Boosted Object Tagging With Deep Networks

Dan Guest[1] Kevin Bauer[1] Shih-Chieh Hsu[3] Clara Eng[2]
Peter Sadowski[1] Julian Collado[1] Gergor Urban[1]
Pierre Baldi[1] Daniel Whiteson[1]

[1]UC Irvine

[2]UC Berkeley

[3]University of Washington

July 21, 2016

Background: Boosted $H \rightarrow bb$

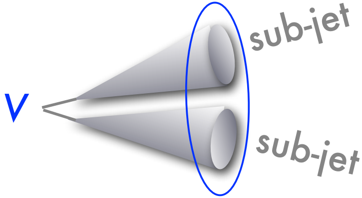
…for the sake of example

Calorimeter

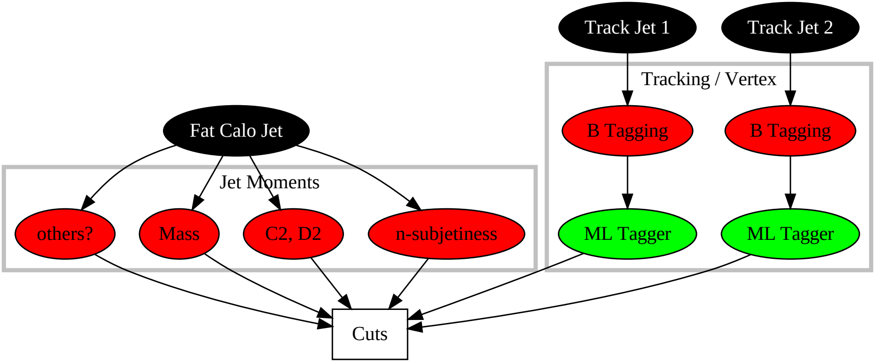
- EW scale particle V recoils against something
- Two-pronged energy deposit in the calorimeter → "fat jet"
- Use jet moments as discriminants
  - τ_{21}, C_2, D_2 etc…

Tracking

- We expect two b jets
- Find subjets
- Run flavor tagging

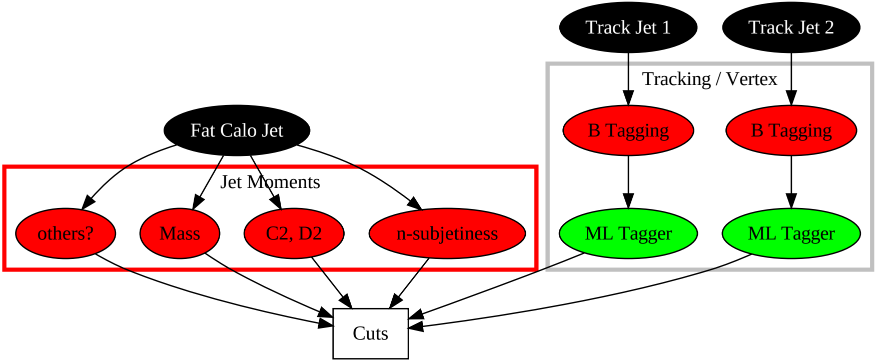


Boosted $H \to bb$: Algorithmic Overview



- Many levels of filtering / compression
  - *Are we missing something?*
- Lots of BDTs/NNs already
  - *Can we simplify this?*

Part 1: The Calorimeter

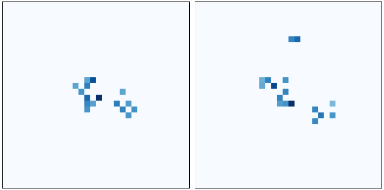


- The idea: replace jet moments with “jet images”
  - Original paper: arXiv:1407.5675
  - NN addition: arXiv:1511.05190
  - **This talk:** arXiv:1603.09349

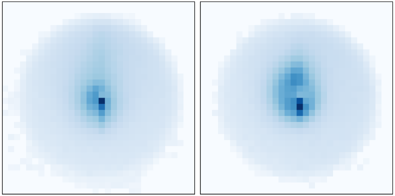
What is this “Image”?

- The idea: a jet is just an image in the calorimeter
- Discretized as $32 \times 32 = 1024$ pixel image
- We assume rotational symmetry (but don’t have to)
- We can use standard image recognition
    - Deep neural networks have shown the best results
- See previous talk!

One Jet



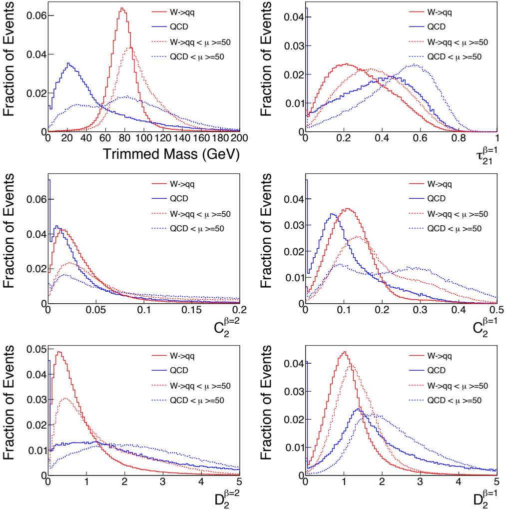
100,000 Jets



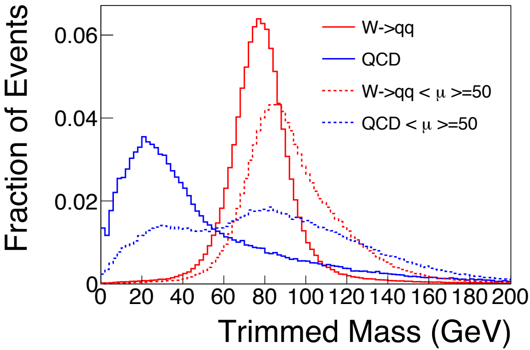
What to compare images to? Jet Moments

- Which one to use?
- One, many?
- Combinations?

- **For the sake of argument feed them *all* to a BDT**



Adding Realism



- We can get (halfway) to reality with Delphes
- Detector response smears hadron momentum
- Pileup interactions make a *huge* difference
  - . . . even with pileup suppression applied

Our Setup

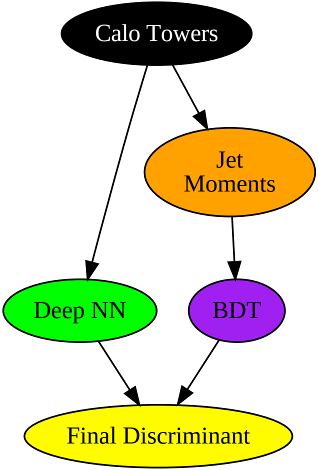
- Used Delphes, $\sqrt{s} = 14\,\text{TeV}$, $\langle\mu\rangle = 50$
- Signal: $pp \to WW \to qqqq$
- Background: $pp \to qq, q, gg$
- Anti-k_{T} jets $\Delta R = 1.2$
- $300\,\text{GeV} < p_{\text{T}} < 400\,\text{GeV}$
- Apply pileup suppression (trimming)

The Question

If we compare

- A BDT on engineered variables, to
- A deep network on the jet image

which one is a better classifier?



Training

- 10M Jets (500k for validation)
- Used Spearmint Bayesian optimization (arXiv:1206.2944)

Deep NN

	Range		Optimum	
Hyperparameter	Min	Max	No pileup	Pileup
Hidden units per layer	100	500	425	500
Fully-connected layers	1	5	4	5
Locally-connected layers	0	5	4	3

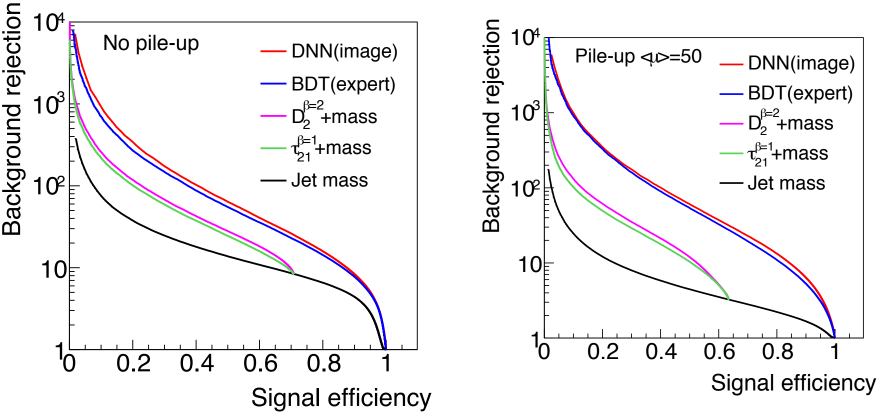
BDT

	Range		Optimum	
Hyperparameter	Min	Max	No pileup	Pileup
Tree depth	15	75	49	49
Learning rate	0.01	1.00	0.07	0.07
Minimum split percent	0.0001	0.1000	0.0021	0.0021

- 4×4 locally connected $\rightarrow$ fully connected
- Training: Theano + Keras
- ADAM optimizer
- **750k free parameters**

- Training: Scikit-Learn
- **750k free parameters**

Results



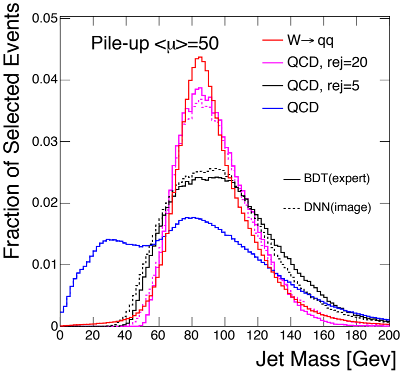
- Image NN outperforms moment BDT (slightly)
- Both seem to use more information than single moments
- Are they using the same information?

Moment Sculpting in Background

- Initially, background doesn't look look like signal
- Cut on discriminant background → signal
- ...unless the NN learns different information

- $\boxed{\text{Rej} = 5}$: bgjets starts to peak
- Rej = 20 looks *a lot* like signal
- BDTs similar to NN

- **NNs *learn* mass, τ_{21}, etc**

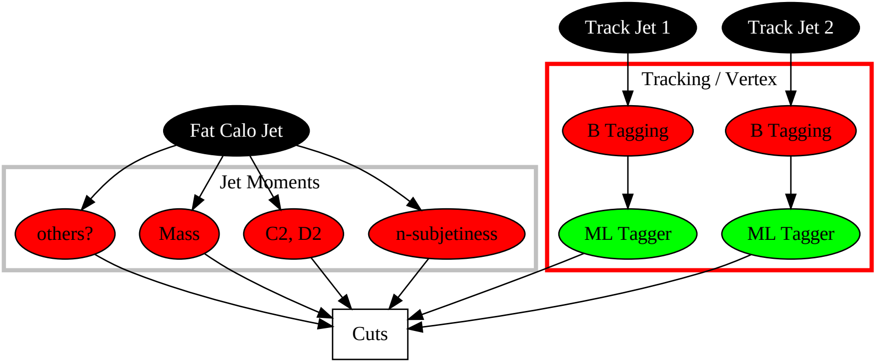


Calorimeter Conclusions

- Deep Network on image works slightly better than moment BDT
- But moments do *very well*
- *Most* of the information seems to be encoded in 6 variables
    - Good job QCD theorists!

What About Tracking?

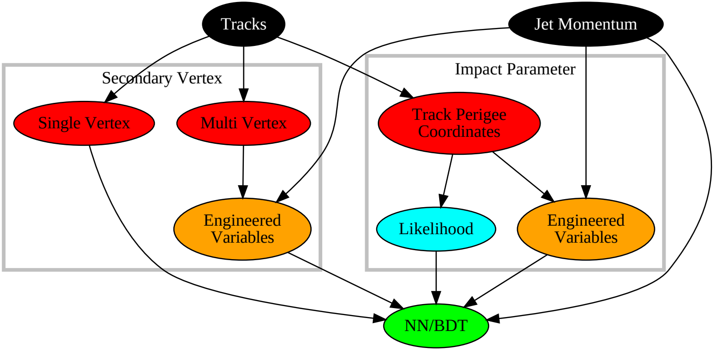
Flavor tagging based discriminants



- We can replace jet moments with raw inputs
- Can we do something similar for tracks?
- ML is more standard in flavor-tagging

Flavor Tagging is *Complicated*

Based on (simplified) ATLAS framework

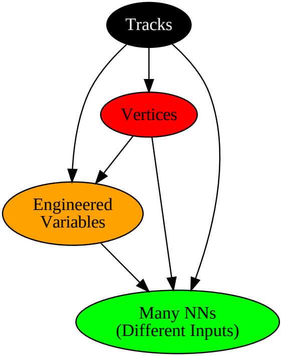


- Basically two algorithms + dimensionality reduction
- Each new feature $\rightarrow$ better discrimination
  - We still haven't found the right basis
- Hard to optimize, *can we simplify this?*

Our (even more) Simplified Setup

Details will appear on arXiv very soon

- $pp \to (qq, bb) \to$ Delphes
- Anti-k_T jets $\Delta R = 0.4$
- Start with tracks
  - Usually < 15
- Fit **vertices**
  - $\sim$ two or three
- Build **HL Features**
  - **Always 14**

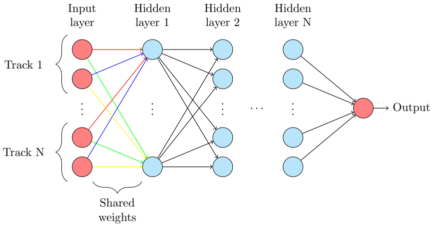


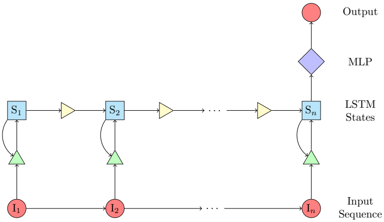
- HL is strictly derived from Vertex and Tracks
- Showing: $20\,\mathrm{GeV} < p_\mathrm{T} \lesssim 100\,\mathrm{GeV}$
  - More boosted studies ongoing

Challenges

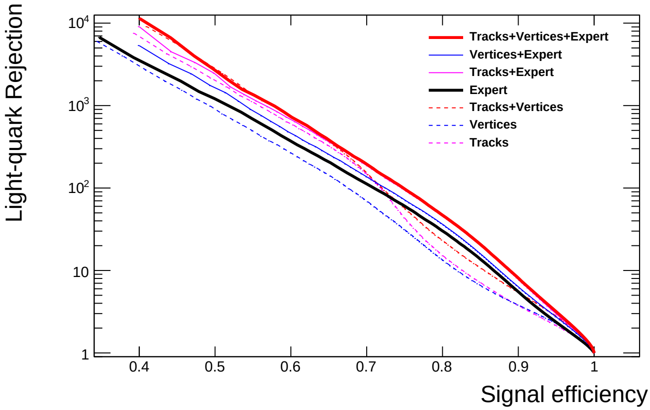
Why hasn't someone already done this?

- No FastJet for vertices
  - use Rave (thanks CMS)
- High (variable) dimensional input space
  - 15 tracks $\times$ 5 parameters $=$ 75 inputs (but usually more)
  - That's a lot for HEP ML
  - But not bad for modern image / text processing





Results



- Low level information helps
- ...but tracks + vertices + expert features still beats all

Conclusions, Part 2

- Compared to the calo problem:
  - The tracking problem is more difficult for neural nets:
    - NN with track inputs *doesn't* beat expert features ...
    - ...even though the expert features are derived from tracks
  - The expert features gain more from the tracks
- Something like FastVertex would be useful
  - Rave is a good start (bit hard to compile...)
  - Recent Delphes branch includes crude vertex algorithms

Conclusions and Ongoing Work

Summary

- Calo image as good as jet moments
- Raw tracks help the more traditional flavor tagging approach

Next Steps

- Many questions applying flavor tagging to $H \rightarrow bb$
  - How should subjets be selected? (if at all)
  - How much do we gain by including track + calo in one NN?

BACKUP

Flavor Tagging Inputs