Forced hadronisation

Michal Kreps

LHCb-UK, Upgrade 2 meeting
Birmingham, 9-10 July 2024

Signal generation at LHCb

➡ To get as good description as possible for beauty and charm hadrons, generate minimum bias events and search for corresponding hadron

➡ Pileup adds complication as we have additional minimum bias events in the same bunch crossing

➡ Algorithm:

❖ Decide number of interaction in bunch crossing

❖ In loop generate minimum bias events and if we do not have signal yet, decay all heavier particles, check for presence of signal and check whether it passes cuts

❖ If we had signal event, we keep whole set, if we do not have signal, throw away

➡ Event generator does full hadronisation even if it gets beyond point where signal particle can be created
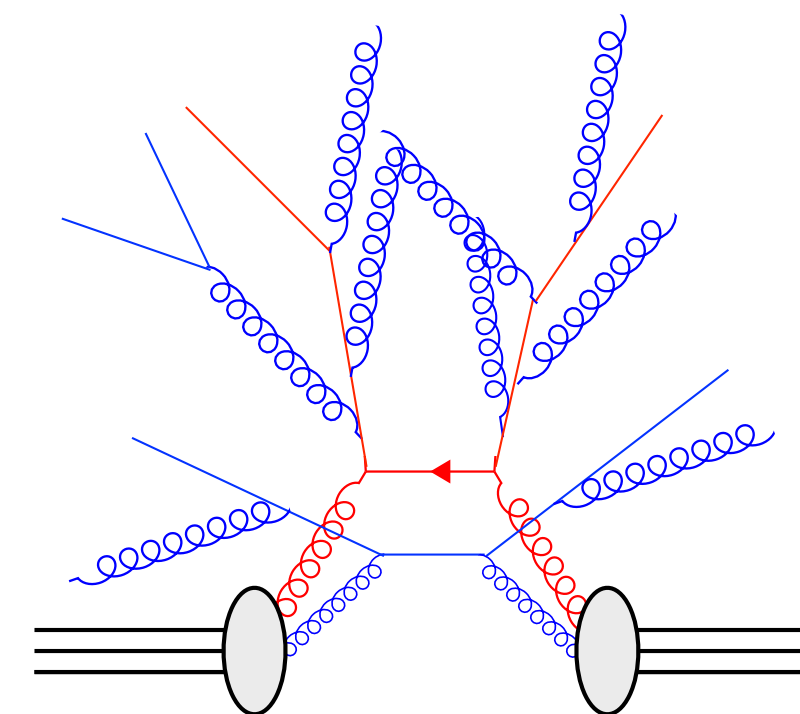
How to speed things up

➡ Event generation proceeds through several steps defined by appropriate energy scales

➡ At each step can check whether event contains required quark or there is still possibility to create one

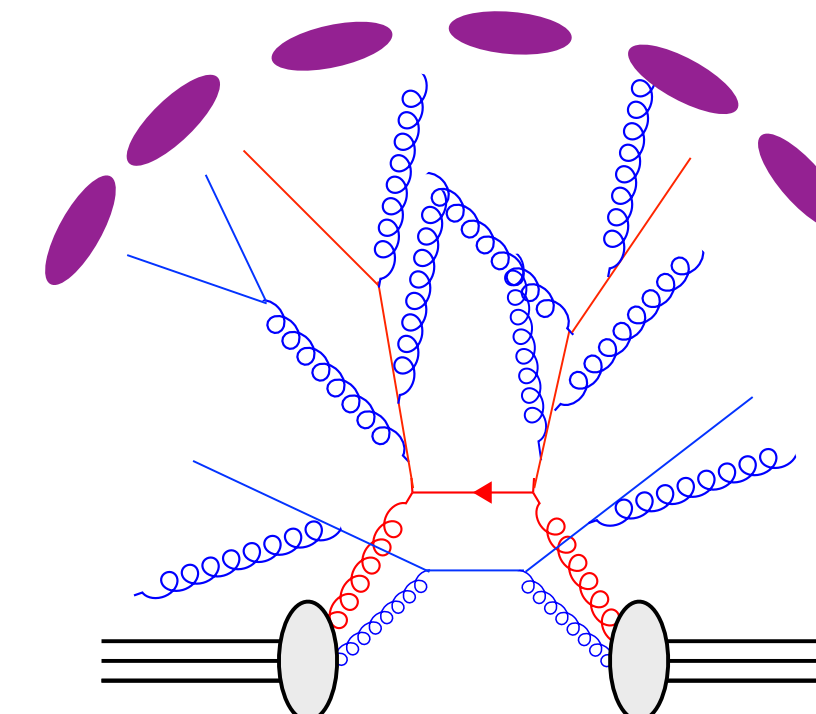➡ If there is no chance to obtain desired quark, abort process and generate event again
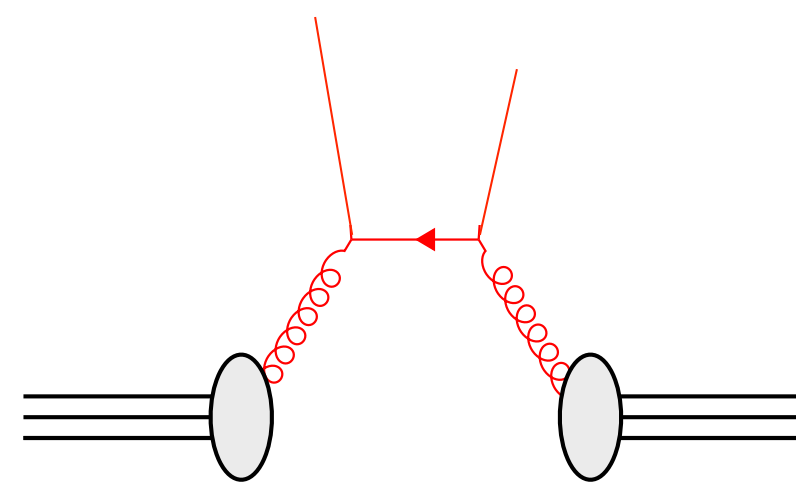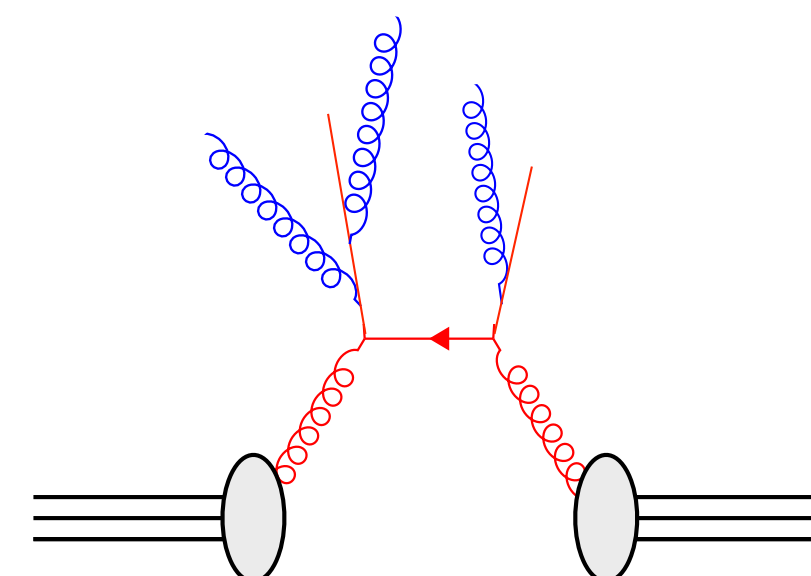
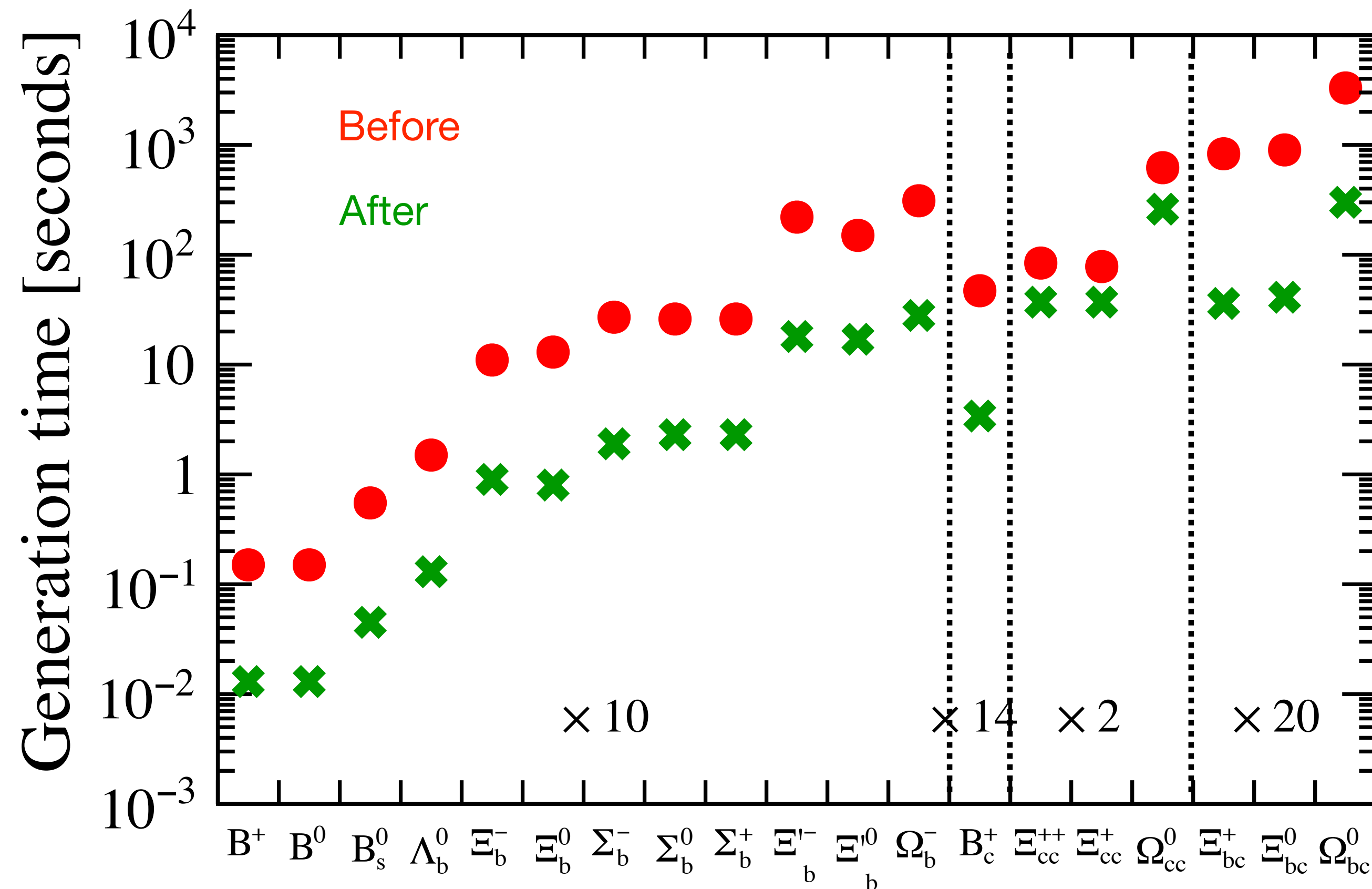(a) Process Level. (b) During evolution. (c) Parton Level. (d) Hadronisation.

Egede, Hadavizadeh, Singla, Skands, Vesterinen: Eur. Phys. J. C 82, 773 (2022)
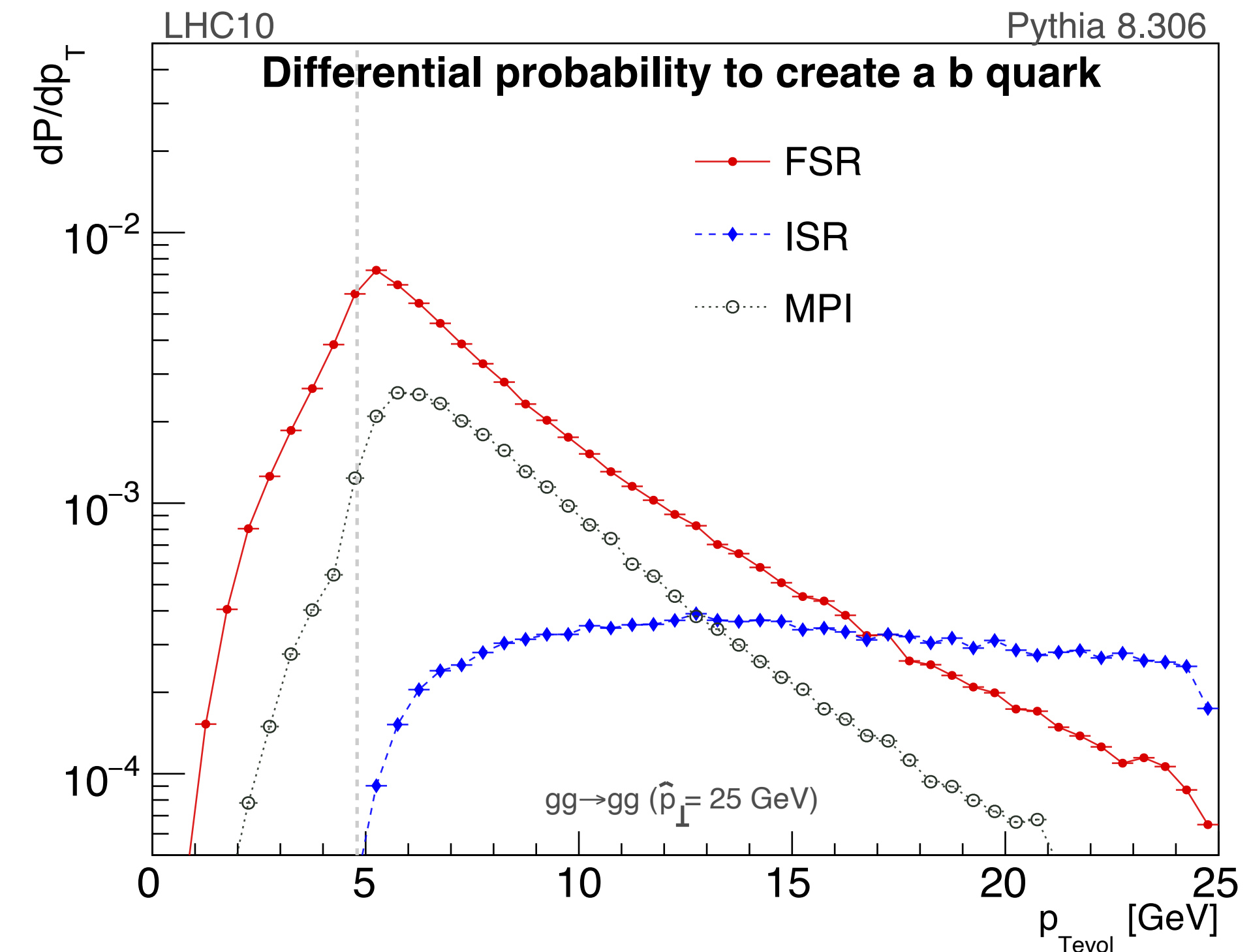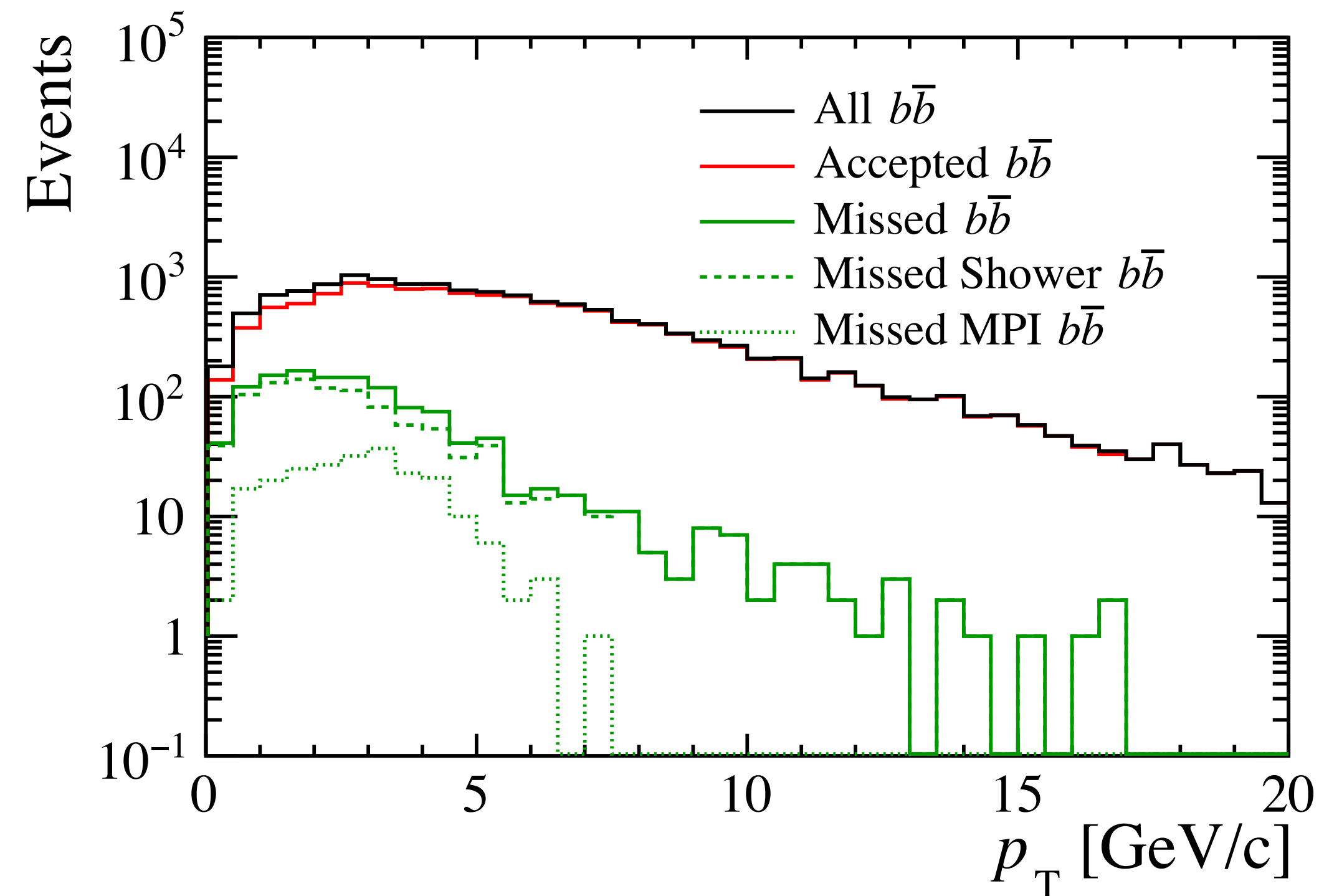
Speed gain

➡ Typically about order of magnitude gain in

➡ As usual, there is small bias

➡ There is some chance to get *b*-quark quite down in energy evolution

Egede, Hadavizadeh, Singla, Skands, Vesterinen: Eur. Phys. J. C 82, 773 (2022)

How to use it in LHCb

➡ Our algorithms use single Pythia8 instance

➡ Once we employ these tricks, we get particle with given quark every time
  ❖ Cannot use same instance for pile-up events

➡ Had some tricks to use different instances, but then proper configuration was very fragile and messy
  ❖ not suitable for production of many different samples

➡ T. Hadavizadeh developed new algorithm, which uses two Pythia8 instances, one for signal and other for pileup
  ❖ Available in MR Gauss!944

➡ Still need to sort out configuration with DecFiles package to be able to use it in general productions and verify setup

Colour reconnection

➡ For some time Pythia8 contains colour reconnection model with junctions, which improves production kinematics

➡ Especially baryons get better

➡ But code was too slow be usable in our production
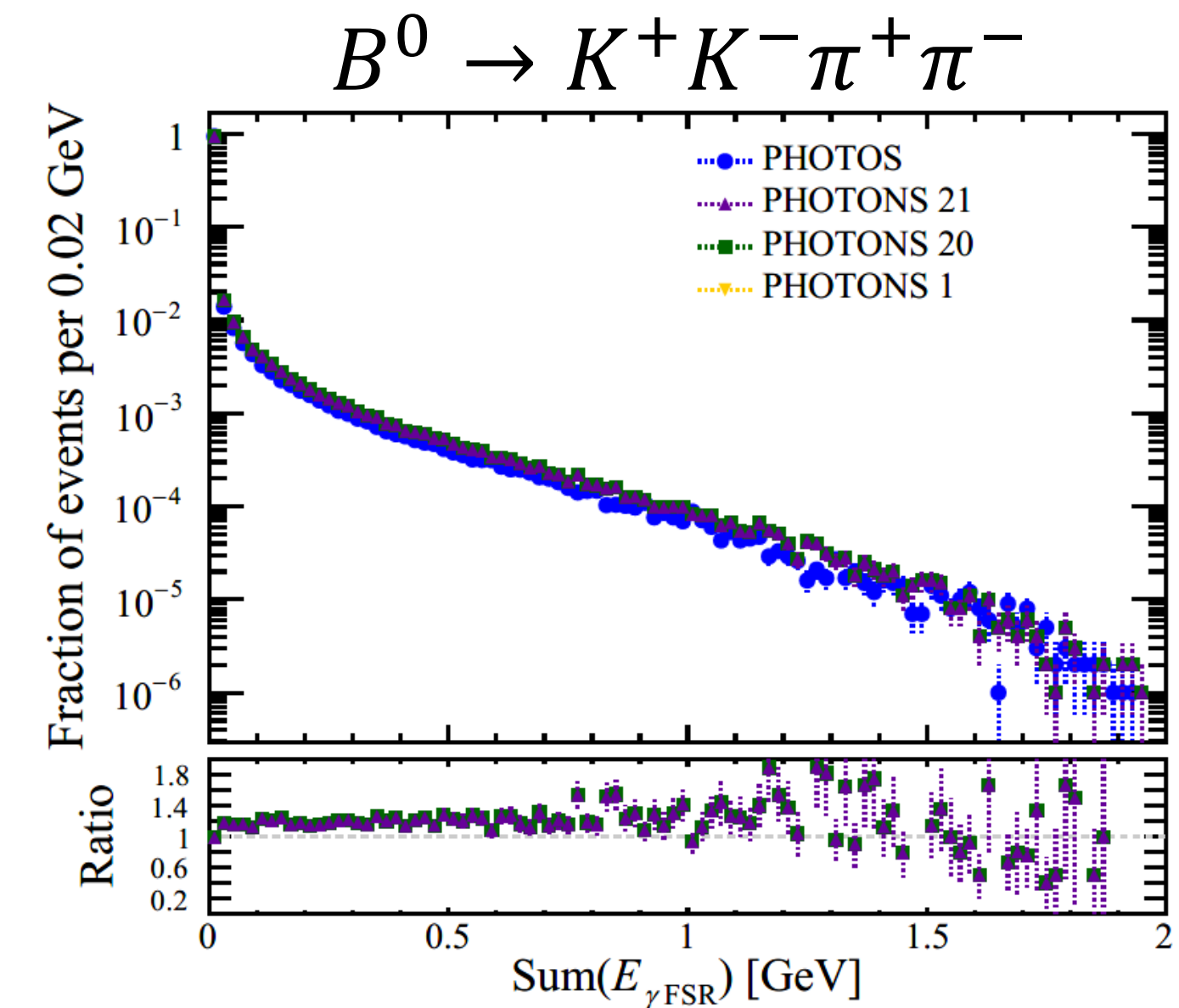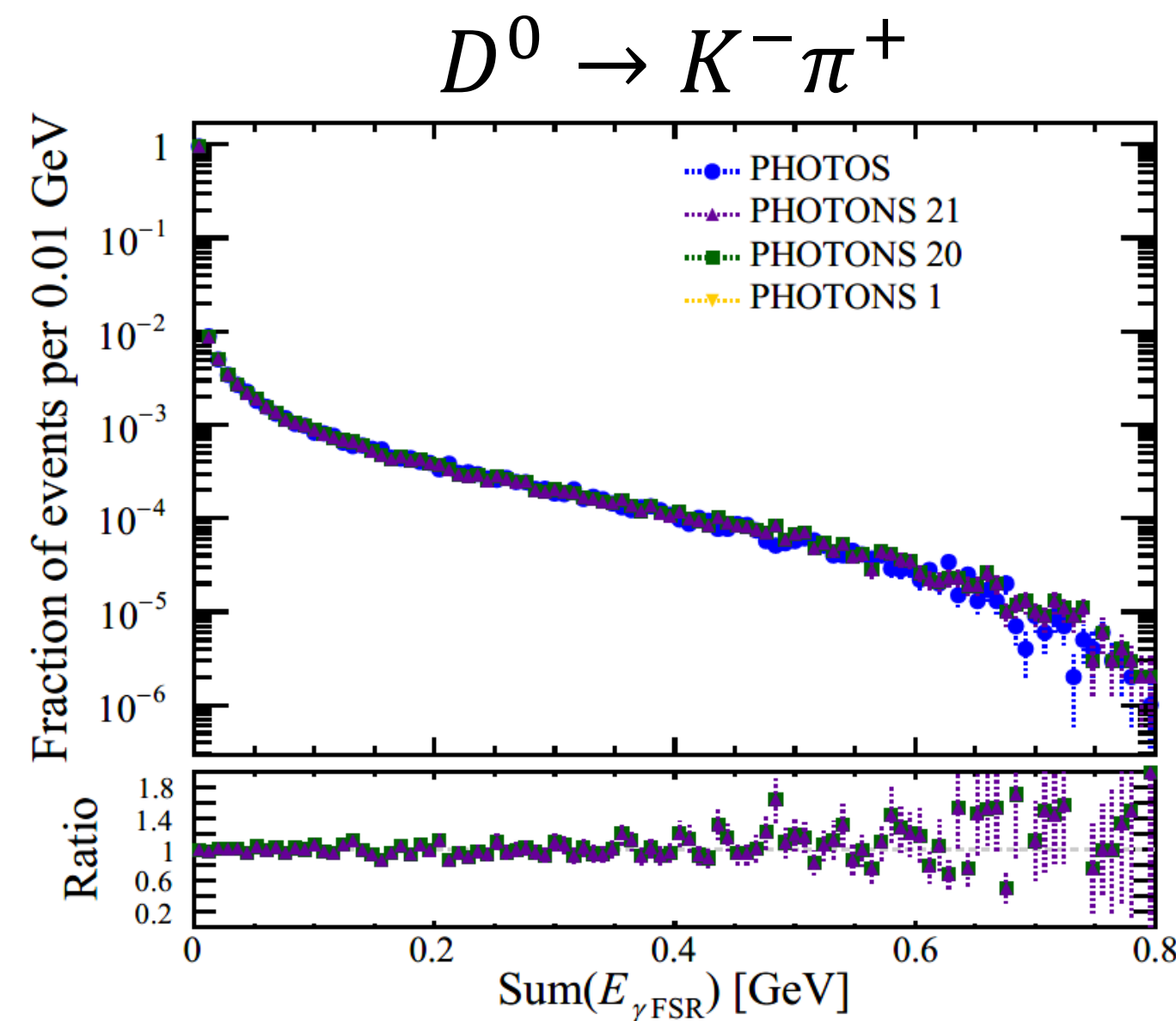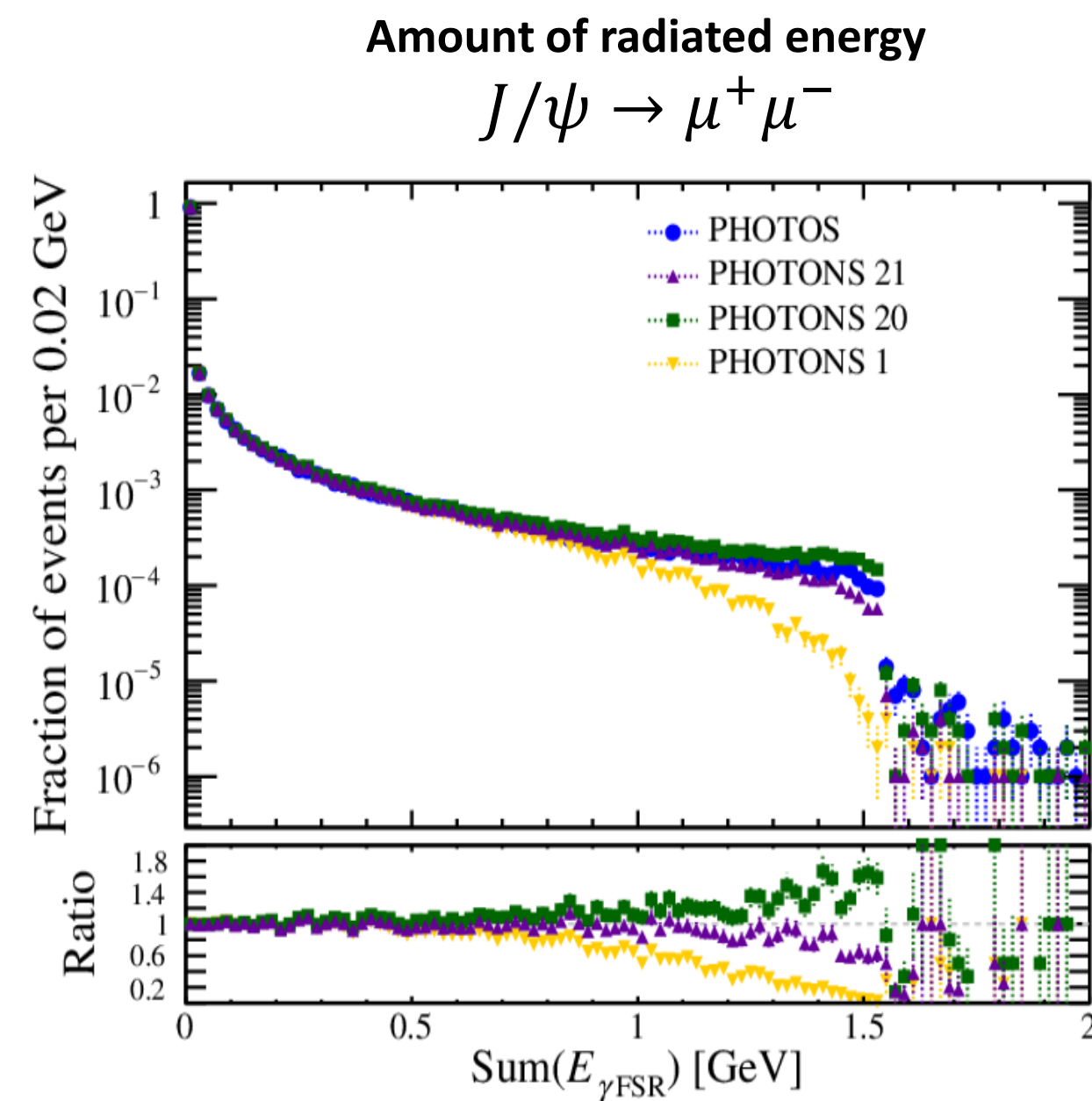
➡ Recently significant improvement in speed

Christiansen & Skands, 1505.01681
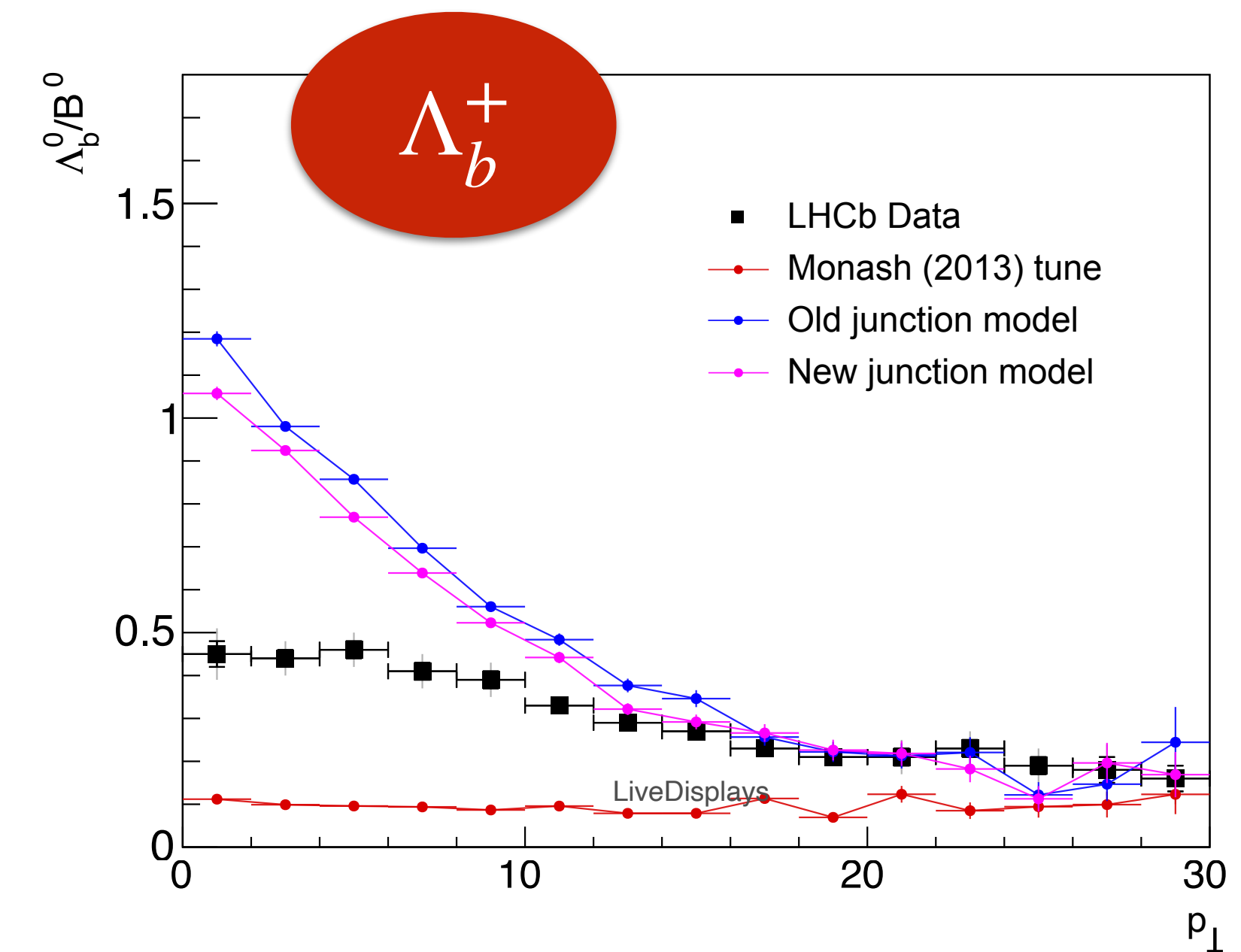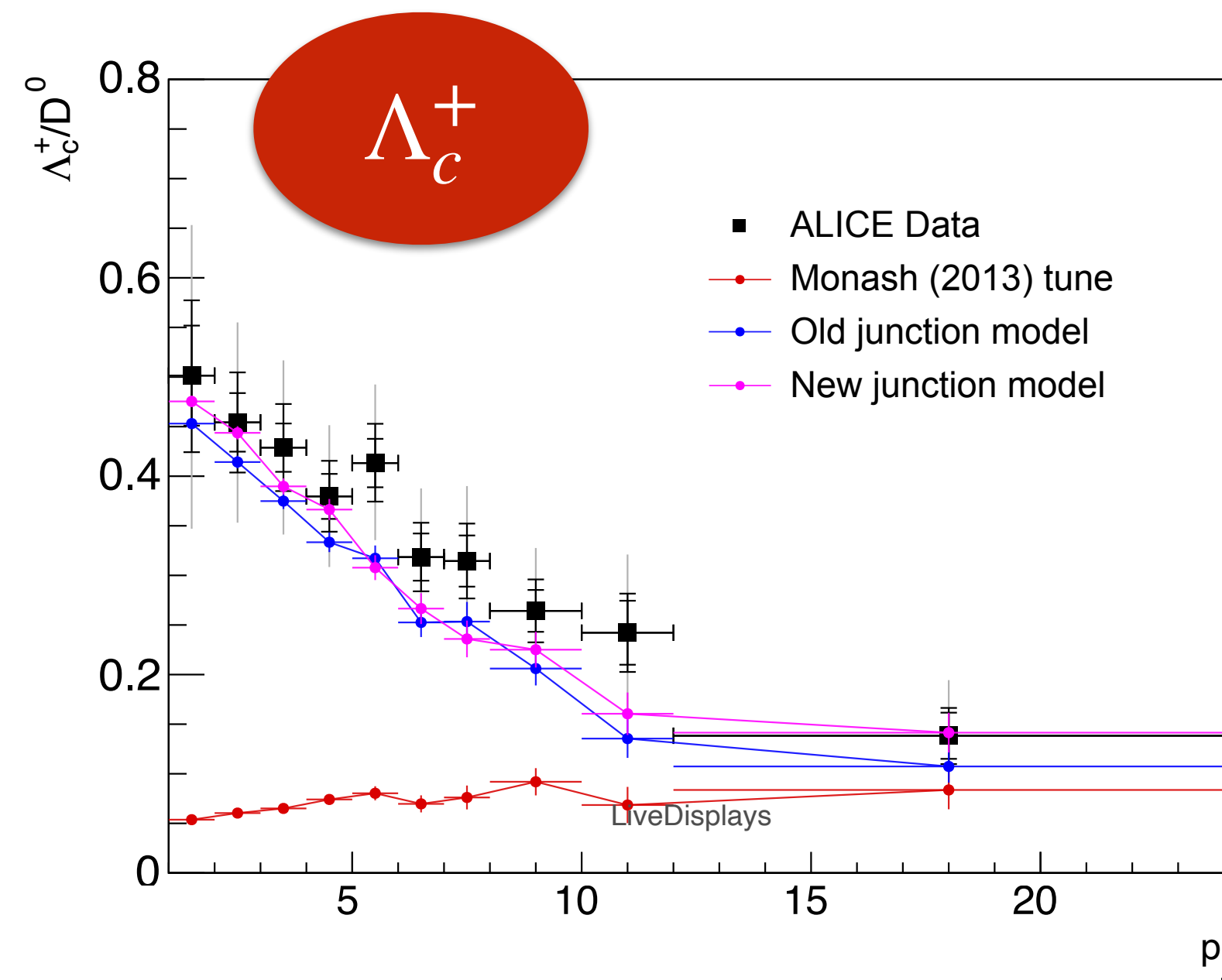
FSR alternative in EvtGen

➡ Most general generators have now some FSR fra[...]

   ❖ Vincia in Pythia8 is coming, Photons++ in Sherpa rea[...]

➡ We have implemented interface in EvtGen, so or[...]
Photos and Photons++ (and Vincia when release[...]

Amount of radiated energy

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

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

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

Summary

➡ Couple of simple tricks can speed up event generation by Pythia8
  ❖ Small bias is probably not too important

➡ Relevant part is prepared for Gauss and need work on configuration to use it in production

➡ Potentially might be able to replace also BcVegPy and GenXicc generators

➡ Colour reconnection code in Pythia8 was optimised and we expect better description of production kinematics

➡ FSR alternatives in EvtGen soon to be available in LHCb simulation

➡ Work continues by Pythia8 authors on further improvements
  ❖ We have close contact and should benefit relatively quickly