Quantum Invariant and Equivariant Graph Neural Networks for HEP Analysis

Google Summer of Code (GSOC) Contributor under the Machine Learning for Science (ML4SCI) Organization

Roy T. Forestano

ML4SCI Quantum Machine Learning for HEP (QMLHEP) Group
University of Florida
Department of Physics

October 23, 2023

Overview

Dataset [7]

Graphically Structured Data $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$

$$\begin{matrix} (\alpha,1,1) & (\alpha,2,1) & \cdots & (\alpha,m_\alpha,1) \\ (\alpha,1,2) & (\alpha,2,2) & \cdots & (\alpha,m_\alpha,2) \\ \vdots & \vdots & \ddots & \vdots \\ & & & (\alpha,m_\alpha,l) \end{matrix}$$

$$\begin{matrix} (2,1,1) & (2,2,1) & \cdots & (2,m_2,1) \\ (2,1,2) & (2,2,2) & \cdots & (2,m_2,2) \\ \vdots & \vdots & \ddots & \vdots \\ & & & (2,m_2,l) \end{matrix}$$

$$\underbrace{\begin{matrix} (1,1,1) & (1,2,1) & \cdots & (1,m_1,1) \\ (1,1,2) & (1,2,2) & \cdots & (1,m_1,2) \\ \vdots & \vdots & \ddots & \vdots \\ (1,1,l) & (1,2,l) & \cdots & (1,m_1,l) \end{matrix}}$$

(Jet (*n*), Multiplicity (*m*), Feature (*l*))

Dataset [7]

Graphically Structured Data $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$

$$\begin{matrix} (\alpha,1,1) & (\alpha,2,1) & \cdots & (\alpha,m_\alpha,1) \\ (\alpha,1,2) & (\alpha,2,2) & \cdots & (\alpha,m_\alpha,2) \\ \vdots & \vdots & \ddots & \vdots \\ & & & (\alpha,m_\alpha,l) \end{matrix}$$

$$\begin{matrix} (2,1,1) & (2,2,1) & \cdots & (2,m_2,1) \\ (2,1,2) & (2,2,2) & \cdots & (2,m_2,2) \\ \vdots & \vdots & \ddots & \vdots \\ & & & (2,m_2,l) \end{matrix}$$

$$\underbrace{\begin{matrix} (1,1,1) & (1,2,1) & \cdots & (1,m_1,1) \\ (1,1,2) & (1,2,2) & \cdots & (1,m_1,2) \\ \vdots & \vdots & \ddots & \vdots \\ (1,1,l) & (1,2,l) & \cdots & (1,m_1,l) \end{matrix}}$$

(Jet (n), Multiplicity (m), Feature (l))

Multiplicity $(m) \equiv$ Nodes with Features (l)

$$\boxed{x_\alpha^{(il)} \in \{p_T, \eta, \phi, m_p\}} \quad (1)$$

Dataset [7]

Graphically Structured Data $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$

$$\begin{matrix} (\alpha,\mathbf{1},\mathbf{1}) & (\alpha,\mathbf{2},\mathbf{1}) & \cdots & (\alpha,\mathbf{m}_\alpha,\mathbf{1}) \\ (\alpha,\mathbf{1},\mathbf{2}) & (\alpha,\mathbf{2},\mathbf{2}) & \cdots & (\alpha,\mathbf{m}_\alpha,\mathbf{2}) \\ \vdots & \vdots & \ddots & \vdots \\ & & & (\alpha,\mathbf{m}_\alpha,\mathbf{l}) \end{matrix}$$

$$\begin{matrix} (\mathbf{2},\mathbf{1},\mathbf{1}) & (\mathbf{2},\mathbf{2},\mathbf{1}) & \cdots & (\mathbf{2},\mathbf{m}_2,\mathbf{1}) \\ (\mathbf{2},\mathbf{1},\mathbf{2}) & (\mathbf{2},\mathbf{2},\mathbf{2}) & \cdots & (\mathbf{2},\mathbf{m}_2,\mathbf{2}) \\ \vdots & \vdots & \ddots & \vdots \\ & & & (\mathbf{2},\mathbf{m}_2,\mathbf{l}) \end{matrix}$$

$$\underbrace{\begin{matrix} (\mathbf{1},\mathbf{1},\mathbf{1}) & (\mathbf{1},\mathbf{2},\mathbf{1}) & \cdots & (\mathbf{1},\mathbf{m}_1,\mathbf{1}) \\ (\mathbf{1},\mathbf{1},\mathbf{2}) & (\mathbf{1},\mathbf{2},\mathbf{2}) & \cdots & (\mathbf{1},\mathbf{m}_1,\mathbf{2}) \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{1},\mathbf{1},\mathbf{l}) & (\mathbf{1},\mathbf{2},\mathbf{l}) & \cdots & (\mathbf{1},\mathbf{m}_1,\mathbf{l}) \end{matrix}}$$

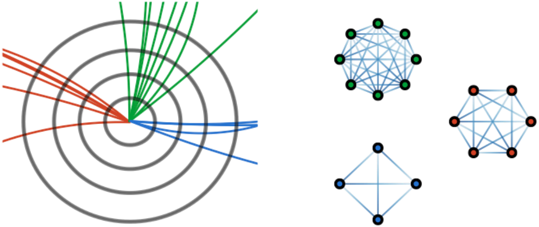
(Jet (n), Multiplicity (m), Feature (l))

Multiplicity $(m) \equiv$ **Nodes with Features (l)**

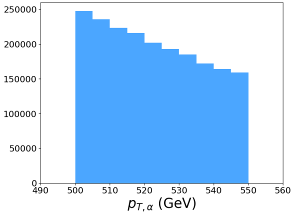
$$\boxed{x_\alpha^{(il)} \in \{p_T, \eta, \phi, m_p\}} \qquad (1)$$

Jet $(n) \equiv$ **Graph with Labels (not shown)**

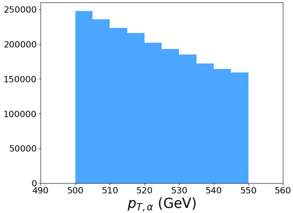
$$\boxed{y_n \in \{0,1\} \text{ for Binary Classification}} \qquad (2)$$

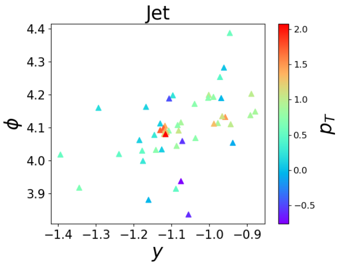


Data Distributions & Feature Engineering @ ML4SCI

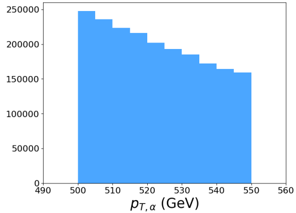


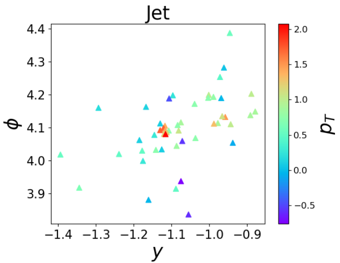
Data Distributions & Feature Engineering @ ML4SCI





Data Distributions & Feature Engineering @ ML4SCI

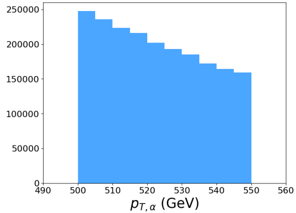


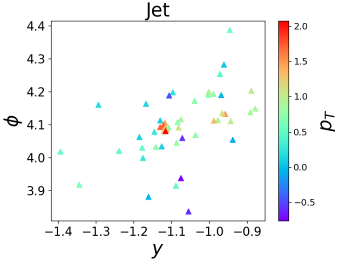


Feature set $h_\alpha^{(il)}$ with $l = 0, 1, 2, \ldots, 7$:

$$
\begin{aligned}
h_\alpha^{(il)} \equiv \{ & p_{T,\alpha}^{(i)}, y_\alpha^{(i)}, \phi_\alpha^{(i)}, \\
& m_{T,\alpha}^{(i)} = \sqrt{m_\alpha^{(i)2} + p_{T,\alpha}^{(i)2}}, \\
& E_\alpha^{(i)} = m_{T,\alpha}^{(i)} \cosh(y_\alpha^{(i)}), \\
& p_{x,\alpha}^{(i)} = p_{T,\alpha}^{(i)} \cos(\phi_\alpha^{(i)}), \\
& p_{y,\alpha}^{(i)} = p_{T,\alpha}^{(i)} \sin(\phi_\alpha^{(i)}), \\
& p_{z,\alpha}^{(i)} = m_{T,ij} \sinh(y_\alpha^{(i)}) \}
\end{aligned}
$$

Data Distributions & Feature Engineering @ ML4SCI





Feature set $h_\alpha^{(il)}$ with $l = 0, 1, 2, \ldots, 7$:

$$h_\alpha^{(il)} \equiv \{p_{T,\alpha}^{(i)}, y_\alpha^{(i)}, \phi_\alpha^{(i)},$$

$$m_{T,\alpha}^{(i)} = \sqrt{m_\alpha^{(i)2} + p_{T,\alpha}^{(i)2}},$$

$$E_\alpha^{(i)} = m_{T,\alpha}^{(i)} \cosh(y_\alpha^{(i)}),$$

$$p_{x,\alpha}^{(i)} = p_{T,\alpha}^{(i)} \cos(\phi_\alpha^{(i)}),$$

$$p_{y,\alpha}^{(i)} = p_{T,\alpha}^{(i)} \sin(\phi_\alpha^{(i)}),$$

$$p_{z,\alpha}^{(i)} = m_{T,ij} \sinh(y_\alpha^{(i)})\}$$

Edge Connections a_{ij}

$$\Delta R_\alpha^{(ij)} = \sqrt{\left(\phi_\alpha^{(i)} - \phi_\alpha^{(j)}\right)^2 + \left(y_\alpha^{(i)} - y_\alpha^{(j)}\right)^2}$$

Training, Validation, and Testing

- Only consider jets with at leat 10 particles $\Longrightarrow$ $N = 2$ million to $1,997,445$ jets.

Training, Validation, and Testing

- Only consider jets with at leat 10 particles $\implies N = 2$ million to $1,997,445$ jets.
- Sort the particles per jet by largest to smallest transverse momentum p_T.

Training, Validation, and Testing

- Only consider jets with at leat 10 particles $\implies N = 2$ million to $1,997,445$ jets.
- Sort the particles per jet by largest to smallest transverse momentum p_T.
- Quantum entangled state scales as $2^n \implies$ choose $n_\alpha = 3$ with $\mathbf{h}_\alpha \in \mathbb{R}^{3\times 8}$

Training, Validation, and Testing

- Only consider jets with at leat 10 particles $\implies N = 2$ million to $1,997,445$ jets.
- Sort the particles per jet by largest to smallest transverse momentum p_T.
- Quantum entangled state scales as $2^n \implies$ choose $n_\alpha = 3$ with $\mathbf{h}_\alpha \in \mathbb{R}^{3\times 8}$
- Choose a subset to test the models: $N = 12,500$

Training, Validation, and Testing

- Only consider jets with at leat 10 particles $\implies N = 2$ million to $1,997,445$ jets.
- Sort the particles per jet by largest to smallest transverse momentum p_T.
- Quantum entangled state scales as $2^n \implies$ choose $n_\alpha = 3$ with $\mathbf{h}_\alpha \in \mathbb{R}^{3\times 8}$
- Choose a subset to test the models: $N = 12,500$

Data Splits

	N	$N_{quark:1}$	$N_{gluon:0}(0)$
Training	$10,000$	$4,982$	$5,018$
Validation	$1,250$	658	592
Testing	$1,250$	583	667

Invariance and Equivariance

Definition

A function $\varphi : X \to Y$ is **equivariant** with respect to a set of group transformations $T_g : X \to X$, $g \in G$, acting on the input vector space X, if there exists a set of transformations $S_g : Y \to Y$ which similarly transform the output space Y, i.e.

$$\varphi(T_g x) = S_g \varphi(x). \tag{3}$$

A function is said to be **invariant** when for all $g \in G$, S_g becomes the set containing only the trivial mapping, i.e. $S_g = \{\mathbb{I}_G\}$, where $\mathbb{I}_G \in G$ is the identity element of the group G .

Invariance vs. Equivariance [6, 5]

Invariance

Invariance

A function φ is invairant with respect to a group G transformation $g \in T_g \subset G$ if

$$\varphi(g \cdot x) = \varphi(x) \tag{4}$$

best for

$\downarrow$

Invariance vs. Equivariance [6, 5]

Invariance

Invariance

A function φ is invairant with respect to a group G transformation $g \in T_g \subset G$ if

$$\varphi(g \cdot x) = \varphi(x) \tag{4}$$

best for

$\downarrow$

Input Embedding

Classical	**Quantum**
$\mathbf{m}_{ij}(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}) \rightarrow$	$\mathcal{U}_{ij}(x_i, x_j) \rightarrow$
$\mathbf{m}_{ij}(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}, \boxed{\lvert\mathbf{x}_i - \mathbf{x_j}\rvert})$	$\mathcal{U}_{ij}(\boxed{\lvert\mathbf{x}_i - \mathbf{x_j}\rvert})$

Invariance vs. Equivariance [6, 5]

Invariance

Invariance

A function φ is invairant with respect to a group G transformation $g \in T_g \subset G$ if

$$\varphi(g \cdot x) = \varphi(x) \tag{4}$$

best for

$\downarrow$

Input Embedding

Classical	Quantum
$\mathbf{m}_{ij}(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}) \rightarrow$	$\mathcal{U}_{ij}(x_i, x_j) \rightarrow$
$\mathbf{m}_{ij}(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}, \boxed{\lvert \mathbf{x}_i - \mathbf{x}_j \rvert})$	$\mathcal{U}_{ij}(\boxed{\lvert \mathbf{x}_i - \mathbf{x}_j \rvert})$

Equivariance

Equivariance

A function φ is equivariant with respect to group G, G' transformations $g \in T_g \subset G, g' \in S_g \subset G'$ if

$$\varphi(g \cdot x) = g' \cdot \varphi(x) \tag{5}$$

best for

$\downarrow$

Invariance vs. Equivariance [6, 5]

@ ML4SCI

Invariance

Invariance

A function φ is invairant with respect to a group G transformation $g \in T_g \subset G$ if

$$\varphi(g \cdot x) = \varphi(x) \tag{4}$$

best for

$\downarrow$

Input Embedding

Classical

$\mathbf{m}_{ij}(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}) \rightarrow$

$\mathbf{m}_{ij}(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}, \boxed{|\mathbf{x}_i - \mathbf{x_j}|})$

Quantum

$\mathcal{U}_{ij}(x_i, x_j) \rightarrow$

$\mathcal{U}_{ij}(\boxed{|\mathbf{x}_i - \mathbf{x_j}|})$

Equivariance

Equivariance

A function φ is equivariant with respect to group G, G' transformations $g \in T_g \subset G, g' \in S_g \subset G'$ if

$$\varphi(g \cdot x) = g' \cdot \varphi(x) \tag{5}$$

best for

$\downarrow$

Layer Structure

Classical

$gx_i^l \rightarrow$

$gx_i^l + C\sum_{j \neq i}(gx_i^l - gx_j^l)\phi_x(\mathbf{m}_{ij})$

Quantum

$\mathcal{U}(gx) \rightarrow$

$\mathcal{U}_g \mathcal{U}(x) \mathcal{U}_g^\dagger$

Equivariant coordinate update function @ ML4SCI

Proposition

Let $T_g : X \to X$ be the set of translational and rotational group transformations with elements $g \in T_g \subset G$ which act on the vector space X. The function $\varphi : X \to X$ defined by

$$\varphi(x) = x_i + C \sum_{j \neq i} (x_i - x_j) \tag{6}$$

is equivariant with respect to T_g.

Equivariant coordinate update function

Proposition

Let $T_g : X \to X$ be the set of translational and rotational group transformations with elements $g \in T_g \subset G$ which act on the vector space X. The function $\varphi : X \to X$ defined by

$$\varphi(x) = x_i + C\sum_{j \neq i}(x_i - x_j) \tag{6}$$

is equivariant with respect to T_g.

Proof

Let a general transformation $g \in T_g$ of this form act on X by $gX = RX + T$,

$$\begin{aligned}
\varphi(gx) &= (gx_i) + C\sum_{j \neq i}(gx_i - gx_j) \\
&= (Qx_i + T) + C\sum_{j \neq i}(Qx_i + T - Qx_j - T) \\
&= Qx_i + C\sum_{j \neq i}Q(x_i - x_j) + T \\
&= Q[x_i + C\sum_{j \neq i}(x_i - x_j)] + T \\
&= g\varphi(x),
\end{aligned}$$

where $\varphi(gx) = g\varphi(x)$ shows φ transforms equivariantly under transformations $g \in T_g$ acting on X.

Graph Neural Network Theory [3, 2, 6]

@

Graph Neural Network (GNN)

Message Passing GCNN Layer

Nodes $v_i \in \mathcal{V}$, edges $e_{ij} \in \mathcal{E}$

$$\mathbf{m}_{ij} = \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}) \tag{7}$$

$$\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij} \tag{8}$$

$$\mathbf{h}_i^{l+1} = \phi_h(\mathbf{h}_i^l, \mathbf{m}_i) \tag{9}$$

where $\mathbf{h}_i$ are node features, a_{ij} are edge attributes, $\mathcal{N}(i)$ is the set of neighbors of node v_i, and ϕ_e, ϕ_h are edge and node operations typically approximated using Multiplayer Perceptrons (MLPs) (Kipf & Welling 2016, Gilmer et al. 2017).

Graph Neural Network Theory [3, 2, 6]

@

Graph Neural Network (GNN)

Message Passing GCNN Layer

Nodes $v_i \in \mathcal{V}$, edges $e_{ij} \in \mathcal{E}$

$$\mathbf{m}_{ij} = \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}) \tag{7}$$

$$\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij} \tag{8}$$

$$\mathbf{h}_i^{l+1} = \phi_h(\mathbf{h}_i^l, \mathbf{m}_i) \tag{9}$$

where $\mathbf{h}_i$ are node features, a_{ij} are edge attributes, $\mathcal{N}(i)$ is the set of neighbors of node v_i, and ϕ_e, ϕ_h are edge and node operations typically approximated using Multiplayer Perceptrons (MLPs) (Kipf & Welling 2016, Gilmer et al. 2017).

SE(2) Equivariant GNN (EGNN)

Message Passing EGCNN Layer

Nodes $v_i \in \mathcal{V}$, edges $e_{ij} \in \mathcal{E}$

$$\mathbf{m}_{ij} = \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}, |\mathbf{x}_i^l - \mathbf{x}_j^l|) \tag{10}$$

$$\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij} \tag{11}$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + C \sum_{j \neq i} (\mathbf{x}_i^l - \mathbf{x}_j^l) \phi_x(\mathbf{m}_{ij}) \tag{12}$$

$$\mathbf{h}_i^{l+1} = \phi_h(\mathbf{h}_i^l, \mathbf{m}_i) \tag{13}$$

where we update the coordinates via x_i^{l+1}, include the invariant distance $|\mathbf{x}_i^l - \mathbf{x}_j^l|$ in ϕ_e, and ϕ_x is the coordinate MLP. The equivariant regularization parameter $C(n) < 1$.

Quantum GNN Theory [8]

Quantum GNN (QGNN)

QGCNN Layer

Nodes $v_i \in \mathcal{V}$, edges $e_{ij} \in \mathcal{E}$

$$H(\mathcal{W}_{ij}, \mathcal{M}_i, \mathcal{Q}_0) = \underbrace{\sum_{(i,j)\in\mathcal{E}} \mathcal{W}_{ij}\sigma_i^z\sigma_j^z}_{\substack{\text{Interactions}\\ \text{Between}\\ \text{Nodes (qubits)}}} + \underbrace{\sum_i \mathcal{M}_i\sigma_i^z}_{\substack{\text{Node}\\ \text{(qubit) weights}}} + \mathcal{Q}_0 \underbrace{\sum_i \sigma_i^x}_{\textit{Transverse}} \tag{14}$$

$$U_{ij} = \phi_u(\mathcal{W}_{ij}, \mathcal{M}_i, \mathcal{Q}_0) = e^{-i\sum_{q=1}^{Q} \gamma_{lq} H_q(\mathcal{W}_{ij}, \mathcal{M}_i, \mathcal{Q}_0)}, \tag{15}$$

$$|\psi^{l+1}\rangle = \phi_{|\psi\rangle}(|\psi^l\rangle, U_{ij}) = U_\theta^l U_{ij} U_\theta^{l\dagger} |\psi^l\rangle \tag{16}$$

where $\mathcal{W}$, $\mathcal{M}$, $\mathcal{Q}$ are pre-determined or learned, γ_{lq} is a learnable infinitesimal parameter, $|\psi_i^l(\mathbf{h}_i)\rangle$ is the quantum state at layer l, and U_θ^l is a trainable unitary matrix. Applying this transformation Q times will correspond to running our network over P layers, thus, building up a full trainable unitary parameter transformation (Verdon et al. 2019).

Equivariant Quantum GNN Theory [4]

Equivariant Quantum GNN (EQGNN)

EQGCNN Layer

Nodes $v_i \in \mathcal{V}$, edges $e_{ij} \in \mathcal{E}$

$$H(A_{ij}, \mathcal{M}_i, \mathcal{Q}_0) = \underbrace{\sum_{(i,j)\in\mathcal{E}} A_{ij}\sigma_i^z\sigma_j^z}_{\text{Interactions Between Nodes (qubits)}} + \underbrace{\sum_i \mathcal{M}_i\sigma_i^z}_{\text{Node (qubit) weights}} + \mathcal{Q}_0 \underbrace{\sum_i \sigma_i^x}_{\textit{Transverse}} \tag{17}$$

$$U_{ij} = \phi_u(A_{ij}, \mathcal{M}_i, \mathcal{Q}_0) = e^{-i\sum_{q=1}^{Q}\gamma_{lq}H_q(A_{ij},\mathcal{M}_i,\mathcal{Q}_0)}, \tag{18}$$

$$|\psi^{l+1}\rangle = \phi_{|\psi\rangle}(|\psi^l\rangle, U_{ij}) = U_\theta^l U_{ij} U_\theta^{l\dagger} |\psi^l\rangle \tag{19}$$

$$U_{ij}(\mathcal{U}_g A_{ij} \mathcal{U}_g^\dagger) = \tilde{\mathcal{U}}_{g'} U_{ij}(A_{ij}) \tilde{\mathcal{U}}_{g'}^\dagger \implies U_{ij}(A_{ij}) = \tilde{\mathcal{U}}_{g'}^\dagger U_{ij}(\mathcal{U}_g A_{ij} \mathcal{U}_g^\dagger) \tilde{\mathcal{U}}_{g'} \tag{20}$$

where we restrict the trainable interaction matrix to the adjacency matrix of the graph, i.e. $\mathcal{W}_{ij} \to A_{ij}$, such that $\mathcal{U}_g \in \mathbb{C}^{n\times n}$ and $\tilde{\mathcal{U}}_{g'} \in \mathbb{C}^{2^n\times 2^n}$ are different representations of group $T_g \cong S_g$ elements $\mathcal{U}_g \in T_g, \tilde{\mathcal{U}}_{g'} \in S_g$.

A Note on Permutation Equivariance

Fact

A GNN is permutation equivariant with respect to the sum of the graphically transformed node features corresponding to each graph. This can be written as a map $\varphi : V^{m\times n} \to V^n$ which takes the node feature matrix of each graph to a single feature vector such that $\varphi(\mathbf{h}^P) = \sum_i \mathbf{h}_i^P$.

Proposition

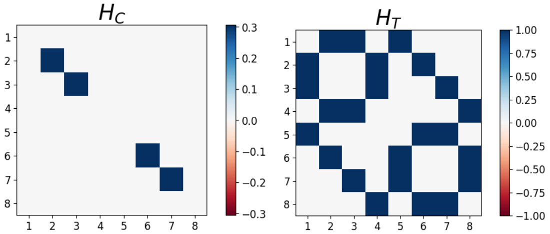
For V a commutable vector space, the product state $\bigotimes_{i=1}^m \mathbf{v}_i : V^n \times \cdots \times V^n \to V^{n^m}$ is permutation equivariant with respect to the sum of its entries. We prove the $n = 2$ case for all $m \in \mathbb{Z}_{>0}$.

Note: Here, m is the number of nodes per graph and n is the number of features.

Hamiltonian and its General Linear Form

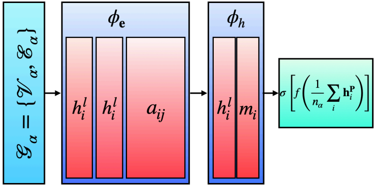
With inspiration from quantum unconstrained binary optimization (QUBO) problems which utilize Ising Hamiltonians, we choose a Hamiltonian which best exploits the properties of a graph by mapping the classical scalar form to the quantum operator form

$$H(a_{ij}) = \underbrace{\sum_{(i,j)\in\mathcal{E}} a_{ij} \left(\frac{\hat{\mathbb{I}}_i - \sigma_i^z}{2} - \frac{\hat{\mathbb{I}}_j - \sigma_j^z}{2} \right)^2}_{H_C} + \underbrace{\sum_{i\in\mathcal{V}} \sigma_i^x}_{H_T} \tag{21}$$



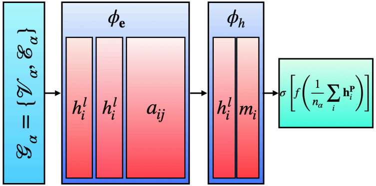
Model Architectures

GNN

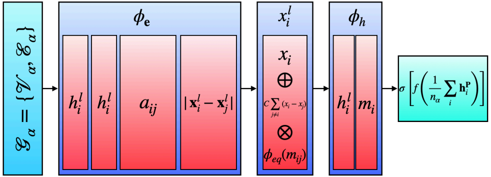


Model Architectures

GNN

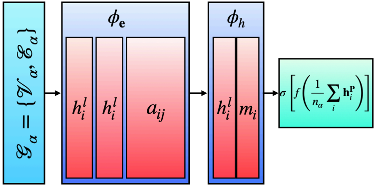


EGNN

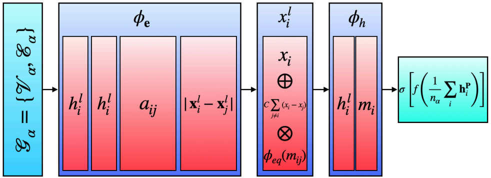


Model Architectures

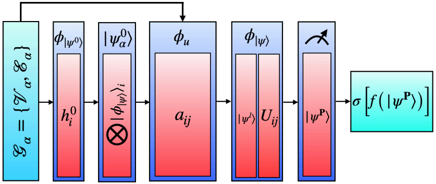
GNN



EGNN

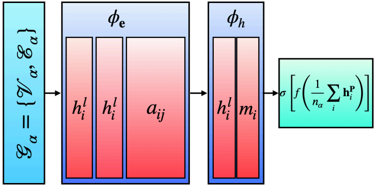


QGNN

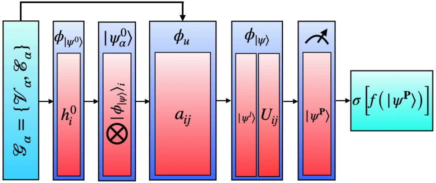


Model Architectures

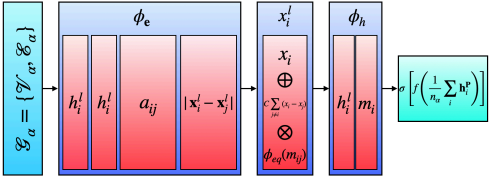
GNN



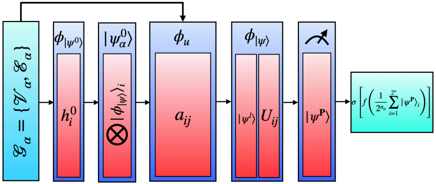
QGNN



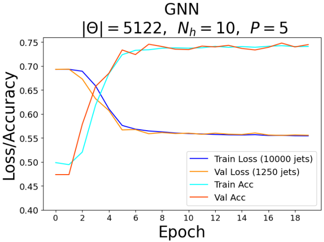
EGNN



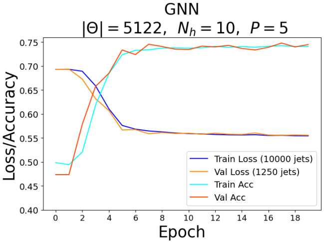
EQGNN

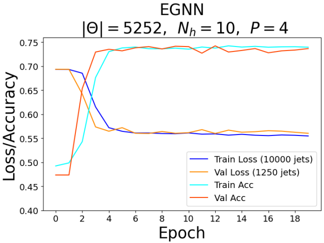


Model Performances

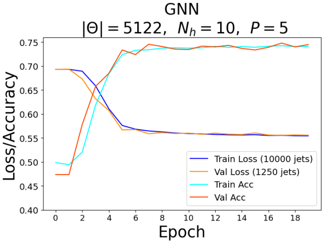


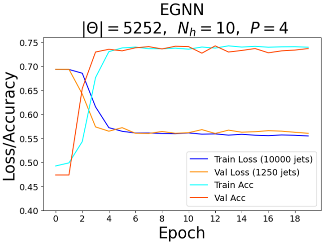
Model Performances

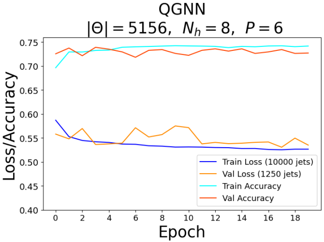




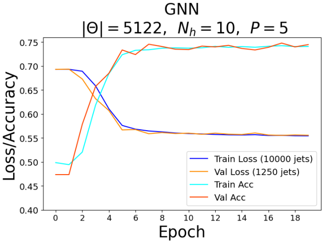
Model Performances

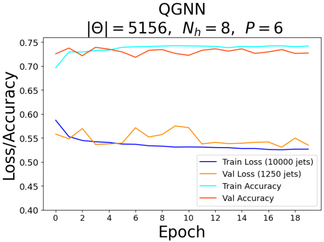


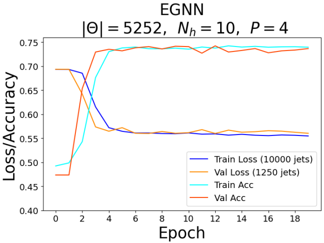


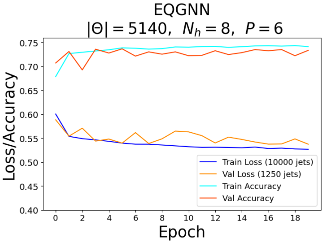


Model Performances





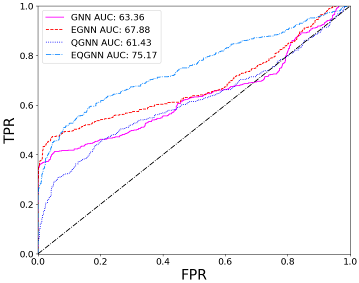


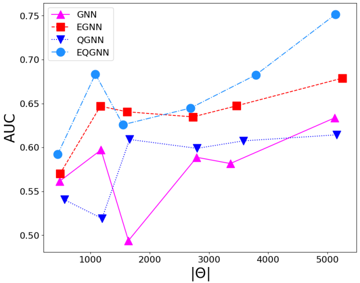


Results

Table: Metric comparison between the classical and quantum graph models.

Model	$\lvert\Theta\rvert$	N_h	P	Train ACC	Val ACC	Test AUC
GNN	5122	10	5	74.25%	74.80%	**63.36%**
EGNN	5252	10	4	73.66%	74.08%	**67.88%**
QGNN	5156	8	6	74.00%	73.28%	**61.43%**
EQGNN	5140	8	6	74.42%	72.56%	**75.17%**





Conclusion and Outlook

Takeaways

- **Statement: Quantum GNNs exhibit enhanced classifier performance over their classical GNN counterparts based on the best test AUC scores produced after the training of the models while relying on a similar number of parameters, hyperparameters, and model structures.**
- **However, the community requires a significant improvement in quantum APIs.**
  - E.g. Pennylane does not support broadcastable operators, i.e. train on one graph at a time.
  - Quantum algorithms took nearly 100 times as long to train.
  - Difficult to construct the quantum layers with enough trainable parameter flexibility.
- **Model improvements include**
  - Further theoretical foundations for complex quantum algorithms.
  - More general equivariance, e.g. unitary SU(2), Lorentz SO(1,3) etc.
  - Greater complexity, e.g. quantum attention mechanism (AT).
  - Testing among different tasks, e.g. classification, regression, etc.
  - Improved quantum optimizers and API integration.

Resources and Software

@ ML4SCI

Developing and Documentation

Packages and APIs

NumPy

PENNYLANE

PyTorch

TensorCircuit

Computing and Testing

jupyter

NVIDIA.

HiPerGator 3.0

Blogging and Connecting

Mattermost

Medium

Resources and Code

Figure: **Code** (left) and **website** (right).

References I

[1] El Amine Cherrat et al. “Quantum Vision Transformers”. In: (2022). arXiv: 2209.08167 [quant-ph].

[2] Justin Gilmer et al. “Neural Message Passing for Quantum Chemistry”. In: (2017). arXiv: 1704.01212 [cs.LG].

[3] Thomas N. Kipf and Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: (2017). arXiv: 1609.02907 [cs.LG].

[4] Peter Mernyei, Konstantinos Meichanetzidis, and Ismail Ilkan Ceylan. “Equivariant Quantum Graph Circuits”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 15401–15420. URL: https://proceedings.mlr.press/v162/mernyei22a.html.

References II

[5] Johannes Jakob Meyer et al. "Exploiting Symmetry in Variational Quantum Machine Learning". In: *PRX Quantum* 4.1 (2023). DOI: 10.1103/prxquantum.4.010328. URL: https://doi.org/10.1103%2Fprxquantum.4.010328.

[6] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. "E(n) Equivariant Graph Neural Networks". In: (2022). arXiv: 2102.09844 [cs.LG].

[7] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. "Graph neural networks in particle physics". In: *Machine Learning: Science and Technology* 2.2 (2021), p. 021001. DOI: 10.1088/2632-2153/abbf9a. URL: https://doi.org/10.1088%2F2632-2153%2Fabbf9a.

[8] Guillaume Verdon et al. "Quantum Graph Neural Networks". In: (2019). arXiv: 1909.12264 [quant-ph].