Seminar of the laboratory
14 February 2024

# Gradient Boosted Decision Tree for Particle Identification at MPD

V. Papoyan[1,3]

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

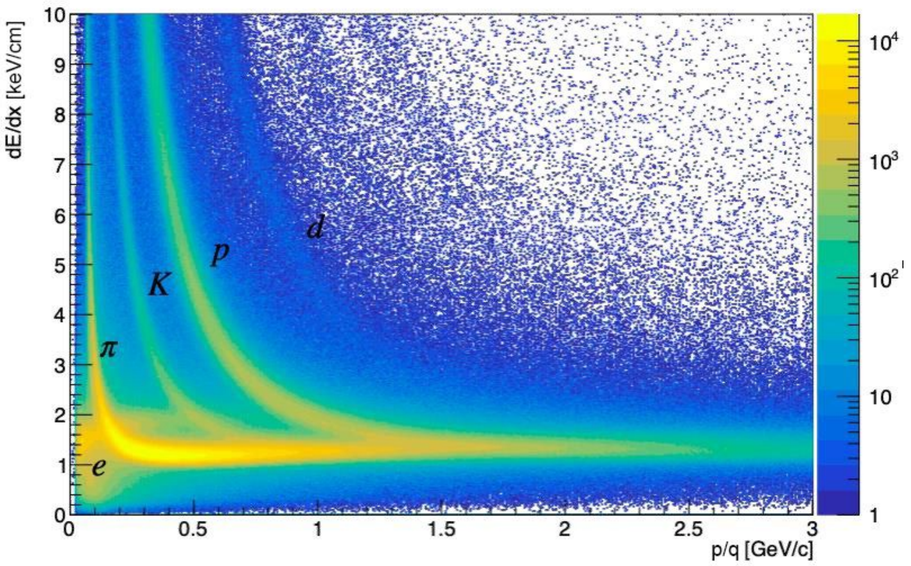
[1]MLIT JINR, [2]VBLHEP JINR, [3]AANL (YerPhi)

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

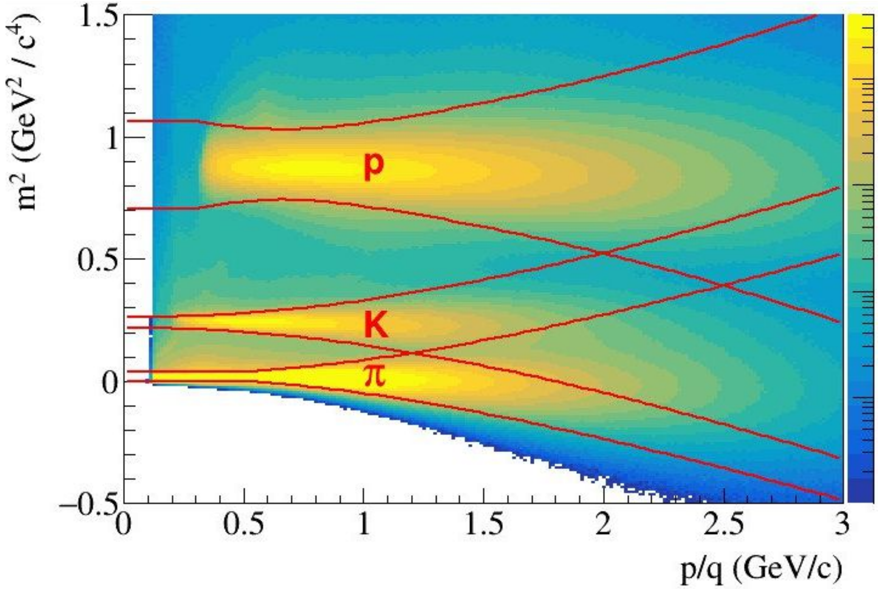
# Particle Identification at MPD experiment

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;



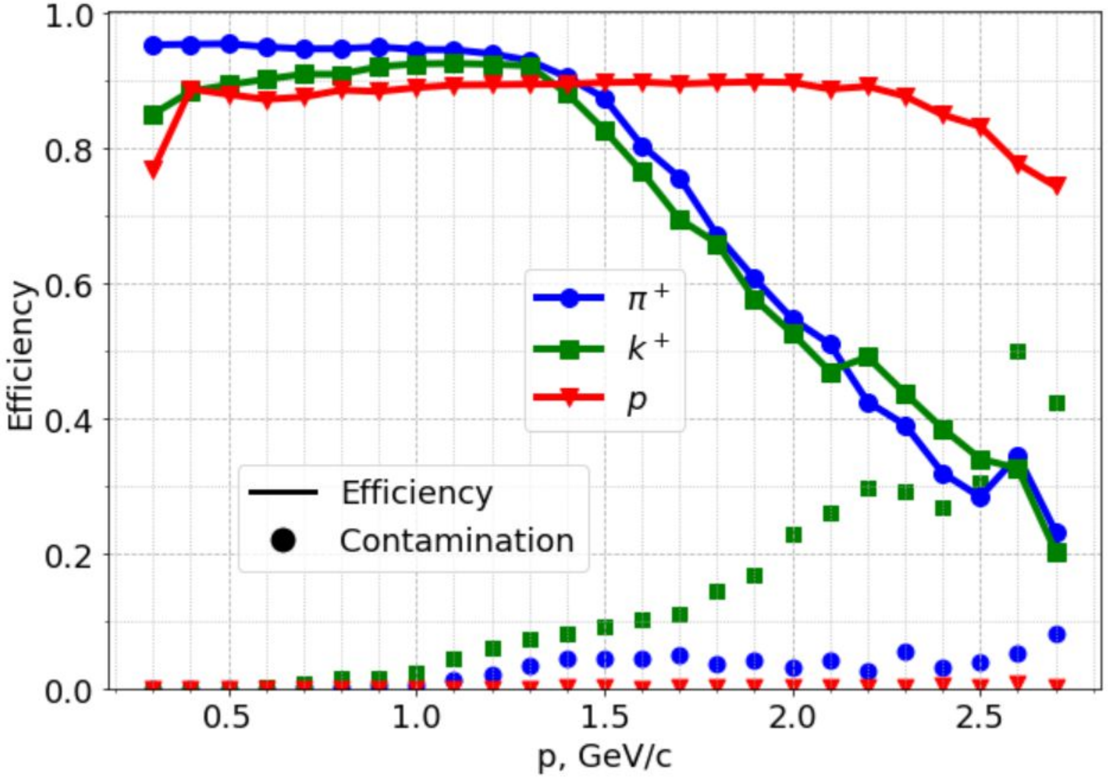

Knowing the particle **momentum** (from TPC) one obtains the **mass squared** and thus identity of the particle.

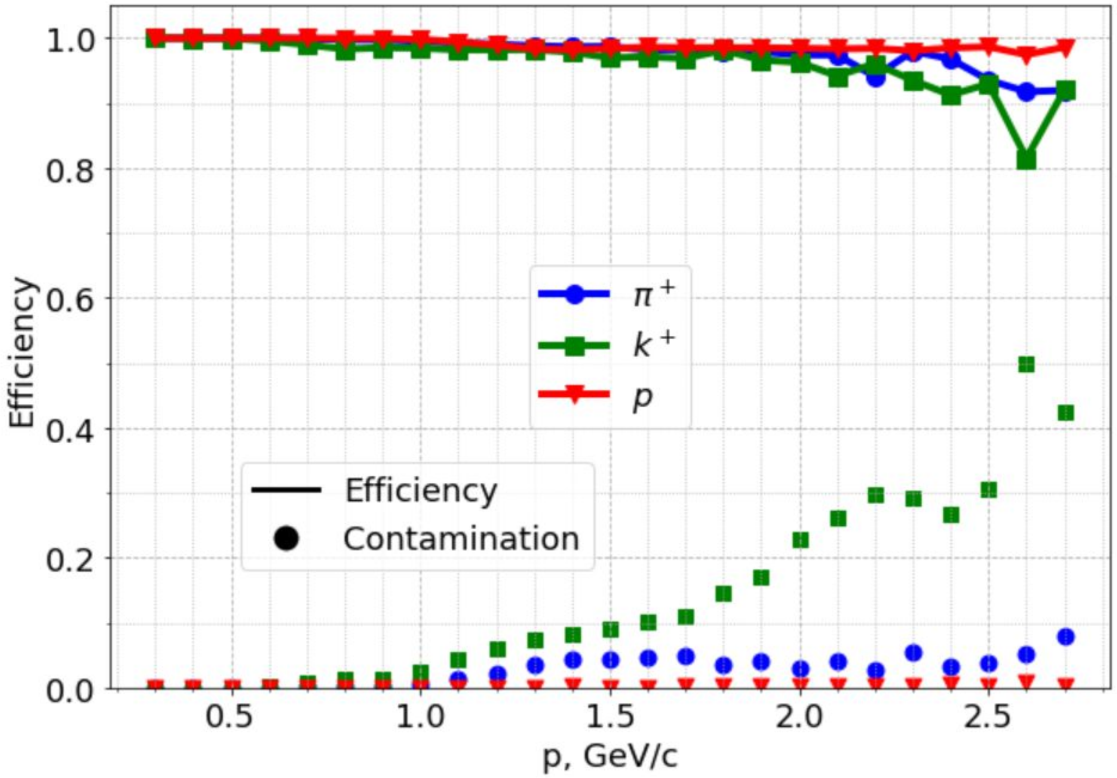
*Klempt W. Review of particle identification by time of flight techniques*

# Baseline PID at MPD - N-sigma

$$E^S = \frac{N^S{}_{corr}}{N^S{}_{true}}$$

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



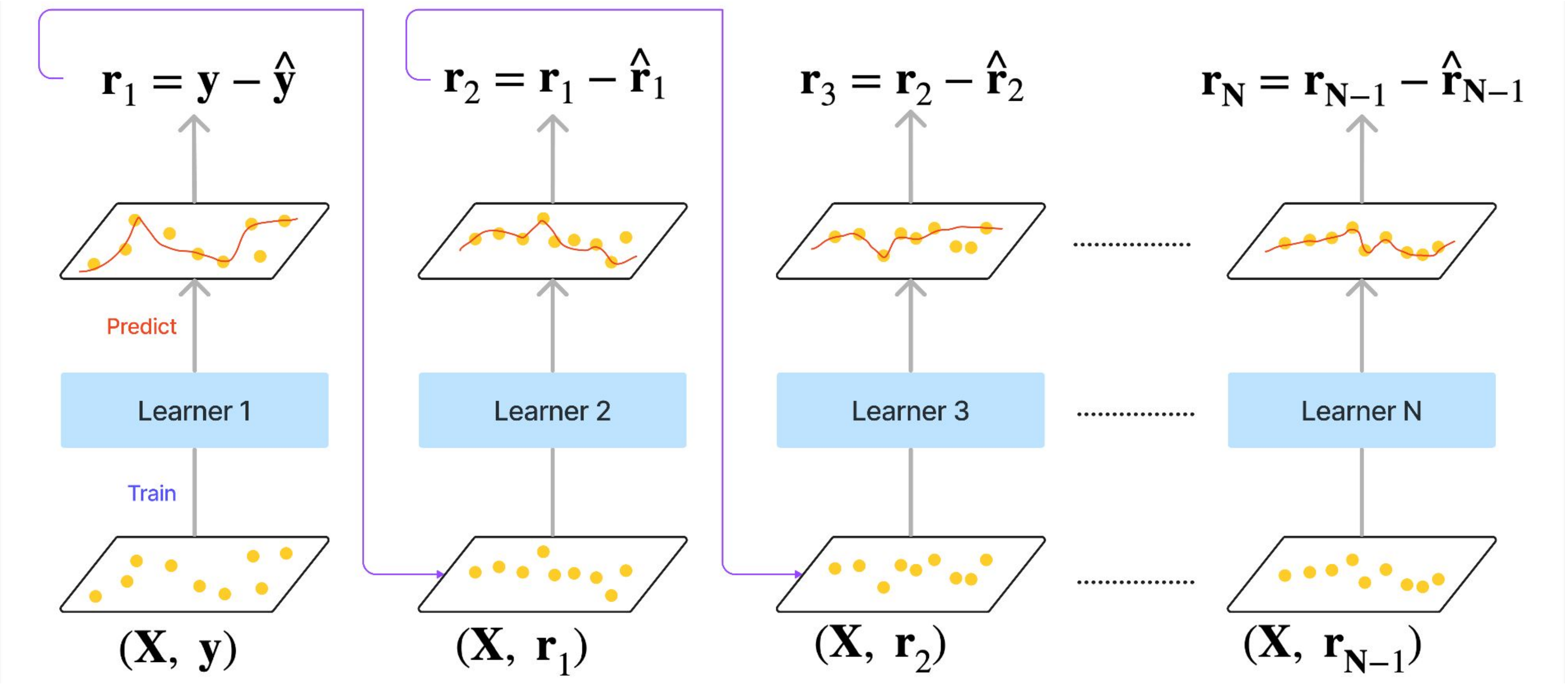




PID efficiency and contamination for all tracks (left) and only identified tracks (right)

in Bi+Bi collisions at 9.2 GeV

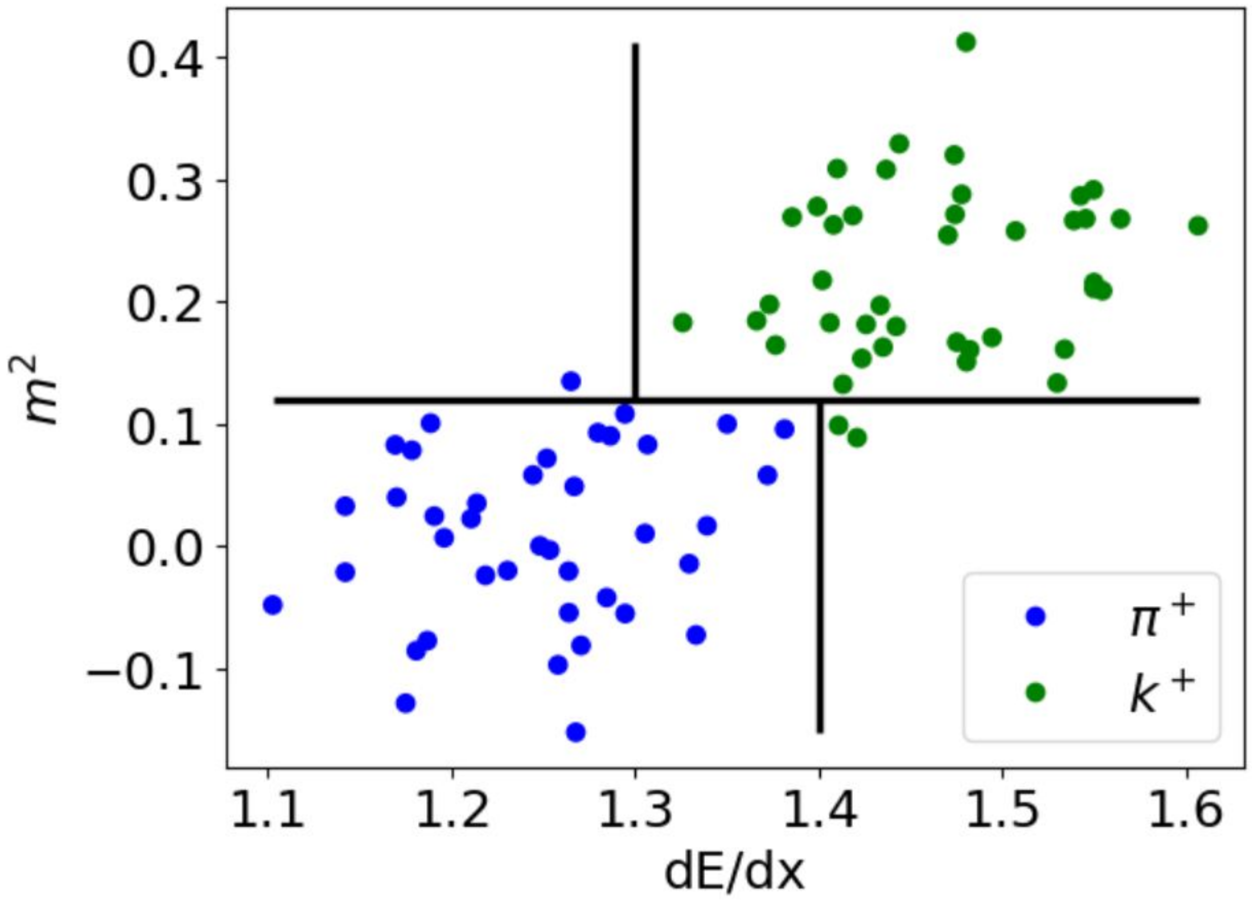
# Gradient Boosting

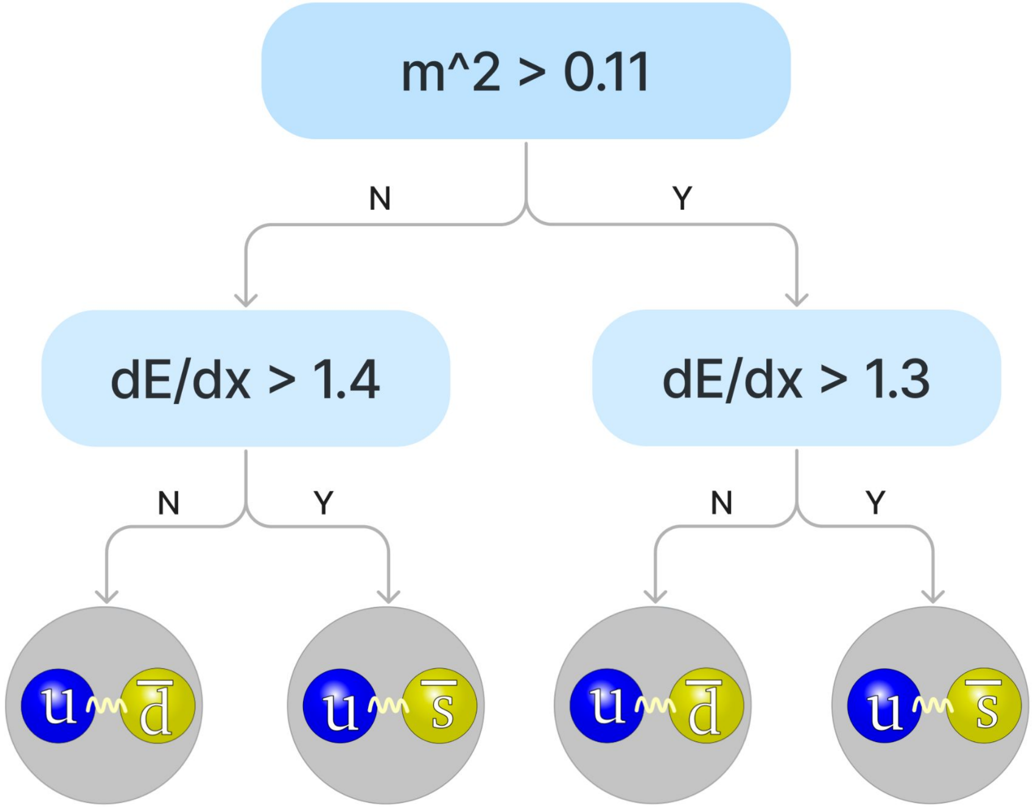
**Gradient boosting** is a machine learning technique which combines weak learners into a single strong learner in an iterative fashion



# Gradient Boosted Decision Tree

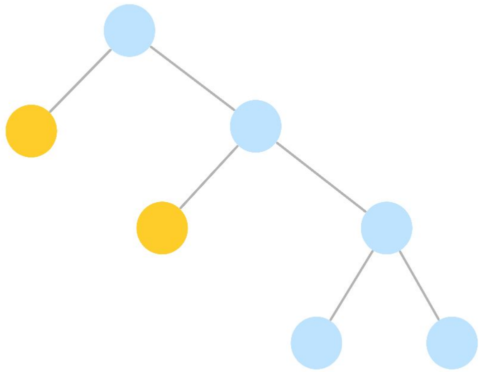
**Gradient Boosted Decision Tree** (GBDT) uses decision trees as weak learner. They can be considered as automated multilevel **cut-based** analysis



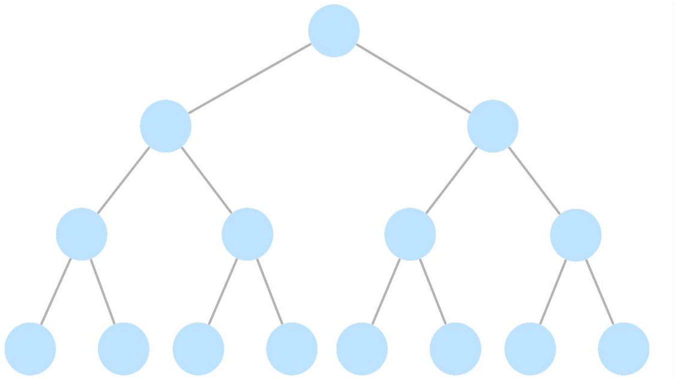



# XGBoost vs LightGBM vs CatBoost vs SketchBoost

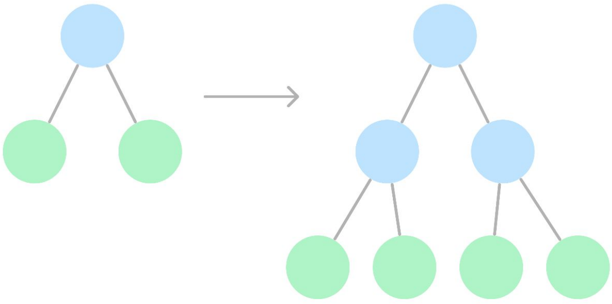
Asymmetric Tree (XGB, LGBM)



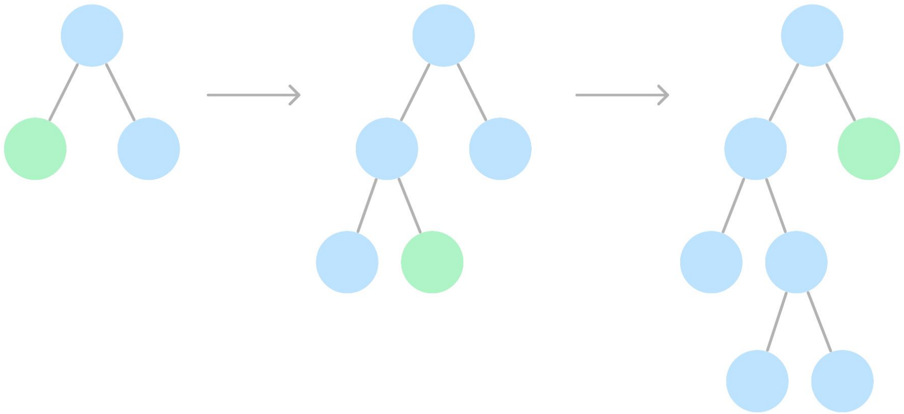
Symmetric Tree (CatBoost, SketchBoost)



Level-wise Tree Growth (XGB)



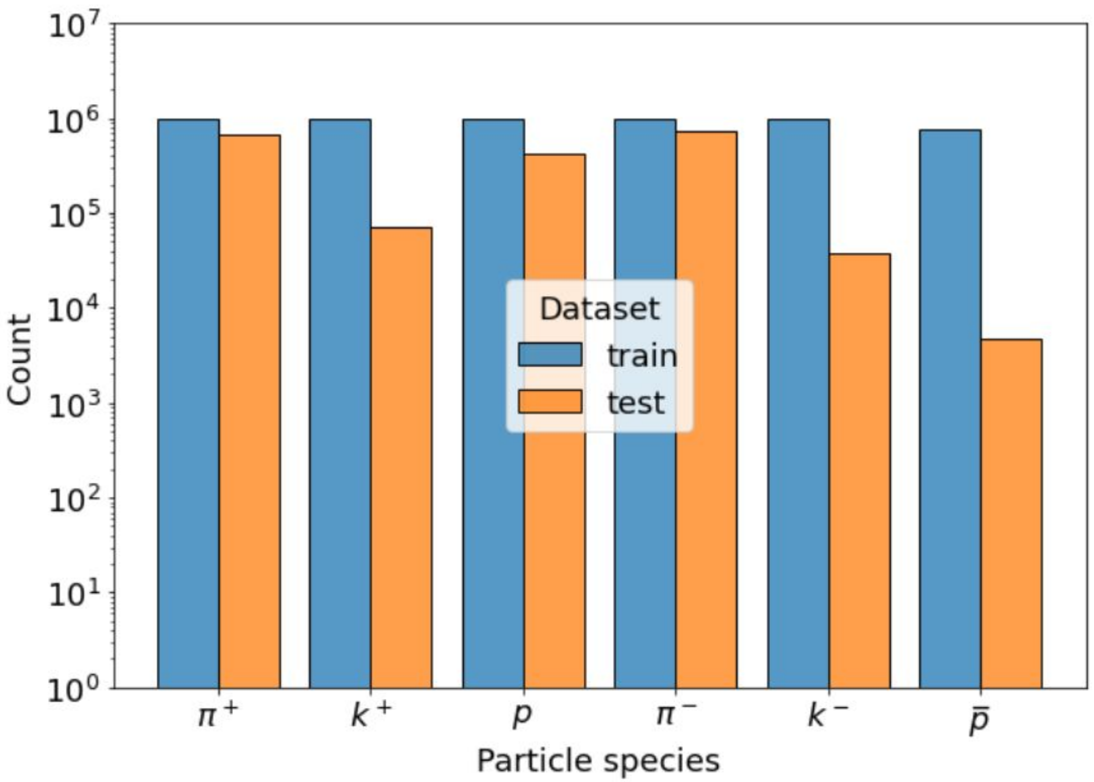
Leaf-wise Tree Growth (LGBM)



# Datasets

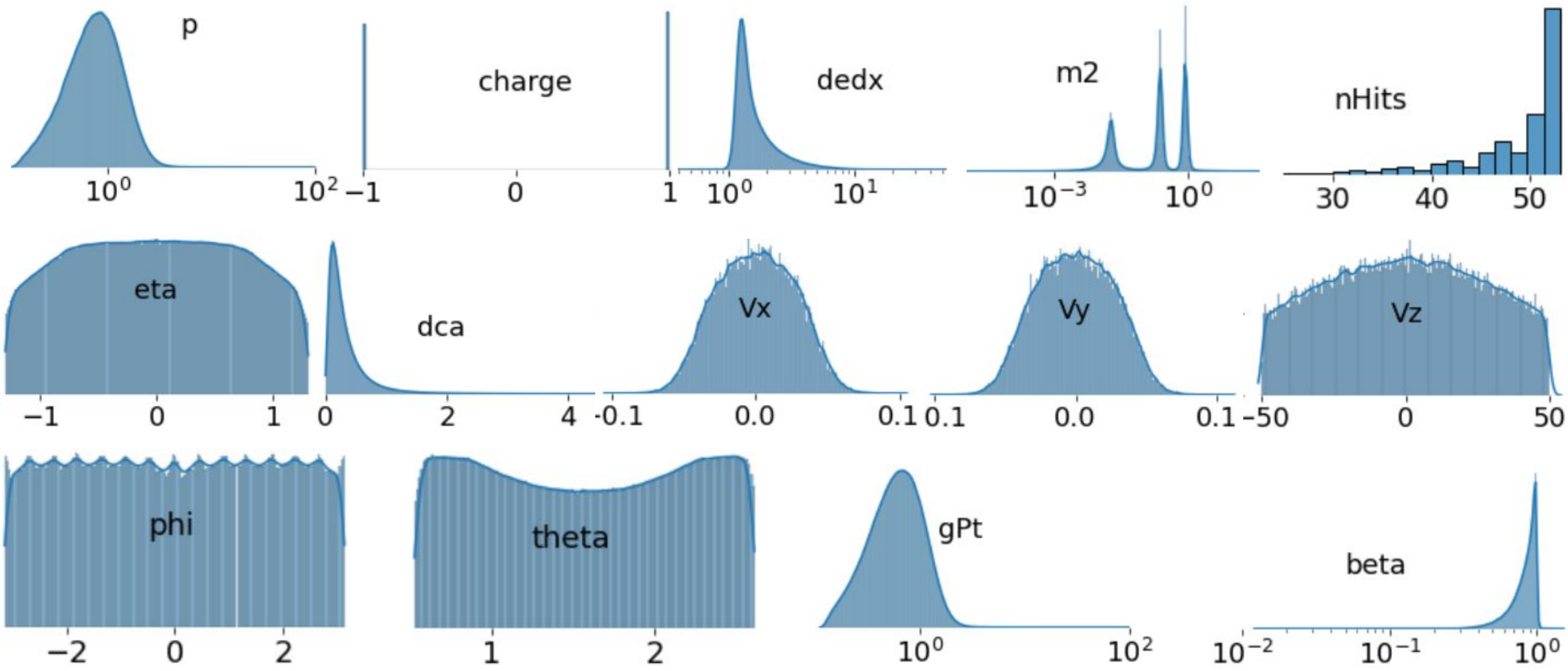
Subsamples of the two MPD Monte-Carlo productions have been used (Request 25 & Request 29)

| | **prod05** | **prod06** |
|---|---|---|
| Event generator | UrQMD | PHQMD |
| Transport | Geant 4 | Geant 4 |
| Impact parameter ranges | 0-16 fm (mb) | 0-12 fm |
| Smear Vertex XY | 0.1 cm | 0.1 cm |
| Smear Vertex Z | 50 cm | 50 cm |
| Colliding system | Bi+Bi | Bi+Bi |
| Energy | 9.2 GeV | 9.2 GeV |

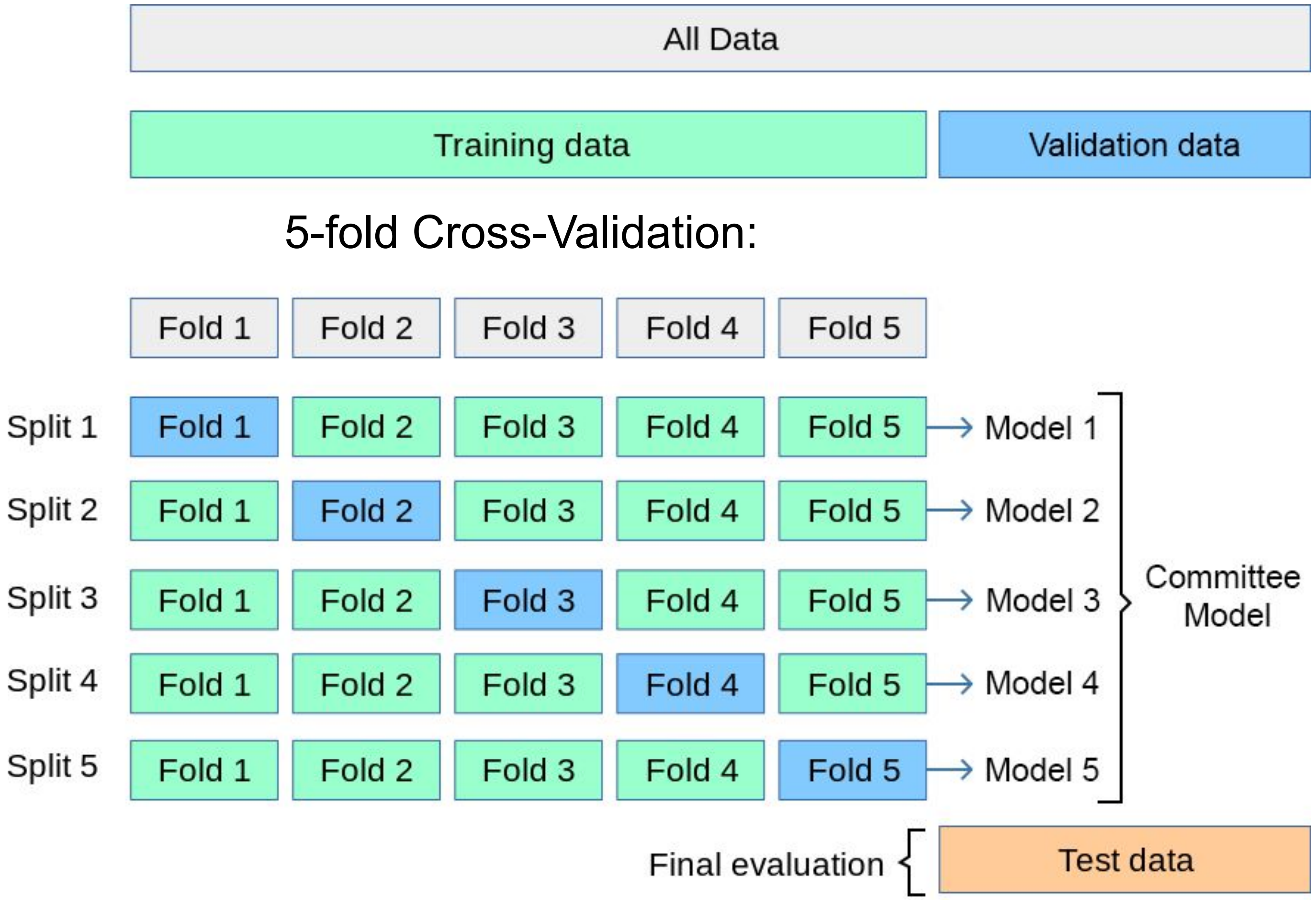



**track selection criteria**: (p < 100) & ($|m^2|$ < 100) & (nHits > 15) & (|eta|<1.5) & (dca < 5) & (|Vz| < 100)

# Data description



# 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 |

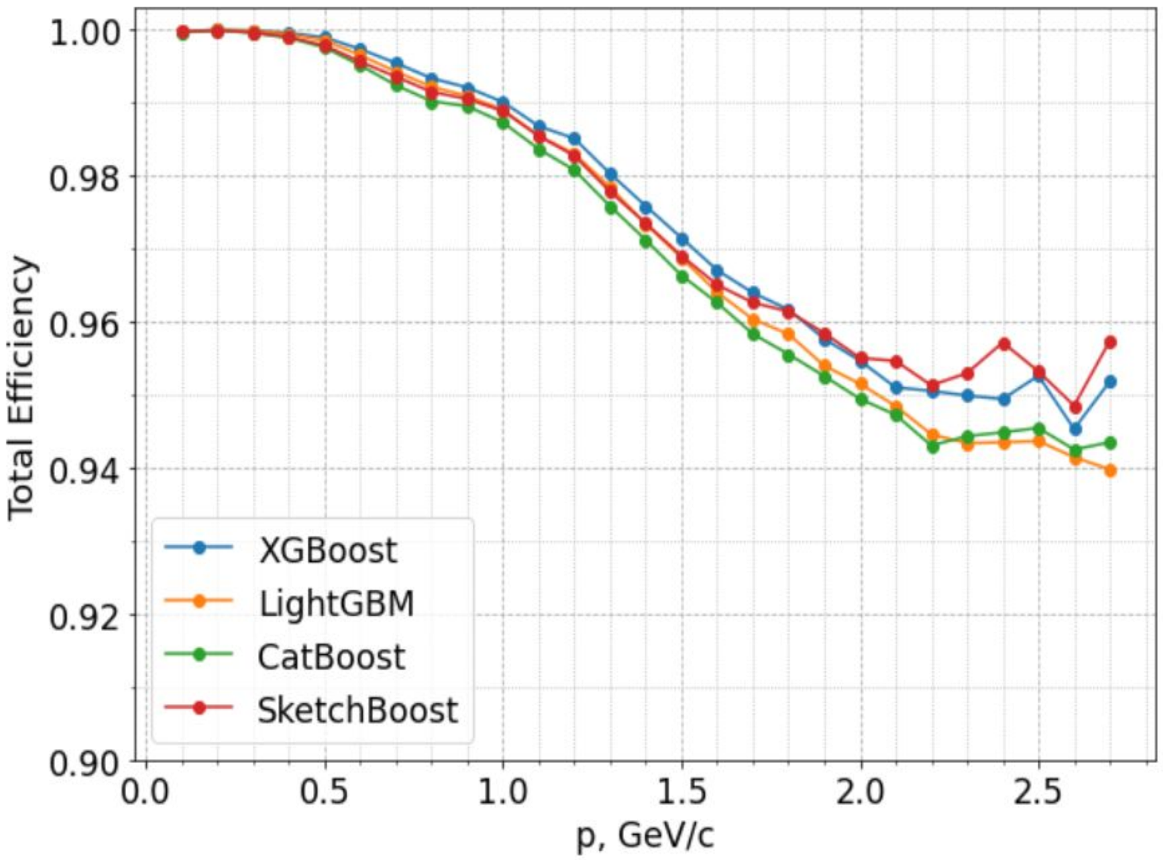
Results for hyperparameter tuning (after **30 iterations** of the TPE algorithm for each GBDT)

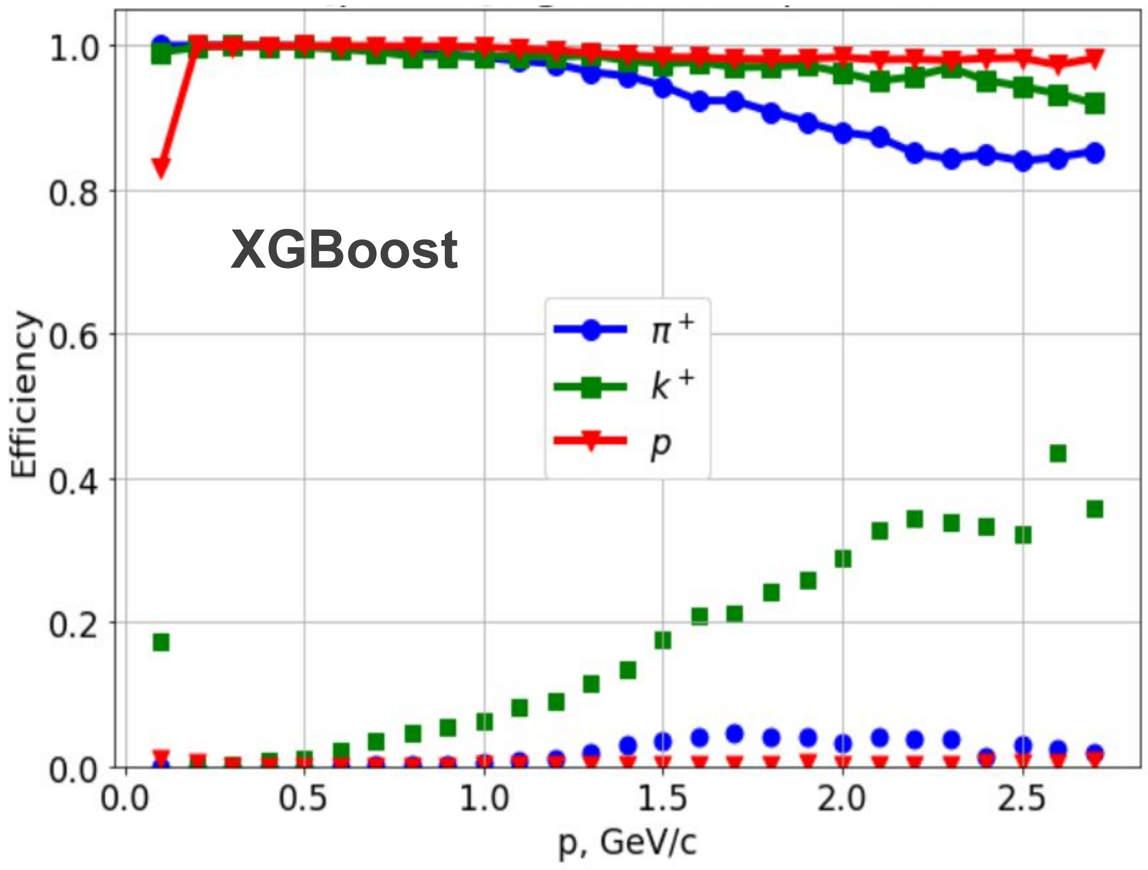
| Framework | Max. Depth | L2 leaf reg. | Min. data in leaf | Rows sampling rate |
|---|---|---|---|---|
| XGBoost | 8 | 2.3 | 0.00234 | 0.942 |
| LightGBM | 12 | 0.1 | 4 | 0.981 |
| CatBoost | 8 | 3.0 | 5 | 0.99 |
| SketchBoost | 8 | 3.0 | 5 | 0.99 |

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

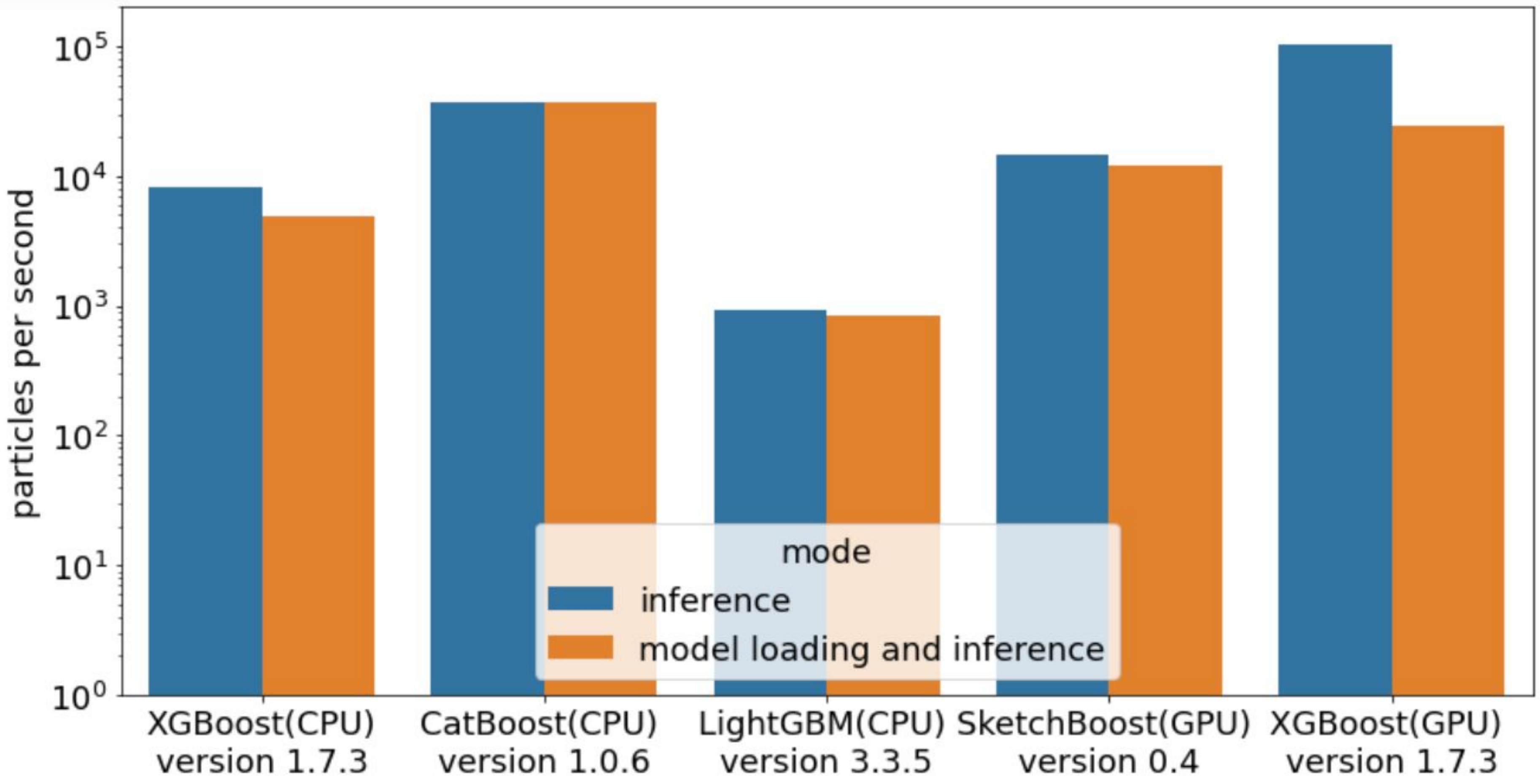
# Comparative analysis of the algorithms. Efficiency

| | XGBoost | LightGBM | CatBoost | SketchBoost |
|---|---|---|---|---|
| Total Efficiency | 0.99327 | 0.99235 | 0.99138 | 0.99239 |



# Comparative analysis of the algorithms. Inference time

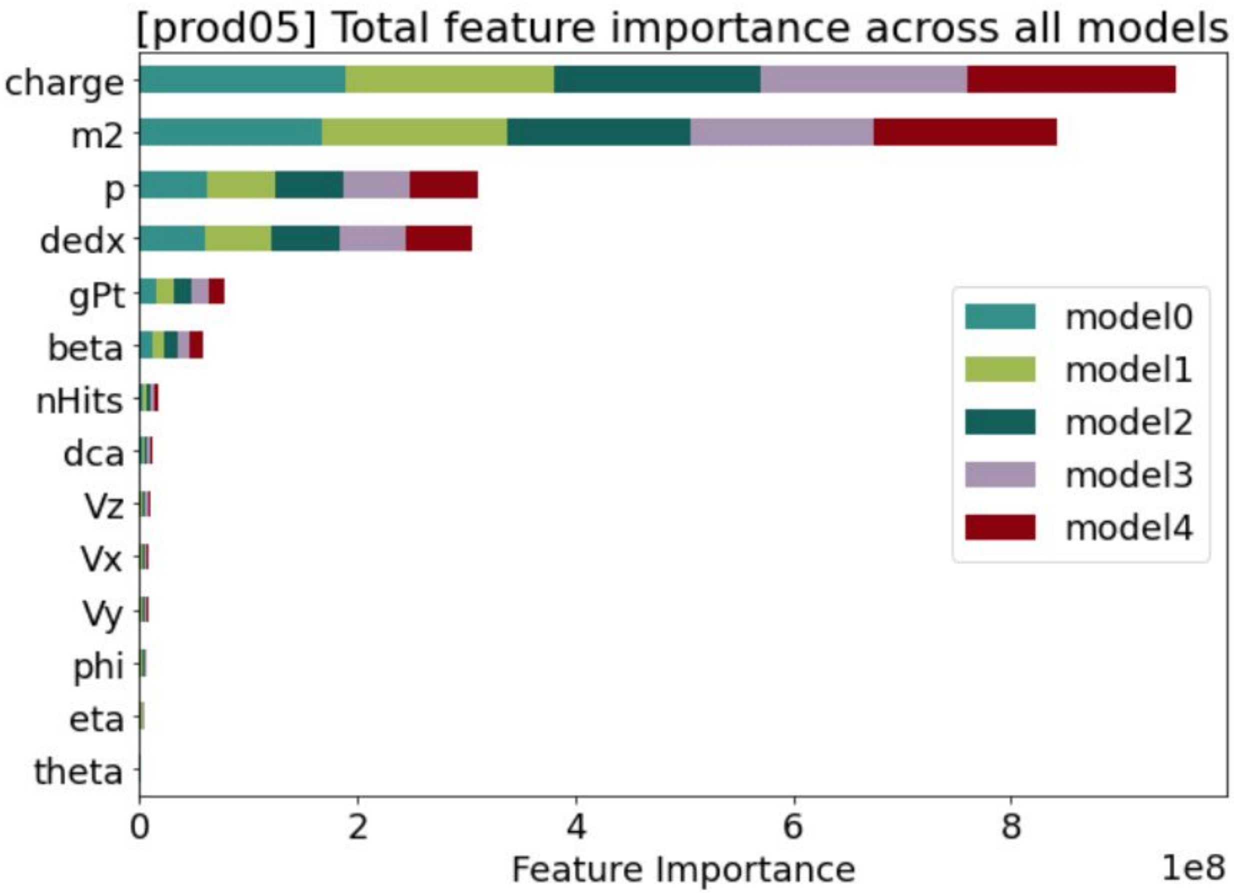



**GPU**: Nvidia Tesla V100-SXM2 NVLink 32GB HBM2

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

# XGBoost Model Interpretation. Feature Importance

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



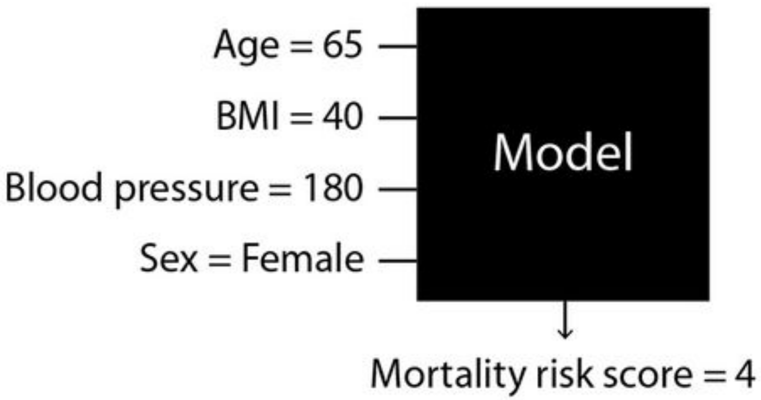

This approach are sensitive when input variables are correlated, and may lead for instance to unreliability in the importance ranking

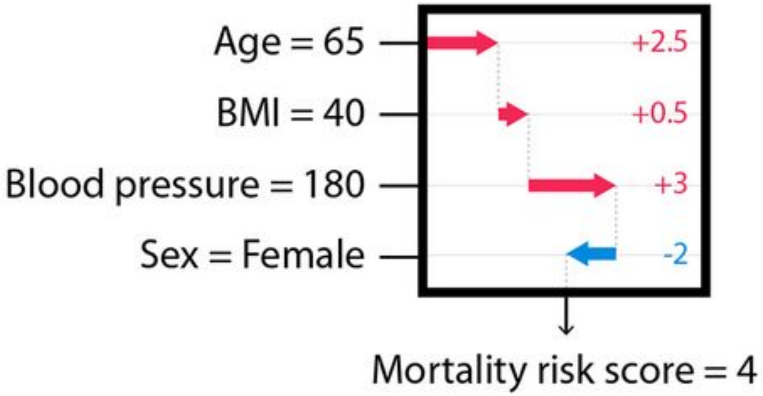
# Model Interpretation. Shapley Additive exPlanations

**SHAP** is a game theoretic approach to explain the output of any ML model

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[ f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right].$$

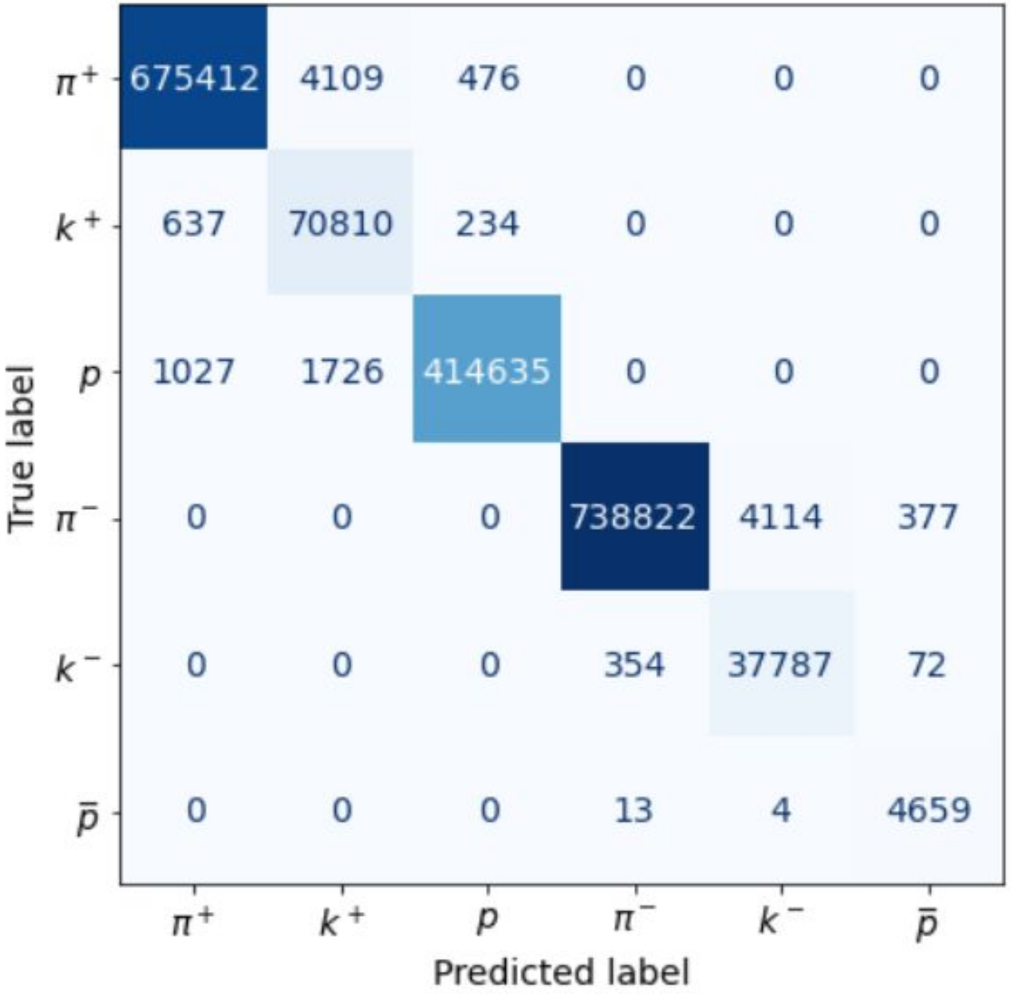
**|F|** is the size of the full coalition. **S** represents any subset of the coalition that doesn't include player **i**. The bit at the end is just "how much bigger is the payoff when we add player **i** to this particular subset **S**"

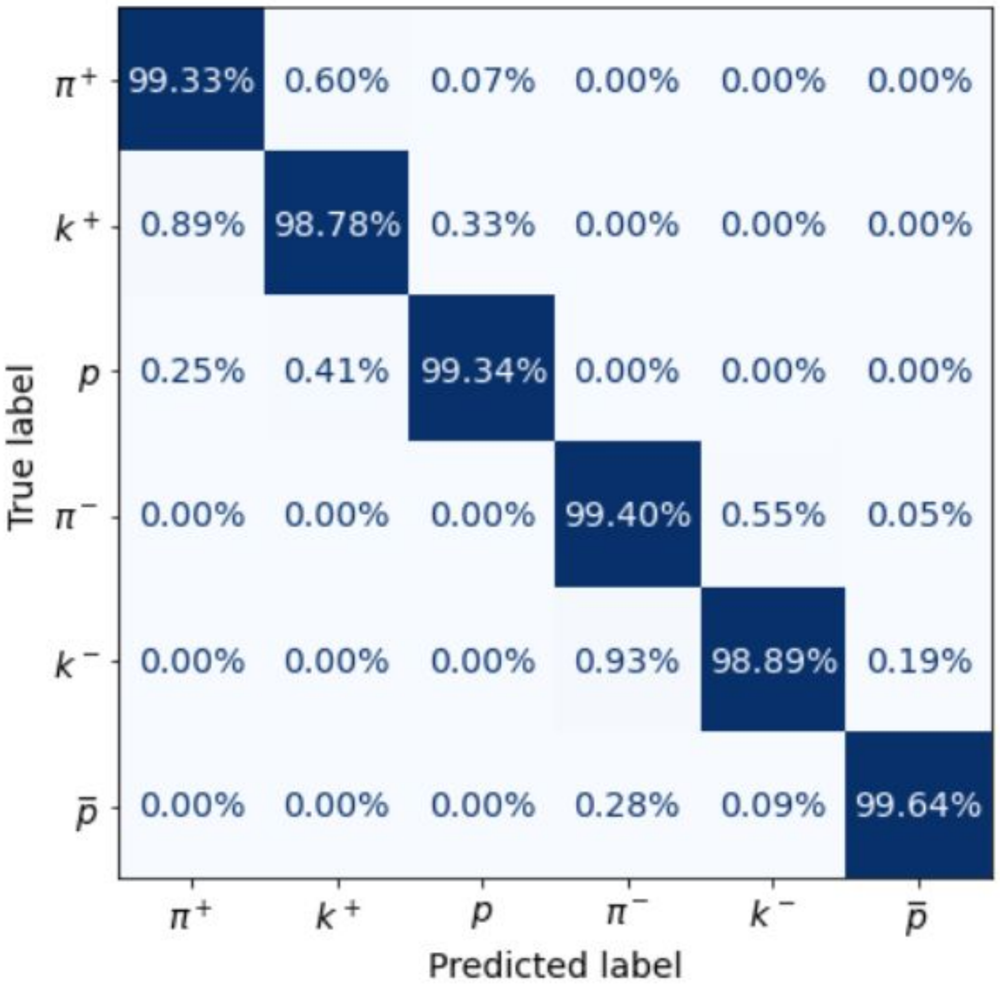


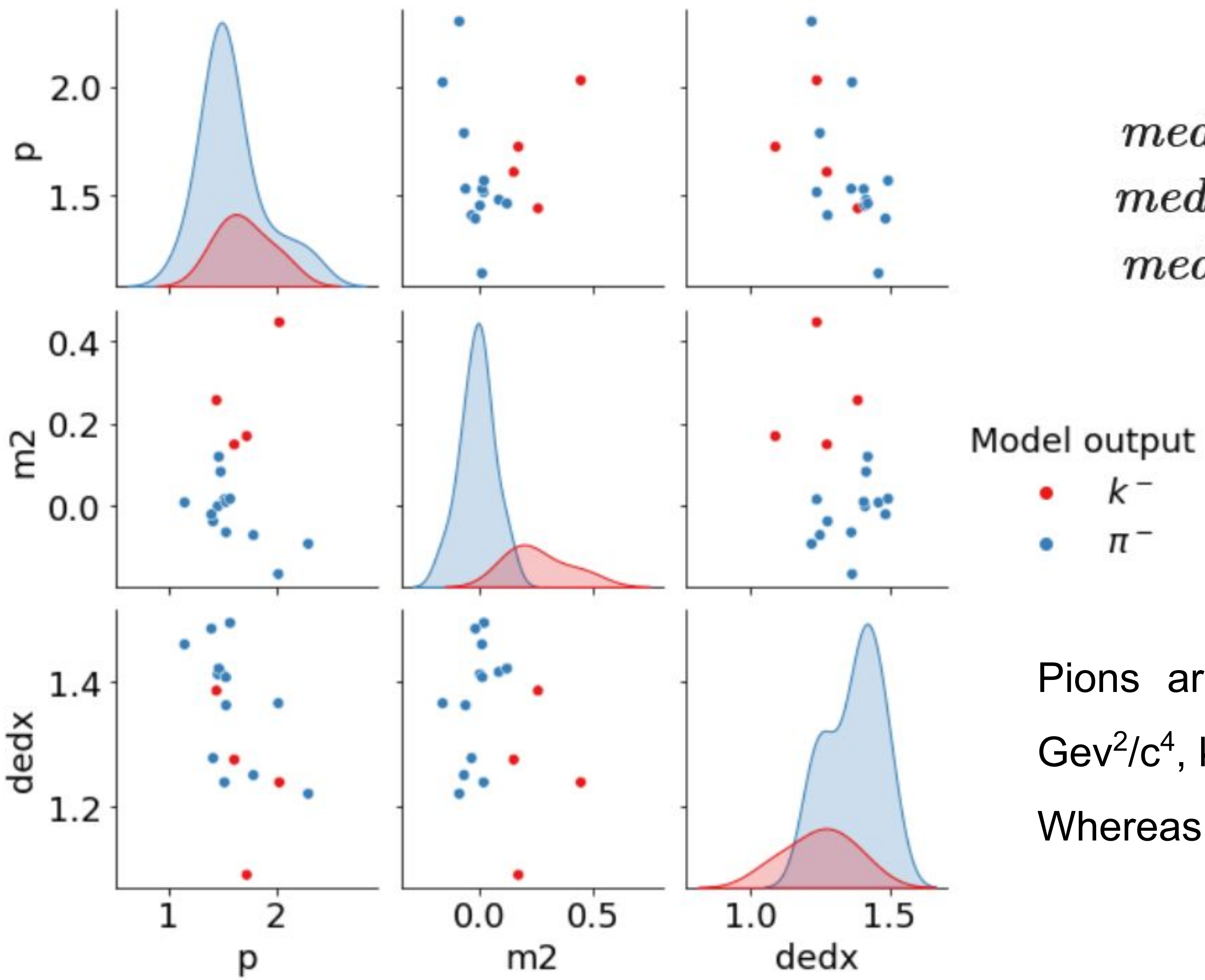

*Lundberg S. M. et al. From local explanations to global understanding with explainable AI for trees //Nature machine intelligence. – 2020. – T. 2. – №. 1. – C. 56-67.*

# Misclassification. Confusion Matrices



# Misclassification. Antiprotons



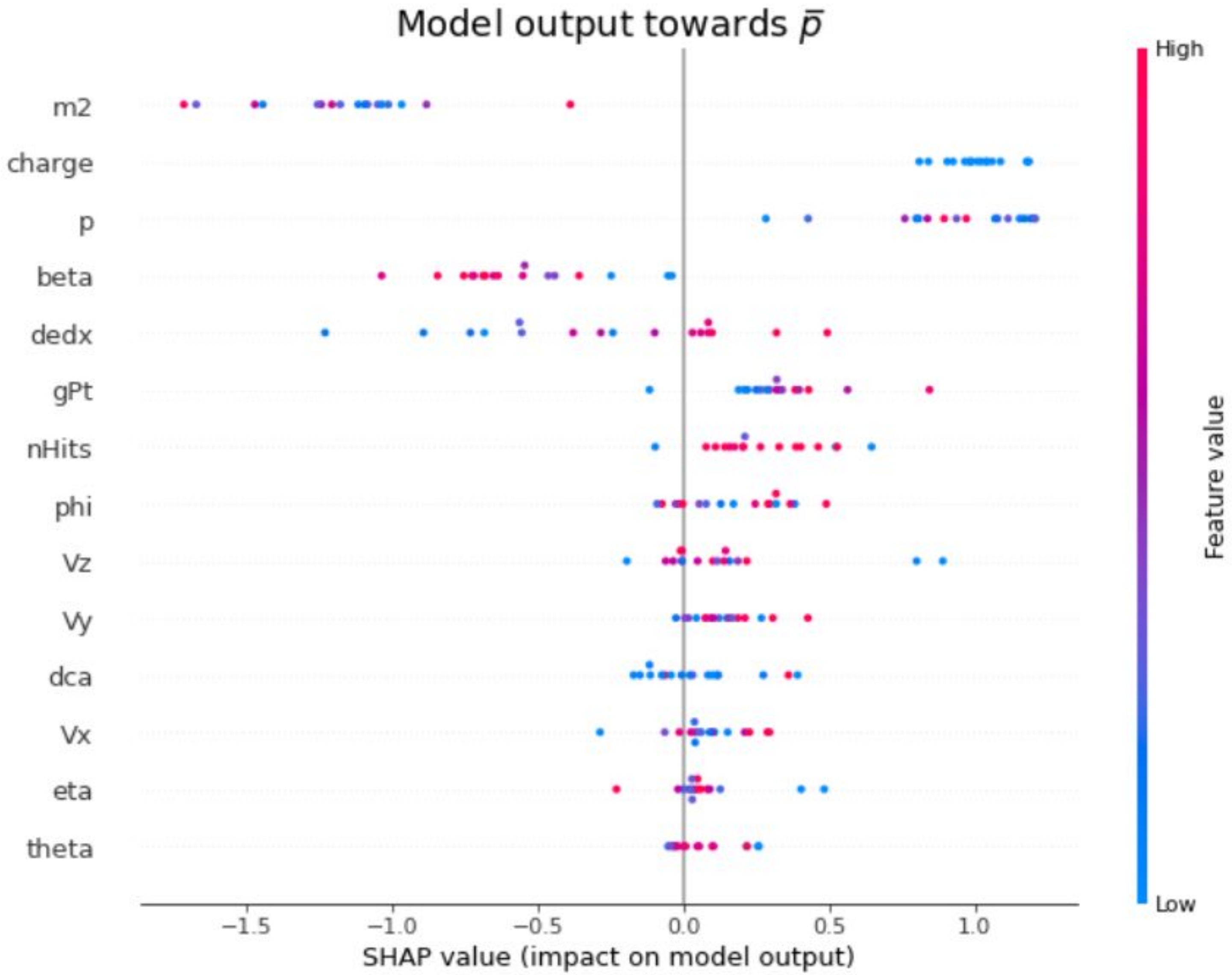

**Median** mass squared:

$$median(m_\pi^2) = 0.0178 \quad GeV^2/c^4$$
$$median(m_K^2) = 0.2362 \quad GeV^2/c^4$$
$$median(m_p^2) = 0.8664 \quad GeV^2/c^4$$

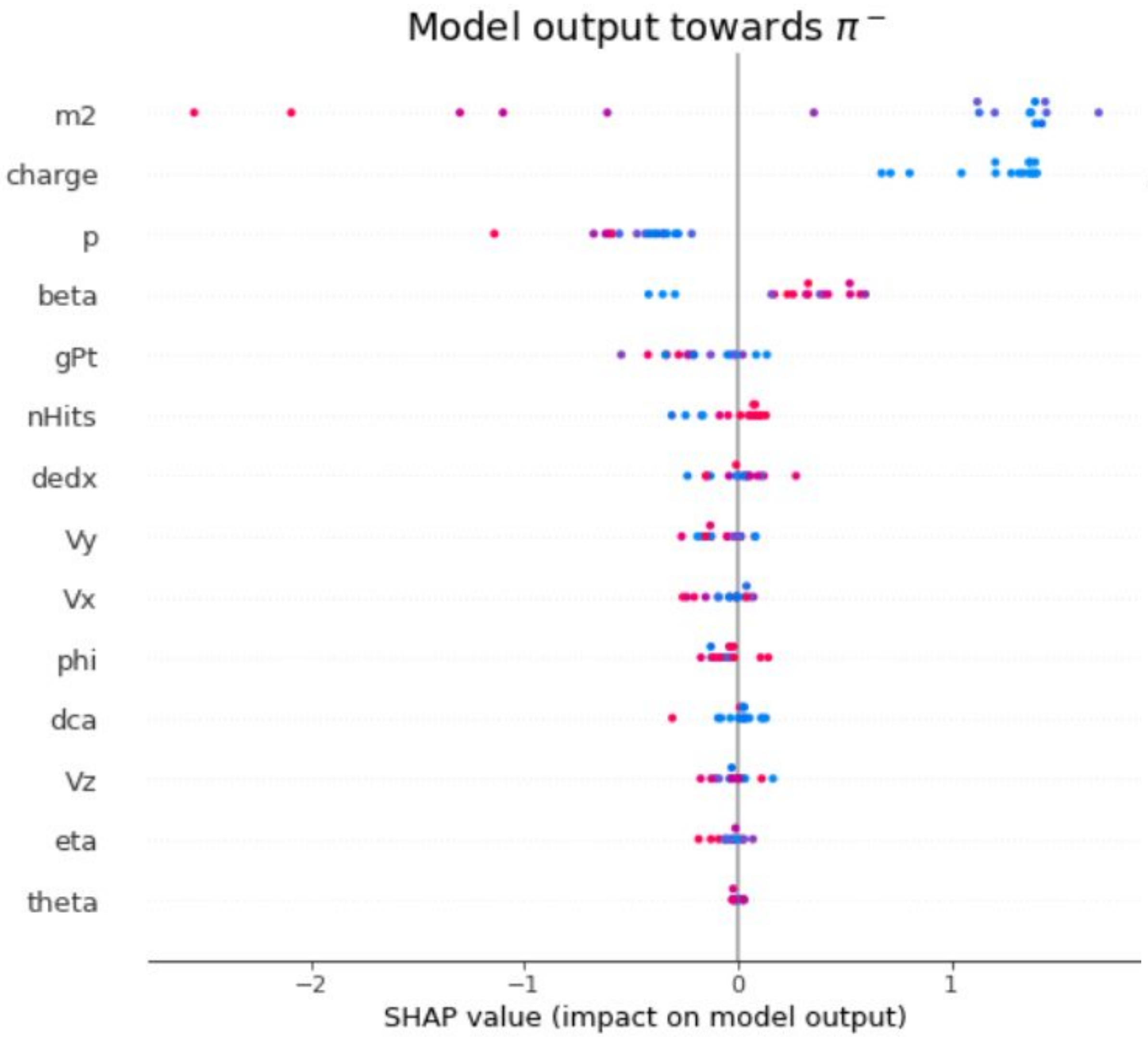
Pions are located in the vicinity of $m^2$=**0.01** Gev$^2$/c$^4$, kaons are closed to **0.2** Gev$^2$/c$^4$.

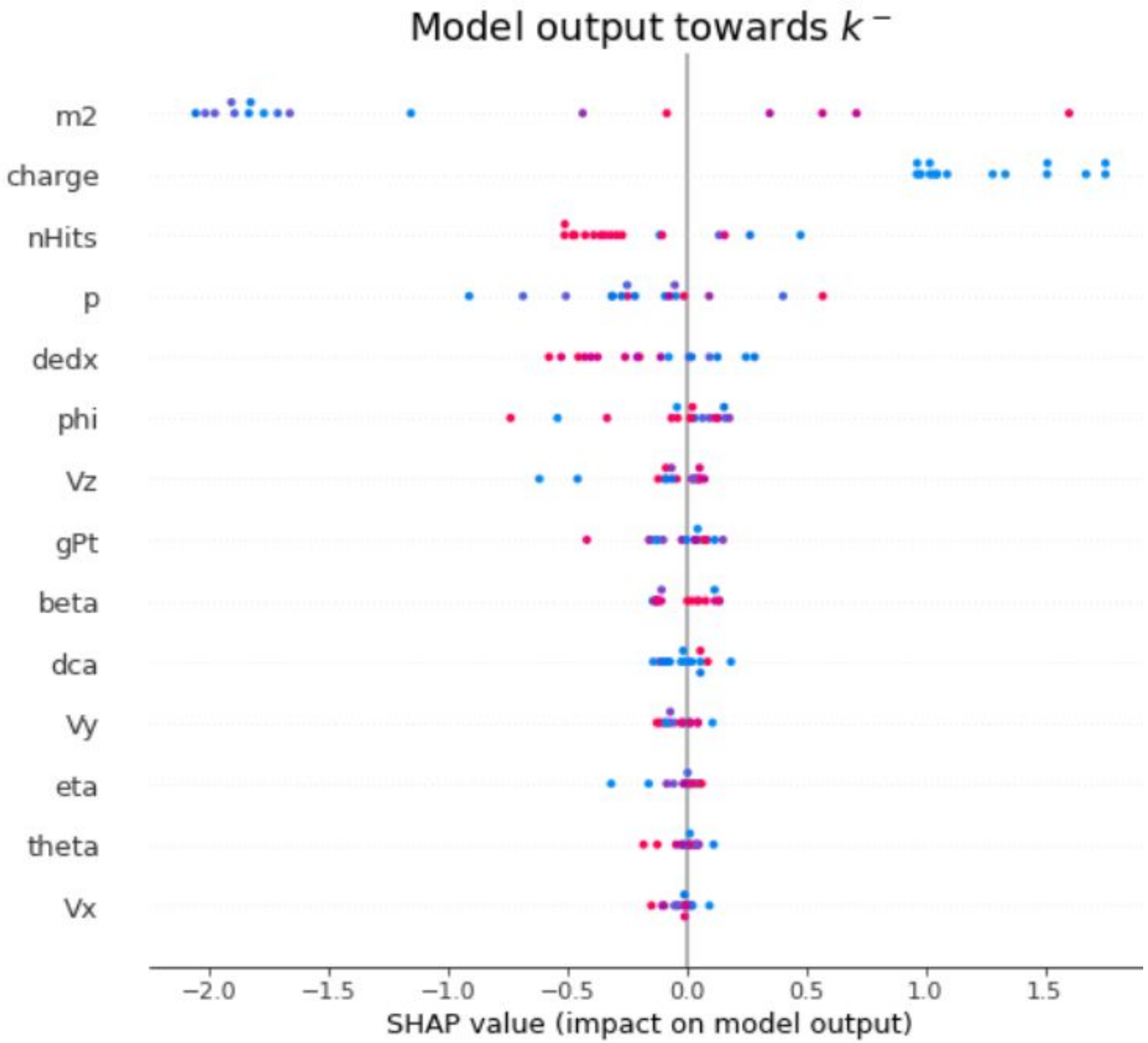
Whereas $m^2$=**0.88** GeV$^2$/c$^4$ is typical for protons.

# Misclassification. Antiprotons



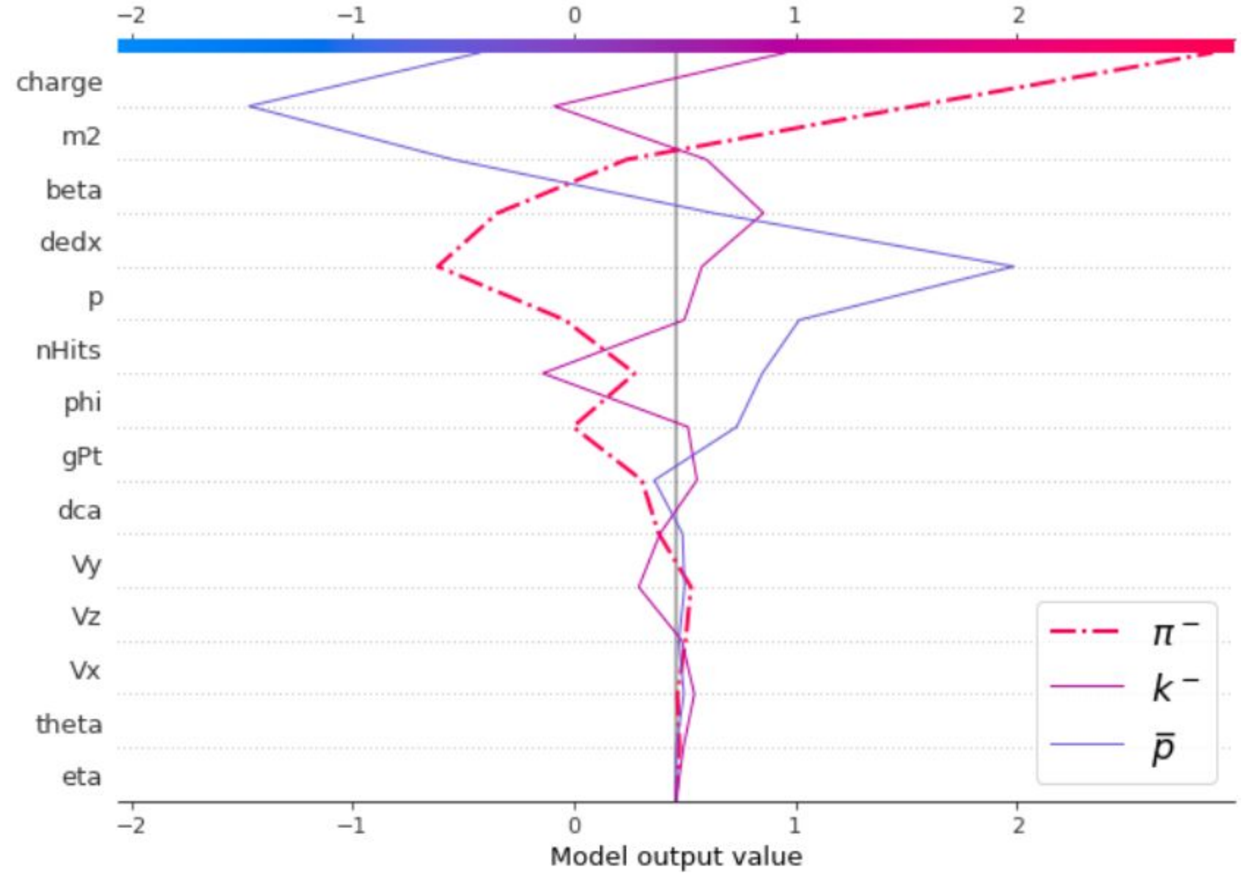
# Misclassification. Antiprotons



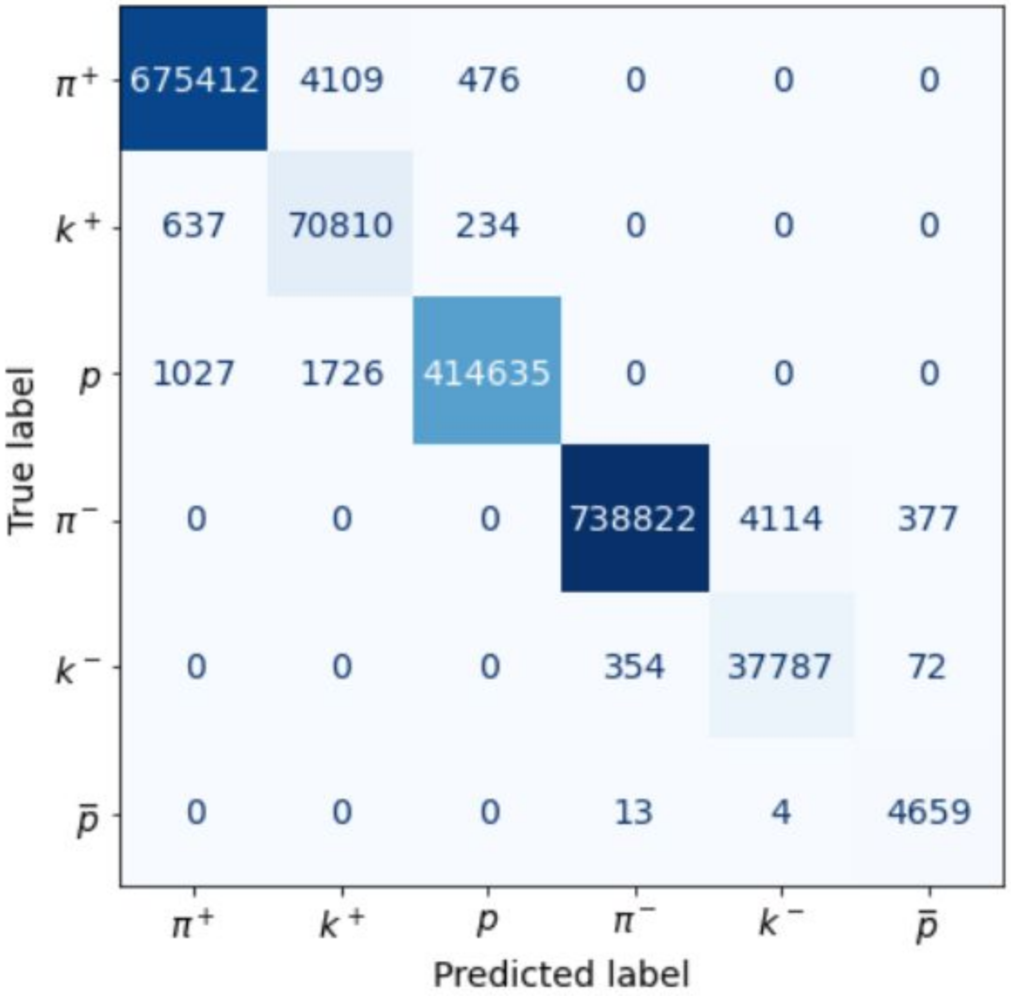



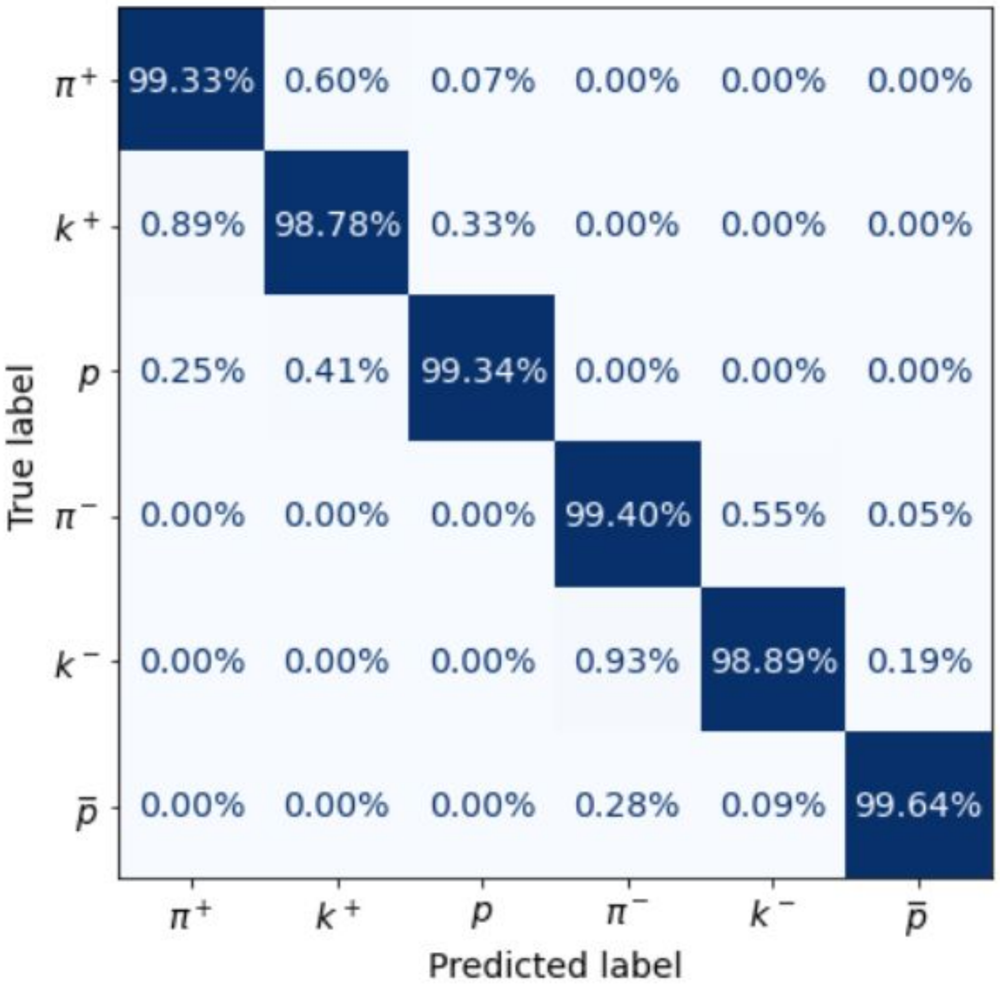
# Misclassification. Antiprotons

| | p | charge | dedx | m2 | nHits | eta | dca | Vx | Vy | Vz | phi | theta | gPt | beta |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **383509** | 1.51686 | -1 | 1.23853 | 0.015994 | 32 | -0.644238 | 0.088488 | 0.00004 | -0.024725 | 41.5421 | 2.29702 | 2.1746 | 1.24865 | 0.9973 |



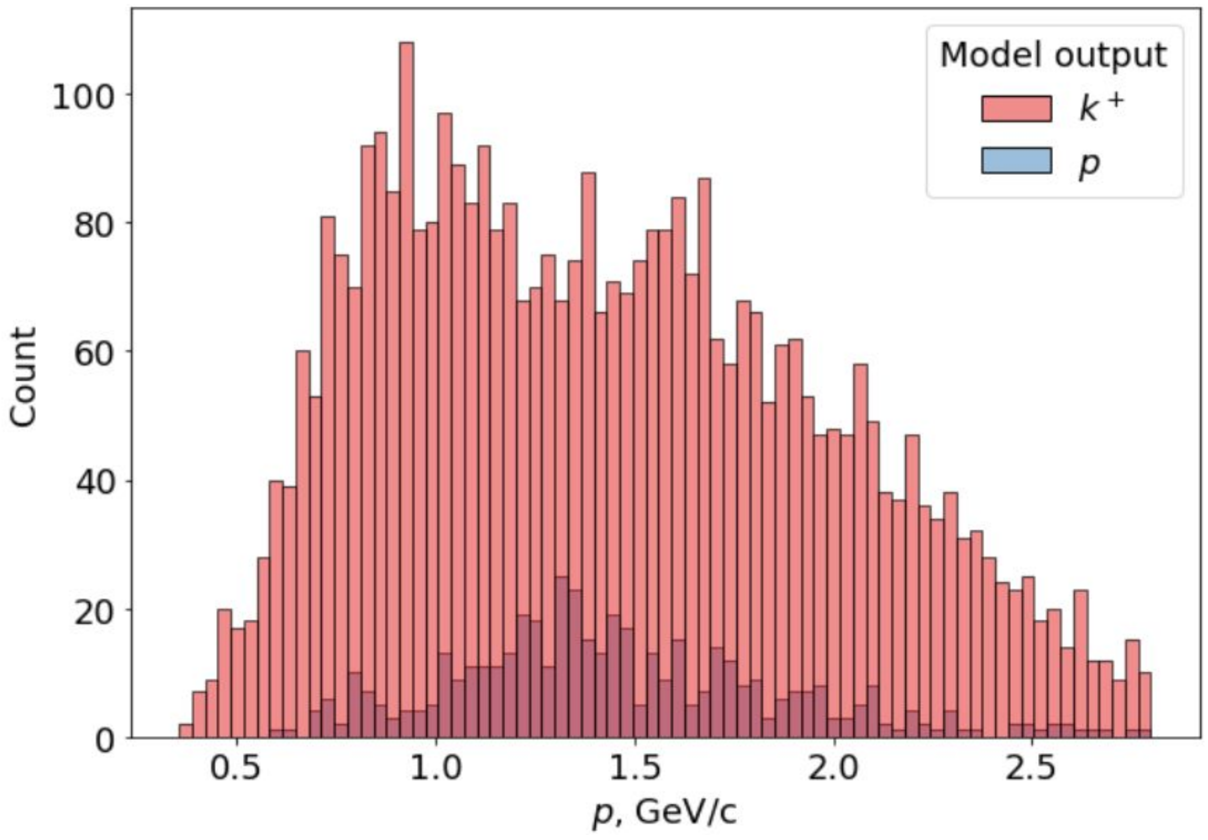
# Misclassification. Confusion Matrices



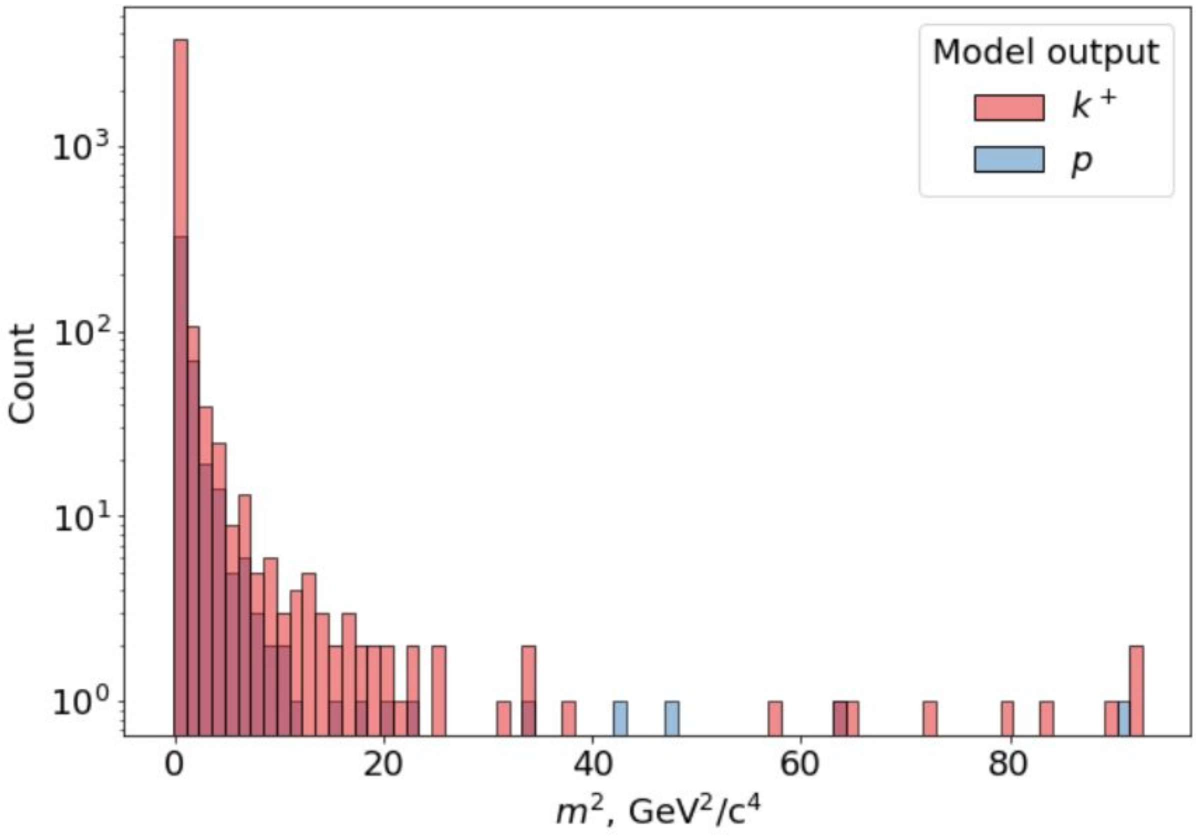



# Misclassification. Positive pions

$\pi^+$ errors

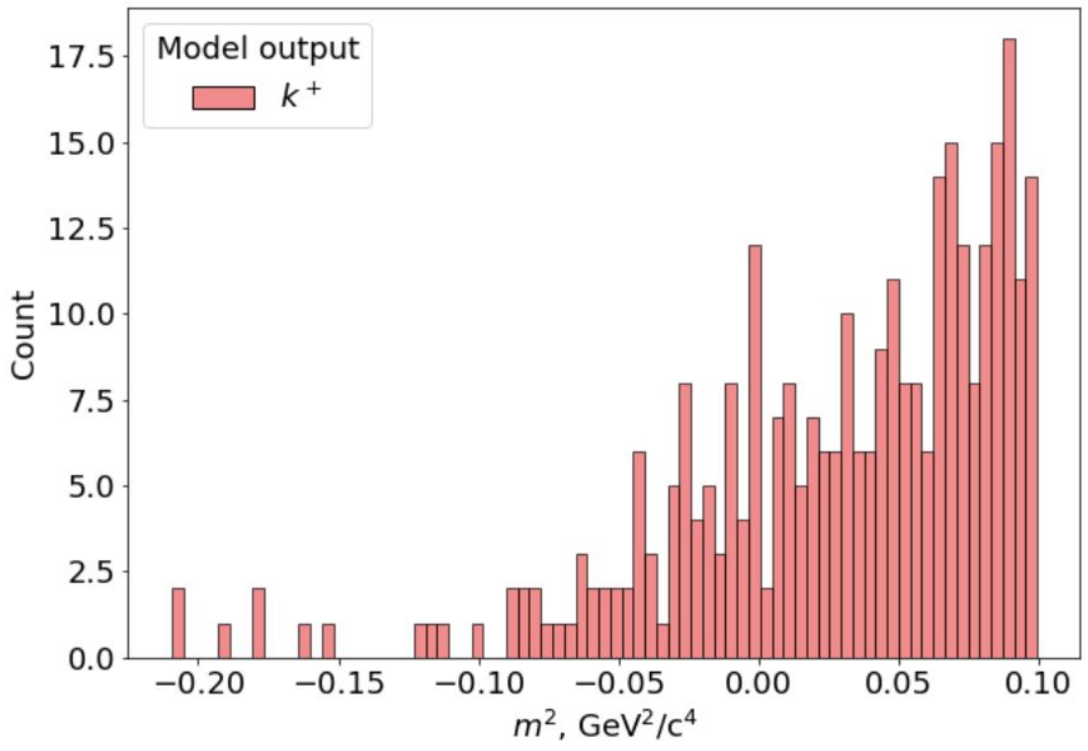


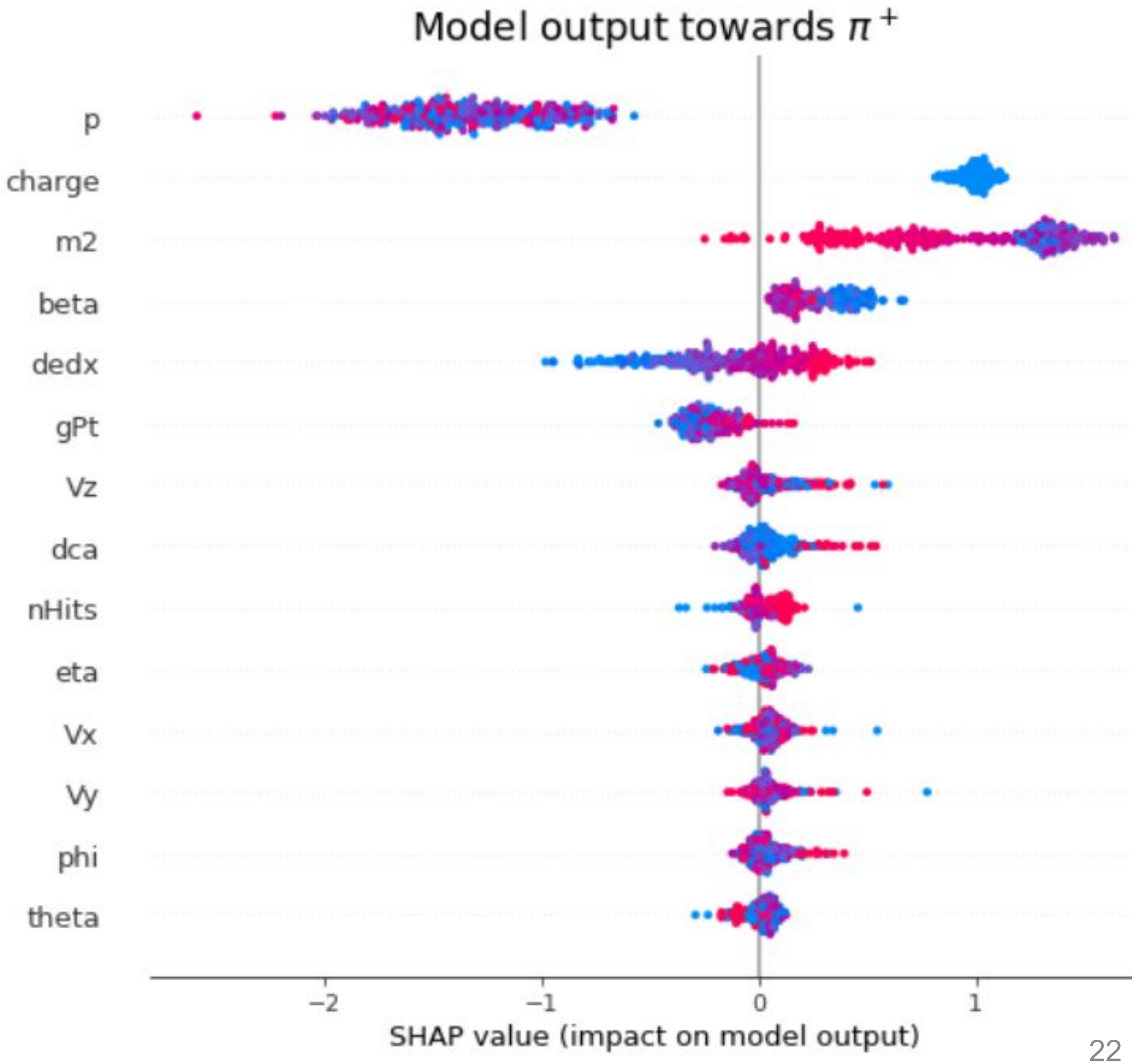

$\pi^+$ errors when 2.0 GeV/c < p < 2.8 GeV/c



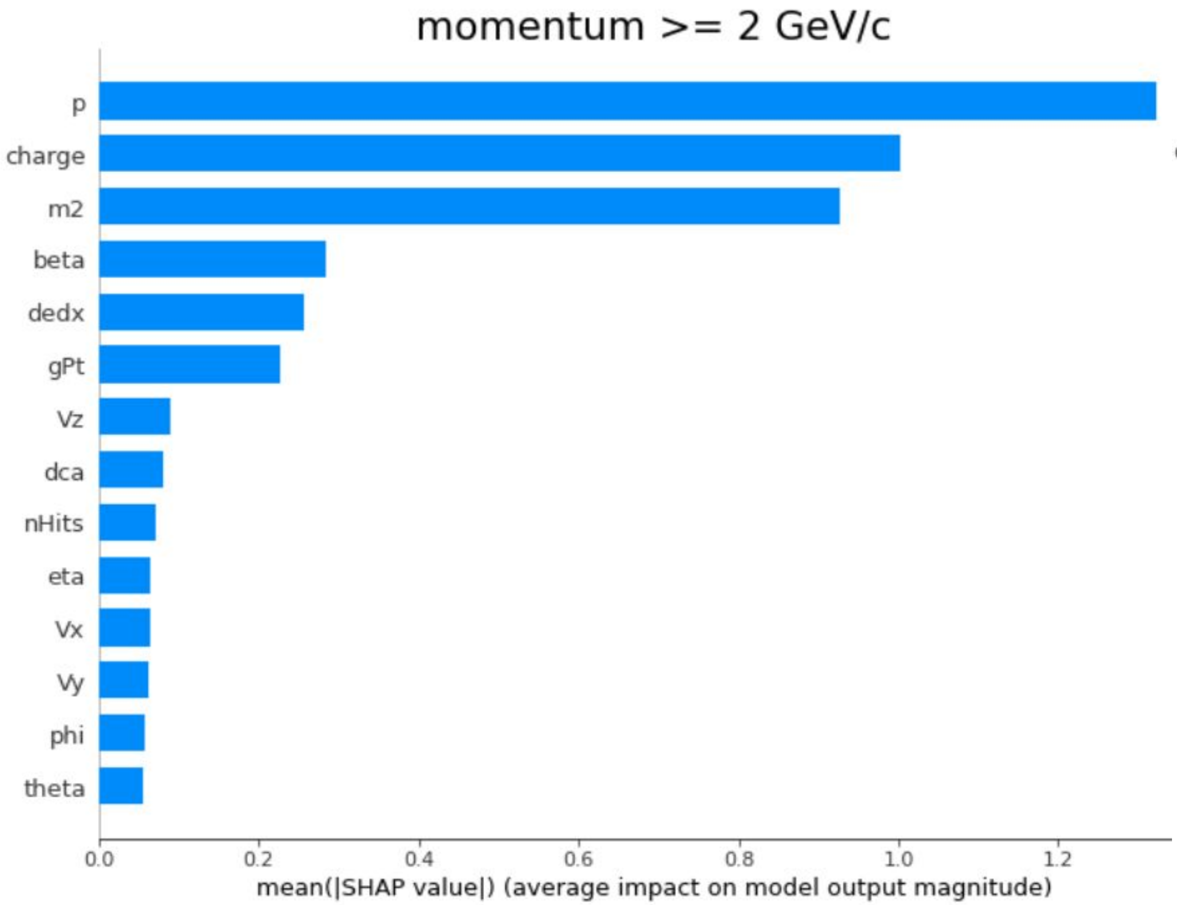
# Misclassification. Positive pions

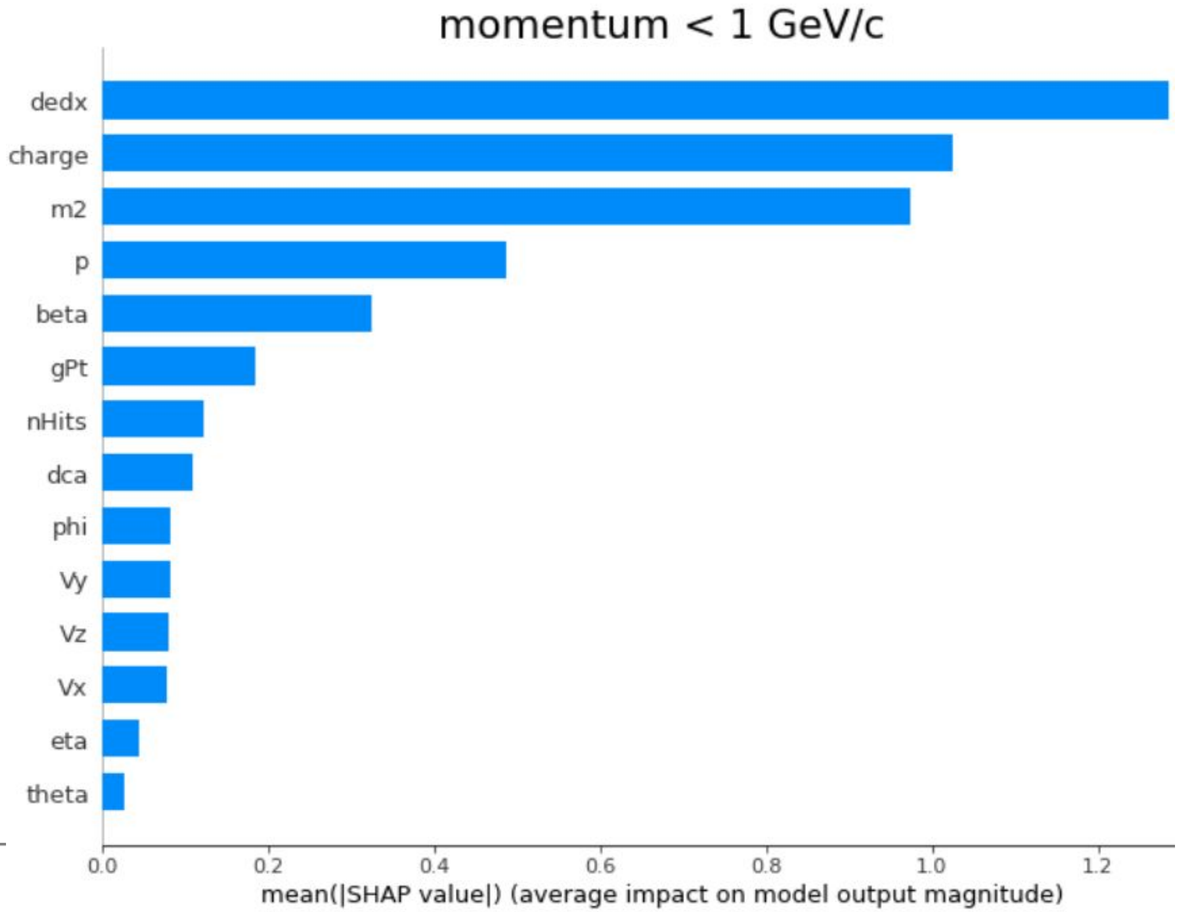
$\pi^+$ errors when 2.0 GeV/c < p < 2.8 GeV/c



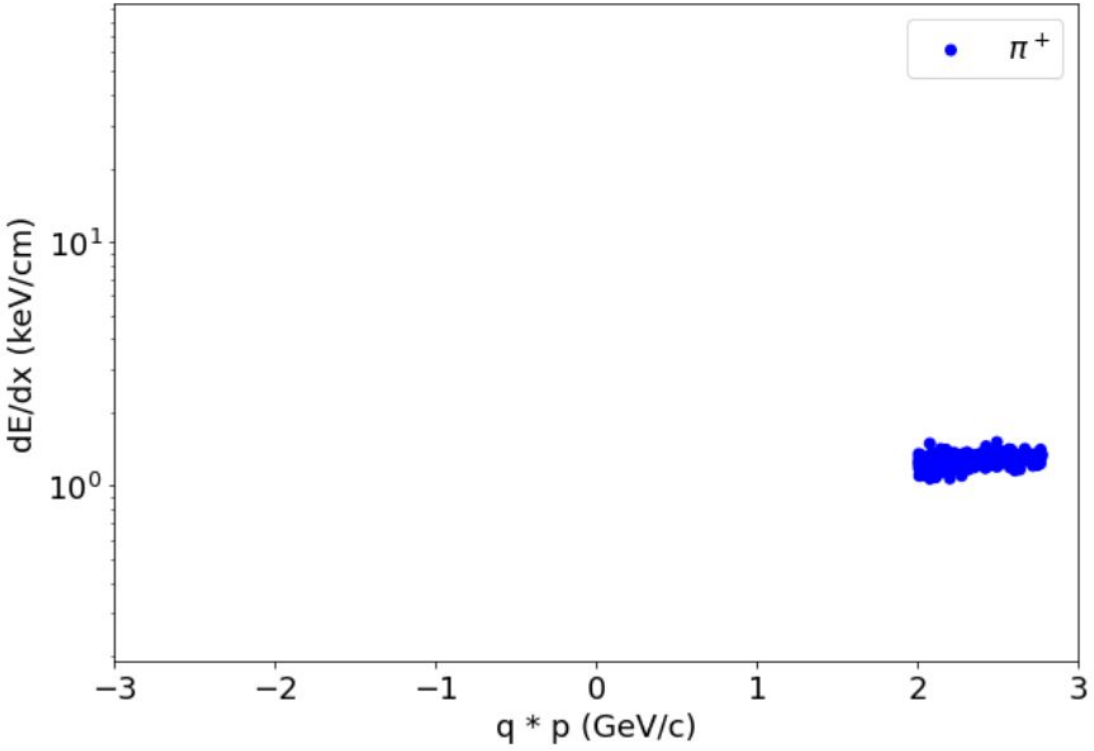
# Misclassification. Positive pions

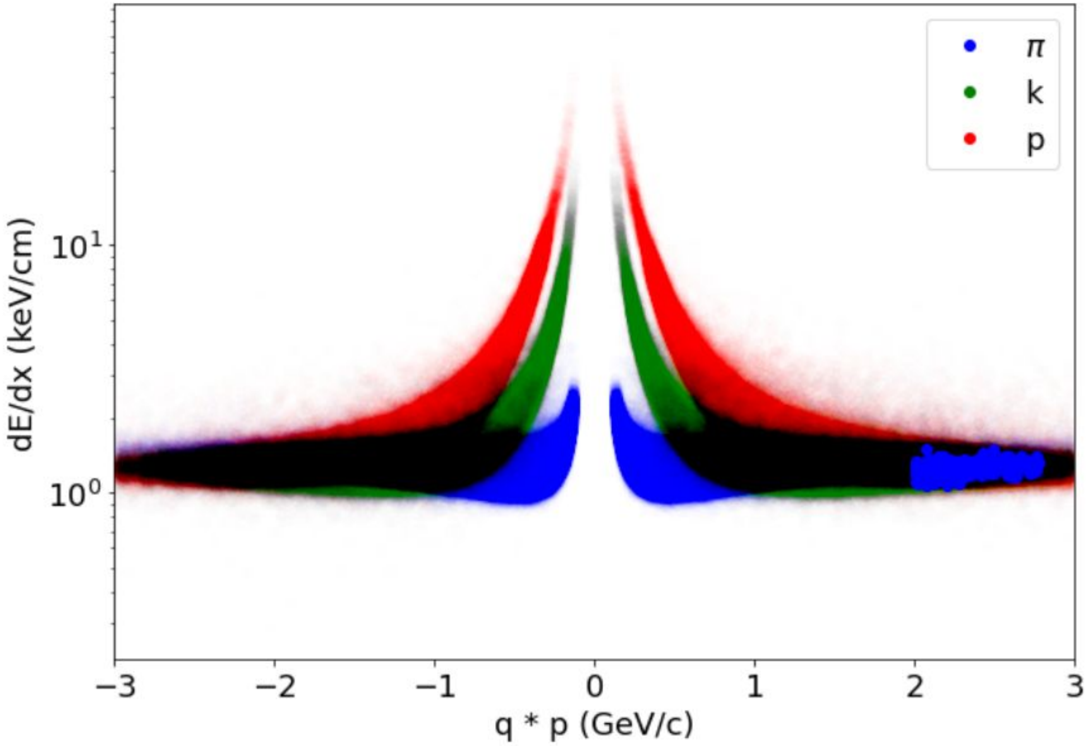


# Misclassification. Positive pions

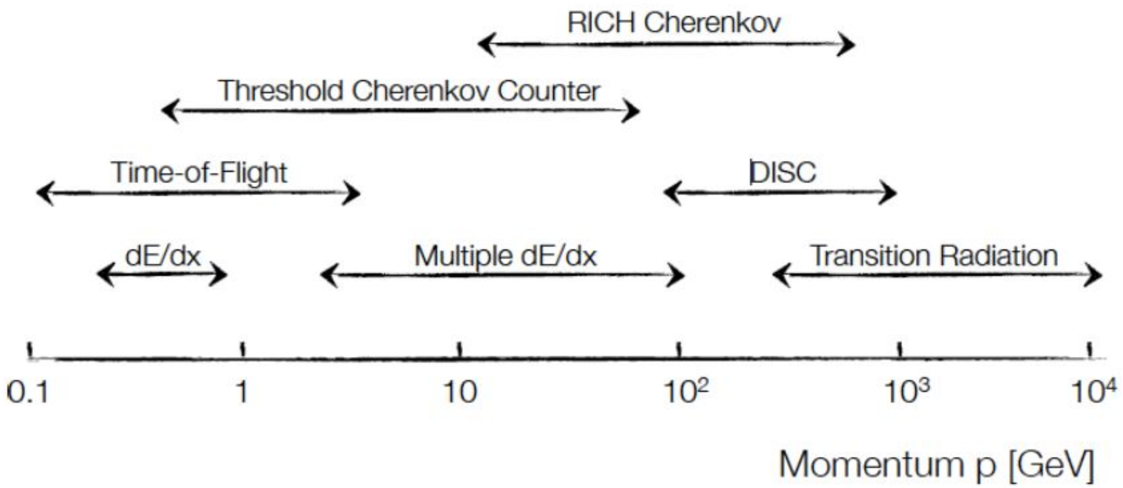
$\pi^+$ errors when 2.0 GeV/c < p < 2.8 GeV/c



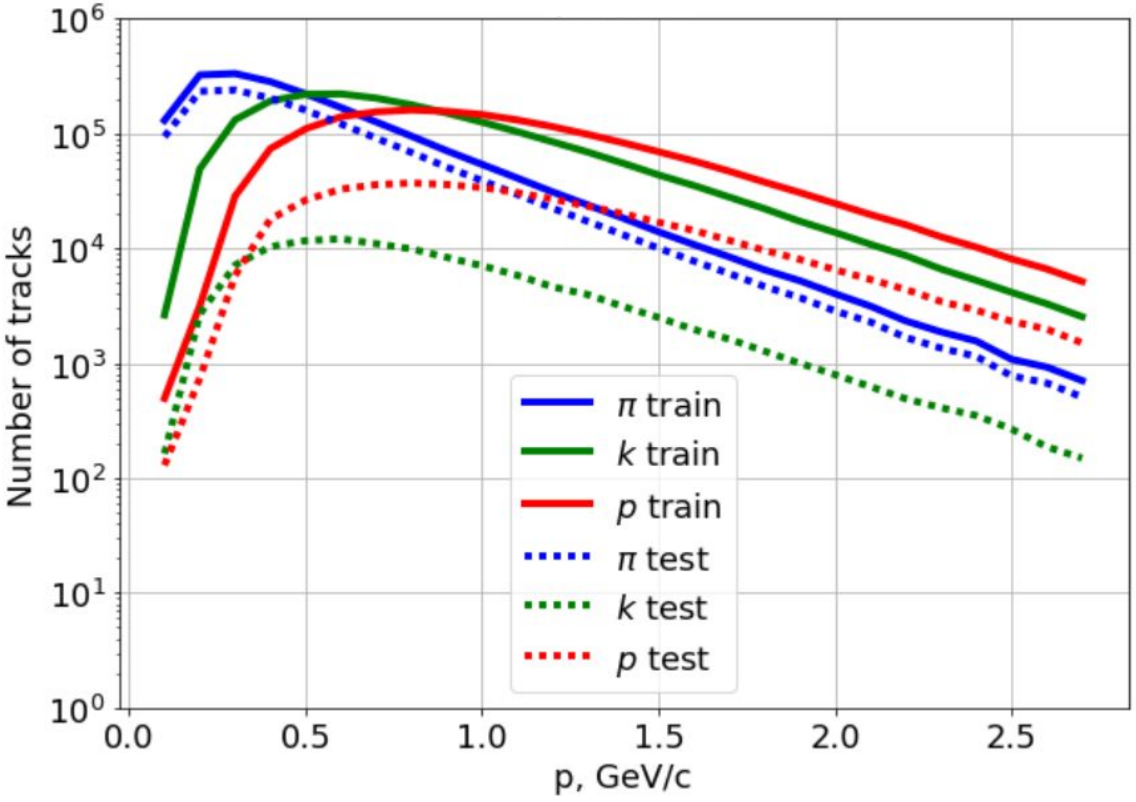



# Global PID and Class imbalance

Separation with different PID methods



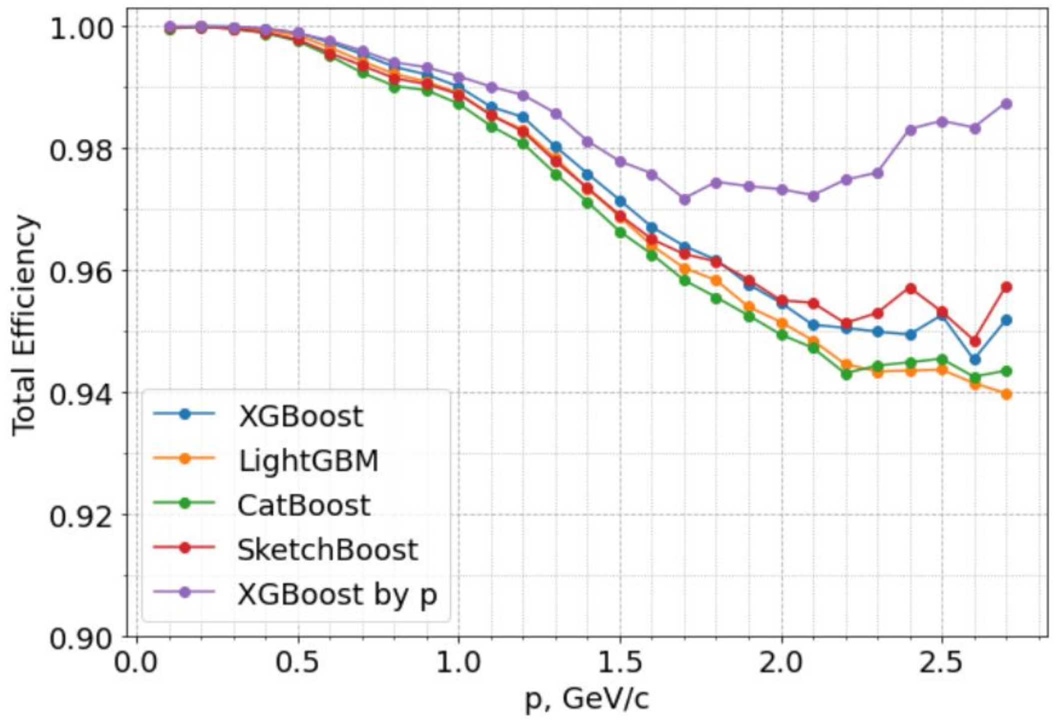

https://www.desy.de/~garutti/LECTURES/ParticleDetectorSS12/L12_PID.pdf

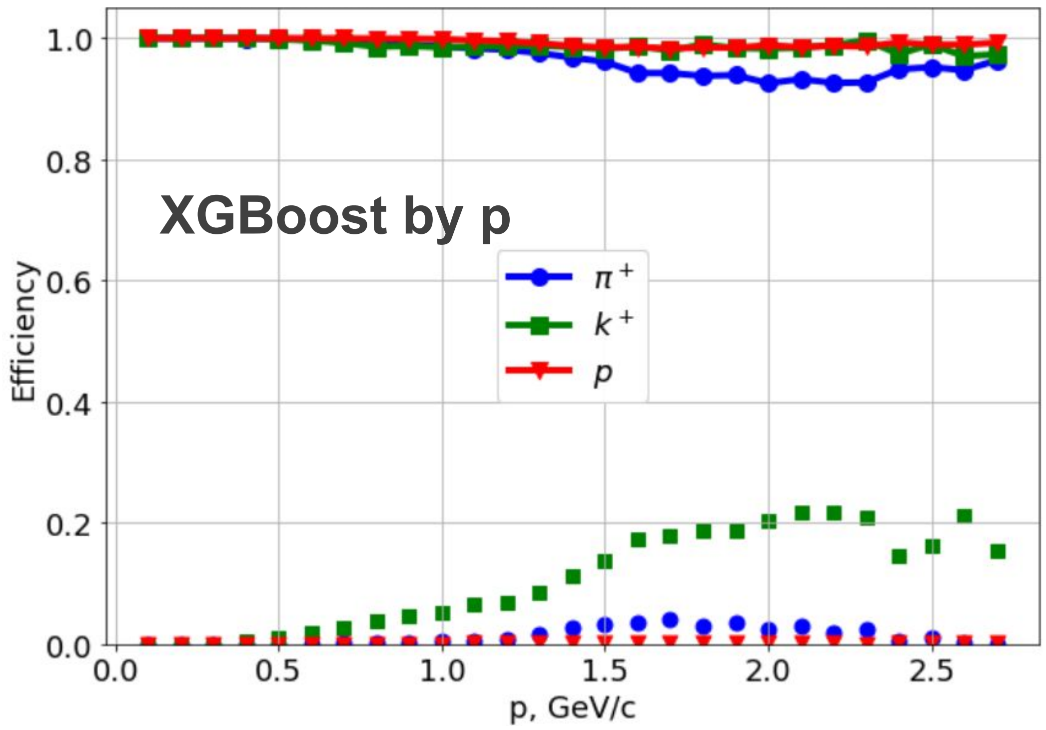



- The statistic is changing with a growing momentum
- It was being taken into account by the model
- Making the move from Global PID to Local

# XGBoost. Local models

The comparison of Local approach with Global models

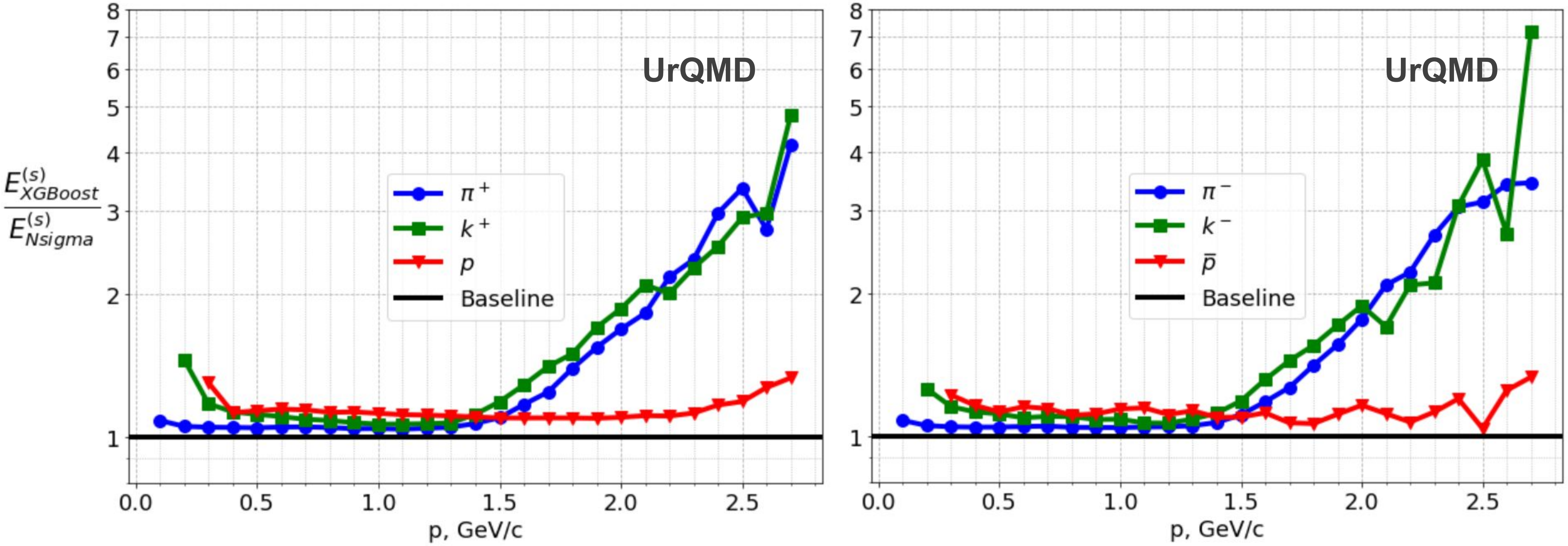



- What are more effective way to split momentum?
- Are the additional computational costs justified?

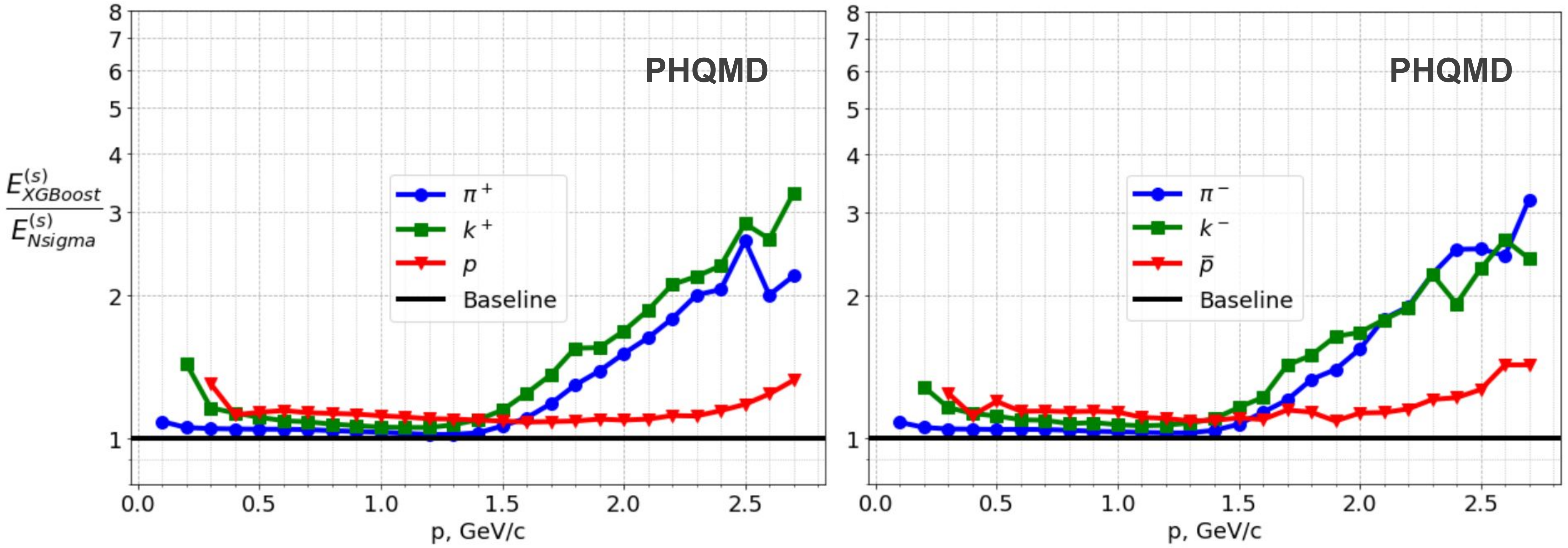
$$E = 0.9952$$

# Comparison with N-sigma




Efficiency ratio of XGBoost and n-sigma method

# Comparison with N-sigma




Efficiency ratio of XGBoost and n-sigma method

# List of papers


Ayriyan A., Grigorian H., Papoyan V. Sampling of Integrand for Integration Using Shallow Neural Network. *Discrete & Continuous Models & Applied Computational Science*. **2024** (accepted).

Papoyan V., Gori G., Papoyan V. (Jr.), Trombettoni A., Ananikian N. Logarithmic negativity of the 1D antiferromagnetic spin-1 Heisenberg model with single-ion anisotropy. *Physica E: Low-dimensional Systems and Nanostructures*. **2024**, 158, 115899.

Papoyan V., Aparin A., Ayriyan A., Grigorian H., Korobitsin A., and Mudrokh A. Machine Learning Application for Particle Identification in MPD. *Physics of atomic nuclei*. **2023**, 86, 5, 869-873.

Kadochnikov I., Papoyan V. Blocking Strategies to Accelerate Record Matching for Big Data Integration. *CEUR Workshop Proceedings*. **2019**, 2507. 219-224.

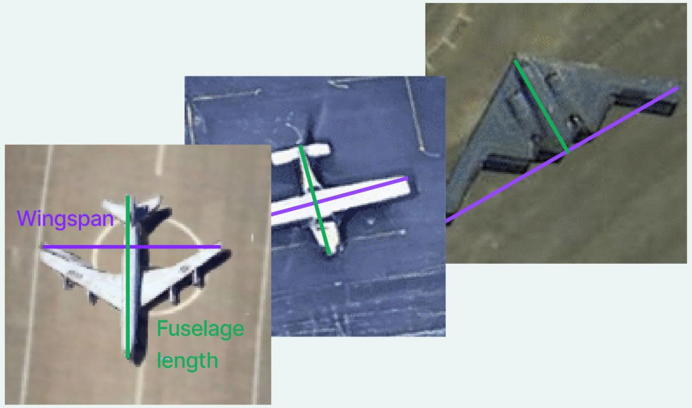
Папоян В., Кадочников И., Кореньков В. Связывание текстовых записей в задаче интеграции данных в условиях больших данных. *Системный анализ в науке и образовании*. **2019**, 3. 71-78.

Папоян В., Кадочников И., Кореньков В. Применение технологий больших данных для организации сбора, потоковой обработки и хранения информации о компаниях-нерезидентах. *Системный анализ в науке и образовании*. **2019**, 3. 65-70.

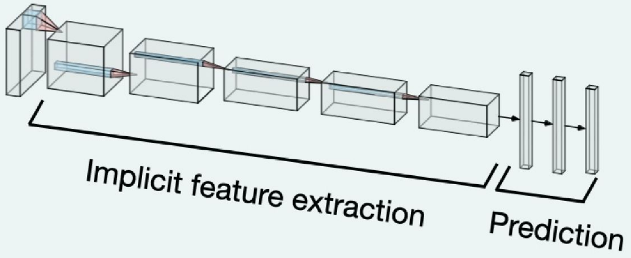
# Backup

# Tabular Data: Deep Learning vs Gradient Boosting

## Unstructured data



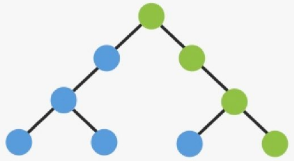

Deep learning




## 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*

# 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

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

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

# 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) |

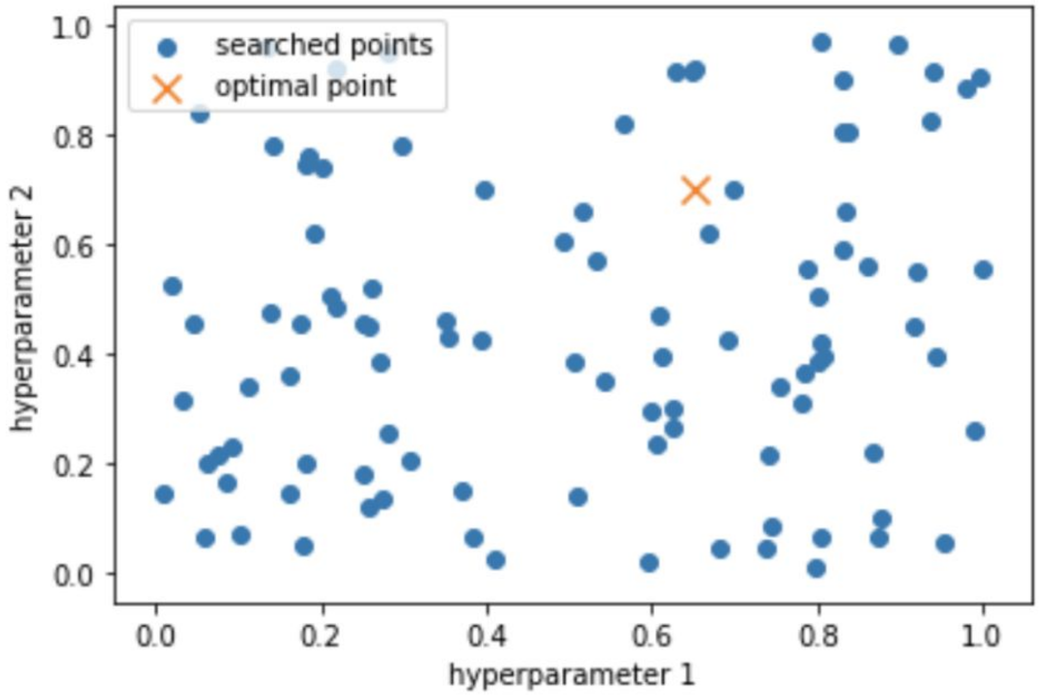
| feature | values range |
|---|---|
| Vx | (-0.106, 0.106) |
| Vy | (-0.103, 0.112) |
| Vz | (-50, 54.1) |
| phi | (-3.1415, 3.1415) |
| theta | (0.53, 2.61) |
| gPt | (0.106, 98) |
| beta | [0.012, 1.564] |

# Hyperparameters tuning

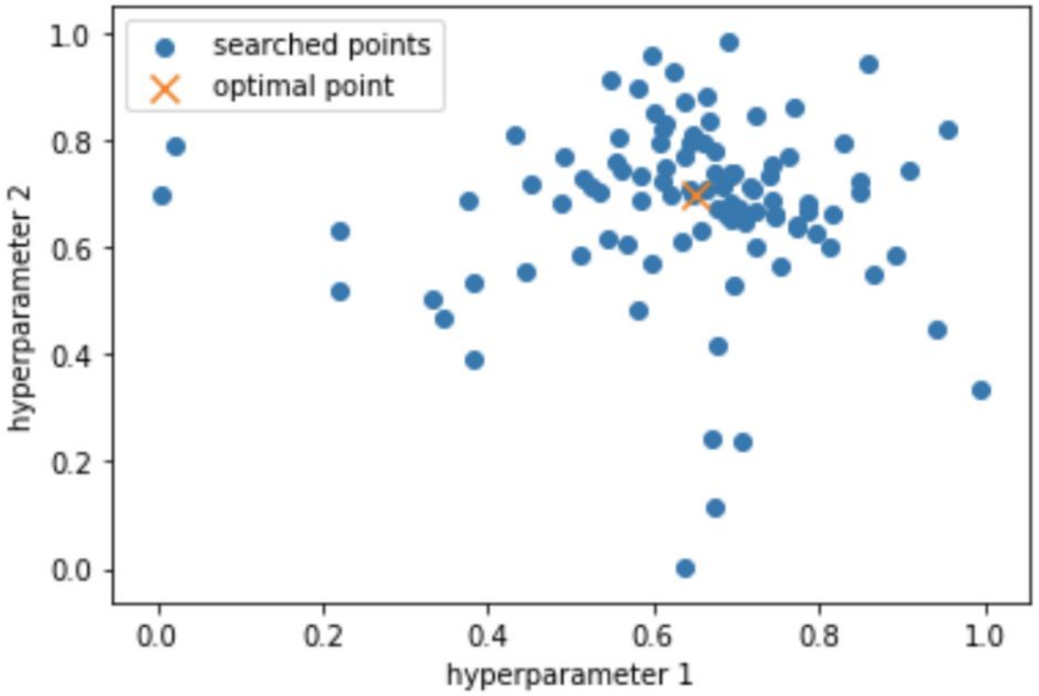
Tree-structured Parzen Estimator (TPE) was used to find the optimal hyperparameters;

TPE is a form of Bayesian Optimization.

Random search

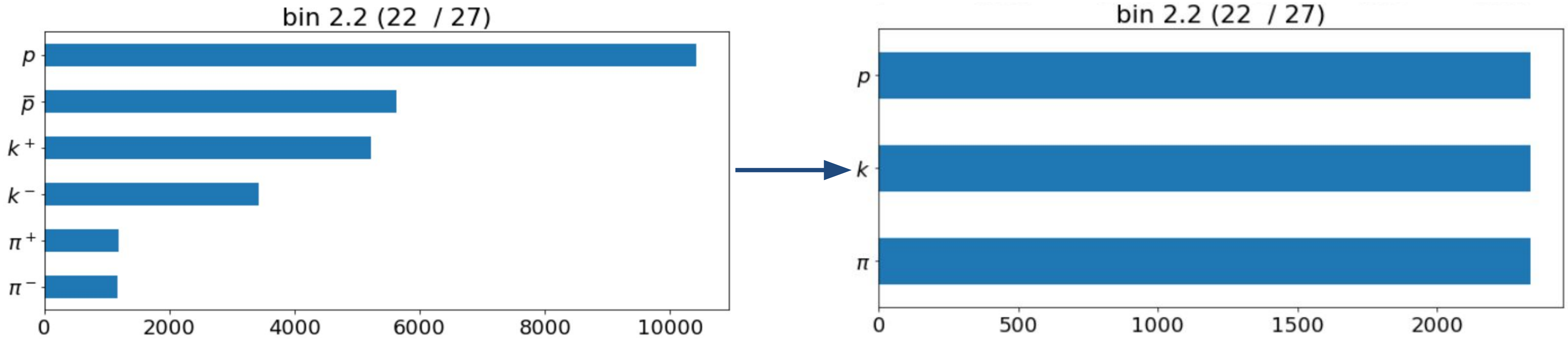



TPE search



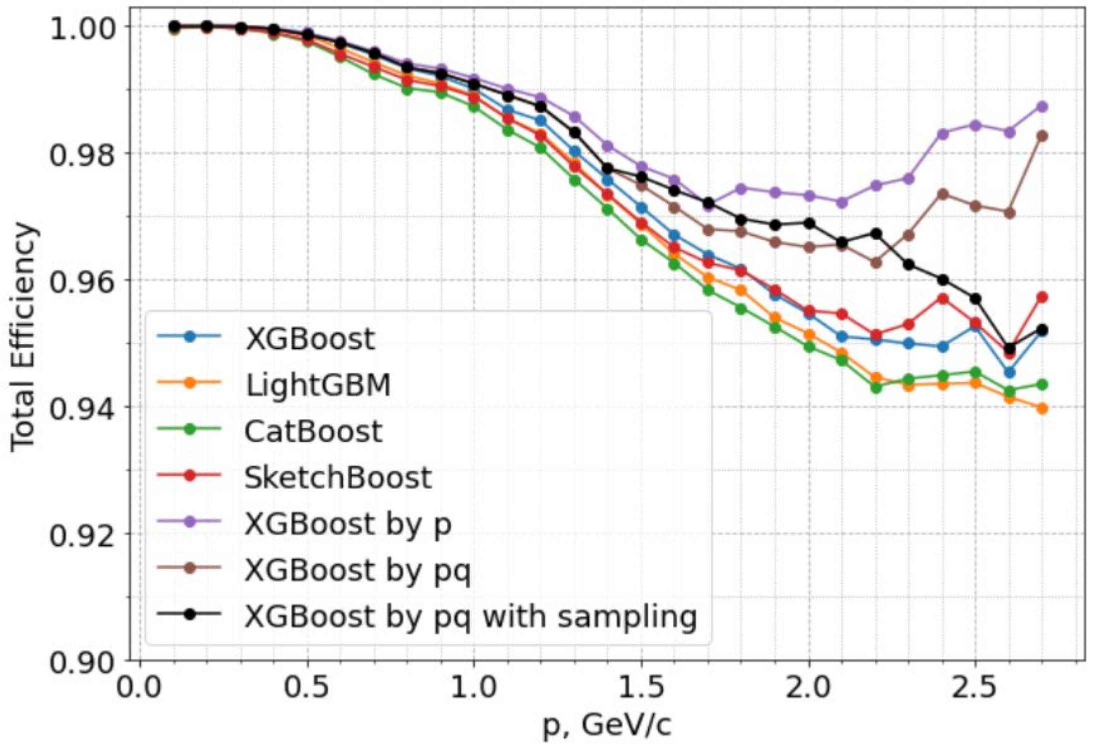

https://miro.medium.com/max/4800/1*tYWqO5BwNDVaM3kP3w1IAg.png

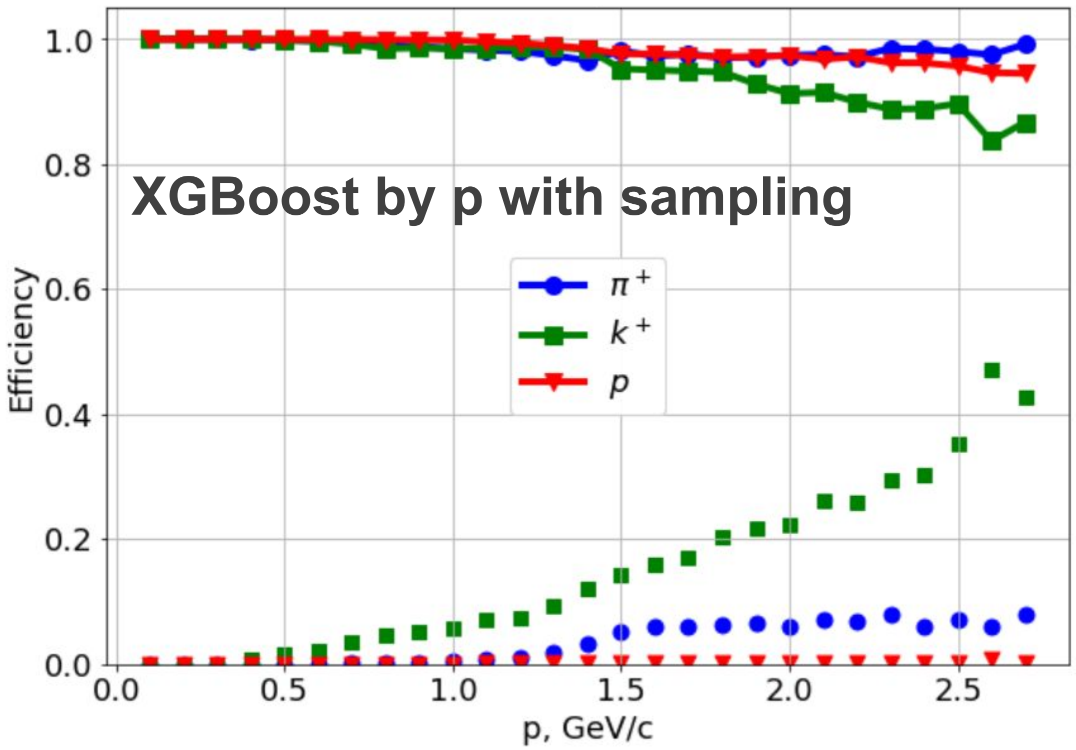
# Oversampling and undersampling



# Class imbalance. Undersampling

1. Particles and corresponding antiparticles are combined to increase the number of examples in the minority class
2. Undersampling: randomly delete examples in the majority class (protons and kaons)



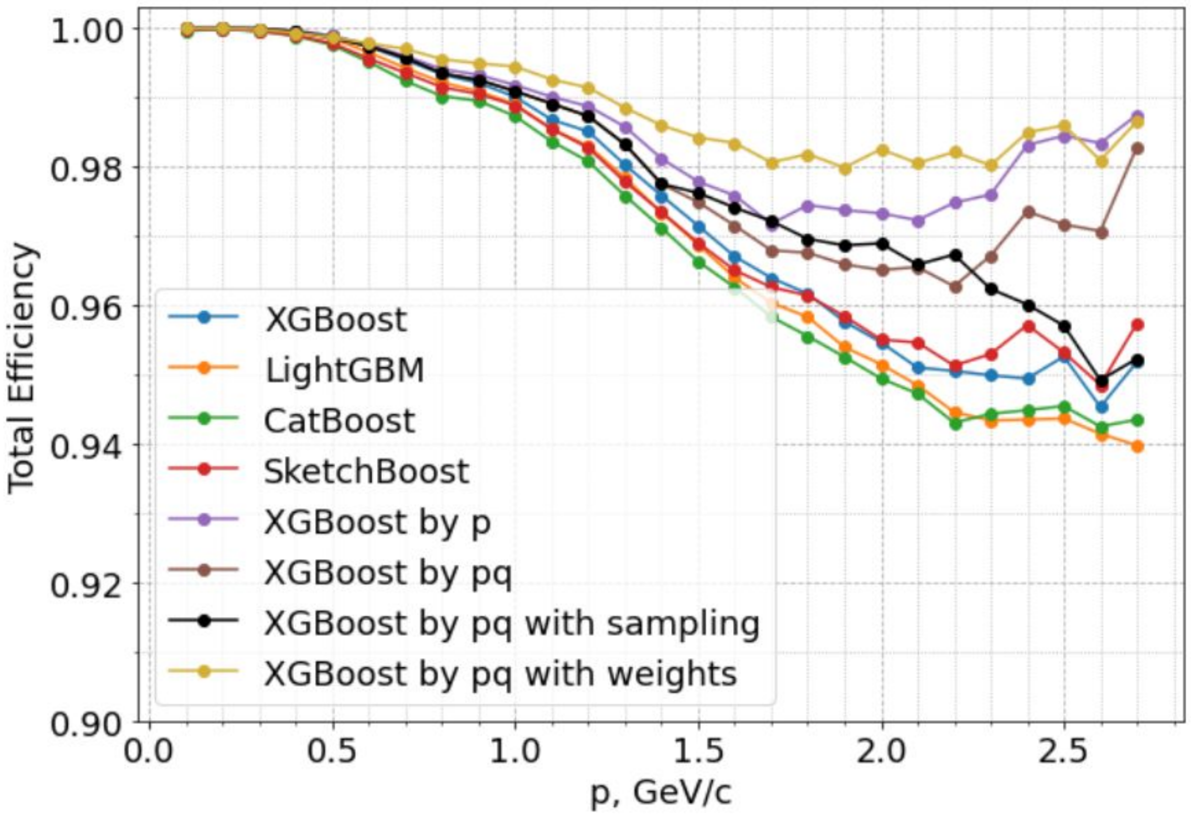
# Sample weights

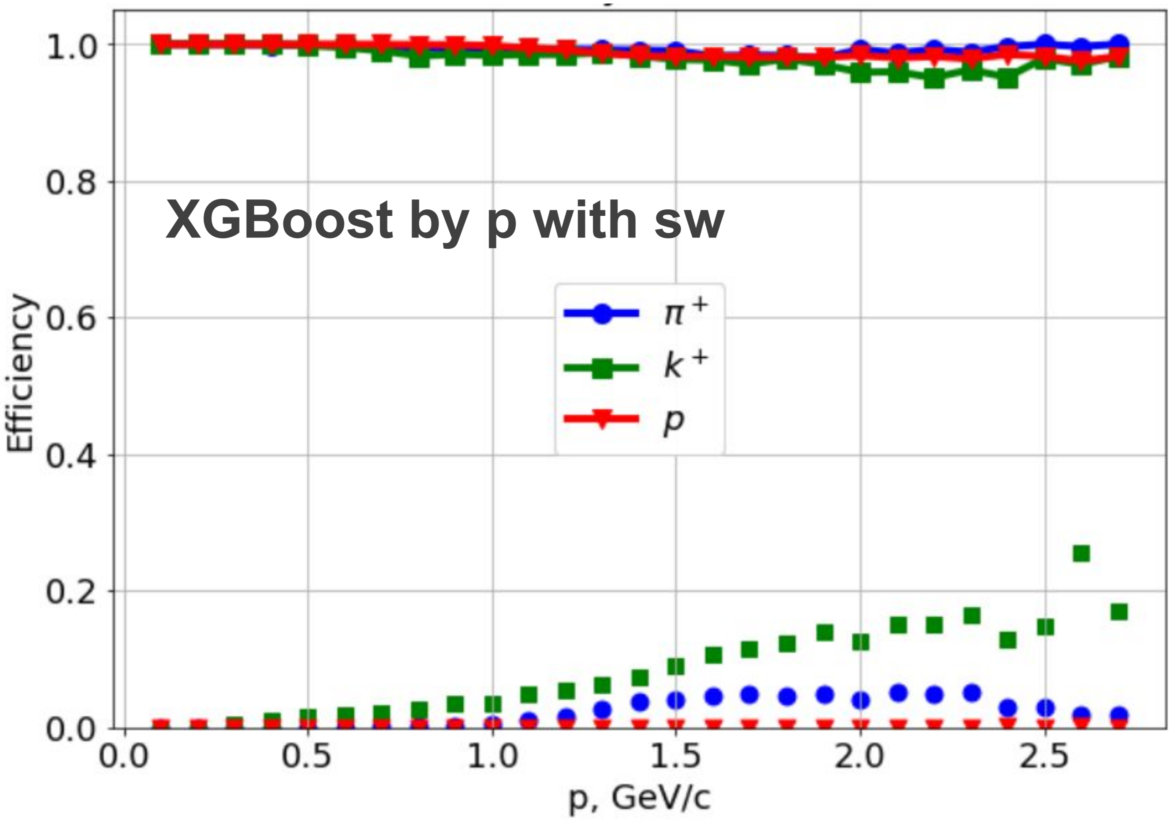
The idea is to weigh the cost function computed for different tracks differently based on whether they belong to the **majority** or the **minority** classes

$$L = \frac{-\sum_{i=1}^{N} w_i(c_i log(p_i) + (1 - c_i) log(1 - p_i))}{\sum_{i=1}^{N} w_i}$$

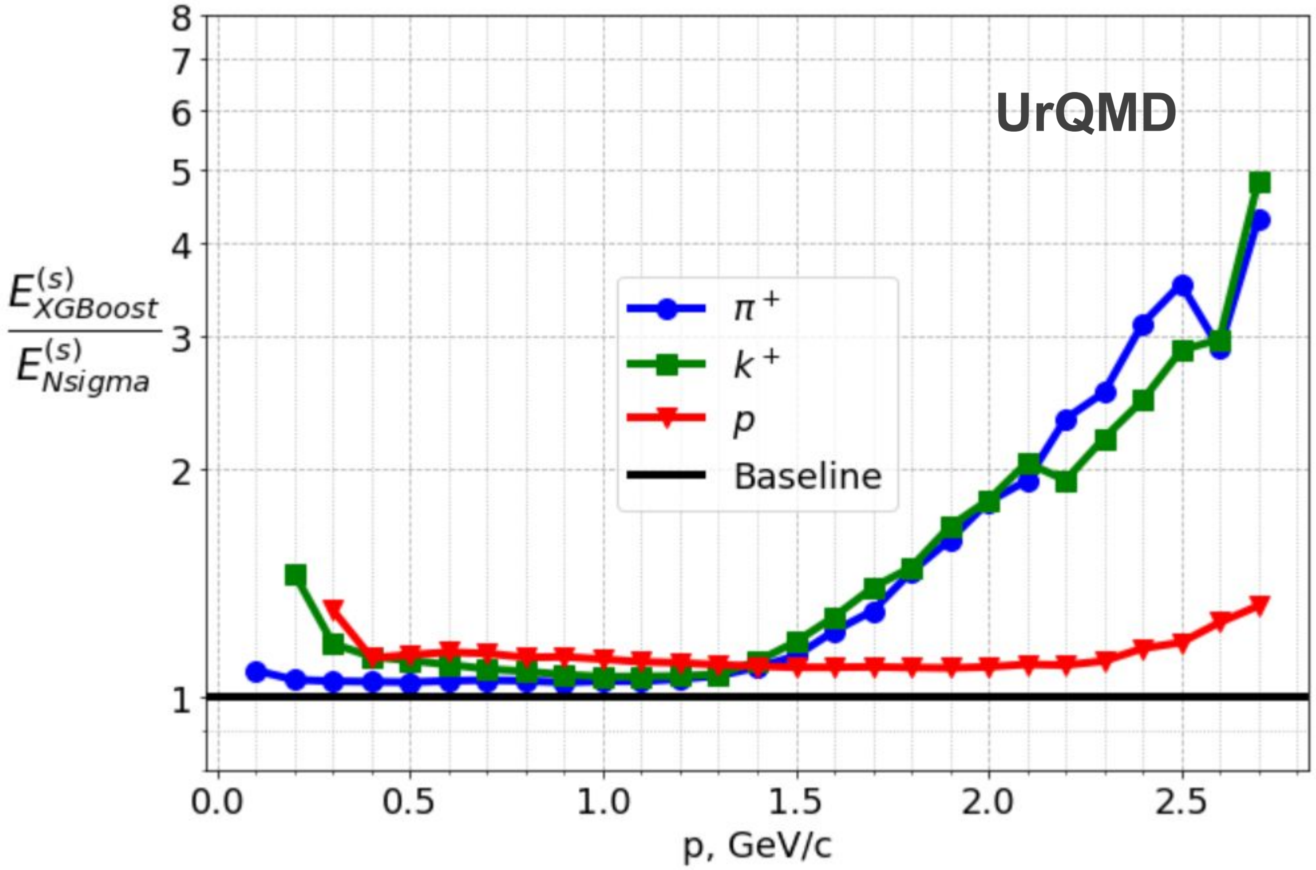
$$w_i^{(j)} = \begin{cases} \frac{M^{(j)}}{N_\pi} \text{if } i \in \boldsymbol{\pi} \\ \frac{M^{(j}}{N_k} \text{if } i \in \mathbf{K} \\ \frac{M^{(j)}}{N_p} \text{if } i \in \mathbf{p} \end{cases} \quad M^{(j)} = \max_s(N_s^{(j)})$$

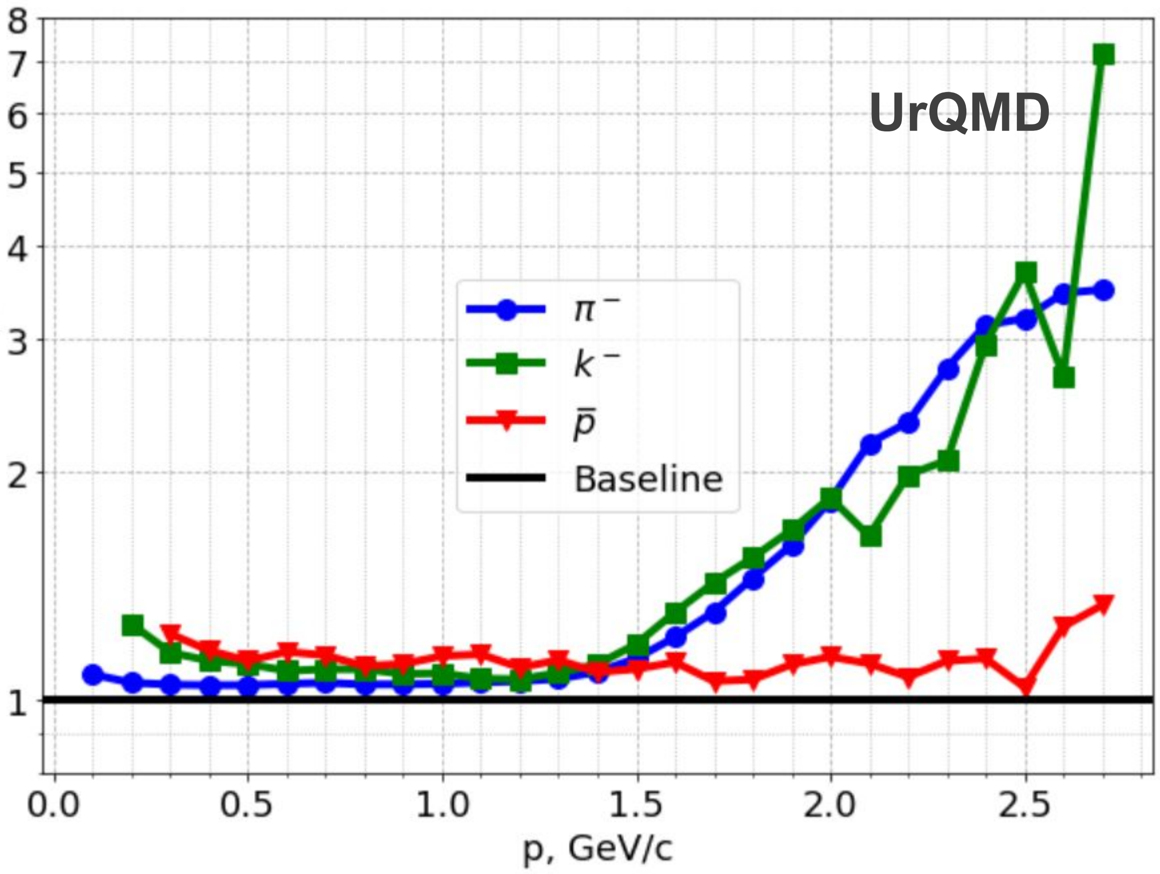
# Sample weights



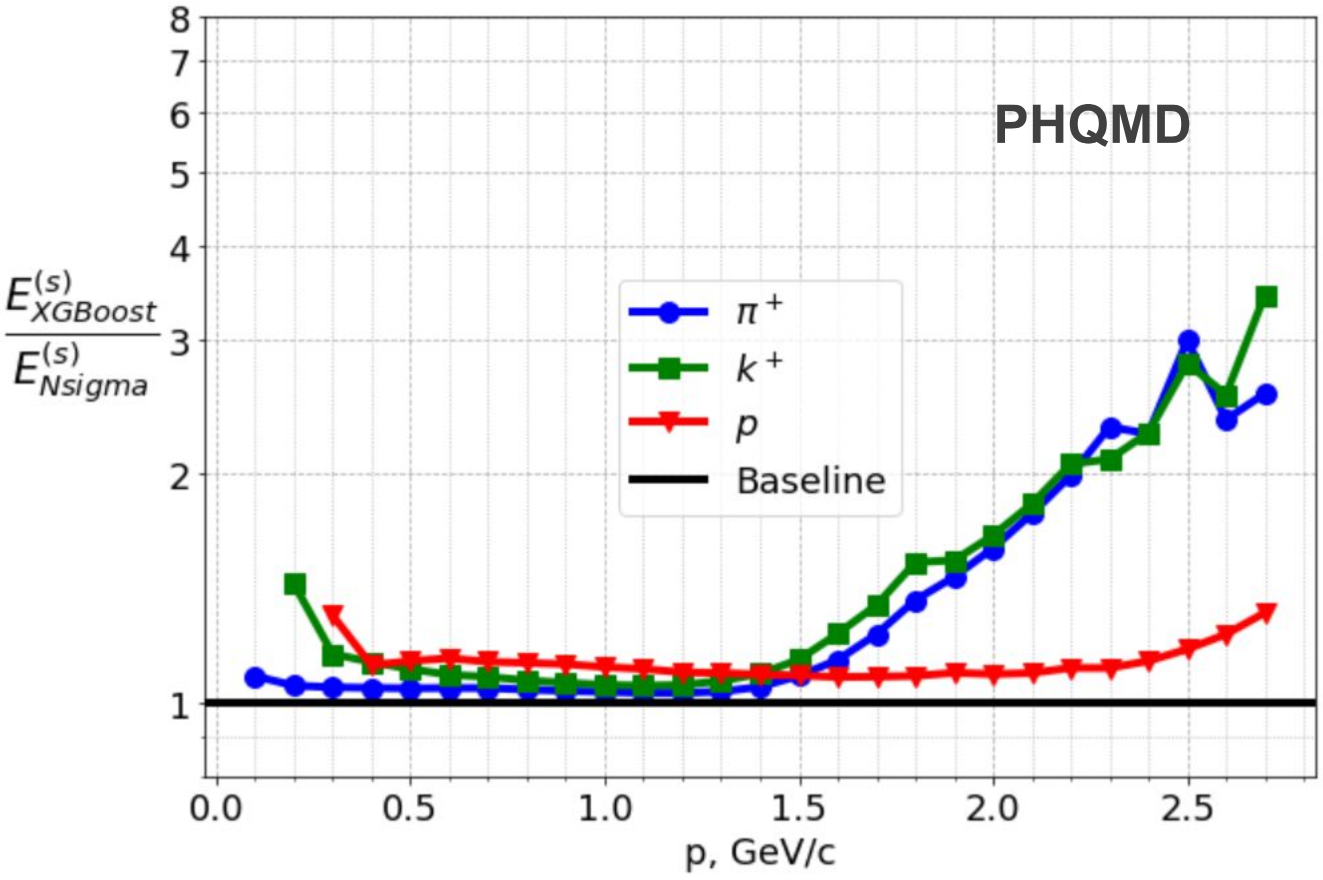
# Comparison with N-sigma

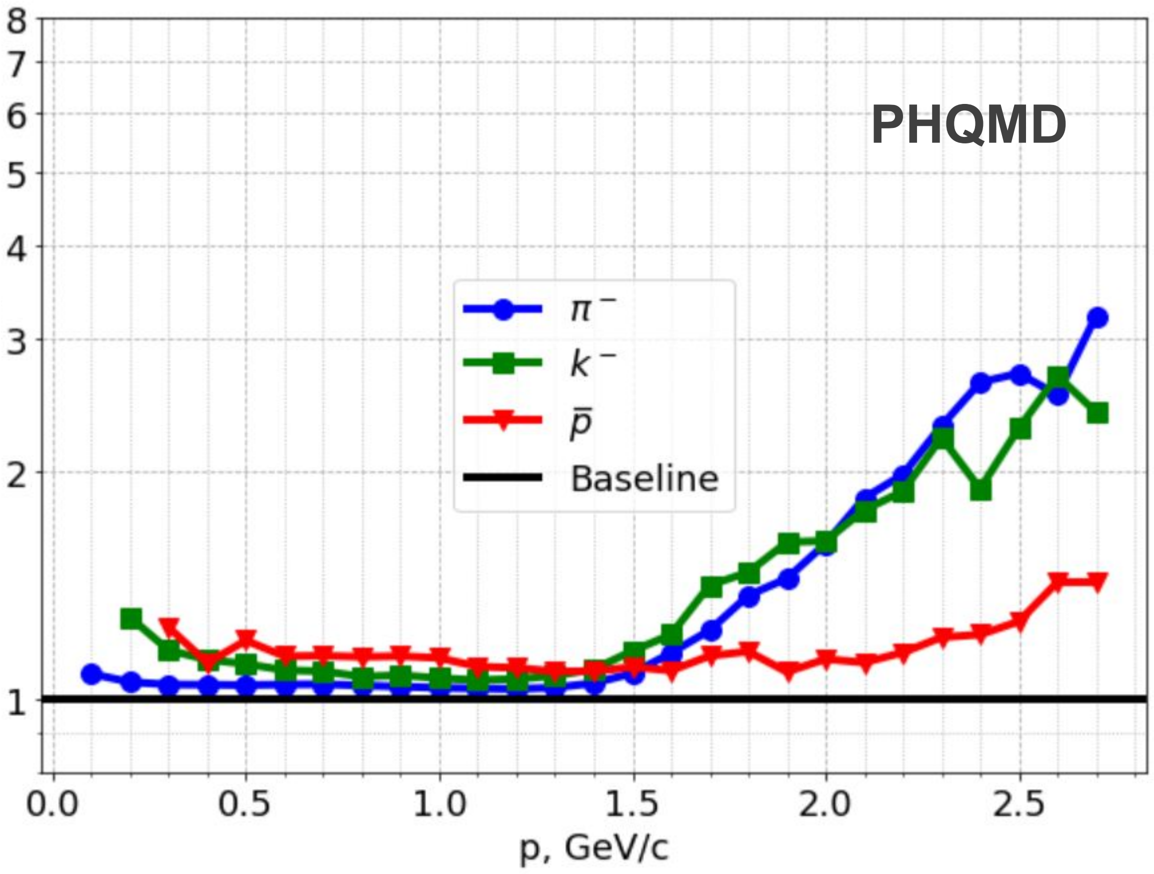



Efficiency ratio of XGBoost and n-sigma method

# Comparison with N-sigma




Efficiency ratio of XGBoost and n-sigma method