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

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*))

Dataset [7]

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

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

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

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

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

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

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

Dataset [7]

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

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

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

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

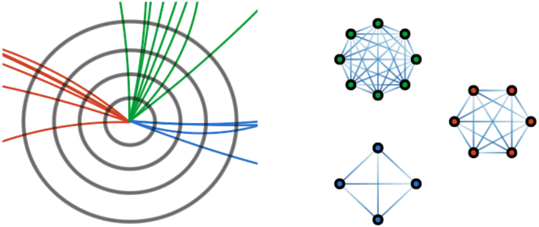
(Jet (n), Multiplicity (m), Feature (l))

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

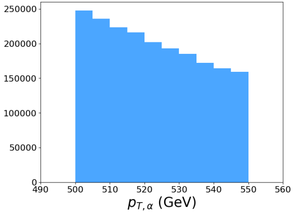
$$\boxed{x_\alpha^{(il)} \in \{p_T, \eta, \phi, m_p\}} \tag{1}$$

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

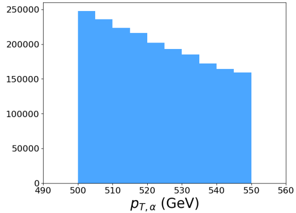
$$\boxed{y_n \in \{0,1\} \text{ for Binary Classification}} \tag{2}$$

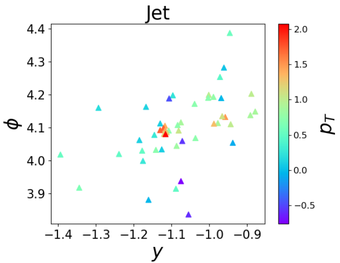


Data Distributions & Feature Engineering @ ML4SCI

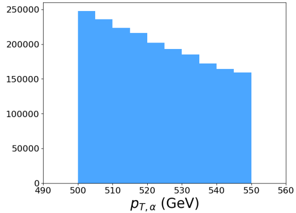


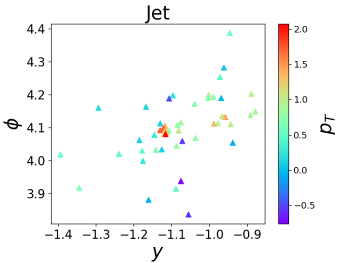
Data Distributions & Feature Engineering @ ML4SCI





Data Distributions & Feature Engineering @ ML4SCI

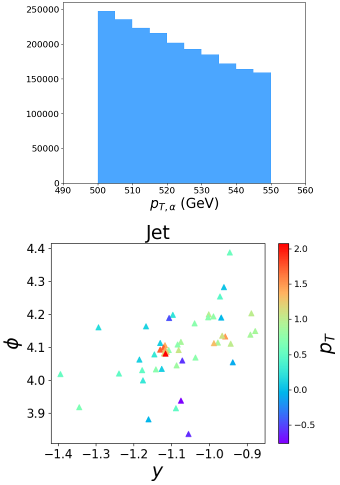




Feature set $h_\alpha^{(il)}$ with $l = 0, 1, 2, \ldots, 7$:

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

Data Distributions & Feature Engineering @ ML4SCI



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

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

Edge Connections a_{ij}

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

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

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

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

Quantum

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

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

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

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

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

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

$\downarrow$

Invariance vs. Equivariance [6, 5]

@ ML 4 SCI

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

$$gx_i^l \rightarrow$$

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

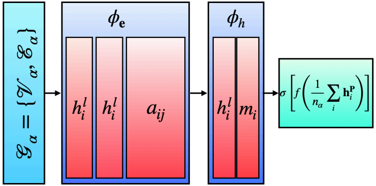
Quantum

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

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

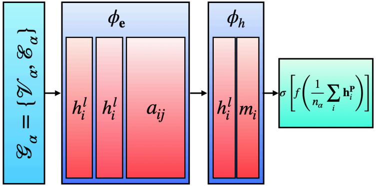
Model Architectures

GNN

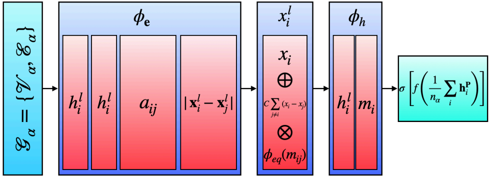


Model Architectures

GNN

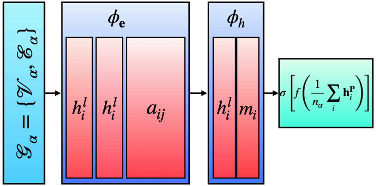


EGNN

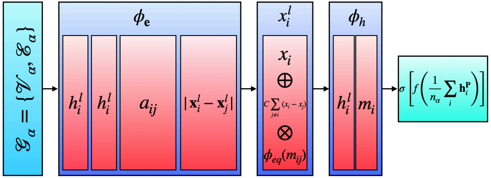


Model Architectures

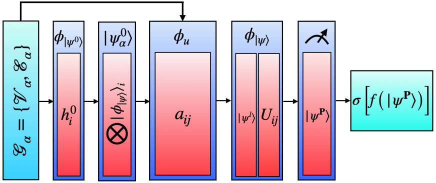
GNN



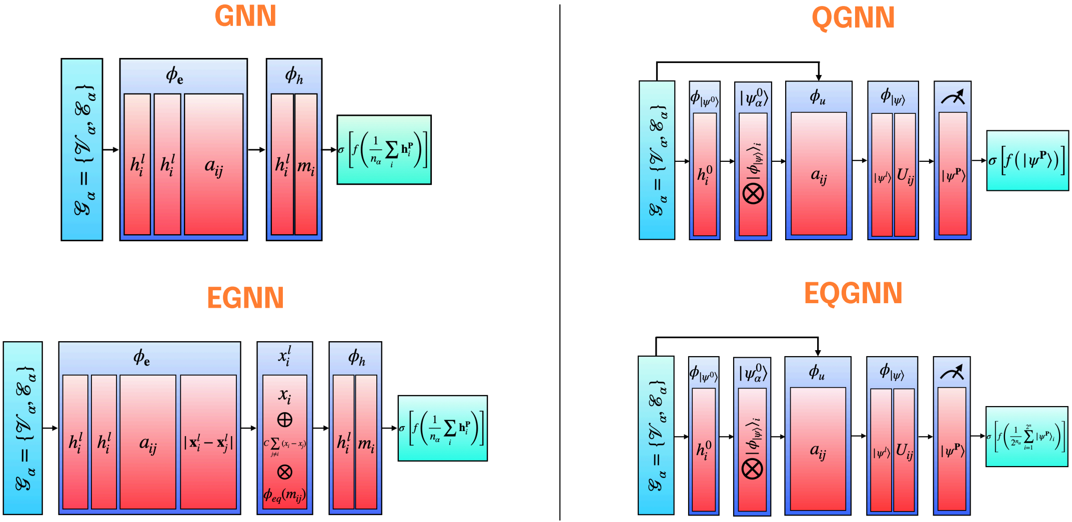
EGNN



QGNN



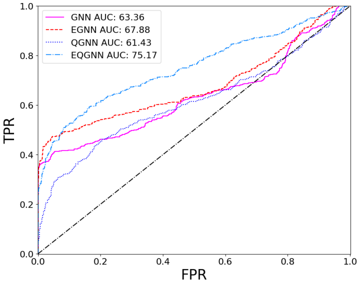
Model Architectures

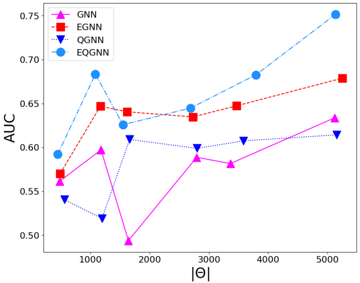


Results

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

Model	$\|\Theta\|$	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

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).