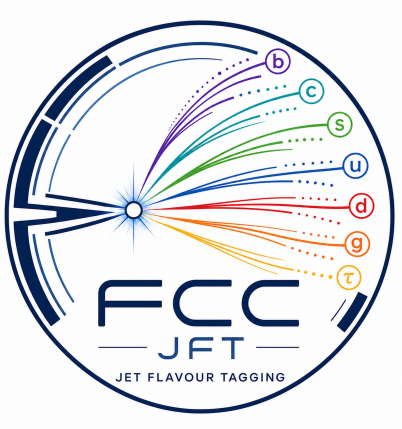
ATLAS-Style Truth Labelling For FCC-ee Jet Flavour Tagging



Alexander Froch, Tobias Golling, Vincent Riechers
Université de Genève

07.10.2026 - Analysis Tools & MC Productions

CHEF Swiss High Energy Physics for FCC

Computing

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

Today the jet flavour comes from the input file name

In examples/FCCee/weaver/stage2.py every jet in an Hbb file is a b-jet. There is no per-jet truth, and no truth information at all for the jet constituents, which the auxiliary tasks of a modern tagger need.

20%

of the jets in the light samples

are really gluon jets, but are trained on as u, d or s

7% c, 3% b

inside the Hgg sample

real heavy flavour from gluon splitting, labelled as gluon

none

of the ATLAS constituent labels

no truth origin, type, source or vertex index per constituent

Fractions from 5000 events of each pre_summer2026 IDEA wzp6 vvH sample

Ghost association: let the clustering decide

Truth particles are added to the jet clustering at a negligible momentum, 10^{-18} of their own, and we read back which jet each of them ended up in. No cone, no ΔR matching, no assumption about the jet algorithm.

`step 1`

Scale down

hadrons, partons and taus enter as ghosts at 1e-18 of their momentum

`step 2`

Recluster

with the same clustering functor the analysis already uses

`step 3`

Map back

jets are identified by their shared real constituents, not by their order

`step 4`

Label

each jet gets the ghosts it contains, outside every jet gives -1

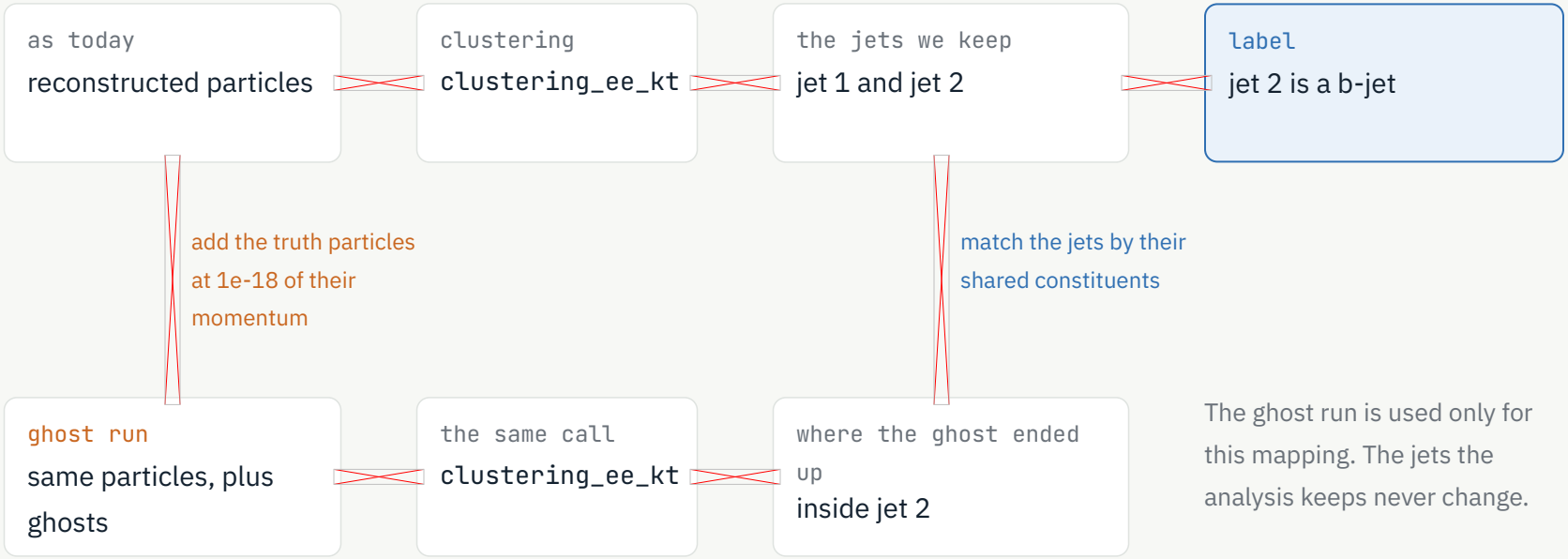
Works with any JetClustering::clustering_* functor, inclusive or exclusive

Jet axes are unchanged: the measured shift is 0.000e+00

Every b and c hadron ghost ended up in a jet in the checks

PR #549 · JetGhostLabels::associate_ghosts

How ghost association works



The ghost run is used only for this mapping. The jets the analysis keeps never change.

The jet axes do not move: the measured shift is 0.000e+00 · PR #549, JetGhostLabels::associate_ghosts

Six small PRs, stacked in merge order

PR	What it adds	New code	State
#549	Ghost association of truth particles to jets	JetGhostLabels	ready for review
#550	Hadron ghost labels, standard and extended	TruthLabelUtils	draft, after #549
#551	Parton label from the leading ghost parton	JetGhostLabels	draft, after #550
#552	Truth labels for every jet constituent	ConstituentTruthLabels	draft, after #551
#553	Truth jets matched to reconstructed jets	TruthJets	draft, after #552
#554	All of it wired into the weaver example	stage1.py, config.py	draft, after #553

Each PR is reviewable on its own and says in its description which one it follows. Only the last one touches an example; the first five only add new analyzers. Unit tests ship with each: ctest 61/61 on the full stack.

github.com/HEP-FCC/FCCAnalyses · from the fork afroch/FCCAnalyses

Jet labels follow the ATLAS conventions

Branch	Values
jet_HadronGhostTruthLabelID	0 light, 4 c, 5 b, 15 tau
jet_HadronGhostExtendedTruthLabelID	adds 44 cc, 54 bc, 55 bb, 1515 tautau
jet_HadronGhostInitialTruthLabelID	same, from the production flavour before B mixing
jet_HadronGhostTruthLabelPdgId, Pt	the hadron that defines the label
jet_GhostBHadronsFinalCount, C…	number of ghost b and c hadrons in the jet
jet_PartonTruthLabelID	-1 none, 1 d, 2 u, 3 s, 4 c, 5 b, 21 gluon
jet_PartonTruthLabelPt, DR	pT of the parton and its ΔR to the jet

Taus are ghost-associated as well, and a c hadron coming from a b hadron decay is not counted as a separate c hadron. The hadron pT threshold defaults to 1 GeV. The hadron label alone cannot separate u, d, s and g, which is why the parton label is needed next to it.

PRs #550 and #551

ONE DECISION WORTH YOUR ATTENTION

The parton is found from the decay tree, not from the generator status

We take the last partons before hadronisation, meaning partons that have no parton daughter, and label the jet with the highest energy one associated to it.

Why not status 71 to 79

The status codes differ between productions. The Whizard and Pythia6 samples only use status 1 and 2, so the Pythia8 selection finds zero partons there and every jet would stay unlabelled.

How well the two agree

In a Pythia8 $Z \to b\bar{b}$ sample, where both definitions are available, the topological selection and status 71 to 79 agree for 93 to 96% of the partons.

So the definition is generator independent and we do not have to carry a per-production list of status codes in the framework.

Every jet constituent gets its own truth labels

Branch	Values
pfcand_truthOriginLabel	2 Primary, 3 FromB, 4 FromBC, 5 FromC, 6 FromTau, 7 OtherSecondary, -1 no MC link
pfcand_truthTypeLabel	0 NoTruth, 1 Other, 2 pion, 3 kaon, 4 lambda, 5 electron, 6 muon, 7 photon, signed by the charge
pfcand_truthSourceLabel	1 NotSecondary, 3 StrangeMesonDecay, 4 StrangeBaryonDecay, 5 GammaConversion
pfcand_truthVertexIndex	0 at the MC primary vertex, then 1, 2, … per secondary vertex, -2 no MC link
pfcand_truthPdgId	PDG id of the matched MC particle, 0 if there is no match

What these feed

the track origin, track type and vertexing auxiliary tasks of the tagger

In H→bb jets

49% Primary, 25% FromBC, 21% FromB, 4% OtherSecondary

Delphes caveat

no pile-up, fakes or hadronic interactions, so those ATLAS classes never appear

PR #552 · based on the origin and vertex labels by V. Riechers

Truth jets, matched one to one

How they are built

- clustered from stable visible MC particles, generator status 1, no neutrinos, no dressing
- with the same clustering functor as the reconstructed jets, not a fixed setup
- matched through the reco-MC link, taking the pairs that share the most energy first
- one to one, so two jets can never claim the same truth jet;

What you get per jet

- truthJetPt and truthJetEnergy, a real target for energy or pT regression
- truthJetDR between the two jets
- truthJetPurity, the jet energy fraction inside its truth jet
- truthJetCompleteness, the truth jet energy fraction the jet covers

99.7 to 99.95%
of jets are matched, across the seven samples

97 to 99%
have purity above 0.8 and completeness above 0.5

0.995 to 0.999
median reconstructed over truth jet energy for those jets

PR #553 · replaces get_genJetPt by V. Riechers, which took the nearest truth jet in ΔR

The labels find the decay flavour, sample by sample

Sample	Dominant hadron ghost label	Dominant parton label
Hbb	5 b, 98.3%	5 b, 90.8%
Hcc	4 c, 98.0%	4 c, 82.2%
Hss	0 light, 98.9%	3 s, 78.1%
Hdd	0 light, 99.0%	1 d, 77.7%
Huu	0 light, 99.0%	2 u, 77.6%
Hgg	0 light, 90.1%	21 gluon, 76.9%
Htautau	15 tau, 99.7%	-1 none, 100%

The remainder is physics
most of it is gluon jets, and in Hgg real c and b from gluon splitting

Htautau has no partons
these events contain no coloured partons, so -1 is correct, not a failure

Internally consistent
labels agree with the ghost hadron counts and PDG ids for every jet

5000 events of each pre_summer2026 IDEA wzp6 vvH sample

A full production already runs on the branch

input
EDM4hep
7 wzp6 samples

done
stage1
per event, labels in

next
stage2
flatten per jet

ours
converter
to h5

ours
training
UPP and Salt

On disk now

697 files, 289 GB, all seven samples, produced with the stacked branch on the Key4hep 2026-04-08 stack through SLURM. All 697 array tasks finished, no empty files, label fractions reproduce the small scale validation.

What it means for the PRs

The code has been exercised at production scale, not only on test files, and the 74 output branches have been checked per sample. Nothing in the first five PRs changes any existing output.

Produced on the UNIGE cluster from the branch behind PR #554

WHAT I AM ASKING FOR

Reviewers for the stack, starting with #549

A review of #549 first

It is out of draft and the other five sit behind it. Once the ghost association is agreed, the rest is mostly conventions.

A CI run

No pipeline has started on the PRs since 16 September. As a first time contributor I cannot trigger it myself.

Opinions on conventions

Especially the parton definition, the branch naming, and whether this should live in the FastJet addon or elsewhere.

Happy to split, rename or rebase whatever makes the review easier. The samples and the production are there, so we can rerun anything on request.

`github.com/HEP-FCC/FCCAnalyses/pull/549 … /554`