Science & Technology Facilities Council
Rutherford Appleton Laboratory

GENIE Tutorial

Costas Andreopoulos

Last update: March 18th 2009

Recommended GENIE version >= 2.5.1

Science & Technology Facilities Council

Introduction

Goals

Tutorial structure / Activities

Tutorial Goals

At the end of this tutorial you should:

- Feel confident with using GENIE – Understand the project evolution
- Know where to look for more information
- Have a better understanding of the modelled physics
- Know how to turn its physics knobs
- Know how to ask GENIE for any cross section
- Know how to ask GENIE for any event sample

Know how to use the GENIE reweighting tools in your analysis

Science & Technology Facilities Council

Tutorial structure

A set of "learning tasks"

including many hands-on activities for you!

to be completed here before moving on to the next task

Interrupt me at any point with any question you may have

Tasks 1/4

stuff in red indicates hands-on activities for you

Understanding the GENIE package structure
How to build & update – releases & versions

Understanding events
Understanding the GHEP event record & the interaction summary

A.1 **Neutrino interaction quiz!**
Figure out what happened in this event...

Basic GENIE usage

A.2 **Warming up!**
A simple event generation case; Inspecting log files and output event files

Tasks 2/4

stuff in red indicates hands-on activities for you

A.3 **Converting event files.**
Generating summary analysis ntuples. Plotting basic quantities

A.4 **Using GENIE cross section splines in your code**

A.5 **Tweak physics parameters.**
Generate tweaked sample and compare with nominal sample.

A.6 **Setting the seed number.**

A.7 **Understand what goes into the log files.**
Control verbosity and print-out the information you want.

Tasks 3/4

stuff in red indicates hands-on activities for you

A.8 **Generate events for a simple flux**

A.9 **Verify correct energy dependence of the event sample**

The event loop

A.10 **Writing your own neutrino event loop**

GENIE as a hadron-nucleus event generator

A.11 **Writing an event loop to analyse a hadron-nucleus event sample**

Tasks 4/4

stuff in red indicates hands-on activities for you

Using complicated fluxes and realistic geometries

Template application: T2K event generation driver

A.12 **Generate your own events using the detailed T2K/nd280 geometry & flux**

Event reweighting tools in GENIE

A.13 **Experiment with reweighting your own samples**

GENIE algorithms / algorithm factory / configurations

A.14 **Experiment with accessing GENIE algorithms directly**

Material (*) for completing your tasks

Can be found at:

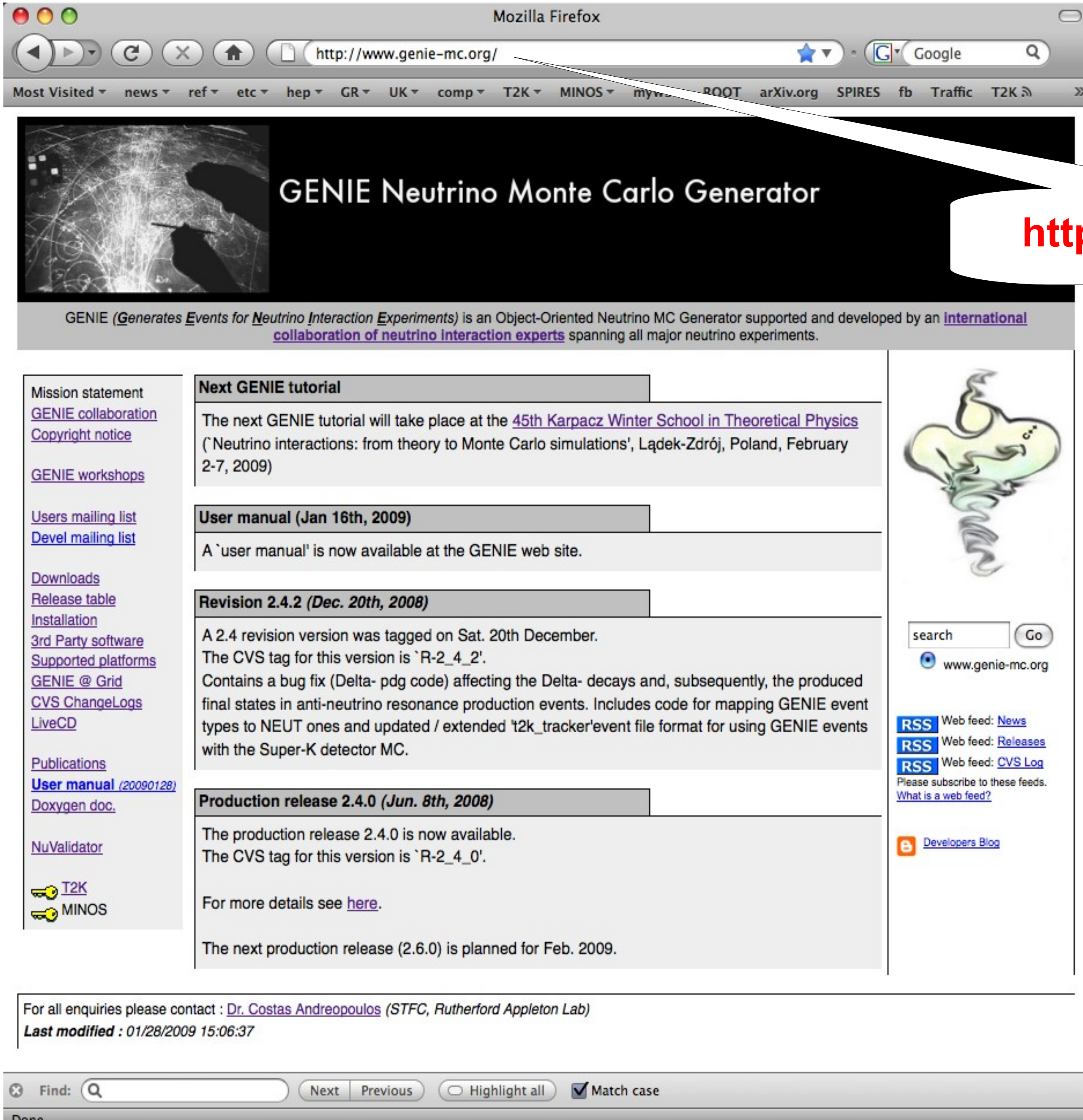
http://hepunx.rl.ac.uk/~candreop/outbox/genie/tutorial/fermilab/

(*) flux files, geometry files, cross section splines

Finding out more …

Getting support …

Staying in touch …


Mozilla Firefox

http://www.genie-mc.org/

Most Visited news ref etc hep GR UK comp T2K MINOS ROOT arXiv.org SPIRES fb Traffic T2K

GENIE Neutrino Monte Carlo Generator

GENIE (*Generates Events for Neutrino Interaction Experiments*) is an Object-Oriented Neutrino MC Generator supported and developed by an international collaboration of neutrino interaction experts spanning all major neutrino experiments.

Mission statement
GENIE collaboration
Copyright notice

GENIE workshops

Users mailing list
Devel mailing list

Downloads
Release table
Installation
3rd Party software
Supported platforms
GENIE @ Grid
CVS ChangeLogs
LiveCD

Publications
User manual *(20090128)*
Doxygen doc.

NuValidator

T2K
MINOS

Next GENIE tutorial

The next GENIE tutorial will take place at the 45th Karpacz Winter School in Theoretical Physics (`Neutrino interactions: from theory to Monte Carlo simulations', Lądek-Zdrój, Poland, February 2-7, 2009)

User manual (Jan 16th, 2009)

A `user manual' is now available at the GENIE web site.

Revision 2.4.2 *(Dec. 20th, 2008)*

A 2.4 revision version was tagged on Sat. 20th December.
The CVS tag for this version is `R-2_4_2'.
Contains a bug fix (Delta- pdg code) affecting the Delta- decays and, subsequently, the produced final states in anti-neutrino resonance production events. Includes code for mapping GENIE event types to NEUT ones and updated / extended 't2k_tracker'event file format for using GENIE events with the Super-K detector MC.

Production release 2.4.0 *(Jun. 8th, 2008)*

The production release 2.4.0 is now available.
The CVS tag for this version is `R-2_4_0'.

For more details see here.

The next production release (2.6.0) is planned for Feb. 2009.

search Go
www.genie-mc.org

RSS Web feed: News
RSS Web feed: Releases
RSS Web feed: CVS Log
Please subscribe to these feeds.
What is a web feed?

Developers Blog

For all enquiries please contact : Dr. Costas Andreopoulos *(STFC, Rutherford Appleton Lab)*
***Last modified :** 01/28/2009 15:06:37*

Find: Next Previous Highlight all Match case

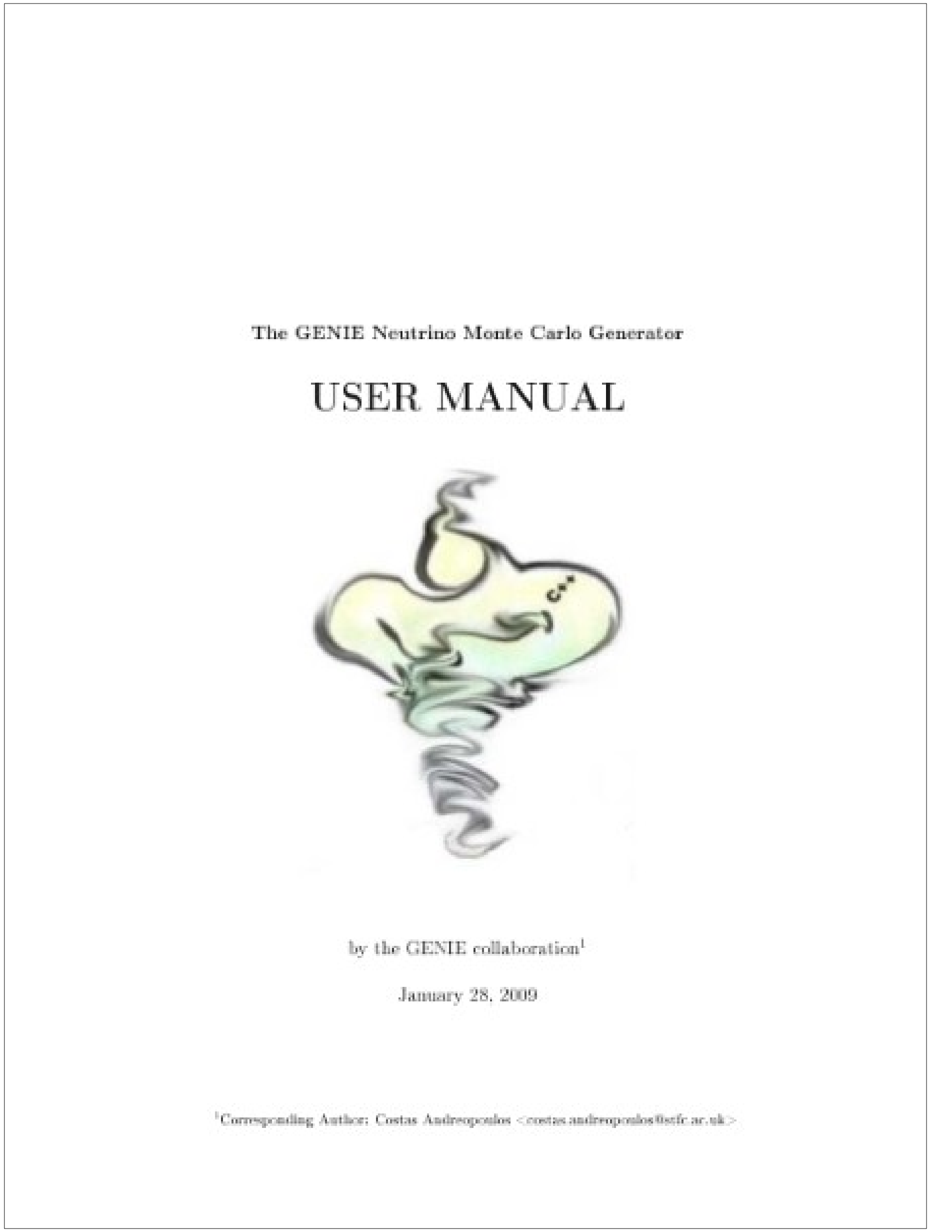
Done

http://www.genie-mc.org

Official GENIE web site

You can find:

- *Physics documentation*
- *A user manual*
- *Release tables*
- *Download instructions*
- *Installation instructions*
- *Doxygen documentation*
- *News feeds*
- *...*
- *...*



The GENIE Neutrino Monte Carlo Generator

USER MANUAL

by the GENIE collaboration[1]

January 28, 2009

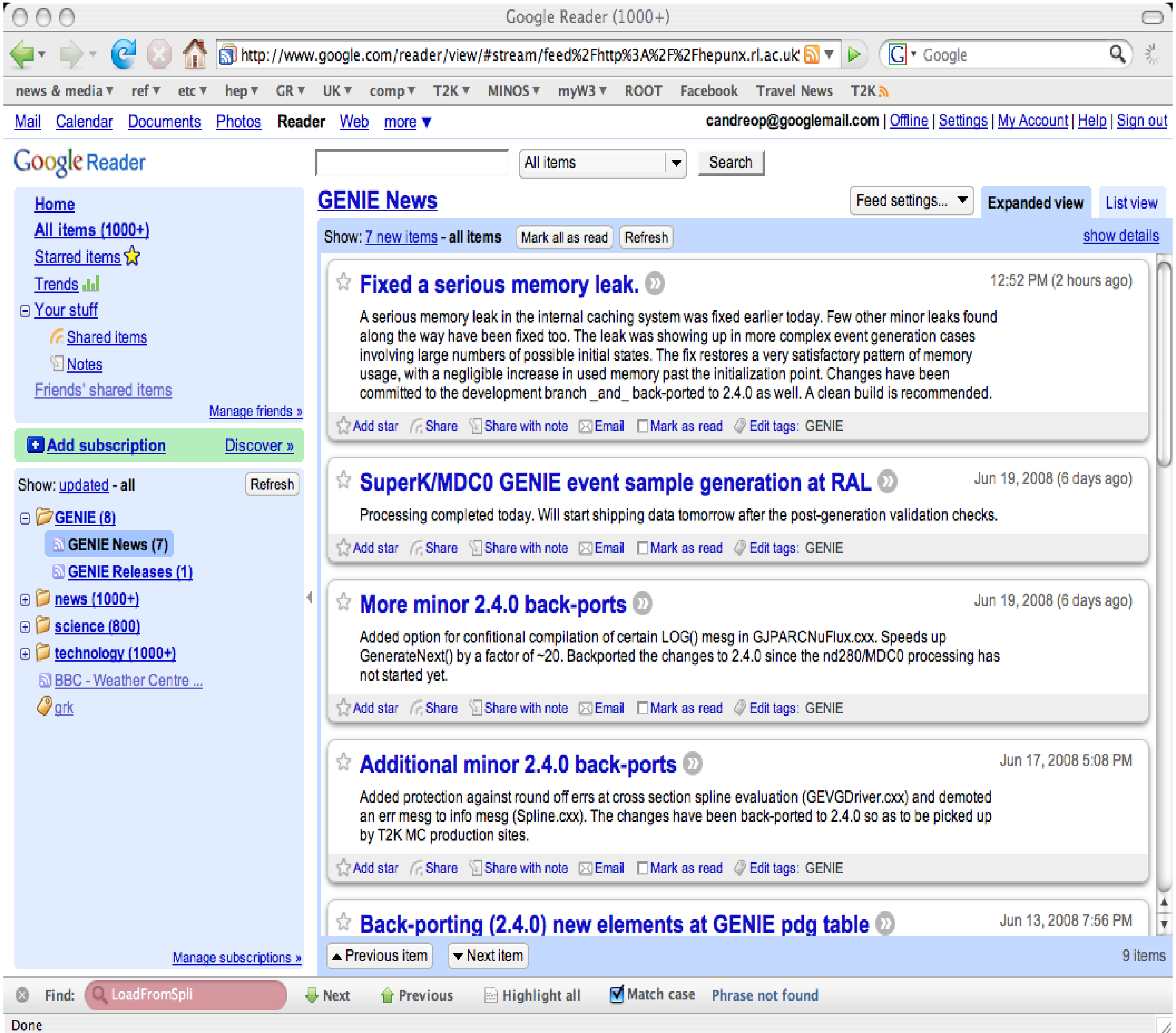
[1]Corresponding Author: Costas Andreopoulos <costas.andreopoulos@stfc.ac.uk>

NEW

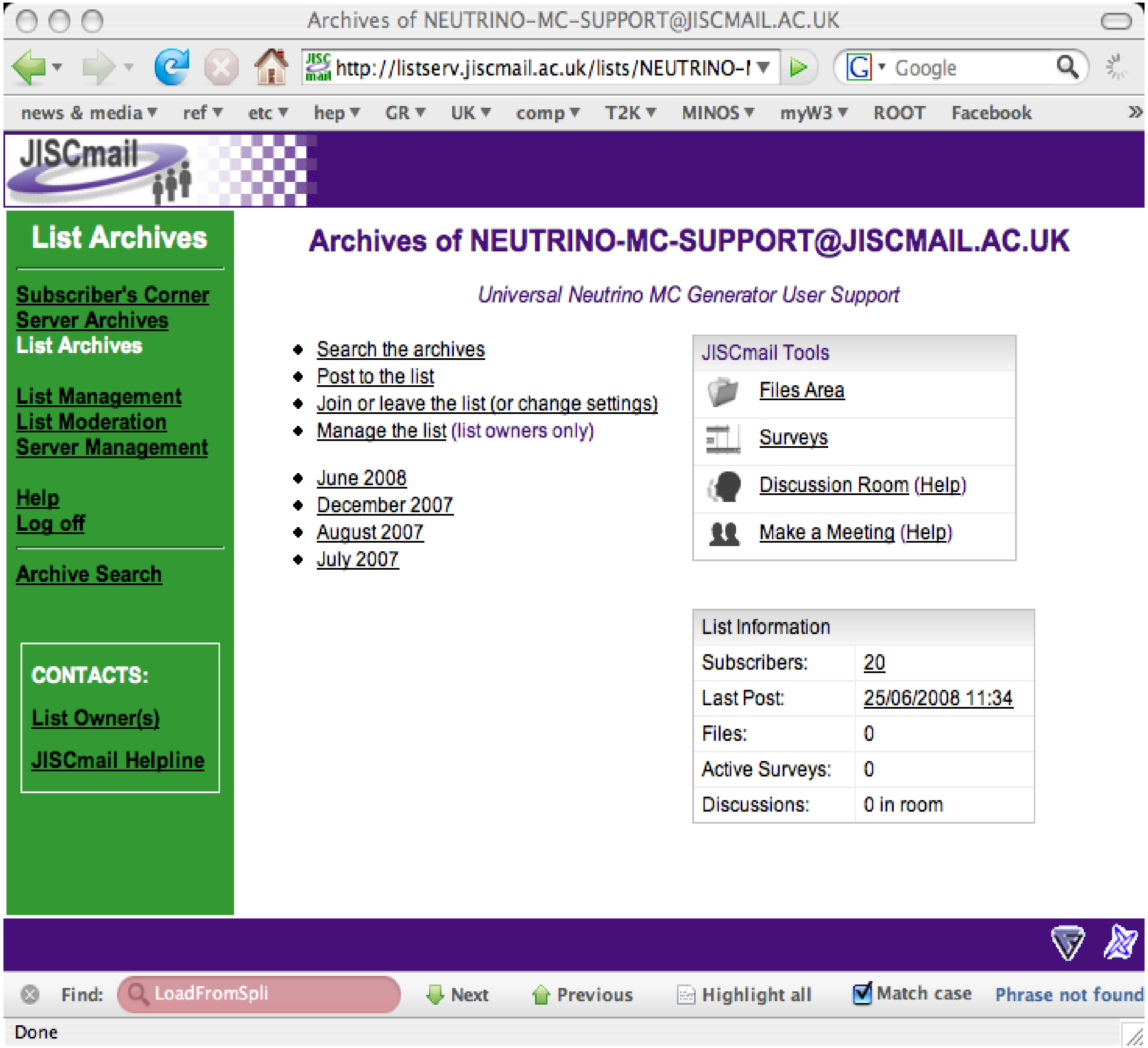
Read the GENIE users manual

Costas Andreopoulos, *Rutherford Appleton Lab.*

Science & Technology Facilities Council



Subscribe to the GENIE RSS news feeds !



Subscribe to the GENIE users mailing list !

CVS Tags, Versions

CVS Tags

In the GENIE version numbering scheme, releases are tagged in the CVS source code repository as ***R-major_minor_revision***[1]. When a number of significant functionality improvements or additions have been made, the major index is incremented. The minor index is incremented in case of significant fixes and/or minor feature additions. The revision number is incremented for minor bug fixes and updates.

Version number semantics

- Following the LINUX kernel versioning scheme, the versions with even minor number (eg 2.0.*, 2.4.*) correspond to stable, fully validated physics production releases[2].
- Versions with odd minor number (eg. 2.3.*, 2.5.*) correspond to the development version or 'candidate' releases tagged during the validation stage preceding a physics release.
- Tagged versions always have an even revision number (eg 2.2.2 is a revision of the 2.2.0 physics production release whereas 2.3.2, 2.3.4, 2.3.6,... are 'candidate' releases for the 2.4.0 physics production release).
- Odd revision numbers are used for the CVS head / development version.

Release Code-Names and qualifiers

The major production-quality releases are code-named after modern extinct or endangered species (series of production releases: ***Auk***, ***Blueback***, ***Cheetah***, ***Dodo***, ***Elk***, ***Fox***, ***Gazelle***, ***Hippo***, ***Ibex***,...).

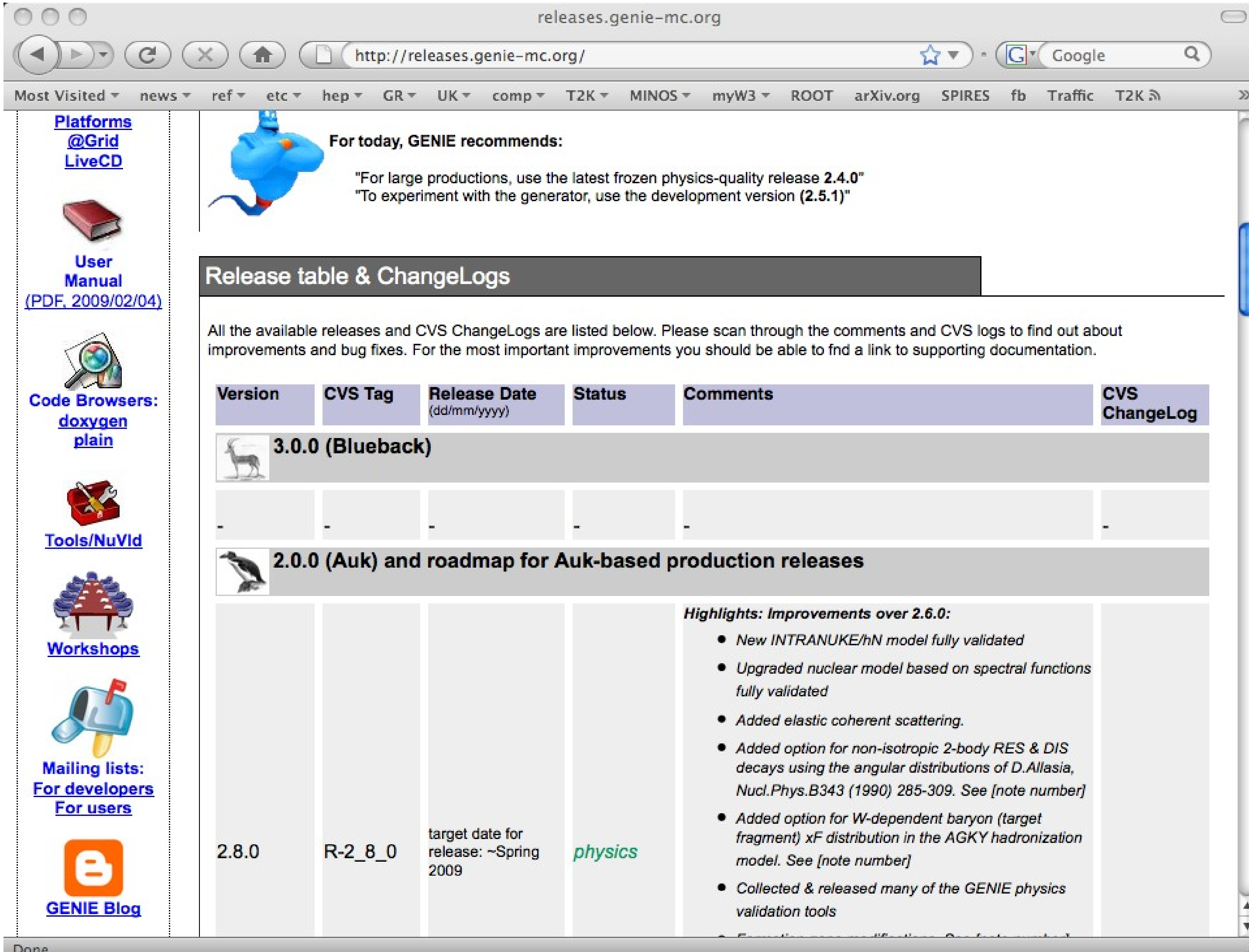
Auk: v2.*.*

Blueback: v3.*.*

- ***physics*** : Validated production-quality versions recommended for physics studies.
- ***depreciated***: Older 'physics' versions that have been greatly superseded by newer versions. Versions marked as 'depreciated' become unsupported. We appreciate that experiments get highly attached on specific versions due to the enormous amount of work invested in generating high statistics samples and calculating MC-dependent corrections and systematics. We strive to support as many physics versions as reasonably possible.
- ***pre-release***: Test releases you may not use for physics studies.
- ***special***: Releases prepared for a particular reason or event such as a) the evaluation of an experiment systematic with an appropriately modified version of GENIE, or b) a GENIE tutorial or a summer / winter school. You may not use these releases outside the intended context.

Lifetime for physics releases: ~ 18-24 months

Follow progress on new releases @ http://releases.genie-mc.org

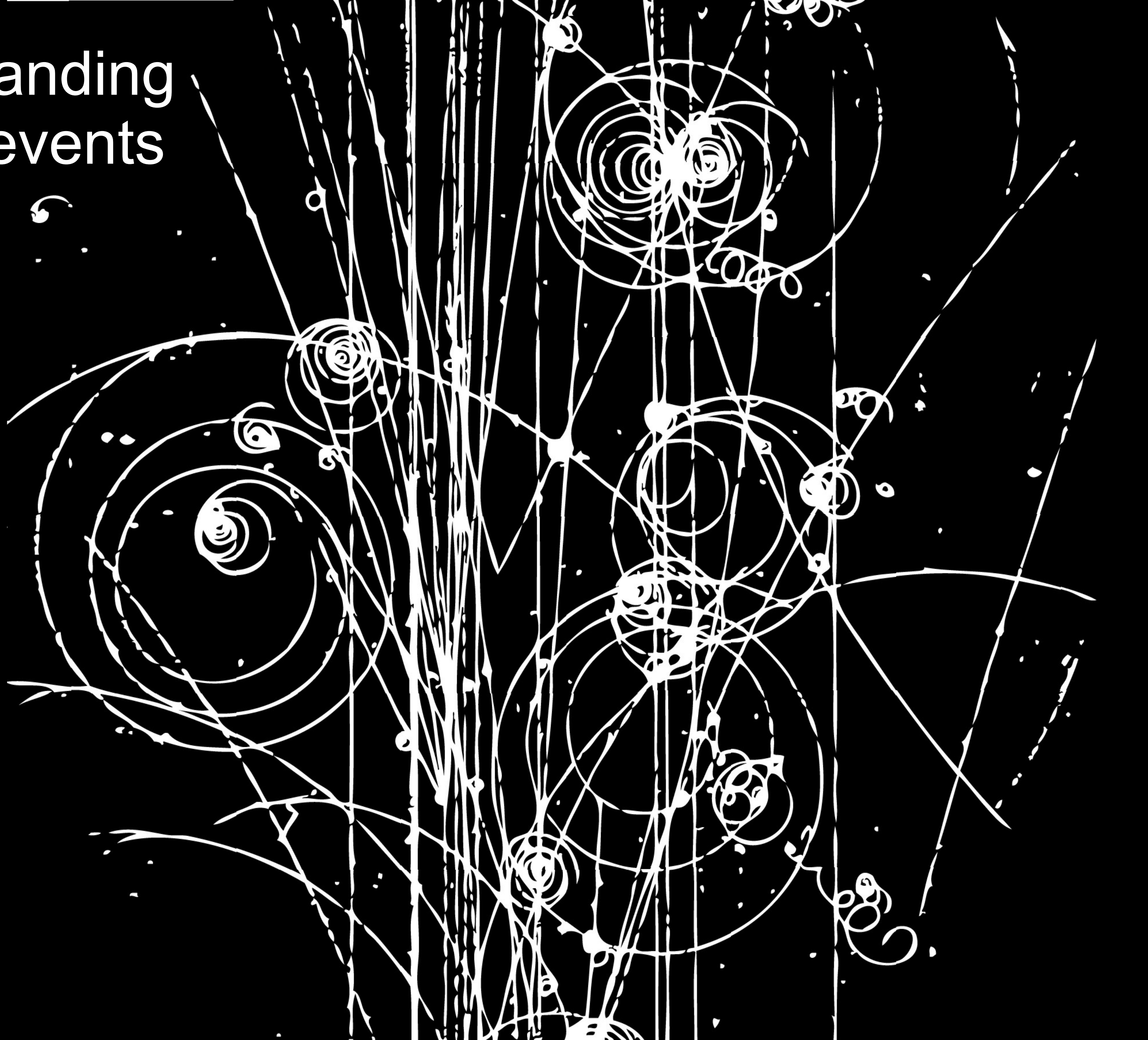


Web page includes a summary of improvements and links to supporting documentation.

CVS changelogs provide a complete history of code changes

New releases are also announced at the RSS feed and at the mailing list.

Understanding GENIE events



Understanding GENIE events

Stored in `GHEP' event trees (GHEP: a customized StdHep-like event record)

GHepParticles :

- *Holds info about generated particles*
  (4-momentum, 4-position in nucleus coord syst, charge, mass, name, polarization, ...)
- *Generated particles can be initial / intermediate / final-state particles or generator book-keeping actions (pseudo-particles)*

GHepRecord:

- A TClonesArray of GHepParticles
- Holds info with event-wide scope
  (weights, flags, vertex in detector coord syst, …)
- Also contains an "summary information" for the generated interactions
  (to be described in a second...)

Understanding GENIE events

PDG codes

Standard codes for all particles
PDG-2006 codes for ions (10LZZZAAAI), eg Fe56: 1000260560

Status codes

Description	*GHepStatus_t*	As *int*
Undefined	*kIStUndefined*	-1
Initial state	*kIStInitialState*	0
Stable final state	*kIstStableFinalState*	1
Intermediate state	*kIStIntermediateState*	2
Decayed state	*kIStDecayedState*	3
Nucleon target	*kIStNucleonTarget*	11
DIS pre-fragm. hadronic state	*kIStDISPreFragmHadronicState*	12
Resonant pre-decayed state	*kIStPreDecayResonantState*	13
Hadron in the nucleus	*kIStHadronInTheNucleus*	14
Remnant nucleus	*kIStFinalStateNuclearRemnant*	15

Mother / daughter links

Understanding GENIE events - 1st example

<u>A **nu_mu + Fe56 resonance** event</u>

initial state

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

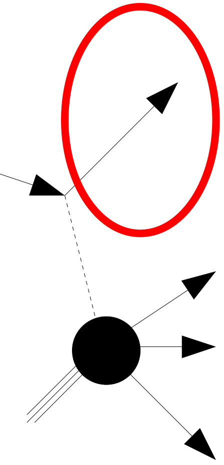
Understanding GENIE events - 1st example

Fe56 = { hit nucleon } + { `remnant' nucleus } = p + Mn55

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 1st example

Incoming neutrino → final state primary lepton (eg. numu CC → mu-)



Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 1st example

Hit proton excited to Delta++

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 1st example

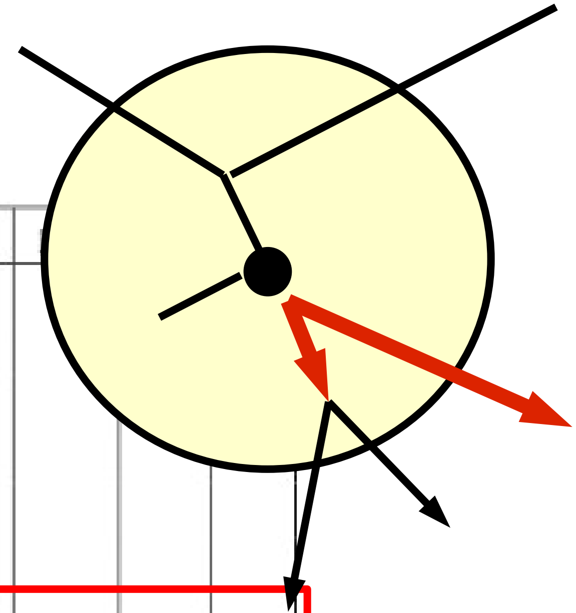
Delta++ decays (selected decay channel: proton pi+)
Decay happened in nuclear environment → Decay products marked as `hadrons in the nucleus (14)'

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 1st example

GENIE sees particles marked `hadrons in the nucleus (14)'
Begin **intra-nuclear hadron transport**

Idx	Name	ISt	PDG	Mom	Kids	E
0	nu_mu	0	14	-1	5 5	...
1	Fe56	0	1000260560	-1	2 3	
2	proton	11	2212	1	4 4	
3	Mn55	2	1000250550	1	12 12	
4	Delta++	3	2224	2	6 7	
5	mu-	1	13	0	-1 -1	
6	proton	14	2112	4	8 8	
7	pi+	14	211	4	11 11	
8	proton	3	2212	6	9 10	
9	proton	1	2212	8	-1 -1	
10	proton	1	2212	8	-1 -1	
11	pi+	1	211	7	-1 -1	
12	HadrBlob	15	2000000002	3	-1 -1	



Costas Andreopoulos, ***Rutherford Appleton Lab.***

Understanding GENIE events - 1st example

Multi-nucleon knock-out

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 1st example

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 1st example

Nuclear remnant

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 1st example

(Neutrino generator) Final state particles

To be passed-on to detector (eg Geant4-based) simulation

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	5 5	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	proton	11	2212	1	4 4				
3	Mn55	2	1000250550	1	12 12				
4	Delta++	3	2224	2	6 7				
5	mu-	1	13	0	-1 -1				
6	proton	14	2112	4	8 8				
7	pi+	14	211	4	11 11				
8	proton	3	2212	6	9 10				
9	proton	1	2212	8	-1 -1				
10	proton	1	2212	8	-1 -1				
11	pi+	1	211	7	-1 -1				
12	HadrBlob		2000000002	3	-1 -1				

Costas Andreopoulos, *Rutherford Appleton Lab.*

Understanding GENIE events - 2nd example

A **nu_mu + Fe56 DIS** event

initial state

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob		2000000002	3	-1 -1				

Understanding GENIE events - 2nd example

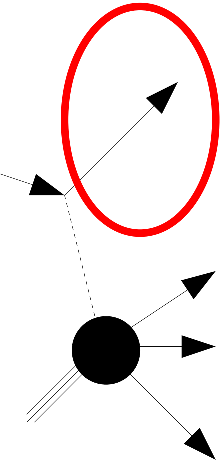
Fe56 = { hit nucleon } + { `remnant' nucleus } = n + Fe55

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob	15	2000000002	3	-1 -1				

Understanding GENIE events - 2^{nd} example

Final state primary lepton

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob		2000000002	3	-1 -1				

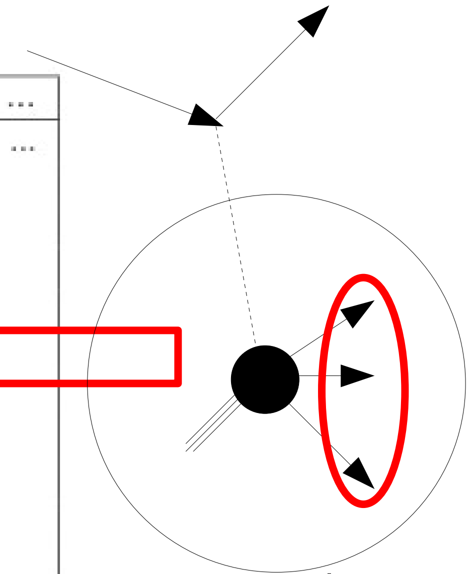


Costas Andreopoulos, *Rutherford Appleton Lab.*

Understanding GENIE events - 2nd example

nu_mu + n → mu- + **X** (X: pre-fragmented hadronic system)

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob	15	2000000002	3	-1 -1				



Understanding GENIE events - 2nd example

Hadronization

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob		2000000002	3	-1 -1				

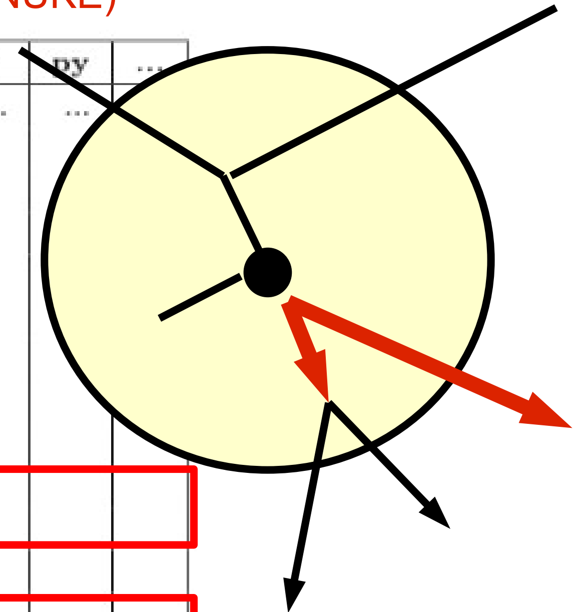
\omega decay

Costas Andreopoulos, *Rutherford Appleton Lab.*

Understanding GENIE events - 2nd example

Particles to be tracked by GENIE intranuclear hadron transport (INTRANUKE)

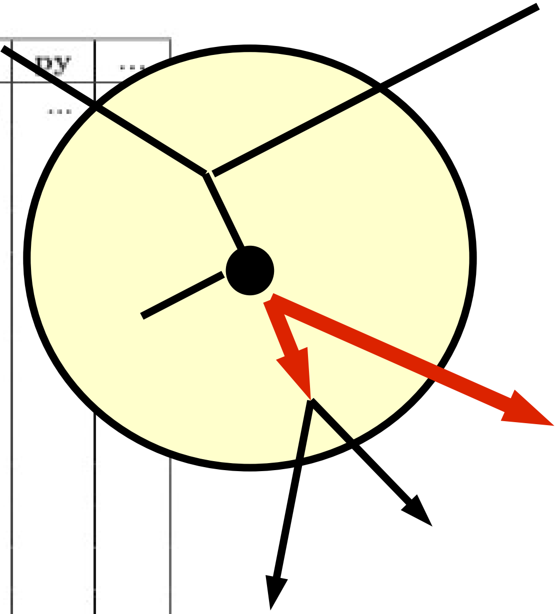
Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob		2000000002	3	-1 -1				



Understanding GENIE events - 2nd example

A pi- re-scattering (absorption followed by nucleon emission)

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob		2000000002	3	-1 -1				



Understanding GENIE events - 2nd example

Nuclear remnant

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob	15	2000000002	3	-1 -1				

Science & Technology Facilities Council

Understanding GENIE events - 2nd example

(Neutrino generator) Final state particles (= Geant4 primaries)

Idx	Name	ISt	PDG	Mom	Kids	E	px	py	...
0	nu_mu	0	14	-1	4 4	...	...	...	...
1	Fe56	0	1000260560	-1	2 3				
2	neutron	11	2112	1	5 5				
3	Fe55	2	1000260550	1	22 22				
4	mu	1	13	0	-1 -1				
5	HadrSyst	12	2000000001	2	6 7				
6	u	12	2	5	8 8				
7	ud_1	12	2103	5	-1 -1				
8	string	12	92	6	9 11				
9	pi0	14	111	8	14 14				
10	proton	14	2212	8	15 15				
11	omega	12	223	8	12 13				
12	pi-	14	-211	11	16 16				
13	pi+	14	211	11	21 21				
14	pi0	1	111	9	-1 -1				
15	proton	1	2212	10	-1 -1				
16	pi-	3	-211	12	17 20				
17	neutron	1	2112	16	-1 -1				
18	neutron	1	2112	16	-1 -1				
19	proton	1	2212	16	-1 -1				
20	proton	1	2212	16	-1 -1				
21	pi+	1	211	13	-1 -1				
22	HadrBlob		2000000002	3	-1 -1				

Costas Andreopoulos, *Rutherford Appleton Lab.*

The Event Record & <u>Interaction</u> summary

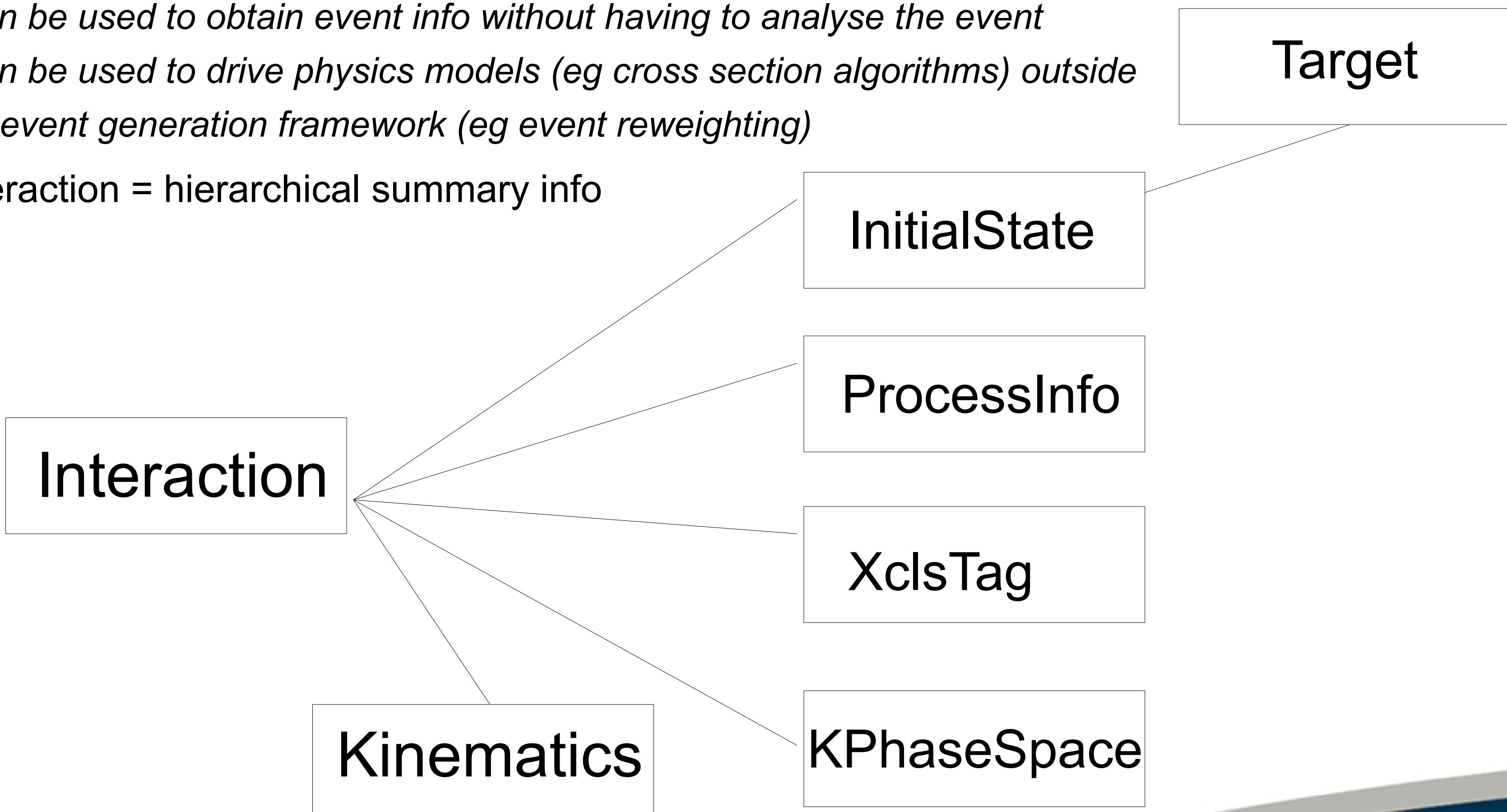
The event record contains <u>all info</u> about a generated event

An interaction object contains summary information about the event.

** Can be used to obtain event info without having to analyse the event*

** Can be used to drive physics models (eg cross section algorithms) outside The event generation framework (eg event reweighting)*

Interaction = hierarchical summary info



Extracting information from the Interaction summary

```
If (interaction.ProcInfo().IsQuasiElastic()) { ... }

double Q2 =  interaction.Kine().Q2();

double Ev = interaction.InitState().ProbeE(kRfLab);

int Z = interaction.InitState().Tgt().Z();

double Ethr = interaction.PhaseSpace().Threshold();

...
...
```

Extracting event information / Example 1

Extract the interaction summary for the given event and check whether it is a QEL CC event (excluding QEL CC charm production):

```
{
 ...

 Interaction * in = event.Summary();

 const ProcessInfo & proc = in->ProcInfo();
 const XclsTag & xclsv    = in->ExclTag();

 bool qelcc = proc.IsQuasiElastic() && proc.IsWeakCC();
 bool charm = xclsv.IsCharm();

 if (qelcc && !charm)
 {
    ...
 }
 ...
}
```

Extracting event information / Example 2

Calculate the momentum transfer Q^2, the energy transfer ν, the Bjorken x variable, the inelasticity y and the hadronic invariant mass W directly from the event record:

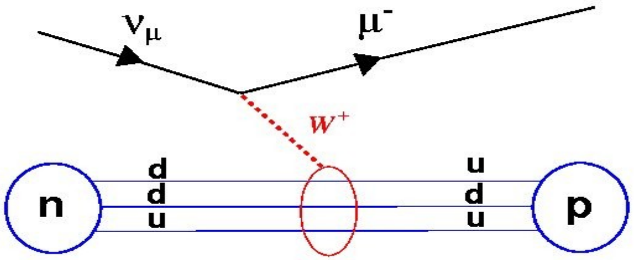
```
{
  ...
  // get the neutrino, f/s primary lepton and hit
  // nucleon event record entries
  //
  GHepParticle * neu = event.Probe();
  GHepParticle * fsl = event.FinalStatePrimaryLepton();
  GHepParticle * nuc = event.HitNucleon();

  // the hit nucleon may not be defined
  // (eg. for coherent, or ve- events)
  //
  if(!nuc) return;

  // get their corresponding 4-momenta (@ LAB)
  //
  const TLorentzVector & k1 = *(neu->P4());
  const TLorentzVector & k2 = *(fsl->P4());
  const TLorentzVector & p1 = *(nuc->P4());

  // calculate the kinematic variables
  // (eg see Part.Phys. booklet, page 191)
  //
  double M = kNucleonMass;

  TLorentzVector q = k1 - k2;

  double Q2 = -1 * q.M2();
  double v  = q.Energy();
  double x  = Q2 / (2*M*v);
  double y  = v / k1.Energy();
  double W2 = M*M - 2*M*v - Q2;
  double W  = TMath::Sqrt(TMath::Max(0., W2));

  ...
}
```

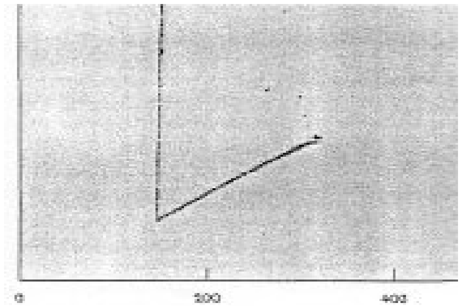
Check out http://doxygen.genie-mc.org

Take 5 minutes to familiarize yourselves with the public methods of the classes
In the GHEP and Interaction packages

Neutrino interactions

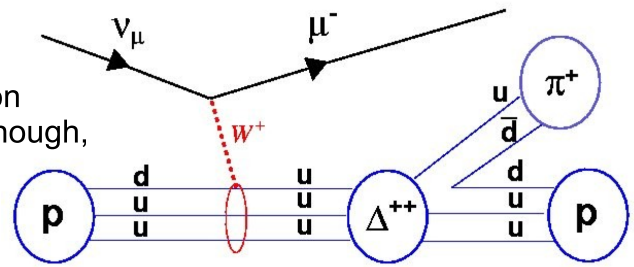
QEL, RES, DIS dominant
in different kinematical regimes

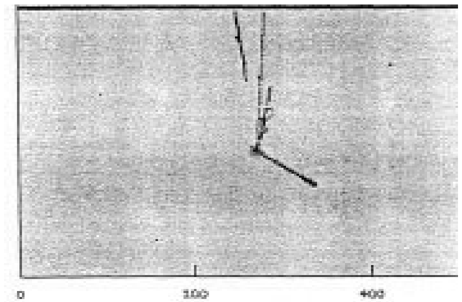




QEL

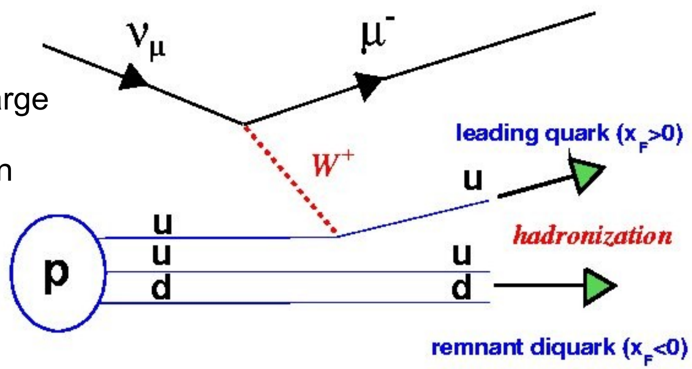
Usually single pion + nucleon
Other final states possible though,
eg Delta -> N + gamma

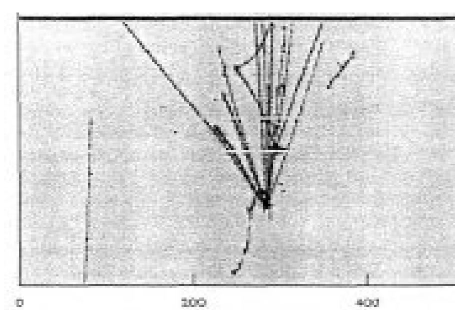




RES

q, qq materialize to give large
number of f/s.
At low masses though can
look like RES





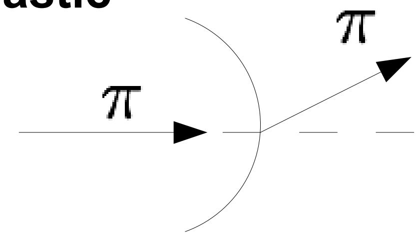
DIS

Increasing multiplicity

Note: Free Nucleon interactions

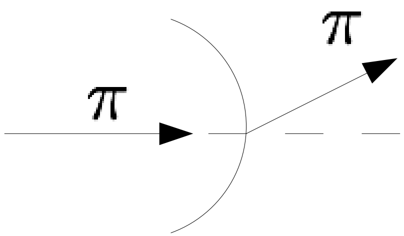
Hadronic re-interactions

elastic



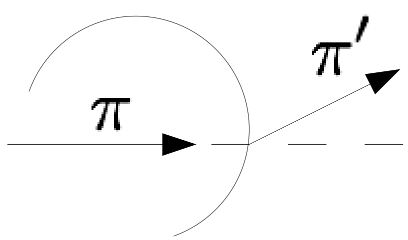
Pion deflected. Its kinetic energy stays the same.

inelastic



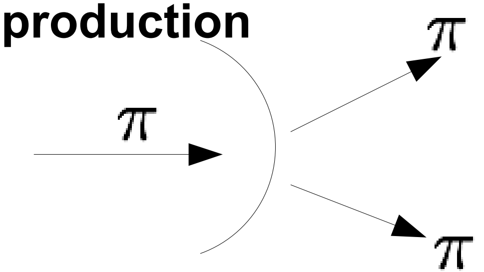
Pion deflected. Its kinetic energy is degraded.

charge exchange

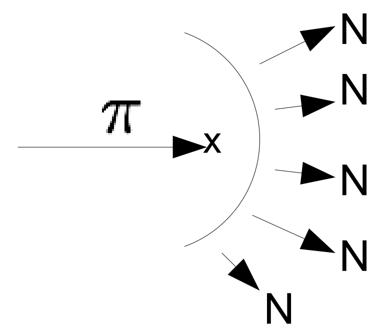


Pion deflected.
Its kinetic energy is degraded.
Changes sign:
pi+ → pi0
pi0 → pi-
...

pion production



absorption



followed by emission of low energy nucleons

~ Similar fates for nucleons

Time for our first activity!

>>>

Activity.1

Neutrino interaction quiz!

Go to http://hepunx.rl.ac.uk/~candreop/generators/GENIE/faq/events.txt

- Find QEL, RES, DIS and COH pi production events.
- Find events with pion absorption followed by nuclear break-up.
- Find events with pion charge exchange.
- Find a Delta++ resonance. Find a Delta+ resonance. Find a N0 resonance.
- Notice Delta decay at event 17.
- Find an event with photons from nuclear de-excitation.
- Find a high-W DIS event & see the quark – diquark system passed to JETSET.
-
-
-
-

Basic GENIE usage

Now we will learn how to generate events for simple cases

gevgen syntax

***gevgen*: basic event generation driver**

The source code for this utility may be found in '$GENIE/*src*/*stdapp*/gEvGen.cxx'.

In summary, the *gevgen* syntax is:

```
shell$ gevgen [-h] -n nev [-s] -e E -p nu -t tgt [-r run#] [-f flux] [-w]
```

where [] denotes an optional argument

Details on the command-line arguments can be found below:

- **-h** Prints-out help on *gevgen* syntax and exits.
- **-n** Specifies the number of events to generate.
- **-r** Specifies the MC run number.
- **-s** Instructs the driver to enable cross section spline interpolation.
- **-e** Specifies the neutrino energy or energy range.
- **-p** Specifies the neutrino PDG code.
- **-t** Specifies the target PDG code
- **-f** Specifies the neutrino flux spectrum.
- **-w** Forces generation of weighted events.

Science & Technology Facilities Council

gevgen: more on flux options

-f Specifies the neutrino flux spectrum.

This generic event generation driver allows to specify the flux in any one of three simple ways:

- As a 'function'.
  For example, in order to specify a flux that has the $x^2 + 4e^{-x}$ functional form, type:
  `'-f 'x*x+4*exp(-x)"`

- As a 'vector file'.
  The file should contain 2 columns corresponding to energy (in GeV), flux (in arbitrary units).
  For example, in order to specify that the flux is described by the vector file '*/data/fluxvec.data*', type:
  `'-f /data/fluxvec.data'`

- As a '1-D histogram (*TH1D*) in a ROOT file'.
  The general syntax is: `'-f /full/path/file.root,object_name'`.
  For example, in order to specify that the flux is described by the 'nue' *TH1D* object in '*/data/flux.root*', type:
  `'-f /data/flux.root,nue'`

* Later we will learn how to use more complex fluxes

gevgen: more on target options

-t Specifies the target PDG code

Eg. `-t 1000260560' for specifying Fe56

You can also use the -t option to specify a “target mix”

(list of comma separated PDG codes with corresponding weight fraction in sqr. brackets)

Eg. `-t 1000080160[0.96],1000010010[0.04]' for specifying 96%O16+4%H1

* Later we will learn how to use ROOT / Geant4 detector geometry descriptions

Science & Technology Facilities Council

gevgen examples

To generate 20,000 ν_μ (PDG code: 14) scattered off Fe^{56} (PDG code: 1000260560) at an energy of 6.5 GeV, type:

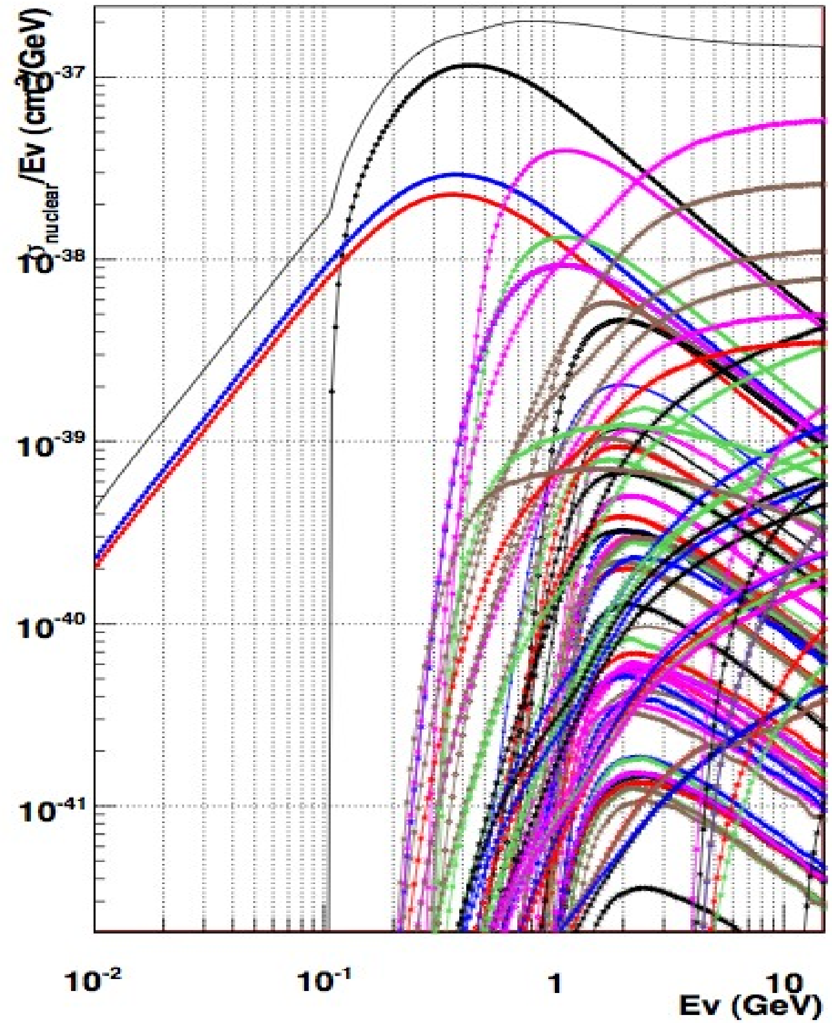
```
shell$ gevgen -n 20000 -s -e 6.5 -p 14 -t 1000260560
```

To generate a similar sample as above, but with the ν_μ energies, between 1 and 4 GeV, selected from a spectrum that has the $x^2 e^{(-x^2+3)/4}$ functional form, type:

```
shell$ gevgen -n 20000 -s -e 1,4 -p 14 -t 1000260560
-f 'x*x*exp((-x*x+3)/4)'
```

Science & Technology Facilities Council

Why using pre-computed cross sections?



Example: **nu_mu + O16:** All processes

~1E+2 \`processes'
~1E+5 diff. cross section evaluations per numerical integration

~1E+7 differential cross section evaluations per target

Typically ~1E+2 targets =>
<u>1E+9 xsec calc just in order to start generation!</u>

Impossible to calculate at generation time
Cross sections pre-computed and interpolated

---

gmkspl syntax

***gmkspl*: utility for building cross section splines**

In summary, the *gmkspl* syntax is:

```
shell$ gmkspl -p nu <-t tgt, -f geom> [-o out_file] [-n nknots] [-e Emax]
```

where:
[] marks optional arguments, and
<> marks a list of arguments out of which only one can be selected at any given time.

Details on the command-line arguments can be found below:

- **-p** Specifies the neutrino PDG codes.
  Multiple neutrino codes can be specified as a comma separated list.
- **-t** Specifies the target PDG codes.
  Multiple target PDG codes can be specified as a comma separated list.
- **-f** Specifies a ROOT file containing a ROOT/GEANT detector geometry description.
- **-o** Specifies the name of the output XML file
  By default GENIE writes-out the calculated cross section splines in an XML file named '*xsec_splines.xml*' created at the current directory.
- **-n** Specifies the number of knots per spline
  By default GENIE is using 15 knots per decade of the spline energy range and at least 30 knots overall.
- **-e** Specifies the maximum neutrino energy in spline
  By default the maximum energy is set to be the declared upper end of the validity range of the event generation thread responsible for generating the cross section spline.

gmkspl examples

1. To calculate the cross section splines for ν_μ (PDG code: 14) and $\bar{\nu_\mu}$ (PDG code: -14) scattered off Fe^{56} (PDG code: 1000260560), type:

   ```
   shell$ gmkspl -p 14,-14 -t 1000260560
   ```

   The cross section splines will be saved in an output XML file named '*xsec_splines.xml* ' (default name).

2. To calculate the cross section splines for ν_μ (PDG code: 14) and $\bar{\nu_\mu}$ (PDG code: -14) scattered off all the targets in the input ROOT geometry file '*/data/mygeometry.root*' and write out the splines in a file named '*mys-plines.xml*', type

   ```
   shell$ gmkspl -p 14,-14 -f /data/mygeometry.root -o mysplines.xml
   ```

gmkspl output

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- generated by genie::XSecSplineList::SaveSplineList() -->
<genie_xsec_spline_list version="2.00" uselog="1">

      <spline
          name   = "{algorithm/reaction; string}"
          nknots = "{number of knots; int}">
      <knot>
        <E>    {energy; double}        </E>
        <xsec> {cross section; double} </xsec>
      </knot>
      <knot>
        <E>    {energy; double}        </E>
        <xsec> {cross section; double} </xsec>
      </knot>
      ... ...
      </spline>

  ... ... ...
  ... ... ...
  </genie_xsec_spline_list>
```

Activity.2

Warming up!
A simple example using gmkspl and gevgen.

We will tell GENIE

- to focus on just a simple class of interactions (QEL-CC)
- to use a single initial state (nu_mu + 12C) at a single energy
- to generate cross section splines for nu_mu + 12C QEL
- to generate a nu_mu + 12C QEL event sample

We will

- examine the outputs
- learn what to do with the output event file

Activity.2

Tells GENIE we only care about QEL
To be explained by the tutor:

- *Where to find possible GEVGL values?*
  - *$GENIE/config/EventGeneratorListAssembler.xml*
- *What is the default?*
- *How to add your own?*

shell& export **GEVGL**=QEL-CC

shell& **gmkspl** <u>-p 14 -t 1000060120</u> <u>-n 100 -e 40</u> <u>-o my_qelcc_splines.xml</u>

probe & target PDG codes

knots & emax

output xml file

examine

Cross section spline generation driver
To be explained by the tutor:

- *Why do I need those splines?*
- *What if I do not pre-compute them?*
- *Where do I find pre-computed splines?*

Tells GENIE event generation where to find cross section splines – Automatically loaded at init

Option to examine input splines & add any missing (note - overheads)

shell& export **GSPLOAD**=my_qelcc_splines.xml

shell& **gevgen** -p 14 -t 1000060120 <u>-n 5000</u> <u>-e 1</u> <u>*-r 1000*</u> -s &> out.log

events

Energy
Can also be E range if a flux is given

Run nu.

Basic event generator driver

Activity.2

During generation:

Periodically, check status file (genie-mcjob-<run number>.status)

Activity.2

On completion:

Examine outputs

- A ROOT file containing the event tree

Print-out the generated events: **shell$ gevdump -f /path/to/file.root**

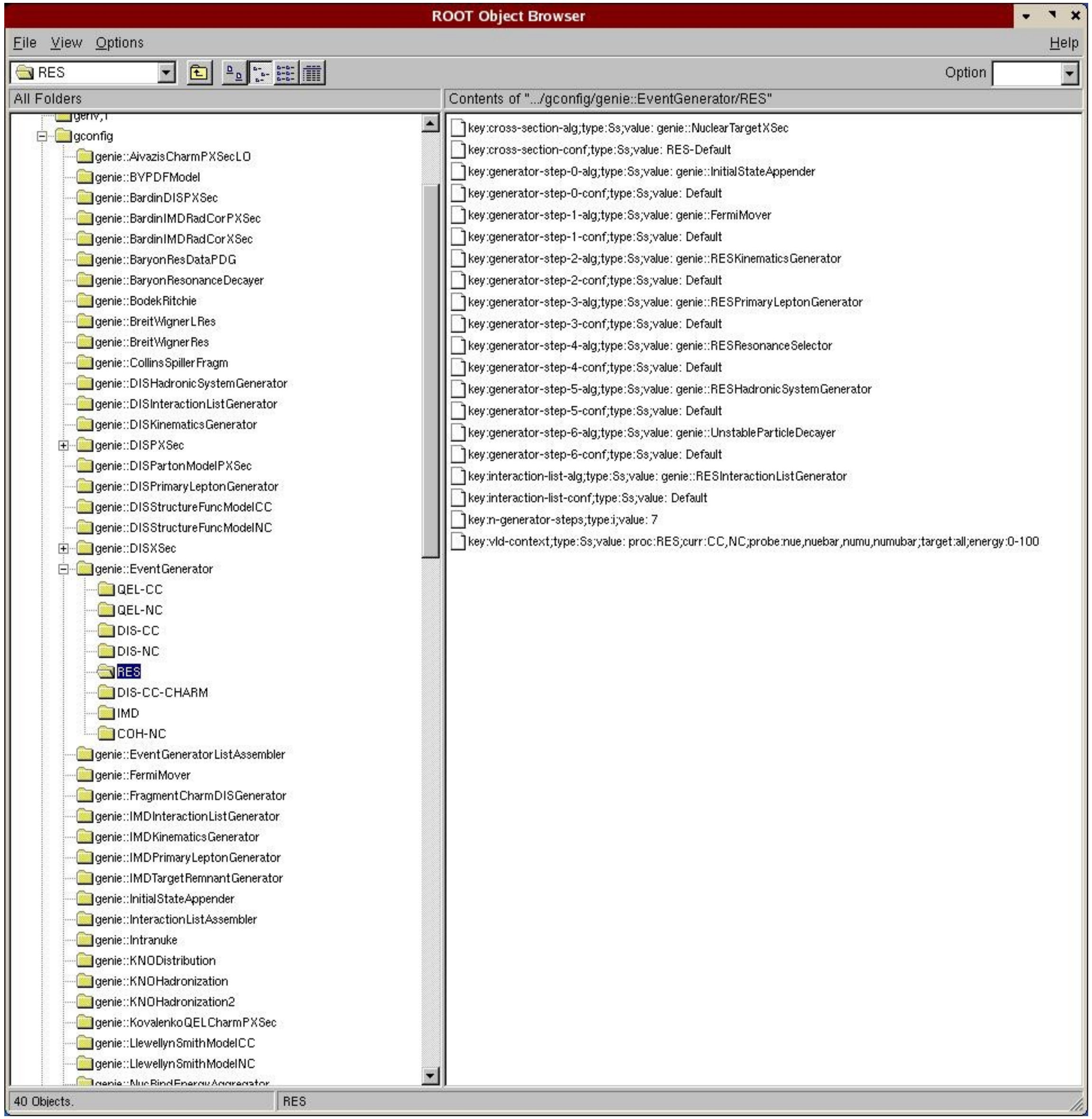
Open the file in a bare ROOT session

- Spot the event tree
- Use TBrowser to

- check the folder containing snapshots of the environment
- check the folder containing snapshots of the configuration

Activity.2

verify physics settings
used in event generation



Science & Technology Facilities Council

Activity.2

On completion:

Examine outputs

- Check the log file

  follow print-outs for 2-3 events / verify event generation steps

Science & Technology Facilities Council

Ntuple conversions

GENIE's native GHEP event ROOT files can be **converted** to a variety of

- summary ntuples for plotting basic physics quantities in bare ROOT sessions,
- text event files (tracker,...) used by various experiments (SK,...)
- bare-ROOT event files (rootracker) used by various experiments (nd280,...)
- ...

All the conversions can be performed using GENIE's gntpc utility

See $GENIE/src/stdapp/gNtpConv.cxx for the source code

gntpc syntax

In summary, the *gntpc* syntax is:

```
shell$ gntpc -i input_file [-o output_file] -f format [-n nev]
```

where [] denotes an optional argument.

Details on the utility options can be found below:

- **-n** Specifies the number of events to convert.

  By default, gntpc will convert all events in the input file.

- **-i** Specifies an input GENIE *GHEP* event tree file.

- **-f** Specifies the output file format.

  This can be any of the following strings:
  'gst', 'gxml', 'rootracker', 't2k_rootracker', 't2k_tracker', or 'nuance_tracker'.
  Details on these formats are given below.

  Additional formats are supported by *gntpc* but they are used mainly for GENIE testing purposes and will not be described here.

- **-o** Specifies the output file name.

Supported formats

-f specifies the output file format

Can be any of:

gst
A GENIE summary ntuple

rootracker
A STDHEP-like bare-ROOT event format

t2k_rootracker
The rootracker variance used by the T2K near detector MC. Contains additional flux pass-through info.

nuance_tracker
An emulation of the nuance output for interfacing GENIE with legacy systems

t2k_tracker
A tracker variance used for passing GENIE events into the SuperK detector MC. Contains additional flux pass-through info and event tweaks (K0,K0bar → K0l, K0s, skipping some intermediates and ions).

gxml
A XML event format

More conversions can be supported

The 'gst' format

A GENIE Summary Tree

An “old-style” flat ntuple with summary event info.

You can use `gst' ntuples to generate plots at a bare -ROOT session really easily.

The 'gst' ntuple

- **iev** (*int*): Event number.
- **neu** (*int*): Neutrino PDG code.
- **tgt** (*int*): Nuclear target PDG code (10LZZZAAAI).
- **Z** (*int*): Nuclear target Z.
- **A** (*int*): Nuclear target A.
- **hitnuc** (*int*): Hit nucleon PDG code (not set for COH, IMD and NuEL events).
- **hitqrk** (*int*): Hit quark PDG code (set for DIS events only).
- **sea** (*bool*): Hit quark is from sea (set for DIS events only).
- **resid** (*bool*): Produced baryon resonance id (set for resonance events only).
- **qel** (*bool*): Is it a QEL event?
- **res** (*bool*): Is it a RES event?
- **dis** (*bool*): Is it a DIS event?
- **cohpi** (*bool*): Is it a COH pion production event?
- **cohel** (*bool*): Is it a COH elastic event?
- **imd** (*bool*): Is it an IMD event?

continued...

The 'gst' ntuple

- **nuel** (*bool*): Is it a ve- elastic event?
- **cc** (*bool*): Is it a CC event?
- **nc** (*bool*): Is it a NC event?
- **charm** (*bool*): Produces charm?
- **neut_code** (*int*): The equivalent NEUT reaction code (if any).
- **nuance_code** (*int*): The equivalent NUANCE reaction code (if any).
- **wght** (*double*): Event weight.
- **xs** (*double*): Bjorken x (as was generated during the kinematical selection / off-shell kinematics).
- **ys** (*double*): Inelasticity y (as was generated during the kinematical selection / off-shell kinematics).
- **ts** (*double*): Energy transfer to nucleus at coherent events (as was generated during the kinematical selection).
- **Q2s** (*double*): Momentum transfer Q^2 (as was generated during the kinematical selection / off-shell kinematics) (in GeV^2).
- **Ws** (*double*): Hadronic invariant mass W (as was generated during the kinematical selection / off-shell kinematics).
- **x** (*double*): Bjorken x (as computed from the event record).
- **y** (*double*): Inelasticity y (as computed from the event record).
- **t** (*double*): Energy transfer to nucleus at COHPi events (as computed from the event record).
- **Q2** (*double*): Momentum transfer Q^2 (as computed from the event record) (in GeV^2).
- **W** (*double*): Hadronic invariant mass W (as computed from the event record).

continued...

The 'gst' ntuple

- **Ev** (*double*): Incoming neutrino energy (in GeV).
- **En** (*double*): Initial state hit nucleon energy (in GeV).
- **pxn** (*double*): Initial state hit nucleon px (in GeV).
- **pyn** (*double*): Initial state hit nucleon py (in GeV).
- **pzn** (*double*): Initial state hit nucleon pz (in GeV).
- **El** (*double*): Final state primary lepton energy (in GeV).
- **pxl** (*double*): Final state primary lepton px (in GeV).
- **pyl** (*double*): Final state primary lepton py (in GeV).
- **pzl** (*double*): Final state primary lepton pz (in GeV).
- **nfp** (*int*): Number of final state p and $\bar{p}$ (after intranuclear rescattering).
- **nfn** (*int*): Number of final state n and $\bar{n}$.
- **nfpip** (*int*): Number of final state π^+.
- **nfpim** (*int*): Number of final state π^-.
- **nfpi0** (*int*): Number of final state π^0.
- **nfkp** (*int*): Number of final state K^+.
- **nfkm** (*int*): Number of final state K^-.
- **nfk0** (*int*): Number of final state K^0 and $\bar{K}^0$.
- **nfem** (*int*): Number of final state γ, $e^-$and e^+.
- **nfother** (*int*): Number of heavier final state hadrons (D+/-,D0,Ds+/- ,Lamda,Sigma,Lamda_c,Sigma_c,...).

continued...

The 'gst' ntuple

- **nip** (*int*): Number of 'primary' ('primary' : before intranuclear rescattering) p and $\bar{p}$.
- **nin** (int): Number of 'primary' n and $\bar{n}$.
- **nipip** (*int*): Number of 'primary' π^+.
- **nipim** (*int*): Number of 'primary' π^-.
- **nipi0** (*int*): Number of 'primary' π^0.
- **nikp** (*int*): Number of 'primary' K^+.
- **nikm** (*int*): Number of 'primary' K^-.
- **nik0** (*int*): Number of 'primary' K^0 and $\bar{K^0}$.
- **niem** (*int*): Number of 'primary' γ, $e^-$and e^+ (eg from nuclear de-excitations or from pre-intranuked resonance decays).
- **niother** (*int*): Number of other 'primary' hadron shower particles.
- **nf** (*int*): Number of final state particles in hadronic system.
- **pdgf** (*int[kNPmax]*): PDG code of k^{th} final state particle in hadronic system.
- **Ef** (*double[kNPmax]*): Energy of k^{th} final state particle in hadronic system (in GeV).
- **pxf** (*double[kNPmax]*): Px of k^{th} final state particle in hadronic system (in GeV).
- **pyf** (*double[kNPmax]*): Py of k^{th} final state particle in hadronic system (in GeV).
- **pzf** (*double[kNPmax]*): Pz of k^{th} final state particle in hadronic system (in GeV).

continued...

The 'gst' ntuple

- **ni** (*int*): Number of particles in the 'primary' hadronic system ('primary' : before intranuclear rescattering).
- **pdgi** (*int[kNPmax]*): PDG code of k^{th} particle in 'primary' hadronic system.
- **Ei** (*double[kNPmax]*): Energy of k^{th} particle in 'primary' hadronic system (in GeV).
- **pxi** (*double[kNPmax]*): Px of k^{th} particle in 'primary' hadronic system (in GeV).
- **pyi** (*double[kNPmax]*): Py of k^{th} particle in 'primary' hadronic system (in GeV).
- **pzi** (*double[kNPmax]*): Pz of k^{th} particle in 'primary' hadronic system (in GeV).
- **vtxx** (*double*): Vertex x in detector coord system (in SI units).
- **vtxy** (*double*): Vertex y in detector coord system (in SI units).
- **vtxx** (*double*): Vertex z in detector coord system (in SI units).
- **vtxt** (*double*): Vertex t in detector coord system (in SI units).

Science & Technology Facilities Council

Examples using the 'gst' ntuple

1. To draw a histogram of the final state primary lepton energy for all ν_μ CC DIS interactions with an invariant mass $W > 3$ GeV, then type:
`root[0] gst->Draw("El","dis&&cc&&neu==14&&Ws>3");`

2. To draw a histogram of all final state π^+ energies in CC RES interactions, then type:
`root[0] gst->Draw("Ef","pdgf==211&&res&&cc");`

Activity.3

Convert the sample generated in Activity.2 into various formats and inspect.

Generate a gst ntuple, load it in ROOT and start looking at the variables

In the mean time generate a full event sample (all processes enabled) – see next

Activity.3

Let's generate 5k nu_mu+12C events at Ev = 1 GeV

shell% export GSPLOAD=/some/path/gxspl-c12-2.3.1.xml

shell% export GEVGL=Default *(or just 'unset GEVGL')*

shell% **gevgen** -p 14 -t 1000060120 -n 5000 -s *-r 2000* -e 1 | grep -i error

Activity.3

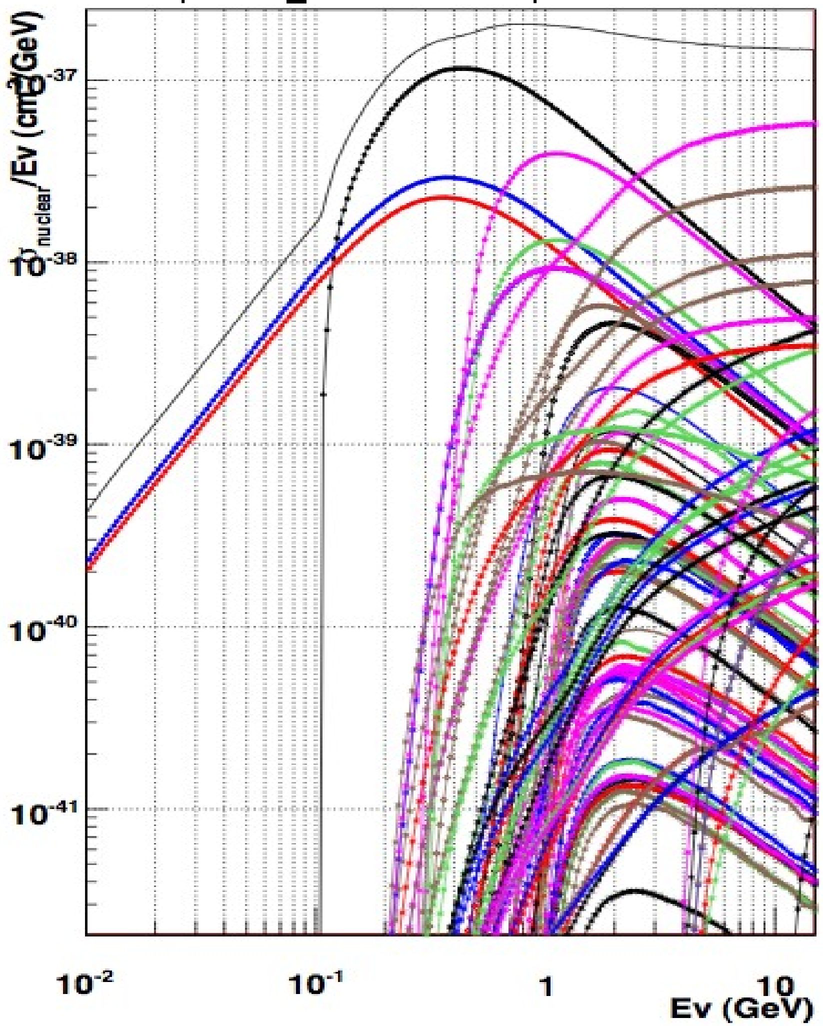
Convert that last GHEP event file (run: 2000) to gst format.

Check:

- Ratio of interaction types
- Selected kinematics for each interaction type
- Hadronic multiplicities for each interaction type (before/after hadron transport)
- ...

Using gmkspl's cross section outputs

Example: **nu_mu + O16:** All processes



gmkspl output is in XML format

- Convenient for moving data between GENIE applications

- Not convenient for the student who wants to get GENIE cross sections into his ROOT analysis macro.

You convert XML splines into ROOT format (graphs)

All extracted info can be stored into a single ROOT file.

Graphs for each neutrino+target combination are stored at a different ROOT directory

Each graph can be interpolated

That single ROOT file can contain all the GENIE cross section "functions" you need!

gspl2root syntax

***gspl2root*: utility to convert GENIE XML cross section splines into a ROOT format**

In summary, the *gspl2root* utility syntax is:

```
shell$ gspl2root -f xml_file -p nu -t tgt [-e emax] [-o root_file] [-w]
```

where [] denotes an optional argument.

Details on the *gspl2root* options can be found below:

- **-f** Specifies the input XML cross section spline file.
- **-p** Specifies the neutrino PDG code.
- **-t** Specifies the target PDG code (format: 10LZZZAAAI).
- **-e** Specifies the maximum energy for the generated graphs.
- **-o** Specifies the output ROOT file name.
- **-w** Instructs *gspl2root* to write-out plots in a postscipt file.

The spline data written out have the energy in GeV and the cross section in $10^{-38} cm^2$.

Science & Technology Facilities Council

gspl2root example

In order to extract all $\nu_\mu + n$, $\nu_\mu + p$ and $\nu_\mu + O^{16}$ cross section splines from the input XML file '*mysplines.xml*', convert splines into a ROOT format and save them into a single ROOT file '*xsec.root*', type:

```
shell$ gspl2root -f mysplines.xml -p 14 -t 1000000010 -o xsec.root
shell$ gspl2root -f mysplines.xml -p 14 -t 1000010010 -o xsec.root
shell$ gspl2root -f mysplines.xml -p 14 -t 1000080160 -o xsec.root
```

Same file;
Different directory

A large number of graphs (one per simulated process and appropriate totals) will be generated in each case. Each set of plots is saved into its own ROOT *TDirectory* named after the specified initial state.

The stored graphs can be used for cross section interpolation. For instance, the '*xsec.root*' file generated in this example will contain a 'nu_mu_O16' *TDirectory* (generated by the last command) which will include cross section graphs for all $\nu_\mu + O^{16}$ processes. To extract the $\nu_\mu + O^{16}$ DIS CC cross section graph for hit u valence quarks in a bound proton and evaluate the cross section at energy E, type:

```
root[0] TFile file(“xsec.root”,”read”);
root[1] TDirectory * dir = (TDirectory*) file->Get("nu_mu_O16");
root[2] TGraph * graph = (TGraph*) dir->Get("dis_cc_p_uval");
root[3] cout << graph->Evaluate(E) << endl;
```

Science & Technology Facilities Council

Activity.4

Give it a try!

Use the carbon cross section file you used in Activity.3
(contains free nucleon spines too)

Activity.5

Generate sample / Tweak physics / Generate another sample / Compare

Examine the contents of $GENIE/config

Every algorithm has a corresponding configuration file

All **"user physics options"** are in $GENIE/config/UserPhysicsOptions.xml

Examine its contents

Contains all the important stuff

That is the <u>only physics configuration we recommend messing around with</u>

Now, your task >>

Activity.5

Tweak params

Edit $GENIE/config/UserPhysicsOptions.xml

Change QEL-Ma from 0.99 GeV to 1.3 GeV and save

(you could modify any param and any number of params)

Generate sample with tweak params

shell& unset GSPLOAD

shell& export GEVGL=QEL-CC

shell& **gmkspl** -p 14 -t 1000060120 -n 100 -e 40 -o my_new_qelcc_splines.xml
shell& export GSPLOAD=my_new_qelcc_splines.xml

shell& **gevgen** -p 14 -t 1000060120 -n 5000 -e 1 *-r 1001* -s | grep -i error

Activity.5

Compare your last sample (run: 1001) with the nominal QEL-CC sample (run: 1000) generated at Activity.2

Plot Q2, and f/s lepton energy distributions

Using GENIE for MC production:

- Controlling the seed number
- Controlling log files

Activity.6

Setting the seed

Generate 3 events (run 5001): say, numu+C12 at 2 GeV - (sample A)
(Uses 'default' seed)

Generate another 3 events (run 5002) - (sample B)
(Uses 'default' seed)

Change the seed (other than its default value – find it in $GENIE/src/Conventions/Controls.h)

shell% export GSEED=8329839

Generate another 3 events (run 5003) - (sample C)

Print-out event records & verify that A,B are identical and C differs...

Activity.7

Controlling the output

Is important to know how to control what gets printed out

- Keep log file size under control for large MC productions
- Tweak mesg priorities / thresholds to get selected information printed-out

GENIE uses the log4cpp library

- Each message belongs to a particular “stream”
- Each message has a “priority level” (debug; info; notice; warn; error; fatal)
- Each stream has a threshold

  See $GENIE/config/Messenger.xml

Science & Technology
Facilities Council

Activity.7

Controlling the output

- Create a file similar to $GENIE/config/Messenger.xml
- List only the mesg streams whose thresholds you want to override
- Set your preferred thresholds
- Set the **GMSGCONF** environmental variable to point to that file

Generate few events and check the effect on the output

For productions:

Set the **GPRODMODE** environmental variable to YES

That sets all mesg stream priorities to `WARNING'
(minimal output; only in case of problems)

Generate few events and check the effect on the output

Activity.8

Generate events using a simple flux

Again using the **gevgen** event generation driver

You can specify the flux with the -f option. Can be either:

- *a functional form:* ... ... **-f 'x*x*exp((-x*x+3)/4)'** ... ...
- *an ascii file with 2 columns (E, flux)*: ... ... **-f /path/to/asciifile.data** ... ...
- *a histogram in a ROOT file:* ... ... **-f /path/to/file.root,hist_name** ... ...

If a flux is specified, then the GMCJDriver driver object is partially utilized
with a simple flux driver (1 neutrino species, flux from histogram) and a simple 'point geometry'

Activity.8

Let's generate nu_mu+12C for the T2K flux

shell& export GEVGL=Default

shell& export GSPLOAD=*/path/to/carbon/splines.xml*

Use that energy range of the input flux

shell& **gevgen** -p 14 -t 1000060120 -n 5000 -s -e 0,20 -r 3000 (cmd continues at next line)

-f $GENIE/data/flux/t2kflux.root,h30000 | grep -i “error\|warn”

Use flux from histogram h30000 taken from file t2kflux.root

Analyze the sample & generate plots

Check event break-down, average multiplicities, kinematics – as generated for t2k flux

Activity.9

Verify the correct energy dependence of your last event sample

Remember:

generated_sample(E) ~ flux(E) x cross_section(E)

Activity.9 / cheat sheet

```
{
  TFile fflux("/path/to/the/input/flux/file.root","read");
  TH1D * hflux = (TH1D*) fflux.Get("h30000");

  TFile fsig("/path/to/the/gspl2root/splines.root","read");
  TDirectory * curr = (TDirectory *) fsig.Get("nu_mu_O16");
  TGraph * tot_cc = (TGraph*) curr->Get("tot_cc");

  TFile gstfile("/path/to/the/gst/summary/ntuple.root","read");
  TTree * gst = (TTree*)gstfile.Get("gst");

  hflux->SetLineColor(2);
  hflux->Scale(1. / hflux->Integral("width"));
  hflux->Draw();

  TH1D * evt_cc = new TH1D("evt_cc","",100,0,16);
  gst->Draw("Ev>>evt_cc","cc", "goff");
  evt_cc->Scale(1. / evt_cc->Integral("width"));
  evt_cc->Draw("same");

  TH1D * hrate = new TH1D("hrate","", hflux->GetNbinsX(), hflux->GetXaxis()->GetXmin(), hflux->GetXaxis()->GetXmax());

  for(int i=1; i<= hflux->GetNbinsX(); i++) {
       double e = hflux->GetBinCenter(i);
       double y = hflux->GetBinContent(i);
       y *= tot_cc->Eval(e);
       hrate->SetBinContent(i,y);
  }
  hrate->SetLineColor(4);
  hrate->Scale(1. / hrate->Integral("width"));
  hrate->Draw("same");
}
```

The `event loop'

In past activities

you learned how to get information out of the generated event tree

by relying on GENIE (gntpc utility) to

analyse the event sample and generate a summary ntuple.

We will learn how to write our own 'event loop' :

You can extract any event info you want; Can use the example event loop to push GENIE particles through your detector simulation.

The `event loop' skeleton

```
{
  ...
  // Open the GHEP/ROOT file
  string filename = /data/sample.ghep.root;
  TFile file(filename.c_str(), READ);

  // Get the tree header & print it
  NtpMCTreeHeader * header =
    dynamic_cast<NtpMCTreeHeader*> (file.Get("header"));
  LOG(test, pINFO) << *header;

  // Get the GENIE GHEP tree and set its branch address
  TTree * tree = dynamic_cast<TTree*> (file.Get(gtree));
  NtpMCEventRecord * mcrec = 0;
  tree->SetBranchAddress(gmrec, &mcrec);

  // Event loop
  for(Long64_t i=0; i<tree->GetEntries(); i++){
    tree->GetEntry(i);

    // print-out the event
    EventRecord & event = *(mcrec->event);
    LOG(test, pINFO) << event;

    // put your event analysis code here
    ...
    ...


    mcrec->Clear();
  }
  ...
}
```

Examples of programs looping over events and extracting event information (to use as templates / copy code from):

$GENIE/src/stdapp/gNtpConv.cxx

Activity.10

Write your own event loop and extract some information

- Create a new directory in your home area
- Copy $GENIE/src/test/testEventLoop.cxx there
- Copy $GENIE/src/test/Makefile there
- **Start tweaking the event loop to print out X,Y or save the X,Y histogram (your choice)**
- Edit the Makefile, find the 'gtestEventLoop' target and change `**-o $(GENIE_BIN_PATH)/gtestEventLoop**' to `**-o gtestEventLoop**'
- Build: shell% **gmake gtestEventLoop**
- Run: shell% **./gtestEventLoop** -f /path/to/an/event/file.root

Science & Technology Facilities Council

GENIE as a hadron-nucleus event generator

Driver to generate hadron-nucleus interactions (*ghAevgen*)

Used in validating intranuclear hadron transport model against hadron data.

Syntax:

ghAevgen -n nevents **-p** hadron_code **-t** nucleus_code **-r** run_num
-k hadron_kinetic_energy [**-f** flux]

Science & Technology Facilities Council

ghAevgen examples

100k pi+ Fe56 events, pion kinetic energy = 165 MeV

ghAevgen -n 100000 -p 211 -t 1000260560 -k 0.165

100k pi+ Fe56 events, pion kinetic energy uniformly distributed in 165 MeV, 800 MeV

ghAevgen -n 100000 -p 211 -t 1000260560 -k 0.165,0.800

100k pi+ Fe56 events, pion kinetic energy in the 165 MeV, 800 MeV range with an 1/E^2 dependence

ghAevgen -n 100000 -p 211 -t 1000260560 -k 0.165,0.800 -f '1/(x*x)'

Activity.11

Generate and analyse a pi+ Fe56 event file

- Generate 10k pi+ Fe56 events for pi+ kinetic energy distributed uniformly between 100 and 1000 MeV

- Use **gevdump** and print-out the first 100 events

- Modify the event loop from Activity.10 to analyse the pi+ Fe56 sample

- Your goal is to find the fraction of **pi production, charge exchange** and **'no re-interaction'** events as a function of the kinetic energy.

- **Hint:** Use the INukeFateHA_t ReconstructHadronFateHA (GHepRecord * event, int i, bool hA_mode=false); function from $GENIE/src/HadronTransport/INukeUtils.h/cxx to “reconstruct” the hadron “fate”

Using arbitrarily complex neutrino fluxes and detector geometry descriptions

Event Generation Drivers

gevgen we used extensively at previous activities is just a front-end program

Gets user inputs, sets up a job & calls the right **driver objects**

[GEVGDriver]

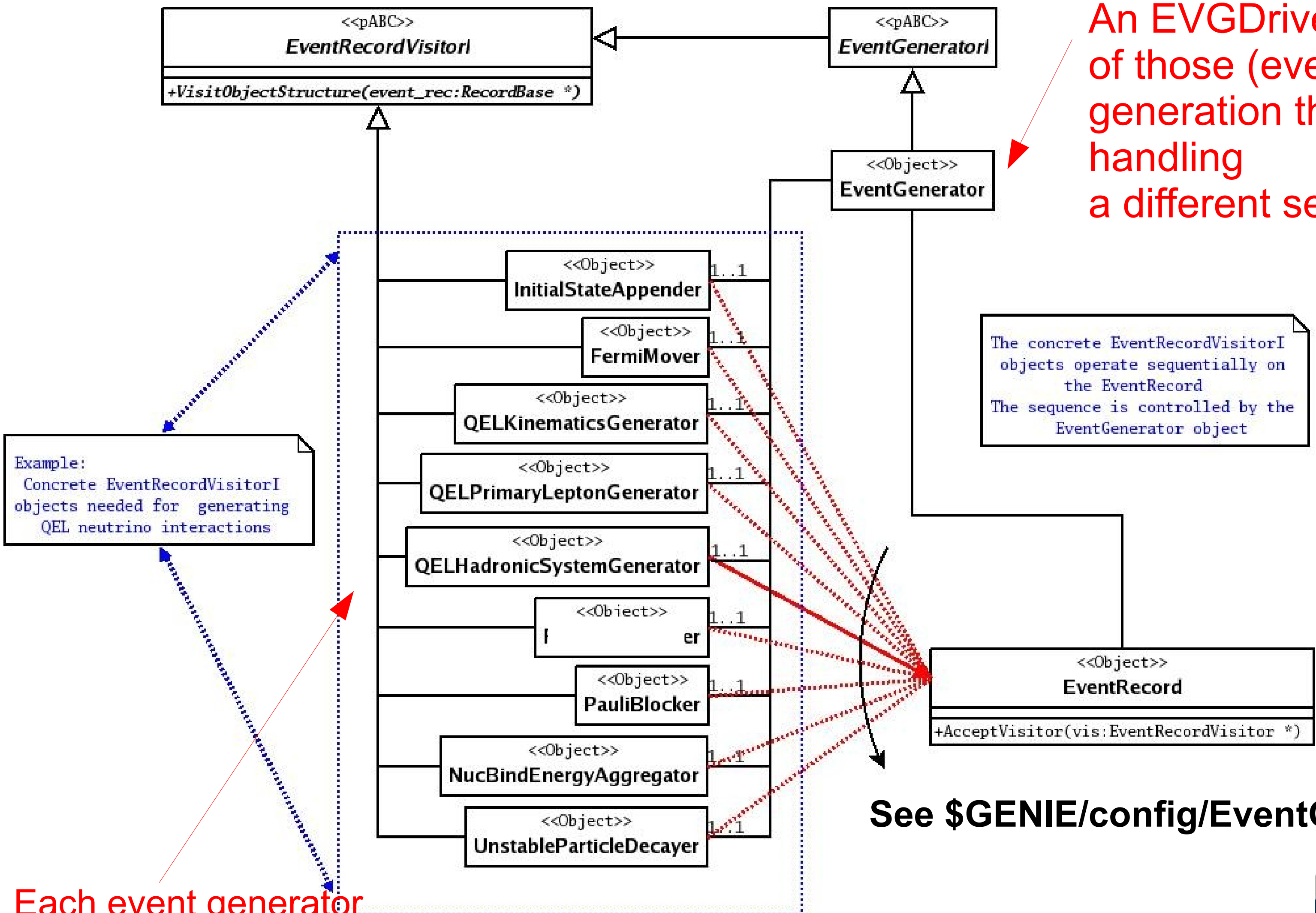
- Provides event generation capabilities for a given initial state.
- Properly configured for that state
- Knows <u>all</u> neutrino interaction physics for that initial state
- Knows <u>all</u> cross section models associated with each process
- Knows which event generation thread to call to generate any process
- Coordinates process selection
- Bootstraps event generation
- Quite seriously complex – but very transparent

[GMCJDriver]

- Provides a more "extended" functionality: event generation for flux & realistic geometries
- Handles fluxes / geometries via standardized interfaces
- Owns a list of GEVGDriver objects for each possible initial state
- Knows very little physics by its own (purely generation mechanics)

See $GENIE/src/EVGDrivers/

Event Generation Threads and Modules



An EVGDriver holds many of those (event generation threads), each handling
a different set of processes

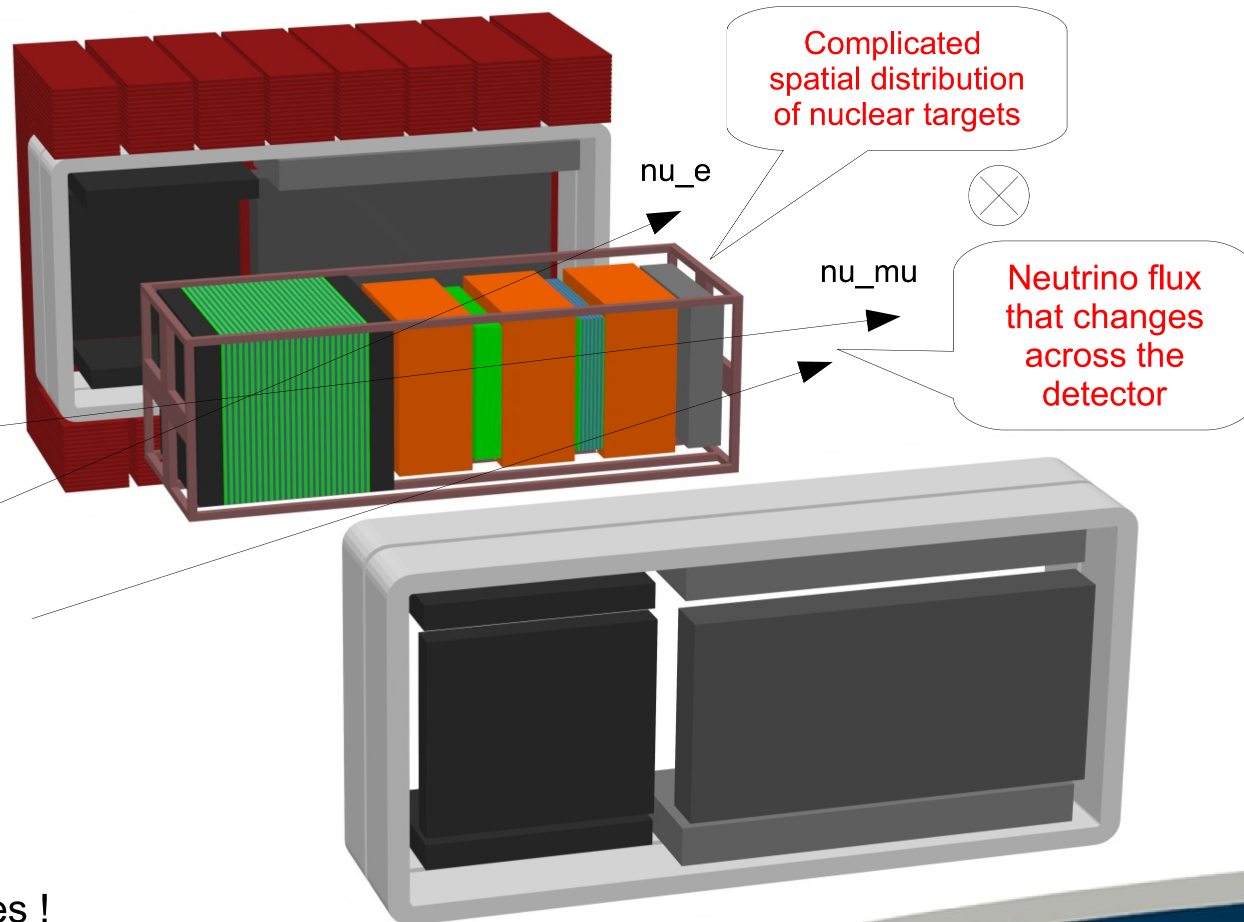
See $GENIE/config/EventGenerator.xml

Each event generator thread can have multiple processing steps

Using fluxes / geometries

Event generation:
A complicated convolution of things:

Neutrino generator's job is:
Generate an event
once it is handed over
an initial state
(nu + target, at a given energy)



The **problem** here is
how to select that initial state
(and take into account its energy and spatial dependence)

Eg in MINOS:
6 neutrino flavours X
~60 (!) isotopes in detector geom =
360 possible initial states

BTW, the generator
should handle all these initial states !

Hooks for fluxes and geometries

The GFluxI interface

To describe a flux, *one should* ***subclass GFluxI*** *and* ***provide an object that "knows" how to:***

- generate 'flux' neutrinos
  - ***pdg code,***
  - ***position 4-vector***
    (at a volume surrounding the detector)
  - ***4-momentum***
- declare all neutrino species it can generate
- declare the maximum neutrino energy
- can generate more?

The GeomAnalyzerI interface

To describe a geometry description, *one should* ***subclass GeomAnalyzerI*** *and* ***provide an object that "knows" how to:***

- find all materials in the geometry
- compute path lengths for all materials crossed by any flux neutrino
- *compute the maximum path length for all geometry materials*
- *generate vertices in a selected material along a specified neutrino directions*

Using fluxes and geometries

All it takes to generate neutrinos for arbitrarily complex flux & geometry **(conforming to the GFluxI and GeometryAnalyzerI interface)** is:

GMCJDriver * mcj_driver = new GMCJDriver;

mcj_driver->**UseFluxDriver**(flux_driver);

mcj_driver->**UseGeomAnalyzer**(geom_driver);

mcj_driver->Configure();

Flux Drivers

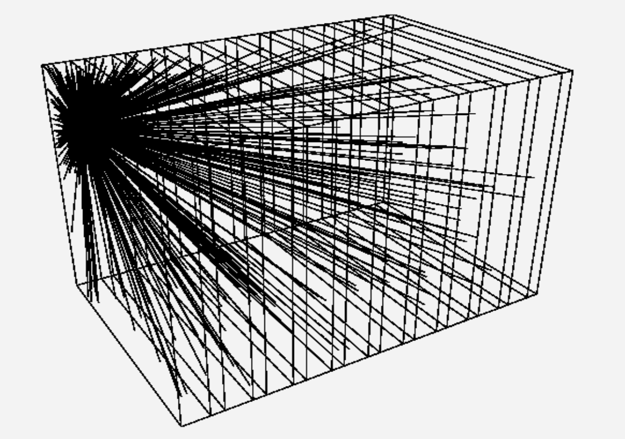
- *GJPARCNuFlux*: An interface to the JPARC neutrino beam simulation [8] used at SK, nd280, and INGRID.
- *GNuMIFlux*: An interface to the NuMI beam simulations [9] used at MINOS, NOvA, MINERvA and ArgoNEUT.
- *GBartolAtmoFlux*: A driver for the Bartol atmospheric flux by G. Barr, T.K. Gaisser, P. Lipari, S. Robbins and T. Stanev (cite)
- *GFlukaAtmo3DFlux*: A driver for the FLUKA-based 3-D atmospheric neutrino flux by A. Ferrari, P. Sala, G. Battistoni and T. Montaruli [?]
- *GCylindTH1Flux*: A generic flux driver, describing a cylindrical neutrino flux of arbitrary 3-D direction and radius. The radial dependence of the neutrino flux is configurable (default: uniform per unit area). The flux driver may be used for describing a number of different neutrino species whose (relatively normalised) energy spectra are specified as ROOT 1-D histograms. This driver is being used whenever an energy spectrum is an adequate description of the neutrino flux.
- *GMonoEnergeticFlux*: A trivial flux driver throwing mono-energetic flux neutrinos along the +z direction. More that one neutrino species can be included, each with its own weight. The driver is being used in simulating a single initial state at a fixed energy mainly for probing, comparing and validating neutrino interaction models.

Geometry Drivers

- *ROOTGeomAnalyzer*: A geometry driver handling detector geometries specified using ROOT. As detector geometries specified using Geant4 or *GDML* can be converted into ROOT geometries, this driver is being used in all cases where a detailed detector geometry is being passed on to GENIE.
- *PointGeomAnalyzer*: A trivial geometry corresponding to a single nuclear target or a target mix (a set of nuclear targets each with its corresponding weight fraction) at a fixed position. This driver is being used to simulate only given initial states as a means for probing the neutrino interaction physics modeling or in experimental situations where the detector is being illuminated by a spatially uniform neutrino beam and where the generated interaction vertices do not have any spatial dependence and can be generated uniformly within volumes of given nuclear targets.

ROOT/Geant4 Geometry Driver

(Fig. ray-tracing in a Lar-Pb slab detector scanning for maximum path lengths)



Flux driver fully validated in preparation for v2.4.0 and the T2K MDC0
(Jim Dobson, Pawel Guzowski)

New in v2.6.0: Ability to generate events in parts of the geometry
(Pawel Guzowski; Jacek Holeczek)

Navigation code clearly not optimum as pointed out by Robert Hatcher.
Speed improvements – in the pipeline.

Selecting initial states

GENIE uses the input flux driver to **throw flux neutrinos**

For **<u>each flux neutrino</u>**, GENIE computes **interaction probabilities** **<u>for each isotope</u>**

Remember,
for <u>each</u> (flux neutrino + isotope) pair:

Total interaction cross section
for given neutrino + isotope,
at given E
(get this from generator's own
cross section 'libraries')

Path length X density
for given isotope,
along the current flux neutrino direction
integrated across the detector
(get this from a "geometry driver")

Interaction
probability

$$P \sim \frac{\sigma L \varrho}{A}$$

Selection of initial state is based
on these interaction probabilities

Selecting initial states:
The <u>no-interaction</u> probability

Obviously, **interaction probabilities are very small numbers**

Of course the generator doesn't have to throw zillions of flux neutrinos to get an interaction.
<u>Probabilities are scaled-up</u> to reduce the number of trials

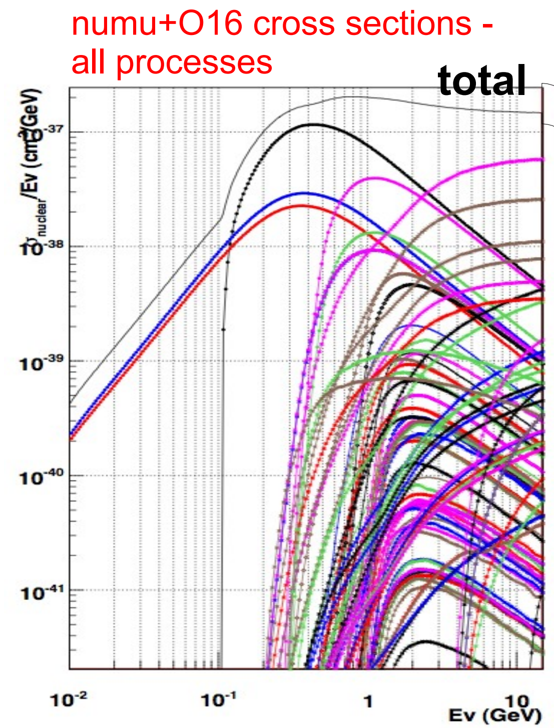
Probability scale is the maximum interaction probability
(i.e. Probability at maximum energy --so, max cross section-- and for the maximum possible path length)
summed over initial states

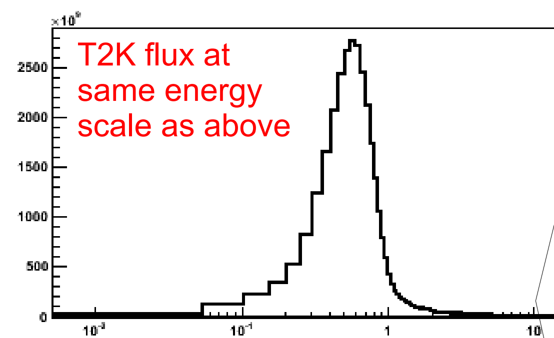
The probability scale is determined at an energy with low flux.

This is the source of the typically large no interaction probability.

eg. election inefficiency for nd280 event generation: rejecting ~500 flux neutrinos / interaction

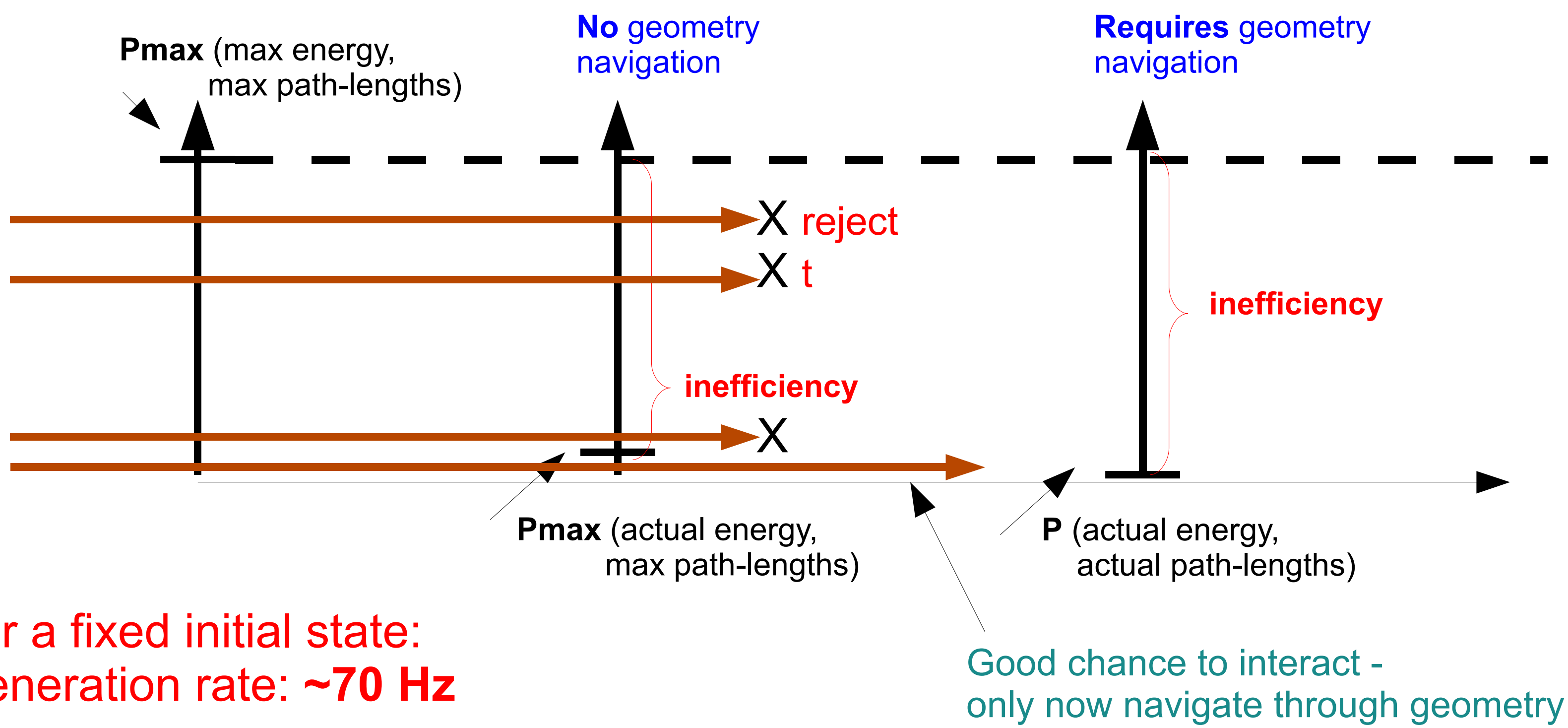
Irreducible / but use speed improvement tricks (see later)





Speed improvement trick

Most of the selection inefficiency at T2K / MINOS etc is due to neutrinos having E << spectrum tail, not due to path lengths << max path lengths
Can reject most flux neutrinos assuming max path lengths (no geometry navigation)



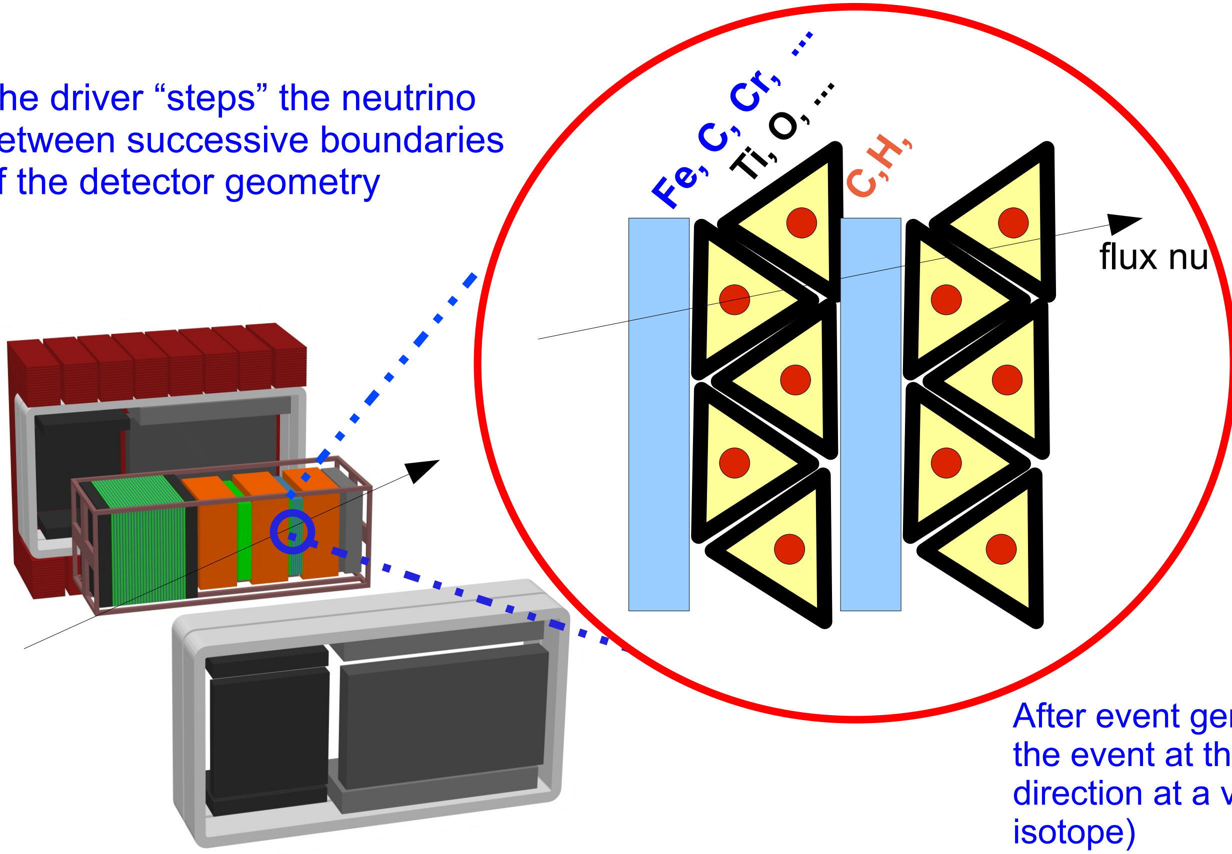
For a fixed initial state:
Generation rate: **~70 Hz**

For **realistic** nd280 geom & flux:
Generation rate: **~6 Hz**

Navigating the detector geometry

A "geometry driver" propagates neutrino through the detector, compute neutrino path lengths within each isotope.

The driver "steps" the neutrino between successive boundaries of the detector geometry



After each step, increments the density-weighted path-length for the appropriate isotope.

For compound materials, each step counts towards the integrated density-weighted path-length of many isotopes (taking also into account its weight fraction in the compound material)

After event generation, the geometry driver shifts the event at the right position (along the neutrino direction at a volume containing the selected isotope)

Doesn't have to stick with ROOT geometries only.
Any odd geometry can be integrated in GENIE event gen. drivers as long as an appropriate geometry driver can be supplied.

gT2Kevgen
An event generation driver customized for T2K

gT2Kevgen can be used as a template for building customized drivers for other experiments

A similar driver for the NuMI expts is in devel; Is using the NuMI flux driver contributed by Robert Hatcher. Will need your help in validation.

An event generation driver customized for T2K

```
\brief   A GENIE event generation driver 'customized' for T2K.

         This driver can handle the JPARC neutrino flux files generated by
         jnubeam and use the realistic detector geometries / target mix of
         T2K detectors. It can be used for full-blown GENIE event generation
         in nd280, 2km and SuperK.

         ** Syntax:

           gT2Kevgen
                   [-h]
                   [-r    run#]
                    -f    flux
                    -g    geometry
                    [-p  pot_normalization_of_flux_file]
                    [-t   top_volume_name_at_geom]
                    [-m max_path_lengths_xml_file]
                    [-L  length_units_at_geom]
                    [-D density_units_at_geom]
                    [-n  n_of_events]
                    [-c  flux_cycles]
                    [-e, -E exposure_in_POTs]
                    [-o  output_event_file_prefix]

... ...
... ... detailed description of inputs
... ...
```

Extract from the code documentation

See $GENIE/src/support/t2k/EvGen/gT2KEvGen.cxx for more details

gT2Kevgen geometry descriptions

- A ROOT file containing a ROOT/Geant4-based geometry description. This is the standard option for generating events in the nd280, 2km and INGRID detectors.

  Example:
  To use the ROOT detector geometry description stored in the '*nd280-geom.root*' file, type:
  '`-g /some/path/nd280-geom.root`'

  By default the entire input geometry will be used. Use the '`-t`' option to allow event generation only on specific geometry volumes.

- A mix of target materials, each with its corresponding weight. This is the standard option for generating events in the Super-K detector where the beam profile is uniform and distributing the event vertices uniformly in the detector volume is sufficient. The target mix is specified as a comma-separated list of nuclear PDG codes (in the PDG2006 convention: 10LZZZAAAI) followed by their corresponding weight fractions in brackets, as in:
  '`-t code1[fraction1],code2[fraction2],...`'

  Example 1:
  To use a target mix of 95% O^{16} and 5% H type:
  '`-g 1000080160[0.95],1000010010[0.05]`'

gT2Kevgen flux descriptions

-f Specifies the input 'neutrino flux'.

This option can be used to specify any of:

- A JNUBEAM flux ntuple and the detector location. The general sytax is:
  `'-f /path/flux_file.root,detector_loc'`

  The detector location can be any of 'sk' or the near detector positions 'nd1',...,'nd6' simulated by JNUBEAM. The ntuple has to be in ROOT format and can be generated from the distributed HBOOK ntuples using ROOT's *h2root* utility.

  When a JNUBEAM ntuple is used for describing the neutrino flux, GENIE is able to calculate the POT exposure for the generated event sample and any one of the exposure setting methods ('`-e`', '`-E`', '`-c`', '`-n`', see below) can be used.

  All JNUBEAM information on the flux neutrino parent (parent PDG code, parent 4-position and 4-momentum at the production and decay points etc) are stored in a 'flux' branch of the output event tree and is associated with the corresponding generated neutrino event.

  Example 1:
  To use the Super-K JNUBEAM flux ntuple from the '*/t2k/flux/jnubeam001.root*' file, type:
  `'-f /t2k/flux/jnubeam001.root,sk'`

- A set of flux histograms stored in a ROOT file. The general syntax is:
  `'-f /path/file.root,nu_code[histo],...'`

  Example:
  To use the histogram 'h1' (representing the ν_μ flux) and the histogram 'h2' (representing the ν_e flux) from the '*/data/flux.root*' file, type:
  `'-f /data/flux.root,14[h1],12[h2]'`

gT2Kevgen / examples

nd280 / 2km / ingrid

Generate events for **nd280** ('**nd5**' according to the flux file conventions), using
the /data/t2k/flux/07a/jnubeam001.root flux file and
the detailed geometry definition from /data/t2k/geom/nd280.root
Exit after generating so many events corresponding to 1E+16 POT

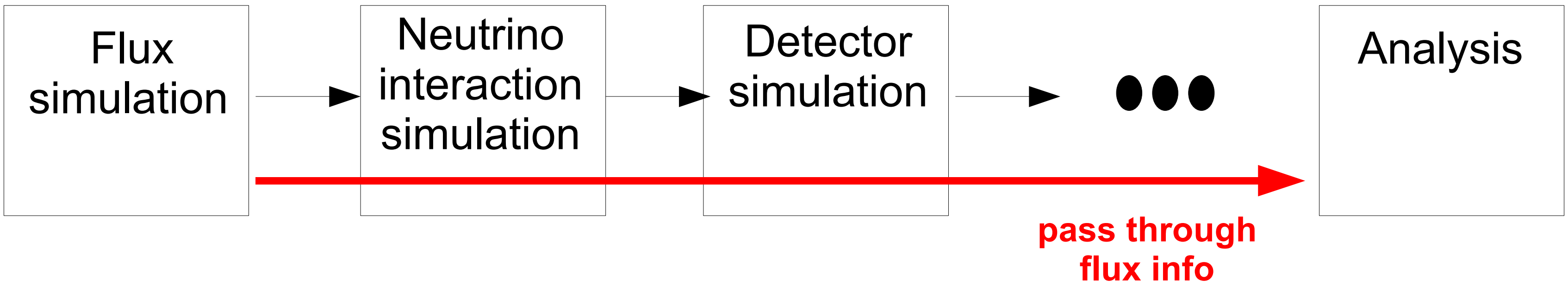
shell% **gT2Kevgen**
-r 2001 -E 1E+16
-f /data/t2k/flux/07a/jnubeam001.root,nd5
-g /data/t2k/geom/nd280.root **-L** mm **-D** clhep_def_density_unit

SuperK

Generate events for **SK**, using flux histograms from
the /data/t2k/flux/07a/flux.root file (histogram h1 for nu_e, h2 for nu_e_bar and h3 for nu_mu) and
a 95% O16 + 5% H target mix
Exit after generating 10000 events

shell% **gT2Kevgen**
-r 1001 -n 10000
-f /data/t2k/flux/07a/flux.root,12[h1],-12[h2],14[h3]
-g 1000080160[0.95],1000010010[0.05]

Pass through flux information



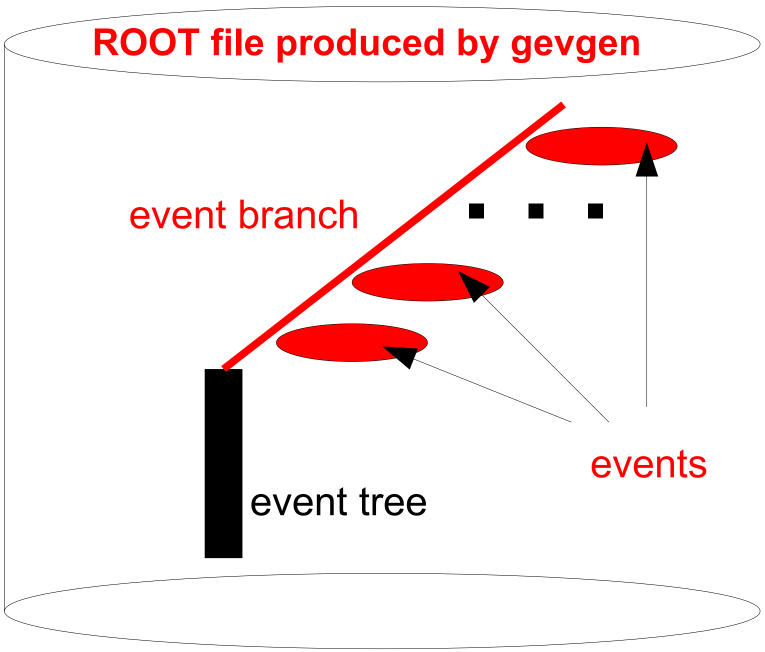
Neutrino generator sees flux simulation

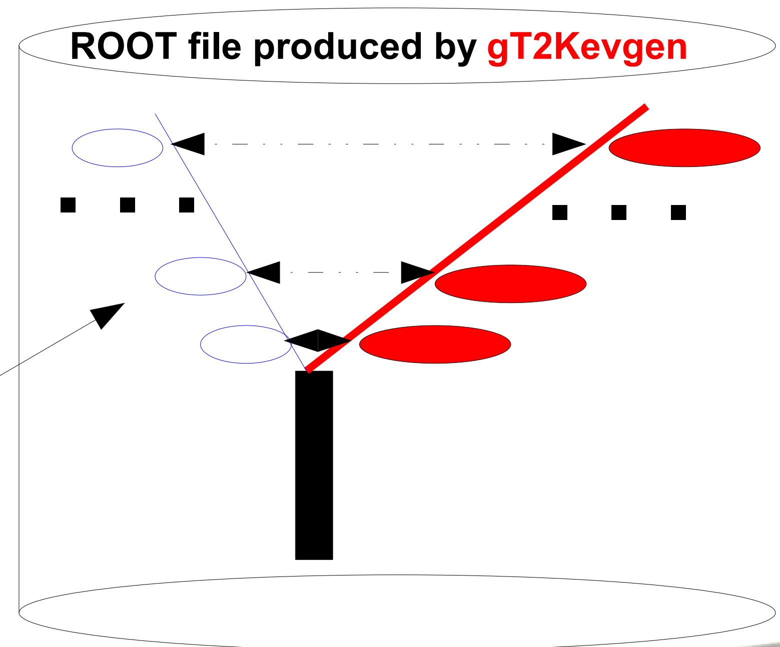
Neutrino generator has no use for certain flux information (eg, parent pion decay kinematics)

But the **neutrino generator** should **pass through all the flux info needed by analyses**

Pass through flux information

Modified the code writing out the GENIE event tree.
User-defined branches
with info linked to each generated event
(in this case flux info) can be trivially added.





flux branch
(complete neutrino parent
info for each generated T2K
event)

Activity.12

Generate events using the detailed nd280 detector geometry
and the jnubeam flux simulation ntuples

Find example inputs in
http://hepunx.rl.ac.uk/~candreop/outbox/genie/fermilab/

Full list of T2K cross section splines in
http://hepunx.rl.ac.uk/~candreop/generators/GENIE/data/ (use v2.5.1)

Generate 1E+16 POT worth of events in nd280

```
shell% export GSPLOAD=/path/to/splines.xml
shell% unset GEVGL
shell% gT2evgen -g /path/to/geom.root -f /path/to/flux.root,nd5
                -L mm -D chlep_def_density_unit -e 1E+16
```

Examine outputs

Discuss applications to your experiment

Event reweighting

Event reweighting

Use one sample to emulate another...

Can be used to propagate vA uncertainties to analyses, bug-fix precious large samples

2 popular use-cases

- **Reweight from a fixed set of {models/configuration} A to another fixed set B**

  eg reweight a generated sample to an improved / bug-fixed release

- **Given a set of models, reweight for changes in the configuration**

  eg, given QEL model, propagate effect of Ma uncertainty

What can be reweighed ?

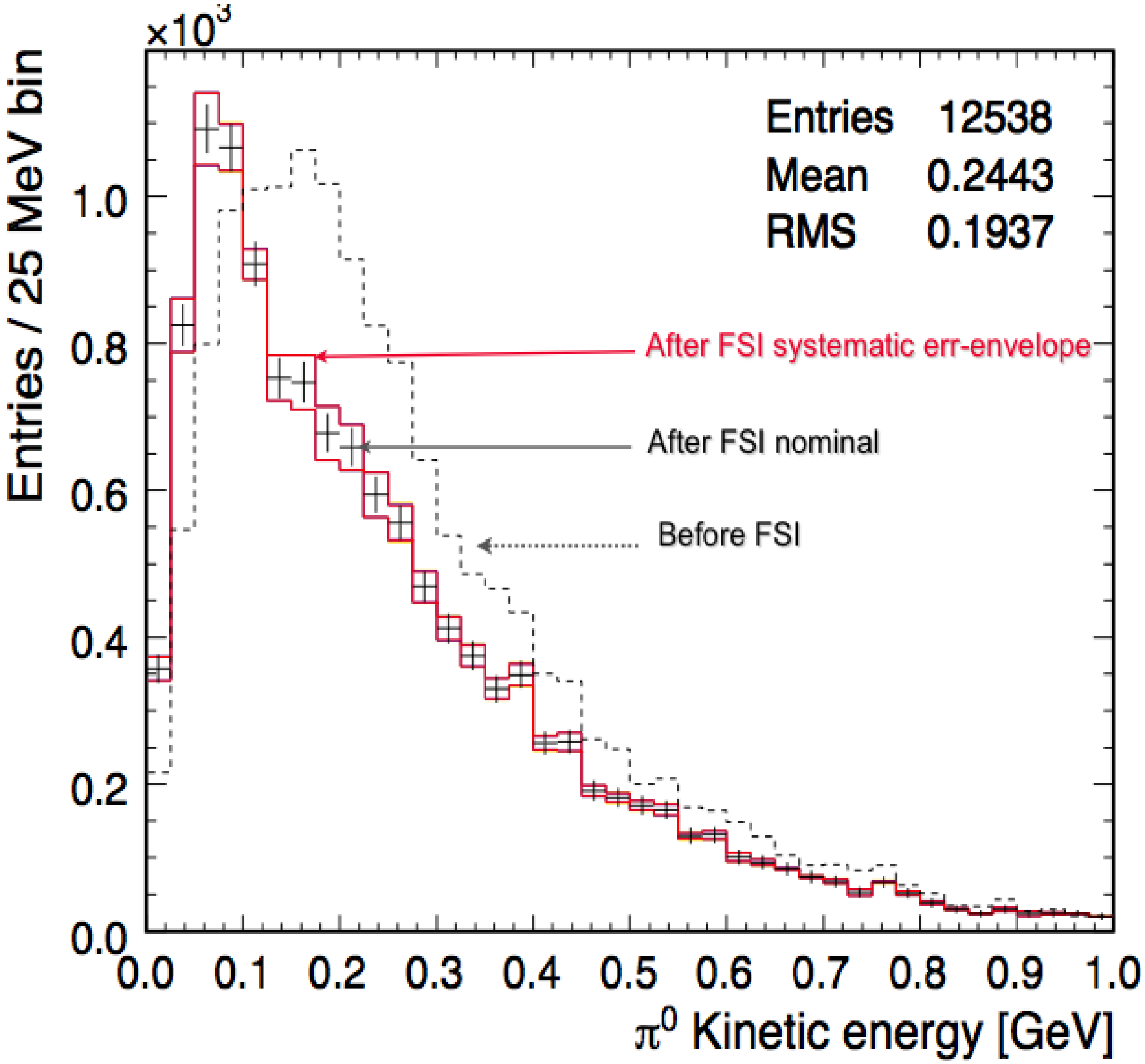
- **Quite easily / generically**
  - *The cross section model*
- **Less easily / generically but perfectly doable**
  - *Many aspects of the hadronization model*
  - *Many aspects of the Intranuclear hadron transport model*
- **Not easily or not at all doable**
  - *Nuclear model?*
  - *Cascades?*
  - *External (black-box) packages – eg JETSET, FLUKA*

What can be reweighed ?

There is a draft t2k document (reweighting.pdf) at
http://hepunx.rl.ac.uk/~candreop/outbox/genie/fermilab/

We will try to make that work available through GENIE

Hadron transport reweighing: example



NC 1pi0

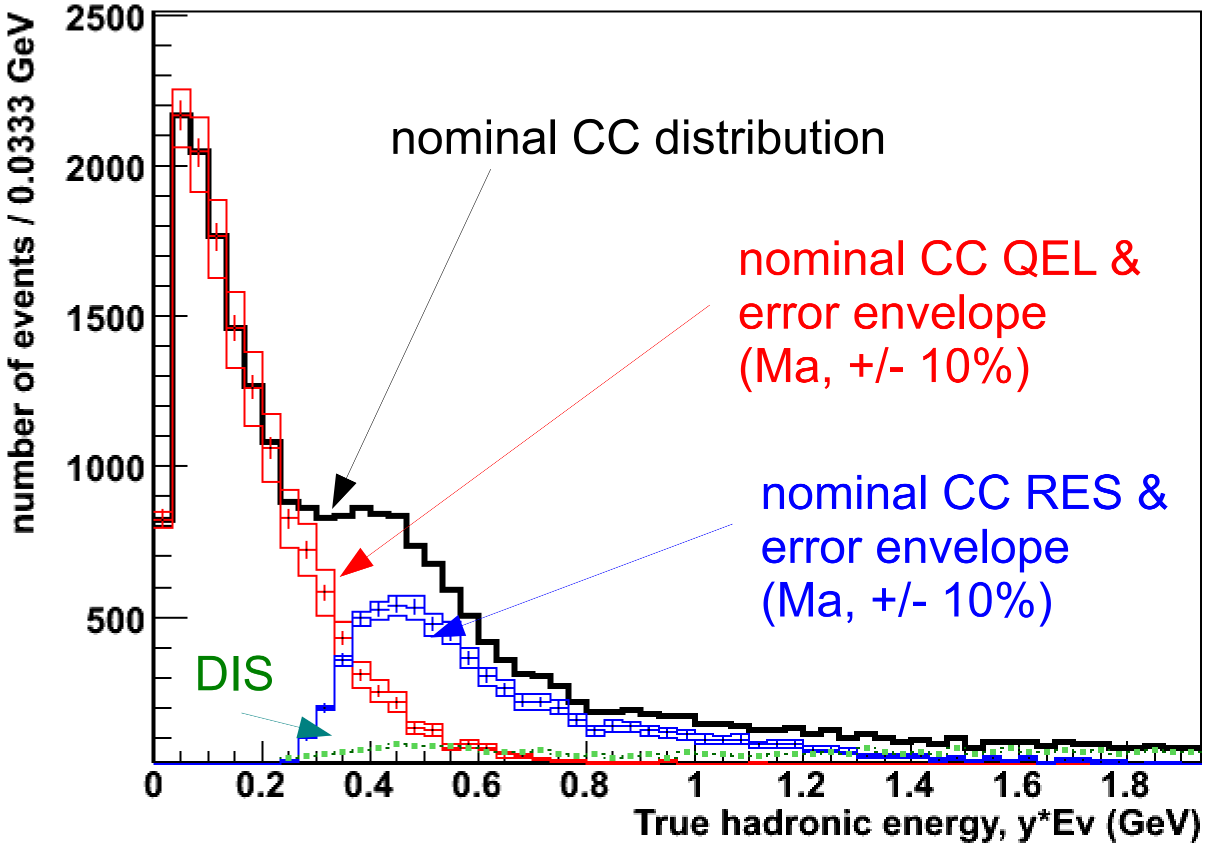
numu+O16

[Jim Dobson]

Cross section reweighing: example

Took single file (40k events) from the GENIE numu+O16 sample generated for January PAC

Tweaked Ma-QEL and Ma-RES by +/- 10%



However a basic cross section tool already exists within GENIE

How does cross section reweighting works in GENIE?

1st use case

```
...

// create a weight calculator
ReWeightCrossSection wcalc;

...

// event loop
for(int i = 0; i< tree->GetEntries(); i++)
{
   tree->GetEntry(i);

   EventRecord &  event = *(mcrec->event);

  // reweight the event
  double wght = wcalc.NewWeight(event);

  ... ...
 }
```

The cross section reweighing tool

get next mc record from
the event tree

get event record

Reweight to current settings
(set in std genie config files)
Any internal weight is taken into account

How does cross section reweighting works in GENIE?

2nd use case

```
... ...
// access the algorithm factory and the configuration pool
AlgFactory *  algf = AlgFactory::Instance();
AlgConfigPool * algc = AlgConfigPool::Instance();

// get the physics parameters that GENIE developers find meaningfull for a user to tweak
Registry * user_conf = algc->GlobalParameterList();
user_conf->UnLock();

for(/* some_parameter_loop */)
{
 user_conf->Set("QEL-Ma", some_ma_value);
 algf->ForceReconfiguration();

 LOG("test", pINFO) << "User options / current " << *user_conf;

 //event loop
 for(int i = 0; i< tree->GetEntries(); i++)
 {
  tree->GetEntry(i);
  EventRecord &  event = *(mcrec->event);

  // reweight the event
  double wght = wcalc.NewWeight(event);

 } // event loop
} // physics parameter loop
```

Update params & reconfigure
Valid params names can be found in
$GENIE/config/UserPhysicsOptions.xml

Get event & reweight
Any internal weight is taken into account

Activity.13

Reweight a generated sample –
Compare with a sample generated with tweaked params

Your task:

Reweight:
Ma: 0.99 -> 1.3

Nominal QEL sample
from Activity.1
(Ma=0.99)

Reweighted
Activity.1 sample

Compare with

Tweaked QEL sample
from Activity.6
(Ma=1.3)

If the reweighing works then the sample generated with Ma=1.3 GeV and the sample generated with Ma=0.99 GeV nut reweighed to 1.3 GeV should be ~ identical (modulo statistics)...

Activity.13

Now please proceed with your activity.

There are 2 reweighting test programs in $GENIE/src/test
called:
testReWeightP2P.cxx (recall 1st reweighting use case)
testReWeightLoop.cxx (recall 2nd reweighting use case)

Feel free to tweak them

---

1 - We want to compare true Q^2 & final state lepton E distributions

2 - Can you assess the effect of Ma tweaking in the “experiment-like” energy and Q2 reconstruction for v QELCC events?

cheat sheet:

$$E_\nu = \frac{m_N E_\mu - m_\mu^2/2}{m_N - E_\mu + p_\mu cos\theta_\mu} \qquad Q^2 = -2E_\nu(E_\mu - p_\mu cos\theta_\mu) + m_\mu^2$$

Accessing algorithms directly

In past activities we relied on GENIE utilities to access all information we needed

We will learn how to access algorithms (eg a cross section model) directly

This might be necessary of you have a specialized need
(for which no standard GENIE tool exists)

- All Algorithms are accessed by an Algorithm factory
  - specified by name and configuration
  - come ready for use (initialized / configured)
- Usually accept an *Interaction* object as argument

Activity.14

Accessing a cross section algorithms directly

- Get example code from:
  http://hepunx.rl.ac.uk/~candreop/outbox/genie/fermilab/activity.14/

- **Inspect / understand the code**

- Clean-up your running environment (unset GEVGL, GSPLOAD etc). Read note below

- **Build:** shell% gmake
- **Run:** shell% ./test.exe | grep Main

- **Start tweaking the example;**
  eg get a QELCC cross section instead; plot a differential cross section plot

---

NOTE: For some time-consuming computations, GENIE will (transparently) attempt building splines first so that it can just do spline evaluations later on. For this example you need to prevent that because you only want to print-out a couple of cross-sections anyway...
Type: shell$ **export GDISABLECACHING=1**

Hope you enjoyed your GENIE user experience & learned something new

Please keep on using GENIE & let me know for any question you may have

The End

Appendix:

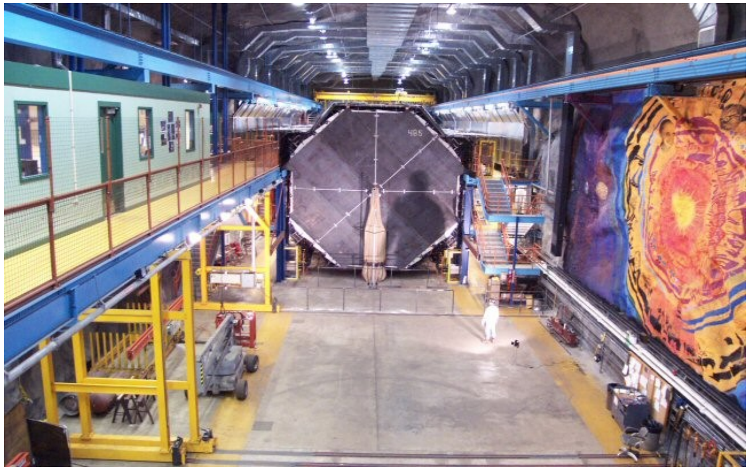
GENIE project info & physics

A bit of history

GENIE evolved from *neugen* / Many models within GENIE have long history.

<u>Neugen originates from the Soudan2 expt.</u>

Soudan2: A proton decay experiment in the ~80's
Back then: vA a background.



<u>Heavily re-developed for MINOS analyses</u>

Cross section model partially re-written / re-tuned.

Hadronic simulations almost completely re-written.
<u>Many year*FTE effort!</u>

NuINT01 / 'Call to arms'

[early ~2000]

- **Entering a precision era in neutrino physics**:
  Neutrino interaction uncertainties start to matter!
- **Also, changes in software devel paradigm**:
  C++ expt. offline softw., Geant4, ROOT

Needed a Universal Neutrino MC

Many (~ 6+) major fortran generators in use.
Developed by small groups / very experiment-specific.
Mostly 'similar' but with no trivial / not understood differences.

For the longer term, the efforts of many will be required to produce a carefully-tested and universal model of neutrino interactions. In addition to purely technical considerations, theoretical guidance and new experimental data will be vital. Still, with the success of NUINT'01 and the promise of renewed and expanded collaboration punctuated and reinforced by future NUINT workshops, it is not too optimistic to hope that within a relatively few years, members of the neu-

Weak Interactions (Springer, Berlin 2000).
10. R. A. Smith and E. J. Moniz, Nucl. Phys. B 43 (1972) 605. [Erratum-ibid. B 101 (1975) 547].
11. K. F. Liu, S. J. Dong, T. Draper and W. Wilcox, Phys. Rev. Lett. 74 (1995) 2172 [arXiv:hep-lat/9406007].
12. L. A. Ahrens et al., Phys. Rev. D 35 (1987) 785.
13. A. Pais, Annals Phys. 63 (1971) 361.

From D.Casper's NuINT01 conference proceedings

What is GENIE?

Generates **E**vents for **N**eutrino **I**nteraction **E**xperiments

A Neutrino Monte Carlo Generator (and extensive toolkit)

Validity:
from few MeV to many hundreds of GeV / handles all nuclear targets

Large scale effort:
110,000 lines of C++

Modularity / Flexibility / Extensibility:
Models can be swapped in/out. Models can be easily reconfigured. All done consistently.

Licensed:
To ensure openness and synergies between experiments

State of the art physics:
GENIE has lots of developers & support. Draws heavily from many people's expertise

Costas Andreopoulos, ***Rutherford Appleton Lab.***

Science & Technology Facilities Council

Who is using GENIE now?

Primary clients are the current / near future medium energy expts:

- T2K
  - nd280
  - SK
  - ingrid
- MINOS
- NovA
- MINERvA
- MicroBooNE
- EU LAr R&D projects
- ...

After ~4 yrs of development (from scratch)
now have a nearly universal neutrino physics MC
(an important tool for physics exploitation for the next decade++)

NEUTRINO EXPT. SYNERGIES !!

GENIE already interfaced to most of these expts & used in physics MC prod.

Could trivially extend GENIE in new kinematical regimes (reactor expts. / neutrino telescopes)
if there is avail. manpower from these communities.

Neutrino Interactions

>>>

Science & Technology Facilities Council

Kinematical variables

$\nu = \frac{q \cdot P}{M} = E - E'$ is the lepton's energy loss in the nucleon rest frame (in earlier literature sometimes $\nu = q \cdot P$). Here, E and E' are the initial and final lepton energies in the nucleon rest frame.

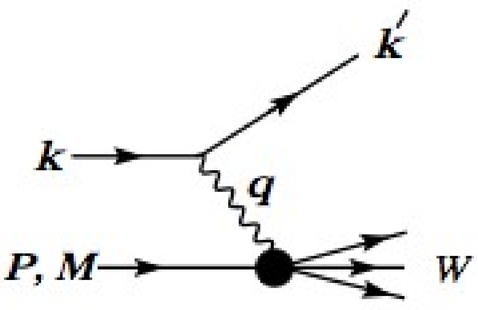
$Q^2 = -q^2 = 2(EE' - \vec{k} \cdot \vec{k}') - m_\ell^2 - m_{\ell'}^2$ where $m_\ell(m_{\ell'})$ is the initial (final) lepton mass. If $EE' \sin^2(\theta/2) \gg m_\ell^2, m_{\ell'}^2$, then

$\approx 4EE' \sin^2(\theta/2)$, where θ is the lepton's scattering angle with respect to the lepton beam direction.

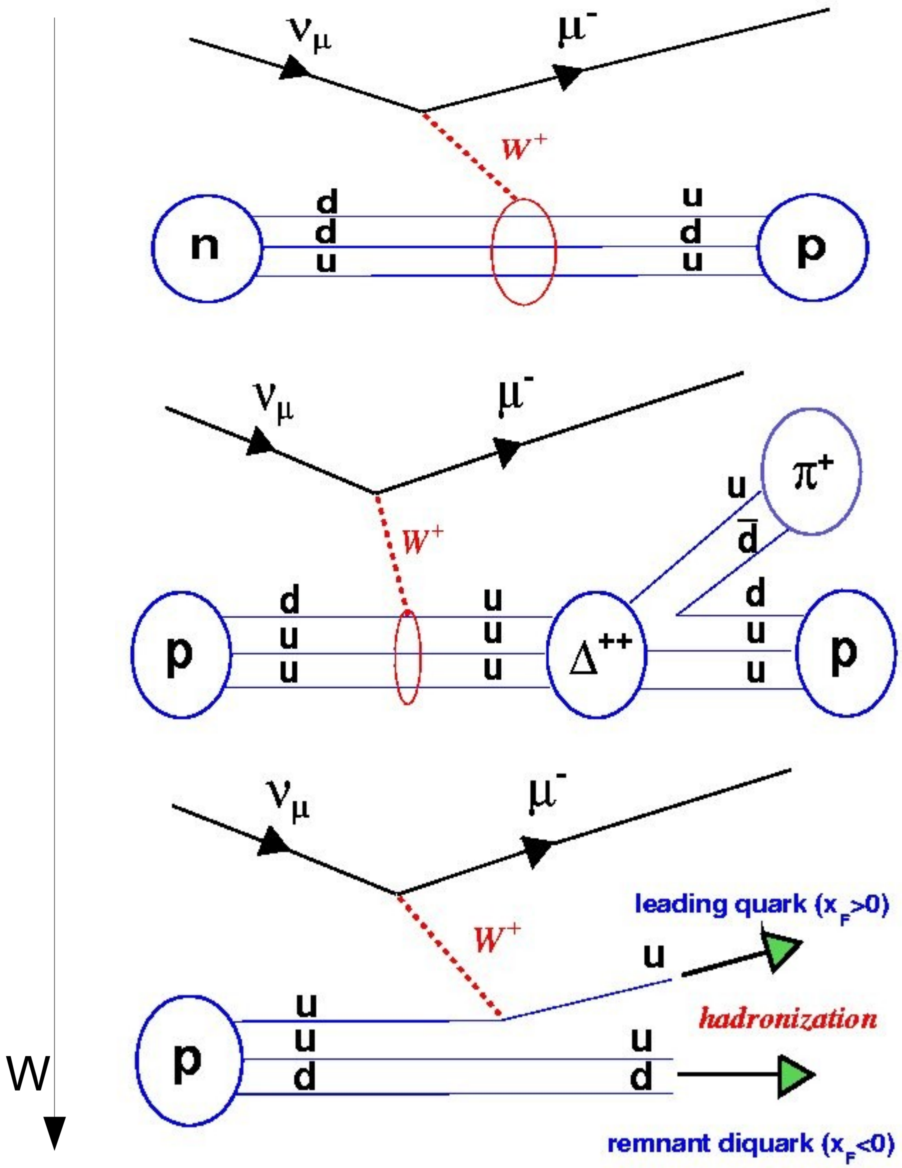
$x = \frac{Q^2}{2M\nu}$ where, in the parton model, x is the fraction of the nucleon's momentum carried by the struck quark.

$y = \frac{q \cdot P}{k \cdot P} = \frac{\nu}{E}$ is the fraction of the lepton's energy lost in the nucleon rest frame.

$W^2 = (P+q)^2 = M^2 + 2M\nu - Q^2$ is the mass squared of the system X recoiling against the scattered lepton.



Dominant interaction modes at ~1 GeV *(nu_mu CC)*



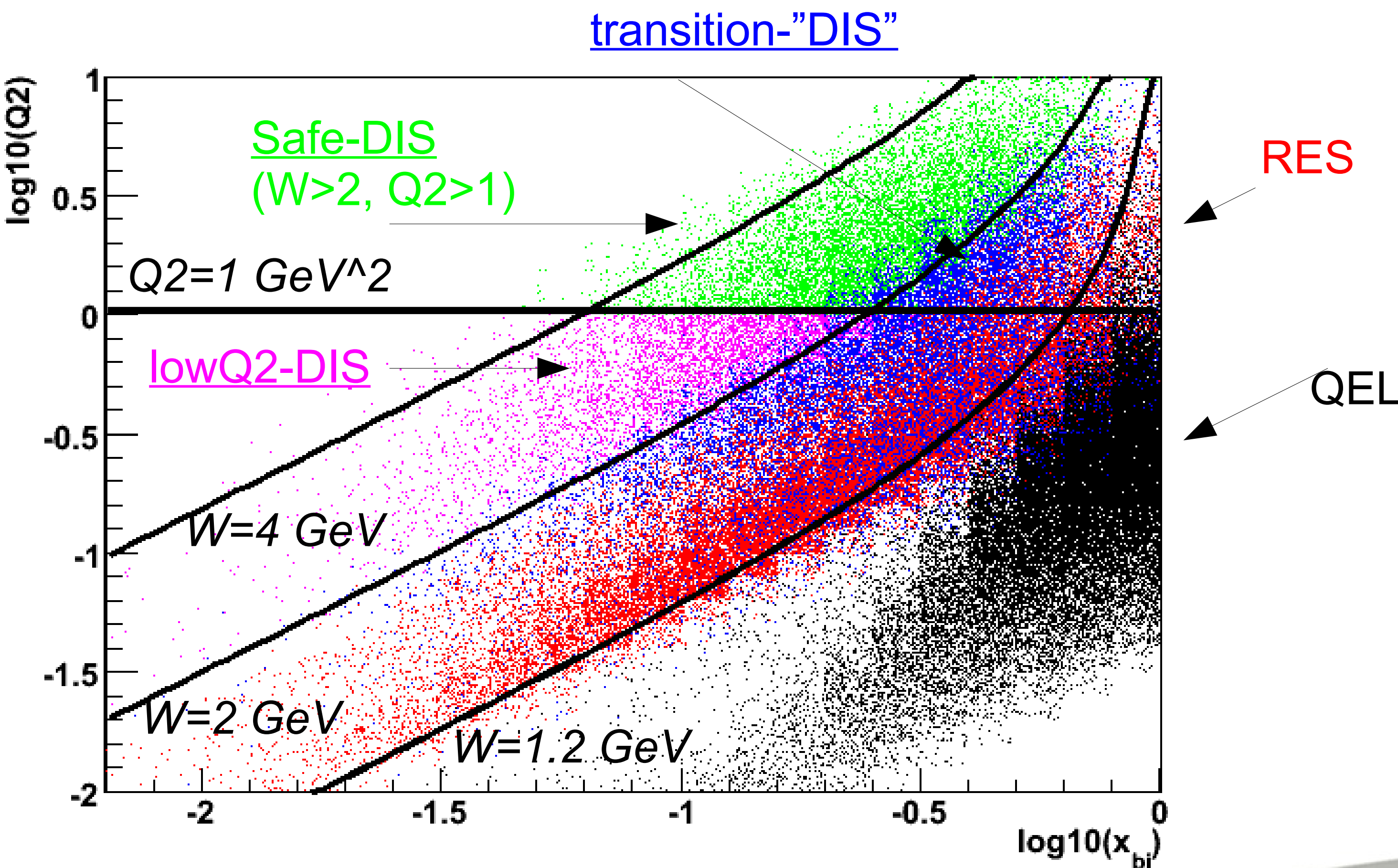
Quasi-Elastic Scattering (QEL)

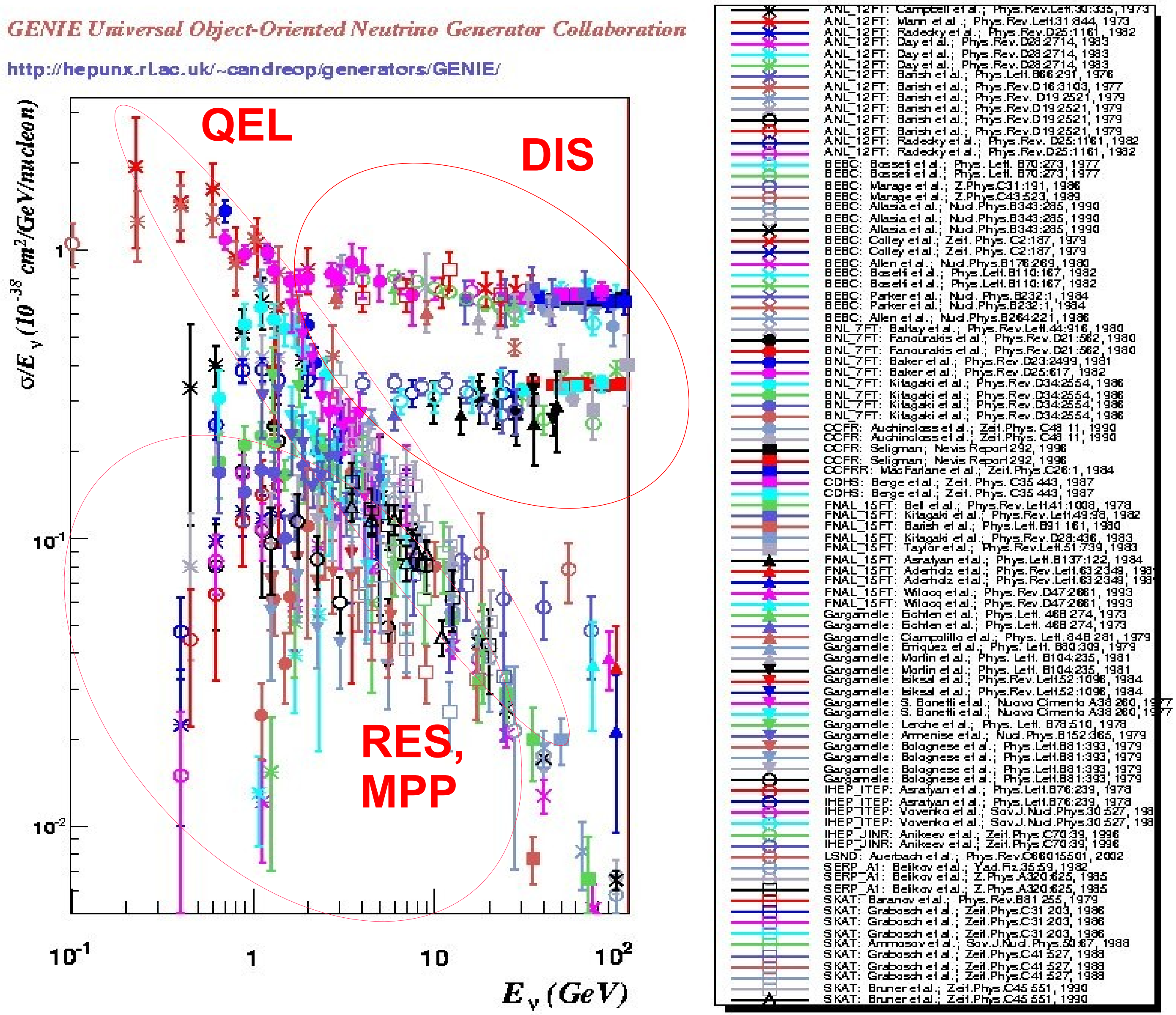
Resonance production (RES)

Deep-Inelastic Scattering (DIS)

Kinematical coverage

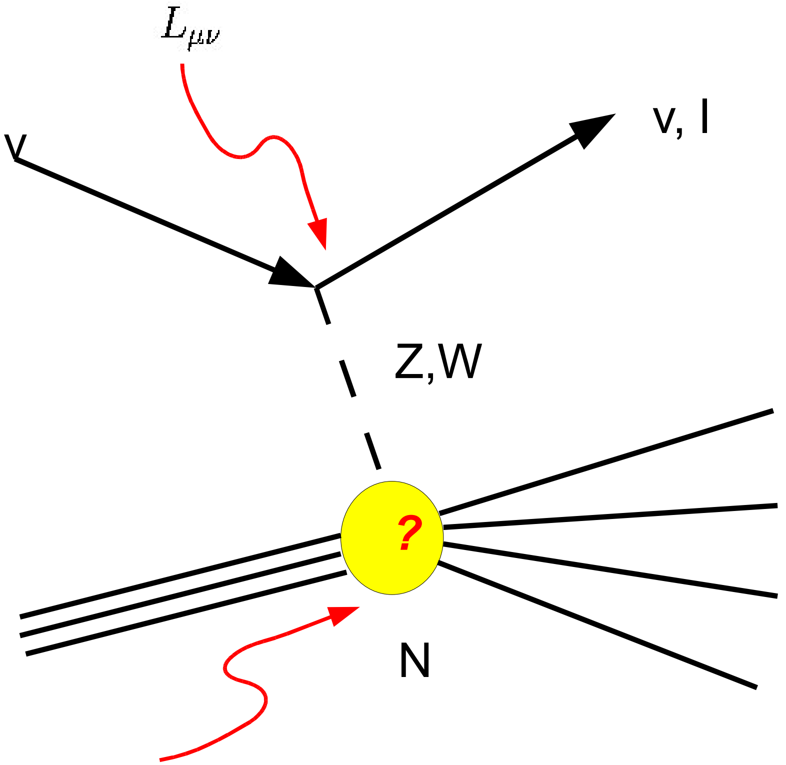
example shown for the JPARC neutrino beam @ nd280 site





Historical World Data

Neutrino scattering off free nucleons



process dynamics described by the invariant amplitude

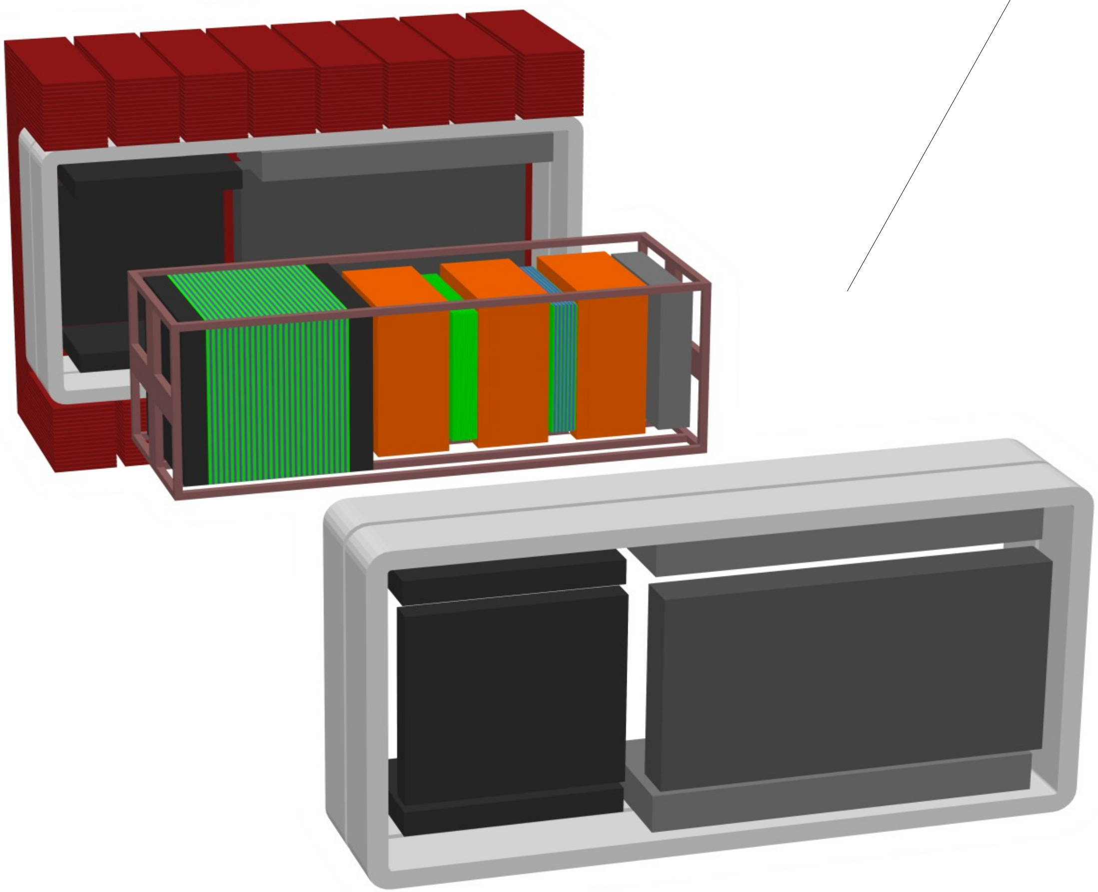
$$|M|^2 = L_{\mu\nu} W^{\mu\nu}$$

$$W_{\mu\nu} = W_1 \delta_{\mu\nu} + W_2 p_\mu p_\nu + W_3 \epsilon_{\mu\nu\alpha\beta} p^\alpha p^\beta + W_4 q_\mu q_\nu + W_5 (p_\mu q_\nu + p_\nu q_\mu) + W_6 (p_\mu q_\nu - p_\nu q_\mu)$$

Complicated!!

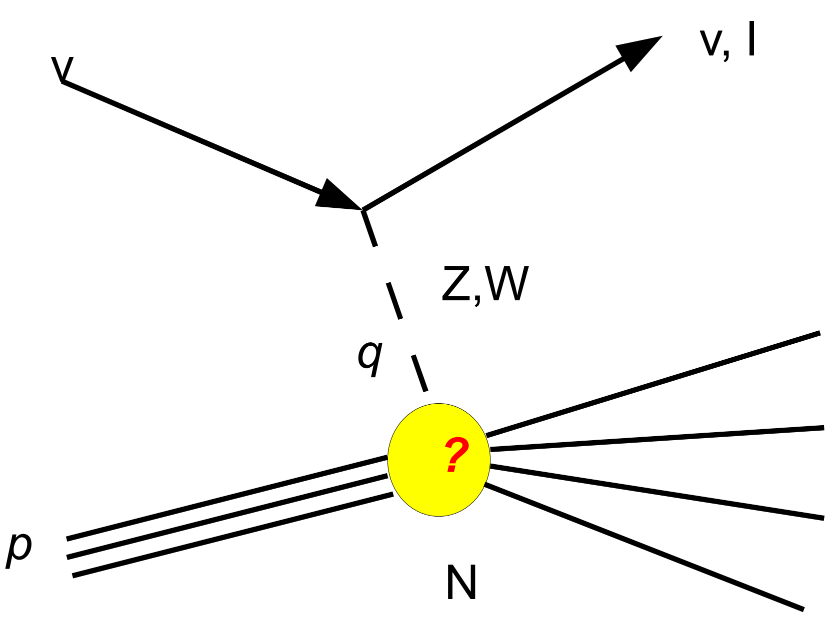
We usually have to deal with nuclear targets

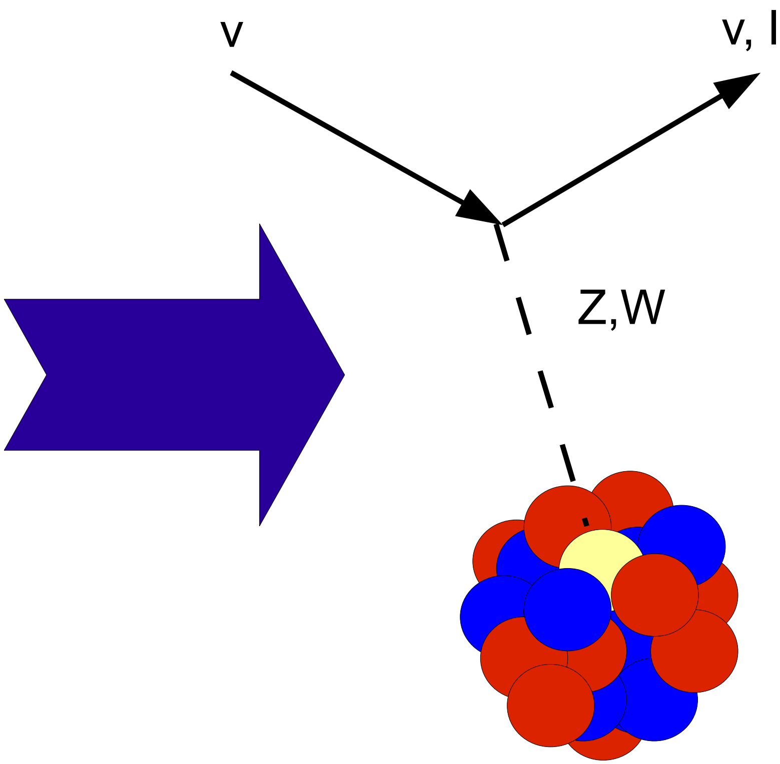
Example: nuclear targets in the nd280 geometry



H2,
B10, B11,
C12, C13,
N14, N15,
O16, O17, O18,
Na23,
Al27,
Si28, Si29, Si30,
Cl35, Cl37,
Ar36, Ar38, Ar40,
Fe54, Fe56, Fe57, Fe58
Cu63, Cu65,
Zn64, Zn66, Zn67, Zn68, Zn70,
Co69,
Pb204, Pb206, Pb207, Pb208

Free-nucleon cross section → Nuclear cross sections

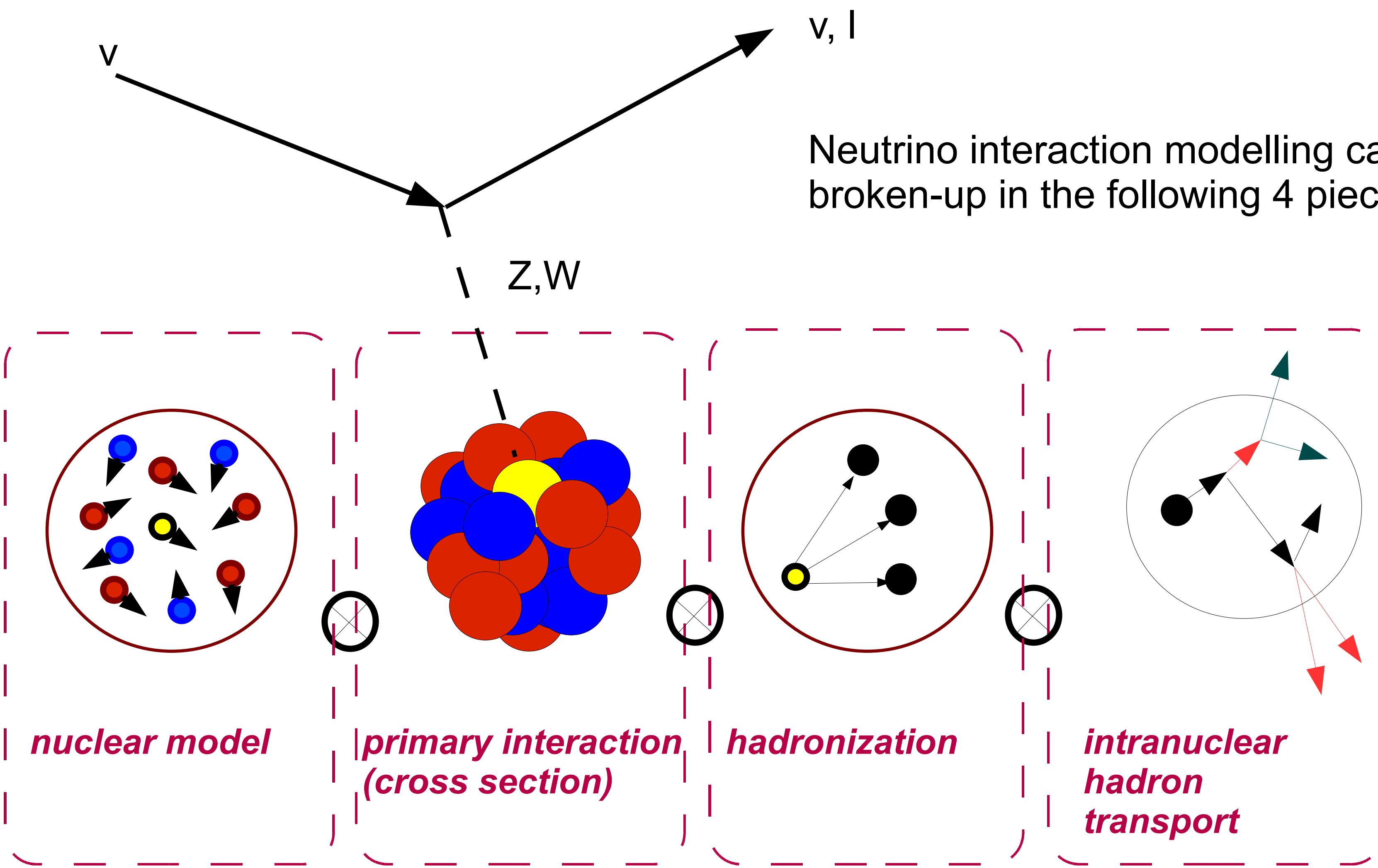




Welcome to the `nuclear physics hell' !

Neutrino Interaction Simulation `steps'

Neutrino interaction modelling can be broken-up in the following 4 pieces :



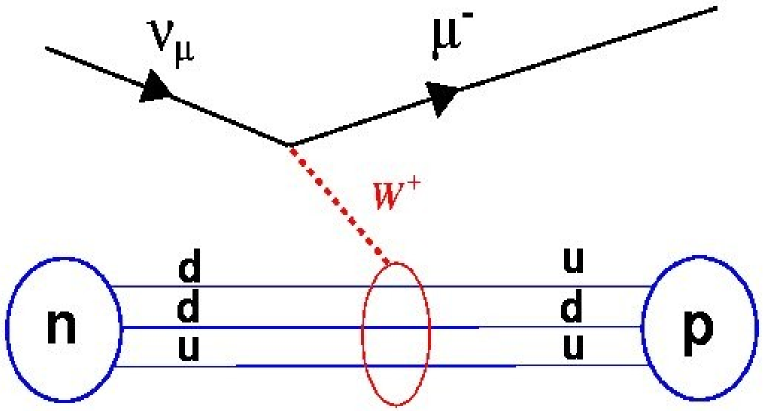
Note: A simplified picture

Calculating Neutrino Interaction Cross Sections within GENIE
>>>

Costas Andreopoulos, *Rutherford Appleton Lab.*

QEL scattering

- Critical for current accelerator LBL oscillation experiments
- *> ~50% of total CC cross section at ~1 GeV*



Full kinematical reconstruction just by looking at the leptonic system:

$$E_\nu = \frac{m_N E_\mu - m_\mu^2/2}{m_N - E_\mu + p_\mu cos\theta_\mu} \qquad Q^2 = -2E_\nu(E_\mu - p_\mu cos\theta_\mu) + m_\mu^2$$

QEL cross section

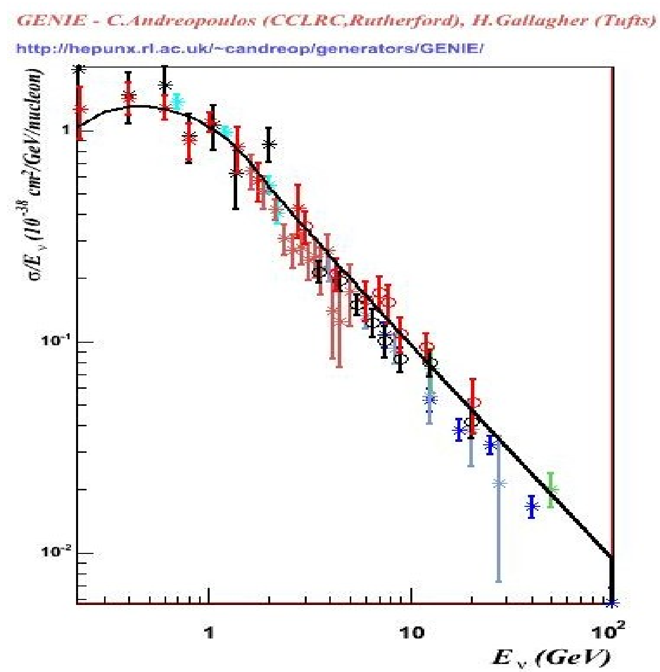
$$\frac{d\sigma^{\mathrm{QES}}}{dQ^2} = \frac{G_F^2 \cos^2\theta_C M^2 \kappa^2}{2\pi E_\nu^2}\left[A\left(q^2\right) + \left(\frac{s-u}{4M^2}\right) B\left(q^2\right) + \left(\frac{s-u}{4M^2}\right)^2 C\left(q^2\right)\right]$$

A,B,C = $f(F_A, F_{v1}, F_{v2})$

vector form factors:
determined from e-N via CVC

dipole axial form factor:

$$F_A = g_A \left(1 + \frac{Q^2}{M_A^2}\right)^{-2}$$



Elastic nucleon form factors

vN QEL xsec expressed in terms of vector & axial form factors

$$F_V^1(Q^2) = \frac{G_E^V(Q^2) - \tau G_M^V(Q^2)}{1-\tau}$$

$$\xi F_V^2(Q^2) = \frac{G_M^V(Q^2) - G_E^V(Q^2)}{1-\tau}$$

CVC allows us to determine Gve, Gvm from the elastic form factors

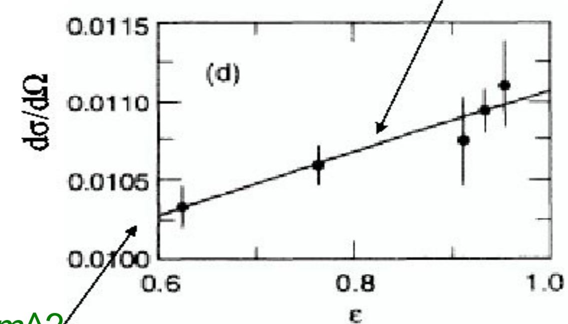
$$G_E^V(Q^2) = G_{ep}(Q^2) - G_{en}(Q^2),$$
$$G_M^V(Q^2) = G_{mp}(Q^2) - G_{mn}(Q^2)$$

Elastic form factor measurements:

- ***Rosenbluth separation:***

$$\frac{d\sigma}{d\Omega} = \frac{\alpha^2 E_e' \cos^2\frac{\theta_e}{2}}{4E_e^3 \sin^4\frac{\theta_e}{2}} \left[G_e^2 + \frac{\tau}{\varepsilon} G_m^2\right]\left(\frac{1}{1+\tau}\right)$$

slope measures Ge^2



*offset measures t * Gm^2*

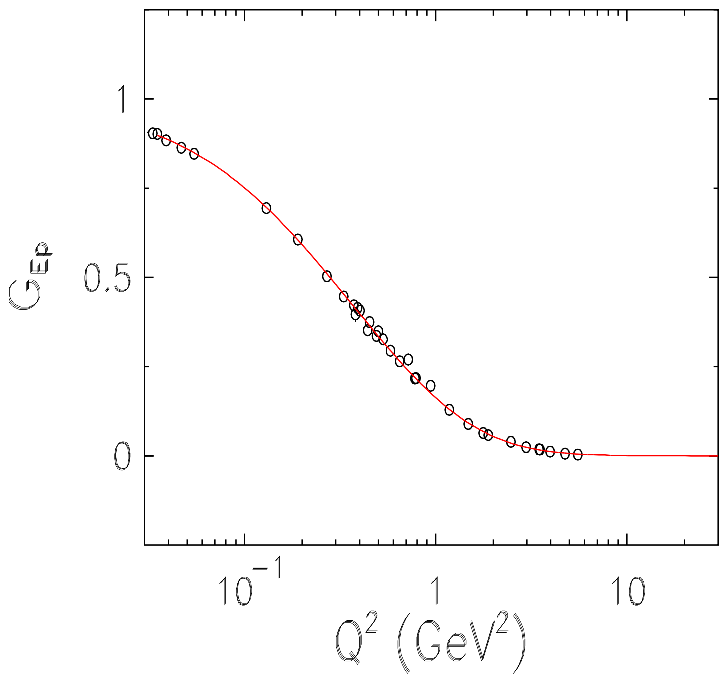
- ***Polarization measurements:***

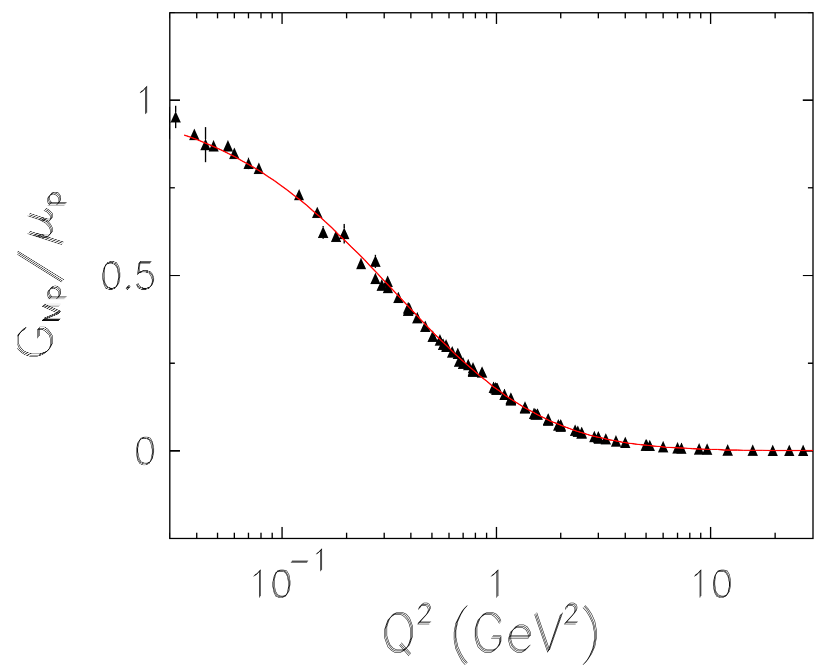
$$\frac{G_e}{G_m} = -\frac{P_t}{P_l}\frac{(E_e + E_{e'})}{2M}\tan\left(\frac{\theta_e}{2}\right)$$

- The 2 methods do not agree
- Polarization measurements seen as more reliable

Beyond the dipole form factors

BBA fit based mostly on polarisation data (eg Budd / Bodek / Arrington. See hep-ex/0308005)



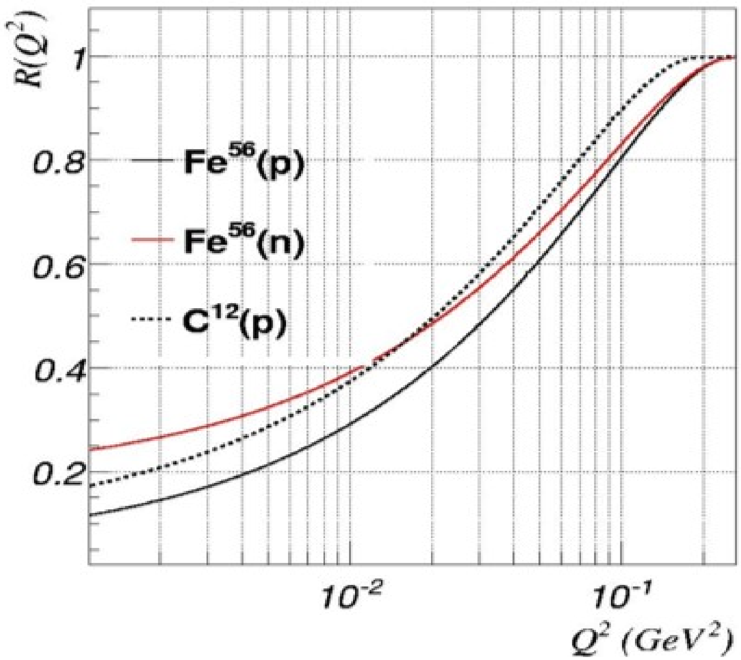


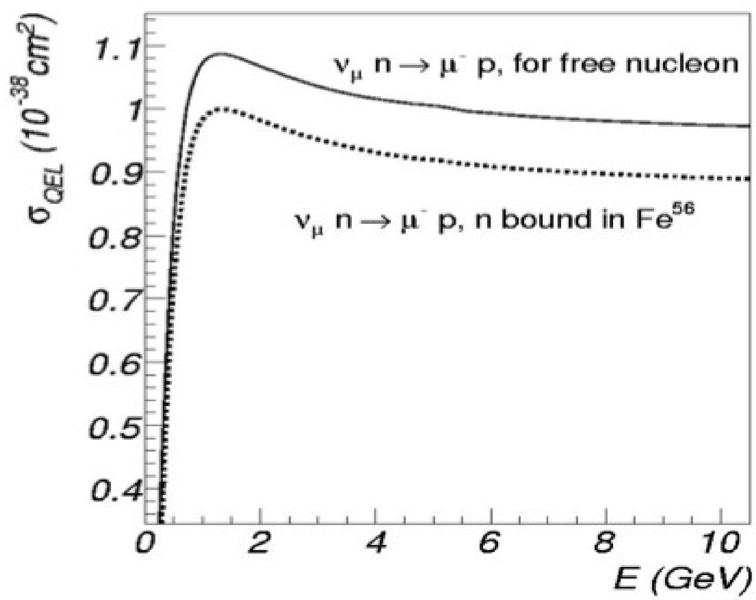
[R.Bradford et al, NuINT05]

QEL cross section for nuclear targets

Off-shell kinematics

A suppression factor R(Q^2), derived from an analytical calculation of the Pauli blocking effect, is included.

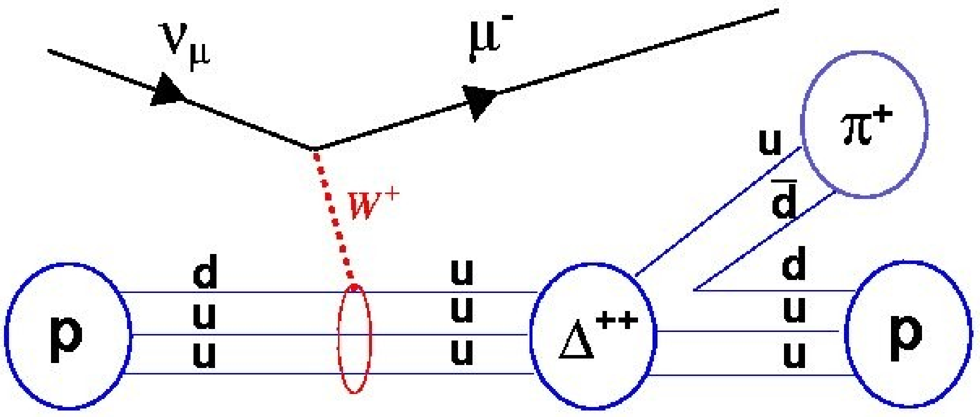




Neutrino-production of resonances

$$\nu + N \rightarrow l + Resonance$$

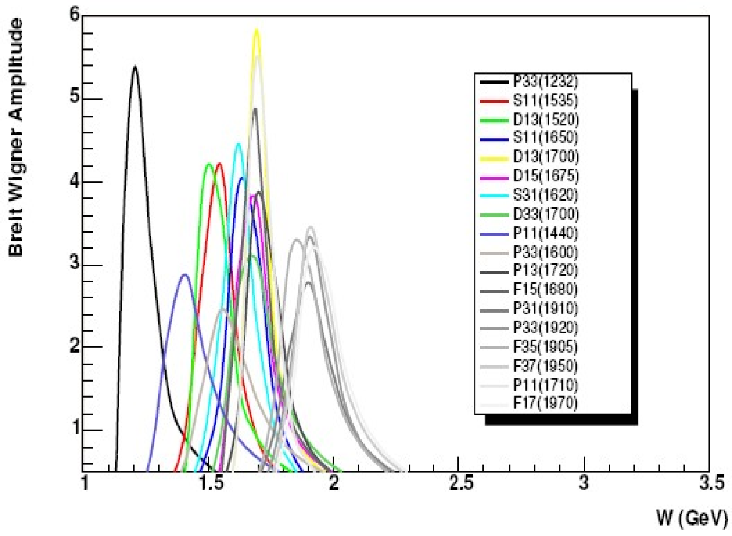
$\hookrightarrow$ *N + pion*
(usually)



Very important channel(s)
~30% of the total CC xsec around ~ 1 GeV

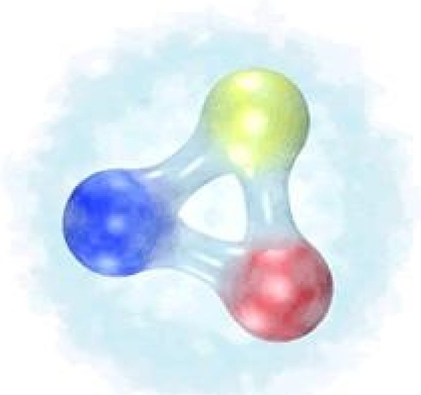
Very complicated!

Resonance Mass (MeV)	L(2I,2J)	PDG status	Breit – Wigner Width (MeV)	FKR n	BR $N^* \to N\pi$
$\Delta(1232)$	P_{33}	****	120	0	100%
N(1535)	S_{11}	****	150	2	45%
N(1520)	D_{13}	****	120	1	55%
N(1650)	S_{11}	****	150	1	73%
N(1700)	D_{13}	***	100	1	10%
N(1675)	D_{15}	****	150	1	45%
$\Delta(1700)$	D_{33}	****	300	1	15%
N(1440)	P_{11}	****	350	1	65%
$\Delta(1600)$	P_{33}	***	350	2	18%
N(1720)	P_{13}	****	150	2	15%
N(1680)	P_{15}	****	130	1	65%
$\Delta(1910)$	P_{31}	****	250	2	23%
$\Delta(1920)$	P_{33}	***	200	2	13%
$\Delta(1905)$	F_{35}	****	350	2	10%
$\Delta(1950)$	F37	****	300	2	10%
N(1710)	P_{11}	***	100	2	38%



Neutrino-production of resonances

The most widely used model for resonance neutrinoproduction
(*D.Rein, L.M Sehgal*, **Ann.Phys.133, 79 (1981)**)
uses the FKR dynamical model
(*R.P.Feynman, M.Kislinger, F.Ravndall*, **Phys.Rev.D 3, 2706 (1971)**)
to describe excited states of a 3 quark bound system.



kinematical factors

$$\frac{d^2\sigma}{dW\,dq^2} \propto u^2\sigma_L(q^2,W) + v^2\sigma_R(q^2,W) + 2uv\sigma_S(q^2,W)$$

Helicity Cross Sections *(L,R,S)*
They depend on the details of the FKR model
(and “maybe” a snapshot of the PDG resonance tables as they were in early '70's ?)

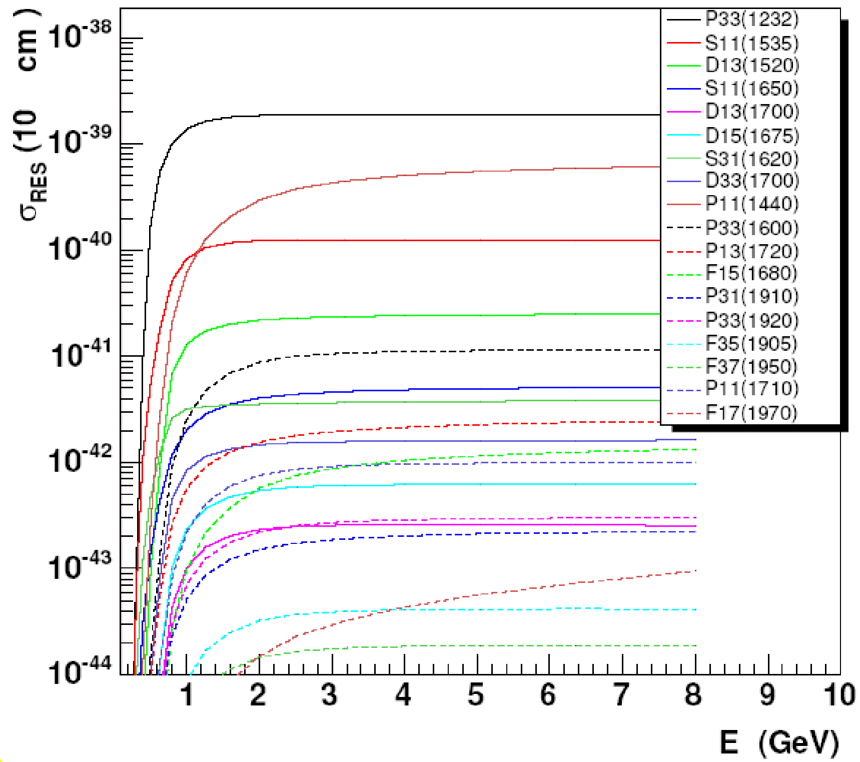
Axial & Vector transition form factors:
assuming dipole form Q^2 dependence

$$G^{V,A}(Q^2) = \left(1+\frac{Q^2}{4M^2}\right)^{1/2-n}\left(1+\frac{Q^2}{M^2_{V,A}}\right)^{-2}$$

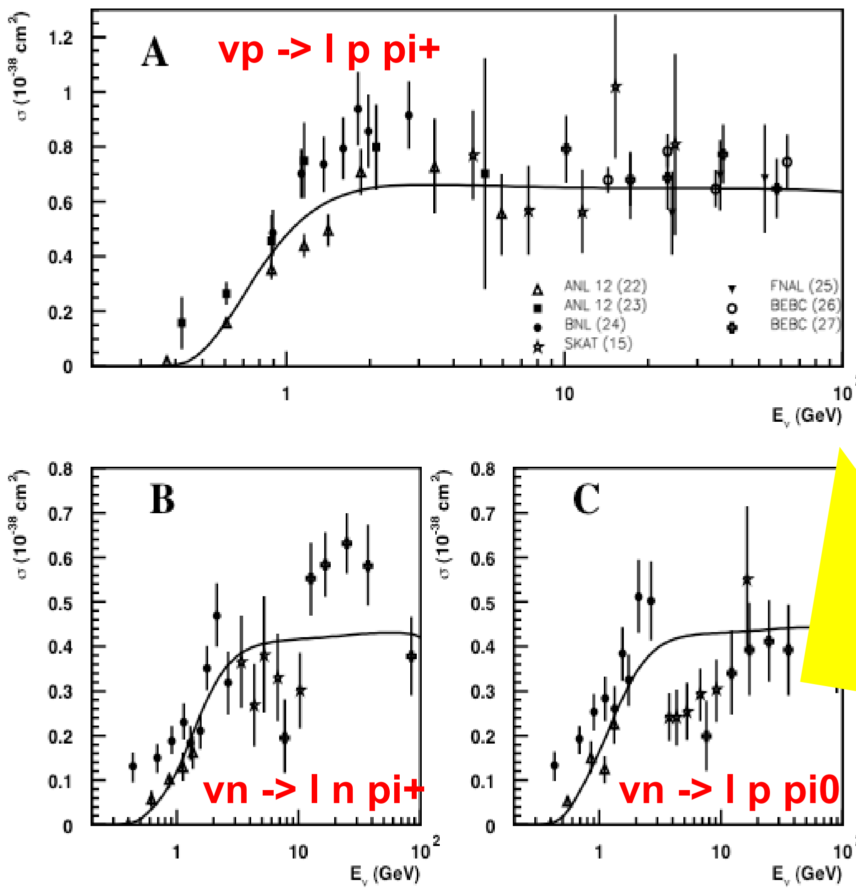
M_V=0.84 GeV/c^2, M_A

Neutrino-production of resonances

Resonance excitation cross sections
(as a function of energy / for muon neutrinos)



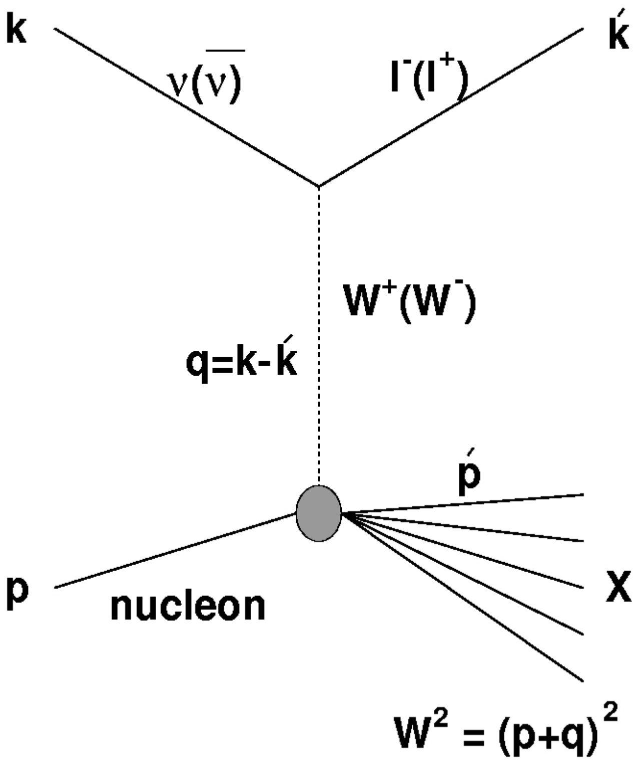
Single pion production cross sections



Include isospin amplitudes and 1pi BR to weight the contribution of each resonance to exclusive single pion reactions

Can add coherently
For simplicity, many calculations add incoherently

Deep Inelastic Scattering



Differential cross section in terms of 5 structure functions:

$$\frac{d^2\sigma^{\nu(\bar{\nu})}}{dxdy} = \frac{G_F^2 M_N E}{\pi(1+Q^2/M_W^2)^2} \sum_{i=1}^{5} A_i\,(x,y,E)\,F_i\,(x,Q^2)$$

where:

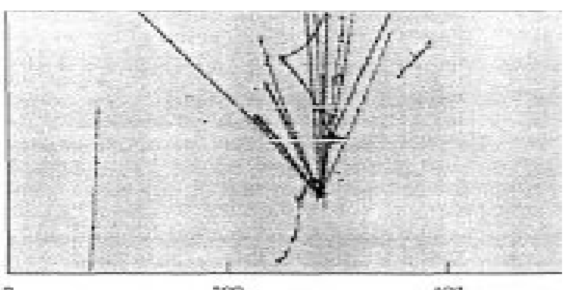
$$A_1 = y\left(xy + \frac{m_\mu^2}{2M_N E}\right),$$

$$A_2 = 1 - \left(1 + \frac{M_N x}{2E}\right) y - \frac{m_\mu^2}{4E^2},$$

$$A_3 = \pm y\left[x\left(1 - \frac{y}{2}\right) - \frac{m_\mu^2}{4M_N E}\right],$$

$$A_4 = \frac{m_\mu^2}{2M_N E}\left(y + \frac{m_\mu^2}{2M_N E x}\right),$$

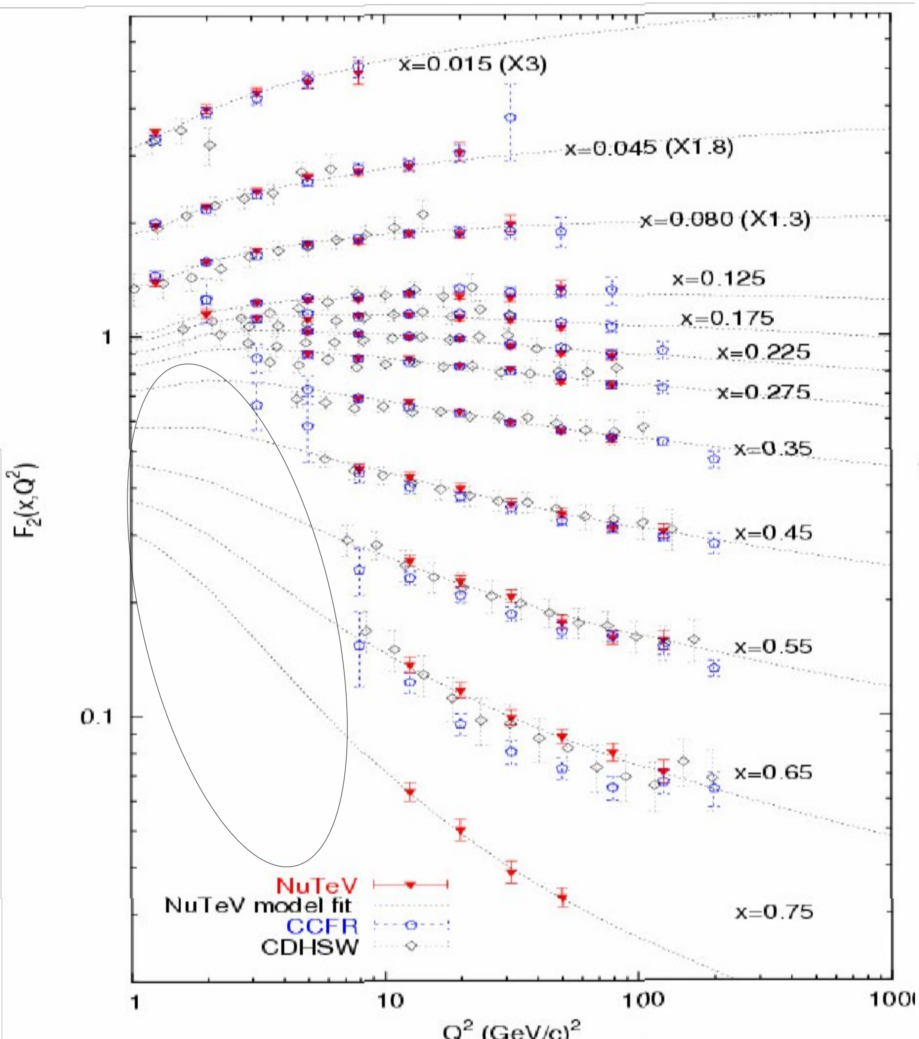
$$A_5 = -\frac{m_\mu^2}{M_N E}.$$



LAr images, courtesy A.Curioni

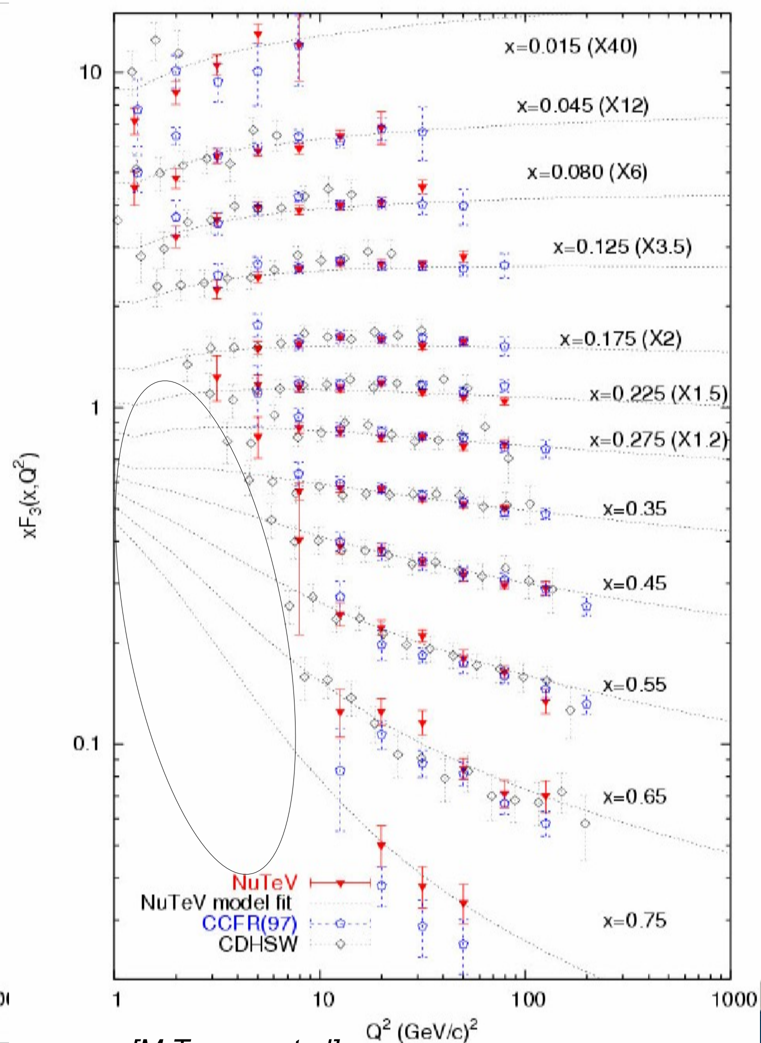
Deep Inelastic Scattering / Structure functions

F2



[M.Tzanov et al]

xF3



[M.Tzanov et al]

Bodek / Yang model

Based on LO cross section models with new scaling variable to account for higher twists and modified PDFs to describe low-Q2 data

$$\xi_w = \frac{2x(Q^2 + {M_f}^2 + B)}{Q^2[1+\sqrt{1+(2Mx)^2/Q^2}] + 2Ax}$$

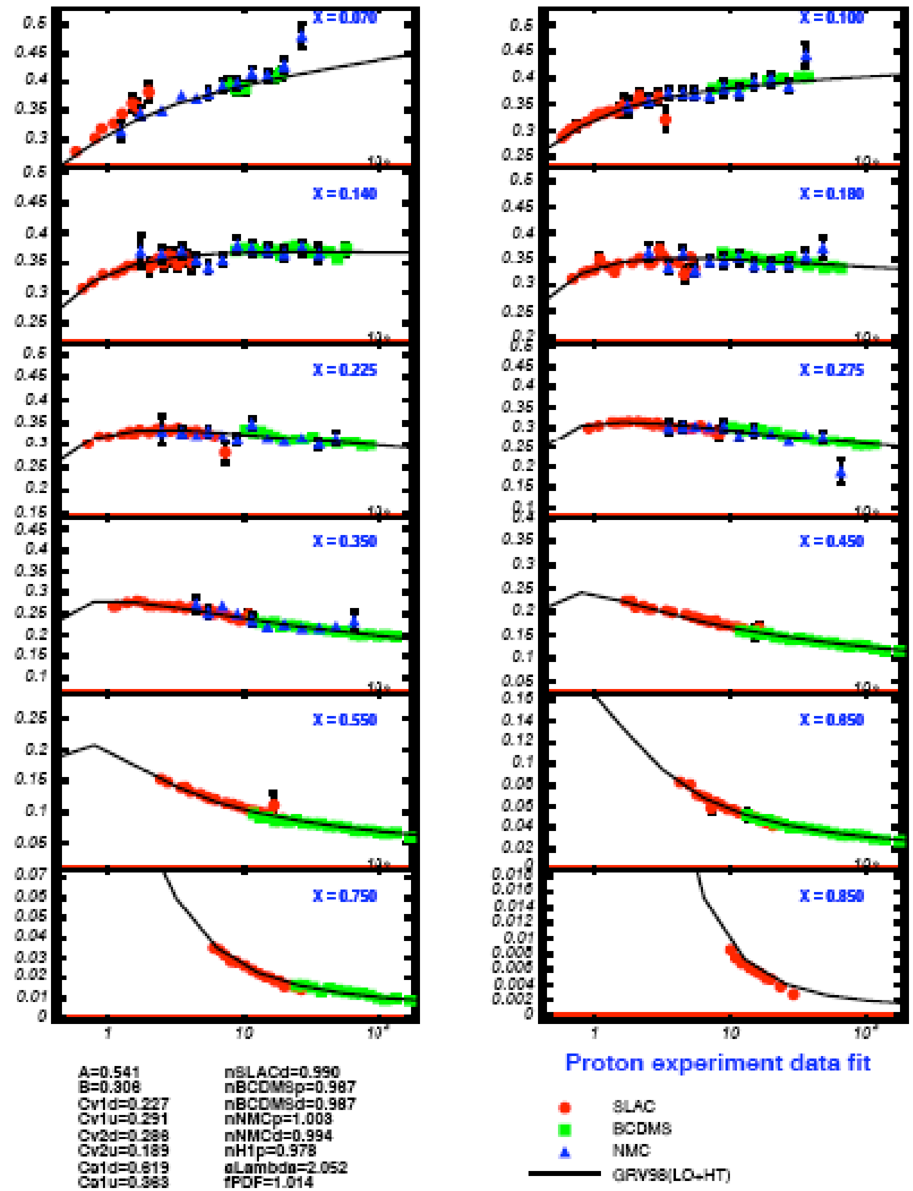
$$K_{sea}(Q^2) = \frac{Q^2}{Q^2 + C_s}$$

$$K_{valence}(Q^2) = [1 - G_D^2(Q^2)] \times \left(\frac{Q^2 + C_{v2}}{Q^2 + C_{v1}}\right)$$

Fits based on GRV98LO and free nucleon charged lepton data

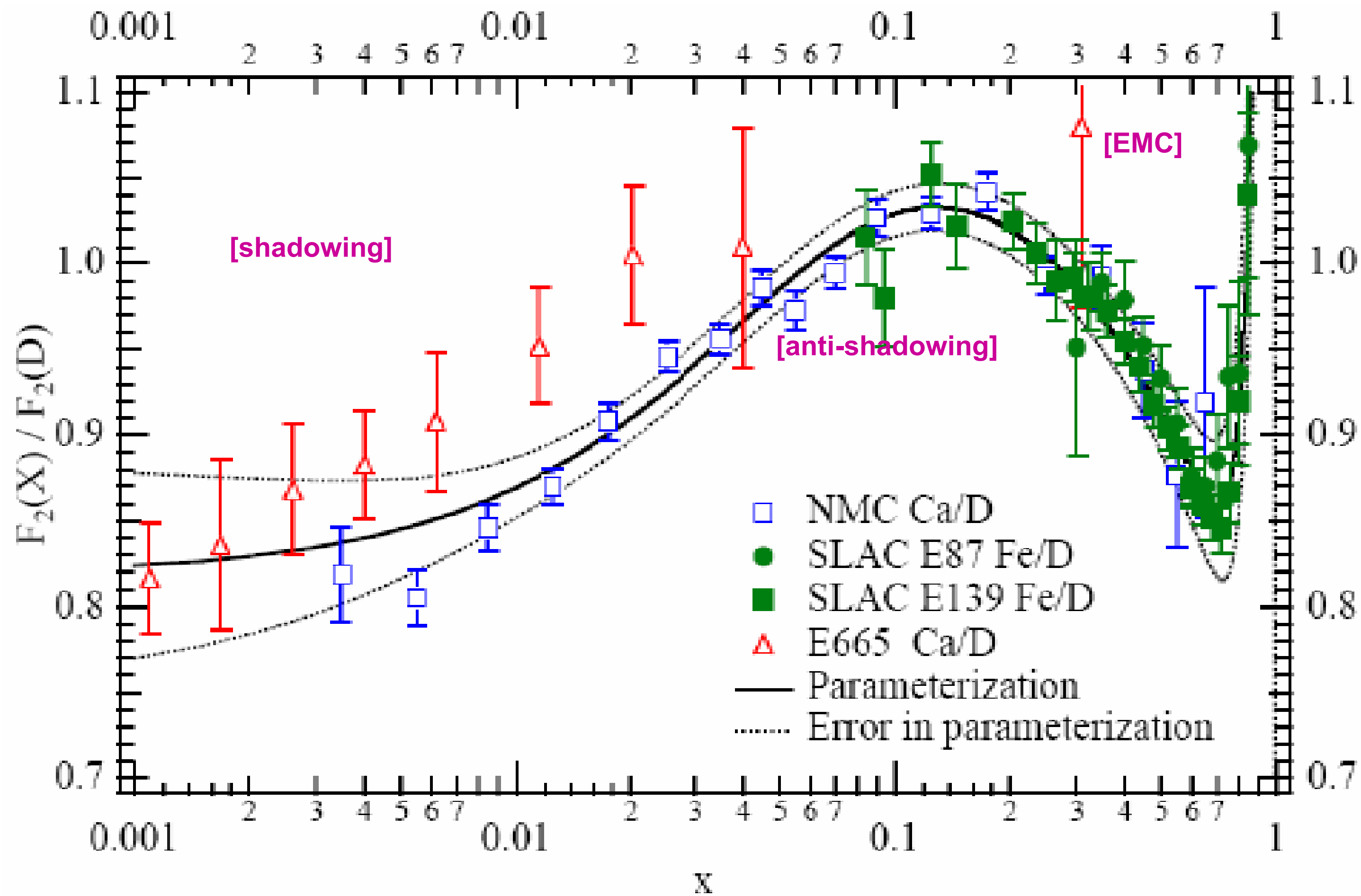
[hep-ph/0411202]

[Bodek & Yang]



Costas Andreopoulos, *Rutherford Appleton Lab.*

Deep Inelastic Scattering / Nuclear corrections



Costas Andreopoulos, *Rutherford Appleton Lab.*

Putting everything together

total inclusive cross section
= sum of contributions from:

- *exclusive channels*
- *DIS*

$$\sigma_{\nu N}^{\text{tot}} = \sigma_{\nu N}^{\text{(Q)ES}} \oplus \boxed{\sigma_{\nu N}^{1\pi} \oplus \sigma_{\nu N}^{2\pi} \oplus \ldots \oplus \sigma_{\nu N}^{1K} \oplus \ldots} \oplus \sigma_{\nu N}^{\text{DIS}}.$$

$$\sigma_{\nu N}^{\text{tot}} = \sigma_{\nu N}^{\text{(Q)ES}} \oplus \sigma_{\nu N}^{\text{RES}} \oplus \sigma_{\nu N}^{\text{DIS}}$$

Putting everything together

$$\frac{d\sigma}{d\theta dE'} = \frac{d\sigma}{d\theta dE'}^{RES} + \frac{d\sigma}{d\theta dE'}^{DIS}$$

where

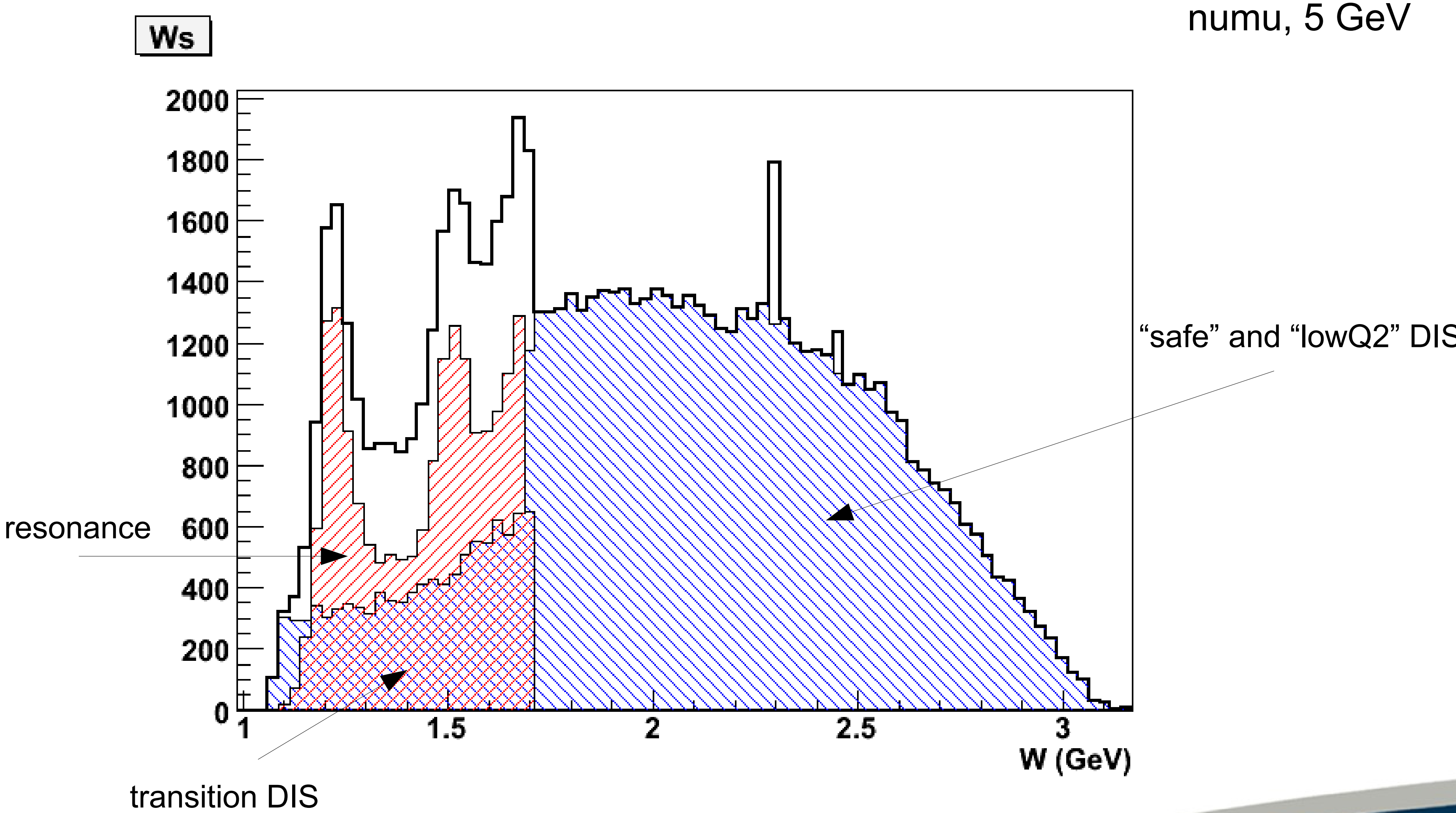
$$\frac{d\sigma}{d\theta dE'}^{RES} = \sum_{i=1}^{17} \frac{d\sigma}{d\theta dE'}_i^{RS} \Theta(Wcut - W)$$

where the summation is over the 17 resonances in the Rein-Seghal model.
and

$$\frac{d\sigma}{d\theta dE'}^{DIS} = \frac{d\sigma}{d\theta dE'}^{DIS-BY} \Theta(W - Wcut) + \frac{d\sigma}{d\theta dE'}^{DIS-BY} \Theta(Wcut - W) \sum_k f_k$$

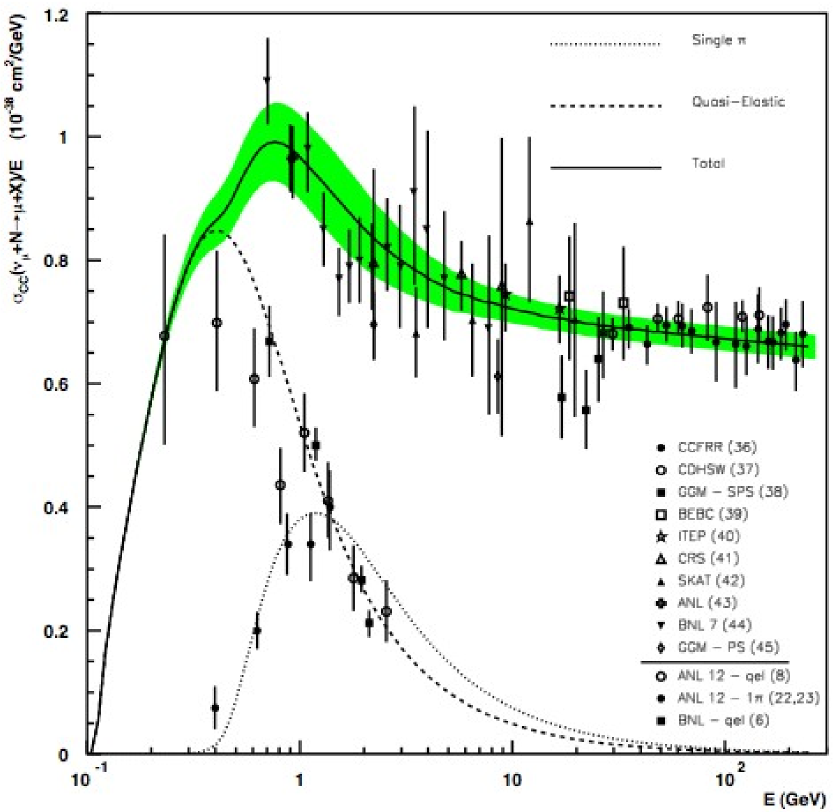
Putting everything together

numu, 5 GeV

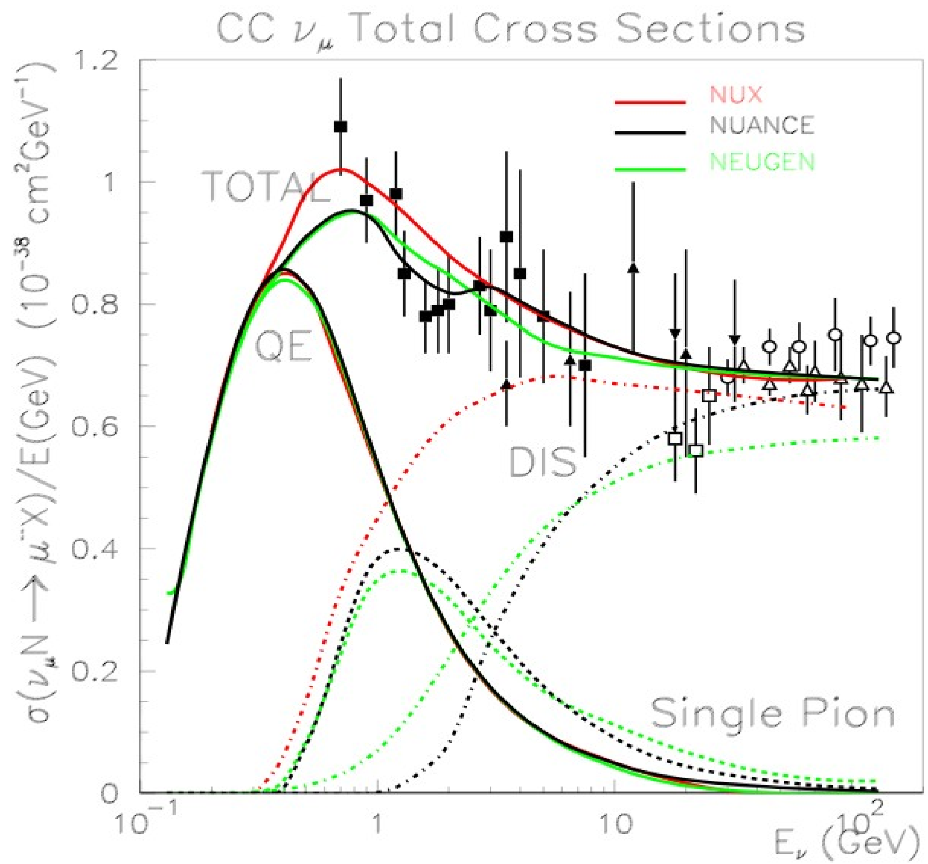


The GENIE cross section model

2.5.1 free nucleon cross section prediction vs B/C data & <u>estimated uncertainty</u>

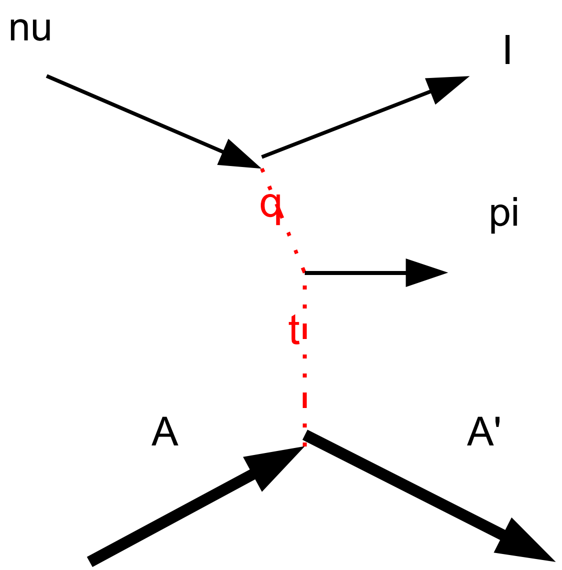


Sam Zeller. circa-2002 / Cross-generator comparisons



- (Incl. & low multiplicity excl.) free-nucleon cross section differences between generators not significant
- Within uncertainty band
- **Understanding the uncertainty on a prediction is <u>more important</u> than any given prediction.**

Coherent pion production



Cross section computed as in
Rein, Sehgal, hep-ph/0606185

Including the PCAC formula with the non-vanishing muon mass causing destructive interference between AV and PS amplitudes.

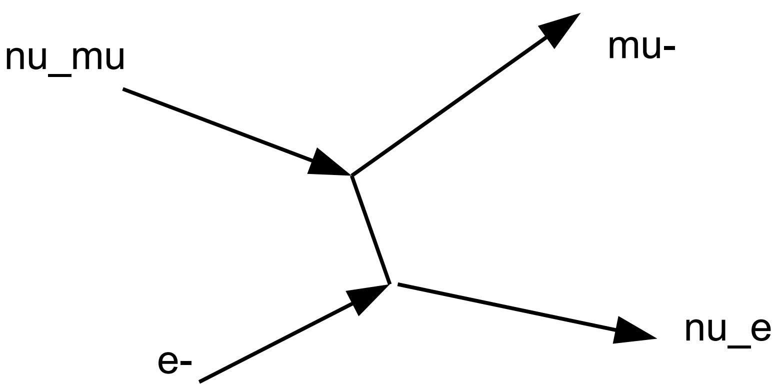
Coherent elastic

In progress – not implemented yet

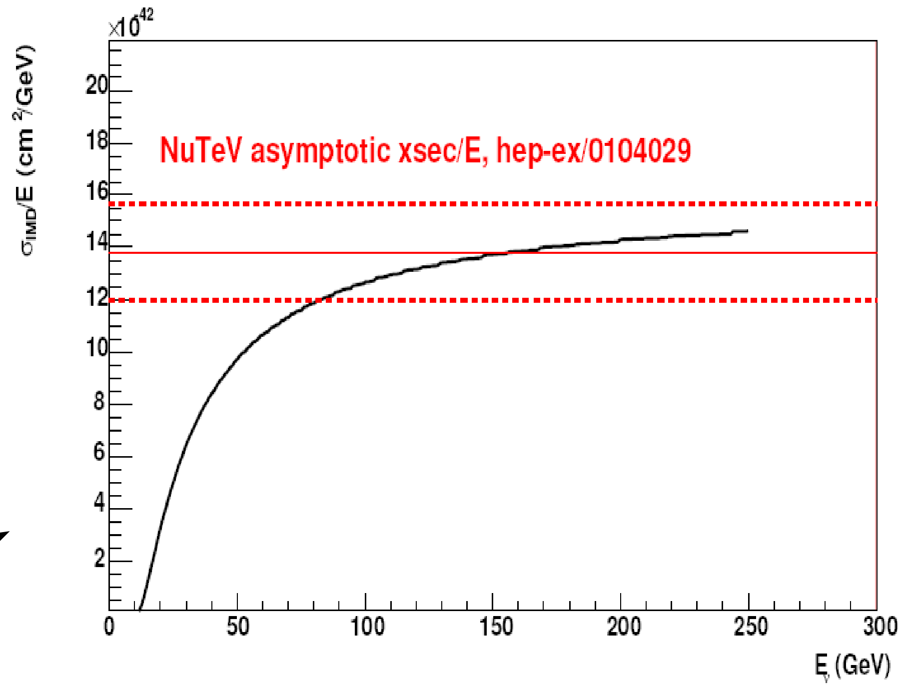
ve- elastic

Fairly standard.
Cross sections implemented as in W.J.Marciano and Z.Parsa, J.Phys.G: Nucl.Part. Phys.29 (2003) 2629.
Radiative corrections currently neglected.

Inverse muon decay



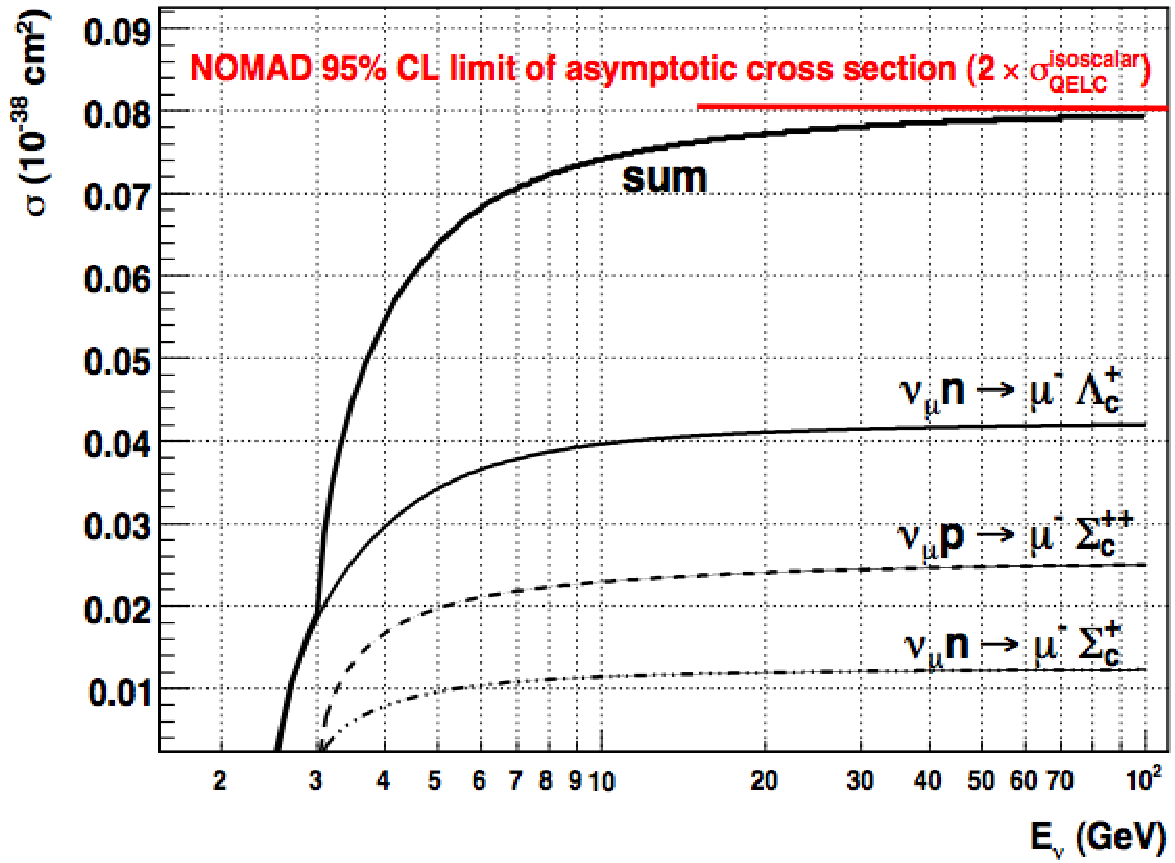
D.Yu.Bardin and V.A.Dokuchaeva, Nucl.Phys.B287:839 (1987), includes all 1-loop radiative corrections



QEL charm production

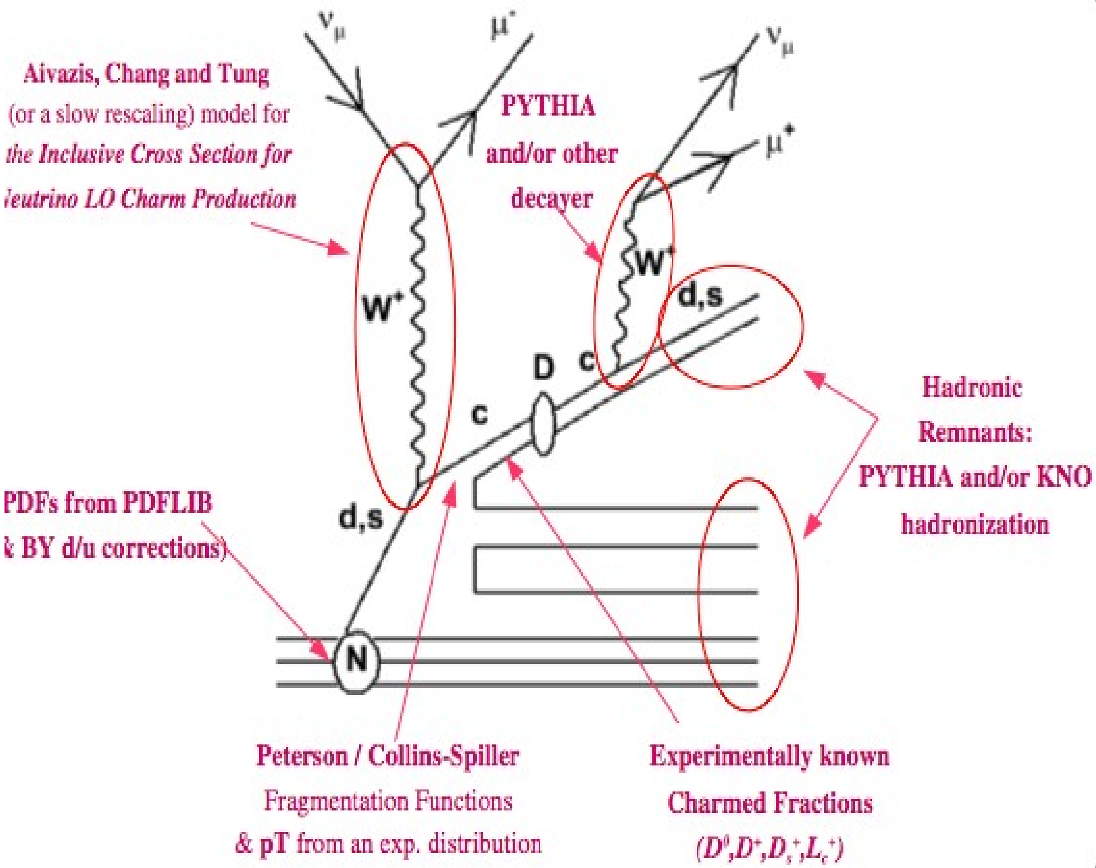
$$\nu + n \rightarrow l^- + \Lambda_c^+(2285)$$
$$\nu + n \rightarrow l^- + \Sigma_c^+(2455)$$
$$\nu + p \rightarrow l^- + \Sigma_c^{++}(2455)$$

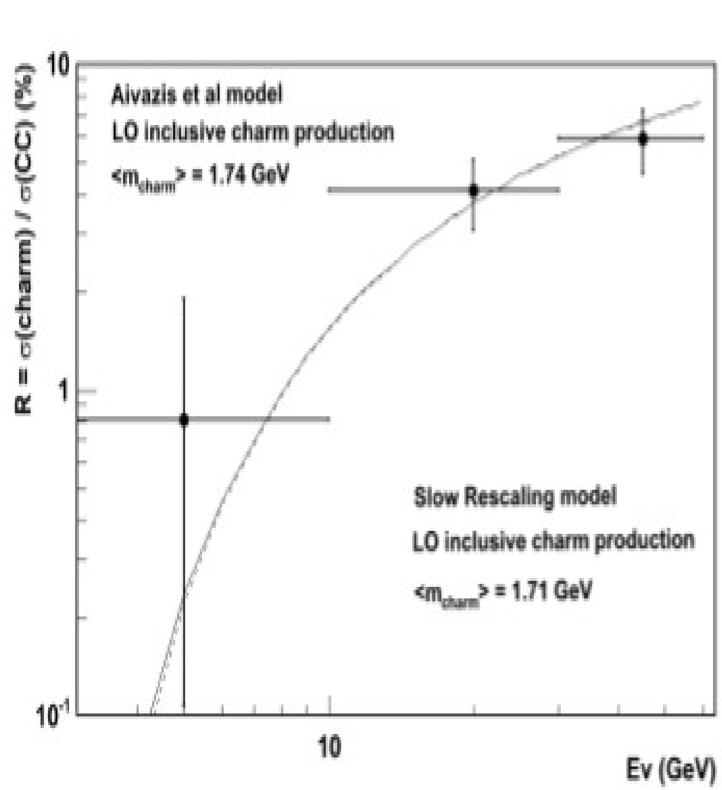
S.G.Kovalenko, Sov.J.Nucl.Phys.52:934 (1990)
rescaled to NOMAD limit



Science & Technology Facilities Council

DIS charm production





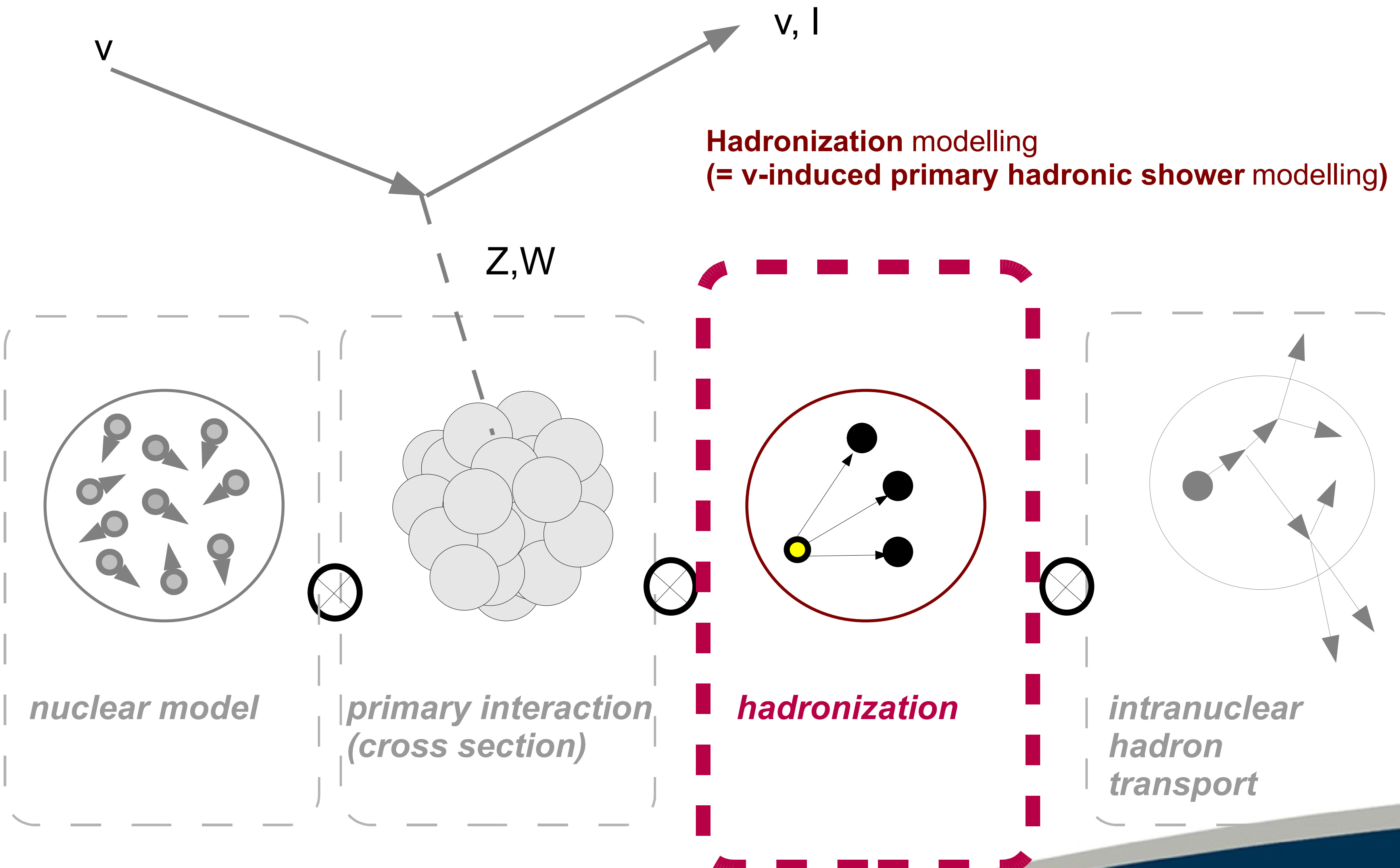
Costas Andreopoulos, *Rutherford Appleton Lab.*

Hadronic simulations within GENIE
>>>

Costas Andreopoulos, ***Rutherford Appleton Lab.***

Science & Technology Facilities Council

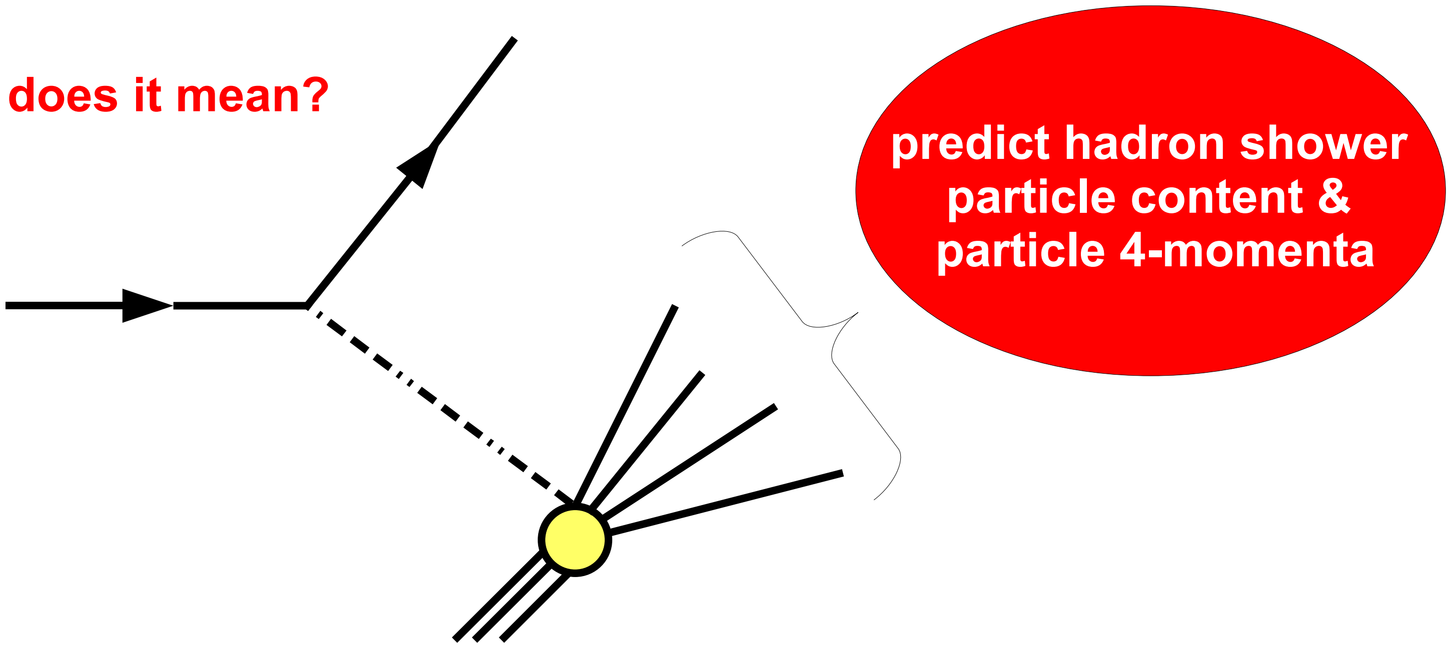
Hadronization modelling



Costas Andreopoulos, *Rutherford Appleton Lab.*

Hadronization modelling

what does it mean?



- Standard tools of the trade *(PYTHIA/JETSET, HERWIG)* don't work at the low hadronic invarant masses which are of interest to us

- Important to get that right
  - Determines shower shapes & particle content
    - *Eg, electromagnetic / pi0 fraction of the shower -> nue backgrounds*
    - *Eg, CC/NC shower shapes -> CC/NC PIDs*
  - Used to decompose inclusive vN->IX to exclusive contributions
    - *Eg, Contribution of 1 pi DIS channels in RES/DIS transition region*

The GENIE hadronization model

Andreopoulos-Gallagher-Keyahias-Yang (AGKY) model

Effective low-mass hadronization model (custom model) (W<W1)

Linear Transition between low/high mass models (W1>W>W2)

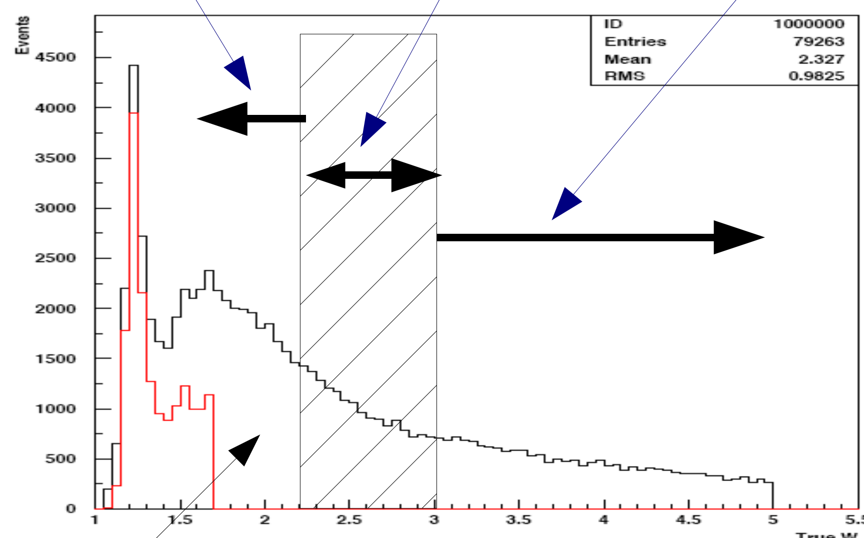
Dynamic high-mass hadronization model (PYTHIA/JETSET) (W>W2)

At low hadronic invariant masses:

- severe kinematical constraints – limit dynamics
- effective model using KNO scaling and data-driven modelling of average multiplicities, forward/backward asymmetries, pT-dep. Etc...

At high hadronic invariant masses:

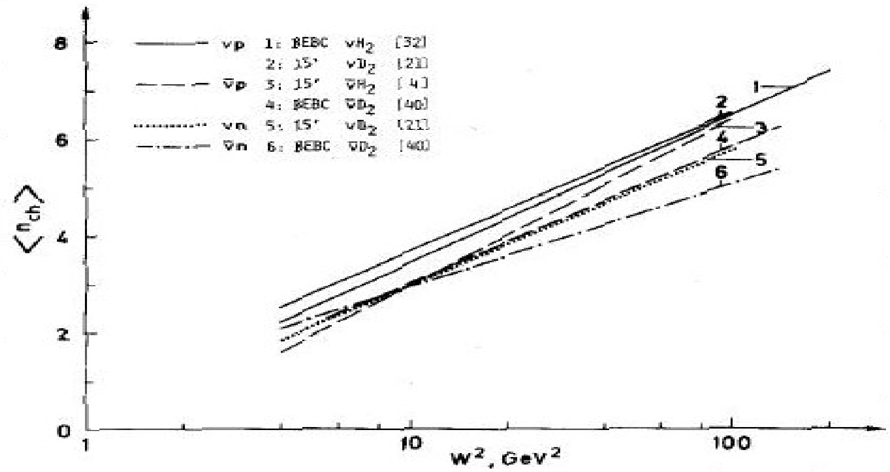
- rich dynamics
- using JETSET model
- tuned energy cutoff, pT, ssbar suppression (as in NUX)
- not really relevant at t2k energies



W (hadronic invariant mass)

Minos kinematical coverage at PH2LE beam (spans a large area of kinematical phase space space - t2k much more limited)

The GENIE hadronization / AGKY low-W model



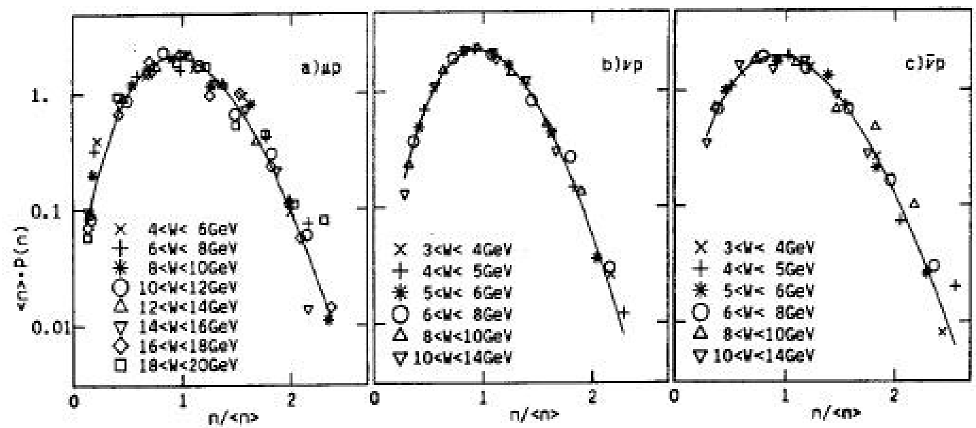
Get average multiplicity
from empirical parameterizations

<n> = a + b * lnW^2

Generate the actual multiplicity
using the KNO scaling law:

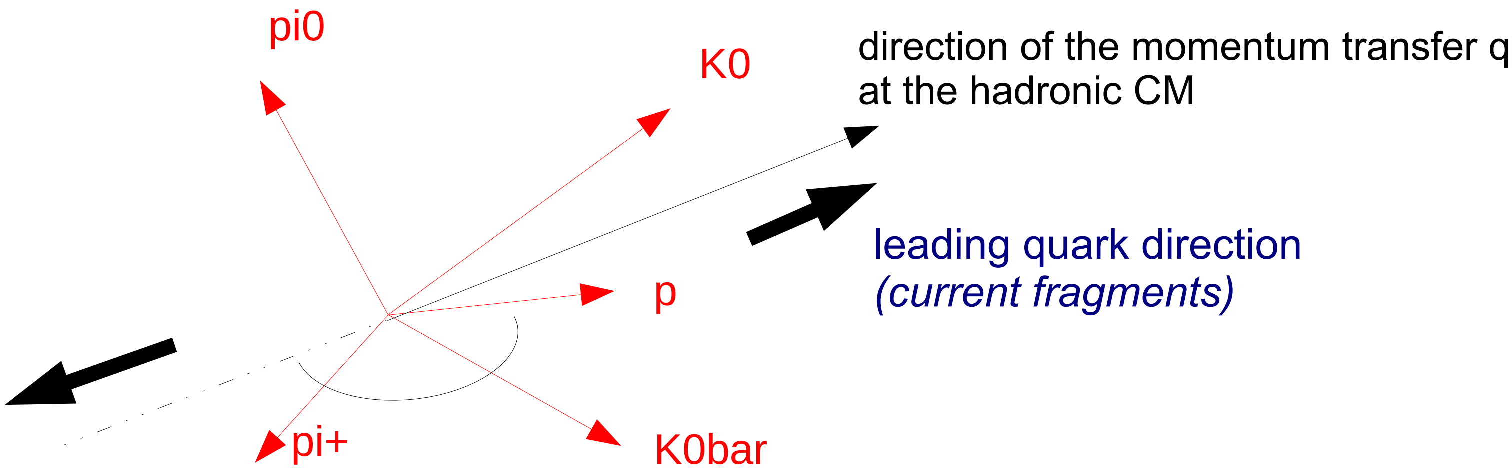
<n>P(n) = f (n/<n>)

(takin into account that
<n_neutral> = 0.5 * <n_ch>



Simple arguments (charge, isospin conservation) to derive particle spectrum

The GENIE hadronization / AGKY low-W model



At the hadronic CM, **the nucleon direction be should be correlated with the diquark direction** *(opposite to the direction of the momentum transfer q)*

The GENIE hadronization / AGKY low-W model

Building in experimental data on nucleon pT and xF (= pL/pLmax =2*pL/W)

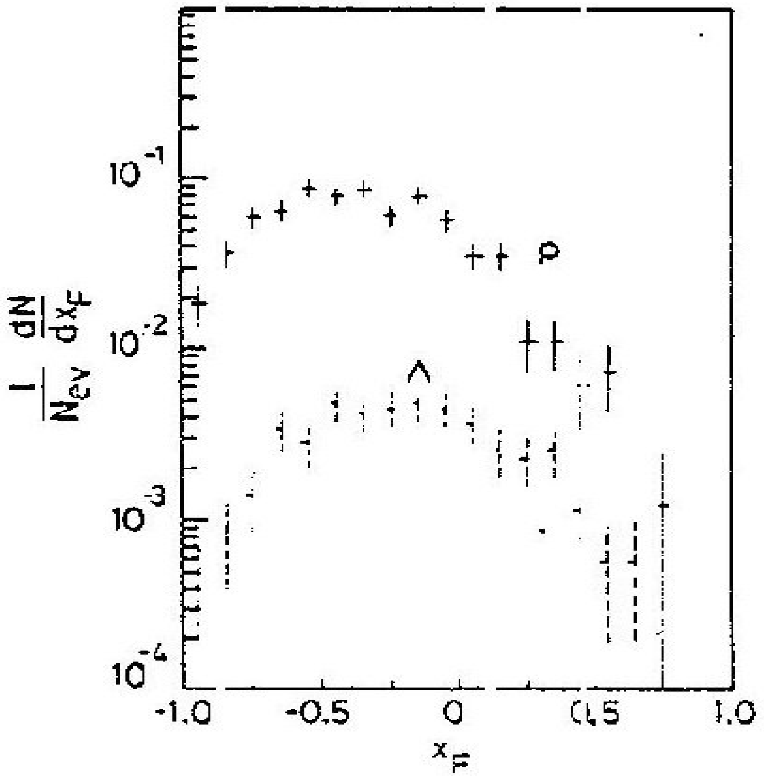
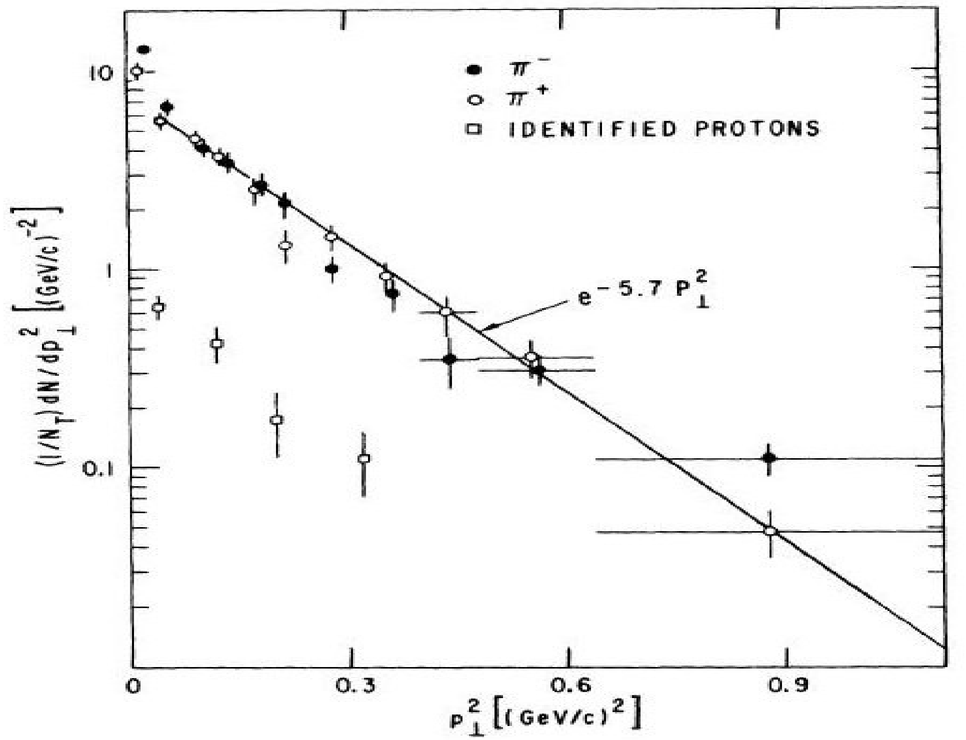


Fig. 11: x_F normalised distributions for protons (ν WA24 data) and lambda's (WA21 data, Bossetti et al, Nucl. Phys. B194, 1 (1982)).



Cooper, Neutrino 82, proceedings

The GENIE hadronization / AGKY low-W model

p4_{meson `remnants'} =
p4_{X} –
p4_{nucleon from target fragments}

Meson 4-momenta:

Boosting to the remnant hadronic system CM and performing a phase space decay.

A pT-limited decay to match experimental pion PT distribution.

pT-limited
phase space decay

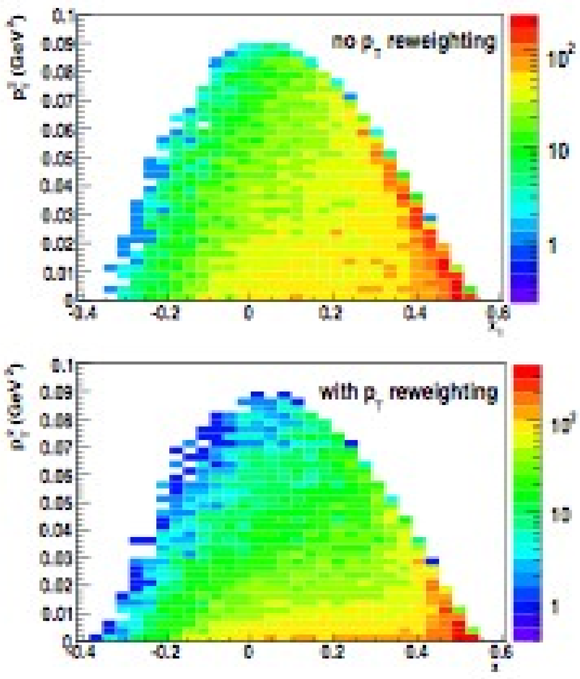


Figure 5. Pion x_F vs p_T^2, in the hadronic CM, for a (p, π^0, π^+) system decayed with invariant mass W = 1.6 GeV, where, for convenience, the

Formation zone

SKAT parameterization:

Hadron momentum

$$fzone = \frac{P \times ct_0 \times m}{m^2 + K \times P_T^2}$$

Transverse hadron momentum

K=0, ct0 = 0.342 fm

(SKAT) model dependence

No intranuclear rescattering within formation zone

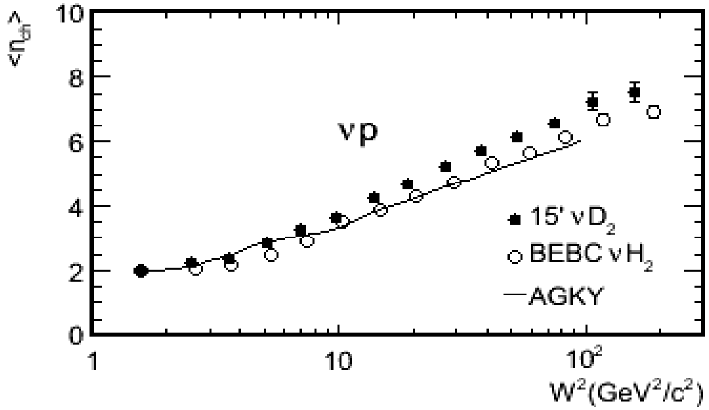
Used for hadrons generated by both the low-W KNO-based hadronization model
<u>and</u> for the small fraction of events hadronized by JETSET (override JETSET positions)

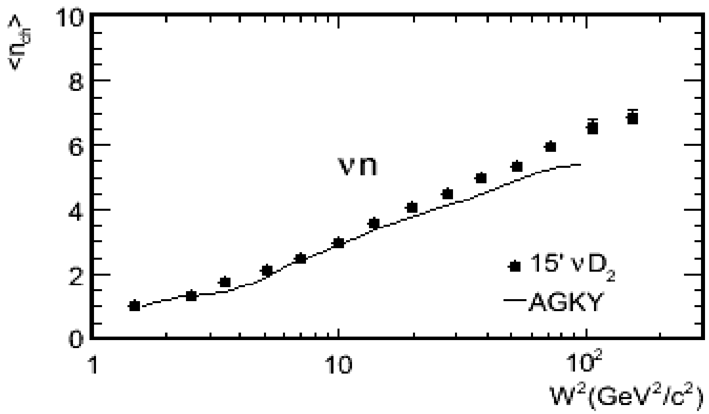
The GENIE hadronization model – Data/MC comparisons

Model does very good job against a diverse host of data

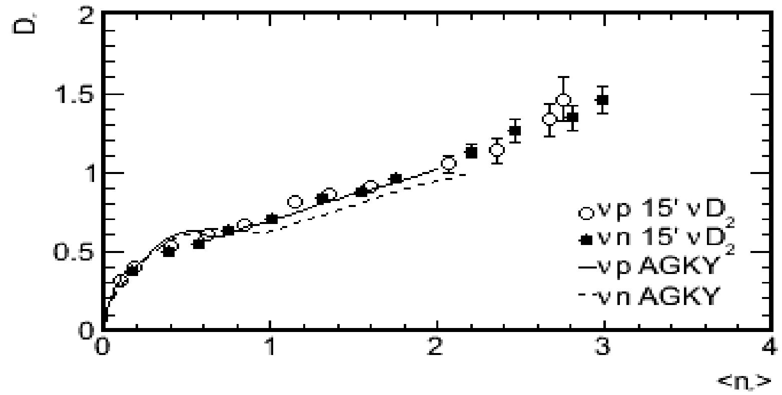
examples:

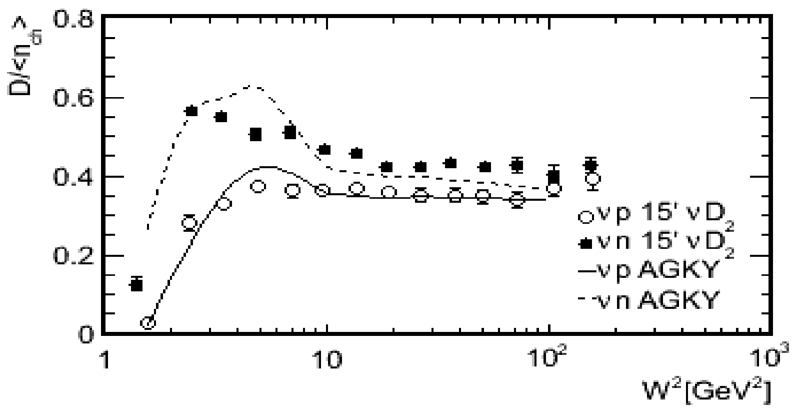
Charged pion multiplicities





Charged pion dispersion





Costas Andreopoulos, *Rutherford Appleton Lab.*

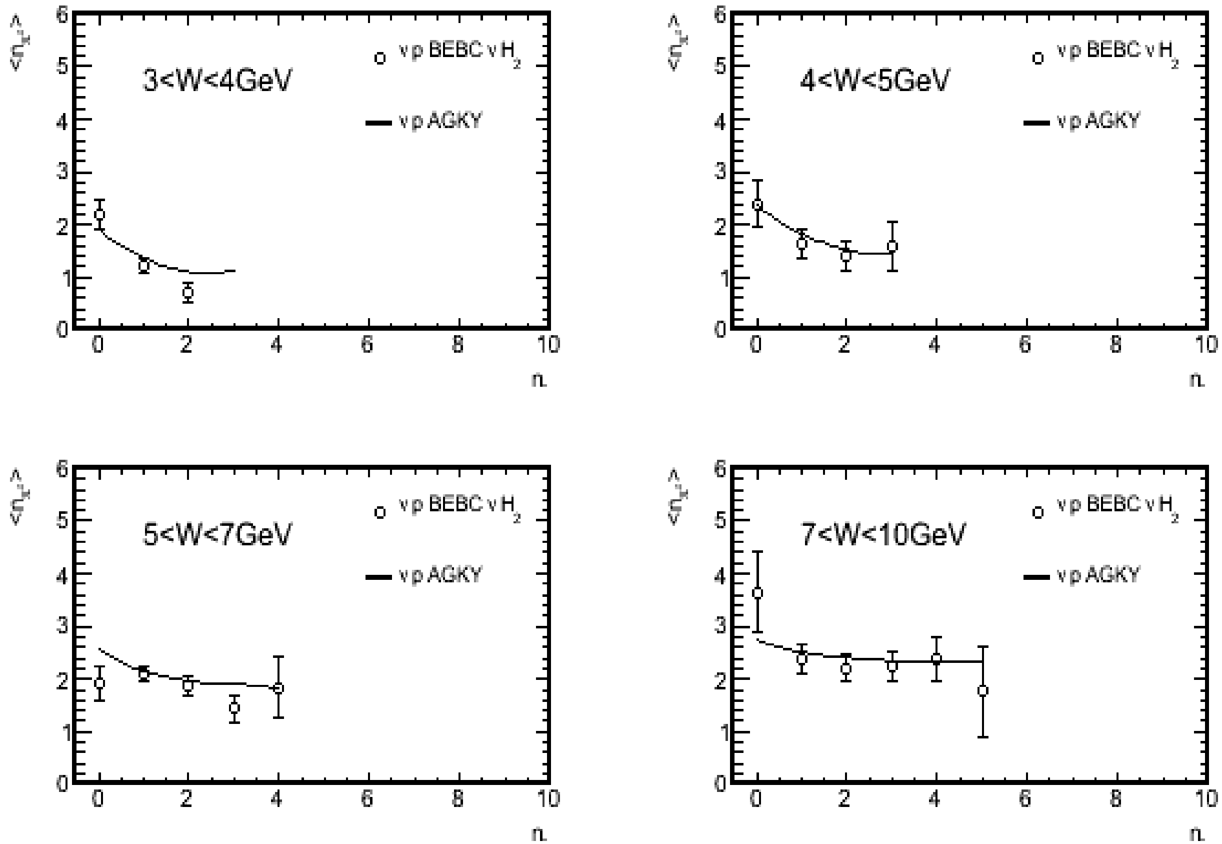
Science & Technology Facilities Council

The GENIE hadronization model – Data/MC comparisons

Model does very good job against a diverse host of data

example:

Neutral / charged pion correlation

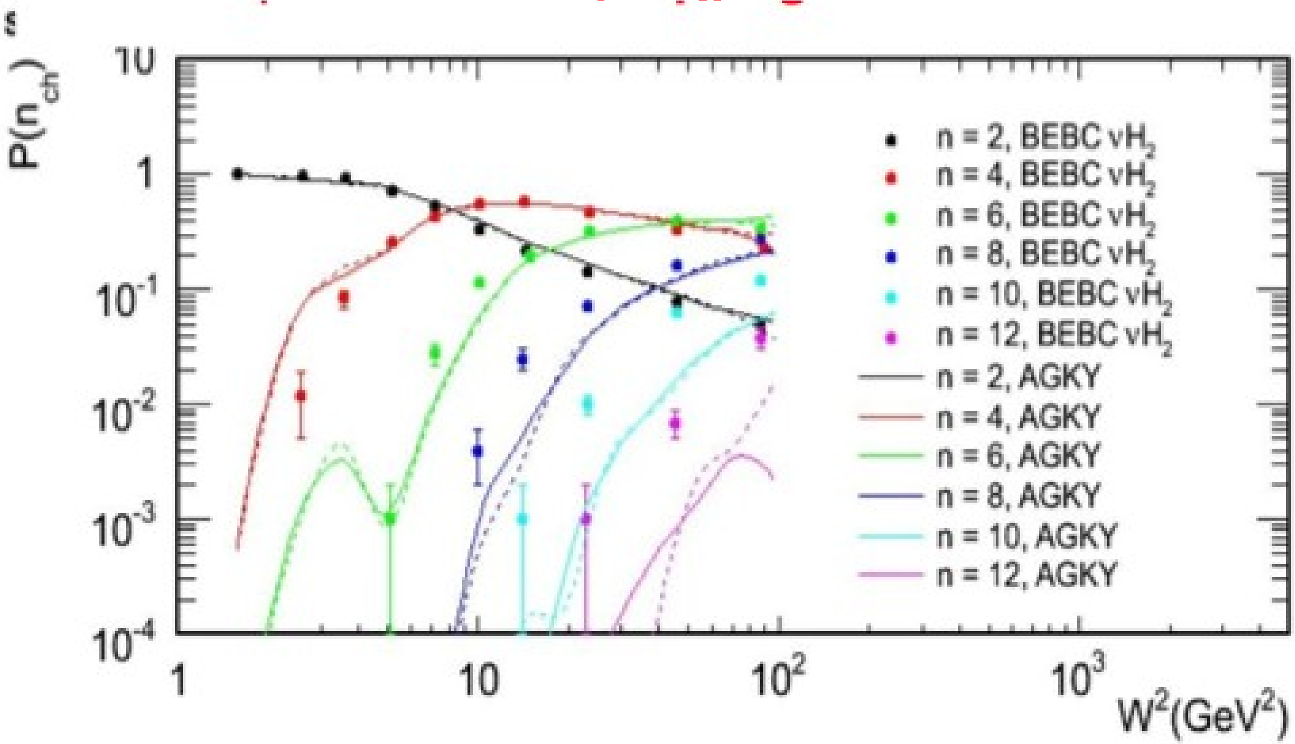


The GENIE hadronization model – Data/MC comparisons

Model does very good job against a diverse host of data

example:

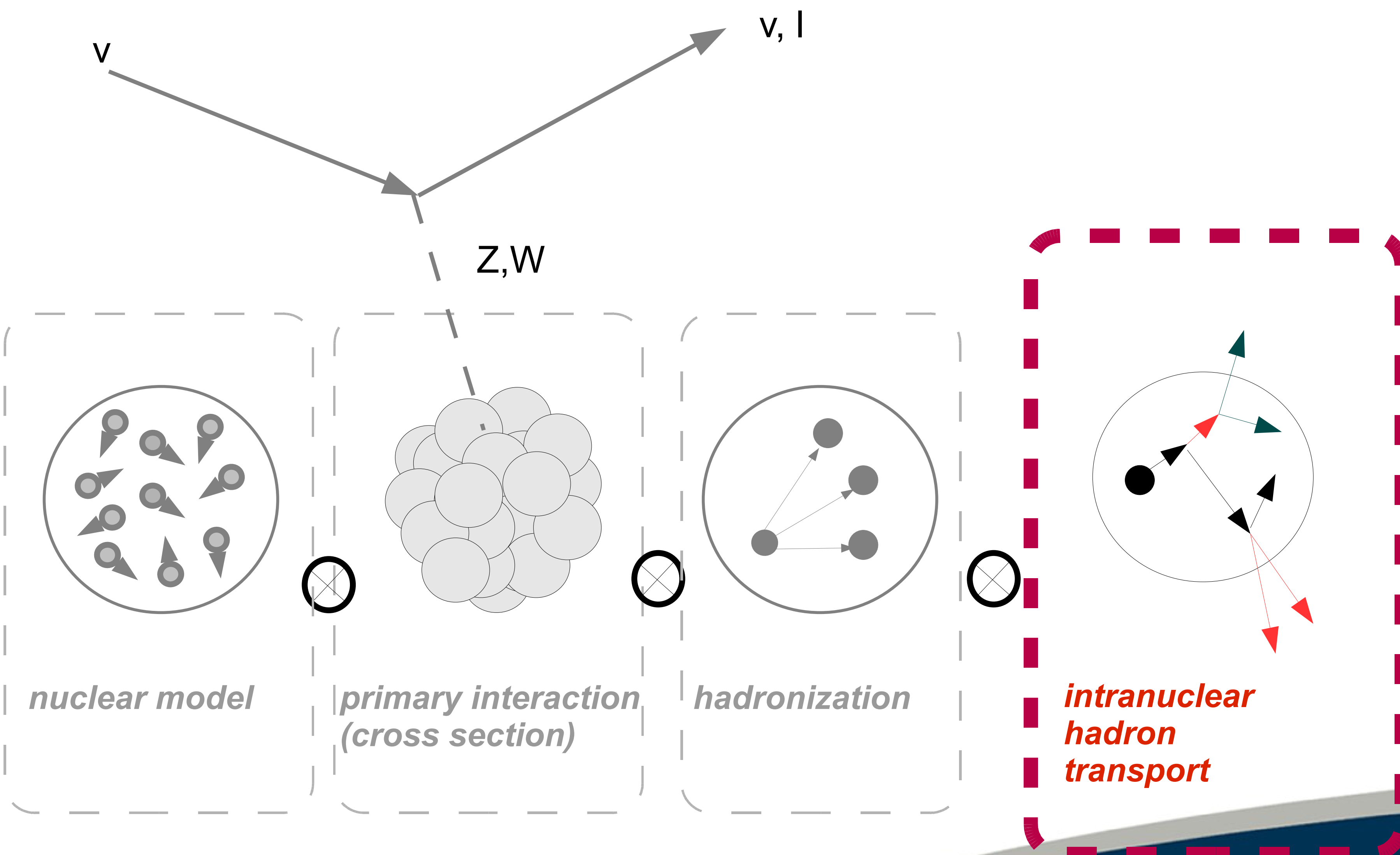
Normalized topological cross sections



For more data/mc comparisons see GENIE-PUB/2007/002 and T.Yang's talk/proceedings at NuINT07

The model and its shortcomings are very well understood.
Improvements for low multiplicity (n=2) hadronic systems under way

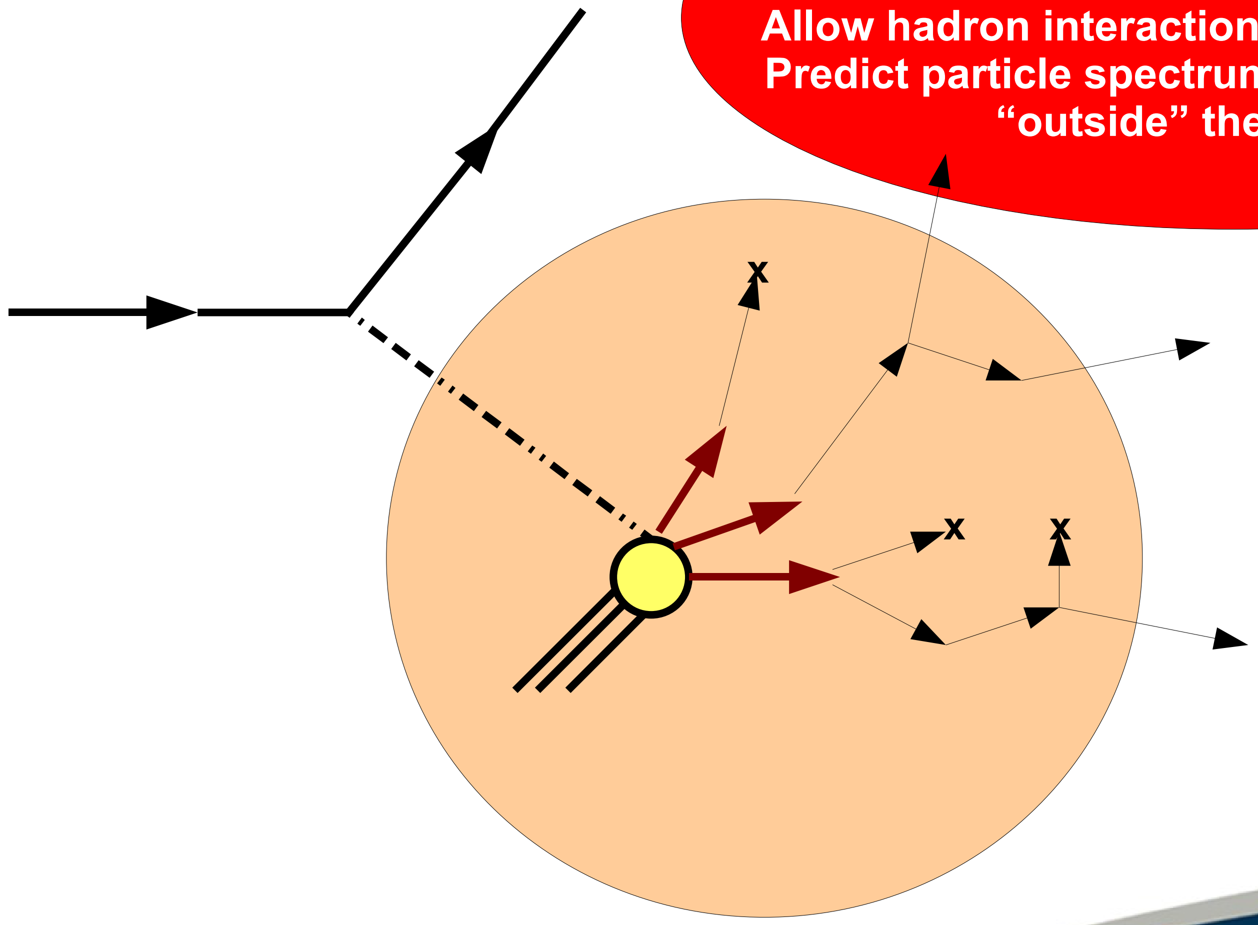
Intranuclear hadron transport



Science & Technology Facilities Council

The GENIE hadron transport modelling

what does it mean?



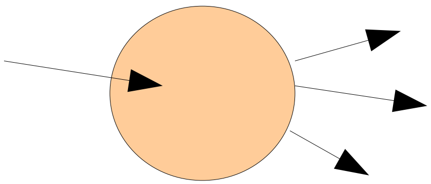
Transport primary (and secondary, tertiary, ...) hadrons out of the hit nucleus.
Allow hadron interactions in the nuclear matter.
Predict particle spectrum & particle 4-momenta "outside" the hit nucleus

The GENIE hadron transport modelling

Currently **have 2 alternative models** (using different techniques) –

Development of both is **led by Steve Dytman**

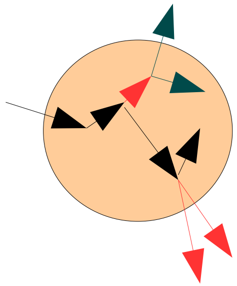
Intranuke / hA
(effective MC)



Anchored to a large body of experimental data
(including hadron+nucleus data)

available since 2.0.0

Intranuke / hN
(true cascade MC)



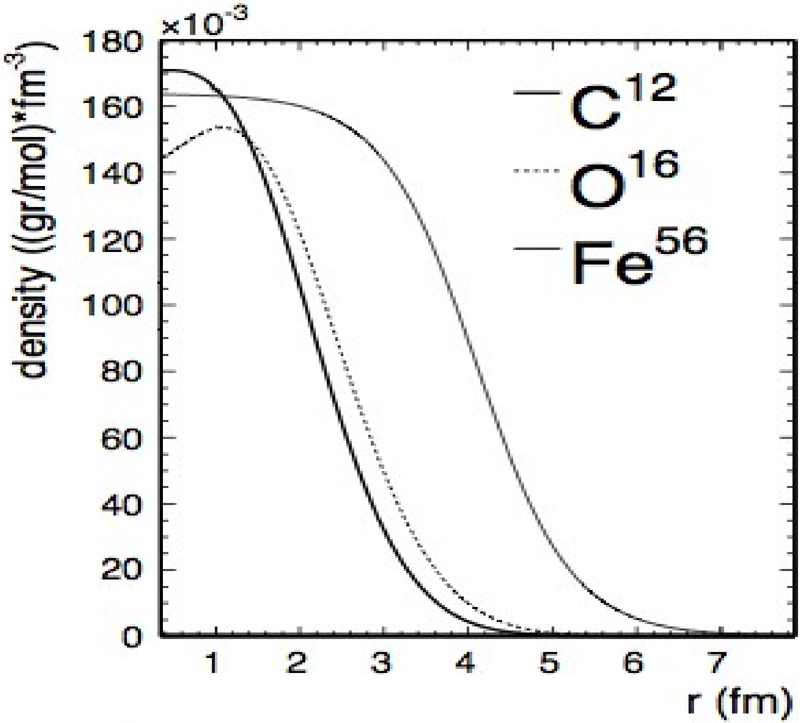
Builds everything up from hadron-nucleon xsecs

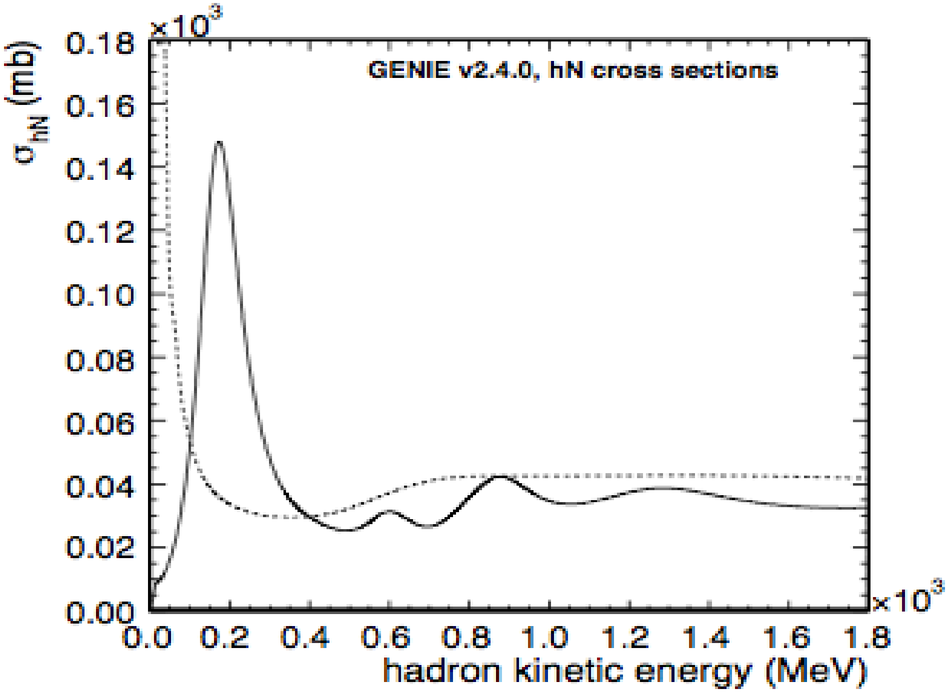
In advanced development stage
to become available soon

The GENIE hadron transport modelling (INTRANUKE/hA)

Stepping primary hadrons within the target nucleus

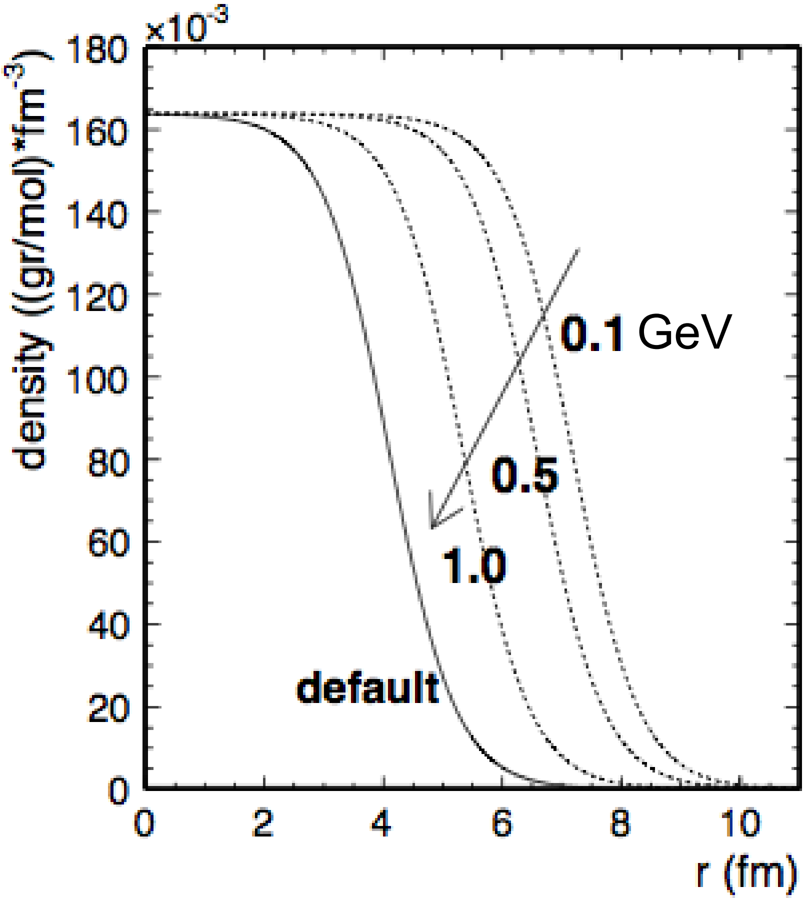
$$P^h_{rescat} = 1 - P^h_{surv} = 1 - \int e^{-r/\lambda(\vec{r},h,E_h)} dr$$





Costas Andreopoulos, ***Rutherford Appleton Lab.***

The GENIE hadron transport modelling (INTRANUKE/hA)



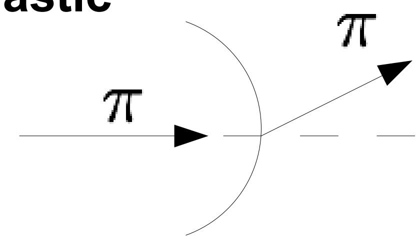
- Hadrons stepped by 0.05 fm at a time
-
- Hadrons traced till they reach
  r_max = N * R_nucl = N * R0 * A^(1/3)
  (R0 = 1.4, N = 3.0)
  so as to include the effects of the tails
  (Fe56: R_nucl=5.36fm, r_max=16.07fm)

- The nuclear density distribution is `stretched' by n times the de Broglie wavelength of the tracked particle (n=1 for nucleons, n=0.5 for pions).

The GENIE hadron transport modelling (INTRANUKE/hA)

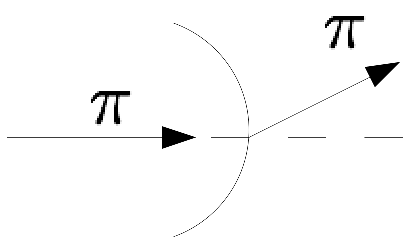
INTRANUKE/hA considers 5 types of 'hadron fates' (some may include many channels)

elastic



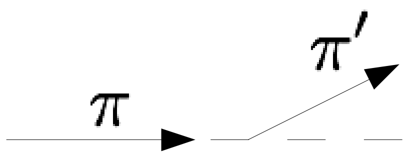
Pion deflected. Its kinetic energy stays the same.

inelastic

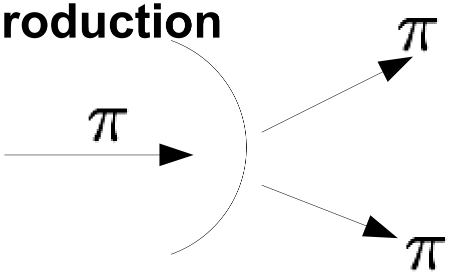


Pion deflected. Its kinetic energy is degraded.

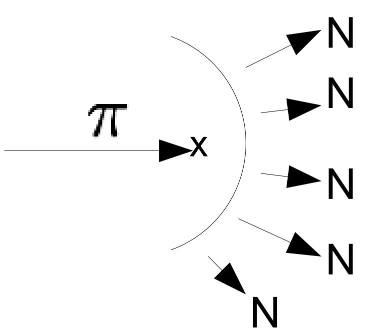
charge exchange



pion production



absorption

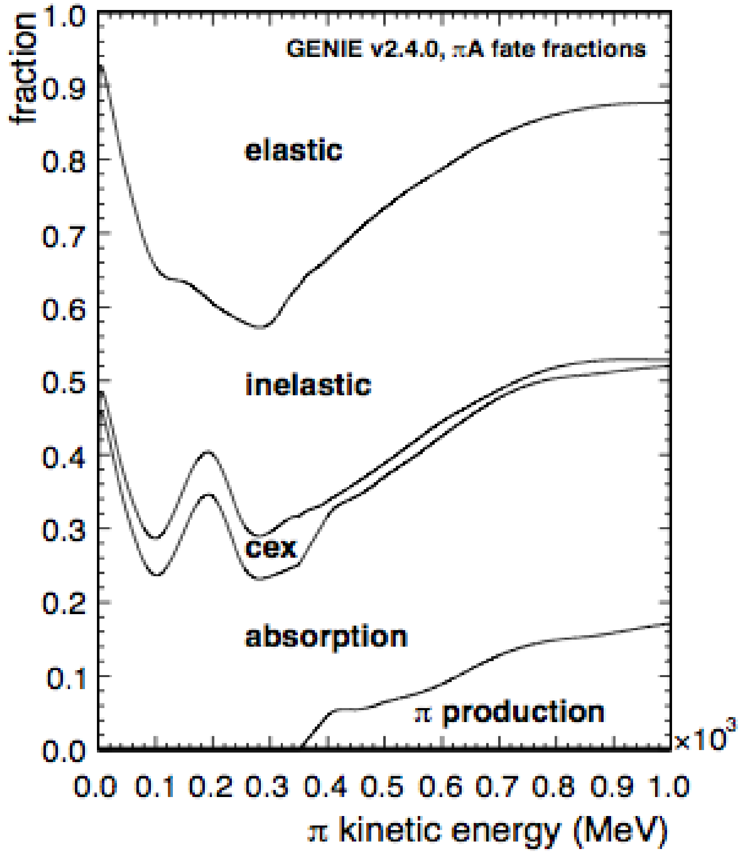


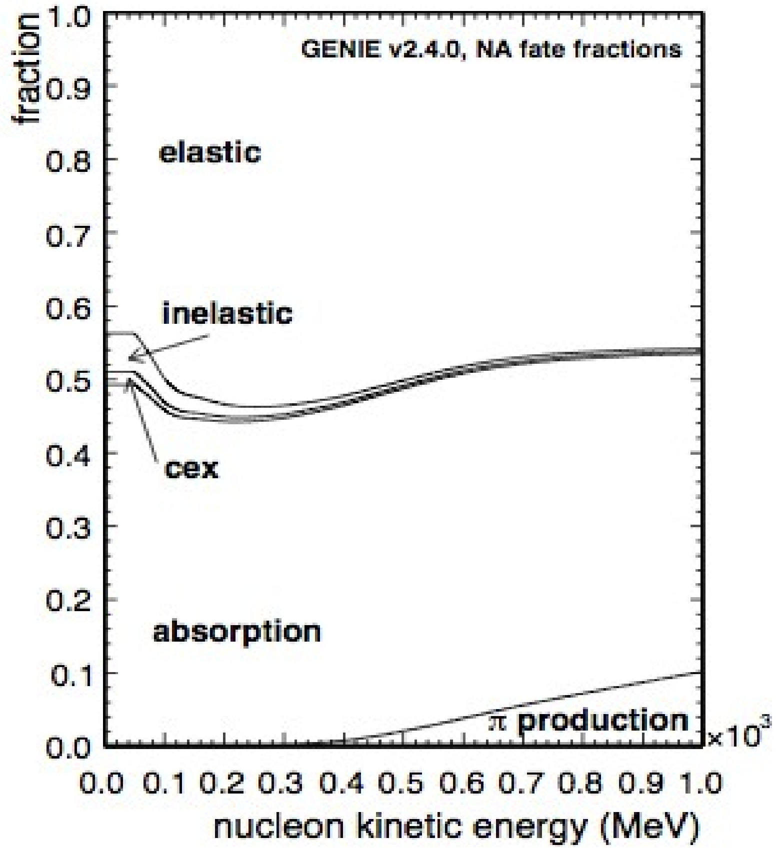
followed by emission of low energy nucleons

~ Similar fates for nucleons

The GENIE hadron transport modelling (INTRANUKE/hA)

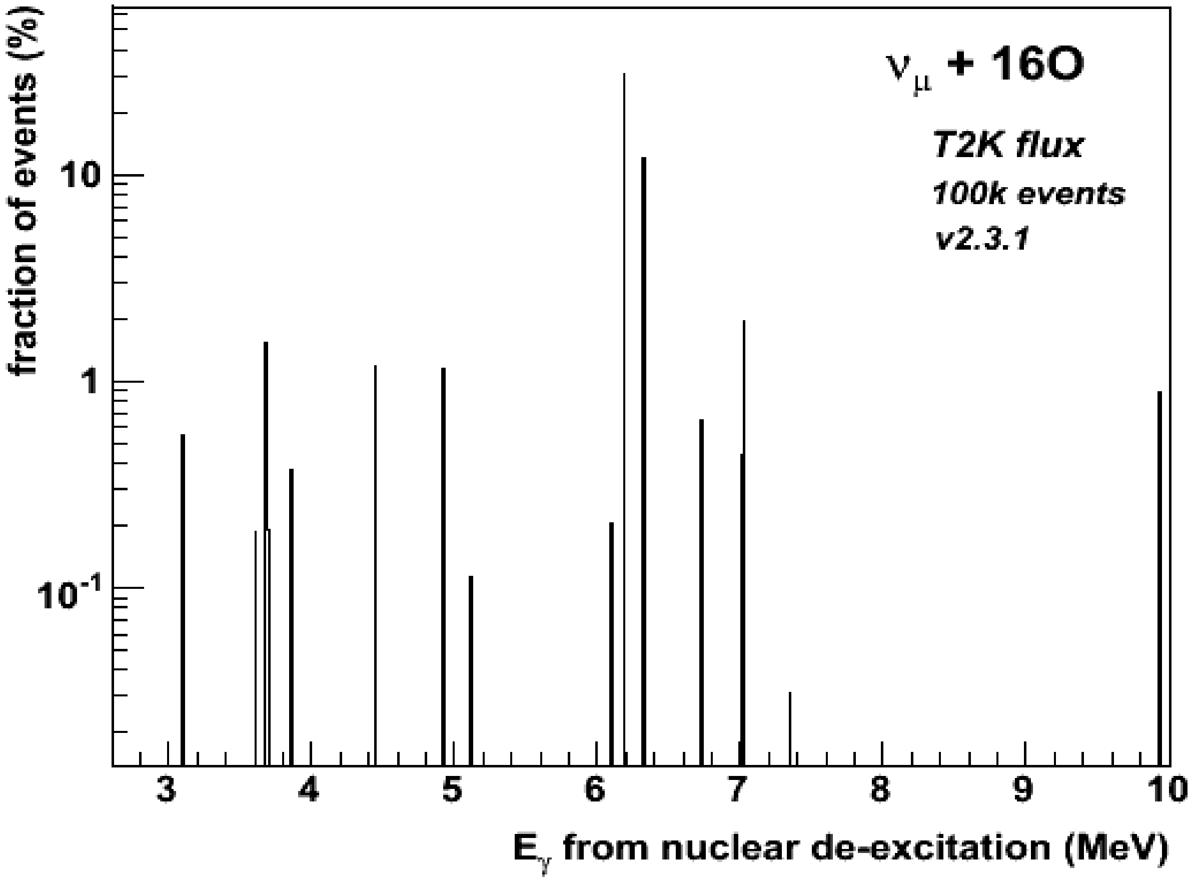
Fractions taken mostly from data





Costas Andreopoulos, *Rutherford Appleton Lab.*

Nuclear excitations



Included in an ad-hoc way

Only for O16

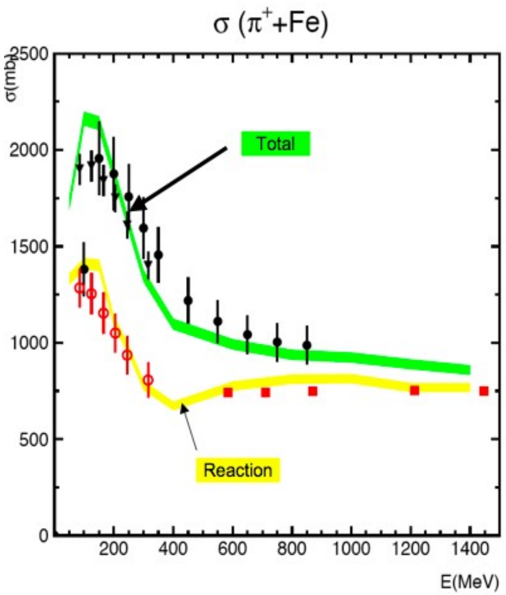
INTRANUKE/hA Data/MC comparisons

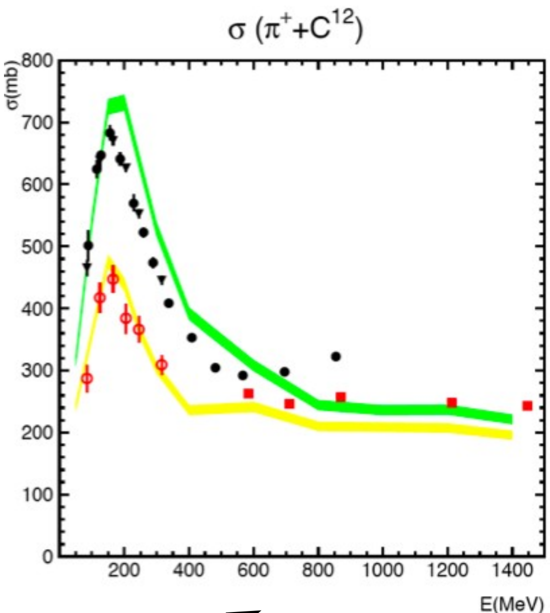
Much effort went into validation –
utilizing experience from non-neutrino probes, mainly hadron+A reactions

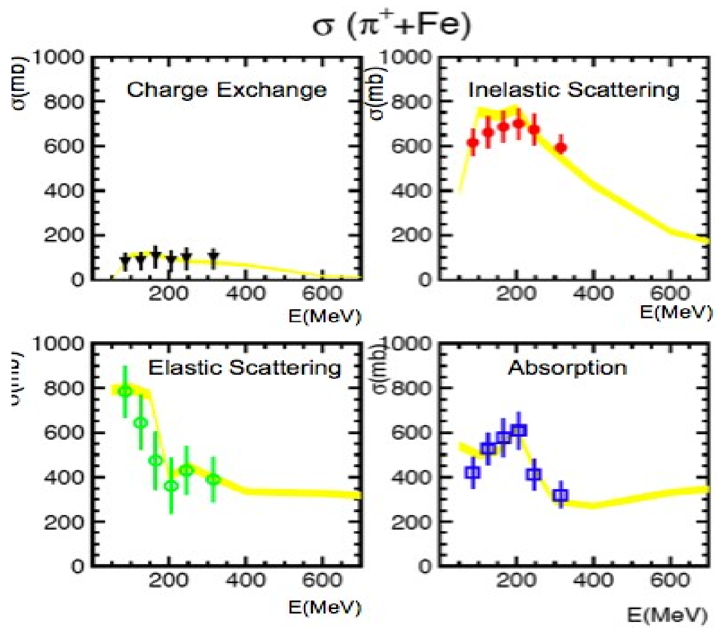
Lot of effort in tuning mean free path &
including the elastic contrib – difficult to model in context of INC

total = reaction + elastic
reaction = cex + inel + absorption + pi prod

Then, components modelled directly from data – requires total xsec to be modelled correctly first



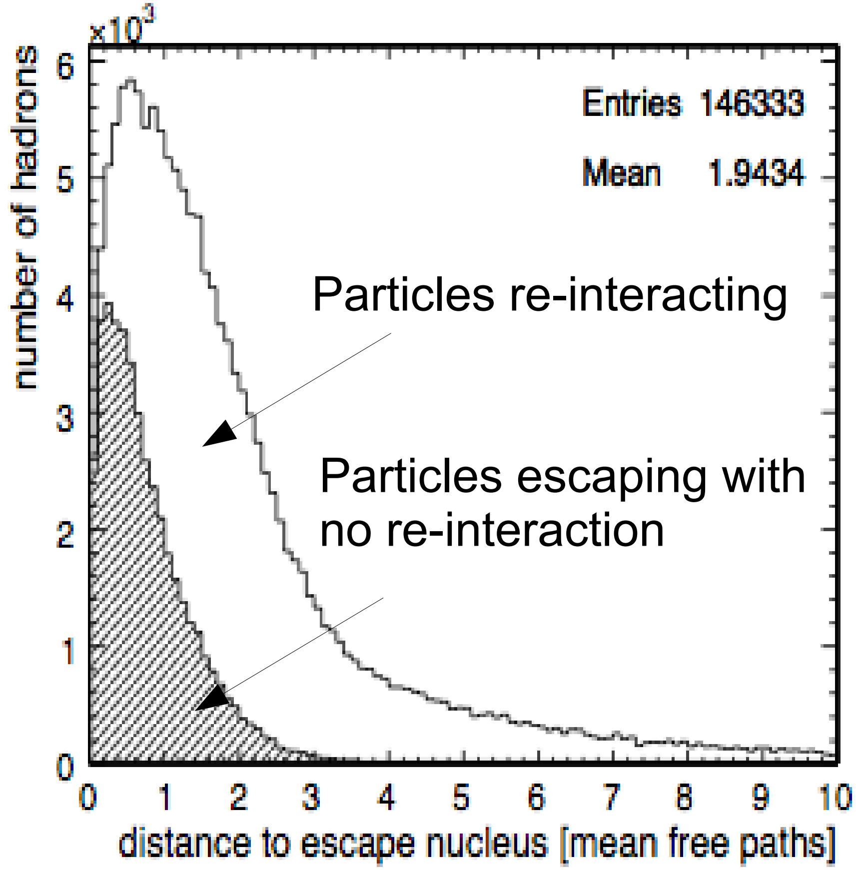




`MC experiments': throw hadrons into nuclei,
'measure cross sections' and compare with data.

Costas Andreopoulos, *Rutherford Appleton Lab.*

Intranuclear rescattering effect



nu_mu + C12

1 GeV

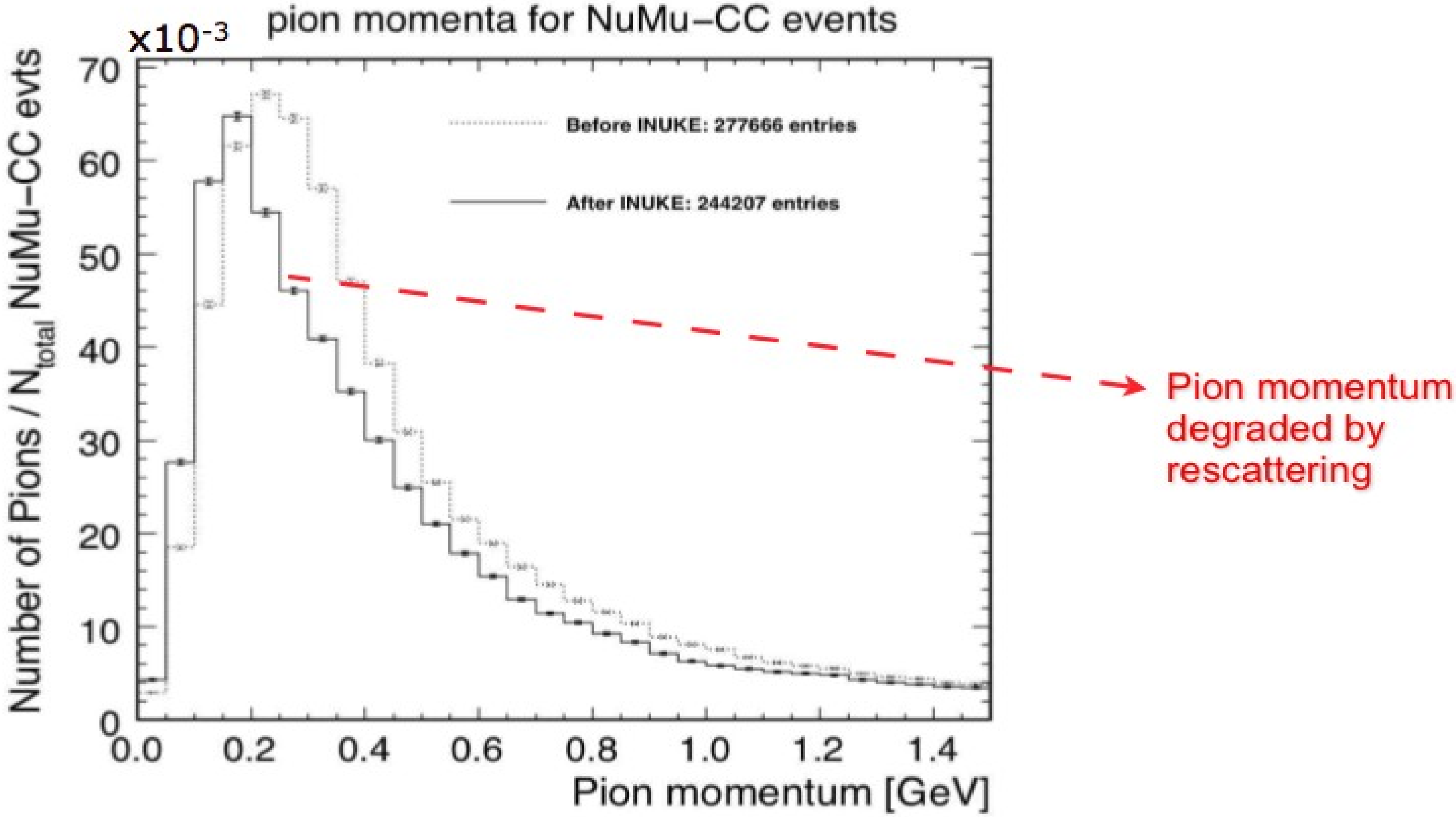
Most particles (2/3) re-interact

Intranuclear rescattering effect

Severe effect on observed topologies

Topology after	Topology before										
	$0\pi X$	$\pi^0 X$	$\pi^+ X$	$\pi^- X$	$\pi^0\pi^+ X$	$\pi^0\pi^- X$	$\pi^+\pi^- X$	$2\pi^0 X$	$2\pi^+ X$	$2\pi^- X$	$\geq 3\pi X$
$0\pi X$	6053177	291116	520783	72611	9949	1843	6236	3037	2073	195	2390
$\pi^0 X$	26265	902112	87831	11465	42229	7916	1746	23933	616	49	10371
$\pi^+ X$	42820	26243	1655899	481	41826	157	24599	483	16408	0	12490
$\pi^- X$	4502	24564	15	243424	700	7874	24536	435	0	1253	6633
$\pi^0\pi^+ X$	9948	21378	28679	5758	194323	594	5082	2770	2877	24	41100
$\pi^0\pi^- X$	0	44	2	1	93	35773	3630	1690	0	198	17552
$\pi^+\pi^- X$	16804	183	146	1846	3058	584	108396	38	0	3	40218
$2\pi^0 X$	0	0	0	0	6002	1171	113	54246	52	0	21323
$2\pi^+ X$	1225	128	9496	19	3533	1	298	24	37812	0	18160
$2\pi^- X$	0	0	0	13	0	584	0	20	0	2833	2891
$\geq 3\pi X$	5352	6480	11459	2221	13563	2661	8282	4133	2416	126	566980
Total	6160093	1272248	2314310	337839	315276	59158	182918	90809	62254	4681	740108

Intranuclear rescattering effect



Science & Technology Facilities Council

GENIE models – A summary

- **Cross section model**
  - QEL: Llewellyn-Smith with any of Sachs/BBA03/BBA05 elastic f/f
  - RES: Rein-Sehgal
  - COH pi production: Rein-Sehgal / includes updated PCAC
  - DIS: latest Bodek-Yang
    - Including parametrization of the longitudinal structure function FL
    - Including NuTeV parameterization of nuclear effects
  - Many other more rare channels: DIS & QEL charm / ve- elastic / inv.mu-decay/...
- **Nuclear model**
  - Fermi Gas model
  - Including high momentum tail due to N-N correlations modelled from eN data
  - "Standard" FG prescription for off-shell kinematics...

Fairly standard at all ν MCs

Careful implementation as MINOS spans a huge kinematical region (E ~ <1 to >100 GeV)

- **Transition region cross section modelling**
  - Non resonance background modelled from DIS & AGKY hadronization
  - Tuned to the world exclusive multi-pion cross section data
- **Neutrino-induced primary hadronic shower modelling**
  - AGKY
  - Effective KNO-based hadronization at low-W
  - Switching gradually to PYTHIA/JETSET at high-W
  - SKAT-type formation zone parametrization
- **Intranuclear hadron transport**
  - INTRANUKE/hA model
  - Anchored to a set of hadron+Fe56
  - Scaled to all nuclei

Unique to GENIE