Boosted W Tagging using Lund Jet Plane

Rafael Andrei Vinasco Soler

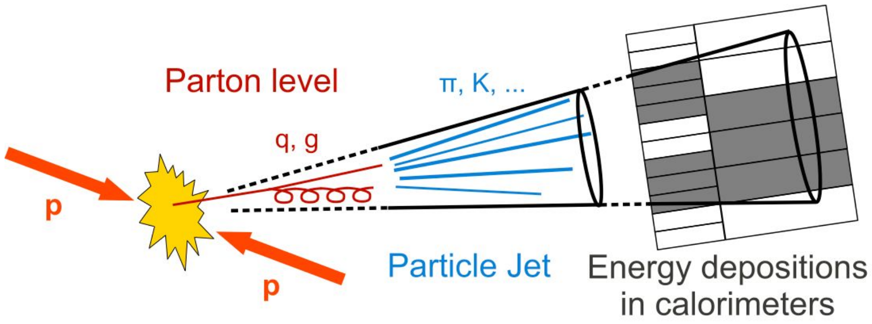
Supervisors: Reina Camacho Toro, Carlos Sandoval.

What is our goal?

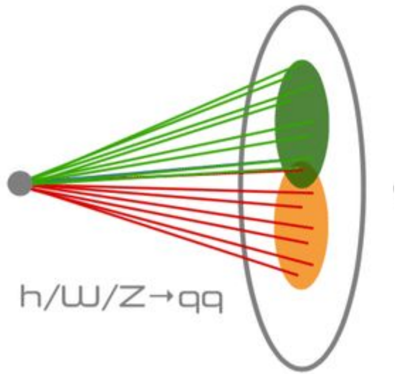
Use Lund plane variables as input for machine learning methods to develop a new **tagging methods for boosted W boson**.

Jet: A set of collimated particles produced in the hadronization of a quark or gluon.

Background



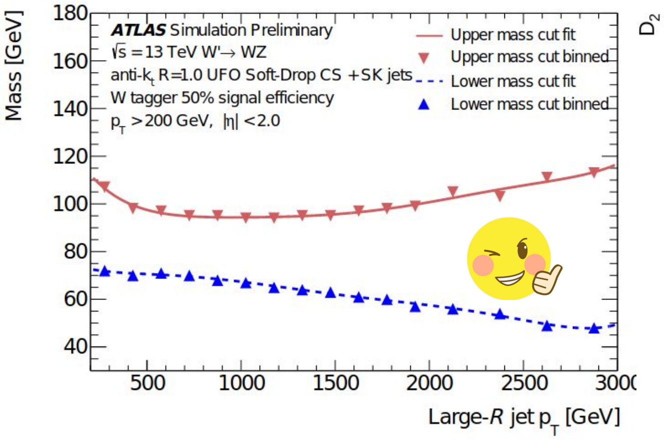
Signal

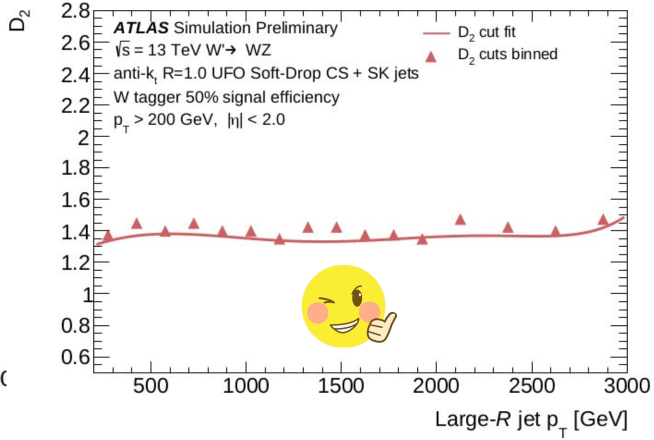


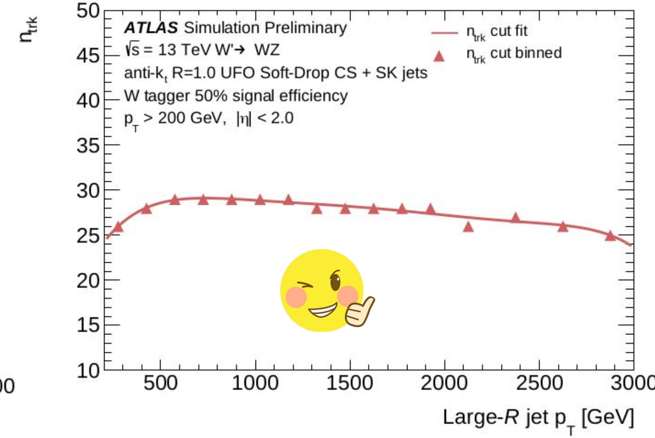
How we identify Boosted boson now?

Currently is used a tagger that perform cuts on **3 Jet substructure variables.**

These cuts are made according to the Jet transverse momentum (**pT**).



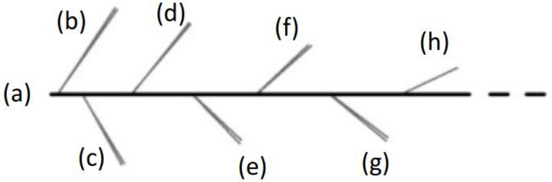


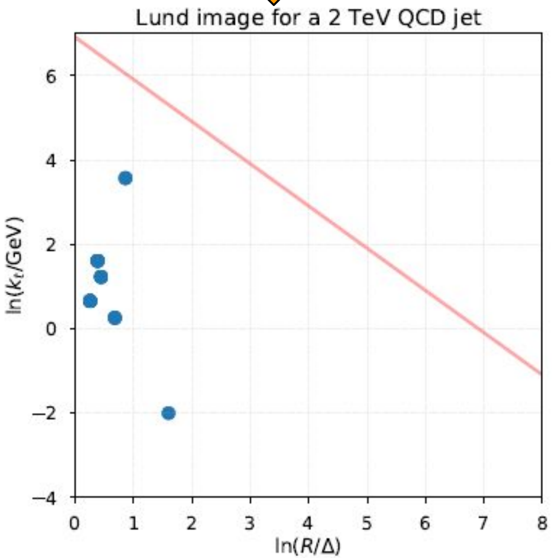


Jet Tagged!!

Plots taken from: **Performance of W/Z taggers using UFO jets in ATLAS**
https://cds.cern.ch/record/2777009/files/ATL-PHYS-PUB-2021-029.pdf

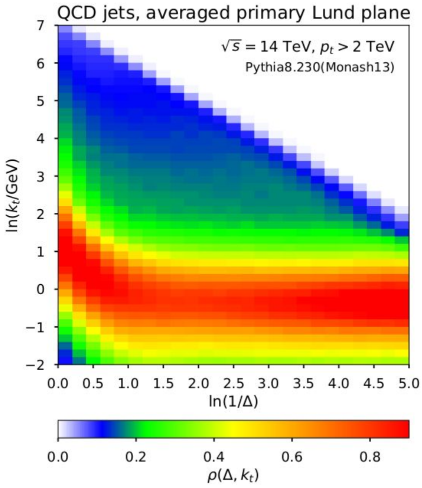
Lund plane: Is a way to represent the phase space of jet constituents reconstructed by reversing jet clustering sequence.

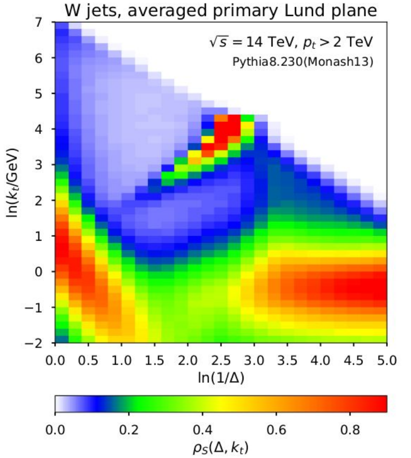




- **Great to separate QCD and W-jets**

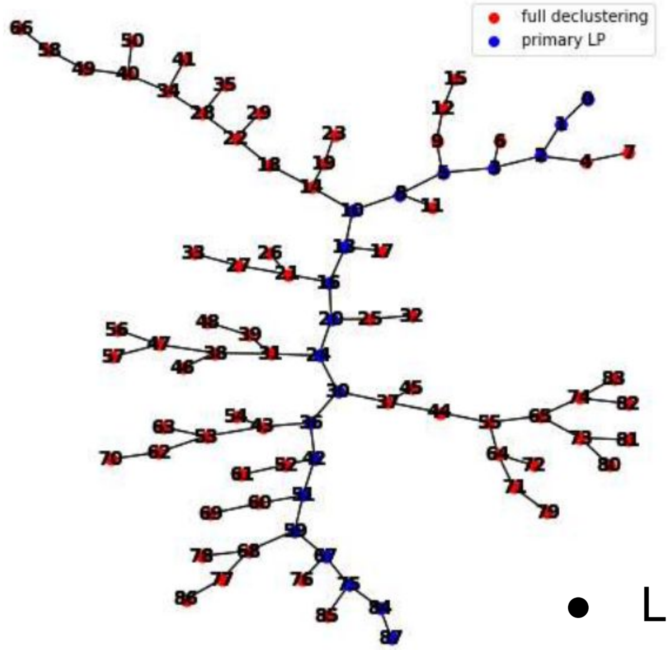
- **Lund plane variables:**
  - **kT** : Transverse momentum of the emission.
  - **Δ** : Emission angle
  - **Z** : Momentum fraction of branching





Plots taken from: **Dreyer, F.A., Salam, G.P. and Soyez, G. (2018). The Lund jet plane**. https://arxiv.org/pdf/1807.04758.pdf

Full Lund plane



- Using the Lund Plane we are going inside the hadronization history. Every single emission is represented!

- If is used the information of each emision instead of using jet global variables we can do a better background discrimitation

More information used → **Better performance**

- Lund planes is made up as a set of vertices and their connection edge, so this is an ideal input for Graph Neural Networks!

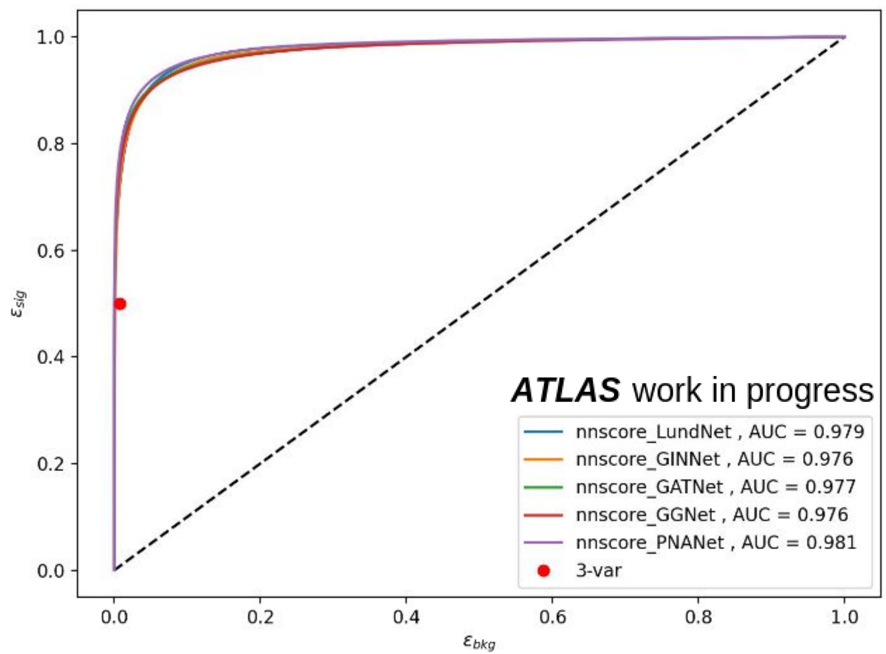
pT > 200 GeV

$$\frac{1}{\epsilon_{\text{background}}} = \frac{N^{\text{identified}}_{\text{background}}}{N^{\text{total}}_{\text{background}}} \longrightarrow \text{“Background rejection”}$$

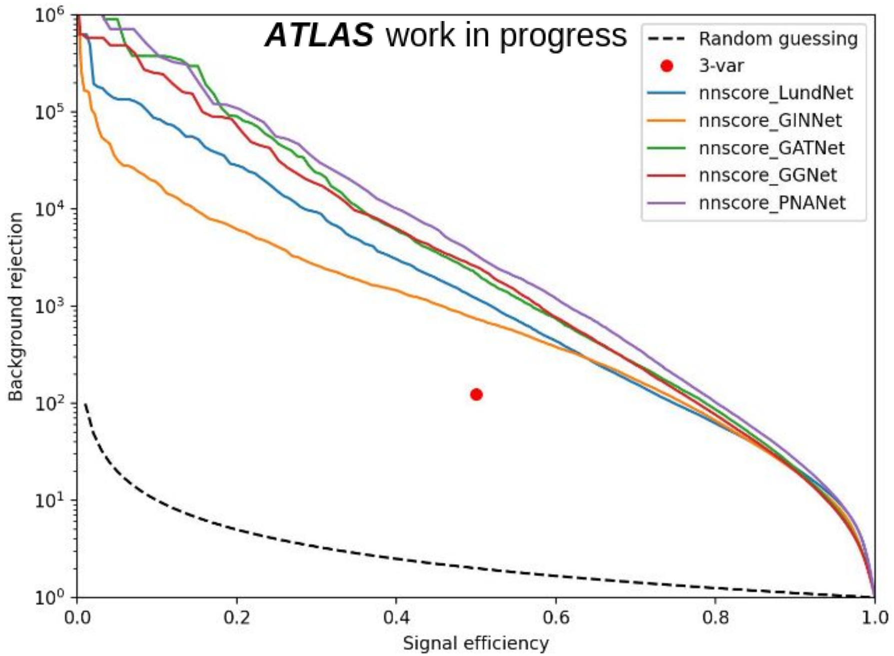
$$\epsilon_{\text{signal}} = \frac{N^{\text{identified}}_{\text{signal}}}{N^{\text{total}}_{\text{signal}}} \longrightarrow \text{“Signal efficiency”}$$

Five different GNN structures have been tested.

NN taggers ROC curve



QCD rejection vs W tagging efficiency



- 50 GeV < Jet mass < 100 GeV
- | Jet eta | < 2
- Jet truth match with W boson
- Number of b Hadrons = 0

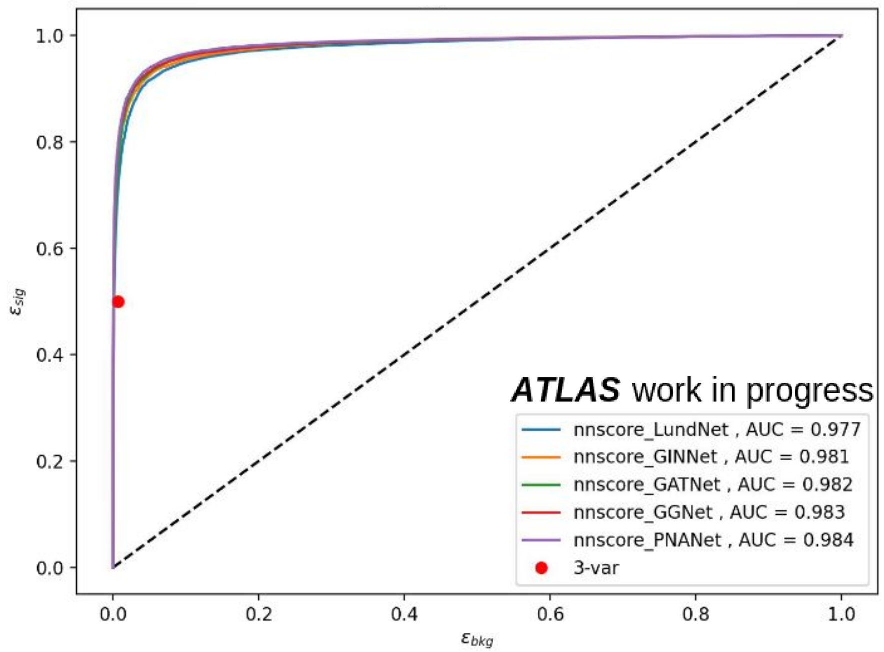
pT > 500 GeV

$$\frac{1}{\epsilon_{\text{background}}} = \frac{N^{\text{identified}}_{\text{background}}}{N^{\text{total}}_{\text{background}}} \longrightarrow \text{“Background rejection”}$$

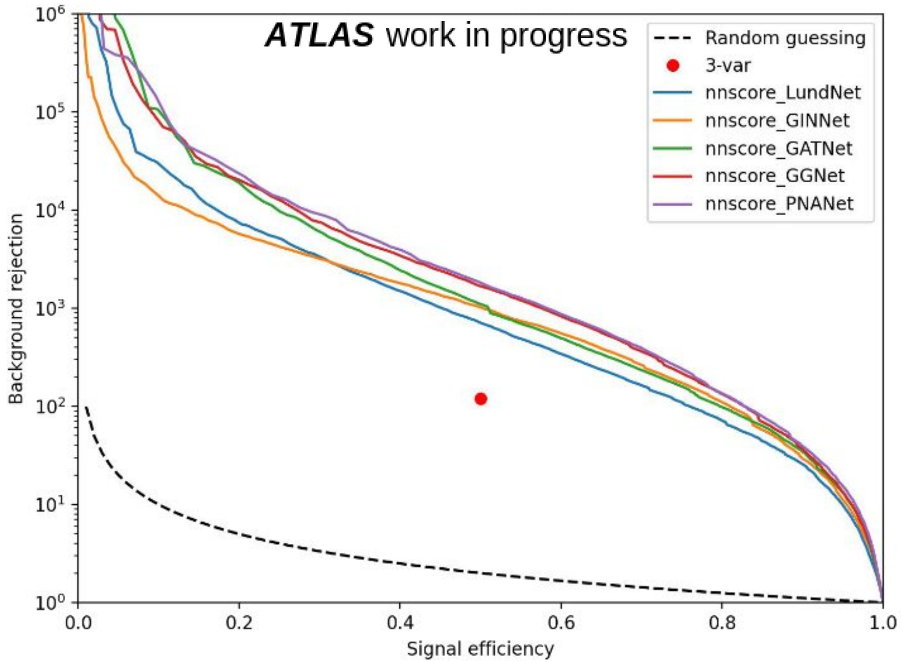
$$\epsilon_{\text{signal}} = \frac{N^{\text{identified}}_{\text{signal}}}{N^{\text{total}}_{\text{signal}}} \longrightarrow \text{“Signal efficiency”}$$

Five different GNN structures have been tested.

NN taggers ROC curve



QCD rejection vs W tagging efficiency



- 50 GeV < Jet mass < 100 GeV
- | Jet eta | < 2
- Jet truth match with W boson
- Number of b Hadrons = 0

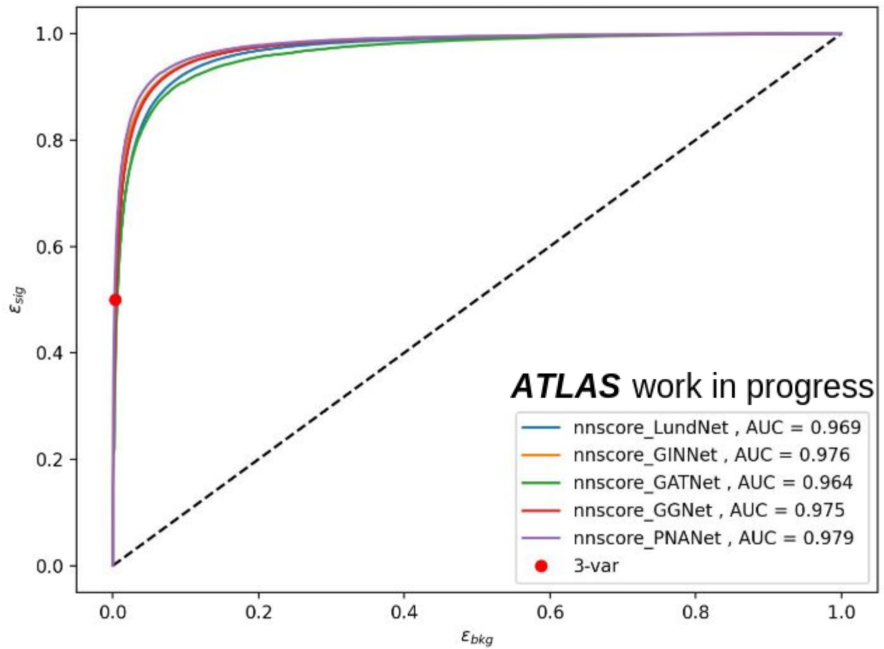
pT > 500 GeV

$$\frac{1}{\epsilon_{\text{background}}} = \frac{N^{\text{identified}}_{\text{background}}}{N^{\text{total}}_{\text{background}}} \longrightarrow \text{“Background rejection”}$$

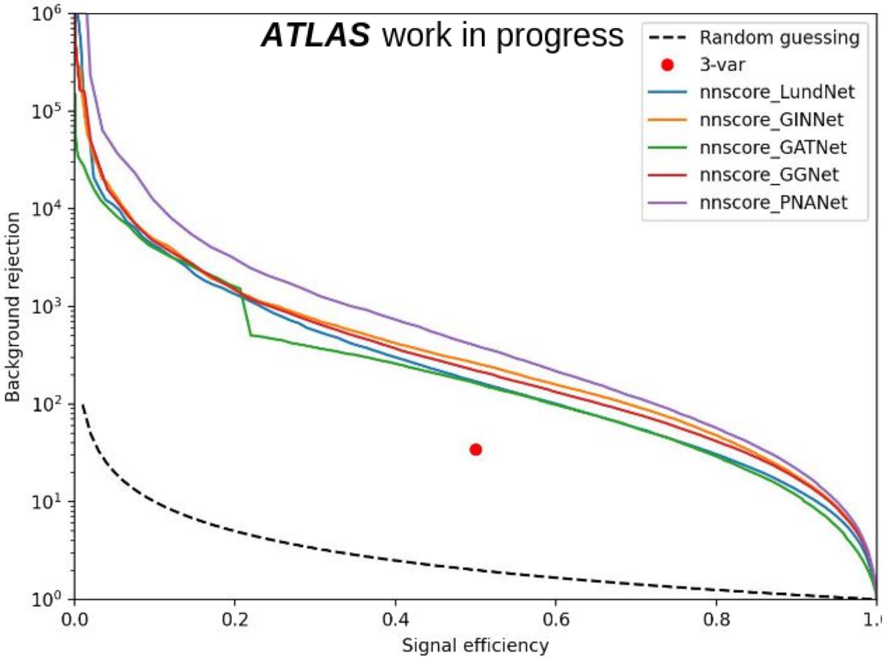
$$\epsilon_{\text{signal}} = \frac{N^{\text{identified}}_{\text{signal}}}{N^{\text{total}}_{\text{signal}}} \longrightarrow \text{“Signal efficiency”}$$

Five different GNN structures have been tested.

NN taggers ROC curve



QCD rejection vs W tagging efficiency



- 50 GeV < Jet mass < 100 GeV
- | Jet eta | < 2
- Jet truth match with W boson
- Number of b Hadrons = 0

Conclusions

- Presented 5 GNN architectures with improved performance over the currently boosted W boson taggers.

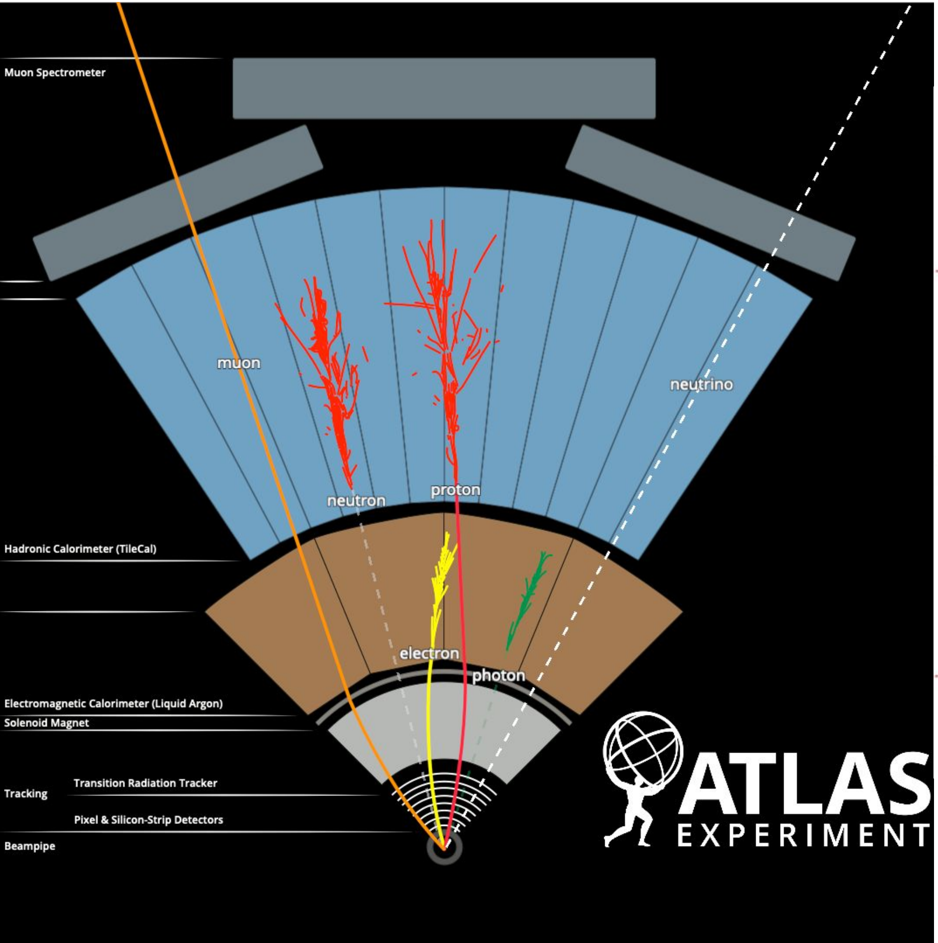
Work in progress

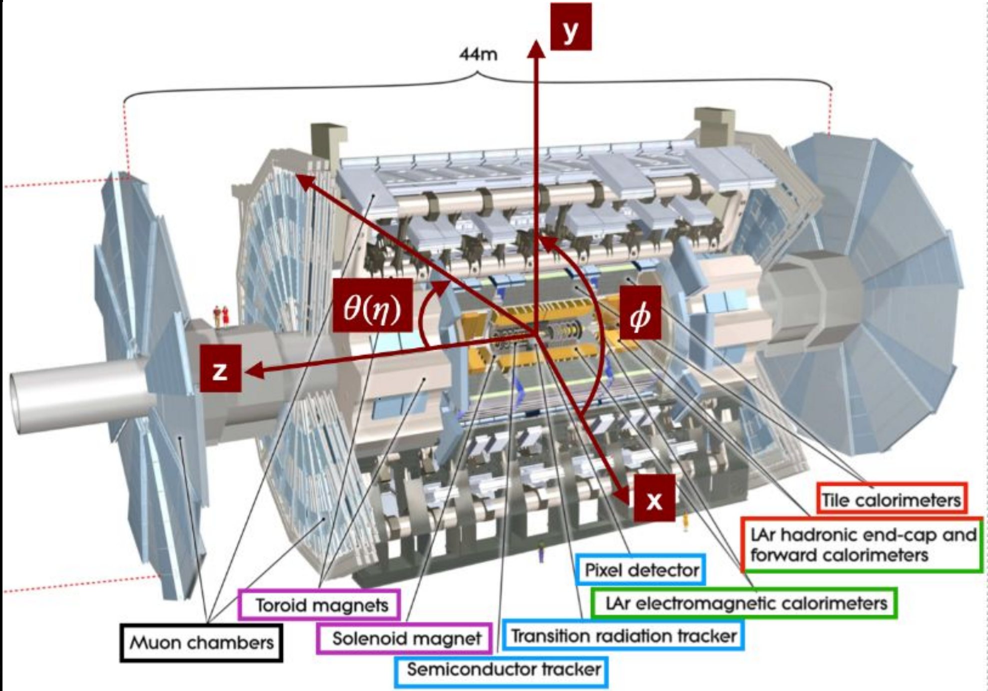
- Optimization in the implementation of the differents models.

Thanks for your attention :)

BACKUP ;)

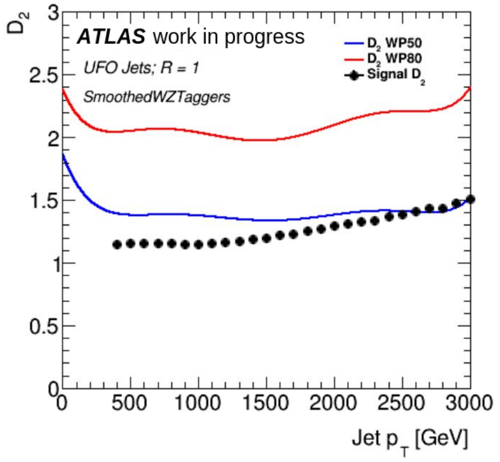
ATLAS DETECTOR

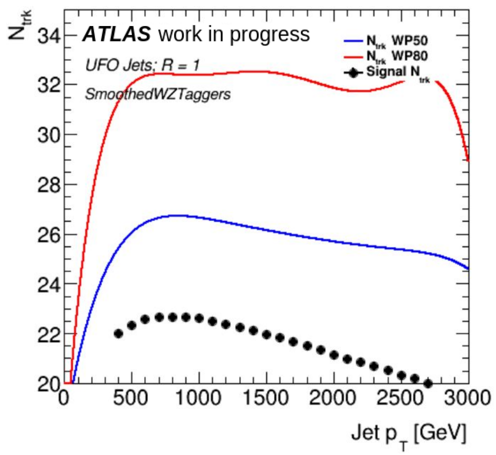




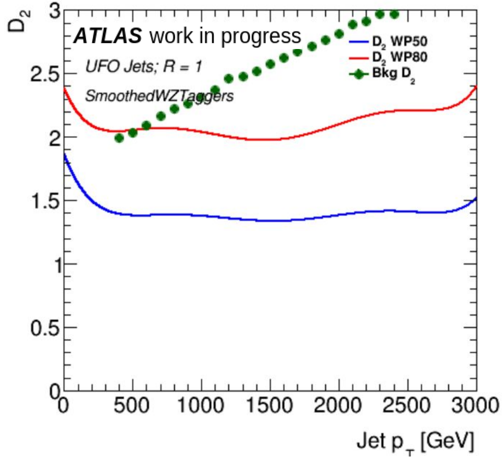
Standard Tagger performance

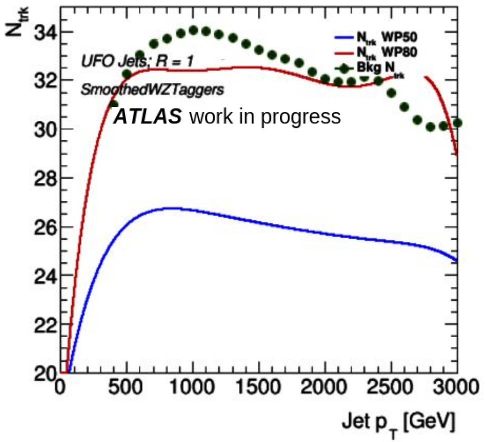
Signal performance





Background performance





Datasets

Events were generated using Monte Carlo simulations in Powheg and Pythia 8 and the detector is simulated using Geant4. Precisely, this is the data used:

- Dijets: 8,277,229 total events.
  mc16_13TeV.3647[03,09].Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ[03,09]WithSW.deriv.DAOD_JETM8.e7142_s3126_r10201_p4355
- W prime (W boson): 3,343,338 total events.
  mc16_13TeV.426347.Pythia8EvtGen_A14NNPDF23LO_WprimeWZ_flatpT.deriv.DAOD_JETM8.e6880_s3126_r10201_p4355

Train size:

10% of the Dataset used for training - 90% of the Dataset used for testing.

Cut-based tagger

- Jet pT > 200 GeV
- 50 GeV < Jet mass < 100 GeV
- | Jet eta | < 2
- Jet truth match with W boson
- $|\Delta R| < 0.75$
- Number of b Hadrons = 0

Models (GNN architectures)

Traditional Neural Networks require input to be of fixed length whilst Graph Neural Networks do not have this limitation, whether the input graph has 2 nodes or 20, the GNN model can handle it!

GNN architectures

- LundNet (https://arxiv.org/pdf/2012.08526.pdf) → **our inspiration**
- Graph Isomorphism Network (GINConv)
- Graph Attention Network (GATConv)
- Gated Graph Sequence Neural Network (GatedGraphConv)
- Principal Neighbourhood Aggregation Network (PNAConv)

All documented as GINConv, GATConv, GatedGraphConv, and PNAConv, respectively, at :
https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#convolutional-layers

New Selection

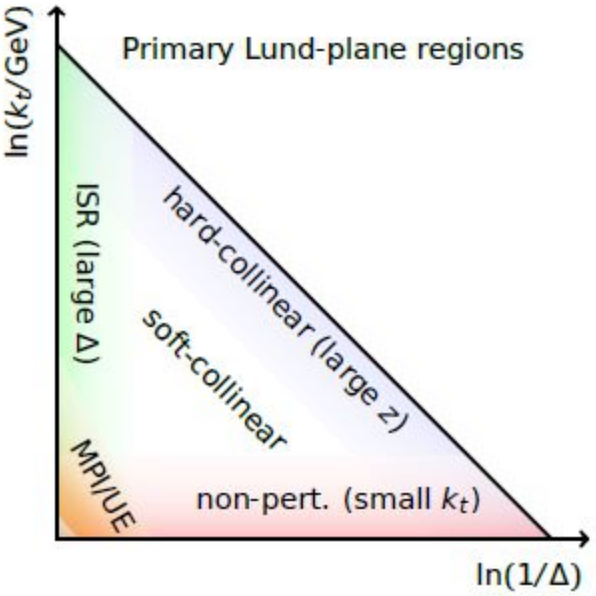
Signal and Background cuts:

Ungroomed Jet_pt > 200 GeV,
Jet_pt > 200 GeV,
Jet_pt < 3000 GeV,
Jet_mass > 40 GeV,
Jet_mass < 300 GeV,
Jet_D2 > 0,

Signal definition

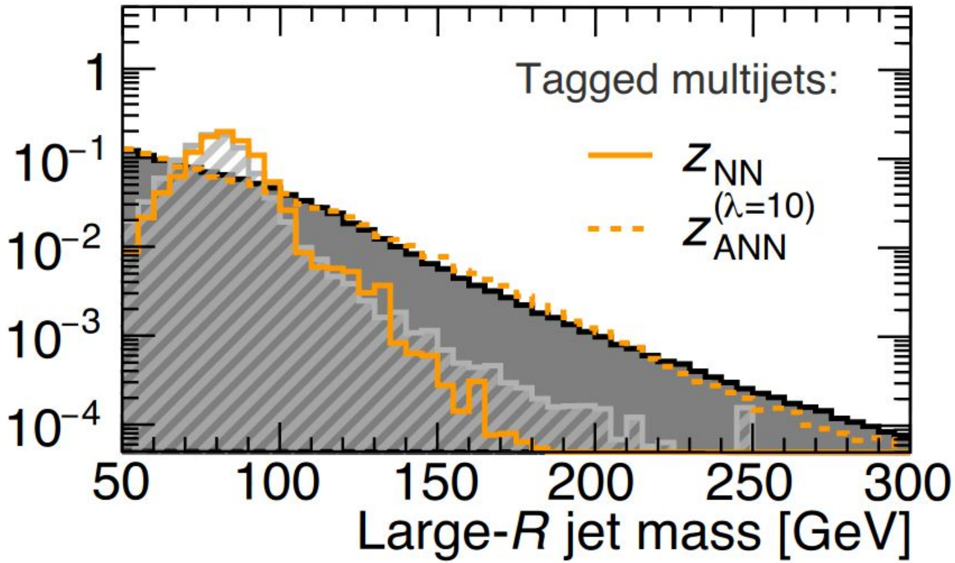
Jet truth match with W boson
Ungroomed Jet_mass > 50 GeV
Number of b Hadrons = 0

Lund Plane regions



Using **ln(Kt)** and **ln(1/Δ)** is easy to identify differents regions.

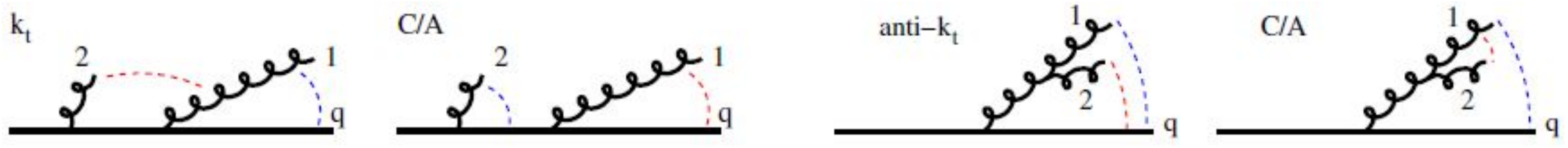
Mass sculpting



After the selection the mass profile of the background signal changed! To avoid that we could use an Adversarial Neural Network!

ATLAS EXPERIMENT

Declustering algorithms



- The Declustering algorithms tries to go inside the hadronization history in order to determine where each emission is coming from.

Contribuciones a NLO:

$$\bar{\rho}_2^{(k_t)}(\Delta,\kappa) \simeq -4C_F^2 \ln^2 \frac{\Delta}{\kappa} + \mathcal{O}(L) .$$ → Kt algorithm

$$\bar{\rho}_2^{(\text{anti-}k_t)}(\Delta,\kappa) \simeq +8C_F C_A \ln^2 \frac{\Delta}{\kappa} + \mathcal{O}(L) .$$ → Anti-Kt algorithm

$$\bar{\rho}_{2,\text{rc}}^{(\text{C/A})}(\Delta,\kappa) = \bar{\rho}_1(\Delta,\kappa)\, 4\pi b_0 \ln \frac{1}{\kappa} + \mathcal{O}(1) .$$ → C/A algorithm