Working Group 1.1: Neutrino Generators and How Models are Added

Laura Fields, Yoshinari Hayato, Gabe Perdue, Walter Giele

ECT Workshop on Testing and Improving Models of Neutrino Nucleus Interactions in Generators

June 2019

Charge

1. **Organizing event generator work**

    1. **Neutrino generators and how models are added**

        1. **What it is: lead a discussion of how we could restructure generators to be contributed parts, rather than the current "all inclusive" model. Are there intermediate steps toward this vision?**

        2. **Work goals:**

            - **Emerge from the meeting with some vision of whether or not our current generators could lead to, or accommodate this path.**

            - **How would development of this path be practically realized?**

Disclaimer

- I am a pretty **odd choice for leader** of this working group
  - I do not work on any event generator
  - I don't think I've ever personally checked out and run an event generator
- But there is some method to the madness
  - I am a **Co-Spokesperson of MINERvA**, so spend a lot of time thinking about how best to make measurements that will be used to tune generators
  - At Fermilab, I **'supervise' the Fermilab GENIE team**:


J Yarba


R Hatcher


S. Gardiner


W. Giele


G. Perdue

Disclaimer

- So why am I here
  - Because the organizers invited the **head of Fermilab Scientific Computing Division (SCD), Jim Amundson**, but he could not be here, so he sent me instead
- A message from Jim
  - From this workshop, he would like to see **a statement from the community on what we need** from generators
  - If we are able to do that, **he will find a way to make sure those needs are meet**
    - If significantly more computing or people resources are needed, he will fight for those resources
    - But he needs a coherent statement from us in order to do that
  - Currently, Fermilab is supporting GENIE because the Fermilab neutrino community has voted with its feet by using GENIE for almost all generator needs
    - Going forward, **we will continue to support what the community needs**, whether that is GENIE or other efforts
  - He recognizes that there are likely to be **differences of opinion** on what is needed
    - But some point, he has to ask for funding for *something*
- The rest of the slides are my words (not Jim's or MINERvA's or anyone else I might represent), attempting to **collect opinions of the rest of the working group** — Gabe, Walter, and Hayato-san.

Coordination With Theory API Framework

- This working group topic has **significant overlap with Topic 1.5**, led by Steven Gardiner (who does have experience developing neutrino event generators!)
- **Charge of that group**:
  - Design of an universal interface for theorists to provide models.

    1. **What it is: can we define an universal interface for all the possible model generators such that the integration with different generators is simple ?**
    2. **Work goals:**
       - Analyse the requirements from the different generators and define an universal minimal interface.
       - Interface design: how can we package theory models (in Fortran, C++?, Python?) such that code may be used with minimal adaptation on the generator side.
       - How do we work with multiple different generator architectures and methods for factoring the physics computations?

- My working group and Steven's **will meet jointly**
  - I will concentrate on summarizing the "**organization**" aspects of the discussion
    - The first parallel session (**Tuesday afternoon**) will focus on this
  - Steven will concentrate on the details of the **theory API**
    - The **Wednesday parallel sessions** will focus on the Theory API topic

Why Are We Here

- We are having this conversation because the neutrino world is **several "Pain Point" related to neutrino event generators** (© Gabe Perdue; most of the words on this slide are his):
  - The **time between model development and incorporation into experimental simulations** is quite long
  - It is **difficult for theorists to engage**, and they are generally underserved in terms of citations when they do
  - We **duplicate effort** in re-implementing models many time
  - **Release schedules are slow, and lots of bugs** are unearthed once code gets to large users, leading to lots of headaches in neutrino software and missed production deadlines
  - In general the current structure of generator collaborations puts **bottlenecks on model development** and makes it hard to leverage a large community (not because the generator authors are bad people, but because they need to be deeply involved in putting models into the generators and there are only so many generator folks)
  - Some of **our best physics models are very hard to use** because they don't have geometry and flux support (and it is hard to ask developers to add that sort of stuff - that isn't their expertise)
- Restructuring a generator (or generators) might be the right answer to these points, or **there may be other ways to address them**

All-Inclusive Model

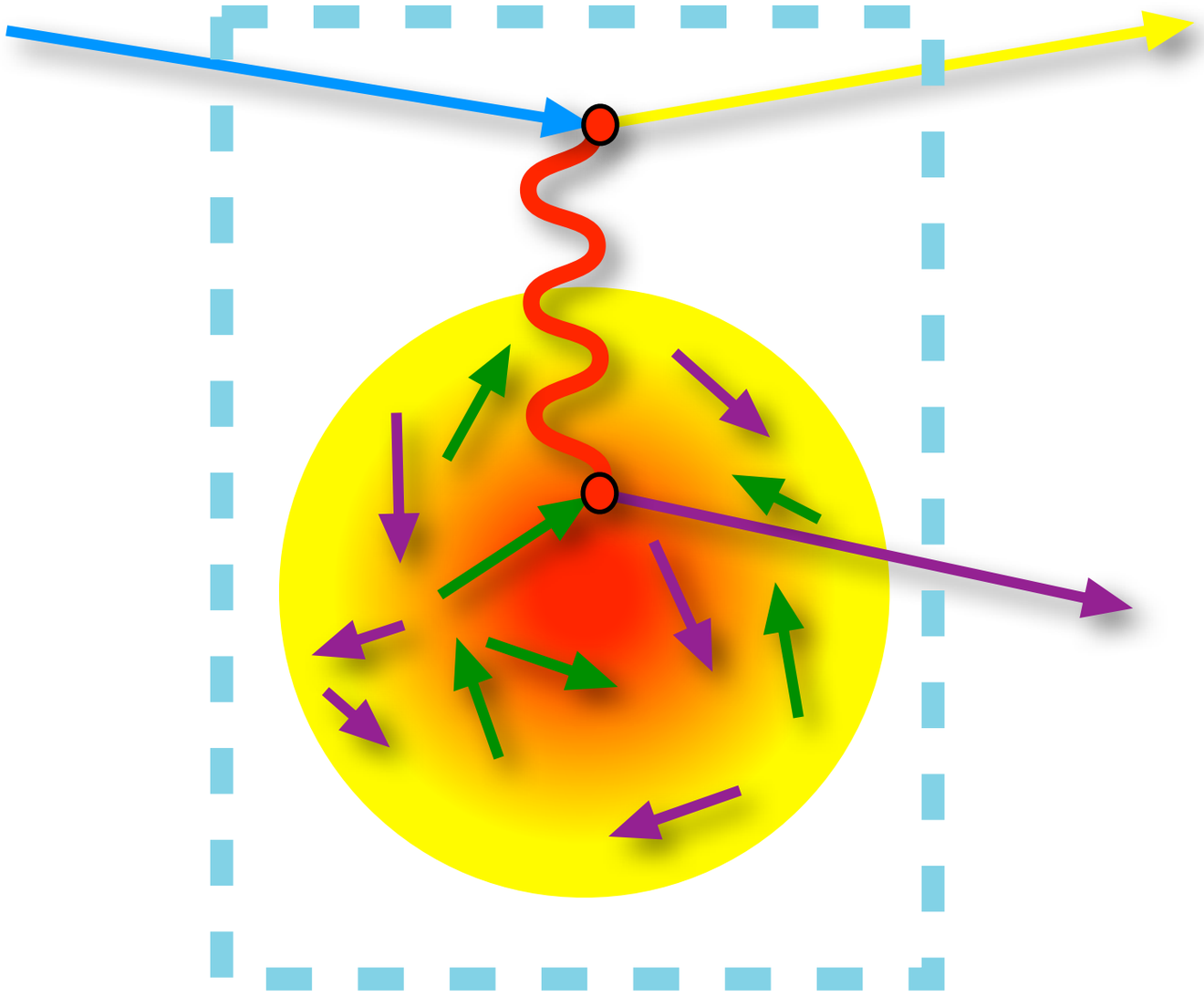
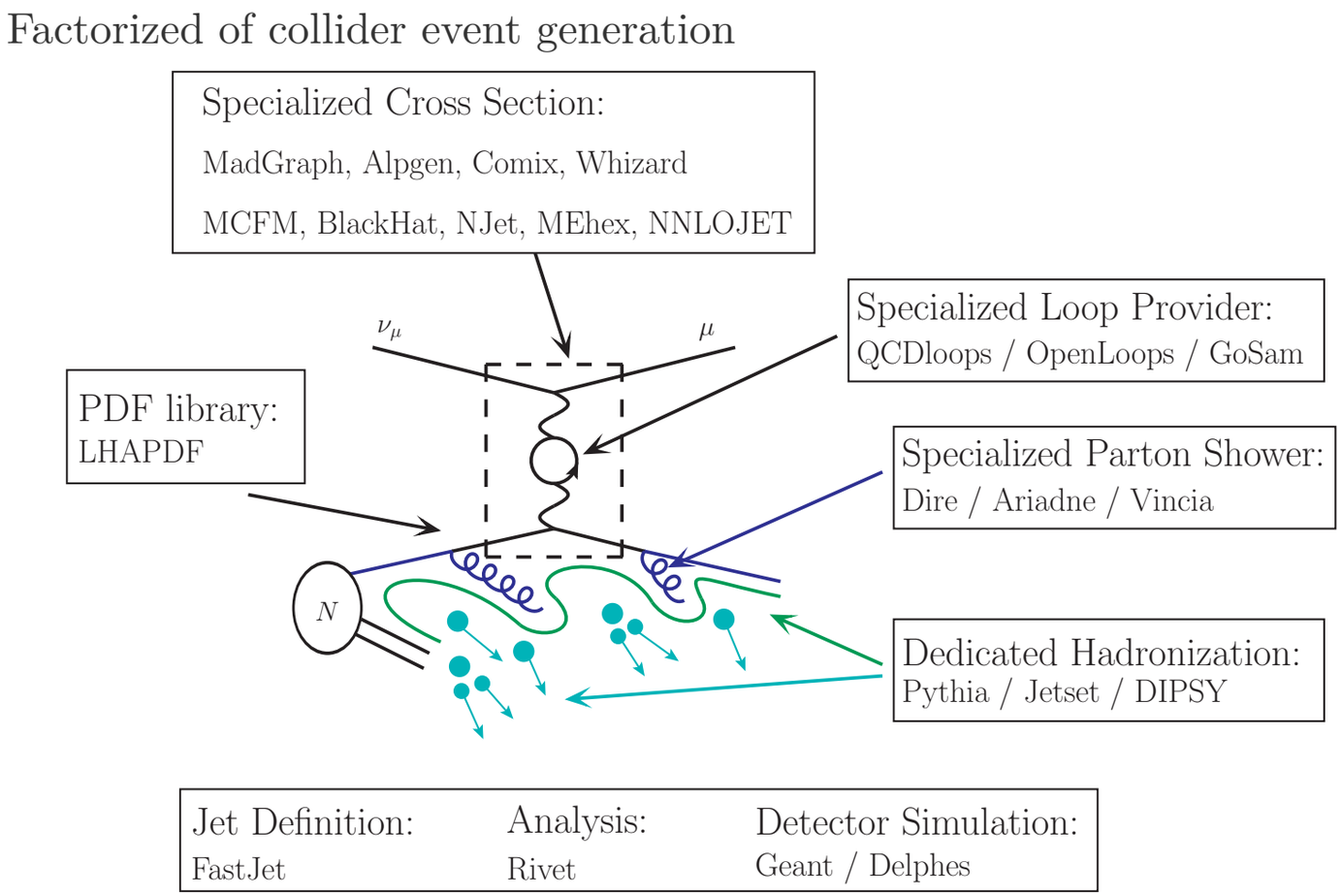


Fig Courtesy Gabe Perdue

- Today's generators use an "**all-inclusive**" model
- Users **pass in flux + geometry** (or just nucleus), **and get back final state particles** + some record of what happened in the middle
- The code to simulate everything in between is **developed and maintained by the generator authors**

The Collider Model



More factorized – i.e. more codes contributing – where it matters most (precision hard scattering)

4 / 17

Fig Courtesy Stefan Prestel

- Our **collider colleagues do things much differently**
- The code that does various steps of the **simulation is factorized** and maintained by different groups
- Talking to (other) Pythia authors, I **did not get the impression that this works perfectly**, but it does seem to work better than our model

If Neutrino Generators Used the Collider Model

- This working group is charged with identifying **whether neutrinos should move to some model like this**:

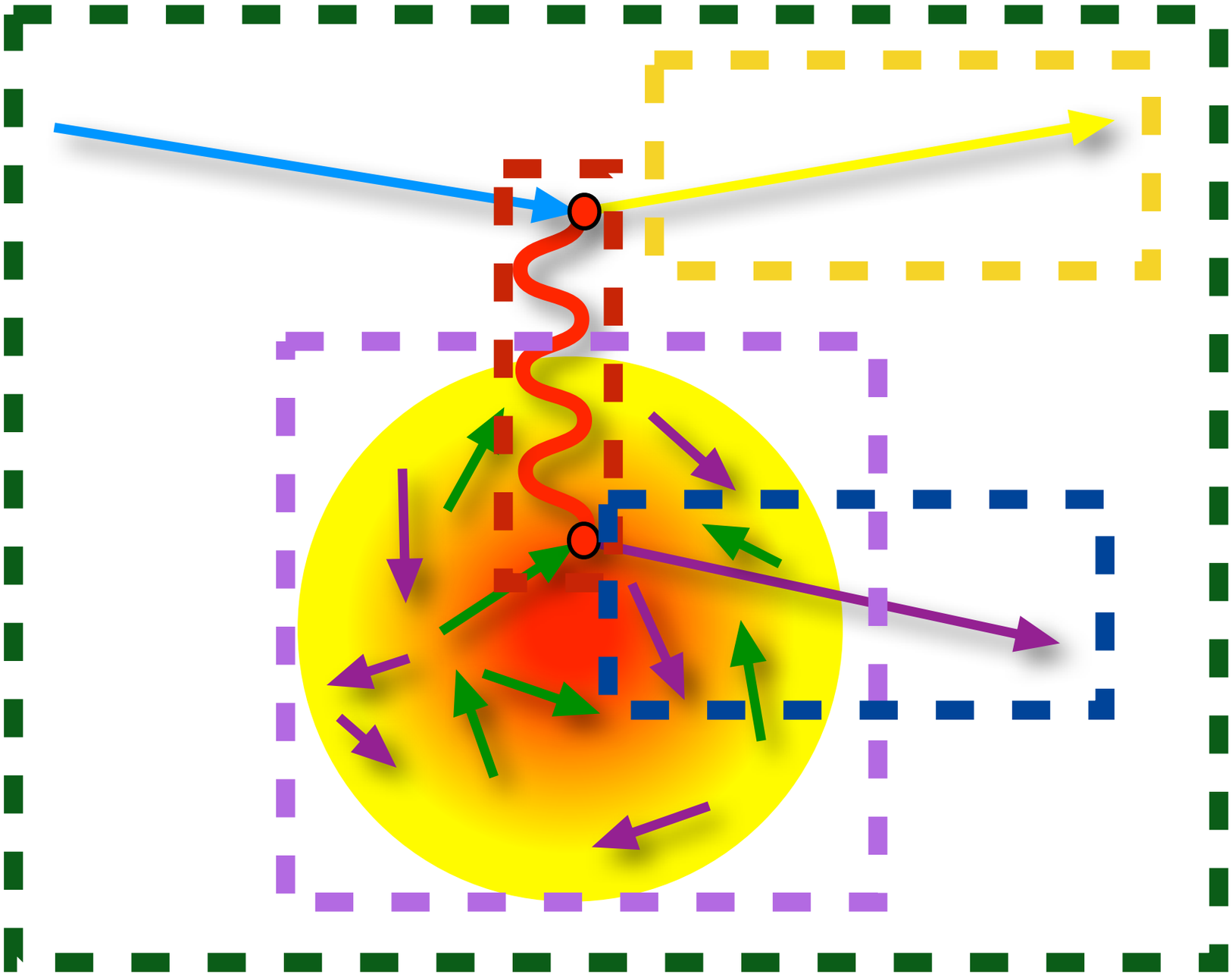


Fig Courtesy Gabe Perdue

- And **if so, how should we do it**?
- In this model, **generators** do not necessarily provide all components, but **become APIs** that define inputs/ outputs and interfaces to the different pieces

Should We Move to a Factorized Model?

- Advantages
  - Experiments get access to a **wider array of models** for neutrino analysis
    - **Time between development** of models and use by experiments shrinks
  - **Efficiency**: less time is spent re-writing the same model in multiple language, flux drivers that do the same thing, etc
  - Various **participants can concentrate on their area of expertise** — theorists do not have to figure out how to write flux drivers, generator groups do not have to translate 1000's of lines of Fortran, etc
  - **Experts on the models can be responsible for maintaining them**
  - **Citations** — generators and experiments cite theory papers and visa versa

Should We Move to a Factorized Model?

- Disadvantages
  - A major reason *not* to do it is **inertia**
    - However we go about it, this transition would be a lot of effort, and the neutrino community has a lot of things to do; we have to **choose our battles wisely**
    - If we want to go this route, it **will require major pressure from experiments** to overcome this inertia (at least this is what got it done for the LHC)
  - Also: instead of a core group of people funded for the task, **maintenance of the various pieces is spread to many people**
    - This is both good and bad; it spreads the load, but will also relies on a lot of people doing what they need to do
  - With less of the pieces controlled/maintained by the generators, **we risk less support for generators themselves** — and we need more support for generators, not less

How Would We Get Started?

- If we want to do this, what are **possible first steps**?
  - Following the LHC models, we would have a lot of **workshops to define a common event model**
    - Walter G reports that this was a lot of work, but the output was event format(s) that all theory groups are capable of reading/writing, and that both theorists and experimentalists are very happy with
      - It was not set in stone (the event format that agreed on in the ~90s was Fortran common blocks!)
    - This workshop is a good start, but largely I'm saying the same thing that Gabe said here a year ago, and **more dedicated effort is going to be required** to actually move this forward
- There are **baby steps** we can take at this workshop and before a dedicated workshop:
  - **Are there generators that are open to evolving** towards this model?
  - Identify the **basic list of components** that the generator would be factorized into
    - Can presumably further separate later, but we have to start somewhere
  - Fleshing out of **details of Theory API** (See talk from Steven G)
  - Another possible first step: a **stand-alone flux/geometry driver** that can be used with any generator
  - A **common output file format**, so that experiments can easily read in events from many generators
  - A **white paper** that lays out all of this

A Completely Different Approach

- If we decide we want a factorized neutrino event generator, there is **an alternative** to evolving an neutrino event generator into something factorized (or building something entirely new)
- It's possible that a factorized event generator could be evolved into a neutrino event generator
  - I understand that **Pythia** collaborators are attempting to simulate neutrino-nucleon DIS scattering as a sort feasibility study of this option
- But there is **a lot that needs to be added** to make Pythia an effective neutrino event generator
  - I think we basically have to take a "**wait-and-see**" approach to this option

Workshop Goals

- **Questions to be discussed** in the Working Group Parallel Session:
  - **What problems are we trying to solve** exactly?
    - Are there more pain points beyond what I listed in an earlier slide?
  - What are the **advantages and disadvantages** of a factorized generator model that I have not mentioned?
  - Is factorizing generators something **we should invest a lot of effort in**?
    - Do we write this **new generator** from the ground up, **or evolve** one of the existing generators?
    - If so, are there generators are open to becoming ‘factorized’?
    - Is a n**ew collaboration** or other organizational structure necessary to get this done?
      - What should that organization structure look like?
  - Can we develop a **first draft of ‘factors’** that would be needed?
    - What are the basic inputs and outputs of each of these?
  - What are the **next steps after this workshop**?