

# Gradient Boosted Decision Tree for Particle Identification in MPD

V. Papoyan<sup>1,3</sup>

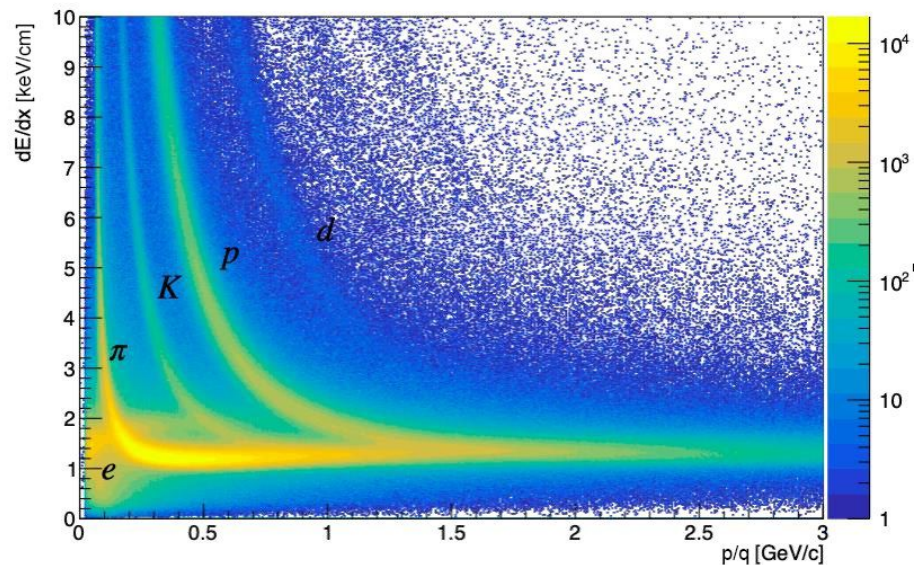
Coauthors: A. Aparin<sup>2</sup>, A. Ayriyan<sup>1,3</sup>, H. Grigorian<sup>1,3</sup>, A. Korobitsin<sup>2</sup>, A. Mudrokh<sup>2</sup>

<sup>1</sup>MLIT JINR, <sup>2</sup>VBLHEP JINR, <sup>3</sup>AANL (YerPhi)

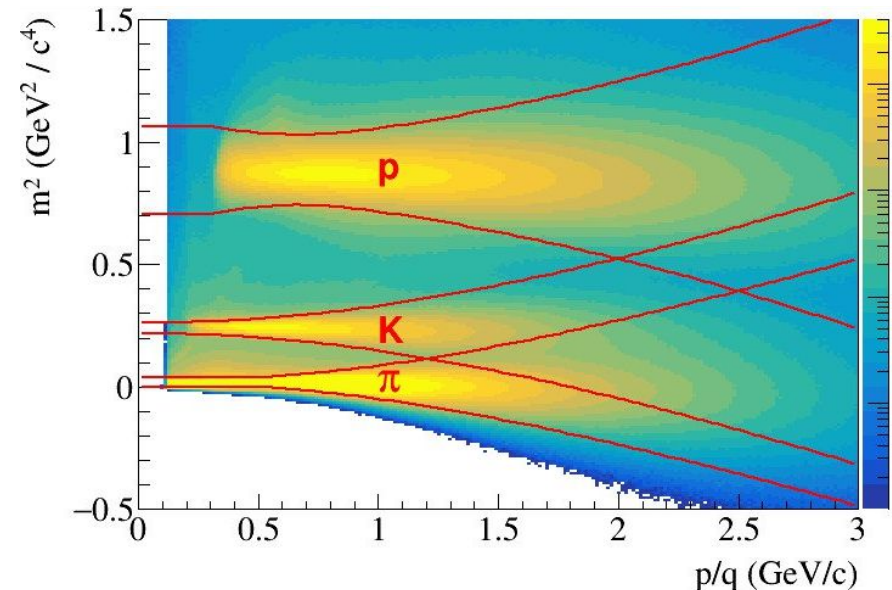
# Particle Identification in MPD experiment

MPD particle identification (PID) 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;

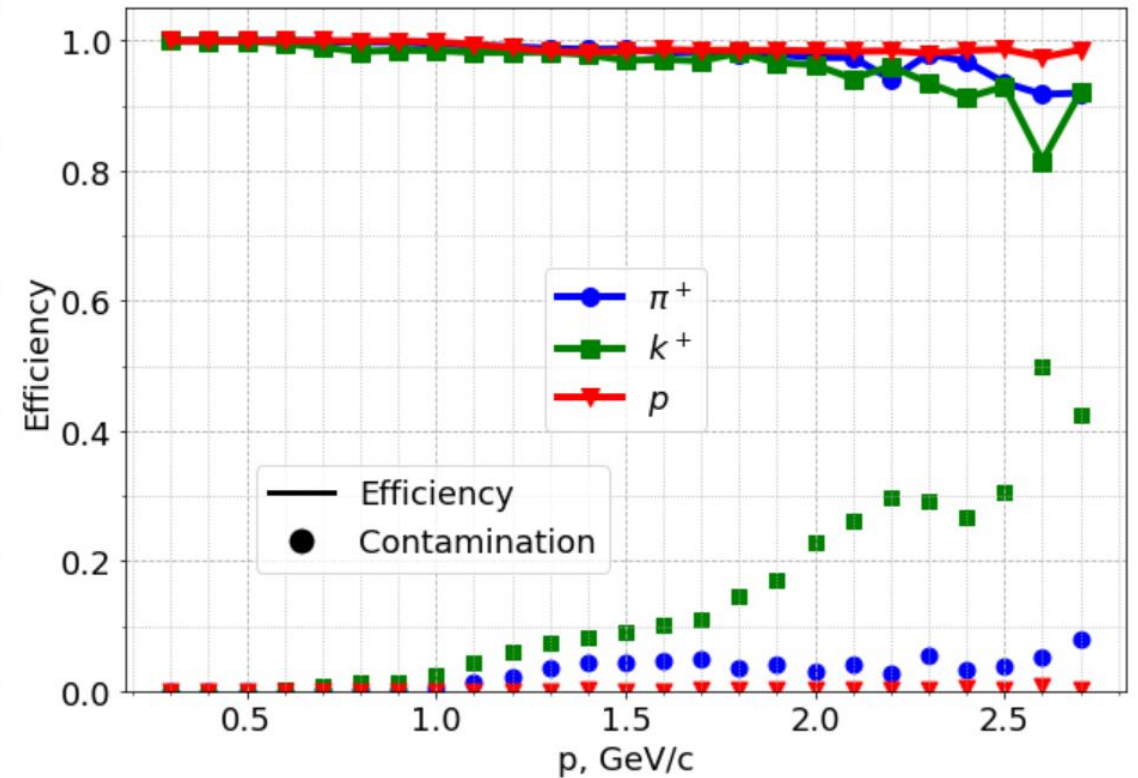
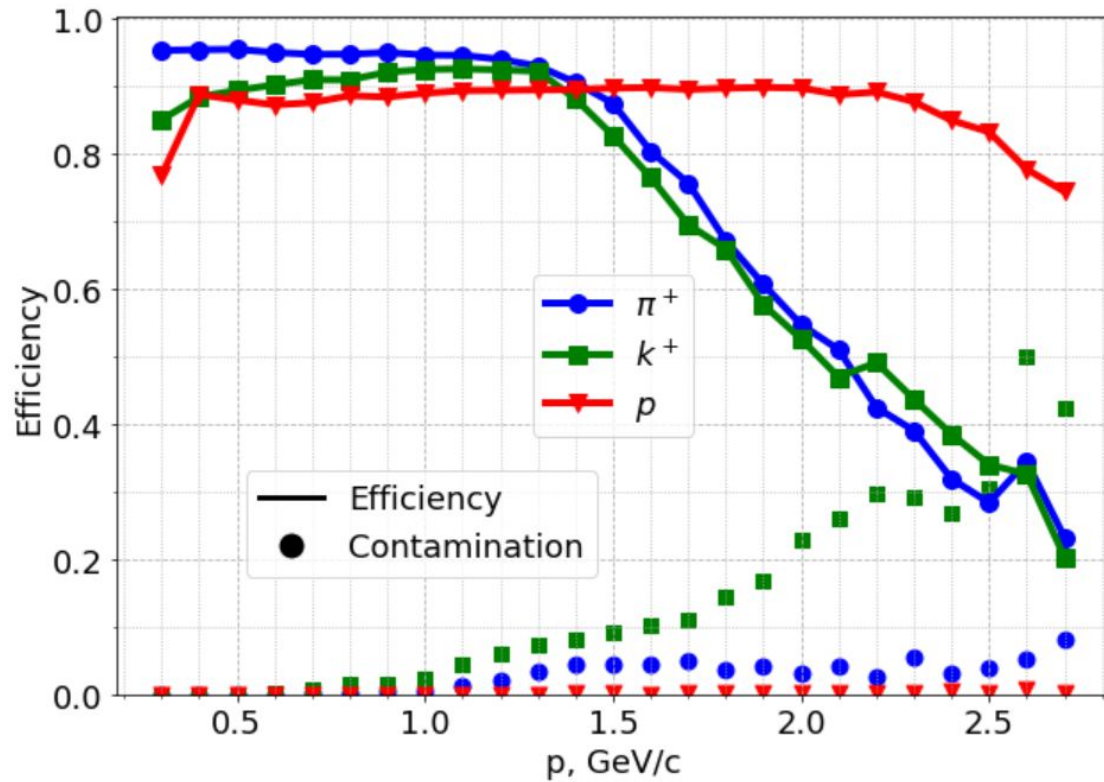


Knowing the particle **momentum** (from TPC) one obtains the **mass squared** and thus identity of the particle.

# Baseline PID in MPD - N-sigma

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

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

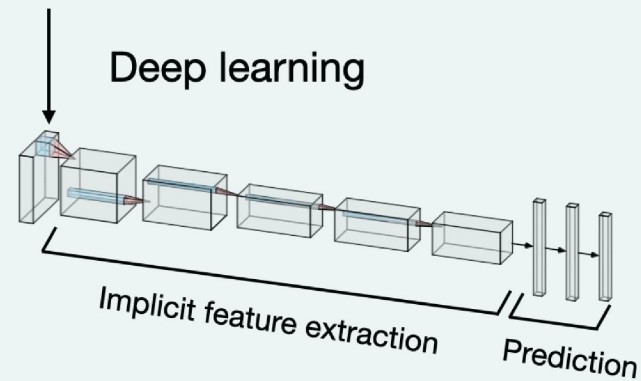
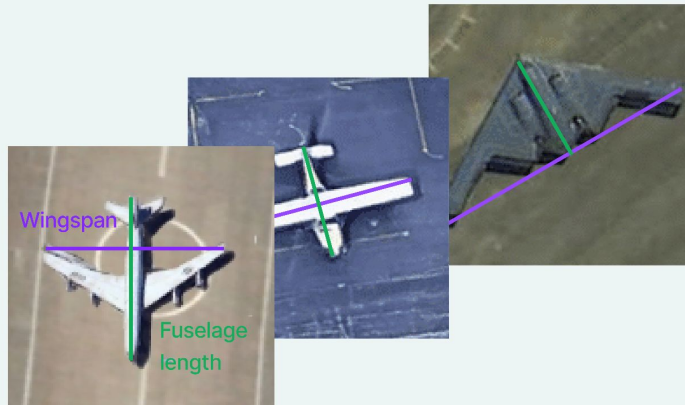


PID efficiency and contamination for all tracks (left) and only identified tracks (right)

in Bi+Bi collisions at 9.2 GeV

# Tabular Data: Deep Learning vs Gradient Boosting

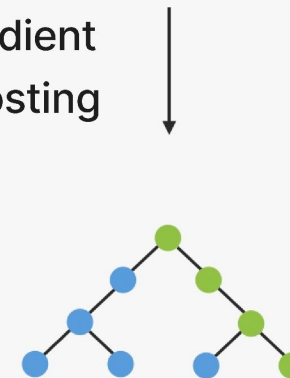
## Unstructured data



## Structured data

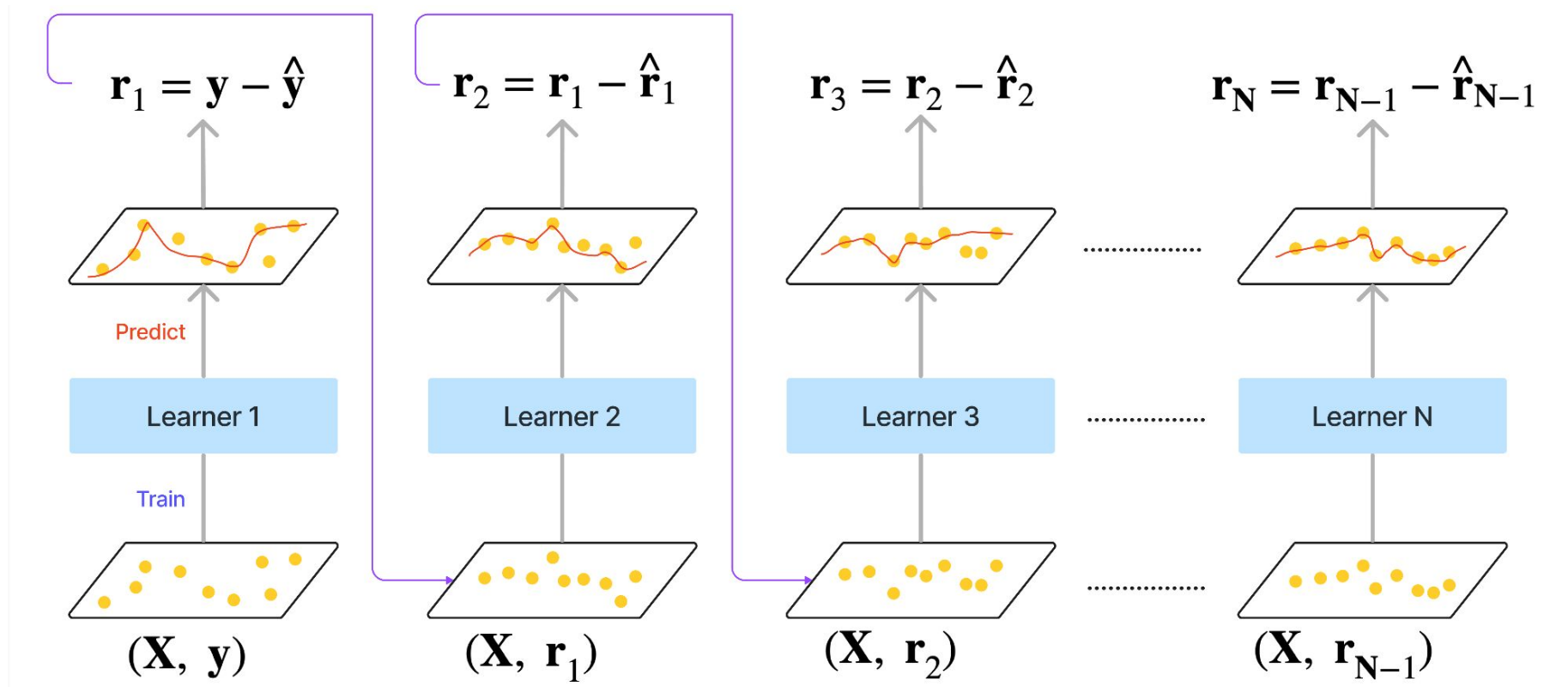
	Fuselage length	Wingspan
Boeing 707	44,07	39,9
Cessna 172	8,28	11
B-2 Spirit	20,90	52,12

Gradient Boosting



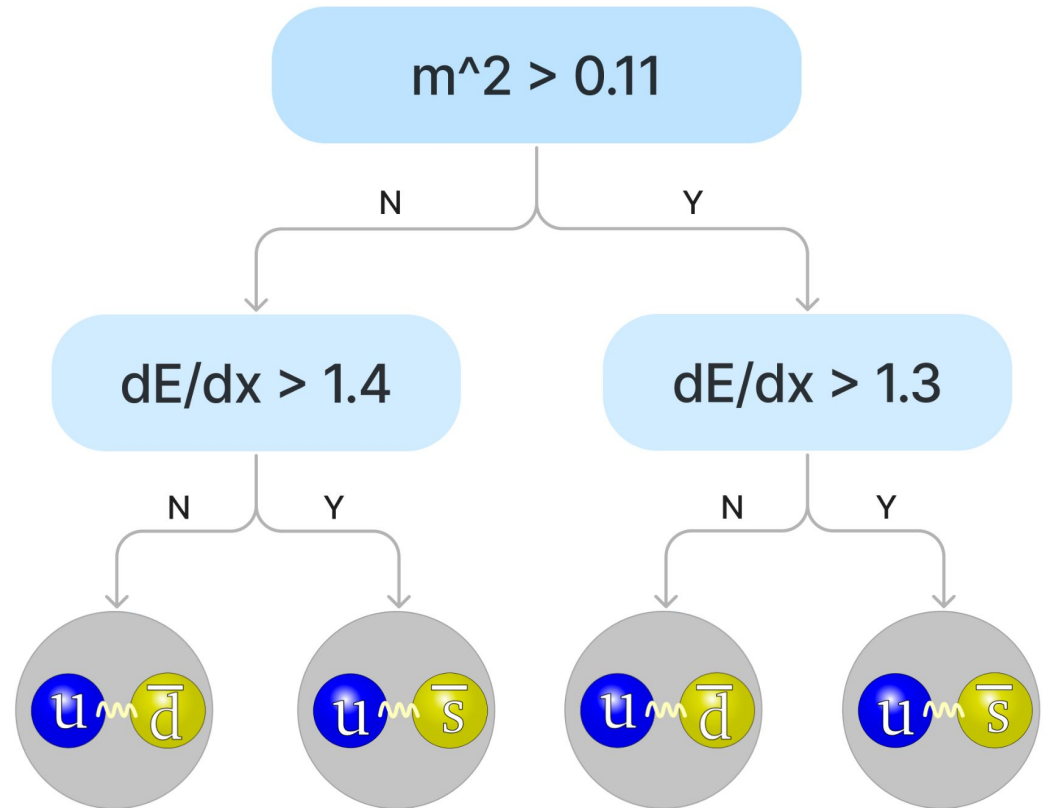
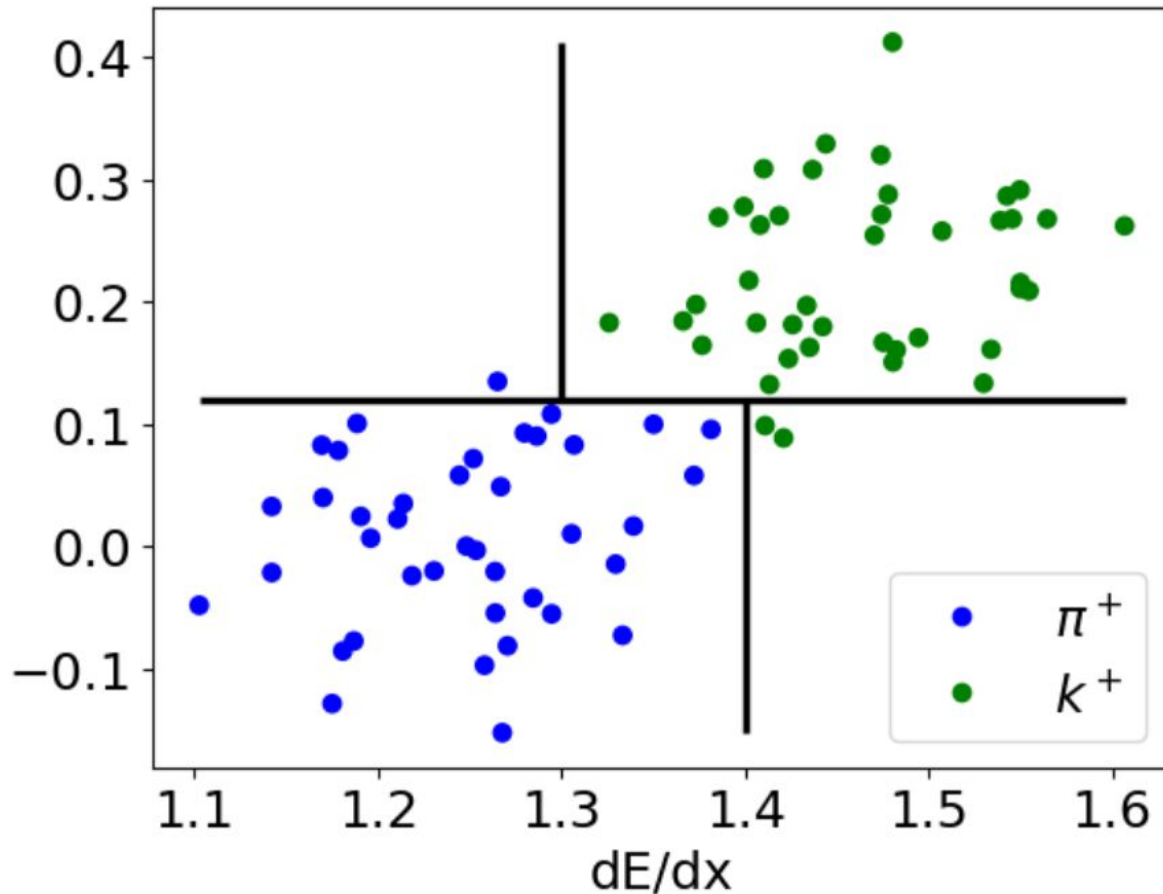
# Gradient Boosting

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



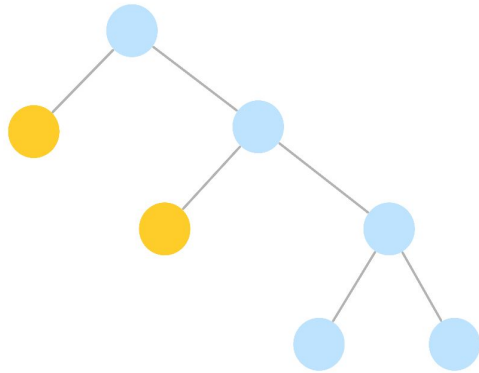
# Gradient Boosted Decision Tree

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

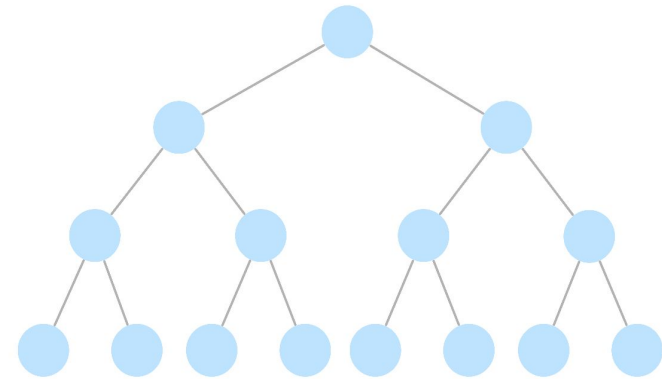


# XGBoost vs LightGBM vs CatBoost vs SketchBoost

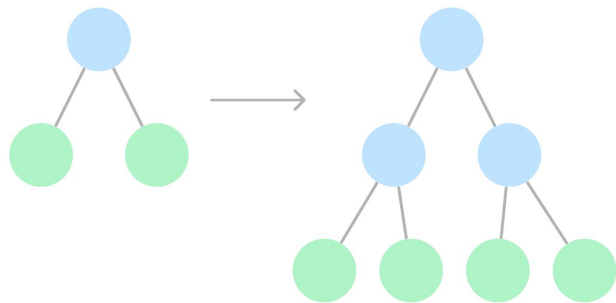
Asymmetric Tree (XGB, LGBM)



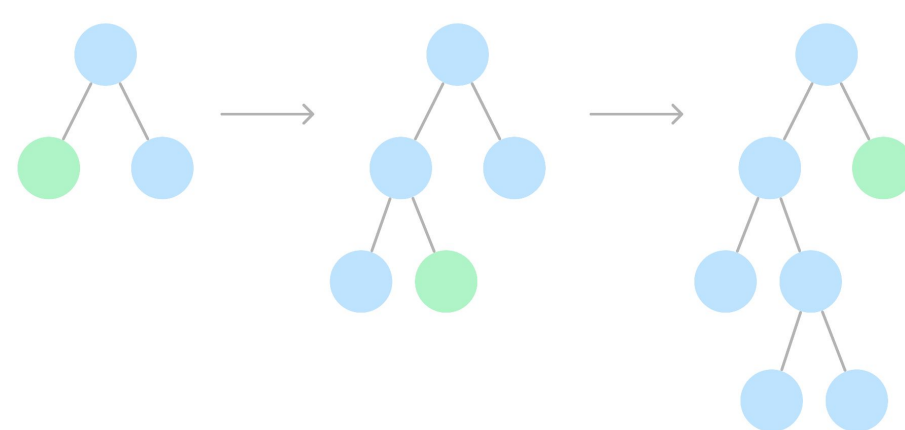
Symmetric Tree (CatBoost, SketchBoost)



Level-wise Tree Growth (XGB)



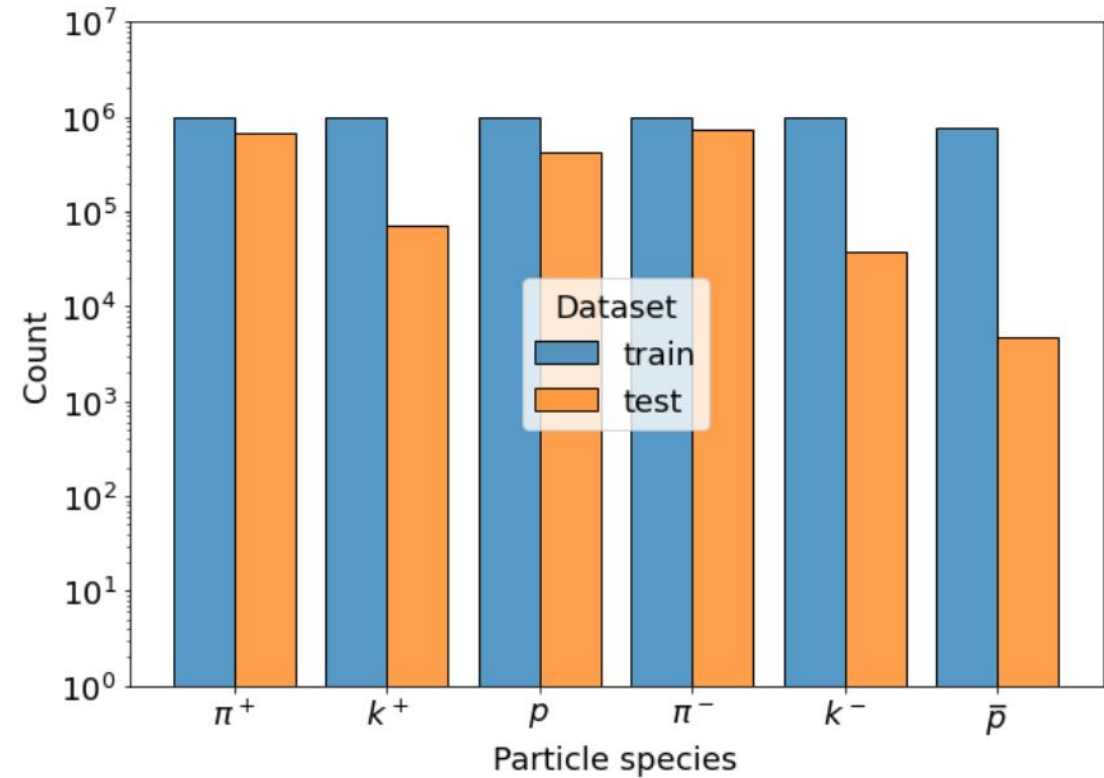
Leaf-wise Tree Growth (LGBM)



# Datasets

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

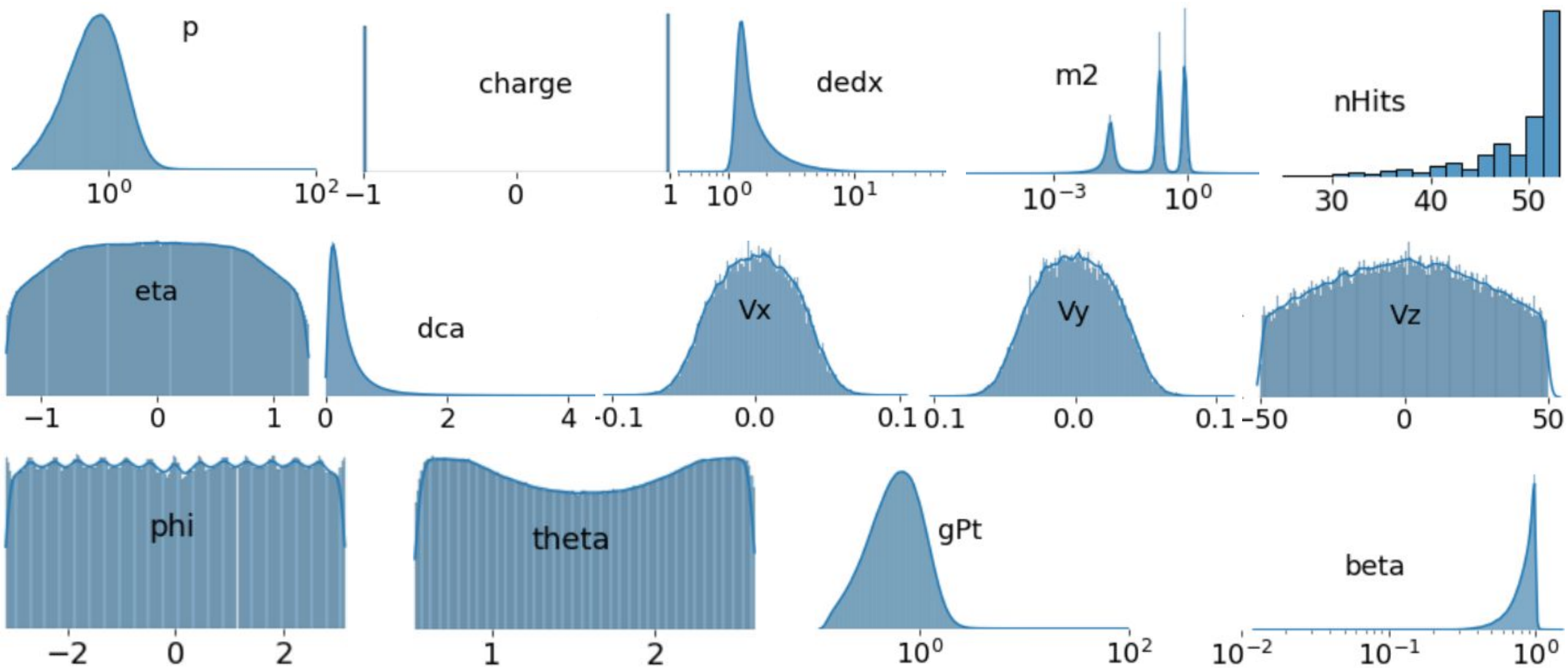
	<b>prod05</b>	<b>prod06</b>
Event generator	UrQMD	PHQMD
Transport	Geant 4	Geant 4
Impact parameter ranges	0-16 fm (mb)	0-12 fm
Smear Vertex XY	0.1 cm	0.1 cm
Smear Vertex Z	50 cm	50 cm
Colliding system	Bi+Bi	Bi+Bi
Energy	9.2 GeV	9.2 GeV



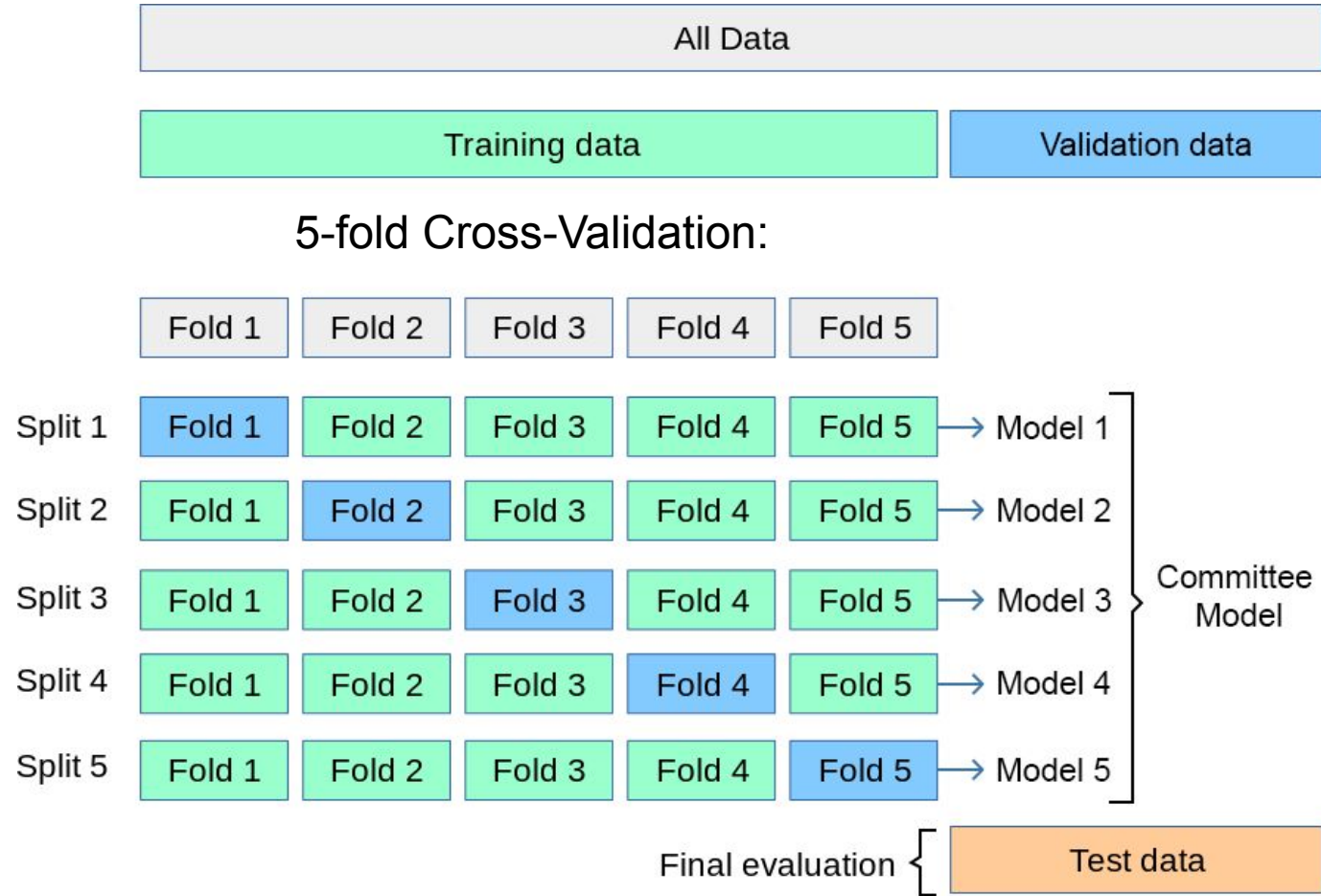
**track selection criteria:** ( $p < 100$ ) & ( $|m^2| < 100$ ) & ( $nHits > 15$ ) & ( $|\eta| < 1.5$ ) & ( $dca < 5$ ) & ( $|Vz| < 100$ )



# 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

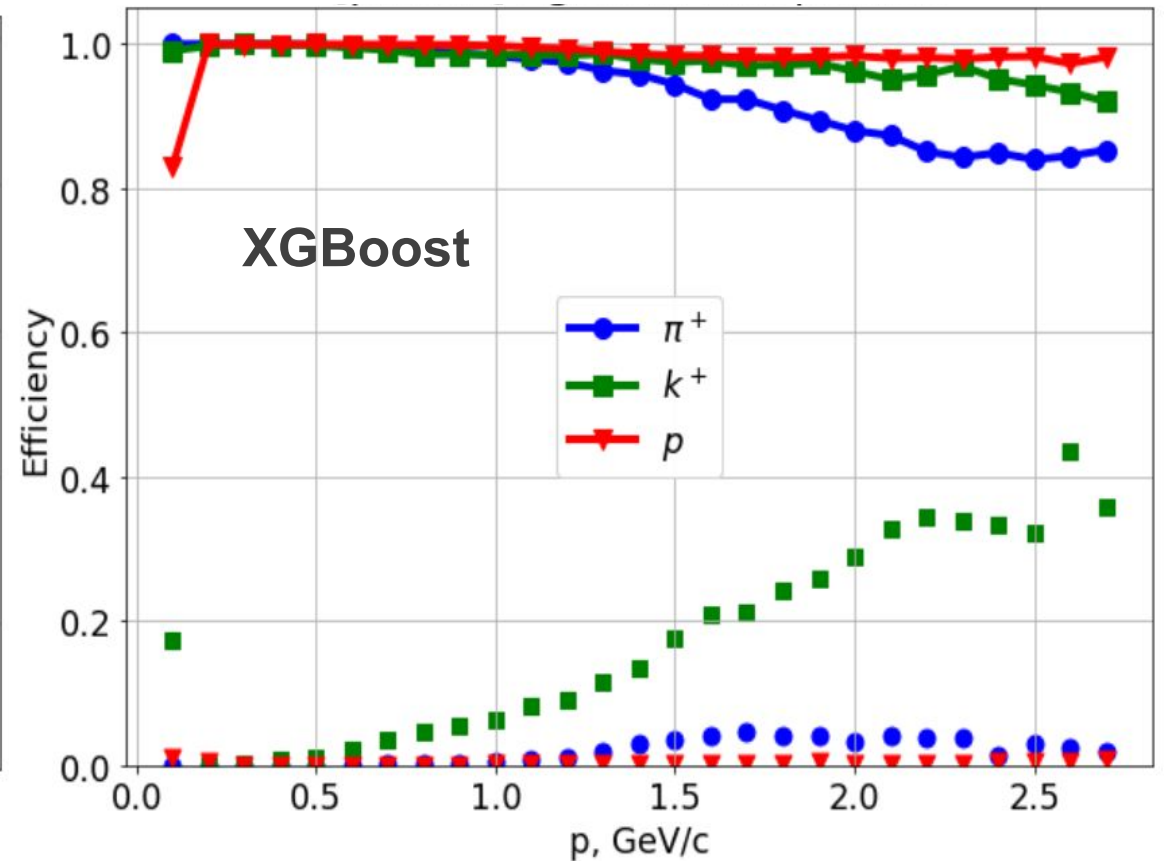
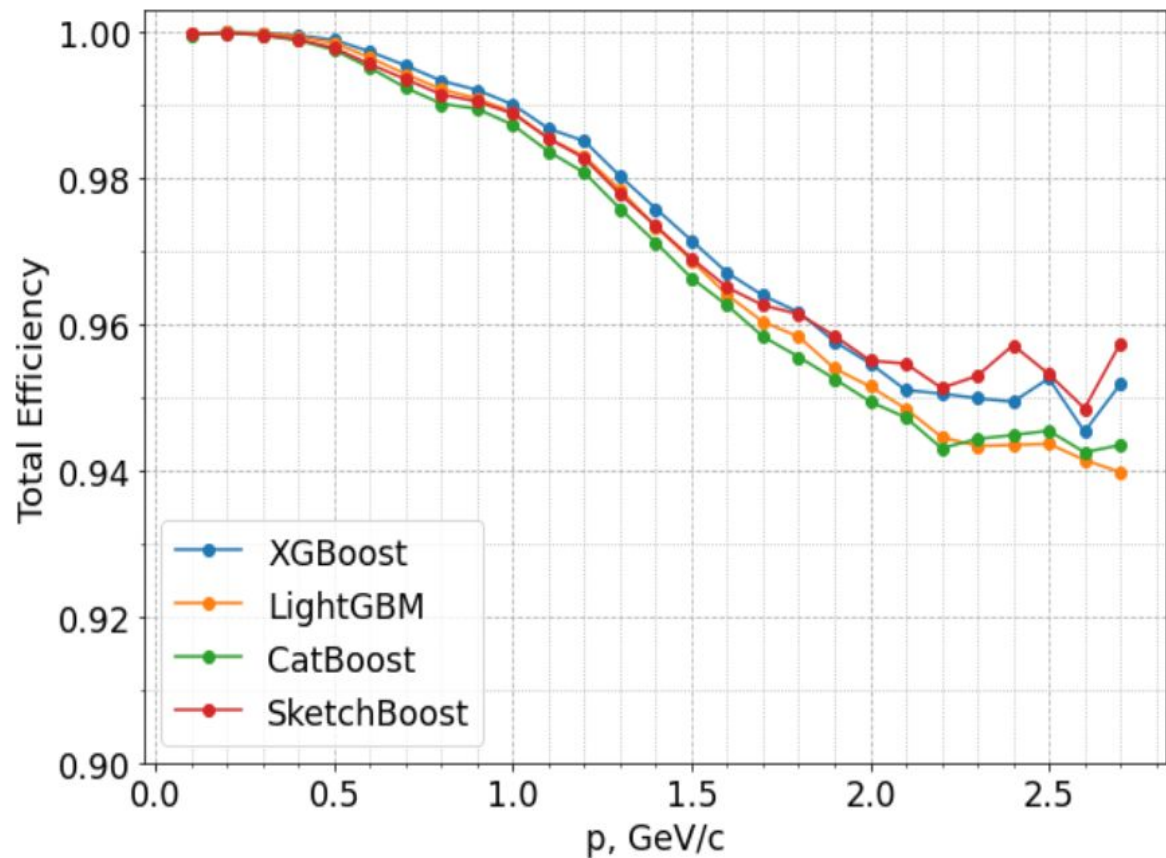
Stage	Learning Rate	Max Number of Iterations	Early Stopping
Tuning	0.05	5 000	200
Model Evaluation	0.015	20 000	500

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

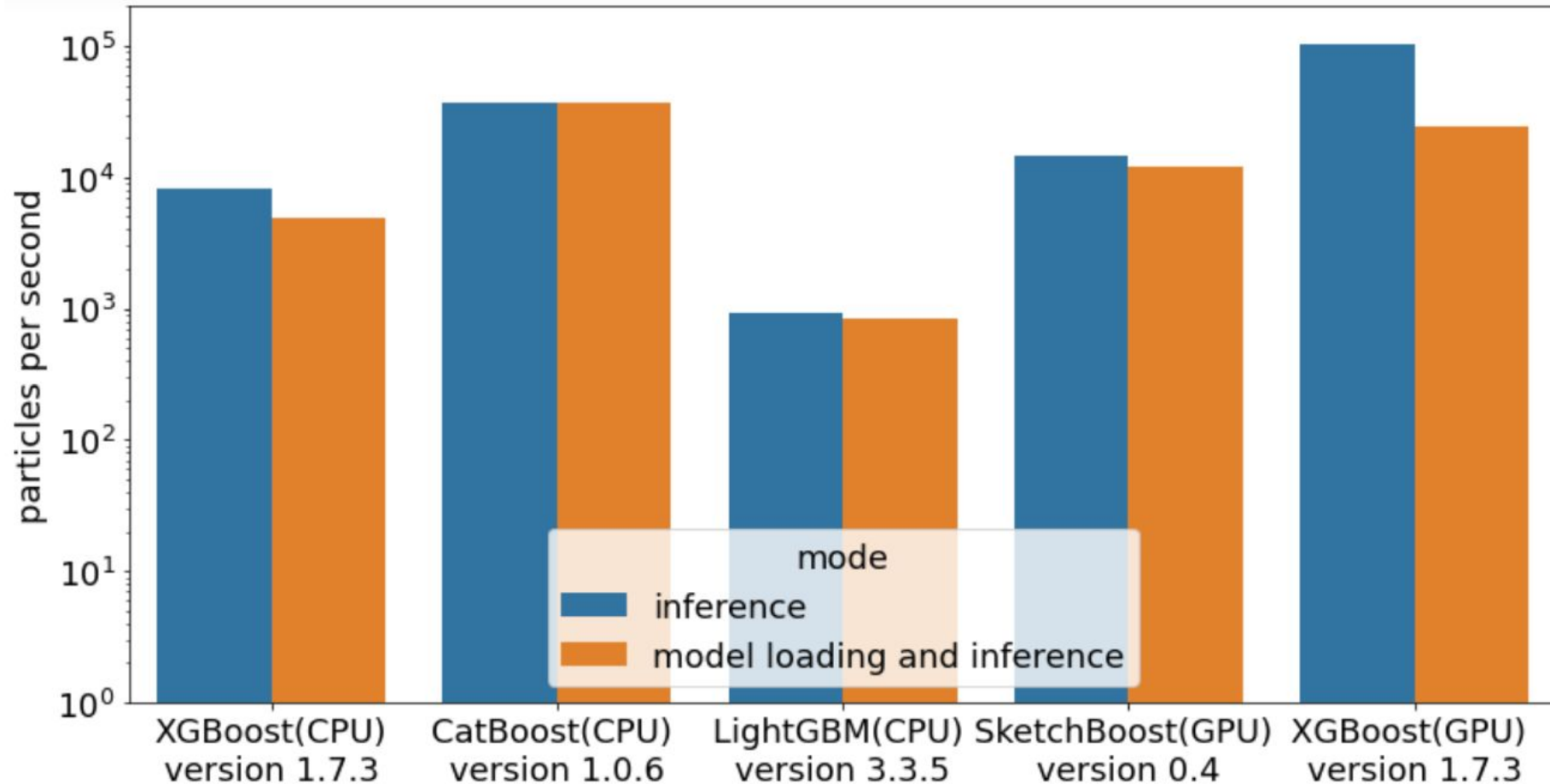
Framework	Max. Depth	L2 leaf reg.	Min. data in leaf size	Rows sampling rate
XGBoost	8	2.3	0.00234	0.942
LightGBM	12	0.1	4	0.981
CatBoost	8	3.0	5	0.99
SketchBoost	8	3.0	5	0.99

# 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

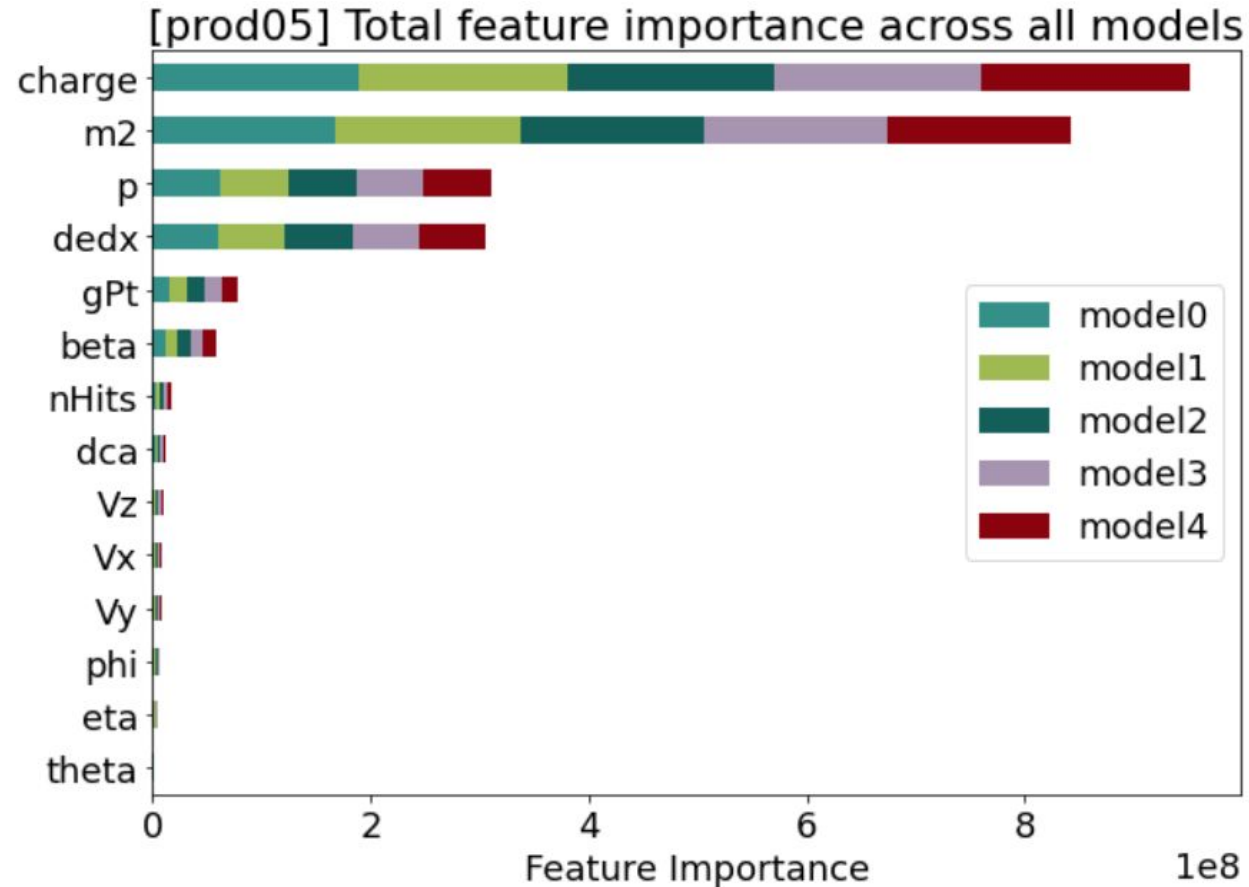


**GPU:** Nvidia Tesla V100-SXM2 NVLink 32GB HBM2

**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

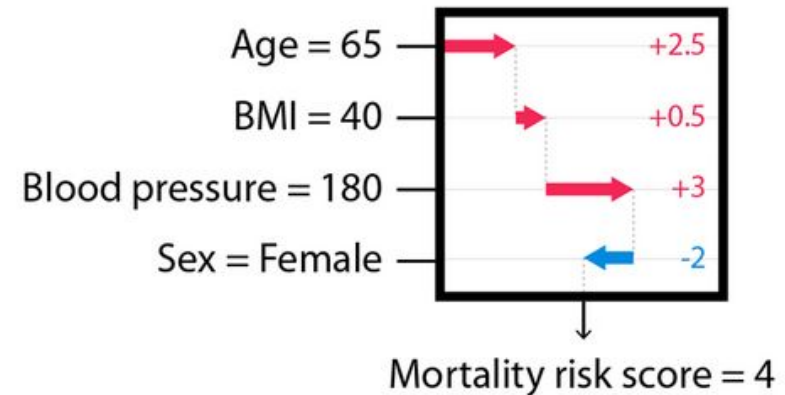
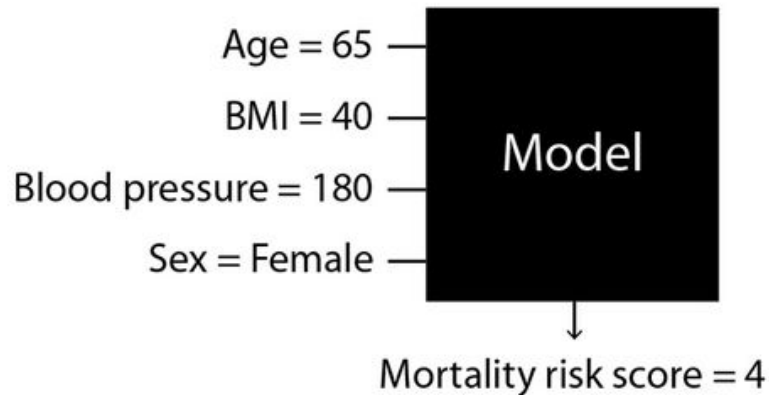
# Model Interpretation. Shapley Additive exPlanations

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



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

**|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**"



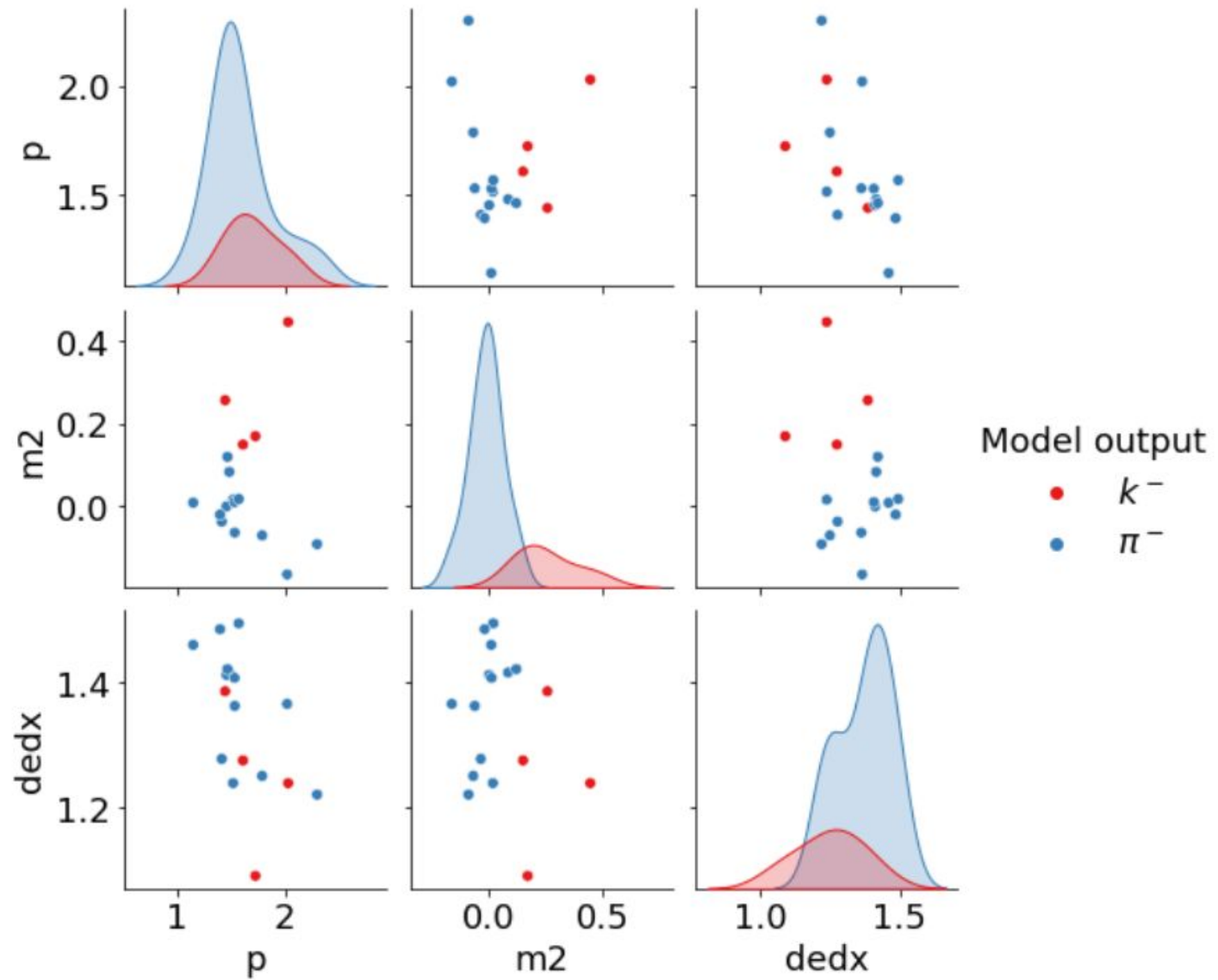
# Misclassification. Confusion Matrices

$\pi^+$	675412	4109	476	0	0	0
$k^+$	637	70810	234	0	0	0
$\rho$	1027	1726	414635	0	0	0
$\pi^-$	0	0	0	738822	4114	377
$k^-$	0	0	0	354	37787	72
$\bar{p}$	0	0	0	13	4	4659
True label	$\pi^+$	$k^+$	$\rho$	$\pi^-$	$k^-$	$\bar{p}$
	Predicted label					

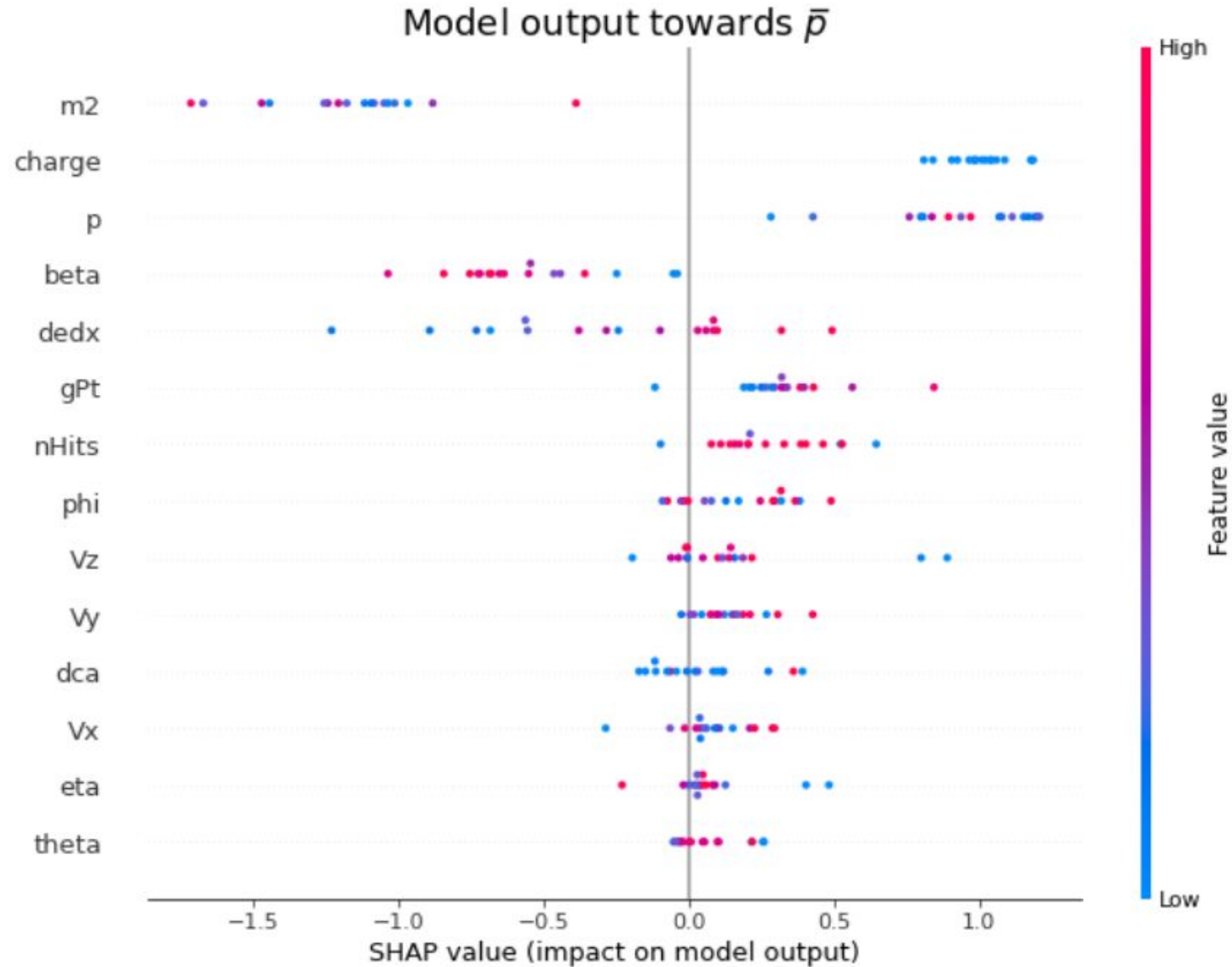
$\pi^+$	99.33%	0.60%	0.07%	0.00%	0.00%	0.00%
$k^+$	0.89%	98.78%	0.33%	0.00%	0.00%	0.00%
$\rho$	0.25%	0.41%	99.34%	0.00%	0.00%	0.00%
$\pi^-$	0.00%	0.00%	0.00%	99.40%	0.55%	0.05%
$k^-$	0.00%	0.00%	0.00%	0.93%	98.89%	0.19%
$\bar{p}$	0.00%	0.00%	0.00%	0.28%	0.09%	99.64%
True label	$\pi^+$	$k^+$	$\rho$	$\pi^-$	$k^-$	$\bar{p}$
	Predicted label					



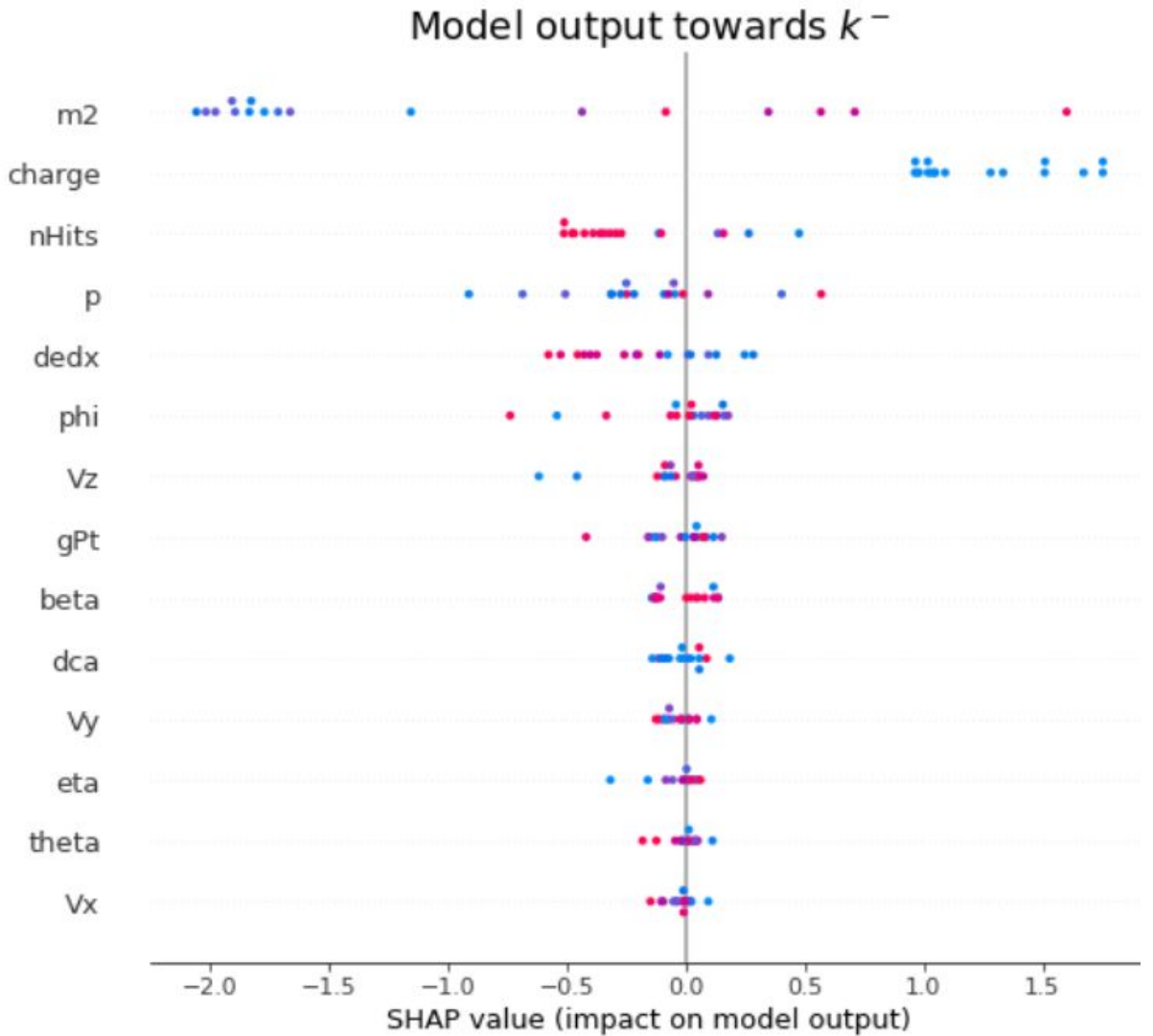
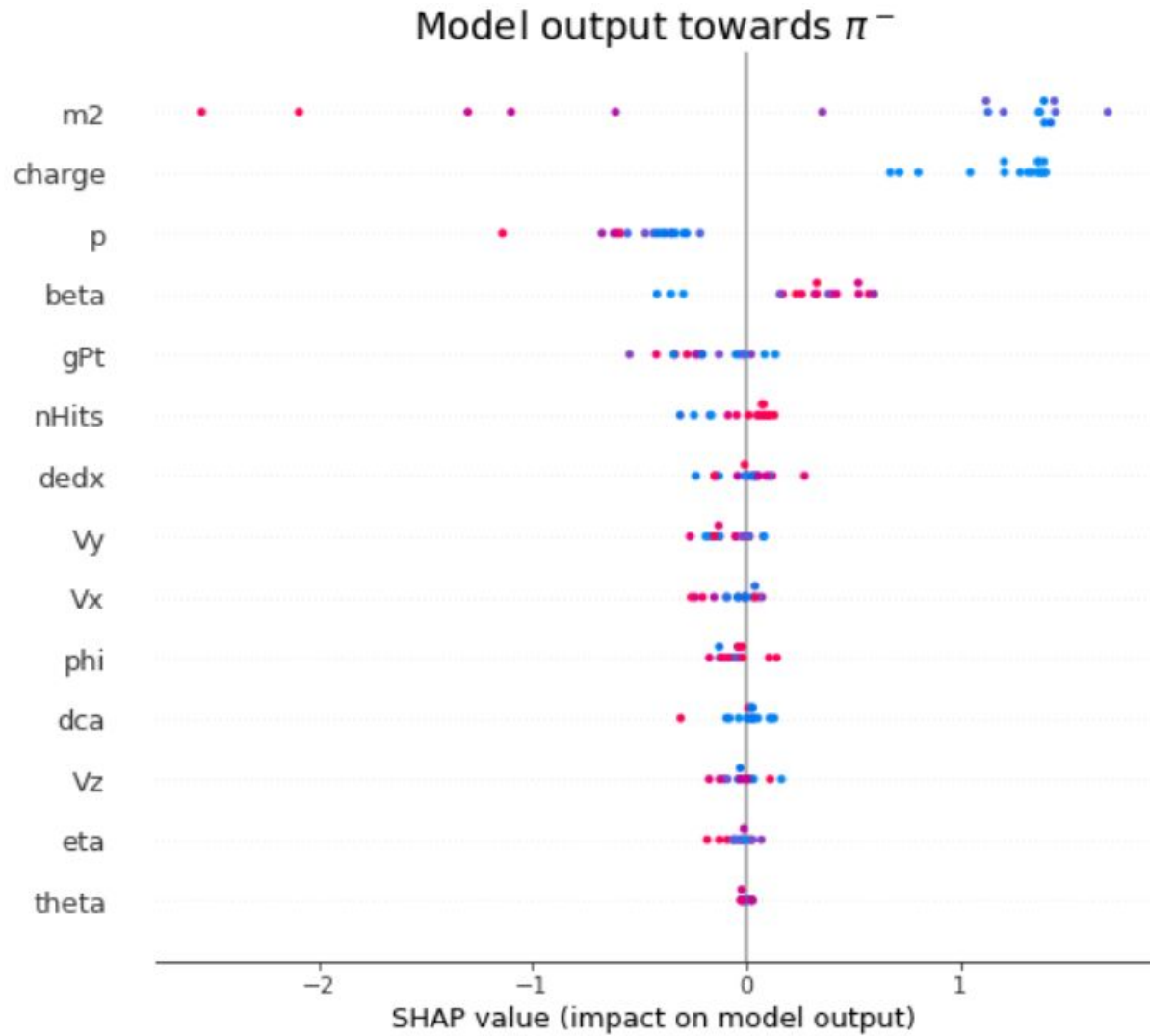
# Miss Classification. Antiprotons



# Miss Classification. Antiprotons

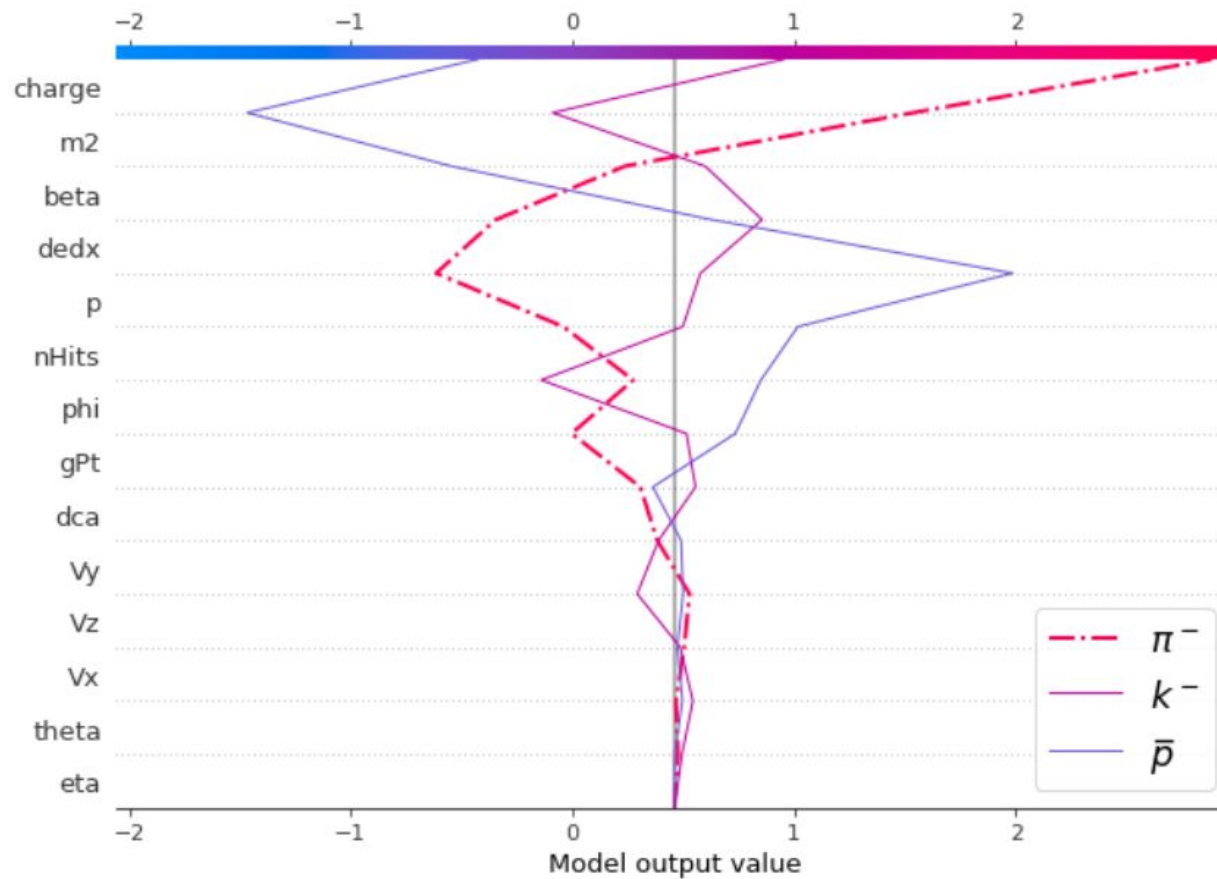


# Miss Classification. Antiprotons

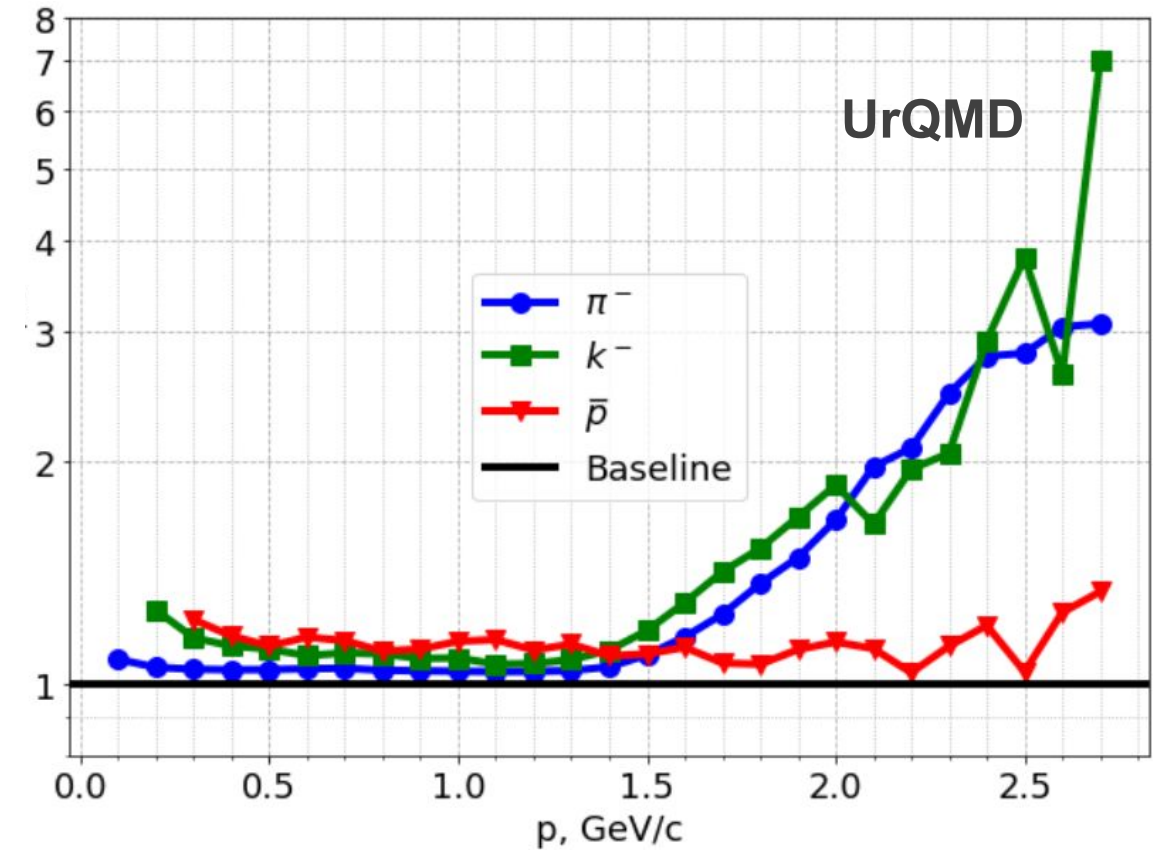
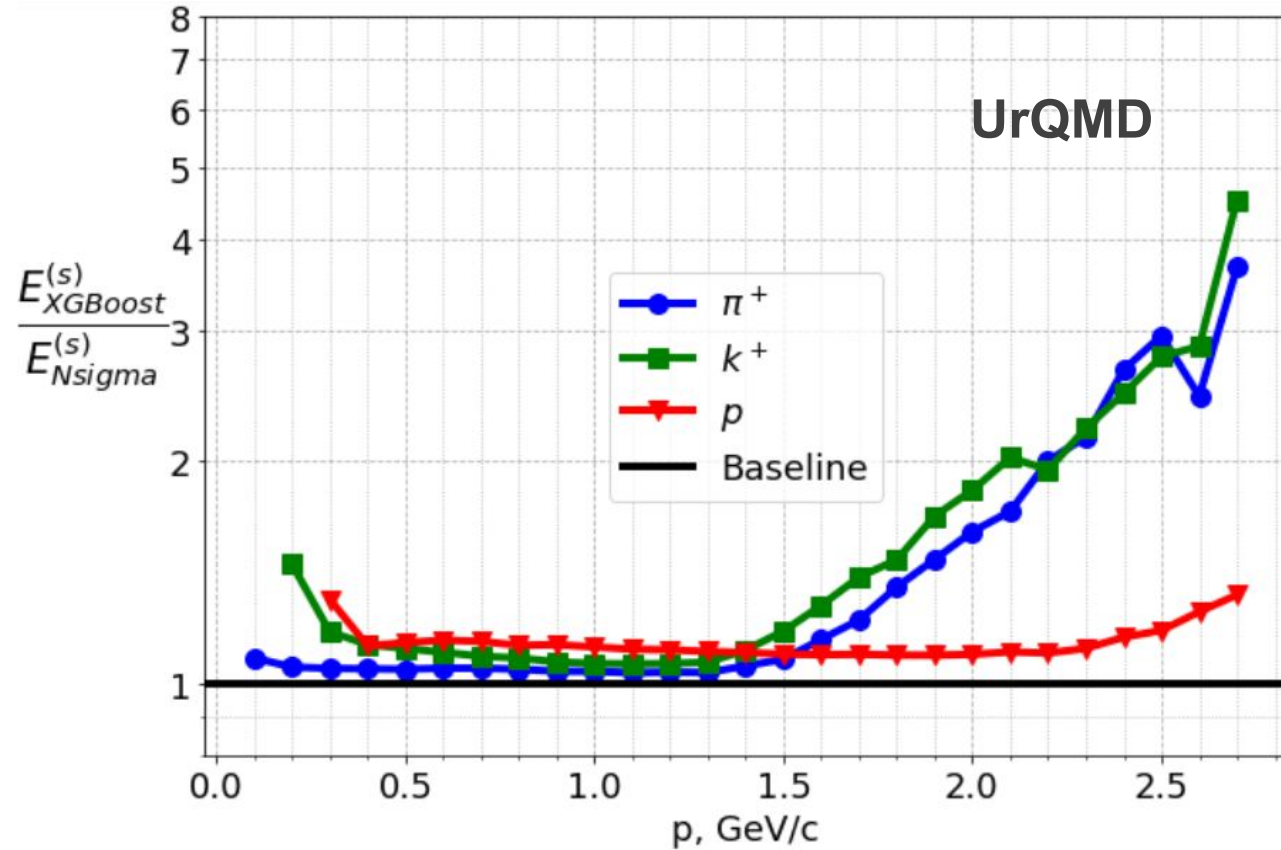


# Miss Classification. Antiprotons

	p	charge	dedx	m2	nHits	eta	dca	Vx	Vy	Vz	phi	theta	gPt	beta
<b>383509</b>	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

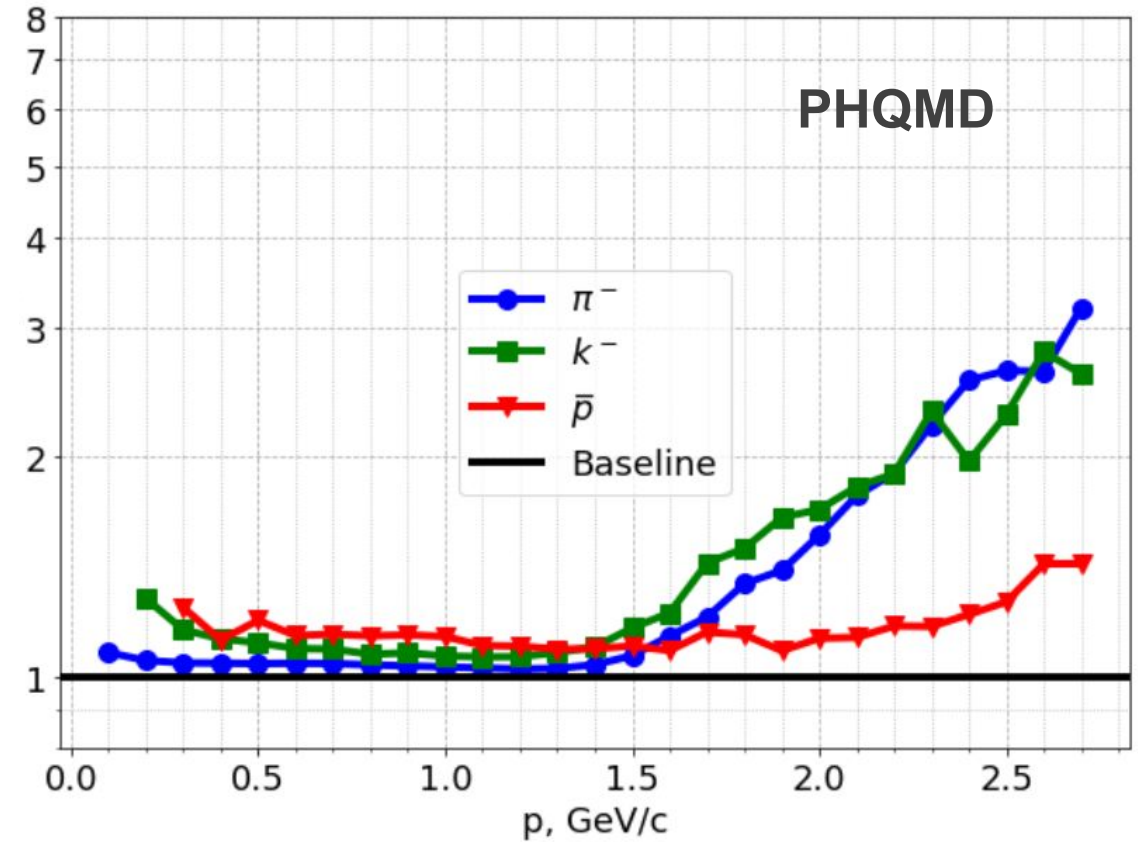
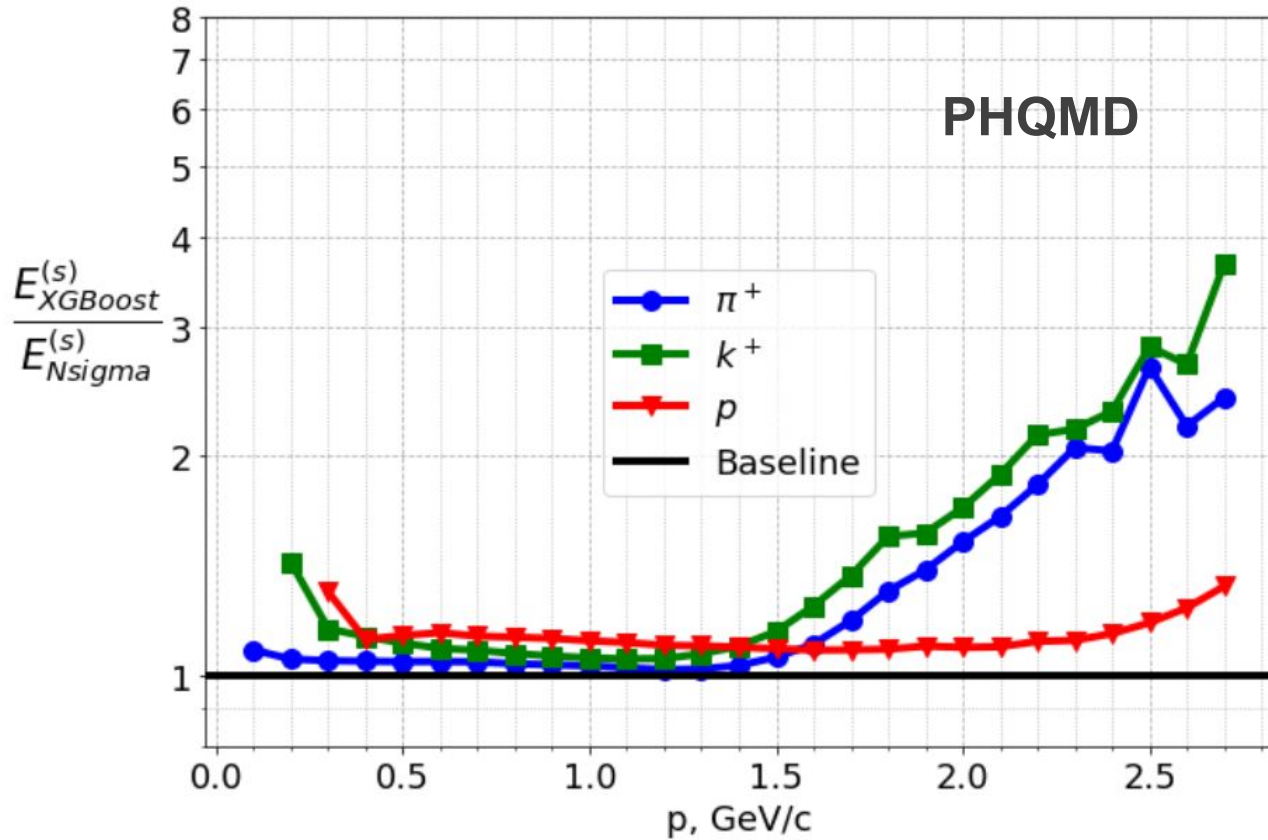


# Comparison with N-sigma



Efficiency ratio of XGBoost and n-sigma method

# Comparison with N-sigma



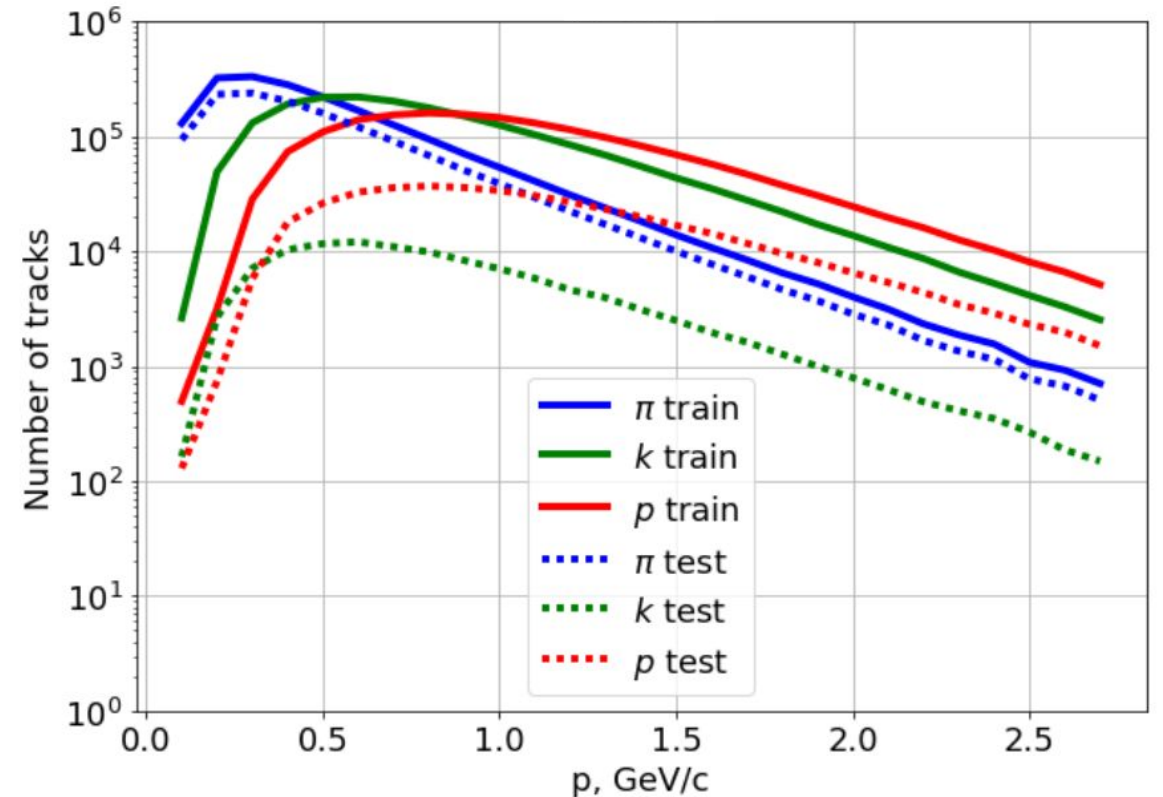
Efficiency ratio of XGBoost and n-sigma method

# Conclusion and Outlooks

In general XGBoost has been demonstrated highest PID efficiency in comparison with considered algorithms of GBDT.

Next we are going to do additional testing to characterize identification stability of the model on data produced with different initial parameters of generated MC tracks at the MPD detector;

Also we are going to analyse the nature of the misclassifications and investigate the class imbalance problem.



# Backup



# Formulas

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

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

# 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:  $dE/dx$ ,  $m^2$ ,  $\beta$ ,  $q$ , etc)

**Y** - is the output space (particle species such as:  $\pi$ ,  $k$ ,  $p$ , etc)

**Unknown** mapping exists

$$m : X \rightarrow Y,$$

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

$$X^n = (x_1, y_1), \dots, (x_n, y_n),$$

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

$$a : X \rightarrow Y.$$

# Data description

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

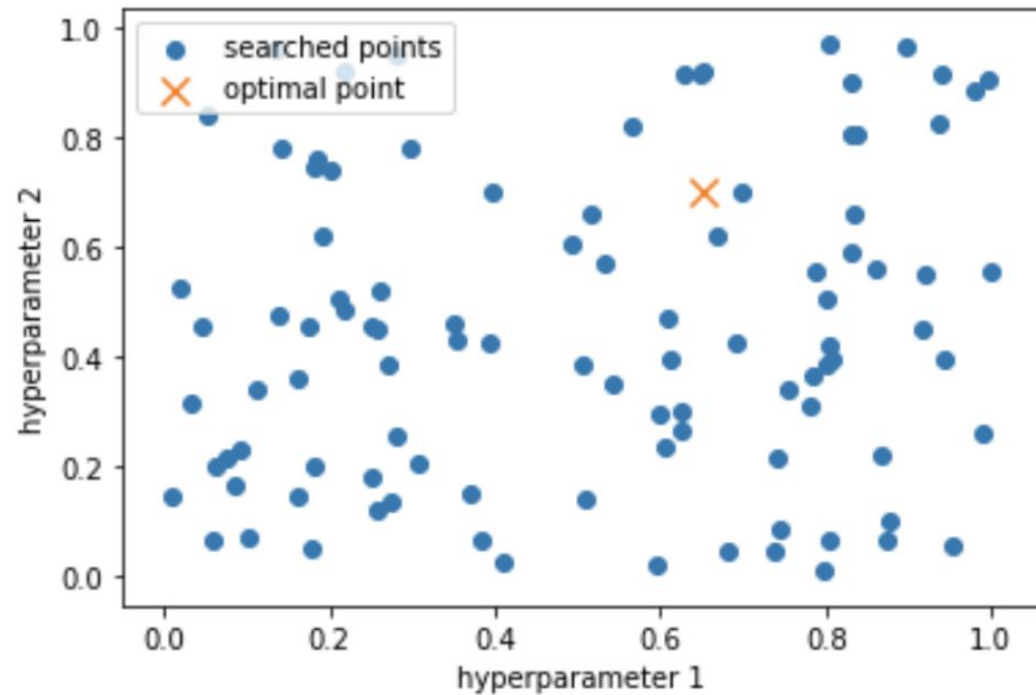
<b>feature</b>	<b>values range</b>
Vx	(-0.106, 0.106)
Vy	(-0.103, 0.112)
Vz	(-50, 54.1)
phi	(-3.1415, 3.1415)
theta	(0.53, 2.61)
gPt	(0.106, 98)
beta	[0.012, 1.564]

# 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

