Studying the Tracking Resolution of the MUSE Straw Tube Trackers

Kyle Salamone

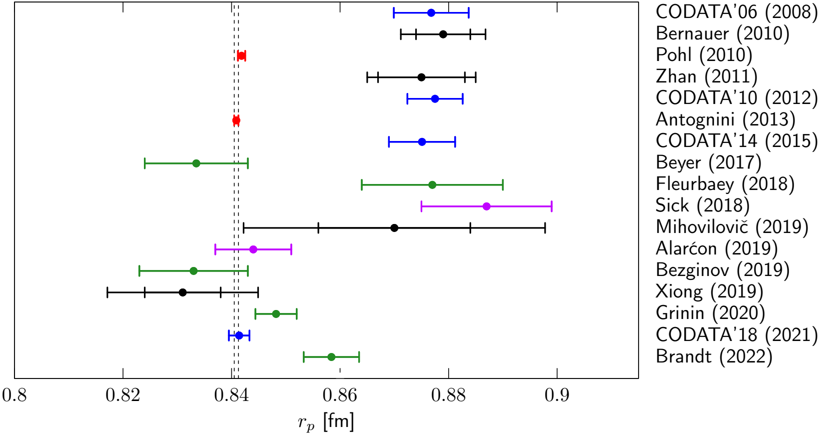
Center for Frontiers in Nuclear Science, Stony Brook University

August 15, 2025

This material is based upon work supported by the National Science Foundation under NSF Grant PHY-2412703. The MUSE experiment is supported by the Department of Energy, NSF, PSI and the US-Israel Binational Science Foundation.

The MUon Scattering Experiment (MUSE)

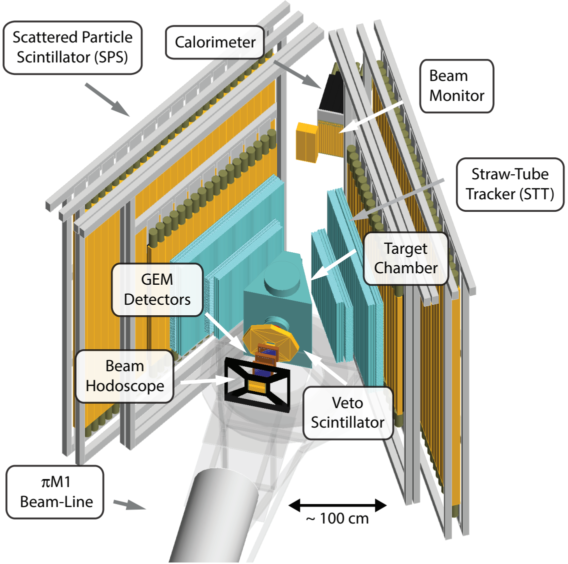
- 2010: CREMA extract r_p through muonic hydrogen spectroscopy
  - $\sim 7.9\sigma$ from average ep scattering value at time
- Birth of Proton Radius Puzzle



MUSE

- The MUon Scattering Experiment (MUSE) was directly inspired by the proton radius puzzle
- Goals:
  - Precision measurement of r_p via ep and μp scattering
  - Precision study of TPE in ep and μp scattering
  - Direct test of lepton universality
- Housed at the πM1 beamline at the Paul Scherrer Institute

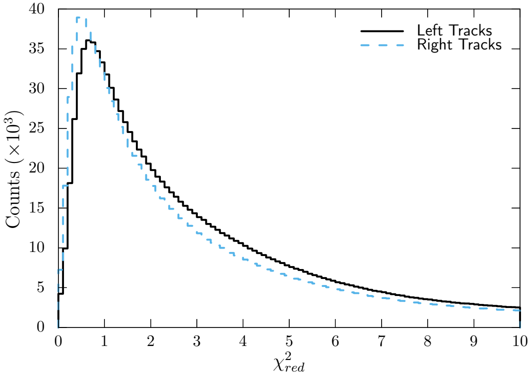
- θ acceptance: $20 - 100^\circ$
- $\pi M1$ Beam Line:
  - p$\in 115, 160, 210$ MeV/c
  - Mixed beam of e, μ, π
  - Both polarities of particles!

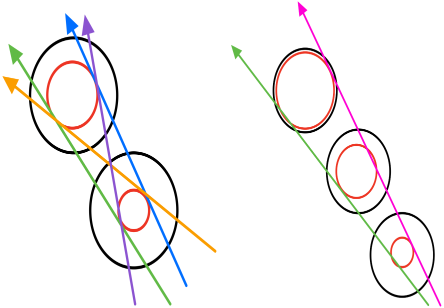


The Straw Tube Trackers (STT)

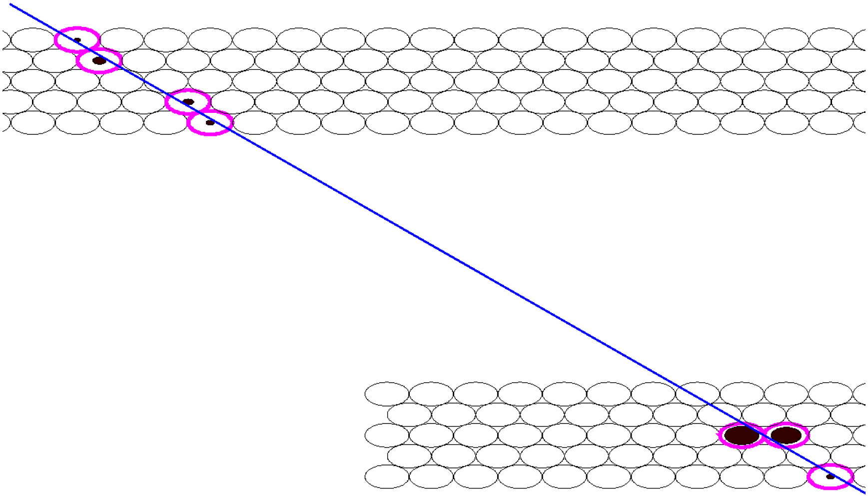
- Primary scattered particle tracking detector in MUSE
- Mirrored setup:
  - 20 planes of straws (10 horizontal, 10 vertical)
    - Vertical planes: θ
    - Horizontal planes: ϕ
  - Smaller front chamber, larger rear chamber
  - 5.1mm straw radius, 60 and 90 cm long
  - $\sim$ 3000 straws total

- Process:
  - Filter out noise hits/group tracks together
  - Parametrize track using spherical coordinates → 4 free parameters
  - Minimize χ^2 of track to hits (represented by cylinders)
  - Difficulty: "Left-Right Ambiguity"

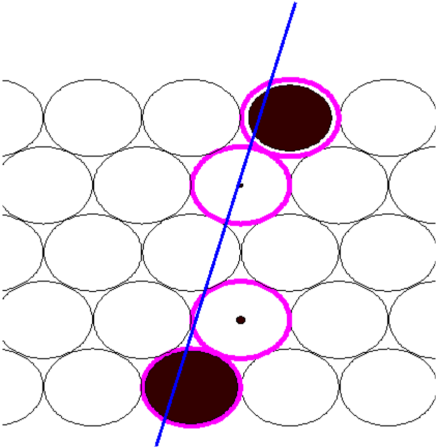




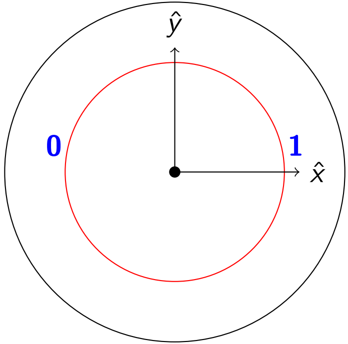
STT Tracking: Sample Event



STT Tracking: Sample Failure Event



- As discussed last time: ML possibly can be used to assist
- Idea of NN:
  - Work in local frame of straw (top down view)
  - Find which "side" simulated track passes on
  - Output: binary left/right ($y \approx 0$ more often than not)

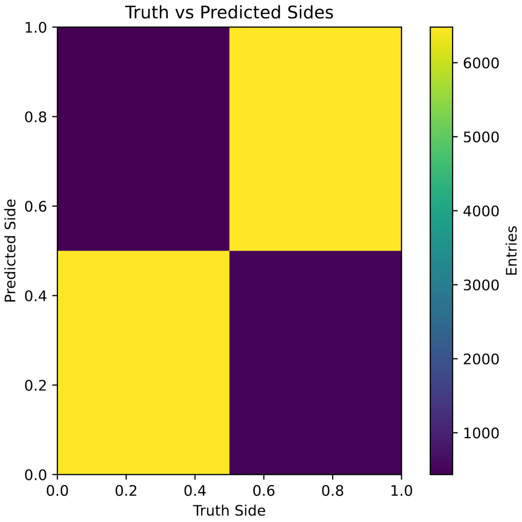


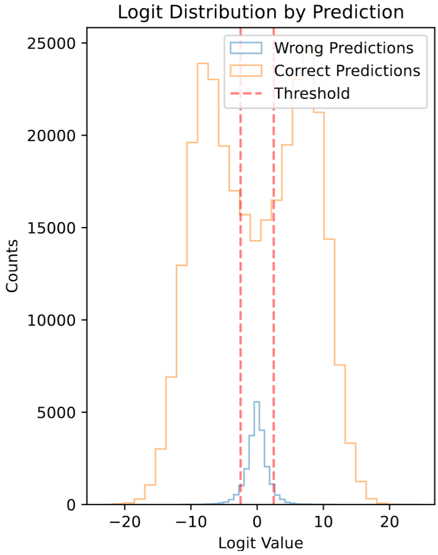
NN Structure

- Shallow network, bias=False because of batch norms
- Input: $(BATCH, 2, 10, 89)$ (binary 0/1 hit per straw, then the fired radii)
- Output: logit (strength of prediction and side) for each straw
- Loss function: BCEWithLogitsLoss (masked to only fired straws)

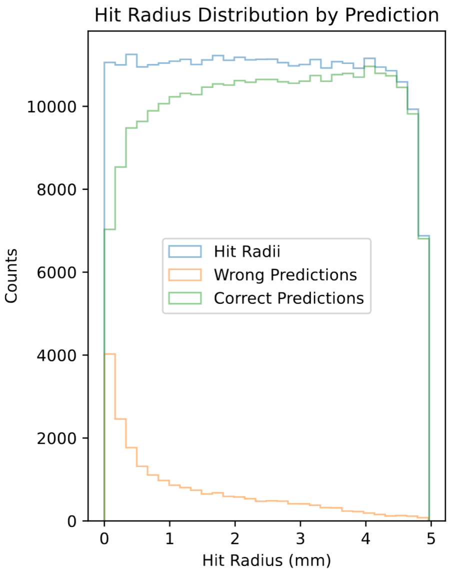
```python
self.sequence = nn.Sequential(
    nn.Conv2d(2, 8, kernel_size=(10,4), padding=(5,2), bias=False), # 0
    nn.BatchNorm2d(8), # 1
    nn.ReLU(inplace=True), # 2
    nn.Dropout2d(0.075), # 3

    nn.Conv2d(8, 16, kernel_size=(5,4), padding=(2,2), bias=False), # 4
    nn.BatchNorm2d(16), # 5
    nn.ReLU(inplace=True), # 6
    nn.Dropout2d(0.075), # 7

    nn.Conv2d(16, 1, kernel_size=2), # 8
    nn.ReLU(inplace=True), # 9
    nn.Dropout2d(0.075), # 10

    nn.Flatten(), # 11
    nn.Linear(900, 256, bias=False), # 12
    nn.ReLU(inplace=True), # 13
    nn.BatchNorm1d(256), # 14

    nn.Linear(256, input_size) # 15
)
```

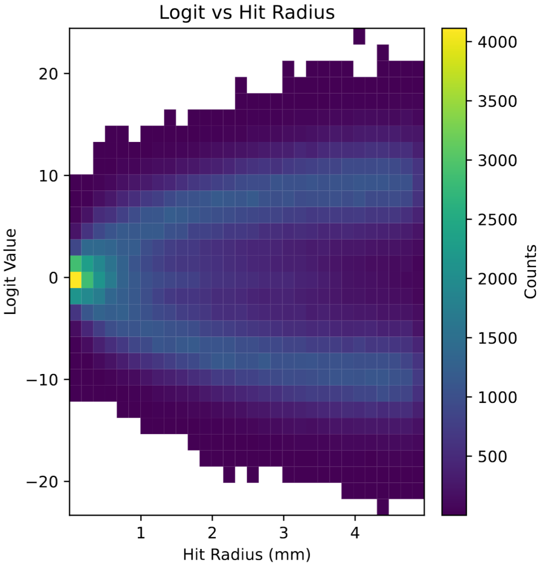
Results: Correlation and Logits





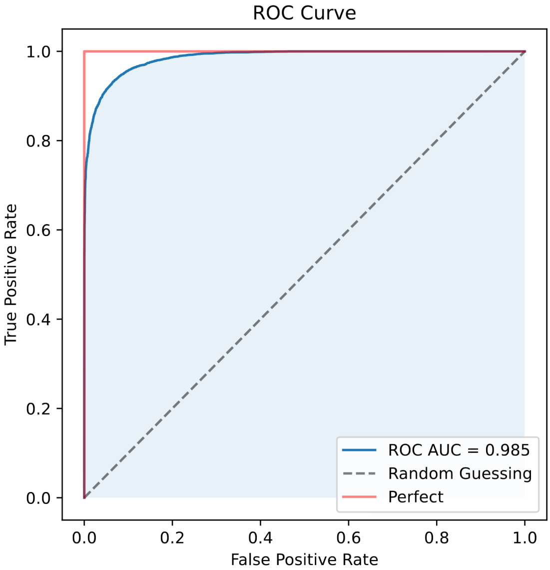
Hit Radii vs Correctness





Model Behavior

- Measure of performance: Receiver Operating Characteristic (ROC) Curve
  - Used primarily in medical diagnostics
  - Graphical way to gauge binary classifier performance
  - Higher area under curve (AUC) is better
- AUC of 0.985 - incredibly strong model at predicting left/right
- Cut on: $|\text{logit}| \geq 2.5$ and $r_{\text{drift}} \geq 1mm$ radii: 99.4% LR correct on validation set!



Implementation in Trackers

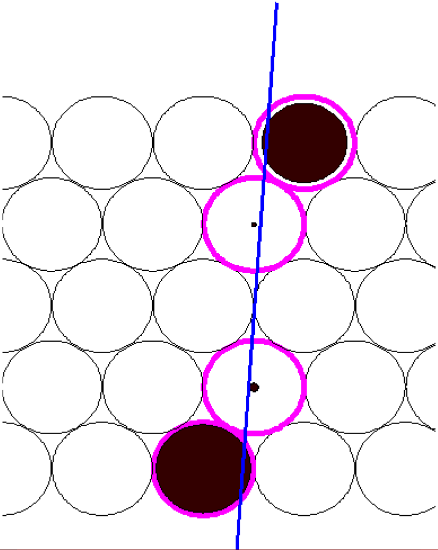
- Trained in Python/PyTorch: need to translate to C++
- Package of choice: *ONNXRunTime*
  - Highly optimized package to read ONNX files in C++
- Inference in C++ on CPU isn't the fastest: quantization/fusing layers!
- Able to get inference speed on data to 0.10 ± 0.01 ms/prediction
- Implementation:
  - If we have enough hits to infer: use ML predictions to fit (in local straw frame) to left/right of each hit (based on ML predictions)
  - Feed output of this to standard tracking pipeline

Performance on Simulation

- Wanted some comparisons of with and without ML
- Ran small simulation, ran tracking with and without ML interface
- Checked how often tracker got LR correct

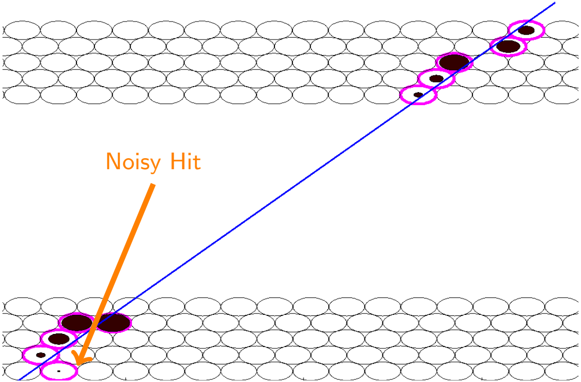
LR% Correct/Model	Without ML	With ML
Total	90.1%	92.6%
< 1mm	81.4%	82.2%
≥ 1mm	92.2%	95.0%
< 2.5mm	85.6%	87.7%
≥ 2.5mm	95.0%	97.6%
Vertical Straws	89.6%	93.3%
Horizontal Straws	90.9%	91.8%
LH Straws	91.5%	93.4%
LV Straws	88.1%	91.5%
RH Straws	90.1%	90.0%
RV Straws	91.0%	94.9%

STT Tracking: Back to the Failure



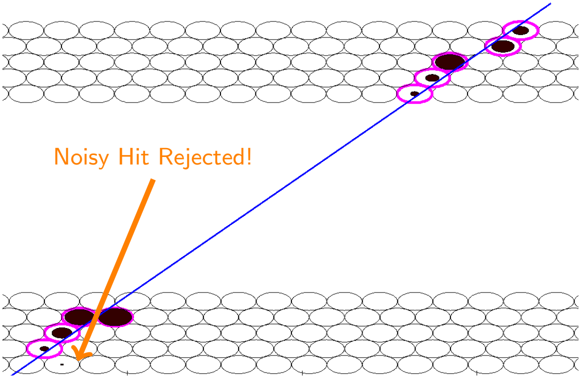
Noise Filtering

- Have an algorithm to group hits together - capable of some noise filtering and multi tracking
- Noise is too close to good hits - current algorithm fails
- True noise, multiple scattering, TTD inefficiency, etc.
- Want only best hits contributing to track to be given to ML



Better Noise Filtering

- Initial seed: OLS to straw centers
- Based on residual distribution from OLS fit: can reject hits as noise based on median average deviation (MAD)
- Takes χ^2_{red} of this track from $12.2 \rightarrow 0.8$!



- Machine learning is proving incredibly valuable in MUSE scattered particle tracking
- Enhancing the left-right ambiguity and noise filtering will only make our tracking stronger
- More improvements in both regards on the way!