NUISANCE and you

JINST 12 P01016 (2017)
nuisance.hepforge.org
github.com/NUISANCEMC/nuisance/
nuisance-xsec.slack.com


Clarence Wret


Luke Pickering


Callum Wilkinson


Stephen Dolan


Patrick Stowell

UNIVERSITY OF OXFORD

ROYAL HOLLOWAY UNIVERSITY OF LONDON

BERKELEY LAB
Lawrence Berkeley National Laboratory

CERN

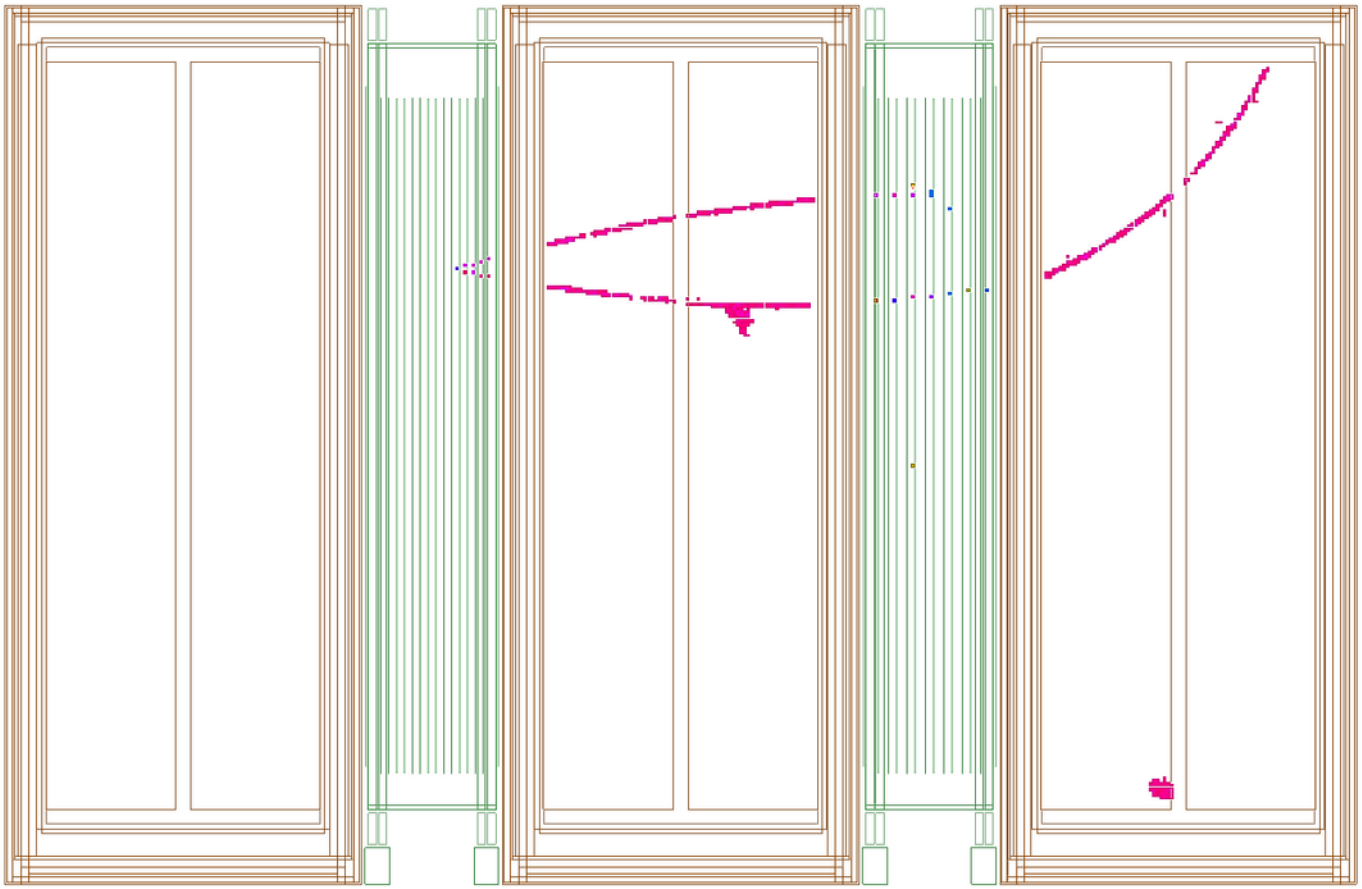
Durham University

Neutrino event-generator workshop
Fermilab, 17 March 2023

With contributions C. Riccio, K. Duffy, A. Mastbaum, S. Gardiner, J. Kim, D. Ruterbories, K. Mahn, K. McFarland, Y. Hayato, J. Sobzcyk, K. Niewczas, U. Mosel, and others

Introduction

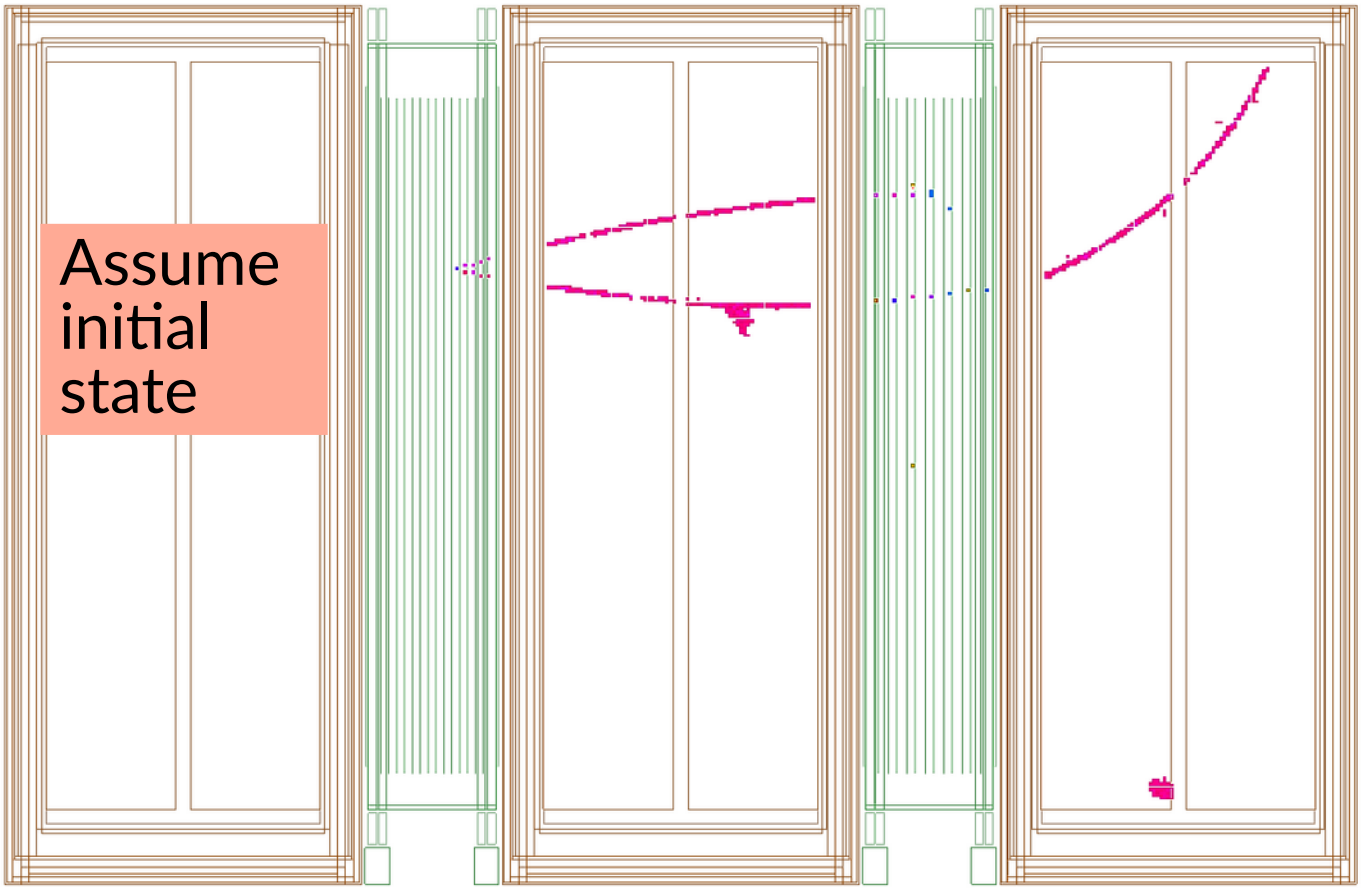
- Why do we need neutrino cross-section dependence at all?
  - Can't reliably measure the fundamental interaction quantities that our models depend on (E_ν, Q^2, W, q_0, $\mathbf{q_3}$, **...**)



- Need to relate observed event to the true quantity through some model

Introduction

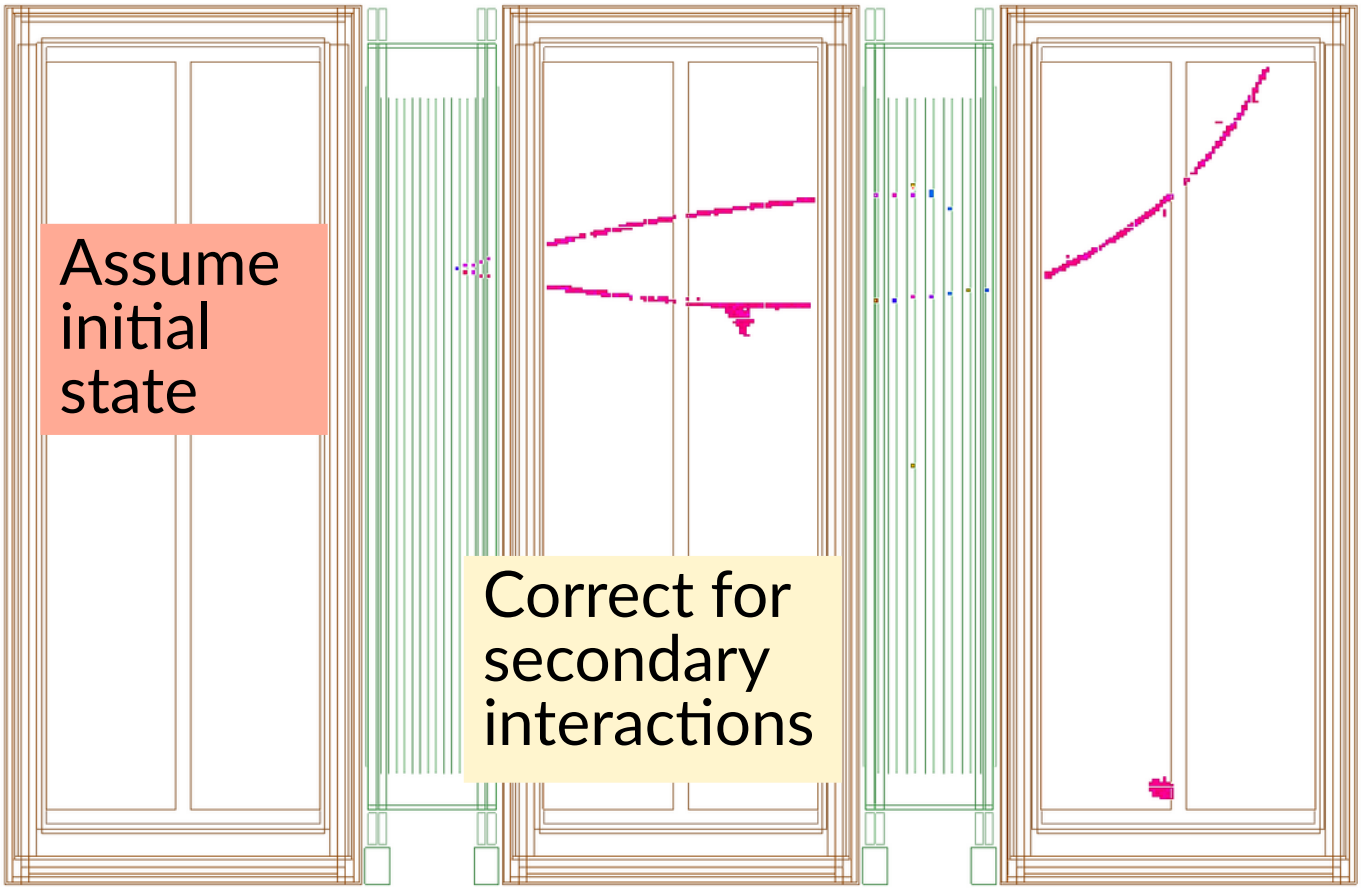
- Why do we need neutrino cross-section dependence at all?
  - Can't reliably measure the fundamental interaction quantities that our models depend on (E_ν, Q^2, W, q_0, $\mathbf{q_3}$, **...**)



- Need to relate observed event to the true quantity through some model

Introduction

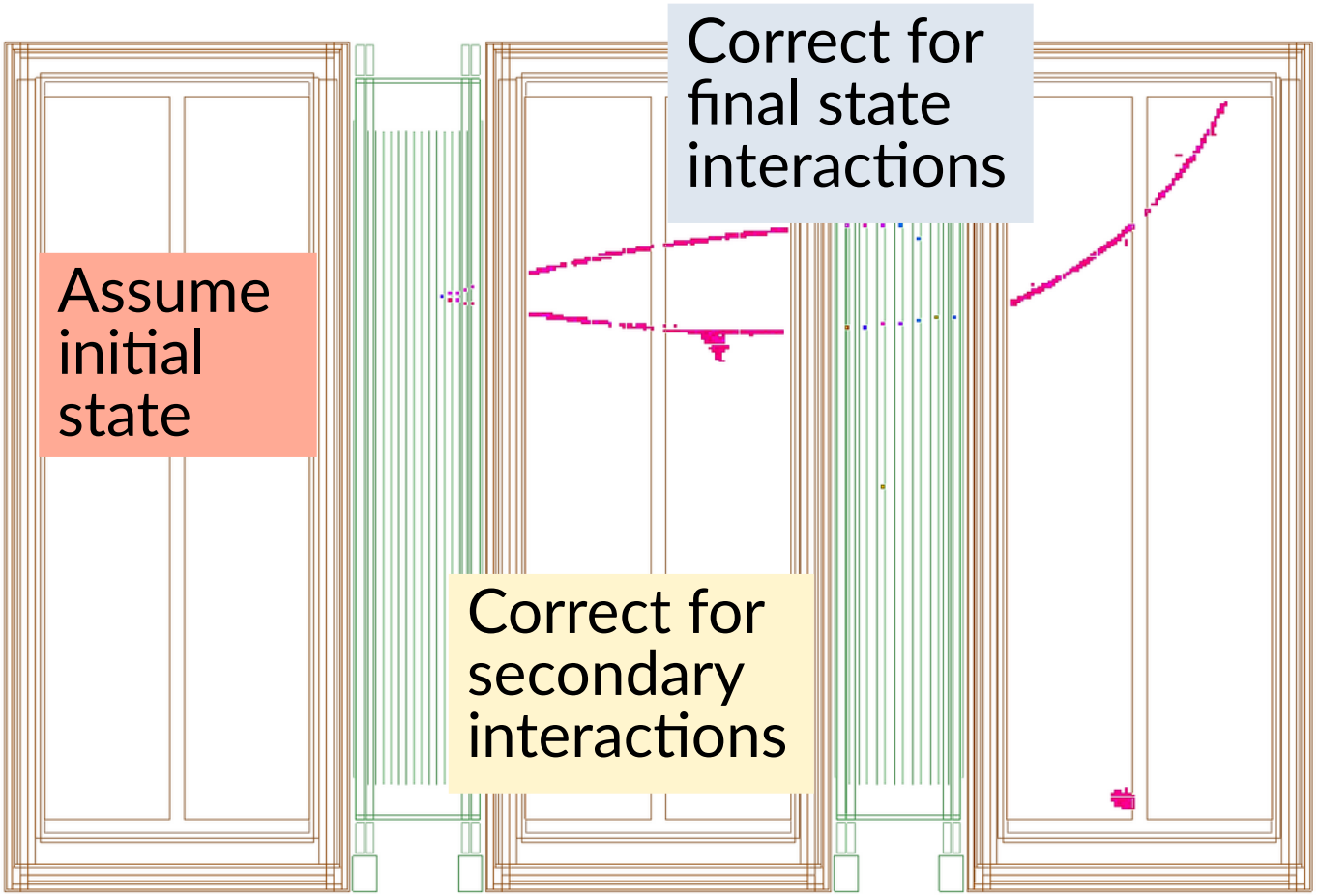
- Why do we need neutrino cross-section dependence at all?
  - Can't reliably measure the fundamental interaction quantities that our models depend on (E_ν, Q^2, W, q_0, $\mathbf{q_3}$, **...**)



- Need to relate observed event to the true quantity through some model

Introduction

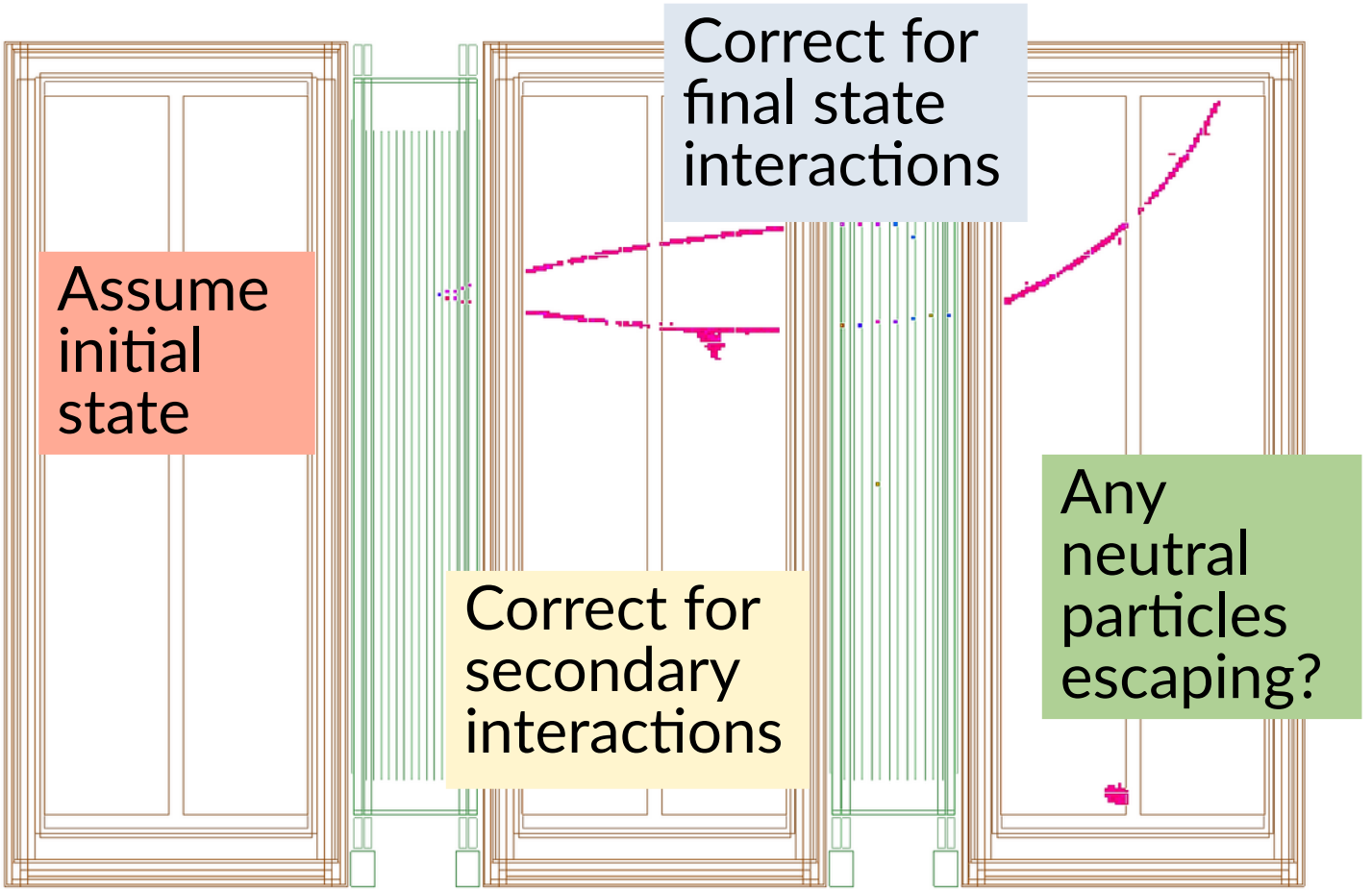
- Why do we need neutrino cross-section dependence at all?
  - Can't reliably measure the fundamental interaction quantities that our models depend on (E_ν, Q^2, W, q_0, $\mathbf{q_3}$, **...**)



- Need to relate observed event to the true quantity through some model

Introduction

- Why do we need neutrino cross-section dependence at all?
  - Can't reliably measure the fundamental interaction quantities that our models depend on (E_ν, Q^2, W, q_0, $\mathbf{q_3}$, **...**)



- Need to relate observed event to the true quantity through some model

UNIVERSITY OF OXFORD

Introduction

- Can we escape model dependence? Arguably not
  - Even a perfect detector won't be able to tell you about final-state interactions, or the initial state
- But we can **remove dependence on models that have shaky foundations**!
  - Does the model fail to describe reliable data?
  - How does the model compare to other currently approved calculations and models?
  - Etc...
- The community needed tools to inform us of where models are doing well, and where they aren't
  - Design physics analyses to expose weaknesses in modelling
  - Avoid physics analyses that depend on unreliable model predictions
  - Rinse, repeat, and get more robust and valuable measurements!

Introduction

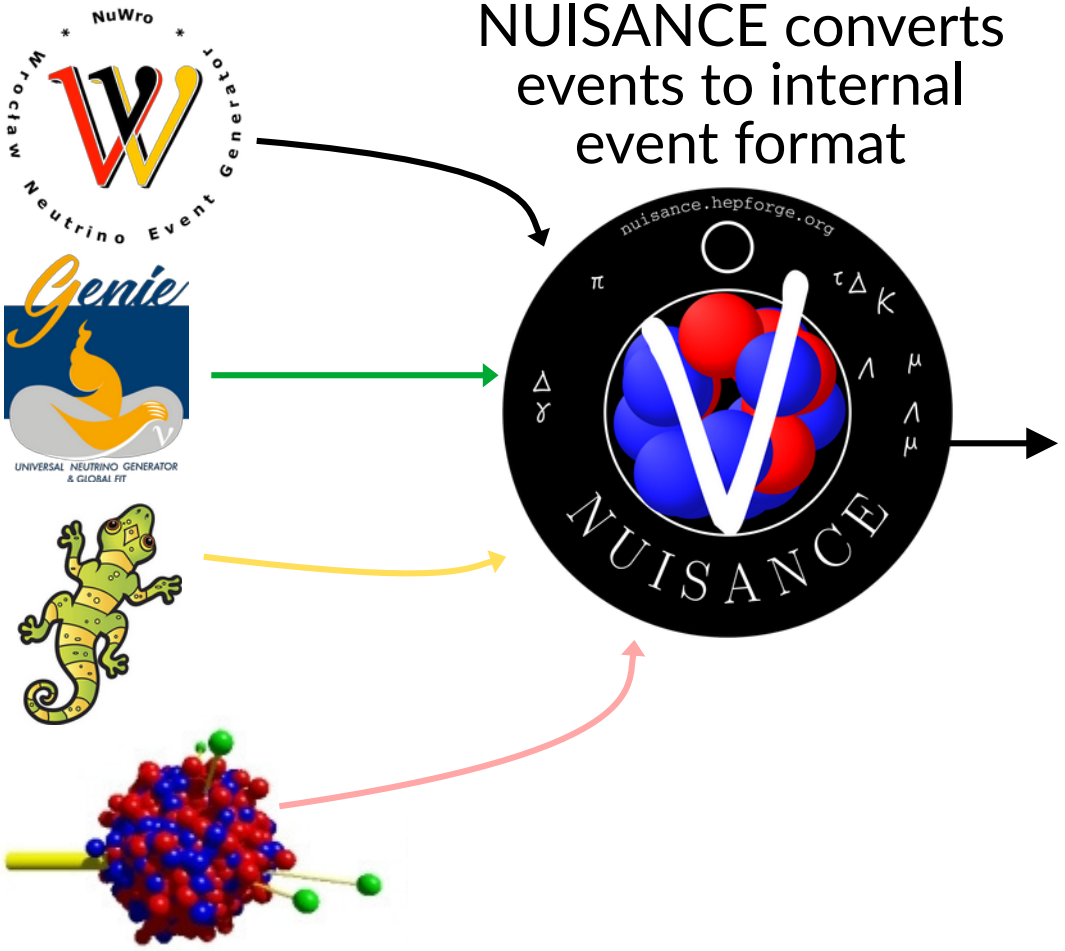
- The generator market is quite vast, and expanding!
  - GENIE, NEUT, NuWro, GiBUU, Achilles, NUANCE, …
  - **No clear winner for experiments**: some generators have excellent integration into experiments, others have very detailed nuclear model implementations but less developed uncertainty model, and so on
- Identified a need to easily compare **different** generator predictions to each other and to data
  - Develop and estimate uncertainties in analyses, using both generators and external data
  - Expose differences between generators and models for improved analyses
  - Identify interesting measurements for experimentalists to pursue and maximise profit of their analyses
  - Check effects of theory and phenomenology implementations against data and previous calculations
  - Get an idea of how model-dependent measurements may be
- Keep it **open source** with ease-of-usability in mind

The NUISANCE process

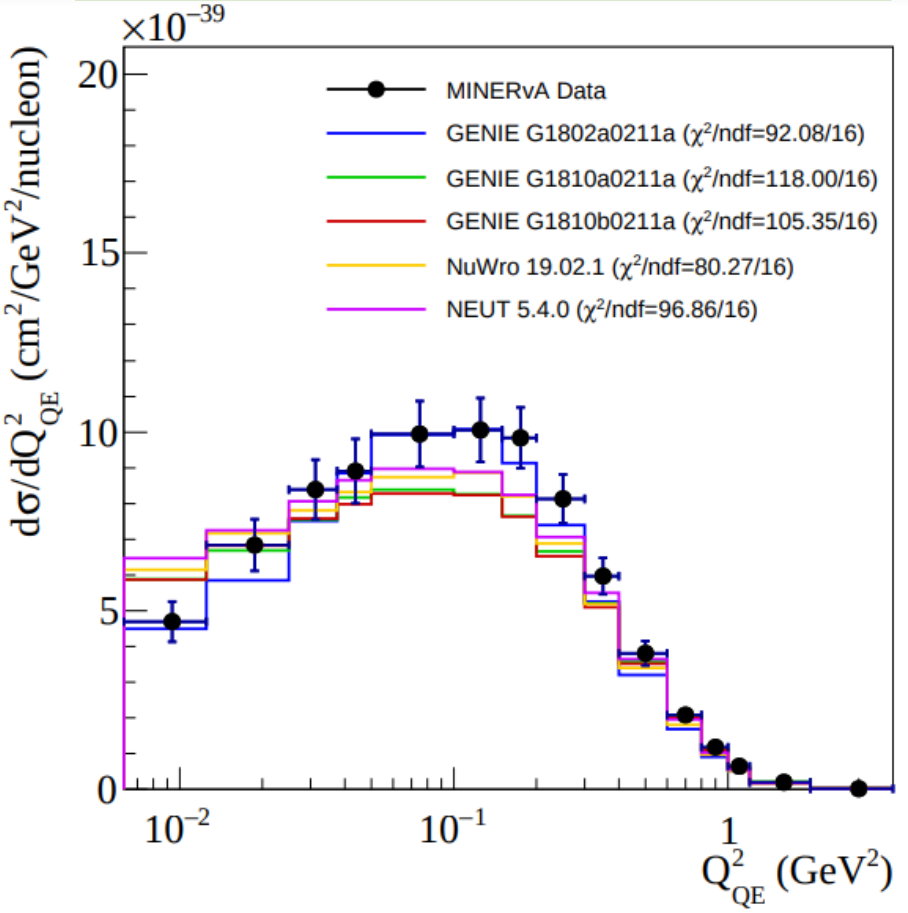
- All driven by simple commands, where a config file with the measurement and systematic parameters are provided

Generate events

NUISANCE converts events to internal event format



Compare generators to data



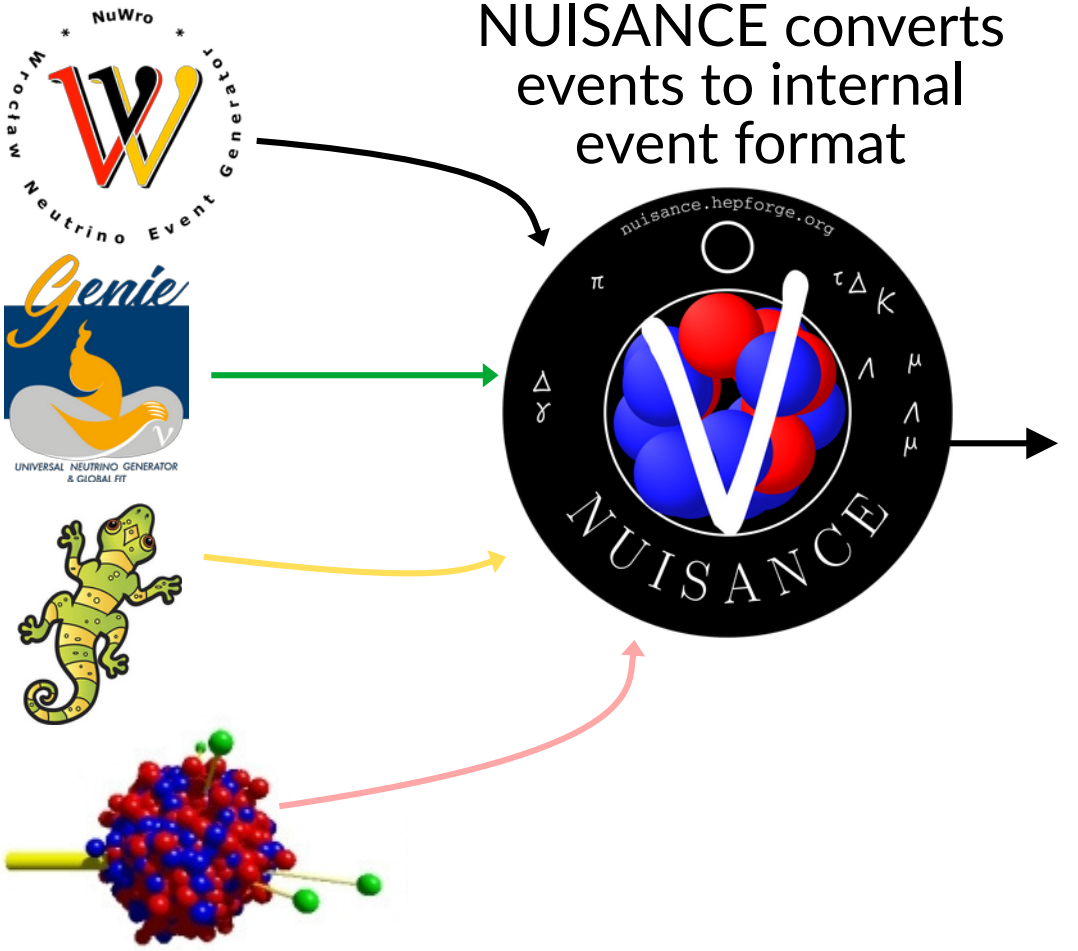
The NUISANCE process

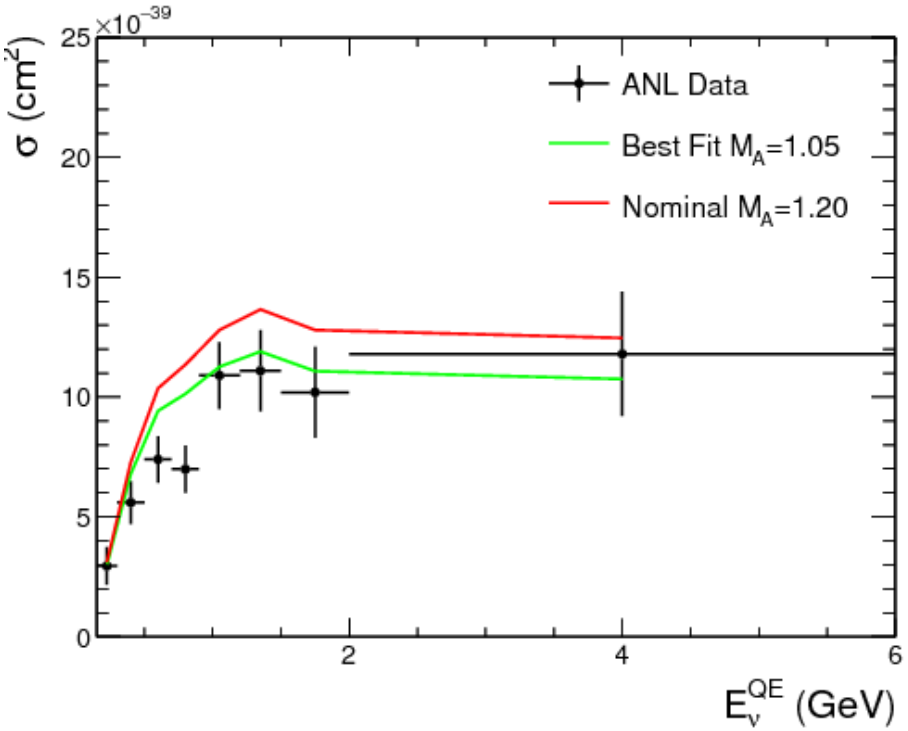
- All driven by simple commands, where a config file with the measurement and systematic parameters are provided

Generate events

NUISANCE converts events to internal event format

Fit generators model parameters to specific data





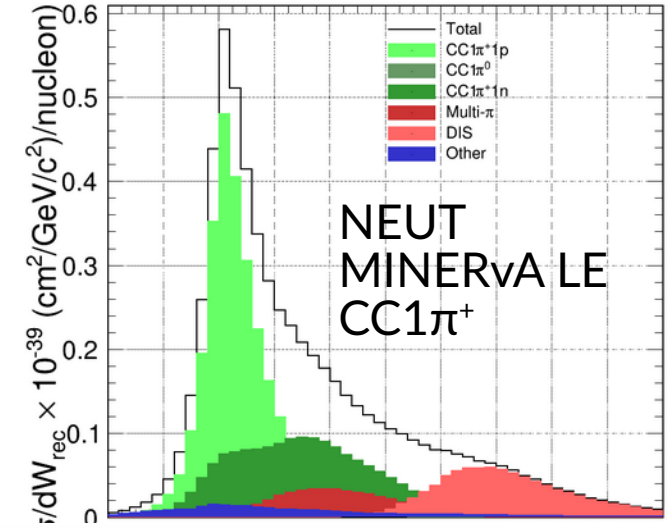
The NUISANCE process

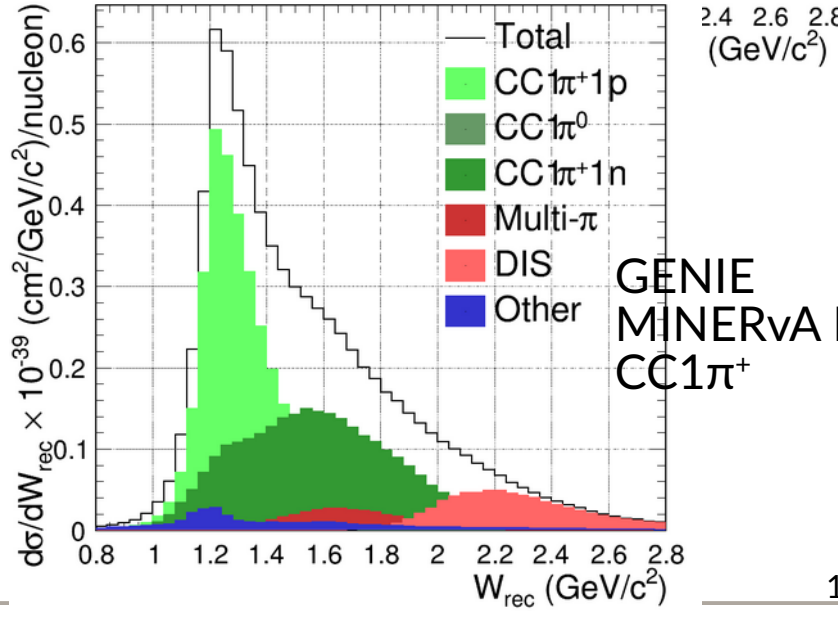
- All driven by simple commands, where a config file with the measurement and systematic parameters are provided

Generate events

NUISANCE converts events to internal event format

Compare generator features



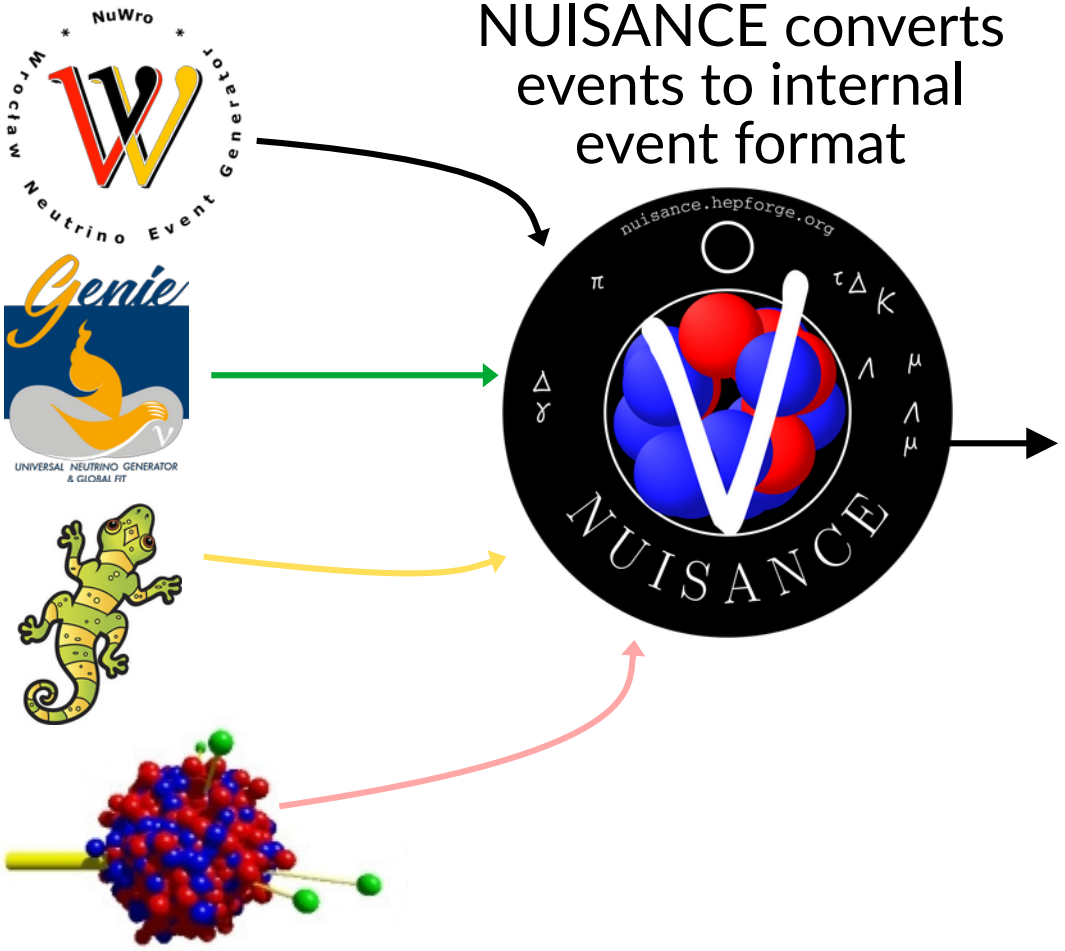


The NUISANCE process

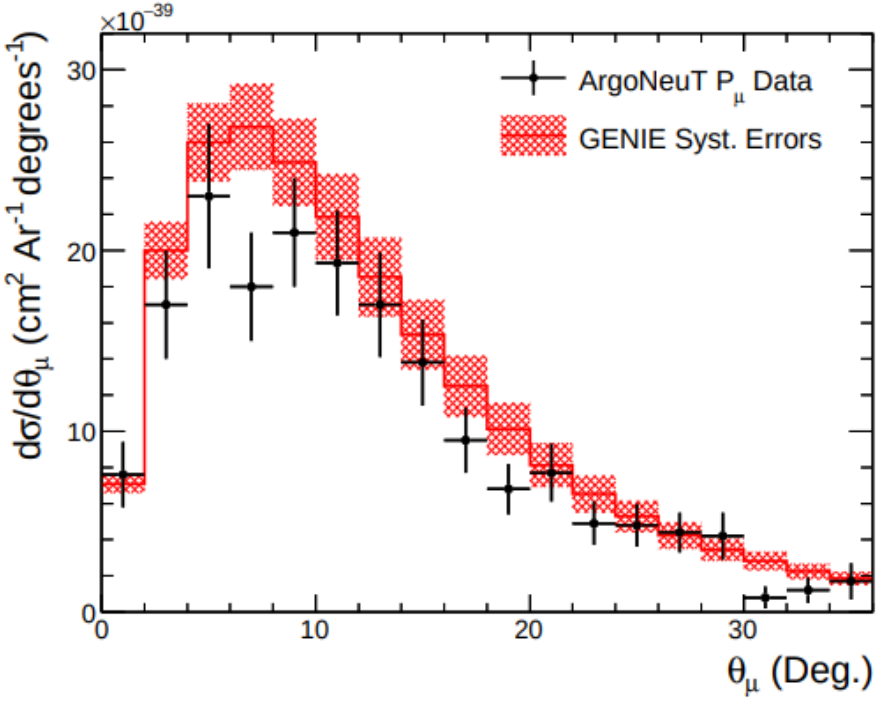
- All driven by simple commands, where a config file with the measurement and systematic parameters are provided

Generate events

NUISANCE converts events to internal event format



Evaluate uncertainties in the model against data



What can NUISANCE do?

- At its heart, NUISANCE is an event converter, with interfaces to:
- Compare your generators to over **350 implemented data sets**
- Interfaces with **reweighting engines**
  - GENIE, NEUT and T2K ReWeight, custom reweighting, MINERvA reweighting, DUNE's systematics packages, etc
  - Interfaces directly to physics; no assumptions on distributions etc
  - You can also add your own!
- Estimate the **uncertainty band of your model** against a vast array of data
- JINST paper reviewer referred to it as "RIVET for neutrino cross-section models and data"
- Interfaces with an array of minimisers (e.g. Minuit, GSL, MCMC) to **fit your model to data**
  - Fit whatever model you want, to whatever data you want
  - Can also fit GENIE model to NuWro fake data, and so on
  - NUISANCE does **not** ensure that your physics model is sound

Typical implementation

- Open to collaboration, with open source code base, operating on a pull request basis
  - Documentation available, slack workspace, maintained containers
- We have ~360 data sets implemented
  - And this list is rapidly growing with multiple publications every year!
  - Some collaborators have implemented their own measurements into NUISANCE and used it for multi-generator predictions in publication, excellent!

- A NUISANCE implementation needs to define
  - The signal definition
  - The dependent variable(s)
  - An interface to the data and covariance matrix
- All the generator conversions, event loops, reweighting procedures, etc is all done internally, under the hood
  - Shouldn't have to be an expert to perform studies!

Typical implementation

```
//********************************************************************
void MINERvA_CC0pi_XSec_2D_nu::FillEventVariables(FitEvent *event) {
  //******************************************************************
  // Checking to see if there is a Muon
  if (event->NumFSParticle(13) == 0)
    return;

  // Get the muon kinematics
  TLorentzVector Pmu = event->GetHMFSParticle(13)->fP;
  TLorentzVector Pnu = event->GetNeutrinoIn()->fP;

  Double_t px = Pmu.X() / 1000;
  Double_t py = Pmu.Y() / 1000;
  Double_t pt = sqrt(px * px + py * py);

  // y-axis is transverse momentum for both measurements
  fYVar = pt;

  // Don't want to assume the event generators all have neutrino coming along
  // z pz is muon momentum projected onto the neutrino direction
  Double_t pz = Pmu.Vect().Dot(Pnu.Vect() * (1.0 / Pnu.Vect().Mag())) / 1000.;
  // Set Hist Variables
  fXVar = pz;
};
```

A NUISANCE event (from your generator of choice)

Define the y-variable

Define the x-variable

Example of MINERvA CC0π 2D $p_t p_z$ implementation

Typical implementation

Muon cuts

Count up particles, make multiplicity cuts

Link to source code

```
//********************************************************************
bool isCC0pi_MINERvAPTPZ(FitEvent *event, int nuPDG, double emin, double emax) {
  //********************************************************************
  // Check it's CCINC
  if (!SignalDef::isCCINC(event, nuPDG, emin, emax))
    return false;

  // Make Angle Cut > 20.0
  TLorentzVector pnu = event->GetHMISParticle(14)->fP;
  TLorentzVector pmu = event->GetHMFSParticle(13)->fP;
  double th_nu_mu = FitUtils::th(pmu, pnu) * 180. / M_PI;
  if (th_nu_mu >= 20.0)
    return false;

  int genie_n_muons = 0;
  int genie_n_mesons = 0;
  int genie_n_heavy_baryons_plus_pi0s = 0;
  int genie_n_photons = 0;

  for (unsigned int i = 0; i < event->NParticles(); ++i) {
    FitParticle *p = event->GetParticle(i);
    if (p->Status() != kFinalState)
      continue;

    int pdg = p->fPID;
    double energy = p->fP.E();

    if (pdg == 13) {
      genie_n_muons++;
    } else if (pdg == 22 && energy > 10.0) {
      genie_n_photons++;
    } else if (abs(pdg) == 211 || abs(pdg) == 321 || abs(pdg) == 323 ||
               pdg == 111 || pdg == 130 || pdg == 310 || pdg == 311 ||
               pdg == 313 || abs(pdg) == 221 || abs(pdg) == 331) {
      genie_n_mesons++;
    } else if (pdg == 3112 || pdg == 3122 || pdg == 3212 || pdg == 3222 ||
               pdg == 4112 || pdg == 4122 || pdg == 4212 || pdg == 4222 ||
               pdg == 411 || pdg == 421 || pdg == 111) {
      genie_n_heavy_baryons_plus_pi0s++;
    }
  }
```

Example of MINERvA CC0π 2D $p_t p_z$ implementation

Typical running

Define samples, input generator, and input ROOT files containing events

```
<nuisance>
  <!-- define samples -->
  <sample name="MINERvA_CC0pi_XSec_1DQ2_antinu_H" input="NEUT:MIN_CH_ME_numub_radcorr_H_2003_2_megamerge.root" />
  <sample name="T2K_NuMu_CC0pi_CH_XSec_2DPcos" input="NEUT:neut_T2K_numu_fhc.root" />

  <!-- define some generator systematic parameters -->
  <parameter name="RadCorrQ2" nominal="1" type="custom_parameter" state="FIX"/>
  <parameter name="AxlFFCCQE" nominal="1" type="neut_parameter" state="FIX"/>
  <parameter name="MaCCQE" nominal="1.05" type="neut_parameter" state="FIX"/>

  <config NEUT_CARD="MIN_CH_ME_numub.card" />
</nuisance>
```

Use systematics from NUISANCE, or suitable generator (here NEUT)

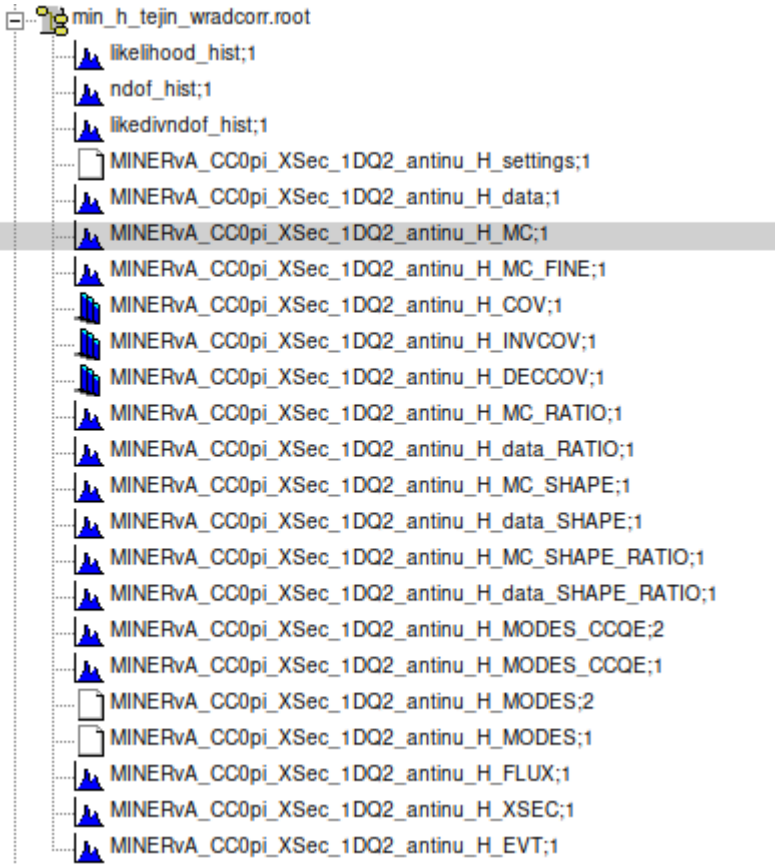
Turn on radiative corrections
Change form-factors
Change M_A^{QE}

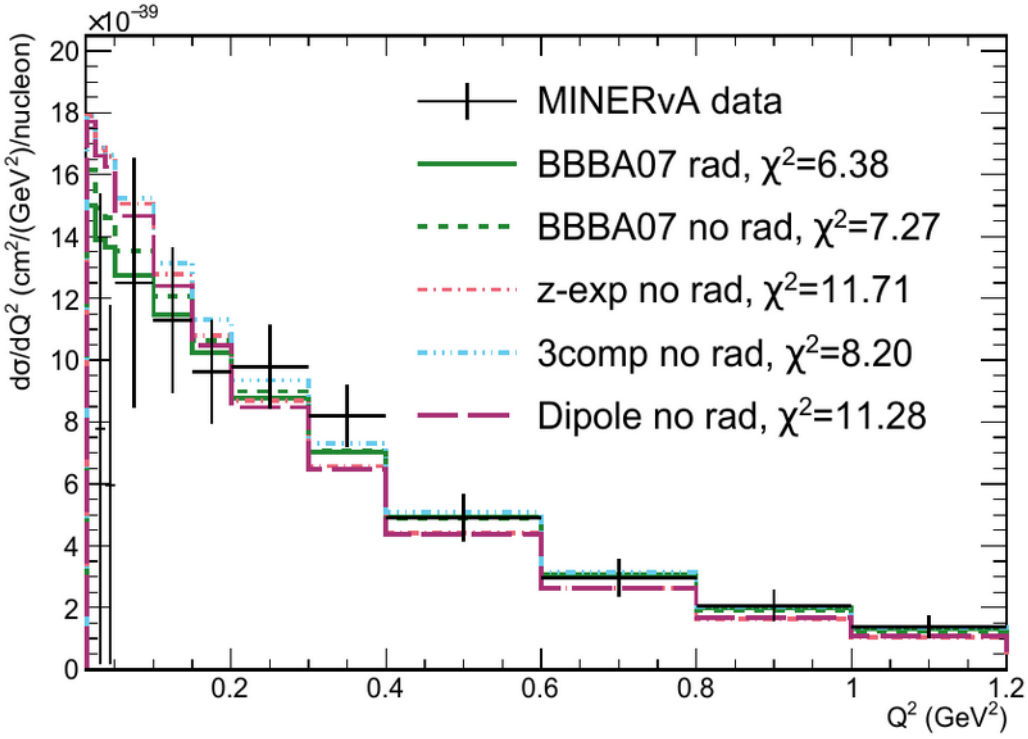
Typical output

Run comparison executable

```
[cvw09@lx02 app]$ ./nuiscomp -c example.card -o example.root
```

Output ROOT file with histograms and metadata





Compare multiple models against data, with full χ^2 using covariance matrices

Comparing generators

- Have heard a few times at workshops “would be great to compare generators to generators”

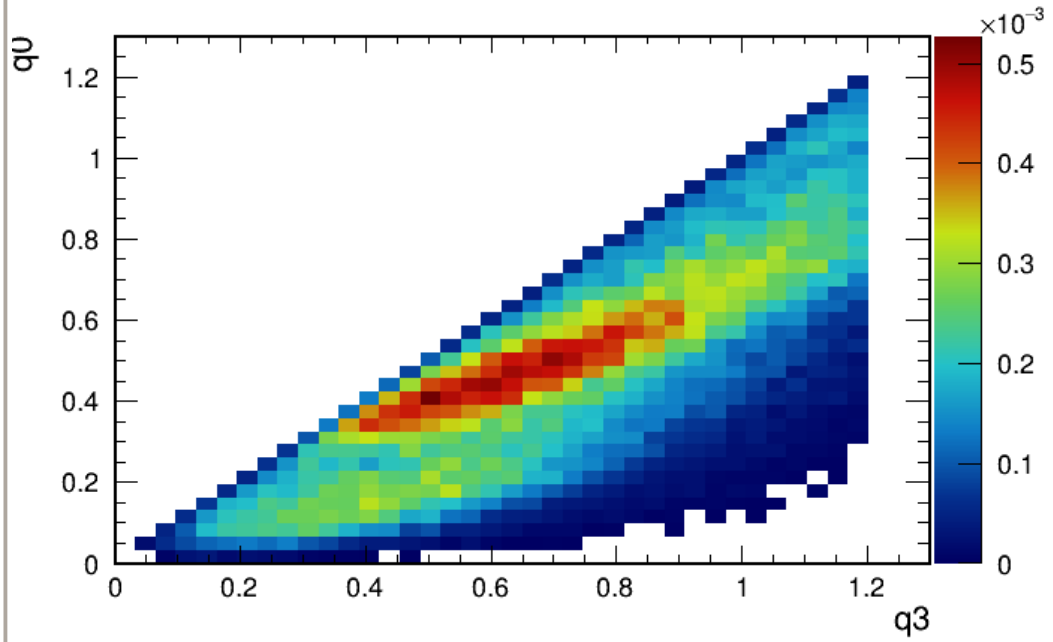
Draws the 2p2h prediction for MINERvA for NuWro and GENIE, both with Nieves 2p2h

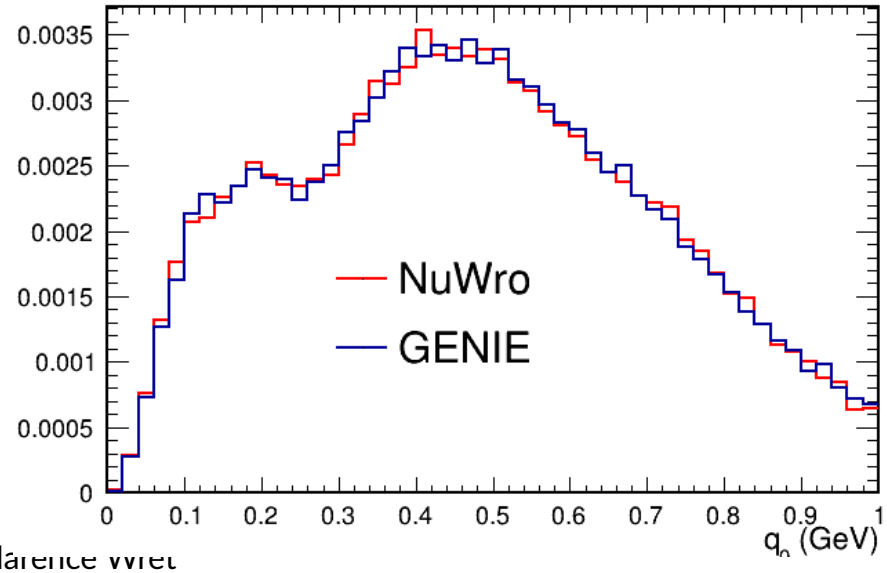
```
root [1] TFile *nuwro = new TFile("nuwro_minerva_numu_CH_2mega_NUISPREP_flat.root",
"open")
root [2] TFile *genie = new TFile("tensions-2019-MINERvA-numu-G18_02a_02_11a-
NUISANCE.flat.root", "open")
root [3] nuwro->cd()
root [4] FlatTree_VARS->Draw("q0:q3", "fScaleFactor*1E38*(Mode==2 && q0 < 1.5 && q3
< 1.5)","colz")
root [5] genie->cd()
root [6] FlatTree_VARS->Draw("q0:q3", "fScaleFactor*1E38*(Mode==2 && q0 < 1.5 && q3
< 1.5)","colz")
```

- Had a dedicated workshop with details, Tensions 2019 (Phys. Rev. D 105, 092004)

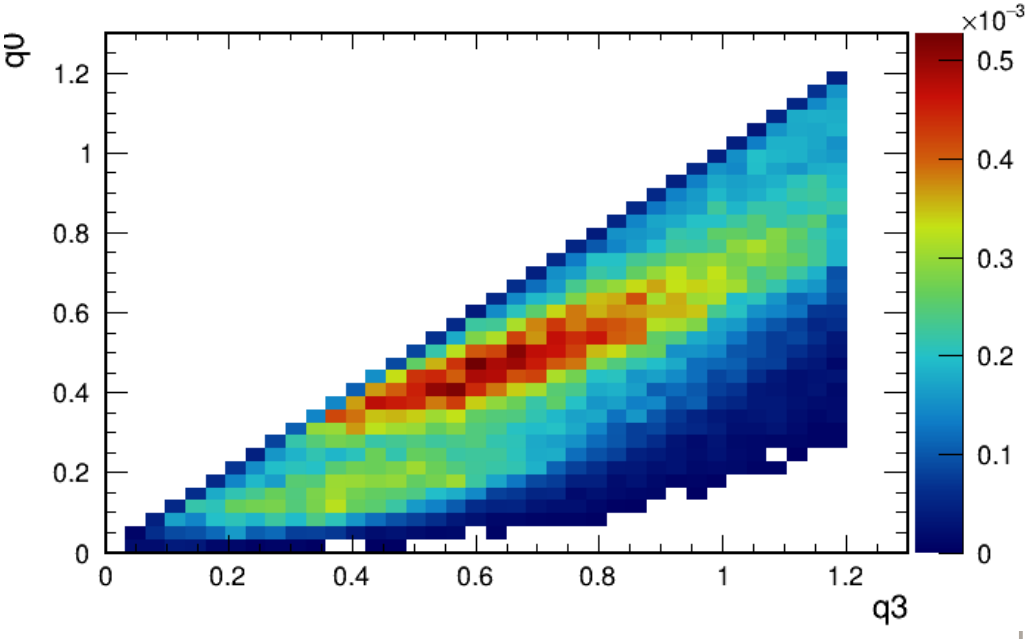
Comparing generators

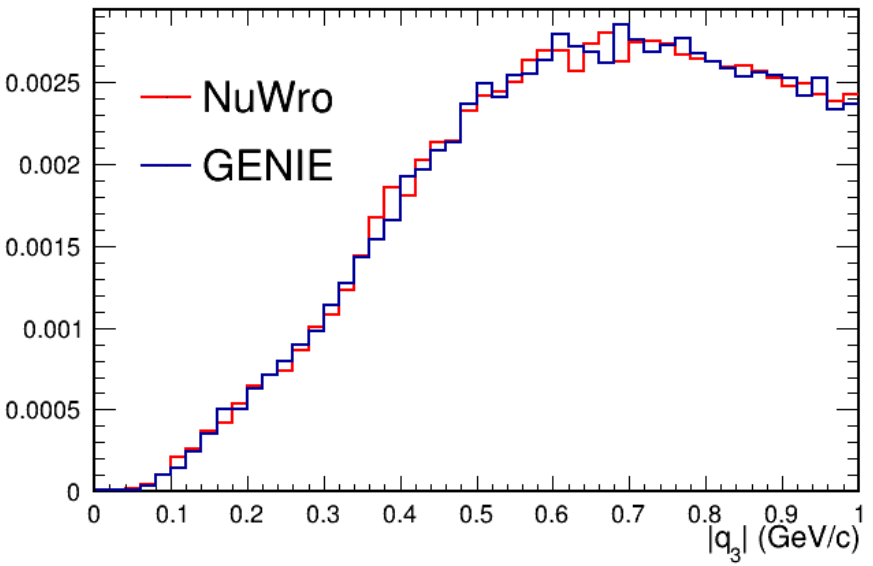
NuWro





GENIE

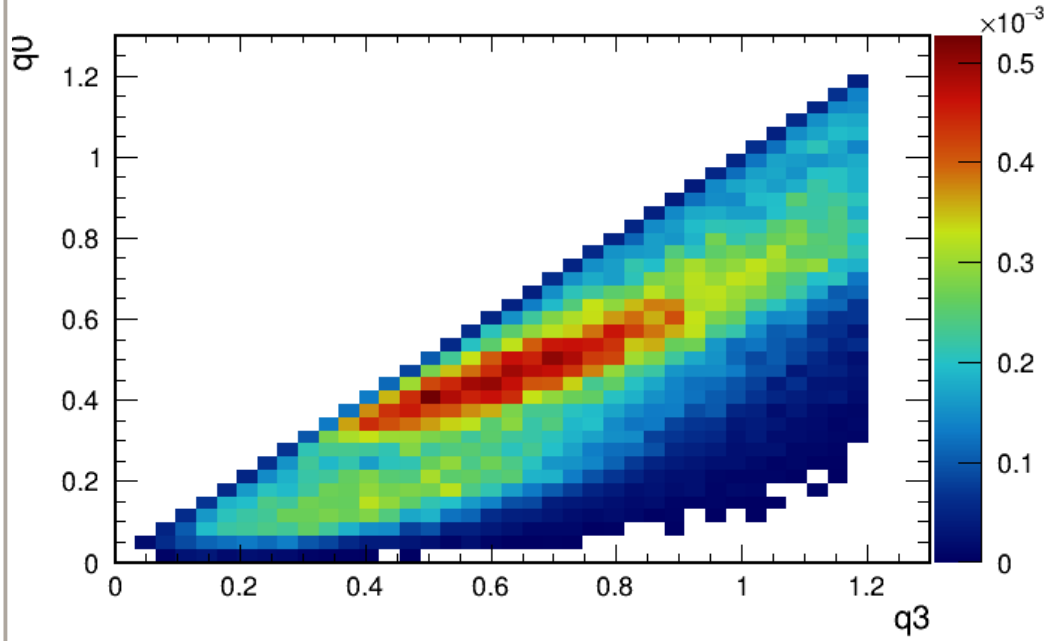




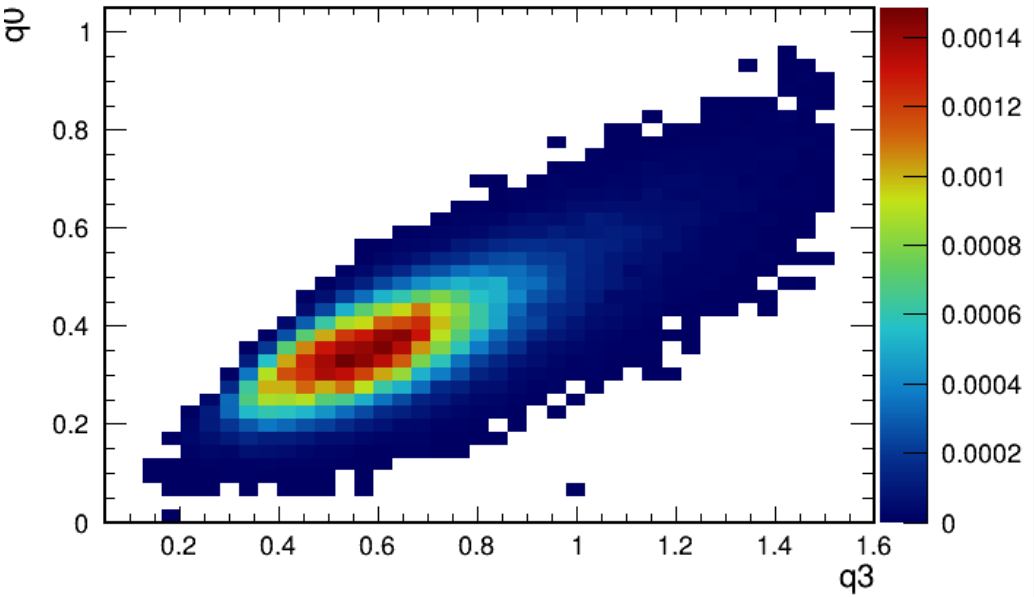
Comparing generators

Or compare different 2p2h models
in different generators

NuWro



GENIE Empirical MEC



Typical implementation

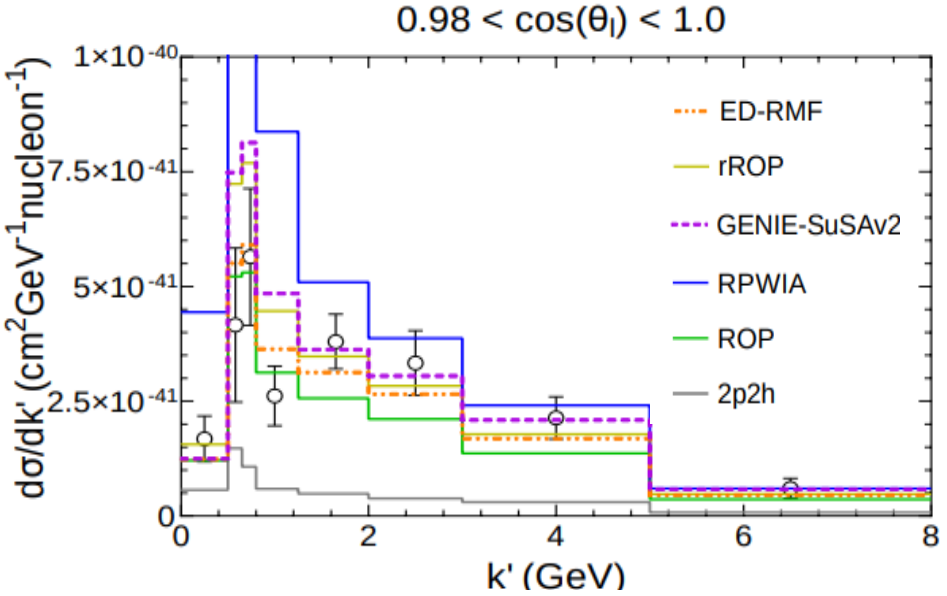
- Actively collaborate with experiment on implementations
- Validate against the generator prediction that is published using same generator
- Signal definition clarifications, defining variables, etc
- Work together on data releases, or at least identify needs
  - Avoids revisiting data release due to broken covariance matrix, unclear signal definitions, typos in papers… (all of which have happened)

Carefully validated and implemented data release, ensuring physics usage for years

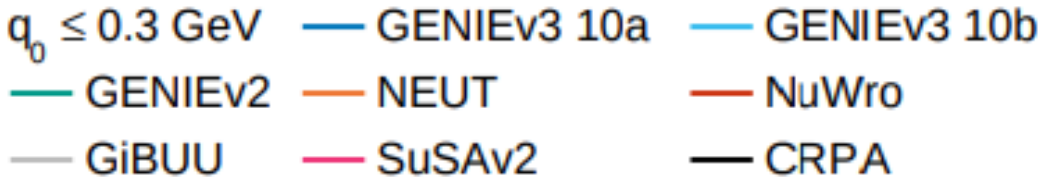
Uninvertible covariance matrix, where student leaves for industry after graduation, leaving a publication without any practical application

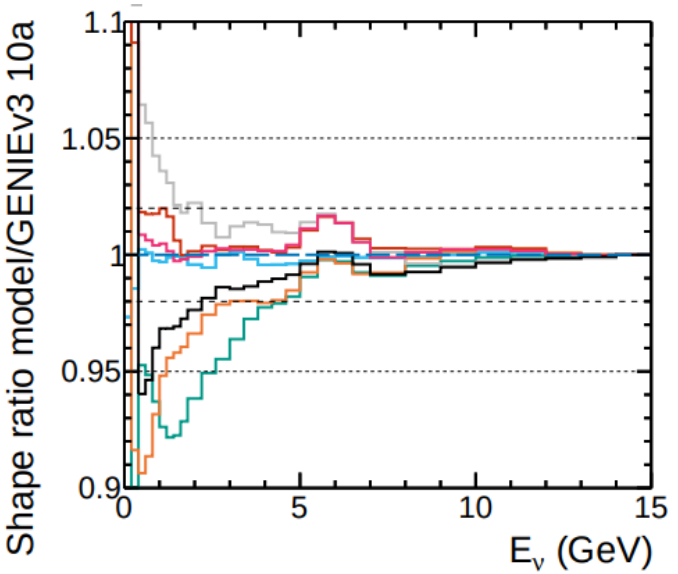
Recent examples

- Use GENIE predictions with SuSAv2 and compare to other 1p1h calculations (J.M. Franco-Patino et al. Phys. Rev. D 106, 113005)



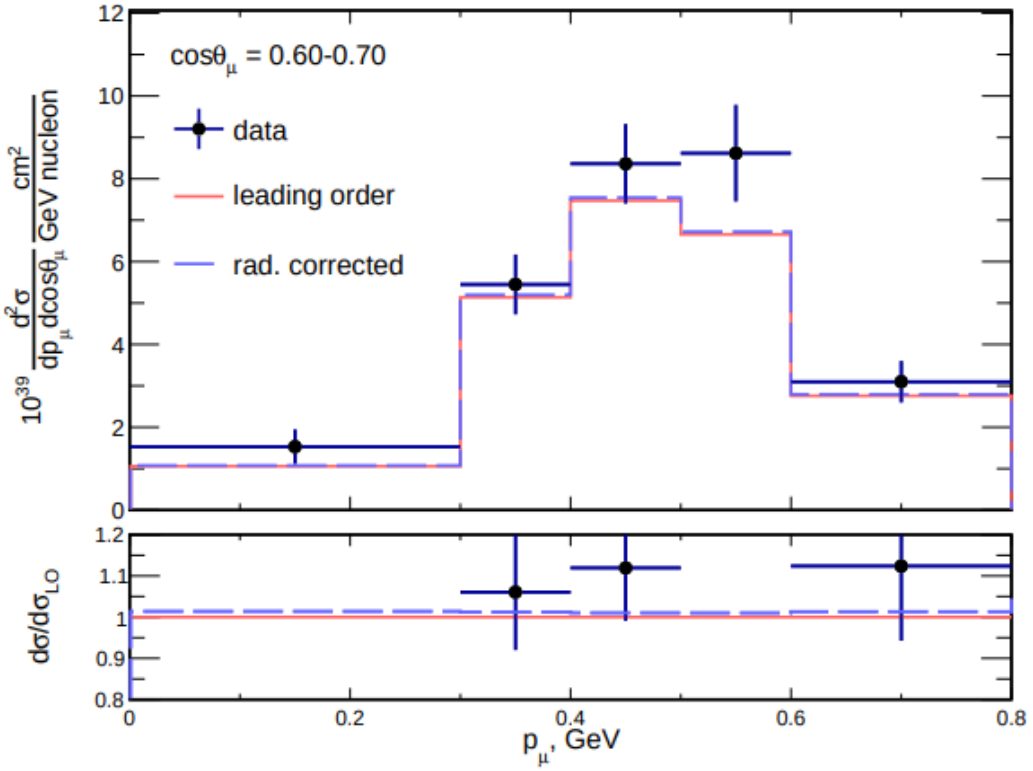
- Studies of low-ν method using multiple generators (C. Wilkinson et al. Eur.Phys.J.C 82 (2022) 9, 808)

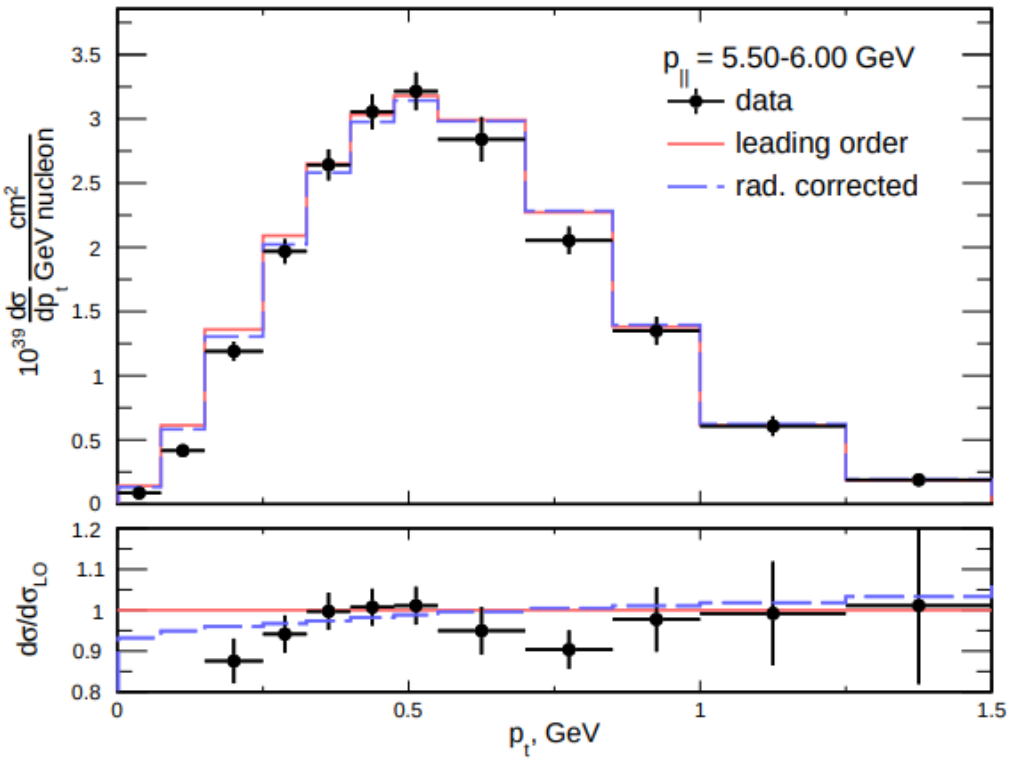




Recent examples

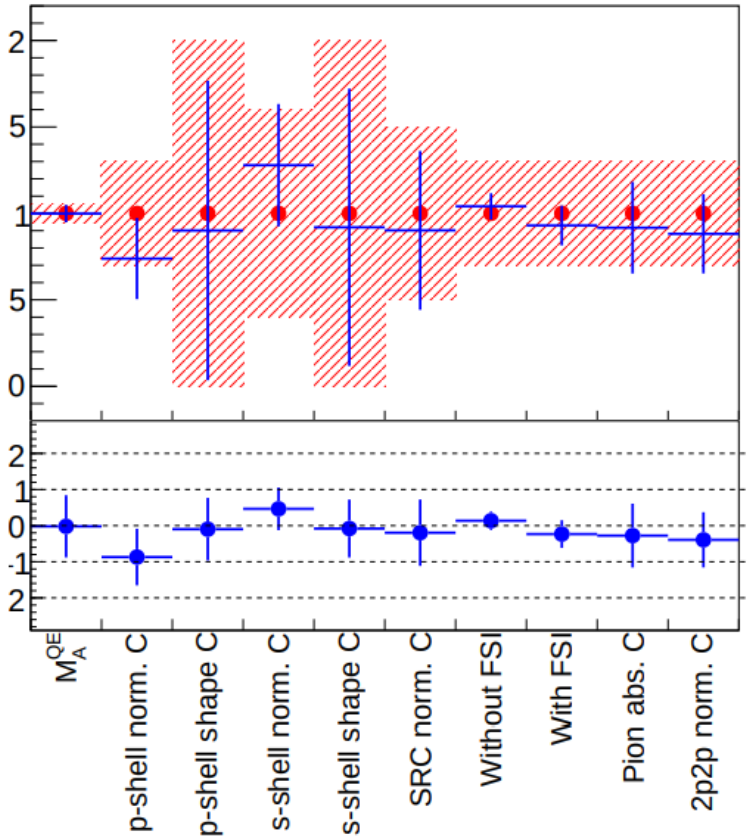
- Radiative corrections (O. Tomalak et al., Phys. Rev. D 106, 093006)
  - Found large effect for MINERvA, smaller effect for T2K
  - Implemented in NUISANCE; **<u>you can test it too!</u>**

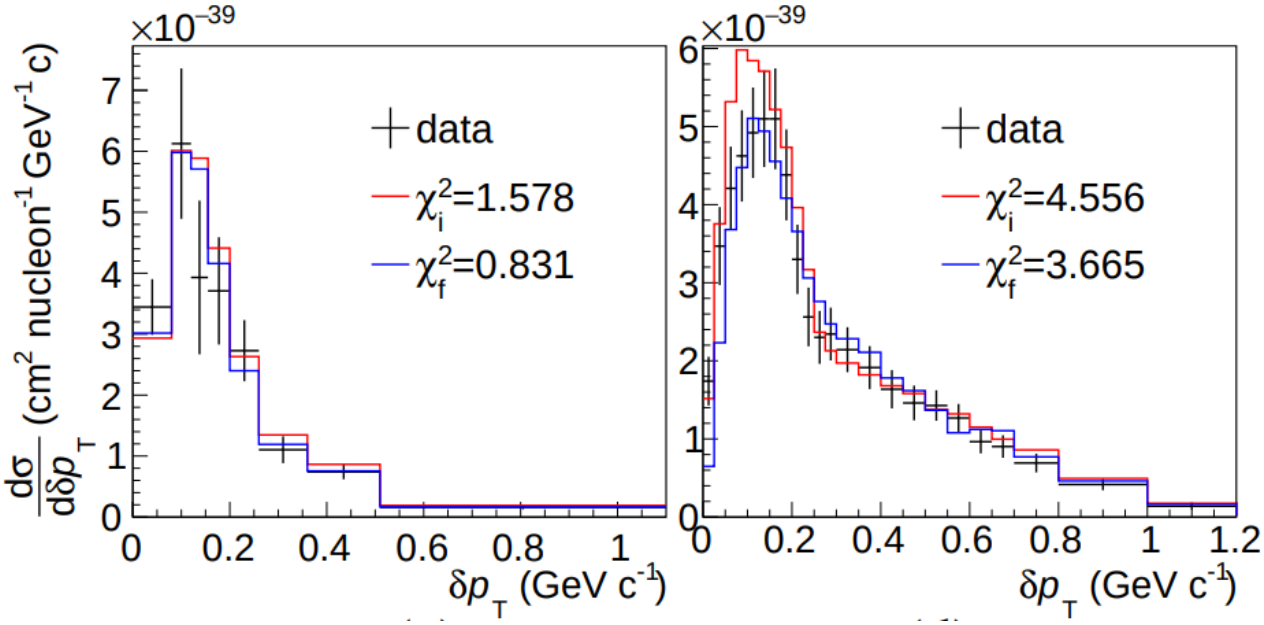




Recent examples

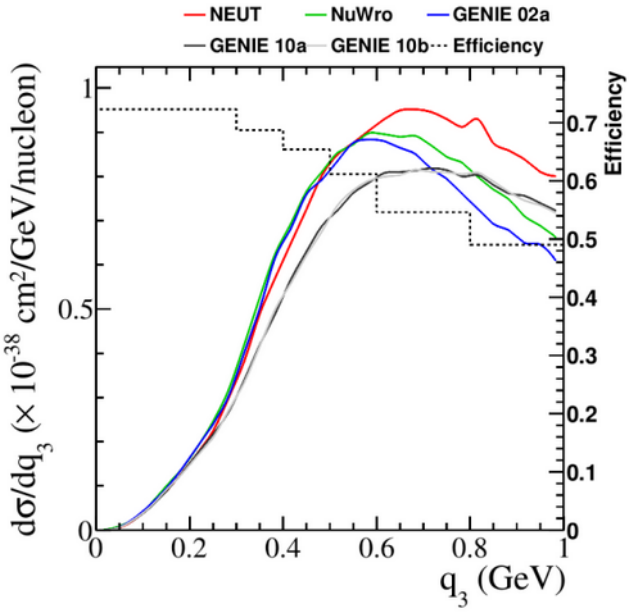
- ND280 Upgrade sensitivity studies and development of T2K interaction model (J. Chakrani et al. arXiv:2202.03219 [hep-ph])
  - Fit multiple T2K model parameters against published T2K and MINERvA data

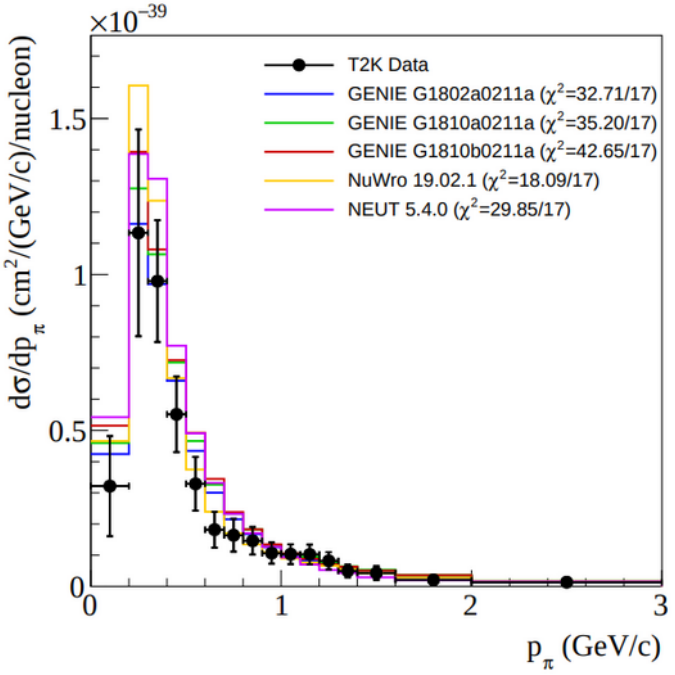


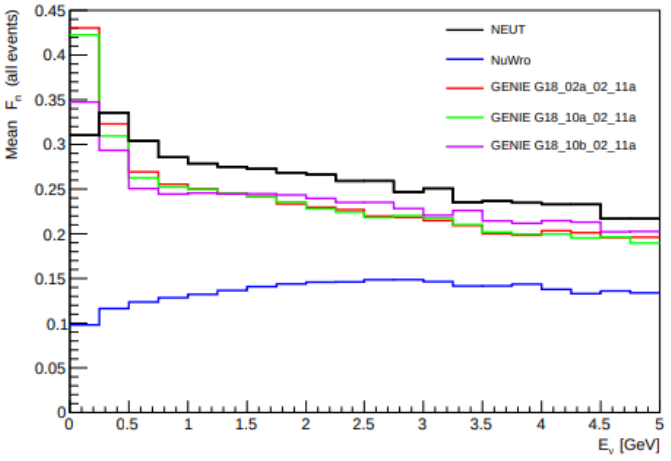


Recent examples

- Pittsburgh tensions workshop (M. Buizza Avanzini et al., Phys.Rev.D 105 (2022) 9, 092004)
  - Aimed to get experiment and generator experts together to understand model dependence and current experimental data (amongst others!)
  - Used multiple generators to form predictions against data, against efficiency curves, and how much energy carried away by neutral particles

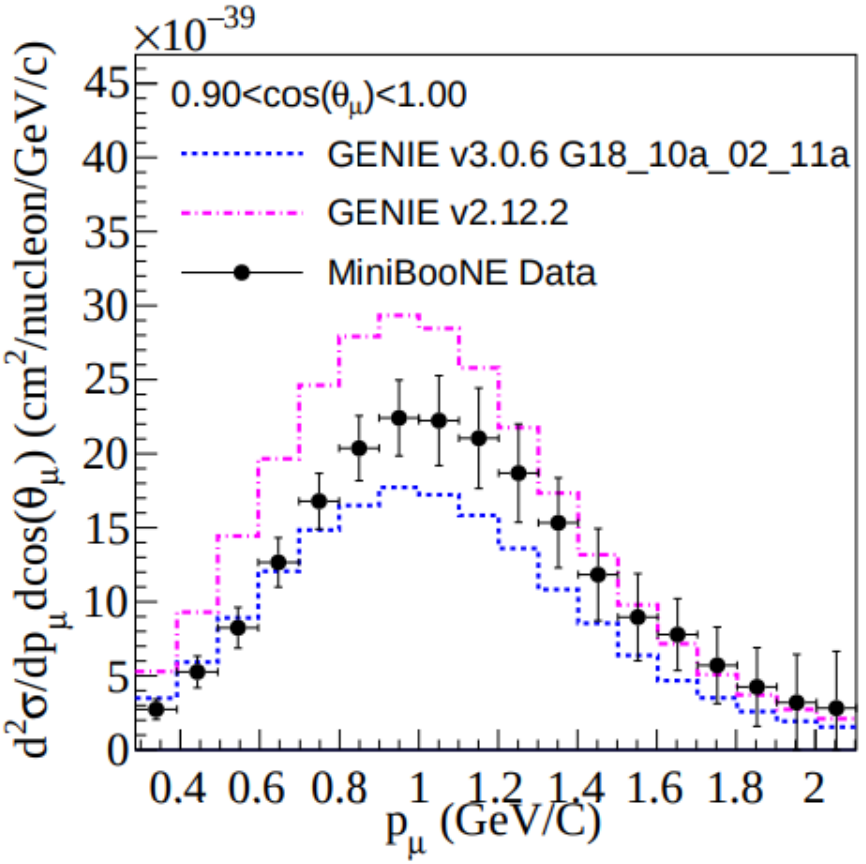


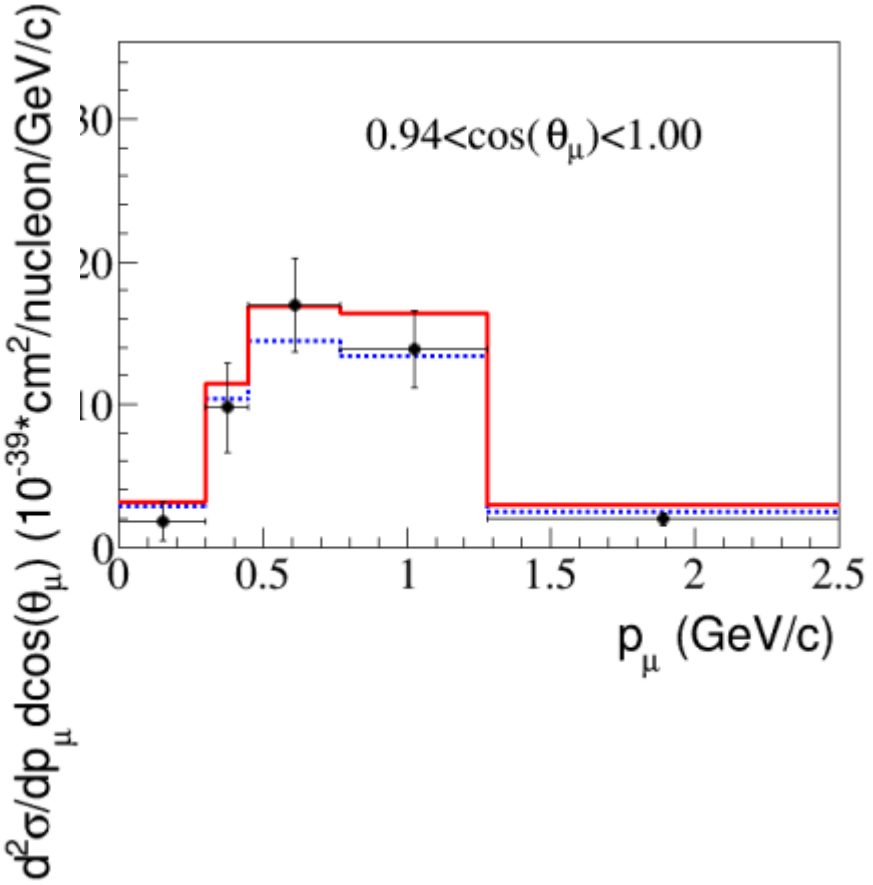




Recent examples

- MicroBooNE CC0π model tuning (P Abratenko et al. Phys.Rev.D 105 (2022) 7, 072001)

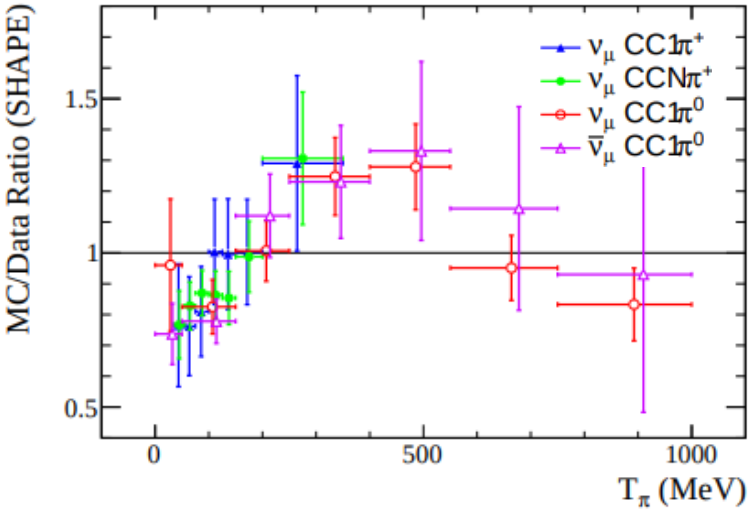


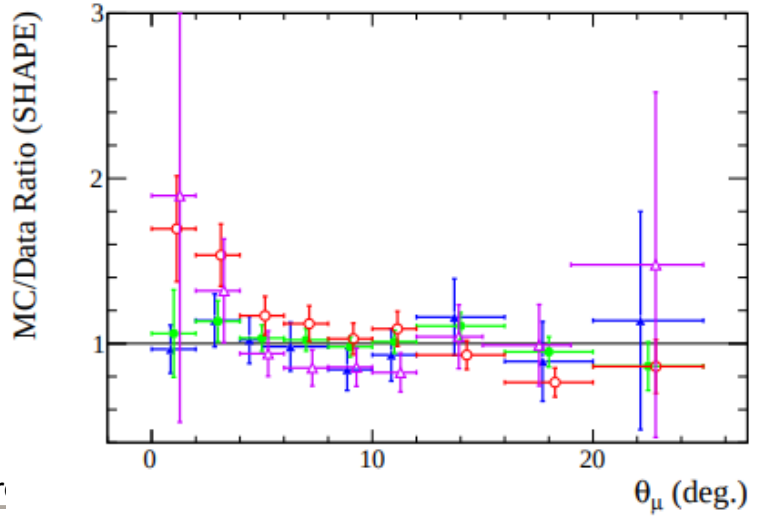


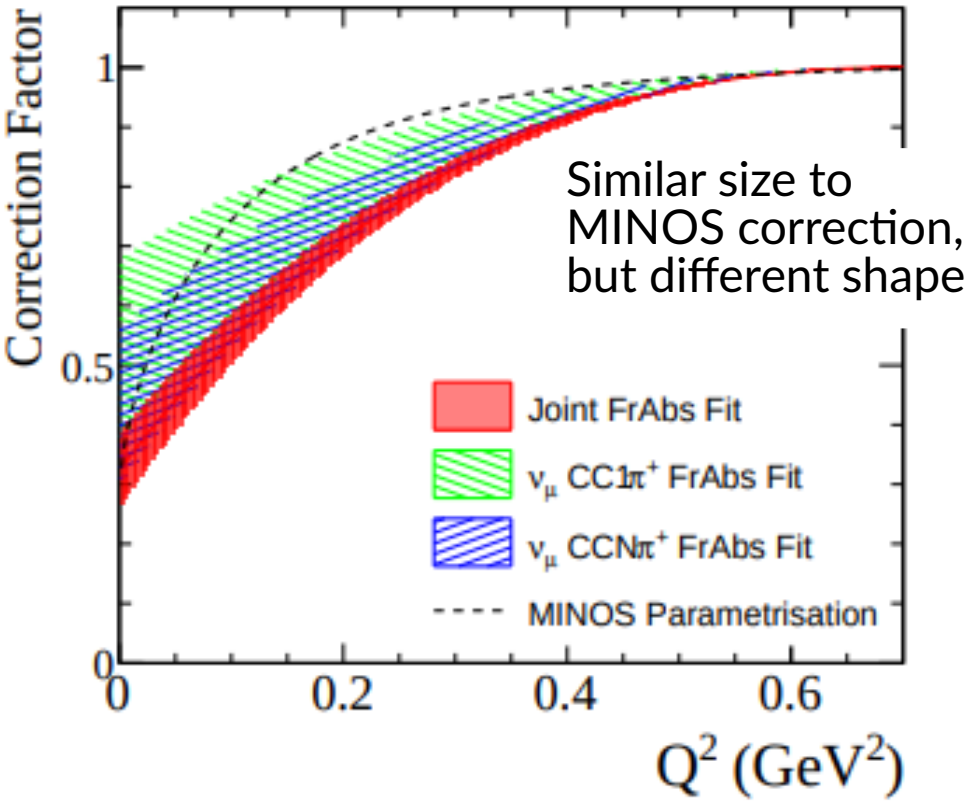
  - Tuned CCQE and 2p2h model to T2K CC0π to estimate cross-section uncertainties going into sterile oscillation ananlysis

Recent examples

- MINERvA single pion tune (P. Stowell et al., Phys.Rev.D 100 (2019) 7, 072005)
  - Used publicly available CC pion data from MINERvA to develop a low Q^2 suppression for GENIE v2

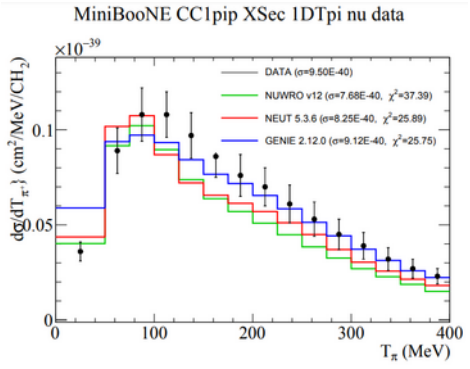


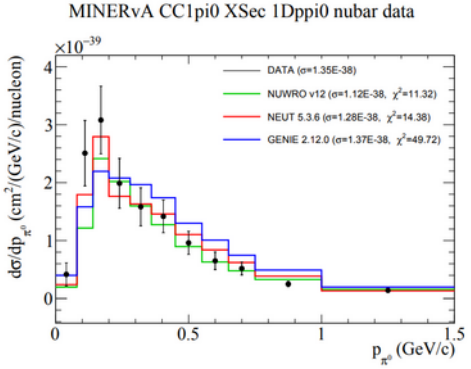


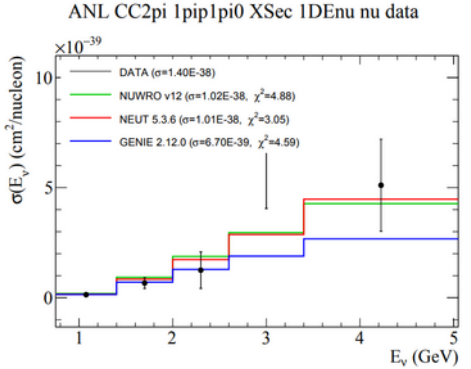


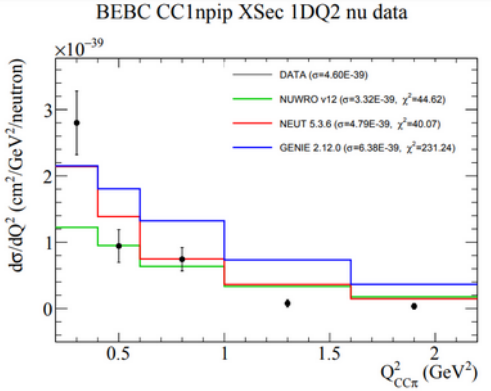
Future

- Continue implementing measurements as they are released
- Support the generators as they expand
- Produce easy-to-check generator comparisons for community



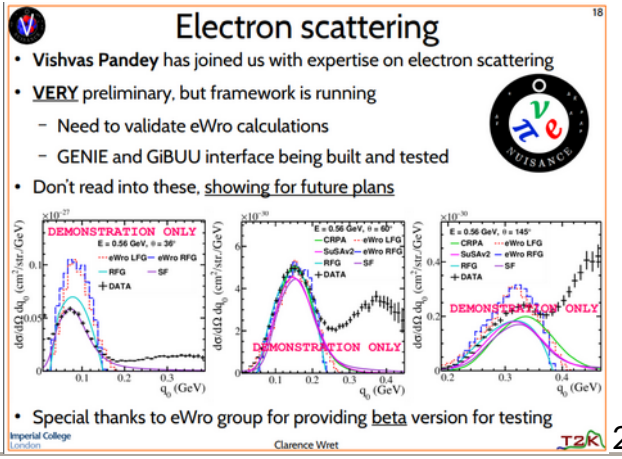






- Officialise a set of containers to further simplify user experience
  - Possibly also include large MC samples if this is of interest
- Work on expanding NUISANCE for electron and pion scattering has started and was working, but needs some love

NuInt 2017



UNIVERSITY OF OXFORD

Summary

- NUISANCE compares neutrino interaction generators to themselves and external data
- Open source tool (GPLv3) developed for the community, and open to collaboration and use
- Interfaces to generator reweighting libraries and supports its own customs reweighting machinery
- Interfaces to minimisation routines (e.g. Minuit, MCMC) to fit models to data or mock data
- Builds uncertainty bands against selected data or generators
- (Hopefully) making your voyage across the seas of neutrino interaction uncertainties a little smoother!

Thanks

nuisance-xsec.slack.com