Jet flavour labelling at ATLAS

Radosław Grabarczyk

Flavoured Jets at the LHC, 11.06.2024

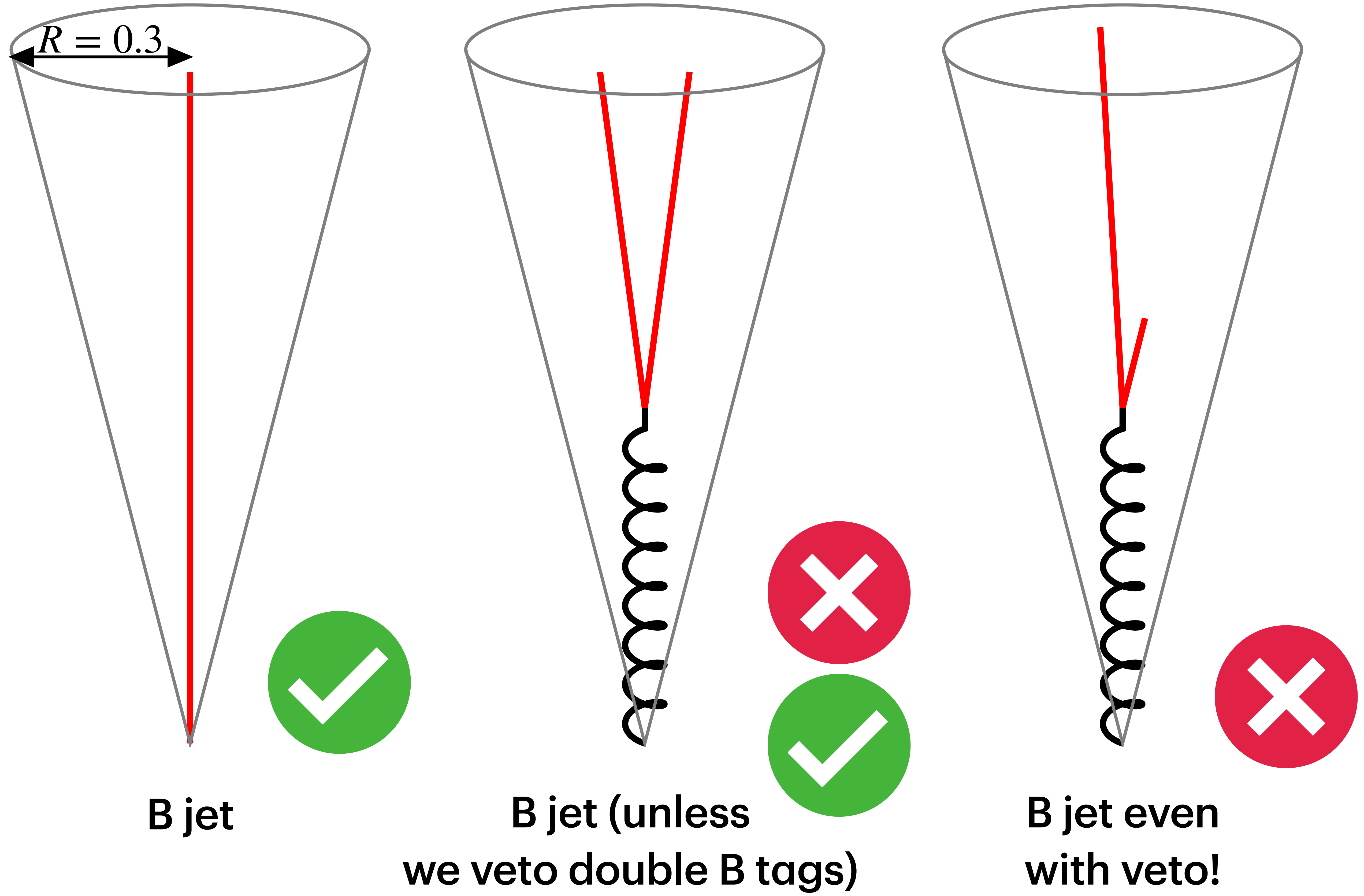
Outline

- Current flavour tagging tactics
- Idea to ensure IRC safety of jet flavour
- Our reference *b*-tagging samples clustered with the new algorithms

Current ATLAS jet flavour definition

ATLAS deltaR labelling:

- Take the 4-momenta of reconstructed jets and B hadrons with $p_T > p_{T\mathrm{cut}} = 5$ GeV
- Assign a B to the jet if $\Delta R = \sqrt{\Delta y^2 + \Delta\phi^2} < 0.3$
- If at least one B is assigned, call it a B-jet



+ inherited flavour problems from anti-k_T...

ATLAS EXPERIMENT

Samples that we train on

$20\,\text{GeV} < p_T < 250\,\text{GeV}$ jet tagging: **semileptonic $t\bar{t}$ events**

Nominally PowhegBox+Pythia 8.230

require $\eta < 2.5$

$250\,\text{GeV} < p_T < 5000\,\text{GeV}$ jet tagging: **high mass $Z' \to q\bar{q}$**

Nominally Pythia 8.212

require $\eta < 2.5$

we also considered variations from MadGraph + Herwig that led to very large parton shower systematics

ATLAS EXPERIMENT

Samples that we train on

$20\,\text{GeV} < p_T < 250\,\text{GeV}$ jet tagging: **semileptonic $t\bar{t}$**

Nominally PowhegBox+Pythia 8.230

require $\eta < 2.5$

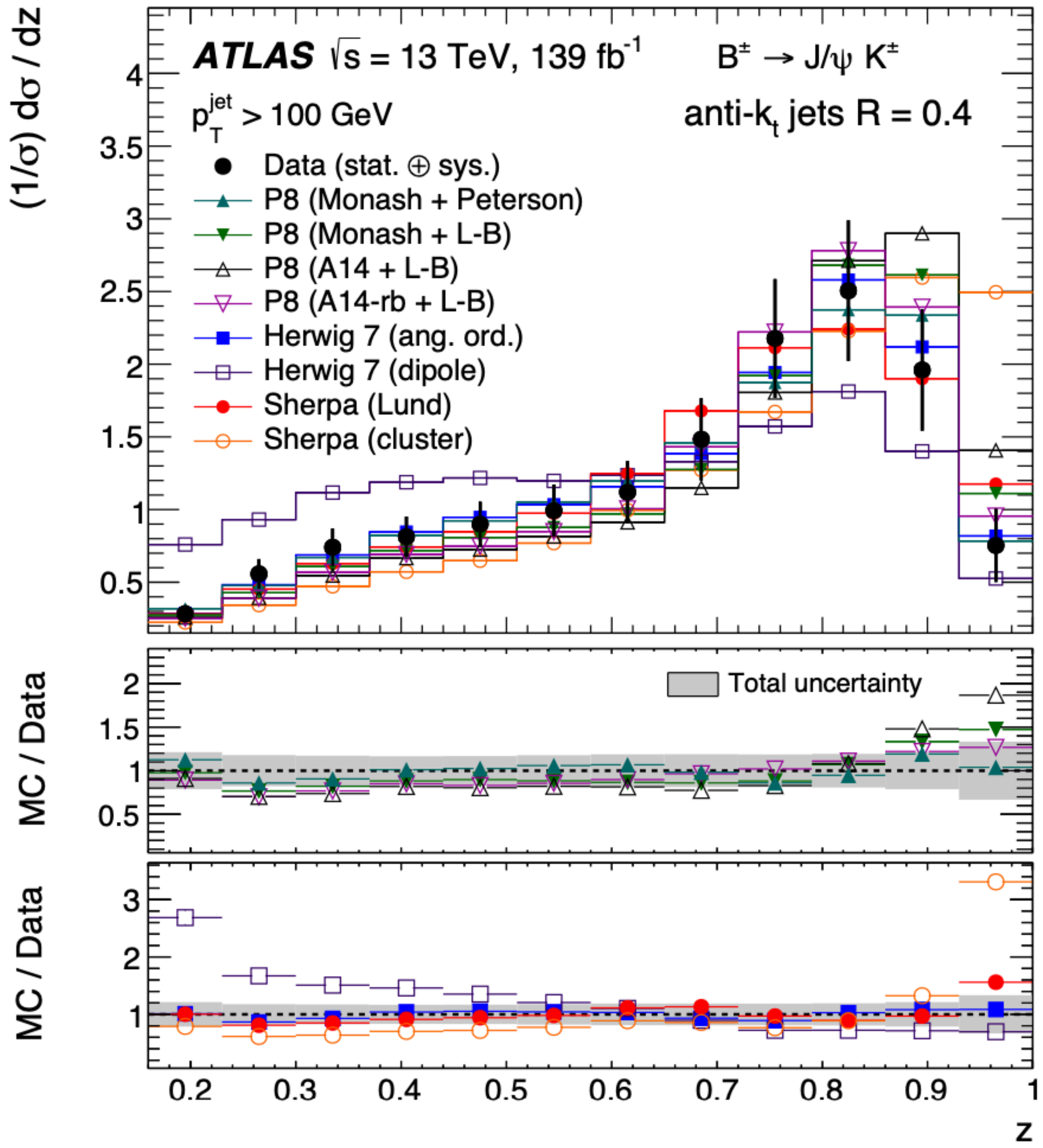
$250\,\text{GeV} < p_T < 5000\,\text{GeV}$ jet tagging: **high mass Z'**

Nominally Pythia 8.212

require $\eta < 2.5$

we also considered variations from MadGraph + Herv
shower systematics

(2108.11650, $p_T > 100\text{GeV}$)



ATLAS EXPERIMENT

Goal for ATLAS

Ensure IRC safety of jet flavour in our analyses:

Use taggers trained on IRC safe labels
→ good to know how many configurations that could spoil IRC safety there are in the training dataset.

Unfold to particle level with phase space defined in an IRC safe way

Goal for ATLAS

Ensure IRC safety of jet flavour in our analyses:

Use taggers trained on IRC safe labels
→ good to know how many configurations that could spoil IRC safety there are in the training dataset.

Unfold to particle level with phase space defined in an IRC safe way

Wishful flavour labelling implementation

For tests of flavour labelling we adopt a wishful thinking tactic:

- Take all truth particles that are inputs to jet algorithms
- Remove all children of all B hadrons (with no p_{tcut} applied on the B hadrons)
- Undecayed B hadrons + other jet input truth particles = input to jet flavour algorithms

The result is a list of jets made from (truth particles + undecayed B hadrons):

→ Match them under a $\Delta R < 0.3$ condition to reco jets

(highest p_T match wins)

(this is not the best thing to do for flavour dressing - dressing reco jets with “accumulated” B-hadrons directly to be implemented...)

ATLAS EXPERIMENT

Algorithms implemented

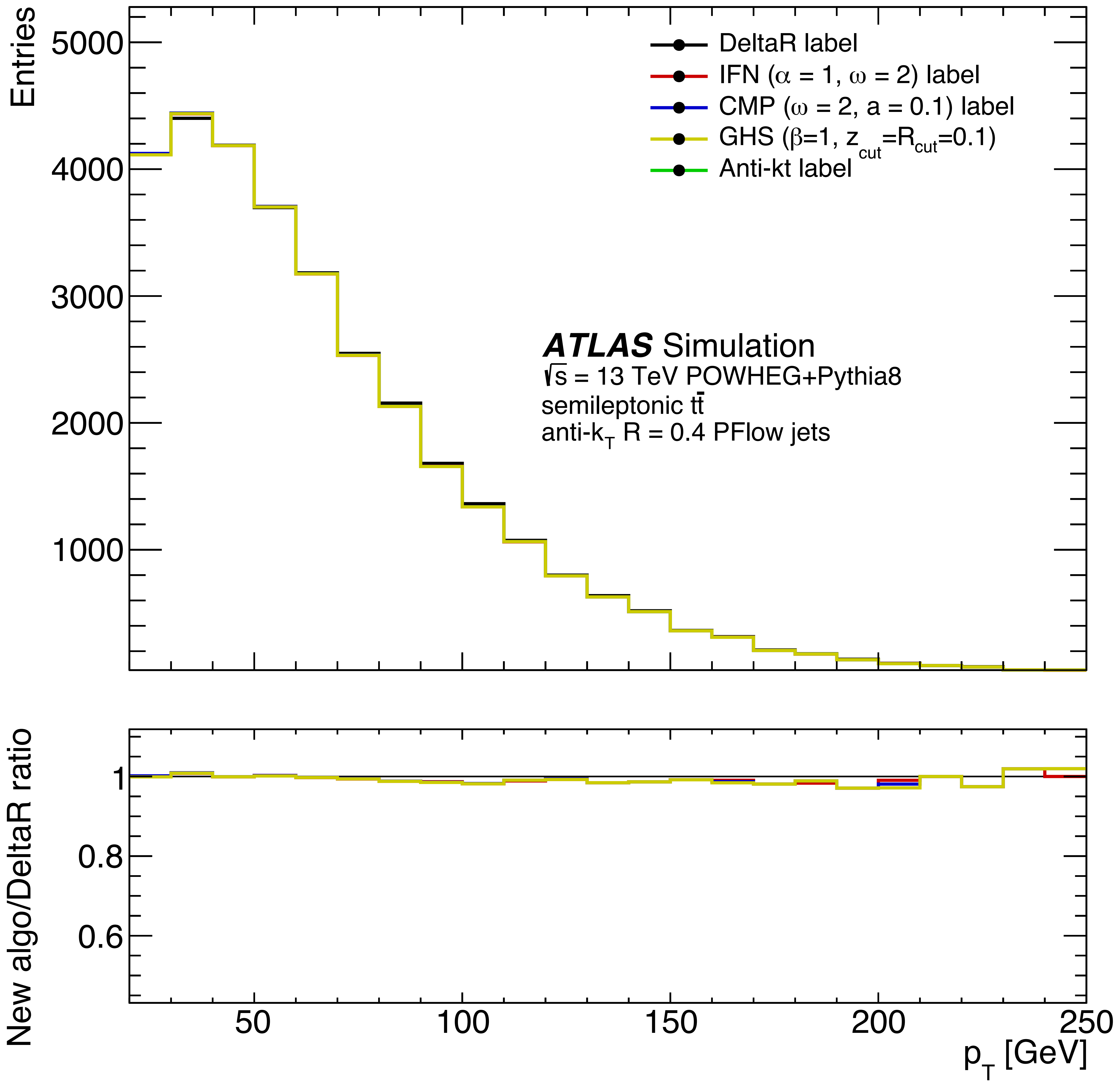
- Flavour anti-k_T (CMP): $\omega = 2, a = 0.1$
- Flavour dressing (GHS): $\beta = 1, R_{\text{cut}} = z_{\text{cut}} = 1$
- Interleaved Flavour Neutralisation (IFN), $\omega = 2, \alpha = 1$

Soft Drop flavour remains work in progress for us due to problems with ATLAS software and fjcontrib… (but once it becomes part of fjcontrib, will be trivial to add)

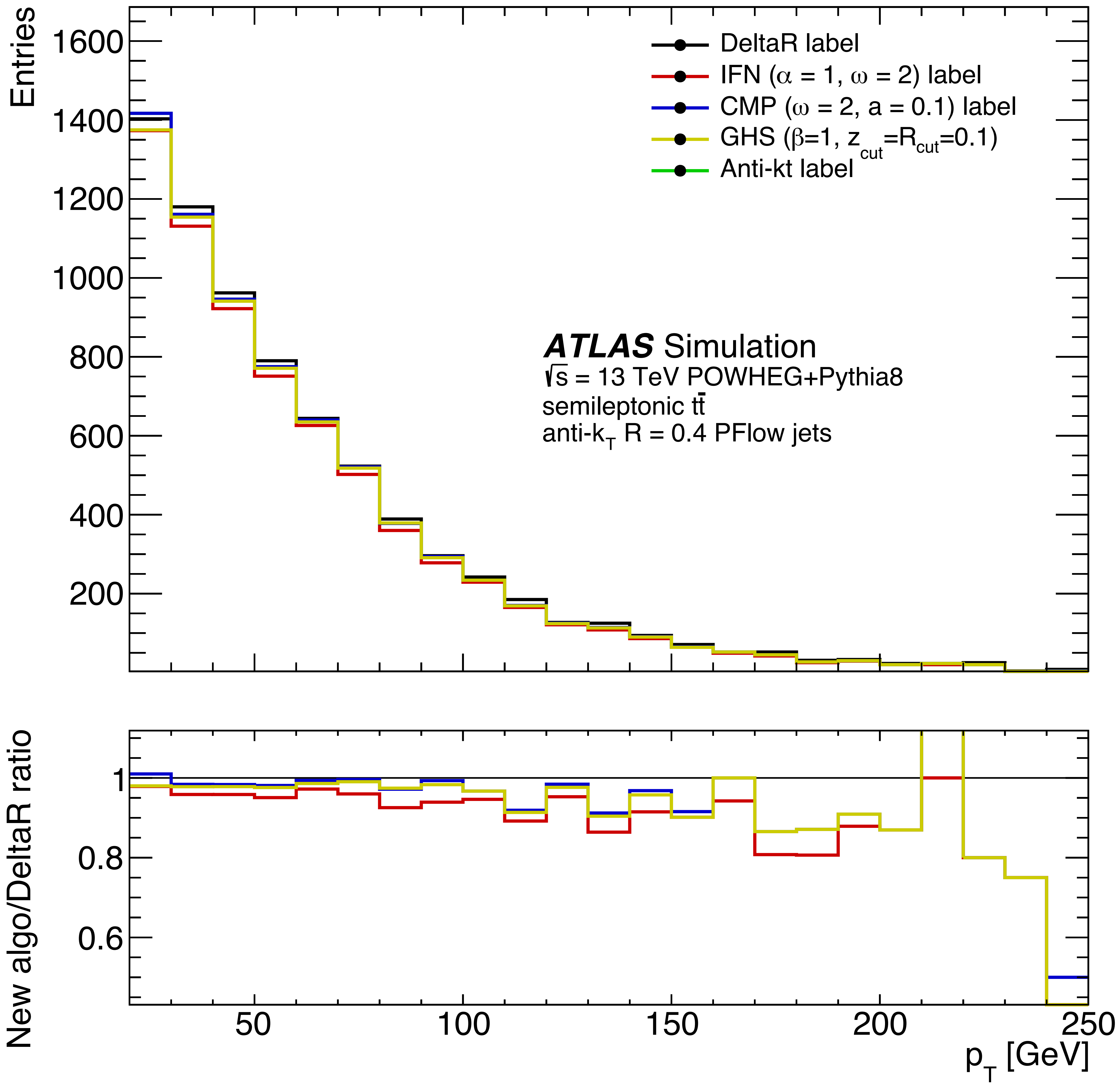
ATLAS EXPERIMENT

Results: ttbar

b-tagging:

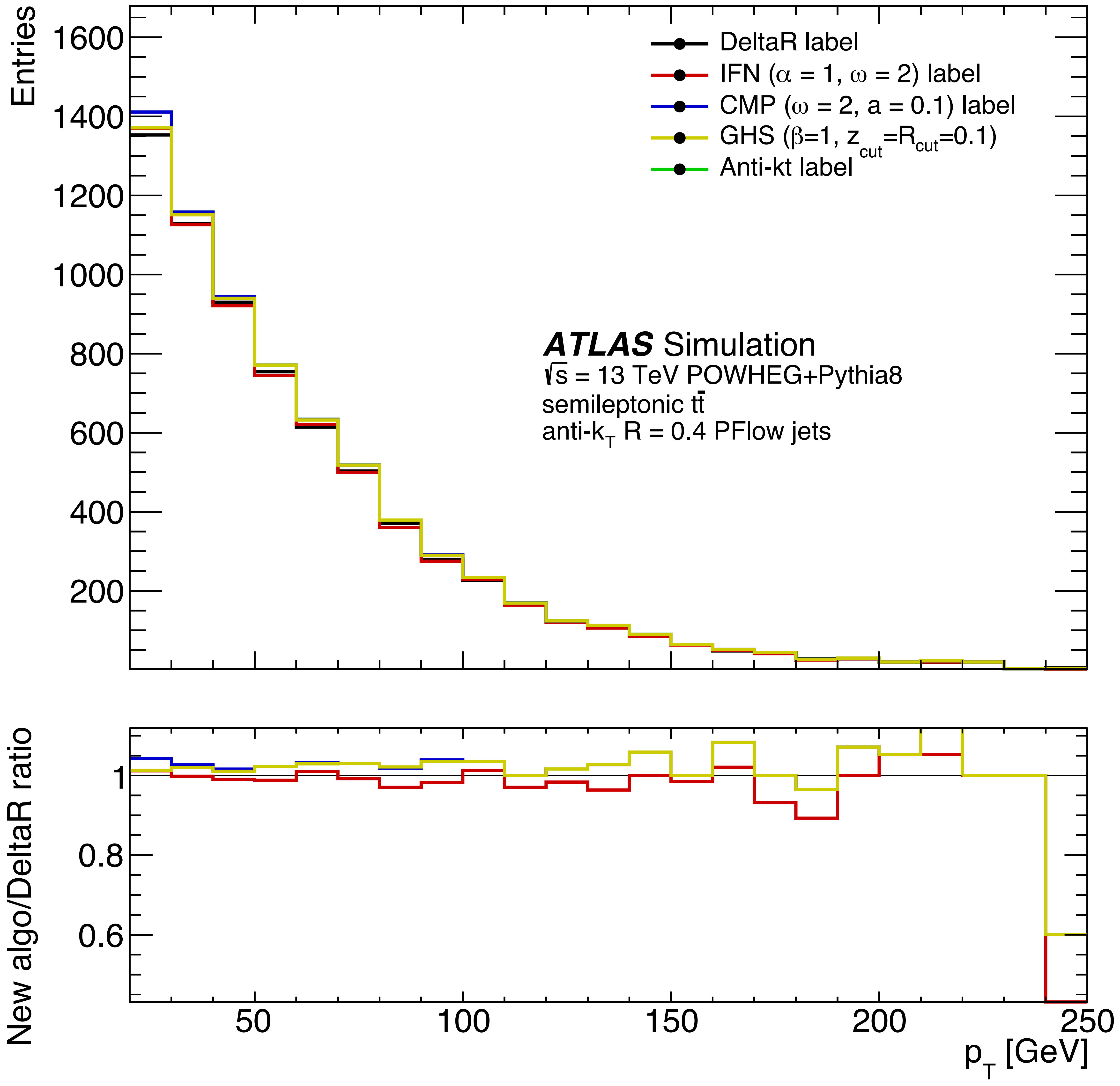


c-tagging:

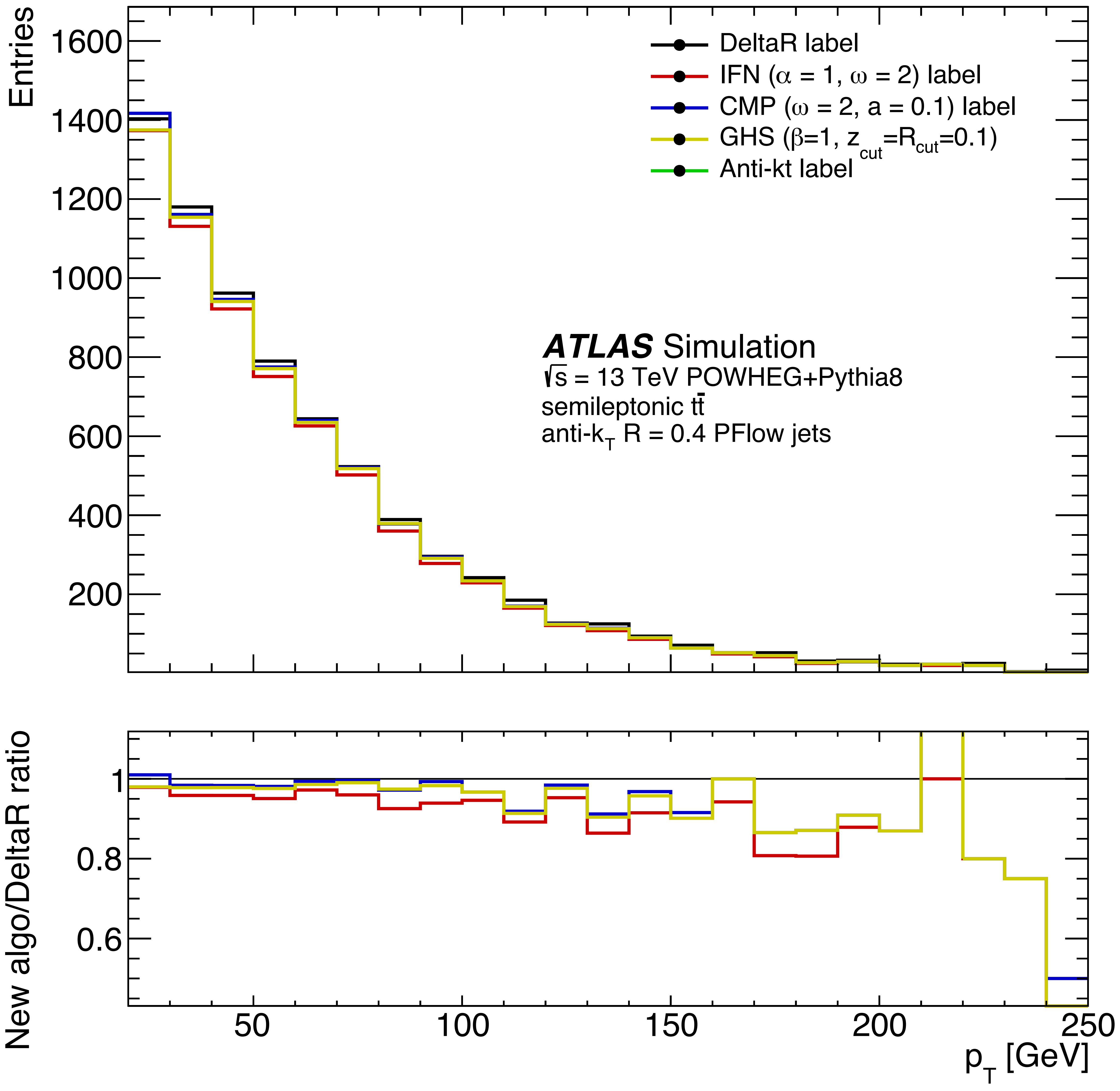


Results: ttbar

c-tagging with ATLAS double c-tag veto:

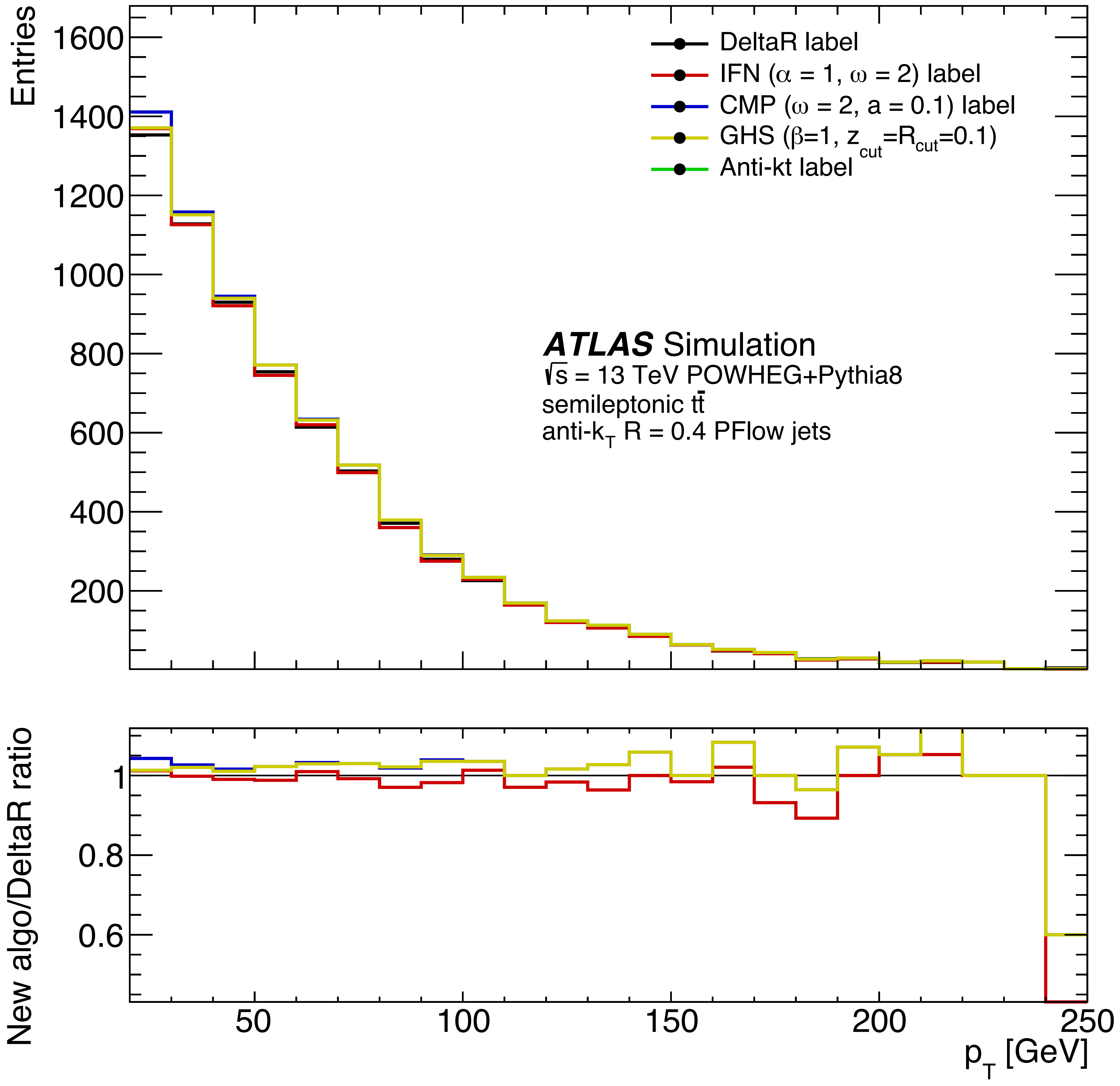


c-tagging:

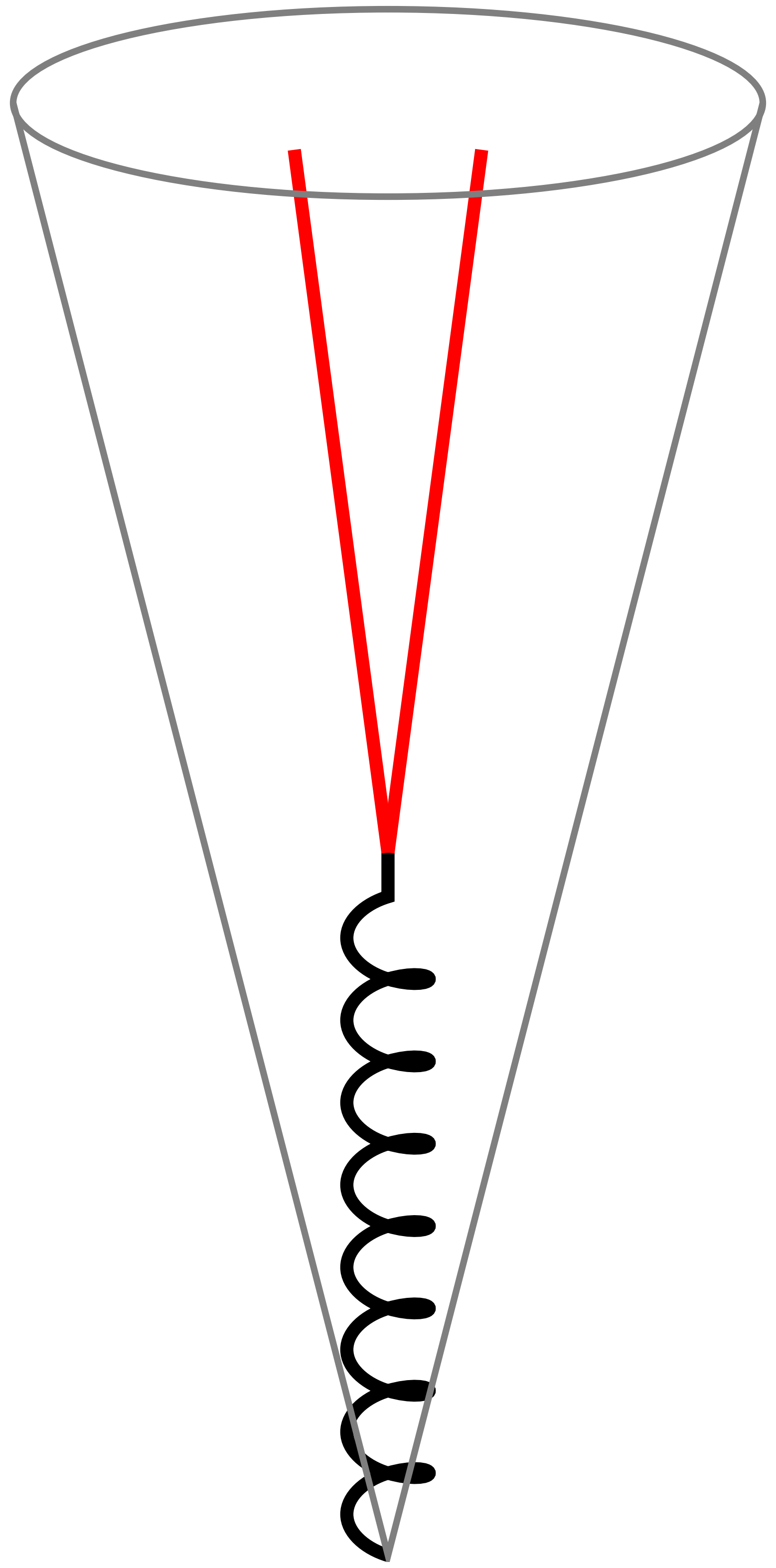


Results: ttbar

c-tagging with ATLAS double c-tag veto:

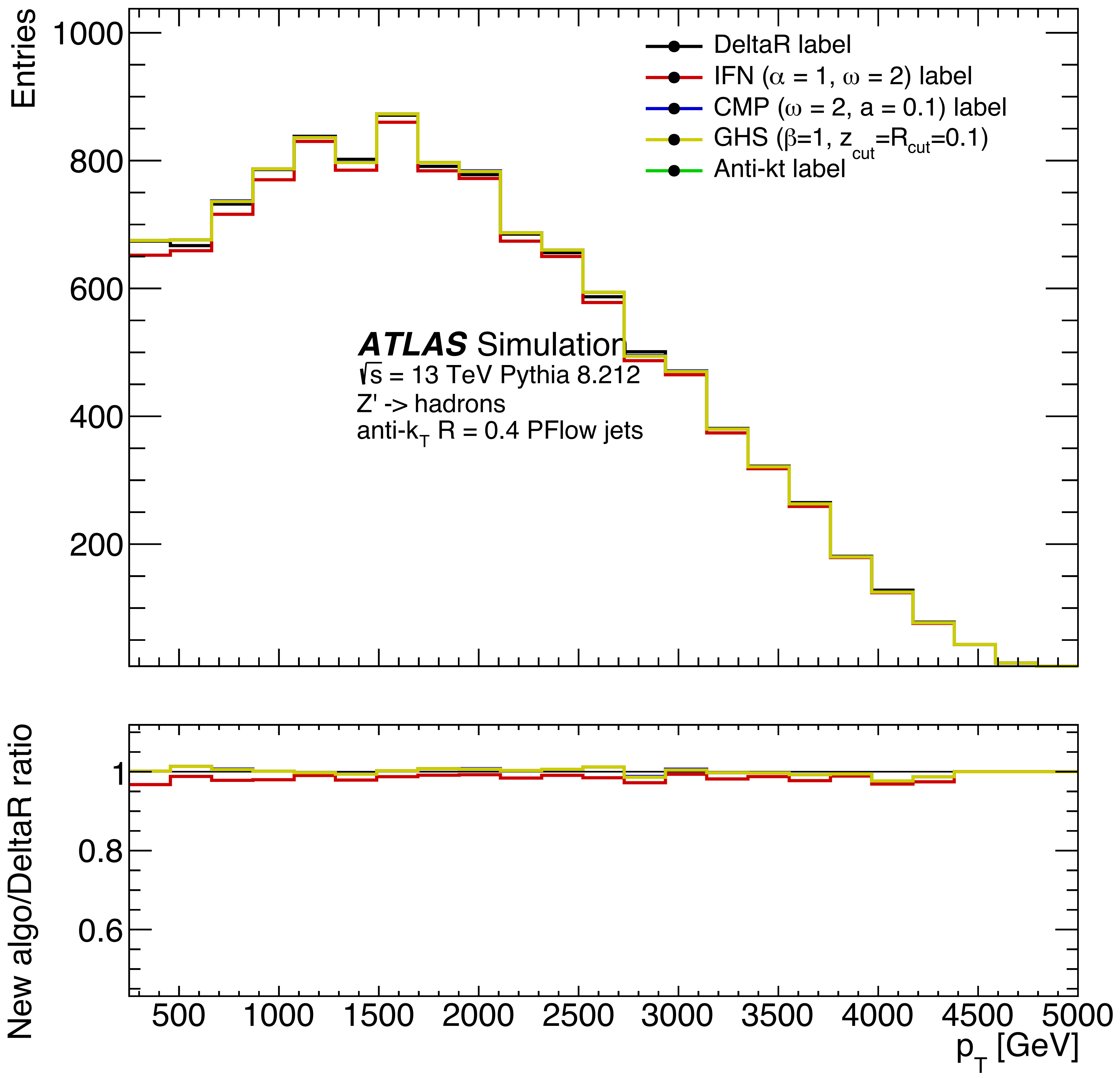


Most of the differences explained by...

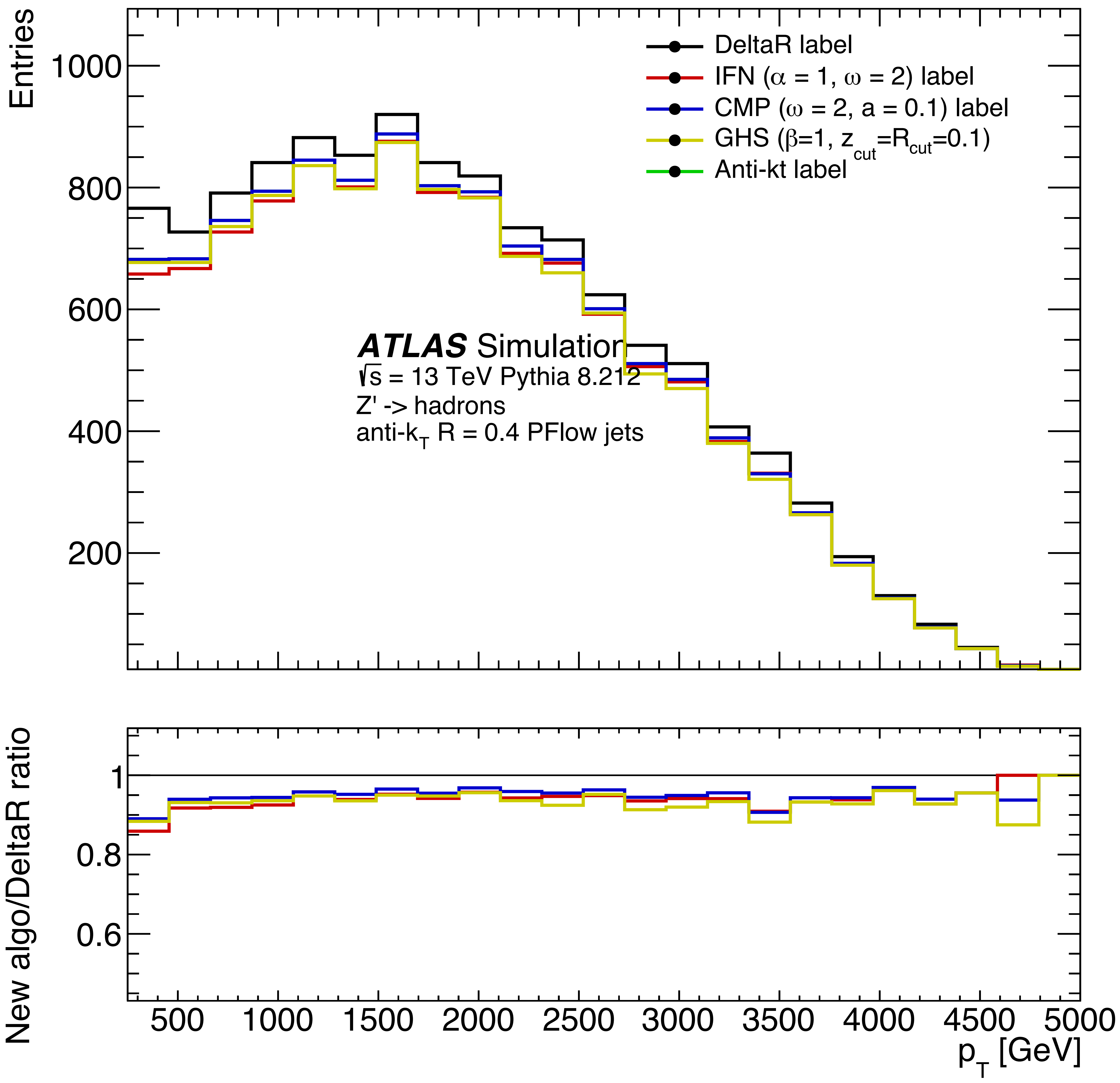


Results: Z'

b-tagging with ATLAS double b-tag veto:

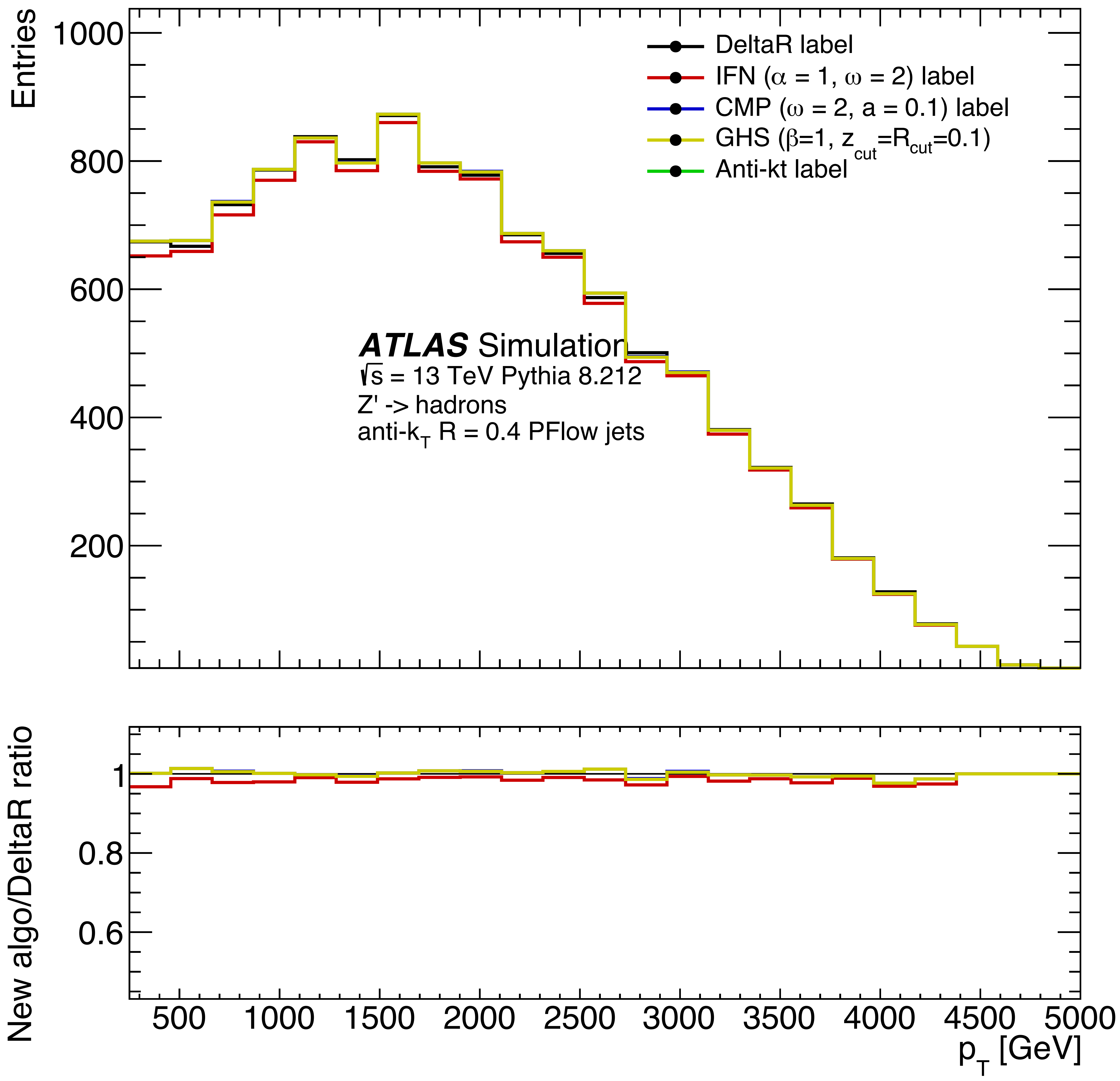


b-tagging:

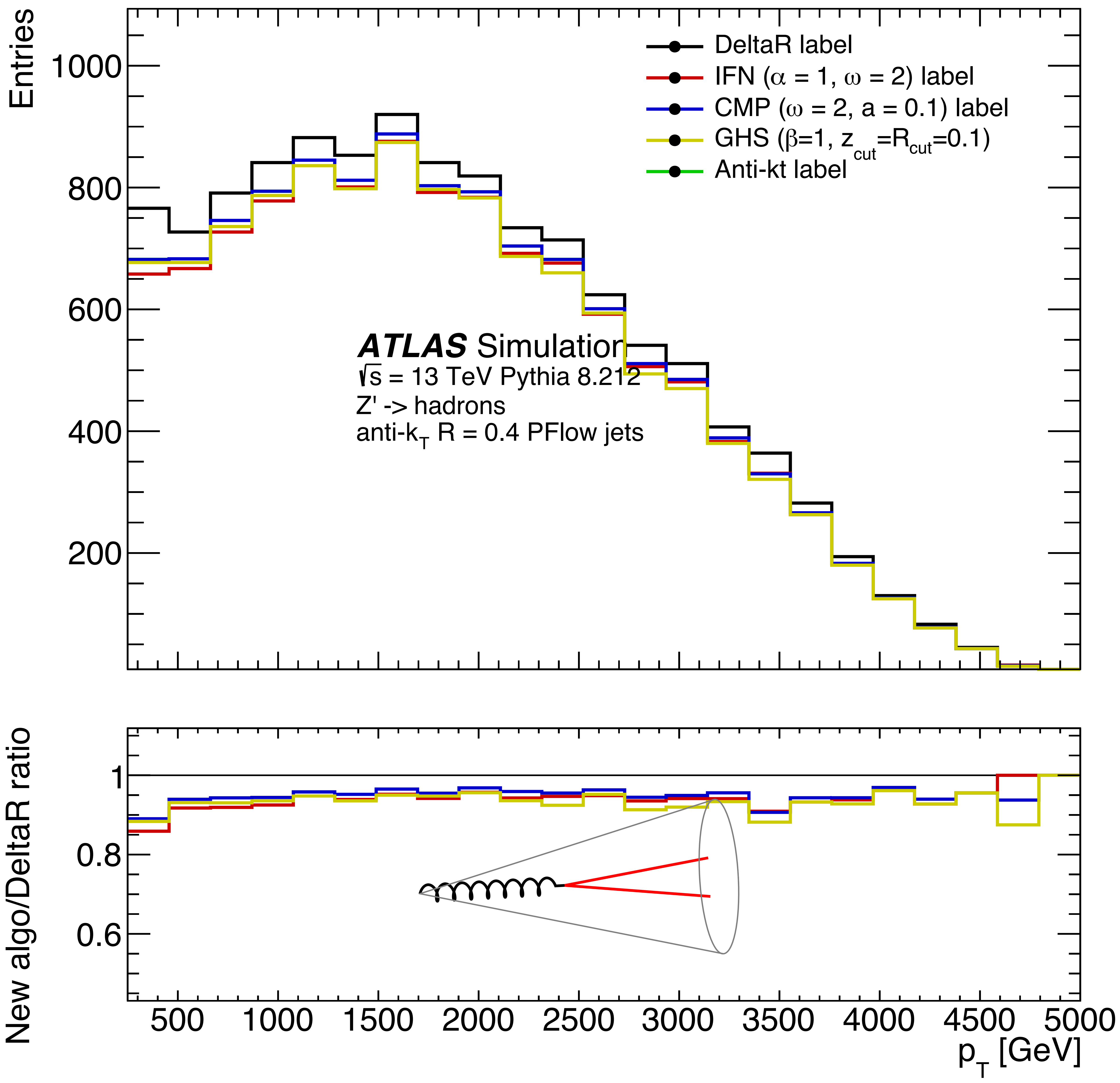


Results: Z'

b-tagging with ATLAS double b-tag veto:

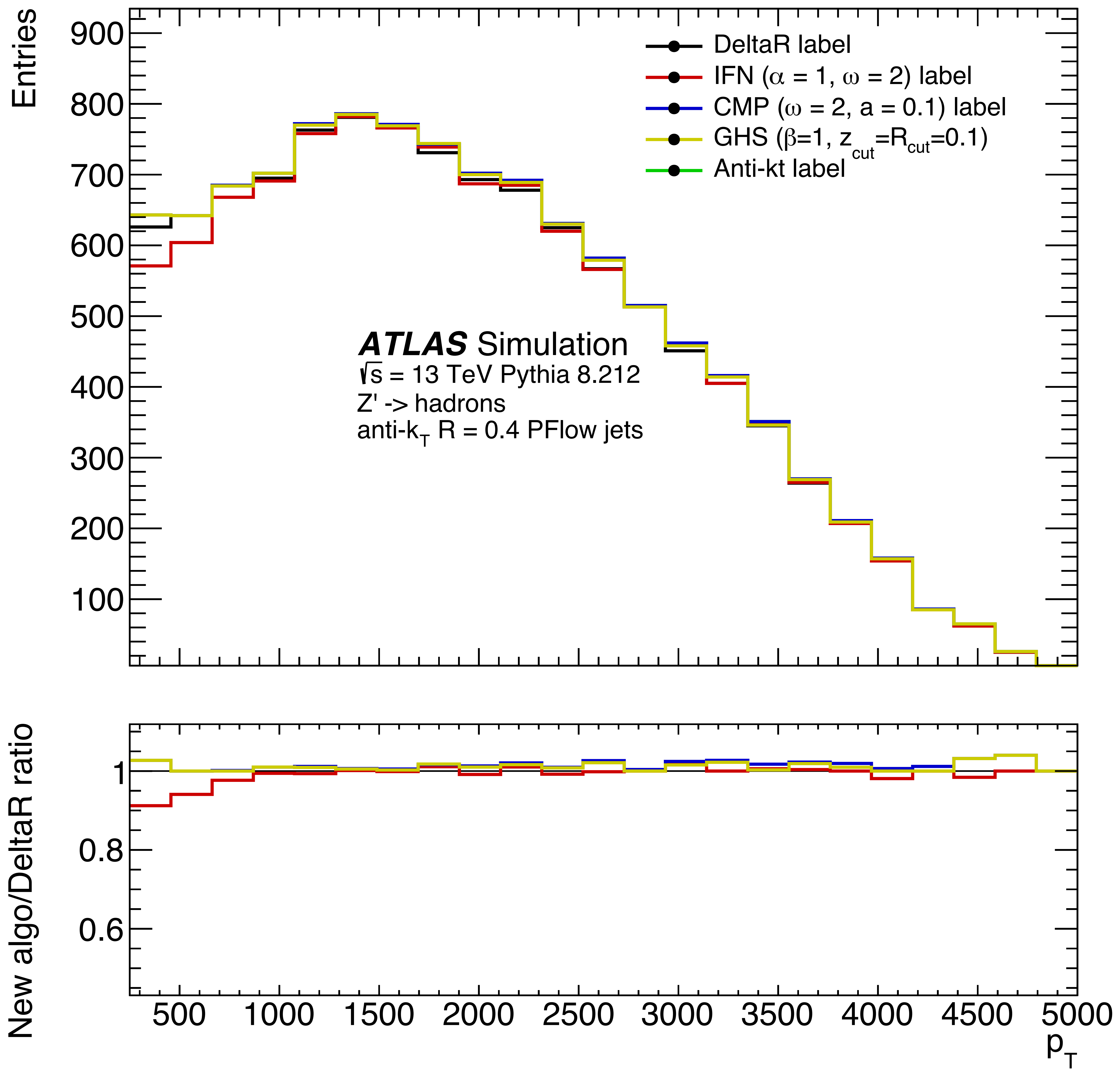


b-tagging:

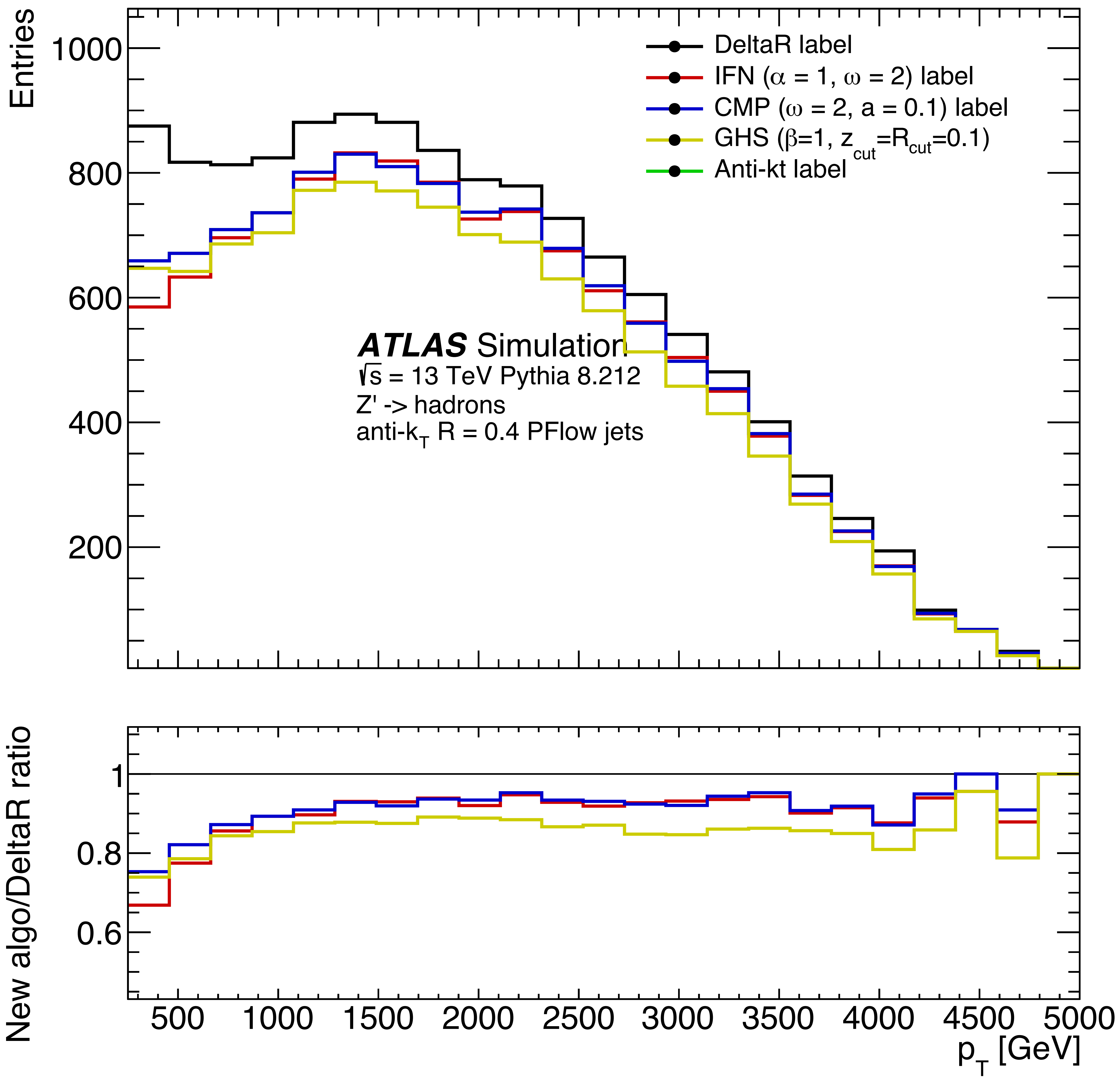


Results: Z'

c-tagging with ATLAS double c-tag veto

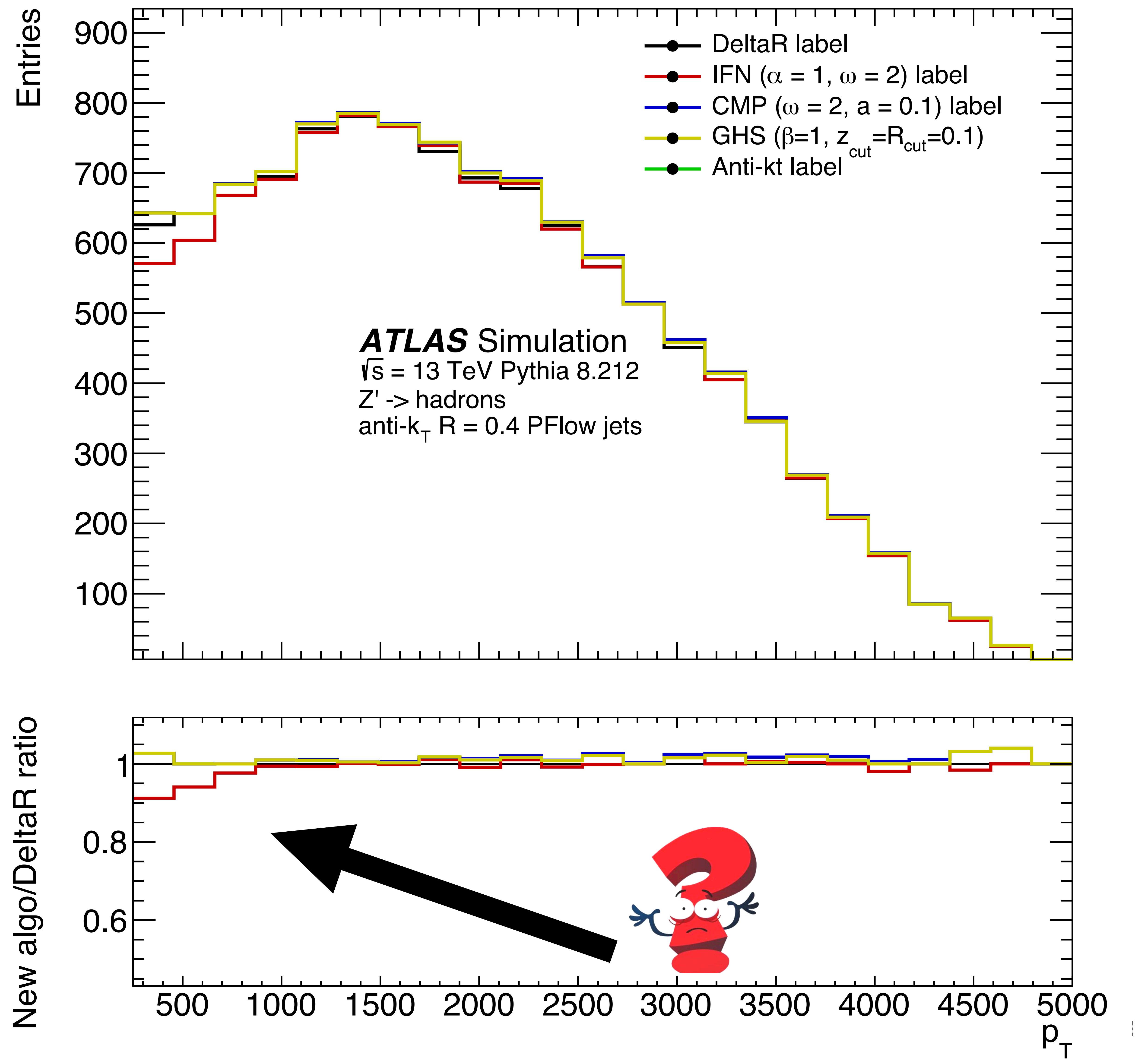


c-tagging

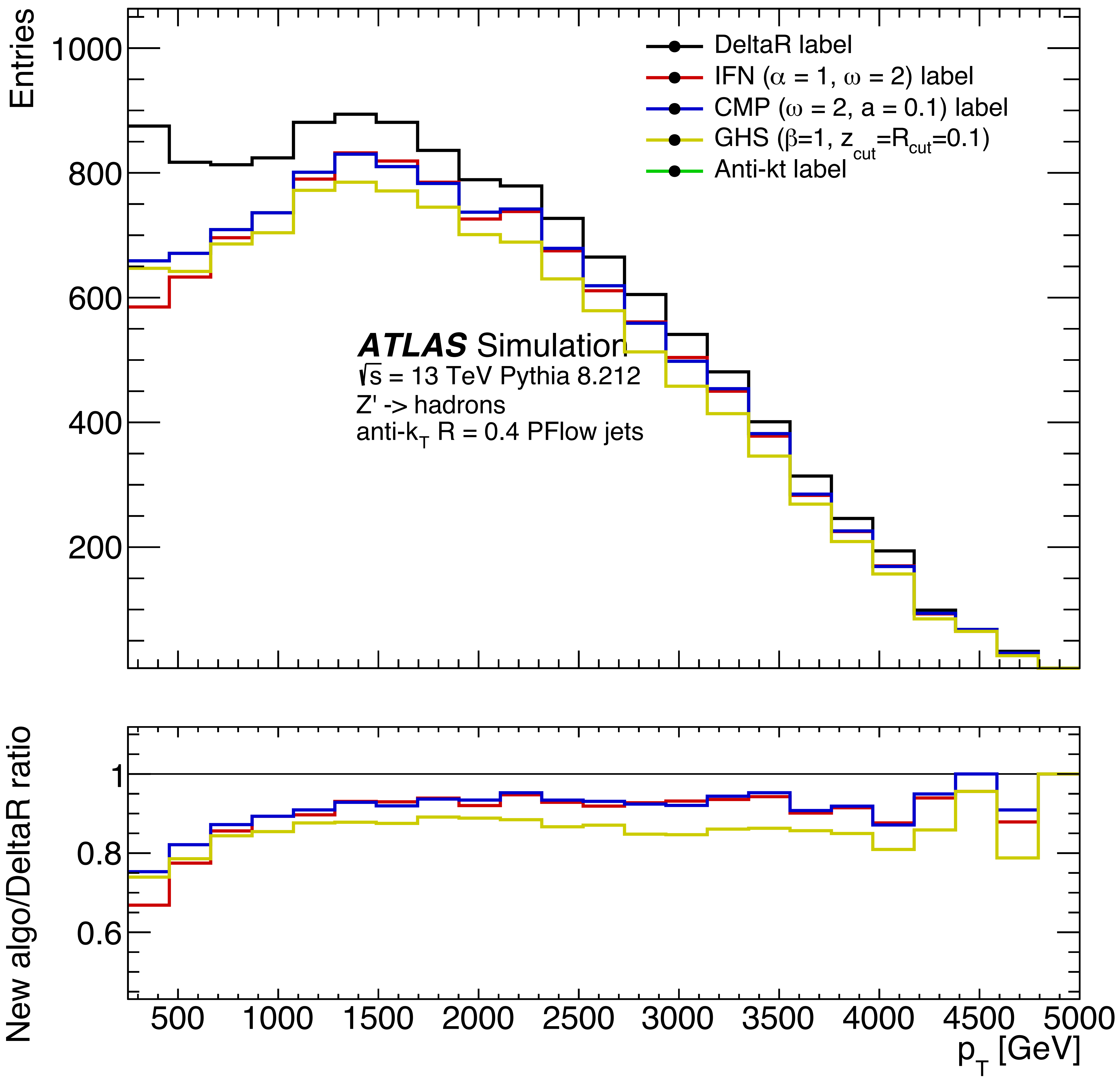


Results: Z'

c-tagging with ATLAS double c-tag veto



c-tagging

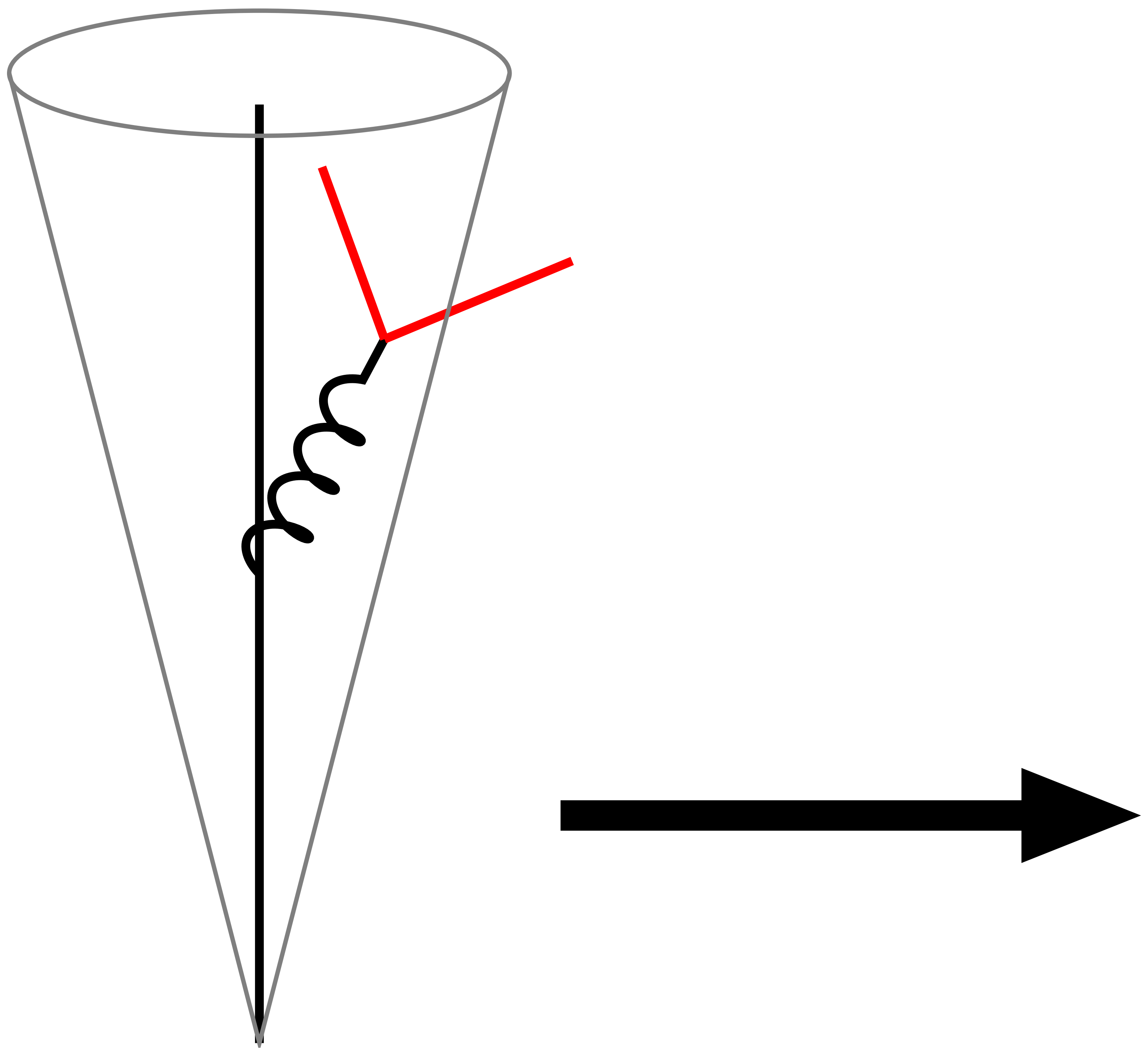


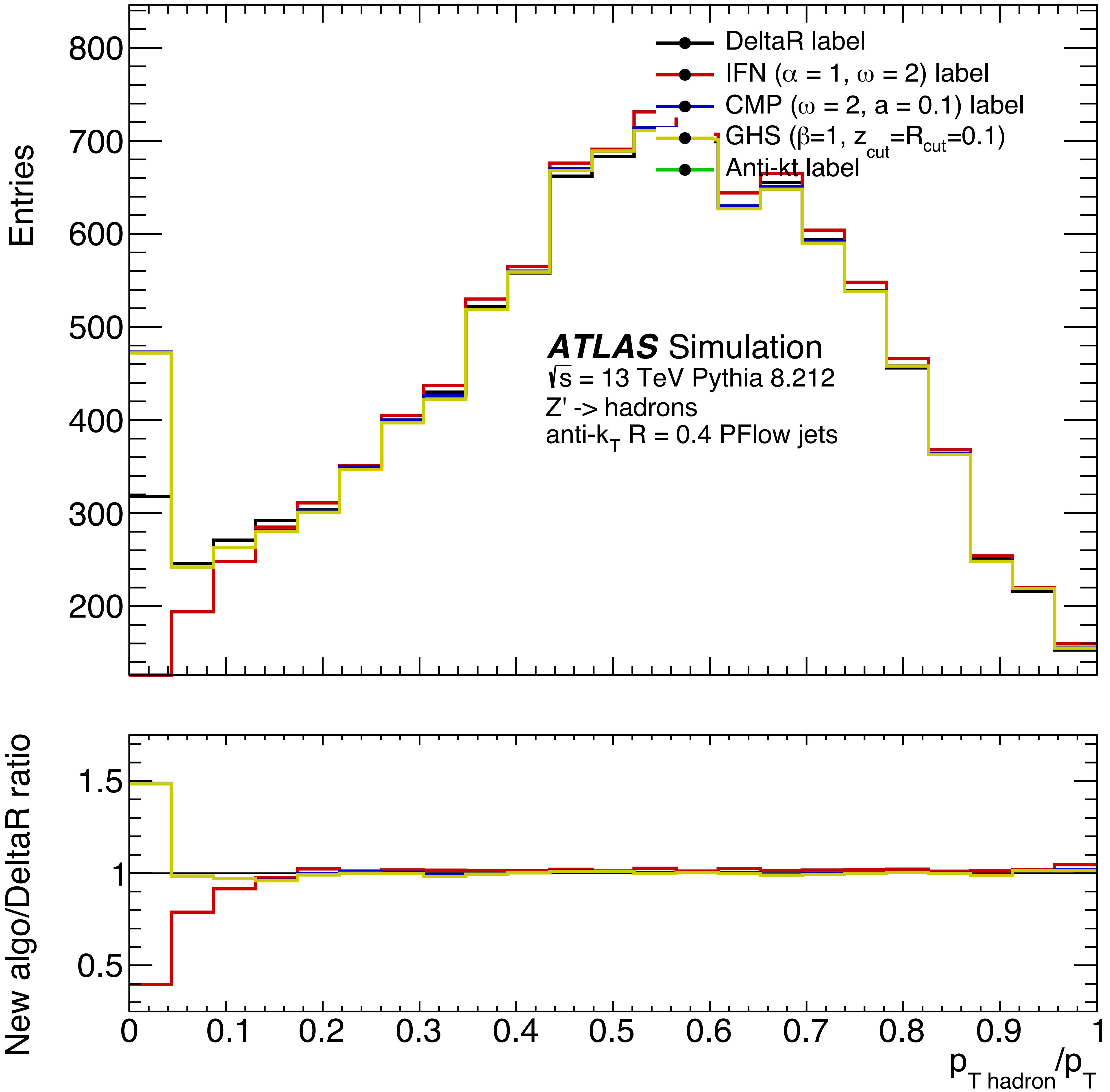
Results: Z'

We can investigate this difference by requiring 1 c-hadron inside the c-tagged jets and looking at the $p_{Thadron}/p_T$ ratio

Results: Z'

For jets with 1 ATLAS matched c-hadron...

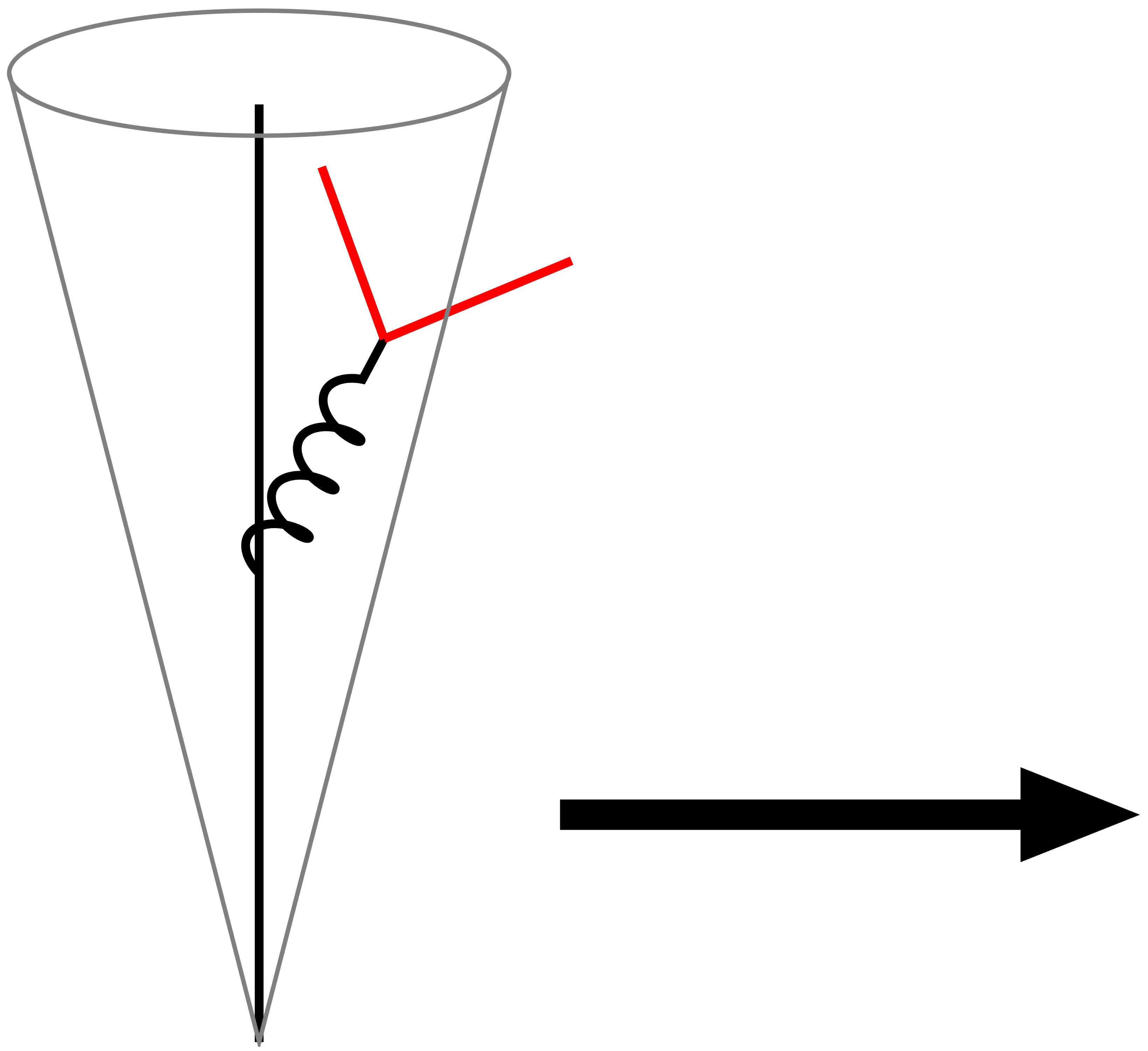


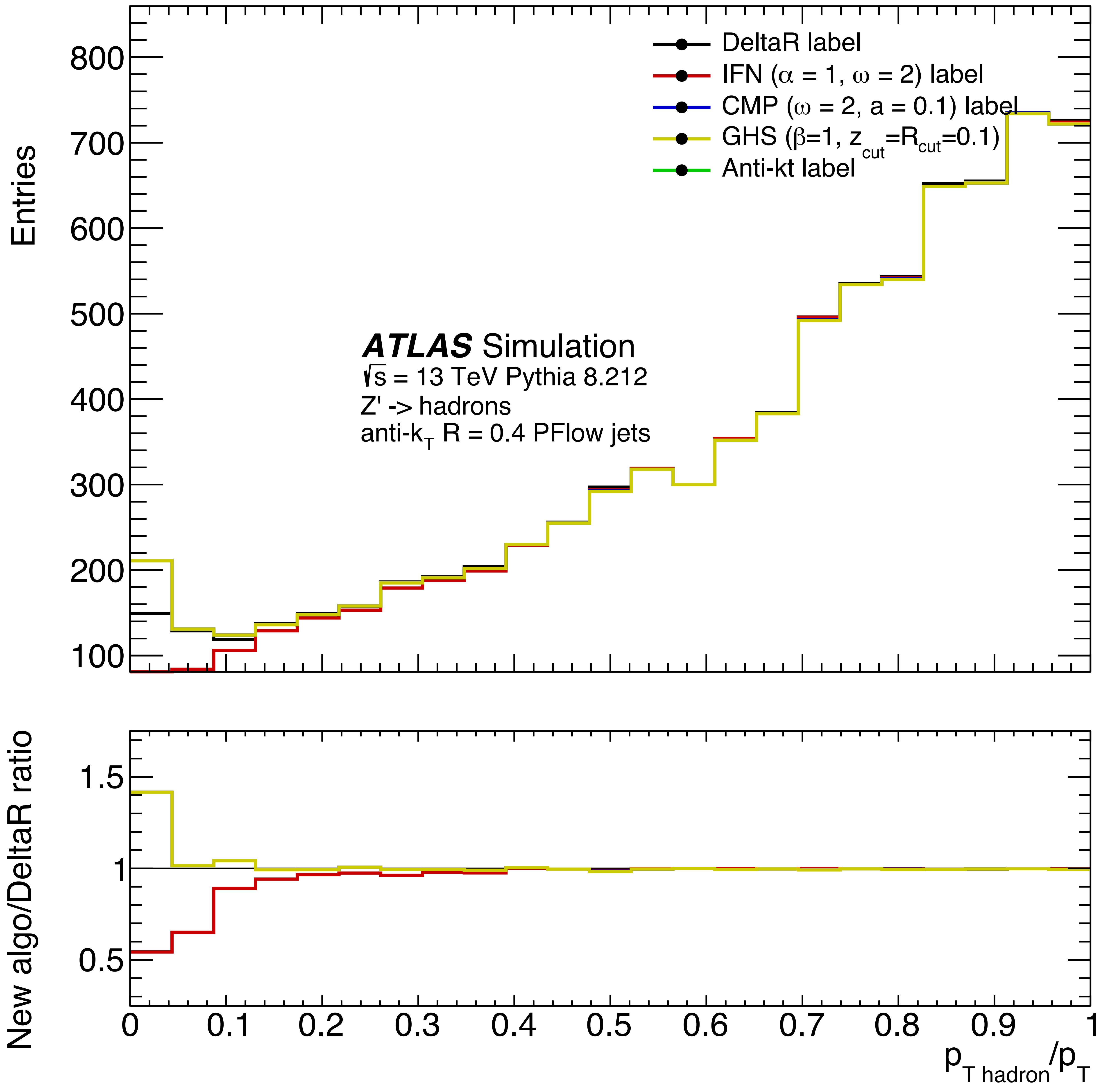


Results: Z'

We see that this also happens for b-tagging...

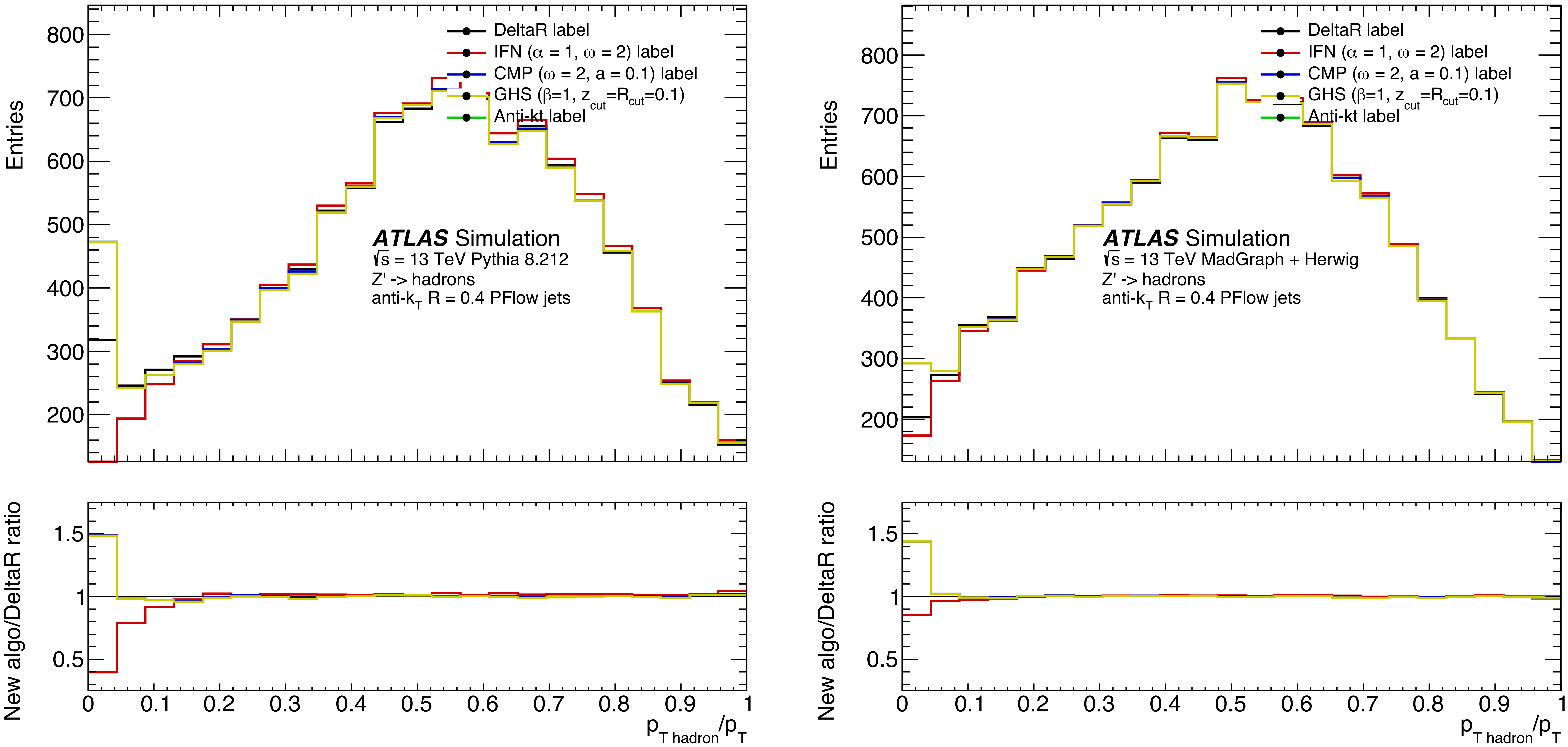
(but is dominated statistically by "good" b-jets)





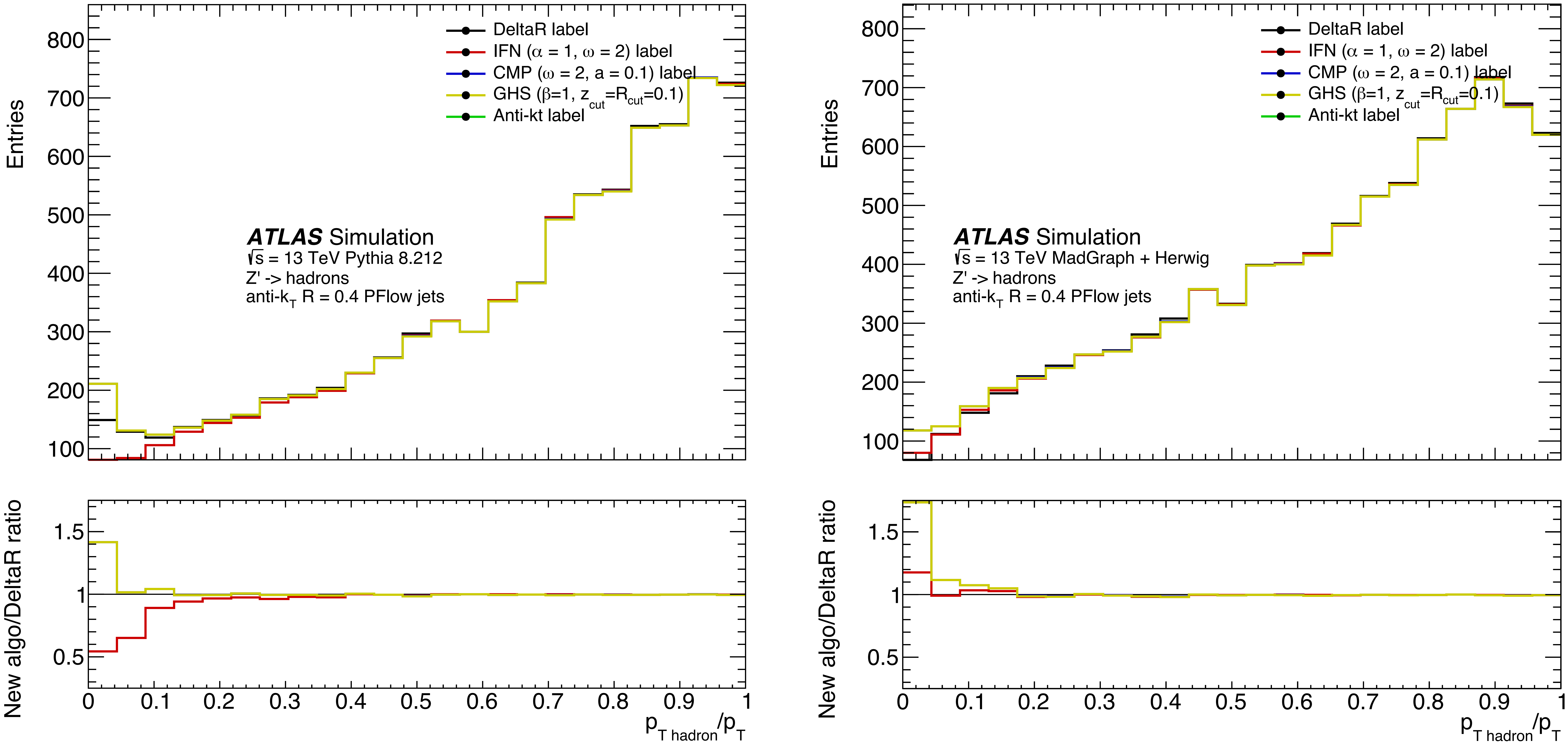
Results: Z', comparison with Herwig

For jets with 1 ATLAS matched c-hadron, we can compute $p_{Thadron}/p_T$:



Results: Z', comparison with Herwig

For jets with 1 ATLAS matched b-hadron, we can compute $p_{Thadron}/p_T$:



Summary

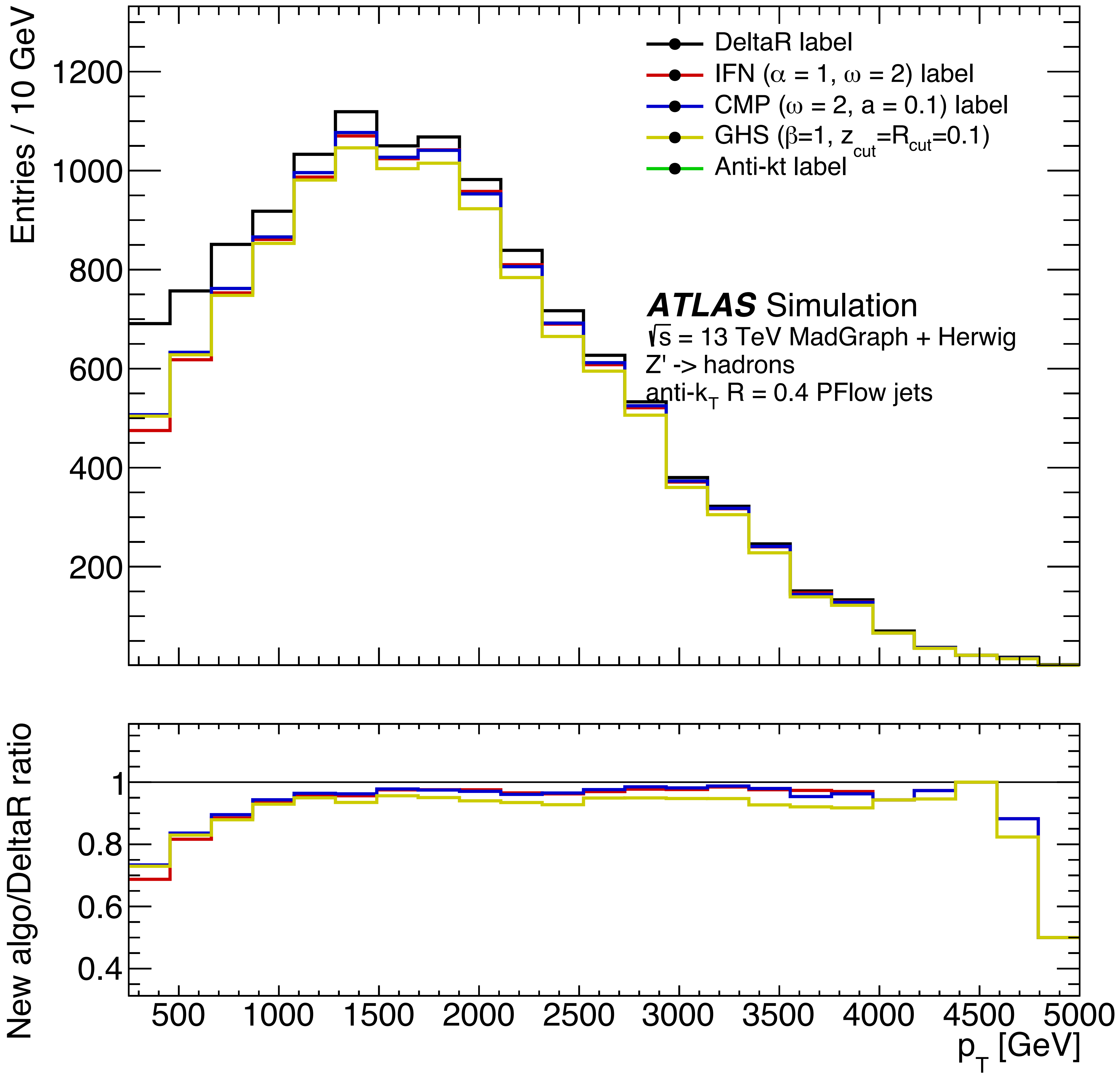
- Flavour dressing, flavour anti-k_T and IFN were implemented in ATLAS software
- The results for low-p_T tagging only slightly deviate from ATLAS labels at high-pt, due to collinear splittings inside the jet
- For high-p_T samples, we see large deviations at the lower end of the p_T spectrum
- The deviations are between ATLAS labels, and the algorithms themselves (IFN deviates)
- The reason is low-z hadrons being neutralised by IFN more often
- Event generators show large differences where the algorithms disagree
- The origin of heavy flavour jets at the lower end of the Z' events p_T spectrum is to be determined

Thank you!

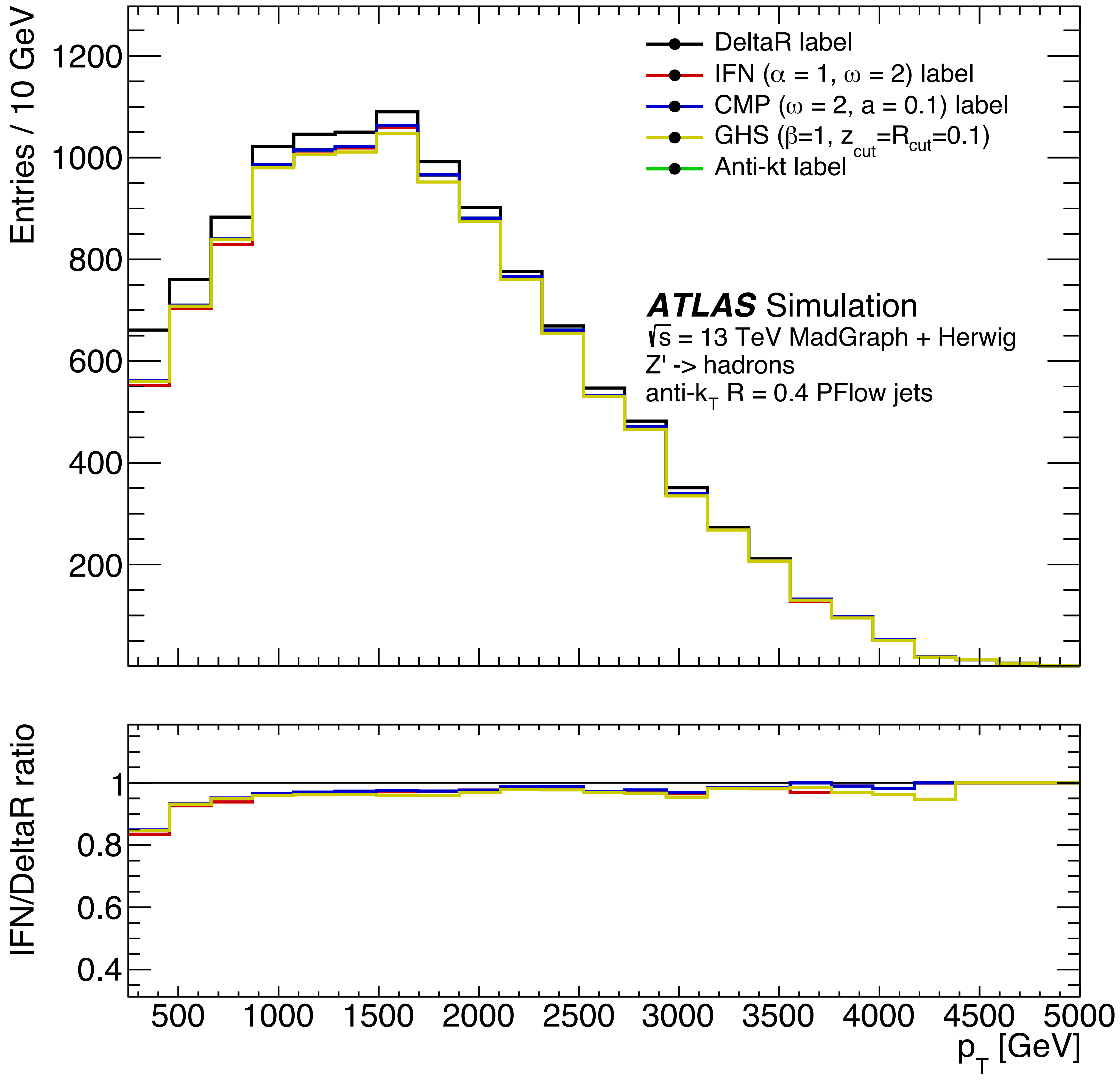
Backup

Herwig Z' distributions

c-tagging

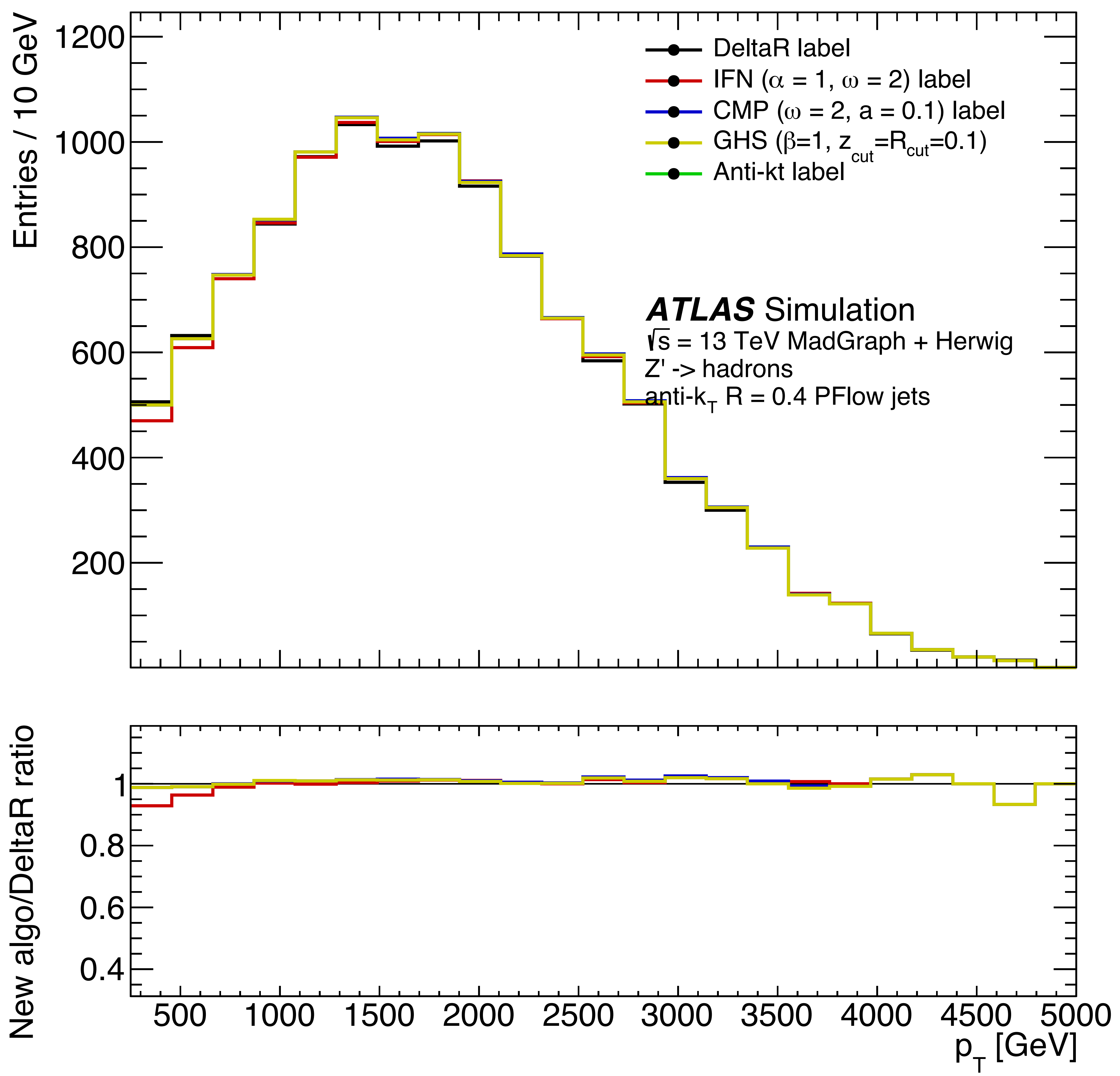


b-tagging

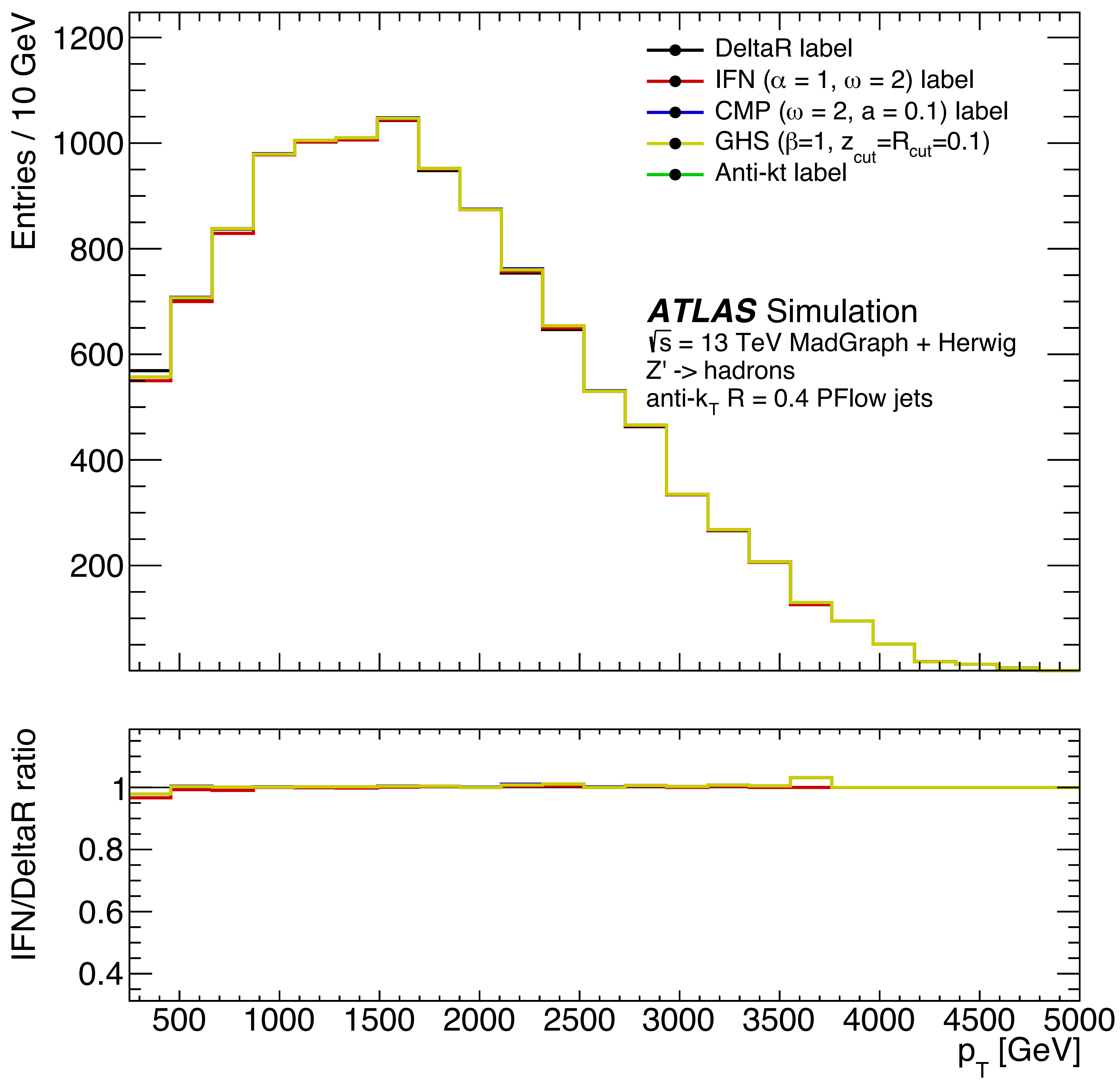


Herwig Z' distributions

c-tagging with double ctag veto



b-tagging with double btag veto



Technical details

Truth labelling happens during DAOD production. A labeling tool's life:

- The actual tool goes to:
  athena/PhysicsAnalysis/AnalysisCommon/ParticleJetTools/
  with an entry in python/ParticleJetToolsConfig.py
- Needs to be defined as a JetModifier in
  athena/Reconstruction/Jet/JetRecConfig/python/StandardJetMods.py
- Then, can be retrieved in the DerivationFramework, say in FTAG1, just like the other jet modifiers (see my implementation here https://gitlab.cern.ch/ragrabar/athena)
- I have the documentation for the code ready, available on request

Technical details

For now, massive hacks were needed to use FastJet modified algorithms in Athena, such as compiling FastJet with a different name to make things work with an older version of Fastjet used by Athena in other parts of the Derivation.

```
#include "radekjet/PseudoJet.hh"
#include "radekjet/JetDefinition.hh"
#include "FlavNeutraliserPlugin.hh"

 #include <iostream>
 #include <fstream>

using namespace std;
using namespace xAOD;
using namespace radekjet;
using namespace radekjet::contrib;
```

Best way out: the algorithms become a part of fjcontrib .

I have all the tools needed to move all algorithms to fjcontrib

- but did not manage to finalise this yet...

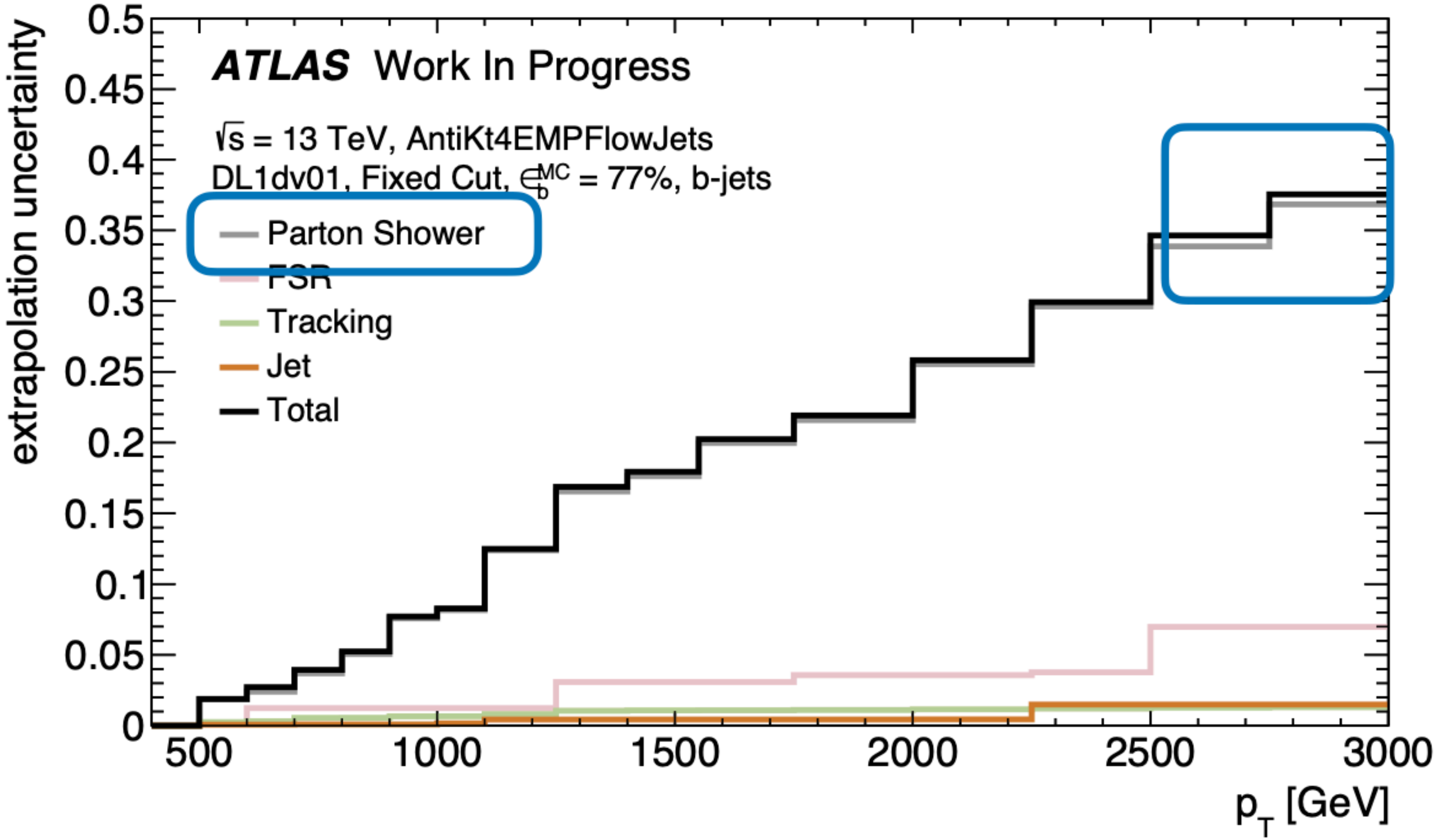
(my supervisor is one of the creators of fastjet so we have a direct contact there)

High p_T aspect

- What do the secondary vertices for "mistagged" jets look like in the high p_T region? Are they individually resolvable?

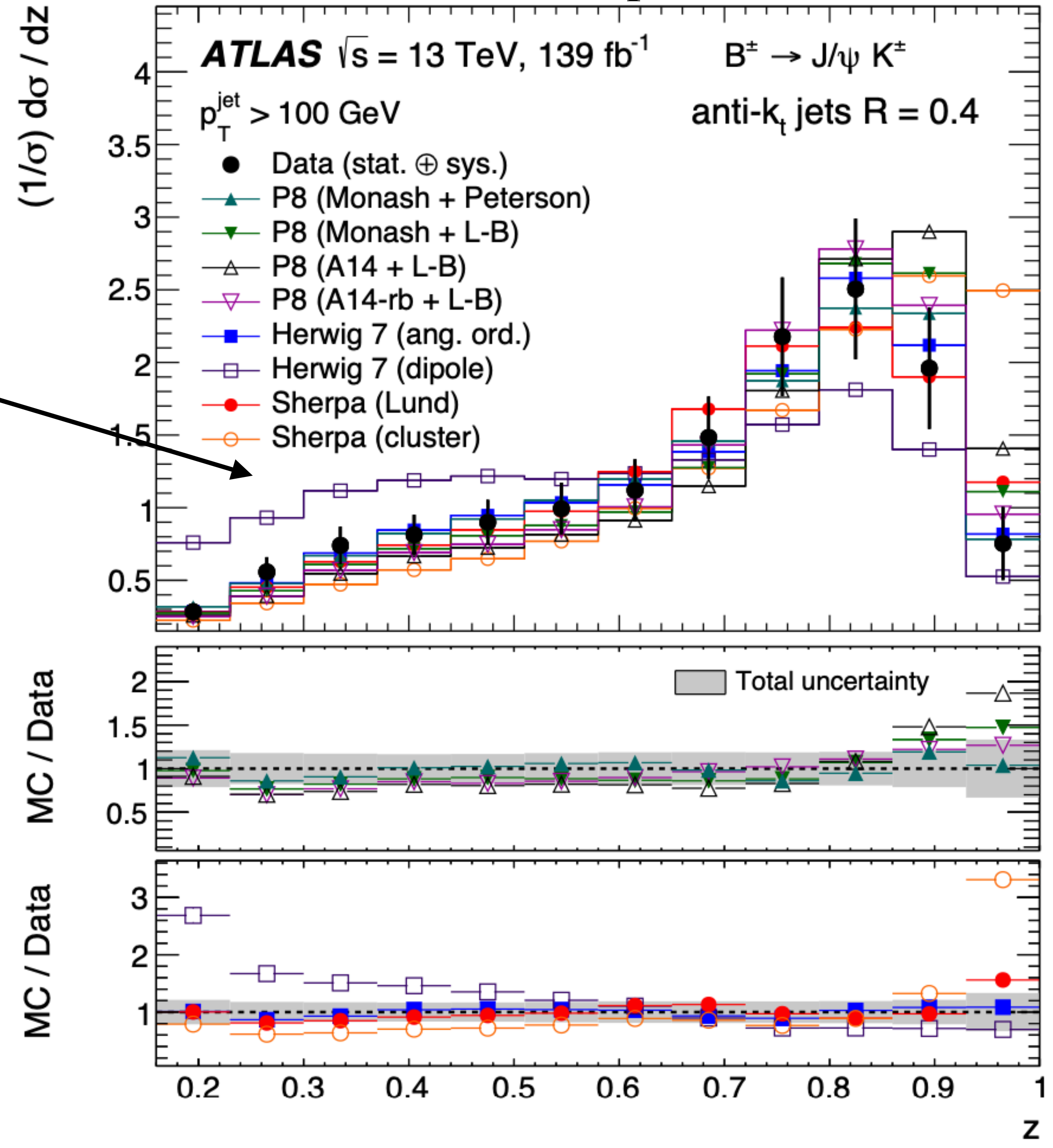
- How much do the "mistagged" jets contribute to high-p_T tagging systematics?

From Tom Dingley's AQP summary:



Low z, could be gluon splittings!

(2108.11650, $p_T > 100\text{GeV}$)

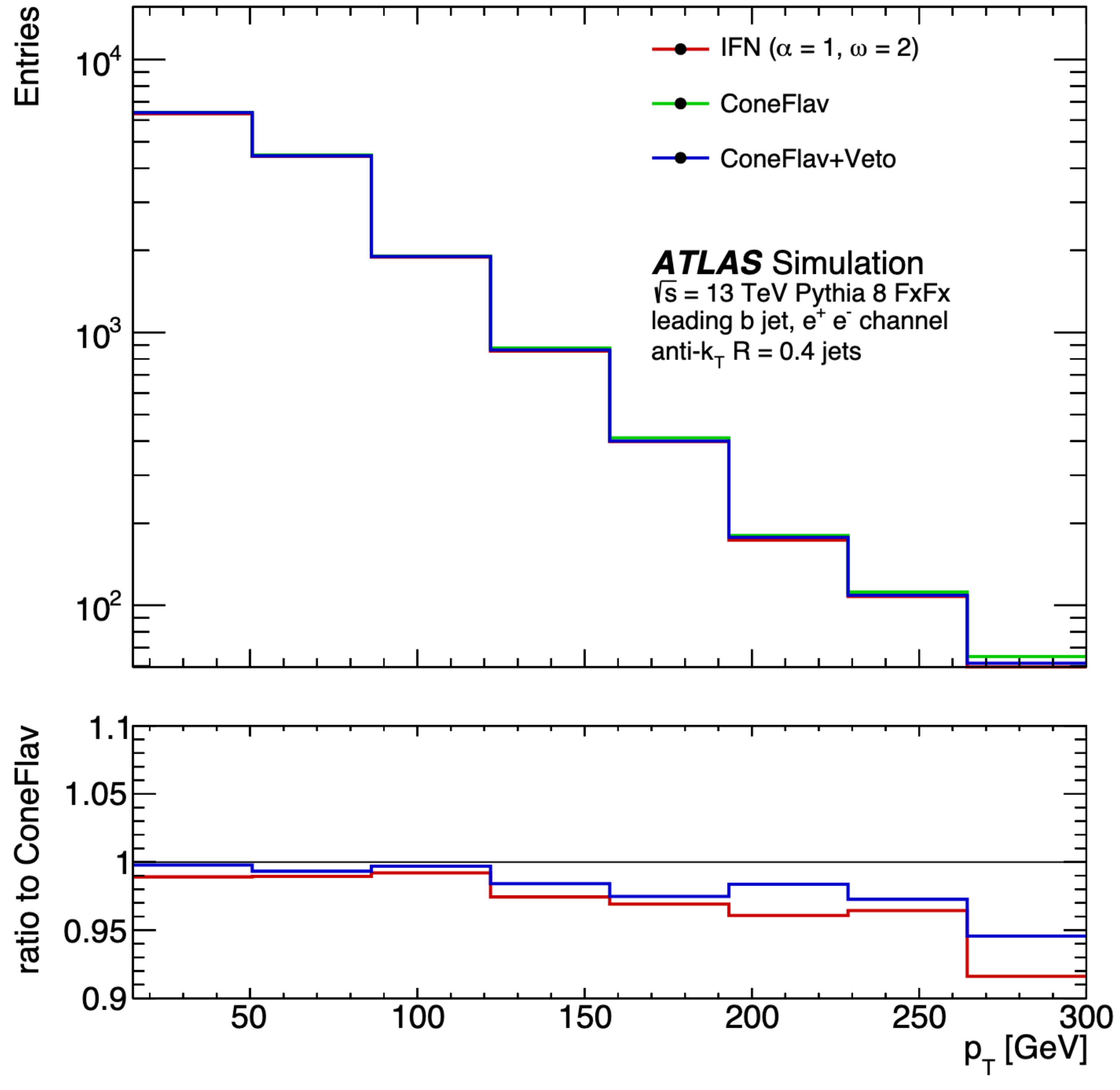


Some results: b tagging in $Z + b\bar{b}$

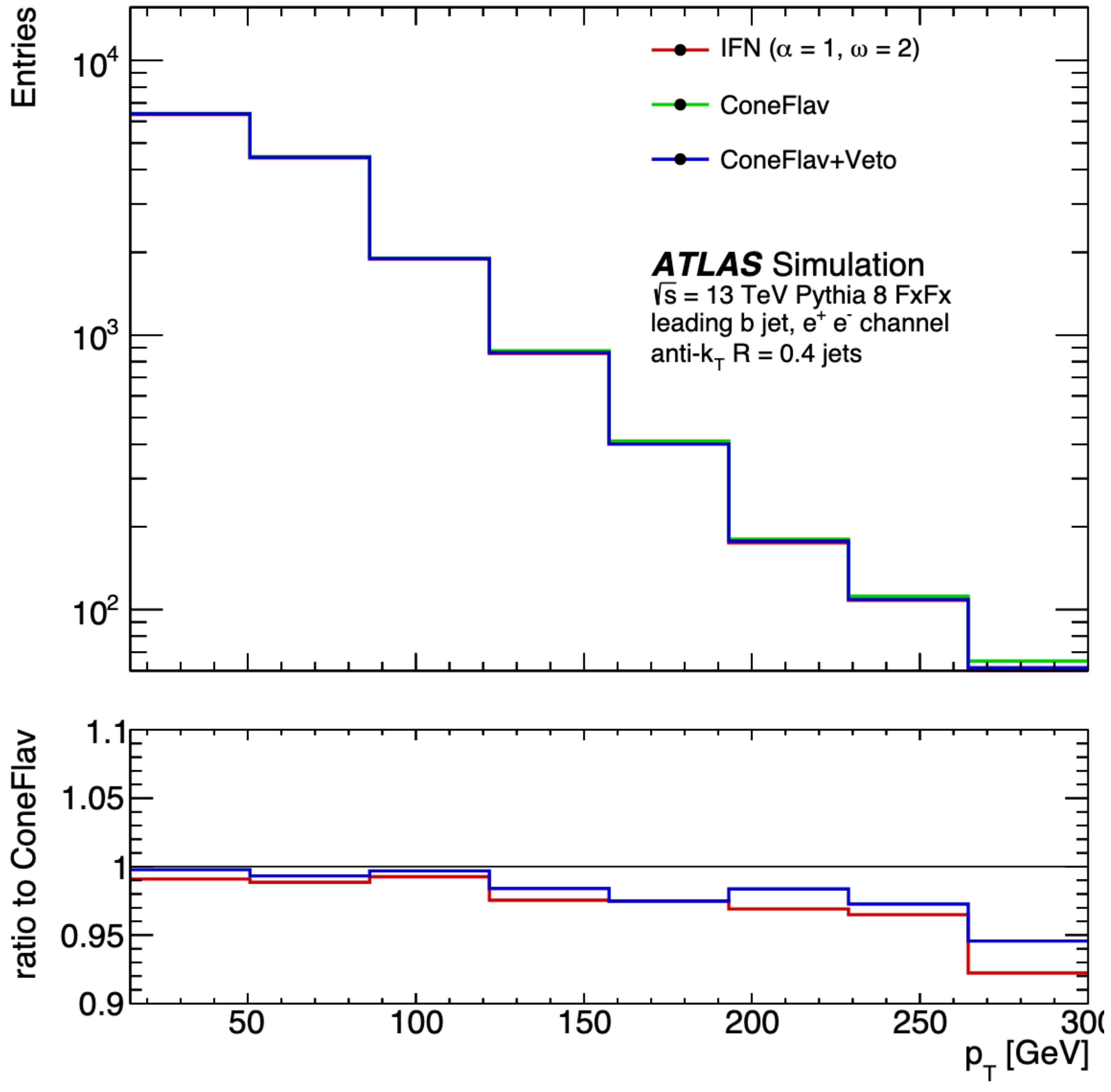
Tried 0 GeV, 5 GeV, 10 GeV cuts on B hadrons:

(Leading b-jet p_T)

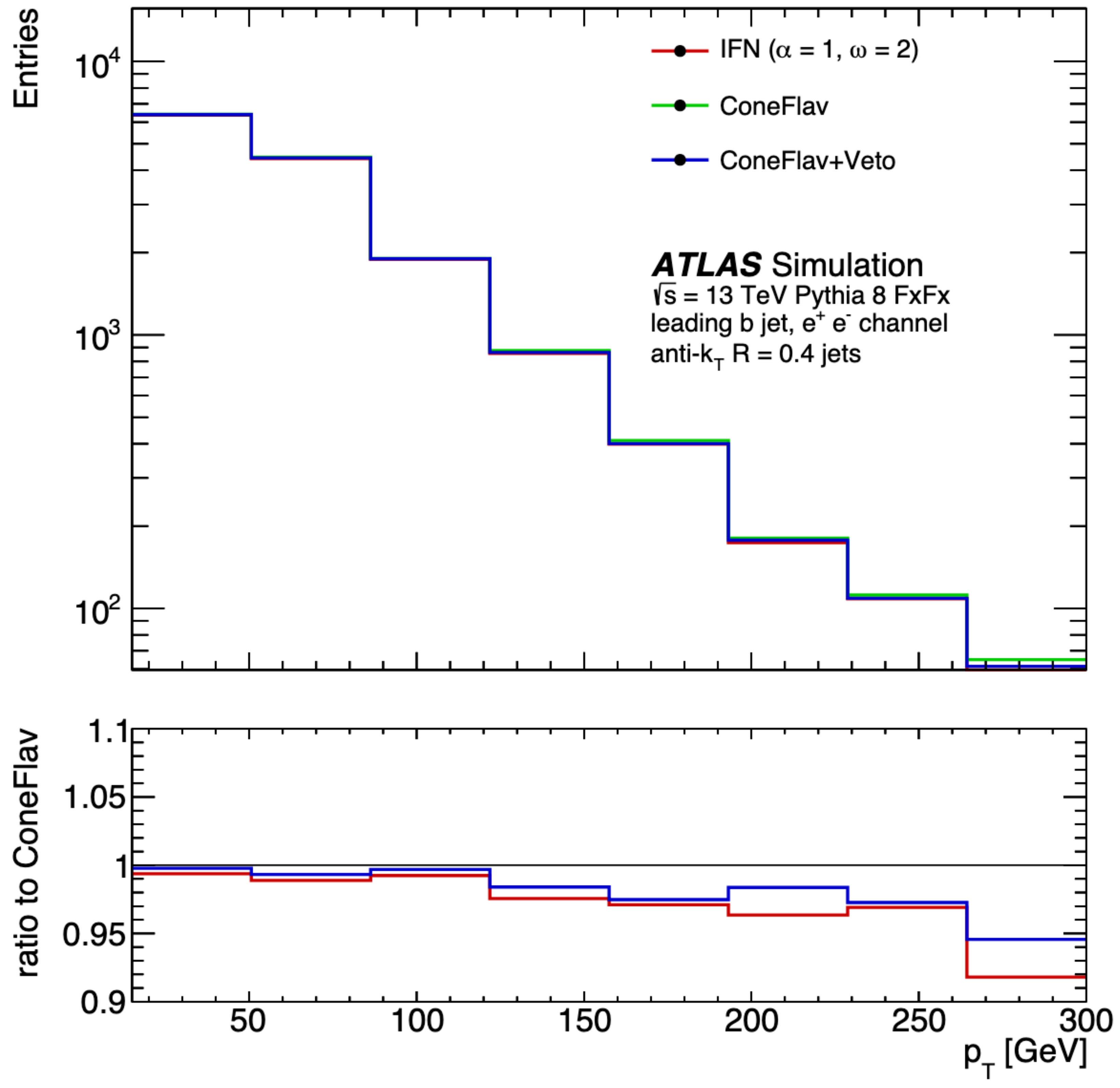
no b-hadron p_T cut



b-hadrons with $p_T > 5\mathrm{GeV}$



b-hadrons with $p_T > 10\mathrm{GeV}$



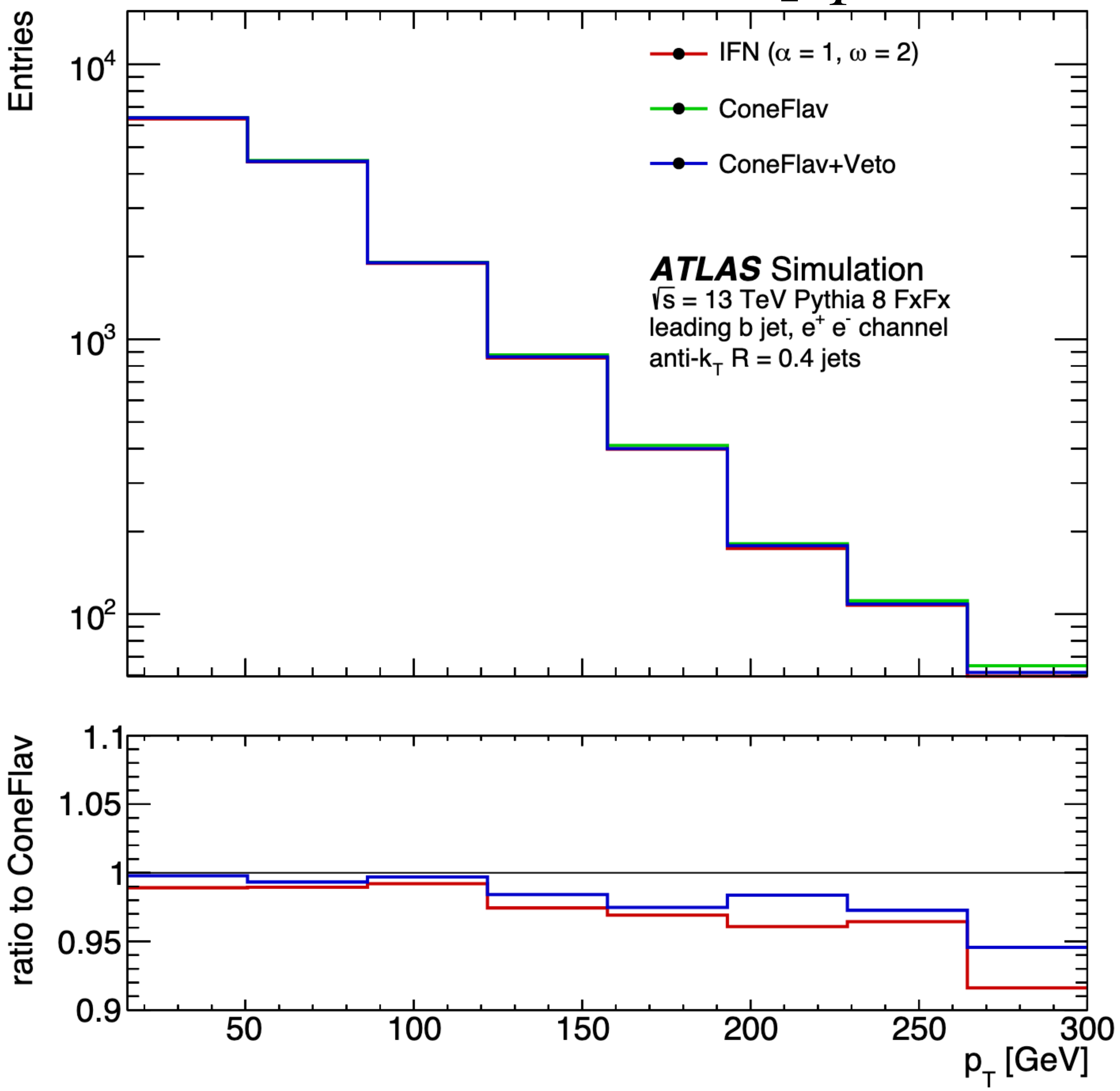
Switch to these algorithms ≈ getting rid of high p_T double b-tags

Example: $Z + b\bar{b}$ analysis

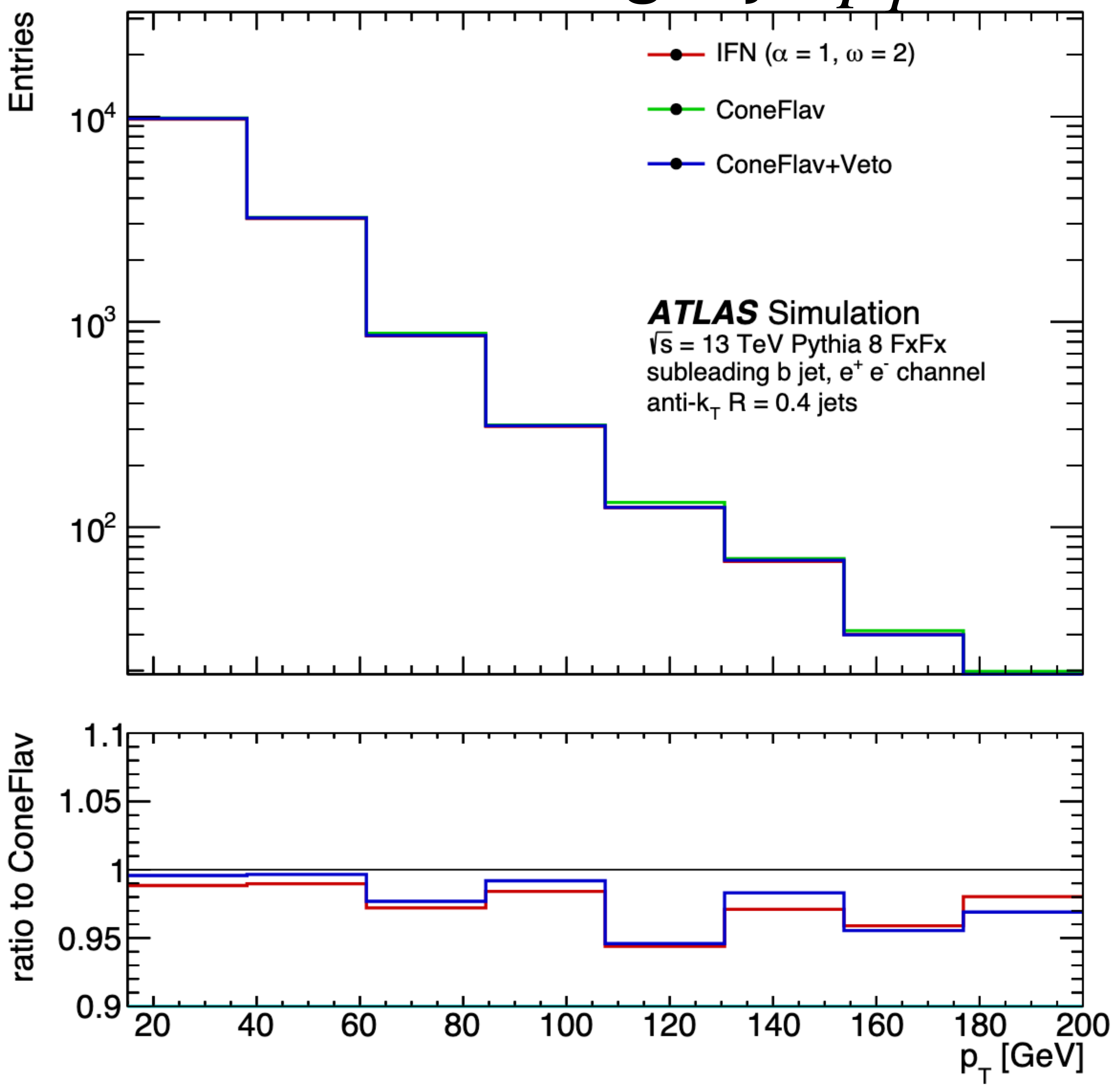
A double b-tagged jet unlikely in the "2 resolved b-jets" case

($Z + b + (g \rightarrow b\bar{b})$ is a very unlikely final state)

Leading b-jet p_T



Subleading b-jet p_T



Practicalities: Athena implementation

In FastJet, jet flavour algorithms modify the internal "jets" array + "user_info" dynamically (a must-have for quick implementations using nnh)

2 May 2023: fastjet-3.4.1 (manual, doxygen, fjcore).

- New features:
  - JetDefinition has a new clustering strategy, N2PlainEEAccurate, intended to work with the ee_kt_algorithm and ee_genkt_algorithms. It replaces the dot-product evaluation of (1-cos(theta)) with a more accurate cross-product evaluation when two particles are close in angle (~10^{-4}). It has a timing penalty of ~15% but can calculate inter-particle angles down to ~epsilon rather than sqrt(epsilon), where epsilon is machine precision. This work has been ported from PanScales.
  - two new functions to make it possible for plugin algorithms to modify user info, as needed for modern flavoured jet-clustering algorithms:
    - PseudoJet::set_user_info_shared_ptr(...), to allow user info to easily be copied from one jet to another and
    - ClusterSequence::plugin_non_const_jet(...) to get non-const access to the initial jets.
- Bug fixes:
  - Resolved MAJOR BUG that arose with full thread-safety enabled (reported by Ludo Scyboz), where PseudoJet::reset_momentum and PseudoJet::operator= failed to update internally cached values of phi() and rap(). In some instances related issues could also lead to race conditions.
  - eliminated NaN from square-root of negative mean areas in background estimation (now returns zero)
- Other changes:
  - auto_ptr interface is now disabled by default (deprecated since c++11, removed in c++17); can be manually enabled at configure-time with --enable-auto-ptr

where all FastJet magic happens

http://fastjet.fr/all-releases.html

Needed two new functions to be defined, only available in 3.4.1 (released halfway through this QT)

It seems that Athena >23.0 is compiled with fastjet >3.4.1! (not sure when this changed)