中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

ML applications at ATLAS

Ke Li (like@ihep.ac.cn)
IHEP, Beijing
2024/04/03

Personal overview: focusing on object identification, anomaly detection and LLM

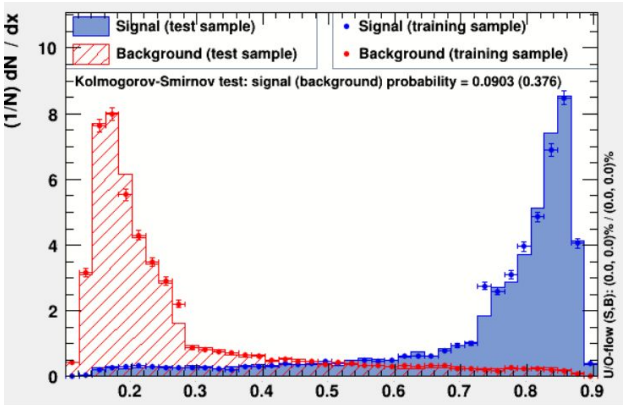
About myself

- 2012-2017: **BESIII**, search for and study of exotic hadron states, e.g. X(3872), Y(2175), Ds0(2317)
- 2017-2019: **CEPC**, simulation of synchrotron radiation and optimization of Machine-Detector-Interface
- 2017-2024: **ATLAS**, online/offline tracking and vertexing, GPU-based accelerator, quark-gluon tagging and search for heavy higgs
- 2019-2024: **FASER**, tracking and detector alignment, observation of collider neutrino and search for long-lived particle
- Now, **BESIII again**, combine **software and physics** to build an **AI scientist**, i.e. Dr. Sai

Outlines

- Introduction to machine learning
- Selected state-of-art ML applications at ATLAS
  - Jet taggers
  - Track and vertex reconstruction
  - Anomaly detection
  - AI assistant
- How about BESIII ?
- Summary

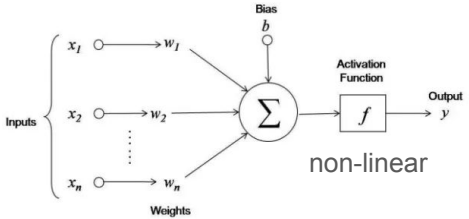
No technical details in this talk

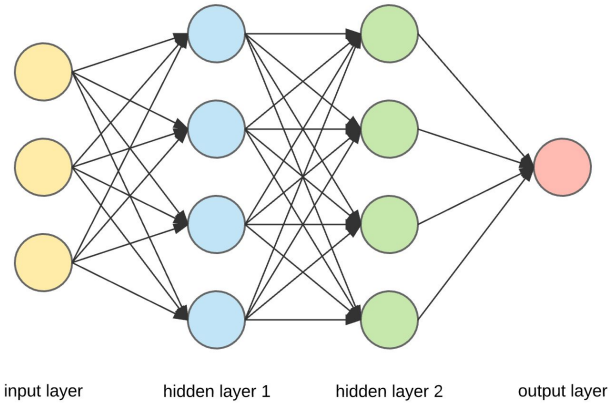


Wide usage in addition to signal-background separation

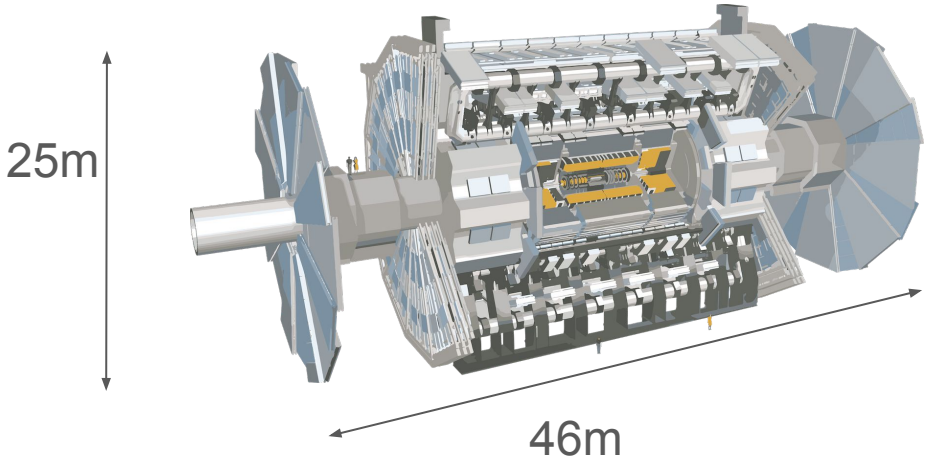
How machine learn

- In CS, neuron is a simple function $\sigma(\sum x_i w_i + b)$
- Neural network: multiple connected neurons
  - An extremely complex function which maps the input to output
  - Millions/Billions float parameters
  - Training: adjust the parameters to make outputs move towards target, e.g. truth, similar to fitting
  - Two key parts:
    - Data representation
    - Function (model)





ATLAS experiment



- The largest detector, will collect the biggest dataset at HL-LHC over the world
- I will focus on three questions:
  - How to reconstruct event correctly
    - 200M readout channels
    - ~50 soft collisions (pile-up) per bunch-crossing
  - How to select event efficiently
    - 2×10^{16} collisions so far, only 1/2,000,000,000 have higgs, much less BSM particles
  - How to make the whole process easier and faster
    - > 3 years for one physics result

→ AI/ML

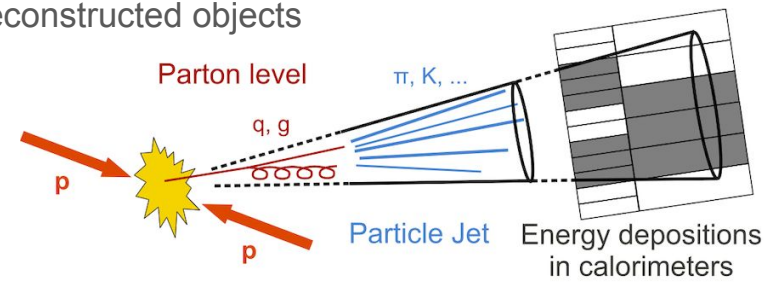
Selected state-of-art ML applications @ ATLAS

- Jet tagger
  - b(bb)-tagger
  - W-tagger
  - Quark-gluon tagger
- Tracking and vertexing
  - GNN-based track finder
  - CNN-ased primary vertex finder
- Anomaly detection
  - CWoLa
  - AutoEncoder
- AI assistant
  - chATLAS
  - Educational outreach

Not covered in this talk but also important:
GNNC for jet/missing-ET calibration, fastGN1 for bjet trigger, ML-based pixel clustering and track seed filter, GAN for fast calorimeter simulation …

Jet-tagging

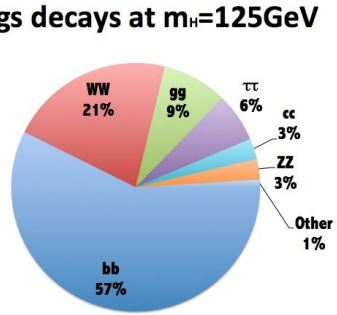
How to get the truth labels from reconstructed objects

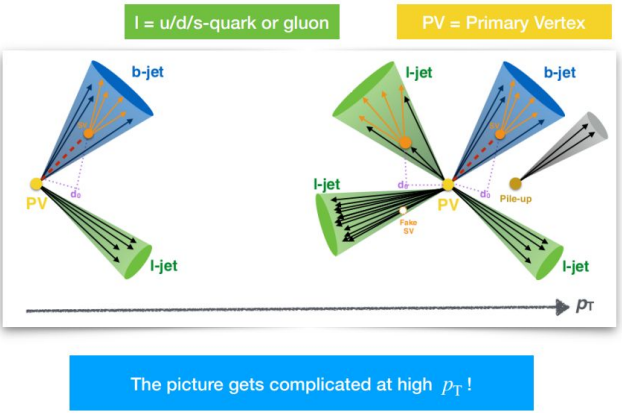


- Convert physics problem to computing problem
- Three directions for improvements
  - More input features : jet -> jet+track -> jet constituents
  - More reasonable data representation : jet+track -> Lund plane
  - More complex model: CNN -> GNN -> Transformer

b-tagger

- H->bb is dominant decay of Higgs
- But difficult to be identified
  - Heavy flavor (b,c) jet are overwhelming minority w.r.t. other jets
- b-jet efficiency and purity is very important in many areas of the physics programme



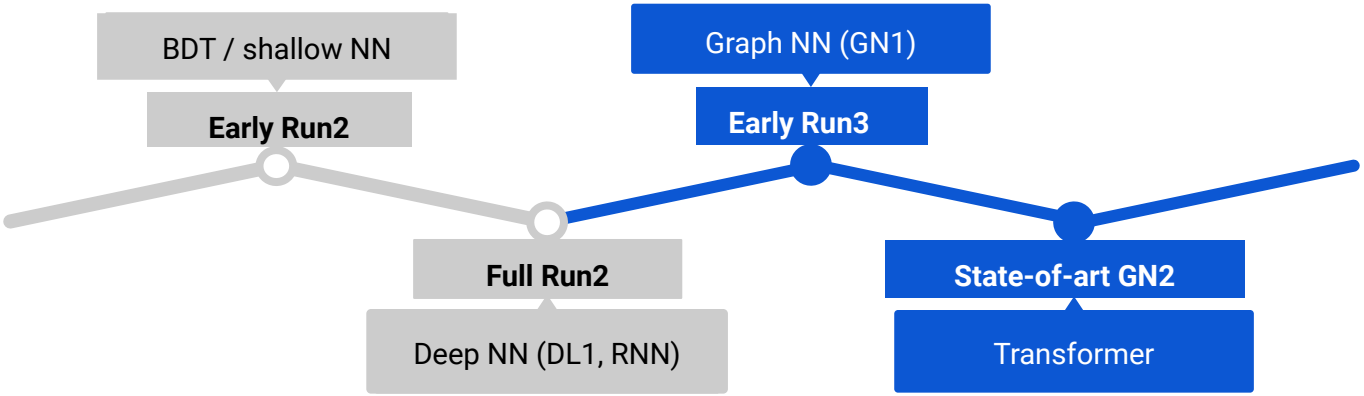


Borrow from S. Stroud

Features of b-jets (relies on B-hadron properties)

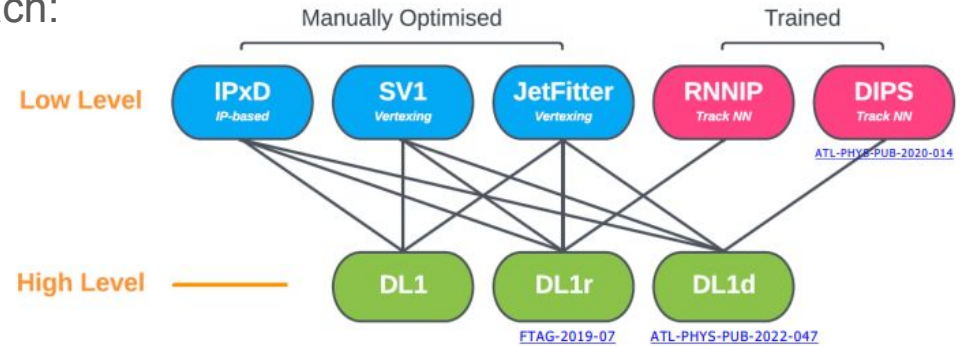
- Relatively long life time
- Displaced (secondary) vertex
- Large impact parameter (d0) tracks
- Large B-hadron mass
- Semileptonic decays (e/μ from b-hadron decay)

History of b-taggers at ATLAS



Two-stage approach:

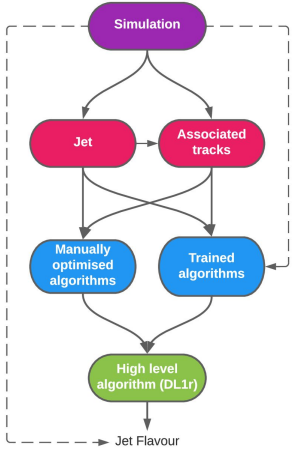
Low + high level algorithms

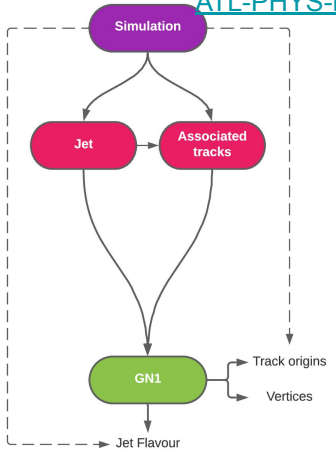


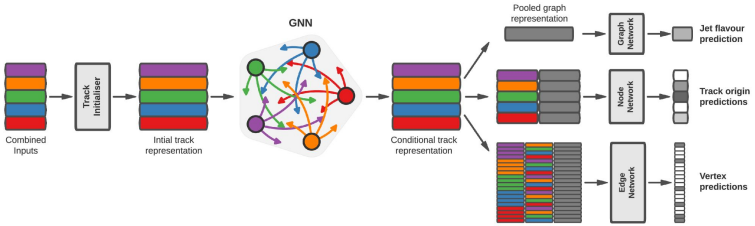
We have new approaches now

GN1

- GNN based all-in-one tagger
- Inputs are jet, tracks and associated hits
  - Jet variables are concatenated with each track
  - Lepton identification for GN1 Lep
  - Treat each track as a node
  - Fully connected, similar to Transformer
- No need for low level algorithm, e.g. secondary vertices
- A factor of 2 improvement in c-jet rejection
- 2 auxiliary tasks
  - Track classification and vertex finding

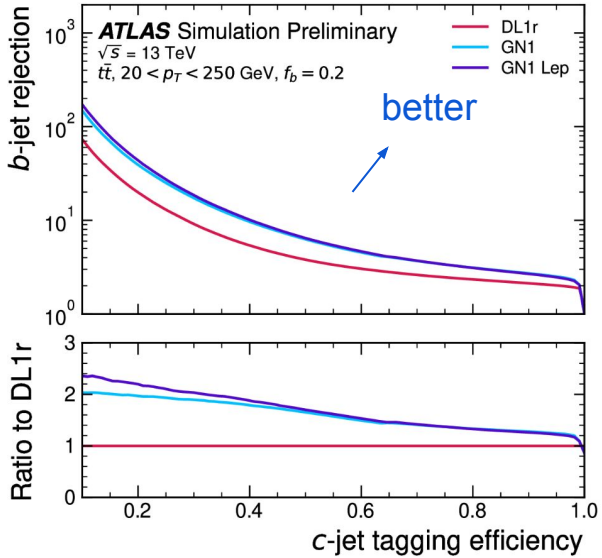






Graph Neural Network, with multihead attention

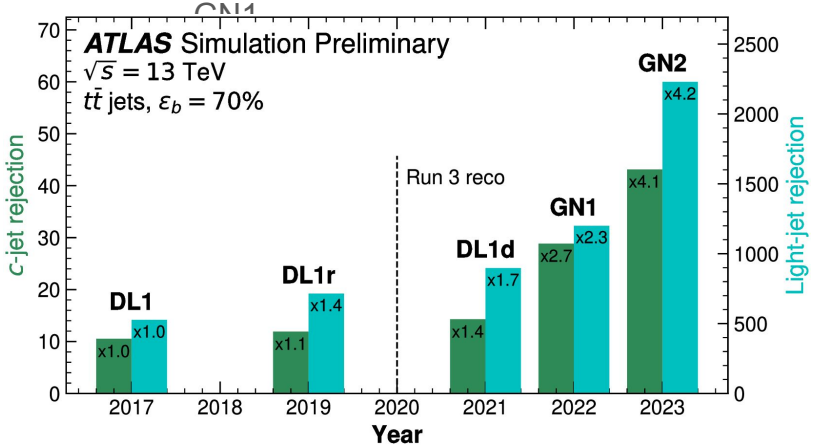
Layers	3
Num nodes	128
Attention heads	2

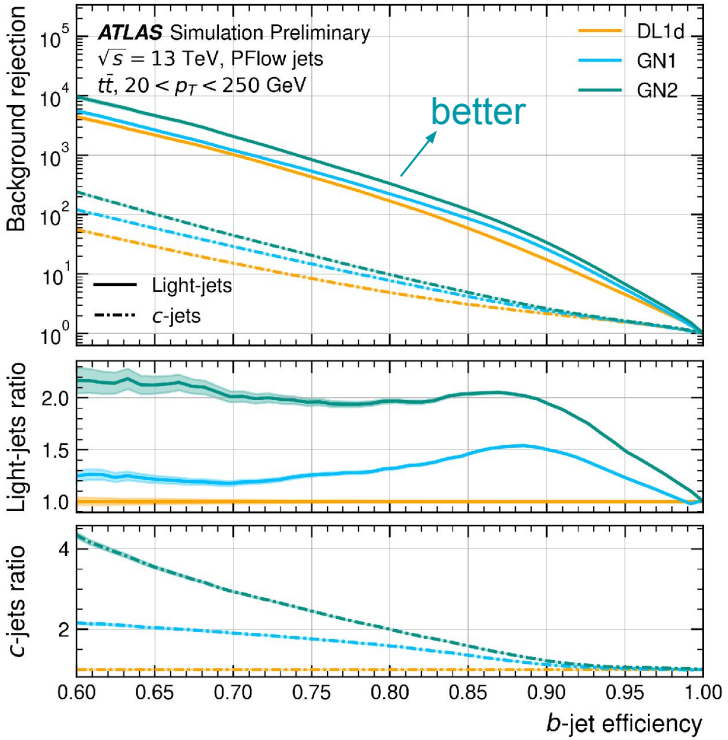


FTAG-2023-01

State-of-art b-tagger: GN2

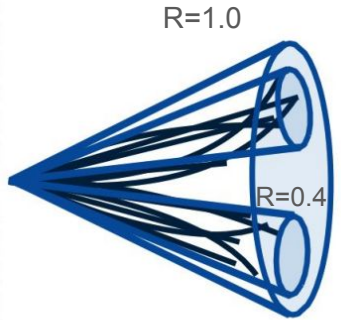
- GN1->GN2
  - Follows transformer architecture more closely, improves the training time and memory footprint
  - Optimisations for the model hyper parameters
  - More layers, more parameters, 0.8M->1.5M
  - A factor of 1.5(2) better c(light) rejection w.r.t. GN1

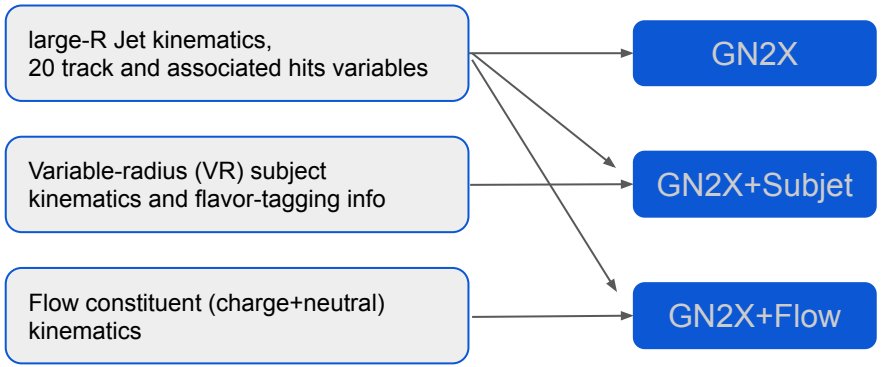




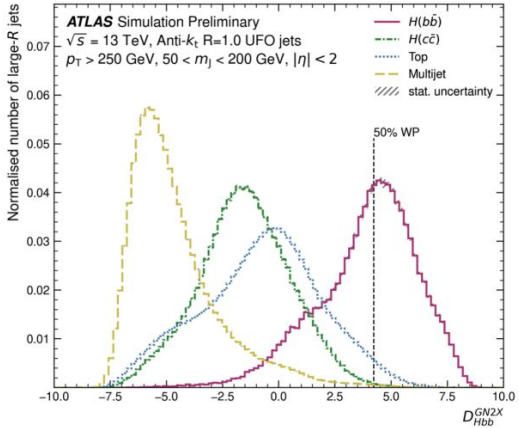
bb tagger: GN2X

- Similar to GN2 but trained on large-radius(R) jet
- Aim to improve improve the sensitivity of H->bb and searches for new resonances Beyond the Standard Model,





The network generates probability scores that indicate the likelihood of a given jet being identified as H->bb, H->cc, top or multijet



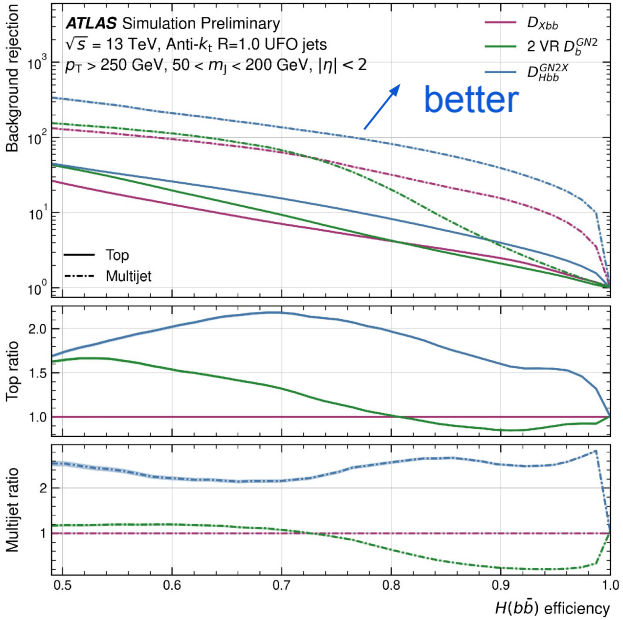
Reference taggers:

- 2 VR : tagger using the same inputs as GN2X but training uses R=0.4 jets
- tagger based on DL1 (used in Run2)

ATL-PHYS-PUB-2023-021

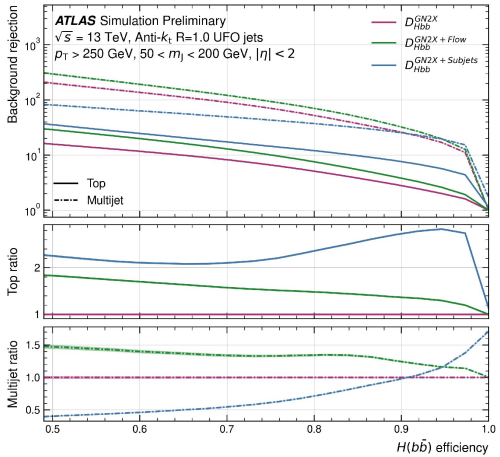
Performance

- Significant improvement from GN2X
- At a 50% H->bb signal efficiency, GN2X provides an increase of 1.6(2.5) in the top jet (multijet) rejection
- GN2X also outperforms the 2-tag VR across all efficiencies



Two heterogeneous input types are tested

- GN2X + subjets provides a 100% increase in the top rejection benefit from the information on the large-R jet substructure, but a reduction in the multijet rejection
- GN2X + Flow has 50% improvement in multijet rejection benefit from neutral jet information
- Further work combining the flow and subjets is therefore warranted



ATL-PHYS-PUB-2023-017

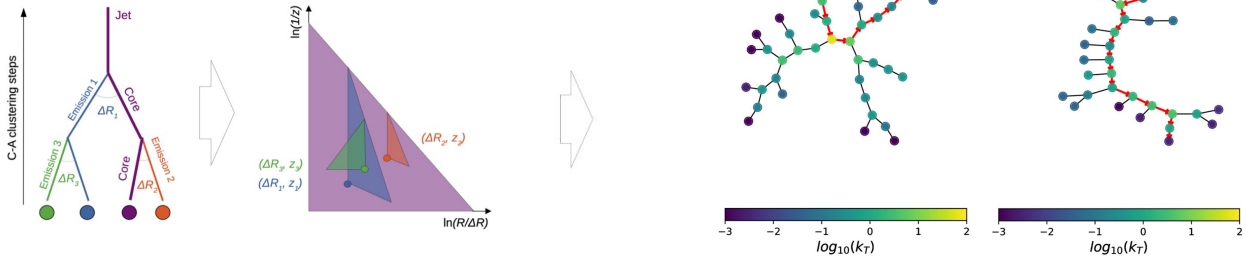
ATL-PHYS-PUB-2023-020

W-taggers

Two approaches

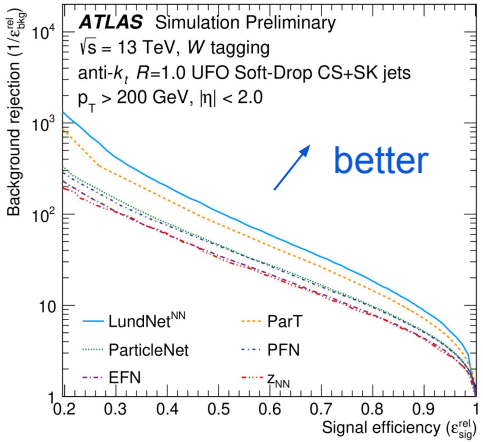
- Jet constituents (ParticleTransformer, ParticleNet, ParticleFlowNetwork, EnergyFlowNetwork)
  - Try to fully utilize jet constituents information using state-of-the-art ML/DL algorithms
  - ParticleTransformer achieves the best performance, see next page
- Lund jet plane (LundNet)
  - Inspired by theory calculation of jet formation
  - Build Lund plane with approximation: core -> hard constituents, emission-> soft constituents
  - Each node has 3 variables: momentum fraction of the branching, transverse momentum, emission angle
  - Each emission represented by a point in the kT-emission angle plane

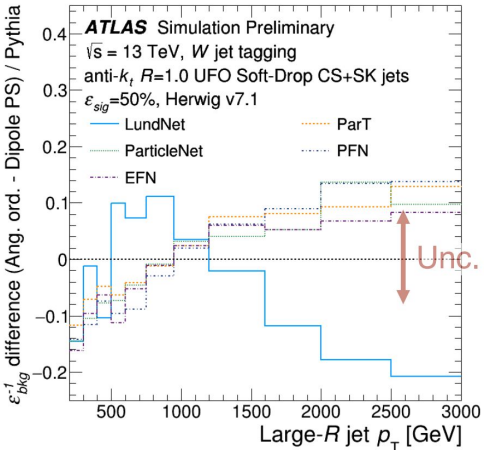
$$\Delta R_{ij} = \sqrt{\Delta y_{ij}^2 + \Delta\phi_{ij}^2}, \qquad z = \frac{p_{\mathrm{T}}^j}{p_{\mathrm{T}}^i + p_{\mathrm{T}}^j}, \qquad k_t = p_{\mathrm{T}}^j \Delta R_{ij}$$

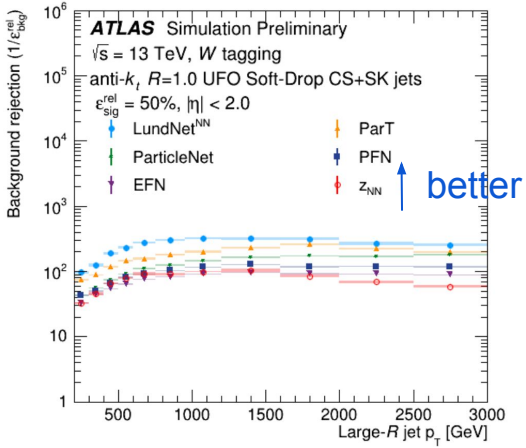


W-taggers

- LundNet achieves the best performance, followed by constituent based tagge
- At 50% signal efficiency, the background rejection of LundNet(ParticleTransformer) is roughly 3(2.8) times better than the previous tagger.
- The more complex or more low-level information the model is/inputs are, the more it is affected by modeling uncertainties
- Next, understand the source of this model-dependence for complex taggers

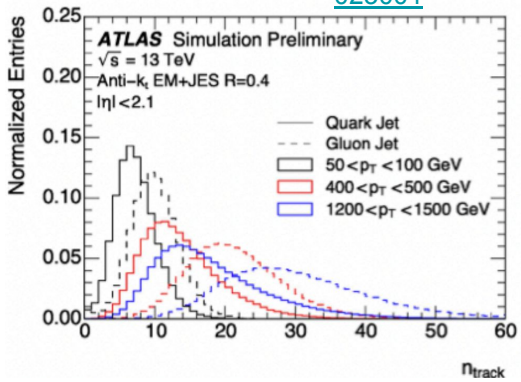




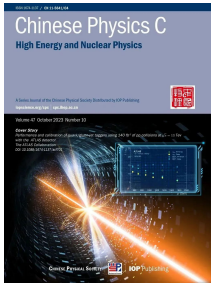


Quark-gluon tagger

Chinese Phys. C 48 (2024) 023001



First ATLAS paper in Chinese journal



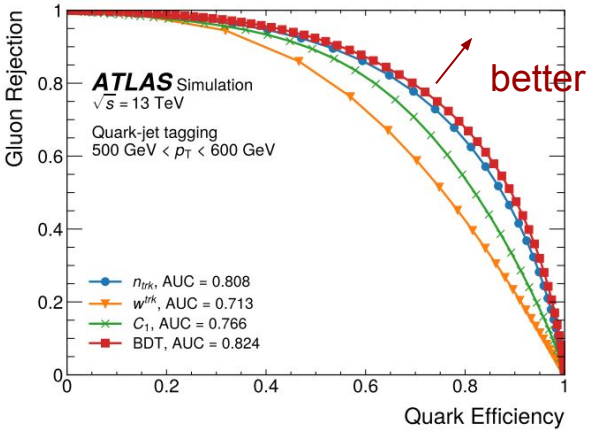
- Quark and gluon jets are difficult to distinguish but many analyses need to know the origin of the jet
- Major discriminator: Gluon jets tend to be wider and have more charged constituents than quark jets

Two taggers are defined and calibrated:

- One based on charged-particle constituent multiplicity
- The second combined several jet kinematic and jet substructure variables using a Boosted Decision Tree (and MLP).
  - BDT outputs the higher AUC
- Both are in-situ calibrated

$$w_{\text{track}} = \frac{\sum_{i\in\text{Jet}} p_{\text{T},i}\Delta R_{i,\text{Jet}}}{\sum_{i\in\text{Jet}} p_{\text{T},i}}, \text{ tracks } i$$

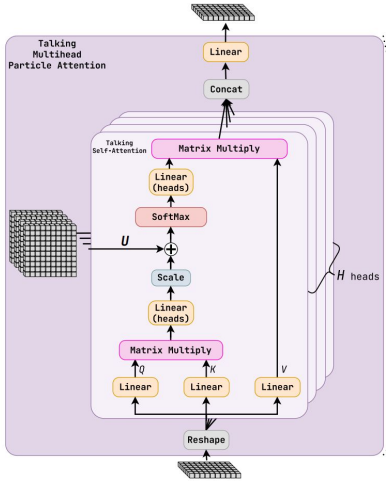
$$C_{1,\text{ track}}^{\beta=0.2} = \frac{\sum_{i,j\in\text{Jet}} p_{\text{T},i}p_{\text{T,j}}(\Delta R_{i,j})^{\beta}}{\left(\sum_{i\in\text{Jet}} p_{\text{T},i}\right)^2}, \text{ tracks } i,j$$

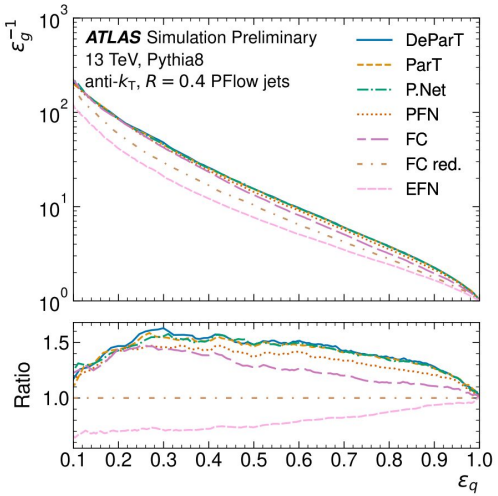


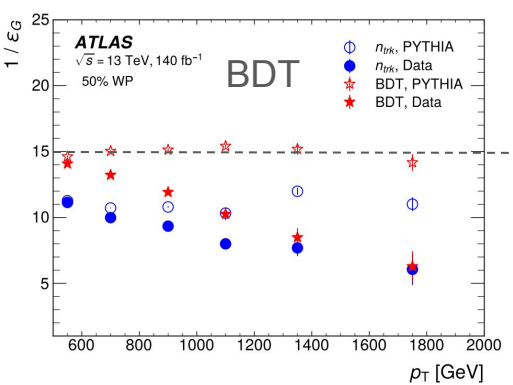
Quark-gluon tagger

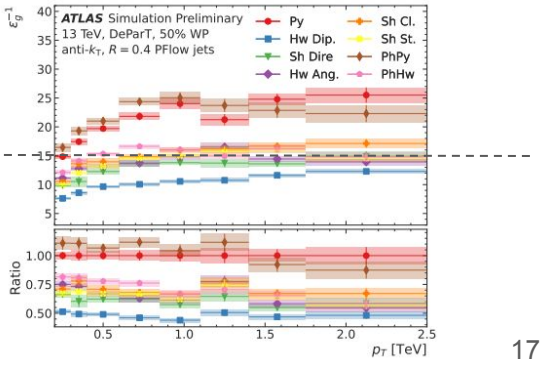
ATL-PHYS-PUB-2023-032

- Similar to W-tagger, constituents based tagger, PFN, EFN, ParticleNet, ParticleTransformer
  - ~2.6M parameters for each model
- New particle attention block (Dynamically Enhanced Particle Transformer, DeParT)
  - Allow heads to communicate
- DeParT and ParT provide the best rejection
  - Obvious improvement w.r.t. BDT
  - But large dependence on MC modeling
    - Same with W-tagger



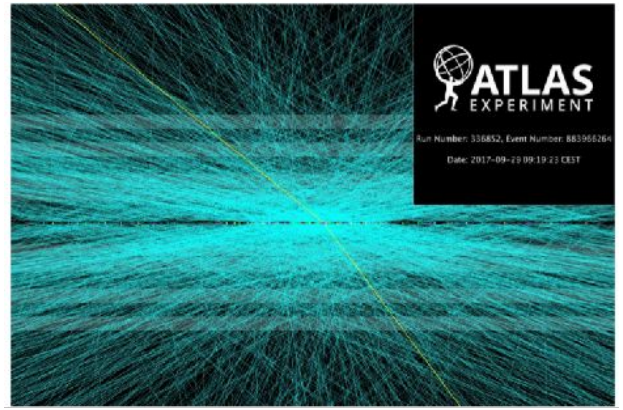






Track and vertex reconstruction

- One of the main input features for jet-tagging - tracks and vertices
- Recently many studies on ML-based track/vertex finder
  - Different problem with jet-tagging
  - More like a clusterization problem
    - Find a cluster of hits to form a track
    - Find a cluster of tracks to form a vertex
  - But one collision could produce ~10k tracks and 200 pile-up
    - How to find them efficiency and with high purity ? and quickly ?
      - Biggest CPU consumer in trigger system
    - Traditional approaches can provide >90% and >90% purity
    - How ML can improve ?
      - GNN-based track finder
      - CNN-based primary vertex (PV) finder

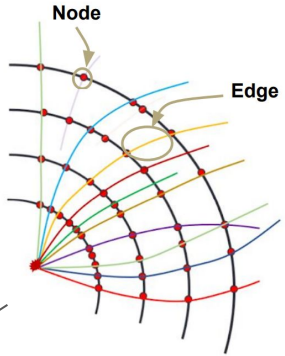


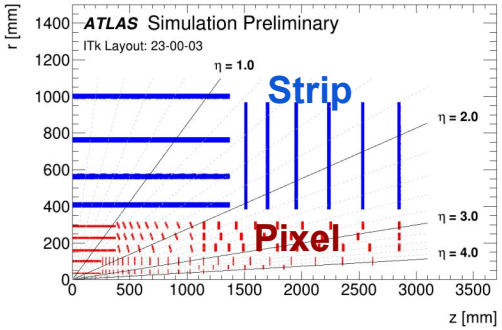
IDTR-2022-01

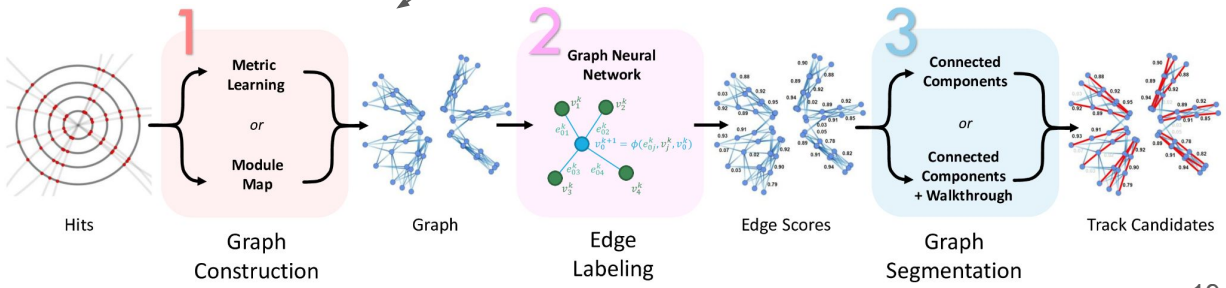
GNN track finder

Convert event to graph:

- Represent each hit as a node
- Connect nodes by edges
  - Edge means two hits belong to the same track
- Edge classification
  - Classifies edges as true or false by assigning score to each edge
- Graph: track candidate, a list of nodes based on edges
- Trained on simulated events in InnerTracker (ITk)



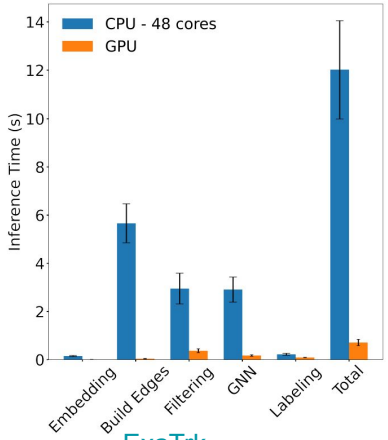




GNN track finder

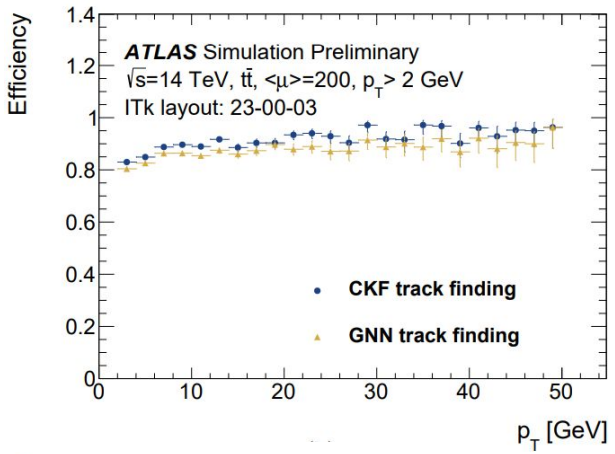
Preliminary comparison to the traditional approach, combinatorial KalmanFilter (CKF)

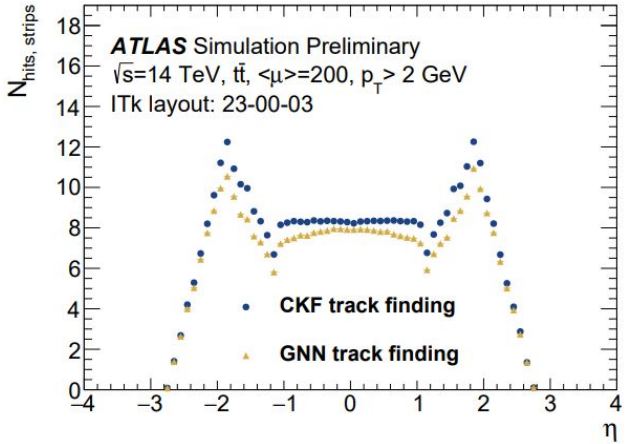
- Slightly lower efficiency, but at the same level
  - Need further optimization
- Less strip clusters
  - Overlap strip modules are not considered yet
- One advantage:
  - Much faster on GPU



ExaTrk

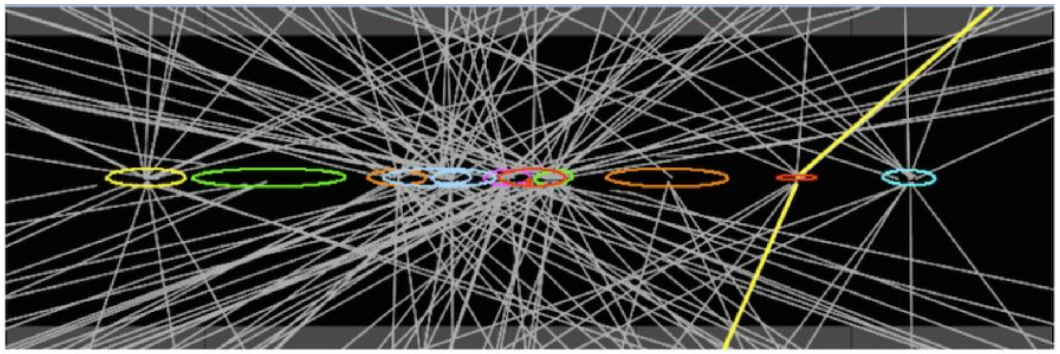
ATL-SOFT-PROC-2023-047





Primary Vertex (PV) finder

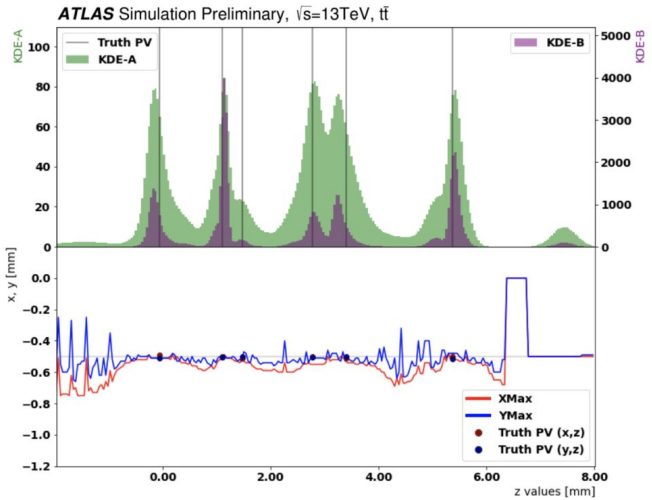
- Among 200 pile-ups, only one hard-scatter vertex
- Crucial for all the downstream processes, e.g. jet tagging, physics analysis
- Best traditional approach:
  - Adaptive Multi-Vertex Finder (AMVF), assign and fit the tracks to vertices
    - >90 efficiency

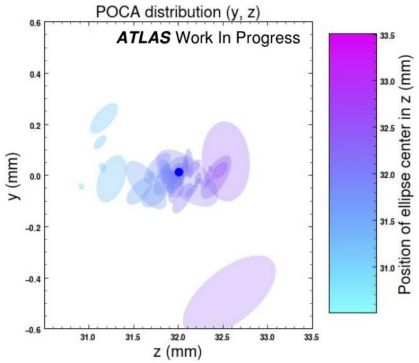


ATL-PHYS-PUB-2023-011

Primary Vertex (PV) finder

- PV-finder, a deep learning primary vertex finder
  - Migrated from LHCb
  - First build track density
    - Kernel Density Estimator (KDE) from track impact parameters and uncertainties
    - 3D point of closest approach (POCA) -> 1D density
  - CNN-based finder with inputs from track density
    - a series of convolutions layers
    - UNet and UNet++ (more connections between layers)



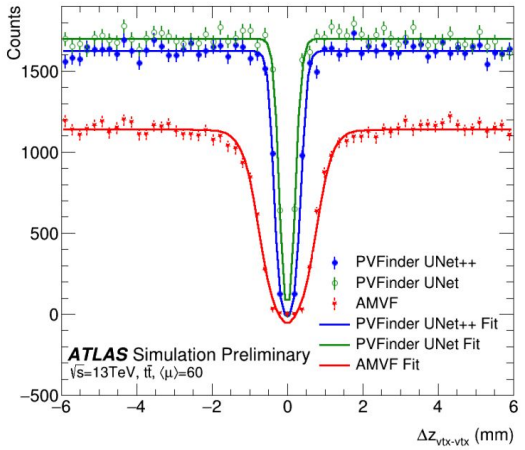


1. `KDE-A`: Sum of track probability values
2. `KDE-B`: Sum of the squares of track probability values
3. `XMax`: Location of the maximum summed track probability in x (mm)
4. `YMax`: Location of the maximum summed track probability in y (mm)

ATL-PHYS-PUB-2023-011

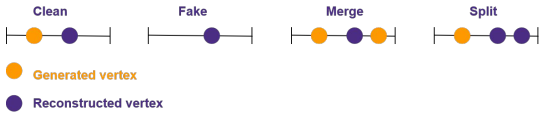
Preliminary results

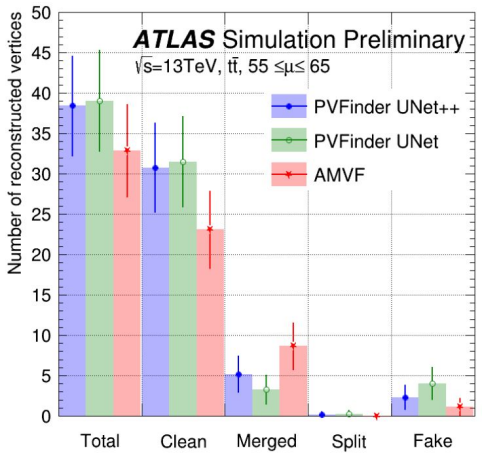
- Initial results look promising
- Compared to AMVF
  - Better vertex-vertex resolution, ~0.8 -> 0.3 mm
  - More vertices found
    - More clean vertices, more fakes as well
  - But slower (not a issue for vertexing)
- Next:
  - Study the hard-scatter vertex selection
  - GNN-based PV finder



Model	Efficiency	False Positive Rate (# / event)
PV-Finder UNet++	94.2%	1.5
PV-Finder UNet	88.7%	2.6
AMVF	93.9%	0.8

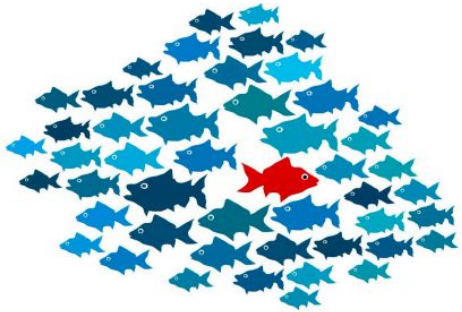
clean+merge





Anomaly detection

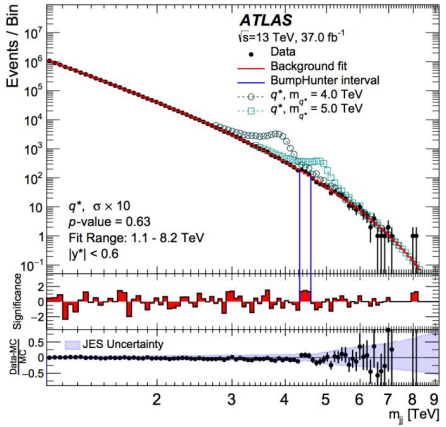
- Traditional search
  - Looking for specific physics motivated signal
  - Not very useful for other models
- ML-based model-independent search for BSM
  - Look for unknown from known events
  - Less sensitive to specific model
- At ATLAS
  - Classification WithOut Labels (CWoLa)
  - Anomaly detection using unsupervised ML

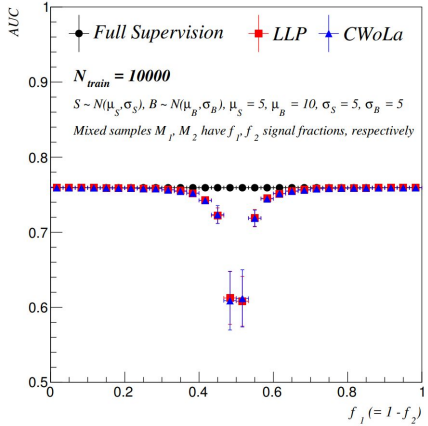


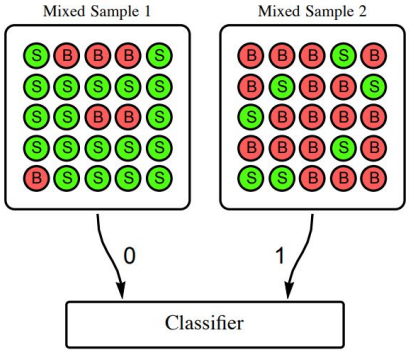
Classification WithOut Labels (CWoLa)

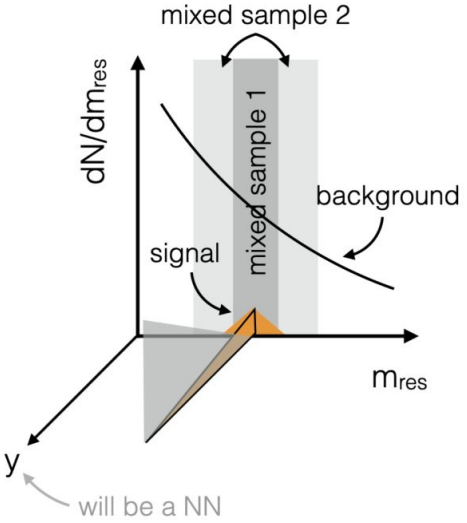
- New bump hunting technique
- Builds on generic dijet search to be more sensitive to a broad class of models
- No need of MC info
- Trained on two mixed samples (real data) which have different fractions of signal and background
  - Fractions could be known (LLP) or unknown (CWoLa)
  - No sensitivity if fractions are the same in 2 regions

JHEP 10 (2017) 174









PhysRevLett.125.131801

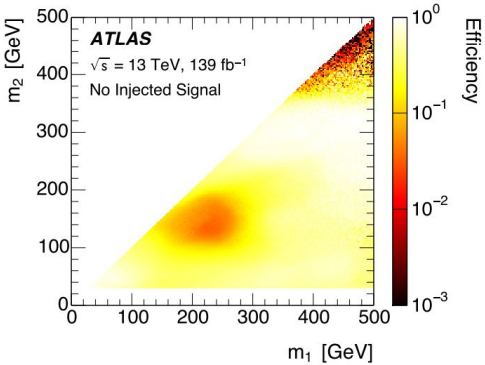
Universe 2022, 8(10), 494

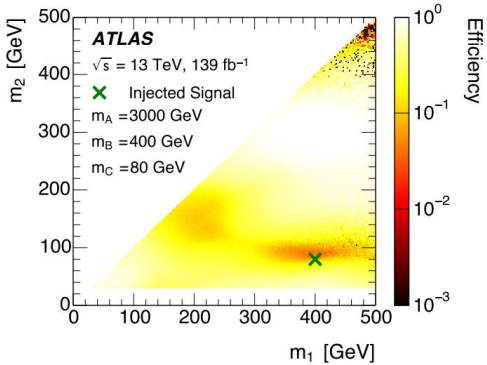
Classification WithOut Labels (CWoLa)

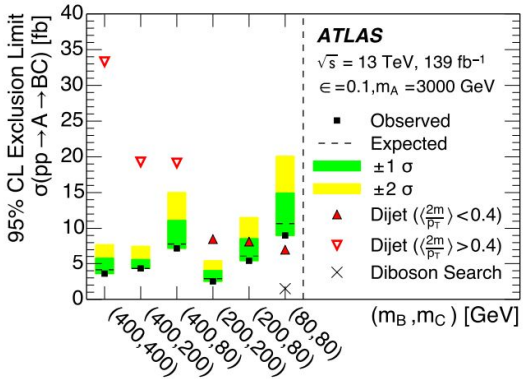
- Performed on dijet search
  - Slicing window in mjj as signal region and two neighboring sideband regions
- Thresholds applied on NN outputs = Efficiency
- Apply classifier score cuts: enhance signal sensitivity
- Fit the dijet mass over the NN score threshold (eff = 0.1/0.01)

- Limits for some signals >2x better
- Dedicated diboson searches show greater sensitivity
  - But no sensitivity at other mass points
  - CWoLa is sensitive at everywhere

NN output vs. inputs jet masses, it is able to catch the signal



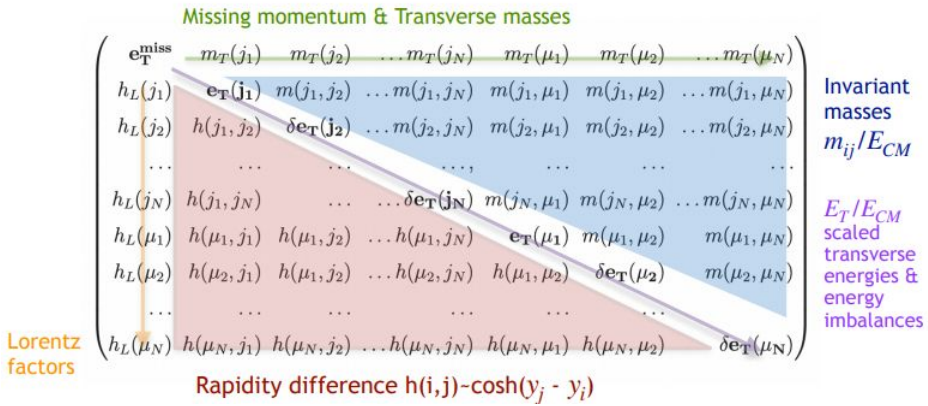




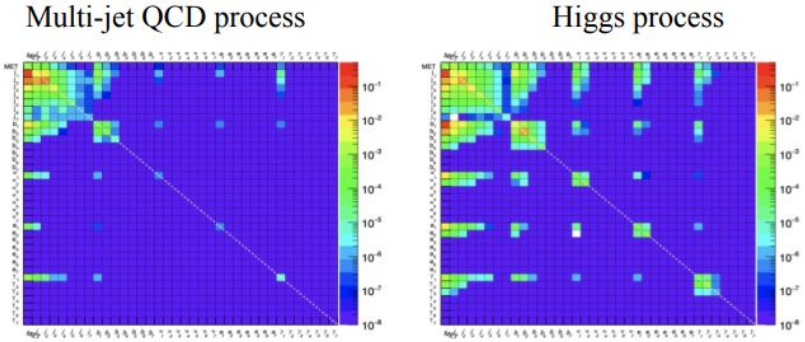
PhysRevLett.132.081801

Anomaly detection using unsupervised ML

- Model independent search
- Use ML to define anomaly region
  - Train the NN to learn the SM events from data
  - Input features are Rapidity Mass Matrix constructed from final states object kinematics
    - RMM was found to produce more robust AutoEncoder(AE) training
    - Expected to have different characteristics for different processes



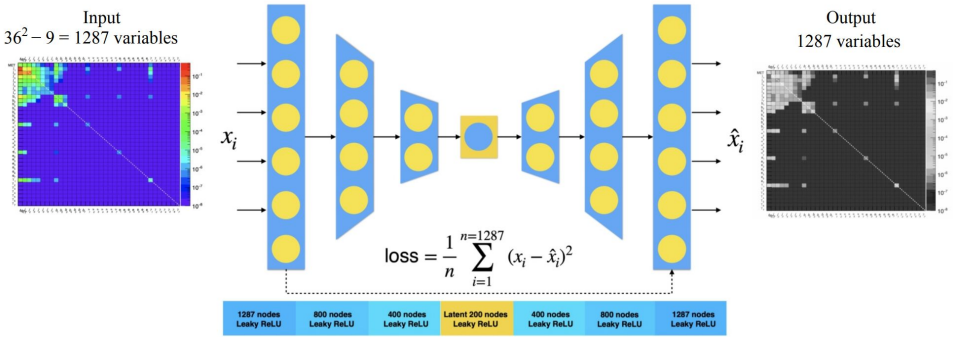
Example

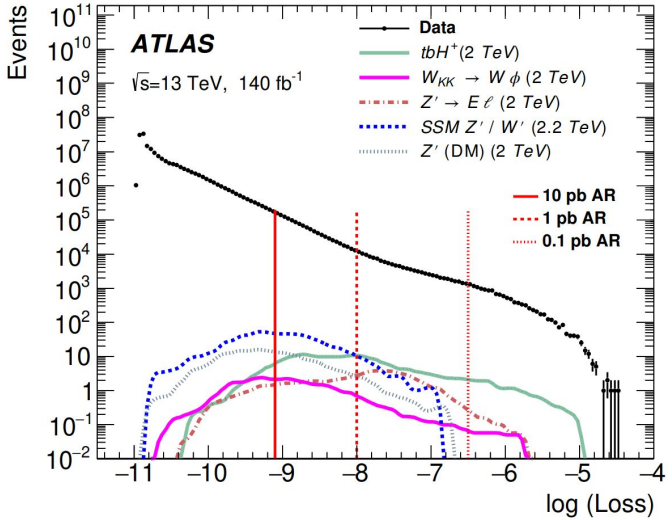


PhysRevLett.132.081801

Anomaly detection using unsupervised ML

- Trained with randomly selected 1% events
  - Sufficient statistics to train and well represent the full collision dataset
  - Expected to have no anomaly (signal) events
  - Even if there are, shape of anomaly score is not expected to produce bumps, so search for the enhancements should not be affected significantly
- Define anomaly region
  - Should enhance BSM signal
  - 3 regions for different assumption on cross sections
- Then do bump-hunting on 2 object mass spectrum

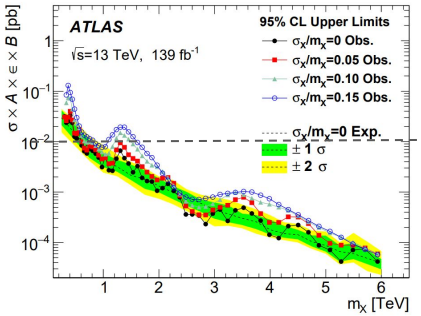


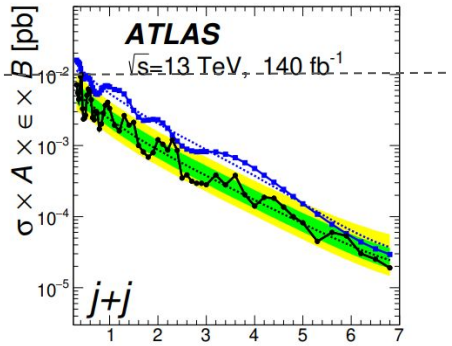


PhysRevLett.132.081801

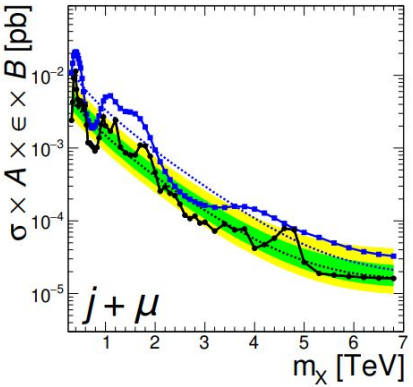
Anomaly detection using unsupervised ML

- Searched in 9 invariant masses including j+j, b-jet+j, j+e/μ
- Largest deviation reported by BumpHunter is at m(j+μ) ~ 4.8 TeV, local significance 2.9σ
- For j+j, the limits are factor 2-3 better than a dedicated search
- Demonstrated the successful application of unsupervised anomaly detection using event level info





[Dedicated search for jj]

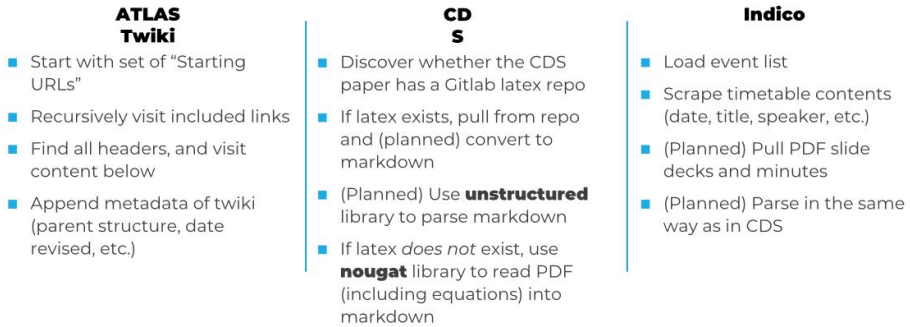


AI Assistant

- Many generic LLM models, e.g. GPT/LLaMa/Gemini
  - astonishing capabilities in recognition and generation of text/code
- How it can help us
  - Text-related works
  - Coding and debugging
  - Understand heterogeneous sources of knowledge
    - Assistant, could be a teacher
- chATLAS
- Educational outreach

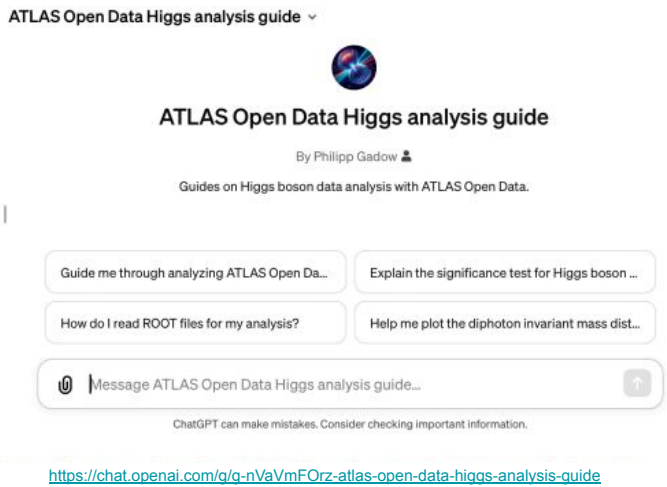
AI assistant: chATLAS

- An AI Assistant for the ATLAS Collaboration, chATLAS
- Inspired by the chatGPT
- Motivation:
  - An assistant which understand the internal heterogeneous sources of knowledge, e.g. twiki, indico, cds
  - Provide quick and accurate search result and summary
  - Long-term plan: debugging software
- Use GPT3.5/4 as backend
- PDF files from cds/indico scraping shown with Nougat and Marker
- Status:
  - Prototype finished
  - Preparing more data



AI assistant for educational outreach

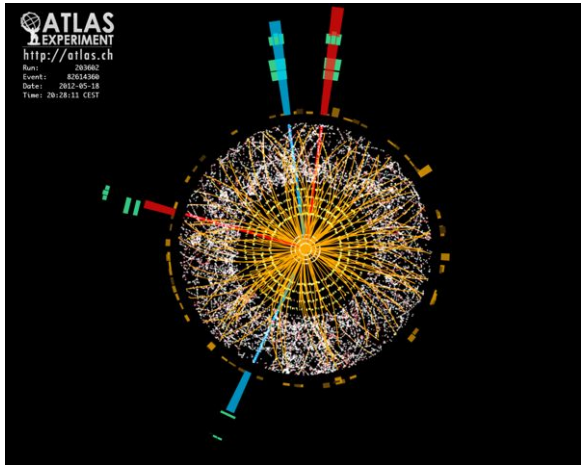
- Goal: large-language model assistant to help people without expert knowledge (junior student) to analyze ATLAS open data
- Status:
  - Customised GPT for ATLAS Diphoton Open Data based on GPT 4.0 model

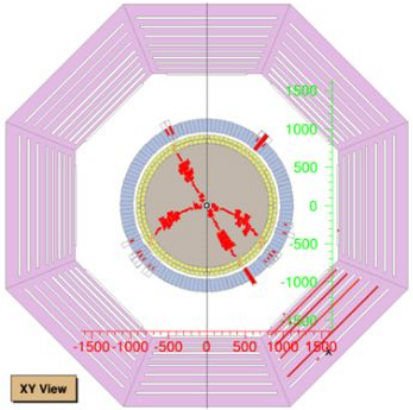


https://chat.openai.com/g/g-nVaVmFOrz-atlas-open-data-higgs-analysis-guide

- **Initial prompt:** *guide students through data analysis of ATLAS Open Data to identify Higgs boson candidates in a dataset which contains proton proton collision events with two photons. Students should design their python data analysis which reads in ROOT files, creates plots and performs significance tests to extract the Higgs boson mass from a fit of the di-photon invariant mass.*
- Emphasis on python and scikit-hep ecosystem
- Crawled websites of:
  - open data and its documentation
  - introduction to di-photon analysis link
  - example for analysis with ROOT link
  - HSF tutorials for software tools link
- Provided structure and data format of CERN Open Data files

How about BESIII





- Very successful ML application at ATLAS
  - Jet tagging, event classification
- At BESIII
  - no jet, much more clean environment than ATLAS-> will not gain too much from ML
  - One possible direction: build AI assistant/scientist using Large Language Model, i.e. Dr. Sai (more details in slides)
    - Heterogeneous inputs: BESIII hypernews, internal memo/draft/indico/source code, arXiv papers, HaiChat history …
    - Functionality: > chATLAS + educational outreach assistant
    - ~ 10 active developers

Summary

- Now ML is widely used at ATLAS in addition to signal-background separation
  - Jet-tagger (overwhelming better than traditional tagger)
    - More low level features + more complex model (Transformer) always give best result
    - But large uncertainty due to MC modeling
  - Track and Vertex finder
    - Approaching to traditional CKF and AMVF
    - Could be much faster from introducing GPU
  - Anomaly detection
    - Generic search, sensitive to a wide range of BSM models
    - Sensitivity comparable to dedicated search
- Next
  - AI assistant based on Large Language Model (LLM)
    - chATLAS, outreach assistant
    - Dr. Sai (preliminary version is coming, stay tune)

back-up

Quark-gluon tagger

Constituent Variables
$\Delta\eta = \eta - \eta^{\text{jet}}$
$\Delta\phi = \phi - \phi^{\text{jet}}$
$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$
$\log p_{\text{T}}$
$\log E$
$\log \frac{p_{\text{T}}}{p_{\text{T}}^{\text{jet}}}$
$\log \frac{E}{E^{\text{jet}}}$
m

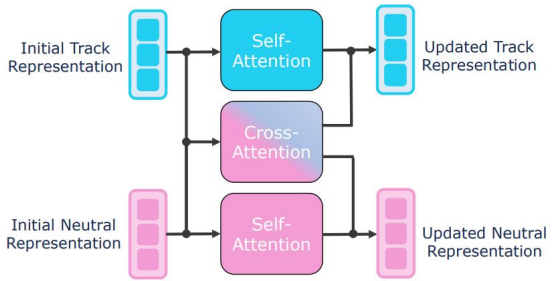
Constituent Interaction Variables
$\log \Delta^{ab} = \log \sqrt{(\eta^a - \eta^b)^2 + (\phi^a - \phi^b)^2}$
$\log k_{\text{T}}^{ab} = \log \left(\min \left(p_{\text{T}}^a, p_{\text{T}}^b\right)\Delta^{ab}\right)$
$z^{ab} = \min \left(p_{\text{T}}^a, p_{\text{T}}^b\right)/(p_{\text{T}}^a + p_{\text{T}}^b)$
$\log m^{2,ab} = \log \left(p^{\mu,a} + p^{\mu,b}\right)^2$

Linear Constituent Variables
$\Delta\eta = \eta - \eta^{\text{jet}}$
$\Delta\phi = \phi - \phi^{\text{jet}}$
$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$
$\frac{p_{\text{T}}}{p_{\text{T}}^{\text{jet}}}$

FTAG-2023-01

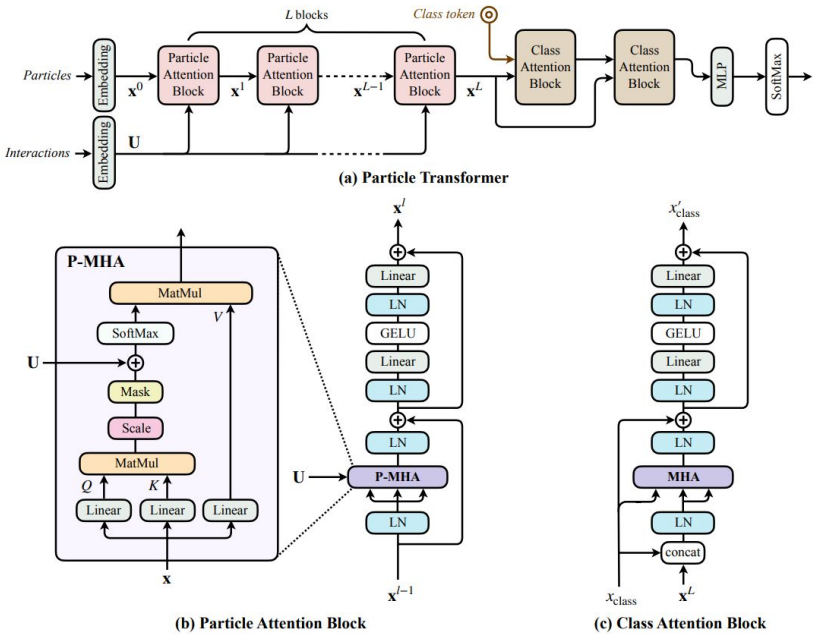
ATLAS: GN1 -> GN2 difference

Type	Name	GN1	GN2
Hyperparameter	Trainable parameters	0.8M	1.5M
Hyperparameter	Learning rate	$1e{-}3$	OneCycle LRS (max LR $4e{-}5$)
Hyperparameter	GNN Layers	3	6
Hyperparameter	Attention Heads	2	8
Hyperparameter	Embed. dim	128	192
Architectural	Attention type	GATv2	ScaledDotProduct
Architectural	Dense update	No	Yes (dim 256)
Architectural	Separate value projection	No	Yes
Architectural	LayerNorm + Dropout	No	Yes
Inputs	Num. training jets	30M	192M



ParticleTransformer

arXiv:2202.03772



UNet and UNet++