Expanding the Scope of Track Reconstruction for the Mu2e Experiment

Matthew Stortini (Yale University)

Oct 2023 - CTD Toulouse, France

Outline

1. The Mu2e Experiment
2. Measurement Technique
3. Current Track Reconstruction
4. Shortcomings & Examples
5. Current Work and Preliminary Results
6. Outlook

The Mu2e Experiment

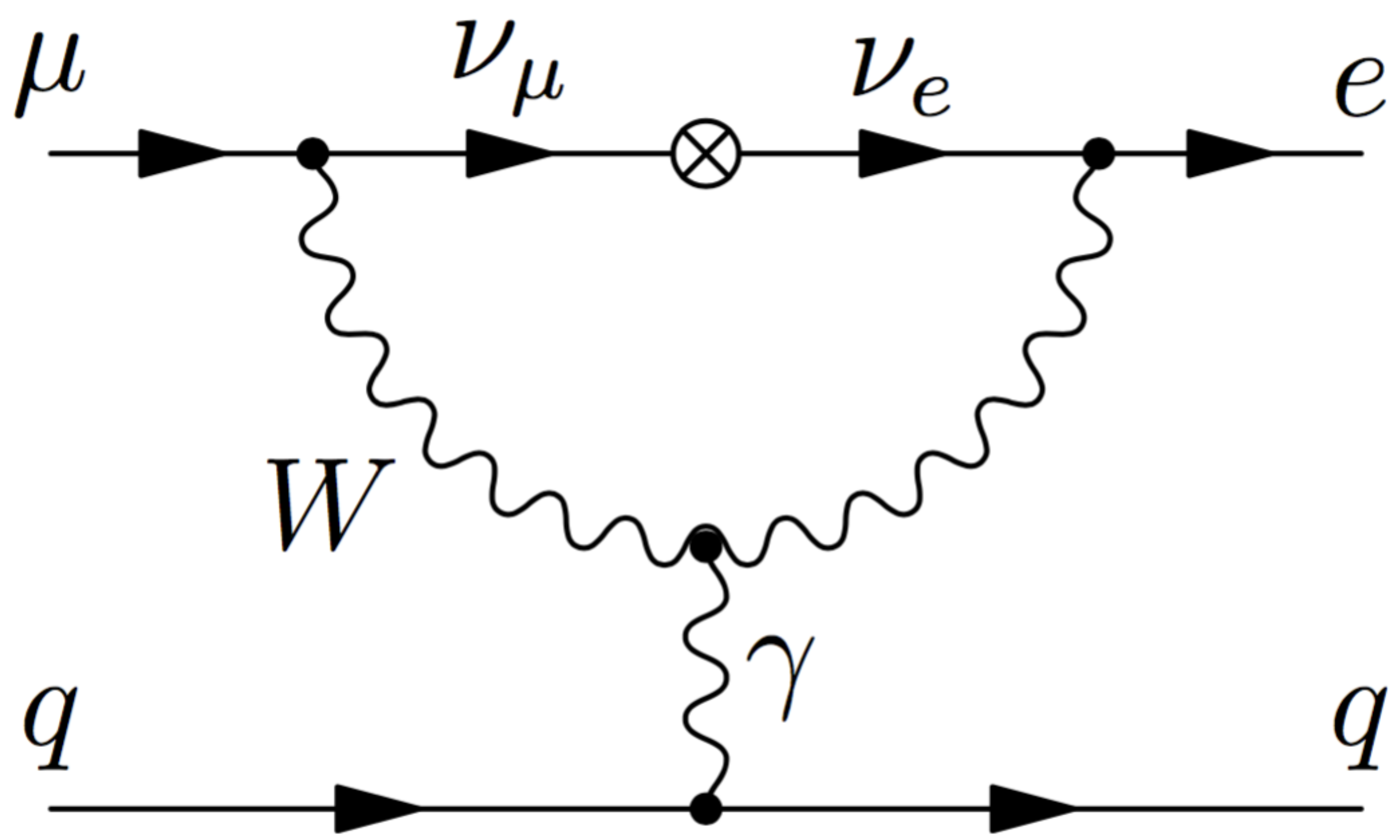
- Mu2e sets out to measure **neutrinoless muon to electron conversion** in the field of an Al-27 nucleus
  - This conversion exhibits charged-lepton flavor violation (CLFV)
- Three processes are considered for a muon orbiting an Al-27 nucleus:
  1) Nuclear Capture: $N + \mu^- \rightarrow N' + \nu_\mu + \gamma$ (~60%)
  2) Decay-In-Orbit (DIO): $N + \mu^- \rightarrow N + e^- + \overline{\nu}_e + \nu_\mu$ (~40%)
  3) **Neutrinoless DIO:** $N + \mu^- \rightarrow N + e^-$ (???)
- Mu2e aims to reach a single event sensitivity of $R_{\mu e} = 3 \times 10^{-17}$ where:

$$R_{\mu e} = \frac{\Gamma(\mu^- + N(A,Z) \rightarrow e^- + N(A,Z))}{\Gamma(\mu^- + N(A,Z) \rightarrow \nu_\mu + N(A,Z-1))}$$

- 4 orders of magnitude improvement on previous upper limit
- Under construction at the Fermilab Muon Campus in Batavia, Illinois; set to begin data taking in 2025

Motivation

- Standard Model w/ small neutrino masses allows CLFV with branching ratio $B \sim 10^{-54}$



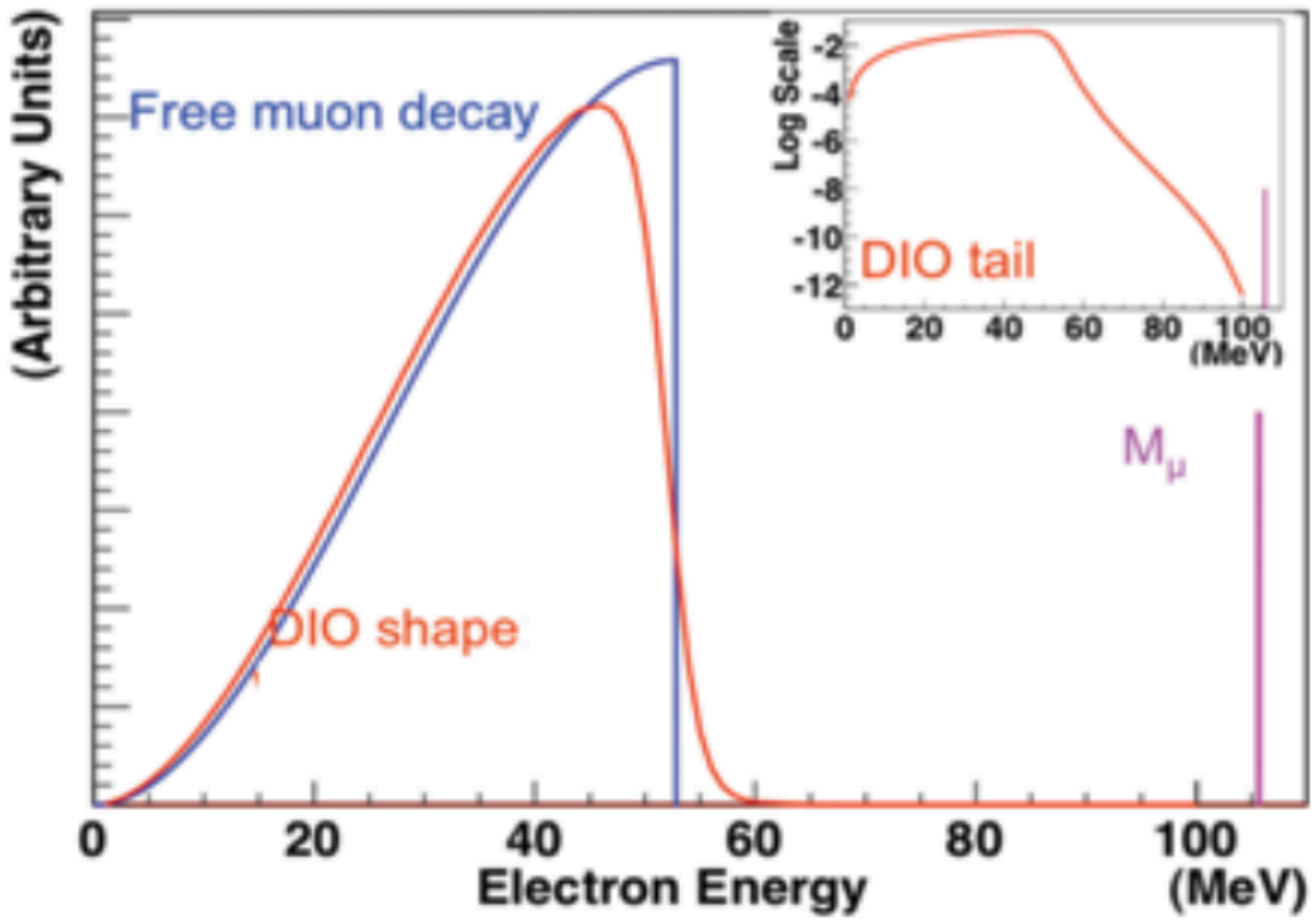
- Such a branching ratio is experimentally unobservable
    - **A measurement of neutrinoless $\mu^- \rightarrow \mathrm{e}^-$ is thus physics beyond the Standard Model (BSM)**
- Many BSM theories predict enhanced CLFV branching ratios

Outline

1. The Mu2e Experiment
2. Measurement Technique
3. Current Track Reconstruction
4. Shortcomings & Examples
5. Current Work and Preliminary Results
6. Outlook

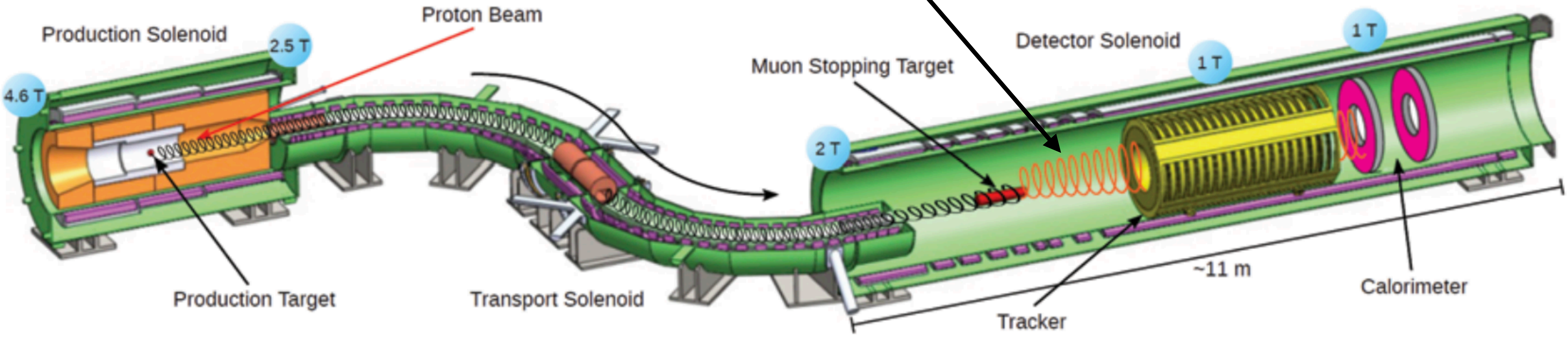
Experimental Signature

- Searching for neutrinoless $\mu^- \rightarrow e^-$ conversion
- Expect to see standard DIO e^- spectrum as a background, with endpoint energy 104.97 MeV
- **Bump at endpoint of standard DIO spectrum signals neutrinoless conversion has occurred**
- Mu2e is designed to measure momenta of e^- in a window of 80 < p < ~ few hundred MeV/c



Experimental Design

Conversion electron follows helical trajectory, making a few loops in the tracker

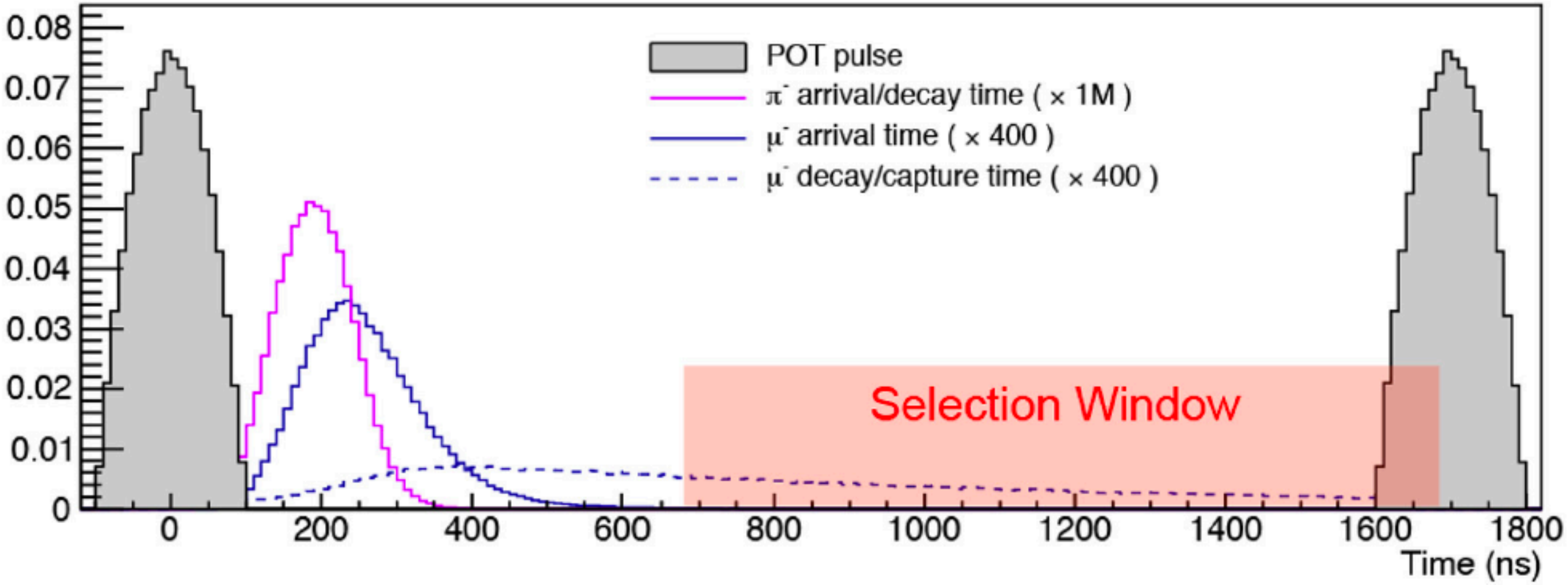


Downstream

Upstream

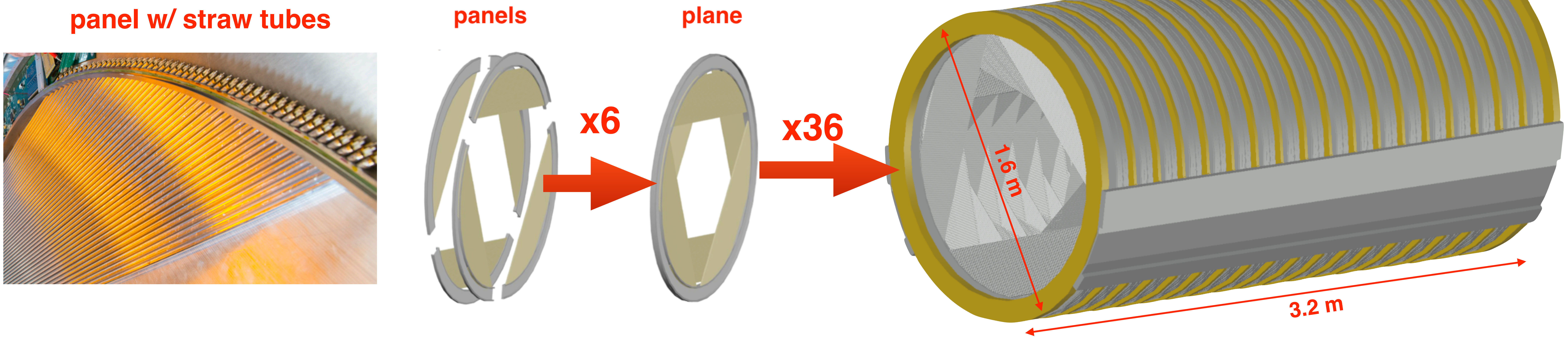
Pulsed Proton Beam

- 8 GeV proton beam with beam intensity of 3.9 x 10^7 protons/pulse, period of 1.7 μs
  - We call each pulse an "event"
- Will stop ~10^{18} muons on target over 3 years of running ($\approx 3.6 \times 10^{20}$ protons on target)
  - ~0.005 muons/proton-on-target reach aluminum stopping target, ~40% of those are stopped



Tracker Design

- Tracker made up of panels, which consist of straw tubes
- Inside straw tube is a wire, and gas (80:20 Argon:CO_2 at operating voltage of 1,500 V)
  - Electron passing through straw tube ionizes gas
  - Ionization charge is collected by wire, hardware reads out current, data is produced
  - Based on the straw tubes that are hit, we can **reconstruct electron track**



Outline

1. The Mu2e Experiment
2. Measurement Technique
3. Current Track Reconstruction
4. Shortcomings & Examples
5. Current Work and Preliminary Results
6. Outlook

General Structure

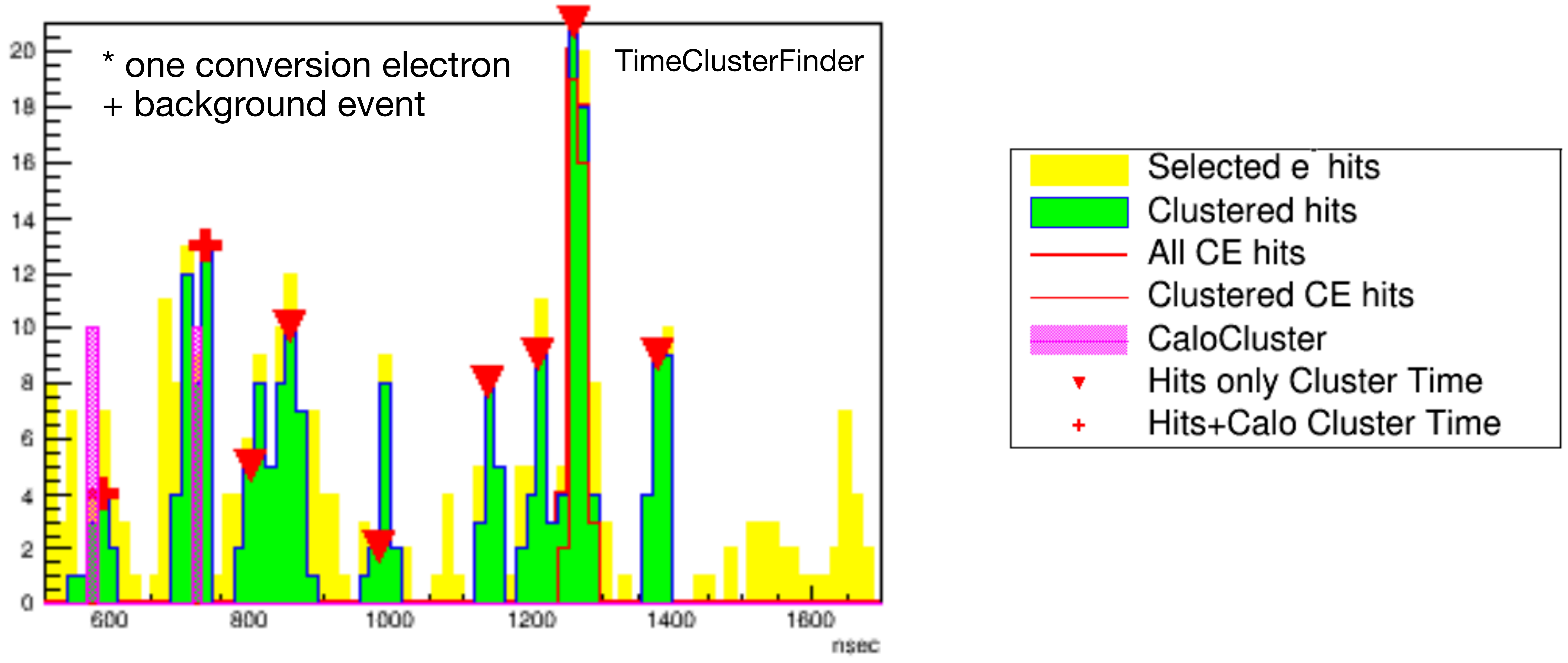
- Software written within the Fermilab developed event-processing framework, *art*
    - Write modules and create sequences, then call sequences and configure modules at run time using FHiCL files
- Mu2e track reconstruction follows the general structure:

- Mu2e track reconstruction algorithms are broken into two main types:
    - Tracker seeded algorithms (**TPR**) — this type always processes tracker data
    - Calorimeter seeded algorithms (**CPR**) — needs a calorimeter cluster to perform reconstruction

<table>
<tr><th></th><th>Hit Preparation</th><th>Bkg Flagging</th><th>Time Clustering</th><th>Pattern Recognition</th><th>Track Reco</th></tr>
<tr><td>CPR</td><td rowspan="2">PrepareHits</td><td rowspan="2">FlagBkgHits</td><td>CalTimePeakFinder</td><td>CalHelixFinder</td><td rowspan="2">KinKal</td></tr>
<tr><td>TPR</td><td>TimeClusterFinder</td><td>RobustHelixFinder</td></tr>
</table>

Time Clustering

- **TimeClusterFinder** works by histogramming hit times and searching for peaks
  - Time-of-flight correction used to map hit times from a particle to the time it was at z=0 (center of tracker)
    - Time-of-flight correction makes peak more visible
    - Requires assuming particle's pitch angle and velocity (assumption tuned to conversion electron)
- **CalTimePeakFinder** uses a time window (with time-of-flight correction) around calorimeter cluster to make time clusters



Pattern Recognition

- Exact logic for these modules is provided in the backup slides
- Like TimeClusterFinder, pattern recognition modules make assumptions tuned to conversion electrons:
  - **RobustHelixFinder**
    - Applies min/max radius cut (p_t constraint)
    - Applies min/max $d\phi/dz$ cut (pitch constraint)
  - **CalHelixFinder**
    - Assumes track comes from stopping target and leaves calorimeter cluster
    - Applies min radius cut
    - Applies min/max $d\phi/dz$ cut

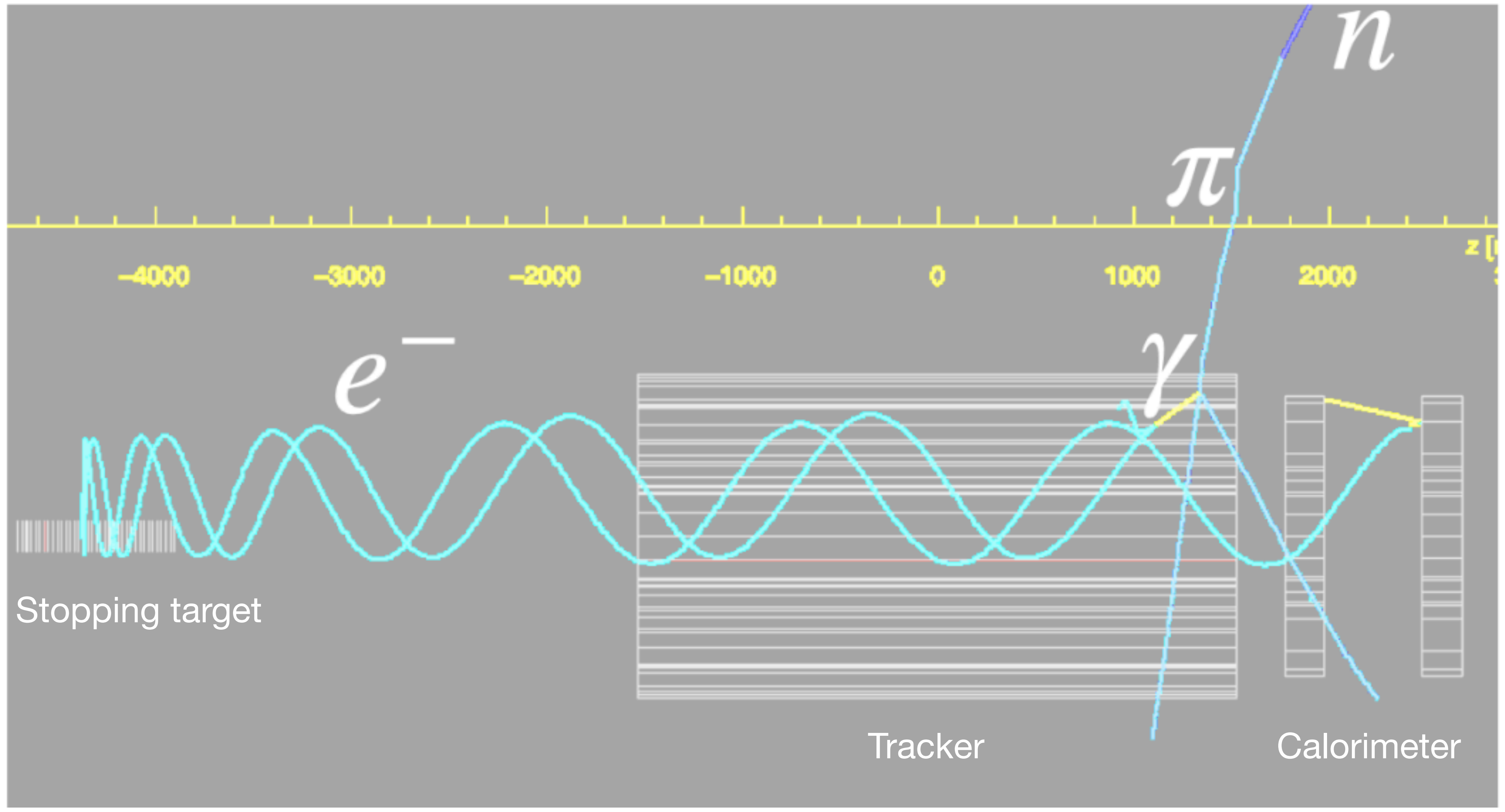
Outline

Shortcomings

- Assumptions specified lead to a track reconstruction highly tuned to $\mu^- \rightarrow e^-$:
  - Single track, downstream, near endpoint energy, coming from stopping target
- Struggles with other topologies
  - Low efficiency reconstructing upstream tracks, and non-electron tracks
  - Not configured to reconstruct more than one track per time cluster

Example 1: Upstream e^-

- Cosmics missed by cosmic ray veto can induce upstream moving e^-
  - e^- can "bounce" in the magnetic field gradient and move back downstream, looking like $\mu^- \to e^-$
    - Can reduce this background if able to reconstruct upstream track and reject event
- Energy losses in detector material lead to different upstream and downstream momenta
  - Can be used to calibrate low-energy tail of momentum resolution function



Projected Backgrounds for Run I

Channel	Mu2e Run I
SES	2.4×10^{-16}
Cosmic rays	0.046 ± 0.010 (stat) ± 0.009 (syst)
DIO	0.038 ± 0.002 (stat) $^{+0.025}_{-0.015}$ (syst)
Antiprotons	0.010 ± 0.003 (stat) ± 0.010 (syst)
RPC in-time	0.010 ± 0.002 (stat) $^{+0.001}_{-0.003}$ (syst)
RPC out-of-time ($\zeta = 10^{-10}$)	$(1.2 \pm 0.1$ (stat) $^{+0.1}_{-0.3}$ (syst)$) \times 10^{-3}$
RMC	$< 2.4 \times 10^{-3}$
Decays in flight	$< 2 \times 10^{-3}$
Beam electrons	$< 1 \times 10^{-3}$
Total	0.105 ± 0.032

Example 2: $p\bar{p}$ Annihilations

- Antiprotons 3rd largest background for Mu2e (after cosmic rays and DIO)
- Background has large systematic uncertainties due to uncertainty on the production cross section
- $p\bar{p}$ annihilations can produce pion and muon pairs in the tracker, a signature with a small background
- We can measure the antiproton background and better constrain it if we can:
  - Reconstruct multiple tracks per time cluster
  - Better reconstruct muons and pions

Projected Backgrounds for Run I

Channel	Mu2e Run I
SES	2.4×10^{-16}
Cosmic rays	0.046 ± 0.010 (stat) ± 0.009 (syst)
DIO	0.038 ± 0.002 (stat) $^{+0.025}_{-0.015}$ (syst)
Antiprotons	0.010 ± 0.003 (stat) ± 0.010 (syst)
RPC in-time	0.010 ± 0.002 (stat) $^{+0.001}_{-0.003}$ (syst)
RPC out-of-time ($\zeta = 10^{-10}$)	$(1.2 \pm 0.1$ (stat) $^{+0.1}_{-0.3}$ (syst)$) \times 10^{-3}$
RMC	$< 2.4 \times 10^{-3}$
Decays in flight	$< 2 \times 10^{-3}$
Beam electrons	$< 1 \times 10^{-3}$
Total	0.105 ± 0.032

Example 3: Proton Counting

- Nuclear capture creates unstable nucleus which then fragments, releasing protons
- Counting these protons per event in real time and saving the number has two uses:
  - Can be used to compute the number of muons stopped on target
  - Can be used to provide feedback to the accelerator people in real time
- Energy deposited in straw by proton much higher than electrons of interest
  - No need for full reconstruction to count protons, just time clustering
  - Only have a few MeV of kinetic energy … usually die quickly in tracker
- Time-of-flight correction in TimeClusterFinder leaves it unable to cluster protons

Outline

1. The Mu2e Experiment
2. Measurement Technique
3. Current Track Reconstruction
4. Shortcomings & Examples
5. Current Work and Preliminary Results
6. Outlook

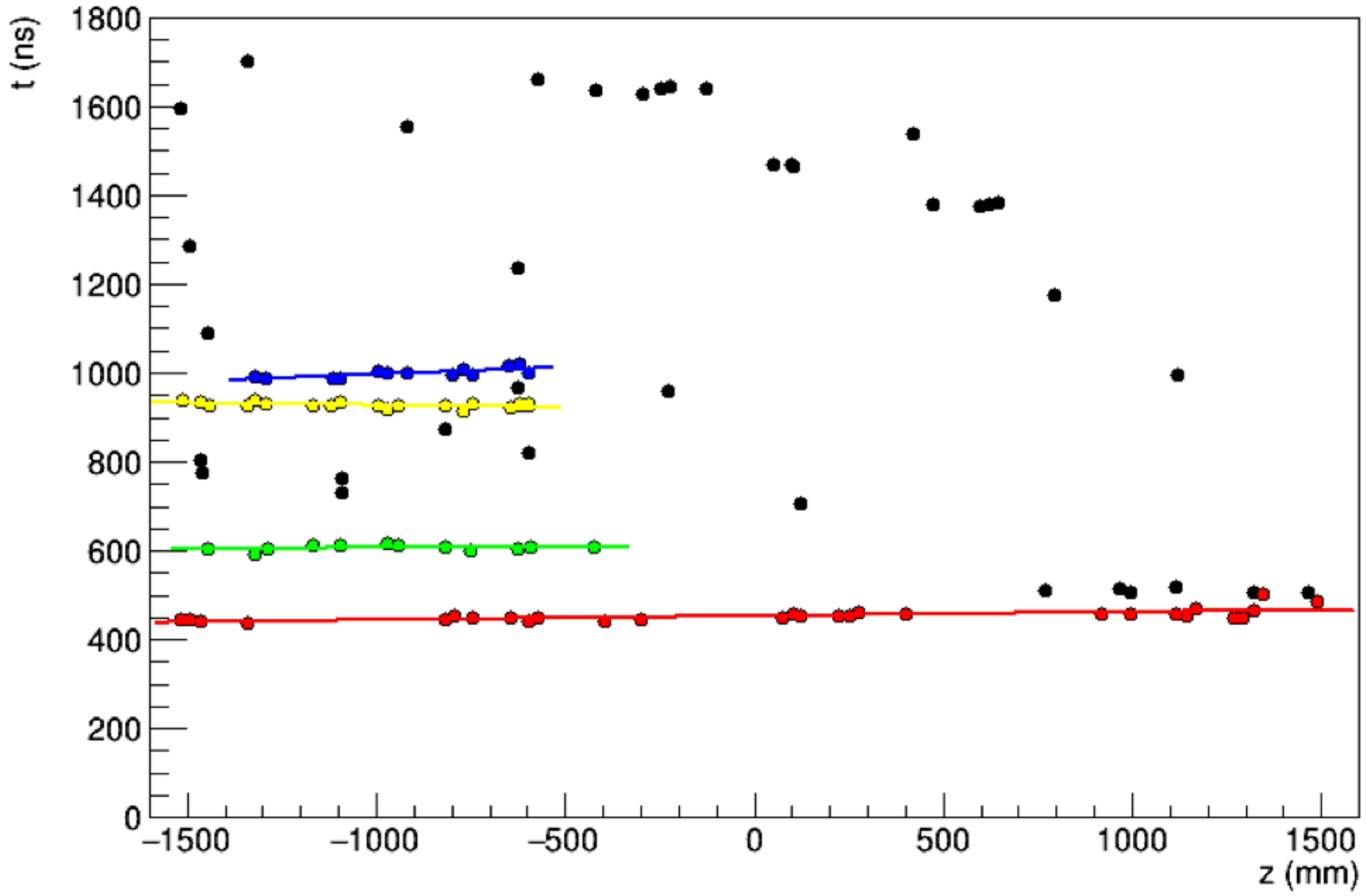
New Direction

- Can we create new time clustering and pattern recognition modules that are less finely tuned, without sacrificing timing performance or conversion electron efficiency?

- Developed new time clustering module, TZClusterFinder, and have started developing new pattern recognition module, HelixFinder

TZClusterFinder Logic

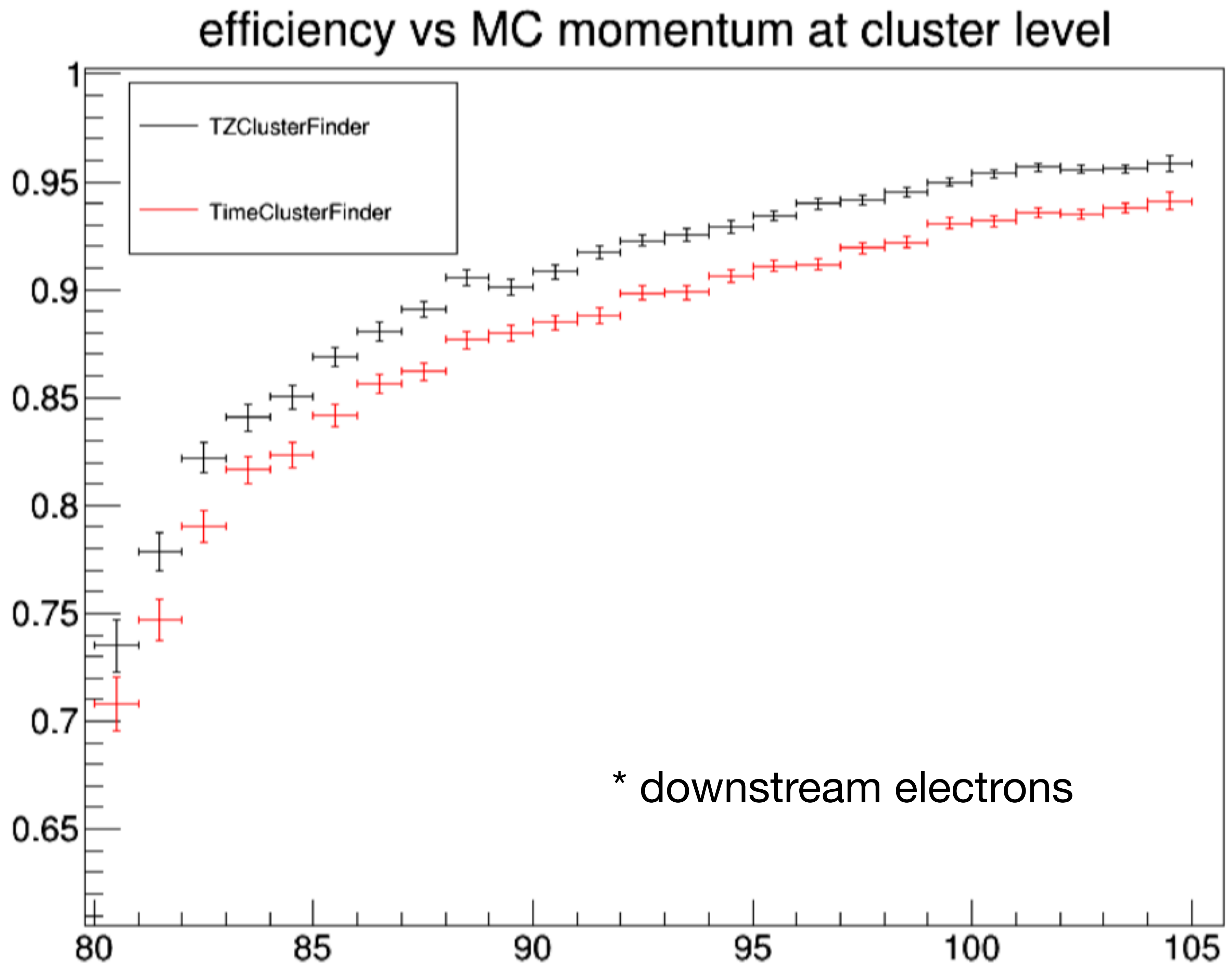
- Moves away from the histogramming method of TimeClusterFinder
- Searches for linear lines in t vs. z space
- No time-of-flight corrections or assumptions about particle direction

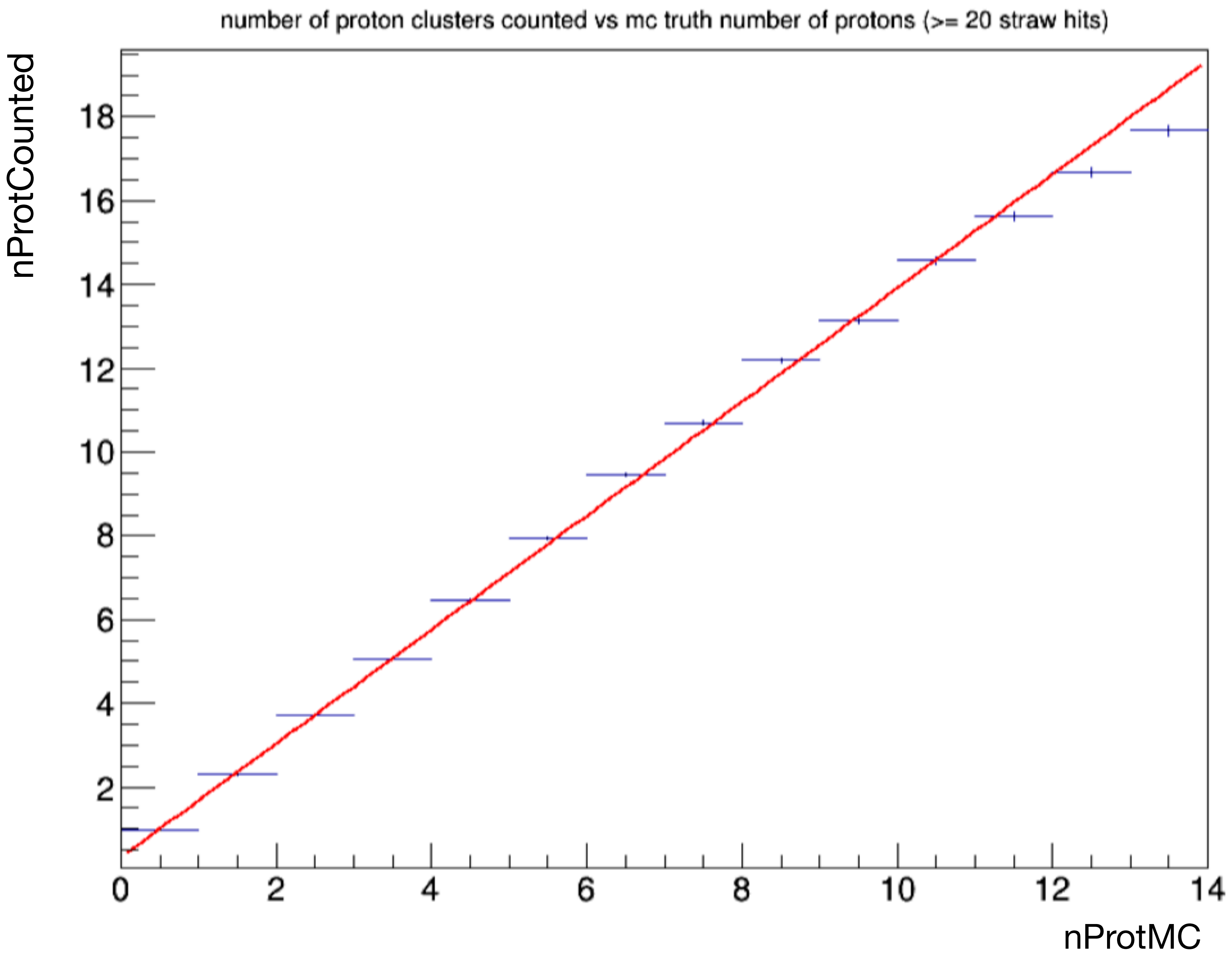


- Black points are non-clustered hits
- Every other color corresponds to a cluster found
  - Blue cluster is a proton
  - Red cluster is conversion electron

TZClusterFinder Performance

- More efficiently clusters all particles/topologies considered
- Faster than TimeClusterFinder : ~0.3 ms/event vs. ~0.5 ms/event
- Can count protons … linearity respected quite well in plot below up until ~15 MC truth protons (which is ~3σ past the mean number of MC truth protons)



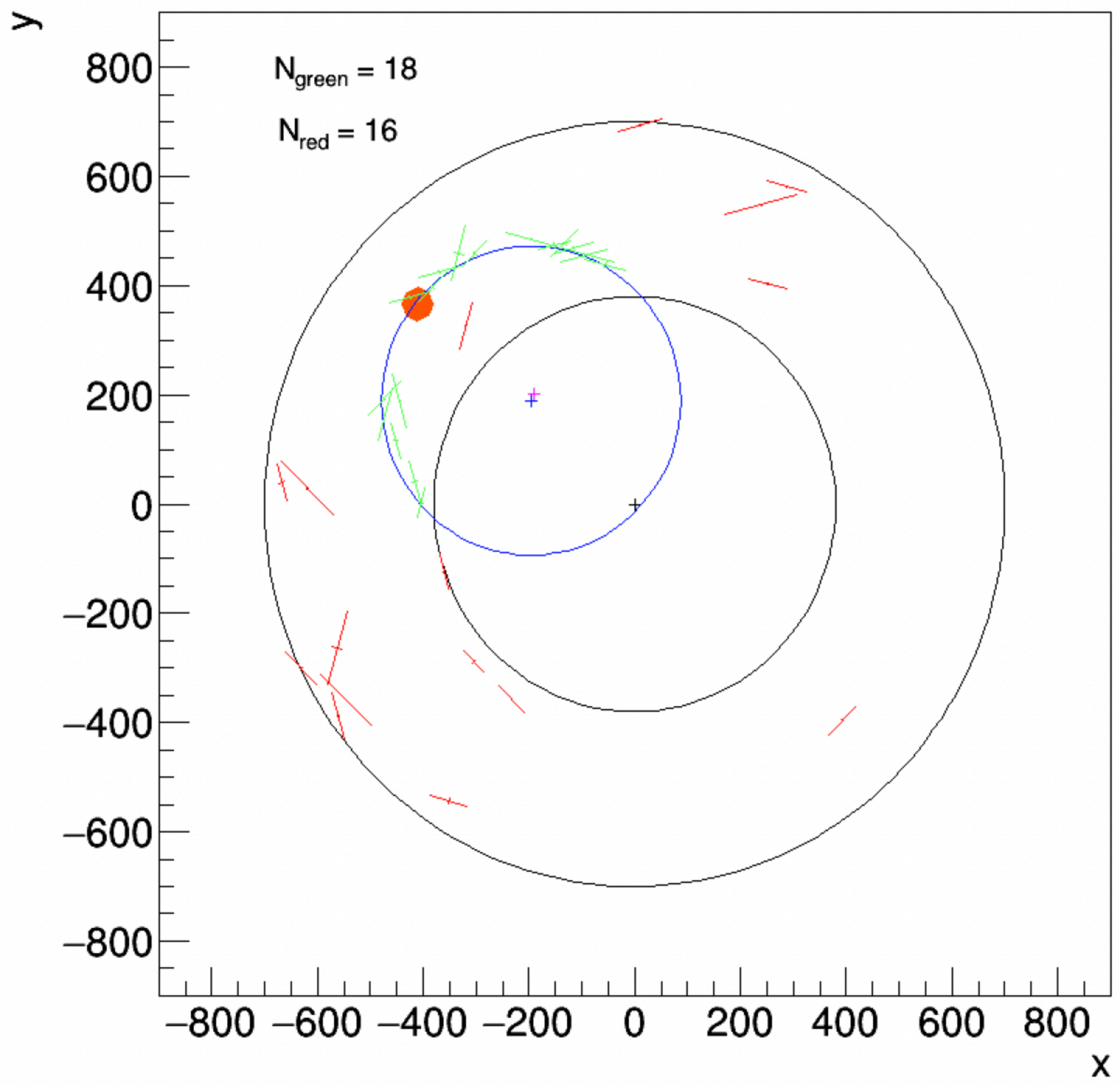


HelixFinder Logic

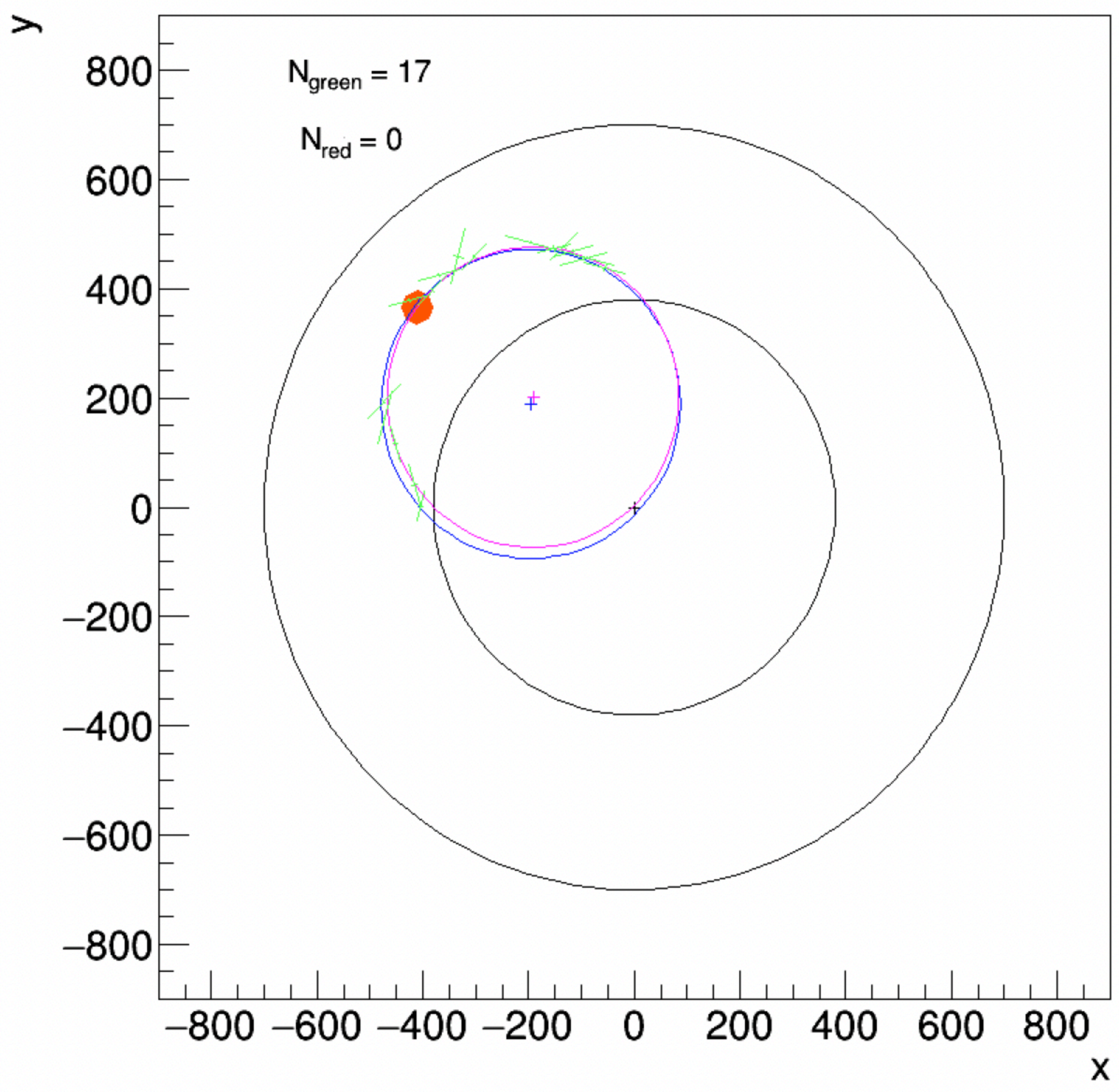
- Operating on a single time cluster, HelixFinder follows the logic:
    1. Flag hits that are “bad” for seeding search (flagging methods not fully developed)
    2. Project hits onto XY plane, and find peaks in phi
    3. Fit a given phi peak (loops over all peaks found) to a circle
    4. Remove the hit with the largest residual and update the circle
    5. Repeat step 4 until all hits are within 1 sigma of the circle

HelixFinder Logic

XY view of entire time cluster



XY view after step 5 of logic



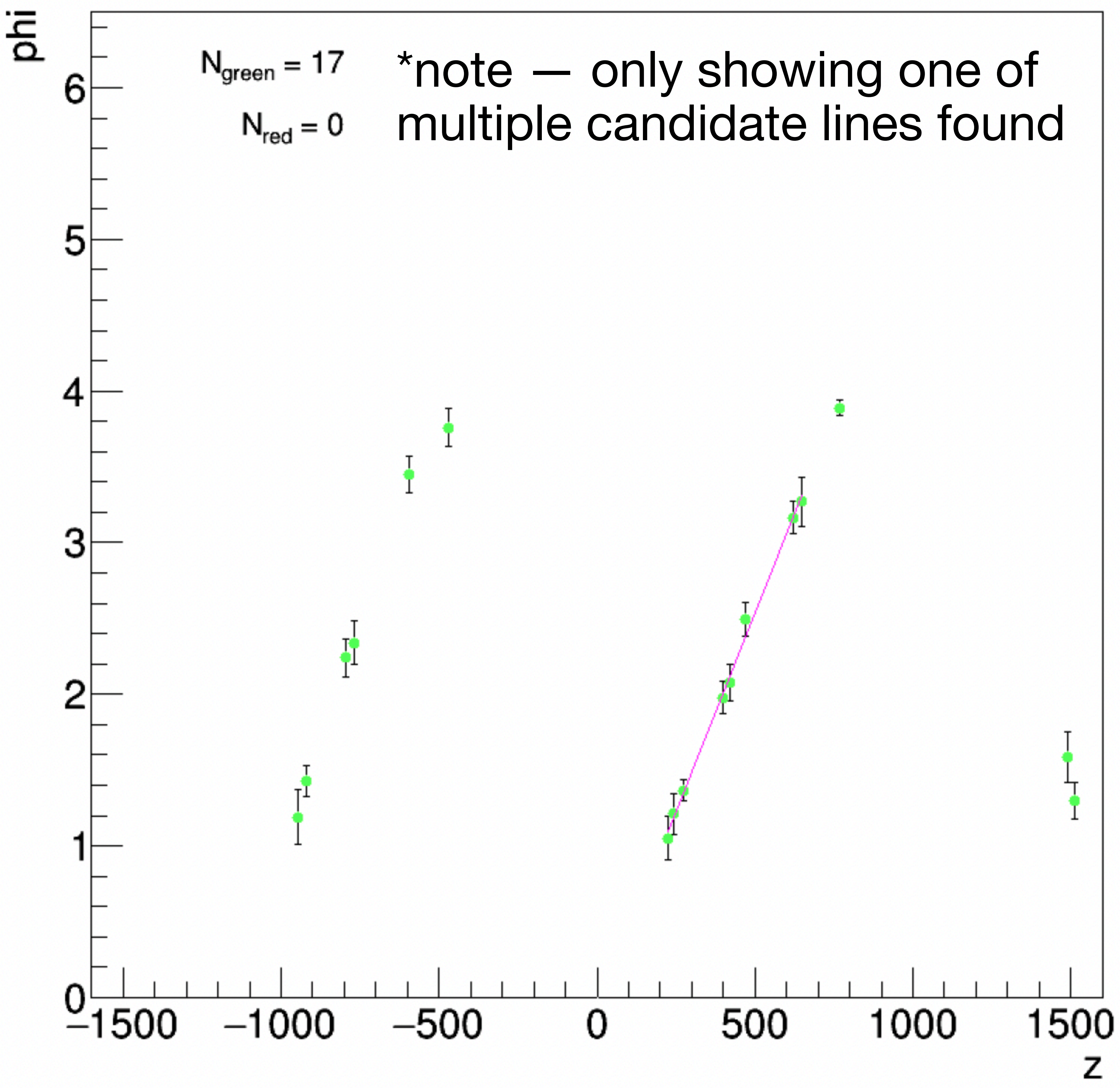
- Black circles = tracker outline
- Green lines = conversion electron straw hits
- Red lines = background straw hits
- Orange solid circle = calorimeter cluster
- Blue circle = MC helix circle
- Pink circle = fit helix circle

HelixFinder Logic

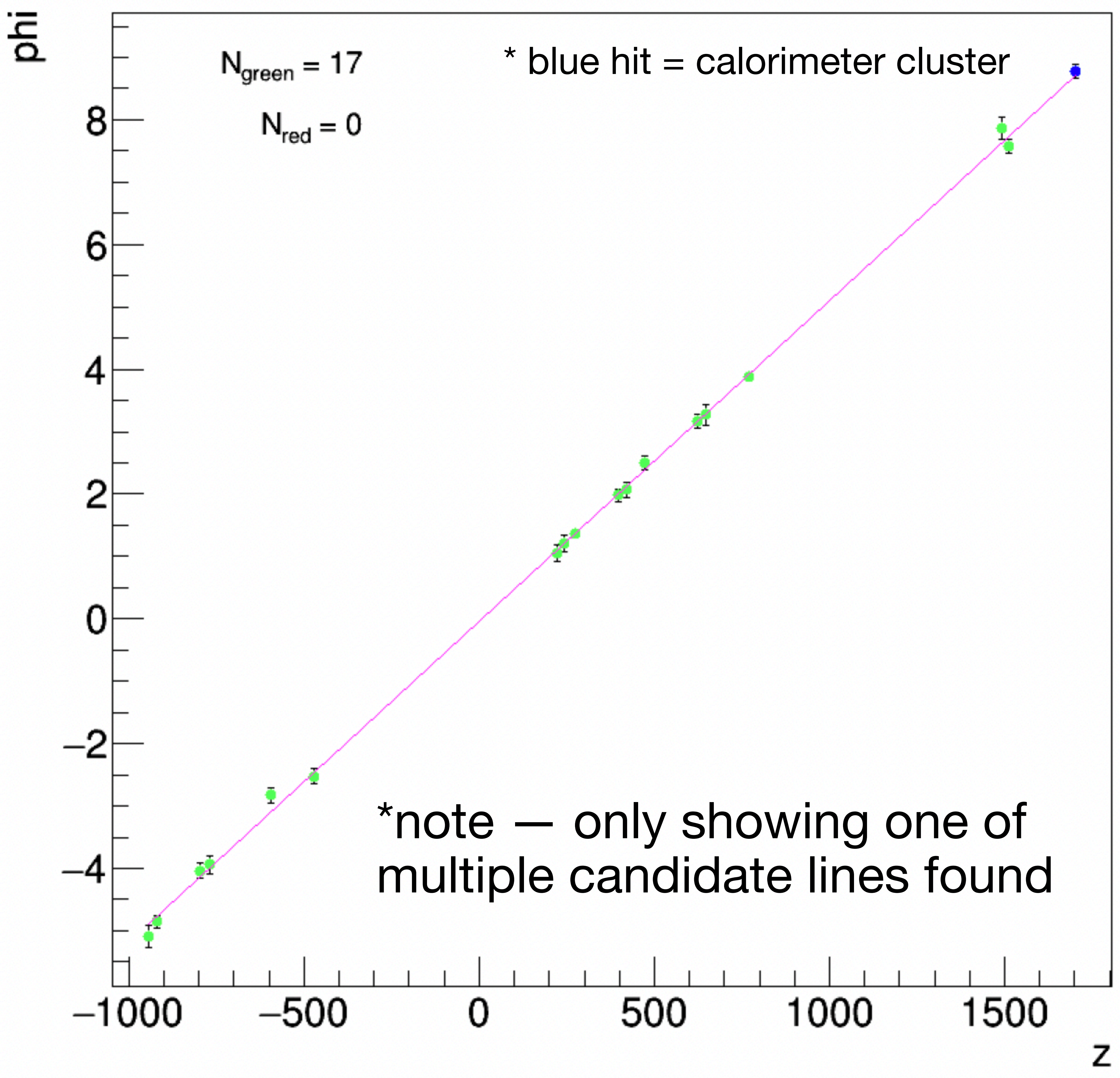
6. Take hits from circle to compute phi of each hit relative to circle center
7. Plot phi vs. z and use a rolling window to search for linear lines
8. For each line found, resolve 2π ambiguity for each hit and compute residuals, only keeping hits within some residual threshold

HelixFinder Logic

Phi-z view after step 7



Phi-z view after step 8

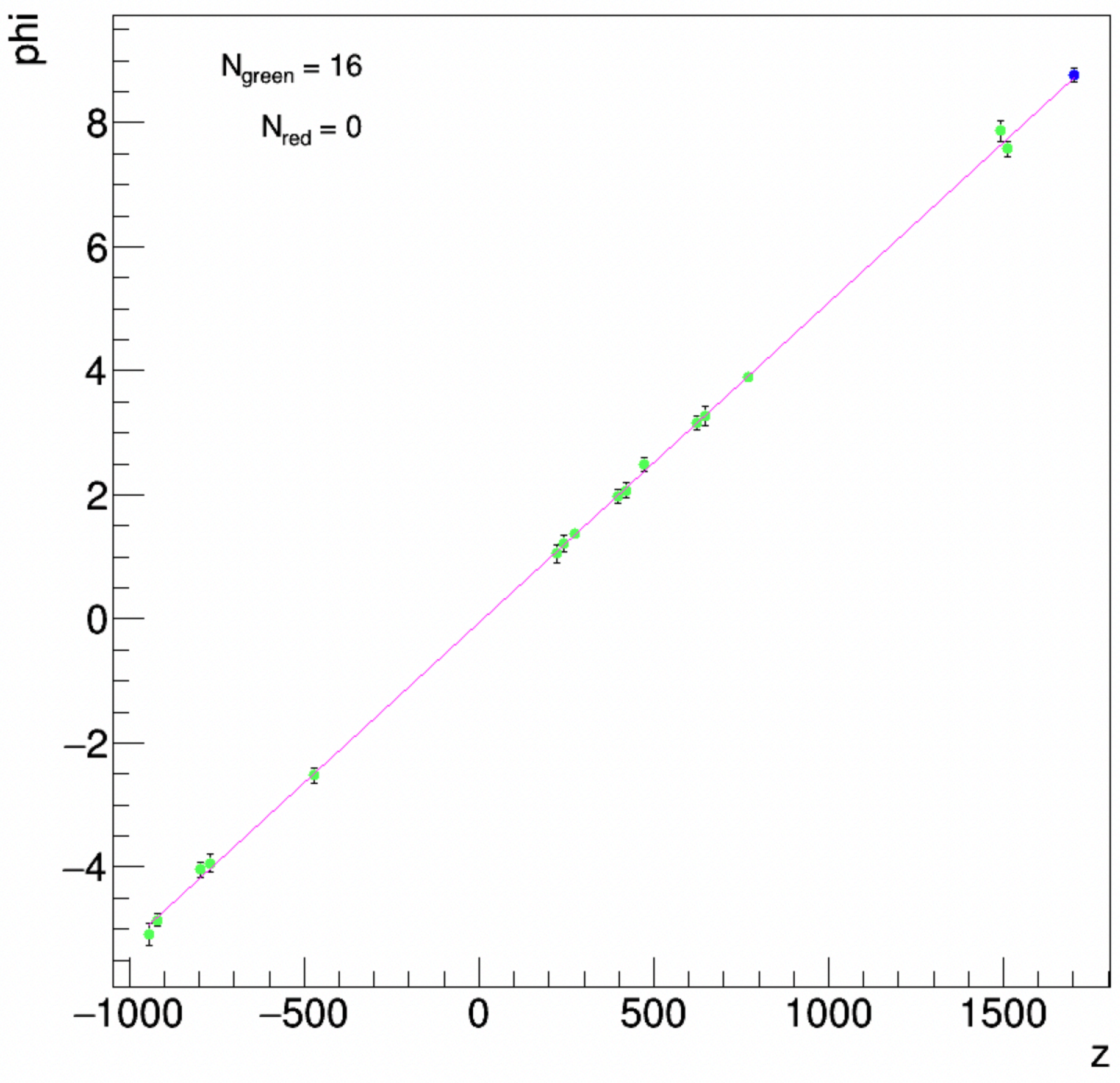


HelixFinder Logic

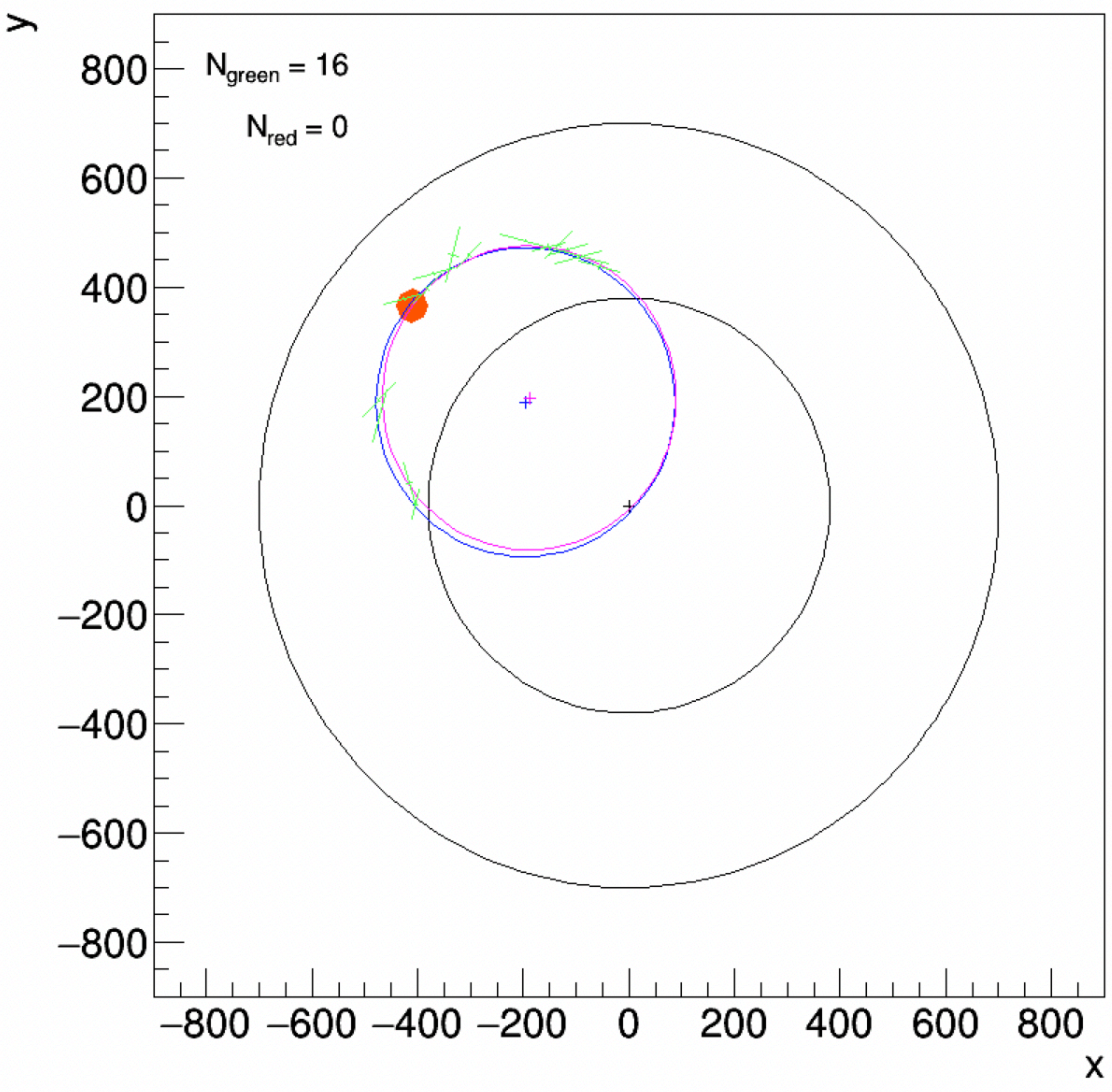
9. Refine all of the candidate lines by removing worst hit and updating line until all hits are within some residual
10. Keep the line with the most hits as the final candidate, re-do circle fit using these hits to get final circle seed
11. Update phi’s with new circle to get final line seed

HelixFinder Logic

Phi-z view after step 10



XY view after step 10

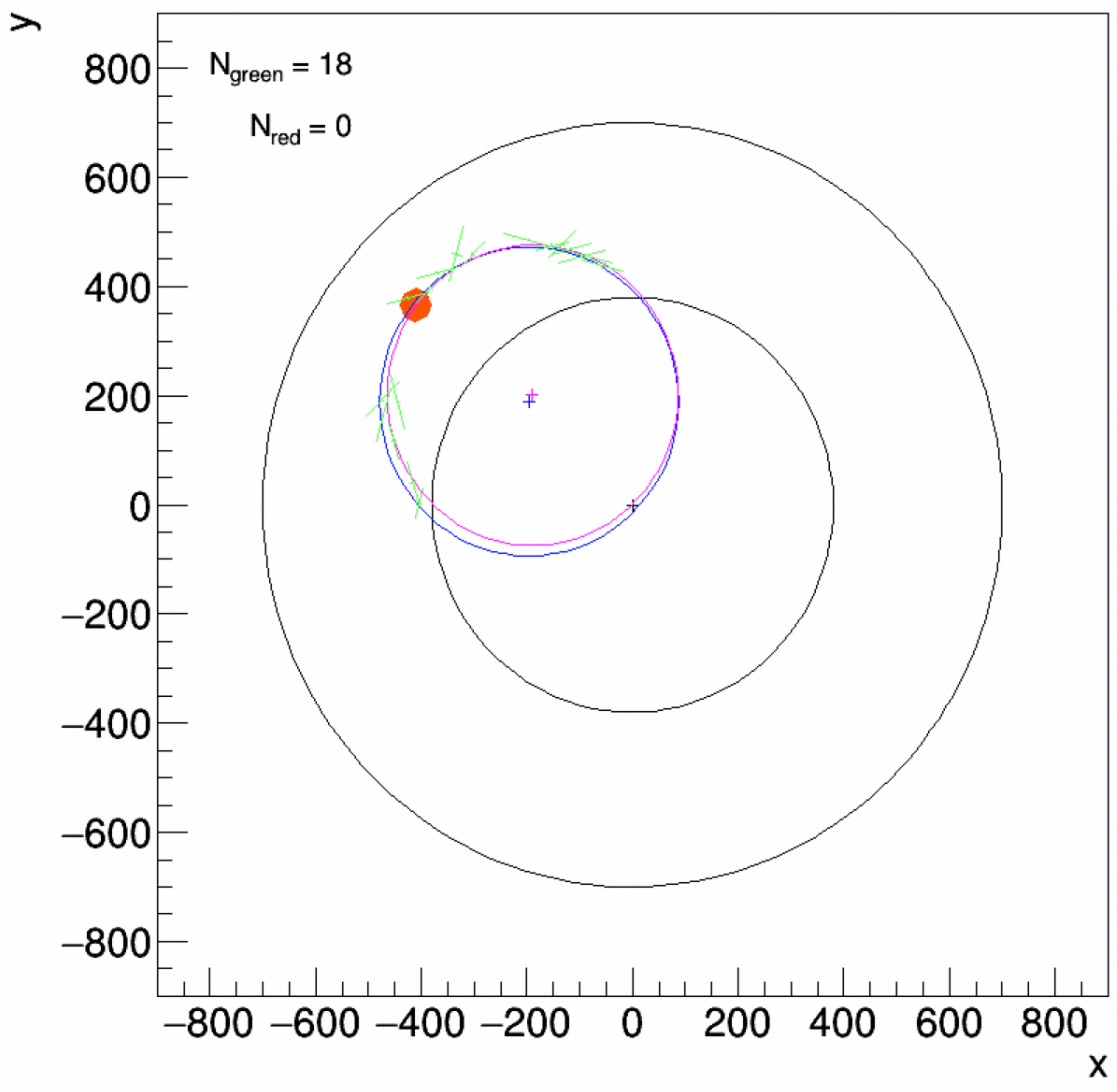


HelixFinder Logic

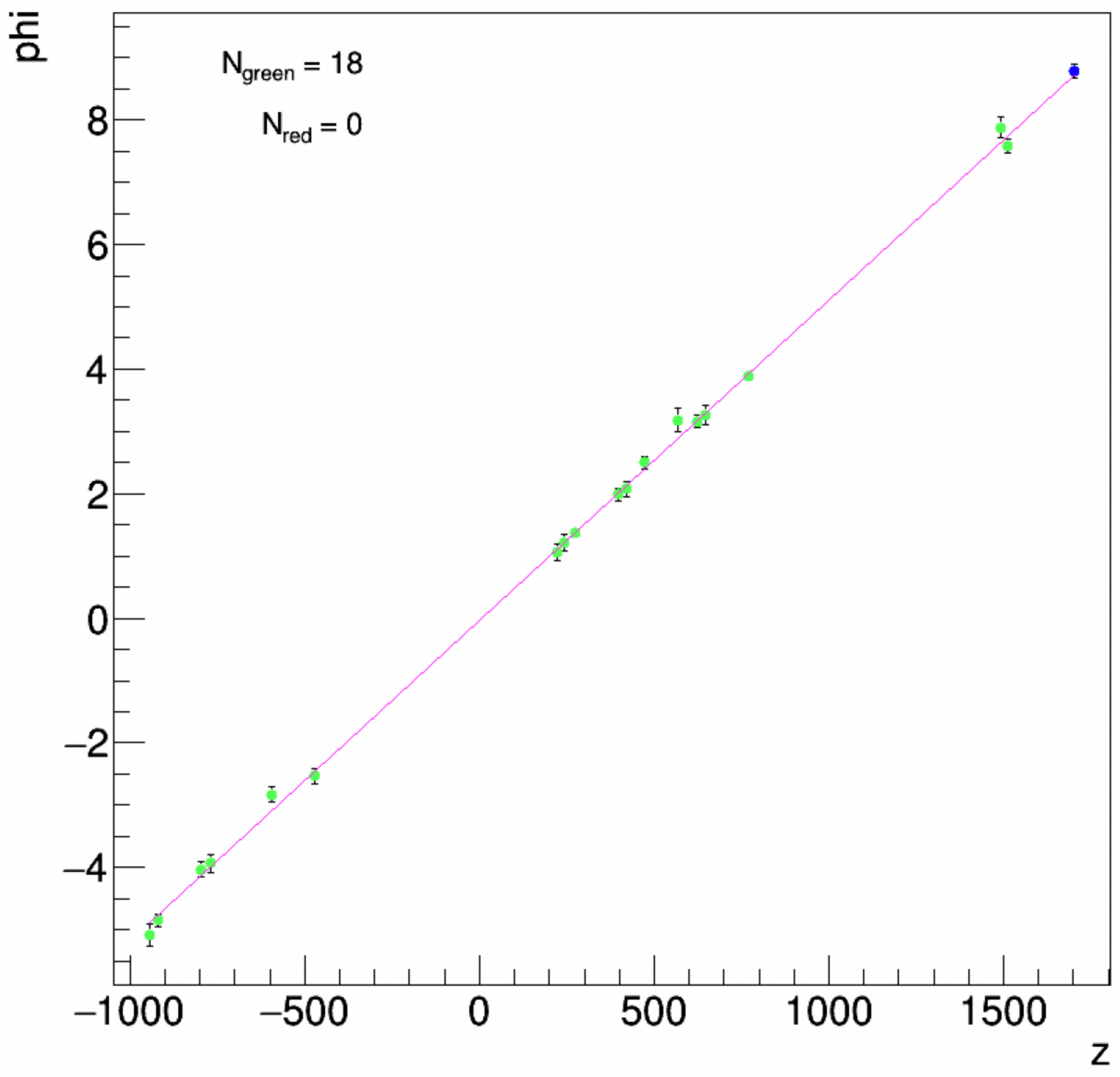
12. Recover hit with smallest residual if it is below some threshold, and update circle
13. Repeat step 12 until no hits are left to be recovered
14. If a helix is found, repeat steps 1-13 to search for another one, and repeat until one isn't found

HelixFinder Logic

XY view of final helix circle

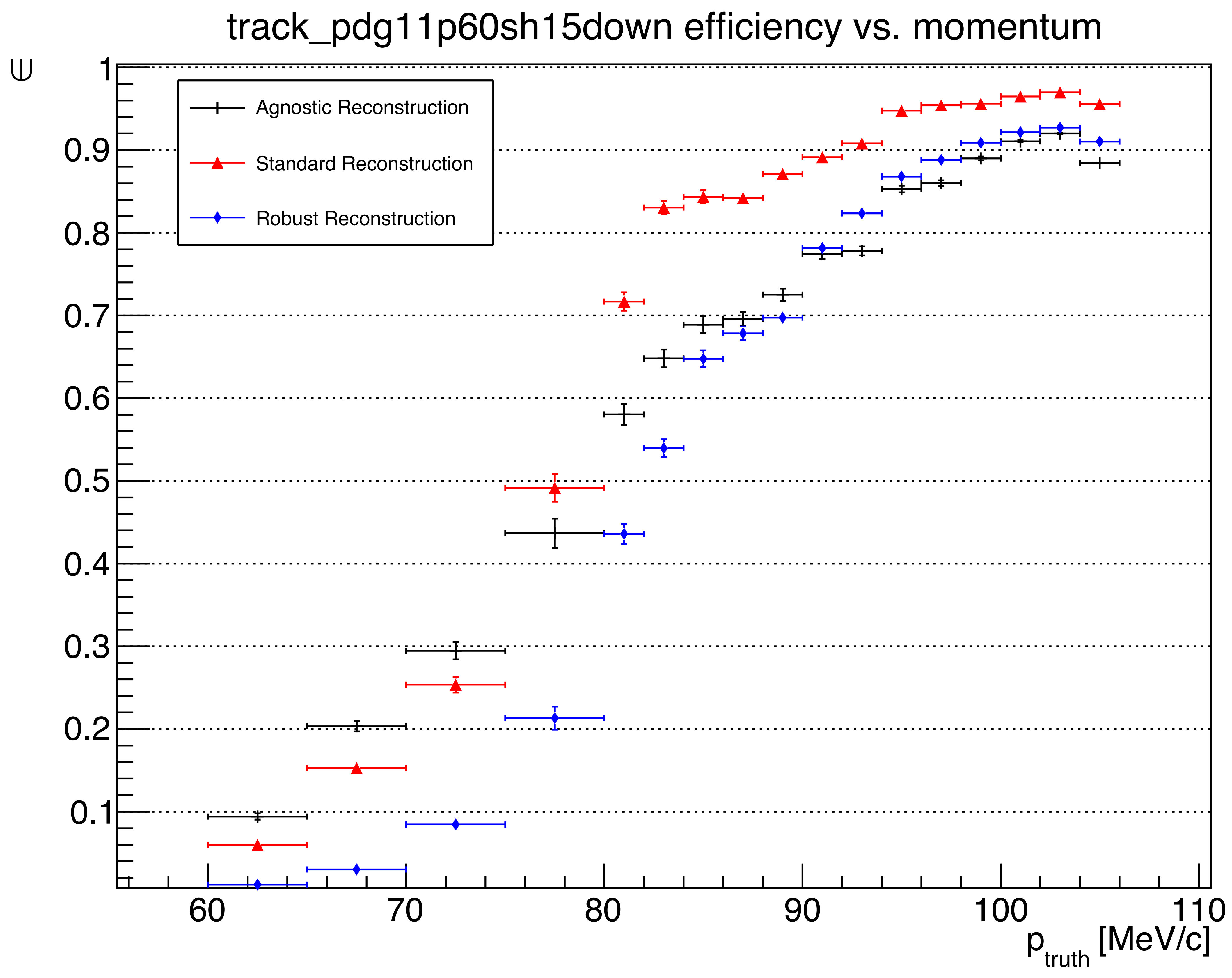


phi-z view of final helix line



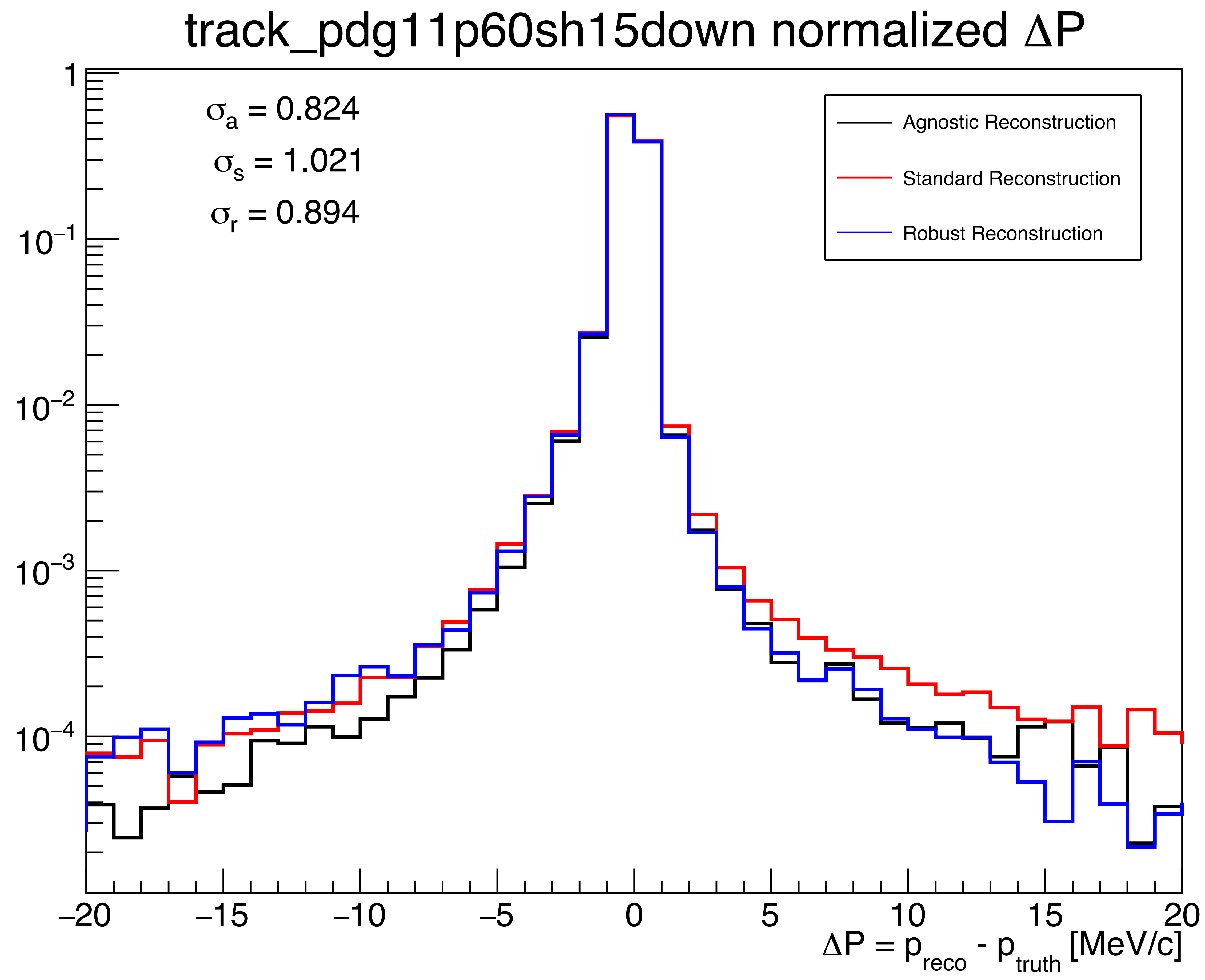
HelixFinder Downstream Performance

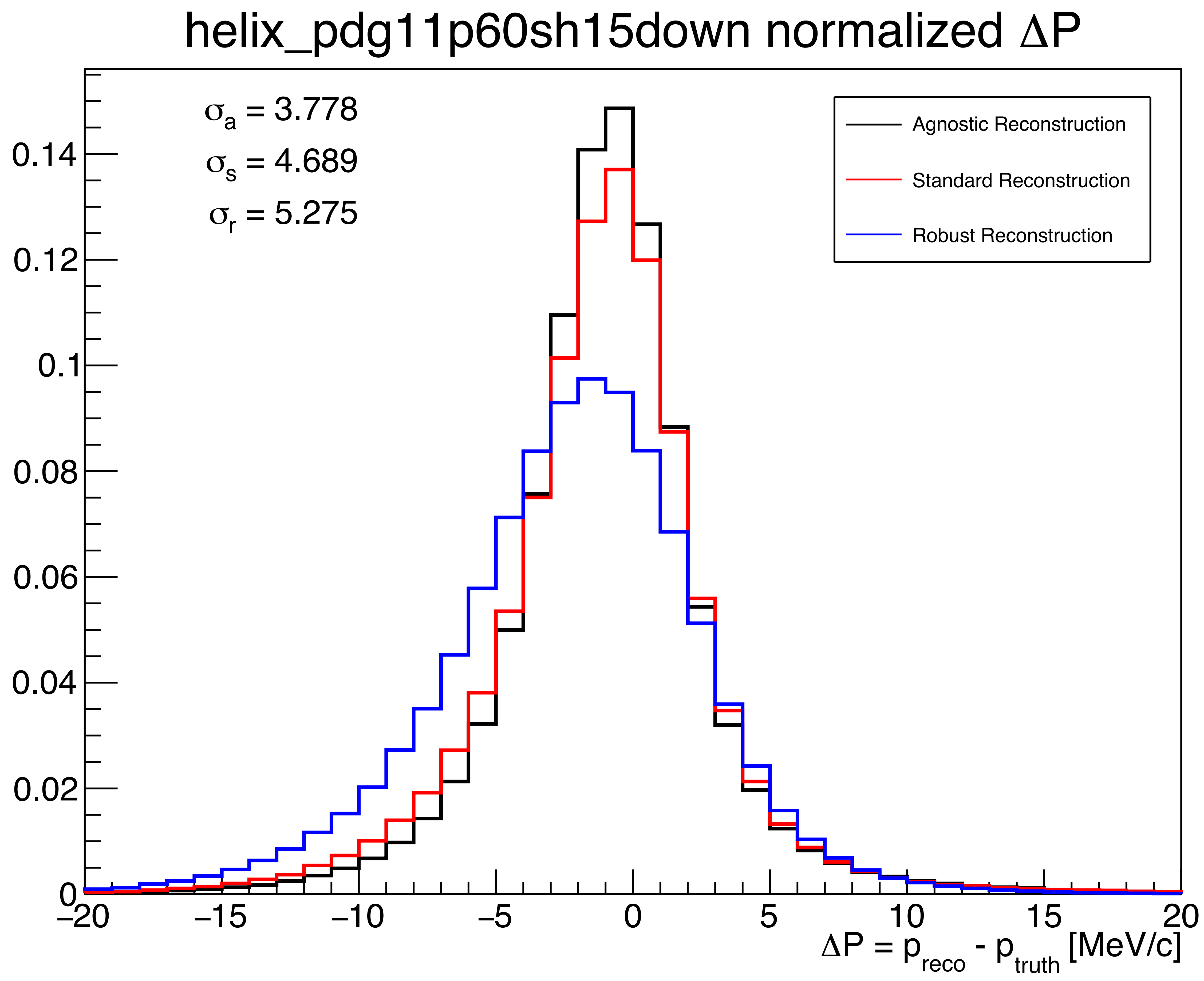
- We can see how new agnostic reconstruction sequence (TZClusterFinder + HelixFinder) compares to TPR and TPR+CPR in reconstructing downstream electrons leaving at least 15 straw hits in tracker



- Below 80 MeV/c, agnostic reconstruction is outperforming the current reconstruction sequences
- The "robust reconstruction" (TPR) and the agnostic reconstruction are performing similarly for downstream electrons above 80 MeV/c
  - Gap between standard reconstruction (TPR+CPR) and agnostic reconstruction mostly due then to gains from calorimeter seeded helix finder, CalHelixFinder
    - Stopping target assumption providing stability

HelixFinder Downstream Performance

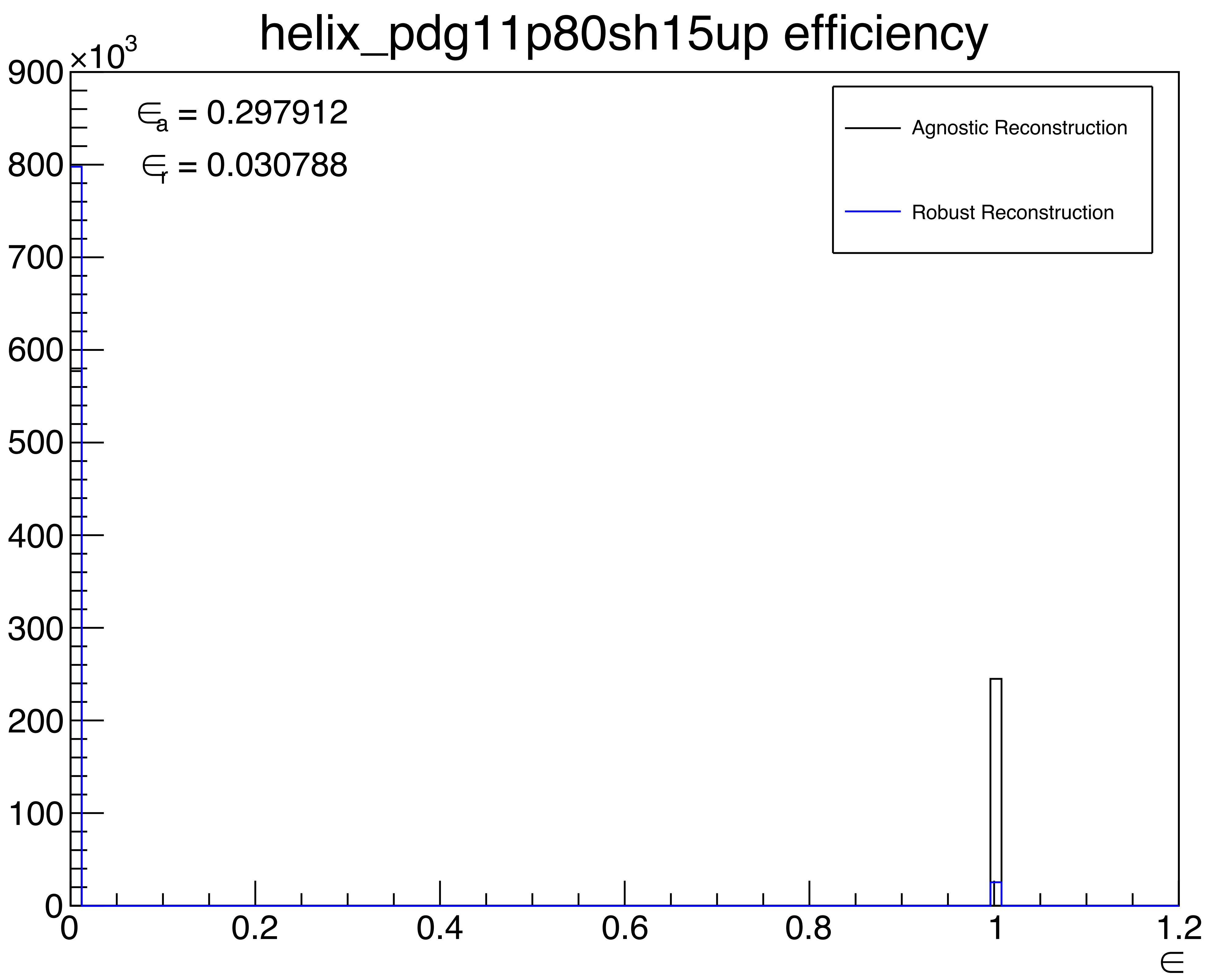




- Agnostic track reconstruction showing similar/better momentum resolution at the track level
- Agnostic reconstruction visibly better at helix level

HelixFinder Upstream Performance

- HelixFinder greatly outperforming current reconstruction when it comes to upstream electrons with p>80 MeV/c leaving at least 15 straw hits in the tracker



- Only comparing agnostic sequence and robust sequence since CalHelixFinder is written specifically for tracks coming from stopping target
- Can see agnostic reconstruction is performing much better than robust reconstruction
- The gains in the agnostic reconstruction come partially from TZClusterFinder outperforming TimeClusterFinder, and partially from HelixFinder outperforming RobustHelixFinder

Outline

1. The Mu2e Experiment
2. Measurement Technique
3. Current Track Reconstruction
4. Shortcomings & Examples
5. Current Work and Preliminary Results
6. Outlook

Outlook

- New time clustering module achieves what it set out to achieve
- Work on the HelixFinder still in early stages
  - Look more closely into failures
  - Consider methods to flag what hits are good for initial seeding
- Already an argument to replace RobustHelixFinder with HelixFinder
  - Wouldn’t lose conversion electron efficiency, would gain in other areas
- Could greatly simplify book-keeping of sequences if HelixFinder can outperform both CalHelixFinder and RobustHelixFinder combined

Backup

RobustHelixFinder

- Part of track seeded reconstruction
- Follows the logic:
  1. Takes all triplets (excluding those not meeting certain conditions) in a given time cluster and computes a circle center (x_o, y_o)
  2. The median x_o and y_o are chosen to be the circle center
  3. The distance of each point is computed from this center and the median distance is chosen as the circle radius
  4. Using hits consistent with the circle, $d\phi/dz$ is computed for each pair of hits (within some dz of each other) and histogrammed. The peak of this histogram is taken to be $d\phi/dz$.
  5. The $d\phi/dz$ found in step 4 is used to add $2\pi n$ to ϕ for each hit
  6. The new ϕ's from step 5 are used to repeat step 4 and find an updated $d\phi/dz$.
  7. Hits that aren't consistent with the circle and ϕ vs. z line are thrown away
  8. Steps 1-3 and steps 6-7 are repeated until no hits are thrown away

CalHelixFinder

- Part of calorimeter seeded reconstruction
- Follows the logic:

1. Takes hit at largest z + stopping target + calorimeter cluster to make a triplet to derive circle parameters
2. Moves backwards in z using assumed $d\phi/dz$ and adds hits consistent with predicted location
3. Once enough hits are collected, the circle parameters and $d\phi/dz$ are updated (circle parameters updated using simple chi2 fit, $2\pi n$ corrections found using histogramming method similar to RobustHelixFinder, then $d\phi/dz$ updated using chi2 fit).
4. Step 2 is repeated using new circle parameters and $d\phi/dz$. With each added hit the circle parameters and $d\phi/dz$ are updated
5. Step 4 is repeated until all hits have been tested.
6. Steps 1-5 are repeated, except in step 1 the next highest z point is used for the original triplet, and so on.
7. The helix with the most hits is taken as the candidate