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

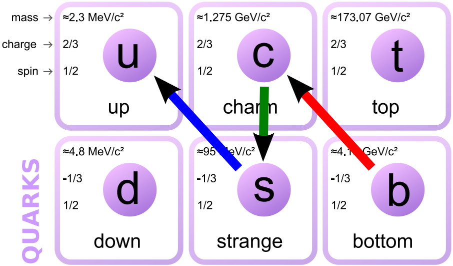
ATL-PHYS-PUB-2017-003

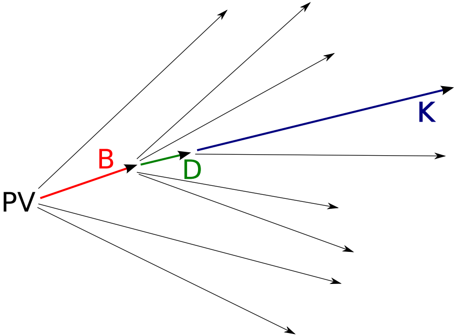
Dan Guest
For the ATLAS Collaboration

UC Irvine

March 21, 2017

Background: *b*-tagging

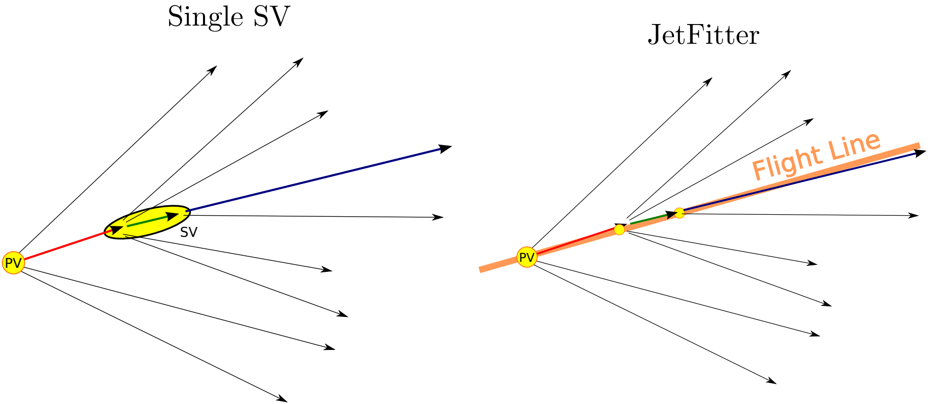




- b-hadrons decay through cascade
- $\beta\gamma c\tau \approx 6.4\,\mathrm{mm}$ for B with $p_{\mathrm{T}} = 70\,\mathrm{GeV}$
- But many decay distances are O(detector resolution)

Reconstructing Secondary Vertices

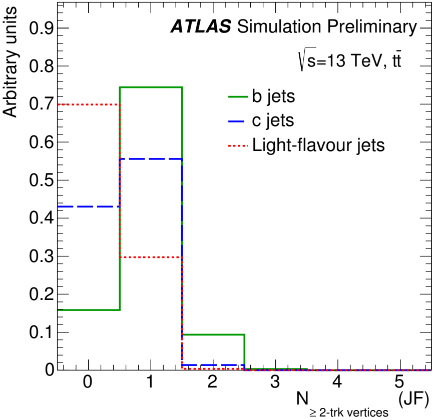
The ATLAS approaches



- Many discriminants come from vertices, combine them with ML

The problem with SV tagging

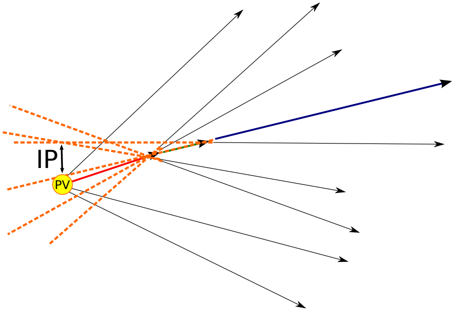
- Sometimes we don't find a vertex
- Requires cutting on track-vertex compatibility
  - This is also a good thing
- Tuned "by hand"
- Experiment-specific



- 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 $\mathcal{L}_f(\text{track})$ with $f \in \{b, c, \text{light}\}$
- Per-jet likelihood $p_f = \prod_{\text{trk}} \mathcal{L}_f(\text{track variables})$
- **IP based tagging is the problem we solve with RNNs**
  - More on this later

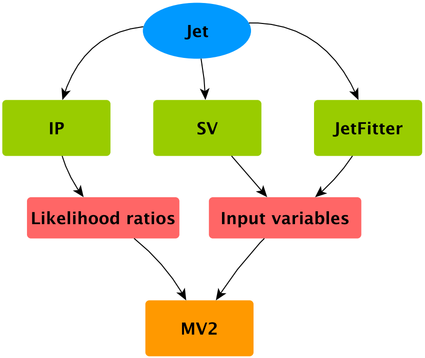
Putting it all together

Low-Level

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

High-level

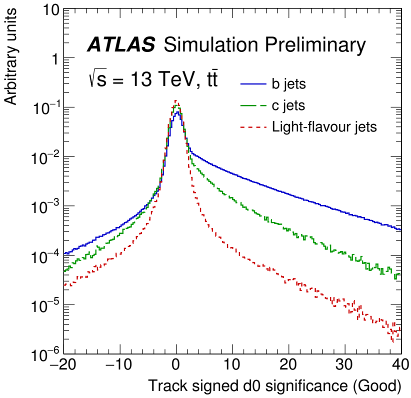
- MV2: combine with BDT



- It's easy to focus on the high-level tagger (MV2), but upstream is important too

IP3D: ATLAS's IP Tagger

- Need to define $\mathcal{L}_f(\text{track})$
  - $\mathcal{L}_f(S_{d_0}, S_{z_0}, \text{category})$
  - 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

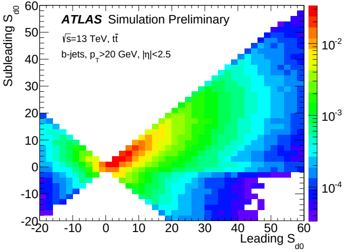


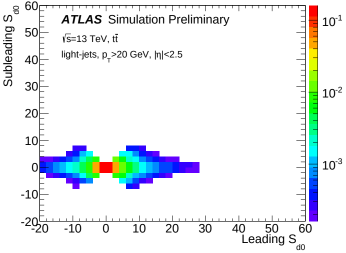
Improving Upstream Taggers: What IP3D misses

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

Curse of Dimensionality

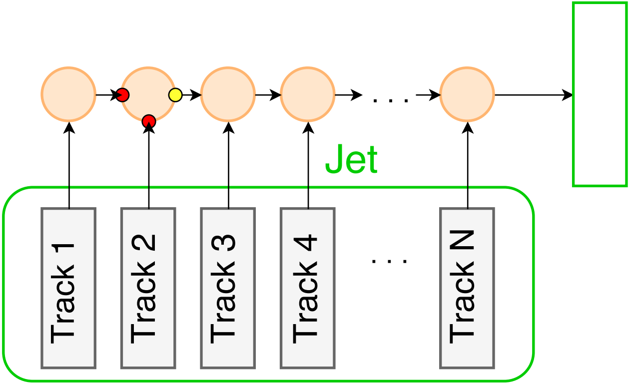
- Already 29,400 bins
- New variable $\rightarrow \sim 10\times$ bins (and events to "train")

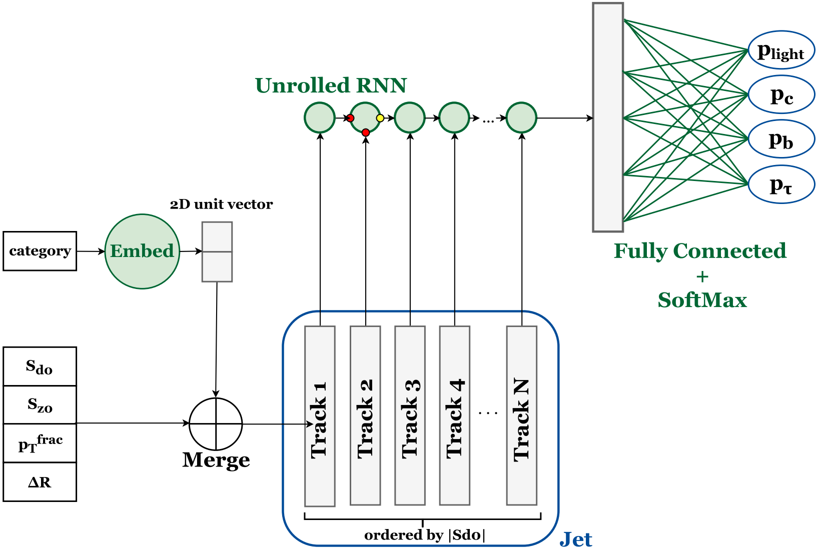




Recurrent Neural Networks (RNNs)

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




Unrolled RNN
2D unit vector
category
Embed
S_{do}
S_{zo}
p_T^{frac}
ΔR
Merge
Track 1
Track 2
Track 3
Track 4
Track N
ordered by |Sdo|
Jet
p_{light}
p_c
p_b
p_τ
Fully Connected
+
SoftMax

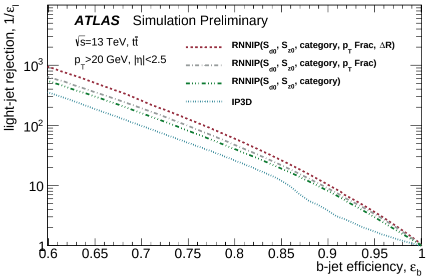
ROC Curves for a Multi-Background Discriminant

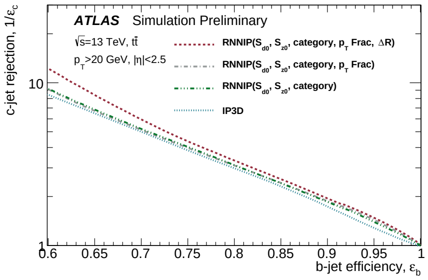
- Eventually: feed all four outputs to a high-level discriminant
- Conventional HEP discriminants are binary
  - Train against a mix of backgrounds (i.e. MV2 is 7% c-jets)
- We use 4 outputs:
  - p_b: bottom jet
  - p_c: charm jet
  - p_{light}: “light” jet (u, d, s, g)
  - p_τ: τ jet
- Combine everything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

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

- The f weighting parameters can be adjusted post-training
- For this talk: $f_c = 0.07$, $f_\tau = 0$

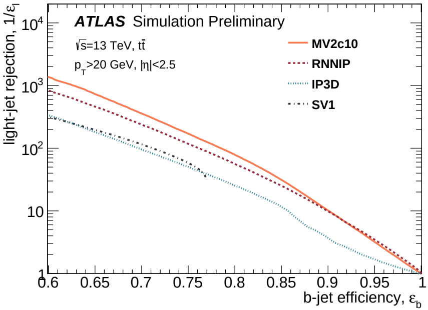
RNN Performance (compared to IP3D)

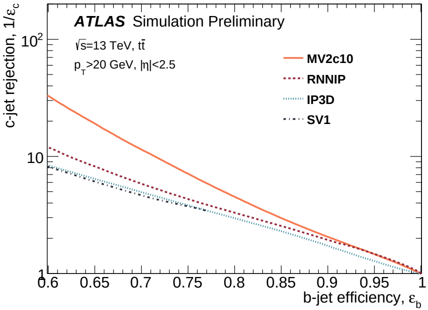




- Lowest line is IP3D
- Next up: RNN with IP3D inputs
- Each new variable adds discrimination
- At 70% working point:
  - RNN with IP3D inputs improves light rejection by 1.7
  - With $\Delta R(\text{track}, \text{jet})$ and $p_{\text{T}}^{\text{frac}}$, improves light rejection by 2.5

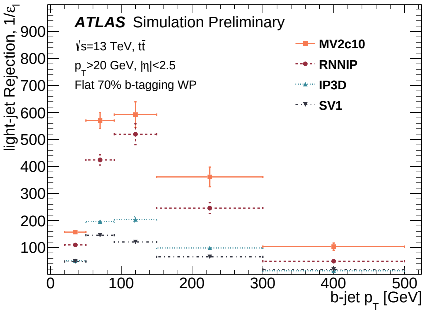
RNN Performance (compared to high-level tagger)

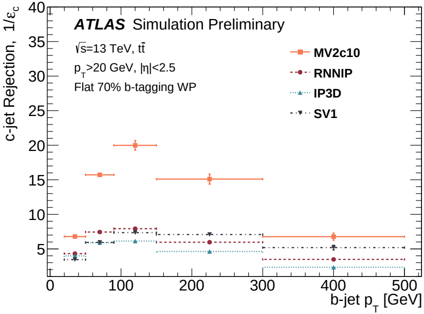




- MV2 using IP3D still rejects more background for $\varepsilon_b < 0.9$
- But this uses JetFitter and SV → much more information
- RNN as input for MV2 is outside the scope of this talk
  - But we can imagine replacing IP3D with the RNN

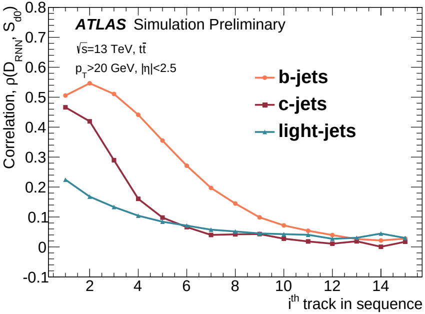
RNN Performance by p_{T}

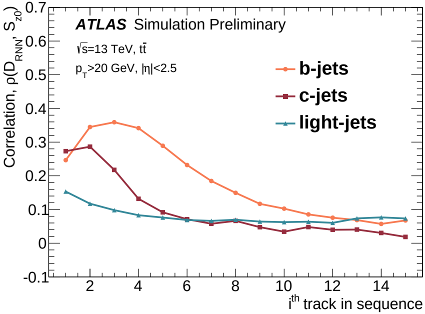




- Cut on the discriminant such that $\varepsilon_b = 0.7$ in each p_{T} bin
- Same trend as previous slide: rejection for IP3D $<$ RNN $<$ MV2
- RNN tagger is no more p_{T} dependent than other taggers

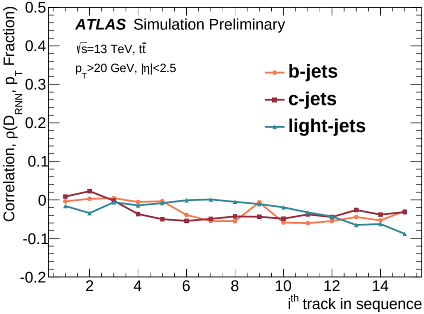
RNN output correlation with input: S_{d_0} and S_{z_0}

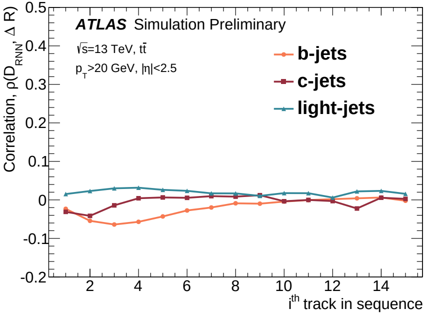




- D_{RNN} output is highly correlated with jet S_{d_0} for "early" tracks in $|S_{d_0}|$ ordering
  - Interesting, but maybe not surprising: b hadrons have ~ 5 tracks
- Effect is less pronounced for S_{z_0}

RNN output correlation with input: ΔR and $p_{\mathrm{T}}^{\mathrm{frac}}$





- Much less correlation between D_{RNN} and $\Delta R(\mathrm{track, jet})$ or $p_{\mathrm{T}}^{\mathrm{frac}}$
- 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

#	Category	Fractional contribution [%] b-jets	c-jets	light-jets
0	No hits in first two layers; expected hit in IBL and b-layer	1.9	2.0	1.9
1	No hits in first two layers; expected hit in IBL and no expected hit in b-layer	0.1	0.1	0.1
2	No hits in first two layers; no expected hit in IBL and expected hit in b-layer	0.04	0.04	0.04
3	No hits in first two layers; no expected hit in IBL and b-layer	0.03	0.03	0.03
4	No hit in IBL; expected hit in IBL	2.4	2.3	2.1
5	No hit in IBL; no expected hit in IBL	1.0	1.0	0.9
6	No hit in b-layer; expected hit in b-layer	0.5	0.5	0.5
7	No hit in b-layer; no expected hit in b-layer	2.4	2.4	2.2
8	*Shared* hit in both IBL and b-layer	0.01	0.01	0.03
9	At least one *shared* pixel hits	2.0	1.7	1.5
10	Two or more *shared* SCT hits	3.2	3.0	2.7
11	*Split* hits in both IBL and b-layer	1.0	0.87	0.6
12	*Split* pixel hit	1.8	1.4	0.9
13	*Good*	83.6	84.8	86.4

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

Track Selection

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