Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

GENIE 1p1h

Gabriel Perdue // @gnperdue
Fermi National Accelerator Laboratory // @Fermilab
ECT, Trento, July 2018

Outline

- GENIE's 1p1h model(s)
  - "Nuclear" models for the ground state.
  - Free nucleon model:
    - Form factors
    - Differential cross section algorithms

- *Note*: we will largely ignore the remnant nucleus in this discussion, although of course it is important.
- *Note*: we will also largely ignore hadron transport in this discussion, although that, also, is quite important!

UNIVERSAL NEUTRINO GENERATOR & GLOBAL FIT

Documentation

- Physics and Users Manual contains some detail (non-exhaustive example below):
  - https://arxiv.org/abs/1510.05494
  - Please feel empowered to contribute. (Classic OSS "first contribution" is documentation.)

Quasi-Elastic Scattering: Quasi-elastic scattering (e.g. $\nu_\mu + n \to \mu^- + p$) is modeled using an implementation of the Llewellyn-Smith model [31]. In this model the hadronic weak current is expressed in terms of the most general Lorentz-invariant form factors. Two are set to zero as they violate G-parity. Two vector form factors can be related via CVC to electromagnetic form factors which are measured over a broad range of kinematics in electron elastic scattering experiments. Several different parametrizations of these electromagnetic form factors including Sachs [32], BBA2003 [33] and BBBA2005 [34] models are available with BBBA2005 being the default. Two form factors - the pseudo-scalar and axial vector, remain. The pseudo-scalar form factor is assumed to have the form suggested by the partially conserved axial current (PCAC) hypothesis [31], which leaves the axial form factor $F_A(Q^2)$ as the sole remaining unknown quantity. $F_A(0)$ is well known from measurements of neutron beta decay and the Q^2 dependence of this form factor can only be determined in neutrino experiments and has been the focus of a large amount of experimental work over several decades. In GENIE a dipole form is assumed, with the axial vector mass m_A remaining as the sole free parameter with a default value of 0.99 GeV/c^2.

For nuclear targets, the struck a suppression factor is included from an analytic calculation of the rejection factor in the Fermi Gas model, based on the simple requirement that the momentum of the outgoing nucleon exceed the fermi momentum k_F for the nucleus in question. Typical values of k_F are 0.221 GeV/c for nucleons in ^{12}C, 0.251 GeV/c for protons in ^{56}Fe, and 0.256 GeV/c for neutrons in ^{56}Fe.

Elastic Neutral Current Scattering: Elastic neutral current processes are computed according to the model described by Ahrens et al. [35], where the axial form factor is given by:

$$G_A(Q^2) = \frac{1}{2}\frac{G_A(0)}{(1+Q^2/M_A^2)^2}(1+\eta). \tag{2.1}$$

The adjustable parameter η includes possible isoscalar contributions to the axial current, and the GENIE default value is $\eta = 0.12$. For nuclear targets the same reduction factor described above is used.

Event generators

- GENIE has an internal set of 'Event Generators' that are used to set the kinematics for each event. For 1p1h, there are two - a legacy generator and a more modern one:

```
<param_set name="QEL-CC">
  <param type="string" name="VldContext"> </param>
  <param type="int"    name="NModules">   12                                           </param>
  <param type="alg"    name="Module-0">   genie::InitialStateAppender/Default          </param>
  <param type="alg"    name="Module-1">   genie::VertexGenerator/Default               </param>
  <param type="alg"    name="Module-2">   genie::FermiMover/Default                    </param>
  <param type="alg"    name="Module-3">   genie::QELKinematicsGenerator/CC-Default     </param>
  <param type="alg"    name="Module-4">   genie::QELPrimaryLeptonGenerator/Default     </param>
  <param type="alg"    name="Module-5">   genie::QELHadronicSystemGenerator/Default    </param>
  <param type="alg"    name="Module-6">   genie::PauliBlocker/Default                  </param>
  <param type="alg"    name="Module-7">   genie::UnstableParticleDecayer/BeforeHadronTransport </param>
  <param type="alg"    name="Module-8">   genie::NucDeExcitationSim/Default            </param>
  <param type="alg"    name="Module-9">   genie::HadronTransporter/Default             </param>
  <param type="alg"    name="Module-10">  genie::NucBindEnergyAggregator/Default       </param>
  <param type="alg"    name="Module-11">  genie::UnstableParticleDecayer/AfterHadronTransport  </param>
  <param type="alg"    name="ILstGen">    genie::QELInteractionListGenerator/CC-Default </param>
</param_set>
```

From `$GENIE/config/G00_00a/EventGenerator.xml`

Event generators

- The new event generator does not invoke `FermiMover` as a separate module (ground state invocation). We now re-sample the ground state on every accept-reject loop when choosing kinematics:

```
<param_set name="QEL-CC">
   <param type="string" name="VldContext"> </param>
   <param type="int"    name="NModules">   9                                         </param>
   <param type="alg"    name="Module-0">   genie::InitialStateAppender/Default        </param>
   <param type="alg"    name="Module-1">   genie::VertexGenerator/Default             </param>
   <param type="alg"    name="Module-2">   genie::QELEventGenerator/Default           </param>
   <param type="alg"    name="Module-3">   genie::PauliBlocker/Default                </param>
   <param type="alg"    name="Module-4">   genie::UnstableParticleDecayer/BeforeHadronTransport </param>
   <param type="alg"    name="Module-5">   genie::NucDeExcitationSim/Default          </param>
   <param type="alg"    name="Module-6">   genie::HadronTransporter/Default           </param>
   <param type="alg"    name="Module-7">   genie::NucBindEnergyAggregator/Default     </param>
   <param type="alg"    name="Module-8">   genie::UnstableParticleDecayer/AfterHadronTransport  </param>
   <param type="alg"    name="ILstGen">    genie::QELInteractionListGenerator/CC-Default </param>
 </param_set>
```

From `$GENIE/config/EventGenerator.xml`

Interaction vertex

- Users may optionally use a uniform distribution (in x, y, z independently, each between -R and R), but by default the vertex is chosen from a realistic (and simple) density profile.
  - Modified Gaussian for A < 20, 2-parameter Woods-Saxon for the rest.
- Coherent and neutrino-electron scattering events are set to the nuclear boundary.

```
if(realistic) {
  double ymax = -1;
  double rmax = 3*R;
  double dr   = R/40.;
  for(double r = 0; r < rmax; r+=dr) {
     ymax = TMath::Max(ymax, r*r * utils::nuclear::Density(r,(int)A));
  }
  ymax *= 1.2;
```

From `$GENIE/src/Physics/Common/VertexGenerator.cxx`
`$GENIE/src/Physics/NuclearState/NuclearUtils.cxx`

```
if(realistic) { ...

   while(1) {
     double r = rmax * rnd->RndFsi().Rndm();
     double t = ymax * rnd->RndFsi().Rndm();
     double y = r*r * utils::nuclear::Density(r,(int)A);
     bool accept = (t < y);
     if(accept) {
       double phi      = 2*kPi * rnd->RndFsi().Rndm();
       double cosphi   = TMath::Cos(phi);
       double sinphi   = TMath::Sin(phi);
       double costheta = -1 + 2 * rnd->RndFsi().Rndm();
       double sintheta = TMath::Sqrt(1-costheta*costheta);
       vtx.SetX(r*sintheta*cosphi);
       vtx.SetY(r*sintheta*sinphi);
       vtx.SetZ(r*costheta);
```

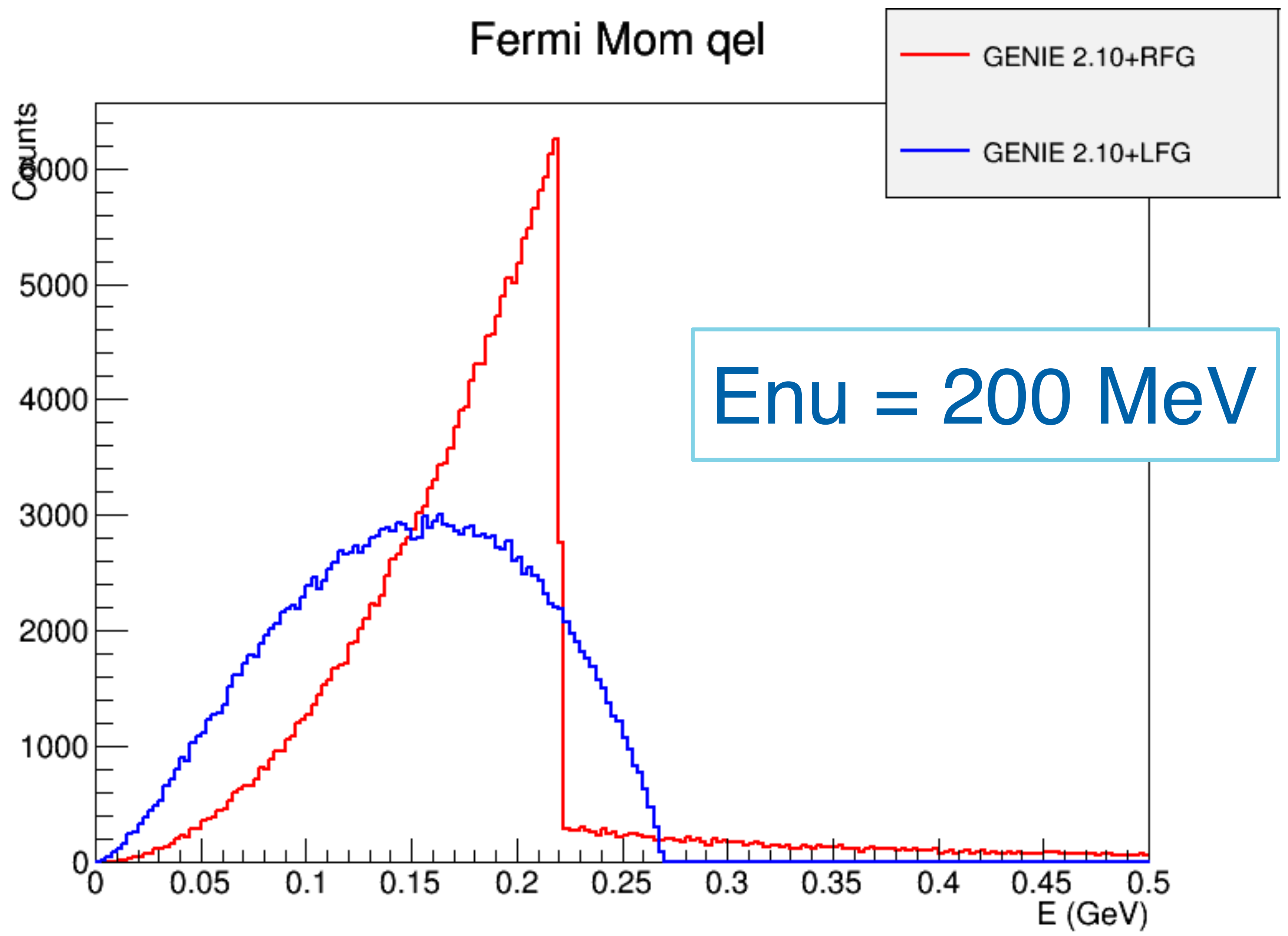
Nuclear ground states

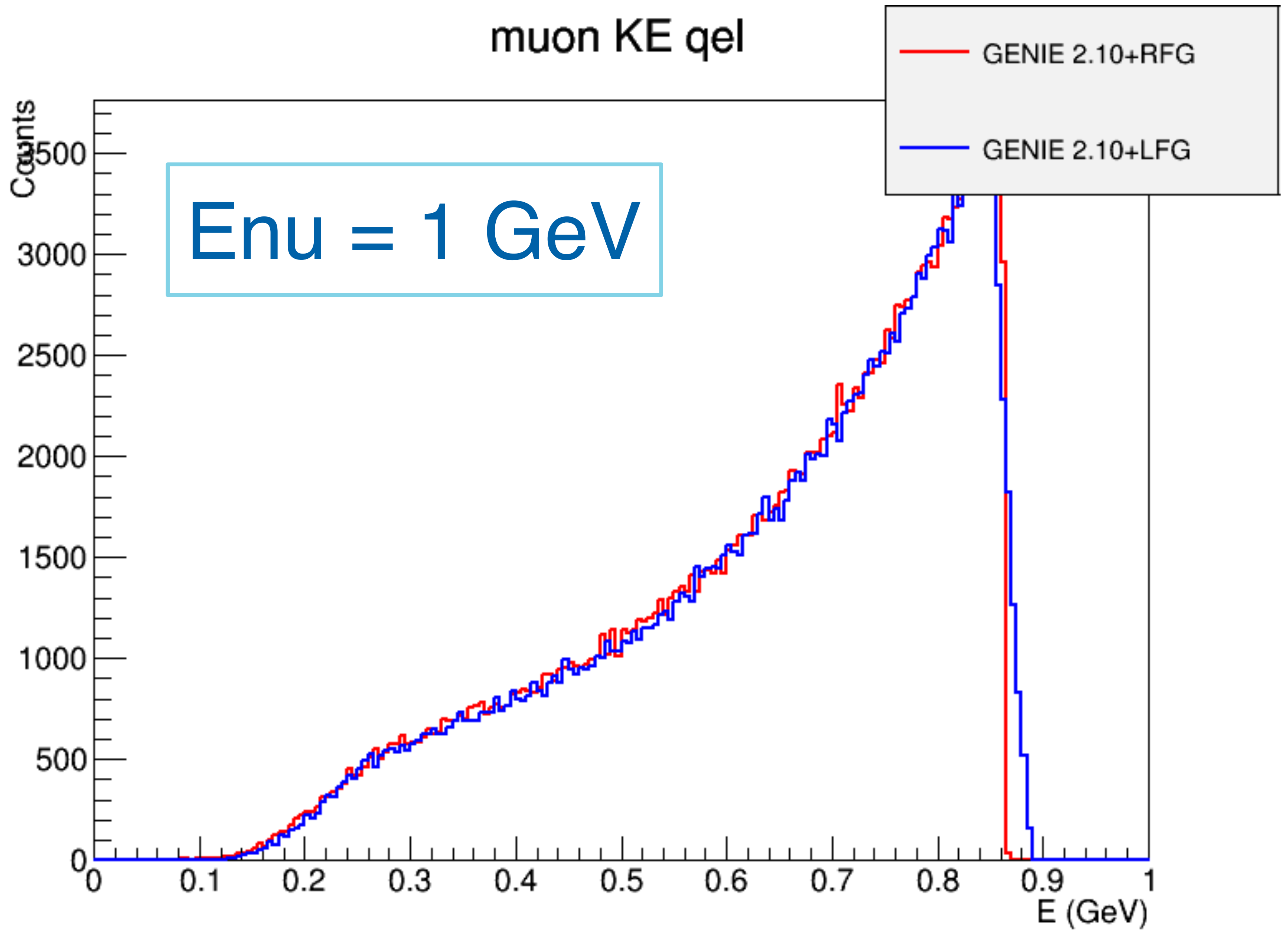
- Effective Spectral Function (Bodek et al)
- Local Fermi Gas
- Relativistic Fermi Gas with Bodek-Ritchie

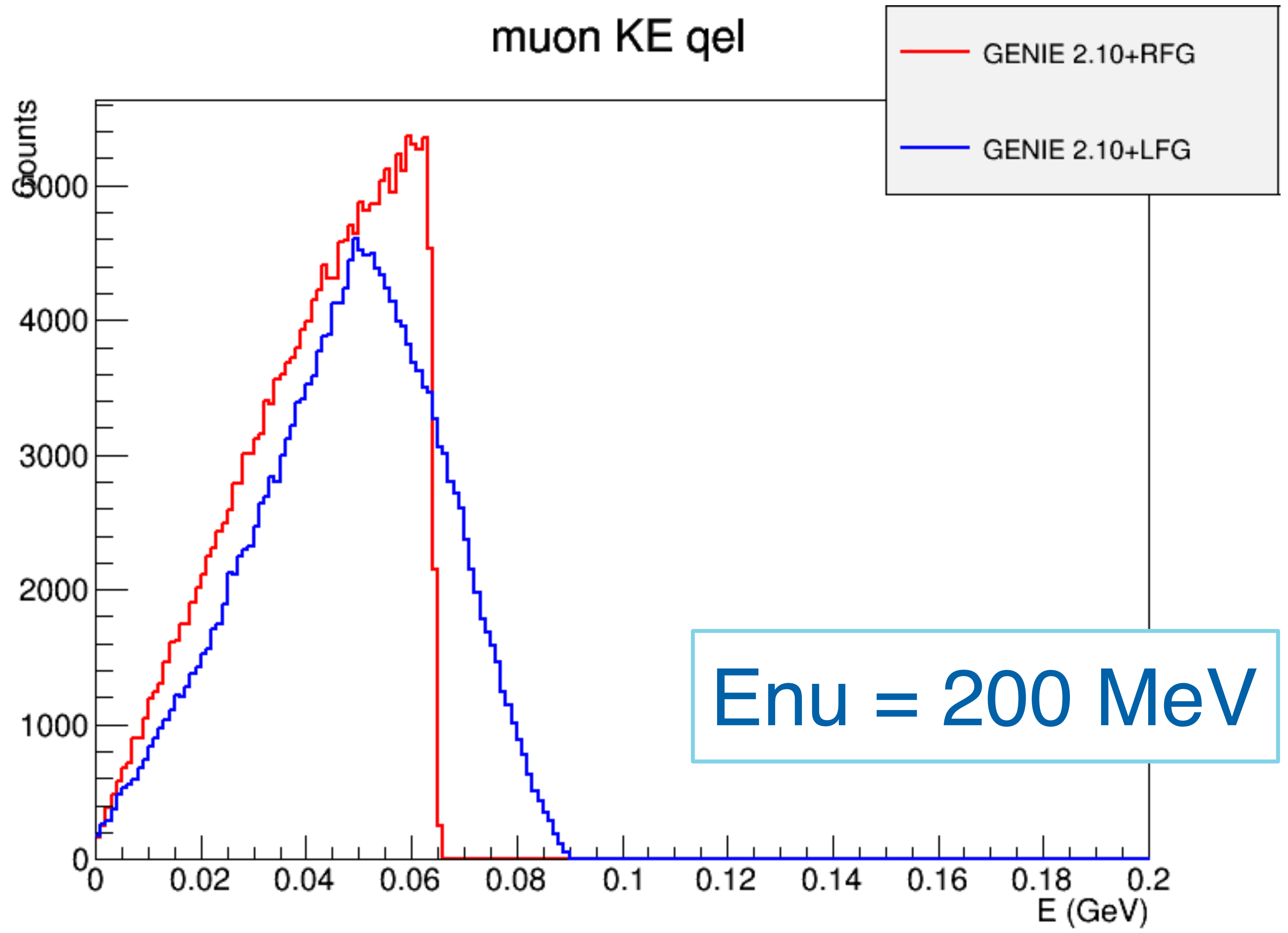
- Initial interaction is often with `FermiMover` class - built to give the target nucleon an initial momentum.
- `FermiMover` has a pointer to a `fNuclModel` which is responsible for actually computing the momentum (one of the LFG, RFG, etc. from above) and computing a removal energy.
- Fermi momentum (kF) drawn from a table or a PDF, and then used to compute a PDF for the initial nucleon momentum.
- `FermiMover` can also eject an extra nucleon (esp. for Eff. Spec. Func.).
- Finally, `FermiMover` will fix the remnant nucleus recoil.

Local Fermi Gas

- In the LFG model, the Fermi momentum is a function of position in the nucleus.
- Target is C12 in the plots.

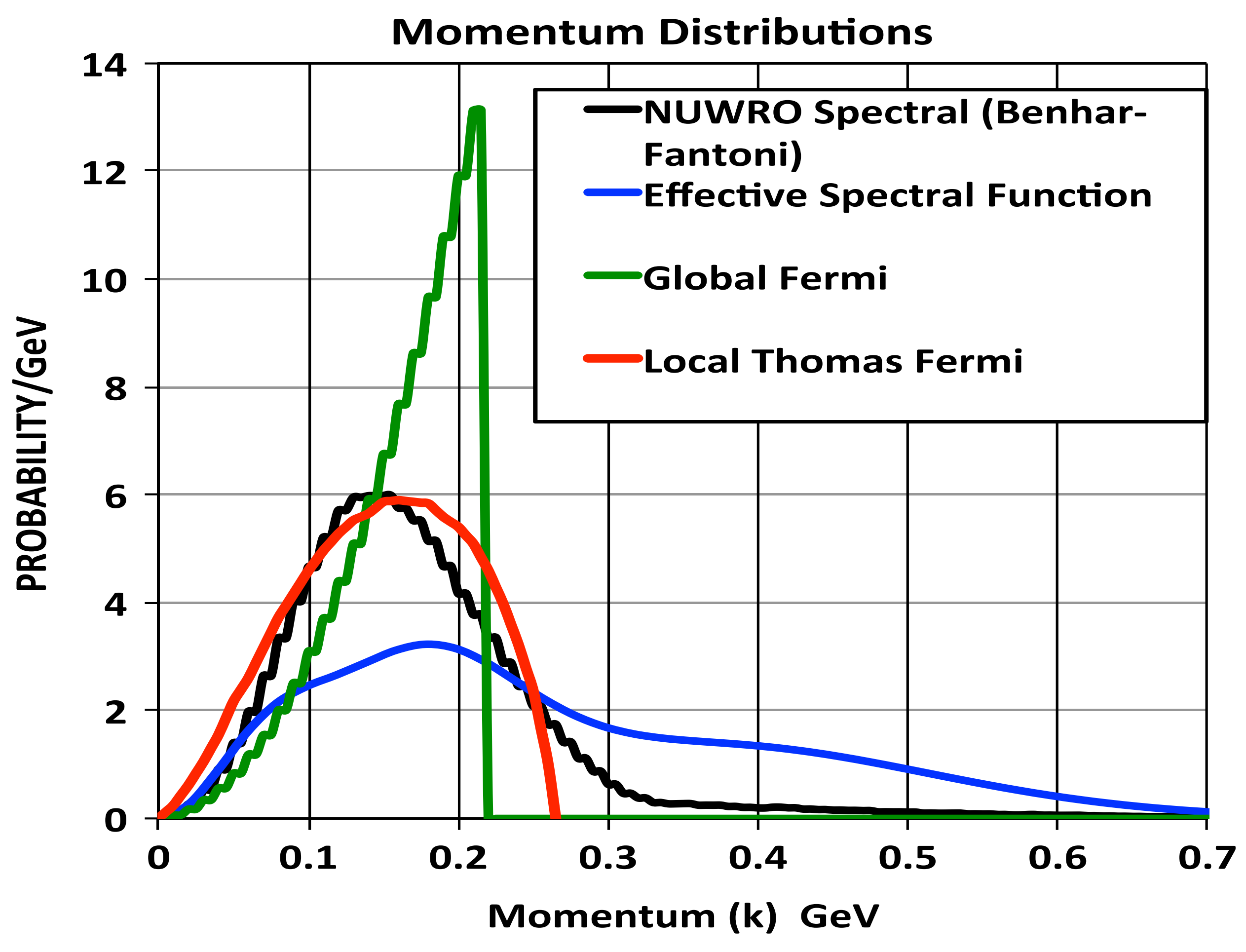




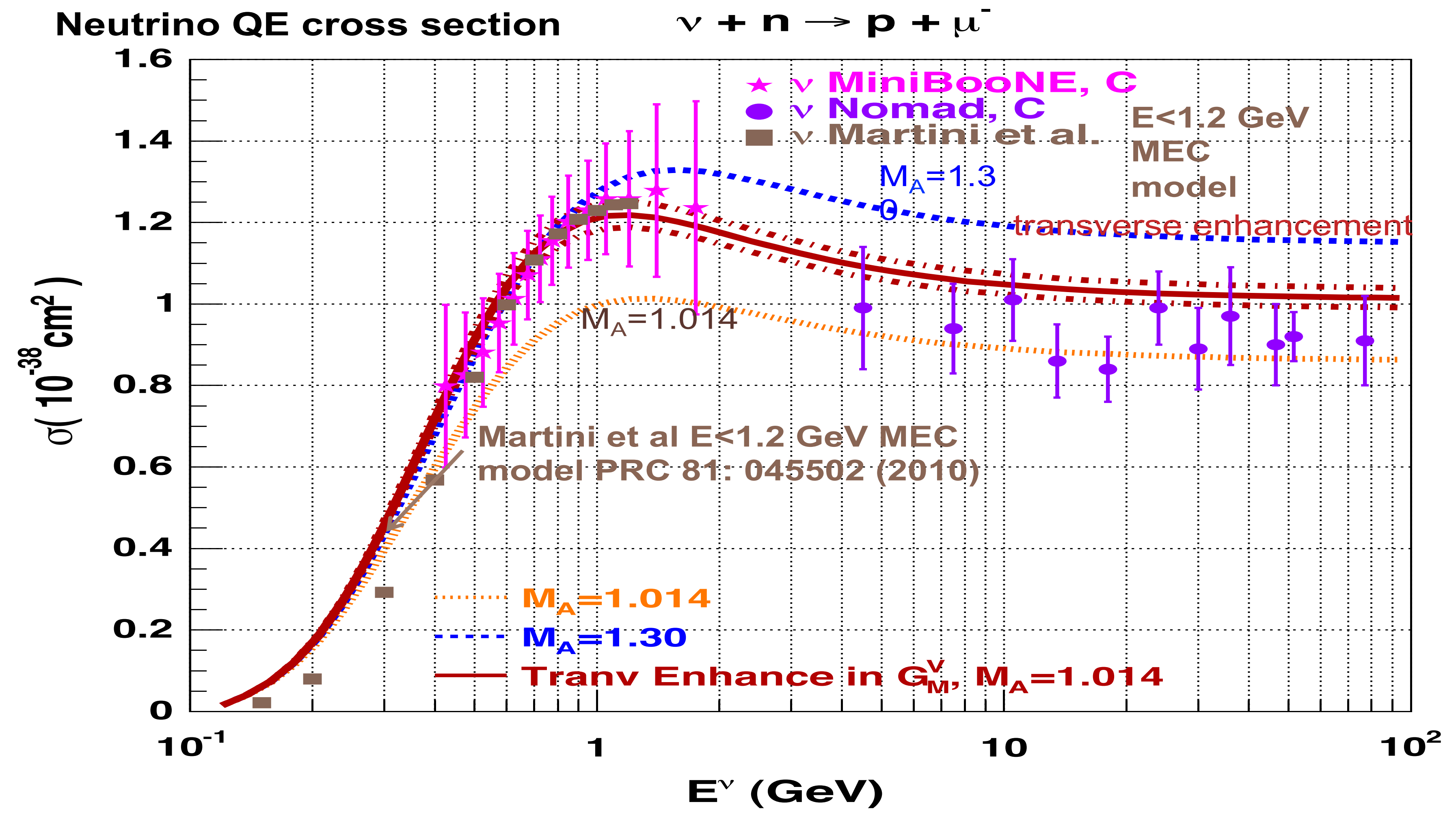


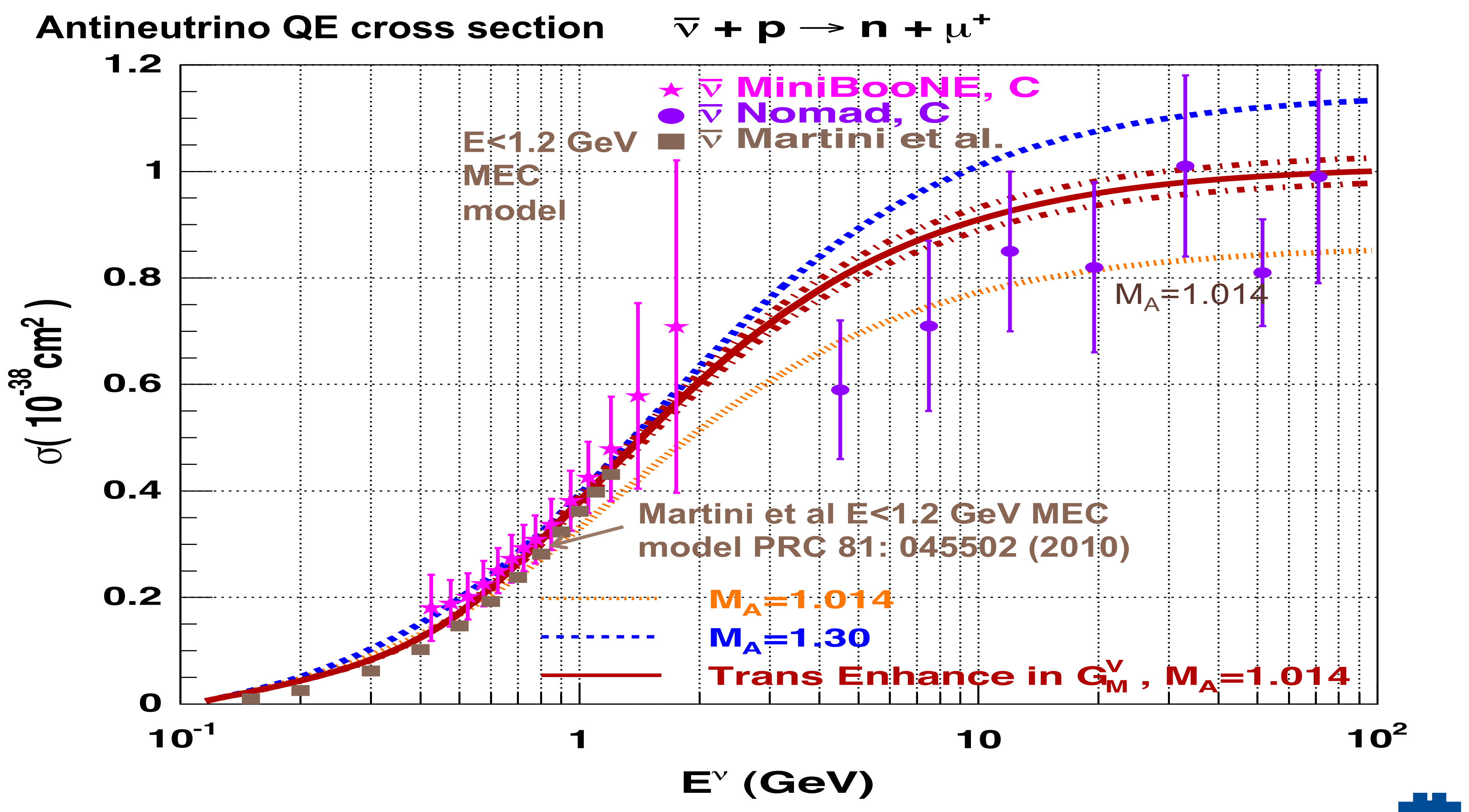
Effective Spectral Functions

- The Effective Spectral Function model combines a superscaling formalism together with hadronic energy sharing prescription to form a complete QE model.
  - An eight parameter spectral function is fit to the superscaling function extracted from electron scattering data (plus two parameters for binding energy and 2p2h fraction).
- Implemented by B. Coopersmith (also implemented Transverse Enhancement Model)

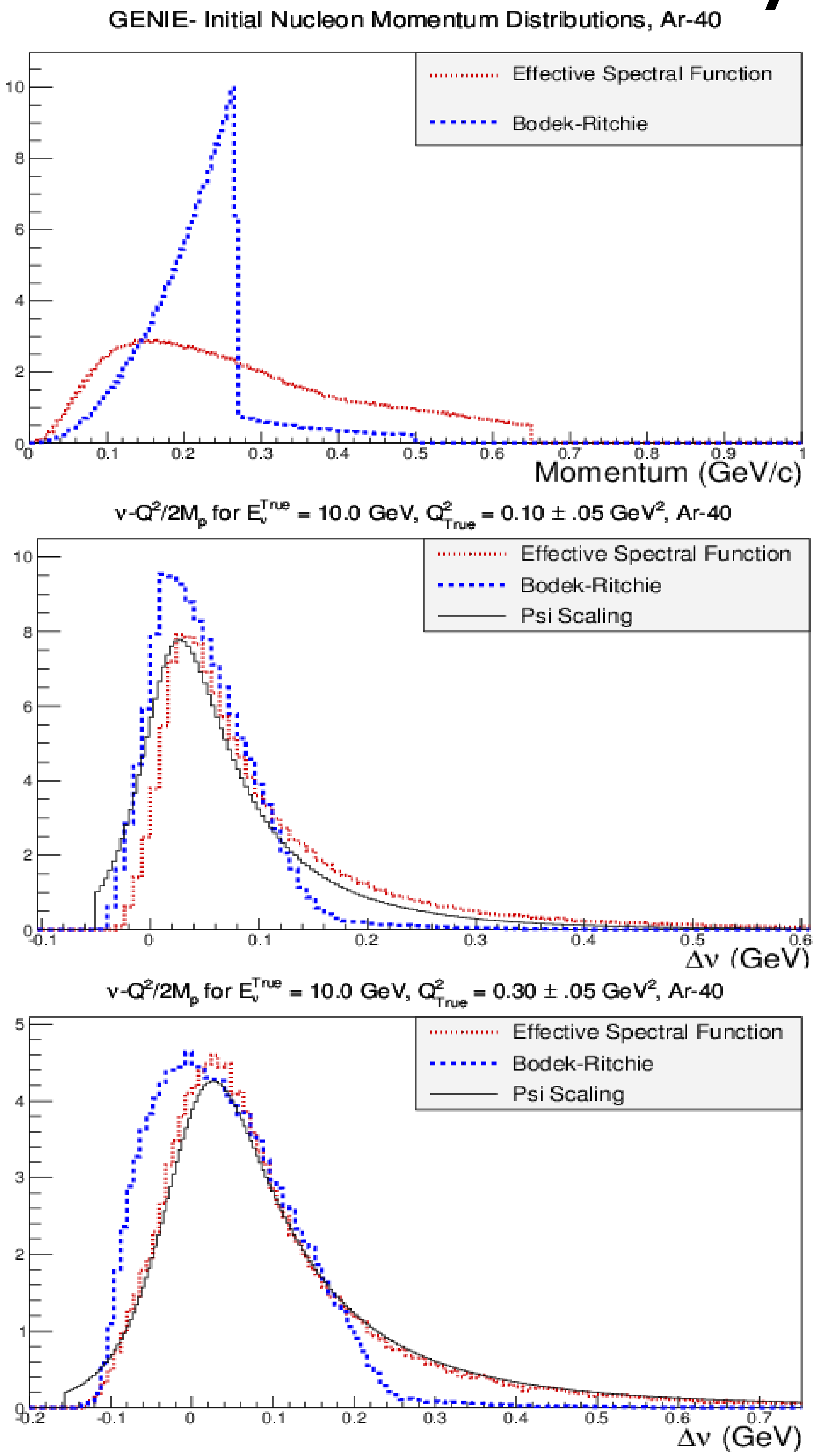


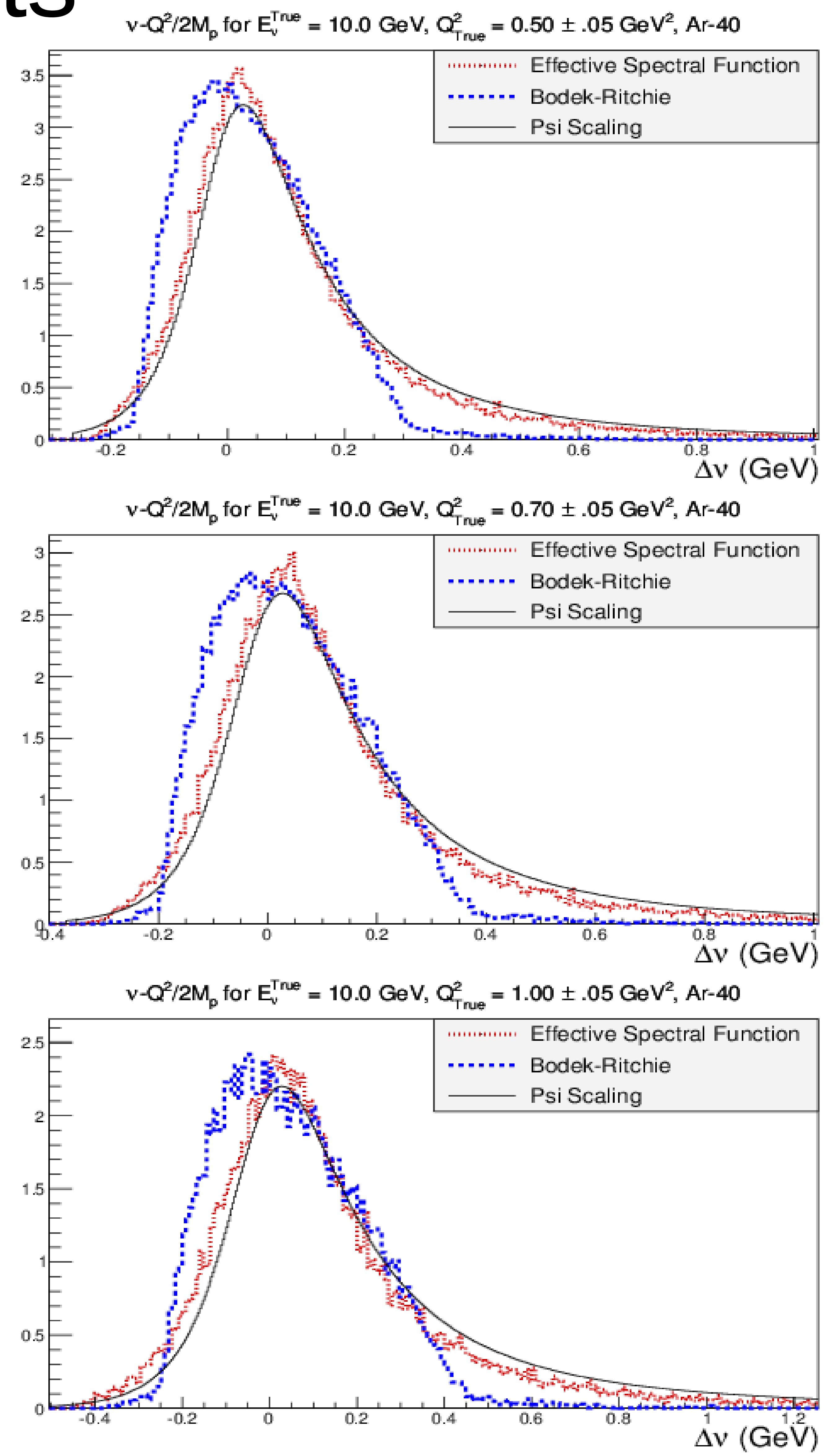
Bodek, Christy, Coopersmith EPJ C (2014) 74:3091





Ar-40 Results





Brian Coopersmith- University of Rochester

Back to event generators: legacy

- We assume a free nucleon cross section (even in nuclear targets) to avoid double-counting nuclear effects (also, if a QE event is Pauli-blocked, we re-throw a new QE event).
- Throw accept-reject against the partial differential cross section in Q^2. The maximum Q^2 is a function of W (nucleon mass here).
- Once the kinematics are set, the lepton is computed using Q^2, y, and Enu (and the lepton mass).
- The recoil baryon is computed by four-momentum conservation (neutrino + struck nucleon - lepton).

```
$GENIE/src/Physics/QuasiElastic/EventGen/QELKinematicsGenerator.cxx
$GENIE/src/Physics/QuasiElastic/EventGen/QELHadronicSystemGenerator.cxx
```

Back to event generators: modern

- Written with `NievesQELPXSec` algorithm in mind.
- Simultaneously choose Fermi momentum, binding energy, and Q^2 (don't re-throw hit nucleon radius in accept-reject loop).
  - Physically more correct treatment regardless of xsec algorithm.
    - Will eventually be used everywhere, but the code runs much slower with this approach, so we are thinking about the trade-offs involved and how to be reasonably efficient.

`$GENIE/src/Physics/QuasiElastic/EventGen/QELEventGenerator.cxx`

Cross section algorithms

- `LwlynSmithQELCCPXSec`
- `NievesQELPXSec`
- `AhrensNCELPXSec`
- `RosenbluthPXSec` (EM)
- SmithMonizQELPXSec*

*[1] R.A.Smith and E.J.Moniz, Nuclear Physics B43, (1972) 605-622
[2] K.S. Kuzmin, V.V. Lyubushkin, V.A.Naumov Eur. Phys. J. C54, (2008) 517-538

- Cross section integration performed with `QELXSec`, configured as:

```
<param_set name="Default">
  <param type="string" name = "gsl-integration-type">          adaptive  </param>
  <param type="int"    name = "gsl-max-size-of-subintervals">     40000  </param>
  <param type="double" name = "gsl-relative-tolerance">           0.001  </param>
  <param type="int"    name = "gsl-rule">                             3  </param>
</param_set>
```

Form factor models

- `ZExpAxialFormFactorModel`
- `DipoleAxialFormFactorModel`

```
<param_set name="Default">

  <param type="string"   name="CommandParam"> QuasiElastic  </param>

  <param type="bool"   name="QEL-Q4limit"> true   </param>
  <param type="int"    name="QEL-Kmax">    4      </param>

  <param type="double" name="QEL-T0">     -0.28   </param>
  <param type="double" name="QEL-Tcut">    0.1764 </param> <!-- 9*m_pi^2 -->

  <param type="double" name="QEL-Z_A1">    2.30   </param>
  <param type="double" name="QEL-Z_A2">   -0.6    </param>
  <param type="double" name="QEL-Z_A3">   -3.8    </param>
  <param type="double" name="QEL-Z_A4">    2.3    </param>
  <!-- more factors can be added, if necessary according to Kmax -->

</param_set>
```

```
<param_set name="HistoricalFit">
  <param type="double" name="QEL-Ma"> 0.990 </param>
  <param type="double" name="QEL-FA0"> -1.2670 </param>
</param_set>
```

(Note by A. Meyer)

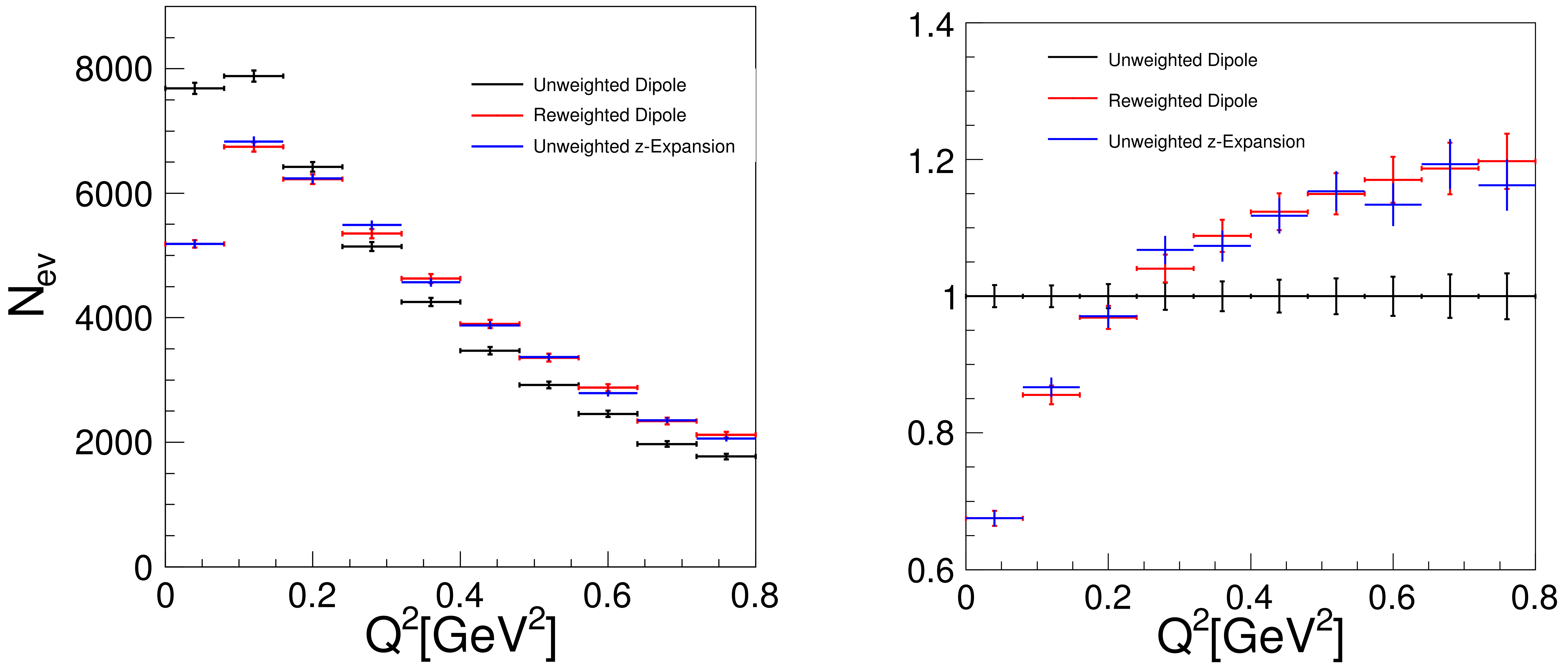


Figure 2: A nominal dipole event sample which has been reweighted to a z-expansion sample. The dipole Monte Carlo sample is represented in black, with statistical error bars. The reweighted dipole sample is shown in red, and the independent sample with z expansion values is shown in blue. The left plot shows the raw number of events in each bin for a 50k event sample of pure CCQE, and the right plot shows the events normalized by the nominal sample. The agreement between red and blue is a validation of the reweighting procedure. The study was done using a carbon target at 1 GeV.

LwlynSmithQELCCPXSec

```
double F1V   = fFormFactors.F1V();
double xiF2V = fFormFactors.xiF2V();
double FA    = fFormFactors.FA();
double Fp    = fFormFactors.Fp();

//...

// Compute free nucleon differential cross section
double A = (0.25*(ml2-q2)/M2) * (
       (4-q2_M2)*FA2 - (4+q2_M2)*F1V2 - q2_M2*xiF2V2*(1+0.25*q2_M2)
            -4*q2_M2*F1V*xiF2V - (ml2/M2)*(
             (F1V2+xiF2V2+2*F1V*xiF2V)+(FA2+4*Fp2+4*FA*Fp)+(q2_M2-4)*Fp2));
double B = -1 * q2_M2 * FA*(F1V+xiF2V);
double C = 0.25*(FA2 + F1V2 - 0.25*q2_M2*xiF2V2);

double xsec = Gfactor * (A + sign*B*s_u/M2 + C*s_u*s_u/M4);
```

$GENIE/src/Physics/QuasiElastic/XSection/LwlynSmithQELCCPXSec.cxx

LwlynSmithQELCCPXSec

```
//____________________________________________________________________________
void LwlynSmithQELCCPXSec::LoadConfig(void)
{
  // Cross section scaling factor
  GetParamDef( "QEL-CC-XSecScale", fXSecScale, 1. ) ;

  double thc ;
  GetParam( "CabibboAngle", thc ) ;
  fCos8c2 = TMath::Power(TMath::Cos(thc), 2);

   // load QEL form factors model
  fFormFactorsModel = dynamic_cast<const QELFormFactorsModelI *> (
                                             this->SubAlg("FormFactorsAlg"));
  assert(fFormFactorsModel);
  fFormFactors.SetModel(fFormFactorsModel); // <-- attach algorithm

   // load XSec Integrator
  fXSecIntegrator =
      dynamic_cast<const XSecIntegratorI *> (this->SubAlg("XSec-Integrator"));
  assert(fXSecIntegrator);

  // Get nuclear model for use in Integral()
  RgKey nuclkey = "IntegralNuclearModel";
  fNuclModel = dynamic_cast<const NuclearModelI *> (this->SubAlg(nuclkey));
  assert(fNuclModel);

...
```

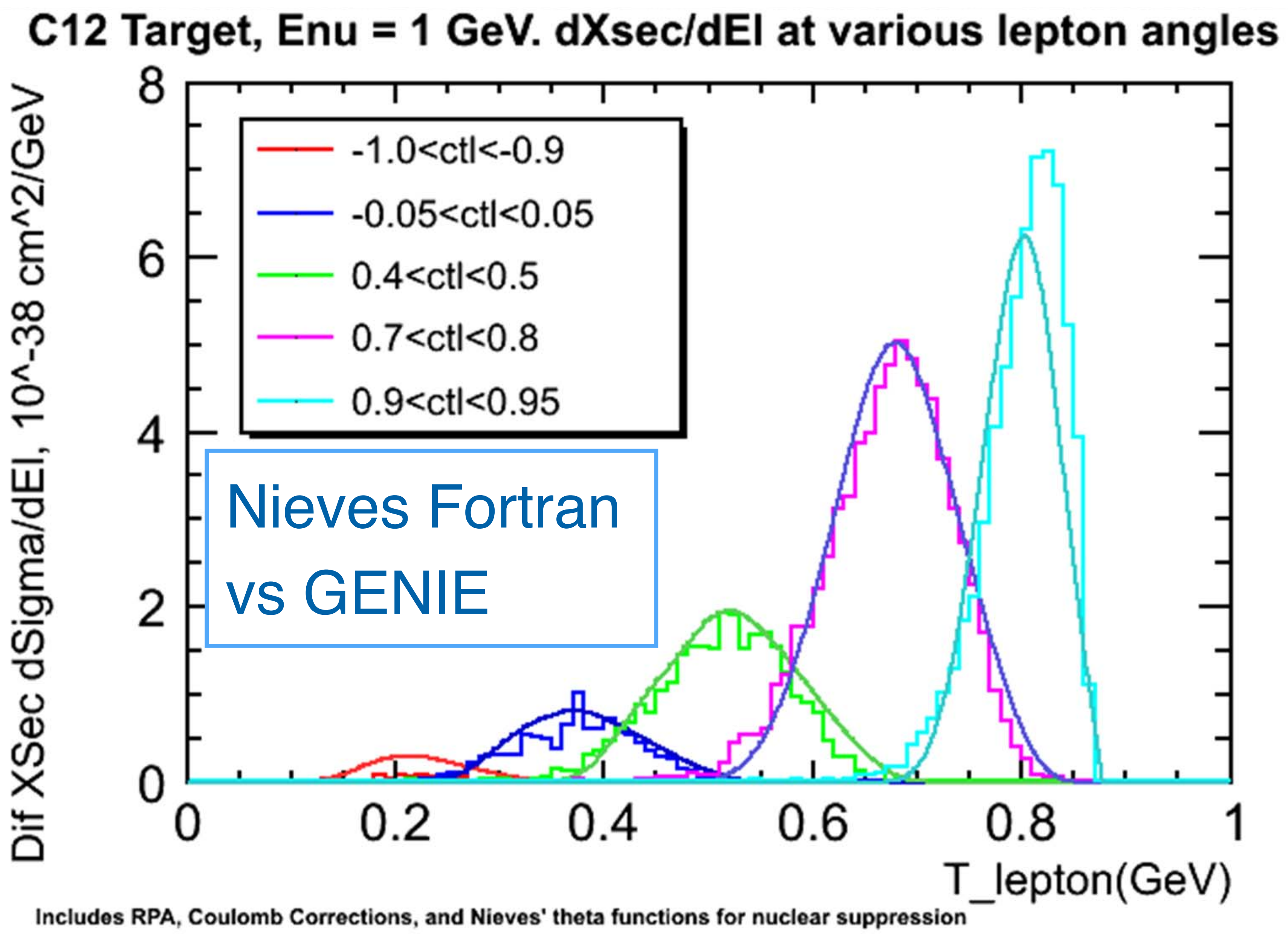
LwlynSmithQELCCPXSec

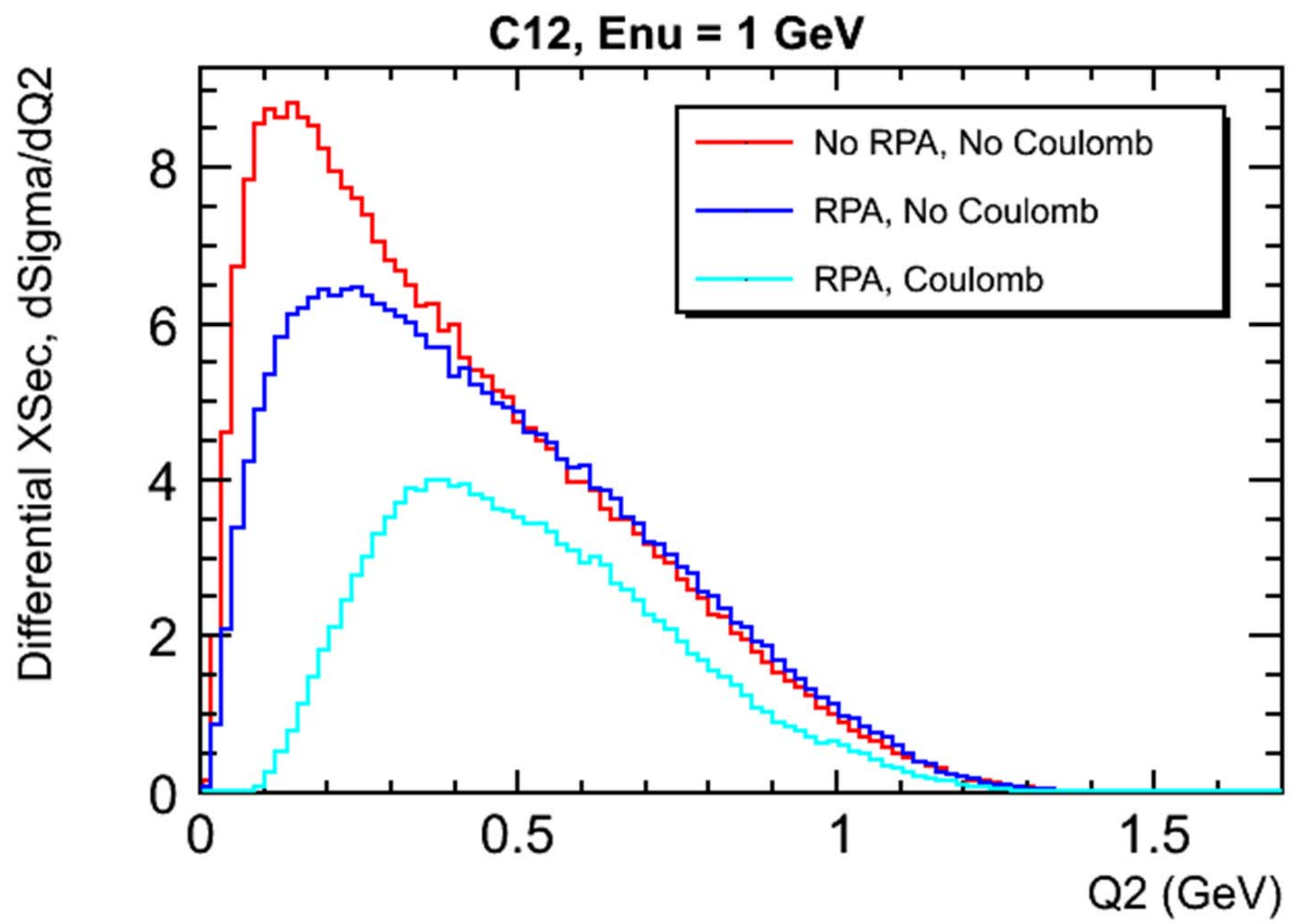
```
...

  fLFG = fNuclModel->ModelType(Target()) == kNucmLocalFermiGas;

  bool average_over_nuc_mom ;
  GetParamDef( "IntegralAverageOverNucleonMomentum", average_over_nuc_mom, false ) ;
  // Always average over initial nucleons if the nuclear model is LFG
  fDoAvgOverNucleonMomentum = fLFG || average_over_nuc_mom ;

  fEnergyCutOff = 0.;

  if(fDoAvgOverNucleonMomentum) {
    // Get averaging cutoff energy
     GetParamDef("IntegralNuclearInfluenceCutoffEnergy", fEnergyCutOff, 2.0 ) ;
  }
}
```

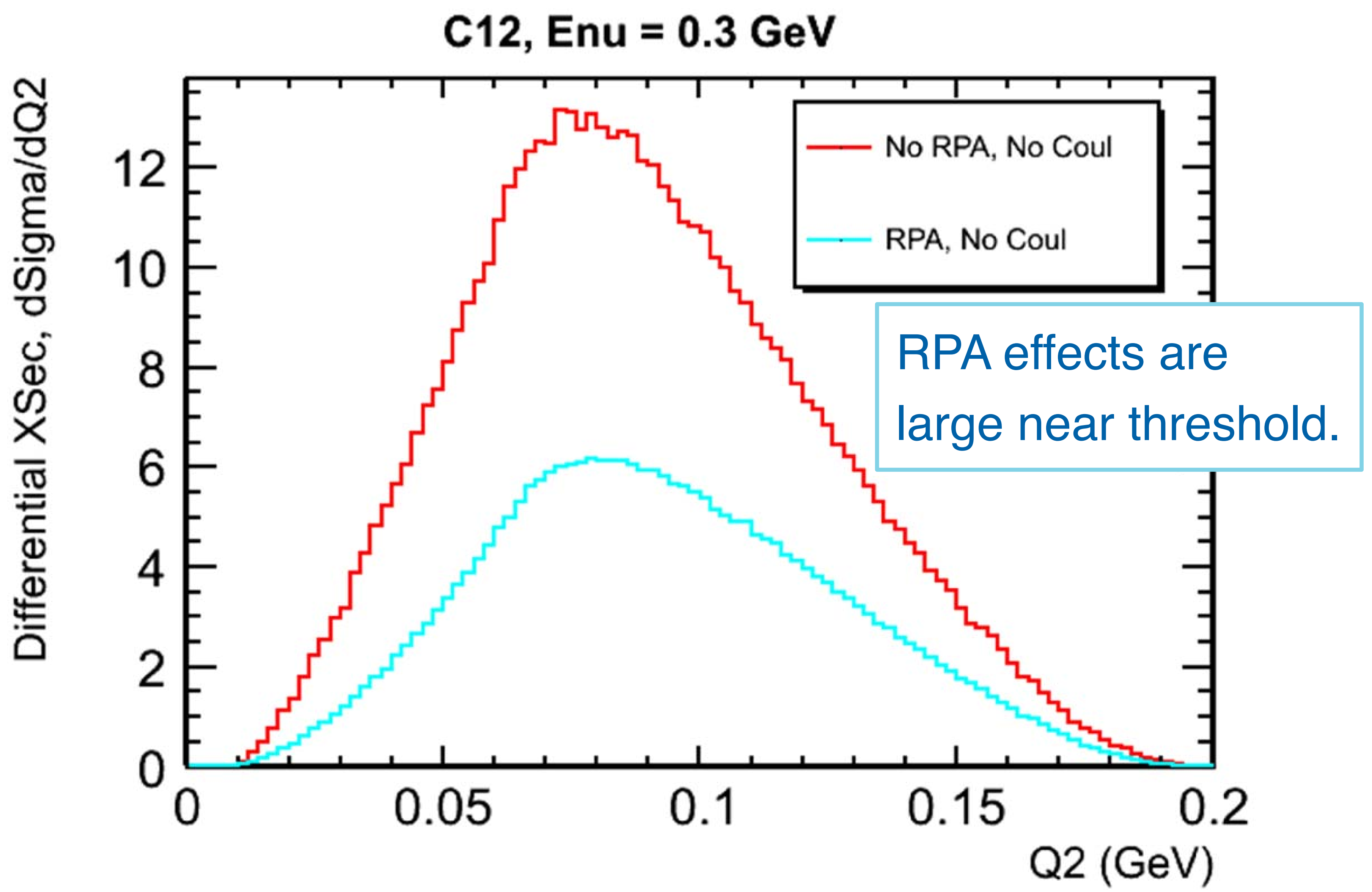
`NievesQELCCPXSec`

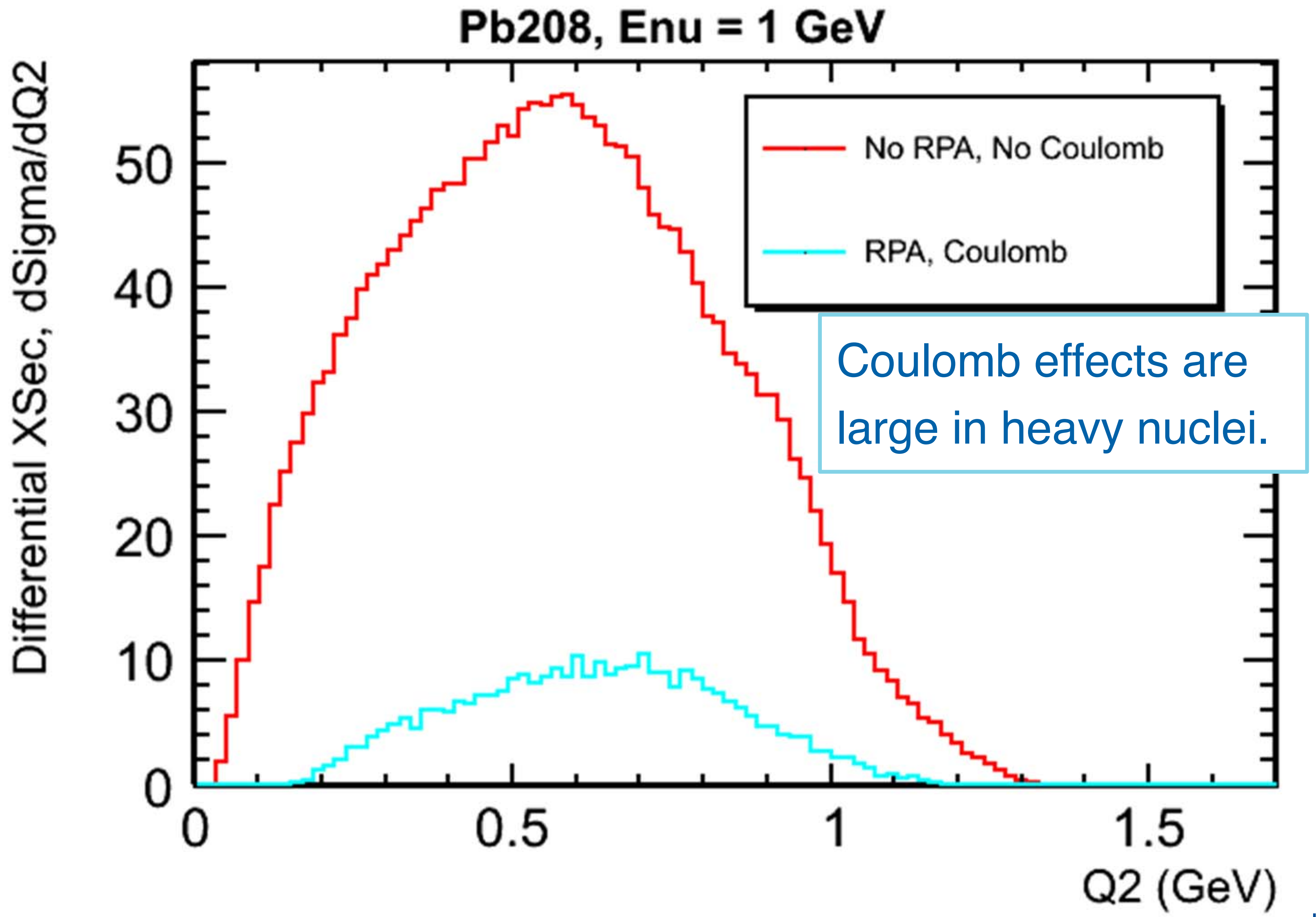
- Physical Review C 70, 055503 (2004)
- Meant to sit on top of LFG (LFG added for this model).
  - Some of the nuclear physics is baked directly into the cross section algorithm.
  - Code will run with RFG.
- New event generator algorithm meant to be paired with this model (developed together).
- Coulomb corrections.
- RPA.
  - Substantially more complex set of calculations than in `LwlynSmithQELCCPXSec`.

`$GENIE/src/Physics/QuasiElastic/XSection/NievesQELCCPXSec.cxx`


C12 Target, Enu = 1 GeV. dXsec/dEl at various lepton angles
Dif XSec dSigma/dEl, 10^-38 cm^2/GeV
T_lepton(GeV)
-1.0<ctl<-0.9
-0.05<ctl<0.05
0.4<ctl<0.5
0.7<ctl<0.8
0.9<ctl<0.95
Nieves Fortran
vs GENIE
Includes RPA, Coulomb Corrections, and Nieves' theta functions for nuclear suppression


C12, Enu = 1 GeV
Differential XSec, dSigma/dQ2
Q2 (GeV)
No RPA, No Coulomb
RPA, No Coulomb
RPA, Coulomb


C12, Enu = 0.3 GeV
Differential XSec, dSigma/dQ2
Q2 (GeV)
No RPA, No Coul
RPA, No Coul
RPA effects are
large near threshold.


Pb208, Enu = 1 GeV
Differential XSec, dSigma/dQ2
Q2 (GeV)
No RPA, No Coulomb
RPA, Coulomb
Coulomb effects are
large in heavy nuclei.

NievesQELCCPXSec

```
//____________________________________________________________________________
void NievesQELCCPXSec::LoadConfig(void)
{
  double thc;
  GetParam( "CabibboAngle", thc ) ;
  fCos8c2 = TMath::Power(TMath::Cos(thc), 2);

  // Cross section scaling factor
  GetParam( "QEL-CC-XSecScale", fXSecScale ) ;

  // hbarc for unit conversion, GeV*fm
  fhbarc = kLightSpeed*kPlankConstant/genie::units::fermi;

   // load QEL form factors model
  fFormFactorsModel = dynamic_cast<const QELFormFactorsModelI *> (
                                           this->SubAlg("FormFactorsAlg"));
  assert(fFormFactorsModel);
  fFormFactors.SetModel(fFormFactorsModel); // <-- attach algorithm

   // load XSec Integrator
  fXSecIntegrator =
      dynamic_cast<const XSecIntegratorI *> (this->SubAlg("XSec-Integrator"));
  assert(fXSecIntegrator);

...
```

$GENIE/src/Physics/QuasiElastic/XSection/NievesQELCCPXSec.cxx

NievesQELCCPXSec

```
...

  // Load settings for RPA and Coulomb effects

  // RPA corrections will not effect a free nucleon
  GetParamDef("RPA", fRPA, true ) ;

  // Coulomb Correction- adds a correction factor, and alters outgoing lepton
  // 3-momentum magnitude (but not direction)
  // Correction only becomes sizeable near threshold and/or for heavy nuclei
  GetParamDef( "Coulomb", fCoulomb, true ) ;

  LOG("Nieves",pNOTICE) << "RPA=" << fRPA << ", useCoulomb=" << fCoulomb;

  // Get nuclear model for use in Integral()
  RgKey nuclkey = "IntegralNuclearModel";
  fNuclModel = dynamic_cast<const NuclearModelI *> (this->SubAlg(nuclkey));
  assert(fNuclModel);

...
```

$GENIE/src/Physics/QuasiElastic/XSection/NievesQELCCPXSec.cxx

NievesQELCCPXSec

```
...

 // Check if the model is a local Fermi gas
 fLFG = fNuclModel->ModelType(Target()) == kNucmLocalFermiGas;

 if(!fLFG){
     // get the Fermi momentum table for relativistic Fermi gas
     GetParam( "FermiMomentumTable", fKFTableName ) ;

     fKFTable = 0;
     FermiMomentumTablePool * kftp = FermiMomentumTablePool::Instance();
     fKFTable = kftp->GetTable(fKFTableName);
     assert(fKFTable);
 }

 // Always average over initial nucleons if the nuclear model is LFG
 bool average_over_nuc_mom ;
 GetParamDef( "IntegralAverageOverNucleonMomentum", average_over_nuc_mom, false ) ;
 fDoAvgOverNucleonMomentum = fLFG || average_over_nuc_mom ;

 fEnergyCutOff = 0.;

 if(fDoAvgOverNucleonMomentum) {
   // Get averaging cutoff energy
     GetParamDef( "IntegralNuclearInfluenceCutoffEnergy", fEnergyCutOff, 2.0 ) ;
 }
```

$GENIE/src/Physics/QuasiElastic/XSection/NievesQELCCPXSec.cxx

PauliBlocker

- Computes (LFG) or looks up the Fermi momentum.
- If the recoil momentum is less than the Fermi momentum, throw the event away and re-generate a new QE event (with, probably the same neutrino energy).

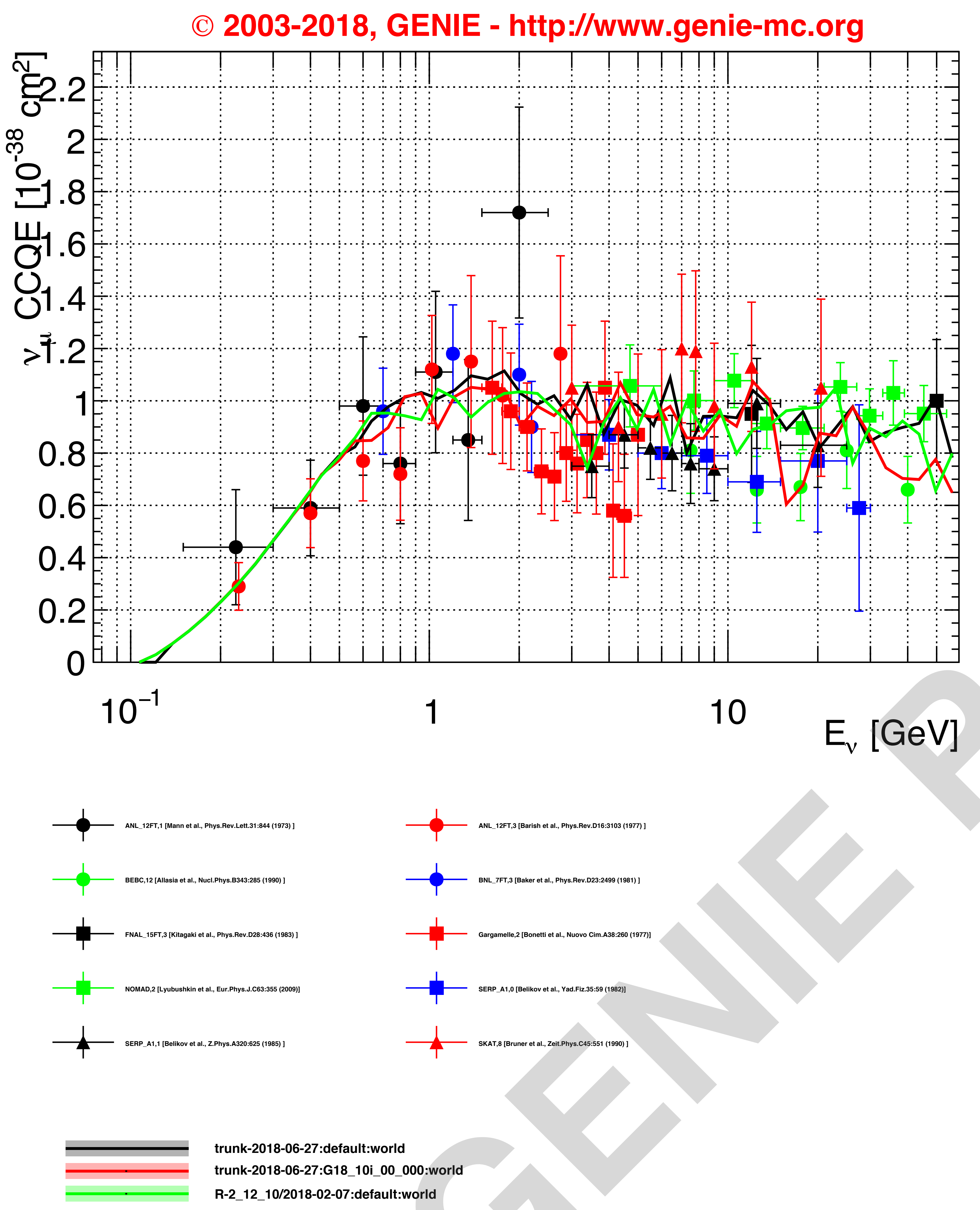
```
// get the Fermi momentum
double kf;
if(fLFG){
  int nucleon_pdgc = hit->Pdg();
  assert(pdg::IsProton(nucleon_pdgc) || pdg::IsNeutron(nucleon_pdgc));
  Target* tgt = interaction->InitStatePtr()->TgtPtr();
  int A = tgt->A();
  bool is_p = pdg::IsProton(nucleon_pdgc);
  double numNuc = (is_p) ? (double)tgt->Z():(double)tgt->N();
  double radius = hit->X4()->Vect().Mag();
  double hbarc = kLightSpeed*kPlankConstant/units::fermi;
  kf= TMath::Power(3*kPi2*numNuc*
           genie::utils::nuclear::Density(radius,A),1.0/3.0) *hbarc;
}else{
  kf = fKFTable->FindClosestKF(tgt_pdgc, nuc_pdgc);
}
LOG("PauliBlock", pINFO) << "KF = " << kf;
```

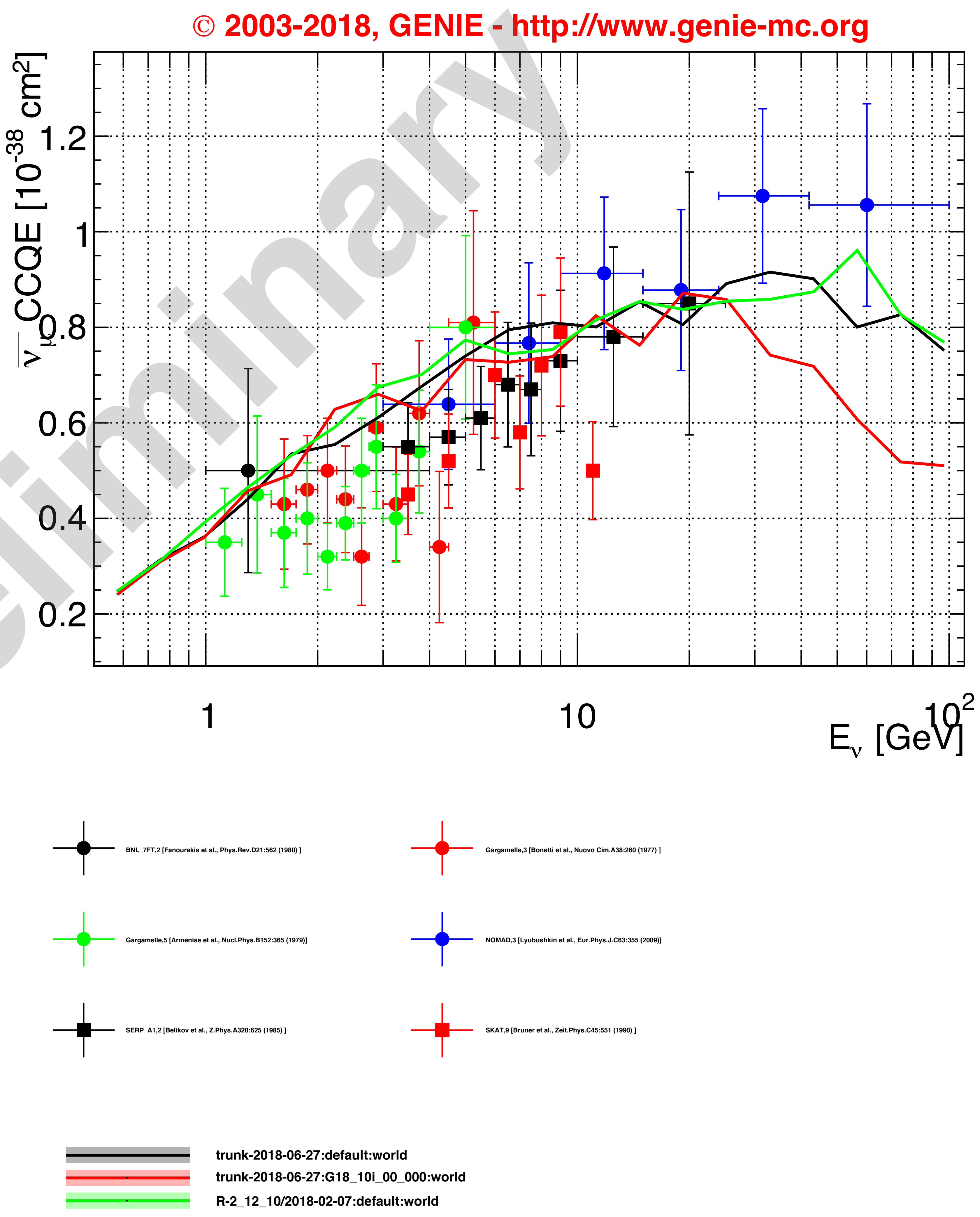
$GENIE/src/Physics/NuclearState/PauliBlocker.cxx

Validation plots

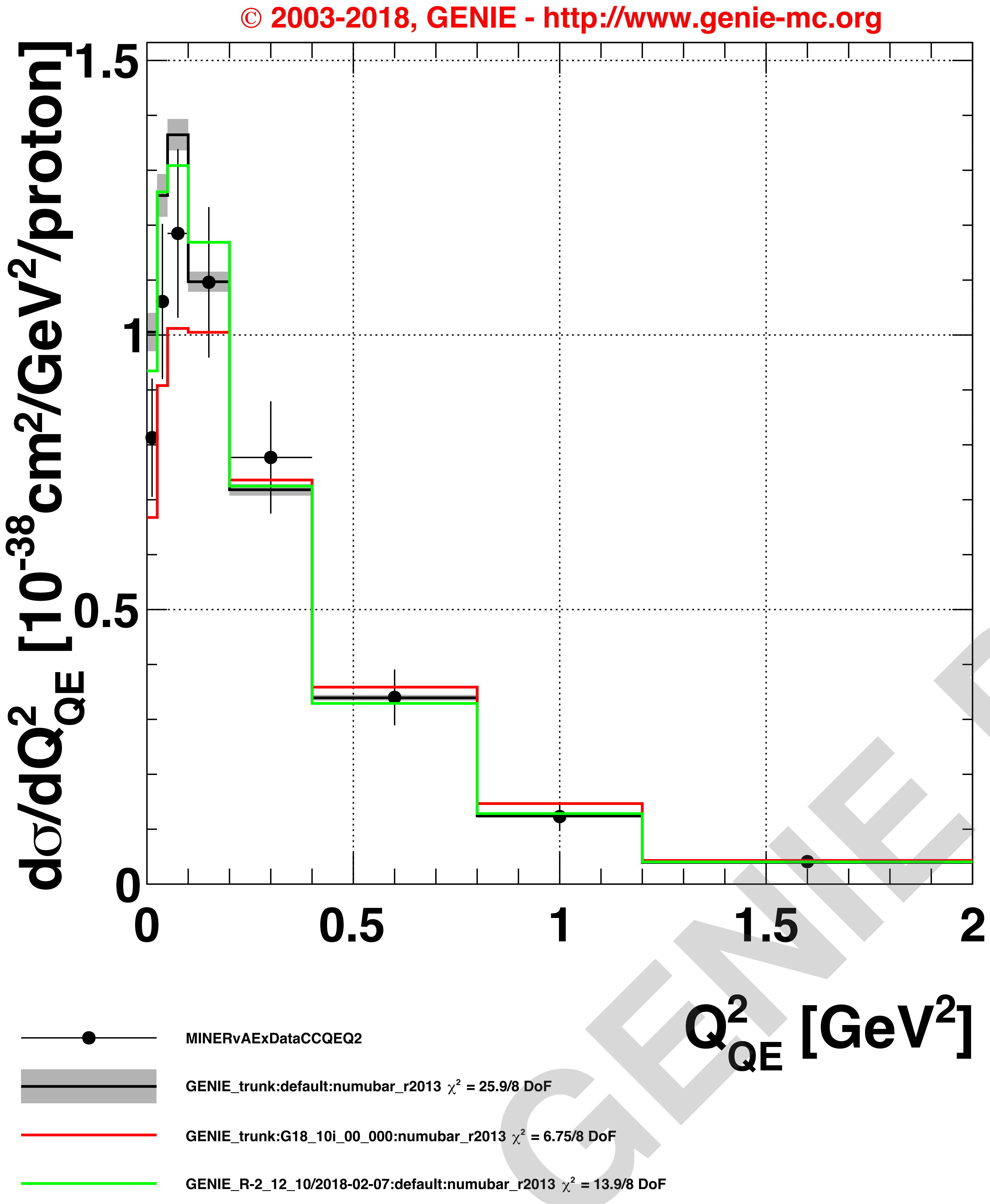
- Plots presented here based on an automated validation system that runs weekly.
- Event samples are low statistics (we run increased event counts for release validation, but try to be conservative about computing resources week-to-week).
- Additionally, we use a small number of knots in computing the total cross section "splines" to reduce the computational overhead, which makes results less reliable at higher energies (again we run more complete samples for data releases and for release validation).

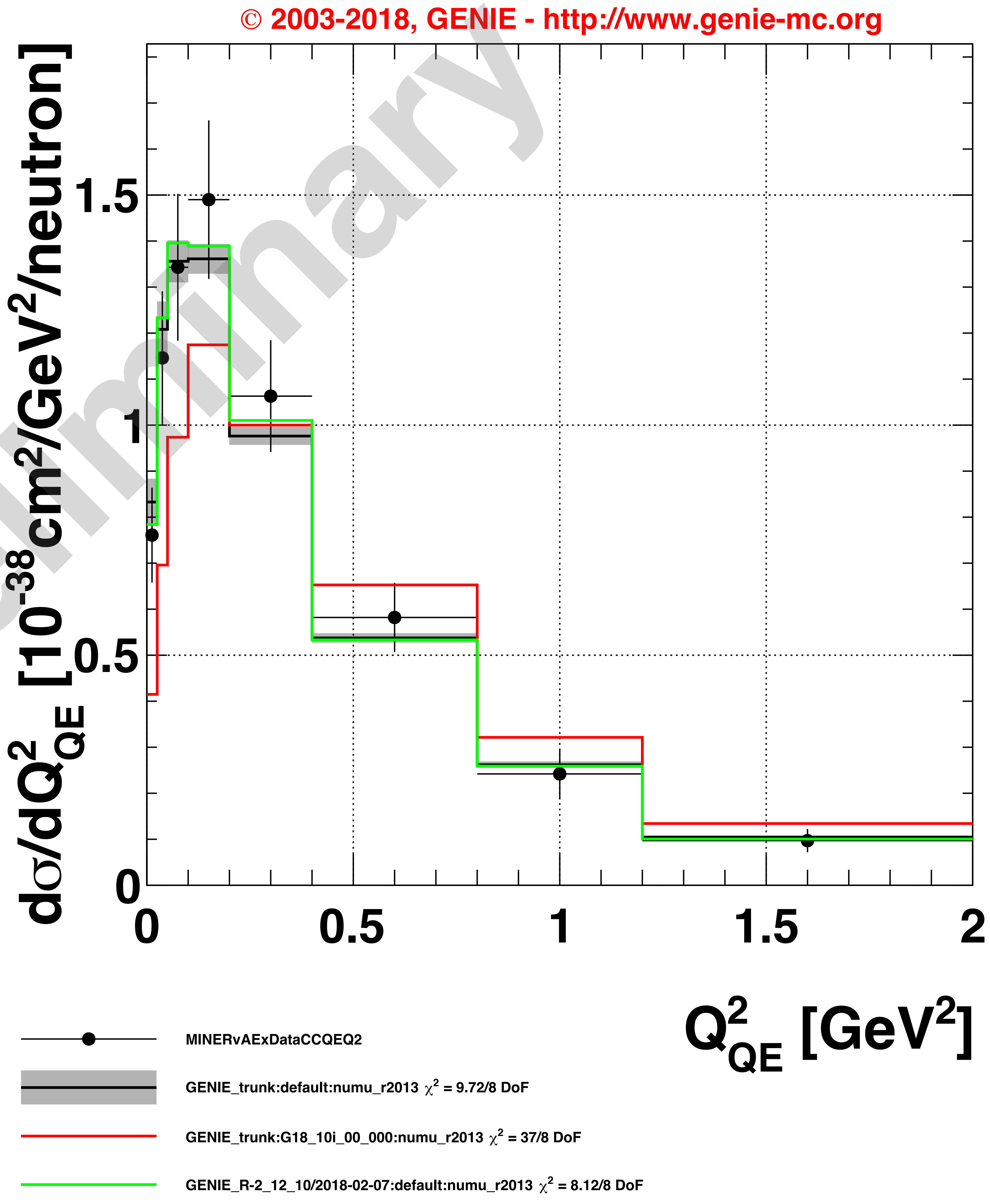
Recall, low MC statistics





MINERvA (2013) - MC == GENIE QE - likely old flux, validation still required

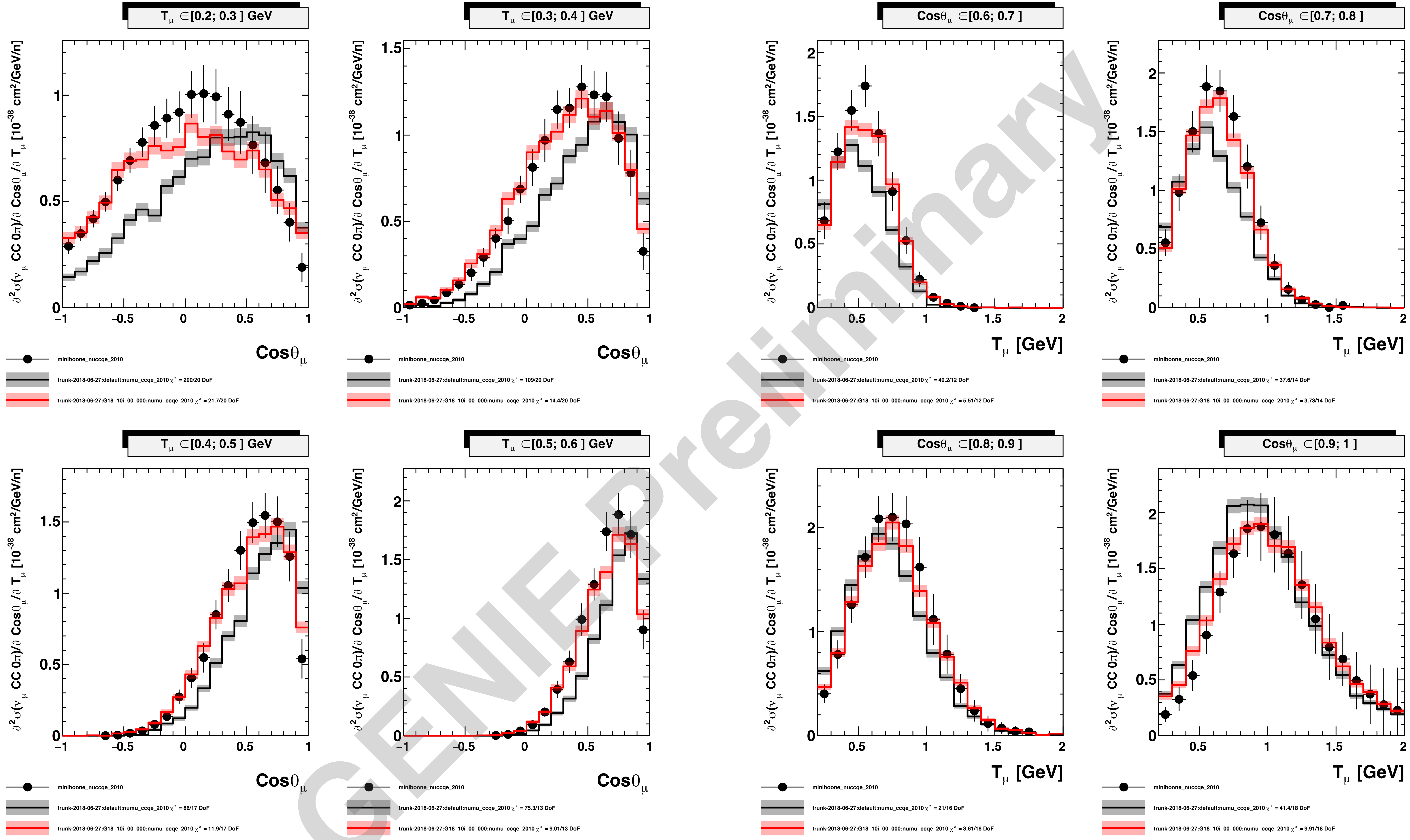




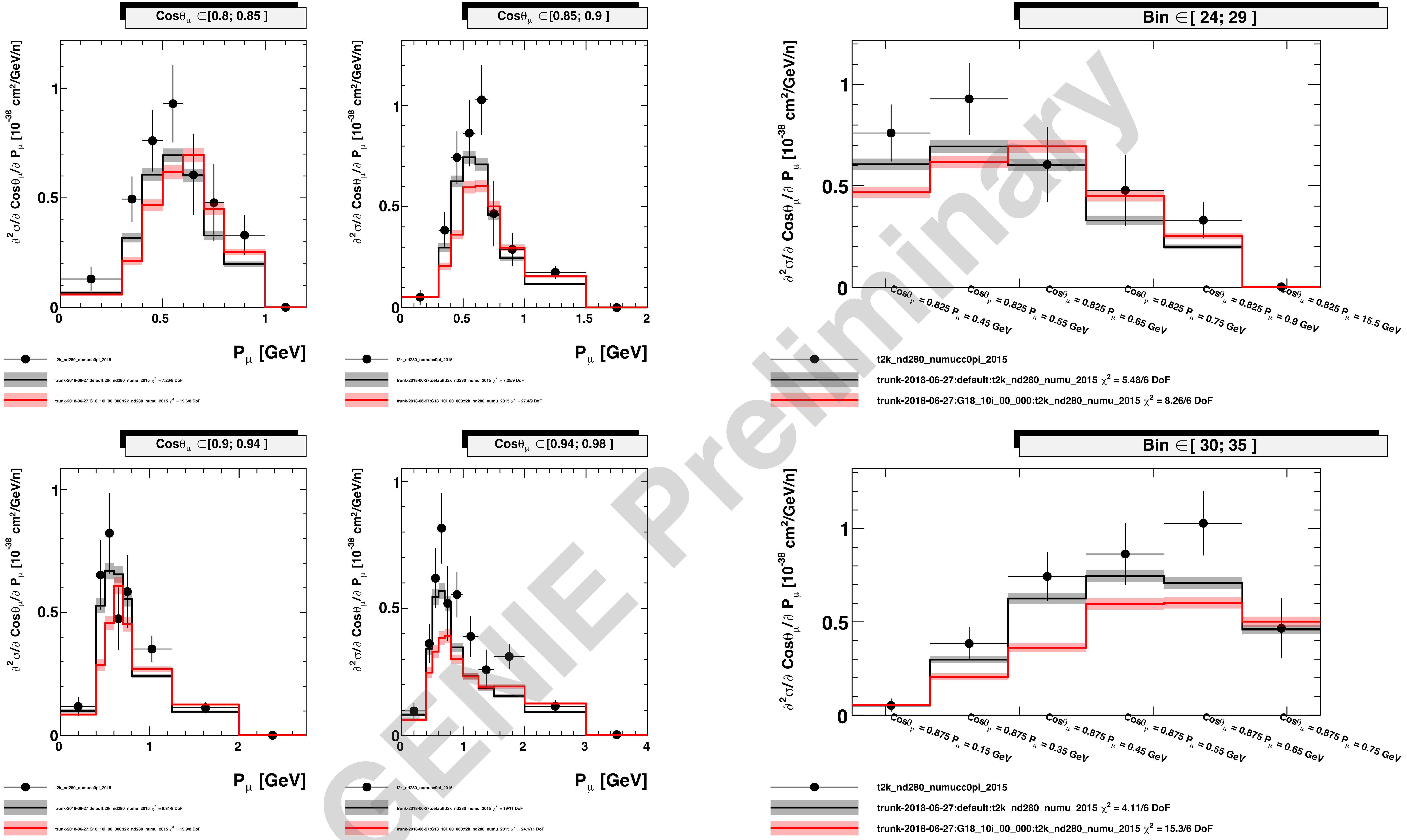
Genie
UNIVERSAL NEUTRINO GENERATOR & GLOBAL FIT

Fermilab

MiniBooNE (2010) - MC == *QE-like* - validation still required

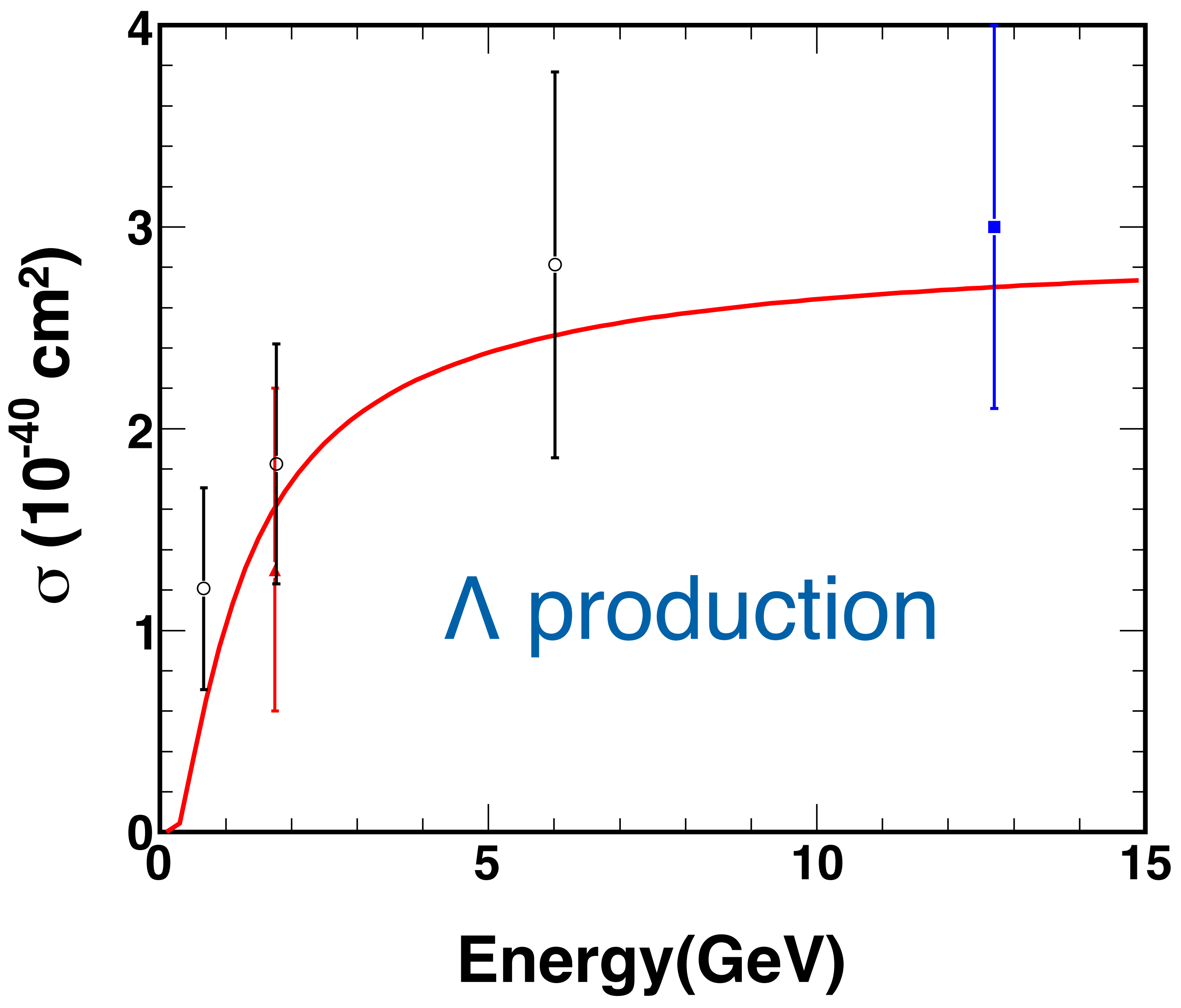


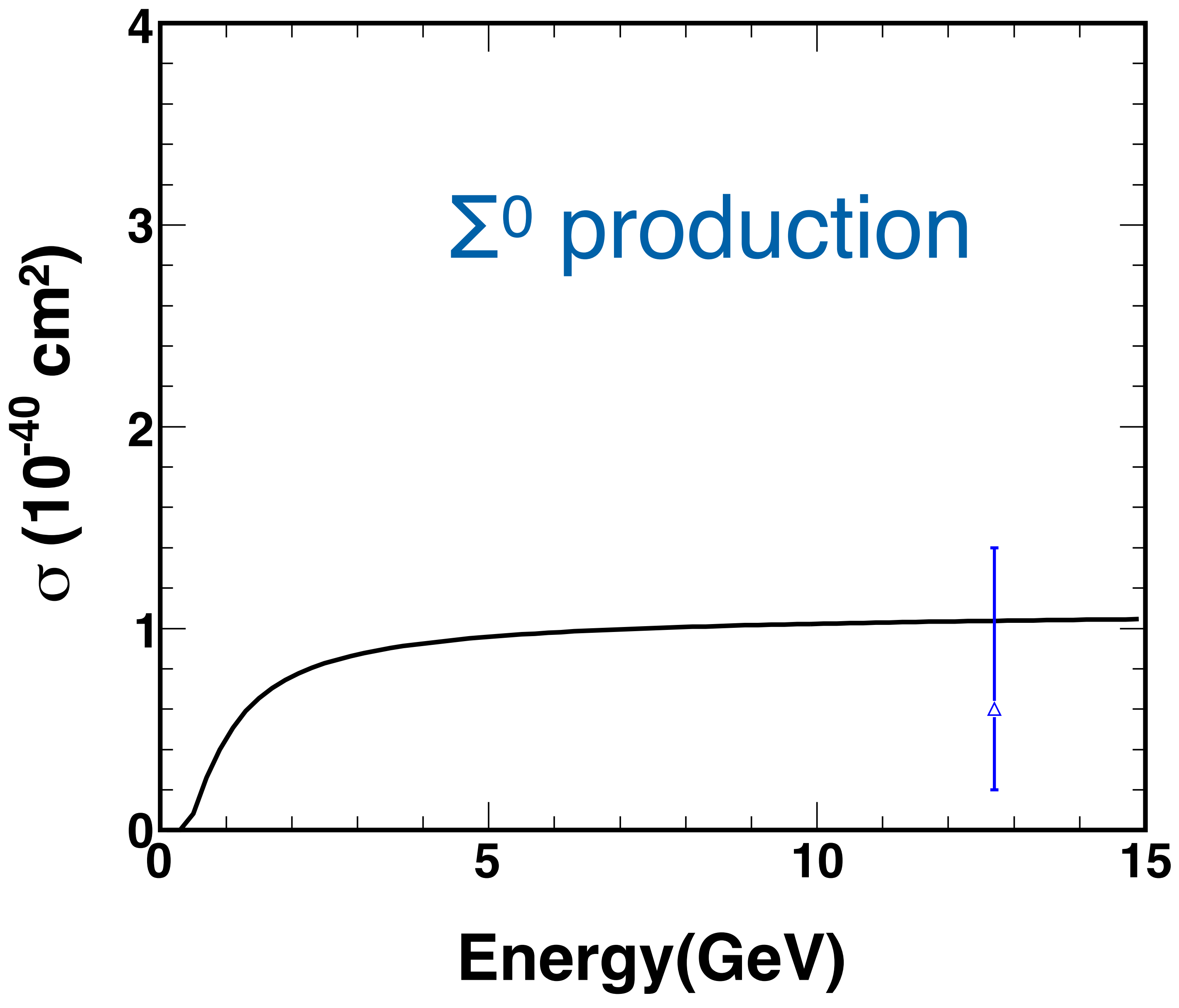
T2K (2015) - MC == *QE-like* - validation still required



QEL hyperon production

- Original calculation in Weak Interactions at High Energies, A. Pais, Annals Phys. 63 (1971) 361
- Model processes ΔS = 1 events, produced by antineutrinos in three related channels (below).





$$\overline{\nu}_\mu + p \rightarrow \Lambda + \mu^+, \overline{\nu}_\mu + p \rightarrow \Sigma^0 + \mu^+, \overline{\nu}_\mu + n \rightarrow \Sigma^- + \mu^+$$

Outlook and conclusions

- Moving towards "nuclear first" models that contain, e.g., 1 body and 2 body currents along with their interference terms (e.g. Carlson and Pastore).
- This sort of model integrates over a more sophisticated ground state (fuses a point where the GENIE model is currently factorized).
- GENIE features a flexible, highly configurable suite of 1p1h QE models.
- The most commonly used models are all available (and even some less commonly used models).

Please enjoy responsibly...

Fermilab

Thanks!

Luis Alvarez-Ruso [8], Costas Andreopoulos (*) [2,5], Christopher Barry [2],
Francis Bench [2], Steve Dennis [2], Steve Dytman [3], Hugh Gallagher [7], Tomasz Golan [1,4],
Robert Hatcher [1], Rhiannon Jones [2], Libo Jiang [3], Anselmo Meregaglia [6],
Donna Naples [3], Gabriel Perdue [1], Marco Roda [2], Julia Tena Vidal [2],
Jeremy Wolcott [7], Julia Yarba [1]

(The GENIE Collaboration)

(1) Fermi National Accelerator Laboratory, USA
(2) University of Liverpool, UK
(3) University of Pittsburgh, USA
(4) University of Rochester, USA
(5) STFC Rutherford Appleton Laboratory UK
(6) Strasbourg IPHC, France
(7) Tufts University, USA
(8) University of Valencia, Spain

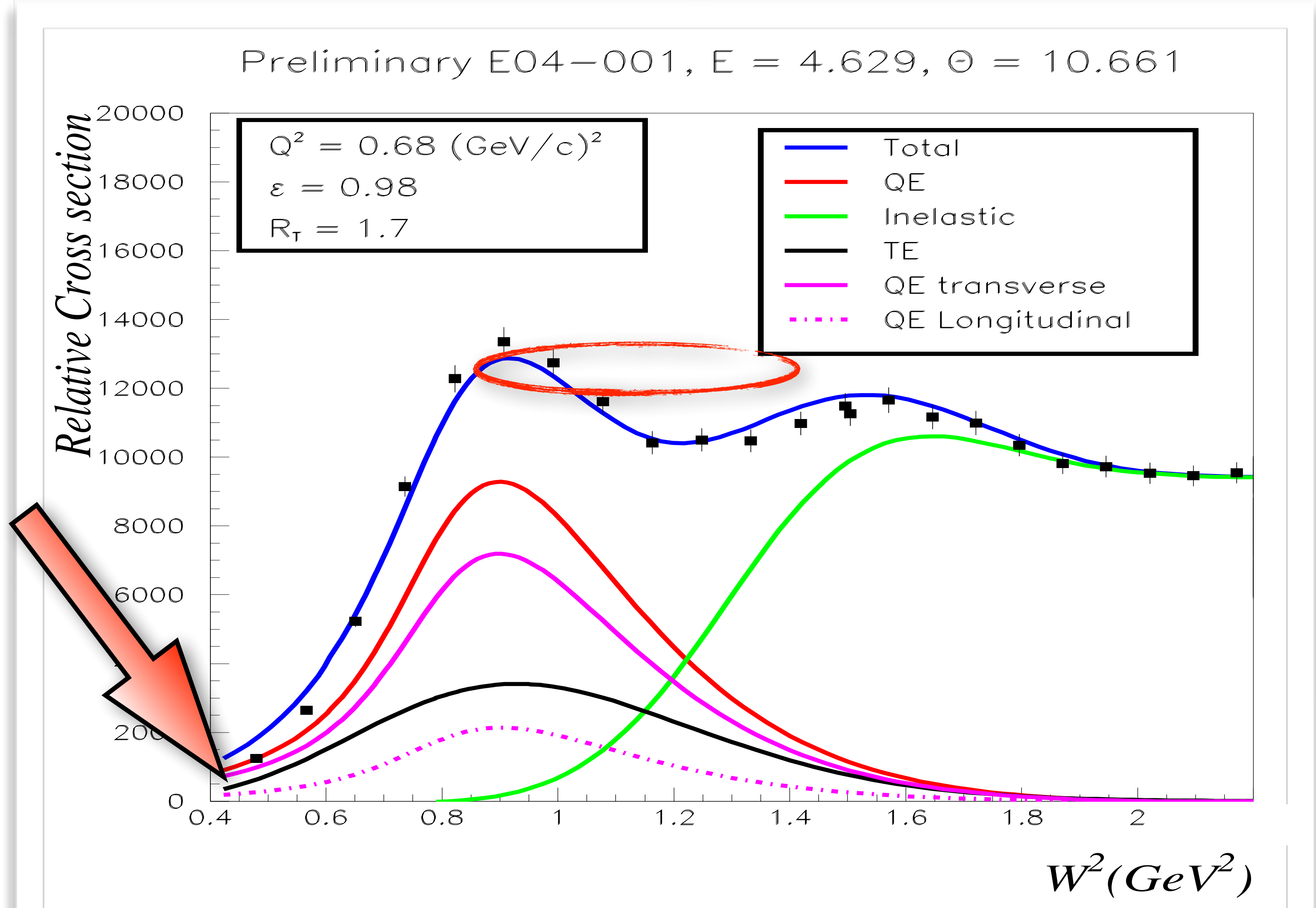
Thanks!

now...

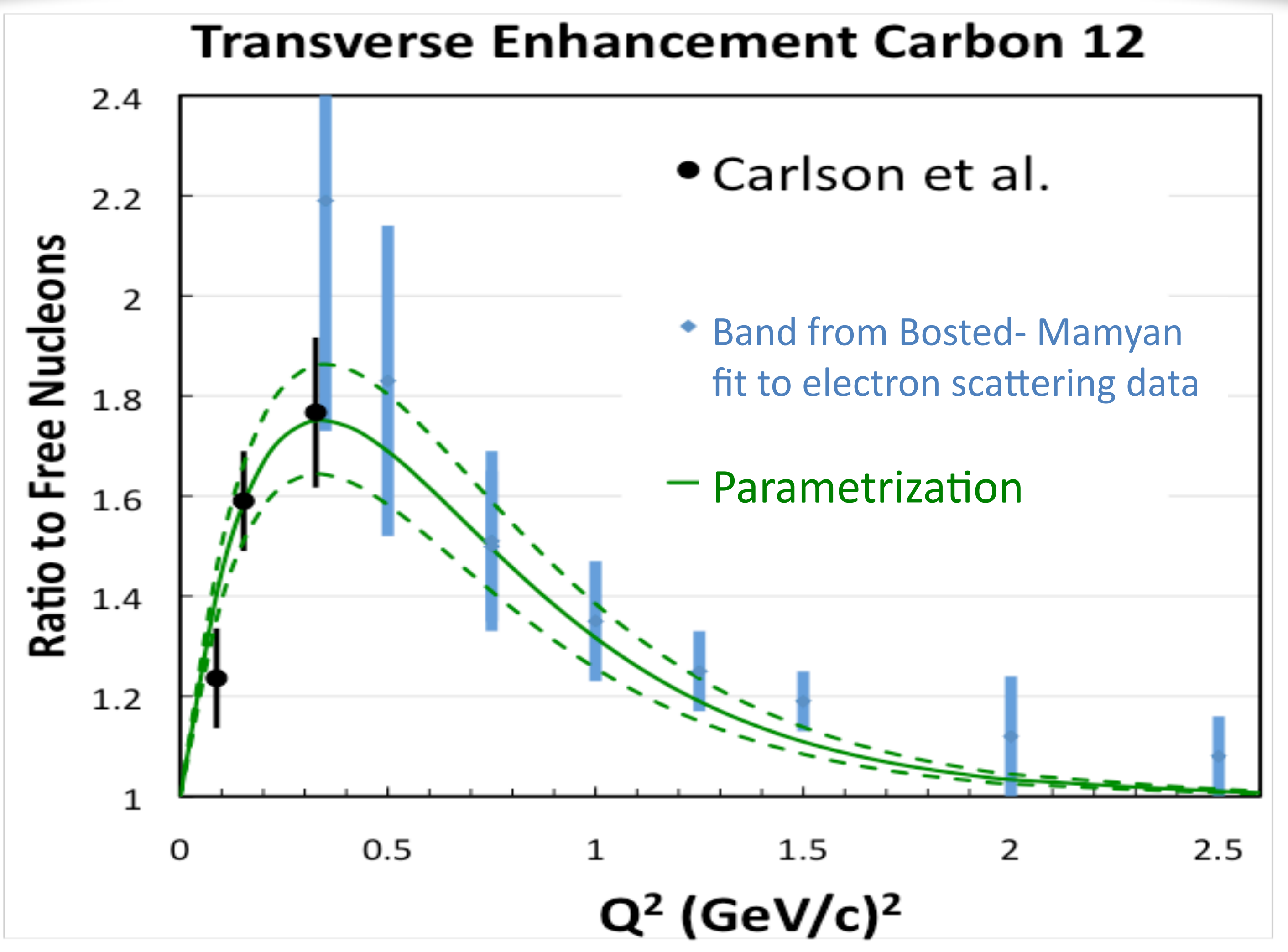
Back up!

Transverse Enhancement Model

- Separate the cross section into "longitudinal" and "transverse" components (polarization of the virtual photon) in electron scattering.
- Modify only vector magnetic form factors with e- scattering data - everything else is single free nucleon.
- e- scattering data suggests only the longitudinal portion of the QE x-section is ~universal free nucleon response function - the transverse component shows an enhancement relative to this approach.



Bodek, Budd, and Christy Eur.Phys.J. C71 (2011) 1726

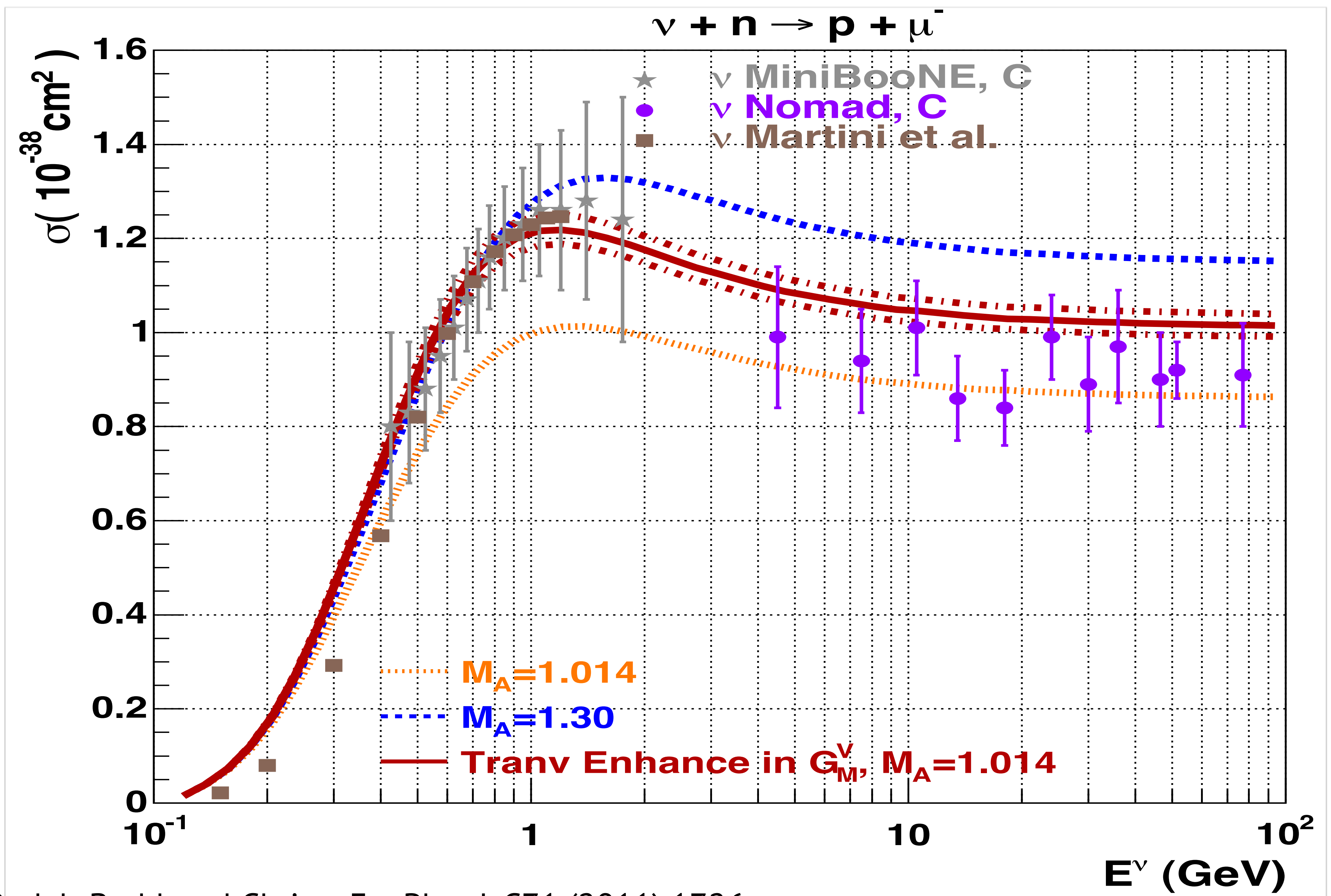


Fit to electron scattering data from JUPITER (JLab E04-001) to extract enhancement as a function of Q^2.

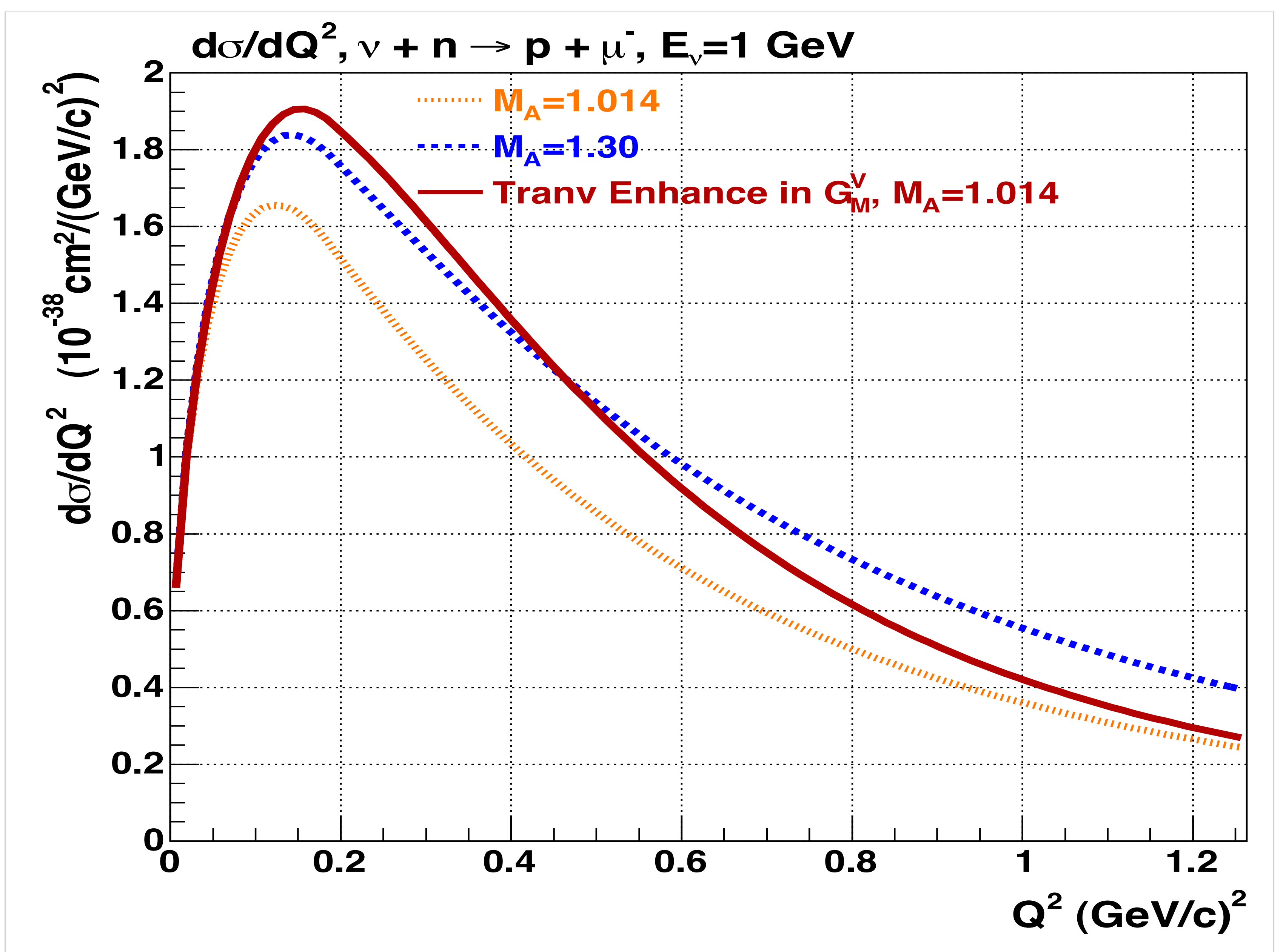
$$\frac{d^2\sigma}{d\Omega d\omega} = \Gamma\left[R_T(q,\omega) + \epsilon \cdot R_L(q,\omega)\right]$$

Transverse Enhancement

- $d\sigma/dQ^2$ w/ M_A = 1.014 GeV & TEM is very similar to the result for M_A = 1.3 GeV for $Q^2 < 0.6$ $(GeV/c)^2$.
- For high Q^2, the TEM contribution is small.
- Experiments at high energy often remove low Q^2 values from their M_A fits - predict an even lower M_A due to steep slope for $d\sigma/dQ^2$ at M_A = 1.014 GeV.



Bodek, Budd, and Christy Eur.Phys.J. C71 (2011) 1726



Effective Spectral Functions and Superscaling

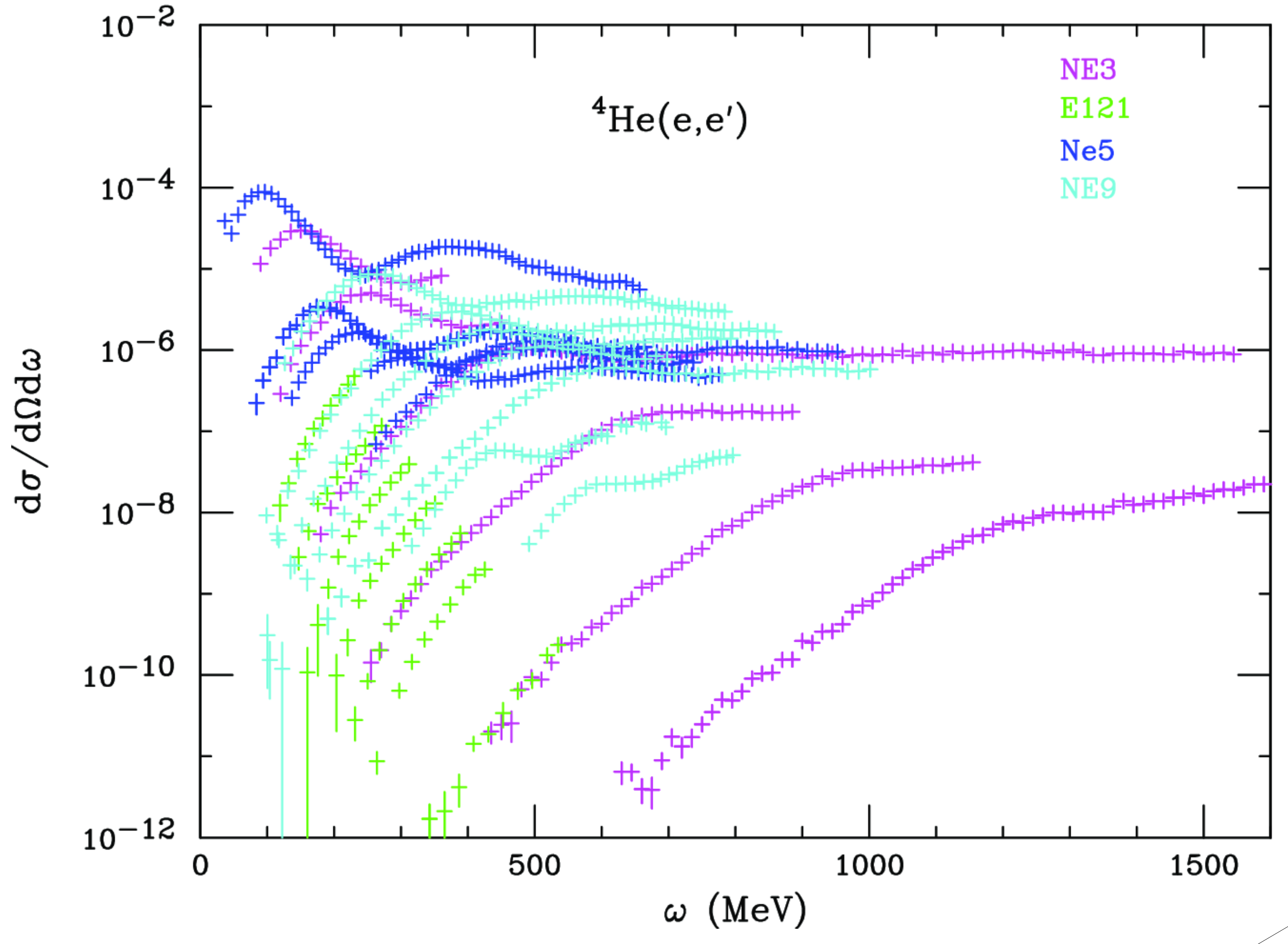
- The idea of superscaling* originated in attempts to explain inclusive electron scattering.
  - Compute a "reduced" single-nucleon cross section for a nuclear target with A nucleons, in the quasielastic region (assuming a "real" quasielastic cross section, so use single nucleon form factors and an appropriate Fermi motion model in the computation).
  - Plot against a selection of variables...
    - If the results don't depend on the variables and a universal behavior emerges, the results *scale*.
      - Scaling of the first kind: no dependence on momentum transfer, q.
      - Scaling of the second kind: no dependence on the momentum scale that characterizes specific nuclei (the Fermi momentum)
      - Superscaling: both kinds of scaling are present.

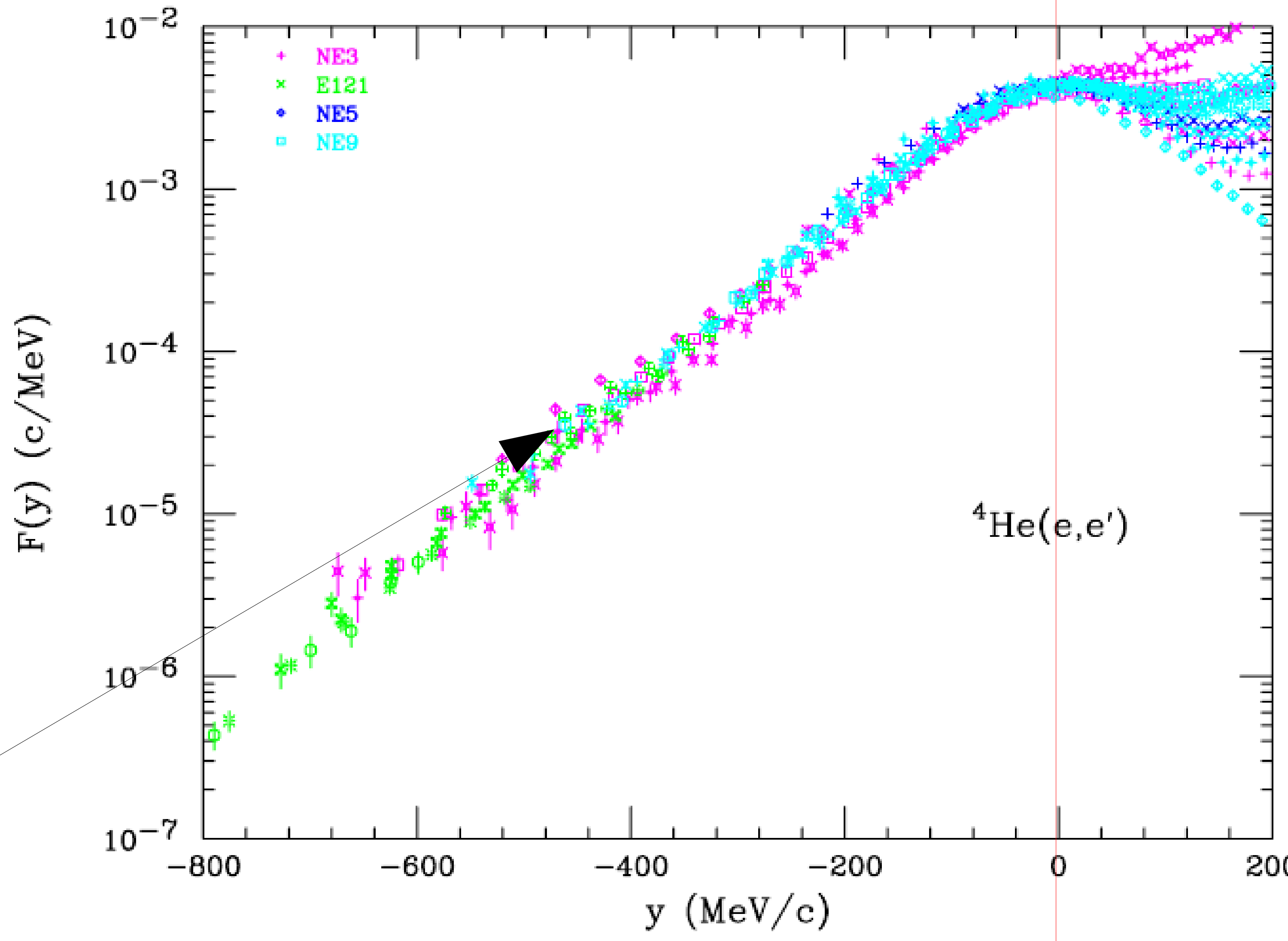
*See, e.g., Amaro, Barbaro, Caballero, Donnelly, Molinari, and Sick PRC 71, 015501 (2005)

Example: ^{4}He SLAC data

ω<ω$_{QEP}$ (x>1, y<0)

ω>ω$_{QEP}$ (x<1, y>0)





$$y \cong y_\infty = \sqrt{\tilde{\omega}(2m_N + \tilde{\omega})} - q$$

Independence of *q* of the scaling function *F(q,y)*:

$$F(y,q) \longrightarrow F(y) \equiv F(y,\infty) \quad \text{for} \quad q \to \infty$$

Scaling of the first kind (or y-scaling)

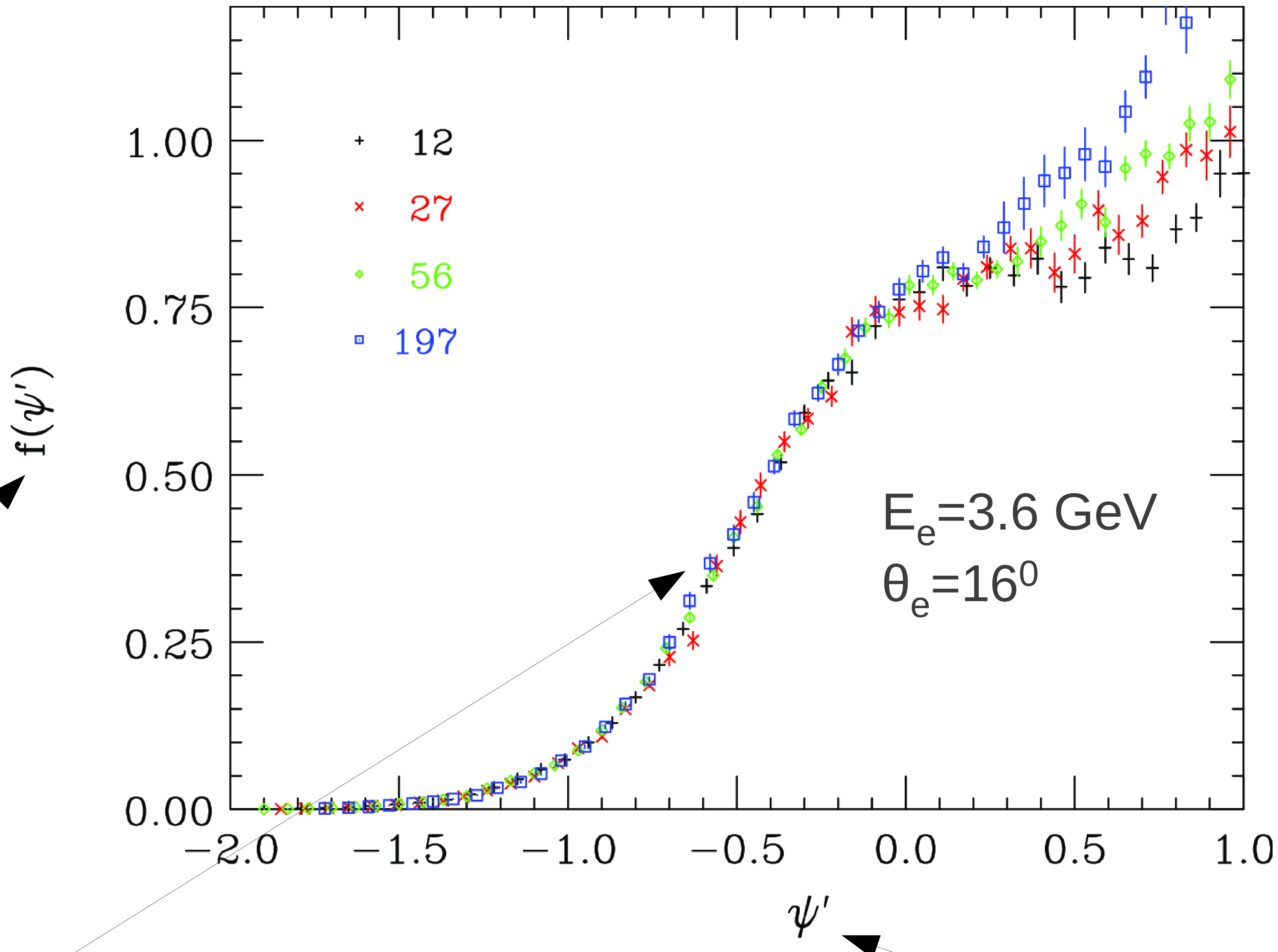
[Day,McCarthy,Donnelly,Sick,Ann.Rev.Nucl.Part.Sci.40(1990); Donnelly & Sick, PRC60(1999),PRL82(1999)]

Plotting f(q,ψ') at fixed kinematics (q) for different nuclei (A) one gets

M. Barbaro, INT'13

k_A*F

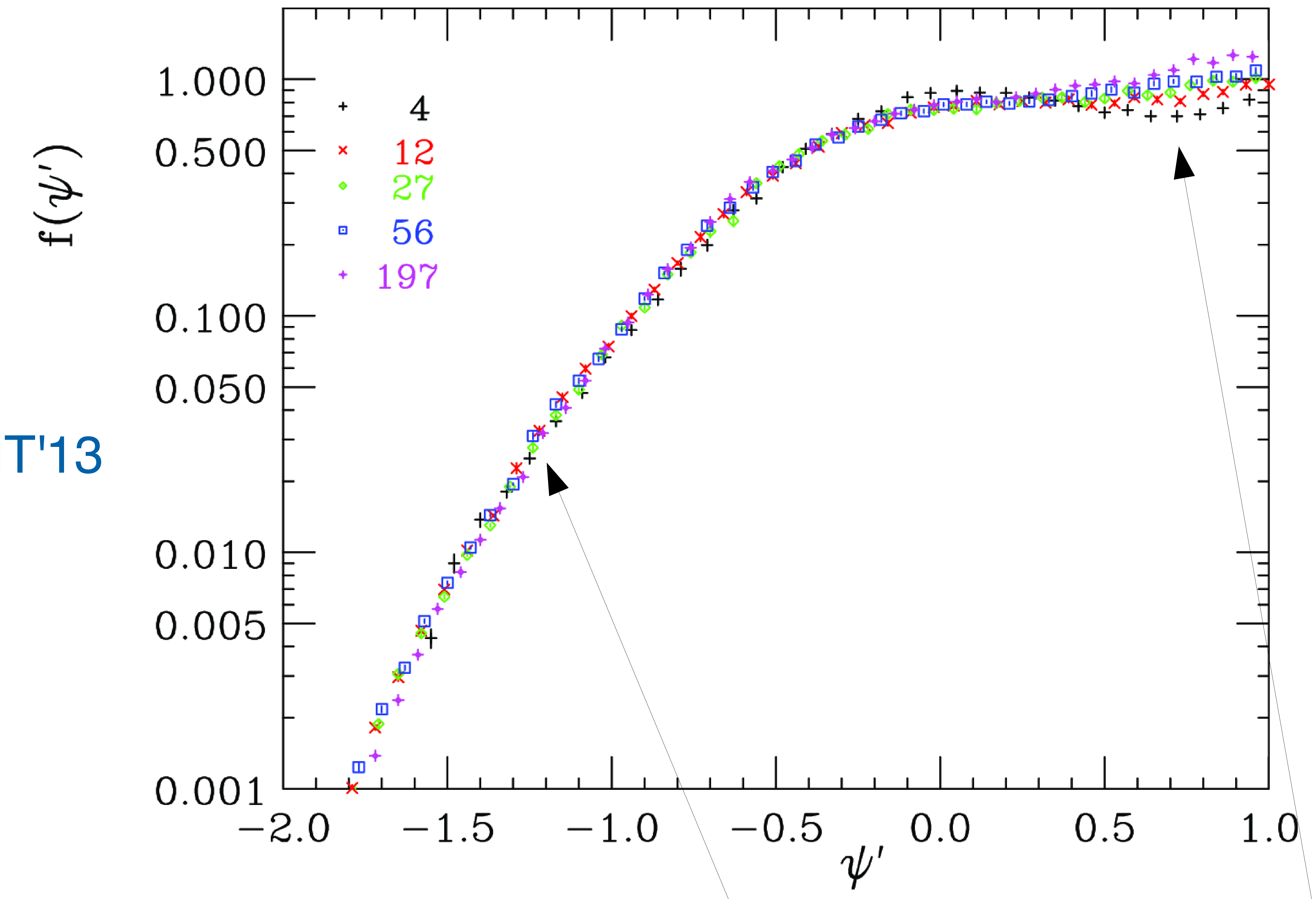
k_A = characteristic momentum scale for each nucleus



~y/k_F

Second kind scaling = A-independence for ψ' < 0

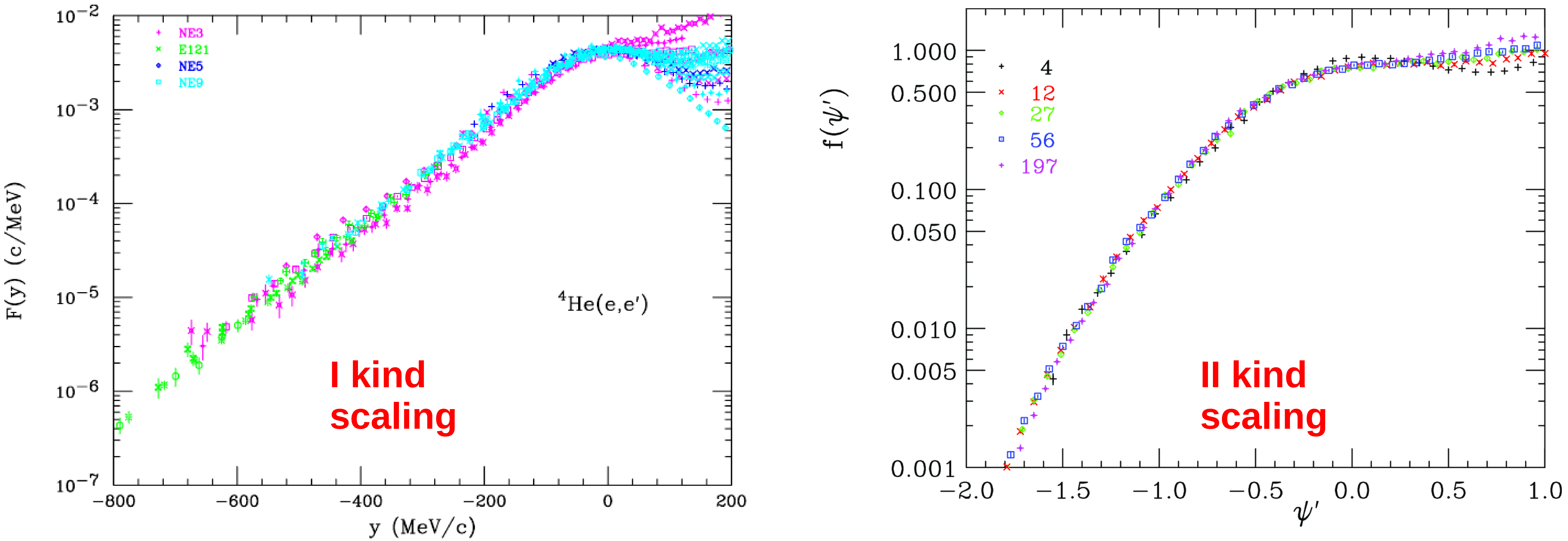
In semi-logarithmic scale:



M. Barbaro, INT'13

Violations for $\psi' > 0$ due to non QE contributions

Second kind scaling = A-independence for $\psi' < 0$



We define **"Super-Scaling"** the simultaneous occurrence of

I kind scaling (independence of q)

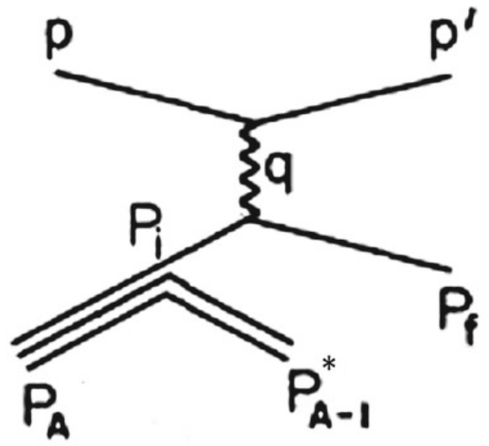
and

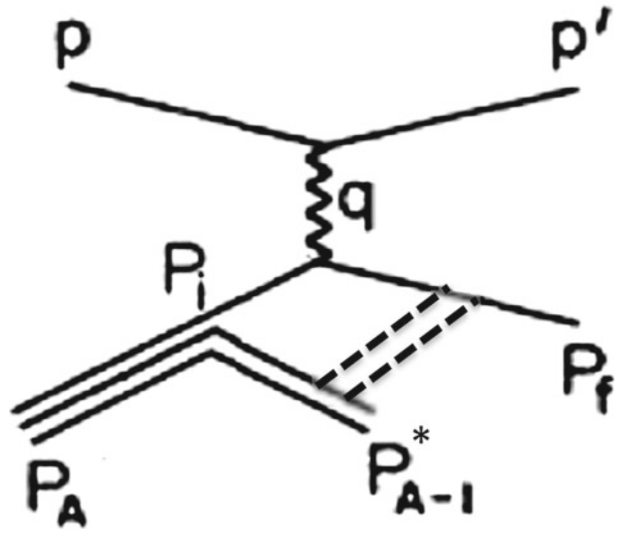
II kind scaling (independence of A)

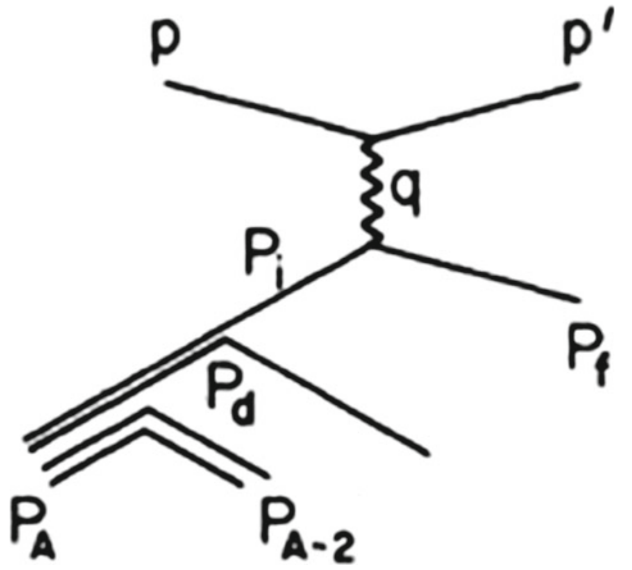
Effective Spectral Functions

- Superscaling formalism created to explain inclusive electron scattering.
  - The basic idea is to find a set of variables that allow you to compute a "reduced" (per nucleon) cross section that scales with A.
    - Very successful for electron scattering in the quasielastic region.
    - When separating longitudinal and transverse pieces of the cross section (polarization of intermediate photon), scaling is violated in the transverse piece - there are non-scaling contributions there from meson-exchange currents and other correlation effects.
- The formalism may be "inverted" to make neutrino cross section predictions.
  - The same scaling function is used for the transverse and longitudinal parts of the cross section (and for the vector and axial components).
  - In principle, this scaling function captures all relevant nuclear effects (initial state momentum, the removal energy distribution, two nucleon correlations, and final state interactions (as they impact the lepton)).
  - It makes no prescription for the final state hadronic system.

Bodek, Christy, Coopersmith EPJ C (2014) 74:3091







- 1p1h vs. 2p2h processes
- Final state interactions of the "first kind"
  - Distinguished from interactions of the second kind (which is the kind neutrino experimentalists usually mean when they say "FSI").

Bodek, Christy, Coopersmith EPJ C (2014) 74:3091