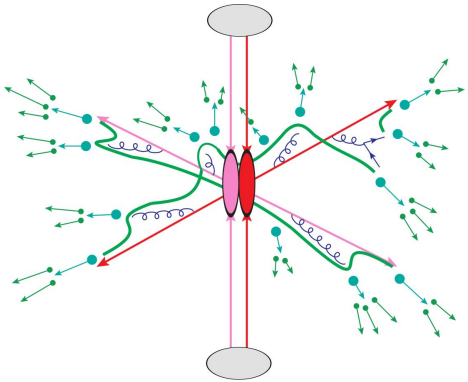
HepMC (and associated formats & tools)

Andy Buckley
University of Glasgow
for the HepMC3 team

ECFA Higgs Factory Generator Workshop
21 June 2023

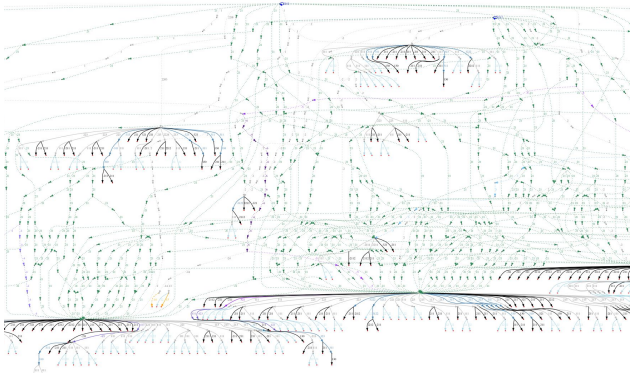
HepMC context

- **MC generation: where theory meets experiment**
  - The fundamental collision, *in vacuo.* Modelling split into:
  - matrix element: few particles, complex differential distribution, inefficient & expensive (at high orders, with current methods)
  - parton shower + hadronisation: approx, factorised completion of ME, plus empirical modelling of strong-coupling regime + decays
- **Generators**
  - MC codes also split between ME and shower+hadronisation (SHG) generators, e.g. MG5_aMC, PoWHEG, WHiZard (+ loops) vs Pythia, Herwig and Sherpa (actually, Sherpa does both)
  - Loosen technical coupling via intermediate file formats: LHE (and HDF5 successor) at ME level, HepMC at hadron level.
- **HepMC**
  - HepMC is as much an in-memory interface & data structure as a file format: designed for general representation of particle production, interactions, and decays
  - Remember: none of this is "true"! Just a record of the algs



HepMC history and content

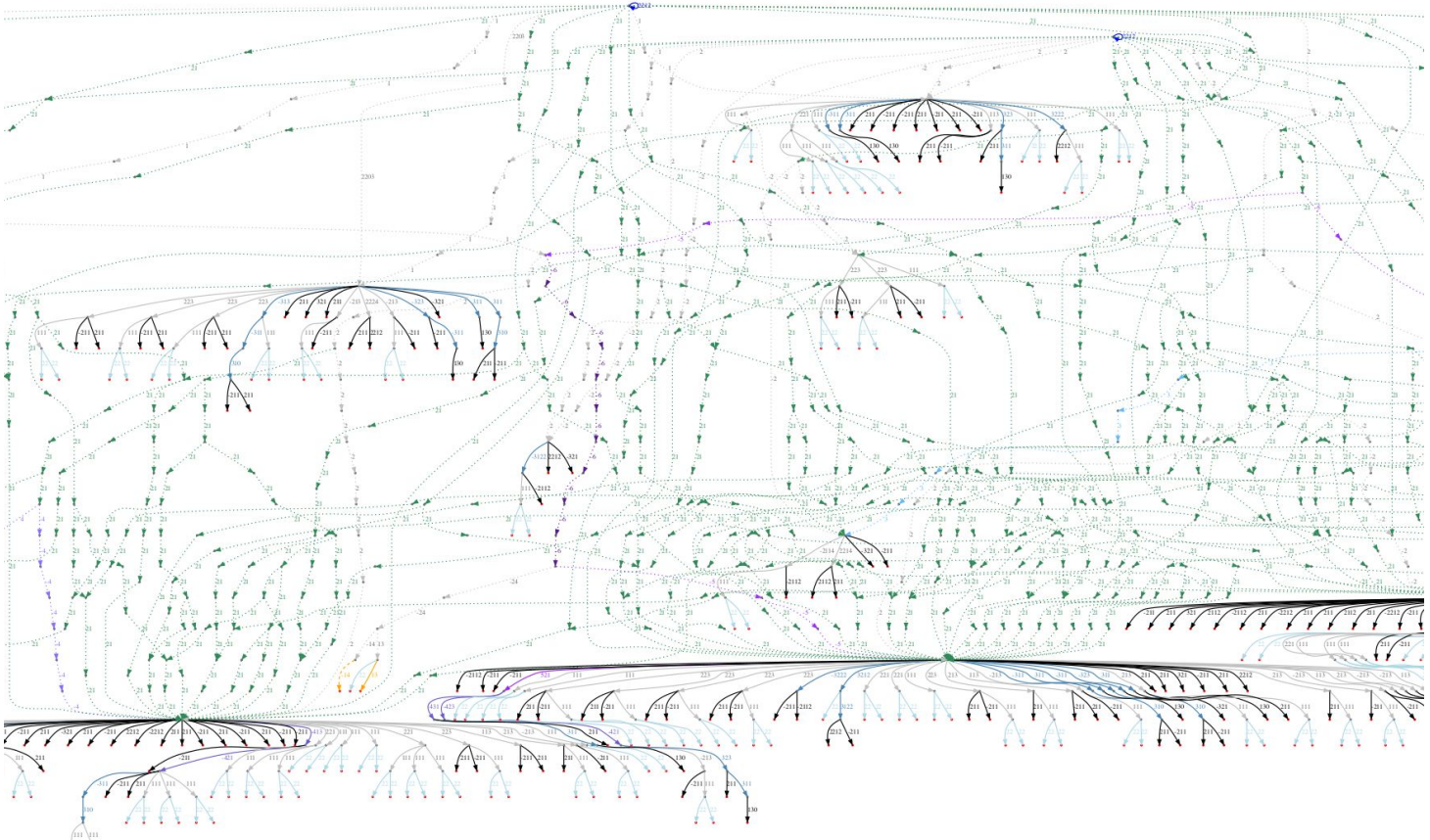
- **History**
  - General graph model, connecting GenParticles via GenVertices
  - v1 in 2001: raw pointers, lack of standardisation for values
  - v2 in 2008: resolve expt forks and CLHEP depn, add units
  - v3 in ~2019: smart ptrs, resolve constness inconsistencies, clarify ownerships, make memory contiguous → ROOT-friendly



- **Standardisation**
  - v2 introduced standardisation of particle status codes (and later vertex statuses, but these are far less used)
  - More recent MCnet SHG consensus on standard multiweights
    - Compatibility in LHE feed-ins from ME gens would help

- **HepMC v3 new features**
  - Can embed LHE records: completeness
  - Removal of unused attrs e.g. color flow, couplings, *barcode*
    - Latter always declared unphysical: now use ID + statuses
    - EDM4HEP MC defn includes some “historical” quantities
  - Arbitrary attributes (not super-efficient, but general) + GenRun

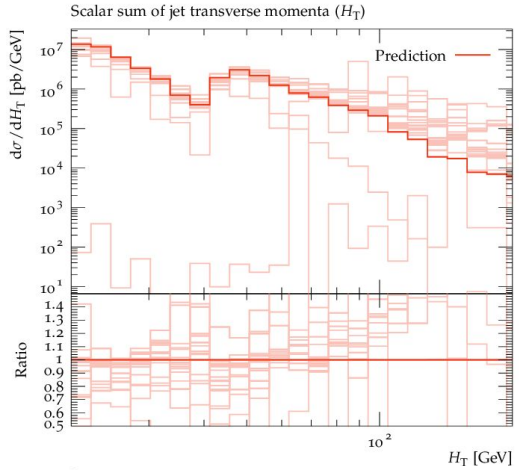
HepMC example (⅓ of a Py8 *pp*→ttbar event)

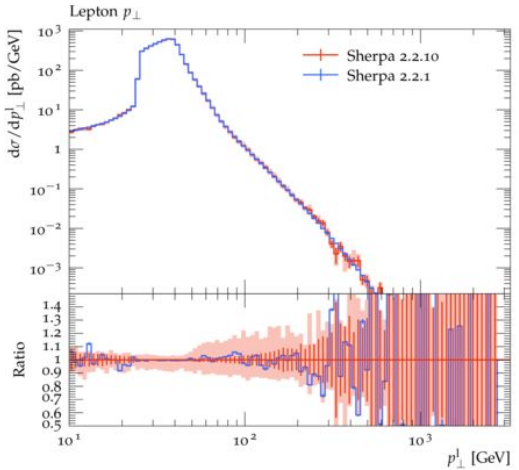


HepMC open issues

- **Multiweights**
  - Event multiweights are hugely important in modern MC: everywhere a random choice is made, can multiply weights by class to represent event probability under other assumptions
  - Highly efficient way to propagate MC theory uncertainties, under certain assumptions: only one event stream to sim+reco

- **Variation cross-sections**
  - Post-hoc propagate weight corrections to xsec manually: nominal x ratio of event-weight sums. Xsec errs are harder
  - →HepMC3 can store multiple xsecs per event.
  - Not trivial how to convert HepMC2 event files: keep one weight, multiply up to the number of weight streams? Two passes?!?
  - And what to do about AUX weights that can be unphysical and don't have a corresponding xsec variation?
  - Halfway solution in 3.2.6 release; not wholly satisfying… yet





HepMC associated tools & developments

- **MCUtils**
  - MCUtils header library supplies tools for event manipulation, searching, testing, filtering, etc. Plus PID decoding.
  - Written for HepMC2, recently updated to HepMC3 by Siddha Hill & Andrii Verbyitskyi. Other improvements underway, e.g. new PileMC pile-up multiplexer started by Kalp Shah (cf. FCC-hh)
  - Several ideas since added to HepMC3 API. Possibly to be brought into the main lib, cf. the “search engine”

- **Changing MC data-flows**
  - Refactoring discussion (this week, Les Houches SM) about interchange formats to allow mix & match MEs (via in-memory representation?) and parton showers (HepMC API and HDF5 format) etc.
  - HDF5 format: some prelim work done; similar to HDF5 for partonic events (in place of LHE), allows better HPC leverage
  - Need to remain flexible: technologies and constraints change!

HepMC summary & outlook

- **Status**
  - Modern C++ implementation of event-record graphs; LHC & MC gens standard
  - Efficient in memory if used well; LHC collaborations switched/switching from HepMC2 *(barcode reliance an issue)*
  - ROOT persistency now very efficient; other binary formats also supported
  - Growths into heavy-ion and neutrino communities
- **Technical evolution**
  - Parallel-I/O, data-sci/HPC standard HDF5 to come (intended new standard formats at parton and particle levels)
  - Supporting event-manipulation tools and standards in development, may be incorporated in future
- **Better work-patterns?**
  - Important: event graph is mainly calculation, not physics history!
    History is only reliably physical after hadronisation
  - Idea to reduce SHG events to physical minimum (+ debug mode). Top quarks, EW bosons, other EW resonances with reasonably narrow width to be kept?
    Extent to strip resonance recoils? ⇒ large memory & tree-walk CPU savings