Data Shift Problem in Machine Learning for Particle Identification

V. Papoyan[1,2]

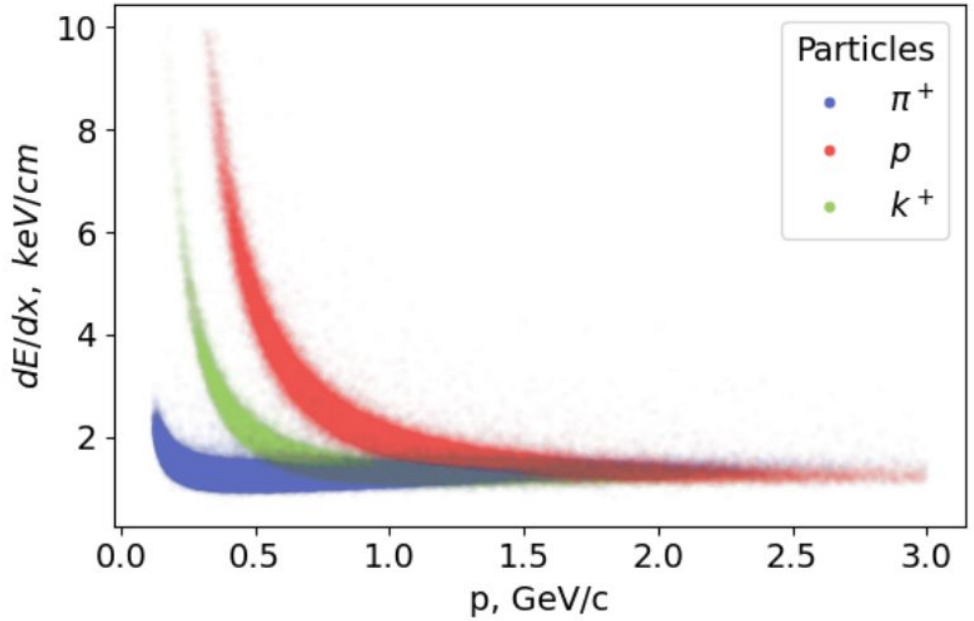
[1]MLIT JINR, [2]AANL (YerPhi)

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

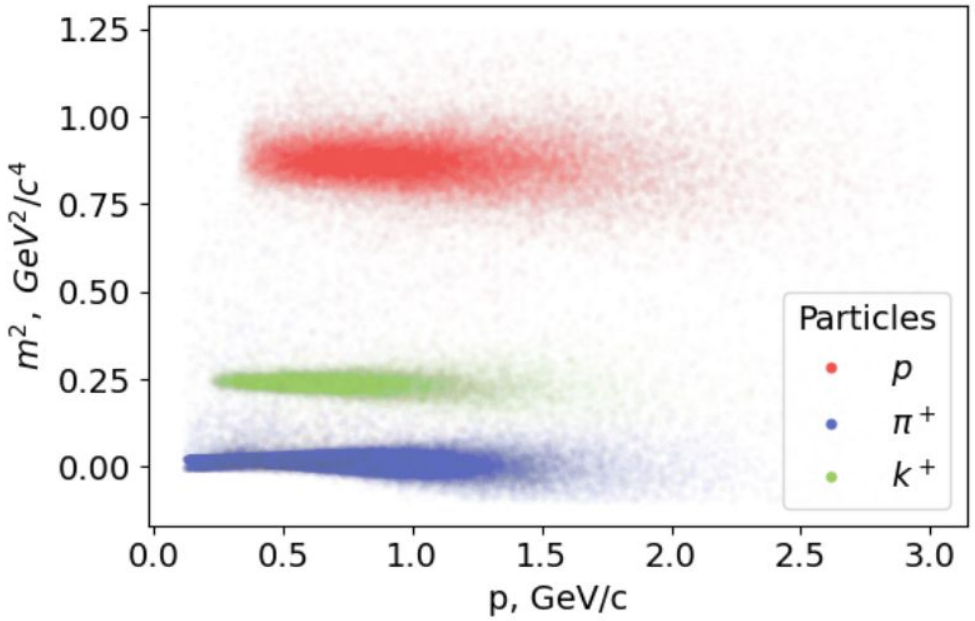
Particle Identification at MPD experiment

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;



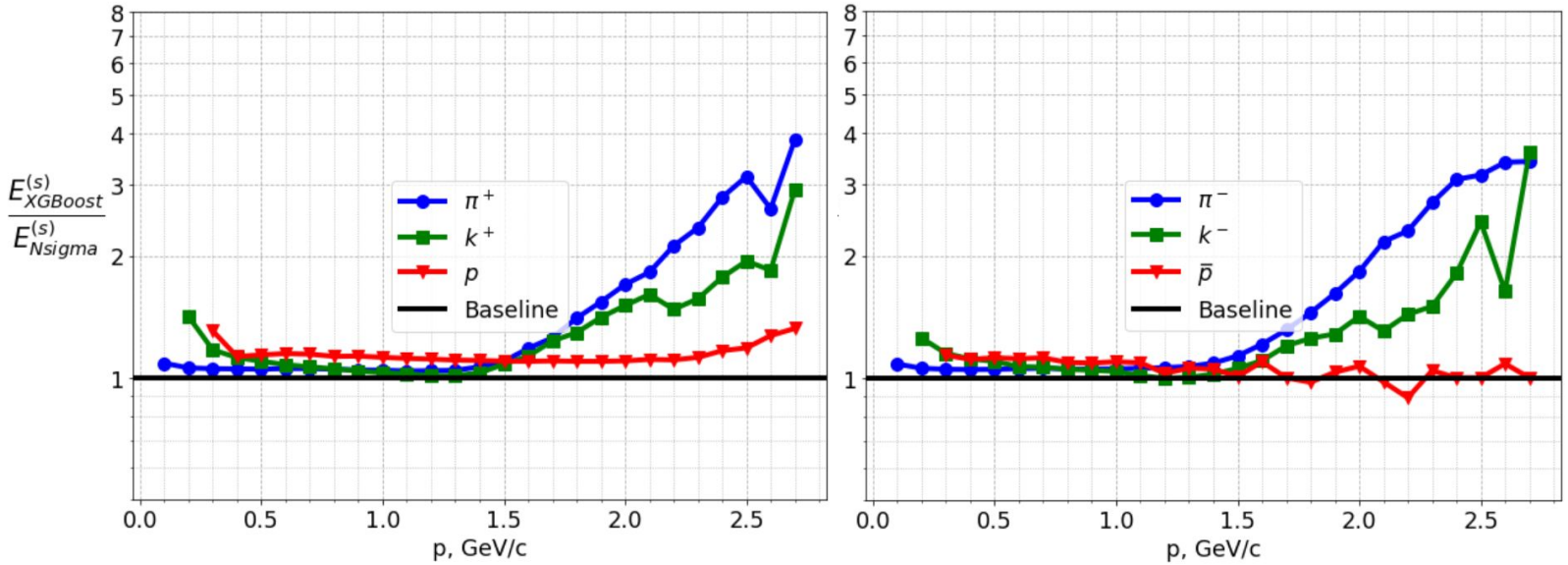
Knowing the particle **momentum** (from TPC) one obtains the **mass squared** and thus identity of the particle.

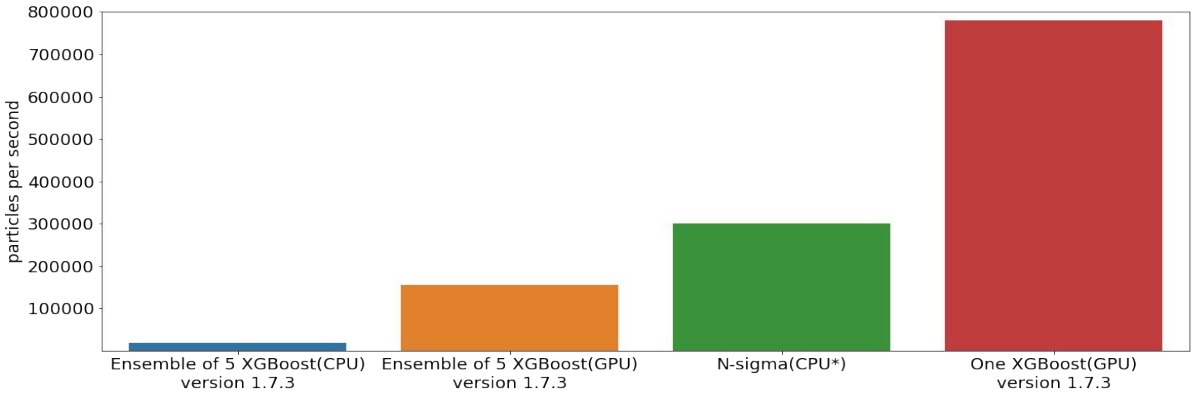
Klempt W. Review of particle identification by time of flight techniques

ML Particle Identification

XGBoost was trained on MPD MC data to perform PID.

The plots compare the XGBoost's performance with baseline MPD PID (N-sigma).





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

GPU: Nvidia Tesla V100-SXM2 NVLink 32GB HBM2

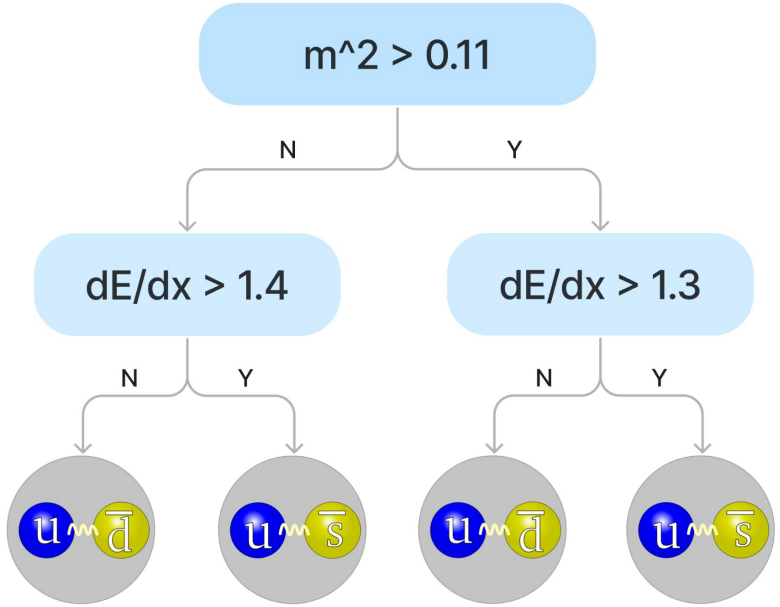
CPU: Intel Xeon Gold 6148 CPU @ 2.40 GHz 20 Cores / 40 Threads

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

Necessity of Monte Carlo Simulations for Training

MPD detector registers signals (e.g., energy deposits, time-of-flight), but there is no direct label indicating whether a particle is a pion or kaon.

Unlike standard ML tasks, there is no ground truth dataset where each particle's type is explicitly known.



ML models must be trained on synthetic data where each particle has a known true label.

How reliable are ML models trained on simulated data? How can we control similarity between training dataset and data which will encounter during deployment?

Causes of Data Shift

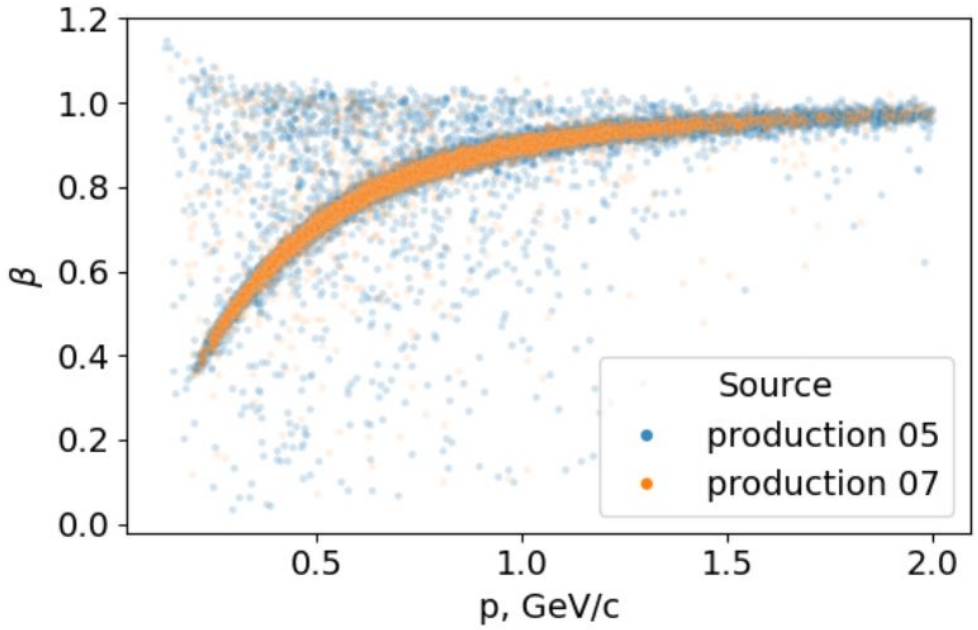
There are several factors when the input particle features' distribution P(X) changes:

- Differences in physical models and parameters of event generators.
- Detector effects and calibration.
- Particle reconstruction algorithms.

Track reconstruction and clustering algorithms may behave differently in simulated and real environments.

Data Shift can decrease classification accuracy and introduce biases.

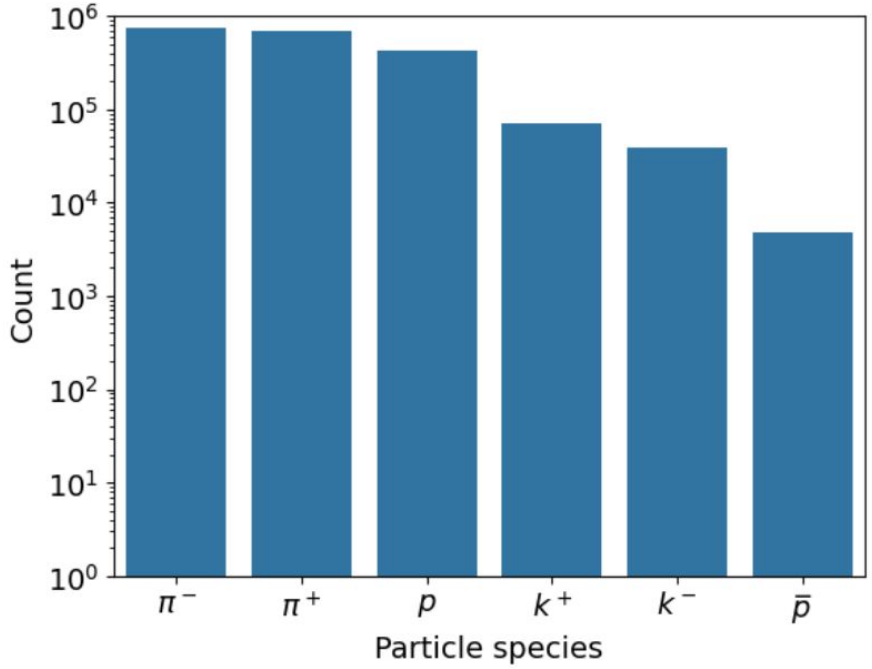
antikaons in different MC datasets



Monte-Carlo datasets

Subsamples of the MPD Monte-Carlo productions (Request 25 and 29) were used.

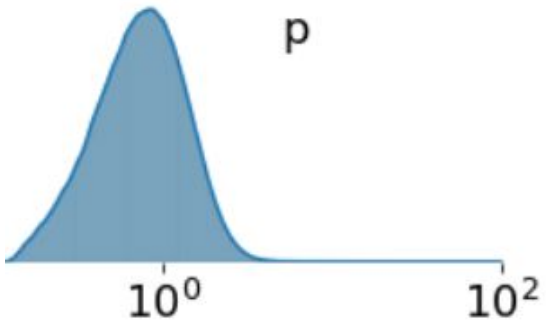
	prod05	prod06	prod07
Event generator	UrQMD	PHQMD	UrQMD
Transport	Geant 4	Geant 4	Geant 4
Impact parameter ranges	0-16 fm (mb)	0-12 fm	0-14 fm
Smear Vertex XY	0.1 cm	0.1 cm	0.1 cm
Smear Vertex Z	50 cm	50 cm	50 cm
Colliding system	Bi+Bi	Bi+Bi	Bi+Bi
Energy	9.2 GeV	9.2 GeV	5.5 GeV

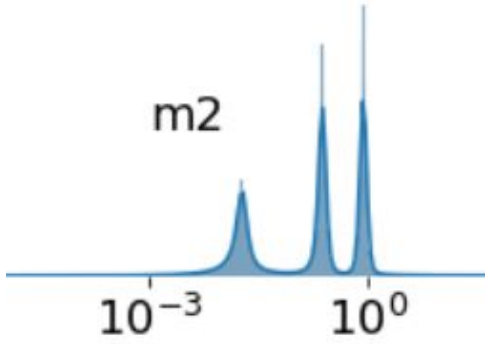


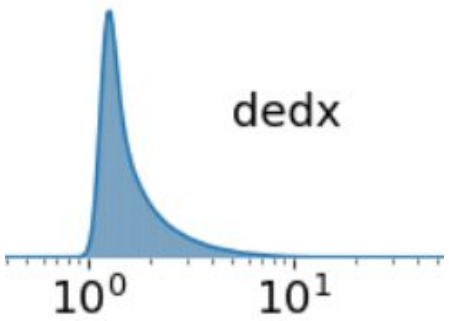
track selection criteria: (p < 100) & (|m^2| < 100) & (nHits > 15) & (|eta|<1.5) & (dca < 5) & (|Vz| < 100)

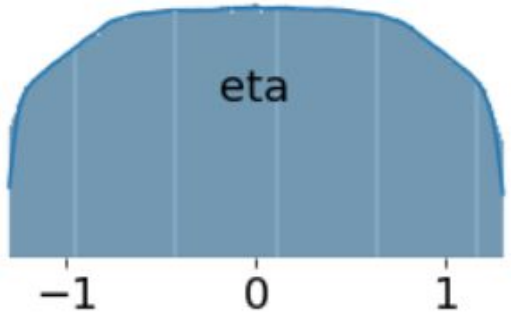
Monte Carlo data were kindly provided by A. Aparin and A. Korobitsin

Features

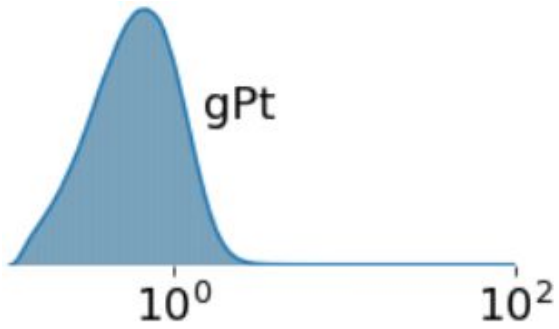

p
10^0
10^2

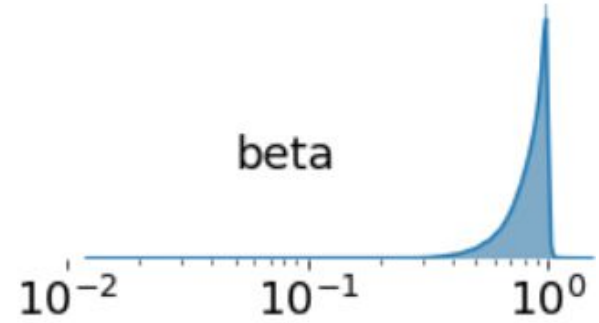

m2
10^{-3}
10^0

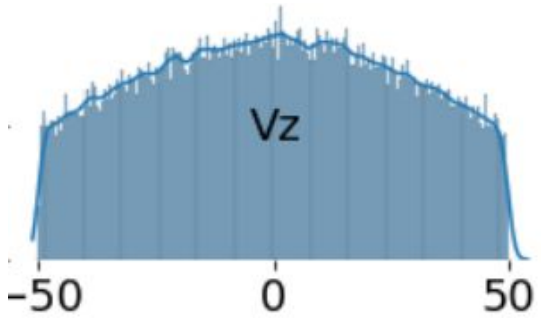

dedx
10^0
10^1

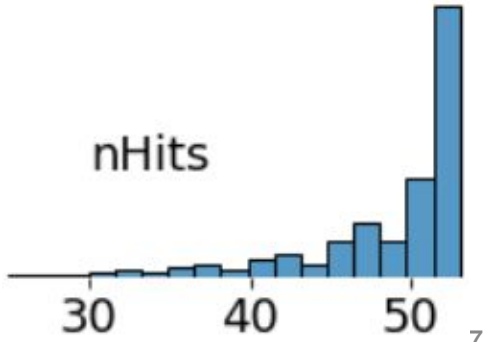

eta
−1
0
1


charge
−1
0
1


gPt
10^0
10^2


beta
10^{-2}
10^{-1}
10^0

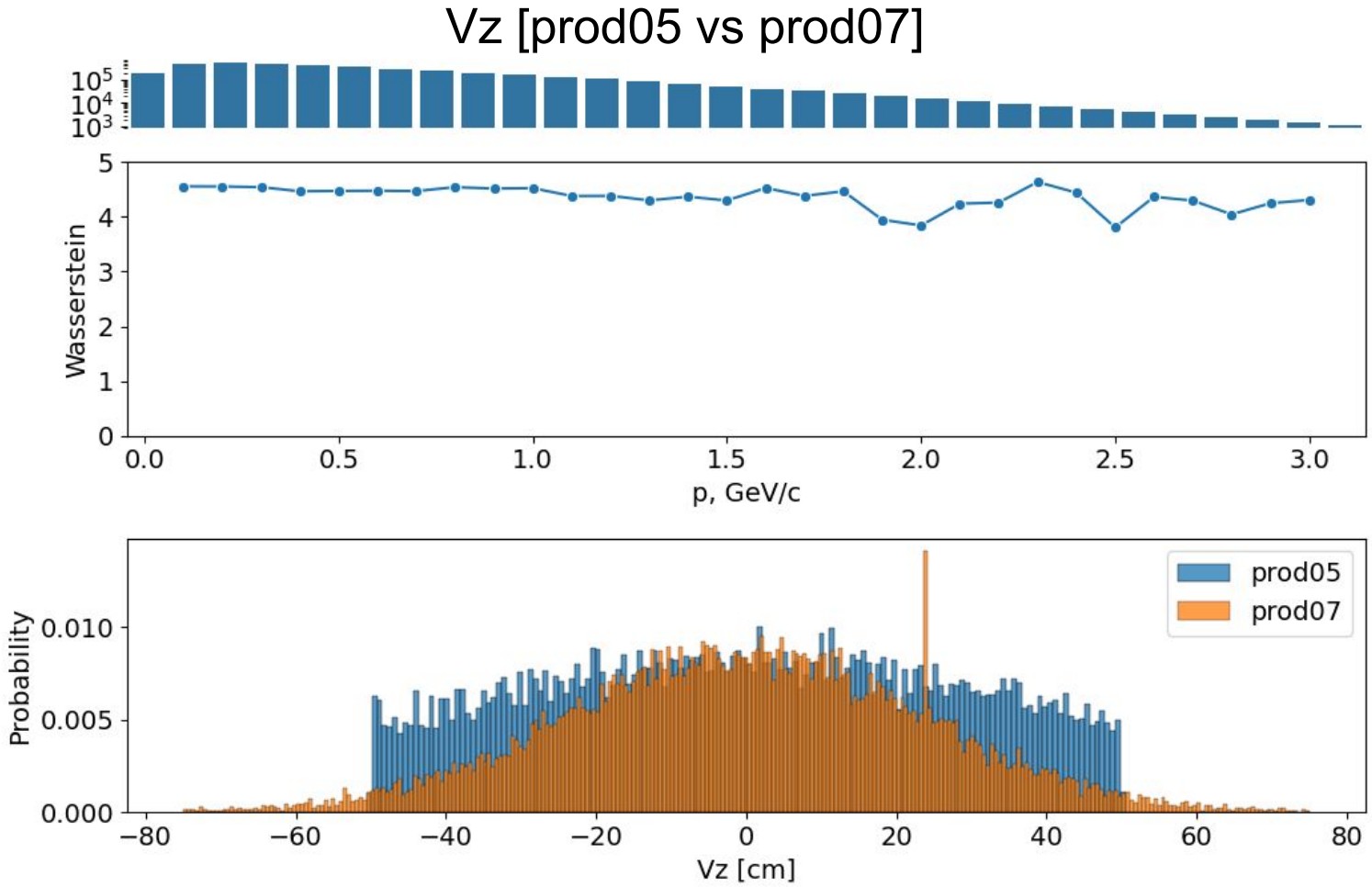

Vz
−50
0
50


nHits
30
40
50

Quantifying Distributional Differences

Wasserstein distance measures the minimal "cost" of transforming one probability distribution into another.

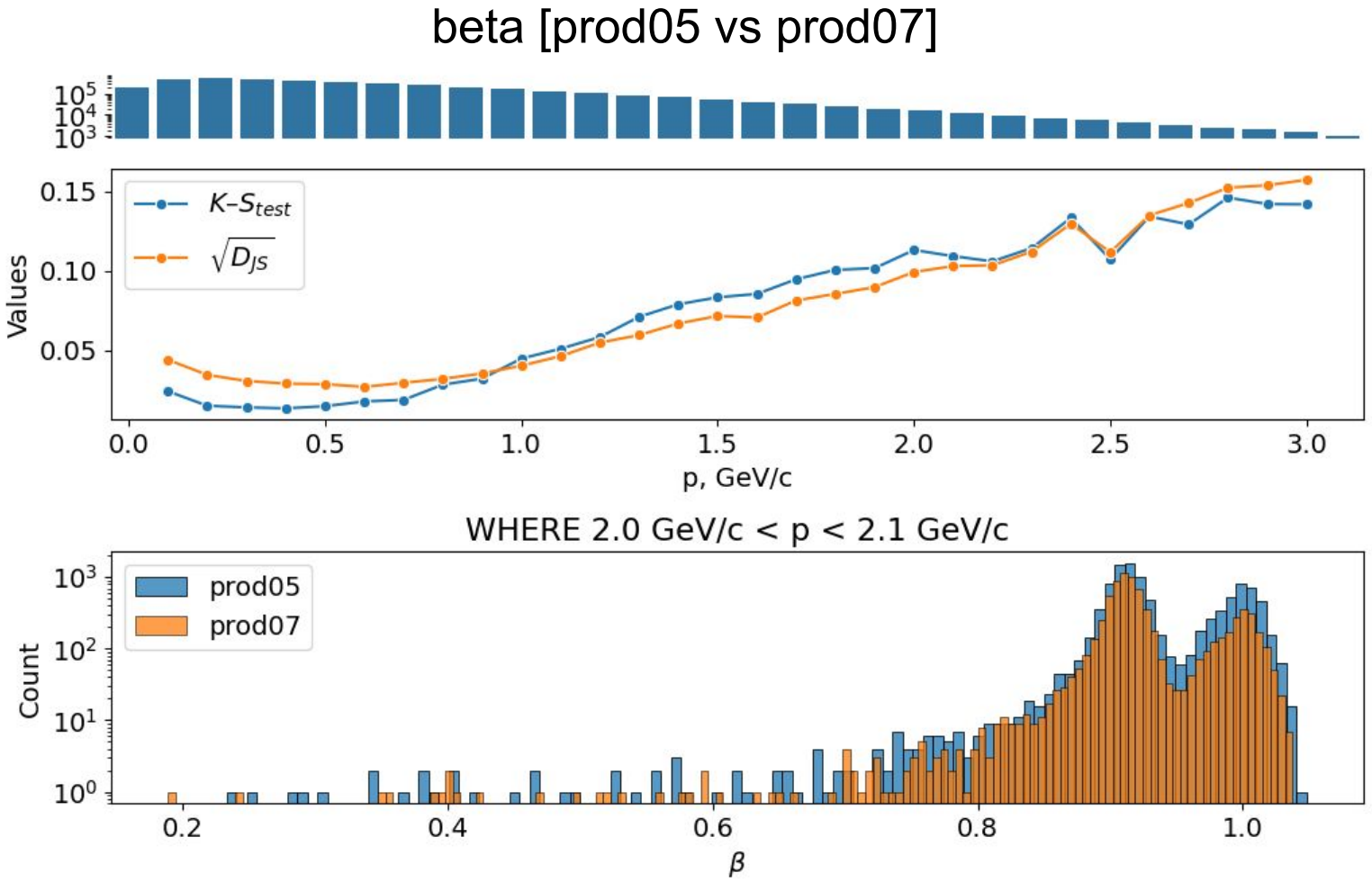
It is sensitive to both location and shape of distributions and works well for continuous features.



Quantifying Distributional Differences

Kolmogorov Smirnov test measures the maximum difference between cumulative distribution functions. Intuition: maximum vertical gap between two CDFs.

Jensen-Shannon Divergence is a symmetric measure that quantifies how different two probability distributions are. It evaluates the similarity by comparing the shape and spread of the distributions.

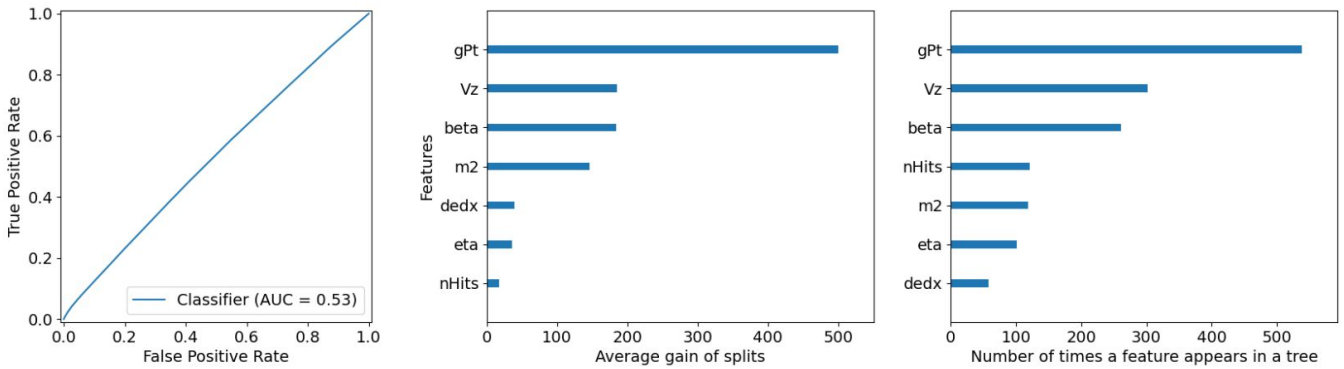


Adversarial Validation

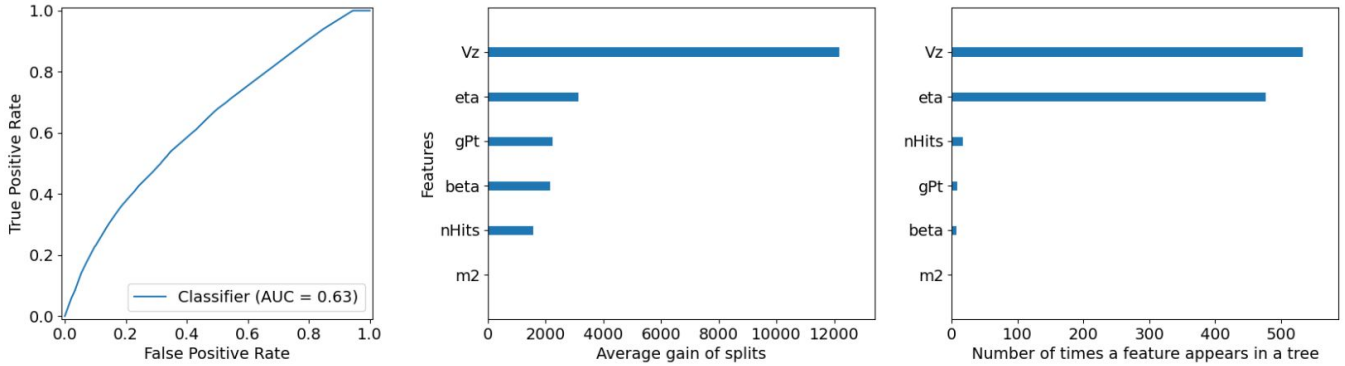
The idea is to treat samples from different MC simulations as belonging to different "pseudo-classes" and train a classifier to distinguish between them.

If the classifier performs poorly (AUC ~ 0.5), the datasets are similar in feature space.
If performance is high (AUC >> 0.5), it indicates a shift in distributions between the simulations.

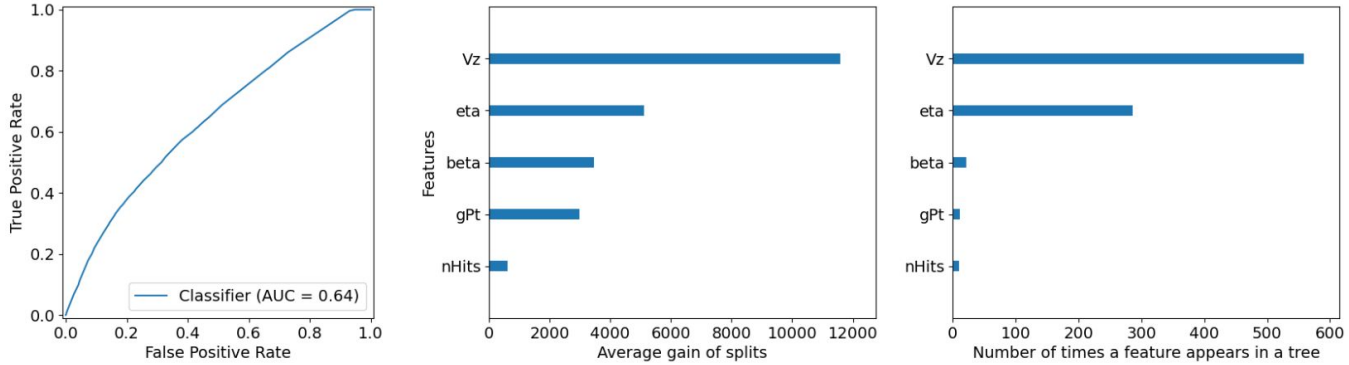
prod05 vs prod06



prod05 vs prod07



prod06 vs prod07

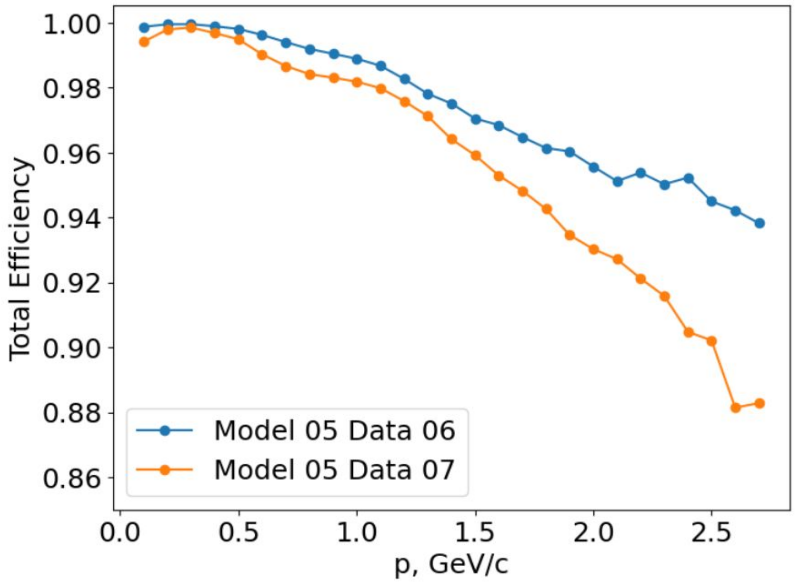


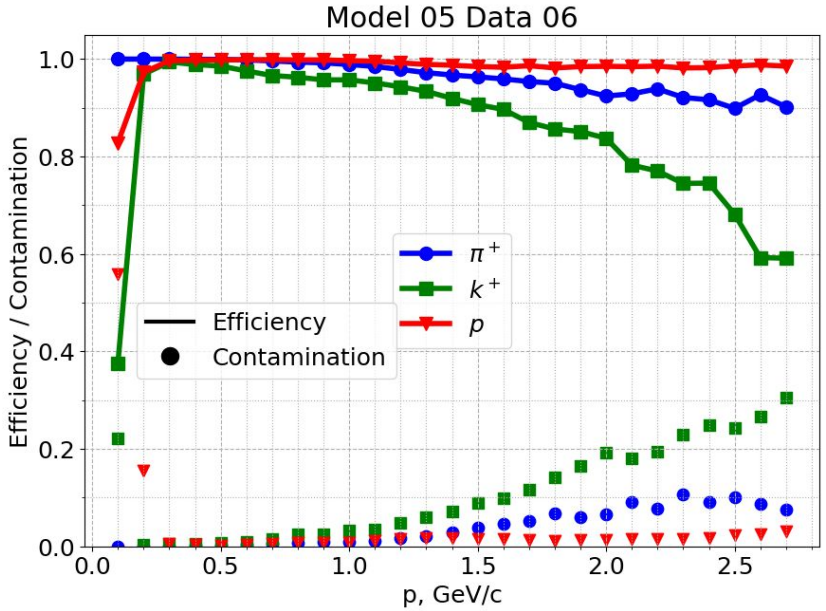
Performance comparison

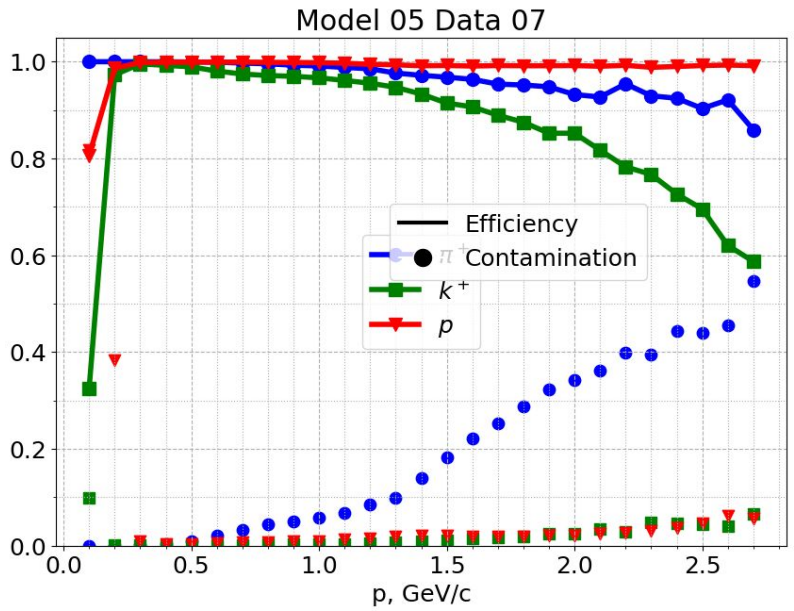
$$E_{total} = \frac{N_{corr}}{N_{true}}$$

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

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







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

Conclusions

- Particle identification models trained on Monte Carlo simulations are inherently sensitive to data shift, since real experimental data lacks ground-truth labels.
- Adversarial validation and statistical metrics (Wasserstein distance, KS test, Jensen–Shannon divergence) can detect and quantify differences between datasets at the feature level.
- Classification performance can vary significantly depending on the training dataset.
- If a significant mismatch is observed between training and real data distributions, it may be necessary to retrain the model on a more representative dataset or fall back to traditional PID techniques.

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

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

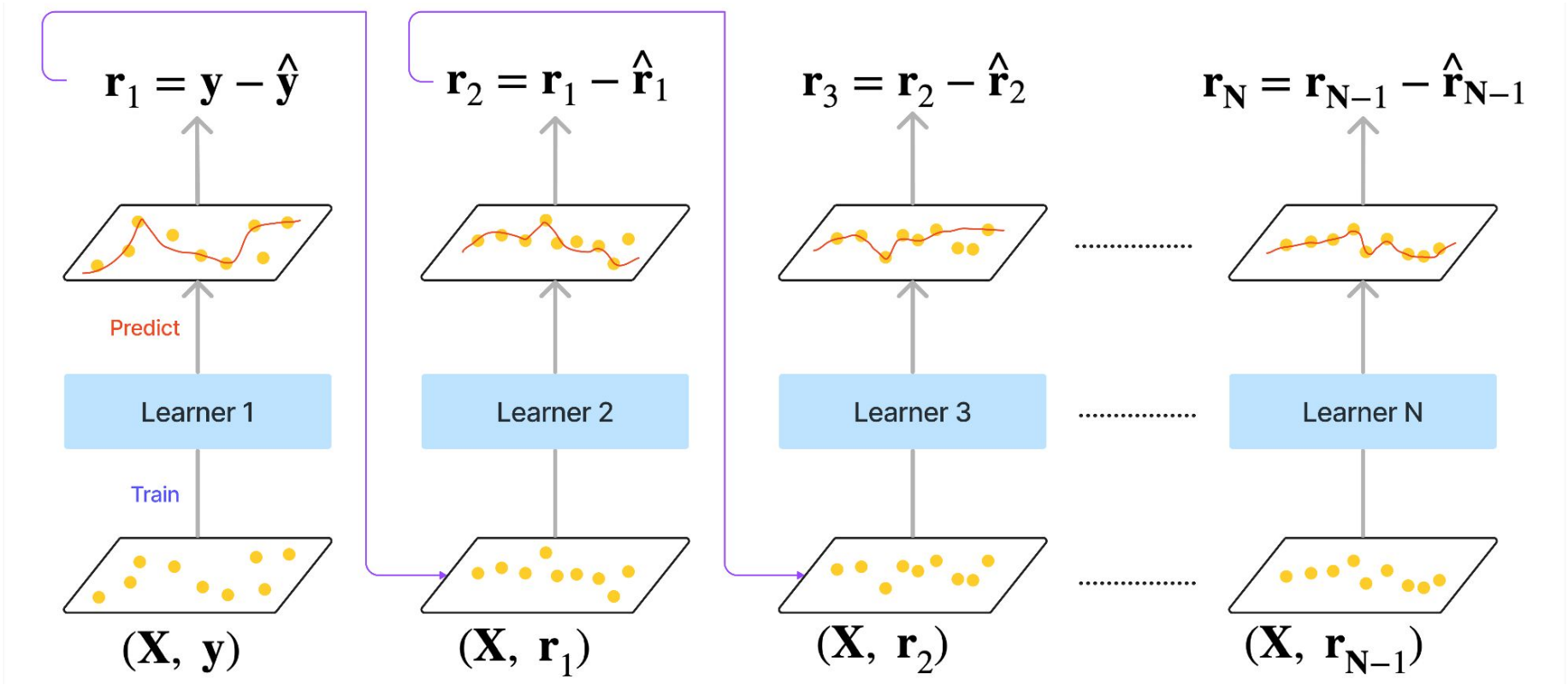
$$-\left(\frac{dT}{dx}\right) = \frac{4\pi n_e z^2 e^4}{m_e v^2}\left[\ln\frac{2m_e v^2}{I} - \ln(1-\beta^2) - \beta^2 - \delta - U\right],$$

$$W_1(P,Q) = \int_{-\infty}^{\infty} |F(x) - G(x)|dx \qquad F(x) = P(X \le x), \quad G(x) = Q(X \le x)$$

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}D_{KL}(P \parallel M) + \frac{1}{2}D_{KL}(Q \parallel M) \qquad D_{KL}(P \parallel Q) = \sum_{i=1}^{n} p_i \log\frac{p_i}{q_i}$$

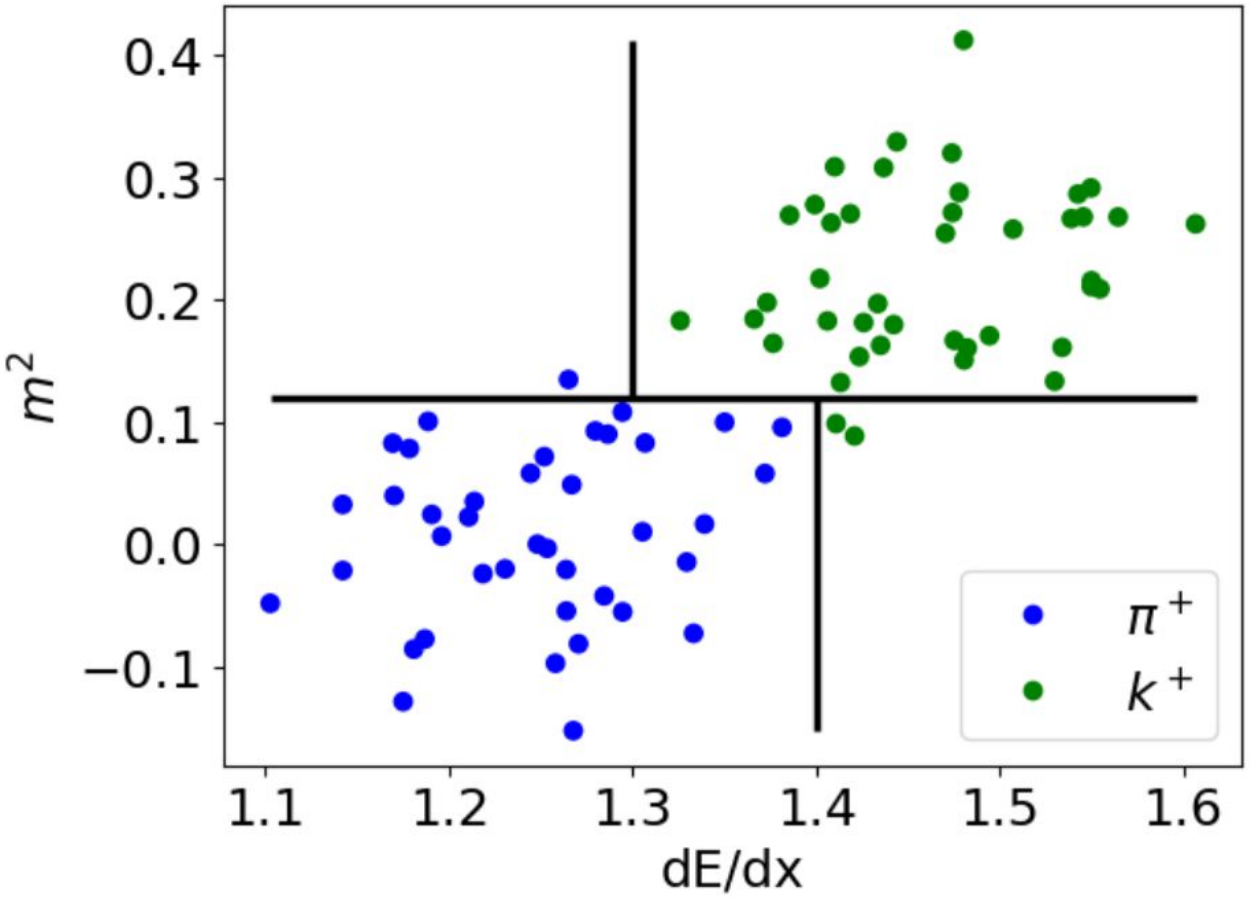
Gradient Boosting

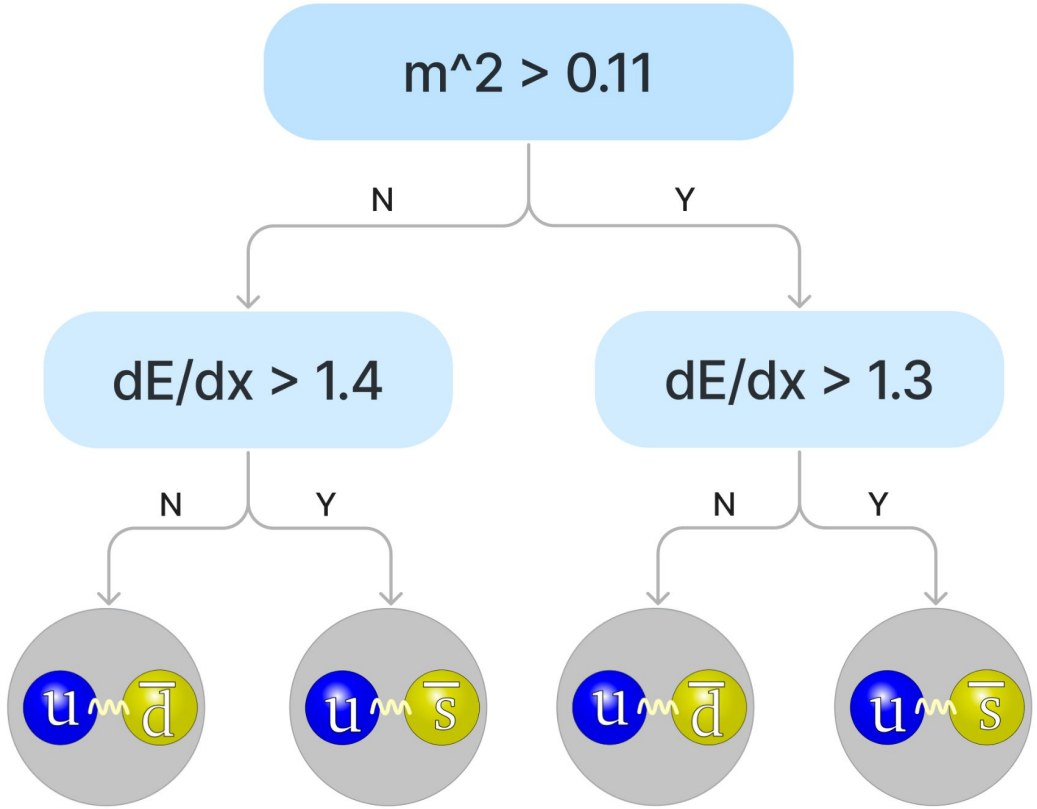
Gradient boosting is a machine learning technique which combines weak learners into a single strong learner in an iterative fashion



Gradient Boosted Decision Tree

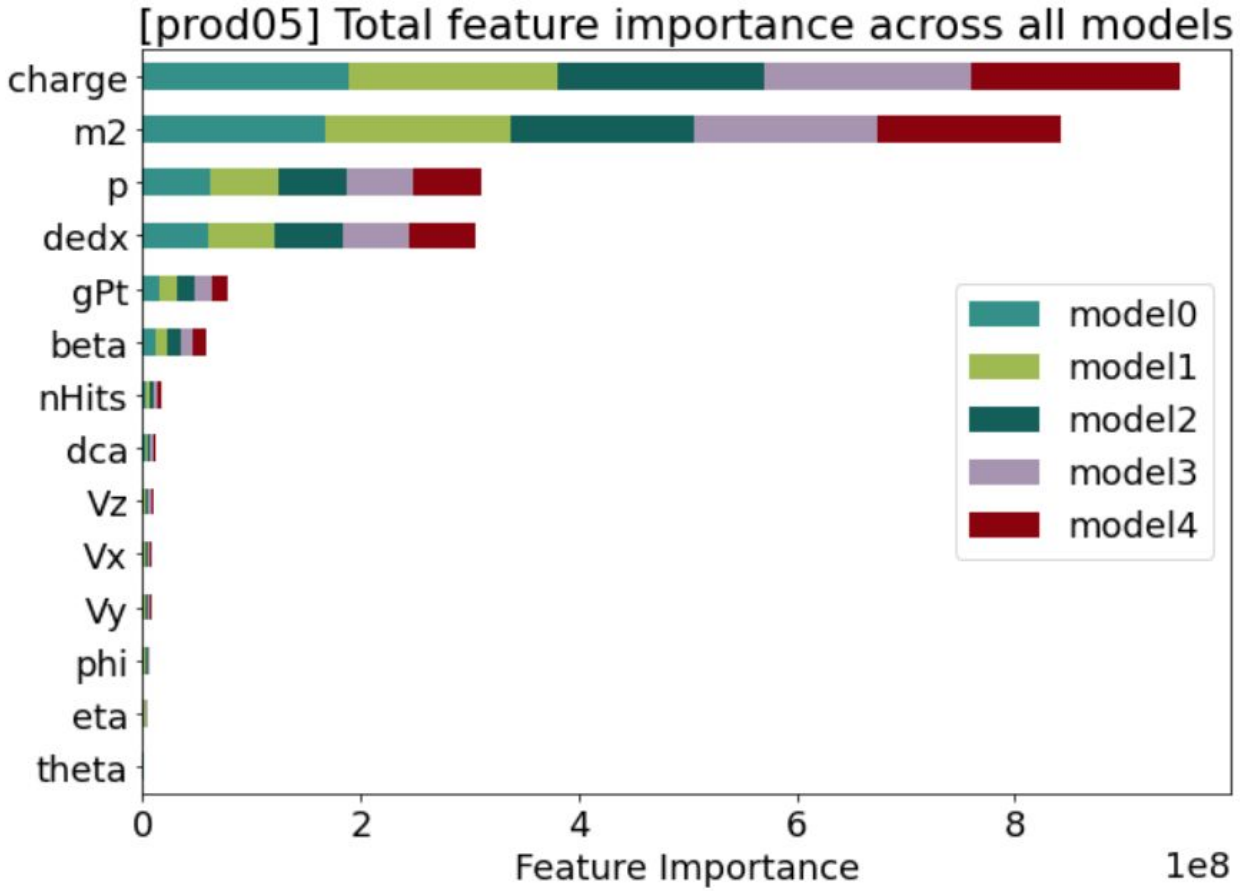
Gradient Boosted Decision Tree (GBDT) uses decision trees as weak learner. They can be considered as automated multilevel **cut-based** analysis





XGBoost Model Interpretation. Feature Importance

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



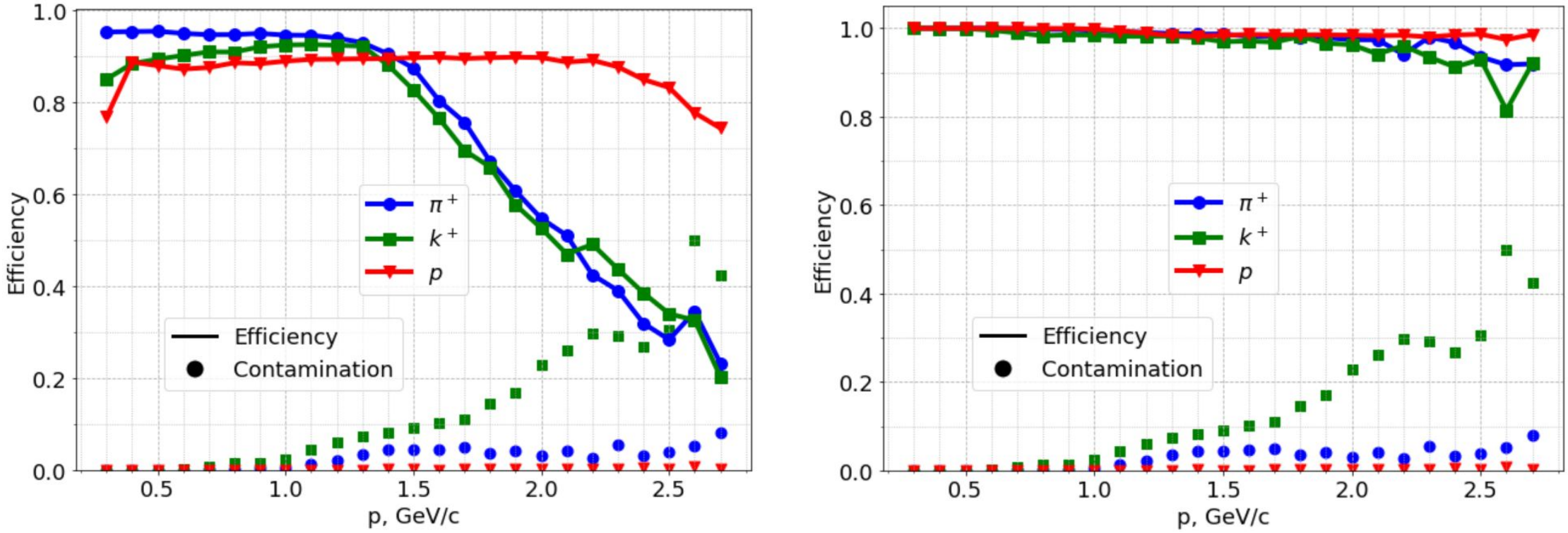
This approach are sensitive when input variables are correlated, and may lead for instance to unreliability in the importance ranking

Baseline PID at MPD - N-sigma

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

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

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



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

in Bi+Bi collisions at 9.2 GeV