Integrating Quantum Synergistics and Machine Learning: A Transformative Approach to Drug Discovery

Krishna Sayori Deb, Shail Shah
Stevens Institute of Technology
Hoboken, NJ, USA
{kdeb, sshah21}@stevens.edu

August 5, 2025

Abstract

This paper explores the potential of Quantum Machine Learning (QML), particularly Quantum Graph Neural Networks (QGNNs), in addressing the challenges of molecular property prediction (MPP) for drug discovery. By leveraging quantum mechanical principles, QGNNs show promise in improving accuracy, efficiency, and scalability in tasks like HOMO-LUMO gap prediction and protein-ligand binding affinity estimation. This study reviews foundational concepts, quantum encoding techniques, and QGNN architectures, presenting results that highlight the advantages of QML in modeling high-dimensional, chemically rich datasets, and discussing its transformative potential in drug discovery.

Drug Discovery, Quantum Graph Neural Networks

1 Introduction

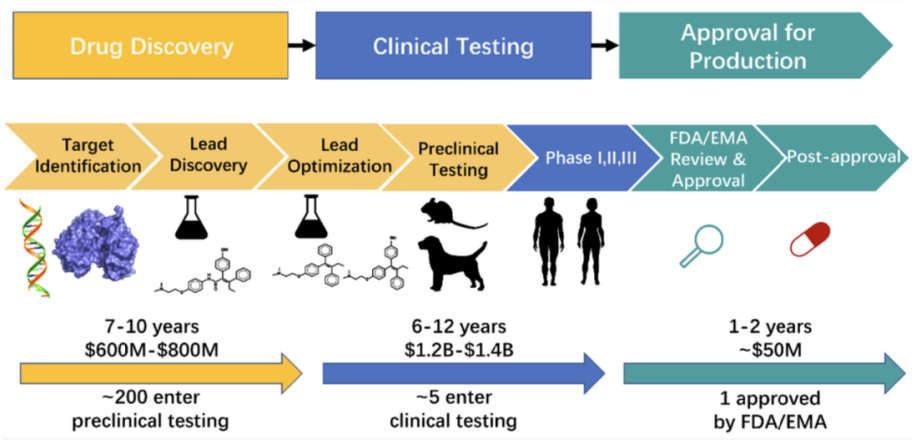
Molecular property prediction (MPP) is vital for drug discovery and materials science, providing insights into molecular behaviors essential for designing therapeutics and materials. However, MPP poses unique challenges for computational models due to complex relationships between structure and properties, as seen in activity cliffs, where minor structural changes cause significant property shifts. Traditional models like XGBoost often outperform deep learning (DL) due to their ability to handle high-dimensional, sparse, and complex molecular features.

Descriptors such as ECFP, SMILES, and graph-based methods like Graph Neural Networks (GNNs) offer ways to represent molecular data but face limitations, including high computational costs, data scarcity, and difficulty modeling quantum interactions. DL models often underperform on small datasets and struggle with capturing the complexity of molecular property functions, where data diversity and quality outweigh quantity. While sequence-based and graph-based representations each excel in specific areas, hybrid approaches combining these modalities offer promise but are challenging to optimize.

In his 1981 paper [1] Richard P. Feynman questioned the ability of classical computers to simulate quantum mechanical systems accurately. He argued that classical computers, constrained by finite operations and elements, cannot process the infinite complexities of physical events, such as discretizing time to scales as small as 10^{-27} seconds, without requiring immense computational resources. Additionally, classical computers struggle with simulating quantum mechanics' probabilistic nature, as they rely on pseudo-random numbers, which are deterministic and seed-based, unlike true quantum randomness. Feynman also highlighted the reversibility of quantum operations where outputs can regenerate inputs contrasting this with the irreversibility of classical computing elements. These limitations led him to propose the need for computers based on quantum mechanical principles, capable of processing data in line with the laws of quantum mechanics, thus introducing the concept of quantum computing.

- **RQ1: Can quantum computers accelerate the exploration of the vast molecular space in drug discovery?**
- **RQ2: Do quantum computers, leveraging qubits and quantum mechanical phenomena, offer any advantages in learning from scientific data compared to classical machine learning models?**

RQ1 tackles a fundamental challenge in drug discovery: the sheer scale of the molecular space. Identifying potential drug candidates within this vast space often involves computationally expensive simulations and experimental screenings. Quantum computers, with their unique capabilities, might offer a solution by accelerating the search and optimization process and enabling the exploration of a broader range of molecular possibilities.



[3]

Figure 1: Drug Discovery Pipeline

RQ2, taking inspiration from [1] delves into a crucial question regarding the potential superiority of quantum machine learning. Quantum computers, harnessing qubits and quantum mechanical phenomena like superposition and entanglement, could potentially have better modeling capabilities when data is related to chemistry, physics or other related sciences.

2 Computing in Molecular Science

2.1 Drug Discovery

Drug discovery is a complex, multi-faceted process that aims to identify and develop molecules capable of treating specific diseases. The process begins with **target identification**, where researchers pinpoint a protein or biomolecule that exhibits abnormal behavior in diseased tissues. This biomolecule, often referred to as the target molecule, plays a crucial role in the progression of the disease. Within the target molecule, the **active site** is identified as the specific region where potential drug molecules can bind to elicit a therapeutic effect [2, 3].

Following target identification, lead molecules, also called **ligands,** are screened using computational and experimental assays. These molecules are evaluated for their ability to interact with the target protein and modulate its function in a desired manner. This stage involves **high-throughput screening** techniques to sift through vast libraries of chemical compounds, narrowing down potential candidates. The identified leads undergo a process of **lead optimization**, where their chemical properties, such as binding affinity, toxicity, solubility, and metabolic stability, are fine-tuned. For instance, a molecule may perfectly fit the active site of a target protein but exhibit high toxicity, disqualifying it from further development [3].

The final stages involve **preclinical testing and clinical trials**, which evaluate the efficacy, safety, and potential side effects of the optimized leads in biological systems. Drug discovery is notoriously **time-intensive and costl**y, often taking years or even decades to bring a single therapeutic agent to market. Additionally, the vast chemical space estimated to contain over 10^{60} possible drug-like molecules makes the identification of optimal candidates an enormous computational and logistical challenge [3]. These factors underscore the need for efficient and accurate computational methods to accelerate and refine the drug discovery pipeline.

2.2 Machine Learning in Molecular Property Prediction

Machine learning (ML) has revolutionized molecular property prediction (MPP), offering data-driven approaches to tackle the challenges of drug discovery. ML models excel at identifying patterns and relationships within vast datasets, enabling significant advancements in key areas like target identification, active site prediction, and lead optimization. For example, ML algorithms can predict the 3D structure of a target protein, even when experimental structures are unavailable, by comparing sequence similarities with known proteins. These predictive capabilities reduce the reliance on resource-intensive experimental techniques.

In lead identification, ML-based virtual screening can rapidly evaluate thousands of chemical compounds against a target protein, predicting their binding affinities and selecting the most promising candidates for experimental validation. Similarly, ML models are used for toxicity prediction and the assessment of other pharmacokinetic properties during lead optimization, ensuring that candidate molecules meet safety and efficacy standards before entering clinical trials. Cheminformatics, a subfield of MPP, leverages ML to design compounds with specific properties, such as increased potency or reduced side effects, by analyzing chemical features and their correlations with biological activity.

Despite its transformative potential, ML faces notable challenges in MPP. Molecular data often exhibit **long-term dependencies**, where small structural modifications, known as **activity cliffs**, lead to drastic changes in molecular properties. Capturing these nuances is particularly difficult for ML models. Furthermore, representations of molecular data, such as **SMILES strings** or graph-based encodings, may not fully capture critical features like bond angles or 3D conformations, impacting model accuracy. The sheer diversity of chemical space also results in datasets with limited coverage, leading to distributional mismatches between training and testing sets.

Nonetheless, ML continues to play a pivotal role in overcoming these challenges, with ongoing advancements in molecular representation learning and the development of specialized algorithms tailored to the unique complexities of molecular data.

2.2.1 Graph Neural Networks in Molecular Property Prediction

Molecular property prediction is a crucial aspect of drug discovery and materials design. The task involves predicting the chemical and physical properties of novel molecules, which is essential for evaluating their suitability as potential drug candidates or materials. However, traditional deep learning methods, like CNNs, struggle with molecular data due to the arbitrary size and shape of molecules. **GNNs offer a solution by representing molecules as graphs, where atoms are nodes and bonds are edges.** This graph representation enables GNNs to capture the structural information of molecules effectively.

GNNs for molecular property prediction work in two stages:

- **Graph Filtering:** GNNs employ graph filters to learn representations of molecules by aggregating information from neighboring nodes. These filters combine node features (e.g., atom type, charge) and graph structure (e.g., bond type, bond order) to generate refined node features. The number of features per node can be expanded to account for multiple characteristics. Spatial filters, like those in GCN, GAT, and GraphSage, aggregate information from neighboring nodes to update node features. MPNN [4] is a general framework for spatial filters that can incorporate edge features like bond type and bond order.
- **Pooling:** For graph-level predictions, GNNs use pooling layers to summarize node-level features into a graph-level representation (molecular fingerprint). This fingerprint encodes the overall structural and chemical information of the molecule and is used for downstream property prediction tasks. Flat graph pooling methods, like sum or mean pooling, simply aggregate node features, but they may not capture hierarchical information. Hierarchical graph pooling methods address this by coarsening the graph, preserving structural hierarchy.

The learned molecule fingerprint is then fed into a prediction model (classifier or regressor) also known as fine-tuning model to predict the desired molecular property. **The advantage of GNNs is that they can learn molecular structural information better for the downstream prediction task, leading to improved accuracy and efficiency.** The model is pre trained using millions of unlabeles data and the fine-tuning model then use the pre-trained weights to further train on small amount of task specific labeled data. The pre-training and fine-tuning model framework has the potential of using both labeled and unlabeled molecular data to train GNNs for better representation learning. This is particularly relevant as labeled molecular data is often scarce in practice.

2.3 Quantum Machine Learning in Molecular Property Prediction

2.3.1 Variational Quantum Circuits

Quantum Neural Networks (QNNs) integrate quantum computing into machine learning, leveraging quantum parallelism for enhanced computational efficiency. **Variational Quantum Circuits (VQCs)** form the backbone of QNNs, using parameterized gates for optimization. These circuits combine quantum computations with classical post-processing, addressing tasks such as classification and regression. Challenges include noise, **barren plateaus** (regions with negligible gradients), and the limited qubit count in current quantum devices.

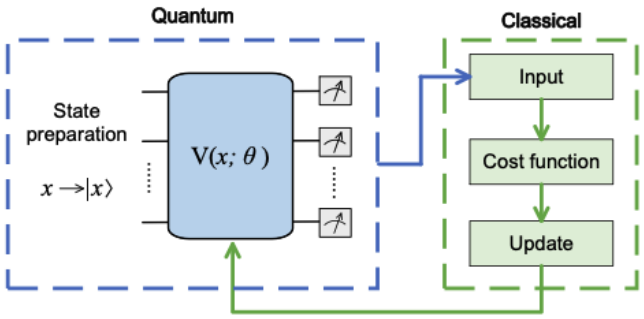
2.3.2 Encoding Techniques

Classical data is encoded into quantum states using strategies such as:

- **Basic Encoding:**[6] Direct mapping of classical data into quantum states, suitable for simple datasets but limited in scalability. The basis encoding equation for representing a classical binary dataset as a quantum state is given by:

$$|D\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} |x_m\rangle.$$

  Where :



[5]

Figure 2: Variational Quantum Circuit Architecture

- $|D\rangle$: This represents the quantum state that encodes the entire classical dataset.
- M: This is the total number of basis states used for the encoding, which corresponds to the number of data points in the dataset.
- x_m: This corresponds to the m-th element in the dataset. Each element x_m is an N-bit binary string of the form $x_m = (b_{m1}, b_{m2}, ..., b_{mN})$, where each b_{mj} is either 0 or 1.
- $|x_m\rangle$: This is the quantum state representing the m-th element of the dataset. Each binary string x_m is mapped to a computational basis state of N qubits. For instance, the binary string (1, 0, 1) would be mapped to the quantum state $|101\rangle$.
- $\frac{1}{\sqrt{M}}$: This is a normalization factor that ensures the quantum state $|D\rangle$ has unit norm. To encode a specific element x_m, Pauli X gates are applied to the qubits that should be 1 in the corresponding basis state. For example, to encode the binary string (1, 0, 1) using three qubits, we would apply Pauli X gates to the first and third qubits, leaving the second qubit in the state $|0\rangle$.

- **Angle Encoding:**[6] Angle encoding is a technique used in quantum machine learning to represent classical data as a quantum state by encoding numerical values as rotation angles of qubits. This method is different from basis encoding, which is limited to binary data. Angle encoding can be used to encode real, floating-point numbers, making it suitable for various applications in quantum machine learning, such as representing molecular properties or features for QML models.

  To use angle encoding, you first normalize the classical data to a specific range using

  $$x_{\text{angles}} = \pi \cdot \frac{x - \min(x)}{\max(x) - \min(x)} = [0, \pi, \frac{2\pi}{5}]$$

  , usually between 0 and 2π or 0 and π. This ensures the data can be mapped to valid rotation angles for qubits. The normalized data points are then used to rotate the state of a qubit around a specific axis on the Bloch sphere by an angle corresponding to the data. This is done by using rotation gates like Rx, Ry, and Rz, which rotate the qubit state around the X, Y, and Z axes, respectively.

  The rotation operator for a data point x_m is given by:

  $$x_m \rightarrow R_k(x_m)|0\rangle = e^{-ix_m\sigma_k/2}|0\rangle,$$

  where k represents the rotation axis, and σ_k is the corresponding Pauli matrix. For instance, if $k = y$, then $R_k(x_m)$ represents a rotation around the Y-axis by an angle of x_m.

  It can be used to represent real-valued data, which is more common in many applications than binary data. It provides a clear way to represent numerical data as quantum states, by mapping values directly to rotation angles. However, angle encoding also has limitations, Rotation gates are periodic, which means that data points that differ by multiples of 2π will be encoded into the same quantum state. This can be addressed by appropriately normalizing the data, but it is important to be aware of this limitation. It's often challenging to find a computational advantage with angle encoding because the number of qubits needed usually scales linearly with the size of the dataset. This means that angle encoding might not be the best choice for problems where you need to encode large amounts of data.

- **Amplitude Encoding:**[6] Amplitude encoding is a technique used in quantum machine learning (QML) to represent classical data as quantum states by encoding the classical data into the amplitudes of those quantum states. This method offers an efficient way to represent data because the number of qubits needed scales logarithmically with the size of the data, potentially leading to a more compact representation compared to other encoding methods like angle or basis encoding.

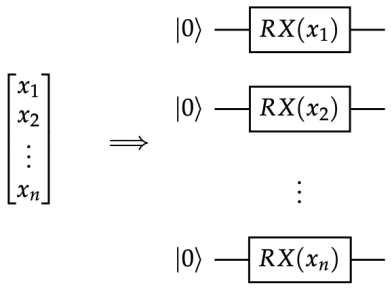


Figure 3: An example of quantum encoding. Individual elements of an input classical vector is used as the rotation angles of rotation Pauli X (RX) gates, creating a quantum state
[7]

In amplitude encoding, a classical data point, typically represented as a normalized complex-valued vector x, is encoded in the amplitudes of a quantum state. Mathematically, this process can be represented as follows:

$$x \rightarrow U_x|0>^{\otimes log(N)} = |x> = \sum_{k=0}^{2^{N-1}} \alpha_k|k>.$$

- x: This represents the classical data point to be encoded.
- U_x: This is a unitary operator that encodes the classical data into the amplitudes of the quantum state. This operator is typically implemented as a sequence of quantum gates acting on the initial state.
- $|0>^{log(N)}$: This represents the initial state of the system, which is a tensor product of $log(N)$ qubits all in the $|0>$ state.
- $|x>$: This is the final quantum state after applying the unitary operator to the initial state. This state has the classical information encoded in its amplitudes.
- $\sum_{k=0}^{2^{N-1}} \alpha_k|k>$: This summation represents the quantum state $|x>$ in terms of the amplitudes $_k$, which contain the encoded information, and the basis states $|k>$, which span the Hilbert space of N qubits.

Amplitude encoding can represent complex-valued data in a very compact way, often requiring fewer qubits than other methods. Amplitude encoding may offer a computational advantage over classical methods for specific tasks. This is because it leverages the exponential nature of quantum states to represent classical data. Constructing the unitary operator U_x needed to encode the data can be complex and require many quantum gates, potentially limiting its use on near-term quantum devices. Amplitude encoding is sensitive to noise, which is common on current quantum computers. This noise can corrupt the amplitudes of the quantum state, leading to errors in the encoded information.

2.3.3 Quantum Graph Neural Networks (QGNNs)

Graph neural networks (GNNs) model molecules as graphs, where nodes represent atoms and edges represent bonds. They are extensively used in drug development for tasks like protein-ligand binding and molecular activity prediction. Quantum Graph Neural Networks (QGNNs) extend this concept by encoding molecular graphs into quantum Hilbert spaces, leveraging quantum operations for graph data processing.

QGNNs have shown promise in tasks like predicting molecular stability and material properties, often outperforming classical counterparts in efficiency and accuracy. Hybrid quantum classical approaches, where quantum layers replace classical layers in GNNs, further demonstrate competitive results. These methods combine parameter efficiency with generalization, showcasing their potential in scaling molecular modeling to larger protein structures and complex molecular systems.

3 QGNN on QM9

3.1 QM9 Dataset

The QM9 dataset from MoleculeNet paper [8] is a widely used resource in computational chemistry and machine learning, featuring a collection of about 134,000 small organic molecules with up to 9 non-hydrogen atoms. This

Table 1: Node features and link features in the processed QM9 dataset.

Type	Feature	Explanation → Data Type
Nodes → Non-Hydrogen Atoms	Atomic Number	C/N/O/F → Integer
	Number of Bonded Hs	0-4 → Integer
	Aromaticity	True/False → Boolean
	Hybridization	sp, sp^2, sp^3 → Integer
Links → Bonds	Bond Type	Single/Aromatic/Double/Triple → Integer

[7]

	Atomic Number	Atomic Number and Number of Hydrogens	Atomic Number, Aromaticity, and Hybridization
RY Rotation Angle	$z = 4 \to 0$, $z = 5,6,7 \to cos(-1/3)$	$\frac{(2z-7)\pi}{4}$	$\frac{(2z-7)\pi}{4}$
RZ Rotation Angle	$z = 4,5 \to 0$, $z = 6 \to 2\pi/3$, $z = 7 \to -2\pi/3$	$\frac{2\pi n_h}{5}$	$\frac{(-1)^a(2h-1)\pi}{6}$

Figure 4: Quantum encoding methods. z is the atomic number; n_H is the number of bonded hydrogen atoms; a is the aromaticity as True(1)/False(0); h is the hybridization type (sp: 1, sp^2: 2, sp^3: 3).
[7]

dataset includes properties calculated using Density Functional Theory (DFT), such as electronic, thermodynamic, and vibrational properties. Each molecule in QM9 is represented by its 3D coordinates, atomic numbers, and connectivity information, making it particularly suitable for tasks involving molecular structure prediction, property estimation, and the development of graph neural network models that predict molecular characteristics.

3.2 Quantum Encoding

Quantum encoding in the paper [7] involves assigning a qubit to each node, with initial qubit states determined by node feature values. Three encoding methods were tested, each using a combination of RY and RZ gates configured according to specific node features like atomic number, number of bonded hydrogens, aromaticity, and hybridization.

3.3 EDU-QGC

The EDU-QGC (Equivariantly Diagonalizable Unitary Quantum Graph Circuit) described in the paper [7] employs a two-part approach:

- Node-local unitaries use single-qubit unitaries that vary based on the node's connection and features.
- Link layers utilize equivariantly diagonalizable unitaries (EDUs), which are defined by the bond type. These unitaries are designed to respect the graph's symmetry by being equivariant under node permutations.

The EDU formulation allows for complex interactions modeled by the quantum system, parameterized by bond-specific variables, which enable the quantum circuit to adapt to different molecular structures.
[7]

3.4 Readout Function

The readout function is designed to be invariant under node permutations, ensuring that the quantum computation outcome is independent of node order. Two forms of readout functions were tested:

- Local readout function: Sums the expectation values of Pauli Z operators across nodes.
- Global readout function: Includes a term that aggregates information across all nodes.

Equations provided in this section describe the mathematical formulations for the EDU-QGC architecture and the readout functions:

- Local Readout Function:

$$r_0 + \frac{r_1}{|V|} \sum_{v \in V} \langle Z_v \rangle$$

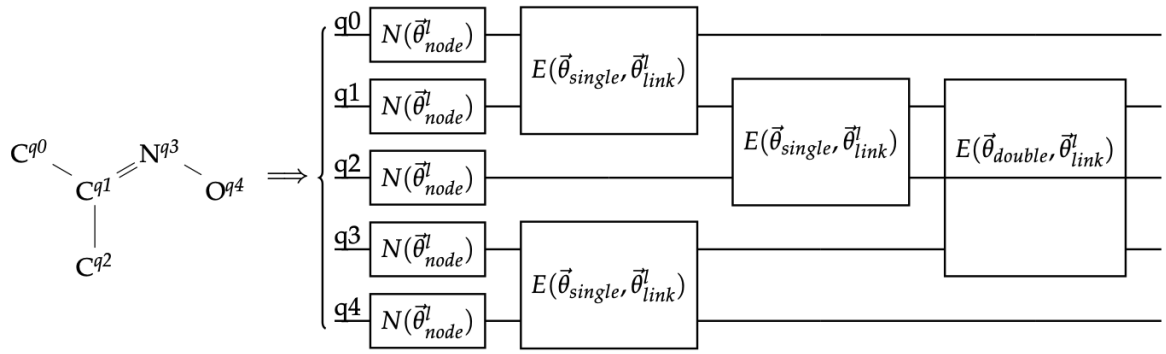


Figure 5: An example of a bond-information-encoding EDU-QGC layer is provided here. The input molecule is illustrated on the left without its hydrogen atoms. The superscripts denote the assigned qubit indices. The circuit diagram on the right depicts one EDU-QGC layer. N and E represent node-local unitaries and EDUs, respectively. The trainable parameters $\vec{\theta}_{\text{node}}$ and $\vec{\theta}_{\text{link}}$ vary depending on the layer number l, while $\vec{\theta}_{\text{single}}$ and $\vec{\theta}_{\text{double}}$ are trainable variables that depend on the bond order. It is noted that the single bonds are applied first, followed by the double bond.

- Global Readout Function:

$$r_0 + \frac{r_1}{|V|} \sum_{v \in V} \langle Z_v \rangle + r_2 \left\langle \bigotimes_{v \in V} Z_v \right\rangle$$

These components together form a quantum graph neural network capable of handling complex molecular data for property prediction, with specific focus on harnessing quantum computing's potential for managing high-dimensional data spaces and complex transformations dictated by quantum mechanics.

3.5 Results

The results of the EDU-QGC (Equivariantly Diagonalizable Unitary Quantum Graph Circuit) from the paper [7] focus on the performance of quantum graph neural networks in predicting molecular properties. The EDU-QGC model demonstrated several key outcomes:

- **Performance Comparison:** EDU-QGC showed improved prediction accuracy compared to classical graph neural networks. This improvement was particularly notable in scenarios where both models had a comparable number of trainable parameters.
- **Training Efficiency:** The quantum models utilizing EDU-QGC converged faster during training sessions than their classical counterparts, indicating a more efficient learning process potentially due to the quantum advantage in handling the molecular data.
- **Scalability and Adaptability:** The EDU-QGC was capable of adapting its architecture based on the bond order within molecules, which allowed it to effectively handle various molecular structures, showing its versatility and potential for scalability in handling complex molecular datasets.
- **Loss Metrics:** Lower test loss metrics were reported for the EDU-QGC models when compared with classical models, suggesting higher overall prediction accuracy and model robustness.

4 Classical VS Quantum Comparison

The paper [9] primarily focuses on the theoretical frameworks and potential advantages of quantum GNNs, particularly in terms of time and space complexity. While they suggest that quantum GNNs could outperform their classical counterparts for large-scale graphs [9], they lack direct comparisons of performance on specific tasks. **The paper [6] highlights that empirical evaluations of quantum GNN performance are needed to assess their capabilities in real-world applications.** However, the paper [6] does mention a few instances where quantum GNNs show promising results:

- **QGNNs for HOMO-LUMO gap prediction:** HOMO (Highest Occupied Molecular Orbital) represents the highest energy level occupied by electrons in a molecule. LUMO (Lowest Unoccupied Molecular Orbital) represents the lowest energy level that is unoccupied by electrons. The energy difference between HOMO and LUMO (the HOMO-LUMO gap) determines the molecule's reactivity. A small gap indicates higher

reactivity, while a large gap suggests greater stability. In some cases, quantum GNNs designed to predict the HOMO-LUMO gap in molecules have outperformed classical GNNs with the same number of parameters. This suggests that quantum GNNs might offer favorable scalability and generalization [6].

  - **Classical HOMO-LUMO Test MAE :** In the paper [4], the best MPNN (GNN) variant (enn-s2s) achieved an MAE ratio of 0.99 for HOMO energy prediction. For LUMO energy prediction, the same MPNN variant (enn-s2s) obtained an MAE ratio of 0.87.
  - **Quantum HOMO-LUMO Test MAE :** The Quantum Graph Neural Networks (QGNN) utilized for predicting the HOMO-LUMO gap of molecules showed the following Mean Absolute Error (MAE) results for different encoding methods:
    * **Atomic Number:** MAE = 1.9975×10^{-2} Hartree
    * **Atomic Number and Number of Hydrogens:** MAE = 1.8793×10^{-2} Hartree
    * **Atomic Number, Aromaticity, and Hybridization:** MAE = 2.0811×10^{-2} Hartree

- **Hybrid QGNNs for protein-ligand binding affinity prediction:** Hybrid quantum-classical GNNs have shown performance on par with state-of-the-art classical models for predicting protein-ligand binding affinities [6]. These hybrid models use amplitude encoding for features and replace classical multi-layer perceptrons (MLPs) with parameterized quantum circuits (PQCs) for convolution operations, leading to a good balance between the number of parameters and generalization capabilities [6].
- **Quantum Convolutional Neural Networks (QCNNs) for drug toxicity prediction:** Paper [6] mentions a study where a hybrid quantum-classical CNN was successfully trained to predict drug toxicity, achieving competitive results compared to classical models while potentially reducing training times [9, 6].

5 Conclusion

In conclusion, the papers discussed in this study indicate that quantum computers have the potential to achieve faster and more efficient convergence for optimization tasks in quantum machine learning compared to classical methods. Additionally, they demonstrate improved accuracy and lower loss on the QM9 dataset. However, further testing on other datasets within MoleculeNet [8] is essential to validate these results. Moreover, whether this improvement stems from the inherent alignment of modeling molecules with qubits, which are naturally more representative of molecular structures than classical bits, remains an open question that requires deeper investigation.

References

[1] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982.

[2] Nathan Brown. Chemoinformatics—an introduction for computer scientists. *ACM Computing Surveys*, 2009.

[3] Y. Zhang et al. Application of computational biology and artificial intelligence in drug design. *International Journal of Molecular Sciences*, 2022.

[4] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1263–1272. PMLR, 2017.

[5] Runqiu Shu, Xusheng Xu, Man-Hong Yung, and Wei Cui. Variational quantum circuits enhanced generative adversarial network. *arXiv preprint arXiv:2308.11062*, 2023.

[6] Anthony M. Smaldone, Yu Shee, Gregory W. Kyro, Chuzhi Xu, Nam P. Vu, Rishab Dutta, Marwa H. Farag, Alexey Galda, Sandeep Kumar, Elica Kyoseva, and Victor S. Batista. Quantum machine learning in drug discovery: Applications in academia and pharmaceutical industries. *arXiv preprint arXiv:2405.08294*, 2024.

[7] Ju-Young Ryu, Eyuel Elala, and June-Koo Kevin Rhee. Quantum graph neural network models for materials search. *arXiv preprint arXiv:2308.11759*, 2023.

[8] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[9] Yidong Liao, Xiao-Ming Zhang, and Chris Ferrie. Graph neural networks on quantum computers. *arXiv preprint arXiv:2308.11897*, 2023.