ML-based fast simulation of FARICH responses

F.Shipilov[1], A.Barnyakov[2,3], V.Bobrovnikov[2], A.Ivanov[4], S.Kononov[5], F.Ratnikov[1]

(1) HSE University (2) Budker Institute of Nuclear Physics (3) Novosibirsk State Technical University (4) JINR (5) Novosibirsk State University

foma@shipilov.ru

LAMBDA • HSE

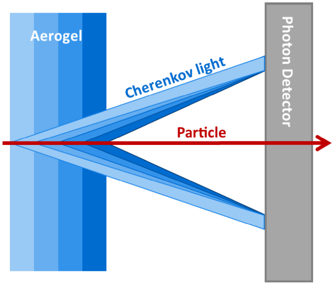
November 26, 2025

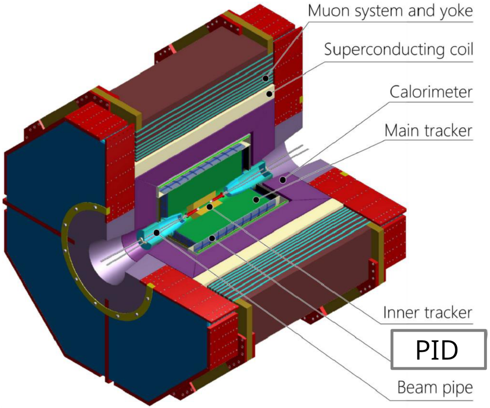
Introduction: FARICH

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

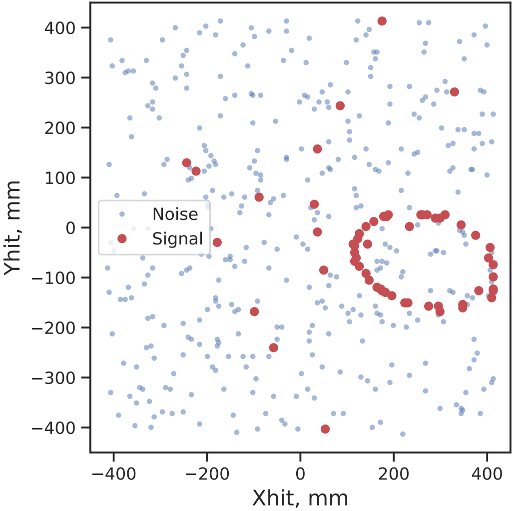




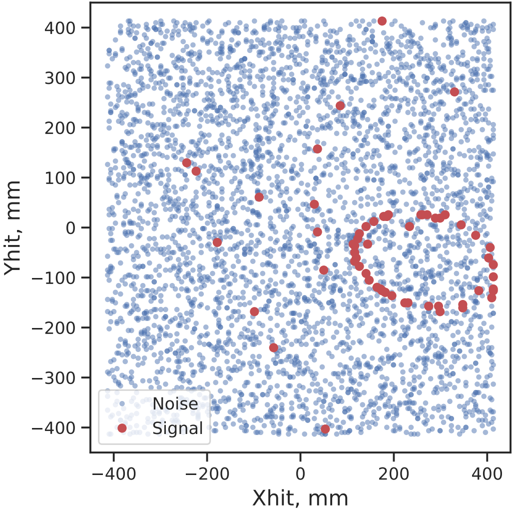
Barnyakov et al., NIM A553 (2005) 70.

Example Event

Optimistic DCR 100 kHz/mm^2



Pessimistic DCR 1 MHz/mm^2



Online Noise Filtering

Online Noise Filtering

Motivation

- SiPMs are noisy with insufficient cooling
- DAQ cannot handle capturing and storing all hits
- **Solution:** Software trigger to reduce data rate
- Drop background hits, preserve signal hits

Requirements

- Online mode: no tracker info
- Do not introduce any bias in RECO hypothesis
- Efficiency $\geqslant 95\%$
- At least 10x noise reduction

Approach 1

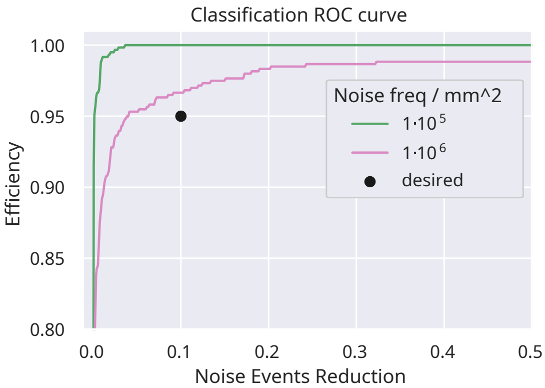
Classification

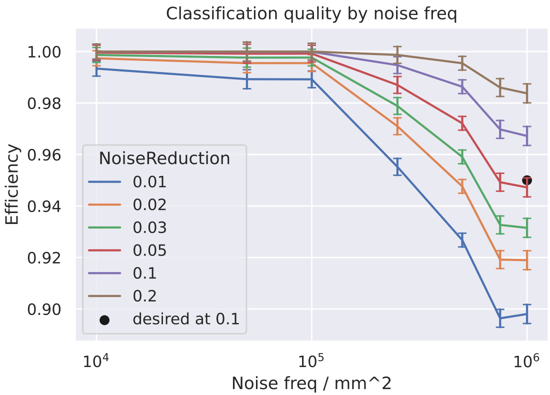
- create a dataset of events with / without signal (positive event $\geqslant 10$ signal photons)
- train Convolutional Neural Network (CNN) to perform binary classification

Metrics:

- Efficiency (events) $= \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
- Noise Events Reduction $= \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}$

Results: Classification





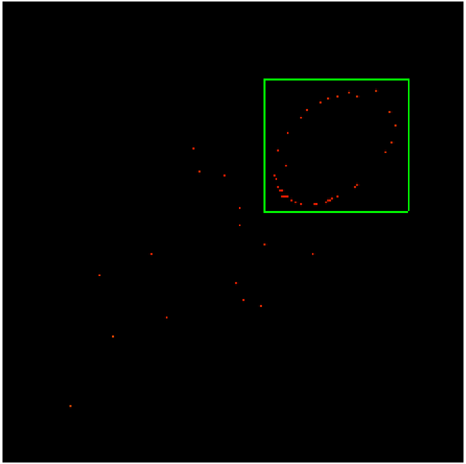
Approach 2

Bounding box (BBox) regression

- compute ground truth bboxes for signal ellipses
- train CNN to extract bbox coordinates

Metrics:

- Efficiency (BBox area) $= \frac{|B \cap B^{\text{gt}}|}{|B^{\text{gt}}|}$
- Reduction (BBox area) $= \frac{|B|}{|B_{\text{max}}|}$



Ground truth bbox example

BBox Regression Objective

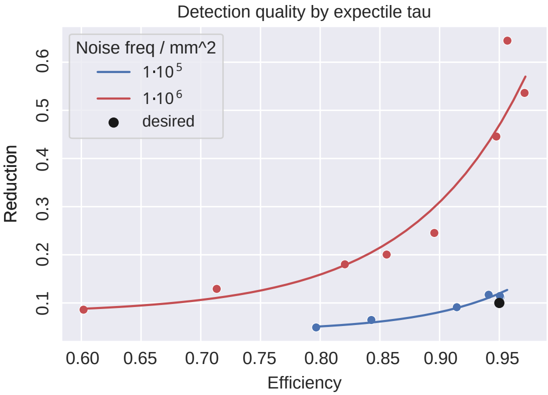
- Simple MSE tends to throw out signal hits
- Solution: use weighted version **Expectile Loss** [1]

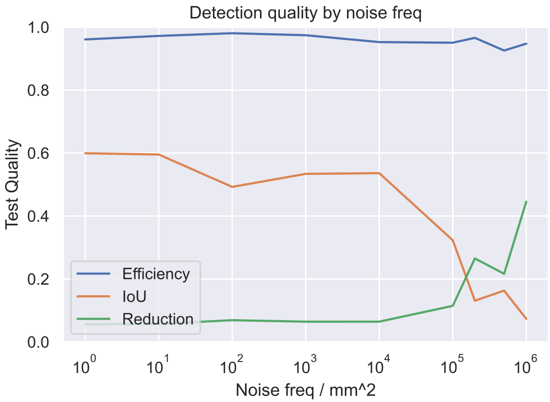
$$\text{quantile:} \quad \operatorname{argmin} \left\{ \sum_{i=1}^{n} w(y_i - q_i, \tau) \cdot |y_i - q_i| \right\},$$

$$\text{expectile:} \quad \operatorname{argmin} \left\{ \sum_{i=1}^{n} w(y_i - q_i, \tau) \cdot (y_i - q_i)^2 \right\},$$

$$\text{where } w(x, \tau) = \begin{cases} 1 - \tau, & x \leqslant 0 \\ \tau, & x > 0 \end{cases}$$

Results: BBox Regression





Summary: Online Noise Filtering

- ResNet-18 CNN provides a significant level of denoising with high efficiency
- It is possible to achieve even higher noise reduction by performing bounding box regression on signal ellipses for positive events
- Real time performance is possible with optimizations

Part 2

Fast Simulation

Fast Simulation

- Detector simulation: the goal is to speed up event generation by means of a surrogate model, avoiding Monte-Carlo simulation
- Given particle rest mass m_0, momentum p and other parameters, obtain realistic samples of Cherenkov angles θ_c and polar angles ϕ_c

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

- Generative adversarial networks can bypass the low-level photon generation while preserving the necessary fine details
- Can be made sufficiently fast and light-weight; easy to parallelize

Objectives: develop a generative neural network to reproduce raw detector responses, evaluate on a synthetic dataset, analyze and compare with the baseline

Assumptions:

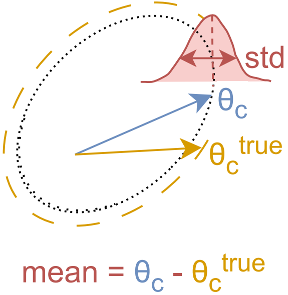
- Cherenkov angle θ_c is independent of azimuth ϕ_c (after accounting for refraction), ϕ_p (particle direction isotropy)

Learning:

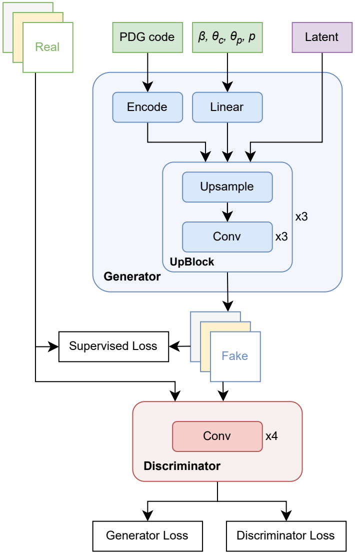
- fit a linear model to predict mean and std from particle parameters: $\beta^{\text{true}}, \theta_c^{\text{true}}, \theta_p^{\text{true}}, p^{\text{true}}$ and engineered features

Inference:

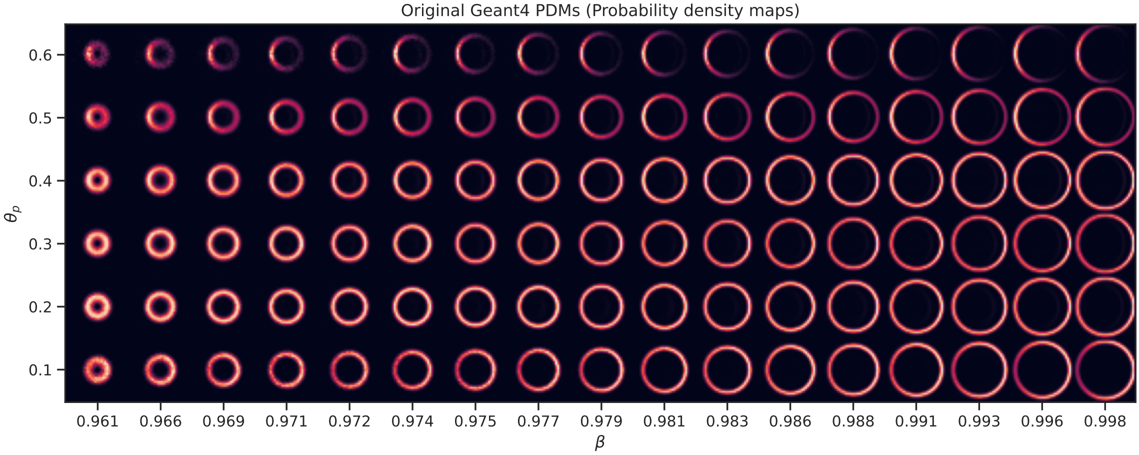
- sample θ_c from a normal distribution with mean, std
- sample ϕ_c from a uniform distribution



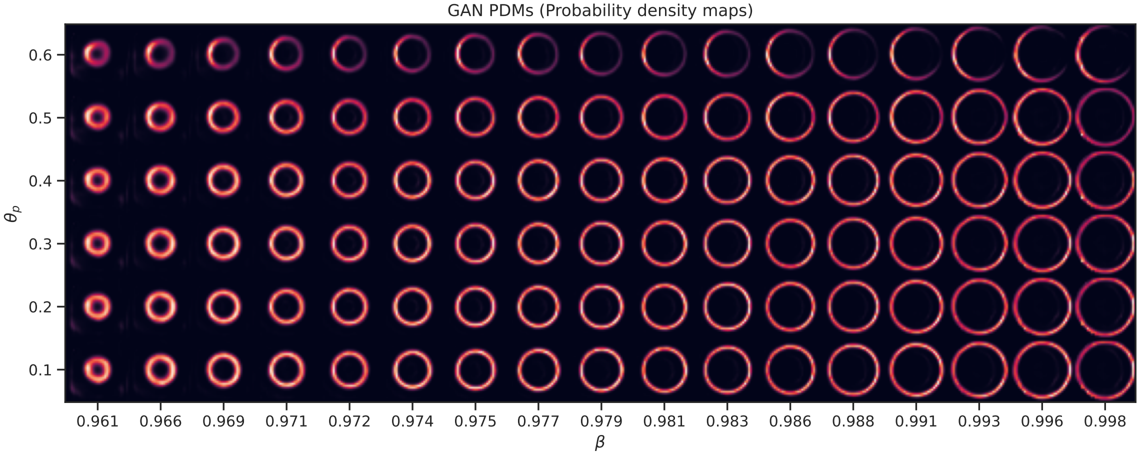
- GANs provide a sample-level mapping from a latent distribution to a complex real-world distribution
- A conditional GAN (cGAN) model is perfect for this task [2]
- **Input condition**: particle parameters $\beta^{\text{true}}, \theta_c^{\text{true}}, \theta_p^{\text{true}}, p^{\text{true}}$
- **Output**: is a projection of the detector response into a circle with refraction factored out
- Lightweight CNN architecture, 184k parameters



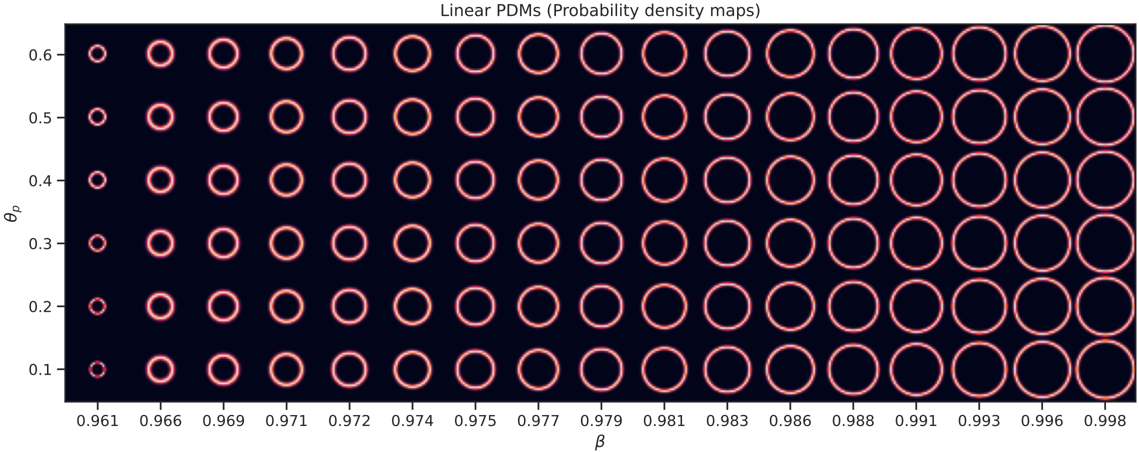
Original Geant4 data



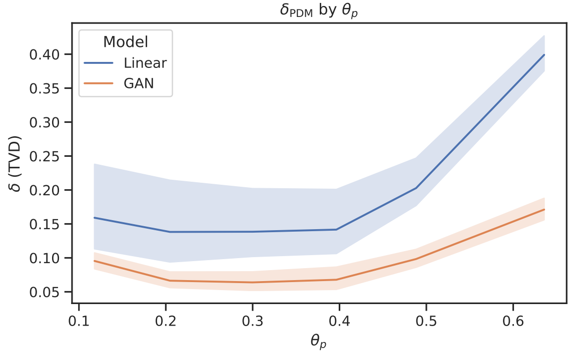
GAN simulation

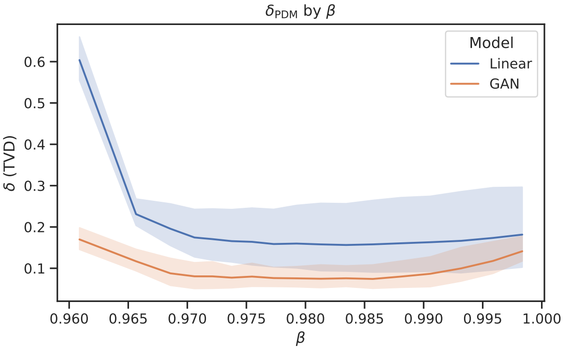


Linear simulation



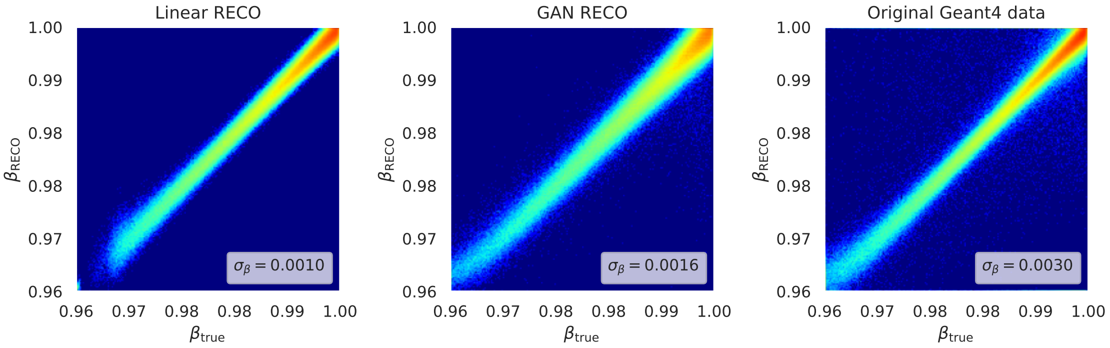
Results: Fast Simulation



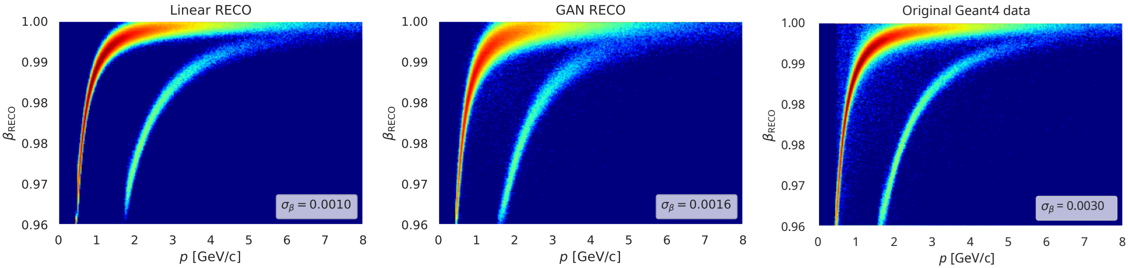


Method	$\delta_{\text{PDM}} \downarrow$
Linear	0.146 ± 0.001
cGAN	$\mathbf{0.105 \pm 0.001}$

Results: Fast Simulation



Results: Fast Simulation



Method	$\delta_{\text{RECO}} \downarrow$
Linear	0.29 ± 0.01
cGAN	$\mathbf{0.18 \pm 0.01}$

Simulation Summary

- cGAN outperforms the baseline by Total Variation Distance $\delta(P_{\text{Fake}}, P_{\text{Geant}})$ for PDMs and reconstructed β distributions. In particular, GAN is more accurate for low momenta
- Generator neural network is very lightweight. A *significant* speed up is achieved compared to Monte-Carlo

Problems/Difficulties

- projected PDMs should be converted to raw detector responses
  - PDM quantization is performed anyway, probably no significant performance penalty
  - another option is tweaking GAN to generate ellipses directly
- GAN underestimates distribution tails
  - incentivize outlier event generation via an additional loss component

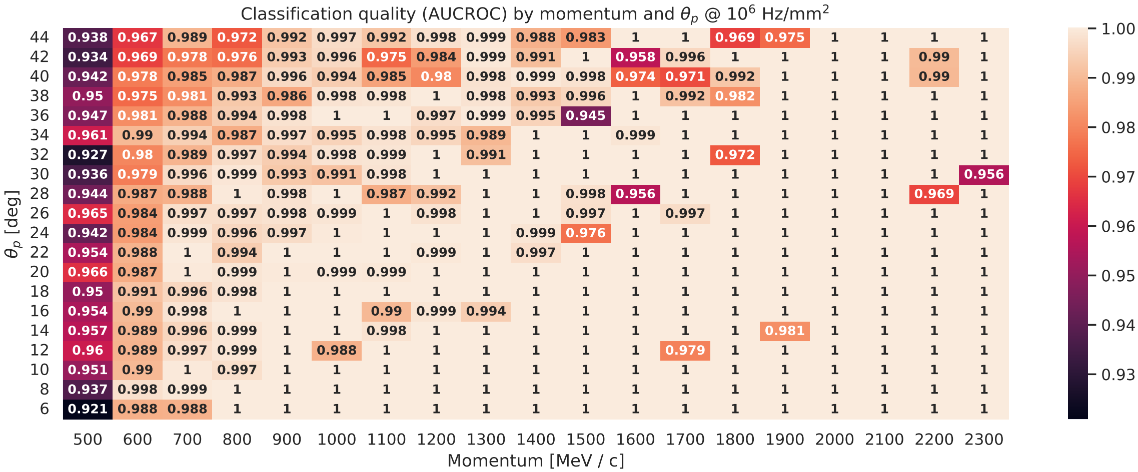
Acknowledgements

The presented scientific results have been prepared following the materials of Mirror Laboratories project of HSE University and Partner Organization.

References

[1] Arthur Charpentier. Quantile and Expectile Regresions. Erasmus School of Economics, May 2017.

[2] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. In CoRR 2014.


Classification quality (AUCROC) by momentum and θ_p @ 10^6 Hz/mm^2
44 0.938 0.967 0.989 0.972 0.992 0.997 0.992 0.998 0.999 0.988 0.983 1 1 0.969 0.975 1 1 1 1
42 0.934 0.969 0.978 0.976 0.993 0.996 0.975 0.984 0.999 0.991 1 0.958 0.996 1 1 1 1 0.99 1
40 0.942 0.978 0.985 0.987 0.996 0.994 0.985 0.98 0.998 0.999 0.998 0.974 0.971 0.992 1 1 1 0.99 1
38 0.95 0.975 0.981 0.993 0.986 0.998 0.998 1 0.998 0.993 0.996 1 0.992 0.982 1 1 1 1 1
36 0.947 0.981 0.988 0.994 0.998 1 1 0.997 0.999 0.995 0.945 1 1 1 1 1 1 1 1
34 0.961 0.99 0.994 0.987 0.997 0.995 0.998 0.995 0.989 1 1 0.999 1 1 1 1 1 1 1
32 0.927 0.98 0.989 0.997 0.994 0.998 0.999 1 0.991 1 1 1 1 0.972 1 1 1 1 1
30 0.936 0.979 0.996 0.999 0.993 0.991 0.998 1 1 1 1 1 1 1 1 1 1 1 0.956
28 0.944 0.987 0.988 1 0.998 1 0.987 0.992 1 1 0.998 0.956 1 1 1 1 1 0.969 1
26 0.965 0.984 0.997 0.997 0.998 0.999 1 0.998 1 1 0.997 1 0.997 1 1 1 1 1 1
24 0.942 0.984 0.999 0.996 0.997 1 1 1 1 0.999 0.976 1 1 1 1 1 1 1 1
22 0.954 0.988 1 0.994 1 1 1 0.999 1 0.997 1 1 1 1 1 1 1 1 1
20 0.966 0.987 1 0.999 1 0.999 0.999 1 1 1 1 1 1 1 1 1 1 1 1
18 0.95 0.991 0.996 0.998 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
16 0.954 0.99 0.998 1 1 1 0.99 0.999 0.994 1 1 1 1 1 1 1 1 1 1
14 0.957 0.989 0.996 0.999 1 1 0.998 1 1 1 1 1 1 1 0.981 1 1 1 1
12 0.96 0.989 0.997 0.999 1 0.988 1 1 1 1 1 1 0.979 1 1 1 1 1 1
10 0.951 0.99 1 0.997 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
8 0.937 0.998 0.999 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
6 0.921 0.988 0.988 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
θ_p [deg]
500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100 2200 2300
Momentum [MeV / c]
1.00 0.99 0.98 0.97 0.96 0.95 0.94 0.93

Backup: Detailed Fast Simulation

Backup: Detailed Fast Simulation