Quantum 3D Graph Learning with Applications to Molecule Embedding

Anonymous Authors[1]

Abstract

Learning 3D graph with spatial position as well as node attributes has been recently actively studied, aiming at better understanding the physical meaning of the 3D topology information. Quantum computing is known a promising direction for its potential theoretical supremacy for large-scale graph and combinatorial problem as well as the increasing evidence for the availability to physical quantum devices in the near term. In this paper, for the first time to our best knowledge, 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. Specifically, the 3D Cartesian coordinates of nodes are converted 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 the output of the PQC is adopted as the final node embedding. Experimental results on two downstream tasks, molecular property prediction and 3D molecular geometries generation, demonstrate the effectiveness of our model. We show the capacity and capability of our model with the evaluation on the QM9 dataset (134k molecules) with very few parameters, and its potential to be executed 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 last decade. 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 leads to the possibility to implement 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 areas for quantum computing, as termed by 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 effective prediction of chemical reaction rates. However, these quantum approaches (Romero et al., 2018; Peruzzo et al., 2014; O'Malley et al., 2016; Yung et al., 2014) are still simulating the energies of certain small molecules like H_2, LiH, etc.

In this paper, we aim to develop a full quantum algorithm, and shed lights on quantum machine learning approaches solving molecular problems instead of showing supremacy over the classical molecule learning approaches. The proposed method is totally different from the mainstream quantum approach Unitary Coupled Cluster (UCC), which is a unsupervised learning approach with uniquely designed circuit for each molecule. Graph learning may not be as precise as molecular simulation approaches for property prediction, but they have the ability to learn hundreds or thousands of molecules and predict the properties for more complex molecules.

Specifically, we first convert the 3D Cartesian coordinates of the atoms into three geometries: distance, rotation angle, and torsion angle. Then we encode the angles and distance as well as the atom type (a discrete variable), into qubits. A distance threshold is used so that each time a focal atom is picked to learn the embedding, one only need to consider

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the neighboring atoms within the threshold, which forms a group of atoms similar to the definition of moiety. Limited by the computational power to simulate the quantum circuits, we are only able to simulate a circuit of 16 qubits, which means we can have a maximum of 7 neighbors for each focal to learn the representation. 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 gradient backpropagation analog to classical neural networks. We apply a Pauli-Z measurement at the end of the circuit and then take it as the node embedding. We conducted numerical experiments on the 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 with much fewer network parameters and much faster convergence rate. **Our contributions are as follows:**

1) To the best of our knowledge, we are the first to use qubits to encode 3D relative positional information, which aims to effectively preserve the property of equivariance and invariance. In fact, 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 not only in molecules but also point clouds in recent studies.

2) The proposed method is fully capable of being executed on a NISQ device. We are using angle encoding for state preparation, hardware efficient ansatz to learn the embedding, Pauli-Z measurement to obtain the final embedding, and parameter shift rule to update the gradients. Moreover, we use two qubits to represent each atom, and we only consider the focal atom and its neighbors at each iteration, which makes the proposed method also very efficient in the usage of qubits (only 16 qubits are needed by now).

3) 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 (384 parameters in the PQC) and can achieve relatively good results. The fact that we are able to process all 134k molecules from QM9 demonstrates the capacity of our model.

2. Preliminaries and related works

We first briefly review basic concepts of quantum computing as well as quantum machine learning. We further present previous works on quantum graph learning.

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. Here we give a compact description of background.

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 $\mathbf{U}$ on Hilbert space $\mathcal{H}$. When we simulate the quantum circuit on a classical computer, we can obtain the overall unitary transformation 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, $M = \sum_m m\mathbf{P}_m$, where $\mathbf{P}_m$ 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 $p(m) = \langle\psi|\mathbf{P}_m|\psi\rangle$.

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 $\boldsymbol{\theta}(i)-\pi/2$

$$\begin{aligned}\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}(i)} =& \frac{1}{2}\times\Big(\mathcal{L}\Big(\boldsymbol{\theta}(1),\cdots,\boldsymbol{\theta}(i)+\frac{\pi}{2},\cdots\Big) \\ & -\mathcal{L}\Big(\boldsymbol{\theta}(1),\cdots,\boldsymbol{\theta}(i)-\frac{\pi}{2},\cdots\Big)\Big) \qquad (1)\end{aligned}$$

Also using gradient backpropagation, 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, which yet show that the quantum counterparts on NISQ device may not be as powerful as the SOTA classical ones (usually with millions of parameters). Involving quantum computing is an interesting experiment to seek potential supremacy and the connection between latent space and the mystery quantum entanglement.

2.3. Unitary Coupled-cluster

One of the most promising area to demonstrate the quantum computing supremacy is quantum chemistry. There have been continuous work in this research area and the mainstream of these work is Unitary Coupled-Cluster (UCC) (Romero et al., 2018; Peruzzo et al., 2014; O'Malley et al., 2016; Yung et al., 2014). UCC focused on solving the time-independent Schrödinger equation for molecular system to predict the chemical properties. The coupled-cluster theory is used to obtain the Hamiltonian of a certain molecule and then use Trotter-Suzuki decomposition to approximate the Hamiltonian on a quantum circuit. The parameters in the rotation gates allow us to train for the minimal ground-state energy. This method provides a hierarchy of wave functions that can be prepared on a quantum computer. It is believed that UCC can provide better accuracy than classical coupled cluster (Wierschke, 1994; Hoffmann & Simons, 1988; Bartlett et al., 1989), which is also regarded as the ”gold standard” of quantum chemistry (Bartlett & Musiał, 2007). However, UCC is an unsupervised learning method with no ground truth and can only evolve one molecule at a time since the circuit is uniquely designed for a certain molecule. There are also evidence showing that the number of parameters in UCC might be still too large to allow practical calculations for large molecules.

2.4. Quantum Graph Learning

Different from evolving Hamiltonian and solving the Schrödinger equation with the quantum circuit, we also have quantum graph learning approaches trying to learn the latent representation of the vertex and the graph. 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 encoding strategy is not carefully designed. (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, which lead us to question whether the quantum layer is necessary. Thus, we propose a full quantum paradigm with quantum friendly encoding specially designed for molecular problems.

3. Methodology

3.1. Problem Setting and Approach 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 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 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 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

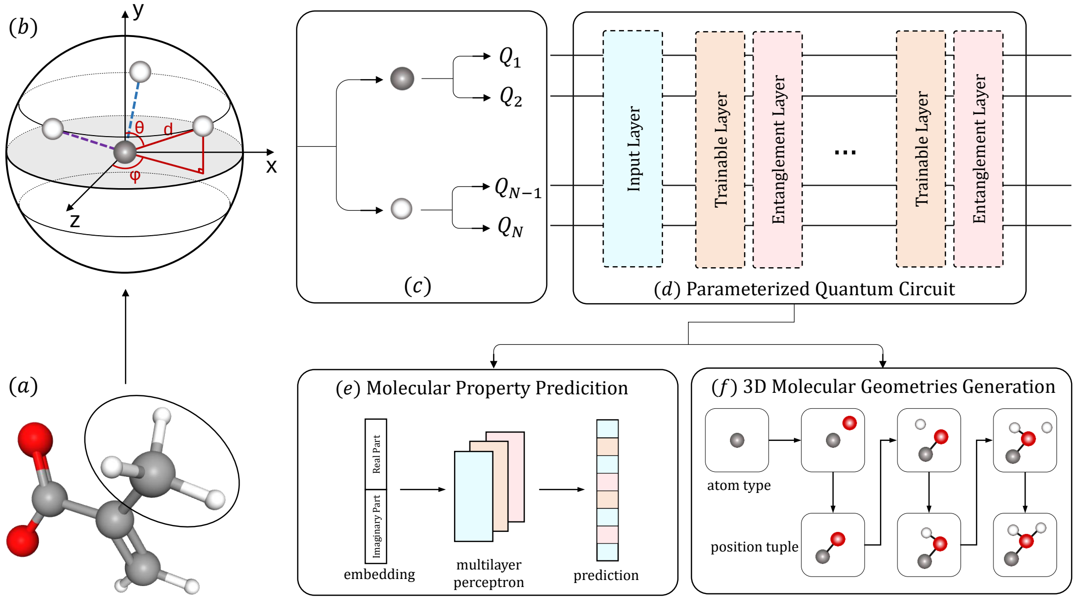

Figure 1. **The quantum 3D embedding scheme.** **(a)** The 3D molecular graph with the gray node (in the black circle) is picked as the focal atom and three white nodes within the distance threshold as the neighbors. **(b)** We convert the 3D Cartesian coordinates of the atoms into the relative position 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.

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 considered 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 calculation of rotation angle $\boldsymbol{\theta}$ and torsion angle $\boldsymbol{\varphi}$ are using the spherical coordinate system as stated in (Liu et al., 2021) to ensure the invariance and equivariance of the molecules. 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 dimensions of 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,

$$|\Psi_1\rangle = \mathbf{U_x}(\boldsymbol{\theta}_{ij}) \times \mathbf{U_y}(\boldsymbol{\varphi}_{ij}) \times |0\rangle \tag{2}$$

$$|\Psi_2\rangle = \mathbf{U_x}(\frac{d_{ij}}{d_{max}} \times 2\pi) \times \mathbf{U_y}(\frac{a_j}{a_{num}} \times 2\pi) \times |0\rangle \tag{3}$$

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

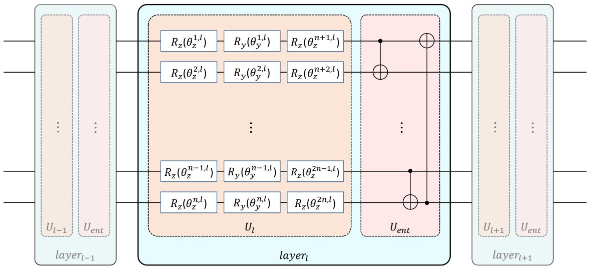

Figure 2. The circuit for our quantum 3D embedding ansatz. Each layer includes trainable parameters block $\mathbf{U}_l$ and entanglement block $\mathbf{U}_{ent}$. We have N qubits in the circuit so there are $3 \times N$ parameters in each layer. The entanglement layer is composed of CNOT gates to pairwisely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

PQC in Sec. 3.3 after the Atom2Qubit encoding stage is

$$|\Psi^0\rangle = |\Psi_1\rangle \otimes |\Psi_2\rangle \otimes \cdots \otimes |\Psi_{2n-1}\rangle \otimes |\Psi_{2n}\rangle \tag{4}$$

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. 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 66-bit superconducting quantum processor *Zuchongzhi* by (Huang et al., 2021).

Analog to classical neural networks, the PQC is constructed by layers and each layer has an identical arrangement of quantum gates. Fig. 2 shows the general framework of the quantum 3D embedding ansatz. The overall unitary is

$$\mathbf{U}(\boldsymbol{\theta}) = \prod_{l=1}^{L} \Big(\mathbf{U}_{ent} \mathbf{U}_l(\boldsymbol{\theta}) \Big) \tag{5}$$

where $\mathbf{U}_{ent}$ is the entanglement layer and $\mathbf{U}_l(\boldsymbol{\theta})$ is the l-th trainable layer. In particular, we have the l-th trainable layer

$$\begin{aligned} \mathbf{U}_l(\boldsymbol{\theta}) = & \bigotimes_{k=N+1}^{2N} \Big(\mathbf{U}_\mathbf{z}(\boldsymbol{\theta}_\mathbf{z}^{(k,l)}) \Big) \times \bigotimes_{k=1}^{N} \Big(\mathbf{U}_\mathbf{y}(\boldsymbol{\theta}_\mathbf{y}^{(k,l)}) \Big) \\ & \times \bigotimes_{k=1}^{N} \Big(\mathbf{U}_\mathbf{z}(\boldsymbol{\theta}_\mathbf{z}^{(k,l)}) \Big) \end{aligned} \tag{6}$$

where $\mathbf{U}_\mathbf{z}$ is the unitary of gate $\mathbf{R}_\mathbf{z}$ and $\boldsymbol{\theta}_\mathbf{z}^{(k,l)}$ is the parameter for $\mathbf{R}_\mathbf{z}$ at the l-th layer on the k-th qubit. The entanglement layer $\mathbf{U}_{ent}$ consists of CNOT gates and it entangles the adjacent qubits together shown in Fig. 2. The quantum state $|\Psi^l\rangle$ after l layers is

$$|\Psi^l\rangle = \mathbf{U}_{ent} \times \mathbf{U}_l \times \Big(\cdots \big(\mathbf{U}_{ent} \times \mathbf{U}_1 |\Psi^0\rangle \big) \Big) \tag{7}$$

The quantum state $|\Psi^0\rangle$ is the initial state, which is also the output of the Atom2Qubit stage.

With the parameters $\boldsymbol{\theta}_\mathbf{z}^{(k,l)}$ 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 downstream model. The loss function $\mathcal{L}$ which is employed to optimize the trainable parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathbf{z}^{(k,l)} \odot \boldsymbol{\theta}_\mathbf{y}^{(k,l)}$, where $\odot$ is concatenation, for our model $\boldsymbol{M}$ it 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 by gradient decent from Eq. 1.

The essence of the ansatz is to learn a unitary transformation from the input quantum state (which is the encoded tuple) to the output quantum state (which is the embedding we need). Therefore, we need an embedding ansatz with enough expressivity. We first discuss why we use the combination of $\mathbf{R_z R_y R_z}$ in the trainable layer.

Lemma 3.1. *Any single-qubit quantum gate U can be decomposed into a sequence of R_z, R_y and R_z gates, and a phase (Barenco et al., 1995).*

$$U = e^{i\alpha} R_z(\theta_2) R_y(\theta_1) R_z(\theta_0) \tag{9}$$

Proof. The matrices of all the quantum gates are unitary. Hence, U can be rewritten as

$$U = e^{\mathrm{i}\alpha} \begin{bmatrix} a & -b^* \\ b & a^* \end{bmatrix} = e^{\mathrm{i}\alpha} V, \tag{10}$$

where a, b are complex while α is real, and $\det V = aa^* + bb^* = |a|^2 + |b|^2 = 1$ (* denotes the conjugate operator). Then, we have

$$\det U = e^{2\mathrm{i}\alpha} \det V = e^{2\mathrm{i}\alpha}. \tag{11}$$

where i is the imaginary unit. We obtain the phase angle α:

$$\alpha = \frac{1}{2} \arctan 2 \big(\mathrm{Im}(\det U), \mathrm{Re}(\det U) \big). \tag{12}$$

where Re denotes the real part and Im denotes the imaginary part. Then, we plug in the matrices of rotation gates:

$$\begin{aligned} U &= e^{\mathrm{i}\alpha}\begin{bmatrix} e^{-\mathrm{i}\frac{\theta_2}{2}} & 0 \\ 0 & e^{\mathrm{i}\frac{\theta_2}{2}} \end{bmatrix}\begin{bmatrix} \cos\frac{\theta_1}{2} & -\sin\frac{\theta_1}{2} \\ \sin\frac{\theta_1}{2} & \cos\frac{\theta_1}{2} \end{bmatrix}\begin{bmatrix} e^{-\mathrm{i}\frac{\theta_0}{2}} & 0 \\ 0 & e^{\mathrm{i}\frac{\theta_0}{2}} \end{bmatrix} \\ &= e^{\mathrm{i}\alpha}\begin{bmatrix} e^{-\mathrm{i}\frac{\theta_0+\theta_2}{2}}\cos\frac{\theta_1}{2} & -e^{\mathrm{i}\frac{\theta_0-\theta_2}{2}}\sin\frac{\theta_1}{2} \\ e^{\mathrm{i}\frac{\theta_2-\theta_0}{2}}\sin\frac{\theta_1}{2} & e^{\mathrm{i}\frac{\theta_0+\theta_2}{2}}\cos\frac{\theta_1}{2} \end{bmatrix} \\ &= e^{\mathrm{i}\alpha}\begin{bmatrix} V_{00} & V_{01} \\ V_{10} & V_{11} \end{bmatrix}. \end{aligned} \tag{13}$$

Hence, we derive the angles of rotation gates:

$$\begin{aligned} \theta_1 &= 2\arccos|V_{00}|, \\ \theta_0+\theta_2 &= 2\arctan 2\big(\mathrm{Im}(|V_{11}|), \mathrm{Re}(|V_{11}|)\big), \\ \theta_2-\theta_0 &= 2\arctan 2\big(\mathrm{Im}(|V_{10}|), \mathrm{Re}(|V_{10}|)\big), \\ \theta_2 &= \arctan 2\big(\mathrm{Im}(|V_{11}|), \mathrm{Re}(|V_{11}|)\big) \\ &\quad + \arctan 2\big(\mathrm{Im}(|V_{10}|), \mathrm{Re}(|V_{10}|)\big), \\ \theta_0 &= \arctan 2\big(\mathrm{Im}(|V_{11}|), \mathrm{Re}(|V_{11}|)\big) \\ &\quad - \arctan 2\big(\mathrm{Im}(|V_{10}|), \mathrm{Re}(|V_{10}|)\big). \end{aligned} \tag{14}$$

Therefore, U is decomposed into a sequence of $\mathrm{R_z}$, $\mathrm{R_y}$ and $\mathrm{R_z}$ gates, and a phase. □

Corollary 3.2. *Any single-qubit quantum gate U can be decomposed into a set of $R_zR_yR_z$ and phase shift gates.*

Proof. Based on Lemma 3.1, we just need to decompose $e^{\mathrm{i}\alpha}$ into a set of $\mathrm{R_zR_yR_z}$ and the phase shift gates P:

$$\begin{aligned} U = e^{\mathrm{i}\alpha}V &= \begin{bmatrix} e^{\mathrm{i}\alpha} & 0 \\ 0 & e^{\mathrm{i}\alpha} \end{bmatrix} V \\ &= \begin{bmatrix} e^{\mathrm{i}\alpha} & 0 \\ 0 & e^{-\mathrm{i}\alpha} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & e^{2\mathrm{i}\alpha} \end{bmatrix} V \\ &= \mathrm{R_z}(-2\alpha)\mathrm{P}(2\alpha)V = \mathrm{R_z}(-2\alpha)\mathrm{R_y}(0)\mathrm{R_z}(0)\mathrm{P}(2\alpha)V \\ &= \mathrm{R_z}(-2\alpha)\mathrm{R_y}(0)\mathrm{R_z}(0)P(2\alpha)R_z(\theta_2)R_y(\theta_1)R_z(\theta_0). \end{aligned} \tag{15}$$

□

Next we discuss why the proposed Quantum Embedding Ansatz has the expressivity and why using $\mathrm{R_zR_yR_z}$ gates and CNOT gates can learn the embedding from the input and output.

Theorem 3.3. *The union of the set of single-qubit gates and CNOT is universal. (Williams et al., 1998)*

Since the $\mathrm{R_zR_yR_z}$ gate and the phase shift gate can express any single-qubit gate according to Corollary 3.2, the set of $\mathrm{R_zR_yR_z}$, P and CNOT is universal based on Theorem 3.3. Hence, the gate set $\{\mathrm{R_zR_yR_z}$, P, CNOT$\}$ can approximate any unitary matrix. We remove the phase shift gate P because the global phase $e^{\mathrm{i}\alpha}$ caused by P makes no difference to our results. Supposing the output state vector is $|\psi\rangle$, we add a global phase $e^{\mathrm{i}\alpha}$ to it and get $|\psi_p\rangle = e^{\mathrm{i}\alpha}|\psi\rangle$.

$$|\,|\psi\rangle^{(i)}\,|^2 = |e^{\mathrm{i}\alpha}|^2|\,|\psi\rangle^{(i)}\,|^2 = |e^{\mathrm{i}\alpha}\,|\psi\rangle^{(i)}\,|^2 = |\,|\psi_p\rangle^{(i)}\,|^2. \tag{16}$$

Hence, we ignore the impact of phase as consensus. Since all the single-qubit gates and the two-qubit gate CNOT can form a universal set, we can conclude that the $\mathrm{R_zR_yR_z}$ gate and the CNOT gate also form a universal set.

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 Pauli-Z measurement at the end of the circuit to extract the information from quantum superposition. The embedding after measurement is an N dimensional real vector

$$\mathcal{E}_N = \big(\langle\Psi^l_1|\sigma_z|\Psi^l_1\rangle, \langle\Psi^l_2|\sigma_z|\Psi^l_2\rangle, ..., \langle\Psi^l_N|\sigma_z|\Psi^l_N\rangle\big) \tag{17}$$

where $|\Psi^l_i\rangle$ denotes the i-th qubit of $|\Psi^l\rangle$, and σ_z is Pauli-Z.

Remarks on the NISQ device. We have been dedicated to design a quantum molecular learning algorithm that is capable of running on the NISQ device. Therefore, all the circuits are designed for the purpose of hardware efficient. For the data encoding part, we use angle encoding to avoid the state preparation difficulty. The embedding ansatz is hardware efficient which means all the rotation gates are easy to conduct on a quantum device and all the entanglement CNOT only connects two adjacent qubits. The readout of the circuit is using Pauli-Z measurement to avoid state tomography which is time consuming. The gradients are updated through the parameter shift rule as in Eq. 1. Thus, we can conclude that the proposed method is fully executable on a NISQ device.

4. Experiments

4.1. Protocols

All the experiments are performed on 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 by PyTorch, and we use TorchQuantum (Wang et al., 2022a) as the quantum simulator. 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 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 consist of up to 9 heavy atoms CONF, not counting hydrogen, and their corresponding 3D molecular geometries are computed by density functional theory (DFT).

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.

Setting. We follow the experimental setting as (Liu et al., 2021). The dataset is split into training/validation/test sets. The training set contains 110,000, the validation set contains 10,000, and the test set contains 10,831 molecules. Training molecules are used to optimize the model parameters. The validation molecules are used to fine-tune the hyperparameters as well as conduct the early stopping, and then we report the results on test molecules. 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 five baselines in the classical domain: the seminal work in SE(3)-invariant NN: SchNet (Schütt et al., 2017), DimeNet++ (Klicpera et al., 2020), SphereNet (Liu et al., 2021), ComENet (Wang et al., 2022b), and SE(3)-equivariant GNN: EGNN (Satorras et al., 2021). The results are extracted from the original papers since we are using the exact same experimental setting as these baselines, and more baselines can be directly obtained from (Wang et al., 2022b; Satorras et al., 2021).

Prediction model. The obtained embeddings are fed to a simple predictor, which is a multilayer perceptron reducing the size of the embedding from 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 0.01.

Results. The results of the property prediction task are presented in Table 1. As illustrated in the table, we achieve comparable results with all the classical baselines. We notice that there is still a gap between our results and the SOTA baselines. As stated in Sec. 2.2, QML algorithms barely take SOTA classical ML algorithms as their baselines. Moreover, we need to take into account the number of parameters we use when evaluating the efficiency of the model. We can see that our results are very close to EGNN, which uses 745,224 parameters. Our model uses only 384 rotation parameters in the quantum circuits and 1,040 parameters in linear predictor (the rotation parameters are not defined exactly the same as conventional parameters in neural networks, thus we list them separately).

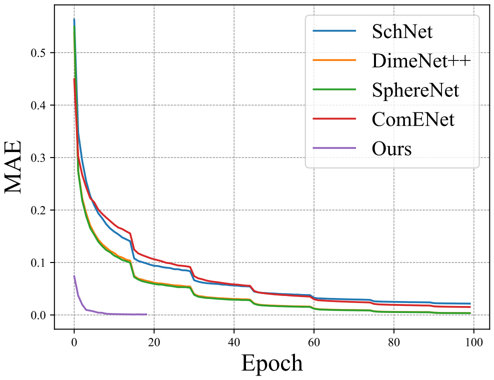

Figure 3. Training loss on the prediction task.

Table 2. Performance on three properties in QM9.

Property (meV)	10 qubits	12 qubits	14 qubits	16 qubits
ϵ_{HOMO}	39.7	38.6	35.6	28.9
ϵ_{LUMO}	47.1	34.8	29.1	27.1
Δ_ϵ	49.7	54.2	41.6	42.7

Moreover, our model converges faster than the classical baselines. As illustrated in Fig. 3, our model requires very few epochs to converge. We omit the comparison between the running time of each epoch since it varies a lot from the simulator to a NISQ device. We think it is not convincing to report the inference time per epoch on the simulator against all the baseline methods. To sum up, we believe the results can demonstrate the efficiency and the capability of the proposed quantum model.

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.

Setting. We also use filtered QM9 for evaluation. 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 **??**. 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

Table 1. Performance comparison between the baselines and our proposed method on QM9 in terms of MAE and the std. MAE for all twelve properties. We also list the number of parameters we need for all the methods.

Property	Unit	SchNet	DimeNet++	SphereNet	ComeNet	EGNN	Ours
μ	D	0.033	0.0297	0.0245	0.0245	0.029	
α	a_0^3	0.235	0.0435	0.0449	0.0452	0.071	
ϵ_{HOMO}	meV	41	24.6	22.8	23.1	29.0	28.9
ϵ_{LUMO}	meV	34	19.5	18.9	19.8	25.0	27.1
$\Delta\epsilon$	meV	63	32.6	31.1	32.4	48.0	42.7
$\langle R^2 \rangle$	a_0^2	0.073	0.331	0.268	0.259	0.106	
ZPVE	meV	1.7	1.21	1.12	1.20	1.55	
U_0	meV	14	6.32	6.26	6.59	11	
U	meV	19	8.02	6.26	6.82	12	
H	meV	14	7.90	6.36	6.86	12	
G	meV	14	8.11	6.33	7.98	12	
c_v	$\frac{\text{cal}}{\text{mol K}}$	0.033	0.0230	0.0215	0.024	0.031	
std. MAE	%	1.76	0.98	0.91	0.93		
$\#params$	—	455,809	1,887,110	1,898,566	4,185,857	745,224	1,040 (linear predictor) 384 (quantum circuit)

Table 3. Performance of the G-SphereNet and our proposed method on 500 randomly generated molecules for chemical validity percentage and MMD distances of bond length distributions.

		MMD distances ↓						
Method	Validity ↑	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**

Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) distances of bond length distributions to evaluate the 3D structural accuracy of the generated molecular geometries. 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 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, our model slightly outperforms the baseline model on MMD distances for 3 types of bond length, which shows that our method bears 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. Experiments on two well-studied downstream tasks show 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. It is aimed to extending our experiments to 10 thousand molecules and reaching the chemical accuracy of 1.6×10^{-3} Hartree.

References

Ai, X., Zhang, Z., Sun, L., Yan, J., and Hancock, E. Decompositional quantum graph neural network. *arXiv preprint arXiv:2201.05158*, 2022.

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G., Buell, D. A., et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

Aspuru-Guzik, A., Dutoi, A. D., Love, P. J., and Head-Gordon, M. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005.

Bai, L., Jiao, Y., Cui, L., Rossi, L., Wang, Y., Yu, P., and Hancock, E. Learning graph convolutional networks based on quantum vertex information propagation. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., and Weinfurter, H. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.

Bartlett, R. J. and Musiał, M. Coupled-cluster theory in quantum chemistry. *Reviews of Modern Physics*, 79(1): 291, 2007.

Bartlett, R. J., Kucharski, S. A., and Noga, J. Alternative coupled-cluster ansätze ii. the unitary coupled-cluster method. *Chemical physics letters*, 155(1):133–140, 1989.

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

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

Chen, S. Y.-C., Wei, T.-C., Zhang, C., Yu, H., and Yoo, S. Hybrid quantum-classical graph convolutional network. *arXiv preprint arXiv:2101.06189*, 2021.

Chen, S. Y.-C., Yoo, S., and Fang, Y.-L. L. Quantum long short-term memory. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8622–8626, 2022.

Cong, I., Choi, S., and Lukin, M. D. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

Dernbach, S., Mohseni-Kabir, A., Pal, S., and Towsley, D. Quantum walk neural networks for graph-structured data. In *Complex Networks and Their Applications*. Springer, 2018.

Gasteiger, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019.

Gasteiger, J., Becker, F., and Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34: 6790–6802, 2021.

Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Regep, C., Liu, G., Hayter, J. B., Vickers, R., Roberts, C., Tang, J., et al. Utilizing graph machine learning within drug discovery and development. *Briefings in bioinformatics*, 22(6):bbab159, 2021.

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

Gong, M., Wang, S., Zha, C., Chen, M.-C., Huang, H.-L., Wu, Y., Zhu, Q., Zhao, Y., Li, S., Guo, S., et al. Quantum walks on a programmable two-dimensional 62-qubit superconducting processor. *Science*, 372(6545): 948–952, 2021.

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

Hoffmann, M. R. and Simons, J. A unitary multiconfigurational coupled-cluster method: Theory and applications. *The Journal of chemical physics*, 88(2):993–1002, 1988.

Huang, H.-L., Wu, D., Fan, D., and Zhu, X. Superconducting quantum computing: a review. *Science China Information Sciences*, 63(8):1–32, 2020.

Huang, H.-L., Du, Y., Gong, M., Zhao, Y., Wu, Y., Wang, C., Li, S., Liang, F., Lin, J., Xu, Y., Yang, R., Liu, T., Hsieh, M.-H., Deng, H., Rong, H., Peng, C.-Z., Lu, C.-Y., Chen, Y.-A., Tao, D., Zhu, X., and Pan, J.-W. Experimental quantum generative adversarial networks for image generation. *Physical Review Applied*, 16(2), 2021.

Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J. M., and Gambetta, J. M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, sep 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Liu, Y., Wang, L., Liu, M., Lin, Y., Zhang, X., Oztekin, B., and Ji, S. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2021.

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

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., and Neven, H. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9 (1):1–6, 2018.

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

Nielsen, M. A. and Chuang, I. Quantum computation and quantum information, 2002.

O'Malley, P. J., Babbush, R., Kivlichan, I. D., Romero, J., McClean, J. R., Barends, R., Kelly, J., Roushan, P., Tranter, A., Ding, N., et al. Scalable quantum simulation of molecular energies. *Physical Review X*, 6(3):031007, 2016.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., and O'brien, J. L. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

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

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Romero, J., Babbush, R., McClean, J. R., Hempel, C., Love, P. J., and Aspuru-Guzik, A. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Science and Technology*, 4(1): 014008, 2018.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

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

Wang, H., Ding, Y., Gu, J., Lin, Y., Pan, D. Z., Chong, F. T., and Han, S. Quantumnas: Noise-adaptive search for robust quantum circuits. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 692–708. IEEE, 2022a.

Wang, L., Liu, Y., Lin, Y., Liu, H., and Ji, S. ComENet: Towards complete and efficient message passing for 3d molecular graphs. *arXiv e-prints*, pp. arXiv–2206, 2022b.

Wierschke, S. G. *Computational studies of organic biradicals*. PhD thesis, Purdue University, 1994.

Williams, C. P., Clearwater, S. H., et al. *Explorations in quantum computing*. Springer, 1998.

Yung, M.-H., Casanova, J., Mezzacapo, A., Mcclean, J., Lamata, L., Aspuru-Guzik, A., and Solano, E. From transistor to trapped-ion computers for quantum chemistry. *Scientific reports*, 4(1):1–7, 2014.

Zhang, Z., Chen, D., Wang, J., Bai, L., and Hancock, E. R. Quantum-based subgraph convolutional neural networks. *Pattern Recognition*, 88:38–49, 2019.

Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Qin, J., Wu, D., Ding, X., Hu, Y., et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

A. Discussion about Barren Plateau

Barren Plateau is a commonly faced problem for all the quantum machine learning methods. We have to admit that barren plateau is an infamous problem that haunts the development of quantum neural networks. However, according to (McClean et al., 2018), barren plateau will occur when the number of quantum layers is larger than 50 (with the number of qubit at 8), which is much more than the number of layers we use in our experiments (4 layers of rotation and entanglement). Apart from that, the Hardware Efficient Ansatz (HEA) we use to learn the latent embedding is a well-known and well-studied quantum ansatz, which has also been proved on real quantum devices (see also section 3.3). HEA to quantum neural network is just like fully-connected layer to the classical neural network. Thus, we omit the theoretical analysis of convergence.

B. More Insights and Motivation

This work might not be very innovative from a classical NN perspective, but it is a new branch in quantum computing to learn from molecules. It is totally different from the mainstream quantum ml approach UCC and saves us the effort to simulate a Hamiltonian when using Bessel function etc in classical learning. We use the neighbors of a focal atom to learn the embedding, which is similar to the definition of moieties. With the help of type, angles, and distance, we can encode the whole group of atoms around the focal atom into a quantum presentation. The core of our method is more like subgraph matching instead of message passing or GCN.

Classical algorithms using 3D coordinates can be summarized into two main categories. The first one is directly using them as three real numbers and concatenating them as a new dimension of the atom feature. Another one is using the spherical Bessel function to pre-process the coordinates. However, the spherical wave function or Bessel function is in the form of a Hamiltonian (or to be more specific a Unitary) which is a perfect fit for quantum computing. If we encode the coordinates as well as the distance and atom type into qubits, the circuit can be performed as a trainable Unitary. Setting all the encoding and training processes into a quantum version seems to be more suitable for such a problem.

C. Discussion about the Bond Length

It might be a concern that we do not include 2-hop messages in the experiments. We do realize that 2-hop messages are very important in quantum properties. But 2-hop messages can significantly increase the number of qubits we need. 1-hop is the atoms directly linked by the chemical bond, and the number is closely related to valence. With the maximum degree of 5, we can use 12 qubits to encode the whole group of atoms around the focal atom. If we take 2-hop messages into account, we might need a maximum of 42 qubits to encode the whole neighborhood. We are unable to simulate the quantum circuits over 14 qubits with our classical computing unit. We use the neighbors of a focal atom to learn the embedding, which is similar to the definition of moieties. With the help of type, angles, and distance, we can encode the whole group of atoms around the focal atom into a quantum presentation. The core of our method is more like subgraph matching instead of message passing or GCN, which is commonly used in SE(3)-invariant GNNs. As for the super parameter bond length in the paper, we set it as 1.77 so that all the direct links by the chemical bond are included in the neighborhood.