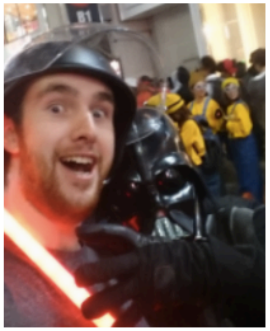
Imperial College London



The University Of Sheffield.

MICHIGAN STATE UNIVERSITY

UNIVERSITÄT BERN

Extracting Cross-sections

Clarence Wret, Patrick Stowell, Luke Pickering,
Callum Wilkinson

14/10/17

https://nuisance.hepforge.org

T2K

Lectures

- **Introduction to neutrino cross-section extraction**

Simple extraction, topologies, efficiency and background corrections, problems with model dependence, likelihood calculations, NUISANCE in a nutshell.

- **Towards Global Fits**

From bubble chamber to nuclear target data, MiniBooNE, MINERvA and T2K experiments, current and future experiments, problems observed in generator comparisons.

NUISANCE Tutorial

- Lots of talk during the school about how we need to start trying to benchmark neutrino generators against both electron and neutrino scattering data.

- NUISANCE is being specifically designed to try and address this issue.

- **Why do I need to learn it?**
  - Want to show how good your theory is compared to our awful generators? Get the generator predictions all at once!
  - Dominated by interactions you don't have a sideband for? Get constraints from external data!
  - About to release a cross-section measurement paper? Make lots of pretty plots!
  - Easily look at effects of free model parameters on kinematic distributions.
  - Add your fancy new data into our database so people can easily use it as soon as its released!

NUISANCE Tutorial

- Please try and download the NUISANCE code from our site.

  https://nuisance.hepforge.org

- Links to lots of possible ways to get our code can be found here:

  https://nuisance.hepforge.org/GettingTheCode.html

- Those working on a laptop the VirtualBox option is the recommended option. Those with access to Fermilab gpvm's can use the CVMFS option when logged into a gpvm machine.

- To do **all** examples you will also need the tutorial events

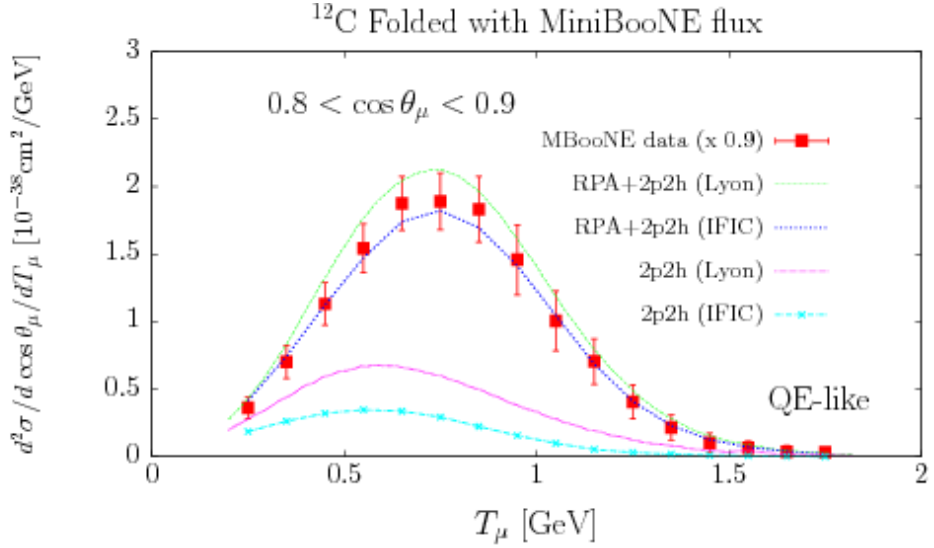
  https://nuisance.hepforge.org/TutorialEvents.html

- I want to expand the user base as much as possible so please speak to me in a break if you have issues building/running the code ☺

Introduction to neutrino cross-section extraction

Introduction

arXiv:1411.7821



- Theoretical model comparisons to publicly released cross-section data have spurred pushes to implement new models implemented into MC generators.
- Further data comparisons help us test generator implementations.
- **But how reliable is each set of published data?**
- Important to understand the experimental method used in each dataset. Subtle details can mean comparisons aren't as straightforward as you may think.

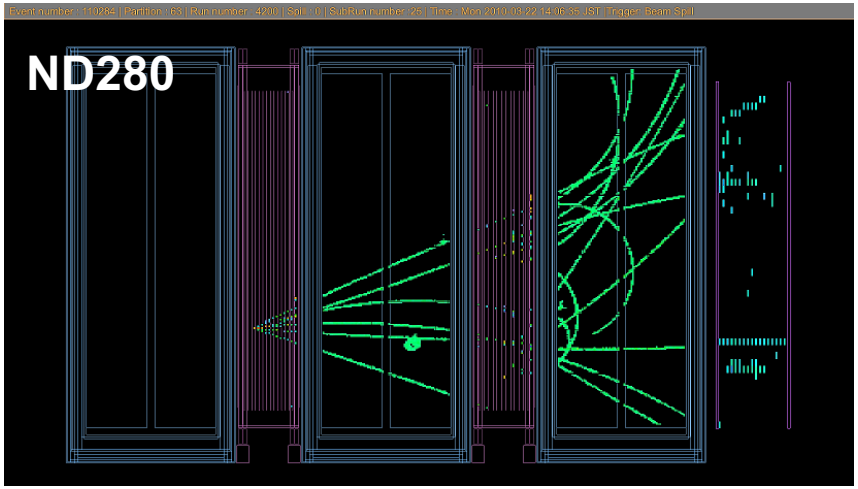
Measuring a Cross-section

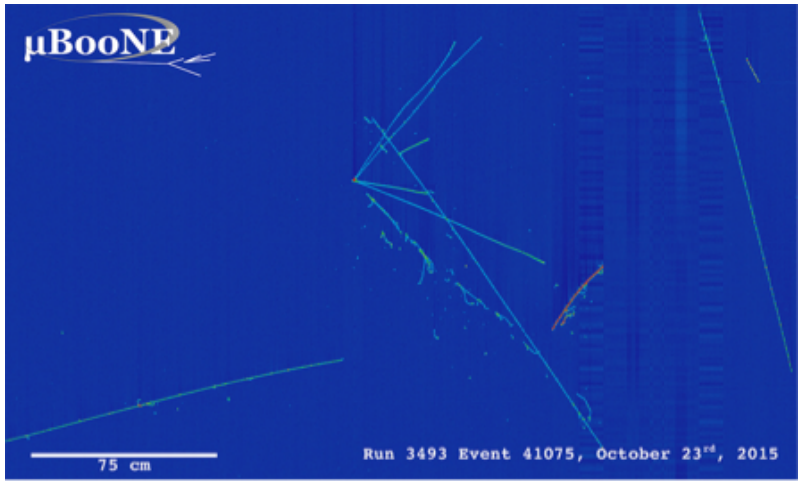
- Exact method used differs from experiment to experiment but the general idea is the same. Lets consider a single bin.

- If we measure a true total event rate for a given topology N :

$$\sigma = \frac{N}{T\Phi}$$

- N : True event rate for a given signal topology.
- T : Total number of targets in our detector
- Φ : Integrated incident neutrino flux



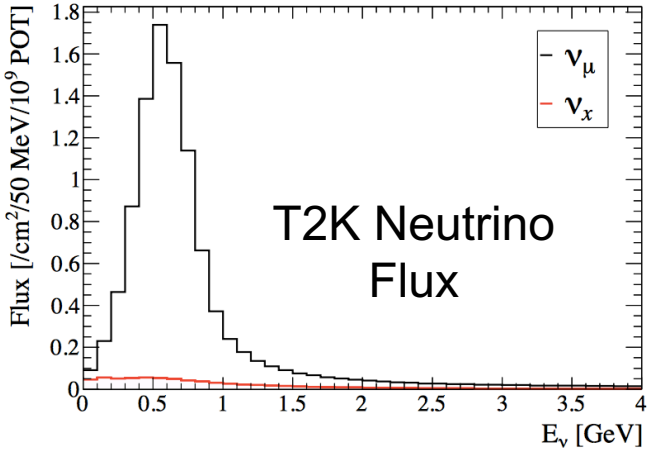


Measuring a Cross-section

- Exact method used differs from experiment to experiment but the general idea is the same. Lets consider a single bin.

- If we measure a true total event rate for a given topology N :

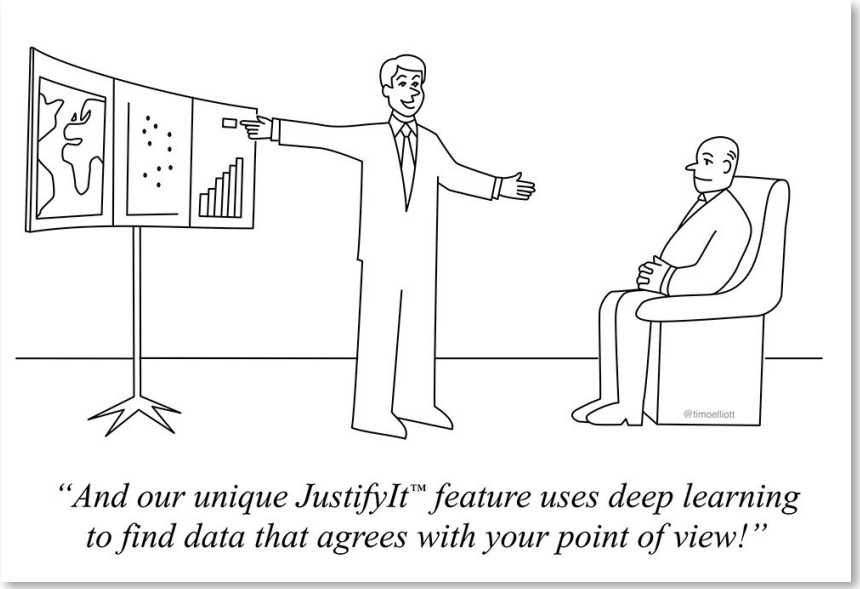
$$\sigma = \frac{N}{T\Phi} \qquad \textit{Uncertainties} \begin{cases} \Delta N \sim \textit{Large} \\ \Delta T \sim \textit{Small} \\ \Delta \Phi \sim \textit{Large} \end{cases}$$



- We have broadband beams and don't see the initial neutrino so flux uncertainties can be large.

- Underestimation of ΔN requires the most consideration by users of data when making model comparisons.

Measuring N events



- **Four main steps to think about**
  1. Choose a signal event topology you care about
  2. Develop a set of event selection criteria to extract your signal whilst rejecting background events
  3. Estimate remaining background contamination
  4. Correct for detector effects

1. Topologies Definition

- Collaborations seem to have different definitions of topologies with the same name ending up in discussions on crossed wires.

People like to be confusing and interchange these

- **Experimental definitions I will use today:**
  - CCQE : Quasi-elastic 1p1h scattering of a nucleon (bound or not)
  - CCQE-like : Quasi-elastic 1p1h or 2p2h scattering
  - CC-0PI : Any event with 1 muon, and no pions in final state

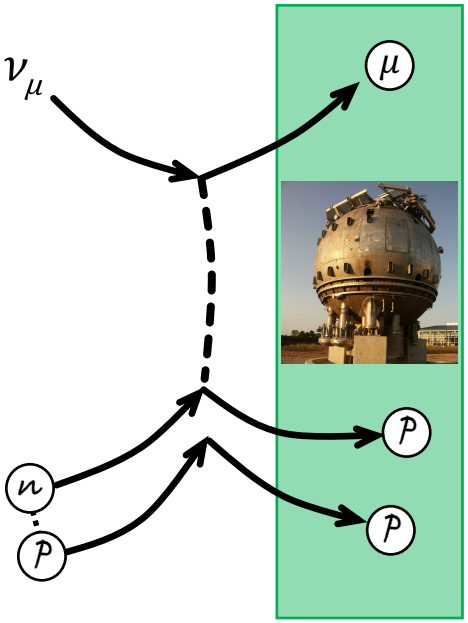
  - CCRES-1PI : Resonant 1 pion production of a nucleon (bound or not)
  - CCRES-NPI : Resonant N pion prouction of a nucleon (bound or not)
  - CC-1PI : Any event with 1 muon, and 1 pion in final state
  - CC-NPI : Any event with 1 muon, and N pions in final state

General rule should be if definition is CC-(Particles) it means it's a final state topology.

Topologies

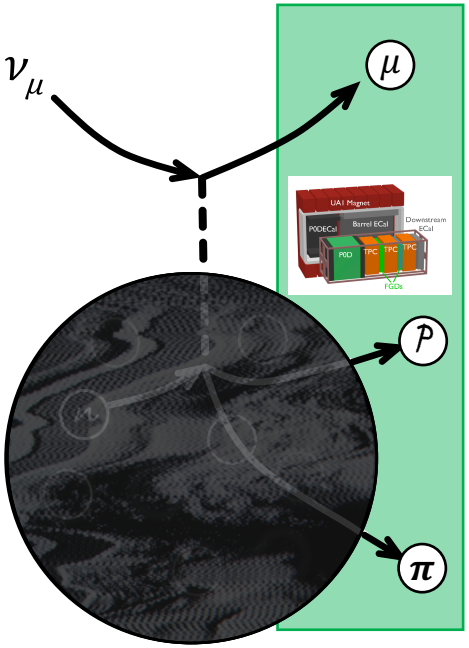
- A signal topology can be any event type we wish to extract a cross-section from.

True CCQE on D^2



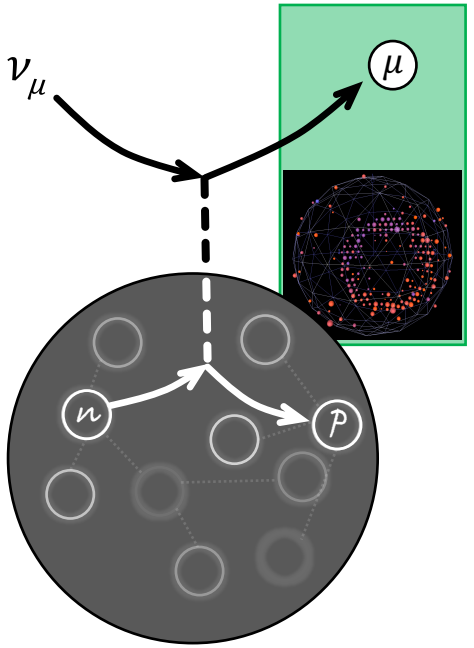
Model Independent

CC1π Final State



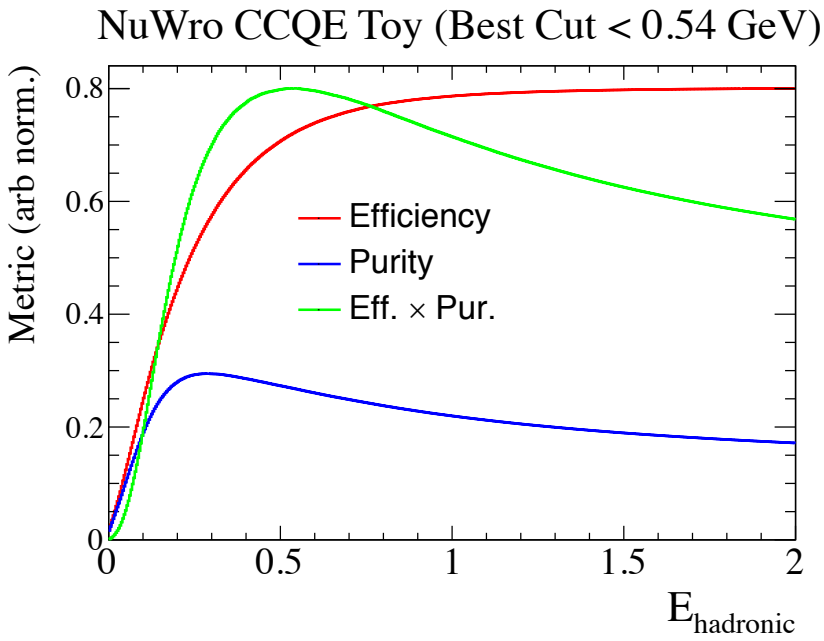
Model Independent

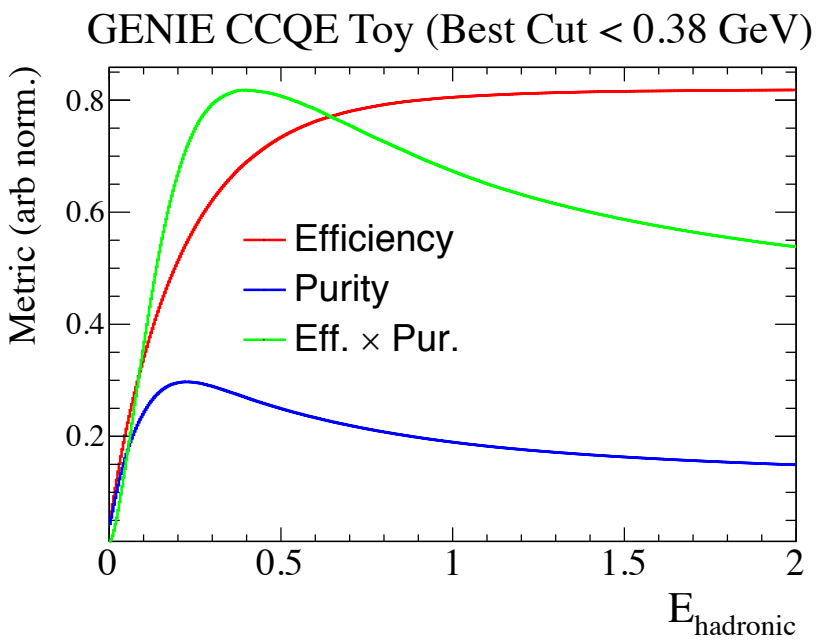
True CCQE on C



Model Dependent

2. Event Selection

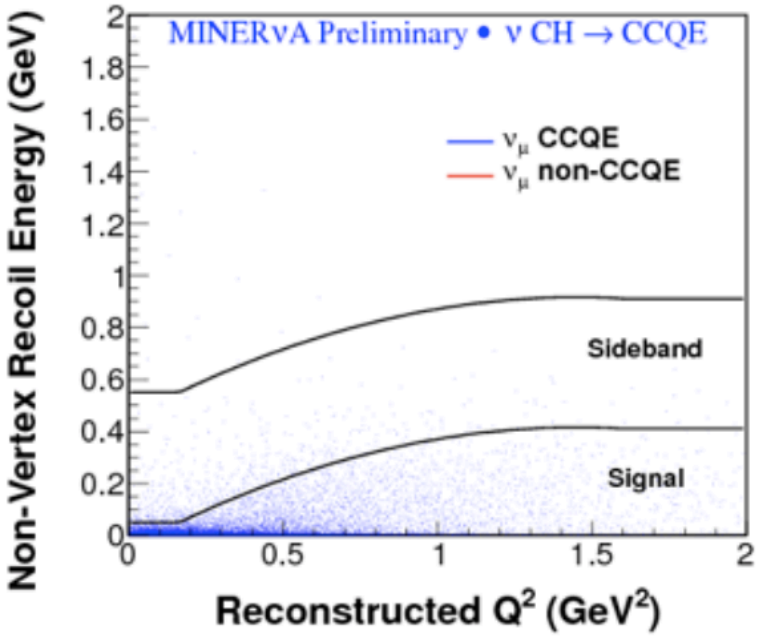


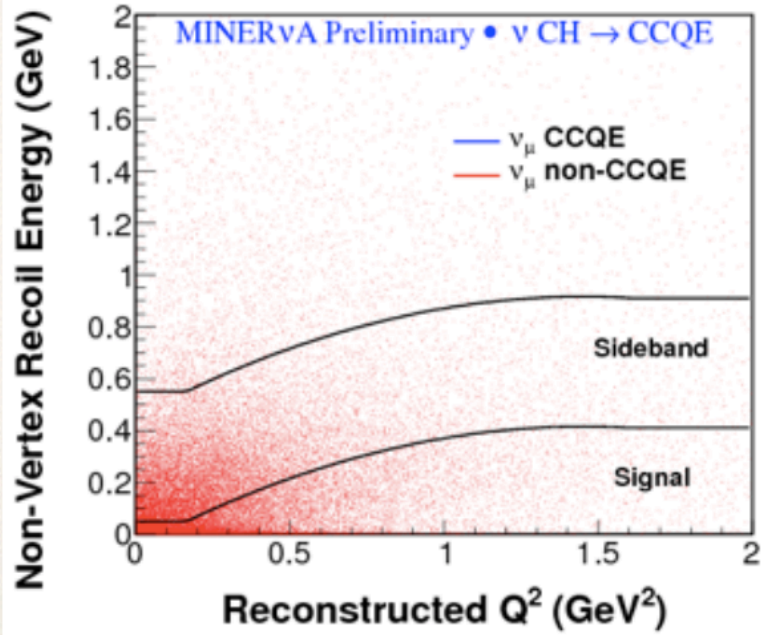


- "Signal" events obtained by cuts on reconstructed event variables.
- Tuned by choosing a cut which maximizes efficiency & purity, or reduces systematic uncertainties.
- **Assumption** : Efficiencies at the chosen cut values do not drastically change with changes in the MC cross-section model.

Selection Cuts Assumption

- Can be difficult to properly assess efficiencies of all possible models.



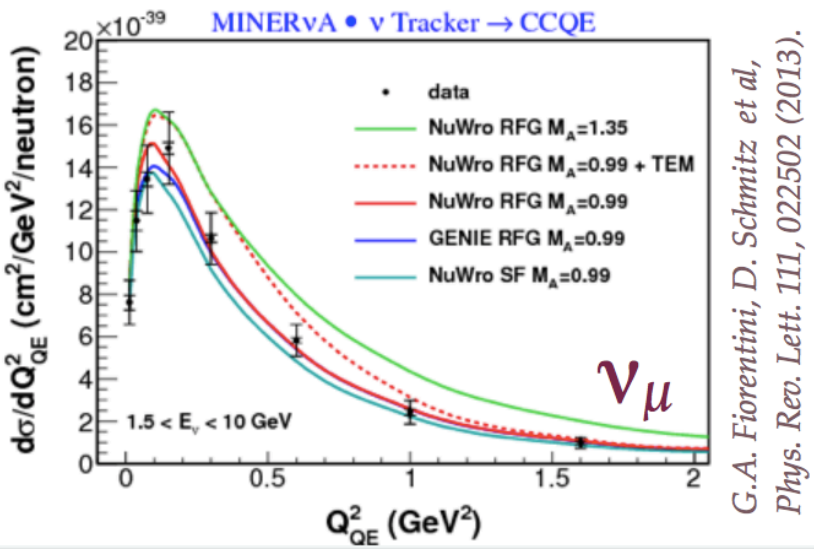


Cheryl Patrick NuInt2014

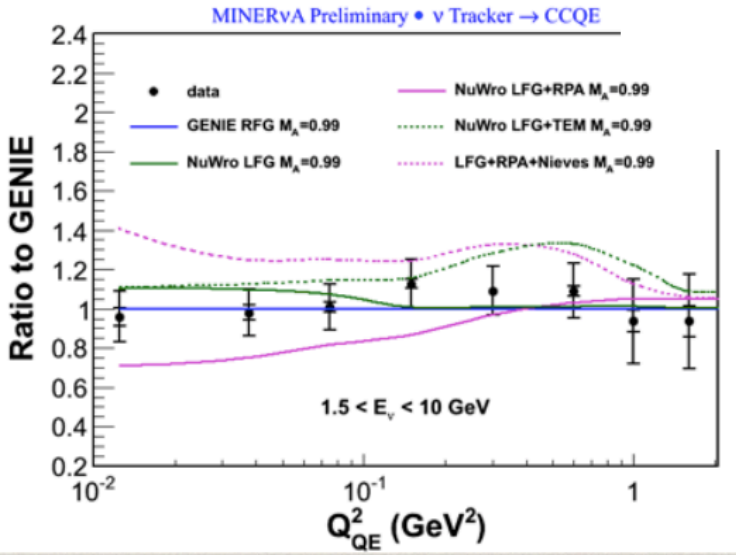
- First MINERvA publications defined their signal as “CCQE”.
- They cut on ”Non-vertex” energy as a function of Q^2. But no GENIE 2p2h model was available at the time, so its unclear how much 2p2h events could be removed by this cut.

CCQE-like Comparisons

- Efficiency corrections are based on CCQE definitions, but comparisons to this data commonly use CCQE-like definition.



G.A. Fiorentini, D. Schmitz et al, Phys. Rev. Lett. 111, 022502 (2013).



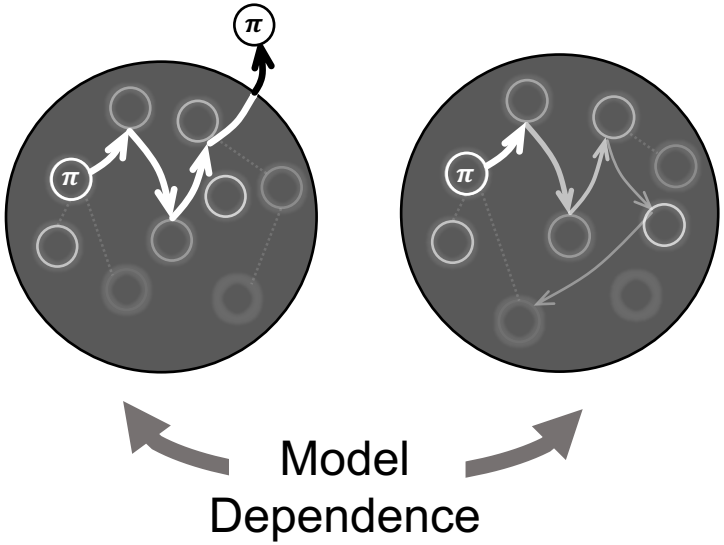
Cheryl Patrick NuInt2014

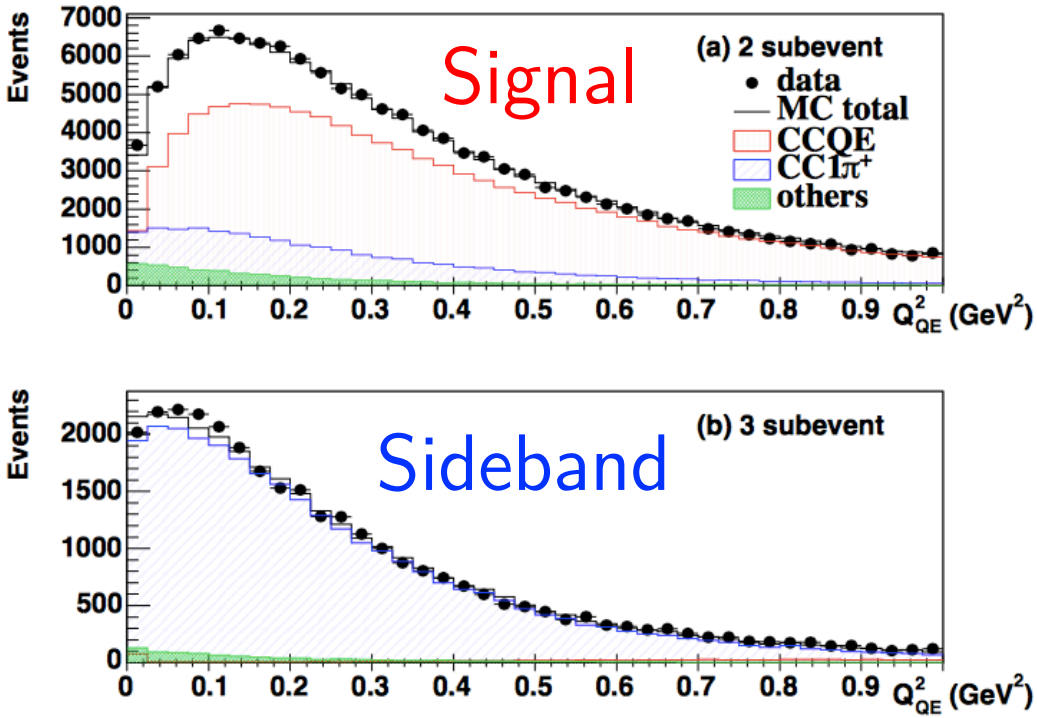
- Possible that the correct comparison to make is somewhere between CCQE and CCQE-like but no way for theorist to tell how much 2p2h was removed from each bin.

3. Background Estimation

- Signal selections are not perfect, will have some associated backgrounds, B, that need to be removed from selected events S

$$\sigma = \frac{S - B}{T\Phi}$$



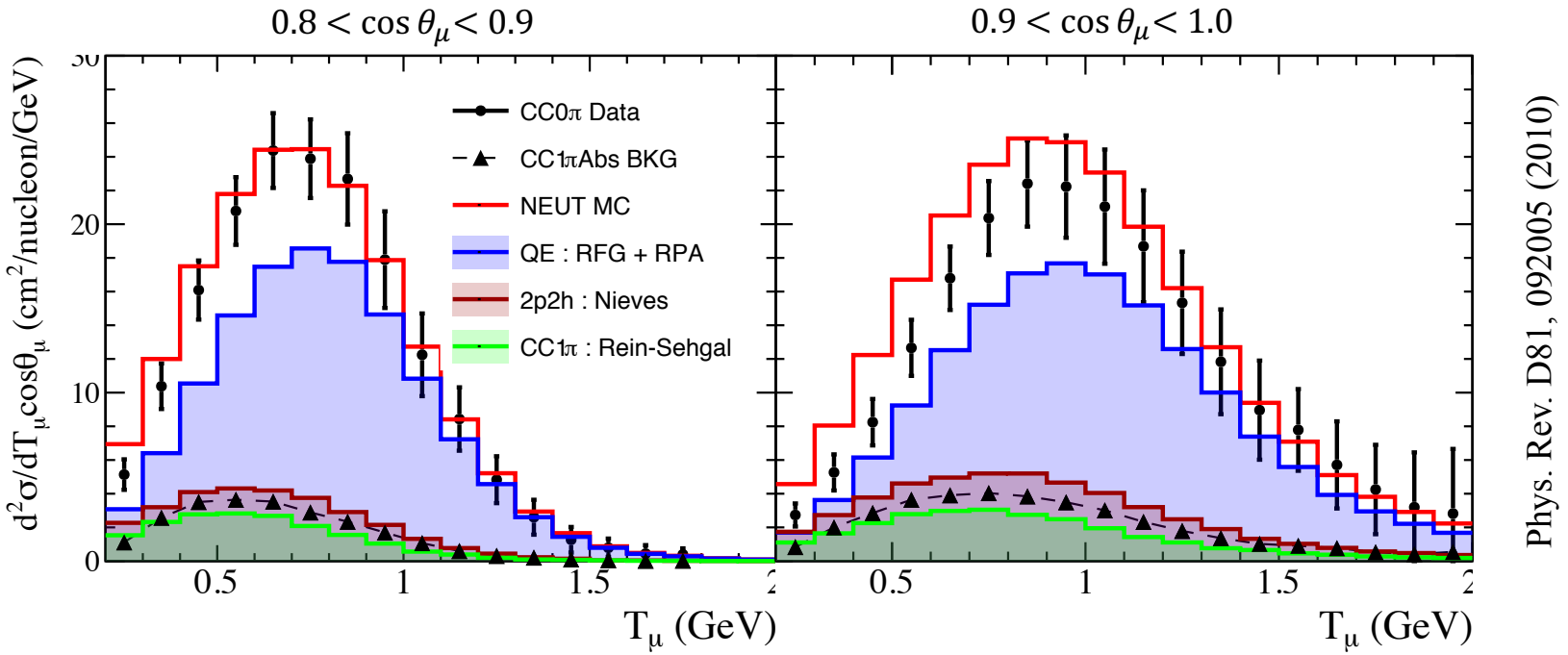


Phys. Rev. D81, 092005 (2010)

- **Assumption** : Sidebands can be reliably propagated into signal region, and uncertainties cover all possible background shapes.

Background Predictions

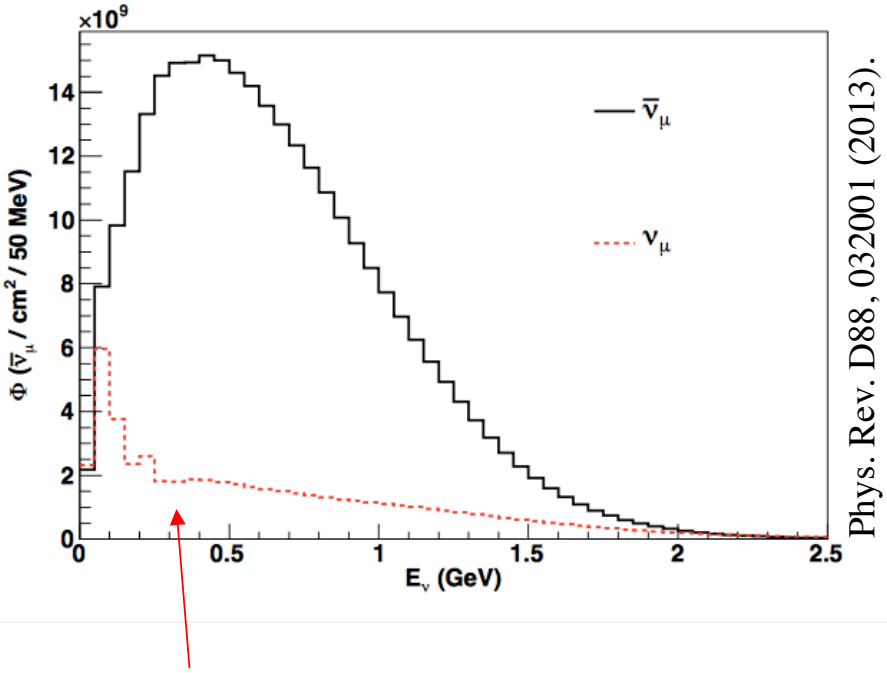
- Subtracted background should depend little on cross-section model.
- MiniBooNE provide model independent cross-section results **without** the pion contribution subtracted from the data to avoid this (CC0PI data release)



- **Datasets like this should always be preferred over model dependent nucleon level measurements. (Nuclear Final State > Nucleon Vertex Level).**

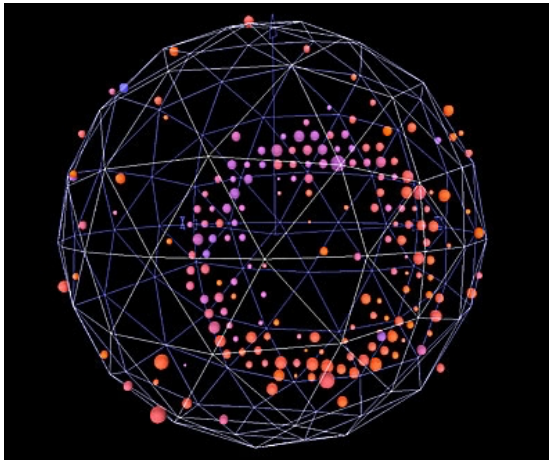
Anti-neutrino Background

- Background subtraction even more awkward in the anti-neutrino case for unmagnetised detectors which can't discriminate on muon charge.

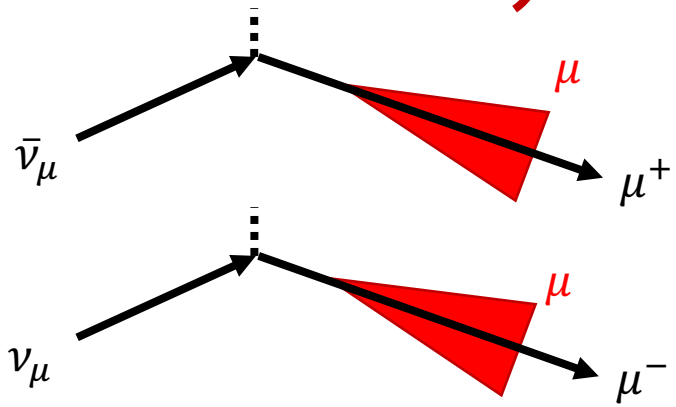


Phys. Rev. D88, 032001 (2013).

Non-negligible neutrino background present in the anti-neutrino beam.

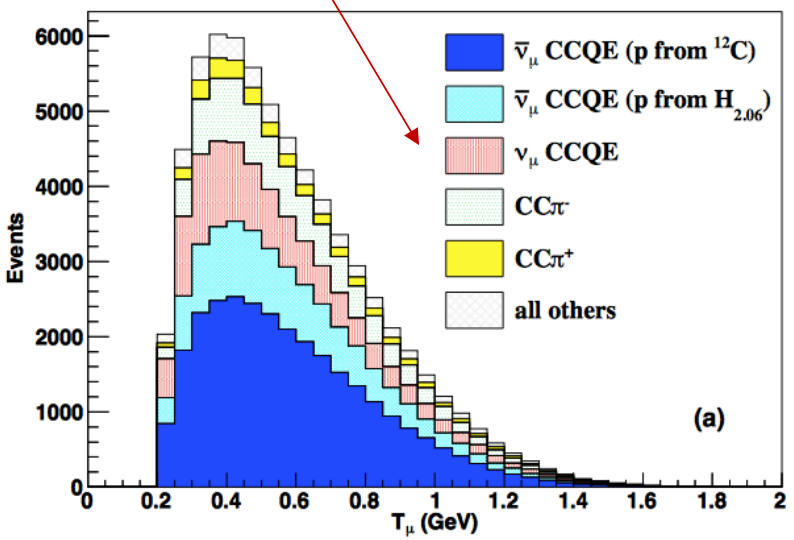


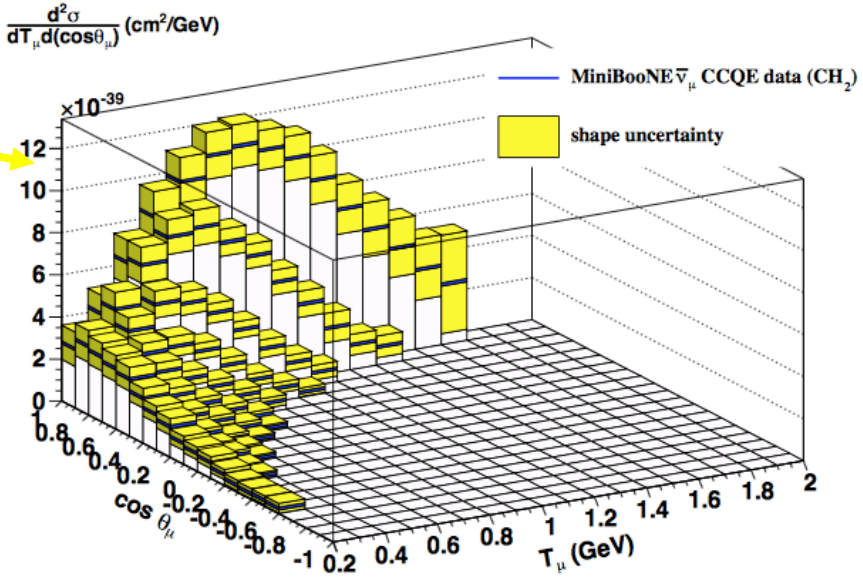
Detector just sees a muon



Anti-neutrino Background (2)

The neutrino signal model is located in the backgrounds!





Phys. Rev. D88, 032001 (2013).

Background Subtracted

- MiniBooNE anti-neutrino CCQE analysis has to first tune the model to what they think the neutrino-CCQE cross-section should look like.
- Anti-neutrino data is correlated with the neutrino measurement, and has implicit dependence on the CCQE model in it. Use with caution!

4. Detector Effects

- Detectors are not perfect, and neither are selections. Number of events successfully reconstructed, R, unlikely to match true rate.

$$R = (S - B) \neq N$$

- Have to correct the number of reconstructed events ("unfolding") accounting for smearing and efficiencies of the detector.

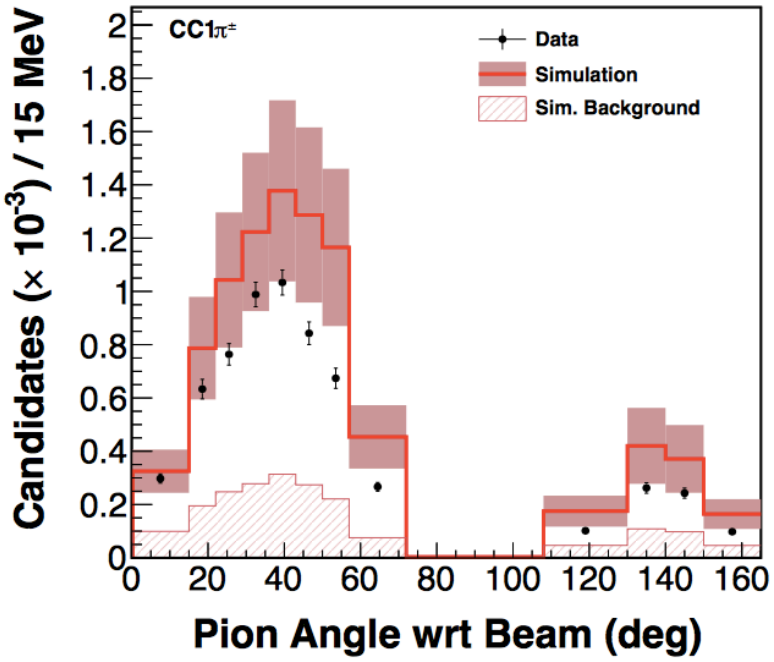
$$N = \frac{R}{\varepsilon} = \frac{(S - B)}{\varepsilon}$$

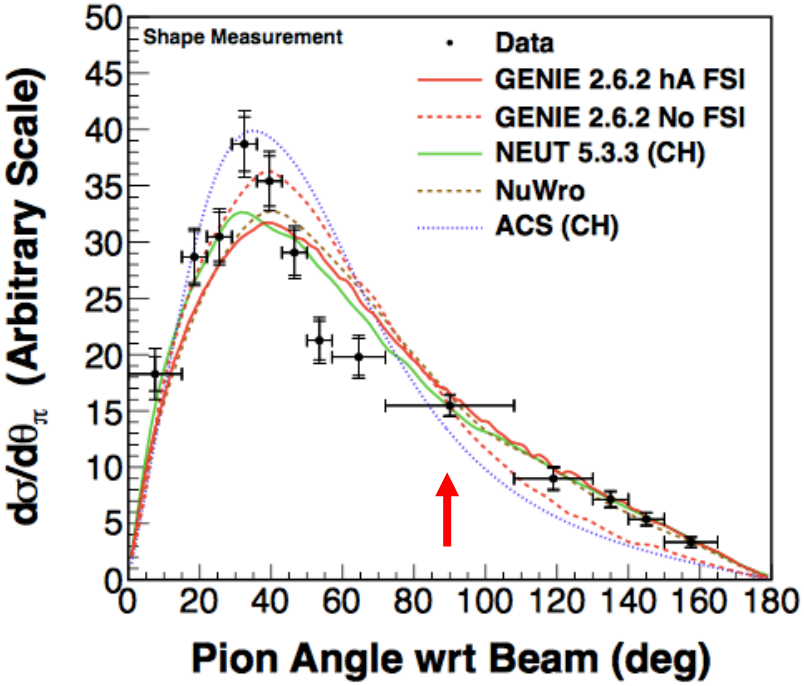
Efficiency in our single bin

- Assumption : Again relies on your efficiency correction being as model independent as possible.

MINERvA CC1π^+

- Particularly a problem in regions of very low efficiency.



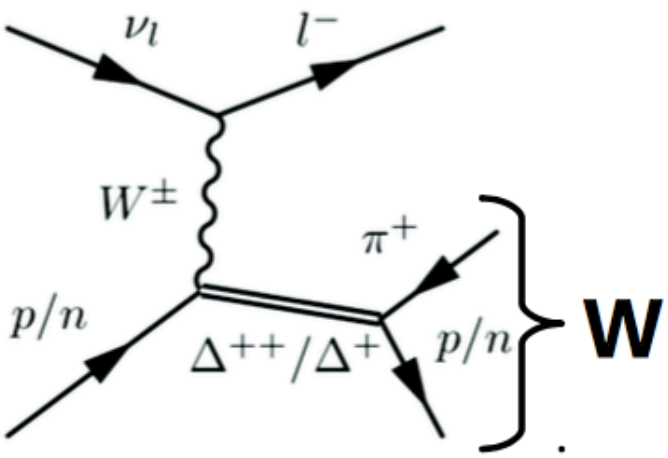


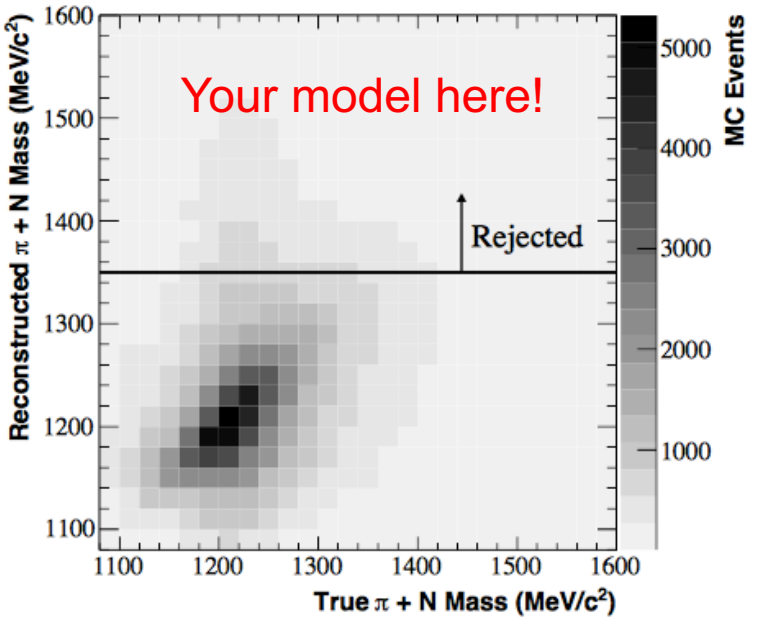
Phys.Rev. D92, 092008 (2015)

- First MINERvA CC1π^+ data had one angular bin with zero efficiency.
- Unfolded bin heavily dependent on GENIE shape prediction.

MiniBooNE CC1π^+

- MiniBooNE CC1pi+ analysis cuts on the invariant mass to remove backgrounds.
- Cut is not reflected in the true MC signal definition.





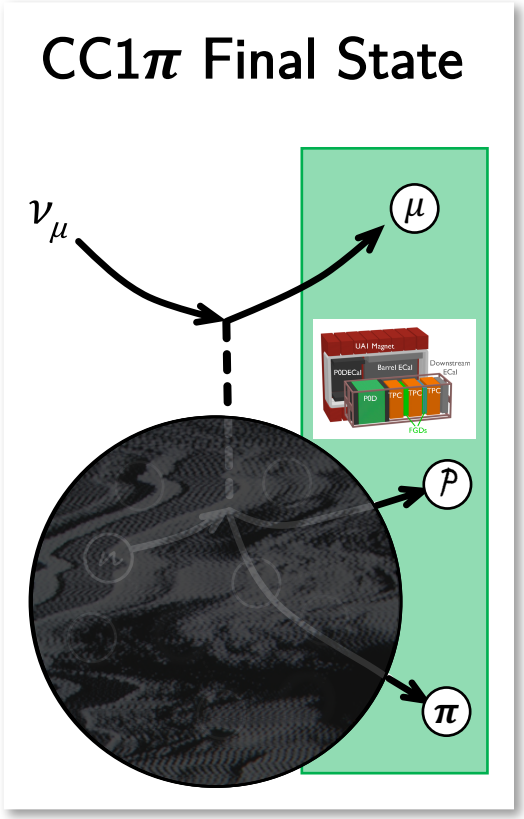
arXiv:1011.3572 [hep-ex]

- Efficiency corrections then fill these events back in using the MC prediction.
- Estimate ~30% correction to published cross-section comes from NUANCE.

Avoiding Dependence

- Has been a large shift in the community to change the definitions of signal topologies to "Final State" topologies to avoid model dependent background subtractions.

- If we look at events based on what came out of the nucleus, and only in the regions our detector can see we reduce dependence on the neutrino cross-section model in our extraction.



- Advantage : No FSI model dependent background subtractions.
- Disadvantage : Restricted phase-space after FSI so more difficult to compare individual channels to data (requires a MC generator in some cases).

Extraction

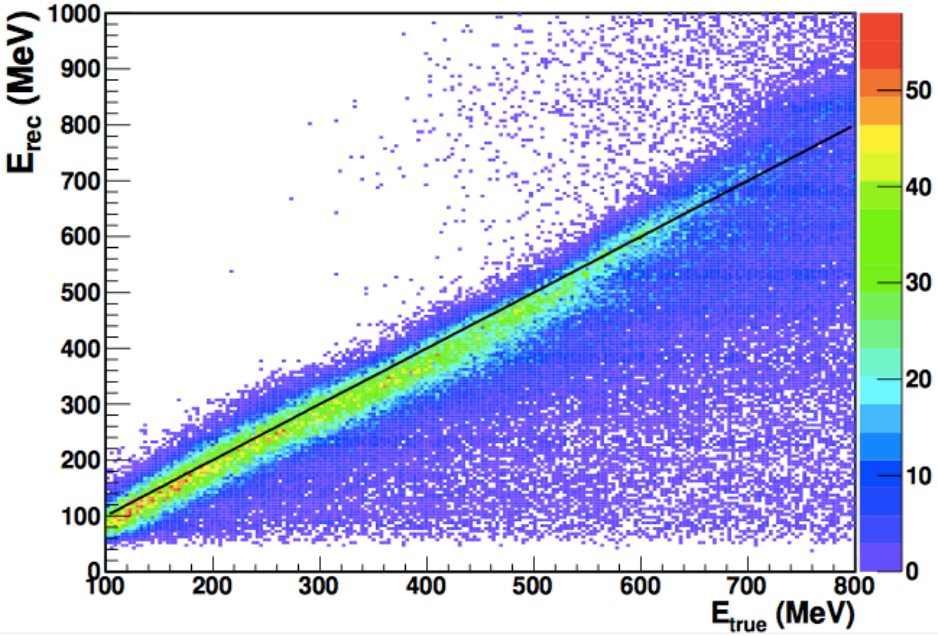
- This method is actually a simplification for only a single bin.

$$\sigma = \frac{N}{T\Phi} = \frac{(S-B)}{\varepsilon T\Phi}$$

- Relies on a number of questionable assumptions:
  1. Good understanding of detector efficiencies
  2. Good understanding of selection efficiencies
  3. Good understanding of backgrounds

- In the case of N-bins >1 choices of unfolding techniques and how they impact the results also need to be considered.

Handling Nbins > 1

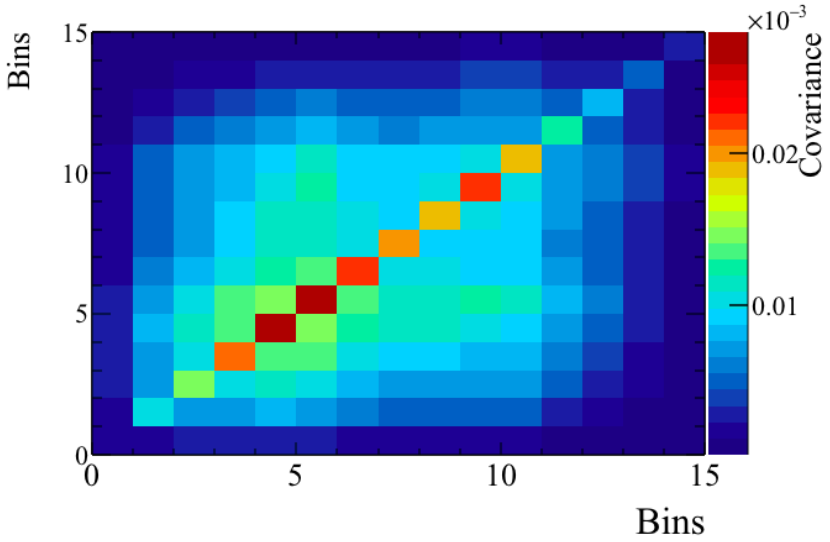


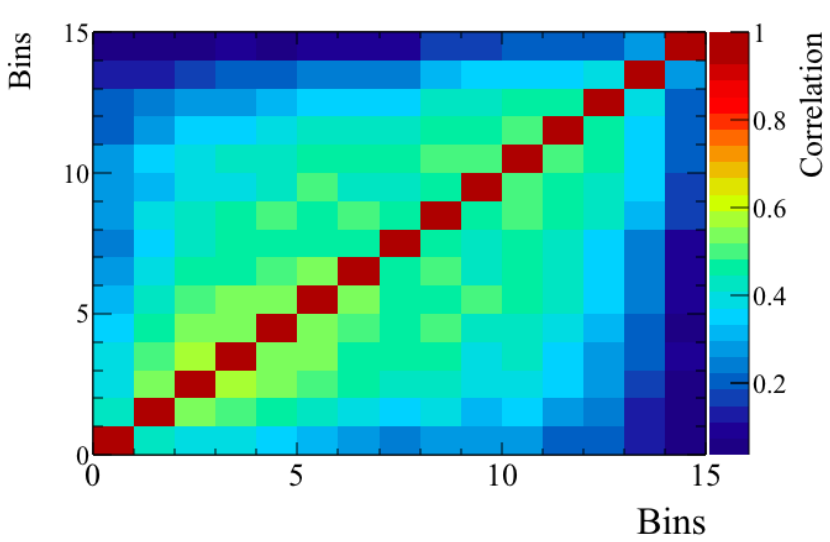
D. Perevalov, FERMILAB-THESIS-2009-47".

- When measuring distributions with many bins experiments need to "unfold".
- Removes detector smearing and efficiency effects and tries to get at the true underlying distribution.
- Usually have to use a regularization method to avoid statistical fluctuations in the reco data being blown up into large features in the unfolded data.

Covariance

- Systematic uncertainties in neutrino cross-section extraction introduces strong correlations between bins.





Phys. Rev. Lett. 116, 081802 (2016)

- Means calculating χ^2 without accounting for correlations can be very biased. **Making "by-eye" comparisons to data is misleading**.

$$\chi^2 = \sum \left(x_i^{data} - x_i^{MC}\right) \left(M^{-1}\right)_{ij} \left(x_j^{data} - x_j^{MC}\right)$$

Summary

- Neutrino cross-sections rely on good understanding of detector efficiency and its dependence on the model to reliably extract a signal.
- Users of data need to be wary of how differences in experimental methods could be shaping the signal.
- Shift in the past few years for experiments to produce "restricted phase space" results with more complex signal definitions.
  - Advantage : Minimise model dependence in results.
  - Disadvantage : need to use MC generators to reliably compare theoretical models to newer measurements.

NUISANCE

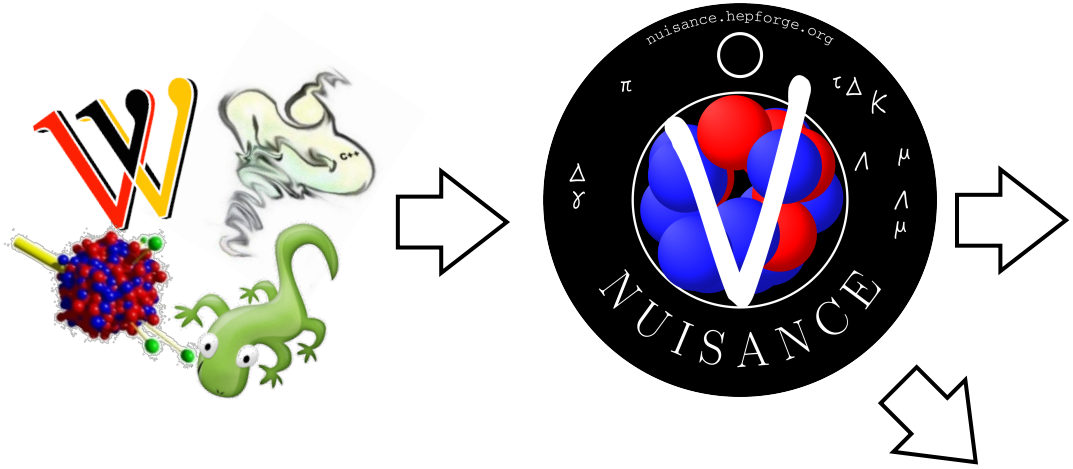
- Growing need in the community to have a place to consistently test generator models through comparisons to each other and data.

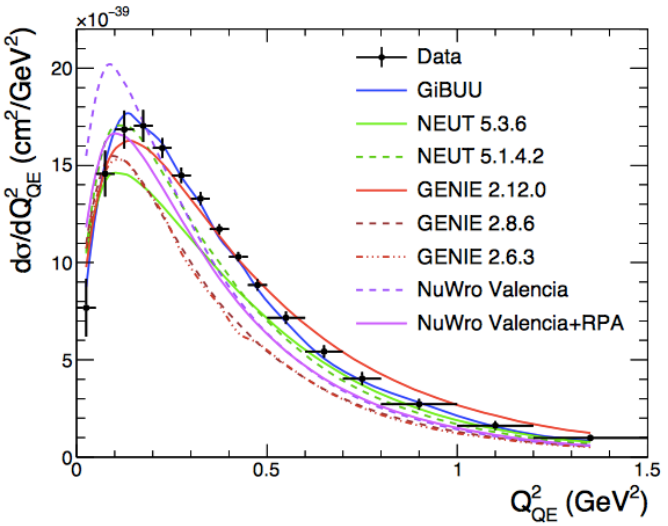
- Several approximations usually need to be made when going from a theory model to a generator implementation. Need to test implementations can describe the relevant phase-space as expected.

> *There is a hope, however, that a joint global fit to the existing data could reduce the [cross-section model] uncertainties. When tuning generators in this kind of global fits, a mechanism for examining "tensions" in datasets should be established. A useful goal would be a universal or global tune as achieved by QCD global fits of parton distribution functions.*

NuSTEC White Paper : arXiv:1706.03621

NUISANCE in Nutshell

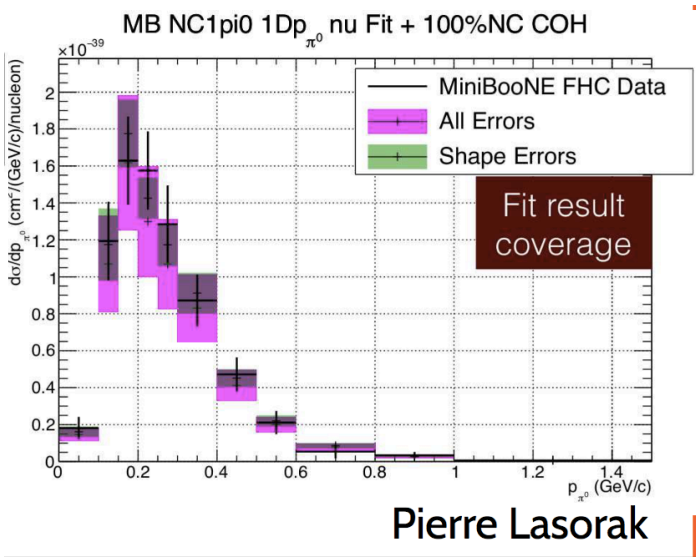




https://nuisance.hepforge.org

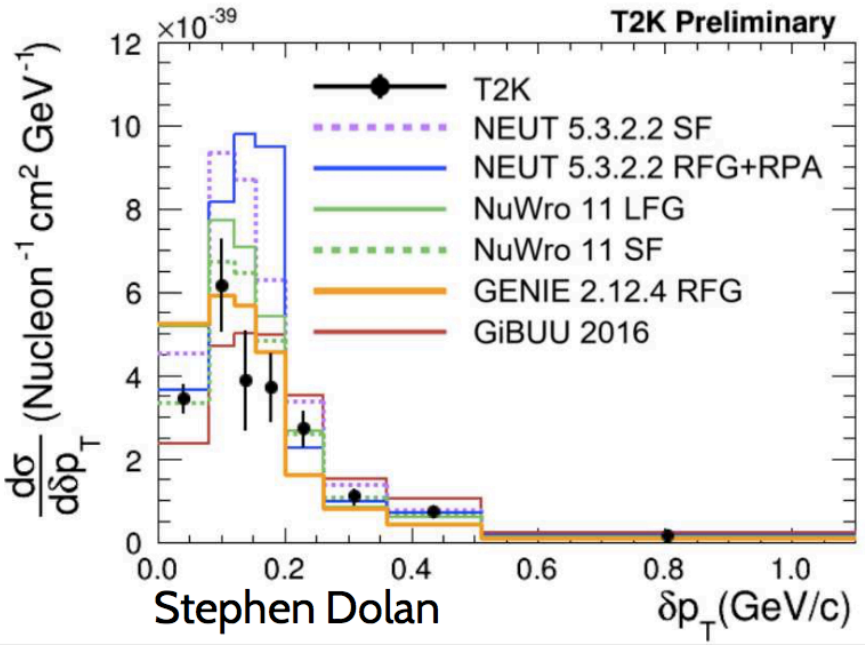
arXiv:1612.07393

- Converts generators to common format
- Automated comparisons to exclusive data
- Estimates systematic uncertainty
- Runs global tunes to scattering data



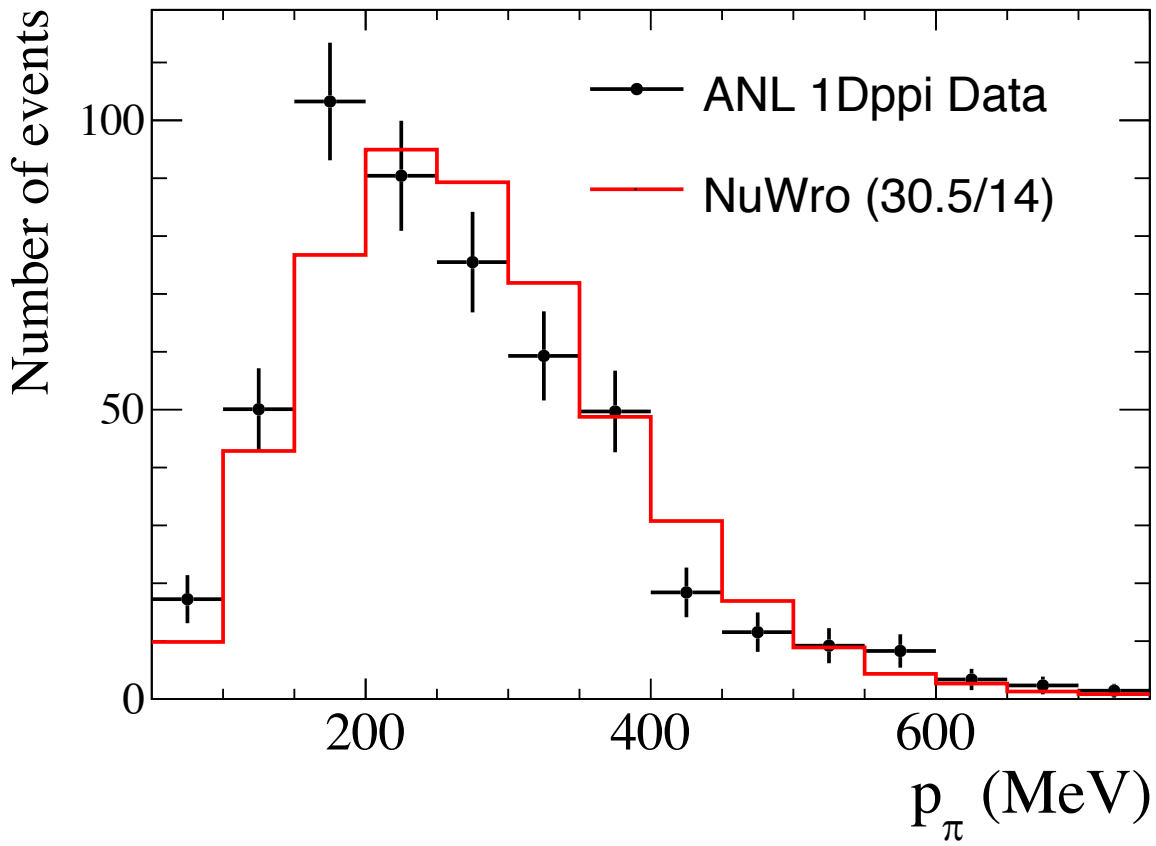
NUISANCE Relevance

- **Relevant for both cross-section analysers and model builders.**
  - Compare many different generators at once to data.
  - Easily look at effects of free model parameters on kinematic distributions.
  - Constrain additional interactions that you don't have a sideband for.
  - Add new measurements into our database to validate them/help the community use them/make pretty thesis plots!



Example

- Generate NuWro ANL events and compare them to CC1pi+1p pion momentum bubble chamber data in NUISANCE.



Solution (1)

Need to generate some inclusive NuWro events for this example

Make sure NUISANCE is setup! (Build type dependent)

```
$ source $HOME/NUISANCEMC/nuisance/v2r8/build/Linux/setup.sh
$ source $NUWRO/build/Linux/setup.sh
```

```
$ emacs params.txt
```

Open a new params file

```
@beam/ANL.txt
nucleus_p=1
nucleus_n=1
nucleus_target=0
kaskada_on=0
pauli_blocking=0
```

Save these lines into it

Tells NuWro to generate free nucleons target

```
$ nuwro -i params.txt -o ANL-nuwro-events.root
```

Run nuwro with new params file

```
$ PrepareNuwro -f ANL-nuwro-events.root
```

Prepare new events for NUISANCE

Solution (2)

Can now compare prepared events in NUISANCE easily.

```
$ emacs ANL.xml
```

Lets write a new NUISANCE card file

Have to tell NUISANCE its a NUWRO file

```
<nuisance>
  <sample name="ANL_CC1ppip_Evt_1Dppi_nu" input="NUWRO:ANL-nuwro-events.root" />
</nuisance>
```

Dataset name

Pass our new prepared event file in

Can search names for other datasets using nuissamples

```
$ nuissamples [substring]
```

```
$ nuissamples CC1ppip
ANL_CC1ppip_XSec_1DEnu_nu
ANL_CC1ppip_XSec_1DEnu_nu_W14Cut
ANL_CC1ppip_XSec_1DEnu_nu_Uncorr
```

Solution (3)

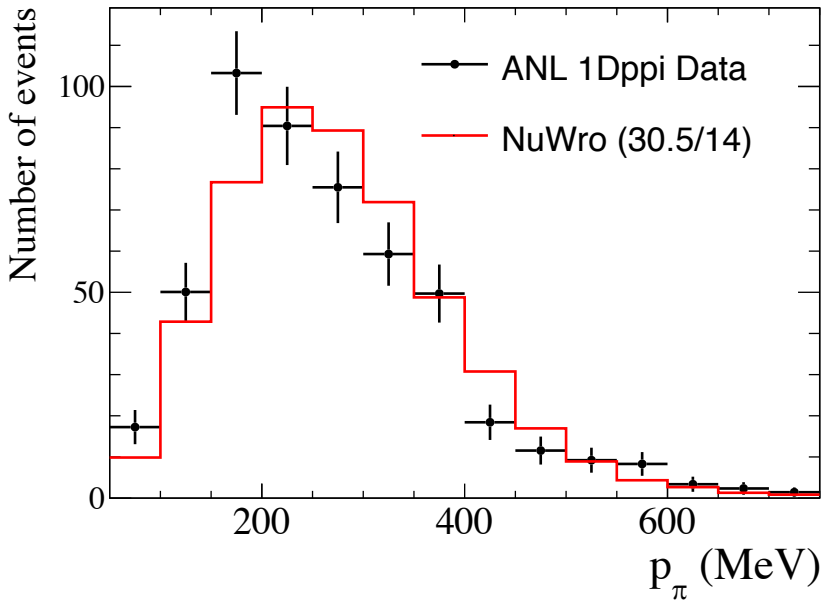
Can now compare prepared events in NUISANCE easily.

```
$ nuiscomp -c ANL.xml -o ANL.root
```

Run the standard comparison app

```
$ root ANL.root
root [0] ANL_CC1ppip_Evt_1Dppi_nu_data->Draw("E1");
root [1] ANL_CC1ppip_Evt_1Dppi_nu_MC->Draw("SAME HIST C")
```

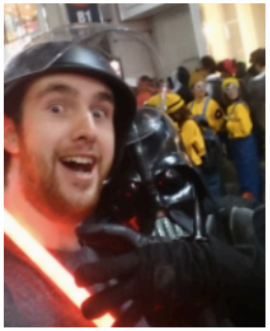
Compare histograms in TBrowser



NUISANCE Future

- Framework is open source under the GPLv3 license. We want the community to take it and use it as they see fit.
- The only way we achieve the uncertainty goals of future neutrino experiments is through a joint community effort to ensure what is being used in the MC generators is actually theoretically consistent and agrees with data.

Imperial College London



The University Of Sheffield.

MICHIGAN STATE UNIVERSITY

UNIVERSITÄT BERN

Towards Global Fits

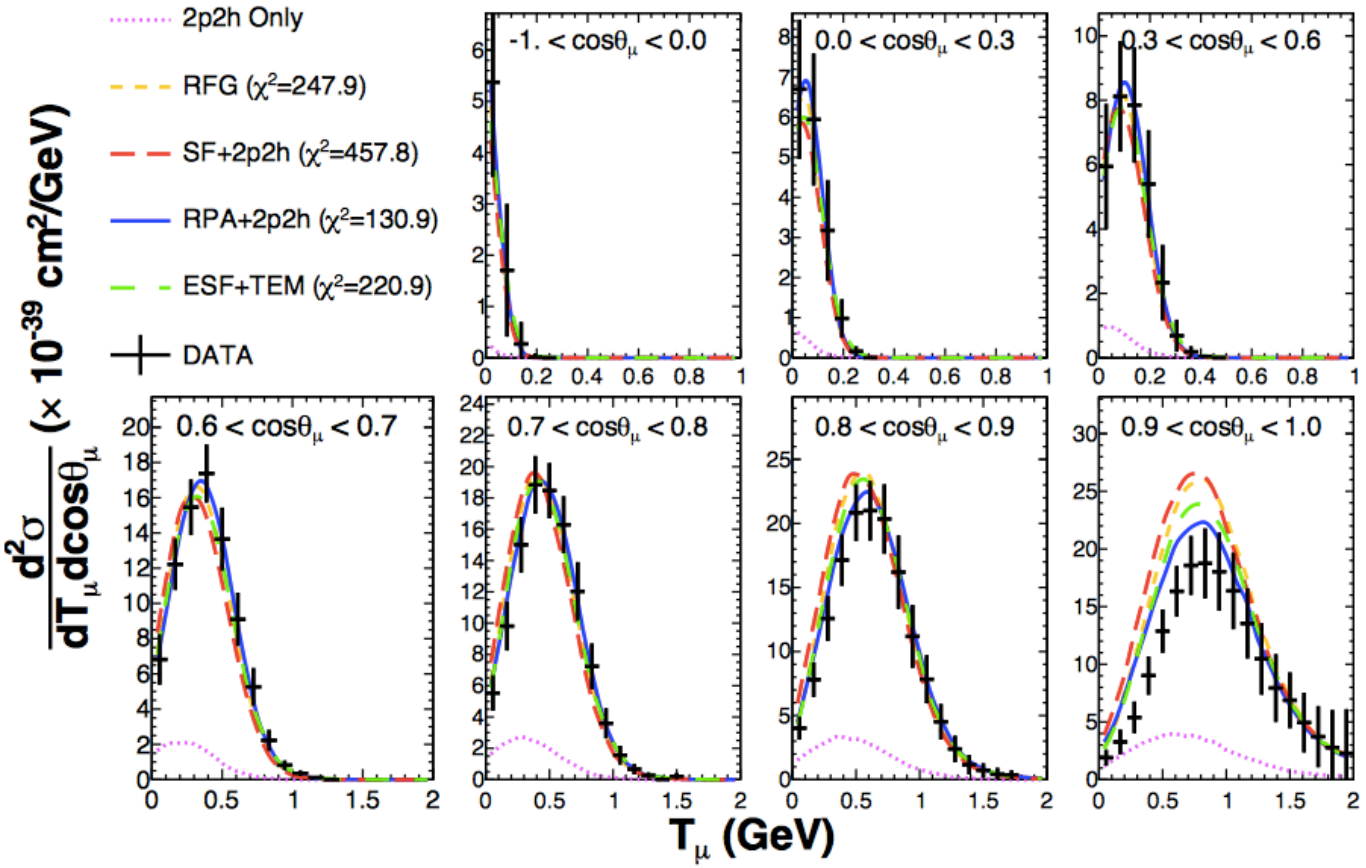
Clarence Wret, Patrick Stowell, Luke Pickering,
Callum Wilkinson

15/10/17

T2K

NUISANCE Introduction

- Grew out of efforts on T2K to tune NEUT to external cross-section data to produce model uncertainty inputs for the long baseline oscillation.



arXiv:1601.05592

What is inside NUISANCE?

1. Generator Convertors
   - Direct interface for generators and their corresponding reweight engines.

2. Measurement Base
   - Automated Signal Selections
   - Data/MC Comparisons
   - Likelihood Calculations

3. Analysis Routines
   - Model tuning
   - Systematics predictions
   - Interface with Migrad.
   - Effective detector smearing

Generator Convertors

- At the core of NUISANCE is a set of routines that convert generators into a common format.
- Designed with final state particle analysis routines in mind.

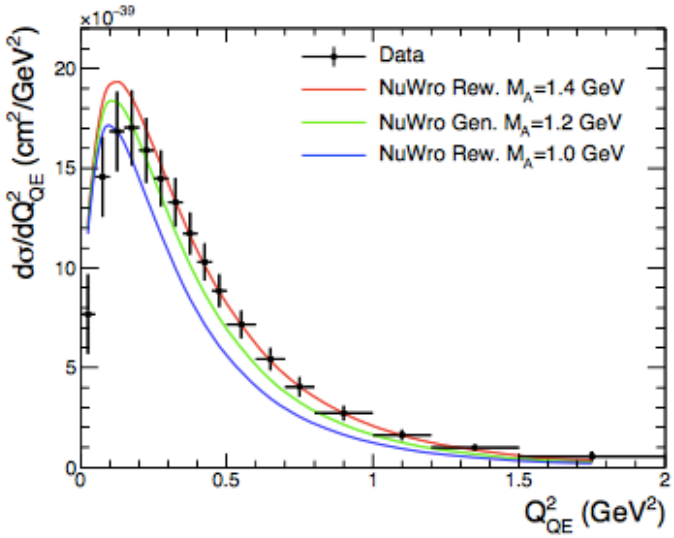
GeneratorEvent **InputHandlers** **Simple FitEvent**

- FitEvent class is a common generator event wrapper

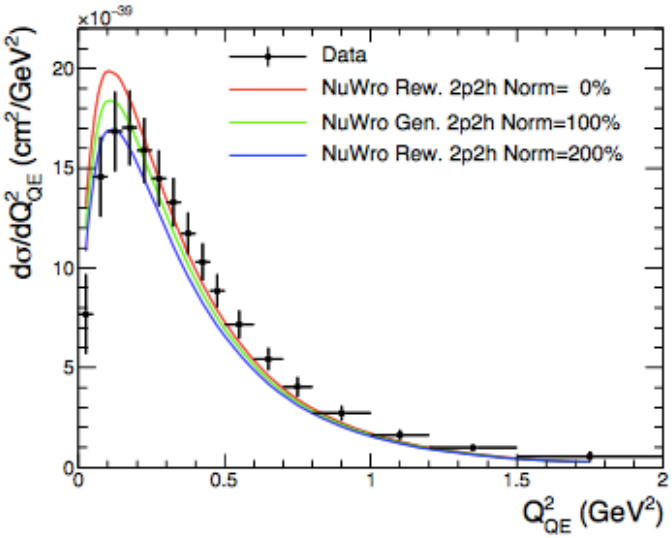
Event Reweighting

- FitEvent contains the original generator event that it was derived from enabling ReWeight support (e.g. GENIEReWeight, T2KReWeight).

Model Variations in NuWro ReWeight



(a) M_A variations



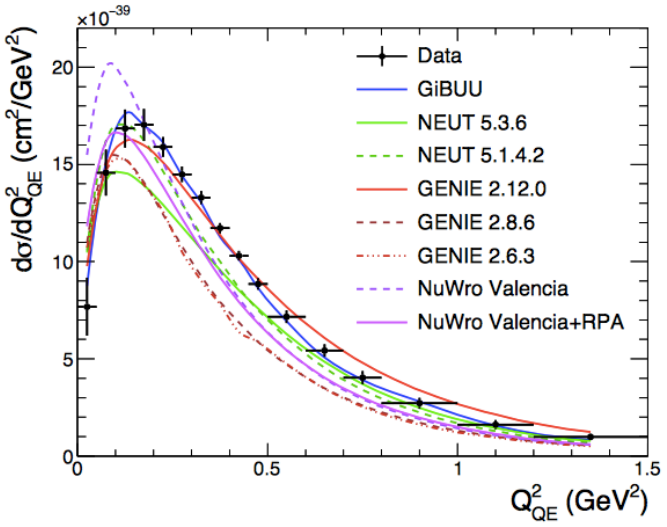
(b) 2p2h variations

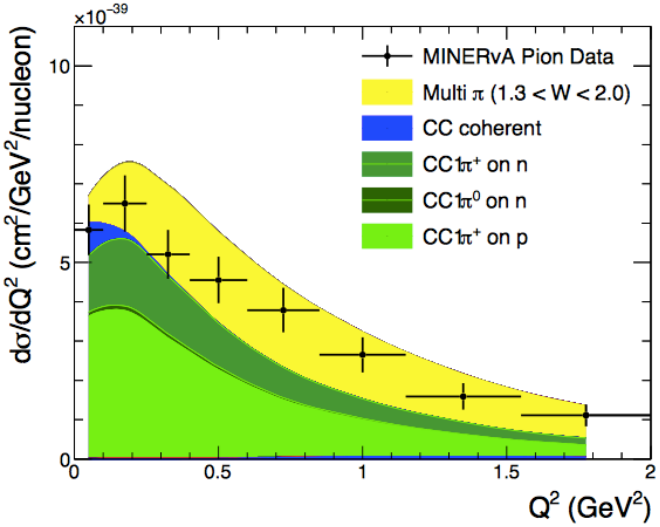
Event weight for binning in histograms →

$$W(\vec{e}) = \frac{\sigma(\vec{e},\vec{p})}{\sigma(\vec{e},\vec{g})}$$

σ : Cross-section
$\vec{e}$: Event kinematics
$\vec{g}$: Generated Model
$\vec{p}$: New Model

Common Formats





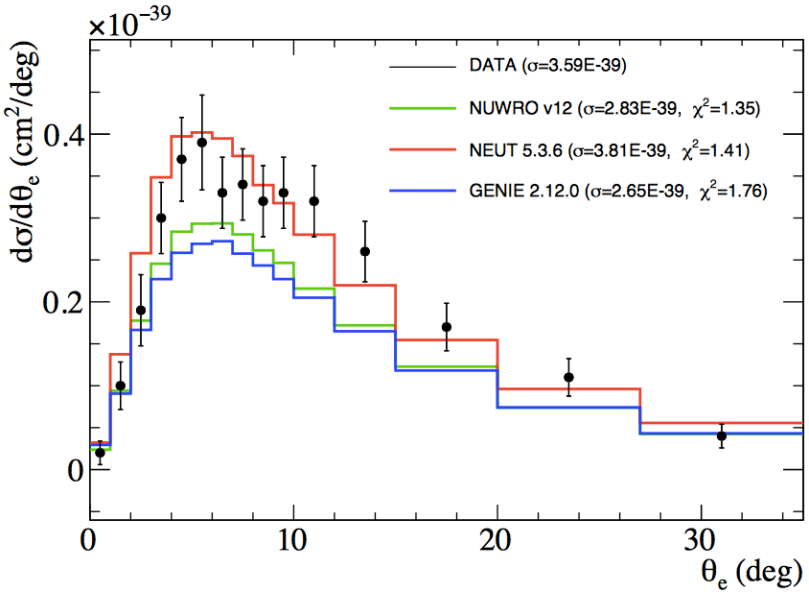
- FitEvent/FitWeight class design tries to keep any event analysis that happens above the core InputHandlers as generator independent as possible.
- Cross-section data shouldn't depend on the generator. Method we use to compare MC events to the data also shouldn't.
- **Design provides two advantages:**
  - Any new analysis automatically supports all generators.
  - Any new generator automatically supports all analyses.

Measurements

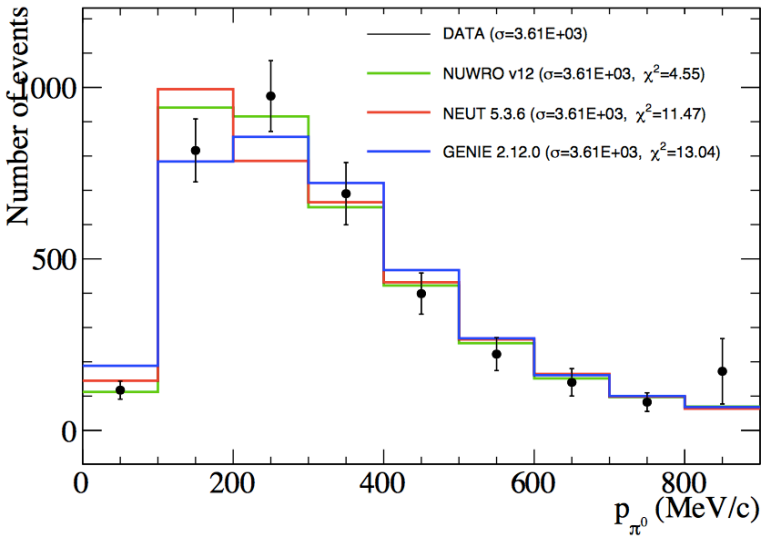
- Broad range of different cross-section neutrino measurements implemented into the NUISANCE framework.

```
<sample name="MINERvA_CC0pi_Xsec_1DThetae_nue"
        input="GENIE:@GENIE_DIR/gntp.MIN.nue.root" />
<sample name="K2K_NC1pi0_Evt_1Dppi0_nu"
        input="GENIE:@GENIE_DIR/gntp.K2K.numu.root" />
```

MINERvA CC0pi XSec 1DThetae nue data



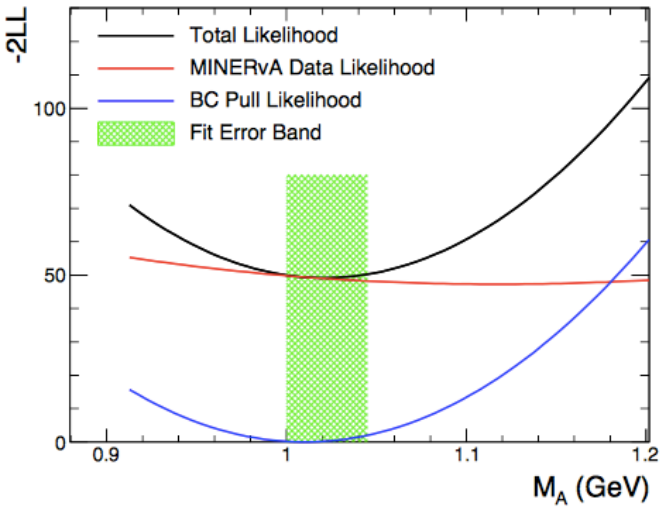
K2K NC1pi0 Evt 1Dppi0 nu data

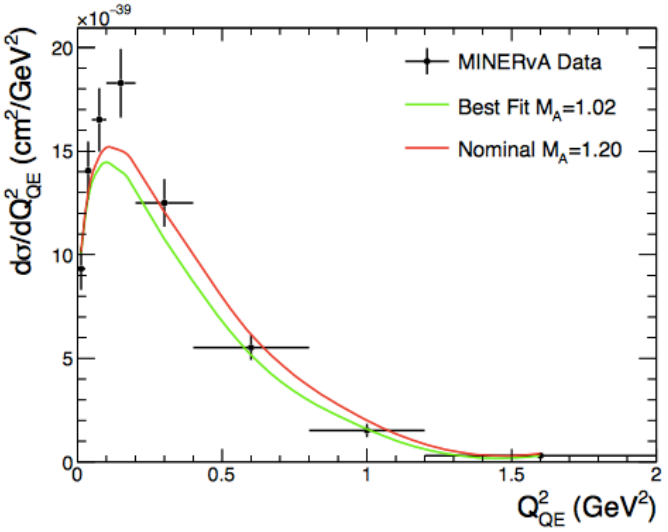


https://nuisance.hepforge.org/files/validation/nuisancevalidation_v1r0_280217/nuisance_v1r0_validation_280217.pdf

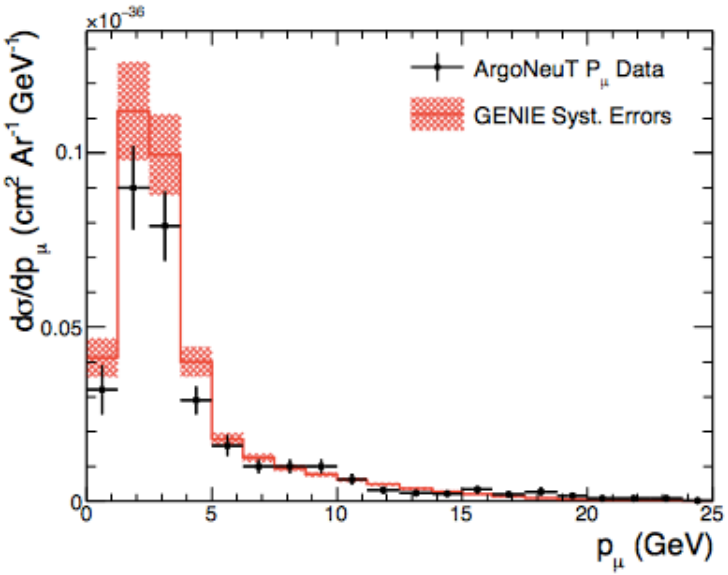
Generator Tuning

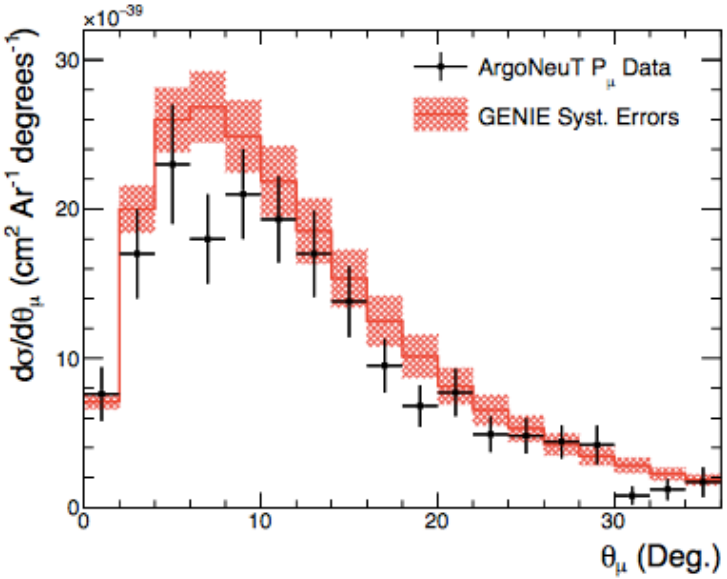
- Tools provided to perform joint fits between any measurement class included in the framework.
- Used extensively by T2K to extract constraints in NEUT on CC0PI and CC1PI scattering data.
- Support for more advanced non-linear parameter space searches, arbitrary parameter priors, measurement cross-correlations.





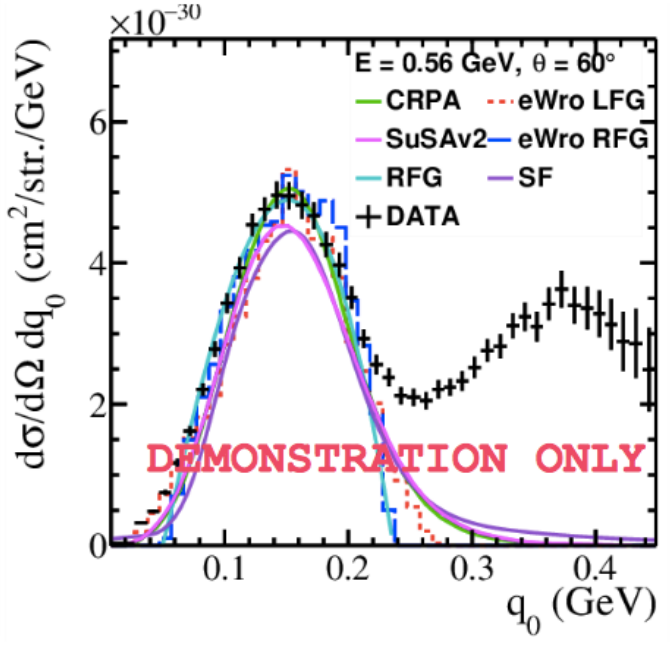
Systematic Analysis

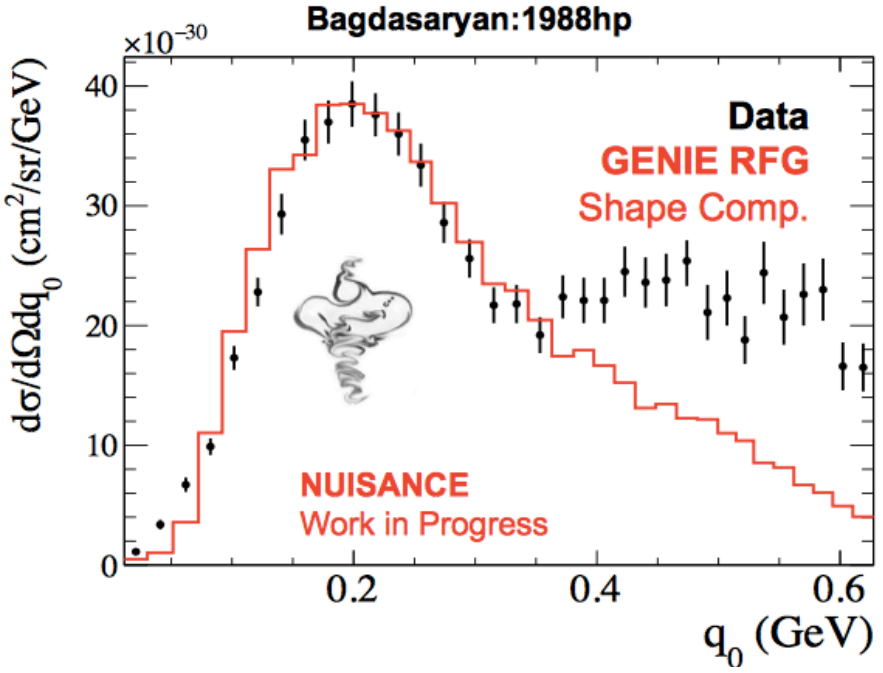




- Many experiments use GENIE's provided 1-sigma uncertainties to define their error budget in experiments.
- Can pass arbitrary lists of dials and constraints to NUISANCE and it will generate systematic error bands for the model.
- Useful to know how much coverage of the data the uncertainties being used actually provide.

NUISANCE Future



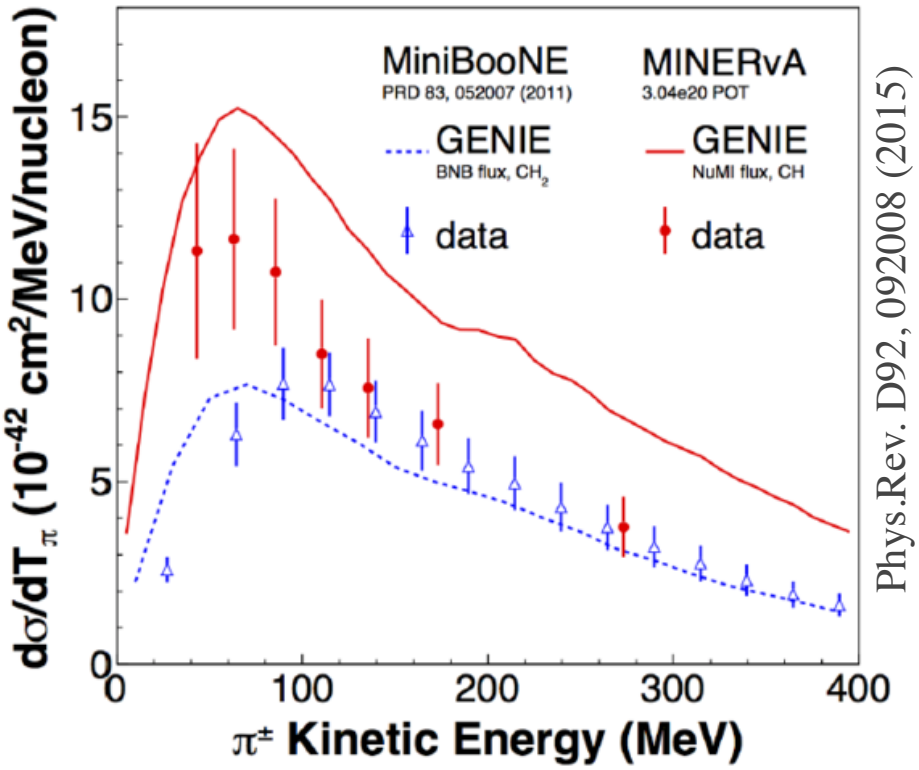


- End goal of the NUISANCE project is to define a set of standard unit tests that we benchmark generators against.
- Not just limited to neutrino scattering data. Already started work to add the huge amount of existing electron scattering databases, and perform studies of CLAS data with NUISANCE.

NUISANCE Comparisons

- Already trying to use NUISANCE to develop a suitable benchmark
- Lets consider two different NuSTEC problems mentioned in the white paper.

- Tensions seen between GENIE and MINERvA and MiniBooNE CC1pi+.
- Pion data-MC disagreement considered by some to be due to deficiency in the nuclear/FSI models in generators.
- Can we make an updated plot in NUISANCE?



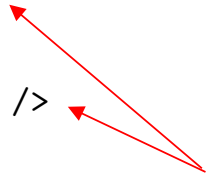
Phys.Rev. D92, 092008 (2015)

Comparison Guide

```
$ emacs comparison.xml
```

```
<nuisance>
  <sample name="MINERvA_CC1pip_XSec_1DTpi_nu_2017"
          input="GENIE:gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH.500000.ghep.root" />

  <sample name="MiniBooNE_CC1pip_XSec_1DTpi_nu"
          input="GENIE:gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root" />
</nuisance>
```



Can cheat and use two different models because pion model same in both datasets.

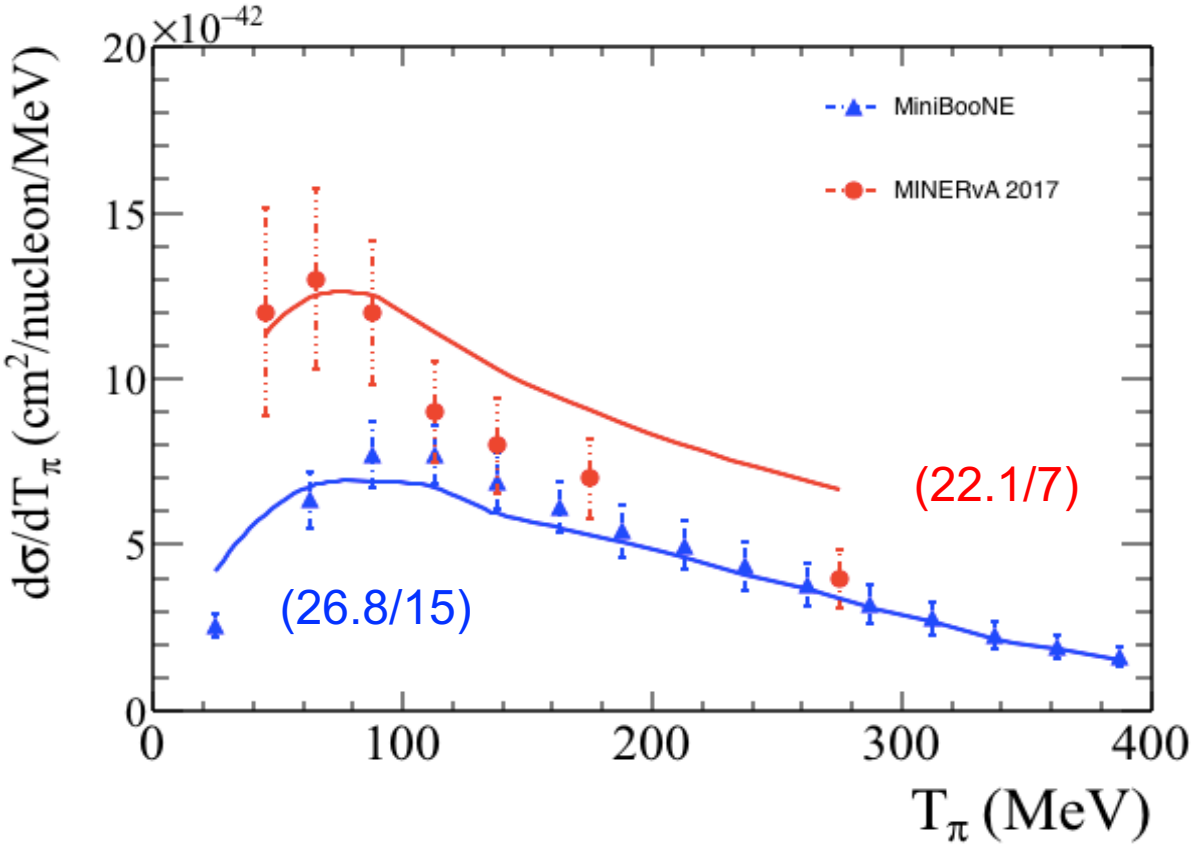
```
$ source setupnuisance.sh
$ nuiscomp -c comparison.xml -o comparison.root
```

```
$ root comparison.root
root [0] MiniBooNE_CC1pip_XSec_1DTpi_nu_data->Scale(1.0 / 14.0);
root [1] MiniBooNE_CC1pip_XSec_1DTpi_nu_MC->Scale(1.0 / 14.0);
root [2] MiniBooNE_CC1pip_XSec_1DTpi_nu_data->Draw("E1");
root [3] MiniBooNE_CC1pip_XSec_1DTpi_nu_MC->Draw("SAME HIST C");
root [4] MINERvA_CC1pip_XSec_1DTpi_nu_2017_data->Draw("SAME E1");
root [5] MINERvA_CC1pip_XSec_1DTpi_nu_2017_MC->Draw("SAME HIST C");
```

Have to scale MB to /nucleon

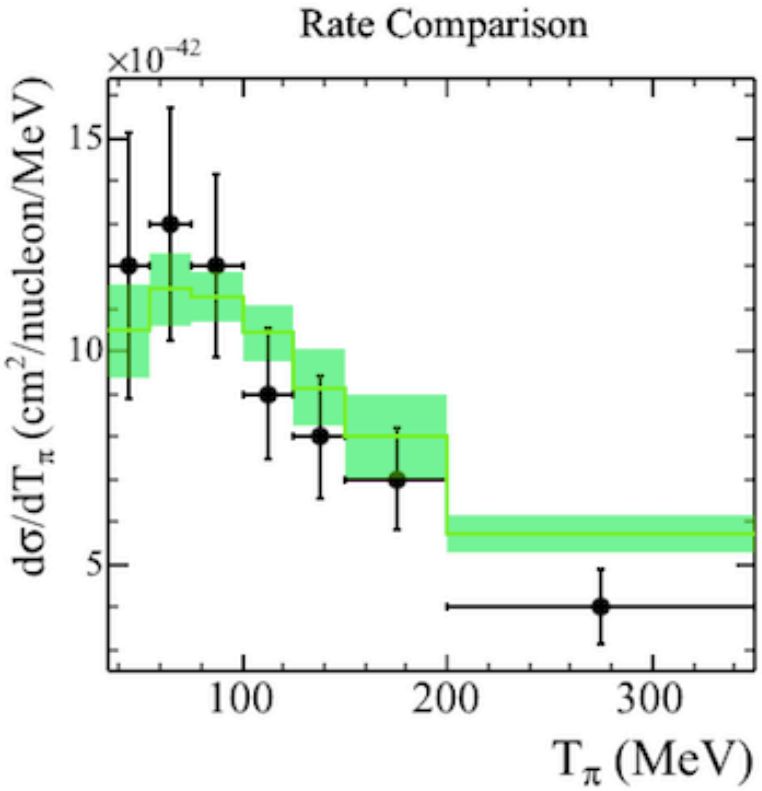
MINERvA/MiniBooNE

- MINERvA updated their pion production dataset. Improved flux constraint and removed some model dependence in the data selection.
- Reduced tensions but still large disagreement at high kinetic energy



Model Uncertainties

Dial	Value
MaCCRES (GeV)	0.94 +- 0.05
NormRES (%)	115 +- 7
NormNonRes (%)	46 +- 4

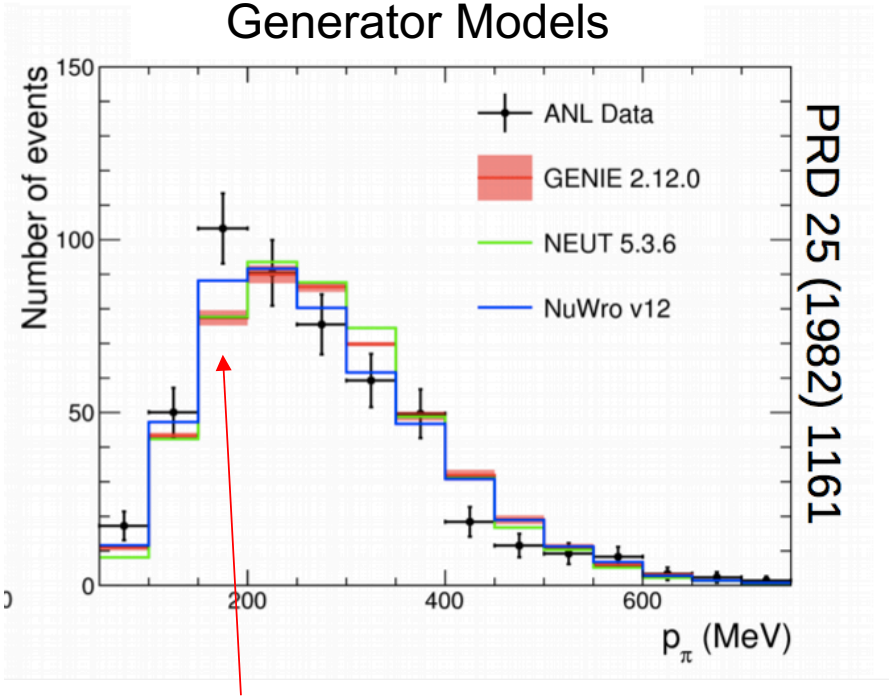


- Not a fair comparison to show just GENIE nominal, tuning to bubble chamber data has found a suppression in Non-Resonent so need to account for that.
- When we update the predictions, and also float FSI uncertainties we still see a disagreement in the total. Average chi2/dof $\sim$ 2.0 due to shape disagreement.

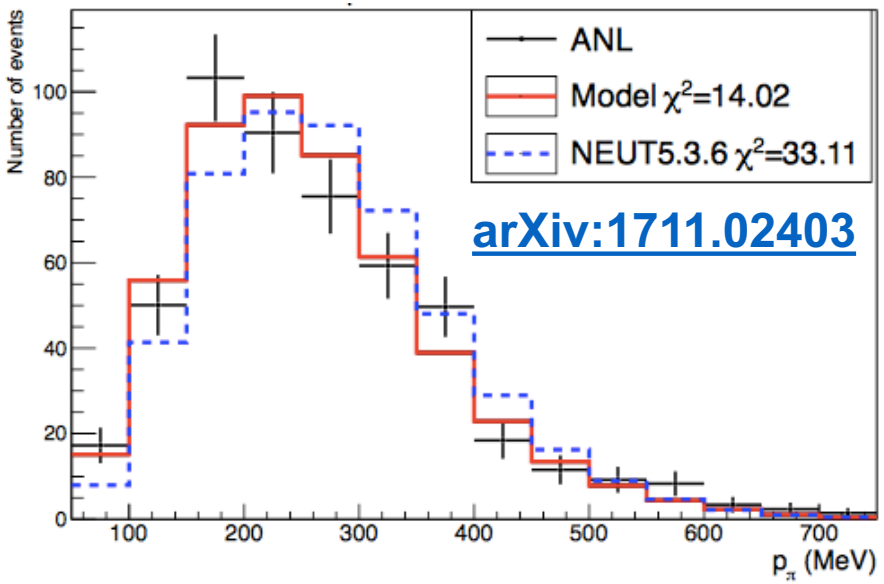
ANL/BNL Pion Kinematics

- We assume FSI could be the problem, but pion production models in generators have almost no freedom to change shape of the kinematics.

Generator Models



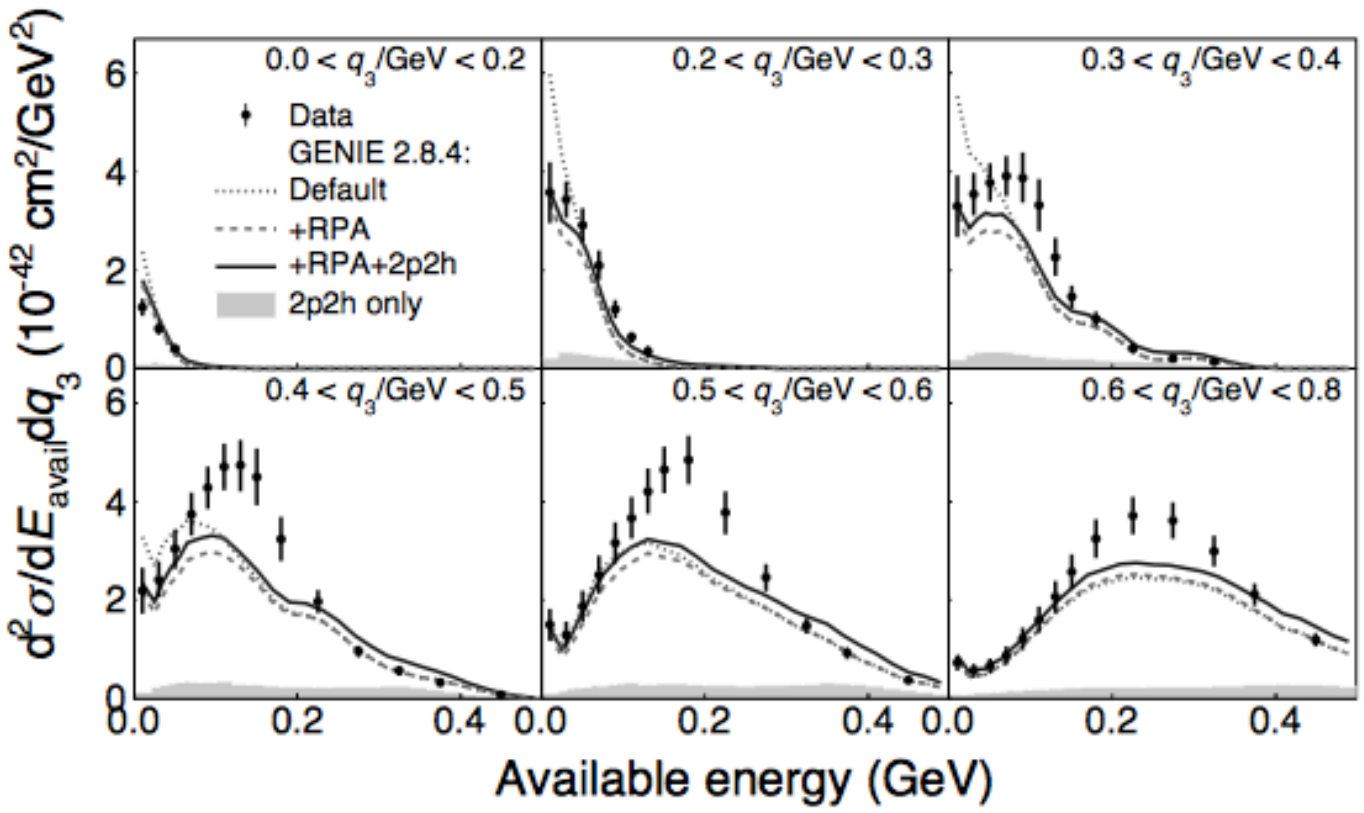
M. Kabirnezhad Model



Shape uncertainty here is from every GENIE dial thrown at once within its errors.

MINERvA CC-inclusive

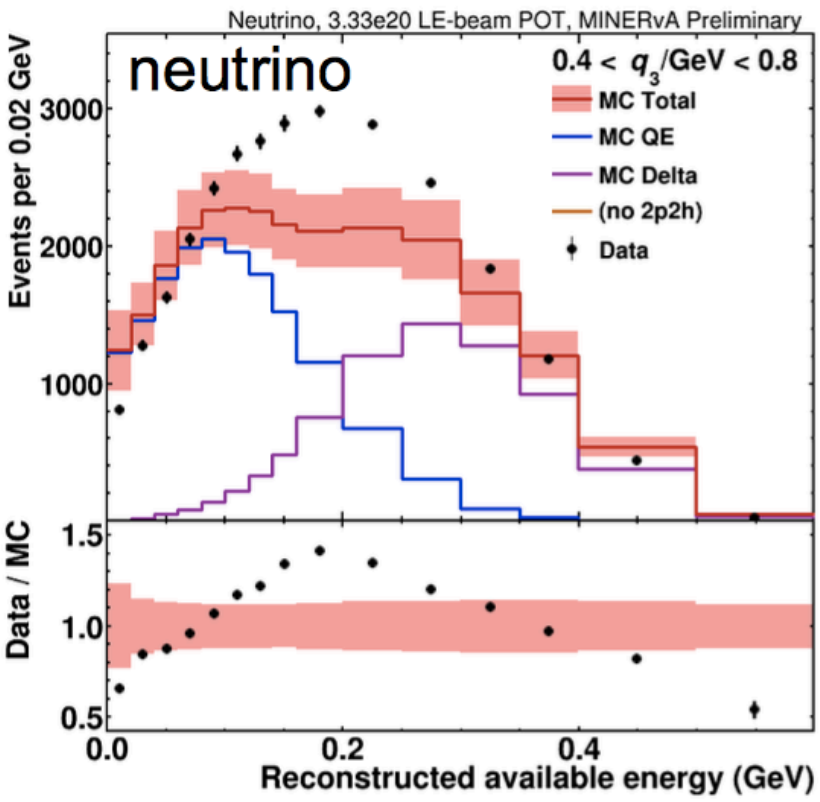
- Latest CC-inclusive data uses observed calorimetric energy to extract cross-section similar to electron scattering q0-q3 distributions.

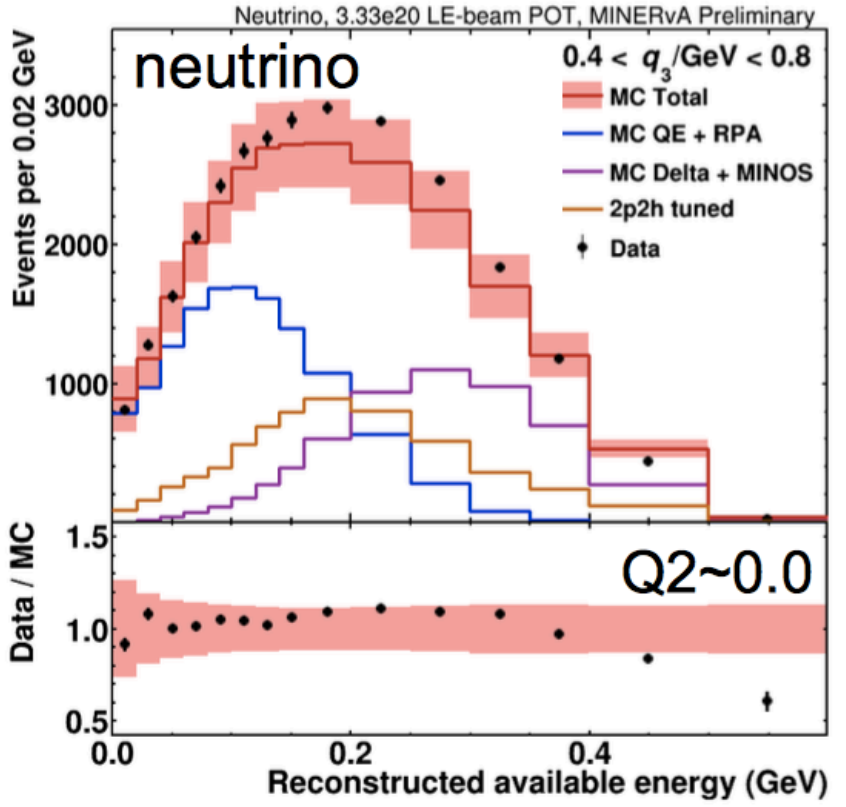


Phys. Rev. Lett. 116, 071802 (2016).

MINERvA CC-inclusive

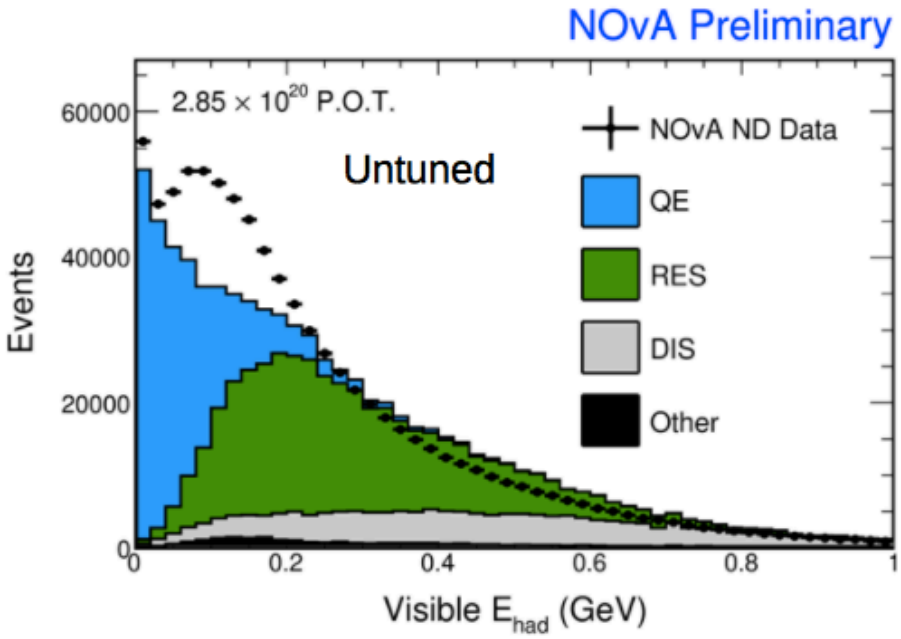
- Disagreement seen in the dip region used as justification for a modification to be applied to the 2p2h model.

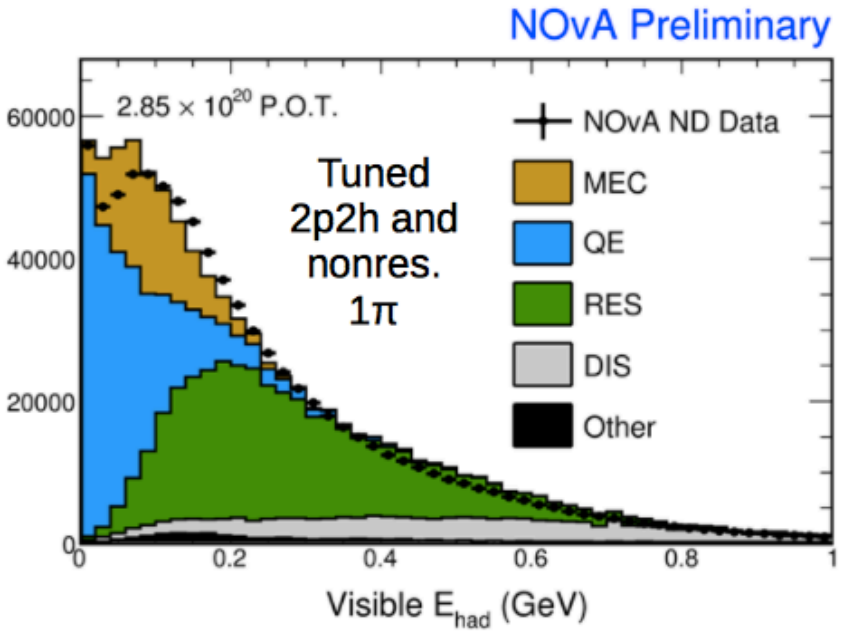




MINERvA CC-inclusive

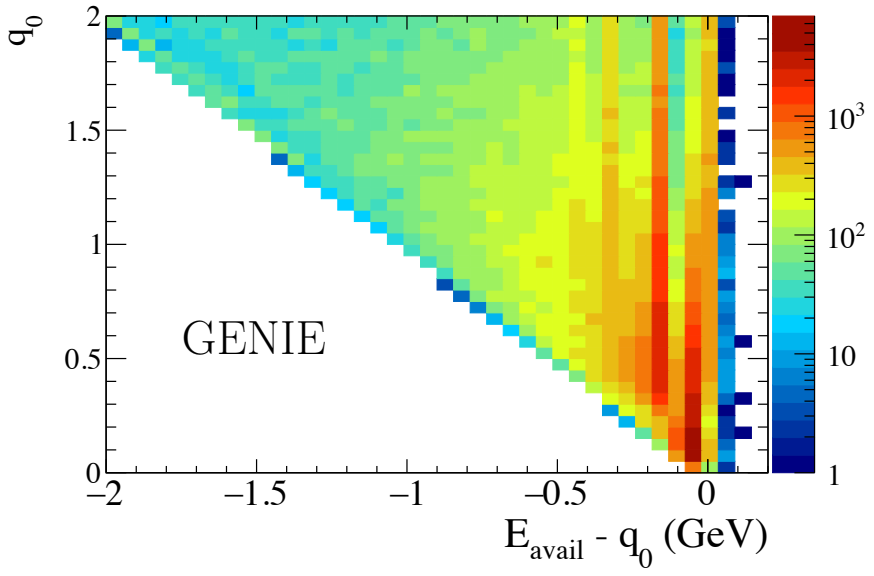
- Disagreement seen in the dip region used as justification for a modification to be applied to the 2p2h model.

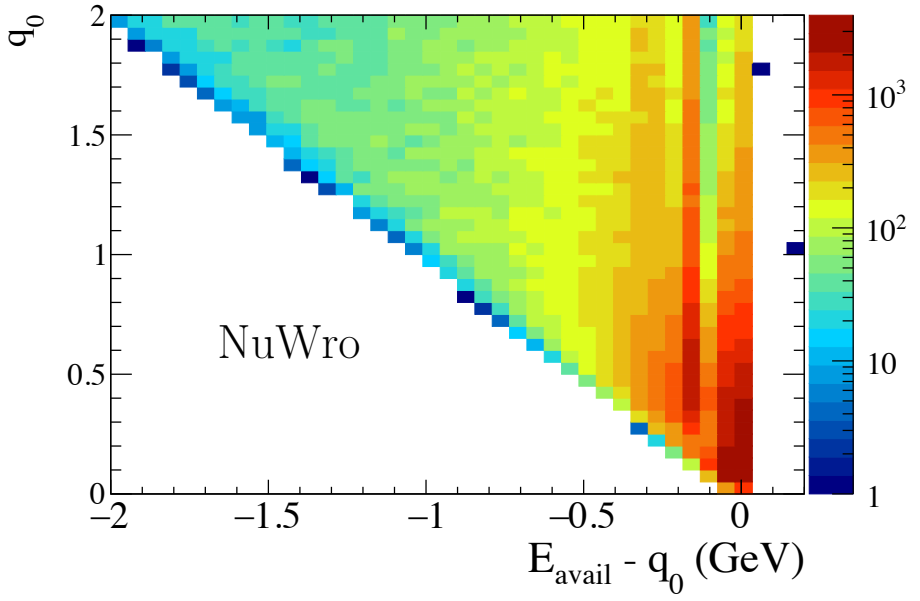




Energy Available

- Measure hadronic energy in the final state (excluding the muon+neutrons) and use this to extract q3 on an event-by-event basis. Binned as a 2D differential cross-section.





- Differences in nuclear and FSI model changes generator E_{avail} predictions.

Comparison Guide

```
$ nuisflat -i TYPE:FILE.root -o OUTPUT.root -f GenericFlux
```

```
$ nuisflat -i GENIE:gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH.500000.ghep.root \
           -o gntp.flat.root -f GenericFlux

$ root gntp.flat.root
root [0] FlatTree_VARS.Draw("q0_true/-1.E3:(Erecoil_minerva-(q0_true*-1.0))/1.E3", \
                            "flagCCINC", \
                            "COLZ") \
```

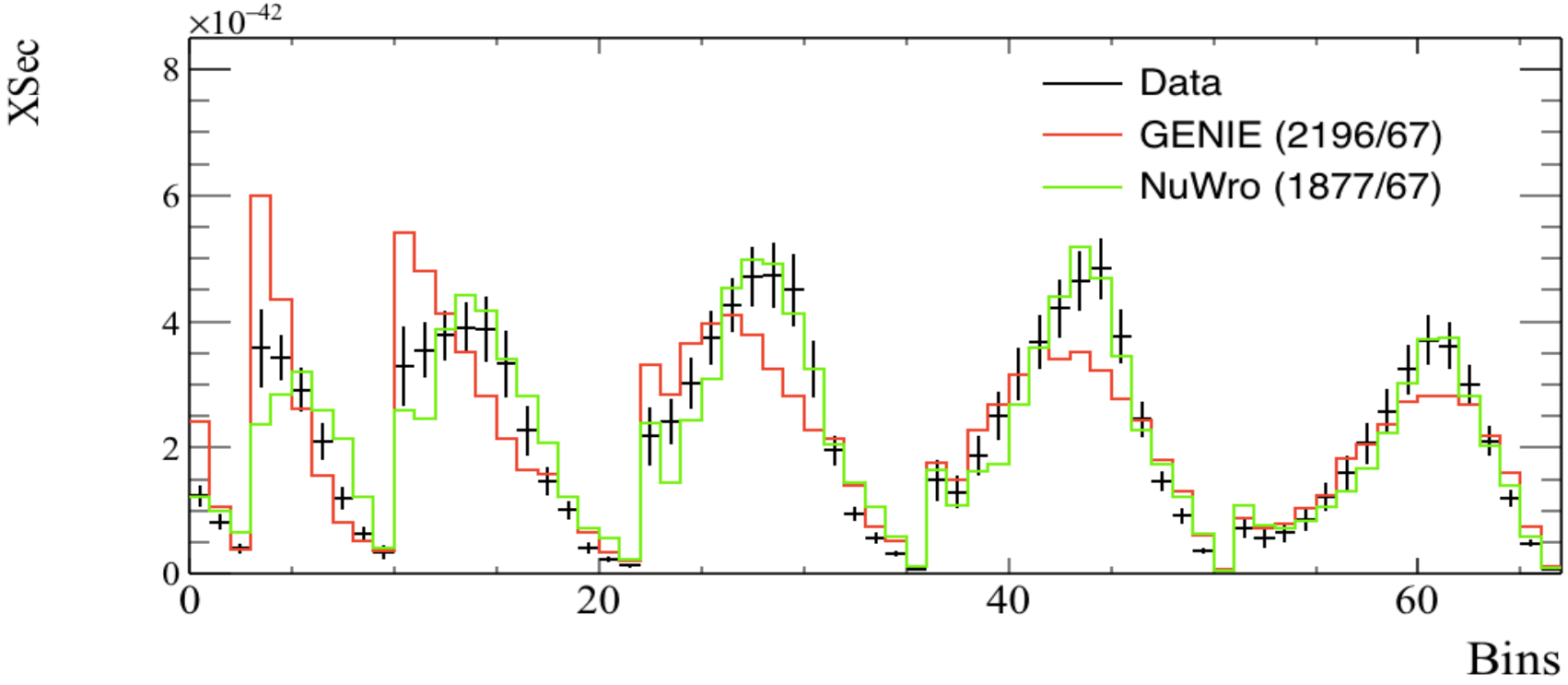
```
$ nuisflat -i NUWRO:nuwrogen.LocalFGNievesQEMEC.MINERvA_fhc_numu.CH.500000.nuwroev.root \
           -o nuwrogen.flat.root -f GenericFlux

$ root nuwrogen.flat.root
root [0] FlatTree_VARS.Draw("q0_true/-1.E3:(Erecoil_minerva-(q0_true*-1.0))/1.E3", \
                            "flagCCINC", \
                            "COLZ") \
```

GENIE vs NuWro

- Difference in the dip region quite significant between GENIE and NuWro
- Using an LFG model in NuWro completely shifts the quasi-elastic peak.

chi2/dof always very large for this data.



Comparison Guide

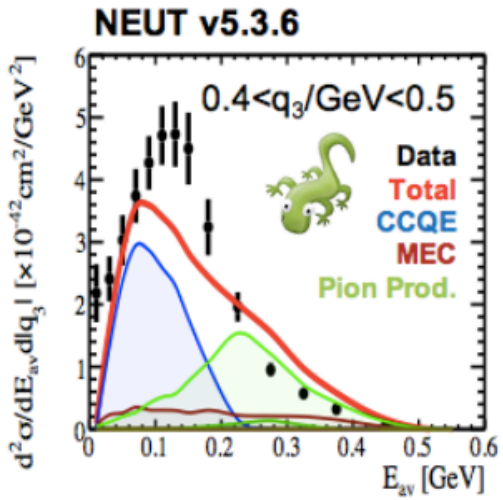
```
$ emacs comparison.xml
```

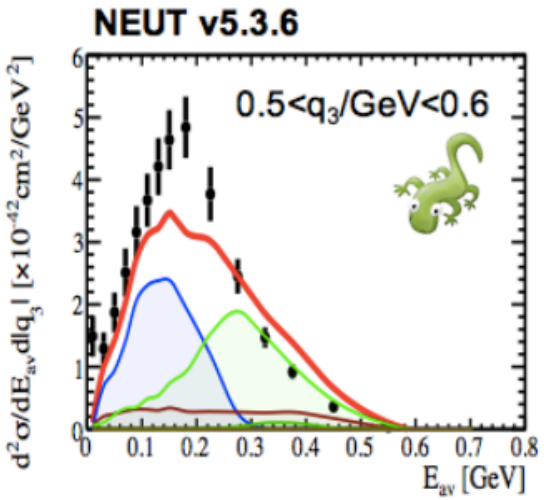
```
<nuisance>
  <sample name="MINERvA_CCinc_XSec_2DEavq3_nu"
          input="GENIE: gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH.500000.ghep.root" />
</nuisance>
```

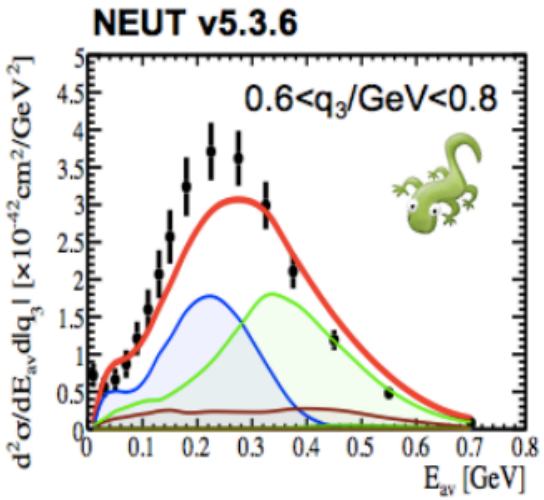
```
$ source setupnuisance.sh
$ nuiscomp -c comparison.xml -o comparison.root
```

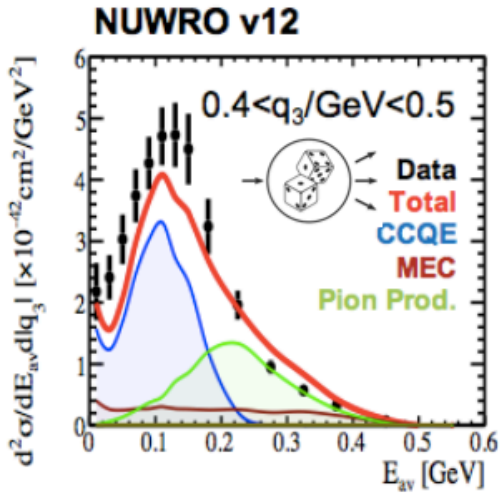
```
$ root comparison.root
root [0] MINERvA_CCinc_XSec_2DEavq3_nu_data_1D->Draw("E1");
root [1] MINERvA_CCinc_XSec_2DEavq3_nu_MC_1D->Draw("SAME HIST");
```

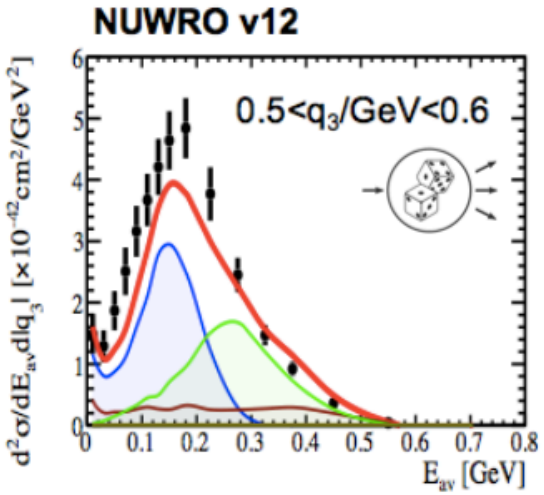
NEUT 5.3.6 vs NuWro v11

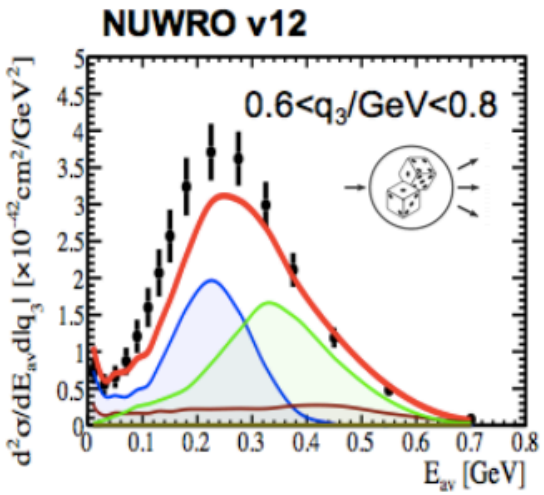












Global Tuning

> *There is a hope, however, that a joint global fit to the existing data could reduce the [cross-section model] uncertainties. When tuning generators in this kind of global fits, a mechanism for examining "tensions" in datasets should be established. A useful goal would be a universal or global tune as achieved by QCD global fits of parton distribution functions.*

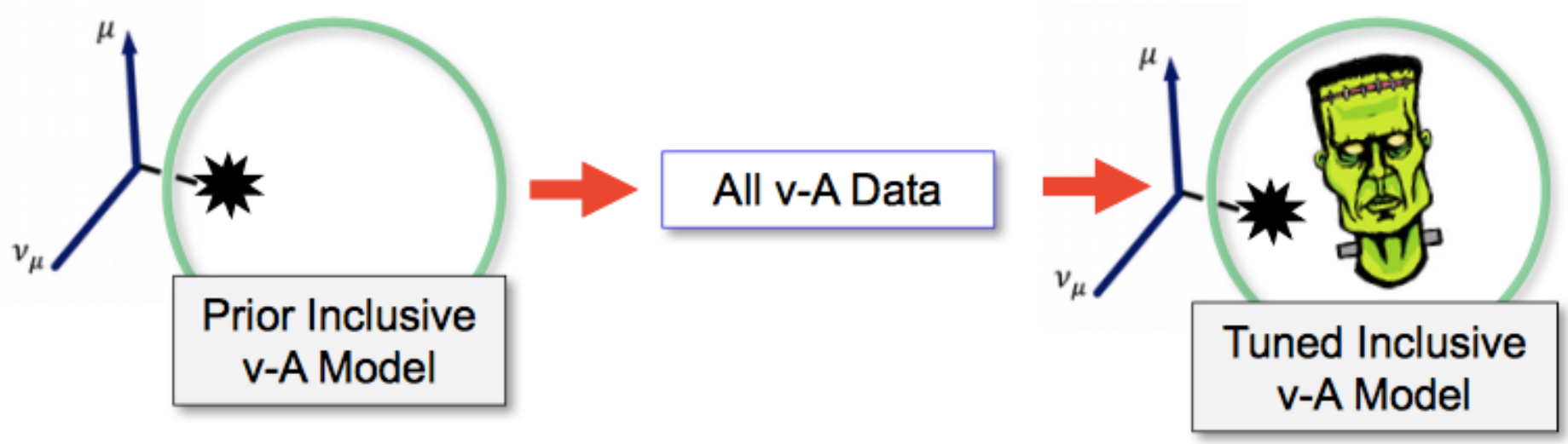
- To achieve the goals of next generation experiments we need to reduce the systematic uncertainties in generators significantly.

- Major problem is that a number of studies have hinted our current relatively large uncertainties are insufficient to get good coverage of the global dataset.

Franken-models

- Neutrino generator models have been referred to as Franken-models.
- We piece together different distinct cross-section components and hope that the combined total is consistent.

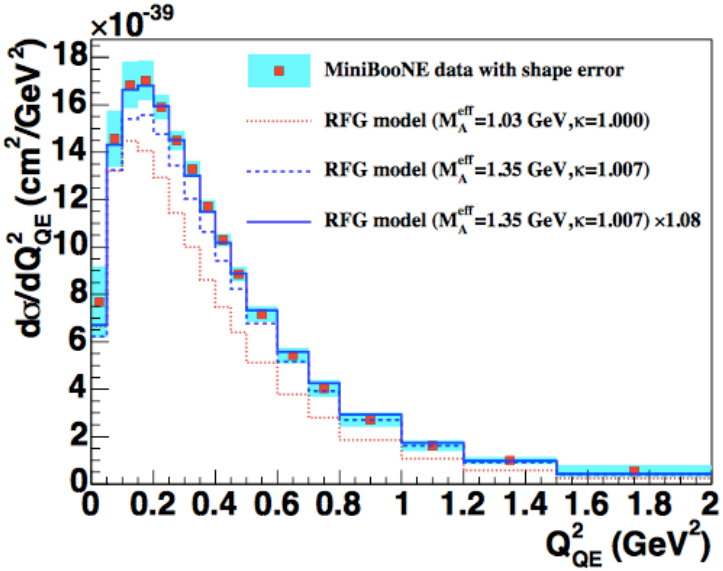
- Comparison to data used as a guide that everything is working correctly in our Franken-model construction.



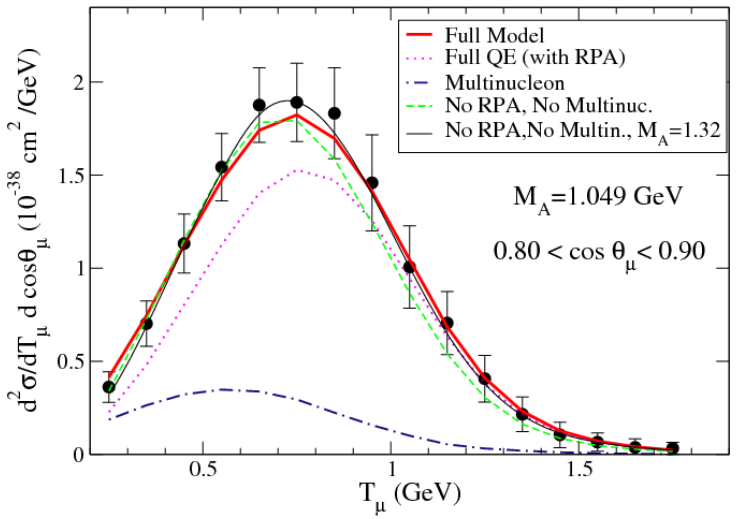
Generator Tuning

- If the models don't agree with the data we [experimentalists] usually take whatever free parameters we can get our hands on and tune them until the data and MC agree.

- Sometimes we tune "bad" parameters. This lets us parametrize disagreements in regions where no theoretical uncertainty is provided.



Phys. Rev. D81, 092005 (2010)

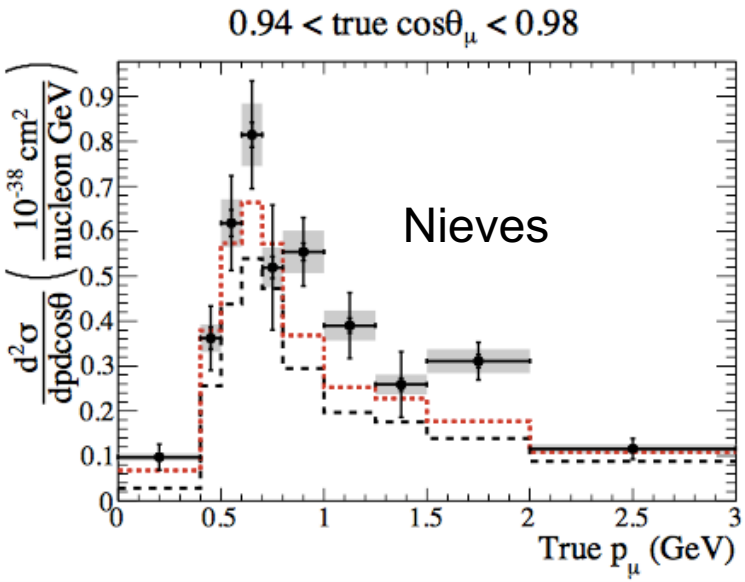


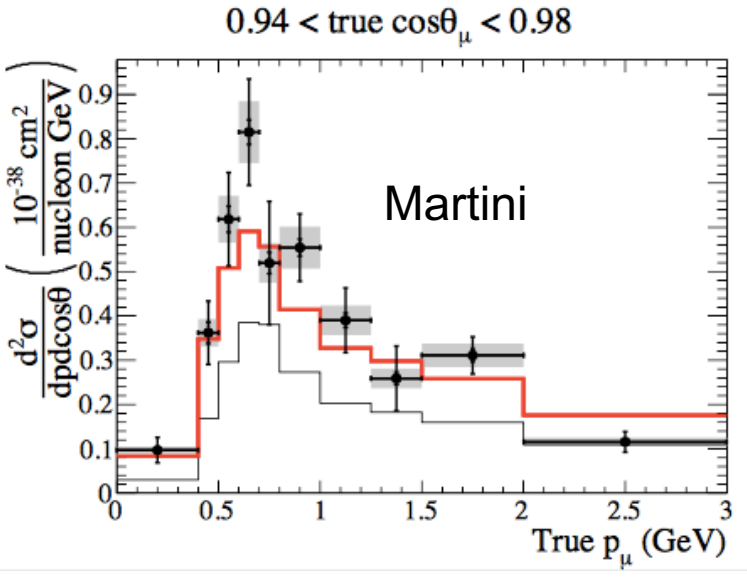
arXiv:1411.7821

NIWG Tunings

- If no theoretical uncertainty is given we still have to tune to data, but at least we have a more theoretically sound model right?

- T2K Neutrino Interactions group did this in joint fits to MINERvA and MiniBooNE CCQE data by floating 2p2h normalization.



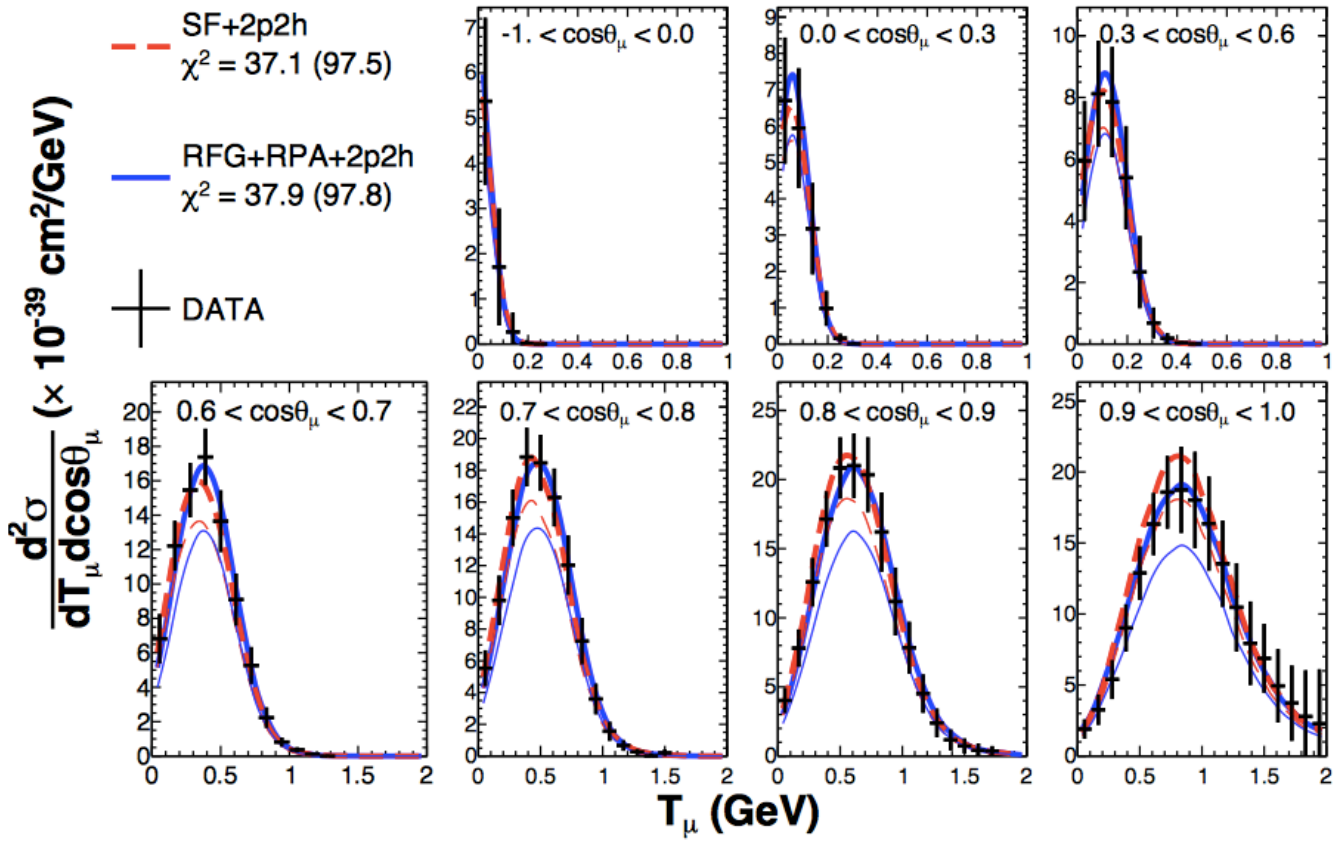


arXiv:1602.03652

T2K CC0pi Data 2p2h Model Comparisons

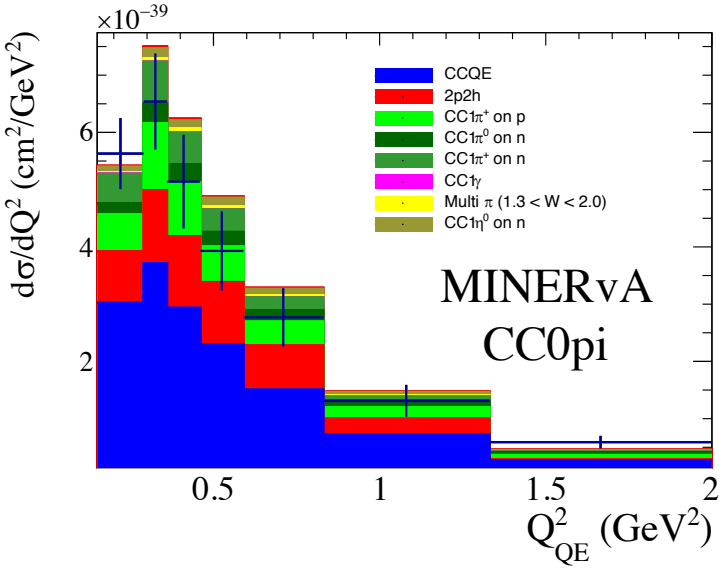
NIWG Tunings

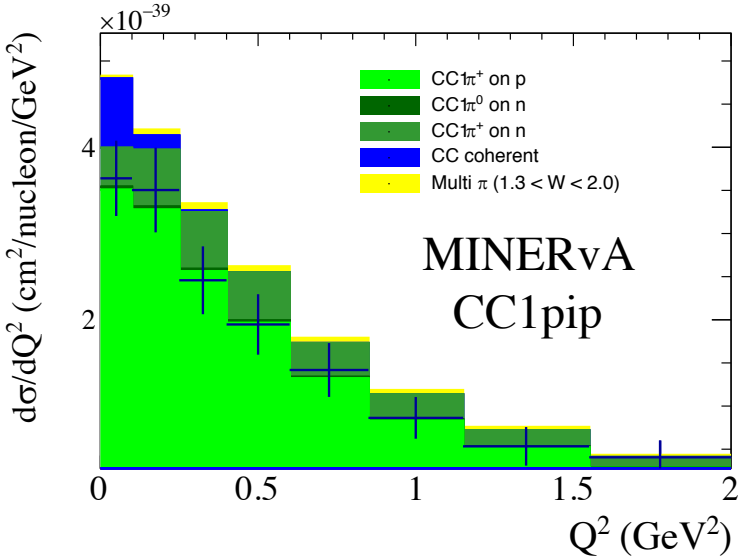
- Best fit was 2p2h = ~27% Nieves and M_A ~1.15 GeV
- Found strong tensions between MINERvA and MiniBooNE data.



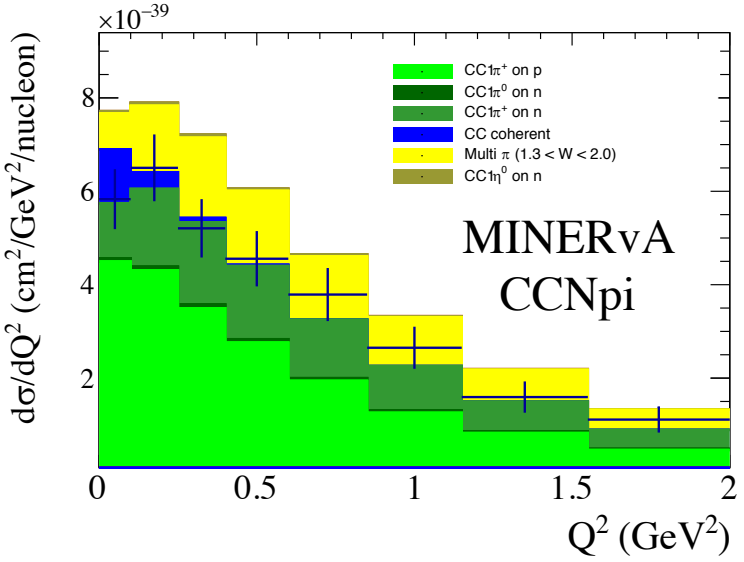
arXiv:1601.05592

The "Kitchen Sink"





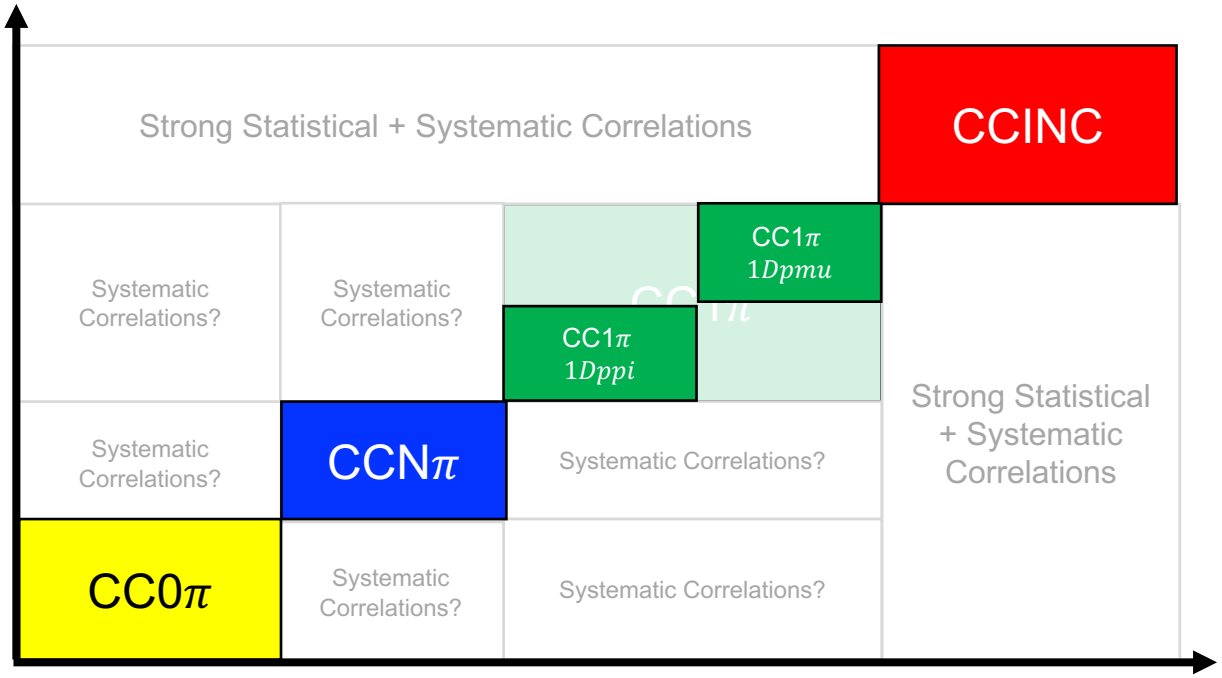
- Need a way to constrain many features of the model at once in a large many-dimensional fit.



Problem of Correlations

- Likely that the neutrino global dataset is highly correlated.

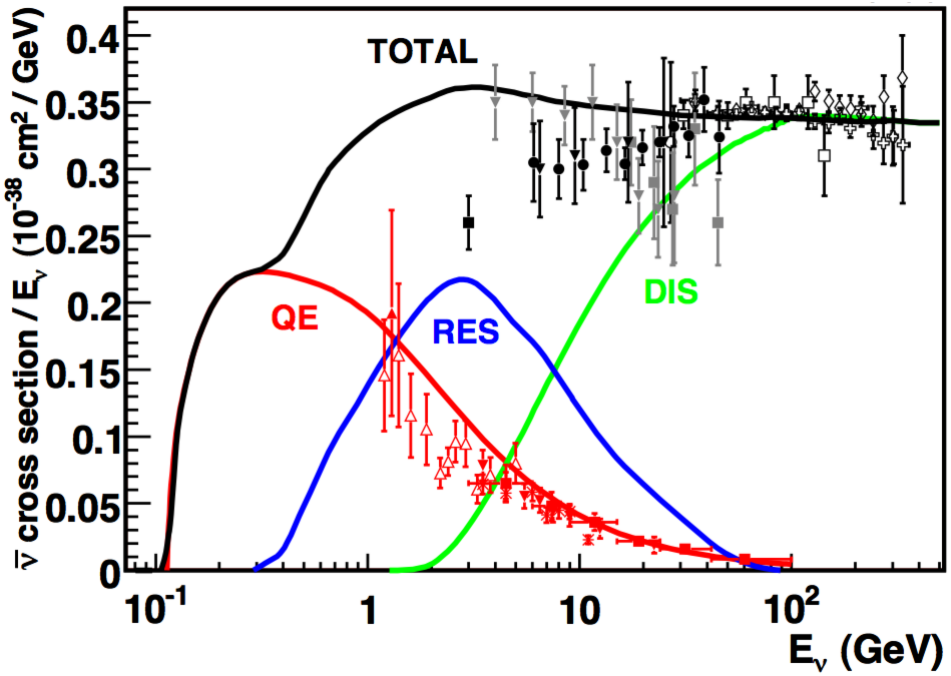
- In most datasets we don't get correlations between different topologies, or even different 1D distributions (e.g. 1Dppi / 1Dpmu). Sometimes none between bins!



Joint Model Fits

- Clearly a number of issues which need to be addressed when trying to produce an Inclusive + Exclusive generator model constraint.

- Tuning experiment-by-experiment at least limits the energy range we have to handle somewhat.



Joint Model Fits

- Insufficient data currently exists for global fits to be entirely data driven.

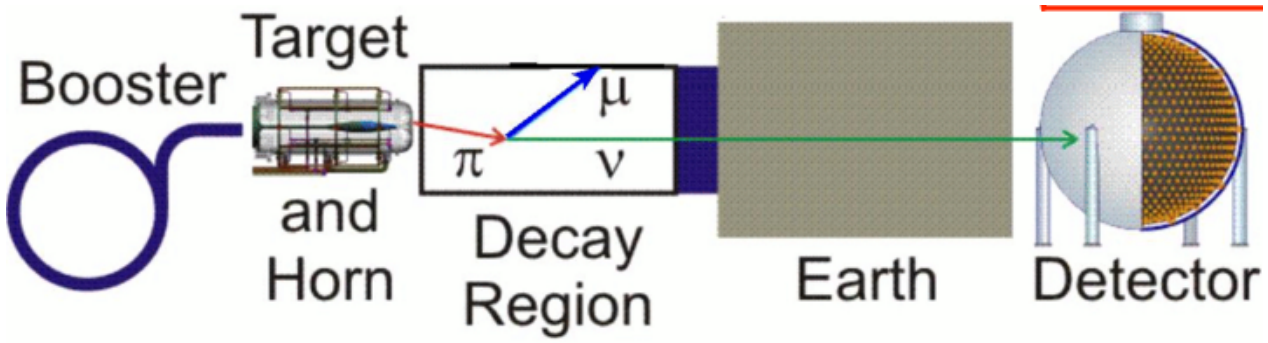
- **NUISANCE Method:**
  1. Tune first to bubble chamber data for free nucleon parameters.
  2. Float nuclear / FSI uncertainties to try and fix tensions at nuclear targets
  3. Tune experiment-by-experiment before worrying about propagating the model between different neutrino energy.

- Still lots of headaches to worry about in this approach, but at least ensures your QE tunings don't significantly mess up your pion data-MC agreement.

- Trying to provide a framework that is consistent across generators and repeatable/extendable so experiments can modify universal tunes to suit their specific needs.

Sterile Neutrino Mixing Aside

- Experiments performing short baseline oscillation fits need a good choice of cross-section model.
- What if other SBL experiments used to measure cross-sections are also sensitive to these oscillations?



- Can't use cross-section model constraints from other experiments unless you've shown that those fits are insensitive to sterile mixing.

Summary

- Very difficult to disentangle all the possible model deficiencies when comparing neutrino cross-section data to generators.
- Extensive comparisons to broad range of data over multiple targets is required.
- Need more work to implement appropriate model freedom in generators to make sure our generator uncertainties are appropriate.
- Steps made towards a global tune, but care is needed since the global neutrino dataset is not that large.

NUISANCE Future

- Framework is open source under the GPLv3 license. We want the community to take it and use it as they see fit.
- The only way we achieve the uncertainty goals of future neutrino experiments is through a joint community effort to ensure what is being used in the MC generators is actually theoretically consistent and agrees with data.

NUISANCE Tutorial

NUISANCE Tutorial

- Please try and download the NUISANCE code from our site.

  https://nuisance.hepforge.org

- Links to lots of possible ways to get our code can be found here:

  https://nuisance.hepforge.org/GettingTheCode.html

- Those working on a laptop the VirtualBox option is the recommended option. Those with access to Fermilab gpvm's can use the CVMFS option when logged into a gpvm machine.

- To do **all** examples you will also need the tutorial events

  https://nuisance.hepforge.org/TutorialEvents.html

- I want to expand the user base as much as possible so please speak to me in a break if you have issues building/running the code ☺

Examples

1. Compare Erecoil distributions between GENIE and NuWro

2. Compare GENIE to MINERvA and MiniBooNE CC1pip Tpi data

3. Generate NuWro events for ANL and compare to ANL_CC1pip_Evt_1Dppi_nu

4. Compare GENIE to MINERvA CC1pip data with MaCCRES reweighted down to -1.0 sigma.

5. Tune MaCCQE to MiniBooNE CCQE data with a prior constraint from bubble chamber.

Example 1

```
$ nuisflat -i TYPE:INPUT_FILE  -o OUTPUT_FILE  -f GenericFlux [-n NEVENTS]
```

Have to tell NUISANCE our input is a GENIE file

```
$ nuisflat -i GENIE:gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH.500000.ghep.root \
           -o gntp.flat.root -f GenericFlux

$ root gntp.flat.root
root [0] FlatTree_VARS.Draw("Erecoil_minerva", "COLZ");
```

Tell it we want the GenericFlux output format.

Have to tell NUISANCE our input is a NUWRO file in the second case.

```
$ nuisflat -i NUWRO:nuwrogen.LocalFGNievesQEMEC.MINERvA_fhc_numu.CH.500000.nuwroev.root \
           -o nuwrogen.flat.root -f GenericFlux

$ root nuwrogen.flat.root
root [0] FlatTree_VARS.Draw("Erecoil_minerva", "COLZ");
```

Format is the same in both files so we can compare them easily.

Example 2 (1)

```
$ nuissamples CC1pip
MiniBooNE_CC1pip_XSec_1DTpi_nu
...
MINERvA_CC1pip_XSec_1DTpi_nu_2017
```

Search NUISANCE dataset names using "$ nuissamples [substring]"

```
$ emacs miniboone_minerva.xml
```

Open a new NUISANCE card file

Add a new "sample" entry for each dataset we care about.

```
<nuisance>
  <sample name="MiniBooNE_CC1pip_XSec_1DTpi_nu"
          input="GENIE:gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root" />
  <sample name="MINERvA_CC1pip_XSec_1DTpi_nu_2017"
          input="GENIE:gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH.500000.ghep.root" />
</nuisance>
```

nuissamples name goes here

Put file input here. Same format as nuisflat in Example 1.

Comparison app called nuiscomp.

```
$ nuiscomp -c CARDFILE.xml -o ROOTOUTPUT.root [ -n NEVENTS ]
```

```
$ nuiscomp -c miniboone_minerva.xml -o miniboone_minerva.root -n 100000
```

Run with our card

Example 2 (2)

```
$ emacs generateplots.py
```

PyROOT macro

```
from ROOT import *
infile = TFile("miniboone_minerva.root","READ")

MiniBooNE_CC1pip_XSec_1DTpi_nu_data.Scale(1.0/14.0)
MiniBooNE_CC1pip_XSec_1DTpi_nu_MC.Scale(1.0/14.0)
MiniBooNE_CC1pip_XSec_1DTpi_nu_data.SetLineColor(kBlue)
MiniBooNE_CC1pip_XSec_1DTpi_nu_MC.SetLineColor(kBlue)
MiniBooNE_CC1pip_XSec_1DTpi_nu_data.SetTitle("MiniBooNE")

MINERvA_CC1pip_XSec_1DTpi_nu_2017_data.SetLineColor(kRed)
MINERvA_CC1pip_XSec_1DTpi_nu_2017_MC.SetLineColor(kRed)
MINERvA_CC1pip_XSec_1DTpi_nu_2017_data.SetTitle("MINERvA 2017")

MiniBooNE_CC1pip_XSec_1DTpi_nu_data.GetYaxis().SetRangeUser(0.0,240E-42)

MiniBooNE_CC1pip_XSec_1DTpi_nu_data.Draw("E1")
MINERvA_CC1pip_XSec_1DTpi_nu_2017_data.Draw("SAME E1")
gPad.BuildLegend(0.6,0.7,0.83,0.88)
MiniBooNE_CC1pip_XSec_1DTpi_nu_MC.Draw("SAME HIST C")
MINERvA_CC1pip_XSec_1DTpi_nu_2017_MC.Draw("SAME HIST C")

gPad.Update()
raw_input("Complete")
```

Have to scale because one is in units of cm2/nucleon and another in cm2/CH2

Script is just providing simple ROOT formatting.

Example 3 (1)

Need to generate some inclusive NuWro events for this example

Make sure NUISANCE is setup! (Build type dependent)

```
$ source $HOME/NUISANCEMC/nuisance/v2r8/build/Linux/setup.sh
$ source $NUWRO/build/Linux/setup.sh
```

```
$ emacs params.txt
```

Open a new params file

```
@beam/ANL.txt
nucleus_p=1
nucleus_n=1
nucleus_target=0
kaskada_on=0
pauli_blocking=0
```

Save these lines into it

Tells NuWro to generate free nucleons target

```
$ nuwro -i params.txt -o ANL-nuwro-events.root
```

Run nuwro with new params file

```
$ PrepareNuwro -f ANL-nuwro-events.root
```

Prepare new events for NUISANCE

Example 3 (2)

Can now compare prepared events in NUISANCE easily.

```
$ emacs ANL.xml
```

Lets write a new NUISANCE card file

Have to tell NUISANCE its a NUWRO file

```
<nuisance>
  <sample name="ANL_CC1ppip_Evt_1Dppi_nu" input="NUWRO:ANL-nuwro-events.root" />
</nuisance>
```

Dataset name

Pass our new prepared event file in

Can search names for other datasets using nuissamples

```
$ nuissamples [substring]
```

```
$ nuissamples CC1ppip
ANL_CC1ppip_XSec_1DEnu_nu
ANL_CC1ppip_XSec_1DEnu_nu_W14Cut
ANL_CC1ppip_XSec_1DEnu_nu_Uncorr
```

Example 3 (3)

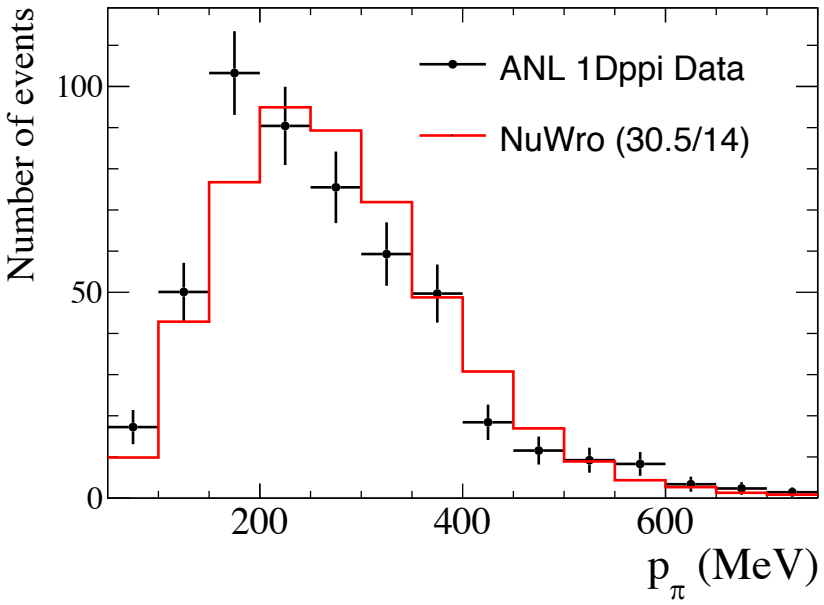
Can now compare prepared events in NUISANCE easily.

```
$ nuiscomp -c ANL.xml -o ANL.root
```

Run the standard comparison app

```
$ root ANL.root
root [0] ANL_CC1ppip_Evt_1Dppi_nu_data->Draw("E1");
root [1] ANL_CC1ppip_Evt_1Dppi_nu_MC->Draw("SAME HIST C")
```

Compare histograms in TBrowser



Example 4

```
$ emacs minerva_nominal.xml
<nuisance>
   <sample name="MINERvA_CC1pip_XSec_1DTpi_nu"
           input="GENIE:gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH2.500000.ghep.root" />
</nuisance>
```

Write sample entry as before.

```
$ emacs minerva_mareweight.xml
<nuisance>
   <parameter name="MaCCRES" type="genie_parameter" nominal="-1.0" state="FIX" />
   <sample name="MINERvA_CC1pip_XSec_1DTpi_nu"
           input="GENIE:gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH2.500000.ghep.root" />
</nuisance>
```

Add a new "parameter" xml entry to tell NUISANCE to apply GENIE RW.

```
$ nuiscomp -c minerva_nominal.xml -o minerva_nominal.root -n 100000
$ nuiscomp -c minerva_mareweight.xml -o minerva_mareweight.root -n 100000
```

MC histograms in this file will have been reweighted.

Example 5 (1)

```
$ emacs miniboone_nominal.xml
<nuisance>
   <sample name="MiniBooNE_CCQE_XSec_2DTcos_nu" type="SHAPE/DIAG"
           input="GENIE:gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root" />
</nuisance>
```

Make a nominal sample card.

Have to tell NUISANCE to do a SHAPE-only comparison

```
$ emacs miniboone_ma.xml
<nuisance>
   <parameter name="MaCCQE" type="genie_parameter"
              nominal="0.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
   <sample name="MiniBooNE_CCQE_XSec_2DTcos_nu" type="SHAPE/DIAG"
           input="GENIE:gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root" />
</nuisance>
```

This time also provide low-high limits and step size for Minuit

Also tell NUISANCE to treat this parameters state as FREE

Example 5 (2)

```
$ emacs miniboone_ma.xml
<nuisance>
   <parameter name="MaCCQE" type="genie_parameter"
              nominal="0.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
   <sample name="MiniBooNE_CCQE_XSec_2DTcos_nu" type="SHAPE/DIAG"
           input="GENIE:gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root" />
</nuisance>
```

```
$ nuiscomp -c miniboone_nominal.xml -o miniboone_nominal.root  -n 100000
$ nuismin -c miniboone_ma.xml -o miniboone_ma.root  -n 100000
```

Running nuiscomp on our first card will make the nominal comparison.

Running nuismin on our minimiser card will make NUISANCE vary all "FREE" parameters until a the joint chi2 is minimised.

Joint chi2 formed by adding chi2 from each "sample" loaded. For a joint fit include multiple sample entries.

Outputs of nuismin will be the same as nuiscomp, but will have MA scaled down.

Example 5 (3)

```
$ emacs miniboone_ma.xml
<nuisance>
   <parameter name="MaCCQE" type="genie_parameter"
              nominal="0.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
   <covar name="MaPull" input="DIAL:MaCCQE;0.0;1.0" type="GAUSPULL" />
   <sample name="MiniBooNE_CCQE_XSec_2DTcos_nu" type="SHAPE/DIAG"
           input="GENIE:gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root" />
</nuisance>
```

```
$ nuiscomp -c miniboone_mawithbcprior.xml -o miniboone_mawithbcprior.root  -n 100000
```

Include a "covar" term to tell NUISANCE to place a pull on the free parameter.

Should create a new fit result where a Gaussian prior of MaCCQE = 0.0 +- 1.0 sigma is applied to the fit results.

See backup slides for more guidance on how to use the priors.

NUISANCE Notes

Getting the code

- Number of different options to download and use NUISANCE
- Visit following page for more details (linked on school website)

https://nuisance.hepforge.org/GettingTheCode.html

- You will also need the tutorial events, also linked on our website.

https://nuisance.hepforge.org/TutorialEvents.html

- Note: The Virtual Box disk image comes pre-packaged with the required Monte-Carlo events.

nuisflat

- Simplest NUISANCE application "nuisflat" takes an event input and converts it into a simpler TTree format.

- Very similar to GENIE's gntpc convertor.

- Given the path to a NUISANCE-ready MC file of a given "INTYPE" it can be ran using the following:

```
$ nuisflat -i INTYPE:/path/to/inputfile.root \
           -o output.root \
           -f GenericFlux \
         [ -n NEVENTS ]
```

NUISANCE Ready?

- NUISANCE needs flux and cross-section histograms to normalize events to the correct rate

$$R(E_\nu) = \Phi(E_\nu) \times \sigma(E_\nu) \times T$$

N-Targets

Predicted rate given the flux

Flux

Total Xsec spline

- Standard gevgen doesn't save this in the exact format we need.
- Have custom NUISANCE applications that can generate/prepare events with this information, but time consuming so won't cover this today.

```
$ nuwro -i nuwro_params.txt -o nuwro_output.root
$ PrepareNuwro nuwro_output.root
$ nuisflat -i NUWRO:nuwro_output.root -f GenericFlux -o flat.root
```

NUISANCE Prepare

- Some MC files have already been prepared for you. Download from link below.

  https://nuisance.hepforge.org/TutorialEvents.html

- On the command line

```
$ wget https://www.dropbox.com/s/7qjlug1sgg3og7d/tutorial_events.tar.gz?dl=0
$ tar -zxvf tutorial_events.tar.gz\?dl\=0
$ cd tutorial_events
```

- Tarball contains example scripts on how each event file was generated.

- We also keep MC on our website, if you are ever in need of some MC events to run NUISANCE with!

Models

- There are a number of models for you to choose from in those MC folders. I'll refer to each model by its tag in the exercises.

```
$ ls tutorial_events/
gntp.DefaultPlusValenciaMEC.MINERvA_fhc_numu.CH.500000.ghep.root
gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root
nuwrogen.LocalFGNievesQEMEC.MINERvA_fhc_numu.CH.500000.nuwroev.root
```

- Format: **GENERATOR.TAG.FLUX.TARGET.NEVENTS.root**

Tag (model)	Notes
gntp.Default	GENIE 2.12.6 Default
gntp.DefaultPlusValenciaMEC	GENIE Default+Valencia 2p2h
nuwrogen.LocalFGNievesQEMEC	NuWro Default+LFG+Valencia RPA/2p2h

Input Types

- We have our event sample, now we just have to tell NUISANCE what type it is when loading them in.

- Input Format:

```
"FILETYPE:/path/to/eventfile.root"
```

- Format is the same for all applications as this string is passed to the InputHandler creator.
- Uses `FILETYPE` to figure out what InputHandler to create.

- Some possible `FILETYPE`s:

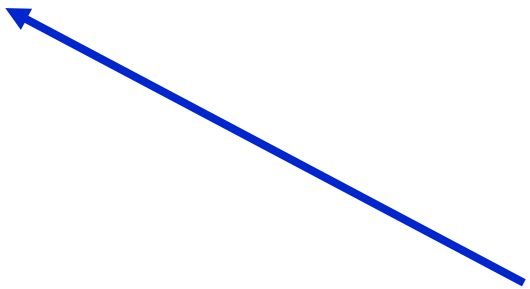
```
GENIE:/path/geniefile.root
NUWRO:/path/nuwrofile.root
NEUT:/path/neutfile.root
GiBUU:/path/gibuufile.root
FEVENT:/path/fitevent.root
```

Problem 1

1. Plot the event spectrum as a function of Neutrino energy for GENIE Default events.

- Can generate nuisflat output by running the following command

```
$ nuisflat \
  -i GENIE:genie/gntp.Default.MiniBooNE_fhc_numu.CH2.500000.ghep.root \
  -o genie.Default.flat.root \
  -f GenericFlux \
  -n 50000
```



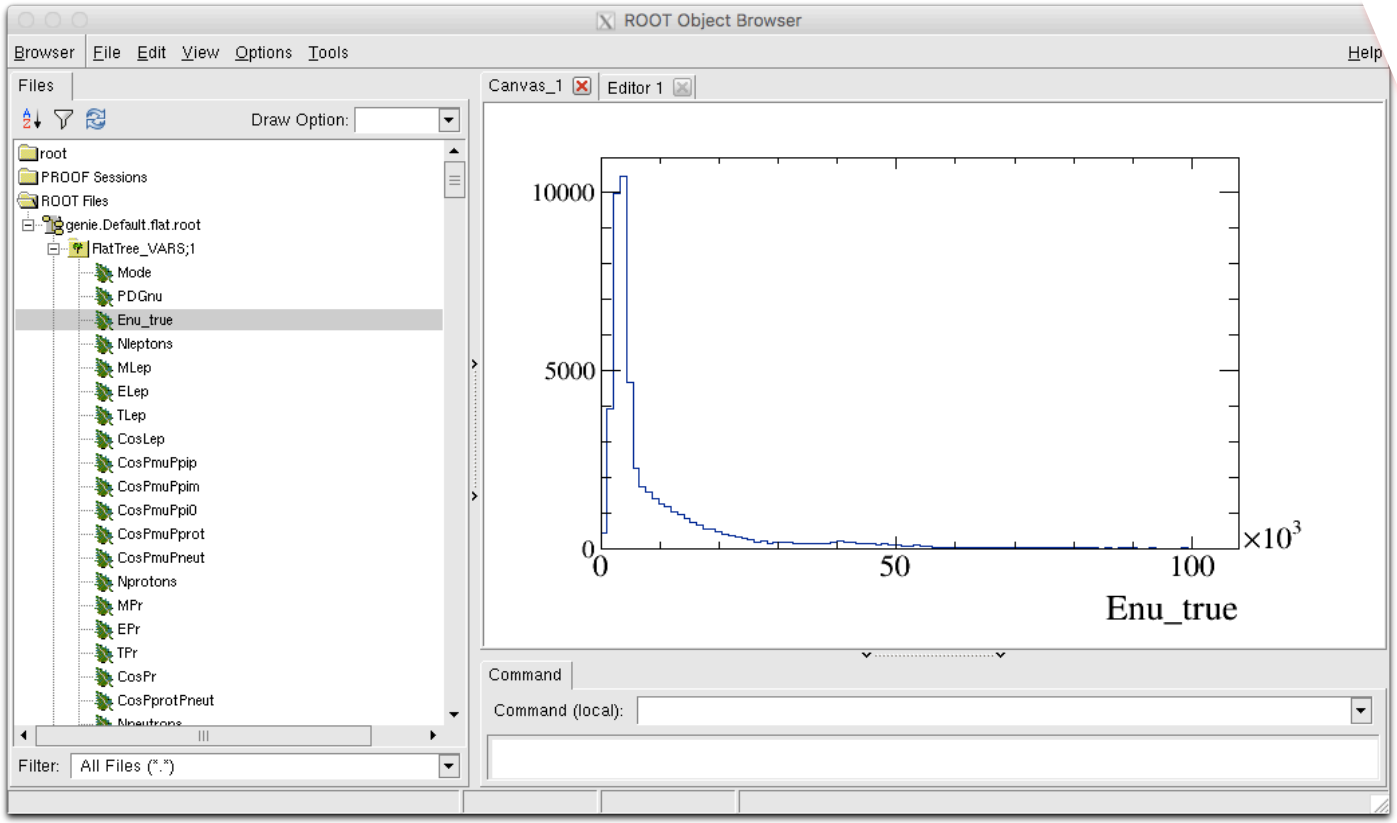
All applications let you restrict the total MC events processed using the (-n) argument.

For examples today we will be restricting the number of events to save time.

Problem 1

```
[jstowell@minervagpvm03 MC]$ root genie.Default.flat.root
root [0] Attaching file genie.Default.flat.root as _file0…
root [1] TBrowser b
root [2] FlatTree_VARS->Draw("ELep", "fScaleFactor");
```

Hint for problem 2.



Data Comparison Classes

- Above the convertors are the measurement classes.
- Set of analysis classes that loop over a collection of FitEvent's and generate distributions.

- **These are called "samples" inside NUISANCE.**

```
$ ls $NUISANCE/src/MINERvA/

MINERvA_CCQE_XSec_1DQ2_nu.cxx
MINERvA_CCQE_XSec_1DQ2_nu
MINERvA_CCinc_XSec_1DEnu_nu.cxx
MINERvA_CCinc_XSec_1DEnu_nu.h
```

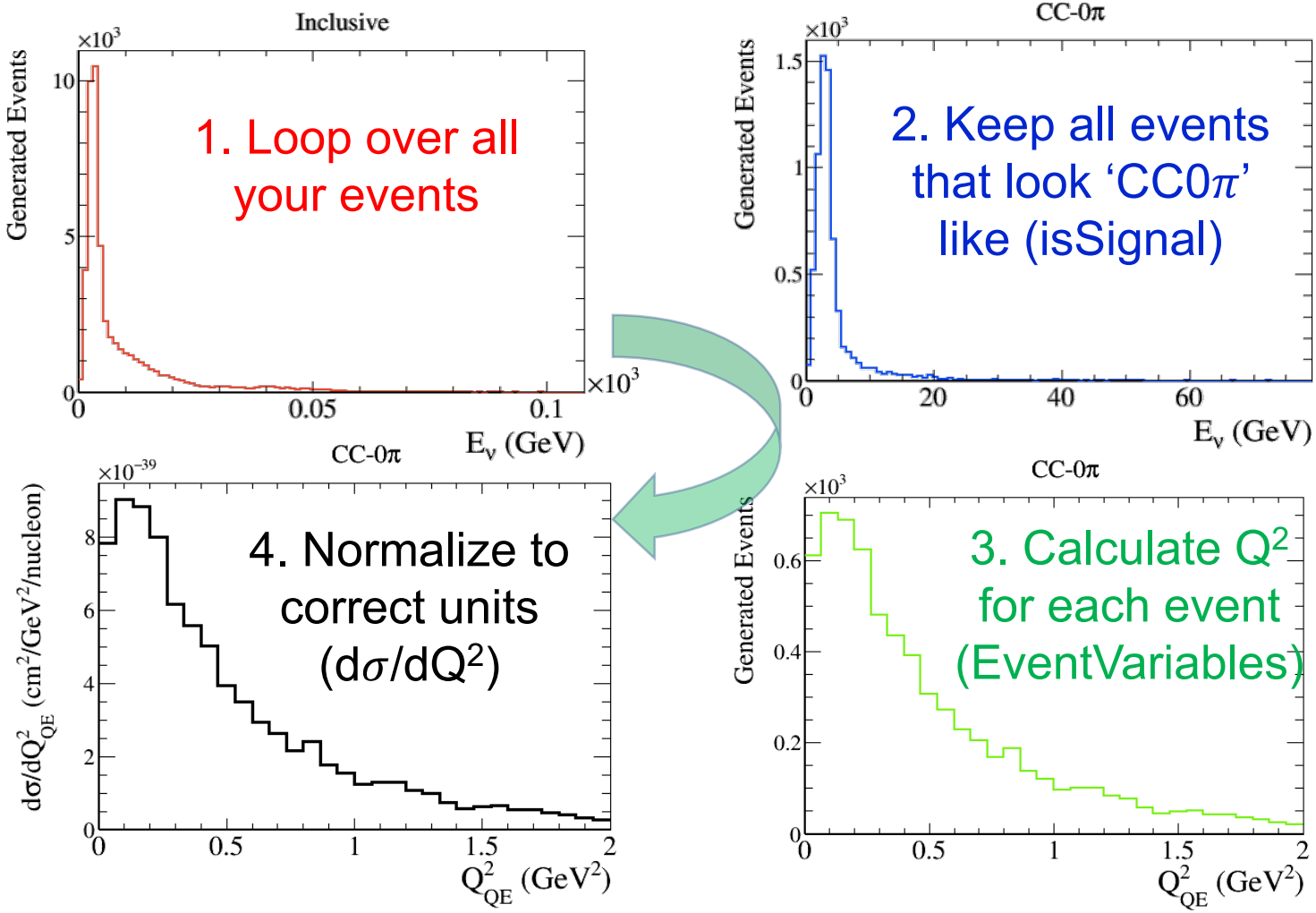
Usually one class implementation for each sample

- Each sample inherits from a base "Measurement" class containing useful functions.
- Base classes help automate the processing chain.

Analysis Process

- Samples function in a similar way to a real cross-section (without detector/systematics…)

E.g. CC0π 1DQ^2_{QE} Distribution



Comparisons

- Event processing requirements (isSignal/EventVariables) on the previous slide are implemented in the sample class.

- Samples can then be called easily by string, and passed an input file in a similar way to how we passed files to nuisflat.

```
<nuisance>

    <!-- Samples -->
    <sample name="MINERvA_CCQE_XSec_1DQ2_nu" input="GENIE:@GENIE_DIR/gntp.CH.root" />
    <sample name="MINERvA_CC1pip_XSec_1DTpi_nu" input="GENIE:@GENIE_DIR/gntp.CH.root" />
    <sample name="MINERvA_CC1pip_XSec_1Dth_nu" input="GENIE:@GENIE_DIR/gntp.CH.root" />

</nuisance>
```

- Adding your own sample that can be called here will be covered in the next workshop!

Running comparison

- Comparison application : **nuiscomp**

- nuiscomp requires you to write a card file to tell you what comparison you want it to create.

```
nuiscomp -c cardfile.xml -o output.root
```

cardfile.xml

```
<nuisance>
    <config GENIE_DIR="/path/to/my/genie/events/" />
    <parameter type="genie_parameter" name="MaCCQE" nominal="1.0" state="FIX" />
    <sample name="MINERvA_CCQE_XSec_1DQ2_nu" input="GENIE:@GENIE_DIR/gntp.CH.root" />
</nuisance>
```

- Cardfile is just a xml file listing the parameters and samples we want and the input files for each one.

Writing a card file

```
<nuisance>
  <!-- List of Samples -->

</nuisance>
```

- Open a new file in a text editor ***"samplecard.xml"***
- XML Card files wrapped in a nuisance statement so first we need to add those.
- Comments given as standard XML comments.

Writing a card file

```
<nuisance>
  <!-- List of Samples -->
  <sample name=""
          input="" />
</nuisance>
```

- To tell NUISANCE to load a new sample we need to include a "sample" XML structure.

```
<sample name="NAMEDEF"  input="INPUTDEF" />
```

Required Keys:

- "NAMEDEF" = Name of the sample to load
- "INPUTDEF" = Input MC File Information

Same format as nuisflat inputs!

Finding a sample

- Want to compare to MINERvA CCQE data.
- "nuissamples" script provided to search for sample names that can be used.

```
$ nuissamples [substring]
```

1st argument is a search substring. If none given, the full sample list is returned.

```
$ nuissamples MINERvA_CCQE
MINERvA_CCQE_XSec_1DQ2_nu
MINERvA_CCQE_XSec_1DQ2_nu_20deg
MINERvA_CCQE_XSec_1DQ2_nu_oldflux
MINERvA_CCQE_XSec_1DQ2_nu_20deg_oldflux
MINERvA_CCQE_XSec_1DQ2_antinu
...
```

This is a sample we could use!

Format: EXPERIMENT_CHANNEL_TYPE_DISTRIBUTION_EXTRAIDs

Writing a card file

```
<nuisance>
  <!-- List of Samples -->
  <sample name="MINERvA_CCQE_XSec_1DQ2_nu"
          input="" />
</nuisance>
```

- Now we just need to include our input file. Can use the same input files we used for nuisflat. e.g.

```
input="FILETYPE:/path/to/file.root"
```

- Want to compare to GENIE Default:

```
input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root"
```

Writing a card file

```
<nuisance>
  <!-- List of Samples -->
  <sample name="MINERvA_CCQE_XSec_1DQ2_nu"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
</nuisance>
```

- Combining everything we should have a cardfile ready.
- We can now run it using

```
$ nuiscomp -c samplecard.xml -o sampleccqe.root -n 100000
```

Again we are using 1E5 events to save time, but proper comparisons should use full event sample.

NUISCOMP Output

```
[LOG Minmzr]:- Getting likelihoods...                          : -2logL
[LOG Minmzr]:- -> MINERvA_CCQE_XSec_1DQ2_nu                    : 17.289/8
[LOG Fitter]: Likelihood for JointFCN:                          17.289
[LOG Fitter]: ------------
[LOG Fitter]: Saving current full FCN predictions
[LOG Minmzr]:- Writing each of the data classes...
[LOG Sample]:-- Written Histograms: MINERvA_CCQE_XSec_1DQ2_nu
[LOG Fitter]: ------------------------------------ -
[LOG Fitter]: Comparison Complete.
[LOG Fitter]: ------------------------------------ -
```

- NUISANCE automatically calculates you a χ^2/NDOF value for the data/MC comparison.
- Tries to use full covariance information where possible.
- Saves all the histograms produced into our output file.

NUISCOMP Output (2)

- Number of different objects saved into the output file.
- Each one has been prepended with the sample name.

- **Examples**
  - samplename**_data** : Data distribution
  - samplename**_MC** : MC distribution in data binning
  - samplename**_MC_FINE** : MC in fine binning
  - samplename**_MC_SHAPE** : MC normalised to data
  - samplename**_data_ratio** : data/MC ratio
  - samplename**_MODES** : THStack true interaction channels

Hint: Use gPad->BuildLegend() to see all the included true interaction channel labels.

Multiple Samples

```
<nuisance>
  <!-- List of Samples -->
  <sample name="MINERvA_CC1pip_XSec_1DTpi_nu"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
  <sample name="MINERvA_CC1pip_XSec_1Dth_nu"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
</nuisance>
```

- NUISANCE reads events from disk, and then distributes them to relevant sample classes.
- Minimal extra overhead when loading a number of different distributions or datasets from one MC file.
- Just add an extra sample xml entry for every dataset you care about and NUISANCE will load them all at once.

Multiple Sample Output

- Run this joint sample in a similar fashion.

```
$ nuiscomp -c samplecc1pip.xml -o samplecc1pip.root -n 100000
```

- Likelihoods for both samples added uncorrelated, to form a joint total likelihood for the comparison.

```
[LOG Minmzr]:- Getting likelihoods...                    : -2logL
[LOG Minmzr]:- -> MINERvA_CC1pip_XSec_1DTpi_nu           : 44.6792/7
[LOG Minmzr]:- -> MINERvA_CC1pip_XSec_1Dth_nu            : 260.352/13
[LOG Fitter]: Likelihood for JointFCN:                    305.031
```

- Two sets of histograms also now contained in the output file.

ReWeighting Predictions

```
<nuisance>
  <!-- List of parameters -->
  <parameter type="" name="" nominal="" />

  <!-- List of Samples -->
  <sample name="MINERvA_CC1pip_XSec_1DQ2_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
  <sample name="MINERvA_CC1pip_XSec_1Dth_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
</nuisance>
```

- Reweight parameters can be added to NUISANCE card files using a "parameter" xml structure.

```
<parameter type="DIAL_TYPE" name="NAME" nominal="DIAL_VALUE" />
```

- **Requirements:**
  - NAME : Name of the dial inside the RW Engine
  - DIAL_TYPE : RW Type (e.g. genie_parameter)
  - DIAL_VALUE : Current Value to use

Finding ReWeight Dials

```
<nuisance>
  <!-- List of parameters -->
  <parameter type="genie_parameter" name="" nominal="" />

  <!-- List of Samples -->
  <sample name="MINERvA_CC1pip_XSec_1DQ2_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
  <sample name="MINERvA_CC1pip_XSec_1Dth_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
</nuisance>
```

- Any dial name recognised by the RW engine is supported.
- List is in **$GENIE/src/ReWeight/GSyst.h**

```
$ grep case $GENIE/src/ReWeight/GSyst.h
case ( kXSecTwkDial_MaNCEL          ) : return "MaNCEL";              break;
case ( kXSecTwkDial_EtaNCEL         ) : return "EtaNCEL";             break;
case ( kXSecTwkDial_NormCCQE        ) : return "NormCCQE";            break;
...
```

- We want the Resonant Axial Mass ("MaCCRES")

Adding ReWeight

```
<nuisance>
  <!-- List of parameters -->
  <parameter type="genie_parameter" name="MaCCRES" nominal="" />

  <!-- List of Samples -->
  <sample name="MINERvA_CC1pip_XSec_1DQ2_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
  <sample name="MINERvA_CC1pip_XSec_1Dth_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
</nuisance>
```

- Now need to choose our current dial value.

```
nominal="DIAL_VALUE"
```

- Units are whatever the reweight engine uses.
- E.g. GENIE ReWeight usually considers dials in units of "1-sigma" from nominal with 0.0 being the default value

Running nuiscomp

```
<nuisance>
  <!-- List of parameters -->
  <parameter type="genie_parameter" name="MaCCRES" nominal="-1.0" />

  <!-- List of Samples -->
  <sample name="MINERvA_CC1pip_XSec_1DQ2_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
  <sample name="MINERvA_CC1pip_XSec_1Dth_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
</nuisance>
```

- Can rerun our edited card file and save the output somewhere else for later comparisons

```
$ nuiscomp -c samplecc1pip.xml -o samplecc1pip-rw.root -n 100000
```

- All MC curves have now been weighted with GENIE ReWeight set to MaCCRES = -1

Minimiser Routines

- Final layer of NUISANCE takes the convertor+comparison routines and tries to use it for model tuning.

- Direct interface with ROOT's minimiser libraries.

(Creating a Minimizer via the Plug-In Manager)

https://root.cern.ch/numerical-minimization

- Can treat any reweight parameter as free and try to minimise the joint χ^2 value between all samples.

- Mostly automated procedure, just list the datasets you want and the parameters you want to tune and leave it to run.

NuisMin Application

- Tuning application : **nuismin**

```
$ nuismin -c samplecard.xml \
          -o sampleccqe.root \
         [-n NEVENTS ]
```

- Runs using XML card just like nuiscomp. Can take our previous card files as a starting point.

- Main difference is in nuismin we must specifiy that we want some parameters to be treated as FREE.

Free Parameter Structures

- Free parameter structures are very similar to fixed parameters, but they require you to tell NUISANCE the limits, and what state it is in FIX or FREE.

```
<parameter type="genie_parameter" name="MaCCRES"
           nominal="" low="" high="" step="" state="" />
```

Requirements:

- Nominal = Current (Starting) Value
- Low = Lower Limit
- High = Upper Limit
- Step = Migrad Starting Step Size
- State = Parameter state : FIX or FREE

Multiple Parameters

- NUISANCE will take all "FREE" parameters and load them into a multi-dimensional ROOT minimisation.

```
<parameter type="genie_parameter" name="MaCCRES"
            nominal="1.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
<parameter type="genie_parameter" name="MaCCQE"
            nominal="1.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
```

- Possible to keep some parameters fixed at nominal in the fit by instead putting the state to "FIX"

```
<parameter type="genie_parameter" name="MaCCRES"
            nominal="1.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
<parameter type="genie_parameter" name="MaCCQE"
            nominal="1.0" low="-3.0" high="3.0" step="1.0" state="FIX" />
```

Tuning MaCCRES

```
<nuisance>
  <!-- List of parameters -->
  <parameter type="genie_parameter" name="MaCCRES"
             nominal="0.0" low="-3.0" high="3.0" step="1.0" state="FREE" />

  <!-- List of Samples -->
  <sample name="MINERvA_CC1pip_XSec_1DQ2_nu_2017"
          input="GENIE:genie/gntp.Default.MINERvA_fhc_numu.CH.2500000.root" />
</nuisance>
```

- We just want to float MaCCRES freely so we change our new parameter line to treat it as FREE.
- This card file will vary MaCCRES between $\pm 3\sigma$ and find the best fit it can to the 1DQ2 dataset.

Making a nominal plot

- Before we start tuning, we can use this new card file to create a nominal prediction for later comparisons.

- Running **nuiscomp** first over your new minimisercard.xml will produce the MC output at your starting values.

```
$ nuiscomp -c minimisercard.xml \
           -o minimiser-nominal.root \
           -n 100000
```

- This is a good practice as it also lets you check if all your samples are setup correctly before running a long fit.

Running

- Once you've written your card file it can be run in the same fashion but using nuismin this time.

```
$ nuismin -c minimisercard.xml \
          -o minimiser-tuned.root \
          -n 100000
```

- Minuit will scan the parameter space and try to find best fit.

```
[LOG Reconf]:---  -> Par 0. MaCCRES 0
[LOG Reconf]:--- Starting Reconfigure iter. 0
[LOG Minmzr]:- -> MINERvA_CC1pip_XSec_1DQ2_nu_2017          : 21.4312/8
…
[LOG Reconf]:---  -> Par 0. MaCCRES 0.0101951
[LOG Reconf]:--- Starting Reconfigure iter. 1
[LOG Minmzr]:- -> MINERvA_CC1pip_XSec_1DQ2_nu_2017          : 21.4962/8
```

Long Iterations

- Fits with multiple parameter scan take on the order of a day.
- NUISANCE is very I/O heavy. Have to read the full MC event so that it can be passed to the RW engine.
- 90% of each event loop is just reading events from disk. ☹
- Event loop optimized for multiple samples (each event is read from disk only once per event loop)
- Additional config flag (SignalReconfigures) can speed it up further by looping over only signal events after the first pass.

```
$ nuismin [ options ]  -q SignalReconfigures=1
```

Running Faster

```
$ nuismin -c minimisercard.xml \
          -o minimiser-tuned.root \
          -n 100000 \
          -q SignalReconfigures=1
```

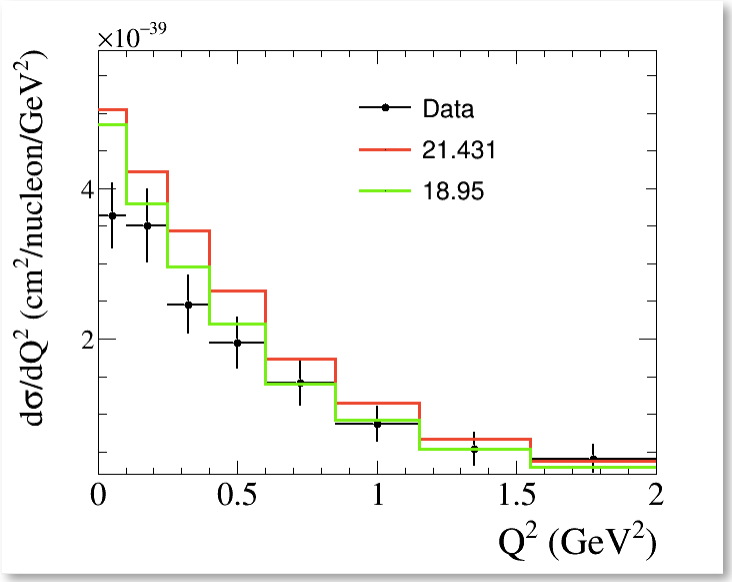
- Turning on SignalReconfigures will speed things up quite a bit.

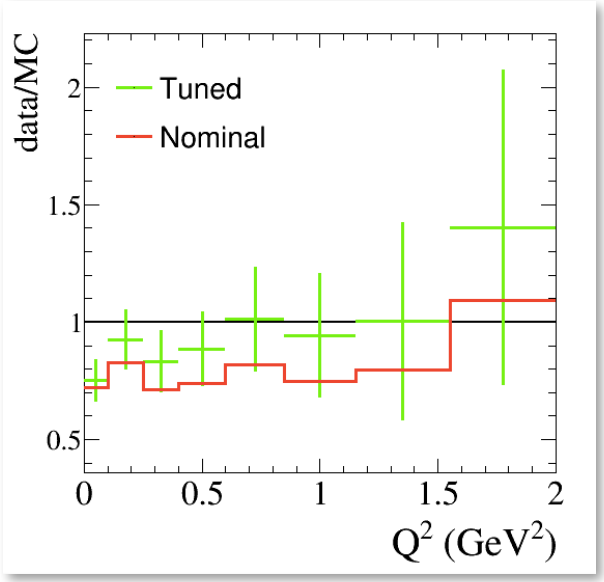
```
[LOG Minmzr]:- Finished Reconfigure iter. 4 in 3s
[LOG Minmzr]:- Getting likelihoods...                    : -2logL
[LOG Minmzr]:- -> MINERvA_CC1pip_XSec_1DQ2_nu_2017       : 21.4239/8
```

Minimiser Output

```
#    Parameter  = Value      +- Error      (Units)  Conv. Val  +- Conv. Err  (Units)
0  . MaCCRES    = -0.789184  +- 0.510301   (sig.)   -0.789184  +- 0.510301   (sig.)
```

- **nuismin** saves the same MC histograms as **nuiscomp** did.



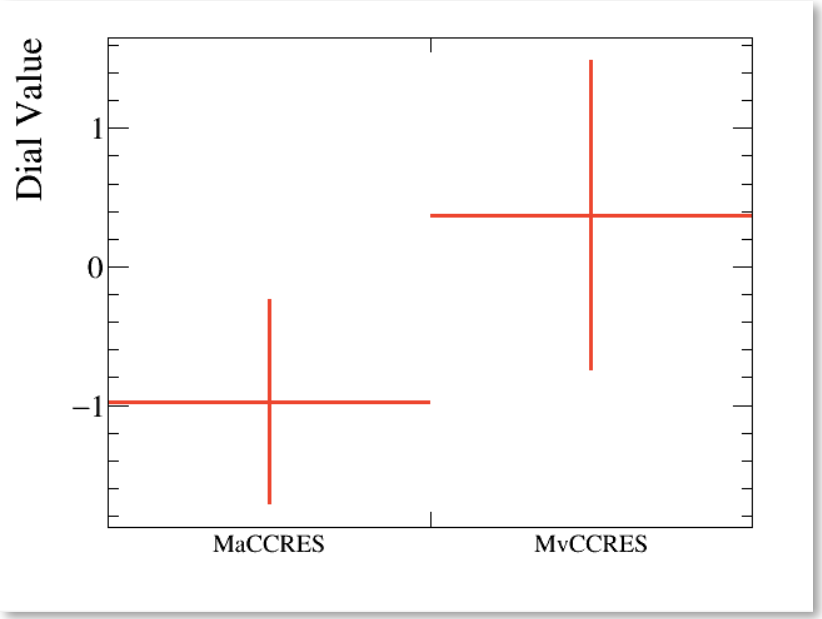


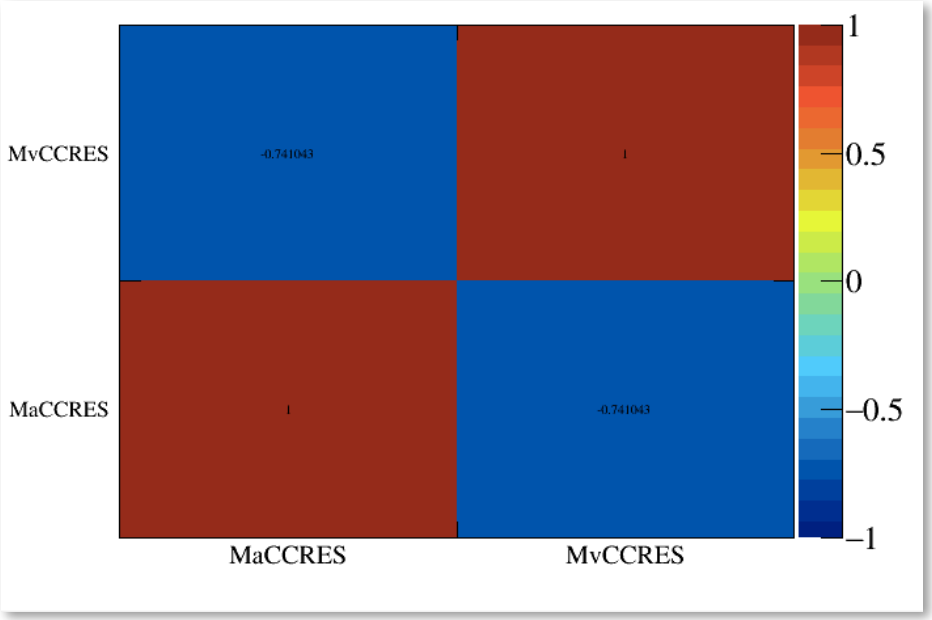
- Can compare our "nominal" and "tuned" results

```
$ root minimiser-tuned.root minimiser-nominal.root
```

Minimiser Output (2)

- Lots of other fit information saved into the output file.





- Some interesting ones:
  - **start_dials** : starting parameters
  - **fit_dials** : best fit parameters
  - **fit_iterations** : Parameter/Likelihood state at each iter.
  - **fit_result** : best fit parameter result (CHI2 = best joint fit likelihood)

Fit Types

- Extra handling options can be passed to each sample through the optional "type" field.

```
<sample name="NAME"    type="TYPEDEF"    input="INPUT" />
```

- Usually handles likelihood options. Many non-conflicting terms can be passed at once (e.g. DIAG/FREE/NORM).

- **A few examples for TYPEDEF:**
  - DIAG : Use diagonal errors instead of a covariance
  - SHAPE : Treat as a shape-only likelihood
  - FREE : Freely float the normalisation as a fit parameter

Changing the Routines

- Minimiser interface has a few different minimiser routines.

```
Brute
Simplex
Minuit
Combined
Fumili
```

```
ConjugateFR
ConjugatePR
BFGS
BFGS2
SteppDesc
```

```
GSLSimAn
```

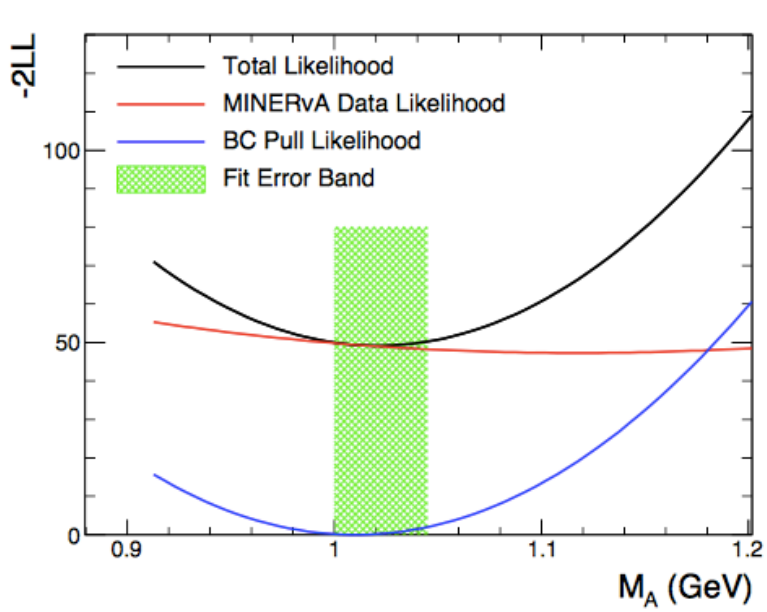
- Can use the (-f) flag in nuismin to choose which routine to run.

- Routines are comma seperated and ran in sequence, with the results of one routine being passed into the other.
- **Example** : run a brute force ND scan, then run Minuit from the new starting point.

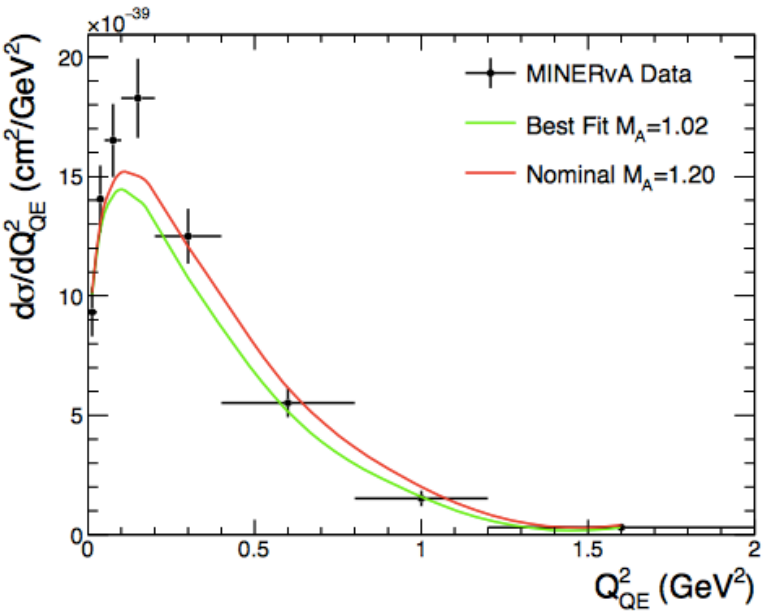
```
$ nuismin -c card.xml -o out.root -f Brute,Minuit
```

Running with simple priors

- Can include χ^2 penalty terms in fit using "covar" objects.



(a) χ^2 scan across the M_A parameter space



(b) Nominal and best fit MINERvA distribution

- Allows you to use the results of previous internal/external fits as an additional constraint in your tuning.

Simple Priors : DIAL

- Simplest "covar" type is an uncorrelated Gaussian **DIAL** pull.

- If you already have a parameter included in your card file called PARNAME you can place a Gaussian prior on it with:

```
<covar name="PARNAME_prior"
       input="DIAL:PARNAME;CENTRAL_VALUE;ERROR_VALUE"
       type="GAUSPULL" />
```

- **Example:** Float MaCCRES, but include an additional constraint of MaCCRES = -0.5 +- 1.0 sigma in the fit.

```
<parameter name="MaCCRES" type="genie_parameter"
           nominal="0.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
<covar name="MaCCRES_prior" input="DIAL:MaCCRES;-0.5;1.0" type="GAUSPULL" />
```

Simple Priors : ROOT

- Can also specify similar Gaussian priors with full correlations between parameters.

- Requires TH1D central values (TH1DCV) and TH2D covariance (TH2DCOV) with the bin labels set to match the dial names.

```
<covar name="ID" input="ROOT:FILEPATH;TH1DCV;TH2DCOV" type="GAUSPULL" />
```

- Exact histogram format that NUISANCE saves outputs, so can easily use a previous fit result as a future prior.

```
<parameter name="MaCCRES" type="genie_parameter"
           nominal="0.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
<parameter name="NonResBkgvnCC1pi" type="genie_parameter"
           nominal="0.0" low="-3.0" high="3.0" step="1.0" state="FREE" />
<covar name="fit_prior" input="ROOT:result.root;fit_dials;covariance" type="GAUSPULL" />
```

gevgen_nuisance

- Custom gevgen app can be used to generate GENIE events specially for NUISANCE.
- Automatically saves required histograms into output file.
- Can also run with both combined targets and combined beams (i.e. nue+nuebar)

```
gevgen_nuisance [-h]
         [-r run#]
          -n nev
          -e energy (or energy range)
        -p neutrino_pdg
        -t target_pdg       [DIFFERENT TO GENIE's]
        -f flux_description [DIFFERENT TO GENIE's]
       [-o outfile_name]
       [-w]
       [--seed random_number_seed]
       [--cross-sections xml_file]
       [--event-generator-list list_name]
       [--message-thresholds xml_file]
       [--unphysical-event-mask mask]
       [--event-record-print-level level]
       [--mc-job-status-refresh-rate  rate]
       [--cache-file root_file]
```

gevgen_nuisance

- Options are similar to the standard gevgen app, but target and flux are different (and easier!)

```
gevgen_nuisance -f MINERvA_fhc_numu -t CH  <other arguments>
```

- Only works with GENIE 2.12 and later!

- To build this application, build NUISANCE with the following flags

```
cmake -DUSE_GENIE=1 -DBUILD_GEVGEN=1
```

gevgen_nuisance (2)

- Possible to generate events with the standard gevgen application and "prepare" them for NUISANCE if needed.

- Example generate MINERvA CH events and prepare them.

```
gevgen -f minerva_flux.root,numu -e 0.0,100.0
                  -t 1000060120[0.9231],1000010010[0.0769]
                 -r 1 -n 2500000 --cross-sections gxspl.gz
```

- Once the sample is made, need to prepare it using our PrepareGENIE application (note the target def is different!)

```
PrepareGENIE -i gntp.ghep.1.root -f minerva_flux.root,numu
              -t 1000060120,1000010010
```

Target is not fractional! For CH2 use
`1000060120,1000010010,1000010010`

Have to pass in the same flux again too..

PrepareNuWro

- NuWro events generated with nuwro-reweight (our special branch) automatically saves the information needed.

- Otherwise, there is another PrepareNuWro app.

```
[USAGE]: PrepareNuwro [-h]
                      [-f]
                      [-F <FluxRootFile>,<FluxHistName>]
                      [-o output.root]
                      inputfile.root [file2.root ...]

-h : Print this message.
-f : Pass -f argument to '$ hadd' invocation.
-F : Read input flux from input descriptor.
-o : Write full output to a new file.
```

Configs

- NUISANCE keeps a global configuration list accessible throughout the code.
- Defaults kept in `$NUISANCE/parameters/config.xml`
- These can be overriden at run time in the card file or on the command line
- Card File

```
<config NAME="OVERRIDE_VALUE" />
```

- Command Line (all applications support –q)

```
nuiscomp -c cardfile.xml -o output.root -q NAME=OVERRIDE_VALUE
```