Mathematical Modeling and Computational Physics
24 October 2024

Machine Learning for Particle Identification at MPD

V. Papoyan[1,3]

Coauthors: A. Aparin[2], A. Ayriyan[1,3], H. Grigorian[1,3], A. Korobitsin[2]

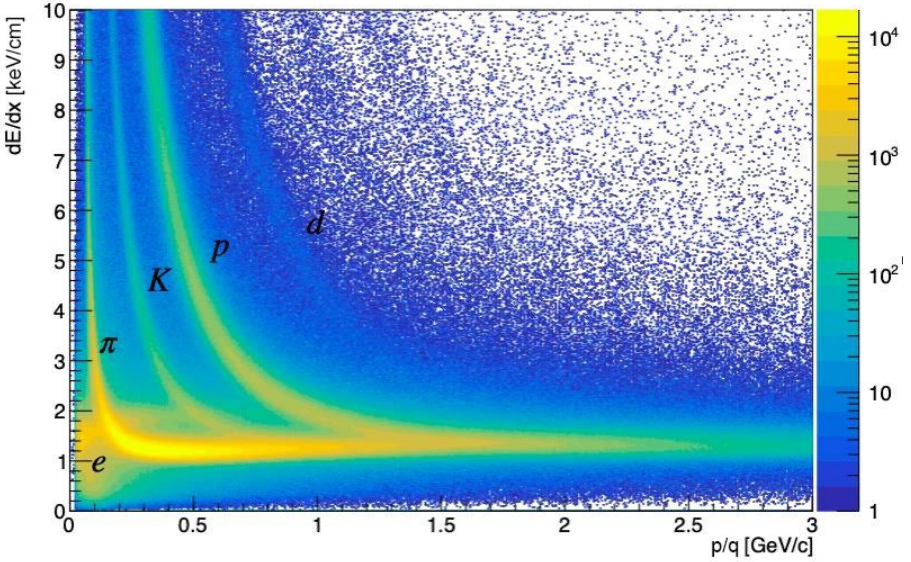
[1]MLIT JINR, [2]VBLHEP JINR, [3]AANL (YerPhi)

This work was done with support from the Russian Science Foundation under Grant No. 22-72-10028

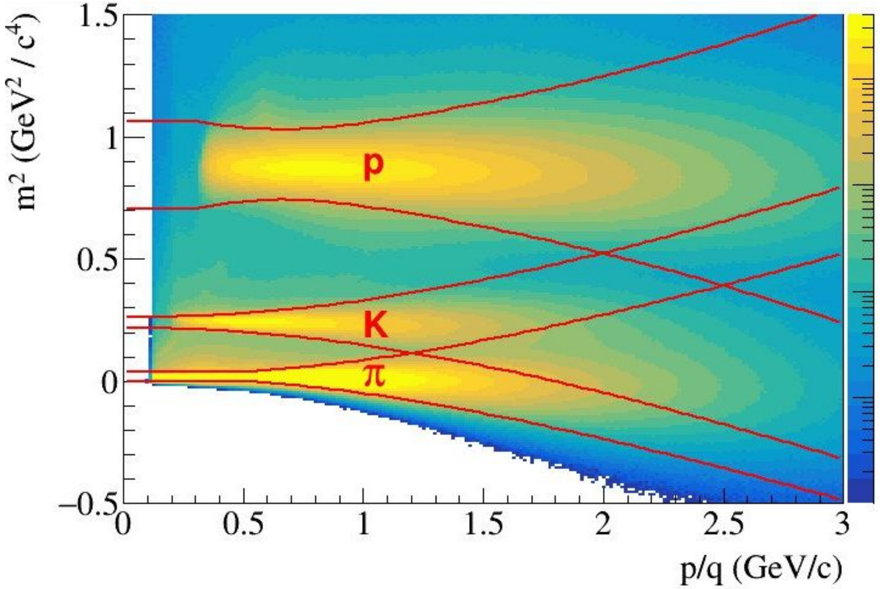
Particle Identification at MPD experiment

MPD particle identification (PID) is based on **Time-Projection Chamber** (TPC) and **Time-of-Flight** (TOF).

A TPC can identify charged particles by measuring their specific ionization **energy losses** (dE/dx);



A TOF measures the particle flight **time** over a given **distance** along the track trajectory;



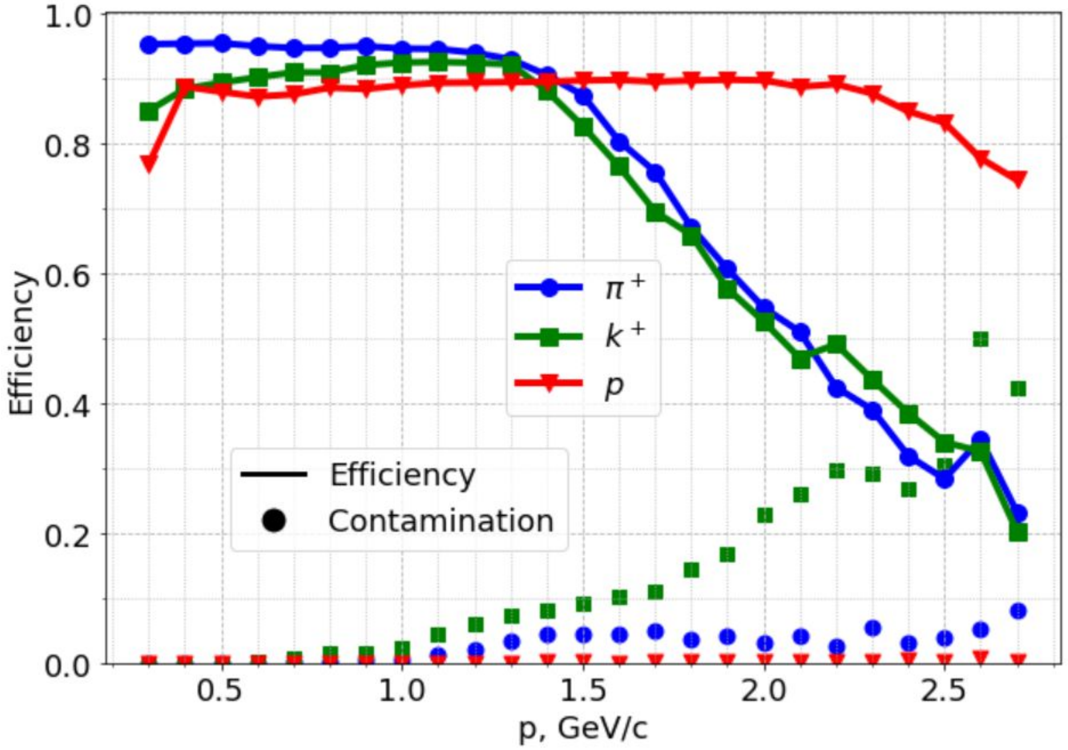
Knowing the particle **momentum** (from TPC) one obtains the **mass squared** and thus identity of the particle.

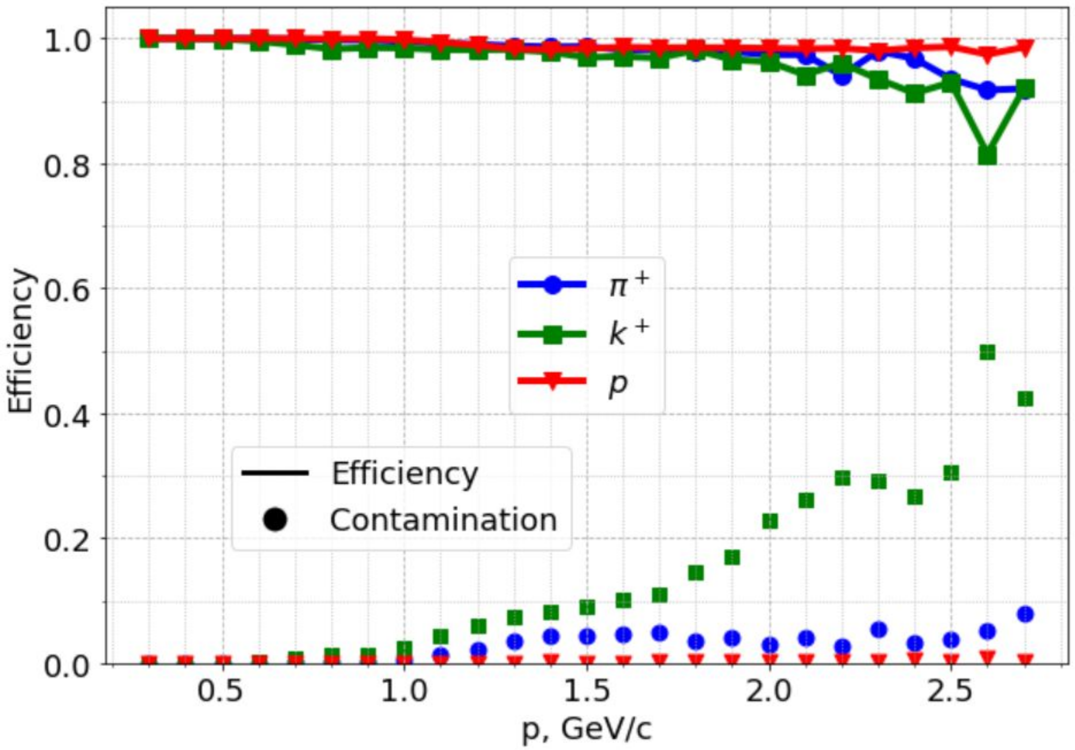
Klempt W. Review of particle identification by time of flight techniques

Baseline PID at MPD - N-sigma

There are two ways of calculating PID efficiency. The difference is the number of tracks in the denominator

$$E^S = \frac{N^S{}_{corr}}{N^S{}_{true}} \qquad C^S = \frac{N^S{}_{incorr}}{N^S{}_{corr}+N^S{}_{incorr}}$$





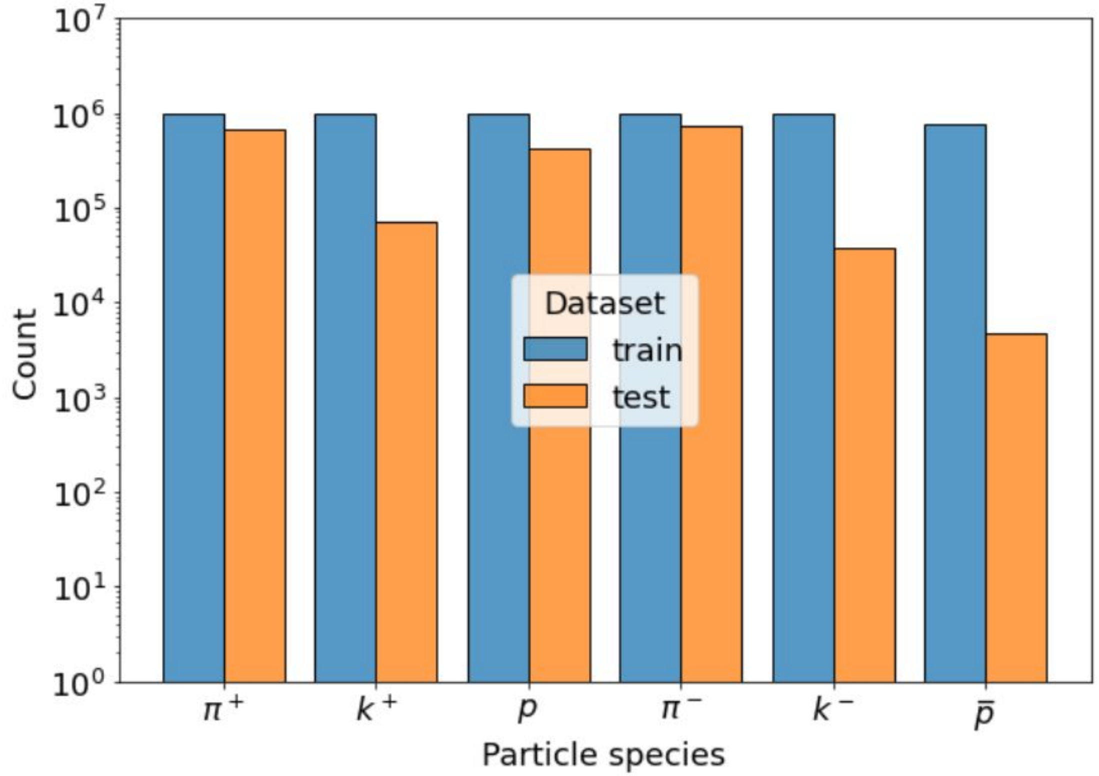
PID efficiency and contamination for all tracks (left) and only identified tracks (right)

in Bi+Bi collisions at 9.2 GeV

Training and Test data

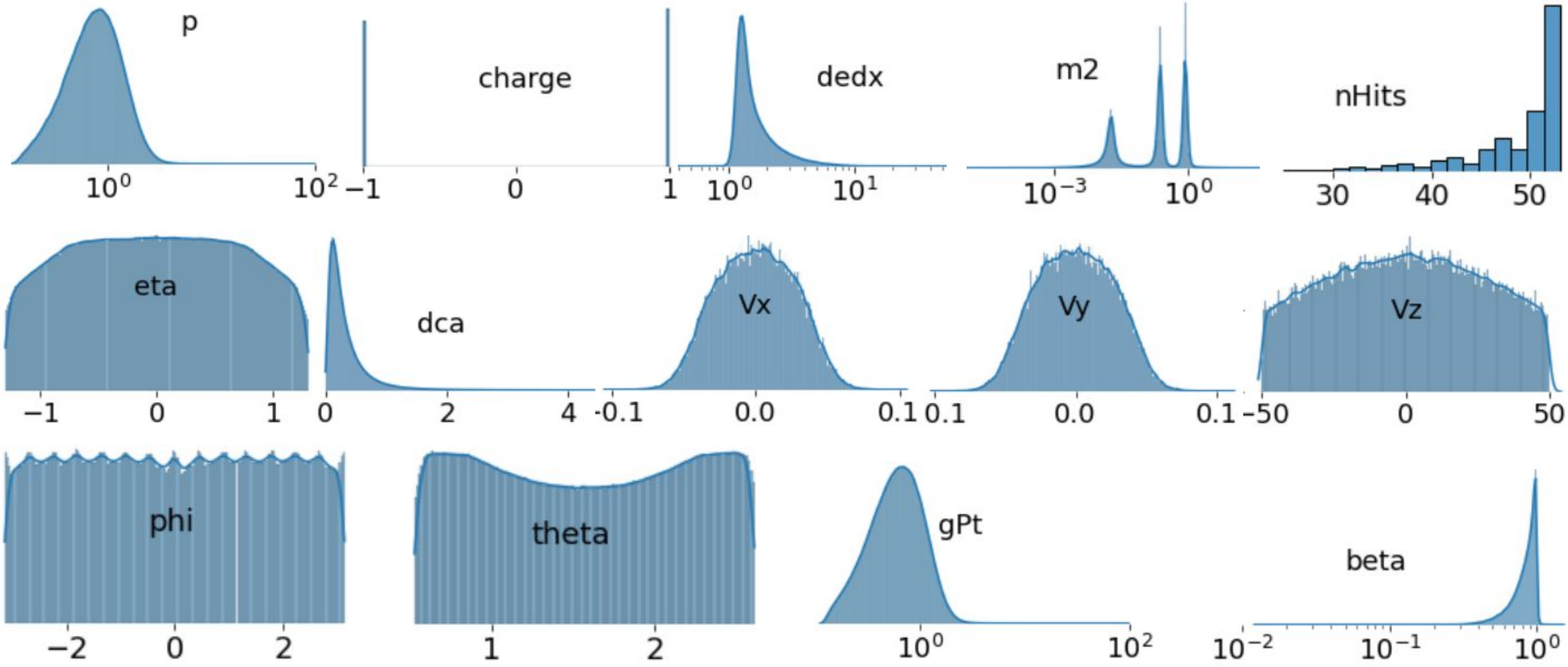
Subsample of the MPD Monte-Carlo production (Request 25) was used for training XGBoost model.

Event generator	UrQMD
Transport	Geant 4
Impact parameter ranges	0-16 fm (mb)
Smear Vertex XY	0.1 cm
Smear Vertex Z	50 cm
Colliding system	Bi+Bi
Energy	9.2 GeV

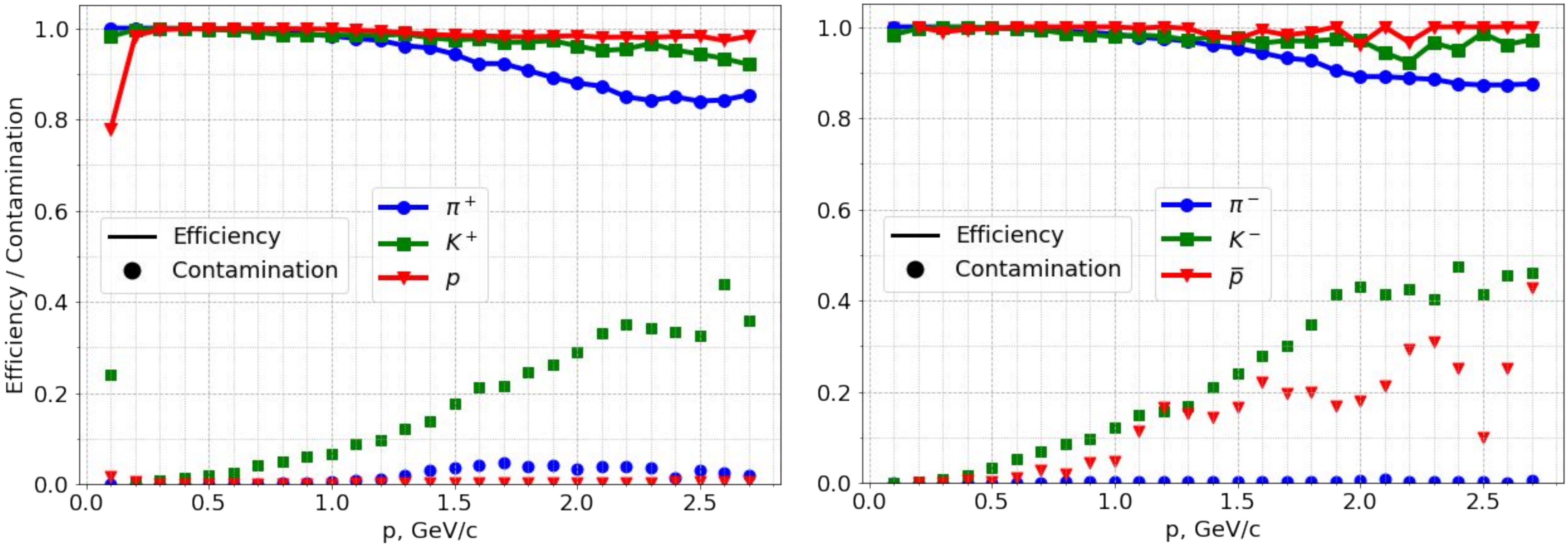


track selection criteria: (p < 100) & ($|m^2|$ < 100) & (nHits > 15) & (|eta|<1.5) & (dca < 5) & (|Vz| < 100)

Input data description

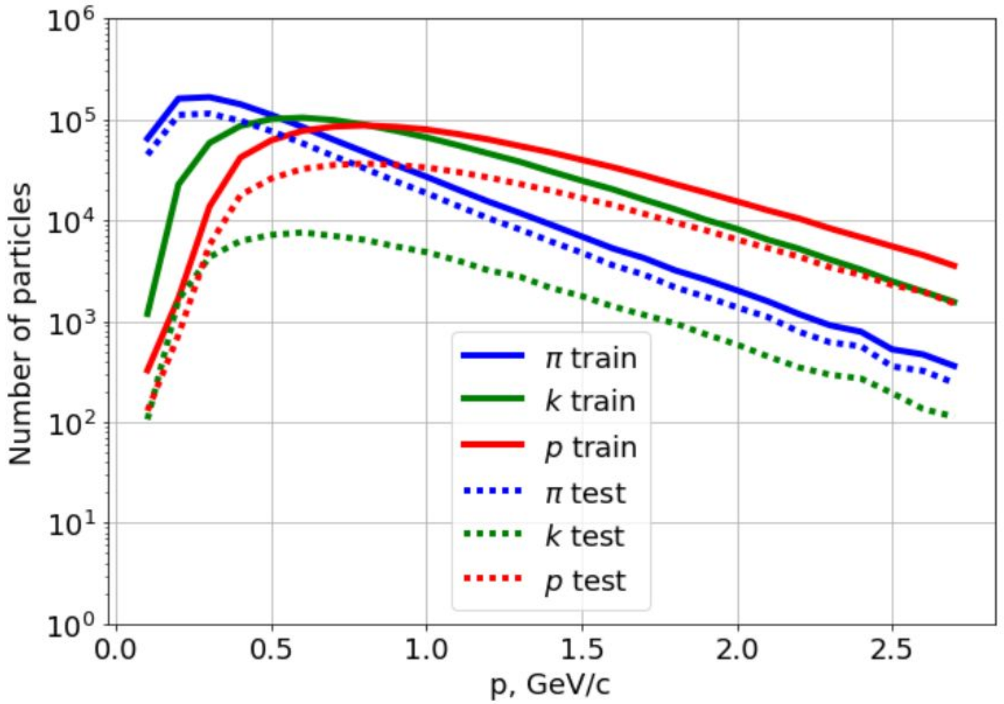


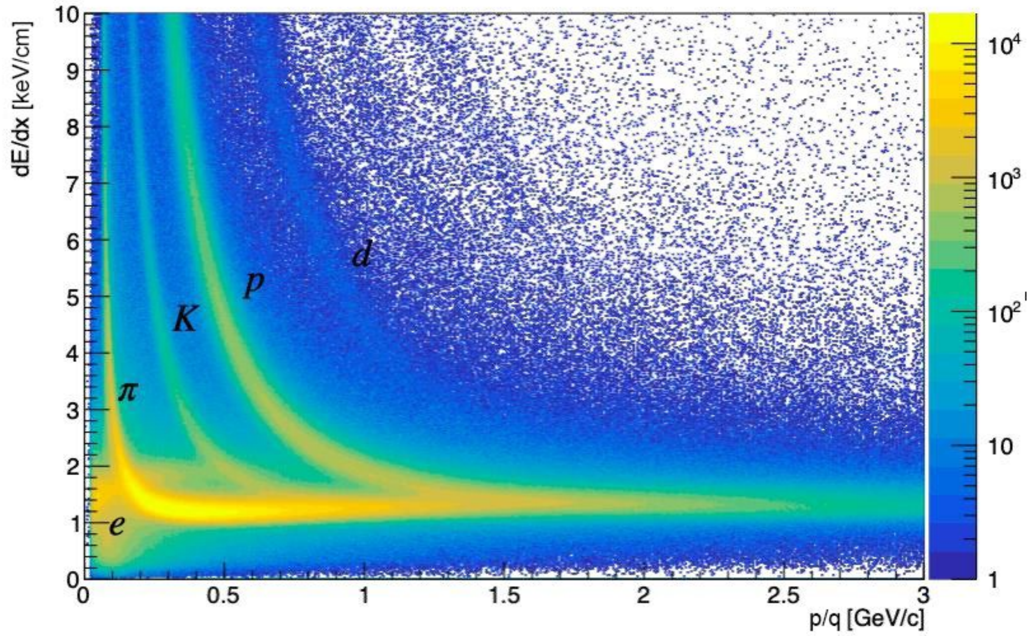
Test XGBoost classifier on Request 25 subsample



Efficiency and contamination of XGBoost

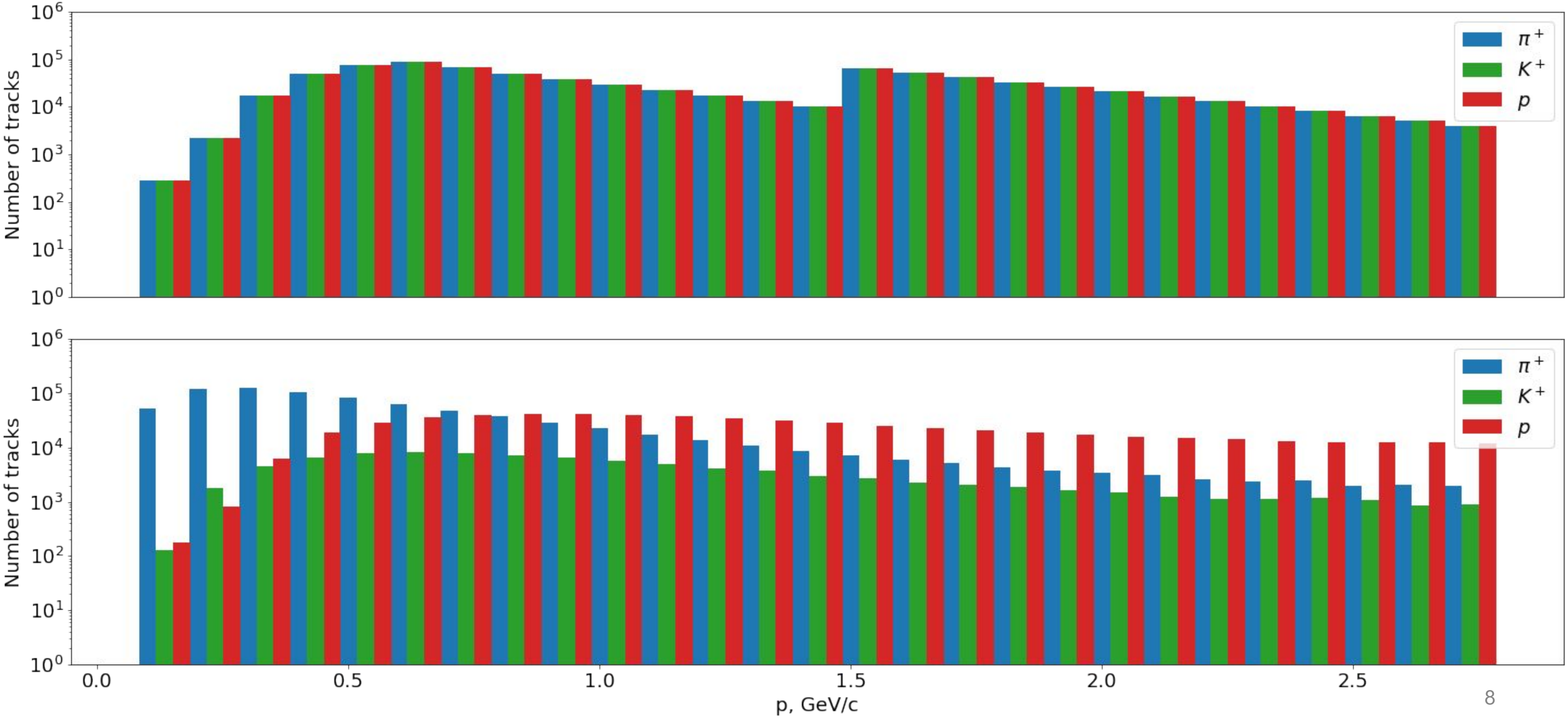
Class distribution and features informativeness





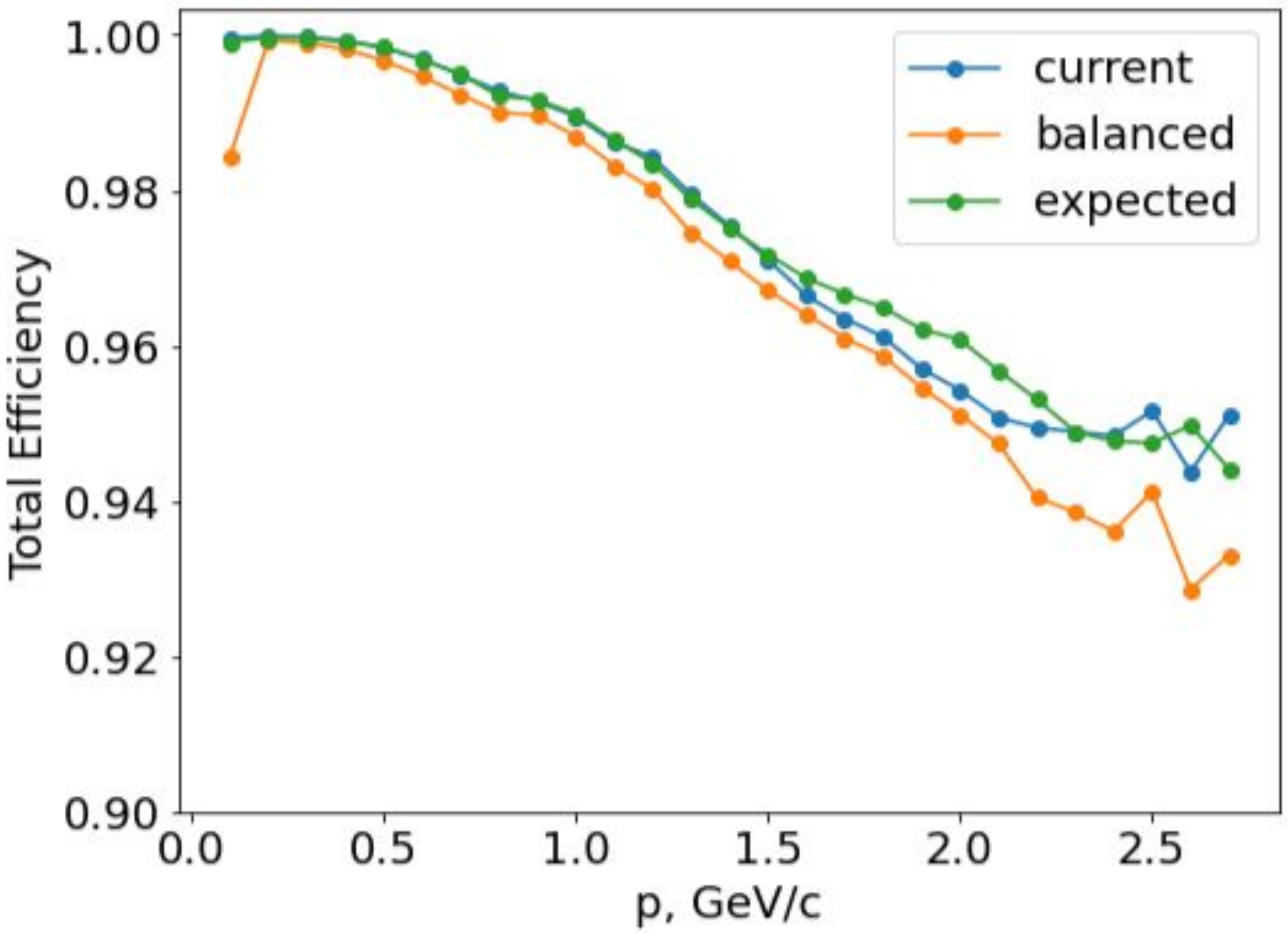
- In case of high momentum values amount of useful information (dEdx and m^2) is not a sufficient to make robust predictions.
- In the absence of informativeness features, the model may use statistics to predict particle type.
- But particle distribution did not correspond to expected distribution in training data (number of Kaons is higher than number of Pions).

Balanced class distribution vs Expected

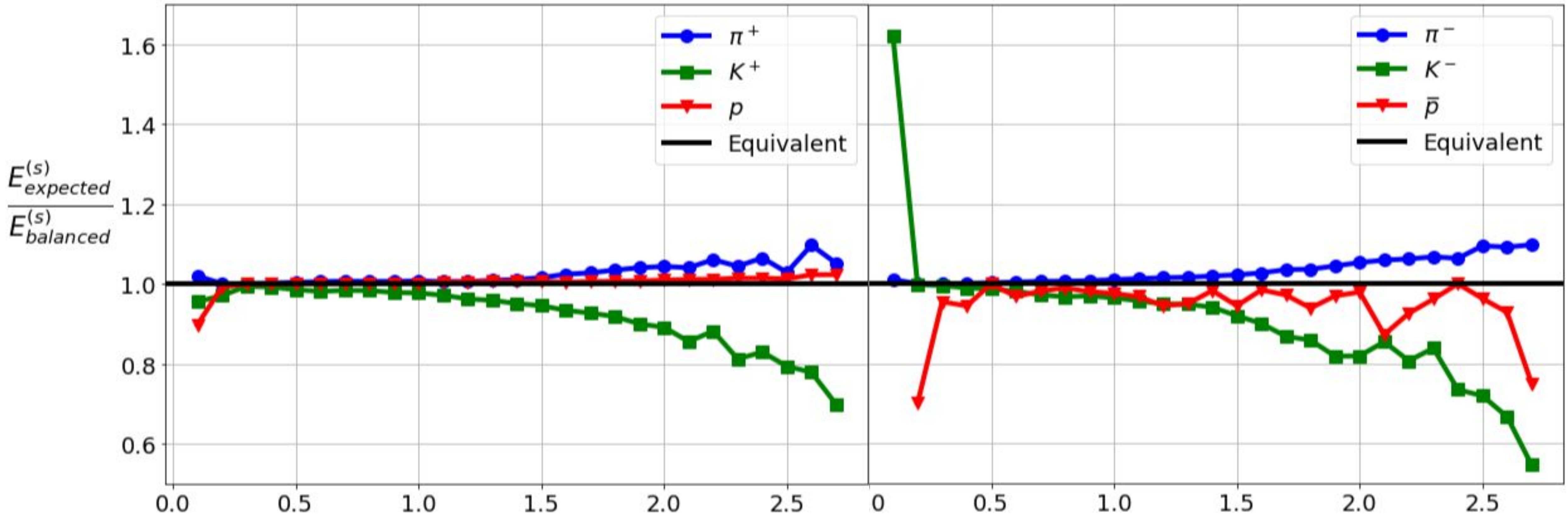


Balanced class distribution vs Expected

$$E_{total} = \frac{N_{corr}}{N_{true}}$$



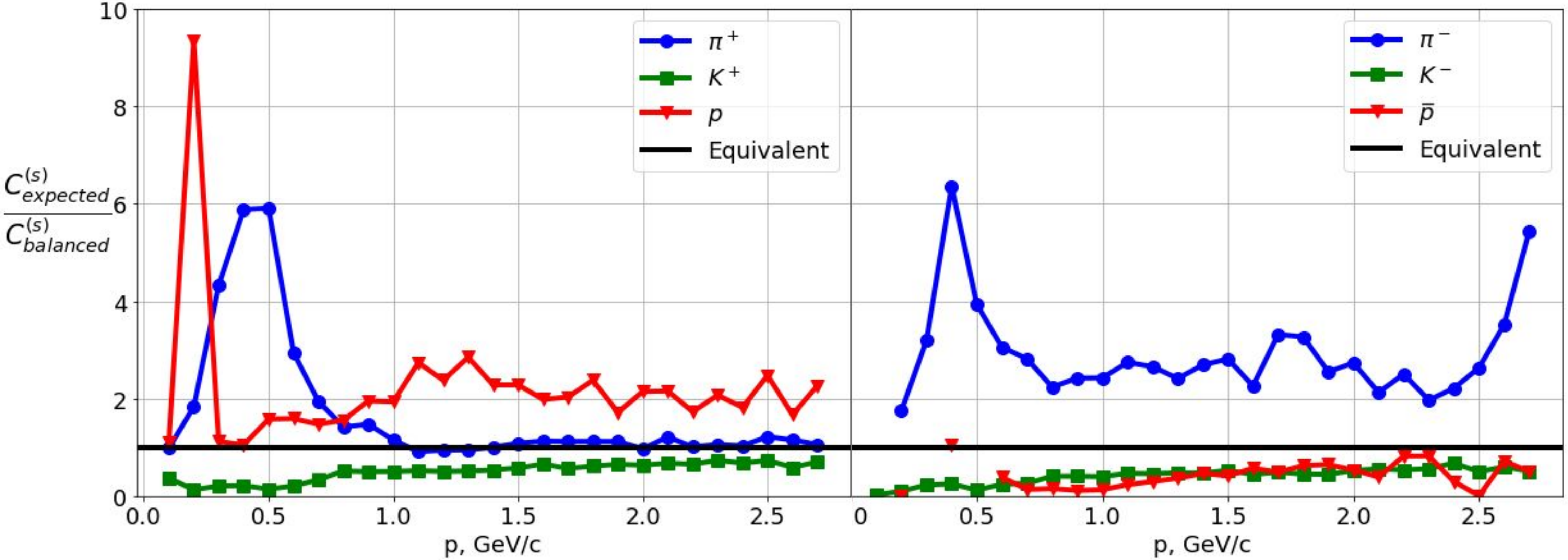
Balanced class distribution vs Expected



Efficiency ratio of XGBoost with expected distribution and balance

Balanced class distribution vs Expected

Despite the fact that expected particle distribution allowed for improve PID efficiency, contamination became worse. There is a trade-off between efficiency and contamination, and a balance should be found in the future.



Contamination ratio of XGBoost with expected distribution and balance

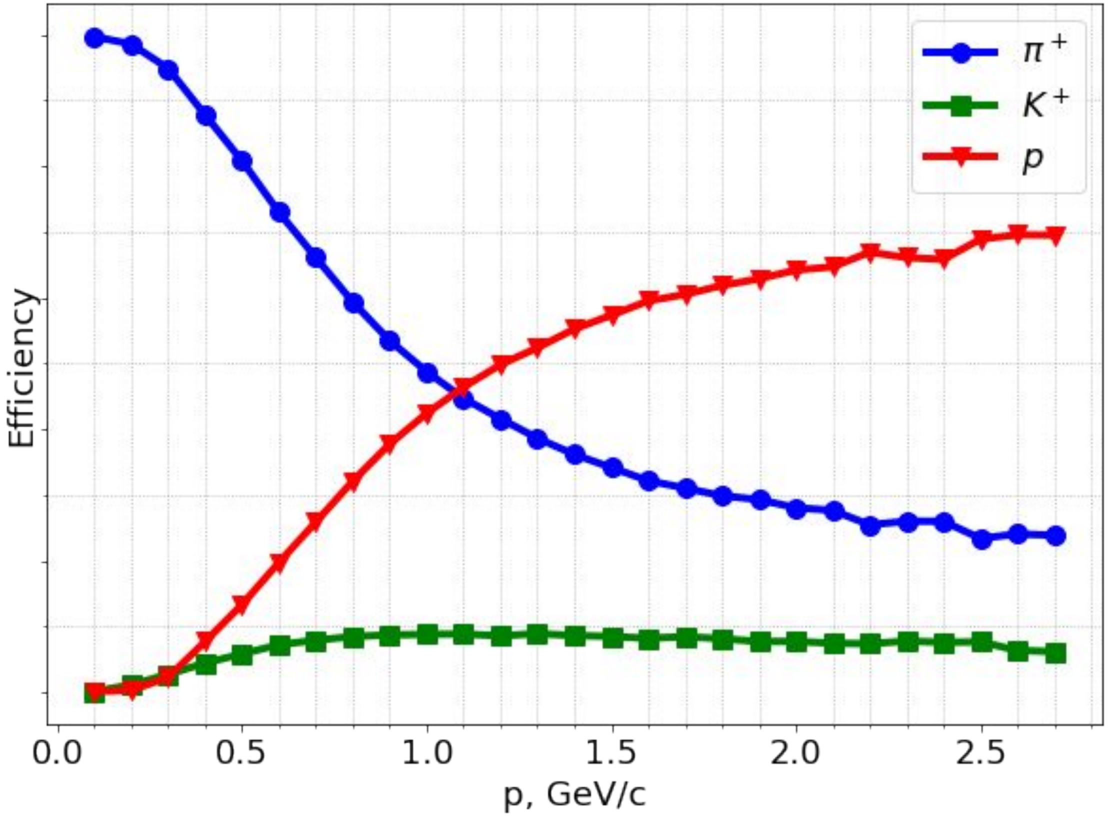
ML vs Blind method

For a given momentum range

3 classes: Pion-Kaon-Proton

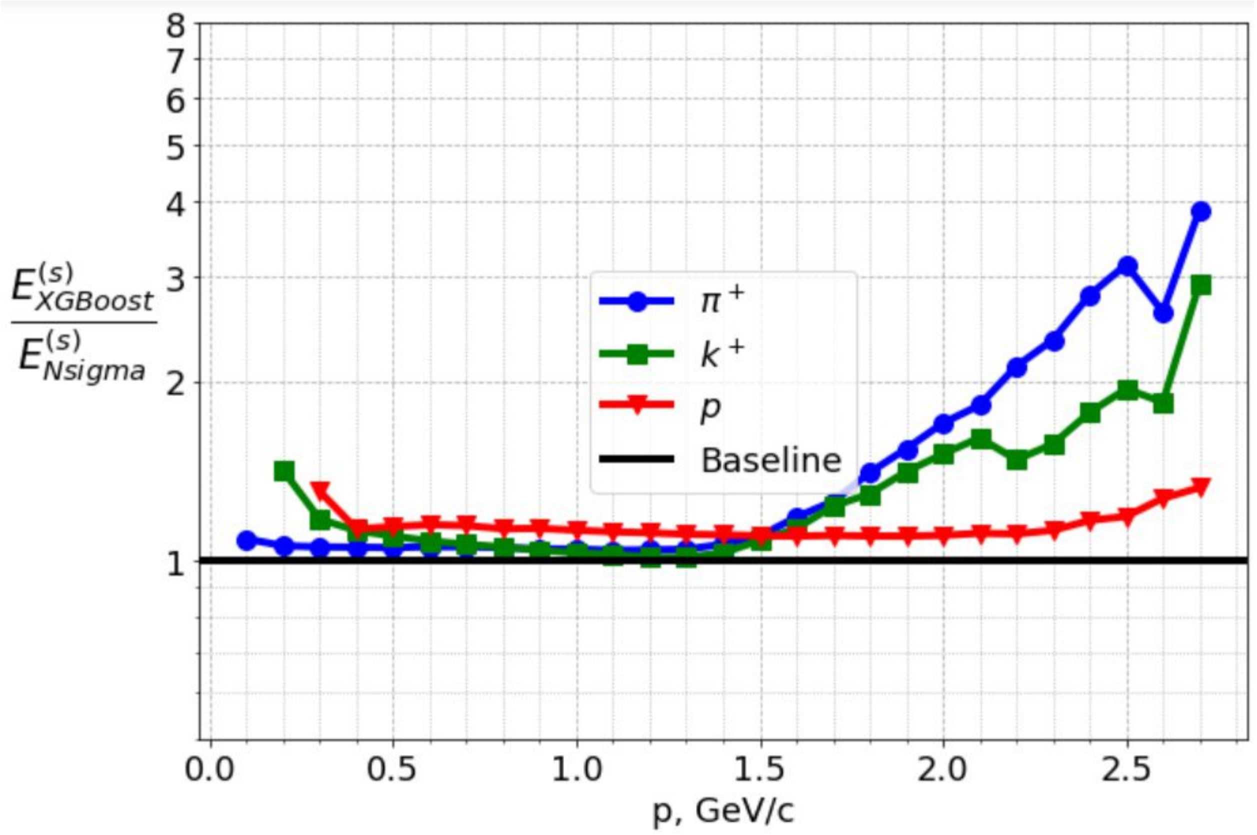
Frequencies: **60 - 30 - 10**

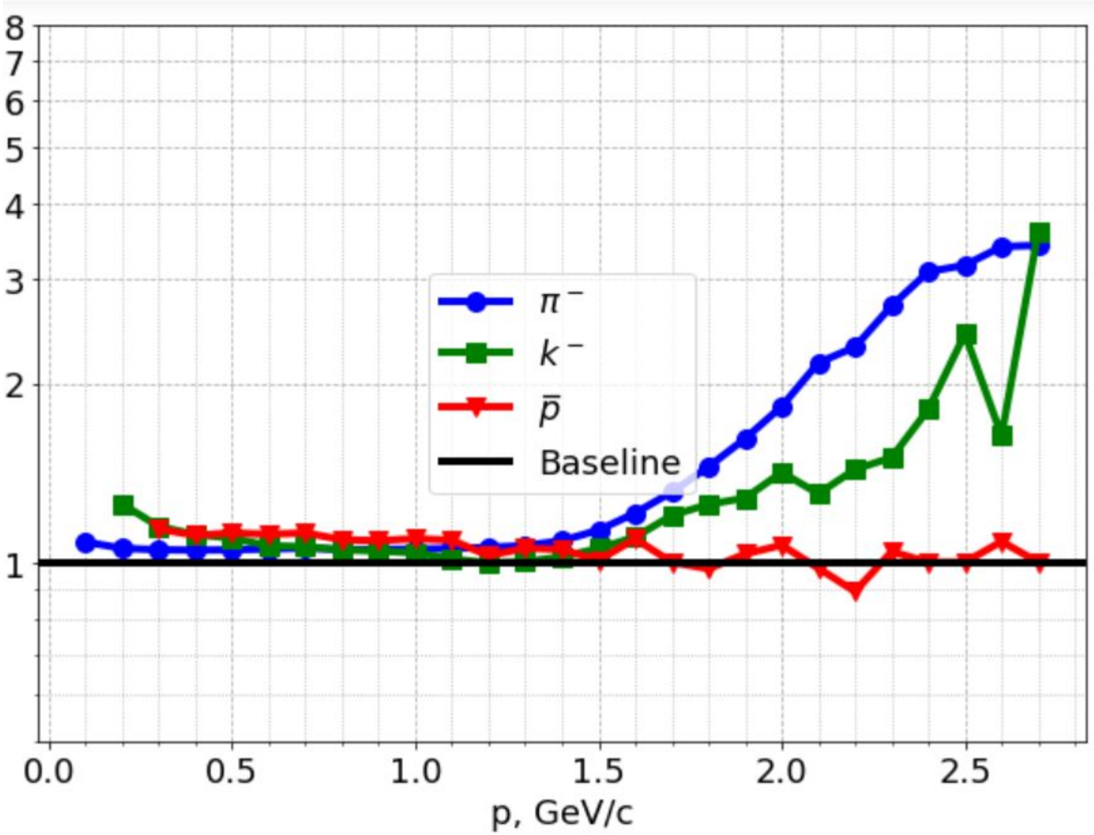
ksi is a uniformly distributed random number




```
for pi (i_th particle):
    ksi = rand[0, 1]
    if ksi in [0, 0.6] pi is Pion
    else if ksi in [0.6, 0.9] pi is Kaon
    else pi is Proton
```

XGBoost vs N-sigma on Request 25 subsample

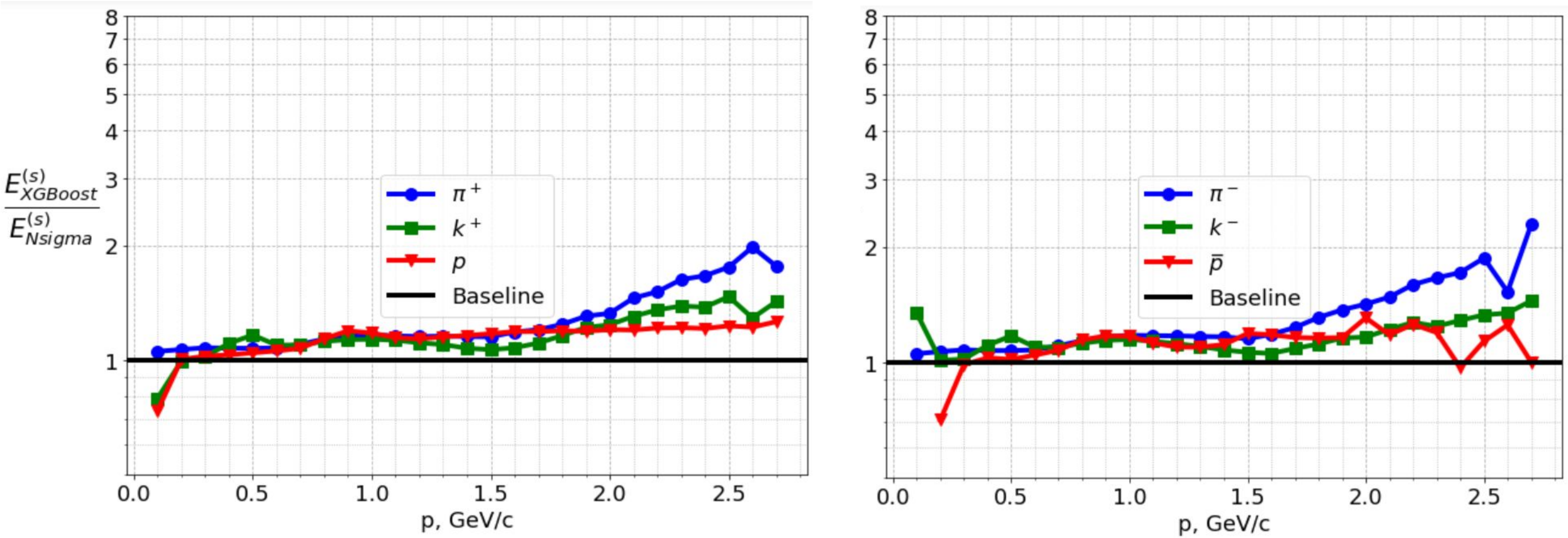




Efficiency ratio of XGBoost and n-sigma method

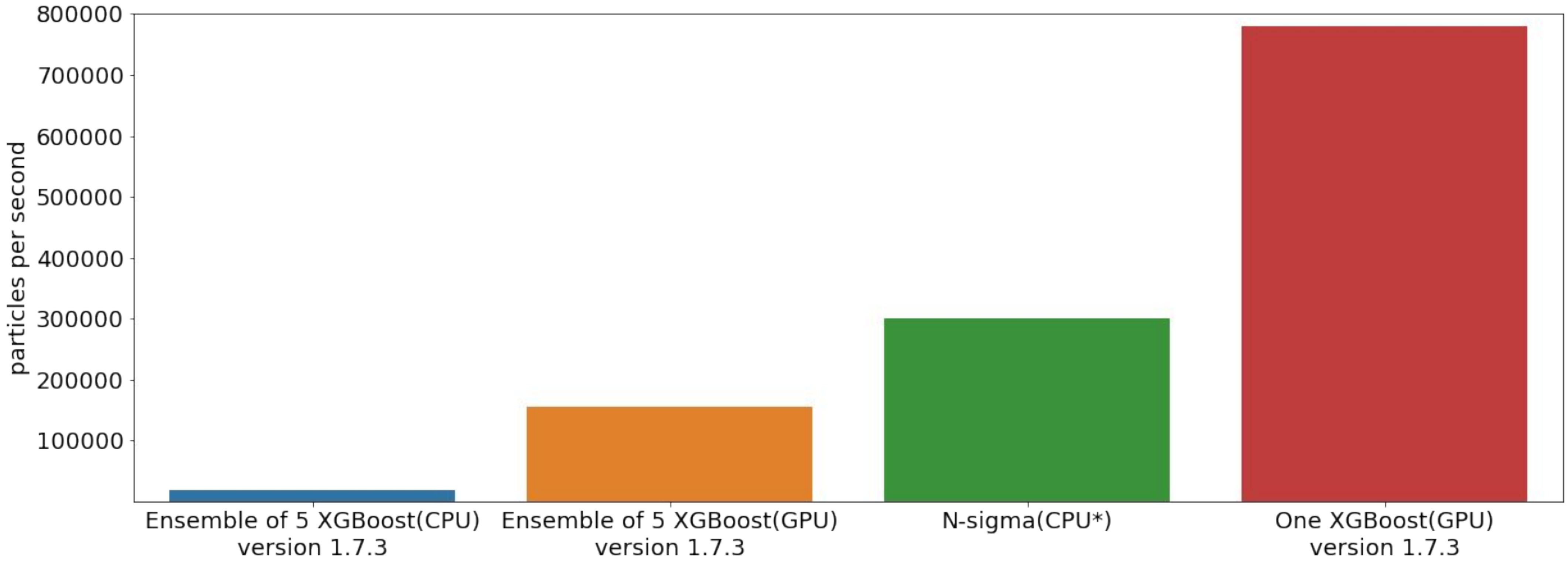
XGBoost vs N-sigma

Impact parameter	0-14 fm
Energy	5.5 GeV



Efficiency ratio of XGBoost and n-sigma method

Inference time of the algorithms



GPU: Nvidia Tesla V100-SXM2 NVLink 32GB HBM2

CPU: Intel Xeon Gold 6148 CPU @ 2.40 GHz 20 Cores / 40 Threads

CPU*: Intel® Core™ i7-8700 CPU @ 3.20GHz × 12

Conclusion and Outlook

1. The distribution of particles in the training dataset plays a crucial role in the performance of gradient boosting models.
2. There is some useful information in spite of features overlapping in high momentum region. It was taken into account by ML classifier.

Next we are going to:

- do additional testing to characterize identification stability of the classifier on data produced with different initial parameters of generated MC tracks at the MPD;
- investigate the ways of recognizing and addressing the problem of distribution shift to avoid decline of classifier performance.

Backup

Classification of Charged Particles

In Machine Learning terms PID can be considered as **classification** task (**Supervised** learning).

Let

X - is the input space (particle characteristics such as: dE/dx, m^2, β, q, etc)

Y - is the output space (particle species such as: π, k, p, etc)

Unknown mapping exists

$$\mathbf{m : X \rightarrow Y},$$

for values which known only on objects from the finite training set

$$\mathbf{X^n = (x_1, y_1), \ldots, (x_n, y_n)},$$

Goal is to find an algorithm **a** that classifies an arbitrary new object $\mathbf{x \in X}$

$$\mathbf{a : X \rightarrow Y}.$$

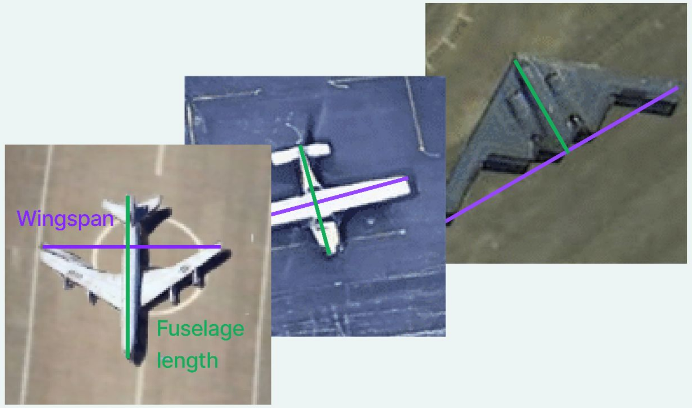
Formulas

$$m^2 = \frac{p^2}{c^2}\left[\frac{t^2c^2}{L^2} - 1\right] \qquad \beta = \frac{L}{ct}$$

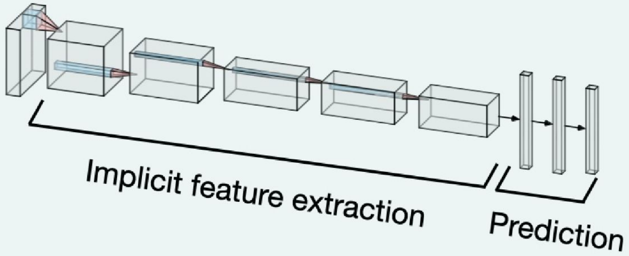
$$-\left(\frac{dT}{dx}\right) = \frac{4\pi n_e z^2 e^4}{m_e v^2}\left[\ln\frac{2m_e v^2}{I} - \ln(1-\beta^2) - \beta^2 - \delta - U\right],$$

Tabular Data: Deep Learning vs Gradient Boosting

Unstructured data



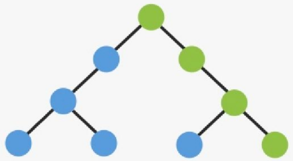
Deep learning



Structured data

	Fuselage length	Wingspan
Boeing 707	44,07	39,9
Cessna 172	8,28	11
B-2 Spirit	20,90	52,12

Gradient Boosting



https://sebastianraschka.com/blog/2022/deep-learning-for-tabular-data.html

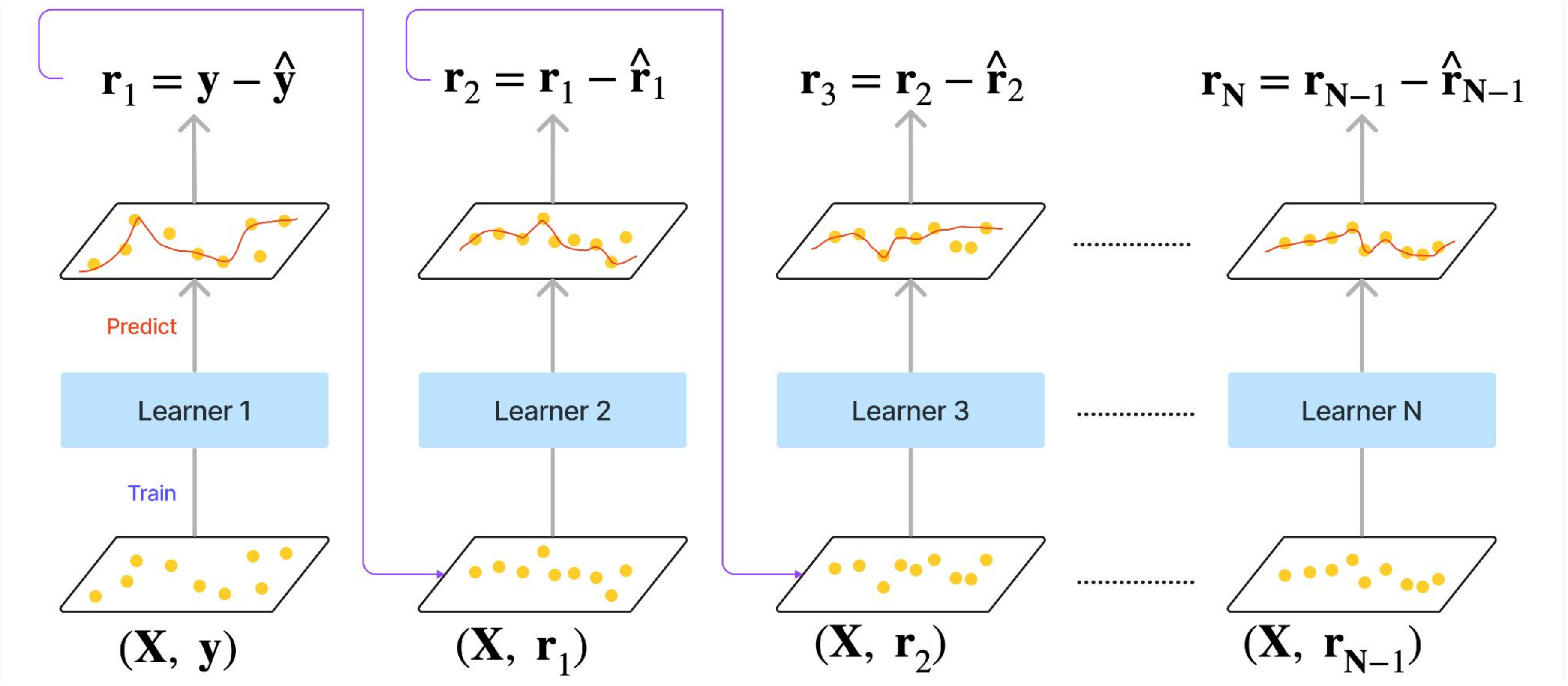
Data description

feature	values range
p	(0.1, 100)
q	{-1, 1}
dedx	(0, 72)
m2	(-100, 100)
nHits	[20, 53]
eta	[-1.3, 1.3]
dca	(0, 5)

feature	values range
Vx	(-0.106, 0.106)
Vy	(-0.103, 0.112)
Vz	(-50, 54.1)
phi	(-3.1415, 3.1415)
theta	(0.53, 2.61)
gPt	(0.106, 98)
beta	[0.012, 1.564]

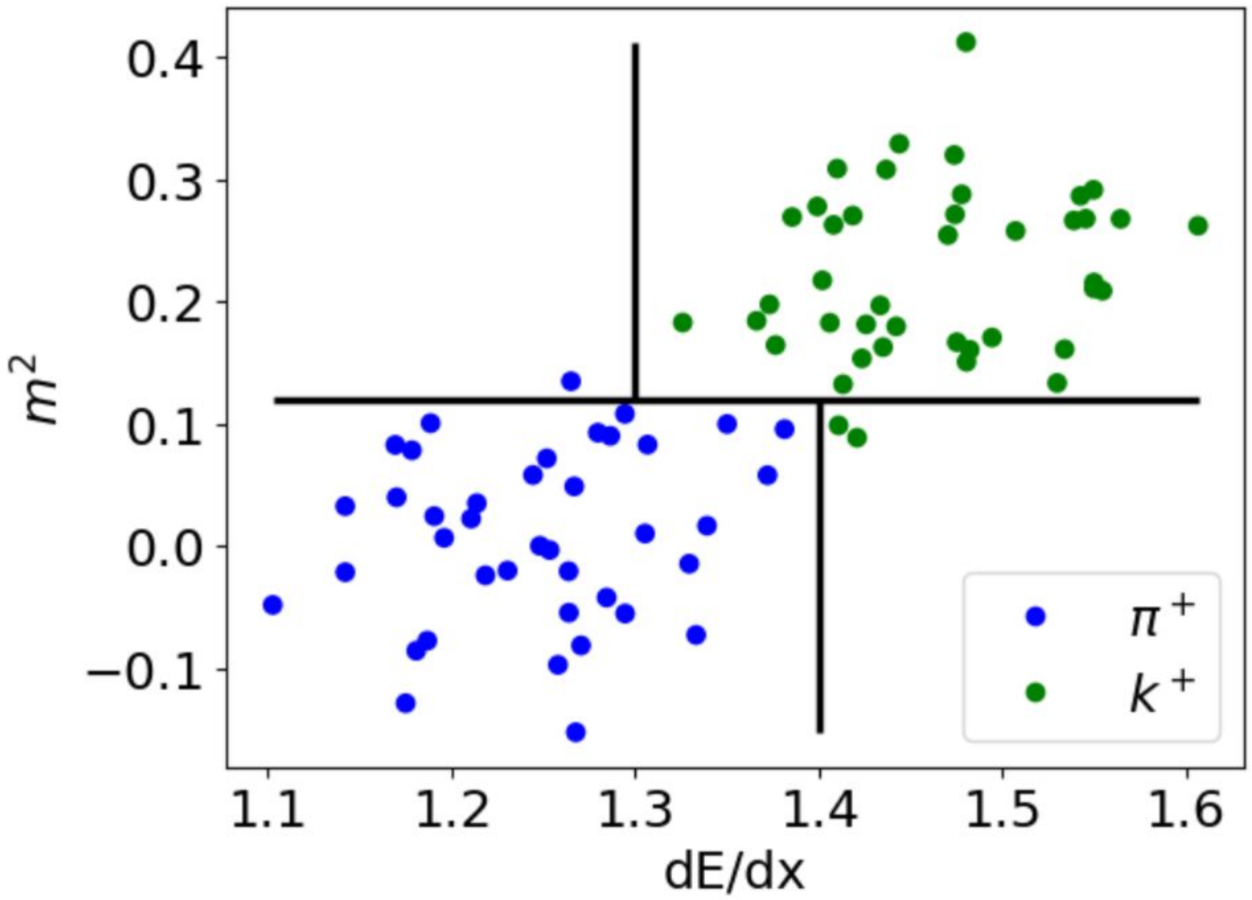
Gradient Boosting

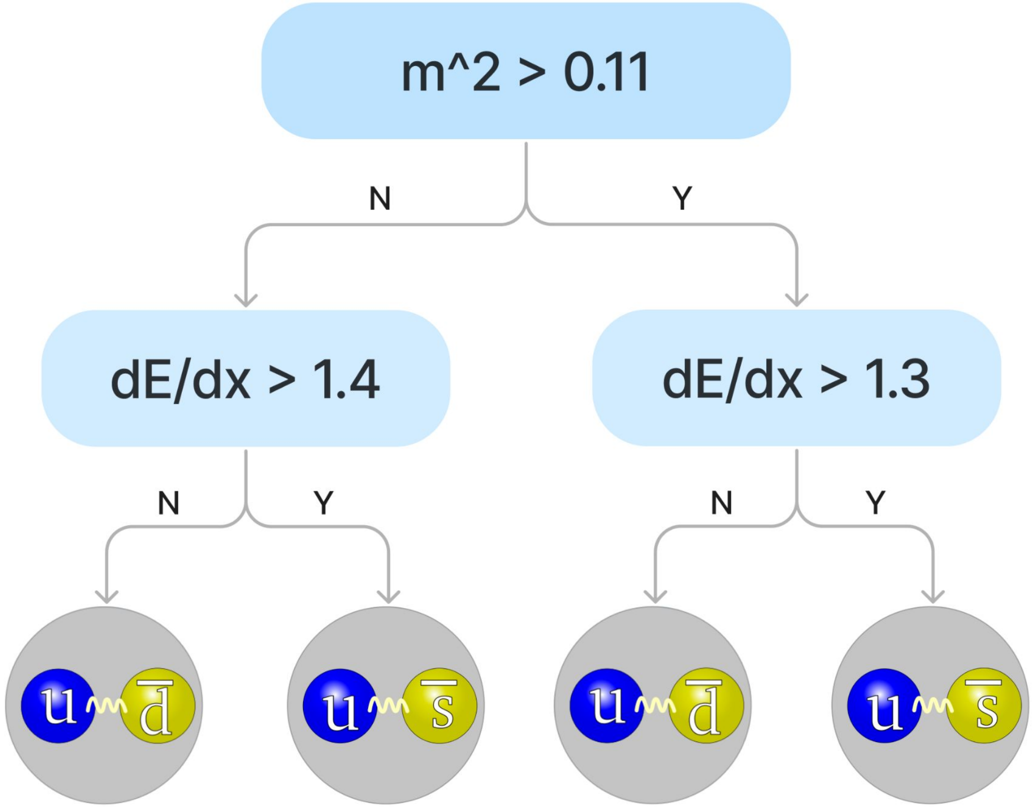
Gradient boosting is a machine learning technique which combines weak learners into a single strong learner in an iterative fashion



Gradient Boosted Decision Tree

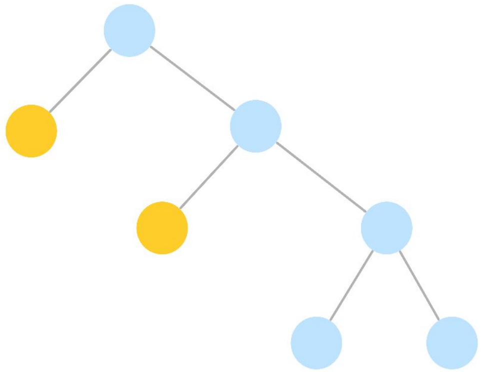
Gradient Boosted Decision Tree (GBDT) uses decision trees as weak learner. They can be considered as automated multilevel **cut-based** analysis



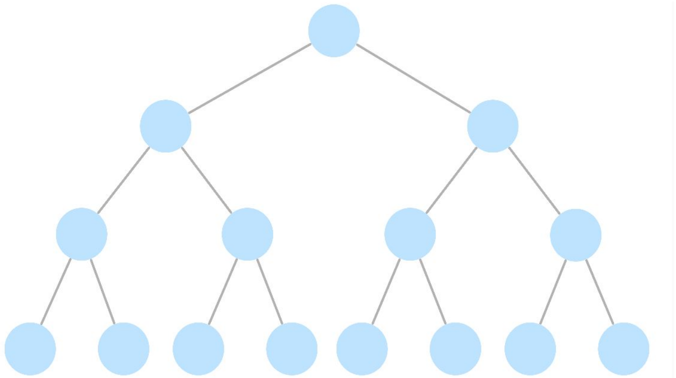


XGBoost vs LightGBM vs CatBoost vs SketchBoost

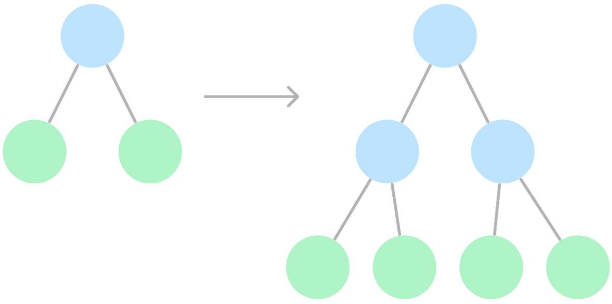
Asymmetric Tree (XGB, LGBM)



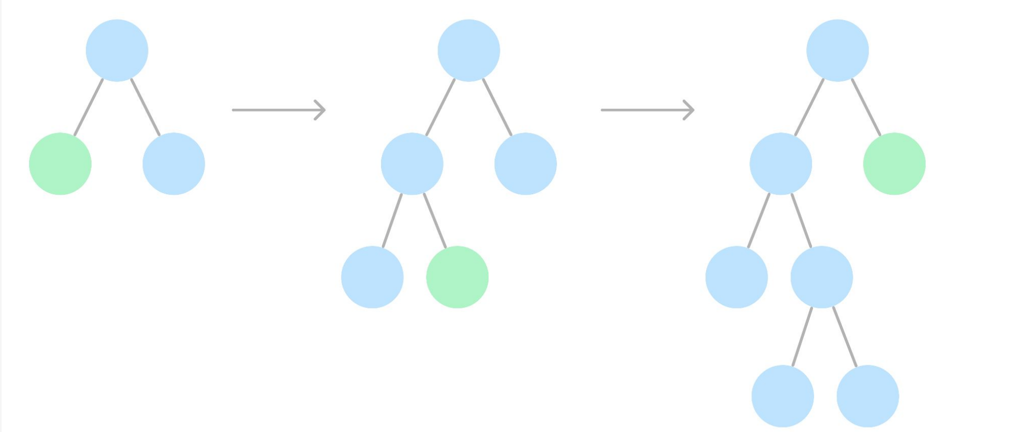
Symmetric Tree (CatBoost, SketchBoost)



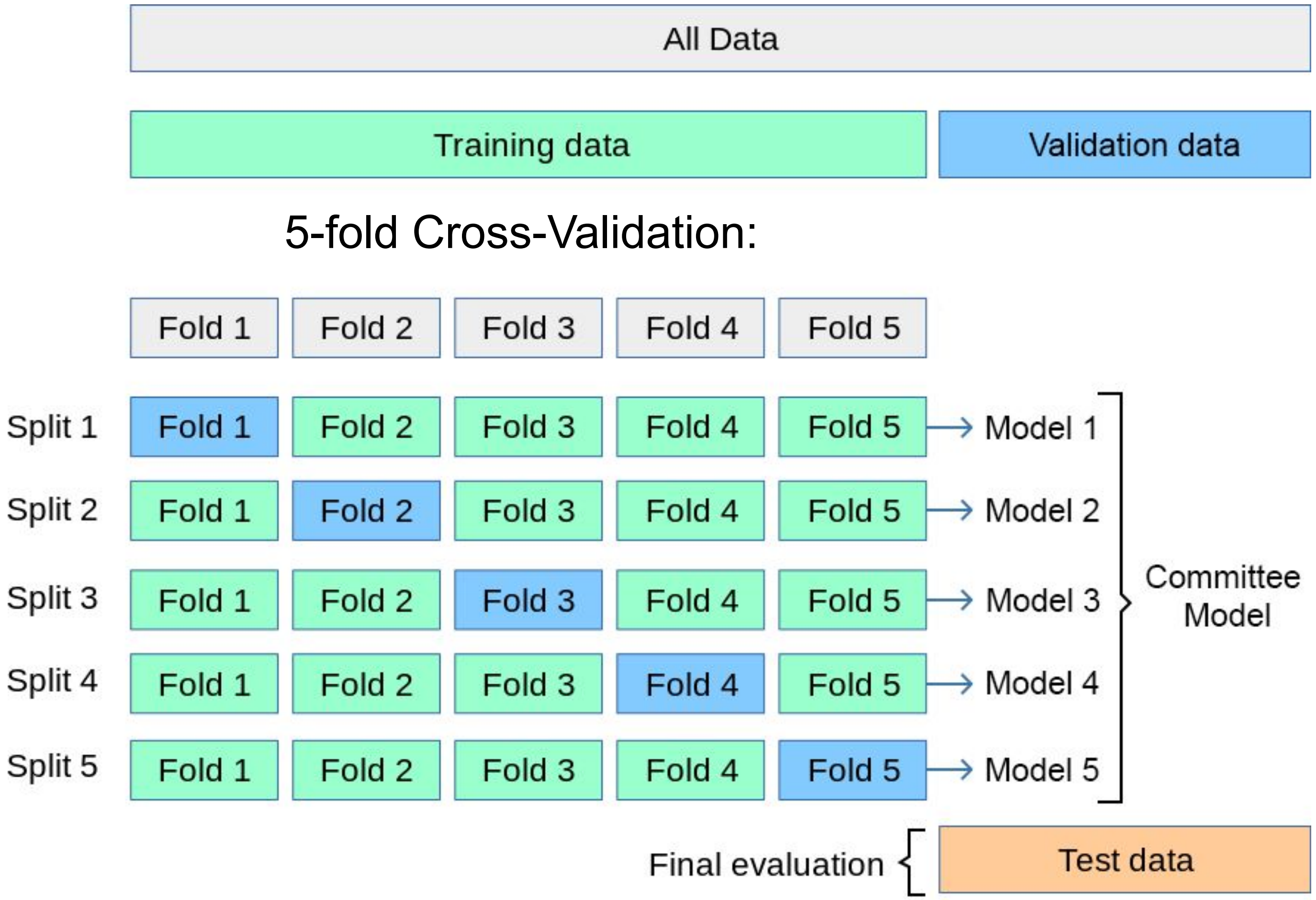
Level-wise Tree Growth (XGB)



Leaf-wise Tree Growth (LGBM)



Experiment design



All classifiers have been trained using the Nvidia Tesla V100-SXM2 NVLink 32GB HBM2 within the ecosystem for tasks of machine learning, deep learning, and data analysis at **HybriLIT** platform

Two stages of the experiments

Some parameters for the tuning and model evaluation stages

Stage	Learning Rate	Max Number of Iterations	Early Stopping
Tuning	0.05	5 000	200
Model Evaluation	0.015	20 000	500

Results for hyperparameter tuning (after **30 iterations** of the TPE algorithm for each GBDT)

Framework	Max. Depth	L2 leaf reg.	Min. data in leaf	Rows sampling rate
XGBoost	8	2.3	0.00234	0.942
LightGBM	12	0.1	4	0.981
CatBoost	8	3.0	5	0.99
SketchBoost	8	3.0	5	0.99

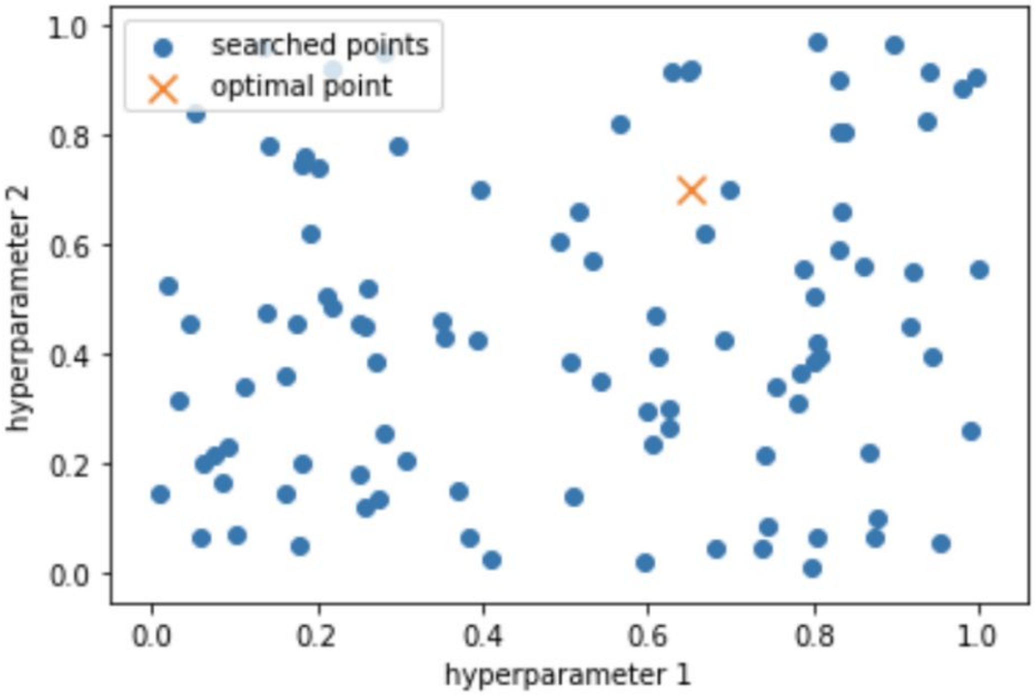
Iosipoi L., Vakhrushev A. SketchBoost: Fast Gradient Boosted Decision Tree for Multioutput Problems

Hyperparameters tuning

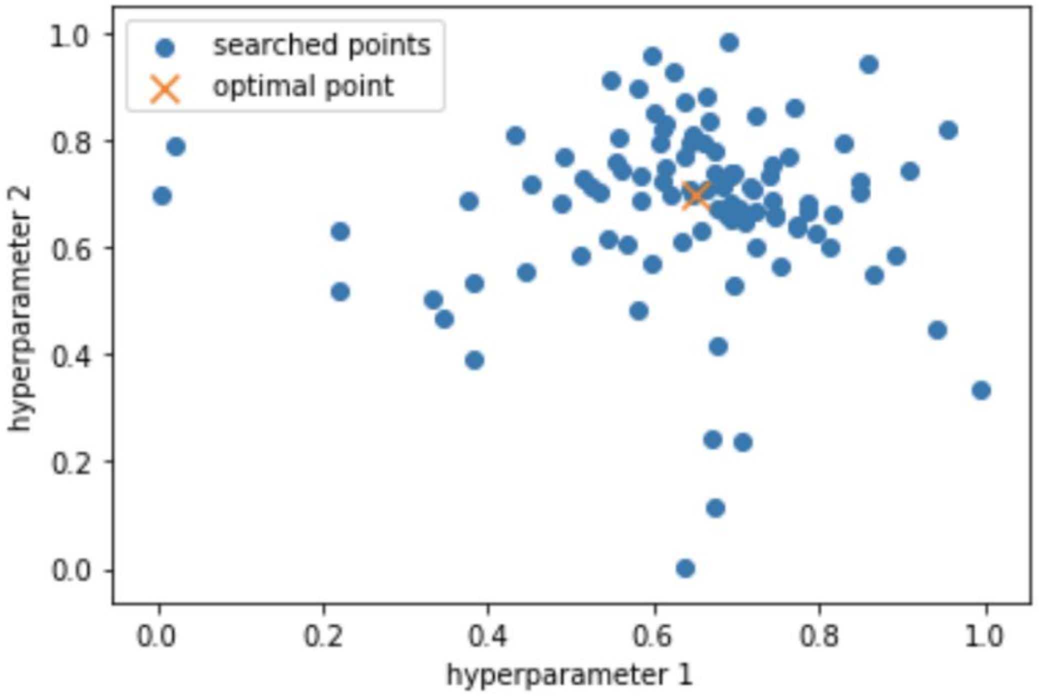
Tree-structured Parzen Estimator (TPE) was used to find the optimal hyperparameters;

TPE is a form of Bayesian Optimization.

Random search



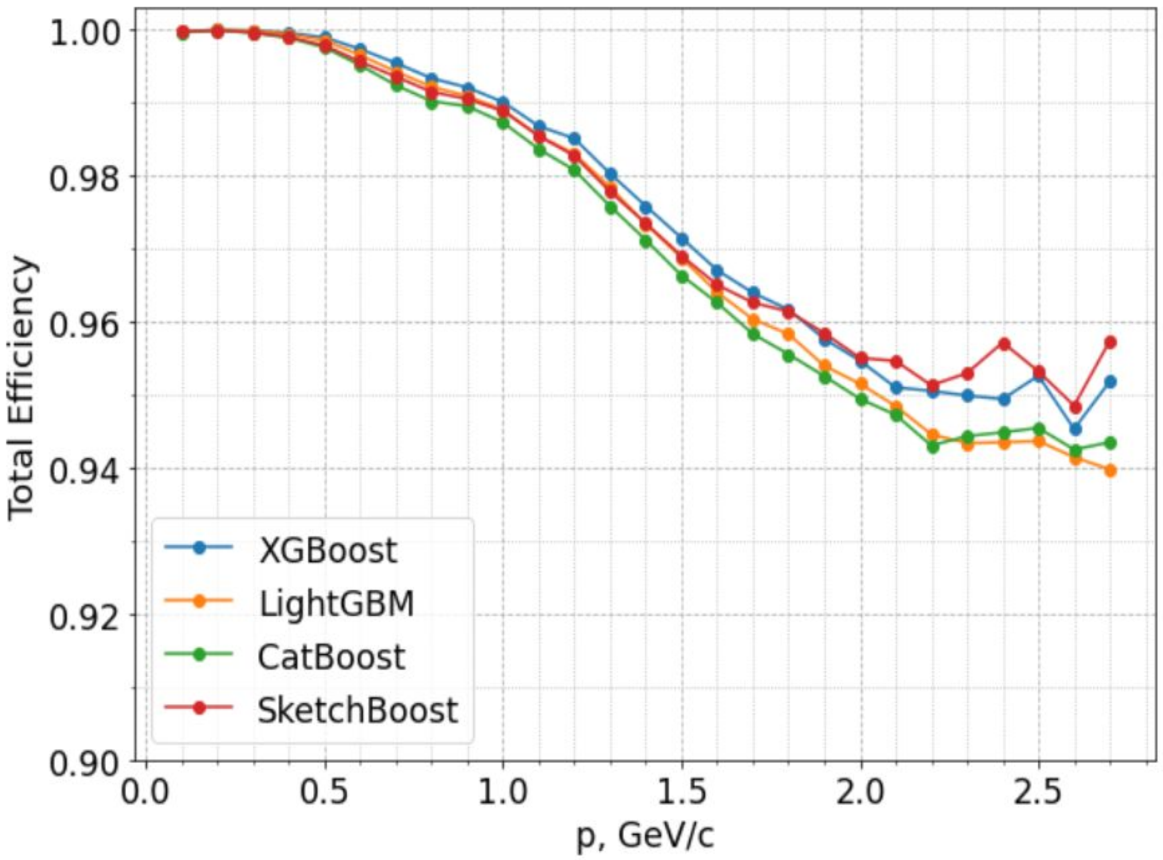
TPE search

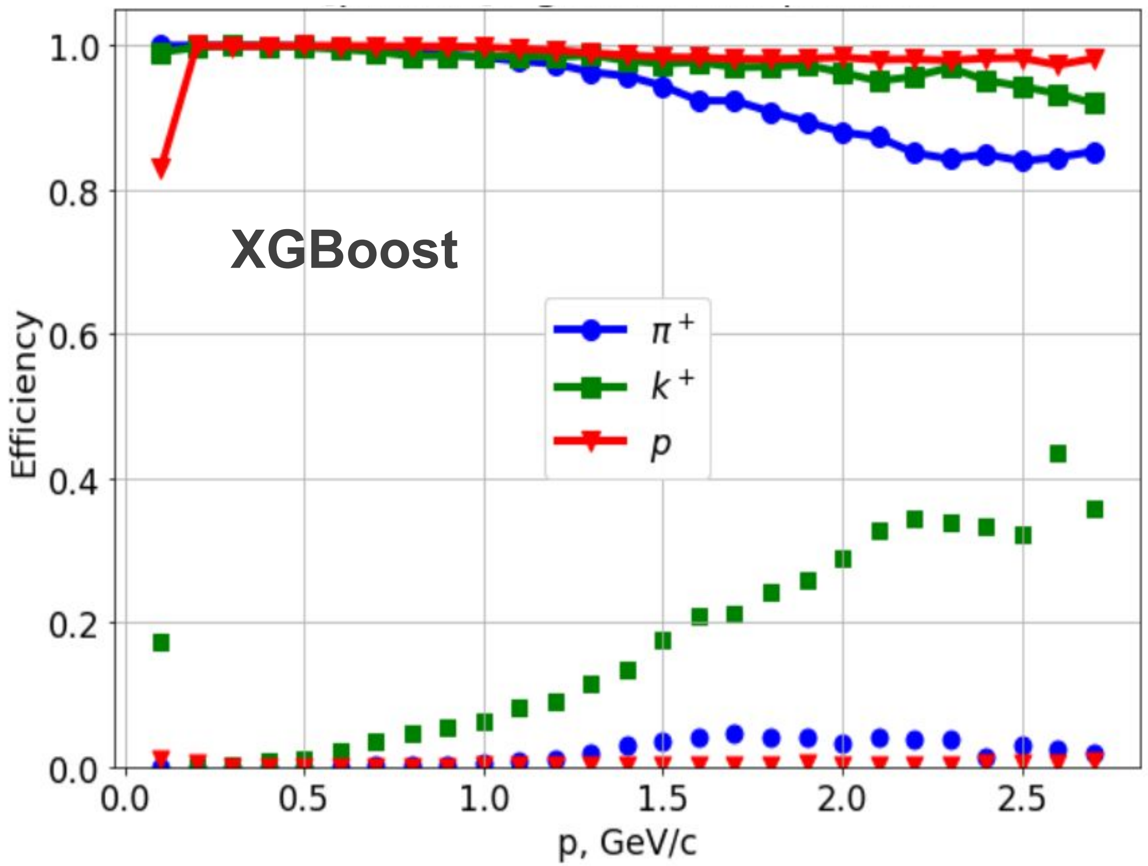


https://miro.medium.com/max/4800/1*tYWqO5BwNDVaM3kP3w1IAg.png

Comparative analysis of the algorithms. Efficiency

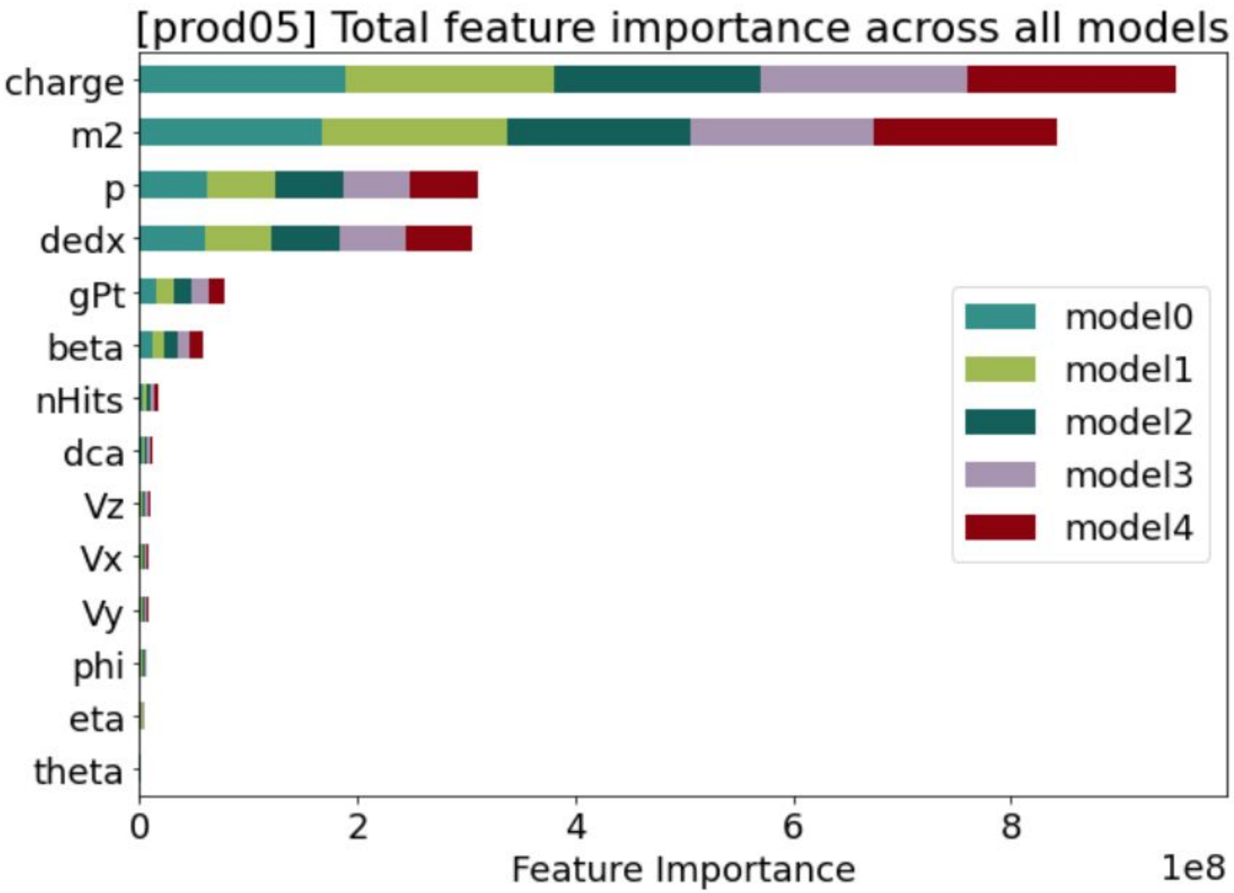
	XGBoost	LightGBM	CatBoost	SketchBoost
Total Efficiency	0.99327	0.99235	0.99138	0.99239





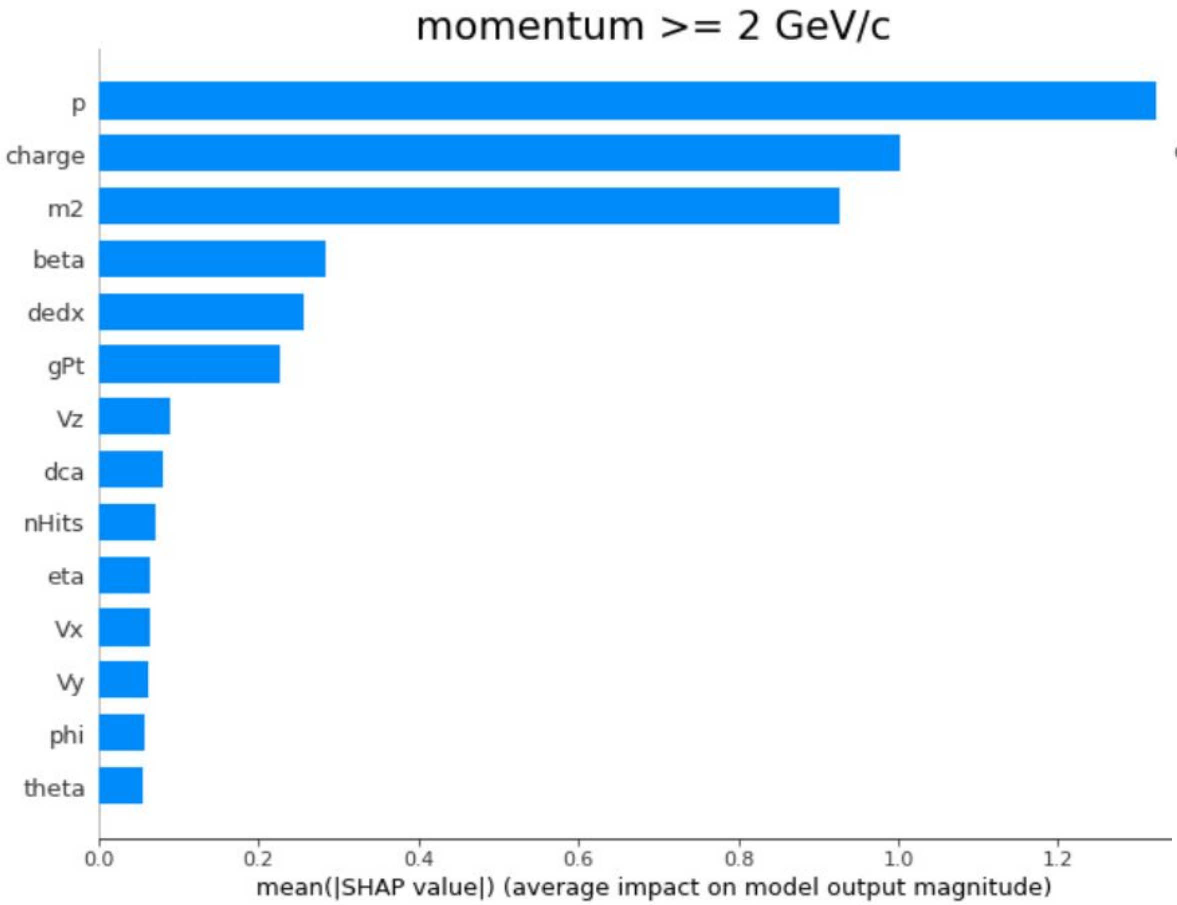
XGBoost Model Interpretation. Feature Importance

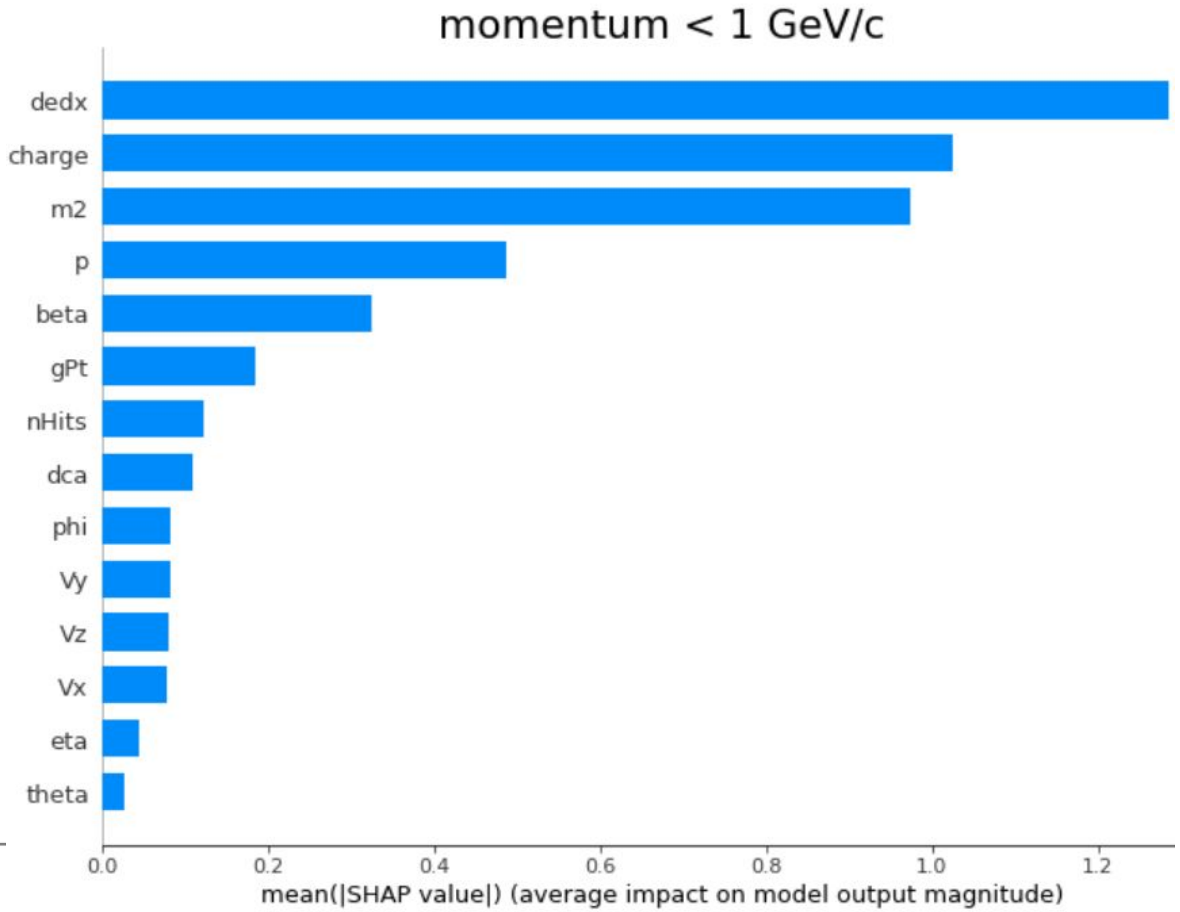
Importance type can be defined as the total gain across all splits the feature is used in



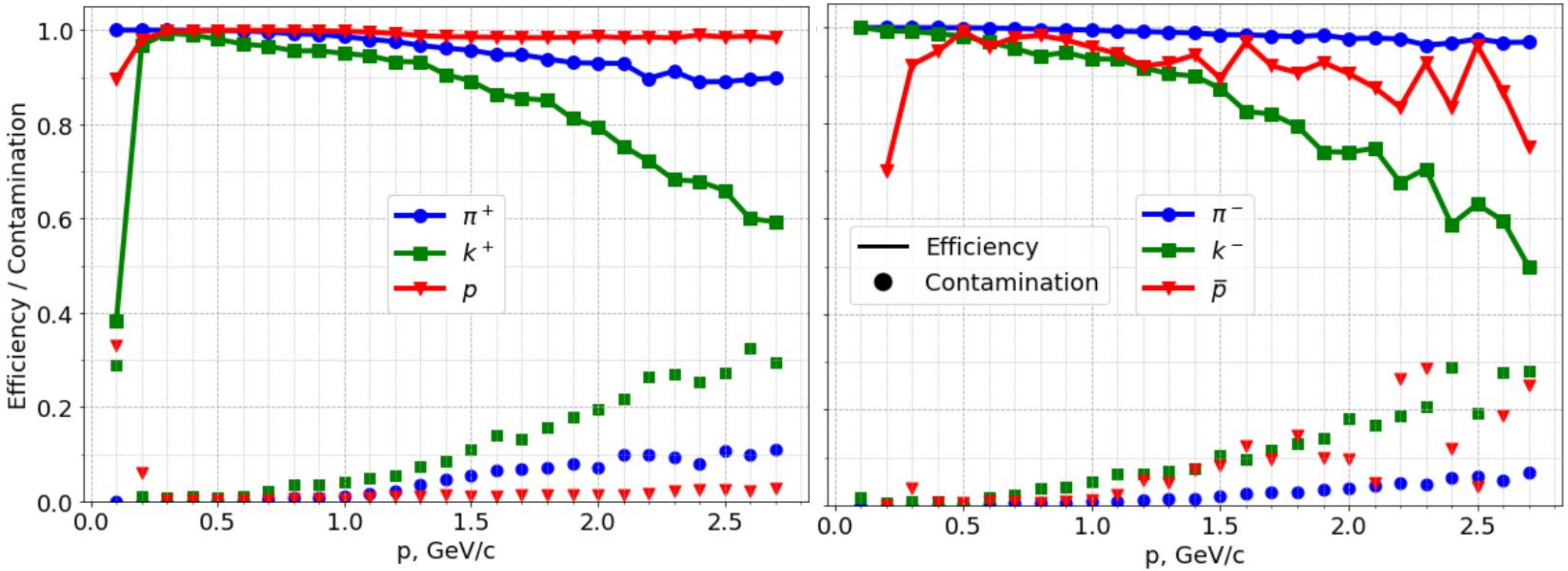
This approach are sensitive when input variables are correlated, and may lead for instance to unreliability in the importance ranking

Misclassification. Positive pions





Test XGBoost with expected distribution on Request 25



Efficiency and contamination of XGBoost