Jet tagging Kaggle competition

Jet tagging

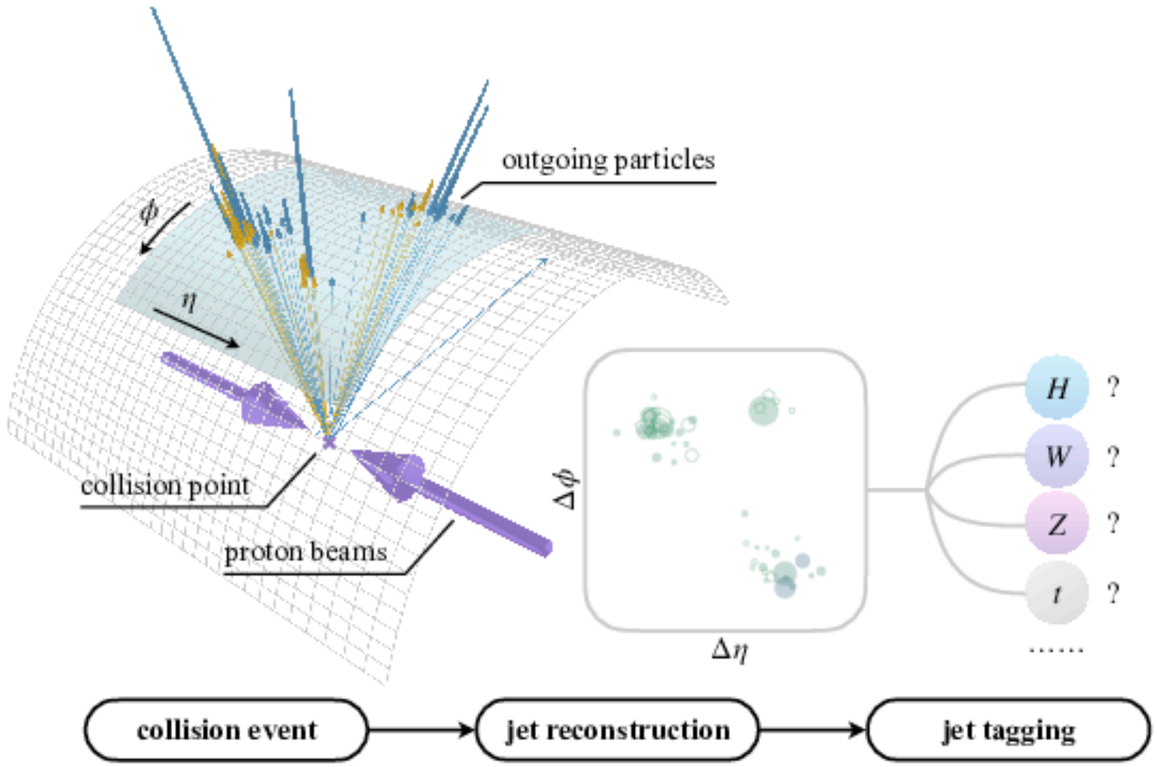
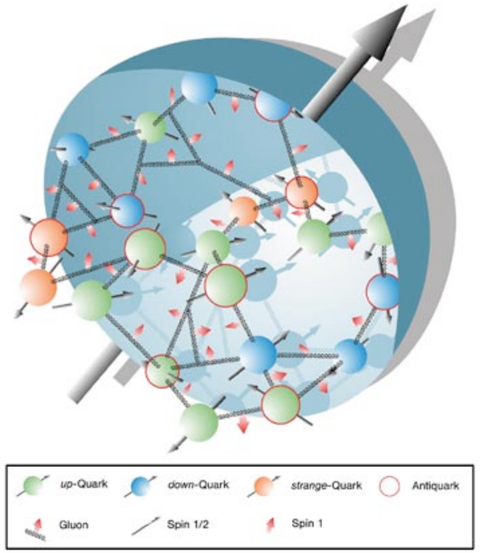


Figure from [1]

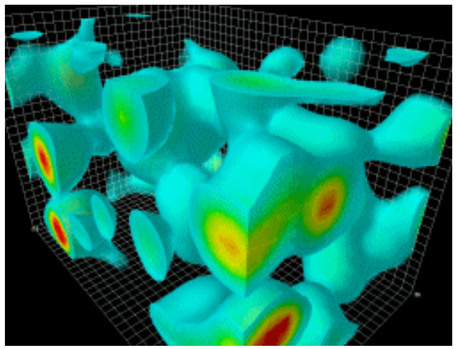
- When protons collide, they produce other particles
- These particles often form jets - bundles of particles moving in similar directions
- We usually want to know how the jet was made to understand if it was made by any interesting physics, such as a Higgs boson
- Most jets are made by normal interactions that we aren't very interested in, so we need to filter out the interesting jets

Quick look at Quantum Chromodynamics (QCD)

- QCD is the theory describing the strong force, which holds nuclei together. It is carried by gluons
- Quarks and gluons can't exist freely on their own, the further away they get, the stronger the force - known as asymptotic freedom
- When you try to pull a quark away, like in an energetic collision, the energy can cause new particles to be created
- Some of these are short lived, some long lived
- This causes messy jets, where particles decay and create new particles



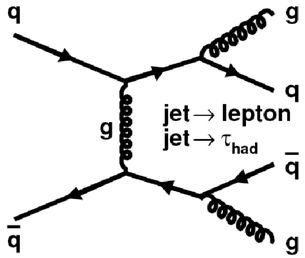
Inner structure of a proton [2]

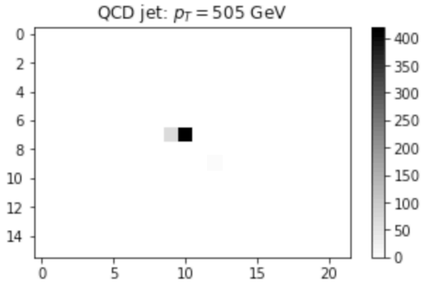


QCD in vacuum [3]

QCD jets and TT jets

QCD jets

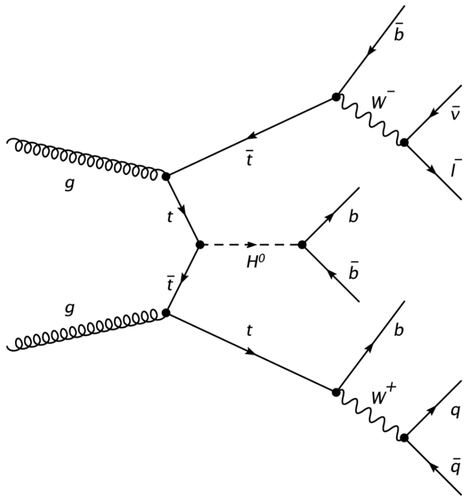


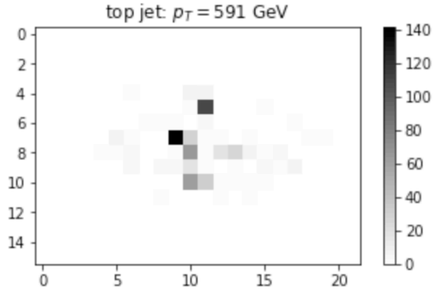


Figures from [4], [5]

- Tend to stay close together
- Balanced energy distribution
- Typically fewer particles, at lower energies

TT jets





Figures from [6], [5]

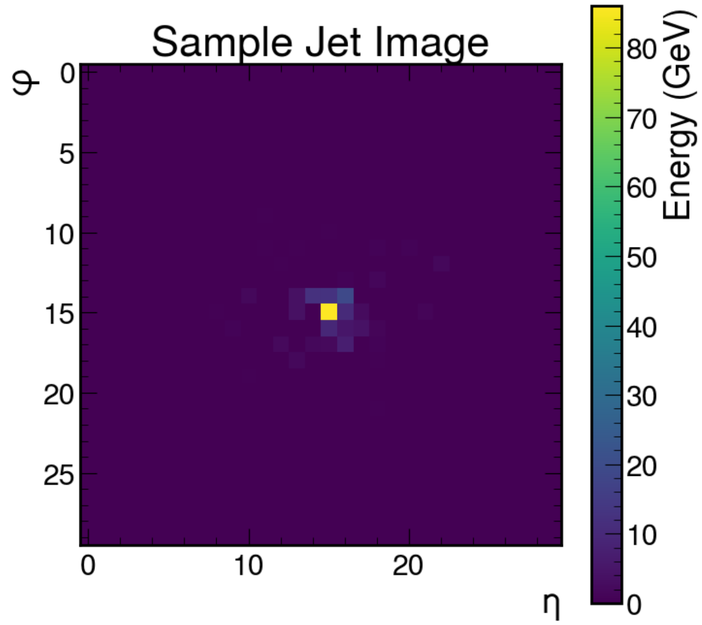
- Often contain clusters further away, so distinct clusters
- More asymmetric energy distribution
- More particles, at a variety of energies

The task

- Distinguish TT jets and QCD jets (binary classification)
- You have both processed numpy arrays and images - use either or both
- There are simple code examples for BDT, DNN, CNN, and GNN, deep sets, and transformers
- The highest AUC score wins - see Kaggle for a description of this
- Do whatever you like to climb the leaderboard, there's also a document with ideas to try if you want inspiration

The data

- Two formats that contain the same information in different ways
- Data frames were made from the images using the anti-kT algorithm
- Different algorithms go between these two
- You are free to use whatever you want, including new features or images you process



	n_clusters	max_cluster_pt	mean_cluster_pt	std_cluster_pt	max_cluster_size	mean_cluster_size	std_cluster_size	total_pt
0	3	2.0	1.652439	0.411616	2	2.0	0.0	4.957317
1	6	2.0	0.931529	0.740799	2	2.0	0.0	5.589176
2	10	2.0	0.721542	0.637740	2	2.0	0.0	7.215421
3	3	2.0	1.189460	0.667869	2	2.0	0.0	3.568379
4	4	2.0	1.174908	0.538286	2	2.0	0.0	4.699632

Quick glossary

ϕ (phi) is the angle around the beamline, like on a compass

η describes how forward or backward a particle flew — how close it stayed to the beamline.

p_t is how much momentum a particle has perpendicular to the beamline.

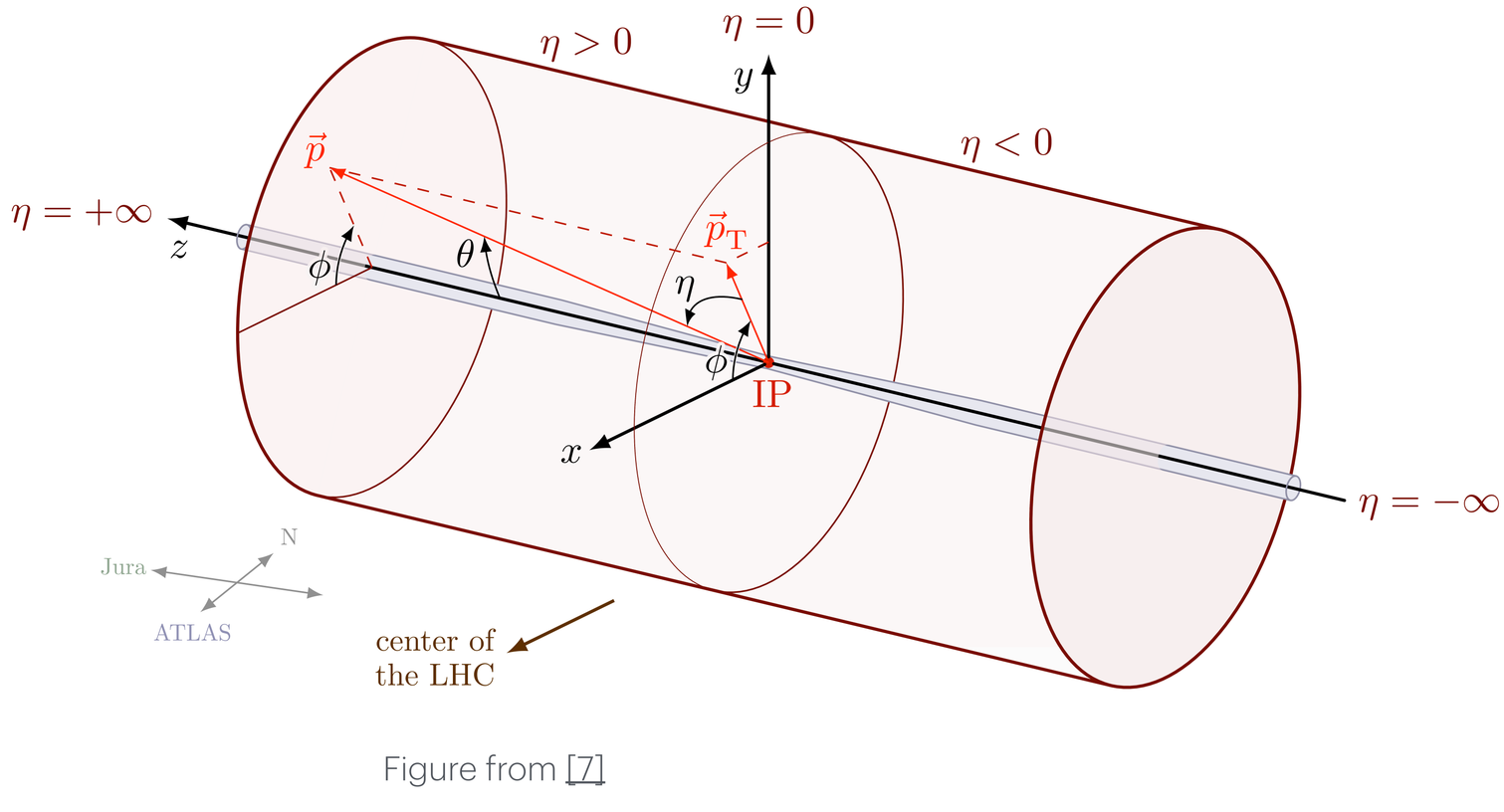


Figure from [7]

Supporting python package

- There are four utility scripts that help you load/process and plot the data
- It's imported in Kaggle and used in the notebooks, you don't have to worry about it if you don't want
- To use them in your notebooks, either copy my notebooks, or see the last slide for how to add utility scripts
- In these scripts you'll find the code for generating the data frames from the images
  Feel free to use it to create new data frames if you'd like!
- If you find yourself writing the same code repeatedly, consider making your own utility scripts!

The submission

- **A valid submission must have the same format as is shown on Kaggle**
- You can evaluate your performance on the training and validation set as you like
- You can see your performance on the public test set by making a submission to the contest
- You have 20 submissions a day
- Beware of the private test set!
- There is a simple benchmark, see if you can beat it :)

Private dataset

- After the competition closes, your performance on the private data will be released
- It is possible that the private test data is larger than the public test data and that the classes occur in different ratios :)
- If you have overfitted, you will drop on the leaderboard
- To make sure this doesn't happen, try to avoid overfitting (cross validation, ensemble learning etc.)

7	▼2	The Zoo	0.92435	125	8d
8	–	MSS	0.92422	75	8d
9	▲5	MN-VGG	0.92422	143	8d
10	▲1	jacobkie	0.92421	23	8d
11	▼2	Santrouble	0.92410	69	8d
12	–	T&S&D&Q	0.92387	101	8d
13	▲4	A Kind of Magic	0.92386	146	8d
14	▲2	ensemble is useless :(	0.92385	23	8d
15	▼5	Gryffindor	0.92379	131	8d

Figure from [8]

How to use Kaggle

- Log in using a google or GitHub account
- Accept the competition rules
- Teams are encouraged
- You can run your code on kaggle on GPU (30hrs/week) or CPU
- Submission can be done by using the toolbar to the left of the competition
- Ask Liv if you have an issues
- Start with the notebooks that are there. Don’t feel any pressure to explore them all, it’s a lot!

Prizes

- We have prizes for 1st place
- Also 2 others that may be given for performance, potentially something else

Adding utility scripts