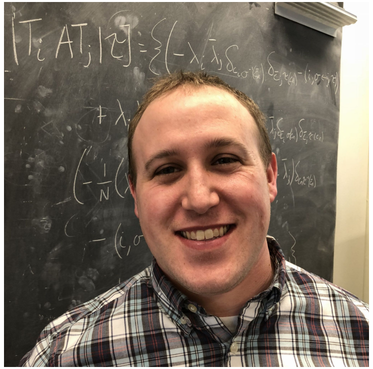
NuHepMC:

A standardized event record format for neutrino event generators



Luke Pickering, S. Gardiner, J. Isaacson

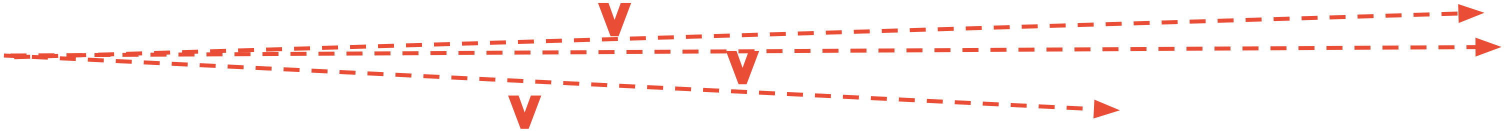
WCLG/HSF Workshop 2024, DESY, Hamburg

Science and Technology Facilities Council

Supported by URF\R1\211661

THE ROYAL SOCIETY

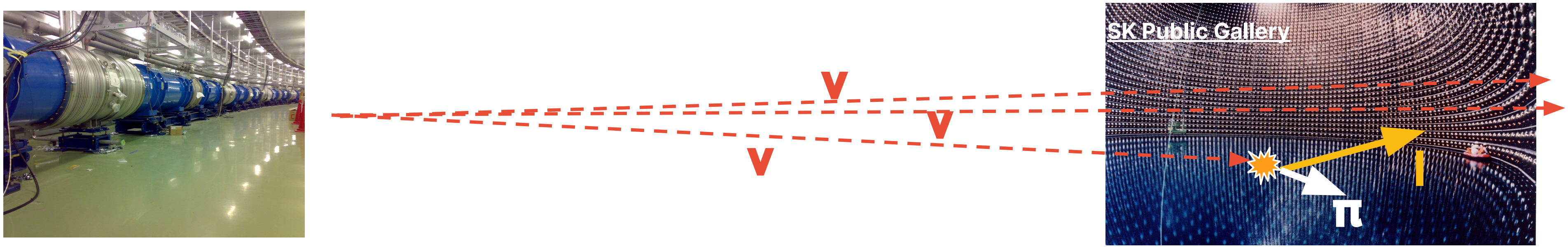
Anatomy of an Oscillation Experiment



Neutrino Source Φ

1. Find or make a source of neutrinos

Anatomy of an Oscillation Experiment



Neutrino Source Φ

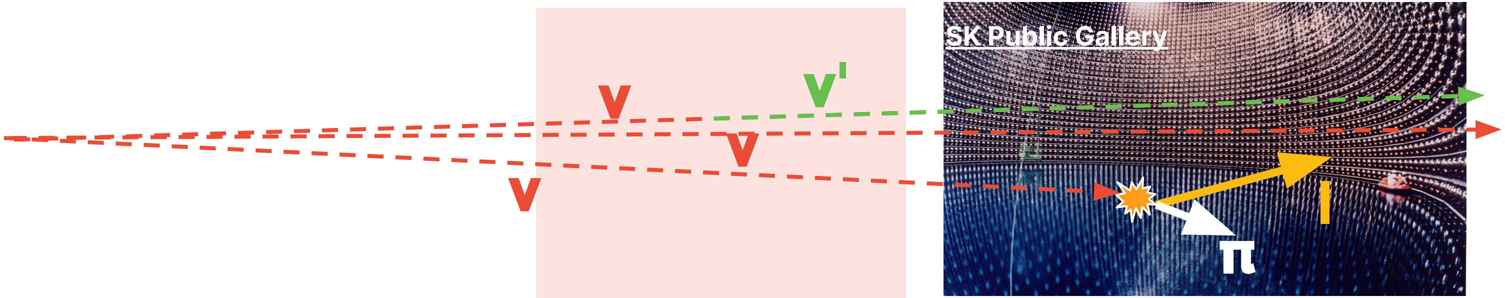
Detector

$$N(E_{\mathrm{obs}}) = \int dE_\nu \, \Phi(E_\nu) \cdot \sigma(E_\nu) \cdot \mathrm{D}$$

1. Find or make a source of neutrinos
2. …
3. Predict the expected rate with a flux/cross-section/detector model
4. Look in your detector/box…

Anatomy of an Oscillation Experiment

Neutrino Source Φ



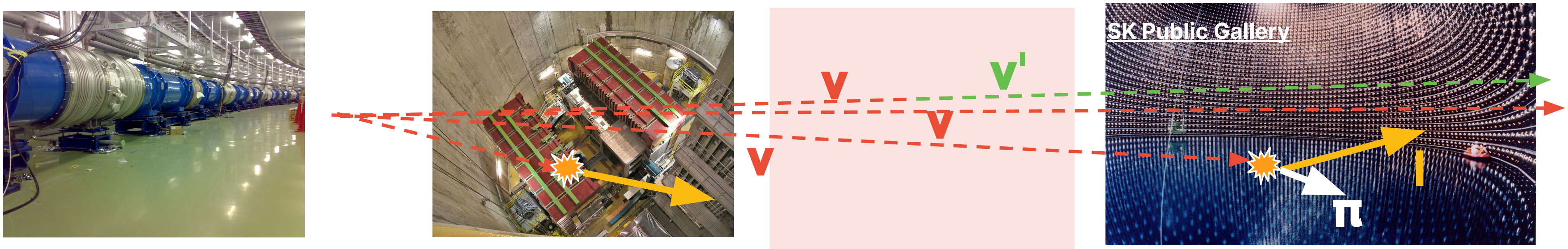
Oscillate!?!?

Detector

$$N(E_{\mathrm{obs}}) = \int dE_\nu \, \Phi(E_\nu) \cdot P_{osc}(E_\nu) \cdot \sigma(E_\nu) \cdot \mathrm{D}$$

1. Find or make a source of neutrinos
2. ...
3. Predict the expected rate with a flux/cross-section/detector model
4. Look in your detector/box... **See flavor change?**

Anatomy of an Oscillation Experiment



Neutrino Source Φ

***Near* Detector**

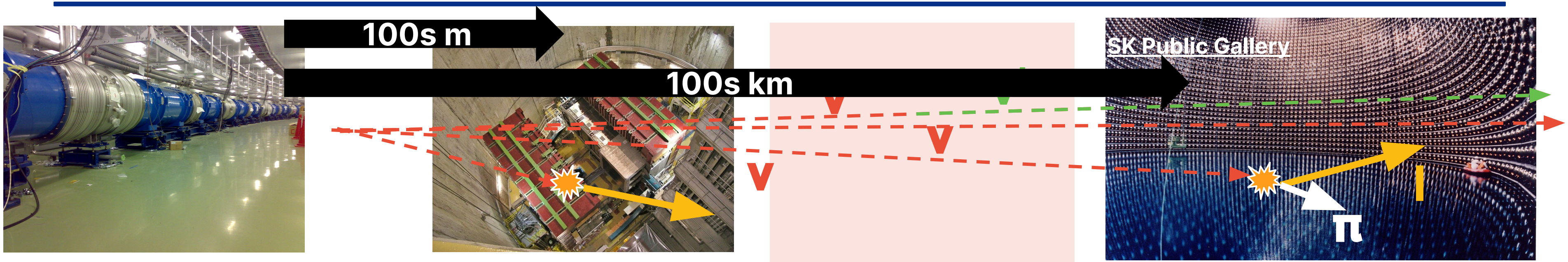
Oscillate!?!?

***Far* Detector**

$$N(E_{\mathrm{obs}}) = \int dE_\nu \, \Phi(E_\nu) \cdot P_{osc}(E_\nu) \cdot \sigma(E_\nu) \cdot \mathrm{D}$$

1. Find or make a source of neutrinos
2. **Constrain model uncertainties before oscillation with *Near* Detector**
3. Predict the expected rate with a flux/cross-section/detector model
4. Look in your detector/box... **See flavor change?**

Anatomy of an Oscillation Experiment



Neutrino Source Φ

Near Detector

Oscillate!?!?

Far Detector

$$N(E_{\mathrm{obs}}) = \int dE_\nu \, \Phi(E_\nu) \cdot P_{osc}(E_\nu) \cdot \sigma(E_\nu) \cdot \mathrm{D}$$

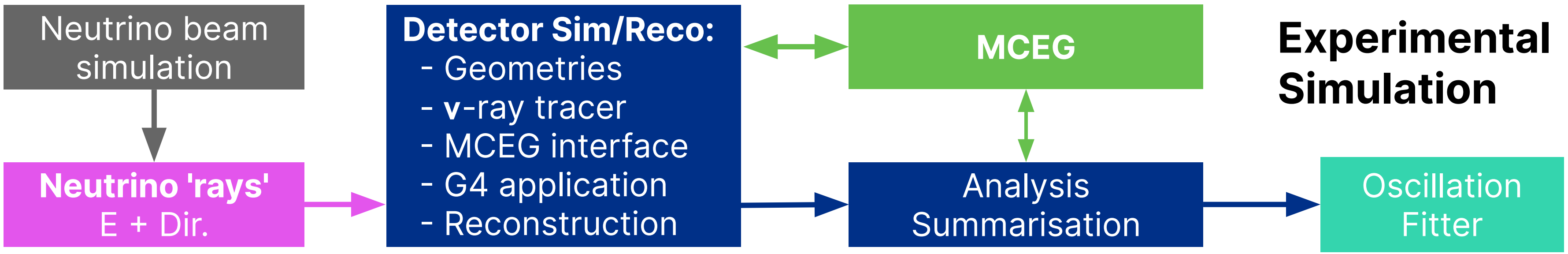
1. Find or make a source of neutrinos
2. **Constrain model uncertainties before oscillation with *Near* Detector**
3. Predict the expected rate with a flux/cross-section/detector model
4. Look in your detector/box... **See flavor change?**

Disclaimer

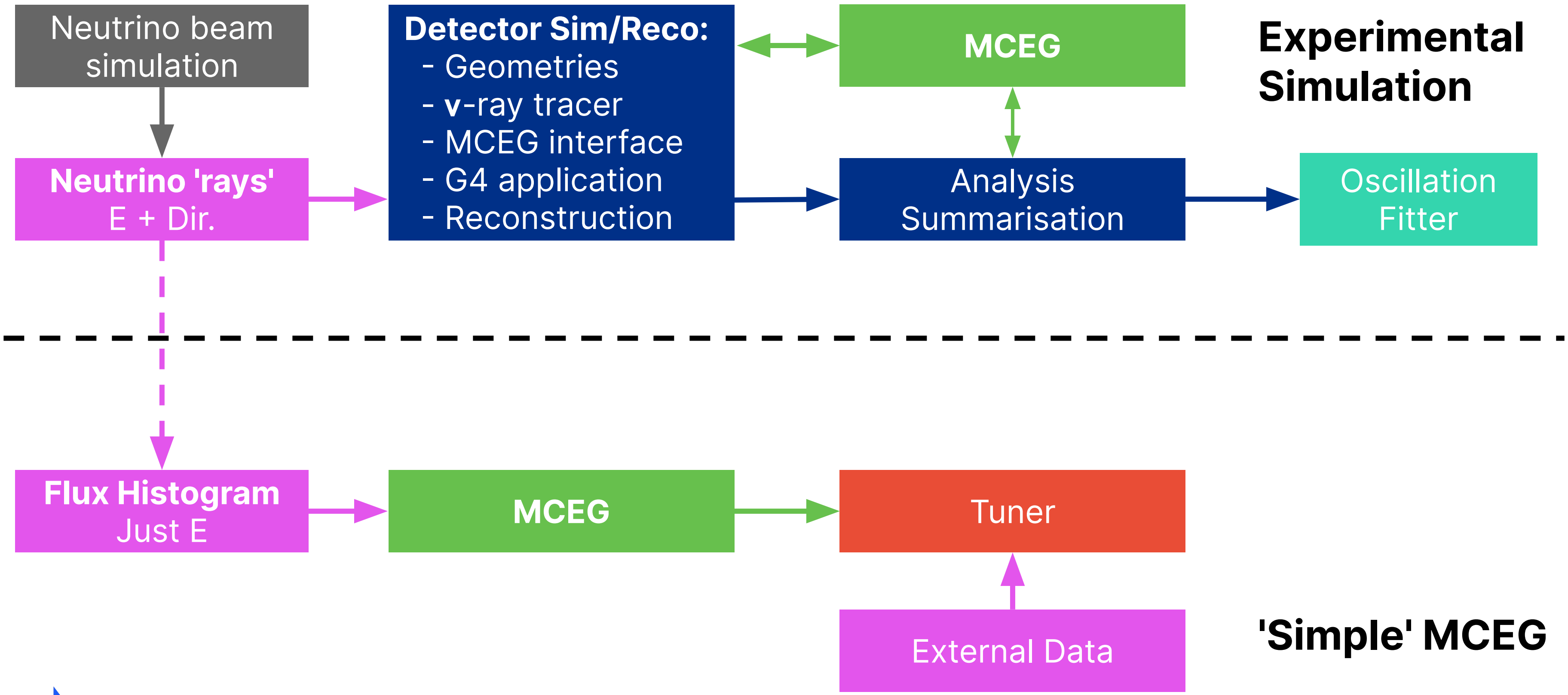
- Aware that this talk will be preaching to the choir
- In Neutrino MCEG's seeing problems that LHC software addressed 10-20 years ago
  - Seems sensible for us to build on those solutions...
- I have perspectives/biases:
  - Neutrino–Nucleus interaction systematic uncertainties for long-baseline oscillation experiments

The Problem

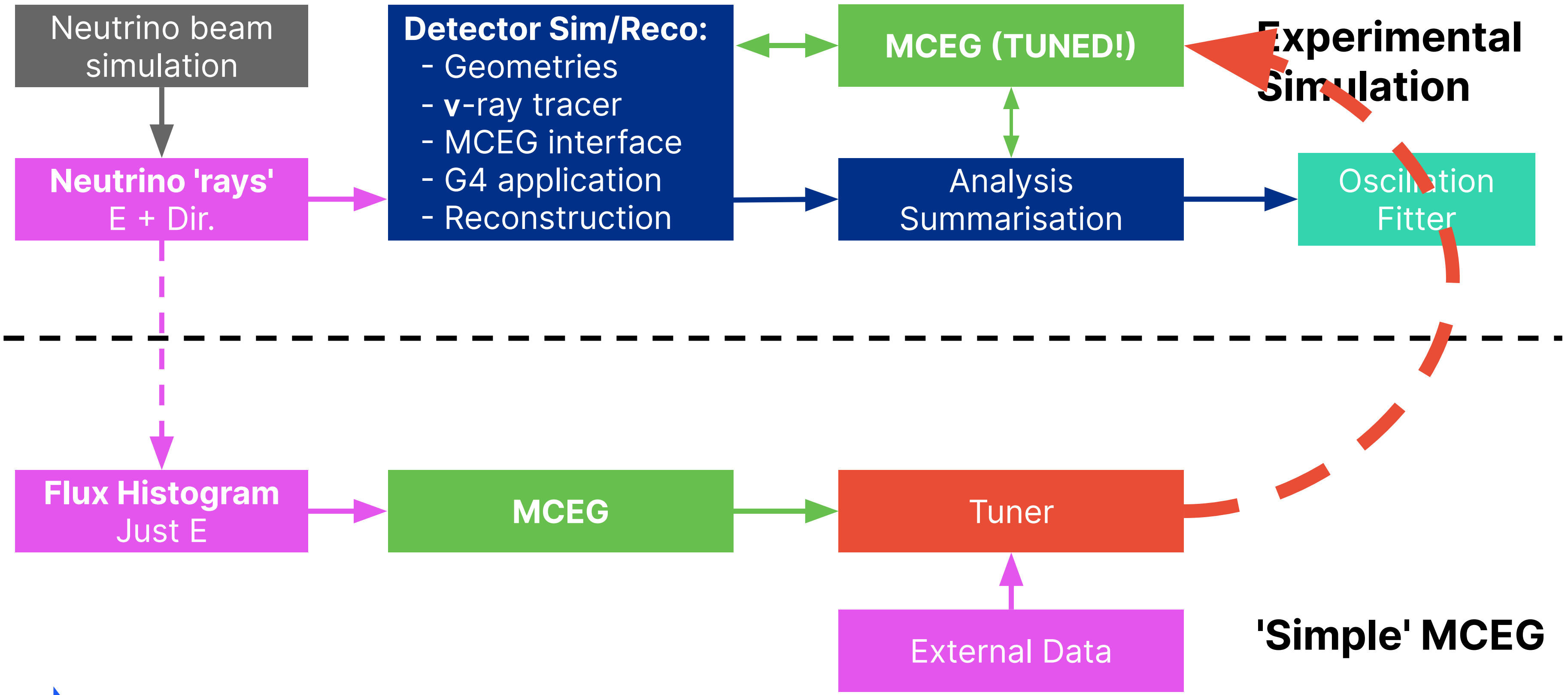
Anatomy of an Oscillation *Analysis*



Anatomy of an Oscillation *Analysis*



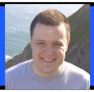
Anatomy of an Oscillation *Analysis*



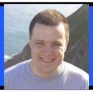
State of the ν-tion: MCEG Format Edition

Generator Name	Vector Format	Comments
NEUT	object TTree, 'flat' TTree	- Mostly F77, Pythia5, CERNLIB… - T2K/HK stacks built on NEUT 'flat' TTree - Event reweighting, flux/det. geom tools
Achilles	HepMC3, NuHepMC	- C++17, interfaces with LHC MCEGs, e.g. SHERPA
GENIE	object TTree, 'flat' TTree, XML	- FNAL **ν** stacks built on GENIE object TTree - Event reweighting, flux/det. geom tools
NuWro	ROOT 'flat' tree	- Provides flux/geometry drivers
GiBUU	LHA XML, 'flat' TTree	- Sophisticated treatment of hadron transport
Marley	NuHepMC	- Argon-target 'low-energy' MCEG

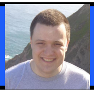
State of the ν-tion: MCEG Format Edition

Generator Name	Vector Format	Comments
NEUT 	object TTree, 'flat' TTree	- Mostly F77, Pythia5, CERNLIB… - T2K/HK stacks built on NEUT 'flat' TTree - Event reweighting, flux/det. geom tools
Achilles 	HepMC3, NuHepMC	- C++17, interfaces with LHC MCEGs, e.g. SHERPA
GENIE 	object TTree, 'flat' TTree, XML	- FNAL **ν** stacks built on GENIE object TTree - Event reweighting, flux/det. geom tools
NuWro	ROOT 'flat' tree	- Provides flux/geometry drivers
GiBUU	LHA XML, 'flat' TTree	- Sophisticated treatment of hadron transport
Marley 	NuHepMC	- Argon-target 'low-energy' MCEG

State of the ν-tion: MCEG Format Edition

Generator Name	Vector Format	Comments
NEUT 	object TTree, 'flat' TTree	- [redacted] - T2K/HK stacks built on NEUT 'flat' TTree - Event reweighting, flux/det. geom tools
Achilles 	HepMC3, NuHepMC	- C++17, interfaces with LHC MCEGs, e.g. SHERPA
GENIE 	object TTree, 'flat' TTree, XML	- FNAL **ν** stacks built on GENIE object TTree - Event reweighting, flux/det. geom tools
NuWro	ROOT 'flat' tree	- Provides flux/geometry drivers
GiBUU	LHA XML, 'flat' TTree	- Sophisticated treatment of hadron transport
Marley 	NuHepMC	- Argon-target 'low-energy' MCEG

State of the ν-tion: MCEG Format Edition

Generator Name	Vector Format	Comments
NEUT 	object TTree, 'flat' TTree	- [redacted] **CERNLIB2023!** - T2K/HK stacks built on NEUT 'flat' TTree -
Achilles 	HepMC3, NuHepMC	- S
GENIE 	object TTree, 'flat' TTree, XML	- ee - Event reweighting, flux/det. geom tools
NuWro	ROOT 'flat' tree	- Provides flux/geometry drivers
GiBUU	LHA XML, 'flat' TTree	- Sophisticated treatment of hadron transport
Marley 	NuHepMC	- Argon-target 'low-energy' MCEG

Thanks Andrii and CERNLIB preservation team for materially improving my life

The Problem

- We have 7 different ROOT formats in *common* use
    - 2 experimental MCEG x ('simple', experiment sim), +NuWro, +GiBUU, +NUISANCE
    - **Completely non-interoperable**
    - Every analysis needs custom code and MCEG expertise per MCEG.

A very common task that is currently a ***utter*** pain in the neck:

- Make a correctly-normalized generator prediction for your measurement with multiple generators for your publication

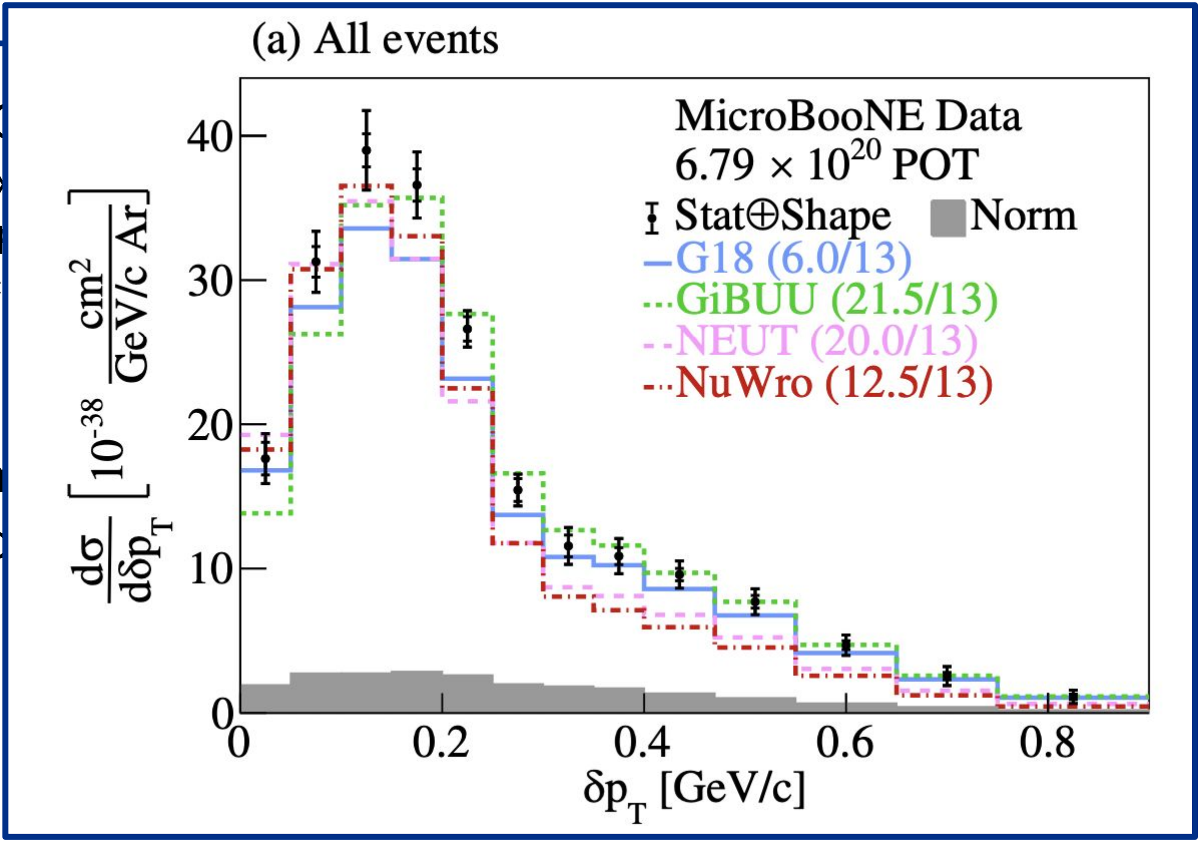
The Problem

PRD 108, 053002 (2023)

- We have 7 different R(
  - 2 experimental MCEG x
  - Completely non-interop
  - Every analysis needs c

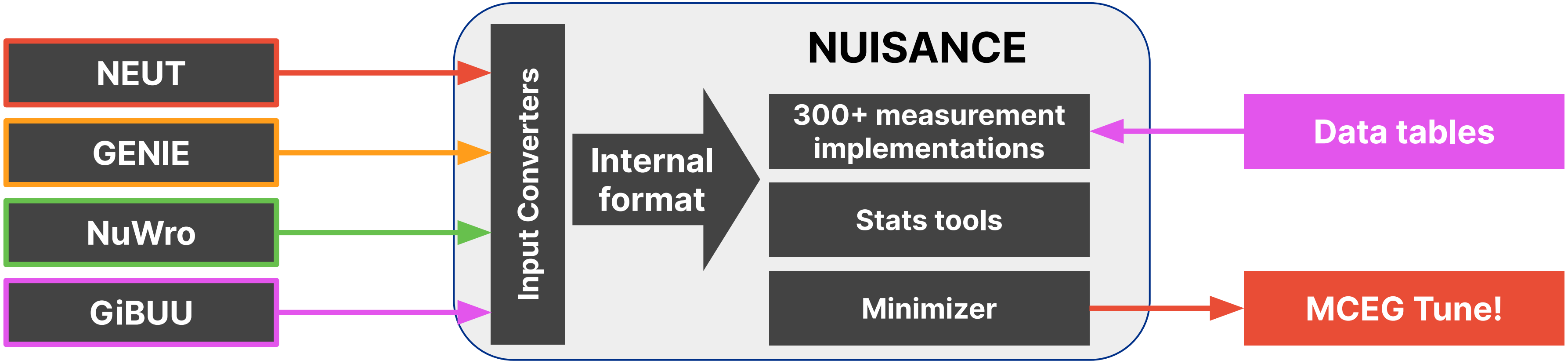
A very common task that i

- Make a correctly-norm
  with multiple generato



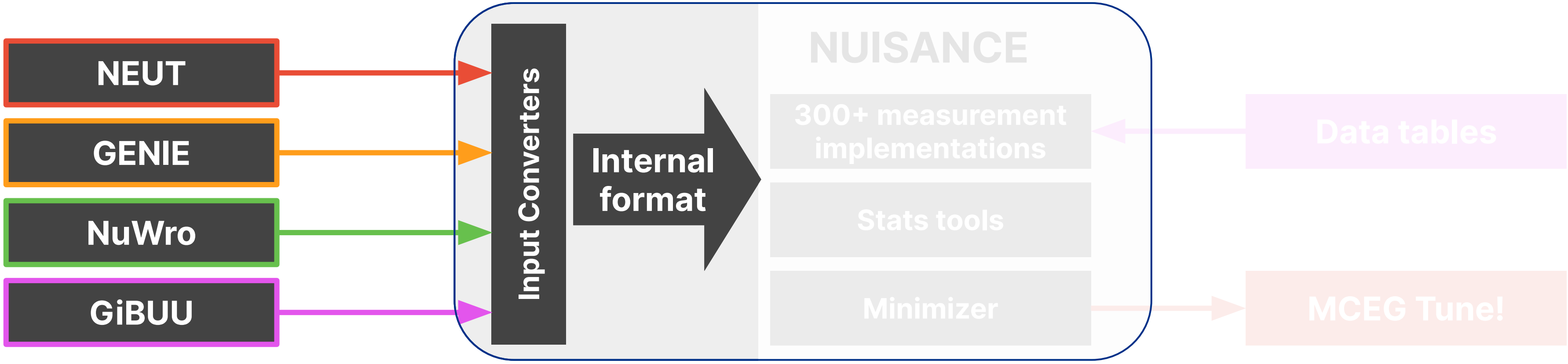
NUISANCE: A Case Study

- NUISANCE: a multi-MCEG data-comparison and tuning framework
  - Format converters 'in house'
  - Widely used by the community...



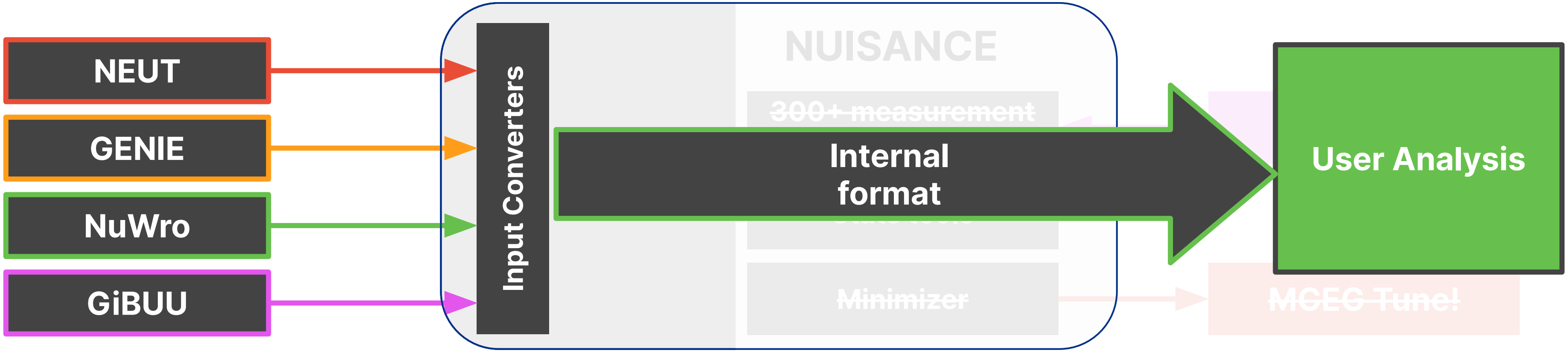
NUISANCE: A Case Study

- NUISANCE: a multi-MCEG data-comparison and tuning framework
  - Format converters 'in house'
  - Widely used by the community… but mostly not for its original purpose
  - Very significant fraction of users, code, maintenance is **just** making cross-section predictions from different MCEGs



NUISANCE: A Case Study

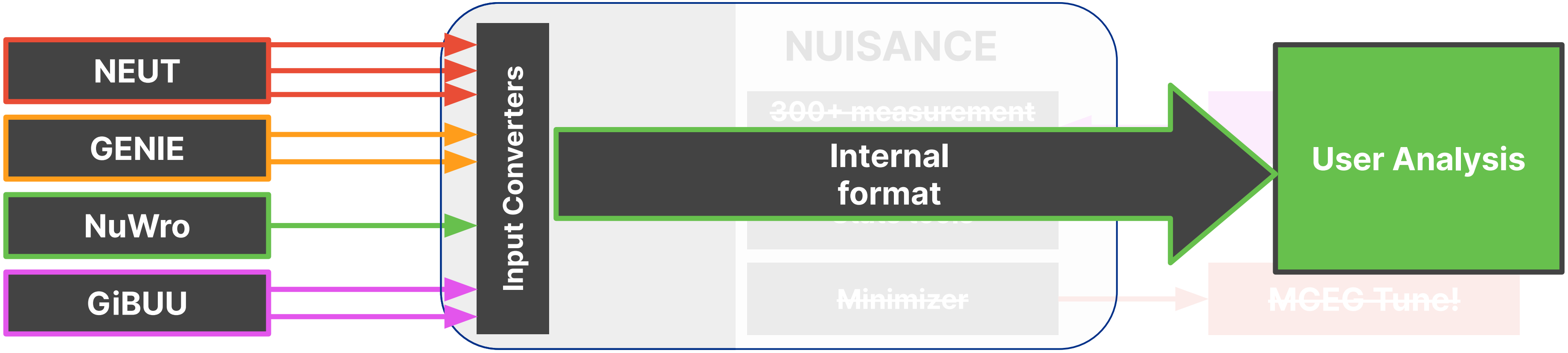
- NUISANCE: a multi-MCEG data-comparison and tuning framework
  - Format converters 'in house'
  - Widely used by the community... but mostly not for its original purpose
  - Very significant fraction of users, code, maintenance is **just** making cross-section predictions from different MCEGs



NUISANCE: A Case Study

- NUISANCE: a r... ...nework
  - Format conve...
  - Widely used ...
  - Very significa...
    cross-section...

Scaling law is worse than ~NMCEGs due to breaking changes/different formats



NUISANCE: A Case Study

NuHepMC is one part of how NUISANCE is trying to get back into the **Physics** game, rather than the 'maintaining MCEG's event converters for them' game...

For more: see Patrick's talk on Thursday

The Solution

It is mechanically trivial...

The Solution

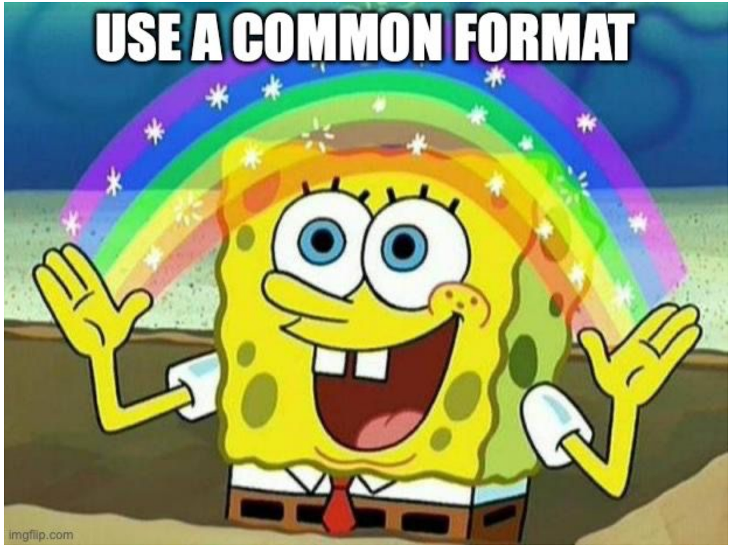
It is mechanically trivial...

but sociologically tricky

The Solution

It is mechanically trivial...

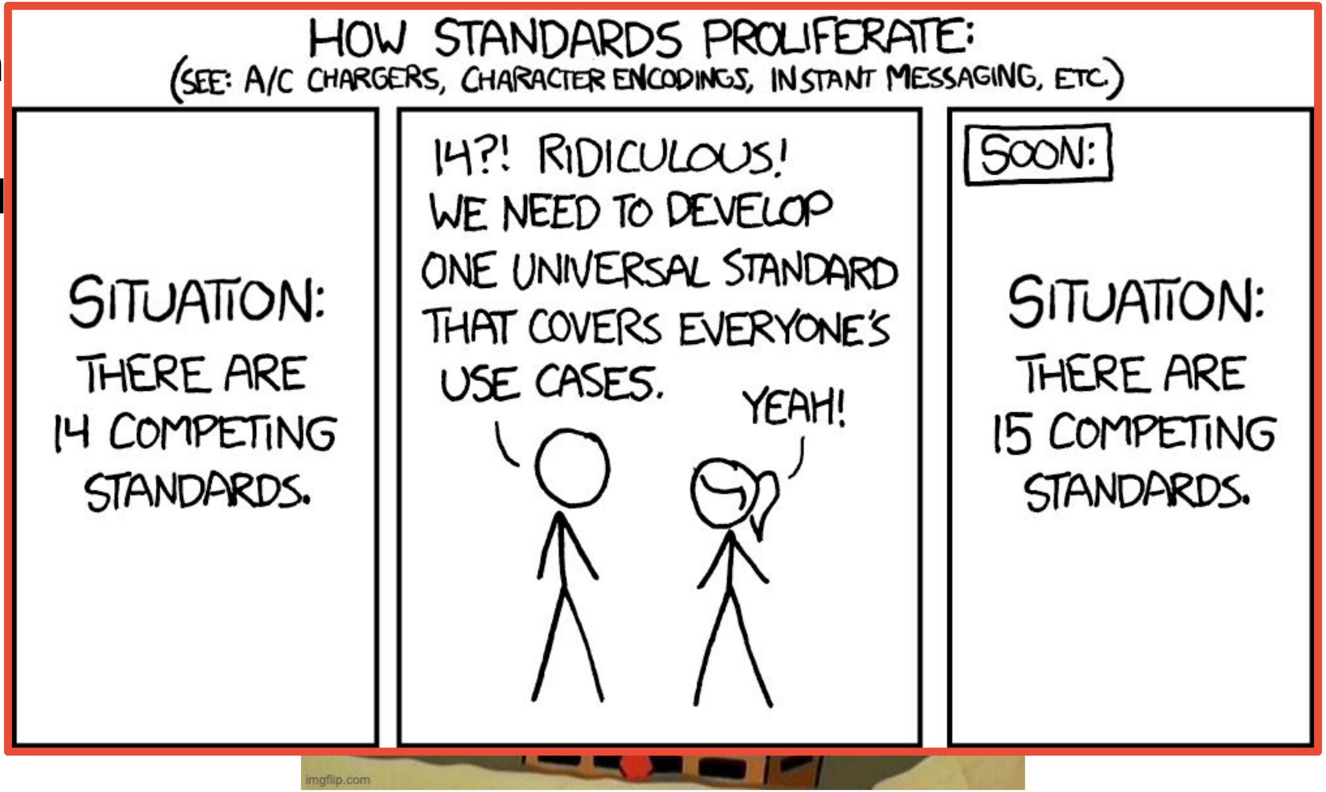
but sociologically tricky



The Solution

It is mech

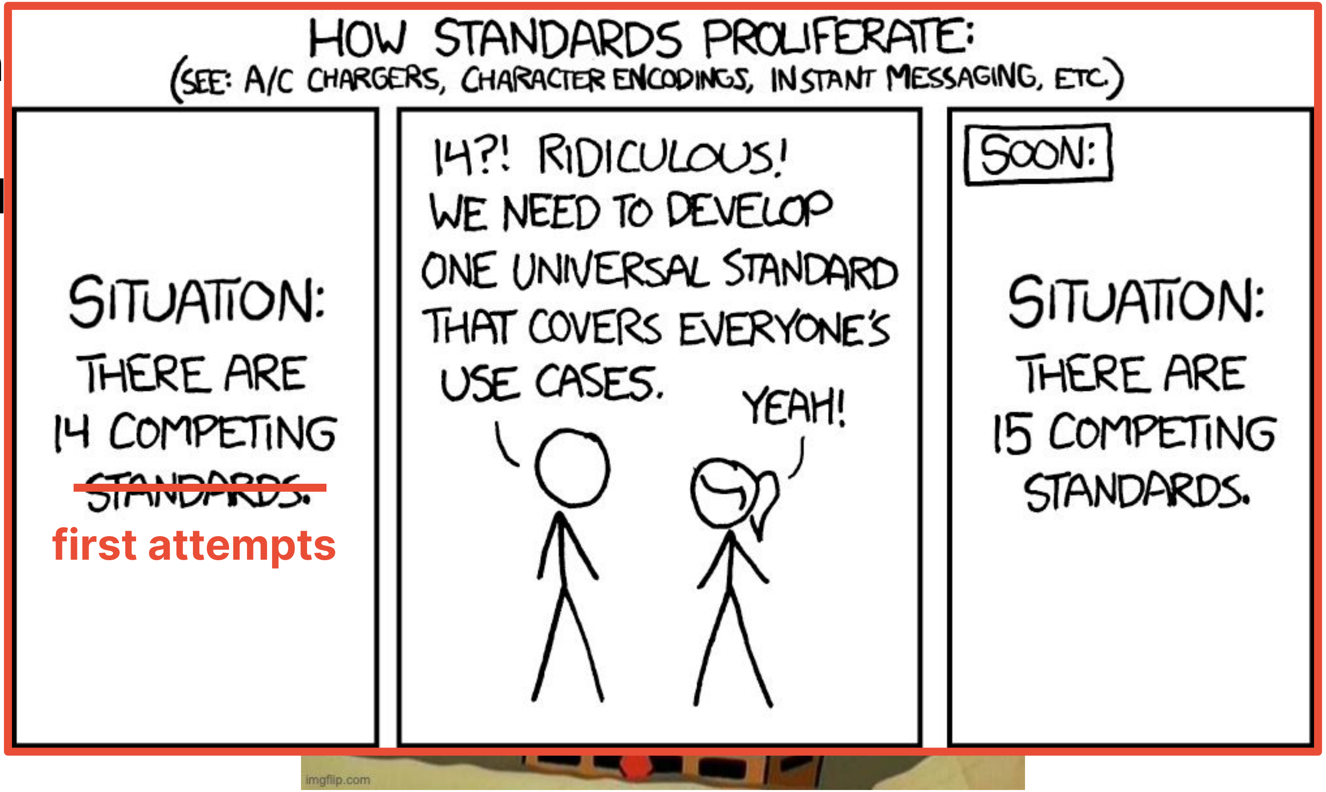
but sociol



The Solution

It is mech

but sociol



The Solution

Generator Name	Vector Format	Comments
NEUT	object TTree, 'flat' TTree	- Mostly F77, Pythia5, CERNLIB… - T2K/HK stacks built on NEUT 'flat' TTree - Event reweighting, flux/det. geom tools
Achilles	HepMC3, NuHepMC	- C++17, interfaces with LHC tools, e.g. SHERPA
GENIE	object TTree, 'flat' TTree, XML	- FNAL **ν** stacks built on GENIE object TTree - Event reweighting, flux/det. geom tools
NuWro	ROOT 'flat' tree	- Provides flux/geometry drivers
GiBUU	LHA XML, 'flat' TTree	- Sophisticated treatment of hadron transport
Marley	NuHepMC	- Argon-target 'low-energy' MCEG

High-level Requirements For A Common Format

- **Loose coupling:**
  - Should not need a copy of the MCEG binary to parse an event vector
- **Extensible:**
  - We can't pre-specify all information that future MCEGs might want to communicate to users
- **Self-describing:**
  - Should not need a copy of the MCEG source code to interpret an event vector
  - Process identifiers, custom particle statuses

Some Design Assumptions Before Diving In!

- A solution that cannot fully encode MCEG information in existing formats will not get adopted:
    - Generator-specific dialects may evolve after adoption: expected, good!

Some Design Assumptions Before Diving In!

- A solution that cannot fully encode MCEG information in existing formats will not get adopted:
    - Generator-specific dialects may evolve after adoption: expected, good!
- We can make multi-MCEG event processing simpler and more flexible by agreeing on common interfaces **without losing any functionality**

Some Design Assumptions Before Diving In!

- A solution that cannot fully encode MCEG information in existing formats will not get adopted:
  - Generator-specific dialects may evolve after adoption: expected, good!
- We can make multi-MCEG event processing simpler and more flexible by agreeing on common interfaces **without losing any functionality**
- Such a common interface must not constrain potential modelling choices

Some Design Assumptions Before Diving In!

- A solution that cannot fully encode MCEG information in existing formats will not get adopted:
  - Generator-specific dialects may evolve after adoption: expected, good!
- We can make multi-MCEG event processing simpler and more flexible by agreeing on common interfaces **without losing any functionality**
- Such a common interface must not constrain potential modelling choices
- The simplest solution that needs the least maintenance is the best

Some Design Assumptions Before Diving In!

- A solution that cannot fully encode MCEG information in existing formats will not get adopted:
    - Generator-specific dialects may evolve after adoption: expected, good!
- We can make multi-MCEG event processing simpler and more flexible by agreeing on common interfaces **without losing any functionality**
- Such a common interface must not constrain potential modelling choices
- The simplest solution that needs the least maintenance is the best
- Event *parsing* is not the computational bottleneck in any compute-intensive tasks (e.g. experiment simulation).
    - If a boatload of extensibility makes event reading off disk a little slower, this isn't a problem

NuHepMC

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
  - All NuHepMC events are valid HepMC3 events, a strict subset.
  - **Not restricted to neutrino probes, have electron probe users, expect nucleon and pion probe analyses!**

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
  - All NuHepMC events are valid HepMC3 events, a strict subset.
  - **Not restricted to neutrino probes, have electron probe users, expect nucleon and pion probe analyses!**

Why HepMC3?

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
  - All NuHepMC events are valid HepMC3 events, a strict subset.
  - **Not restricte**
    probe analy

Why HepMC3?

High-level Requirements For A Common Format

✅ **Loose coupling:**
  - Should not need a copy of the MCEG binary to parse an event vector

- **Link to HepMC3 library: Extensively tested, maintained**

- **Self-describing:**
  - Should not need a copy of the MCEG source code to interpret an event vector
  - Process identifiers, custom particle statuses

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
  - All NuHepMC events are valid HepMC3 events, a strict subset.
  - **Not restricte
    probe analy**

Why HepMC3?

High-level Requirements For A Common Format

✅ **HepMC3::Attribute system is perfect for arbitrary run-, event-, particle-, and process-level metadata**

✅ **Extensible:**
- We can't pre-specify all information that future MCEGs might want to communicate to users

✅ **Self-describing:**
- Should not need a copy of the MCEG source code to interpret an event vector
- Process identifiers, custom particle statuses

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
    - All NuHepMC
    - **Not restricte**
      probe analys

Why HepMC3?

Some Design Assumptions Before Diving In!

✅ A solution that cannot fully encode MCEG information in existing formats will not get adopted:
  - Generator-specific dialects may evolve after adoption: expected, good!
- y

HepMC3::Attributes to the rescue again

- Such a common interface must not constrain potential modelling choices
- The simplest solution that needs the least maintenance is the best
- Event *parsing* is not the computational bottleneck in any compute-intensive tasks (e.g. experiment simulation).
    - If a boatload of extensibility makes event reading off disk a little slower, this isn't a problem

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
    - All NuHepMC
    - **Not restricte**
      probe analys

Why HepMC3?

Some Design Assumptions Before Diving In!

✅ A solution that cannot fully encode MCEG information in existing formats will not get adopted:
- Generator-specific dialects may evolve after adoption: expected, good!

- We can make multi-MCEG event processing simpler and more flexible by

HepMC3 is an existing, well maintained and supported solution with many stakeholders

- s

✅ The simplest solution that needs the least maintenance is the best

- Event *parsing* is not the computational bottleneck in any compute-intensive tasks (e.g. experiment simulation).
    - If a boatload of extensibility makes event reading off disk a little slower, this isn't a problem

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
  - All NuHepMC
  - **Not restricte**
    probe analys

Why HepMC3?

Some Design Assumptions Before Diving In!

✅ A solution that cannot fully encode MCEG information in existing formats will not get adopted:
- Generator-specific dialects may evolve after adoption: expected, good!

HepMC3 offers a 🚢 of extensibility, but relies on string parsing.
But: only parses strings of attributes that you need to read

I assert strongly that this will not be a bottleneck for ν workflows.

- Event *parsing* is not the computational bottleneck in any compute-intensive tasks (e.g. experiment simulation).
  - ✅ If a boatload of extensibility makes event reading off disk a little slower, this isn't a problem

NuHepMC

- **What is NuHepMC?** A metadata specification on top of HepMC3
  - All NuHepMC events are valid HepMC3 events, a strict subset.
  - **Not restricted to neutrino probes, have electron probe users, expect nucleon and pion probe analyses!**

Why HepMC3?

- Separation of in-memory and on-disk formats proving very useful
  - ASCII great for file preservation, low-level debugging
  - Binary great for size & de-serialization speed
  - HepMC3::Reader/Writer plugin framework is simple, extensible, very easy to use

The Specification

- Full working spec on Arxiv: https://arxiv.org/pdf/2310.13211.pdf
- More 'living' version on Github: https://github.com/NuHepMC/Spec

The Specification

- Full working spec on Arxiv: https://arxiv.org/pdf/2310.13211.pdf
- More 'living' version on Github: https://github.com/NuHepMC/Spec
- Individual specifications take the form:

  <Component>.<Category>.<Index>

  - <Component>:
    - **G:** GenRunInfo
    - **E:** Event
    - **V:** Vertex
    - **P:** Particle
  - <Category>:
    - **R:** Requirement – Must be implemented by all
    - **C:** Convention – Recommend specifications
    - **S:** Suggestion – Optional details

The Specification

- Full working spec on Arxiv: https://arxiv.org/pdf/2310.13211.pdf
- More 'living' version on Github: https://github.com/NuHepMC/Spec
- Individu

  <Con

  - <Con
  - <Cat
    - **S:** Suggestion – Optional details

Hope to get wider community engagement as/if it gets adopted:
Experimental analysers
Experimental software developers
MCEG developers
MCEG tuning groups

Specification development, if needed, expected to happen via the github (PRs, comments, etc...)

Specification Highlights

Self-Describing Event Vectors

What extra bits we need to specify:

- How to signal adherence to optional parts of the specification?
- How to communicate what 'interaction 43' means in a NEUT file?
- How to correctly normalize a prediction?

Self-Describing Event Vectors

What extra bits we need to specify:

- **How to signal adherence to optional parts of the specification?**
- How to communicate what 'interaction 43' means in a NEUT file?
- How to correctly normalize a prediction?

G.C.1 SIGNALING FOLLOWED CONVENTIONS:
To signal to a user that an implementation follows a named convention from this specification, a `HepMC3::VectorStringAttribute` should be added to the `HepMC3::GenRunInfo` instance named `"NuHepMC.Conventions"` containing the names of the conventions adhered to.

Self-Describing Event Vectors

What extra bits

- How to sig
- How to co
- How to co

```
A NuHepMC.Citations.Generator.DOI 10.1016/j.nima.2009.12.009 10.1140/epjs/s11734-021-002
95-7
A NuHepMC.Citations.Process[100].DOI 10.1103/PhysRevD.79.053003
A NuHepMC.Citations.Process[200].DOI 10.1103/PhysRevC.70.055503 10.1103/PhysRevC.72.0199
02
A NuHepMC.Citations.Process[300].DOI 10.1103/PhysRevD.91.073004
A NuHepMC.Citations.Process[400].DOI 10.1103/PhysRevD.76.113004
A NuHepMC.Citations.Process[600].DOI 10.1103/PhysRevD.50.3085 10.1103/PhysRevD.65.033002
A NuHepMC.Conventions E.C.1 E.C.2 E.C.4 E.C.5 G.C.1 G.C.4 G.C.6 G.S.2 P.C.1
A NuHepMC.ParticleStatusIDs 0 1 2 4 11 21 22 23 24 25 26 27 28 29
A NuHepMC.ParticleStatusInfo[0].Description Not meaningful
A NuHepMC.ParticleStatusInfo[0].Name Not defined
```

G.C.1 SIGNALING FOLLOWED CONVENTIONS:
To signal to a user that an implementation follows a named convention from this specification, a `HepMC3::VectorStringAttribute` should be added to the `HepMC3::GenRunInfo` instance named `"NuHepMC.Conventions"` containing the names of the conventions adhered to.

Self-Describing Event Vectors

What extra bits we need to specify:

- How to signal adherence to optional parts of the specification?
- **How to communicate what 'interaction 43' means in a NEUT file?**
- How to correctly n

G.R.4 Process Metadata:

A NuHepMC `HepMC3::GenRunInfo` instance must contain a `HepMC3::VectorIntAttribute` named `"NuHepMC.ProcessIDs"` listing all physics process IDs as integers. For each valid process ID, the `HepMC3::GenRunInfo` instance must also contain two other attributes giving a name and description of each:

- type: `HepMC3::StringAttribute`, name: `"NuHepMC.ProcessInfo[<ID>].Name"`
- type: `HepMC3::StringAttribute`, name: `"NuHepMC.ProcessInfo[<ID>].Description"`

where <ID> enumerates all process IDs present in `"NuHepMC.ProcessIDs"`. (See also E.C.1).

Self-Describing Event Vectors

```
A NuHepMC.ProcessIDs 276 275 675 488 487 486 485 575 490 489 175 185 479 478 476 475 625
 436 435 525 437 125 135 427 426 425 325 225 200 300 400 401 402 110 100 412 500 410 411
 600 450 451 452 452 160 150 464 465 550 460 461 462 463 601 250 251
A NuHepMC.ProcessInfo[100].Description neutmode=16
A NuHepMC.ProcessInfo[100].Name CC_COH_nu
A NuHepMC.ProcessInfo[110].Description neutmode=15
A NuHepMC.ProcessInfo[110].Name CC_DIF_nu
A NuHepMC.ProcessInfo[125].Description neutmode=-16
A NuHepMC.ProcessInfo[125].Name CC_COH_nubar
A NuHepMC.ProcessInfo[135].Description neutmode=-15
A NuHepMC.ProcessInfo[135].Name CC_DIF_nubar
A NuHepMC.ProcessInfo[150].Description neutmode=36
A NuHepMC.ProcessInfo[150].Name NC_COH_nu
A NuHepMC.ProcessInfo[160].Description neutmode=35
A NuHepMC.ProcessInfo[160].Name NC_DIF_nu
```

?

contain a
s" listing
ess ID, the
ributes giv-

- type: HepMC3::StringAttribute,
  name: "NuHepMC.ProcessInfo[<ID>].Description"

where <ID> enumerates all process IDs present in "NuHepMC.ProcessIDs". (See also E.C.1).

Self-Describing Event Vectors

What extra bits we need to specify:

- How to signal adherence to optional parts of the specification?
- How to communicate what 'interaction 43' means in a NEUT file?
- **How to correctly normalize a prediction?**

Key Goal: If a user has an event vector file, they should be able to make correctly normalized predictions with it and retrieve provenance information

- Currently that is categorically untrue. User's often need to know the provenance of a given file to even open it.
- **NuHepMC:** Breaks tight coupling to MCEG binaries/source/versions!

Key Example: Predicting a Cross Section

$$N\quad(E_{\mathrm{obs}}) = \int dE_\nu\, \Phi\quad(E_\nu) \qquad\qquad \cdot\, \sigma\,(E_\nu)$$

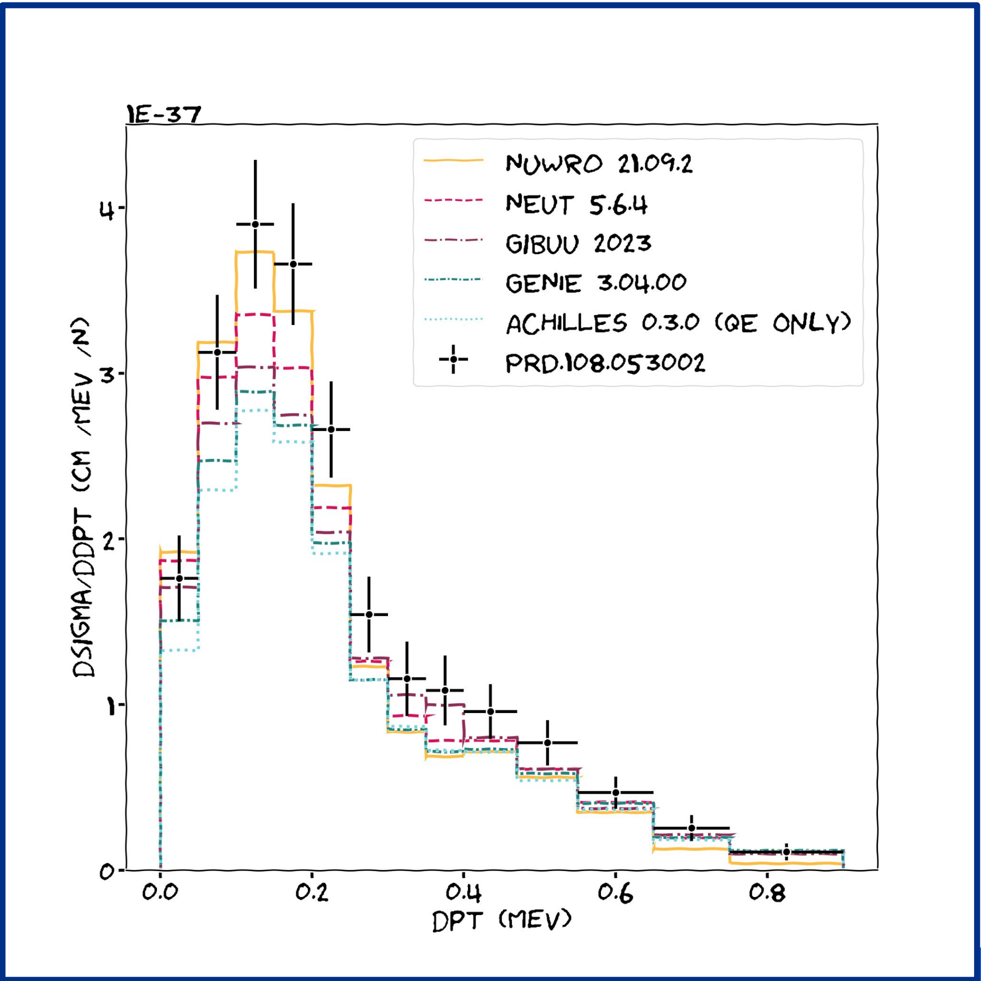
- MCEGs perform above integration numerically
    - Ask MCEG for 1E6 events, or 1E21 POT exposure... but what if you need to know the cross-section?
        - Different MCEGs provide cross-section scaling information in fundamentally different ways due to different numerical approaches to the above
        - Need to provide **flux-averaged total-cross section**: ~equivalent to total cross-section for collider MCEGs.

Key Example: Predictin

$$N \quad (E_{\mathrm{obs}}) = \int dE_\nu \; \Phi \quad (E$$

As far as we know, this is the first neutrino measurement prediction made directly with MCEG output without caring which MCEG produced which output

This might seem minor... but I hope I've given you the context to emphasise that I don't think it will be for our users



Key Example: Flux Descriptions

$$N\left(E_{\text{obs}}\right) = \int dE_{\nu}\, \Phi\left(E_{\nu}\right) \cdot P_{osc}\left(E_{\nu}\right) \cdot \sigma\left(E_{\nu}\right) \cdot \mathbf{D}$$

- MCEGs perform above integration numerically
  - Different predictions will use different experimental fluxes
  - Universally useful to have neutrino flux provenance information stored in a vector
- Neutrino energy spectrum plays a similar role to PDFs for collider MCEGs
  - But a number of subtleties...
    - Both energy distributions and mono-energetic fluxes are used for truth studies
    - Flux predictions presented in different formats and MCEGs expect different formats (/bin width or not)

Key Example: Flux Desc

$$N\quad(E_{\text{obs}}) = \int dE_{\nu}\,\Phi\quad(E$$

Store as bin edges and content

Trivial to convert back to histogram format of your choosing


```
A NuHepMC.Beam.EnergyUnit MEV
A NuHepMC.Beam.RateUnit #frac{Number of #nu_{#mu}}{1.592 #upoint 10^{20} POT cm^{2} 0.05
 GeV}
A NuHepMC.Beam.Type Histogram
A NuHepMC.Beam[14].Histogram.BinContent 24627200.000000 94533000.000000 122210496.000000
 145618000.000000 180872496.000000 199318496.000000 213119008.000000 223194496.000000 23
6011488.000000 246111008.000000 251703504.000000 252103984.000000 248851504.000000 24629
3504.000000 243755008.000000 238064512.000000 229157984.000000 219608496.000000 21006249
6.000000 198866000.000000 188562000.000000 176640496.000000 164375504.000000 152730000.0
00000 141608496.000000 128901504.000000 117326000.000000 105118000.000000 92988000.00000
0 81045496.000000 70605496.000000 60962504.000000 51810000.000000 43676452.000000 369121
52.000000 30574598.000000 25366850.000000 20513850.000000 16732200.000000 13520550.00000
0 10791800.000000 8823650.000000 7267900.000000 5763500.000000 4954480.000000 4263865.00
0000 3693304.750000 3221545.000000 2926920.000000 2653900.000000 2452360.000000 2303495.
000000 2212870.000000 2144590.000000 2055755.000000 1913940.000000 1887230.000000 182959
5.000000 1794625.000000 1670220.000000 1602450.000000 1550370.000000 1589250.000000 1485
175.000000 1439095.000000 1405175.000000 1335075.000000 1329055.000000 1275775.000000 12
41390.000000 1221660.000000 1142725.000000 1109855.000000 1053900.000000 1030610.000000
1002905.000000 909215.000000 874435.000000 833995.000000 822615.000000 777285.000000 767
800.000000 696840.000000 670190.000000 622995.000000 598625.000000 592985.000000 569185.
000000 545385.000000 527910.000000 458068.500000 461064.500000 432699.500000 398820.0000
00 387983.000000 359204.000000 336549.000000 320215.500000 303928.500000 271809.000000 2
62296.000000 246877.500000 232074.000000 222882.000000 206728.500000 199420.000000 17531
8.500000 167569.500000 160968.000000 170336.500000 139083.500000 145643.500000 146479.50
0000 108922.000000 96823.000000 95441.000000 86009.500000 77651.000000 70785.500000 6382
8.000000 59740.000000 50549.000000 48476.699219 40512.250000 32961.699219 27337.550781 2
5562.449219 19584.099609 15100.049805 13846.700195 10259.000000 8117.049805 6200.500000
4114.879883 1916.519897 0.000000
A NuHepMC.Beam[14].Histogram.BinEdges 0.000000 50.000000 100.000000 150.000000 200.00000
0 250.000000 300.000000 350.000000 400.000000 450.000000 500.000000 550.000000 600.00000
0 650.000000 700.000000 750.000000 800.000000 850.000000 900.000000 950.000000 1000.0000
00 1050.000000 1100.000000 1150.000000 1200.000000 1250.000000 1300.000000 1350.000000 1
400.000000 1450.000000 1500.000000 1550.000000 1600.000000 1650.000000 1700.000000 1750.
000000 1800.000000 1850.000000 1900.000000 1950.000000 2000.000000 2050.000000 2100.0000
00 2150.000000 2200.000000 2250.000000 2300.000000 2350.000000 2400.000000 2450.000000 2
500.000000 2550.000000 2600.000000 2650.000000 2700.000000 2750.000000 2800.000000 2850.
000000 2900.000000 2950.000000 3000.000000 3050.000000 3100.000000 3150.000000 3200.0000
00 3250.000000 3300.000000 3350.000000 3400.000000 3450.000000 3500.000000 3550.000000 3
600.000000 3650.000000 3700.000000 3750.000000 3800.000000 3850.000000 3900.000000 3950.
000000 4000.000000 4050.000000 4100.000000 4150.000000 4200.000000 4250.000000 4300.0000
00 4350.000000 4400.000000 4450.000000 4500.000000 4550.000000 4600.000000 4650.000000 4
700.000000 4750.000000 4800.000000 4850.000000 4900.000000 4950.000000 5000.000000 5050.
000000 5100.000000 5150.000000 5200.000000 5250.000000 5300.000000 5350.000000 5400.0000
00 5450.000000 5500.000000 5550.000000 5600.000000 5650.000000 5700.000000 5750.000000 5
800.000000 5850.000000 5900.000000 5950.000000 6000.000000 6050.000000 6100.000000 6150.
000000 6200.000000 6250.000000 6300.000000 6350.000000 6400.000000 6450.000000 6500.0000
00 6550.000000 6600.000000 6650.000000 6700.000000 6750.000000 6800.000000
A NuHepMC.Beam[14].Histogram.ContentIsPerWidth 0
```

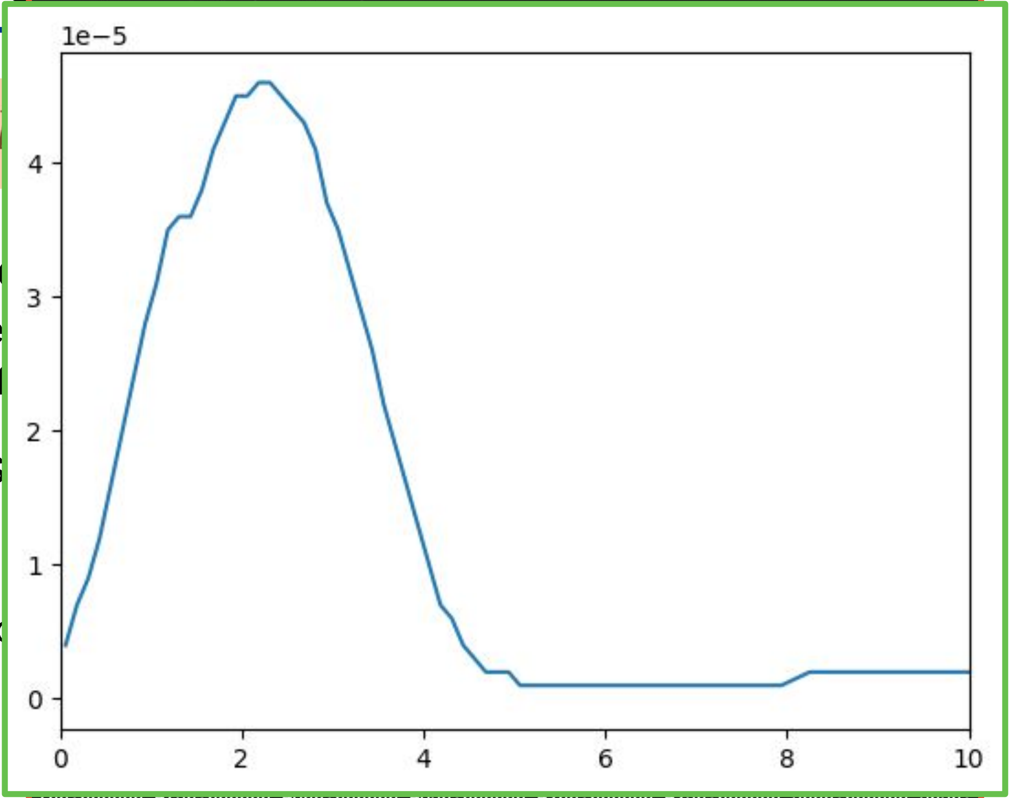
Key Example: Flux Desc

```
A NuHepMC.Beam.EnergyUnit MEV
A NuHepMC.Beam.RateUnit #frac{Number of #nu_{#mu}}{1.592 #upoint 10^{20} POT cm^{2} 0.05
 GeV}
A NuHepMC.Beam.Type Histogram
```

$N \quad (E_{\mathrm{obs}}) = \int dE_{\nu}\, \Phi \quad (I$

Store as bin edges and content

Trivial to convert back to histogram format of your choosing




```
000000 5100.000000 5150.000000 5200.000000 5250.000000 5300.000000 5350.000000 5400.0000
00 5450.000000 5500.000000 5550.000000 5600.000000 5650.000000 5700.000000 5750.000000 5
800.000000 5850.000000 5900.000000 5950.000000 6000.000000 6050.000000 6100.000000 6150.
000000 6200.000000 6250.000000 6300.000000 6350.000000 6400.000000 6450.000000 6500.0000
00 6550.000000 6600.000000 6650.000000 6700.000000 6750.000000 6800.000000
A NuHepMC.Beam[14].Histogram.ContentIsPerWidth 0
```

Key Example: Simple Citations

- We worry about attribution/funding/job security in HEP software
  - This is just as, if not more true, for early career theory/pheno people who often put the *physics* in the software.
  - **Common Issue:** MCEG gets citation in expt. papers, individual model components don't
- Bake in reference/attribution metadata

Key Example: Simple Citations

- We worry about attribution/funding/job security in HEP software
    - This is just as, if not more true, for early career theory/pheno people who often put the *physics* in the software.
    - **Common Issue:** MCEG gets citation in expt. papers, individual model components don't
- Bake in reference/attribution metadata

```
A NuHepMC.AdditionalParticleNumbers -2000030000 -2000020000 -2000010000 91 92 93 2000000001 2000000002 20
A NuHepMC.Citations.Generator.DOI 10.1016/j.nima.2009.12.009 10.1140/epjs/s11734-021-00295-7
A NuHepMC.Citations.Process[100].DOI 10.1103/PhysRevD.79.053003
A NuHepMC.Citations.Process[150].DOI 10.1103/PhysRevD.79.053003
A NuHepMC.Citations.Process[200].DOI 10.1016/0370-1573(72)90010-5
A NuHepMC.Citations.Process[250].DOI 10.1103/PhysRevD.19.779 10.1103/PhysRevD.35.785
A NuHepMC.Citations.Process[400].DOI 10.1103/PhysRevD.76.113004
A NuHepMC.Citations.Process[450].DOI 10.1103/PhysRevD.76.113004
A NuHepMC.Citations.Process[600].DOI 10.1103/PhysRevD.50.3085 10.1103/PhysRevD.65.033002
A NuHepMC.Citations.Process[650].DOI 10.1103/PhysRevD.65.033002
A NuHepMC.Citations.Process[701].DOI 10.1016/0550-3213(87)90131-3
A NuHepMC.Citations.Process[703].DOI 10.1088/0954-3899/29/11/013
A NuHepMC.Conventions E.C.1 E.C.2 E.C.4 E.C.5 G.C.1 G.C.4 G.C.6 G.S.2 P.C.1
```

Key Example: Sim

○ This is just as, if not more ... e

physics in the software.

○ **Common Issue:** MCEG g ... 't

● Bake in reference/attrib

```
15 A NuHepMC.AdditionalParticleNumber
16 A NuHepMC.Citations.Generator.DOI
17 A NuHepMC.Citations.Process[100].D
18 A NuHepMC.Citations.Process[150].D
19 A NuHepMC.Citations.Process[200].D
20 A NuHepMC.Citations.Process[250].D
21 A NuHepMC.Citations.Process[400].D
22 A NuHepMC.Citations.Process[450].D
23 A NuHepMC.Citations.Process[600].D
24 A NuHepMC.Citations.Process[650].D
25 A NuHepMC.Citations.Process[701].D
26 A NuHepMC.Citations.Process[703].D
27 A NuHepMC.Conventions E.C.1 E.C.2
```

```
@article{Berger:2008xs,
    author = "Berger, Ch. and Sehgal, L. M.",
    title = "{PCAC and coherent pion production by low energy neutrinos}",
    eprint = "0812.2653",
    archivePrefix = "arXiv",
    primaryClass = "hep-ph",
    volume = "79",
    pages = "053003",
    year = "2009"
}
@article{Berger:2008xs,
    author = "Berger, Ch. and Sehgal, L. M.",
    title = "{PCAC and coherent pion production by low energy neutrinos}",
    eprint = "0812.2653",
    archivePrefix = "arXiv",
    primaryClass = "hep-ph",
    doi = "10.1103/PhysRevD.79.053003",
    journal = "Phys. Rev. D",
    volume = "79",
    pages = "053003",
    year = "2009"
}
@article{LlewellynSmith:1971uhs,
    author = "Llewellyn Smith, C. H.",
    title = "{Neutrino Reactions at Accelerator Energies}",
    reportNumber = "SLAC-PUB-0958",
    doi = "10.1016/0370-1573(72)90010-5",
    journal = "Phys. Rept.",
    volume = "3",
    pages = "261--379",
    year = "1972"
}
```

```
~ > HEPReference/NuHepMCParse myevents.hepmc3 citations.bib
```

Key Example: Sim

```
@article{Berger:2008xs,
    author = "Berger, Ch. and Sehgal, L. M.",
    title = "{PCAC and coherent pion production by low energy neutrinos}",
    eprint = "0812.2653",
    archivePrefix = "arXiv",
    primaryClass = "hep-ph",
```

```
~ > HEPReference/NuHepMCParse myevents.hepmc3 citations.bib
```

```
    volume = "79",
```

○ This is just as, if not more

Event vectors themselves preserve citation information for all relevant components used at generation time

No excuse for under-citing of theory/pheno contributions to generators

```
26 A NuHepMC.Citations.Process[703].D
27 A NuHepMC.Conventions E.C.1 E.C.2
```

```
    pages = "261--379",
    year = "1972"
}
```

Status, Plans, and Prospects

Implementation Status

'Simple' Generator formats:

- Native Internal format: Marley
- Native output format: Achilles, GENIE (PR), GiBUU (Proof of Concept)
- On-the-fly conversion of native format to NuHepMC: NuWro, NEUT
- On-the-fly 2-way conversion: GENIE (PR)

Implementation Status

'Simple' Generator formats:

- Native Internal format: Marley
- Native output format: Achilles, GENIE (PR), GiBUU (Proof of Concept)
- On-the-fly conversion of native format to NuHepMC: NuWro, NEUT
- On-the-fly 2-way conversion: GENIE (PR)

Future Experiment stacks:

- Positive noises coming from LArSoft technical leadership (FNAL, DUNE) but no real movement due to lack of person power
    - Some small hope that support/interest from HSF might help build higher-level will to push for common/interoperable formats
- Working with Hyper-K people to build their stack on HepMC3/NuHepMC

Implementation Status

'Simple' Generator formats:

- Native Internal format: Marley
- Native … Concept)
- On-the… o, NEUT
- On-the…

Future Expe…

- Positiv… NAL, DUNE) but no …
  - Son… er-level will to push for common/interoperable formats
- Working with Hyper-K people to build their stack on HepMC3/NuHepMC

Not currently pushing on existing experiments as the cost/benefit doesn't seem good enough to warrant the effort from us or them

Plans, and Prospects

- **ASAP:** Submit NuHepMCv1 for publication (hopefully this week)
- **Short-term:** Get PRs merged, work with MCEG developers to add native output where possible to minimize maintenance
- **Medium-term:** work with next generation experiment stack developers to address missed requirements
    - Pressure them to build stacks on a common format, doesn't have to be this one, but it is possible and will make everyone involved's lives easier
- **Longer term:** Foster a community of related tools and support smaller user group with HepMC3 workflows.

Summary

- NuHepMC is a set of metadata specifications on top of HepMC3 software infrastructure.
  - Targeting, but not limited to neutrino probe analyses

Summary

- NuHepMC is a set of metadata specifications on top of HepMC3 software infrastructure.
  - Targeting, but not limited to neutrino probe analyses
- No community attempts to improve the **ν**-MCEG format quagmire during the T2K/NOvA era (since ~mid 2000s).
  - We need more interoperability, greater transparency, less cognitive load when using/improving MCEGs for the precision DUNE/Hyper-K era (next 30 years)

Summary

- NuHepMC is a set of metadata specifications on top of HepMC3 software infrastructure.
  - Targeting, but not limited to neutrino probe analyses
- No community attempts to improve the **ν**-MCEG format quagmire during the T2K/NOvA era (since ~mid 2000s).
  - We need more interoperability, greater transparency, less cognitive load when using/improving MCEGs for the precision DUNE/Hyper-K era (next 30 years)
- Solving the same problems that have been solved in the Collider world, 15 years later.
  - What else are we missing?
  - How do we get the support needed to fund this (and similar) community efforts for the long term? (I'm hoping the answer can start with: Engagement with HSF...)

Summary

- NuHepMC is a set of metadata specifications on top of HepMC3 software infrastructure.
  - Targeting, but not limited to neutrino probe analyses
- No commun gmire during the T2K/NO
  - We need r when using/imp ars)
- Solving the ider world, 15 years late
  - What else are we missing?
  - How do we get the support needed to fund this (and similar) community efforts for the long term? (I'm hoping the answer can start with: Engagement with HSF...)

Massive thanks to Andrii especially, and the rest of the HepMC3 core team for enthusiastic support!

Summary

- NuHepMC is a set of metadata specifications on top of HepMC3 software infrastructure.
    - Targeting, but not limited to neutrino probe analyses
- No community attempts to improve the **ν**-MCEG format quagmire during the T2K/NOvA era (since ~mid 2000s).
    - We need more interoperability, greater transparency, less cognitive load when using/improving MCEGs for the precision DUNE/Hyper-K era (next 30 years)
- Solving the same problems that have been solved in the Collider world, 15 years later.
    - What else are we missing?
    - How do we get the support needed to fund this (and similar) community efforts for the long term? (I'm hoping the answer can start with: Engagement with HSF...)

Done! Thanks for listening