JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

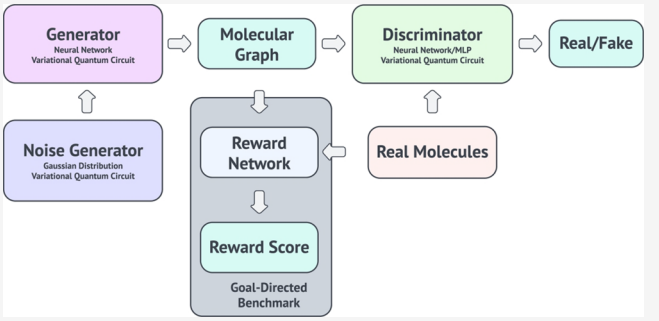
Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry

Po-Yu Kao, Ya-Chu Yang, Wei-Yin Chiang, Jen-Yueh Hsiao, Yudong Cao, Alex Aliper, Feng Ren, Alán Aspuru-Guzik, Alex Zhavoronkov,* Min-Hsiu Hsieh,* and Yen-Chu Lin*

Cite This: *J. Chem. Inf. Model.* 2023, 63, 3307−3318 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: De novo drug design with desired biological activities is crucial for developing novel therapeutics for patients. The drug development process is time- and resource-consuming, and it has a low probability of success. Recent advances in machine learning and deep learning technology have reduced the time and cost of the discovery process and therefore, improved pharmaceutical research and development. In this paper, we explore the combination of two rapidly developing fields with lead candidate discovery in the drug development process. First, artificial intelligence has already been demonstrated to successfully accelerate conventional drug design approaches. Second, quantum computing has demonstrated promising potential in different applications, such as quantum chemistry, combinatorial optimizations, and machine learning. This article explores hybrid quantum-classical generative adversarial networks (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 a 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. 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 both generated molecule properties and the achieved KL divergence. However, the hybrid quantum-classical GANs still face challenges in generating unique and valid molecules compared to their classical counterparts.



INTRODUCTION

The drug development process includes discovery and development, preclinical research, clinical research, Food and Drug Administration (FDA) review, and FDA postmarket safe monitoring. The entire process is time- and resource-consuming and has a low probability of success, with ∼4% of preclinical drugs being 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 deep learning technology have improved and reduced the cost of pharmaceutical R&D.[4−9] For example, our team[10] discovered 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 a 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 druglike compounds.[15]

Received: April 12, 2023
Published: May 12, 2023

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

Aside from LBDD and FBDD, SBDD is based on the properties of the active site of a biological target.

Artificial intelligence (AI) has made a breakthrough in the recent de novo molecule design.[16−18] We identify four well-known generative machine learning algorithms[19] in the field: evolutionary algorithms (EA), recurrent neural networks (RNNs) (such as gated recurrent unit (GRU) and long short-term memory (LSTM)), autoencoders (such as adversarial autoencoder (AAE) and variational autoencoder (VAE)),[20] and generative adversarial network (GAN). GAN[21] has become a popular network architecture for generating highly realistic data,[22] and it has shown remarkable results for generating data that mimics a data distribution in different tasks.[23−26] GAN consists of a generator and a discriminator defined by an artificial neural network (ANN). The parameters of a GAN can be learned by back propagation. 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. A GAN is trained until the discriminator cannot distinguish the generated data from the real data.

GANs are one of the most successful generative models in the drug discovery field, and several different GAN architectures have been proposed in the past decades in de novo drug discovery.[27−29] Zhavoronkov et al.[10] proposed a deep generative model called generative tensorial reinforcement learning (GENTRL) for de novo small-molecule generation. The GENTRL generates novel drugs with better synthetic feasibility and biological activity. Guimaraes et al.[30] proposed an objective-reinforced generative adversarial network (ORGAN) that combines the GAN and reinforcement learning (RL) algorithm. ORGAN is built on SeqGAN[31] 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.[32] 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.[33] This method allows LatentGAN to focus on optimizing the sampling without worrying about SMILES syntax issues. Cao and Kipf introduced MolGAN[34] 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. These generative models have the potential to be improved with quantum machine learning algorithms.

The applications of quantum computing can be found in different fields, such as solving routing problems,[35,36] stock price forecasting,[37,38] multitask classification,[39] the two-player zero-sum game,[40] high-resolution handwritten digits generation,[41] the discovery of molecular properties,[42−49] tautomeric state prediction,[50] and drug design.[51,52] 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. Several studies have demonstrated that variational quantum circuit (VQC)[53] performs the advantages in expression power,[54] learnability,[55] and robustness.[56] It indicates that VQC can greatly boost the solution to the problem, e.g., drug discovery, which is hard to tackle by the classical neural network.

Quantum generative adversarial network (QuGAN)[57] 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 architectural complexities could easily bypass their classical counterparts.[52,58] Dallaire-Demers et al. provided the first feasible implementation of QuGAN using quantum circuits in a simulator.[59] Analogous proposals of QuGAN for continuous functions have been proposed during the same period.[60] Later on, QuGANs demonstrated its first successful training on the MNIST dataset on a physical quantum device.[58] Li et al.[52] pushed the research to the 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[61] 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.

COMPUTATIONAL RESULTS AND DISCUSSION

In this study, we would like to computationally explore the potential benefits of our algorithms as compared to classical algorithms in small molecule discovery. We endeavored to find the best hyper-parameters for the base MolGAN. We first examined different complexities of generators and observed that MolGAN-HR (high reduction) has the best performance, compared to other generator complexities (see Table S1 in the Supporting Information). We then examined different input noise dimensions of generators and determined that the number of unique and valid molecules saturated at the input dimension was equal to 4 (z_dim = 4) (see Table S2 in the Supporting Information). In addition, we also tested different numbers of parametrized layers in the variational quantum circuit (VQC) and observed that MolGAN-HR with three

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

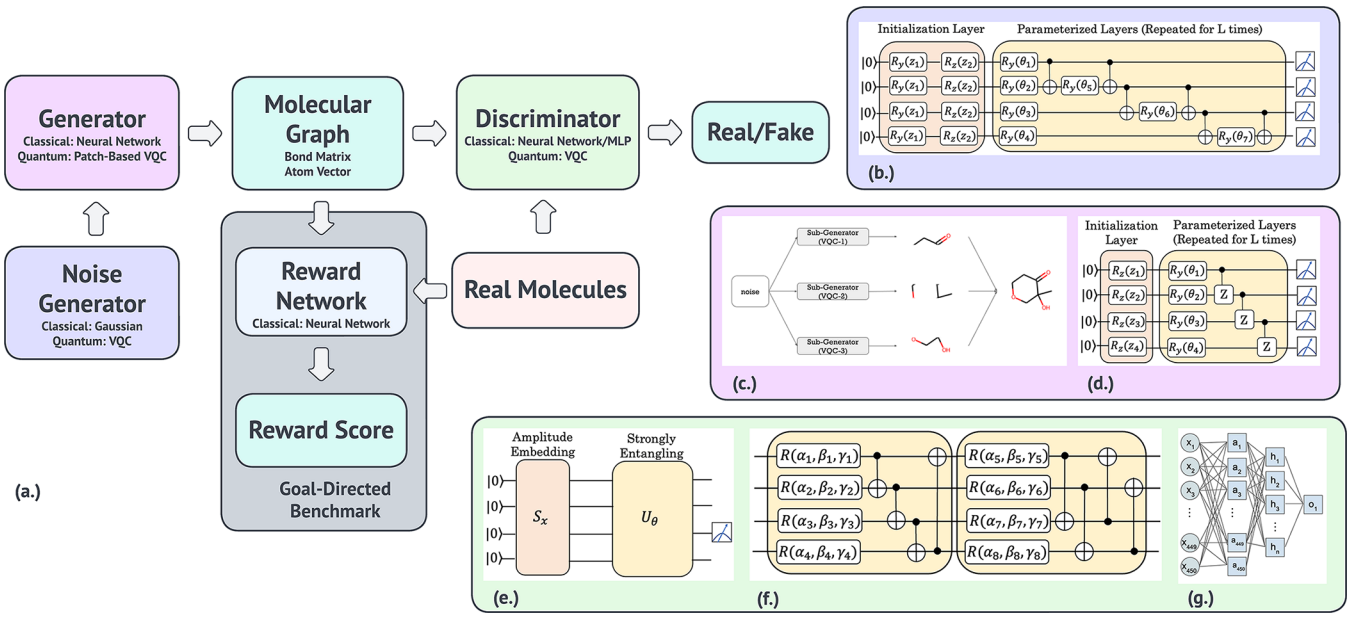


Figure 1. Overall pipeline. (a) The overall pipeline of MolGAN with different combinations of classical/quantum components. The reward neural network branch is enabled in the goal-directed benchmark. The classical noise generator samples from the Gaussian distribution, and the quantum one uses the variational quantum circuit (VQC). The classical generator is built by neural networks, and the quantum one uses the patch-based VQC to generate the molecular graph. The molecular graph is represented by a bond matrix and atom vector. The classical discriminator is built by a graph-based neural network or multilayer perceptron (MLP), and the VQC is used in the quantum one. (b) The example of VQC in the noise generator. (c) The patch method uses multiple VQCs as subgenerators. Each subgenerator 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. (d) The example of VQC in the quantum generator. (e) The VQC of the quantum discriminator consists of the amplitude embedding circuit (S_x), the strong entanglement layers (U_θ), and the measurement. (f) The VQC of strongly entanglement layers contains multiple CNOT gates and parametrized rotational gates (R). (g) MLP-based discriminator architecture in MolGAN-CC.

Table 1. Comparing Generated Molecules of QuMolGAN and MolGAN in Drug Properties[a]

	z_dim = 2			z_dim = 3			z_dim = 4		
	QuMolGAN	MolGAN	p-value	QuMolGAN	MolGAN	p-value	QuMolGAN	MolGAN	p-value
number of molecules[b]	363	657	–	414	2163	–	511	3085	–
QED ↑	***0.489***	***0.475***	***<0.01***	***0.489***	***0.465***	***<0.01***	***0.473***	***0.465***	***<0.05***
solubility ↑	***0.343***	***0.324***	***<0.05***	***0.370***	***0.305***	***<0.01***	***0.317***	***0.298***	***<0.01***
SA ↑	***0.367***	***0.336***	***<0.05***	***0.310***	***0.307***	***<0.05***	0.308	0.296	0.246
KL score (S)[c] ↑	0.653	0.824	–	0.797	0.913	–	0.846	0.957	–

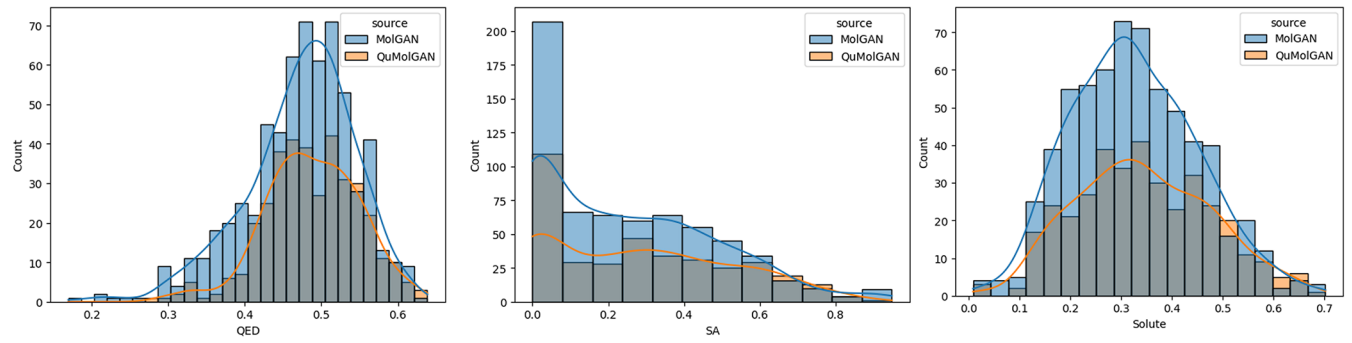
[a]Bold numbers highlight the better scores in QuMolGAN, compared to the corresponding MolGAN. Note that the QED, Solute, and SA scores in this table are calculated from the valid and ***unique*** molecules. [b]Number of valid and unique molecules from 5000 samples. [c]From eq 2.

parametrized layers of VQC as a noise generator has the best performance, compared to other numbers of parametrized layers (Table S3 in the Supporting Information). Therefore, in the following experiments, MolGAN-HR is used as the base model, and VQC with three parametrized layers is used as the base quantum circuit.

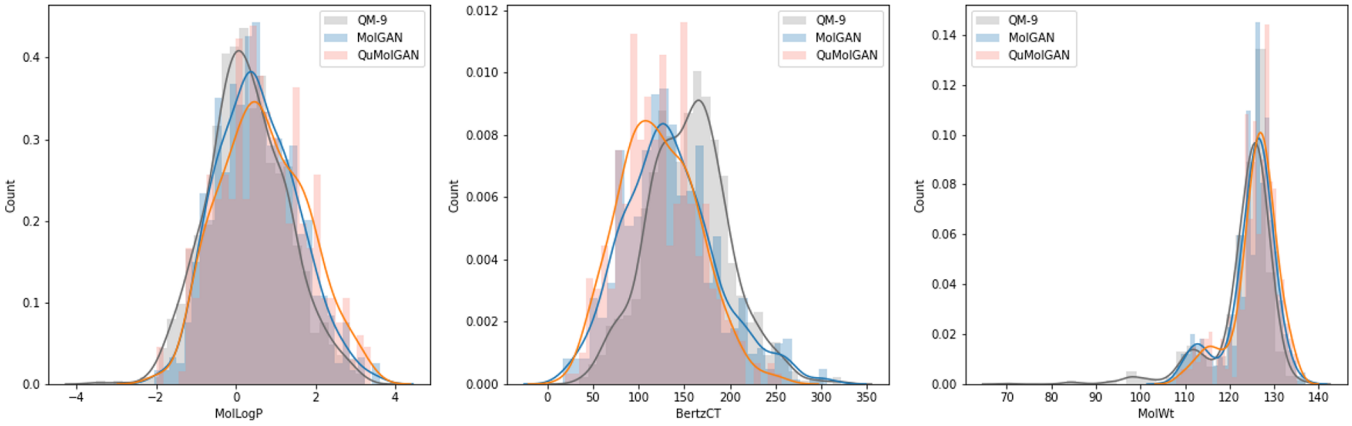
In the first experiment, we substitute the noise generator of MolGAN with a VQC and discover the quantum advantage in the generated molecules with better drug properties. In the second experiment, we replace the generator of MolGAN with a VQC and show the potential of generating small molecules using a VQC. In the third experiment, we supplant the discriminator of MolGAN with a VQC and demonstrate the quantum advantage. Figure 1a illustrates different combinations of the classical/quantum noise/generator/discriminator, and their corresponding model name is shown in Table S4 in the Supporting Information. MolGAN uses a classical noise generator, a generator, and a graph-based discriminator. QuMolGAN uses a *quantum noise generator*, a classical generator, and a classical graph-based discriminator. MolGAN-QC uses a classical noise generator, a *quantum generator*, and a classical graph-based discriminator. MolGAN-CQ uses a classical noise generator, a classical generator, and a *quantum discriminator*. MolGAN-CC uses a classical noise generator, a classical generator, and a *classical MLP-based discriminator*. All experiments are implemented by using Pennylane[63] and PyTorch.[64]

Quantum Noise Generator. In the first experiment, we would like to compare the performance of QuMolGAN and classical MolGAN. We have examined different qubits for VQC and found that its performance saturates at 4 qubits (Table S5 in the Supporting Information). In addition, we have also tested different numbers of parametrized layers in the VQC and observed that QuMolGAN with three parametrized layers of VQC as a noise generator has the best performance, compared to other numbers of parametrized layers (Table S3). We use the same hyperparameters to train the QuMolGAN and MolGAN,[34] except for the learning rate. The learning rate

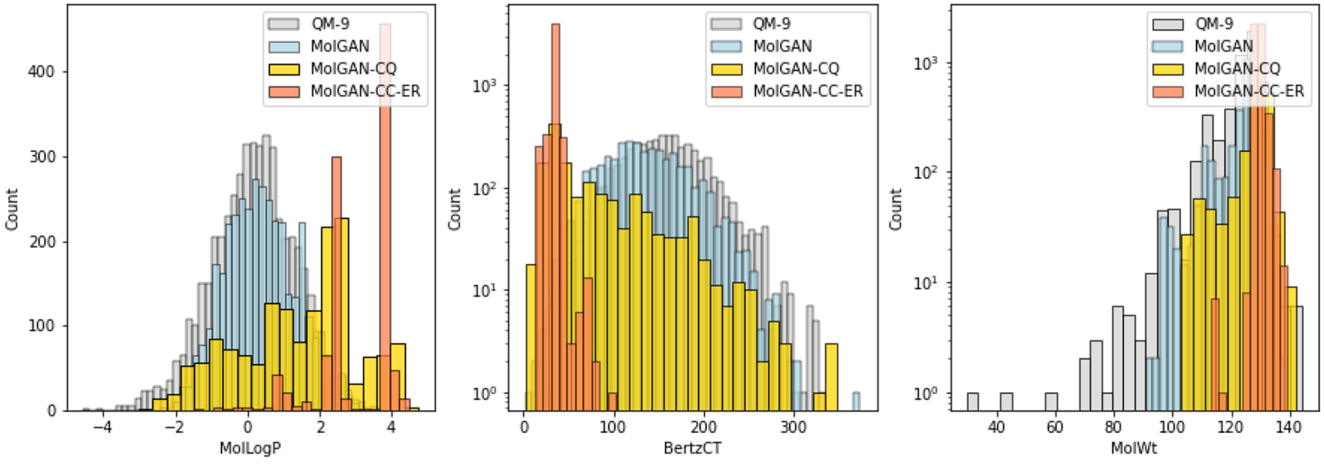
https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318



(a.)



(b.)



(c.)

Figure 2. Property distributions of molecules. (a) Drug properties distributions (left to right: QED, SA, and Solute) from valid and unique MolGAN-generated (in blue) and QuMolGAN-generated (in orange) molecules. (b) KL-divergence distributions (left to right: MolLogP, BertzCT, and MolWt) of valid and unique MolGAN-generated (blue), QuMolGAN-generated (orange), and QM9 (gray) molecules. (c) KL-divergence distributions (from left to right: MolLogP, BertzCT, and MolWt) of MolGAN-CC-ER-generated (orange), MolGAN-CQ-generated (yellow), MolGAN-generated (blue), and QM9 (gray) molecules.

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[65] is used to train the models. QuMolGAN with three parametrized layers is used in this experiment. We have examined the input dimension of the generator at 2, 3, and 4, and the results are shown in Table 1. The performance of MolGAN and QuMolGAN saturates at z_dim = 4. 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 MolGAN,

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

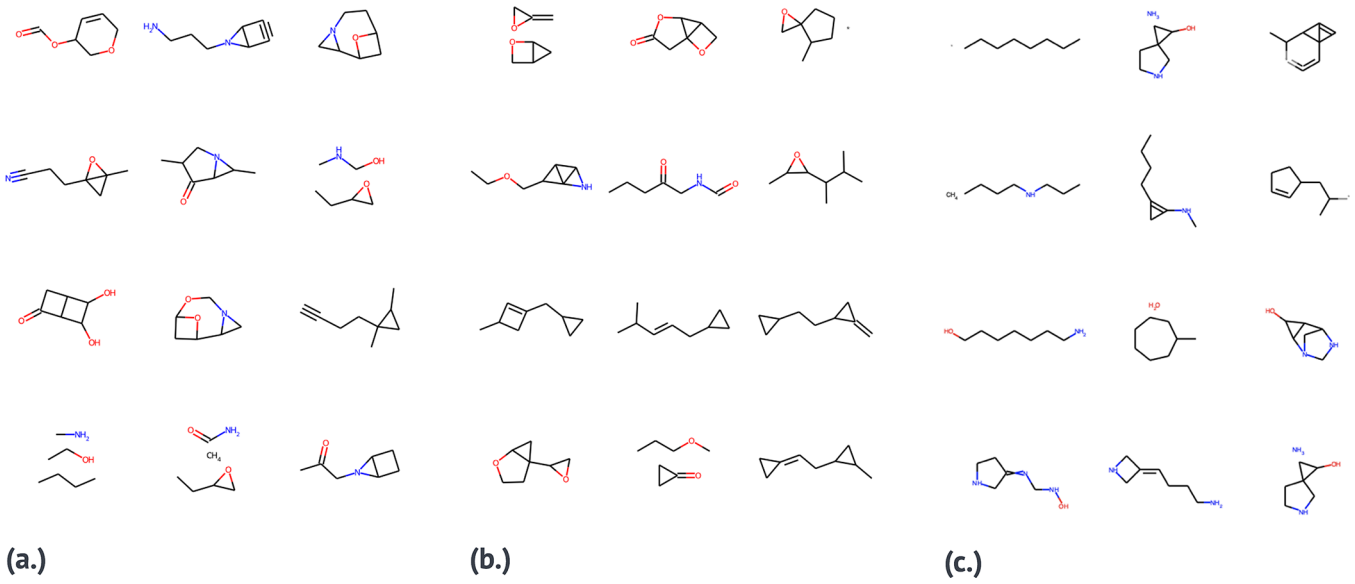


Figure 3. Example molecules. (a) Example molecules of MolGAN with z_dim = 2. (b) Example molecules of QuMolGAN with z_dim = 2. (c) Example molecules of MolGAN-CQ.

Table 2. Performance Comparison between MolGAN and QuMolGAN with (α = {0.5, 0.01}) and without (α = 1.0) Reinforcement Learning Loss in the Goal-Directed Benchmark[a]

	MolGAN			QuMolGAN		
	α = 1.0	α = 0.5	α = 0.01	α = 1.0	α = 0.5	α = 0.01
number of molecules[b]	**2890**	2700	696	**534**	309	116
validity ↑	**80.40**	78.48	68.76	70.02	**70.32**	42.94
uniqueness ↑	**71.89**	68.81	20.24	**15.25**	8.78	5.40
QED ↑	0.47	0.48	**0.52**	0.47	0.49	**0.57**
Solute ↑	0.31	0.31	**0.45**	0.32	0.30	**0.44**
SA ↑	0.30	0.31	**0.65**	0.29	0.28	**0.76**
KL Score (S)[c] ↑	**0.95**	0.94	0.58	**0.92**	0.82	0.31

[a]Note 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 α, and the underlined numbers indicate the best scores across different types of models. [b]Number of valid and unique molecules from 5000 samples. [c]From eq 2.

particularly when the input noise dimension is small (z_dim = 2 and z_dim = 3). The drug properties distributions of MolGAN-generated and QuMolGAN-generated molecules are shown in Figure 2a for z_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 <0.4. For the KL-divergence task, the MolLogP, BertzCT, and MolWt distributions of generated molecules are also shown in Figure 2b for z_dim = 2. The classical MolGAN (black) tends to generate molecules with similar distributions to the training set (gray), which results in a better KL-divergence score. In the end, we randomly sample 12 valid and unique molecules from both MolGAN and QuMolGAN for z_dim = 2, and the example molecules are shown in Figures 3a and 3b, respectively. QuMolGAN can generate training-set-like molecules with better drug properties.

Goal-Directed Benchmark. As mentioned in ref 66, 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 enable the reward network branch of the original schema as shown in the gray block in Figure 1a. At this time, the generator is trained using a linear combination of the WGAN[65] loss and the RL[67] (reinforcement learning) loss:

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

where $\alpha \in [0, 1]$ is a hyperparameter that controls the tradeoff between WGAN loss and RL loss, and ω are the inputs to networks. Here, we set α = {1.0, 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 (α = {0.5, 0.01}) and without (α = 1.0) RL loss using the same hyperparameters for 150 epochs, and the results are shown in Table 2. The input noise dimension to the generator is 4 (z_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 counterparts in different weights of RL loss. However, QuMolGAN can achieve a higher goal (from 0.47 to 0.57 in QED and from 0.29 to 0.76 in SA), compared to

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

Table 3. Performance of MolGAN with the Quantum Generator[a]

# epoch	number of molecules[b]	validity ↑	uniqueness ↑	novelty ↑	diversity ↑	QED ↑	solubility ↑	SA ↑	KL Score (S)[c] ↑
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

[a]The QED, Solute, and SA scores in this table are calculated from the valid molecules. [b]Number of valid and unique molecules from 2048 samples from Gaussian distribution. [c]From eq 2.

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.

Quantum Generator. In the second experiment, we try to benchmark 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 ref 58 for the small molecule generation. The performance of MolGAN with the quantum generator (MolGAN-QC) is reported in Table 3.

Although the integration works smoothly, the training processing is time-consuming and resource-consuming. The average training time per step takes ∼39 s in the Amazon EC2 C6a Metal Instance, which results in ∼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 10 epochs of training. Random-picked and cherry-picked examples of generated molecules are shown in Figure S1. Most generated molecules are similar to the randomly sampled molecules in Figure S1. However, we demonstrate that the quantum generator has the potential to generate small molecules.

Quantum Discriminator. In the third experiment, in addition to substituting the generator architecture, the architecture with a quantum discriminator combined with the classical generator described in MolGAN has been examined. Our goal is to determine 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 perceptron (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 S4. In this experiment, the noises of the generator are always sampled from the Gaussian distribution.

In the following subparagraphs, 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.

MolGAN-CQ Architecture and Training Details. As illustrated in Figure 1a, the MolGAN-CQ consists of three components: a classical noise generator, a classical generator, and a 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 a size of 450. After that, the resulting vector will be the input of the Q-Discriminator, composed of an amplitude embedding layer followed by three strongly entangling layers as described in Figure 1e. For the training details, at first, we followed the best hyperparameters 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 1×10^{-3} to 1×10^{-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.

Comparison with Classical MolGAN. In this section, we compare MolGAN-CQ with MolGAN. We trained MolGAN according to the best hyperparameter 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 4 demonstrates that MolGAN-CQ can generate valid and druglike molecules. In addition, MolGAN-CQ can generate molecules with better drug properties, particularly in Solute and SA. However, compared to classical MolGAN, MolGAN-CQ does not have an advantage in KL divergence with training data probability. Figure 3c shows some molecules generated from MolGAN-CQ.

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 a multiple-layer perceptron (MLP), as shown in Figure 1g.

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

Table 4. Performance Comparison between MolGAN and MolGAN-CQ[a]

	MolGAN	MolGAN-CQ
# of molecules[b]	**2693**	730
Validity ↑	**76**	31.34
Uniqueness ↑	**70.87**	46.59
QED ↑	0.47	**0.48**
Solute ↑	0.31	**0.44**
SA ↑	0.31	**0.66**
KL Score (S)[c]↑	**0.94**	0.75

[a]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. [b]Number of valid and unique molecules from 5000 samples. [c]From eq 2

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 consists of a classical noise generator, a classical generator, and the classical MLP-based discriminator, as shown in Figure 1a. 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 evaluation. However, the input vector is already a size of 450, it is not possible to decrease the parameter size to 50 in the classical MLP. Therefore, we have tried MolGAN-CC with three different parameter sizes of discriminator, as shown in Table S6 in the Supporting Information. MolGAN-CC-ER (extremely reduction) has three layers with sizes (450, 50, 1) in the discriminator, and the total parameter size is 22 000. MolGAN-CC-HR (highly reduction) has three layers with size (450, 100, 1) in the discriminator, and the total parameter size is 45 000, MolGAN-CC-NR (no reduction) has four layers with sizes (450, 150, 50, 1) in the discriminator, and the total parameter size is 82 000.

Table 5 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 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 2c. As we can see from Figure 2c, molecular properties from MolGAN are closer 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 22 000 parameters.

Table 5. Performance Comparison between MolGAN-CQ and MolGAN-CC with Three Different Sizes of the MLP-Based Discriminator[a]

	MolGAN-CQ	MolGAN-CC-ER	MolGAN-CC-HR	MolGAN-CC-NR
number of parameters	50	22K	45K	82K
number of molecules[b]	730	104	1919	**2284**
Validity ↑	31.34	**99.78**	44.1	54.7
Uniqueness ↑	46.59	2.08	**87.03**	83.51
QED ↑	0.48	**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)[c]↑	0.75	0.28	**0.84**	0.81

[a]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. [b]Number of valid and unique molecules from 5000 samples. [c]From eq 2

CONCLUSIONS

In this paper, we have explored the quantum advantage in small-molecule drug discovery by substituting each part of MolGAN[34] with a VQC step by step and comparing 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, including QED, SA, and LogP, particularly when the input dimension to the generator is small, e.g., z_dim = 2 or z_dim = 3. However, QuMolGAN has difficulty generating as many unique molecules compared to classical MolGAN, which results in 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 substitute the classical generator with a VQC with the patch method.[58] We 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, we replace the classical discriminator with a VQC and compare its performance with the classical counterpart. This hybrid model MolGAN-CQ outperforms the classical counterpart, in terms of generated molecule properties and the KL score. We also demonstrated that the hybrid model could generate valid molecules with only tens of learnable parameters in a quantum discriminator. The proposed hybrid model has the potential to be integrated into the Insilico Medicine Chemistry42[62] platform (see Figure S2 in the Supporting Information).

Finally, expressive power illustrates the capability of a model to capture the distribution of the target features. Increasing model complexity can normally enhance this capability. In our work, we demonstrate that a quantum circuit consisting of only 50 learnable parameters is able to generate a distribution similar to what classical generates but with 20K learnable parameters. Therefore, the quantum advantage in expressive power is observed.

MATERIALS AND METHODS

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

Evaluation Method and Metrics. To evaluate the performance of different generative models, we first generate 5000 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

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

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 refs 68 and 69 and three drug properties, i.e., quantitative estimation of *drug-likeness (QED)*, *solubility*, and *synthesizability (SA)*, is 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 that 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 (*number of molecules*) from 5000 noise samples. QED[70] 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*)[71] of the molecule that is the degree of a molecule being hydrophilic. SA[72] 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.[73]

The Kullback−Leibler (KL) divergences[74] 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[71]), MolWt (molecular weight), TPSA (molecular polar surface area), NumHAcceptors (number of hydrogen acceptors), NumHDonors (number of hydrogen donors), NumRotatableBonds (number of rotatable bonds), 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 $D_{\mathrm{KL},i}$ 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)[66] used in this paper is computed by

$$S = \frac{1}{9}\sum_{i=1}^{9} \exp(-D_{\mathrm{KL},i}) \tag{2}$$

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

DATASET

All experiments of this work use the QM9 (Quantum Machines 9)[75] dataset. The QM9 dataset is curated from the GDB-17 chemical database,[76] which has ∼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 S3 in the Supporting Information shows some molecules from the QM9 dataset, and the average QED, solubility, and SA of those 133 171 molecules are 0.461, 0.289, and 0.327, respectively.

METHODOLOGY

Variational Quantum Circuits (VQCs). 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 single-qubit rotations (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.

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3}$$

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

$$\mathrm{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{4}$$

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 θ in radians around the x-axis, and it is defined as

$$R_x(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{5}$$

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

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{6}$$

The R_z gate is a single-qubit rotation through an angle θ in radians around the z-axis, and it is defined as

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{7}$$

The R gate is a single-qubit rotation through arbitrary angles α, β, and γ (in radians), and it can be decomposed into R_y and R_z gates. It is defined as

$$\begin{aligned} R(\alpha, \beta, \gamma) &= R_z(\gamma)R_y(\beta)R_z(\alpha) \\ &= \begin{pmatrix} e^{-i(\alpha+\gamma)/2}\cos(\beta/2) & -e^{-i(\alpha-\gamma)/2}\sin(\beta/2) \\ e^{-i(\alpha-\gamma)/2}\sin(\beta/2) & e^{i(\alpha+\gamma)/2}\cos(\beta/2) \end{pmatrix} \end{aligned} \tag{8}$$

A variational quantum circuit (VQC) shown in Figures 1b and 1d 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 parametrized layers, which could be repeated for L times could have CNOT gates, CZ gates, and parametrized rotational gates whose parameters (rotation angle) can be learned through the back-propagation. The measurement takes the expected value of each qubit.

VQC of Noise Generator. In this work, MolGAN[34] 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[34] is an implicit and likelihood-free generative model for small molecular graph generation. In contrast to the

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

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,[21] 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 1a. 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 1b 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[34] paper.

VQC of Quantum Generator. We implement the patch method[58] in the quantum generator of MolGAN (MolGAN-QC). This method uses multiple VQCs as subgenerators, and each subgenerator 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 1c. The subgenerator shares the same ansatz architecture as shown in Figure 1d. Each ansatz circuit consists of the preparation of the initialization state, a single layer of a 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 parametrized layers (could be repeated for L times) have CZ gates and one type of parametrized rotational gates, R_y whose parameters (θ_k) can be learned through back-propagation.

VQC of Quantum Discriminator. The quantum discriminator takes the molecular graph as input and determines if this molecular graph is fake (from the generator) or real (from the data distribution). The VQC of the quantum discriminator consists of the amplitude encoding layer[77] (S_x), the strongly entangling layers (U_θ) inspired by ref 78 and the measurement, as shown in Figure 1e. Amplitude encoding is used to encode the atom vector and bond matrix. The strongly entangling layers[78] have multiple CNOT gates and parametrized rotational gates $R(\alpha,\beta,\gamma)$, as shown in Figure 1f. In each layer, each qubit starts with parametrized rotational gates $R(\alpha_i,\beta_i,\gamma_i)$ followed by a CNOT gate. The parametrized angles $\alpha_i,\beta_i,\gamma_i$ can be learned through back-propagation. The measurement takes the expected 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 nine qubits to encode the molecular graph and three layers of strongly entanglement layers.

■ ASSOCIATED CONTENT

Data Availability Statement

The data acquisition code and source codes associated with this study are publicly available at https://github.com/pykao/QuantumMolGAN-PyTorch. The discriminator branch contains a classical/quantum noise generator, a classical generator, and a classical/quantum discriminator. This branch can be used for the quantum noise generator and the quantum discriminator. The generator branch contains a classical/quantum noise generator, a classical/quantum generator, and a classical discriminator. This branch can be used for the quantum generator.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.3c00562.

Different complexities of generators in MolGAN; different input noise dimensions of the generator in MolGAN-HR; different parametrized layers of QuMolGAN-HR; all the combinations of the classical/quantum noise/generator/discriminator and their corresponding model name; different numbers of qubits in the quantum circuit of QuMolGAN-HR; the details of MolGAN-CC models with the varied size of discriminators; example molecules from the quantum generator; integration of proposed hybrid generative models with Insilico Medicine Chemistry42[62] platform; example molecules of QM9 (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Alex Zhavoronkov − *Insilico Medicine Hong Kong Ltd., Hong Kong SAR 999077, China*; Email: alex@insilico.com

Min-Hsiu Hsieh − *Hon Hai (Foxconn) Research Institute, Taipei 114699, Taiwan*; Email: min-hsiu.hsieh@foxconn.com

Yen-Chu Lin − *Insilico Medicine Taiwan Ltd., Taipei 110208, Taiwan; Department of Pharmacy, National Yang Ming Chiao Tung University, Taipei 112304, Taiwan*; Email: jimmy.lin@isilico.com

Authors

Po-Yu Kao − *Insilico Medicine Taiwan Ltd., Taipei 110208, Taiwan*; orcid.org/0000-0002-9439-8819

Ya-Chu Yang − *Insilico Medicine Taiwan Ltd., Taipei 110208, Taiwan*

Wei-Yin Chiang − *Hon Hai (Foxconn) Research Institute, Taipei 114699, Taiwan*

Jen-Yueh Hsiao − *Hon Hai (Foxconn) Research Institute, Taipei 114699, Taiwan*

Yudong Cao − *Zapata Computing, Inc., Boston, Massachusetts 02110, United States*

Alex Aliper − *Insilico Medicine AI Limited, Abu Dhabi 145748, UAE*

Feng Ren − *Insilico Medicine Shanghai Ltd., Shanghai 201203, China*

Alán Aspuru-Guzik − *Department of Chemistry, University of Toronto, Toronto, ON M5S 3H6, Canada; Department of Computer Science, University of Toronto, Toronto, ON M5S 2E4, Canada; Vector Institute for Artificial Intelligence, Toronto, ON M5S 1M1, Canada; Lebovic Fellow, Canadian Institute for Advanced Research, Toronto, ON M5S 1M1, Canada*; orcid.org/0000-0002-8277-4434

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.3c00562

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Hingorani, A. D.; Kuan, V.; Finan, C.; Kruger, F. A.; Gaulton, A.; Chopade, S.; Sofat, R.; MacAllister, R. J.; Overington, J. P.;

Hemingway, H.; Denaxas, S.; Prieto, D.; Casas, J. P. Improving the odds of drug development success through human genomics: modelling study. *Sci. Rep.* **2019**, *9*, 1−25.
(2) Abreu, J. L. Ivermectin for the Prevention of COVID-19 So...WHO is Telling the Truth. *Revista Daena (Int. J. Good Conscience)* **2020**, *15* (2), 1−30.
(3) Wouters, O. J.; McKee, M.; Luyten, J. Estimated research and development investment needed to bring a new medicine to market, 2009−2018. *Jama* **2020**, *323*, 844−853.
(4) Aliper, A.; Plis, S.; Artemov, A.; Ulloa, A.; Mamoshina, P.; Zhavoronkov, A. Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data. *Mol. Pharmaceutics* **2016**, *13*, 2524−2530.
(5) Kadurin, A.; Aliper, A.; Kazennov, A.; Mamoshina, P.; Vanhaelen, Q.; Khrabrov, K.; Zhavoronkov, A. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* **2017**, *8*, 10883.
(6) Zhavoronkov, A. Artificial intelligence for drug discovery, biomarker development, and generation of novel chemistry. *Mol. Pharmaceutics* **2018**, *15*, 4311.
(7) Carracedo-Reboredo, P.; Liñares-Blanco, J.; Rodríguez-Fernández, N.; Cedrón, F.; Novoa, F. J.; Carballal, A.; Maojo, V.; Pazos, A.; Fernandez-Lozano, C. A review on machine learning approaches and trends in drug discovery. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 4538−4558.
(8) Kao, P.-Y.; Kao, S.-M.; Huang, N.-L.; Lin, Y.-C. Toward Drug-Target Interaction Prediction via Ensemble Modeling and Transfer Learning. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2021; pp 2384−2391.
(9) Kolluri, S.; Lin, J.; Liu, R.; Zhang, Y.; Zhang, W. Machine Learning and Artificial Intelligence in Pharmaceutical Research and Development: a Review. *AAPS J.* **2022**, *24*, 1−10.
(10) Zhavoronkov, A.; Ivanenkov, Y. A.; Aliper, A.; Veselov, M. S.; Aladinskiy, V. A.; Aladinskaya, A. V.; Terentiev, V. A.; Polykovskiy, D. A.; Kuznetsov, M. D.; Asadulaev, A.; Volkov, Y.; Zholus, A.; Shayakhmetov, R. R.; Zhebrak, A.; Minaeva, L. I.; Zagribelnyy, B. A.; Lee, L. H.; Soll, R.; Madge, D.; Xing, L.; Guo, T.; Aspuru-Guzik, A. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **2019**, *37*, 1038−1040.
(11) Chan, H. S.; Shan, H.; Dahoun, T.; Vogel, H.; Yuan, S. Advancing drug discovery via artificial intelligence. *Trends Pharmacol. Sci.* **2019**, *40*, 592−604.
(12) Schneider, G.; Fechner, U. Computer-based de novo design of drug-like molecules. *Nat. Rev. Drug Discovery* **2005**, *4*, 649−663.
(13) Fischer, T.; Gazzola, S.; Riedl, R. Approaching target selectivity by de novo drug design. *Expert Opin. Drug Discovery* **2019**, *14*, 791−803.
(14) Mouchlis, V. D.; Afantitis, A.; Serra, A.; Fratello, M.; Papadiamantis, A. G.; Aidinis, V.; Lynch, I.; Greco, D.; Melagraki, G. Advances in de novo drug design: from conventional to machine learning methods. *Int. J. Mol. Sci.* **2021**, *22*, 1676.
(15) Speck-Planche, A. Recent advances in fragment-based computational drug design: tackling simultaneous targets/biological effects. *Future Med. Chem.* **2018**, *10*, 2021−2024.
(16) Mamoshina, P.; Ojomoko, L.; Yanovich, Y.; Ostrovski, A.; Botezatu, A.; Prikhodko, P.; Izumchenko, E.; Aliper, A.; Romantsov, K.; Zhebrak, A.; Ogu, I. O.; Zhavoronkov, A. Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare. *Oncotarget* **2018**, *9*, 5665.
(17) Zhavoronkov, A.; Zagribelnyy, B.; Zhebrak, A.; Aladinskiy, V.; Terentiev, V.; Vanhaelen, Q.; Bezrukov, D. S.; Polykovskiy, D.; Shayakhmetov, R.; Filimonov, A.; Filimonov, A.; Bishop, M.; McCloskey, S.; Lejia, E.; Bright, D.; Funakawa, K.; Lin, Y.-C.; Huang, S.-H.; Liao, H.-J.; Aliper, A.; Ivanenkov, Y. Potential non-covalent SARS-CoV-2 3C-like protease inhibitors designed using generative deep learning approaches and reviewed by human medicinal chemist in virtual reality. *ChemRxiv*, **2020**.
(18) Paul, D.; Sanap, G.; Shenoy, S.; Kalyane, D.; Kalia, K.; Tekade, R. K. Artificial intelligence in drug discovery and development. *Drug Discovery Today* **2021**, *26*, 80.
(19) Martinelli, D. Generative machine learning for de novo drug discovery: A systematic review. *Comput. Biol. Med.* **2022**, *145*, 105403.
(20) Gao, K.; Nguyen, D. D.; Tu, M.; Wei, G.-W. Generative Network Complex for the Automated Generation of Drug-like Molecules. *J. Chem. Inf. Model.* **2020**, *60*, 5682−5698.
(21) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*; Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; Weinberger, K. Q., Eds.; 2014.
(22) Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint*, 2017, arXiv:1710.10196.
(23) Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017; pp 1125−1134.
(24) Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017; pp 2223−2232.
(25) Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019; pp 4401−4410.
(26) Jangid, D. K.; Brodnik, N. R.; Khan, A.; Goebel, M. G.; Echlin, M. P.; Pollock, T. M.; Daly, S. H.; Manjunath, B. 3D Grain Shape Generation in Polycrystals Using Generative Adversarial Networks. *Integr. Mater. Manuf. Innov.* **2022**, *11*, 71−84.
(27) Kadurin, A.; Nikolenko, S.; Khrabrov, K.; Aliper, A.; Zhavoronkov, A. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol. Pharmaceutics* **2017**, *14*, 3098−3104.
(28) Vanhaelen, Q.; Lin, Y.-C.; Zhavoronkov, A. The advent of generative chemistry. *ACS Med. Chem. Lett.* **2020**, *11*, 1496−1505.
(29) Xu, M.; Cheng, J.; Liu, Y.; Huang, W. DeepGAN: Generating Molecule for Drug Discovery Based on Generative Adversarial Network. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, 2021; pp 1−6.
(30) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv preprint*, arXiv:1705.10843, 2017.
(31) Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017; pp 2852−2858.
(32) Prykhodko, O.; Johansson, S. V.; Kotsias, P.-C.; Arús-Pous, J.; Bjerrum, E. J.; Engkvist, O.; Chen, H. A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminf.* **2019**, *11*, 1−13.
(33) Kotsias, P.-C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E. J. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nat. Mach. Intell.* **2020**, *2*, 254−265.
(34) De Cao, N.; Kipf, T. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint* arXiv:1805.11973 2018,.
(35) Neukart, F.; Compostella, G.; Seidel, C.; Von Dollen, D.; Yarkoni, S.; Parney, B. Traffic flow optimization using a quantum annealer. *Front. ICT* **2017**, *4*, 29.
(36) Harwood, S.; Gambella, C.; Trenev, D.; Simonetto, A.; Bernal Neira, D.; Greenberg, D. Formulating and solving routing problems on quantum computers. *IEEE Trans. Quantum Eng.* **2021**, *2*, 1−17.
(37) Orus, R.; Mugel, S.; Lizaso, E. Quantum computing for finance: Overview and prospects. *Rev. Phys.* **2019**, *4*, 100028.

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

(38) Liu, G.; Ma, W. A quantum artificial neural network for stock closing price prediction. *Inf. Sci. (N.Y.)* **2022**, *598*, 75−85.
(39) Du, Y.; Yang, Y.; Tao, D.; Hsieh, M.-H. Demystify Problem-Dependent Power of Quantum Neural Networks on Multi-Class Classification. *arXiv preprint*, arXiv:2301.01597, 2022.
(40) Yin, X.-F.; Du, Y.; Fei, Y.-Y.; Zhang, R.; Liu, L.-Z.; Mao, Y.; Liu, T.; Hsieh, M.-H.; Li, L.; Liu, N.-L.; Tao, D.; Chen, Y.-A.; Pan, J.-W. Efficient Bipartite Entanglement Detection Scheme with a Quantum Adversarial Solver. *Phys. Rev. Lett.* **2022**, *128*, 110501.
(41) Rudolph, M. S.; Toussaint, N. B.; Katabarwa, A.; Johri, S.; Peropadre, B.; Perdomo-Ortiz, A. Generation of high-resolution handwritten digits with an ion-trap quantum computer. *Phys. Rev. X* **2022**, *12*, 031010.
(42) Aspuru-Guzik, A.; Dutoi, A. D.; Love, P. J.; Head-Gordon, M. Simulated quantum computation of molecular energies. *Science* **2005**, *309*, 1704−1707.
(43) Lanyon, B. P.; Whitfield, J. D.; Gillett, G. G.; Goggin, M. E.; Almeida, M. P.; Kassal, I.; Biamonte, J. D.; Mohseni, M.; Powell, B. J.; Barbieri, M.; Aspuru-Guzik, A.; White, A. G. Towards quantum chemistry on a quantum computer. *Nat. Chem.* **2010**, *2*, 106−111.
(44) Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.-H.; Zhou, X.-Q.; Love, P. J.; Aspuru-Guzik, A.; O'brien, J. L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 1−7.
(45) O'Malley, P. J. J.; Babbush, R.; Kivlichan, I. D.; Romero, J.; McClean, J. R.; Barends, R.; Kelly, J.; Roushan, P.; Tranter, A.; Ding, N.; Campbell, B.; Chen, Y.; Chen, Z.; Chiaro, B.; Dunsworth, A.; Fowler, A. G.; Jeffrey, E.; Lucero, E.; Megrant, A.; Mutus, J. Y.; Neeley, M.; Neill, C.; Quintana, C.; Sank, D.; Vainsencher, A.; Wenner, J.; White, T. C.; Coveney, P. V.; Love, P. J.; Neven, H.; Aspuru-Guzik, A.; Martinis, J. M. Scalable quantum simulation of molecular energies. *Phys. Rev. X* **2016**, *6*, 031007.
(46) Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J. M.; Gambetta, J. M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242−246.
(47) Cao, Y.; Romero, J.; Olson, J. P.; Degroote, M.; Johnson, P. D.; Kieferová, M.; Kivlichan, I. D.; Menke, T.; Peropadre, B.; Sawaya, N. P. D.; Sim, S.; Veis, L.; Aspuru-Guzik, A. Quantum Chemistry in the Age of Quantum Computing. *Chem. Rev.* **2019**, *119*, 10856−10915.
(48) Quantum, G. A.; Collaborators*†; Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J. C.; Barends, R.; Boixo, S.; Broughton, M.; et al. Hartree-Fock on a superconducting qubit quantum computer. *Science* **2020**, *369*, 1084−1089.
(49) Gao, Q.; Nakamura, H.; Gujarati, T. P.; Jones, G. O.; Rice, J. E.; Wood, S. P.; Pistoia, M.; Garcia, J. M.; Yamamoto, N. Computational investigations of the lithium superoxide dimer rearrangement on noisy quantum devices. *J. Phys. Chem. A* **2021**, *125*, 1827−1836.
(50) Shee, Y.; Yeh, T.-L.; Hsiao, J.-Y.; Yang, A.; Lin, Y.-C.; Hsieh, M.-H. Quantum Simulation of Preferred Tautomeric State Prediction. *arXiv preprint*, arXiv:2210.02977, 2022.
(51) Cao, Y.; Romero, J.; Aspuru-Guzik, A. Potential of quantum computing for drug discovery. *IBM J. Res. Dev.* **2018**, *62*, 1−20.
(52) Li, J.; Topaloglu, R. O.; Ghosh, S. Quantum generative models for small molecule drug discovery. *IEEE Trans. Quantum Eng.* **2021**, *2*, 1−8.
(53) Bharti, K.; Cervera-Lierta, A.; Kyaw, T. H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J. S.; Menke, T.; Mok, W.-K.; Sim, S.; Kwek, L.-C.; Aspuru-Guzik, A. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004.
(54) Du, Y.; Hsieh, M.-H.; Liu, T.; Tao, D. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.* **2020**, *2*, 033125.
(55) Du, Y.; Hsieh, M.-H.; Liu, T.; You, S.; Tao, D. Learnability of quantum neural networks. *PRX Quantum* **2021**, *2*, 040337.
(56) Du, Y.; Hsieh, M.-H.; Liu, T.; Tao, D.; Liu, N. Quantum noise protects quantum classifiers against adversaries. *Phys. Rev. Res.* **2021**, *3*, 023153.
(57) Lloyd, S.; Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **2018**, *121*, 040502.
(58) Huang, H.-L.; Du, Y.; Gong, M.; Zhao, Y.; Wu, Y.; Wang, C.; Li, S.; Liang, F.; Lin, J.; Xu, Y.; Yang, R.; Liu, T.; Hsieh, M.-H.; Deng, H.; Rong, H.; Peng, C.-Z.; Lu, C.-Y.; Chen, Y.-A.; Tao, D.; Zhu, X.; Pan, J.-W. Experimental quantum generative adversarial networks for image generation. *Phys. Rev. Appl.* **2021**, *16*, 024051.
(59) Dallaire-Demers, P.-L.; Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **2018**, *98*, 012324.
(60) Romero, J.; Aspuru-Guzik, A. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *Adv. Quantum Technol.* **2021**, *4*, 2000003.
(61) Li, J. Quantum GAN with Hybrid Generator. https://github.com/jundeli/quantum-gan, accessed Aug. 2, 2021.
(62) Ivanenkov, Y. A.; Polykovskiy, D.; Bezrukov, D.; Zagribelnyy, B.; Aladinskiy, V.; Kamya, P.; Aliper, A.; Ren, F.; Zhavoronkov, A. Chemistry42: An AI-Driven Platform for Molecular Design and Optimization. *J. Chem. Inf. Model.* **2023**, *63*, 695−701.
(63) Bergholm, V.; Izaac, J.; Schuld, M.; Gogolin, C.; Ahmed, S.; Ajith, V.; Alam, M. S.; Alonso-Linaje, G.; AkashNarayanan, B.; Asadi, A.; Arrazola, J. M.; Azad, U.; Banning, S.; Blank, C.; Bromley, T. R.; Cordier, B. A.; Ceroni, J.; Delgado, A.; Di Matteo, O.; Dusko, A.; Garg, T.; Guala, D.; Hayes, A.; Hill, R.; Ijaz, A.; Isacsson, T.; Ittah, D.; Jahangiri, S.; Jain, P.; Jiang, E.; Khandelwal, A.; Kottmann, K.; Lang, R. A.; Lee, C.; Loke, T.; Lowe, A.; McKiernan, K.; Meyer, J. J.; Montañez-Barrera, J. A.; Moyard, R.; Niu, Z.; O'Riordan, L. J.; Oud, S.; Panigrahi, A.; Park, C.-Y.; Polatajko, D.; Quesada, N.; Roberts, C.; Sá, N.; Schoch, I.; Shi, B.; Shu, S.; Sim, S.; Singh, A.; Strandberg, I.; Soni, J.; Száva, A.; Thabet, S.; Vargas-Hernández, R. A.; Vincent, T.; Vitucci, N.; Weber, M.; Wierichs, D.; Wiersema, R.; Willmann, M.; Wong, V.; Zhang, S.; Killoran, N. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint*, arXiv:1811.04968, 2018.
(64) Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. Presented at the *NIPS 2017 Autodiff Workshop*, **2017**.
(65) Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017; pp 214223.
(66) Brown, N.; Fiscato, M.; Segler, M. H.; Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **2019**, *59*, 1096−1108.
(67) Kaelbling, L. P.; Littman, M. L.; Moore, A. W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237−285.
(68) Samanta, B.; De, A.; Ganguly, N.; Gomez-Rodriguez, M. Designing random graph models using variational autoencoders with applications to chemical design. *arXiv preprint*, arXiv:1802.05283, 2018.
(69) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A. Molecular sets (MOSES): a benchmarking platform for molecular generation models. *Front. Pharmacol.* **2020**, *11*, 565644.
(70) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **2012**, *4*, 90−98.
(71) Wildman, S. A.; Crippen, G. M. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Model.* **1999**, *39*, 868−873.
(72) Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminf.* **2009**, *1*, 1−11.
(73) Landrum, G.; Tosco, P.; Kelley, B.; Sriniker, R.; Gedeck; Vianello, R.; Schneider, N.; Kawashima, E.; Dalke, A.; N, D.; Cosgrove, D.; Cole, B.; Swain, M.; Turk, S.; Savelyev, A.; Jones, G.; Vaucher, A.; Wójcikowski, M.; Take, I.; Probst, D.; Ujihara, K.; Scalfani, V. F.; godin, G.; Pahl, A.; Berenger, F.; Varjo, J. L.; Strets, J.

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318

P.; Doliath Gavid rdkit/rdkit: 2022_03_1 (Q1 2022) Release. 2022, DOI: 10.5281/zenodo.6388425.
(74) Kullback, S.; Leibler, R. A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79−86.
(75) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 1−7.
(76) Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864−2875.
(77) Schuld, M.; Petruccione, F. *Supervised Learning with Quantum Computers*, Vol. *17*; Springer, 2018.
(78) Schuld, M.; Bocharov, A.; Svore, K. M.; Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **2020**, *101*, 032308.

https://doi.org/10.1021/acs.jcim.3c00562
J. Chem. Inf. Model. 2023, 63, 3307−3318