Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

# Implementing Spectral Function Model Into GENIE via Wrapper Infrastructure

Syrian Truong, University of Chicago
Supervisors: Dr. Minerba Betancourt and Dr. Steven Gardiner, Fermilab
Joint Meeting between Theorists and Experimentalists
12 August 2021

In partnership with:

GEM THE NATIONAL GEM CONSORTIUM | THE UNIVERSITY OF CHICAGO

# Introduction

- Fermilab is a leader in experimental neutrino research.
- This research would benefit from improved simulation models of neutrino interactions, for analysis of data.
- Improved understanding of neutrino properties though this, could lead to discovering fundamentally new physics.

**Figure 1: A few neutrino related experiments associated with Fermilab: MicroBooNE (left), DUNE (middle), MINERvA (right). (Credit: Fermilab)**

# Introduction (Continued)

- Fermilab utilizes **GENIE** for neutrino based simulations.
- GENIE is a large framework, using Monte Carlo event generators for neutrino interaction simulation & analysis.
- However, it would benefit from better neutrino scattering mechanism models, in particular for **quasielastic (QE)**.

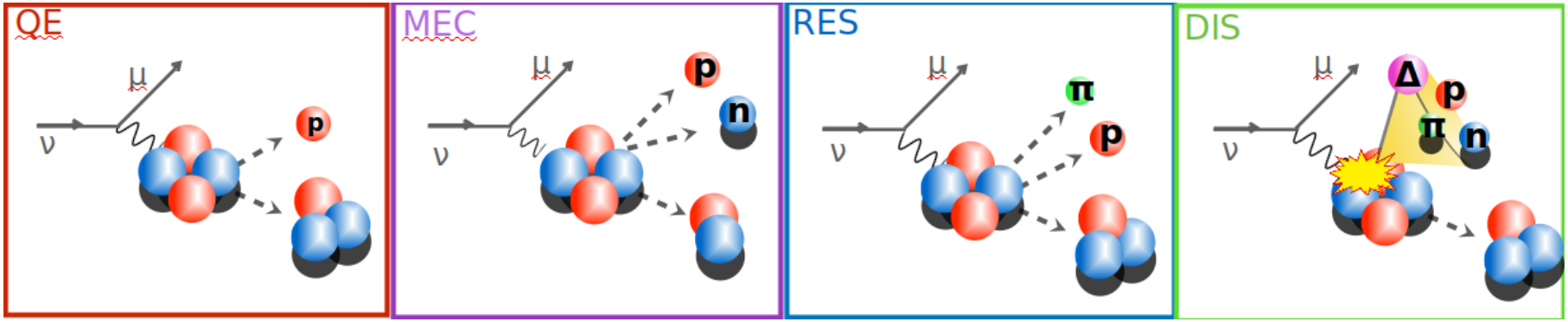



**Figure 2: Different neutrino scattering reaction mechanisms:**
**QE = Quasielastic, MEC = Meson Exchange Current, RES = Resonance, DIS = Deep Inelastic.**
**(Credit: Dr. Noemi Rocco)**

# Introduction (Continued)

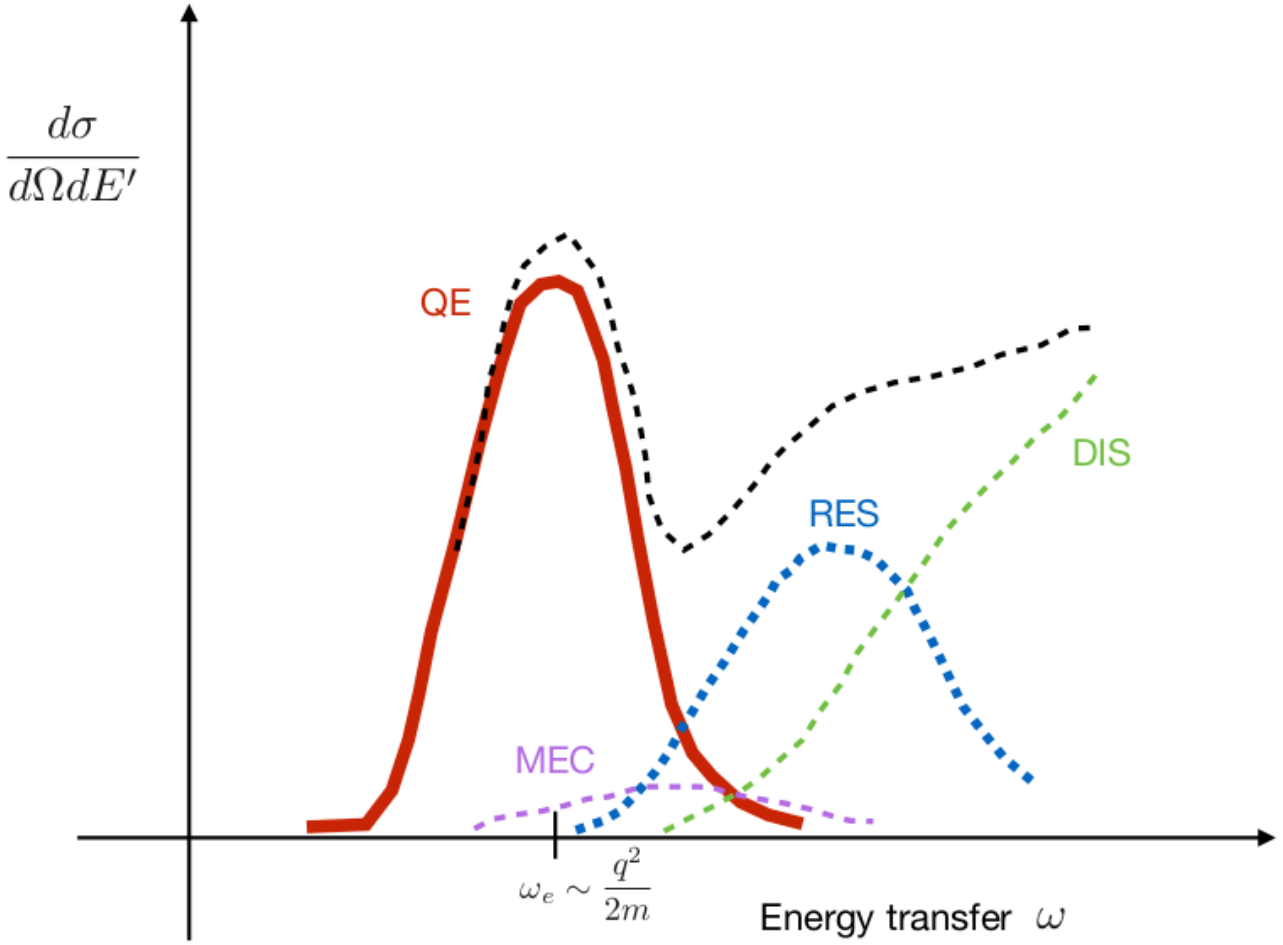



**Figure 3: Example of scattering mechanism models' contributions (colored lines) to final observed experimental data result (black dotted line) for these quantities, which are associated with neutrino interaction events. (Credit: Dr. Noemi Rocco)**

# Improvements to the Model

- Improvements to the quasielastic scattering models exist via Dr. Noemi Rocco's **Spectral Function (SF)** model.

- Electron and neutrinos interact similarly, many identical nuclear effects. This would allow us to constrain GENIE.

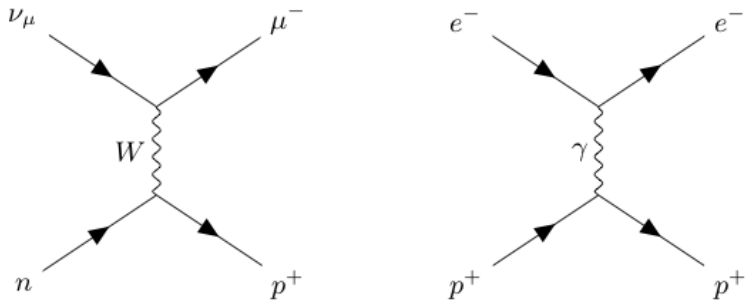


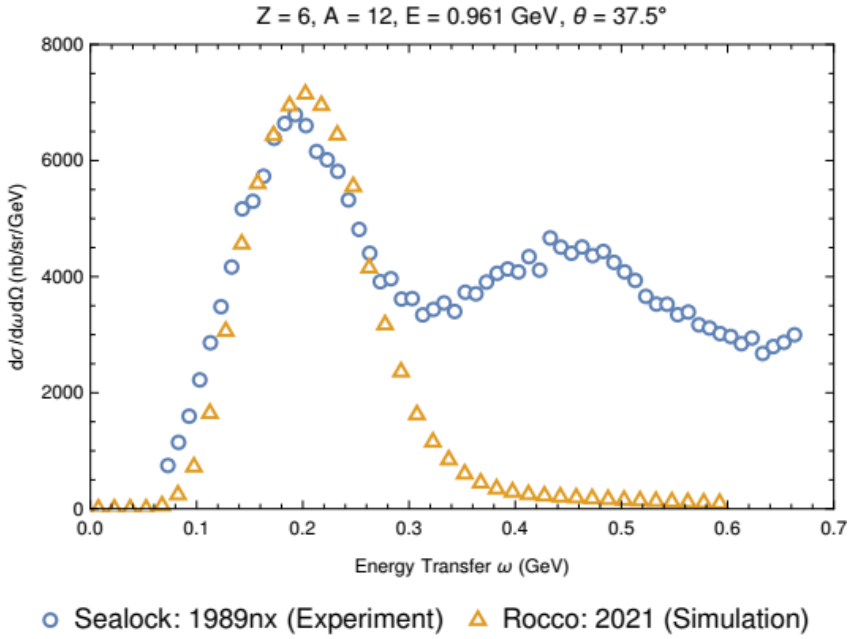

**Figure 4: Feynman diagrams show similarity for the charged current quasielastic neutrino (left) and quasielastic electron (right) scattering processes. (Credit: K. Ewart, J. Ellis, M. Allen)**

- **These calculations are available for neutrino and electron scattering in Fortran 90, but GENIE is written in C++.**

# Improvements to the Model (Continued)

- Below: differential cross sections vs. electron energy loss, for experimental/simulation electron scattering data.

**Figure 5: Experimental (blue) and simulation (orange) data of electron scattering off of Carbon-12 nucleus with 0.961 GeV beam energy and 37.5 degree angle. (Credit: Sealock 1989nx, Rocco 2021)**




- Good agreement within the quasielastic dominated region.
- These results can be extrapolated via electron-neutrino relations into better agreement for neutrino scattering.

## Wrapper Implementation

- We utilize the strengths of both GENIE and the Spectral Function model, via wrapping the SF model into GENIE.
- Removing GENIE's current quasielastic model related C++ code lines with C++ code that "calls" Fortran 90 SF code.
- This is our wrapper, which executes only part of Dr. Rocco's (slightly edited) SF code's improved calculations.
- Thus GENIE now generates inputs for part of the SF code, which then returns cross section data for final GENIE/C++ calculations.

# Wrapper Implementation (Continued)

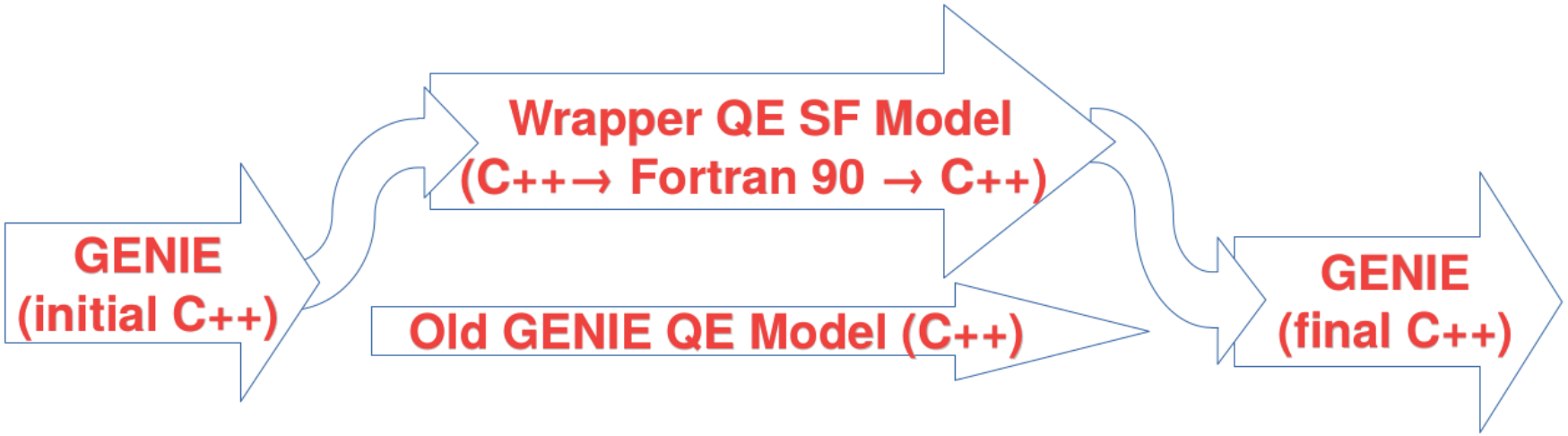



Figure 6: Flowchart depicting new wrapper infrastructure.

## Wrapper Contribution Specifics - Overview

- I edited/added part of four main files:
  - 1. Makefile
    (added Fortran 90 related compile lines)
  - 2. FortranWrapperQELPXSec.cxx
    (added wrapper, inputs, and subroutine calls)
  - 3. diracmatrices.f90
    (created this file to call currents_opt_v1.f90 dirac_matrices_in() subroutine)
  - 4. xsec_fact_new.f90
    (removed loop to just focus on one nphi value per cc1 subroutine call)
- Note: Aside from this, there was additional “GENIE related plumbing” along with making sure the Pauli blocking was happening (as this was done in the main Fortran 90 file, rather than in the subroutines), which Steven will likely explain.

# Wrapper Contribution Specifics - Makefile

- 1. Makefile
  (added Fortran 90 related compile lines)

```
12  + test_xsec: test_xsec.o FortranWrapperQELPXSec.o root_dictionary.o
    currents_opt_v1.o xsec_fact_new.o diracmatrices.o \
13  +   nform.o
14  +       $(CXX) $(ROOT_LIBS) $(GENIE_LIBS) -o $@ $^ -lgfortran
19  + %.o: %.f90
20  +       gfortran -o $@ -c $^
21  +
```

- Just making sure that the Fortran 90 files compile correctly with the C++ files.

# Wrapper Contribution Specifics - FortranWrapperQELPXSec.cxx

- 2. FortranWrapperQELPXSec.cxx
  (added wrapper, inputs, and subroutine calls)

```
44  + extern"C"
45  + {
46  + void diracmatrices_(double *xmn_in);
47  + }
48  +
```

```
50    extern"C"
51    {
52  + void cc1_(double *xq, double *w, double *wt, double *xk, double *xp,
    double *ee0, double *theta, int *ig, double *xsec, double *nuphi);
53    }
54
```

- These extern lines are needed to partly enable the ability to call the specific Fortran 90 subroutines from this C++ class:

```
73  double FortranWrapperQELPXSec::XSec(const Interaction* interaction,
74    KinePhaseSpace_t kps) const
75  {
76    // This is the main "missing piece" for interfacing with Noemi's code.
77    // Given an Interaction object which will be pre-loaded with the correct
78    // variables, we need to extract the inputs to Noemi's function, call that
79    // function, and then return a differential cross section in natural units.
```

# Wrapper Contribution Specifics – FortranWrapperQELPXSec.cxx (Cont.)

- 2. FortranWrapperQELPXSec.cxx (added wrapper, inputs, and subroutine calls)
  - Skipping over the numerous lines utilized to setup the specific variable inputs from GENIE to input into the subroutine calls, we note that there needs to be some unit conversions between GENIE and Dr. Rocco's code (i.e. GeV to 1/fm, GeV to MeV, and etc. where appropriate), along with some additional GENIE plumbing for accessing the final particle information (as variables that previously held such information get wiped by this time), and Pauli blocking / phase space considerations. Subroutine calls below:

```
220     double xsec;
217 +   // Set up dirac matrices via calling diracmatrices Fortran90
    subroutine
218 +
219 +    diracmatrices_(&xmn_in);
242     // Calculate sigma via calling cc1 Fortran90 subroutine
243
244 +    cc1_(&xq, &w, &wt, &xk, &xp, &ee0, &theta, &ig, &xsec, &nuphi);
245
246     return xsec;
247   }
```

# Wrapper Contribution Specifics - diracmatrices.f90

- 3. diracmatrices.f90
  (created this file to call dirac_matrices_in() subroutine in currents_opt_v1.f90)

```
+ subroutine diracmatrices(xmn_in)
+       use dirac_matrices
+       IMPLICIT NONE
+       real*8 :: xmn_in
+
+       call dirac_matrices_in(xmn_in)
+
+       return
+
+ end subroutine diracmatrices
```

- This file is needed because of a structure setup that needs to happen before the cc1 subroutine can be called, which handles the main part of the sigma or cross section calculation, otherwise it would result in incorrect/undefined values.

# Wrapper Contribution Specifics - xsec_fact_new.f90

- 4. xsec_fact_new.f90
  (removed loop to just focus on one nphi value per cc1 subroutine call)

```
10  -
11  -    subroutine cc1(xq,w,wt,xk,xp,ee0,theta,ig,sigma)
```

```
10  + ! GENIE edit below, added in nucleon phi (nuphi) variable
11  +    subroutine cc1(xq,w,wt,xk,xp,ee0,theta,ig,sigma,nuphi)
```

```
do i=1,nphi
   phi=(dble(i)-0.5d0)*hphi
   xk_x=xk*sqrt(sina2)*cos(phi)
   xk_y=xk*sqrt(sina2)*sin(phi)
   xk_z=xk*cosa
   !...define the initial and final nucleon 4-momentum
   p_4(1)=ek
   p_4(2)=xk_x
   p_4(3)=xk_y
   p_4(4)=xk_z
   pp_4(1)=epf
   pp_4(2)=xk_x
   pp_4(3)=xk_y
   pp_4(4)=xk_z+xq
   call current_init(p_4,pp_4,q_4)
   call define_spinors()
   call sigccc(sig,ig)
   sigma=sigma+sig*hphi
enddo
```

```
!       GENIE edit below, commented out loop start
!       do i=1,nphi
!       GENIE edit below, changed i to nuphi
          phi=(dble(nuphi)-0.5d0)*hphi
          xk_x=xk*sqrt(sina2)*cos(phi)
          xk_y=xk*sqrt(sina2)*sin(phi)
          xk_z=xk*cosa
          !...define the initial and final nucleon 4-momentum
          p_4(1)=ek
          p_4(2)=xk_x
          p_4(3)=xk_y
          p_4(4)=xk_z
          pp_4(1)=epf
          pp_4(2)=xk_x
          pp_4(3)=xk_y
          pp_4(4)=xk_z+xq
          call current_init(p_4,pp_4,q_4)
          call define_spinors()
          call sigccc(sig,ig)
          sigma=sigma+sig*hphi
!       GENIE edit below, commented out loop end
!       enddo
```

- Removing loop to calculate correctly with GENIE's Monte Carlo method

# Results/Conclusion

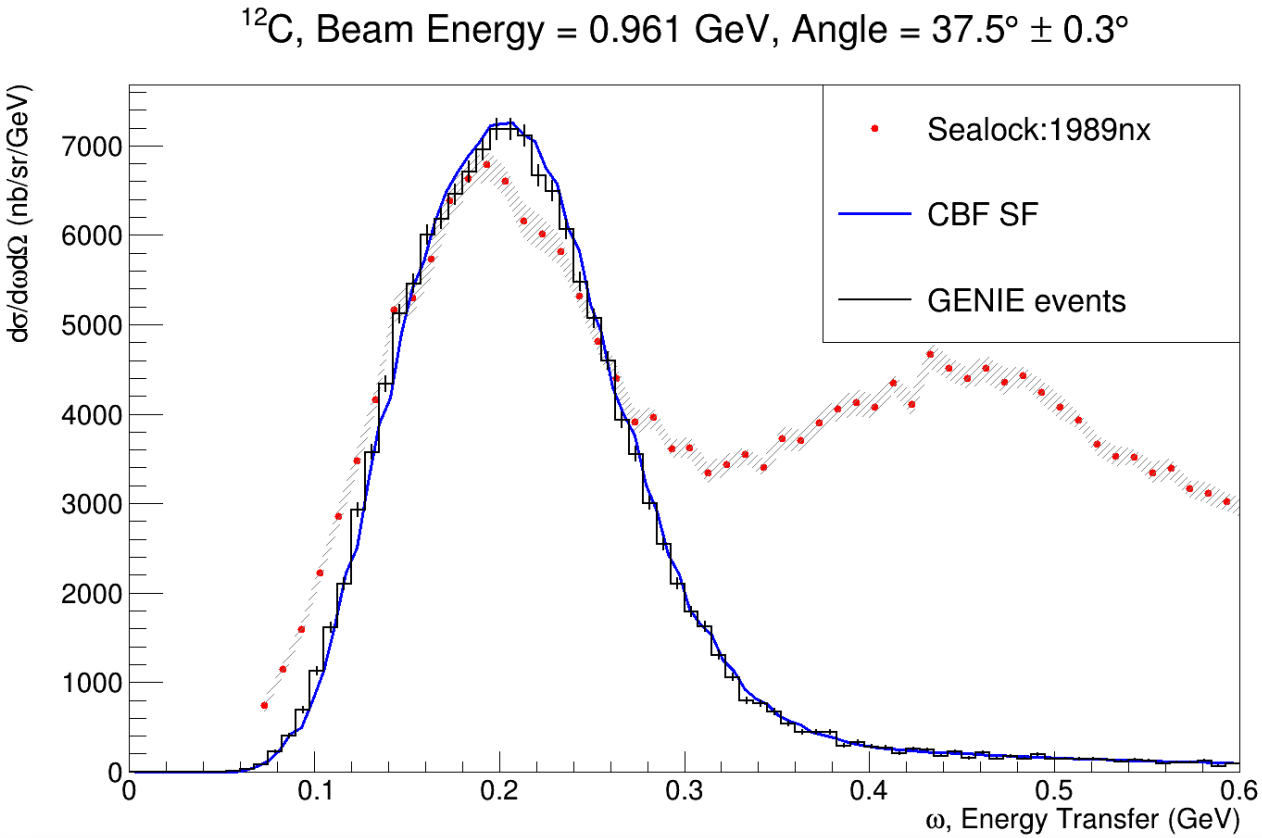



**Figure 7: Experimental (red), Fortran 90 only (blue), GENIE + wrapper (black) data of electron scattering off of Carbon-12 nucleus with 0.961 GeV beam energy and 37.5 degree angle. (Credit: Gardiner and Truong 2021)**

- Wrapper-based infrastructure addition result success for GENIE motivates the possibility for further similar wrapper additions to improve GENIE's other models (such as MEC, RES, DIS), while still utilizing its overall framework.

# Event generation procedure for the wrapper approach (1)

- Define the incident electron at fixed momentum along +z
- Draw a nucleon at random from the spectral function
  - Momentum magnitude, removal energy chosen based on Noemi's table file
  - Initial direction is isotropic
- Outgoing 4-momenta (e-, nucleon) chosen using a TGenPhaseSpace object
  - Takes CM energy (off-shell initial nucleon) and final masses as input
  - Samples uniformly over the allowed phase space

$$dV_n = \delta^{(4)}\left(p_a + p_b - \sum_{f=1}^{n} p_f\right) \prod_{f=1}^{n} \delta(p_f^2 - m_f^2)\,\theta(p_f^0)\,d^4p_f$$

- Full set of initial/final 4-momenta passed to wrapper for cross-section calculation

## Event generation procedure for the wrapper approach (2)

- Apply needed factors to wrapper output for GENIE units, transformation to $V_n$ phase space, and Pauli blocking
- Two options for finalizing the event:
  - Accept/reject based on maximum differential cross section, accepted events have unit weight
  - Accept the event unconditionally, but weight by the differential cross section
- First option preferred, but naive approach to MC sampling/integration gave poor performance (related to divergence at low $Q^2$)
- Settled on second option as a stopgap solution. Better numerical techniques are known and could be implemented.
- Full event details are passed to downstream parts of GENIE simulation (for hadronic final-state interactions, etc.)

## Outlook

- Syrian's summer project provides a first real-world look at two "theory API" ideas
  - Interfacing directly with a theory code with minimal modifications
  - Using the universal $V_n$ phase space versus bespoke solutions based on a specific choice of working variables
- Performance issues related to the second item require further investigation
- This strategy has the potential for a lot of flexibility
  - Other standard processes
  - BSM (even for high-multiplicity final states)
- Experience from LHC-style generators will continue to be helpful
- Interface for passing model parameters, etc. back and forth remains to be explored

## Acknowledgements


- Huge thank you to my supervisors Minerba Betancourt and Steven Gardiner, my mentors and peers from group 3, Mike Geelhoed, Ahmed Syed, Linden Carmichael, Tonie Butler, Caitlynn Hanna, Matthew Hoppesch, Anne Murray, Divas Subedi, and Fermilab staff Judy Nunez, Arden Warner, Jimmy Mcleod, as well as UChicago, Fermilab, the GEM program, and those who listened to my talk!


- Questions? Enjoy a picture of my dog Mindy in the meantime.