Graph Neural Network: Its applications to constrain BSM models and EFTs

Akanksha Bhardwaj

Oklahoma State University

JHEP 08 (2021) 080  O. Atkinson, A.B, C. Englert, V.Ngairangbam, M. Spannowsky

JHEP 04 (2022) 137  O. Atkinson, A.B, S. Brown, C. Englert, D. Miller, P. Stylianou
Graph Neural Networks (GNNs) are powerful deep learning algorithm for automatic feature extraction on graph structured data

- General enough for non-Euclidean physics data
- Can encode pairwise relation through edge features

Graph Autoencoder for unsupervised detection of non-QCD jets:

- Design a symmetric decoder capable of simultaneously reconstructing edge features and node features

Application of GNNs in a supervised scenario: an SMEFT analysis of top-antitop pair production and its decay

- Multiclass classification of thirteen independent Wilson coefficients switched on simultaneously
Artificial Neural Networks

\[ z_1 = \sigma(\sum \alpha_i^1 x_i + \alpha_0^1) \]

\[ z_2 = \sigma(\sum \alpha_i^2 x_i + \alpha_0^2) \]

\[ z_3 = \sigma(\sum \alpha_i^3 x_i + \alpha_0^3) \]

\[ z_4 = \sigma(\sum \alpha_i^4 x_i + \alpha_0^4) \]

\[ z_5 = \sigma(\sum \alpha_i^5 x_i + \alpha_0^5) \]

\[ z_6 = \sigma(\sum \alpha_i^6 x_i + \alpha_0^6) \]

\[ z_7 = \sigma(\sum \alpha_i^7 x_i + \alpha_0^7) \]

\[ Y_1 = \frac{e^{y_1}}{\sum_{i=1}^{5} e^{y_i}} \]

\[ Y_2 = \frac{e^{y_2}}{\sum_{i=1}^{5} e^{y_i}} \]

\[ Y_3 = \frac{e^{y_3}}{\sum_{i=1}^{5} e^{y_i}} \]

\[ Y_4 = \frac{e^{y_4}}{\sum_{i=1}^{5} e^{y_i}} \]

\[ Y_5 = \frac{e^{y_5}}{\sum_{i=1}^{5} e^{y_i}} \]

\[ y_j = \sum_{i=1}^{7} w_j^i z_i + w_0^j \]
What are Graphs?

- A set of objects, and the relations between a pair of objects are naturally expressed as a *graph*.
- Graph Neural Networks (GNNs) operate on graph data.
- To further describe each node, edge or the entire graph, we can store information in each of these pieces of the graph.
  - Vertex (or node) embedding
  - Edge (or link) attributes and embedding
  - Global (or graph) embedding
Message passing operation

1. Message Passing
   \[ m_{ij} = \mathbf{M}(h_i, h_j, e_{ij}) \]

   Neural Network Shared for all edges

2. Node Readout
   \[ h'_i = \mathbf{A}(h_i, \{m_{ik} \mid k \in \mathcal{N}(i)\}) \]
Jet Substructure at LHC

First splitting/decay at Parton-level

Dominantly Soft or collinear
\( z_2 \ll z_1 \) or \( \theta_{12} \to 0 \)

Democratic splitting
\( z_2 \sim z_1 \)

QCD jets (Background)

\( z_i \) is relative hardness

for hadronic colliders:

\[
    z_i = \frac{p_T^i}{\sum_{j=1}^n p_T^j}
\]

\[
    \hat{p}_i = (y_i, \phi_i)
\]

Higgs jets (2-prong Signal)
Autoencoders are neural networks that map an input space to a bottleneck dimension (the latent dimension) and then back again to a space identical to the input.

Train on background (most likely events) => higher reconstruction error on previously unseen events = Anomaly
Graph Autoencoder

Adjacency matrices:

\[ A^a_{ij} = A^a_{ji} = \begin{cases} e_{ij}^a & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases}, \]

Inner-product layer:

\[ \hat{A}_{ij} = \vec{h}_i \cdot \vec{h}_j, \]

\( e_{ij}^a \): components of Edge feature vector.
\( \vec{h}_i \): Node-features of preceding layer vectors.

Network also reconstructs three \( N \times N \) adjacency matrices for each graph.
Graph Autoencoder

Loss-function:

$$L_{node} = \sqrt{\sum_{ia} \frac{(\hat{x}_i^a - x_i^a)^2}{N \times 5}}$$
$$L_{edge} = \sum_a \sqrt{\sum_{ij} \frac{(\hat{A}_{ij}^a - A_{ij}^a)^2}{N \times N}}$$

$$L_{auto} = \lambda_{node} L_{node} + \lambda_{edge} L_{edge}$$

$$\lambda_{node} = 0.3 \text{ and } \lambda_{edge} = 1.$$
Latent dimension scan

Choose latent dimension = 6
Loss distribution for latent dimension 6

![Graph showing loss distribution with different categories and signal efficiency plot with AUC values for QCD vs W, QCD vs t, and QCD vs $\phi$.](image-url)
Loss correlations

Shared weights for all edges per layer

QCD jets:

Learning a uniform structure regardless of the number of nodes?
Loss correlation

Signal Jets: Extra radiation other than their multiplicities arise from QCD radiation
Graph Neural Network: Application to EFT in supervised learning framework
Effective Field Theory

- Lack of experimental evidence of new physics may indicate a mass gap between SM and BSM scales.
- SMEFT Lagrangian

\[ \mathcal{L}_{\text{SMEFT}} = \mathcal{L}_{\text{SM}} + \sum_{i>4} \sum_{k} \frac{C^{(i)}_k}{\Lambda^{i-4}} \mathcal{O}^{(i)}_k \]

- Top-down:
  - Integrate out heavy BSM states of UV-complete theory.
  - Match Wilson Coefficients (WCs) to variables of full theory
Effective Field Theory

- Bottom-up:
  - Agnostically include all allowed operator deformations as an expansion around $\Lambda^{-1}$.
  - Truncate the series to and interpret LHC results as bounds on Wilson Coefficients $C_k^{(i)}$ of the operators.

- Only one $\mathcal{O}^{(5)}$ which is relevant for neutrino physics.

- With minimal flavour violation and baryon number conservation there are 59 operators $\mathcal{O}_k^{(6)}$.

- 16 dimension-6 operators relevant for top physics.

- Keep terms in Lagrangian only up to $\Lambda^{-2}$. 
Any differential cross section follows:

\[ d\sigma = d\sigma_{\text{SM}} + \frac{C_i}{\Lambda^2} d\sigma_i + \frac{C_i C_j}{\Lambda^4} d\sigma_{ij} \]

- $\Lambda^{-4}$ terms are suppressed, truncate series at $\Lambda^{-2}$
- Differential distributions can be used to quantify allowed range on WCs.
- Optimised selection of signal region can result in improved bounds.
- Any improvement for linear case should generalise to $\Lambda^{-4}$ terms.
Improving SMEFT results with GNNs

- Focus on process $pp \rightarrow t\bar{t} \rightarrow \ell b \bar{b} j j + E_T$
- 13 relevant SMEFT operators in this process.

\[ X^3: \mathcal{O}_G \]
\[ \psi^2 X \varphi: \mathcal{O}_u^{33}, \mathcal{O}_u^{33} \]
\[ \psi^2 \varphi^2 D: \mathcal{O}_q^{33} \]
\[ \psi^4: \mathcal{O}_{qq}^{(1)i33i}, \mathcal{O}_{qq}^{(3)i33i}, \mathcal{O}_{qq}^{(3)ii33}, \mathcal{O}_{qu}^{(8)33ii}, \mathcal{O}_{qu}^{(8)ii33}, \mathcal{O}_{ud}^{(8)33ii}, \]
\[ \mathcal{O}_{uu}, \mathcal{O}_{qd}, \mathcal{O}_{lq}^{(3)ii33} \]

lepton-quark operator
Particle event as a graph

1. Usually graphs are constructed either w.r.t. the distance between the nodes or fully connected.

2. Here instead attempt to use systematic procedure to construct physics-inspired graphs from final states:
   a. Impose basic selection criteria on jets, leptons and b-quarks
   b. Create nodes for jets, lepton, b-quarks and missing transverse momentum (MTM).
   c. Attempt to reconstruct invariant mass of the two W bosons from lepton-MTM and dijet.
   d. If reconstructed masses are relatively close to actual W mass create node.
   e. Attempt to reconstruct top quarks and create nodes.
Particle events as Graph

Edges are connected according to the decay from their parent particles
Graph Features

Node Features

- Transverse momentum
- Azimuthal angle
- Pseudorapidity
- Energy
- Invariant mass
- Particle Identity

No Edge feature in our set-up

**Goal:** GNN should utilize the features and graph structure to classify events according to the operator that gave rise to it or as purely-SM
GNN Architecture

Edge Convolution:

$$
\tilde{x}_i^{(l+1)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \text{ReLU} \left( \Theta \cdot (\tilde{x}_j^{(l)} - \tilde{x}_i^{(l)}) + \Phi \cdot (\tilde{x}_i^{(l)}) \right),
$$

- Updates features of vertex $i$ using other nodes in its neighbourhood, $\mathcal{N}(i)$
- Node aggregation is done by taking mean
- $\Theta$ and $\Phi$ are linear layer
- Graph readout done by taking mean
Two Operator scenario

- Test a simple scenario with only two four fermion operator

\[ O^{(8)ii33}_{qu} = (\bar{q}_i \gamma_\mu T^A q_i)(\bar{u}_3 \gamma_\mu T^A u_3) , \]
\[ O^{(3)ii33}_{qq} = (\bar{q}_i \gamma_\mu T I q_i)(\bar{q}_3 \gamma_\mu T I q_3) . \]

Normalized \( p_T(b_1) \) distributions at the 13 TeV LHC for the two operators used in the three-class example.
Schematics

Final States
\[ \ell : \{ p_T, \eta, \phi, E, m, \text{PID} \} \]

\[ \bar{\ell} : \{ p_T, \eta, \phi, E, m, \text{PID} \} \]

Compare with actual result, (e.g. for SM event \( y = \{0, 0, 1\} \)) to optimize network parameters.

Network Scores
\[ \left\{ P(\mathcal{O}_{qu}^{(8)\bar{i}33}), P(\mathcal{O}_{qq}^{(3)\bar{i}33}), P(\text{SM}) \right\} \]
Training the GNN

Optimize parameters by minimizing categorical cross-entropy function

\[ \mathcal{L} = - \sum_{i=1}^{N_{\text{classes}}} y_i \log \hat{y}_i. \]

Optimized hyperparameters & architecture to achieve better convergence.
Performance of GNN via ROC curve

- Different architectures and embeddings were compared with Receiver Operating Characteristic curves.
- ROC curves calculated for each operator in a one-vs-rest scheme.
- Additionally calculate curve for overall EFT performance using

\[ P(\text{BSM}) = P(\mathcal{O}_{qq}^{(3)i33}) + P(\mathcal{O}_{qu}^{(8)i33}) = 1 - P(\text{SM}). \]
Probability distribution of the network output

The network essentially separates events into three different regions.
Fit

- Performed $\chi^2$ fit using $p_T$ (b1) distributions at an extrapolated luminosity $3/ab$.
- Cuts on network scores result in improved performance over just the selection cuts.
Fit-alternative

Can also perform fit on flattened two-dimensional histograms from scores.

Example two-dimensional histograms for each contribution, normalised to the cross-section rate.
Contours obtained from 2D Histogram

Cutting on the BSM score provides similar contours with the 2D score histogram approach.

95% CL
3 ab$^{-1}$

Selection cuts
P(BSM) > 0.8
2D score histo
Full Fit constraints with GNN selection

- Extend setup to 13 SMEFT operators relevant to the process.
- ROC curves indicate the capability of the network to distinguish operators.
Baseline analysis

- Performed analysis of **CMS (1610.04191)** for comparison with results based on GNN score cut.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Observable</th>
<th>Binning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{d</td>
<td>y_t^h</td>
<td>}$</td>
</tr>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{d</td>
<td>y_t^l</td>
<td>}$</td>
</tr>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{d</td>
<td>y_t^t</td>
<td>}$</td>
</tr>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{dp_{t,h}^\perp}$</td>
<td>$p_{t,h}^\perp$</td>
<td>[0, 45, 90, 135, 180, 225, 270, 315, 400, 800] GeV</td>
</tr>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{dp_{t,t}^\perp}$</td>
<td>$p_{t,t}^\perp$</td>
<td>[0, 45, 90, 135, 180, 225, 270, 315, 400, 800] GeV</td>
</tr>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{dm_{t,t}}$</td>
<td>$m_{t,t}$</td>
<td>[300, 375, 450, 530, 625, 740, 850, 1100, 2000] GeV</td>
</tr>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{d</td>
<td>y_{t,t}</td>
<td>dm_{t,t}}$</td>
</tr>
<tr>
<td>$\frac{1}{\sigma} \frac{d\sigma}{dp_{t,h}^\perp dm_{t,t}}$</td>
<td>$p_{t,h}^\perp$</td>
<td>[0, 45, 90, 135, 180, 225, 270, 315, 400, 800] GeV</td>
</tr>
</tbody>
</table>

**Table 1:** Distributions included in the fit in this work.
<table>
<thead>
<tr>
<th></th>
<th>2.3 fb$^{-1}$</th>
<th></th>
<th>3 ab$^{-1}$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Individual</td>
<td>Profiled</td>
<td>Individual</td>
<td>Profiled</td>
</tr>
<tr>
<td>$\bar{C}_{G}$</td>
<td>(-0.0543, 0.0535)</td>
<td>(-0.1736, 0.1731)</td>
<td>(-0.0015, 0.0015)</td>
<td>(-0.0046, 0.0046)</td>
</tr>
<tr>
<td>$\bar{C}_{qq}^{(3)33}$</td>
<td>(-0.0317, 0.0326)</td>
<td>(-0.0781, 0.0736)</td>
<td>(-0.0009, 0.0009)</td>
<td>(-0.0021, 0.0021)</td>
</tr>
<tr>
<td>$\bar{C}_{uG}^{33}$</td>
<td>(-0.0253, 0.0247)</td>
<td>(-0.0602, 0.0635)</td>
<td>(-0.0007, 0.0007)</td>
<td>(-0.0017, 0.0017)</td>
</tr>
<tr>
<td>$\bar{C}_{uW}^{33}$</td>
<td>(-0.0234, 0.0228)</td>
<td>(-0.0527, 0.0562)</td>
<td>(-0.0006, 0.0006)</td>
<td>(-0.0015, 0.0015)</td>
</tr>
<tr>
<td>$\bar{C}_{qq}^{(1)33i}$</td>
<td>(-0.0202, 0.0204)</td>
<td>(-0.048, 0.0469)</td>
<td>(-0.0006, 0.0006)</td>
<td>(-0.0013, 0.0013)</td>
</tr>
<tr>
<td>$\bar{C}_{qq}^{(3)33i}$</td>
<td>(-0.0101, 0.0102)</td>
<td>(-0.024, 0.0234)</td>
<td>(-0.0003, 0.0003)</td>
<td>(-0.0007, 0.0007)</td>
</tr>
<tr>
<td>$\bar{C}_{qq}^{(3)ii33}$</td>
<td>(-3.2964, 3.3259)</td>
<td>–</td>
<td>(-0.0917, 0.0917)</td>
<td>(-0.2982, 0.2955)</td>
</tr>
<tr>
<td>$\bar{C}_{qu}^{(8)33ii}$</td>
<td>(-0.0867, 0.0875)</td>
<td>(-0.2063, 0.2015)</td>
<td>(-0.0024, 0.0024)</td>
<td>(-0.0056, 0.0056)</td>
</tr>
<tr>
<td>$\bar{C}_{qu}^{(8)ii33}$</td>
<td>(-0.0577, 0.0583)</td>
<td>(-0.1373, 0.1342)</td>
<td>(-0.0016, 0.0016)</td>
<td>(-0.0038, 0.0038)</td>
</tr>
<tr>
<td>$\bar{C}_{ud}^{(8)33ii}$</td>
<td>(-0.1598, 0.1613)</td>
<td>(-0.3802, 0.3714)</td>
<td>(-0.0044, 0.0044)</td>
<td>(-0.0104, 0.0104)</td>
</tr>
<tr>
<td>$\bar{C}_{uu}^{33i}$</td>
<td>(-0.0225, 0.0228)</td>
<td>(-0.0536, 0.0524)</td>
<td>(-0.0006, 0.0006)</td>
<td>(-0.0015, 0.0015)</td>
</tr>
<tr>
<td>$\bar{C}_{lq}^{(3)ii33}$</td>
<td>–</td>
<td>–</td>
<td>(-0.3289, 0.3288)</td>
<td>(-1.8400, 1.8832)</td>
</tr>
</tbody>
</table>

**Table 2:** Baseline $2\sigma$ bounds for different luminosities with $\bar{C}_i = C_i \frac{v^2}{\Lambda^2}$. 
## Improvements on bounds

<table>
<thead>
<tr>
<th></th>
<th>2.3 fb$^{-1}$</th>
<th>3 ab$^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Individual</td>
<td>Profiled</td>
</tr>
<tr>
<td>$\bar{C}_G$</td>
<td>0.07%</td>
<td>14.53%</td>
</tr>
<tr>
<td>$\bar{C}^{(3)33}_{\ell q}$</td>
<td>33.74%</td>
<td>34.16%</td>
</tr>
<tr>
<td>$\bar{C}^{33}_{uG}$</td>
<td>28.29%</td>
<td>32.12%</td>
</tr>
<tr>
<td>$\bar{C}^{33}_{uW}$</td>
<td>34.86%</td>
<td>35.36%</td>
</tr>
<tr>
<td>$\bar{C}^{(1)33i}_{q q}$</td>
<td>3.50%</td>
<td>3.52%</td>
</tr>
<tr>
<td>$\bar{C}^{(3)33i}_{q q}$</td>
<td>4.35%</td>
<td>4.31%</td>
</tr>
<tr>
<td>$\bar{C}^{(3)3i33}_{q q}$</td>
<td>63.83%</td>
<td></td>
</tr>
<tr>
<td>$\bar{C}^{(8)33i i}_{q q}$</td>
<td>3.45%</td>
<td>3.45%</td>
</tr>
<tr>
<td>$\bar{C}^{(8)i33}_{q q}$</td>
<td>3.74%</td>
<td>3.80%</td>
</tr>
<tr>
<td>$\bar{C}^{(8)33i i}_{q u}$</td>
<td>4.62%</td>
<td>4.63%</td>
</tr>
<tr>
<td>$\bar{C}^{33i i}_{u u}$</td>
<td>3.38%</td>
<td>3.41%</td>
</tr>
<tr>
<td>$\bar{C}^{(3)i33}_{i q}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 3:** Maximum improvements in 2$\sigma$ bounds via a cut on the ML score.
Comments on the results

- GNN performs well in discriminating non-resonant top decay contributions.
- Sizeable improvement when momentum enhancement is present.
- Operators with small improvements are relatively under control via the inclusive rate and baseline selection.
- Improvements on profiled bounds can be greater than individual ones since a cut on the EFT score can select a region where the impact of other operators is reduced.
- Improvements should generalise to $\Lambda^{-4}$ terms of cross-section expansion.
Conclusion

● Highly non-trivial task to design representation/algorithms which would achieve optimal knowledge of the background

● Graphs are an efficient way to represent jets, with the ability to incorporate relational (via edge-features) information between constituents

● Graph autoencoders can learn both local as well as global features (via edge-reconstruction) of QCD jets thereby making it a “promising candidate”

● Shown to be robust to complexity bias with the added benefit of an efficient representation with no inherent Euclidean bias

● Integration of physics-knowledge very much important to achieve the goal of learning non-trivial topologies of collider events

Thank you
Back up
Applications to HEFT:

- Quartic Gauge-Higgs couplings:

Feynman diagrams contributing to WBF-production of di-Higgs
Fully connected Graph: constraining the $\kappa_{2V}$
Latent graph-representation

\[
\tilde{f}^a = \frac{1}{N} \sum_{i \in G} f_i^a
\]
Latent graph-representation
Jet dataset details

$\sqrt{s} = 13$ TeV pp collisions, MadGraph5

**QCD jets** (Training and validation): dijet events

**Signal benchmarks** (Testing):

(i) boosted hadronically-decaying $W$ bosons
(ii) boosted hadronically-decaying top quarks
(iii) a boosted scalar $\phi$ decaying as $\phi \rightarrow W^+ W^- \rightarrow 4j$, with $m_{\phi} = 700$ GeV

Jets definition and cuts:

- anti-$k_t$ algorithm with $R = 1.5$ with FastJet
- use final state particles after showering and hadronization with Pythia8

- Require $|y| < 2.5$ and $p_T > 1$ TeV
- Select hardest $p_T$ jet from each event.

Final input for graph construction: reclustering with anti-$k_T$ jet algorithm into microjets with $R = 0.1$ and $p_T \geq 5$ GeV
Feature distribution of Microjets
Feature distribution of Microjets
Message-passing:

$$ab m_{ij}^{(1)} = ab F_w(\bar{e}_{ij}) \times ab \tilde{h}_j^{(0)},$$

$ab \tilde{h}_j^{(0)}$ is formed by repeating $\tilde{h}_j^{(0)}$ $n$ times.
$F_w = \text{Edge-function (a Neural network)}$

Node-readout: Takes the mean of $ab m_{ij}^{(1)}$ over all neighbouring nodes $j$, and then sums over the $a$ index of the matrix:

$$b h_i^{(1)} = \sum_a \text{mean}_{j \in N(i)} \left( \left\{ ab m_{ij}^{(1)} \right\} \right),$$
Message-passing:

$$\tilde{m}_{ij}^{(l)} = \Theta_w (\tilde{h}_j^{(l)} - \tilde{h}_i^{(l)}) + \Phi_w \bar{h}_i^{(l)},$$

$\Theta_w$ and $\Phi_w$ are weights.

Node-readout:

$$a_h_i^{(l+1)} = \max_{j \in \mathcal{N}(i)} \{a_m_{ij}^{(l)}\},$$