AI Progress in Forward Tracking

Tongtong Cao
Jefferson Lab

CLAS Collaboration Meeting
March 10-13, 2026

Jefferson Lab
Thomas Jefferson National Accelerator Facility

clas
CEBAF Large Acceptance Spectrometer

About Forward Tracking

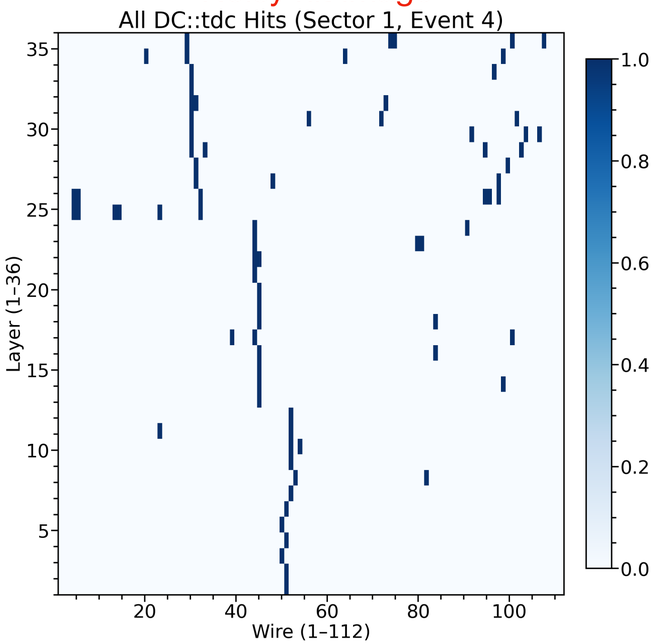
Pass2	Forward tracking	New
AI denoising	Hit denoising	Optimized AI denoising model (coatjava 13.5.2 / 2026-01)
Clustering algorithms	Clustering	Optimized clustering algorithms (coatjava 11.1.2 / 2025-02)
AI cluster-combo prediction	Track finding for construction of cluster combos	Optimized AI cluster-combo prediction (coatjava 13.5.2 / 2026-01)
Kalman Filter tracking	HB tracking with HB hit measurements	AI estimator for HB tracks (coatjava 13.5.3 / 2026-01)
Kalman Filter tracking	TB tracking with TB hit measurements	Optimized tracking by incorporating Deterministic Annealing Filter within Kalman Filter (coatjava 10.1.0 / 2024-04)

- Since pass2, forward tracking has been comprehensively upgraded across all stages of the reconstruction pipeline.
- New clustering and new TB tracking have been reported.
- Toady, new AI models for hit denoising, track finding and estimation of HB track state will be presented.

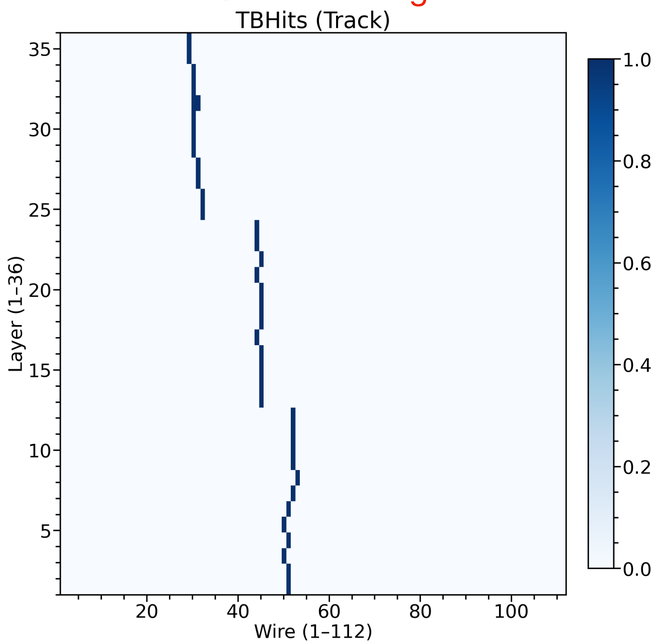
AI Denoising by Convolutional Auto-Encoder

- CLAS12 suffers from high DC hit occupancy, especially at R1.
- A convolutional-neural-network (CNN) based auto-encoder learns global spatial correlations of tracks and suppresses non-track-like patterns automatically.
- The auto-encoder takes a noisy hit image as input, compresses it into a latent representation, and reconstructs a cleaned version of the same image.
- The CNN-based auto-encoder is ideal because DC hits live on a 2D grid (layer × wire), tracks form correlated structures, and noise is mostly spatially uncorrelated.

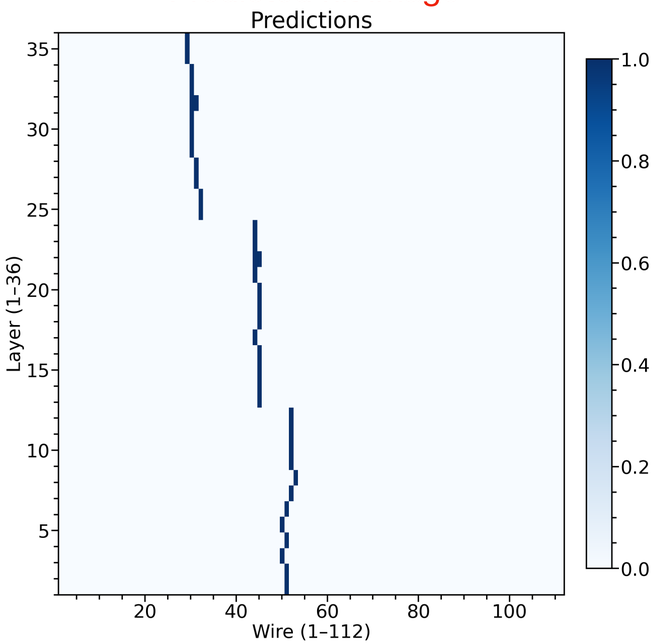
Noisy hit image



Clean hit image



Predicted hit image



New AI-denoising Model

- The original AI-denoising model was developed by Gagik Gavalian. To apply the model, a C++ package was developed, and it must be applied before reconstruction, independent of coatjava. The package has been applied by pass2.

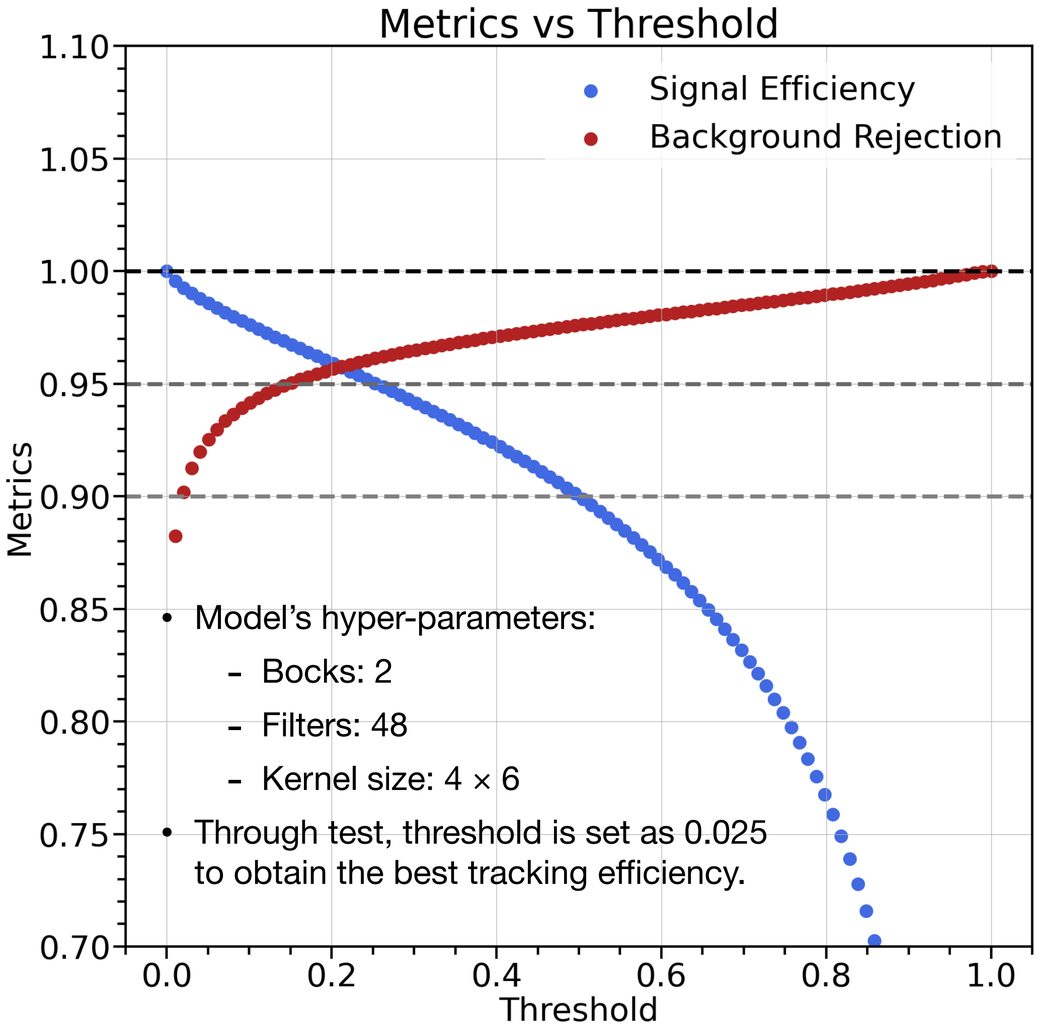
- To get full control of denoising model and let model directly integrated into coatjva, a new model was developed with the same AI technique. The new model development was led by Richard Tyson.

- An engine, called as DCDenoiseEngine, was developed to apply the new model into coatjava. In a yaml file, the engine for hit denoising is added before DC clustering:

```
- class: org.jlab.service.ai.DCDenoiseEngine
  name: DCDN
- class: org.jlab.service.dc.DCHBClustering
  name: DCCR
```

- With the new denoising, the old denoising should be cancelled in workflow commands: clas12-workflow ~~--denoise~~ --model decrec …

Trained by samples from RGA run 5342

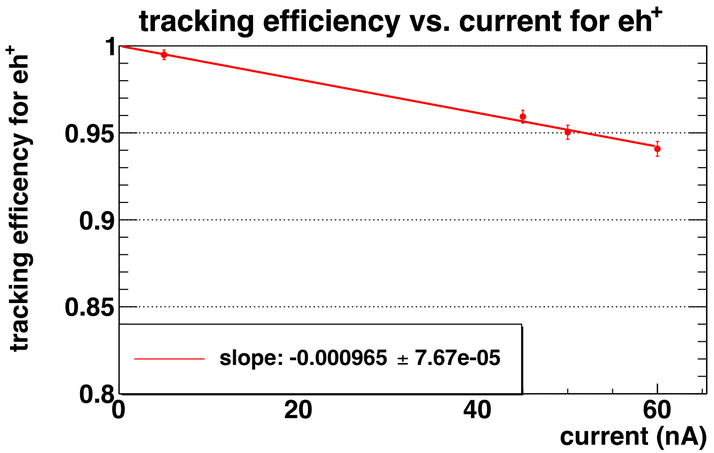


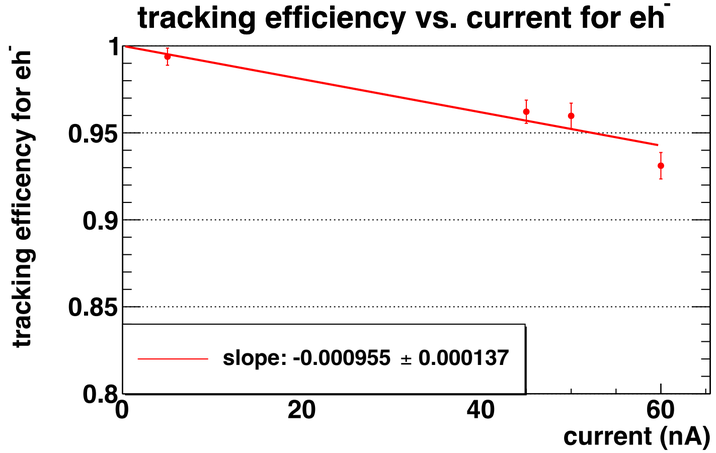
RGA Inbending

Tracking Efficiency vs. Luminosity

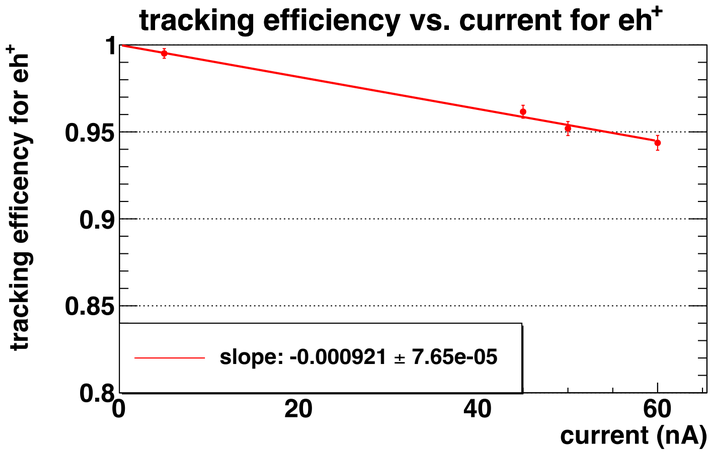
- Cuts for final-state particles: vz [-15, 5] cm, p > 0.5 GeV and |chi2pid| < 3.
- The number of eh^+/eh^- events is normalized to the number of events with an electron.

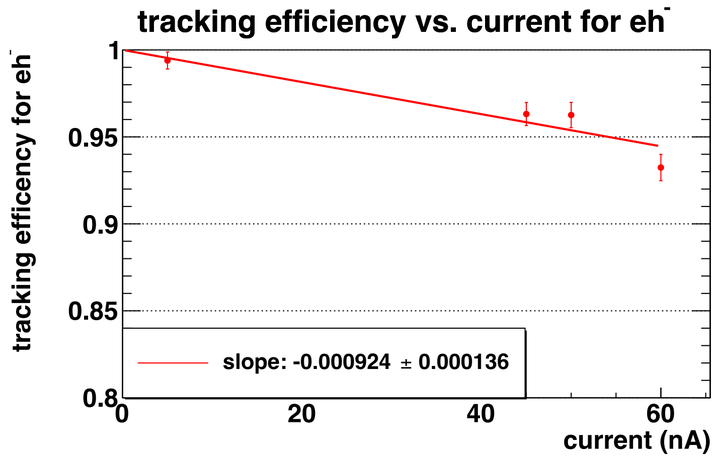
Old Denoising





New Denoising



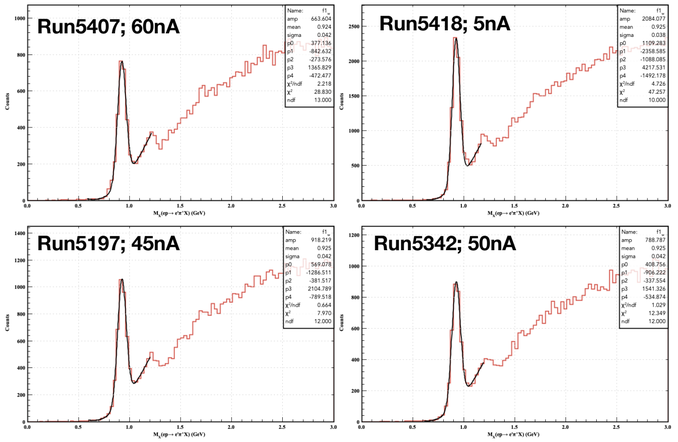


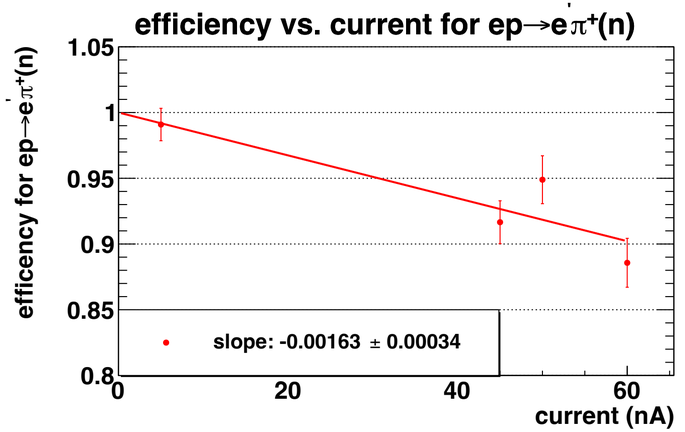
RGA Inbending

Efficiency for $ep \rightarrow e'\pi^+(n)$

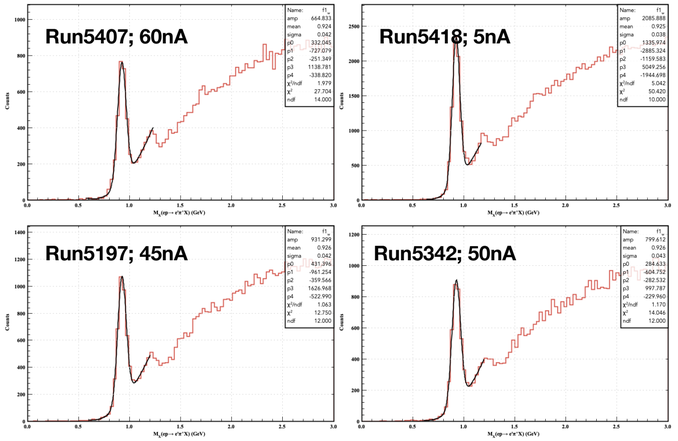
- Events of are selected with cuts for all final-state particles: vz [-15, 5] cm, p > 0.5 GeV and |chi2pid| < 3. Note: all π^+s are used to calculate missing mass if multiple exist.
- The missing mass distributions at each beam current are fitted with Gaussian+pol4 to extract number of events for the reaction.
- The number of exclusive events is then normalized to the number of inclusive events with an electron for efficiency study.

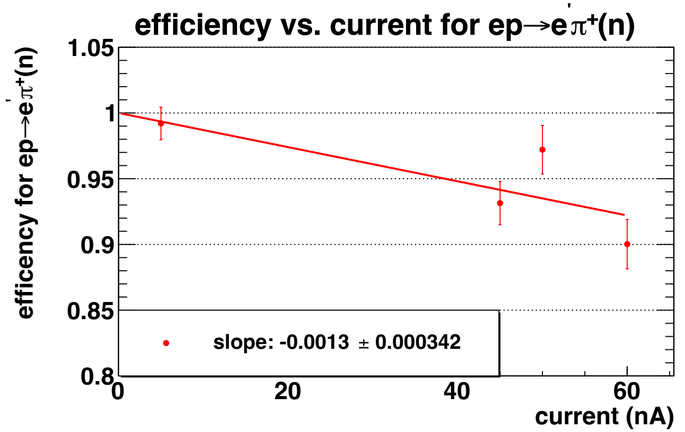
Old Denoising





New Denoising





AI Models for FD Track Finding

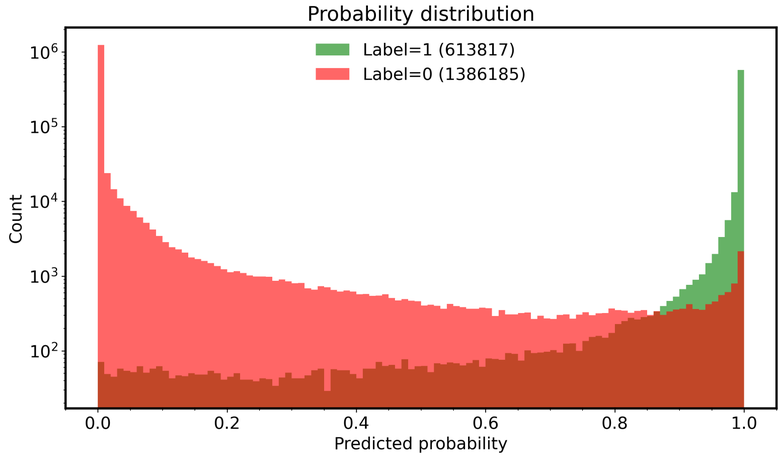
- AI models are employed to predict 6-cluster and 5-cluster combinations as seeds for tracking.
- Old Models, which were developed by Gagik Gavalian and were used in pass2
  - A Multi-Layer Perceptron (MLP) classifier with features of average wires of clusters is to predict if a 6-cluster combo is a track candidate.
  - A auto-encoder estimator is trained to estimate average wire of missing cluster for 5-cluster combos. Then, a 5-cluster combo with an estimated average wire of missing cluster was classified using the same MLP model trained for 6-cluster combos.
- New models,
  - A new MLP classifier is trained for 6-cluster combos using both features of average wires and slopes
  - A separate MLP classifier is trained for 5-cluster combos using features of average wires, slopes and missing cluster's superlayer number

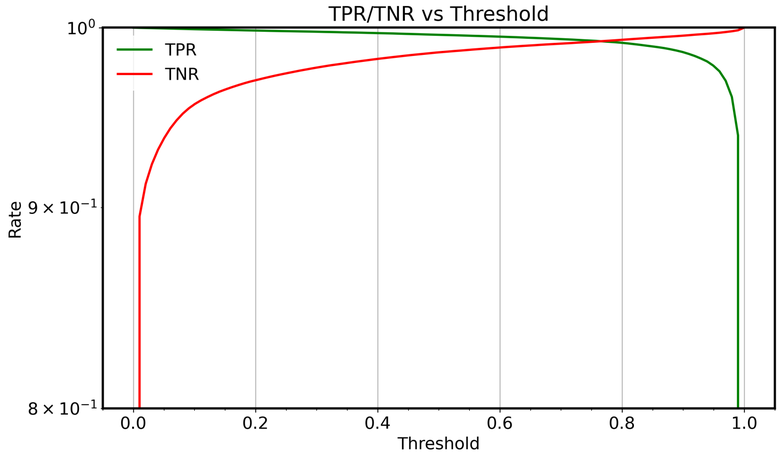
New MLP Models

Models are trained by samples from RGA run 5342

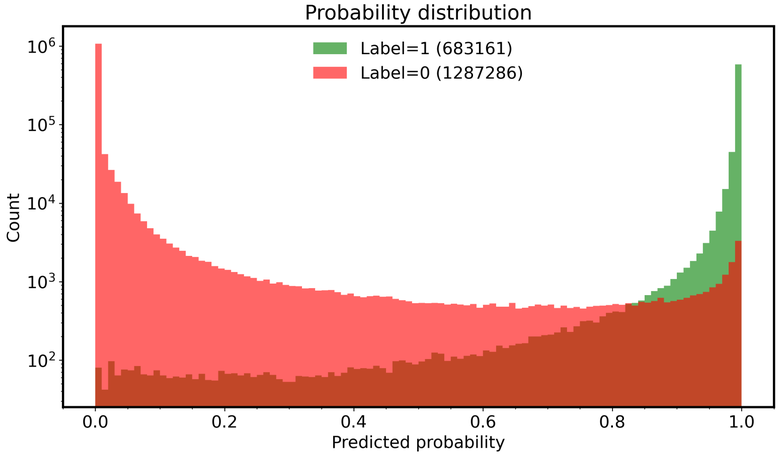
- TPR: True Positive Rate
- TNR: True Negative Rate

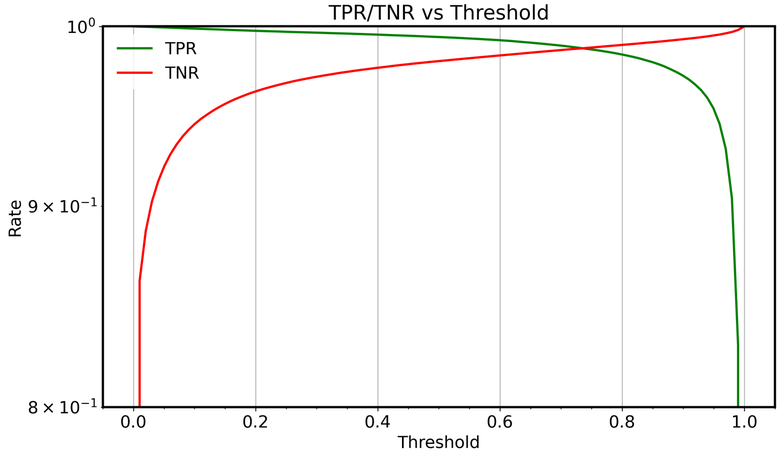
- MLP Model for 6-cluster combos
  - # of layers = 4
  - Dim. of hidden layers = 64
- As test, threshold is set as 0.95 for the best balance of 6-cluster and 5-cluster tracks





- MLP Model for 5-cluster combos
  - # of layers = 3
  - Dim. of hidden layers = 64
- Threshold is set as 0.05 for keeping almost all positive cases and eliminating most negative cases.





Engine for New Models

- An engine, called as DCClsComboEngine, was developed in coatjava to apply the new models.

- In yaml files, to apply the new engine with application of the new models, the old engine with application the old models should be replaced.

Old engine

```
- class: org.jlab.service.mltn.MLTDEngine
  name: MLTD
```

New engine

```
- class: org.jlab.service.ai.DCClsComboEngine
  name: DCCC
```

Observations through Event-by-Event Comparison between Two Models

- Compared with the old models, the new models retain significantly more cluster combos while missing far fewer combos. Many combos that were missed by the old models are successfully predicted by the new models, especially for 5-cluster combos.

- The new models demonstrate stronger capability in handling complicated scenarios, particularly events with multiple tracks in a sector.

- In validation tests, the number of valid 6-cluster track shows a slight increase, while the number of valid 5-cluster track is increased over 30%.

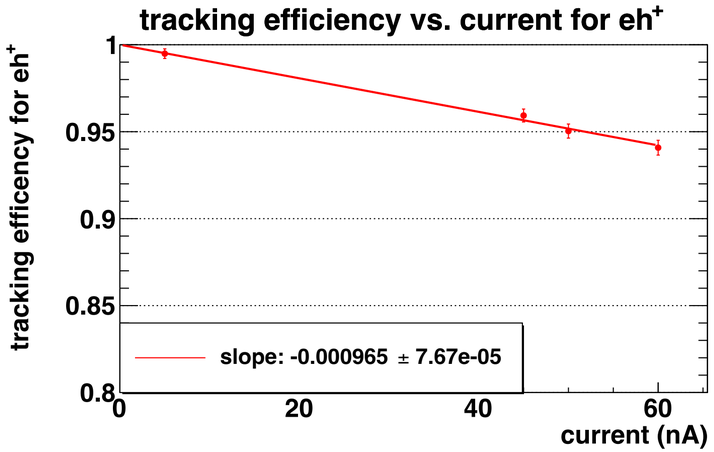
- The primary source of this improvement is the inclusion of cluster slope information as input features in the new models, which provides additional geometric constraints and enhances their discriminating power.

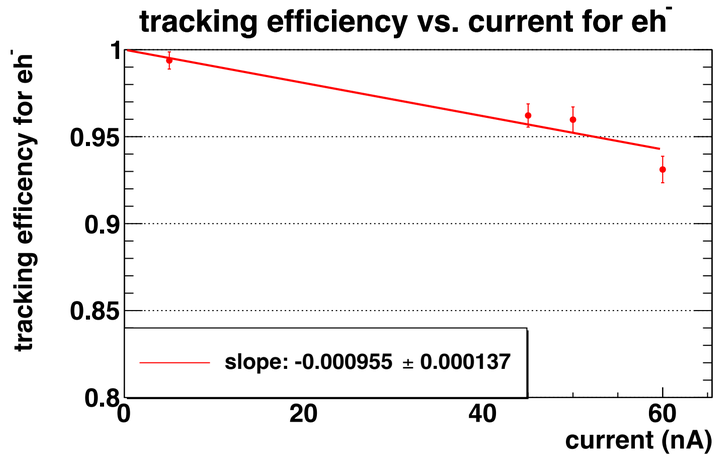
RGA Inbending

Tracking Efficiency vs. Luminosity

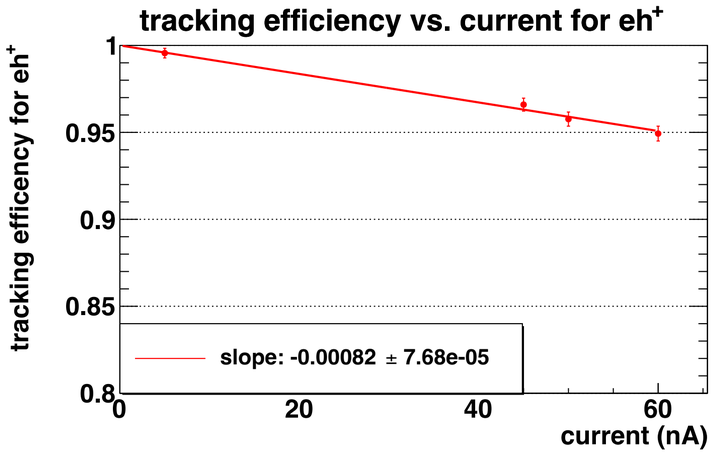
- Cuts for final-state particles: vz [-15, 5] cm, p > 0.5 GeV and |chi2pid| < 3.
- The number of eh^+/eh^- events is normalized to the number of events with an electron.

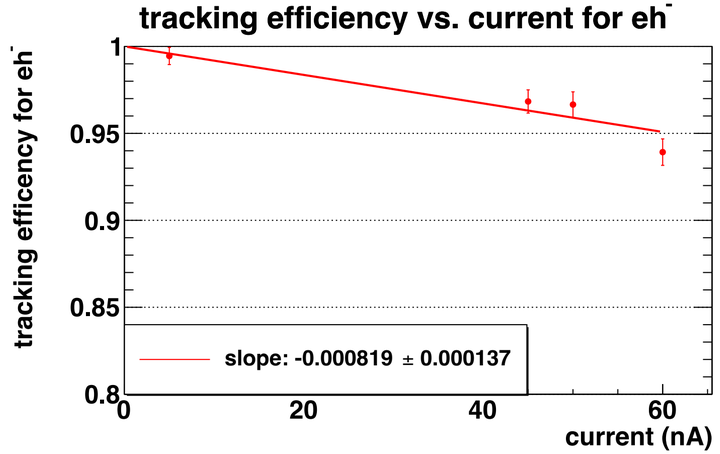
Old Models





New Models



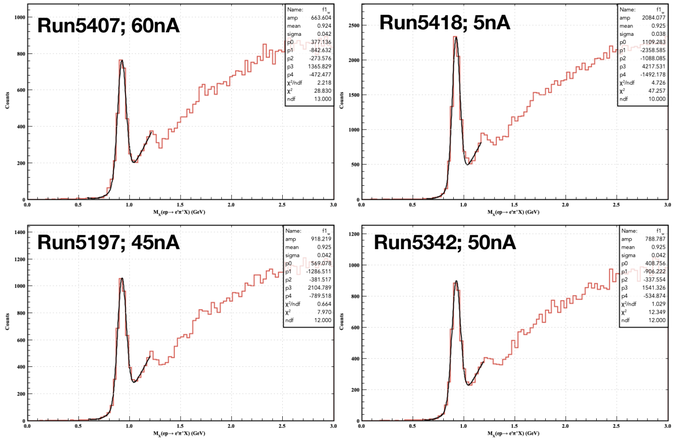


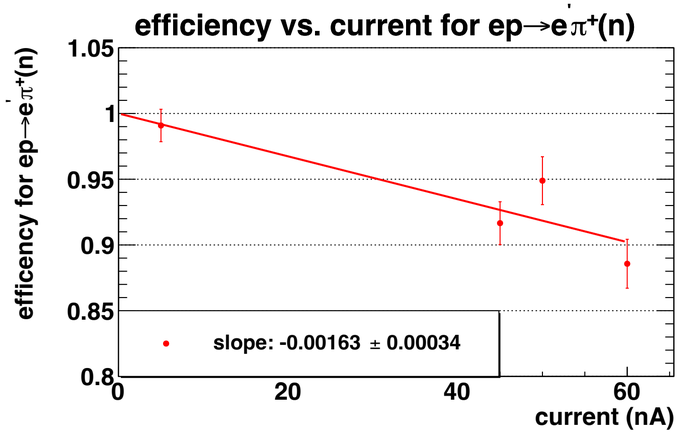
RGA Inbending

Efficiency for $ep \rightarrow e'\pi^+(n)$

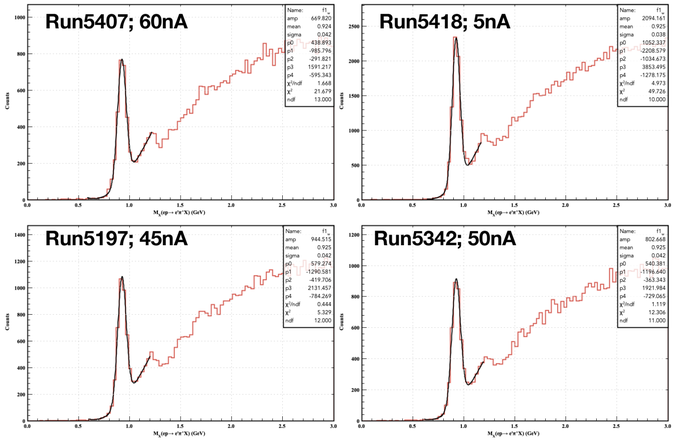
- Events of are selected with cuts for all final-state particles: vz [-15, 5] cm, p > 0.5 GeV and |chi2pid| < 3. Note: all π^+s are used to calculate missing mass if multiple exist.
- The missing mass distributions at each beam current are fitted with Gaussian+pol4 to extract number of events for the reaction.
- The number of exclusive events is then normalized to the number of inclusive events with an electron for efficiency study.

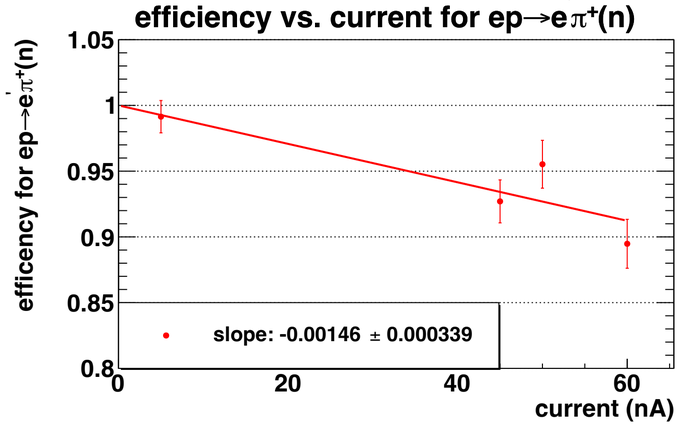
Old model





New model

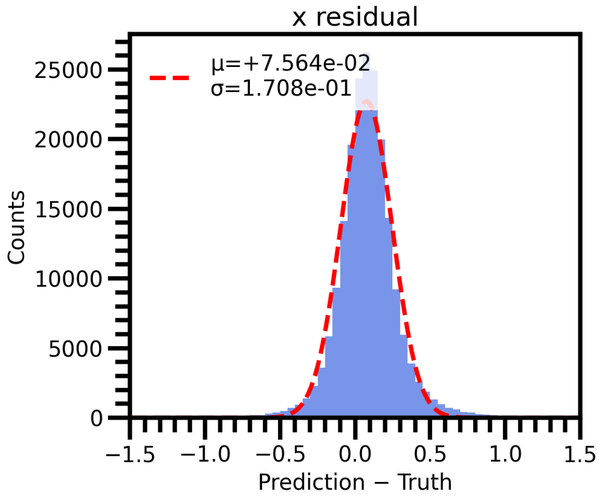


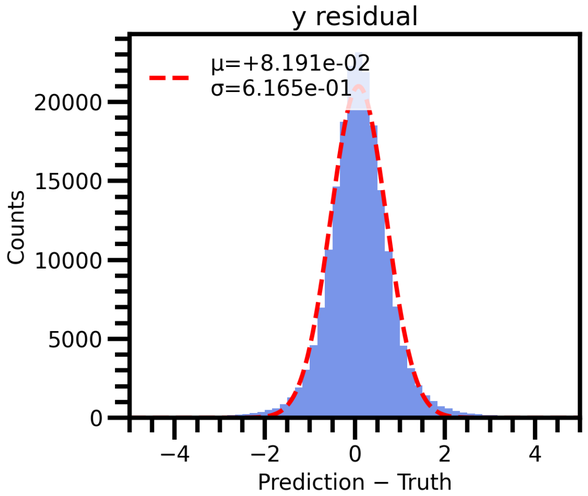


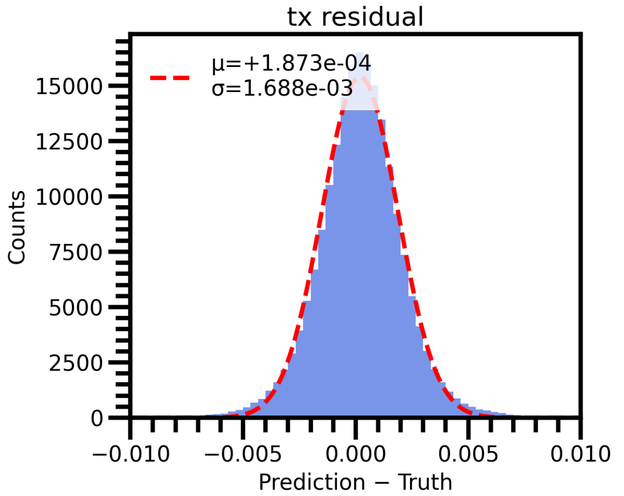
AI Model for HB Tracking

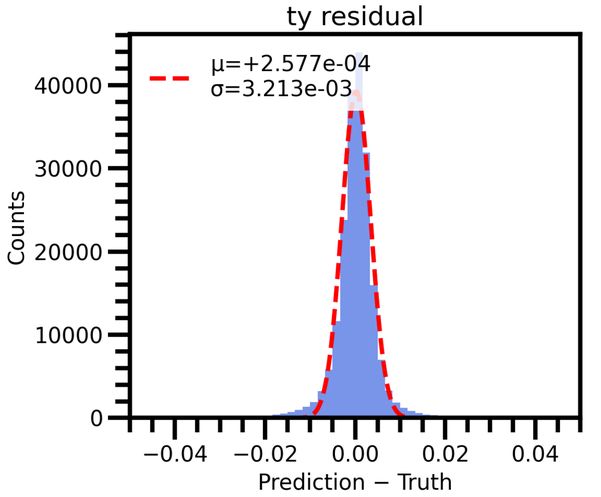
- HB tracking by Kalman Filter (KF) faces challenges:
  - Approximate HB measurements: HB measurements are constrained by cell size, and do not precisely represent hit positions.
  - Left–right ambiguity: DC hits suffer from left–right ambiguity relative to wires.
  - Rough initial state estimation: Initial track parameters from the pattern recognition rely on an over-simplified quadratic fit to three crosses.
- The AI model based on the transformer technique treats track reconstruction as a sequence-to-state regression problem.
- The transformer leverages the self-attention mechanism to capture global correlations among hits from different detector layers, and it can directly estimate HB track state by hit pattern:
  $hit\ pattern \rightarrow track\ state\ (x, y, t_x, t_y, Q)$

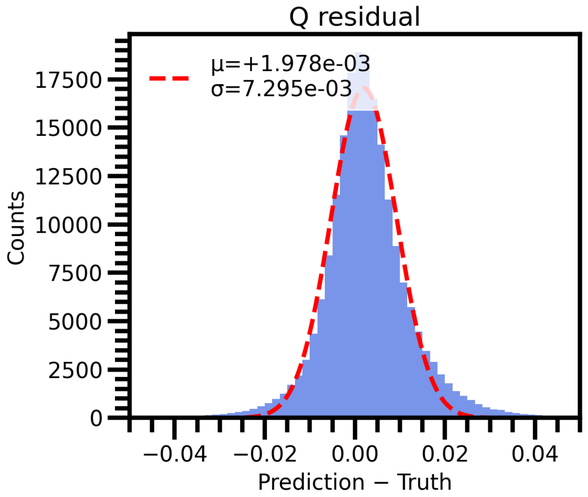
Validation for Model Trained by Inbending Run 5342





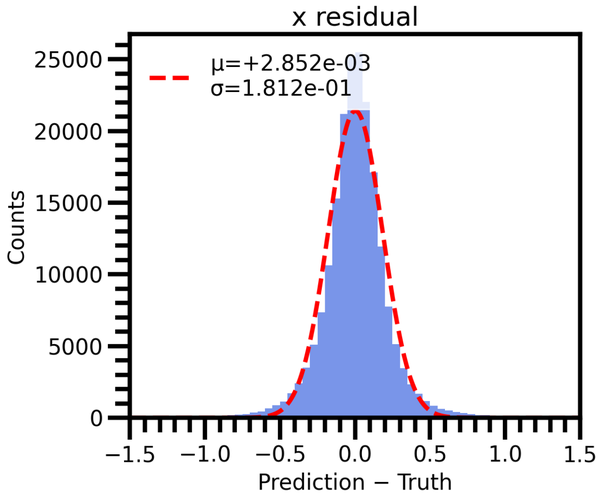


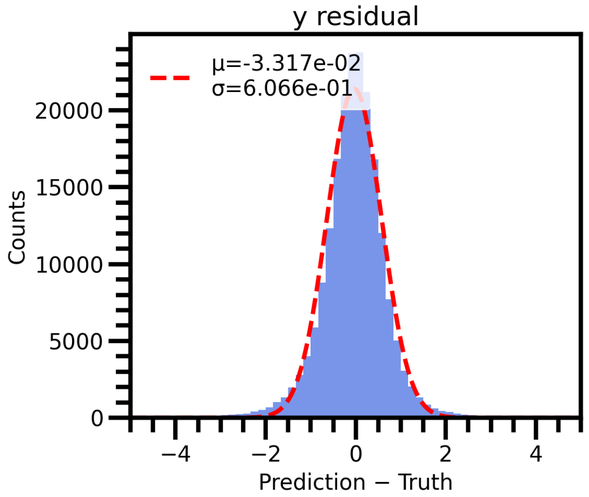


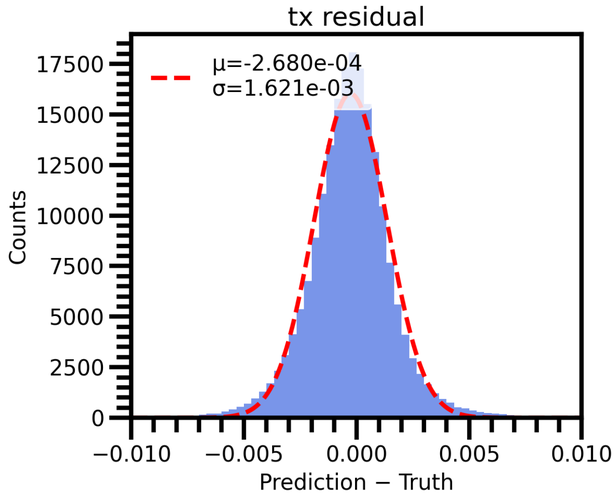


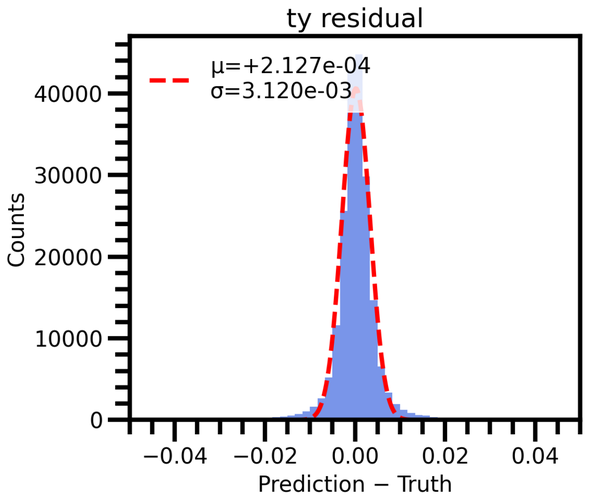
Truth is TB track state

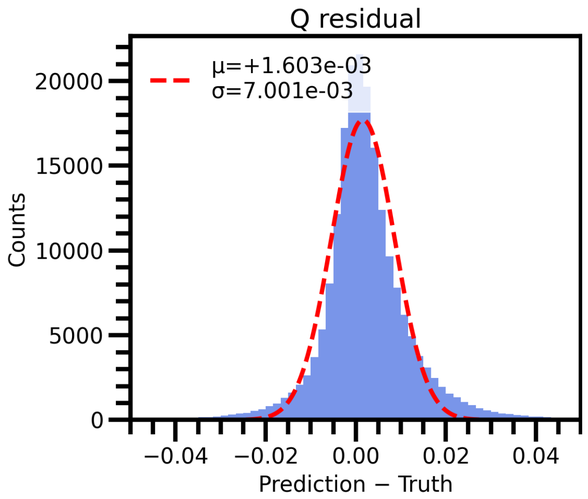
Validation for Model Trained by Outbending Run 5543











Truth is TB track state

New Engine for HB Tracking by AI in coatjava

- A new engine with application of the AI model for estimation of HB track state, called as DCHBTrackingAI, is developed.

- With the new engine, there are three options for HB tracking:

    1. DCHBPostClusterConv: All possible cluster combos are tried, and tracking is processed with KF.

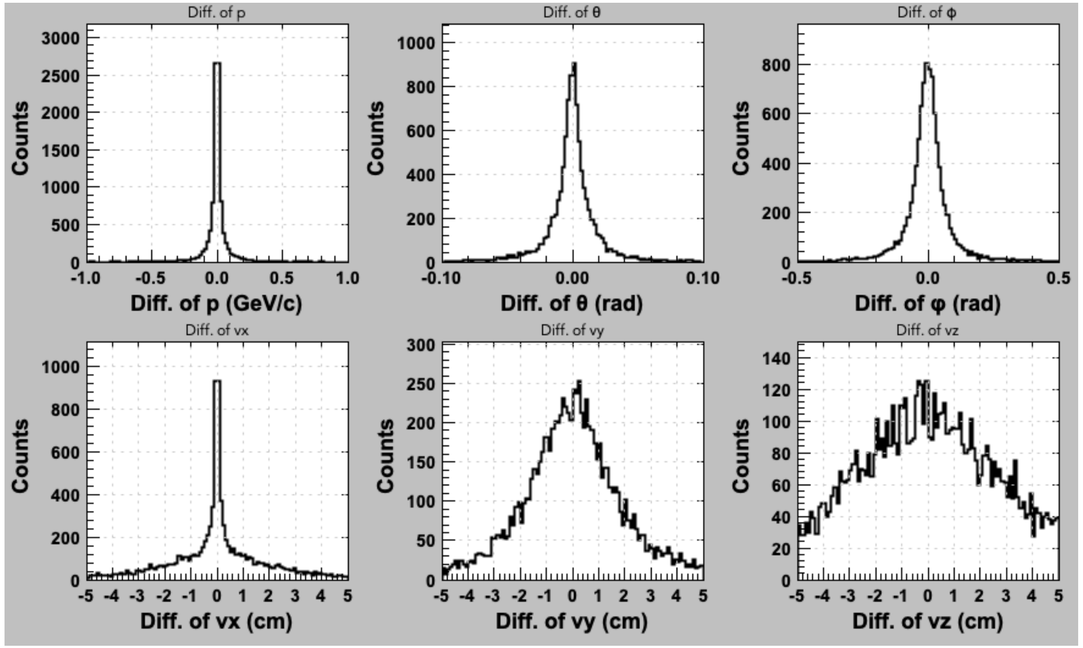
    2. DCHBPostClusterAI: Cluster combos are predicted and pre-selected by AI, and tracking is processed with KF.

    3. DCHBTrackingAI: Cluster combos are predicted and pre-selected by AI, and track states are estimated by AI.

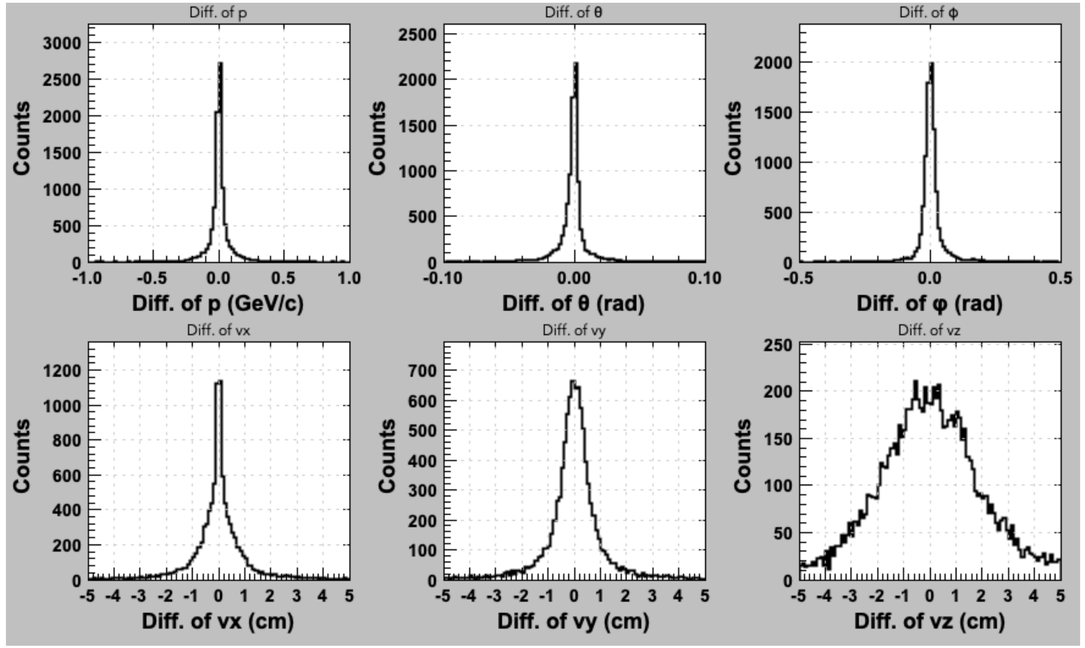
- Next, track states from options 2 & 3 are compared. One is extracted by KF, and the other is estimated by AI.

Check HB Track Resolution with TB Tracks as Reference

HB tracking by KF



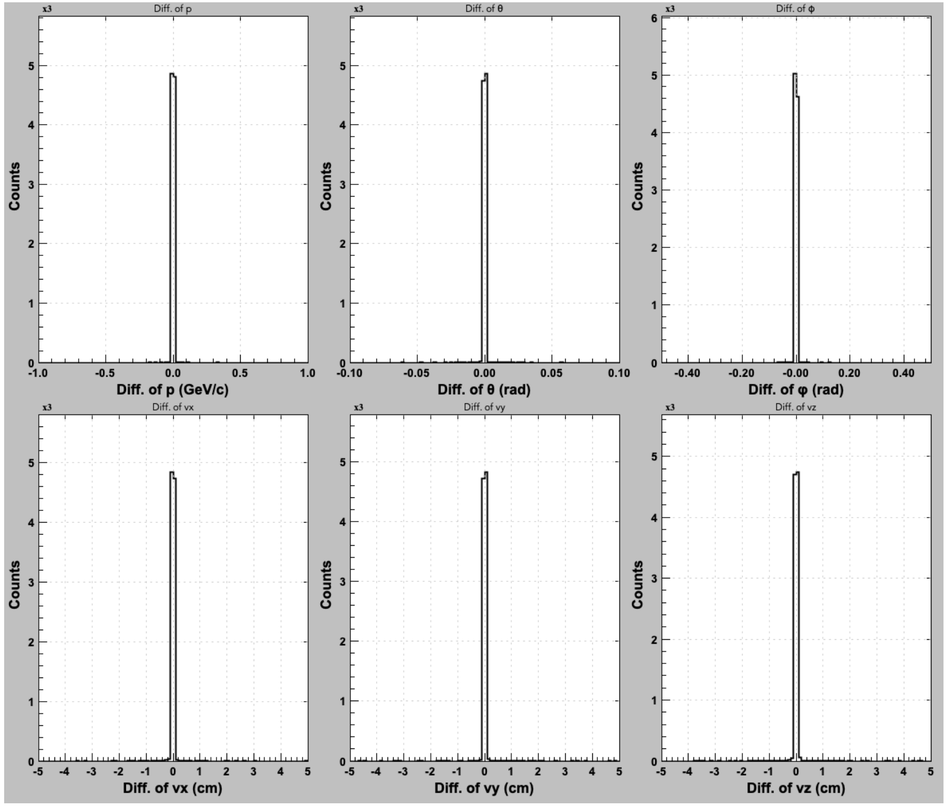
HB tracking by AI estimator



Difference between HB and TB track state at vertex

- Resolution for tracks by AI estimator is much better than HB tracking by KF.
- It will change quality for the following HB reconstruction based on HB tracks, and further affect TB reconstruction.

Comparison of TB Tracks between Different HB Tracks as Initial State of TB Tracking



- TB KF results remain consistent, although different HB track states are used for initialization.
- Therefore, impact on TB tracking resolution is negligible when the hit resolution definition remains unchanged.
- However, changes in HB track states affect TB hit resolution. Consequently, TB tracking results are influenced when the TB hit resolution is updated.

Discussion for AI and KF Tracking

- KF processes hits sequentially and locally. Since large uncertainty of HB measurements, KF fits trajectories through thick tubes.

- Instead of propagating a track step by step, AI captures global correlations among hits, and learns a direct mapping from hit pattern to track state. In doing so, the model implicitly absorbs effects, such as left–right ambiguity, magnetic field inhomogeneity, and cell geometry effects.

- As a result, the HB tracking resolution obtained with AI is better than that achieved with KF.

- However, the AI model is fundamentally a pattern-based estimator. It does not explicitly model track propagation in the magnetic field, nor does it account for multiple scattering or energy loss. When accurate TB measurements are available, KF tracking outperforms the AI approach.

- Therefore, the AI-predicted HB track state is used as the initial state for TB KF tracking, allowing the KF to refine the trajectory and obtain the final high-precision track parameters.

Observations through Event-by-Event Comparison between Two Models

- Comparing to HB tracking by KF, HB tracking by AI estimator preserves more tracks, while almost not missing tracks.

- HB tracking by KF losses tracks at two levels:

  - HB-level: A primary cause is KF divergence due to poor initialization.

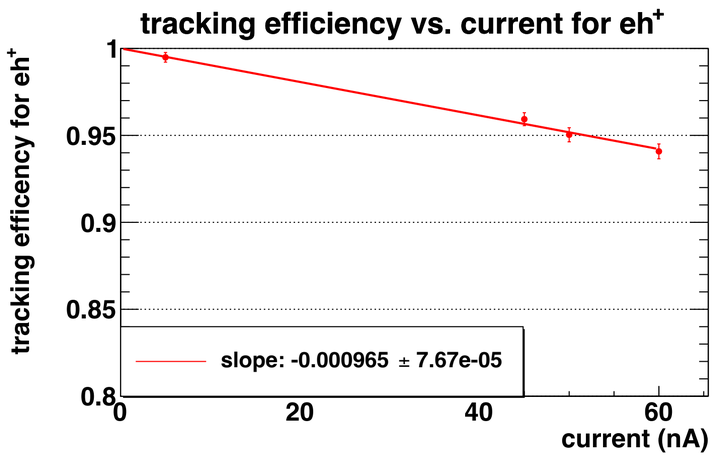
  - TB-level: Biased HB track states lead to incorrect TB hit association and KF failure.

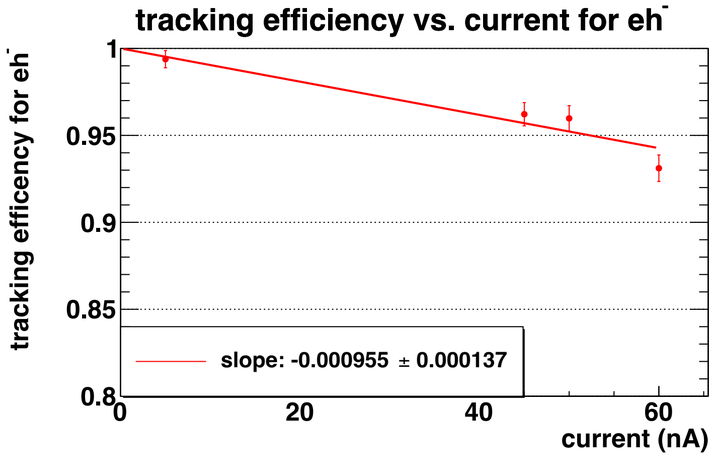
RGA Inbending

Tracking Efficiency

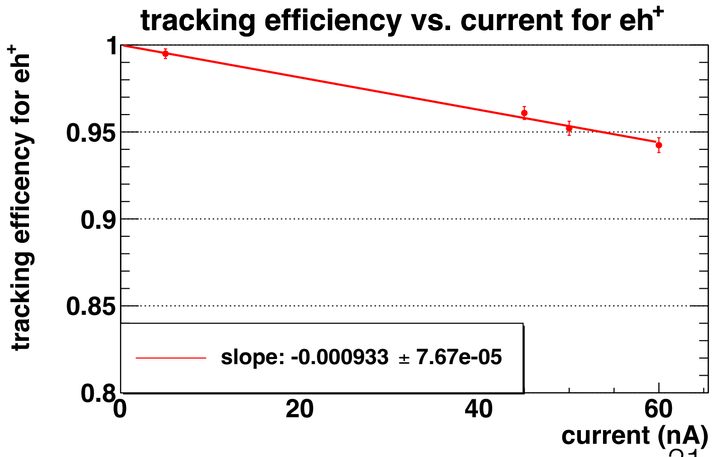
- Cuts for final-state particles: vz [-15, 5] cm, p > 0.5 GeV and |chi2pid| < 3.
- The number of eh^+/eh^- events is normalized to the number of events with an electron.

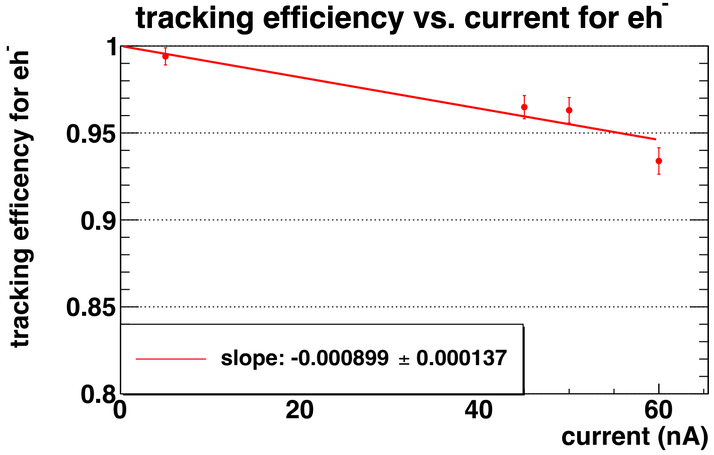
HB tracking by KF





HB tracking by AI estimator



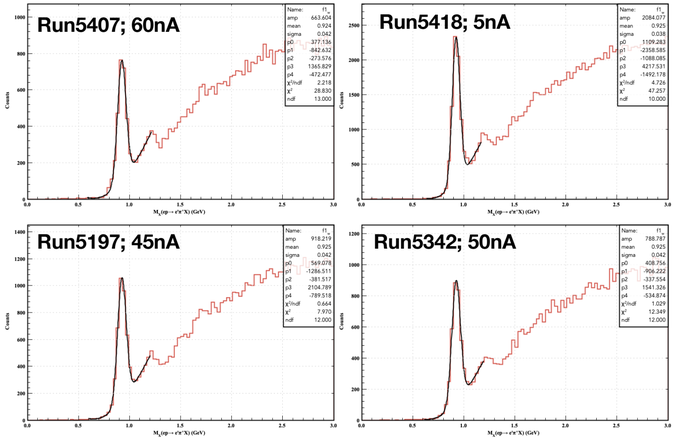


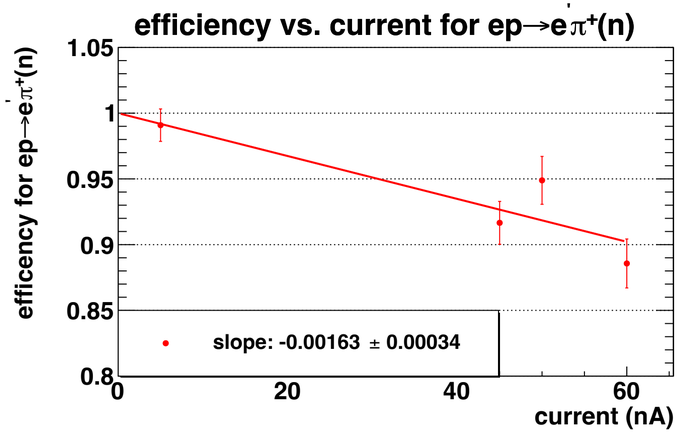
RGA Inbending

Efficiency for $ep \rightarrow e'\pi^+(n)$

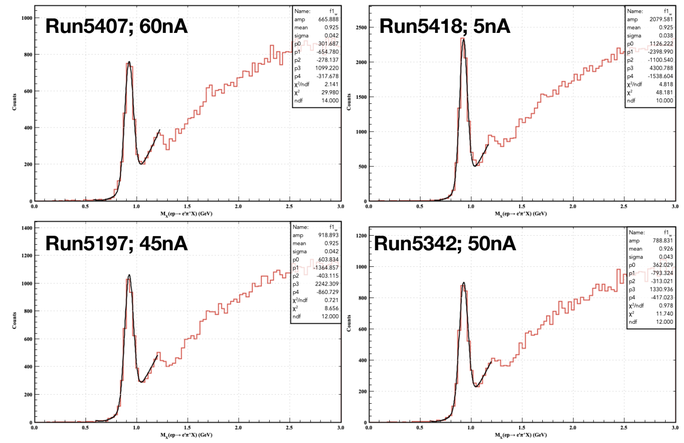
- Events of are selected with cuts for all final-state particles: vz [-15, 5] cm, p > 0.5 GeV and |chi2pid| < 3. Note: all π^+s are used to calculate missing mass if multiple exist.
- The missing mass distributions at each beam current are fitted with Gaussian+pol4 to extract number of events for the reaction.
- The number of exclusive events is then normalized to the number of inclusive events with an electron for efficiency study.

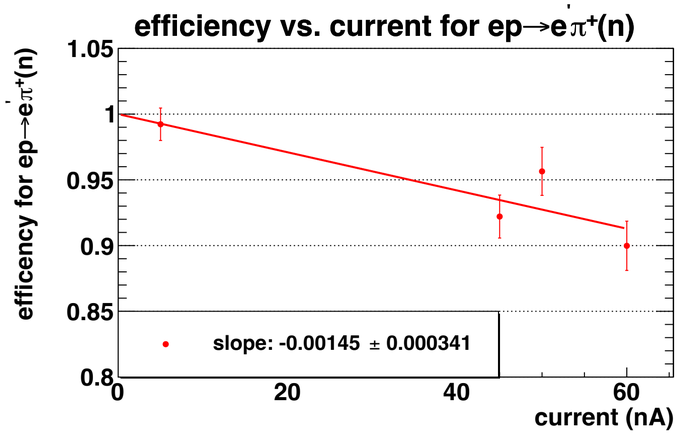
HB tracking by KF





HB tracking by AI estimator





Data cooking with All New AI Models

- In yaml files for DST cooking,
  - DCDenoiseEngine is added to apply new denoising with replacement of old denoising.
  - DCClsComboEngine replaces MLTDEngine for the cluster-combo prediction.
  - DCHBTrackingAI replaces DCHBPostClusterAI for HB tracking.
- For future updates and optimization of models, model version and threshold for the engines are optional in configuration of yaml files.
- Next, new data with application of all new AI models will be compared to data cooked by old versions of reconstruction.

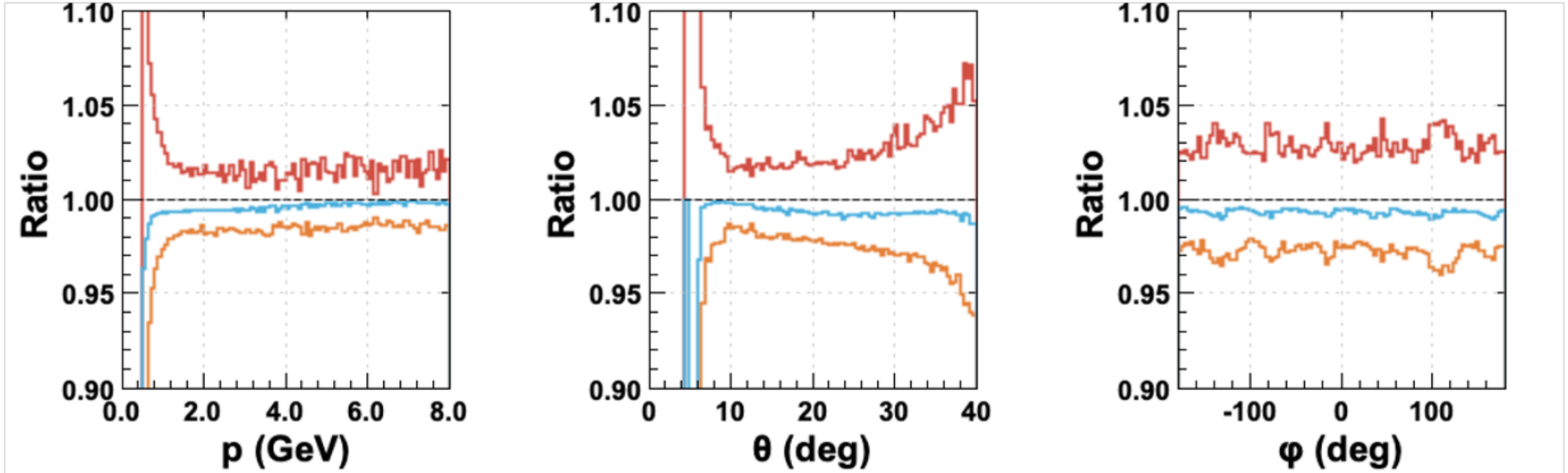
```
services:
  - class: org.jlab.clas.swimtools.MagFieldsEngine
    name: MAGFIELDS
  - class: org.jlab.service.swaps.SwapEngine
    name: SWAPS
  - class: org.jlab.service.raster.RasterEngine
    name: RASTER
  - class: org.jlab.service.ai.DCDenoiseEngine
    name: DCDN
  - class: org.jlab.service.dc.DCHBClustering
    name: DCCR
  - class: org.jlab.service.ai.DCClsComboEngine
    name: DCCC
  - class: org.jlab.service.dc.DCHBTrackingAI
    name: DCHTAI
  - class: org.jlab.service.ftof.FTOFHBEngine
    name: FTOFHB
  - class: org.jlab.service.eb.EBHBEngine
    name: EBHB
  - class: org.jlab.service.dc.DCTBEngine
    name: DCTB
  - class: org.jlab.service.ftof.FTOFTBEngine
    name: FTOFTB
  - class: org.jlab.service.eb.EBTBEngine
    name: EBTB
```

RGA Run 5407; 55nA

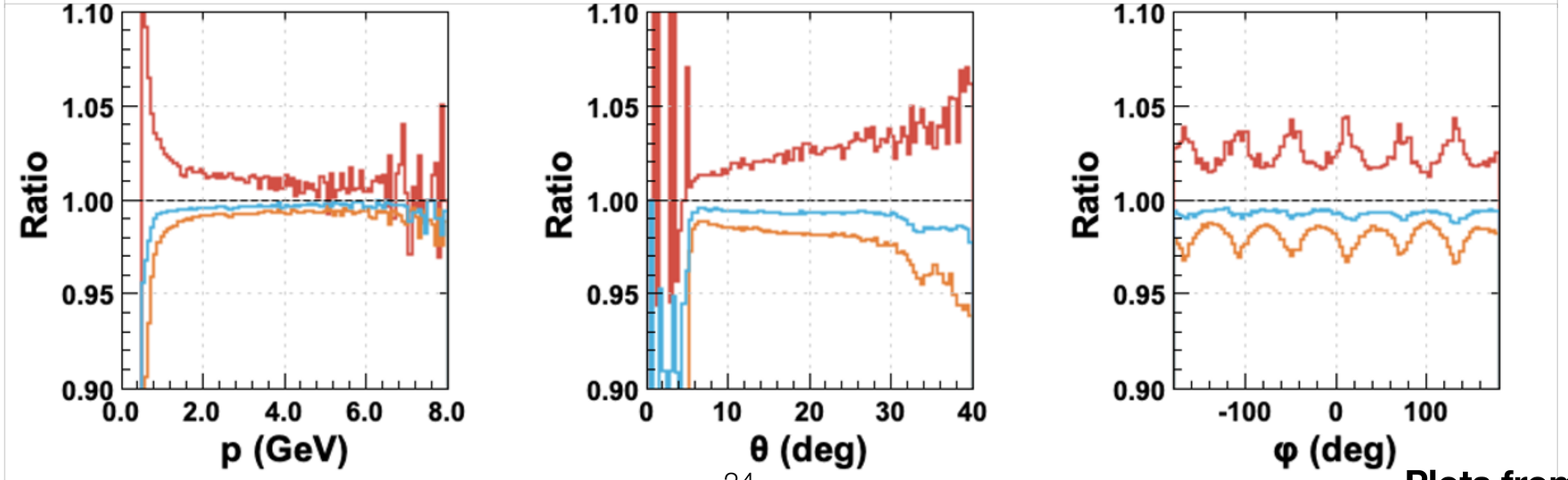
Comparison between Old and New AIs

- **Gain** = ratio of the number of tracks with the new AIs over the old AIs
- **oEfficiency** = fraction of the new-AI tracks found with the old AIs
- **nEfficiency** = fraction of the old-AI tracks found with the new AIs

Negative



Positive



Plots from Raffaella De Vita

RGA Run 5407; 55nA

Comparison between Old and New AIs

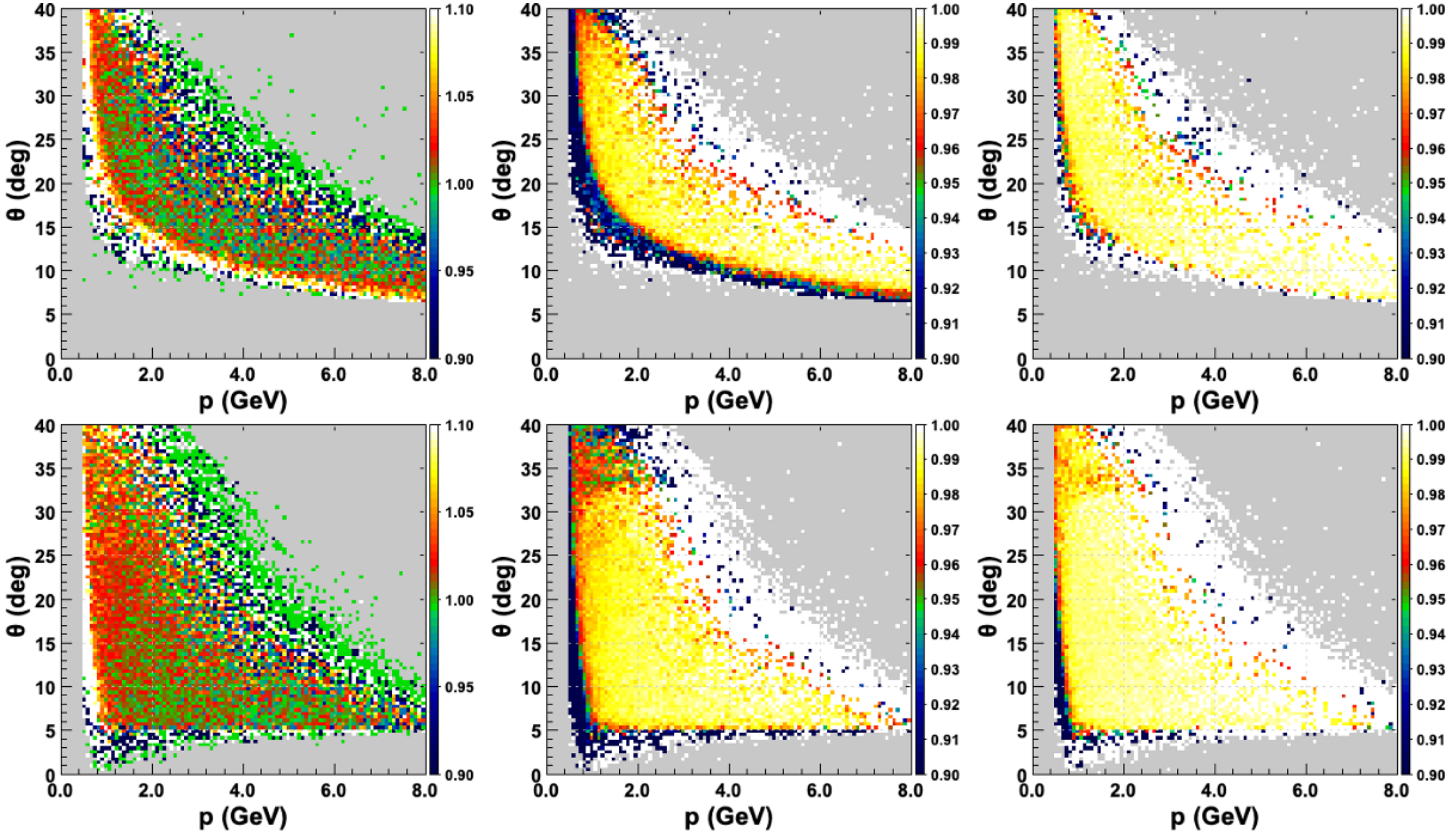
Gain = ratio of the number of tracks with the new AIs over the old AIs

oEfficiency = fraction of the new-AI tracks found with the old AIs

nEfficiency = fraction of the old-AI tracks found with the new AIs

Negative

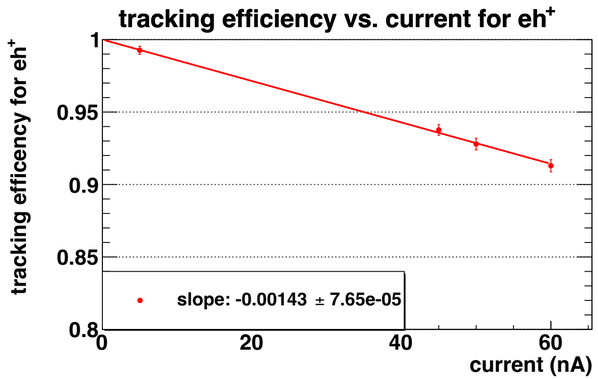
Positive

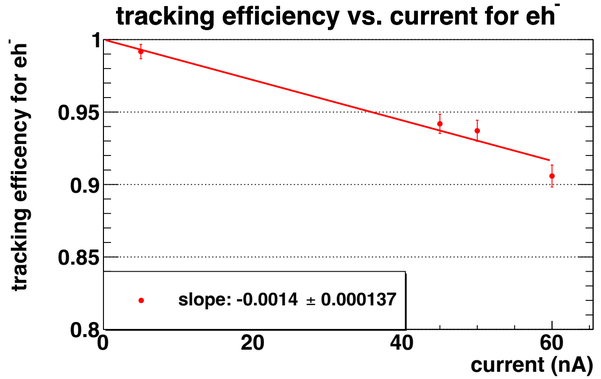


Plots from Raffaella De Vita

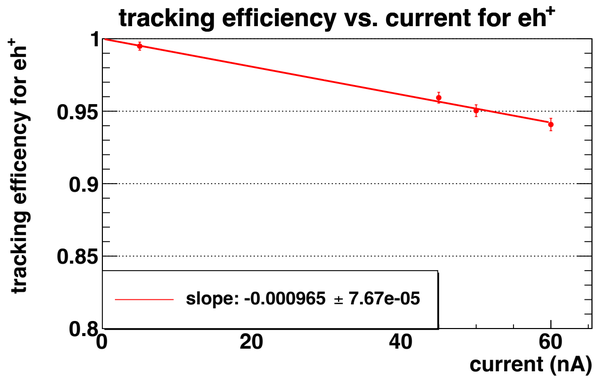
Tracking Efficiency for RGA Inbending

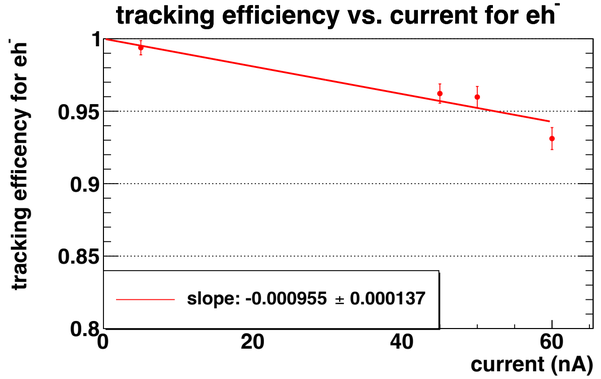
Pass2



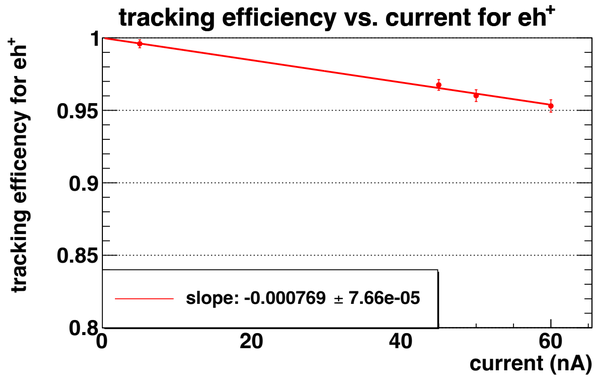


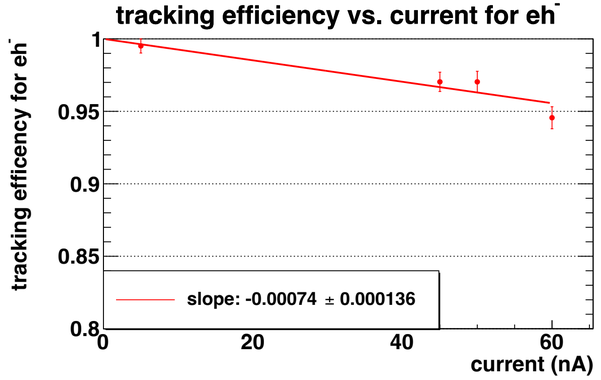
+ new clustering and new tracking





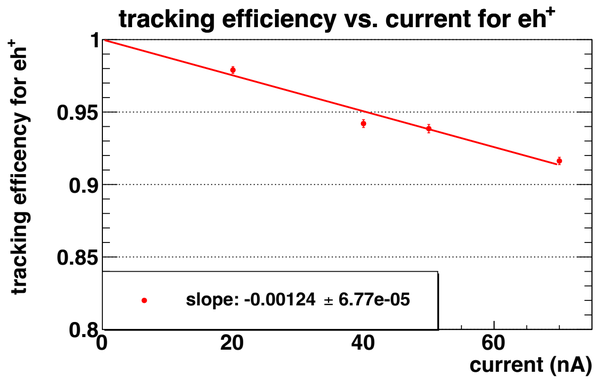
++ new AIs

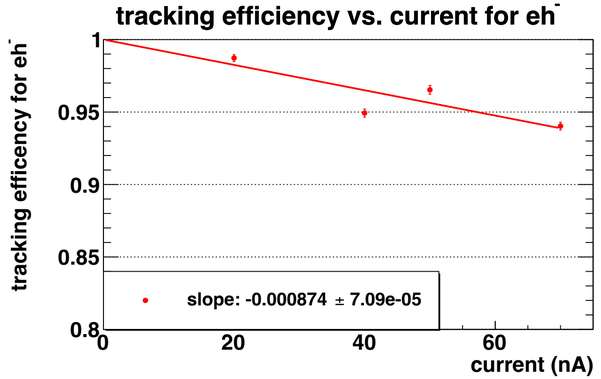




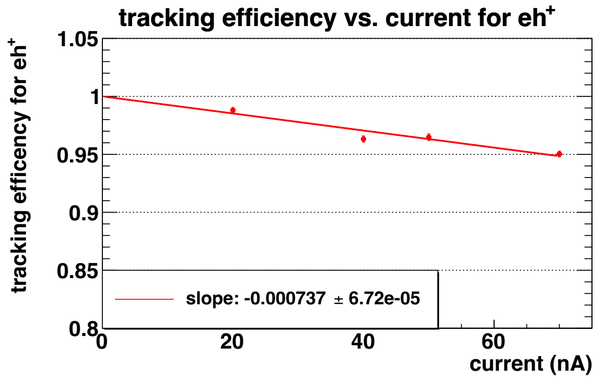
Tracking Efficiency for RGA Outbending

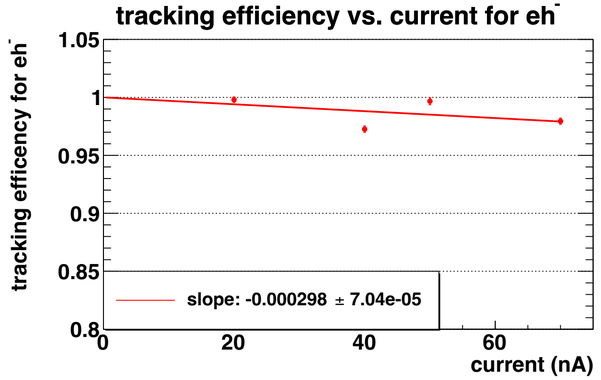
Pass2



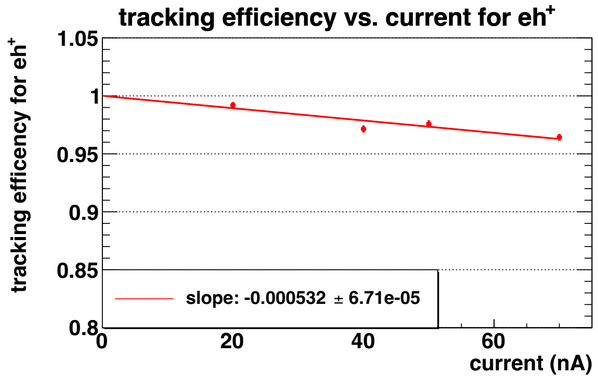


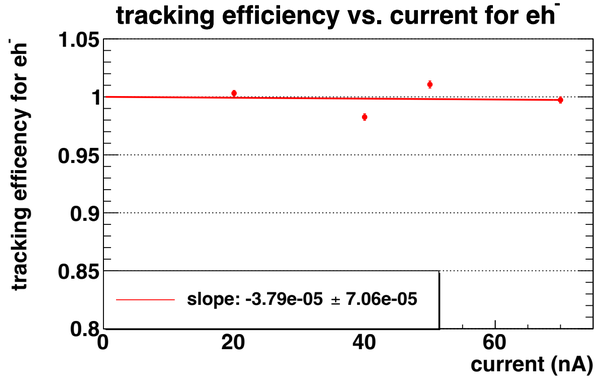
+ new clustering and new tracking





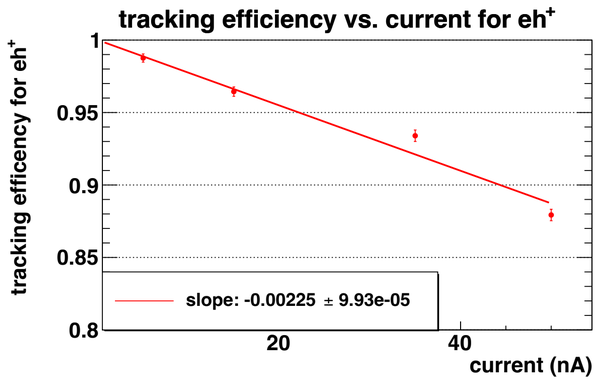
++ new AIs

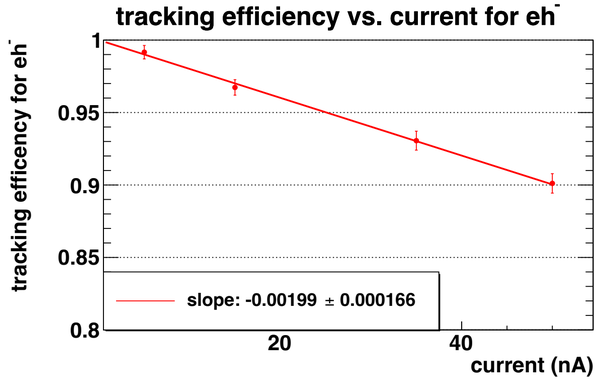




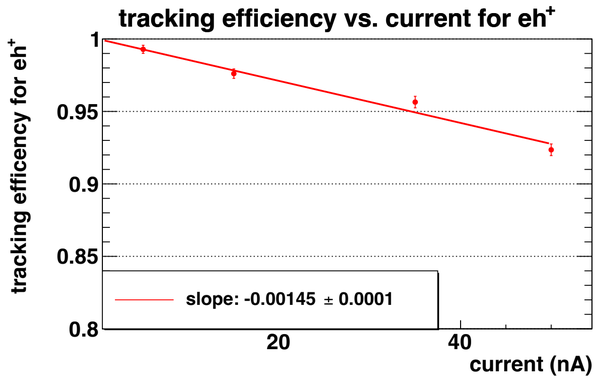
Tracking Efficiency for RGB Inbending

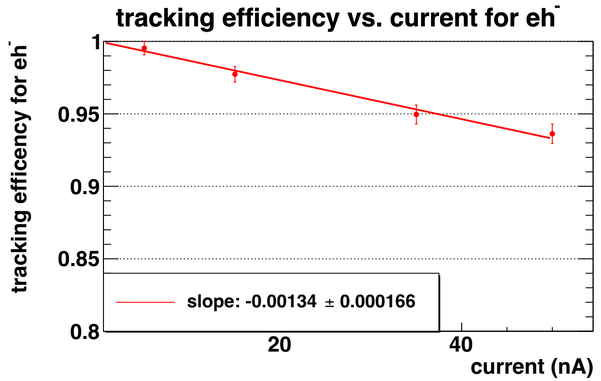
Pass2



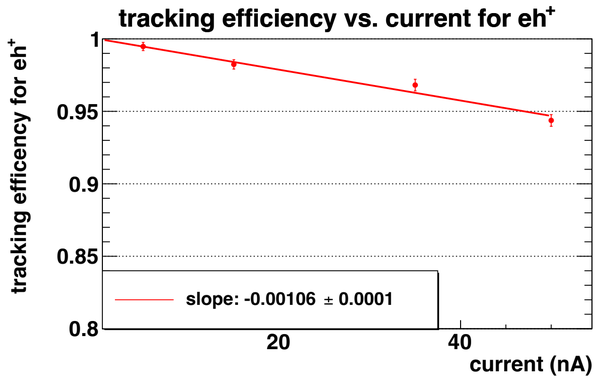


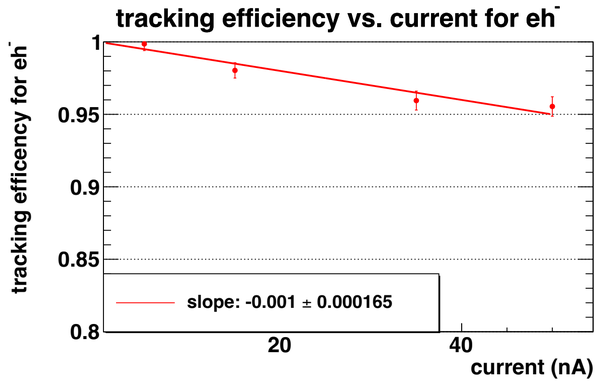
+ new clustering and new tracking





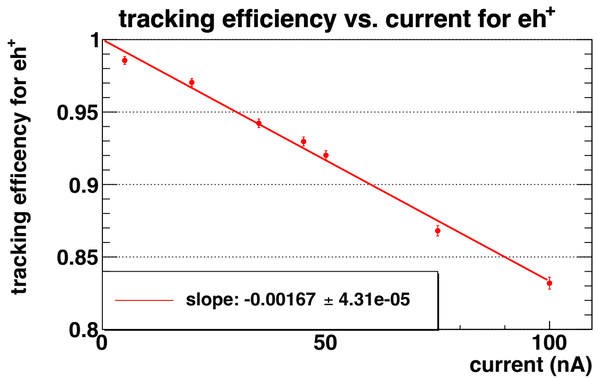
++ new AIs

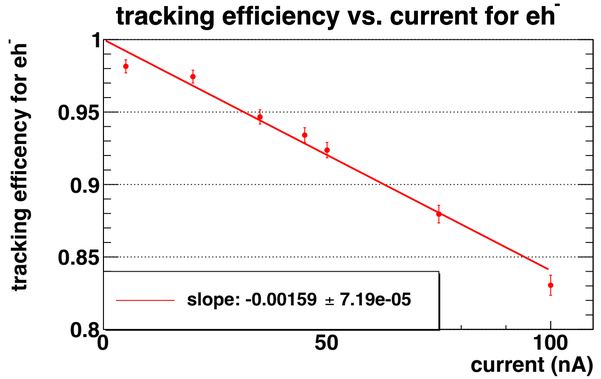




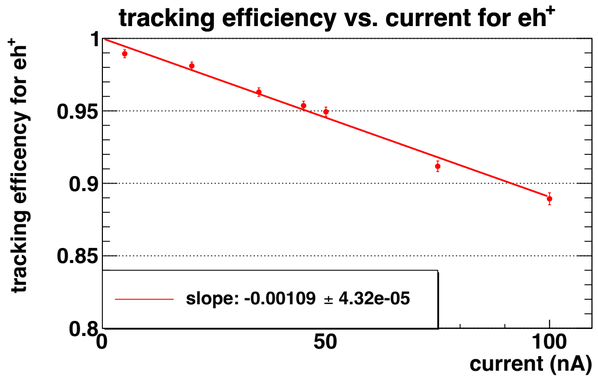
Tracking Efficiency for RGD Inbending

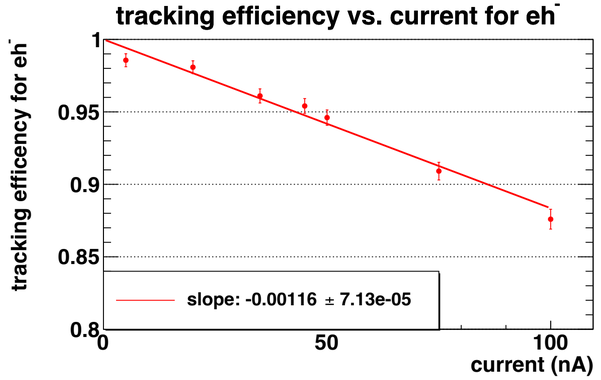
Pass2



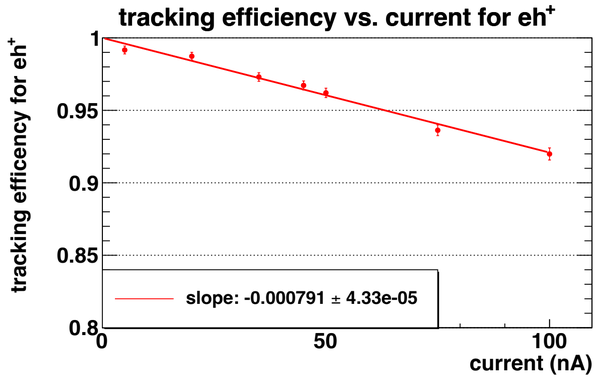


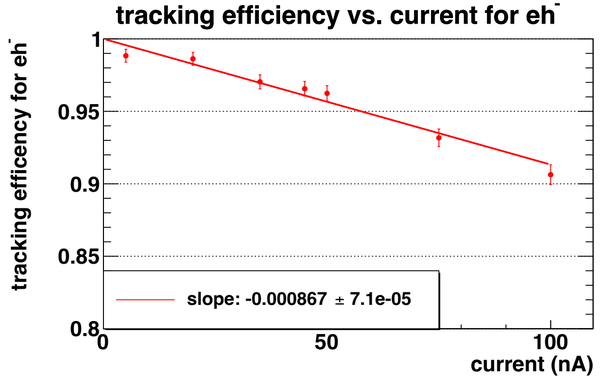
+ new clustering and new tracking





++ new AIs





Discussion for Tracking Efficiency

- Rate of tracking efficiency lost for inbending runs:

Lost rate (per nA)		Pass2	+ new clustering and new tracking	++ new AIs
RGA	eh^+	0.143%	0.0965%	0.0769%
	eh^-	0.140%	0.0955%	0.0740%
RGB	eh^+	0.225%	0.145%	0.106%
	eh^-	0.199%	0.134%	0.100%
RGD	eh^+	0.167%	0.109%	0.0791%
	eh^-	0.159%	0.116%	0.0867%

Tracking efficiency at a given beam current = 100% - lost rate * beam current

- Rate of tracking efficiency lost for outbending runs is smaller than inbending runs.
- With comprehensive updates for forward tracking, rate of tracking efficiency lost is improved to be better than 0.1% per nA for RGA and RGD in-bending, and close to 0.1% per nA for RGB in-bending, whose tracking efficiency is relatively worse since forward tagger is on.

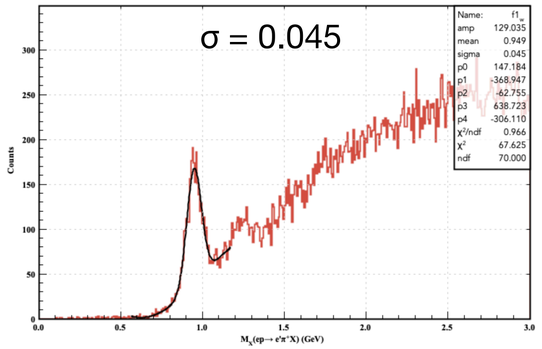
RGA Run 5342; 50nA

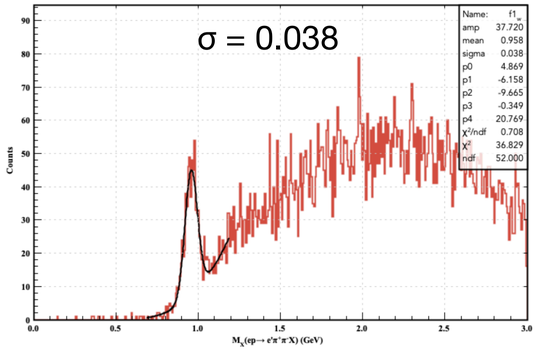
Resolution

$M_x(ep \rightarrow e'\pi^+X)$

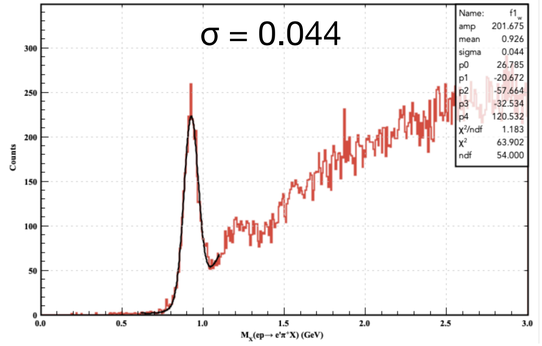
$M_x(ep \rightarrow e'\pi^+\pi^-X)$

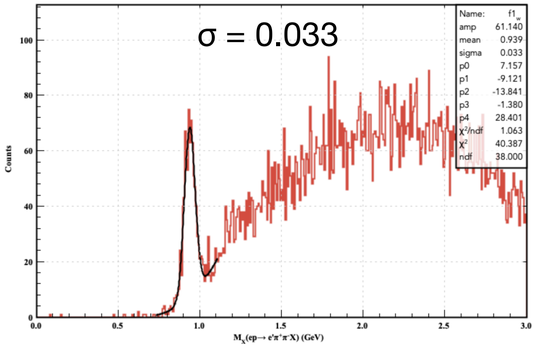
Pass2



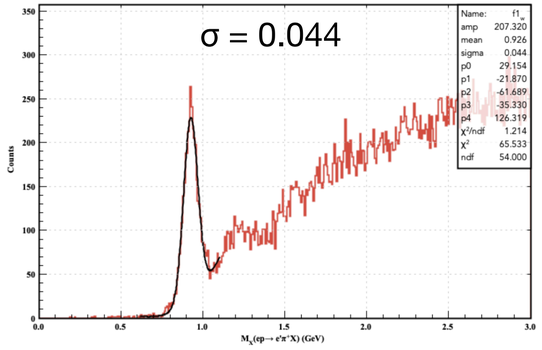


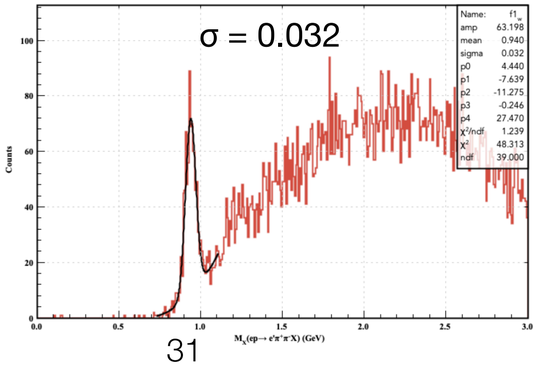
+ new clustering and new tracking





++ new AIs



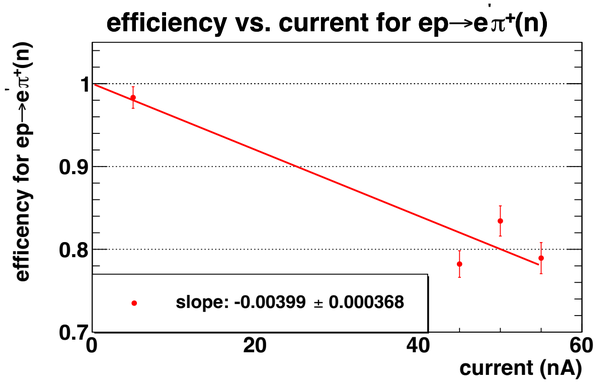


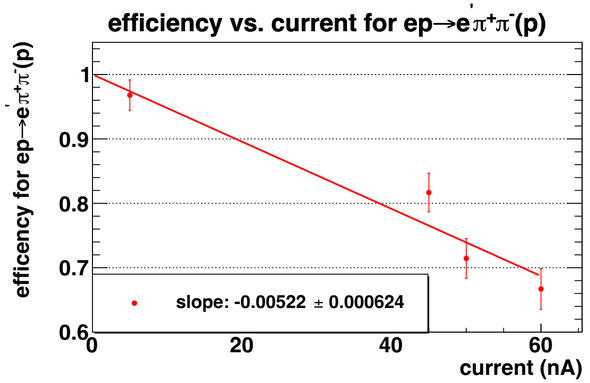
- Events of are selected with cuts for all final-state particles: vz [-15, 5] cm, p > 0.5 GeV and |chi2pid| < 3.
- For $ep \rightarrow e'\pi^+X$, all π^+s are used to calculate missing mass if multiple exist.
- For $ep \rightarrow e'\pi^+\pi^-X$, the first π^+ and π^- are used to calculate missing mass if multiple exist.

RGA Inbending

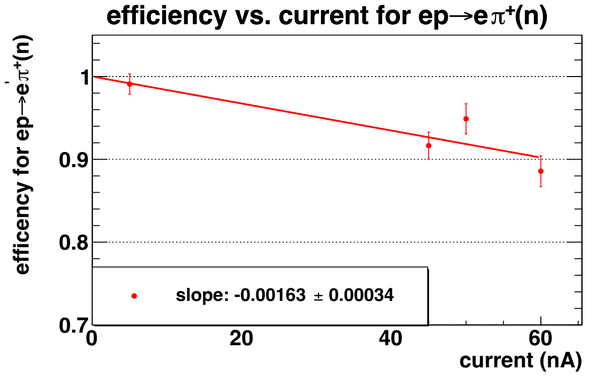
Efficiency for $ep \rightarrow e'\pi^+(n)$ and $ep \rightarrow e'\pi^+\pi^-(p')$

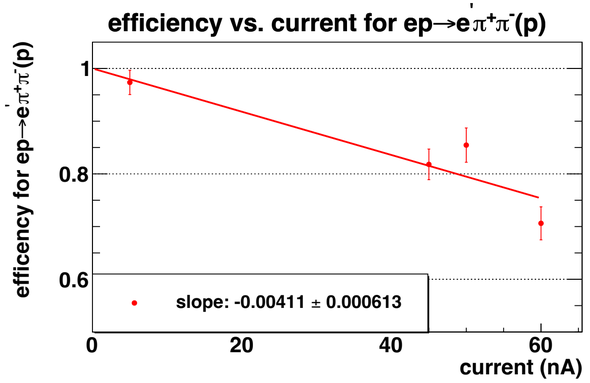
Pass2



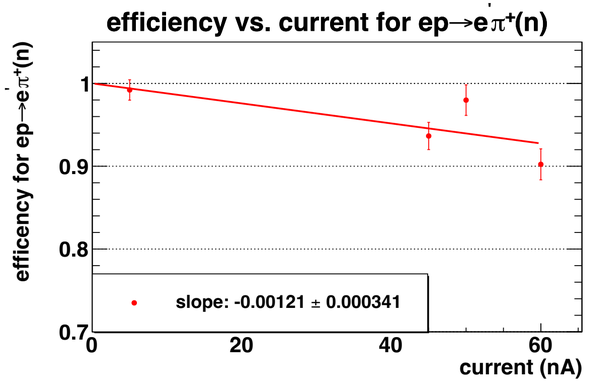


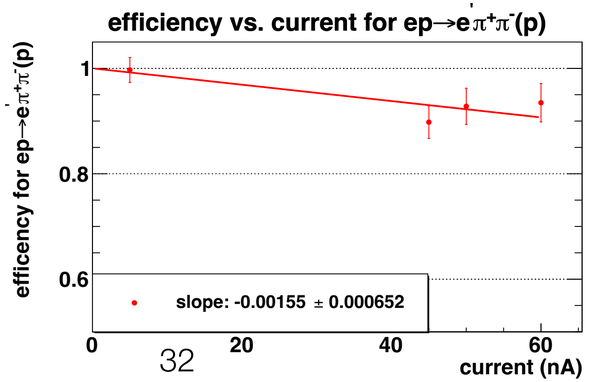
+ new clustering and new tracking





++ new AIs





- The missing mass distributions at each beam current are fitted with Gaussian+pol4 to extract number of events for the reaction, shown in the previous slides.
- The number of exclusive events is then normalized to the number of inclusive events with an electron for efficiency study.

RGA Run 5342; 50nA

Run Time in Clara Workflow

Old denoising is processed before reconstruction.

```
==================================
      BAND:     0.04 ms   0.01%
     CALIB:     0.10 ms   0.01%
       CND:     0.69 ms   0.10%
      CTOF:     2.07 ms   0.30%
     CVTFP:    64.03 ms   9.25%
     CVTSP:    26.85 ms   3.88%
      DCCC:     1.44 ms   0.21%
      DCCR:     9.30 ms   1.34%
      DCDN:    90.36 ms  13.05%
      DCHB:    55.30 ms   7.99%
    DCHTAI:    47.56 ms   6.87%
     DCRAI:     8.22 ms   1.19%
     DCTAI:   170.24 ms  24.59%
      DCTB:   161.67 ms  23.36%
     EBHAI:     0.83 ms   0.12%
      EBHB:     0.85 ms   0.12%
     EBTAI:     1.02 ms   0.15%
      EBTB:     1.08 ms   0.16%
        EC:     1.80 ms   0.26%
       FMT:     0.04 ms   0.01%
    FTOFHB:     2.25 ms   0.33%
    FTOFTB:     2.35 ms   0.34%
      HTCC:     0.07 ms   0.01%
      LTCC:     0.04 ms   0.01%
 MAGFIELDS:     0.02 ms   0.00%
      MLTD:    35.58 ms   5.14%
    RASTER:     0.02 ms   0.00%
    READER:     0.10 ms   0.02%
      RICH:     2.56 ms   0.37%
      RTPC:     0.05 ms   0.01%
     SWAPS:     0.24 ms   0.03%
    WRITER:     5.42 ms   0.78%
==================================
     TOTAL:   692.20 ms
```

New track finding

New denoising
Old HB tracking
New HB tracking

Old track finding

CPU-time/ event/thread	Denoising	Track finding	HB tracking	Total
Old	~150 ms	35.58 ms	55.30 ms	~240.88 ms
New	90.36 ms	1.44 ms	47.56 ms	139.36 ms
Improved	~40%	~24 times	~14%	~42%

- New denoising processes hits in each of 6 sectors as batch, instead of sector by sector.
- New cluster-combo prediction avoids to estimate missing cluster for 5-cluster combos by auto-encoder, which is much more complicated, so much slower than MLP. Also, the new engine predicts cluster-combos as batch, instead of one by one cluster-combo.

Summary

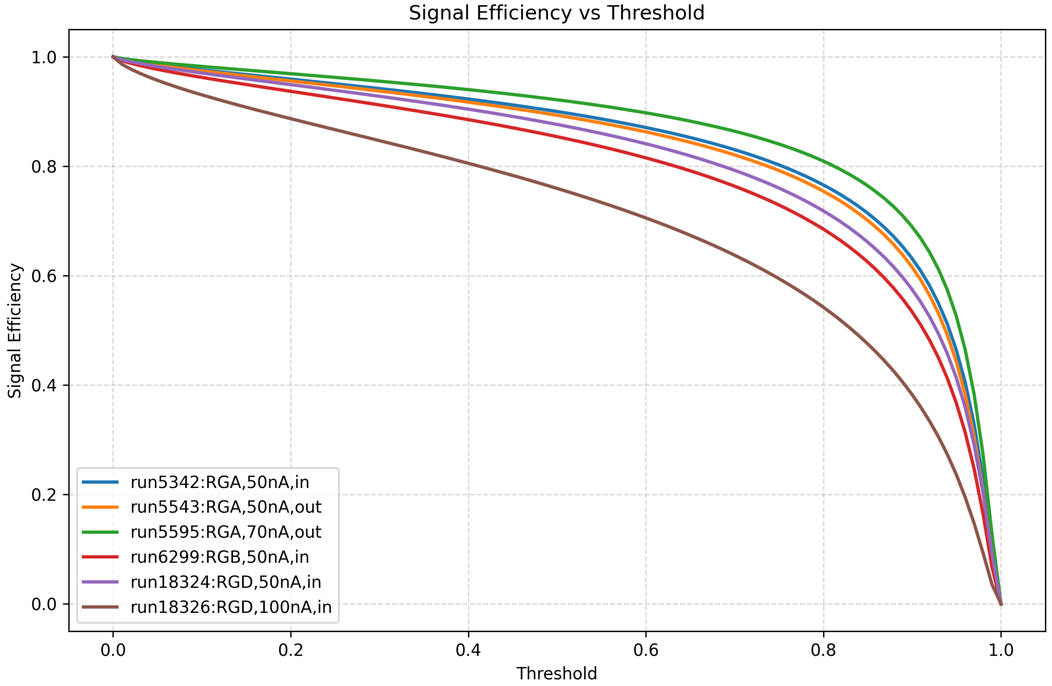
- Traditional reconstruction algorithms often struggle to capture global correlations among hits or clusters along particle trajectories, particularly in the presence of inhomogeneous magnetic fields and high background conditions. In contrast, AI-based models are capable of learning these global spatial correlations, thereby providing effective assistance to track reconstruction and significantly reducing background impact.
- At different stages of the reconstruction pipeline, models employing appropriate AI techniques with carefully tuned hyperparameters are applied, and thresholds are optimized to maximize performance.
- With the new AI models and earlier improvements to clustering and tracking algorithms, the tracking efficiency loss rate is below 0.1% per nA for RGA and RGD in-bending and approximately 0.1% per nA for RGB in-bending, with smaller losses for out-bending.
- The new AI models also significantly speed up reconstruction. Combined processing for hit denoising, track finding and HB tracking is approximately 42% faster.
- The new AI models do not significantly TB tracking resolution, while HB tracking resolution is substantially improved with the AI track-state estimator replacing KF.
- All models depend on the magnetic field configuration, which is common for most runs. Validation studies indicate that the models are not sensitive to polarity of magnetic field.
- Further validations are ongoing, particularly to assess potential dependencies of the models on run groups and luminosity, especially for the denoising model since it is more sensitive to experiment conditions than the models for track finding and HB tracking.
- Thank Richard Tyson, Nathan Baltzell, Raffaella De Vita, and other collaborators for their valuable contributions.

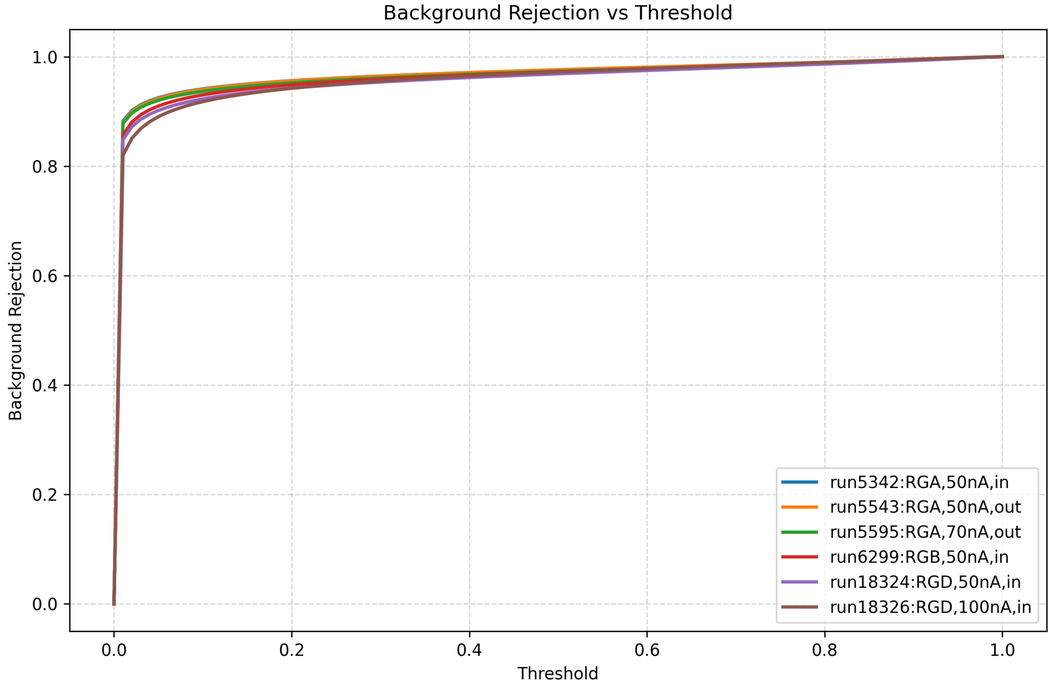
Backup Slides

Abstract

- Traditional reconstruction algorithms often struggle to capture global correlations among hits or clusters along particle trajectories, particularly in the presence of inhomogeneous magnetic fields and high background conditions. In contrast, AI-based models are capable of learning these global spatial correlations directly from data, thereby providing effective assistance to track reconstruction and significantly reducing background impact. This presentation will report recent progress in AI-assisted forward tracking and discuss its impact on tracking efficiency, resolution, and reconstruction processing speed.

Performance vs. Threshold Among Different Runs



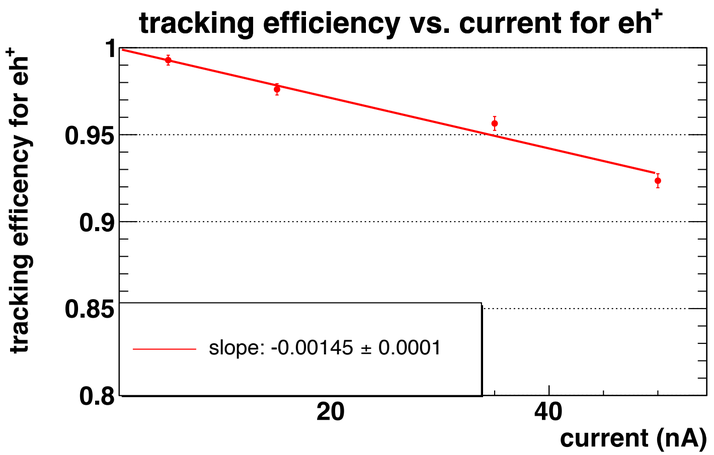


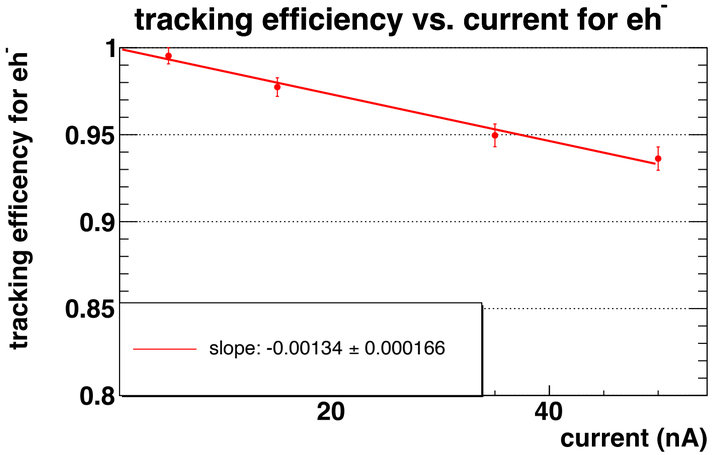
- The model is trained by samples from RGA run 5342, which is a inbending run with 50nA beam current.
- Performance for background rejection vs. threshold is almost the same for inbending/outbending and different luminosity RGA runs, while a little bit worse for RGB and RGB runs.
- For performance for signal efficiency vs. threshold:
  - Similar for in-bending and out-bending of RGA runs with the same luminosity
  - Different for different luminosity
  - Worse for RGB and RGD runs
- It tells that model is independent of inbending and outbending, but dependent of luminosity and run group.
- Different model for different run group with luminosity of production runs will be trained and tested.

RGB Inbending

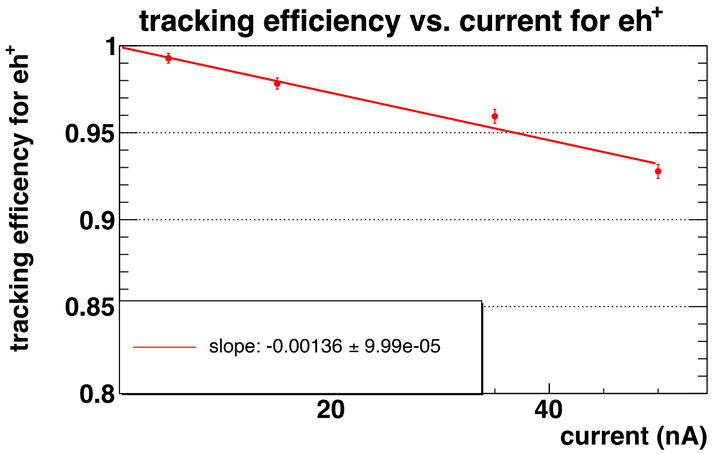
Tracking Efficiency vs. Luminosity

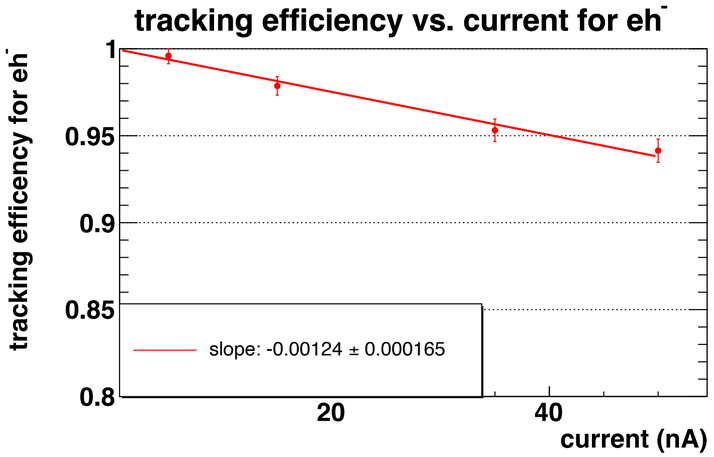
Old Denoising





New Denoising

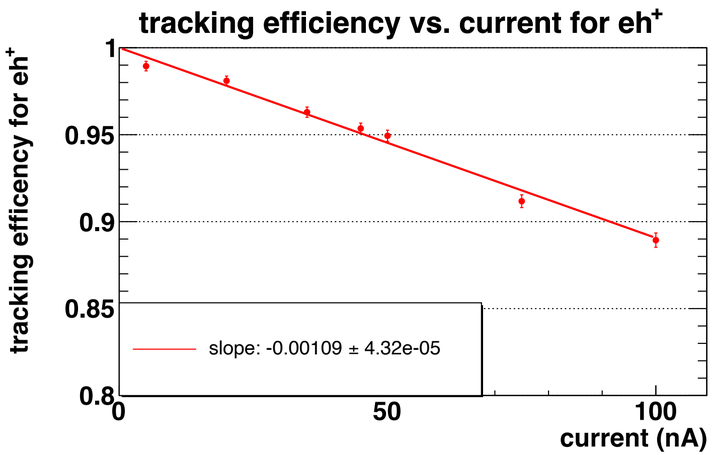


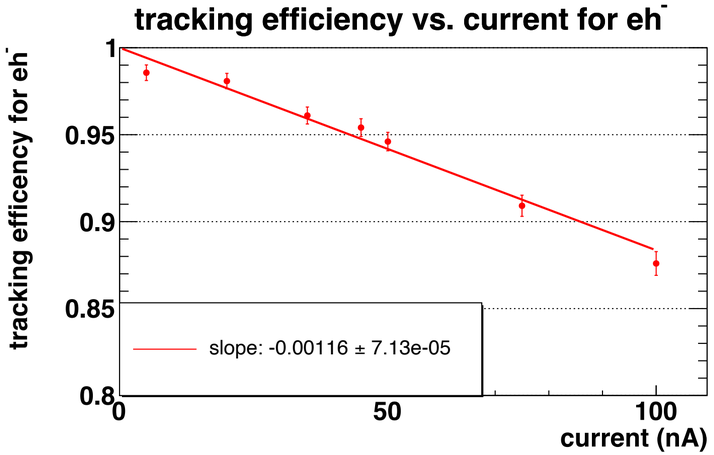


RGD Inbending

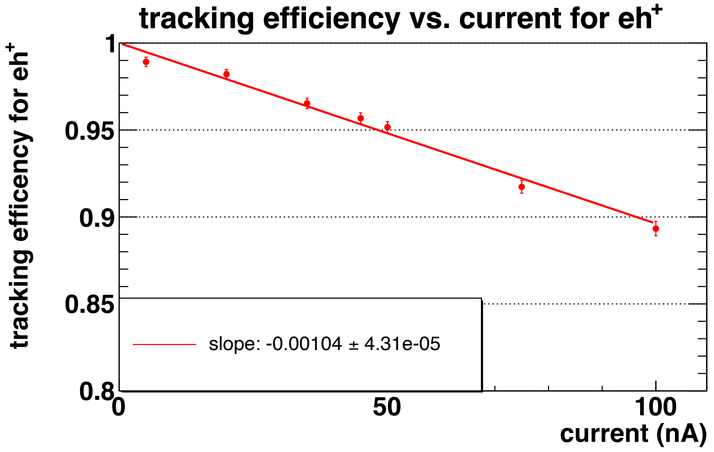
Tracking Efficiency vs. Luminosity

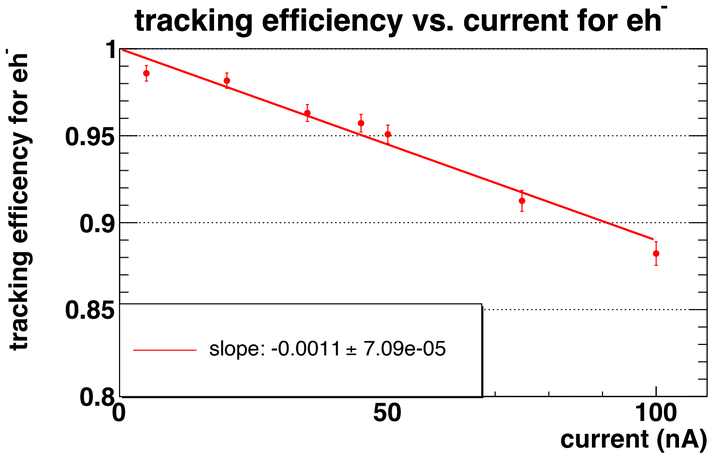
Old Denoising





New Denoising





RGA Run 5342; 50nA

Run Time in Clara Workflow

BAND:	0.04 ms	0.01%	BAND:	0.04 ms	0.01%
CND:	0.62 ms	0.08%	CND:	0.62 ms	0.08%
CTOF:	1.56 ms	0.21%	CTOF:	1.56 ms	0.21%
CVTFP:	69.50 ms	9.46%	CVTFP:	69.50 ms	9.46%
CVTSP:	28.73 ms	3.91%	CVTSP:	28.73 ms	3.91%
DCCR:	7.22 ms	0.98%	DCCR:	7.22 ms	0.98%
DCDN:	90.50 ms	12.31%	DCDN:	90.50 ms	12.31%
DCHAI:	59.11 ms	8.04%	DCHAI:	59.11 ms	8.04%
DCHB:	75.68 ms	10.30%	DCHB:	75.68 ms	10.30%
DCTAI:	179.56 ms	24.43%	DCTAI:	179.56 ms	24.43%
DCTB:	174.82 ms	23.79%	DCTB:	174.82 ms	23.79%
EBHAI:	0.76 ms	0.10%	EBHAI:	0.76 ms	0.10%
EBHB:	0.80 ms	0.11%	EBHB:	0.80 ms	0.11%
EBTAI:	0.93 ms	0.13%	EBTAI:	0.93 ms	0.13%
EBTB:	0.99 ms	0.13%	EBTB:	0.99 ms	0.13%
EC:	1.57 ms	0.21%	EC:	1.57 ms	0.21%
FMT:	0.04 ms	0.01%	FMT:	0.04 ms	0.01%
FTOFHB:	1.96 ms	0.27%	FTOFHB:	1.96 ms	0.27%
FTOFTB:	1.99 ms	0.27%	FTOFTB:	1.99 ms	0.27%
HTCC:	0.06 ms	0.01%	HTCC:	0.06 ms	0.01%
LTCC:	0.04 ms	0.01%	LTCC:	0.04 ms	0.01%
MAGFIELDS:	0.03 ms	0.00%	MAGFIELDS:	0.03 ms	0.00%
MLTD:	30.48 ms	4.15%	MLTD:	30.48 ms	4.15%
RASTER:	0.02 ms	0.00%	RASTER:	0.02 ms	0.00%
READER:	0.09 ms	0.01%	READER:	0.09 ms	0.01%
RICH:	2.44 ms	0.33%	RICH:	2.44 ms	0.33%
RTPC:	0.05 ms	0.01%	RTPC:	0.05 ms	0.01%
SWAPS:	0.26 ms	0.04%	SWAPS:	0.26 ms	0.04%
WRITER:	5.07 ms	0.69%	WRITER:	5.07 ms	0.69%
TOTAL:	734.93 ms			734.93 ms	

Speed of the new model is faster than the old model (~150 ms/event/thread).

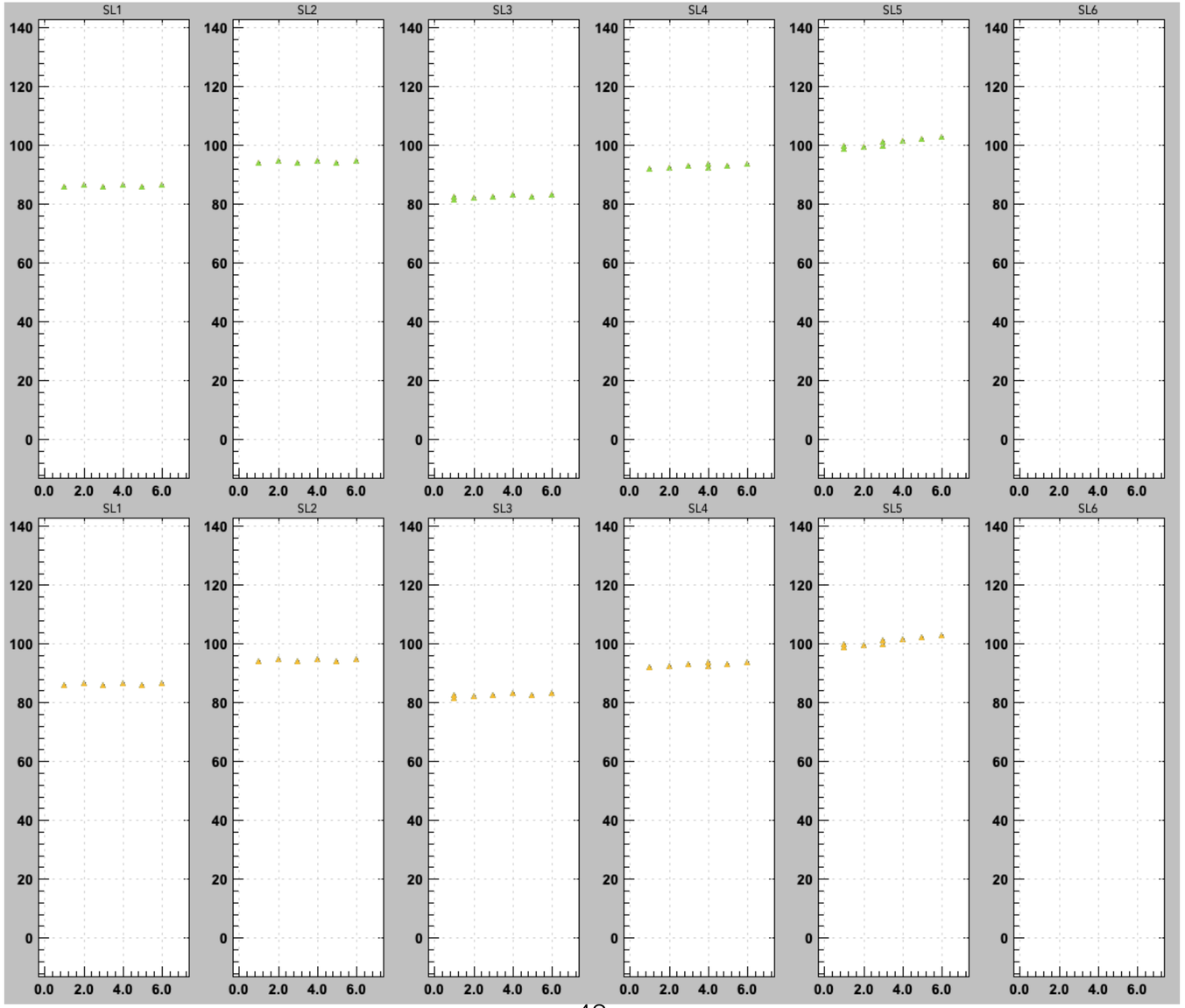
Process in the New Engine

- Firstly, predict 6-cluster combos from all clusters:
  - For clusters in each sector, build 6-cluster combos, where 6 clusters are separately from 6 superlayers for each combo
  - Input all combos into the 6-cluster model for prediction by batch process
  - Combos are cut off with threshold as 0.95 for the best balance of 6-cluster and 5-cluster tracks.
  - For combos with overlapped clusters, a combo with the highest probability is chosen.
- Then, predict 5-cluster combos from remaining clusters:
  - For clusters in each sector, build 5-cluster combos, where 5 clusters are separately from any 5 superlayers for each combo
  - Input all combos into the 5-cluster model for prediction by batch process
  - Combos are cut off with threshold as 0.05 for keeping most positive cases and eliminating most negative cases.
  - For combos with overlapped clusters, a combo with the highest probability is chosen.
- Noah’s model is in development, and is supposed to be sensitive to distinguish combos, where one or more clusters are close, and other clusters are shared. The model might be applied to update treatment for combos with overlapped clusters.

Example 1

Missed by old model

Predicted by new model



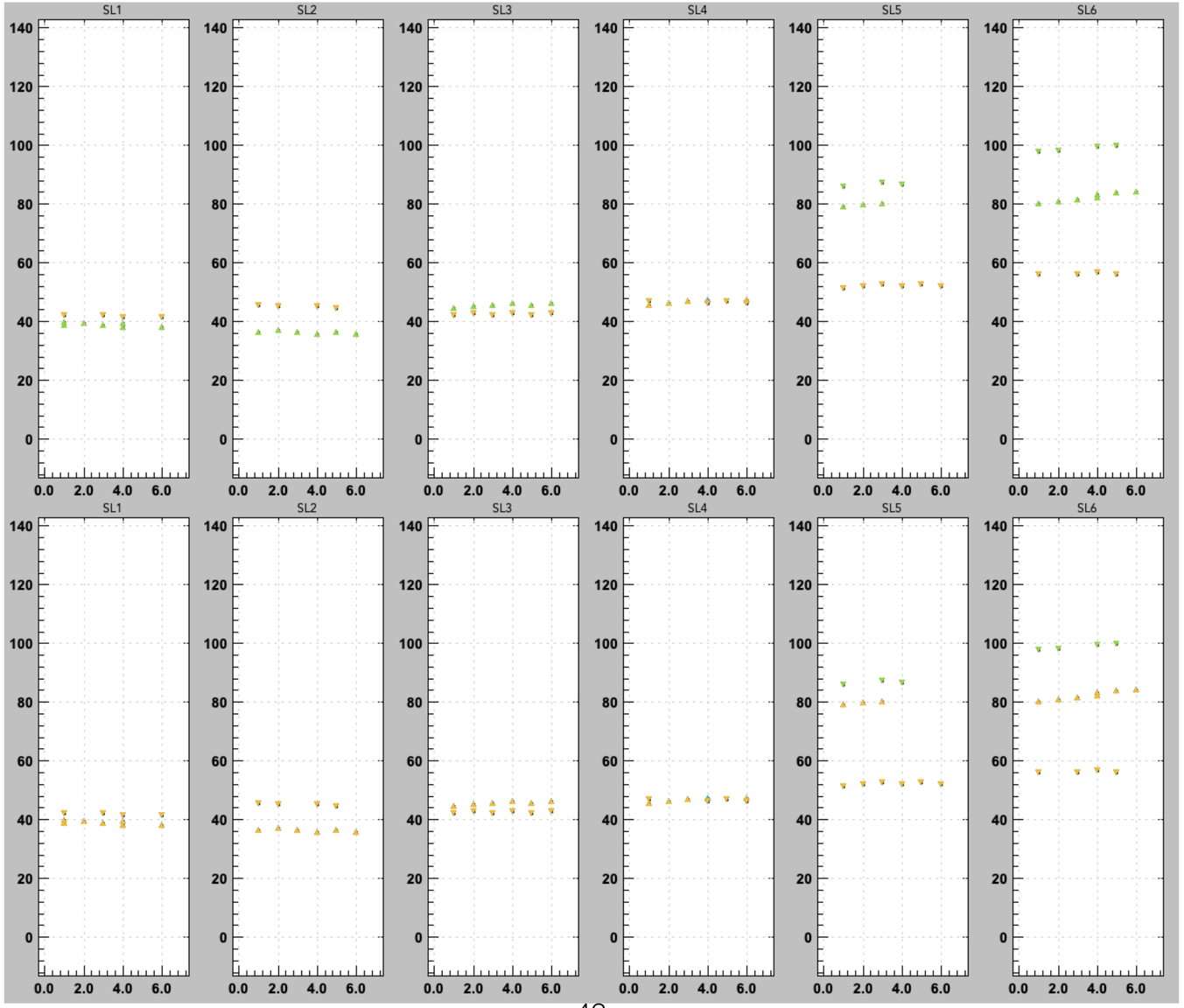
- Up-triangle: signal hits
- Down-triangle: noise hits

- Green: Failed at HB tracking
- Blue: passed at HB tracking
- Yellow: Passed at TB tracking

Example 2

A combo with mixture of signal and noise cluster is predicted by old model

Two expected combos predicted by new model



- Up-triangle: signal hits
- Down-triangle: noise hits

- Green: Failed at HB tracking
- Blue: passed at HB tracking
- Yellow: Passed at TB tracking

RGA Run 5342; 50nA

Run Time in Clara Workflow

Old models

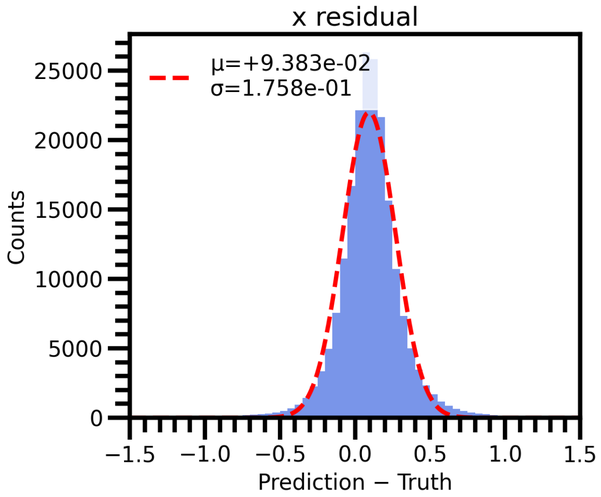
BAND:	0.03 ms	0.01%
CND:	0.64 ms	0.11%
CTOF:	1.69 ms	0.29%
CVTFP:	61.91 ms	10.76%
CVTSP:	25.92 ms	4.51%
DCCR:	9.29 ms	1.62%
DCHAI:	52.15 ms	9.07%
DCHB:	71.77 ms	12.47%
DCTAI:	156.37 ms	27.18%
DCTB:	151.86 ms	26.40%
EBHAI:	0.76 ms	0.13%
EBHB:	0.79 ms	0.14%
EBTAI:	0.92 ms	0.16%
EBTB:	0.98 ms	0.17%
EC:	1.60 ms	0.28%
FMT:	0.04 ms	0.01%
FTOFHB:	2.02 ms	0.35%
FTOFTB:	2.02 ms	0.35%
HTCC:	0.06 ms	0.01%
LTCC:	0.04 ms	0.01%
MAGFIELDS:	0.02 ms	0.00%
MLTD:	26.86 ms	4.67%
RASTER:	0.02 ms	0.00%
READER:	0.09 ms	0.02%
RICH:	2.29 ms	0.40%
RTPC:	0.04 ms	0.01%
SWAPS:	0.25 ms	0.04%
WRITER:	4.89 ms	0.85%
TOTAL:	575.30 ms	

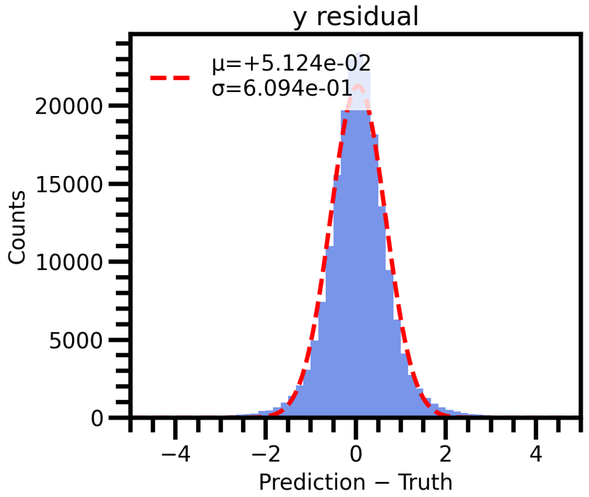
New models

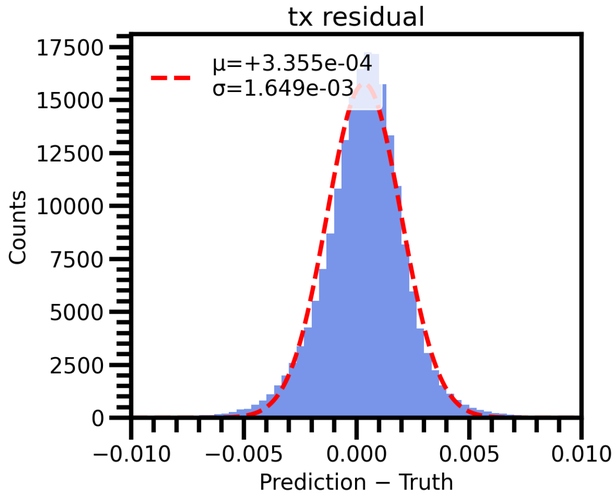
BAND:	0.02 ms	0.00%
CND:	0.52 ms	0.10%
CTOF:	1.16 ms	0.22%
CVTFP:	57.77 ms	10.95%
CVTSP:	24.30 ms	4.61%
DCCC:	0.90 ms	0.17%
DCCR:	8.34 ms	1.58%
DCHAI:	50.01 ms	9.48%
DCHB:	67.32 ms	12.76%
DCTAI:	157.31 ms	29.83%
DCTB:	145.30 ms	27.55%
EBHAI:	0.66 ms	0.13%
EBHB:	0.66 ms	0.13%
EBTAI:	0.80 ms	0.15%
EBTB:	0.81 ms	0.15%
EC:	1.29 ms	0.25%
FMT:	0.02 ms	0.00%
FTOFHB:	1.53 ms	0.29%
FTOFTB:	1.51 ms	0.29%
HTCC:	0.04 ms	0.01%
LTCC:	0.02 ms	0.00%
MAGFIELDS:	0.01 ms	0.00%
RASTER:	0.01 ms	0.00%
READER:	0.07 ms	0.01%
RICH:	1.95 ms	0.37%
RTPC:	0.03 ms	0.01%
SWAPS:	0.24 ms	0.05%
WRITER:	4.77 ms	0.90%
TOTAL:	527.38 ms	

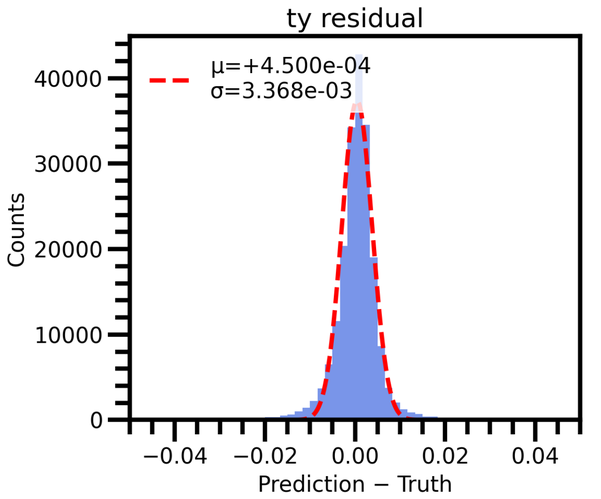
- The old models use the auto-encoder to predict features of missing clusters for 5-cluster combos, while the new models directly use MLP to take binary classification.
- The auto-encoder is must slower than MLP.

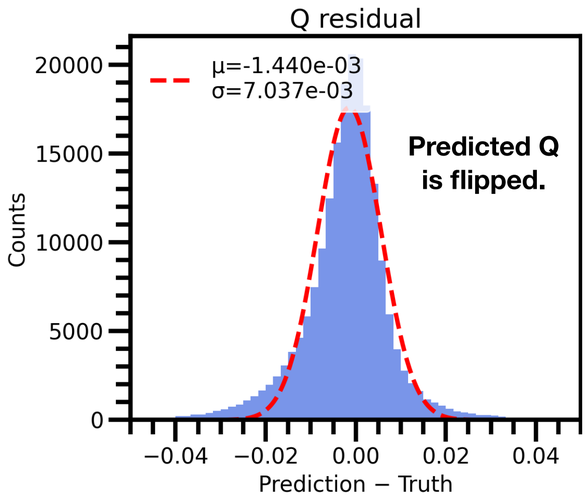
Validation for Model Trained by Inbending Run Applied into Outbending Run





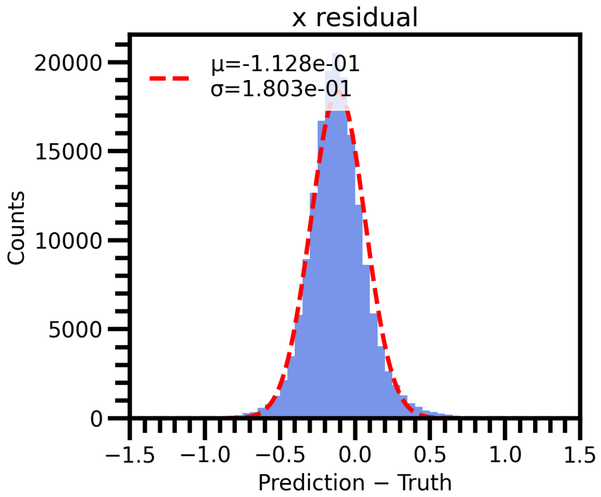


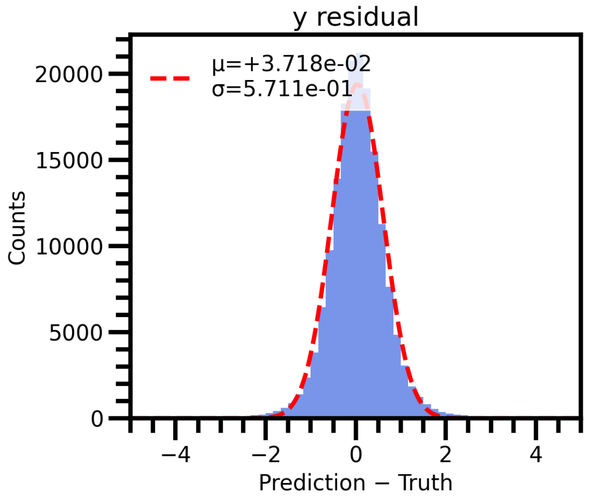


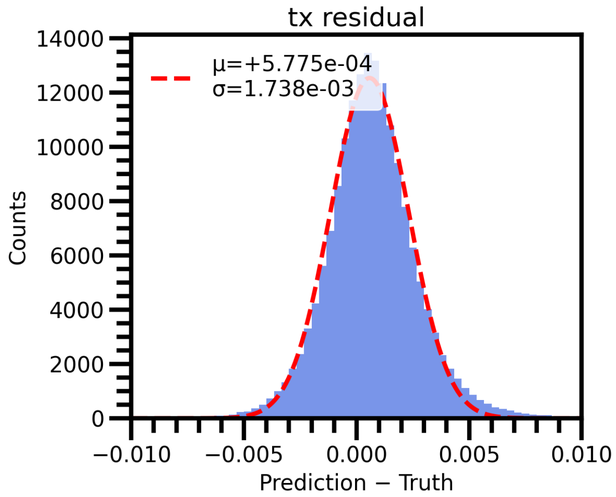


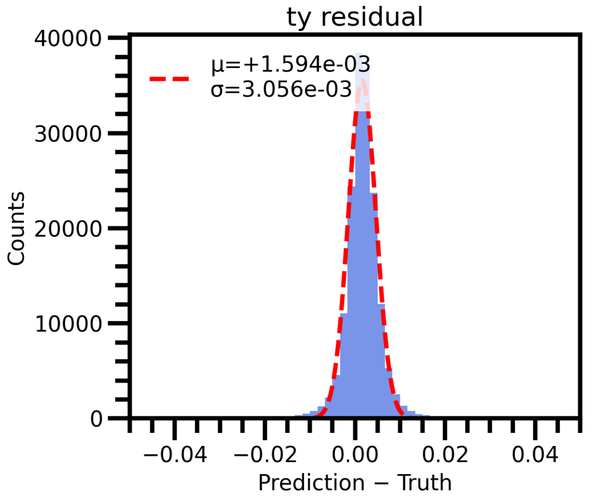
Truth is TB track state

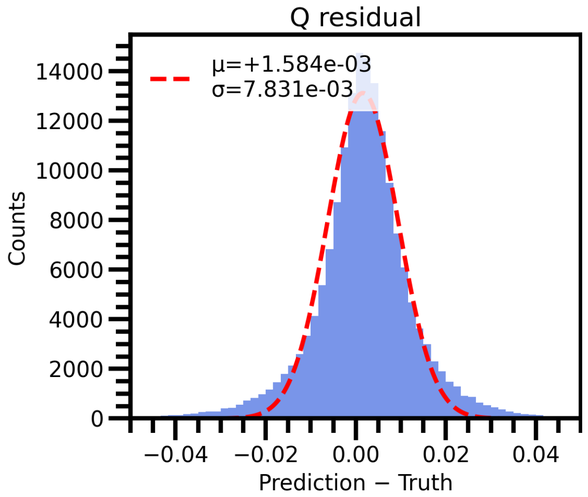
Validation for Model Trained by Data Applied into RGA + Background MC





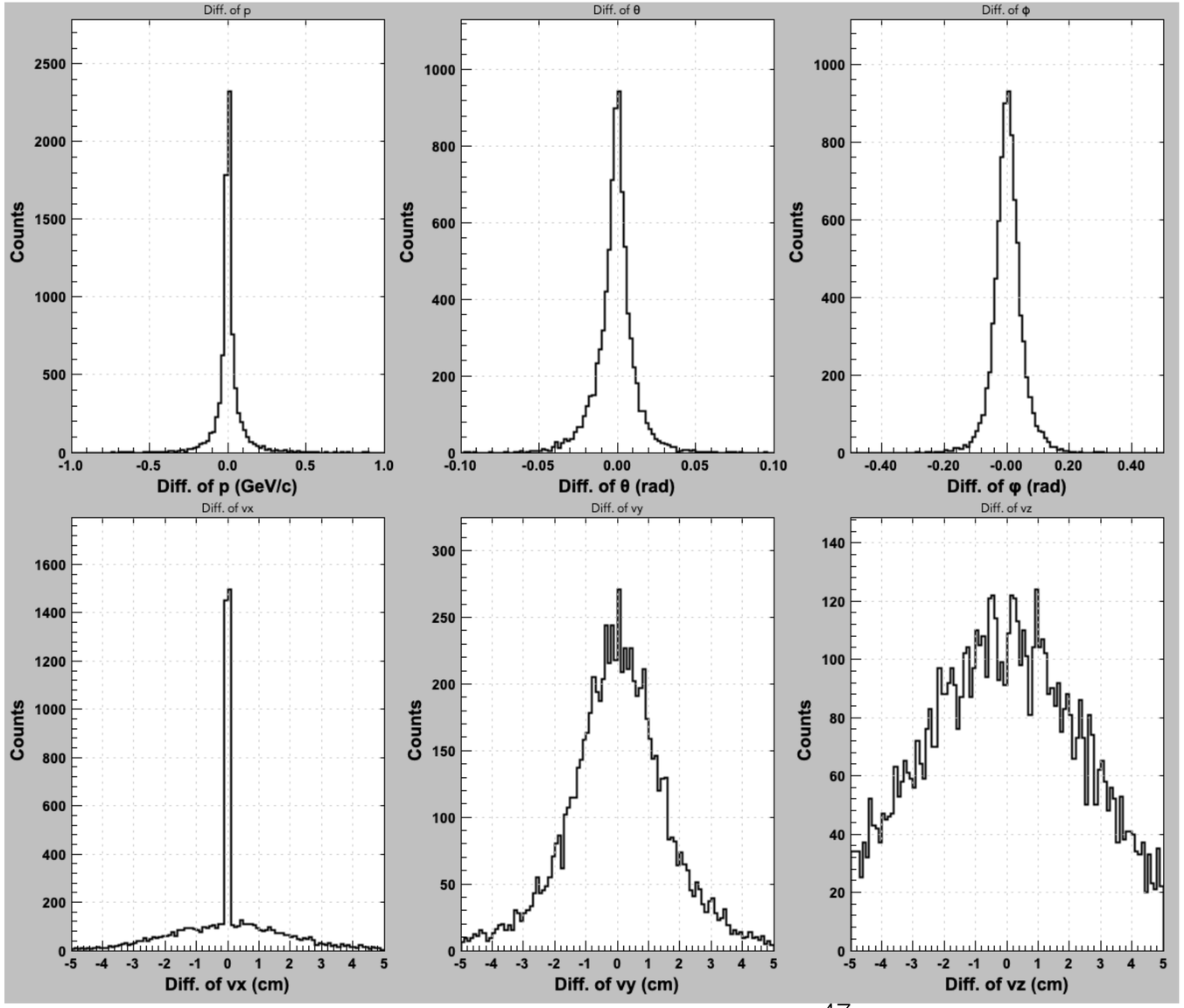






Truth is TB track state

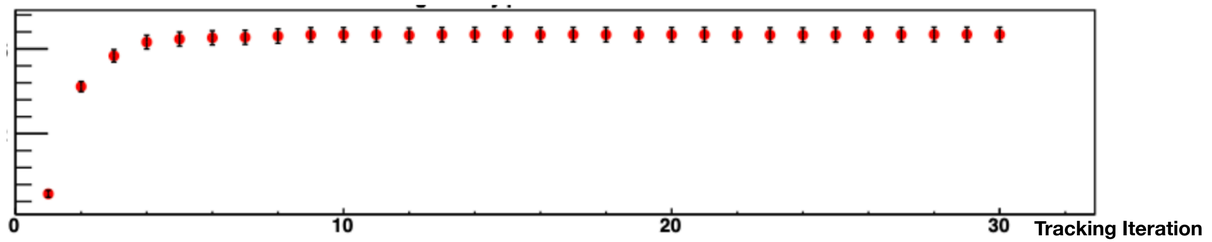
Difference between HB Tracks



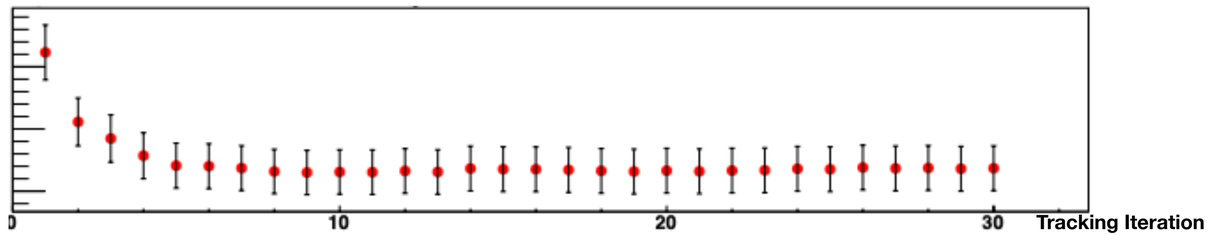
No shift for peaks, but significant difference for resolution.

If Track State by AI as Initial State for HB KF Tracking

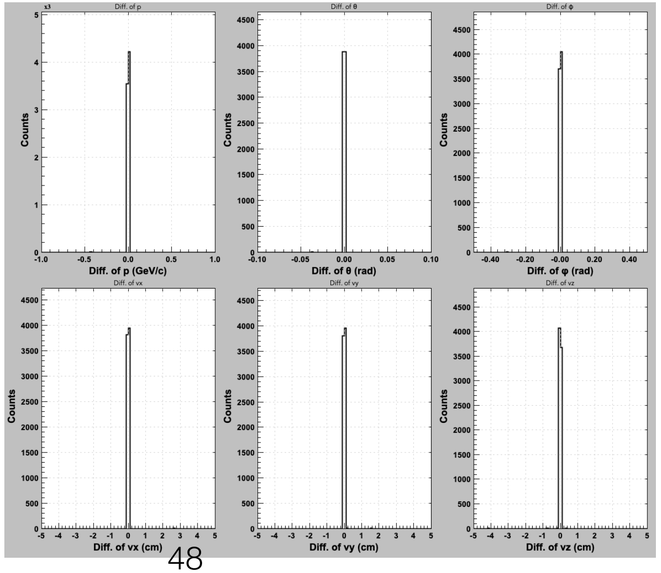
- If track state by AI is applied into HB KF tracking as initial state, track resolution becomes worse along iterations, until stable, like:



- For current HB KF tracking with initial state, roughly determined by 3 crosses, track resolution becomes better along iterations, until stable, like:



- Finally, tracking results are close. Plots show difference of HB tracks with two different initial state as input for KF.

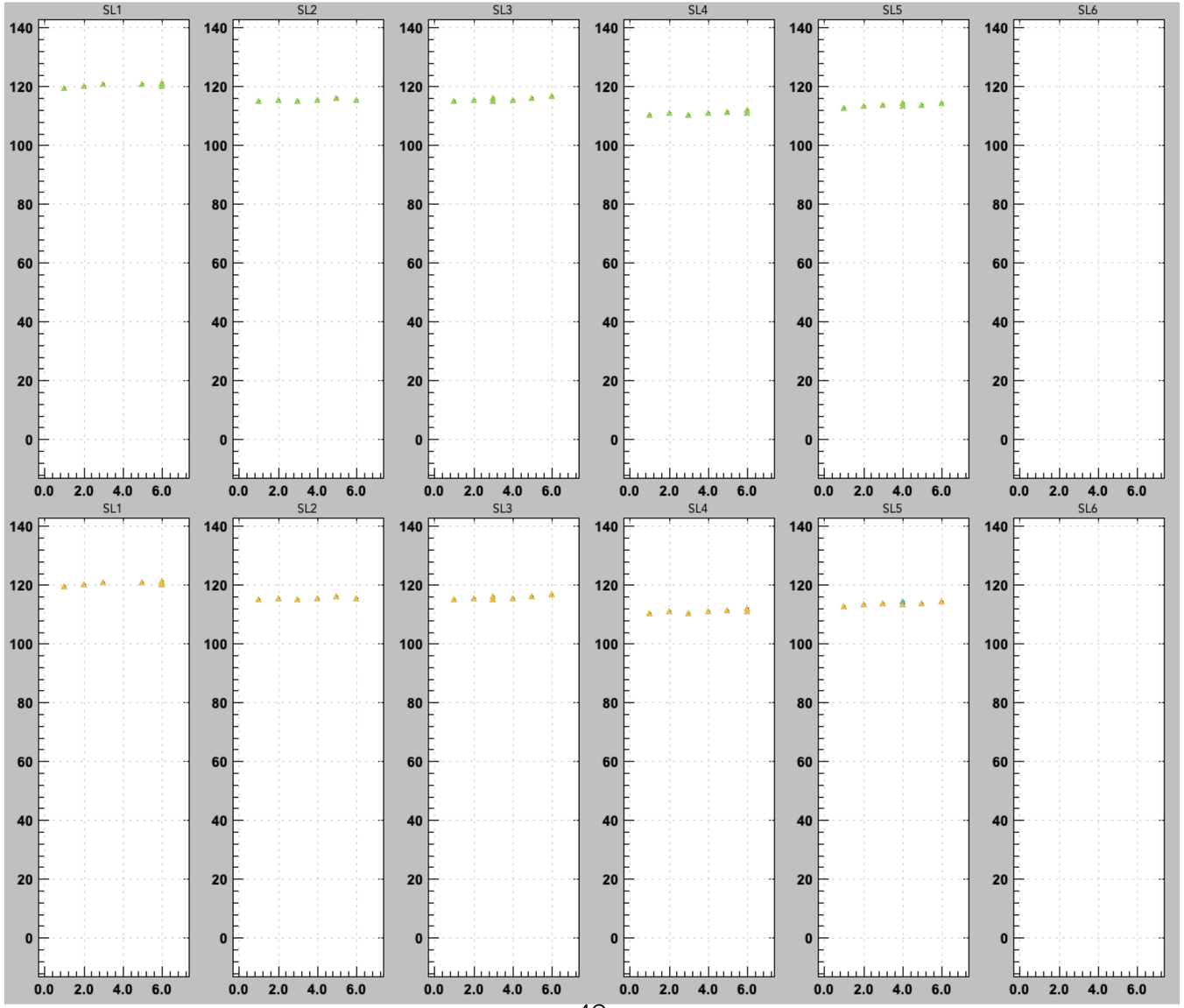


Example 1

Failed for HB tracking

The reason could be that KF explodes due to rough initial state.

Passed for TB tracking with HB track estimated by AI



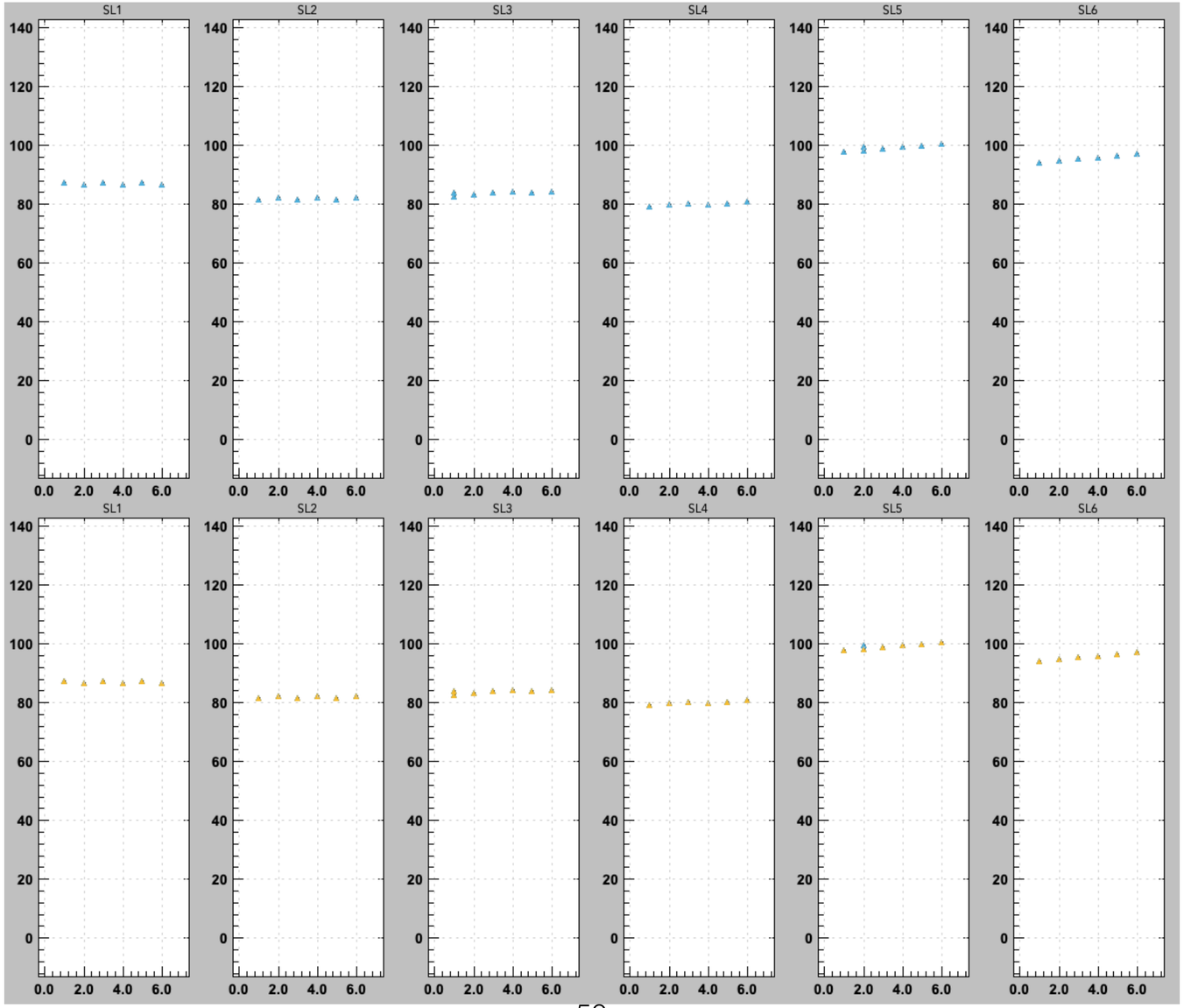
- **Up-triangle: signal hits**
- **Down-triangle: noise hits**

- **Green: Failed at HB tracking**
- **Blue: passed at HB tracking**
- **Yellow: Passed at TB tracking**

Example 2

Passed for HB tracking, while failed for TB tracking

Passed for TB tracking with HB track estimated by AI



- **Up-triangle: signal hits**
- **Down-triangle: noise hits**

- **Green: Failed at HB tracking**
- **Blue: passed at HB tracking**
- **Yellow: Passed at TB tracking**

CPU Time

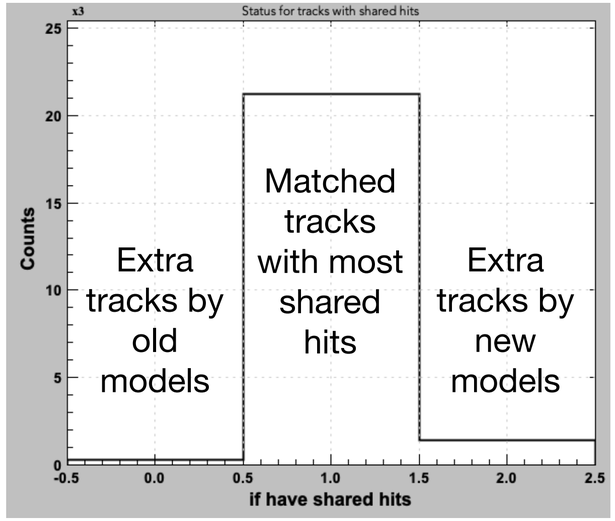
AI-assisted HB tracking

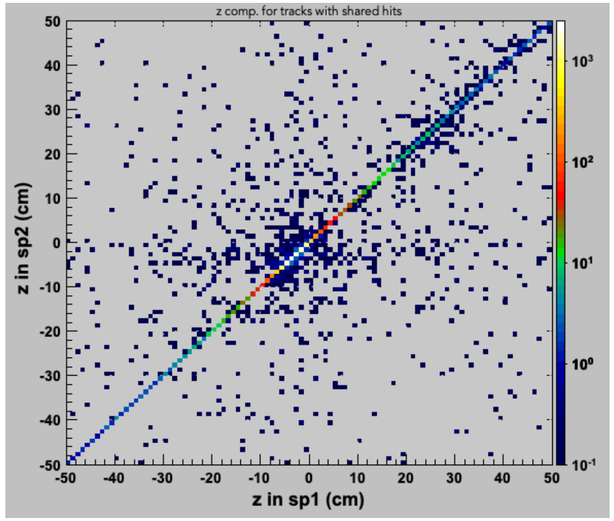
```
==================================
      BAND:     0.03 ms  0.01%
       CND:     0.64 ms  0.11%
      CTOF:     1.69 ms  0.29%
     CVTFP:    61.91 ms 10.76%
     CVTSP:    25.92 ms  4.51%
      DCCR:     9.29 ms  1.62%
     DCHAI:    52.15 ms  9.07%
      DCHB:    71.77 ms 12.47%
     DCTAI:   156.37 ms 27.18%
      DCTB:   151.86 ms 26.40%
     EBHAI:     0.76 ms  0.13%
      EBHB:     0.79 ms  0.14%
     EBTAI:     0.92 ms  0.16%
      EBTB:     0.98 ms  0.17%
        EC:     1.60 ms  0.28%
       FMT:     0.04 ms  0.01%
    FTOFHB:     2.02 ms  0.35%
    FTOFTB:     2.02 ms  0.35%
      HTCC:     0.06 ms  0.01%
      LTCC:     0.04 ms  0.01%
 MAGFIELDS:     0.02 ms  0.00%
      MLTD:    26.86 ms  4.67%
    RASTER:     0.02 ms  0.00%
    READER:     0.09 ms  0.02%
      RICH:     2.29 ms  0.40%
      RTPC:     0.04 ms  0.01%
     SWAPS:     0.25 ms  0.04%
    WRITER:     4.89 ms  0.85%
==================================
      TOTAL:   575.30 ms
```

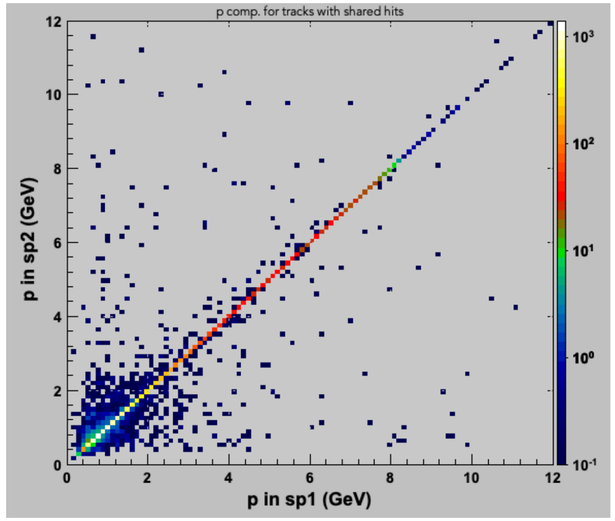
HB tracking by AI estimator

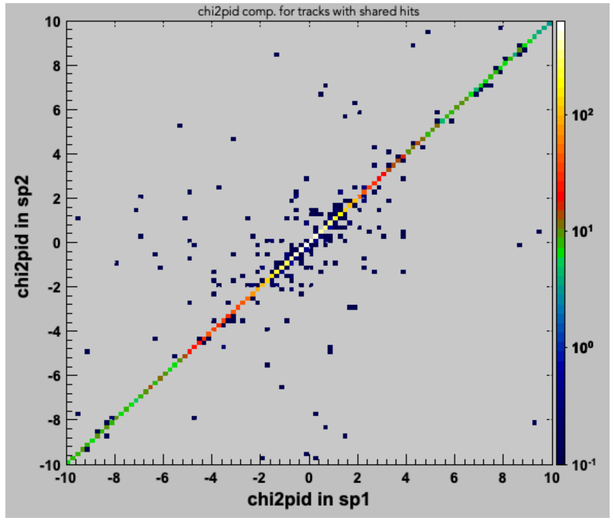
```
==================================
      BAND:     0.03 ms  0.01%
       CND:     0.61 ms  0.10%
      CTOF:     1.56 ms  0.26%
     CVTFP:    65.67 ms 10.85%
     CVTSP:    27.62 ms  4.56%
      DCCR:    10.22 ms  1.69%
      DCHB:    76.67 ms 12.66%
    DCHTAI:    45.63 ms  7.54%
     DCTAI:   167.37 ms 27.64%
      DCTB:   162.79 ms 26.89%
     EBHAI:     0.77 ms  0.13%
      EBHB:     0.81 ms  0.13%
     EBTAI:     0.95 ms  0.16%
      EBTB:     1.00 ms  0.17%
        EC:     1.56 ms  0.26%
       FMT:     0.04 ms  0.01%
    FTOFHB:     1.89 ms  0.31%
    FTOFTB:     1.91 ms  0.32%
      HTCC:     0.06 ms  0.01%
      LTCC:     0.04 ms  0.01%
 MAGFIELDS:     0.02 ms  0.00%
      MLTD:    30.31 ms  5.01%
    RASTER:     0.02 ms  0.00%
    READER:     0.10 ms  0.02%
      RICH:     2.37 ms  0.39%
      RTPC:     0.04 ms  0.01%
     SWAPS:     0.27 ms  0.05%
    WRITER:     5.10 ms  0.84%
==================================
      TOTAL:   605.43 ms
```

Event-by-Event Comparison between Old and New Models

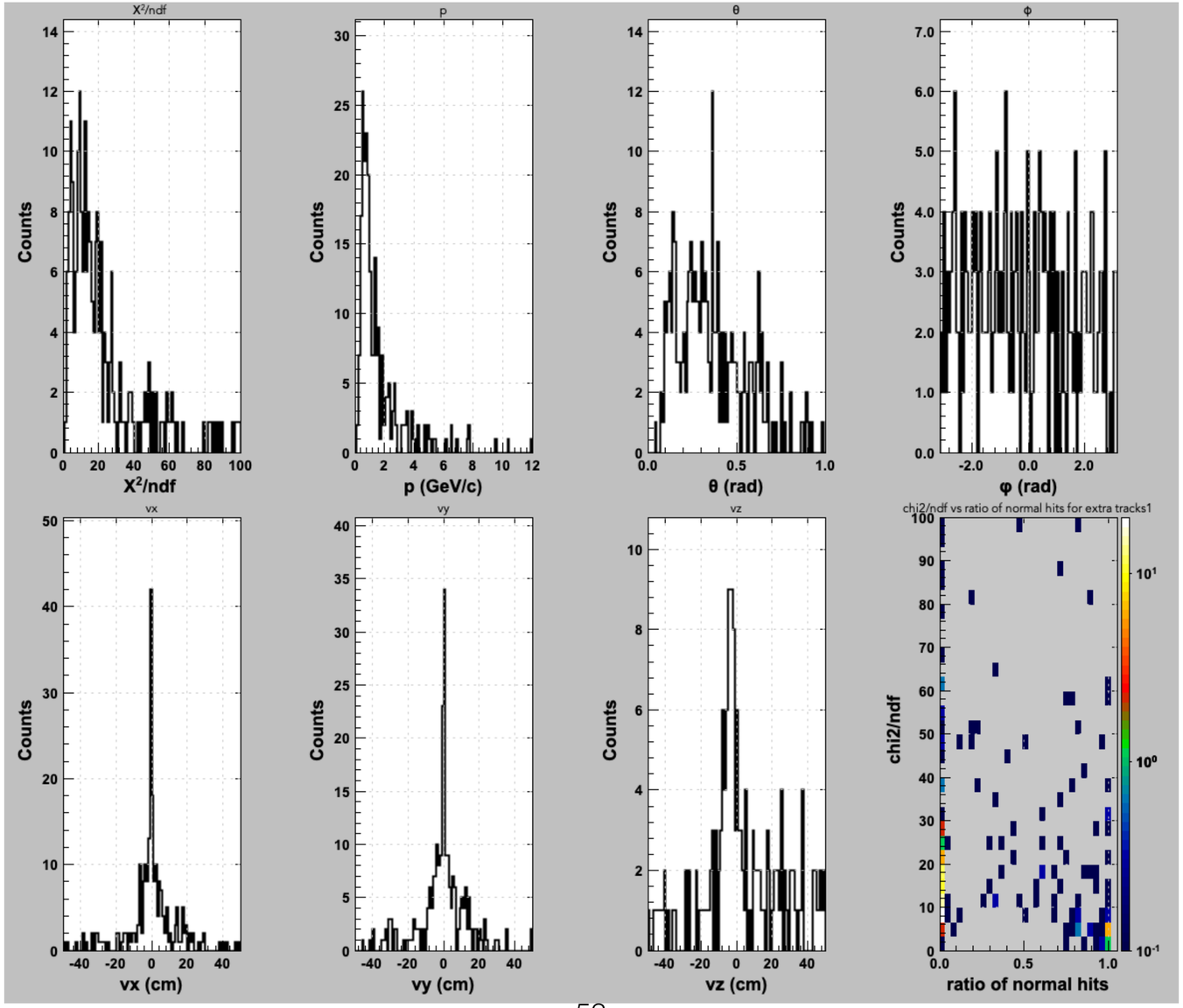




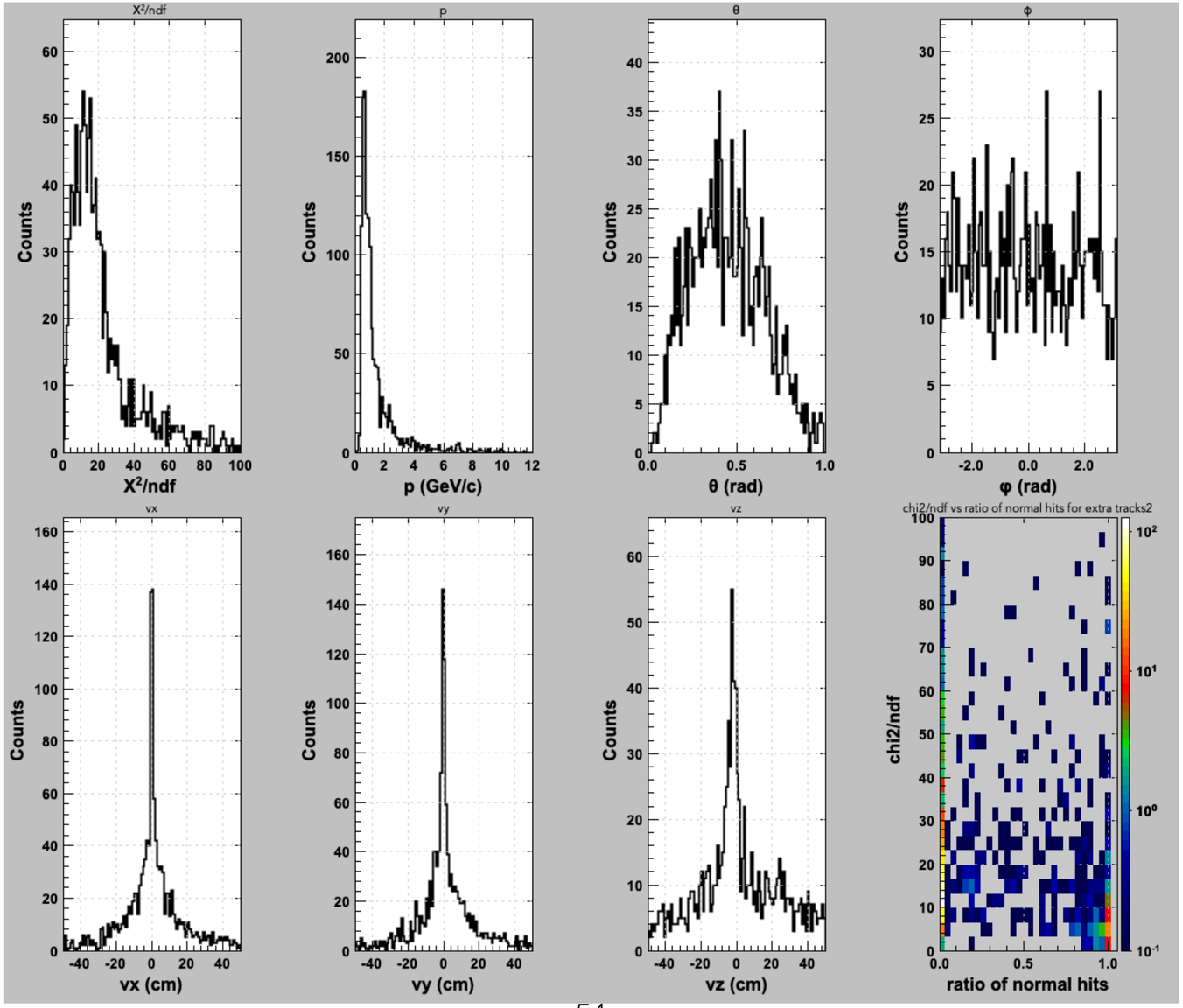




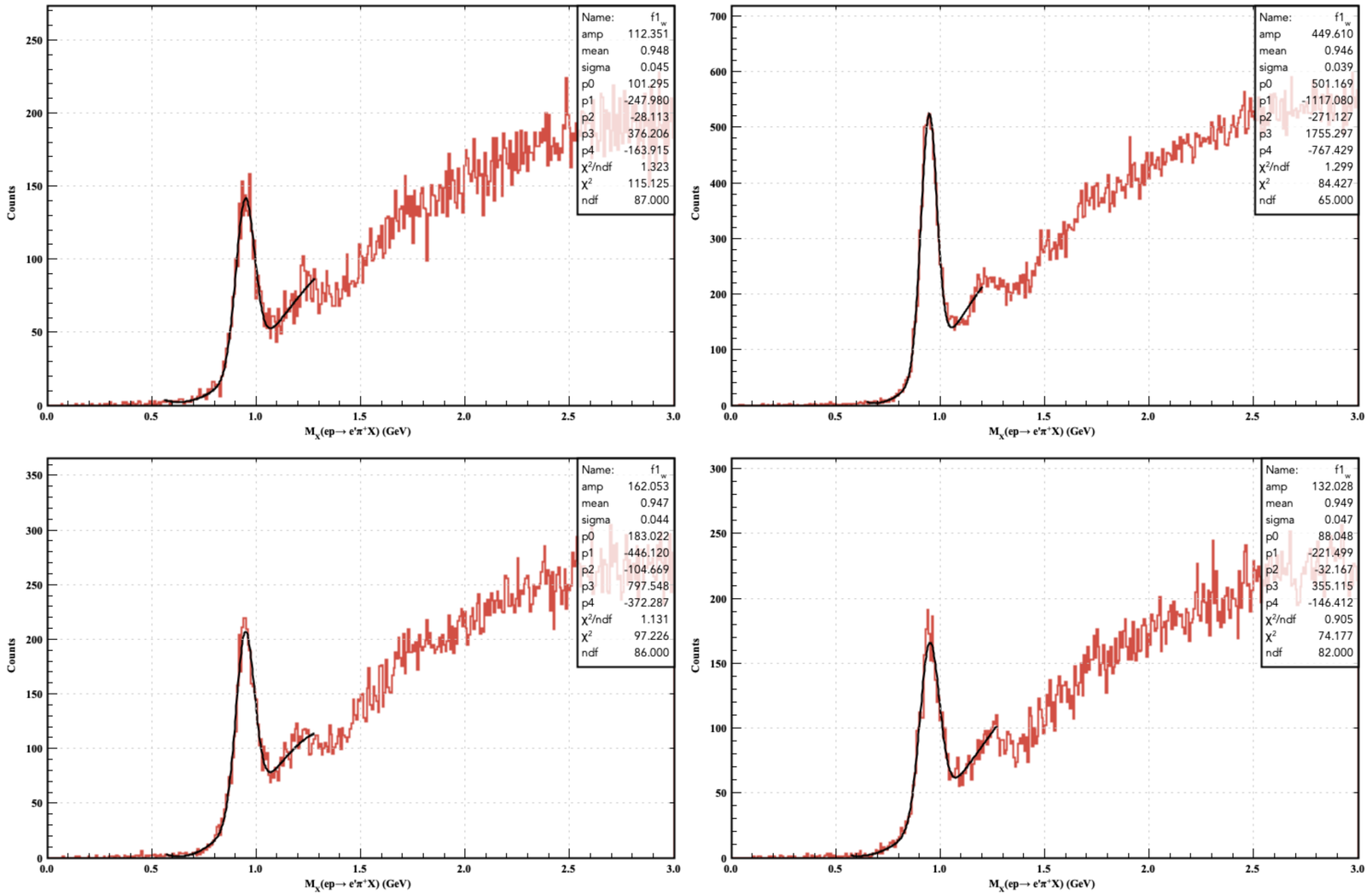
Extra Tracks by Old Model



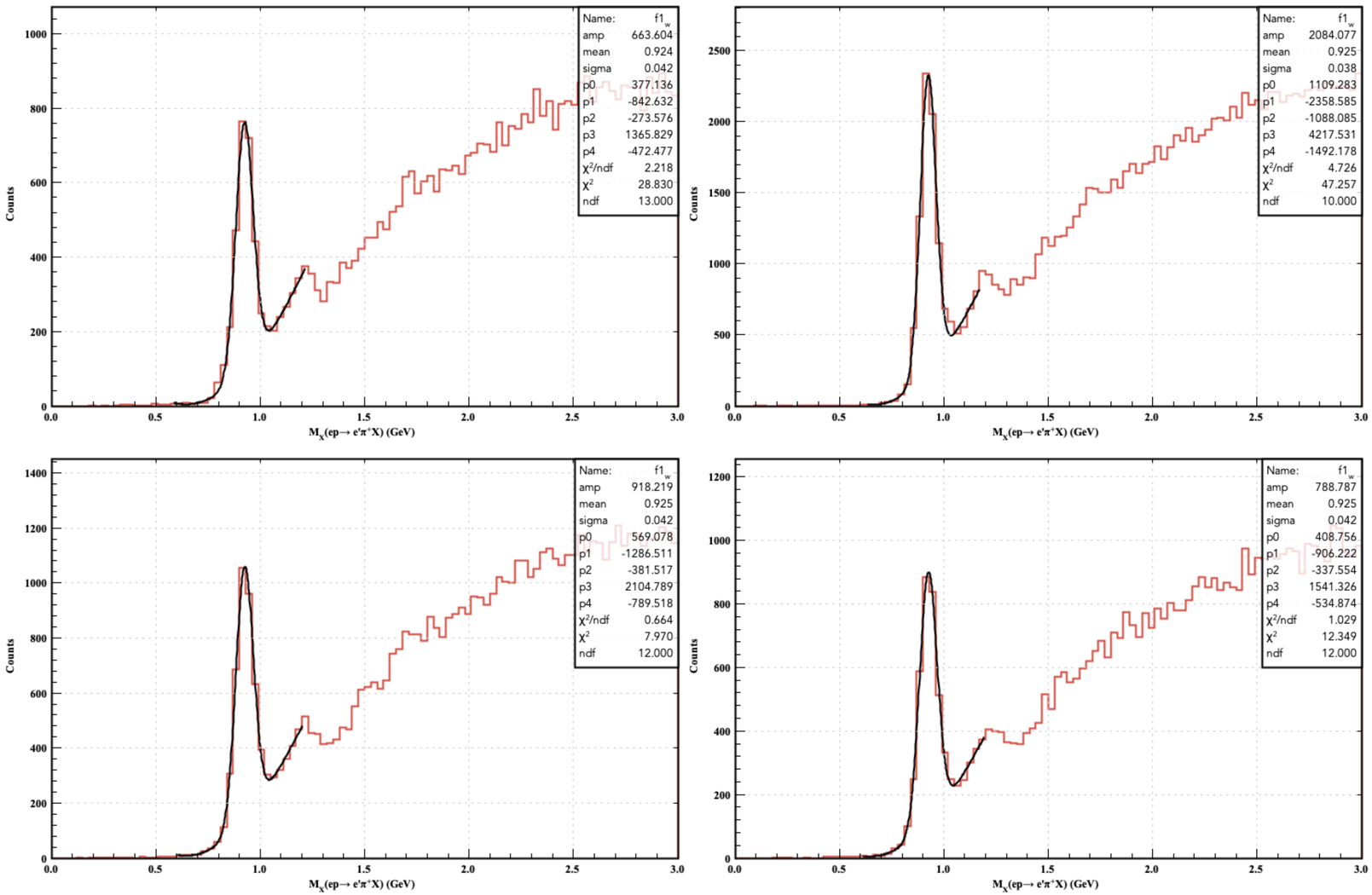
Extra Tracks by New Model



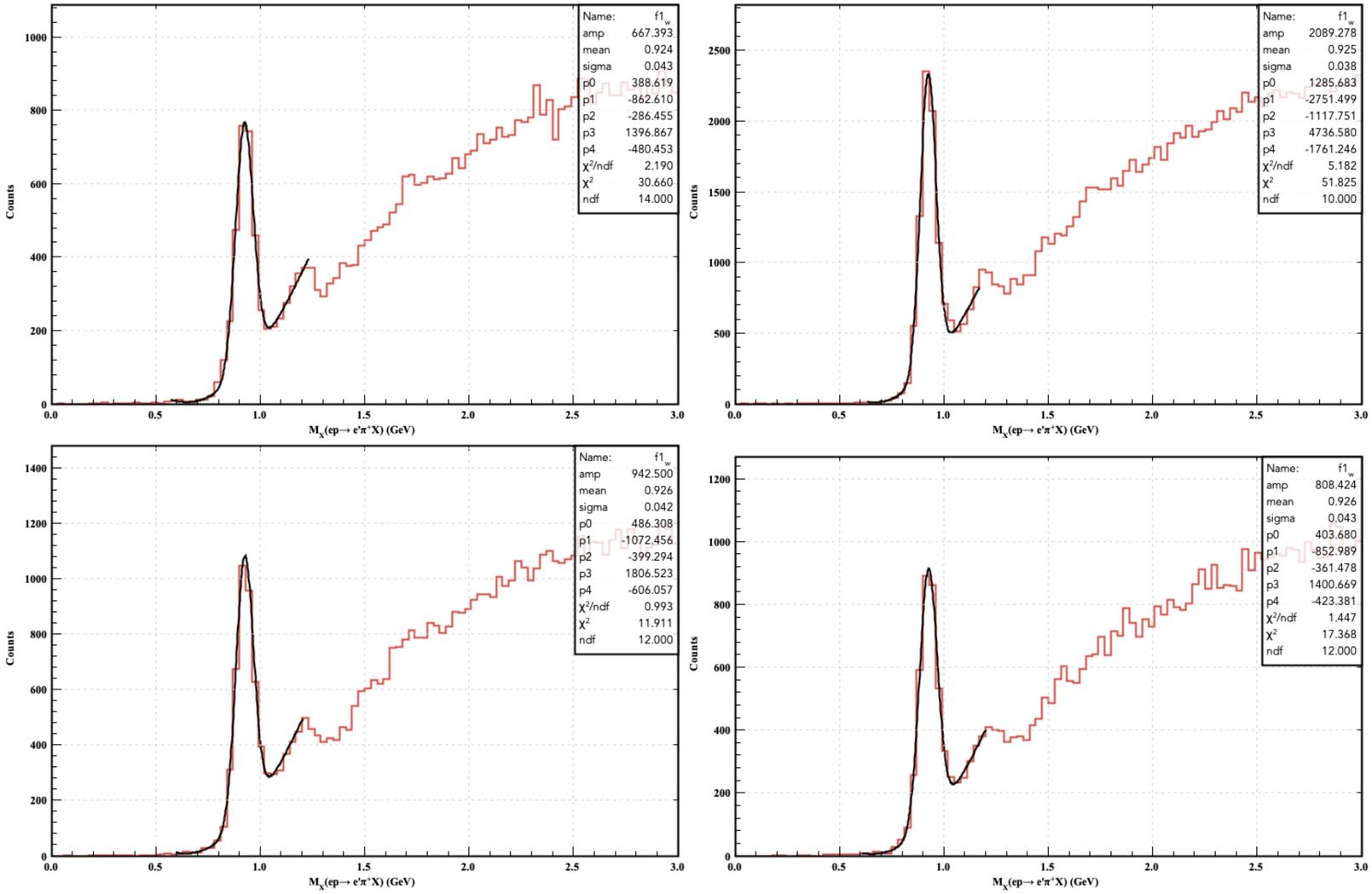
Pass2 for $ep \rightarrow e'\pi^+(n)$



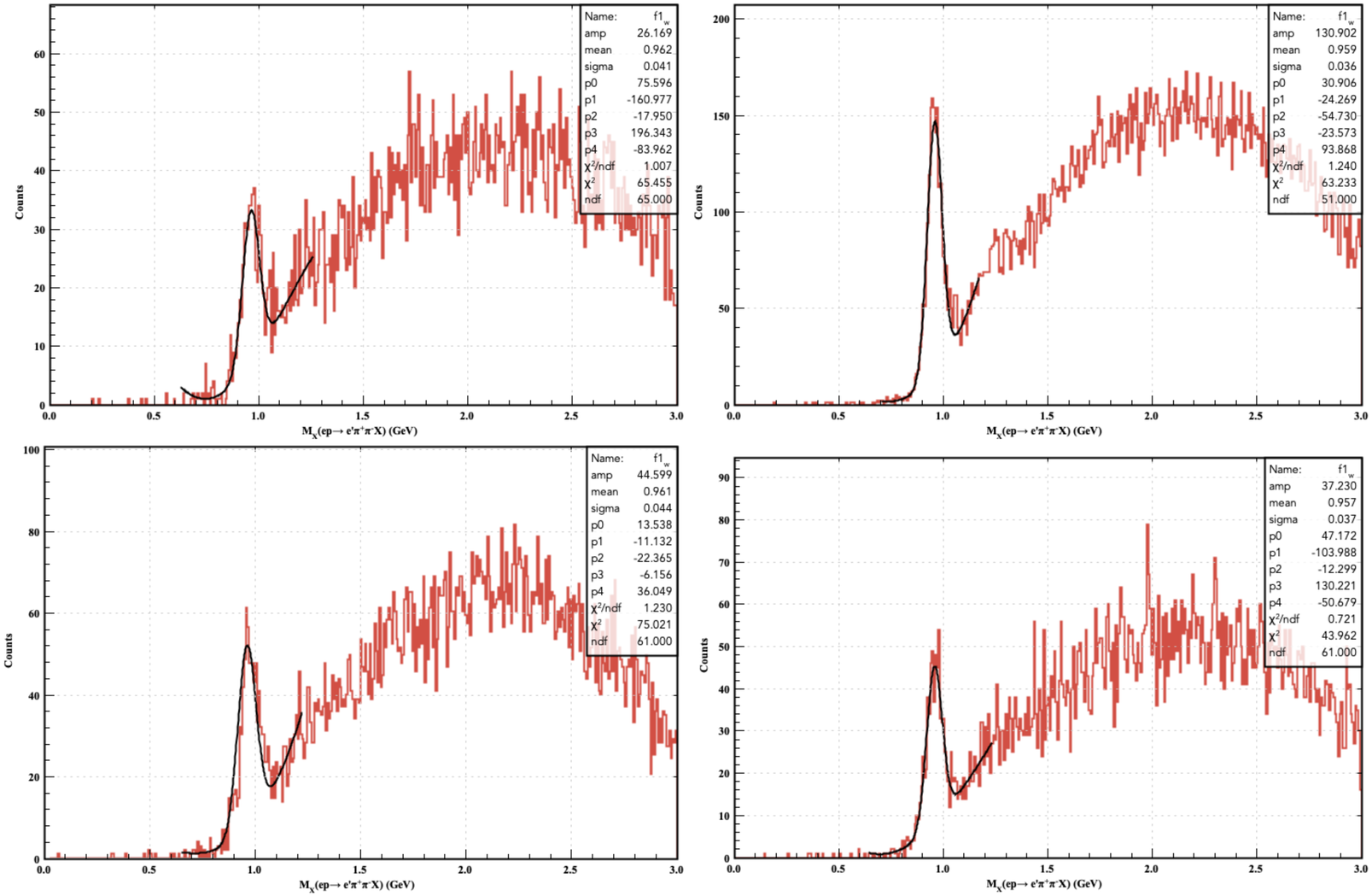
New Clustering and New Tracking for $ep \rightarrow e'\pi^+(n)$



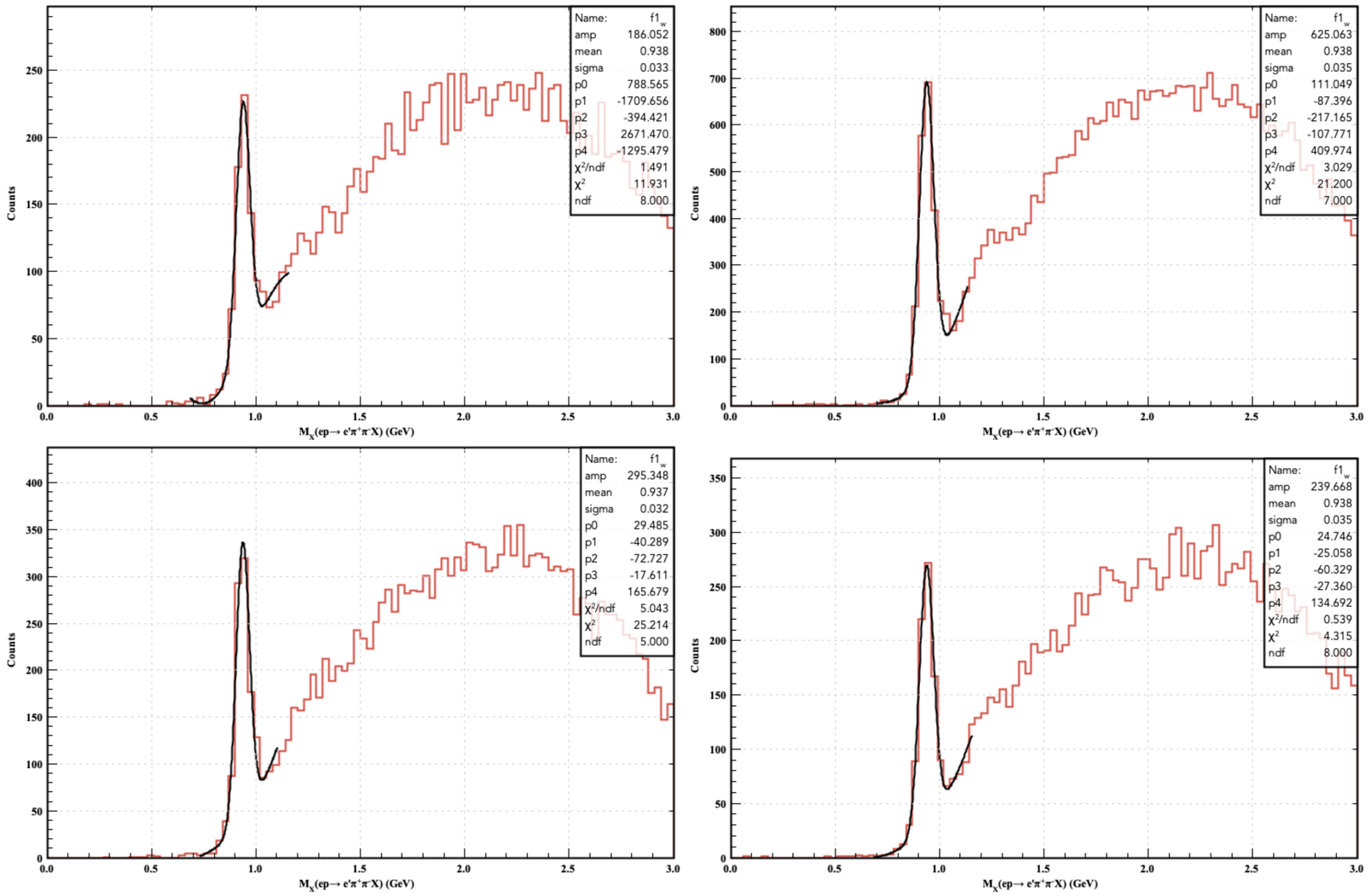
New AI Models for $ep \rightarrow e'\pi^+(n)$



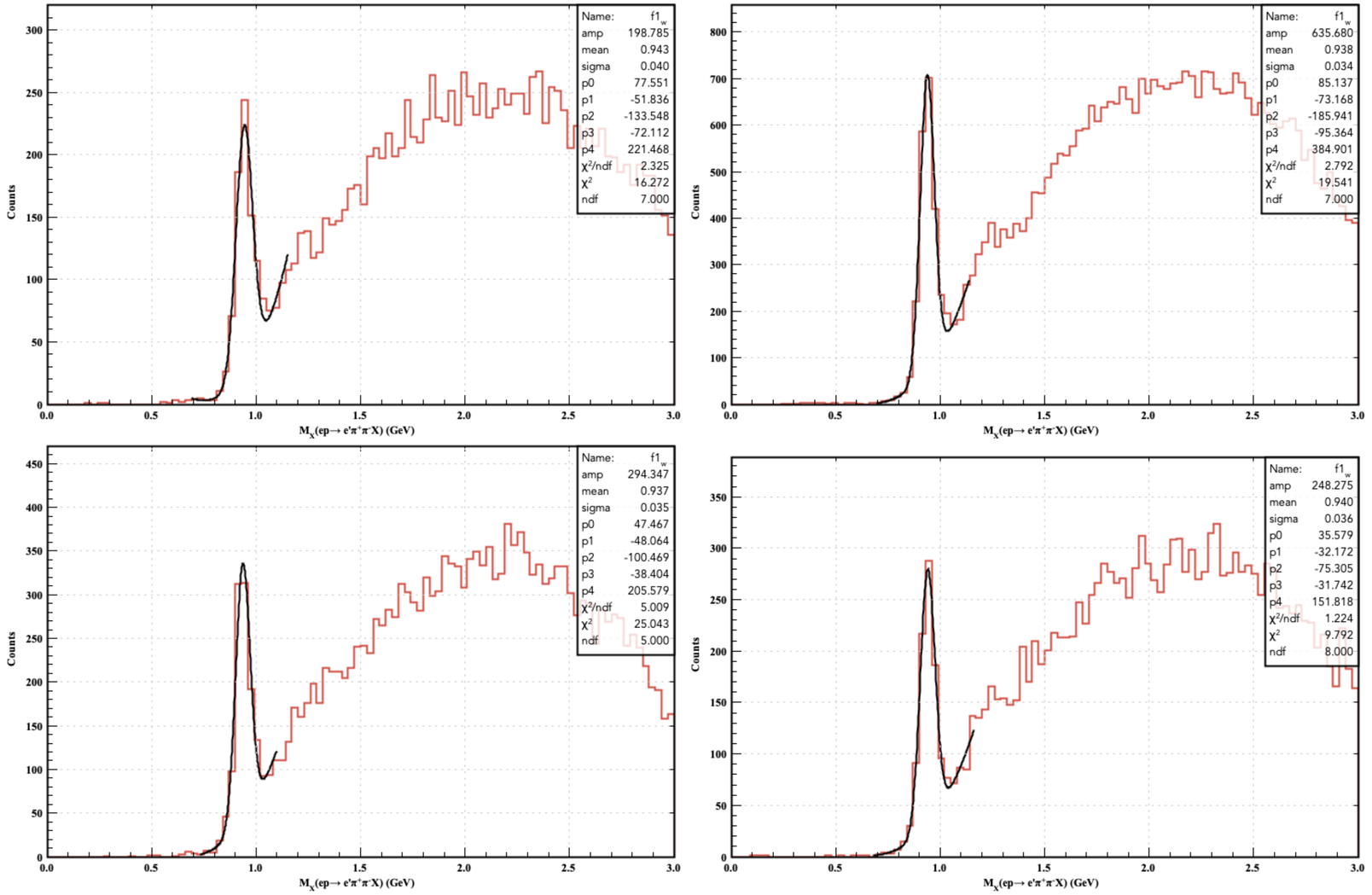
Pass2 for $ep \rightarrow e'\pi^+\pi^-(p')$



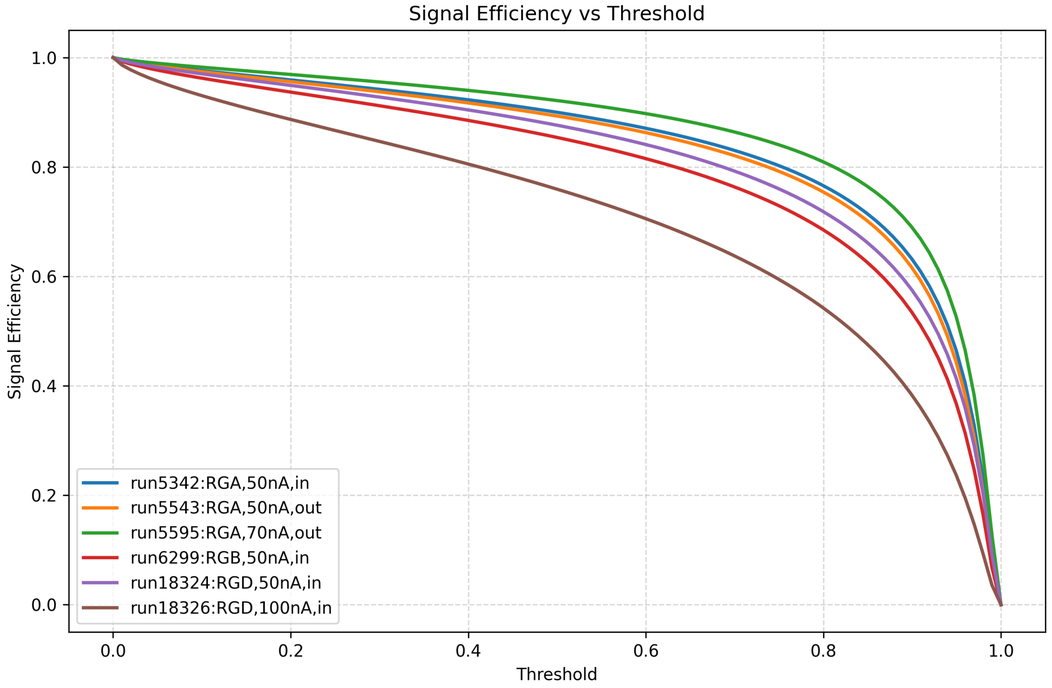
New Clustering and New Tracking for $ep \rightarrow e'\pi^+\pi^-(p')$

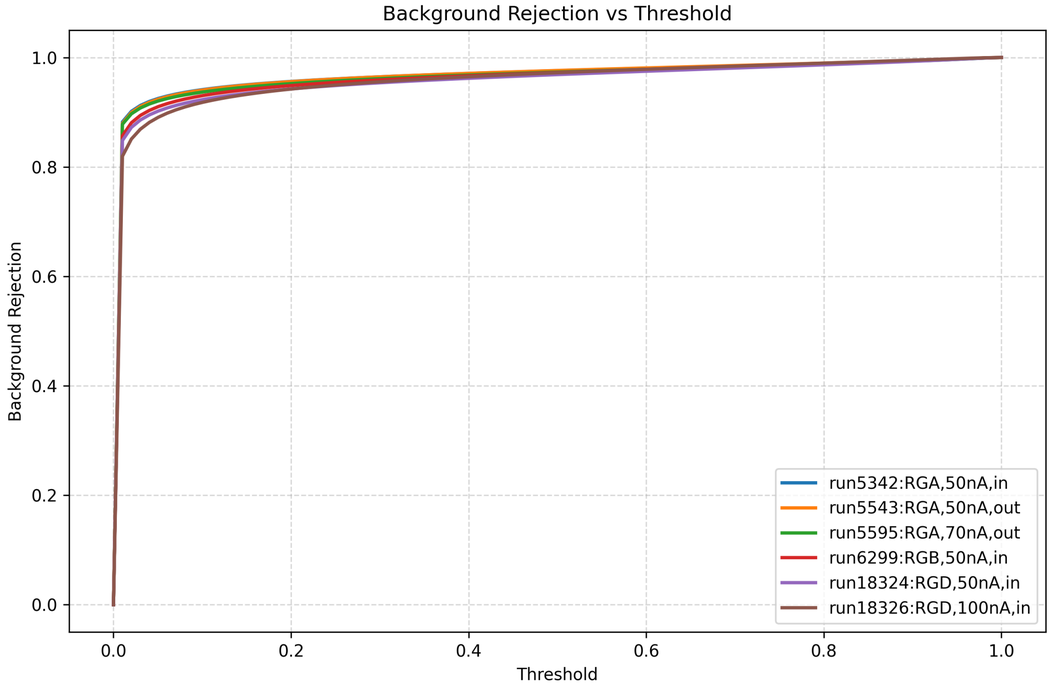


New AI Models for $ep \rightarrow e'\pi^+\pi^-(p')$



AI Denoising

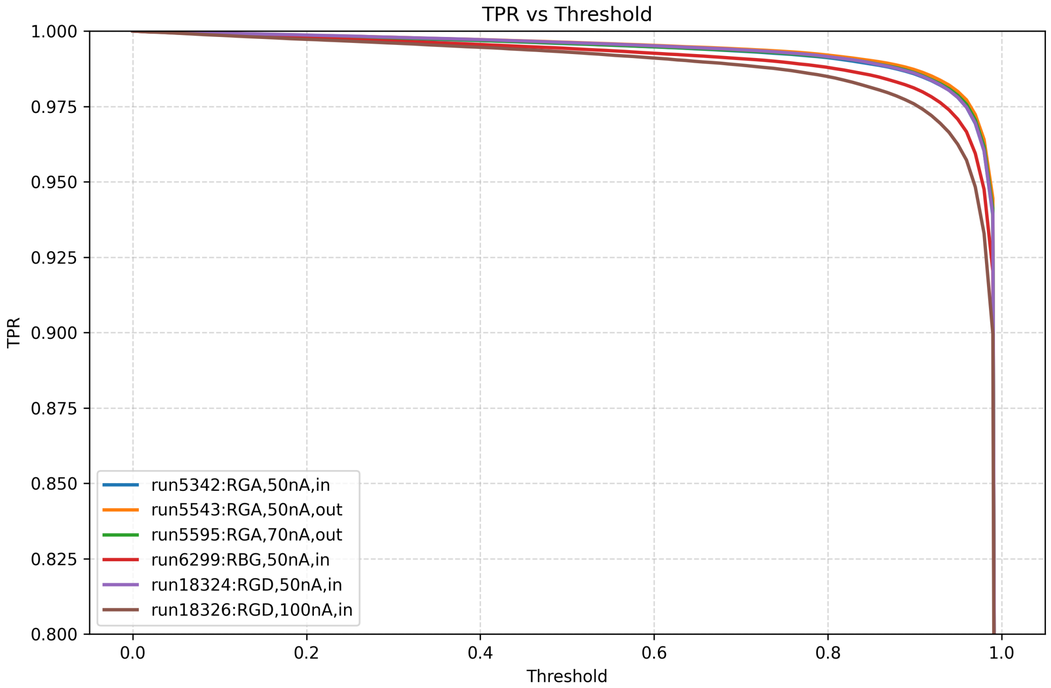


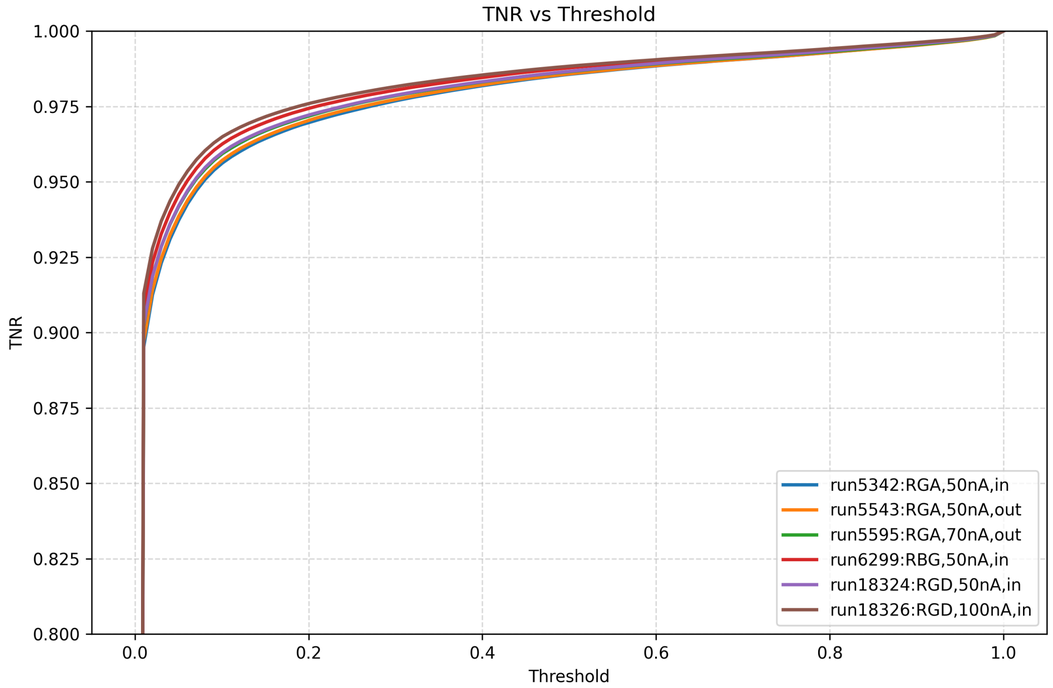


Discussion

- The model is trained using samples from RGA run 5342, an in-bending run with a 50 nA beam current.
- The background rejection performance as a function of threshold is very similar between in-bending and out-bending configurations, and is also consistent across RGA runs with different luminosities. In contrast, the performance is slightly worse for RGB and RGD runs.
- Signal efficiency vs. threshold:
  - For RGA runs with the same luminosity, the signal efficiency shows similar behavior for in-bending and out-bending configurations.
  - For RGA runs with different luminosities, the signal efficiency differs noticeably.
  - The signal efficiency is generally worse for RGB and RGD runs.
- These observations indicate that the model performance is independent of the magnetic field polarity, but shows dependence on luminosity and run group.
- The current model, with a small threshold (default value of 0.025), is available for RGA, RGB, and RGD production runs. However, for optimal performance, it is recommended to train separate models for different run groups and to validate them with dedicated performance tests.

AI Track Finding for 6-Cluster Combos

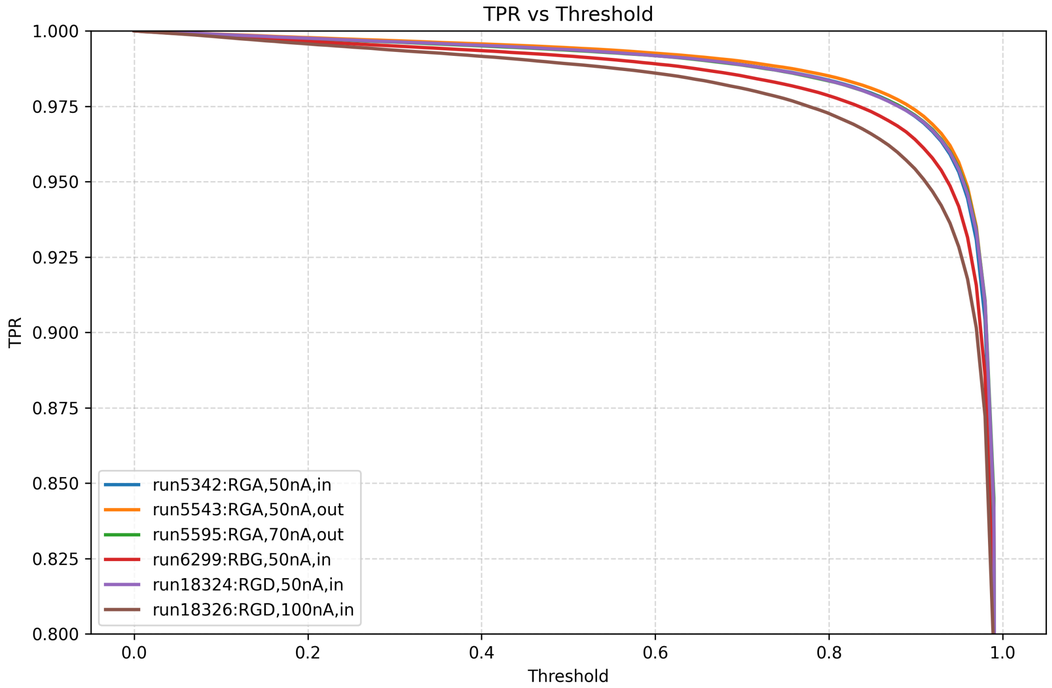


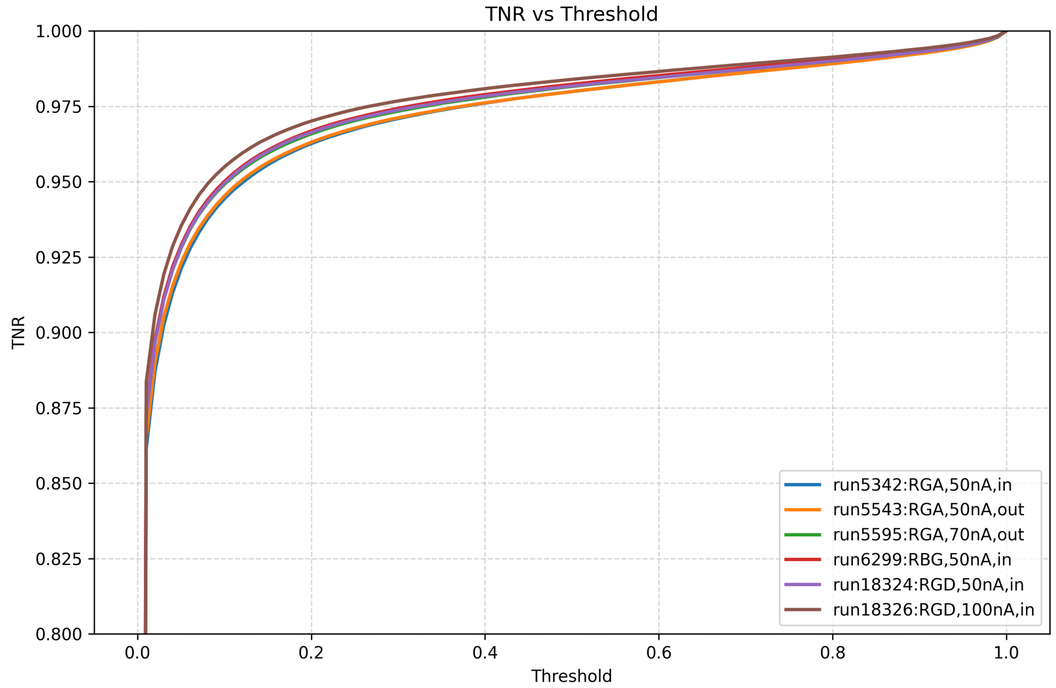


Discussion

- By default, threshold is set as 0.95.
- At such threshold, TNR has no big difference among runs.
- At such threshold, TPR is different for RGB run6299 (more background due to FT on) and RGD run18326 with 100nA beam (RGD production runs with 50nA beam).
- RGB run6299 will cut off somewhat more real combos. It does not mean that corresponding tracks will be lost. 5-cluster combos for those tracks will be predicted by AI model for 5-cluster combos. So effects are supposed to be not significant.
- It implies that we might need different model for future experiments with much higher luminosity. Also, we can test different threshold setting for RGB, or train a model with training samples from RGB.

AI Track Finding for 5-Cluster Combos

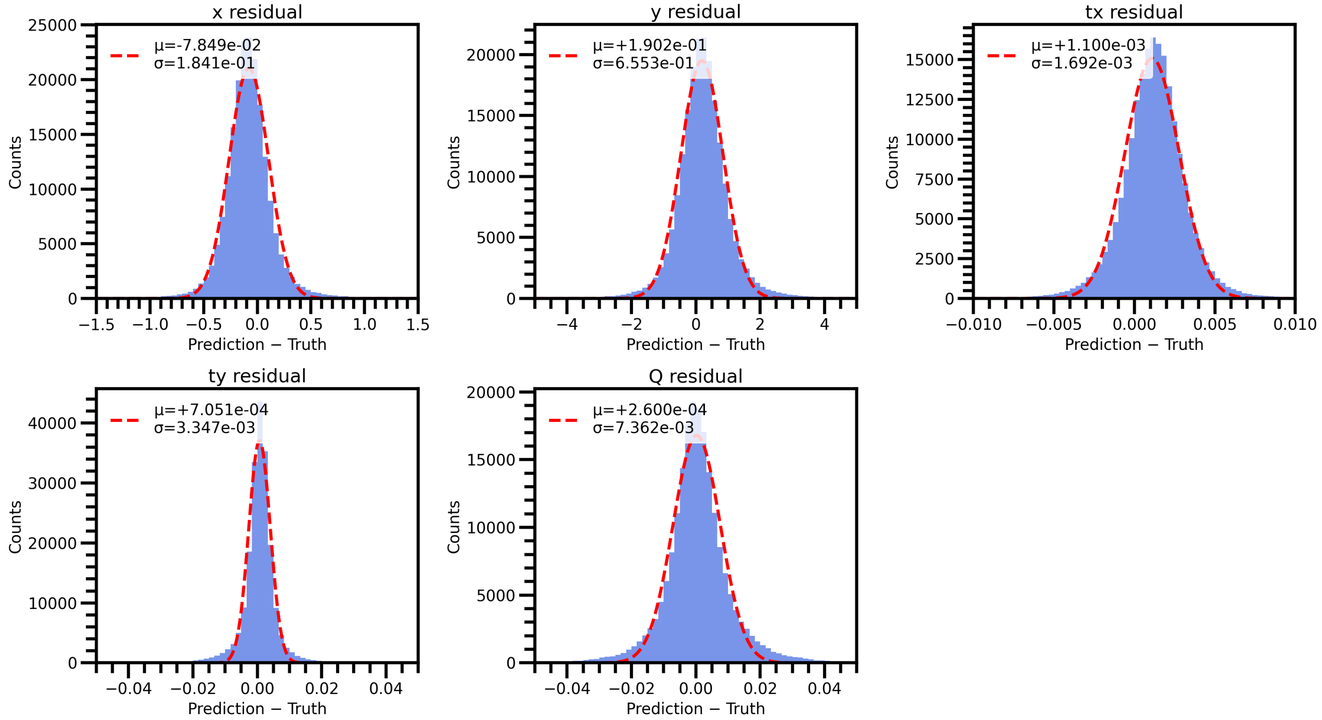


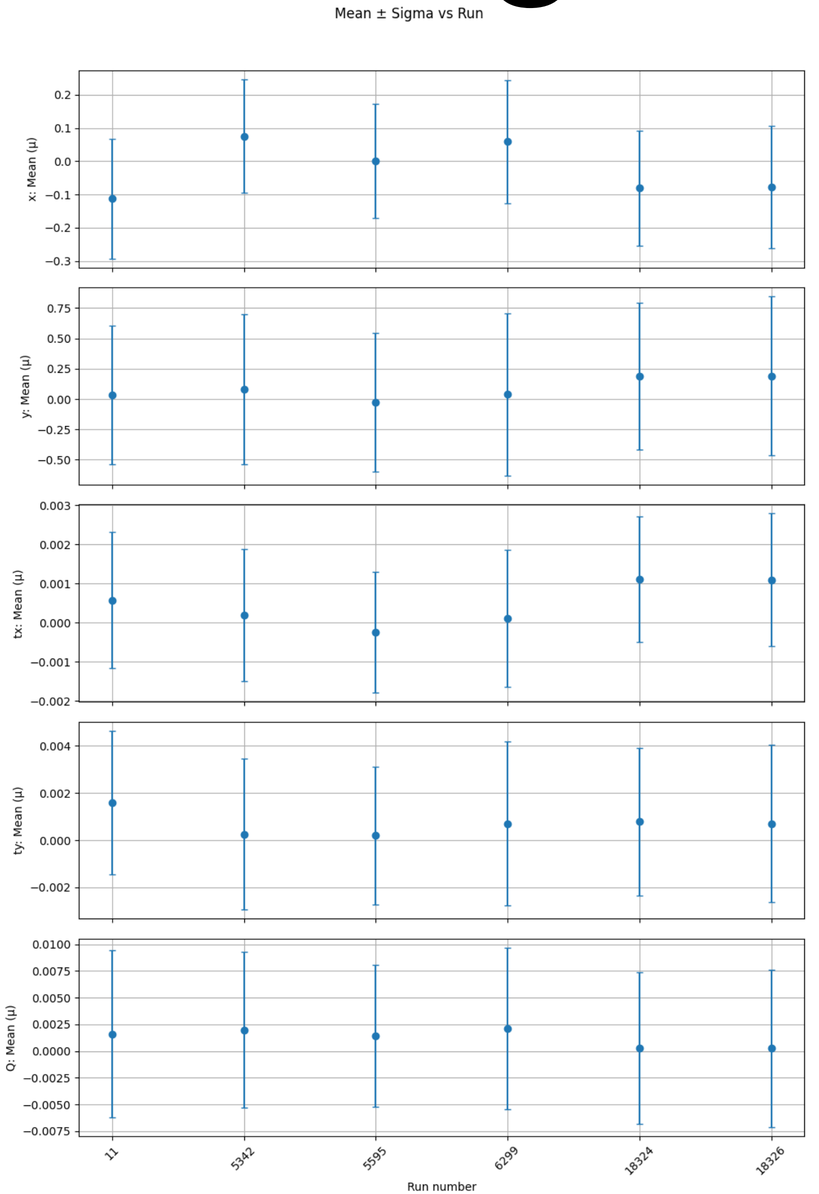


- Performance is similar for the model of 6-cluster combos.
- At default threshold of 0.05, TPR and TNR have no significant difference among runs.

AI Model for HB Tracking

Run18326 as example





Almost consistent among runs.

Proposal: AI-Driven Hit-Level Track Finding

AI Hit denoising

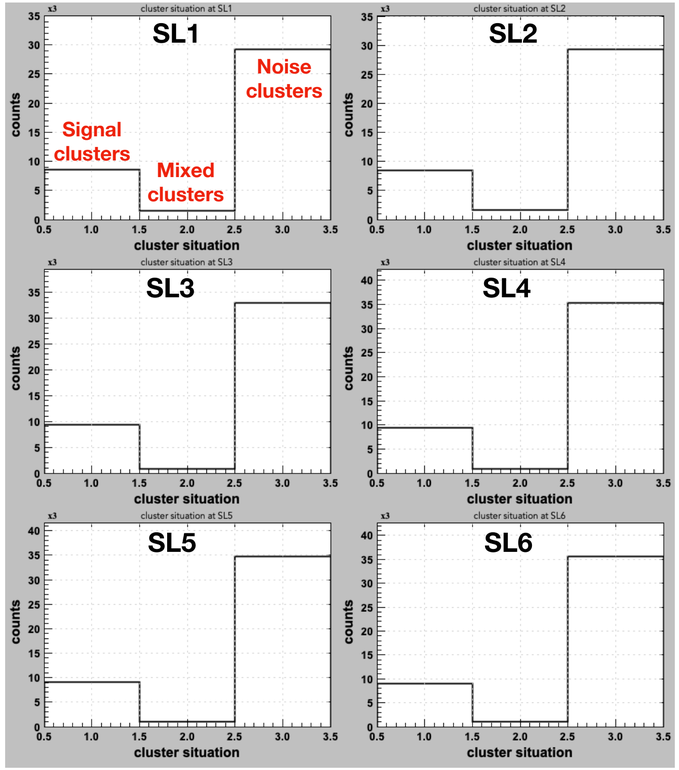
Conventional clustering

AI cluster-combo prediction for track Finding

AI Hit denoising

AI hit-combo prediction for track finding

Cluster situation in RGA MC + 100nA Bg



- In the reconstruction pipeline, clustering is the only stage still handled by traditional algorithms. Conventional clustering treats hits independently within each superlayer and does not utilize cross-superlayer correlations, resulting in increased susceptibility of reconstructed tracks to noise contamination.
- An alternative approach is to bypass clustering entirely and perform track finding directly at the hit level using AI. However, this approach faces several challenges:
  - Input complexity: The model must handle a variable number of hits per event.
  - Memory & computation: Hit-level data significantly increases resource requirements.
  - Training data: High-quality simulated events with complete truth information are essential for optimal performance.
- A graph neural network (GNN) is a promising candidate for this task.