Fitting the DiRC output

Strategy

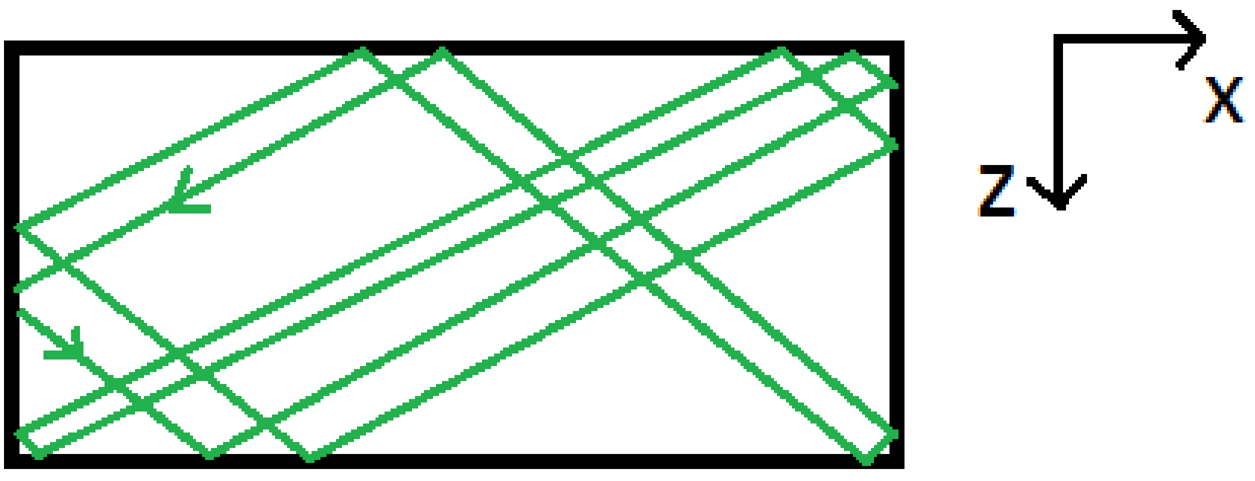
- Generate expected pdf of particle "on the fly"
  - Use known position, angles and momentum
  - Generate 1 pdf for each candidate particle
- Compare likelihood ratios

Generating the PDFs

- Propagate light rays from particle trajectory
  - Generated uniformly along Cerenkov cone's phi and particle trajectory
- Intersect these rays with the sense plane
- "Spread" the points on the sense plane with some function
  - Currently using a Gaussian or linear decrease
- Problem: it's slow – many many reflections

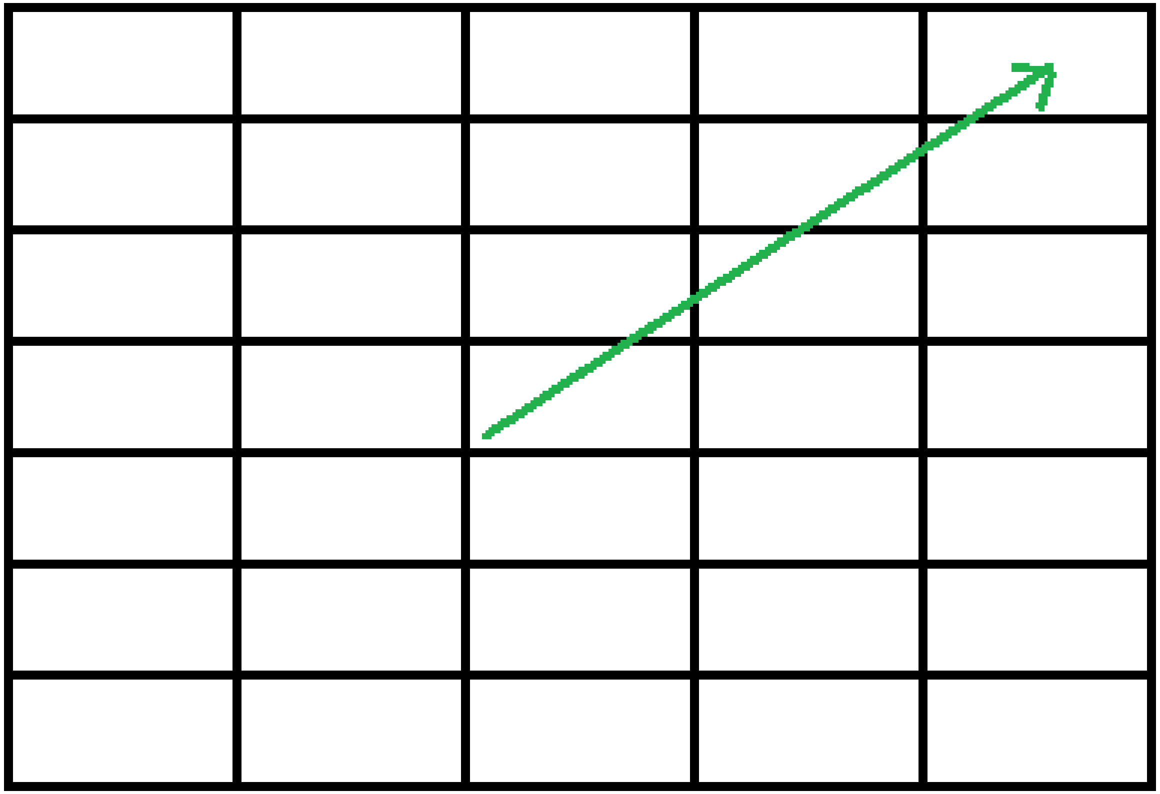
Solution: Analytically propagate

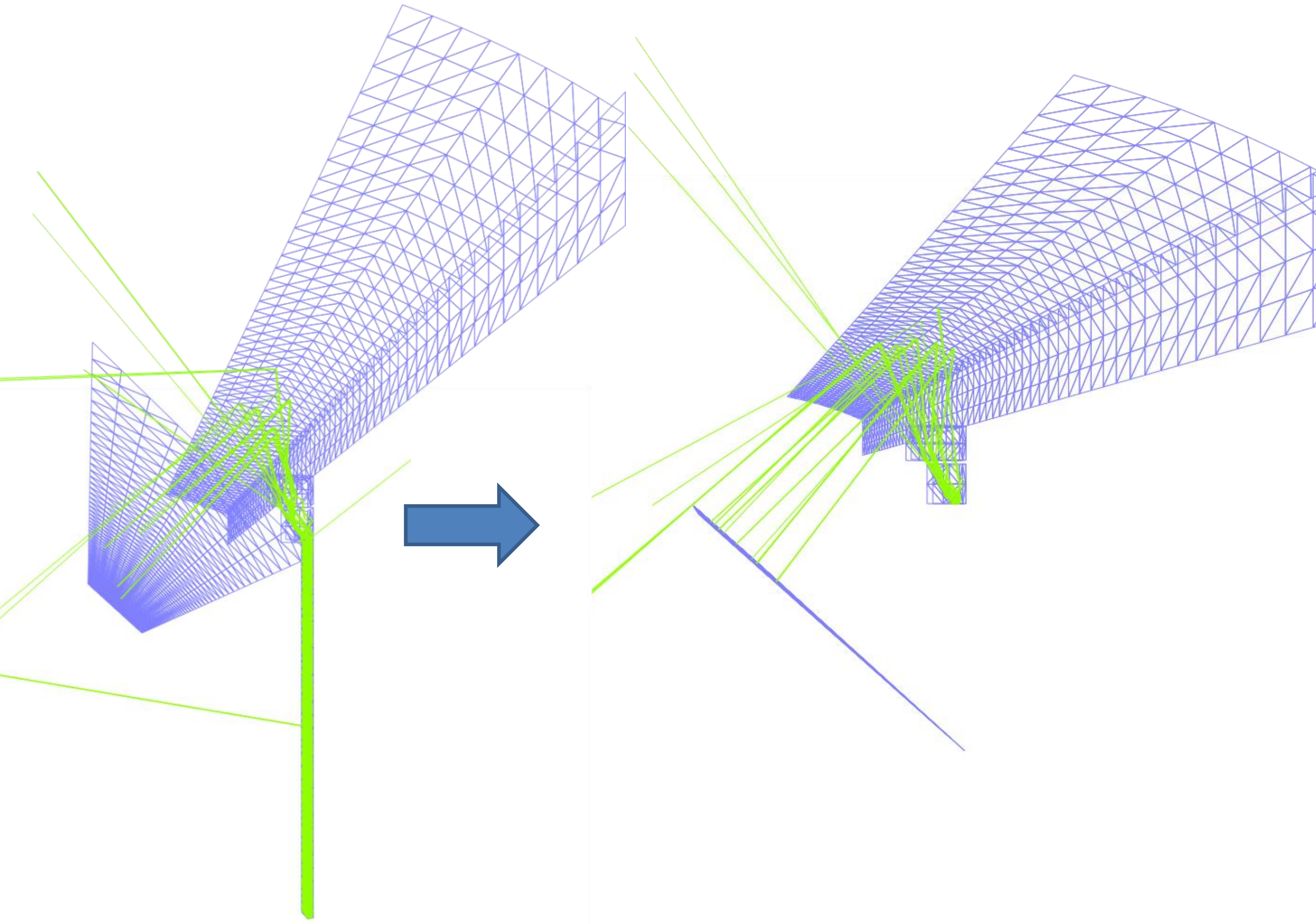
- Most of the reflections are internal
- ~1000 bounces:



Analytic Reflections

- Instead treat as line through a grid:





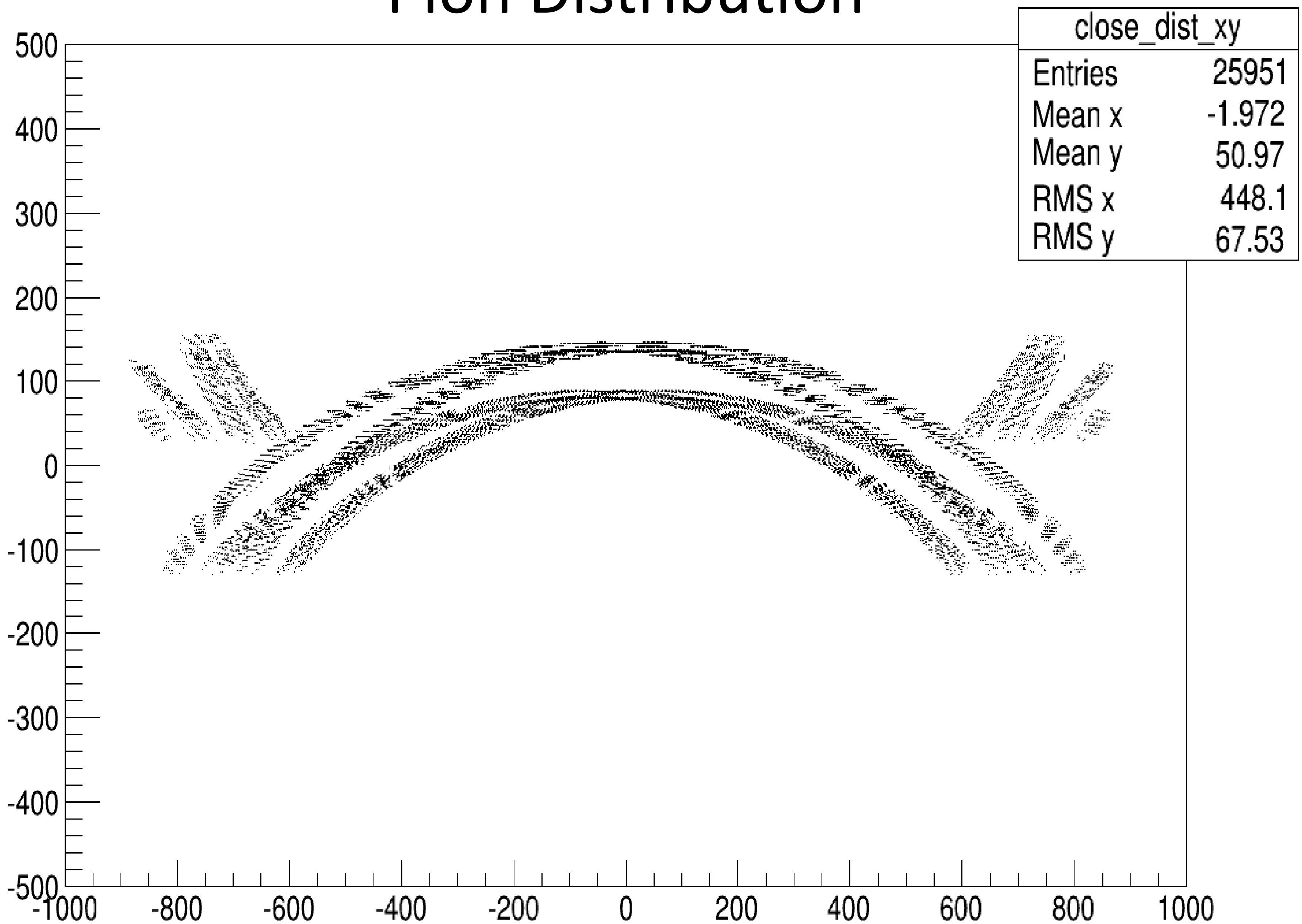
Speedup

- 43s -> 375ms per evaluation (my laptop)
  - Using 40k support points
  - Need at least 2 evaluations per particle
- Separation depends on number of points
  - 20k support points gives very similar performance (efficiency and missid cross ~2% lower)
- Looking to do more reflection explicitly – cut out goptical entirely
  - still ~4 reflections being done possibly slowly

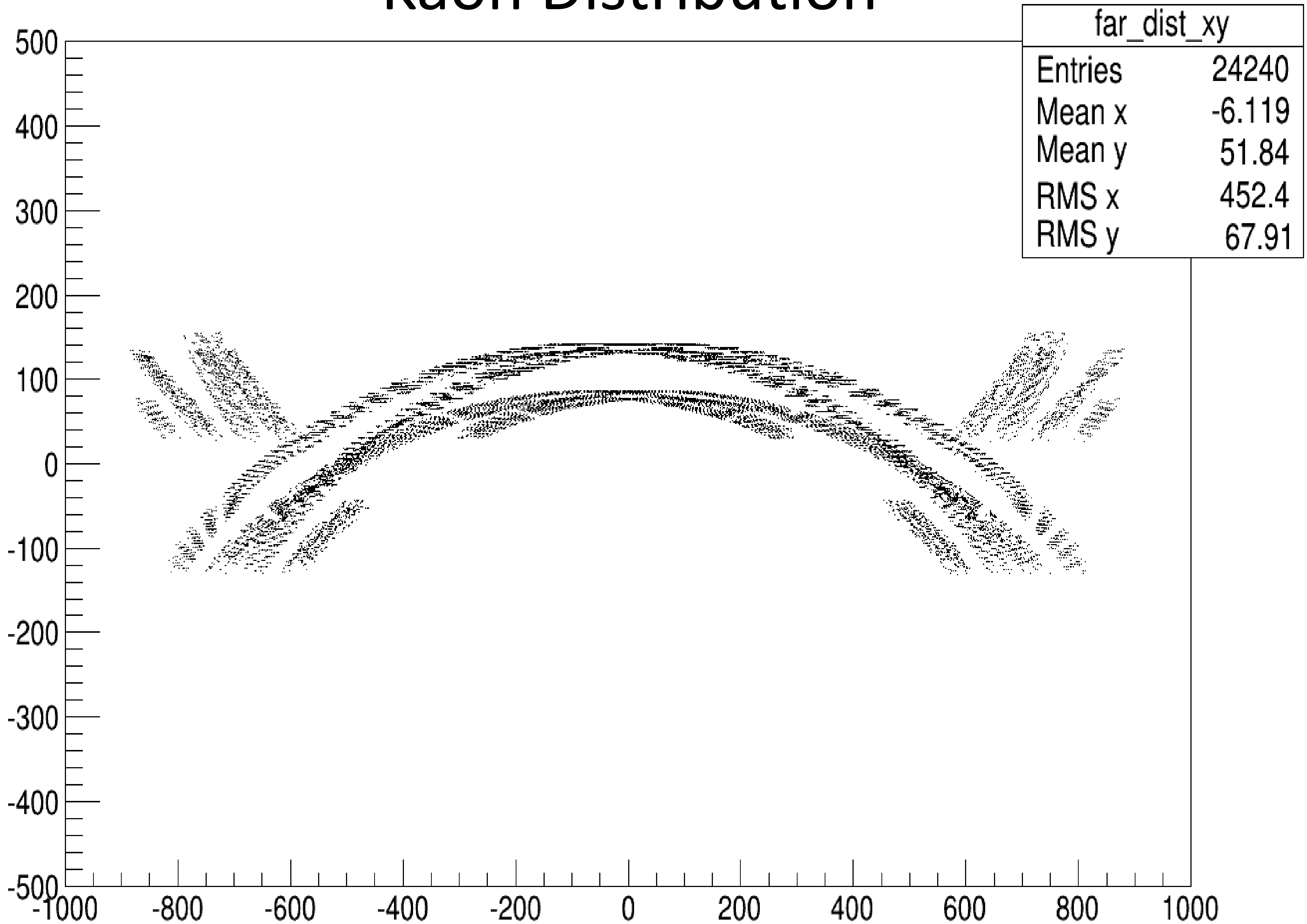
Testing

- Examine Kaon and Pion at 4.5 GeV
  - Pion Cerenkov Theta = 47.11 deg
  - Kaon Cerenkov Theta = 46.82 deg
- Particle Trajectory:
  - Theta = 4 deg
  - Phi = 40 deg
  - Also tried perpendicular (0,0) – single curve, distribution not shown
  - Spread 1.5mrad – worst case tracking in PAC
  - Shown with and without chromatic/optical aberation (~7.1mrad)
- Ignoring side box mirrors
- Have ~24 photons from each particle hit mirror

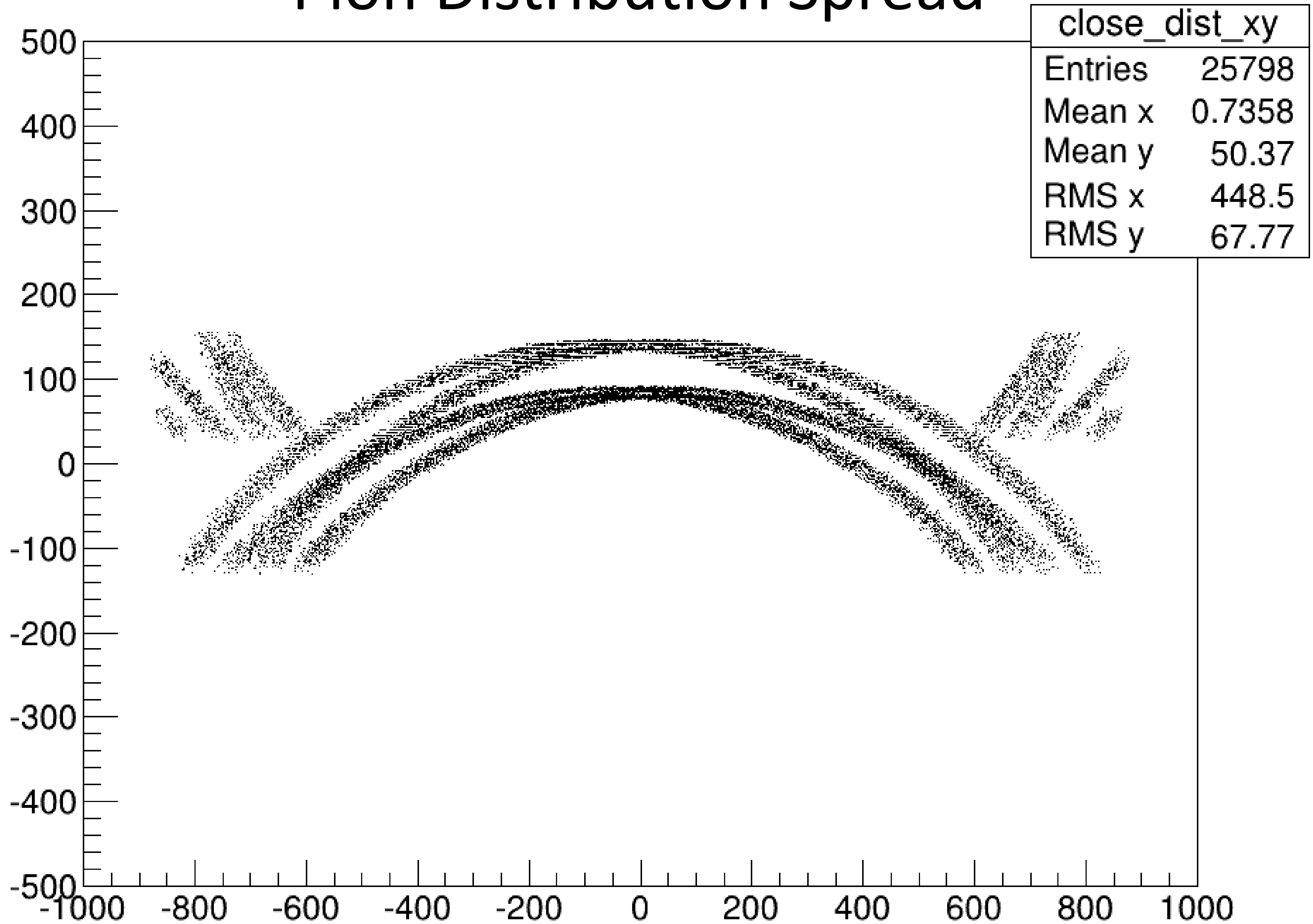
Pion Distribution



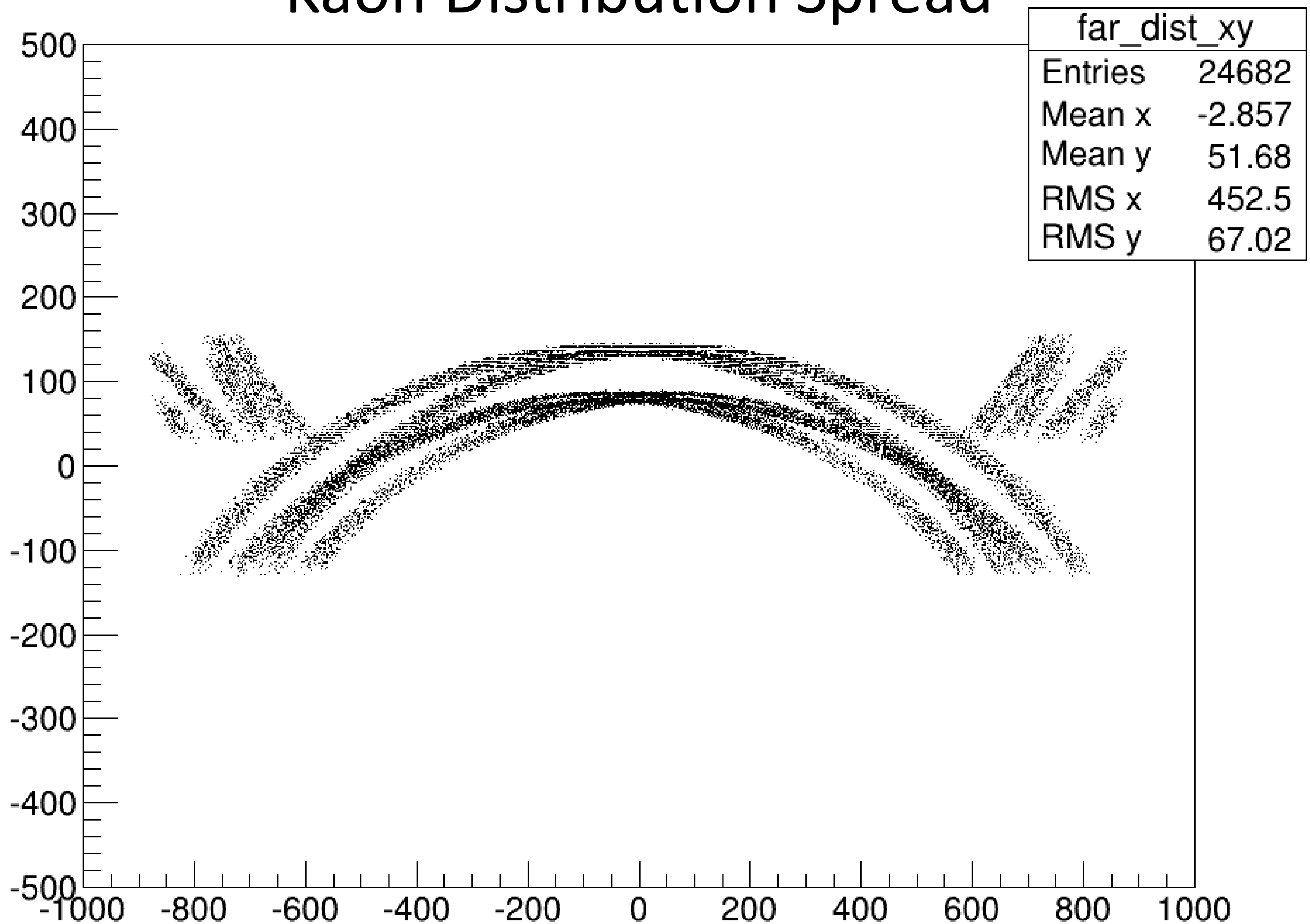
Kaon Distribution



Pion Distribution Spread

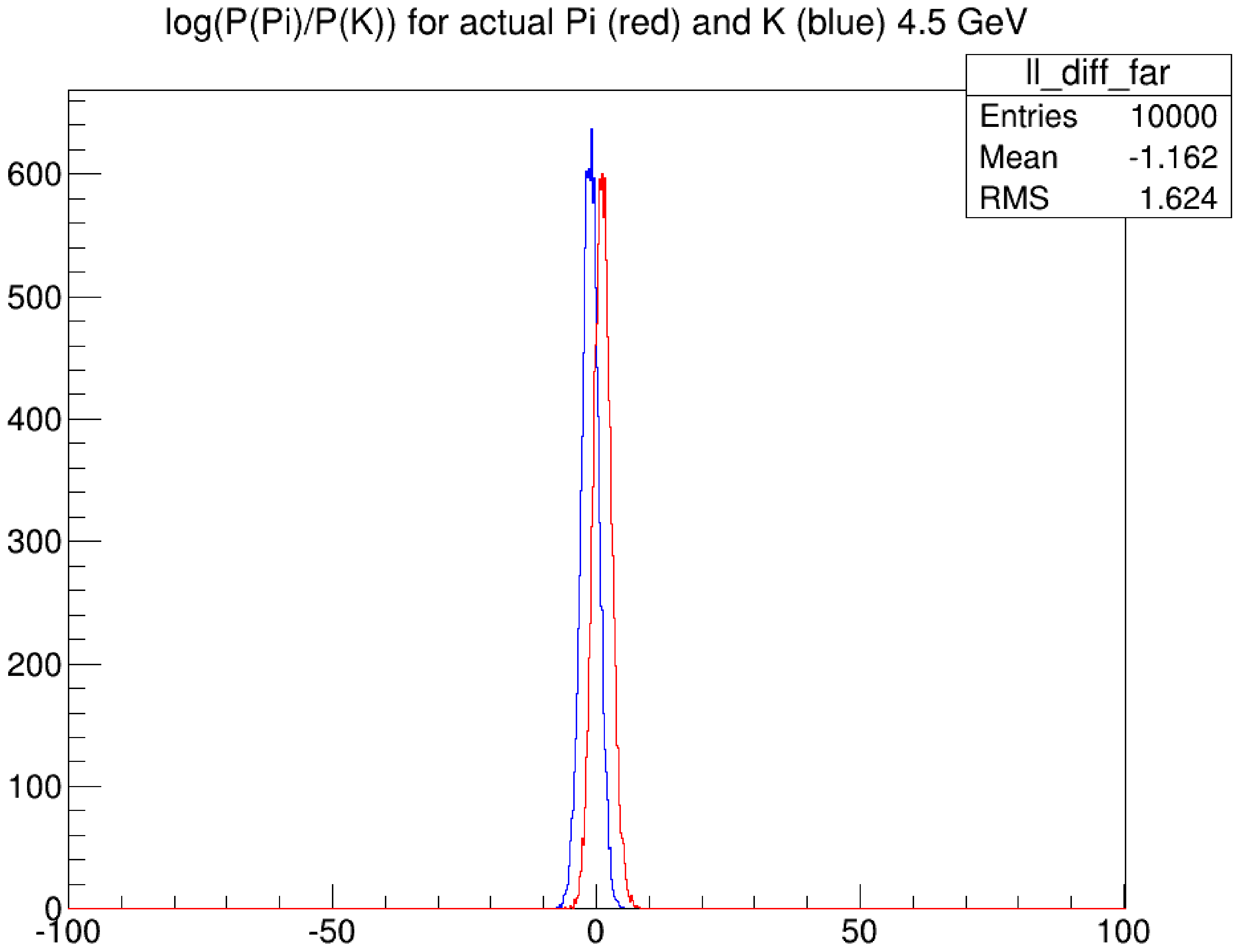


Kaon Distribution Spread

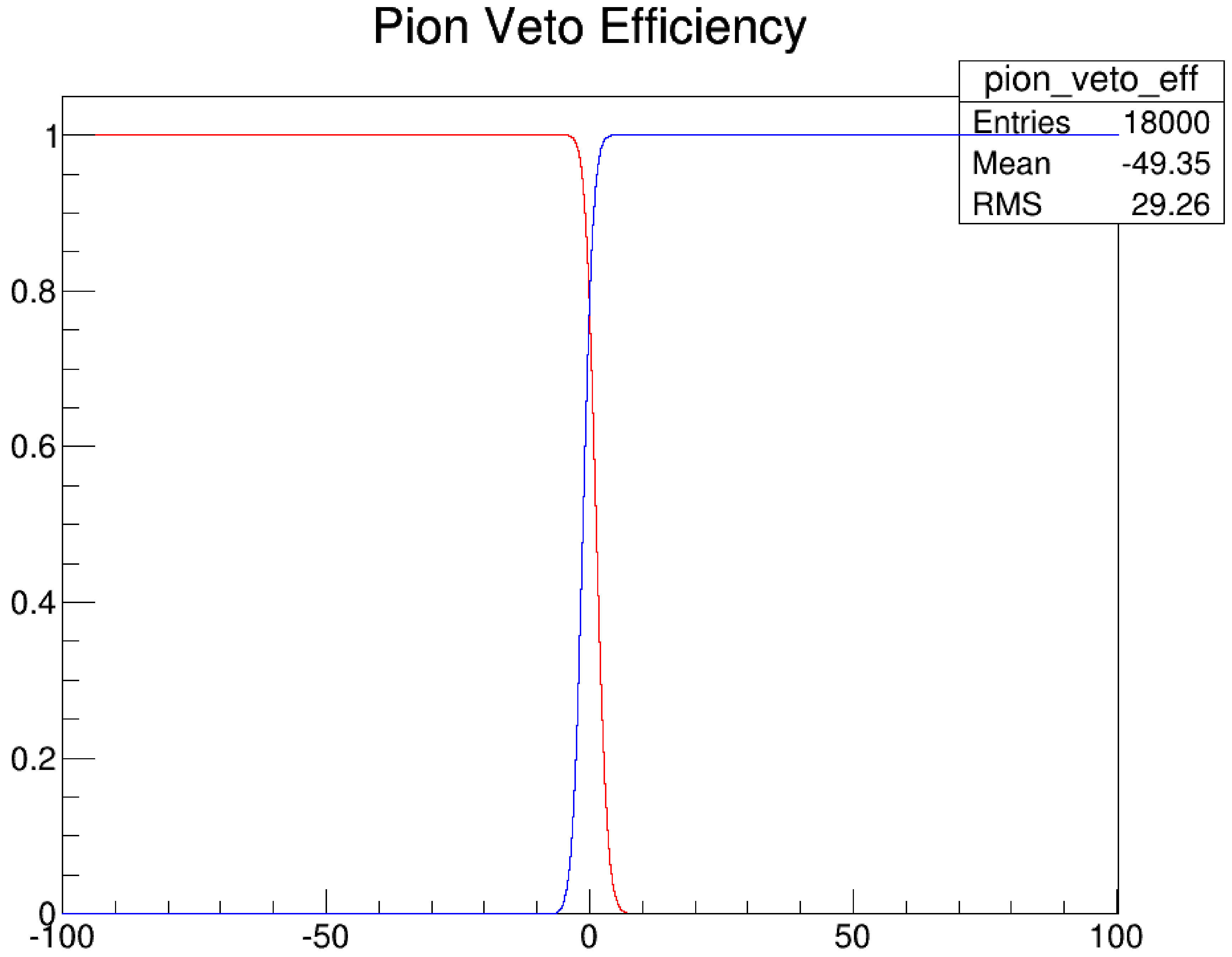


Pdfs

- Used "exponential tent" around each support point with width 2mm
- Also tried linear slope down around each point and exponential decay
- Sum the spreading function over all support points and divide by the number of support points for each "real" hit ray. The sum the logs of these values for the "loglikelihood" of the Pion or Kaon. Get the following plots:



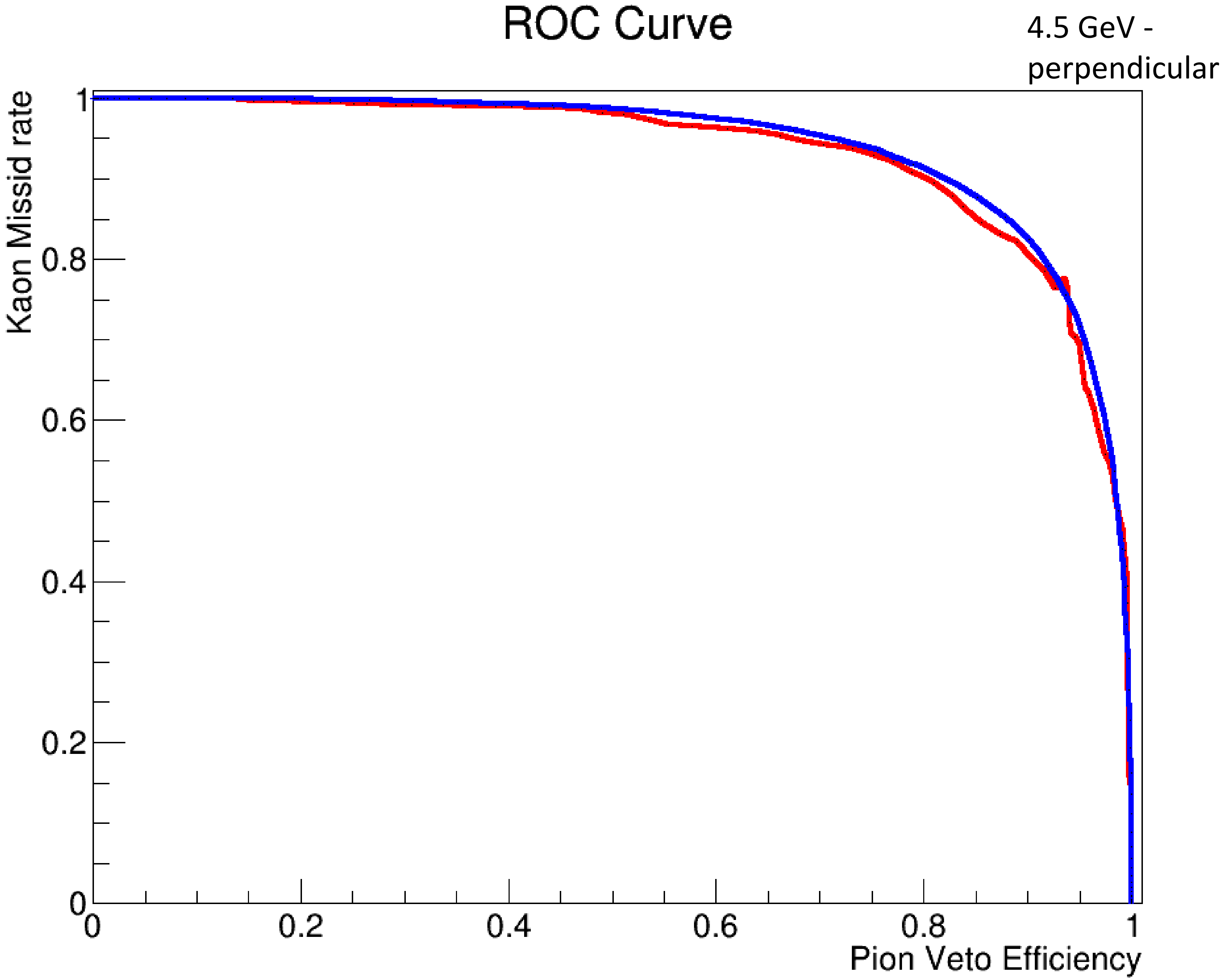
log(P(Pi)/P(K)) for actual Pi (red) and K (blue) 4.5 GeV
ll_diff_far
Entries 10000
Mean -1.162
RMS 1.624
600
500
400
300
200
100
0
-100
-50
0
50
100

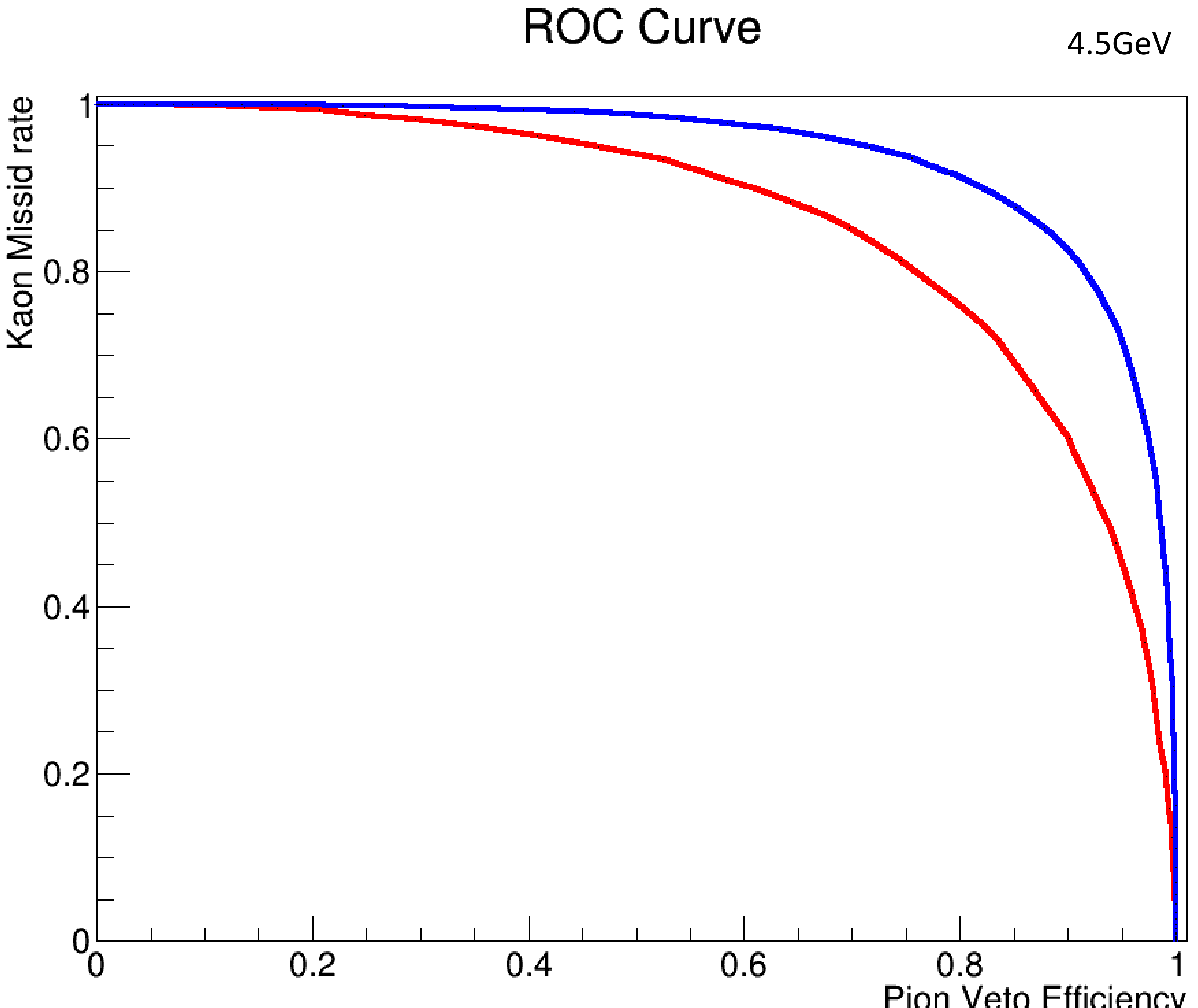


Pion Veto Efficiency
pion_veto_eff
Entries 18000
Mean -49.35
RMS 29.26
1
0.8
0.6
0.4
0.2
0
-100
-50
0
50
100

Results

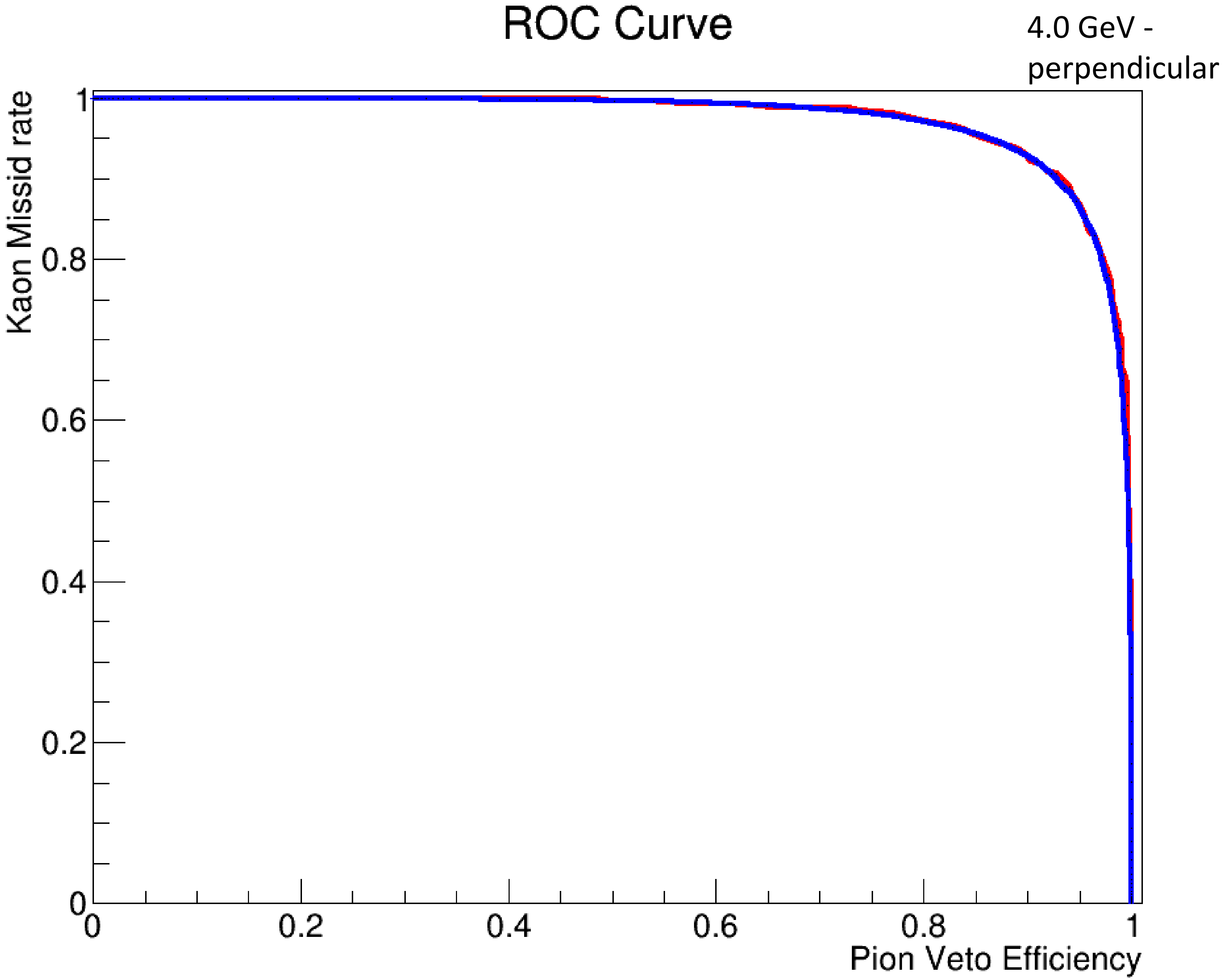
- Integrate red curve from right and blue curve from left at each point
  - Get missID and veto efficiency for a cut at that value.
- Plot these as a ROC curve (in red)
- Compare to expected ROC curve with Gaussian separation of 2.5mrad (in blue, angles are 5.23 mrad apart)
- Achieved ROC curve comparable to 2.5 mrad uncertainty for the perpendicular case
  - Get ROC curve comparable to 3.8mrad with particle not perpendicular to the plane - looking into how to fix this


ROC Curve
4.5 GeV -
perpendicular
Kaon Missid rate
Pion Veto Efficiency
0
0.2
0.4
0.6
0.8
1

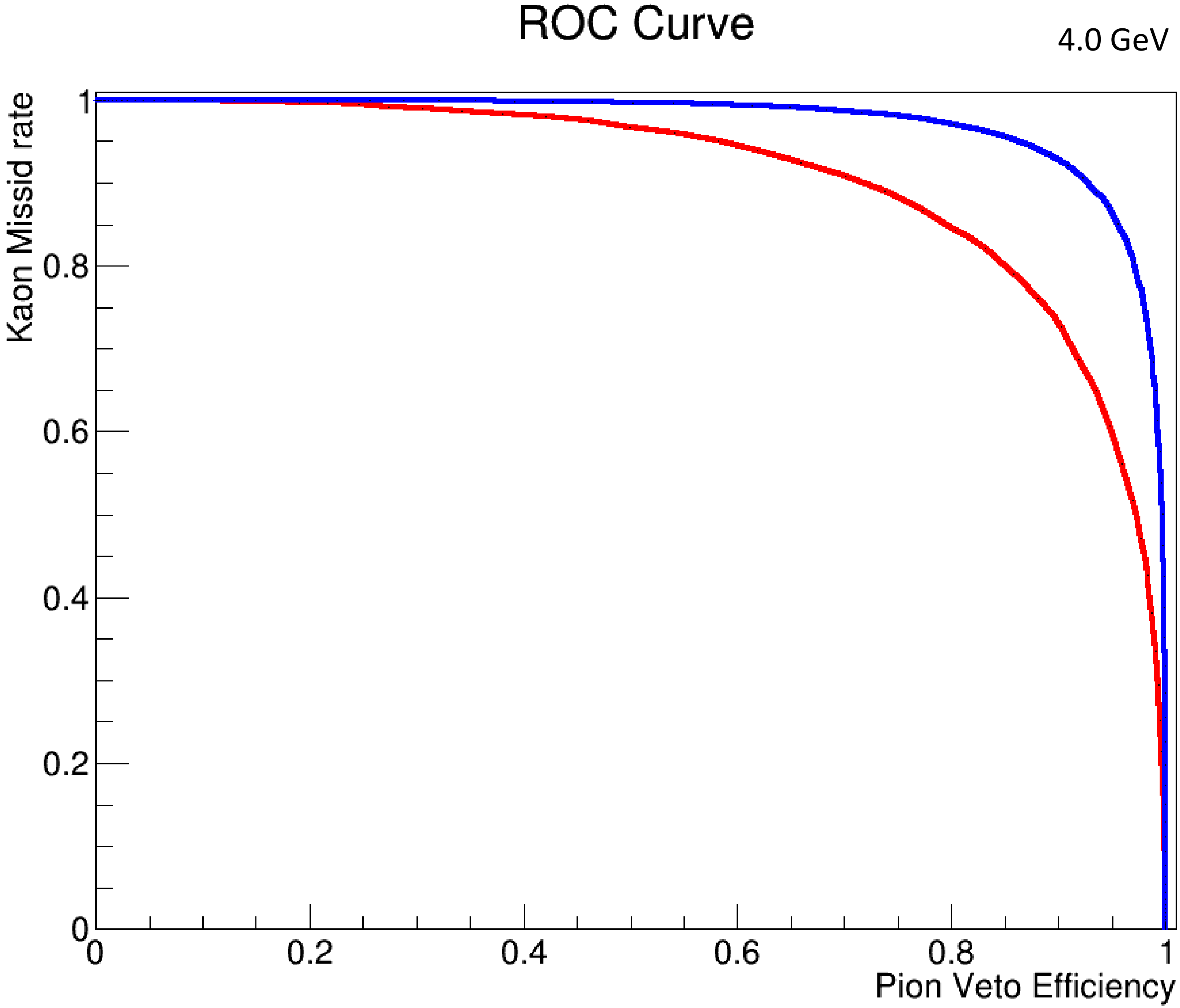

ROC Curve
4.5GeV
Kaon Missid rate
Pion Veto Efficiency
0
0.2
0.4
0.6
0.8
1

Data for 4 GeV

- 4 GeV is the limit quoted in PAC
- Separation is 6.63mrad
- Rerun and get ROC curves
  - Better results due to larger separation
  - Comparable to 2.5 mrad in perpendicular case
  - Achieved curve still comparable to 3.8 mrad in non-perpendicular case



ROC Curve
4.0 GeV -
perpendicular
Kaon Missid rate
Pion Veto Efficiency
0
0.2
0.4
0.6
0.8
1



ROC Curve
4.0 GeV
Kaon Missid rate
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Pion Veto Efficiency