PWA Challenge with polarized photon beam

Florida International University 2020

Mariana Khachatryan

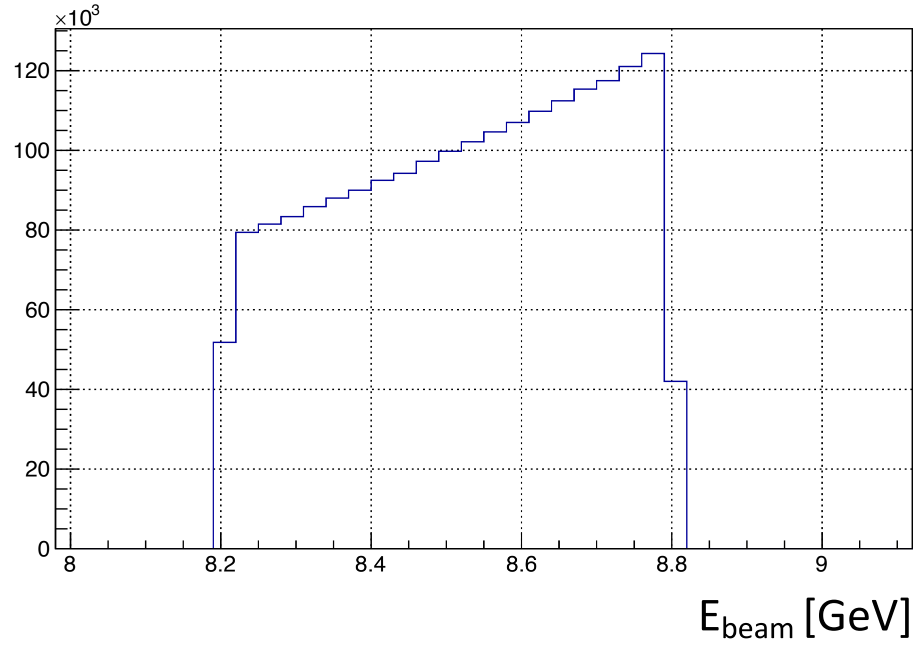
Generated 2*10⁶ $(p\eta'\pi^0)$events with AmpTools

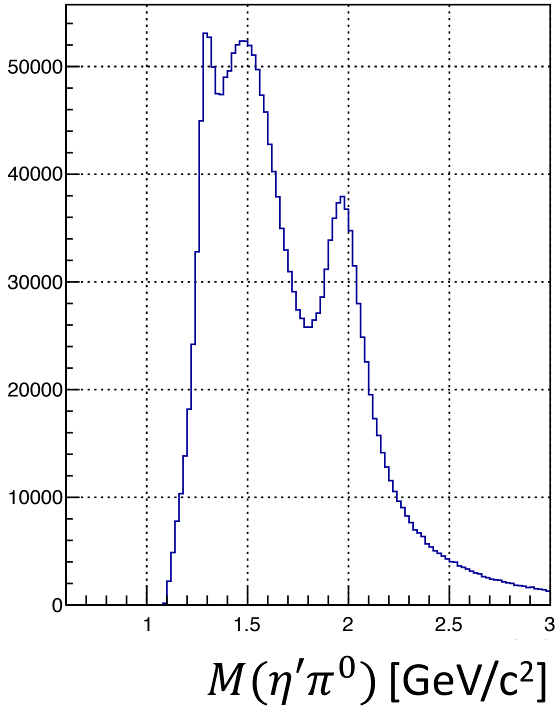
Generated amplitudes are
- P1/π_1(1600 MeV) (exotic)
- D1/a2 (1320 MeV)
- G1/a_4 (1995)

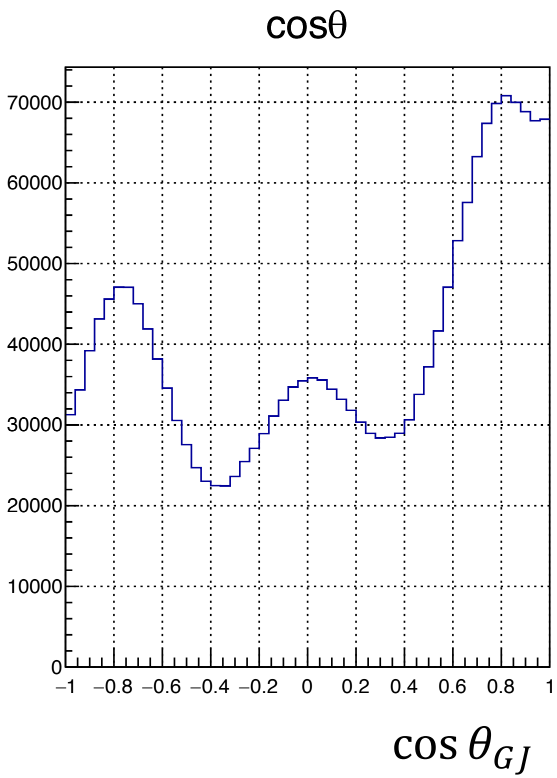
J	M	ε	Real	Imaginary	BW Mass	BW Width
1	0, 1	+1	200	200	1.564	0.492
2	0, 1	+1	50	50	1.306	0.114
4	2	+1	5	0	1.996	0.255

Φ=1.77 Deg.

P_γ =0.3







Config file for fitting with generated amplitudes

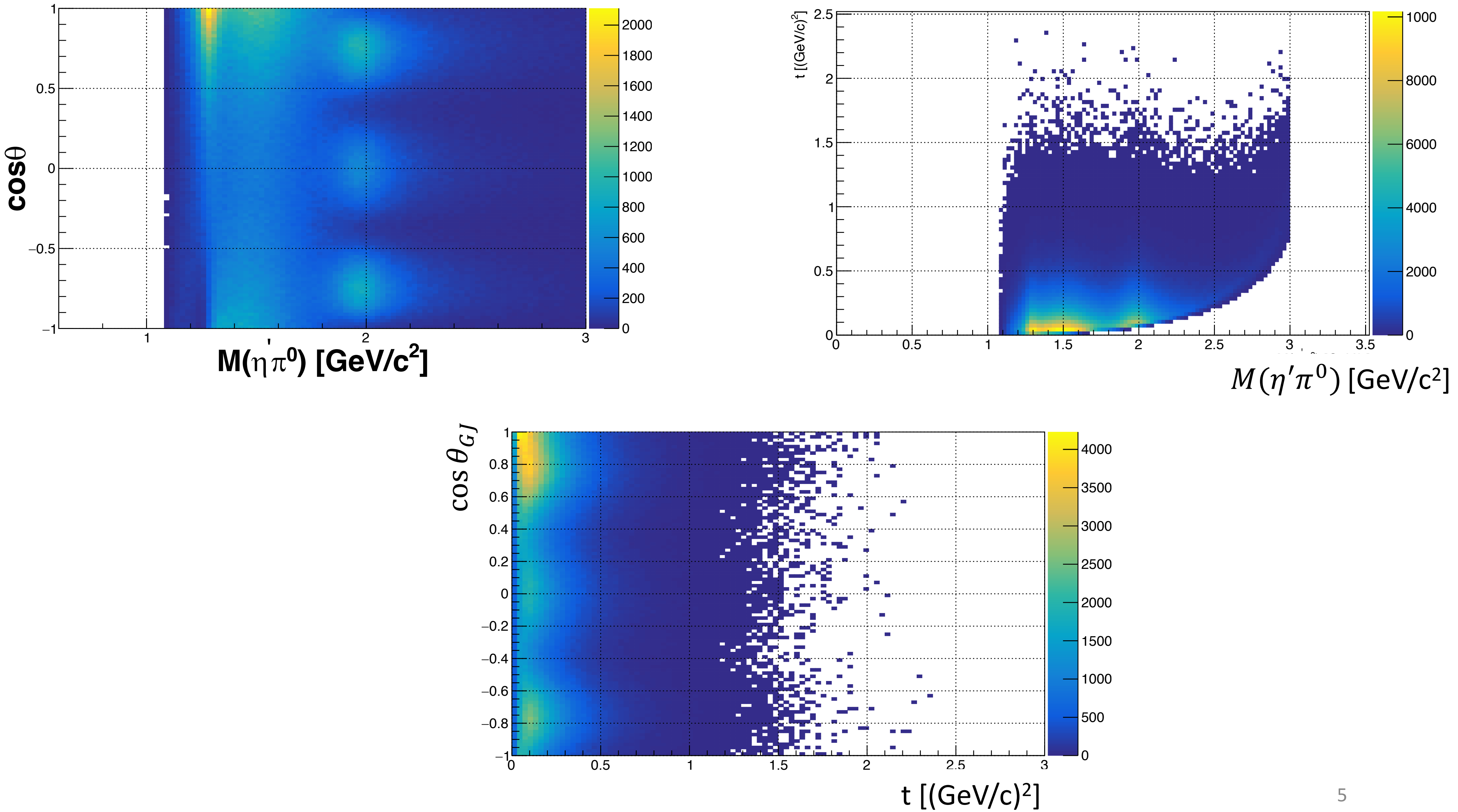
```
define polVal 0.3

# definition of resonances
define azero 0.980 0.075
define atwolight 1.306 0.114
define pione 1.564 0.492
define afour 1.996 0.255

fit FITNAME

reaction EtaPrimePi0 Beam Proton EtaPrime Pi0

sum EtaPrimePi0 PositiveRe
sum EtaPrimePi0 PositiveIm

genmc  EtaPrimePi0 ROOTDataReader GENMCFILE
accmc  EtaPrimePi0 ROOTDataReader ACCMCFILE
data EtaPrimePi0 ROOTDataReader DATAFILE

parameter polAngle 1.77 fixed

# a2(1320)
amplitude EtaPrimePi0::PositiveIm::D0+ Zlm 2 0 -1 -1 polAngle polVal
amplitude EtaPrimePi0::PositiveRe::D0+ Zlm 2 0 +1 +1 polAngle polVal
amplitude EtaPrimePi0::PositiveIm::D1+ Zlm 2 1 -1 -1 polAngle polVal
amplitude EtaPrimePi0::PositiveRe::D1+ Zlm 2 1 +1 +1 polAngle polVal
# pi1(1600)
amplitude EtaPrimePi0::PositiveIm::P0+ Zlm 1 0 -1 -1 polAngle polVal
amplitude EtaPrimePi0::PositiveRe::P0+ Zlm 1 0 +1 +1 polAngle polVal
amplitude EtaPrimePi0::PositiveIm::P1+ Zlm 1 1 -1 -1 polAngle polVal
amplitude EtaPrimePi0::PositiveRe::P1+ Zlm 1 1 +1 +1 polAngle polVal
# a4(1996)
amplitude EtaPrimePi0::PositiveIm::G2+ Zlm 4 2 -1 -1 polAngle polVal
amplitude EtaPrimePi0::PositiveRe::G2+ Zlm 4 2 +1 +1 polAngle polVal
```

```
Zlm as suggested in GlueX doc-4094 (M. Shepherd)
argument 1 : j
argument 2 : m
argument 3 : real (+1) or imaginary (-1) part
argument 4 : 1 + (+1/-1) * P_gamma
argument 5 : polarization angle (in Deg.)
argument 6 : beam properties config file or fixed
polarization
```

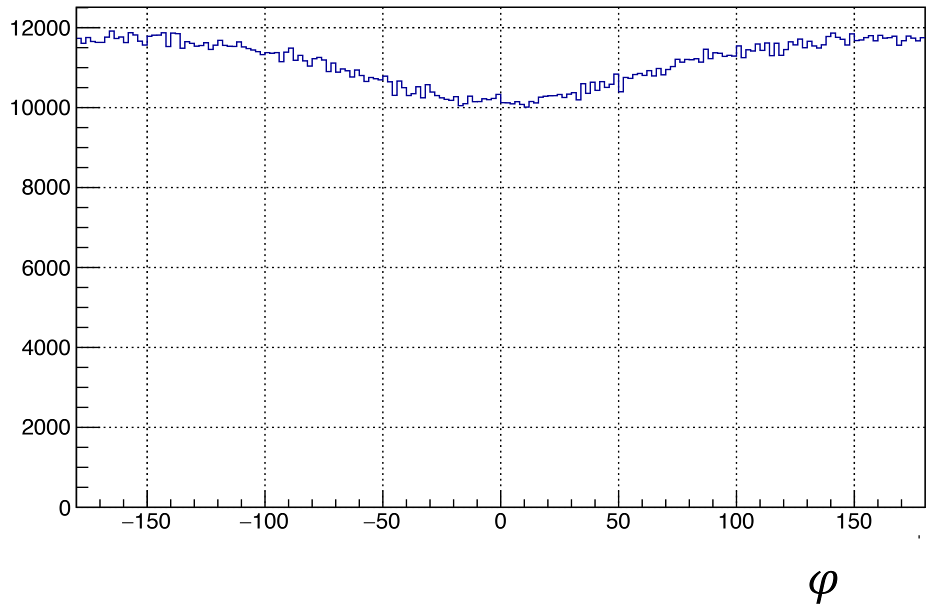
Config file for fitting with generated amplitudes

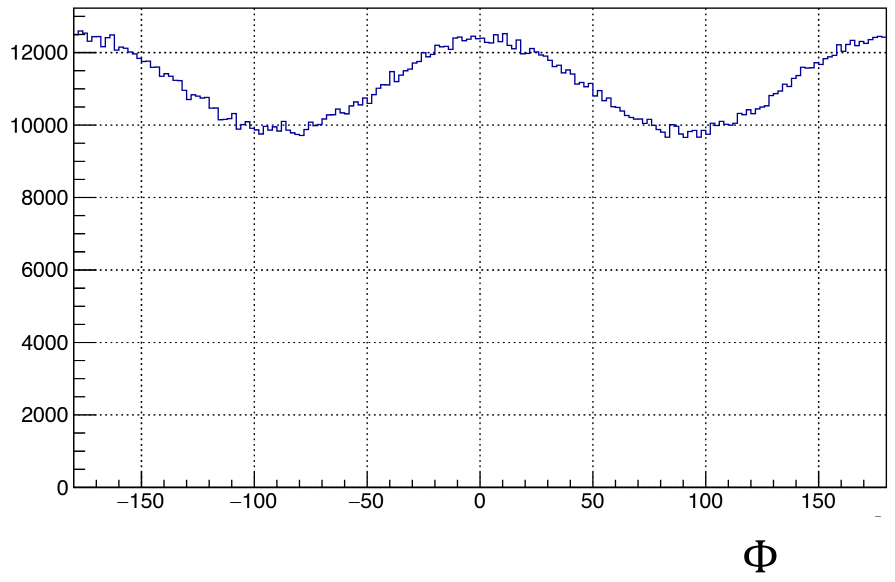
```
# initialize only positive reflectivity components
initialize EtaPrimePi0::PositiveIm::P0+ cartesian 200.0 200.0
initialize EtaPrimePi0::PositiveRe::P0+ cartesian 200.0 200.0
initialize EtaPrimePi0::PositiveIm::P1+ cartesian 200.0 200.0
initialize EtaPrimePi0::PositiveRe::P1+ cartesian 200.0 200.0


initialize EtaPrimePi0::PositiveIm::D0+ cartesian 50.0 50.0
initialize EtaPrimePi0::PositiveRe::D0+ cartesian 50.0 50.0
initialize EtaPrimePi0::PositiveIm::D1+ cartesian 50.0 50.0
initialize EtaPrimePi0::PositiveRe::D1+ cartesian 50.0 50.0


initialize EtaPrimePi0::PositiveIm::G2+ cartesian 5.0 0.0 real
initialize EtaPrimePi0::PositiveRe::G2+ cartesian 5.0 0.0 real


constrain EtaPrimePi0::PositiveIm::P0+ EtaPrimePi0::PositiveRe::P0+
constrain EtaPrimePi0::PositiveIm::P1+ EtaPrimePi0::PositiveRe::P1+
constrain EtaPrimePi0::PositiveIm::D0+ EtaPrimePi0::PositiveRe::D0+
constrain EtaPrimePi0::PositiveIm::D1+ EtaPrimePi0::PositiveRe::D1+
constrain EtaPrimePi0::PositiveRe::G2+ EtaPrimePi0::PositiveIm::G2+
```

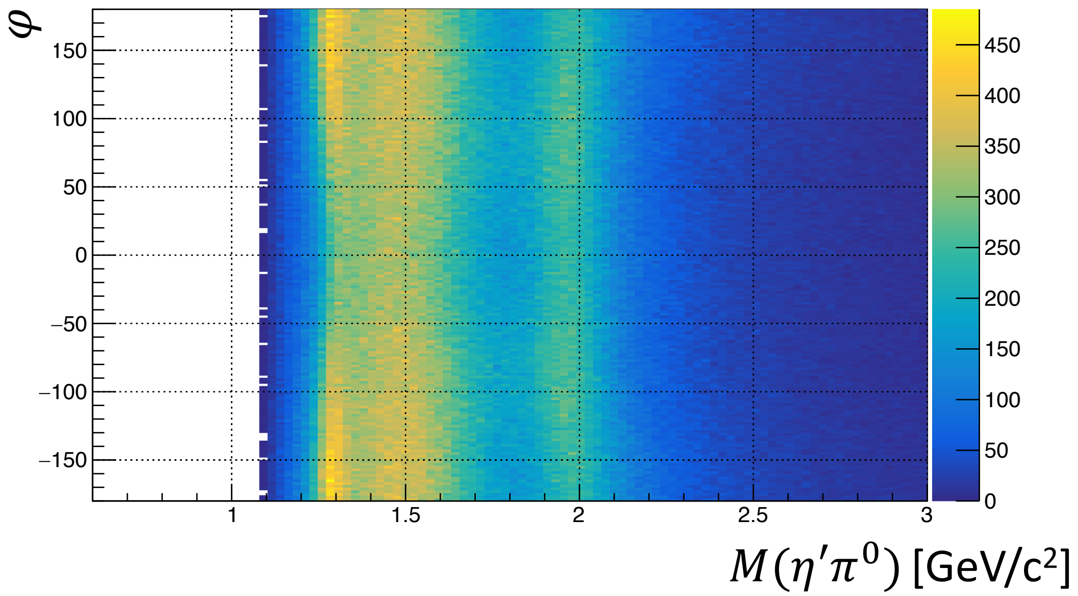
Generated $2*10^6$ $(p\eta'\pi^0)$events with AmpTools

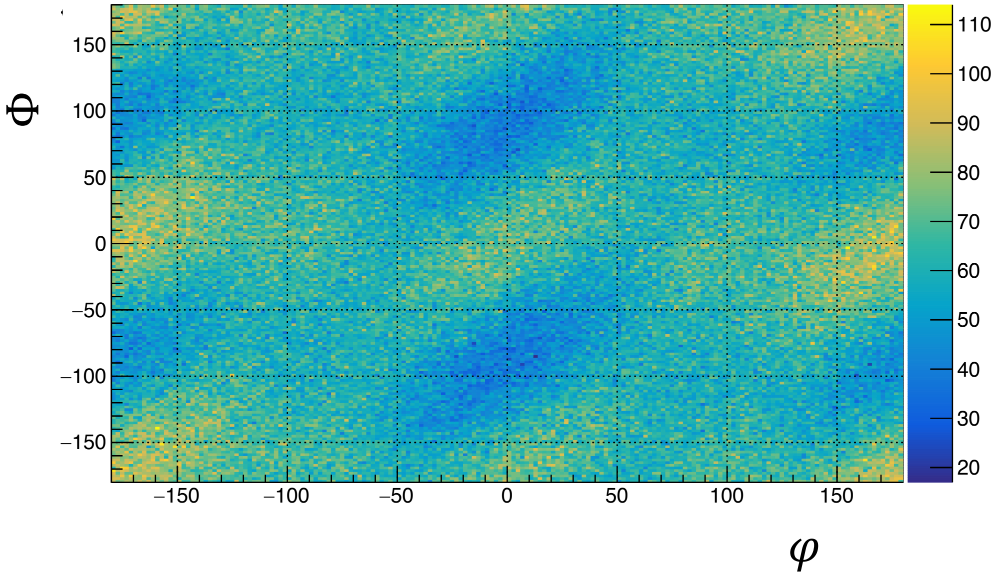


Generated $2*10^6$ $(p\eta'\pi^0)$events with AmpTools









Procedure

Fitting in M and t bins

1. divideData.pl

Will create folders for each mass and t bin with corresponding data file and .cfg for fitting using split_mass_t.cc.
Edit .pl to give the path for data, generated and accepted Monte Carlo files.

2. ./driveFit.pl

Script that performs the fit in each mass and t bin.

3. plot_etapi_delta_mass_t_bins –o etaprimepi0_fit.txt

Program that gathers the results (amplitude intensities) of the fits

4. Python Drawing_waves_M_t_bins.py “P0+ P1+” 40 4 etaprimepi0_fit.txt “P0P1phasediff.”

Reads the intensities of different waves, phase differences as well as total intensity from etapi_fit.txt and draws them

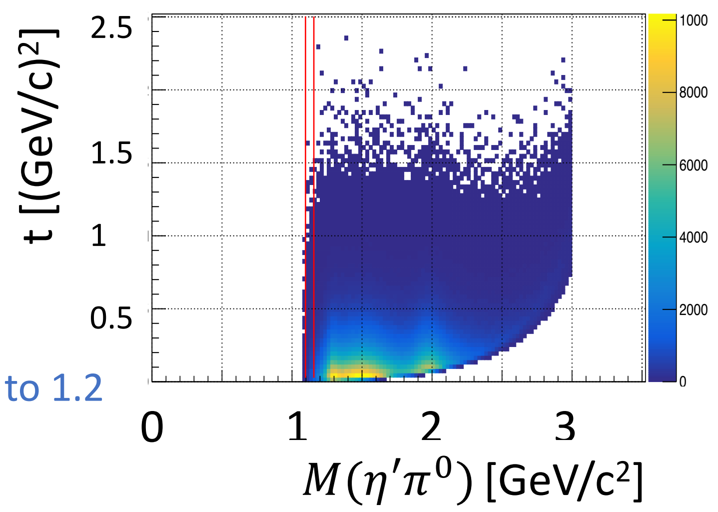
Results for bin M=1.12 and t<0.3

Amplitudes used in fitting are D0+, D1+, P0+, P1+, G2+

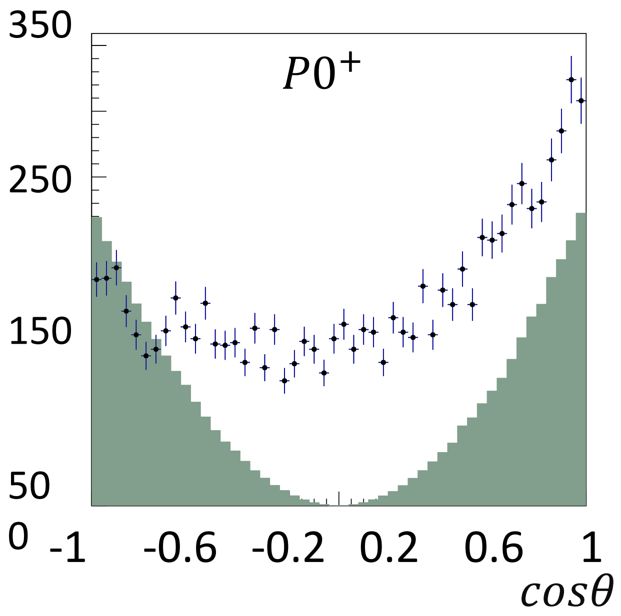
Bin M , t

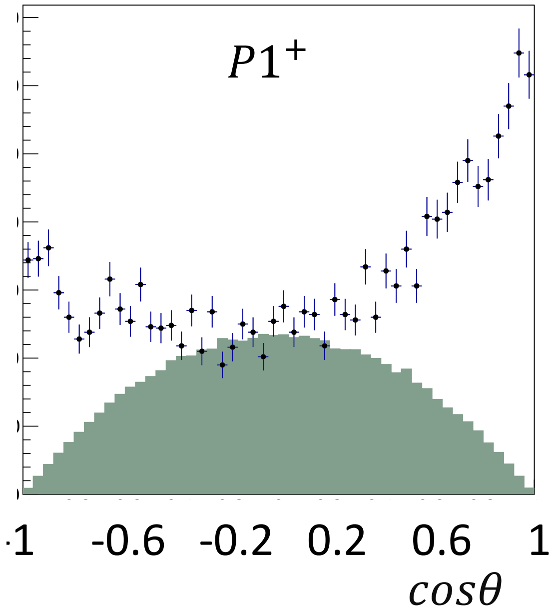
$M(\eta\pi^0)$ range from 1.1 to 3
N bins=40
Bin width≈ 0.0475 GeV/c²

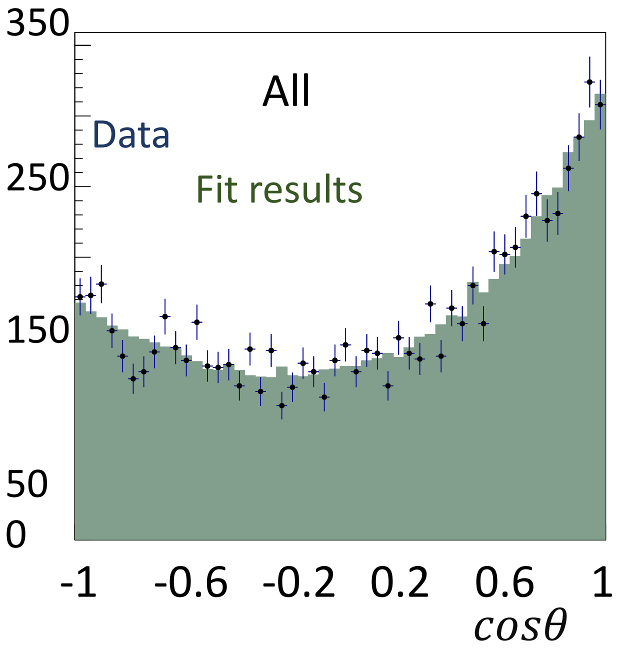
t range from 0 to 1.2
N bins=4
Bin width≈ 0.3 (GeV/c)2

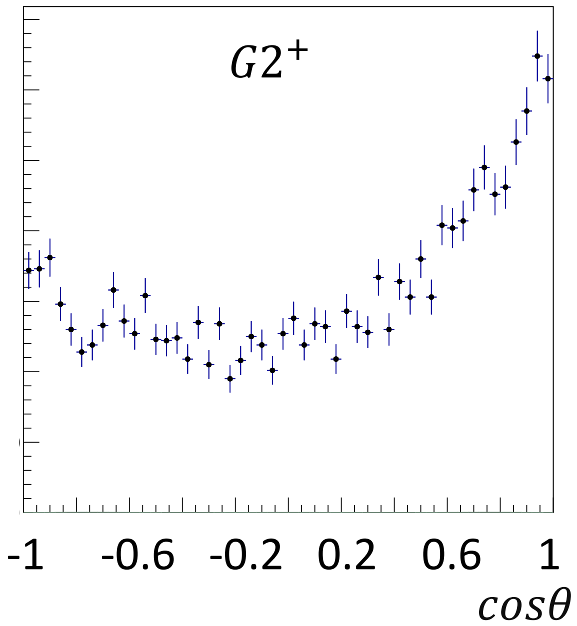


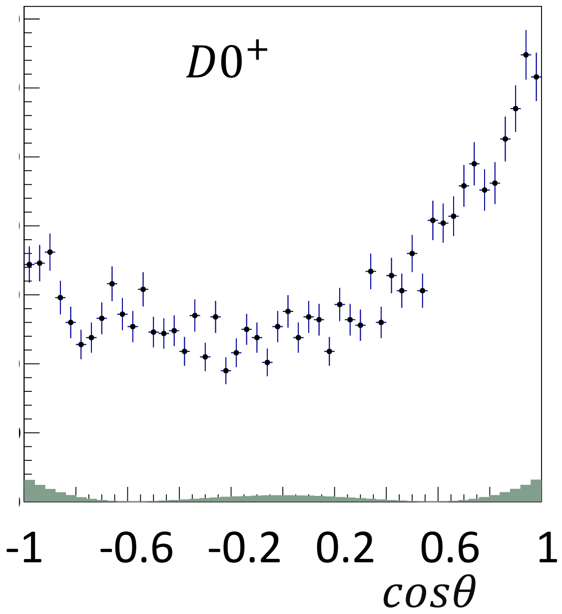
Fit results

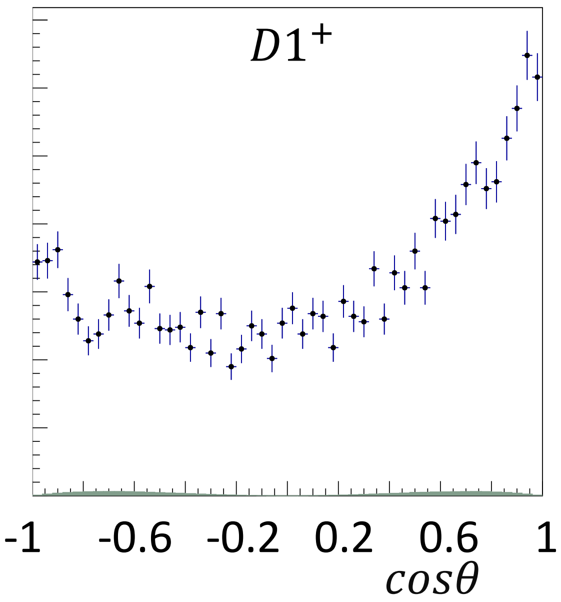












Procedure

Bootstrapping in M and t bins

1. divideData_Mass_t_Bins_Bootstrap.py
Divides data, generated and accepted Monte Carlo .root data files into different M and t bins.

2. fit_etapi_TEMPLATE.cfg configuration used in fitting.

3. One needs to copy fit_TEMPLATE.csh, fit_TEMPLATE.py
To sets up environment and fit 100 bootstrapping samples for given M and t bin

3. Create workflow "swif create -workflow WORKFLOW". "WORKFLOW" needs to be the same as the name in script_Fit.py

4. Run script_Fit.py to submit one job per mass and t bin to the ifarm. To run the jobs do "swif run -workflow WORKFLOW”

5. Bootstrap_plot_etapi_delta_ampmatch_mass_t_bins.cc
This program will create a text files in each mass and t bin that contains the intensity and uncertainties from MINUIT of each partial wave, total intensity and its error as well as phase differences between different partial waves and corresponding uncertainties.

6. Drawing_Bootstrap_errors_M_t_bins_phasediff.py
Plots the results from fitting the original data in different M and t bins, with the uncertainties from bootstrapping.
It takes as argument the wave set used in the fitting, number of M bins, number of t bins, the path of the directory that contains directories corresponding to M_t bins, the path for the text file with fit results from fitting the original data and list of phase differences to be plotted.

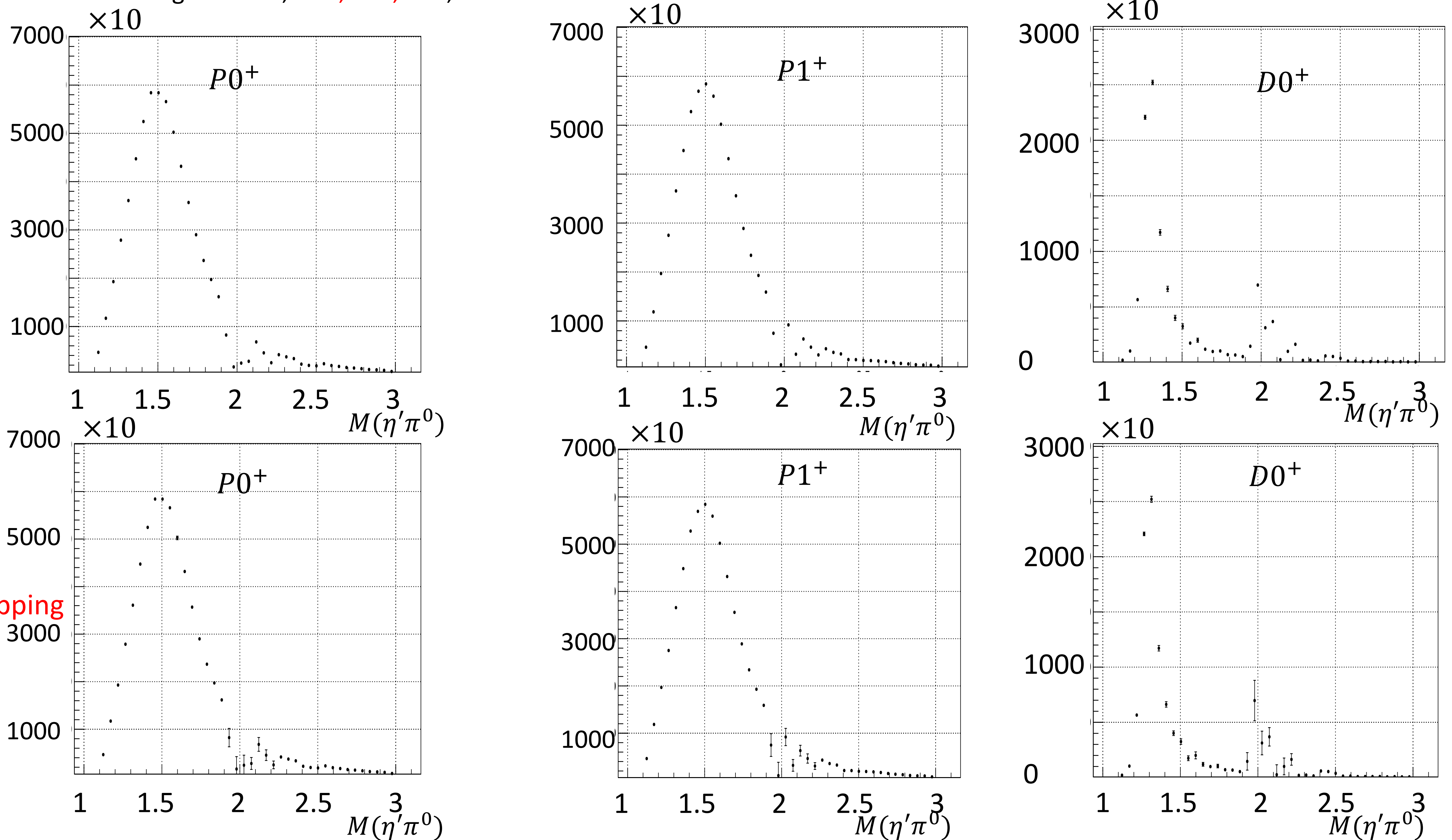
Initializing with the true values of production coefficients used in generation

Intensity as a function of invariant mass of $\eta'\pi^0$

Amplitudes used in fitting are D0+, D1+, P0+, P1+, G2+

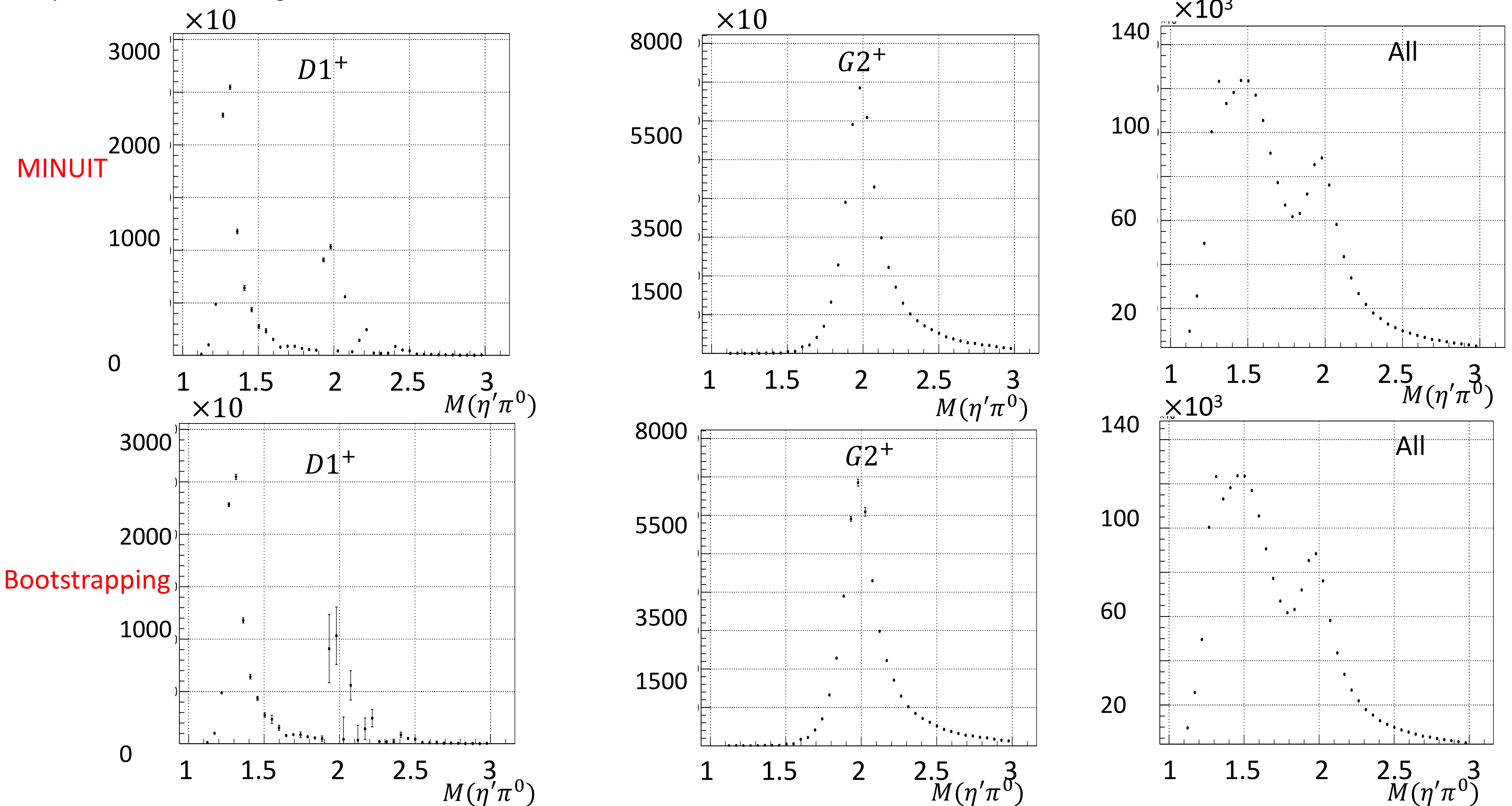
MINUIT

Bootstrapping



Intensity as a function of invariant mass of $\eta'\pi^0$

Amplitudes used in fitting are D0+, D1+, P0+, P1+, G2+

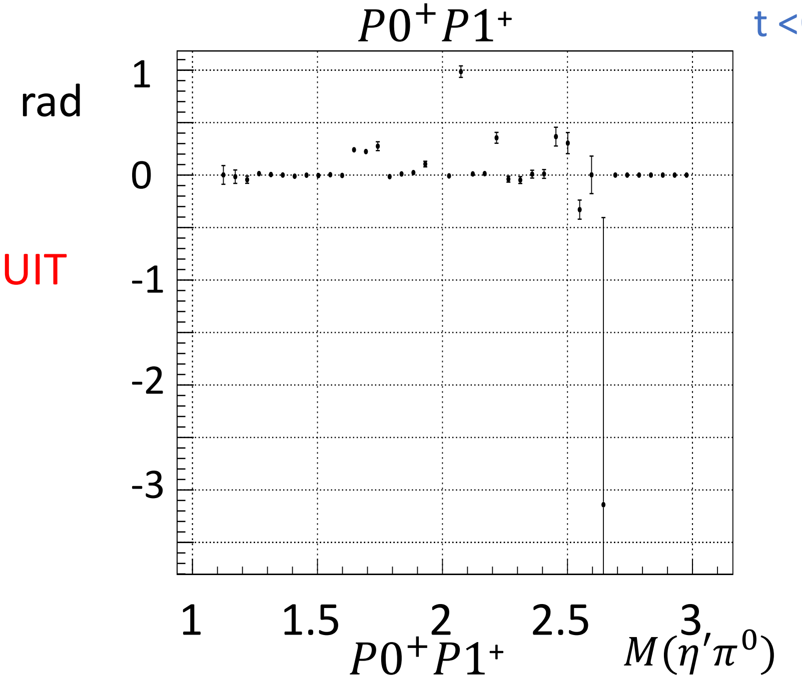


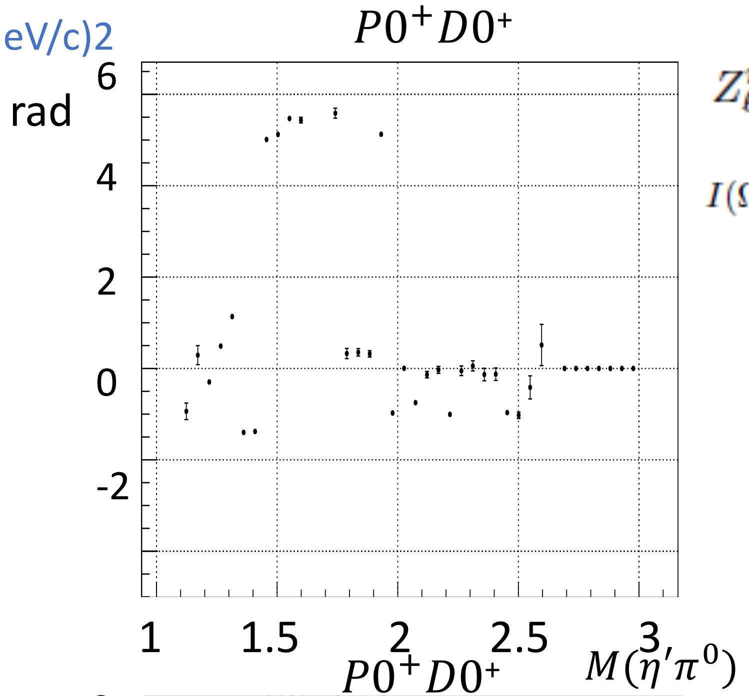
Phase difference as a function of invariant mass of $\eta'\pi^0$

Amplitudes used in fitting are D0+, D1+, P0+, P1+, G2+

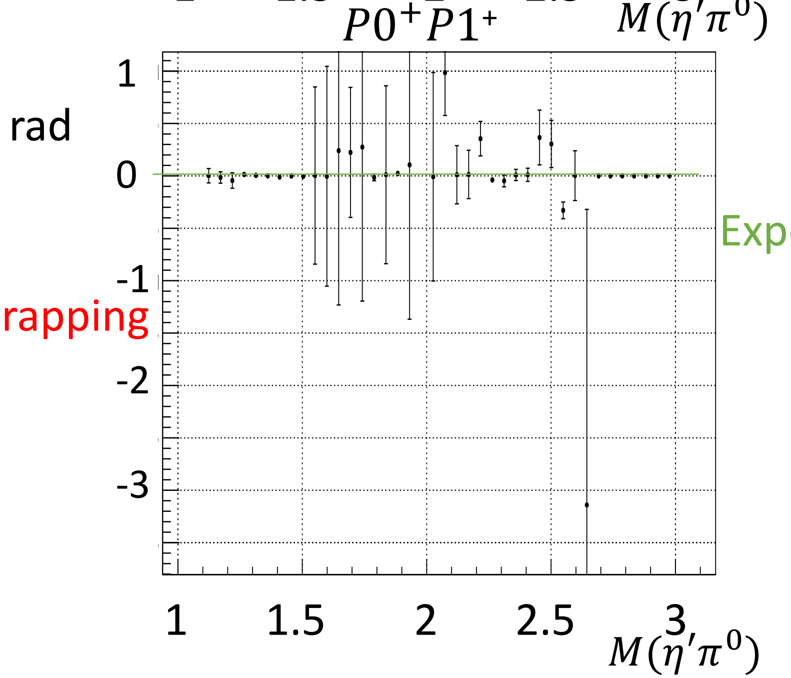
t <0.3 (GeV/c)2

MINUIT

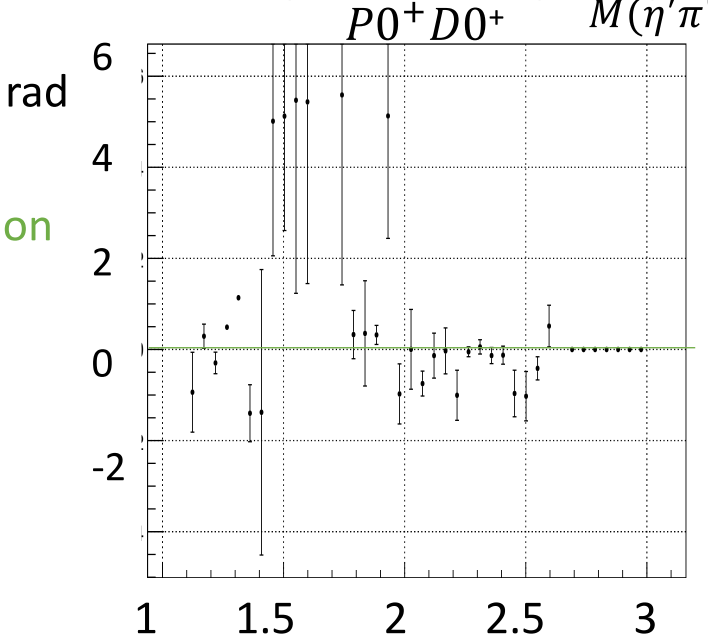




Bootstrapping



Expectation



$$Z_\ell^m(\Omega,\Phi) \equiv Y_\ell^m(\Omega)e^{-i\Phi}$$

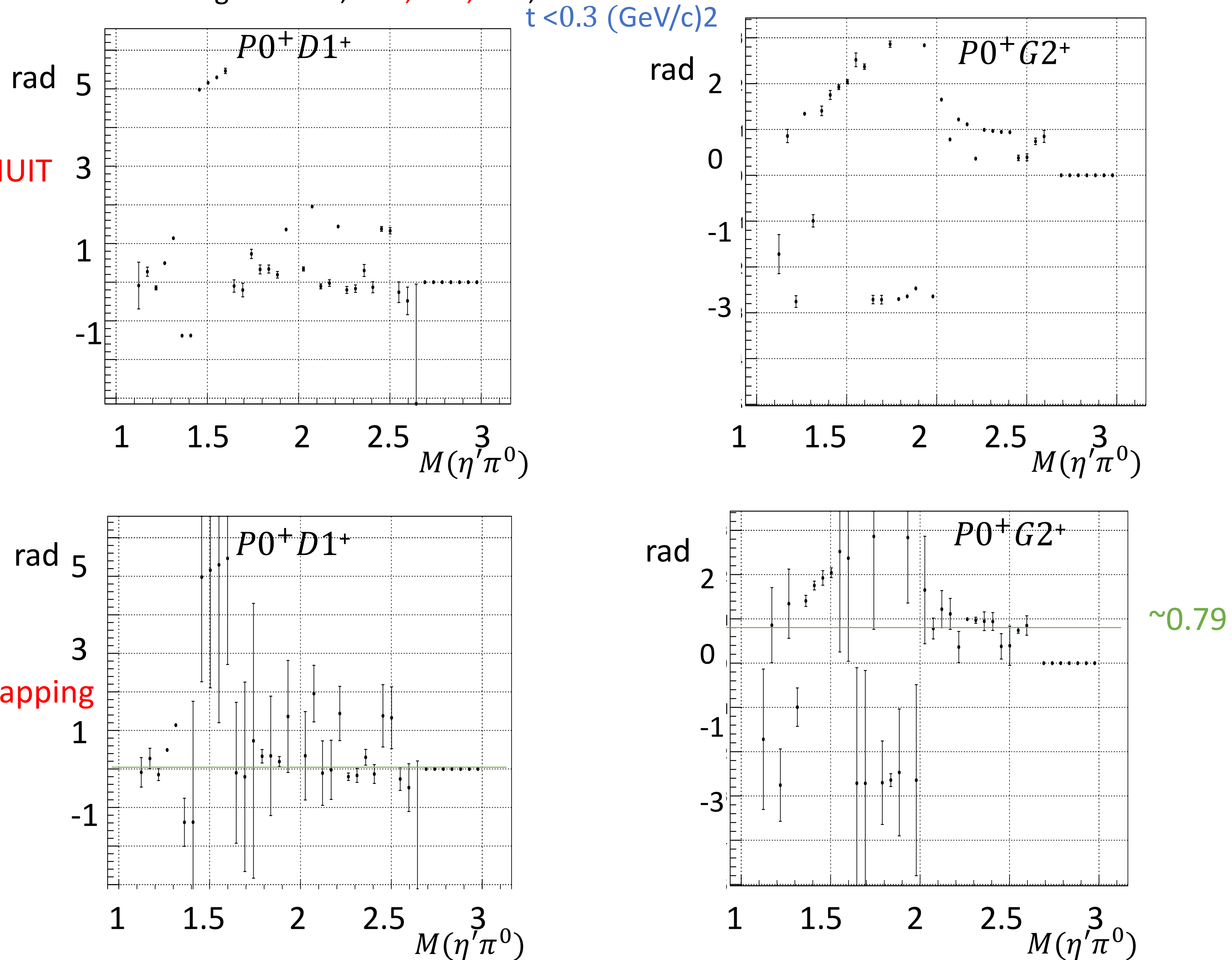
$$I(\Omega,\Phi) = 2\kappa\sum_k \Bigg\{ (1-P_\gamma)\Big|\sum_{\ell,m}[\ell]^{(-)}_{m;k}\mathrm{Re}[Z_\ell^m(\Omega,\Phi)]\Big|^2 + (1-P_\gamma)\Big|\sum_{\ell,m}[\ell]^{(+)}_{m;k}\mathrm{Im}[Z_\ell^m(\Omega,\Phi)]\Big|^2 + (1+P_\gamma)\Big|\sum_{\ell,m}[\ell]^{(+)}_{m;k}\mathrm{Re}[Z_\ell^m(\Omega,\Phi)]\Big|^2 + (1+P_\gamma)\Big|\sum_{\ell,m}[\ell]^{(-)}_{m;k}\mathrm{Im}[Z_\ell^m(\Omega,\Phi)]\Big|^2 \Bigg\}$$

Phase difference as a function of invariant mass of $\eta'\pi^0$

Amplitudes used in fitting are D0+, D1+, P0+, P1+, G2+

MINUIT

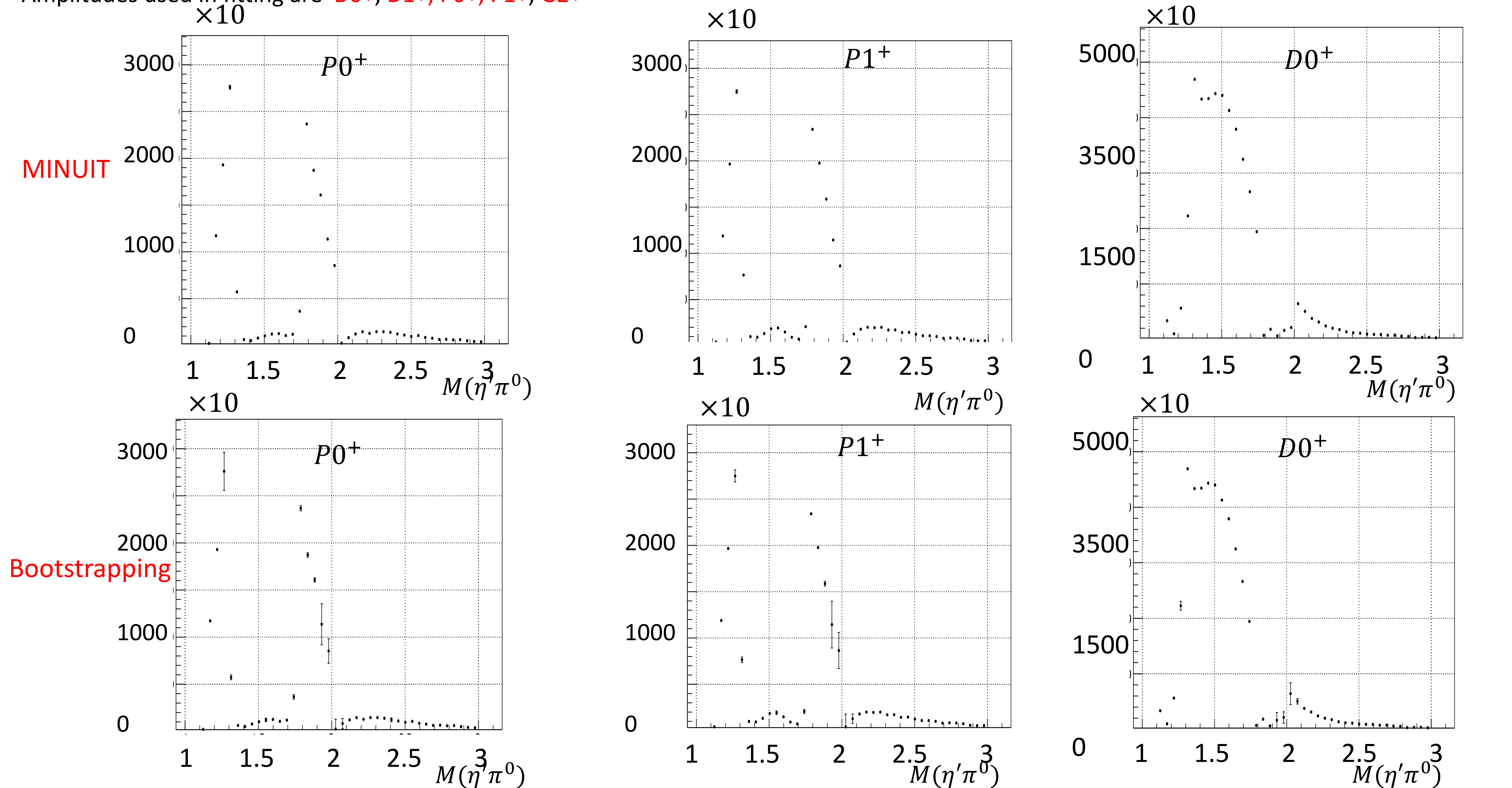
Bootstrapping



Initializing all the production coefficients with one

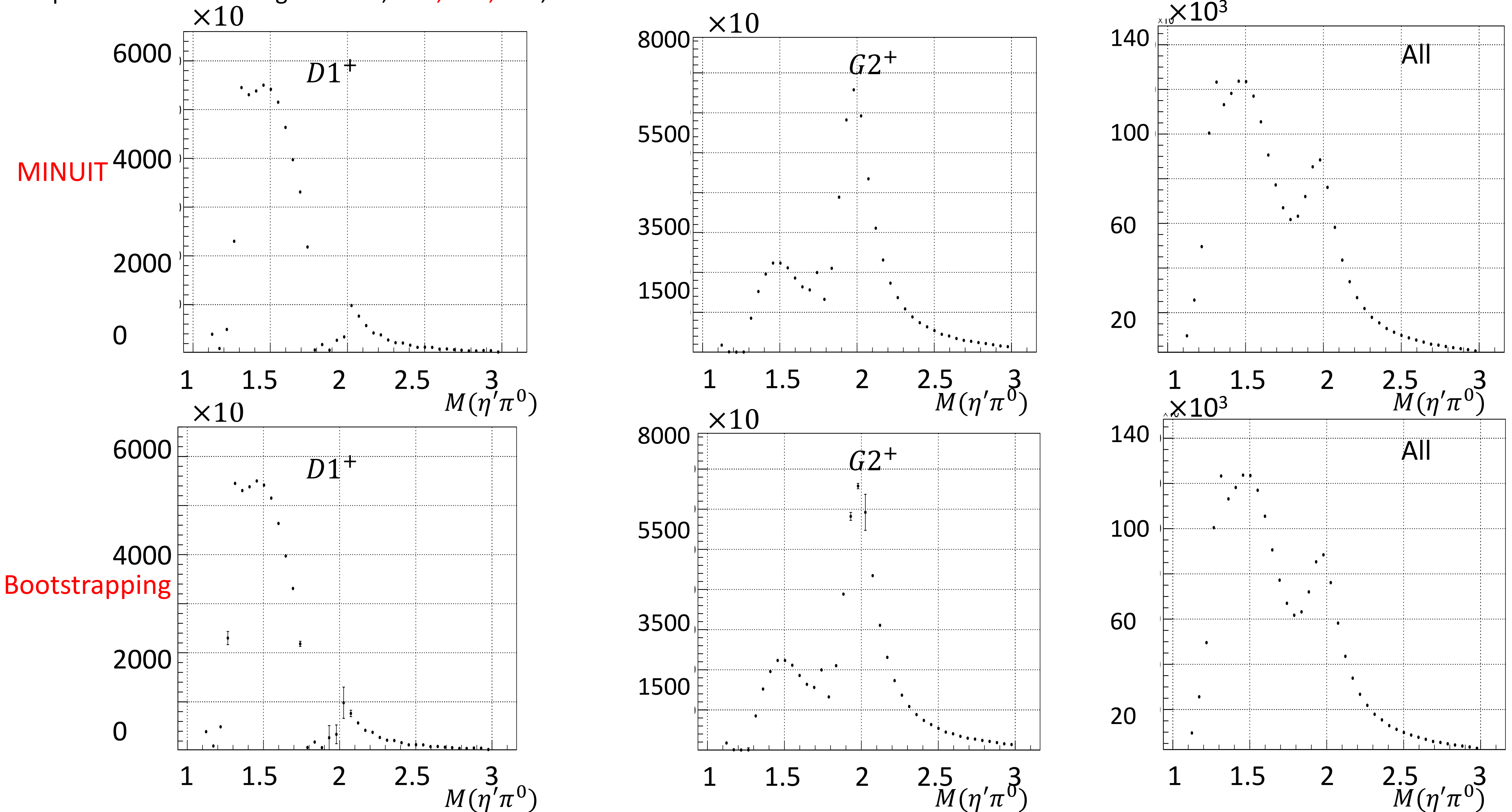
Intensity as a function of invariant mass of $\eta'\pi^0$

Amplitudes used in fitting are D0+, D1+, P0+, P1+, G2+



Intensity as a function of invariant mass of $\eta'\pi^0$

Amplitudes used in fitting are D0+, D1+, P0+, P1+, G2+



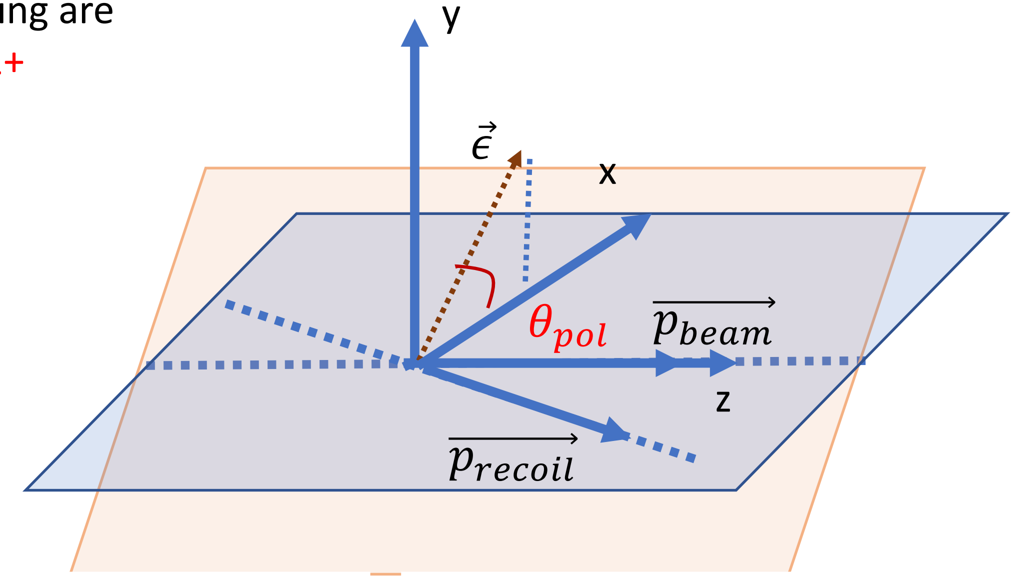
Backup

Mass dependent fit

1. Assume acceptance $\eta(\theta, \varphi) = 1$ and use same MC sample for both accepted and generated MC.
2. Execute fitting:
fit -c Mar_fit_etaprime_polarized.cfg
3. To plot the results in GUI:
twopi_plotter_mom etaprimepi0.fit -g

Results with angle between production and polarization planes Φ

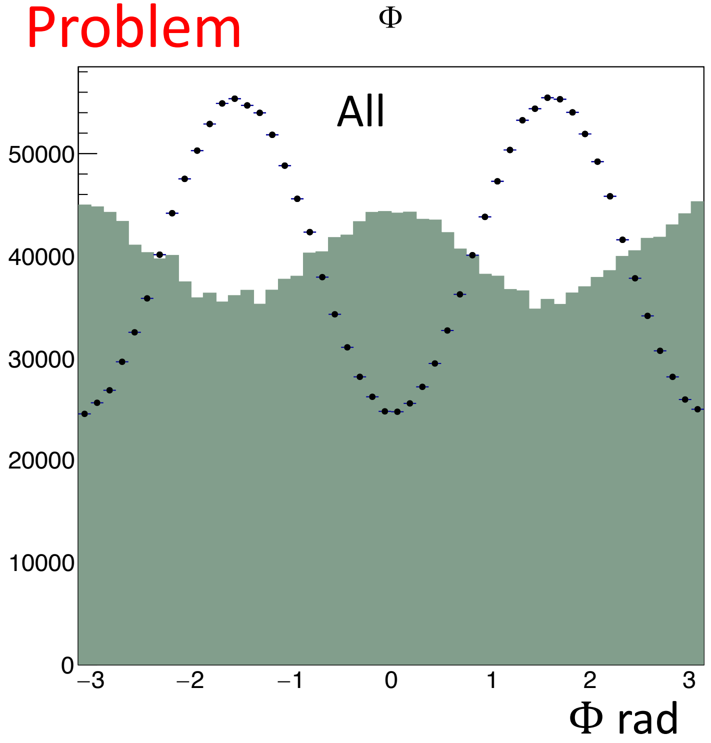
Amplitudes used in fitting are
D0+, D1+, P0+, P1+, G2+

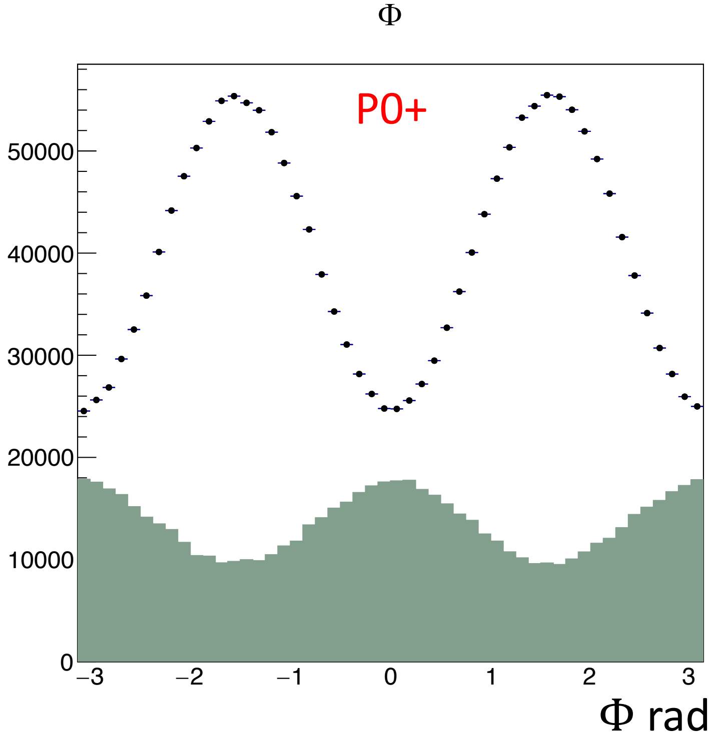


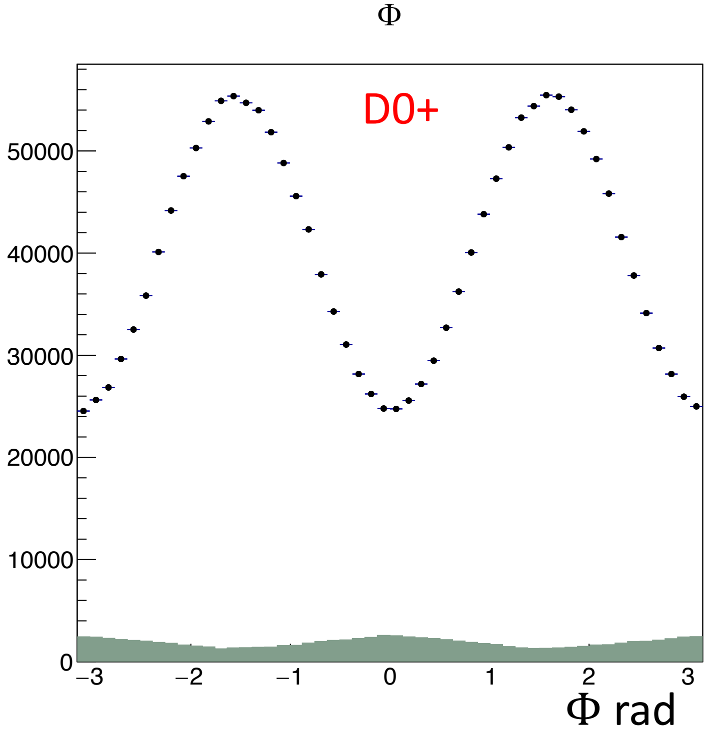
Lab frame

$\vec{y}=\overrightarrow{p_{beam}}\times(-\overrightarrow{p_{recoil}})$
$\vec{x}=\vec{y}\times\overrightarrow{p_{beam}}$
$\vec{z}=\vec{x}\times\vec{y}$
θ_{pol}=1.7 Deg.
$\vec{\epsilon}=(\cos(\theta_{pol}),\ \sin(\theta_{pol}),\ 0)$
$\Phi = \mathrm{arctg}(\vec{y}\cdot\vec{\epsilon},\ \overrightarrow{p_{beam}}\cdot(\vec{\epsilon}\times\vec{y}))$

Problem

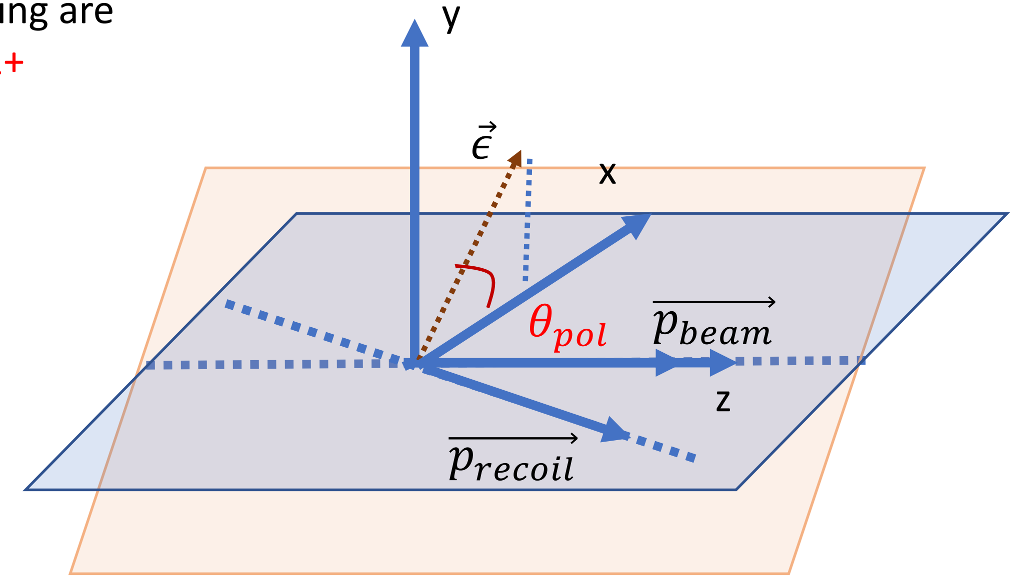






Results with angle between production and polarization planes Φ

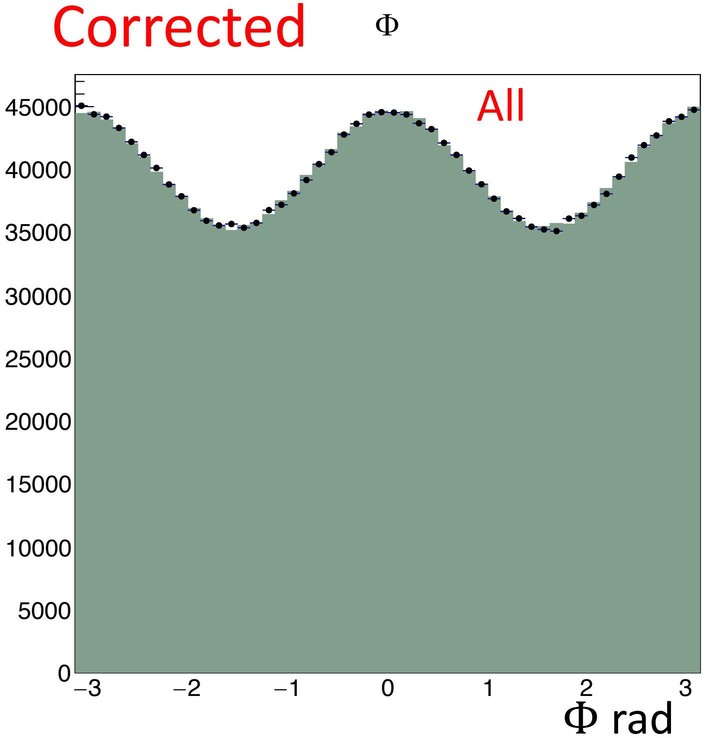
Amplitudes used in fitting are
D0+, D1+, P0+, P1+, G2+

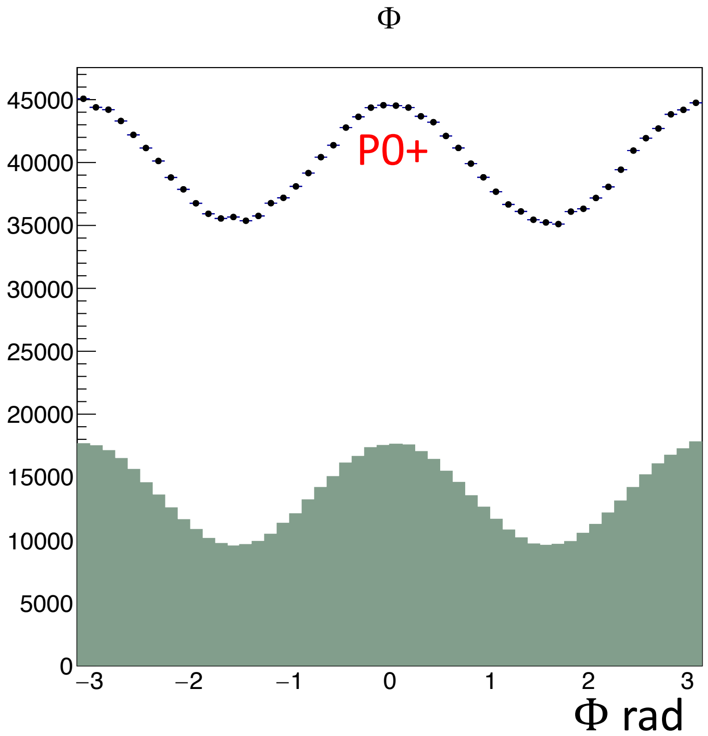


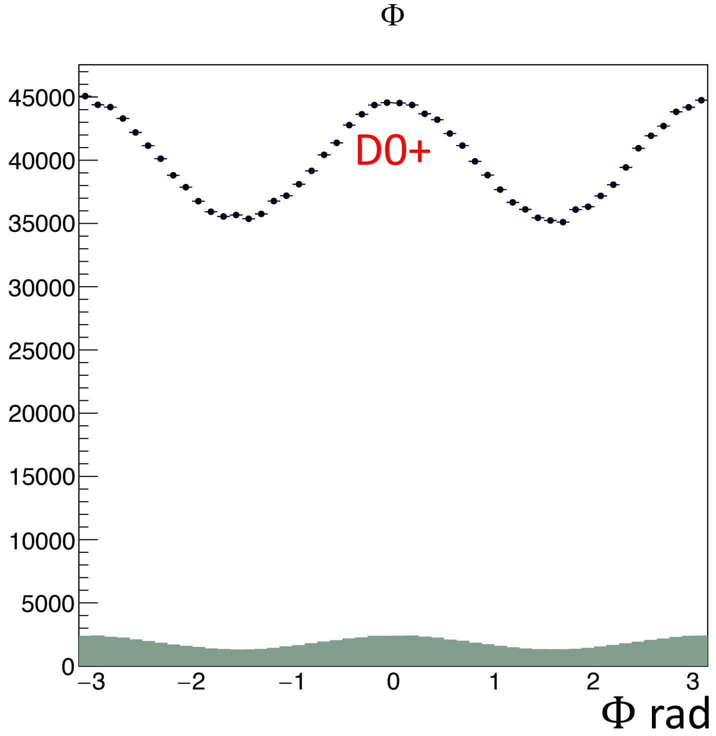
Lab frame

$\vec{y} = \overrightarrow{p_{beam}} \times (-\overrightarrow{p_{recoil}})$
$\vec{x} = \vec{y} \times \overrightarrow{p_{beam}}$
$\vec{z} = \vec{x} \times \vec{y}$
θ_{pol}=1.7 Deg.
$\vec{\epsilon} = (\cos(\theta_{pol}), \sin(\theta_{pol}), 0)$
$\Phi = \mathrm{arctg}(\vec{y} \cdot \vec{\epsilon}, \overrightarrow{p_{beam}} \cdot (\vec{\epsilon} \times \vec{y}))$

Corrected

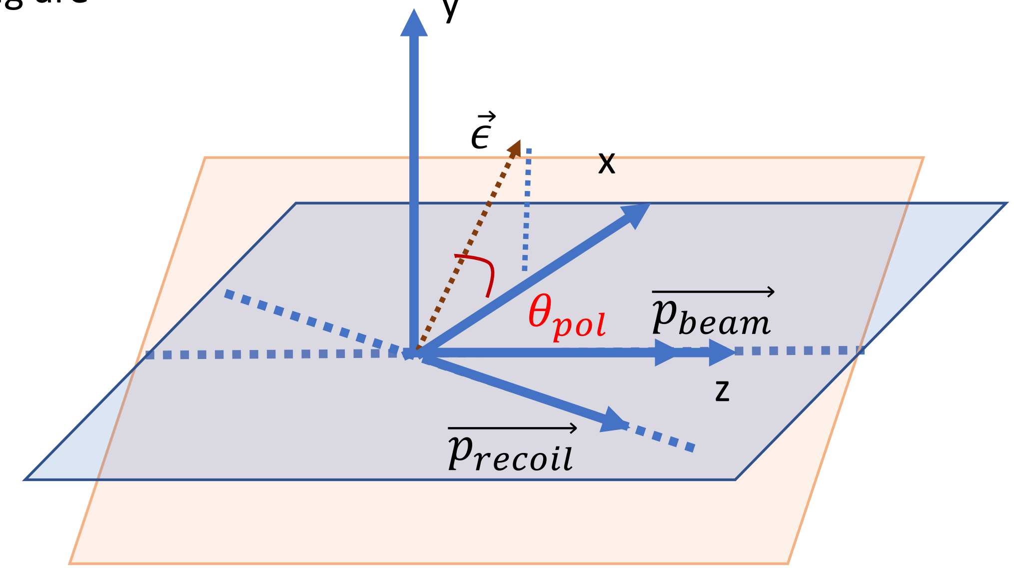






Results with angle between production and polarization planes Φ

Amplitudes used in fitting are
D0+, D1+, P0+, P1+, G2+



Lab frame

$\vec{y} = \overrightarrow{p_{beam}} \times (-\overrightarrow{p_{recoil}})$
$\vec{x} = \vec{y} \times \overrightarrow{p_{beam}}$
$\vec{z} = \vec{x} \times \vec{y}$
θ_{pol}=1.7 Deg.
$\vec{\epsilon} = (\cos(\theta_{pol}),\ \sin(\theta_{pol}),\ 0)$
$\Phi = \mathrm{arctg}(\vec{y} \cdot \vec{\epsilon},\ \overrightarrow{p_{beam}} \cdot (\vec{\epsilon} \times \vec{y}))$

Corrected