Reaching for the Top with GNN

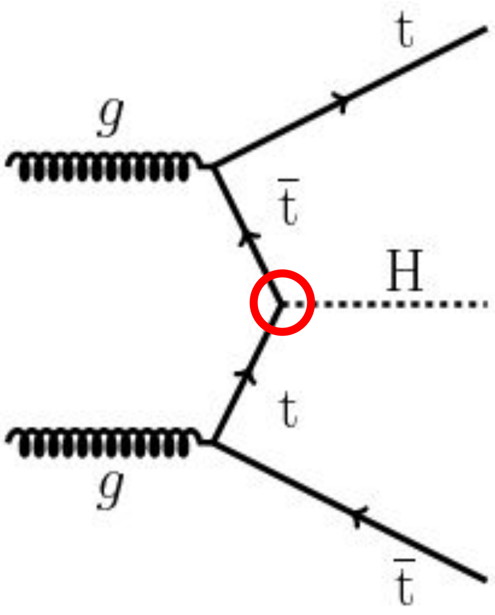
Applications of a Graph Neural Network for HEP Analysis

Presented at the 4th IML Machine Learning Workshop 21 October, 2020

Shuo Han, Xiangyang Ju, Pamela Pajarillo, Ryan Roberts, Haichen Wang, Allison Xu

Difficulty of Event Classification for ttH(H→$\gamma\gamma$)

- Collider data analysis often involves processes with complex final states, making the task of classifying events a challenging one
- For example ttH(H→$\gamma\gamma$), physically interesting for measurements of the top-Higgs Yukawa coupling, involves the decays of 3 heavy particles, leading to high multiplicity events
- There are 2 major backgrounds, presenting different challenges
  - $\gamma\gamma$+jets - large combinatoric background from events with a lot of light jet activity and 2 photons
  - tt$\gamma\gamma$ - rarer background, but very hard to distinguish from signal due to the presence of 2 real tops
- We model our pre-selection and BDT training on the ATLAS ttH observation paper: arxiv



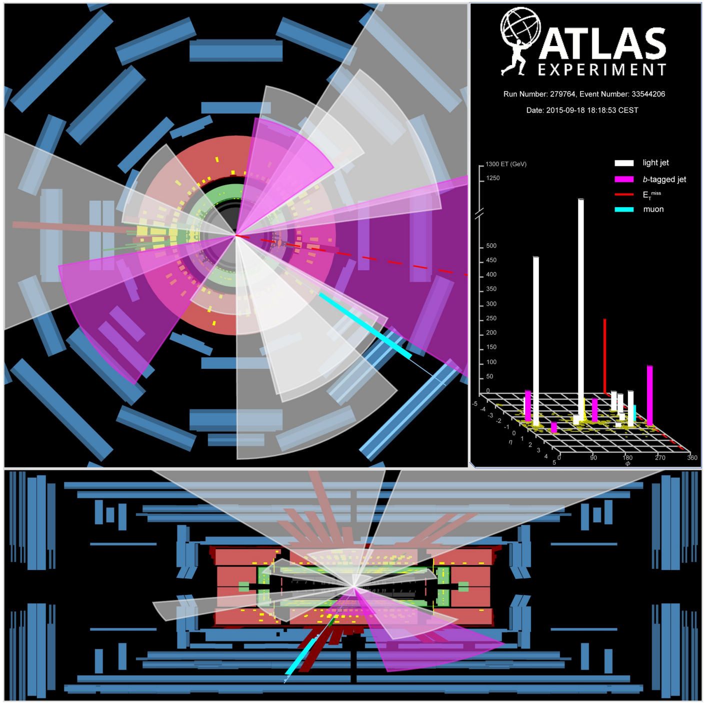
Our Solution

We use a Graph Neural Network (GNN) based on the `graph_nets` package for Event Classification and Top Reconstruction in the ttH(H→$\gamma\gamma$) process. See e.g. Peter Battaglia's talk for more info on GNN's.

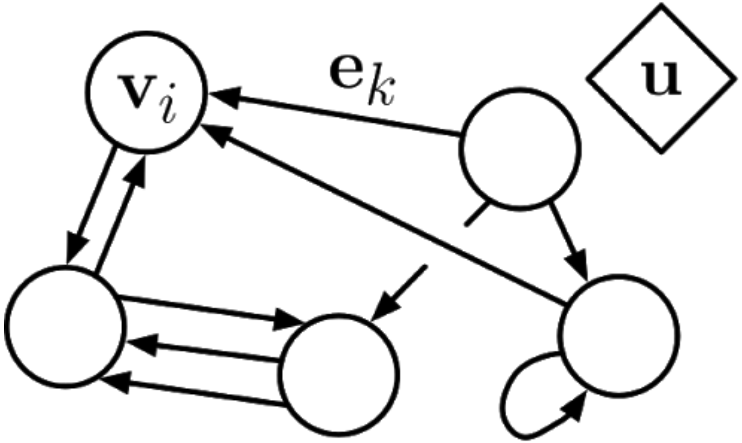
1. How to represent HEP events as graphs
2. GNN Event Classification
3. Top reconstruction

https://arxiv.org/pdf/1806.01261.pdf

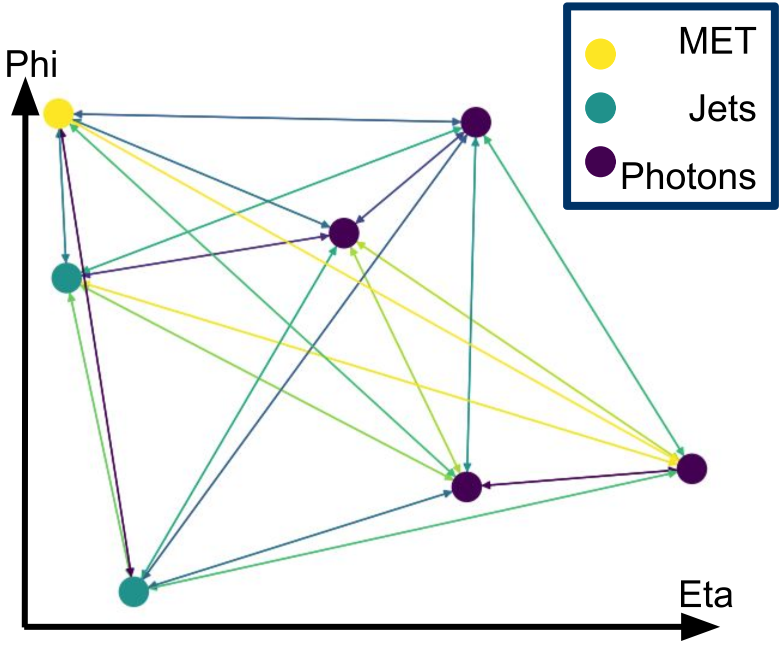
GNN for HEP Analysis: How to Make a Graph



?

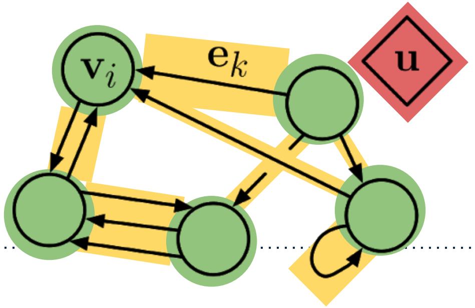


Representing HEP Events as Graphs



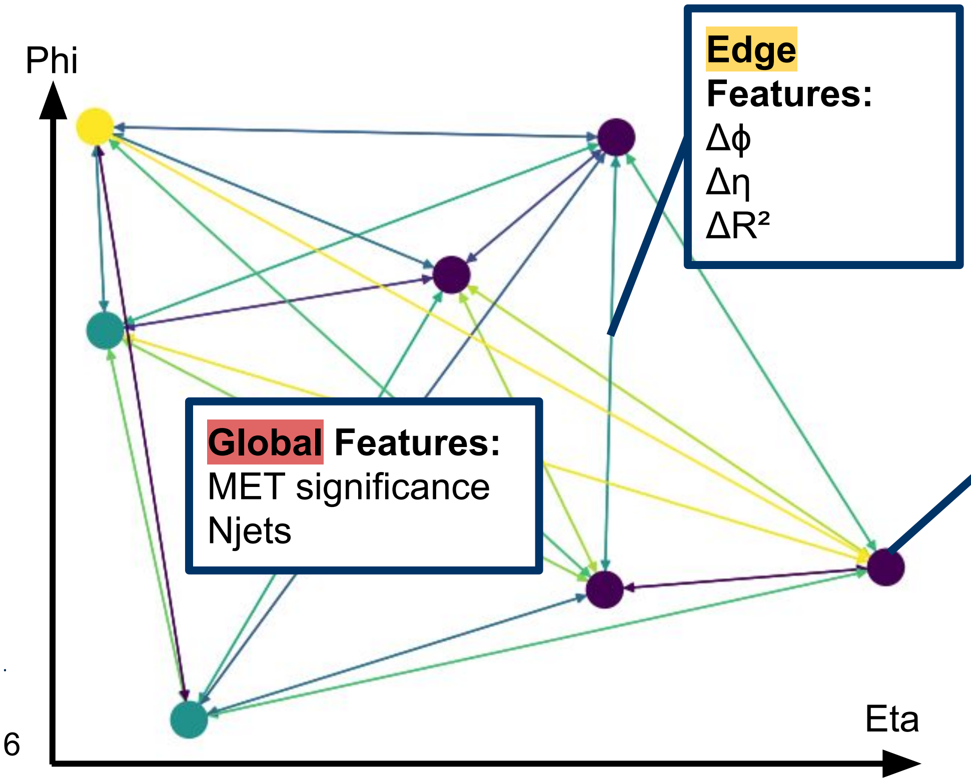
Nodes are colored by particle type, edges by ΔR^2

We use reconstructed objects as **nodes** (jets, photons, leptons, and MET).
The **edges** are bidirectional and connect all pairs of reconstructed objects.
We can also supply some **global** features.
The graph uses all reconstructed objects and has no ordering dependence.



Representing HEP Events as Graphs

- Store 4-momenta and other information as node features
- Fully connected graph with angular information stored as edge features
- Implies bidirectional message passing between all nodes



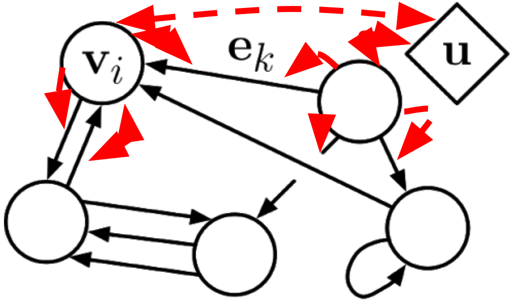
Edge Features:
$\Delta\phi$
$\Delta\eta$
ΔR^2

Global Features:
MET significance
Njets

Node Features:
Pt
Eta
Phi
E
B-tagging (if jet)
Charge (if lepton)
Node type

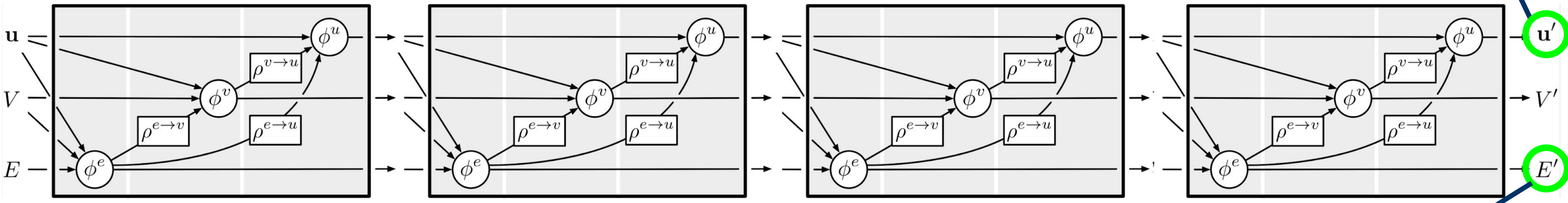
- Event level variables can be used as global features

GNN Refresher



- Message Passing - Local and global sharing of information around the graph
- We expect the GNN to be particularly well-suited for processes with high multiplicity and complex structure.

GNN Score for Event Classification



GNN Score for Edge Classification

https://arxiv.org/pdf/1806.01261.pdf

Sample Preparation/Event Selection

- We generated samples of SM ttH at NLO, tt+$\gamma\gamma$, and $\gamma\gamma$+jets at LO using Pythia8, MadGraph5_aMC@NLO, and Delphes for event generation, parton showering, and detector simulation, respectively
- We use a pre-selection matching the ATLAS ttH(H→$\gamma\gamma$) analysis:
  - Require 2 reconstructed photons with diphoton mass 105 GeV < m_{yy} < 160 GeV
  - Cut on relative photon pt: $pt_{y1}/m_{yy} > 0.35$ and $pt_{y1}/m_{yy} > 0.25$
  - Require at least 3 jets, including at least 1 b-tagged jet
- No separation into hadronic and leptonic channels
- We train on
  - 900k ttH events
  - 375k ttyy events
  - 150k yy events

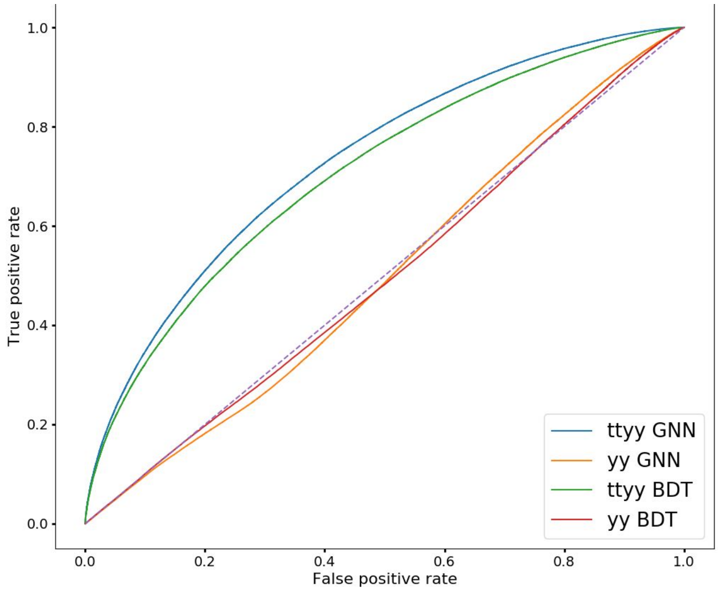
GNN and BDT Training Preparation

As a comparison for the GNN, we trained a BDT using the XGBoost package, which is designed to be as similar as possible to the GNN

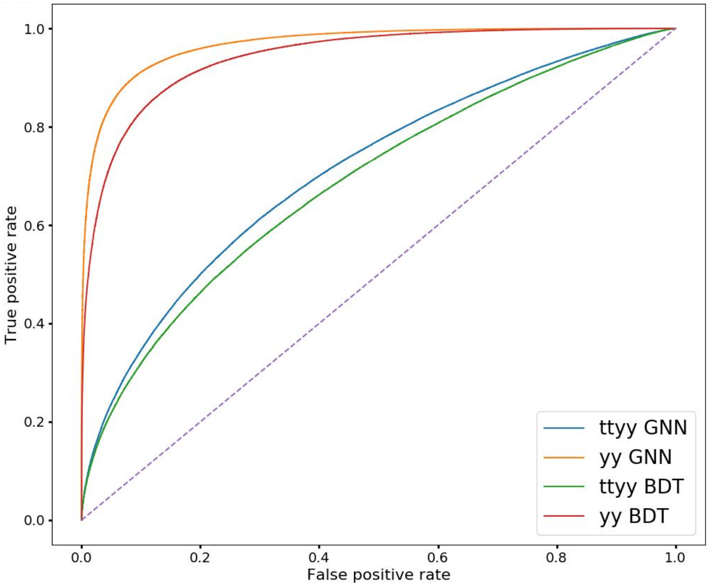
- The BDT and GNN use the same event pre-selection and the same division into testing and training sets
- As input, the BDT uses all 4-momenta of photons, jets, leptons, and MET (up to 8 leading jets), as well as b-tagging of jets, essentially identical to the input information of the GNN
- While the GNN uses our “default” hyperparameters, which are chosen loosely based on past experience, the BDT goes through a process of hyperparameter optimization as done in ATLAS HIGG-2019-01

GNN and BDT Event Classification ROC’s

ttH signal vs. ttyy background



ttH signal vs. yy background

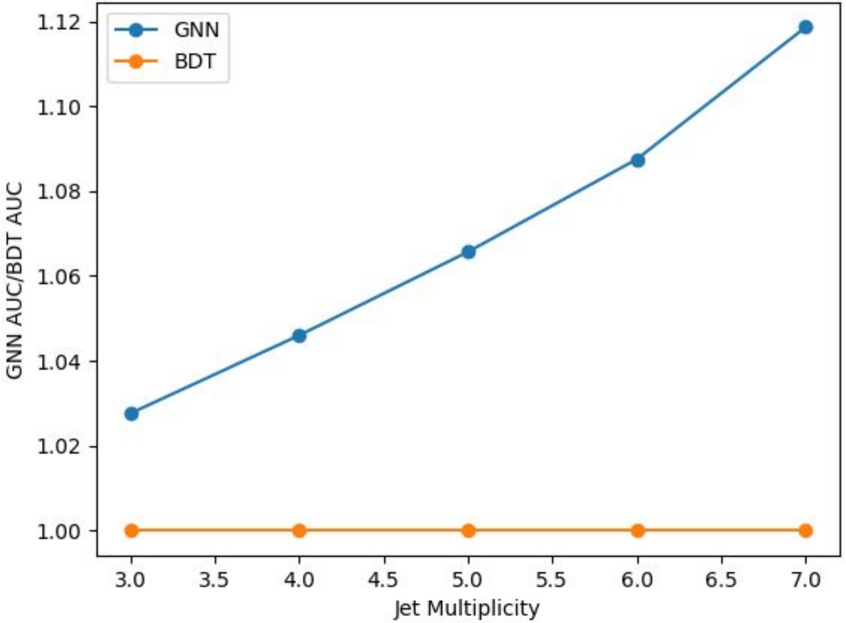


Model	Bkg	AUC
GNN	ttyy	0.730
BDT	ttyy	0.706
GNN	yy	0.968
BDT	yy	0.942

We train the GNN and BDT separately on each background sample in order to evaluate the performance in two very different scenarios. A realistic analysis is an interpolation of these two results. The GNN outperforms the BDT in both cases.

GNN Improves with Jet Multiplicity

We expect that the GNN's natural way of representing information and flexible number of objects will lead to performance increasing with multiplicity/event complexity. In both trainings, we see significant improvement in the GNN performance relative to the BDT as we go to regions of higher jet multiplicity.

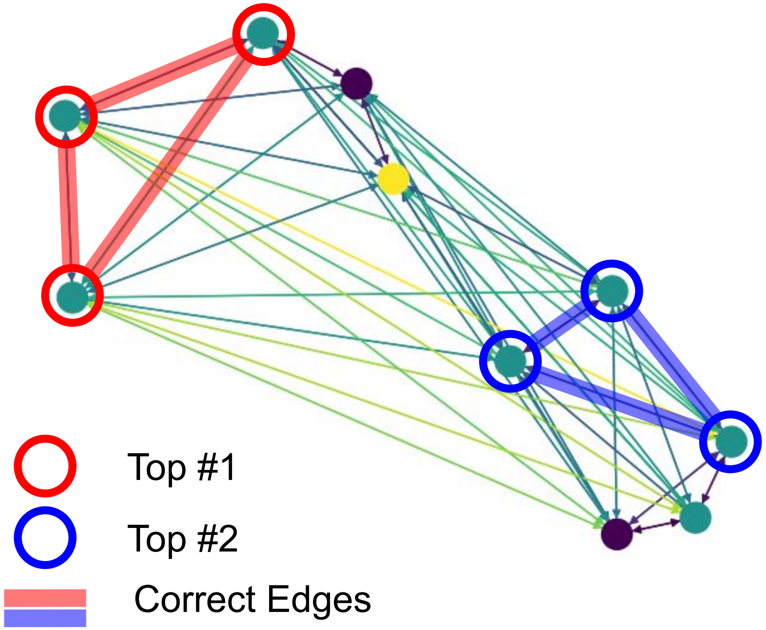


Ratio of GNN and BDT AUC's as jet multiplicity increases from 3 to 6. The last bin is inclusive: >= 7 jets.

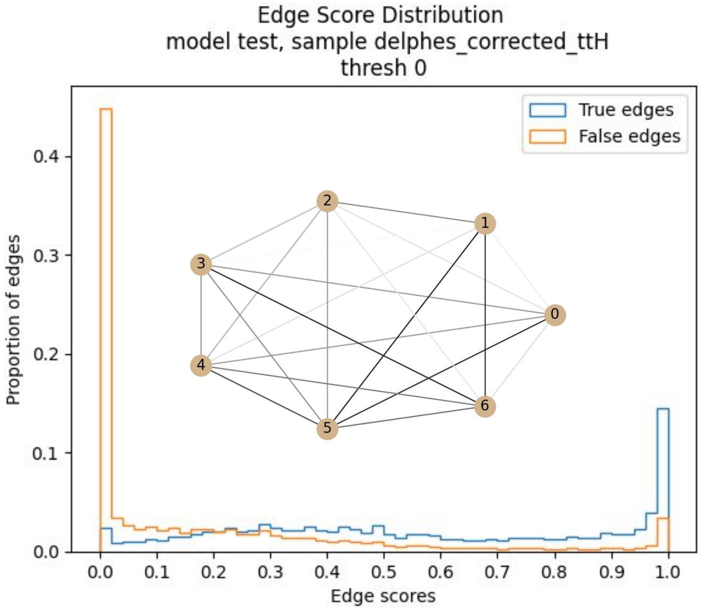
Edge Classifier based Top Reconstruction with GNN

For us, a top is a set of 3 jets, called a triplet. Top reconstruction is the process of correctly identifying triplets which correspond to a true top, and is a difficult combinatorial problem.

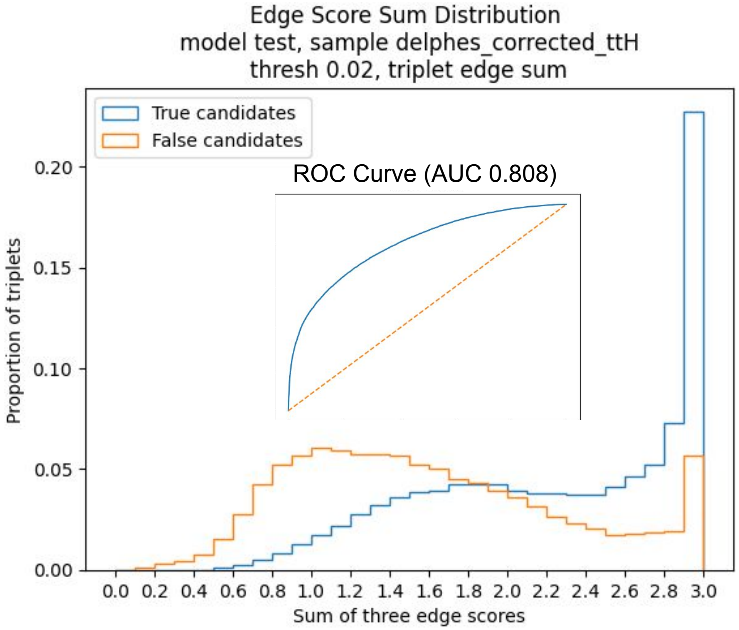
We perform edge classification with the GNN to reconstruct tops. True edges are those that connect 2 jets in the same triplet. To reconstruct triplets, we combine sets of edges which form triplets and which have high edge scores.



GNN Top Reconstruction with Edge Classification



Edge Score Distribution
model test, sample delphes_corrected_ttH
thresh 0



Edge Score Sum Distribution
model test, sample delphes_corrected_ttH
thresh 0.02, triplet edge sum

The GNN output is a score assigned to each edge of the graph. The GNN effectively learns to discriminate true and false edges.

To construct triplets from these edge scores, we first apply a threshold on the edge score, and then use the sum of edge scores in a triplet as a discriminant.

This post-threshold sum of edge scores is an effective discriminant, yielding a top reconstruction efficiency of 53-56%.

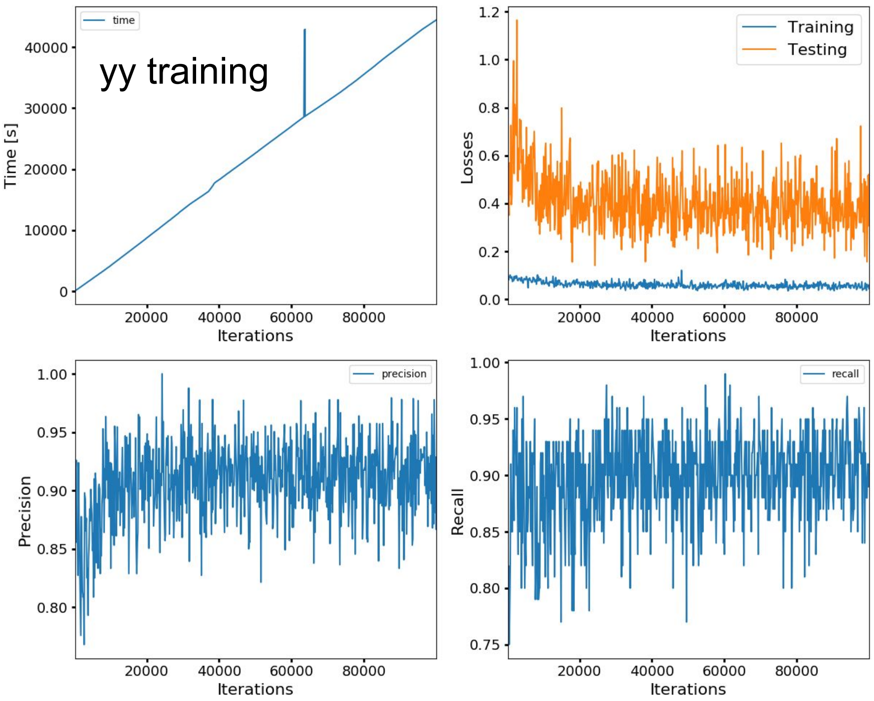
Outlook

- The GNN is a natural solution for event classification in the ttH($H \rightarrow \gamma\gamma$) process
- We have a robust demonstration that the GNN is capable of outperforming a BDT, especially at high multiplicity
- Studies are underway to further elucidate where the performance gain comes from
- It could be generalized in a straightforward way to any analysis dealing with complex final states
- We have also demonstrated a novel method of top reconstruction based on edge classification
- The possibility of using edge and node classification in tandem with event classification is an exciting prospect

Backup

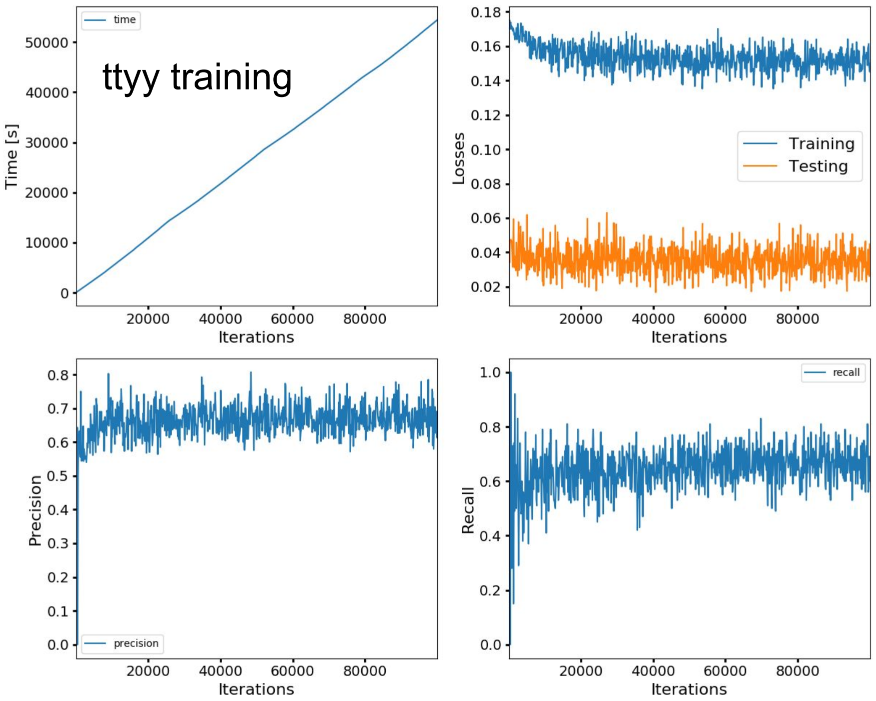
GNN Training Logs

Training info as function of time delphes_tth_yy4



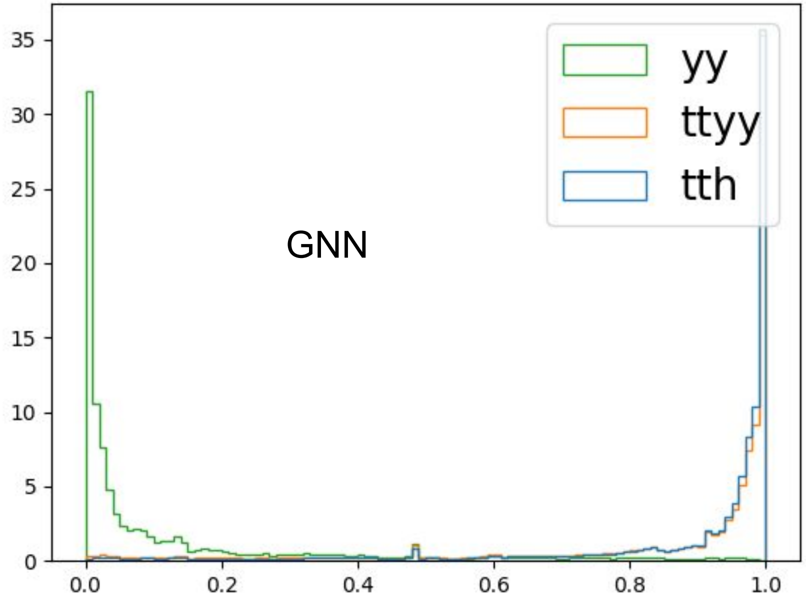
yy training

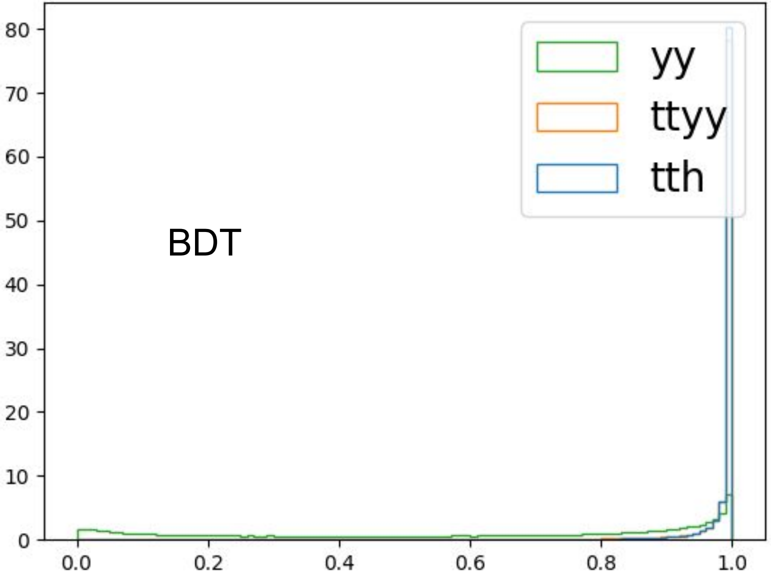
Training info as function of time delphes_tth_ttyy4



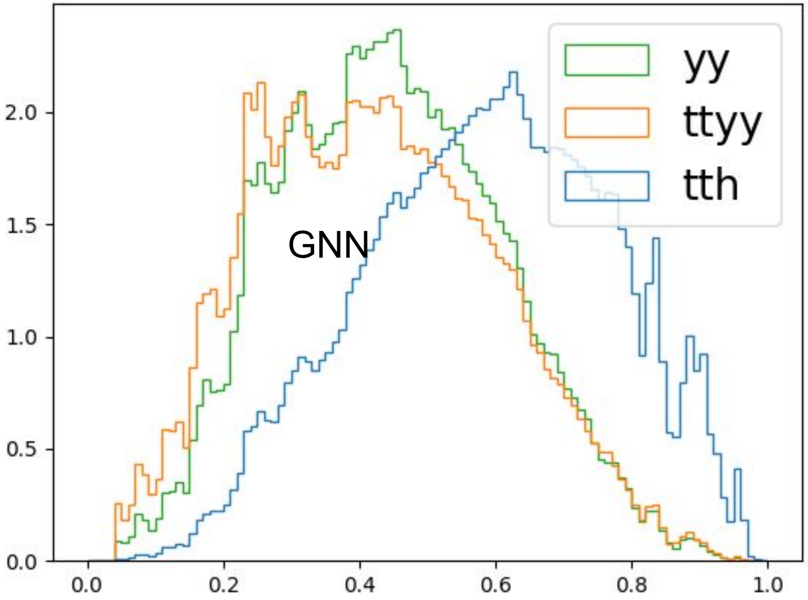
ttyy training

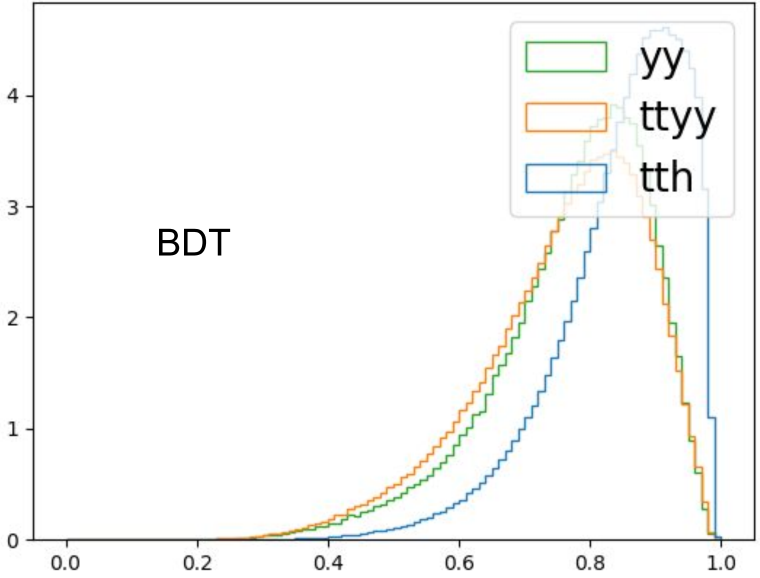
yy training Score Distributions





ttyy training Score Distributions





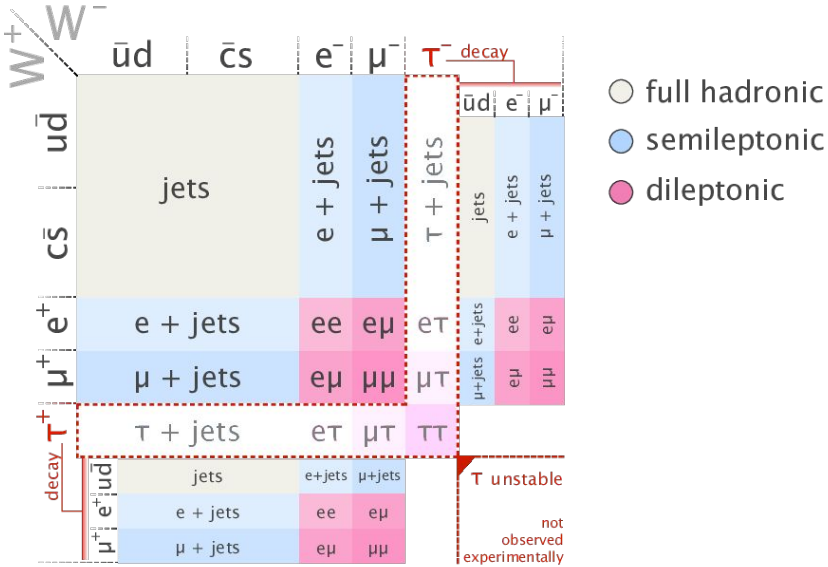
AUC Tables

Train Bkg	Eval Bkg	Fold	GNN AUC	BDT AUC	N b-jets AUC	#Signal Evts	#Bkg Evts
ttyy	ttyy	Training	0.735	0.719	0.514	905445	376243
ttyy	ttyy	Testing	0.730	0.706	0.514	904782	376407
ttyy	yy	Training	0.715	0.694	0.681	905445	150791
ttyy	yy	Testing	0.712	0.688	0.681	904782	150155
yy	ttyy	Training	0.495	0.492	0.514	905445	376243
yy	ttyy	Testing	0.497	0.495	0.514	904782	376407
yy	yy	Training	0.970	0.947	0.681	905445	150791
yy	yy	Testing	0.968	0.942	0.681	904782	150155

Train Bkg	Eval Bkg	Njets	GNN AUC	BDT AUC	N b-jets AUC	#Signal Evts	#Bkg Evts	GNN-BDT	GNN/BDT
ttyy	ttyy	3	0.737	0.716	0.518	158026	62334	0.021	1.030
ttyy	ttyy	4	0.731	0.709	0.520	241389	95419	0.022	1.031
ttyy	ttyy	5	0.727	0.702	0.517	234351	98138	0.025	1.035
ttyy	ttyy	6	0.728	0.699	0.513	156082	69093	0.029	1.041
ttyy	ttyy	7	0.725	0.690	0.507	114934	51423	0.035	1.051
ttyy	yy	3	0.712	0.688	0.681	904782	150155	0.024	1.035
ttyy	yy	4	0.709	0.677	0.680	746756	57654	0.032	1.047
ttyy	yy	5	0.697	0.676	0.678	505367	18180	0.021	1.032
ttyy	yy	6	0.683	0.666	0.676	271016	4901	0.018	1.027
ttyy	yy	7	0.668	0.650	0.676	114934	1150	0.019	1.029
yy	ttyy	3	0.480	0.489	0.518	158026	62334	-0.009	0.982
yy	ttyy	4	0.496	0.499	0.520	241389	95419	-0.003	0.993
yy	ttyy	5	0.520	0.525	0.517	234351	98138	-0.006	0.989
yy	ttyy	6	0.529	0.548	0.513	156082	69093	-0.019	0.966
yy	ttyy	7	0.522	0.543	0.507	114934	51423	-0.021	0.961
yy	yy	3	0.968	0.942	0.681	904782	150155	0.026	1.028
yy	yy	4	0.955	0.913	0.680	746756	57654	0.042	1.046
yy	yy	5	0.941	0.883	0.678	505367	18180	0.058	1.066
yy	yy	6	0.932	0.857	0.676	271016	4901	0.075	1.088
yy	yy	7	0.924	0.826	0.676	114934	1150	0.098	1.118

Physics Application: Top Reconstruction

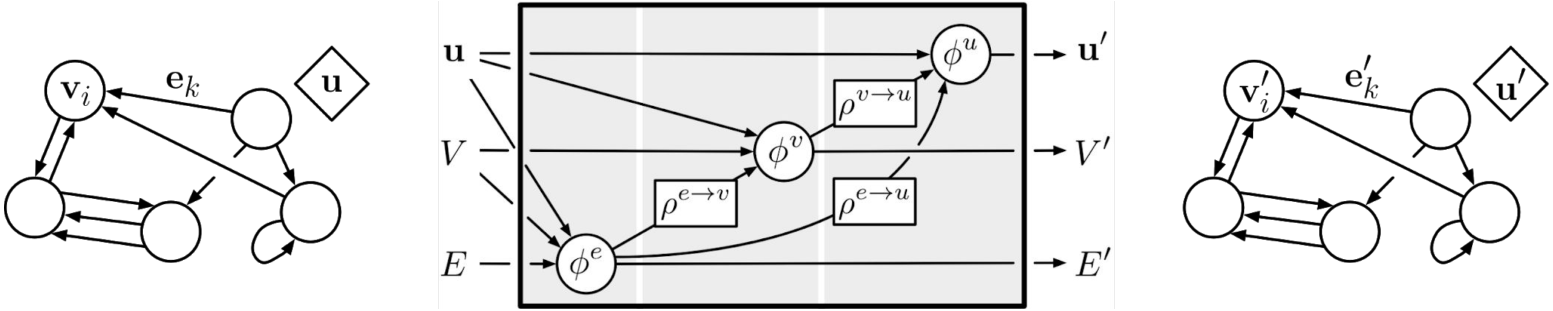
- Top quark decays produce 2-3 reconstructed physics objects with a variety of combinations
- Correctly identifying which groups of jets and leptons which come from the same top quark is a difficult combinatorial problem
- Effective top reconstruction is useful for background rejection and is crucial for differential measurements
- Current approaches include BDT based top reconstruction and some non-ML approaches
  - Unit of analysis is the triplet set of 3 jets
  - Leptonic and hadronic cases treated separately



https://en.wikipedia.org/wiki/Top_quark

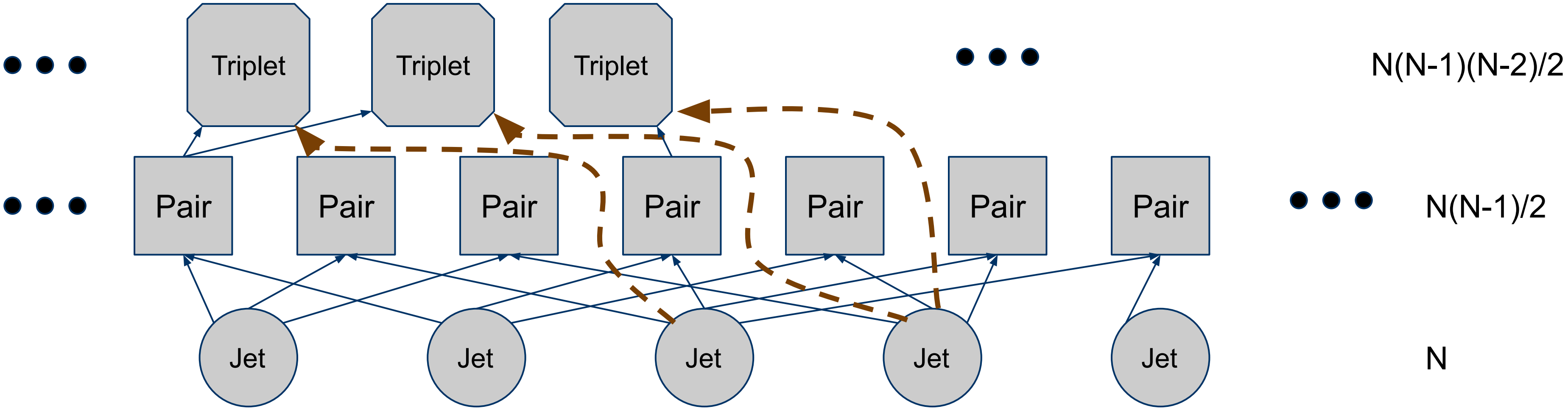
GNN Basics: Functions of Graphs

- Take as a graph as input and as output
  - Simplest case is to output a single number
- Φ - Generic function $\mathbb{R}^n$□$\mathbb{R}^m$ (neural network block)
- ρ - Message passing step - aggregates outputs, must be symmetric in its inputs $(\mathbb{R}^n)$□$\mathbb{R}^n$ (average over inputs)
-



https://arxiv.org/pdf/1806.01261.pdf

Top Reconstruction with GNN Node Classification and Hierarchical Graphs

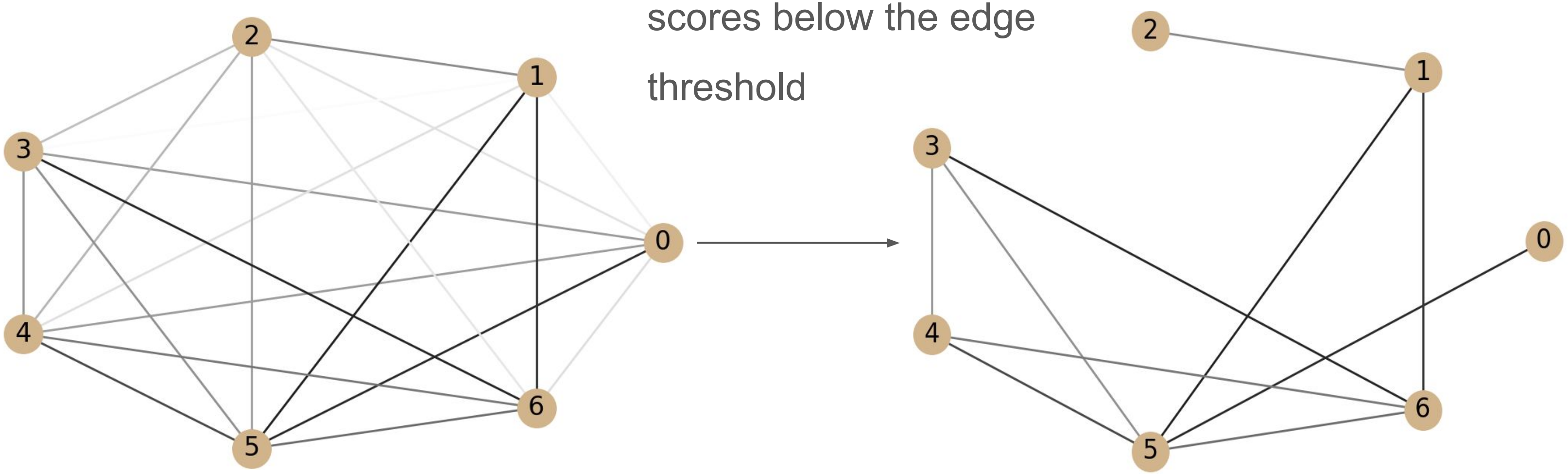


We are free to design our graphs any way we want. We want to identify triplets, so can we make nodes that correspond to triplets?

This graph structure is partially inspired by the idea that a top triplet should consist of a pair of jets from a W decay and a b-jet

GNN Top Reconstruction Algorithm

1. Remove edges with scores below the edge threshold



GNN Top Reconstruction Algorithm

2. Construct all possible triplets from the remaining edges
3. Score each triplet (e.g. sum of three edge scores)

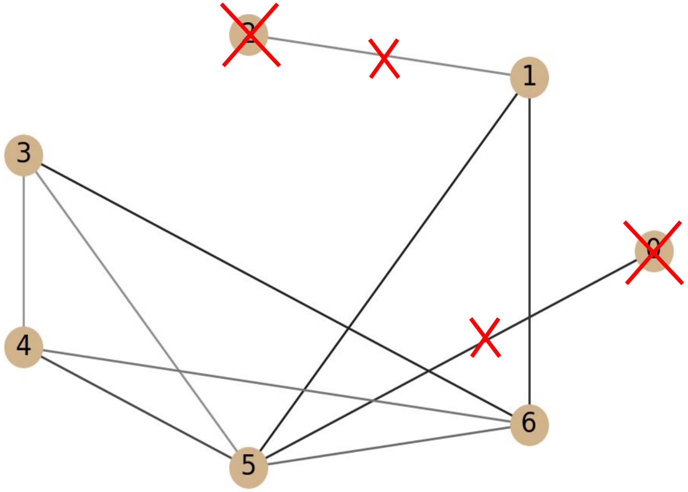
(1, 5, 6)

(3, 4, 5)

(3, 4, 6)

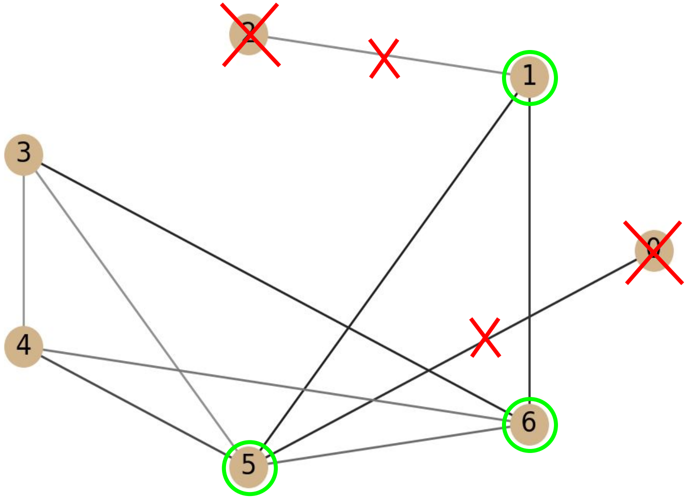
(3, 5, 6)

(4, 5, 6)



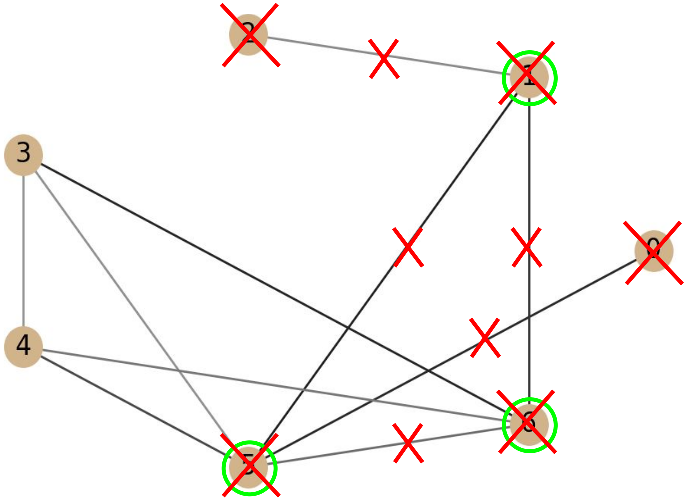
GNN Top Reconstruction Algorithm

4. Select the highest scoring triplet, if possible



GNN Top Reconstruction Algorithm

5. Eliminate triplets containing any of the jets in the highest scoring triplet



GNN Top Reconstruction Algorithm

5. Eliminate triplets containing any of the jets in the highest scoring triplet
6. Select the next highest scoring triplet, if possible