QUANTUM 3D GRAPH STRUCTURE LEARNING WITH APPLICATIONS TO MOLECULE COMPUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph representation learning has been extensively studied over the last decades, and recent models start to focus on an under-explored area of 3D graph learning with 3D spatial position as well as node attributes. Despite the progress, the ability to understand the physical meaning of the 3D topology data is still a bottleneck for existing models. On the other hand, quantum computing is known to be a promising direction for theoretically verified supremacy as well as increasing evidence for access to a physical quantum device in the near term. For the first time, we propose a quantum 3D embedding ansatz that learns the latent representation of 3D structures from the Hilbert space composed of the Bloch sphere of each qubit. We convert the 3D Cartesian coordinates of nodes into rotation and torsion angles and then encode them into the form of qubits. Moreover, Parameterized Quantum Circuit (PQC) is applied to serve as the trainable layers and we take the output of the PQC as the node embedding. Experimental results on two downstream tasks, molecular property prediction and 3D molecular geometries generation, demonstrate the effectiveness of our model. Though the results are still restricted by computational power, we have shown the capability of our model with very few parameters and the potential to execute on a real quantum device.

1 INTRODUCTION

Graph representation, or specifically 3D graph representation as considered in this paper, has received extensive attention over the decades. Beyond tasks like node classification or link prediction, it further facilitates various downstream applications such as molecular property prediction (Liu et al., 2021) and drug design (Gaudelet et al., 2021). Recently, machine learning approaches have been well developed for learning latent node embedding on molecules (Schütt et al., 2017; Unke & Meuwly, 2019; Gasteiger et al., 2019; 2021). However, the mainstream of such researches is still facing the challenges of better processing the 3D Cartesian coordinates and learning the latent representation of the 3D graph structure.

On the other hand, there are also emerging lines of researches in the area of quantum computing. State-of-the-art quantum computing hardwares are now stepping into the Noisy Intermediate-Scale Quantum (NISQ) era, which gives us the possibility to implement useful applications in specific scientific domains in the near term (Preskill, 2018; Arute et al., 2019; Zhong et al., 2020; Huang et al., 2020). The overlap between quantum computing and machine learning has emerged as one of the most encouraging applications for quantum computing, which is quantum machine learning (Biamonte et al., 2017). Quantum paradigms or hybrid paradigms have been carefully designed to fulfill quantum supremacy in quantum chemistry problems (Aspuru-Guzik et al., 2005; O'Malley et al., 2016). Existing approaches mainly focus on the quantum simulation of molecular energies, which enables numerically exact prediction of chemical reaction rates. However, these approaches are still simulating the energies of certain small molecules like H_2, LiH, etc.

In this paper, we aim to use quantum machine learning approaches to learn the latent representation of the 3D graph structure of molecules instead of directly simulating the molecular energies with Hamiltonians. Graph learning approaches may not be as precise as molecular simulation approaches when predicting properties, but they have the ability to learn hundreds or thousands of molecules and predict the preperties for more complex molecules. We first convert the 3D Cartesian coordinates of the atoms into three geometries, which are distance, rotation angle, and torsion angle. Then we

encode the angles and distance as well as the atom type, which is a discrete variable, into qubits. We set up a distance threshold so that each time we pick a focal atom to learn the embedding, we only need to consider the neighboring atoms within the threshold. Considering the size of the molecules and the size of the neighborhood, we only require up to ten qubits to learn the representation, which makes our proposed model easy to simulate on a classical processor and capable of running on a NISQ device. Analog to the hardware efficient ansatz (Kandala et al., 2017; Huang et al., 2021), we apply a Parameterized Quantum Circuit (PQC) after the encoding stage. The trainable parameters are the $\boldsymbol{\theta}$s of the rotation gates $\mathbf{R_x}$ and $\mathbf{R_y}$ in the PQC. The gradient of each parameter $\boldsymbol{\theta}$ is calculated by the shifting technique (Mitarai et al., 2018), and those parameters are updated by the backpropagation and gradient descent approach analog to classical neural networks. We apply a tomography at the end of the circuit and concatenate the real part and imaginary part of the output vector and then take it as the node embedding. We conducted numerical experiments on the filtered QM9 dataset for both molecular property prediction task and molecular geometries generation task. Experimental results show that compared with classical state-of-the-art baseline models, our quantum 3D embedding model achieves comparable results on small datasets with much fewer network parameters. **We summarize our contributions as follows:**

1) To the best of our knowledge, we are the first to use qubits to encode 3D relative positional information, which preserves the property of equivariance and invariance. Using a qubit on a Bloch sphere to encode the rotation and torsion angle of two atoms is more intuitive than using 3D Cartesian coordinates, which is also supported by the success of spherical representation on point clouds and molecules in recent studies.

2) We use two qubits to represent each atom, and we only consider the focal atom and its neighbors at each iteration. Therefore, the maximum number of qubits is 10 in our model. So we are able to test our model on Qiskit (http://qiskit.org) with quantum cloud service from IBM-Q, which shows the potential ability to run on a NISQ device.

3) We manage to implement a quantum circuit full-amplitude simulator with transition unitary for the PQC on a classical processor. It replicates the results but 23 times faster than the QASM simulator from Qiskit, which enables us to conduct more experiments on various tasks.

4) The numerical experiments on two different well-studied molecular tasks show that our embedding approach is able to extract geometry and neighborhood information with very few parameters (only 64 parameters in the PQC) and achieve relatively good results.

2 Preliminaries and Related Works

In this section, we first briefly review basic concepts of quantum computing as well as quantum machine learning. We further present some previous works on quantum graph learning approaches.

2.1 Quantum Computing

In quantum computing, qubit (abbreviation of quantum bit) is a key concept which is similar to a classical bit with a binary state. The two possible states for a qubit are the state $|0\rangle$ and $|1\rangle$, which correspond to the state 0 and 1 for a classical bit respectively. We refer the readers to the textbook (Nielsen & Chuang, 2002) for comprehension of quantum information and quantum computing. In this paper, we give a compact description of background for self-containess.

A quantum state is commonly denoted in bracket notation. It is also common to form a linear combinations of states, which we call a superposition: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Formally, a quantum system on n qubits is an n-fold tensor product Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes d}$ with dimension 2^d. For any $|\psi\rangle \in \mathcal{H}$, the conjugate transpose $\langle\psi| = |\psi\rangle^\dagger$. The inner product $\langle\psi|\psi\rangle = ||\psi||_2^2$ denotes the square of the 2-norm of ψ. The outer product $|\psi\rangle\langle\psi|$ is a rank 2 tensor. Computational basis states are given by $|0\rangle = (1, 0)$, and $|1\rangle = (0, 1)$. The composite basis states are defined by e.g. $|01\rangle = |0\rangle \otimes |1\rangle = (0, 1, 0, 0)$.

Analog to a classical computer, a quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate the quantum information. A quantum gate is a unitary operation $\boldsymbol{U}$ on Hilbert space $\mathcal{H}$. When we simulate the quantum circuit on a classical computer, we can obtain the overall transition unitary by tensoring and multiplying those unitary gate operators together.

A projective measurement is described by an observable, M, a Hermitian operator on the state space of the system being observed. The observable has a spectral decomposition, $M = \sum_m m\boldsymbol{P}_m$, where $\boldsymbol{P}_m$ is the projector onto the eigenspace of M with eigenvalue m. When measuring the state $|\psi\rangle$, the probability of getting results m is given by $p(m) = \langle\psi|\boldsymbol{P}_m|\psi\rangle$.

2.2 Quantum machine learning

(Cerezo et al., 2021) proposed the concept of Variational Quantum Algorithms (VQA), which leverages quantum advantages to solve machine learning problems on a near-term quantum device. Then, Parameterized Quantum Circuits (PQC) are the concrete implementation of certain VQA. For each qubit we have rotation operator $\mathbf{R_x}(\boldsymbol{\theta})$ which rotate through angle $\boldsymbol{\theta}$ (radias) around the x-axis. A PQC is mainly composed of $\mathbf{R_x}(\boldsymbol{\theta})$, $\mathbf{R_y}(\boldsymbol{\theta})$ and $\mathbf{R_z}(\boldsymbol{\theta})$ with $\boldsymbol{\theta}$ as the parameters. The parameters $\boldsymbol{\theta}$ are updated by a classical optimizer to minimize the loss function $\mathcal{L}(\boldsymbol{\theta})$ which evaluates the dissimilarity between the output of PQC and the target result. The derivative of the i-th parameter $\boldsymbol{\theta}(i)$ can be computed by using the shifting technique proposed by (Mitarai et al., 2018). It requires running the whole circuit twice but with shifting $\boldsymbol{\theta}(i)$ to $\boldsymbol{\theta}(i) + \pi/2$ and $\boldsymbol{\theta}(i) - \pi/2$

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}(i)} = \frac{\mathcal{L}(\boldsymbol{\theta}(1), \cdots, \boldsymbol{\theta}(i) + \pi/2, \cdots) - \mathcal{L}(\boldsymbol{\theta}(1), \cdots, \boldsymbol{\theta}(i) - \pi/2, \cdots)}{2} \tag{1}$$

The gradient update for the parameter $\boldsymbol{\theta}$ at the k-th iteration is

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} - \eta \frac{\partial \mathcal{L}(\boldsymbol{\theta}^{(k-1)})}{\partial \boldsymbol{\theta}^{(k-1)}} \tag{2}$$

where η is the learning rate. With the paradigm of backpropagation and gradient descent, numerous classical learning models are adapted into their quantum version, e.g. QCNN (Cong et al., 2019), QRNN (Bausch, 2020), QGAN (Huang et al., 2021), QLSTM (Chen et al., 2022), and etc. Those papers have shown a pattern that the quantum machine learning models on NISQ device may not be as powerful as the SOTA classical big models (usually with millions of parameters), but they are able to achieve competitive results comparing to the primary classical machine learning models. Involving quantum computing is an interesting experiment to seek potential supremacy and the connection between latent space and the mystery quantum entanglement.

2.3 Quantum graph learning

Motivated by the success of classical graph learning models, researches tried to encode the graphs into quantum expressions and use the quantum computational power to perform graph learning, leading to a new application of the quantum computing. A hierarchical architecture based on quantum random walks is employed to extract multi-scale properties of the graph (Dernbach et al., 2018). However, it is vague that how to efficiently construct the diffusion matrix from the quantum states generated by the quantum walkers. The information aggregation is performed by the classical system, which further incurs additional expenses as a consequence of the interaction between quantum and classical environment. (Zhang et al., 2019) and (Ai et al., 2022) suggest to exploit the quantum Hilbert space to rebuild the quantum representation of the graph in the quantum state. But the number of qubits to represent a graph with its node attributes scales linearly with the number of nodes, and the width of the quantum circuit increases with the complexity of the graph structure as well as the dimension of node features, which is extremely harsh for current quantum devices. Besides, the data extraction process necessitates quantum tomography, which requires exponential measurement overheads. (Bai et al., 2021) and (Chen et al., 2021) develop a hybrid graph learning model which consists of quantum layer and classical layer aimed at reflecting richer graph characteristics. But they both lack formal justifications for the quantum model selections. It is unclear whether the classical part or the quantum part contributes more to the performance of the whole model.

3 Methodology

3.1 Problem setting and method overview

Problem Setting. In this paper, we aim to develop a quantum machine learning approach for learning node embedding with node-wise 3D coordinates. We take molecules with 3D graph structures

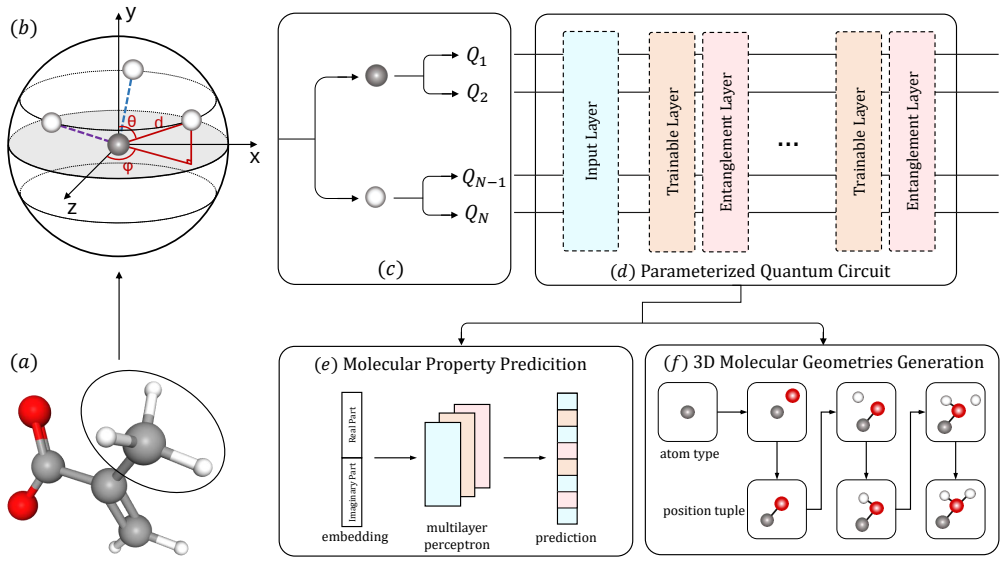

Figure 1: **The quantum 3D embedding scheme.** **(a)** The 3D molecular graph with the gray node (in the black circle) is picked as the focal atom and three white nodes within the distance threshold as the neighbors. **(b)** We convert the 3D Cartesian coordinates of the atoms into the relative position tuple $(d, \boldsymbol{\theta}, \boldsymbol{\varphi})$. **(c)** We encode the position tuple as well as the atom type into two qubits for each atom. **(d)** The PQC for our model, the input layer includes $\mathbf{R_x}$ and $\mathbf{R_y}$ on each qubit, which encodes the up mentioned data. Trainable layers with parameters $\boldsymbol{\theta}$s and entanglement layers are applied alternately to analog the classical machine learning layers. **(e)** The task of property prediction. We use the embeddings from the PQC to predict chemical properties and compare them with the labels. **(f)** The task of 3D molecular geometries generation. We generate a molecule from scratch based on autogressive flow model with picking one focal atom and then deciding the relative position.

as an example. Let $\mathcal{G}$ denotes the graph of a certain molecule and $\mathcal{V}$ denotes the node set of graph $\mathcal{G}$. The number of nodes (in other words atoms) is $n = |\mathcal{V}|$. Each node $v_i \in \mathcal{V}$ has an attribute a_i, which is the atom type in our setting. Our target is to learn the embedding for each atom and then obtain the final embedding for the molecule. The embeddings are then tested on different molecular tasks (e.g. molecular property prediction, 3D molecular geometries generation, etc.).

Method Overview. We develop a quantum machine learning approach to learn the embedding on 3D graph. The trainable parameter refers to the $\boldsymbol{\theta}$ in those rotation gates in the PQC.

Specifically, we first encode the 3D coordinates and the atom types into qubits. We use relative coordinates instead of the 3D Cartesian coordinates in order to ensure both equivariance and invariance. The relative coordinates can be written in the form of a position tuple $(d, \boldsymbol{\theta}, \boldsymbol{\varphi})$, where $d, \boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ denote the radial distance, polar angle, and the azimuthal angle, respectively. We set up a distance threshold to pick the neighbors which can interact with the focal atom. A PQC is then used to learn the latent variables and entangle the qubits together. We further apply a tomography at the end of the PQC and then concatenate the real part and the imaginary part. The overall pipeline of our proposed method is shown in Fig. 1.

3.2 THE PROPOSED ATOM2QUBIT

Considering a molecule with n atoms, we take it mathematically as a graph $\mathcal{G}$ with n nodes. For each node v_i, we have a corresponding attribute a_i, which denotes the atom type and a 3D Cartesian coordinate set $\{x_i, y_i, z_i\}$. Without loss of generality, we first pick v_i as the focal atom and learn the embedding of node v_i. The distance between v_i and other nodes $v_j \in \mathcal{V}$ is $d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$. Note that not all of the node pairs in the graph have interaction in the pairs, we set a maximum distance threshold d_{max} as a hyperparameter. So that $v_j \in \mathcal{N}(v_i)$, if $i \neq j$ and $d_{ij} \leq d_{max}$, which means only the nodes v_j with $d_{ij} \leq d_{max}$ are consid-

ered as the neighbors of v_i. We then need to convert the 3D Cartesian coordinates of $v_j \in \mathcal{N}(v_i)$ into the position tuple $(d_{ij}, \boldsymbol{\theta}_{ij}, \boldsymbol{\varphi}_{ij})$. The definition of rotation angle $\boldsymbol{\theta}$ and torsion angle $\boldsymbol{\varphi}$ are shown in Fig. 1 (b). Now each node $v_j \in \mathcal{N}(v_i)$ can be uniquely defined by $\{a_j, d_{ij}, \boldsymbol{\theta}_{ij}, \boldsymbol{\varphi}_{ij}\}$.

When we encode classical information into the quantum form, we have two different ways. The first one is amplitude encoding and the second one is angle encoding. The amplitude encoding can encode a classical one-hot vector of dimension n with only $\log_2(n)$ qubits, but it is quite hard to encode continuous variables while requires $\mathcal{O}(n)$ times to encode the information. On the contrary, the angle encoding requires a minimum of $n/3$ qubits to encode n classical information, but it is capable of encoding both discrete and continuous variables. Furthermore, the angle encoding is a better fit for the rotation parameters in the circuit. In this paper, we pick angle encoding as our way to encode the information set $\{a_j, d_{ij}, \boldsymbol{\theta}_{ij}, \boldsymbol{\varphi}_{ij}\}$ into qubits.

For each qubit, we have three rotation operators $\mathbf{R_x}$, $\mathbf{R_y}$ and $\mathbf{R_z}$. We can theoretically encode three different pieces of information on one qubit. However, if we consider the qubit on a Bloch sphere, we can uniquely define the rotation track on the Bloch sphere using only two rotation operators. To avoid the decomposition of the third input, we only use two of the rotation operators $\mathbf{R_x}$ and $\mathbf{R_y}$ in this paper ($\mathbf{R_z}$ does not change the outputs of our measurement method). Therefore, we need two qubits $|\Psi_1\rangle$ and $|\Psi_2\rangle$ to encode each node v_j,

$$|\Psi_1\rangle = \boldsymbol{U}_{\mathbf{x}}(\boldsymbol{\theta}_{ij}) \times \boldsymbol{U}_{\mathbf{y}}(\boldsymbol{\varphi}_{ij}) \times |0\rangle \tag{3}$$

$$|\Psi_2\rangle = \boldsymbol{U}_{\mathbf{x}}(\frac{d_{ij}}{d_{max}} \times 2\pi) \times \boldsymbol{U}_{\mathbf{y}}(\frac{a_j}{a_{num}} \times 2\pi) \times |0\rangle \tag{4}$$

where a_{num} denotes the number of atoms occurred in the dataset and a_j is an integer $\in [1, a_{num}]$. $|\Psi_1\rangle \otimes |\Psi_2\rangle$ is the quantum encoding state of one node generated from initial state $|00\rangle$. If $n = |v_i \cup \mathcal{N}(v_i)|$, the initial state $|\Psi^0\rangle$ for the PQC in Sec. 3.3 after the atom2qubit encoding stage is

$$|\Psi^0\rangle = |\Psi_1\rangle \otimes |\Psi_2\rangle \otimes \cdots \otimes |\Psi_{2n-1}\rangle \otimes |\Psi_{2n}\rangle \tag{5}$$

3.3 Quantum 3D embedding ansatz

We first discuss the number of qubits we need for our approach on molecule problems. Each time we learn the embedding of node v_i, we need to encode the information of $v_i \cup \mathcal{N}(v_i)$ into qubits. Therefore the qubit number is linear with the size of $\mathcal{N}(v_i)$. The interaction between atoms in a molecule is bounded by the bond length between atoms. As the bond length increases, the interaction becomes much weaker, which means we barely have multi-hop message passing in our graph. This gives us the possibility to run the test on an existing near-term quantum device. Therefore, we choose hardware-efficient ansatz that has been proved on a superconducting quantum processor with six fixed-frequency transmon qubits by (Kandala et al., 2017) and a 56-bit superconducting quantum processor *Zuchongzhi* by (Huang et al., 2021).

Analog to classical neural network models, the PQC is constructed by layers and each layer has an identical arrangement of quantum gates. Fig. 2 illustrates the general framework of the quantum 3D embedding ansatz. The overall unitary $\boldsymbol{U}(\boldsymbol{\theta}) = \Pi_{l=1}^{L}(\boldsymbol{U}_{ent}\boldsymbol{U}_l(\boldsymbol{\theta}))$, where $\boldsymbol{U}_{ent}$ is the entanglement layer and $\boldsymbol{U}_l(\boldsymbol{\theta})$ is the l-th trainable layer. In particular, we have the l-th trainable layer $\boldsymbol{U}_l(\boldsymbol{\theta}) = \bigotimes_{k=1}^{N}(\boldsymbol{U}_{\mathbf{y}}(\boldsymbol{\theta}_{\mathbf{y}}^{(k,l)})) \times \bigotimes_{k=1}^{N}(\boldsymbol{U}_{\mathbf{x}}(\boldsymbol{\theta}_{\mathbf{x}}^{(k,l)}))$, where $\boldsymbol{U}_{\mathbf{x}}$ is the unitary of gate $\mathbf{R_x}$ and $\boldsymbol{\theta}_{\mathbf{x}}^{(k,l)}$ is the parameter for $\mathbf{R_x}$ at the l-th layer on the k-th qubit. The entanglement layer $\boldsymbol{U}_{ent}$ consists of CNOT gates and it entangles all the qubits together shown in Fig. 2. The quantum state $|\Psi^l\rangle$ after l layers is

$$|\Psi^l\rangle = \boldsymbol{U}_{ent} \times \boldsymbol{U}_l \times (\boldsymbol{U}_{ent} \times \boldsymbol{U}_{l-1} \times (\cdots(\boldsymbol{U}_{ent} \times \boldsymbol{U}_1|\Psi^0)))) \tag{6}$$

$$= \boldsymbol{U}_{ent} \times \bigotimes_{k=1}^{N}(\boldsymbol{U}_{\mathbf{y}}(\boldsymbol{\theta}_{\mathbf{y}}^{(k,l)})\boldsymbol{U}_{\mathbf{x}}(\boldsymbol{\theta}_{\mathbf{x}}^{(k,l)})) \times (\cdots(\boldsymbol{U}_{ent} \times \bigotimes_{k=1}^{N}(\boldsymbol{U}_{\mathbf{y}}(\boldsymbol{\theta}_{\mathbf{y}}^{(k,1)})\boldsymbol{U}_{\mathbf{x}}(\boldsymbol{\theta}_{\mathbf{x}}^{(k,1)}))|\Psi^0)))) \tag{7}$$

The quantum state $|\Psi^0\rangle$ is the initial state, which is also the output of the Atom2Qubit stage. With the parameters $\boldsymbol{\theta}_{\mathbf{x}}^{(k,l)}$ and $\boldsymbol{\theta}_{\mathbf{y}}^{(k,l)}$, we can learn the latent representation of each node. Note that the model we proposed is a graph representation learning model, thus we need to further attach downstream tasks to test the efficiency of our model, and the loss function is also obtained from the

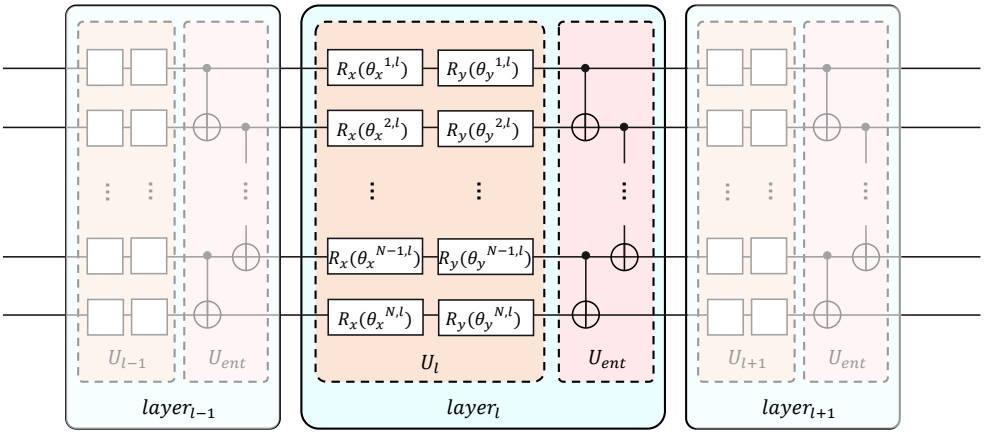

Figure 2: The circuit for our quantum 3D embedding ansatz. Each layer includes trainable parameters block $\boldsymbol{U}_l$ and entanglement block $\boldsymbol{U}_{ent}$. We have N qubits in the circuit so there are $2 \times N$ parameters in each layer. The entanglement layer are composed of CNOT gates to pairwisely entangle all the N qubits.

downstream model. The loss function $\mathcal{L}$ which is employed to optimize the trainable parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathbf{x}}^{(k,l)} \odot \boldsymbol{\theta}_{\mathbf{y}}^{(k,l)}$, where $\odot$ is concatenation, for our model $\boldsymbol{M}$ yields

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{M}_{\boldsymbol{\theta}}(|\Psi^0\rangle)) \tag{8}$$

The parameters $\boldsymbol{\theta}$ are then updated at each iteration using the derivative in Eq. 1 and the gradient update method is shown in Eq. 2.

3.4 Atom embedding

The quantum circuit we mentioned above is a N qubit circuit and it works in a 2^N dimensional Hilbert space. We apply a tomography at the end of the circuit thus we can get a 2^N dimensional vector with a complex number α_i for each dimension . A quantum state $|\psi\rangle$ can be written in the form of a combination of the computational basis states,

$$|\psi\rangle = \alpha_1 \underbrace{|0\cdots00\rangle}_{2^N} + \alpha_2 \underbrace{|0\cdots01\rangle}_{2^N} + \alpha_3 \underbrace{|0\cdots10\rangle}_{2^N} + \cdots + \alpha_{2^N} \underbrace{|1\cdots11\rangle}_{2^N} \tag{9}$$

where $\alpha_i, 1 \leq i \leq 2^N$ are the complex coefficients and the vector $(\alpha_1, \alpha_2, \cdots, \alpha_{2^N})$ is the result of the tomography. Each α_i can be written in the form of $\alpha_i = \mathrm{Re}(\alpha_i) + \mathrm{Im}(\alpha_i) \cdot \mathrm{i}$, where Re denotes the real part, Im denotes the imaginary part and i is the imaginary unit. We then concatenate the real part and the imaginary part of the tomography and get the node embedding vector $(\mathrm{Re}\,\alpha_1, \mathrm{Re}\,\alpha_2, \cdots, \mathrm{Re}\,\alpha_{2^N}, \mathrm{Im}\,\alpha_1, \mathrm{Im}\,\alpha_2, \cdots, \mathrm{Im}\,\alpha_{2^N})$ with the dimension of 2^{N+1}.

4 Numerical Experiments

All the experiments are performed on a workstation with a single machine with 1TB memory, one physical CPU with 28 cores Intel(R) Xeon(R) W-3175X CPU @ 3.10GHz), and two GPUs (Nvidia Quadro RTX 8000). The source code is written using the framework of PyTorch, where we simulate the whole quantum circuit process using transition unitary. We have also implemented a Qiskit version of our model, and it is much slower while achieving the same results. On QM9-pred, the average training time for each epoch is over 2 hours using QASM simulator on Qiskit and it takes an average of 310s for our simulator, which is about 23 times faster. Note that all our models are not implemented on quantum hardware yet, but the model and the circuit we proposed are easy to adapt to NISQ devices. To test the performance of our embedding model, we perform numerical experiments on two different tasks and compare the results with state-of-the-art classical 3D molecular representation learning models.

Dataset. The benchmark dataset we used is QM9 (Ramakrishnan et al., 2014), which is widely used for predicting various properties of molecules and 3D molecules generating tasks. It includes quantum chemistry structures and properties of up to 134k stable small organic molecules. These molecules consists of up to 9 heavy atoms CONF, not counting hydrogen, and their corresponding 3D molecular geometries are computed by density functional theory (DFT).

Table 1: Statistics of dataset

Datasets	QM9-pred	QM9-gen
#Max Nodes	10	10
#Nodes/Graph	9.73	9.39
#Edges/Graph	9.37	9.16

4.1 Implementation details

In Sec. 3.2 we have shown how to convert the information of each atom into rotation angles on qubits. Now we discuss more precisely how to calculate the position tuple $(d, \boldsymbol{\theta}, \boldsymbol{\varphi})$. The 3D coordinates for atom v_i in QM9 is three real numbers x_i, y_i and z_i. If we pick v_i as the focal atom, $\forall v_j \in \mathcal{N}(v_i)$, we need to calculate the position tuple for atom v_j against v_i.

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \tag{10}$$

$$\boldsymbol{\theta}_{ij} = \arctan\left(\frac{\sqrt{(x_j - x_i)^2 + (z_j - z_i)^2}}{y_j - y_i}\right) \tag{11}$$

$$\boldsymbol{\varphi}_{ij} = \arctan \frac{x_j - x_i}{z_j - z_i} \tag{12}$$

In order to fit the definition of rotation angle and torsion angle, those angles should fit into the domain $\boldsymbol{\theta}_{ij} \in [0, \pi]$ and $\boldsymbol{\varphi}_{ij} \in [0, 2\pi)$. We need to adjust the results in Eq. 11 and Eq. 12.

$$\boldsymbol{\theta}_{ij} \leftarrow \boldsymbol{\theta}_{ij} + \pi, \quad \text{if } y_j - y_i < 0 \tag{13}$$

$$\boldsymbol{\theta}_{ij} \leftarrow 0, \quad \text{if } y_j - y_i = 0 \tag{14}$$

$$\boldsymbol{\varphi}_{ij} \leftarrow \boldsymbol{\varphi}_{ij} + \pi, \quad \text{if } z_j - z_i < 0 \wedge (x_j - x_i < 0 \vee x_j - x_i > 0) \tag{15}$$

$$\boldsymbol{\varphi}_{ij} \leftarrow \boldsymbol{\varphi}_{ij} + 2\pi, \quad \text{if } z_j - z_i > 0 \wedge x_j - x_i < 0 \tag{16}$$

For the hyperparameters, we set d_{max} as 1.77, which is the numerical value of the maximum bond length in our dataset, and a_{num} is set to be 6 since we have five different types of atom in our dataset as well as a null type.

Now we discuss more details of the transition unitary based full-amplitude circuit simulation on a classical processor. The quantum gates we used in our PQC are only $\mathbf{R_x}(\boldsymbol{\theta})$, $\mathbf{R_y}(\boldsymbol{\theta})$ and CNOT. The matrix representations of the single-qubit gates are as follows

$$\mathbf{R_x}(\boldsymbol{\theta}) = \begin{pmatrix} \cos(\frac{\boldsymbol{\theta}}{2}) & -\mathrm{i}\sin(\frac{\boldsymbol{\theta}}{2}) \\ -\mathrm{i}\sin(\frac{\boldsymbol{\theta}}{2}) & \cos(\frac{\boldsymbol{\theta}}{2}) \end{pmatrix}, \quad \mathbf{R_y}(\boldsymbol{\theta}) = \begin{pmatrix} \cos(\frac{\boldsymbol{\theta}}{2}) & -\sin(\frac{\boldsymbol{\theta}}{2}) \\ \sin(\frac{\boldsymbol{\theta}}{2}) & \cos(\frac{\boldsymbol{\theta}}{2}) \end{pmatrix} \tag{17}$$

The two-qubit gate unitary matrix is as follows

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{18}$$

With these matrices of the basic gates, we obtain the unitary of a circuit block. We first divide the block by layers, and each layer has at most one gate for each qubit. We tensor the gate matrix within each layer and thus we get the unitary for the layer. We then use matrix multiply between different layers, so the final unitary is calculated through torch.tensor() and then torch.matmul(). For a N qubit circuit, the overall unitary $U \in \mathbb{C}^{2^N \times 2^N}$. The maximum number of qubits we need is 10, so the largest unitary we need is in $\mathbb{C}^{1024 \times 1024}$, which is still affordable on a classical processor.

4.2 Molecular property prediction

We first conduct experiments on the task of molecular property prediction to evaluate our embedding model. The downstream model we used is a simple multilayer perceptron predictor, which can perform linear regression on the embeddings from the embedding model.

Table 2: Performance comparison between the baselines and our proposed method on QM9-pred in terms of MAE for three properties (ϵ_{HOMO}, ϵ_{LUMO} and $\Delta\epsilon$) and the std. MAE for all three properties. We use the unit eV for these three energy-related properties.

Property	Unit	**SchNet** (Schütt et al., 2017)	**DimeNet++** (Klicpera et al., 2020)	**SphereNet** (Liu et al., 2021)	**ComENet** (Wang et al., 2022)	**Ours**
ϵ_{HOMO}	eV	0.683	0.427	**0.349**	0.504	0.419
ϵ_{LUMO}	eV	0.605	0.451	**0.287**	0.538	0.451
$\Delta\epsilon$	eV	0.704	0.576	**0.403**	0.665	0.486
std.MAE	%	65.2	46.4	**34.0**	54.5	43.5

Setting. We filter the QM9 to generate the dataset for our prediction task. Our model suffers from the extremely high time cost of simulating the quantum circuit, so it is not possible to run on the whole 134k molecules in the QM9. We sieve the dataset with molecules no more than 10 atoms, and randomly pick 500 of them to form our dataset for the prediction task. We denote the dataset as QM-pred, and statistics of it are listed in Table 1. We split the dataset into training/validation/test sets with a ratio of 8:1:1. Training molecules are used to optimize the model parameters. The validation molecules are used to fine-tune the hyper-parameters as well as conduct the early stopping, and then we report the results on test molecules. Among all sixteen properties listed in QM9, we selected three important energy-related properties, namely ϵ_{HOMO}, ϵ_{LUMO} and $\Delta\epsilon$. $\Delta\epsilon$, also known as the HOMO-LUMO gap, is one of the most practically-relevant quantum chemical properties of molecules (Bredas, 2014). In line with (Liu et al., 2021), we report the mean absolute error (MAE) for each property as well as the overall mean standardized MAE (std. MAE) for all these three properties.

Baselines. To the best of our knowledge, there are no other quantum models considering representation learning for 3D graphs, thus we compare our method with four baselines in the classical domain: the seminal work in this area SchNet (Schütt et al., 2017), DimeNet++ (Klicpera et al., 2020), SphereNet (Liu et al., 2021) and ComENet (Wang et al., 2022).

Prediction model. The embeddings obtained from our model are fed to a simple predictor, which is a multilayer perceptron reducing the size of the embedding from 2^N to 1. We use stochastic gradient descent (SGD) with Adam optimizer (Kingma & Ba, 2014) to train our model for a maximum of 100 epochs with a batch size of 32 and a learning rate of 0.01. Meanwhile, as the running time will increase dramatically when the number of trainable layers increases, we set the number of layers in the PQC as four to balance the training time and accuracy.

Results. The numerical results of the property prediction task are presented in Table 2. We achieved second best on all three properties. SphereNet is 21.8% better than us on the overall mean standardized MAE and is 17.1% better than us on the HOMO-LUMO gap. However, SphereNet used $1,898,566$ parameters and we only used $102,881$ parameters (64 of them are from the PQC). Compared to the fundamental baseline, we are 33.3% better than SchNet on overall mean standardized MAE and 31% on HOMO-LUMO gap. Notice that SchNet still used $455,809$ parameters. As shown in Figure 3, our model converges very fast in the first few epoches, which also demonstrates the efficiency of our model.

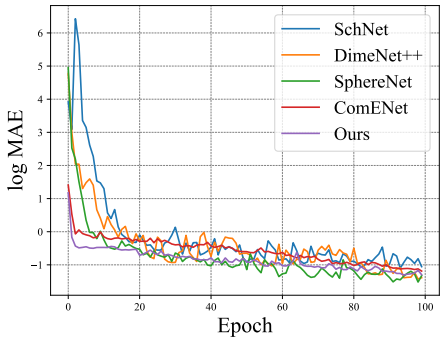

Figure 3: Training loss on the prediction task.

4.3 3D Molecular Geometries Generation

This study evaluates the performance of our proposed embedding model when adapted to the existing random molecular geometry generation method. To be more specific, the embeddings from our model are used to extract 3D conditional information in the generation process.

Table 3: Performance comparison between the G-SphereNet and our proposed method on 500 randomly generated molecules for chemical validity percentage and MMD distances of bond length distributions. Here ↑ means higher value indicates better performance, while ↓ means the opposite.

Method	Validity ↑	MMD distances ↓						
		H-C	H-N	H-O	C-C	C-N	C-O	Average
G-SphereNet	**68.55%**	**0.161**	**0.280**	1.104	0.399	0.438	**0.277**	0.443
Ours	67.00%	0.237	0.409	**0.770**	**0.326**	**0.407**	0.378	**0.421**

Setting. For the molecular geometry generation task, we also use filtered QM9 as the evaluation dataset. Different from QM-pred, we select 806 molecules that contain no more than 10 atoms to form our dataset, 50 of them are used for validation and the remaining are used for training. We entitled this filtered QM9 as QM9-gen and the statistics of QM9-gen is presented in Table 1. The generated molecular geometries can be converted to molecular graphs according to the approach proposed in (Gebauer et al., 2019). As for metrics, we use the chemical validity percentage (Validity) which is defined as the percentage of molecular graphs that obey the chemical valency rules to evaluate the generation accuracy. In addition, we adopt Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) distances of bond length distributions to evaluate the 3D structural accuracy of the generated molecular geometries. In specific, we calculate the length distribution in the generated geometries and in the dataset geometries separately for each type of bond, then we can obtain the statistical discrepancy between them with the MMD distance. In line with (Luo & Ji, 2022), we compute the MMD on hydrogen-carbon single bonds (**H-C**), hydrogen-nitrogen single bonds (**H-N**), hydrogen-oxygen single bonds (**H-O**), carbon-carbon single bonds (**C-C**), carbon-nitrogen single bonds (**C-N**), carbon-oxygen single bonds (**C-O**) these six types of chemical bonds respectively as they are most frequently appeared.

Baseline. We use G-SphereNet (Luo & Ji, 2022) as the baseline in this molecular geometries generation task. We select G-SphereNet (also from ICLR 22) produced by the same group as SphereNet, which uses SphereNet as the embedding model to extract 3D conditional information.

Generation Model. As for generation model, we employ the same generation pipeline as G-SphereNet, which adopts a flexible sequential generation strategy by adding atoms in 3D space one by one based on autoregressive flow models. We use Adam optimizer to train the our model for 100 epochs, with a batch size of 64 and a learning rate of 0.001. Also, we set the maximum number of atoms that can be generated for each molecule as 13.

Results. We present the performance of our model against G-SphereNet in Table 3. We reach comparable results with baseline model on QM-gen. More specifically, we slightly outperform the baseline model on MMD distances for 3 types of bond length, which shows that our method have a strong capability of extracting the 3D conditional information of molecular geometries.

5 CONCLUSION

3D information is important for graphs such as molecules in quantum chemistry and learning the 3D representation for those graphs has attracted increasing attention. Existing classical models face the inherent challenge of understanding the physical meaning of the 3D Cartesian coordinates. To our best knowledge, we are the first to use qubits to encode 3D spatial information and use a Parameterized Quantum Circuit (PQC) to learn the representation of each node as the embedding. The experiments on two well-studied downstream tasks demonstrate the efficiency and capability of our model, and the potential to execute on real quantum devices.

Limitation & future works. Our method is limited by the time consumption when simulating quantum circuits, while superconducting NISQ device is entering the 50+ qubit era (Gong et al., 2021), which gives us the confidence to test our model on one of them. Meanwhile, the noise on the gates are not fatal with such shallow circuits. But we will need to adjust the readout procedure of our embedding when testing on NISQ device. We are looking forward to extending our experiments to 10 thousand molecules and reaching the chemical accuracy of 1.6×10^{-3} Hartree.

REFERENCES

Xing Ai, Zhihong Zhang, Luzhe Sun, Junchi Yan, and Edwin Hancock. Decompositional quantum graph neural network. *arXiv preprint arXiv:2201.05158*, 2022. 3

Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. 1

Alán Aspuru-Guzik, Anthony D Dutoi, Peter J Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005. 1

Lu Bai, Yuhang Jiao, Lixin Cui, Luca Rossi, Yue Wang, Philip Yu, and Edwin Hancock. Learning graph convolutional networks based on quantum vertex information propagation. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 3

Johannes Bausch. Recurrent quantum neural networks. *Advances in neural information processing systems*, 33:1368–1379, 2020. 3

Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017. 1

Jean-Luc Bredas. Mind the gap! *Mater. Horiz.*, 1:17–19, 2014. 8

Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021. 3

Samuel Yen-Chi Chen, Tzu-Chieh Wei, Chao Zhang, Haiwang Yu, and Shinjae Yoo. Hybrid quantum-classical graph convolutional network. *arXiv preprint arXiv:2101.06189*, 2021. 3

Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang. Quantum long short-term memory. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8622–8626, 2022. 3

Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019. 3

Stefan Dernbach, Arman Mohseni-Kabir, Siddharth Pal, and Don Towsley. Quantum walk neural networks for graph-structured data. In *Complex Networks and Their Applications*. Springer, 2018. 3

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019. 1

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021. 1

Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy BR Hayter, Richard Vickers, Charles Roberts, Jian Tang, et al. Utilizing graph machine learning within drug discovery and development. *Briefings in bioinformatics*, 22(6):bbab159, 2021. 1

Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32, 2019. 9

Ming Gong, Shiyu Wang, Chen Zha, Ming-Cheng Chen, He-Liang Huang, Yulin Wu, Qingling Zhu, Youwei Zhao, Shaowei Li, Shaojun Guo, et al. Quantum walks on a programmable two-dimensional 62-qubit superconducting processor. *Science*, 372(6545):948–952, 2021. 9

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 9

He-Liang Huang, Dachao Wu, Daojin Fan, and Xiaobo Zhu. Superconducting quantum computing: a review. *Science China Information Sciences*, 63(8):1–32, 2020. 1

He-Liang Huang, Yuxuan Du, Ming Gong, Youwei Zhao, Yulin Wu, Chaoyue Wang, Shaowei Li, Futian Liang, Jin Lin, Yu Xu, Rui Yang, Tongliang Liu, Min-Hsiu Hsieh, Hui Deng, Hao Rong, Cheng-Zhi Peng, Chao-Yang Lu, Yu-Ao Chen, Dacheng Tao, Xiaobo Zhu, and Jian-Wei Pan. Experimental quantum generative adversarial networks for image generation. *Physical Review Applied*, 16(2), 2021. 2, 3, 5

Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, sep 2017. 2, 5

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

Johannes Klicpera, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020. 8

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2021. 1, 8

Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International Conference on Learning Representations*, 2022. 9

K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Phys. Rev. A*, 98: 032309, Sep 2018. 2, 3

Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002. 2

Peter JJ O'Malley, Ryan Babbush, Ian D Kivlichan, Jonathan Romero, Jarrod R McClean, Rami Barends, Julian Kelly, Pedram Roushan, Andrew Tranter, Nan Ding, et al. Scalable quantum simulation of molecular energies. *Physical Review X*, 6(3):031007, 2016. 1

John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018. 1

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014. 7

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017. 1, 8

Oliver T Unke and Markus Meuwly. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15(6):3678–3693, 2019. 1

Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. ComENet: Towards complete and efficient message passing for 3d molecular graphs. *arXiv e-prints*, pp. arXiv–2206, 2022. 8

Zhihong Zhang, Dongdong Chen, Jianjia Wang, Lu Bai, and Edwin R Hancock. Quantum-based subgraph convolutional neural networks. *Pattern Recognition*, 88:38–49, 2019. 3

Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020. 1