Identification of Jets Containing $b$-Hadrons with Recurrent Neural Networks at the ATLAS Experiment

ATL-PHYS-PUB-2017-003

Dan Guest
For the ATLAS Collaboration

UC Irvine

March 21, 2017
Background: $b$-tagging

- $b$-hadrons decay through cascade
- $\beta\gamma c\tau \approx 6.4\text{ mm for } B \text{ with } p_T = 70\text{ GeV}$
- But many decay distances are $O(\text{detector resolution})$
Reconstructing Secondary Vertices

The ATLAS approaches

- Many discriminants come from vertices, combine them with ML
The problem with SV tagging

- Sometimes we don’t find a vertex
- Requires cutting on track-vertex compatibility
  - This is also a good thing
- Tuned “by hand”
- Experiment-specific

- There is no FASTJET for vertex reconstruction
Impact parameter (IP) tagging

- Take all tracks in a jet
- Apply some selection
- Extrapolate to perigee
- Per-track discriminants:
  - \( S_{d_0} \equiv d_0/\sigma_{d_0} \)
  - \( S_{z_0} \equiv z_0/\sigma_{z_0} \)
  - track “quality”

- Compute per-track likelihood \( L_f(\text{track}) \) with \( f \in \{b, c, \text{light}\} \)
- Per-jet likelihood \( p_f = \prod_{\text{trk}} L_f(\text{track variables}) \)
- IP based tagging is the problem we solve with RNNs
  - More on this later
Putting it all together

Low-Level

- **IP**: track-based variables
- **Likelihood**: gives $p_b$, $p_c$, $p_{\text{light}}$
- **SV**: gives vertex variables
- **JetFitter**: similar to SVx

High-level

- **MV2**: combine with BDT

- It’s easy to focus on the high-level tagger (MV2), but upstream is important too
IP3D: ATLAS’s IP Tagger

- Need to define $L_f(\text{track})$
  - $L_f(S_{d_0}, S_{z_0}, \text{category})$
  - $S_{d_0}$ shown right
- Use histograms from simulation
- 3D binning scheme:
  - 35 bins in $S_{d_0}$
  - 20 bins in $S_{z_0}$
  - 14 track categories
- Track category represents quality of track

![Graph showing track signed $d_0$ significance]

ATLAS Simulation Preliminary
\( \sqrt{s} = 13 \text{ TeV}, \bar{t}t \)}
Improving Upstream Taggers: What IP3D misses

- Relations among tracks:
  - relation to neighbor bins
  - relation to neighbor tracks
- These are important (see right)
- New (SV inspired) track variables:
  - $p_T^{\text{frac}} \equiv \frac{p_T^{\text{track}}}{p_T^{\text{jet}}}$
  - $\Delta R(\text{track}, \text{jet})$

Curse of Dimensionality

- Already 29,400 bins
- New variable $\rightarrow \sim 10 \times$ bins (and events to “train”)
Recurrent Neural Networks (RNNs)

- RNNs can process an arbitrarily length sequence
- Output is a fixed dimensional vector for each jet
RNN b-tagging

Unrolled RNN

Fully Connected + SoftMax

Jet

ordered by |Sdo|

Track 1 Track 2 Track 3 Track 4 ... Track N

ΔR

$S_{do}$

$S_{zo}$

$p_{T}^{frac}$

category

Embed

2D unit vector

Jet
ROC Curves for a Multi-Background Discriminant

- Eventually: feed all four outputs to a high-level discriminant
- Conventional HEP discriminants are binary
  - Train against a mix of backgrounds (i.e. MV2 is 7% c-jets)
- We use 4 outputs:
  - \( p_b \): bottom jet
  - \( p_c \): charm jet
  - \( p_{\text{light}} \): “light” jet (u, d, s, g)
  - \( p_{\tau} \): \( \tau \) jet
- Combine everything for the sake of plots

\[
D_{\text{RNN}} = \ln \frac{p_b}{f_c p_c + f_{\tau} p_{\tau} + (1 - f_c - f_{\tau}) p_{\text{light}}}
\] (1)

- The \( f \) weighting parameters can be adjusted post-training
- For this talk: \( f_c = 0.07, f_{\tau} = 0 \)
**RNN Performance (compared to IP3D)**

- Lowest line is IP3D
- Next up: RNN with IP3D inputs
- Each new variable adds discrimination
- At 70% working point:
  - RNN with IP3D inputs improves light rejection by 1.7
  - With $\Delta R(\text{track}, \text{jet})$ and $p_T^{\text{frac}}$, improves light rejection by 2.5
RNN Performance (compared to high-level tagger)

MV2 using IP3D still rejects more background for $\varepsilon_b < 0.9$

But this uses JetFitter and SV $\rightarrow$ much more information

RNN as input for MV2 is outside the scope of this talk
  But we can imagine replacing IP3D with the RNN
Cut on the discriminant such that $\varepsilon_b = 0.7$ in each $p_T$ bin.

- Same trend as previous slide: rejection for IP3D < RNN < MV2

- RNN tagger is no more $p_T$ dependent than other taggers.
RNN output correlation with input: $S_{d0}$ and $S_{z0}$

- $D_{\text{RNN}}$ output is highly correlated with jet $S_{d0}$ for “early” tracks in $|S_{d0}|$ ordering
  - Interesting, but maybe not surprising: $b$ hadrons have $\sim 5$ tracks
- Effect is less pronounced for $S_{z0}$
RNN output correlation with input: $\Delta R$ and $p_T^{\text{frac}}$

Much less correlation between $D_{\text{RNN}}$ and $\Delta R(\text{track, jet})$ or $p_T^{\text{frac}}$

But these are useful discriminants nonetheless
Notes on Software
Since we always talk about software in IML

- We train with **Keras**
  - Use 3.2 million jets from simulated $t\bar{t}$
  - Use **Theano** backend
  - Training time: with a CPU, a few days on a (busy) cluster
  - We only train on first 15 tracks (0.5% of jets 15+ tracks)

- Within our reconstruction, we evaluate with **LWTNN**
  - Used in ATLAS for top and W tagging (see tomorrow)
  - Also used by CMS for DeepFlavour (see next talks)
Conclusions

- RNNs are a promising tool for flavor tagging
  - Use relatively low-level variables
  - Can augment vertex-based approaches
- We’ve successfully integrated an RNN-based tagger into the ATLAS reconstruction framework
- Many interesting questions:
  - What other low-level variables could we include?
  - How do we mitigate modeling issues?
  - Can we “understand” (visualize) what we’ve learned?
  - How does this complement a high-level tagger (e.g. MV2, DeepFlavour)?
- Thanks for listening, ideas are welcome!
BACKUP
Thanks

- Michela Paganini and Jonathan Shlomi for the graphics
- Zihao Jiang, Michael Kagan, Michela, and the rest of the RNN team for training lots of networks
- The ATLAS flavor tagging group for a good problem
- ATLAS for all the simulation
IP3D Categories

<table>
<thead>
<tr>
<th>#</th>
<th>Category</th>
<th>Fractional contribution [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>b-jets</td>
</tr>
<tr>
<td>0</td>
<td>No hits in first two layers; expected hit in IBL and b-layer</td>
<td>1.9</td>
</tr>
<tr>
<td>1</td>
<td>No hits in first two layers; expected hit in IBL and no expected hit in b-layer</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>No hits in first two layers; no expected hit in IBL and expected hit in b-layer</td>
<td>0.04</td>
</tr>
<tr>
<td>3</td>
<td>No hits in first two layers; no expected hit in IBL and b-layer</td>
<td>0.03</td>
</tr>
<tr>
<td>4</td>
<td>No hit in IBL; expected hit in IBL</td>
<td>2.4</td>
</tr>
<tr>
<td>5</td>
<td>No hit in IBL; no expected hit in IBL</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>No hit in b-layer; expected hit in b-layer</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>No hit in b-layer; no expected hit in b-layer</td>
<td>2.4</td>
</tr>
<tr>
<td>8</td>
<td>Shared hit in both IBL and b-layer</td>
<td>0.01</td>
</tr>
<tr>
<td>9</td>
<td>At least one <em>shared</em> pixel hits</td>
<td>2.0</td>
</tr>
<tr>
<td>10</td>
<td>Two or more <em>shared</em> SCT hits</td>
<td>3.2</td>
</tr>
<tr>
<td>11</td>
<td>Split hits in both IBL and b-layer</td>
<td>1.0</td>
</tr>
<tr>
<td>12</td>
<td>Split pixel hit</td>
<td>1.8</td>
</tr>
<tr>
<td>13</td>
<td>Good</td>
<td>83.6</td>
</tr>
</tbody>
</table>

- Fractions are based on simulated $t\bar{t}$
Track Selection

- Jet Algorithm: Anti-$k_t$, $R = 0.4$
- Track $p_T > 1$ GeV
- $|d_0| < 1$ mm, $|z_0 \sin \theta| < 1.5$ mm
- $n_{\text{Si hits}} \geq 7$, $n_{\text{Si holes}} \leq 2$, $n_{\text{pixel holes}} \leq 1$