Machine Learning Application for Focusing Aerogel Detectors in High Energy Physics

Student: Foma Shipilov
Research Advisor: Fedor Ratnikov
Co-Advisor: Aleksandr Katrutsa

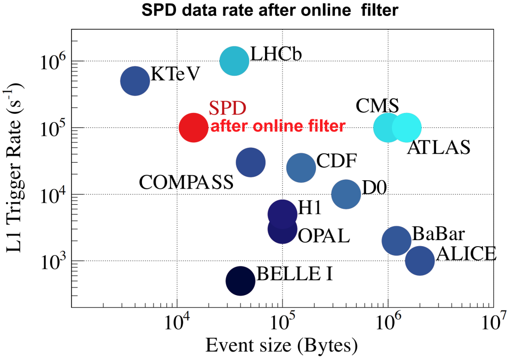
Data Science

April, 2025

Skoltech

Introduction: Spin Physics Detector

- A universal detector proposed by the SPD collaboration at NICA
- Study the Drell–Yan (DY) processes, J/Ψ production processes, elastic reactions, spin effects in one and two hadron production processes, polarization effects in heavy ion collisions, and more
- The SPD is a medium energy experiment, offering unique possibilities of beam operation and bridging the gap between the low-energy measurements, e.g. ANKE-COSY and the high-energy measurements, such as RHIC
- High luminosity up to 10^{32} cm^{-2}s^{-1} and free-flowing (triggerless) running mode

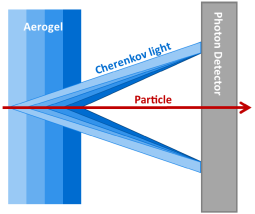


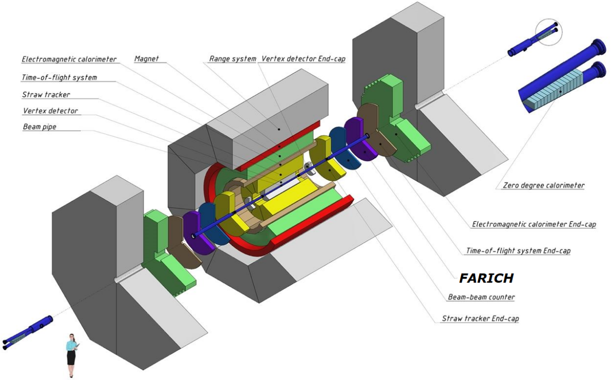
Introduction: FARICH

Focusing Aerogel Ring Imaging Cherenkov

Compact particle ID detector

- Trade off between resolution and number of Cherenkov photons
- Improve resolution by using layers of aerogel with varying refractive index





https://nica.jinr.ru/projects/spd.php

Aim and Objectives

- Particle identification (PID): the goal is to identify the most probable particle type
- Ambiguous with FARICH only, momentum p from straw tracker is needed

$$\beta = \frac{1}{n \cos \theta_c}, \quad m_0 = \frac{p\sqrt{1-\beta^2}}{\beta} \tag{1}$$

- In terms of optimization, we seek a solution to a classification problem

$$\mathbb{E}_{(X,y)\sim q(X,y)}\left[Q\big(f_W(X), y\big)\right] \to \max_W$$

- End-to-end neural networks can combine both steps from (1)
- Potentially better performance by optimizing a more appropriate objective

Objectives: develop and implement a variety of both β-outputting and end-to-end models, evaluate on a synthetic dataset, analyze and compare with the baseline

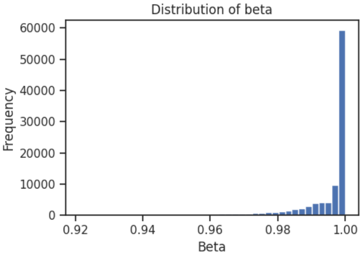
Preliminary Dataset

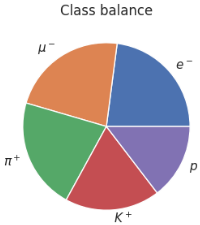
Provided input:

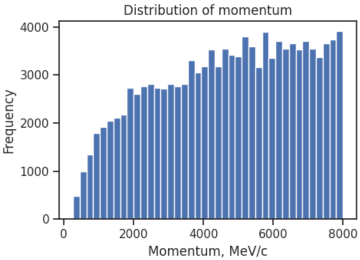
- Track parameters: x_p, y_p, z_p – coordinates of the particle when entering aerogel detector, θ_p, ϕ_p – polar angle and azimuth of the direction of travel, momentum p
- Photon hits: x_c, y_c, z_c – coordinates of triggered pixels in the photosensitive matrix

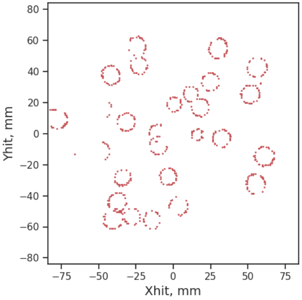
Expected output:

- PID statistics
- Particle type









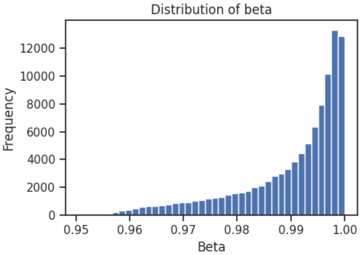
Production Dataset

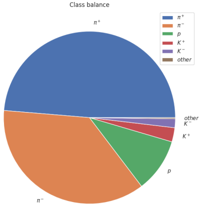
Provided input:

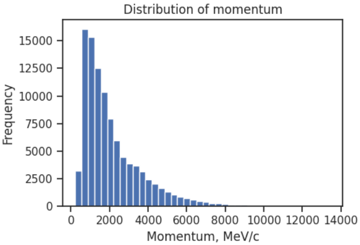
- Track parameters: x_p, y_p, z_p – coordinates of the particle when entering aerogel detector, θ_p, ϕ_p – polar angle and azimuth of the direction of travel, momentum p
- Photon hits: x_c, y_c, z_c – coordinates of triggered pixels in the photosensitive matrix

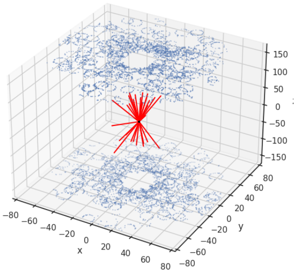
Expected output:

- PID statistics
- Particle type



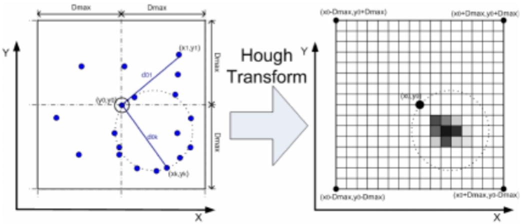






Preliminary Baseline: Hough Transform

- Based on the RICH reconstruction from the CBM experiment at FAIR
- Utilizes Hough transform for ellipses, more precisely the Taubin method
- Outputs estimated parameters of the ellipse, such as a, b – semi-axes, x, y – center coordinates, and ϕ – angle of rotation
- Does not account for refraction, does not use the information from the straw tracker, such as θ_p and ϕ_p

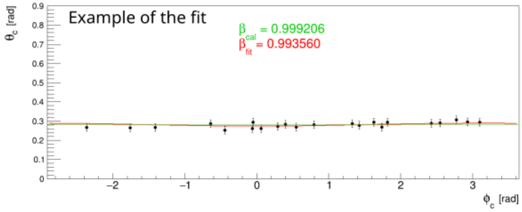


Production Baseline: θ_c by ϕ_c Fit

- Based on fitting

$$\theta_c(\phi_c|\beta, n, \theta_t) = \arccos\left(\frac{1}{n\beta}\right) + \arccos\left(n(1-(\mathbf{n}_0\mathbf{n}_\gamma)^2)\right) + (\mathbf{n}_0\mathbf{n}_\gamma)\sqrt{1-n^2(1-(\mathbf{n}_0\mathbf{n}_\gamma)^2)}$$

- Outputs estimated β – particle velocity
- Accounts for refraction, uses track information
- Is incomplete / broken (we have no control here)



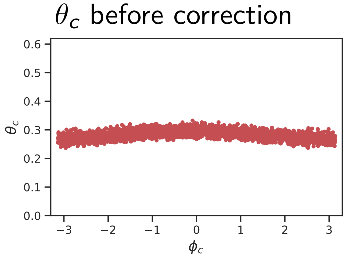
Refraction Correction

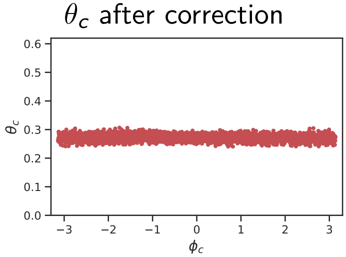
First order approximation

$$\alpha \approx \beta - (n-1)\tan\beta \tag{2}$$

Or fixed-point iteration

$$\sin\alpha_{k+1} = \left(n\sqrt{1+\left(\frac{d}{r-h\tan\alpha_k}\right)^2}\right)^{-1} \tag{3}$$





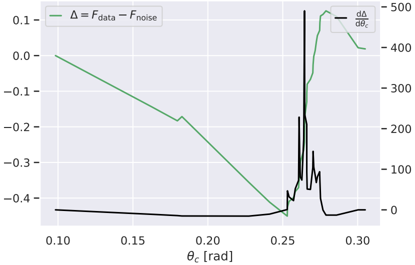
Algorithmic Methods (Not Learnable)

Median:

- Compute θ_c distribution
- Take a median value $\hat{\theta}_c$

MLE:

- Compute θ_c distribution
- Drop unphysical velocities $\beta = v/c \geqslant 1$
- Construct eCDF F_{data}
- Take a numerical derivative and find its peak $\hat{\theta}_c = \arccos(1/n\hat{\beta})$

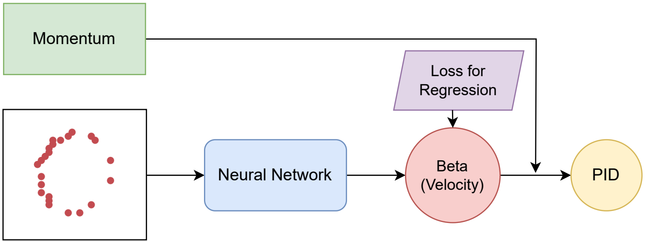


MLE example

Machine Learning Models: Beta NN

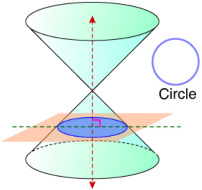
- Algorithmic models output $\hat{\theta}_c$ that can be converted to the rest mass of the particle using momentum p from the straw tracker
- The underlying task is regression
- A simple approach is to train a neural network to predict particle velocity β and then convert it into mass

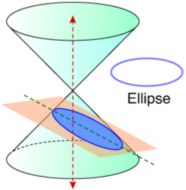
We implement a neural regressor that learns β by optimizing MSE objective.

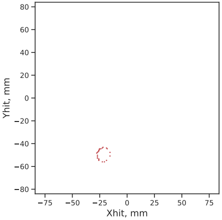


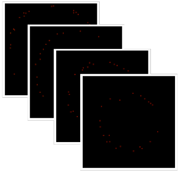
Data Transform

Simple reproject + Fourier features





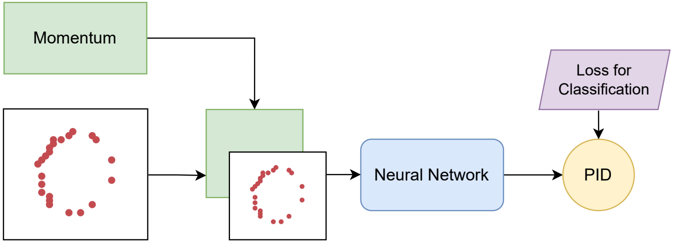




Machine Learning Models: Extra Channel NN

- The objective of RICH detector is to separate different particle types, in particular π/K separation
- ML end-to-end models are able to skip the intermediate step of computing $\hat{\theta}_c$
- Moreover, the regression task is not well suited for an ML model because of the significant value imbalance

A neural classifier can receive **momentum** p as an extra channel in the input tensor along with **hit coordinates** x_c, y_c, optimizing cross entropy objective directly.

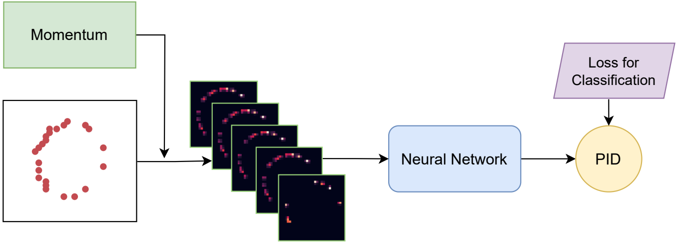


Machine Learning Models: Renormalize NN

The non-linear dependence of the ellipse dimensions on p can be disentangled:

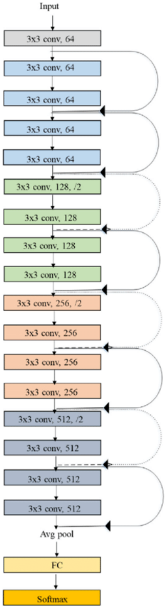
$$\cos\theta_c = \frac{1}{n}\sqrt{m_0^2/p^2 + 1} \tag{4}$$

- Use (4) with rest masses m_0 of all N particle types in the data to transform θ_c into N rings, diameter of the ring corresponding to true particle type (class) is constant and known



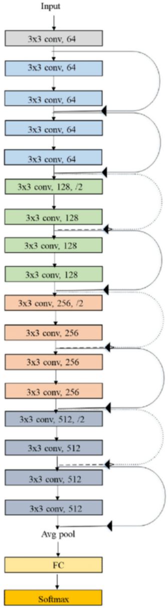
Training Details (Preliminary)

- 3 configurations
  - Beta NN
  - Extra Channel NN
  - Renormalize NN
- ResNet-18 CNN architecture
  - Changed input conv, max pooling and classifier head to accommodate input and output data formats
- Input format: $N \times 32 \times 32$ tensor
- Output format: 5 class classification
- Trained for 800000 samples with Adam optimizer, cosine annealing scheduler, learning rate $3 \cdot 10^{-4}$ and batch size 128

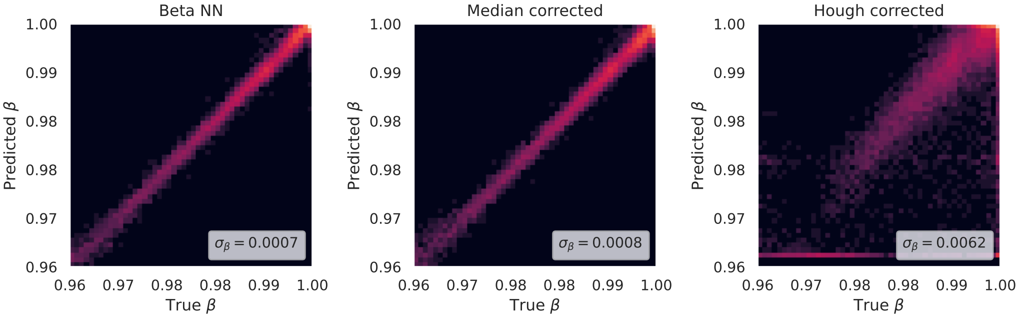


Training Details (Production)

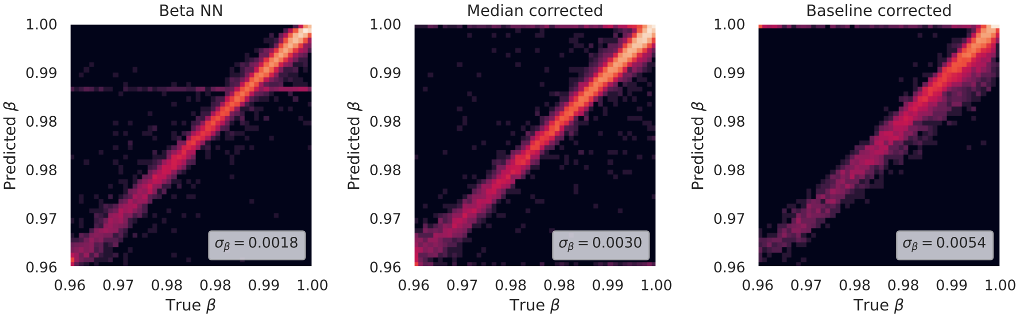
- Output format: binary classification (π/K)
- Countering class imbalance 20/1
  - Positive class weight
  - Downsampling
  - No adjustment (control)
- For β-outputting NN MSE can be weighted according to class
- The best performing training procedure (AUCROC on validation) evaluated on the test set



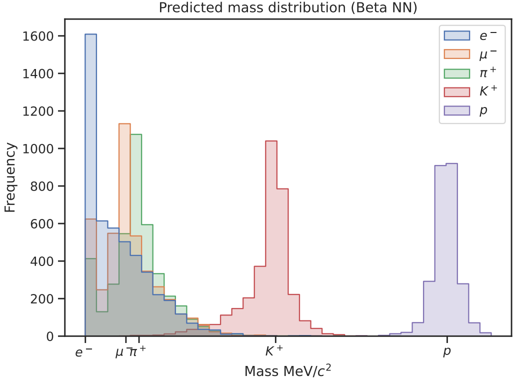
Results (Preliminary)

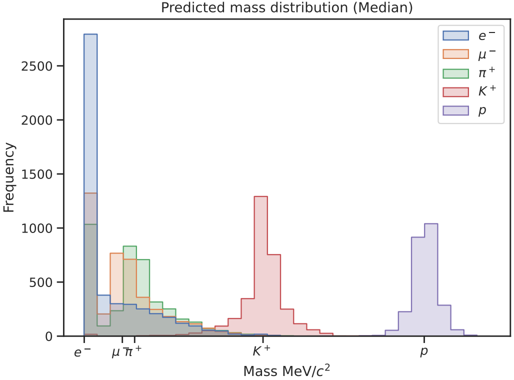


Results (Production)

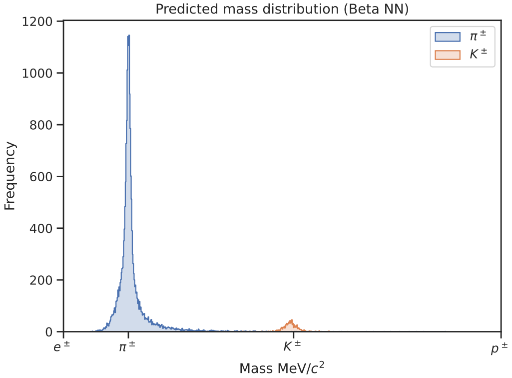


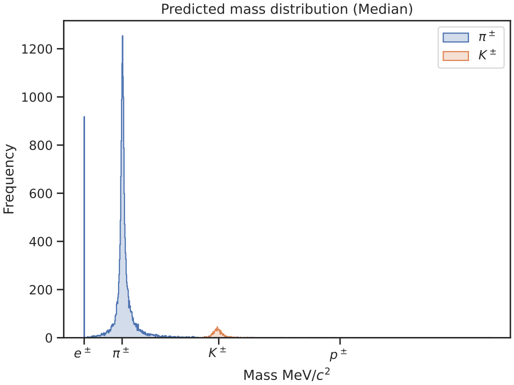
Results (Preliminary)



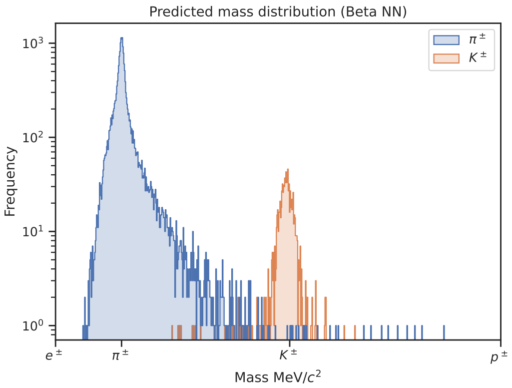


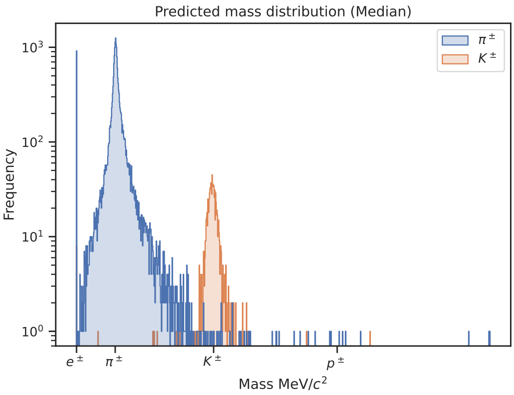
Results (Production)



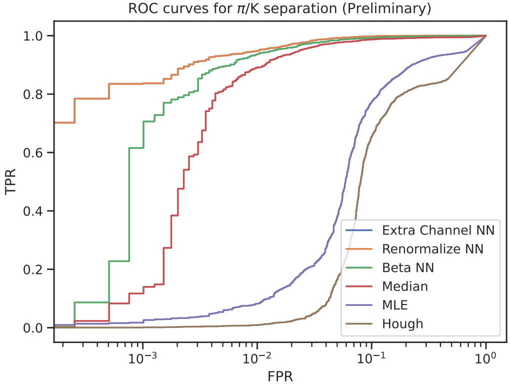


Results (Production)

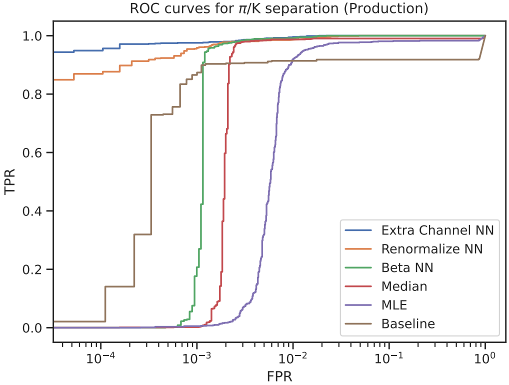




Results



ROC curves for π/K separation (Preliminary)
TPR
FPR
1.0
0.8
0.6
0.4
0.2
0.0
10^{-3}
10^{-2}
10^{-1}
10^{0}
Extra Channel NN
Renormalize NN
Beta NN
Median
MLE
Hough



ROC curves for π/K separation (Production)
TPR
FPR
1.0
0.8
0.6
0.4
0.2
0.0
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
Extra Channel NN
Renormalize NN
Beta NN
Median
MLE
Baseline

Results

Method	π, K TPR@FPR$= 10^{-2}$	π, K AUROC	5 class Accuracy	σ_β
Ex Channel NN	**0.947**	**0.998**	**0.68**	N/A
Renormalize NN	0.947	0.998	0.67	N/A
Beta NN	0.936	0.995	0.64	0.0007
Median	0.891	0.989	0.65	0.0008
MLE	0.082	0.883	0.53	0.0018
Hough	0.009	0.815	0.50	0.0062

Table 1. Preliminary dataset

Method	TPR@FPR$= 10^{-2}$	AUROC	AP	σ_β
Ex Channel NN	**0.996**	**0.9998**	**0.997**	N/A
Renormalize NN	0.986	0.9996	0.993	N/A
Beta NN	0.991	0.999	0.919	0.0018
Median	0.987	0.989	0.870	0.0030
MLE	0.919	0.978	0.733	0.0037
Baseline	0.914	0.923	0.905	0.0054

Table 2. Production dataset. Only binary π, K classification is considered

Discussion of Results

- Beta NN and Median – nearly identical performance
- Indicates the limit of β-based approaches
- End-to-end models are also close to each other; qualitative improvement over β-based models
- Almost perfect π/K separation on production data
- Machine learning-based approaches demonstrate flexibility and achieve better quality

Discussion of Results

Problems/Difficulties

- ResNet-18 is a heavy model, its real time use in the SPD DAQ is challenging
- Prior data transforms (reprojection, Fourier features extraction, renormalization) are even more computationally expensive, training and inference are CPU-bound
  - We implemented data parallelism and achieved 7x speed up

Future Plans

- Architecture optimization
  - knowledge distillation, pruning and quantization
- Integration into the SPD DAQ pipeline
- FARICH fast simulation

Conclusions

- A novel end-to-end FARICH reconstruction model was developed
- The results were presented at DLCP2024 and ICPPA-2024 conferences (preliminary dataset)
- A paper was published in Moscow University Physics Bulletin journal (Section 3). The journal is peer-reviewed, published in English by Springer and indexed by Scopus and WoS
- Planned work since Status Review – all done
- Production dataset results will be reported at IX SPD Collaboration Meeting (12-16 May 2025)

Thx!

Skoltech