

Data Releases and Preservation

Luke Pickering

Measuring neutrino interactions for next-generation oscillation experiments

ECT*, Trento, 2024/10/25

Data Releases

Why Release?

- Data is very valuable (and expensive: economic and human time)
- Improving interaction models without high-quality, accessible measurements to predict is both very difficult and ~pointless
- For neutrino oscillations: We need improved models for the headline precision physics goals

What do we want from data releases?

- An unambiguous prescription for making a prediction of a measurement and calculating a goodness of fit for the prediction

Data Releases

Why Release?

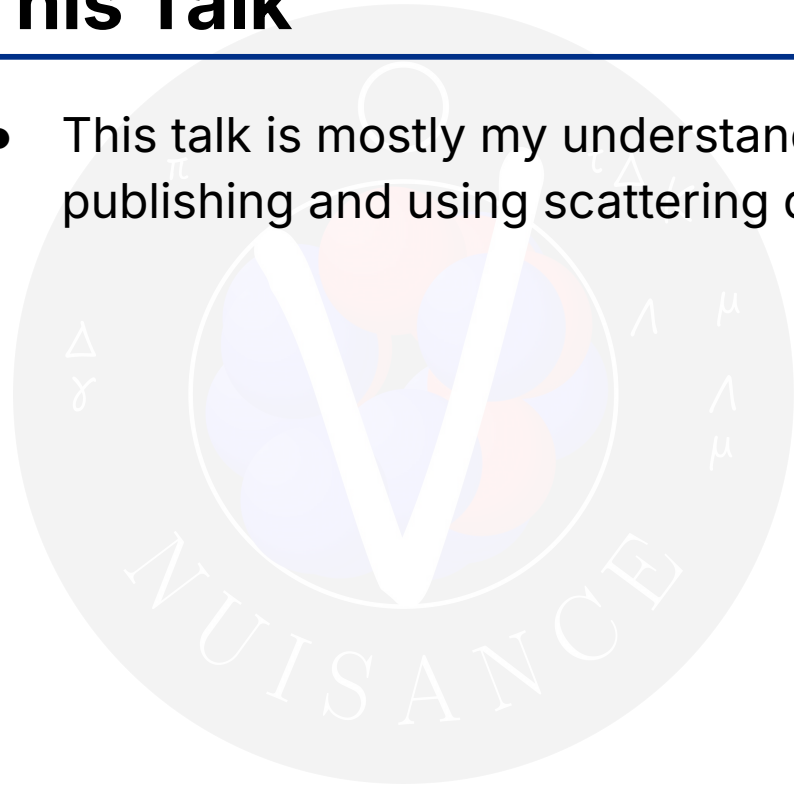
- Data is very valuable (and expensive: economic and human time)
- Improving interaction models without high-quality, accessible measurements to predict is both very difficult and ~pointless
- For neutrino oscillations: We need improved models for the headline precision physics goals

What do we want from data releases?

- An unambiguous prescription for making a prediction of a measurement and calculating a goodness of fit for the prediction
- Simple, right?

This Talk

- This talk is mostly my understanding of the **problem domain** of publishing and using scattering data



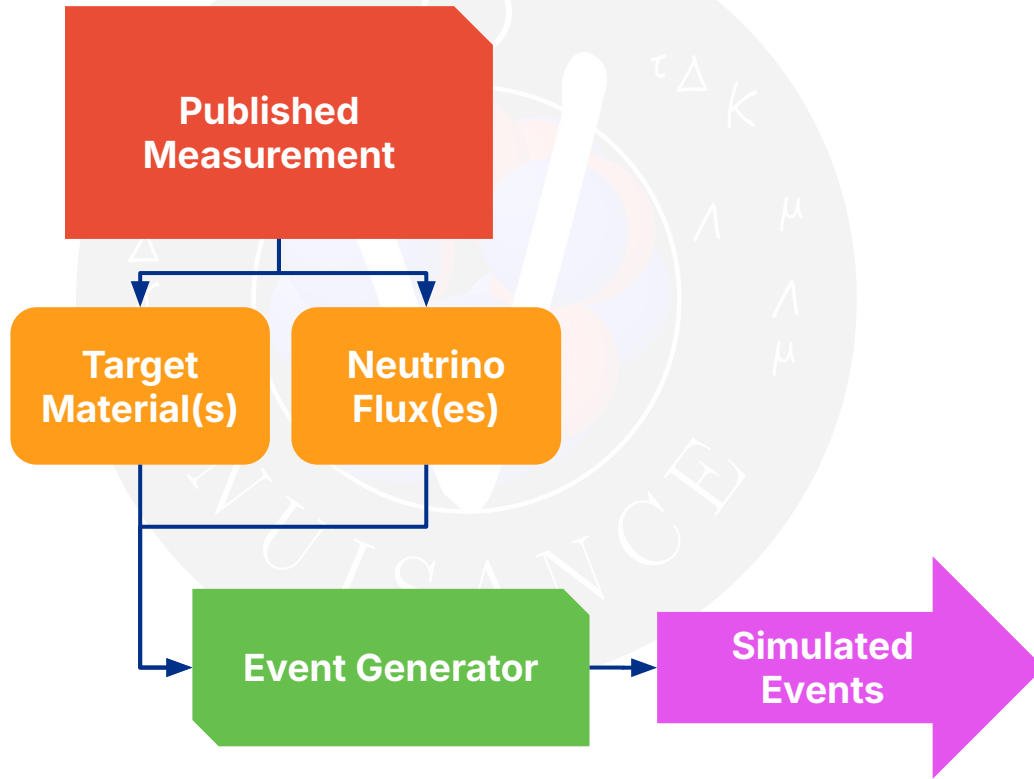
This Talk

- This talk is mostly my understanding of the **problem domain** of publishing and using scattering data
 - Ultimately, the problem is not difficult to solve unlike other problems discussed at this workshop:
 - Whats are best measurements to make?
 - Which statistical meat grinder to use to summarise measurement?
 - How to model neutrino-nucleus interactions in the few-GeV region
 - ...

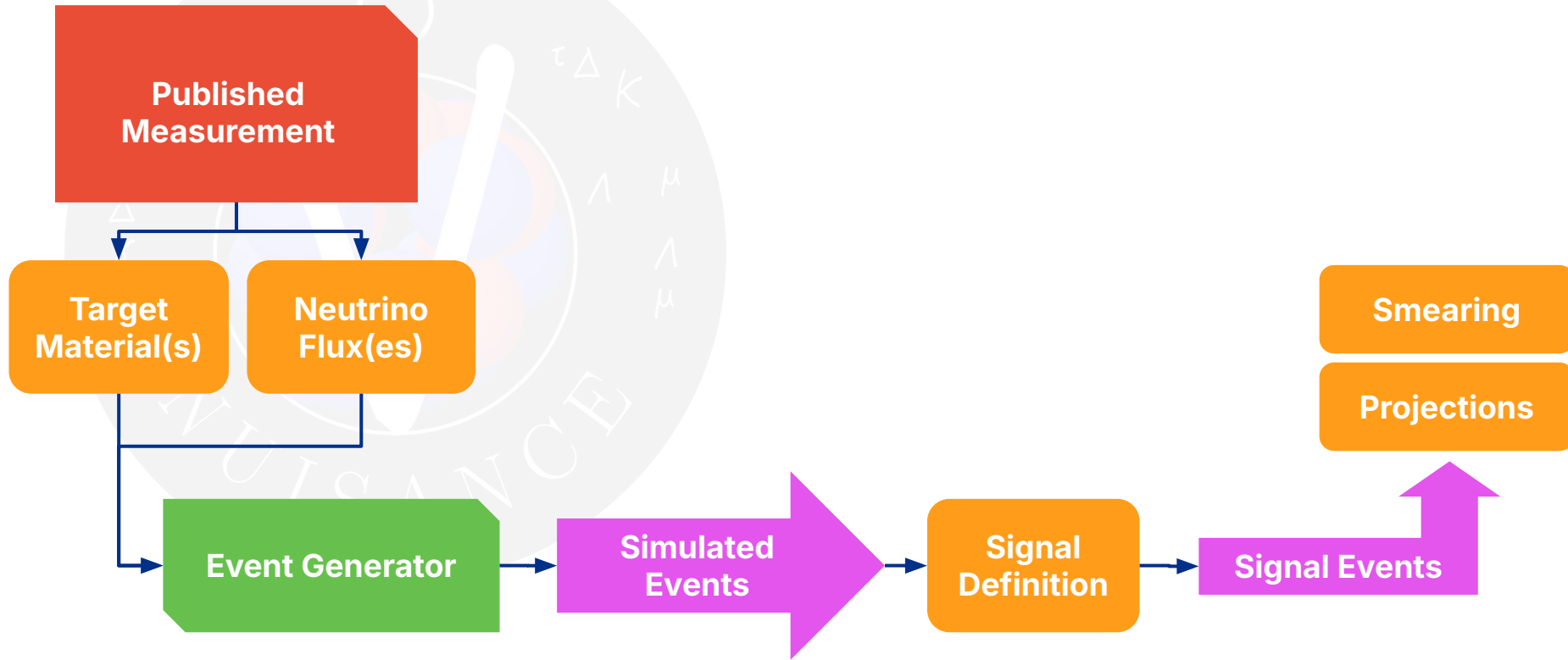
This Talk

- This talk is mostly my understanding of the **problem domain** of publishing and using scattering data
 - Ultimately, the problem is not difficult to solve unlike other problems discussed at this workshop:
 - Whats are best measurements to make?
 - Which statistical meat grinder to use to summarise measurement?
 - How to model neutrino-nucleus interactions in the few-GeV region
 - ...
- I will also show a quick example of a possible solution that we've been working on
- **BUT:** The most useful solution is the one that is used, so this is also a call to engage!

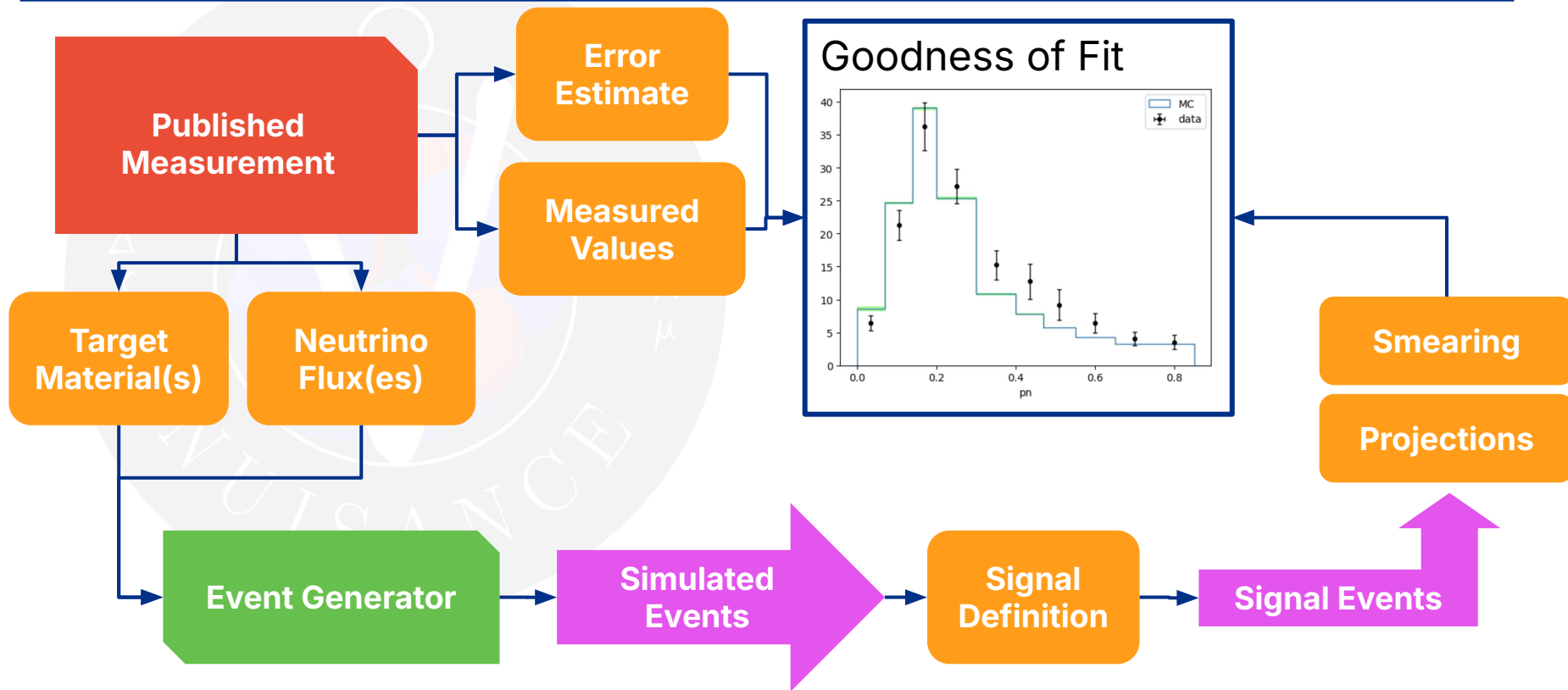
Measurement Predictions: A Workflow



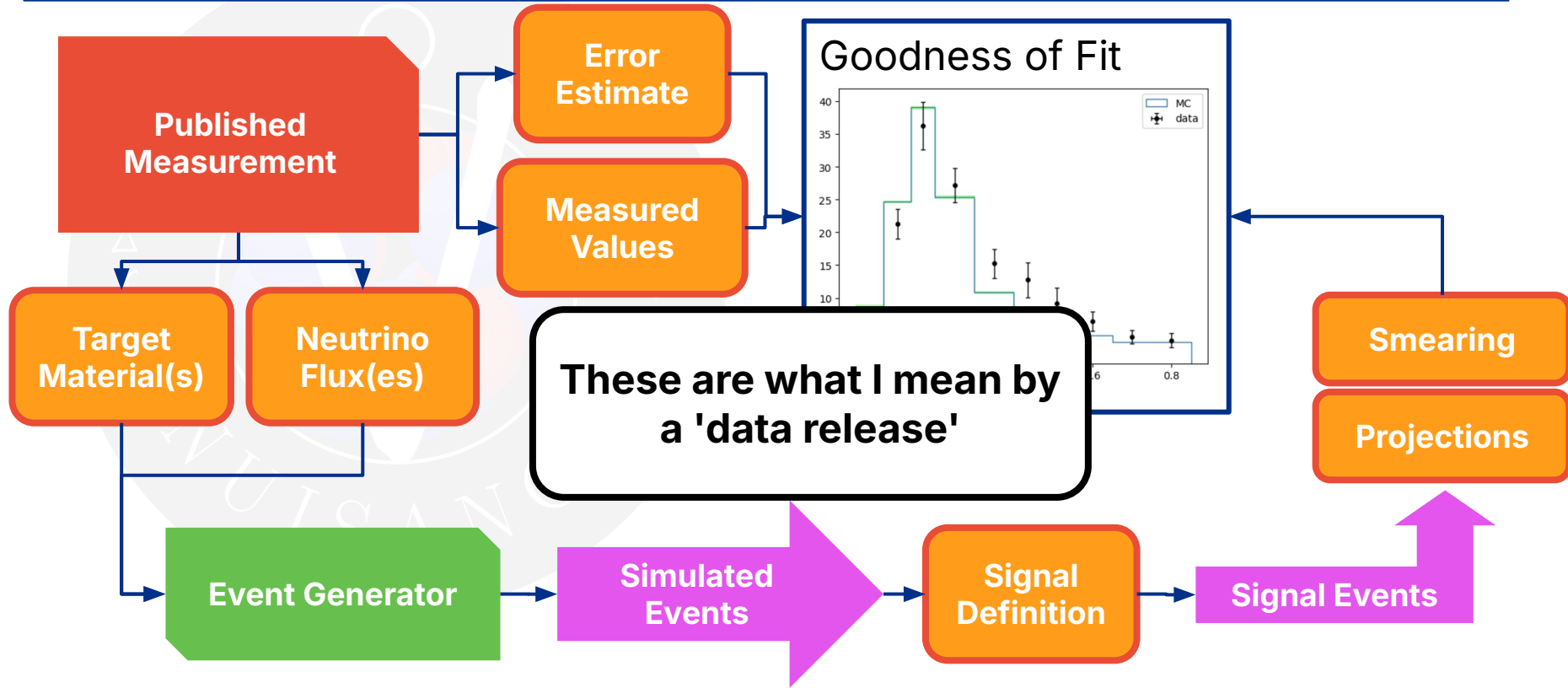
Measurement Predictions: A Workflow



Measurement Predictions: A Workflow

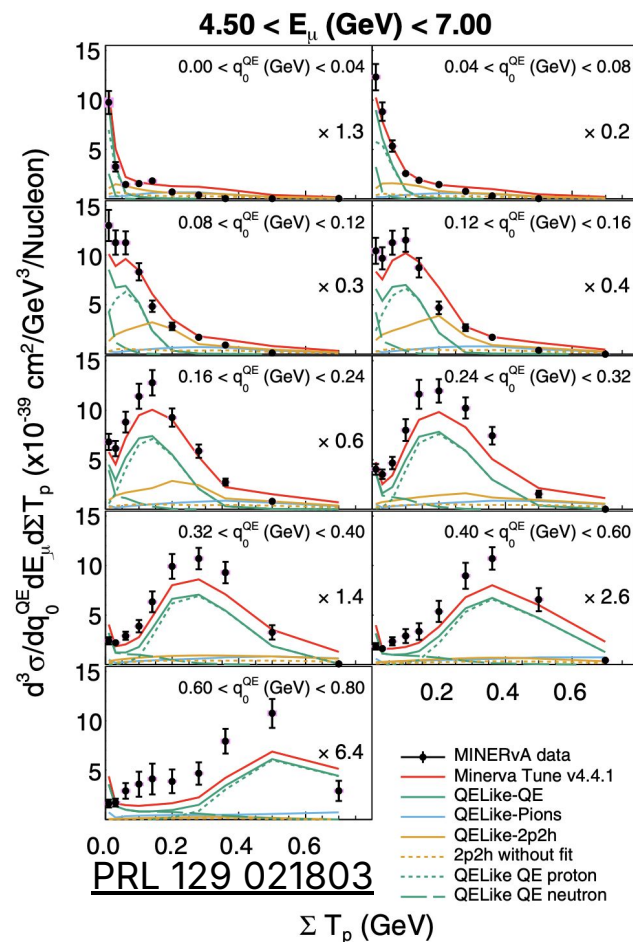
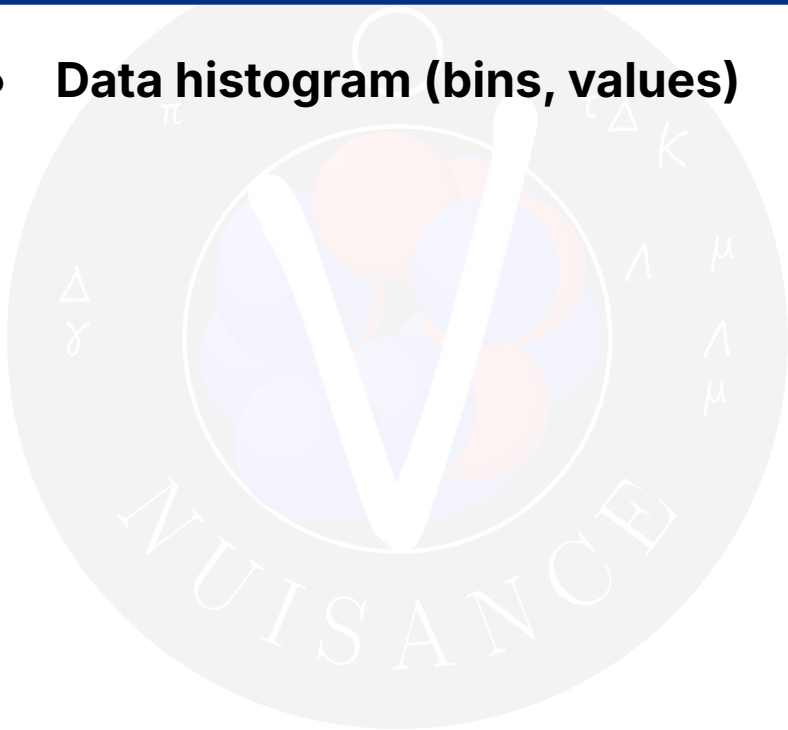


Measurement Predictions: A Workflow



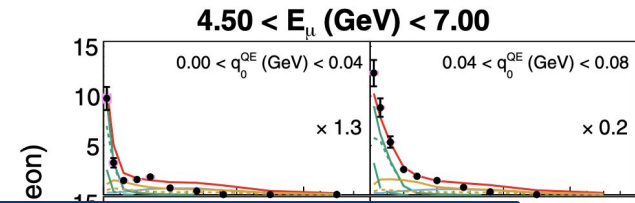
An Example from MINERvA

- Data histogram (bins, values)



An Example from MINERvA

- Data histogram (bins, values)
- Errors (covariance in bin number)
- **Flux shape**



NuMI Low Energy Flux Prediction Release

Neutrino Flux Predictions for the NuMI Beam
Phys. Rev. D 94, 092005, hep-ex/1607.00704

Data

- Ancillary data files for this result are available on arXiv at <http://arxiv.org/src/1607.00704/anc>
- Among the available data files are:
 - pdf file describing format of all the available files
 - root file of all the available fluxes
 - python code to read and process MINERvA's flux predictions
 - Text Files of the flux, uncertainties, and covariance matrix, with units of neutrinos/m²POT, in 0.5 GeV bins going up to 100 GeV. The first line in the file is the flux, the second is the uncertainty (200 numbers each), then there are 200 lines, each with 200 entries for the covariance matrix.
 - Neutrino Mode (FHC) Muon Neutrino Flux
 - Neutrino Mode (FHC) Electron Neutrino Flux
 - Neutrino Mode (FHC) Muon Antineutrino Flux
 - Neutrino Mode (FHC) Electron Antineutrino Flux
 - Antineutrino Mode (RHC) Muon Neutrino Flux
 - Antineutrino Mode (RHC) Electron Neutrino Flux
 - Antineutrino Mode (RHC) Muon Antineutrino Flux
 - Antineutrino Mode (RHC) Electron Antineutrino Flux

Contact Information

- For information on use of this MINERvA public data or for inquiries about additional data not linked from this page, contact Mike Kordosky or Kevin McFarland and Debbie Harris

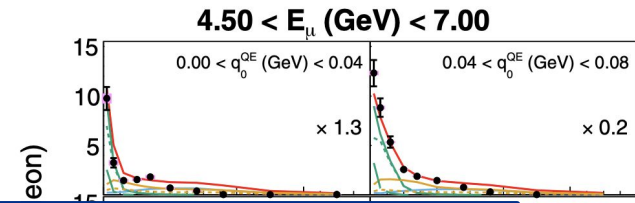
0.6
A data
Tune v4.4.1
QE
QE

0.0 0.2 0.4 0.6
PRL 129 021803
Σ T_p (GeV)

- QELike-Pions
- QELike-2p2h
- - - 2p2h without fit
- - - QELike QE proton
- - - QELike QE neutron

An Example from MINERvA

- Data histogram (bins, values)
- Errors (covariance in bin number)
- Flux shape
- **Target: CH**



NuMI Low Energy Flux Prediction Release

Neutrino Flux Predictions for the NuMI Beam
Phys. Rev. D 94, 092005, hep-ex/1607.00704

Data

- Ancillary data files for this result are available on arXiv at <http://arxiv.org/src/1607.00704/anc>
- Among the available data files are:
 - pdf file describing format of all the available files
 - root file of all the available fluxes
 - python code to read and process MINERvA's flux predictions
 - Text Files of the flux, uncertainties, and covariance matrix, with units of neutrinos/m²POT, in 0.5 GeV bins going up to 100 GeV. The first line in the file is the flux, the second is the uncertainty (200 numbers each), then there are 200 lines, each with 200 entries for the covariance matrix.
 - Neutrino Mode (FHC) Muon Neutrino Flux
 - Neutrino Mode (FHC) Electron Neutrino Flux
 - Neutrino Mode (FHC) Muon Antineutrino Flux
 - Neutrino Mode (FHC) Electron Antineutrino Flux
 - Antineutrino Mode (RHC) Muon Neutrino Flux
 - Antineutrino Mode (RHC) Electron Neutrino Flux
 - Antineutrino Mode (RHC) Muon Antineutrino Flux
 - Antineutrino Mode (RHC) Electron Antineutrino Flux

Contact Information

- For information on use of this MINERvA public data or for inquiries about additional data not linked from this page, contact Mike Kordosky or Kevin McFarland and Debbie Harris

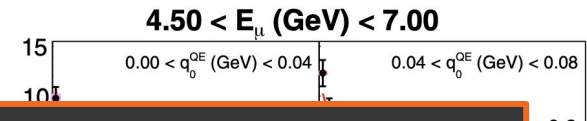
— QELike-Pions
— QELike-2p2h
- - - 2p2h without fit
- - - QELike QE proton
- - - QELike QE neutron

PRL 129 021803

ΣT_p (GeV)

An Example from MINERvA

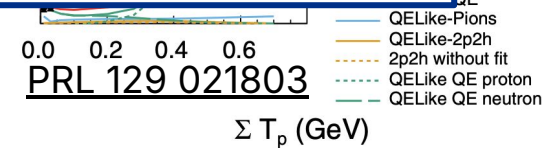
- Data histogram (bins, values)
- Errors (covariance in bin numbers)
- Flux shape
- Target: CH
- **Signal definition**



```

PRL.129.021803
README
1 Data release README. All files are contained in the associated compressed
2 tarball - MINERvA_TripleDiffQELike_DataRelease.tar.gz
3
4 Signal definition-
5
6 Kinematic windows:
7
8 P|| > 1.5 GeV
9 Muon Angle w.r.t neutrino < 20 degrees
10
11 Interaction and final state information:
12
13 Charge Current interaction of a muon type neutrino with the following
14 particles allowed in the final state:
15 muon, any number of nucleons, gammas <10 MeV (nuclear deexcitation)
    
```

- Antineutrino Mode (RHC) Muon Antineutrino Flux
 - Antineutrino Mode (RHC) Electron Antineutrino Flux
 - Antineutrino Mode (RHC) Muon Neutrino Flux
 - Antineutrino Mode (RHC) Electron Neutrino Flux
- Contact Information**
- For information on use of this MINERvA public data or for inquiries about additional data not linked from this page, contact Mike Kordosky or Kevin McFarland and Debbie Harris



An Example from MINERvA

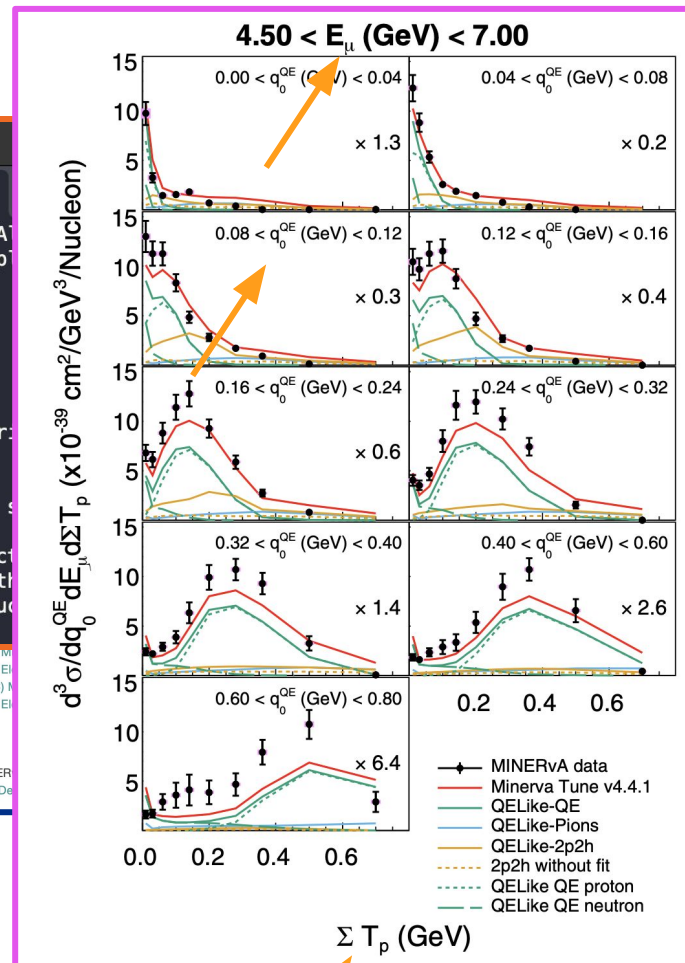
- Data histogram (bins, values)
- Errors (covariance in bin number)
- Flux shape
- Target: CH
- Signal definition
- **Projection operators**

```
PRL.129.021803
README
1 Data release README. A
tarball - MINERvA_Trip
2
3 Signal definition-
4
5 Kinematic windows:
6
7 P|| > 1.5 GeV
8 Muon Angle w.r.t neutr
9
10
11 Interaction and final s
12
13 Charge Current interact
particles allowed in the
14 muon, any number of nu
15
```

```
36
37 ###
38 SumTp vs q0_qe vs Muon E
39 ###
40 ----
```

Contact Information

- For information on use of this MINERvA data release, contact Kordosky or Kevin McFarland and De



An Example

- Data histograms
- Errors (covariance)
- Flux shape
- Target
- Signal definition
- Projection options

NuMI Low Energy Flux Prediction Release

Neutrino Flux Predictions for the NuMI Beam
 Phys. Rev. D 94, 092005, hep-ex/1607.00704

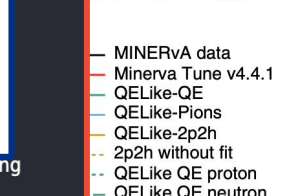
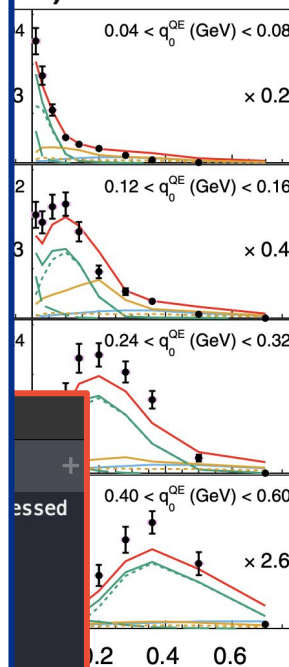
Data

- Ancillary data files for this result are available on arXiv at <http://arxiv.org/src/1607.00704/anc>
- Among the available data files are:
 - pdf file describing format of all the available files
 - root file of all the available fluxes
 - python code to read and process MINERvA's flux predictions
 - Text Files of the flux, uncertainties, and covariance matrix, with units of neutrinos/m²/POT, in 0.5 GeV bins going up to 100 GeV. The first line in the file is the flux, the second is the uncertainty (200 numbers each), then there are 200 lines, each with 200 entries for the covariance matrix.
 - Neutrino Mode (FHC) Muon Neutrino Flux
 - Neutrino Mode (FHC) Electron Neutrino Flux
 - Neutrino Mode (FHC) Muon Antineutrino Flux
 - Neutrino Mode (FHC) Electron Antineutrino Flux
 - Antineutrino Mode (RHC) Muon Antineutrino Flux
 - Antineutrino Mode (RHC) Electron Antineutrino Flux
 - Antineutrino Mode (RHC) Muon Neutrino Flux
 - Antineutrino Mode (RHC) Electron Neutrino Flux

Contact Information

- For information on use of this MINERvA public data or for inquiries about additional data not linked from this page, contact [Mike Kordosky](#) or [Kevin McFarland](#) and [Debbie Harris](#)

(eV) < 7.00



13 Charge Current interaction of a muon type neutrino with the following
 14 particles allowed in the final state:
 15 muon, any number of nucleons, gammas <10 MeV (nuclear deexcitation)

Overview of Encountered Approaches

Cross section data/Flux predictions/Error

- Plots in papers

Requires manual digitisation, infeasible for many bin measurements

I've tried asking ChatGPT to do it... don't recommend

Overview of Encountered Approaches

Cross section data/Flux predictions/Error

- Plots in papers
- Tables in papers

ChatGPT is probably better at this one, but it still requires human effort to 'get' the data

Overview of Encountered

Cross section data/Flux predictions/Error

- Plots in papers
- Tables in papers
- Text tables in supplementary materials
- ROOT files in supplementary materials
- File storage on experiment webspace/third party repository

Better! As there is no standard format between or within experiments, requires (sometimes significant) effort on the part of the user to understand the form of the data

The Ill-posed Analysis Problem

```
analysis.cxx -- neutrino_data
<F> #include <...>
11 [part::event::CPM_C2M_SpR3C3Spa] [part::event::cont::Gen] {
12     if (part::event::is_mom_part(ev, ps::pdp::isMOM)) {
13         ps::event::is_mom_part(ev, ps::pdp::isMOM, 1) {
14             ps::event::is_mom_part(ev, ps::pdp::isMOM) {
15                 ps::event::is_mom_part(ev, ps::pdp::isMOM) {
16                     ps::event::is_mom_part(ev, ps::pdp::isMOM) {
17                         ps::pdp::isMOM(ev, ps::pdp::isMOM, ps::pdp::isMOM) {
18                             ps::pdp::isMOM(ev, ps::pdp::isMOM) {
19                                 return false;
20                             }
21                         }
22                     }
23                 }
24             }
25         }
26     }
27     auto fu = ps::event::is_mom_part(ev, ps::pdp::isMOM);
28     auto mu = ps::event::is_mom_part(ev, ps::pdp::isMOM);
29     auto p_m_dev = mu - momentum::p_mom() / ps::unit::GeV_C;
30     if (p_m_dev < 0.3) { (1.2 * p_m_dev) {
31         }
32         return false;
33     }
34     auto fu_pretans = ps::event::is_mom_part(ev, ps::pdp::isMOM);
35     auto fu_pretans_m_s = ps::part::is_mom_part(ev, ps::pdp::isMOM);
36     [part::is_mom = true = ps::unit::GeV_C] [ps::pdp::isMOM = ps::unit::GeV_C];
37     fu_pretans;
38     if (fu_pretans_m_s.isMOM() != 1) {
39         }
40         return false;
41     }
42     auto fu_cpl = ps::part::is_mom_part(ev, ps::pdp::isMOM);
43     auto fu_cpl_observable = ps::part::is_mom_part(ev, ps::pdp::isMOM);
44     ps::part::is_mom_part(ev, ps::pdp::isMOM);
45     if (fu_cpl_observable.isMOM() != 1) {
46         }
47         return false;
48     }
49     return true;
50 }
51 }
```

Expert analyser writes code for signal/projection

AN

- Vaguely hinted at throughout paper text
- Clearly signposted signal definition in paper and mathematical representation of independent projections
- Official example implementation (in NUISANCE or elsewhere)

The Ill-posed Analysis Problem

```

analysis_cxx --neutrino_data
4 #include
5
6 // [ps: event::nu_muon_pos(ev, ps: id): id(mu)]
7 if (ps: event::nu_muon_pos(ev, ps: id): id(mu)) {
8     ps: event::nu_muon_pos(ev, ps: id): id(mu)
9     ps: event::nu_muon_pos(ev, ps: id): id(mu)
10 }
11
12 # [ps: event::nu_muon_pos(ev, ps: id): id(mu)]
13 # [ps: event::nu_muon_pos(ev, ps: id): id(mu)]
14
15 auto m0 = ps: event::nu_muon_pos(ev, ps: id): id(mu);
16 auto m1 = ps: event::nu_muon_pos(ev, ps: id): id(mu);
17
18 if (m0 < m1) {
19     m0 = m1;
20 }
21
22 auto f0 = ps: event::nu_muon_pos(ev, ps: id): id(mu);
23 auto f1 = ps: event::nu_muon_pos(ev, ps: id): id(mu);
24
25 if (f0 > f1) {
26     f0 = f1;
27 }
28
29 auto f0 = ps: event::nu_muon_pos(ev, ps: id): id(mu);
30 auto f1 = ps: event::nu_muon_pos(ev, ps: id): id(mu);
31
32 if (f0 > f1) {
33     f0 = f1;
34 }
35
36 return true;
37 }
38 }
39
40 }

```

Expert analyst writes code for signal/projection

PHYSICAL REVIEW LETTERS 129, 021803 (2022)

measurement of \mathcal{E}_T^* and of photon showers from $a^0 \rightarrow \gamma\gamma$. Occasionally, final-state neutrons leave a small amount of energy that is tagged as a photon or included in \mathcal{E}_T^* ; the reference simulation predicts and corrects for this effect. The average \mathcal{E}_T^* for protons is ≈ 250 MeV; neutrons contribute less than 10 MeV of energy in 74% of events and an average of 85 MeV for the rest.

The MINERVA detector response is simulated using GEANT4 [40] version 4.9.4p2 with the QGSP_BERT physics list. The optical and electronics performance is also simulated. Through-going muons are used to determine the absolute energy scale. Full descriptions of calibrations are given in Ref. [36, 41]. The absolute energy response to charged hadrons is set according to measurements using a charged particle test beam [42] and a scaled-down version of the MINERVA detector. The effects of accidental activity at a function of beam intensity are simulated by overlaying hits from data in both MINERVA and MINOS.

The reference signal and background models for this analysis are based on a modified version of the GENIE [43] v2.12.0 event generator. Quasielectronic interactions are modeled using the Llewellyn Smith formalism [44] with BBMSO vector form factors [45] and an axial-vector form factor based on a χ -expansion fit to deuterium data [46]. Resonance production is simulated using the Rein-Sehgal model [47] with a dipole axial mass of $M_{\text{eff}}^{(A)} = 1.12$ GeV/c². The nuclear initial state is a relativistic Fermi gas model [48] with $k_F = 0.221$ GeV/c and with a Brodeur-Richie high momentum tail [49]. Multinucleon quasielectronic-like interactions are simulated by the Valencia model¹ described in Ref. [10, 11, 50]. Intranuclear final-state interactions of produced hadrons are modeled using the re-scattering package [51].

To better describe MINERVA data, a number of modifications are made to the reference model that are collectively denoted MINERVA-stm v4.1. The quasielectronic section is modified as a function of energy and three-momentum transfer based on the random phase approximation of the Valencia model [52, 53] appropriate for a Fermi gas [54, 55] to account for long-range correlations between nucleons. To account for an observed excess in specific regions of three-momentum transfer and \mathcal{E}_T^* , the multinucleon cross section is increased based on fits to MINERVA data [17] from a lower energy beam configuration. Additionally, based on fits to ν -hydrogen data [56], the nonresonant charged-current pion production is decreased by 45%, the overall baryon-resonance pion production is increased by 15%, and $M_{\text{eff}}^{(A)}$ is set to 0.94 GeV.

Samples for simulating quasielectronic-like interactions and their backgrounds require a muon track that starts in the fiducial volume and is identified in MINOS as negatively charged. All other tracked particles originating from the interaction vertex at the beginning of the muon track must have dE/dx consistent with a proton. Signal and background samples are formed by counting the number of Michel electron candidates within 600 mm long 600 mm diameter cylinders centered on the neutrino vertex and on endpoints of tracked particles, and by counting isolated clusters connected from two-dimensional clusters with at least 1 MeV visible energy. The former identify a^0 , and the latter identify photons from a^0 decays. Clusters with an

```

gen_reader
1 // User/Run/Python
2
3 # This code is intended to be compatible with both python2 and python3
4 from __future__ import print_function
5
6 import sys
7 import argparse
8 import os
9 import sys
10
11 # Don't spam pygccol deprecation warnings
12 # Suppress warnings
13 import logging
14 logging.getLogger('pygccol').setLevel(logging.WARNING)
15
16 # Globals
17 FILE = None
18 PATH = None
19 FILE = None
20
21 # Extra member functions for specific classes.
22 class to_addon = ()
23
24
25 def friendly_name(klass):
26     if getattr(klass, 'friendly_name', None):
27         return klass.friendly_name
28     else:
29         return klass.__name__ + 'Proxy'
30
31 # Don't use all member arguments from vectors
32 def short_name(klass):
33     if hasattr(klass, 'short_name'):
34         return short_name(klass)
35     else:
36         return klass.__name__
37
38 inner = vector_content(klass)
39 if hasattr(klass, 'inner'):
40     inner = short_name(inner)
41 if pygccol.get_compiler().is_compiler('gcc'):
42     inner = full_name(inner)
43     return 'gen::vector<' + inner + '>'
44
45

```

Non-expert user implements prediction code

- Vaguely hinted at throughout paper text
- Clearly signposted signal definition in paper and mathematical representation of independent projections
- Official example implementation (in NUANSANCE or elsewhere)



The Ill-posed Analysis Problem

Lossy projection operator

Ill-posed unfolding problem

```

5 # analysis_cxx -- neutrino_data
6
7 #include
8
9 // The MINERVA CERN data set (MINERVA)
10 #define MINERVA_CERN_DATA_PATH "/home/minerva/minerva_data"
11 #define MINERVA_CERN_DATA_PATH "/" + MINERVA_CERN_DATA_PATH
12 #include <string>
13 #include <string>
14 #include <string>
15 #include <string>
16 #include <string>
17 #include <string>
18 #include <string>
19 #include <string>
20 #include <string>
21 #include <string>
22 #include <string>
23 #include <string>
24 #include <string>
25 #include <string>
26 #include <string>
27 #include <string>
28 #include <string>
29 #include <string>
30 #include <string>
31 #include <string>
32 #include <string>
33 #include <string>
34 #include <string>
35 #include <string>
36 #include <string>
37 #include <string>
38 #include <string>
39 #include <string>
40 #include <string>
41 #include <string>
42 #include <string>
43 #include <string>
44 #include <string>
45 #include <string>
46 #include <string>
47 #include <string>
48 #include <string>
49 #include <string>
50 #include <string>
51 #include <string>
52 #include <string>
53 #include <string>
54 #include <string>
55 #include <string>
56 #include <string>
57 #include <string>
58 #include <string>
59 #include <string>
60 #include <string>
61 #include <string>
62 #include <string>
63 #include <string>
64 #include <string>
65 #include <string>
66 #include <string>
67 #include <string>
68 #include <string>
69 #include <string>
70 #include <string>
71 #include <string>
72 #include <string>
73 #include <string>
74 #include <string>
75 #include <string>
76 #include <string>
77 #include <string>
78 #include <string>
79 #include <string>
80 #include <string>
81 #include <string>
82 #include <string>
83 #include <string>
84 #include <string>
85 #include <string>
86 #include <string>
87 #include <string>
88 #include <string>
89 #include <string>
90 #include <string>
91 #include <string>
92 #include <string>
93 #include <string>
94 #include <string>
95 #include <string>
96 #include <string>
97 #include <string>
98 #include <string>
99 #include <string>
100 #include <string>

```

Expert analyser writes code for signal/projection

PHYSICAL REVIEW LETTERS 126, 021803 (2021)

measurement of ΣT_1 and of photon showers from $n^+ \rightarrow \pi^+$. Occasionally, final-state neutrons leave a small amount of energy that is tagged as a photon or included in ΣT_1 ; the reference simulation predicts and corrects for this effect. The average ΣT_1 for protons is ≈ 250 MeV; neutrons contribute less than 10 MeV of energy in 74% of events and an average of 85 MeV for the rest.

The MINERVA detector response is simulated using GENIE [40] version 4.9.4p2 with the QGSP_BERT physics list. The optical and electronic performance is also simulated. Though glowing muons are used to determine the absolute energy scale, full descriptions of calibrations are given in Ref. [38, 41]. The absolute energy response to charged hadrons is set according to measurements using a charged particle test beam [42] and a scaled-down version of the MINERVA detector. The effects of accidental activity as a function of beam intensity are simulated by overlaying hits from data in both MINERVA and MINOS.

The reference signal and background models for this analysis are based on a modified version of the GENIE [43] v2.12.0 event generator. Quasineutrino interactions are modeled using the Llewellyn Smith formalism [44] with BBWV's vector form factors [45] and an axial-vector form factor based on a χ^2 -optimal fit to deuterium data [46]. Neutrino production is simulated using the Reiss-Seiberg model [47] with a dipole axial mass of $M_{A0}^{(3)}$ 1.12 GeV/c². The nuclear initial state is a relativistic Fermi gas model [48] with $\mu_n = 0.221$ GeV/c and with a Brode-Ricchi high momentum tail [49]. Multinucleon quasiparticle-like interactions are simulated by the Valencia model⁵

Paper committee describes analyser code in English

```

5 // ...
6 #include <string>
7 #include <string>
8 #include <string>
9 #include <string>
10 #include <string>
11 #include <string>
12 #include <string>
13 #include <string>
14 #include <string>
15 #include <string>
16 #include <string>
17 #include <string>
18 #include <string>
19 #include <string>
20 #include <string>
21 #include <string>
22 #include <string>
23 #include <string>
24 #include <string>
25 #include <string>
26 #include <string>
27 #include <string>
28 #include <string>
29 #include <string>
30 #include <string>
31 #include <string>
32 #include <string>
33 #include <string>
34 #include <string>
35 #include <string>
36 #include <string>
37 #include <string>
38 #include <string>
39 #include <string>
40 #include <string>
41 #include <string>
42 #include <string>
43 #include <string>
44 #include <string>
45 #include <string>
46 #include <string>
47 #include <string>
48 #include <string>
49 #include <string>
50 #include <string>
51 #include <string>
52 #include <string>
53 #include <string>
54 #include <string>
55 #include <string>
56 #include <string>
57 #include <string>
58 #include <string>
59 #include <string>
60 #include <string>
61 #include <string>
62 #include <string>
63 #include <string>
64 #include <string>
65 #include <string>
66 #include <string>
67 #include <string>
68 #include <string>
69 #include <string>
70 #include <string>
71 #include <string>
72 #include <string>
73 #include <string>
74 #include <string>
75 #include <string>
76 #include <string>
77 #include <string>
78 #include <string>
79 #include <string>
80 #include <string>
81 #include <string>
82 #include <string>
83 #include <string>
84 #include <string>
85 #include <string>
86 #include <string>
87 #include <string>
88 #include <string>
89 #include <string>
90 #include <string>
91 #include <string>
92 #include <string>
93 #include <string>
94 #include <string>
95 #include <string>
96 #include <string>
97 #include <string>
98 #include <string>
99 #include <string>
100 #include <string>

```

Non-expert user implements prediction code

- Vaguely hinted at throughout paper text
- Clearly signposted signal definition in paper and mathematical representation of independent projections
- Official example implementation (in MINISANCE or elsewhere)

Overview of Encountered Approaches

Cross section data/Flux predictions/Error matrices/Smearing matrices:

- Plots in papers
- Tables in papers
- Text tables in supplementary materials
- ROOT files in supplementary materials
- File storage on experiment webspace/third party repository

Conceptually simple to release:
Values + minimal metadata (bin definitions, variable units, ...)

Just needs standardisation!

Analysis Steps:

- Vaguely hinted at throughout paper text
- Clearly signposted signal definition in paper and mathematical representation of independent projections
- Official example implementation (in NUISANCE or elsewhere)

Overview of Encountered Approaches

Cross section data/Flux predictions/Error matrices/Smearing matrices:

- Plots in papers
- Tables in papers
- Text tables in supplementary materials
- ROOT files in supplementary materials
- File storage on experiment webspace/this

Conceptually simple to release:
Values + minimal metadata (bin definitions, variable units, ...)

Just needs standardisation!

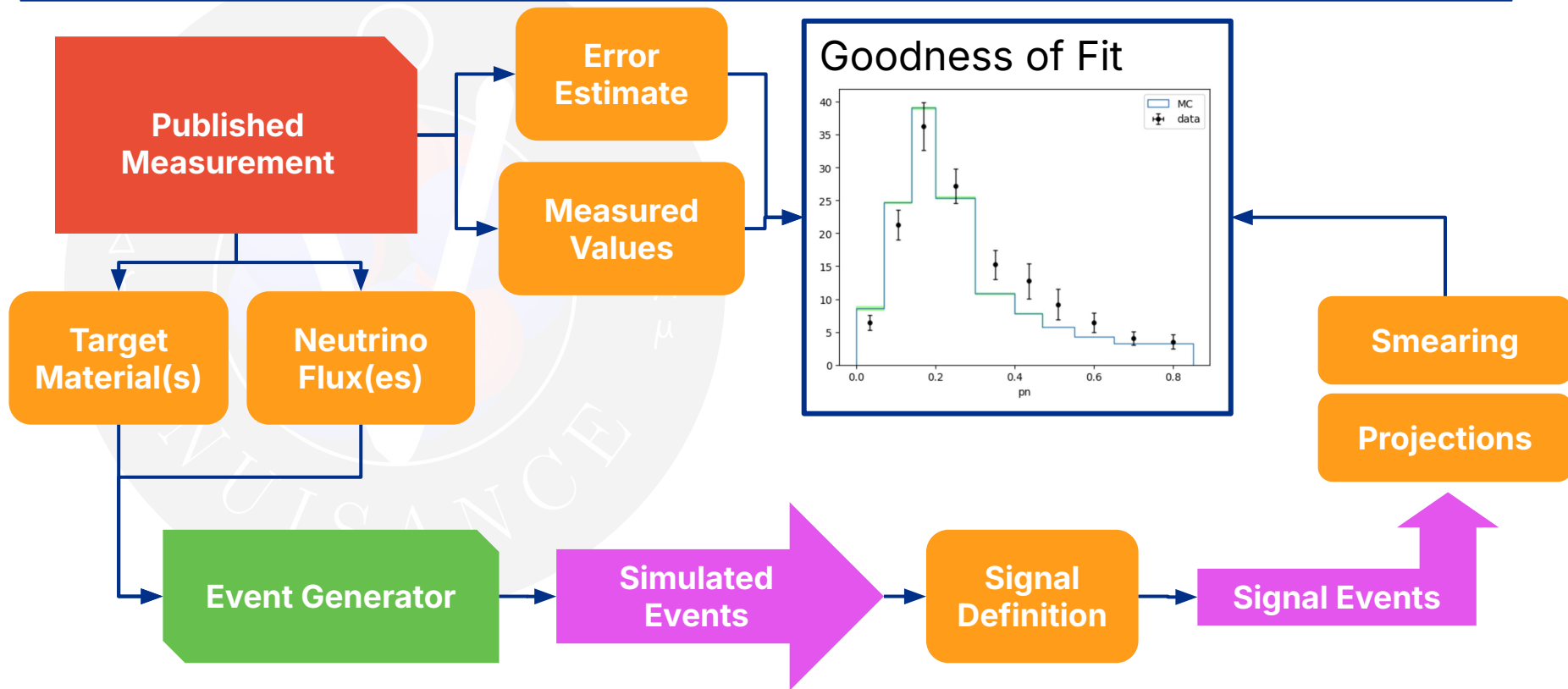
More difficult to release:
How to publish operators that are executable over a simulated event stream?

Just learn NUISANCE?

Analysis Steps:

- Vaguely hinted at throughout paper text
- Clearly signposted signal definition in paper and mathematical representation of independent projections
- Official example implementation (in NUISANCE or elsewhere)

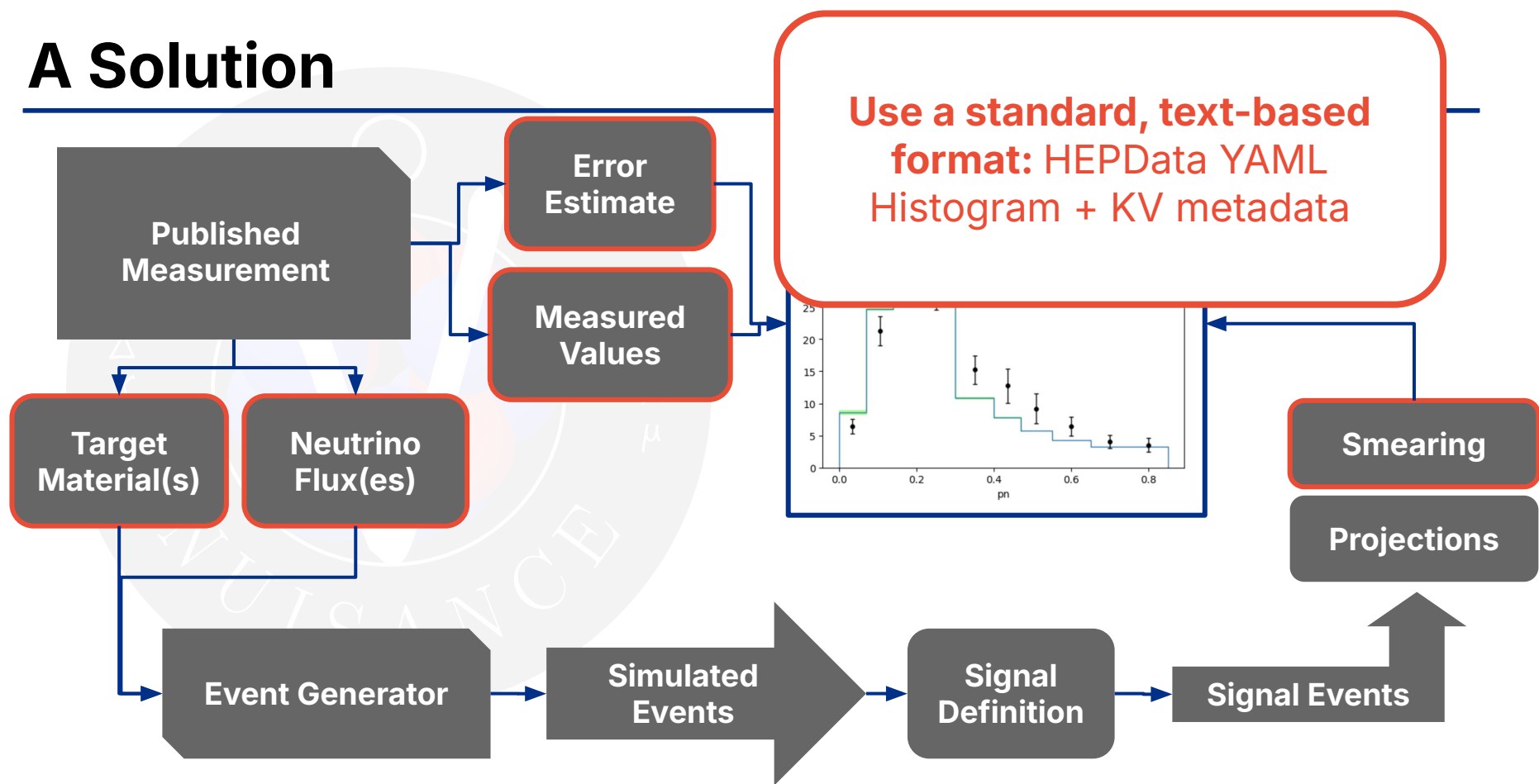
Measurement Predictions: A Workflow



Personal View: The Way Forward

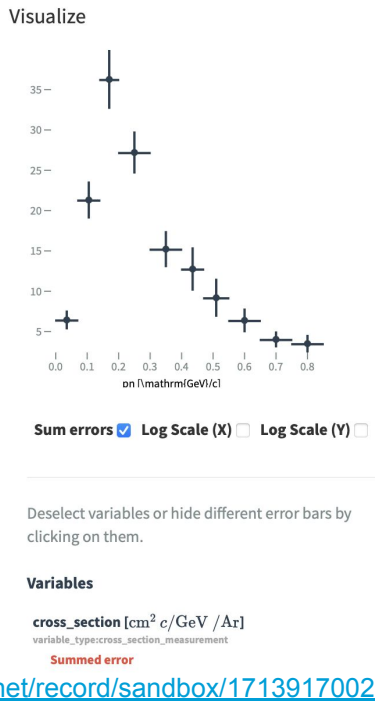
- Common Formats, Standards, Documentation:
 - **Data Release Aphorism:** Most of human progress has been driven by shared "languages"
- We need more community engagement on the 'boring' bits, so we can all do the interesting bits.
- Standardised, machine-readable formats for data with appropriate metadata
- Well-documented, open source, community tools to build/validate standardised data releases
- Measurements are getting more complicated, and more powerful!

A Solution



A Solution

variable_type	cross_section_measurement
selectfunc	analysis.cxx:MicroBooNE_CC0Pi_GKI_nu_SelectSignal
pn:projectfunc	analysis.cxx:MicroBooNE_CC0Pi_GKI_nu_pn
target	Ar
pn:prettyname	δp_n
prettyname	$d\sigma/d\delta p_n$
cross_section_units	1e-38 cm ² PerTarget per_bin_width
probe_flux	microboone_flux_numu
errors	covariance-deltapn
smearing	smearing-deltapn
pn [GeV/c]	cross_section [cm² c/GeV /Ar]
0.0 - 0.07	6.4406 ±1.1679 total
0.07 - 0.14	21.314 ±2.2968 total
0.14 - 0.2	36.266 ±3.6505 total
0.2 - 0.3	27.206 ±2.6118 total
0.3 - 0.4	15.223 ±2.2399 total
0.4 - 0.47	12.758 ±2.6894 total



<https://www.hepdata.net/record/sandbox/1713917002>

Use a standard, text-based format: HEPData YAML
Histogram + KV metadata

```
HEPData-1713917002-v1-cross_section-deltapn.yaml x
1 dependent_variables:
2   - header:
3     name: cross_section
4     units:  $\text{cm}^2 \text{c} / \text{GeV} \text{Ar}$ 
5     qualifiers:
6   - name: variable_type
7     value: cross_section_measurement
8   - name: selectfunc
9     value: analysis.cxx:MicroBooNE_CC0Pi_GKI_nu_SelectSignal
10  - name: pn:projectfunc
11    value: analysis.cxx:MicroBooNE_CC0Pi_GKI_nu_pn
12  - name: target
13    value: Ar
14  - name: pn:prettyname
15    value:  $\delta p_n$ 
16  - name: prettyname
17    value:  $\sigma(\text{GeV}/c)$ 
18  - name: cross_section_units
19    value: 1e-38 cm2|PerTarget|per_bin_width
20  - name: probe_flux
21    value: microboone_flux_numu
22  - name: errors
23    value: covariance-deltapn
24  - name: smearing
25    value: smearing-deltapn
26  values:
27  - errors:
28    - label: total
29      symerror: 1.1679
30      value: 6.4406
31  - errors:
32    - label: total
33      symerror: 2.2968
34      value: 21.314
35  - errors:
36    - label: total
37      symerror: 3.6505
38      value: 36.266
39  - errors:
40    - label: total
41      symerror: 2.6118
```

A Solution

Use a standard, event record format:
NuHepMC, based on HepMC3

SciPost Physics Codebases

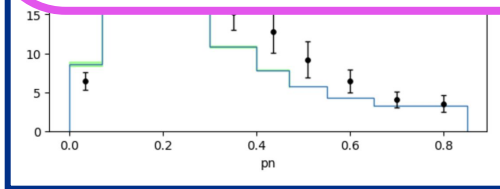
Submission

NuHepMC: A standardized event record format for neutrino event generators

Steven Gardiner^{1*}, Joshua Isaacson^{1†} and Luke Pickering^{2‡}

1 Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510, USA
2 STFC, Rutherford Appleton Laboratory, Harwell Oxford, United Kingdom

*gardiner@fnal.gov, †isaacson@fnal.gov, ‡luke.pickering@stfc.ac.uk



Smearing

Projections

Event Generator

Simulated Events

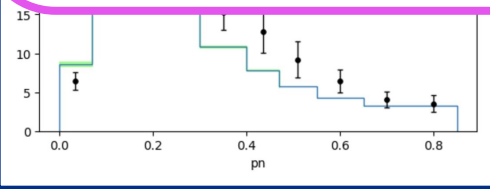
Signal Definition

Signal Events

A Solution

Use a standard, event record format: NuHepMC, based on HepMC3

Implementations for: NEUT, NuWro, GENIE, GiBUU, ACHILLES, Marley



Smearing

Projections

SciPost Physics Codebases

Submission

NuHepMC: A standardized event record format for neutrino event generators

Steven Gardiner^{1*}, Joshua Isaacson^{1†} and Luke Pickering^{2‡}

1 Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510, USA
2 STFC, Rutherford Appleton Laboratory, Harwell Oxford, United Kingdom

*gardiner@fnal.gov, †isaacson@fnal.gov, ‡luke.pickering@stfc.ac.uk

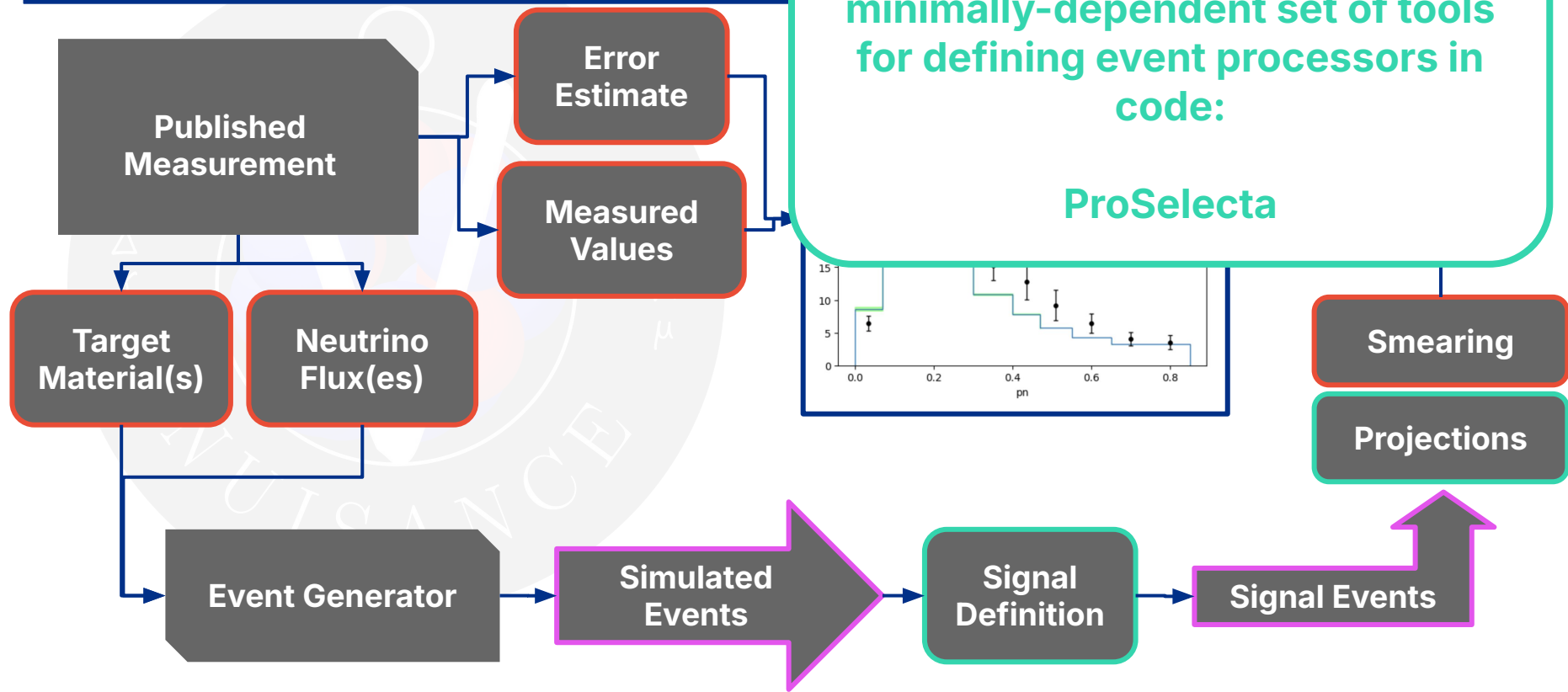
Event Generator

Simulated Events

Signal Definition

Signal Events

A Solution

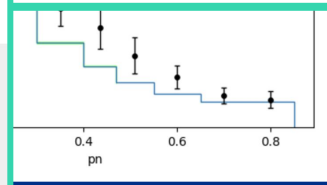


A Solution

Error Estimate

Use a standard, minimally-dependent, declarative set of tools for writing portable event processors:

ProSelecta



Smearing

Projections

Signal Definition

Signal Events

HEPData Search HEPData Sandbox About Submission Help

Browse all View all Publication Data ProSelecta File

Hide Publication Information

analysis.cxx License: CC0

Selection and projection function examples. Can be executed in the ProSelecta environment v1.0.

```
int T2K_nueCCPiplus_2024_Select(HepMC3::GenEvent const &ev) {
    using namespace ps::pdg;

    if (!ps::event::has_beam_part(ev, kNuE) ||
        !ps::event::has_at_least_out_part(ev, ps::pids(kElectron, kPIplus),
            {1, 1})) {
        return false;
    }

    return true;
}

double T2K_nueCCPiplus_2024_ElectronMomentum(HepMC3::GenEvent const &ev) {
    using namespace ps::pdg;

    if (!ps::event::has_beam_part(ev, kNuE) ||
        !ps::event::has_out_part(ev, kElectron)) {
        return ps::kMissingDatum<double>;
    }

    return ps::event::hm_out_part(ev, kElectron)->momentum().p3mod() /
        ps::unit::GeV;
}
```

Download

<https://www.hepdata.net/record/sandbox/1713917002>

1. Download and parse HEPData record:
 - a. Read data, errors, smearing, compile selection/projection operations. Fully automated.

```
• [2]: import pyNUISANCE as pn

rf = pn.RecordFactory()
hepdata_rec = rf.make_record({"type": "hepdata",
                             "recordref": "hepdata-sandbox:1713917002"})
```

<https://www.hepdata.net/record/sandbox/1713917002>

1. Download and parse HEPData record:
 - a. Read data, errors, smearing, compile selection/projection operations. Fully automated.
2. Interrogate for available analyses

```
print(hepdata_rec.get_analyses())
```

```
['cross_section-deltapn', 'cross_section-deltaalpha3dq', 'cross_section-deltaphi3d', 'cross_section-deltapnpar', 'cross_section-deltapnperp', 'cross_section-deltapnperpx', 'cross_section-deltapnperpy', 'cross_section-serialdeltapn_deltaalpha3dq', 'cross_section-serialdeltaalpha3dq_deltapn']
```

MicroBooNE GKI in Action

[Example notebook](#)

<https://www.hepdata.net/record/sandbox/1713917002>

1. Download and parse HEPData record:
 - a. Read data, errors, smearing, compile selection/projection operations. Fully automated.
2. Interrogate for available analyses
3. Interrogate analysis for flux/target information

```
[6]: ana = hepdata_rec.analysis("cross_section-deltapnperpx")
```

```
[10]: [ str(tgt) for tgt in ana.get_target() ]
```

```
[10]: ['{ A = 40, Z = 18, weight_by_mass = 1 }']
```

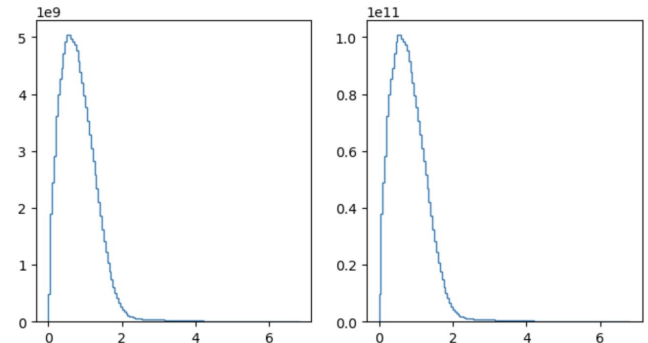
```
[7]: import matplotlib.pyplot as plt

probe_part, probe_flux_count = ana.get_probe_flux(False)
_, probe_flux_count_density = ana.get_probe_flux(True)

fig, axes = plt.subplots(1,2, figsize=(8,4))

probe_flux_count.mpl().hist(histtype="step",plot_axis=axes[0])
probe_flux_count_density.mpl().hist(histtype="step",plot_axis=axes[1])

plt.show()
```



MicroBooNE GKI in Action

[Example notebook](https://www.hepdata.net/record/sandbox/1713917002)

<https://www.hepdata.net/record/sandbox/1713917002>

1. Download and parse HEPData record:
 - a. Read data, errors, smearing, compile selection/projection operations. Fully automated.
2. Interrogate for available analyses
3. Interrogate analysis for flux/target information

```
[6]: ana = hepdata_rec.analysis("cross_section-deltapnperpx")
```

```
[10]: [ str(tgt) for tgt in ana.get_target() ]
```

```
[10]: ['{ A = 40, Z = 18, weight_by_mass = 1 }']
```

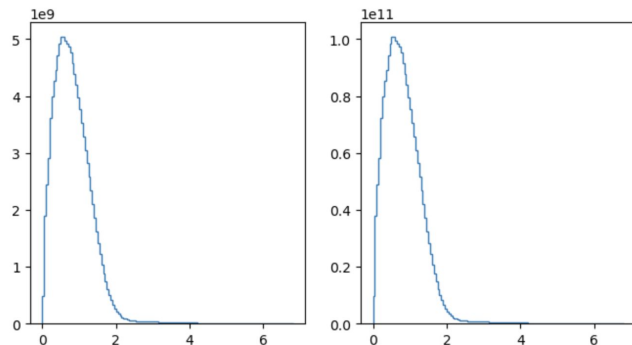
```
[7]: import matplotlib.pyplot as plt

probe_part, probe_flux_count = ana.get_probe_flux(False)
_, probe_flux_count_density = ana.get_probe_flux(True)

fig, axes = plt.subplots(1,2, figsize=(8,4))

probe_flux_count.mpl().hist(histtype="step",plot_axis=axes[0])
probe_flux_count_density.mpl().hist(histtype="step",plot_axis=axes[1])

plt.show()
```



<https://www.hepdata.net/record/sandbox/1713917002>

1. Download and parse HEPData record:
 - a. Read data, errors, smearing, compile selection/projection operations. Fully automated.
2. Interrogate for available analyses
3. Interrogate analysis for flux/target information
4. Load in event stream

```
[13]: neut_events = pn.EventSource("events_for_inspirehep_2709091v1.hepmc3.gz")
      if not neut_events:
          print("Failed to read file")
```

<https://www.hepdata.net/record/sandbox/1713917002>

1. Download and parse HEPData record:
 - a. Read data, errors, smearing, compile selection/projection operations. Fully automated.
2. Interrogate for available analyses
3. Interrogate analysis for flux/target information
4. Load in event stream
5. Build comparison

```
[14]: comparison = ana.process(neut_events)
```

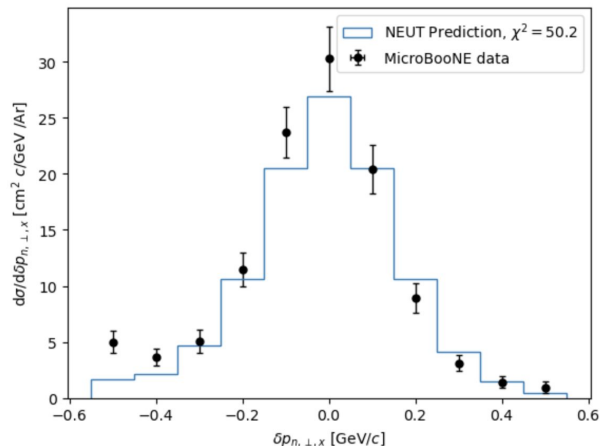
<https://www.hepdata.net/record/sandbox/1713917002>

1. Download and parse HEPData record:
 - a. Read data, errors, smearing, compile selection/projection operations. Fully automated.
2. Interrogate for available analysis
3. Interrogate analysis for flux/ta
4. Load in event stream
5. Build comparison
6. Make plots

```
[15]: comparison.data[0].mpl().data_hist(label="MicroBooNE data")
      comparison.predictions[0].mpl().hist(histtype="step",label="NEUT Prediction, $\chi^2$=$"+f"{comparison.likelihood():}
      print(comparison.likelihood())
      plt.legend()
      plt.plot()
```

50.22179776047446

```
[15]: []
```



How to Get Involved

Experimentalists:

- Publish new measurement data releases in a standardised format
 - **Bonus points:** convince your experimental collaboration to use a standard format
- **Be brave:** Read the ProSelecta specification and use it in defining your measurement

Theorists:

- If a measurement you want to use isn't in the common format, convert it!
We can help.
- **Be brave:**
 - Use HepMC3, use ProSelecta, use a container if you're worried about building the software
 - Get outside your comfort zone, that's where the useful data is and will continue to be

How to Get Involved

Expect

- P

- E

Theo

- H

- V

- E

```
picker24/pittpacbox:root $ dnf install root
Last metadata expiration check: 0:00:13 ago on Fri Oct 25 07:55:57 2024.
Dependencies resolved.
=====
Package                               Architecture      Version           Size
-----
Installing:
root                                   aarch64           6.32.06-1.e19    156 k
Installing dependencies:
annobin                                aarch64           12.31-2.e19      1.0 M
dwz                                     aarch64           0.14-3.e19       125 k
efi-srpm-macros                         noarch            6-2.e19_0.0.1    21 k
file                                    aarch64           5.39-16.e19      48 k
fonts-srpm-macros                       noarch            1:2.0.5-7.e19.1  27 k
gcc-plugin-annobin                      aarch64           11.4.1-3.e19.alma.1 43 k
ghc-srpm-macros                          noarch            1.5.0-6.e19      7.8 k
glibc-gconv-extra                       aarch64           2.34-100.e19_4.3 1.6 M
gnu-free-fonts-common                   noarch            20120503-25.e19  122 k
gnu-free-mono-fonts                     noarch            20120503-25.e19  414 k
gnu-free-sans-fonts                      noarch            20120503-25.e19  740 k
gnu-free-serif-fonts                    noarch            20120503-25.e19  1.4 M
go-srpm-macros                           noarch            3.2.0-3.e19      26 k
json-devel                               aarch64           3.11.3-1.e19     156 k
kernel-srpm-macros                       noarch            1.0-13.e19       15 k
lua-srpm-macros                          noarch            1-6.e19           8.4 k
ocaml-srpm-macros                        noarch            6-6.e19           7.7 k
openblas-srpm-macros                     noarch            2-11.e19          7.3 k
perl-srpm-macros                          noarch            1-41.e19          8.1 k
pyproject-srpm-macros                     noarch            1.12.0-1.e19     12 k
python-srpm-macros                       noarch            3.9-53.e19       17 k
qt5-srpm-macros                           noarch            5.15.9-1.e19     7.8 k
redhat-rpm-config                         noarch            207-1.e19.alma.1 66 k
root-cling                                aarch64           6.32.06-1.e19    27 M
root-core                                 aarch64           6.32.06-1.e19    68 M
=====
```

our

ert it!

- Get outside your comfort zone, that's where the useful data is and will continue to be

Linky Dinky Dump

- ProSelecta: <https://github.com/NUISANCEMC/ProSelecta>
- NuHepMC:
 - HepMC3: <https://gitlab.cern.ch/hepmc/HepMC3>
 - Specification: <https://arxiv.org/pdf/2310.13211>
 - C++/python tools: <https://github.com/NuHepMC/cpputils>
- HEPData:
 - Example HEPData Record: <https://www.hepdata.net/record/sandbox/1713917002>
 - HEPData YAML format: https://hepdata-submission.readthedocs.io/en/latest/submission_yaml.html
 - HEPData python tools: https://github.com/HEPData/hepdata_lib/
 - NUISANCE conventions and tools: <https://github.com/NUISANCEMC/HEPData>
 - Some example of converted neutrino data sets: https://github.com/NUISANCEMC/neutrino_data
- NUISANCE slack: <https://www.nuisance-xsec.slack.com>

//What I Didn't Cover

- Standards for releasing and using data and MC events from experiments
 - c.f. MINERvA data-preservation effort
 - Fantastic effort, I think the problem domain isn't yet fully explored, so the solution requires more work. That should be done!
 - Here's we're talking about easy things that we should just agree how to do and do them

Summary

- The problem domain is clear
 - No new research is needed
- We have sketched a solution that seems promising:
 - Unless a better one appears, we will work on converting the historical corpus of measurements to this format for use in NUISANCE3
- Engagement will improve the tools, reduce your cognitive load, and improve your productivity if you produce or use scattering data:
 - There is some upfront cost, but I think the increased productivity and reduction in scope for bugs/misinterpretation is worth it
 - We are committed to writing training material, supporting the tools and users using the tools
 - If you have ideas for training materials, please let us know: API docs, readthedocs, jupyter examples, youtube videos of Luke programming...

Summ

<https://www.youtube.com/watch?v=YnL9vAFphmE>

I like this



ai you know what that is i can give you a quick tutorial

- The
 - M
- We h
 - U
 - r
- Enga
 - V

tools

- If you have ideas for training materials, please let us know: API docs, readthedocs, jupyter examples, youtube videos of Luke programing...

Summary

- The problem domain is clear
 - No new research is needed
- We have sketched a solution that seems promising:
 - Unless a better one appears, we will work on converting the historical corpus of measurements to this format for use in NUISANCE3
- Engagement will improve the tools, reduce your cognitive load, and improve your productivity if you produce or use scattering data:
 - There is some upfront cost, but I think the increased productivity and reduction in scope for bugs/misinterpretation is worth it
 - We are committed to writing training material, supporting the tools and users using the tools
 - If you have ideas for training materials, please let us know: API docs, readthedocs, jupyter examples, youtube videos of Luke programming...



Backup
