Quantum-Classical Graph Neural Networks & Jet Tagging

Haemanth Velmurugan, Roy Forestano, Sergei Gleyzer,
Kyoungchul Kong, Konstantin Matchev, Katia Matcheva

Introduction & Motivation

- Quantum machine learning - rapidly advancing due to possibilities for high speedups and low resource usage
- Multiple studies on QML for HEP applications – particle track reconstruction, jet generation, etc.
- Jet tagging – especially important at CERN to efficiently identify useful data and process it in real time
- Graph networks are well suited for jet tagging due to the sparse, irregular and heterogenous nature of the data

Outline

Brief about GNN

- GNN working
- Cost Analysis
- Graph Attention

Quantum GNN Proposal

Experiments and Results

- Dataset
- Model Architecture
- Results

Brief about GNN (GCN)

GCN - Three basic steps

- Learning Node Embedding
- Neighborhood aggregation

→ Graph Convolution Layer

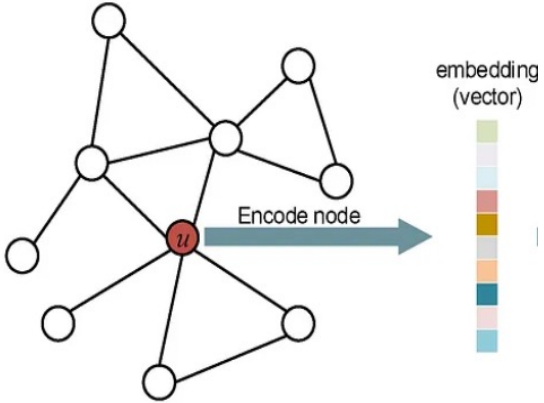
- Classification/Prediction
  - Node level
  - Edge level
  - Graph level

01

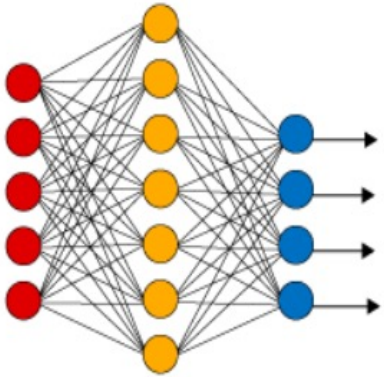
Node Updation/ Transformation

Learn better node embeddings using Neural Network

Input size = D



Source



Source

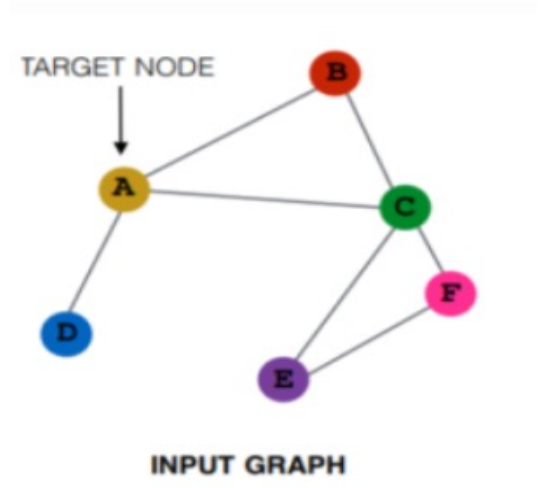
AAA Lateef, Sufyan Al-Janabi, and Belal Al-Khateeb. "Survey on intrusion detection systems based on deep learning." Periodicals of Engineering and Natural Sciences 7, no. 3 (2019).
Hussein, Omar. "Graph Neural Networks Series: Part 3: Node Embedding." Medium, May 16, 2023. https://medium.com/the-modern-scientist/graph-neural-networks-series-part-3-node-embedding-36613cc967d5.

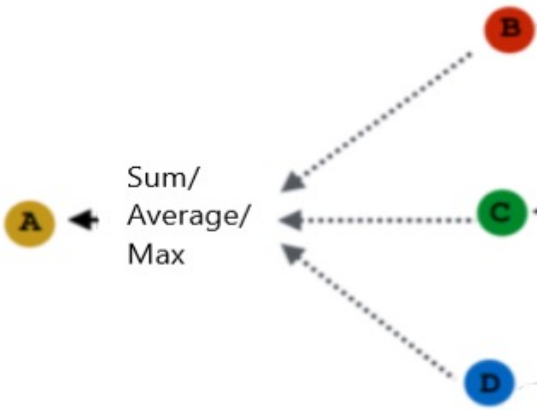
02

Neighborhood Aggregation

Apply a function (sum/max/min) to the embeddings of neighbors for every node in graph

Scales as O(ND)





Source

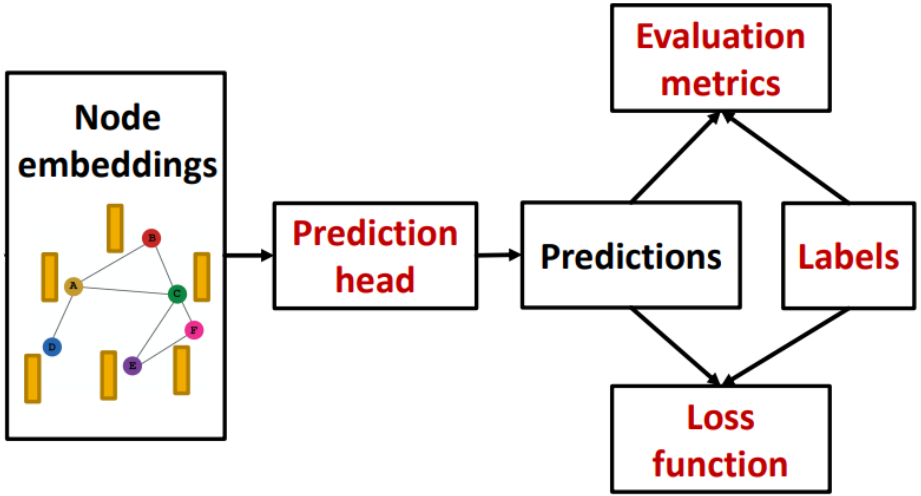
Jure Leskovec, "Stanford CS224W: Graph Neural Networks", Slides at Stanford University, Stanford, CA, January 10, 2024

03

Classification/ Prediction

Neural Network - predicts graph label from learned embeddings

Aggregate node embeddings to get graph embedding



Source

Prediction head :

- Global Max/Average over all vertices in Graph
- Represents Graph Embedding

Jure Leskovec, "Stanford CS224W: Graph Neural Networks", Slides at Stanford University, Stanford, CA, January 10, 2024

GCN - Analysis (Graph classification)

For a graph with N nodes and D features for each node

- Node Embedding Updation - input size is **D**
- Neighborhood aggregation - input size is **O(ND)**
- Classification/Prediction - input size is **D**

Aggregation/Message passing using quantum circuit is not scalable (NISQ)

Can we also learn the jet graph structure dynamically through attention?

Graph Attention

Assign importance to neighbors during aggregation – learnable

$$\sum_{j \in N_i} W^k h_j^{k-1} \rightarrow \sum_{j \in N_i} \boldsymbol{\alpha_{ij}^k} W^k h_j^{k-1}$$

$$\alpha_{ij}^k = softmax_j\left(A^k \left[W^k h_i^{k-1}, W^{kh} j^{k-1}\right]\right)$$

Attention parameters α_{ij} - learnable using quantum neural network with **2*D** qubits

Quantum GNN Proposals

GCN Pseudocode

$Pseudocode$:

$for\ every\ graph$:

    $for\ every\ node$:

        $features \leftarrow NN(features)$

        $features \leftarrow aggregate\ over\ neighbours$

    $Graph\ embedding \leftarrow aggregate\ all\ node\ features$

    $prediction \leftarrow NN(Graph\ embedding)$

Graph Convolution layer

Proposed Approach

Pseudocode:

for every jet:

for every particle:

features←*NN*(*features*) → Quantum

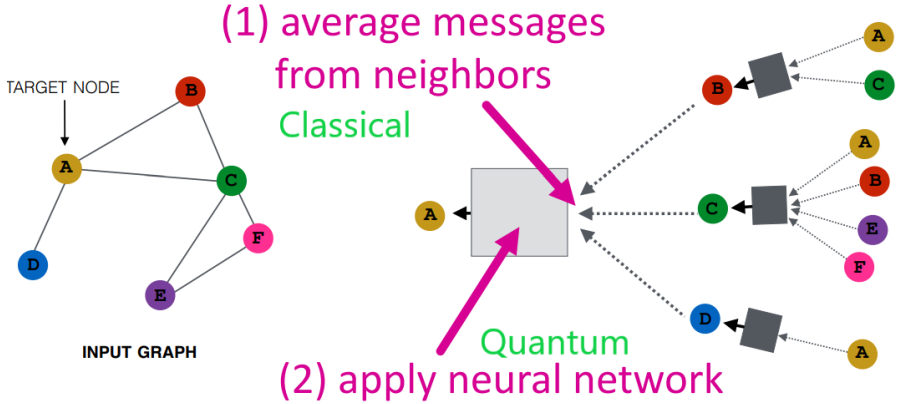
features←*aggregate over neighbours* → Classical

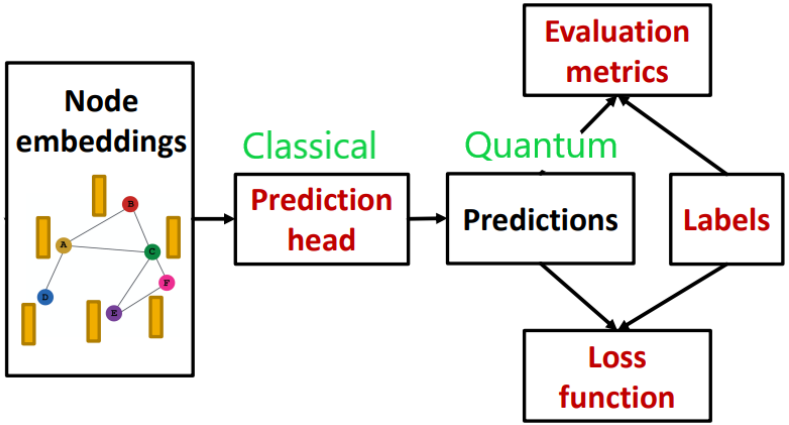
Graph embedding←*aggregate all node features* → Classical

prediction←*NN*(*Graph embedding*) → Quantum

Quantum GNN with Classical Message Passing

- Quantum Node Updation – D qubits
- Classical Aggregation
- Quantum Classification – D qubits





Jure Leskovec, "Stanford CS224W: Graph Neural Networks", Slides at Stanford University, Stanford, CA, January 10, 2024

Pros of the Proposed Approach

- Scalable – only D qubits per QNN, the number of node features
  - Scales well for larger graph sizes – just run the circuit more times

- Splitting up operations using multiple QNNs - allows applying non-linear activations to the intermediate QNN outputs.

- No classical NN layers required (for reasonable feature vector size D)

Experiments and Results

Dataset and pre-processing

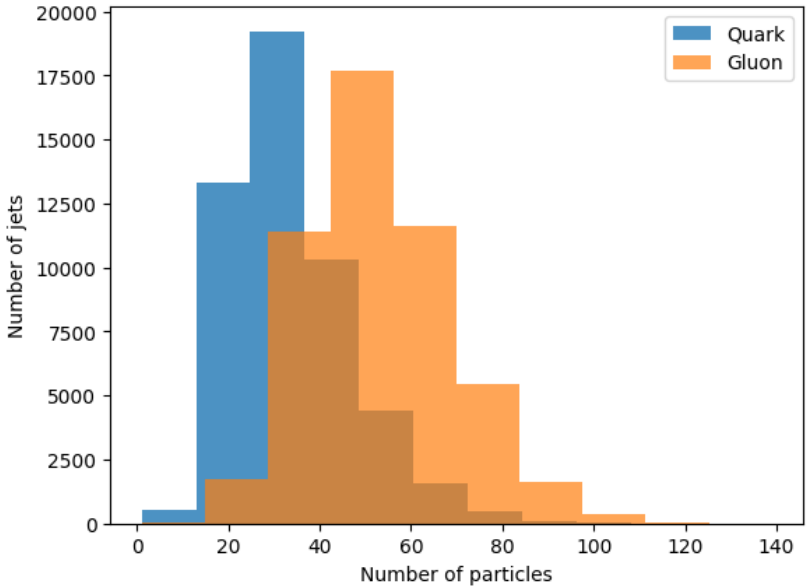
- Pythia8 Quark and Gluon jet dataset [3]
- Select top 3 particles per jet (ordered by transverse momentum)
- Extract kinematic features - transverse mass, energy and cartesian momentum using Particle package
- Remove 0-padded particles and normalize the data
- Construct Graphs – edges determined by Euclidean distance metric in the (rapidity, azimuthal angle) plane

With just 200 trainable parameters!

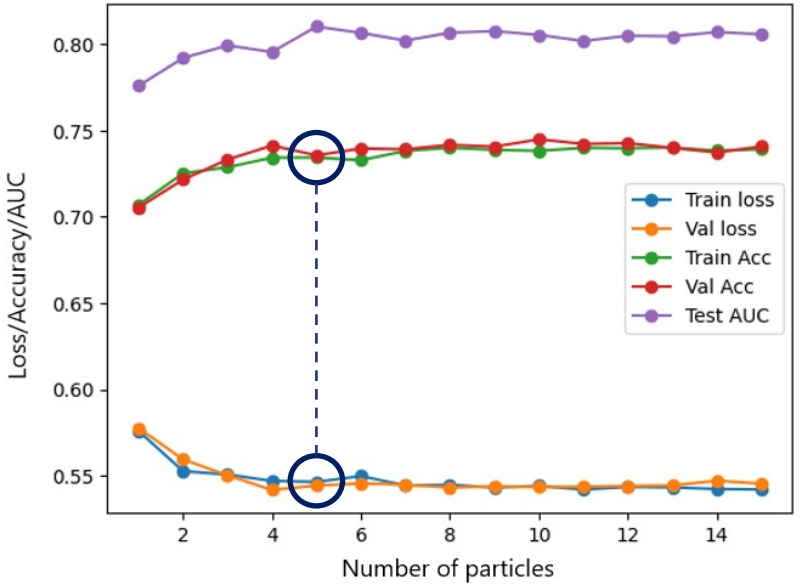
Results

S.No	Model	Train loss	Val. loss	Train ACC	Val. ACC	Test AUC
1.	Classical GCN	0.540	0.543	74	74	79.5
2.	QCGCN (classical classifier)	0.550	0.559	73	73	79.0
3.	QCGCN (MPS classifier)	0.549	0.554	73	73	79.0
4.	QCGCN (TTN classifier)	0.554	0.555	73	72	79.0

Data Analysis



Distribution of observed number of particles



Number of particles/jet vs model performance

Results contd.

AUC consistent with using all particles!

S.No	Model	Train loss	Val loss	Train ACC	Val ACC	Test AUC
1.	All particles	0.480	0.478	78	78	85
2.	5 particles per jet	0.527	0.535	74	74	80
3.	5 particles per jet + #particles observed	0.4822	0.496	77	78	85

Not much improvement with attention ☹

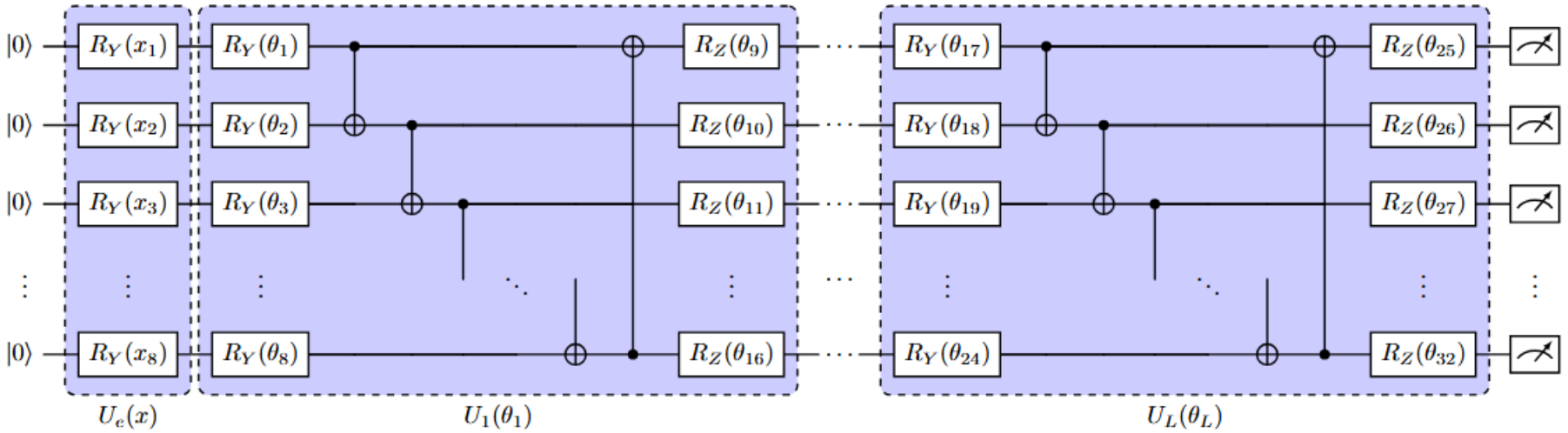
Results contd.

S.No	Model	Train loss	Val loss	Train ACC	Val ACC	Test AUC
1.	GAT	0.487	0.49	78	77	85.3
2.	QCGCN with classical attention	0.494	0.503	76	77	84.0
3.	QCGAT	0.502	0.508	77	77	84.2

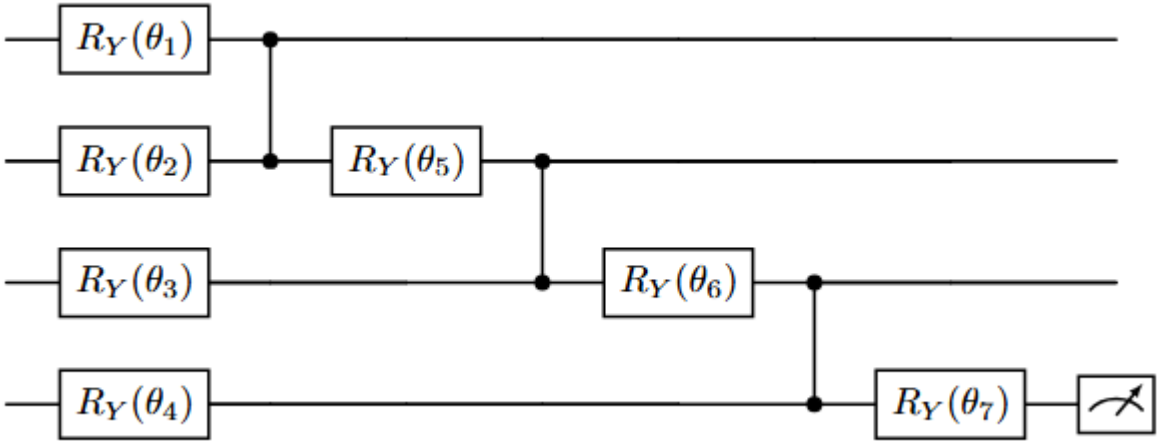
Summary

- Designed and developed Quantum-classical GCN for jet tagging
- Quantum models achieved similar performance as GCN with fewer parameters
- Adding #features to graph embedding improved AUC by 5% - consistent with using all particles (150 particles as opposed to 5!)
- Naïve adoption of Graph attention, though promising, did not lead to improvements in performance **To be explored!**

QNN - Node Embedding Learner

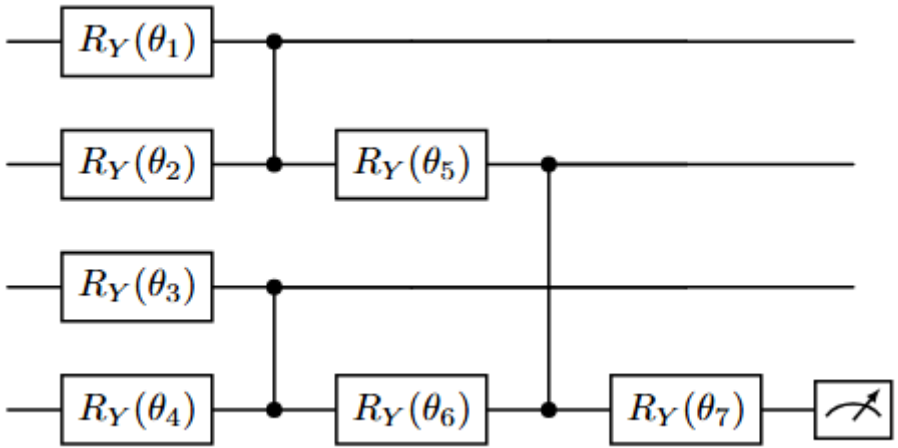


QNN – Classifier



Matrix Product State (MPS) Classifier

QNN – Classifier



Tensor Tree Network (TTN) Classifier

References

1. Tüysüz, Cenk, Carla Rieger, Kristiane Novotny, Bilge Demirköz, Daniel Dobos, Karolos Potamianos, Sofia Vallecorsa, Jean-Roch Vlimant, and Richard Forster. "Hybrid quantum classical graph neural networks for particle track reconstruction." Quantum Machine Intelligence 3 (2021): 1-20.
2. Velickovic, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. "Graph attention networks." stat 1050, no. 20 (2017): 10-48550.
3. P. T. Komiske, E. M. Metodiev, J. Thaler, Energy Flow Networks: Deep Sets for Particle Jets, JHEP 01 (2019) 121, arXiv:1810.05165
4. Rodrigues, E.; Schreiner, H. Scikit-Hep/Particle: Version 0.23.0; Zenodo: Geneva, Switzerland, 2023.