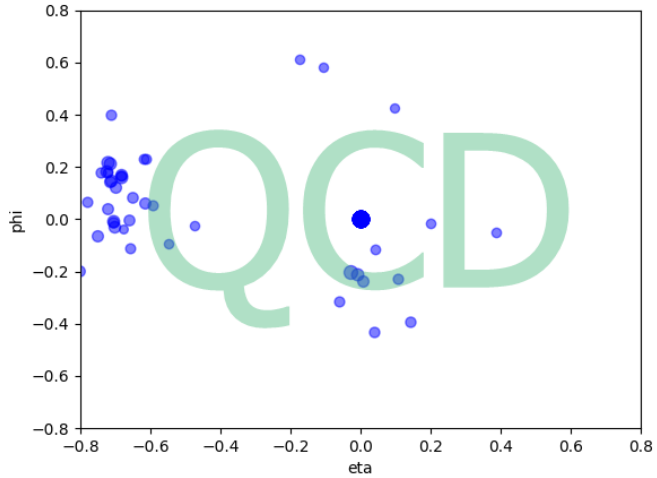
How to find new physics without physicists

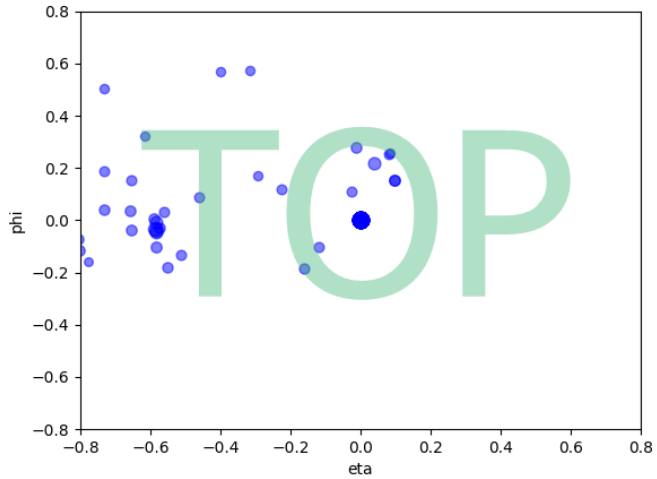
Simon Kluettermann

12. Februar 2021

Top Tagging

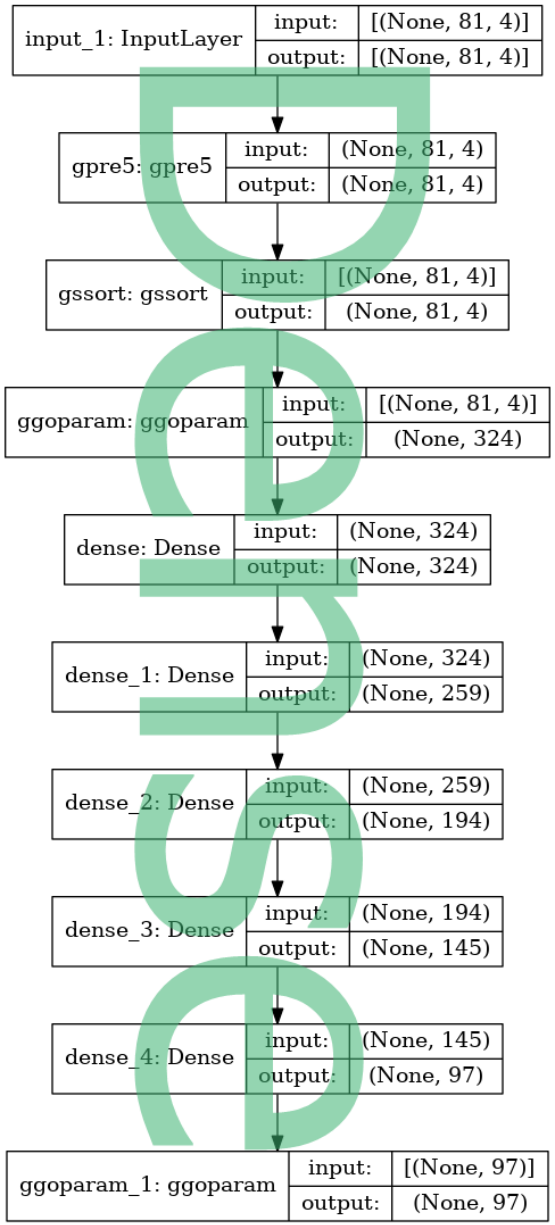
- Looking at LHC jets (particle showers)
- Millions of jets, maybe some of them weird in some way
- But at most a few and in some unknown way
- So use ML to filter them out
- Usually tested by trying to find top jets in a qcd background

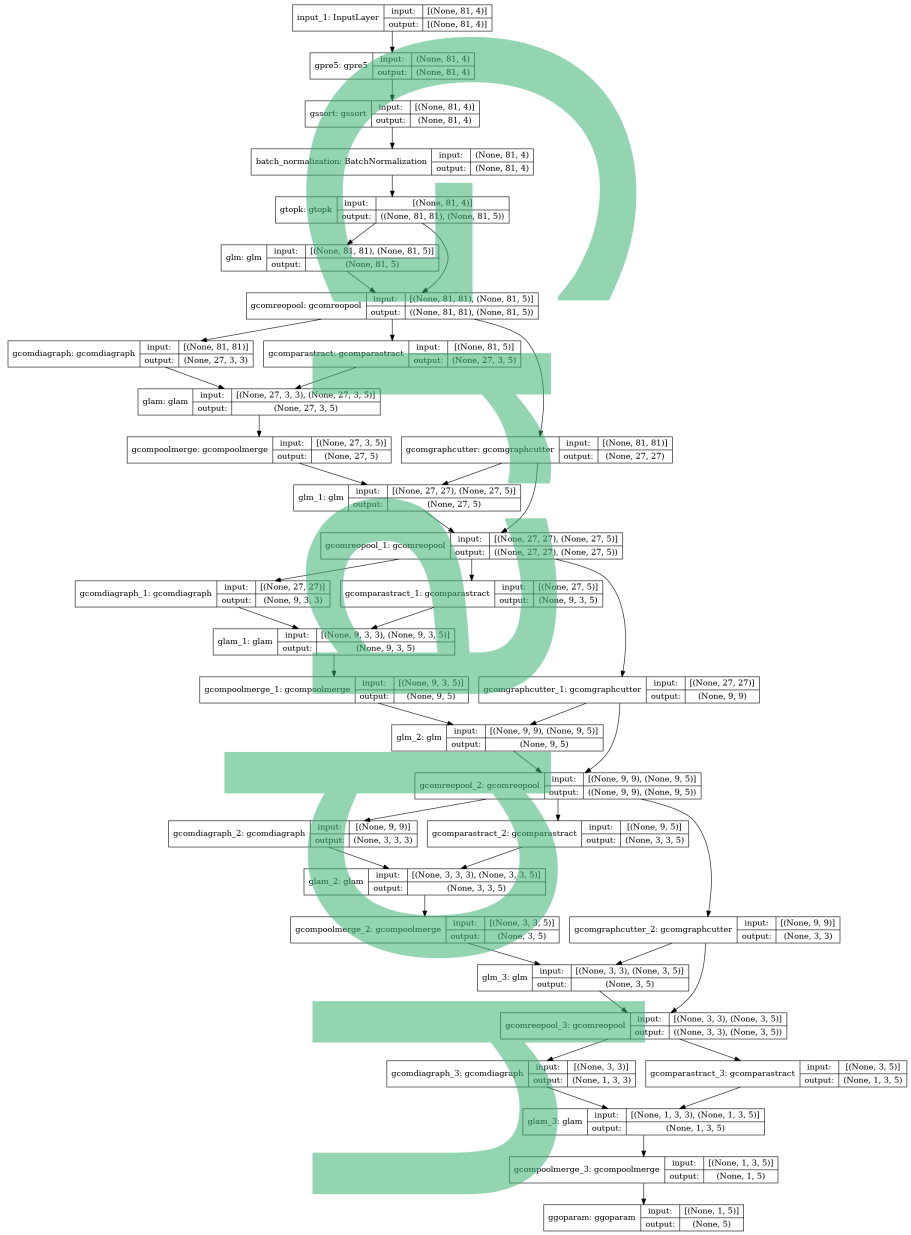




- ParticleNet
  - supervised
  - Graph Neuronal Networks
- Qcd Or What?
  - unsupervised
  - Convolutional Networks

- Combine ideas from both
- Into a Graph Autoencoder


input_1: InputLayer
input: [(None, 81, 4)]
output: [(None, 81, 4)]
gpre5: gpre5
input: (None, 81, 4)
output: (None, 81, 4)
gssort: gssort
input: [(None, 81, 4)]
output: (None, 81, 4)
ggoparam: ggoparam
input: [(None, 81, 4)]
output: (None, 324)
dense: Dense
input: (None, 324)
output: (None, 324)
dense_1: Dense
input: (None, 324)
output: (None, 259)
dense_2: Dense
input: (None, 259)
output: (None, 194)
dense_3: Dense
input: (None, 194)
output: (None, 145)
dense_4: Dense
input: (None, 145)
output: (None, 97)
ggoparam_1: ggoparam
input: [(None, 97)]
output: (None, 97)


input_1: InputLayer
gpre5: gpre5
gssort: gssort
batch_normalization: BatchNormalization
gtopk: gtopk
glm: glm
gcomreopool: gcomreopool
gcomdiagraph: gcomdiagraph
gcomparastract: gcomparastract
glam: glam
gcompoolmerge: gcompoolmerge
gcomgraphcutter: gcomgraphcutter
glm_1: glm
gcomreopool_1: gcomreopool
gcomdiagraph_1: gcomdiagraph
gcomparastract_1: gcomparastract
glam_1: glam
gcompoolmerge_1: gcompoolmerge
gcomgraphcutter_1: gcomgraphcutter
glm_2: glm
gcomreopool_2: gcomreopool
gcomdiagraph_2: gcomdiagraph
gcomparastract_2: gcomparastract
glam_2: glam
gcompoolmerge_2: gcompoolmerge
gcomgraphcutter_2: gcomgraphcutter
glm_3: glm
gcomreopool_3: gcomreopool
gcomdiagraph_3: gcomdiagraph
gcomparastract_3: gcomparastract
glam_3: glam
gcompoolmerge_3: gcompoolmerge
ggoparam: ggoparam

Assuming

The higher the loss of the comparison network

The more likely this is a top jet.

You reach an AUC of

0.908

- Already fairly satisfied
- If I beat this, my graph networks work

- AUC score
- between 0 and 1, higher=better
- 1:perfect,0.5:random

Assuming

The higher the loss of my graph network

The more likely this is a top jet.

You reach an AUC of

0.910

- So my network works
- But...

- AUC score
- between 0 and 1, higher=better
- 1:perfect,0.5:random

Assuming

The higher R**2

The more likely this is a top jet.

You reach an AUC of

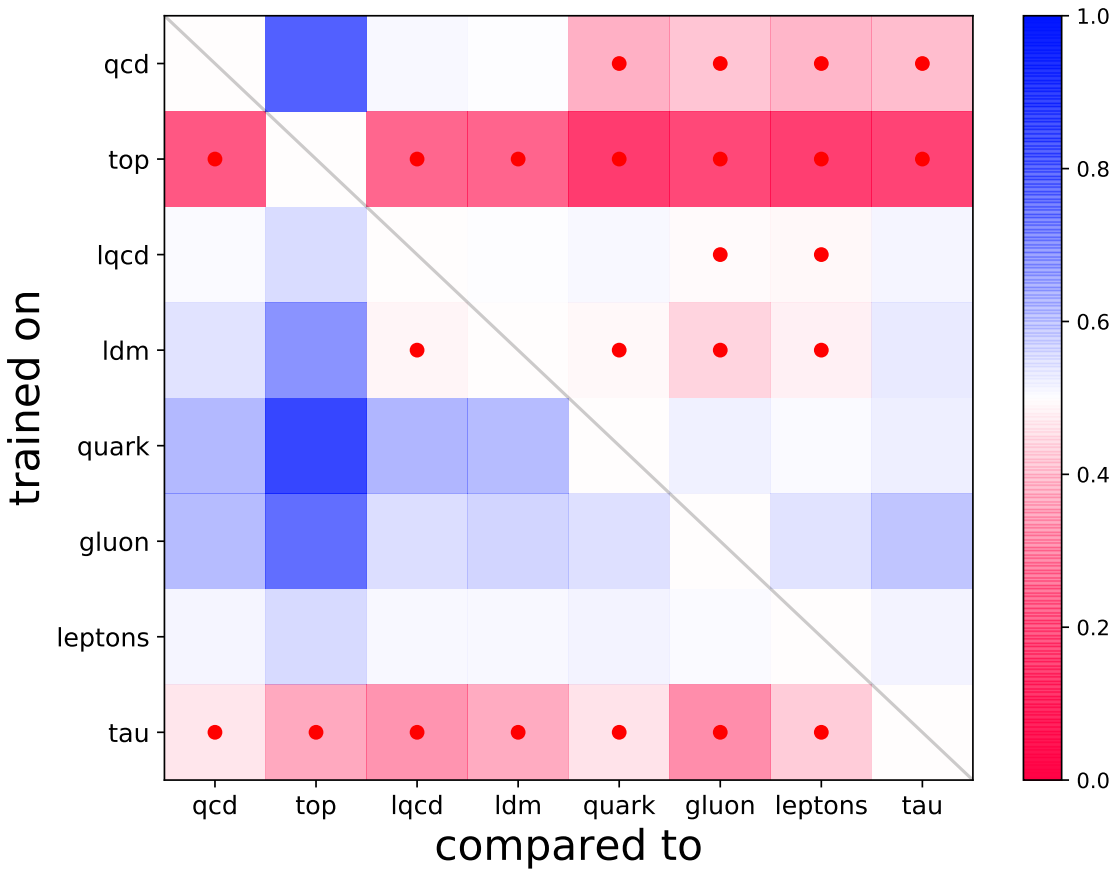
0.915

- trivial network
- best score yet

- AUC score
- between 0 and 1, higher=better
- 1:perfect,0.5:random

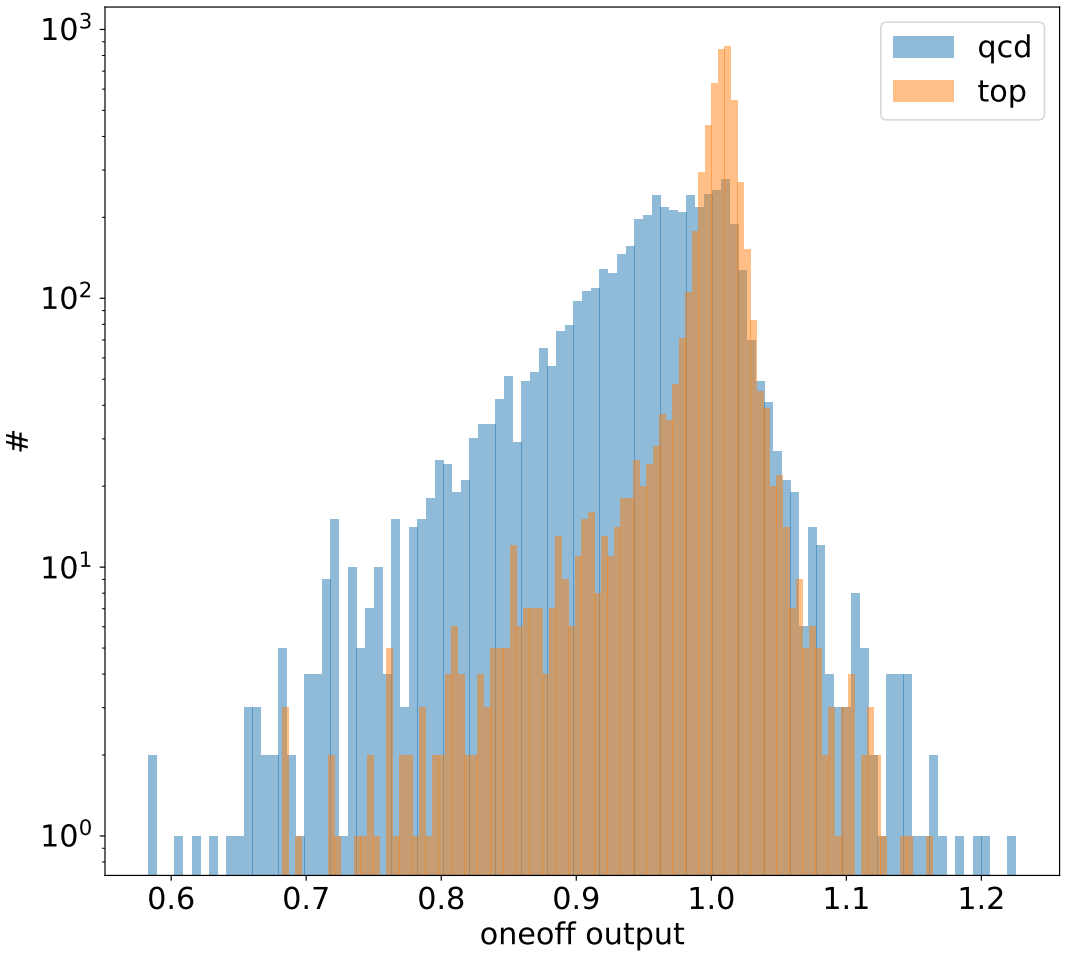
Results

- The more blue the better, and if a pixel is red (dot) it is not detectable
- For the comparison work only 10(32)/56 are detectable
- useless, except for qcd vs top



OneOff Networks

- So improve it using my own algorithm:
- Train a network to output a constant
- $loss = (f(x) - 1)^2$
- Anomalous data usually does not reproduce the same constant



Results

- The more blue the better, and if a pixel is red (dot) it is not detectable
- Quality is not final, but
- here do 48(52)/56 comparisons work