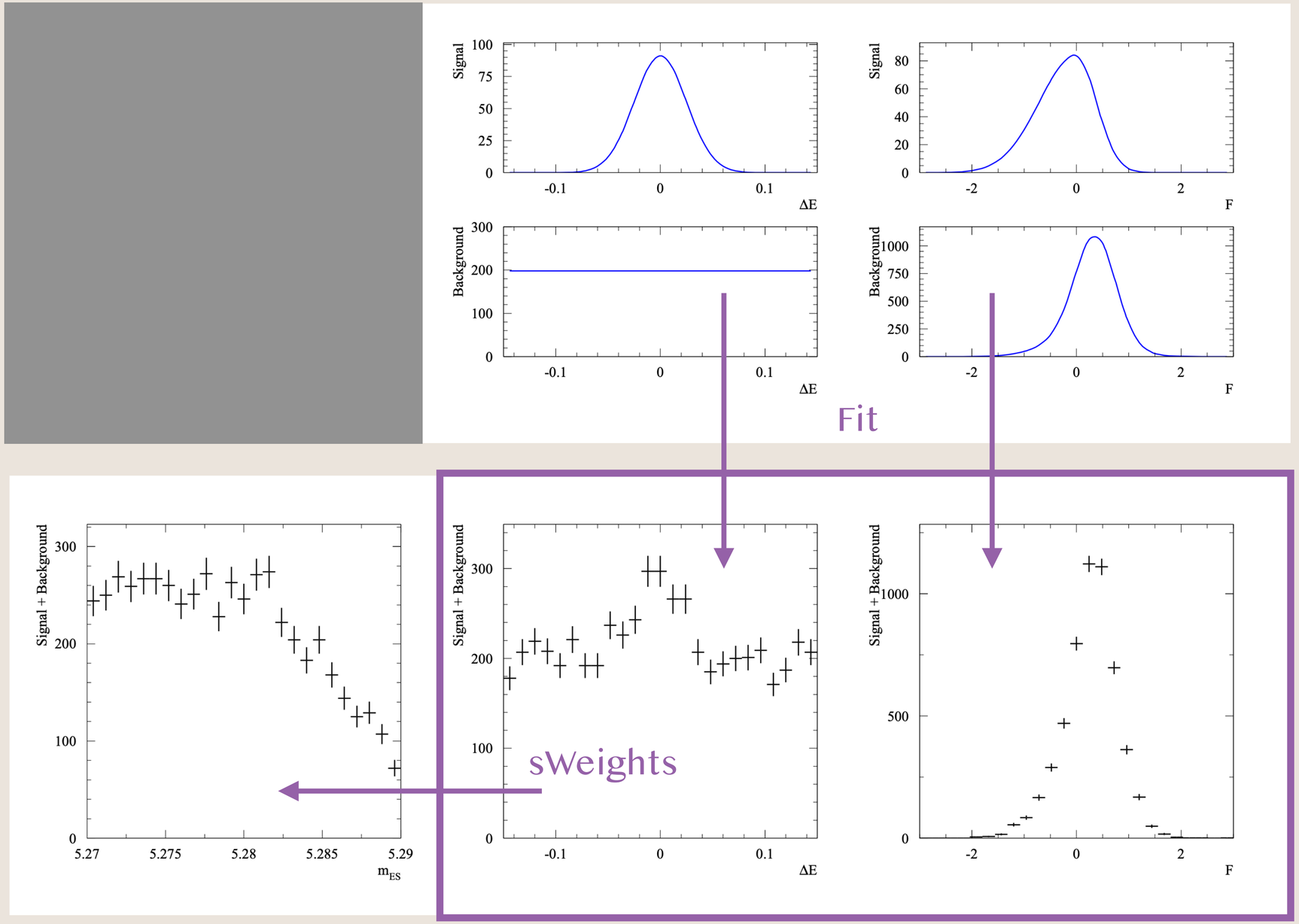
Discriminating variables



Can we split this histogram into two components?

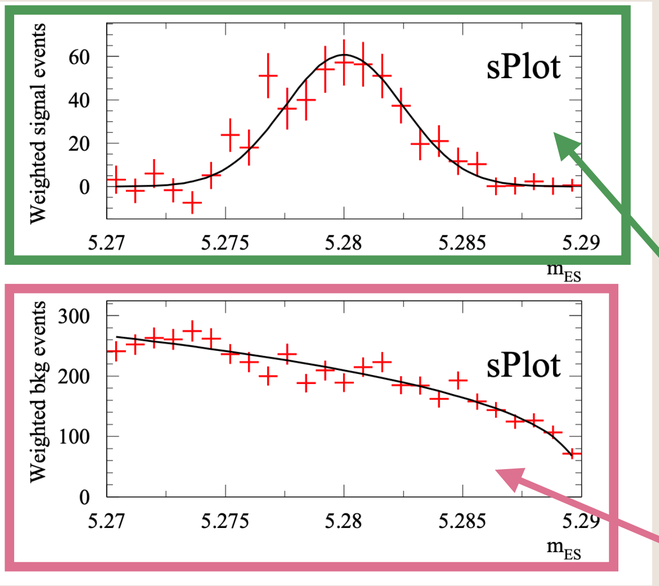
sPlot

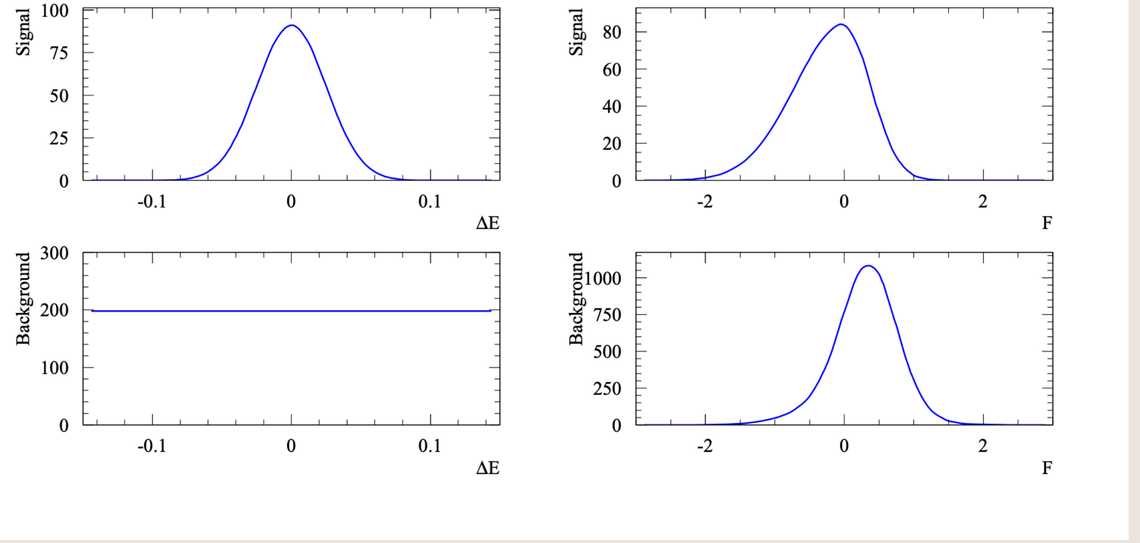
$$ {}_s\mathscr{P}_n(y_e) = \frac{\sum_j V_{nj} f_j(y_e)}{\sum_j N_j f_j(y_e)} $$



sPlot

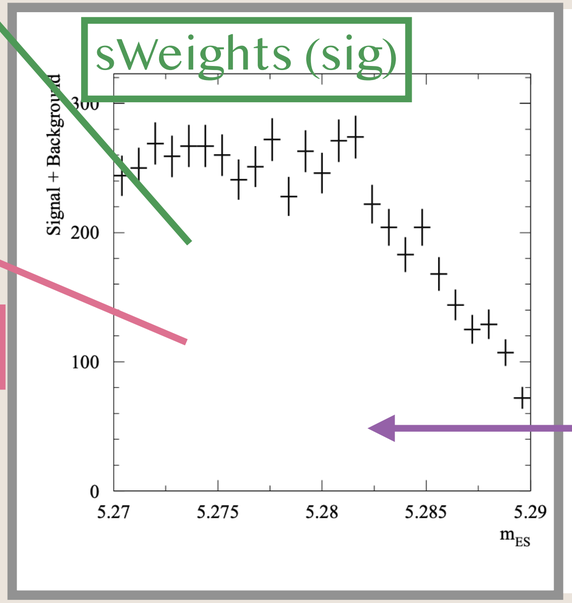
Ignore black fit line
for the moment

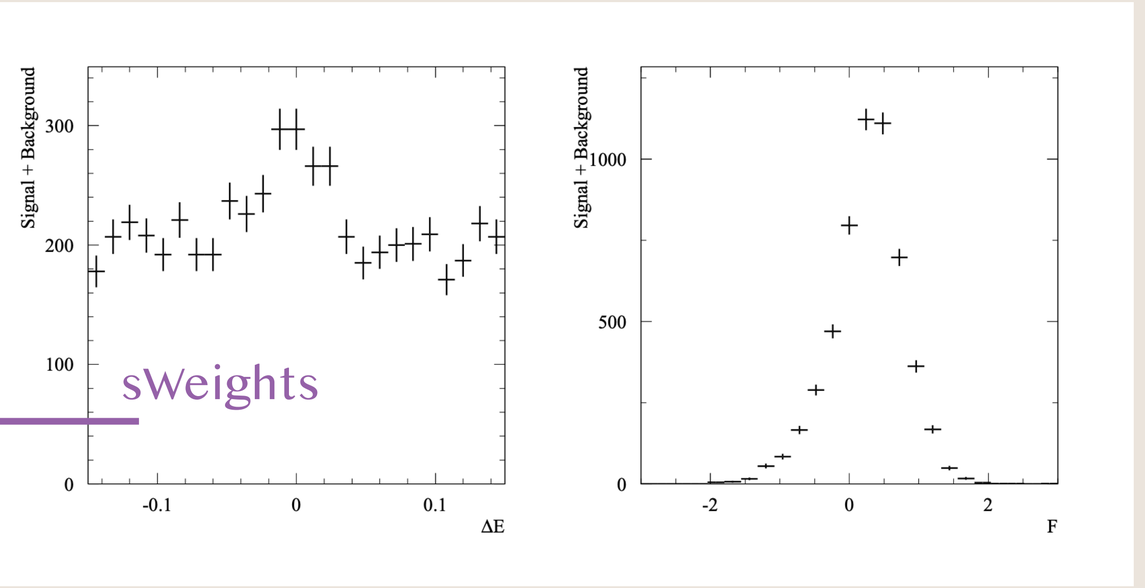




sWeights (sig)

sWeights (bkgr)

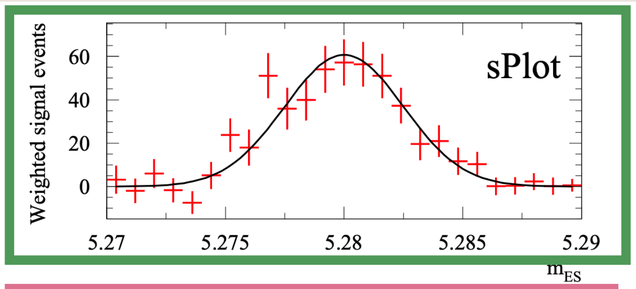


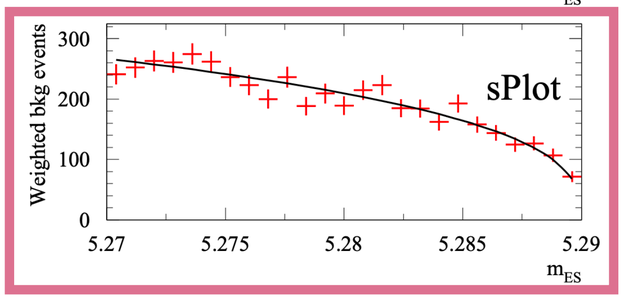


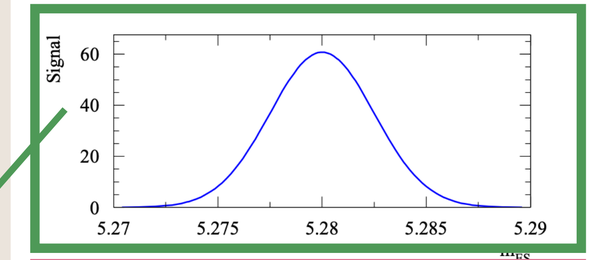
$$ {}_s\mathscr{P}_n(y_e) = \frac{\sum_j V_{nj} f_j(y_e)}{\sum_j N_j f_j(y_e)} $$

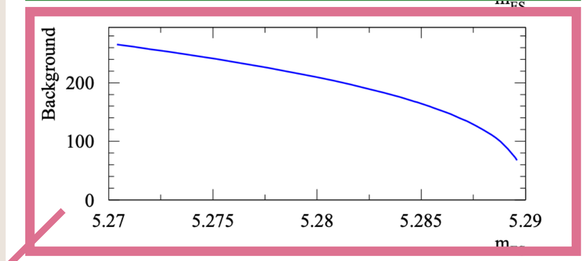
sPlot

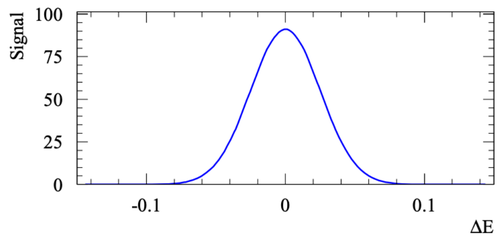
Check with known shapes
(black fit line)

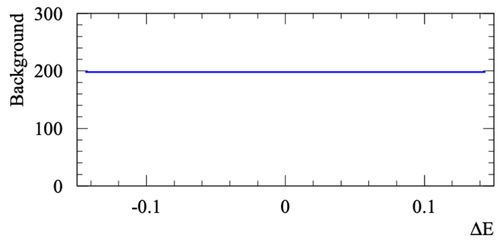


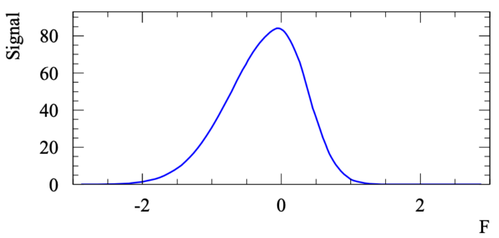


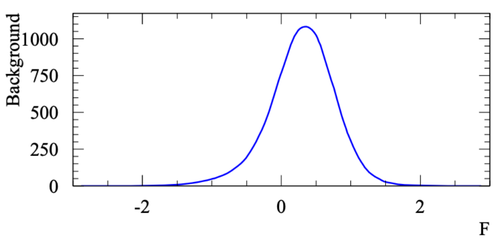


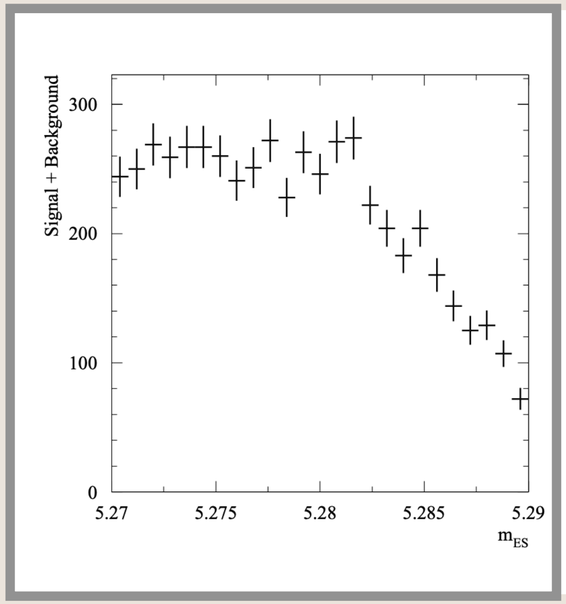


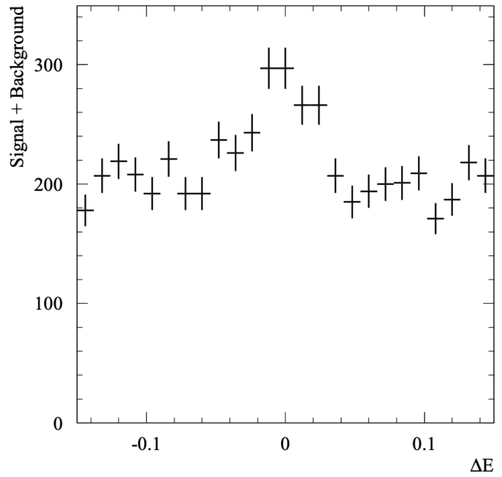


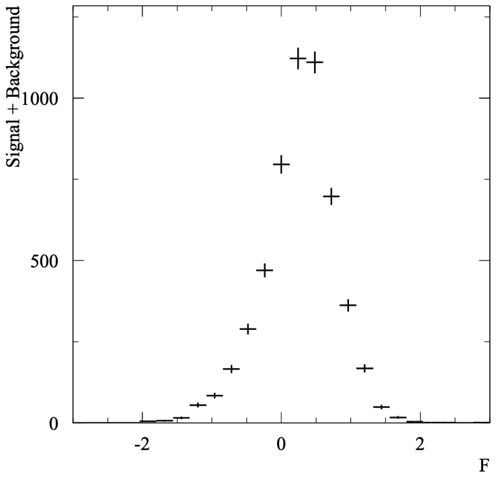












$$ {}_s\mathscr{P}_n(y_e) = \frac{\sum_j V_{nj} f_j(y_e)}{\sum_j N_j f_j(y_e)} $$