# ND-GAr sim/reco updates

**ND-GAr Weekly Meeting**
**01.12.2023**

Francisco Martínez López
f.martinezlopez@qmul.ac.uk

# Introduction

- As discussed in the Phase II ND workshop, one of our priorities is to quantify the physics impact of ND-GAr, in order to deliver a physics-driven design.
- One of the key ingredients is to implement ND-GAr in the LBL analysis, putting together a pipeline from the generators (GENIE, NuWro, …) to the fitters (Mach3 and CAFAna).
  - This will make it easier to test and compare the impact of changes in the ND-GAr design.
- The first samples we are thinking of providing to the LBL analysis will be divided in pion multiplicity.
  - We should be able to select topologies with $0\pi$, $1\pi$ and $\geq 1\pi$.
- To that end, we need a reliable PID able to identify pions with a high purity and across a broad energy range.
- This is a summary of some reconstruction topics I've been focused on in order to improve the reconstructed pion multiplicity estimation.

# Outline

I. Introduction

II. TPC charge saturation

III. Track breakpoints and charged pion decays

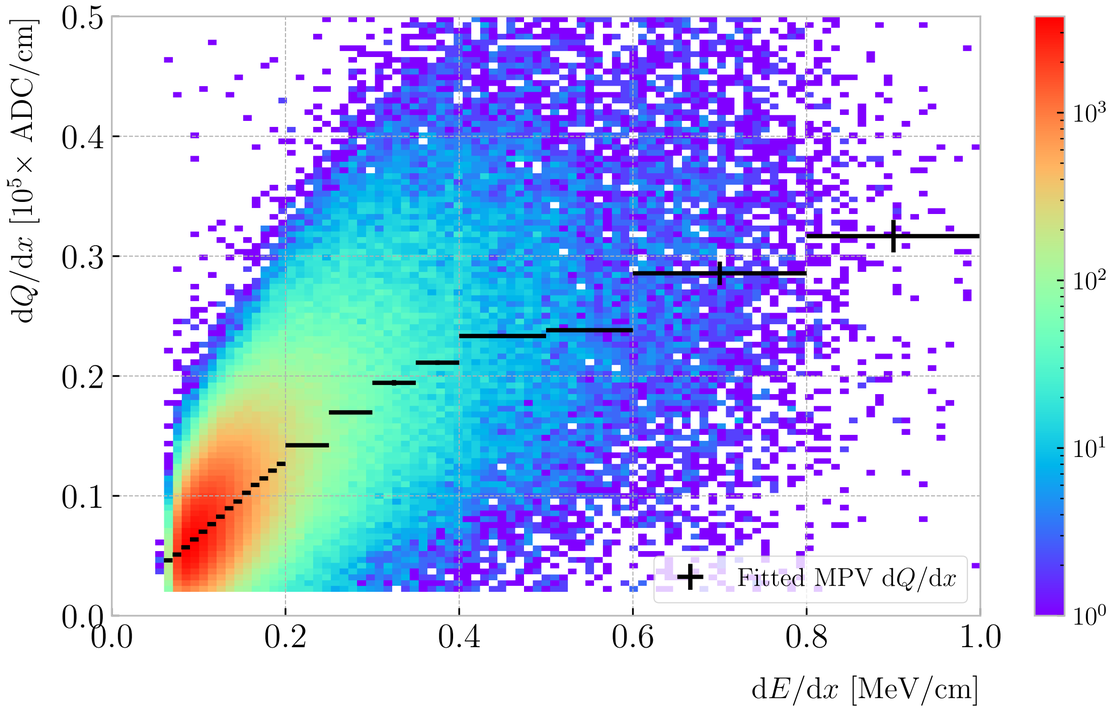
IV. ECal clustering and neutral pion reconstruction

V. Track-ECAL associations

VI. Conclusions

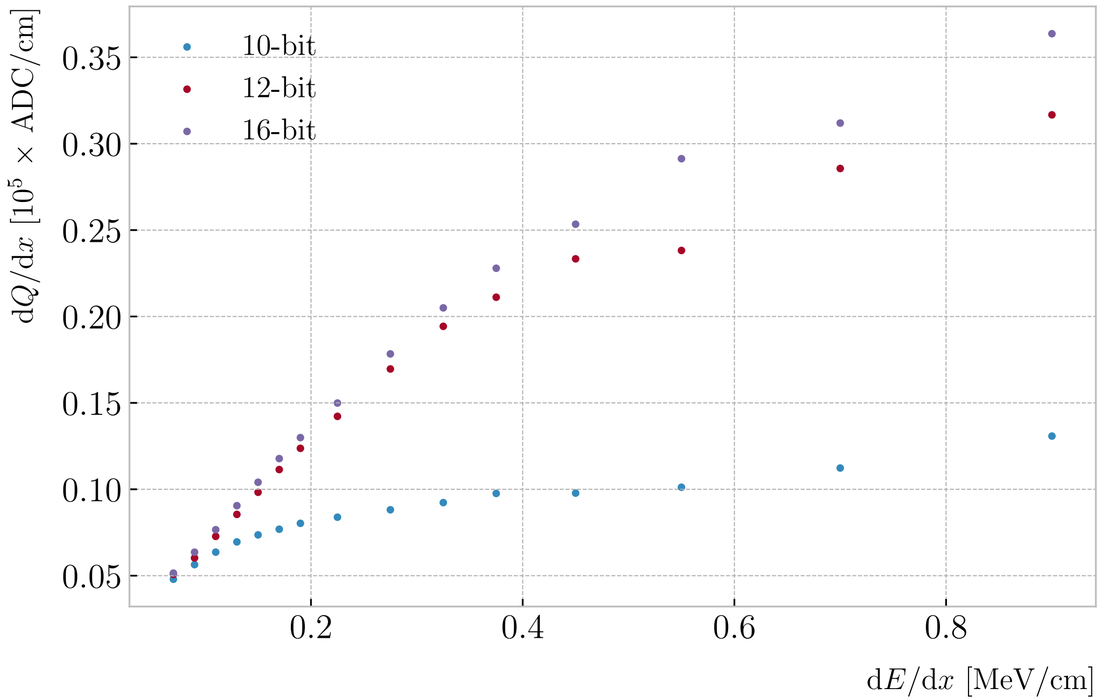
# dQ/dx vs dE/dx

- Currently, GArSoft does not simulate electron-ion recombination in the GAr, $N_e \propto E$.
  - The only effects that modify the amount of electrons that reach the readout planes are transverse diffusion and electron lifetime.
  - Once the electrons reach the readout chambers, the pad response functions are applied, together with the electrons-to-ADC conversion and the ADC saturation limit.
- When we compare the energy and charge deposited per unit length, we can see that the relation between the two appears to be non-linear.



# Charge saturation

- Effects like diffusion and attenuation could be accounted for by applying a $\mathrm{d}Q/\mathrm{d}x$ uniformity calibration.
  - However, those two alone do not completely explain the previous non-linearity.
- ADC saturation can explain (at least part of) this behaviour.
- I tried using different values for the ADC limit to see the impact on the charge as a function of the energy.
  - 16-bit is the maximum, as the ADC are stored in a `std::vector<short>`.
- Should we try to correct for this effect in the reconstruction?



# Charged pion decays

- In some cases, the pattern recognition algorithm of GArSoft is unable to identify discontinuities in possible track candidates, e.g. particle decays.

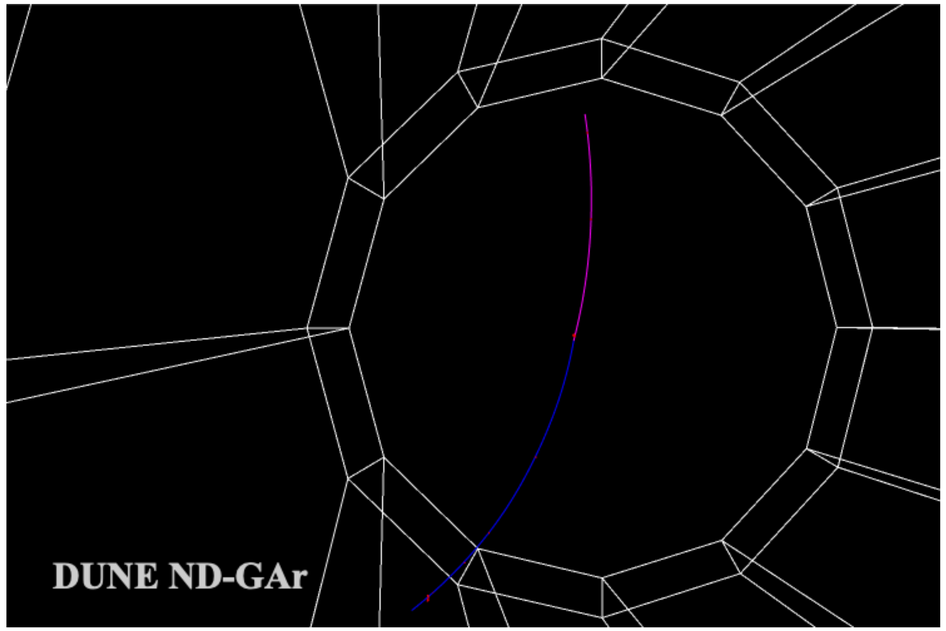
- Based on NOMAD's approach, I tried to construct different test statistics to identify pion decays in the TPC for which GArSoft only produced a reconstructed track for the pion+muon.

- The simplest test we can think about is computing the $\chi^2$ of the mismatch between the parameters in the forward and the backward fits:

$$\chi_k^{2\ (FB)} = (\hat{\mathrm{x}}_k^B - \hat{\mathrm{x}}_k^F)^T [V^{(\hat{\mathrm{x}}_k, B)} + V^{(\hat{\mathrm{x}}_k, F)}]^{-1} (\hat{\mathrm{x}}_k^B - \hat{\mathrm{x}}_k^F)$$

- Another possibility is using a parametrisation of the Kalman filter state vector that allows some fit parameters to be discontinuous at certain points.

$$\alpha = \left(y, \quad z, \quad 1/R_F, \quad 1/R_B, \quad \phi_F, \quad \phi_B, \quad \tan\lambda_F, \quad \tan\lambda_B\right)^T$$



# Breakpoint variables

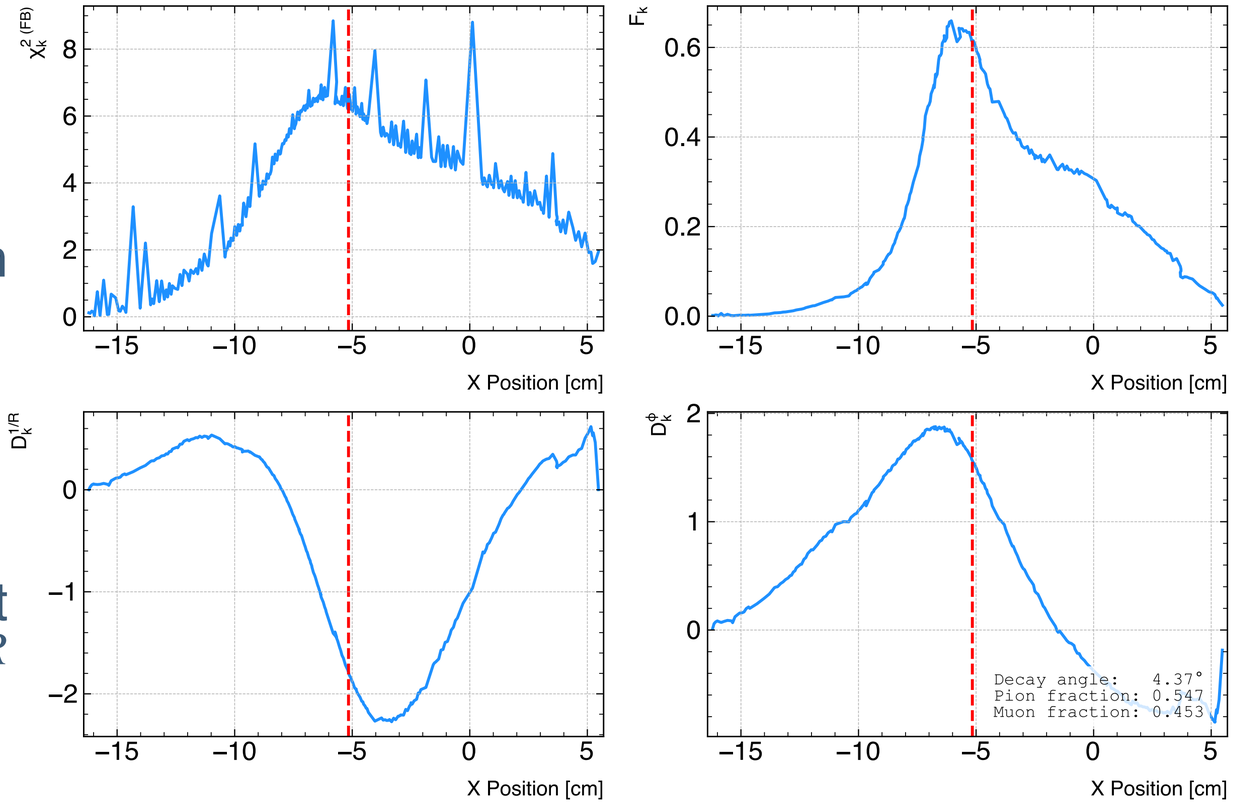
- Because we already have the results from the standard Kalman filter at each point and the model relating the default and the modified state vectors is linear, we can get the new estimates at each point as the values that minimise the new $\chi^2$ of the forward-backward mismatch:

$$\chi^{2\ (FB)}_{\alpha,k} = (\hat{\mathrm{x}}^F_k - H^F\alpha)^T\left[V^{(\hat{\mathrm{x}}_k,F)}\right]^{-1}(\hat{\mathrm{x}}^F_k - H^F\alpha)$$
$$+(\hat{\mathrm{x}}^B_k - H^B\alpha)^T\left[V^{(\hat{\mathrm{x}}_k,B)}\right]^{-1}(\hat{\mathrm{x}}^B_k - H^B\alpha)$$

- From these new fit estimates we can compute the $F$ statistic as:

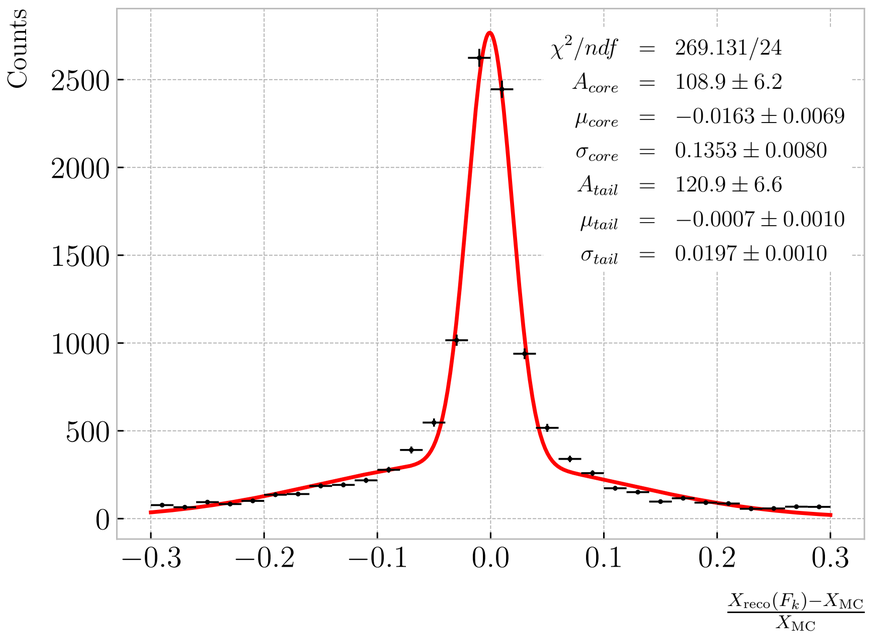
$$F_k = \left(\frac{\chi^2_{\mathrm{x},k} - \chi^2_{\alpha,k}}{8-5}\right)/\left(\frac{\chi^2_{\alpha,k}}{N-8}\right)$$

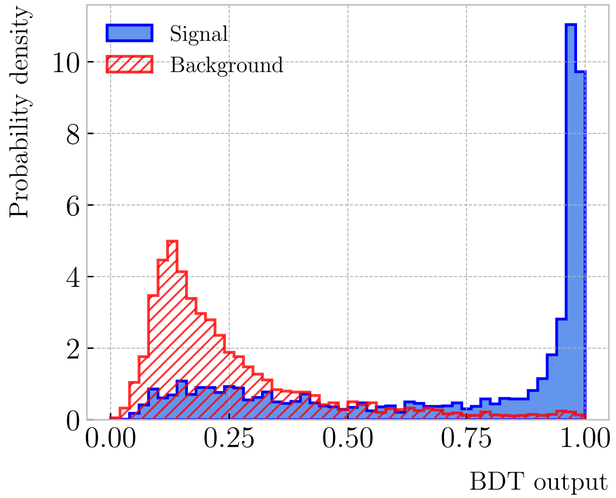
- We can also get the signed difference at each point for the duplicated variables, $D^{1/R}_k$ and $D^{\phi}_k$ in particular.

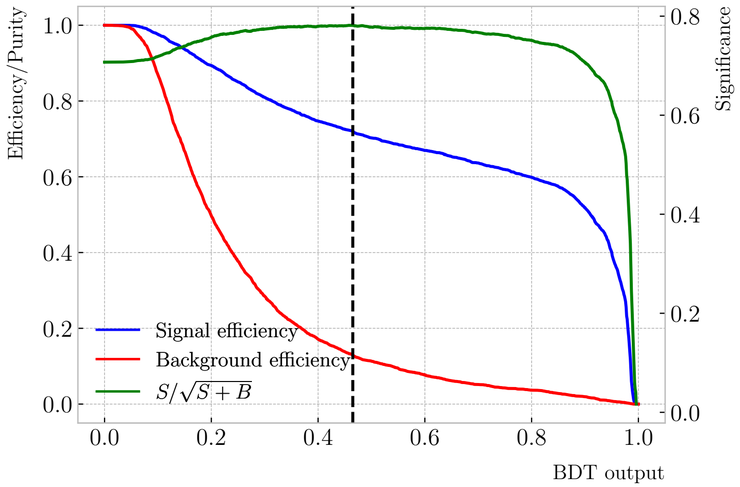


# Breakpoint performance

- The location of the $F_k$ maximum provides a good estimate of the position of the decay.
  - Using a double Gaussian fit, we find a resolution of $7.45\,\%$.
- Other variables, like $\chi^{2\ (FB)}_k$, can also be used but the resolution is significantly worse.




- A BDT could provide good separation between pion decay events and non-decaying pions.
  - The most important variable turned out to be $D_k^{1/R\ (min)}$.

# ECal clustering

- Thinking about how we can achieve a good neutral pion reconstruction, probably one of the reconstruction items that will affect this the most is the clustering of ECal hits.
- Current ECal clustering module consists of a NN algorithm that treats all hits in the same way, independent of their origin.
  - The scintillator layers of our ECal are a mix of two different technologies, the inner layers are made out of tiles whereas the rest are cross-strips.
  - We could try a clustering that behaves differently depending on where the hits originated.
- Grabbing some inspiration from T2K's ND280 DsECal reconstruction, I tried to put together a clustering module that first builds clusters for the different ECal views, and then tries to match them together to form the final clusters.
  - A first working version of it is finished and tested on different samples, but there are still lots of aspects to be refined.

# Clustering algorithm

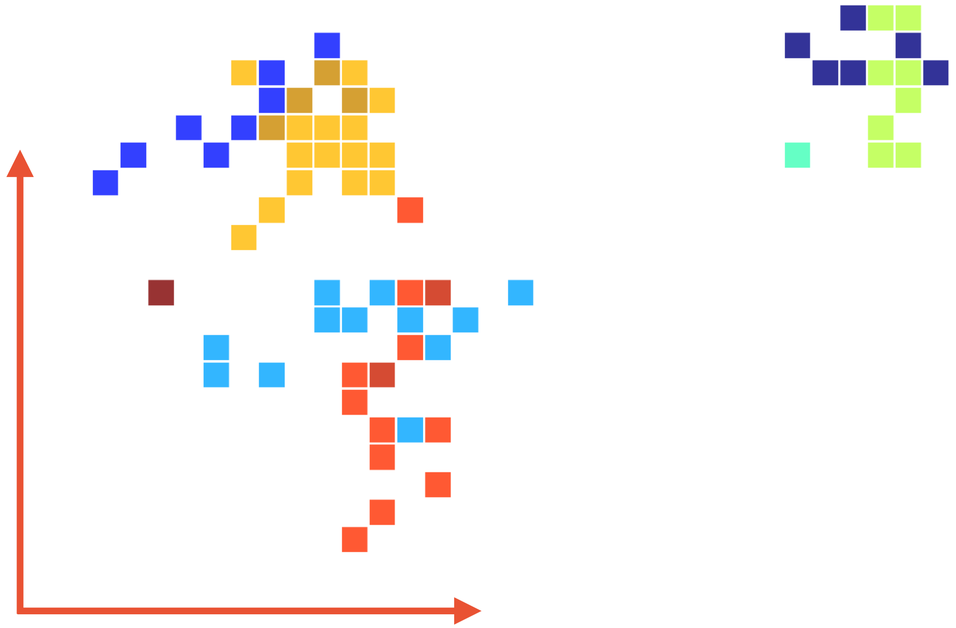
- Working on a module-by-module basis, the algorithm first separates the hits depending on the kind of layer they come from: `Tile`, `StripX` or `StripY`.

- It first performs a NN clustering for the 3 sets of hits separately, applying then a recursive re-clustering for each collection of strip clusters based on a PCA method.

- The clusters in each strip view are combined and then we try to merge them with tile clusters that point in a similar direction.

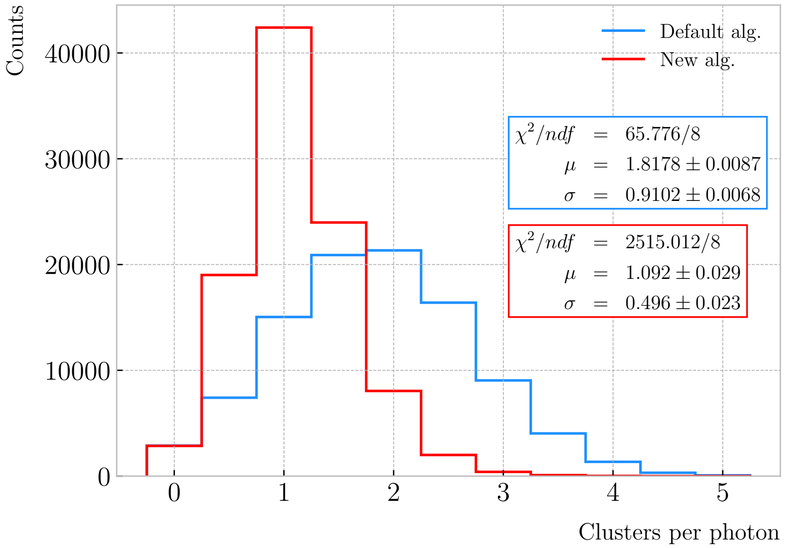
- As a last step, we check if clusters in neighbouring modules should be merged together.

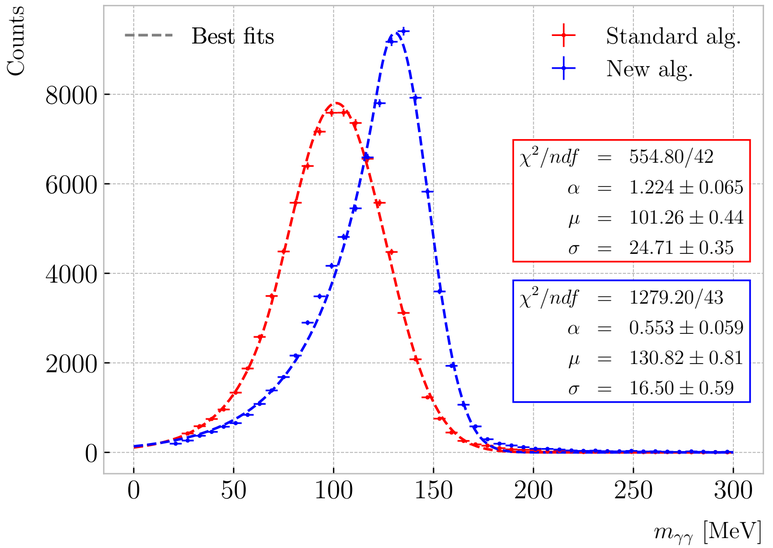
- The algorithm depends on 8 parameters, that were optimised using a $\nu_\mu$ CC sample.



# Neutral pion identification

- To test the potential impact of the new algorithm in $\pi^0$ reconstruction, I simulated single, monoenergetic, forward-going $\pi^0$s inside the TPC.
- One thing to notice is that the number of clusters produced per photon has decreased. Now it peaks at around $1$.
- The invariant mass is computed for all possible cluster pairs, using their position together with the true decay position.
  - In a more realistic scenario, e.g. $\nu_\mu$ CC interaction, we could use the position of the reconstructed primary vertex instead.
  - Tried to use cluster direction to determine the opening angle, but didn't really worked.



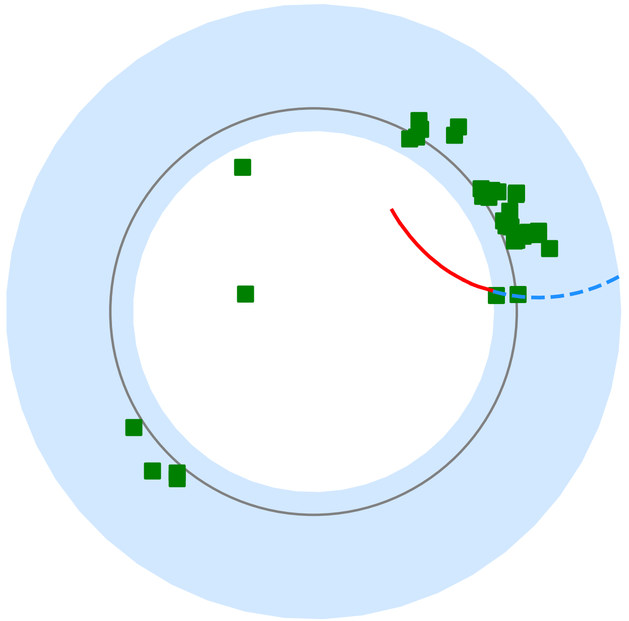
# Track-ECal associations

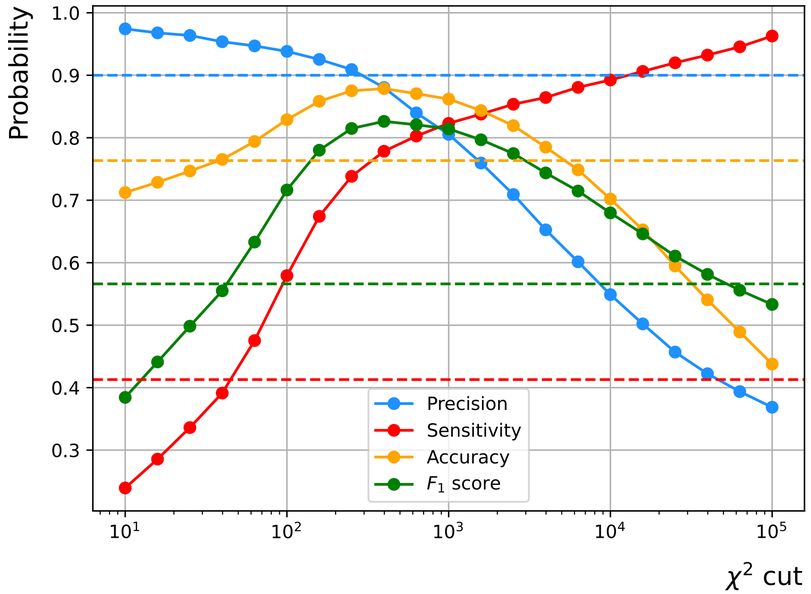
- One of the main players in the PID, in particular for muon/pion separation, is the way we associate clusters in the ECal to reconstructed tracks in the TPC.
  - Missing some associations or making wrong ones can bias the ECal quantities that we can use for classifying particles.
- The current algorithm in GArSoft (`TPCECALAssociation`) provides precise associations, but it appears to miss an important number of them (at least with the current configuration).
- A feature of the code to only associate one end of a track (if any) to a cluster, but it can associate more than one track to the same cluster.
  - This makes sense, as different particles can contribute to the same cluster, but it makes it difficult to quantify the relative contributions.
- While trying to understand the default algorithm I ended up writing a new module for the associations.

# Naive associations

- A simpler algorithm based on propagating the helix up to the radial position of the cluster using the Kalman fit parameters at each end of the track.

- For each reco track the code provides two collections of $\chi^2$ values, one for each ECal cluster and track end.






- To associate a cluster to a track we take all clusters with a $\chi^2$ value in the range $[0, \chi^2_{cut})$.
  - We keep the track end with more entries below the cut.
  - If a cluster has been assigned to more than one track, we associate it to the one with a lowest $\chi^2$.

# Conclusions

- Non-linear relation between $\mathrm{d}Q/\mathrm{d}x$ and $\mathrm{d}E/\mathrm{d}x$ is expected, but is ADC saturation the only responsible? Is the 12-bit ADC a design choice? If so, should we correct for this effect in the reco?
- Pion breakpoint finding can help getting the pion multiplicity right in some cases, but still needs to be tested with neutrino events.
- New ECal clustering works, performance is good and it has an effect in $\pi^0$ reconstruction. However, it's a quite complex algorithm, so more careful study is probably needed.
- The simpler Track-ECal associations may work for interactions in ND-GAr, need to see the impact in pion/muon classification.
- None of these add-ons are in the GArSoft repo yet.
  - I've written the producers that modify parts of the default reco chain as new, independent modules, so they can be added to the producers list in the fcl.
  - I want to do a few final checks, but should be able to make some PRs soon.

# Conclusions

- There are other reco items I'd like to look into but didn't because of lack of time:
  - Right now TPC waveforms are simulated without noise, also it looks like the only noise model available is uncorrelated Gaussian noise.
  - I noticed strange effects in the TPC cluster charge, could be good to study the performance of that clustering algorithm.
  - Pattern recognition usually fails in crowded environments, also not sure about track end point finding resolution.
  - Sometimes vertices are not reconstructed (~1/3 of the time in $\nu_\mu$ CC events).
  - There's a weird glitch when visualising the 12 sided ECal (default) geometry, could it be an overlap?
  - Wouldn't it be better for pointing to have 4 scintillator layers in the MuID?

# Backup slides

# Track-ECAL associations

- The current TPC track - ECAL cluster association algorithm in GArSoft is basically divided in 4 parts:

1. Identify which end of the track we're dealing with and check whether the point fulfils the conditions to be extrapolated. [Cuts: `TrackEndXCut`, `TrackEndRCut`]

2. Get the coordinates of the centre of curvature using the Kalman fit parameters $(y,\ z,\ 1/R,\ \phi)$, compute the distance between it and the cluster in the $(z,\ y)$ plane and compare with $R$. [Cut: `PerpRCut`]

3. Here it depends if the cluster is in the barrel or one of the endcaps:

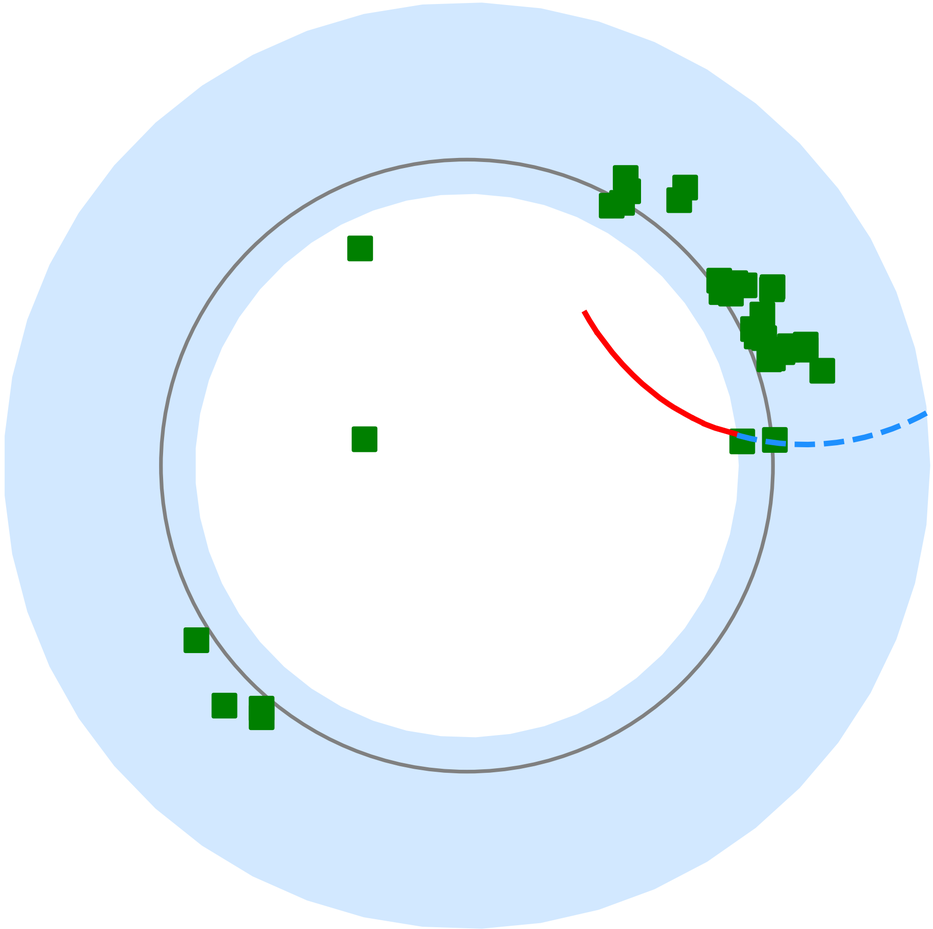
   3.a. If it's in the barrel, extrapolate the track up to the radial distance of the cluster. There are 3 possible outcomes: it cuts the cylinder of radius $r_{clus}$ two, one or zero times. Get the cut point that is closer to the cluster and check that it's either in the barrel or the endcaps. Compute the difference between the $x$ coordinates of the cluster and the extrapolation, and check it's not greater than a certain cut (minus a correction from the $x$ position uncertainty related to not knowing the $t_0$). [Cut: `BarrelXCut`]

   3.b. If it's in the endcap, propagate the track up to the $x$ position of the cluster. Then, compute the angle in the $(z,\ y)$ plane between the centre of curvature and the cluster ($\alpha$) and the centre of curvature and the propagated point ($\alpha'$). Apply a cut to $(\alpha' - \alpha)\ R$. [Cut: `EndcapRphiCut`]

4. Get the direction of the track at the propagated $x$ value obtained before and compute the dot product with the cluster direction if there's a minimum number of hits in the cluster. [Cuts: `ClusterDirNhitCut`, `ClusterDirCut`]

# Naive helix propagation association I

- For each event $e$ and reco track $i$, select the forward or backward fit direction based on position of true vertex.
  - Get the fit parameters at that point together with the x position, $x_0$, ($y_0$, $z_0$, $1/R$, $\phi_0$, $\lambda$).
- For each ECAL cluster $c$, compute the radial distance to the centre of the TPC and find the $\phi$ value in the range $[\phi_0, \phi_0 + \mathrm{sign}(R)\phi_{max})$ that makes the propagated helix intersect with the circle defined with such radius.
- Compute the $(x, y, z)$ position of the helix for the $\phi$ value found (if any).
  - In case there are two intersections keep the one that minimises the distance between $(y, z)$ and $(y_c, z_c)$.
- Compute $\chi^2$ value and store. If there was no intersection then store a $-1$.



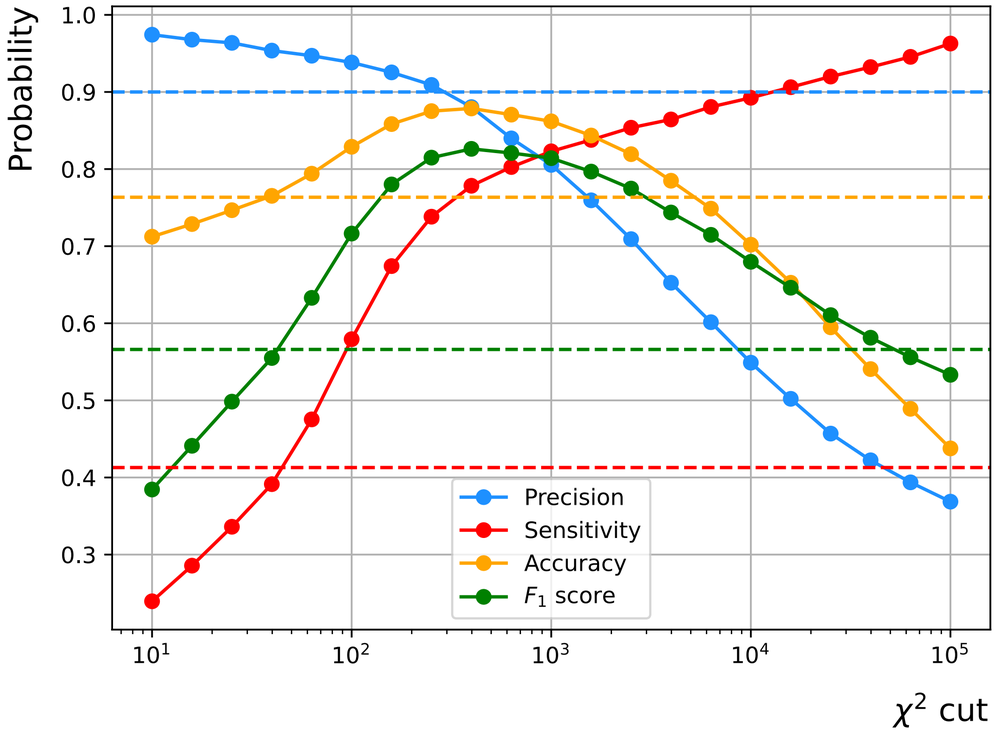
$$\chi^2/dof = \frac{\sum_{n=0}^{2}\left(x^{(n)} - x_c^{(n)}\right)^2}{3}$$

# Naive helix propagation association II

- For each reco track in the event you'll have a collection of $\chi^2$ values, one for each ECAL cluster.
- To associate a cluster to a track we take all clusters with a $\chi^2$ value in the range $[0, \chi^2_{cut})$.
  - If a cluster has been assigned to more than one track we leave it with the one with a lowest $\chi^2$.
- We can evaluate the efficiency of the association method for different values of $\chi^2_{cut}$.
  - Also, we can compare its performance against the standard GArSoft approach.

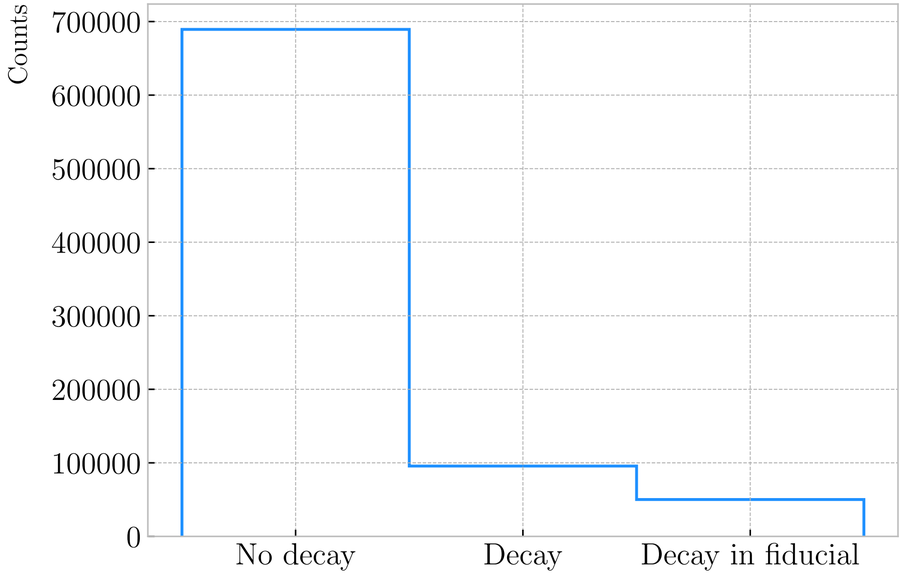
$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad \mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}$$

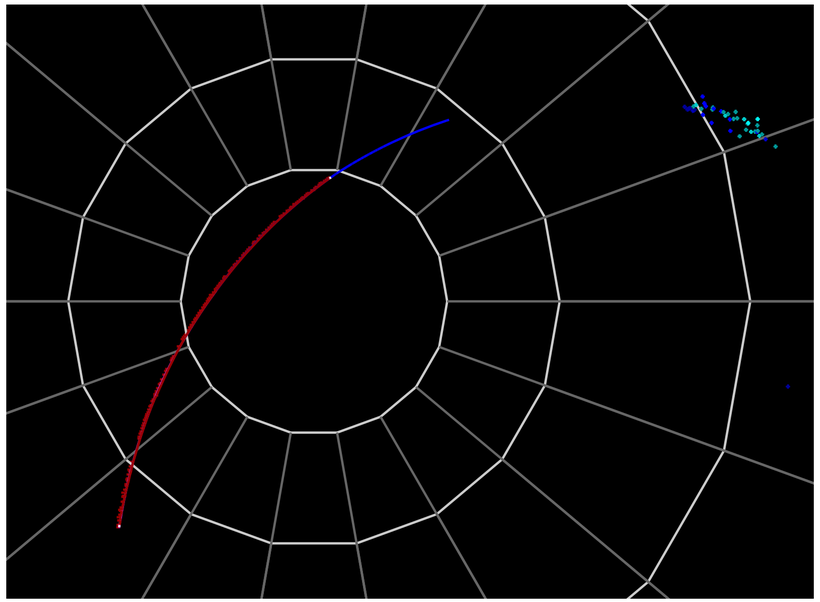
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$



# Pion decays in TPC

- Considering the mean life of the charged pion, $\tau = (2.6033 \pm 0.0005) \times 10^{-8}$ s, we can estimate that about $12\,\%$ of the pions with momentum $p \sim \mathcal{O}(500\ \mathrm{MeV})$ (roughly the peak of the pion momentum distribution in $\nu_\mu$ CC interactions) decay inside the TPC.



- The pion (red) decays in flight inside the TPC but because the angle of the muon (blue) is small both are reconstructed as one single track.
- The “composite” track reaches the ECAL, where it undergoes a muon-like interaction, thus being classified as a muon.

# Track breakpoint analysis I

- In order to identify potential decays we can use the information we obtain from the Kalman filter at each step of the fitted track.
  - The simplest test we can think about is computing the $\chi^2$ of the mismatch between the parameters in the forward and the backward fits:

$$\chi_k^{2\ (FB)} = (\hat{\mathrm{x}}_k^B - \hat{\mathrm{x}}_k^F)^T [V^{(\hat{\mathrm{x}}_k,B)} + V^{(\hat{\mathrm{x}}_k,F)}]^{-1} (\hat{\mathrm{x}}_k^B - \hat{\mathrm{x}}_k^F)$$

- An alternative could be using a fit with a more elaborate breakpoint hypothesis, so we can perform a comparison of the $\chi^2$'s with and without breakpoints.
  - This can be achieved by using some alternative parametrisation with extra parameters, which allows some of the track parameters to be discontinuous at certain points.
  - A decay changes the momentum magnitude and direction, so we can use:

$$\alpha = \left(y, \quad z, \quad 1/R_F, \quad 1/R_B, \quad \phi_F, \quad \phi_B, \quad \tan\lambda_F, \quad \tan\lambda_B\right)^T$$

# Track breakpoint analysis II

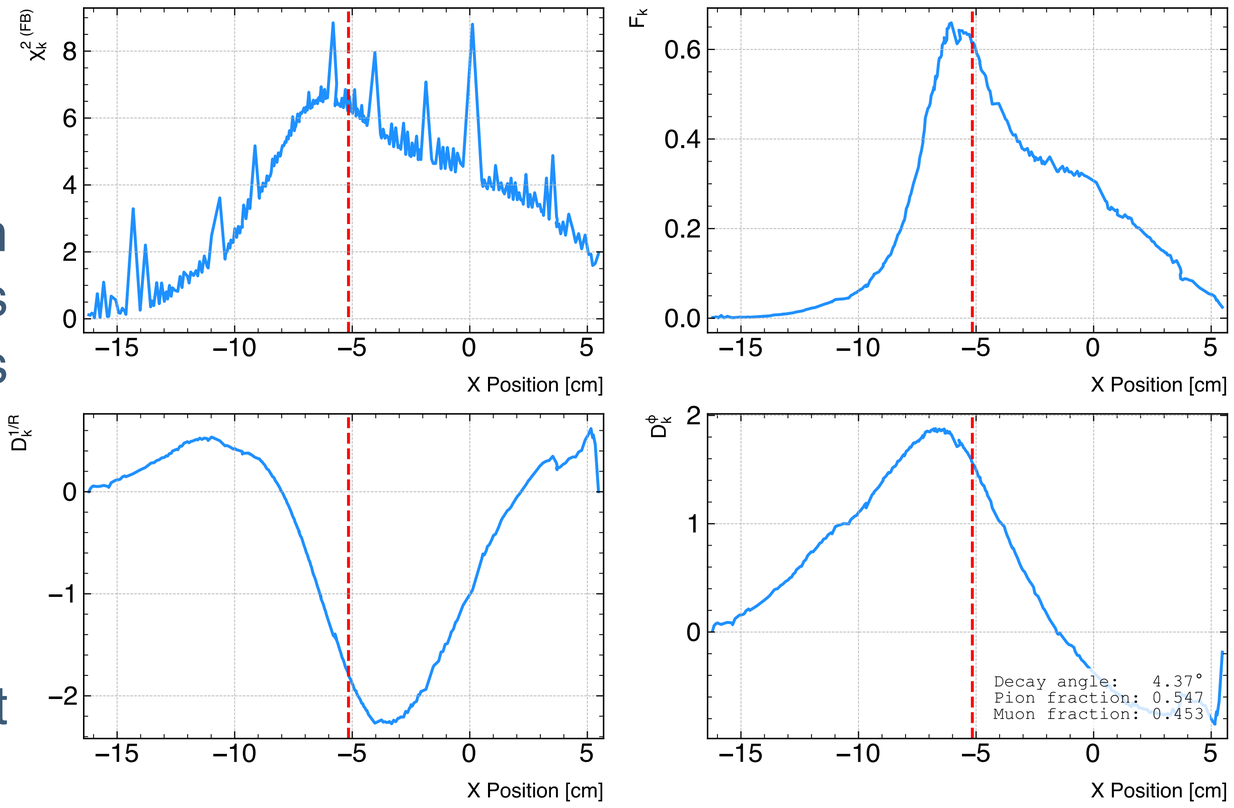
- Because we already have the results from the standard Kalman filter at each point and the model relating $\alpha$ and $\{\hat{\mathrm{x}}_k^B, \hat{\mathrm{x}}_k^F\}$ is linear, we can get the estimates $\hat{\alpha}_k$ at each point as the values that minimise the new $\chi^2$ of the forward-backward mismatch:

$$\chi_k^{2\ (FB)}(\alpha) = (\hat{\mathrm{x}}_k^F - H^F\alpha)^T\left[V^{(\hat{\mathrm{x}}_k,F)}\right]^{-1}(\hat{\mathrm{x}}_k^F - H^F\alpha)$$
$$+(\hat{\mathrm{x}}_k^B - H^B\alpha)^T\left[V^{(\hat{\mathrm{x}}_k,B)}\right]^{-1}(\hat{\mathrm{x}}_k^B - H^B\alpha)$$

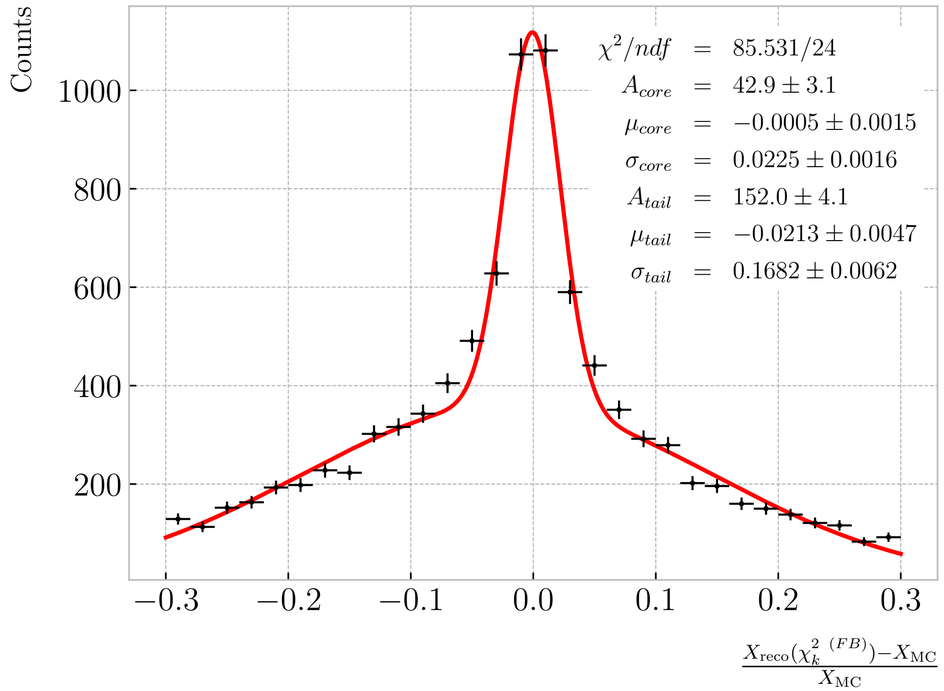
- From these new fit estimates we can compute the $F$ statistic, which tells us whether the model with breakpoint provides a statistically significant better fit:

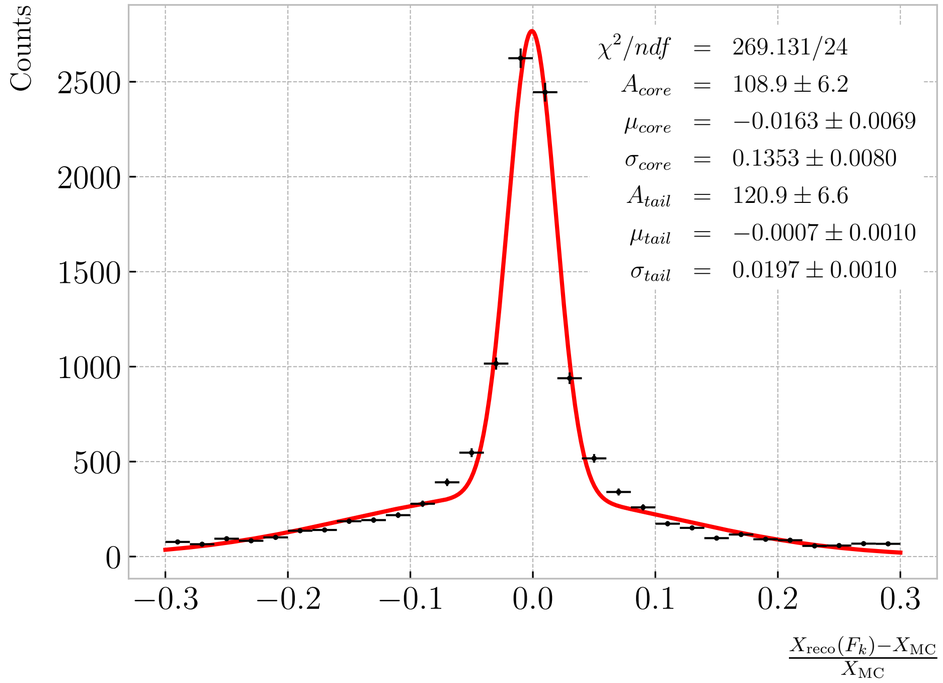
$$F_k = \left(\frac{\chi^2_{track,k} - \chi^2_{full,k}}{8-5}\right) / \left(\frac{\chi^2_{full,k}}{N-8}\right)$$

- We can also get the signed difference at each point for the duplicated variables.



# Track breakpoint analysis III




- We can compare the position of the decay along the drift direction ($X$) with the location of the maxima of $\chi^{2\ (FB)}$ and $F$.
  - Fitting a double Gaussian to both distributions we find a resolution of $13.62\,\%$ and $7.45\,\%$ respectively.

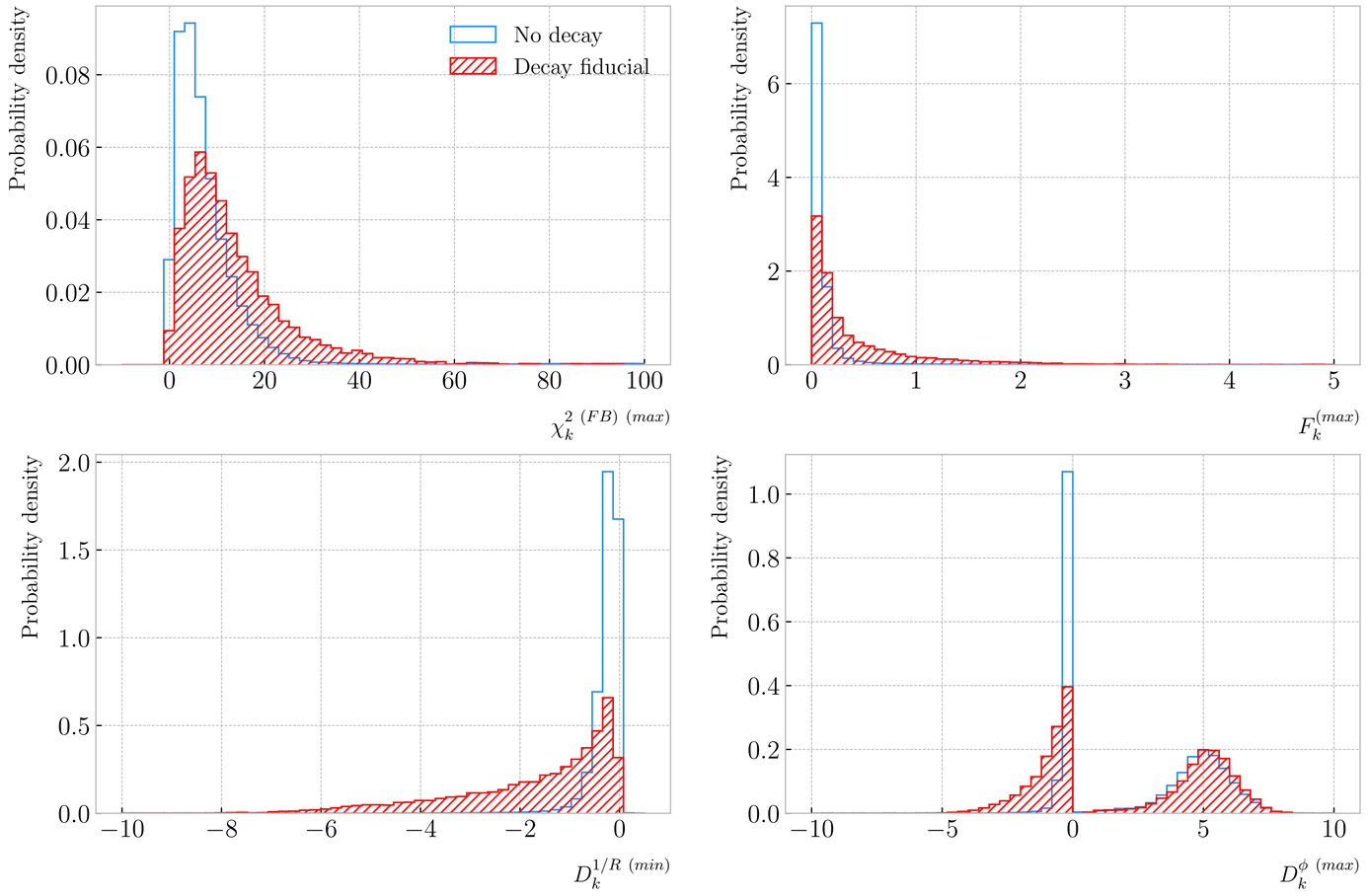
# Track breakpoint analysis IV

- In principle, the $F$ test should follow a Fisher distribution with $(8-5)$ and $(N-8)$ degrees of freedom under the null hypothesis.
- In most of our cases $N \sim \mathcal{O}(100)$, so the PDFs look pretty much the same.

$$\tilde{f}(x;a,b) = \lim_{N\to\infty} f(x;a,b,N)$$

$$= \frac{2^{-\frac{a-b}{2}}}{\Gamma\left(\frac{a-b}{2}\right)} (a-b)^{\frac{a-b}{2}} x^{\frac{a-b}{2}-1} \mathrm{e}^{-\frac{a-b}{2}x}$$

- For this distribution, we obtain a p-value of $0.05$ at $x = 2.60$.
  - From a practical point of view, that's not an efficient way of selecting the decay events.



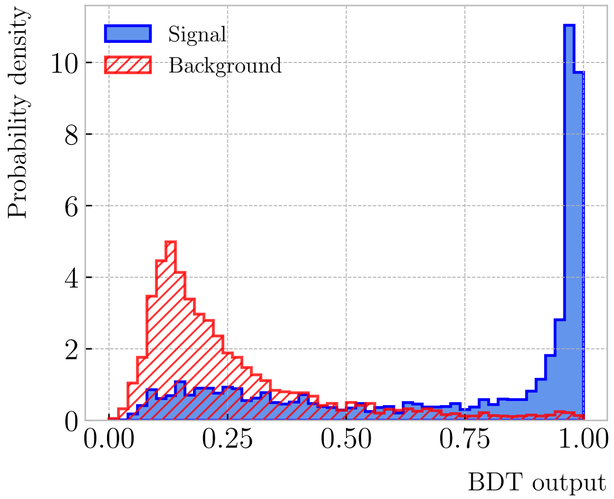
# Track breakpoint analysis V

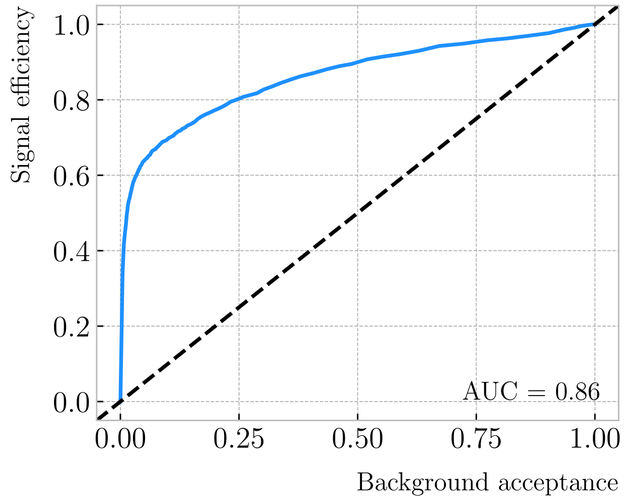
- We can use a combination of our four variables to distinguish between the pion decay events (signal) and the non-decaying pions (background).

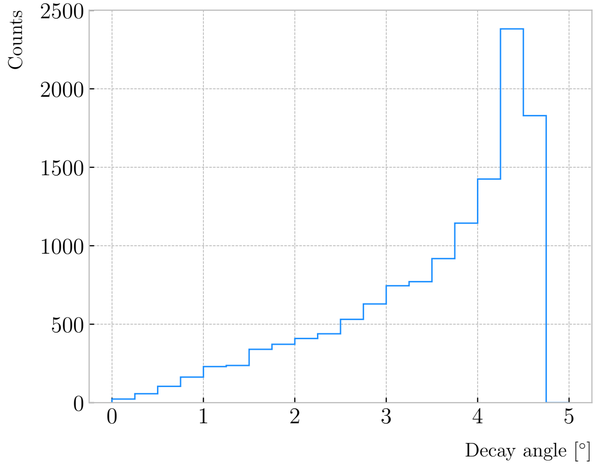
- A way of doing this could be using a BDT (`n_estimators` $= 400$, `max_depth` $= 4$).

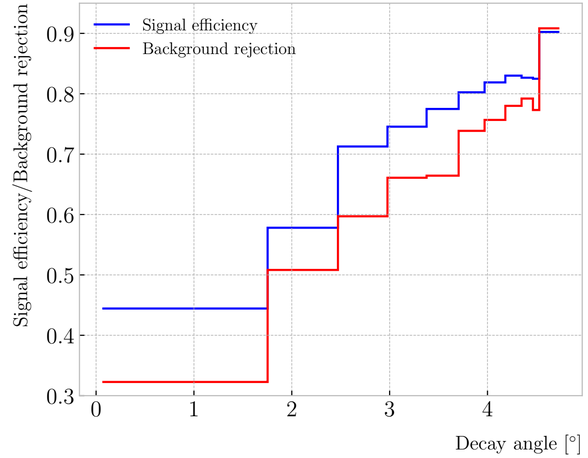
  - The most important variable turned out to be $D_k^{1/R\ (min)}$.

- We can check how the signal efficiency and the background rejection changes with the true decay angle.



# ECAL clustering I

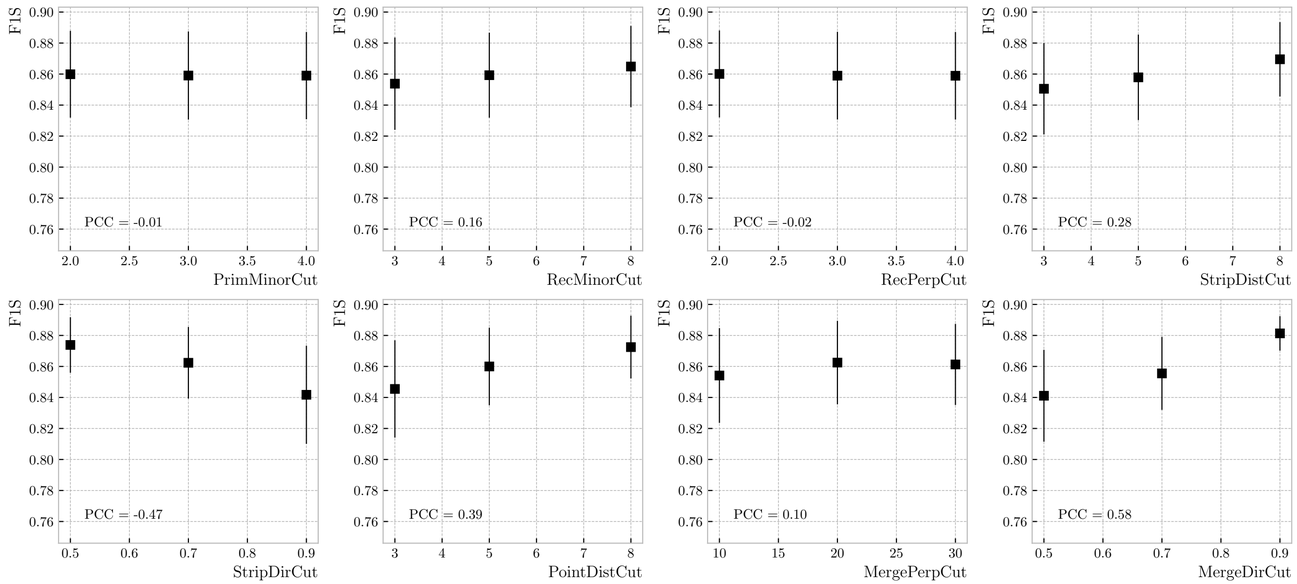
- For the first part I cluster together all the hits which are in nearest-neighbouring strips and next-to-nearest-neighbouring layers (as the layers with strips along the two directions are alternated).
  - An additional cut in the direction along the strip length is needed.
- Then we loop over the clusters with $N \geq 2$, computing the centre of mass and three principal components. I propagate these three axes up to the layers of the rest of the clusters, and if the propagated point and the centre of mass of the second cluster are within next-to-nearest-neighbouring strips I merge them.
  - Again, an additional cut in the direction along the strip length is needed.
  - Require also that the two closest hits across the two clusters are at most in next-to-nearest-neighbouring strips.
  - Repeat this re-clustering until no more pairs pass the cuts.

# ECAL clustering II

- For tile layer, I now require hits to be in next-to-nearest-neighbouring tiles and nearest-neighbouring layers in order to be clustered together.
- At the moment, I am merging together strip clusters with different strip directions if their centres of mass are close enough and they point in the same direction.
  - Would like to instead compute the overlap between the ellipsoids defined by the principal axes of the clusters, and merge the pair if the overlap exceeds some threshold.
- To merge the tile clusters to the combined strip clusters I check for pairs that point in the same direction.
- The last step is to check if clusters in neighbouring modules (both barrel-barrel and barrel-endcap) should be merged together.
- This algorithm has a total number of 8 free parameters that need to be optimised.

# ECAL clustering III

- I used a sample of 1000 $\nu_\mu$ CC interactions in order to obtain the optimal configuration of clustering parameters.
  - I prepared the sample up to the old ECAL hit clustering level and then ran the new clustering algorithm, each time with a different configuration of parameters.
  - As the number of parameters is big, I only performed a coarse-grained scan of the parameter space.
- Select all configurations with purity $\geq 90\,\%$. Among those, choose the combination that yields the maximum $F_1$-score.



# ECAL clustering IV

- For each cluster, identify the matching MC TrkID and energy fraction of each hit.
- Assign to each cluster the MC TrkID with the highest total energy fraction.
- For each of the different TrkIDs associated to the clusters, select the cluster with the highest energy (only from the hits with that TrkID). That'll be the main cluster for that TrkID.
- We call TPs to the hits with the correct TrkID in each main cluster.
- FPs are the hits with the incorrect TrkID for the cluster they are in (not only main clusters).
- FNs are the hits with the correct TrkID in non-main clusters.

# ECAL clustering V

| | Default alg. | New alg. |
|---|---|---|
| TP | 9424 | 9822 |
| FP | 940 | 1628 |
| FN | 2905 | 1819 |
| Precision | 0.91 | 0.86 |
| Sensitivity | 0.76 | 0.84 |
| $F_1$ Score | 0.83 | 0.85 |

**Table 1.** Clustering metrics for 100 $\nu_\mu$ CC events.

| | Default alg. | New alg. |
|---|---|---|
| TP | 4042 | 4704 |
| FP | 194 | 347 |
| FN | 1282 | 467 |
| Precision | 0.95 | 0.93 |
| Sensitivity | 0.76 | 0.90 |
| $F_1$ Score | 0.85 | 0.92 |

**Table 2.** Clustering metrics for 100 $\pi^0$ events.