

**Double Spin Asymmetry in Inclusive π^0 Production for Longitudinally
Polarized proton proton collisions at $\sqrt{s} = 200\text{GeV}$ at the Endcap
ElectroMagnetic Calorimeter at STAR**

Weihong He



Submitted to the faculty of the University Graduate School in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
In the Department of Physics,
Indiana University Bloomington
August 2008

Acceptance Page

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

(Chair: Scott W. Wissink)

(Charles Horowitz)

(William Jacobs)

(James Sowinski)

September 09, 2008

Acknowledgements

During my PhD study at Indiana University Bloomington, I have learned a lot from many friends and scientists. Now, I am to finish my thesis, and I would like to thank many people who have helped me, so that I can get the PhD degree in physics. Please allow me to thank my advisor Scott Wissink first; I could not get the PhD without your instructions, considering me as a foreigner coming from the far China and knowing little about the America. I would like to thank Jan Balewski, William Jacobs, James Sowinski, Steven Vigdor and Jason Webb. I have benefited from you very much because we had been working closely as a research group. I would like to thank Charles Horowitz for the service of the defense committee. I would like to thank Pibero Djawotho and Ilya Selyuzhenkov. I am very happy that we worked together and discussed many research details. I want to thank my family too, because you are always behind me and supporting me.

I appreciate the chance to pursue my PhD at Indiana University Cyclotron Facility and the department of physics; I appreciate the chance to study at Bloomington; I appreciate the chance to join the STAR spin program. Bloomington, I will miss you.

Abstract

Student's Name: Weihong He

Dissertation Title:

Double Spin Asymmetry in Inclusive π^0 Production for Longitudinally Polarized proton
proton collisions at $\sqrt{s} = 200\text{GeV}$ at the Endcap ElectroMagnetic Calorimeter at STAR

Measurement of the double-spin asymmetry A_{LL} for inclusive π^0 production in polarized proton proton collisions can provide important constraints on gluonic contributions to the proton's spin. The STAR Endcap Electromagnetic Calorimeter (EEMC) is well suited for these studies, providing full azimuthal coverage for $1.086 < \eta < 2.0$, and with separate readout of two pre- and one post-shower layers, and a fine grain scintillator-based shower maximum detector (SMD) that can distinguish between single photons or electrons, charged hadrons, and neutral mesons (π^0 's and η 's) via the observed transverse shower profile. The EEMC also provides fast triggering on significant energy deposition in individual towers, trigger patches, or jet patches ($\Delta\eta \times \Delta\phi \approx 0.007, 0.06, 1$, respectively).

Details of the π^0 reconstruction algorithm employed, and the first analysis results in this pseudorapidity range for inclusive π^0 s will be reported. Results are based on the 2006 (run6) longitudinally polarized pp data set (sampled luminosity $\approx 3.5\text{pb}^{-1}$) acquired at the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Lab (BNL), at a center of mass energy $\sqrt{s}=200\text{ GeV}$ and with average proton beam polarizations of 54% (blue) and 56% (yellow). The measured double spin asymmetry A_{LL} is consistent with

next-to-leading order perturbative quantum chromodynamics (NLO pQCD) calculations,
and expectations from fits to polarized deep-inelastic scattering data.

(Chair: Scott W. Wissink)

(Charles Horowitz)

(William Jacobs)

(James Sowinski)

Contents

1	Introduction	1
1.1	Thesis outline.	.1
1.2	RHIC spin program overview	.3
1.3	Physics motivation	.4
1.3.1	Origin of the proton's spin.	.4
1.3.2	Double Spin Asymmetry A_{LL} in ΔG study.	.9
1.3.3	Neutral pion yield from the proton proton collision.	.13
1.4	Inclusive neutral pion analysis overview at RHIC.	.16
2	Experimental Setup	18
2.1	Polarized Proton Collider at RHIC.	.18
2.2	STAR hardware system	.22
2.2.1	Magnet system	.24
2.2.2	Time Projection Chamber.	.24
2.2.3	Forward Time Projection Chamber	.25
2.2.4	Silicon Vertex Tracker.	.25
2.2.5	Silicon Strip Detector	.25
2.2.6	Photon Multiplicity Detector	.25
2.2.7	Forward Pion Detector.	.26

2.2.8 Barrel Electro-Magnetic Calorimeter26
2.2.9 Beam Beam Counter.26
2.3 The Endcap Electro-Magnetic Calorimeter.27
2.3.1 EEMC physics performance requirements28
2.3.2 Tower mechanical structure.31
2.3.3 Shower Maximum Detector.33
2.3.4 Pre-shower and post-shower detectors36
2.3.5 EEMC readout and trigger system37
2.4 STAR Data Acquisition (DAQ) system40
3 Software Development	42
3.1 π^0 decay in the endcap.42
3.2 π^0 finder software introduction43
3.3 The Cluster-Maker45
3.4 The Point-Maker50
3.5 The Pion-Maker.55
4 Simulation Study	59
4.1 Introduction.59
4.2 Energy sharing among EEMC towers.60
4.3 π^0 energy deposition study in the EEMC.65
4.4 π^0 reconstruction and efficiency study.74

4.4.1	di- π^0 Monte-Carlo sample analysis.76
4.4.2	Single π^0 Monte-Carlo sample analysis.87
4.4.3	Pythia simulation sample analysis.91
4.4.4	Reconstruction efficiency summary.95
5	Real Data Analyses	96
5.1	Data selection.96
5.2	π^0 finder software setup.99
5.2.1	SMD fluctuation study100
5.2.2	Edge effect.109
5.2.3	The optimization of software setup.115
5.3	The Double spin asymmetry A_{LL}118
5.3.1	Overview.119
5.3.2	Yields after normalization.120
5.3.3	Fit consistency check on pythia events.128
5.3.4	A_{LL} and statistical uncertainties calculation.130
5.3.5	Systematic errors.131
5.3.6	The single spin asymmetries133
5.3.7	The background asymmetry.135
5.3.8	Final A_{LL} result136
6	Conclusions	140

Appendix

A. Run Selection.142

B. Beam Polarization.144

C. π^0 Software Development Details.146

Bibliography.184

Table of Tables

Table 1.1: Global QCD analysis for first moments at $Q^2=10 \text{ Gev}^2$8
Table 2.1: Main parameters of RHIC acceleration and polarized proton beams21
Table 2.2: Main physics performance requirements for the EEMC30
Table 2.3: Sector-dependent structure of SMD in the EEMC36
Table 5.1: The meta values for equation (5.1) according to the eta-bins.113
Table 5.2: Fit ranges and fixed parameters in the final fitting to extract π^0121
Table 5.3: Summary of the π^0 yields from all pT bins127
Table 5.4: The summary of double spin asymmetries and statistical errors131
Table 5.5: Three sets of fitting ranges.131
Table 5.6 Fit Parameters to extract π^0 yields.132
Table 5.7: Summary of systematic for the double spin asymmetry.133
Table 5.8: Summary of the inclusive π^0 double spin asymmetries and uncertainties .	.137
Table 5.9: A_{LL} theoretical prediction for $\Delta G=+G$ from GRSV138

Table of Figures

Figure 1.1: Parton densities and their uncertainties at $Q^2 = 10\text{GeV}^2$6
Figure 1.2: Gluon densities from different theoretical assumptions by GRSV (2001).7
Figure 1.3: ΔG variations from different data sources at $Q^2 = 10\text{GeV}^2$8
Figure 1.4: $A_1^N(x_q, Q^2 = 5\text{GeV}^2)$ from pDIS GRSV2000.11
Figure 1.5: $\hat{\alpha}_{LL}$ distributions from all possible parton reactions.12
Figure 1.6: Some Feynman diagrams of partonic interactions.14
Figure 1.7: Fractional contribution of each partonic sub-process at $\eta=0$15
Figure 1.8: Fractional contribution of each partonic sub-process at $\eta=3.3$15
Figure 2.1: The accelerator complex at BNL.19
Figure 2.2: STAR detector and a schematic profile of the 2006 geometry23
Figure 2.3: Schematic of half of the EEMC in a quarter of the STAR detector.27
Figure 2.4 Half of the EEMC tower structure.31
Figure 2.5: Megatile sample from a 6° subsector.32
Figure 2.6: Schematic of the SMD from a 30° sector.34
Figure 2.7: Asymmetric layout of the SMD sublayers in the EEMC.35
Figure 3.1: A flow chart from π^0 to SMD clusters.44
Figure 3.2: A typical SMD cluster scheme.46
Figure 3.3: The floor settings of SMD to find seed strips.46
Figure 3.4: Double-counted strip example from simulation48
Figure 3.5: The invariant mass spectrum of neutral pion49
Figure 3.6: SMD-only points distributed in the SMD U and V cluster network.52
Figure 3.7: Zratio distribution between U and V shared clusters.53

Figure 3.8: Shared cluster example.54
Figure 4.1: The energy fraction within two neighboring towers with fixed $\eta=1.541$61
Figure 4.2: The energy fraction within two neighboring towers with fixed $\phi=0$61
Figure 4.3: Energy sharing among nearby towers along η and ϕ63
Figure 4.4: Energy distributions from far sides of tower boundaries.64
Figure 4.5: The uniform distribution in the EEMC in the energy response study.66
Figure 4.6: ratio=reconstructed energy/generated energy.67
Figure 4.7: Plots of linearity between reconstructed and generated energies.68
Figure 4.8: The ratio between energies as a function of generated η69
Figure 4.9: The ratio between energies as a function of generated energy.70
Figure 4.10: π^0 eta dependence of sampling fraction in the EEMC based on figure 4.8	.71
Figure 4.11: π^0 energy dependence of sampling fraction based on figure 4.9.72
Figure 4.12: Generated Z vertex for the di- π^0 sample77
Figure 4.13: Geometry distribution in the EEMC for the di- π^0 sample77
Figure 4.14: Generated energy distribution of the di- π^0 sample78
Figure 4.15: Generated transverse momentum distribution of the di- π^0 sample78
Figure 4.16: Invariant mass spectra peaked at 0.135 GeV from the di- π^0 sample.79
Figure 4.17: Distance ΔR between the reconstructed π^0 and the closer generated π^0 . .	.82
Figure 4.18: η distributions from the match between reconstructed and generated π^0 . .	.83
Figure 4.19: ϕ distributions from the match between reconstructed and generated π^0 . .	.83
Figure 4.20: p_T distributions from the match between reconstructed and generated π^0 .	.84
Figure 4.21: E distributions from the match between reconstructed and generated π^0 . .	.84
Figure 4.22: The π^0 reconstruction efficiency vs transverse momentum in GeV/c.86

Figure 4.23: The π^0 reconstruction efficiency vs energy in GeV.86
Figure 4.24: Generated geometry for the single π^0 sample.87
Figure 4.25: Invariant mass spectra peaked at 0.135 GeV from the single π^0 sample.88
Figure 4.26: Distance between the reconstructed and the generated π^089
Figure 4.27: The π^0 reconstruction efficiency vs p_T in GeV/c for the single π^0 sample90
Figure 4.28: The π^0 reconstruction efficiency vs E in GeV for the single π^0 sample90
Figure 4.29: Generated π^0 z vertex distribution from the pythia sample91
Figure 4.30: Generated π^0 geometry for the pythia event sample.92
Figure 4.31: Distance between the reconstructed and generated π^0 from the pythia.93
Figure 4.32: The π^0 reconstruction efficiency at vs p_T in GeV/c for pythia events94
Figure 4.33: The π^0 reconstruction efficiency at vs E in GeV for pythia events95
Figure 5.1: The 2002 layout of EEMC trigger patches97
Figure 5.2: Abnormal SMD clusters example1	100
Figure 5.3: Abnormal SMD clusters example2	101
Figure 5.4: Abnormal SMD clusters example3	101
Figure 5.5: The ET distribution from all point candidates for run 7078044.	103
Figure 5.6: π^0 p_T versus invariant mass from run 7078044.	104
Figure 5.7: π^0 pseudo-rapidity versus invariant mass from run 7078044.	104
Figure 5.8: π^0 energy-sharing versus invariant mass from run 7078044.	105
Figure 5.9: The separation distribution between two seed strips from the same plane	106
Figure 5.10: Spatial distributions between the pairs of points in π^0 candidates.	106
Figure 5.11: Sector-dependent SMD seed distribution	107
Figure 5.12: The π^0 yield as a function of η vs invariant mass	110

Figure 5.13: The π^0 yield as a function of ϕ vs invariant mass110
Figure 5.14: The π^0 yield as a function of η with mass cut [0.08, 0.18] GeV.	111
Figure 5.15: The π^0 yield as a function of ϕ with mass cut [0.08, 0.18] GeV.111
Figure 5.16: The ϕ yield of π^0 s by rescaling the EEMC every 30 degrees.112
Figure 5.17: The bivariate Gaussian distributions of the tower ET cuts.114
Figure 5.18: The π^0 peak asymmetry parameter k from fits at different pT bins.116
Figure 5.19: Invariant mass spectrum fit from pT bin [8, 9] GeV/c.117
Figure 5.20: The invariant mass spectrum of reconstructed π^0 s from 347 runs.119
Figure 5.21: Reconstructed π^0 pT vs invariant mass from 347 runs119
Figure 5.22: Reconstructed π^0 distribution in the endcap.120
Figure 5.23: Vertex and ET distribution from real data120
Figure 5.24: Independent fits to the four spin-dependent invariant mass spectra.123
Figure 5.25: Final fits on the invariant mass spectra for the pT bin [4, 5) GeV/c.123
Figure 5.26: Same as figure 5.25, but for the [5, 6) GeV/c pT bin.124
Figure 5.27: Same as figure 5.25, but for the [6, 7) GeV/c pT bin.124
Figure 5.28: Same as figure 5.25, but for the [7, 8) GeV/c pT bin.125
Figure 5.29: Same as figure 5.25, but for the [8, 9) GeV/c pT bin.125
Figure 5.30: Same as figure 5.25, but for the [9, 10) GeV/c pT bin.126
Figure 5.31: Same as figure 5.25, but for the [10, 25] GeV/c pT bin.126
Figure 5.32: Result of fit to the pythia event sample with partonic pT [11, 15] GeV/c. .	.128
Figure 5.33: Width $\sigma=p_4$ comparison between the pythia sample and real data.129
Figure 5.34: Peak $\mu=p_3$ comparison between the pythia sample and real data.129
Figure 5.35: The single spin asymmetry for the yellow beam along pT.134

Figure 5.36: The single spin asymmetry for the blue beam along pT.135

Figure 5.37: The background asymmetry from a side band analysis.136

Figure 5.38: Double spin asymmetry A_{LL} with statistical error bars along pT bins. . .137

Chapter 1

Introduction

1.1 Thesis outline

In this thesis, we will report the preliminary result of neutral pion double spin asymmetry in the Endcap Electro-Magnetic Calorimeter (EEMC) from the STAR experiment. Details about the STAR experiment, physics motivation, detectors, software development, simulation studies and data analyses for neutral pions will be discussed in different chapters in the thesis. Conclusion will be made finally.

In Chapter 1, we will present an overview of the RHIC spin program and the STAR experiment in very high energy situation. Then we will explain why we are interested in probing the double spin asymmetry from neutral pion production in the forward-rapidity region, because it can provide us with strong constraints on the gluon's contribution to the proton's spin structure.

In Chapter 2, we will introduce the STAR detector system. The hardware system includes many complex detectors. We will introduce the detail of the Endcap Electro-Magnetic Calorimeter in the near forward-rapidity region of STAR, because this is the most important hardware we are employing for this thesis. Other sub-systems will be presented briefly.

In Chapter 3, we are introducing the endcap pion finder software development along with the physics process of neutral pions in the specific EEMC mechanical structure. A neutral pion will generally decay into two photons in the endcap, the energy deposition and position information can be obtained according to our special-designed Shower

Maximum Detectors (SMD). For each photon point candidate, the energy is decided by the corresponding tower cluster. A pion candidate will be reconstructed after a set of selection criteria. Please check Chapter 3 and appendix C for the software detail information.

In Chapter 4, we report the simulation study result. There will be three main topics in the simulation study: energy sharing among the EEMC adjacent towers, neutral pion energy deposition study in the EEMC, and pion reconstruction and efficiency study. This part plays an very important role in the early stage of our development for the pion finder software package, and help us optimize our software framework and setup for real data analyses.

In Chapter 5, we report the real data analysis result from the run6 (2006) longitudinally polarized proton-proton runs. We select the data from run6 with successful finishing note marked online at STAR computing page, and require selected runs having valid luminosity information from the luminosity file for run6 data. We optimize the parameter setup in the pion software and generate neutral pion trees. Based on these trees and luminosity information, we normalize the yield by four spin-dependent states and grab the yield by proper fitting procedure. The double spin asymmetry is then calculated with statistical uncertainties and systematic errors.

In Chapter 6, based on the statistical analyses from our calculating results and theory prediction, we believe that our double spin asymmetry results rule out large ΔG situation for the gluon's contribution in a proton's spin structure.

1.2 RHIC spin program overview

During the past few decades, scientists around the world have been probing the inner structure of the proton's spin. The Relativistic Heavy Ion Collider (RHIC) was constructed at Brookhaven National Lab (BNL) to study high energy collisions of polarized protons, gold ions, etc. The Solenoidal Tracker At RHIC (STAR) is one of two large detectors at BNL that is being used to investigate the gluon's polarization ΔG , and determine whether the gluon carries part of the proton's spin. The IUCF STAR spin group built and installed the Endcap ElectroMagnetic Calorimeter (EEMC) in the forward rapidity region of STAR to detect the produced jets of particles, particularly gamma rays and neutral mesons, which will give us important information about the nature of the proton's spin.

Compared to the polarized beam energy of 24.6 GeV in the AGS in the 1990's, the current achievement in polarized proton beams at the RHIC rings reaches to a high center of mass energy $\sqrt{s}=200$ GeV, with "blue" and "yellow" beams of 100 GeV energy separately. The total integrated luminosity for the 2006 longitudinal runs was about 8.5 pb^{-1} . The plan is to provide a center of mass energy up to $\sqrt{s}=500$ GeV and higher luminosity of the polarized proton beams in the future. With the high quality proton beam collisions so far and in the future, we can carry out a good statistical and systematic study of jet and neutral pion production at high transverse momentum p_T , where we can compare our results reliably with predictions from next to leading order perturbative quantum chromo-dynamics (NLO pQCD).

In this thesis, we report the first measurement of the inclusive neutral pion double spin asymmetry A_{LL} from the EEMC at STAR at a center of mass energy $\sqrt{s}=200$ GeV,

by analyzing the run6 (2006) longitudinally polarized proton-proton data. In this chapter, we will give the theoretical explanation for the physics motivation of doing this research.

1.3 Physics Motivation

A primary goal of the RHIC spin program is to understand the gluon's spin contribution ΔG to the proton's spin structure. The STAR detector and the RHIC facility (see Chapter 2) are well designed to do this research. But why do we want to understand ΔG in the proton's spin structure? How can we relate the double spin asymmetry of the inclusive neutral pion production to ΔG ? And how can the longitudinally polarized proton-proton collisions studied with STAR allow us to determine these asymmetries? The answers to these three questions will give us the physics motivation for this study.

1.3.1 Origin of the proton's spin

In a simple model of the nucleon, the proton's spin structure can be decomposed into four parts: the quark's spin contribution, the gluon's spin contribution, and the orbital angular momentum of the quarks and gluons. We express this as the following formulae,

$$S_z = \frac{1}{2} \Delta \Sigma + \Delta G + L_z^q + L_z^G = \frac{1}{2} \quad (1.1)$$

In this equations, $\Delta \Sigma$ is the quarks' polarization in the proton, $\Delta \Sigma = \sum_{i=1}^{n_f} \Delta q_i$, where $\Delta q_i = \int_0^1 [q_i^+(x) + \bar{q}_i^+(x) - q_i^-(x) - \bar{q}_i^-(x)] dx$, is summed over all quark flavors. ΔG is the fraction of the proton's spin carried by the gluons' spin. (a primary focus of the RHIC spin program); and L_z^q and L_z^G are the orbital angular momentum contributions from

quarks and gluons respectively. We can describe the components in formula (1.1) in a more general way by using the quark/anti-quark and gluon helicity parton distribution functions (PDFs):

$$\Delta f_i(x, Q^2) \equiv f_i^+(x, Q^2) - f_i^-(x, Q^2) \quad (1.2)$$

where $f_i^+(x, Q^2)/f_i^-(x, Q^2)$ denotes a type i partonic distribution with positive/negative helicity in the proton, and x is the Bjorken parameter which denotes the momentum fraction carried by the parton in the nucleon. Q^2 denotes the hard scale. The integral

$$\Delta f_i(Q^2) = \int_0^1 \Delta f_i(x, Q^2) dx = \int_0^1 (f_i^+(x, Q^2) - f_i^-(x, Q^2)) dx \quad (1.3)$$

gives us the spin contribution to the proton from type i parton. The orbital angular momentum contributions are unknown so far. We will focus our discussion on the quarks' and gluons' contributions.

There have been many studies of the polarized parton distributions of the nucleon in recent years [1-25], carried out by world-wide collaborations and programs, such as SLAC-Yale E80, CERN-EMC, CERN-SMC, SLAC-E143, SLAC-E142, DESY-HERMES, SLAC-E154, and SLAC-E155. Polarized deep inelastic scattering (pDIS) of leptons on nucleons has been the most widely used method for probing the partonic contributions in the nucleon. These experimental data give us more and more information about the quark, anti-quark, and the gluon polarizations. The basic conclusion from these pDIS experiments is that only a small fraction of the proton's spin is carried by quarks and anti-quarks [26-37], which means $1/2\Delta\Sigma$ is small compared to the proton's total spin of $1/2$. Several theoretical groups, such as E. Leader, A.V. Sidorov, and D.B. Stamenov (LSS), M. Glueck, E. Reya, M. Stratmann, and W. Vogelsang (GRSV), T. Gehrmann and W.J. Stirling (GS), J. Bluemlein and H. Boettcher (BB), Asymmetry Analysis

Collaboration (AAC), D.de. Florian and R. Sassot (DS), D.de. Florian, G.A. Navarro and R. Sassot (DNS), G. Altarelli, R. Ball, S. Forte and G. Ridolfi (ABFR), etc, have attempted to extract the polarized parton distribution functions (PDFs) from the experimental measurements, and have verified this conclusion [38]. Several recent analyses will be shown below.

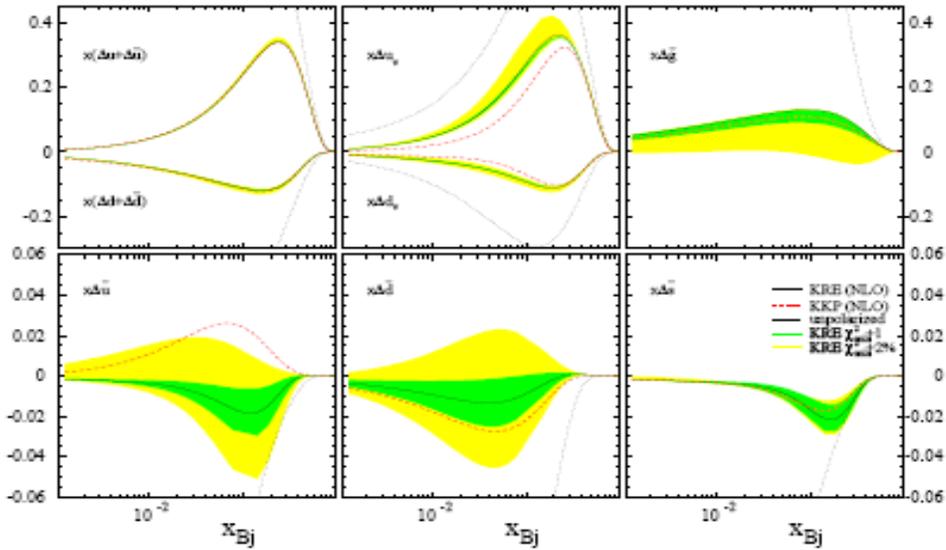


Figure 1.1: Parton densities and their uncertainties at $Q^2 = 10\text{Gev}^2$ (2005).

Figure 1.1 shows a recent parton densities study from the DNS group [39]. The partonic contribution from each flavor quark or gluon, multiplied by the Bjorken parameter x_{Bj} is prescaled. The green and yellow uncertainty bands correspond to $\Delta\chi^2=1$ and $\Delta\chi^2=2\%$.

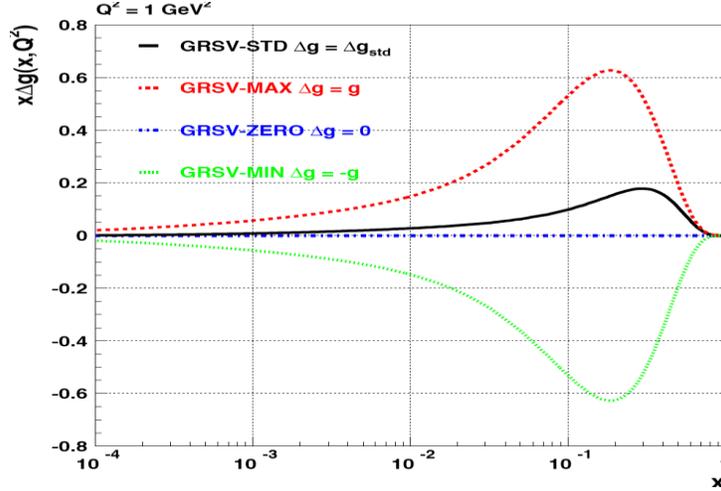


Figure 1.2: Gluon densities from different theoretical assumptions by GRSV (2001).

Figure 1.2 shows the gluon's helicity distribution in x from the GRSV study [40]. The four curves in figure 1.2 represent four different theoretical assumptions, $\Delta G = \text{std}$ (best fit to DIS 2001) (GRSV-STD), $\Delta G = g$ (GRSV-MAX), $\Delta G = -g$ (GRSV-MIN) and $\Delta G = 0$ (GRSV-ZERO). From this detailed plot, we can see that $x\Delta G$ varies significantly, especially when the momentum fraction in the proton carried by the gluon is larger than 0.01. Comparing the upper-left and upper-right plots of figure 1.1, it is clear that the quark, and anti-quark contributions have a much smaller uncertainty band than the gluon one, which is consistent with what we see in the variation of ΔG on figure 1.2. Theorists are in agreement that ΔG is poorly determined by the pDIS data alone.

D.d. Florian, R. Sassot, M. Stratmann and W. Vogelsang (DSSV) have done the latest global analysis of parton helicity densities and their uncertainties for the nucleon [1]. We cite a table below.

Table 1.1: Global QCD analysis for first moments at $Q^2=10 \text{ Gev}^2$

	$x_{\min} = 0$	$x_{\min} = 0.001$	
	best fit	$\Delta\chi^2 = 1$	$\Delta\chi^2/\chi^2 = 2\%$
$\Delta u + \Delta \bar{u}$	0.813	0.793 $^{+0.011}_{-0.012}$	0.793 $^{+0.028}_{-0.034}$
$\Delta d + \Delta \bar{d}$	-0.458	-0.416 $^{+0.011}_{-0.009}$	-0.416 $^{+0.035}_{-0.025}$
$\Delta \bar{u}$	0.036	0.028 $^{+0.021}_{-0.020}$	0.028 $^{+0.059}_{-0.059}$
$\Delta \bar{d}$	-0.115	-0.089 $^{+0.029}_{-0.029}$	-0.089 $^{+0.090}_{-0.080}$
$\Delta \bar{s}$	-0.057	-0.006 $^{+0.010}_{-0.012}$	-0.006 $^{+0.028}_{-0.031}$
Δg	-0.084	0.013 $^{+0.106}_{-0.120}$	0.013 $^{+0.702}_{-0.314}$
$\Delta \Sigma$	0.242	0.366 $^{+0.015}_{-0.018}$	0.366 $^{+0.042}_{-0.062}$

The latest global analysis [1] of parton helicities in the nucleon in the above table 1.1 shows that $\Delta\Sigma$ is around 0.2~0.3, and is constrained very well, while for the gluon's contribution ΔG , the uncertainty is relatively large compared to the value calculated, and is also dependent on the integral range of the Bjorken parameter x .

From the same global analysis, we show the following plot:

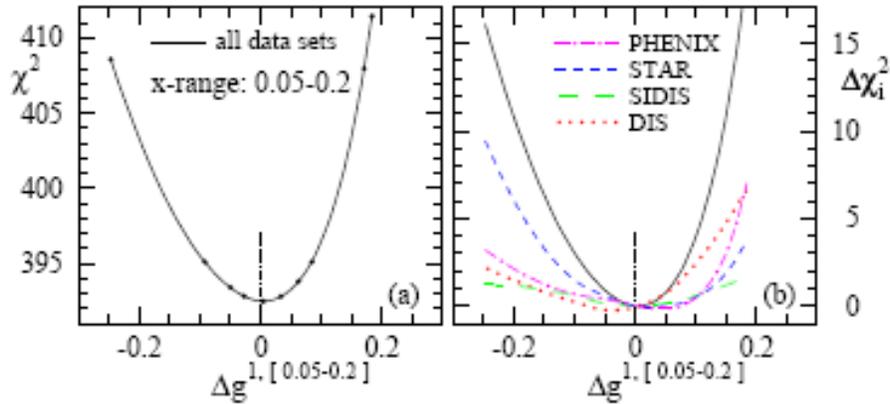


Figure 1.3: ΔG variations from different data sources at $Q^2 = 10 \text{ Gev}^2$ [1].

By adding the new data from PHENIX, STAR, SIDIS and DIS, theorists show the ΔG contribution on the above plot. This suggests that ΔG is small in the accessible range $0.05 < x < 0.2$ of momentum fraction, but it is important to use other probes and explore other ranges of x in order to better understand ΔG in the proton's spin structure.

1.3.2 Double Spin Asymmetry A_{LL} in ΔG study

From the above, we see that it is very important for the RHIC spin program to provide more constraints on ΔG . In proton-proton collisions at RHIC, jets are produced and detected in large numbers. From the double spin asymmetry of the inclusive jet yields, which can be measured directly, we can test theoretical predictions for ΔG . There is a complication in this process, however, because the experimentalists and the theorists must agree on how to define a "jet" in order to make these comparisons.

We can also obtain useful information on ΔG by measuring the double spin asymmetry A_{LL} for inclusive neutral pions, using analysis techniques similar to those used for jets. Pions are not produced directly in the hadronic interactions at RHIC, but are created as the outgoing partons fragment. In this case, one avoids the difficulty of needing to define a jet, but in order to make theoretical predictions, one needs to understand the details of the fragmentation process. This can not be calculated using pQCD, and theorists rely on experimental measurements of how often, and with what energy, pions are produced in jets. If these details are understood, and the pions can be detected and reconstructed efficiently, then measuring A_{LL} for neutral pions will help constrain ΔG .

Replacing the gluon role into formulae (1.2) and (1.3), we can get:

$$\Delta G(Q^2) = \int_0^1 \Delta G(x, Q^2) dx = \int_0^1 [G^+(x, Q^2) - G^-(x, Q^2)] dx \quad (1.4)$$

where G^+ or G^- denotes the gluon polarization in a proton either parallel or anti-parallel to the proton's longitudinal polarization. The unpolarized gluon distribution in a proton is expressed by:

$$G(x, Q^2) = G^+(x, Q^2) + G^-(x, Q^2) \quad (1.5)$$

The actual gluon polarization is then defined as: $\frac{\Delta G(x, Q^2)}{G(x, Q^2)}$.

The measured double spin asymmetry is defined from the spin dependence of the reaction cross section:

$$A_{LL} \equiv \frac{d\Delta\sigma}{d\sigma} \equiv \frac{d\sigma^{++} - d\sigma^{+-}}{d\sigma^{++} + d\sigma^{+-}} \quad (1.6)$$

For the high transverse momentum p_T collisions in polarized pp scattering at RHIC, we can describe the cross section by [46]:

$$d\Delta\sigma = \sum_{ab} \int dx_a \int dx_b \Delta f_a(x_a, Q^2) \Delta f_b(x_b, Q^2) \times d\Delta\hat{\sigma}_{ab}(x_a, x_b, p_T, \alpha_s(Q^2), p_T/Q) \quad (1.7)$$

where a and b denote the type of interacting partons, and $d\Delta\hat{\sigma}_{ab}$ denotes the corresponding spin-dependent partonic cross section. If we now consider the spin effects from pQCD for the process-specific spin correlation coefficient [41], $\hat{\alpha}_{LL}^{proc}(\hat{s}, \hat{t}, \hat{u})$, expressed in terms of the Mandelstam variables $(\hat{s}, \hat{t}, \hat{u})$ for partonic hard scattering, we can relate the measured and partonic spin correlations as:

$$A_{LL} \approx \frac{\Delta f_a(x_1, Q^2)}{f_a(x_1, Q^2)} \frac{\Delta f_b(x_2, Q^2)}{f_b(x_2, Q^2)} \hat{\alpha}_{LL}^{proc}(\hat{s}, \hat{t}, \hat{u}) \quad (1.8)$$

where $\Delta f/f$ denotes the partonic polarizations, and a and b denote the colliding partons in the process of interest.

At RHIC, we can study longitudinally polarized pp collisions. In this case, the primary source of direct photons is mainly from the QCD Compton scattering from the quark-gluon sub-process interaction [44]. Formula (1.8) can be written more specifically as :

$$A_{LL} \approx \frac{\Delta G(x_g, Q^2)}{G(x_g, Q^2)} A_1^p(x_q, Q^2) \hat{\alpha}_{LL}^{Compton}(\theta^*) \quad (1.9)$$

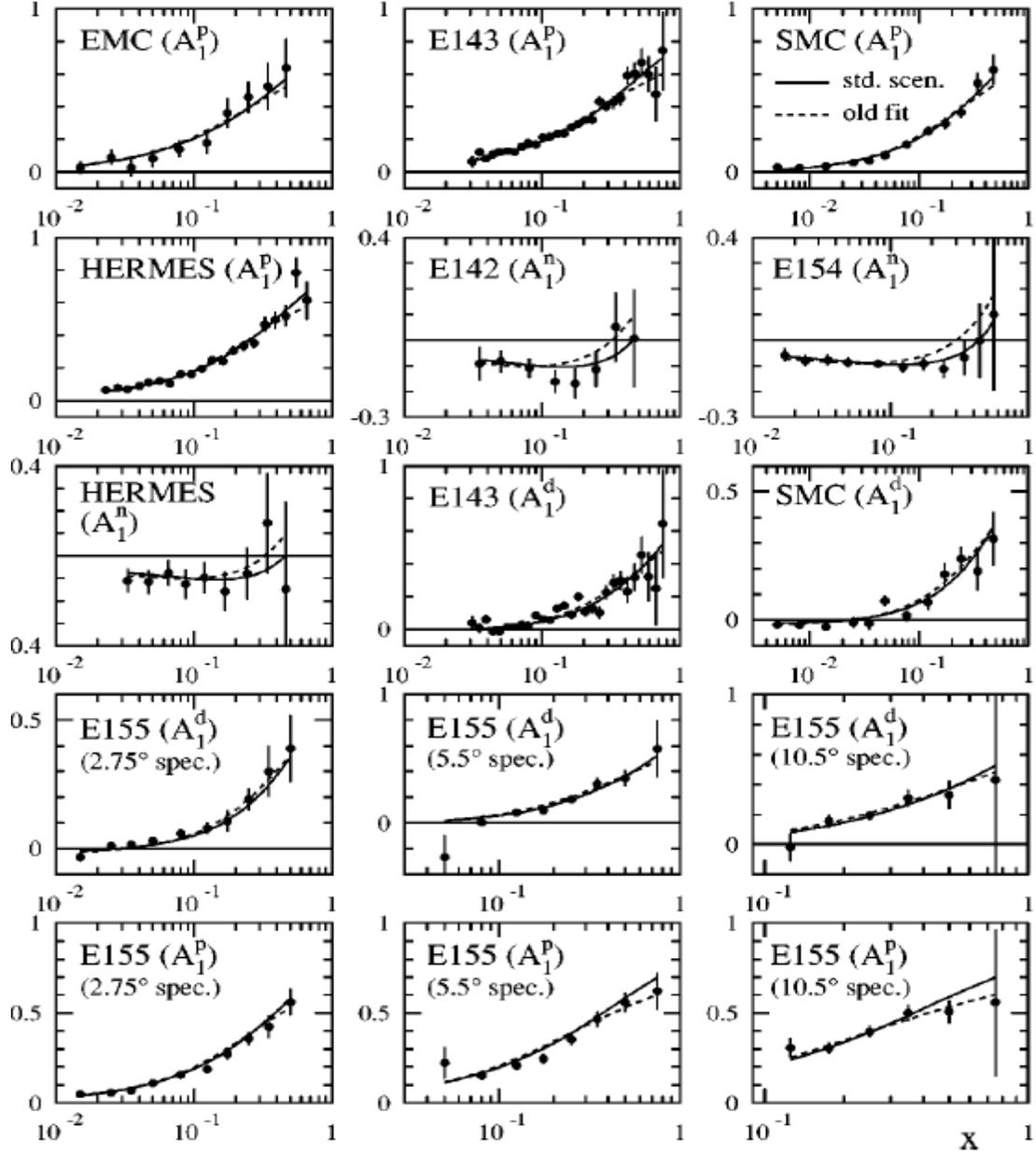


Figure 1.4: $A_1^N(x_q, Q^2 = 5\text{GeV}^2)$ from pDIS GRSV2000 [40].

With this formula (1.9), we can see the direct relation between the measured double spin asymmetry and the gluon's polarization in our experiment. The quark spin asymmetry $A_1^p(x_q, Q^2)$ is measured in the polarized deep inelastic lepton-nucleon scattering experiments; the GRSV2000 [40] analysis of these measurements is shown in figure 1.4.

From figure 1.4 we can see that scientists have done a lot of work on the measurements of the quark spin asymmetry, and very good progress has been made. The third term on the right hand side of formula (9) is the process-specific spin-correlation coefficient in gluon–quark Compton scattering, which is predicted at high energies by perturbative Quantum Chromo-Dynamics (pQCD) [45].

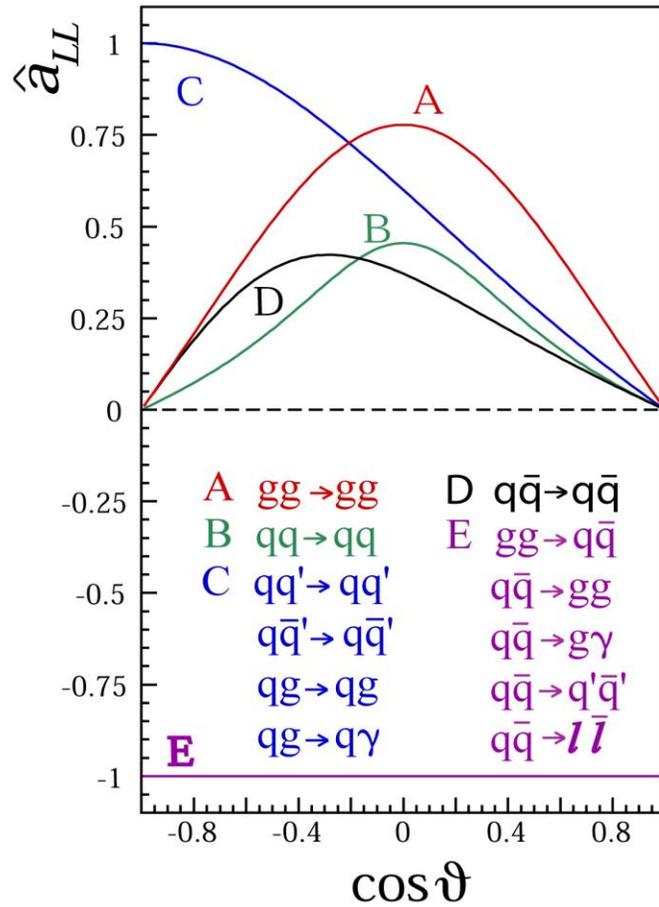


Figure 1.5: \hat{a}_{LL} distributions from all possible parton reactions

Figure 1.5 shows the process-specific spin-correlation coefficient distributions for different parton reactions. In our experiment, the main processes of interest are described by curve C. We will discuss more about this later.

We have fully discussed two terms on the right hand side of formula (1.9): the quark spin asymmetry $A_1^p(x_q, Q^2)$ and the process-specific spin-correlation coefficient $\hat{\alpha}_{LL}$. Since these two terms are well understood, we can now see that research on the double spin asymmetry A_{LL} is very useful if we want to better understand ΔG .

In our experiment, A_{LL} is an observable which can be calculated from the spin-dependent yields of neutral pions. Formula (1.6) can be expressed as:

$$A_{LL} = \frac{\sigma_{++} - \sigma_{+-}}{\sigma_{++} + \sigma_{+-}} = \frac{1}{P_y * P_b} * \frac{\left(N_{++} + \frac{L_{++}}{L_{--}} N_{--} \right) - \left(\frac{L_{++}}{L_{+-}} N_{+-} + \frac{L_{++}}{L_{-+}} N_{-+} \right)}{\left(N_{++} + \frac{L_{++}}{L_{--}} N_{--} \right) + \left(\frac{L_{++}}{L_{+-}} N_{+-} + \frac{L_{++}}{L_{-+}} N_{-+} \right)} \quad (1.10)$$

where P_y and P_b denote the beam polarizations for the yellow and blue beam; ++, +-, -+ and -- denote the four spin states from the two polarized proton beams; N_{++} , N_{+-} , N_{-+} and N_{--} denote the four spin-dependent neutral pion yields; and L_{++} , L_{+-} , L_{-+} and L_{--} denote the four spin-dependent luminosities. This means we can measure the double spin asymmetry from our experiment by performing several different measurements, as we discuss below.

1.3.3 The Neutral pion yield from proton-proton collisions

In a high-energy proton-proton collision, the hadronic reaction can be described in terms of its long-distance and short-distance contributions [46]. The long-distance parts are from the structure of the nucleon in terms of its parton' distributions, which describe the universal properties of the nucleon. The short-distance parts, on the other hand, describes the interactions of the partons, and can be calculated by using perturbative

Quantum Chromo-Dynamics (pQCD) theory, which means the short-distance ones are process-dependent.

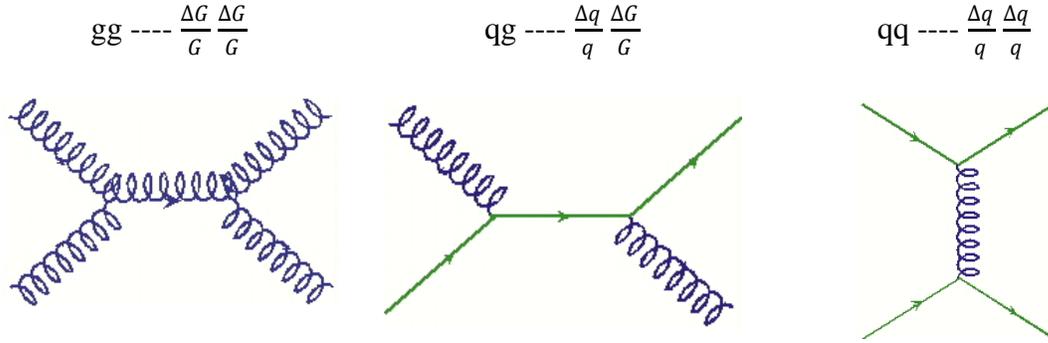


Figure 1.6: Some Feynman diagrams of partonic interactions in a high energy pp collision.

Three main sub-processes are involved in the hard partonic interactions during a high energy pp collision: the gluon-gluon, quark-gluon and quark-quark interactions. In the RHIC spin program, we want to know the gluon's contribution to a nucleon's spin. So we are most interested in the short-distance interactions that involve a gluon. As we discussed above, π^0 production in pp collision is parton-dependent shown in figure 1.6; different sub-processes will give different neutral pion contributions in the $\vec{p}\vec{p} \rightarrow \pi^0 + X$ reaction. We study inclusive neutral pion production in the STAR experiment because high p_T π^0 production in pp collisions is a high-yield signature of hard partonic interactions, so physically it is easy to produce and detect.

Theorists have studied the theoretical cross section for inclusive neutral pion production in the reaction of $\vec{p}\vec{p} \rightarrow \pi^0 + X$. By performing a next-to-leading order perturbative Quantum Chromo-Dynamics calculation [47- 49] on a set of proton parton distributions [50] with contributions from initial gg, qg and qq states [51], theorists make the following predictions [45]:

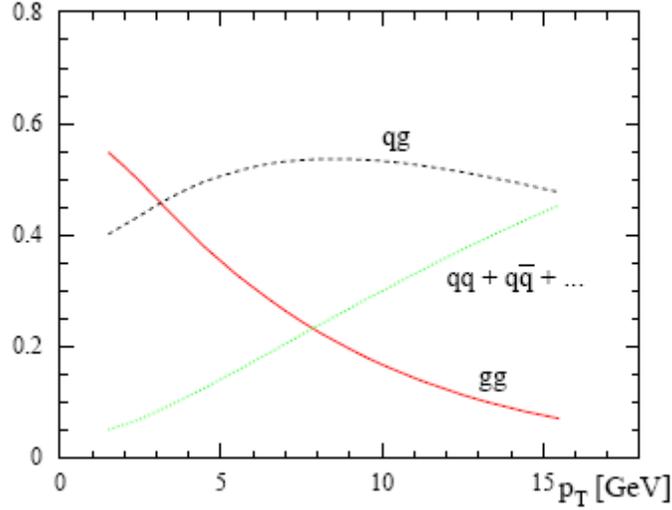


Figure 1.7: Fractional contribution of each partonic sub-process to the inclusive π^0 yield as a function of pion p_T by NLO pQCD calculations for $\sqrt{s} = 200$ GeV pp collisions at pseudo-rapidity $\eta=0$.

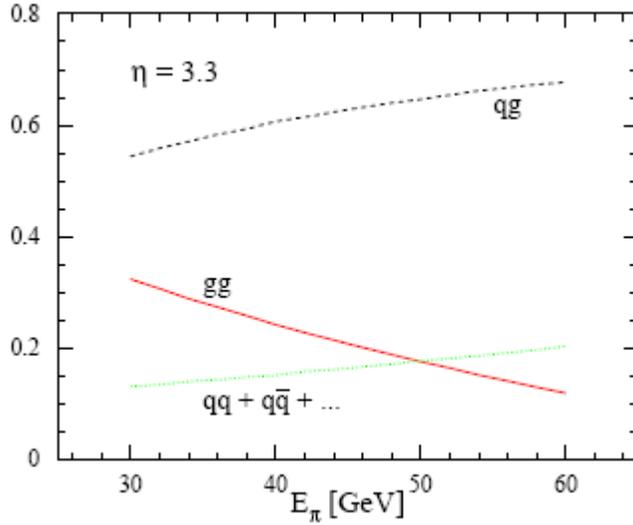


Figure 1.8: Fractional contribution of each partonic sub-process to the inclusive π^0 yield as a function of pion energy by NLO pQCD calculations for $\sqrt{s} = 200$ GeV pp collisions at the pseudo-rapidity $\eta=3.3$.

Figures 1.7 and 1.8 show the separated partonic (gg, qg, qq) contributions to inclusive π^0 production in proton-proton collisions $\vec{p}\vec{p} \rightarrow \pi^0 + X$ at the center of mass

energy $\sqrt{s} = 200$ GeV in the mid- and forward-rapidity regions of the STAR detectors, respectively. From the two plots above we can see that the qg (quark-gluon) sub-process dominates the reaction, which means that inclusive neutral pion production in the STAR experiment is sensitive to gluon's helicity preferences. In other words, the double spin asymmetry A_{LL} for inclusive π^0 s is sensitive to the gluon's polarization in the STAR kinematics. Since our experiment is a high momentum-transfer reaction, it is useful to choose the transverse momentum that will optimize our analysis of inclusive π^0 . From figure 1.7 we can see that the π^0 's greatest sensitivity to gluon polarization is in the intermediate p_T range (~ 5 - 15 GeV/ c).

From the above three sections about the physics motivation for the inclusive neutral pion analysis at STAR, we can conclude: first, polarized p+p collisions at RHIC allow us to study the proton's spin structure with strongly interacting probes because of the sensitivity to gluon polarization in the nucleon; and second, measuring A_{LL} , and using pQCD and previous results from pDIS, combine to provide strong constraints on ΔG .

1.4 Inclusive neutral pion analysis overview at RHIC

As we discussed previously, we report here the inclusive neutral pion double spin asymmetry A_{LL} result for the first time from the EEMC at STAR. The results were obtained at the center of mass energy of $\sqrt{s}=200$ GeV by analyzing the run6 (2006) longitudinally polarized proton proton data. There are several other experimental programs going on at RHIC that also involve inclusive neutral pion analyses. Here we will give a very brief introduction to these programs.

STAR is a huge collaboration with many detector subsystems, as we discussed in Chapter 2. The Barrel EMC [52] is also one of the main calorimeters. Our Barrel EMC colleagues perform inclusive neutral pion analysis in a different pseudo-rapidity range of $-1 \leq \eta \leq 1$. The BEMC “sees” the same polarized proton beams as we do, and our colleagues have shown their preliminary results from the 2005 and 2006 data at DIS meeting [53].

Another group at STAR reported their inclusive neutral pion cross section analysis for 2003 d+Au data at a center of mass energy $\sqrt{S_{NN}}=200$ GeV. They used only half of the BEMC, with pseudo-rapidity range $0 \leq \eta \leq 1$ [54].

PHENIX is another big collaboration at RHIC. They have reported inclusive neutral pion results since 2004 [55] using the longitudinally polarized proton proton collisions. Their results are consistent with theoretical expectations. PHENIX, however, can only do π^0 analysis over a narrow range of pseudo-rapidity, $|\eta| < 0.30$.

Chapter 2

Experimental Setup

2.1 Polarized Proton Collider at RHIC

The Relativistic Heavy Ion Collider (RHIC) provides a user facility for the study of spin physics [57]. Figure 2.1 shows how the scientists and engineers designed the accelerators and storage rings for polarized protons at RHIC, with highlighted details such as the ion source, Siberian snakes, beam energies and polarization, and collider luminosities. The complex is a frontier facility in the high energy spin physics world in handling polarized ion beams in synchrotrons and storage rings. We will discuss the working functions of the complex step by step in the following.

The polarized proton beam is accelerated from a recently replaced optically pumped polarized ion source (OPPIS) H^- , which was constructed from the KEK OPPIS source [58]. The polarized ion source produces a $300 \mu\text{s}$ pulse with at least $0.5 \text{ mA } H^-$ ion current with 80% polarization, which means about 9×10^{11} polarized H^- per pulse. A future goal is to improve the source intensity to over 10 mA current, and a polarization of 90%, in $100 \mu\text{s}$ pulses with production frequency 7.5 Hz [59]. This should increase the beam polarization in the Alternating Gradient Synchrotron (AGS) and RHIC to 65 – 70%.

The polarized H^- ions are accelerated to 200 MeV with a radio-frequency quadrupole (RCF) accelerator and a 200 MHz Linear Accelerator (LINAC), which was built in the late 1960's. As the H^- ion pulse goes into the AGS, the polarized proton injector to RHIC, it is strip-injected and constrained into a single polarized proton bunch in the AGS Booster. The polarized proton beam is accelerated to 1.5 GeV in the AGS

Booster, then is transferred to the AGS and undergoes another increase in energy up to 25 GeV.

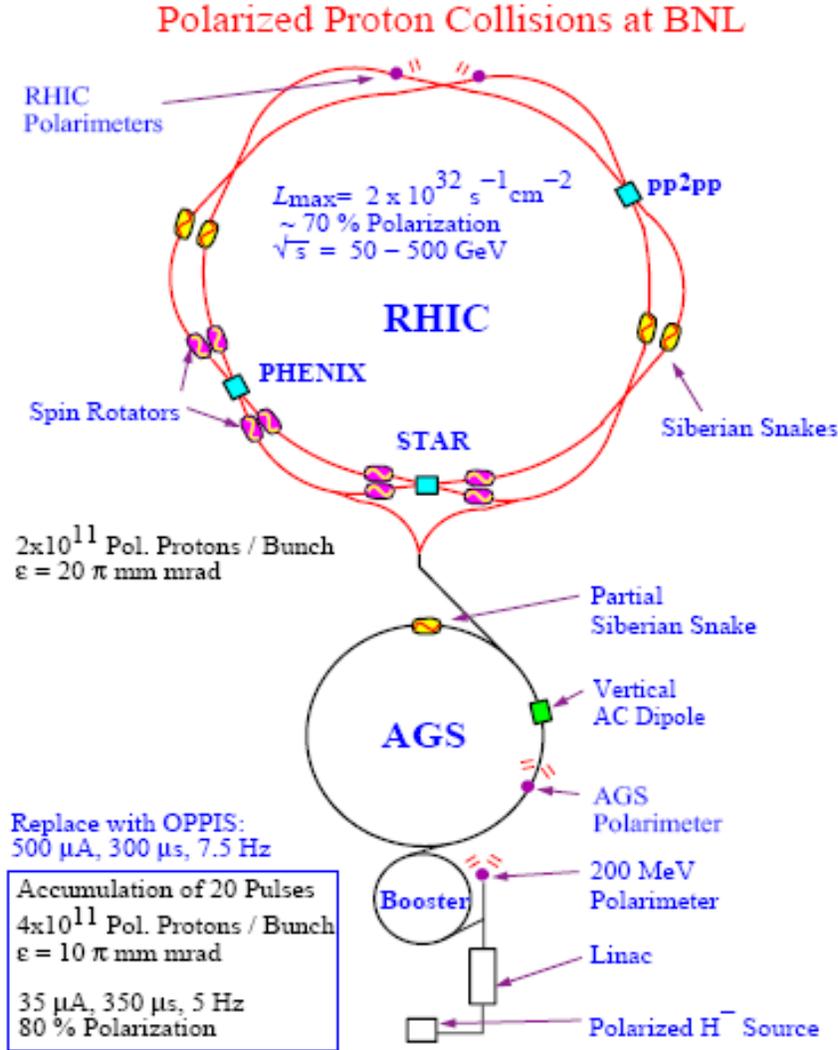


Figure 2.1: The accelerator complex at BNL, which includes the important elements for the acceleration of polarized protons: The polarized proton source, the 200 MeV Linac and its polarimeter, the AGS Booster, the AGS and RHIC. To run the RHIC spin program, two “Siberian snakes” and four spin rotators are installed for each detector [56].

To inject polarized proton beams from the AGS into RHIC with optimal polarization transfer and at the proper injection energy, the AGS to RHIC (AtR) transfer line has been developed [60]. The proton beam’s polarization transfer efficiency at the RHIC injection

point is dependent on the proton energy [56]. The injection energy of the proton beams can be varied from 20.58 GeV to 28.3 GeV [61].

Depolarizing resonances have long been a severe problem during the acceleration of polarized beams in a circular accelerator. Whenever the spin precession frequency of the polarized proton beam matches (actually any rational fraction) that of the oscillations of the focusing magnetic fields, a depolarizing resonance occurs. So there are always a lot of depolarizing resonances during acceleration, especially in an experiment with high energy polarized proton beams. Imperfection resonances and intrinsic resonances are the two main types of depolarizing resonances. But for acceleration to high energies at RHIC, the traditional techniques used in the AGS to overcome these resonances, such as betatron tune jump and harmonic corrections of the vertical orbit [62], are not effective ways to solve these problems. A new technique, the ‘Siberian Snake’ [63], has been introduced to overcome the large number of depolarizing resonances at RHIC. The Siberian snake rotates the spin vector and reverses the beam polarization direction each orbit as the beam passes through the snakes. This idea was first experimentally verified at the Indiana University Cyclotron Facility (IUCF) Cooler Ring [64]. At RHIC, two full Siberian Snakes are applied on opposite sides of the RHIC rings to overcome the depolarizing resonances.

In addition, two spin ‘rotators’ are installed on each side of the STAR and PHENIX detector (see figure 2.1). These helical dipole-magnet spin rotators were proposed by V. Ptitsin and Y. Shatunov, and were funded by RIKEN in Japan [65]. A total of 8 spin rotators and 48 helical dipole magnets are placed at RHIC. For the 2006 pp data set, experimentalists used the spin rotators to change the beam spin orientations prior to

collisions, so we could “dial in” transversely or longitudinally polarized proton beams before collisions. Our analysis is based on longitudinally polarized beams.

The polarized proton beams travel around RHIC in two separate rings, called the blue beam, which moves through STAR from east to west; and the yellow beam, which moves from west to east. The polarized proton beams can be accelerated in RHIC to higher center of mass energies. For the 2006 data, this energy was $\sqrt{s}=200$ GeV. Collisions of the two polarized proton beams happen in multiple collision points, for example, at STAR, PHENIX, BRAHMS, and so on. STAR is located at the 6 o’clock position on the map of Figure 2.1.

A summary of the main RHIC accelerator and polarized proton beam parameters is given below:

Table 2.1: Main parameters of RHIC acceleration and polarized proton beams

Parameter	
Peak c.m. energy	500 GeV
Initial luminosity	$2 \times 10^{32} \text{cm}^{-2} \text{sec}^{-1}$
Interactions per crossing (60 mb)	1
Protons per bunch	2×10^{11}
Bunches per ring	120
Normalized emittance (95%)	$20 \pi \text{ mm mrad}$
β^*	1 m
Average polarization	70%
Stable Spin direction at Interaction Point	vert. or long.
RF voltage per turn	6 MV
RF harmonic number	2520
Long. emittance (95%)	0.3 eV sec
Beam Momentum Spread	2.6×10^{-4}
Beam-beam tune spread (per IR)	0.007

2.2 The STAR detector

STAR was originally envisioned to study the Quark-Gluon-Plasma (QGP) created in relativistic heavy ion collisions [68]. As research has progressed, the spin physics program has become an important priority at STAR. Because of the very large number of high momentum particles produced in central heavy ion interactions, STAR was designed to measure many observables. The double spin asymmetry of neutral pions produced in polarized pp collisions is one of the important topics we can study efficiently at STAR.

STAR is a sophisticated complex with many subsystems. The detector systems include: the Endcap Electro-Magnetic Calorimeter (EEMC), the Barrel Electro-Magnetic Calorimeter (BEMC), a Silicon Vertex Tracker (SVT), a Silicon Strip Detector (SSD), the Time Projection Chamber (TPC), a Time of Flight system (TOF), two Forward Time Projection Chambers (FTPC), Beam-Beam Counters (BBC), the Forward Pion Detector (FPD), a Photon Multiplicity Detector (PMD), and some subsystems of the TPC as shown in figure 2.2. The STAR TRiGger (TRG) and Data AcQuisition (DAQ) systems will be discussed in a separate section.

As we can see from the lower plot in Figure 2.2, the center of the STAR geometry is the nominal collision point of the polarized beams, which is defined as $z=0$ at the center and with z increasing along the beam direction from east to west. In the STAR geometry, pseudo-rapidity is an important physics concept, and is defined as:

$$\eta = -\log\left(\tan\left(\frac{\theta}{2}\right)\right) \quad (2.1)$$

where θ is the polar angle measured from the z axis. The azimuthal angle ϕ is another parameter used to identify the position of detectors.

IUCF's main hardware contribution to STAR is the EEMC, and this work will focus on neutral pion analysis from the EEMC. So we will give only a brief introduction to the other subsystems at STAR, and provide more details of the EEMC in a later section.

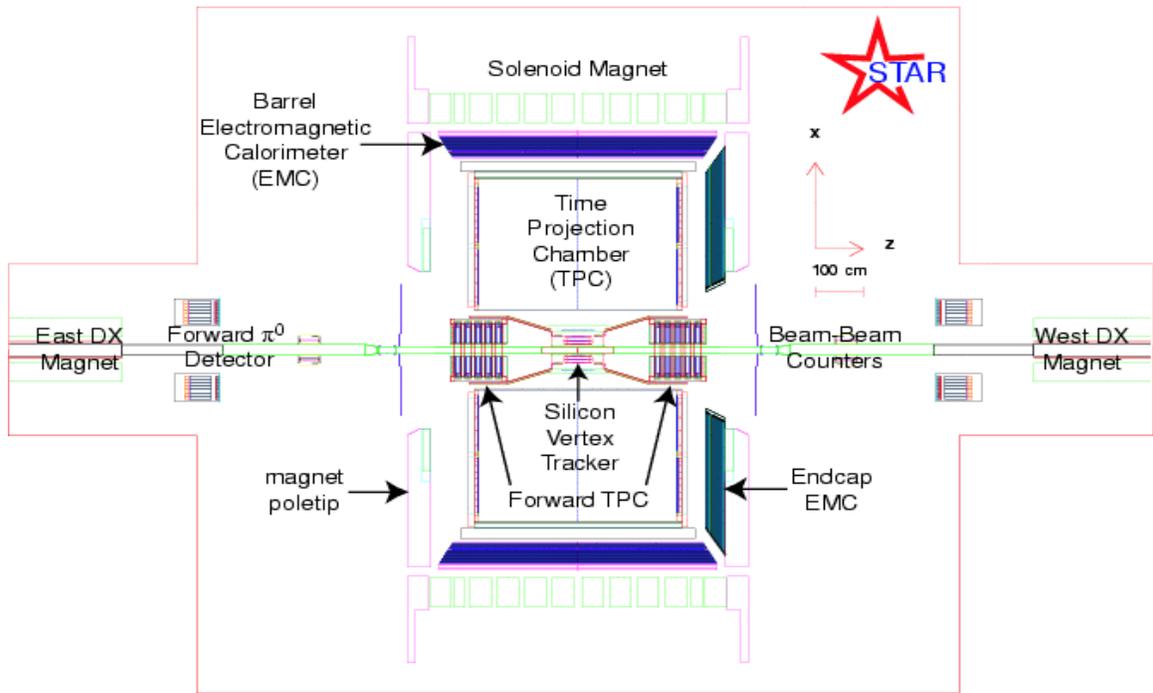
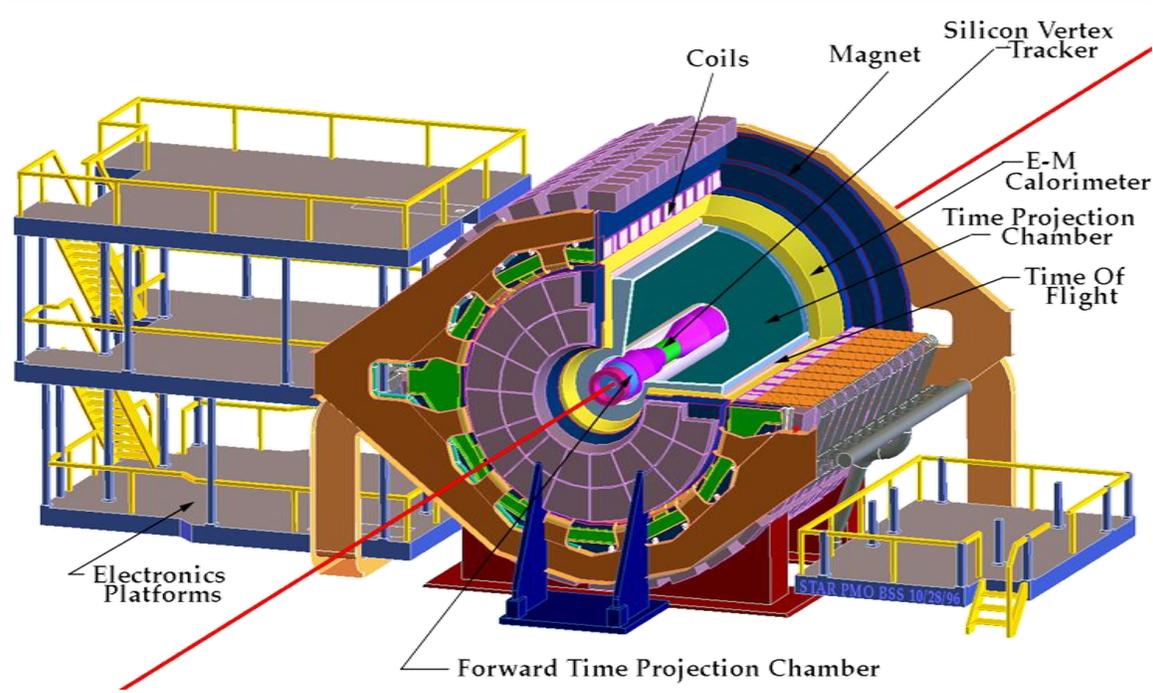


Figure 2.2: STAR detector [66] and a schematic profile of the 2006 geometry [67].

2.2.1 Magnet subsystem

The 0.5 T solenoidal magnet is a central feature of the STAR detector, It has a roughly cylindrical geometry and is capped by two pole tips. The main coils lie outside of the TPC. The magnetic field has been mapped very precisely [69], allowing us to accurately measure and track the transverse momentum of the produced charged particles.

2.2.2 Time Projection Chamber

The TPC is the primary tracking detector of STAR, with a large geometric acceptance, and capable of handling the very high particle multiplicities produced in central heavy ion collisions. The TPC fills the central region of STAR, with a coverage of $-210\text{cm} < z < 210\text{cm}$, $50\text{cm} < r < 200\text{ cm}$, $0 < \varphi < 2\pi$ and $-1.8 < \eta < 1.8$. It is divided into two halves at $z=0$. As the world's largest TPC, it can be used to track particles, measure the particles' transverse momenta, and identify particles with $p_T > 100\text{ MeV}/c$ [70].

The TPC has several subsystems. To reconstruct the large number of particles produced in the central region with good momentum resolution and energy loss measurements, a very efficient TPC readout system [71] has been developed. To determine the spatial shift, and drift velocity, needed to calibrate the TPC during a run, a laser system [72] was developed, which uses ultraviolet (UV) laser beams to ionize the drift gas [73]. A sophisticated gas handling system is required for the TPC. The first of two mixtures of gases is used in the TPC during runs so far: P10 (90% Ar + 10% CH_4) or 50% He + 50% C_2H_6 , with great care required for the temperature and pressure monitoring [74].

2.2.3 Forward Time Projection Chambers

To cover the acceptance of particles produced at high pseudo-rapidity, two cylindrical FTPCs were installed at STAR, one on each side of the central collision region [75]. As we can see from Figure 2.2, the two FTPCs are located at $2.5 < |\eta| < 4.0$. The FTPCs are a good complement to the TPC, and make STAR better at tracking particles over a larger region of space.

2.2.4 Silicon Vertex Tracker

The SVT consists of three layers of silicon drift detectors surrounding and very close to the collision point at STAR [76]. The SVT provides two-dimensional hit position measurements and increases the tracking capabilities of the TPC by helping to identify the primary vertex of a collision. This increases the track momentum resolution and improves measuring the particle's energy loss. The SVT can also be used to help reconstruct short-lived particles.

2.2.5 Silicon Strip Detector

The SSD can be treated as a fourth tracking layer beyond the SVT. The SSD plus SVT provides a complete small tracking system inside the TPC [77]. This improves the tracking of produced charged particles, by recording more information on two-dimensional hit positions and energy loss measurements.

2.2.6 Photon Multiplicity Detector

The PMD is located behind the east FTPC, and covers the forward region at $z=550\text{cm}$ over the range $2.5 < \eta < 3.5$. It was installed to measure the spatial distribution of photons for further analysis [78]. The PMD is not used in our neutral pion analysis.

2.2.7 Forward Pion Detectors

The FPD's are located on both sides of STAR at $z=7$ m (west side) and $z=-8$ m (east side), and very close to the beam pipe. The FPDs were installed to measure and reconstruct neutral pions in the far forward rapidity region, with average pseudo-rapidities $\langle\eta\rangle=3.7$ (east side) and $\langle\eta\rangle=3.3$ (west side) [79].

2.2.8 Barrel Electro-Magnetic Calorimeter [52]

The BEMC is another important calorimeter besides the EEMC at STAR. The BEMC was constructed to trigger on and reconstruct high p_T processes such as jets, pions, eta mesons, and direct photons. The BEMC is an annulus located between the magnet coils and the TPC, with a large acceptance in pseudo-rapidity $|\eta|\leq 1$, and azimuthal angle $0\leq\phi<2\pi$. The inner surface is at $r=220$ cm, and the BEMC is about 6 m in length. This is a traditionally designed sampling calorimeter with a total of 4800 towers. No more details on the BEMC will be given here since it was not used in this analysis.

2.2.9 Beam Beam Counters

The BBC's, basically two arrays of hexagons and use same light readout techniques as EMC's scintillator counters, are installed on each side of the STAR detector to collect signals which are used to identify collisions [80]. The BBC's measure both the time of a collision, by the average flight time of produced particles, and the position of the interaction, by taking the difference of arrival times between the east BBC and the west BBC, and knowing the distance between the two BBCs. This fundamental trigger device provides a good constraint on the collision, and is also a primary means of determining the experimental luminosity.

2.3 The Endcap Electro-Magnetic Calorimeter

The Endcap Electro-Magnetic Calorimeter was designed and constructed at IUCF, and installed as one of the main calorimeters at STAR. This unique device was mounted on the west side of the STAR detector in order to increase the acceptance for particles such as direct photons, electrons, W^\pm , jets, and pion and eta mesons produced in the forward rapidity region from high energy collisions at STAR. The Endcap is crucial to the study of the gluon's contribution to the nucleon spin structure using the high energy polarized proton beam collisions, because of its large acceptance and triggering in jet production, gamma-jet coincidence measurement, and providing discrimination of γ/π^0 .

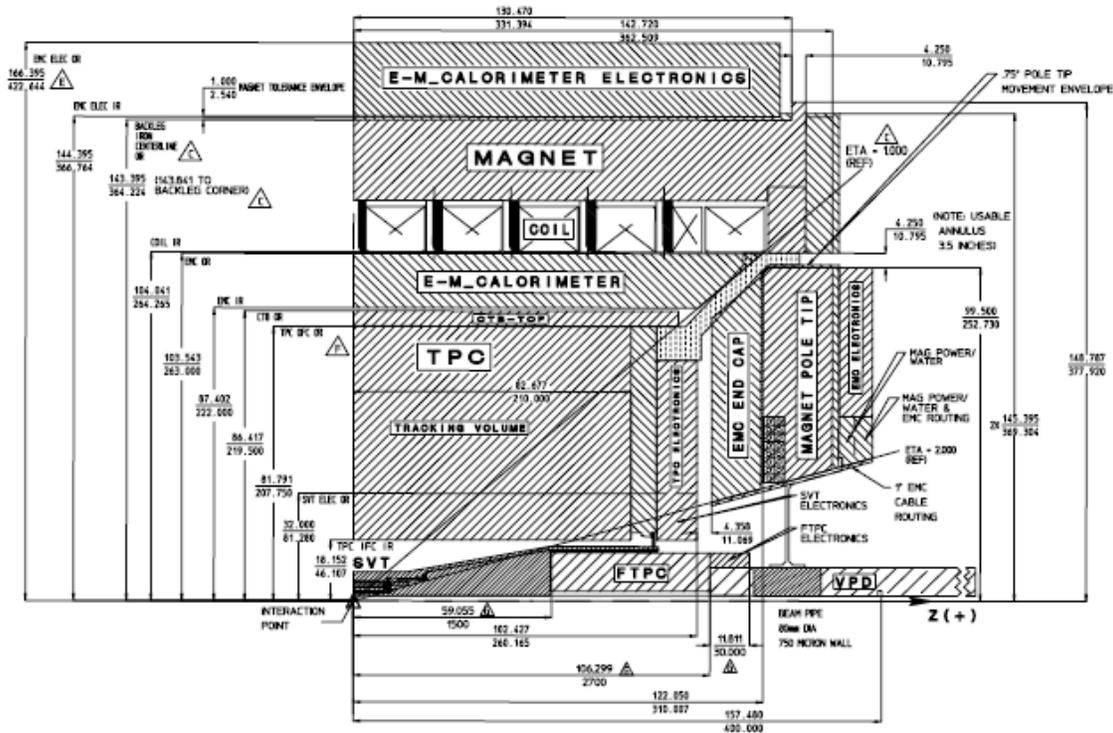


Figure 2.3: Schematic of half of the EEMC in a quarter of the STAR detector [81].

The EEMC is located on the inner face of the west pole-tip of STAR, as we show in figure 2.3. The Endcap is an annulus with projective geometry of an inner radius starting from 75 cm, an outer radius starting from 215 cm, and a longitudinal depth about 34 cm. The acceptance in pseudo-rapidity covers a range of $1.086 \leq \eta \leq 2.0$ and azimuthal angle $0 \leq \phi \leq 2\pi$. A small services gap is located between the Endcap and the Barrel.

2.3.1 EEMC physics performance requirements

We have discussed the core physics tasks previously. To achieve these goals, the EEMC has been designed very well in many respects. The Endcap is designed to cover most of the forward solid angle, to enhance the acceptance for particles produced at STAR. The detector hardware has a wide energy acceptance range from less than 1 GeV to over 100 GeV. The fine transverse segmentation of the calorimeter is designed to aid in identifying high energy photons and electrons, and distinguishing between photons and neutral mesons, and electrons vs hadrons. STAR is basically a slow detector, but the EEMC is designed to have a signal response fast enough to record and digitize signals for every bunch crossing (110 ns). The EEMC is also well equipped longitudinally; in addition to towers in different sectors, there are pre-shower, post-shower and Shower Maximum Detector (SMD) layers that provide important information about the particles that interact in the EEMC.

Energy measurements in the EEMC play a vital role in reconstructing incident particles during STAR runs, especially when RHIC is to run at 500 GeV center-of-mass energy for polarized proton beams. To meet the requirements for detecting W^\pm , the EEMC must be able to measure transverse momenta up to $p_T=40\text{GeV}/c$. Because of the

forward position of the Endcap at STAR, this corresponds to electron energies of up to 150 GeV. To contain the electron's energy requires that the depth of the EEMC should be at least 20 radiation lengths. The EEMC was built to have 21.8 radiation lengths at $\eta=2$, and 27.6 radiation lengths at $\eta=1$ from the collision point of view.

A detailed discussion of all the performance requirements of the EEMC can be found in reference [82]. A summary of this information is given below in table 2.2.

Table 2.2: This table summarizes the main physics performance requirements for the EEMC [82].

Feature	Requirement	Driving physics goals
Geom. acceptance	$1 \lesssim \eta \lesssim 2$; full ϕ	γ +jet sensitivity to $0.01 \lesssim x_g \lesssim 0.3$
E_{\min} in one tower	≈ 0.2 GeV	MIPs for calibration; γ 's from asymmetric π^0 decay; $\approx 2\%$ shower leakage from $p_T = 10$ GeV/c γ 's
E_{\max} in one tower	150 GeV	e^\pm from W^\pm decay at $\eta = 2$
Linearity	$< 10\%$ integral non-linearity, ~ 1 –150 GeV	Correct to give W^\pm daughter p_T to ± 1 GeV/c from lower- E calibrations
Depth	$\geq 20 X_0$, ≤ 1 hadron interaction length	$< 10\%$ shower leakage for 150 GeV e^\pm ; minimize hadron sensitivity to fit within existing space
Energy resolution	$(\frac{\sigma_E}{E}) \leq (\frac{16\%}{\sqrt{E}}) + (2\%)$	x_g uncertainty $\lesssim \pm 0.01$ for W^\pm reconstruction at $p_{T,e} \leq 30$ GeV/c
γ/π^0 discrimination	π^0/γ suppress factor > 3 for $p_T \approx 10$ –20 GeV/c \Rightarrow SMD	Keep background subtraction from enlarging $\Delta G(x)$ errors by more than a factor of 2
e^\pm/h^\pm discrimination	Suppress h^\pm/e^\pm by > 10 for $p_T \gtrsim 5$ GeV/c \Rightarrow pre/post-shower	Reach $> 3:1$ W signal/hadronic bkgd. ratio for $p_T > 20$ GeV/c; enhance Drell-Yan signal/bkgd.
Jet reconstruction	$\sigma \approx 0.1$ for η_{jet}, ϕ_{jet}	Reconstruct $x_{1,2}$ values for colliding partons in γ +jet coinc. to $\approx \pm 0.01$
Segmentation/rate capability	Tower occupancy $< 10\%$ @ $L_{pp} = 2 \times 10^{32}$; tower size $\gtrsim 2 \times$ shower diam.	Trigger on isolated γ or e^\pm vs. jet; Minimize tower hadron occupancy; obey WLS fiber min. bend radius
Tower calibration	Absolute E calibration to $\pm 2\%$ ($\pm 10\%$ online)	Minimize: systematic errors in x_g , $\int \Delta G(x) dx$ (p_T , hence x , threshold at acceptable trigger rates)
Coverage gaps	$< 2\%$ systematic shower E loss in cracks	Minimize systematic errors in extracted x_g , $\int \Delta G(x) dx$
SMD calibration	Relative gains of adjacent strips known to $\lesssim \pm 10\%$	Maintain sufficient γ vs. π^0 shower shape discrimination
Timing response	< 1 RHIC beam period (110 ns)	Aid TPC pileup reject; no occupancy from neighboring beam crossings

2.3.2 Tower mechanical structure

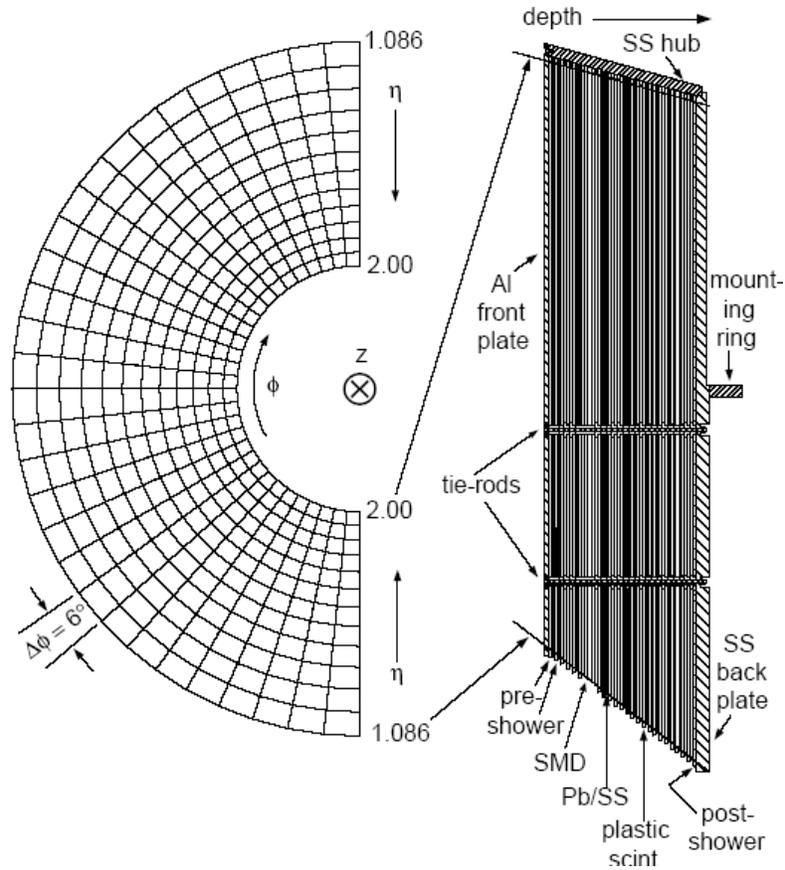


Figure 2.4: The left side shows half of the EEMC tower structure. The right side shows a profile cut from the lower region [82].

The EEMC is installed 2.7 m away from the center of STAR, measured to the front (east) face. The EEMC is divided into two halves, and one half is shown in figure 2.4. The segmentation of the calorimeter divides it into 720 towers, with each tower oriented projectively back to the nominal collision vertex. The span of each tower is 6° , which means $\Delta\phi=0.1$ in azimuthal angle. The towers cover increasing ranges of pseudo-rapidity, going from $\Delta\eta=0.057$ to $\Delta\eta=0.099$ as one moves from the outer radius near $\eta=1$ to the

inner radius at $\eta=2$. The calorimeter is a sampling detector, and is built based on the Pb (radiator) and plastic scintillator (layer) method. Every lead and stainless steel radiator is followed by a layer of 4 mm thick plastic scintillator. For each tower, there are 23 layers of Pb and stainless steel radiator, and 24 layers of plastic scintillator. The two halves of the EEMC are mounted on the pole-tip with the dividing line 15° away from the horizontal direction of STAR. The mass of the whole EEMC is about 25,000 kg. We can also see the pre-shower 1 and 2, SMD, and post-shower layers located at various depths in figure 2.4. These layers will be discussed later.

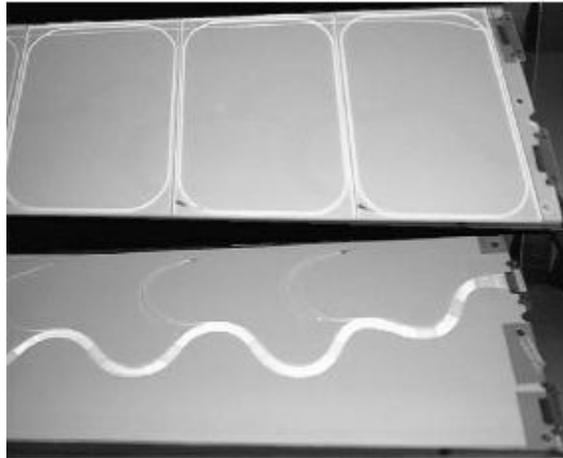


Figure 2.5: Megatile sample from a 6° subsector. The upper part shows the front face of the scintillator with σ -grooves; the lower part shows the back with wavelength-shifting fibers in place.

The optical system is organized according to the tower segmentation, and plays an essential role in energy collection for the EEMC. Based on the symmetric geometry of the EEMC, the calorimeter is divided into 12 sectors in ϕ , with 5 subsectors in each sector, and 12 towers in each subsector. The 12 towers in a sub-sector are counted along the pseudo-rapidity direction. The 5 subsectors cover a 30° region in ϕ and so the 12

sectors contain a total of 720 towers. Each layer of each sector includes two 12° megatiles and one 6° small megatiles as shown in figure 2.5. Each megatiles spans the full η range of the EEMC. To transfer the scintillation light from these megatiles, wavelength-shifting (WLS) optical fibers are inserted into the σ -grooves of the tiles, and optical connectors are used to couple the WLS fibers from the 12 tiles from each subsector megatiles. So for each subsector, there are a total of 24 layers of fiber bundles from the plastic scintillator. After these fiber bundles from each sub-sector are routed to a photo-multiplier tube (PMT) box, they are redistributed by tower. A tower grouping contains 24 fibers, each from the corresponding tile in the 24 layers, and each subsector has 12 towers. So there are 12 photo-multiplier tubes in each PMT box, one for each tower in the subsector.

2.3.3 Shower Maximum Detector

The Shower Maximum Detector (SMD) is an essential detector in the EEMC for this thesis work. It is important because it provides precise position information for particles, and can help discriminate between single photons vs photon pairs decayed from π^0 or η^0 mesons by measuring angle between 2γ 's and relative energy, based on the transverse shower profile characteristics.

The SMD is located after the fifth radiator, which is about five radiation lengths deep in the EEMC. The fundamental application of the SMD is to provide a fine measurement of the electro-magnetic transverse shower shape produced by particles. For example, in our π^0 analysis, it is absolutely critical in achieving high reconstruction efficiencies, especially at high pT. The location of the SMD layer is well chosen to maximize its discrimination power and to optimize the transverse energy profile in the

SMD planes.

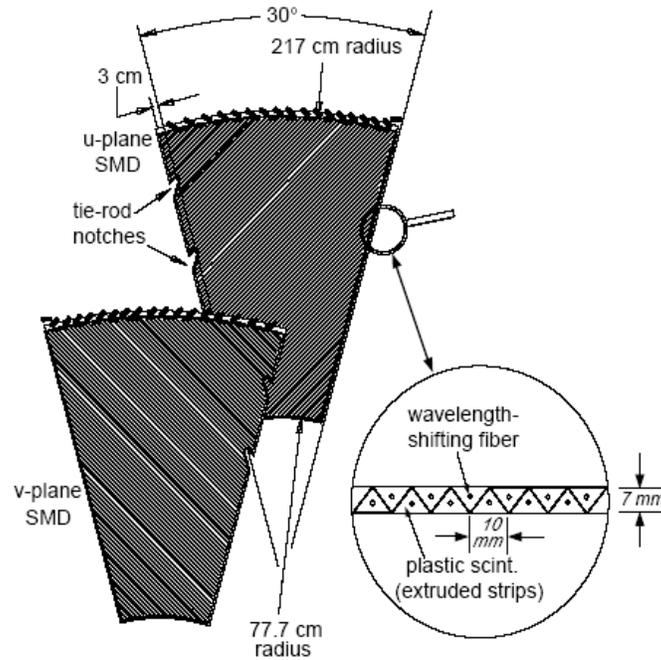


Figure 2.6: Schematic of the SMD from a 30° sector. A layer of SMD includes two planes, U and V, which are oriented orthogonally to each other [82].

The SMD is an orthogonal system in design with two planes, a U plane and a V plane, which are overlapped at the sector edges. Physically the SMD layer is divided by 30 degree sections. So there are 12 sectors of SMD covering the whole EEMC azimuthally with a U and a V plane in each sector. Figure 2.6 shows the layout of the SMD planes from a sector. We can see that it covers all 60 towers in each sector of the EEMC. Within each SMD plane, there are 288 scintillator strips that vary in length from several centimeters to a maximum of 126 cm, as we can see from figure 2.6. These strips are triangular in cross section and adjacent to each other. This overlap between strips makes each SMD plane gapless, which means an electromagnetic shower will share its

energy among adjacent strips. This construction improves the position and shape resolution of an electromagnetic shower.

The optical system of the SMD is an independent construction from the EEMC. WLS fibers are embedded along each strip in each plane to transfer the scintillation light from the SMD to the outer optical connectors, where the light is transferred to 16-anode photo-multiplier tubes. Three boxes, each equipped with twelve such multi-anode PMTs (MAPMT), collect the energy from a total $12 \times 16 \times 3 = 576$ strips from both SMD planes ($288 \times 2 = 576$). From these MAPMT signals, we derive the energy deposition information from each strip, which is a critical part of our analysis.

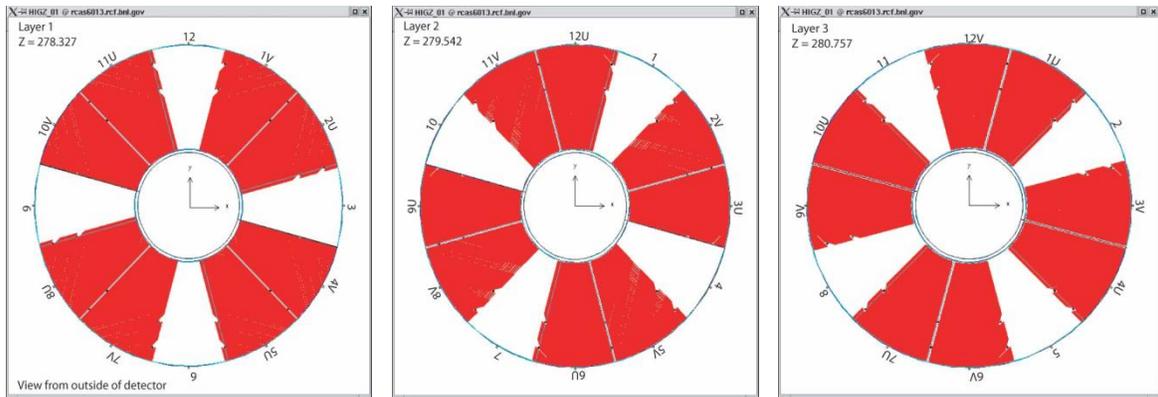


Figure 2.7: Asymmetric layout of the SMD sublayers in the EEMC [83].

During its construction, the SMD layer was actually designed to contain three sub-layers in depth. These sublayers are sector dependent, as we can see from figure 2.7. In each sector, the U plane and V plane occupy two of the sub-layers. The third sub-layer is filled with passive plastic. The order of the three sub-layers depends on the sector, as shown in figure 2.7. In this analysis, this sector-dependent material sub-layer will be marked as ‘space’. The SMD planes are colored as red, and space is colored as white in

figure 2.7. This structure produces noticeable effects, as described in the later section on data analysis. The sector dependent structure of the SMD is summarized in the following table.

Table 2.3: Sector-dependent structure of SMD in the EEMC.

Sector	1	2	3	4	5	6	7	8	9	10	11	12
Layer1	V	U	Space									
Layer2	Space	V	U									
Layer3	U	Space	V									

Layer1 is located at $z=278.327$ cm, layer2 is located at $z=279.542$, and layer3 is located at $z=280.757$, where z is the longitudinal distance measured from the center of STAR. Table 2.3 shows details of the structure distribution. We can see this asymmetric layout has a special property: sectors 1, 4, 7 and 10 have the same structure of V, Space and U; sectors 2, 5, 8, and 11 have the same structure of U, V and Space; and sectors 3, 6, 9 and 12 have their same structure of Space, U, V. U planes are in front of V planes for 2/3 of the whole EEMC. This tiny spatial difference will make our analysis a little complicated when examining the two SMD planes because of our great sensitivity to electromagnetic shower shape.

2.3.4 Pre-shower and post-shower detectors

On the front side of the EEMC, the first two layers of scintillators are pre-shower1 and pre-shower2, respectively. The Post-shower is the last layer (layer 24) at the far side of the EEMC. The segmentation of the pre-shower and post shower detectors is identical to that of the towers. These three special layers of the EEMC subsystem are mainly used

for e^\pm /hadron discrimination because of the energy deposition differences in these special areas. The pre-showers also provide help on γ/π^0 discrimination, since photons are less likely to deposit energy in pre-showers than π^0 s (di-photons) do. But applying any cut on the pre-showers would decrease the statistics of our π^0 yield, therefore they were not used in this work.

The optical system for the pre-showers and post-shower is also based on the WLS fibers, like the towers and SMD's. Each layer of the pre-/post-showers has two independent WLS fibers inserted to collect the light from energy deposition. For each layer, one of the two fibers is used to transfer the light from each tile in each sub-sector, and the energy is added to those from the towers beneath the same tile in geometry. The second fiber is used to transfer the light from each tile independently to a channel of an MAPMT. By this design, the energy is recorded twice in different fibers independently. The purpose of doing this is to improve the capability to distinguish e^\pm /hadrons by comparing the energy deposition from these special layers to that from towers, to differentiate based on the longitudinal shower development. To improve the quality of these signals, the scintillators in the pre-shower and post-shower layers are made a little thicker and brighter than the normal tower megatiles, so that we can achieve more photostatistics here.

2.3.5 EEMC readout and trigger system

The EEMC readout system contains two different subsystems one for the calorimeter towers, and one for the SMD and three layers of pre-shower1, pre-shower2 and post-shower detectors. All these detectors are based on the same technology of

plastic scintillator coupled to optical fibers, but there are big differences in the front-end electronics (FEE) signal analysis and generation of triggers because of the different requirements on detector response and signal processing time.

The EEMC tower readout system uses the technology similar to that used in the Barrel EMC [84], with small modifications based on the differences in geometries and material choices. As discussed above, the light collected for each tower is combined from 24 layers of scintillating tiles in the EEMC in a single anode photo multiplier tube. The collected energy information from each tower is then sent to a tower digitizer card and is digitized in a 12-bit linear flash ADC for each input channel for every RHIC beam crossing, which occurs about every 110 ns. The ADC outputs are stored in a digital pipeline for readout on a valid trigger, and sent to digital data storage and manipulation (DSM) boards for use in the trigger specifically the ADC from the highest tower in a patch of 8, 10 or 12 towers as well as the sum is transferred to the DSMs for use in triggering. For example, upon the request of a valid trigger, the ADC output is read out by a crate controller and passed over optical fibers, then to DAQ or to the L2 trigger.

As a trigger detector, it is important for the EEMC tower data to be integrated into the STAR Level-0 (L0) trigger. These data, collected as discussed above briefly, are propagated through 3 layers of DSM boards to form the level 0 trigger. The purpose of L0 triggering is to quickly identify events in which individual towers (high towers) and adjacent clusters of towers (trigger-patch) contain high energy deposition as determined by passing various preset thresholds. For the EEMC, if large transverse energy (E_T) above the trigger threshold is recorded from a single tower, this is identified as a high tower (HT) trigger event. These triggers are most efficient for detecting direct photons

and high $p_T \pi^0$ s in the EEMC. Once an event is identified, all of the ADC information from individual towers of this event will be sent to DAQ (see below). It is also possible to transfer this information to the next stage for further processing in Level 2 (L2) or Level 3 (L3) trigger schemes.

The readout systems for the SMD, two layers of pre-shower detectors and one layer of post-shower detector use the same technology. The SMD strips, pre-shower and post-shower layers use plastic scintillators coupled by optical fibers to transfer energy information, although shapes and sizes are different between the SMD and pre-/post-shower detectors. A front-end electronics (FEE) system was constructed at IUCF for readout of the SMD, pre-shower and post-shower detectors, independent from the towers readout system.

The SMD and pre-/post-shower information is not used in the generation of L0 triggering. The energy information is sent to multi-anode photon-multiplier tubes (MAPMT) in boxes, and each MAPMT anode has an independent preamplifier, gated integrator, and linear 12-bit ADC digitizer card to digitize the energy signal. Twelve of the 16-anode PMT's are located in the same box, which gives a total of 192 channels per box. As with the tower signals the ADC convert on every bunch crossing (~ 110 ns) and are stored in a digital pipeline. On a valid trigger the data is transmitted via optical fiber to a data collector and from there to DAQ.

The philosophies for the SMD and pre/post-showers are similar, from scintillator to fiber to PMT, and MAPMT due to large number of signals. Although the SMD, pre-shower and post-shower information is not used in the L0 trigger system, in principle, it can be used in L2 or higher triggering decisions to provide hadron or jet suppression

during the pp collisions. This capability is not currently being used at STAR.

2.4 STAR Data Acquisition (DAQ) System

STAR has been constructed to collect large amounts of information from collisions that occur frequently in time. It is therefore very important to have an efficient data acquisition (DAQ) system [85, 86] for STAR. The fundamental function of the DAQ system is to read the data collected from all the detectors, which produces an event size of 80 MB with an input rate of up to 100 Hz, then to reduce the data rate to 30 MB/s and to store the data into the HPSS facility [87].

To understand the process of data acquisition at STAR, it is important to examine the trigger system more closely. The STAR trigger designed a four-level system with Level 0 (L0), Level 1 (L1), Level 2 (L2) and Level 3 (L3) capabilities [88]. L0 and L1 are hardware triggers based on the fast information from pipelines, and are used to make quick decisions during each beam crossing. L1 is there but passes all. L2 is software used in the experiment. L3 is a software trigger based on the tracking information from slow detectors such as the TPC and SVT. L3 has mainly been used in the online reconstruction of TPC tracks at STAR and not for triggering. The different trigger levels have different processing times. For L0, the processing time is 1.5 μ s; for L1, it is 100 μ s; for L2, it is 5 ms; and for L3, it is about 200 ms.

The first two levels of triggering are fully pipelined, with fast access to the STAR raw data. We discussed the L0 trigger for the EEMC in the previous section. Signals from a detector are shaped, digitized, and stored in the DSM boards in the L0 trigger, which is also called the raw trigger data. In the DSM boards, signals are combined and analyzed

with other signals in a multi-layer pipeline to form a DSM tree. The output from the DSM tree is transferred to the Trigger Control Unit (TCU), and a decision about this kind of event from each STAR detector will be identified with a 12-bit token [89]. Meanwhile, all of the trigger and fast detector data are sent to the L2 trigger processor, a linux box running C++ code. In the L2 processor, the dataset will be analyzed by criteria subject to different detectors or detector subsystems, and selections will be made for different trigger types of data. If the dataset pass the criteria, it will be transported to the STAR DAQ system. The DAQ system receives data from the traditional VME-bus boxes with a receiver board in each of them. These parallel structures are connected by a Myrinet [90, 91] network.

Datasets are stored in the HPSS system for future analyses. Researchers can retrieve data from HPSS selectively and analyze the data then.

Chapter 3

Software Development

3.1 π^0 decay in the endcap

The purpose of developing this software is to reconstruct the π^0 mesons during p+p collisions in the STAR experiment. A neutral pion meson will generally decay into two photons with a branching ratio of 98.8% [92].

$$\pi^0 \rightarrow \gamma\gamma$$

The lifetime of the π^0 is $\tau=8.4\times 10^{-17}$ s corresponding to a decay length $0.025\mu\text{m}$. Since the decay is not from a long life procedure, we assume the decay photon pair originates from the primary vertex. For each π^0 the invariant mass is calculated then:

$$mass = E \times \sqrt{1 - z_{\gamma\gamma}^2} \sin \frac{\varphi_{\gamma\gamma}}{2} \quad (3.1)$$

We denote E_1 and E_2 as the energies of the two decayed photons. $E = E_1 + E_2$ is the total energy of the photon pair and $\varphi_{\gamma\gamma}$ is the opening angle between them. $Z_{\gamma\gamma}$ is the energy sharing defined as:

$$z_{\gamma\gamma} = \left| \frac{E_1 - E_2}{E_1 + E_2} \right| \quad (3.2)$$

The two photons head into the endcap and deposit energies, which provides us shower profiles in the endcap. Different neutral pions with different energies and transverse momentum will result in different shower profiles. The energy of the photon pair will be mainly deposited in the EEMC towers. The opening angle $\varphi_{\gamma\gamma}$ between the photon pair is determined from collected SMD information. Most decays from neutral

pions are not symmetric. It is also important to know the energy sharing $Z_{\gamma\gamma}$ between the photon pair. The SMD provides information for not only the opening angle, but also the relative energy of two showers. So the idea is to collect information from SMD first, then reconstruct points and pion candidates. Once we get the energies of the two SMD showers and tower showers, we calculate each point candidate's energy based on tower energies after sharing with the SMD showers in a 3×3 tower patch. Then we decide the energy sharing $Z_{\gamma\gamma}$ by equation (3.2) and invariant mass of the neutral pion by equation (3.1).

Our software has been developed accordingly upon the above kinematics, please refer to the following sections for details.

3.2 π^0 finder software introduction

We have made much effort in developing the analysis software required to do the inclusive π^0 measurement in the EEMC since 2005. Our software package was developed upon the framework of Jason Webb, whose codes were stored into STAR CVS [95]. The importance of measuring the double spin asymmetry from high- p_T inclusive π^0 production in longitudinal pp collisions has been explained in chapter 1. To achieve that goal, the reconstruction, efficient detection, and analysis of the high transverse momentum neutral pions in the EEMC is critical. This software includes four generic parts: the A2E-Maker, the Cluster-Maker, the Point-Maker and the Pion-Maker (Pi0-Mixer). In this chapter, we will discuss important routines of our newly developed π^0 finder software. You can also read appendix C for details of our code.

The A2E-Maker is basically a connector between the digitized data stored in the RCF system and the last three makers. This maker translates the ADC values from the MuDst data files into energies with the units of GeV. People can also rescale the energy into MeV if necessary. During this translation, pedestal subtraction and gain corrections for each channel are applied to make the data more useful. In this maker, most of the fundamental energy information in the EEMC are defined and saved in different types of classes and vectors to be utilized in the later analysis.

Based on the specific EEMC structure of towers, pre-shower, post-shower and SMD detectors, a special Cluster-Maker has been developed. In the reconstruction of neutral pions in the EEMC, the nature of the EM shower of the EEMC dictates that we can obtain transverse shower information in almost all layers of the EEMC. It is not necessary to use all the deposition information all the time for the analysis. Most of the showers of interest are distributed in small regions within adjacent towers, pre-shower layers, post-shower layers and SMD strips. So the Cluster-Maker has been developed to produce clusters of EEMC towers, pre-shower and post-shower layers, and SMD strips. This is especially important for the SMD clusters, because we use the SMD position information to locate photon pairs provided in neutral pion decay.

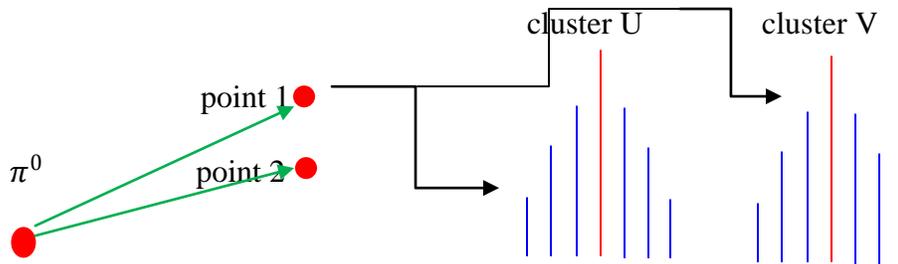


Figure 3.1: A flow chart from π^0 to SMD clusters. A π^0 particle generally decays into two photon points, and each point deposits a pair of SMD clusters in the EEMC. In the above chart, we only show one pair of SMD clusters from point 1.

We reconstruct photon point candidates from the SMD clusters found in the preceding. 98.8% of the neutral pions will decay into two photons [92]. To reconstruct a π^0 , we have to find all possible photon points from the EEMC first. A point candidate is defined as the overlap of two SMD clusters, one from the U plane and another from the V plane, with the intersection occurring underneath an “active” tower, which is defined as a tower with positive energy deposition. We use the SMD clusters to achieve position information for the photons. The energy of the photon is obtained by summing over a 3×3 tower patch centered on the active tower, as long as the active tower is not at the edge of the EEMC. A set of criteria is used to qualify point candidates, and we will discuss details later.

All qualified points are used in the Pion-Maker to reconstruct π^0 candidates. To reconstruct a π^0 , we need to know the total energy, the energy sharing ratio of the two photons, and the opening angle between the two photons. In this class, some basic information about the reconstructed π^0 such as pT, energy, invariant mass, spin dependent yields, and so on, are saved into histograms and trees for further analyses, such as for efficiency studies and double spin asymmetry calculations.

3.3 The Cluster-Maker

If a π^0 meson produced in a pp collision heads toward the EEMC, it will decay into two photons about 98.8% of the time. Ideally, the two photons will produce two showers in the EEMC, and this means we can measure two tower clusters for energy collection, and typically four SMD clusters (two from each plane) to find the positions of the photon pair.

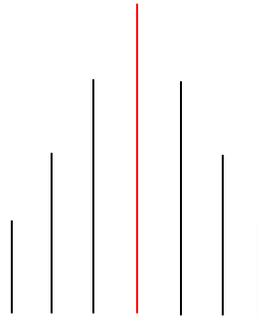


Figure 3.2: A typical SMD cluster scheme. The red line represents the seed strip. Three strips on both sides are added to form a SMD cluster.

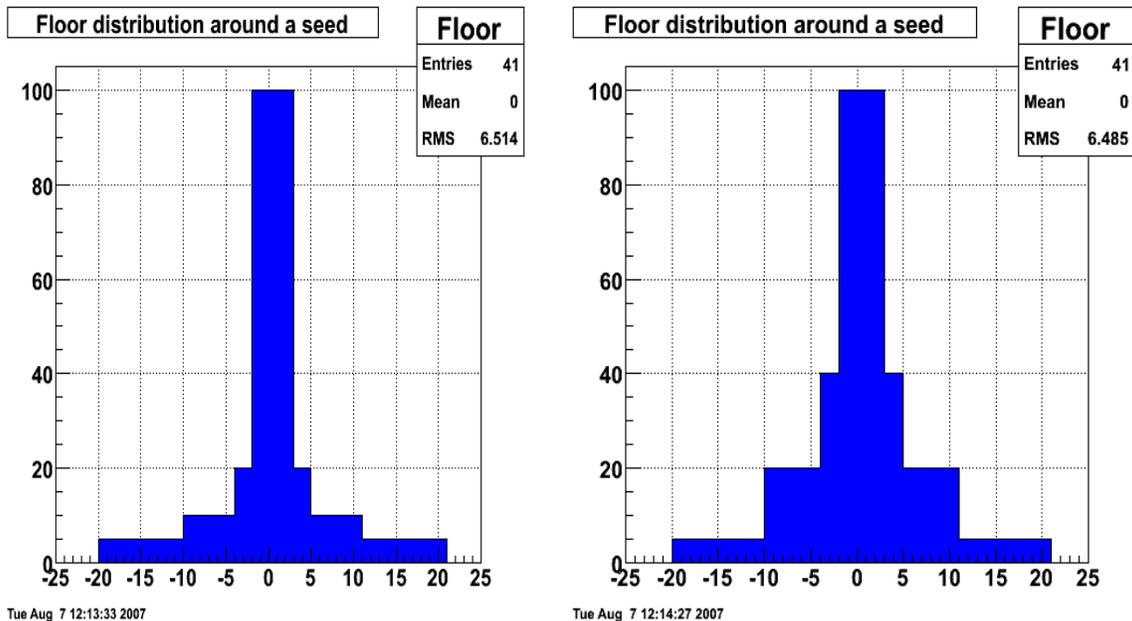


Figure 3.3: The floor settings of SMD to find seed strips. The left plot is designed for the front SMD plane and the right plot is designed for the back SMD plane from the incident point of view for real data analyses to suppress fluctuations.

We reconstruct SMD clusters sector by sector, plane by plane. To qualify as a seed, a SMD strip has to pass the following requirements: First, the strip index must be within the range [3,283]. We do not allow a strip at the very edge of the SMD plane to be a seed. Second, this strip has to be a good strip. Dead strips, as identified by status bits, are disqualified. Third, the energy of the strip should be larger than the threshold value of 1.5 MeV. This threshold cut is at about the MIP value (1.4 MeV, according to simulation

study). But this value can be changed to 2 or 3 MeV easily if desired. Fourth, the energy of the strip from every seed, except the first one, is also required to pass a special “floor” setting designed from the SMD structure. The floor setting is set to 0 for the first seed. The special floor setting is shown below in figure 3.3.

The floor setting basically suppresses the appearance of a second seed around an existing seed. For simulation studies, we used the distribution of energies shown in the left plot on figure 3.3 as the general floor shape for all sectors of the EEMC. For analysis of real data, we used the left plot for the first (front) SMD plane and the right plot for the second (behind) SMD plane in all sectors. The reason we chose different settings in the real data analysis will be explained in more detail in Chapter 5. For now, we will describe the general “floor” algorithm. In figure 3.3, channel 0 means that we have found a seed there; and the floor value within two strips from this seed is set to the energy of the seed. For strips that are three or four strips away from this seed, the floor value is set to 20% or 40% (simulation of data) of the seed energy. Strips that are 5-10 strips away from this seed have a floor value set to 10% or 20% (simulation of data) of the seed energy. And for strips 11-20 strips away from this seed, the floor value is set to 5% or 10% (simulation of data) of the seed energy. Summarizing the third and fourth requirements, we can get the energy requirement for a strip to qualify as a seed:

$$\text{Energy of seed} < (\text{mSeedFloor} * \text{floor}[\text{index}] + \text{mSeedEnergy}[\text{plane}]) \quad (3.3)$$

where mSeedFloor is a parameter defaulted at 1, and floor[index] is the floor value, initially set at 0, but set up according to figure 3.3 after finding the first seed. mSeedEnergy is the 1.5 MeV threshold cut above. After the strip passes all the above four requirements, it will be pushed into the SMD seed pool.

Now we iterate all seeds from the pool, and start from the most energetic one to reconstruct our SMD clusters. Starting from a seed, this strip will be added into a SMD cluster at first. Then the adjacent three strips, if not dead, from both sides from the same plane in the same sector will be also added to the same cluster. All these seven strips are then marked, and if other seeds are found among these marked strips, they cannot be used as seeds to reconstruct SMD clusters anymore. But they can still be used as part of a cluster for another nearby seed. This creates a problem of overcounting strip energies during the reconstruction of SMD clusters. We will fix this problem later. If strips from the adjacent six are dead or failed, their energies will be set to be zero. A minimum requirement of three active (with positive energy deposition) strips is required to save the SMD cluster. Before we fix the above overcounting-strip problem (see below), these SMD clusters are stored in a temporary SMD cluster type of vector.

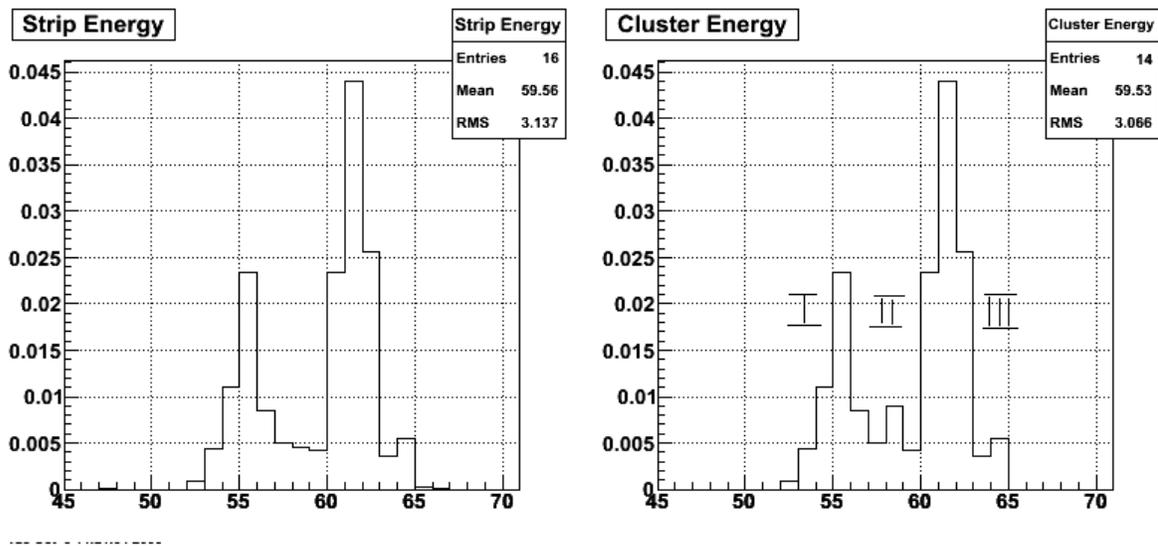


Figure 3.4: Double-counted strip example from simulation sample with flat energy [5,60] GeV and flat EEMC distribution.

Figure 3.4 shows an example of the double-counted strip problem. The left plot in figure 3.4 shows the raw SMD strip energy deposition information from an event from a

simulation sample with a flat energy of [5,60] GeV and flat EEMC distribution. The x-axis denotes the strip index, and the y-axis denotes the energy deposition in GeV. Two SMD clusters are found based on the left plot. We then reconstruct the energy deposition information from these two SMD clusters, as shown in the right plot on figure 3.4. As we can see, strip channel 58 is counted twice, so that it produces a bump when we consider the two clusters. To fix this problem, we divide the SMD in the right plot on figure 3.4 into three regions: I, II and III, based on the two seed strips. We sum over strips in regions I, II, III and get the total energy E_I , E_{II} and E_{III} in each region. We re-assign energy values to those overlapped strips in region II according to these formulae:

$$E1 = E_{II} \times \frac{E_I}{E_I + E_{III}} \quad (3.4)$$

$$E2 = E_{II} \times \frac{E_{III}}{E_I + E_{III}} \quad (3.5)$$

Note that E_{II} is flexible here; it means the energy of the double counted strips in region II.

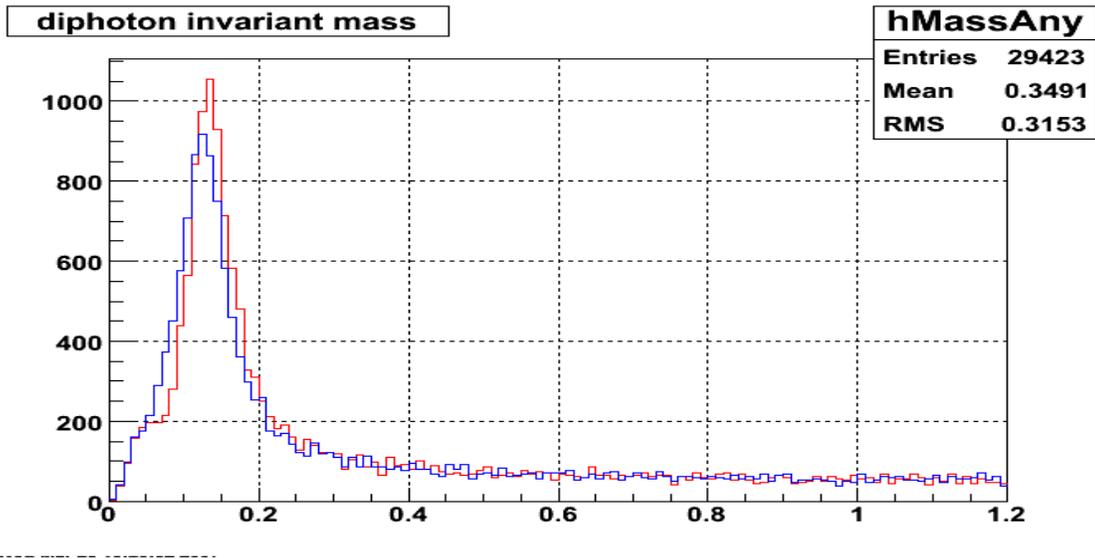


Figure 3.5: the invariant mass spectrum comparison (1:1 legend) when fixing the double-counted strip problem from a MC sample with flat energy [5,60] Gev and flat EEMC distribution.

The result of this improvement in cluster definition is shown in figure 3.5. Based on the Monte-Carlo sample with flat energy and flat EEMC distributions mentioned previously, we show the invariant mass spectrum comparison in figure 3.5. The blue curve shows the invariant mass without fixing the over-counting problem, while the red curve represents the spectrum after fixing the problem. We can see from figure 3.5 that the peak width is sharper after this improvement, changing from $\sigma=0.035$ to 0.031 . The peak position is closer to the theoretical value of 0.135 GeV, although we use a sampling correction factor of 1.3 (more details later) here. This specific sampling correction factor is applied to tune the energy values when we convert ADC values to energies. We point out these two benefits here, because we have also achieved smaller peak widths and more stable peak positions as a function of transverse momentum in the real data analysis.

After the overlapped strips are divided into two new strip energies, each new strip will be added into its original cluster. We now assign a unique key to the SMD cluster, and go to the final step to store them in the storage SMD cluster vector for later maker usage.

3.4 The Point-Maker

The EEMC Point-Maker produces photon point candidates from SMD clusters and related tower clusters. SMD clusters are used to determine the position and energy sharing of points, and related tower clusters are used to determine the absolute energy of points. The SMD clusters are inherited from the EEMC Cluster-Maker described previously. Because the SMD U plane and V plane are orthogonally oriented, an incident

photon is typically going to give an intersection between the two planes. In the Point-Maker, a point is defined as the coincidence of two SMD clusters from U and V planes individually, underneath an active tower.

We reconstruct points sector by sector. Starting from the first sector, we loop over all SMD cluster for the two planes, from inner to outer radius, to build all U-V SMD cluster combinations associated with an active tower. If an isolated U-V pair with an active tower is found, this pair will be identified as a point. An isolated U-V pair means both clusters have only one intersection with the other plane. We sort all combination pairs by the relative energy ratio between the U and V planes. If we find that two or more clusters from one plane are sharing a single cluster from the other plane, we use a splitting algorithm to split the single cluster energy, and do the energy matching with the two or more clusters. In this case we will eventually choose the points with the best energy matching to the point pool. In a third case, if no isolated pair or splitting pair are found, we form multiple combination pairs with the same clusters. We then sort them by the relative energy ratio, and choose the points with the best energy matching between the U and V clusters. No matter which case is considered, after we push an identified point into our point pool, we remove the SMD clusters related to that point from the cluster pool, and loop over all remaining SMD clusters again to find the next point. After all points in the sector are identified, we go to next sector, and continue, until we find all points in the EEMC.

Figure 3.6 illustrates most of the energy deposition situations that arise from the decay photon pairs in the SMD. Each line in both the vertical and horizontal directions represents a SMD U or V cluster separately. The red point on the left represents an

isolated point; the central two points represent the unique points we described above; and the right four points correspond to the two decayed photons in SMD, which is the normal case. The isolated point will be pushed into the point pool. The central part of two unique points will be treated by the splitting algorithm. As for the normal case indicated on the right, we will discuss this later.

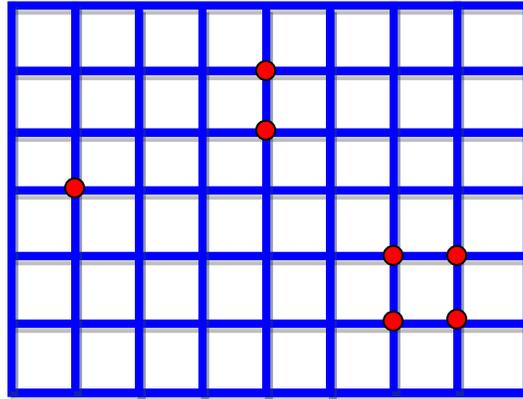


Figure 3.6: SMD-only points distributed in the SMD U and V cluster network above.

The splitting algorithm was developed to treat unique points stored in the temporary vector. The basic feature for these points is that they share the same U or V cluster with each other. But when we save a point into the pool, we need to save the two sets of cluster information for this point. In this case, it would be incorrect to save the common U or V cluster to two unique points. The splitting algorithm is designed to split the common cluster into two new clusters, with new cluster Id keys and energy information. So in practice, if the size of the temporary storage vector is not less than two, we start our splitting procedure; otherwise, the element in the storage vector is an isolated point, and it will eventually be pushed into the final point pool. Before we enter into the splitting, we sort these unique points by relative energy sharing from their SMD clusters. We then loop over all unique points to the mean value of strips for two clusters in both U and V

planes. If two points are found to be sharing the same U or V cluster, the energies between the common shared cluster and other two clusters from the two points will be tested to see if they qualify to be split, using the following equation:

$$zratio = \left| \frac{E_{share} - E_{c1} - E_{c2}}{E_{share} + E_{c1} + E_{c2}} \right| \quad (3.6)$$

where E_{share} is the energy of the shared cluster, and E_{c1} and E_{c2} are the cluster energies from the other plane of the two points. If the zratio is not larger than 0.2, we split the shared cluster to create two new clusters; otherwise, we do not consider the two points sharing a cluster as likely to be two separate showers. The zratio cut of 0.2 was chosen based on the simulation study shown in figure 3.7.

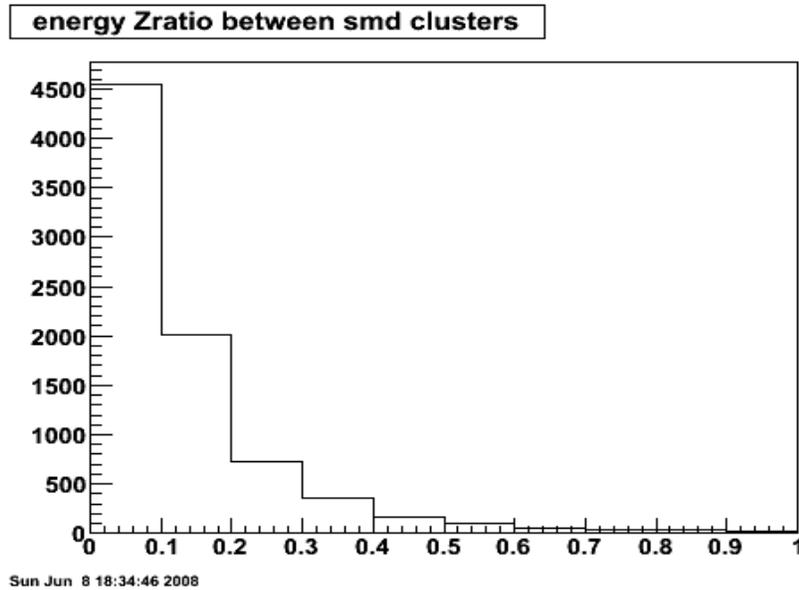


Figure 3.7: Zratio distribution between the shared cluster and two clusters from the other plane with the flat energy [5,60] GeV MC sample.

By applying the 0.2 zratio cut, roughly 80% of the unique points qualify for the splitting algorithm. The key part of the splitting code is to split a single cluster in one SMD plane shared by two clusters in the other plane into two new clusters.

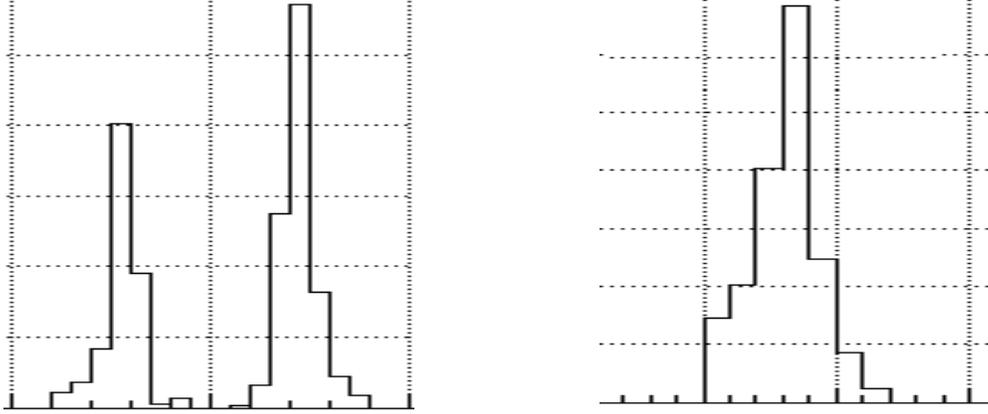


Figure 3.8: The left plot shows two U clusters, which share the same V cluster shown in the right plot. V1' and V2' will be the two generated new clusters.

The two new generated clusters will have the same centroid and strip index numbers as the old one. But they will have new cluster Id keys and energies. The energies will be calculated by the following formulae:

$$E'_1 = E_{share} \times \frac{E_{c1}}{E_{c1}+E_{c2}} \quad (3.7)$$

$$E'_2 = E_{share} \times \frac{E_{c2}}{E_{c1}+E_{c2}} \quad (3.8)$$

where E'_1 and E'_2 are the energies of the two new clusters; E_{share} is the energy of the shared cluster; and E_{c1} and E_{c2} are the cluster energies from the other plane for the two points. After the splitting, a new point 1 with energy E'_1+E_{c1} and a second new point 2 with energy E'_2+E_{c2} will be pushed into the final point pool. Meanwhile, the corresponding clusters will also be saved for related points with new key number.

After analyzing these isolated and unique points, we treat the “normal case” in figure 3.6. The routine is that we loop over the clusters remaining from the previous stages, and see if we can find clusters matching two or more points. These points will be temporarily saved in another vector. We loop over these points to find the point with the

best energy matching between its U and V cluster. We push this point into the final pool and remove the two clusters forming the point from the cluster pool, then search for more points recursively.

We loop over all points from the pool, and apply weights to tower 0 (the tower right above the SMD point) and the neighboring towers of each point. We use the SMD energy of each point as the weight. By doing this, if a tower contributes to only one point's energy, it will be only weighted once; but if a tower contributes to multiple points' energies, it will be weighted multiple times. We then go to the second step, to decide the energy of every point. We loop over all points again, and in the chain of each point, we loop over the neighboring towers around tower 0. The energy of this point will be decided by:

$$E_p = \sum_{i=1}^n E_{Ti} \times \frac{E_{SMD}}{W_i} \quad (3.9)$$

where E_p is the final energy of the point; i sums over tower 0 and all neighbors; E_{Ti} represents the i th tower energy; E_{SMD} represents the point SMD energy; and W_i is the total weight for the i th tower. After the calculation, we replace the energy of each point with the new E_p and save it in the same point pool to reconstruct π^0 . Energies in the pre-shower and post-shower layers are also decided by this method, though they are not used in this analysis.

3.5 The Pion-Maker

The Pion-Maker is a complicated object containing several classes with different functionalities. The Pion-Maker has been developed with three stages: reconstruct all

possible π^0 candidates; filter these candidates by specific cuts and settings to store them into π^0 trees; and build a framework so that we can read the trees for further analysis.

We use the points made in the Point-Maker as the input to reconstruct our π^0 candidates. The points are stored in the final point pool vector defined in the Point class. We call these points, and form “gamma-gamma” pairs on a sector by sector basis for each event. For real data, we use the primary vertex position as the vertex information for both points from the same event; for Monte-Carlo studies, the vertex is set to zero for all events by default, but users can change these settings. For each point, we use the energy and position information to calculate the 4-momentum of each point. When we consider a pair of points, we can then calculate the invariant mass, momentum, and other information for the π^0 candidate based on each point’s basic information and the primary vertex. At this stage, all of these matched gamma-gamma pairs from the same sector provide us with the raw π^0 candidates by a mix-maker class.

Then these π^0 candidates go into the analysis class at the second stage. We filter these candidates by various cuts and settings, sort the survivors by different spin states, and store them in the π^0 tree for further analyses. To save these trees, another class named Mix-Event is used in this class. The Mix-Event class basically sets up all necessary information for the π^0 s before they can be stored to trees and chained, so this Mix-Event class is also used in the third stage.

In our double-spin asymmetry analysis, the π^0 finder program does not do everything for us, but it is very general and prepares the results for future analyses. A Pi0-Reader class was developed to chain over the π^0 tree files, to be very flexible, and accommodate a variety of users. The Pi0-Reader class reads reconstructed pions and

returns events in the type of the Mix-Event class, where we save the information such as invariant mass, pT, eta, phi, $Z_{\gamma\gamma}$, opening angle $\phi_{\gamma\gamma}$, energy, spin-four states, and others. Based on this framework, we can normalize the spin-dependent π^0 yields according to relative luminosities in the asymmetry study, for example.

In the kinematics, the energy E of a π^0 is the summed energy $E_1 + E_2$ of the two points. $z_{\gamma\gamma}$ denotes the energy sharing of the two gamma points in a π^0 as shown in equation (3.2). And, the invariant mass of neutral pion is calculated by equation (3.1). The transverse and longitudinal momenta, pT and pz, are derived from the total momentum, and we can directly call them for further analyses.

At the second stage of the Pion-Maker, we filter the raw pion candidates and store them into π^0 trees by the Pi0-Analysis class. A number of histograms are also stored in root files instead of trees. The basic classes Spin-Cuts, Spin-Histos, and Mix-Event are designed to help the analysis. The Spin-Cuts class provides some fundamental thresholds for the π^0 analysis. The z-component of the vertex for reconstructed π^0 is constrained from -150 cm to 150 cm. The default cut for the energy sharing variable $z_{\gamma\gamma}$ is from 0 to 1, which includes all possibilities. The pseudo-rapidity detector η cut is set based on the EEMC geometry from 1.086 to 2.0. And to satisfy the filter, the tower 0 transverse energy of at least one of the two points which are used to reconstruct a raw neutral pion has to exceed a cut at peaked at 3.0 GeV with a bivariate Gaussian distribution along the pseudo-rapidity and azimuthal directions, chosen to be less sensitive to tower boundary effects found later in real data analysis. The bivariate Gaussian distribution is set by the following formula:

$$Cut = 3.0 \times e^{-0.5 \times \left(\left(\frac{peta - meta}{0.035} \right)^2 + \left(\frac{\phi'}{2.3} \right)^2 \right)} \quad (3.10)$$

where $peta$ is the detector eta for each point in the EEMC, and $meta$ is the central value of detector eta for each etabin. ϕ' is the modified azimuthal angle for each point. The azimuthal angle for each point in formula (3.10) is recalculated as modulo (6 degrees), because zero degrees are exactly located at the center of one subsector. The widths along the two dimensions were set to 0.035 radian and 2.3 degree based on simulation study (see following Chapter). The current setting requires that no more than four points can be found in the tower clusters region from a reconstructed π^0 . The next cut requires that both points have to exceed 1.5 GeV of transverse energy.

In the neutral pion analysis, there are basically four spin states of interest, depending on the RHIC beam polarizations: PP, PN, NP, and NN. Each spin state is indicated by the order of blue and yellow beams with positive or negative polarization. Digital numbers are applied to the four spin states, with 5 to PP, 6 to PN, 9 to NP, 10 to NN by STAR's definition [93]. The π^0 information stored in trees can be recalled by the type of Mix-Event of classes or objects for further analyses.

Chapter 4

Simulation Studies

4.1 Introduction

During the software development, many simulation studies have been performed to simulate the response of π^0 s and photons in the EEMC and improve the efficiency of the code. Our simulation studies focused on the energy response from the tower and SMD layers because these two kinds of detectors are critical in the reconstruction of neutral pions in the EEMC. In this chapter, we will show studies of the energy sharing of an electromagnetic shower among EEMC towers, the photon and neutral pion energy study in the EEMC, and the neutral pion reconstruction efficiency by the π^0 finder program with simulation data samples.

These simulation studies give us confidence in our π^0 finder software package. The original motivation for the energy sharing study among nearby towers was to understand quantitatively how the energy of photons will be distributed among nearby towers. The energy deposition information collected from detectors is important in the STAR experiment, because it helps us understand details of the detector response to electromagnetic interactions, and is vital in the π^0 invariant mass reconstruction. The pion reconstruction efficiency study will be the most important part of this chapter, because high reconstruction efficiency is essential for high statistics in the inclusive

neutral pion asymmetry measurement. Significant improvements in efficiency have been made for this software for our analysis of the 2006 data.

4.2 Energy sharing among EEMC towers

During the early period of developing the π^0 analysis software, there was concern about how to set up the framework for the π^0 finder program. It is important to understand how the energy of an e.m shower is distributed among nearby towers in the EEMC, because using the tower energy distribution information helps us decide how to develop the π^0 finder algorithm.

The Monte-Carlo samples were produced by the GEANT [94] generator at RCF. There are two kinds of data used in this simulation. One throws photons along the η direction with ϕ fixed at the center of a tower; the other throws photons along the ϕ direction with η fixed at the center of a tower, for example $\eta=1.541$ or 1.854 . For each type of data, ten hit points were chosen by evenly dividing the distance from the center of a tower to the center of an adjacent tower across the boundary. So when we throw 25 photons to each hit point along ϕ with fixed $\eta=1.541$, the separation between two points is $\pi/300$ radian. When we throw 25 photons to each hit point along η with fixed $\phi=0$, the separation is about 0.0097 radian. All photons thrown into towers have an energy of $E=9.5$ GeV. For each kind of data, we have ten sets of datasets, with one hit point per dataset. We then calculate the fraction of all detected energy on one side of the tower boundary for each dataset from the two kinds of data. When we show the energy fractions on plots, we then are able to generate ten points along the η direction and another ten points along the ϕ direction.

We were hoping that a single simple Gaussian function would be able to describe the energy sharing between two adjacent towers along both the η and ϕ directions, once geometric effects had been accounted for. But the shower profile appears to be different between the η and ϕ directions according to this simulation, and an asymmetric bivariate Gaussian function worked better to describe the energy distribution among neighboring towers. We show the energy fraction plots from both data types below.

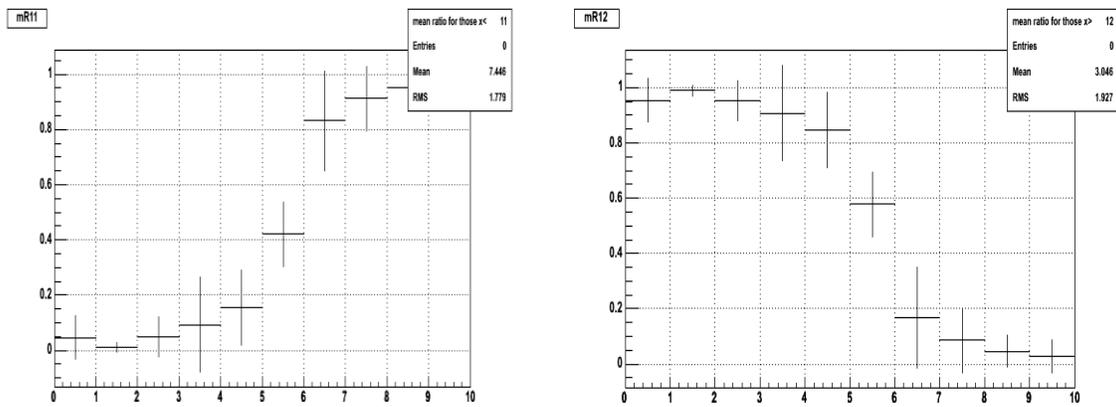


Figure 4.1: The energy fraction within two neighboring towers with fixed $\eta=1.541$ and along ϕ direction with even separation of $\pi/300$ radian.

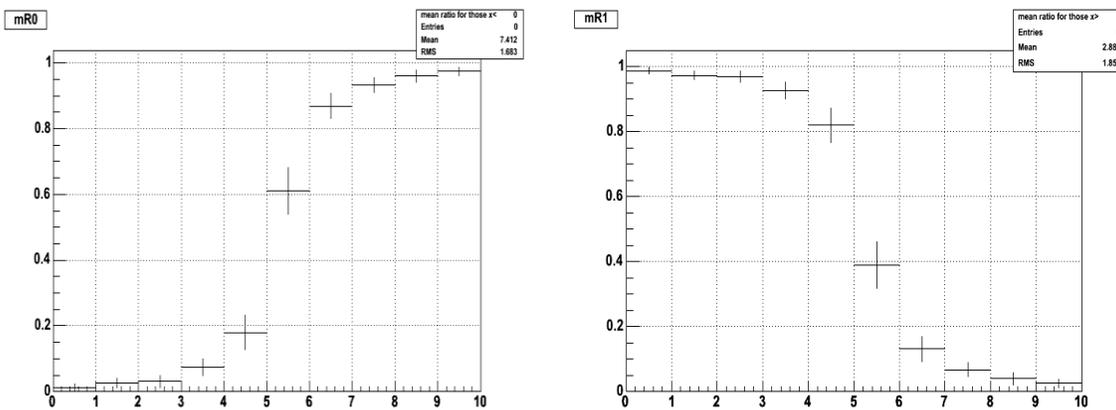


Figure 4.2: The energy fraction within two neighboring towers with fixed $\phi=0$ and along η direction with even separation of 0.0097 radian.

Figure 4.1 shows the energy fraction within two adjacent towers with fixed pseudo-rapidity at 1.541 along the φ direction from 0 to $9 \times \pi/300$ evenly, which basically crosses two neighboring towers with $\text{etabin}=5$. Figure 4.2 shows the energy fraction within two adjacent towers with fixed azimuthal angle at 0 along the η direction from 1.854 to 1.9413 in evenly spaced steps. The left and right plots in both figures correspond to two adjacent towers in the analysis. X-axes correspond to the ten datasets of the ten hit points within towers, and Y-axes are the relative energy fractions. As expected, we can see roughly complementary fractions of energy distributed in the two neighboring towers for each hit points in these plots, based on these center-hit datasets. We fit these datasets shown in the above figures and find that the distributions can be expressed as the integral over a Gaussian function along each direction, but they do not agree with each other in the width. The shower profile seems to have a larger width along the φ direction about 0.040 radian (2.3 degree) while the width along the η direction is about 0.035 radian. This difference can be used in setting the tower ET cut in the software during our neutral pion reconstruction, as we described in chapter 3 when we set a bivariate Gaussian threshold function around a tower. A more intuitive plot is drawn below to show in figure 4.3 the difference in shower widths along the two directions.

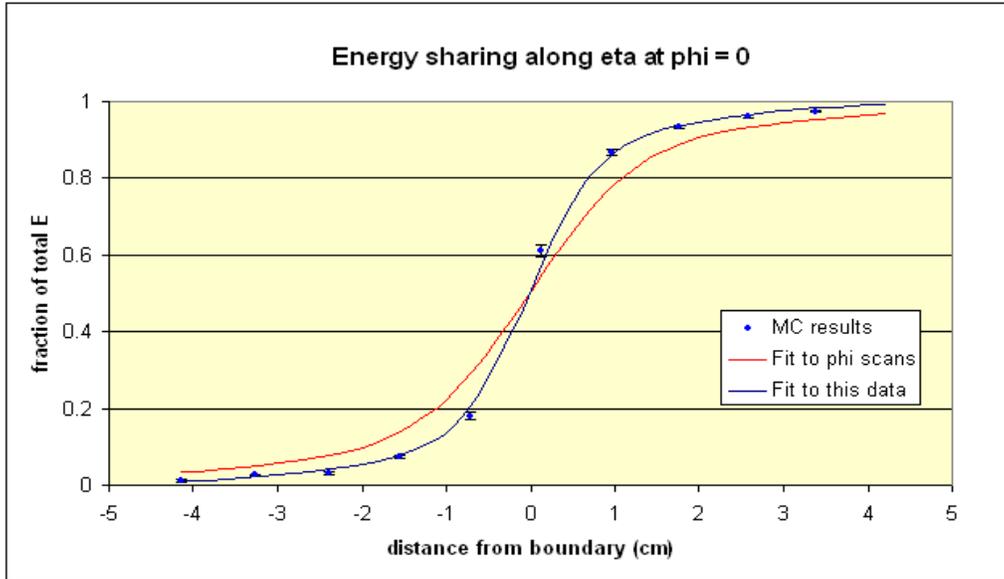


Figure 4.3: Energy sharing among nearby towers along η (blue curve) and ϕ (red curve).

Figure 4.3 shows the shower profile from the boundary point of view, and compares the two kinds of datasets in the same plot. The blue curve represents the fit to datasets along the η direction and the red curve represents the fit to datasets along the ϕ direction. This presents us with a problem: the energy sharing fraction and spread in energy (width) are a little wider in the ϕ -scan than in the η -scan. To track this problem, we generated two new Monte-Carlo samples: we threw 250 photons with energy $E=9.5$ GeV at each of two fixed points, at $(\eta, \phi)=(1.854, 0.04124)$ and $(\eta, \phi)=(1.8894, 0)$ in the EEMC. The two points are located exactly along the center lines in η and ϕ , respectively, but are 1 cm inside the tower boundaries in the other direction. We sum over all tower energies on each side of the corresponding boundary separately, and compare the energy distributions on the “far side” of the boundary event by event. The energy distributions on the far side from the two samples are shown below.

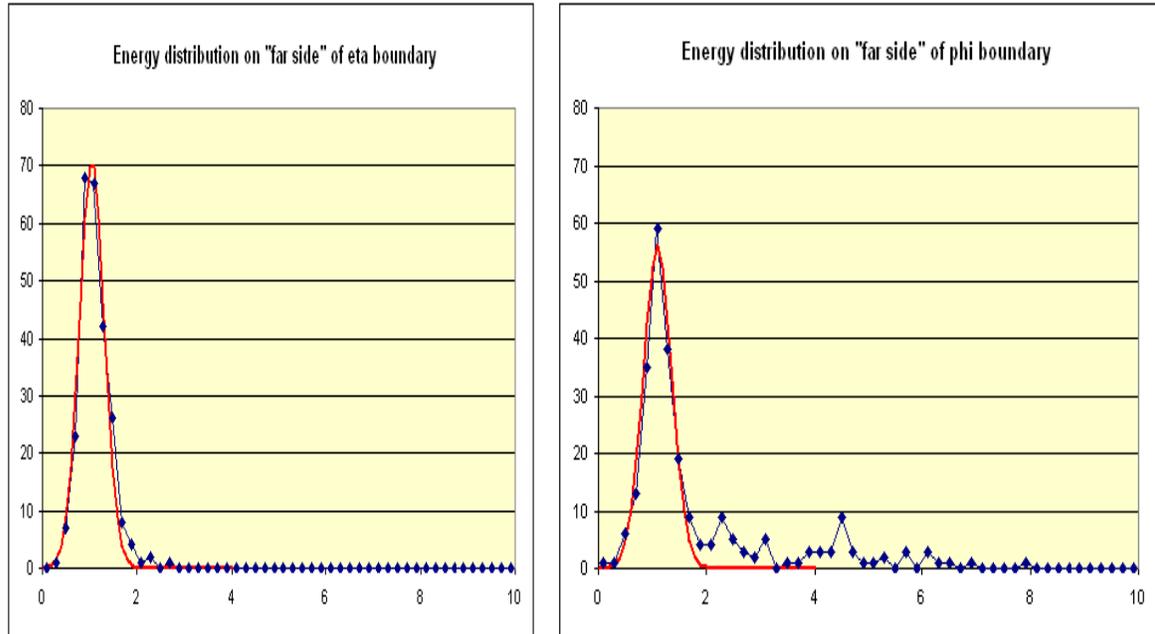


Figure 4.4: Energy distributions from far sides of tower boundaries.

The left plot of figure 4.4 shows the energy distribution from the 250 photons of the dataset at point $(\eta, \varphi)=(1.8894, 0)$ on the far side of an eta boundary. The energy collected across this boundary shows an approximate Gaussian distribution. The right plot of figure 4.4 shows the energy distribution from the 250 photons of the dataset at point $(\eta, \varphi)=(1.854, 0.04124)$ on the far side of a phi boundary. In this case, only 80% of the photon events have the same Gaussian distribution as shown on the left plot from the eta scan. From figure 4.4 we can see that there are two kinds of photon events in this simulation study: one has the same symmetrical appearance shape in both η and φ directions, and the other kind spreads out along the φ direction. The ratio for these two kinds of events in the φ -scan is about 4:1. The second kind of event gives us a wider distribution in energy sharing and width along the φ direction than the η direction. But it was curious that the second kind of event occurs only in the φ direction, and never in the η direction.

To investigate this difference, we re-ran the same simulations by throwing 250 photons at fixed points, but with different generation conditions. When we generated new MC samples, we first deleted all the materials upstream from the EEMC in the STAR geometry; for the second sample, we turned off the the 0.5 T magnetic field for the STAR detector. Four new MC samples (η -scan without materials, η -scan without magnetic field, ϕ -scan without materials, and ϕ -scan without magnetic field) were generated and analyzed. These showed that about 20% of the photons were converting in materials far upstream of the EEMC. The electron/positron pairs produced are emitted along the original photon momentum, and result in large opening angles along the ϕ direction because of the bending effects of the STAR longitudinal magnetic fields. This explains essentially all the features seen from this simulation study: the large energies observed on the far side of the phi boundaries go away when either the upstream material is removed (so no photons convert) or if the magnetic field is turned off (so the conversion pair do not separate in phi).

4.3 π^0 energy deposition study in the EEMC

When we convert the ADC values to electromagnetic energy information in the A2E-Maker, we must apply a sampling correction factor to our measurement of the energy deposited in all layers of the EEMC. It is also very important to check the linearity of the energy response from the π^0 's interaction in the EEMC, so that we are confident we are using the correct sampling correction factor. We start this study by probing the energy response of the EEMC during π^0 reconstruction.

Four Monte-Carlo data samples were generated in this simulation study. Ten thousand π^0 events were thrown in each sample with a flat EEMC distribution in ϕ (range $[0, 2\pi]$) and η (range $[1.1, 1.9]$). Based on the GEANT record, the distribution looks as shown in figure 4.5 below.

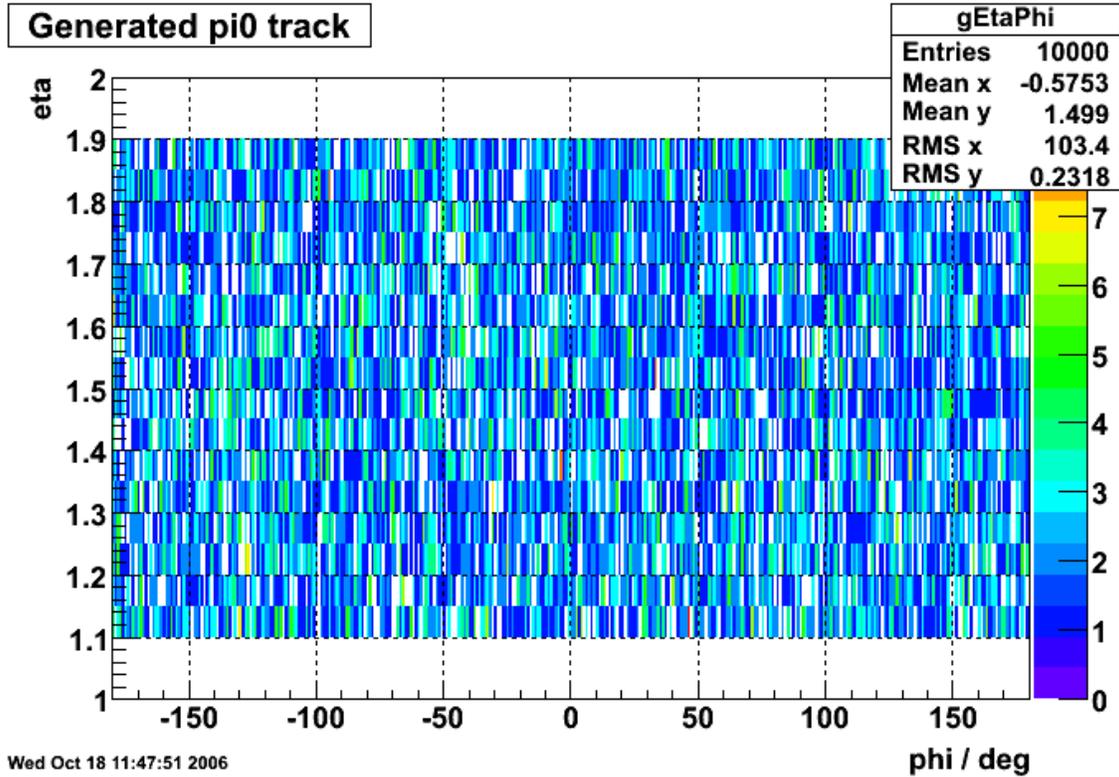


Figure 4.5: The uniform distribution in the EEMC for the four MC samples in the energy response study.

The four MC samples differed from each other due to other features in the generation. We made sample 1 (π^0) with a flat energy generation of $[5, 60]$ GeV, sample 2 (π^0) with a flat transverse momentum distribution of $[5, 30]$ GeV/c, sample 3 (γ) with the same flat energy distribution as sample 1 and all upstream materials included in STAR, and sample 4 (γ) like sample 3 but without materials. To test the energy response in the EEMC, we plot the ratio between the reconstructed and generated energies of the

events below. To reconstruct the energy of an event, we basically sum over all tower energies to get the value.

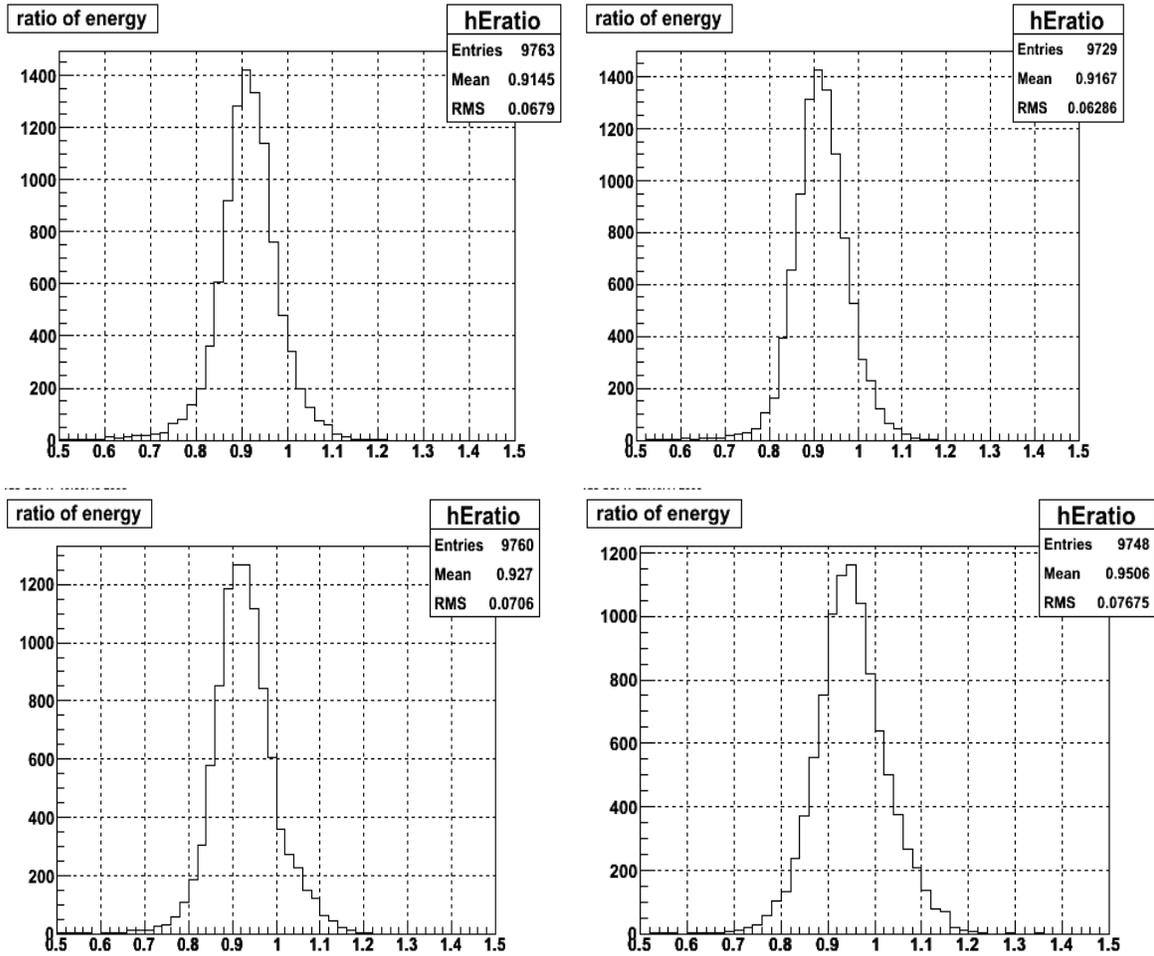


Figure 4.6: ratio=reconstructed energy/generated energy. The four plots in the order of upper-left, upper-right, lower-left and lower-right correspond to MC samples 1, 2, 3 and 4, respectively.

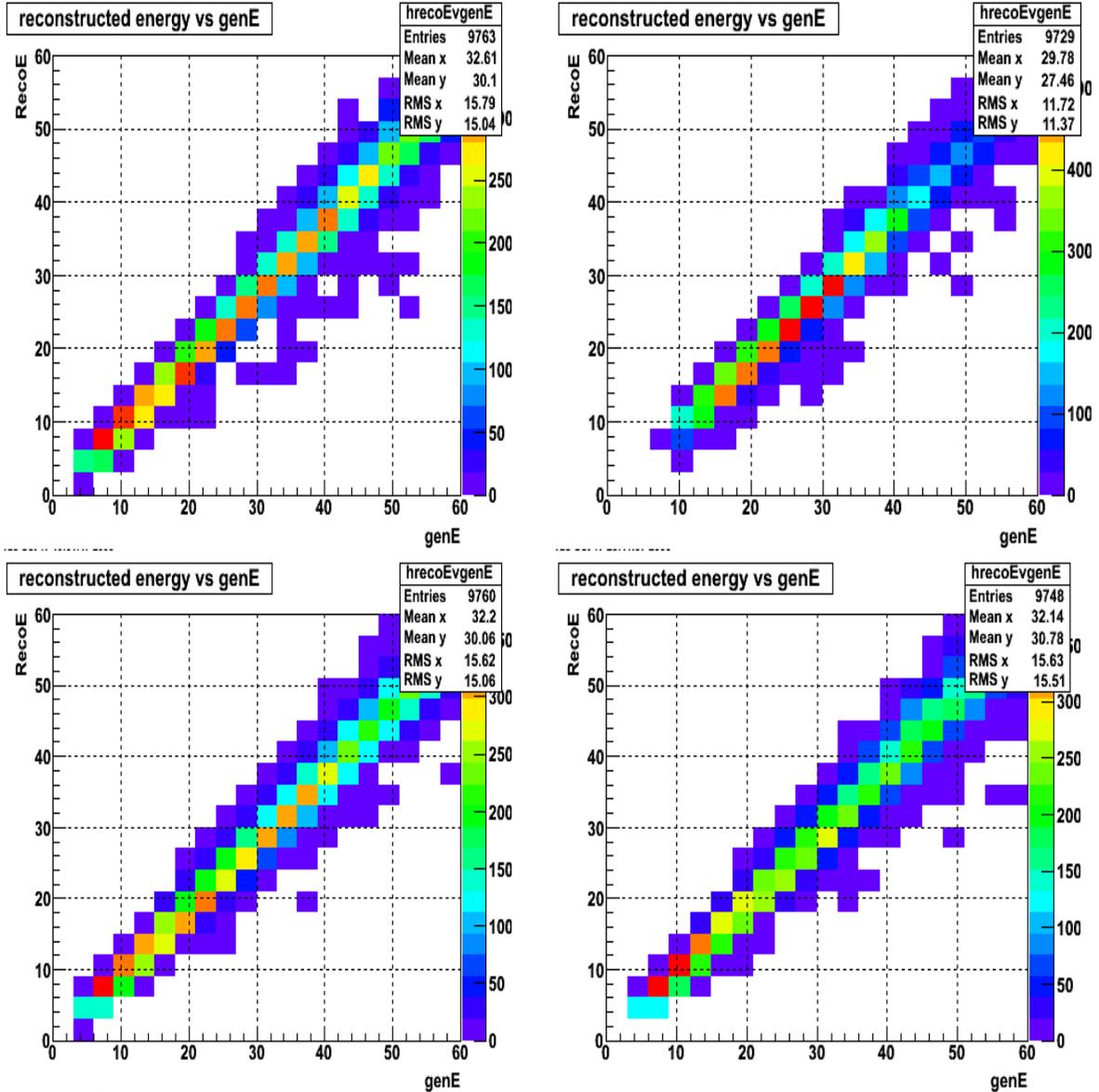


Figure 4.7: Two dimensional plots of linearity between reconstructed and generated energies. The four plots in the order of upper-left, upper-right, lower-left and lower-right correspond to MC samples 1, 2, 3 and 4, respectively.

The energy response information is illustrated in figures 4.6 and 4.7. The energy ratios between reconstructed and generated events are symmetrically distributed as seen in figure 4.6. The linearity of the response from the EEMC towers is shown in figure 4.7. For this specific simulation study, the energy ratio cut $[0.5, 1.5]$ is applied (see figure 4.6). It is necessary to check the linearity in detail before we decide on the sampling correction

factor to use in the A2E-Maker, although we can roughly calculate the value from figure 4.6. The first two MC samples in this study give us the most general condition in the STAR experiment, because they have flat energy and pT distributions, with all materials in front of the EEMC. The two plots from these two samples show the mean ratio at about 0.915. When we generated these samples, a sampling correction factor of 1.2 had already been applied to the reconstructed energy when we convert ADC values to energies. So these simulation studies suggest that the true sampling correction factor should be roughly $1.2/0.915=1.3$ if the linearity of the energy response is good enough for our π^0 analyses.

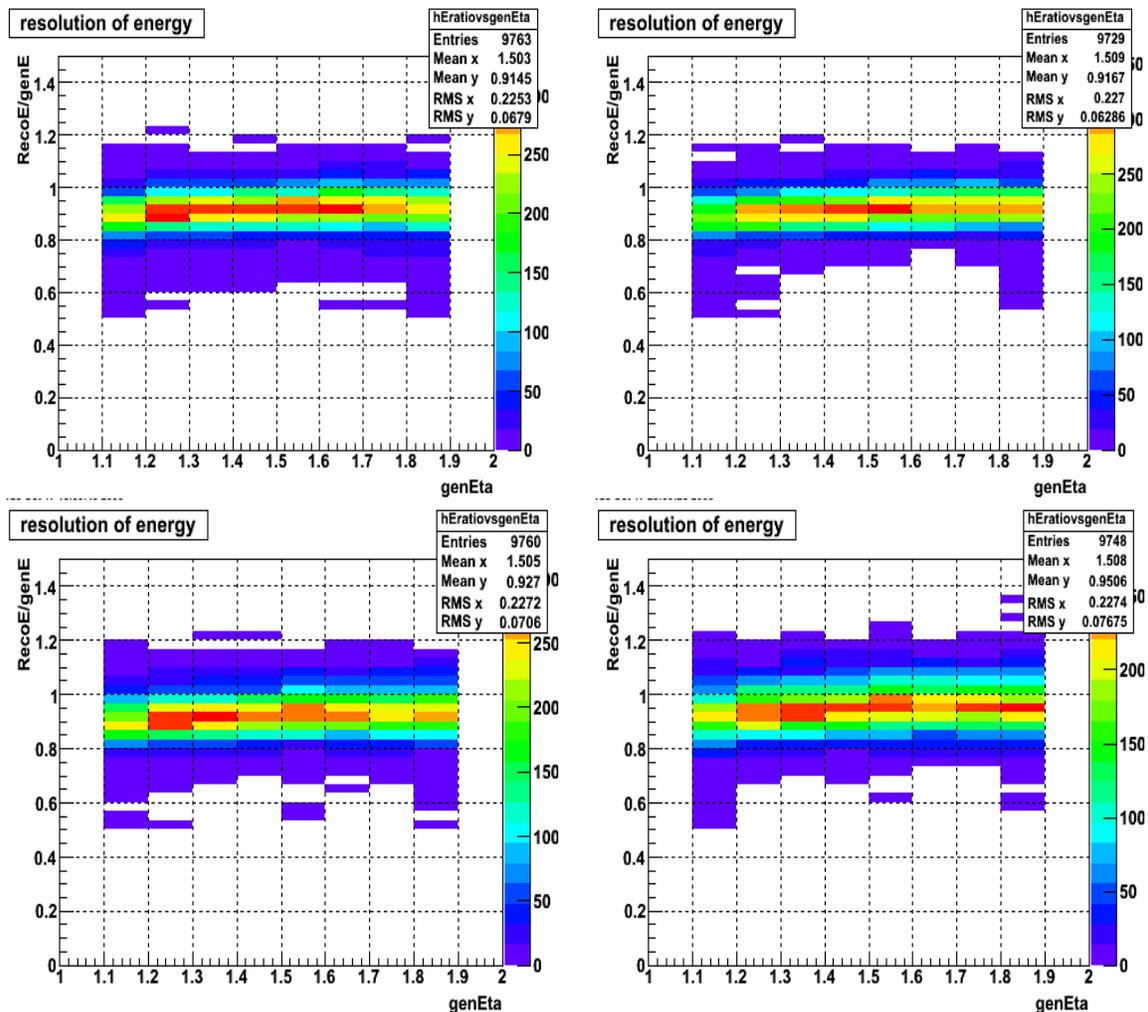


Figure 4.8: The ratio of reconstructed energy/generated energy as a function of generated η . The four plots in the order of upper-left, upper-right, lower-left and lower-right correspond to MC samples 1, 2, 3 and 4, respectively.

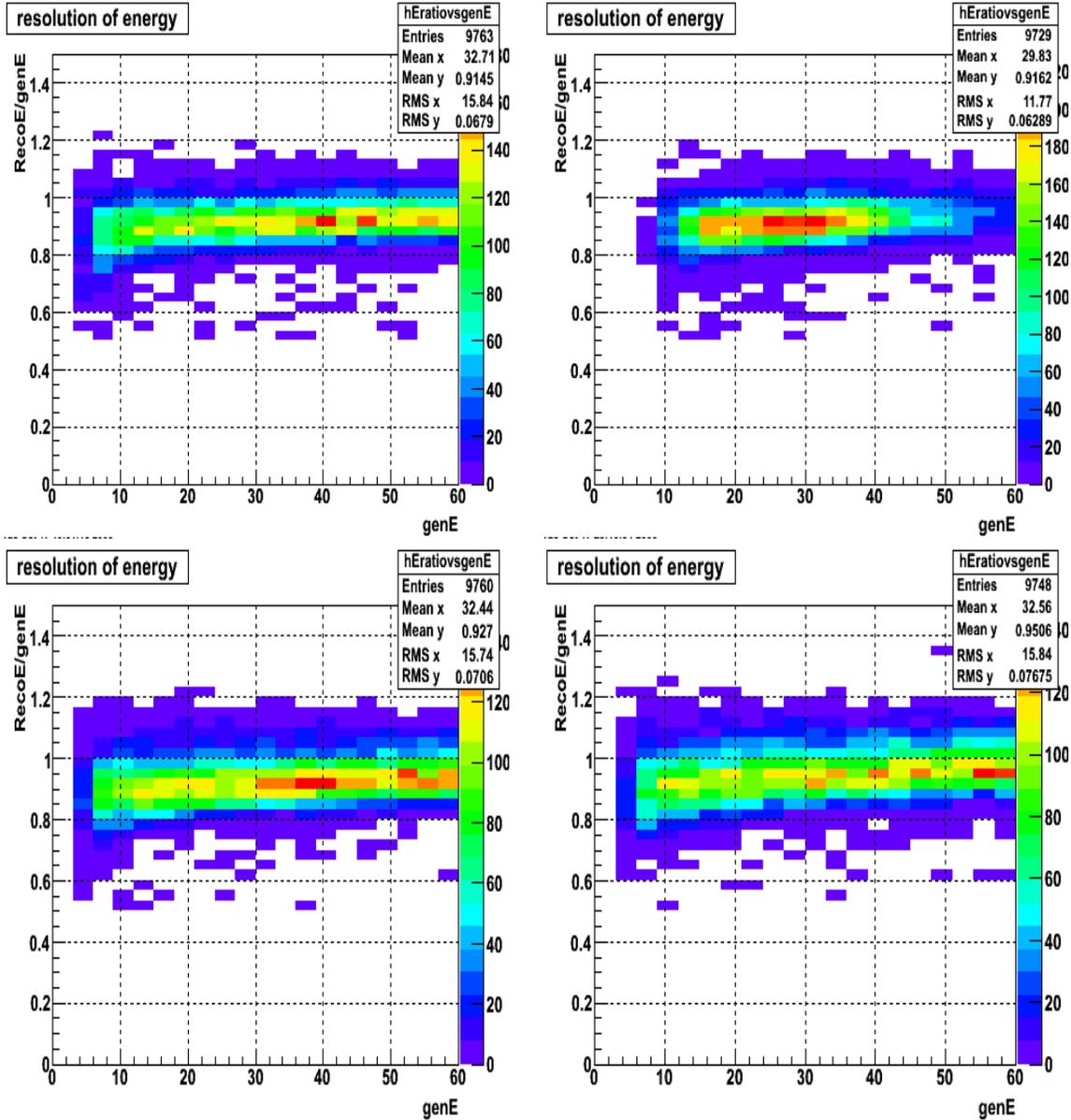


Figure 4.9: The ratio of reconstructed energy/generated energy as a function of generated energy. The four plots in the order of upper-left, upper-right, lower-left and lower-right correspond to MC samples 1, 2, 3 and 4, respectively.

We check the linearity of the π^0 energy response in the EEMC vs. the generated pseudo-rapidity and event energy, as shown in figures 4.8 and 4.9. When we show the generated energy in this chapter, the unit is always GeV. Based on figure 4.8 and 4.9, we

can scan the profile for each bin of generated η and energy, and then fit these by various functions. We performed fits using constant, linear, and quadratic functions for the profiles from figures 4.8 and 4.9, and results show that the linear fits were reasonable in most cases. We only show the fits from linear functions for the four MC samples. The simple linear function $p_0 + p_1 \times x$ was fit and the results are shown below.

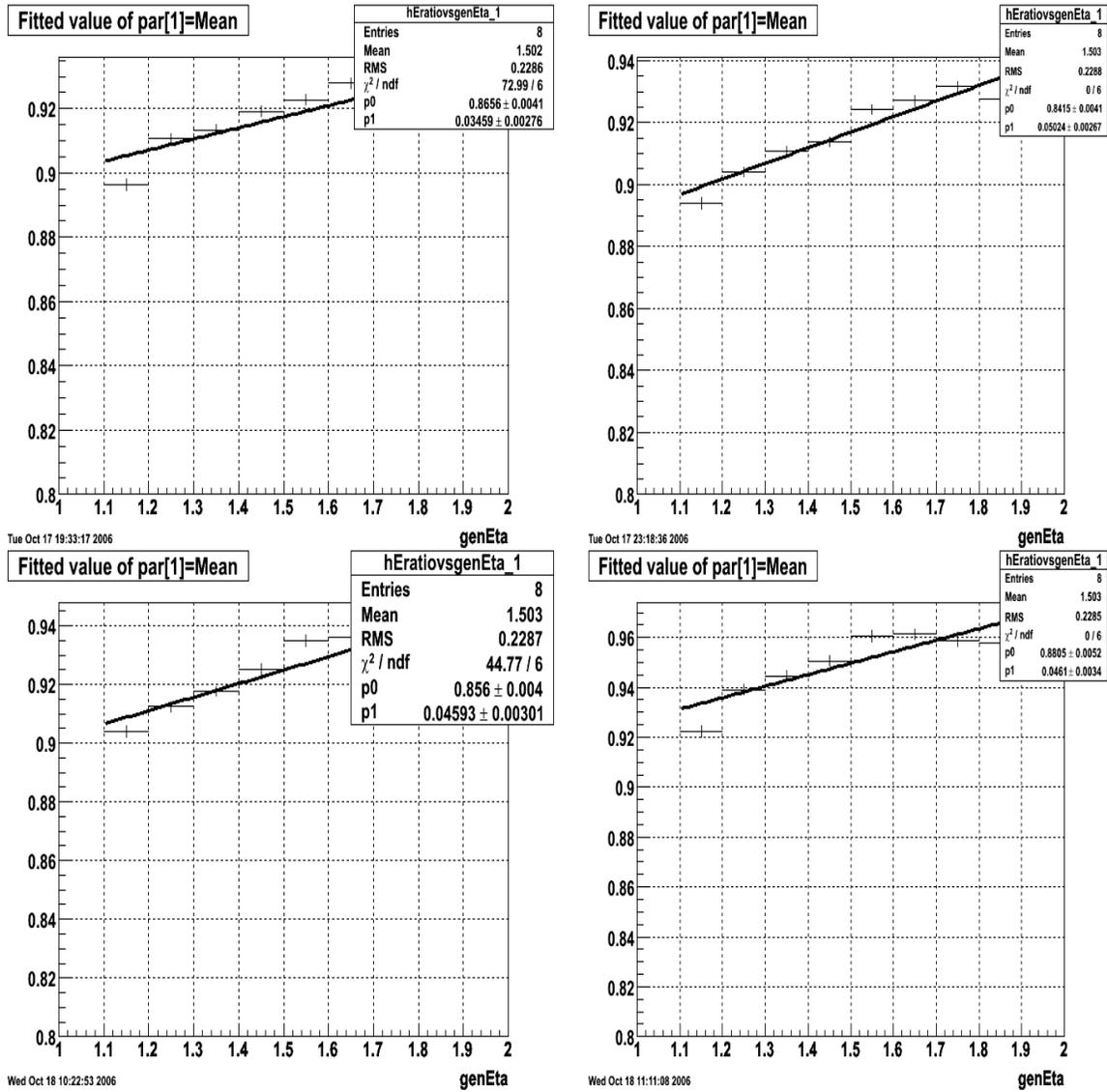


Figure 4.10: π^0 eta dependence of sampling fraction in the EEMC based on figure 4.8. The four plots in the order of upper-left, upper-right, lower-left and lower-right correspond to MC samples 1, 2, 3 and 4, respectively.

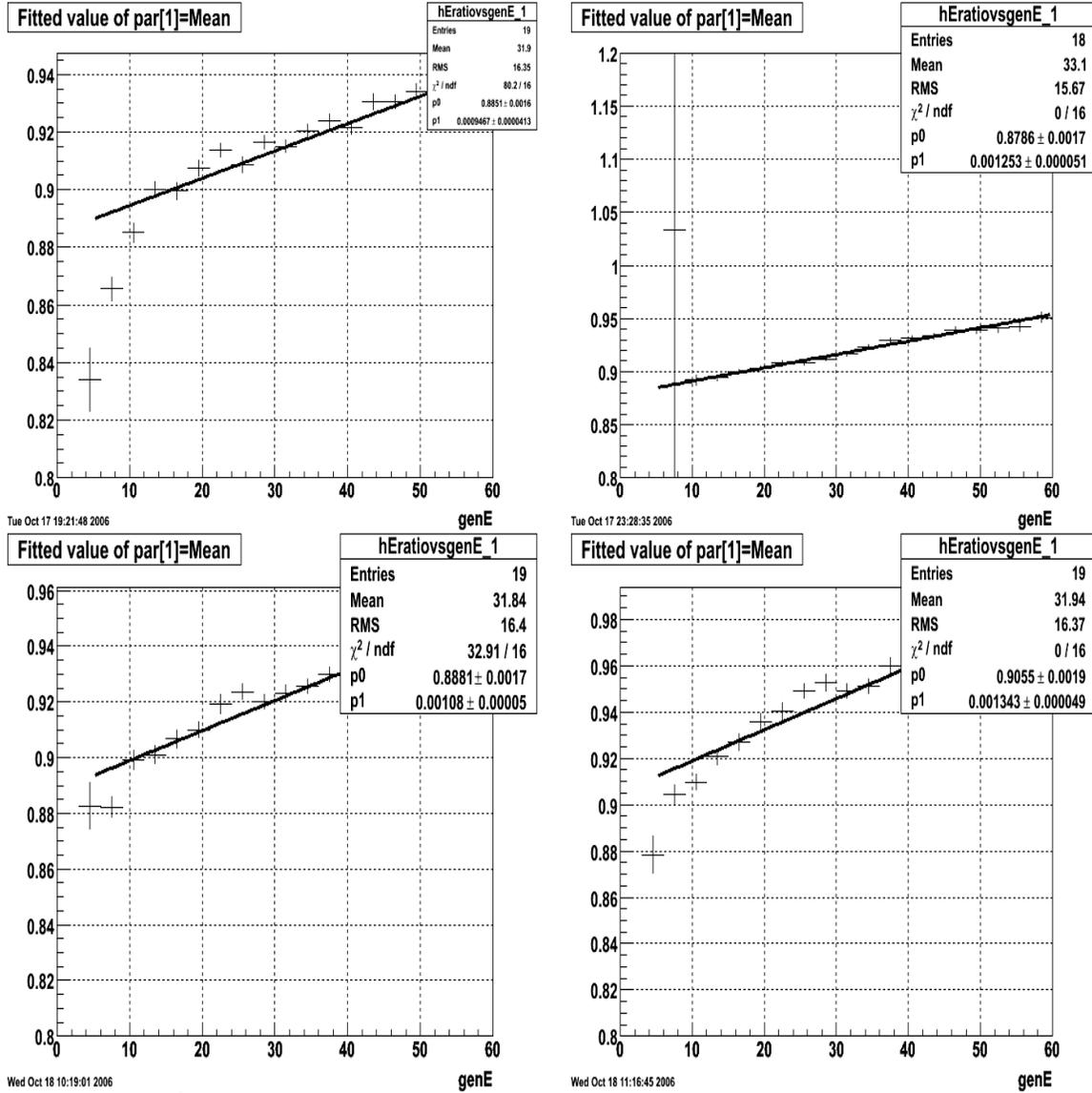


Figure 4.11: π^0 energy dependence of sampling fraction in the EEMC based on figure 4.9. The four plots in the order of upper-left, upper-right, lower-left and lower-right correspond to MC samples 1, 2, 3 and 4, respectively.

The fitting results in figure 4.10 suggest a small (2-3%) eta-dependence in the sampling correction factor. The fitting results in figure 4.11 shows more complicated behavior in the energy response as a function of generated energies. This non-linear effect is most pronounced in the low energy region, so it won't have much influence on our inclusive neutral pion study. Based on these studies, we therefore set the sampling correction factor to a constant 1.3 in the A2E-Maker during the later neutral pion analysis.

Nevertheless, it is disturbing that the sampling fraction found in simulation differs from our expectations by close to 30%. The solution to this problem is not a direct part of the π^0 reconstruction work, so we have not done much work to solve it so far. Before we use the A2E-Maker in the π^0 finder program, the slow simulator, which is used to simulate the digital response of the EEMC, is called to modify the ADC information from the MuDst files using the database settings. So the slow simulator is the right place to work more in order to fix the problem.

For the π^0 reconstruction, we have done some extra work related to the tower and pre-shower energies in order to get a linear and constant response from the EEMC deposited energies. After a comprehensive study of the tower, SMD, pre-showers, and post-shower energies, we can achieve pretty constant response for the reconstructed energy if we apply a transfer formula:

$$E'_{reco} = E_{tower} + f \times (E_{pre1} + E_{pre2}) \quad (4.1)$$

where E'_{reco} is the new reconstructed energy; E_{tower} is the tower's energy; and f is an empirical factor to the pre-shower 1 and 2 energies E_{pre1} and E_{pre2} . This formula was not applied in our π^0 analyses later because this modification does not solve the fundamental problem in the slow simulator, and the energy response in the EEMC is understood well enough for the inclusive neutral pion study. But the above idea is presented here to offer a possible clue for future software development.

According to this simulation study, 1.3 will be the new sampling fraction correction factor during the inclusive π^0 reconstruction and data analysis in this thesis.

4.4 π^0 reconstruction and efficiency study

The reconstruction of π^0 s in the EEMC using the previous π^0 finder software by Jason Webb [95] had been a very tough job because of the low reconstruction efficiency. After the software development described in Chapter 3, the reconstruction efficiency has been improved greatly, from several percent to 40%-60% on different types of simulation data samples. Some features of the reconstruction are also more reasonable. In this part, we will discuss the reconstruction results from three simulation data samples: single thrown π^0 events, di- π^0 events, and a pythia events sample.

The di- π^0 sample has many of the complexities of real data in π^0 production in the EEMC during runs. We start the introduction for this sample. The di- π^0 sample is generated by the GEANT generator at RCF with a flat energy of [5, 60] GeV distribution. The total number of events generated for this sample was ten thousand, so that the total number of thrown π^0 s was twenty thousand. Events were thrown flat in η [1.1, 1.9] and ϕ [20, 40] degree. We chose to narrow the π^0 s into a small ϕ range because we wanted to simulate two π^0 s close to each other. If the π^0 finder software can reconstruct π^0 s with a high efficiency with this Monte-Carlo sample, it will have a higher efficiency for the single π^0 MC sample, and give us confidence in real data analyses.

The single π^0 Monte-Carlo sample was also generated by the GEANT generator at RCF with a flat energy distribution of [5, 60] GeV. But we generated only a thousand events in this sample, with one π^0 in each event. It has the same η range of [1.1, 1.9] as the di- π^0 sample, but a much broader ϕ range [-180, 180] degree. This sample provides us with a consistency check about the reconstruction efficiency of the code.

The pythia sample contains two hundred thousand events and is stored at HPSS. This specific pythia sample simulates collisions in the STAR detector with partonic

transverse momentum p_T of [25, 35] GeV/c. The environment in this sample is very close to real data runs, and the analysis results for this sample should give us the best estimate on π^0 reconstruction quality and efficiency for real data.

We introduced the main classes for the π^0 finder program in Chapter 3. When we run the π^0 finder program, we have to use a main macro to chain over all classes and all events. The main macro is called `runEEmcMcPi0Mixer` for the simulation studies. It basically analyzes a `MuDst.root` file, and saves the result into a new root file with the π^0 tree stored. The macro loads the STAR library for programming first, and then creates the analysis chain. The chain starts with an `ioMaker` to activate the GEANT branch so that we can track the generated MC sample information. A `MuDst-Maker` is then used for reading input files. We connect to the STAR database so that we can call the information required in the π^0 finder software. For MC studies, the ideal gains are used for analysis. Then we initialize the EEMC database and the spin-database maker. For MC studies, the slow simulator maker is also activated before we go to the π^0 finder classes we developed. Now we can set up or change most of the cuts and threshold values for the reconstruction of π^0 s according to the classes of the π^0 finder program.

In the A2E-Maker part, we set the width cut to 3 for all layers of the EEMC, so that the ADC value has to pass the pedestal value by 3 sigma in the database. We set the sampling correction factor to be 1.3 for the tower energy response. In the Cluster-Maker part, the seed threshold for towers is set to 0.8 GeV, and the seed thresholds for both SMD planes are set to 1.5 MeV, which is just above the MIP deposition of about 1.4 MeV. The floating floor parameter is set to 1. The maximum extension from a SMD seed strip is set to 3, so that we have a total of 7 strips in a SMD cluster. In the Point-Maker

part, we select the energy mode to share the tower energies proportional to the SMD clusters. In this Mix-Maker part, we have an option to fix the vertex for MC samples at zero, but we use the actual vertices for the above three samples during our reconstruction of π^0 s. Then the McOutputMaker is called to give us a number of histograms describing the generation of the MC samples. In the Pi0-Analysis part, we decide to use the information from the above makers, so that we can do the final π^0 reconstruction here. In this final class, the tower ET cut is set to 3 GeV for at least one of the points from a reconstructed π^0 candidate, and the η cut is set to [1.086, 2.0]. We do not require the 1.5 GeV point pT cut for the simulation studies, though this was used for the real data analysis.

We will discuss the π^0 reconstruction and efficiency studies from the three simulation data samples in the following sections with the setup described above. The detailed analysis for the di- π^0 sample will be described first to illustrate how the reconstruction compares with the GEANT record, and how the efficiency is calculated. Then we will only show some important features and figures from the analyses of the single π^0 and pythia samples, to demonstrate that there is consistency in our reconstruction efficiency from other kinds of simulation data sample.

4.4.1 di- π^0 Monte-Carlo sample analysis

As we developed the π^0 finder software to improve our reconstruction efficiency, we knew that only when we achieved a high efficiency in reconstructing π^0 s in a complicated simulation sample, then we might have a good reconstruction efficiency for real data. So we start with this special MC sample for our simulation studies.

During this analysis, we can examine the GEANT record and get the generated information for the MC sample. Some basic features of the generated $\text{di-}\pi^0$ sample are shown below.

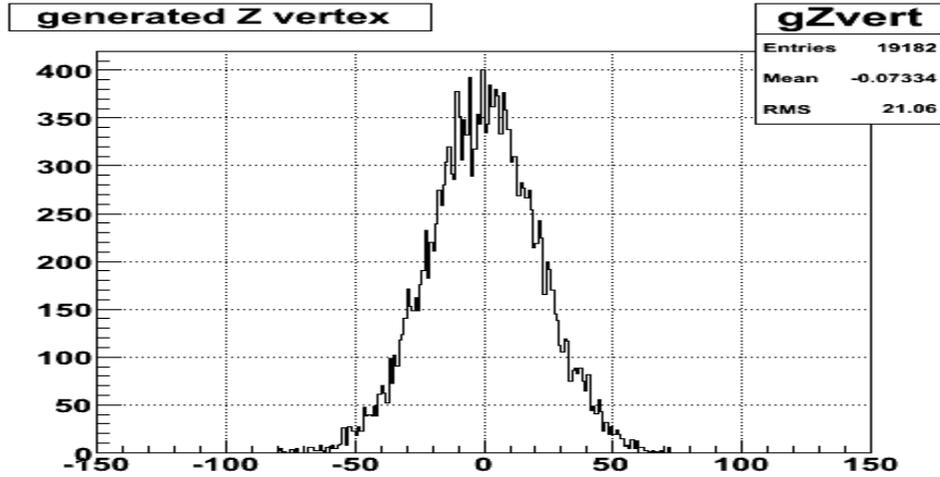


Figure 4.12: Generated Z vertex for the $\text{di-}\pi^0$ sample with a Gaussian distribution centered at zero and a width given by $\sigma=30/\sqrt{2}$.

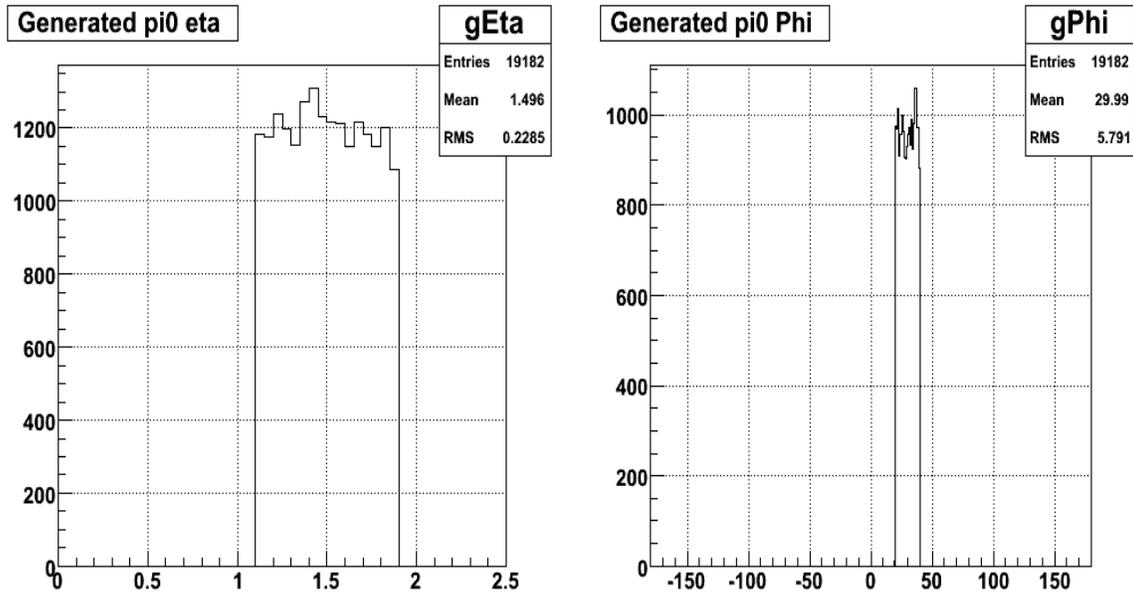


Figure 4.13: Geometry distribution in the EEMC for the $\text{di-}\pi^0$ sample.

We show the vertex and EEMC geometry information in generating the $\text{di-}\pi^0$ sample in figures 4.12 and 4.13.

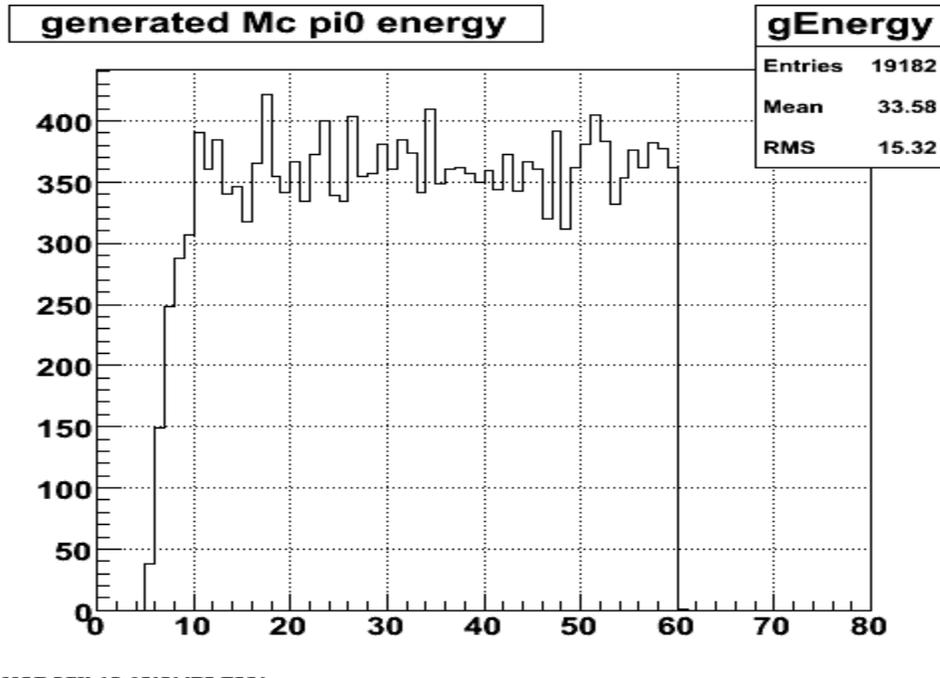


Figure 4.14: Generated energy distribution of the di- π^0 sample.

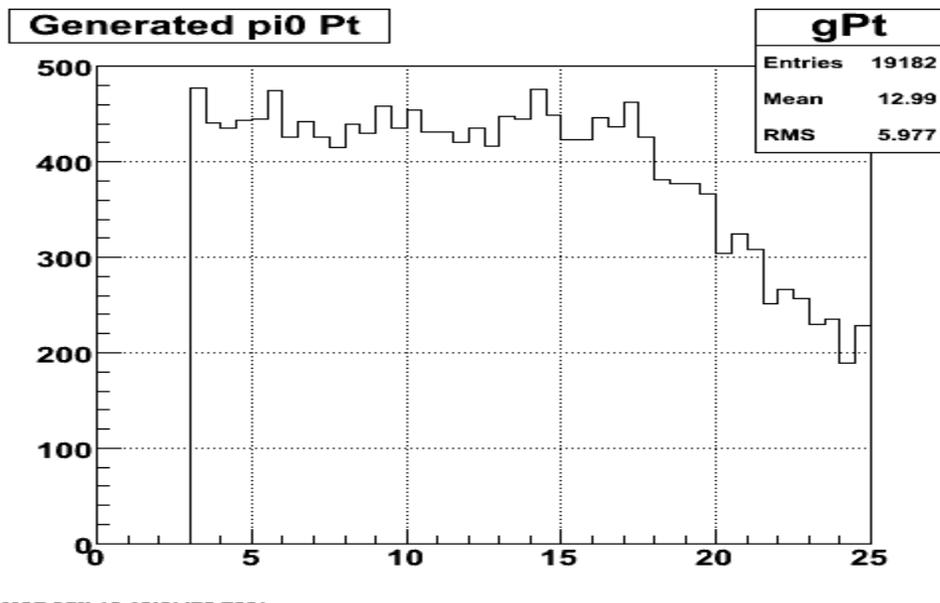


Figure 4.15: Generated transverse momentum distribution of the di- π^0 sample.

The important features of the generated energy and transverse momentum p_T distributions of this sample are shown in figures 4.14 and 4.15. These two plots are eventually used in the efficiency calculation for the π^0 reconstruction.

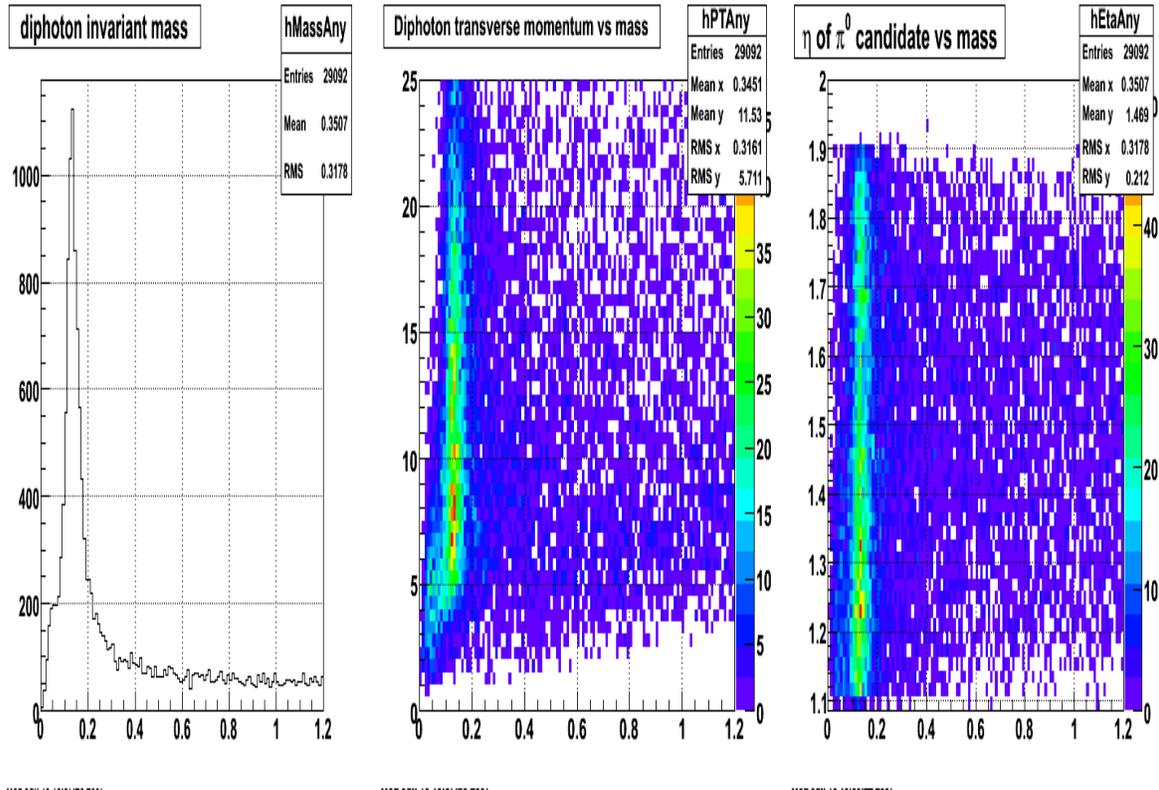


Figure 4.16: Invariant mass spectra peaked at 0.135 GeV from the $di-\pi^0$ sample. These events represent reconstructed π^0 candidates which satisfied various setups in the Pi0-Analysis class and are stored into the tree.

As we discussed in Chapter 3, we will have a large number of π^0 candidates which satisfy various conditions, and are stored in the π^0 tree as shown in figure 4.16. Figure 4.16 includes three plots showing the quality of the reconstructed π^0 candidates before the efficiency study. The left plot shows the one-dimensional invariant mass spectrum peaked at 0.135 GeV from the $di-\pi^0$ sample, with the x-axis denoting the mass in GeV

and the y-axis denoting the π^0 yield. The middle plot shows the di-photon reconstructed transverse momentum in GeV/c along the y-axis versus invariant mass along the x-axis. The right plot shows the di-photon reconstructed pseudorapidity along the y-axis versus invariant mass. These three plots show that we have reconstructed the π^0 s with a reasonable narrow peak shape, and the π^0 transverse momentum pT and its pseudorapidity. It is also noticeable that we reconstruct a long tail on the right side (high mass) of the left plot and a bump in the low mass region. The long tail at high mass is produced from the combinatoric background when we combine unrelated point pairs to form neutral pions. The low mass bump is seen to arise mostly from low pT events, which is illustrated in the middle plot. This will turn out to be a more severe problem in the real data analysis.

We obviously do not want to include all the reconstructed π^0 candidates which are stored into the π^0 tree in our quality and reconstruction efficiency studies. Additional criteria must be added to filter these π^0 candidates so that the survivors passing these strict criteria can be reliably used in the efficiency calculation. The first additional cut is that we require the invariant mass to be from 0.08 to 0.18 GeV. After the invariant mass cut, we then test for a matching between the reconstructed π^0 and the generated one in this event. To match the position, we use the η and ϕ information from both the reconstructed π^0 and the generated π^0 (from the GEANT record) to calculate the distance between them.

$$\Delta\eta = |\eta_{recon} - \eta_{gen}| \quad (4.2)$$

$$\Delta\phi = |\phi_{recon} - \phi_{gen}| \quad (4.3)$$

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} \quad (4.4)$$

where η_{recon} denotes the reconstructed π^0 eta, η_{gen} denotes the generated π^0 eta, and φ_{recon} and φ_{gen} denote the reconstructed and generated π^0 phi, respectively. The reconstructed and generated η and φ are all based on the momentum vector information, so for η , this is the particle pseudo-rapidity and not the detector η . In this case, there will be no problem to calculate the distance using both η measurements from momentum; but if the option to fix the Monte-Carlo sample vertices at zero is activated during the analysis, it is important to calculate the detector eta $\eta_{detector}$ to replace the η_{gen} and then calculate the distance. The detector EEMC $\eta_{detector}$ can be calculated as

$$\eta_{detector} = -\log(\tan((\text{atan}(\frac{H}{270}))/2.0)) \quad (4.5)$$

where 270 cm is the distance from the center of STAR to the EEMC front surface. The distance H is calculated by:

$$H = (270 - V_z) \times \frac{p_T}{p_z} + \sqrt{V_x^2 + V_y^2} \quad (4.6)$$

where V_x , V_y and V_z are the x, y and z components of the vertex for the event, and p_T and p_z are the transverse and longitudinal momenta. These two formulae are also used in the calculation of detector eta when we apply the eta cut [1.086, 2.0] to reconstruct π^0 s.

For this specific sample, there are two π^0 s in each thrown event. After the invariant mass cut, we calculate the distance ΔR for each reconstructed π^0 from the two thrown MC π^0 s, and choose the closer one. Then we store the distance between the reconstructed π^0 and the closer generated π^0 in a histogram. The distribution is shown in the following figure.

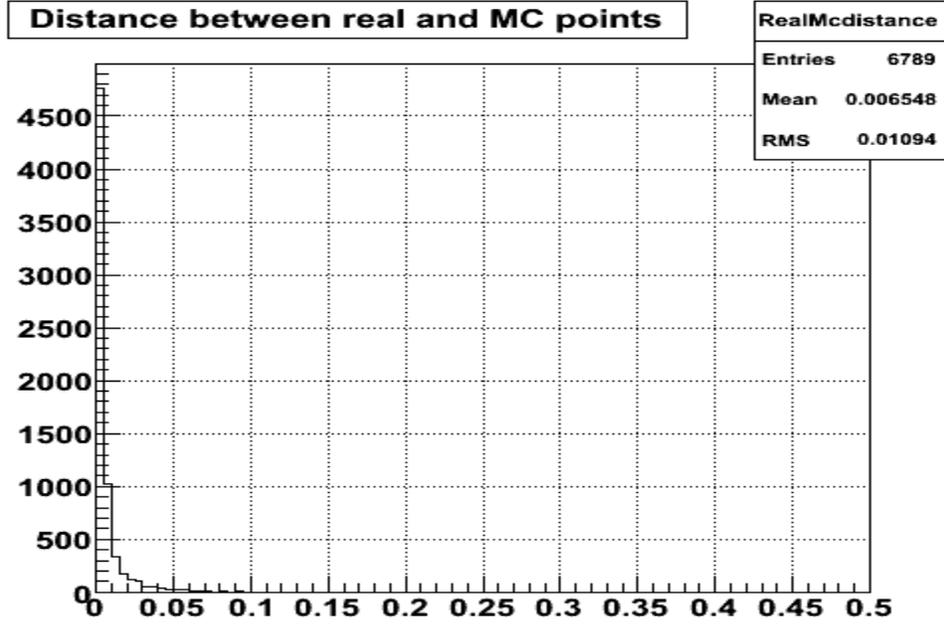


Figure 4.17: Distance ΔR between the reconstructed π^0 and the closer generated π^0 from our di- π^0 sample.

To match the position, we choose a distance cut of $\Delta R \leq 0.02$ for this di- π^0 MC sample based on in figure 4.17. For each reconstructed π^0 candidate, we match the position with either of the two thrown MC π^0 s from the GEANT record, and if the distance between them is less than or equal to 0.02, we consider this final reconstructed π^0 a good one. We then save the reconstructed η , ϕ , pT, energy, $Z_{\gamma\gamma}$, and opening angle information into histograms. Furthermore, the MC information η , ϕ , pT, energy, $Z_{\gamma\gamma}$, opening angle, and z vertex from the matched generated π^0 will be also saved into another set of histograms. The matched MC information from the generated π^0 s will eventually be utilized in the calculation of our reconstruction efficiency, instead of using information from the final reconstructed pions.

Some of the final reconstructed π^0 and matched MC generated π^0 features are shown below, with the reconstructed data on the left and matched MC on the right.

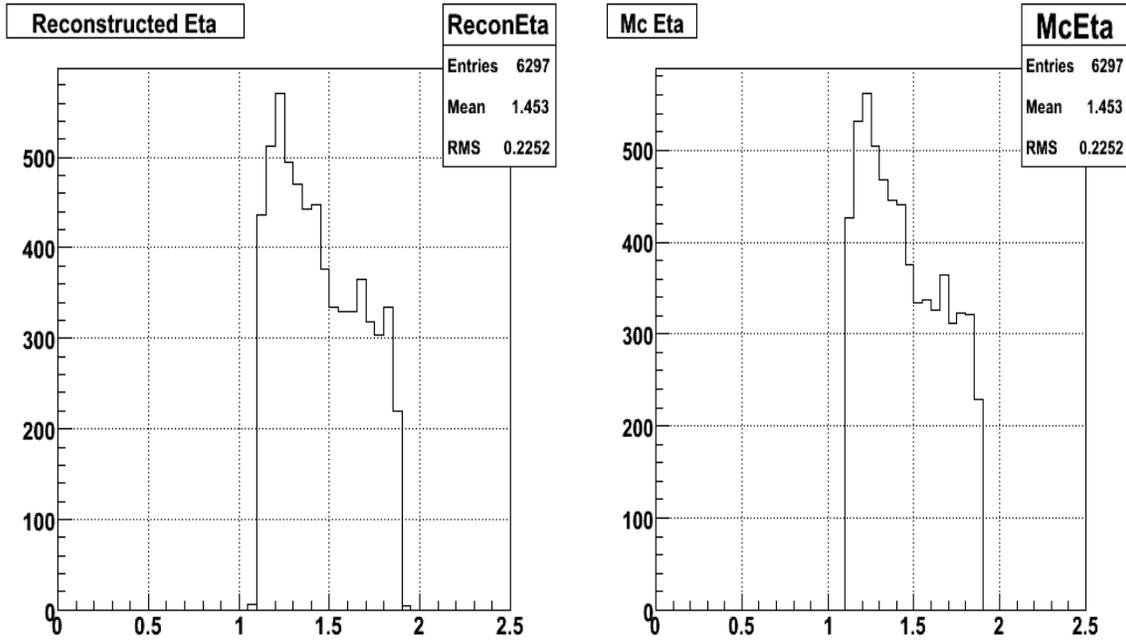


Figure 4.18: η distributions from the match between reconstructed and generated MC π^0 .

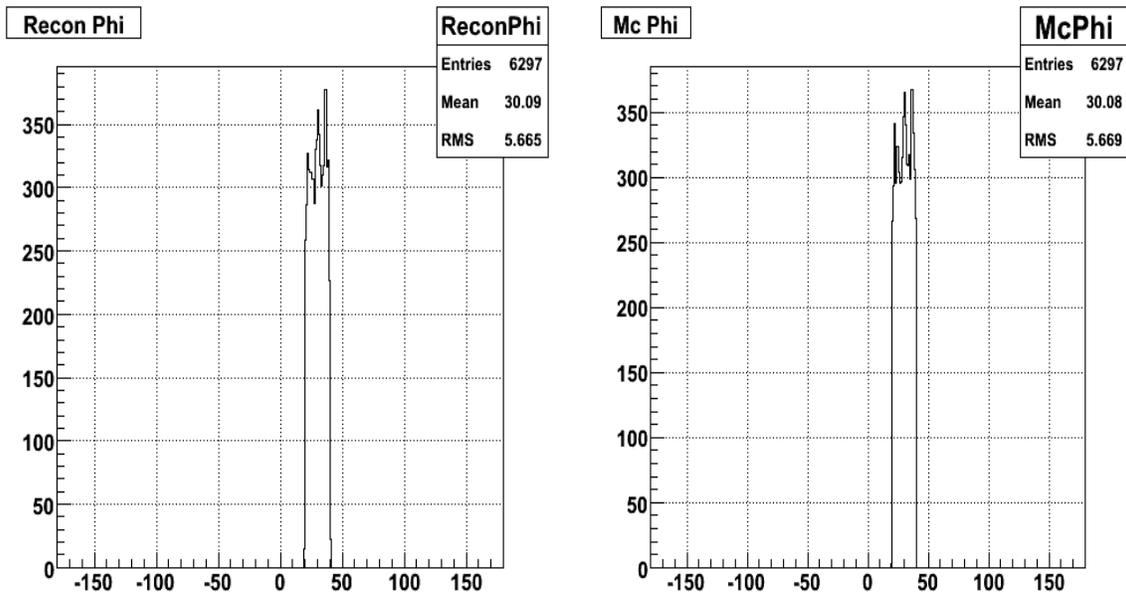


Figure 4.19: ϕ distributions from the match between reconstructed and generated MC π^0 .

The geometry distributions in figures 4.18 and 4.19 comparing the reconstructed π^0 and matched generated MC π^0 should be close to each other because of the tight distance cut. Two important figures in the efficiency calculation are shown below.

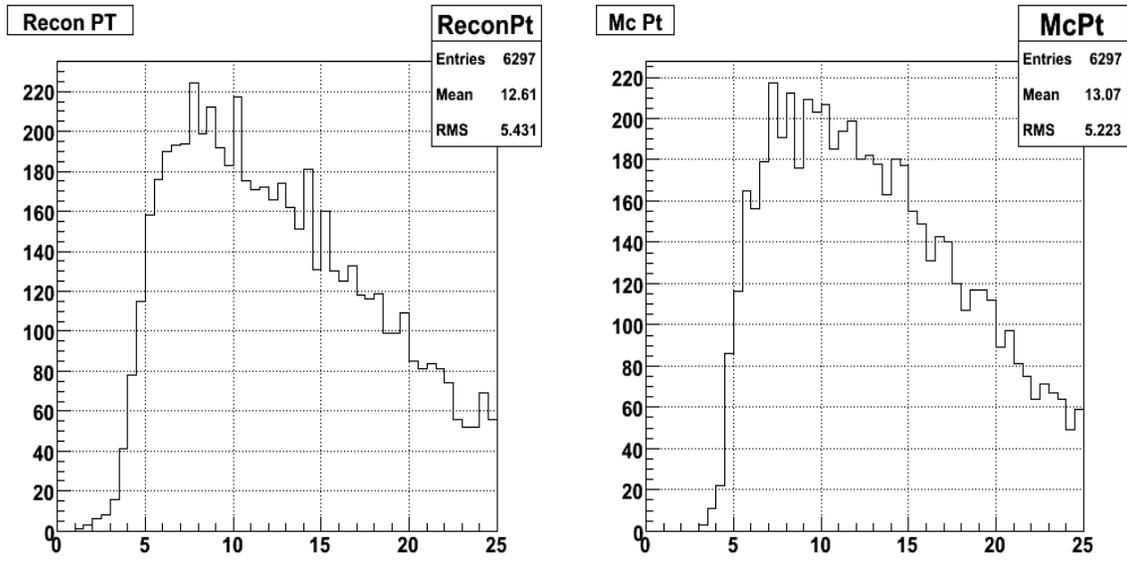


Figure 4.20: pT distributions from the match between reconstructed and generated MC π^0 .

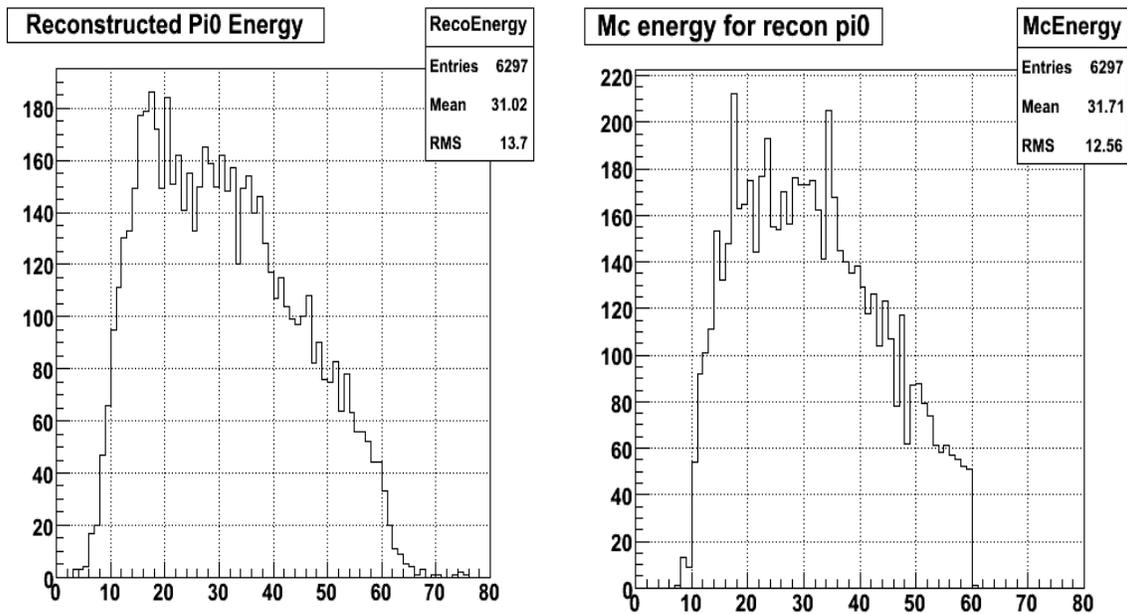


Figure 4.21: Energy distributions from the match between reconstructed and generated MC π^0 .

Now we have all the necessary information to calculate the π^0 reconstruction efficiency of our π^0 finder program for the di- π^0 sample. The generated MC π^0 transverse momentum and energy information that are matched to reconstructed π^0 s are shown on the right in figures 4.20 and 4.21, respectively. The full generated π^0 transverse momentum and energy distributions are shown in figures 4.15 and 4.14. We present the reconstruction efficiency in two ways: as a function of transverse momentum pT and as a function of energy E. The basic idea is to divide the right plots in figures 4.20 and 4.21 by figures 4.15 and 4.14, respectively, so we calculate the efficiency from the matched MC π^0 instead of the reconstructed one. For the efficiency versus pT calculation, we take the count n1 from the same channel of the right plot of figure 4.20 and the count n0 from each channel of figure 4.15. The efficiency for each channel is calculated by:

$$efficiency = \frac{n1}{n0} \quad (4.7)$$

And the uncertainty for each channel is calculated by:

$$error = \sqrt{\frac{n1 \times (n0 - n1)}{n0^3}} \quad (4.8)$$

We then create a new histogram of the efficiency versus transverse momentum. The histogram can be fitted by a dynamic constant function, so it will give us the mean efficiency value in an interesting pT region. For the efficiency versus energy E plot, we follow the same calculations from formulae (4.7) and (4.8), just by using the data from the right plot of figure 4.21 and figure 4.14. The two efficiency plots are shown below.

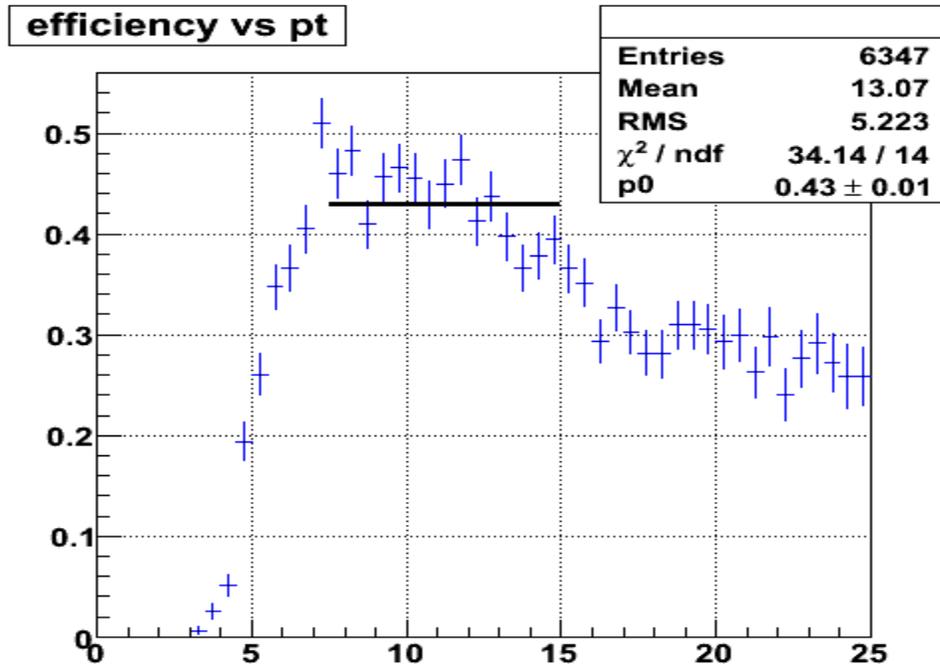


Figure 4.22: The π^0 reconstruction efficiency vs transverse momentum in GeV/c.

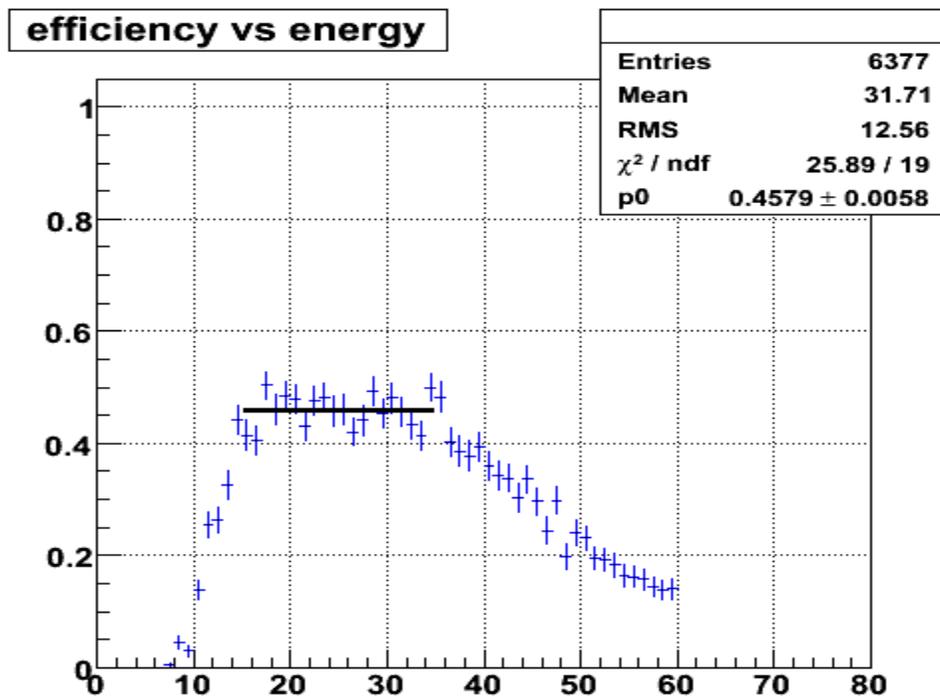


Figure 4.23: The π^0 reconstruction efficiency vs energy in GeV.

Figure 4.22 shows the π^0 reconstruction efficiency versus transverse momentum p_T . We fit the plot from 7 to 15 GeV/c, which is the most interesting region for our inclusive neutral pion analyses. This gives us 43% as the reconstruction efficiency, which represents a substantial improvement compared to the earlier 2005 analyses. Figure 4.23 shows the π^0 reconstruction efficiency versus energy E , and a fit from 15 to 35 GeV gives us the efficiency at 46%. This is consistent with the result in figure 4.22.

4.4.2 Single π^0 Monte-Carlo sample analysis

This sample was generated with only a thousand event with a single thrown π^0 in each event. The vertex for each event was generated at zero to simplify the sample. We used this sample to do simple consistency checks in the reconstruction efficiency of the π^0 finder program.

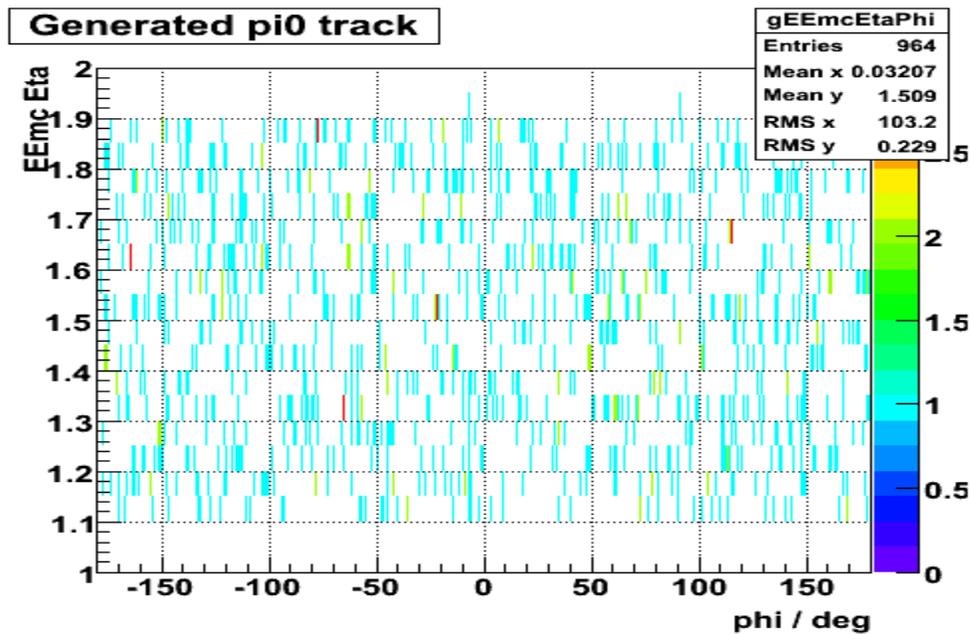


Figure 4.24: Generated geometry for the single π^0 sample with flat distributions in eta and phi.

The generated flat geometry distribution for the single π^0 sample is shown in figure 4.24, which shows this is a small sample. We follow the same logic to calculate the reconstruction efficiency as we did in the di- π^0 sample analysis. In the Pi0-Analysis class, the π^0 candidates are reconstructed with the features illustrated below.

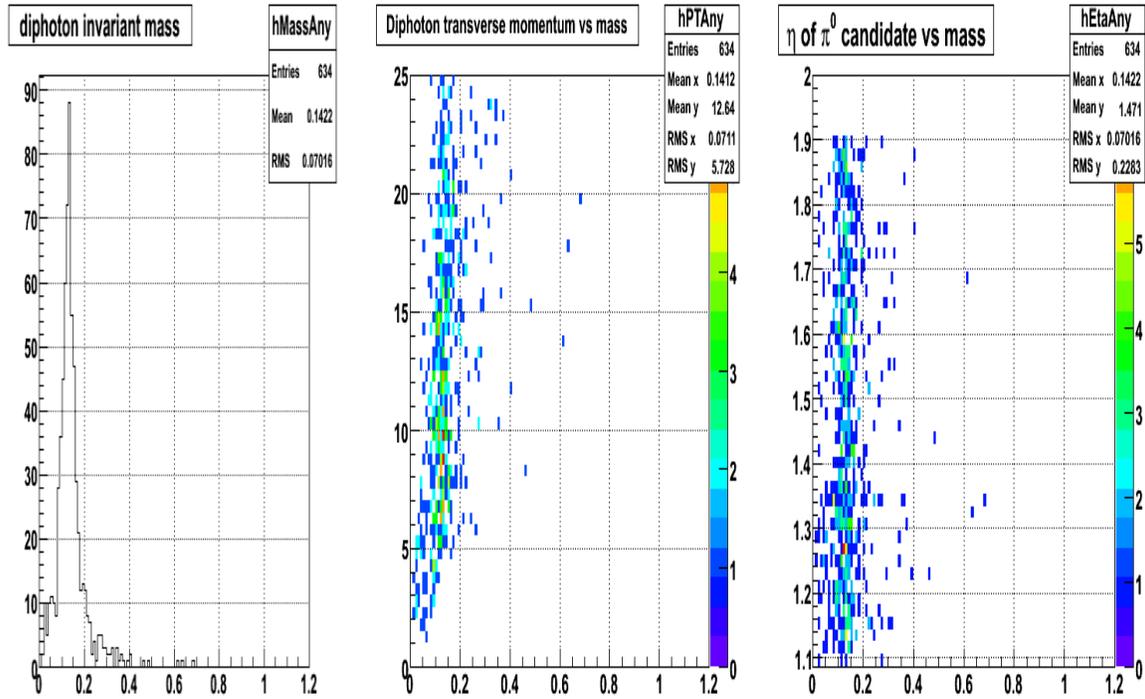


Figure 4.25: Invariant mass spectra peaked at 0.135 GeV from the single π^0 sample of the reconstructed π^0 candidates which satisfy various conditions in the Pi0-Analysis class and are stored into the tree.

Figure 4.25 shows the invariant mass spectra for reconstructed π^0 s from this sample. We can see that these are much cleaner than similar plots from the di- π^0 sample (figure 4.16). This is reasonable because here we only throw a single π^0 for each event into the whole EEMC during simulation generation. Based on these reconstructed π^0 s, we apply the same invariant mass cut from 0.08 to 0.18 GeV as we did for the di- π^0 sample. As

before, the position between the reconstructed π^0 and the generated one is also matched, and we require the distance parameter $\Delta R \leq 0.02$ based on figure 4.26. Those reconstructed π^0 s with distances larger than 0.02 are thrown away for the efficiency study. With this single π^0 sample, the distance between the reconstructed π^0 and matched MC generated π^0 is usually very small because there is only one π^0 particle in each event.

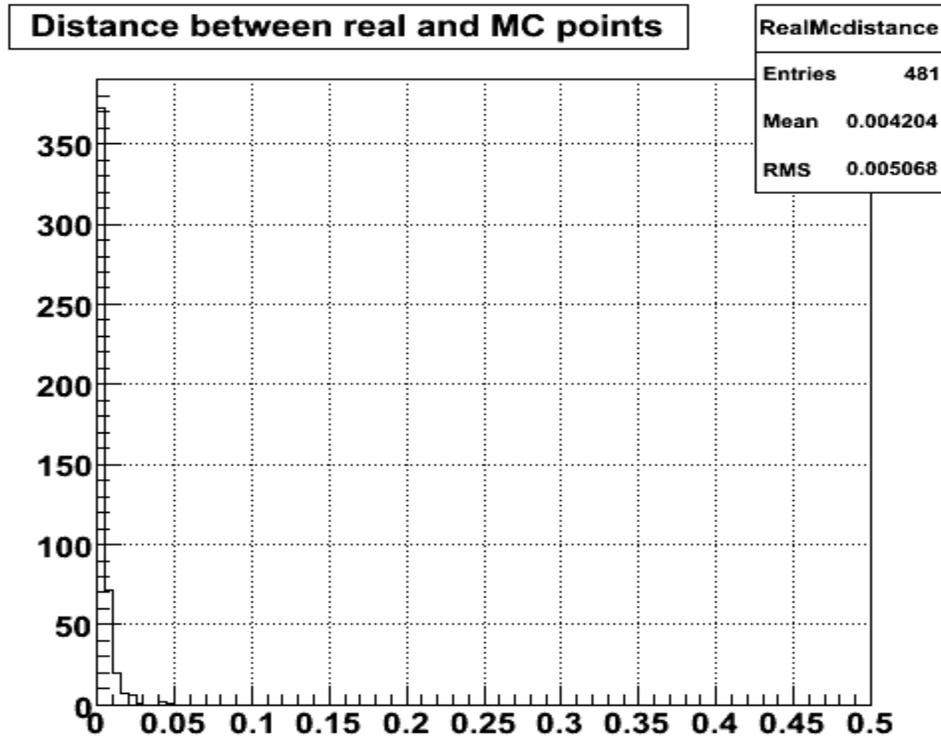


Figure 4.26: Distance between the reconstructed π^0 and the generated π^0 from the single π^0 sample.

Finally, we use the same equations to calculate the reconstruction efficiency for this single π^0 sample as we did in the di- π^0 sample. Two efficiency plots are shown below. The reconstruction efficiencies of 67% for transverse momentum and 68% for energy are both quite a bit larger than the results from the di- π^0 sample. This is not surprising, and

suggests that isolated π^0 s in the real data should be reconstructed quite efficiently by our software.

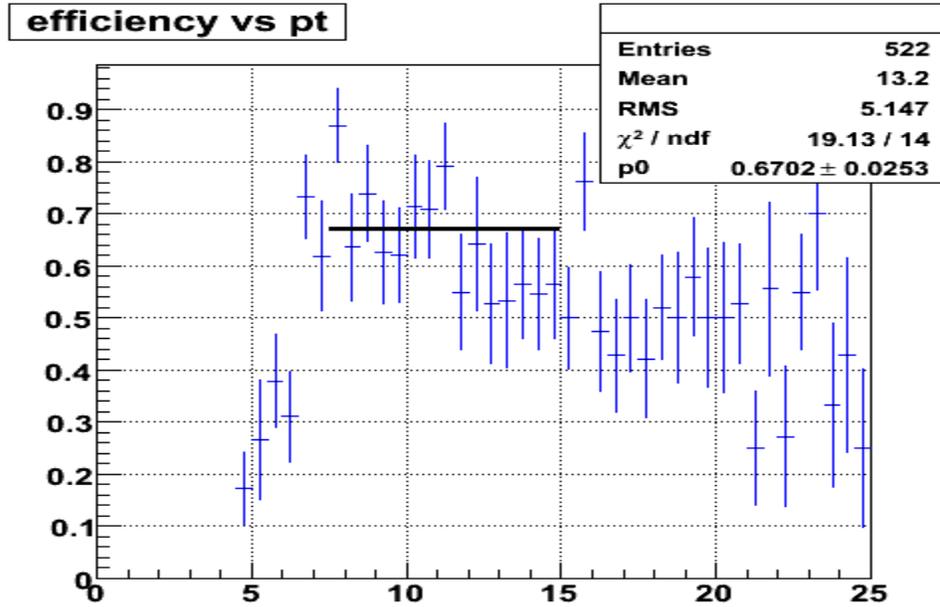


Figure 4.27: The π^0 reconstruction efficiency vs transverse momentum in GeV/c for the single π^0 sample.

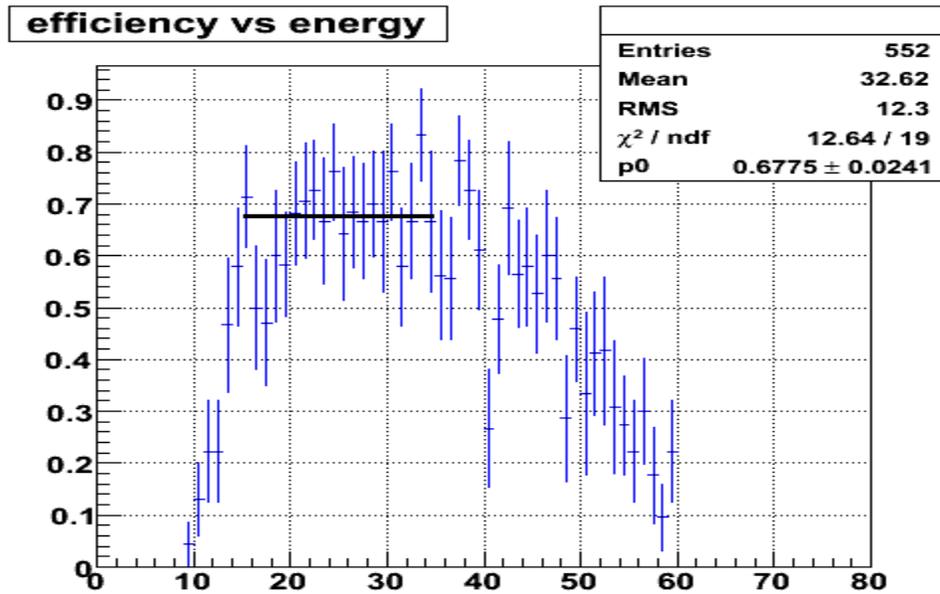


Figure 4.28: The π^0 reconstruction efficiency is vs energy in GeV for the single π^0 sample.

4.4.3 Pythia simulation sample analysis

After the simulation studies with the two Monte-Carlo samples described above, it is important to check the π^0 finder program with a pythia event sample. The STAR database provides a series of pythia event samples for pp collisions at $\sqrt{s}=200$ GeV, and we chose the one with partonic transverse momentum from 25 to 35 GeV/c. This sample has 200,000 events in total, which is likely to give us similar precision in our reconstruction efficiency study. According to the GEANT record, the z vertex distribution for this sample (after a generated π^0 is identified) is spread out at the STAR detector as shown below.

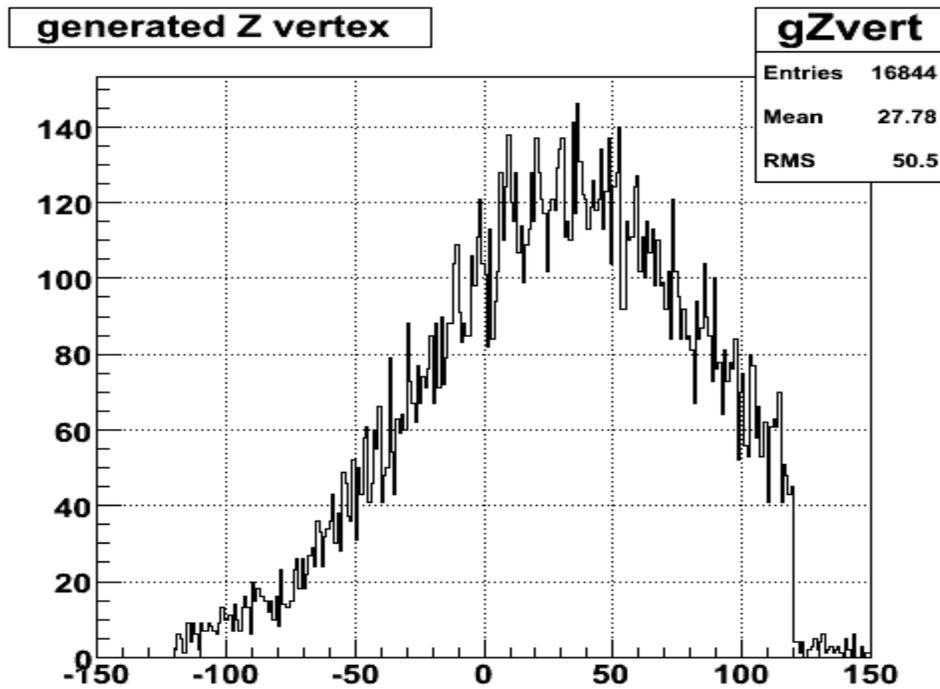


Figure 4.29: Generated π^0 z vertex distribution from the pythia sample.

The z vertex distribution for the generated π^0 s of this pythia sample is right-skewed and spread-out. This is a good feature because we found the vertex dependent π^0

reconstruction in the Mix-Maker class and can treat both the simulation and real data analyses well according to the z vertex analysis in the pythia sample. And the detector η development also benefits from this. The EEMC geometry of π^0 s generated from this pythia sample is shown in the following figure 4.30.

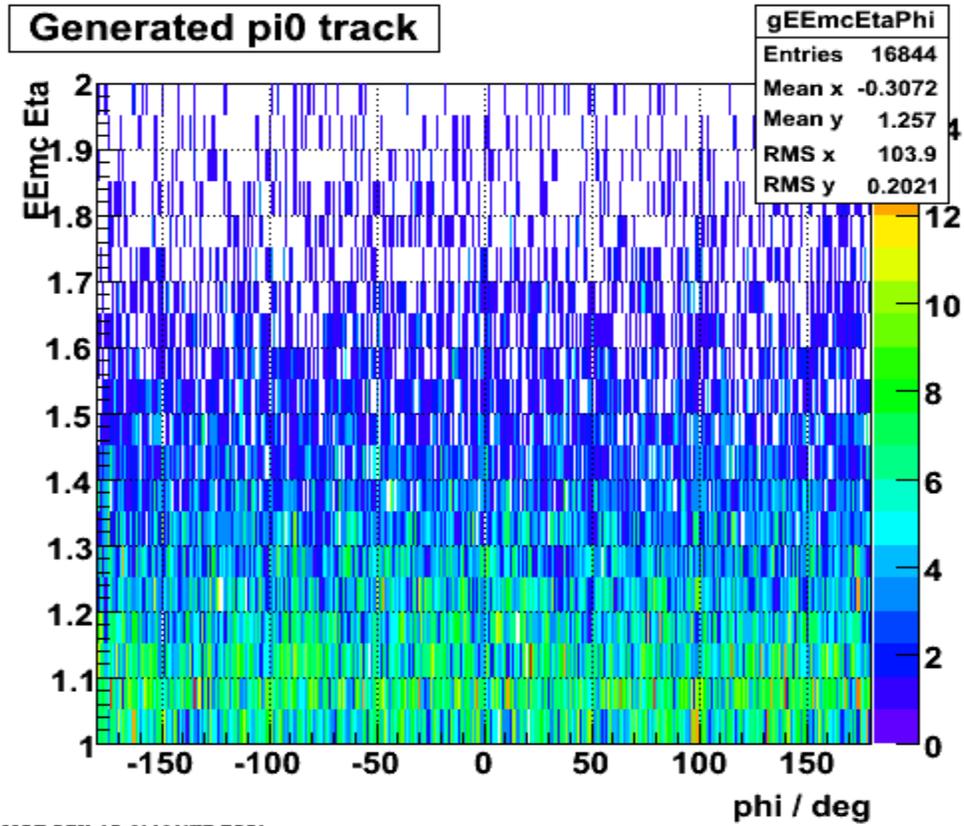


Figure 4.30: Generated π^0 geometry for the pythia event sample.

We basically follow the same logic described in the di- π^0 sample analysis to reconstruct the π^0 candidates and calculate the reconstruction efficiency for this pythia event sample. After the fundamental π^0 reconstruction in the Pi0-Analysis class as described in chapter 3, we apply the invariant mass cut from 0.08 to 0.18 GeV to our reconstructed π^0 s. And then a position match between the reconstructed π^0 and a

generated one is analyzed and applied. But this time we use a different distance cut at $\Delta R=0.04$ because we observe a wider separation as shown in figure 4.31.

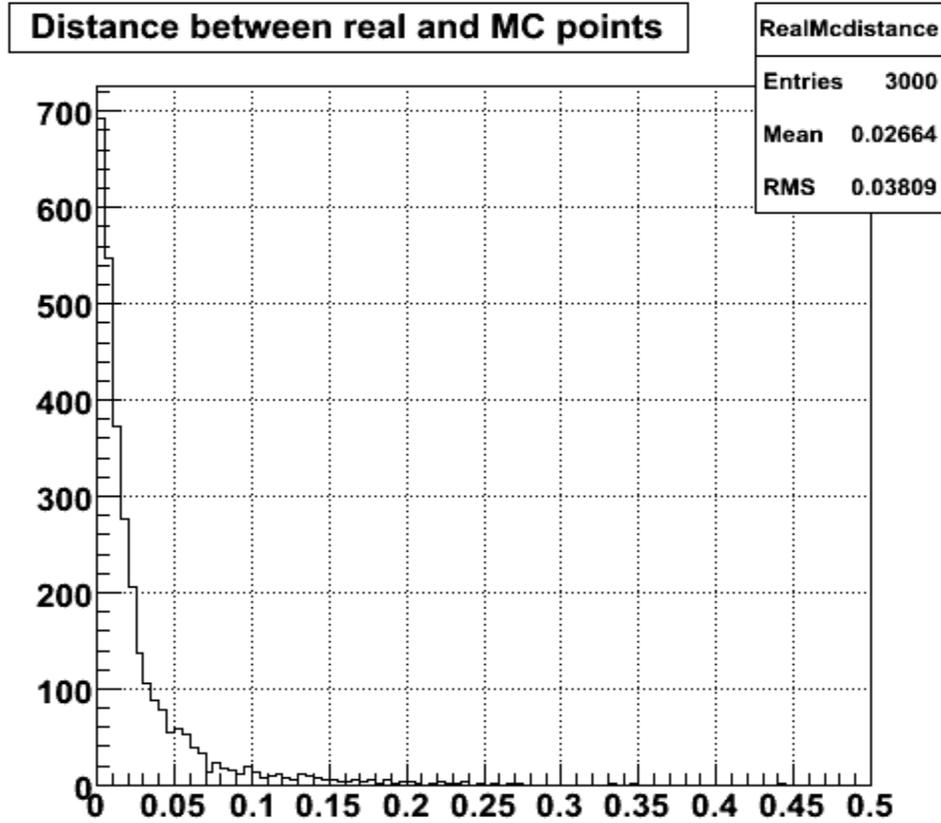


Figure 4.31: Distance between the reconstructed π^0 and the closest generated π^0 from the pythia event sample.

It is reasonable that the distance distribution between the reconstructed π^0 and the closest generated π^0 for the pythia event sample is a little wider than the two we showed for the single and di- π^0 samples, because the pythia event sample simulates more complicated collisions in the STAR detector instead of just throwing individual particles into the detector. Nevertheless, we use the same method to calculate reconstruction efficiencies along transverse momentum and energy for this sample. Our two efficiency plots are shown below.

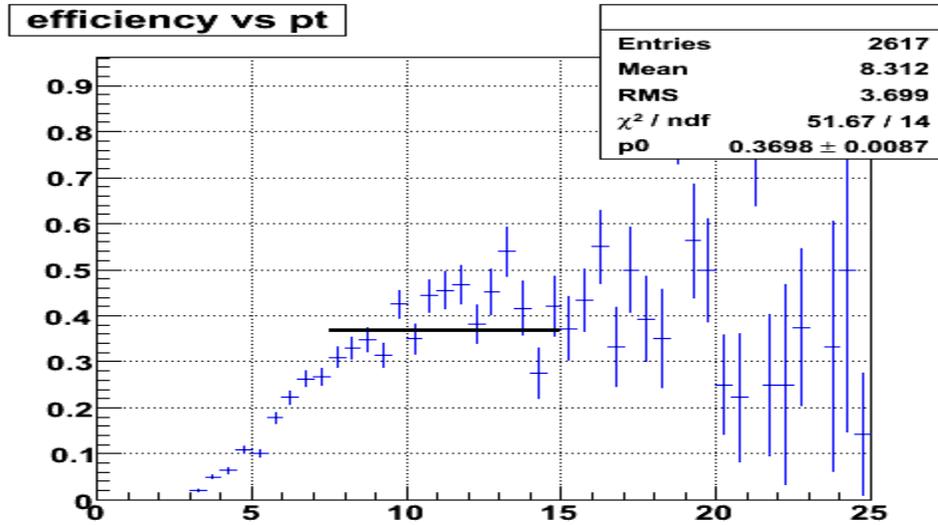


Figure 4.32: The π^0 reconstruction efficiency at vs transverse momentum in GeV/c for pythia events with $25 < \text{partonic } p_T < 35$ GeV/c.

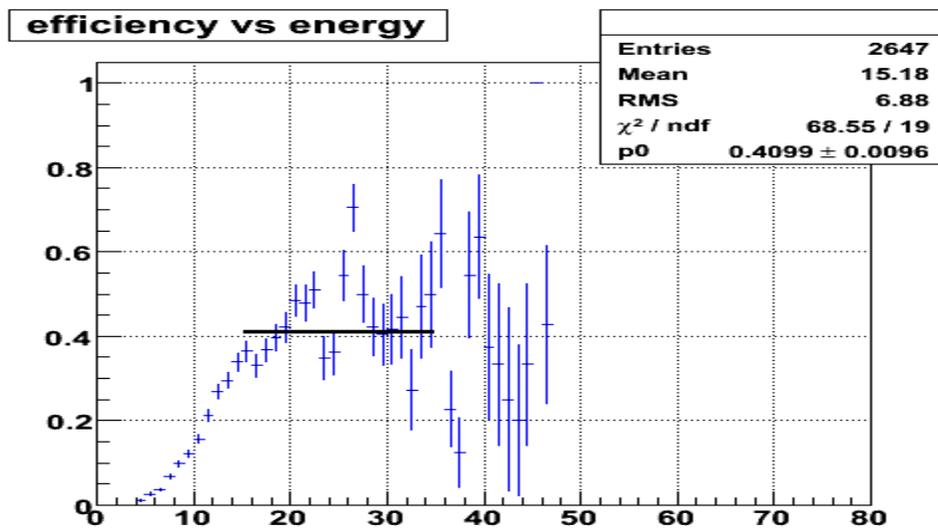


Figure 4.33: The π^0 reconstruction efficiency vs energy in GeV for pythia events with $25 < \text{partonic } p_T < 35$ GeV/c.

We achieve reconstruction efficiencies of $\sim 37\%$ along p_T and $\sim 41\%$ along energy distributions for this pythia event sample. Note in particular that though efficiencies fall off rapidly as p_T and E decrease, they remain quite constant even as p_T and E increase above the fit region.

4.4.4 Reconstruction efficiency summary

For our inclusive neutral pion study, reconstruction of neutral pions in the EEMC detector has been a very important issue. The reconstruction efficiency has been studied and systematically improved over the last three years. The π^0 finder software has been developed with more diagnostic features and detail and new structures, and results in a high reconstruction efficiency compared to the 2-4% reconstruction efficiency of previous work.

We have run the π^0 finder program over three simulation samples to test the π^0 reconstruction and calculate the reconstruction efficiency. Three simulation samples were selected, with two Monte-Carlo samples, (a single thrown π^0 sample and a di- π^0 thrown event sample), and one pythia event sample, so that we can test the reconstruction under different simulation conditions. We have calculated the reconstruction efficiency as a function of both the π^0 transverse momentum and its energy.

The single thrown π^0 sample, with very clean events, gives us the highest π^0 reconstruction efficiency of ~67%. The di- π^0 sample, with both pions generated in a narrow region to make the distribution complex, gives us an excellent π^0 reconstruction efficiency of about 45%. The pythia sample, which most closely simulates collisions in the STAR detector, gives us π^0 reconstruction efficiencies around 36-40% in the kinematic regions of interest, which should be very close to those obtained in real data reconstruction.

Based on these simulation studies, an estimate of π^0 reconstruction efficiency at about 30-40% should be reachable for the 2006 real data analysis. The higher the reconstruction efficiency, the more reliable and less biased the real data analysis will be.

Chapter 5

Data analyses

5.1 Data selection

We selected the 2006 longitudinally polarized proton-proton beam data to calculate the double spin asymmetry of inclusive neutral pion production at the EEMC. The data are stored at HPSS on a run by run basis. Based on trigger and luminosity information, 347 longitudinal pp physics runs from 41 fills were selected for the final analysis, although there are a total of about 400 longitudinal pp physics runs during run6. Some of the longitudinal runs were thrown away because the luminosity information for those runs was questionable. The complete selected run and fill information is provided in appendix A.

For the 2006 pp runs, the full setup of the STAR trigger was utilized. The EMC triggers included information from the Trigger Patches and a total EMC energy threshold for the first time during run6. For each run used in this work, we selected the EEMC-HTTP-MB-L2gamma trigger for our neutral pion analysis. This special trigger is a combination of several triggers in STAR, and is well designed to accept high-pT photons that interest in the EEMC. “HTTP” denotes High Tower and Trigger Patch. High Tower means high energy deposition in a single tower of a trigger patch above a set threshold; Trigger Patch means the sum of energies in a cluster of 6, 8 or 10 adjacent towers also exceeds a certain threshold. For the data used in this analysis, the threshold value set for

HT was 3.5 GeV, and for TP was 4.5 GeV. “MB” denotes minimum bias, which means the STAR beam-beam counters fired for this event. The L2 gamma algorithm is implemented at a higher trigger level to find high tower ET event with L2 thresholds at 3.7, 5.2 for longitudinal running.

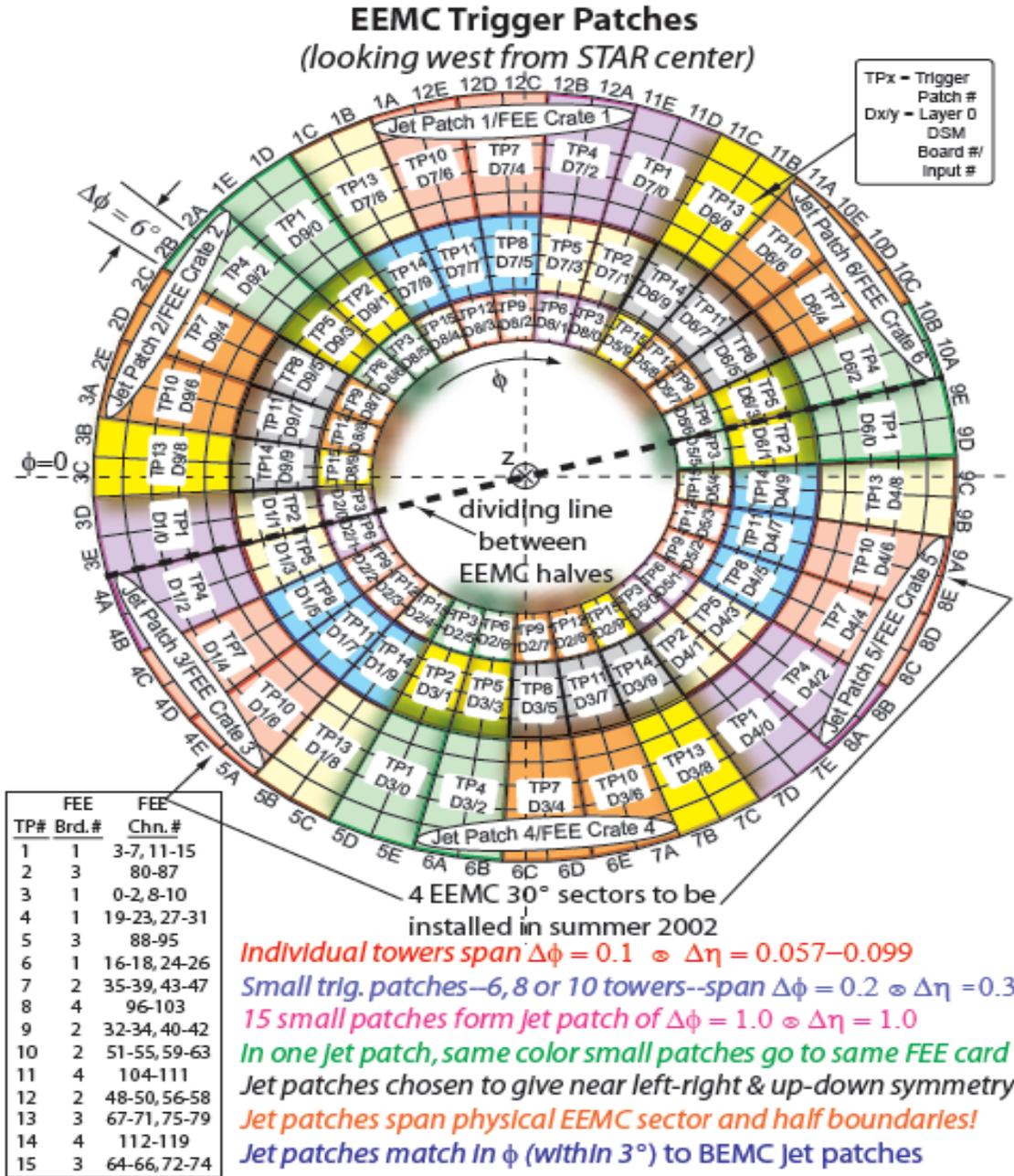


Figure 5.1: The 2002 layout of EEMC trigger patches with detailed labeling scheme and DSM assignments [96].

When we analyze the 2006 data to reconstruct π^0 s, the trigger information is checked in the program. The EEMC-HTTP-MB-L2gamma is the main analysis trigger we used during the study, though the Min-Bias trigger data is also analyzed for consistency. Values are assigned to these two triggers for convenience in the software. The EEMC-HTTP-MB-L2gamma trigger has two ID values, with 137641 denoting runs starting from day 136, and with 6 denoting runs before day 136. The MB trigger has only one ID 117001, during the whole 2006 run. The detailed layout of EEMC trigger patches are shown in figure 5.1.

There are two primary modes of running the π^0 finder program during real data analysis. For small data analysis tasks, we can retrieve specific MuDst files from HPSS to local disks and run the code interactively, or send them to the computing farm by bsub or qsub depending on which server you are on, RCF or PDSF. To retrieve the MuDst files, we can use `get_file_list.pl` to grab information to *.file on run-by-run basis, then use `hpss_user.pl -f *.file` to retrieve the MuDst files and store them in a convenient place. This method was widely used in the early stages of analysis to optimize the running of the code. For large volume data analyses, such as generating the final 347 runs, RCF provides a scheduler so that we can set up the scheduler and submit jobs into the computing farm. These jobs will be running automatically without retrieving and saving the MuDst files to local disk. More technical details about the scheduler will be discussed later in this Chapter.

5.2 π^0 finder software setup

We run through the selected 347 runs of 2006 longitudinal pp data with the π^0 finder program to generate a series of π^0 trees. The π^0 trees are stored in associated root files. A new macro was developed to read these generated π^0 trees and normalize the spin-dependent π^0 yields according to the relative luminosities of the proton beams in the various spin states. We then calculate the double spin asymmetry from the normalized spin-dependent π^0 yields after background subtraction. In this section, we will discuss details of setting up the π^0 finder software to generate the π^0 trees for real data.

Cuts and threshold values have been set in the π^0 finder software for the simulation studies described in Chapter 4. We keep some of these values the same for real data analysis as for simulation studies, and change some other settings a little with the same π^0 finder software framework to maximize the inclusive π^0 yield for the study. The sigma (width) threshold in the A2E-Maker is still kept at 3 for all layers of ADC values in the towers, that is, the signal must be at least 3σ above the pedestal value inserted in the database. The tower seed threshold is set to 0.8 GeV. The SMD floor setting parameter mSeedFloor is kept at 1, as in simulation studies. To form a SMD cluster, we still require three SMD strips to be counted on each side of a seed strip. The above settings are the same as for the simulation studies, but we also changed some parameters for the real data analysis. Real data analyses give us some different features in the reconstructed π^0 s. Based on the fluctuation study described in the next subsection, a new floor setting in the SMD plane and a point transverse energy ET cut were implemented in the code. Based on an edge effect study for the real data, a bivariate Gaussian distribution

tower ET cut peaked at 3 GeV was applied. And finally, the SMD seed threshold was eventually set to 3 MeV to both planes for the final data analysis, instead of the lower value of 1.5 MeV that was used in the simulation studies.

5.2.1 SMD fluctuation studies

When we started to compare results from our simulation studies to those from real data, the first problem we encountered was much larger variations in the real SMD energy response. We could easily “hide” some of these effects by increasing our relative cuts and threshold values in the software. But this approach would damage our final inclusive neutral pion results by greatly reducing the π^0 yield, and possibly introducing large systematic biases in our π^0 reconstruction. To avoid these problems, we first had to examine the SMD response in more detail. The three kinds of typical SMD fluctuations in energy response from both planes are shown below.

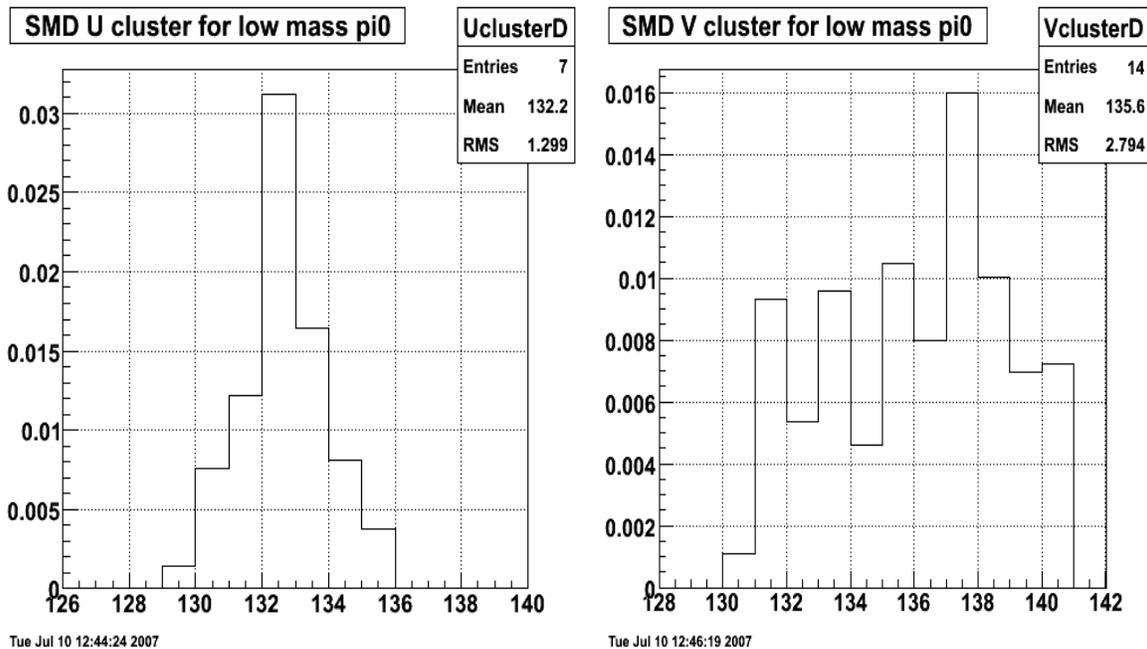


Figure 5.2: One normal cluster is reconstructed from the U plane, while two abnormal clusters were found in the V plane.

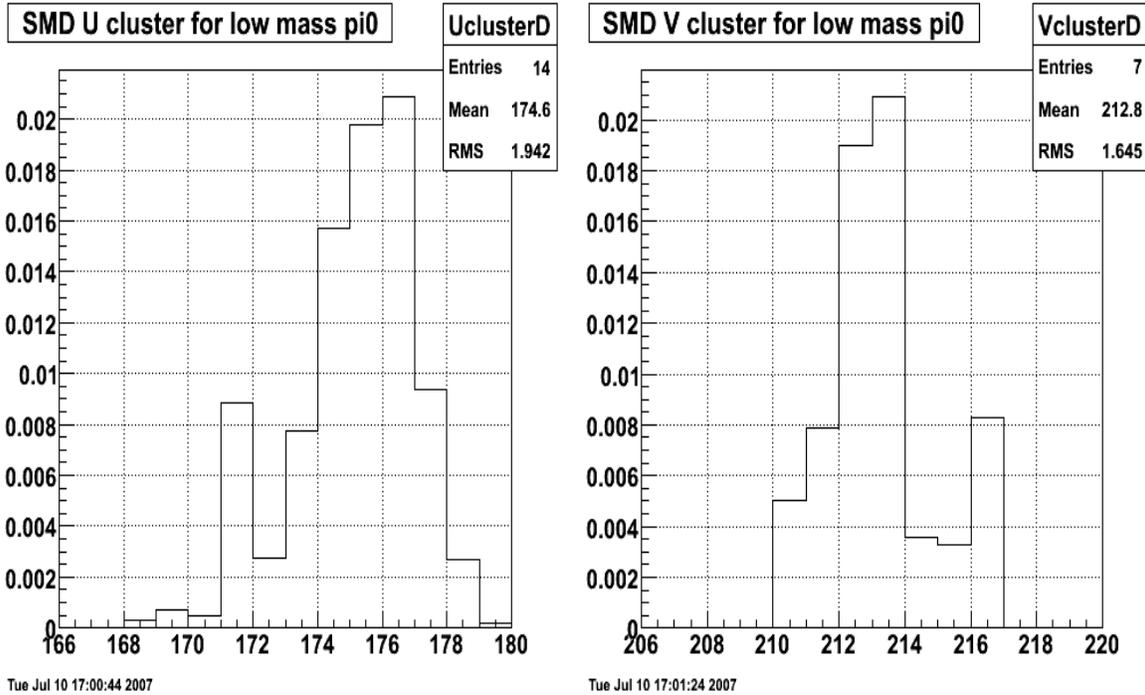


Figure 5.3: Two abnormal clusters are reconstructed from the U plane, with one normal cluster from V plane, is another kind of typical fluctuation from the SMD response.

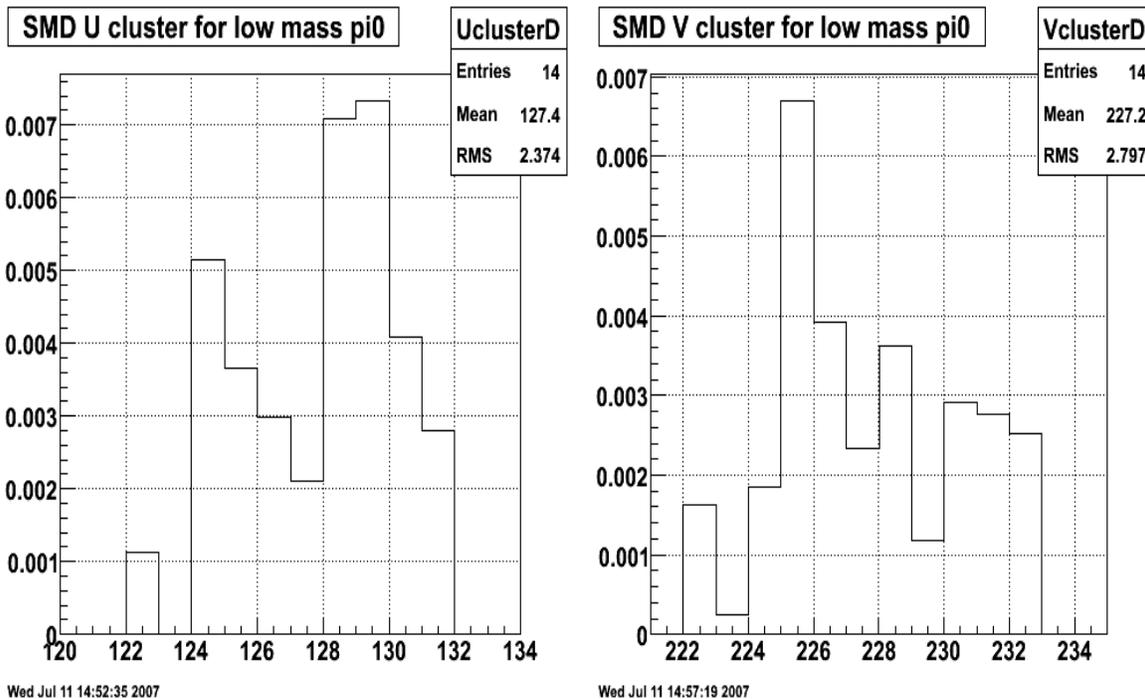


Figure 5.4: Two abnormal clusters are reconstructed from both SMD planes.

We show three examples of typical fluctuation features from the SMD strip energy responses in figures 5.2, 5.3 and 5.4. These profiles can be easily technically generated by retrieving the information from most SMD responses of the real data. We found that our software often reconstructs low invariant mass pion candidates due to these fluctuations, because of the resulting small opening angle and particle energies. Figure 5.2 shows a case with a normal cluster in the U plane, while two “abnormal” clusters were found in the V plane, resulting in a reconstructed low mass π^0 . Figure 5.3 shows a similar situation with U and V reversed. These two events have the same problem: large energy fluctuations, leading to multiple reconstructed SMD clusters in only one plane. Sometimes, events are even more ambiguous because both planes exhibit severe fluctuations and reconstruct multiple clusters, such as in figure 5.4. The proper reconstruction of SMD clusters from both planes is a fundamental part of our π^0 reconstruction procedure. The fluctuations in SMD strip energy response will often result in finding one or more “abnormal” SMD clusters close to normal ones. When we use the splitting algorithm in the Point-Maker to reconstruct points from SMD clusters, these extra clusters will produce many low energy point candidates. These point candidates produced from fluctuations are typically very close to normal points, only a few strips away. So when we match point pairs to reconstruct π^0 candidates, we will generate numerous combinations of low mass, due to their small opening angles, quantities and low energies, which are directly used in the calculation of the invariant mass.

A prominent feature of the SMD points produced by these fluctuations is their low energy distribution. Therefore, we considered adding one more new cut in the code, based on the energy distribution from these points. We picked a specific longitudinal run

(7078044) and examined the transverse energy ET distribution from all reconstructed point candidates. The distribution is shown in figure 5.5. Based on this distribution, and keeping in mind that the most important region of transverse momentum for our inclusive neutral pion analysis is from 5 GeV/c to 15 GeV/c as discussed in Chapter 1, we decided to apply a new point ET cut at 1.5 GeV. This new cut helps a lot in reducing the low mass pair production, although it does not eliminate the whole low mass peak. Unfortunately, this cut also suppresses the reconstruction of (real) π^0 s with large energy sharing $Z_{\gamma\gamma}$, but it does not affect the π^0 relative yield as a function of transverse momentum pT above 5 GeV/c or pseudo-rapidity η .

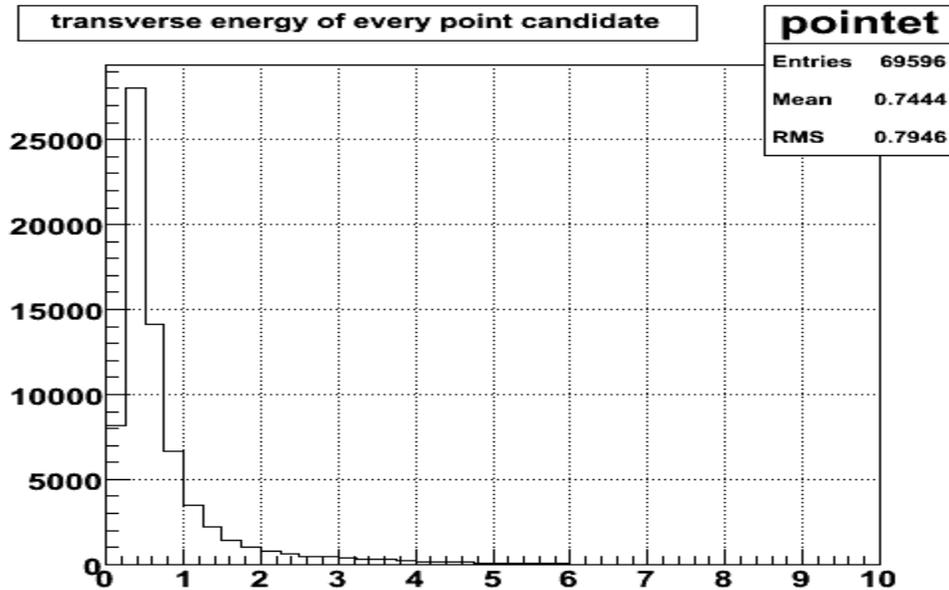


Figure 5.5: The transverse energy ET distribution from all point candidates for run 7078044. The SMD fluctuations produce a large number of low energy points.

To further reduce the fluctuation effects, we also tried several tests separately: (1) an averaging algorithm in the Cluster-Maker, which calculated a new energy value for each strip by averaging its two adjacent strips with the equation $E_{i_new} = (0.25 \times E_{i-1}) + (0.5 \times E_i) + (0.25 \times E_{i+1})$; (2) requiring either of the two strips around a seed strip in

the SMD planes to pass the general floor values setting; (3) and requiring both of the strips around a seed strip in the SMD planes to pass the general floor values setting discussed in Chapter 3. These efforts did not appear to help much in suppressing the low mass peak, as we show below.

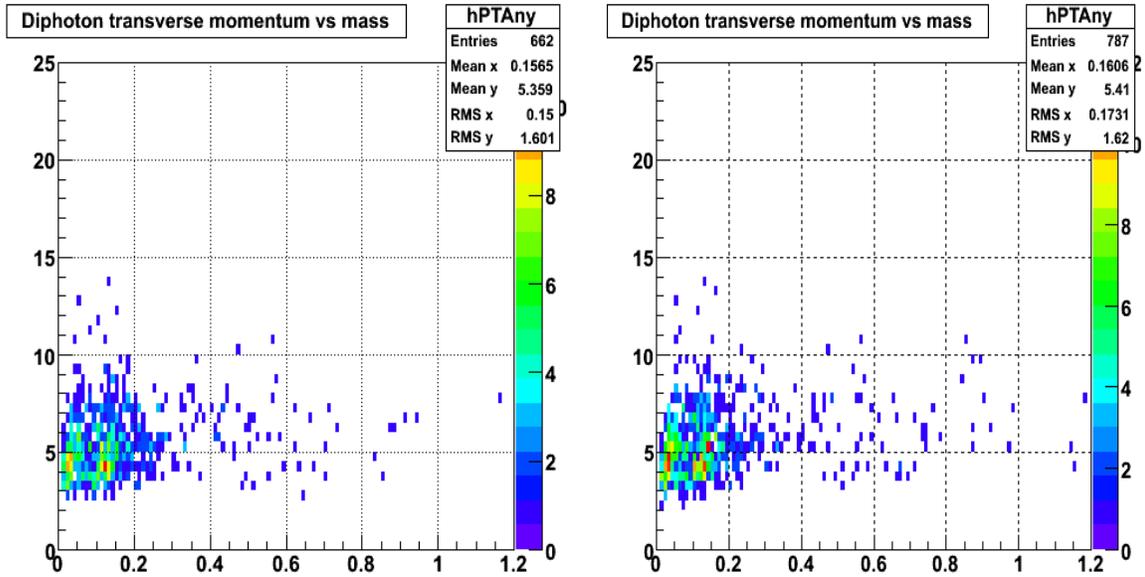


Figure 5.6: π^0 pT versus invariant mass from run 7078044. The left plot is produced from the original algorithm, and the right plot is from the averaging algorithm.

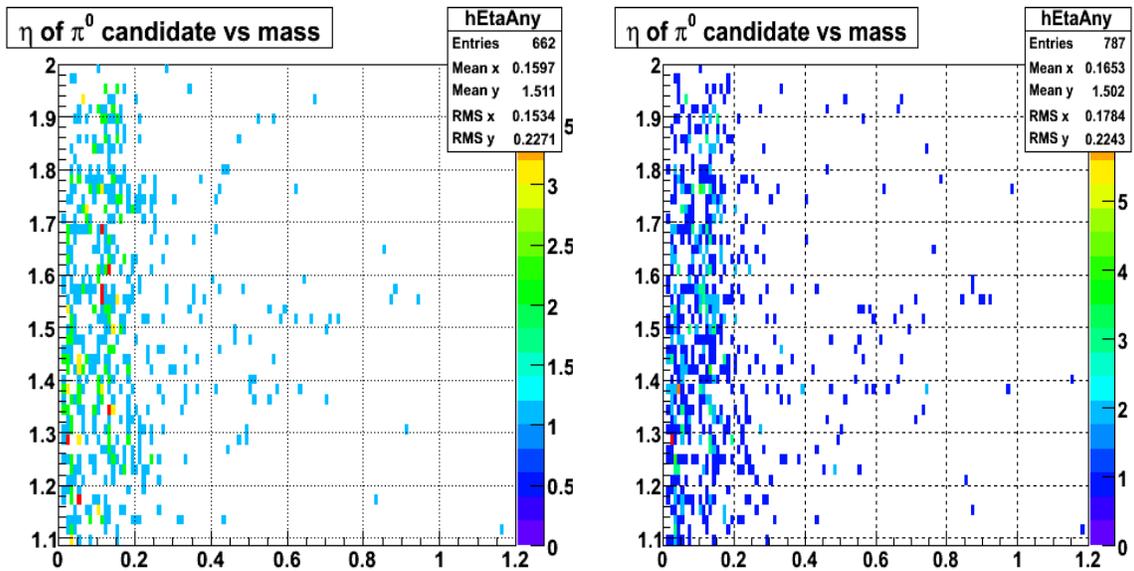


Figure 5.7: π^0 pseudo-rapidity versus invariant mass from run 7078044. The left plot is produced from the original algorithm, and the right plot is from the averaging algorithm.

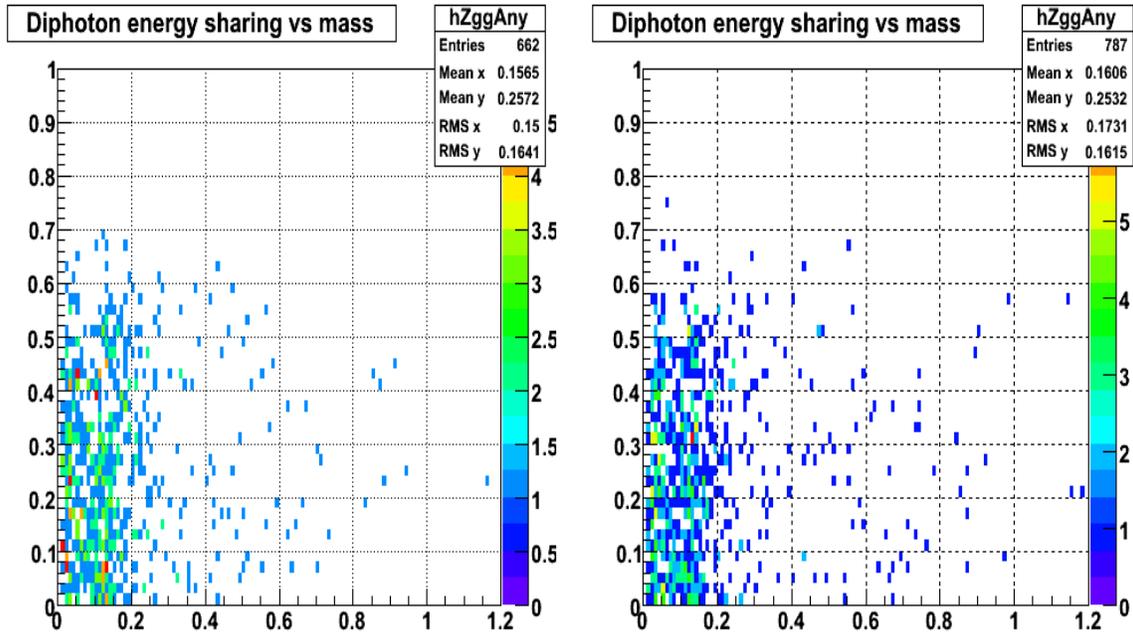


Figure 5.8: π^0 energy-sharing versus invariant mass from run 7078044. The left plot is produced from the original algorithm, and the right plot is from the averaging algorithm.

Comparison of some of the production features between the original and averaging algorithms in the Cluster-Maker for different observables are shown in figures 5.6 to 5.8. The results of last two tests, requiring strips around a seed strip to pass the floor value, are not shown here, since they did not help much in reducing more of the low mass peak. Based on these studies, we decided to only apply the new point ET cut to the original code framework used in the simulation work. But results from the last two tests described above did give us some new ideas on how to further probe the fluctuations from the SMD response, because we were hoping these tests would help suppress fluctuations by requiring the adjacent strips to fire. Each π^0 is reconstructed from two pairs of SMD clusters in the U and V planes. We denote the separation between the two seed strips of the two SMD clusters from a given plane as the ‘seed distance’. We plotted the ‘seed distance’ distributions from both SMD planes, and the spatial distributions between the

point pair, to see if we can find any clues on fluctuations. We analyzed another long run, 7136033 to get more statistics.

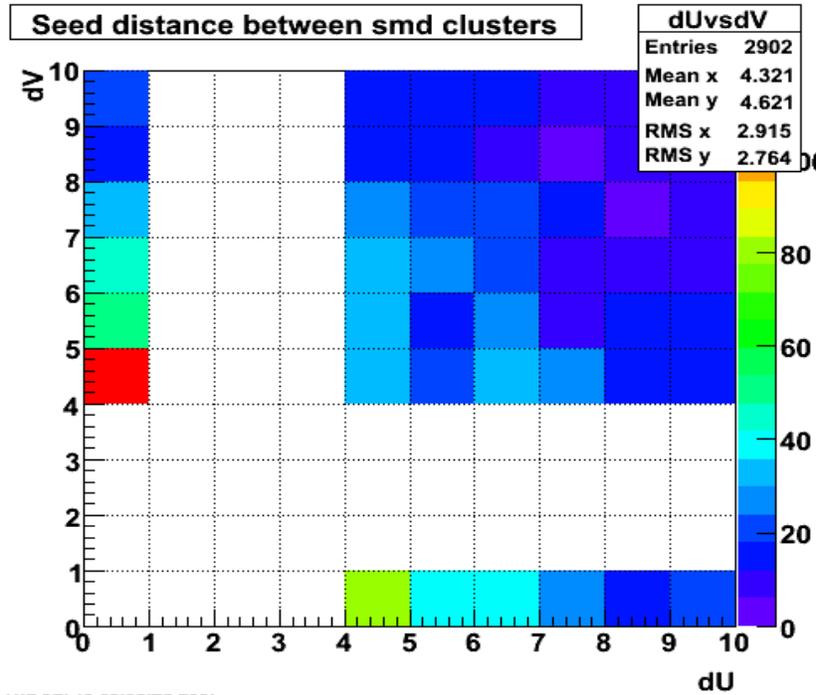


Figure 5.9: The separation distribution between two seed strips from the same plane.

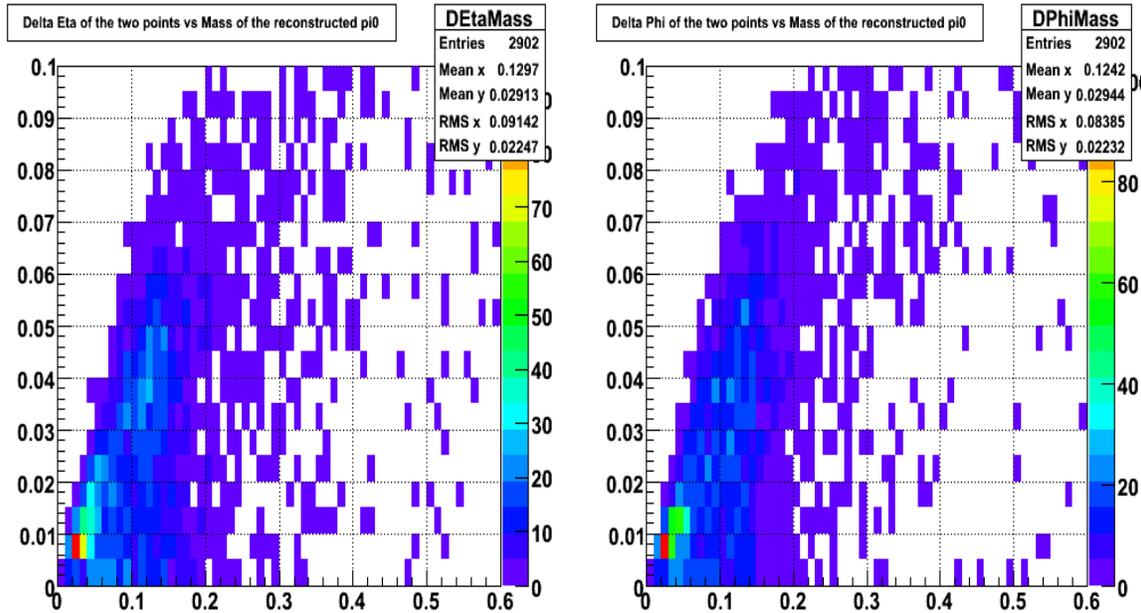


Figure 5.10: Spatial distributions between the pairs of points in π^0 candidates. We show $\Delta\eta$ or $\Delta\phi$ (y-axis) vs mass (x-axis).

The plots shown in figures 5.9 and 5.10 suggest two conclusions: first, the low mass peak generated from SMD fluctuations arises mostly from short spatial separations between the point pair; and second, the SMD fluctuations are geometry-dependent. The first conclusion is not surprising. The second conclusion led us to think more about the geometries design of the SMD planes. As we discussed in Chapter 2, the SMD is designed with a special sector dependence. Sectors 1, 4, 7 and 10 share the same layer structure of V plane, Spacer, U plane; sectors 2, 5, 8 and 11 all have the structure of U, V, Spacer; and sectors 3, 6, 9 and 12 have the structure of Spacer, U, V. Details are summarized in table 2.3 and figure 2.9. To see if the fluctuation dependence is consistent with the design of SMD structure, we looked more closely at the seed distance distributions for the three different SMD structures, for the π^0 candidates with $mass \leq 0.06$ GeV. Our results are shown below.

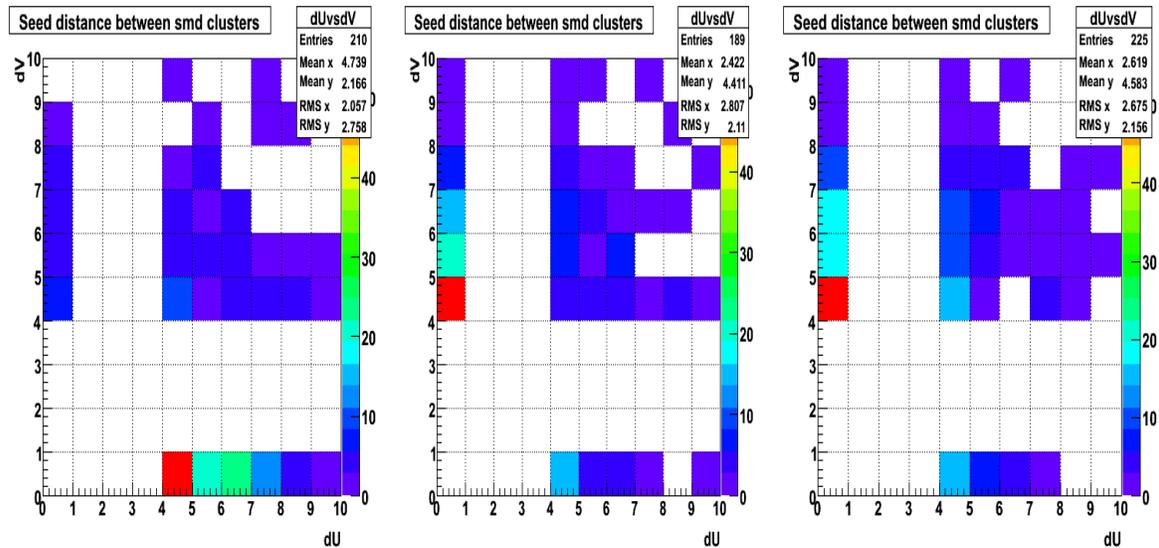


Figure 5.11: The left plot shows the seed distribution from sectors 1, 4, 7 and 10 with the V, S, U structure; the middle plot is from sectors 2, 5, 8 and 11 with the U, V, S structure; the right plot is from sectors 3, 6, 9 and 12 with the S, U, V structure.

A raw calculation was carried out on the seed distance distributions shown in figure 5.11. We summed over the content along the $dV=0$ and $dU=0$ edges for the above three

plots, and obtained the following results going from the left plot to the right plot: the $dV=0$ row summed to 118, 24 and 28; the $dU=0$ column summed to 20, 97 and 104. These numbers demonstrate that when the V plane is in front of the U plane, we reconstruct many low mass π^0 s because of the energy fluctuations in the U plane; but when the U plane is in front of the V plane, we reconstruct low mass π^0 s because of fluctuations in the V plane. If we define the SMD plane closer to the front surface of the EEMC as the first plane, and the plane closer to the back side of the EEMC as the second plane, we can conclude from these results that significant energy fluctuations occur more often in the second plane than in the first plane, which leads to the production of a π^0 low mass peak in the π^0 invariant mass spectrum. This confirms the SMD geometry dependence of these fluctuations. Based on this study, we decided to introduce a new floor shape for the second SMD plane, as described in Chapter 3. We show again the new setting from figure 3.1.

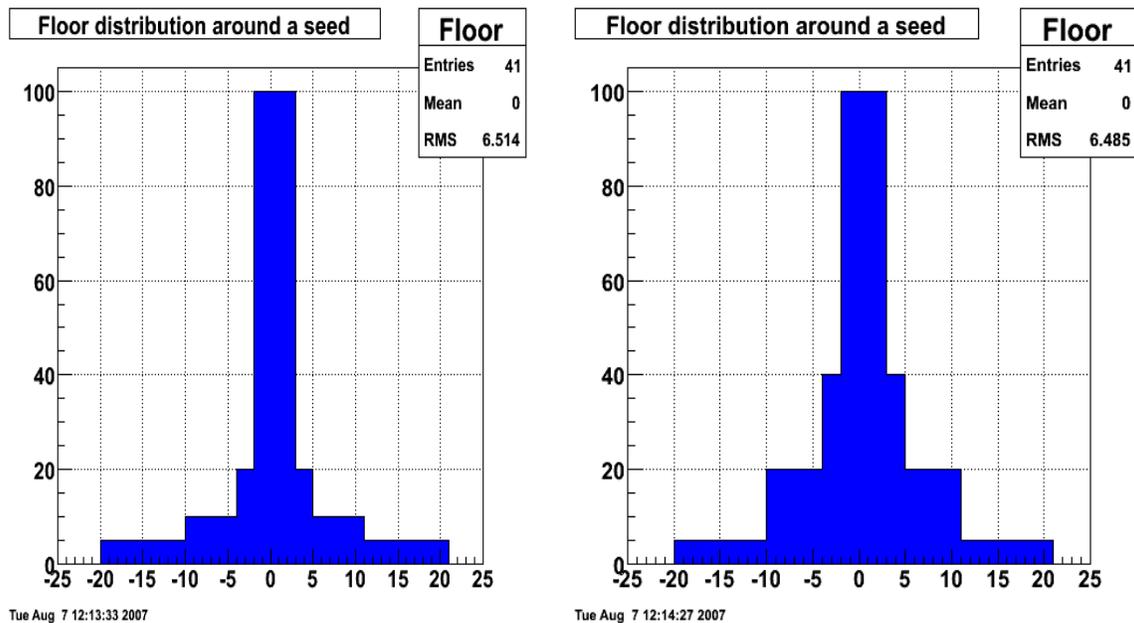


Figure 3.1: The floor settings of the SMD to find seed strips for the front plane (left) and the second plane (right).

Because there are different fluctuation features for the two SMD planes, we decided to apply different floor shapes to the two SMD planes of all sectors, in order to avoid sector-dependent low mass π^0 production. The floor shapes were developed to restrict the SMD strip seed selection, as described in Chapter 3. The left plot in figure 3.1 is applied to the first plane as defined above, and the right plot is applied to the second plane for real data analyses. This means we apply a more stringent requirement on the SMD seed strip selection for the plane which gives more fluctuations.

In summary, we reduced the effects from fluctuations of the SMD energy response for our inclusive neutral pion reconstruction by adding a new point transverse energy ET cut of 1.5 GeV before π^0 reconstruction, and then applying a new floor shape to the second SMD plane. These techniques reduced the low mass peak with little loss in the real π^0 yield. But we emphasize that the true cause of the SMD fluctuations is still not understood. This might be a severe problem in the future photon analyses.

5.2.2 Tower boundary effect

With the new point ET cut and floor shape in the SMD planes added to our analysis software, we ran through more statistics of real data to reconstruct π^0 s with the π^0 finder program. Twenty runs (7136022, 7136033, 7136034, 7137036, 7138001, 7138010, 7138032, 7140046, 7143012, 7144014, 7145018, 7145024, 7146020, 7146077, 7147052, 7148027, 7149005, 7152062, 7153008 and 7155052) from the 2006 longitudinal pp data set were selected to check some important features of the π^0 production. These twenty runs are all long runs, based on the run lengths tabulated in appendix A. We examined the π^0 yield as a function of the tower geometry, as we did in simulation studies, and found

large variations along the η and ϕ directions. As we explain below, we suspected this might be due to our choice of hardware trigger, so we decided to not only analyze events from the twenty runs with our usual EEMC-HTTP-MB-L2gamma trigger, but separately analyze the EEMC-JP1-MB trigger. The structure of the EEMC Jet-Patches is described in figure 5.1. The EEMC-JP1-MB trigger basically requires 8 GeV of summed energy in a 1×1 (in $\eta \times \phi$) area of EEMC towers.

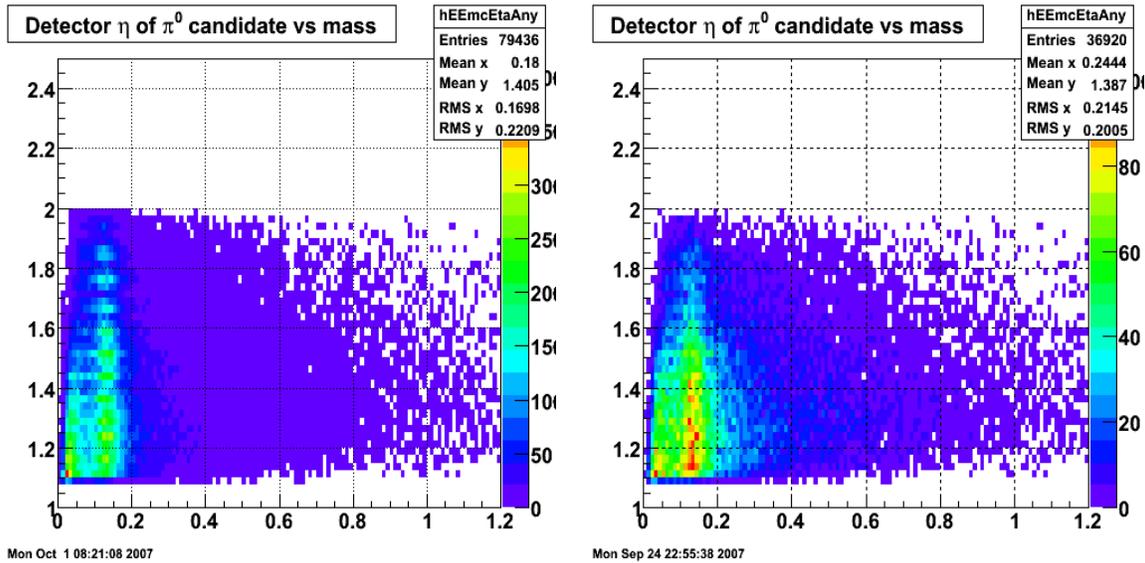


Figure 5.12: The π^0 yield as a function of η vs invariant mass from triggers L2gamma (left) and JP1 (right) for the twenty longitudinal runs.

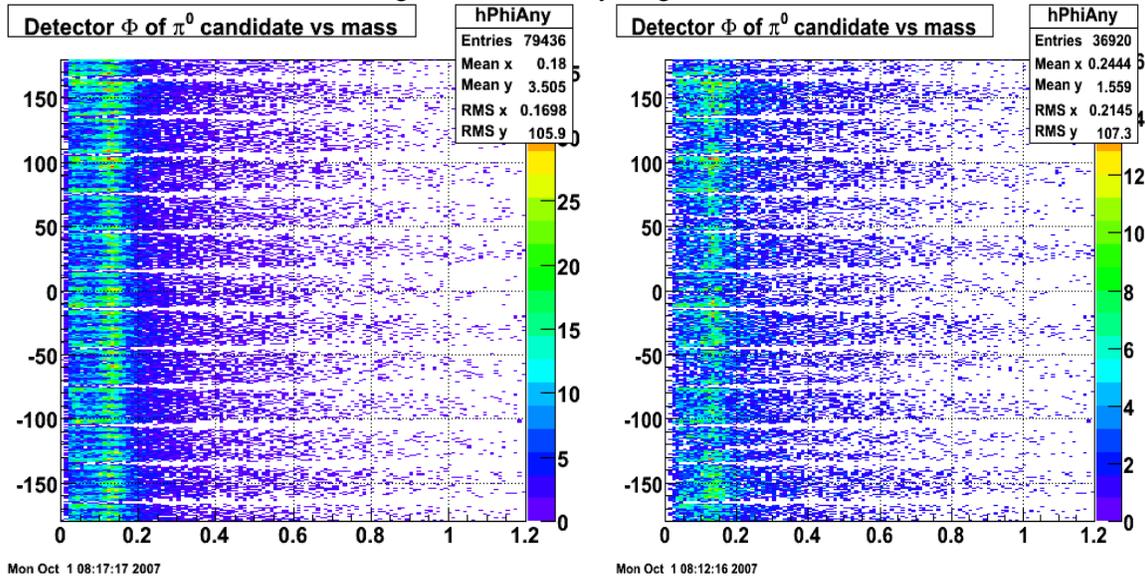


Figure 5.13: The π^0 yield as a function of ϕ vs invariant mass from triggers L2gamma (left) and JP1 (right) for the twenty longitudinal runs.

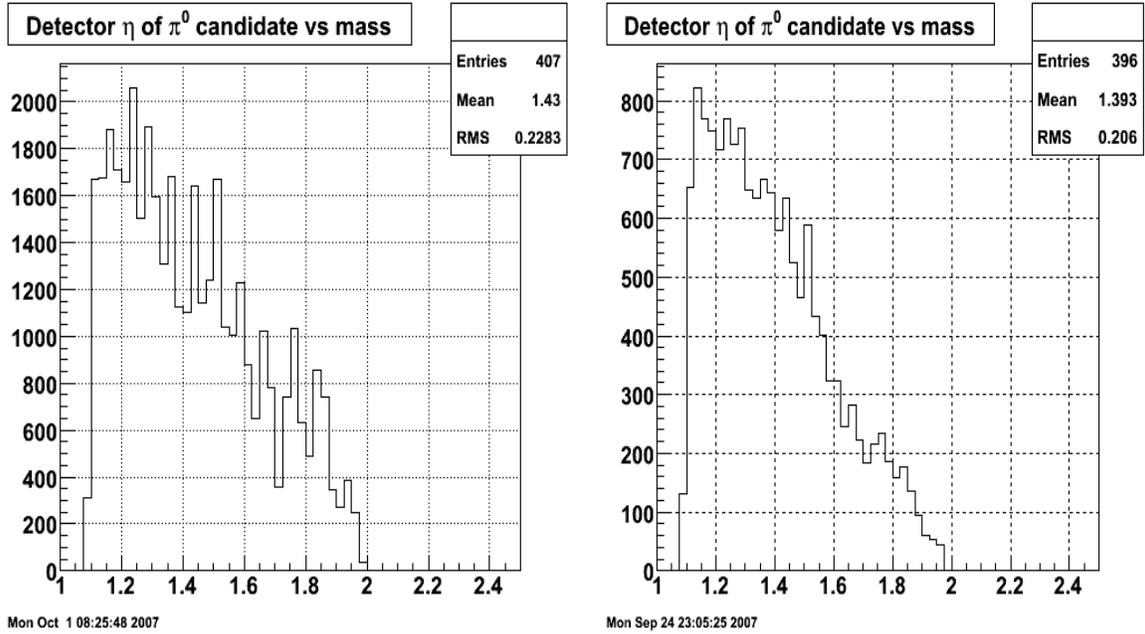


Figure 5.14: The π^0 yield as a function of η with invariant mass cut [0.08, 0.18] GeV for the L2gamma (left) and JP1 (right) triggers.

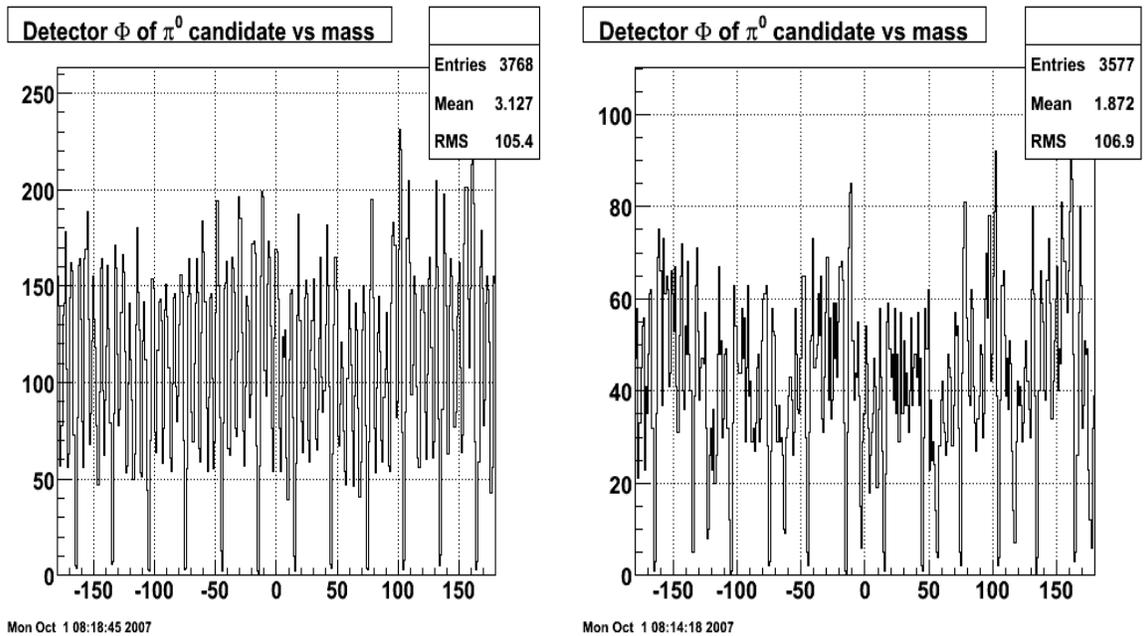


Figure 5.15: The π^0 yield as a function of ϕ with invariant mass cut [0.08, 0.18] GeV for the L2gamma (left) and JP1 (right) triggers.

When we reconstruct neutral pions, we find that more π^0 s are reconstructed in the middle of a tower than at the edge of the tower. This “edge effect” is clearly seen in both the η and ϕ directions from figures 5.12 to 5.15. The edge effect also has a more

complicated tower dependence. If we look at the ϕ plots more closely, we find even fewer π^0 s are reconstructed at the edge between two sectors, because the structure of the EEMC is installed sector by sector. We show this more clearly in figure 5.16 by re-scaling the ϕ distribution from figure 5.16 every 30 degrees, but without the mass cut. This means we show the π^0 yield along ϕ in a “summed sector” instead of twelve sectors.

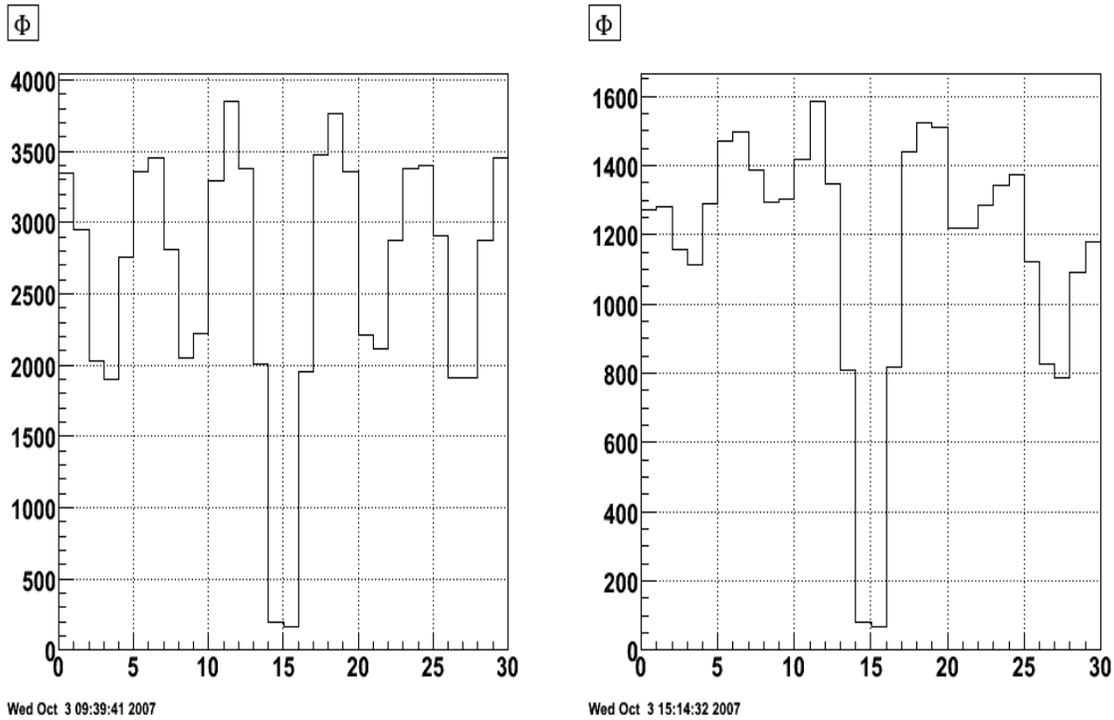


Figure 5.16: The ϕ yield of π^0 s by rescaling the EEMC every 30 degrees. The left plot is for the L2gamma trigger and the right plot is for the JP1 trigger.

We can also see that the η and ϕ yields of π^0 s are smoother from the JP1 trigger than from the L2gamma trigger. This tells us that the edge effects are closely related to the trigger selection. Compared to the EEMC-HTTP-MB-L2gamma trigger, the EEMC-JP1-MB trigger is less sensitive to tower boundaries, because energies are summed over many towers (120), as shown in figure 5.1. But even with the JP1 trigger, some edge effects still exist, which means the algorithm in our π^0 finder software still has a bias in

π^0 position. The π^0 finder program is based on SMD clusters, but a tower transverse energy ET cut of 3.0 GeV was applied to either of the two points in reconstructing a π^0 . To reduce the edge effects, we decided to replace the ET cut at 3 GeV with a bivariate Gaussian cut distribution based on the simulation study presented in Chapter 4. We recall the distribution equation here:

$$Cut = 3.0 \times e^{-0.5 \times \left(\left(\frac{peta - meta}{0.035} \right)^2 + \left(\frac{\varphi'}{2.3} \right)^2 \right)} \quad (5.1)$$

Here *peta* is the η position of a point from the reconstructed π^0 , *meta* is the mean value of the pseudo-rapidity for a specific tower, and φ' is the modified φ position of the point. The new ET cut is set up according to the position of the point. As noted previously, the EEMC is installed symmetrically around the azimuthal angle, with zero degrees located at the center of a tower. So we can easily calculate φ' by taking φ modulo 6 degrees, which is the span of a tower. *Peta* is a variable and *meta* is dependent on the tower etabin. There are a total of 12 etabins along η , which means there are 12 values for *meta*. We summarize *meta* values along the range of *peta* in table 5.1.

Table 5.1: The *meta* values for equation (5.1) according to the eta-bins.

Peta	1.086- 1.146	1.146- 1.205	1.205- 1.268	1.268- 1.334	1.334- 1.403	1.403- 1.476	1.476- 1.552	1.552- 1.633	1.633- 1.718	1.718- 1.807	1.807- 1.901	1.901- 2.0
Meta	1.126	1.174	1.231	1.294	1.36	1.437	1.507	1.585	1.664	1.761	1.845	1.932

We applied this new position-dependent tower ET cut and ran through the twenty longitudinal runs. The π^0 reconstruction yield improves by 9.48%. However, this only smoothed the edge effect a little bit. The edge effect still exists for the real data, and is not well understood from simulation studies. So the new tower ET cut with a bivariate Gaussian distribution was only applied to the π^0 finder software for real data analyses, and not for our simulation studies. We show the bivariate Gaussian distributions used in our analyses for all twelve etabins in figure 5.17.

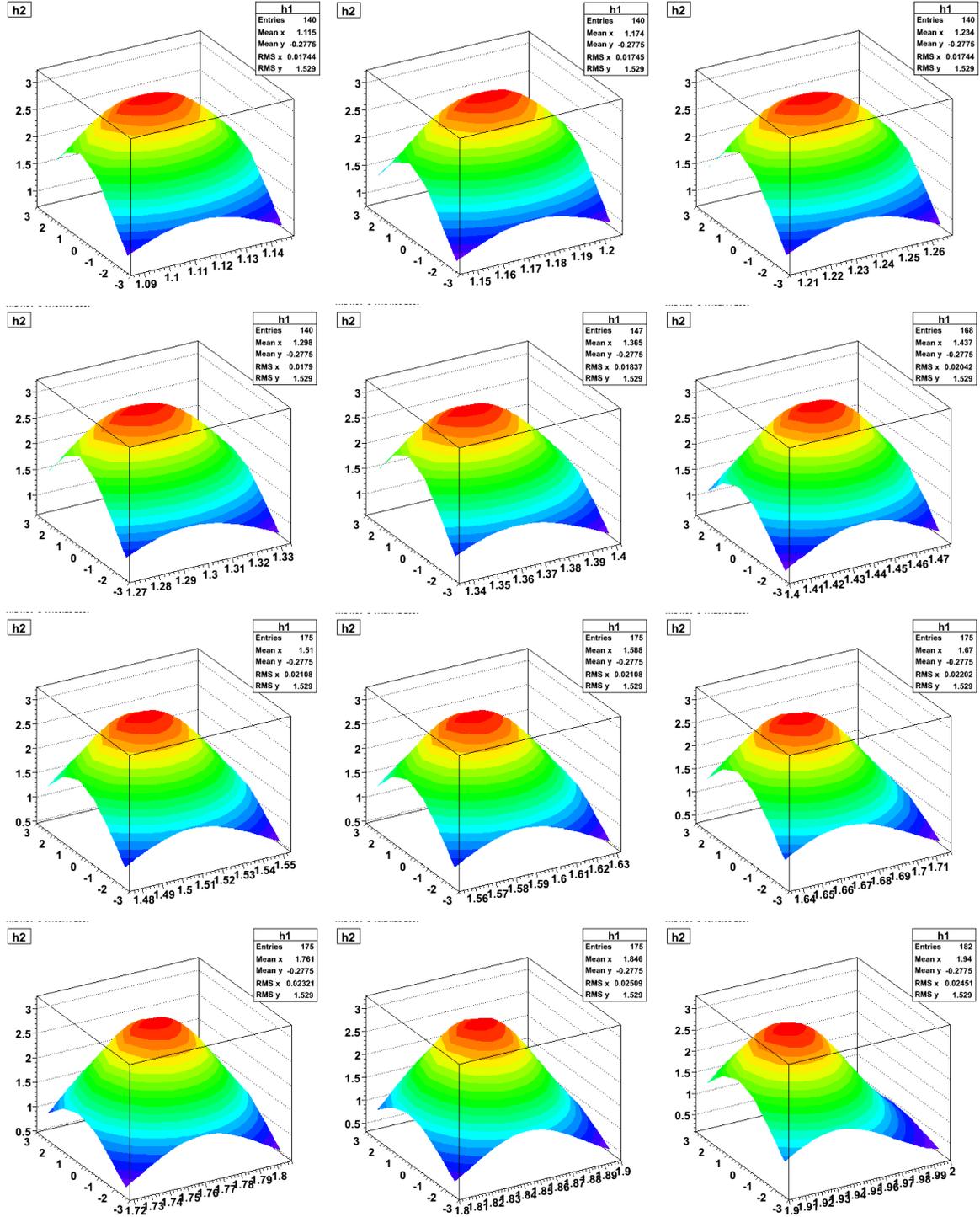


Figure 5.17: The bivariate Gaussian distributions of the tower ET cuts peaked at 3 GeV at the center of individual towers from the twelve etabins. For each plot, the z axis shows the ET cut value, x-axis shows the tower span ϕ from -3 to 3 degree, and the y-axis shows the η range for each etabin.

5.2.3 The optimization of software setup

Before finalizing the algorithm of the π^0 finder software for real data, we tried a few changes in the algorithm and on the cuts in the code, and compared the pT dependent π^0 yields after background subtraction. One other cut that was different from the simulation studies is that the SMD seed strip threshold was raised from 1.5 MeV to 3 MeV. We discuss these changes and optimizations briefly below.

We made six tests of the algorithm for our threshold settings to optimize the software setup. Each time we made one change, everything else in the code was kept unchanged from the default code described above, and then we determined the π^0 yield after background subtraction. We did the background subtraction according to a π^0 peak fitting algorithm. The π^0 invariant mass spectrum was fit with a simple exponential function for the background and a skewed Gaussian function fitting the peak, as shown in equation (5.2):

$$f(x) = A_1 e^{-\alpha x} + A_2 e^{-0.5 \times \left(\frac{x-\mu}{\sigma \times (1+k(x-\mu))} \right)^2} \quad (5.2)$$

Here x is the invariant mass; A_1 denotes the background amplitude; $-\alpha$ denotes the slope of the background shape; A_2 denotes the amplitude of the pion peak; μ denotes the pion peak position; and σ denotes the width of the normal Gaussian distribution. The peak asymmetry parameter k was fixed at 6.11 for all analyses. When we fit the invariant mass spectrum for different pT bins produced from the 155 runs, we allowed k to be variable at first. After examining the k parameter as a function of pT, we decided it is good enough to hold k constant at 6.11. The dependence of k on pT is shown in figure 5.18. Because we only are interested in π^0 s with pT > 5 GeV, a constant k is sufficient.

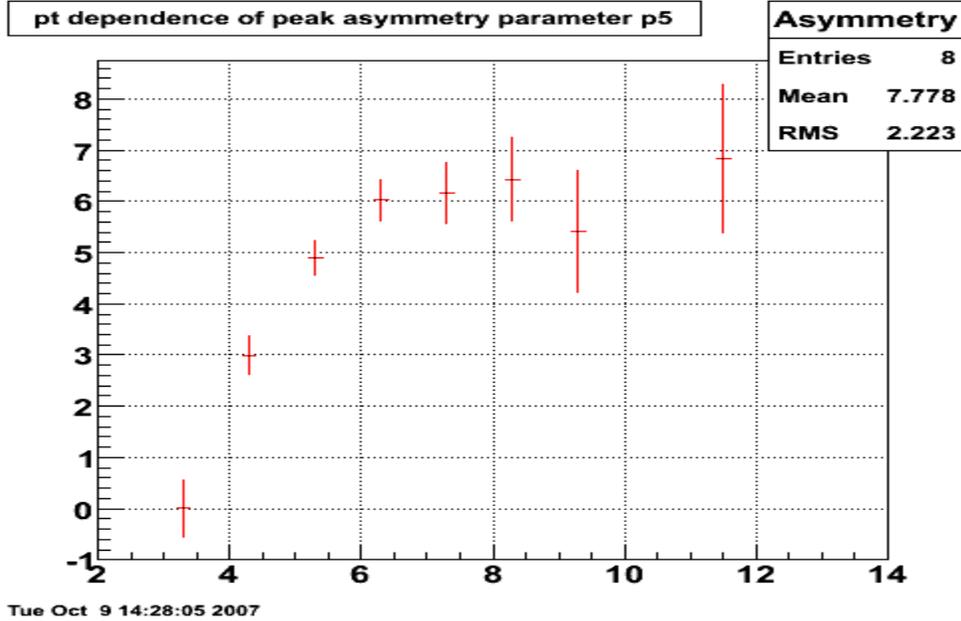


Figure 5.18: The π^0 peak asymmetry parameter k from fits at different p_T bins. In this plot $k=p_5$.

We now discuss briefly some additional tests we made on the π^0 algorithm. The first test was to turn off the splitting algorithm in the Point-Maker. This greatly reduces the low mass peak, but also reduces the spin dependent π^0 yields a lot, so we decided to use the splitting algorithm in real data analyses.

The second test was to raise the point ET cut from 1.5 GeV to 2.0 GeV. This reduces both the background and the pion yield by almost the same factor, without changing the low mass peak much. We decided to keep the 1.5 GeV cut.

The third test was to lower the point number cut around adjacent towers from 4 to 3. This does not change much, so we kept 4 to be consistent with simulation studies.

The fourth test was to only use the two most energetic points from tower clusters to reconstruct the π^0 . We saw little benefit from this new algorithm at high p_T (≥ 7 GeV/c), where we had hoped to see increased π^0 yields. We chose not to use this new algorithm, primarily to be consistent with our simulation studies.

The fifth test was to increase the SMD cluster size from 7 to 11. This suppresses the fluctuations from the SMD strips a lot, which lowers the background, but the π^0 reconstruction efficiency is also very low, especially at high pT when the photon separation is smaller. We therefore kept the SMD cluster size at seven.

The last test was to raise the SMD seed strip threshold from 1.5 to 2 MeV. This increased the π^0 yield a little, as well as the background. We decided to raise the SMD seed strip threshold again, from 2 to 3 MeV. This increased the π^0 yield by a larger factor than background. But both increased. To avoid introducing too much additional background, we decide to fix the seed strip thresh at 3 MeV for the real data analysis, and not increase the threshold anymore. Though we have seven pT bins, we only show one invariant mass spectrum fit for now, from pT bin [8, 9] GeV/c. The data and various fit curves are shown in figure 5.19, and are explained in the figure caption.

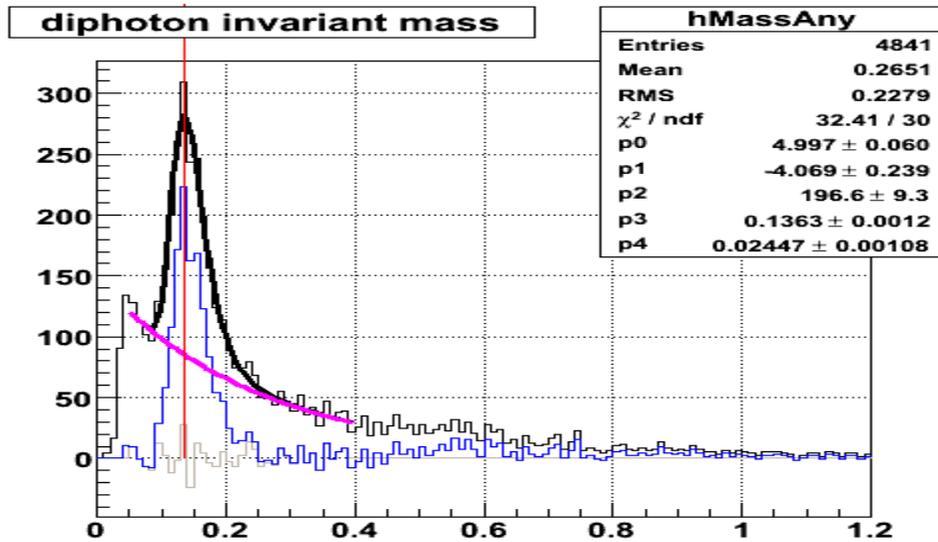


Figure 5.19: Invariant mass spectrum fit from pT bin [8, 9] GeV/c with the SMD seed threshold at 3 MeV. The black curve is the overall fit; the pink curve is the background; the blue curve is the pion yield with raw spectrum minus the pink curve; the gray curve is the fit residual. The red line is the ideal pion peak at 0.135 GeV.

5.3 The double spin asymmetry A_{LL}

5.3.1 Overview

We ran through all 347 selected longitudinal pp runs from the 2006 data to generate a number of π^0 trees with the optimal algorithm and cuts described above. These π^0 trees store the reconstructed π^0 information event by event. The most important quantities to extract for this analysis are the four spin-dependent π^0 yields. We will use equation (10) to calculate the double spin asymmetry A_{LL} . It is important to first normalize the spin-dependent π^0 yields according to the relative luminosities of the colliding proton beams. To extract π^0 and background yields after proper normalization of the four spin-sorted invariant mass spectra, we decided to fit the invariant mass spectra for different pT bins and for the four spin states using the simple function given in equation (38). We only allowed the amplitude A_1 of the exponential background and the amplitude A_2 of the π^0 peak to vary during the fitting procedure for each spin state at a given pT bin. This means we held fixed all other fit parameters (α , μ , and σ) for the four spin states at a given pT bin, to keep the function as simple as possible. Different pT bins were allowed to have different constant parameter values. We also checked the fit function for consistency with a pythia event sample by comparing it to the real data result. Based on the four normalized spin-dependent yields, we can then calculate the double spin asymmetry and the background asymmetry with statistical uncertainties for our inclusive neutral π^0 study. We also calculated the (parity violating) single spin asymmetries for the blue and yellow beams to test for systematic problems. Since we are using an empirical fitting method to deal with the background, it was essential that we studied the sensitivity of these asymmetries to some of the choices we made for the fits. We estimated the size of our

systematic error due to the fits in the following three ways: varying the fitting range, changing the fixed values for α , μ , σ by twice their error for each p_T bin, and using different fit functions.

5.3.2 Yields after normalization

Some fundamental features of the reconstructed π^0 s obtained from analysis of the full statistics (but before normalization) are presented in this section.

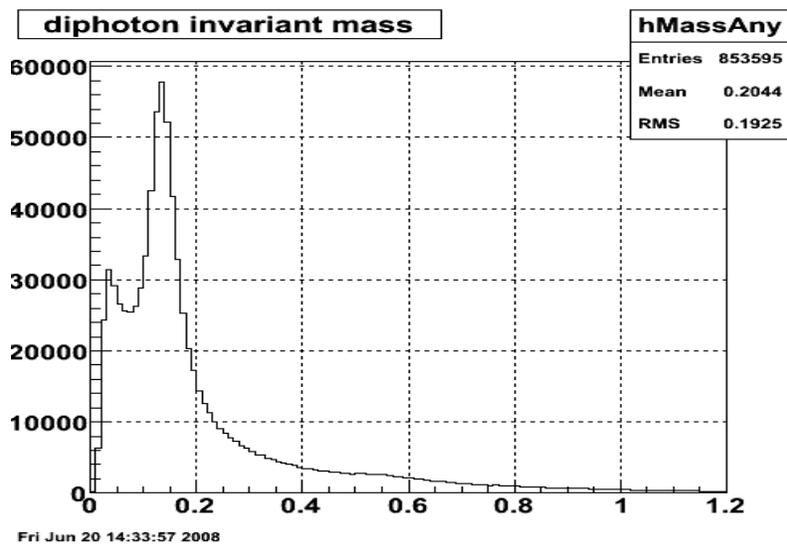


Figure 5.20: The invariant mass spectrum of reconstructed π^0 s from 347 runs, summed over all p_T and spin states.

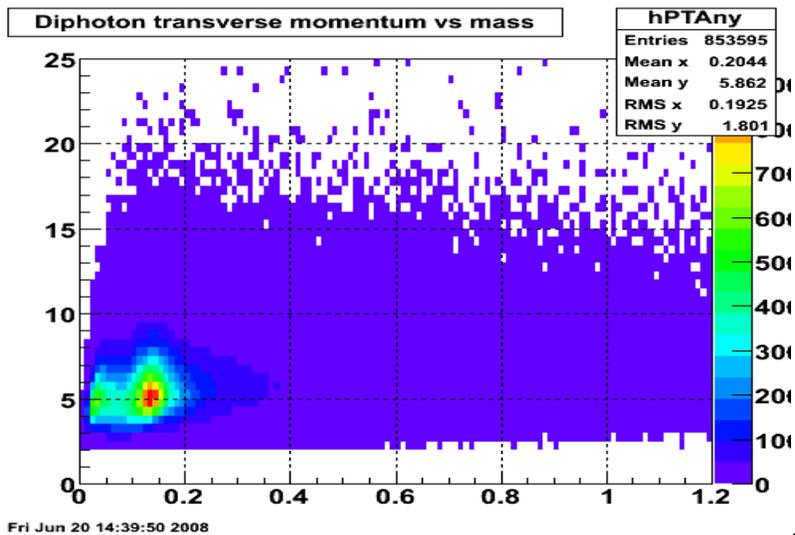


Figure 5.21: Reconstructed π^0 p_T vs invariant mass from 347 runs.

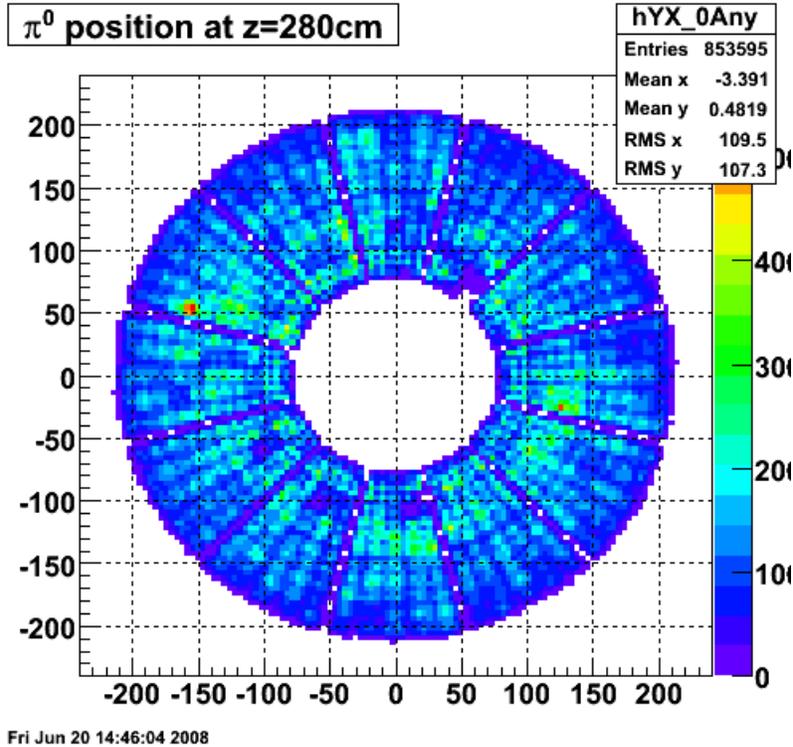


Figure 5.22: Reconstructed π^0 distribution in the endcap. The twelve sector boundaries are clearly visible, as are the individual towers.

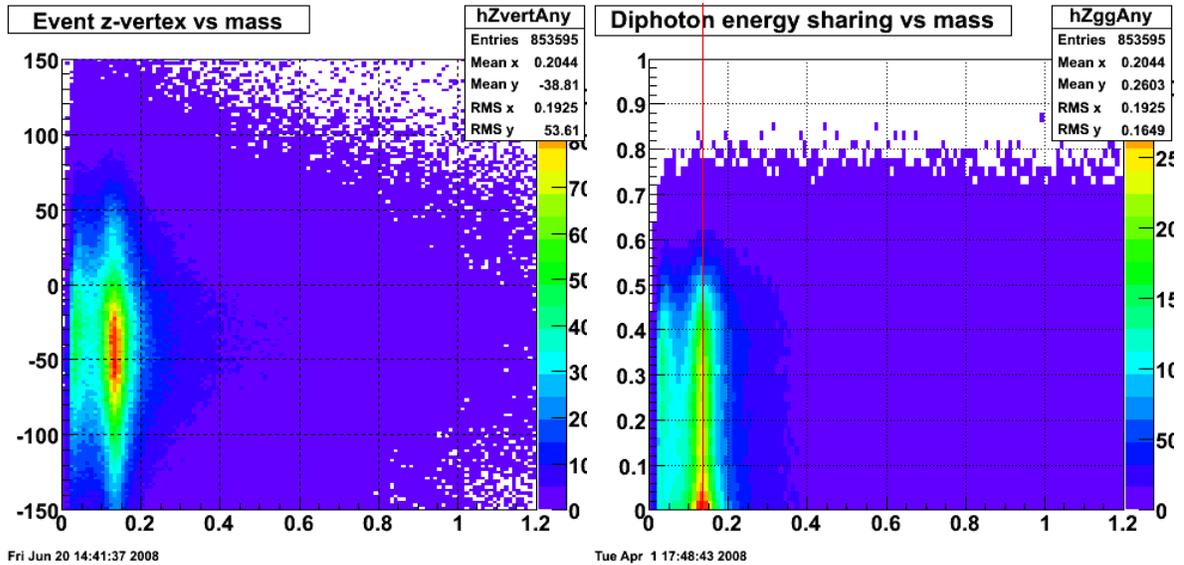


Figure 5.23: (left plot) The collision z -vertex vs π^0 invariant mass. The constant ET tower threshold in the trigger favors events in the negative z region. (right plot) Reconstructed $\pi^0 Z_{\gamma\gamma}$ vs mass. The 1.5 GeV point ET cut greatly reduces the yield at large $Z_{\gamma\gamma}$.

The raw π^0 yields produced using full statistics are indicated in figures 5.20 to 5.23. We extract the π^0 yield after background subtraction within an invariant mass window of [0.1, 0.18] GeV, and obtain about 200k reconstructed π^0 s. This gives us a trigger reconstruction efficiency of about 9%, since the total number of L2gamma trigger events is about 2.2 million.

But we can only calculate spin asymmetries after proper yield normalization. There are four spin states used in our analysis: PP, PN, NP, and NN, arranged in the order of blue and yellow beams with P denoting polarization along beam momentum and N denoting polarization opposite beam momentum in the STAR database. The four spin-dependent π^0 yields are generated and stored in the π^0 trees by calling the STAR database definitions with 5 to PP, 6 to PN, 9 to NP and 10 to NN. We used the relative luminosity file analyzed by Tai Sakuma for the 2006 run [97] to normalize the four spin-dependent π^0 yields. The luminosity file is organized by the BBC time-bin number from 1 to 15 for each run, with the fill number information. With the time-bin information we can normalize the pion yield by each bin. Unfortunately, the organization of the four spin-state luminosities is based on a different convention from the STAR database, and uses uu, du, ud and dd to represent the corresponding luminosity numbers from PP, NP, PN and NN states, with the order of yellow and blue beams reversed. So when we match the luminosity file to our spin-dependent π^0 yields in these trees, we account for this difference in conventions. We read these trees and normalize the yields event by event. We use L_{uu} , L_{du} , L_{ud} and L_{dd} to represent the luminosities from the file, and N_{uu} , N_{du} , N_{ud} and N_{dd} to denote the raw yields in our π^0 trees. The yields after normalization are: $N_{PP}=N_{uu}$, $N_{PN}=N_{ud} \times \frac{L_{uu}}{L_{ud}}$, $N_{NP}=N_{du} \times \frac{L_{uu}}{L_{du}}$, and $N_{NN}=N_{dd} \times \frac{L_{uu}}{L_{dd}}$. After sorting over all these

π^0 trees, and to finish the normalization procedure, the four spin-dependent productions are stored in four different two-dimensional histograms with transverse momentum information, so that we can fit the invariant mass spectra over different ranges of pT.

We chose to fit the π^0 invariant mass spectra with the function given in equation (40) for seven transverse momentum pT bins: [4, 5), [5, 6), [6, 7), [7, 8), [8, 9), [9, 10) and [10, 25] GeV/c. As we discussed, we wanted to keep the fitting function as simple as possible, so we combined an exponential function for the background with a skewed Gaussian function for the π^0 s. Furthermore, we wanted to fix the parameters α , μ , and σ to their optimal values for each pT bin. Our procedure was to add the four normalized spin-dependent production histograms together, do a preliminary fit to the spin-summed invariant mass spectrum for each pT bin, and then fix the above three parameters according to this fitting result.

Table 5.2: Fit ranges and fixed parameters in the final fitting to extract π^0 .

pT bins (GeV/c)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)	[9, 10)	[10, 25)
Fit range (GeV)	[0.05, 0.4]	[0.04, 0.4]	[0.03, 0.4]	[0.04, 0.4]	[0.04, 0.4]	[0.05, 0.4]	[0.06, 0.4]
α	8.262	6.752	5.095	4.041	3.370	2.942	2.668
μ	0.1305	0.1353	0.1359	0.1375	0.1374	0.1370	0.1392
σ	0.02073	0.02164	0.02307	0.02498	0.02666	0.02660	0.02792

These fixed parameters are shown in table 5.2, based on fits to the summed invariant mass spectra for each pT bin. These will be used in the fits to the four spin-dependent invariant mass spectra for the corresponding pT bin. Before doing this, it is very useful to test our assumption that these three parameters are the same for all spin states, within errors. For each pT bin, we have four fit plots for the PP, PN, NP and NN spin states. We fit them all independently and found that the values given in table 5.2 are reasonable for all four spin states. We show only one example in figure 5.24, for the [4,7) GeV/c pT bin.

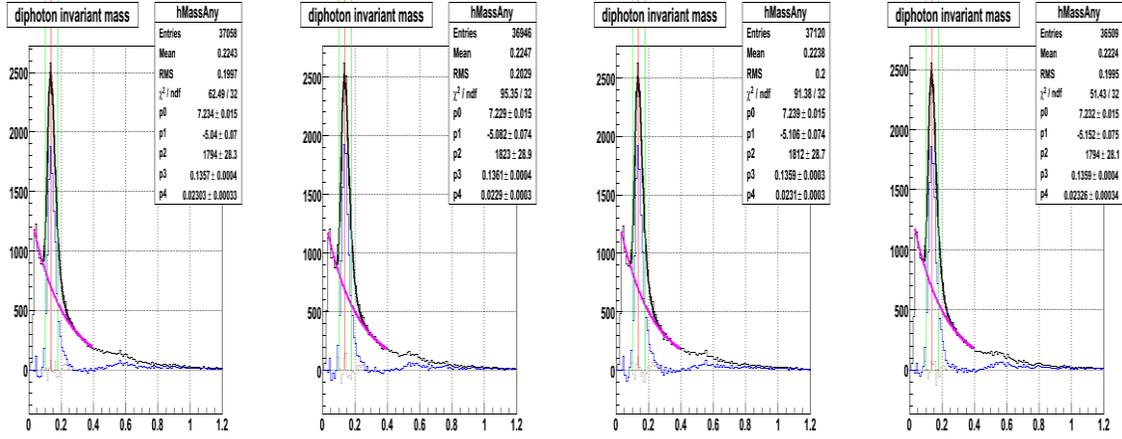


Figure 5.24: Independent fits to the four spin-dependent invariant mass spectra from pT bin [6, 7) GeV/c to check the consistency of fixed parameters α , μ and σ . The four plots from left to right denote states PP, PN, NP and NN.

Finally, we apply the fit ranges and fixed parameters given in table 5.2 to the fitting function, and perform the fits on our spin-dependent invariant mass spectra to extract our normalized π^0 yields at each pT bin.

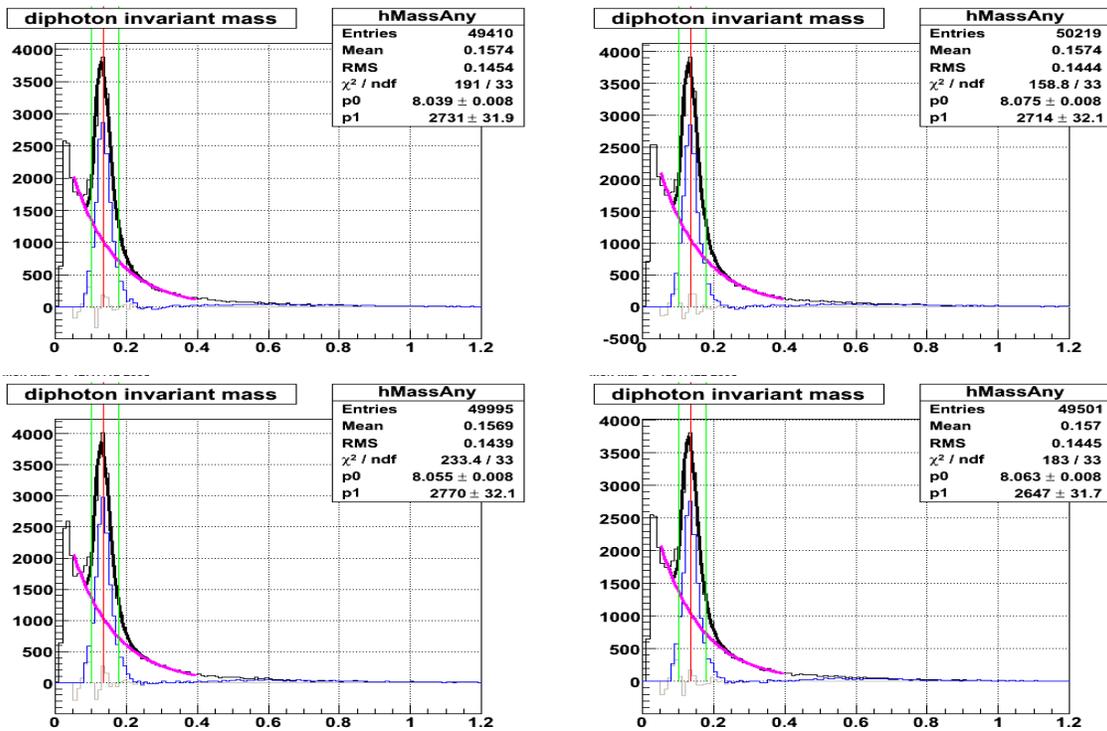


Figure 5.25: Final fits on the four spin dependent invariant mass spectra for the pT bin [4, 5) GeV/c. The four spin states are PP (upper-left), PN (upper-right), NP (lower-left) and NN (lower-right) from

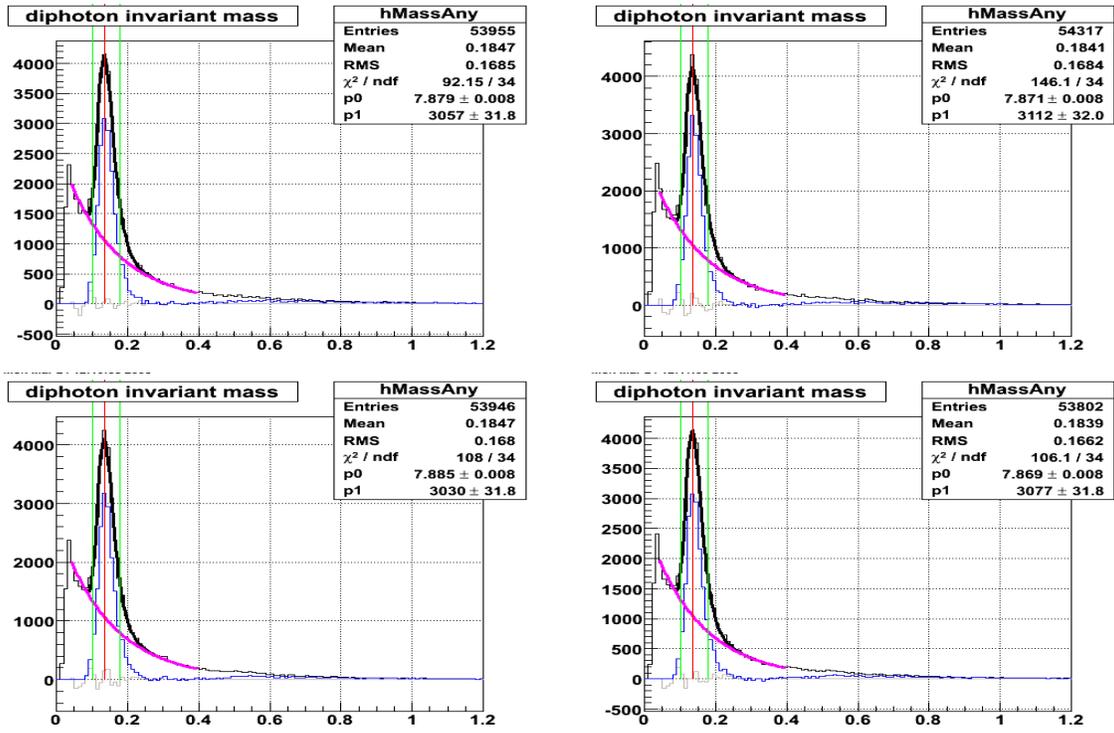


Figure 5.26: Same as figure 5.25, but for the [5, 6) GeV/c pT bin.

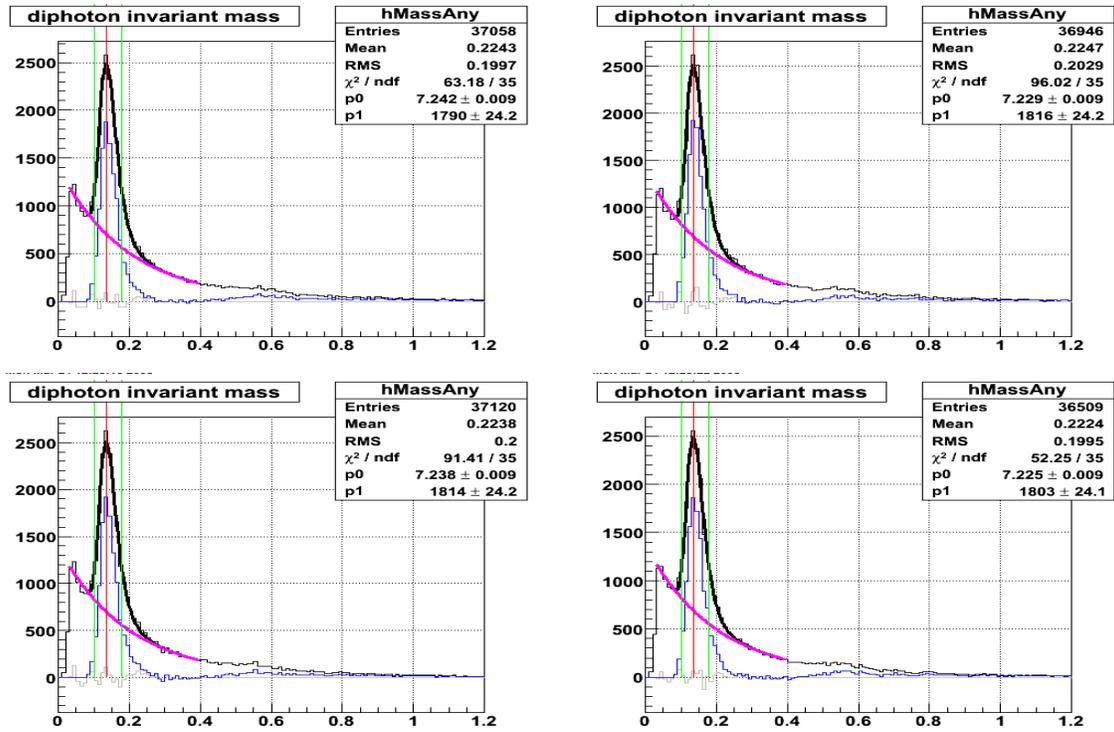


Figure 5.27: Same as figure 5.25, but for the [6, 7) GeV/c pT bin.

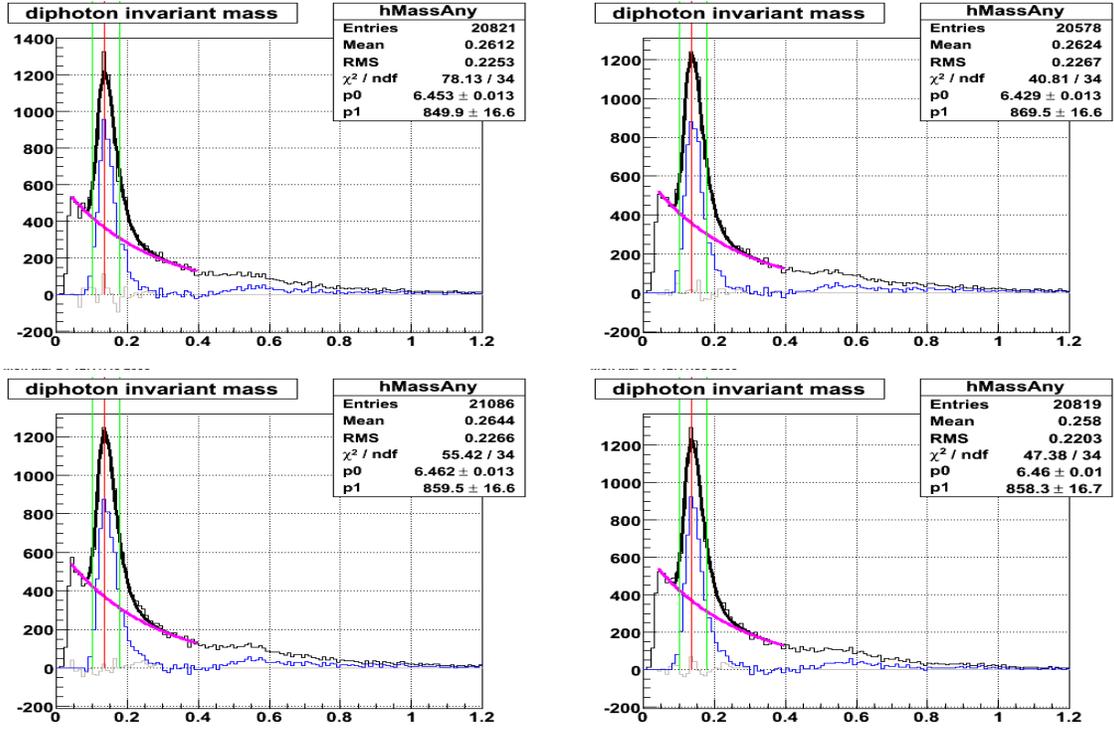


Figure 5.28: Same as figure 5.25, but for the [7, 8) GeV/c pT bin.

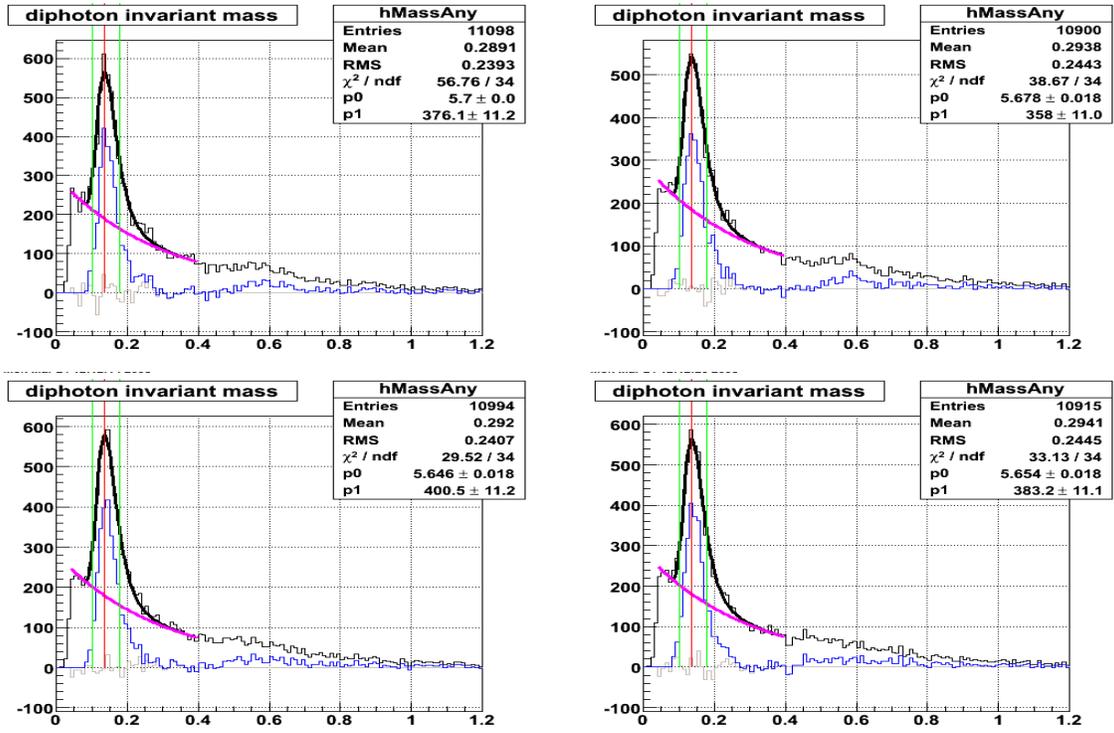


Figure 5.29: Same as figure 5.25, but for the [8, 9) GeV/c pT bin.

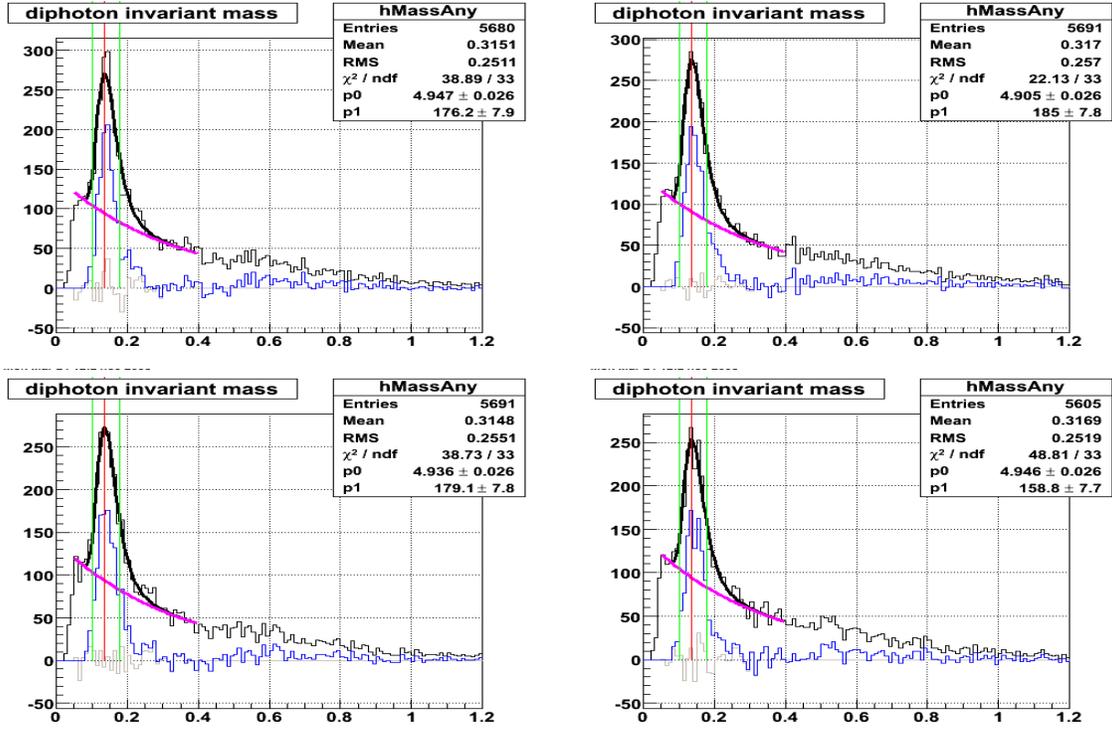


Figure 5.30: Same as figure 5.25, but for the [9, 10) GeV/c pT bin.

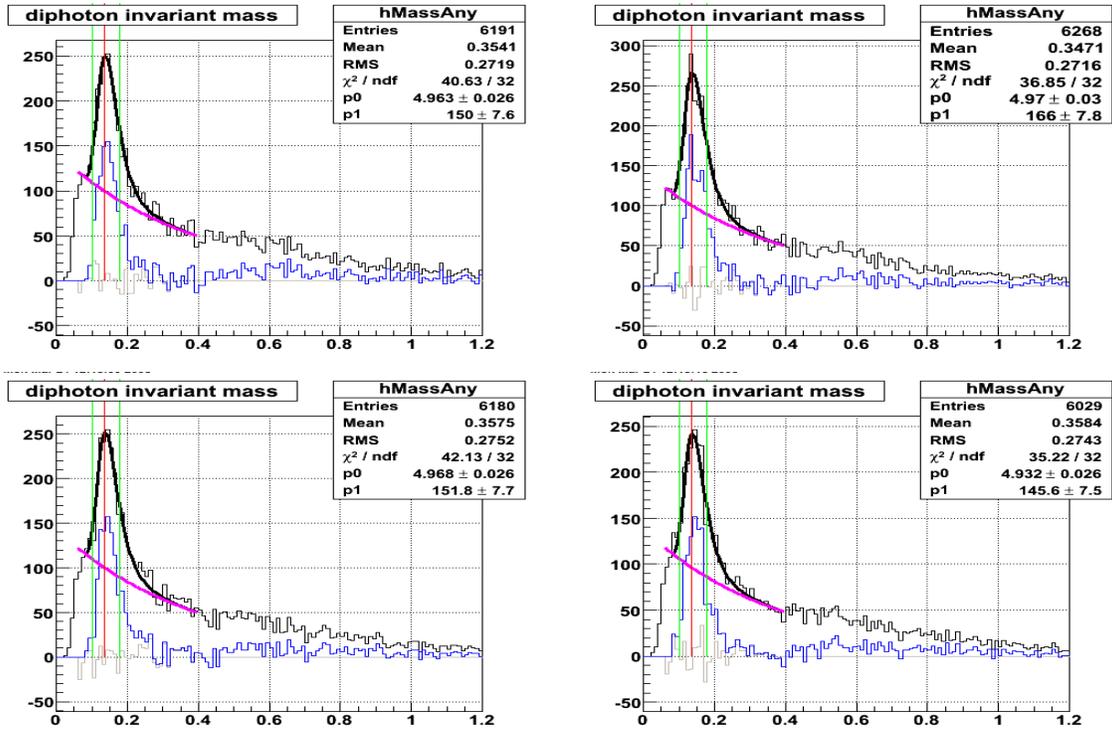


Figure 5.31: Same as figure 5.25, but for the [10, 25] GeV/c pT bin.

All the spin-sorted spectra and their final fits are shown in figures 5.25 to 5.31, for each of the seven pT bins. Each figure has four spin-dependent plots. We extract the π^0 yield and background contribution from each plot to get the yields after normalization. For each plot shown in figures 5.25 to 5.31, the total fit function is shown by the black curve. The pink curve is the exponential background. The blue curve is equal to the original histogram (data) minus the exponential background. The gray curve indicates the fit residual. The two green lines show the mass window of [0.1, 0.18] GeV over which the π^0 yield is summed. The raw yield is the total yield within the mass window, the background is the yield under the pink curve within the mass window, and the final π^0 yield comes from the blue curve within the mass window, which means the π^0 yield is the raw yield after background subtraction in our analyses. All of these yields after normalization and fitting are summarized in the following table 5.3.

Table 5.3: Summary of the π^0 yields from all pT bins. raw= π^0 +bg (background) ± 1 . The additional uncertainty 1 is from the coding difference between integer and float.

pT bins (Gev/c)			[4,5)			[5, 6)			[6, 7)		
Yield type			Raw	π^0	bg	raw	π^0	bg	Raw	π^0	bg
PP			22662	14046	8615	25451	16387	9063	16091	10030	6060
PN			22950	14019	8931	25607	16609	8998	16137	10160	5977
NP			23005	14252	8752	25343	16221	9121	16127	10095	6032
NN			22409	13592	8817	25392	16418	8973	16033	10076	5957
[7,8)			[8, 9)			[9, 10)			[10, 25]		
raw	π^0	bg	Raw	π^0	bg	raw	π^0	bg	raw	π^0	bg
8227	5030	3196	3969	2312	1656	1909	1080	828	1843	966	876
8266	5146	3120	3806	2185	1620	1927	1131	795	1946	1064	882
8280	5054	3225	4020	2450	1569	1945	1125	820	1827	946	880
8259	5039	3219	3960	2379	1581	1828	999	828	1774	924	849

Based on the summary table 5.3, we can calculate spin asymmetries and then statistical errors.

5.3.3 Consistency check on pythia events

As explained above, we fix α , μ and σ to the values shown in table 5.2, and apply them in the final fitting to extract the yields. It is important to see if these values are in good agreement with our simulated data or not. We applied the same fit function given in equation (40) to a pythia event sample, with 208k events and partonic pT from 11 to 15 GeV/c. We do not get very large π^0 yield from pythia events, but the statistics were enough to check the fit consistency of the results. We fit the pythia event sample over the invariant mass range [0.07, 0.4] GeV/c for π^0 candidates in the pT range [6, 25] GeV/c.

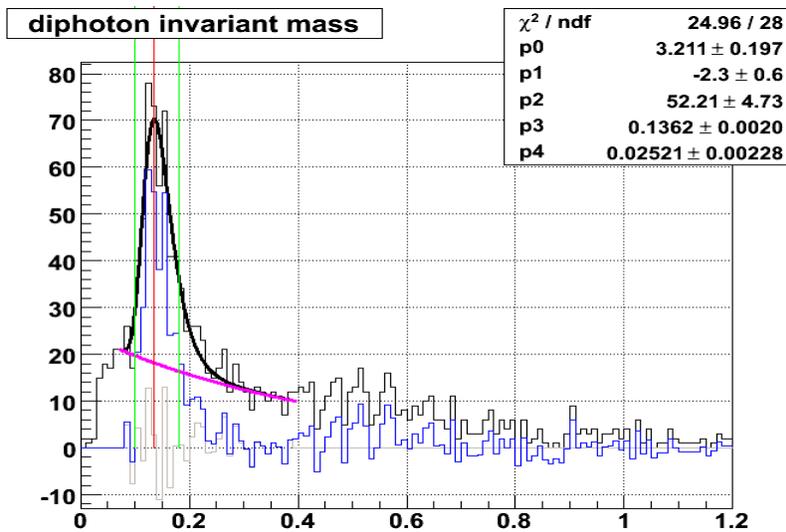


Figure 5.32: Results of a fit to the pythia event sample with partonic pT [11, 15] GeV/c.

The fit results for the pythia event sample are shown in figure 5.32. p_1 is the slope of the exponential background and is very sensitive to the SMD fluctuation, so we do not expect it to agree with the real data. $p_3(\mu)$ denotes the centroid of the π^0 peak, and $p_4(\sigma)$ denotes the width of the peak. It is meaningful to check the consistency of the latter two parameters with the values found in fits to the real data. These comparisons are plotted in

figures 5.33 and 5.34, and the fit results on the real data and on the pythia sample show very good agreement with each other. The peak centroids and widths are plotted versus p_T bin, with the mean p_T and statistical error shown for each point. In each plot, the blue point is from the pythia sample and the red points are from real data.

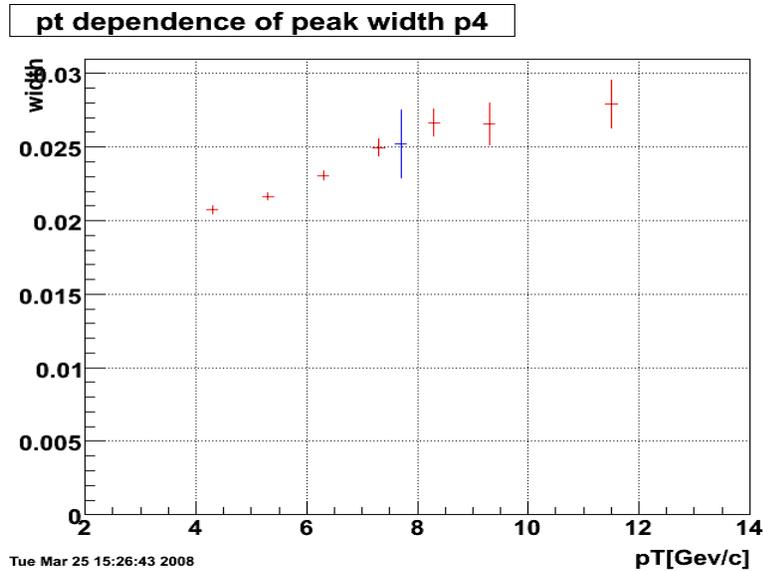


Figure 5.33: Width $\sigma=p_4$ comparison between the pythia sample and real data.

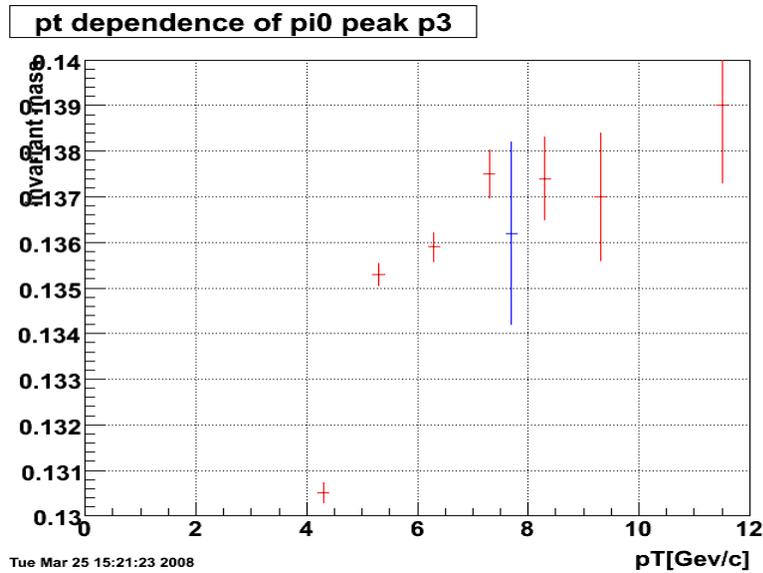


Figure 5.34: Peak centroid $\mu=p_3$ comparison between the pythia sample and real data. Note the greatly expanded vertical scale.

5.3.4 A_{LL} and statistical uncertainties calculation

We calculate the double spin asymmetry A_{LL} according to equation (1.10) shown in chapter 1, which we repeat below:

$$A_{LL} = \frac{\sigma_{++} - \sigma_{+-}}{\sigma_{++} + \sigma_{+-}} = \frac{1}{P_y * P_b} * \frac{\left(N_{++} + \frac{L_{++}}{L_{--}} N_{--} \right) - \left(\frac{L_{++}}{L_{+-}} N_{+-} + \frac{L_{++}}{L_{-+}} N_{-+} \right)}{\left(N_{++} + \frac{L_{++}}{L_{--}} N_{--} \right) + \left(\frac{L_{++}}{L_{+-}} N_{+-} + \frac{L_{++}}{L_{-+}} N_{-+} \right)} \quad (1.10)$$

We extracted the normalized yields in the previous section, and the only other information needed for the A_{LL} calculations are the yellow and blue beams polarization data. The complete polarization data for the 41 fills we used is provided in appendix B. We selected the 347 runs from these 41 fills, and the average polarizations P_y and P_b from the yellow and blue beams respectively can be calculated from the polarization file. We simplify the double spin asymmetry calculation formula by expressing it in terms of the normalized yields as:

$$A_{LL} = \frac{1}{P_y \times P_b} \times \frac{N_{PP} + N_{NN} - N_{PN} - N_{NN}}{N_{PP} + N_{NN} + N_{PN} + N_{NN}} \quad (5.3)$$

Here we are using the final π^0 yield values shown in the middle column for each pT in table 5.3 to calculate A_{LL} for each pT bin. The average polarizations from appendix B are $P_y \approx 56\%$ and $P_b \approx 54\%$ which gives $P_b \times P_y \approx 0.30$. We define the statistical uncertainties for the real π^0 yield as $\sigma(real)$, for the background yield as $\sigma(bg)$, and for A_{LL} as $\sigma(\pi^0)$ and calculate them in the following formulae:

$$\sigma(real) = \frac{1}{P_y \times P_b} \times \frac{1}{\sqrt{\text{Sum of the real } \pi^0 \text{ yield}}} \quad (5.4)$$

$$\sigma(bg) = \frac{1}{P_y \times P_b} \times \frac{1}{\sqrt{\text{Sum of the background } \pi^0 \text{ yield}}} \quad (5.5)$$

$$\sigma(\pi^0) = \sqrt{\sigma^2(\text{real}) + \sigma^2(\text{bg})} \quad (5.6)$$

The results for A_{LL} and the purely statistical uncertainties for each pT bin are summarized in the following table 5.4.

Table 5.4: The summary of double spin asymmetries and statistical errors for the seven pT bins.

pT bins (GeV/c)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)	[9, 10)	[10, 25)
A_{LL}	-0.0377	-0.00121	-0.0122	-0.0214	0.0197	-0.135	-0.102
$\sigma(\pi^0)$	0.0227	0.0218	0.0272	0.0377	0.0540	0.0772	0.0777

5.3.5 Systematic errors in the fitting procedure

We estimate systematic errors in the fitting process in three ways: first, we use the same fit function given in equation (5.2) while varying the fitting range to estimate one source of systematic error; second, we shift the parameters α , μ , and σ (held fixed in the spin-dependent fits) by twice their fitting error to gauge sensitivity to an incorrect choice of their values; and third, we use a different function for the background shape to estimate error.

Table 5.5: Three sets of fitting ranges used to estimate the first source of systematic error.

pT bins (GeV/C)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)	[9, 10)	[10, 25]
Range 1 (GeV)	[0.03,0.4]	[0.03,0.4]	[0.03,0.4]	[0.05,0.4]	[0.05,0.4]	[0.05,0.4]	[0.07,0.4]
Range 2 (GeV)	[0.04,0.4]	[0.04,0.4]	[0.04,0.4]	[0.04,0.4]	[0.04,0.4]	[0.04,0.4]	[0.06,0.4]
Range 3 (GeV)	[0.05,0.4]	[0.05,0.4]	[0.05,0.4]	[0.06,0.4]	[0.06,0.4]	[0.06,0.4]	[0.08,0.4]

We chose three fitting ranges for each pT bin for the same function in equation (5.2).

The selection is shown in table 5.5. We basically shifted the starting channel by ± 10 MeV,

but kept the end channel fixed at 0.4 GeV in the invariant mass spectra. We then extracted spin-dependent π^0 yields from the three different fit ranges with the same function. The extraction procedure is the same as we described in section 5.3.2. Then we calculated the double spin asymmetry for each of the three ranges. The maximal difference among the three asymmetries was used as the magnitude of this systematic error for each pT bin. We assigned a systematic error, in the ascending pT order from [4, 5) to [10, 25] GeV/c as: 0.00158, 0.00488, 0.00422, 0.00294, 0.0116, 0.0101 and 0.00763.

Table 5.6: The fit ranges and fixed parameters used to extract π^0 and estimate the second kind of systematic error.

pT bins (GeV/c)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)	[9, 10)	[10, 25)
Fit range (GeV)	[0.03, 0.4]	[0.03, 0.4]	[0.03, 0.4]	[0.05, 0.4]	[0.05, 0.4]	[0.05, 0.4]	[0.07, 0.4]
α	8.33	6.82	5.15	4.14	3.52	3.0	2.9
μ	0.1308	0.1356	0.136	0.137	0.138	0.1375	0.1385
σ	0.0208	0.2166	0.0231	0.025	0.027	0.027	0.029

To estimate the second type of systematic error, we used the fit ranges and π^0 peak parameters shown in table 5.6. To be consistent, we used range 1 here to fit the spin-dependent invariant mass spectra, and α , μ and σ were fixed at values two sigma away from what we used in table 5.2. We followed the same steps of section 5.3.2 to fit spectra, extract π^0 yields and calculate the double spin asymmetry. The absolute values of the differences between these A_{LL} and those from the final fit range above are defined as the second kind of systematic error for each pT bin. The results are: 0.000174, 0.000056, 0.000028, 0.00052, 0.00002, 0.00008 and 0.0005 along the ascending order of the seven pT bins.

For the third type of systematic error study, we used a more complex function to describe the background under the π^0 peak. We obtained slightly better χ^2 by using

$$f(x) = A_1 e^{-\alpha x + \frac{\beta}{x}} + A_2 \times e^{-0.5 \times \left(\frac{x-\mu}{\sigma \times (1+k \times (x-\mu))} \right)^2} \quad (5.7)$$

We applied the function shown in equation (5.9), and followed the same steps in section 5.3.2 to obtain spin-dependent spectra, extract π^0 yields, and calculate the double spin asymmetry. We still use fit range 1 for each of the pT bins. The absolute values of the differences between these A_{LL} and those from the final fit range defined as the third kind of systematic error. The results are: 0.000638, 0.0002751, 0.0006, 0.00059, 0.00024, 0.00065 and 0.0023 along the ascending order of the seven pT bins.

In summary, we denote the first kind of systematic error (by shifting the fitting ranges) as $s1(\pi^0)$, the second kind of systematic error (from different fitting parameters) as $s2(\pi^0)$, and the third kind of systematic error (from using a different fit function) as $s3(\pi^0)$. The total systematic error $s(\pi^0)$ is summarized below:

Table 5.7: Summary of systematic errors along pT bins for the double spin asymmetry.

pT bins (GeV/C)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)	[9, 10)	[10, 25]
$s1(\pi^0)$	0.00158	0.00488	0.00422	0.00294	0.0116	0.0101	0.00763
$s2(\pi^0)$	0.000174	0.000056	0.000028	0.00052	0.00002	0.00008	0.0005
$s3(\pi^0)$	0.000638	0.0002751	0.0006	0.00059	0.00024	0.00065	0.0023
$s(\pi^0)$	0.00239	0.00521	0.00485	0.00405	0.0120	0.0108	0.0104

5.3.6 The single spin asymmetries

It is useful to examine the single spin asymmetries A_L from both the yellow and blue beams, to check for consistency of the normalized π^0 yields. A_L is parity-violating, and

therefore is expected to be close to zero. We calculate the single spin asymmetries for the yellow beam (ε_{L_Y}) and the blue beam (ε_{L_B}), and their statistical uncertainties with:

$$\varepsilon_{L_Y} = \frac{N_{PP} + N_{PN} - N_{NP} - N_{NN}}{N_{PP} + N_{PN} + N_{NP} + N_{NN}} \quad (5.8)$$

$$\varepsilon_{L_B} = \frac{N_{PP} + N_{NP} - N_{PN} - N_{NN}}{N_{PP} + N_{NP} + N_{PN} + N_{NN}} \quad (5.9)$$

$$(\Delta\varepsilon_{L_Y})^2 = \left(\frac{\Delta\varepsilon_{L_Y}(raw)}{1-w}\right)^2 + \left(\frac{w \times \Delta\varepsilon_{L_Y}(bg)}{1-w}\right)^2 + \left(\Delta w \times \frac{\varepsilon_{L_Y}(raw) - \varepsilon_{L_Y}(bg)}{(1-w)^2}\right)^2 \quad (5.10)$$

$$(\Delta\varepsilon_{L_B})^2 = \left(\frac{\Delta\varepsilon_{L_B}(raw)}{1-w}\right)^2 + \left(\frac{w \times \Delta\varepsilon_{L_B}(bg)}{1-w}\right)^2 + \left(\Delta w \times \frac{\varepsilon_{L_B}(raw) - \varepsilon_{L_B}(bg)}{(1-w)^2}\right)^2 \quad (5.11)$$

Here $w = \frac{\text{sum of background yield}}{\text{total raw yield}}$, $\Delta\varepsilon_{L_Y}(raw) = \Delta\varepsilon_{L_B}(raw) = \frac{1}{\sqrt{\text{sum of total raw yield}}}$,

$\Delta\varepsilon_{L_Y}(bg) = \Delta\varepsilon_{L_B}(bg) = \frac{1}{\sqrt{\text{sum of total background yield}}}$ and $\Delta w = \frac{\sqrt{\text{sum of background}}}{\text{sum of raw}}$.

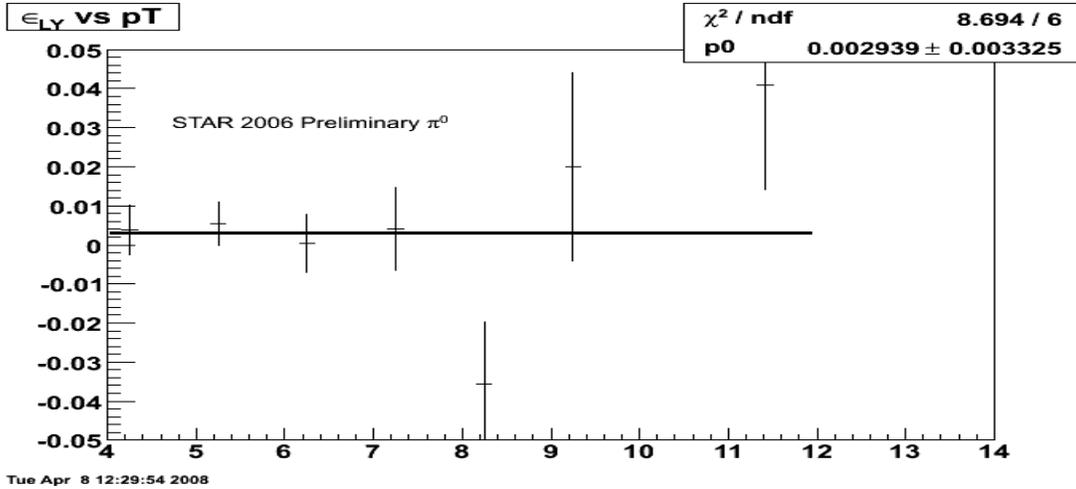


Figure 5.35: The single spin asymmetry for the yellow beam along pT. P_0 is the average value and error.

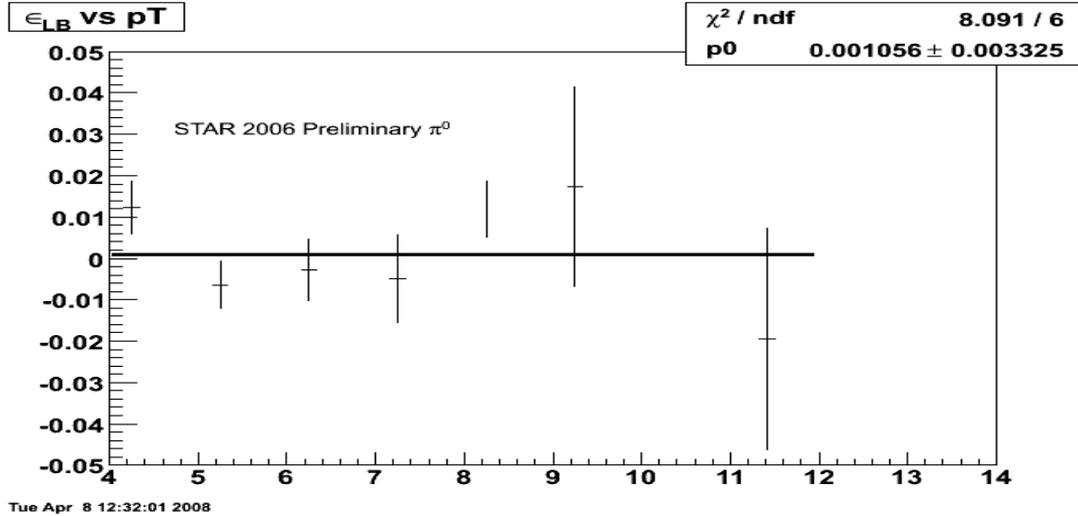


Figure 5.36: The single spin asymmetry for the blue beam along p_T .

The single spin asymmetry results calculated from the data in table 5.3 are shown in figures 5.35 and 5.36. Each asymmetry is plotted vs p_T bins centered at the mean p_T with statistical uncertainties. The constant fit parameter p_0 indicates that the single spin asymmetries from both beams are consistent with zero.

5.3.7 The background asymmetry

We also studied the asymmetry of the background that lies beneath the π^0 peak as another consistency check. We can take the background yields from table 5.3 and do direct calculations using equations (5.3) and (5.5). This will give us a background check. But we also used a different approach to study the background asymmetry: a side-band analysis. The side-band analysis extracts the background yield from two mass windows: one is from the low mass range of [0.04, 0.08] GeV, and the other is from the high mass range of [0.2, 0.4] GeV. We calculate two background asymmetries, A_{LL_bg1} and A_{LL_bg2} , from the two side bands using equation (5.3) with the background yields, and two uncertainties σ_{bg1} and σ_{bg2} from equation (5.5). We then calculate a simple average

of the two side band results to get the final background asymmetry A_{LL_bg} and its uncertainty σ_{bg} , using the following formulae:

$$A_{LL_bg} = 0.5 \times (A_{LL_bg1} + A_{LL_bg2}) \quad (5.12)$$

$$\sigma_{bg} = 0.5 \times (\sigma_{bg1} + \sigma_{bg2}) \quad (5.13)$$

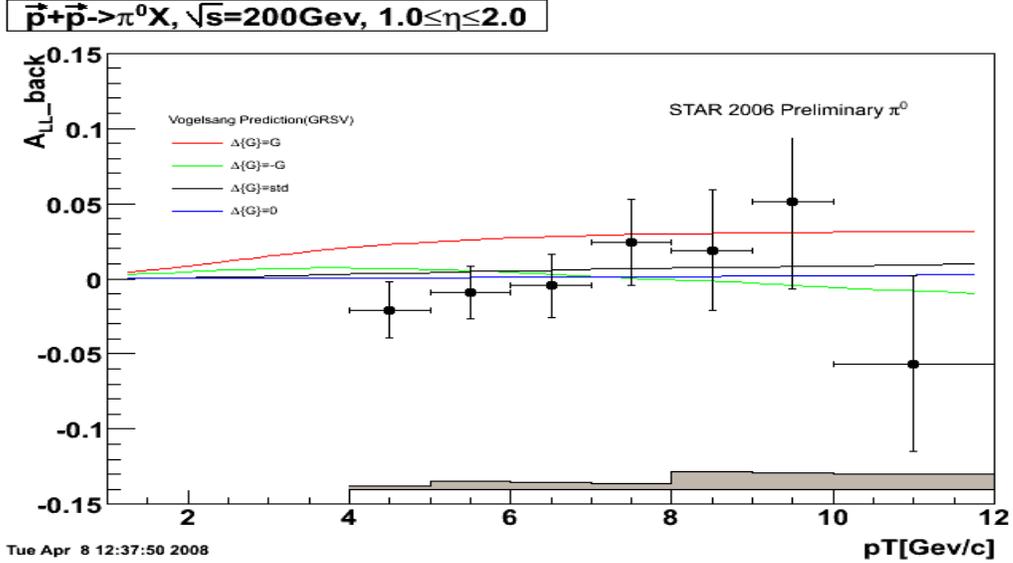


Figure 5.37: The background asymmetry from a side band analysis. The GRSV theoretical predictions are also provided on the same plot purely for scale. Asymmetries are described along p_T bins with statistical uncertainties.

The background asymmetry for this inclusive pion study in the EEMC region is calculated from the side-band analysis. The background asymmetries are consistent with zero within the statistical uncertainties for each p_T point.

5.3.8 Final A_{LL} result

We have described our calculation of double spin asymmetry for inclusive π^0 production in the EEMC. We also calculated the single spin asymmetries from both yellow and blue beams for the 2006 data, and studied the background asymmetry to check for consistency for the whole analysis. The calculation of our double spin

asymmetry and the statistical uncertainty is given in section 5.3.4, while a discussion of systematic errors is presented in section 5.3.5. Here we will give a brief summary of the double spin asymmetry study. We summarize the calculated values for A_{LL} and both uncertainties in table 5.8, and draw a plot to show these features in figure 5.38.

Table 5.8: Final summary of the inclusive π^0 double spin asymmetries and statistical and systematic uncertainties along pT bins from the 2006 longitudinal data in the EEMC.

pT bins (GeV/c)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)	[9, 10)	[10, 25)
A_{LL}	-0.0377	-0.00121	-0.0122	-0.0214	0.0197	-0.135	-0.102
Stat. uncertainty	0.0227	0.0218	0.0272	0.0377	0.0540	0.0772	0.0777
Sys. Error	0.00239	0.00521	0.00485	0.00405	0.0120	0.0108	0.0104

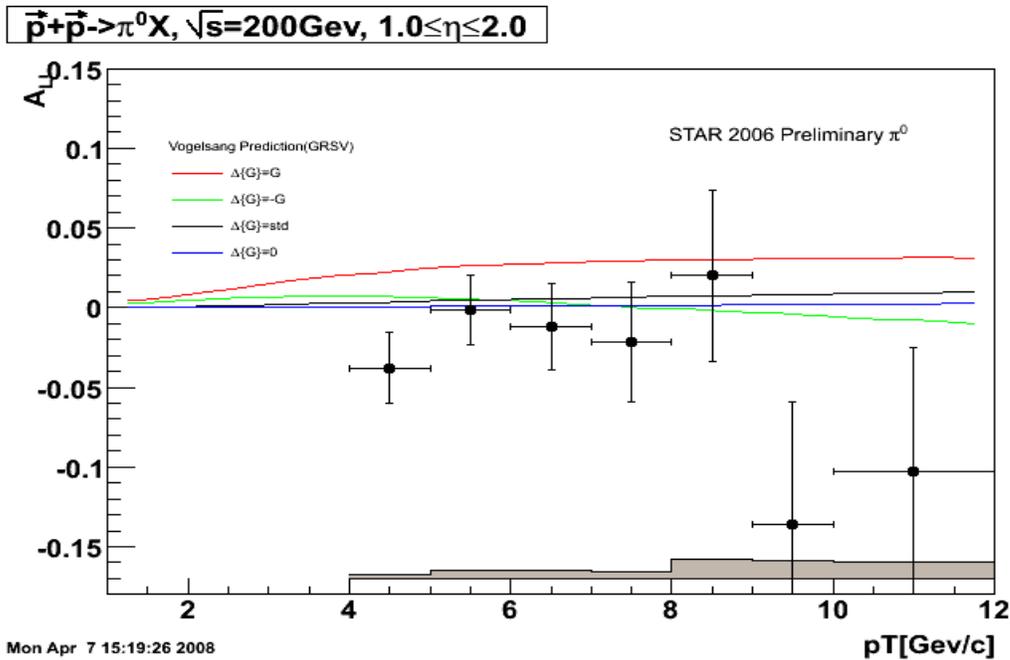


Figure 5.38: Double spin asymmetry A_{LL} with statistical error bars along pT bins. The gray band denotes the systematic errors. The four model predictions from Vogelsang et al. [40] are described in Chapter 1.

We show the A_{LL} plot in figure 5.38. The four colorful curves are theoretical predictions as described in plot. The black A_{LL} points for each p_T bin are accompanied with vertical statistical uncertainties and horizontal p_T ranges. From the plot we can see

that the double spin asymmetries are closely related with the theoretical prediction from at least three cases: $\Delta G=0$, $\Delta G=-G$ and $\Delta G=\text{std}$ within statistical uncertainty bars. But we are not sure about the other situation with large ΔG assumption. For this special $\Delta G=+G$ case, we can calculate the χ^2 and make a statistical test, based on the calculated values and theoretical predictions for each bin.

Table 5.9: A_{LL} theoretical prediction for $\Delta G=+G$ from GRSV [40].

pT bins (GeV/c)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)	[9, 10)	[10, 25)
A_{LL}	0.0226	0.0260	0.0281	0.0295	0.0303	0.0308	0.0314

Based on the non-parametric descriptive statistics method, we calculate the χ^2 from each pair of value between table 5.9 and 5.8 by the following equation:

$$\chi^2 = \sum_i \frac{(O_i - \mu_i)^2}{\sigma_i^2} \quad (5.14)$$

where O_i is the A_{LL} value from real data analyses for each bin, μ_i is the expected value from theory curve, and σ_i is the standard deviation for each point.

In this case ($\Delta G=+G$), we calculate: $\chi^2=17.77$. The number of degree of freedom is 6. The critical value with $\alpha=0.05$ of chi-square is 7.814, and $17.77 > 7.814$. This means the two sets of data are significantly different at 95% confidence level. The critical value with $\alpha=0.01$ of chi-square is 16.8, and $17.77 > 7.814$. This means the two sets of data are significantly different at 99% confidence level.

Based on the analyses, the results shown in figure 5.38 tell us that the double spin asymmetry A_{LL} rules out large ΔG , which is consistent with the STAR inclusive jet results.

Chapter 6

Conclusions

Measurements of the double spin asymmetry A_{LL} for inclusive π^0 production in longitudinally polarized pp collisions can provide important constraints on the gluon's spin contributions to the proton. STAR is well designed to provide good information for the study. We report here preliminary results of the longitudinal double spin asymmetry for inclusive π^0 s in the STAR Endcap Electro-Magnetic Calorimeter (EEMC).

The EEMC provides full azimuthal coverage for $1.086 < \eta < 2.0$ in the forward region of STAR. We have developed the π^0 finder software to reconstruct neutral pions in the EEMC accordingly. Details of the π^0 reconstruction are introduced in the software chapter. The π^0 finder software was tuned up using simulation studies, and provides highly efficient reconstruction of neutral pions in the STAR endcap EMC.

We measured the double spin asymmetry A_{LL} based on the 2006 longitudinally polarized pp data at STAR. 347 runs were selected. We reconstructed π^0 s using the π^0 finder software on events triggered by the EEMC-HTTP-MB-L2gamma trigger, because this trigger is well designed and set up for the neutral pion detection at STAR. While studying π^0 reconstruction from real data, we find a substantial low mass peak in the π^0 invariant mass spectrum. This is due primarily to larger than expected SMD fluctuations in the SMD energy response. To account for this, we decided to extract the π^0 yields by fitting the invariant mass spectrum with a simple function. The function is a simple combination of an exponential function denoting the background and a skewed Gaussian function denoting the π^0 peak. We subtract the background yield from the raw peak

region within the mass window [0.1, 0.18] GeV to get the real π^0 yield. Based on the normalized π^0 yields from four proton spin states, we can calculate spin asymmetries with statistical uncertainties for each of seven pT bins. We also estimated systematic errors based on our specific method of fitting. Three kinds of systematic errors have been calculated, and no large false asymmetries are observed.

We show the first inclusive π^0 double spin asymmetry measurement in this pseudo-rapidity range. Based on the analyses shown in Chapter 5, we see that there are no significant A_{LL} difference between the data result and three theoretical prediction with assumptions of $\Delta G=0$, $\Delta G=-G$ and $\Delta G=std$, because uncertainty bars cross the three curves which makes the χ^2 negligible. But our experimental results differ from the $\Delta G=+G$ prediction at 99% confidence level according to our analysis. As we have discussed at in Chapter 1 and equation (1.8), the double spin asymmetry A_{LL} is proportional to partonic polarization and $\hat{\alpha}_{LL}^{proc}(\hat{s}, \hat{t}, \hat{u})$. Our A_{LL} results rule out large ΔG solutions and are consistent with theoretical equation (1.8) and figure 1.5.

But there are also problems still existing in the EEMC. The complicated SMD response that causes large fluctuations during our analyses is a severe problem. This influences not only the π^0 study, but also the gamma-jet analysis even more. The edge effect noticed in the π^0 reconstruction is due mainly to the trigger selection, but also leaves room for possible improvement in the π^0 finder algorithm. With a bit of work, we might be able to reconstruct π^0 s from point pairs not only from the same sector, but also from different sectors in the future.

This investigation of the double spin asymmetry A_{LL} for inclusive π^0 production in the STAR EEMC from the 2006 longitudinal data has yielded a preliminary result. STAR

is also planning for additional longitudinal pp data running in run9, with greatly increased integrated luminosity. We hopefully will get better statistics to continue our study of the inclusive π^0 double spin asymmetry at forward rapidity.

Appendix A

Run Selection

We select 347 run6 (2006) longitudinal runs from these 41 fills: 7847, 7850, 7851, 7852, 7853, 7855, 7856, 7858, 7863, 7864, 7865, 7871, 7872, 7883, 7886, 7887, 7889, 7890, 7891, 7892, 7893, 7896, 7898, 7901, 7908, 7909, 7911, 7913, 7915, 7916, 7918, 7921, 7922, 7926, 7944, 7946, 7949, 7951, 7952, 7954, and 7957. Runs are:

7132001 7132005 7132006 7132007 7132008 7132009 7132010 7132018 7132023
7132025 7132026 7132027 7132028 7132057 7132059 7132061 7132062 7132066
7132068 7132071 7132072 7133004 7133008 7133011 7133012 7133016 7133018
7133019 7133022 7133025 7133026 7133035 7133036 7133037 7133039 7133041
7133043 7133044 7133045 7133046 7133047 7133049 7133050 7133052 7133054
7133064 7133065 7133066 7133068 7134001 7134005 7134006 7134007 7134008
7134009 7134010 7134013 7134014 7134015 7134016 7134026 7134027 7134028
7134030 7134043 7134046 7134047 7134048 7134049 7134052 7134055 7134056
7134057 7134065 7134066 7134067 7134068 7134069 7134072 7134073 7134074
7134075 7134076 7135003 7135004 7135005 7135016 7135018 7135019 7135022
7135023 7135024 7135025 7135028 7136017 7136022 7136023 7136024 7136027
7136031 7136033 7136034 7136035 7136039 7136040 7136041 7136042 7136045
7136073 7136075 7136076 7136079 7136080 7136084 7137012 7137013 7137035
7137036 7138001 7138002 7138003 7138004 7138008 7138009 7138010 7138011
7138012 7138017 7138029 7138032 7138034 7138043 7139018 7139019 7139025
7139031 7139032 7139033 7139034 7139035 7139036 7139037 7139040 7139043

7140007 7140008 7140009 7140010 7140011 7140015 7140016 7140017 7140018
7140022 7140023 7140024 7140042 7140045 7140046 7140051 7140052 7141010
7141011 7141015 7141016 7141034 7141038 7141039 7141042 7141043 7141044
7141047 7141064 7141066 7141069 7141070 7141071 7141074 7141075 7141076
7141077 7142001 7142005 7142006 7142016 7142017 7142018 7142022 7142024
7142025 7142028 7142029 7142033 7142034 7142035 7142036 7142045 7142046
7142047 7142048 7142049 7142059 7142060 7142061 7143001 7143004 7143005
7143006 7143007 7143008 7143011 7143012 7143013 7143014 7143025 7143031
7143054 7143055 7143056 7143057 7143060 7144011 7144014 7144015 7144018
7145007 7145008 7145009 7145010 7145012 7145013 7145017 7145018 7145019
7145022 7145023 7145024 7145025 7145026 7145030 7145044 7145057 7145064
7145067 7145068 7145069 7145070 7146001 7146004 7146005 7146006 7146007
7146016 7146017 7146019 7146020 7146024 7146025 7146066 7146067 7146068
7146069 7146075 7146076 7146077 7146078 7147052 7147055 7147083 7148020
7148024 7148027 7148028 7148032 7148036 7148037 7148054 7148057 7148059
7148063 7148064 7148065 7148066 7148067 7149003 7149004 7149005 7149018
7149019 7149023 7149026 7150007 7150008 7150013 7152035 7152037 7152049
7152051 7152062 7153001 7153002 7153008 7153014 7153015 7153021 7153025
7153032 7153035 7153103 7154004 7154005 7154044 7154047 7154051 7154068
7154069 7154070 7155009 7155010 7155011 7155013 7155016 7155018 7155019
7155022 7155023 7155026 7155042 7155043 7155044 7155045 7155047 7155048
7155052 7155053 7156006 7156010 7156017 7156018 7156019 7156024 7156025
7156026 7156027 7156028 7156033 7156040

Appendix B

Beam polarization

fill yellow blue

7847 0.4929 0.5083

7850 0.5708 0.5479

7851 0.5583 0.5917

7852 0.5557 0.5715

7853 0.5921 0.5544

7855 0.5783 0.6194

7856 0.5832 0.5806

7863 0.5890 0.6130

7864 0.5894 0.6211

7865 0.5687 0.5200

7871 0.6159 0.6260

7872 0.6248 0.5575

7883 0.5704 0.5497

7886 0.5801 0.5163

7887 0.6258 0.6014

7889 0.5639 0.6076

7890 0.6154 0.5699

7891 0.5864 0.6030

7892 0.6273 0.5958

7893 0.6233 0.6213
7896 0.6059 0.5847
7898 0.5686 0.5779
7901 0.5897 0.6184
7908 0.6011 0.5077
7909 0.5980 0.5300
7911 0.6101 0.5184
7913 0.6052 0.6044
7915 0.6081 0.5644
7916 0.6135 0.4892
7918 0.5825 0.5480
7921 0.5640 0.5779
7922 0.5755 0.5631
7926 0.5882 0.5777
7944 0.5565 0.5522
7946 0.6185 0.5323
7949 0.5488 0.4963
7951 0.5945 0.5509
7952 0.5603 0.5183
7954 0.5814 0.4989
7957 0.5997 0.5202

Appendix C

π^0 Software Development Details

C.0 π^0 decay in the endcap

Before we show people how we develop the software to reconstruct neutral pions in the EEMC region of STAR, we would like to present an overview of π^0 decay kinematics in the endcap region.

A neutral pion meson will generally decay into two photons. The two photons head into the endcap and deposit energies, which provides us shower profiles in the endcap. Different neutral pions with different energies and transverse momentum will result in different shower profiles. The energy of the photon pair will be mainly deposited in the EEMC towers. The opening angle $\phi_{\gamma\gamma}$ between the photon pair is determined from collected SMD information. Most decays from neutral pions are not symmetric. It is also important to know the energy sharing $Z_{\gamma\gamma}$ between the photon pair. The SMD provides information for not only the opening angle, but also the relative energy of two showers. So the idea is to collect information from SMD first, then reconstruct points and pion candidates. Once we get the energies of the two SMD showers and tower showers, we calculate each point candidate's energy based on tower energies after sharing with the SMD showers in a 3×3 tower patch. Then we decide the energy sharing $Z_{\gamma\gamma}$ by equation (C.8) and invariant mass of the neutral pion by equation (C.9).

Our software has been developed accordingly upon the above kinematics, please refer to the following sections for details.

C.1 π^0 finder software introduction

We have made much effort in developing the analysis software required to do the inclusive π^0 measurement in the EEMC since 2005. Our software package was developed upon the framework of Jason Webb, whose codes were stored into STAR CVS [95]. The importance of measuring the double spin asymmetry from high-pT inclusive π^0 production in longitudinal pp collisions has been explained in chapter 1. To achieve that goal, the reconstruction, efficient detection, and analysis of the high transverse momentum neutral pions in the EEMC is critical. This software includes four generic parts: the A2E-Maker, the Cluster-Maker, the Point-Maker and the Pion-Maker (Pi0-Mixer).

The A2E-Maker is basically a connector between the digitized data stored in the RCF system and the last three makers. This maker translates the ADC values from the MuDst data files into energies with the unit of GeV. People can also rescale the energy into MeV if necessary. During this translation, pedestal subtraction and gain corrections for each channel are applied to make the data more useful. In this maker, most of the fundamental energy information in the EEMC are defined and saved in different types of classes and vectors to be utilized in the later analysis.

Based on the specific EEMC structure of towers, pre-shower, post-shower and SMD detectors, a special Cluster-Maker has been developed. In the reconstruction of neutral

pions in the EEMC, the nature of EM shower of the EEMC dictates that we can obtain transverse shower information in almost all layers of the EEMC. It is not necessary to use all the deposition information all the time for the analysis. Most of the showers of interest are distributed in small regions within adjacent towers, pre-shower layers, post-shower layers and SMD strips. So the Cluster-Maker has been developed to produce clusters of EEMC towers, pre-shower and post-shower layers, and SMD strips. This is especially important for the SMD clusters, because we use the SMD position information to locate photon pairs provided in neutral pion decay.

We reconstruct photon point candidates from the SMD clusters found in the preceding. 98.8% of the neutral pions will decay into two photons [92]. To reconstruct a π^0 , we have to find all possible photon points from the EEMC first. A point candidate is defined as the overlap of two SMD clusters, one from the U plane and another from the V plane, with the intersection underneath an active tower. We use the SMD clusters to achieve position information for the photons. The energy of the photon is obtained by summing over a 3×3 tower patch centered on the active tower, as long as the active tower is not at the edge of the EEMC. A set of criteria is used to qualify point candidates, and we will discuss details later.

All qualified points are used in the Pion-Maker to reconstruct π^0 candidates. To reconstruct a π^0 , we need to know the total energy, the energy sharing ratio of the two photons, and the opening angle between the two photons. In this class, some basic information about the reconstructed π^0 such as pT, energy, invariant mass, spin dependent yields, and so on, are saved into histograms and trees for further analyses, such as for efficiency studies and double spin asymmetry calculations.

C.2 The A2E-Maker

The A2E-Maker is a class designed according to the EEMC structure. There is an important difference in the definition of ‘tower’ in the code. We define four layers of towers in the code because of the geometry of the EEMC. The mechanical tower corresponds to layer 0 of the programming ‘tower’; the pre-shower 1 detector corresponds to layer 1 of the ‘tower’; the pre-shower 2 detector corresponds to layer 2 of the ‘tower’; and the post-shower corresponds to layer 3 of the ‘tower’. So there are 720 tower indices for each layer of towers. Based on the geometry of the EEMC, concepts like sector, subsector, etabin and phibin are also introduced in the program. So we have 12 sectors with a range of [0,11], 5 subsectors in each sector which are denoted by ‘A’, ‘B’, ‘C’, ‘D’ and ‘E’, 12 etabins in each subsector with a range of [0,11] from inner ring ($\eta \approx 2$) to outer ring ($\eta \approx 1.1$), and 60 phibins from [0, 59]. The phibin is calculated from the sector and subsector information: $\text{phibin} = 5 \times \text{index of sector} + \text{index of subsector}$. For the two SMD planes, the U plane is denoted by 0 and the V plane is denoted by 1. The SMD is also sector dependent. For each SMD plane, it is also denoted from which sector. And for each plane from a certain sector, there are 288 strips, so that the index is numbered from [0, 287]. The layer parameter is also applied to the two SMD planes, with 4 denoting U plane and 5 denoting V plane for consistency in the algorithm.

We process the ADC values for each channel described in the above paragraph of the EEMC geometry, use information stored in the STAR database, and develop them in several stages: raw ADC information, pedestal-subtracted ADC, and transformed energy

information after gain correction. The energy information is then utilized in the following analyses, and is used to calculate the transverse energy and momentum information. The technical details of how the framework was set up are presented in this section.

C.2.1 Basic classes

Before building up the main A2E-Maker, three basic classes are set up to facilitate the software development. They are the element class, tower class, and strip class.

The element class is produced to provide the basic functionality for EEMC detectors. It is used to save the raw ADC, pedestal-subtracted ADC, and energy information. It also provides the status bits information (stat or fail) for each channel in general. There is also a pointer to represent the hit information from the database.

The tower class inherits the element class with specific formatting to save all the information from the four layers of towers in the EEMC algorithm. For example, tower index, layer, sector, subsector, phibin, etabin, raw ADC, pedestal-subtracted ADC, neighboring towers, energy, status bits, and transverse energy information are all stored; furthermore, it provides a type vector definition of the EEMC tower so that we can use this type of vector in future analysis if necessary.

Just like the tower class, the strip class inherits from the element class. But it is used for the SMD. It provides the SMD sector, plane and index information. It also defines a vector type of the SMD strip, which is further used in the Cluster-Maker and plays an important role.

C.2.2 The main maker

All the information from all layers of towers and all SMD strips from all sectors are cleared and initialized at the beginning to avoid storage and memory leakage problems. A switch is set up to allow us to analyze MuDst data (switch 1) or StEvent data (switch 2) from the STAR database. In the neutral pion analysis, we always use switch 1 to analyze MuDst data.

Once we access the data, we collect the hit information around the whole EEMC. We loop over all hits one by one to grab information. For the tower layers, we can get the raw ADC, sector, subsector, and etabin information directly according to hits. For the SMD, we can get the raw ADC, strip, and sector information according to hits and planes. Two main and similar functions, addTowerHit and addSmdHit, are then developed to do the more complicated job of converting from ADC to energy.

The addTowerHit function was developed to facilitate towers analysis. We first make sure that the information of sector, subsector, etabin and layer from every hit is in the correct region of the EEMC, as described earlier. The tower index is then defined as:

$$\text{index}=60 \times \text{sec}+12 \times \text{subsector}+\text{etabin} \quad (\text{c.1})$$

and we make sure the index is within 720 for each layer of towers. Based on the sector, subsector, etabin and layer information, and ignoring dead towers, we can retrieve the pedestal and gain information for that tower from the STAR database. The pedestal and gain information have already been inserted in the database for common usage. To avoid small fluctuations in the pedestal, all ADC values are required to exceed their corresponding pedestal values by 3σ (σ =pedestal width). After that, the raw ADC and

pedestal-subtracted ADC are both saved for this tower. After making sure that the gain is positive, we calculate the energy as:

$$\text{energy} = (\text{adc} - \text{ped} + 0.5) / \text{gain} \quad (\text{c.2})$$

The energy is then saved as this tower's energy for further analysis. One can also rescale the energy by a certain factor before saving it. The default value for the factor is 1.

The next step in the main maker is to calculate the transverse energy of this tower. We locate the center of this tower using the sector, subsector and etabin information, and define a three-vector based on the EEMC geometry from the center of the STAR detector. After normalization of the three-vector to unity, the normalized vector is multiplied by the tower energy. The transverse component is then saved as the tower's transverse energy. This tower is then identified as a hit tower, and the energy is added to the layer's energy. By looping over all hits at all layers of towers, we can translate all ADC values passing the 3σ pedestal cut to the tower's energies, and save all the information described above.

The addSmdHit function was developed using the same logic as the addTowerHit function, but with a different detector structure to facilitate the SMD study. Dead strips are thrown away, and raw ADC values less than 3σ above the corresponding pedestal values are disqualified. Those surviving strips are recorded as hitstrips, and the energy calculation is performed using the same formula (c.2) as for towers.

The addTowerHit function is also used to collect the information from pre-shower and post-shower detectors. As discussed above, these are considered just different layers of 'towers' in the algorithm. But in later discussions, we will use the detector names "pre-

/post-shower”, or “towers” to be consistent with the mechanical design, for convenience of communication and to make the physics interpretation clearer.

C.3 The Cluster-Maker

This class inherits the geometry definition, utilizes the energy and energy-related information stored in the A2E-Maker, and produces clusters of the EEMC towers, pre-shower and post-shower, and SMD strips.

If a π^0 meson produced in a pp collision heads toward the EEMC, it will decay into two photons about 99% of the time. Ideally, the two photons will produce two showers in the EEMC, and this means we can measure two tower clusters for energy collections, and typically four SMD clusters (two from each plane) to find the positions of the photon pair. But in the real world, it is not so simple to reconstruct tower and SMD clusters because of the complexity of the energy distributions. To reconstruct tower clusters, we basically start searching from seed towers which pass a cut threshold in order of descending energy, construct and loop over a maximum of eight adjacent towers around a seed tower, then conclude one tower cluster. To reconstruct a SMD cluster from either plane, we loop over all the 288 strips in the plane first, and find those seed strips which pass the threshold cut. Starting from the most energetic one through descending order, three adjacent strips on each side are added to the seed strip to form a SMD cluster. If a strip is already used to form a SMD cluster, it is deleted from the seed pool, which means the minimum separation between two SMD clusters in the same plane is four strips. This is very important to keep in mind in some of the following discussions.

The SMD cluster reconstruction is critical in our neutral pion analysis, because we use the SMD cluster information to decide the positions and energy sharing of the photon pair, which are both directly used in the invariant mass calculation of the neutral pion. A great amount of time and thought has been put into developing this part of the analysis.

C.3.1 Basic classes

Two basic classes, the tower cluster class and the SMD cluster class, are developed before we go into the main maker.

The tower cluster class describes clusters of the EEMC towers, pre-shower and post-showers elements. Some fundamental features are described in this class, such as the energy, seedEnergy and numberOfTowers of the tower class. We sort the tower seeds by energy, and reconstruct tower clusters from the most energetic ones. In this basic class, we do not confine the size of a tower class in general. For the neutral pion analysis, the maximum size is nine with one seed plus eight neighbors. But sometimes we do not have eight neighboring towers, or some neighboring towers are dead or inactive, so we can have fewer towers. If you want to know the size of a tower class, you can call the function numberOfTowers to get the information. The energy of a tower cluster is defined as the sum of the energies for all towers in the cluster. A unique parameter key is applied to a tower cluster, which can be used as the unique identification of the cluster. In our algorithm, the seedEnergy is defined as the energy of the first tower during the reconstruction of this cluster. A special function ‘add’ was developed in this class to realize and summarize the following information: EEMC towers with the type definition of the tower class in the A2E-Maker, weights of towers, and total energy of the tower

cluster. The default weight value for each tower is set to 1. Finally, a vector type definition of `StEEmcClusterVec_t` is provided for later analyses in the algorithm.

The SMD cluster class describes the cluster of SMD strips. This class contains more information than the tower-cluster one, because it is very important in our analysis. The basic features in this class include energy, cluster mean, `numberOfStrips`, unique key, and sigma (width) of the SMD cluster. We sort all SMD seed strips by energy, and start from the most energetic one to reconstruct SMD clusters. The seed strip is added into a SMD cluster at first, and is denoted with index 0 in the strip vector. Then we add three more strips from each side of the seed to the cluster, and denote them from 1 to 6 correspondingly. When a SMD cluster is generated, a unique key number is assigned to this cluster as identification for further usage. The size of a SMD cluster is defined as the number of strips in the cluster. A special ‘add’ function is also developed in this class to realize and summarize the following information: SMD strips with the type definition of the strip class in the A2E-Maker, weights of SMD strips, total energy of the SMD cluster, the mean value of the cluster, and the standard deviation sigma of the strip distribution. The default weight value for each strip is set to 1. The SMD strip information of the cluster is stored and can be called back when necessary in future analyses. The energy of the SMD cluster is the sum of all strip energies in this cluster. To calculate the mean, we use this formula:

$$Mean = \frac{\sum_{i=1}^n [(S_i + 0.5) \times E_i]}{E_{tot}} \quad (c.3)$$

where i denotes the number of strips; S denotes the strip index; E_i is the energy of the i th strip; and E_{tot} is the total energy of the cluster. Note that 0.5 is added to the strip index

term to allow float values in the formula and approach the middle of a strip. Furthermore, we calculate the standard deviation of the cluster distribution by the following formulae:

$$Sigma = \sqrt{\frac{\sum_{i=1}^n [(S_i+0.5)^2 \times E_i]}{E_{tot}} - Mean^2} \quad (c.4)$$

Here we use the square root of the mean of the square, minus the square of the mean, to calculate the standard deviation.

Four special features: ‘inner’, ‘outer’, ‘Energy’, and ‘Key’ are also set in the SMD cluster code for sorting purposes. They are all bool logic type of parameters. When we sort the SMD cluster pool, if the mean value defined from formula (c.3) of cluster one is smaller than the mean value of cluster two from the same plane, ‘inner’ is set to be true, which means cluster one is at the inner side; otherwise, false. If the mean value defined from formula (c.3) of cluster one is larger than the mean value of cluster two from the same plane, ‘outer’ is set to be true, which means cluster one is at the outer side; otherwise, false. If the energy of cluster one is larger than the energy of cluster two from the same plane, ‘Energy’ is set to be true, which means we sort them by descending energy order; otherwise, false. If the key value of cluster one is smaller than the key value of cluster two from the same plane, ‘Key’ is set to be true, which means we sort them by ascending unique key order; otherwise, false.

Just as for tower clusters, the type definition of vector `StEEmcSmdClusterVec_t` is also provided in the SMD cluster code for future analyses.

C.3.2 The main maker

To set up the maker, all the information from all layers of towers, pre-shower, post-shower, and SMD planes are cleared and initialized. The maximum extension for a SMD

cluster is set to 3 by default and is applied in all our analyses, which means we add exactly three more strips from both sides around a SMD seed strip to form a cluster. A minimum size request for SMD clusters is also set to 3. In the Make part of this main class, two important functions are developed to realize the goal of our Cluster-Maker. One is the buildTowerClusters, another is the buildSmdClusters.

The buildTowerClusters function is developed to reconstruct the four layers of the tower (including pre-/post-shower) clusters. We first loop over the four layers of towers and initialize the weight values for each incoming event. The seed threshold values for towers, pre-shower and post-shower are set to 0.8, 0.001, 0.001 GeV respectively based on simulation results. To find all possible seed towers which exceed the threshold, we sort all towers in each layer by energy in descending order. The first tower which is below the threshold will be marked as 'last'. We use this 'last' tower as our flag in the tower cluster iteration procedure. If a tower's energy is larger than the 'last' one, its neighbor towers will be weighted by the seed tower energy, and these neighboring towers can not be used as seed towers anymore. Only those towers before the 'last' tower and without weights are pushed into the tower seed pool. We then start to reconstruct tower clusters from this pool. Once we find a seed tower, we add the seed as the first tower in the cluster. Because the neighboring towers could be counted more than one time (that is, in more than one cluster), we do energy sharing before adding these neighboring towers into the tower cluster. Here we can see it is important to weight neighboring towers by the seed energy every time. We rescale the energy in these neighboring towers by the fraction between the seed energy and the weight. After rescaling the energy of these neighboring towers, we can now add them into the same tower cluster. A unique tower

key will be assigned to this new tower cluster. And the tower cluster is then stored in a tower cluster pool for further usage. By iterating all seed towers from the seed pool, we can find all tower clusters from all layers.

The second function, `buildSmdClusters`, is the most important part in our neutral pion reconstruction because it is used to define the SMD clusters which give the core information for later makers. We reconstruct SMD clusters sector by sector, plane by plane. First, we recall all the hit strips from the strip vector of the A2E-Maker and sort them by energy, so that these strips are ordered by descending energy. The next step is to find all possible seed strips from these hit strips. To qualify as a seed, a SMD strip has to pass the following requirements: First, the strip index must be within the range [3,283]. We do not allow a strip at the very edge of the SMD plane to be a seed. Second, this strip has to be a good strip. Dead strips, as identified by status bits, are disqualified. Third, the energy of the strip should be larger than the threshold value of 1.5 MeV. This threshold cut is at about the MIP value (1.4 MeV, according to simulation study). But this value can be changed to 2 or 3 MeV easily if desired. Fourth, the energy of the strip from every seed, except the first one, is also required to pass a special “floor” setting designed from the SMD structure. The floor setting is set to 0 for the first seed. The special floor setting is shown below in figure 3.1.

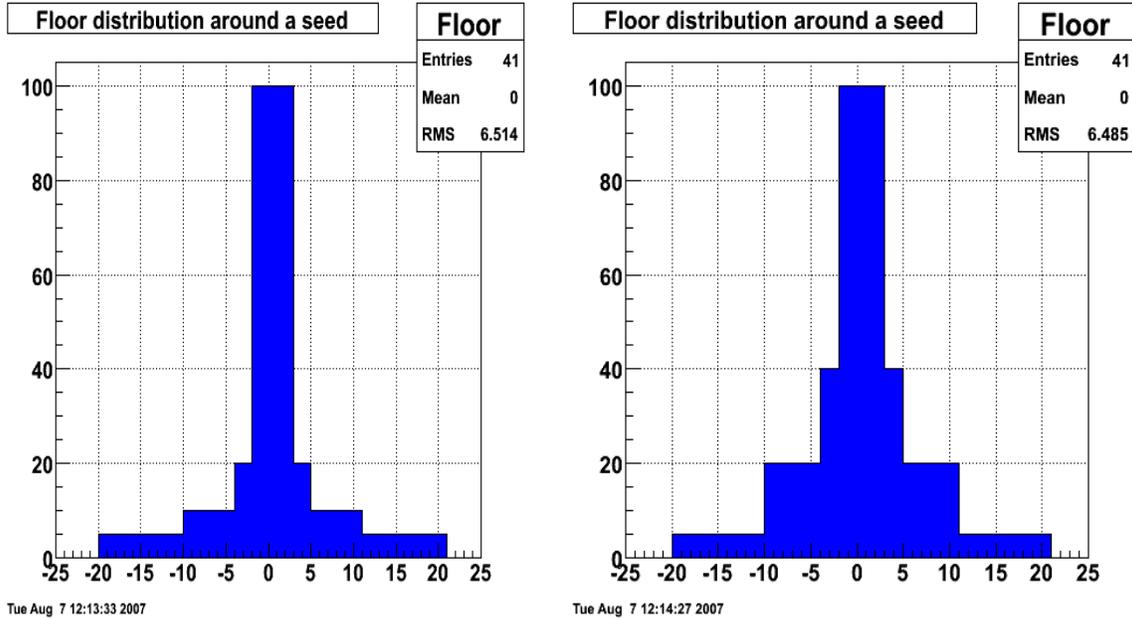


Figure c.1: The floor settings of SMD to find seed strips.

The floor setting basically suppresses the appearance of a second seed around an existing seed. For simulation studies, we used the distribution of energies shown in the left plot on figure c.1 as the general floor shape for all sectors of the EEMC. For analysis of real data, we used the left plot for the first (front) SMD plane and the right plot for the second (behind) SMD plane in all sectors. The reason we chose different settings in the real data analysis will be explained in ore detail in Chapter 5. For now, we will describe the general “floor” algorithm. In figure c.1, channel 0 means that we have found a seed there; and the floor value within two strips from this seed is set to the energy of the seed. For strips that are three or four strips away from this seed, the floor value is set to 20% or 40% (simulation of data) of the seed energy. Strips that are 5-10 strips away from this seed have a floor value set to 10% or 20% (simulation of data) of the seed energy. And for strips 11-20 strips away from this seed, the floor value is set to 5% or 10%

(simulation of data) of the seed energy. By summarizing the third and fourth requirements, we can get the energy requirement for a strip to qualify as a seed:

$$\text{Energy of seed} < (\text{mSeedFloor} * \text{floor}[\text{index}] + \text{mSeedEnergy}[\text{plane}]) \quad (\text{c.5})$$

where mSeedFloor is a parameter defaulted at 1, and floor[index] is the floor value, initially set at 0, but set up according to figure c.1 after finding the first seed. mSeedEnergy is the 1.5 MeV threshold cut above. After the strip passes all the above four requirements, it will be pushed into the SMD seed pool. We can get all possible SMD seeds by this method.

Now we iterate all seeds from the pool, and start from the most energetic one to reconstruct our SMD clusters. Starting from a seed, this strip will be added into a SMD cluster at first. Then the adjacent three strips, if not dead, from both sides from the same plane in the same sector will be also added to the same cluster. All these seven strips are then marked, and if other seeds are found among these marked strips, they cannot be used as seeds to reconstruct SMD clusters anymore. But they can still be used as part of a cluster for another nearby seed. This creates a problem of overcounting strip energies during the reconstruction of SMD clusters. We will fix the problem later. If strips from the adjacent six are dead or failed, their energies will be set to be zero. A minimum requirement of three active strips is required to save the SMD cluster. This code is designed in a flexible way so that users can reset the cluster size from seven to any other number they want by changing the maximum-extension parameter from three on both sides to other values. There is also another option 'Suppress', to suppress more strips on both sides of a seed to avoid another seed found in the suppressed region. The minimum number of active strips can also be reset to other values, if they user feel it is necessary

for their analysis. Before we fix the above overcounting-strip problem (see below), these SMD clusters are stored in a temporary SMD cluster type of vector.

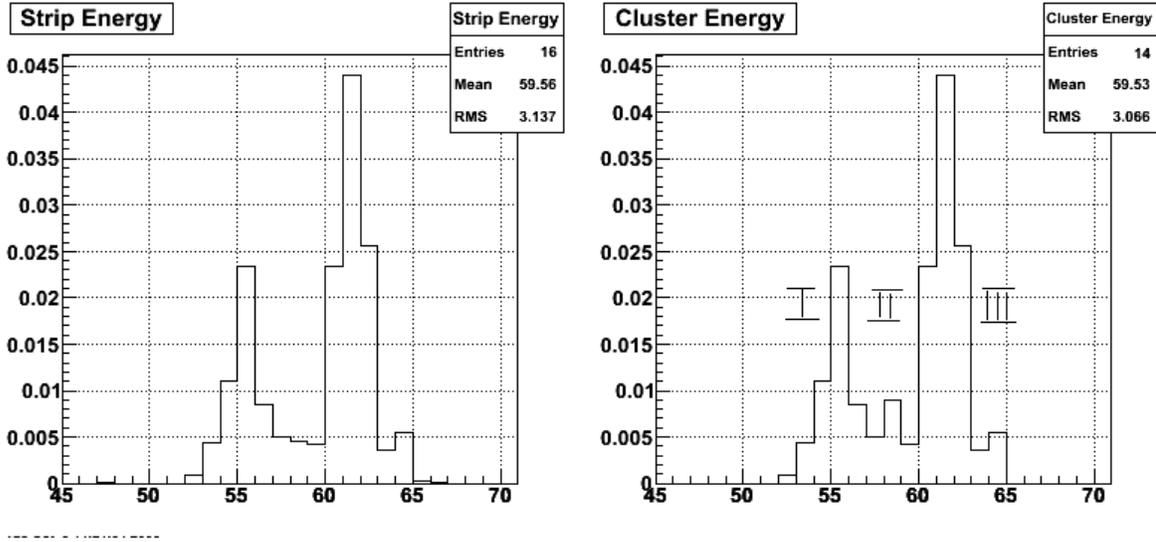


Figure c.2: Double-counted strip example from simulation sample with flat energy [5,60]GeV and flat EEMC distribution.

Figure c.2 shows an example of the double-counted strip problem. The left plot in figure c.2 shows the raw SMD strip energy deposition information from an event from a simulation sample with a flat energy of [5,60] GeV and flat EEMC distribution. The x-axis denotes the strip index, and the y-axis denotes the energy deposition in GeV. Two SMD clusters are found based on the left plot. We then reconstruct the energy deposition information from these two SMD clusters, as shown in the right plot on figure c.2. As we can see, strip channel 58 is counted twice, so that it produces a bump when we consider the two clusters. Figure c.2 gives only a simple example, and we can find many more serious problems, when all three strips in region II are counted twice. A direct result from this problem is that we would measure a smaller opening angle between the photon pair, and energy sharing between the photons becomes confusing and inaccurate.

To fix this problem, we divide the SMD in the right plot on figure c.2 into three regions: I, II and III, based on the two seed strips. We sum over strips in regions I, II, III and get the total energy E_I , E_{II} and E_{III} in each region. We re-assign energy values to those overlapped strips in region II according to these formulae:

$$E1 = E_{II} \times \frac{E_I}{E_I + E_{III}} \quad (c.6)$$

$$E2 = E_{II} \times \frac{E_{III}}{E_I + E_{III}} \quad (c.7)$$

Note that E_{II} is flexible here; it means the energy of the double counted strips in region II. If there is only one overlapped strip, it is easy to treat; if there are more than one overlapped strips, we treat them one by one.

The result of this improvement in cluster definition is shown in figure c.3. Based on the Monte-Carlo sample with flat energy and flat EEMC distributions mentioned previously, we show the invariant mass spectrum comparison in figure c.3. The blue curve shows the invariant mass without fixing the over-counting problem, while the red curve represents the spectrum after fixing the problem. We can see from figure c.3 that the peak width is sharper after this improvement, changing from $\sigma=0.035$ to 0.031 . The peak position is closer to the theoretical value of 0.135 GeV, although we use a correction factor of 1.3 (more details later) here. We point out these two benefits here, because we have also achieved smaller peak widths and more stable peak positions as a function of transverse momentum in the real data analysis.

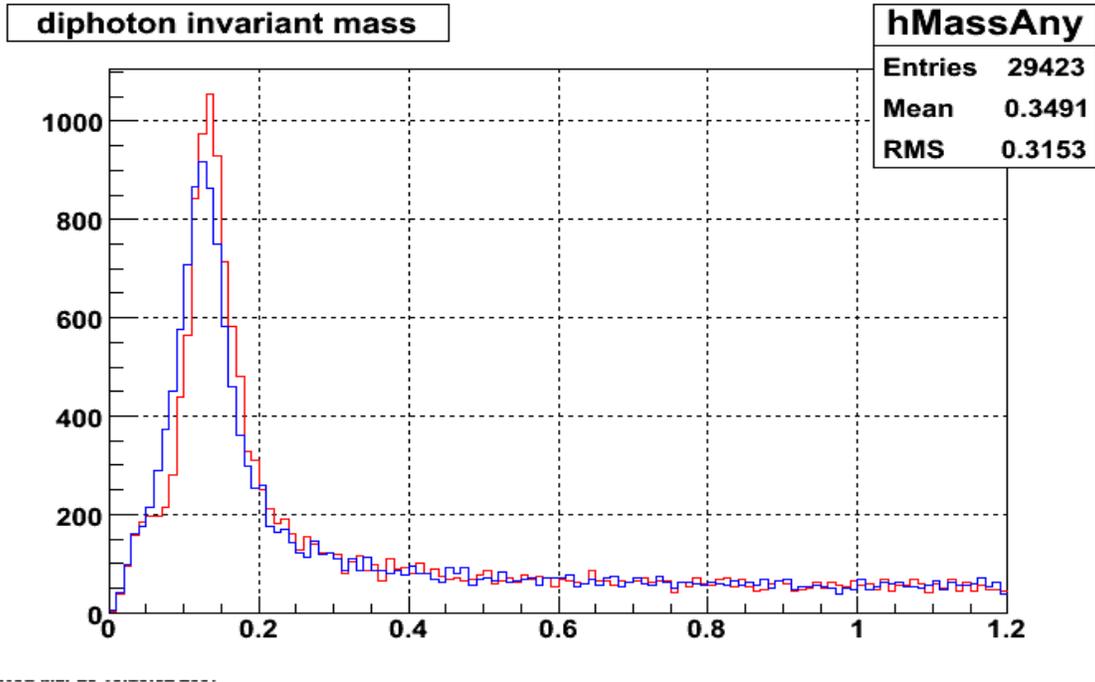


Figure c.3: the invariant mass spectrum comparison when fixing the double-counted strip problem from a MC sample with flat energy [5,60]Gev and flat EEMC distribution.

After the overlapped strips are divided into two new strip energies, each new strip will be added into its original cluster. In the ‘add’ function, the energy of each new cluster is re-calculated, and so is the mean value and standard deviation sigma. We now assign a unique key to the SMD cluster, and go to the final step to store them in the storage SMD cluster vector for later maker usage.

C.4 The Point-Maker

The EEMC Point-Maker produces photon point candidates from SMD clusters and related tower clusters. SMD clusters are used to determine the position and energy sharing of points, and related tower clusters are used to determine the absolute energy of points.

The SMD clusters are inherited from the EEMC Cluster-Maker described previously. Because the SMD U plane and V plane are orthogonally oriented, an incident photon is typically going to give an intersection between the two planes. In the Point-Maker, a point is defined as the coincidence of two SMD clusters from U and V planes individually, underneath an active tower. We first find all possible point candidates from the SMD clusters, and then determine the energy of each point based on tower energy information.

We reconstruct points sector by sector. Starting from the first sector, we loop over all SMD cluster for the two planes, from inner to outer radius, as defined in the SMD cluster class, to build all U-V SMD cluster combinations associated with an active tower. If an isolated U-V pair with an active tower is found, this pair will be identified as a point. An isolated U-V pair means both clusters have only one intersection with the other plane. We sort all combination pairs by the relative energy ratio between the U and V planes. If we find that two or more clusters from one plane are sharing a single cluster from the other plane, we use a splitting algorithm to split the single cluster energy, and do the energy matching with the two or more clusters. In this case we will eventually choose the points with the best energy matching to the point pool. In a third case, if no isolated pair or splitting pair are found, we form multiple combination pairs with the same clusters. We then sort them by the relative energy ratio, and choose the points with the best energy matching between the U and V clusters. No matter which case is considered, after we push an identified point into our point pool, we remove the SMD clusters related to that point from the cluster pool, and loop over all remaining SMD clusters again to find the

next point. After all points in the sector are identified, we go to next sector, and continue, until we find all points in the EEMC.

After all points are pushed into the point pool, we determine the energy of each point based on the SMD cluster energy and related towers energies. For each point, we find which tower it is under. This tower is called tower 0 for this point. Tower 0, plus a maximum of eight neighboring towers, provides information to give energy to the point. If the tower patch contains only one point, the energy of the point is just the sum of the energies of all towers in this patch. If the tower patch contains several points, the energies of these points are then shared from the sum, according to the SMD energies from these points. If several towers in this tower patch are shared by other points, this point will share the energy of these overlapped towers with other points according to the SMD energy ratios.

C.4.1 Basic classes

To reconstruct points from the SMD cluster and tower information as described above, we need the fundamental information from basic classes of the EEMC tower cluster and EEMC Smd Cluster. These two classes have been described in the Cluster-Maker section. A new basic class, the EEMC point class, is developed before we can use the main Point-Maker.

The EEMC point class describes the EEMC points reconstructed from the SMD cluster and tower information. Some fundamental features of points are described in this class, such as their position, energy, tower, SMD cluster, number of relatives which share tower energy, and sector. A kind of type definition about the EEMC point vector is also

provided for further analyses. The position of a point in the code is a three-vector. The energy of a point will be saved in GeV as a fundamental element. Towers will be added to a point with specific weight (default=1). The tower above the SMD point is called tower 0, and the neighboring towers are denoted from 1 to a maximum of 8 depending on where a point is in the EEMC. For each point, it has two SMD clusters: the U cluster and V cluster. For further usage, we define cluster 0 as U and cluster 1 as V. ‘numberOfRelatives’ denotes the number of other points which share the tower energy with this point. The sector information for a point is also saved.

A special boolean function ‘chiSquare’ is also defined in the EEMC point class. When we sort all combination pairs to find possible points, we sort them by the relative energy ratio between the two SMD clusters in the pair. ‘chiSquare’ is used to realize this function. Suppose we compare two points p1 and p2. p1 is associated with two clusters u1 and v1, and p2 has two clusters u2 and v2. The energy values are Eu1, Ev1, Eu2 and Ev2, respectively. Each point is assigned a ‘chiSquare’ by the following calculation:

$$chiSquare\ 1 = \left| \frac{Eu1 - Ev1}{Eu1 + Ev1} \right| \quad (c.8)$$

$$chiSquare\ 2 = \left| \frac{Eu2 - Ev2}{Eu2 + Ev2} \right| \quad (c.9)$$

If $chiSquare\ 1 < chiSquare\ 2$, the function ‘chiSquare’ will return true. This means point 1 has better energy matching between the U and V plane clusters than point 2. This criterion is assigned in the Point-Maker when we try to build all possible points. For multiple points, the code will sort them by the ‘chiSquare’ to decide their array order. Points with better energy matching will be taken for further criteria checks to reach the final point pool. Note that the ‘chiSquare’ function is comparing the relative energy ratio

calculated between the two SMD clusters from the same point, instead of the energy ratio from two points.

C.4.2 The main maker

The class is initialized and cleared before an event comes in for analysis. An active tower is defined as a tower with positive energy. A tower threshold default value is set to zero. Users can change this parameter easily; the energy mode is set to 0 so that we determine the energy of points according to the towers energy in proportion to the SMD energies.

To allow this maker to work most efficiently, we find all possible points sector by sector. Starting from the SMD clusters of the U and V planes for each sector made by the Cluster-Maker, we sort these SMD clusters from inner to outer radius. Then the first step is to build the SMD points. All pairs of U and V clusters in each sector with an intersection that is underneath an active tower are built as the first step SMD points, and are stored in a list with sector dependence. The function ‘buildSmdPoints’ is developed to realize the work. Then the algorithm will find points one by one using another special function ‘findPoints’. Every time the findPoints function finds a point and stores it into a point pool, we will remove the SMD clusters related to that point, and then call the buildSmdPoints function to rebuild all combination pairs again and to find the next point. When we have finished finding all points and have stored them in the same point pool, point energies are initially set to be the energy deposited in the SMD clusters (U+V). Based on simulation studies, we apply a sampling factor of 7% per SMD plane to these SMD points, and overwrite these energies. Then we calculate the final energy of these

photon points according to the SMD cluster energies and related tower energies. A special function ‘shareEnergySmd’ is designed to realize this work. We describe the details of codes for these functions ‘buildSmdPoints’, ‘findPoints’, and ‘shareEnergySmd’ in the following paragraphs.

The buildSmdPoints function builds all combination pairs if we assign it three variables: the EEMC sector, Smd clusters U vector and V vector. We loop over all U SMD clusters at the first chain and loop over all V SMD clusters within the first looping. The intersection of a pair of U and V clusters can be obtained by knowing the mean values of the U and V clusters and the sector from the SMD geometry. If the intersection is not within the EEMC geometry, we skip this U-V pair. If the intersection is within the EEMC and underneath a tower, we check the sector information of the tower to see if it is consistent with the SMD clusters sector or not. Only towers that pass the check go to the next step. We then check the energy of the tower, and if the energy is larger than zero, we call it an active tower and identify this U-V pair as a good combination pair. During this procedure, we also set up a framework to compare the energy matching between the U and V clusters. Users can set the match threshold as they want, but in this analysis we did not use this framework. If a good pair is identified, we save the following information from this pair: the U and V clusters information separately, the energy of this pair (U energy + V energy), the tower information of this pair, and the three-vector of the point position information. Finally, the pair will be pushed into a list of the SMD-only points. We find our final points from lists of the SMD-only points one by one.

The findPoints function is designed to find all qualified point candidates from the above lists of SMD-only points and return them as ‘points’. This function requires four

variables: sector, U cluster, V cluster and points. The first three are used in the function for analysis, and the last variable is used to return results. A temporary vector is created to store point candidates. We sort all U clusters and V clusters from the closest to the beam to the furthest according to the ‘inner’ setting discussed in previous sections. Then we call the above function `buildSmdPoints` to collect the SMD-only points to a list. If the size of the list is less than one, we skip it. Otherwise, we sort the list of SMD-only points by the function ‘`chiSquare`’ as described in the basic classes section. Two standard integer maps, `u2p` and `v2p`, are then created to store the associative arrays matching SMD clusters to SMD-only points. We start scanning the SMD U and V clusters at the same time, and find the one at the inner side of the SMD, which means it has a smaller mean value of strip number. A reference is given to the U or V cluster at the inner side. According to the map we created, we can find how many points are matched to this cluster by simply calling the size of the map for this cluster. If the size of the map for this cluster is one, we say this is a unique SMD cluster which matches a SMD-only point. This point will be pushed back to the temporary storage point vector we defined at the beginning of this function. We loop over all SMD clusters and find all these unique cluster-related points at the first stage of the algorithm. But we do not save these temporary unique points to the final point pool yet, and we also do not remove the unique clusters from the cluster pools. This is because a unique cluster can only be related to a single SMD-only point, but a SMD-only point can be related to multiple unique clusters. If we save a first-found unique point to the pool and remove related clusters, we will not be able to find other normal points related to these removed clusters. Though rather technical (and difficult to explain), this is a significant improvement over previous

algorithm, and allow us to find more points to avoid efficiency loss in reconstruction. This logic leakage is illustrated in figure c.4. To fix this problem, we developed a ‘splitting algorithm’ to store unique points with the correct energy, unique key and position information, and remove related SMD clusters from the SMD cluster pools.

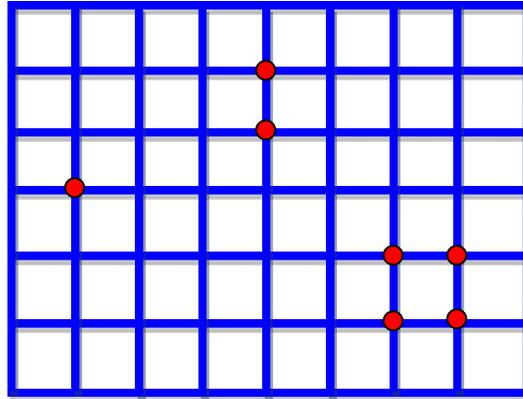


Figure c.4: SMD-only points distributed in the SMD U and V cluster network above.

Figure c.4 illustrates most of the energy deposition situations that arise from the decay photon pairs in the SMD. Each line in both the vertical and horizontal directions represents a SMD U or V cluster separately. The red point on the left represents an isolated point; the central two points represent the unique points we described above; and the right four points correspond to the two decayed photons in SMD, which is the normal case. The isolated point will be pushed into the point pool. The central part of two unique points will be treated by the splitting algorithm. As for the normal case indicated on the right, we will discuss this later.

The splitting algorithm was developed to treat unique points stored in the temporary vector. The basic feature for these points is that they share the same U or V cluster with each other. But when we save a point into the pool, we need to save the two sets of cluster information for this point. In this case, it would be incorrect to save the common

U or V cluster to two unique points. The splitting algorithm is designed to split the common cluster into two new clusters, with new cluster Id keys and energy information. So in practice, if the size of the temporary storage vector is not less than two, we start our splitting procedure; otherwise, the element in the storage vector is an isolated point, and it will eventually be pushed into the final point pool. Before we enter into the splitting, we sort these unique points by chiSquare as defined above. We then loop over all unique points to the mean value of strips for two clusters in both U and V planes. If two points are found to be sharing the same U or V cluster, the energies between the common shared cluster and other two clusters from the two points will be tested to see if they qualify to be split, using the following equation:

$$zratio = \left| \frac{E_{share} - E_{c1} - E_{c2}}{E_{share} + E_{c1} + E_{c2}} \right| \quad (c.10)$$

where E_{share} is the energy of the shared cluster, and E_{c1} and E_{c2} are the cluster energies from the other plane of the two points. If the zratio is not larger than 0.2, we split the shared cluster to create two new clusters; otherwise, we do not consider the two points sharing a cluster as likely to be two separate showers. The zratio cut of 0.2 was chosen based on the simulation study shown in figure c.5.

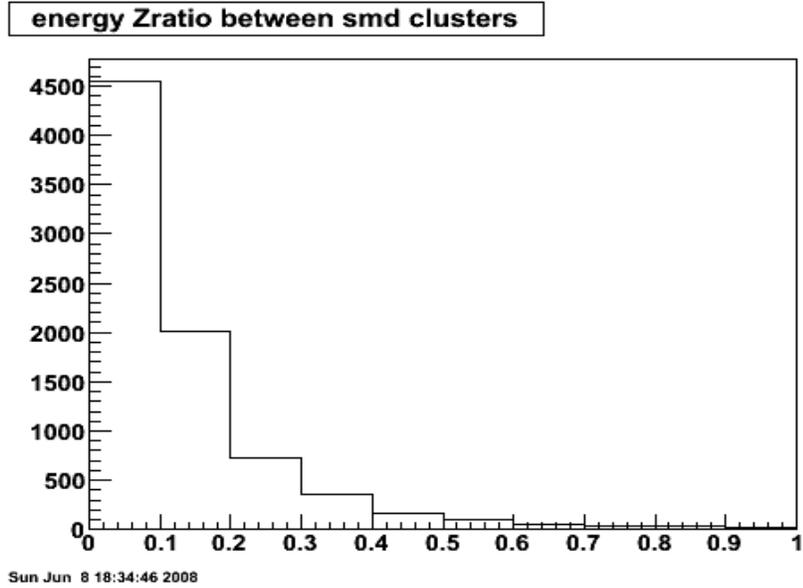


Figure c.5: Zratio distribution between the shared cluster and two clusters from the other plane with the flat energy [5,60] GeV MC sample.

By applying the 0.2 zratio cut, roughly 80% of the unique points qualify for the splitting algorithm. The key part of the splitting code is to split a single cluster in one SMD plane shared by two clusters in the other plane into two new clusters.

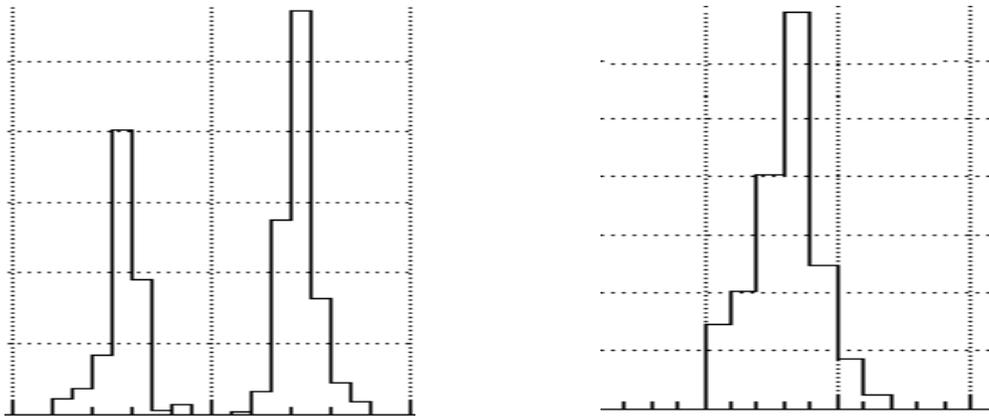


Figure c.6: The left plot shows two U clusters, which share the same V cluster shown in the right plot. V1' and V2' will be the two generated new clusters.

The two new generated clusters will have the same centroid and strip index numbers as the old one. But they will have new cluster Id keys and energies. The energies will be calculated by the following formulae:

$$E'_1 = E_{share} \times \frac{E_{c1}}{E_{c1}+E_{c2}} \quad (c.11)$$

$$E'_2 = E_{share} \times \frac{E_{c2}}{E_{c1}+E_{c2}} \quad (c.12)$$

where E'_1 and E'_2 are the energies of the two new clusters; E_{share} is the energy of the shared cluster; and E_{c1} and E_{c2} are the cluster energies from the other plane for the two points. After the splitting, a new point 1 with energy E'_1+E_{c1} and a second new point 2 with energy $E'_2 + E_{c2}$ will be pushed into the final point pool. Meanwhile, the corresponding clusters will also be saved for related points with new key number. Then the findPoints function will be called again in this chain to find other pairs.

After analyzing these isolated and unique points, we come to the last stage of the findPoints function, which treats the “normal case” in figure c.4. The routine is that we loop over the clusters remaining from the previous stages, and see if we can find clusters matching two or more points. These points will be temporarily saved in another vector. We loop over these points to find the point with the minimum chiSquare, which means this point has the best energy matching between its U and V cluster. We push this point into the final pool and remove the two clusters forming the point from the cluster pool, then call the function findPoints again to find more normal points.

The shareEnergySmd function is designed to reassign the point energy from SMD-based to tower-based. We loop over all points from the pool generated from findPoints above, and apply weights to tower 0 and the neighboring towers of each point. We use the SMD energy of each point as the weight. By doing this, if a tower contributes to only

one point's energy, it will be only weighted once; but if a tower contributes to multiple points' energies, it will be weighted multiple times. We then go to the second step, to decide the energy of every point. We loop over all points again, and in the chain of each point, we loop over the neighboring towers around tower 0. The energy of this point will be decided by:

$$E_p = \sum_{i=1}^n E_{Ti} \times \frac{E_{SMD}}{W_i} \quad (\text{c.13})$$

where E_p is the final energy of the point; i sums over tower 0 and all neighbors; E_{Ti} represents the i th tower energy; E_{SMD} represents the point SMD energy; and W_i is the total weight for the i th tower. After the calculation, we replace the energy of each point with the new E_p and save it in the same point pool to reconstruct π^0 . Energies in the pre-shower and post-shower layers are also decided by this method, though they are not used in this analysis.

C.5 The Pion-Maker

The Pion-Maker is a complicated object containing several classes with different functionalities. The Pion-Maker has been developed with three stages: reconstruct all possible π^0 candidates; filter these candidates by specific cuts and settings to store them into π^0 trees; and build a framework so that we can read the trees for further analysis.

We use the points made in the Point-Maker as the input to reconstruct our π^0 candidates. The points are stored in the final point pool vector defined in the Point class. We call these points, and form “gamma-gamma” pairs on a sector by sector basis for each event. For real data, we use the primary vertex position as the vertex information for both

points from the same event; for Monte-Carlo studies, the vertex is set to zero for all events by default, but users can change these settings. For each point, we use the energy and position information to calculate the 4-momentum of each point. When we consider a pair of points, we can then calculate the invariant mass, momentum, and other information for the π^0 candidate based on each point's basic information and the primary vertex. At this stage, all of these matched gamma-gamma pairs from the same sector provide us with the raw π^0 candidates by a mix-maker class.

Then these π^0 candidates go into the analysis class at the second stage. We filter these candidates by various cuts and settings, sort the survivors by different spin states, and store them in the π^0 tree for further analyses. To save these trees, another class named Mix-Event is used in this class. The Mix-Event class basically sets up all necessary information for the π^0 s before they can be stored to trees and chained, so this Mix-Event class is also used in the third stage. Furthermore, a number of histograms are saved in the resulting root files according to this analysis class, so that some simple analyses can be performed based directly on these histograms instead of the π^0 trees. This class is the main analysis tool for π^0 s, and more technical details will be discussed later.

The reconstructed π^0 s are stored in trees in the Mix-Event class. In our double-spin asymmetry analysis, the π^0 finder program does not do everything for us, but it is very general and prepares the results for future analyses. A Pi0-Reader class was developed to chain over the π^0 tree files, to be very flexible, and accommodate a variety of users. The Pi0-Reader class reads reconstructed pions and returns events in the type of Mix-Event class, where we save the information such as invariant mass, pT, eta, phi, zgg, energy,

spin-four states, and others. Based on this framework, we can normalize the spin-dependent π^0 yields according to relative luminosities in the asymmetry study, for example.

The Spin-Cuts, Spin-Histos, Mix-Event, and EEMC-Pair classes are defined as basic classes working as tools for main analysis. The main analysis tools include the Mix-Maker, Pi0-Analysis and Pi0-Reader classes. We will discuss the technical details in the following sections.

C.5.1 Basic classes

The first stage of the Pion-Maker is to reconstruct raw π^0 candidates using the Mix-Maker class. The basic classes utilized in the Mix-Maker class are the Point class and EEMC-Pair class. The Point class has been described in the Point-Maker section, and the EEMC-Pair class is designed to generate fundamental information for pairs of points, such as the invariant mass, energy, $z_{\gamma\gamma}$, $\phi_{\gamma\gamma}$, pT, pz, momentum and vertex. It also provides a kind of vector type definition of EEMC-Pair. Type definitions are always useful in the chain analysis. The EEMC-Pair class is called in the Mix-Maker class, given two points and point vertex information, which is needed as input to the main kinematics calculations of the Pair class. In the kinematics, the energy E of a π^0 is the summed energy $E_1 + E_2$ of the two points. $z_{\gamma\gamma}$ denotes the energy sharing of the two gamma points in a π^0 :

$$z_{\gamma\gamma} = \left| \frac{E_1 - E_2}{E_1 + E_2} \right| \quad (\text{c.14})$$

Three-vector momenta are calculated for the two points based on the position and vertex information for each point. Directions are determined according to

$$\overrightarrow{momentum}_i = \overrightarrow{position}_i - \overrightarrow{vertex}_i \quad (c.15)$$

for $i=1,2$. To obtain the two momenta, these unit vectors are scaled by the corresponding energies for the two points. The opening angle $\phi_{\gamma\gamma}$ is then calculated from the two momentum vectors. The vector sum of the two momenta gives us the total momentum of the neutral pion. The mean vertex is then calculated from the two points' vertices as (in our analysis, the two vertices are the same):

$$\overrightarrow{vertex} = \frac{E_1 \times \overrightarrow{vertex\ 1} + E_2 \times \overrightarrow{vertex\ 2}}{E_1 + E_2} \quad (c.16)$$

Finally, the invariant mass of neutral pion is calculated as:

$$mass = E \times \sqrt{1 - z_{\gamma\gamma}^2} \sin \frac{\phi_{\gamma\gamma}}{2} \quad (c.17)$$

The transverse and longitudinal momenta, p_T and p_z , are derived from the total momentum, and we can directly call them for further analyses.

At the second stage of the Pion-Maker, we filter the raw pion candidates and store them into π^0 trees by the Pi0-Analysis class. A number of histograms are also stored in root files instead of trees. The basic classes Spin-Cuts, Spin-Histos, and Mix-Event are designed to help the analysis. The Spin-Cuts class provides some fundamental thresholds for the π^0 analysis. The z-component of the vertex for reconstructed π^0 is constrained from -150 cm to 150 cm. The default cut for the energy sharing variable $z_{\gamma\gamma}$ is from 0 to 1, which includes all possibilities. The pseudo-rapidity detector η cut is set based on the EEMC geometry from 1.086 to 2.0. And to satisfy the filter, the transverse energy of at least one of the two points which are used to reconstruct a raw neutral pion has to exceed a cut at 3.0 GeV. This basic class only provides part of the cuts for the Pi0-Analysis maker, and more settings will be introduced in the main class later.

The Spin-Histos class is used to generate a series of histograms. It contains a basic framework called SpinHistos with two variables: name and title. A set of general histograms are defined in the class: hMass, hPT, hZgg, hZvert, hEta, hEEmcEta, hPhi, hRGeo, hYX[0], hYX[1], hYX[2], hE1E2, hPhiggVsEnergy, hEpi, hEsmd, hEpre1, hEpre2, hEpost, and hEpre12, which denote histogram information for the di-photon invariant mass, di-photon transverse momentum vs mass, di-photon energy sharing vs mass, event z-vertex vs mass, reconstructed η of π^0 candidate vs mass, detector η of π^0 candidate vs mass, detector ϕ of π^0 candidate vs mass, reconstructed π^0 geometry at polar coordinates in the EEMC, reconstructed π^0 geometry at Cartesian coordinates in the EEMC, point 1 geometry, point 2 geometry, point 1 energy vs point 2 energy, $\phi_{\gamma\gamma}$ vs π^0 energy, reconstructed π^0 energy vs mass, the ratio of SMD energy over π^0 energy vs π^0 energy, similar ratios from pre-shower 1, pre-shower 2 and post-shower vs π^0 energy, and pre-shower 1 energy vs pre-shower 2 energy, respectively. Whenever users call the Spin-Histos class in the Pi0-Analysis maker, this set of histograms will all be generated automatically. The two flexible variables allow users to define multiple histogram sets with this class type during the analysis, if users simply assign different name and title to the Spin-Histos class. For example, in a simple analysis, we can define a set of histograms for any reconstructed π^0 s, and another four sets of histograms according to the four spin states. These sets of histograms all have the same type of definitions described above, but with different constraints and titles when we set them. These histograms will not be stored in the π^0 trees, and are only saved in the root files generated after chains. The definitions of histograms in this class can therefore be modified as necessary without affecting tree results.

The Mix-Event class stores all the π^0 information into trees realized in the Pi0-Analysis class and also returns the information for further usage realized in the Pi0-Reader class. Four main functions are defined in this class: addPair, setEvent, setSpin4 and Clear. The first three functions are used to store related π^0 information, and the last function is a basic routine in the code to clear storage information between events to avoid overlapping and memory leakage problems. We are analyzing MuDst events in the experiment. When an event enters the Pi0-Analysis class, we use the setEvent function to record the following information from the event: event Id, event number, run Id, event information, run information, L0 trigger, trigger Id collection, magnetic field, bbc trigger, emc trigger detector, and primary vertex position. When a raw π^0 is ready to reside in the tree, we use the addPair function to store this particle with the following information: mass, transverse momentum pT, reconstructed pseudo-rapidity η , azimuthal angle ϕ , energy sharing $z_{\gamma\gamma}$, opening angle $\phi_{\gamma\gamma}$, energy, detector eta η' , pre-shower 1 energy, pre-shower 2 energy, post-shower energy, SMD U energy, SMD V energy, z vertex, tower0 index for point 1, tower0 energy for point 1, tower0 index for point2 and tower0 energy for point2. After chaining over all events, it will also give the total number of π^0 s added in the tree.

The setSpin4 function sets the spin state for the current event during runs. In the neutral pion analysis, there are basically four spin states of interests, depending on the RHIC beam polarizations: PP, PN, NP, and NN. Each spin state is indicated by the order of blue and yellow beams with positive or negative polarization. Digital numbers are applied to the four spin states, with 5 to PP, 6 to PN, 9 to NP, 10 to NN by STAR's

definition [93]. The π^0 information stored in trees can be recalled by the type of Mix-Event of classes or objects.

C.5.2 The Mix-Maker class

The Mix-Maker class takes input points found from the Point-Maker and loops over all pairs of points sector by sector to find all possible π^0 candidates. This class does not allow many cuts during the selection. We check the vertex information of each event, and ignore all events in which a vertex was not found. We check the trigger information to see if the event from a certain MuDst matches our requirement. The only other cut is that we require the two points constructing a π^0 to be from the same sector.

The class is initialized at the beginning. The default vertex is set to (-999, -999, -999) for both real data and simulation. But the main macro to run the code sets the vertex simulation samples to zero. Users should notice this to avoid confusion. Point-Maker and Mudst-Maker (not from the π^0 finder, but from STAR base) are also initialized to provide further information in this class. The first step to make the class is to check the trigger information and see if the event matches our trigger requirement. Only qualified events are allowed to reconstruct π^0 s. Then we loop over all points from the event generated from the Point-Maker. If points are found, we acquire the primary vertex information from this event and throw away those events without a valid found vertex. The vertex information is typically overwritten to zero for simulation data. Then we loop over points in the pool and compare all pairs of points sequentially. If both points are from the same sector, we assign the same vertex information identified above to the two points and call the EEMC-Pair class to reconstruct a π^0 with the kinematics described

above. The reconstructed π^0 is then pushed back to a EEMC-Pair vector to be analyzed in the next Pi0-Analysis class. We loop over all points to find and save all possible π^0 candidates to the same vector. This is convenient, because we can know the total number of raw candidates by calling the size of the vector.

C.5.3 The Pi0-Analysis class

This class takes a list of π^0 candidates found by the above Mix-Maker class as input, disqualifies many candidates by various cuts and settings, sorts events with qualified π^0 candidates with spin-dependency into a number of histograms, and stores π^0 information into trees in the type of Mix-Event class.

During the initialization, a series of Spin-Histos type of histograms are defined. They save the graphical information for all qualified unsorted π^0 s and for spin-sorted π^0 s. A more important function in this part is that a new T-Tree is defined with the class type of Mix-Event, which we call mRealEvent. π^0 s after selections will be added into this new tree for future use. Users of this code can also define their private histograms during the initialization to grab extra information.

The most critical steps happen in the make part of this class. When an event is running in the chain, the first task is to find the spin four state for this event. The spin value can be obtained from the database by the command getSpinState, using the beam crossing information at STAR. Once we get the spin information, we call the setEvent and setSpin4 functions described in the Mix-Event class to save related information to the mRealEvent. The tower information is then copied to the mRealEvent from this event. The trigger information will be checked again here to see if our trigger requirement is

among the trigger collection of this MuDst event. Failed events are skipped. If the event passes the trigger check, all layers of towers and SMD planes energy information is copied to the mRealEvent from this event. Then we filter all π^0 candidates by looping over them and matching various cuts and settings. Candidates are recalled from the Mix-Maker and a special function ‘accept’ has been developed to filter candidates. The ‘accept’ function is first required to pass a two-body-cut. The two-body-cut is based on the pair of points forming the π^0 . For each point, we can find a cluster of towers around tower0 which is the center tower of the point. Thus, a region with a maximum tower count of eighteen is formed by the pair of points. The current setting for the two-body-cut is that no more than four points can be found in this region. If so, this π^0 candidate will be disqualified. The next cut requires that both points have to exceed 1.5 GeV of transverse energy. We require the two tower0s of the pair to have positive ADC values. As we discussed in the Spin-Cuts class, the tower Et cut was set to 3 GeV, which means at least one of the two points has to pass the Et cut with their central towers. To be less sensitive to tower boundary effects found later in real data analysis, a more complicated setting for the Et cut has been designed, which replaces the constant 3 GeV cut with a bivariate Gaussian distribution peaked at 3 GeV along the pseudo-rapidity and azimuthal directions. The bivariate Gaussian distribution is set by the following formula:

$$Cut = 3.0 \times e^{-0.5 \times \left(\left(\frac{peta - meta}{0.035} \right)^2 + \left(\frac{\varphi'}{2.3} \right)^2 \right)} \quad (c.18)$$

where *peta* is the detector eta for each point in the EEMC, and *meta* is the central value of detector eta for each etabin. φ' is the modified azimuthal angle for each point. The azimuthal angle for each point in formula (c.18) is recalculated as modulo (6 degrees), because zero degrees are exactly located at the center of one subsector. The widths along

the two dimensions were set to 0.035 radian and 2.3 degree based on simulation study (see following Chapter). More details about how we set this bivariate Gaussian distribution will be discussed in the later simulation and real data analysis part. But the conclusion is that this change does not help much in improving pion yield by recovering from edge effects.

The final cut in the ‘accept’ routine is the detector η cut, which is set from 1.086 to 2.0. Most of those threshold values are defined in the Spin-Cuts class, and we are just using the parameters here. Once the π^0 candidate qualifies and is accepted by the function, we call the addPair function to add this π^0 into the mRealEvent. A number of histograms are incremented at this point. User-specified histograms can be also filled in this region by their private settings. A simple combinatoric background algorithm was also developed in the code, but eventually we decided not to use this method in our background study for real data analysis, so I will not discuss this part here. At the end of this part, the tree defined above is then filled according to the mRealEvent, which means all qualified π^0 and event information is stored into trees for further analyses.

C.5.4 The Pi0-Reader class

This class provides the framework to open the π^0 tree for further analyses. So it must be utilized with other macros to realize more functions. For example, this class is used in the normalization of spin-dependent pion yields, according to the relative luminosities in the double-spin asymmetry calculation.

In the header file, we define the constructor and destructor for the Pi0-Reader, and the important function 'chainFile' and a Mix-Event type of event are both defined. These definitions open the gate to π^0 trees.

This is a relatively simple class. A name is required to be assigned to the class at the beginning, which should be consistent with what we defined in the Pi0-Analysis maker. A new chain is then defined according to the name, so that when we initialize the class, the branch address can be directed to where we save our trees. Then when we call the event defined in the head file from the macro, it will give us the mRealEvent stored in the trees. The function 'chainFile' is the important running part in this class, which checks the file name to see if it is a root format file, and adds the root format file into the chain. By implementing this class in a certain analysis macro, we only need to play with the root files which will be generated from the π^0 finder program. Thus this Pi0-Reader class makes our whole analysis more flexible and user-friendly.

Bibliography

- [1] D. de Florian, R. Sassot , M. Stratmann , W. Vogelsang. BNL-NT-08-8, Apr 2008.
Phys.Rev.Lett.**101**:072001,2008.
- [2] D. de Florian, R. Sassot, Phys. Rev. D **62**, 094025 (2000).
- [3] D. de Florian, O. A. Sampayo, R. Sassot, Phys, Rev. D **57**, 5803 (1998).
- [4] D. K. Ghosh, S. Gupta, and D. Indumathi, Phys. Rev. D **62**, 094012 (2000).
- [5] M. Glück, E. Reya, M. Stratmann, W. Vogelsang, Phys. Rev. D **53**, 4775 (1996).
- [6] Asymmetry Analysis Collaboratoin. Y. Goto *et al.*, Phys. Rev. D **62**, 094012(2000).
- [7] L. E. Gordon, M. Goshtasbpour, G.P.Ramsey, Phys. Rev. D **58**, 094017 (1998).
- [8] E. A. Leader, A. V. Sidorov, D. B. Stamenov, Int. J. Phys. Lett. B **488**, 283 (2000).
- [9] G. Altarelli, R. D. Ball, S. Forte, G. Ridolfi, Nucl. Phys. **B496**, 337 (1997).
- [10] R. D. Ball, S. Forte, G. Ridolfi, Acta Phys. Pol. B **29**, 1145 (1998).
- [11] S. Tatur, J. Bartelski, M. Kurzela, Acta Physi. Pol. B **31**, 647 (2000).
- [12] C. Bourrely, F. Buccella, O. Pisanti, P. Santorelli, J. Soffer, Prog. Theor. Phys. **99**, 1017 (1998).
- [13] SLAC-Yale E80 Collaboration, M. J. Alguard *et al.*, Phys. Rev. Lett. **37**, 1261 (1976); G. Baum *et al.*, *ibid.* 45, 2000 (1980); SLAC-Yale-E130 Collaboration, G. Baum *et al.*, *ibid.* **51**, 1135 (1983).
- [14] SLAC-E154 Collaboration, K. Abe *et al.*, Phys. Lett. B **405**, 180 (1997).
- [15] SLAC-E154/155 Collaboration, S. Incerti, F. Sabatie, CEA-Saclay PCCF-RI-9901, 1999.
- [16] SLAC-E142 Collaboration, P. L. Anthony *et al.*, Phys. Rev. D **54**, 6620 (1996).
- [17] SLAC-E143 Collaboration, K. Abe *et al.*, Phys. Rev. D **58**, 120003 (1998).

- [18] SLAC-E154 Collaboration, K. Abe *et al.*, Phys. Rev. Lett. **79**, 26 (1997).
- [19] SLAC-E155 Collaboration, P. L. Anthony *et al.*, Phys. Lett. B **463**, 339 (1999).
- [20] SLAC-E155 Collaboration, P. L. Anthony *et al.*, Phys. Lett. B **493**, 19 (2000).
- [21] DESY-HERMES Collaboration, K. Ackerstaff *et al.*, Phys. Lett. B **404**, 383(1997).
- [22] DESY-HERMES Collaboration, A. Airapetian *et al.*, Phys. Lett. B **442**, 484(1998).
- [23] CERN-SM Collaboration, B. Adeva *et al.*, Phys. Rev. D **58**, 112001 (1998).
- [24] CERN-SM Collaboration, B. Adeva *et al.*, Phys. Rev. D **58**, 112002 (1998).
- [25] CERN-EM Collaboration, J. Ashman *et al.*, Phys. Lett. B **206**, 364 (1988).
- [26] G. A. Navarro, R. Sassot, Phys. Rev. D **74**, 011502 (2006).
- [27] M. Glück *et al.*, Phys. Rev. D **63**, 094005 (2001).
- [28] A. Airapetian *et al.*, arXiv: 0803. 2993 [hep-ex].
- [29] A. Adare *et al.*, Phys. Rev. D **76**, 051106 (2007).
- [30] S. D. Bass, Rev. Mod. Phys. **77**, 1257 (2005).
- [31] M. Alekseev *et al.*, Phys. Lett. B **660**, 458 (2008).
- [32] E. S. Ageev *et al.*, Phys. Lett. B **633**, 25 (2006).
- [33] E. A. Leader *et al.*, Phys. Rev. D **73**, 034023 (2006).
- [34] B. Adeva *et al.*, Phys. Lett. B **420**, 180 (1998).
- [35] A. Airapetian *et al.*, Phys. Lett. B **464**, 123 (1999).
- [36] A. Bressan *et al.*, hep-ex/0501040.
- [37] X. Jiang *et al.*, hep-ex/0412010.
- [38] D. de Florian APS talk, April 2008.
- [39] D. de Florian, G. A. Navarro, R. Sassot, Phys. Rev. D **71**, 094018 (2005).
- [40] M. Glück, E. Reya, M. Stratmann, W. Vogelsang, Phys. Rev. D **63**, 094005 (2001).

- [41] C. Bourrely, J. Soffer, F. M. Renard, P. Taxil, Phys. Rep. **177**, 319 (1989).
- [42] T. Gehrmann, W. J. Stirling, Phys. Rev. D **53**, 6100 (1996).
- [43] M. Hirai, S. Kumano, M. Miyama, Comp. Phys. Rep. **177**, 319 (1989).
- [44] L. C. Bland, Workshop on High Energy Nuclear Physcs (EPIC99), 255 (1999).
- [45] Aidala C *et al.*, <http://spin.riken.bnl.gov/rsc/report/masterspin.pdf>
- [46] J. C. Collins, Nucl. Phys. B **394**, 169 (1993).
- [47] B. Jäger, A. Schäfer, M. Stratmann, W. Vogelsang, Phys. Rev. **D67**, 054005(2003).
- [48] D. de Florian, Phys. Rev. **D67**, 054004 (2003).
- [49] F. Aversa, P. Chiappetta, M. Greco, J. P. Gulliet, Nucl. Phys. B **327**, 105 (1989).
- [50] J. Pumplin, *et al.*, JHEP **0207**, 012 (2002).
- [51] V. Guzey, M. Strikman, W. Vogelsang, Phys. Lett. B **603**, 173 (2004).
- [52] M. Beddo *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**:725-739 (2003).
- [53] A. Hoffman, talk at DIS, University College, London, April, 5, 2008.
- [54] O. Grebenyuk, A. Mischke (NIKHEF, Amsterdam & Utrecht U.) , A. Stolpovsky (Wayne State U.) . Oct 2005. 4pp. Rom.Rep.Phys.**58**:025-030,2006.
- [55] S. S. Adler *et al.*, PHENIX Collaboration. Phys. Rev. Lett. **93**:202002 (2004).
- [56] I. Alekseev *et al.*, <http://www.agsrhichome.bnl.gov/RHIC/Spin/design/>
- [57] “RHIC Design Manual”, Brookhaven National Lab.
- [58] Y. Mori, K. Ikegami, Z. Igarashi, A. Takagi, S. Fukumoto, AIP Conf. Proc. No. **117**. P. 123 (1983).
- [59] A. Zelenski, J. Alessi, A. Koponou, J. Ritter, V. Zubets. Proc. PAC07 (2007).
- [60] J. Claus, H. Foelsche, Beam Transfer from AGS to RHIC, RHIC 47 (1988).
- [61] S. Y. Lee, E. D. Courant, BNL Technical Note AD/RHIC-63.

- [62] F. Z. Khiari *et al.*, Phys. Rev. D **39**, 45 (1989).
- [63] Ya. S. Derbenev *et al.*, Part. Accel. **8**, 115 (1978).
- [64] A. D. Krisch *et al.*, Phys. Rev. Lett. **63**, 1137 (1989).
- [65] V. I. Ptitsin, Yu. M. Shatunov, BNL-52453, p. 15.
- [66] <http://www.star.bnl.gov/central/experiment/>
- [67] W. He. talk at APS, St. Louis, MO. April, 12, 2008.
- [68] K. H. Ackermann *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**, 624-632 (2003).
- [69] F. Bergsma *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**, 633-639 (2003).
- [70] M. Anderson *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,659-678(2003).
- [71] M. Anderson *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,679-691(2003).
- [72] J. Abele *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,692-702(2003).
- [73] M. Anderhub, M. J. Devereux, P. G. Seiler, Nucl. Instrum. Meth.**166** (1979) 581.
- [74] L. Kotchenda *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,703-712(2003).
- [75] K. H. Ackermann *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,713-719 (2003).
- [76] R. Bellwied *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,640-651(2003).
- [77] L. Arnold *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,652-658(2003).
- [78] M. M. Arggawal, STAR Collaboration, Nucl. Instrum. Meth. **A499**,751-761(2003).
- [79] L. Nogach, STAR Collaboration, arXiv:hep-ex/0612030v2 (2007).
- [80] M. Allen, STAR Collaboration, Nucl. Instrum. Meth. **A499**,549-559(2003).
- [81] H. S. Martis, STAR Collaboration, Nucl. Instrum. Meth. **A499**,802-813(2003).

- [82] C. E. Allgower *et al.*, STAR Collaboration, Nucl. Instrum. Meth. **A499**,740-750 (2003).
- [83] SMD geometry. <http://www.star.bnl.gov/public/eemc>
- [84] Technical Design Report for the BEMC.
- [85] A. Ljubicic Jr, M. Botlo, F. Heisterman, S. Jacobson, M. J. LeVine, J. M. Nelson, M. Nguyen, H. Roehrich, E. Schaefer, J. J. Schanback, R. Scheetz, D. Schmischke, N. W. Schulz, K. Sulimma, IEEE. Trans. Nucl. Sci. Ns. **45**, (4) (1998) 1907.
- [86] A. Ljubicic Jr et al., IEEE. Trans. Nucl. Sci. Ns. **47** (2) (2000) 99.
- [87] J. M. Landgraf et al., Nucl. Instrum. Meth. **A499**,762-765 (2003).
- [88] F. S. Bieser et al., Nucl. Instrum. Meth. **A499**,766-777(2003).
- [89] H. J. Crawford, V. Lindenstruth, Apparatus and Method for managing digital resources by passing digital source tokens between queues, US patent 5,918,055,1999.
- [90] J. M. Landgraf *et al.*, IEEE Trans. Nucl. Sci. Ns. **48** (3) (2001) 601.
- [91] N. J. Boden *et al.*, IEEE Micro. **15** (1) (1995) 29.
- [92] Particle Data Group. <http://pdg.lbl.gov>
- [93] Four spin states, <http://www.star.bnl.gov/~akio/spinbit.html>
- [94] GEANT Software, <http://www.geant.net/>
- [95] Jason Webb's pi0 finder software:
<http://www.star.bnl.gov/cgi-bin/protected/cvsweb.cgi/StRoot/StEEmcPool/>
- [96] Trigger patch layout. http://www.iucf.indiana.edu/U/STAR/eemc_proj/
- [97] Relative luminosity:
http://deltag5.lns.mit.edu/~sakuma/protected/star/scaler/2006_b/prelim_release2.shtml