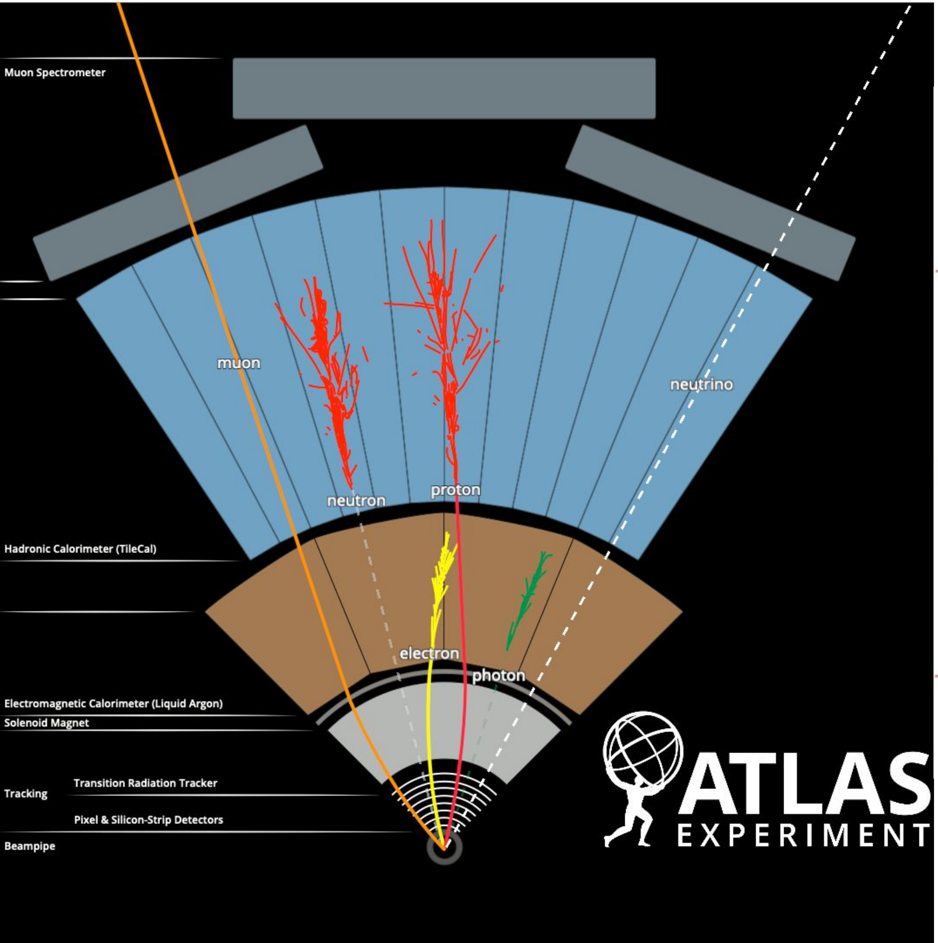
Boosted W and H Tagging using Lund Jet Plane

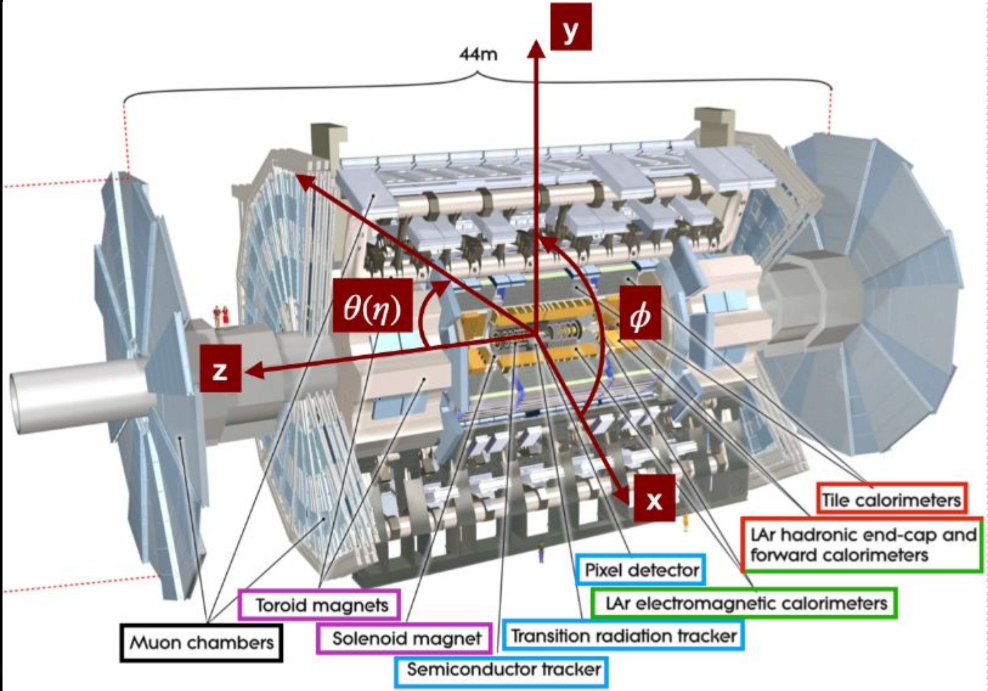
Rafael Andrei Vinasco Soler

Supervisors: Reina Camacho Toro, Carlos Sandoval.

And thanks to Mykola Khandoga and Jad Mathieu Sardain for their full support

ATLAS DETECTOR



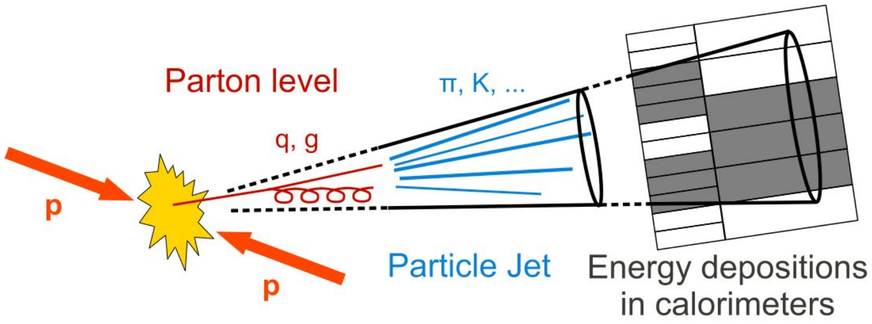


What is our goal?

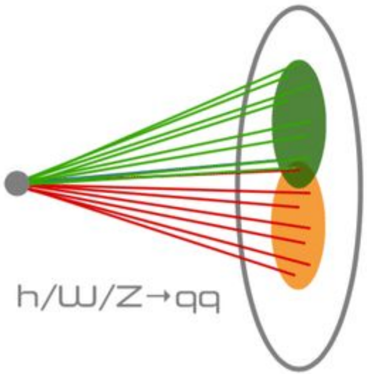
Use Lund plane variables as input for machine learning methods to develop a new **tagging methods for boosted W and Higgs bosons**.

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

Background

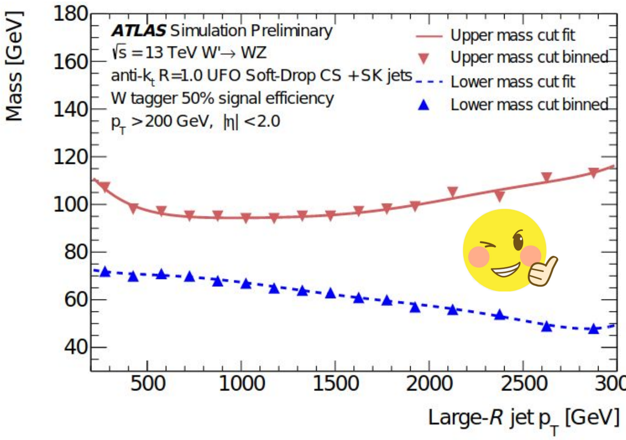


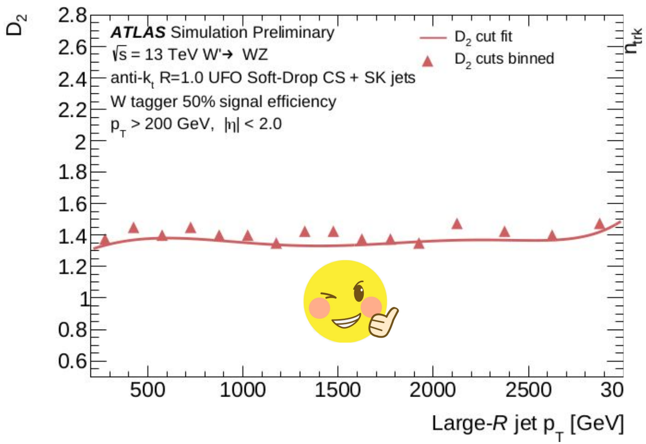
Signal

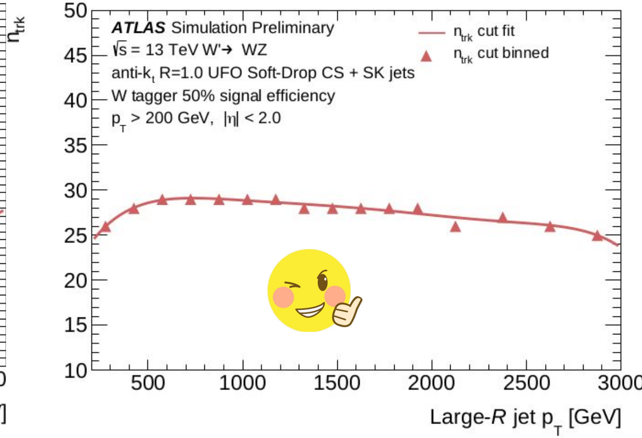


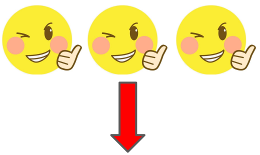
How we identify W Boosted boson now?

Currently is used a tagger that perform cuts on **3 Jet substructure variables.** These cuts are made according to the Jet transverse momentum (**pT**).





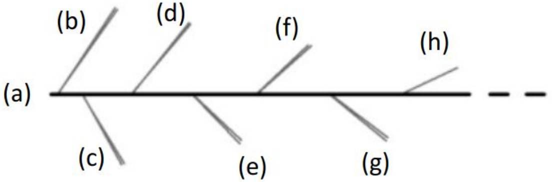


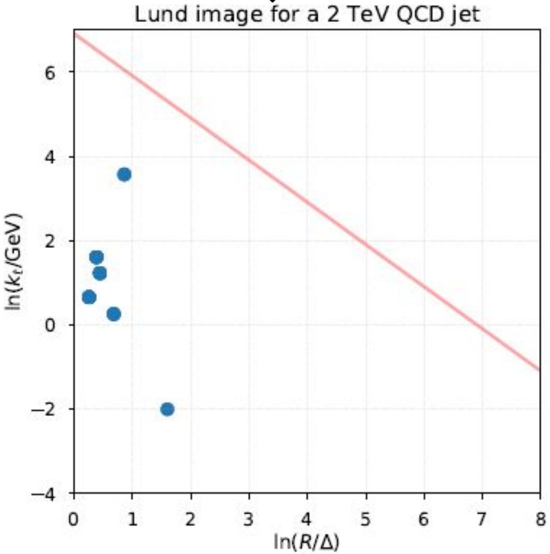


Jet Tagged!!

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

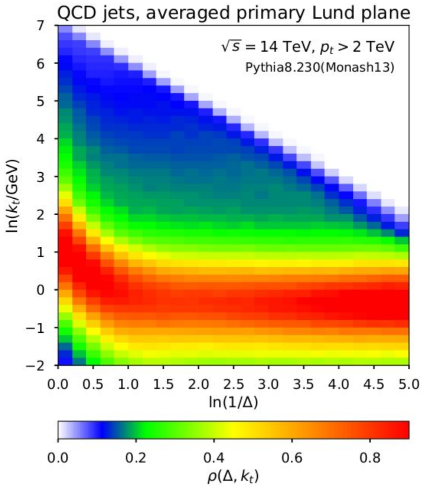
Lund plane: Is a way to represent the phase space of jet constituents reconstructed by reversing jet clustering sequence.

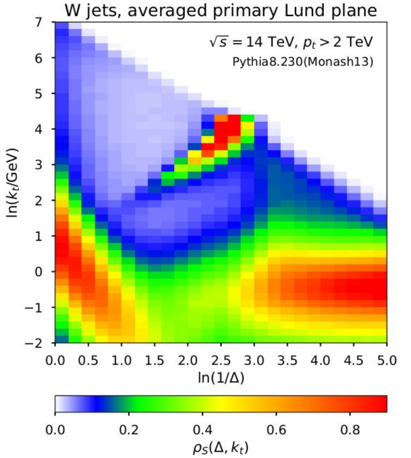




- **Great to separate QCD and W-jets**

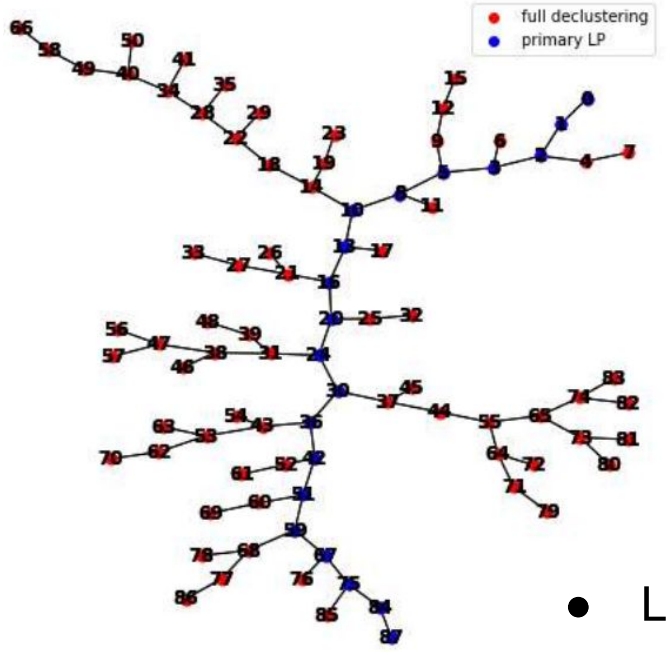
- **Lund plane variables:**
  - ○ **kT** : Transverse momentum of the emission.
  - ○ **Δ** : Emission angle
  - ○ **Z** : Momentum fraction of branching





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!

Models (GNN architectures) tested

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)

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

Data used for W tagge

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**: mc16_13TeV.3647[03,09].Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ[03,09]WithSW.deriv.DAOD_JETM8.e7142_s3126_r10201_p4355
- **W prime** (only channel W' to WZ is included): mc16_13TeV.426347.Pythia8EvtGen_A14NNPDF23LO_WprimeWZ_flatpT.deriv.DAOD_JETM8.e6880_s3126_r10201_p4355

Train size:

2% of dijet background and 10% of W signal.

Signal and Background definitions

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

Background rejection and Signal efficiency

Background rejection: How many Background is discarded for the classifier

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

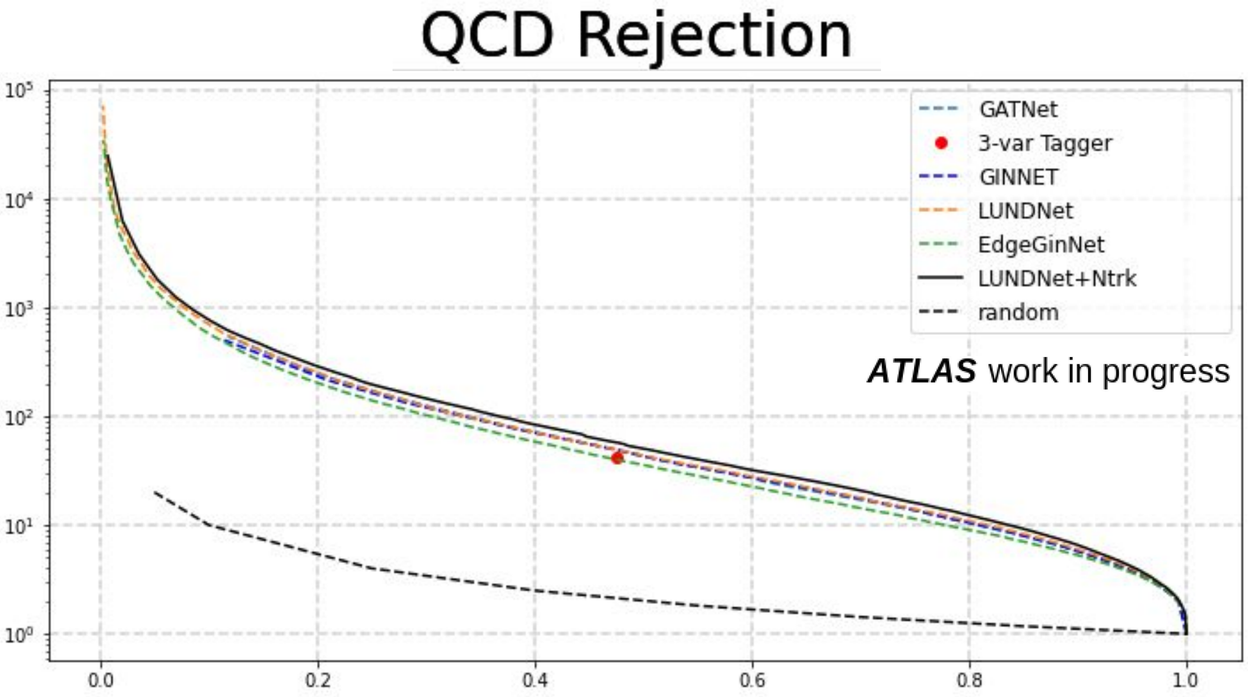
Background rejection=200 ➡ 1 of 200 background jets pass the selection

Signal efficiency: How many signal remain after the selection

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

Tagger results

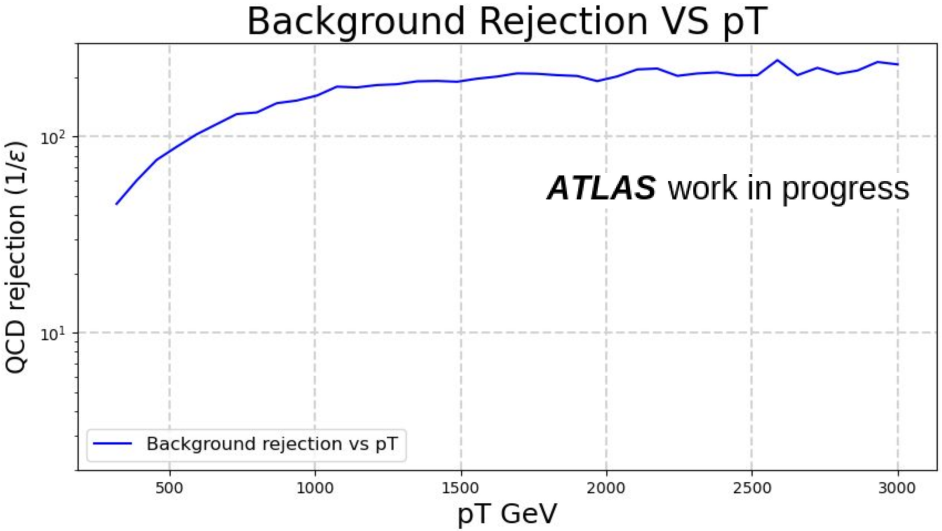
Four different GNN structures have been tested.

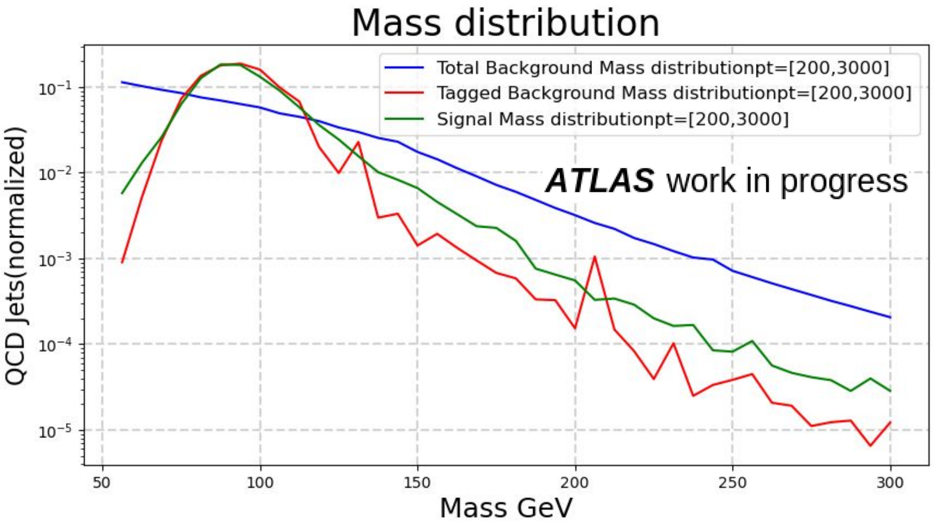


- 50 GeV < Jet mass < 300 GeV
- | Jet eta | < 2
- Jet truth match with W boson
- Number of b Hadrons = 0

Tagger results

LUNDNet + Number of tracks (Ntrk)





- 50 GeV < Jet mass < 300 GeV
- | Jet eta | < 2
- Jet truth match with W boson
- Number of b Hadrons = 0

Adversarial Network

In order to mass decorrelated the tagger an Adversarial Neural Network is added, this network is a Mixture gaussian model that learn how is the mass of the Jet using the output score of the classifier.

A new loss function is used: (f=classifier , r=adversarial)

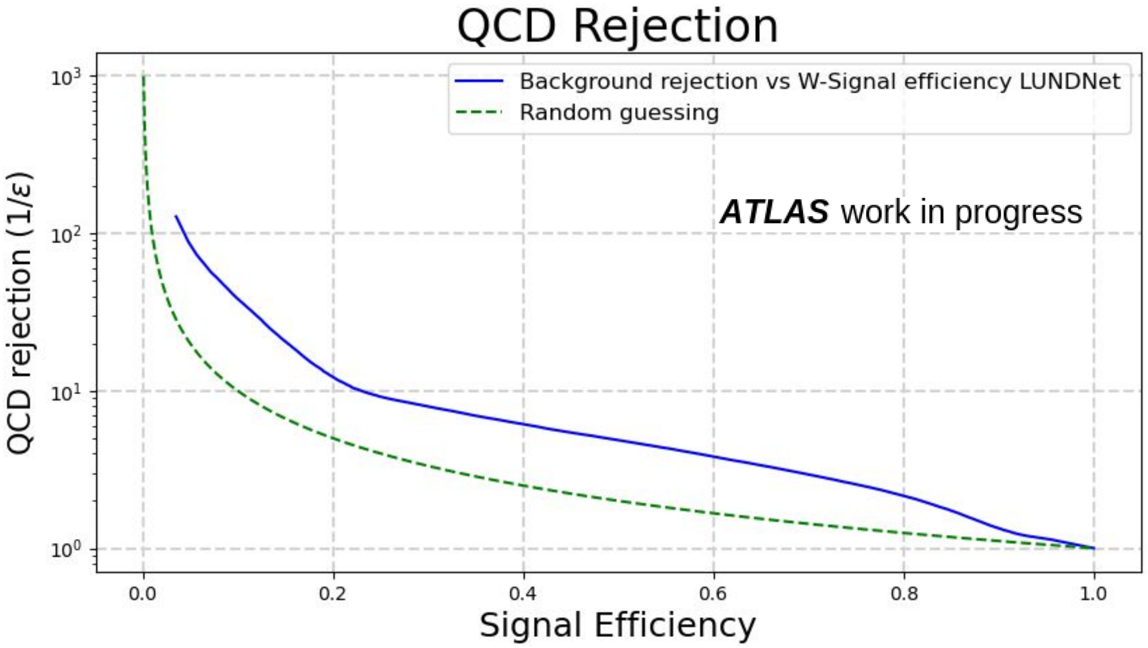
$$E(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \mathcal{L}_r(\theta_f, \theta_r)$$

Where the purpose of the algorithm is:

$$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f}\max_{\theta_r} E(\theta_f, \theta_r).$$

Classifier + adversarial results

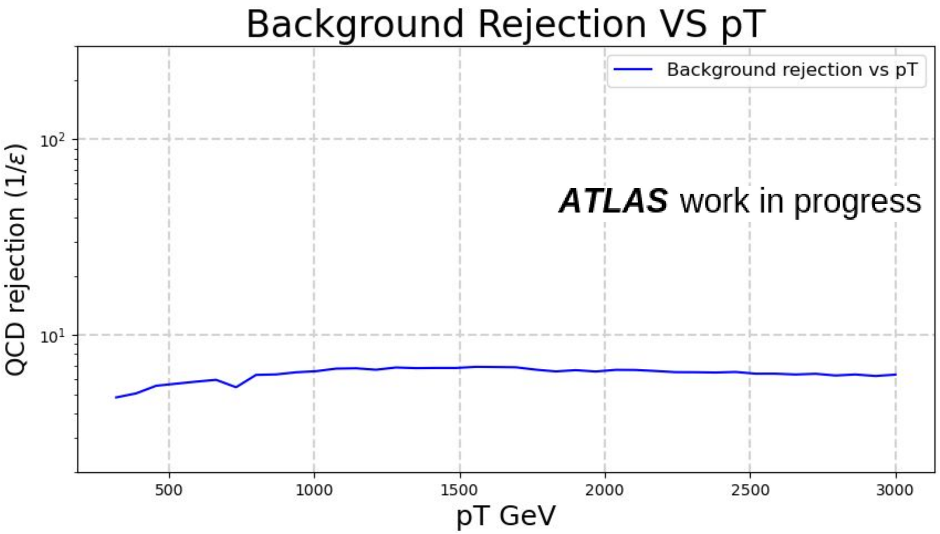
GATNet results (the best one)

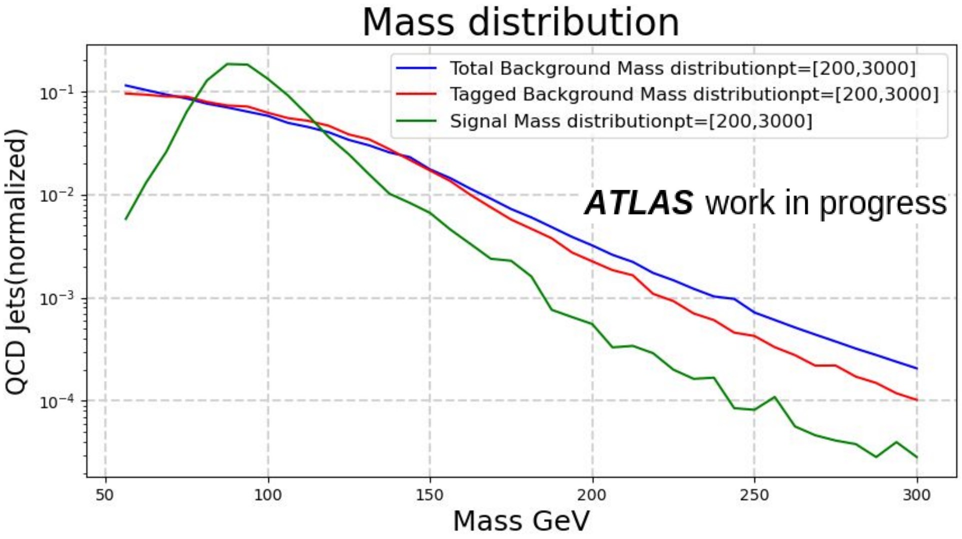


○ 50 GeV < Jet mass < 300 GeV	○ Jet truth match with W boson
○ | Jet eta | < 2	○ Number of b Hadrons = 0

Classifier + adversarial results

GATNet result (the best one)





- 50 GeV < Jet mass < 300 GeV
- | Jet eta | < 2
- Jet truth match with W boson
- Number of b Hadrons = 0

What if…

Is expected that the emissions coming from parton shower processes and not from Hard processes have lower momentum, so this is the idea to do a Cut off in the emissions used.

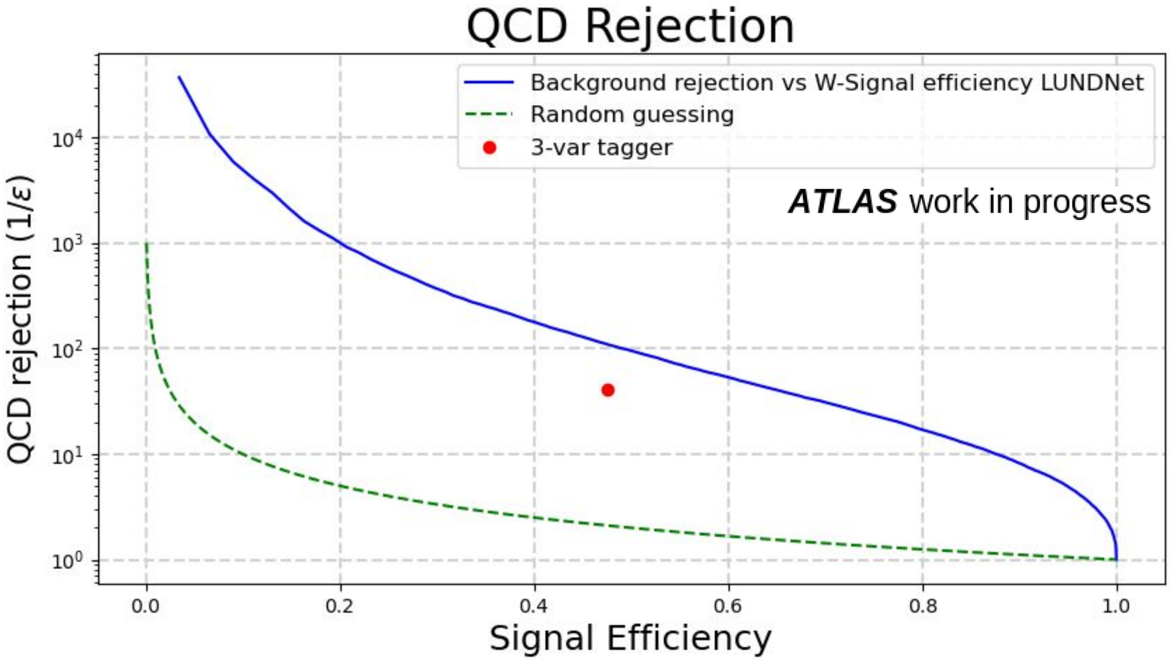
Only the 9 emissions with highest pT are taken.

Parton shower (QCD emissions) → **Expected Lower energy**

This let us use more events and remove possible undesired emissions in the training

Tagger using first 9 emissions

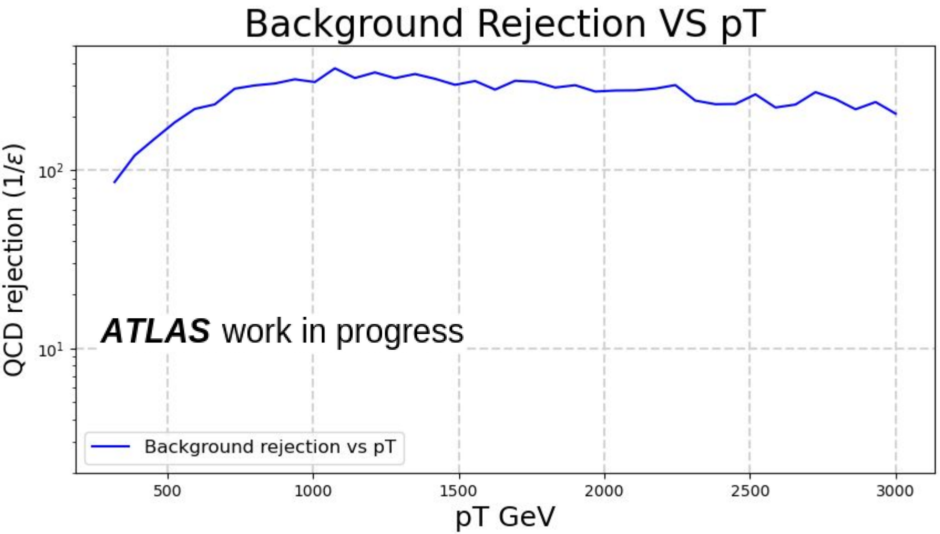
Train size: 5% of dijet background and 5% of W signal. Results for LUNDNet + Ntrk

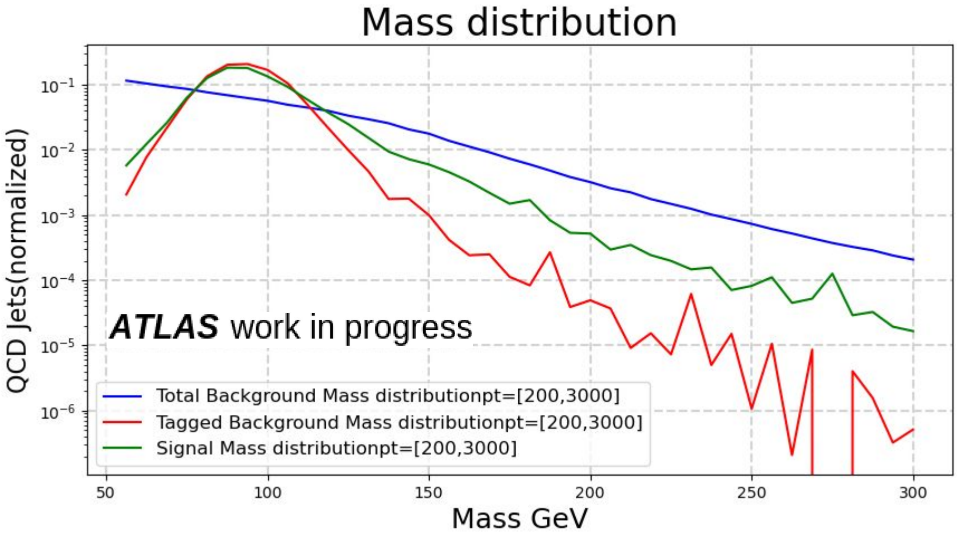


○ 50 GeV < Jet mass < 300 GeV	○ Jet truth match with W boson
○ | Jet eta | < 2	○ Number of b Hadrons = 0

Tagger using first 9 emissions

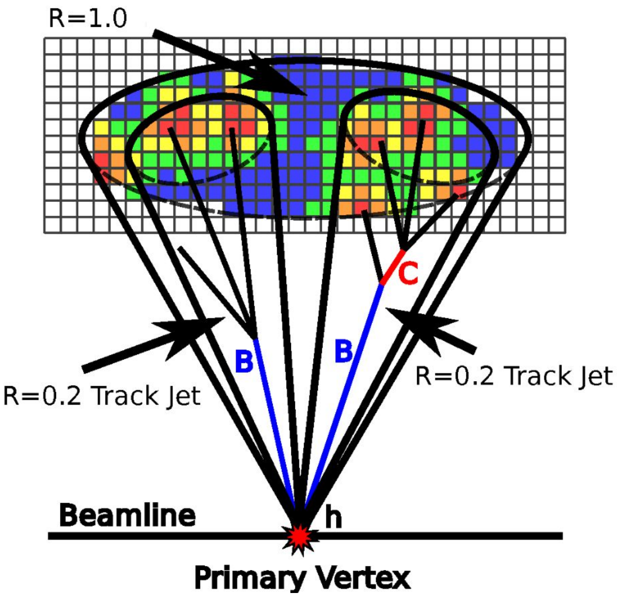
Train size: 5% of dijet background and 5% of W signal. Results for LUNDNet + Ntrk





Classifier + Adversarial result are not done yet!

Higgs to bb tagger



Current Tagger: Use as input global variables for the large Jet R=1.0 and the ouput of a NN flavor tagger for small jets R=0.2~0.4 inside the jet.

Data selection

Monte Carlo data samples used: G’ -> HH -> bbbb samples.

The mass of G’ is in the range of [400,6000] GeV.

Signal definition:

250 GeV < pT < 2800 GeV

$|\eta| < 2$

$76 < M_J /\ [\mathrm{GeV}] > 146$

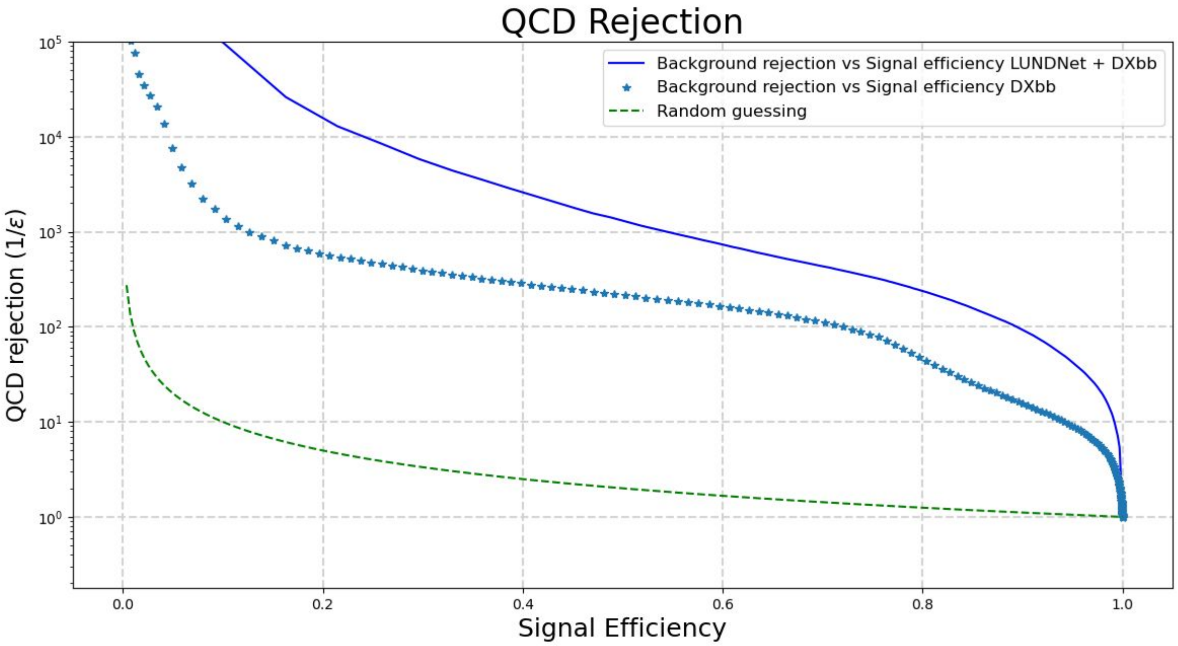
Signal definition

Jet truth match with H boson

Number of b Hadrons > 1

Higgs to bb: Results

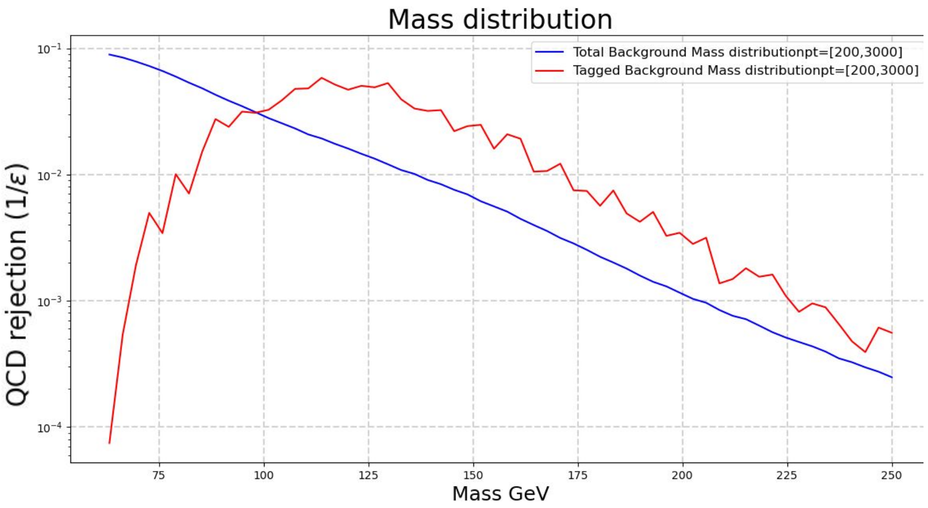
Train size: 2% of dijet background and 40% of H to bb signal. Results for LUNDNet + DXbb tagger

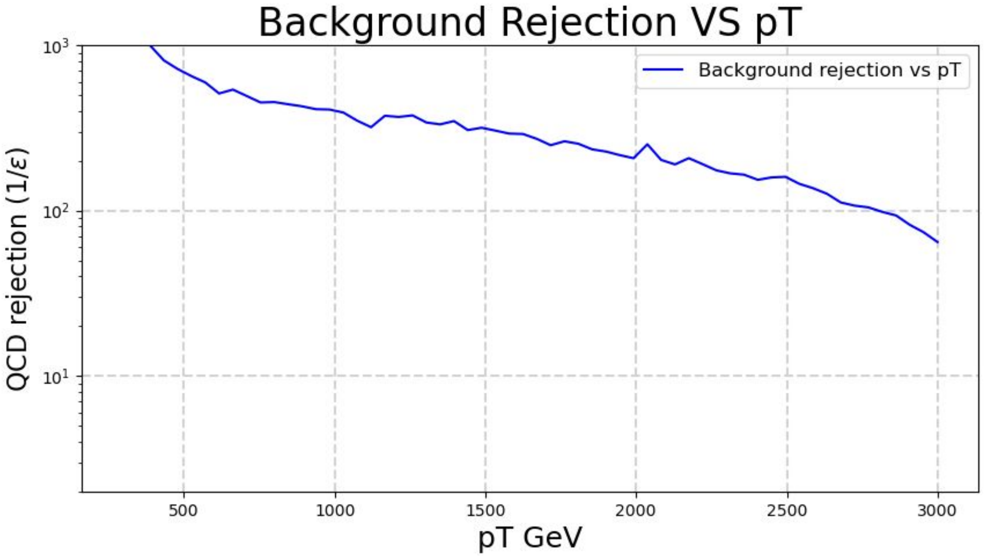


○ 76 GeV < Jet mass < 146 GeV	○ Jet truth match with H boson
○ | Jet eta | < 2	○ Number of b Hadrons > 1

Higgs to bb: Results

Train size: 2% of dijet background and 40% of H to bb signal. Results for LUNDNet + DXbb tagger

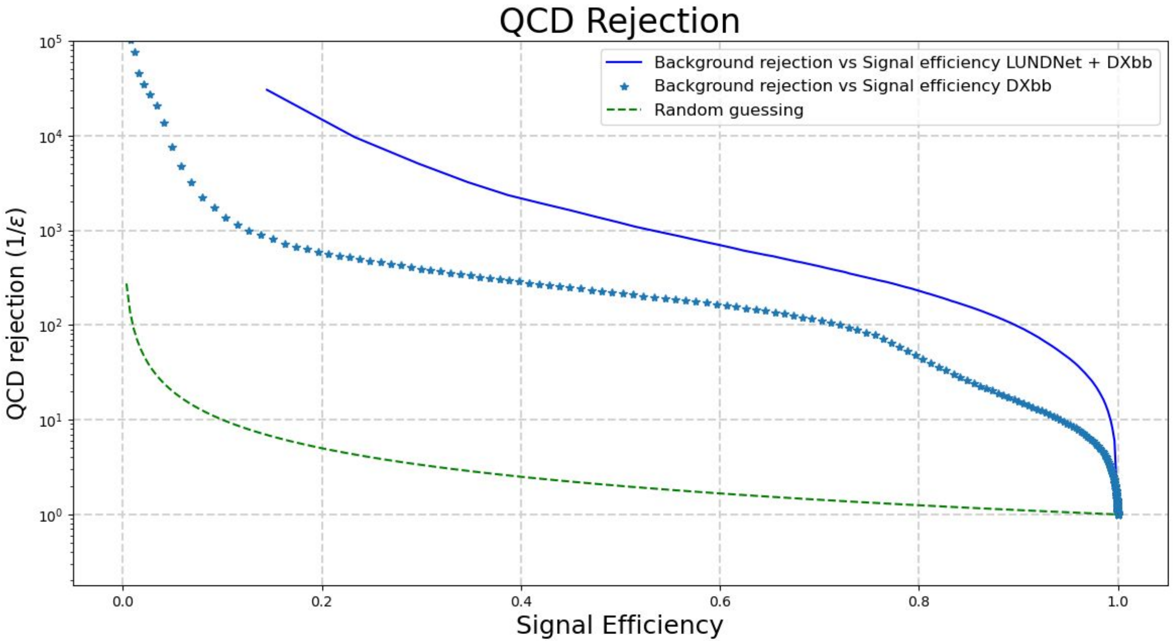




○	76 GeV < Jet mass < 146 GeV	○	Jet truth match with H boson
○	| Jet eta | < 2	○	Number of b Hadrons > 1

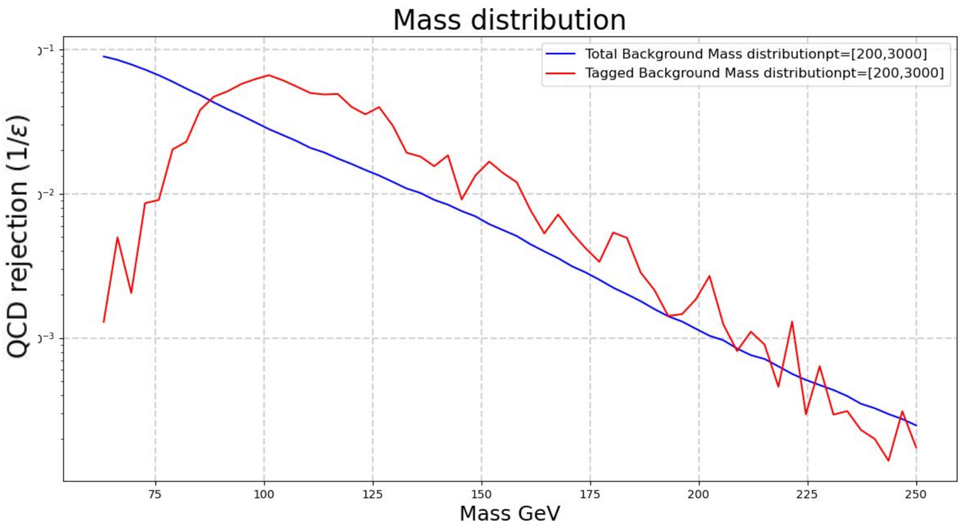
Higgs to bb: Adversarial Results

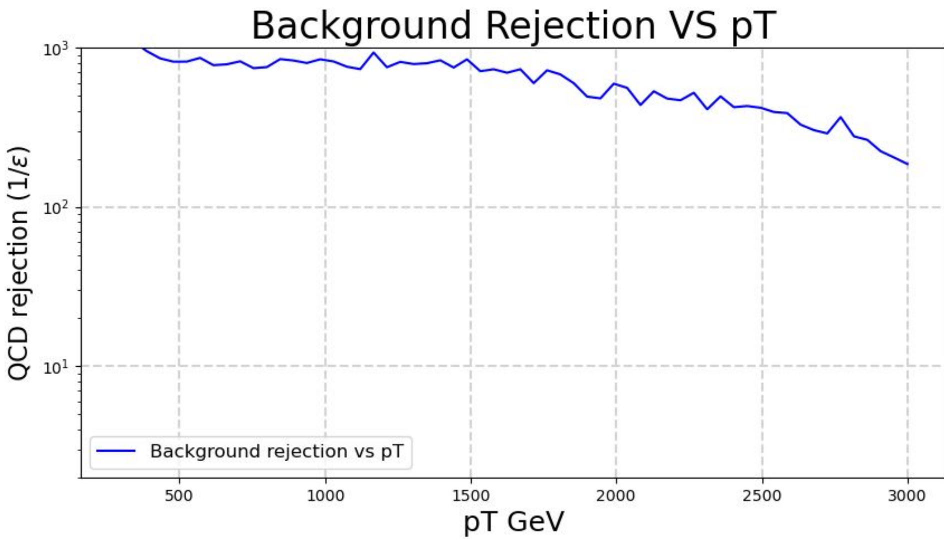
Train size: 2% of dijet background and 40% of H to bb signal. Results for LUNDNet + DXbb tagger



○ 76 GeV < Jet mass < 146 GeV	○ Jet truth match with H boson
○ | Jet eta | < 2	○ Number of b Hadrons > 1

Higgs to bb: Adversarial Results





- 76 GeV < Jet mass < 146 GeV
- | Jet eta | < 2
- Jet truth match with H boson
- Number of b Hadrons > 1

Summary

W tagger:

- Presented 4 GNN architectures with improved performance over the currently boosted W boson taggers.
- Optimized methods outperform current methods by around 50%, however further improvements could be applied.
- Algorithm improvements are still needed to increase the background rejection of the mass decorrelated taggers.

H to bb tagger:

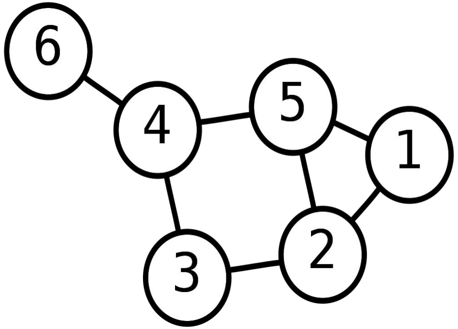
- Using Lund Plane variables to improve the current tagger gives a tagger which it’s around ~10 times better.
- It’s necessary to improve Adversarial + classifier algorithm in order to improve the mass decorrelation of the model.

Thanks for your attention :)

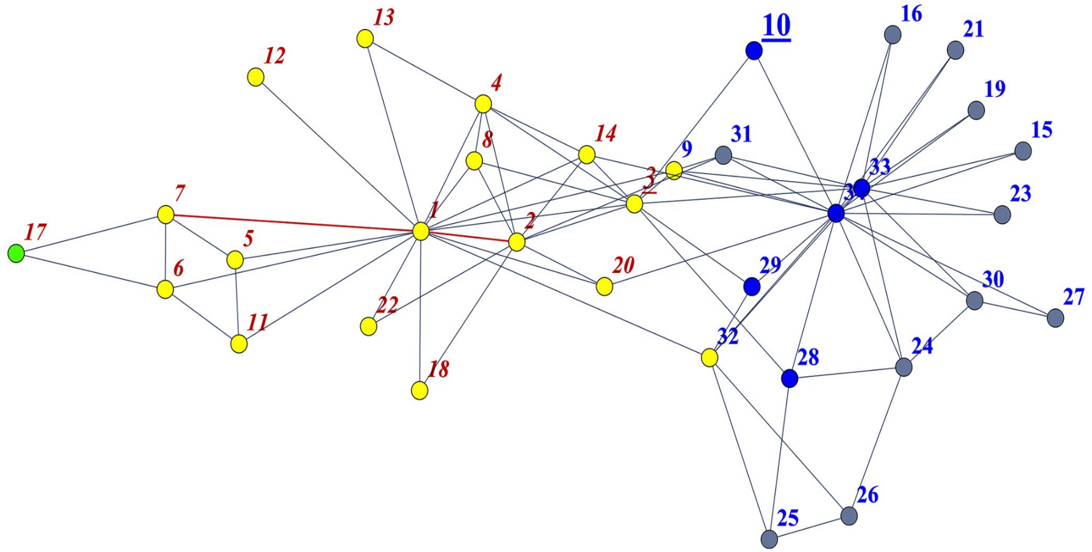
BACKUP ;)

Graph Neural Networks

Garphs: “mathematical structures used to model pairwise relations between objects. A graph in this context is made up of **vertices** (also called *nodes* or *points*) which are connected by **edges** (also called *links* or *lines*)”

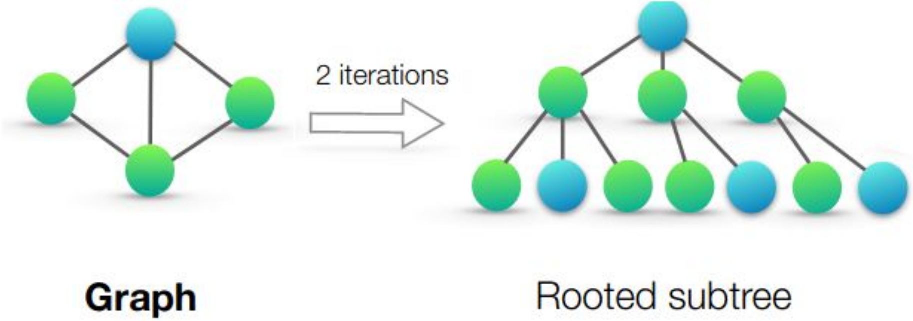


A graph with six vertices and seven edges.



Graph Neural Networks

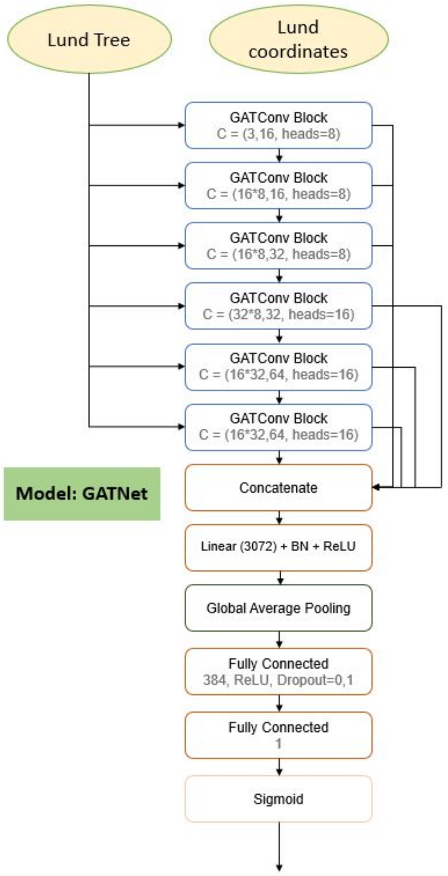
We begin by summarizing some of the most common GNN models and, along the way, introduce our notation. Let $G = (V, E)$ denote a graph with node feature vectors X_v for $v \in V$. There are two tasks of interest: (1) *Node classification*, where each node $v \in V$ has an associated label y_v and the goal is to learn a representation vector h_v of v such that v's label can be predicted as $y_v = f(h_v)$; (2) *Graph classification*, where, given a set of graphs $\{G_1, ..., G_N\} \subseteq \mathcal{G}$ and their labels $\{y_1, ..., y_N\} \subseteq \mathcal{Y}$, we aim to learn a representation vector h_G that helps predict the label of an entire graph, $y_G = g(h_G)$.

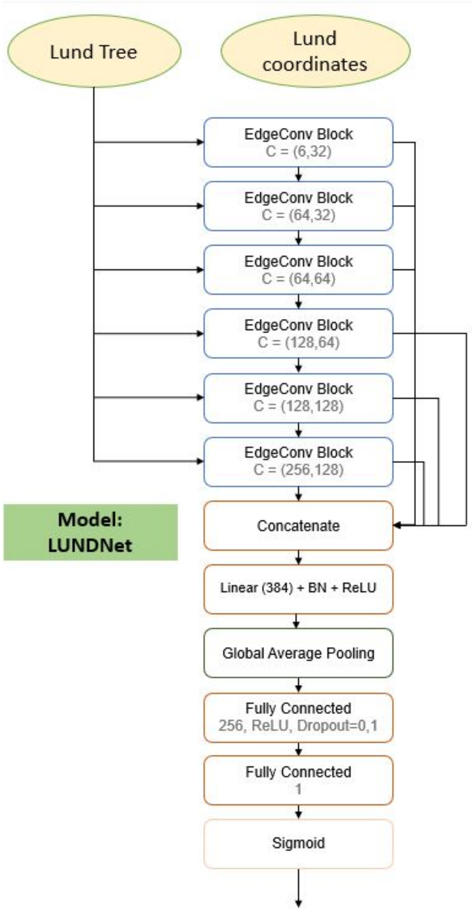


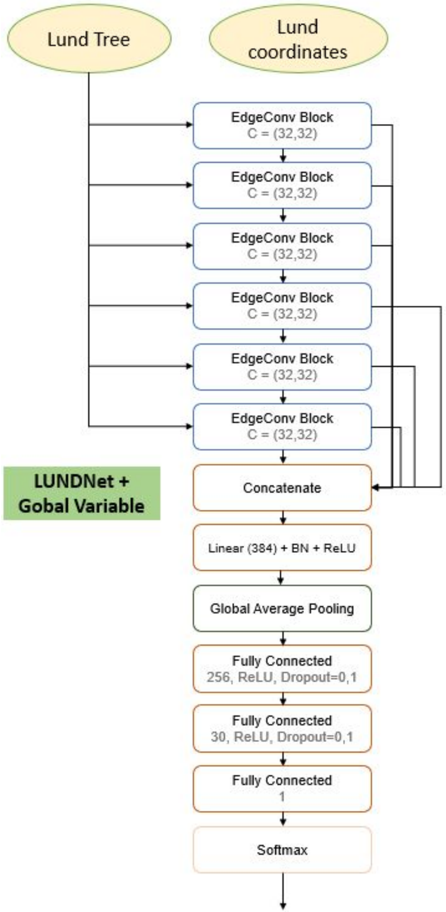
After k iterations of aggregation, a node is represented by its transformed feature vector, which captures the structural information

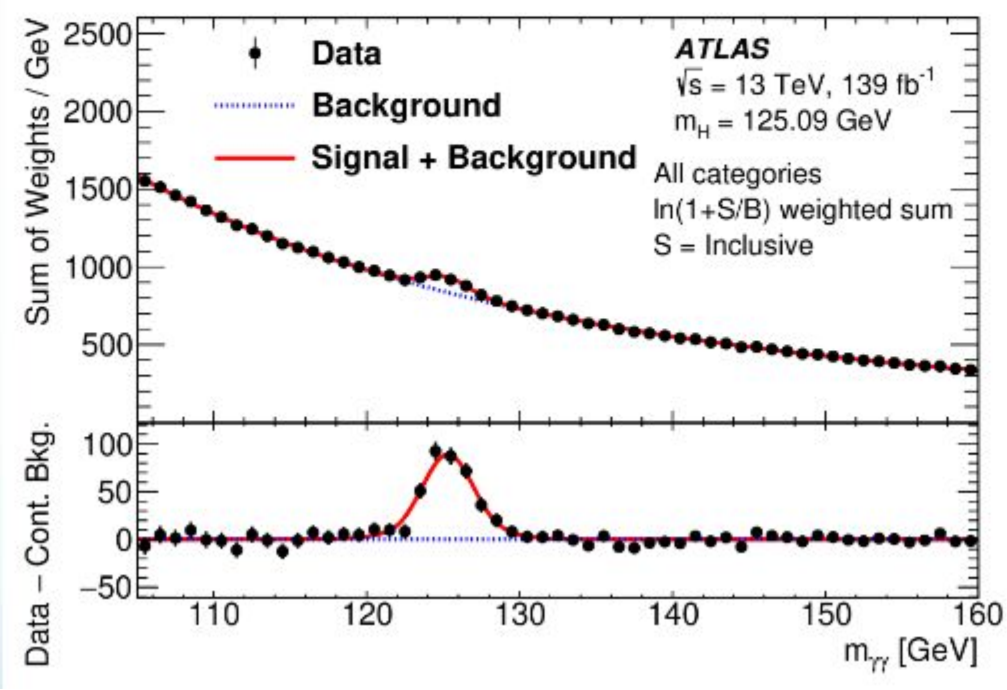
$$h_G = \mathrm{READOUT}\big(\{h_v^{(K)} \mid v \in G\}\big).$$

Plots and definitions taken from: **HOW POWERFUL ARE GRAPH NEURAL NETWORKS?**
https://arxiv.org/pdf/1810.00826.pdf


Lund Tree
Lund coordinates
GATConv Block C = (3,16, heads=8)
GATConv Block C = (16*8,16, heads=8)
GATConv Block C = (16*8,32, heads=8)
GATConv Block C = (32*8,32, heads=16)
GATConv Block C = (16*32,64, heads=16)
GATConv Block C = (16*32,64, heads=16)
Model: GATNet
Concatenate
Linear (3072) + BN + ReLU
Global Average Pooling
Fully Connected 384, ReLU, Dropout=0,1
Fully Connected 1
Sigmoid


Lund Tree
Lund coordinates
EdgeConv Block C = (6,32)
EdgeConv Block C = (64,32)
EdgeConv Block C = (64,64)
EdgeConv Block C = (128,64)
EdgeConv Block C = (128,128)
EdgeConv Block C = (256,128)
Model: LUNDNet
Concatenate
Linear (384) + BN + ReLU
Global Average Pooling
Fully Connected 256, ReLU, Dropout=0,1
Fully Connected 1
Sigmoid

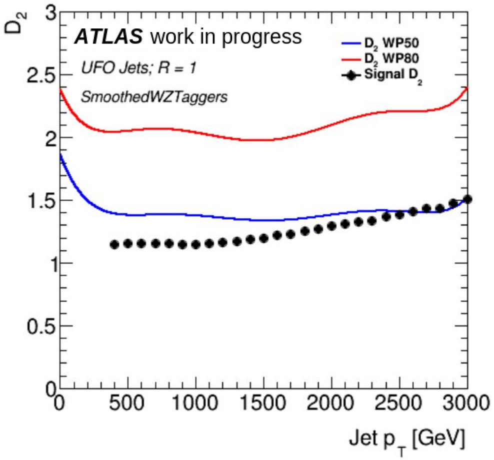

Lund Tree
Lund coordinates
EdgeConv Block C = (32,32)
EdgeConv Block C = (32,32)
EdgeConv Block C = (32,32)
EdgeConv Block C = (32,32)
EdgeConv Block C = (32,32)
EdgeConv Block C = (32,32)
LUNDNet + Gobal Variable
Concatenate
Linear (384) + BN + ReLU
Global Average Pooling
Fully Connected 256, ReLU, Dropout=0,1
Fully Connected 30, ReLU, Dropout=0,1
Fully Connected 1
Softmax

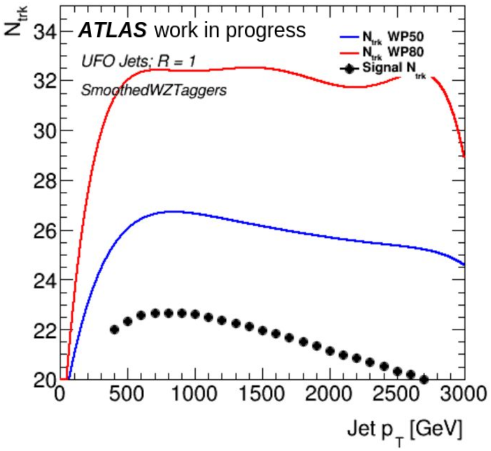


Data
Background
Signal + Background
ATLAS
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
m_H = 125.09 GeV
All categories
ln(1+S/B) weighted sum
S = Inclusive
Sum of Weights / GeV
Data − Cont. Bkg.
$m_{\gamma\gamma}$ [GeV]

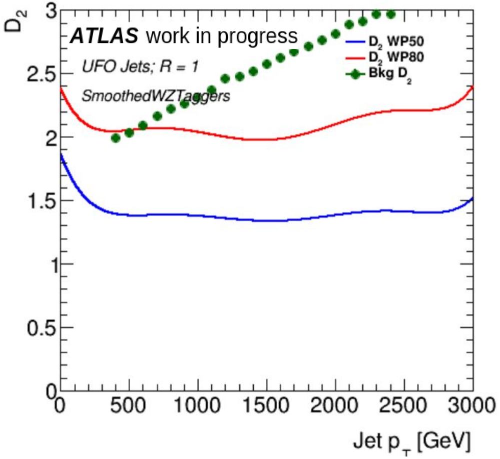
Standard Tagger performance

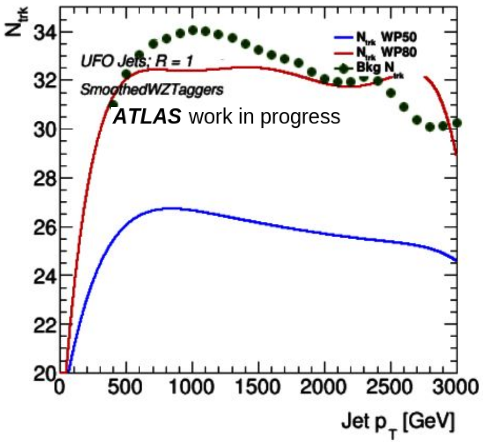
Signal performance



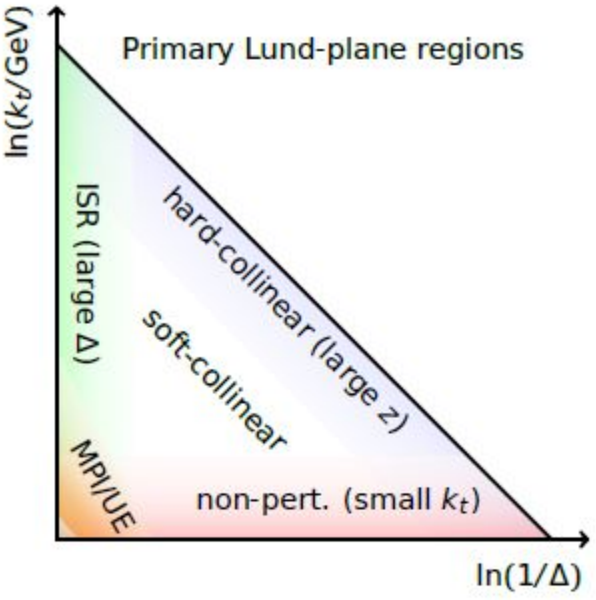


Background performance



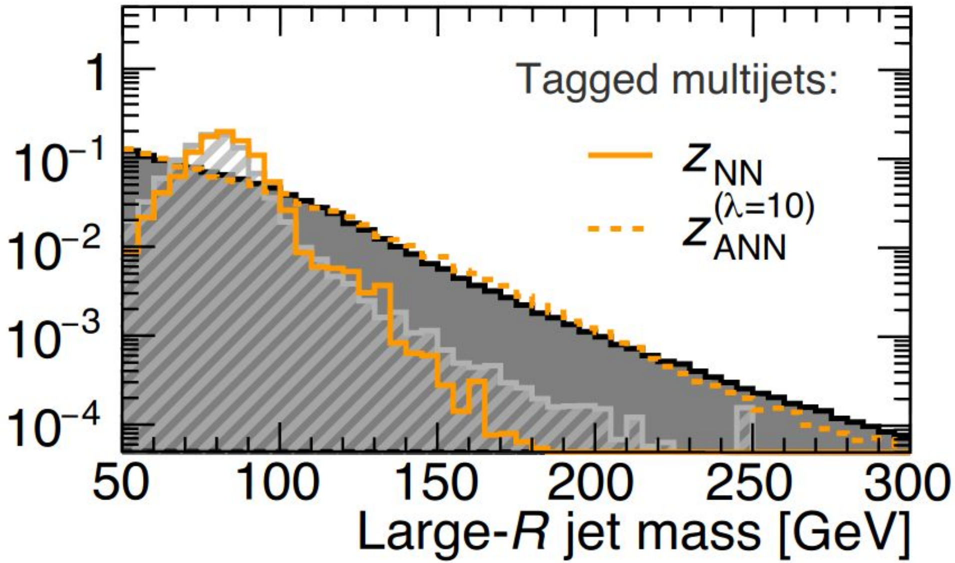


Lund Plane regions



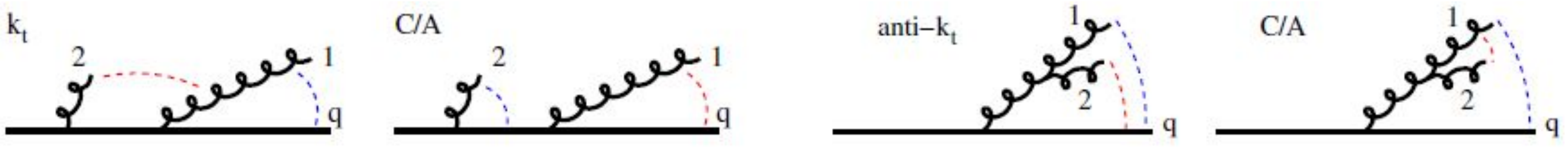
Using **ln(Kt)** and **ln(1/Δ)** is easy to identify differents regions.

Mass sculpting



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

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

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