Jet Flavor Tagging Using Graph Neural Networks

Nilotpal Kakati, *Sam Van Stroud,*
Maxence Draguet, Gabriel Facini, Eilam Gross, Sebastien Rettie,
Tim Scanlon, Jonathon Shlomi

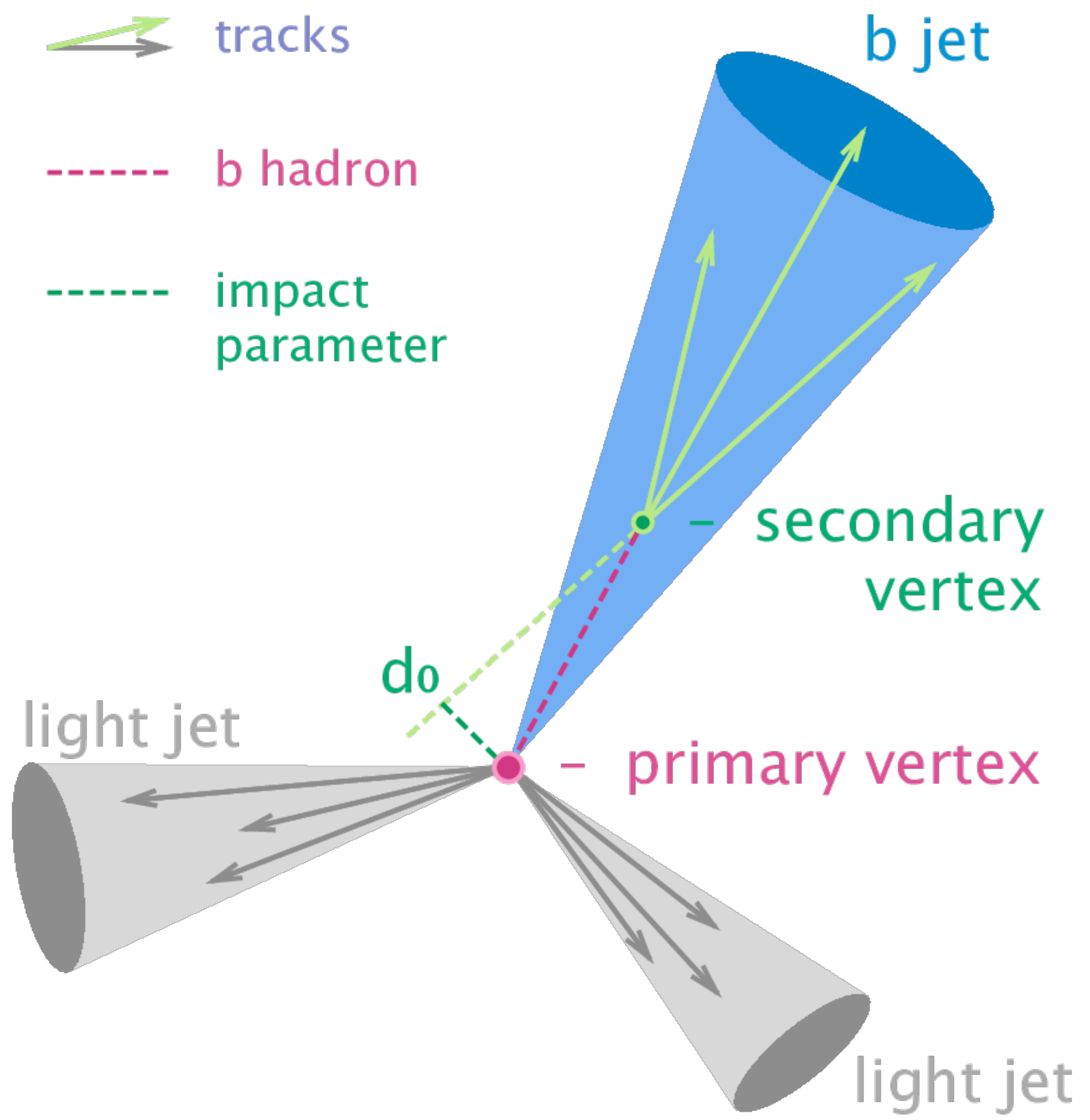
On behalf of the ATLAS collaboration

(nilotpal.kakati@cern.ch)

7th International CTD Workshop
1 June, 2022

Outlook

- Current ATLAS jet flavor tagging
- Motivation for a new tagger
- GN1: the new tagger
- Performance
- Summary

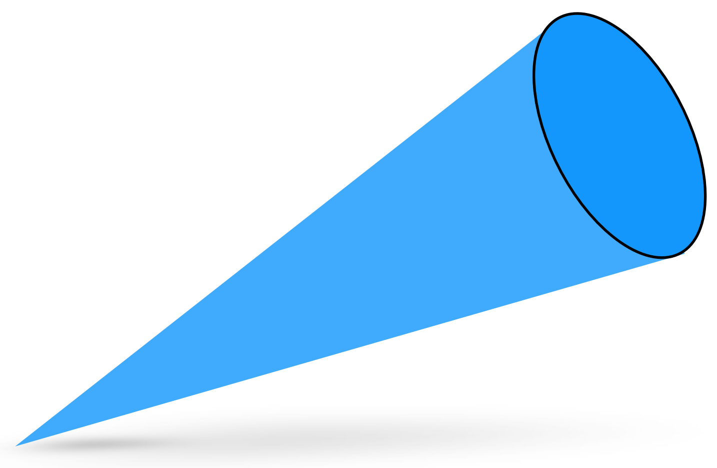


- Aim is to identify b,c and light flavored jets
- Main signatures of jets initiated by b-hadron decay -
  - Incompatibility of track with PV
  - Presence of secondary vertex

Jet Flavor Tagging in ATLAS

ATLAS EXPERIMENT

Jet



Low level vars (Jet, associated tracks)

Low level Taggers

(Impact parameter based)

IP2D | IP3D | RNNIP

(Secondary vertexing based)

Jet Fitter | SV1

And others…

Extracted Physics info

High level Tagger (NN)

DL1r

Probability of the jet being b/c/light flavored

(p_b, p_c, p_l)

$$D_b = \log \frac{p_b}{(1 - f_c)p_l + f_c p_c},$$

The low level taggers

- Low level taggers are important

Low level Variables (Jet, tracks) → ~~Phy info~~ → NN → p_b, p_c, p_u (Significant loss in performance)

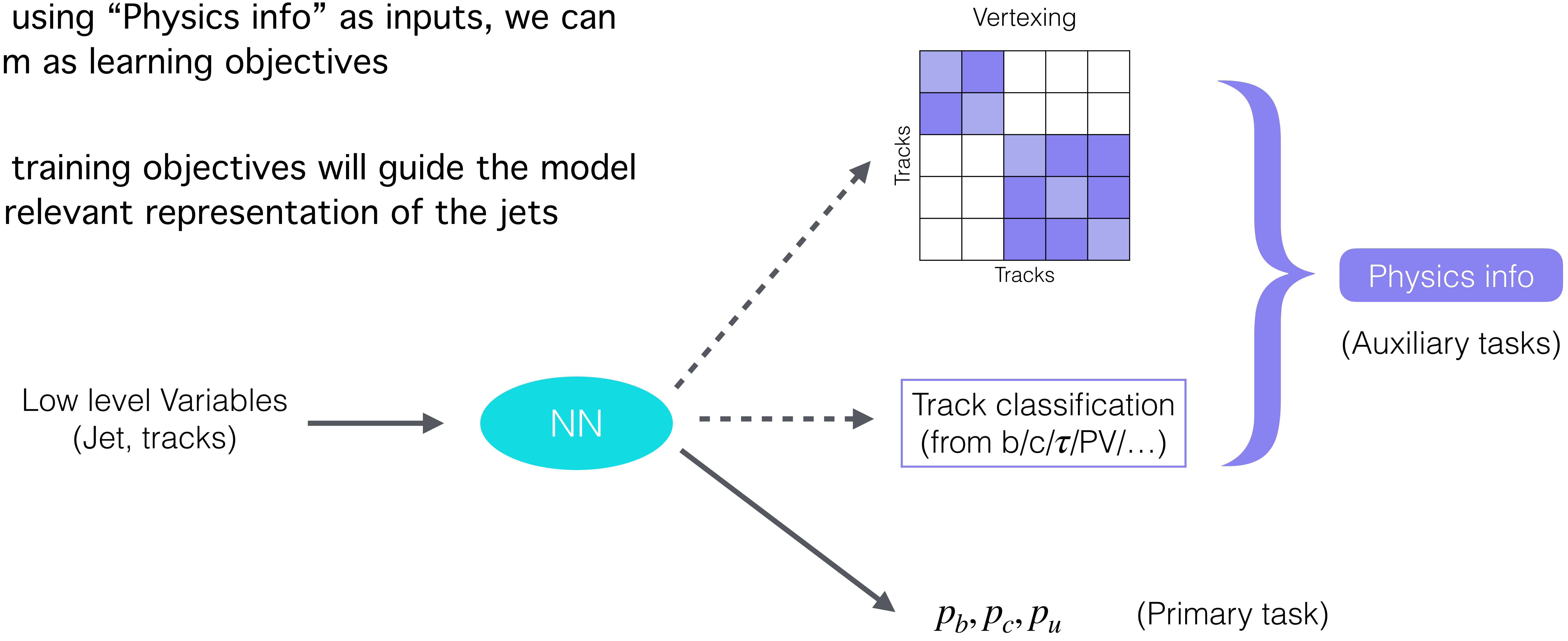
- An “all-in-one” jet tagger would be ideal, as -
    - It’ll remove the dependency on low level taggers
    - Easy to train, easy to maintain
    - Can be more easily optimized for a wide variety of use cases

- But in practice, it’s tricky (need the “Physics info” from the low level taggers!)

GN1 as a solution

- GN1 - a Graph Neural Network (GNN) based flavor tagging algorithm
- Instead of using “Physics info” as inputs, we can define them as learning objectives
- Additional training objectives will guide the model to learn a relevant representation of the jets

Proposed by Jonathan Shlomi et al.
https://arxiv.org/pdf/2008.02831.pdf

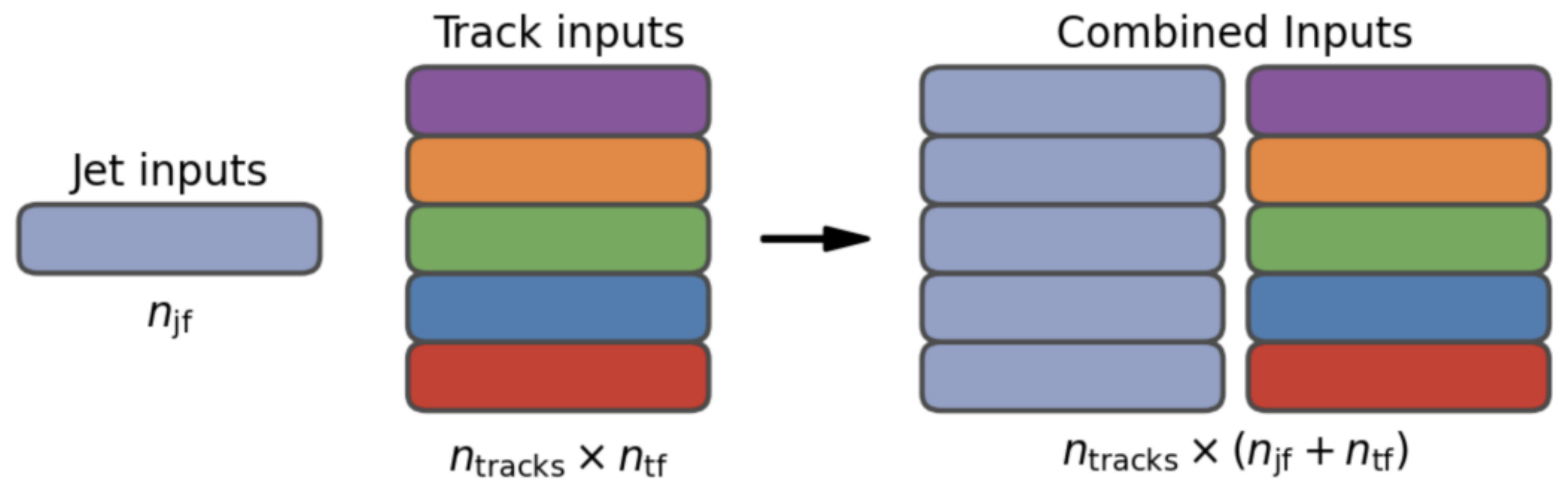


The input to GN1

Jet Input	Description
p_T	Jet transverse momentum
η	Signed jet pseudorapidity

Track Input	Description
q/p	Track charge divided by momentum
$d\eta$	Pseudorapidity of the track, relative to the jet η
$d\phi$	Azimuthal angle of the track, relative to the jet ϕ
d_0	Closest distance from the track to the PV in the longitudinal plane
$z_0 \sin\theta$	Closest distance from the track to the PV in the transverse plane
$\sigma(q/p)$	Uncertainty on q/p
$\sigma(\theta)$	Uncertainty on track polar angle θ
$\sigma(\phi)$	Uncertainty on track azimuthal angle ϕ
$s(d_0)$	Lifetime signed transverse IP significance
$s(z_0)$	Lifetime signed longitudinal IP significance
nPixHits	Number of pixel hits
nSCTHits	Number of SCT hits
nIBLHits	Number of IBL hits
nBLHits	Number of B-layer hits
nIBLShared	Number of shared IBL hits
nIBLSplit	Number of split IBL hits
nPixShared	Number of shared pixel hits
nPixSplit	Number of split pixel hits
nSCTShared	Number of shared SCT hits
nPixHoles	Number of pixel holes
nSCTHoles	Number of SCT holes
leptonID	ID of reconstructed lepton (if present)

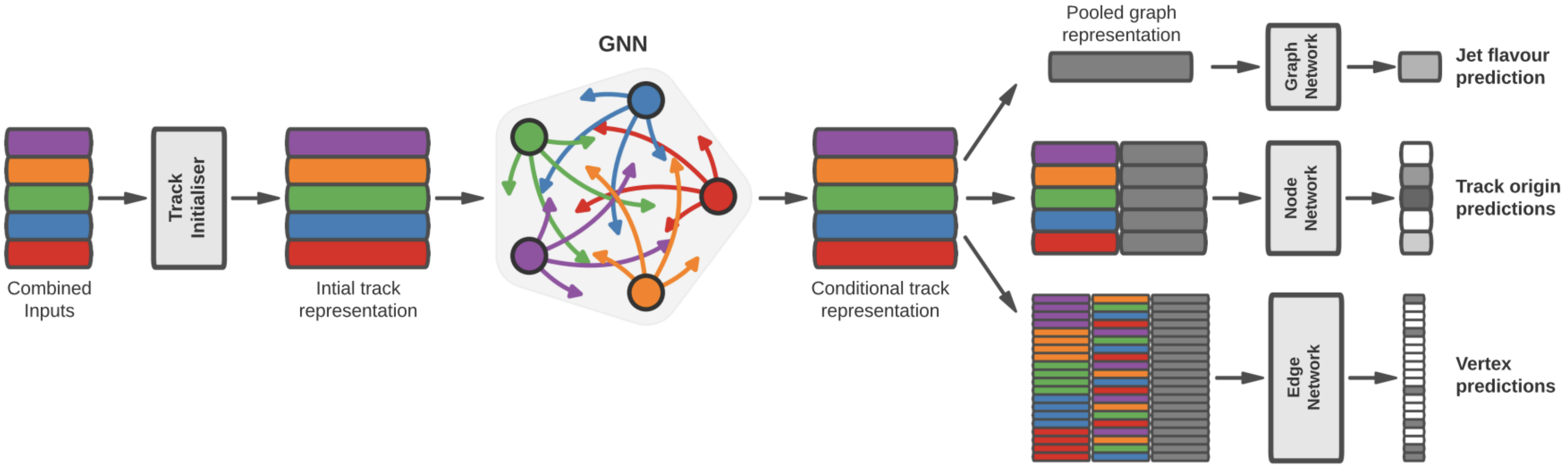
The jet inputs are concatenated with each track's input



n_{jf} = number of jet features, n_{tf} = number of track features
n_{tracks} = number of tracks

- leptonID - is the track used in reconstruction of an electron, muon or neither
- We'll look at plots for two models (later) -
  - GN1 (baseline, without leptonID)
  - GN1Lep (with leptonID)

The complete architecture (GN1)

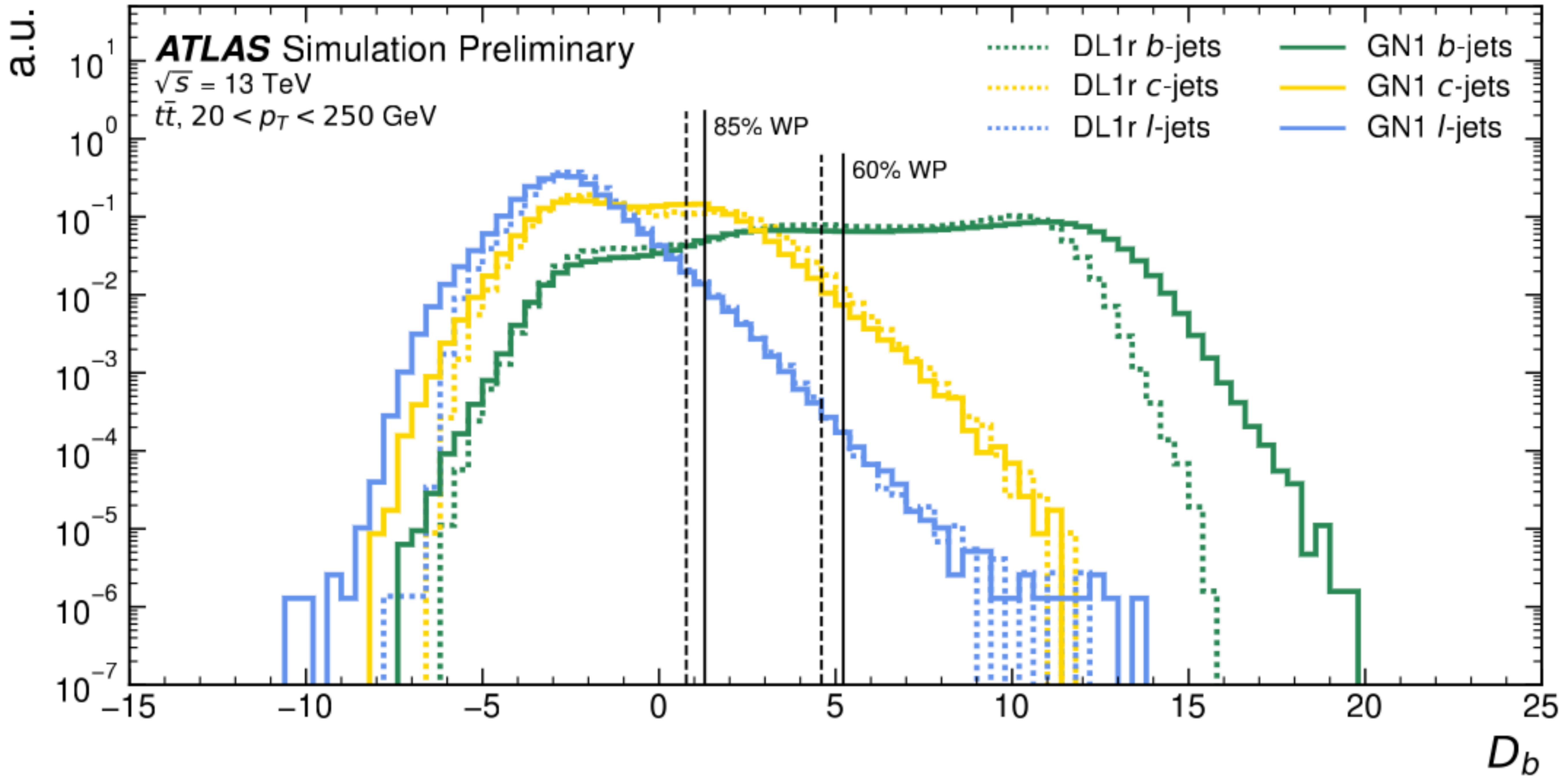


b-tagging performance

- The output probabilities of the model (p_b, p_c, p_l) are combined to form a discriminant D_b
- GN1 shifts the b-jet distribution to higher value, and c/light jet distributions are shifted towards lower value
- Enhanced b vs c/light separation with GN1

$$D_b = \log \frac{p_b}{(1-f_c)p_l + f_c p_c},$$

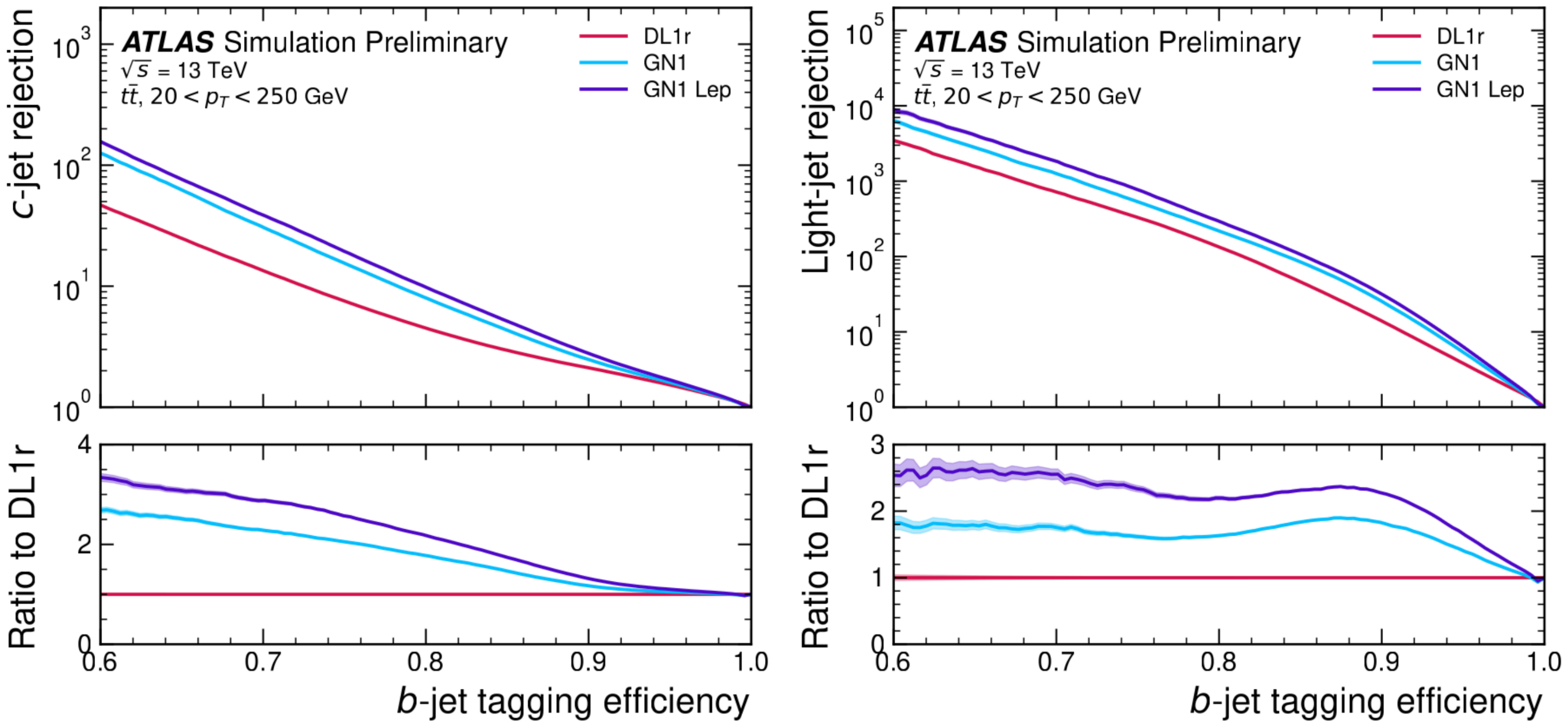
ttbar



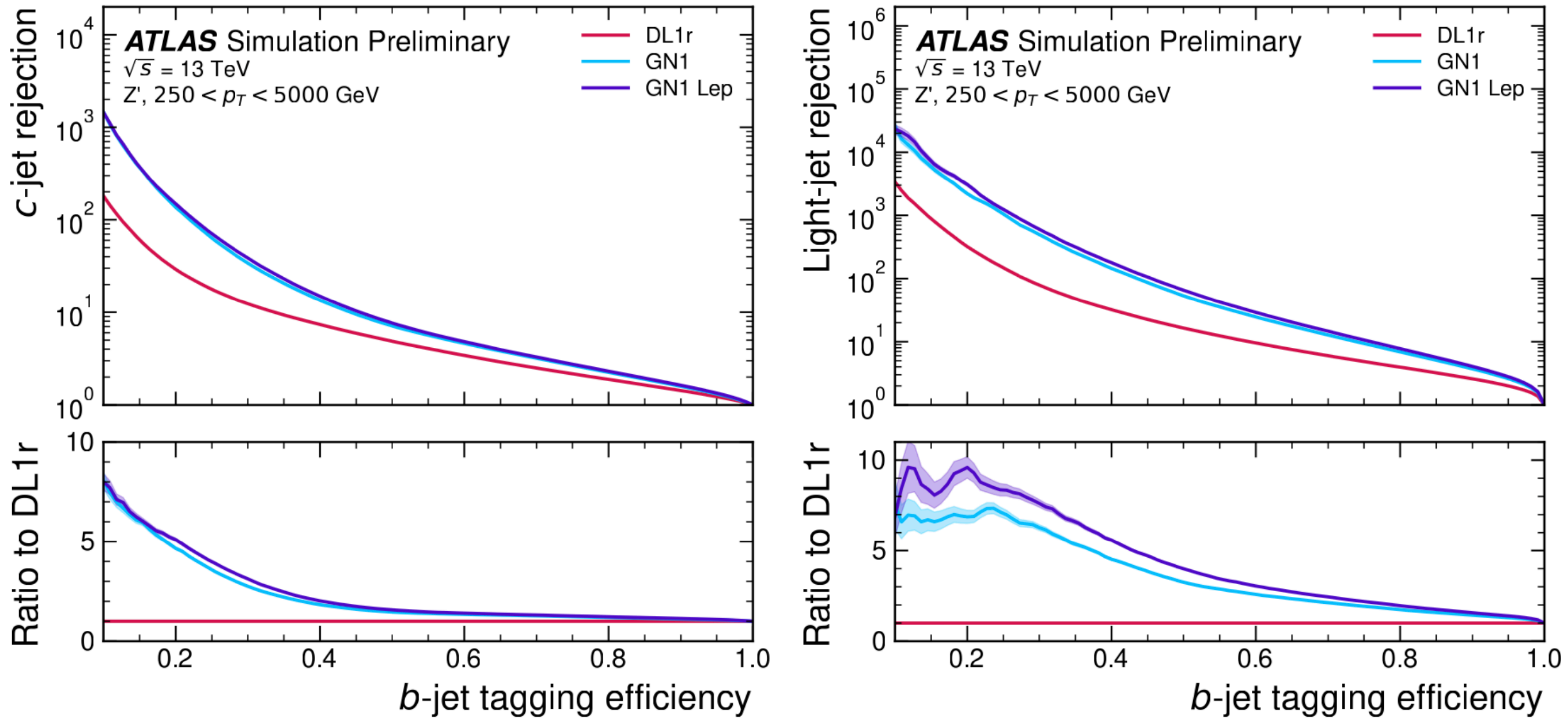
b-tagging performance

- Significant improvement in c/light rejection, for given b-efficiency
- For ttbar (a representative of SM processes, covers the low pT region), at 70% b-efficiency
  - c-rejection - 2.4x (GN1)
  - l-rejection - 1.7x (GN1)
- For Z’ (a representative of BSM processes, covers the high pT region), at 30% b-efficiency
  - c-rejection - 3x (GN1)
  - l-rejection - 6.2x (GN1)
- Including leptonID as a track input also shows further improvement (GN1Lep)

ttbar



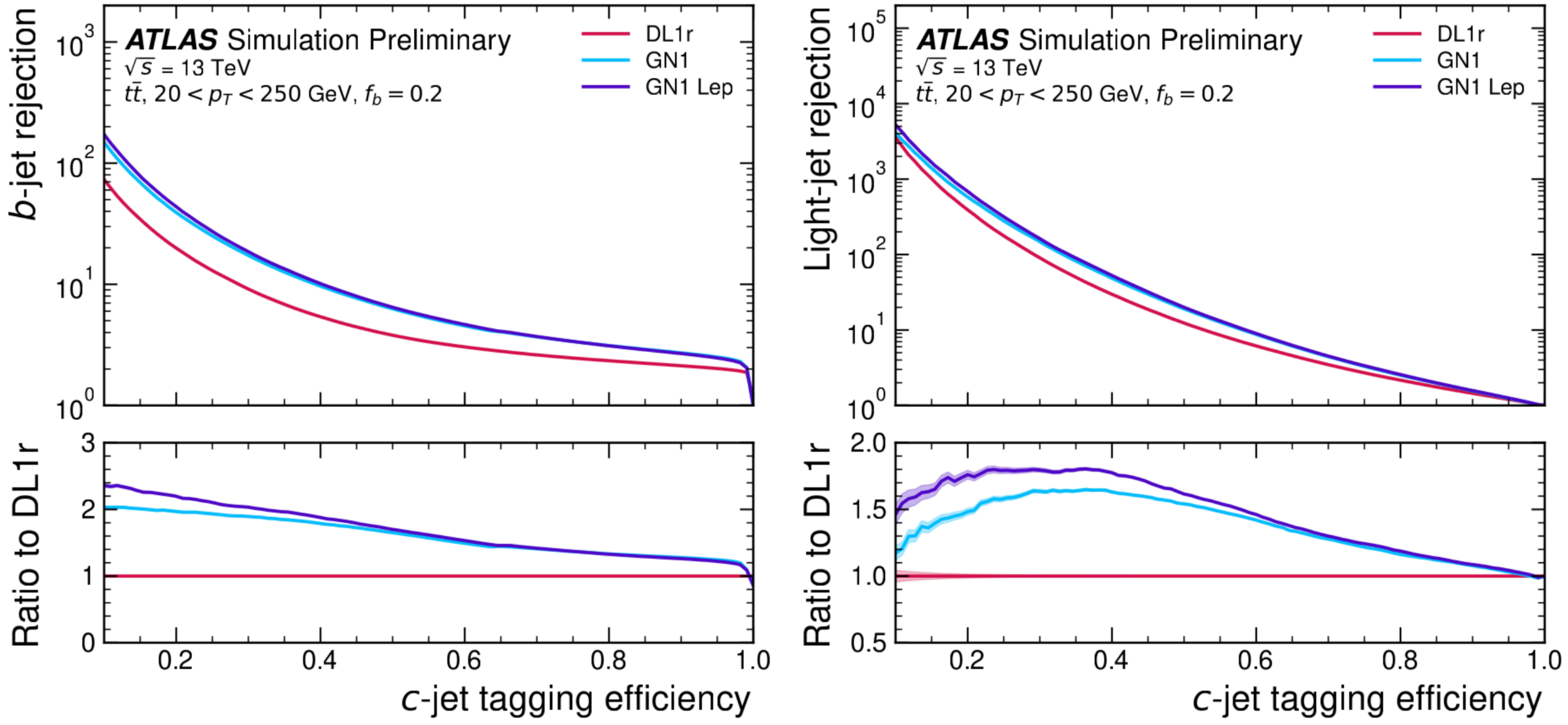
Z’



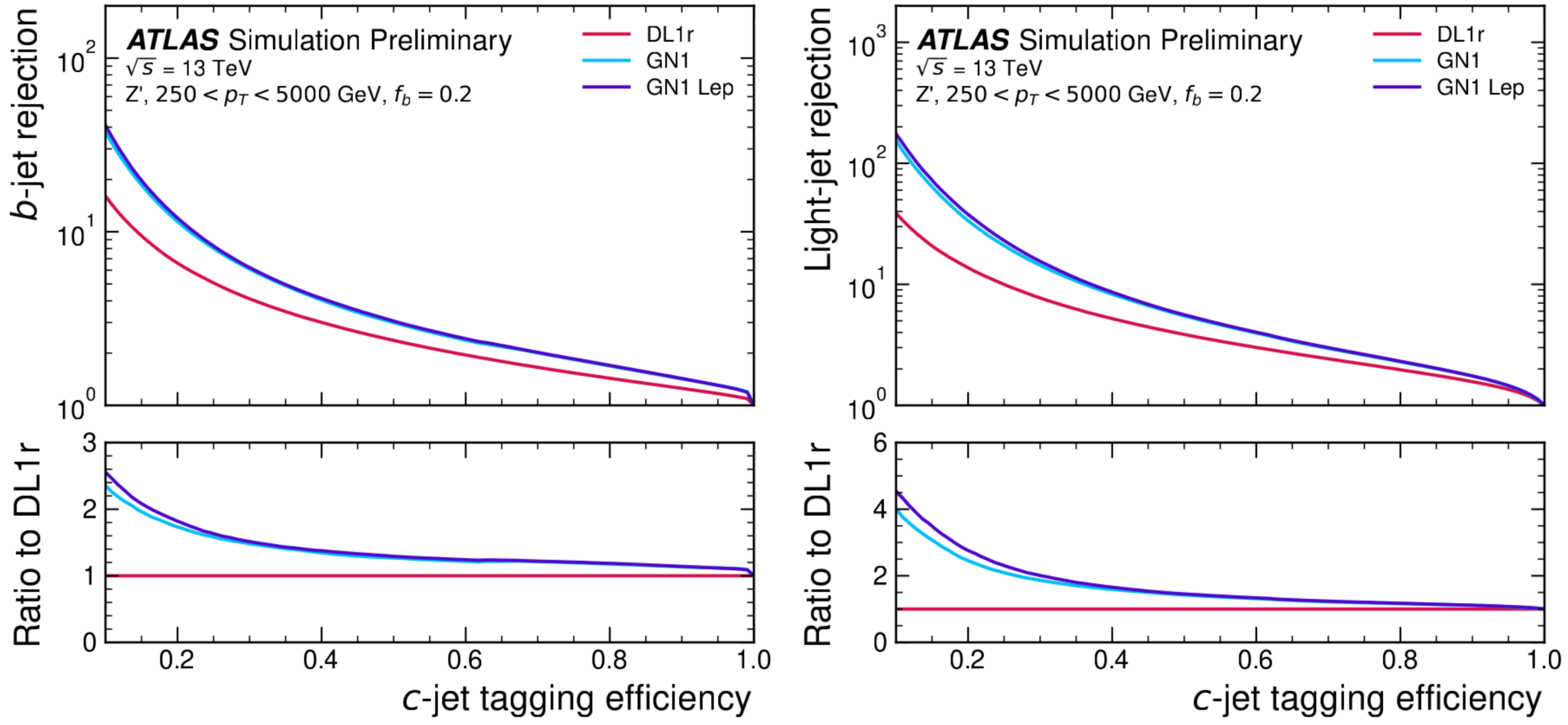
c-tagging performance

- $D_c = log \frac{p_c}{(1-f_b)p_l + f_b p_b}$
- For ttbar, at 25% c-efficiency
  - b-rejection - 2x (GN1)
  - l-rejection - 2x (GN1)
- For Z', at 25% c-efficiency
  - b-rejection - 1.6x (GN1)
  - l-rejection - 2x (GN1)

ttbar



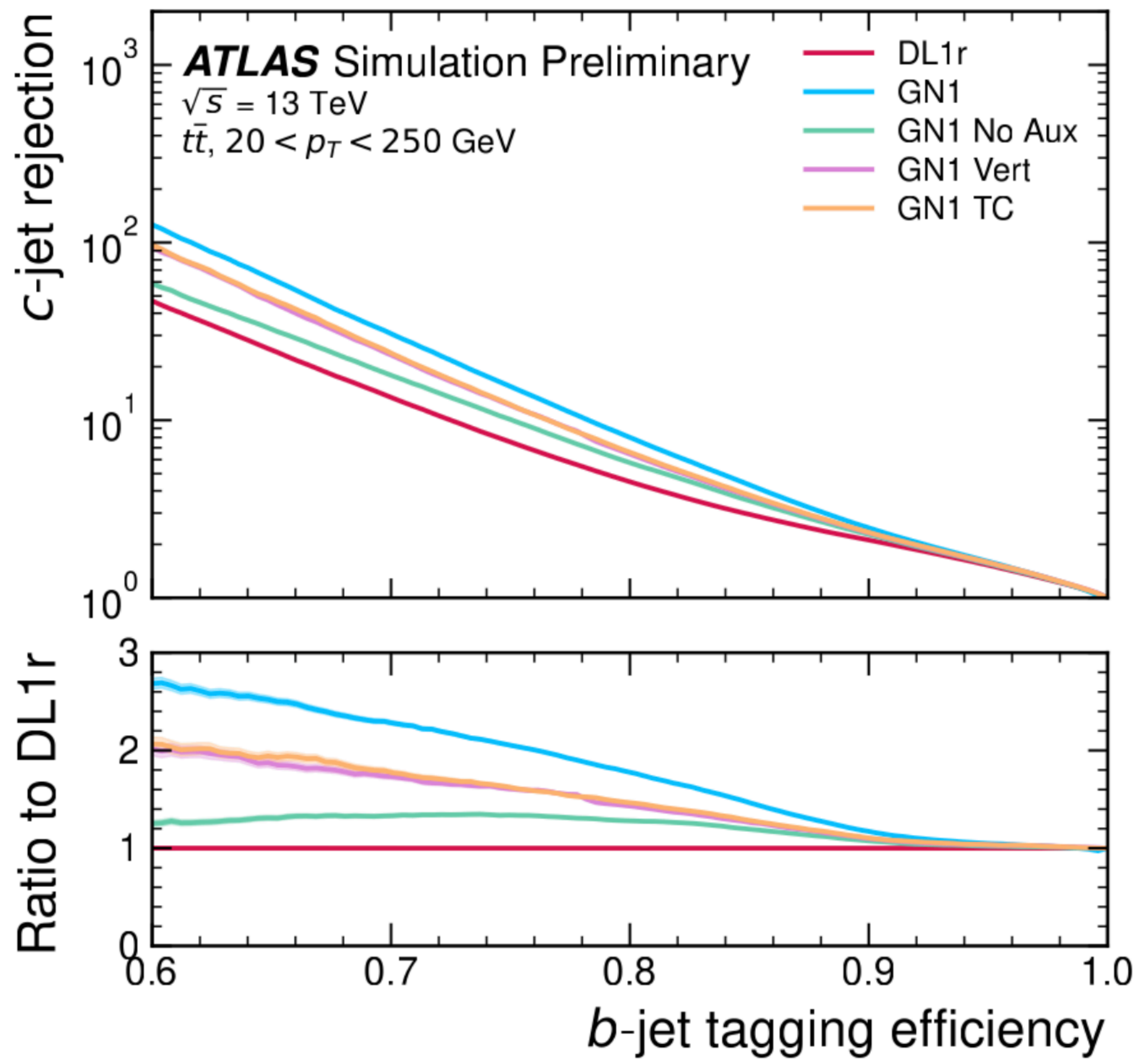
Z'

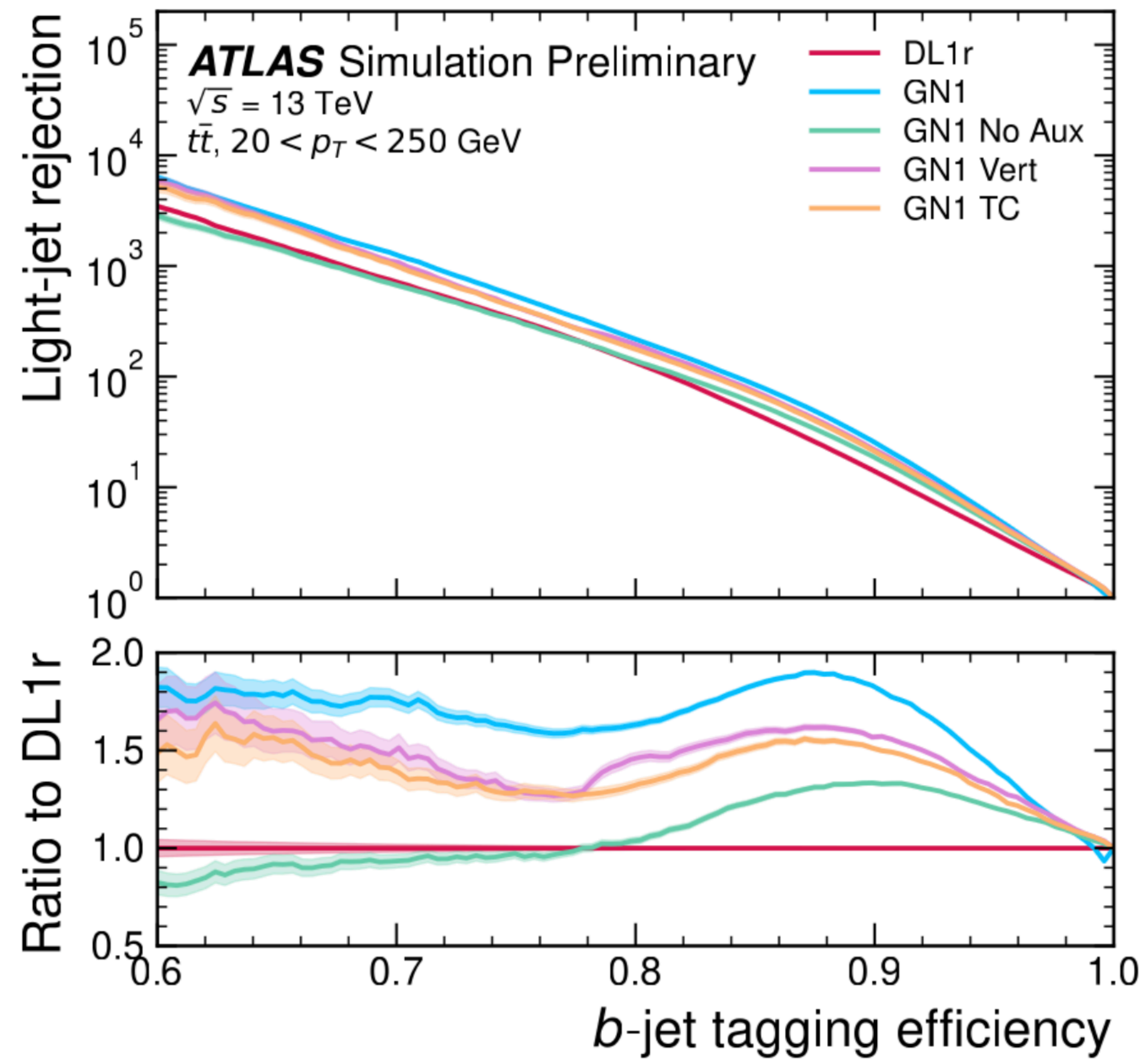


Ablations

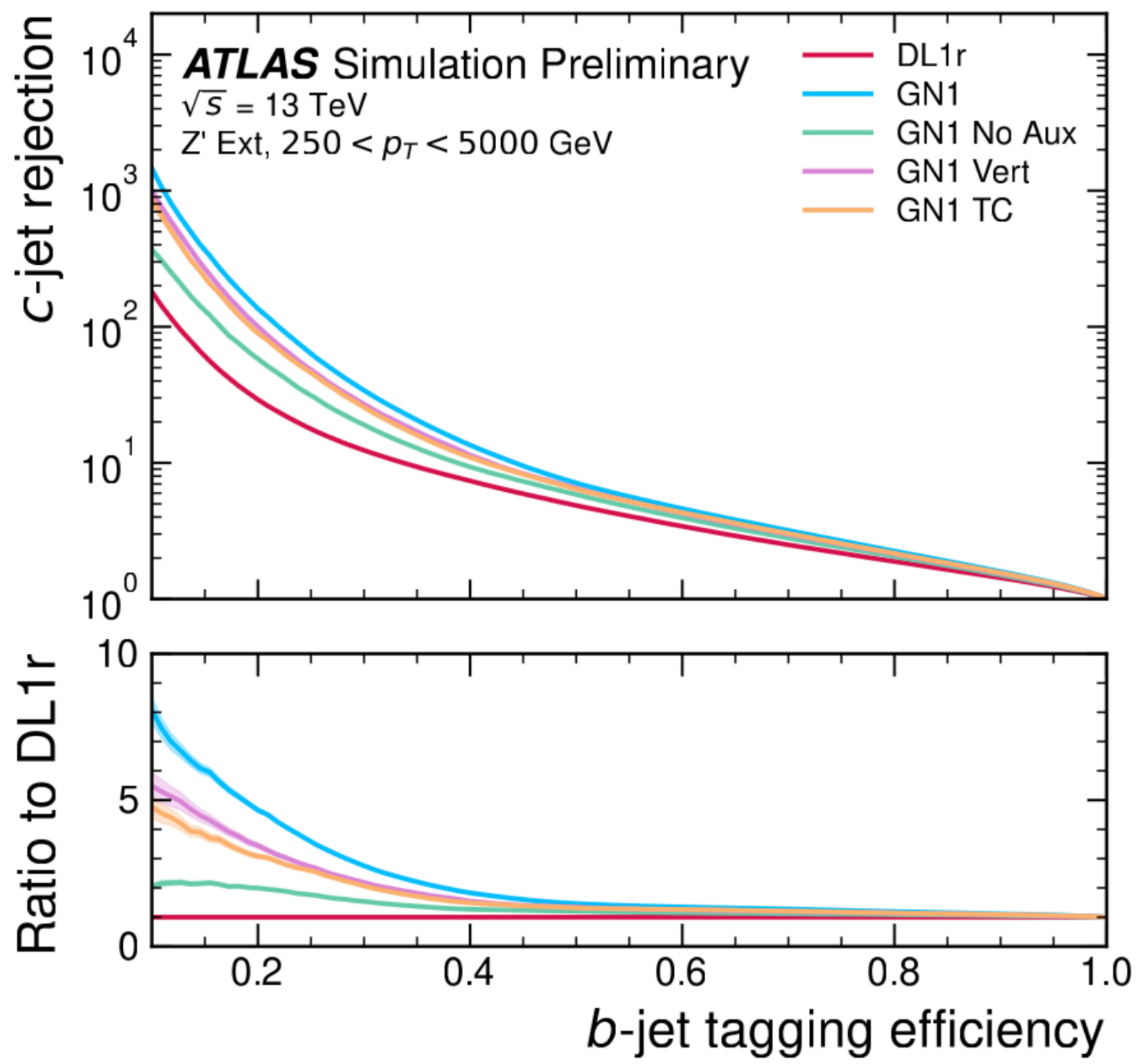
- How important are the auxiliary tasks?
- Significant degradation in performance when removing the auxiliary tasks (GN1 No Aux)
- Track classification (GN1 TC) and vertexing (GN1 Vert) bring similar level of improvement
- Both auxiliary objectives combined show the best performance, demonstrating they are complementary

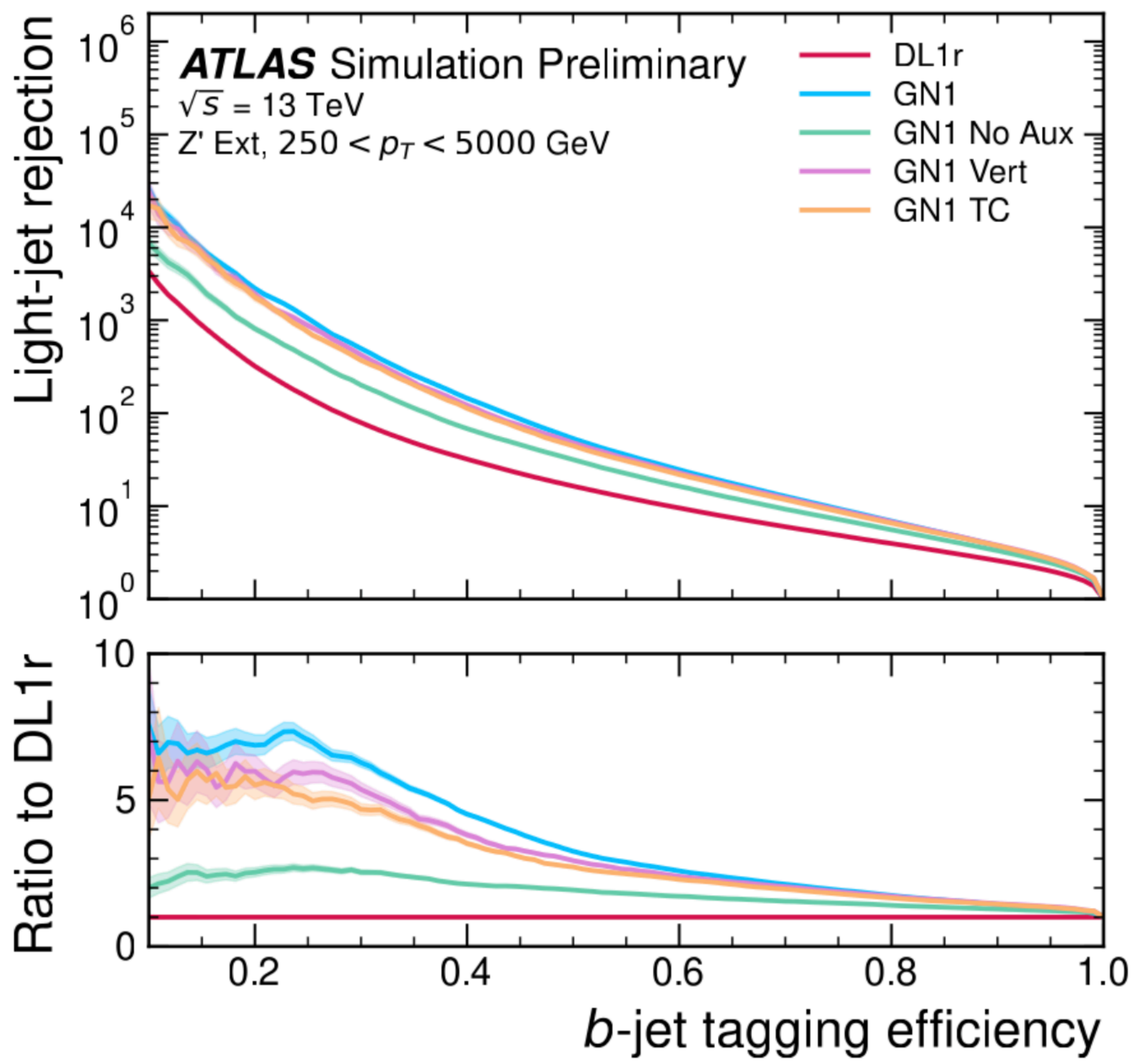
ttbar





Z'

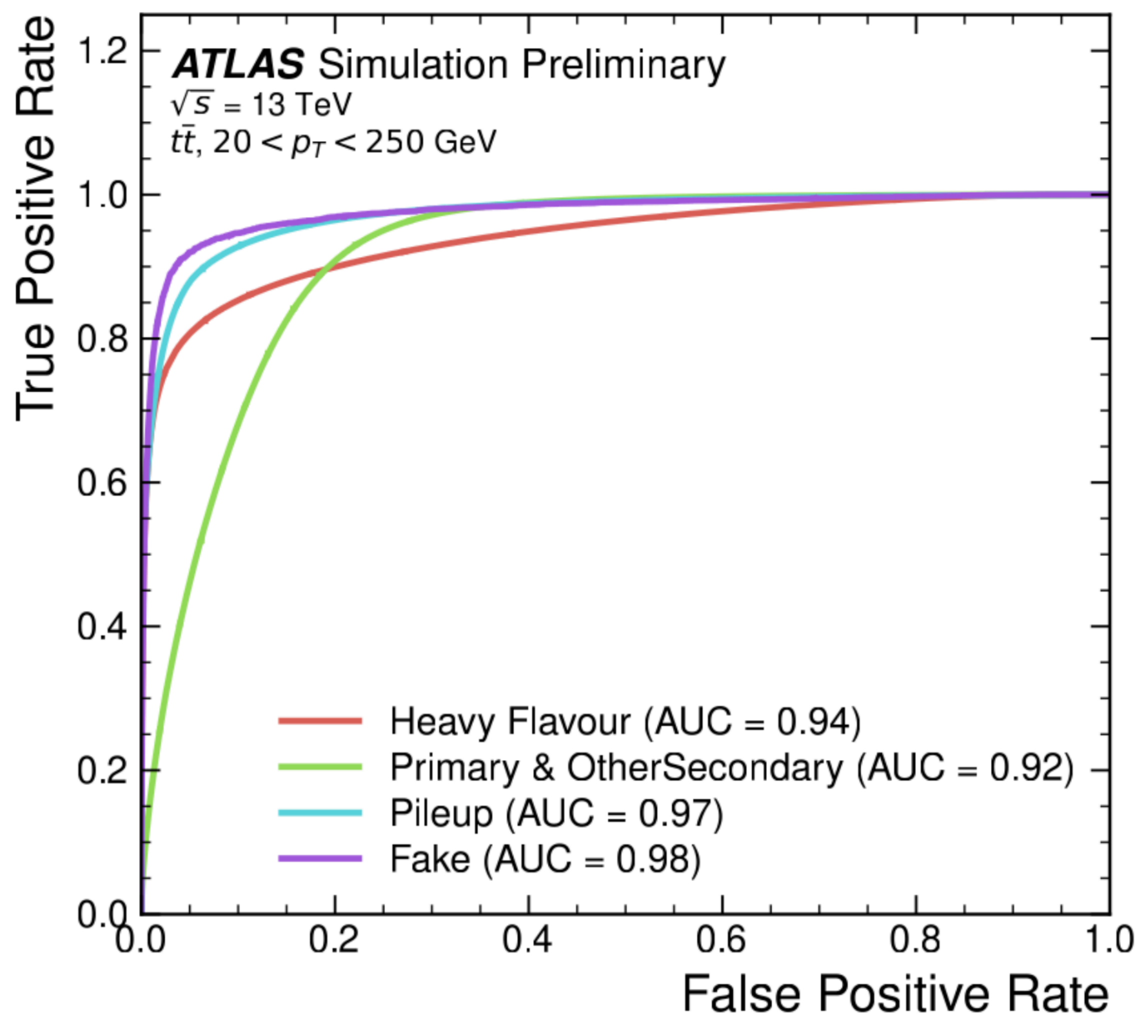




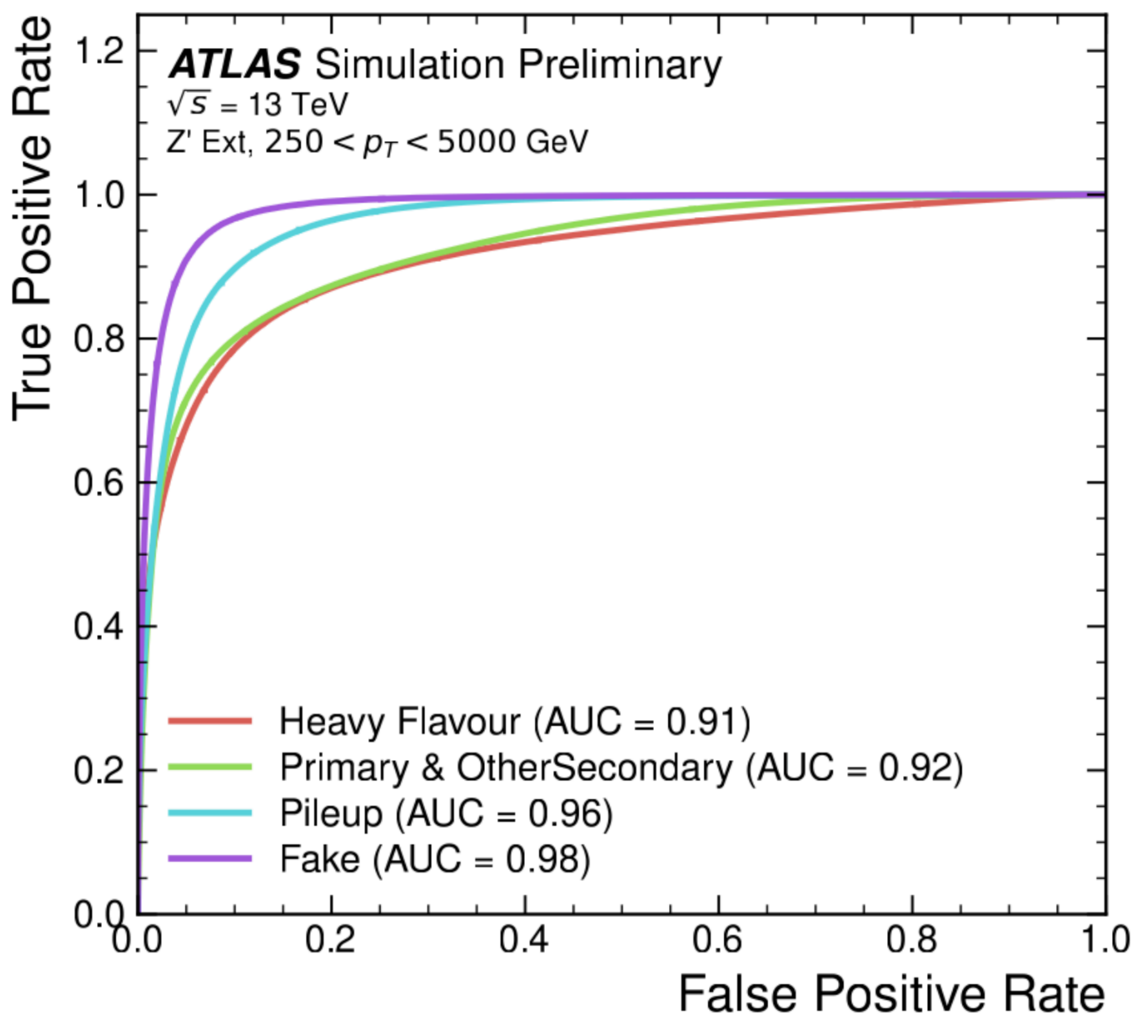
Performance of the Auxiliary tasks

- Vertexing -
  - Initial studies show that GN1 can find ~80% of the displaced vertices from b-hadron decays
- Track classification -
  - Compare to baseline Multi layer Perceptron (MLP) that processes one track at a time
  - GN1 outperforms the MLP on both ttbar and Z’
    - Explanations -
    - Mixing of information among tracks helps
    - Jet classification and vertexing can be considered as aux tasks, improving tracks classification

GN1 (ttbar)



GN1 (Z’)



AUC

		AUC	
		Mean	Weighted
$t\bar{t}$	MLP	0.87	0.89
	GN1	**0.92**	**0.95**
Z'	MLP	0.90	0.94
	GN1	**0.94**	**0.96**

Flexibility of GN1

- On top of the improved performance, GN1 is also quite flexible
- Reduced components makes it easier to optimize the tagger for new region of phase space, or when the detector and the charged particle reconstruction are updated
- Adding new physics signatures that can help in flavor tagging is also straightforward
    - In principle, it can just be another auxiliary task!
- Can easily accommodate new sets of variables (lepton vars etc)

Summary

- New Graph Neural Network based jet flavor tagging tool developed at ATLAS
- An "all-in-one" tagger!
- Shows improved performance with respect to DL1r on simulation
- The key improvement comes from the auxiliary training objectives -
  - Vertexing
  - Track origin classification
- Quite flexible and can easily be adapted to other specific used cases
- PUB Note (ATL-PHYS-PUB-2022-027) released very soon
- Fully implemented in ATLAS software
- Calibration studies on data will follow soon

Summary

- New Graph Neural Network based jet flavor tagging tool developed at ATLAS
- An “all-in-one” tagger!
- Shows improved performance with respect to DL1r on simulation
- The key improvement comes from the auxiliary training objectives -
  - Vertexing
  - Track origin classification
- Quite flexible and can easily be adapted to other specific used cases
- PUB Note (ATL-PHYS-PUB-2022-027) released very soon
- Fully implemented in ATLAS software
- Calibration studies on data will follow soon

Thank you for listening…

Backup

Implementation in ATLAS

Training	In ATLAS Software
PyTorch + DGL DEEP GRAPH LIBRARY →	ONNX

- Athena (The ATLAS software) has support for ONNXRuntime, which can run ONNX models