New training of the track/shower BTD algorithm in Pandora

Neutrino event reconstruction with Time Projection Chamber detector in the ICARUS experiment

Mattia Sotgia, Alice Campani, Lea Di Noto (University of Genoa and INFN), Angela Fava (FNAL)

End term internship report

(Sept. 26th, 2024)

Fermilab

SBN Program

Three experiments, SBND and ICARUS (currently taking data at FNAL) and MicroBooNE (data taking completed in 2019) on the Booster Neutrino Beam at baselines of 110, 470 and 600 meters

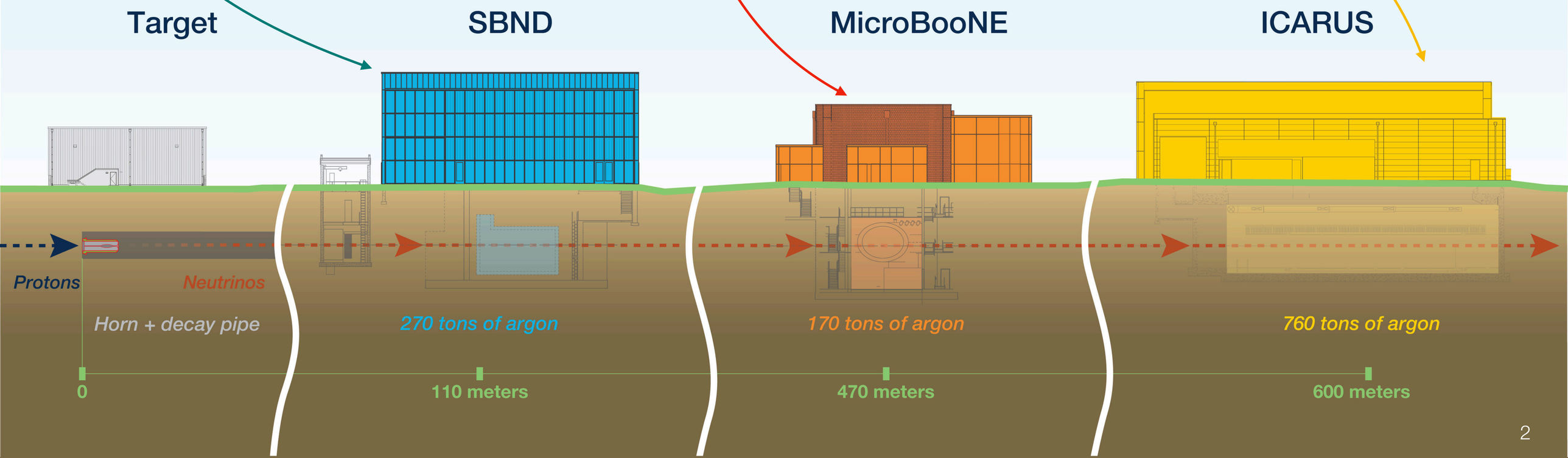
- ICARUS (**SBN-FD**), acting as the Far Detector
  The ICARUS experiment is also interested by **NuMI** beam (6°off-axis)
- SBND (**SBN-ND**), acting as the Near Detector

The third experiment on the **BNB** baseline is the MicroBooNE experiment, in the middle

Scientific goals

- Sensitive search (5σ) for 1eV sterile ν in 3 years of data taking
- Study the interaction ν(~3 GeV)-LAr for future developments in DUNE
- Search for Beyond Standard Model (BSM) physics

Short-Baseline Neutrino Program at Fermilab



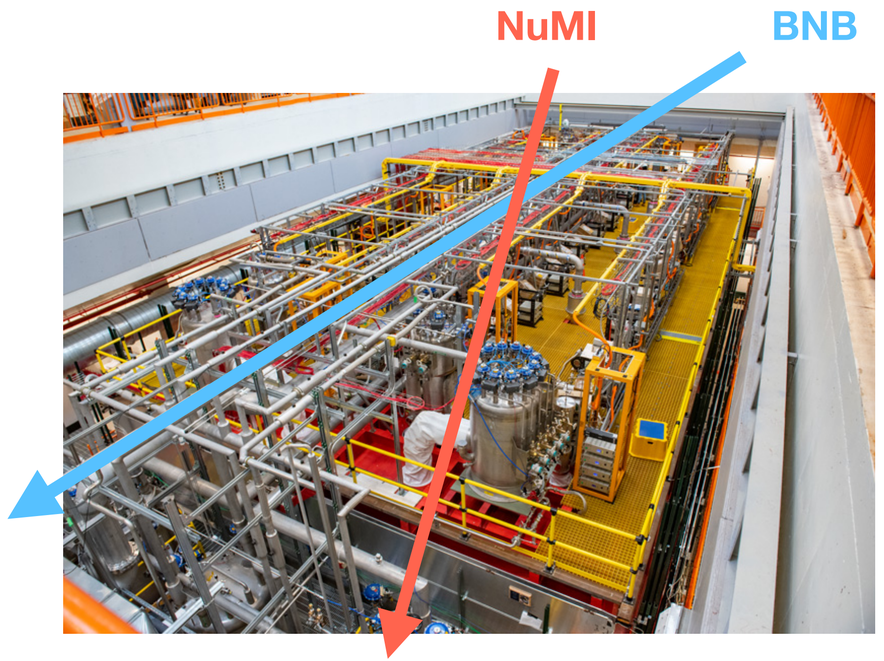
The ICARUS T600 detector

- A Liquid Argon Time Projection Chamber (**LArTPC**) hight granularity self-triggering detector, with 3D imaging and calorimetric capabilities, ideal for ν physics

- Two **cryostats**, each with **2 TPCs** with a common cathode

- Three wire planes (Induction 1, Induction 2, collection, with wire orientation at 0°, +60°, -60° respectively) located at the anode

- The trigger system is based on **360** PMTs and the triggering is done in the 1.6μs/9μs spill window (for BNB/NuMI)



- The detector collect neutrino events both from the Booster Neutrino Beam (**BNB**) and from the Neutrino Main Injectior (**NuMI**)

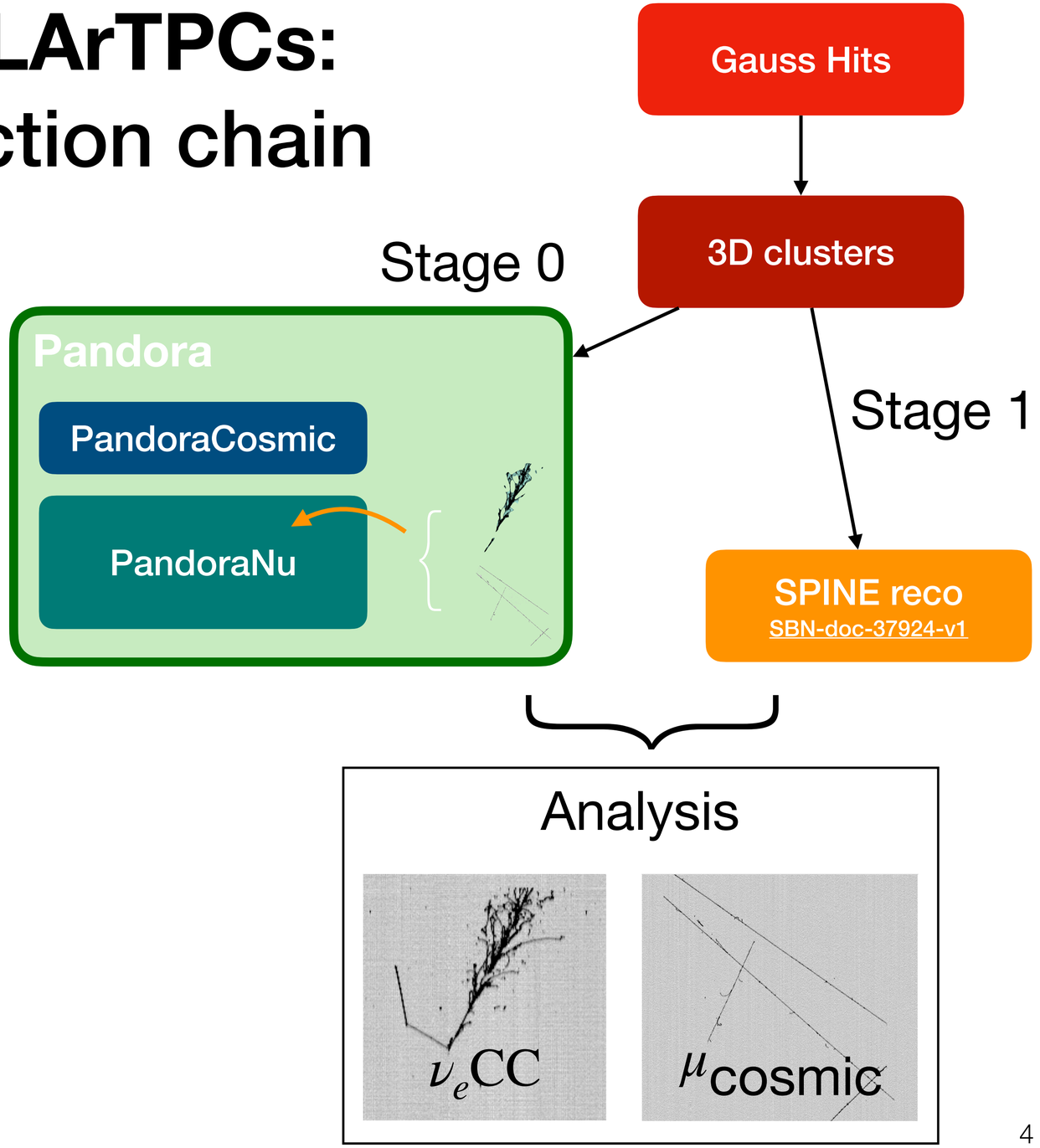
ICARUS at the Fermilab Short-Baseline Neutrino Program - Initial Operation Eur. Phys. J. C 83:467 (2023)

Event reconstruction in LArTPCs: ICARUS event reconstruction chain

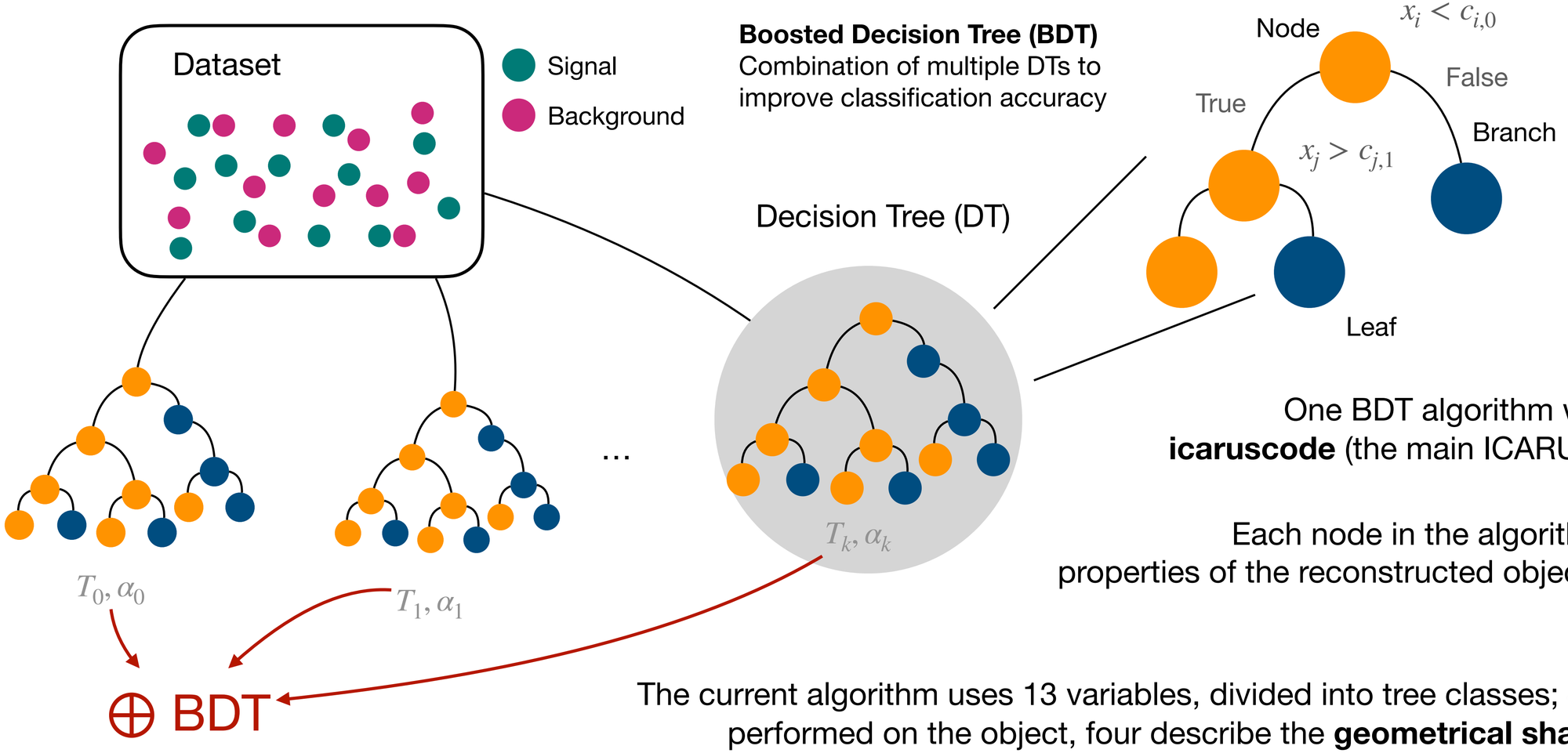
ICARUS analysis is performed trough a chain of subsequent algorithms, performing all the steps from clustering of hits (portions of waveforms with a signal) in 3D to reconstructing the event hierarchy.

From the **Hits** of the single wires, the **2D reconstruction** of the event for each wire plane is performed, and then the **3D Clusters** are made. These then are used as inputs for two different reconstruction algorithms.

I will now focus on a specific step (track/ shower separation) of the Pandora-based reconstruction chain. PandoraPFA is a pattern recognition algorithm.



Track/shower discrimination in the Pandora-based TPC event reconstruction: A Boosted Decision Tree (BDT)



One BDT algorithm was already available in **icaruscode** (the main ICARUS software framework)

Each node in the algorithm uses some physical properties of the reconstructed object to classify it either as a shower or as a track

The current algorithm uses 13 variables, divided into tree classes; 4 are from the **linear fit** performed on the object, four describe the **geometrical shape** of the object, and 5 describe the **energy deposition** of the hits on the TPC wires

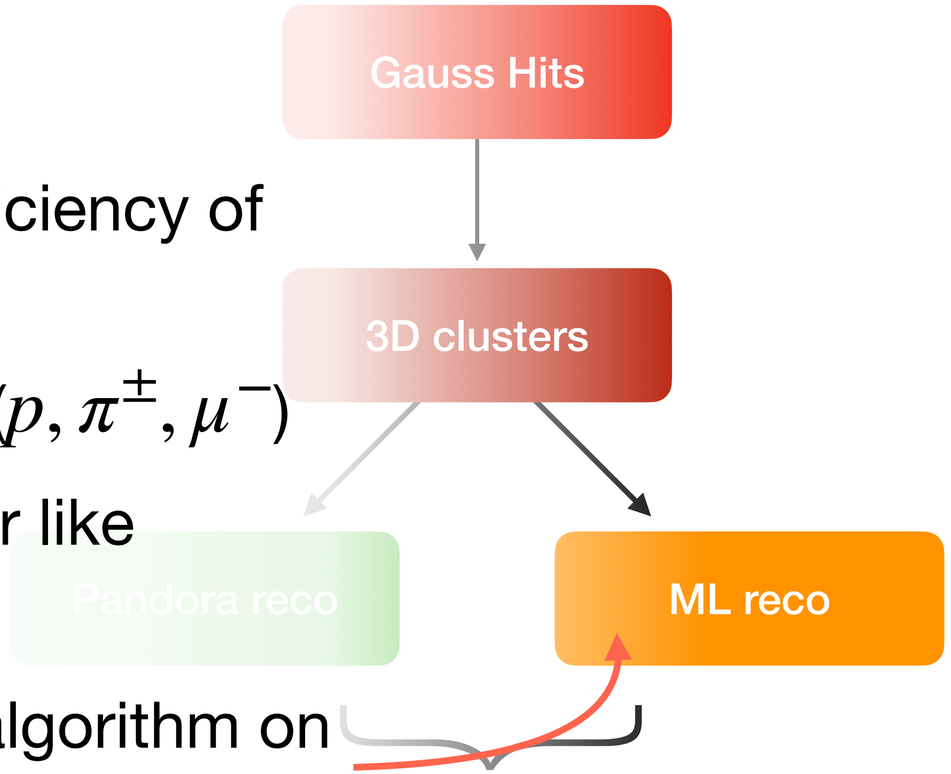
The Tree T_{k+1} starts from the misclassified events of the Tree T_k

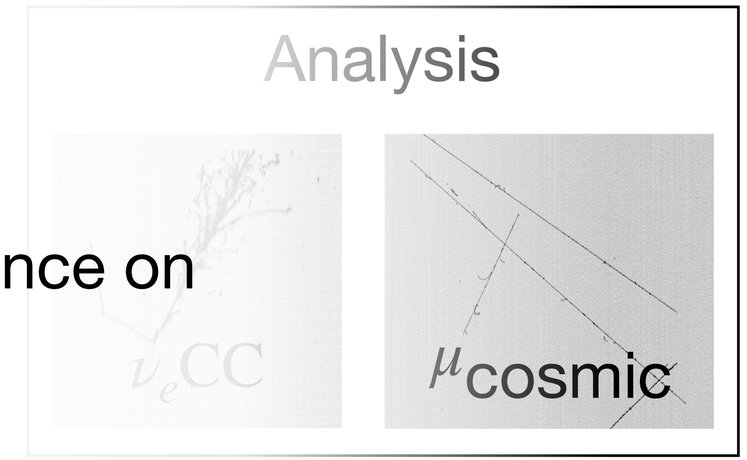
Preliminary tests

Comparing a Booster Neutrino Beam Monte Carlo sample and a uniform energy sample

Track/shower discrimination BDT: testing a new dataset

- The precedent training had a classification efficiency of **~80 %.**
  - Good performance with track like particles ($p, \pi^{\pm}, \mu^{-}$)
  - A slight decrease in performance for shower like particles (e^{-}, γ, π^{0})
- The goal of my internship was to re-train the algorithm on a new Monte Carlo dataset
  - **Different** particle composition
  - **More uniform** energy distribution (no dependence on the signal model)

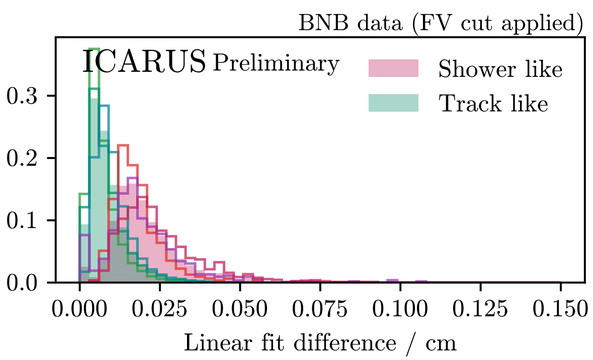


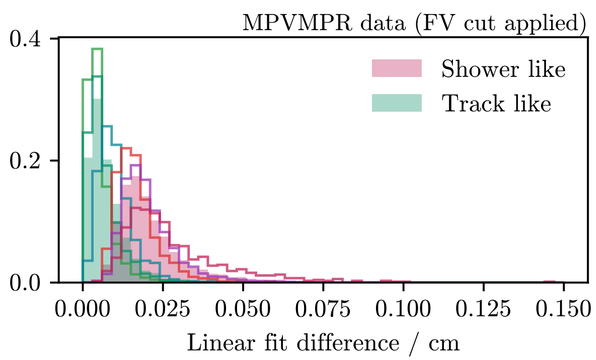


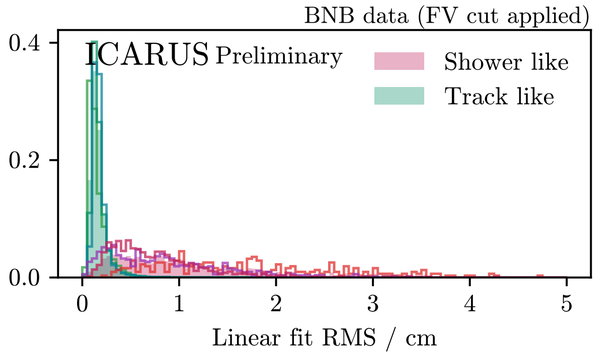
Boosted Decision Tree variables: linear fit variables

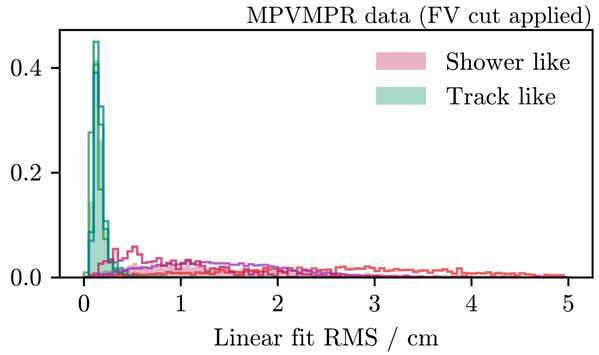
- The best discrimination is obtained by linear fit difference, linear fit RMS and linear fit **gap** length
- Linear fit length is not a useful variable
  - The algorithm uses it **less often** to perform the node cut

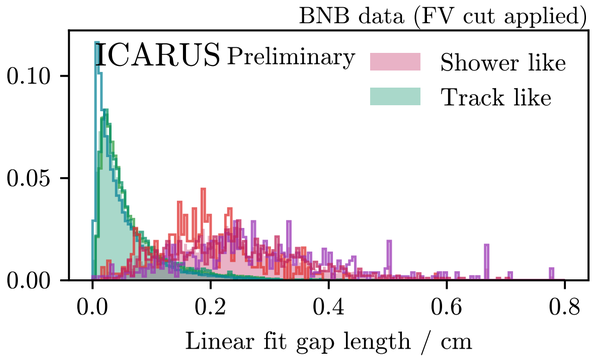
For each BDT variable the filled area show the summed distributions for each class (tracks $\leftrightarrow p, \pi^{\pm}, \mu$ and showers $\leftrightarrow e^{-}, \pi^{0}, \gamma$) and lines show the different particles

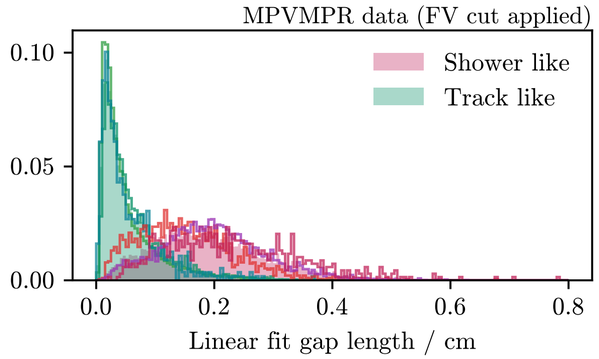


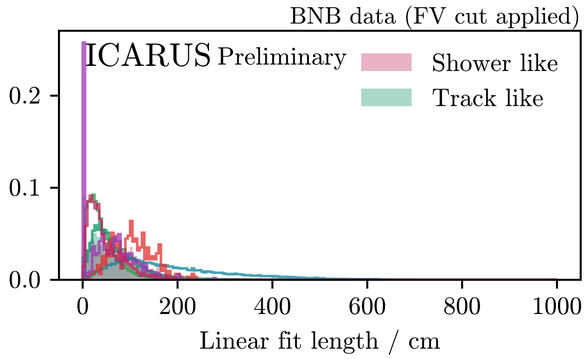


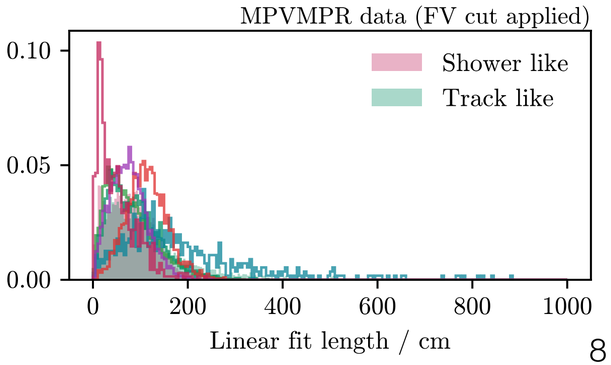










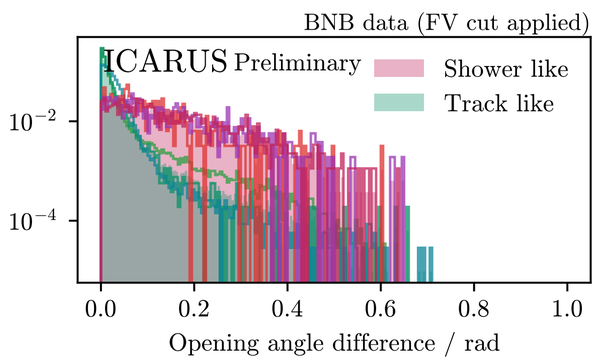


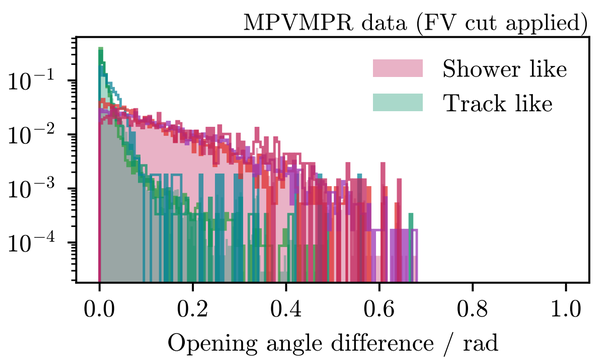
Boosted Decision Tree variables: geometrical variables

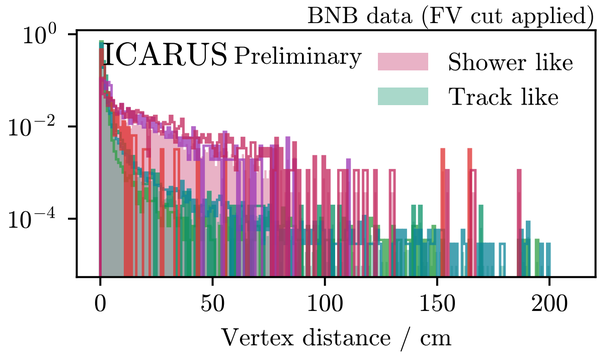
- Track-like and shower-like distributions are often largely overlapped
- PCA2/3 ratios show a high importance in the feature importance plot after the model training

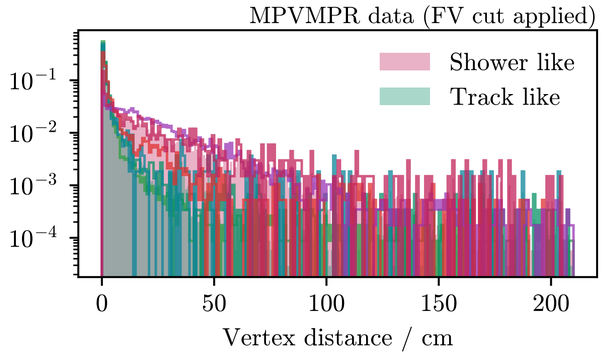
- In the 2D plot, the PCA2 and PCA3 showed a great power wen combined, since their distribution had a different shape

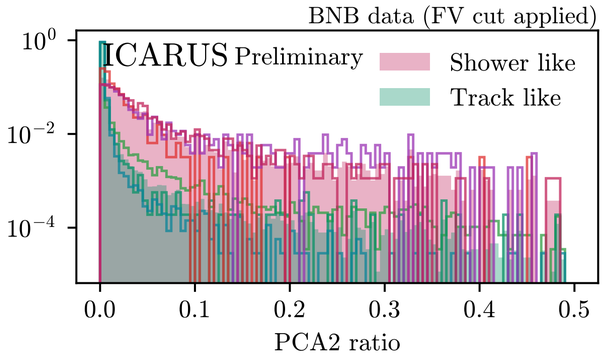
For each BDT variable the filled area show the summed distributions for each class (tracks $\leftrightarrow p, \pi^{\pm}, \mu$ and showers $\leftrightarrow e^{-}, \pi^{0}, \gamma$) and lines show the different particles

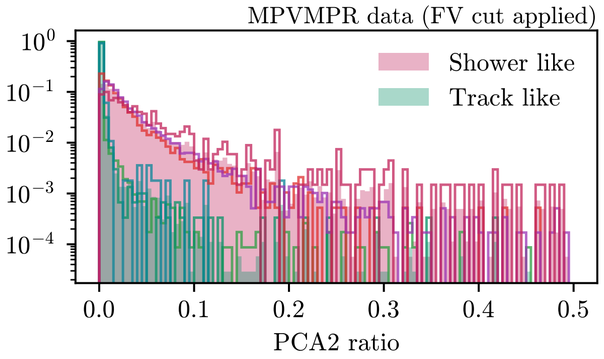


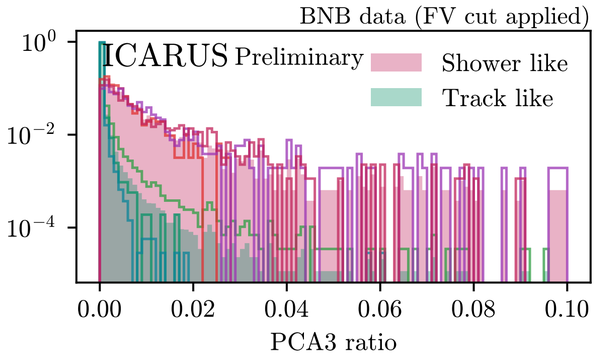


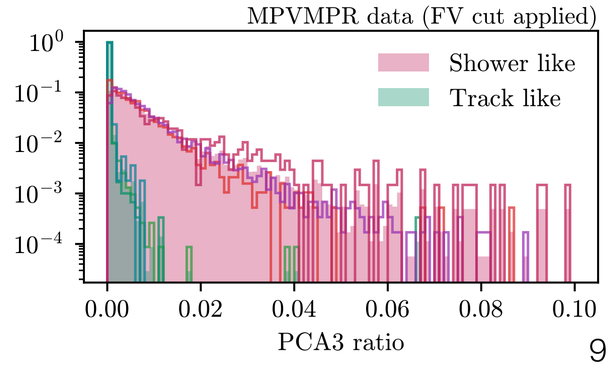






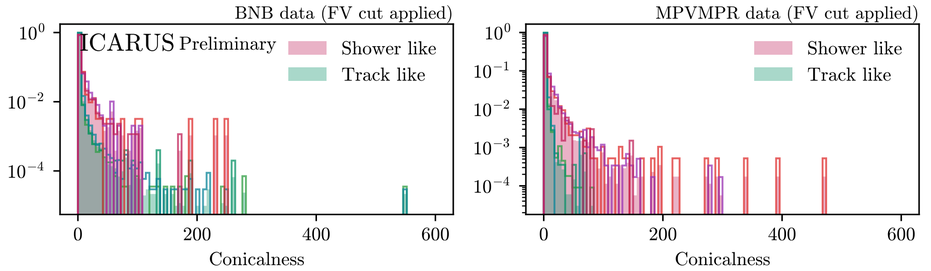


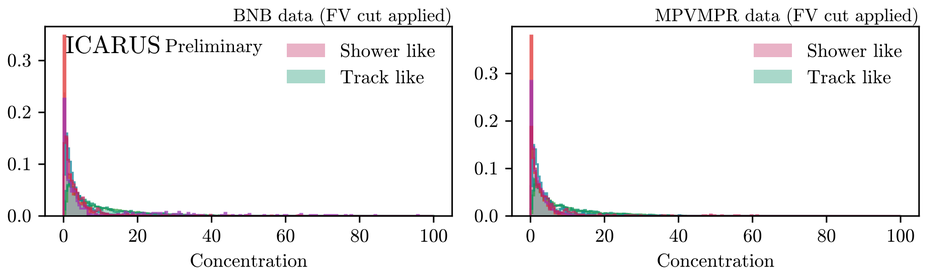


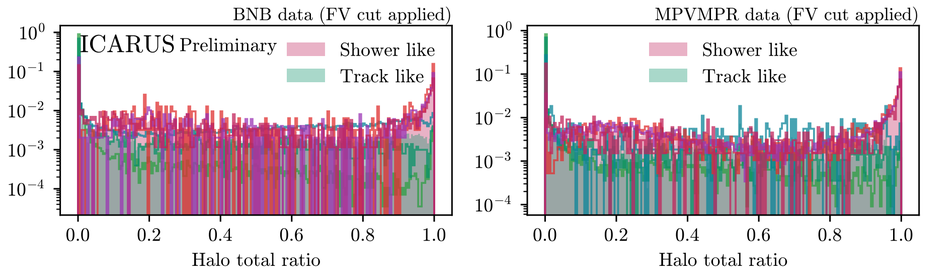


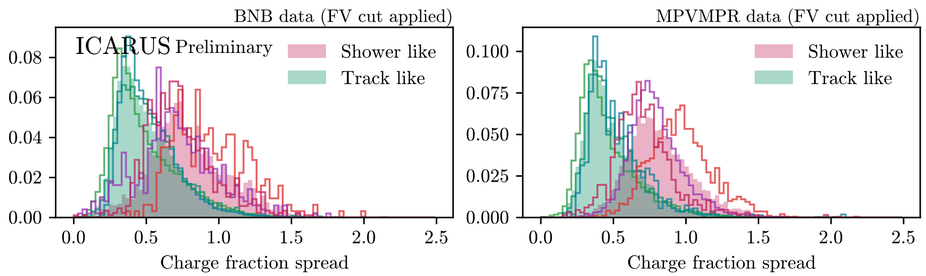
Boosted Decision Tree variables: hit charge variables

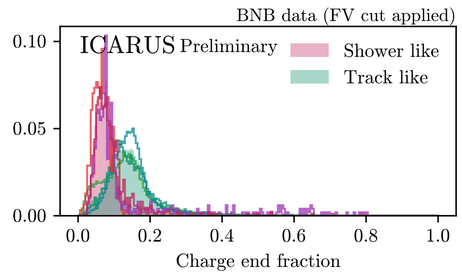
- Charge end fraction and charge fraction spread show great discrimination power
  - When considered together they act as one of the best variable
- The other three (conicalness, concentration, halo total ratio) show a quasi-total overlap between the shower-like distribution and the track-like distribution

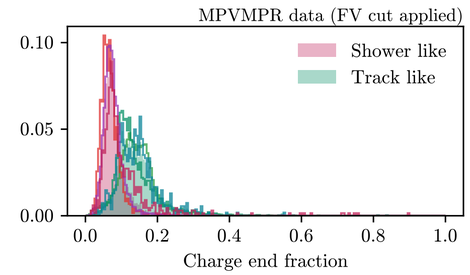












Bad discrimination

Good discrimination

Training the algorithm

Since uniform energy dataset shows a better separation between track-like and shower-like particles, we performed the training on this dataset

Chasing the best configuration

- Other than the parameters on which to make the cut for the nodes (physics driven parameters), the model need also some hyper-parameters
  - Parameters that **boost the trining performance**
  - Can be seen as parameters limits in the fit process

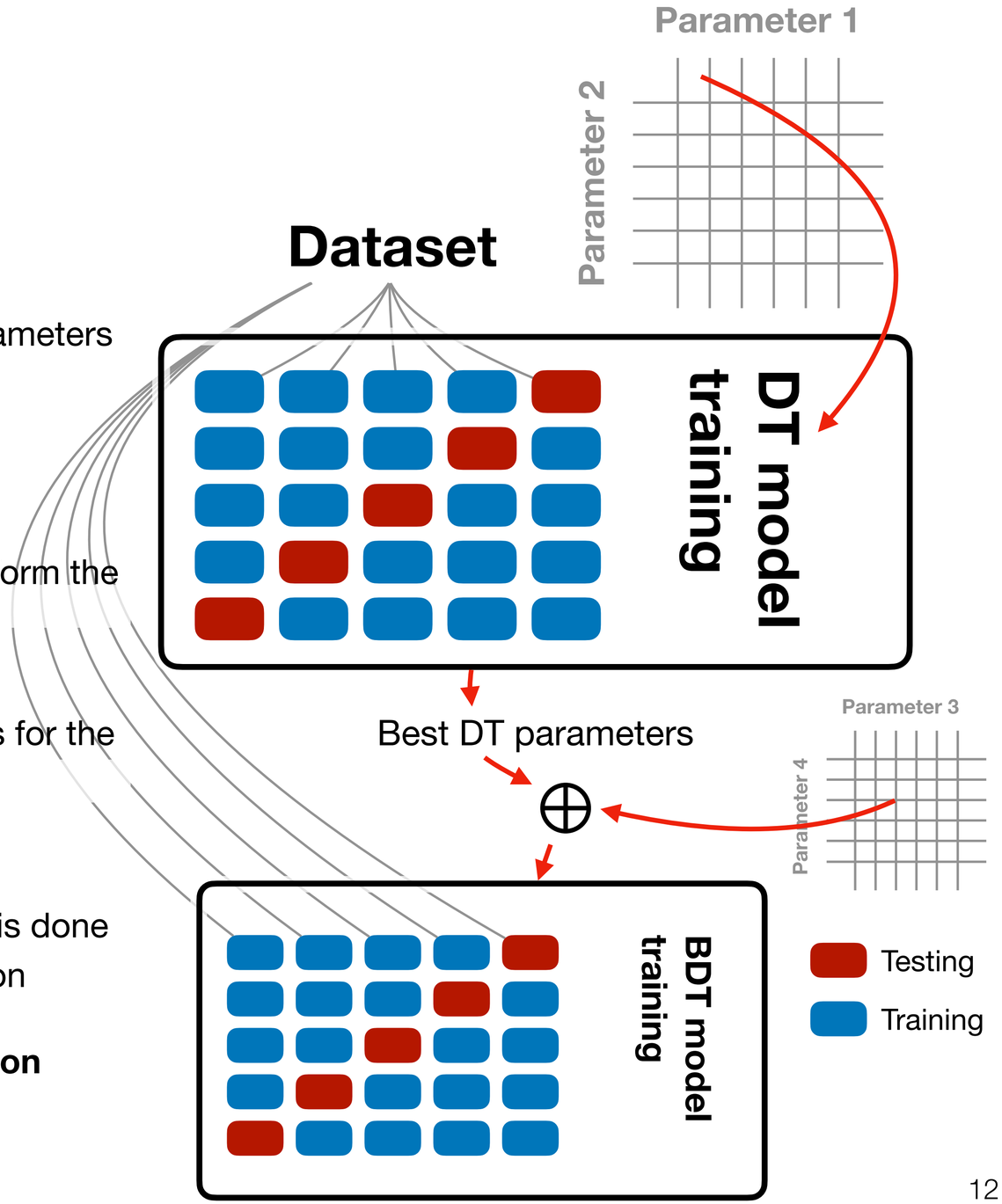
? How can we chose such parameters

  - We create a list of possible values for each parameter and perform the training on each possible combination of those values
  - This is the idea behind the **Cross Validation** process

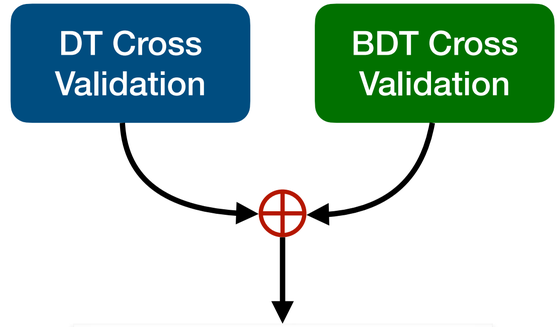
- We first perform Cross validation to find the best hyper parameters for the single Decision Tree, and later for the Boosted Decision Tree

! Each training is also performed using the **k-fold method**

  - The dataset is subdivided into k sub datasets and the training is done k-times, each time with $k-1$ subsets, leaving one for validation
  - This efficiently exploits the dataset to **improve the classification**



Chasing the best configuration



Parameter	Value(s)	
param_max_depth	3	6
param_n_estimators	1000	1000
param_learning_rate	1.25	0.1
mean_test_score	0.8	0.8
std_test_score	0.0023	0.0023

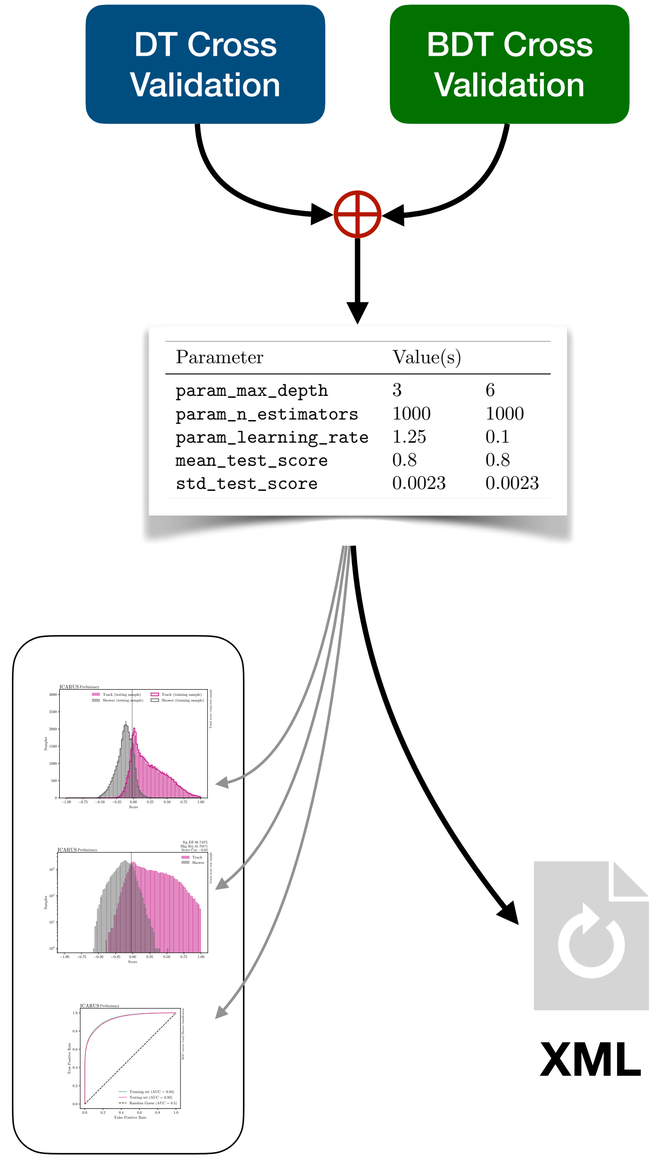
- Results of both **k-fold Cross Validation processes** (single DT and BDT) were ranked
  - **Accuracy** was chosen to be the ranking score metric
- Of the whole grid of parameters, two were essential
  1. (Maximum) Depth of the single Decision Tree
     - result of the Cross Validation on the single Decision Tree
     - Higher depth $\Longrightarrow$ more computationally expensive algorithm
     - Lower depth $\Longrightarrow$ less powerful classification
  2. Number of Decision Trees in the Boosting process
     - result of the CV on the Boosting process
     - More estimators (single Decision Trees) $\Longrightarrow$ more powerful classification, yet more computationally expensive
     - Less estimators $\Longrightarrow$ Less powerful Boosting (extreme case of 1 single Decision Tree)

Training the best model

- We optimized the depth of the single Decision Tree in a range **between 3 and 6**
  - After the second CV **the depth was chosen to be 3**
- The second Cross Validation showed the the best performances were obtained with **1000** Decision Trees per BDT

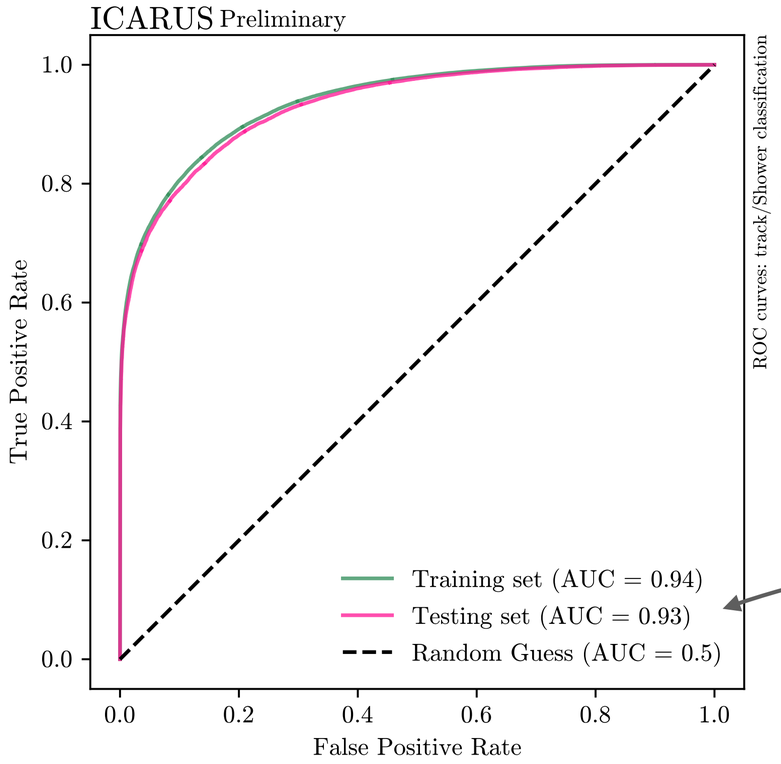
!! The tests show that the training **performs better** if the **number of track-like particles is the same as the number of shower-like particles**

- Using the optimal set of DT and BDT parameters a **final training** was performed



Training the best model

After the training on the 80% of the uniform energy data, the last 20% of the sample was used as test

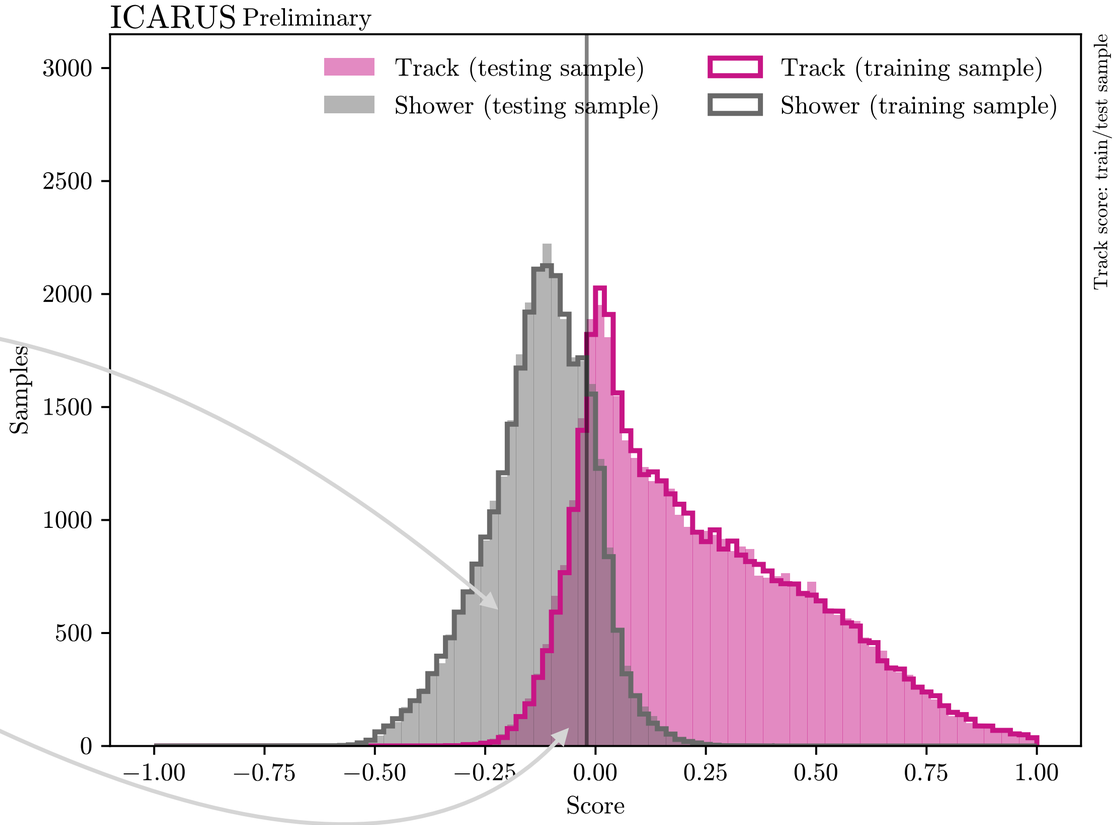


The sample shows **great efficiency** and **reasonable purity** for tracks and showers

Efficiency is the true positive count over the total events

Purity is the false positive count over the total events

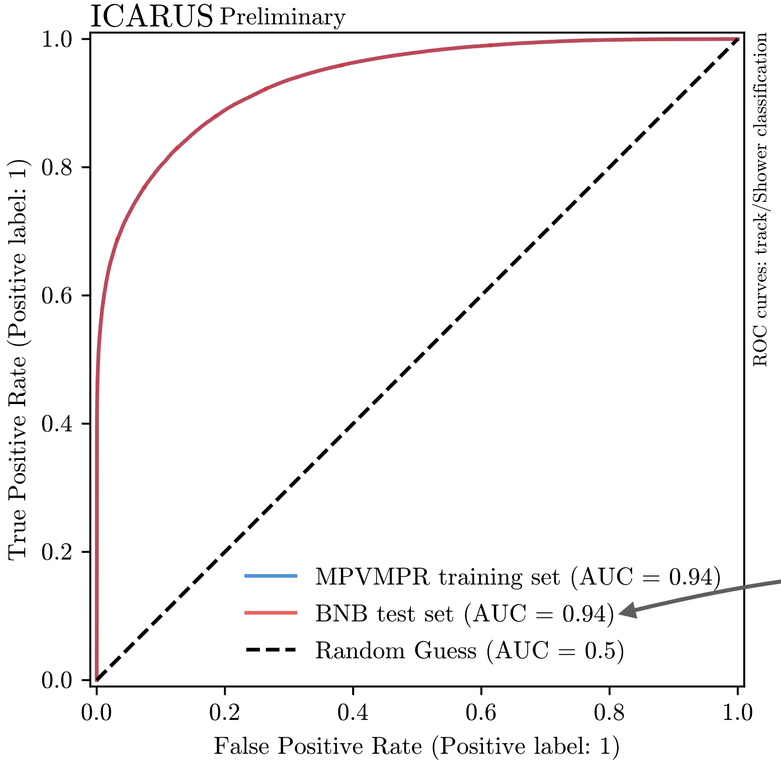
Area(s) Under the Curve are actually similar for training and testing, showing **no overfitting/overtraining**



Testing the algorithm
on a physics driven dataset

Testing on physics driven data

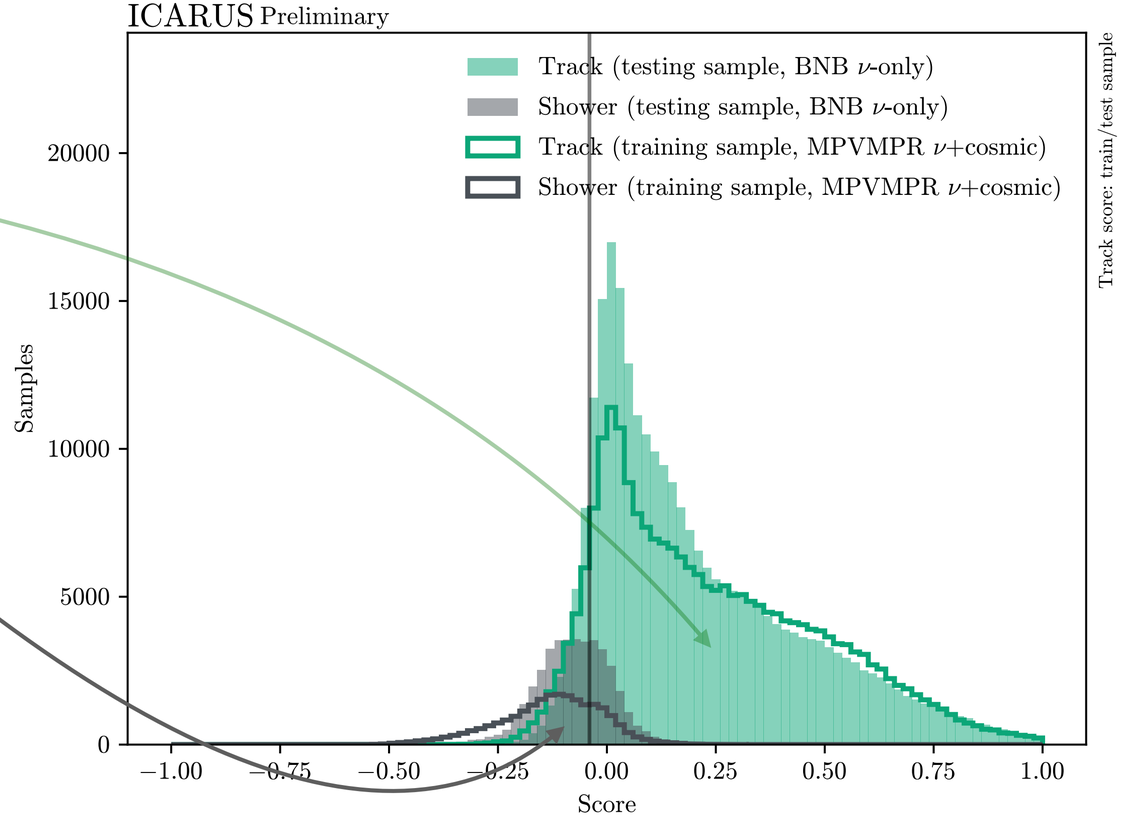
- The newly trained model is useful only if its outperforming the precedent in classifying physics driven data, that is events whose energy distribution and other features reflect the expected beam (BNB) events
- The model is tested against a **Booster Neutrino Beam** simulated dataset of ν**-only events**



The test sample show **great efficiency** and **reasonable purity** for **tracks**

The same is **partially true** for **shower**: good **efficiency**, but lack of **purity**

Area(s) Under the Curve are actually similar for training (**MPVMPR** sample) and testing (**BNB** sample), showing **no overfitting/overtraining**

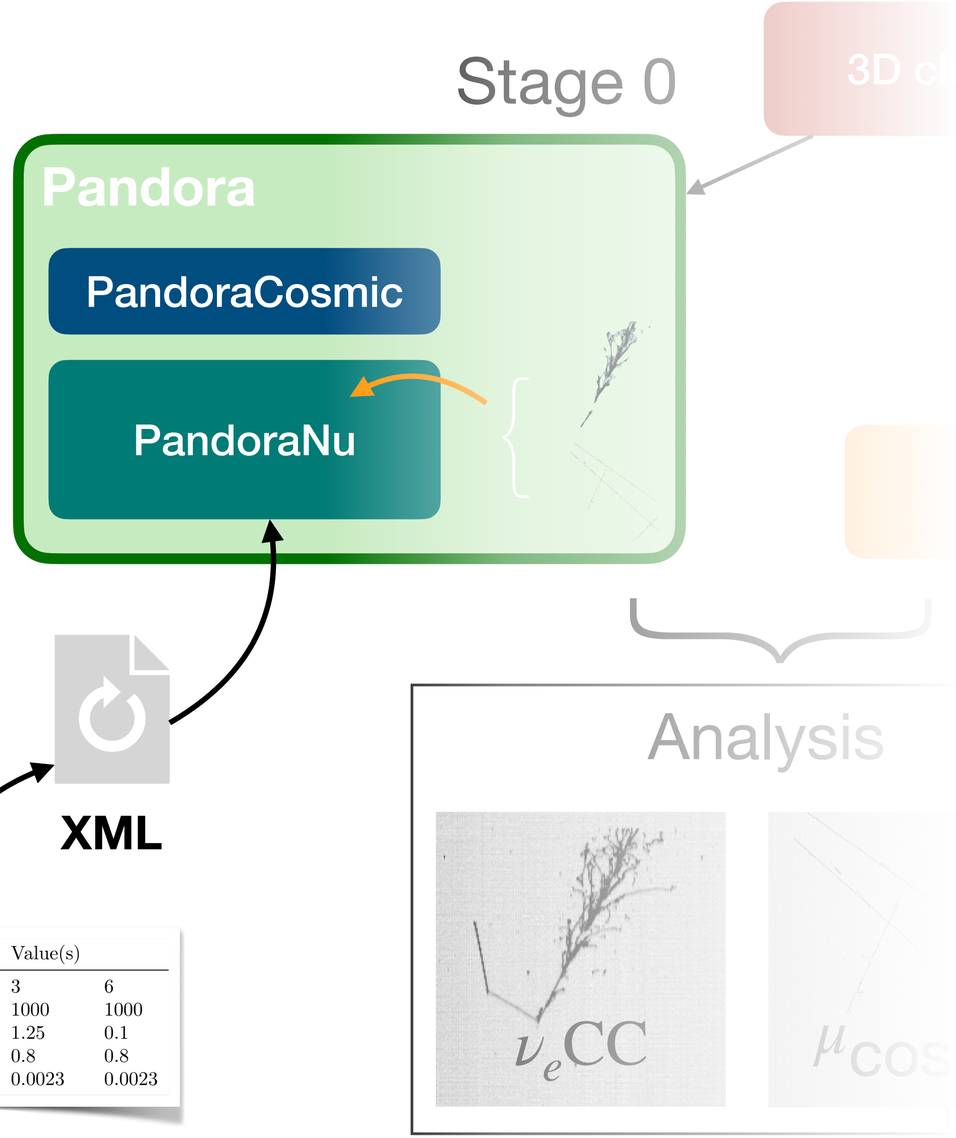


Testing on physics driven data

- The new BDT training show good classification performance (at least in the preliminary stages)

- We can **test** the new BDT training by **including it in the Pandora reconstruction chain** and reprocessing the same data sample

!! This helps us **validating** the **new BDT training**, letting us perform the same study done to compare the two original MC samples

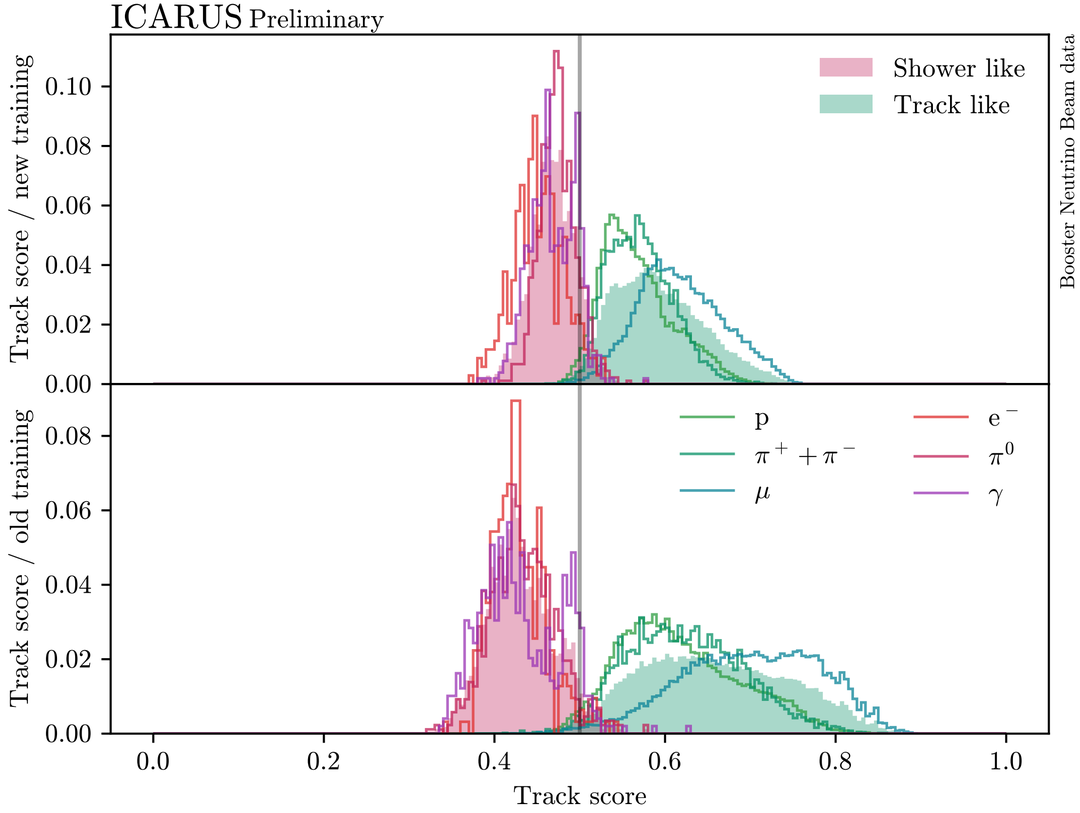


Parameter	Value(s)	
param_max_depth	3	6
param_n_estimators	1000	1000
param_learning_rate	1.25	0.1
mean_test_score	0.8	0.8
std_test_score	0.0023	0.0023

Testing on physics driven data

- The parameter to focus on is the **track score**, that is the outcome of the algorithm in terms of **event classification**

- Using this data we are able to compute the efficiency and the purity for each class of events and each type of particle



The new training shows

- **slight improvement** in the classification **efficiency** of track-like particles
- **slight decrease** in **efficiency** in shower-like particles

		Track-like				Shower-like			
	Training	μ	$\pi^{\pm}$	p		e^-	π^0	γ	
Efficency	Old	99.1%	98.31%	97.81%	98.5%	97.12%	95.96%	95.94%	96.17%
	New	99.67%	99.05%	98.02%	98.93%	95.93%	91.28%	93.6%	92.87%
Purity	Old				99.88%				55.69%
	New				99.74%				57.52%

The new training shows

- **slight improvement** in the classification **purity** of track-like particles and of shower-like particles

Conclusions
And possible outlooks

Conclusions and next steps

- The results of the new training show a small improvement in track-like particle classification, and a slight decrease in efficiency in shower-like particles
- This work (requested by the ICARUS collaboration) is finalized, and I was able to follow all the stages from the comparison of the two datasets to the training and the validation of this training.
  - The work was actively reported in bi-weekly meeting of the ICARUS TPC Reconstruction working group, and the subsequent updates can be found here SBN-doc-37652-v1, SBN-doc-37571-v2, SBN-doc-37825-v1 and SBN-doc-37990-v1
- Among possible improvements of this study there are
  - The choice of a different metric for ranking the Cross Validation results
  - A new training with a selection of the 13 variables
  - A new training with greater statistics

Thank you for the attention!

New training of the track/shower BTD algorithm in Pandora

Mattia Sotgia, Alice Campani, Lea Di Noto (University of Genoa and INFN), Angela Fava (FNAL)
End term internship presentation
(Sept. 26th, 2024)

*msotgia@ge.infn.it, **acampani@ge.infn.it

Fermilab

SBN Program

Backup Slides

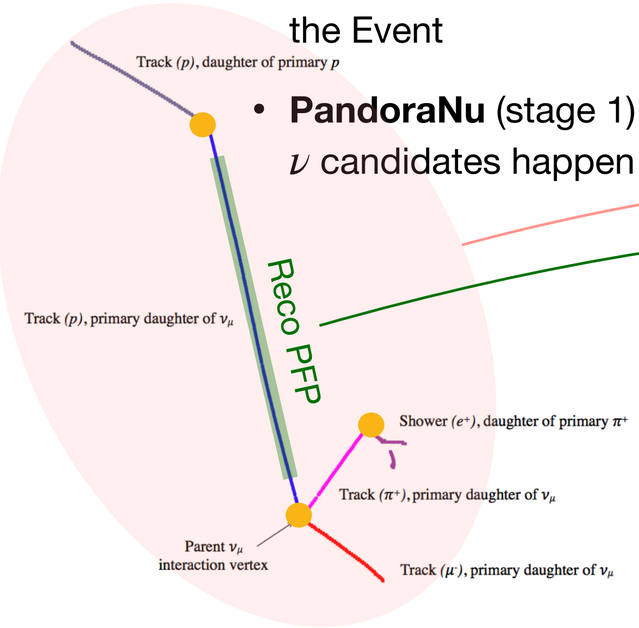
Event reconstruction in LArTPCs: ICARUS event reconstruction chain

ICARUS implements the Pandora-based reconstruction algorithm

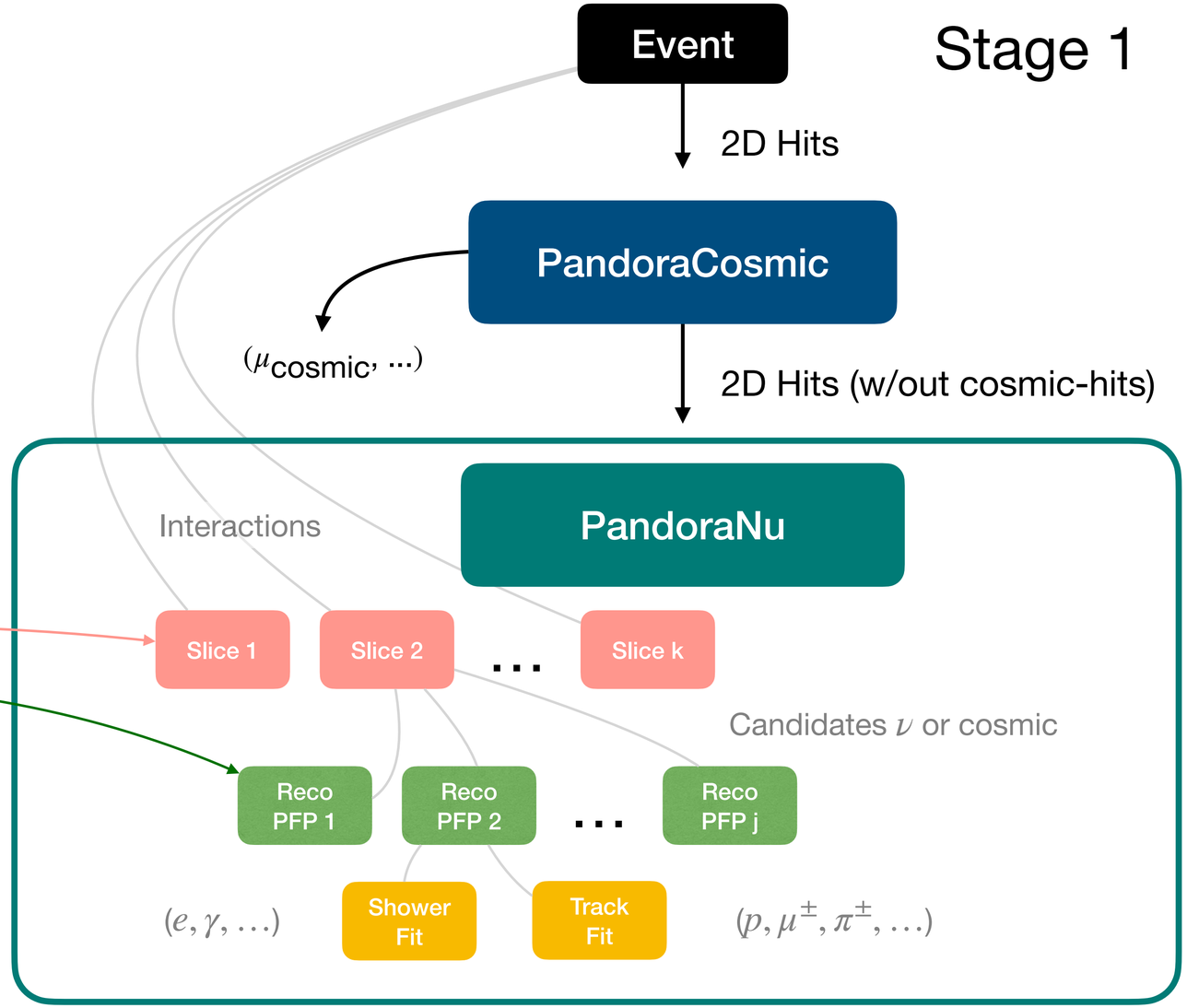
- Based on *clusters, slices* (reconstructed interactions, i.e. groups of particles linked with the same interaction) and pattern recognition

There are two main stages of the reconstruction

- **PandoraCosmic** (stage 0) where the cosmic-like hits are clustered from the 2D hits and are separated from the Event
- **PandoraNu** (stage 1) where the reconstruction of the ν candidates happen



Boosted Decision Trees (BDTs) are used **1.** in candidate ν/cosmic selection, **2.** in finding the true interaction vertex and **3.** in the track/shower discrimination



Definition of hit purity and completeness

Compare MC particles and reconstructed PFPs (Particle Flow Particles, Pandora Objects)

Definitions

Matched hits $\equiv \text{hits}_{\text{MC particle}} \cap \text{hits}_{\text{reco pfp}}$.

For the example on the side Matched $\text{hits}_j \rightarrow 6$ and Matched $\text{hits}_k \rightarrow 2$

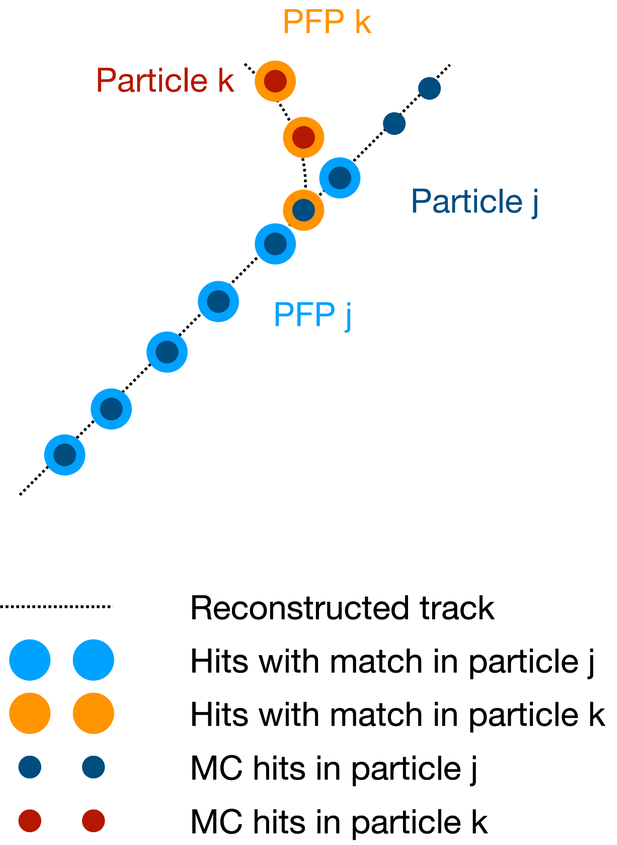
$$\textbf{Purity} \equiv \frac{\text{hits}_{\text{MC particle}} \cap \text{hits}_{\text{reco pfp}}}{\text{hits}_{\text{reco pfp}}}.$$

For the example on the side $\text{Purity}_j \rightarrow \frac{6}{9} \simeq 67\,\%$ and $\text{Purity}_k \rightarrow \frac{2}{9} \simeq 22\,\%$

$$\textbf{Completeness} \equiv \frac{\text{hits}_{\text{MC particle}} \cap \text{hits}_{\text{reco pfp}}}{\text{hits}_{\text{MC particle}}}.$$

For the example on the side $\text{Completeness}_j \rightarrow \frac{6}{9} \simeq 67\,\%$ and

$\text{Completeness}_k \rightarrow \frac{2}{2} \simeq 100\,\%$



Looking forward:
Next steps

- The ML MPVMPR sample has shown **greater discrimination power** in some BDT variables
- The ML MPVMPR sample has also a better balance between track-like and shower-like particles contribution
  - The track/shower ratio is lower than the BNB MC sample

    $$\text{ratio}_{\text{MPVMPR}} = \frac{\#\text{track-like}}{\#\text{shower-like}}\bigg|_{\text{MPVMPR}} \simeq \frac{9052}{6978} \simeq 1.3$$

    $$\text{instead of ratio}_{\text{BNB}} = \frac{\#\text{track-like}}{\#\text{shower-like}}\bigg|_{\text{BNB}} \simeq \frac{64972}{704} \simeq 92.3$$

  - The MPVMPR sample analyzed consist of 325535 events, which is overall lower in respect to the event count of the BNB MC
  - In the view of the training a new MC sample with larger statistic has been produced. The sample contains roughly 200 000 tracks and 150 000 showers per cryostat

BDT variables:
charge variables and cone charge variables

The current version of the BDT track/shower algorithm implements 13 variables (hyper parameters) to perform the cuts of the decision tree

All the BDT charge variables are computed on the Hits of the induction 1 wire plane. The other make use of the full 3D information from the reconstructed PFFs.

The first two are the ‘charge-based variables’.

1. **Charge end fraction (BDT.chendfrac)**, defined as the ratio of the deposited charge in the last 10% of the PFP hits, over the total deposited charge. Tracks are expected to have a more uniform charge distribution than showers. For this variable the expected values are in the range [0, 1]. Smaller values mean a less uniform charge distribution trough the length of the pfp.
2. **Charge fraction spread (BDT.chfracspread)**, defined as the ratio of the variance of the deposited charge of the hits to the deposited charge mean value. Showers are expected to have a more spread variety of charge related to the hits. It is a ratio but it is not normalized (i.e. the range is not in [0, 1]). The binning is chosen to be in [0, 2.5]. Tracks are expected in < 1, whereas showers are expected in > 1.

BDT variables:
charge variables and cone charge variables

The last update of the BDT algorithm introduced three new variables, called ‘cone charge variables’

Defining the chargeCore (the hits inside the 20% of the direction of the primary eigenvector) and chargeHalo (hits beyond the 20% threshold)

3. **Concentration (BDT.concentration)**, defined as $\frac{\text{chargeCon}}{\text{chargeCore} + \text{chargeHalo}}$

   Values are expected in the range [0, 100]

4. **Halo total ratio (BDT.halototratio)**, defined as $\frac{\text{chargeCore}}{\text{chargeCore} + \text{chargeHalo}}$, where chargeCon is the sum of the Hits inside the cone.
   It being a ratio, the values are expected in the range [0, 1].

5. **Conicalness (BDT.conicalness)**, defined as $\sqrt{\frac{\text{chargeConEnd}}{\text{chargeConStart}}} \Big/ \frac{\text{totalChargeEnd}}{\text{totalChargeStart}}$

   Its values are expected in the range [0, 600].

BDT variables:
linear and geometrical variables

There are also the linear variables

6. **Linear fit length (BDT.linfitlen)**, defined as the length of the reco particle. The long tracks ($\mu, \pi^{\pm}$) can be some ~1 m, protons are usually shorter and showers are smaller.
7. **Linear fit difference (BDT.linfitdiff)**, defined as the difference in linearity variation, between the end and the start point. This is expected to be quite small, in the range [0, 0.15] [arb. U.] both for showers and tracks.
8. **Linear fit gap length (BDT.linfitgaplen)**, defined as the gap between the hits on the linear fit. Tracks are expected to have smaller gap length. The common gap length is in the centimeters, so the range is [0, 0.5] cm.
9. **Linear fit RMS (BDT.linfitrms)**, defined as the RMS of the fit. Tracks are expected to have smaller RMS. The binning is in [0, 5], tracks are expected in [0, ~1], and showers are expected in [~1, ~5].

BDT variables:
linear and geometrical variables

And also the geometrical parameters

10. **Distance from vertex (to BDT.vtxdist)**, defined as the distance from the reconstructed vertex and its closest hit. This is usually very short for tracks and normally the distance of an electromagnetic shower from the reconstructed interaction vertex is greater. The range is chosen [0, 200] cm to account for events which were otherwise not included.

12. **PCA2 ratio (BDT.pca2ratio)**, defined as the ratio of the eigenvalue v_2 over the eigenvalue v_1 obtained from the Principal Component Analysis (PCA) algorithm, describing the orientation of the hits in space.
13. **PCA3 ratio (BDT.pca3ratio)**, defines as the ration of the third eigenvalue over the first. It is expected to be a good variable, being shower more tridimensional than tracks, since this value highlight the 3D aspect of the cluster, along with the PCA2 ratio. The chosen range is to get all the events plotted.
14. **Opening angle difference (BDT.openanglediff)**, defined as $\tan^{-1}\left(\sqrt{\text{PCA2}}\,\sin\theta\right)$, where θ is the angle between the two eigenvectors.
    The chosen range is in [0, ~35] deg, but most events are to be expected in the [0, 20] deg range.

The datasets:
MC fractional population of shower- and track-like particles

The simulations were made for the BNB dataset and for the MPVMPR dataset with the particle composition shown on the top right

Two datasets:

- **BNB MC** created for the study on Central Value (CV) systematics for Neutrino 2024
  icaruspro_production_v09_89_01_01_2024A_ICARUS_MC_CV_Sys_2024A_MC_CV_Sys_flatcaf

- **MPVMPR MC** samples, produced by the ICARUS ML WG
  acampani_training_caf_default_v09_89_01_01_mpvmpr

BNB sample is ν-only, whereas ML MPVMPR (Multi Particle Vertex, Multi Particle Rain) is ν + cosmic.

A cut (wellRecoCut) is applied to only select **well reconstructed** particles, which have *hit completeness* and *purity* above 80% (more of their definition is in backup). This avoids biasing the result of the comparison with other mis-reconstruction effects, such as clustering issues, track-splitting, …

Shower-like

Track-like

noSpillCut

	BNB (total 213512)		MPVMPR (total 66331)	
	Fraction	(events)	Fraction	(events)
protons	0,527	(112594)	0,429	(28455)
charged_pi	0,121	(25925)	0,223	(14822)
muons	0,237	(50707)	0,018	(1221)
electrons	0,003	(610)	0,092	(6074)
photons	0,111	(23676)	0,238	(15759)

wellRecoCut

	BNB (total 65676)		MPVMPR (total 16030)	
	Fraction	(events)	Fraction	(events)
protons	0,391	(25704)	0,377	(6050)
charged_pi	0,073	(4807)	0,153	(2450)
muons	0,525	(34461)	0,034	(552)
electrons	0,005	(313)	0,119	(1911)
photons	0,006	(391)	0,316	(5067)

📚 SBN-doc-34318-v2

📚 Neutrino 2024

📚 ML sample update SBN-doc-35469-v1

Generation of the ML MPVMPR data sample
ML MPVMPR Working Group

The MC sample data is generated with the Multi Particle Vertex Multi Particle Rain module in sbncode/EventGenerator/Multipart/gen_mpvmpr.fcl

The data is in the samweb definition icaruspro_production_2024A_MPVMPR_MC_v09_89_01_01_stage1

It is generated in three steps

1. One **Multi-Particle Vertex** is generated, random number (with a flat distribution) of particles sampled from a uniform energy distribution
   - The energy range is taken from the expected energies in the BNB
   - The beam spill is set similar, but slightly longer than NuMI, so MPV are generated in the [0, 10] μs range (NuMI is 9.5 μs, whereas BNB is 1.6 μs)
2. A random number (flat distribution in [3, 5]) of single particles sampled from different energy distribution is generated (**rain2**), covering the kind of cosmic we could see.
   - Generated in time (during the beam spill)
   - Generated in a larger volume than the TPC fiducial (+20 cm each direction)
3. A random number (flat distribution in [2, 4]) of single particles sampled from different energy distribution is generated (**rain**), covering the kind of cosmic we could see.
   - Generated out of time (not during the beam spill)
   - Generated in a smaller volume than the TPC fiducial (-20 cm each direction)

Further details in ML sample SBN-doc-35469-v1