Progress on EvtGen and Pythia developments

EvtGen

Fernando Abudinén, John Back, Michal Kreps, Thomas Latham

Pythia

Ulrik Egede, Tom Hadavizadeh, Philip Ilten, Minni Singla, Peter Skands

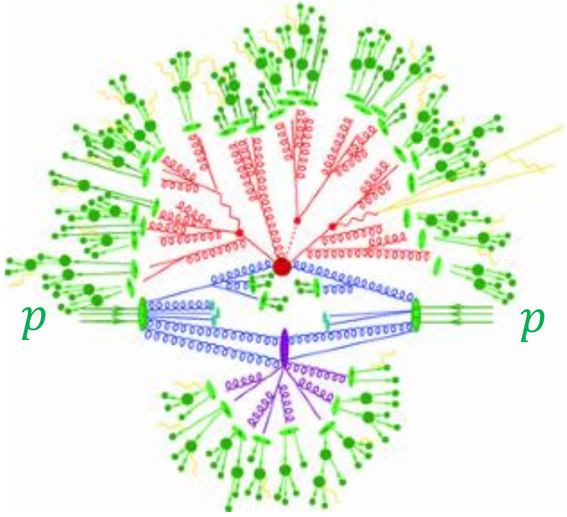
MONASH University

WARWICK
THE UNIVERSITY OF WARWICK

SwiftHep workshop
March 27, 2024

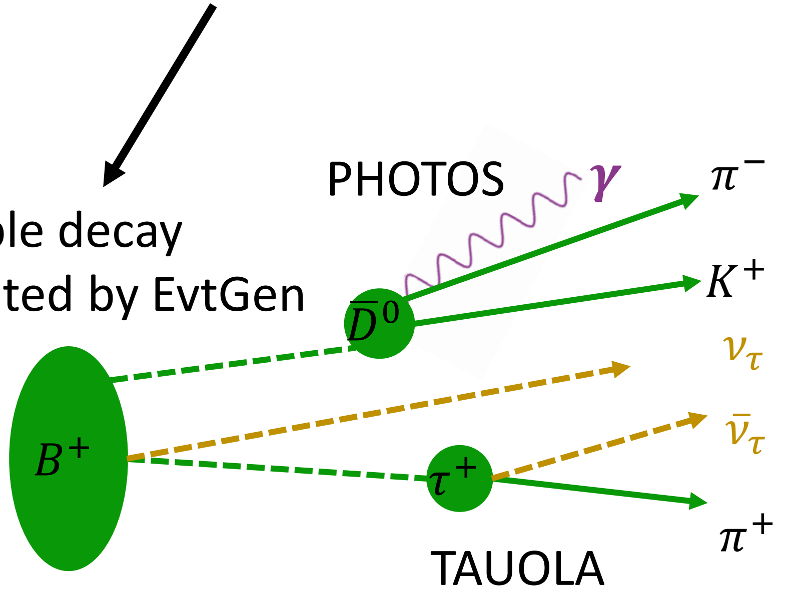
Recap on EvtGen

Example collision simulated by Pythia8



- EvtGen: generator package specialised for heavy-flavour hadron decays
  - Used as well inside simulation of b jets
- Contains about 130 decay models implementing specific dynamics of various decays
- Maintains detailed decay table with large number of explicit decays
  - Known decay branching fractions do not add up to 100%; Remainder is filled up by generating quark configurations and passing those to Pythia8 for fragmentation
  - Fraction of decays passed to Pythia8 depends on particle (b-baryons rely more on Pythia8 than others)
- τ decays simulated using TAUOLA
- Final-state radiation (FSR) simulated using PHOTOS

Example decay simulated by EvtGen



Status and plans

- Developed in the 90's, stable over past 10 years (changes mostly additions of new models)
- Physics wise no plan for changes in near future
- Major goal $\Rightarrow$ enable thread safety

Recent developments

- Work on code modernisation, clean-up, removal of duplications, and documentation
- Implemented global testing framework for validation
- Fixing of broken models
- First core adaptation towards thread safety (full adaptation intended for the future)
- Studies of alternatives for τ simulation (to be continued)
- Studies of alternatives for FSR simulation

Today's focus

Challenges for multithreading in EvtGen

- Internal: structural limitations for multithreading inside EvtGen
  - Global instance of random number generator
  - Global instance of particle properties and decay table

$\Rightarrow$ Needed structural changes identified and first combination of solutions found

- External: limitations from dependences
  - TAUOLA
  - PHOTOS

$\Rightarrow$ Overcoming limitations from dependences are more challenging as they are external
  - TAUOLA and PHOTOS authors currently exploring ways to enable thread safety
  - Exploring use of Pythia8 as alternative to TAUOLA
  - Exploring use of Vincia QED (Pythia8) shower and Sherpa's PHOTONS++ as alternative to PHOTOS

Progress on thread-safety

With help of research-software engineers:
Heather Ratcliffe, Chris Brady

Set of solutions to reach thread-safety (preliminary):

- Converted **static** objects to **static const** where possible
- Global singleton objects made thread-local
- Serialized (mutexed) calls to PHOTOS and TAUOLA

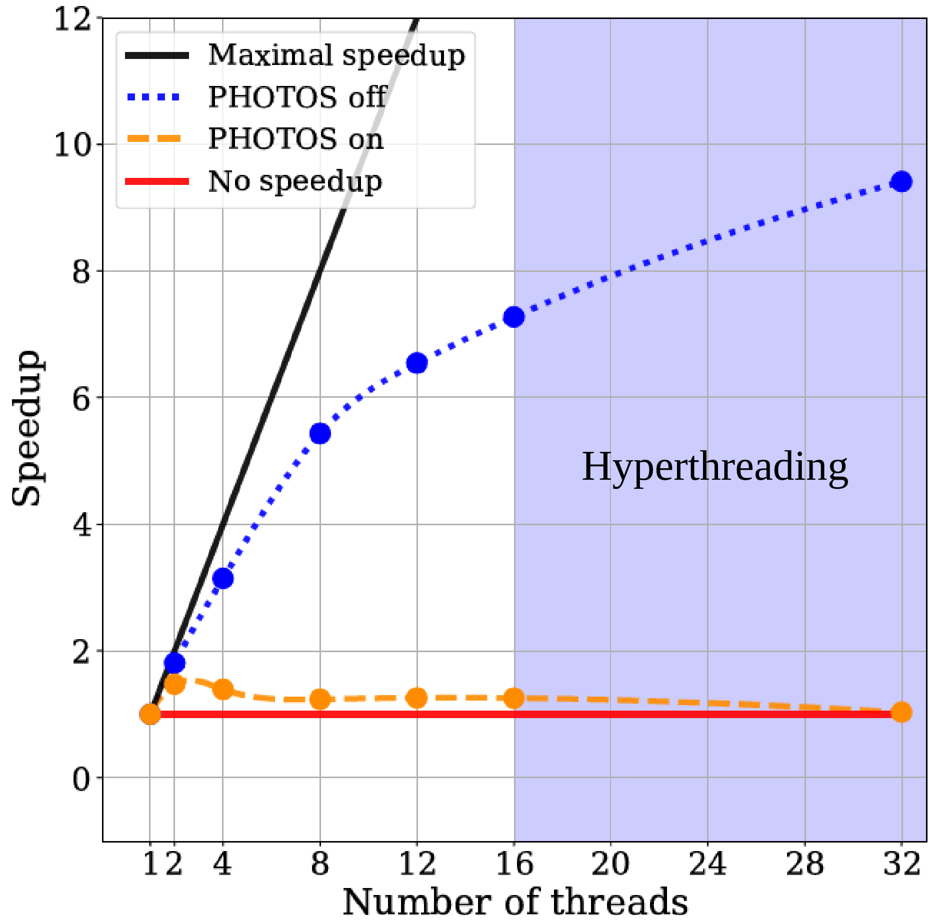
⇒ Deeper structural changes needed to fully exploit multi-threading (plan to continue working on it)

⇒ Current preliminary status reached thread-safety, passing tests for all decay models

⇒ But performance limited by external dependencies

- PHOTOS is used for almost every decay

⇒ Study alternatives for final-state radiation (FSR)



Final-state radiation generators

- Treat the effect of FSR as a correction to the Born-level decay rate (or cross section)

$$\mathrm{d}\Gamma^{\text{radiative}} = \mathrm{d}\Gamma^{\text{Born}}\, f(\Phi)\, \mathrm{d}\Phi$$

Φ: Phase-space of photons

- Example (oversimplified): neutral scalar → e^+e^- (single QED dipole)

From talk by Peter Skands

LO QED $\propto$

$$\frac{2s_{e^-e^+}}{s_{e^-\gamma}s_{\gamma e^+}} + \frac{1}{M_0^2}\left(\frac{s_{\gamma e^+}}{s_{e^-\gamma}} + \frac{s_{e^-\gamma}}{s_{\gamma e^+}} + 2\right)$$

eikonal term — collinear terms

$s_{ij} = p_i \cdot p_j$
Dot product of 4-momenta

PYTHIA

DGLAP

e^--collinear limit, e^+-collinear limit:

$$\frac{P_{e^-\to e^-\gamma}(z_1)}{s_{1\gamma}} + \frac{P_{e^+\to e^+\gamma}(z_2)}{s_{2\gamma}}$$

VINCIA

Antenna

All Singular Terms:

$$\frac{2s_{e^-e^+}}{s_{e^-\gamma}s_{\gamma e^+}} + \frac{1}{M_0^2}\left(\frac{s_{\gamma e^+}}{s_{e^-\gamma}} + \frac{s_{e^-\gamma}}{s_{\gamma e^+}}\right)$$

HERWIG, SHERPA, PHOTOS

YFS

Soft limit:

$$\frac{2s_{e^-e^+}}{s_{e^-\gamma}s_{\gamma e^+}}$$

Final-state radiation generators

Pythia QED

- Determines “best” set of dipoles (no genuine multipole effects)
- Works as parton shower evolution interleaved with QCD, MPI, …

YFS – Yennie-Frautschi-Suura 1961 **(with modern implementations as in PHOTOS and Sherpa)**

- Takes full (multipole) soft interference effects into account
- Scalar QED (no spin dependence)
- Adds a number of photons to final state with predetermined kinematics (no interleaving)

Vincia QED Kleiss-Verheyen 2017**,** Brooks-Verheyen-Skands 2020

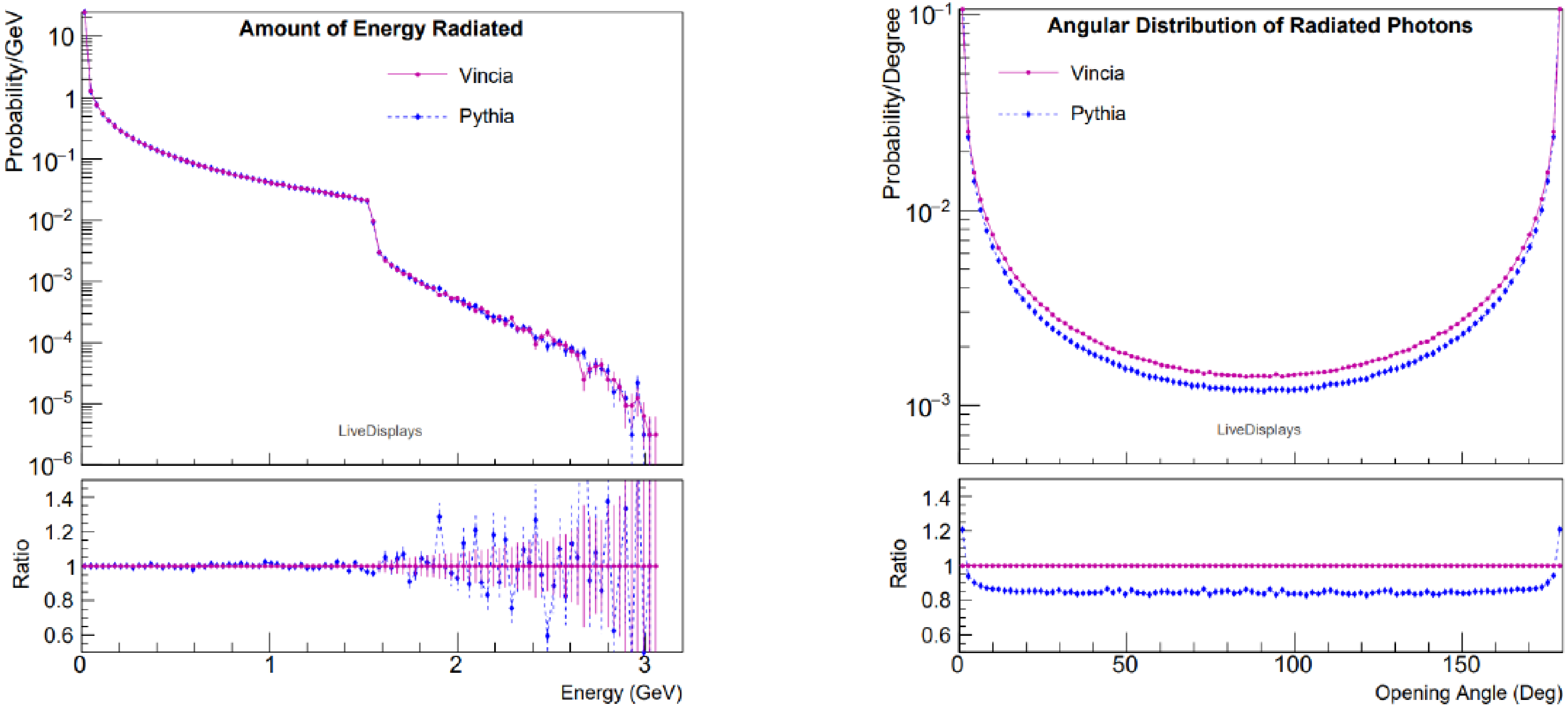
- Takes full (multipole) soft interference effects into account
- Not limited to scalar QED (includes spin dependence)
- Works as parton shower evolution based on antenna approximation (can be interleaved)

Vincia QED shower for FSR

Giacomo Morgante's Honours Thesis

- Explore use of Vincia's multipole QED shower and compare with Pythia and PHOTOS

$$J/\psi \rightarrow e^+e^-$$

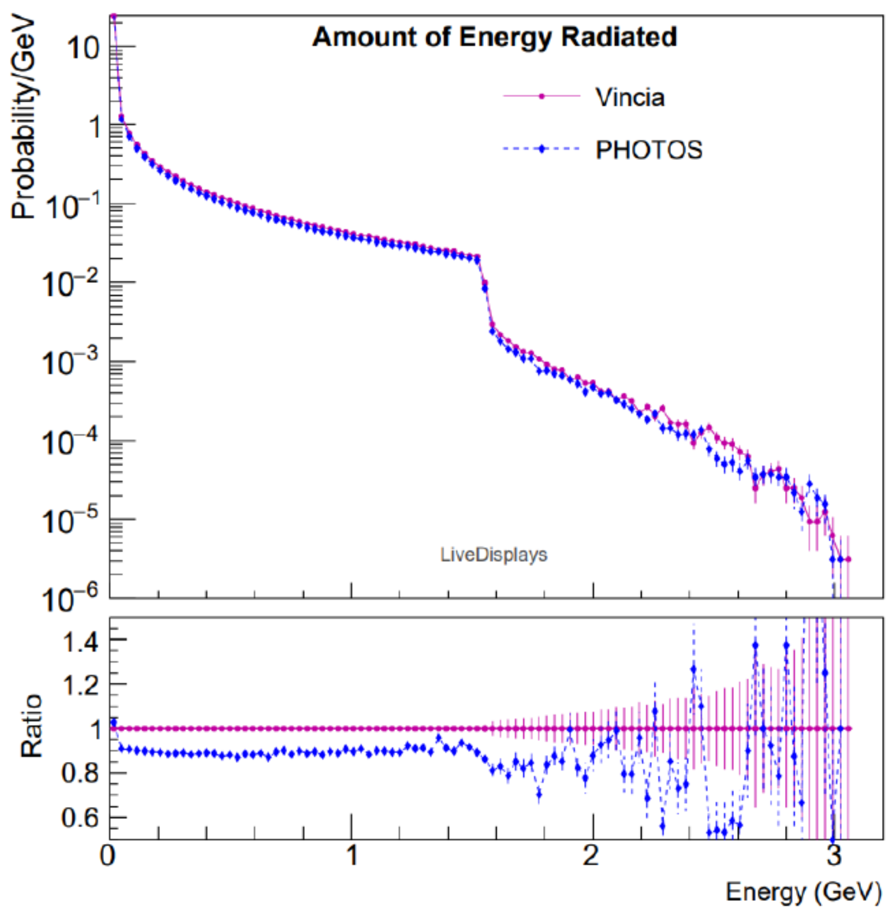


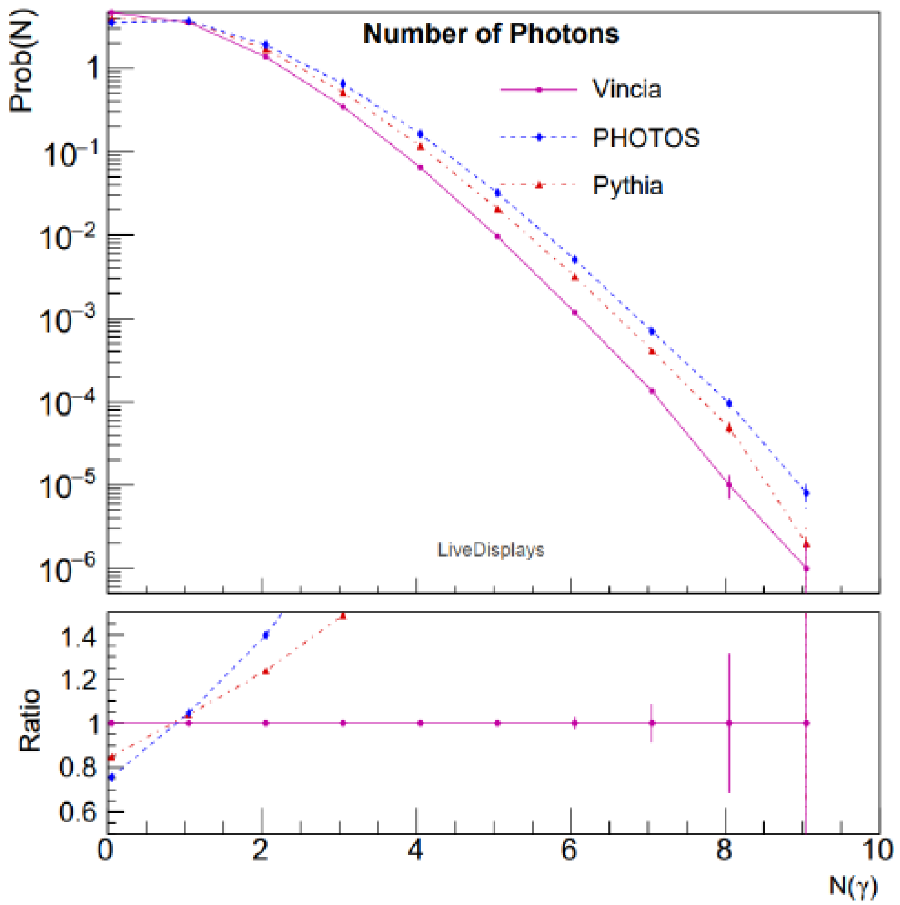
$\Rightarrow$ Good agreement for radiated energy

$\Rightarrow$ Larger impact of collinear singularities on angular distribution in Pythia

Vincia QED shower for FSR

$$J/\psi \rightarrow e^{+}e^{-}$$





⇒ Vincia tends to radiate fewer but harder photons than PHOTOS
⇒ Difference around 10% for modes with electrons

Vincia QED shower for FSR

- Recently adapted to radiate off hadrons (previously supporting only leptons)
- Matrix-element corrections not implemented yet

$\Rightarrow$ Potential main cause for some discrepancies with other generators

$\Rightarrow$ A lot of room for improvement and validation

$\Rightarrow$ However, preliminary results look promising

Technical aspects

- Vincia is embedded in Pythia8
- Algorithm implementation enables thread safety
- Developed EvtGen $\leftrightarrow$ Vincia interface based on existing dependency with Pythia8

$\Rightarrow$ Ready to be used, but needs latest Vincia developments (not released yet)

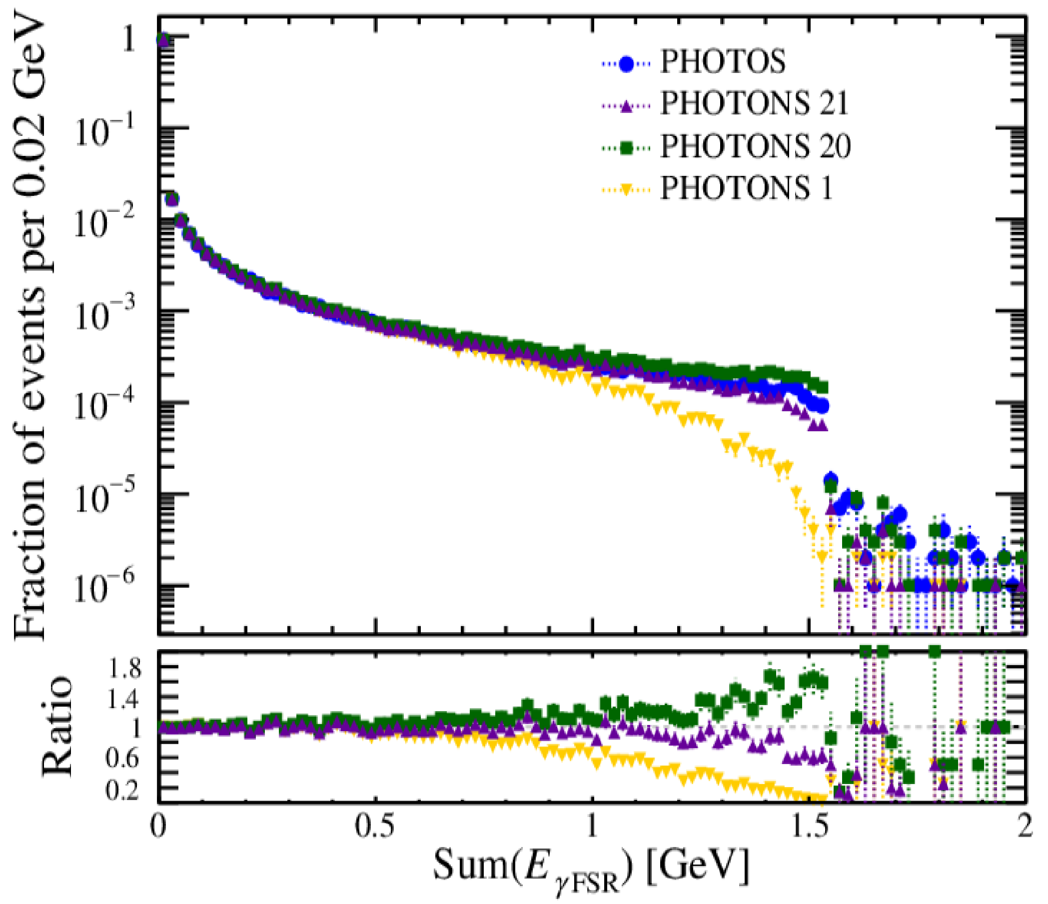
Sherpa's PHOTONS++ for FSR

- PHOTONS++ in Sherpa can simulate emission of soft photons based on YFS approximation (mode 1)
- If switched on also hard photons based on collinear approximation (mode 2), with
  - Approx. matrix-element corrections (mode 20) or
  - Exact matrix-element corrections (mode 21)
- Using option 1, observed fewer hard photons with respect to PHOTOS (note that PHOTOS has matrix-element corrections implemented)
- Generally good agreement with PHOTOS using options 20 and 21

⇒ Will enable user to switch between options for systematic studies

Amount of radiated energy

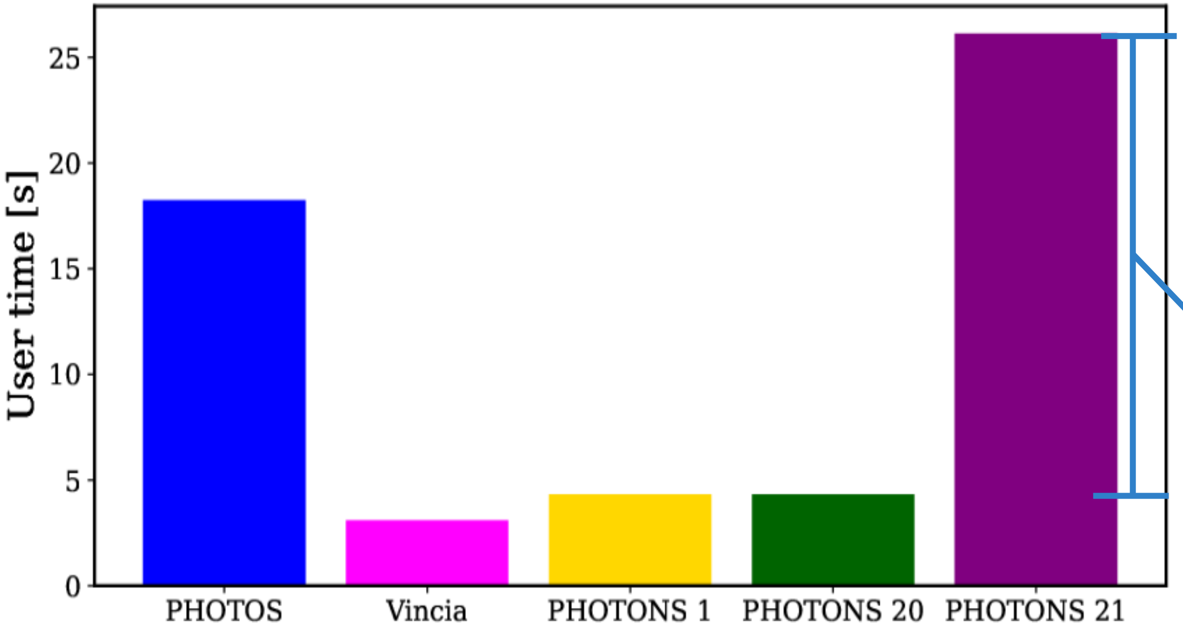
$J/\psi \to \mu^+\mu^-$

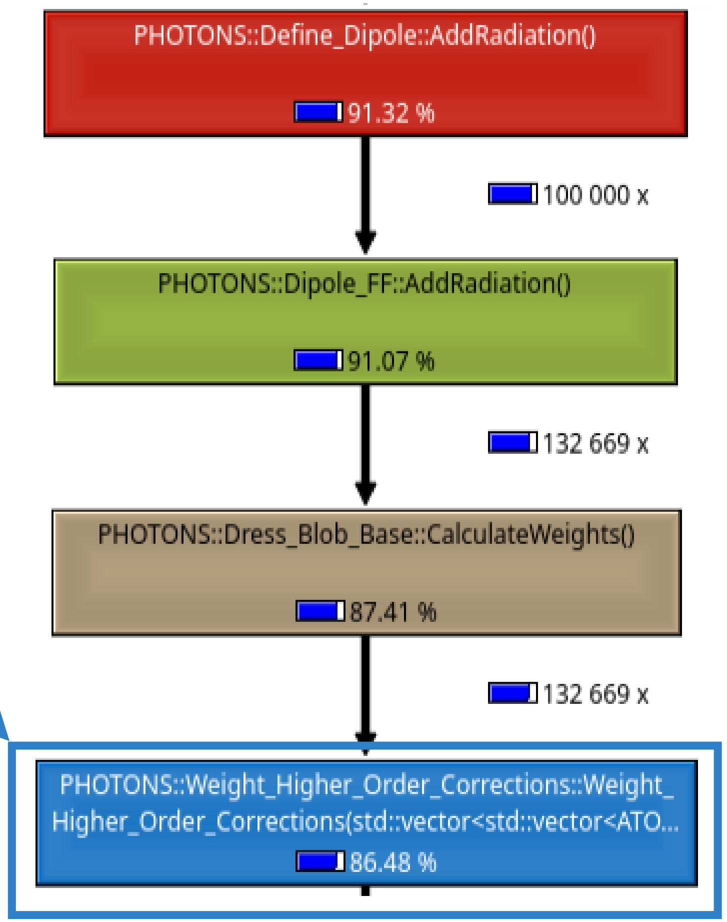


A word on timing

- Compare simulation time using $J/\psi \rightarrow e^+e^-$ decay as benchmark

$\Rightarrow$ Collinear singularities enhanced due to small electron mass





$\Rightarrow$ Largest consumption by exact matrix-element calculation

$\Rightarrow$ Good precision/time trade-off for option 20 (will use as default)

$\Rightarrow$ Potential speedup using Vincia or PHOTONS by about factor 4

Sherpa's PHOTONS++ for FSR

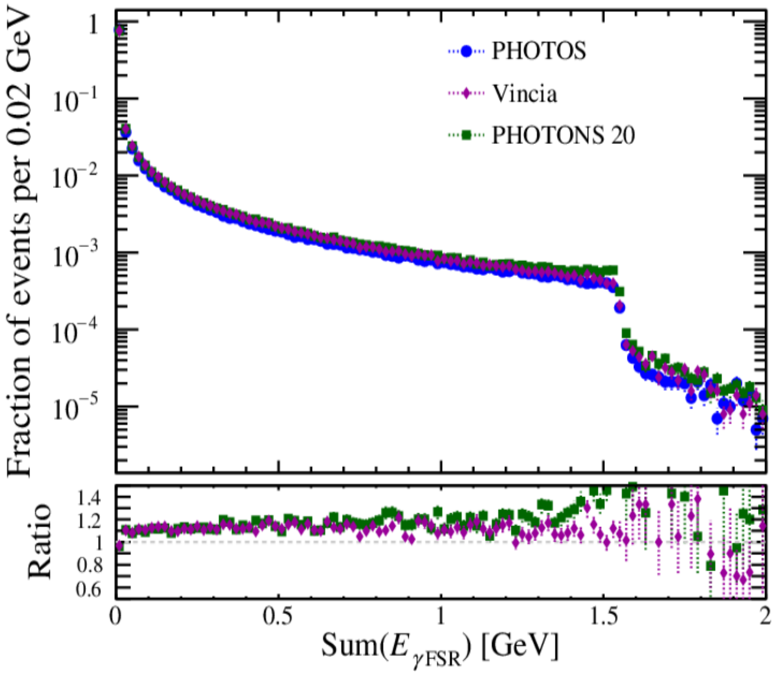
- PHOTONS++ algorithm enables thread safety
- Developed EvtGen $\leftrightarrow$ Sherpa interface (using Sherpa 2.2.15 release)
- A few aspects need to be ironed out
  - Use EvtGen's random-number generator as external generator for Sherpa (as for PHOTOS and Pythia8)
  - Initialisation of Sherpa (might be initialising unnecessary objects)
  - Should Sherpa object be thread-local?

Comparisons between FSR generators

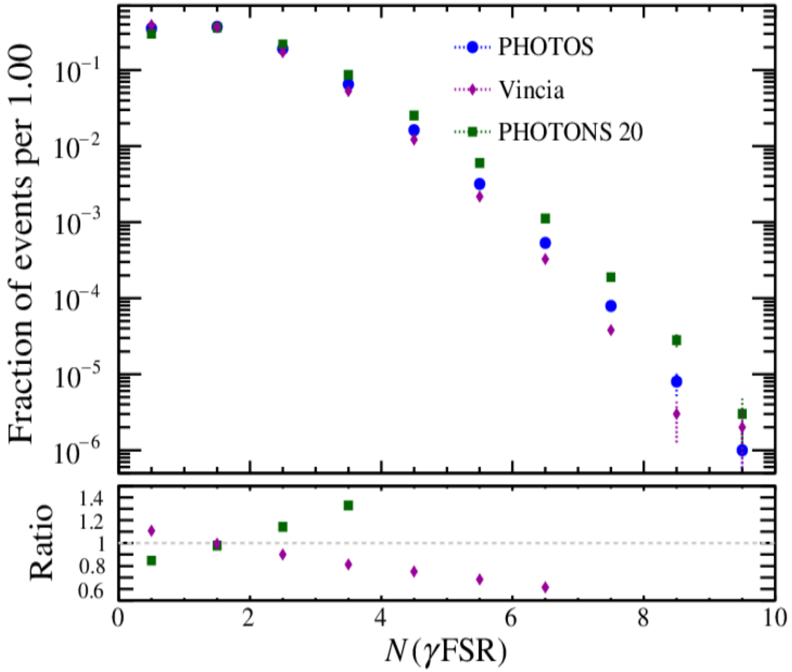
- Performed various comparisons across generators

$$J/\psi \to e^+e^-$$

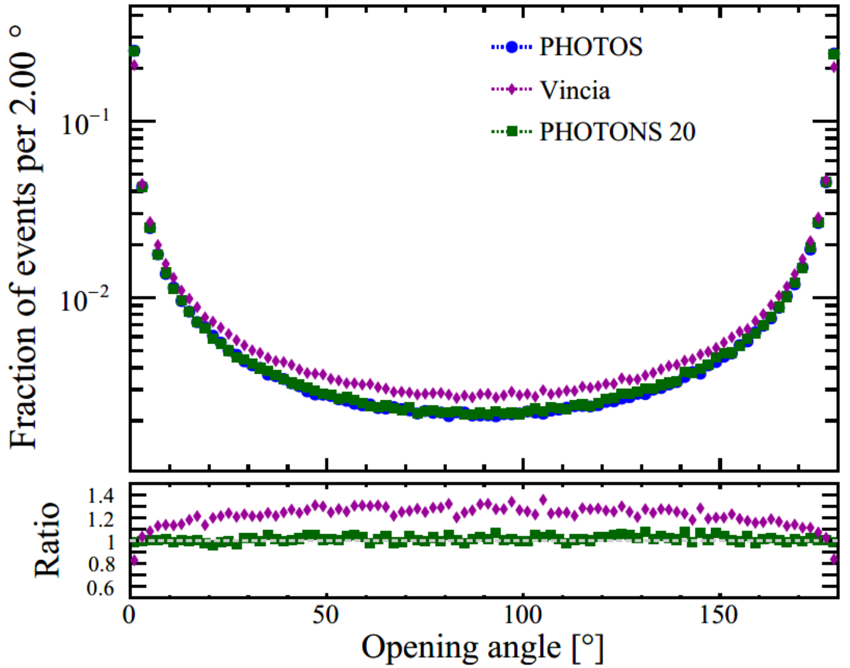
Amount of radiated energy



Number of photons



Angular distribution of photons

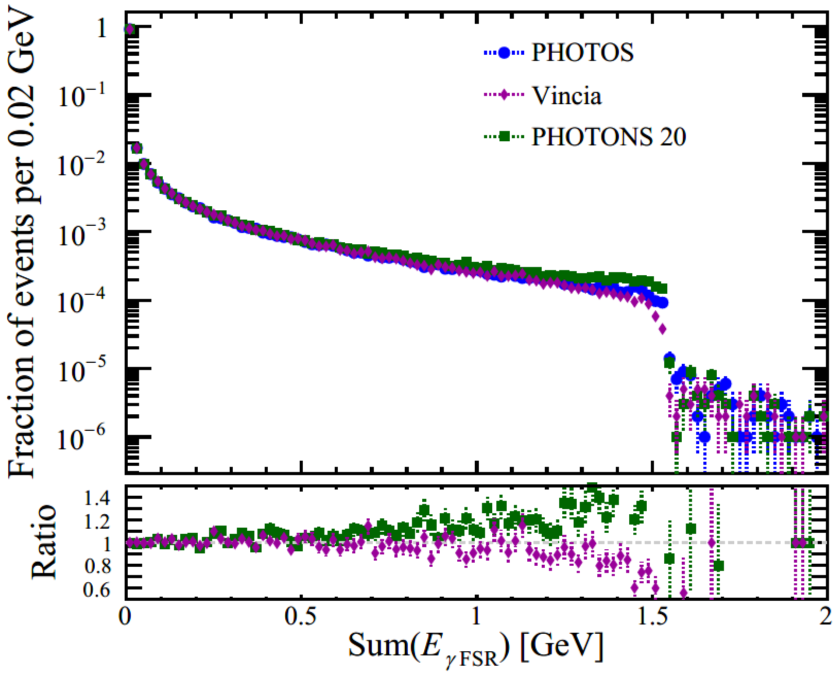


- Vincia and PHOTONS radiate more energy than PHOTOS (around 10% for modes with electrons)
- Vincia tends to radiate smaller and PHOTONS larger numbers of photons
- Collinear singularities in angular distribution less pronounced in Vincia

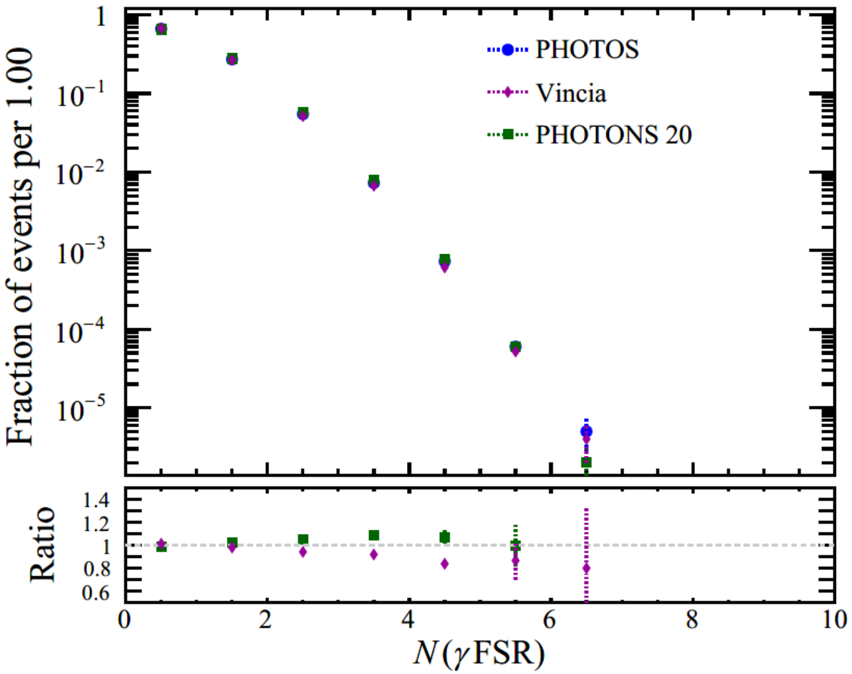
Comparisons between FSR generators

$$J/\psi \to \mu^+\mu^-$$

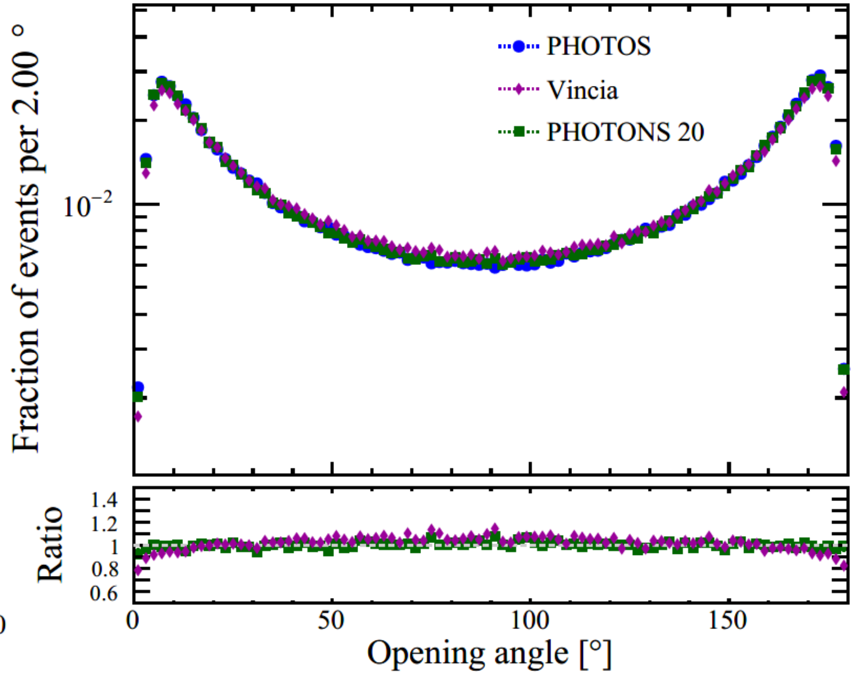
Amount of radiated energy



Number of photons



Angular distribution of photons



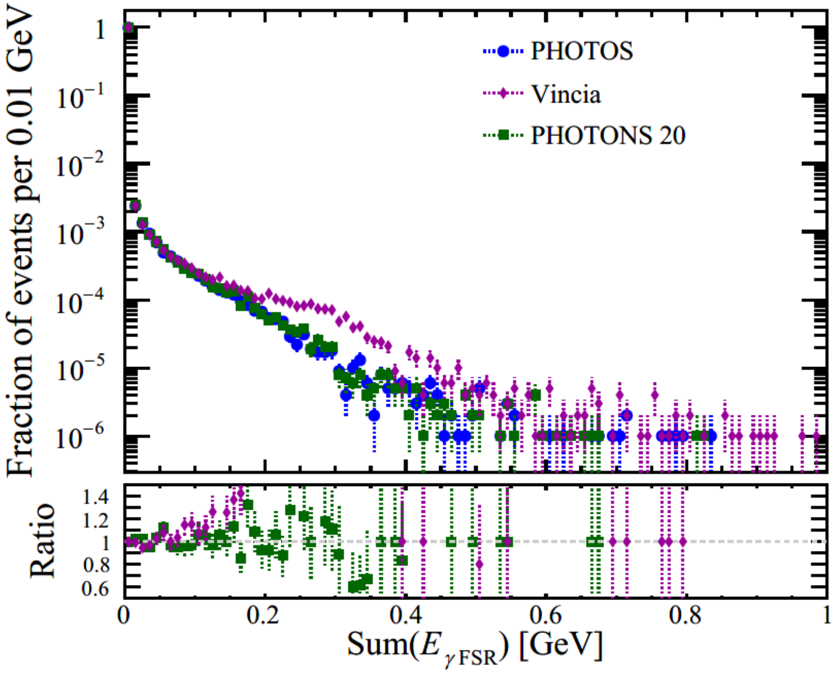
- Differences between generators become smaller for muons (for which the impact of collinear singularities is smaller than for electrons)

$\Rightarrow$ Generally good agreement between generators

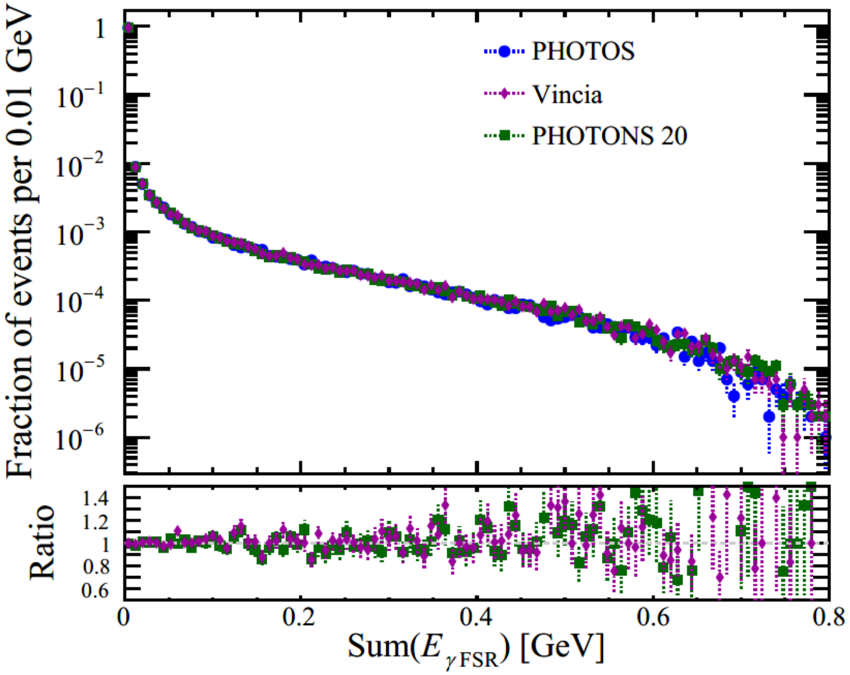
Comparisons between FSR generators

- Some more characteristic examples for the amount of radiated energy

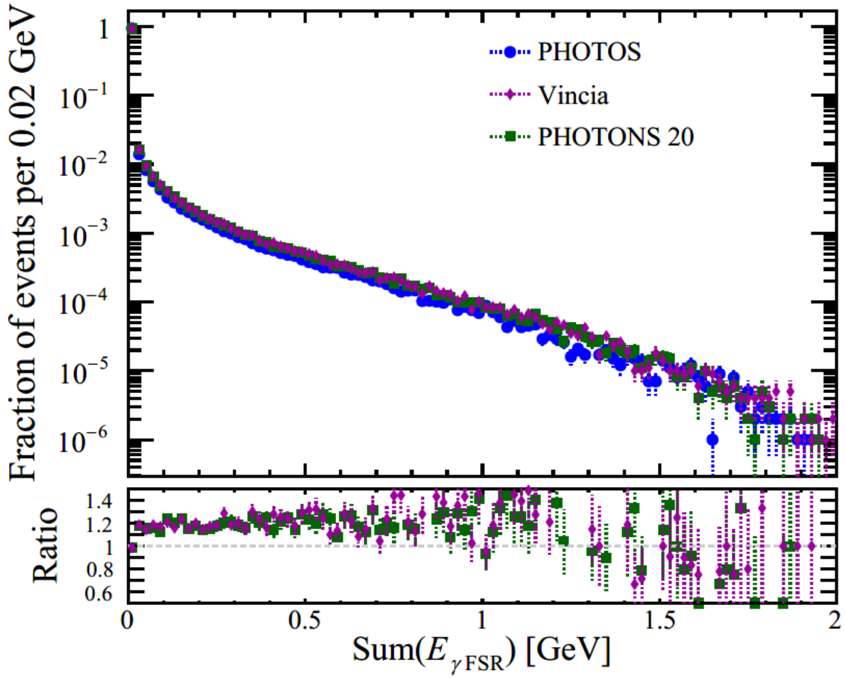
$\rho^+ \to \pi^+\pi^0$



$D^0 \to K^-\pi^+$



$B^0 \to K^+K^-\pi^+\pi^-$



- Generally good agreement between PHOTOS and PHOTONS

$\Rightarrow$ Differences in Vincia potentially due to missing matrix-element corrections

$\Rightarrow$ In some cases all generators show very good agreement

Pythia status and plans

- General purpose generator for simulation of collision events of particles (electrons, protons, photons, heavy nuclei) at high-energies.
- Contains models for several aspects: hard/soft interactions, parton distributions, initial/final-state parton showers, multiparton interactions, fragmentation and decay.

Rencently worked on different aspects aimed at making simulation of b-hadron faster:

- B enhancement
- Doubly-heavy hadrons in Pythia

} To be continued

- Optimised simulation for colour-reconnection models with junctions

New

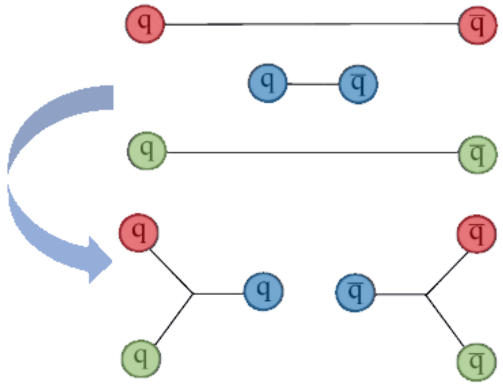
Interests for future developments:

- Implementation of matrix-element corrections in Vincia

Improving colour-reconnection models

- QCD-based colour-reconnection models with junction agree well with collision data (without particular tuning)

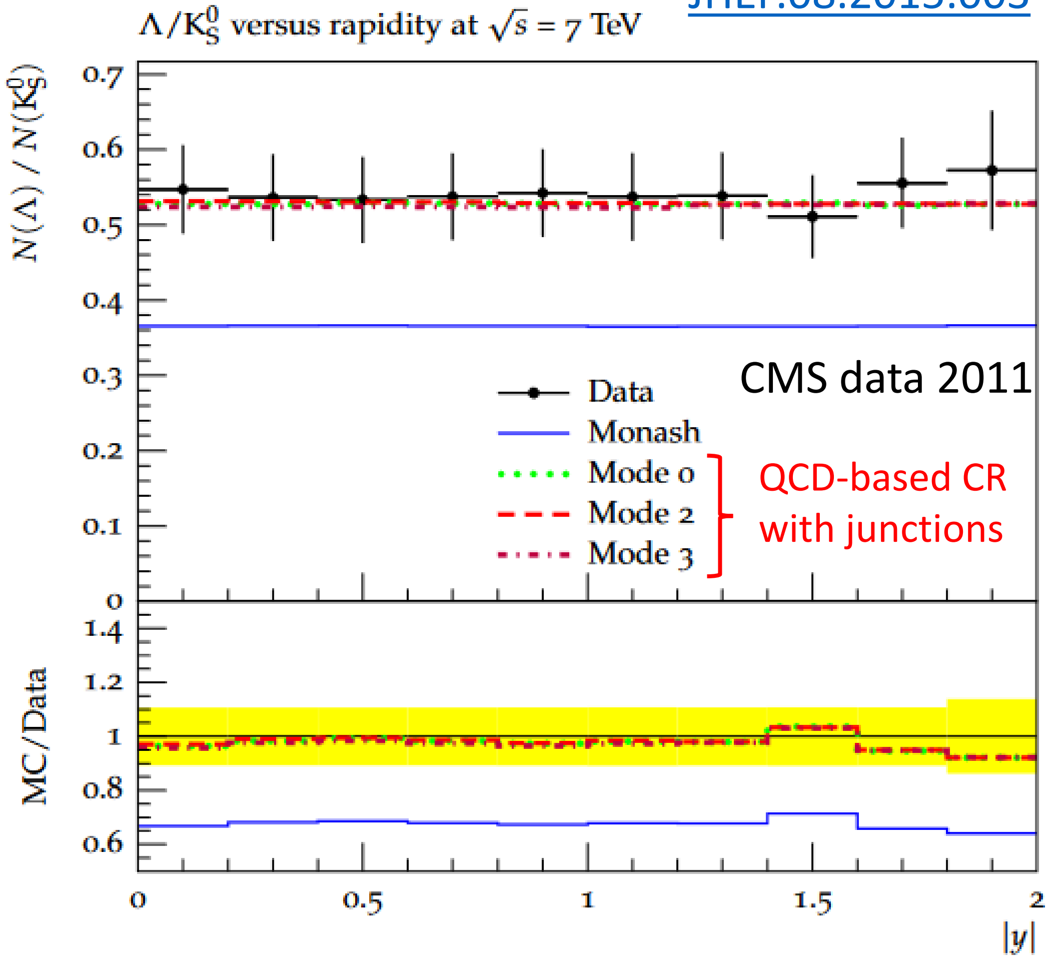
CR junction: choose "shortest" string configuration



- Initial implementation inefficient and CPU expensive
- Various issues related with coding and repeated unnecessary calls with calculations

$\Rightarrow$ Implemented set of code changes to speed up

JHEP.08.2015.003



Improving colour-reconnection models

Code modifications

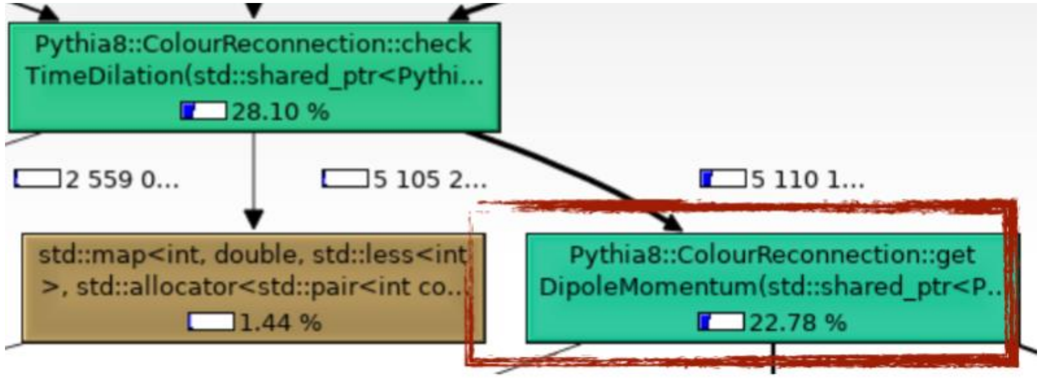
- Various functions took smart pointers as arguments
  - ⇒ Invokes copy constructor and needs locking to update reference count
  - ⇒ Use instead reference to smart pointer as argument
- Implement const correctness in various functions
- Avoid unnecessary repeated function calls, for example calculation of dipole momentum (cache whenever possible)
  - ⇒Significant decrease of CPU time from 22% to 4%
- Further improvements possible, but require major restructuring
- In total reached speedup by about factor 3

⇒ Promising for enhancement of baryon production

```
     void ColourReconnection::singleJunction(ColourDipolePtr dip1,
2.27   ColourDipolePtr dip2, ColourDipolePtr dip3) {

       // Do nothing if one of the dipoles is a junction or antijunction.
0.91   if (dip1->isJun || dip1->isAntiJun) return;
0.68   if (dip2->isJun || dip2->isAntiJun) return;
0.54   if (dip3->isJun || dip3->isAntiJun) return;


       // Check that all dipoles are active.
0.67   if (!dip1->isActive || !dip2->isActive || !dip3->isActive) return;
```



Benchmark test goes from 2m27s to 55s

Summary and outlook

EvtGen:

- Continued work towards thread safety

$\Rightarrow$ Converged on preliminary set of solutions to enable thread-safety of generator (full exploitation of multi-threading will require further structural changes)

$\Rightarrow$ Performance limited by external dependencies

- Implemented interfaces to Vincia QED and Sherpa’s PHOTONS++

$\Rightarrow$ Needs to iron out some details, but generally good agreement across generators

- τ decays: not mentioned today but plan to continue work on spin-basis conversion to use Pythia8 (interesting also for TAUOLA)

Pythia:

- Recently enabled Vincia QED shower to radiate off hadrons

$\Rightarrow$ Plan to work on implementing matrix-element corrections

- Improved implementation colour-reconnection models reaching large speed up

- Plan to continue work on enhancement of b-hadron production to speed up simulation