Quantum Advantage in Small Molecule Drug Discovery

Po-Yu Kao,¹ Ya-Chu Yang,¹ Wei-Yin Chiang,² Jen-Yueh Hsiao,² Min-Hsiu Hsieh,²,† and Yen-Chu Lin¹, ³,†

¹Insilico Medicine Taiwan Ltd., Taipei, Taiwan
²Hon Hai (Foxconn) Research Institute, Taipei, Taiwan
³Department of Pharmacy, National Yang Ming Chiao Tung University, Taipei, Taiwan

(Dated: November 1, 2022)

ABSTRACT

The drug development process is not only time and resource-consuming but also has a low probability of success. Recent advances in machine learning and deep learning technology have reduced costs and improved pharmaceutical research and development. De novo drug design with desired biological activities is crucial for developing novel therapeutics for patients. It is also an important step to keep the drug discovery pipeline moving forward. Artificial intelligence has pushed the limit of conventional drug design approaches, and quantum computing has demonstrated its advantages in different applications, e.g., solving routing problems and stock price forecasting. We proposed a hybrid quantum-classical generative adversarial network (GAN) for small molecule discovery. We substituted each element of GAN with a variational quantum circuit (VQC) and demonstrated the quantum advantages in the small drug discovery. Utilizing a VQC in the noise generator of GAN to generate small molecules achieves better physicochemical properties and performance in the goal-directed benchmark than the classical counterpart. Moreover, we demonstrate the potential of a VQC with only tens of learnable parameters in the generator of GAN to generate small molecules, which is a more complex problem than other quantum computing applications. In the end, we also demonstrate the quantum advantage of a VQC in the discriminator of GAN. In this hybrid model, the number of learnable parameters is significantly less than the classical ones, and it can still generate valid molecules. The hybrid model with only tens of training parameters in the quantum discriminator outperforms the MLP-based one in terms of generated molecule properties and KL-divergence.

I. INTRODUCTION

The drug development process includes discovery and development, pre-clinical research, clinical research, Food and Drug Administration (FDA) review, and FDA post-market safe monitoring. The entire process is time and resource-consuming and has a low probability of success with approximately 4% pre-clinical drugs eventually granted license [1]. The average time for a new medicine to complete the journey from initial discovery to the marketplace takes at least ten years [2]. The estimated median capitalized research and design (R&D) cost per new drug (accounting for the cost of failures) was $985 million between 2009 and 2018 [3]. Recent advances in machine learning and

* Corresponding e-mail: min-hsiu.hsieh@foxconn.com
† Corresponding e-mail: jimmy.lin@insilicomedicine.com
deep learning technology have improved and reduced the cost of pharmaceutical R&D [4–9]. For example, the work [10] discovers potent inhibitors of discoidin domain receptor 1 (DDR1), a kinase target implicated in fibrosis and other diseases, in 21 days. Chan et al. [11] estimated their machine learning algorithms would shrink the drug candidate identification phase from a few months to one year.

*De novo* drug design refers to novel chemical compound design with desired pharmacological and physicochemical properties [12]. The discovery of novel chemical compounds with desired biological activities is a critical step to keep the drug discovery pipeline moving forward [13]. It is also crucial for developing novel therapeutics for patients [14]. Conventional approaches include ligand-based drug design (LBDD), fragment-based drug design (FBDD), and structure-based drug design (SBDD). LBDD is based on known active binders of a biological target, and FBDD identifies small molecular fragments with weak affinity for a biomolecular target of interest and assembles them into fully drug-like compounds [15]. Aside from LBDD and FBDD, SBDD is based on the properties of the active site of a biological target.

Artificial intelligence (AI), including machine learning and deep learning algorithms, has made a breakthrough in the recent *de novo* molecule design [14, 16–18]. We identify four well-known generative machine learning algorithms in the field: evolutionary algorithms (EA), recurrent neural network (RNN) such as gated recurrent unit (GRU) and long short-term memory (LSTM), autoencoders such as adversarial autoencoder (AAE) and variational autoencoder (VAE), and generative adversarial network (GAN) [19]. GAN [20] has become a popular network architecture for generating highly realistic data [21], and it has shown remarkable results for generating data that mimics a data distribution in different tasks [22–25]. GAN consists of a generator and a discriminator defined by an artificial neural network (ANN). The parameters of GAN can be learned by backpropagation. The generator takes random noises as input and tries to imitate the data distribution, and the discriminator tries to distinguish between the fake and real samples. GAN is trained until the discriminator cannot distinguish the generated data from the real data.

Several different GAN architectures have been proposed in the past decades in *de novo* drug discovery [26–28]. Guimaraes et al. [29] proposed an objective-reinforced generative adversarial network (ORGAN) that combines the GAN and reinforcement learning (RL) algorithm. ORGAN is built on SeqGAN [30] and is the first GAN architecture in the *de novo* molecule generation. It is a sequential generative model operating on simplified molecular-input line-entry system (SMILES) string representations of molecules. The generated samples of ORGAN maintain information originally learned from data, retain sample diversity, and show improvement in the desired drug properties. Prykhodko et al. [31] presented a novel neural network architecture called LatentGAN for *de novo* molecular design. It combines an autoencoder and a generative adversarial neural network. The generator and discriminator of LatentGAN take n-dimensional vectors as inputs. These inputs derived from the code layer of an autoencoder are trained as a SMILES heteroencoder [32]. This method allows LatentGAN to focus on optimizing the sampling without worrying about SMILES syntax issues. Cao and Kipf introduced MolGAN [33] for small molecule *de novo* design which operates directly on graph-structured data. It is the first GAN to address the generation of graph-structured data in the context of molecular generation. MolGAN is demonstrated to generate close to 100% valid compounds in experiments on the quantum machines 9 (QM9) chemical database. The generated molecules of MolGAN have better chemical properties particularly in synthesizability and solubility than the generated compounds of ORGAN. Neither ORGAN nor MolGAN is directly compared with LatentGAN in the paper.

AI-driven strategies have the potential to explore adjacent chemical space compared to classical discovery efforts in *de novo* drug design [34]. However, the traditional generative machine learning algorithms have difficulty exploring the new chemical space different from the training dataset. The
state space dimension that represents molecule structure scales quadratically with molecule size. It increases the difficulty for the classical generative model on sampling as the landscape becomes extremely large and impossible to screen all possible configurations. Thus, an efficient sampling method is needed.

The applications of quantum computing can be found in different fields, such as solving routing problem [35, 36], the stock price forecasting [37, 38], molecule properties discovery [39–41], tautomeric state prediction [42] and drug design [43]. We can obtain more efficient and accurate results by utilizing the technique of quantum annealer, quantum machine learning, and quantum algorithm. These advantages that utilize the fundamental properties of quantum mechanics to achieve better performances compared to the classical methods are called quantum advantages. These performances can be on the computational resources requirement in simulation, computational efficiency in algorithm development, and measurement precision in metrology. More specifically, the quantum advantages can be identified in three aspects: scalability, complexity, and accuracy. The idea of utilizing the intrinsic quantum properties of quantum processors to simulate quantum systems was proposed by Feynman in 1982 [44], allowing us to understand the characteristics of large quantum systems with a feasible computational resource and achieving the quantum advantage of scalability. For complexity, researchers found that the complexity of many algorithms [45–48] can be significantly reduced by the quantum properties. Better measurement accuracy can be achieved by sophisticated designed protocols that enhanced the accuracy from the standard quantum limit to the Heisenberg limit [49–52].

Although quantum computing seems to lead a significant leap in solving resource-demanding problems, a few challenges remain to be overcome, such as the imperfect reliability of the quits prohibits the physical implementation of large-scale quantum computers [53]. Nevertheless, noisy intermediate-scale quantum (NISQ) computers can still perform classical challenge tasks accompanied by quantum advantages [54]. For example, the advantages are shown in sampling the output of a pseudo-random quantum circuit [55], the sampling time complexity of a Torontonian of a matrix [56, 57], expressively in unsupervised learning [58] and parameter space complexity in reinforcement learning [59].

Although the idea of data conversion in the quantum circuit through gate operators is similar to the classical neural network, efficient sampling that can be achieved by quantum circuits as the fundamental principle of quantum mechanics allows the coexistence of all possible states exponentially large in the quantum circuit. This fundamental property of superposition in quantum mechanics gives exponentially larger state space, allowing efficient sampling that is difficult for the classical model. In fact, several studies have demonstrated that variational quantum circuit (VQC) performs the advantages in expression power [60], learnability [61] and robustness [62]. It indicates that VQC is the most plausible solution to the problem, e.g., drug discovery, which is hard to tackle by the classical neural network.

The innate advantages of VQC are suited for computational resource-demanding tasks, such as identifying possible chemical reaction pathways and predicting molecule structures. The ground state energy of different molecule configurations reveals the most plausible reaction mechanism. It is demonstrated that the ground state energy of small molecules, such as H₂, LiH, BeH₂, H₁₂ and diazene [40], can be obtained by VQC, in which, the parameterized angles in the circuit are updated till the minimum energies are achieved under the given molecule configuration. Furthermore, the obtained ground state energies are shown to be consistent with the results from the classical Full-CI calculation [41].

Quantum generative adversarial network (QuGAN) [63] provided the first theoretical framework of quantum adversarial learning. QuGAN’s exponential advantages over classical GANs directly result from the ability of quantum information processors to represent N-dimensional features using
$\log N$ qubits with time complexity of $O(poly(\log N))$. Recent studies also showed that generative models implemented by quantum circuits with fewer architecture complexities could easily bypass their classical counterparts \[43, 64\]. Dallaire-Demers et al. provided the first feasible implementation of QuGAN using quantum circuits in a simulator \[65\]. Later on, QuGANs demonstrated its first successful training on the MNIST dataset on a physical quantum device \[64\]. Li et al. \[43\] pushed the research to real-world further by showing the QuGANs can learn or generate the distribution of the QM9 dataset, which provides the quantum chemical properties for small organic molecules in drug design. However, the source code \[66\] they provided struggles with generating training-set-like molecules. In addition, it lacks a detailed comparison between the generated samples from QuGAN and those from classical GAN.

In this work, we perform the training tasks on the QM9 dataset using the classical and quantum GAN. We not only demonstrate that the quantum GAN outperforms the classical GAN in the drug properties of generated compounds and the goal-directed benchmark but ensure that the trained quantum GANs can generate training-set-like molecules by using the variational quantum circuit as the noise generator. In addition, we show the potential of the variational quantum circuit in the generator of GAN to generate small molecules. In the end, we demonstrate that the quantum discriminator of GAN outperforms the classical counterpart in terms of generated molecule properties and KL-divergence score. The source codes will be publicly available once this manuscript is accepted.

II. METHODOLOGY

Generative adversarial net (GAN) \[20\] was first proposed in 2014. It consists of two elements: a generator and a discriminator. The objective of the generator is to generate fake data which mimics the real training data, and the goal of the discriminator is to distinguish real data from fake data. To achieve these goals, the generator and discriminator are trained at the same time using the training data. During the training process, both elements compete with each other and iteratively improve with one another. By the end, the generator can generate novel data highly similar to the data from the training dataset. More details of GAN can be found in the original GAN \[20\] paper. Quantum generative adversarial networks (QuGANs) \[63, 65\] are first introduced around late July 2018. Dallaire-Demers and Killoran \[65\] extended adversarial training to the quantum domain, built generative adversarial networks using quantum circuits, and demonstrated that QuGANs can be trained successfully. Lloyd and Weedbrook \[63\] demonstrated that when the data consists of samples of measurements made on high-dimensional spaces, QuGANs may exhibit an exponential advantage over classical GANs.

In the classical GAN \[20\] model, the inputs to the generator are randomly sampled from a distribution, e.g., uniform distribution or Gaussian distribution, and the generator and discriminator consist of neural networks. To demonstrate the quantum advantage of different components of GAN in small drug discovery, we alter each component into a variational quantum circuit (VQC) step by step. In the first experiment, we replace the sampling part of GAN with a VQC to generate the noise for the generator. In the second experiment, we substitute the classical generator of GAN with a VQC. This quantum generator takes the noises from the Gaussian distribution as input and outputs the molecular graph to the classical discriminator. In the last experiment, we replace the classical discriminator with a VQC. This quantum discriminator takes the molecular graph as input and predicts if the molecules are real or fake using the measurement from only one qubit.

In this work, MolGAN \[33\] is used as the base model. MolGAN with VQC (QuMolGAN) as the noise generator, MolGAN with a quantum generator \[64\] (MolGAN-QC), and MolGAN with a
quantum discriminator (MolGAN-CQ) are used to demonstrate the quantum advantage in the small molecule drug discovery. All the combinations of the classical/quantum noise/generator/discriminator and their corresponding model name are shown in Table I. The whole pipeline is implemented by using PennyLane [67] and PyTorch [68]. The details of VQC, VQC of the noise generator, VQC of the generator, and VQC of the discriminator will be introduced as follows.

### TABLE I: All the combinations of the classical/quantum noise/generator/discriminator and their corresponding model name.

| Model Name | Noise | Generator | Discriminator |
|------------|-------|-----------|---------------|
| MolGAN [33] | classical | classical | classical graph-based |
| QuMolGAN | quantum | classical | classical graph-based |
| MolGAN-QC | classical | quantum | classical graph-based |
| MolGAN-CQ | classical | classical | quantum |
| MolGAN-CC | classical | classical | classical MLP-based |

### A. Variational Quantum Circuit (VQC)

A Quantum gate is a basic quantum circuit operating on a small number of qubits. In this work, two controlled gates (Controlled-X gate and Controlled-Z gate) and four rotation operator gates ($R_x$, $R_y$, $R_z$, and $R$) are used to construct the variational quantum circuit. Controlled-X (CNOT) gate is a two-qubit operation, where the first qubit is usually referred to as the control qubit and the second qubit as the target qubit.

\[
CNOT = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]  

(1)

Controlled-Z (CZ) gate is a two-qubit operation defined as:

\[
CZ = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\]  

(2)

The $R_x$, $R_y$, and $R_z$ gates are the essential rotation operators in the quantum circuit. The $R_x$ gate is a single-qubit rotation through an angle $\theta$ in radians around the x-axis, and it is defined as:

\[
R_x(\theta) = \begin{pmatrix}
\cos(\theta/2) & -\sin(\theta/2) \\
-\sin(\theta/2) & \cos(\theta/2)
\end{pmatrix}
\]  

(3)

The $R_y$ gate is a single-qubit rotation through an angle $\theta$ in radians around the y-axis, and it is defined as:

\[
R_y(\theta) = \begin{pmatrix}
\cos(\theta/2) & -\sin(\theta/2) \\
\sin(\theta/2) & \cos(\theta/2)
\end{pmatrix}
\]  

(4)
The $R_z$ gate is a single-qubit rotation through an angle $\theta$ in radians around the z-axis, and it is defined as:

$$R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$ (5)

The $R$ gate is a single-qubit rotation through an arbitrary angles $\alpha, \beta, \gamma$ in radians, and it can be decomposed into $R_y$ and $R_z$ gates. It is defined as:

$$R(\alpha, \beta, \gamma) = R_z(\gamma)R_y(\beta)R_z(\alpha) = \begin{pmatrix} e^{-i\frac{(\alpha+\gamma)}{2} \cos(\beta/2)} & -e^{-i\frac{(\alpha-\gamma)}{2} \sin(\beta/2)} \\ e^{-i\frac{(\alpha-\gamma)}{2} \sin(\beta/2)} & e^{i\frac{(\alpha+\gamma)}{2} \cos(\beta/2)} \end{pmatrix}$$ (6)

A variational quantum circuit (VQC) shown in Figures 2 and 4 consists of three ingredients: (1) the preparation of a fixed initial state, (2) a quantum circuit, and (3) the measurement. The initialization layer may contain $R_x$, $R_y$, $R_z$, and $R$ gates, and the rotation angles are sampled from a uniform distribution or Gaussian distribution. The parameterized layers which could be repeated for $L$ times could have CNOT gates, CZ gates, and parameterized rotational gates whose parameters (rotation angle) can be learned through the back-propagation. The measurement takes the expectation value of each qubit.

### B. VQC of Noise Generator

In this work, MolGAN [33] is used as the base model for small molecule generation. We extend its noise generation part to the quantum domain and demonstrate the quantum advantage in the small molecule generation. MolGAN [33] is an implicit and likelihood-free generative model for small molecular graph generation. In contrast to the sequence-based models, MolGAN directly works on the graph representation of molecules. MolGAN also bypasses the requirement of expensive graph-matching procedures and node-ordering heuristics of likelihood-based methods. In the classical GANs [20], the inputs to the generator are sampled from a distribution, e.g., uniform distribution or Gaussian distribution. Here, we would like to demonstrate the quantum advantage in the small molecule generation by utilizing a variational quantum circuit to generate the inputs of the generator, and we call this hybrid model QuMolGAN. The schema of MolGAN and QuMolGAN is shown in Figure 1. The generator takes the noise as input and generates the molecular graph including the atom vector and bond matrix. The noise is sampled from the Gaussian distribution for the MolGAN, and the noise is generated from the VQC in Figure 2 for the QuMolGAN. The discriminator tries to distinguish between the fake molecular graph from the generator and the real molecular graph from the data distribution. More details of MolGAN can be found in the original MolGAN [33] paper.

### C. VQC of Quantum Generator

We implement the patch method [64] in the quantum generator of MolGAN (MolGAN-QC). This method uses multiple VQCs as sub-generators, and each sub-generator is responsible for constructing a partial part of the final output, e.g., the molecular graph in this study. The final molecular graph which consists of the atom vector and bond matrix is constructed by concatenating all the partial patches together as shown in Figure 3. The sub-generator shares the same ansatz architec-
FIG. 1: Schema of MolGAN and QuMolGAN. The generator takes the noise as input and generates the molecular graph including the atom vector and bond matrix. The discriminator tries to distinguish between the fake molecular graph from the generator and the real molecular graph from the data distribution.

FIG. 2: The VQC consists of the preparation of the initialization state, a single layer of the 4-qubit ansatz circuit, and the measurement. The initialization layer contains $R_y$ and $R_z$ gates, and the rotation angles ($z_1$ and $z_2$) are sampled from a uniform distribution. The parameterized layers (could be repeated for $L$ times) have CNOT gate and two types of parameterized rotational gates, $R_y$ and $R_z$ gates whose parameters ($\theta_k$) can be learned through back-propagation. The measurement takes the expectation value of each qubit.
FIG. 3: The patch method [64] uses multiple VQCs as sub-generators. Each sub-generator takes noise as input and outputs a partial part of the final molecular graph. The final molecular graph is constructed by concatenating all the partial patches together.

Initialization Layer

Parameterized Layers (Repeated for \(L\) times)

\[
|0\rangle \rightarrow R_y(z_1) R_y(\theta_1) \rightarrow|0\rangle \\
|0\rangle \rightarrow R_y(z_2) R_y(\theta_2) Z \rightarrow|0\rangle \\
|0\rangle \rightarrow R_y(z_3) R_y(\theta_3) Z \rightarrow|0\rangle \\
|0\rangle \rightarrow R_y(z_4) R_y(\theta_4) Z \rightarrow|0\rangle
\]

FIG. 4: The VQC of the quantum generator consists of the preparation of the initialization state, a single layer of the 4-qubit circuit, and the measurement. The initialization layer contains \(R_y\) gates, and the rotation angles \((z_i)\) are sampled from a uniform distribution. The parameterized layers (could be repeated for \(L\) times) have CZ gates and one type of parameterized rotational gates, \(R_y\) whose parameters \((\theta_k)\) can be learned through back-propagation. The measurement takes the expectation value of each qubit.

D. VQC of Quantum Discriminator

The quantum discriminator takes the molecular graph as input and classifies if this molecular graph is fake (from the generator) or real (from the data distribution). The VQC of quantum discriminator consists of the amplitude encoding layer [69] \((S_x)\), the strongly entangling layers \((U_\theta)\) inspired by [70] and the measurement as shown in Figure 5. Amplitude encoding is used to encode the atom vector and bond matrix. The strongly entangling layers [70] have multiple CNOT gates and parameterized rotational gates \(R(\alpha, \beta, \gamma)\) as shown in Figure 6. In each layer, each qubit starts with parameterized rotational gates \(R(\alpha_i, \beta_i, \gamma_i)\) followed by a CNOT gate. The parameterized angles \(\alpha_i, \beta_i, \gamma_i\) can be learned through back-propagation. The measurement takes the expectation value of one qubit, and this value is used to determine if the input molecular graph is real or fake. In our experiment, we use 9 qubits to encode the molecular graph and three-layer of strongly
entanglement layers.

![Amplitude Embedding and Strongly Entangling Layers](image)

FIG. 5: The VQC of quantum discriminator consists of the amplitude embedding circuit \( S_x \), the strong entanglement layers \( U_\theta \), and the measurement. The strongly entangling layers \( U_\theta \) have multiple CNOT gates and parameterized rotational gates \( R \). The measurement takes the expectation value of one qubit to determine if the input molecular graph is real or fake. It is noted that we use 9 qubits to encode the molecular graph in our experiment.

![Strongly Entangling Layers](image)

FIG. 6: The VQC of strongly entanglement layers \( U_\theta \) contain multiple CNOT gates and parameterized rotational gates \( R \). In each layer, each qubit starts with parameterized rotational gates \( R(\alpha, \beta, \gamma) \) followed by a CNOT gate. The parameterized angles \( \alpha, \beta, \gamma \) can be learned through back-propagation. In our experiment, our ansatz structure has 9 qubits and three layers.

### III. EVALUATION METRICS AND DATASET

In this section, we will first introduce the evaluation method and metrics followed by the dataset we used to train the generative models.

#### A. Evaluation Method and Metrics

To evaluate the performance of different generative models, we first generate 5,000 noise samples from Gaussian distribution and variational quantum circuits for MolGAN and QuMolGAN, respectively. The generated noise samples are then fed into the trained generator of models to produce the atom vectors and bond matrices. In the end, these vectors and matrices are used to construct the molecular graphs. Seven evaluation metrics described below will be calculated from the molecular graphs.
Three quality metrics, i.e., validity, uniqueness, and novelty used in [71, 72], and three drug properties, i.e., quantitative estimation of drug-likeness (QED), solubility, and synthesizability (SA), are used to compare different generative models in this work. Validity is the ratio of the valid molecules to all generated molecules, and uniqueness is the ratio of unique molecules to the valid molecules. Novelty is defined as the ratio of valid molecules which are not in the training dataset to all valid molecules. In addition, we also measure the diversity of generated molecules which is defined as how likely the generated molecules are to be diverse to the training dataset. We also report the valid and unique molecules (# molecules) from 5,000 noise samples. QED [73] measures how likely a molecule is to be a drug based on the concept of desirability. Solubility reports the n-octanol-water partition coefficient (logP) [74] of the molecule that is the degree of a molecule being hydrophilic. SA [75] quantifies how easy a molecule is to be synthesized based on the molecular complexity and fragment contributions. These property metrics are calculated by using RDKit [76].

The Kullback–Leibler (KL) divergences [77] are also calculated, and it measures how well a probability distribution approximates another distribution. The probability distributions of a variety of physicochemical descriptors including BertzCT (molecular complexity index), MolLogP (Wildman-Crippen LogP value [74]), MolWt (molecular weight), TPSA (molecular polar surface area), NumHAcceptors (number of hydrogen acceptors), NumHDonors (number of hydrogen donors), NumRotatableBonds (number of rotatable bounds), NumAliphaticRings (number of aliphatic rings), and NumAromaticRings (number of aromatic rings) for the generated molecules and the molecules of the training set are compared, and the corresponding KL-divergence scores $D_{KL,i}$ are computed. Models able to capture the distributions of molecules in the training set will lead to small KL-divergence scores ($D_{KL}$). However, the final KL-divergence score ($S$) [78] used in this paper is computed by

$$S = \frac{1}{9} \sum_{i=1}^{9} \exp(-D_{KL,i})$$

(7)

Therefore, the larger final KL-divergence score ($S$) means how well the model can capture these nine physicochemical distributions of molecules in the training set.

### B. Dataset

All experiments of this work use the QM9 (Quantum Machines 9) [79] dataset. QM9 dataset is curated from GDB-17 chemical database [80] which has approximate 166.4 billion molecules. QM9 consists of 133,171 molecules containing less than or equal to nine non-hydrogen atoms (carbon, nitrogen, oxygen, and fluorine). Figure 7 shows some molecules from the QM9 dataset, and Table II summarizes the drug properties of all molecules of the QM9 dataset.

| # molecules | QED   | Solubility | SA  |
|-------------|-------|------------|-----|
| 133,171     | 0.461 | 0.289      | 0.327 |
FIG. 7: Example molecules of QM9 with canonical SMILES visualized by using RDKit [76].

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we would like to discover the quantum advantage in small molecule discovery. We tried to find the best hyper-parameters for the base MolGAN. We first examined different complexities of generators (in Appendix A) and observed that MolGAN-HR (high reduction) has the best performance compared to other generator complexities. We then examined different input noise dimensions of generators (in Appendix B) and found out that the number of unique and valid molecules saturated at the input dimension equaling 4 (z_dim=4). In addition, we also tested different numbers of parameterized layers in the variational quantum circuit (VQC) (in Appendix C) and observed that MolGAN-HR with 3 parameterized layers of VQC as noise generator has the best performance compared to other numbers of parameterized layers. Therefore, in the following experiments, MolGAN-HR is used as the base model, and VQC with 3 parameterized layers is used as the base quantum circuit. In the first experiment, we substitute the noise generator of MolGAN with a VQC and discover the quantum advantage in the generated molecules with better drug properties. In the second experiment, we replace the generator of MolGAN with a VQC and show the potential of generating small molecules using a VQC. In the third experiment, we supplant the discriminator of MolGAN with a VQC and demonstrate the quantum advantage. All the combinations of the classical/quantum noise/generator/discriminator and their corresponding model name are shown in Table I. All experiments are implemented by using Pennylane [67] and PyTorch [68].
TABLE III: Comparing generated molecules of QuMolGAN and MolGAN in drug properties. Bold numbers highlight the better scores in QuMolGAN compared to the corresponding MolGAN. It is noted that the QED, Solute, and SA scores in this table are calculated from the valid and unique molecules.

| $z_{\text{dim}}$ | QuMolGAN # molecules | MolGAN # molecules | p-value | QuMolGAN # molecules | MolGAN # molecules | p-value | QuMolGAN # molecules | MolGAN # molecules | p-value | QuMolGAN # molecules | MolGAN # molecules | p-value |
|------------------|----------------------|-------------------|---------|----------------------|-------------------|---------|----------------------|-------------------|---------|----------------------|-------------------|---------|
| $z_{\text{dim}}=2$ | 785 | 1,119 | 0.031 | 785 | 1,119 | 0.031 | 785 | 1,119 | 0.031 |
| $z_{\text{dim}}=3$ | 683 | 1,011 | 0.015 | 683 | 1,011 | 0.015 | 683 | 1,011 | 0.015 |
| $z_{\text{dim}}=4$ | 575 | 910 | 0.001 | 575 | 910 | 0.001 | 575 | 910 | 0.001 |
| $z_{\text{dim}}=8$ | 468 | 710 | 0.001 | 468 | 710 | 0.001 | 468 | 710 | 0.001 |

A. Quantum Noise Generator

In the first experiment, we would like to compare the performance of QuMolGAN and classical MolGAN. We use the same hyper-parameters to train the QuMolGAN and classical MolGAN except for the learning rate. The learning rate of the quantum noise generator is 0.04, and the learning rate for the generator and discriminator is 0.001. The models are trained for 150 epochs. The WGAN (Wasserstein generative adversarial networks) loss [81] is used to train the models. QuMolGAN with three parameterized layers is used in this experiment. We have examined the input dimension of the generator at 2, 3, 4, and 8, and the results are shown in Table III. It is noted that the QED, Solute, and SA scores in this table are calculated from the valid and unique molecules. The QuMolGAN is lacking in generating as many molecules as classical MolGAN which results in worse KL Scores. However, the QuMolGAN can generate molecules with significantly ($p < 0.05$) better drug properties compared to the classical MolGAN particularly when the input noise dimension is small ($z_{\text{dim}}=2$ and $z_{\text{dim}}=3$). The drug properties distributions of MolGAN-generated and QuMolGAN-generated molecules are shown in Figure 8 for $z_{\text{dim}}=2$. It shows that QuMolGAN can generate molecules with better drug properties, particularly in QED. QuMolGAN has less probability to generate molecules whose QED is less than 0.4. For the KL-divergence task, the MolLogP, BertzCT, and MolWt distributions of generated molecules are also shown in Figure 9 for $z_{\text{dim}}=2$. The classical MolGAN (blue) tends to generate molecules with similar distributions to the training set (grey) which results in a better KL-divergence score.

In the end, we randomly sample 32 valid and unique molecules from both MolGAN and QuMolGAN for $z_{\text{dim}}=2$, and the example molecules are shown in Figure 10. QuMolGAN can generate training-set-like molecules with better drug properties.

As mentioned in [78], the goal-directed optimization of molecules tries to improve the demanded scores for the generated molecules. These scores reflect how well molecules satisfy the required properties. The goal is to find molecules that maximize the scoring function. In this experiment, we also would like to check if the quantum circuit can bring advantages to the MolGAN in the goal-directed benchmark. Therefore, we add a reward network into the original schema, and the updated schema is shown in Figure 11. At this time, the generator is trained using a linear combination of the WGAN [81] loss and the RL [82] (reinforcement learning) loss:

$$L(\omega) = \alpha \cdot L_{\text{WGAN}}(\omega) + (1 - \alpha) \cdot L_{\text{RL}}(\omega)$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the trade-off between WGAN loss and RL loss, and $\omega$ are the inputs to networks. Here, we set $\alpha = \{0.5, 0.01\}$ to weigh the loss between
two components and the goal of improving SA, and QED to train MolGAN and QuMolGAN as suggested by the medical chemists. In addition, we also add the unique score into the goal to prevent the model from generating the same molecules. We have trained MolGAN and QuMolGAN with ($\alpha = \{0.5, 0.01\}$) and without ($\alpha = 1.0$) RL loss using the same hyperparameters for 150 epochs, and the results are shown in Table IV. The input noise dimension to the generator is 4 ($z_{\text{dim}} = 4$). It is noted that the QED, Solute, and SA scores in this table are calculated from the valid molecules. In this table, the KL Scores of MolGAN are always greater than their quantum counterpart in different weights of RL loss. However, QuMolGAN has ability to achieve higher goal (from 0.47 to 0.57 in QED and from 0.29 to 0.76 in SA) compared to MolGAN (from 0.47 to 0.52 in QED and from 0.30 to 0.60 in SA) while $\alpha = 0.01$. The solute is not the goal so these scores are close for MolGAN and QuMolGAN.
In the second experiment, we try to discover the advantage of the quantum circuit in the generator of GAN. We have tried to substitute the generator of MolGAN with a VQC described in [64] for the small molecule generation. The performance of MolGAN with the quantum generator (MolGAN-
FIG. 11: Schema of MolGAN and QuMolGAN with reward network in the goal-directed benchmark. The generator takes the noise as input and generates the molecular graph including the atom vector and bond matrix. The discriminator tries to distinguish between the fake molecular graph from the generator and the real molecular graph from the data distribution. The reward network tries to guide the generator to generate the molecules with desired properties.

TABLE IV: Performance comparison between MolGAN and QuMolGAN with ($\alpha = \{0.5, 0.01\}$) and without ($\alpha = 1.0$) reinforcement learning loss in the goal-directed benchmark. It is noted that the QED, Solute, and SA scores in this table are calculated from the valid molecules. Bold numbers indicate better scores among the same type of models with different RL weights $\alpha$, and the underlined numbers indicate the best scores across different types of models.

|                  | MolGAN     | QuMolGAN   |
|------------------|------------|------------|
| $\alpha = 1.0$   | $\alpha = 0.5$ | $\alpha = 0.01$ |
| #of molecules$^a$| 2890       | 2700       |
| Validity ↑       | 80.40      | 78.48      |
| Uniqueness ↑     | 71.89      | 68.81      |
| QED ↑            | 0.47       | 0.48       |
| Solute ↑         | 0.31       | 0.31       |
| SA ↑             | 0.30       | 0.31       |
| KL Score (S)$^b$ | 0.95       | 0.94       |

$^a$ Number of valid and unique molecules from 5,000 samples;
$^b$ From Equation (7)

QC) is reported in Table V. Although the integration works smoothly, the training processing is time-consuming and resource-consuming. The average training time per step takes around 39 seconds in the Amazon EC2 C6a Metal Instance which results in approximately 3.5 days per epoch. In addition, the model has difficulty generating more valid and unique molecules after being trained for ten epochs. It fails to generate the training-data-like molecules even after ten epochs of training. Random-picked and cherry-picked examples of generated molecules are shown in Figure 12. Most generated molecules are similar to the randomly picked molecules in Figure 12. However, we
TABLE V: Performance of MolGAN with the quantum generator. The QED, Solute, and SA scores in this table are calculated from the valid molecules.

| # epoch | # molecules | validity | uniqueness | novelty | diversity | QED | Solubility | SA | KL Score (S) |
|---------|-------------|----------|------------|---------|-----------|-----|------------|----|-------------|
| 1       | 73          | 79.39    | 4.49       | 100     | 1.00      | 0.43| 0.75       | 0.24| 0.24        |
| 2       | 54          | 76.37    | 3.45       | 100     | 1.00      | 0.47| 0.75       | 0.24| 0.25        |
| 3       | 43          | 78.47    | 2.68       | 100     | 1.00      | 0.48| 0.75       | 0.11| 0.28        |
| 4       | 29          | 78.61    | 1.80       | 100     | 1.00      | 0.48| 0.75       | 0.13| 0.29        |
| 5       | 30          | 77.93    | 1.88       | 100     | 1.00      | 0.48| 0.75       | 0.14| 0.30        |
| 6       | 40          | 78.37    | 2.49       | 100     | 1.00      | 0.48| 0.75       | 0.09| 0.21        |
| 7       | 29          | 80.27    | 1.76       | 100     | 1.00      | 0.47| 0.75       | 0.06| 0.28        |
| 8       | 39          | 78.91    | 2.41       | 100     | 1.00      | 0.48| 0.75       | 0.16| 0.28        |
| 9       | 41          | 74.66    | 2.68       | 100     | 1.00      | 0.48| 0.75       | 0.22| 0.25        |
| 10      | 29          | 79.74    | 1.78       | 100     | 1.00      | 0.48| 0.75       | 0.08| 0.27        |

* Number of valid and unique molecules from 2,048 samples from Gaussian distribution;  
^ From Equation [7];

FIG. 12: Example molecules from the quantum generator. We input 2048 noises to the quantum generator of MolGAN for every epoch (from #1 to #10). The first row shows the random pick of generated molecules, and the second row shows the cherry-pick of generated molecules.

demonstrate that the quantum generator has the potential to generate small molecules.
C. Quantum Discriminator

In addition to substituting the generator architecture, the architecture with a quantum discriminator combined with the classical generator described in MolGAN has been experimented with. Our goal is to see if the quantum discriminator shows any advantage over its classical counterparts. We have found that the number of learnable parameters is significantly less than the classical ones while the model can still generate valid molecules. Furthermore, to conduct a fair comparison between the classical discriminator and the quantum one, we changed the MolGAN classical graph-based discriminator to the multiple layer perception (MLP) architecture, which is more similar to our proposed quantum discriminator architecture, reduced the number of its training parameters, and found that quantum discriminator with 50 training parameters outperforms the classical discriminator with 22,500 parameters in terms of generated molecule properties and KL-divergence. For simplicity, all the combinations of the classical/quantum noise/generator/discriminator are listed in Table I. In this experiment, the noises of the generator are always sampled from the Gaussian distribution.

In the following subsections, we first present our proposed MolGAN-CQ and the training details. Second, we compare the results of MolGAN-CQ with MolGAN. In the end, we made architecture modifications from the Graph-Classical-Discriminator in MolGAN to the MLP-Classical-Discriminator (MolGAN-CC) and reduced the amount of the learnable parameters to make a fair comparison with MolGAN-CQ.

1. MolGAN-CQ Architecture and Training Details

As illustrated in Figure 13, the model consists of two components: classical generator and quantum discriminator. To encode two generator outputs, a bond matrix, and an atom vector, into the quantum discriminator efficiently, we first flatten the bond matrix into a vector, which is subsequently concatenated with the atom vector to a new vector with the size of 450. After that, the resulting vector will be the input of the Q-Discriminator, composed of an amplitude embedding layer followed by 3 strongly entangling layers as described in Figure 5.

For the training details, at first, we followed the best hyper-parameters set in MolGAN. However, applying the same training details on MolGAN-CQ makes the training process hard to converge. Therefore, we change the learning rate from $1 \times 10^{-3}$ to $1 \times 10^{-4}$ for the generator to make it more stable. Moreover, we have experimented with the alternating training times between the classical generator and quantum discriminator with several sets, including $(G, D) = (1, 5), (1, 8), (1, 10)$. We found out that training 1 step C-Generator followed by 10 steps Q-Discriminator stabilizes the training process. The other hyperparameters remain identical to the ones in MolGAN.

2. Comparison with classical MolGAN

In this section, we compare MolGAN-CQ with MolGAN. We trained MolGAN according to the best hyper-parameter sets found in the previous section. Notice here, since training a quantum network is time-consuming, and the loss curve from classical MolGAN at epoch 30 shows the trend of convergence, both MolGAN-CQ and MolGAN were only trained to epoch 30 and evaluated at epoch 30 instead of epoch 150. After that, both models generate 5000 samples to do a further comparison. Table VI demonstrates that MolGAN-CQ can generate valid and drug-like molecules. In addition, MolGAN-CQ can generate molecules with better drug properties, particularly in Solute
FIG. 13: Schema of MolGAN-CQ. One of the generator outputs, the bond matrix, is flattened to a vector and concatenated with the other generator outputs, the atom vector, resulting in a vector with the size of 1x450, which is fed into the quantum discriminator.

and SA. However, compared to classical MolGAN, MolGAN-CQ does not have an advantage in KL-divergence with training data probability. Figure 14 shows some molecules generated from MolGAN-CQ.

TABLE VI: Performance comparison between MolGAN and MolGAN-CQ. The models are only trained for 30 epochs. Bold numbers indicate better scores. The QED, Solute, and SA scores in this table are calculated from the valid molecules.

|                  | MolGAN | MolGAN-CQ |
|------------------|--------|-----------|
| # of molecules   | 2,693  | 730       |
| QED ↑            | 0.47   | 0.48      |
| Solute ↑         | 0.31   | 0.44      |
| SA ↑             | 0.31   | 0.66      |
| KL Score (S)     | 0.94   | 0.75      |

a Number of valid and unique molecules from 5,000 samples; b From Equation (7)

3. Comparison with MolGAN-CC

In this section, MolGAN-CQ is compared with MolGAN-CC with different numbers of hidden layers to evaluate MolGAN-CQ’s capacity. To have a fair comparison, we modified the original graph-based network to multiple-layer perception (MLP). Under this condition, the input of the MLP-based discriminator would be the same as the one in MolGAN-CQ’s quantum discriminator, a flattened vector instead of a graph. The network architecture of MolGAN-CC is shown in Figure 15.

Furthermore, since the discriminator of MolGAN-CQ only has 50 learnable parameters, we have tried to reduce the discriminator size of MolGAN-CC as small as possible for a fair and reasonable
The number of the hidden layers and neurons varies in the classical MLP-based discriminator as shown in Table VII. Therefore, we have tried MolGAN-CC with 3 different parameter sizes of discriminator as shown in Table VII.

Table VII demonstrates that although MolGAN-CC-NR and MolGAN-CC-HR have a higher capacity to generate molecules whose molecule properties are more similar to the training data, MolGAN-CQ with only 50 parameters can achieve an outstanding performance compared to
TABLE VII: The details of MolGAN-CC models with the varied size of discriminators.

| # of layer in the MolGAN-CC discriminator | parameter size | # of layer in the MolGAN-CC discriminator | parameter size |
|------------------------------------------|----------------|------------------------------------------|----------------|
| MolGAN-CC-ER\textsuperscript{a}          | 3              | (450, 50, 1)                              | 22K            |
| MolGAN-CC-HR\textsuperscript{b}          | 3              | (450, 100, 1)                             | 45K            |
| MolGAN-CC-NR\textsuperscript{c}          | 4              | (450, 150, 50, 1)                         | 82K            |

\textsuperscript{a} Extremely-reduction; \textsuperscript{b} Highly-reduction \textsuperscript{c} No-reduction;

MolGAN-CC-ER with 22K parameters in terms of KL-score which shows the quantum advantage in the expression power. The distributions of molecular properties of MolLogP, MolWt, and BertzCT, generated from MolGAN-CC-ER, MolGAN-CQ, MolGAN, and QM9 are shown in Figure 16. As we can see from Figure 16, molecular properties from MolGAN are more close to the ones from the training data, QM9, compared to MolGAN-CQ and MolGAN-CC-ER. Nevertheless, MolGAN-CQ with only 50 parameters could generate molecules with similar distribution to the training data in comparison to MolGAN-CC-ER with 22K parameters.

TABLE VIII: Performance comparison between MolGAN-CQ and MolGAN-CC with 3 different sizes of the MLP-based discriminator. Bold numbers indicate better performance across different types of models. The QED, Solute, and SA scores in this table are calculated from the valid molecules.

|                  | MolGAN-CQ | MolGAN-CC-ER | MolGAN-CC-HR | MolGAN-CC-NR |
|------------------|-----------|--------------|--------------|--------------|
| # of parameters  | 50        | 22K          | 45K          | 82K          |
| # of molecules\textsuperscript{a} | 2,693     | 104          | 1919         | 2284         |
| QED ↑            | 0.47      | 0.51         | 0.49         | 0.5          |
| Solute ↑         | 0.44      | 0.63         | 0.35         | 0.38         |
| SA ↑             | 0.66      | 0.97         | 0.48         | 0.50         |
| KL Score (S)\textsuperscript{a}↑ | 0.75      | 0.28         | 0.84         | 0.81         |

\textsuperscript{a} Number of valid and unique molecules from 5,000 samples; \textsuperscript{b} From Equation (7)

V. CONCLUSION

In this paper, we have explored the quantum advantage in small molecule drug discovery by substituting each part of MolGAN [33] with a VQC step by step and compared its performance with the classical counterpart. In the first experiment, using a VQC as a noise generator to the classical GAN can generate small molecules with better drug properties but a lower KL Score. In addition, this hybrid model achieves better performance in the goal-directed benchmark compared to the classical counterpart. In the second experiment, we also demonstrated the potential of generating training-set-like small molecules using a quantum generator. However, the training processing is resource-consuming and time-consuming even in the advanced classical computer. In the third experiment, the VQC in the quantum discriminator outperforms the classical counterpart in terms of generated molecule properties and the KL score. We also demonstrated that the hybrid model...
FIG. 16: Molecular properties of generated molecules. Generated molecules from MolGAN-CC-ER, MolGAN-CQ, and MolGAN were analyzed in each column with the molecular properties of MolLogP, MolWt, and BertzCT. In each graph, the histogram with blue indicates the distribution of the training data QM9.

could generate valid molecules with only tens of learnable parameters in a quantum discriminator.

[1] A. D. Hingorani, V. Kuan, C. Finan, F. A. Kruger, A. Gaulton, S. Chopade, R. Sofat, R. J. MacAllister, J. P. Overington, H. Hemingway, et al., Scientific reports 9, 1 (2019).
[2] J. L. Abreu, Revista Daena (International Journal of Good Conscience) 15 (2020).
[3] O. J. Wouters, M. McKee, and J. Luyten, Jama 323, 844 (2020).
[4] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov, Molecular pharmaceutics 13, 2524 (2016).
[5] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, and A. Zhavoronkov, Oncotarget 8, 10883 (2017).
[6] A. Zhavoronkov, Artificial intelligence for drug discovery, biomarker development, and generation of novel chemistry (2018).
[7] P. Carracedo- Reboredo, J. Liñares-Blanco, N. Rodríguez-Fernández, F. Cedrón, F. J. Novoa, A. Carballal, V. Maojo, A. Pazos, and C. Fernandez-Lozano, Computational and Structural Biotechnology Journal 19, 4538 (2021).
[8] P.-Y. Kao, S.-M. Kao, N.-L. Huang, and Y.-C. Lin, in 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (IEEE, 2021) pp. 2384–2391.
[9] S. Kolluri, J. Lin, R. Liu, Y. Zhang, and W. Zhang, The AAPS Journal 24, 1 (2022).
[10] A. Zhavoronkov, Y. A. Ivanenko, A. Aliper, M. S. Veselov, V. A. Aladinskaya, A. V. Aladinskaya, V. A. Terentiev, D. A. Polykovskiy, M. D. Kuznetsov, A. Asadulaev, et al., Nature biotechnology 37, 1038 (2019).
[11] H. S. Chan, H. Shan, T. Dahoun, H. Vogel, and S. Yuan, Trends in pharmacological sciences 40, 592 (2019).
[12] G. Schneider and U. Fechner, Nature Reviews Drug Discovery 4, 649 (2005).
[13] T. Fischer, S. Gazzola, and R. Riedl, Expert opinion on drug discovery 14, 791 (2019).
[14] V. D. Mouchlis, A. Afantitis, A. Serra, M. Fratello, A. G. Papadimitrakis, V. Aidinis, I. Lynch, D. Greco, and G. Melagragi, International journal of molecular sciences 22, 1676 (2021).
[15] A. Speck-Plancher, Future Medicinal Chemistry 10, 2021 (2018).
[16] P. Mamoshina, O. Kotsias, J. Arús, T. Pollock, S. H. Daly, and B. Manjunath, Integrating Materials and Manufacturing Innovation , 1 (2022).
[17] A. Zhavoronkov, B. Zagribelnyy, A. Zhebrak, V. Aladinskaya, V. Terentiev, Q. Vanhaelen, D. S. Bezrukov, D. Polykovskiy, R. Shayakhmetov, A. Filimonov, et al., ChemRxiv (2020).
[18] D. Paul, G. Sanap, S. Shenoy, D. Kalyane, K. Kalia, and R. K. Tekade, Drug Discovery Today 26, 80 (2021).
[19] D. Martinelli, Computers in Biology and Medicine , 105403 (2022).
[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Advances in neural information processing systems 27 (2014).
[21] T. Karras, T. Aila, S. Laine, and J. Lehtinen, arXiv preprint arXiv:1710.10196 (2017).
[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, in Proceedings of the IEEE conference on computer vision and pattern recognition (2017) pp. 1125–1134.
[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, in Proceedings of the IEEE international conference on computer vision (2017) pp. 2223–2232.
[24] T. Karras, S. Laine, and T. Aila, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2019) pp. 4401–4410.
[25] D. K. Jangid, N. R. Brodnik, A. Khan, M. G. Goebel, M. P. Echlin, T. M. Pollock, S. H. Daly, and B. Manjunath, Integrating Materials and Manufacturing Innovation , 1 (2022).
[26] A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, and A. Zhavoronkov, Molecular pharmaceutics 14, 3098 (2017).
[27] Q. Vanhaelen, Y.-C. Lin, and A. Zhavoronkov, ACS Medicinal Chemistry Letters 11, 1496 (2020).
[28] M. Xu, J. Cheng, Y. Liu, and W. Huang, in 2021 IEEE Symposium on Computers and Communications (ISCC) (IEEE, 2021) pp. 1–6.
[29] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik, arXiv preprint arXiv:1705.10843 (2017).
[30] L. Yu, W. Zhang, and Y. Yu, in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (2017) pp. 2852–2858.
[31] O. Prykhodko, S. V. Johansson, P.-C. Kotsias, J. Arús-Pous, E. J. Bjerrum, O. Engkvist, and H. Chen, Journal of Cheminformatics 11, 1 (2019).
[32] P.-C. Kotsias, J. Arús-Pous, H. Chen, O. Engkvist, C. Tyrchan, and E. J. Bjerrum, Nature Machine Intelligence 2, 254 (2020).
[33] N. De Cao and T. Kipf, arXiv preprint arXiv:1805.11973 (2018).
[34] M. K. Jayatungwa, W. Xie, L. Ruder, U. Schulze, and C. Meier, Nature reviews. Drug Discovery (2022).
[35] F. Neukart, G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and B. Parney, Frontiers in ICT 4, 29 (2017).
[36] S. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. Bernal, and D. Greenberg, IEEE Transactions on Quantum Engineering 2, 1 (2021).
[37] R. Orus, S. Mugel, and E. Lizaso, Reviews in Physics 4, 100028 (2019).
[38] G. Liu and W. Ma, Information Sciences 598, 75 (2022).
[39] A. Kandala, A. Mezzacapo, K. Temme, M. Brink, J. M. Chow, and J. M. Gambetta, Nature 549, 242 (2017).
[40] G. A. Quantum, Collaborators*, F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, et al., Science 369, 1084 (2020).
[41] Q. Gao, H. Nakamura, T. P. Gujarati, G. O. Jones, J. E. Rice, S. P. Wood, M. Pistoia, J. M. Garcia, and N. Yamamoto, The Journal of Physical Chemistry A 125, 1827 (2021).
[42] Y. Shee, T.-L. Yeh, J.-Y. Hsiao, A. Yang, Y.-C. Lin, and M.-H. Hsieh, arXiv preprint arXiv:2210.02977 (2022).
[43] J. Li, R. O. Topaloglu, and S. Ghosh, IEEE Transactions on Quantum Engineering 2, 1 (2021).
[44] R. P. Feynman et al., Int. j. Theor. Phys 21 (1982).
[45] L. K. Grover, in Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (1996) pp. 212–219.
[46] D. Coppersmith, arXiv preprint quant-ph/0201067 (2002).
[47] P. W. Shor, in Proceedings 35th annual symposium on foundations of computer science (Ieee, 1994) pp. 124–134.
[48] A. Y. Kitaev, arXiv preprint quant-ph/9511026 (1995).
[49] V. Giovannetti, S. Lloyd, and L. Maccone, Science 306, 1330 (2004).
[50] V. Giovannetti, S. Lloyd, and L. Maccone, Physical review letters 96, 010401 (2006).
[51] S. Lloyd, Science 321, 1463 (2008).
[52] P. Komar, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin, Nature Physics 10, 582 (2014).
[53] R. Chao and B. W. Reichardt, npj Quantum Information 4, 1 (2018).
[54] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, et al., Reviews of Modern Physics 94, 015004 (2022).
[55] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al., Nature 574, 505 (2019).
[56] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, et al., Science 370, 1460 (2020).
[57] N. Quesada, J. M. Arrazola, and N. Killoran, Physical Review A 98, 062322 (2018).
[58] X. Gao, E. R. Anschuetz, S.-T. Wang, J. I. Cirac, and M. D. Lukin, arXiv preprint arXiv:2101.08354 (2021).
[59] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, IEEE Access 8, 141007 (2020).
[60] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Physical Review Research 2, 033125 (2020).
[61] Y. Du, M.-H. Hsieh, T. Liu, S. You, and D. Tao, PRX Quantum 2, 040337 (2021).
[62] Y. Du, M.-H. Hsieh, T. Liu, D. Tao, and N. Liu, Physical Review Research 3, 023153 (2021).
[63] S. Lloyd and C. Weedbrook, Physical review letters 121, 040502 (2018).
[64] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, et al., Physical Review Applied 16, 024051 (2021).
[65] P.-L. Dallaire-Demers and N. Killoran, Physical Review A 98, 012324 (2018).
[66] https://github.com/jundeli/quantum-gan
[67] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, et al., arXiv preprint arXiv:1811.04968 (2018).
[68] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, NIPS 2017 Autodiff Workshop (2017).
[69] M. Schuld and F. Petruccione, Supervised learning with quantum computers, Vol. 17 (Springer, 2018).
[70] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Physical Review A 101, 032308 (2020).
[71] B. Samanta, A. De, N. Ganguly, and M. Gomez-Rodriguez, arXiv preprint arXiv:1802.05283 (2018).
[72] D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, et al., Frontiers in pharmacology 11, 565644 (2020).
[73] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins, Nature chemistry 4, 90 (2012).
[74] S. A. Wildman and G. M. Crippen, Journal of chemical information and computer sciences 39, 868 (1999).
[75] P. Ertl and A. Schuffenhauer, Journal of cheminformatics 1, 1 (2009).
[76] G. Landrum, P. Tosco, B. Kelley, R. Vianello, NadineSchneider, E. Kawashima, A. Dalke, D. N. D. Cosgrove, B. Cole, M. Swain, S. Turk, AlexanderSavelyev, G. Jones, A. Vaucher, M. Wójcikowski, I. Take, D. Probst, K. Ujihara, V. F. Scalfani, guillaume godin, A. Pahl, F. Berenger, JLVarjo, strets123, JP, and DoliathGavid, rdkit/rdkit: 2022031 (q1 2022) release (2022).
[77] S. Kullback and R. A. Leibler, The annals of mathematical statistics 22, 79 (1951).
[78] N. Brown, M. Fiscato, M. H. Segler, and A. C. Vaucher, Journal of chemical information and modeling 59, 1096 (2019).
[79] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, Scientific data 1, 1 (2014).
[80] L. Ruddigkite, R. Van Deursen, L. C. Blum, and J.-L. Reynold, Journal of chemical information and modeling 52, 2864 (2012).
[81] M. Arjovsky, S. Chintala, and L. Bottou, in International conference on machine learning (PMLR, 2017) pp. 214–223.
[82] L. P. Kaelbling, M. L. Littman, and A. W. Moore, Journal of artificial intelligence research 4, 237 (1996).
[83] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, The journal of machine learning research 15, 1929 (2014).
[84] M. Padala, D. Das, and S. Gujar, in International Conference on Neural Information Processing (Springer, 2021) pp. 558–569.
[85] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Proceedings of the IEEE 86, 2278 (1998).
[86] Z. Liu, P. Luo, X. Wang, and X. Tang, in Proceedings of the IEEE international conference on computer vision (2015) pp. 3739–3738.
[87] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Quantum Science and Technology 4, 043001 (2019).
[88] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Advanced Quantum Technologies 2, 1900070 (2019).
[89] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nature communications 9, 1 (2018).
Appendix A: Different Generator Complexities

We would like to explore the relationship between the generator complexity and the training data. The generator of MolGAN consists of multiple dense layers (linear layer followed by activation layer and dropout layer [83]). We train three models with different generator complexities. The generator of the first model has the same amount of parameters as the original MolGAN [33]. The generator of the second model only has approximately 15% of the parameters as the original MolGAN, and we call this model MolGAN-HR (high reduction). The generator of the third model merely has approximately 2% of the parameters as the original MolGAN, and we call this model MolGAN-ER (extreme reduction). These three models are trained using the same hyperparameters. The batch size is 128. The models are trained for 150 epochs. The learning rate is 0.001 for both the generator and discriminator. The discriminator is updated three times followed by updating the generator once. We evaluate each model using the generator from 150 epochs, and the results are shown in Table IX. In this table, MolGAN-HR can generate the most valid and unique molecules which results in the highest KL Score. The QED, Solute, and SA scores in this table are calculated from the valid molecules.

| Model       | # molecules | validity | uniqueness | novelty | diversity | QED   | Solubility | SA   | KL Score (S) |
|-------------|-------------|----------|------------|---------|-----------|-------|------------|------|--------------|
| MolGAN a    | 995         | 78.54    | 25.34      | 59.56   | 0.57      | 0.47  | 0.31       | 0.28 | 0.893        |
| MolGAN-HR b | 3191        | 78.34    | 81.47      | 64.33   | 0.59      | 0.47  | 0.31       | 0.29 | 0.954        |
| MolGAN-ER c | 2826        | 75.84    | 74.53      | 67.14   | 0.67      | 0.47  | 0.31       | 0.32 | 0.945        |

---

Appendix B: Varying the Input Noise Dimension of Generator

Padala et al. [84] has examined the effect of input noise dimension in GANs in generating the Gaussian distribution, the handwritten digit recognition [85], and the celebrity image classification [86]. They observed a significant effect on the results when the input noise dimension is alternated. They also claimed that the optimal noise dimension is depended on the dataset and loss function used. Inspired by their work, we would like to examine the optimal input noise dimension for the task of small molecular generation using the QM9 dataset for MolGAN and QuMolGAN. The input noise dimension is varied from 1 to 8 for the MolGAN-HR, and the results are shown in Table X. For QuMolGAN-HR, we only examine the ansatz with 2, 3, 4, and 8 qubits, and the experimental results are shown in Table XI. These models are trained for 150 epochs. The learning rates of the discriminator and generator are 0.001, and the learning rate of the quantum circuit is 0.04. The QED, Solute, and SA scores in this table are calculated from the valid molecules. From these two tables, the number of valid and unique molecules saturates when the input dimension (z_dim) goes beyond 4. The KL Score of MolGAN-HR oscillates between 0.94 and 0.96, and the KL Score of
TABLE X: Different input noise dimensions of generator in MoGAN-HR. The QED, Solute, and SA scores in this table are calculated from the valid molecules.

| z-dim # | molecules<sup>a</sup> | validity↑ | uniqueness↑ | novelty↑ | diversity↑ | QED↑ | Solubility↑ | SA↑ | KL Score (S)<sup>b</sup>↑ |
|---------|------------------------|-----------|-------------|----------|------------|------|-------------|-----|--------------------------|
| 8       | 3204                   | 80.04     | 80.06       | 64.67    | 0.62       | 0.47 | 0.31        | 0.30| 0.946                    |
| 7       | 3206                   | 80.42     | 79.73       | 65.61    | 0.74       | 0.47 | 0.30        | 0.30| 0.957                    |
| 6       | 2898                   | 77.76     | 74.54       | 61.34    | 0.72       | 0.47 | 0.30        | 0.29| 0.937                    |
| 5       | 3104                   | 78.86     | 78.72       | 63.78    | 0.62       | 0.47 | 0.30        | 0.30| 0.955                    |
| 4       | 2930                   | 81.26     | 72.11       | 63.82    | 0.65       | 0.47 | 0.30        | 0.30| 0.941                    |
| 3       | 1850                   | 75.36     | 49.10       | 66.43    | 0.68       | 0.47 | 0.31        | 0.30| 0.926                    |
| 2       | 571                    | 73.56     | 15.52       | 72.92    | 0.66       | 0.48 | 0.32        | 0.34| 0.868                    |
| 1       | 33                     | 59.16     | 1.12        | 77.45    | 0.72       | 0.47 | 0.33        | 0.29| 0.687                    |

<sup>a</sup> Number of valid and unique molecules from 5,000 samples;  
<sup>b</sup> From Equation (7).

TABLE XI: Different numbers of qubits in the quantum circuit of QuMolGAN-HR. The QED, Solute, and SA scores in this table are calculated from the valid molecules.

| # qubits<sup>a</sup> | # molecules<sup>b</sup> | validity↑ | uniqueness↑ | novelty↑ | diversity↑ | QED↑ | Solubility↑ | SA↑ | KL Score (S)<sup>c</sup>↑ |
|----------------------|--------------------------|-----------|-------------|----------|------------|------|-------------|-----|--------------------------|
| 8                    | 515                      | 73.94     | 13.93       | 60.05    | 0.62       | 0.46 | 0.31        | 0.27| 0.785                    |
| 4                    | 534                      | 70.02     | 15.25       | 71.04    | 0.70       | 0.47 | 0.32        | 0.29| 0.894                    |
| 3                    | 301                      | 64.74     | 9.30        | 66.23    | 0.58       | 0.50 | 0.34        | 0.38| 0.782                    |
| 2                    | 194                      | 63.68     | 6.09        | 66.02    | 0.60       | 0.49 | 0.36        | 0.45| 0.570                    |

<sup>a</sup> Number of qubits of quantum circuit;  
<sup>b</sup> Number of valid and unique molecules from 5,000 samples;  
<sup>c</sup> From Equation (7).

QuMolGAN-HR has the best KL Score as z<sub>dim</sub>=4.

Appendix C: Number of Parameterized Layers of Variational Quantum Circuit

Variational quantum circuits (VQCs) can be considered machine learning models with remarkable expressive power for a variety of data-driven tasks, such as supervised learning and generative modeling. VQCs consists of three ingredients: a fixed initial state, e.g., the zero state, a quantum circuit, parameterized by a set of learnable parameters, and the measurement as shown in Figure 2. A quantum circuit is composed of an initialization layer and parameterized layer(s). The initialization layer brings randomness to the VQC, and the parameters of parameterized layer(s) can be learned through back-propagation. One challenge in implementing variational quantum algorithms is to choose an effective circuit that well represents the solution space while maintaining a low circuit depth and the number of parameters. Increasing the number of parameterized layers of VQC has the potential to improve the expressive power of VQC and its performance. However, it may face the problem of Barren plateaus and longer computational time. In this experiment, we would like to find the optimal number of parameterized layers that can generate the most unique and valid molecules for QuMolGAN-HR. We have examined one to five parameterized layers, and the results are shown in Table XI. In this table, QuMolGAN-HR achieves the best KL Score when the VQC has three parameterized layers.
TABLE XII: Different parameterized layers of QuMolGAN-HR. The QED, Solute, and SA scores in this table are calculated from the valid molecules.

| # layers | # molecules | validity | uniqueness | novelty | diversity | QED | Solubility | SA | KL Score (S) |
|----------|-------------|----------|------------|---------|-----------|-----|-------------|----|-------------|
| 5        | 375         | 70.10    | 10.70      | 70.44   | 0.63      | 0.47| 0.32       | 0.31| 0.738       |
| 4        | 377         | 52.04    | 14.49      | 64.64   | 0.57      | 0.46| 0.32       | 0.27| 0.701       |
| 3        | 443         | 72.40    | 12.24      | 69.20   | 0.68      | 0.50| 0.36       | 0.32| 0.784       |
| 2        | 435         | 76.06    | 11.44      | 71.00   | 0.61      | 0.45| 0.33       | 0.28| 0.765       |
| 1        | 165         | 75.98    | 4.34       | 67.81   | 0.66      | 0.44| 0.32       | 0.26| 0.702       |

* Number of parameterized layers;

b Number of valid and unique molecules from 5,000 samples;

c From Equation (7).