Review of Smith-Monitz model update
A proposal to improve framework

Genie

version 3

I.D. Kakorin, K.S. Kuzmin, V.A. Naumov

April 8, 2022

JINR

Genie

Outline

Goals

Smith-Monitz model update

Comparison with data

CacheBranchFx issue

New normalization channel

Edge issue

Goals

The presentation covers two main issues:

1. An update of the Smith-Monitz model to more natural simulation of the initial nucleon kinematic.
2. A suggestion for a few improvements to the GENIE framework that came up while working on the first point:
   - rearrangement of the class *CacheBranchFx*;
   - introducing a new artificial channel, so-called *Normalization channel*;
   - fixing the previously reported issue of extending the energy range beyond the user defined in the class *GMCJDriver*.

Smith-Monitz model update

- To speed up the event generation we previously implemented the following ersatz:
  1. Generate 4-momentum transfer $q = (\nu, \mathrm{q})$ according to a distribution

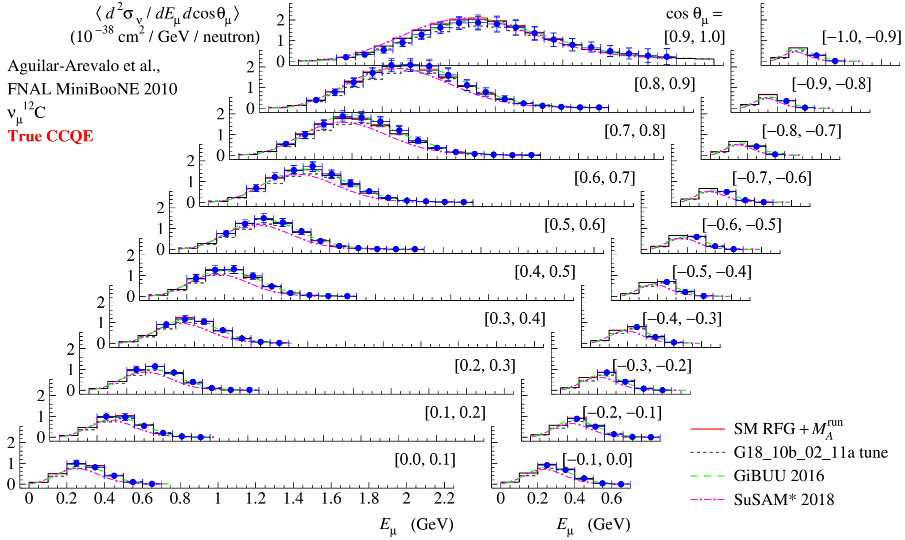
     $$\frac{d^2\sigma}{dQ^2 d\nu} = \int d\mathrm{p} \frac{d^3\sigma}{dQ^2 d\nu d\mathrm{p}} = \int d\mathrm{p} f(\mathrm{p}, \mathrm{q}) W_{\alpha\beta}(p, q) L^{\alpha\beta}(k, q),$$

     where $k = (E_{\mathrm{k}}, \mathrm{p})$ and $p = (E_{\mathrm{p}}, \mathrm{p})$ are the 4-momenta of initial lepton and nucleon, $f(\mathrm{p}, \mathrm{q})$ is a function that takes into account the Pauli blocking and distribution of the initial nucleon momentum p.
  2. Generate independently the momentum p according to the distribution given by the function $f(\mathrm{p}, \mathrm{q})$ with the momentum q generated in the previous step.
- To “frankly” generate an event, one needs to generate both q and p simultaneously according to the distribution $\frac{d^3\sigma}{dQ^2 d\nu d\mathrm{p}}$. The “ersatz” and “frankly” methods would be equivalent if the relative variation of $W_{\alpha\beta}(p, q)$ were small that is $|\Delta W_{\alpha\beta}(p, q)| \ll |W_{\alpha\beta}(p, q)|$. But this is not always the case, and the effect is sometimes not negligible.

Smith-Monitz model update

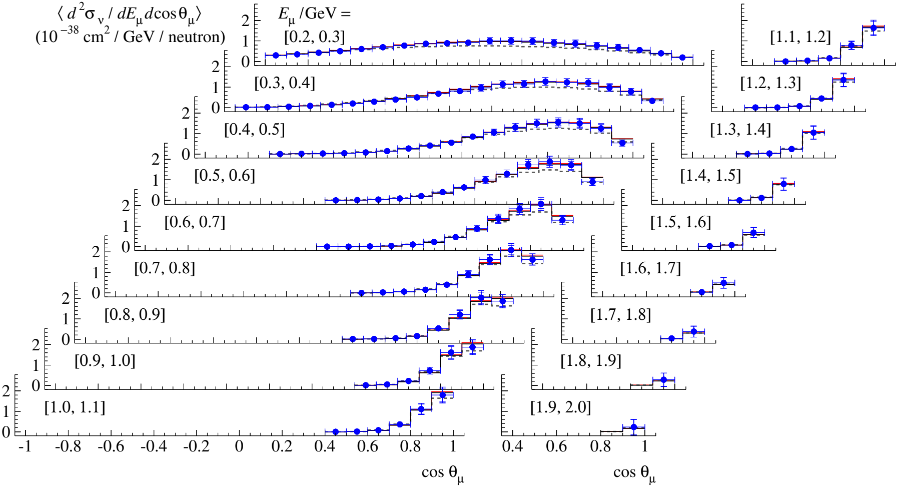
- The Smith-Monitz model + running axial mass (M_A^{run}) ansatz very well describes the CCQE and CCQE-like distribution over the final lepton kinematic variables obtained by MiniBooNE, but it is not so good in describing the T2K and MINERvA distributions over both charged lepton and final hadronic kinematic variables. The last two experiments also apply kinematic cuts to the leptonic and hadronic variables. A comparison of the model with the recent NOvA data shows that the distribution in leptonic variables is described much better than the distribution in hadronic ones. This prompted us to implement "frankly" way for the final nucleon generation.
- To do this, we had to optimize the searching for the maximum. In order not to slow down the generation of events, we tried to make the safety factor as close to 1 as possible.
- In addition, we have adjusted the kinematics (see http://theor.jinr.ru/NeutrinoOscillations/Papers/GENIE_formulas.pdf for detail) but it had little effect. Also, we commented the code more carefully.

Comparison with data: MiniBooNE

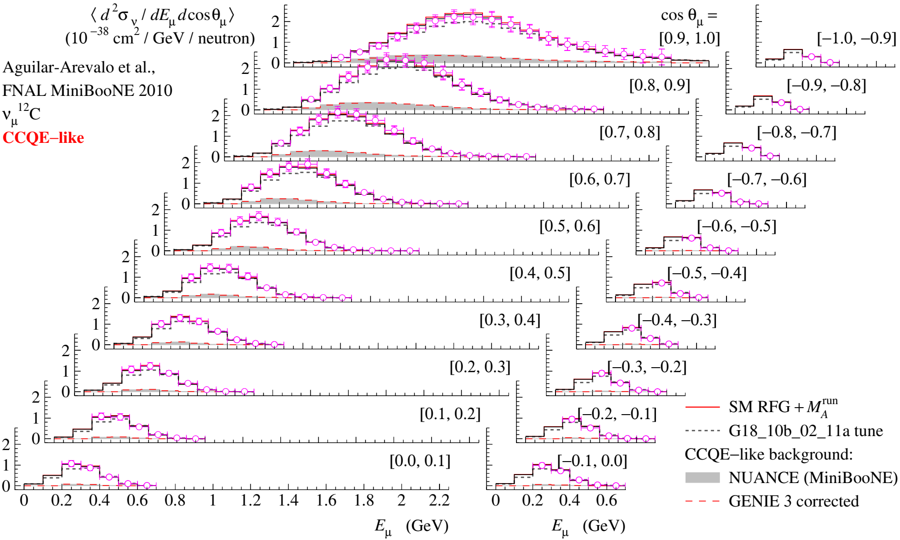


Here and below, the **red**/**black** histograms show the **old**/**new** simulation.

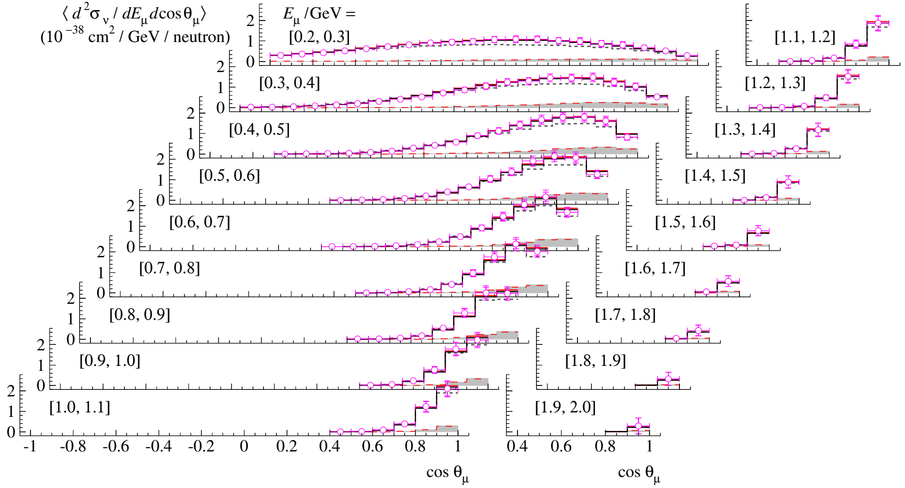
Comparison with data: MiniBooNE



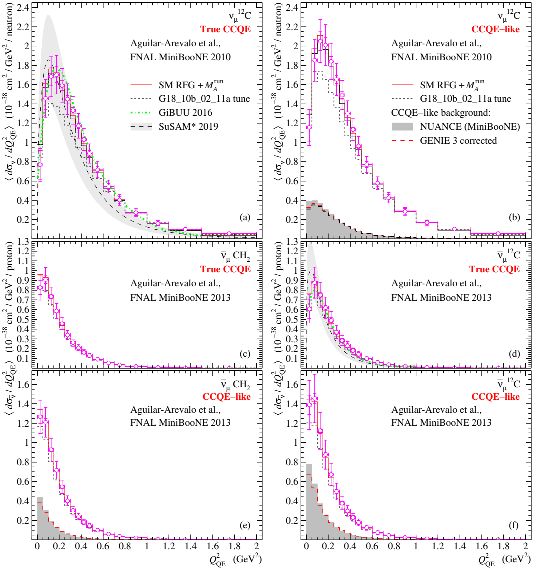
Comparison with data: MiniBooNE



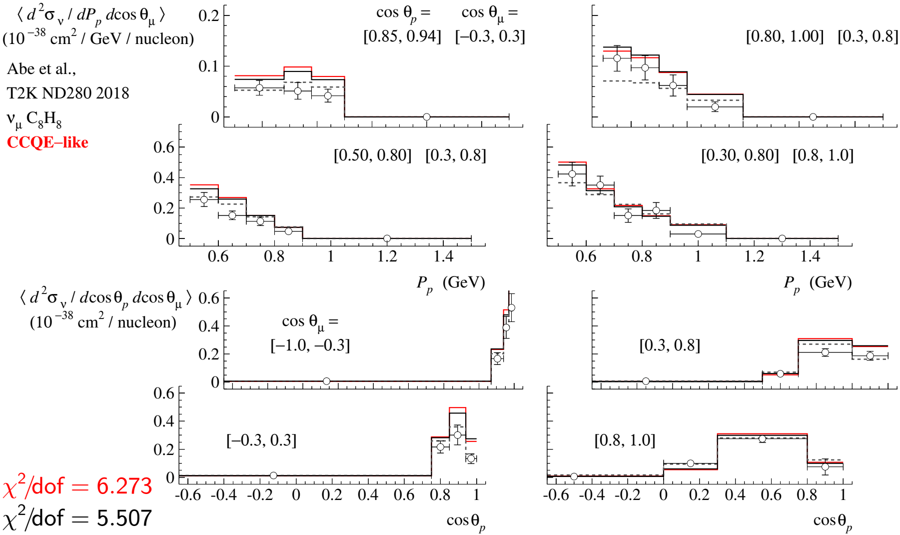
Comparison with data: MiniBooNE



Comparison with data: MiniBooNE



Comparison with data: T2K



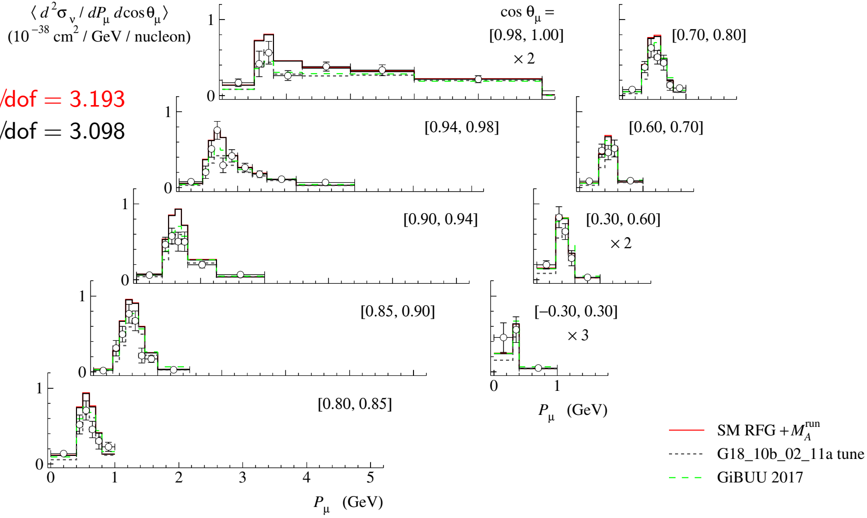
$\chi^2/\mathrm{dof} = 6.273$

$\chi^2/\mathrm{dof} = 5.507$

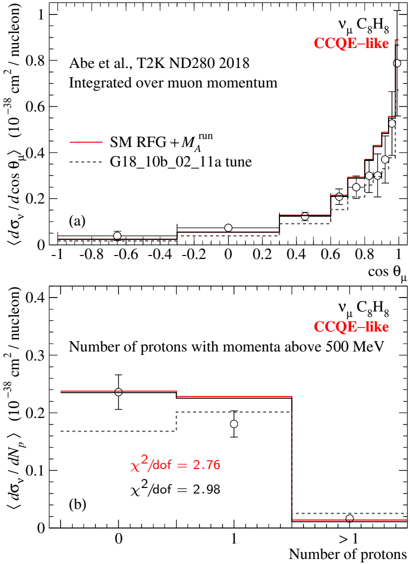
Comparison with data: T2K

$\chi^2/\text{dof} = 3.193$

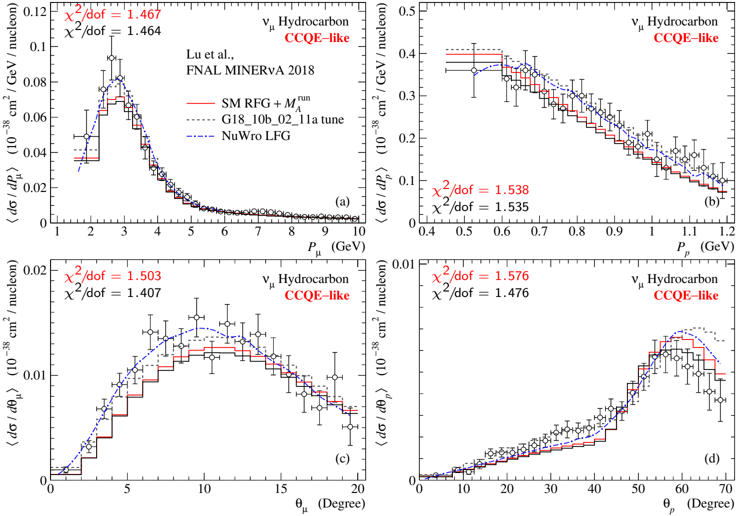
$\chi^2/\text{dof} = 3.098$



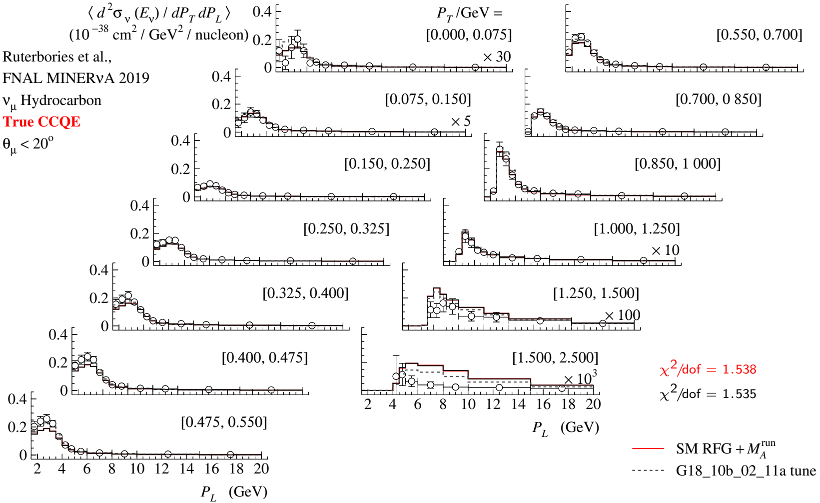
Comparison with data: T2K



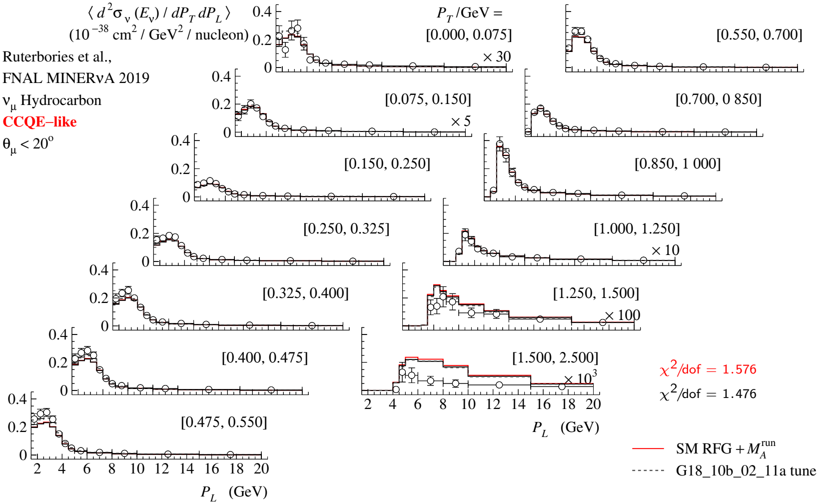
Comparison with data: MINERvA



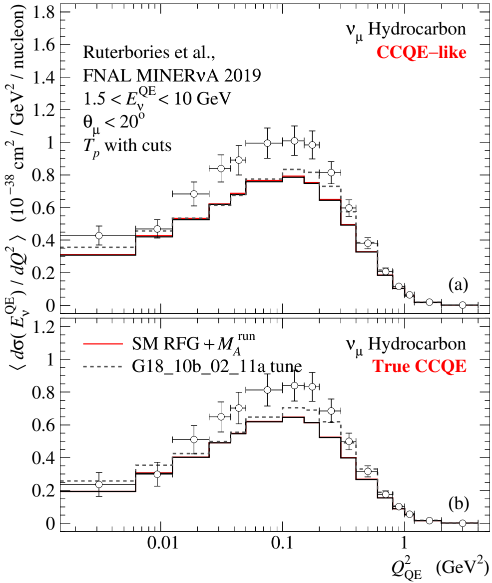
Comparison with data: MINERvA



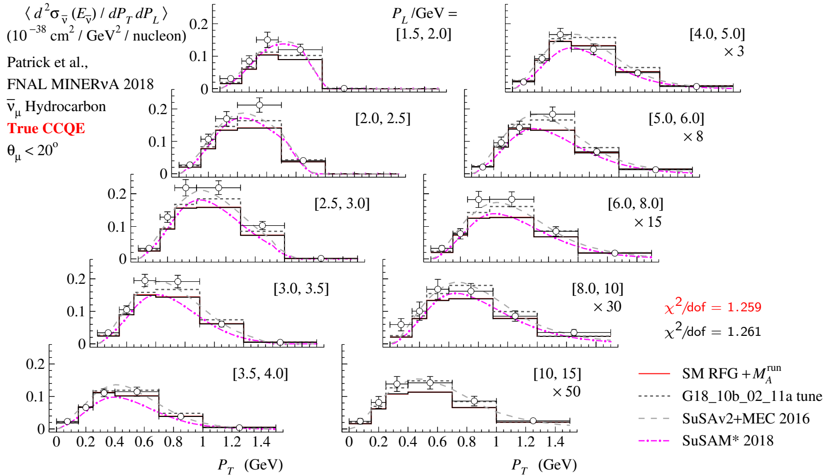
Comparison with data: MINERvA



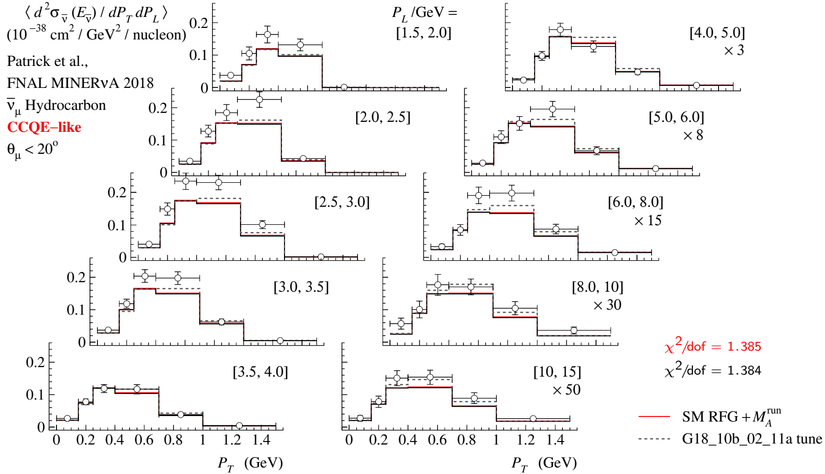
Comparison with data: MINERvA



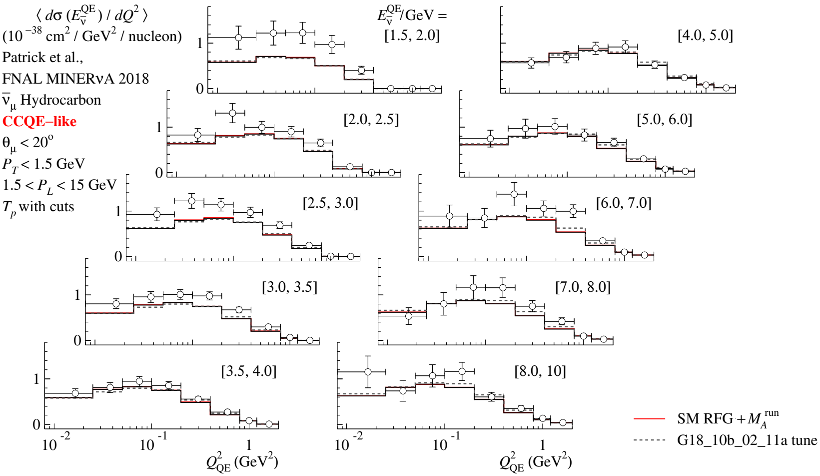
Comparison with data: MINERvA



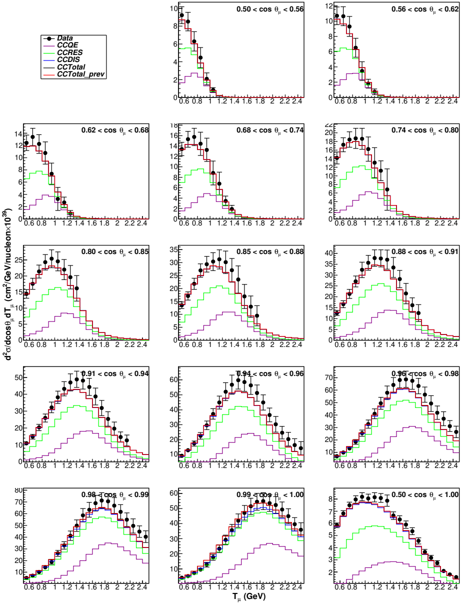
Comparison with data: MINERvA



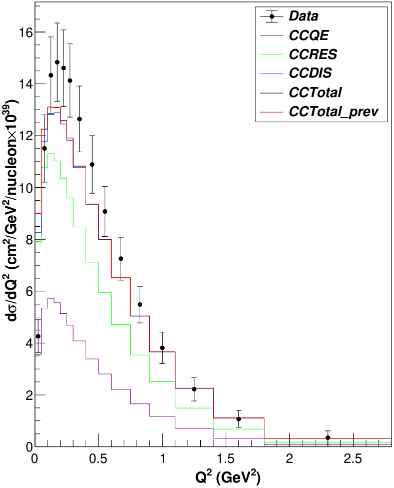
Comparison with data: MINERvA

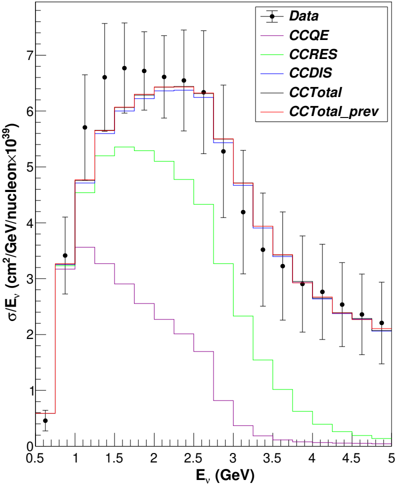


Comparison with data: NOvA



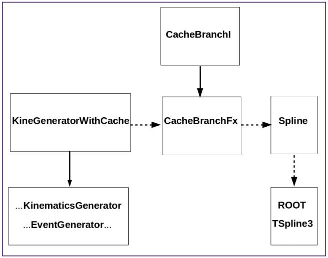
Comparison with data: NOvA

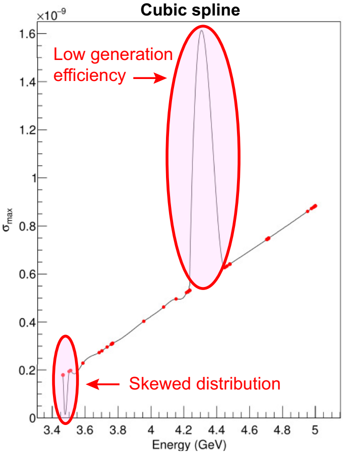




CacheBranchFx issue

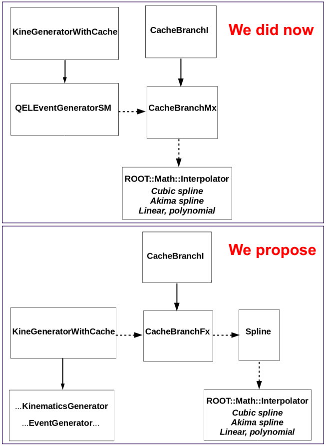
The caching mechanism has a flaw: since the maximum values caching at random energies are interpolated by a **cubic spline**, the output of *CacheBranchFx* at intermediate energies are sometimes weird.

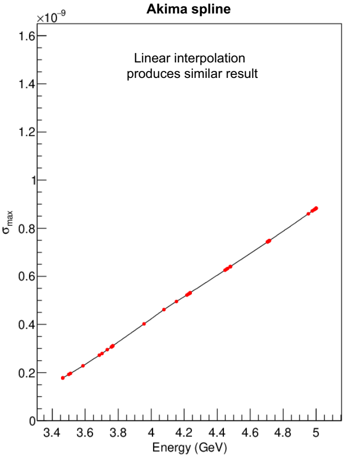




CacheBranchFx issue

To resolve the problem (wich is not specific for the SM model!), we propose to use *ROOT::Math::Interpolator* instead of *TSpline3* as interpolator. It also provides the functionality of the cubic spline interpolator along with others.

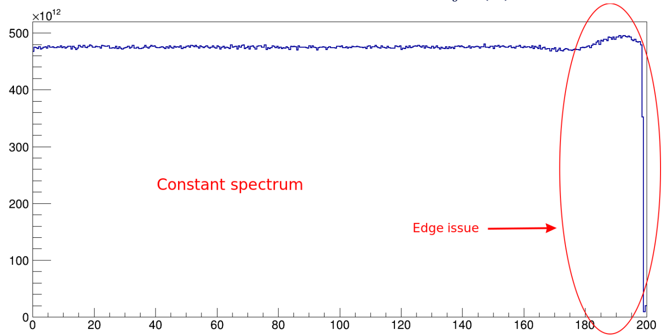




New normalization channel

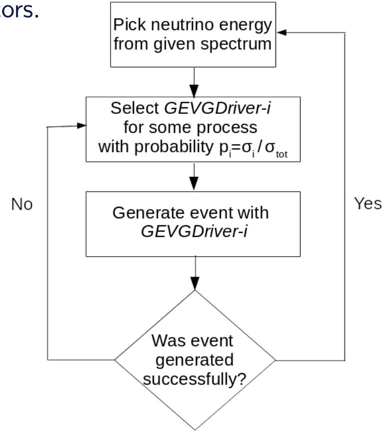
We suggest to introduce a new channel – *normalization channel*. Its main property is a constant cross section per nucleon over the whole energy range. Thereby it may be useful for two purposes:

★ Determining the neutrino spectrum given in arbitrary units: $\phi(E) \propto \frac{1}{\sigma_{\text{norm}}} \frac{dN(E)}{dE}$, where $dN(E)$ is the number of events in the *normalization channel* within the range $(E, E + dE)$ (it allows easy to calculate any cross section, e.g. $\frac{d\sigma_{\text{CCQE}}}{dQ^2} = \frac{\Delta N(Q^2)}{\Delta Q^2 \int \phi(E) dE}$).



New normalization channel

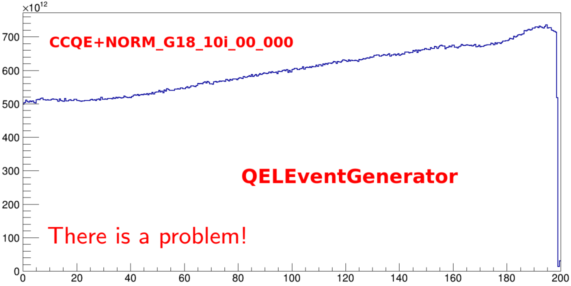
★ Testing the quality of event generators. For this, one should generate events using *GEVGDriver* under test and *NormGenerator* with uniform spectrum, because *NormGenerator* always succeed in generation of events $\nu + A \to \nu + A$ (leaving the target intact). If $q(E)$ is the ratio of failures to generate event at given energy E by *GEVGDriver* under test then the probability of event generation by *NormGenerator* and *GEVGDriver* should be recalculated according to formulae

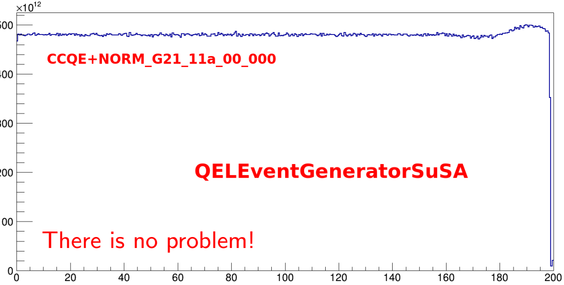


$$p'_{\text{norm}} = \frac{p_{\text{norm}}}{1 - p_{\text{norm}} q(E)} > p_{\text{norm}} \quad \text{and} \quad p'_{\text{test}} = \frac{p_{\text{test}}(1 - q(E))}{1 - p_{\text{test}} q(E)} < p_{\text{test}}.$$

Clearly $\sigma'_{\text{norm}} = p'_{\text{norm}} \sigma_{\text{tot}} > \sigma_{\text{norm}}$.

New normalization channel





Edge issue

If a user wants to generate events in some energy range using an alternative configuration, it will be natural to precalculate splines in that range. However, in this case he/she will meet the problem at the right end of the range because of the following piece of the code in the method *void GMCJDriver::BootstrapXSecSplineSummation*:

```
// decide the energy range for the sum spline - extend the spline a little
// bit above the maximum beam energy (but below the maximum valid energy)
// to avoid the evaluation of the cubic spline around the viscinity of
// knots with zero y values (although the GENIE Spline object handles it)
double dE = fEmax/10.;
double min = rE.min;
double max = (fEmax+dE < rE.max) ?  fEmax+dE : rE.max;
// in the logaritmic binning is important to have a narrow binning to
// describe better the glashow resonance peak.
evgdriver->CreateXSecSumSpline(fXSecSplineNbins,min,max,true);
```

If the user calculates the spline till 200 GeV then max will be equal to 220 GeV. Since there is no precalculated values in the range 200–220 GeV the method *void GEVGDriver::CreateXSecSumSpline* sets them equal to zero, which leads to wobble of the spline close to the right end.