Flavor Tagging Using Graph Neural Networks

Constructing and Calibrating GNNs for Flavor Tagging Jets in ATLAS and CMS

Luke Vaughan
Oklahoma State University
Today’s discussion will include:
- Motivation for b-tagging
- Machine Learning in HEP
- Data Collection and Jet Reconstruction
- Deeps Sets and Message Passing Neural Networks
- Current GNN Implementations in ATLAS and CMS
- Calibration of Taggers
- Unfolding
Why b-tagging?

B-tagging is important for any physics process that includes b-jets in their final state.

Most notably, both Higgs Boson and Top Quark have large branching ratios to b quark, which demands that experimentalist focus their attention on b-tagging.

Table 11.3: The branching ratios and the relative uncertainty [44, 45] for a SM Higgs boson with $m_H = 125$ GeV.

<table>
<thead>
<tr>
<th>Decay channel</th>
<th>Branching ratio</th>
<th>Rel. uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H \to \gamma\gamma$</td>
<td>$2.27 \times 10^{-3}$</td>
<td>$+5.0%$ $-4.9%$</td>
</tr>
<tr>
<td>$H \to ZZ$</td>
<td>$2.62 \times 10^{-2}$</td>
<td>$+4.3%$ $-4.1%$</td>
</tr>
<tr>
<td>$H \to W^+W^-$</td>
<td>$2.14 \times 10^{-1}$</td>
<td>$+4.3%$ $-4.2%$</td>
</tr>
<tr>
<td>$H \to \tau^+\tau^-$</td>
<td>$6.27 \times 10^{-2}$</td>
<td>$+5.7%$ $-7.7%$</td>
</tr>
<tr>
<td>$H \to bb$</td>
<td>$5.84 \times 10^{-1}$</td>
<td>$+3.2%$ $-3.8%$</td>
</tr>
<tr>
<td>$H \to Z\gamma$</td>
<td>$1.53 \times 10^{-3}$</td>
<td>$+9.0%$ $-8.9%$</td>
</tr>
<tr>
<td>$H \to \mu^+\mu^-$</td>
<td>$2.18 \times 10^{-4}$</td>
<td>$+6.0%$ $-5.9%$</td>
</tr>
</tbody>
</table>

Source: PDG
B-Hadrons have many distinct properties such as long lifetime, high number of tracks produced in its decay, as well as displaced secondary vertex compared to primary vertex.

Typically, a B-hadron will travel several millimeters before decaying which causes the formation of a displaced secondary vertex.
As can be seen in the histograms above, b jets show significant separation for transverse, d0, and longitudinal, z0, impact parameter significance.
Effective B-Tagging Methods

To maximize b-jet efficiency vs background rejection, it is useful to explore ML methods.

History of B-Tagging: Slide 14 here.

Source: ATLAS FTAG Public Plots
HEP currently uses ML in nearly all aspects of experimental, phenomenological, and theoretical analyses.

ML is currently used in b-tagging, tau ID, search for new physics, hardware quality control, anomaly detection, unfolding, and many others.

A comprehensive list of current uses for ML in HEP are shown in the link below.

Useful Link: HEPML-LivingReview

Image Credit: Javier Duarte
A simple neural network is a directed acyclic graph that consists of neurons which use a weighted sum over the inputs of the previous layer and an activation function, $K$, to generate an output.

$$f(x) = K \left( \sum_i w_i g_i(x) \right)$$

The weights are “learned” (or rather optimized) during the training process by performing gradient descent of the loss function by evaluating the loss for predicted labels vs true labels. Common loss functions include cross entropy for classification or MSE for regression.

Source: Mathematics of NNs
Credit: LBNL ML July 2023 Workshop
Arbitrarily deep and wide neural networks allow us to approximate any function. If deep NN can approximate any function, why have we seen performance improvement by switching to GNN? The answer lies in better data representation.

Source: Universal Approximation Theorem Wikipedia
Typical events in the ATLAS detector are very “busy”. What variables/objects would be most useful for the NN?

Data is collected as readout from sensors from two main components: the tracker and the calorimeter.

How do we reconstruct the readout from these sensors as tracks and jets? How do we best represent this data for a NN?
Current reconstruction techniques at ATLAS use tracker hits and Calorimeter deposits to reconstruct PFlow Jets.

The anti-kt algorithm, shown left, is used to cluster jets.

Credit: ATLAS Hadronic Calibration Workshop
Anti-k, Source: arxiv 0802.1189
Once the event has been reconstructed, we have access to all the information shown in the table.

Calorimeter information includes pT, eta, and phi of the jet.

Tracker information includes kinematics, secondary vertex information, and tracker hits.
For the purpose of flavor tagging, each jet is represented by an unordered collection of tracks. The number of tracks per jet can vary and the order of the tracks should not affect the output of the classifier.

How do we learn from set data? Deep sets.

**Theorem 2** A function \( f(X) \) operating on a set \( X \) having elements from a countable universe, is a valid set function, i.e., invariant to the permutation of instances in \( X \), iff it can be decomposed in the form \( \rho \left( \sum_{x \in X} \phi(x) \right) \), for suitable transformations \( \phi \) and \( \rho \).

The foundation of Message Passing GNNs

Note: Deep sets representation is similar to point cloud representation.

Source: arxiv 1703.06114
Message Passing Graph Neural Networks

1. Prepare Message
   - Use NN that learns the optimal message

2. Aggregate Messages
   - Use a permutation invariant operation on the messages such as sum, mean, or max.

3. Update Node Embeddings
   - Use NN that learns the optimal node embedding update.

Source: PyTorch Documentation
Consider a jet composed of a set of tracks: $J = \{t_0, t_1, t_2, ..., t_n\}$

**Neighboring Tracks**

$t_1 = \{p_T, \eta, \phi\}$

$t_2 = \{p_T, \eta, \phi\}$

$\vdots$

$t_n = \{p_T, \eta, \phi\}$

1. Prepare Message
2. Aggregate Messages
3. Update Node Embeddings

**Self Track + Aggregated Message**

$t_0 = \{p_T, \eta, \phi\}$

$t_0 = \{f_1, f_2, f_3, ...\}$
The GNN in ATLAS uses message passing neural network to update the node representation of each track based on its neighbors. Then jet classification can be done using a global graph network.

Source: ATL-PHYS-PUB-2022-027
Previous model used in ATLAS, DL1r, uses Deep Learning approach. Notice there is a large improvement in c-jet and light-jet rejection by using a deep set data representation.

Source: ATL-PHYS-PUB-2022-027
GN2 uses the same architecture as GN1 but includes a multi-head attention layer after the updated GNN node embeddings. This has been shown to increase performance.
GN2X is a variant of GN2 and is used to classify boosted jets as $H \rightarrow bb$, $H \rightarrow cc$, top, or multijet (QCD).

GN2X along with GN2 will be crucial for analyses with hadronic channels in LHC Run 3.
ParticleNet and GN1 are both examples of message passing neural networks. However, ParticleNet uses a clever trick by performing dynamic edge convolution using k-nearest neighbors which allows ParticleNet to learn connections between subjects.

Source: ParticleNet Documentation
As ParticleNet performs dynamic edge convolutions, the distribution of the most relevant edges is affected differently for Top and QCD jets.

Source: arxiv 2211.09912
As shown in the linked paper, the use of dynamic edge convolutions allows ParticleNet to learn connections between subjets which is hinting at the physical substructure of the jet. QCD jets do not exhibit this behavior with ParticleNet.

Source: arxiv 2211.09912
Semi-leptonic $t\bar{t}$ decay has a unique signature in the detector, and can be used for tagger calibration. The method is referred to as tag and probe method where the lepton, $b$-tagged jet, and MET are matched to a large-$R$ jet.

Source: ATL-PHYS-PUB-2021-035
Extracting scale factors simply involves counting the number of events in the tagged (SR) and untagged (CR) and performing a likelihood fit.

\[
N_{\text{tag}}^{\text{data}} = \mu \times \frac{\epsilon_{\text{data}}}{\epsilon_{\text{MC}}} \times N_{\text{tag}}^{\ell\ell} + N_{\text{tag}}^{\text{other}}
\]

\[
N_{\text{untag}}^{\text{data}} = \mu \times \frac{1 - \epsilon_{\text{data}}}{1 - \epsilon_{\text{MC}}} \times N_{\text{untag}}^{\ell\ell} + N_{\text{untag}}^{\text{other}}
\]

Figure 9: Data and MC simulated distributions of $p_T$ (a) and mass of the probe jet (b) for the pre-tag selection. The ratio panel shows the data-to-MC ratio. The uncertainty band includes MC statistical and systematic uncertainties.

Source: ATL-PHYS-PUB-2021-035
Calibration

Post-fit distributions in SR, right, and CR, left.

Figure 10: Post-fit distribution of the large-$R$ jet mass in SR_{had} used for calibration of the top jet mis-tag efficiency with $X \rightarrow b\bar{b}$ tagger at 60% efficiency WP for $300 < p_{T}^{med} < 400$ GeV (a); $400 < p_{T}^{med} < 500$ GeV (b); $500 < p_{T}^{med} < 600$ GeV (c) and $600 < p_{T}^{med} < 1000$ GeV (d). The uncertainty band represents the systematic uncertainty.

Figure 11: Post-fit distribution of the large-$R$ jet mass in SR_{had} used for calibration of the top jet mis-tag efficiency with $X \rightarrow b\bar{b}$ tagger at 60% efficiency WP for $300 < p_{T}^{med} < 400$ GeV (a); $400 < p_{T}^{med} < 500$ GeV (b); $500 < p_{T}^{med} < 600$ GeV (c) and $600 < p_{T}^{med} < 1000$ GeV (d). The uncertainty band represents the systematic uncertainty.

Source: ATL-PHYS-PUB-2021-035
Breakdown of uncertainties on the scale factor are obtained with a non-profile likelihood fit.
Notice how this entire talk I have been referring to reconstruction level information. Reconstructed information has unwanted detector effects... (•´_´• )

What if we want to unfold the detector level measurement to get truth level information?

Credit: Returning CP-Observables to The Frames They Belong
Image Source: LBNL ML Workshop 2023 Unfolding Slides
Conclusion

Thank you for listening! （・○・）

Questions?

- Motivation for b-tagging
- Machine Learning in HEP
- Data Collection and Jet Reconstruction
- Deeps Sets and Message Passing Neural Networks
- Current GNN Implementations in ATLAS and CMS
- Calibration of Taggers
- Unfolding