PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB

RECEIVED: *December 23, 2016*
ACCEPTED: *January 3, 2017*
PUBLISHED: *January 19, 2017*

NUISANCE: a neutrino cross-section generator tuning and comparison framework

2017 JINST 12 P01016

The NUISANCE collaboration

P. Stowell,[a,1] C. Wret,[b] C. Wilkinson,[c] L. Pickering,[b] S. Cartwright,[a] Y. Hayato,[d,e] K. Mahn,[f] K.S. McFarland,[g] J. Sobczyk,[h] R. Terri,[i] L. Thompson,[a] M.O. Wascko[b] and Y. Uchida[b]

[a] *University of Sheffield, Department of Physics and Astronomy, Sheffield, U.K.*
[b] *Imperial College London, Department of Physics, London, U.K.*
[c] *University of Bern, Albert Einstein Center for Fundamental Physics,*
Laboratory for High Energy Physics (LHEP), Bern, Switzerland
[d] *Kavli Institute for the Physics and Mathematics of the Universe (WPI),*
Todai Institutes for Advanced Study, University of Tokyo, Kashiwa, Chiba, Japan
[e] *University of Tokyo, Institute for Cosmic Ray Research, Kamioka Observatory, Kamioka, Japan*
[f] *Michigan State University, Department of Physics and Astronomy, East Lansing, Michigan, U.S.A*
[g] *University of Rochester, Department of Physics and Astronomy, Rochester, New York, U.S.A.*
[h] *University of Wrocław, Institute of Theoretical Physics, Wrocław, Poland*
[i] *Queen Mary University of London, School of Physics and Astronomy, London, U.K.*

E-mail: nuisance@projects.hepforge.org

ABSTRACT: NUISANCE is an open source C++ framework which facilitates detailed studies of neutrino interaction cross-section models implemented in Monte Carlo neutrino event generators. It provides a host of automated methods to perform comparisons of multiple generators to published cross-section measurements and each other. External reweighting libraries are used to allow the end-user to evaluate the impact of model parameters variations in the generators with data, or to tune the generator predictions to arbitrary dataset combinations. The design is modular and focusses on ease-of-use to allow new datasets and more generators to be added without requiring detailed understanding of the entire NUISANCE package. We discuss the motivation for the NUISANCE framework and suggested usage cases, alongside a description of its core structure.

KEYWORDS: Software architectures (event data models, frameworks and databases); Analysis and statistical methods; Simulation methods and programs; Data processing methods

ARXIV EPRINT: 1612.07393

[1] Corresponding author.

© 2017 IOP Publishing Ltd and Sissa Medialab srl

doi:10.1088/1748-0221/12/01/P01016

Contents

1 Introduction

Current and future neutrino oscillation experiments have stringent systematic error budgets, which are considerably smaller than are currently achievable. In particular, the uncertainties associated with neutrino interaction cross-section models need to be reduced down to the few percent level to deliver the required sensitivity; required cross-section uncertainties of 4% and 2% have been projected for T2K-II and DUNE respectively [1, 2]. Long baseline oscillation experiments spanning the $0.1 \leq E_\nu \leq 10\,\mathrm{GeV}$ range suffer especially, as at these energies a consistent theoretical interaction picture has yet to emerge [3–5]. Selecting default interaction models from those available and estimating parameter uncertainties are significant challenges currently facing neutrino oscillation and cross-section experiments.

Two main issues complicate the problem of building a consistent neutrino cross-section model when using *nuclear targets*. Firstly, the interaction-level variables which cross-section models are constructed in terms of (e.g., energy and momentum transfer, neutrino energy) cannot be directly measured by experiments. The incoming neutrino four-momentum is not known on an event-by-event basis from the beam, nor can it be reconstructed accurately by using final-state particle kinematics without relying on the experiment's model for nuclear effects, such as initial state nucleon model and particle propagation. The only measurable *model-independent* quantities are the outgoing particle kinematics (e.g., outgoing muon momentum and direction). Secondly, Final State Interactions (FSI)—where the particles leaving the interaction vertex re-interact before leaving the nucleus — can modify the outgoing particle kinematics and event particle content. It is not possible to separate a single interaction process with selection cuts: a simple $\nu_\mu + n \rightarrow \mu^- + p$ Charged-Current

Quasi-Elastic (CCQE) interaction cannot be clearly distinguished from a $\nu_\mu + p \rightarrow \mu^- + \pi^+ + p$ interaction if the pion is absorbed in the nucleus. As a result, model-independent measurements must, in general, describe a final-state *topological* cross section rather than a single interaction mode cross section — such as measuring events with one muon and no pions in the final state (CC0π interactions) in lieu of CCQE interactions.

A number of general purpose neutrino interaction Monte Carlo (MC) event generators are available, simulating a large range of interactions. These make it possible to produce realistic predictions for topological cross-section measurements, and allow the user to modify model parameters and combine different models. Whilst model-independent measurements are essential for arriving at a well-motivated cross-section model with defensible uncertainties, topological cross sections given in terms of final-state particle kinematics provide relatively weak constraints of cross-section model parameters which often have most strength in interaction variables such as four-momentum transfer. This is complicated by the fact that different detector sensitivities and kinematic thresholds mean experiments may probe vastly different regions of phase-space for an observed interaction channel. Therefore model parameters extrapolated from one experiment may not be sufficient to describe all other experiments. Hence it is essential to use data from many experiments, with different energies, target materials and detector designs to constrain a full cross-section model and claim confidence in it.

NUISANCE is a software package written to simplify the task of confronting and comparing neutrino event generators with each other and published world cross-section data. It is an open source C++ package distributed under the GPLv3 license agreement [6]. NUISANCE is the primary tool for evaluating and constraining the cross-section model used in T2K analyses [7] using external scattering data, and grew out of efforts to tune the NEUT interaction model within the T2K Neutrino Interactions Working Group. The main advantage of this framework is its modularity: new datasets can be included with ease by adding "measurement" classes which converts any supported generator's output to a cross-section and compares it to data, without requiring the user to understand the output formats of the generators. Similarly, new generators can be added without requiring detailed understanding of the entire NUISANCE framework. The only dependency of NUISANCE outside the chosen generator(s) themselves is the ROOT library [8].

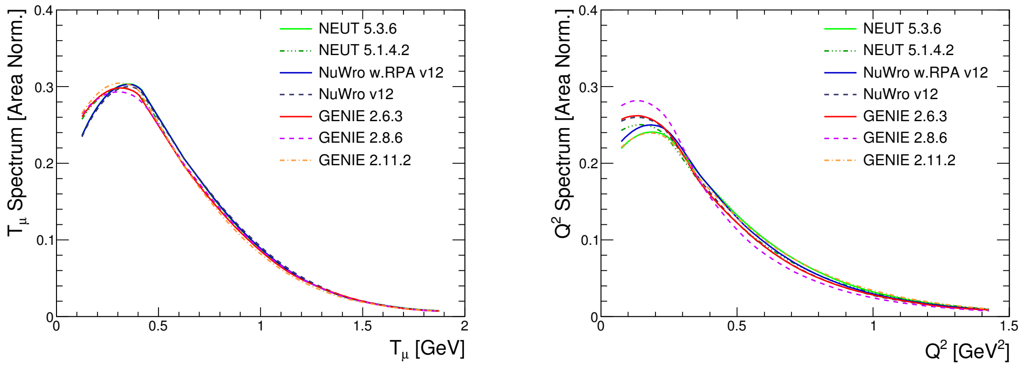
In this paper, we describe the core structure of NUISANCE, give the scope of the supported features, and demonstrate different usage scenarios. Detailed documentation of included datasets, validation plots, and usage instructions with examples can be found at `nuisance.hepforge.org`.

2 NUISANCE

This section gives an overview of the core structure and design principles behind the NUISANCE framework. Full support for the standard output of the GENIE [9, 10], NEUT [11], NuWro [12] and GiBUU [13] neutrino event generators is provided, with limited support for NUANCE[1] [14]. The core structure is designed to be easily extended, with support for different event generators possible in later versions (e.g. neutrino, electron and pion scattering simulations).

[1] Only shape comparisons are possible with NUANCE because of limitations in the generator output.

2017 JINST 12 P01016



(a) Muon kinetic energy.

(b) Four-momentum transfer.

Figure 1. Shape comparison of CC-inclusive ν_μ-CH$_2$ events generated in different versions of the NEUT, NuWro, and GENIE generators using the MiniBooNE neutrino-mode flux [15] (shown in figure 2).

2.1 Input handling

Each event generator has a different output format and event structure, but the underlying content is the same, always including:

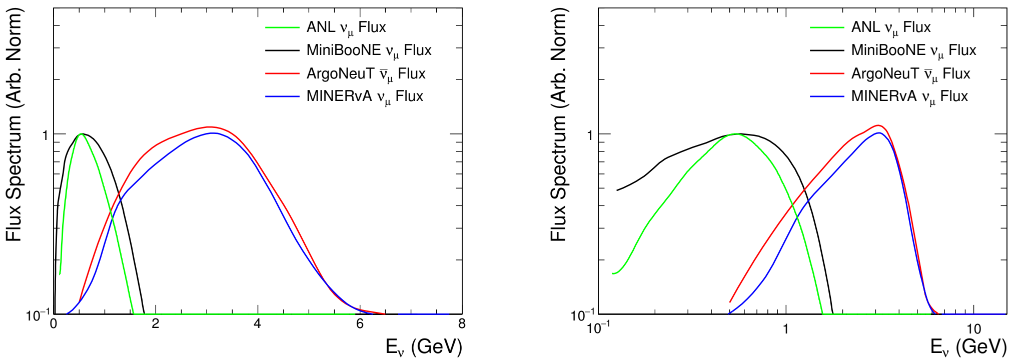
- a list of incoming/outgoing particles, with their four-momentum, PDG code, and status;
- an underlying interaction mode used in the generation,[2] e.g. $\bar{\nu}_\mu$–^{12}C CCQE, ν_e–^{16}O CC1π^+;
- a method for normalising the event distribution to produce a differential cross section;
- (optional) information to support event reweighting, described in section 2.4.

To ensure consistency between generators, and to increase speed, NUISANCE uses a reduced event structure that contains only the information required and unifies the format for the various generators. The conversion to the standardised format is performed by the *InputHandler* class when an event is first used and all subsequent NUISANCE functionality uses this format. The structure provides access to information about the event using common caller functions which are unified for the generators. It also ensures compatibility were new generators to be added to the *InputHandler* in the future. Figure 1 illustrates a simple comparison of CC-inclusive ν_μ-CH$_2$ events generated with a variety of generators using the MiniBooNE neutrino-mode flux shown in figure 2 [15].

2.2 Cross section normalisation

The NUISANCE *InputHandler* calculates all the information needed to weight events correctly such that the final distribution is normalised to an inclusive cross-section prediction. In the case of GiBUU, these weights are calculated by the generator itself and saved with the event. For the other generators — where the number of events from different interaction channels are generated in proportion to their cross section — a single weight is calculated which is applied to all events.

[2] The definition of these channels may vary between generators. Since most measurements are topology based, this is not a problem.



(a) Linear energy scale.

(b) Log energy scale.

Figure 2. Flux distributions used to generated events for NUISANCE comparisons in this work [16–19], also supplied with NUISANCE.

NUISANCE requires the flux distribution used to generate the sample, $\phi(E_\nu)$, and calculates the predicted total event rate per target nucleon, R, from information available in the generator output file,

$$R = \int \sigma_{\text{tot}}(E_\nu) \times \phi(E_\nu) dE_\nu, \tag{2.1}$$

where $\sigma_{\text{tot}}(E_\nu)$ is the *total inclusive cross section* as a function of neutrino energy and the integral runs over the entire energy range the events were generated in.[3] R is provided in the default output of the NuWro and NEUT generators, but must be calculated for GENIE from the event record. A separate application, *PrepareGENIE*, is supplied to reconstruct the predicted GENIE cross section as a function of neutrino energy for each interaction channel. These cross-section “splines” are then used to predict R for the event sample given the input flux.

A final *flux-averaged* cross-section weight, W, can then be calculated for each generator

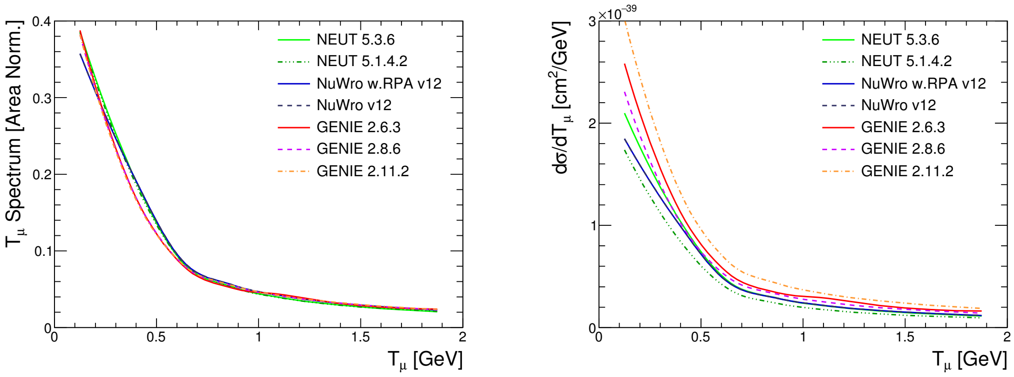
$$W = \frac{R}{N\Phi}, \tag{2.2}$$

where N is the total number of events generated in the generator, and Φ is the integrated neutrino flux between E_ν^{min} and E_ν^{max} (the neutrino energy limits in the signal definition)

$$\Phi = \int_{E_\nu^{\text{min}}}^{E_\nu^{\text{max}}} \phi(E_\nu) dE_\nu. \tag{2.3}$$

Filling a histogram in interaction variable x with the weights W, for events that pass a user-supplied signal definition produces a correctly normalised *flux-averaged* cross section $d\sigma(x)$. Dividing by each bin’s width produces the differential cross section $d\sigma(x)/dx$, shown in figure 3. This is only appropriate when comparing to *flux-averaged* cross-section results. For a *flux-integrated* cross section the flux is instead integrated out on a bin-by-bin basis and the E_ν^{min}, E_ν^{max} limits in 2.3 are instead given by the bin-edges of the relevant E_ν bin that an event resides in.

[3] Even if the signal definition contains a cut on E_ν.



(a) Raw event rate.

(b) Cross section.

Figure 3. Raw event rate and correctly normalised cross-section distributions as a function of muon kinetic energy, T_{μ}, shown for ν_{μ}-CH$_2$ CC1π^{+} events produced using a variety of generators and the MiniBooNE neutrino-mode flux (shown in figure 2).

2.3 Comparisons to cross-section data

To compare different models to a chosen neutrino cross-section dataset a model prediction must be produced that matches the original data analysis, matching true signal and binning definitions. To add a new dataset to NUISANCE a new "measurement" class must be created which defines the analysis method used to turn a set of generated MC events into a matching model prediction. To minimise the work required for a user to add a new dataset, these classes inherit from a small number of "measurement" base classes which define methods common for all cross-section predictions of a certain type (e.g. one-dimensional, two-dimensional). Comparisons can be added into the framework provided the following information is known:

1. *Data distribution:* the measured central values and uncertainties must be supplied in either text or ROOT file format.

2. *Signal definition:* a signal which acts to select events using the particle list must be defined. Utility definitions are available for common signal definitions, e.g. CC0π.

3. *Binning definition:* the kinematic variables to plot must be defined from incoming/outgoing particle list, e.g. lepton momentum. The binning itself is copied from the data distribution.

4. *(Optional) Covariance:* the correlations between each point in the data distribution so that a more accurate likelihood can be formed. If no covariance is provided uncorrelated errors are assumed on each point.

5. *(Optional) Smearing matrix:* a translation matrix to smear true variable distributions, converting them into detector reconstructed variable distributions that can be compared directly to reconstructed data releases.

Every sample has an event loop in the base class which iterates over all events given an input file provided at runtime. The input file to NUISANCE is the output of the generator(s) of interest.[4]

[4] Which has to be pre-processed when using GENIE with the *PrepareGENIE* utility.

Only events which pass the signal definition are retained past the first event loop. For signal events, a cross-section weight is calculated using Equation 2.2 (assuming a flux-averaged cross section), and all histograms in the specific measurement class are filled. This automated event loop ensures that the core handling of event inputs remains the same for every measurement implementation class, although each method can be overloaded if necessary.

An *event manager* can be turned on to avoid iterating over events in the same input file multiple times if two or more measurement classes use the same generator output file. When it is used, the event manager checks whether the signal criteria are met in any of the classes, and retains events which are signal for one or more of them. This can significantly speed up NUISANCE for many of the fitting routines, in which weights need to be recalculated and histograms refilled multiple times for a number of datasets, e.g. comparing multiple kinematic distributions of the same measurement, or different measurements using the same flux.

The most basic measurement implementation class produces a correctly normalised histogram with the same binning as the data and can be compared directly. ROOT histograms showing the data, MC prediction(s) and the input flux(es) are saved in the output file for later comparison. Various utility functions exist to include histograms, e.g. stacking the MC prediction by interaction mode or particle type, shape predictions and data-MC ratios, as in figure 5. It is a trivial exercise to include any additional histograms by overloading the base-class functions.

The data-MC agreement is evaluated by a likelihood which is saved in the output file. The base class defaults to using a covariance matrix if supplied, or reverts to a Gaussian pdf for cross-section measurements and a Poisson pdf for event-rate measurements.

$$\text{(Covariance)} \quad -2LL = \sum_{ij} (\nu_i^{\text{data}} - \nu_i^{\text{MC}})(M^{-1})_{ij}(\nu_j^{\text{data}} - \nu_j^{\text{MC}}) \tag{2.4}$$

$$\text{(Gaussian pdf)} \quad -2LL = \sum_{i} \left(\frac{\nu_i^{\text{data}} - \nu_i^{\text{MC}}}{\sigma_i^{\text{data}}} \right)^2 \tag{2.5}$$

$$\text{(Possion pdf)} \quad -2LL = 2 \sum_{i} \nu_i^{\text{MC}} - \nu_i^{\text{data}} + \nu_i^{\text{data}} log \left(\frac{\nu_i^{\text{data}}}{\nu_i^{\text{MC}}} \right) \tag{2.6}$$

where ν_i is the bin content in i-th bin for data or MC, M is the supplied covariance matrix, and σ_i is the error on the i-th bin in data.

These likelihood functions can easily be overloaded by the user for each measurement class to allow more complex likelihoods to be used for a given analysis such as shape-only and floating normalisation likelihoods.

2.4 Event reweighting

Event reweighting allows MC predictions to be modified after event generation by separating out parts of the cross section that can be recalculated without having to perform the entire MC simulation again. This saves considerable computation time and is useful for both model tuning and the evaluation of model systematic uncertainties since the events are already generated.

NUISANCE has native support for the NEUT, NuWro [20], and GENIE event reweighting libraries. A NUISANCE reweight wrapper is provided which can read in the custom event format and return an event weight for any given parameter set. All reweightable parameters provided by

2017 JINST 12 P01016

the external libraries can be used to calculate new event weights. This allows the user to easily modify the NUISANCE prediction in the various fitting and validation routines, whilst requiring only minimal knowledge of the individual generators reweight engines. The reweighting parameters are specified in the user-supplied card files.

3 NUISANCE applications

A number of different applications are available with the NUISANCE framework, all of which are controlled by card files. In this section, the main NUISANCE applications are introduced and a general overview of their functionality is given. Detailed information on the input format required and example card files are available at `nuisance.hepforge.org`. Additionally, the general behaviour of NUISANCE can be controlled by a configuration file, which can be overridden for specific comparisons if desired either in the card file or with command line arguments.

3.1 Simple data-MC comparisons: *nuiscomp*

The simplest usage case for NUISANCE is to produce an MC prediction for one or multiple measurement(s). The *nuiscomp* application accepts a simple card file with a list of datasets to produce comparisons for and the input file to generate them with. It saves the resulting histograms to a single ROOT output file. Optionally, the user can specify any reweightable parameters which should be set when making this comparison. The default behaviour is to not reweight the prediction.

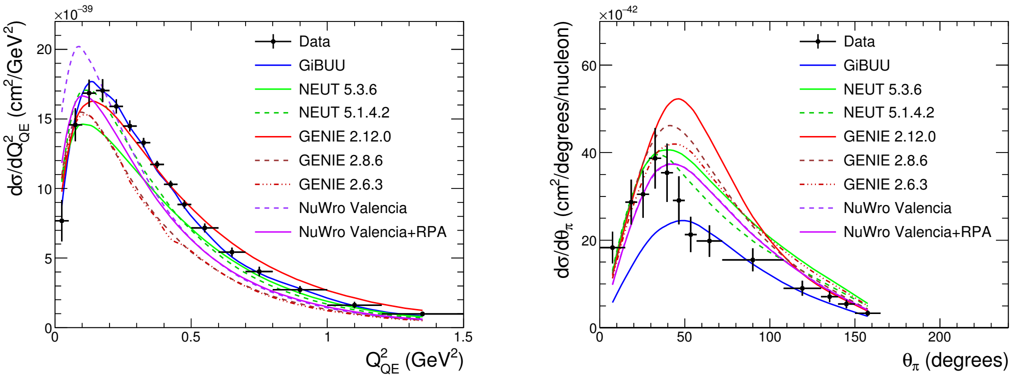
As described in section 2, different input types are automatically handled by NUISANCE. Changing generators only requires changing the input type and the file location, both specified by the user in the card file. Multiple generator, multiple dataset comparisons are straightforward with NUISANCE, as illustrated in figure 4.

Alongside the data and MC histograms, the *nuiscomp* application saves a number of auxiliary MC histograms to help evaluate where there are tensions between the data and models. Examples include predictions separated by true interaction modes, and shape-only comparisons, shown in figure 5.

3.2 Raw generator comparisons: *nuisflat*

The *nuisflat* application is intended to compare generators to each other, rather than to data. A template class converts the generator events into a flat ROOT tree containing particle information for each event, what signal definitions the event passes, its interaction mode, amongst a host of other event variables. Additional quantities can be added to the tree for tailored studies including an option to save the entire NUISANCE event for access to the full input/output particle stack. Once the output has been produce by *nuisflat* it loses all dependencies on the generator libraries. Analysis is only dependent on ROOT to inspect the tree and its contents; the application produces consistent generator comparisons with minimal knowledge of the individual generator.

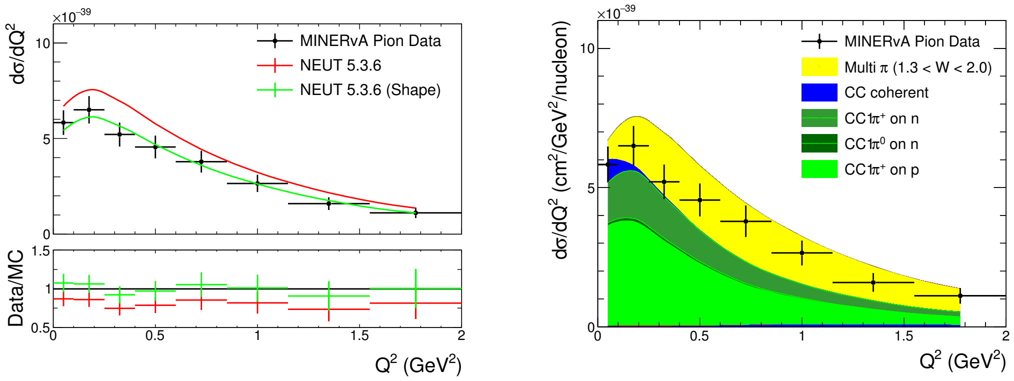
As with all NUISANCE applications, *nuisflat* supports reweighting the generated events to parameter variations specified in the input card file by the user at runtime.



(a) MiniBooNE ν-CH$_2$ CCQE.

(b) MINERvA ν-CH CC1$\pi^{\pm}$.

Figure 4. Comparisons of different generators and their versions to published measurements [17, 18]. The fluxes used to produce these distributions are shown in figure 2.



(a) Rate and shape-only comparisons to data.

(b) MC broken down by true interaction mode.

Figure 5. NEUT prediction for MINERvA CC$N\pi^{\pm}$ [21] produced with the *nuiscomp* application. The fluxes used to produce these distributions are shown in figure 2.

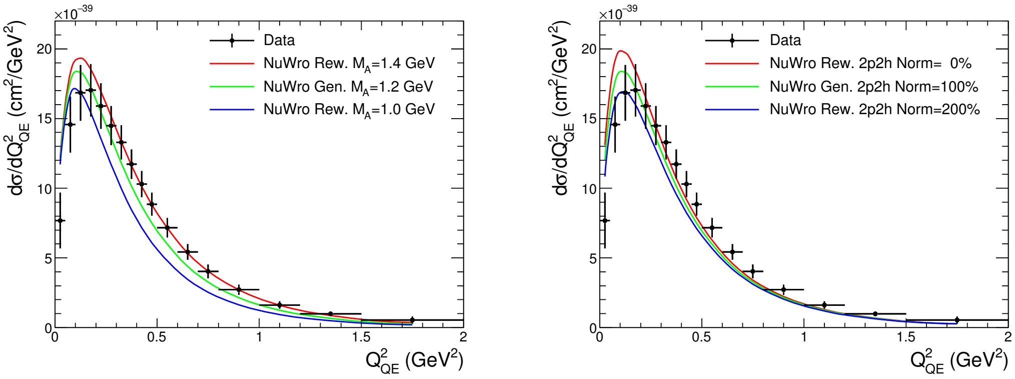
3.3 Systematic validation studies: *nuissyst*

The *nuissyst* application can be used to study the effects of cross-section systematics on user-specified distributions in a number of ways, provided that the generators have reweighting libraries enabled.[5] It can step through a range of values for a reweightable parameter[6] and validate reweighting implementations. It can also compare each generator's implementation of the reweighting engines — e.g. the effect of varying M_A^{CCQE} by 0.1σ in GENIE versus the same variation in NuWro. Examples of the output of this utility can be found in figure 6.

The *nuissyst* application can also make throws of any number of reweightable parameters to build up an error band on a generator prediction across any combination of datasets. The central value and 1σ uncertainty can be defined by the user, and Gaussian throws will be made around that

[5] Possible in GENIE, NEUT and NuWro.

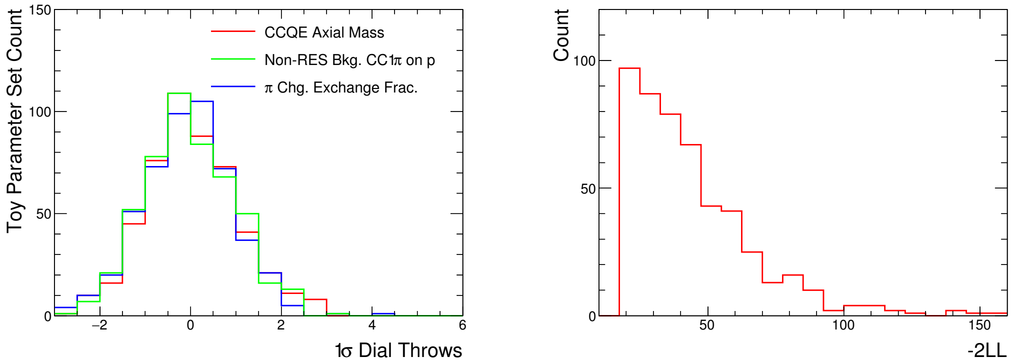
[6] If a parameter is reweightable by a generator's reweighting library it is supported in NUISANCE.



(a) M_A variations.

(b) 2p2h variations.

Figure 6. Reweighting responses for a set of NuWro events compared to MiniBooNE CCQE neutrino data.



(a) Thrown parameter values.

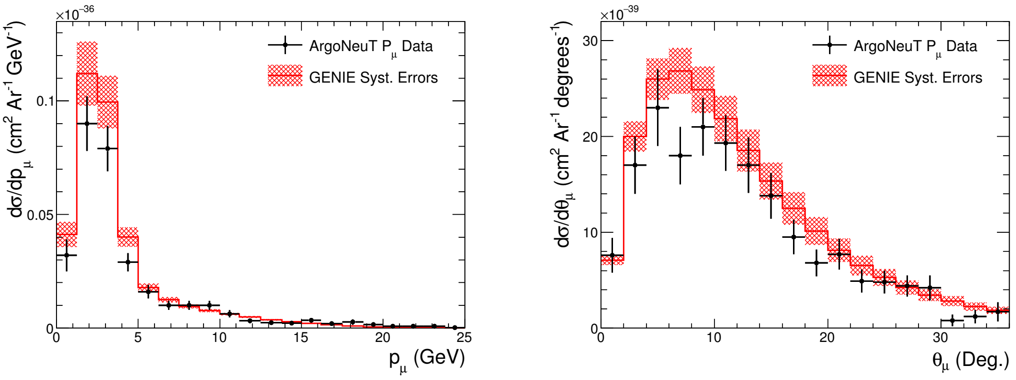
(b) Throws Likelihood distribution.

Figure 7. Examples of thrown parameter values and the likelihood results for 500 different toy datasets generated with GENIE v2.12.0 for comparison to the ArgoNeut $\bar{\nu}$–^{40}Ar CC-inclusive dataset [19].

value with the defined width. Flat parameter throws are also supported, and a method is included for throwing parameters according to a user-supplied parameter covariance matrix. In the latter case the distribution of parameter values for each bin of the requested distributions is used to produce a error band for that bin as shown in figure 7.

A realistic use case for *nuissyst* is shown in figure 8, where 500 throws of the default GENIE v2.12.0 cross-section uncertainties have been used to build up a 1σ error band for the ArgoNeuT $\bar{\nu}$–^{40}Ar CC-inclusive dataset [19]. This functionality enables the user to investigate whether a supplied generator cross-section uncertainty agrees with any particular dataset and aids in robust parameter error inflation studies. A histogram is saved in the output file which shows the level of data-MC agreement for all datasets included in the comparison. In such a case, a χ^2 statistic is calculated as shown in figure 7.

2017 JINST 12 P01016



(a) Muon momentum (p_μ).

(b) Muon angle w.r.t. incoming beam direction (θ_μ).

Figure 8. Default GENIE v2.12.0 prediction for the ArgoNeut $\bar{\nu}_\mu$–^{40}Ar CC-inclusive distributions [19], with error bands produced with 500 throws of all GENIE reweighting parameters according to their nominal uncertainties.

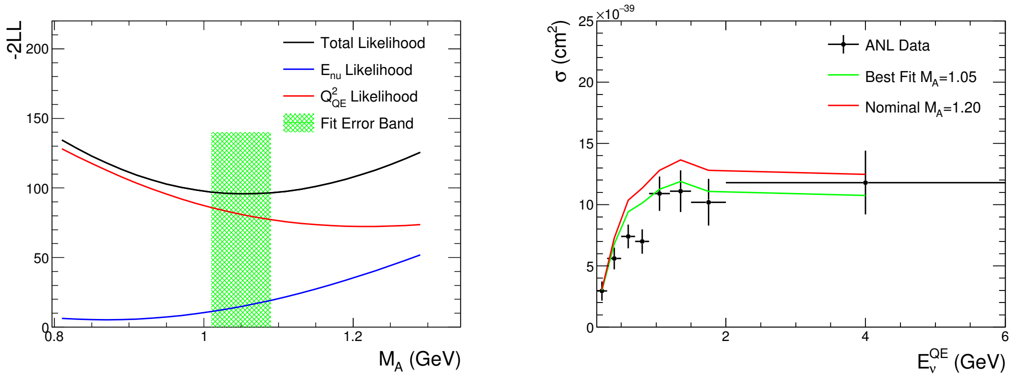
3.4 Parameter fitting: *nuismin*

NUISANCE was originally designed to compare and tune the NEUT generator predictions to external datasets to provide cross-section uncertainties for T2K analyses. The *nuismin* application fits any number of reweightable parameters to any combination of measurements, and uses ROOT minimisation libraries to minimise the test-statistic with respect to the parameters specified. ROOT provides multiple linear and non-linear scan methods which can be chosen at runtime. By default, the MIGRAD steepest gradient descent algorithm from the MINUIT package is used [22]. It is also possible to modify the test-statistic for each dataset used in the minimisation by overloading the base class function. The fit parameters are specified in the card file, and it is possible to fix parameters, set fit boundaries and define starting values.

At each iteration of the fit, NUISANCE recalculates weights on an event-by-event basis using the relevant generator's reweighting libraries with the parameter variations requested by the minimisation algorithm. Weighted histograms are filled for all specified samples and the new test-statistic for the reweighted prediction is calculated and used to inform the minimisation algorithm. The output includes the nominal and best fit histograms, information about the best fit parameters and correlations between them, and basic information about the fit, such as the best fit χ^2 and the number of iterations. Parameter error estimation is determined by the minimiser in ROOT. An example of a simple fit procedure is shown in figure 9, where the NEUT CCQE model is fit simultaneously to ANL CCQE $\sigma(E_\nu)$ and $N_{\text{evt}}(Q^2)$ data [16]. The joint likelihood is the sum of the likelihoods provided by each of the samples, which are treated independently by NUISANCE. The nominal and best fit distributions are also shown, and the error on the fitted parameter is indicated.

Penalty terms on parameters can also be introduced in *nuismin*. The penalties can be applied with a correlation by supplying a covariance matrix. The output parameter covariance matrix of a previous *nuismin* fit is also supported as a penalty term in a subsequent fit. For example, figure 10 shows a fit to MINERvA ν_μ-CH CCQE data [23] where the only free parameter is the axial mass, M_A. In this fit, the result of the free-nucleon fit to ANL data shown in figure 9 has been used as a

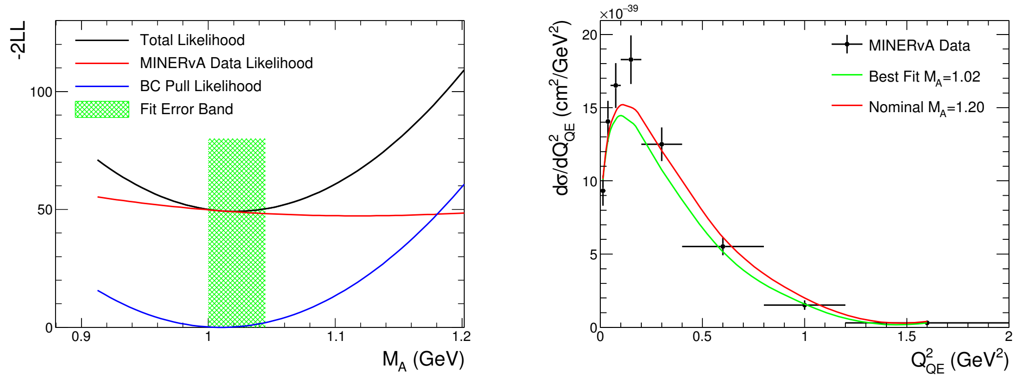
2017 JINST 12 P01016



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

(b) Nominal and best fit E_ν distribution.

Figure 9. Fit of the nominal NEUT v5.3.6 CCQE model to ANL CCQE E_ν and $N_\mathrm{evt}(Q^2)$ distributions [16] where the only free parameter in the axial mass M_A. The χ^2 contributions from each distribution to their joint likelihood is shown as a function of M_A, and the best fit error is indicated. The best fit and nominal distributions are compared to the E_ν distribution.



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

(b) Nominal and best fit MINERvA distribution.

Figure 10. Fit of NEUT v5.3.6 to MINERvA ν_μ-CH CCQE data [23] with a prior constraint set by a fit to ANL CCQE data, shown in figure 9.

prior constraint on M_A which contributes a penalty to the fit. The contribution from the χ^2 from the MINERvA data and the ANL prior is indicated, and it is clear that the MINERvA data favours a higher M_A, contesting the ANL prior. It is also possible to use the output of the *nuismin* fit as an input to most other NUISANCE applications. For example, after running a fit to MiniBooNE CC1π^+ data, it might be desirable to produce error bands showing the effect of the uncertainty on T2K CC1π^+ data, for which the *nuissyst* application can be used.

4 Summary

NUISANCE is a flexible tool for comparing all commonly used neutrino event generators with published cross-section data. It provides a common ground for comparing generators as well as testing and tuning model parameters. It has already proven an invaluable tool for T2K studies of cross-section parameters [7], and has been made open access in the hope that it will prove useful to the wider community. A number of different possible usage cases have been identified for NUISANCE:

- users who wish to make a comprehensive range of well-validated generator comparisons to new cross-section datasets without having to be familiar with all of the generators;
- users who wish to validate their cross-section parametrisation and error budget with a variety of historical cross-section data, or test new parameters against them;
- users who wish to tune and select default cross-section models for a given generator to a wide variety of cross-section data for cross-section and oscillation experiments;
- users who are interested in evaluating systematic uncertainties for systematics by comparing predictions of multiple generators.

This paper provides a number of examples to demonstrate the types of analyses which are straight-forward to perform with NUISANCE. Further documentation, usage examples and guidance can be found at nuisance.hepforge.org. We welcome code contributions, collaboration and new members.

Acknowledgments

The authors would like to thank the members of the T2K collaboration for the help and support when developing NUISANCE. We would also like to thank the authors of the NEUT, NuWro, GiBUU, and GENIE generators for being responsive to questions regarding the interface to their software. L. Pickering would like to extend special thanks to U. Mosel for his help validating the NUISANCE GiBUU integration. We thank the MINERvA, MiniBooNE and T2K collaborations for assistance in understanding their results and helping us to validate the output from NUISANCE. We would like to thank HepForge for hosting the NUISANCE framework. We acknowledge the support of MEXT, Japan; National Science Centre (NCN), Poland; MINECO and ERDF funds, Spain; SNSF and SER, Switzerland; STFC, U.K.; and DOE, U.S.A. KM would like to acknowledge the support of the DOE under award number 145568 and the Alfred P. Sloan Foundation.

References

[1] LBNE collaboration, C. Adams et al., *The Long-Baseline Neutrino Experiment: exploring fundamental symmetries of the Universe*, arXiv:1307.7335.

[2] T2K collaboration, K. Abe et al., *Neutrino oscillation physics potential of the T2K experiment*, *PTEP* **2015** (2015) 043C01 [arXiv:1409.7469].

[3] J.A. Formaggio and G.P. Zeller, *From eV to EeV: neutrino cross sections across energy scales*, *Rev. Mod. Phys.* **84** (2012) 1307.

[4] L. Álvarez-Ruso, Y. Hayato and J. Nieves, *Progress and open questions in the physics of neutrino cross sections at intermediate energies*, *New J. Phys.* **16** (2014) 075015 [arXiv:1403.2673].

2017 JINST 12 P01016

[5] G.T. Garvey, D.A. Harris, H.A. Tanaka, R. Tayloe and G.P. Zeller, *Recent advances and open questions in neutrino-induced quasi-elastic scattering and single photon production*, Phys. Rept. **580** (2015) 1 [arXiv:1412.4294].

[6] GNU General Public License (GPLv3).

[7] C. Wilkinson et al., *Testing charged current quasi-elastic and multinucleon interaction models in the NEUT neutrino interaction generator with published datasets from the MiniBooNE and MINERvA experiments*, Phys. Rev. **D 93** (2016) 072010 [arXiv:1601.05592].

[8] R. Brun and F. Rademakers, *ROOT: an object oriented data analysis framework*, Nucl. Instrum. Meth. **A 389** (1997) 81.

[9] C. Andreopoulos et al., *The GENIE neutrino Monte Carlo generator*, Nucl. Instrum. Meth. **A 614** (2010) 87 [arXiv:0905.2517].

[10] C. Andreopoulos et al., *The GENIE neutrino Monte Carlo generator: physics and user manual*, arXiv:1510.05494.

[11] Y. Hayato, *A neutrino interaction simulation program library NEUT*, Acta Phys. Polon. **B 40** (2009) 2477.

[12] C. Juszczak, *Running NuWro*, Acta Phys. Polon. **B 40** (2009) 2507 [arXiv:0909.1492].

[13] O. Buss et al., *Transport-theoretical description of nuclear reactions*, Phys. Rept. **512** (2012) 1 [arXiv:1106.1344].

[14] D. Casper, *The Nuance neutrino physics simulation and the future*, Nucl. Phys. Proc. Suppl. **112** (2002) 161 [hep-ph/0208030].

[15] MiniBooNE collaboration, A.A. Aguilar-Arevalo et al., *The neutrino flux prediction at MiniBooNE*, Phys. Rev. **D 79** (2009) 072002 [arXiv:0806.1449].

[16] S.J. Barish et al., *Study of neutrino interactions in hydrogen and deuterium. 1. Description of the experiment and study of the reaction neutrino $d \to \mu^- pp(s)$*, Phys. Rev. **D 16** (1977) 3103.

[17] MiniBooNE collaboration, A.A. Aguilar-Arevalo et al., *First measurement of the muon neutrino charged current quasielastic double differential cross section*, Phys. Rev. **D 81** (2010) 092005 [arXiv:1002.2680].

[18] MINERvA collaboration, B. Eberly et al., *Charged pion production in ν_μ interactions on hydrocarbon at $\langle E_\nu \rangle = 4.0$ GeV*, Phys. Rev. **D 92** (2015) 092008 [arXiv:1406.6415].

[19] ArgoNeuT collaboration, R. Acciarri et al., *Measurements of inclusive muon neutrino and antineutrino charged current differential cross sections on Argon in the NuMI antineutrino beam*, Phys. Rev. **D 89** (2014) 112003 [arXiv:1404.4809].

[20] L. Pickering, P. Stowell and J. Sobczyk, *Event reweighting with the NuWro neutrino interaction generator*, in the proceedings of the 27th *International Conference on Neutrino Physics and Astrophysics (Neutrino 2016)*, July 4–9, London, U.K. (2016), arXiv:1610.07053.

[21] MINERvA collaboration, C.L. McGivern et al., *Cross sections for ν_μ and $\bar{\nu}_\mu$ induced pion production on hydrocarbon in the few-GeV region using MINERvA*, Phys. Rev. **D 94** (2016) 052005 [arXiv:1606.07127].

[22] F. James and M. Roos, *Minuit: a system for function minimization and analysis of the parameter errors and correlations*, Comput. Phys. Commun. **10** (1975) 343.

2017 JINST 12 P01016

[23] MINERvA collaboration, G.A. Fiorentini et al., *Measurement of Muon Neutrino Quasielastic Scattering on a Hydrocarbon Target at $E_\nu \sim 3.5$ GeV*, Phys. Rev. Lett. **111** (2013) 022502 [arXiv:1305.2243].

2017 JINST 12 P01016