Quantum Graph Transformers

Georgios Kollias, Vassilis Kalantzis, Theodoros Salonidis, and Shashanka Ubaru

May 2023

EPrint ID: 2023.3

QUANTUM GRAPH TRANSFORMERS

Georgios Kollias *Vassilis Kalantzis* *Theodoros Salonidis* *Shashanka Ubaru*

IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY, USA

ABSTRACT

We propose Quantum Graph Transformers (QGT), a novel approach for realizing the Transformer architecture for graph learning with quantum processors. QGT is built on top of the Graph Transformer (GT) architecture and addresses the main challenge of mapping GT basic functions such as node encodings, graph structure, all-to-all connectivity, and message passing to quantum computing primitives and processors. We empirically demonstrate the training and inference efficacy of our proposed QGT architecture for the graph classification task on quantum devices over various graph datasets.

1. INTRODUCTION

Graphs are ubiquitous and among the most general data structures, spanning diverse application areas from interactions in biology, to drug design, financial transactions, and social relations. Graph analysis holds the key to critically useful insights and optimizations of modelled processes as well as the organization of connected entities. Graph AI has emerged as a new and thriving research subfield of graph analytics to expand learning on graphs using neural network architectures which produce vector representations (embeddings) for vertices, edges and subgraphs. Such resulting embeddings have proven to be highly effective for an abundant repertoire of graph analysis tasks, including node classification, link prediction, and graph property estimation. Graph Neural Networks (GNNs) have been at the forefront of these developments by generalizing the idea of message passing, where exchanged messages are representations of graph primitives that are linearly transformed and combined in a non-linear manner. Such transformations are learnt by optimizing a task-dependent objective while messages flow along graph edges. More recently, graph AI has been enriched with Graph Transformers (GTs) that originate from research in Natural Language Processing (NLP). GTs improve upon GNNs for some analytic tasks because they assume an all-to-all graph connectivity and can learn hidden links which are not present in the original graph structure.

The above discussion considers a classical computing context. In this paper we introduce a Quantum Graph Transformer (QGT) architecture for *quantum computing*[1]. Quantum computing promises a sweeping change on the envelope of what actually lies within the practical limits of computational reach. Within the last few decades, quantum computers have transcended from an exotic mode of performing calculations, to real programmable devices which can challenge the capabilities of very large distributed memory digital (classical) computers on tasks such as Topological Data Analysis [2, 3] and solution of systems of linear algebraic equations [4, 5, 6]. Of particular interest are algorithms which lie at the intersection of AI and quantum computing spanning the research field of Quantum Machine Learning (QML) [7].

Emails: gkollias@us.ibm.com, vkal@ibm.com, tsaloni@us.ibm.com, shashanka.ubaru@ibm.com

Graph learning has not yet been extensively studied from a quantum computing viewpoint. In this paper, we aim to advance practice in this topic. We explore the design space of mapping Graph Transformer (GT) models to quantum computing primitives and identify commonly used patterns: (i) graph node encodings are captured in qubit states and graph edges translate to two-qubit gates of a quantum circuit, (ii) parametric rotation and control gates contain learnable parameters. The QGT architecture uses the above principles and addresses the challenge of mapping GT primitives such as message passing and graph encodings to quantum computing primitives and infrastructure.

QGT is built on top of a GT architecture and incorporates two circuits which translate 1) the graph structure and 2) the all-to-all GT attention and message passing mechanisms to quantum computing primitives. More specifically, QGT provides a direct mapping between qubits and graph nodes. The structure of the graph is represented via an *encoding quantum circuit* which obeys the graph edge connectivity. The state of qubits is manipulated by a combination of Hadamard and controlled-Z gates. The output *quantum graph state* [8] serves as the quantum representation of the given graph. All-to-all node interactions with learnable weights for the GT message passing are represented by a *variational quantum circuit* that connects all possible node pairs. Connection between two nodes is implemented through a controlled-X gate between two qubits.

We perform an empirical evaluation of QGT on quantum devices that solve graph classification problems. Our results on a variety of input graphs demonstrates that QGT is effective and computationally attractive in minimizing training loss. They also confirm the attractiveness of learning variational circuit parameters in QPU device configurations for inference purposes.

Our main contributions can be summarized as follows:

- We propose a QGT architecture that cleanly isolates *graph encoding* and *all-to-all message passing*, which are the two key functions in GTs, in separate, connected circuits: the *encoding* and *variational* quantum circuits.
- We successfully *train* the QGT model for performing a *graph classification task* and demonstrate the learning capability of our architecture by tracking *loss* degradation and *average precision score* improvement during training.
- We implement QGT on a range of simulated *QPU devices*, *calibrate* by directly transferring the parameters of the trained model and experimentally confirm the efficacy in *inference* for our task on the QPU configurations.

To the best of our knowledge, this is the first effort to propose a QGT architecture, successfully experiment with it and reason about its high-level connections to classical counterparts.

2. RELATED WORK

Our proposed QGT builds on key architectural elements found in GTs and targets QML applications. GTs fuse ideas from GNNs and Transformers which are important learning structures in AI.

The GNN model can be traced back in the seminal work in [9]. It was popularized in Graph Convolution Networks (GCNs) [10], where the authors approximate the spectral representation of convolution over the irregular graph structure [11], offering a simplified message passing (MP) view. The latter yields a spatial-based description which generalizes to Message Passing Neural Networks [12]. The Transformer architecture [13] introduced the *attention* mechanism acrosss sequence elements, facilitating the identification of latent, potentially long-range, interactions. In Graph Attention networks [14], it is suggested to learn attention weights for the edges of the input graph. The multi-head attention module in Transformers was generalized and ported for graph learning, resulting in an early instance of Graph Transformer architecture in [15]; this model hardwires a strong inductive bias constraining attention to known neighbor nodes. In Graphormer [16], the GT architecture was extended with implicit, detailed encoding of structure and attention to non-adjacent vertices; in [17] specialized edge channels were added to allow explicit learning of pairwise structural information.

One of the key advantages of QML is its ability to generate exponentially large quantum states for feature representation [18]. QML computations can be organized in stages (pre-processing, parameterized quantum circuit (PQC), post-processing) [19] that successfully transform the input quantum state and measure at the end; PQCs are tunable and are the direct analogues of classic neural networks (NNs). Alternative methods for *unsupervised* QML are studied in [20]. For *graph learning* in particular - see [21] for a survey - [22] introduce recurrent and convolutional architecture variants. A simple quantum GCN architecture is proposed and simulated in [23]. In [24] quantum GNNs that exhibit equivariance under permutations of the adjacency matrix are explored; the idea of sampling graphs for training quantum GCNs in order to manage the limited qubit count of current QPUs is investigated in [25]. BERT is a Transformer model and in [26] they replace some layers of its decoder by a quantum temporal convolution learning framework. A quantum transformer is described and simulated in [27] following the evaluation of a quantum LSTM architecture.

3. QUANTUM GRAPH TRANSFORMER ARCHITECTURE

In this section, we describe our proposed Quantum Graph Transformer (QGT) architecture, also outlined and compared to Graph Transformers (GTs) in Figure 1. In a nutshell, our QGT architecture builds on top of GT. QGT encodes the graph structure with a circuit; in comparison, GT uses a PE vector per node for this purpose. Similarly, all-to-all message passing in GTs is mapped to another quantum circuit with parametric gates and clique connectivity in QGT architecture. Next, we describe the main idea behind GTs, followed by details on their implementation in quantum computers.

3.1. Graph Transformers and Attention

The following discussion assumes a GT architecture similar to the ones in [15, 16, 17]. Consider now an undirected, unweighted graph $G = (V, E)$ of n nodes, where $V = \{1, \ldots, n\}$ denotes the set of vertices and $E = \{(i,j) | i,j \in V\}$ denotes the set of edges, respectively. The connectivity information of G can be represented via an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ where $\mathbf{A}_{ij} = 1$ *iff* $(i,j) \in E$, and zero otherwise.

Let now each vertex i of G be associated with an encoding vector $\mathbf{x}_i \in \mathbb{R}^d$. Such encodings can be initially constructed by composing embedded graph properties and node feature vectors, and can be organized in a matrix form where $\mathbf{x}_i$ denotes the ith row of the encoding matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. The main premise behind Graph Transformers is to perform message passing of node encodings by assuming an all-to-all connectivity between the entities exchanging encodings, i.e., the vertices of G. These node encodings are then transformed by a weight matrix, where the individual weight between any pair (i,j) is decided by the similarity of their transformed encodings; a process referred to as *attention* between nodes i and j. The similarity function is chosen to be a normalized dot product. Thus, the GT adopts the attention head in standard Transformers and updates the encodings of each head at each layer as

$$\mathbf{X} \leftarrow \texttt{softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^\top}{\sqrt{d}}\right) \mathbf{V} \tag{1}$$

where query, key and value vectors are the transformed encoding vectors organized as rows in the respective matrices $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \mathbf{K} = \mathbf{X}\mathbf{W}_K$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_V$, where the weight matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learned during training.

A careful look at (1) indicates that GTs implicitly replace the adjacency matrix $\mathbf{A}$ by the dense matrix $\texttt{softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^\top}{\sqrt{d}}\right)$. This replacement gives GTs the flexibility to learn latent connections between graph nodes that are not reflected in the adjacency structure (global self-attention). This is in contrast to GNNs which can perform message passing only with their immediate neighbors, as dictated by the graph connectivity. For example, GCNs update the encoding matrix via the matrix multiplications $\mathbf{X} \leftarrow \sigma(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{W})$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\sigma(\cdot)$ denotes the sigmoid function, and $\mathbf{W} \in \mathbb{R}^{d \times d}$ are learnable weights [10]. Thus, in contrast to GTs, exchanging information between vertices with long distances requires several GNN/GCN updates.

While GTs can benefit from learning latent connectivity patterns which are not dictated by $\mathbf{A}$, generally it is still beneficial to try and preserve some form of information regarding the connectivity of G. GTs can invariably integrate such information through a positional encoding (PE) scheme, where vertices are enriched with vectors denoting their position within the graph structure In [28] a number of PEs are explored; for undirected graphs, a linear transformation of Laplacian encoding is a common PE, where each node is assigned the corresponding row of the top eigenvector matrix of its graph Laplacian [15]. The PE vector $\mathbf{p}_i$ for node i is added to its encoding and the sum $\mathbf{x}_i + \mathbf{p}_i$ serves as the final input to GT.

Next we describe how graph structure, PEs and message passing are realized in the quantum context.

3.2. Quantum GT Representations

It is evident that GT features two important and seemingly counteracting requirements. On the one hand it should facilitate all-to-all interactions of its nodes, in order to learn new edges, which means "forgetting" the actual graph connections. On the other hand, preserving information of the original graph structure is important since the adjacency matrix is already available. Indeed, GTs meet the first requirement by the global self attention in Equation 1, which allows any two nodes to interact. The second requirement is then fulfilled by PE, which represents an encoding of the whole graph. Given the above discussion, we argue that a QGT can be built as follows:

1. The structure of the graph is represented via an *encoding quantum circuit* which obeys the edge connectivity.
2. All-to-all node interactions with learnable weights are represented by a *variational quantum circuit* that connects all possible node pairs.

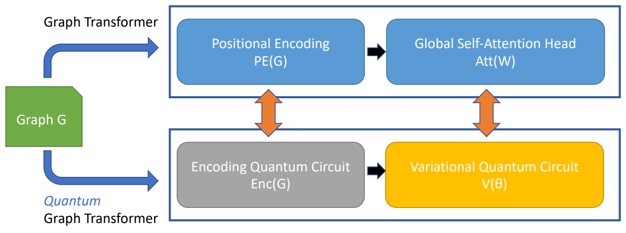

Fig. 1: GT vs QGT architecture.

Encoding Quantum Circuit In this circuit, the Hadamard gate is applied to the input qubits which are assumed prepared in the $|0\rangle$ state producing the superposition state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (computation basis change). Then controlled-Z gates are applied between qubits representing nodes that are connected in the graph. In the computational basis, this gate flips the phase of the target qubit if the control qubit is in the $|1\rangle$ state. The output *quantum graph state* [8] serves as the representation of the given graph.
Variational Quantum Circuit This is a PQC which assumes all-to-all connectivity between qubits [1]. Note that a connection between two nodes is implemented through a controlled-X gate between two qubits. In the computational basis, this gate flips the target qubit if the control qubit is in the $|1\rangle$ state. In this sense it is similar to a classical XOR gate. The learnable entities, akin to weight matrices, are represented by the angles of rotational gates around the Y axis: $RY(\theta) = \exp(-i\frac{\theta}{2}Y)$. We allow two series of such weight-carrying rotational gates, one at each of the input and output sides of this circuit for a total of $2n$ learnable parameters.
Figure 2 includes realizations of the two circuits comprising QGT for graphs with $n = 6$ nodes.

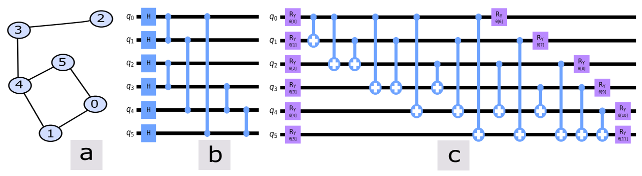

Fig. 2: An example graph for $n = 6$ (a), its encoding circuit (b) and the variational circuit (c). (b) and (c) are connected. Measurement circuits are assumed immediately after (c).

4. EXPERIMENTS

4.1. Datasets

We generate a graph dataset consisting of k undirected, connected graphs $G_r(V_r, E_r)$, $r = 1, \ldots, k$, and assosiated binary class labels $l_r \in \{0, 1\}$. A graph is assigned a label equal to one if it contains a 3-clique, i.e., a subgraph of size three where all three vertices are connected to each other. Similarly, a graph is assigned a label equal to zero if it contains no triangles. Our dataset is formed by generating connected Watts–Strogatz small-world graphs, where the number n of vertices is kept fixed. Throughout our experiments we consider $k = 20$ and $n \in \{5, 6\}$; a snapshot of the generated graphs for the case $n = 6$ is visualized in Figure 4.

[1]RealAmplitudes: `https://qiskit.org/documentation/stubs/qiskit.circuit.library.RealAmplitudes.html`

QPU codename	Initial AP	Learnt AP
`ibm_oslo`	0.624 ± 0.149	0.984 ± 0.021
`ibm_nairobi`	0.588 ± 0.055	0.981 ± 0.019
`ibmq_manila`	0.533 ± 0.087	0.976 ± 0.023
`ibmq_quito`	0.573 ± 0.067	0.973 ± 0.020
`ibmq_lima`	0.604 ± 0.109	0.973 ± 0.017

Table 1: Average precision scores ($n = 5$).

4.2. Experimental Setup and Results

The measurement of the output state results in a vector of 2^n entries, where the jth entry denotes the probability to land to the corresponding configuration. An output configuration consists of a binary string label (e.g., '100101' for $n = 6$). Note that we choose to compute the *parity* of this label (being either 0 or 1) and agree on accumulating its probability on class label 0 or 1. So the probabilities of all output configurations with *parity* 0 are added together and their sum is considered to be the probability of getting *classification* label 0 (i.e., no triangles in the input graph). For a given graph and set of parameters, we compute the negative log loss for the predicted classification probabilities against the true classification label. We average these (binary cross entropy) loss terms over *all* graphs and true classification labels of our dataset: *this is the value of our loss function for the given parameters*. In order to learn parameters that minimize the loss function we leverage AMSGRAD[29], a variant of the Adam optimizer [30], using learning rate $\eta = 10^{-2}$. During training we record the evolution of training loss across each iteration. At the end of the training phase, we compute the *average precision (AP) score* for the initial and learnt ("optimized") parameters in our *variational circuit*. We repeat the training phase 5 times, each with a different seed; the variational circuit is initialized with zero angles in all cases. For graphs with $n = 6$ nodes, AP score improves dramatically from 0.330 ± 0.024 to 0.770 ± 0.079 demonstrating the learning capability of our quantum circuit architecture. Figure 5 demonstrates the evolution of training loss across different experiment instances.

We investigate the robustness of learnt parameters across a range of 5- and 7- qubit devices,with codenames repectively in the sets `ibmq_(manila|quito|lima)` and `ibm_(oslo|nairobi)`. Figure 6 illustrates the gate maps of the latter devices in the order they appear in the text. We generate graph datasets for $n = 5$ nodes, in order to later enable experiments also with the 5-qubit machine configurations and trace the training of QGT model. AP score improves from 0.440 ± 0.037 to 0.994 ± 0.008, confirming the learning efficacy of our approach over the new dataset collection. We then import the coupling map, noise model and basis gates for each of the QPUs and implement QGT circuits. More specifically, the variational circuit in QGT is populated with (i) the *initial* conditions (zero rotation angles) and (ii) the rotation angles *learnt* as in the trained model for all QPU configurations; AP scores are computed in both cases. For sampling purposes, each circuit is repeated 1,024 times (number of shots). Table 1 summarizes our findings.

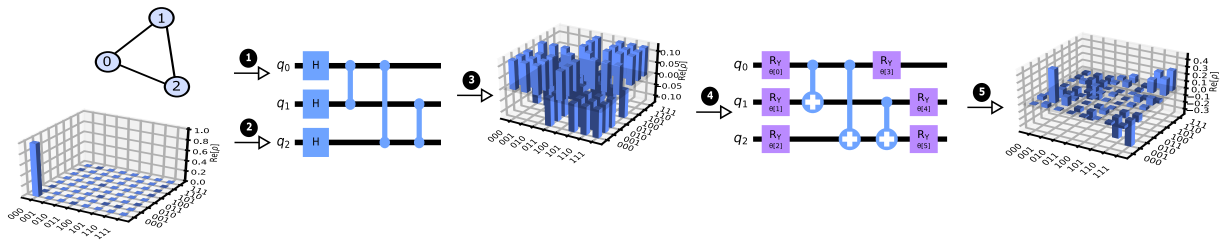

Fig. 3: Steps in the proposed QGT architecture for processing an example graph of $n = 3$ nodes: The input graph, here a 3-clique (1), is used to build the encoding circuit (leftmost circuit). Then the base state $|0\rangle^{\otimes 3}$ is prepared and fed into the circuit (2) producing the encoding state (3). This is transformed by the variational circuit (4) to yield the final output state (5). Here we assume $\theta = \pi/4$ for all 6 rotation angles in the variational circuit; bar graphs of the real part of state density matrices $\mathfrak{Re}(\rho)$ are shown.

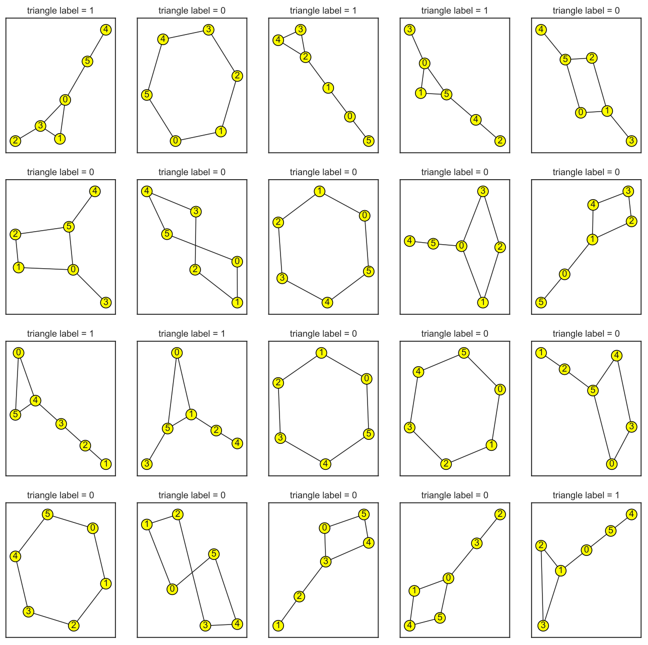

Fig. 4: Snapshot of graphs.

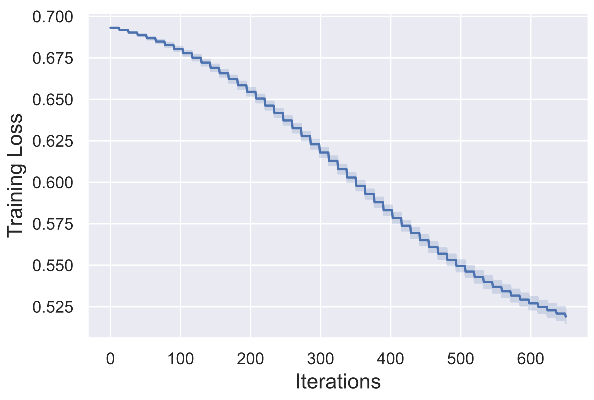

Fig. 5: Training loss.

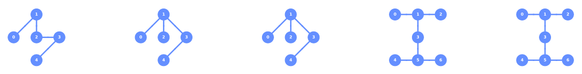

Fig. 6: 5- and 7- qubit devices.

Essentially, we *transfer* (copy) parameters learnt in *training* QGT over an idealized and fast simulator in order to calibrate the QGT architecture to be used for *inference* over real device setups. The robustness of the relative improvements demonstrates that this training/inference pattern is effective and computationally attractive for our proposed QGT.

5. CONCLUDING REMARKS

In this paper we presented a QGT architecture for the task of graph classification. The proposed architecture performs graph positional encoding via inputting the information of the adjacency matrix in a quantum circuit. In addition, an attention mechanism, i.e., all-to-all node interactions, is implemented via a trainable variational quantum circuit. Aiming at simplicity and minimal parameters to learn, our variational circuit uses a single layer of two-qubit entanglement and fixed-type rotation gates leading to quantum states with real amplitudes. We evaluated the learning efficacy of the proposed QGT for the task of graph classification where the dataset consists of small-world graphs and the goal is to identify graphs which contain at least one triangle. Our experiments confirm the effectiveness of learning variational circuit parameters in QPU device configurations for inference purposes.

As part of future work, we plan to study the replacement of single-layer, two-qubit entanglement with multiple-layer, multiple-qubit entanglement circuit configurations with rotation gates from richer unitary groups. Tracing the relative changes in rotation angle vectors as graph edges get encoded, or as entangling connections are removed, can help to translate global self-attention as rotations in the quantum context. The trained QGT model can then encode and map a graph of n nodes to a quantum state of 2^n complex numbers. For graph-level tasks, this is an exponentially richer representation than the d real numbers ($O(1)$) from typical pooling operations on top of the n, d-dimensional, node encodings learnt during typical GT training. This can lead to major advances with potentially important practical applications for n in the order of a few tens or hundreds [31].

6. REFERENCES

[1] Michael A Nielsen and Isaac L Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.

[2] Alexander Schmidhuber and Seth Lloyd, "Complexity-theoretic limitations on quantum algorithms for topological data analysis," 2022.

[3] Vasileios Kalantzis, Mark S. Squillante, Shashanka Ubaru, and Lior Horesh, "On quantum algorithms for random walks in the nonnegative quarter plane," *SIGMETRICS Perform. Eval. Rev.*, vol. 50, no. 2, pp. 42–44, aug 2022.

[4] Ismail Yunus Akhalwaya, Shashanka Ubaru, Kenneth L. Clarkson, Mark S. Squillante, Vishnu Jejjala, Yang-Hui He, Kugendran Naidoo, Vasileios Kalantzis, and Lior Horesh, "Towards Quantum Advantage on Noisy Quantum Computers," *arXiv e-prints*, p. arXiv:2209.09371, Sept. 2022.

[5] X-D Cai, Christian Weedbrook, Z-E Su, M-C Chen, Mile Gu, M-J Zhu, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan, "Experimental quantum computing to solve systems of linear equations," *Physical review letters*, vol. 110, no. 23, pp. 230501, 2013.

[6] Dong An and Lin Lin, "Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm," *ACM Transactions on Quantum Computing*, vol. 3, no. 2, pp. 1–28, 2022.

[7] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[8] Marc Hein, Jens Eisert, and Hans J Briegel, "Multiparty entanglement in graph states," *Physical Review A*, vol. 69, no. 6, pp. 062311, 2004.

[9] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[10] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[11] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun, "Spectral networks and deep locally connected networks on graphs," in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1263–1272.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[14] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio, "Graph attention networks," *6th International Conference on Learning Representations*, 2017.

[15] Vijay Prakash Dwivedi and Xavier Bresson, "A generalization of transformer networks to graphs," *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[16] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu, "Do transformers really perform badly for graph representation?," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28877–28888, 2021.

[17] Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian, "Global self-attention as a replacement for graph convolution," *arXiv preprint arXiv:2108.03348*, 2021.

[18] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.

[19] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, pp. 043001, 2019.

[20] Yannick Deville and Alain Deville, "New single-preparation methods for unsupervised quantum machine learning problems," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–24, 2021.

[21] Yehui Tang, Junchi Yan, and Hancock Edwin, "From quantum graph computing to quantum graph learning: A survey," *arXiv preprint arXiv:2202.09506*, 2022.

[22] Guillaume Verdon, Trevor McCourt, Enxhell Luzhnica, Vikash Singh, Stefan Leichenauer, and Jack Hidary, "Quantum graph neural networks," *arXiv preprint arXiv:1909.12264*, 2019.

[23] Jin Zheng, Qing Gao, and Yanxuan Lü, "Quantum graph convolutional neural networks," in *2021 40th Chinese Control Conference (CCC)*. IEEE, 2021, pp. 6335–6340.

[24] Péter Mernyei, Konstantinos Meichanetzidis, and Ismail Ilkan Ceylan, "Equivariant quantum graph circuits," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15401–15420.

[25] Xing Ai, Zhihong Zhang, Luzhe Sun, Junchi Yan, and Edwin Hancock, "Decompositional quantum graph neural network," *arXiv preprint arXiv:2201.05158*, 2022.

[26] Chao-Han Huck Yang, Jun Qi, Samuel Yen-Chi Chen, Yu Tsao, and Pin-Yu Chen, "When bert meets quantum temporal convolution learning for text classification in heterogeneous computing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8602–8606.

[27] Riccardo Di Sipio, Jia-Hong Huang, Samuel Yen-Chi Chen, Stefano Mangini, and Marcel Worring, "The dawn of quantum natural language processing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8612–8616.

[28] Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun, "Graph-bert: Only attention is needed for learning graph representations," *arXiv preprint arXiv:2001.05140*, 2020.

[29] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[30] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," *arXiv preprint arXiv:2007.08663*, 2020.