Hybrid Quantum Cycle Generative Adversarial Network for Small Molecule Generation

MATVEI ANOSHIN[ID]**, ASEL SAGINGALIEVA**[ID]**, CHRISTOPHER MANSELL**[ID]**,
DMITRY ZHIGANOV**[ID]**, VISHAL SHETE**[ID]**, MARKUS PFLITSCH,
AND ALEXEY MELNIKOV**[ID]
Terra Quantum AG, 9000 St. Gallen, Switzerland

Corresponding author: Alexey Melnikov (e-mail: ame@terraquantum.swiss).

ABSTRACT The drug design process currently requires considerable time and resources to develop each new compound that enters the market. This work develops an application of hybrid quantum generative models based on the integration of parameterized quantum circuits into known molecular generative adversarial networks and proposes quantum cycle architectures that improve model performance and stability during training. Through extensive experimentation on benchmark drug design datasets, quantum machine 9 (QM9) and PubChemQC 9 (PC9), the introduced models are shown to outperform the previously achieved scores. Most prominently, the new scores indicate an increase of up to 30% in the quantitative estimation of druglikeness. The new hybrid quantum machine learning algorithms, as well as the achieved scores of pharmacokinetic properties, contribute to the development of fast and accurate drug discovery processes.

INDEX TERMS Drug design, hybrid quantum neural network (HQNN), quantum generative adversarial network (GAN), quantum machine learning (QML), variational quantum circuit (VQC).

I. INTRODUCTION

In the current pharmaceutical landscape, the drug design process is a prolonged and costly endeavor. It typically spans up to 15 years [1] from target identification to clinical application, incurring expenses of approximately $1 billion for each new drug. Machine learning has shown successful uses through the different stages of drug development, from the search for specific protein inhibitors [2] to the evaluation of pharmacokinetic properties and adverse effects.

Generative adversarial networks (GANs) [3] have gained prominence in molecular design. Their architecture is adept at generating a vast array of potential drug candidates from extensive molecular spaces, thereby facilitating more efficient preliminary screenings. GAN models, especially when compared to recurrent neural networks [4] and variational autoencoders [5], have demonstrated superiority in generating SMILES [6] representations of compounds. A novel quantum approach introduced in [7] used the deep variational autoencoder model trained to construct molecules as SMILES strings. The advancements in molecule representations led to the use of graph representations of compounds. The use of graphs instead of SMILES, which are invariant to the permutation of atom orders, has allowed GANs, particularly

molecular generative adversarial network (MolGAN) [8], to become the state-of-the-art approach in generative chemistry.

Quantum-enhanced GANs, with their inherent probabilistic nature, offer a moderate advantage over their classical counterparts by encompassing a broader and more diverse chemical space [9]. However, in the current Noisy Intermediate-Scale Quantum era, the feasibility of purely quantum algorithms is limited. Here, hybrid algorithms may find a reasonable equilibrium between the high expressivity of modern quantum simulators and the stability of classical approaches.

The study of hybrid quantum neural networks (HQNNs) represents a convergence of classical deep learning architectures with quantum machine learning (QML) algorithms [10], [11], [12], [13], [14], specifically through parameterized quantum circuits [15]. This hybrid approach harnesses the strengths of classical and quantum computing, introducing a system capable of efficiently processing large datasets compared to classical deep learning architectures alone [16], [17]. HQNNs have exhibited promising applications across various industrial domains, including health care [18], [19], [20], chemistry [21], [22], energy industry [23], routing [24], and aerospace [25], as well as in image classification [26],

[27]. While HQNNs have demonstrated efficacy in these fields, further research is essential to explore their potential in drug design.

This article introduces the hybrid quantum cycle MolGAN (HQ-Cycle-MolGAN) for generating graph representations of small molecules. By incorporating a cycle component into the hybrid quantum MolGAN (HQ-MolGAN), where both the generator and the cycle component are represented using an HQNN [10], [28], we have been able to stabilize the training process for molecular samples and improve key metrics. This includes increases of up to 30% in the quantitative estimate of druglikeness (QED) score [29], a composite metric that evaluates a molecule's overall druglikeness based on its chemical structure. The QED score is instrumental in assessing the potential of a compound to qualify as an effective drug, providing a quantitative measure that can guide early drug discovery efforts. In addition, we have observed improvements in the synthesis accessibility (SA) score [30] and the logP score [31]. The SA score quantifies the complexity of synthesizing a given molecular compound, offering insights into the practicality of its production at scale. A lower SA score indicates easier synthesis, which is favorable for drug development. The LogP score measures a compound's solubility and permeability, indicating its balance between hydrophilicity (water solubility) and lipophilicity (fat solubility). This balance is crucial for a drug's absorption, distribution, metabolism, and excretion properties, impacting its effectiveness and safety. Overall, the proposed hybrid models show an advantage over their nearest competitors among quantum models in terms of pharmacokinetic properties, including QED score, SA score, and logP score.

This work contributes insights into the potential of QML for small molecule generation, emphasizing the benefits of hybrid quantum–classical approaches in drug design. The results underscore the significance of employing quantum-enhanced models to achieve improved performance across essential molecular optimization metrics.

II. PRELIMINARIES

In this study, we introduce several models for small molecule generation. First, we present a refined classical MolGAN with a halved parameter count, drawing inspiration from the state-of-the-art classical MolGAN architecture [8]. Second, in Section II-B, we present an HQ-MolGAN, fusing classical and quantum computing approaches for enhanced capabilities. Notably, we propose the novel classical cycle MolGAN, incorporating a multiparameter reward function based on reinforcement learning principles, inspired by the state-of-the-art cycle MolGAN [32]. In addition, in Section II-C, we introduce the innovative HQ-Cycle-MolGAN. Through rigorous experimentation on the quantum machine 9 (QM9) and PubChemQC 9 (PC9) datasets, described in Section II-A, our results demonstrate that models trained on PC9 exhibit higher LogP scores than their QM9 counterparts. Furthermore, the hybrid quantum models showcase better performance, achieving the highest QED, SA, and LogP

metrics scores. Remarkably, our hybrid quantum model outperforms a similar quantum generative adversarial network hybrid generator moderately reduced (QGAN-HG MR) hybrid model from [9] and quantum molecular generative adversarial network (QuMolGAN) from [33], emphasizing the efficacy of our proposed approaches in small molecule generation. We summarize our conclusions and outline future research directions in Section IV.

A. DATASET

This study employed two datasets for model training: QM9 and PC9. The QM9 dataset [34], [35], a well-established benchmark in small molecule drug design since 2012, comprises approximately 134 000 neutral molecules, each with no more than nine atoms (C, O, N, F) apart from hydrogen. Its comprehensive and diverse chemical space makes it particularly relevant for this field.

The second dataset, PC9, is a subset of the extensive PubChem database, containing around 99 000 molecules [36]. A notable distinction between QM9 and PC9 is that the latter includes not only neutral molecules but also those with a multiplicity greater than one. While PC9 was initially proposed as a replacement for the QM9 dataset, its practical usage alongside QM9 has demonstrated benefits in generating a more diverse set of molecular structures.

Fig. 1 shows that the mean QED and LogP scores for molecules in the PC9 dataset are higher than those in QM9, while the mean score for SA is lower. Intuitively, this suggests that models trained on the PC9 dataset might be inclined to generate samples with higher values in these two key metrics than similar models trained on QM9. However, as detailed in Section III, this is not always the case.

It is important to note that during training, normalized values of logP, natural product likeness (NP), and SA scores were evaluated and optimized.

B. HYBRID QUANTUM MOLGAN

In this section, we introduce the HQ-MolGAN, which is based on the classical MolGAN architecture [8]. As depicted in the shaded green rectangle of Fig. 2(a), the architecture of HQ-MolGAN is comprised of three primary components: the generator (G), the discriminator (D), and the reward component (R). This model operates on the principles of the Wasserstein GAN [37], wherein the generator endeavors to synthesize molecular graph representations indistinguishable from authentic ones, thereby "deceiving" the discriminator. The training regimen of HQ-MolGAN encapsulates a min–max optimization game, wherein the generator and the discriminator engage in a continuous adaptive process to refine the generative quality of molecular representations

$$\min_{G} \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{y} \sim P_{\text{data}}} [D(\mathbf{y})] - \mathbb{E}_{\mathbf{z} \sim P_z} [D(G(\mathbf{z}))]$$

$$- \lambda \mathbb{E}_{\hat{\mathbf{y}} \sim P_{\hat{\mathbf{y}}}} \underbrace{\left[\left(\left\| \nabla_{\hat{\mathbf{y}}} D(\hat{\mathbf{y}}) \right\|_2 - 1 \right)^2 \right]}_{\text{gradient penalty}}.$$
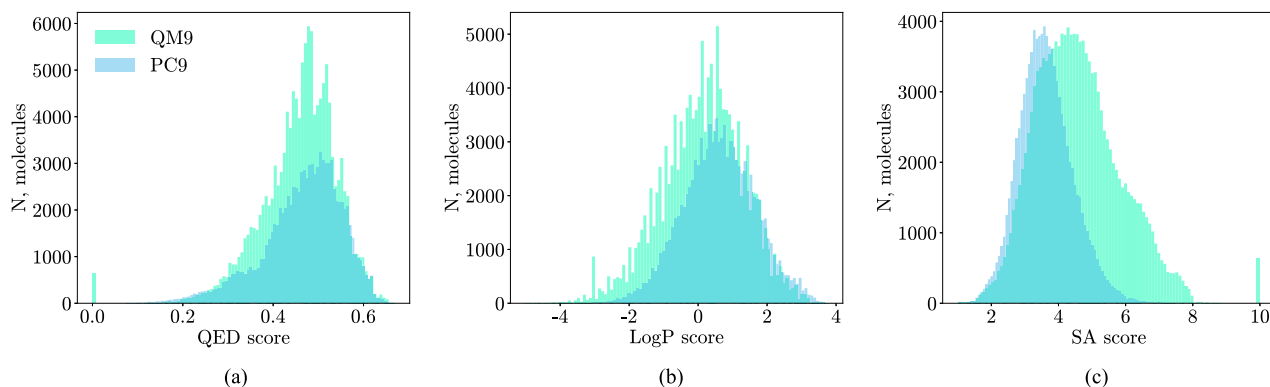
FIGURE 1. Histograms of the distribution of values of QED, SA, and LogP scores in QM9 and PC9 datasets. The mean QED and LogP scores for molecules in the PC9 dataset are greater than those in QM9, while the mean score for SA is lower.

The reward component in our HQ-MolGAN architecture functions as a sophisticated reinforcement learning objective, tasked with evaluating the generator's output based on several chemical property metrics. This evaluation extends beyond the conventional metrics of QED, LogP, and SA scores. It incorporates a comprehensive assessment of "validity," quantified as the ratio of valid molecular samples to the total number of generated samples. Furthermore, it considers "novelty," defined by the proportion of generated valid samples not present in the training dataset. In addition, the reward component assesses "diversity," a measure of the variance in the chemical structures of the generated molecules, and the NP score, as delineated in [38]. These multifaceted evaluation criteria enable a more nuanced and thorough assessment of the generator's performance, aligning the generated molecules more closely with desired chemical characteristics.

In the architecture of HQ-MolGAN, a pivotal role is played by the variational quantum circuit (VQC), which is integrated as the initial layer in MolGAN's generator. The VQC operates by encoding a noise vector into N qubits. Subsequent to the application of rotation and entanglement gates, the VQC outputs a probability distribution vector, denoted as $[p(0), \ldots, p(2^N - 1)]$, where each element represents the probability of a corresponding quantum state. This vector, with a dimensionality of 2^N, undergoes a truncation process where only the first $2^{N-N_{\text{ancilla}}}$ elements are retained. The truncated vector is then fed into the classical fully connected layers, which constitute the remaining component of HQ-MolGAN's generator.

In our experimental analysis, two distinct configurations of the VQC were evaluated: the vanilla variational repetitive quantum (VVRQ) layer [39] and the exponential fourier quantum (EFQ) layer [40]. These configurations offer different approaches to quantum state transformation, thus providing a comparative understanding of their efficacy in the context of molecule generation.

The operational mechanism of the VVRQ layer within our HQ-MolGAN framework [see Fig. 2(a)] involves encoding the noise vector directly onto the qubits in a single step using angle embedding [41] [green rectangles in Fig. 2(a)]. Following this initialization, the VVRQ layer implements several variational layers, which consist of a sequence of Rot (rotation) gates [blue rectangles in Fig. 2(a)]

$$\text{Rot}(\theta_1, \theta_2, \theta_3) = Ry(\theta_1) \cdot Rz(\theta_2) \cdot Ry(\theta_3)$$

$$Ry(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$Rz(\theta) = \begin{pmatrix} \exp\left(-i\frac{\theta}{2}\right) & 0 \\ 0 & \exp\left(i\frac{\theta}{2}\right) \end{pmatrix}$$

and controlled NOT (CNOT) gates

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Originally, this approach was proposed in [42] for image generation, but it can be applied to any generative task. The CNOT gates are applied between sequentially adjacent qubits, i.e., between qubit i and qubit $i + 1$, thereby facilitating quantum entanglement and information propagation across the qubit array.

This process is iteratively repeated for three variational layers, ensuring a thorough and complex manipulation of the quantum state. The final step in the VVRQ process involves measuring the probability distribution of the quantum states of the qubits. These measurements yield a probability vector that encapsulates the resultant quantum state post-entanglement and rotation, reflecting the encoded information from the initial noise vector.

In the EFQ layer, the data encoding process is distinctly characterized by a dual-phase approach. Initially, the input
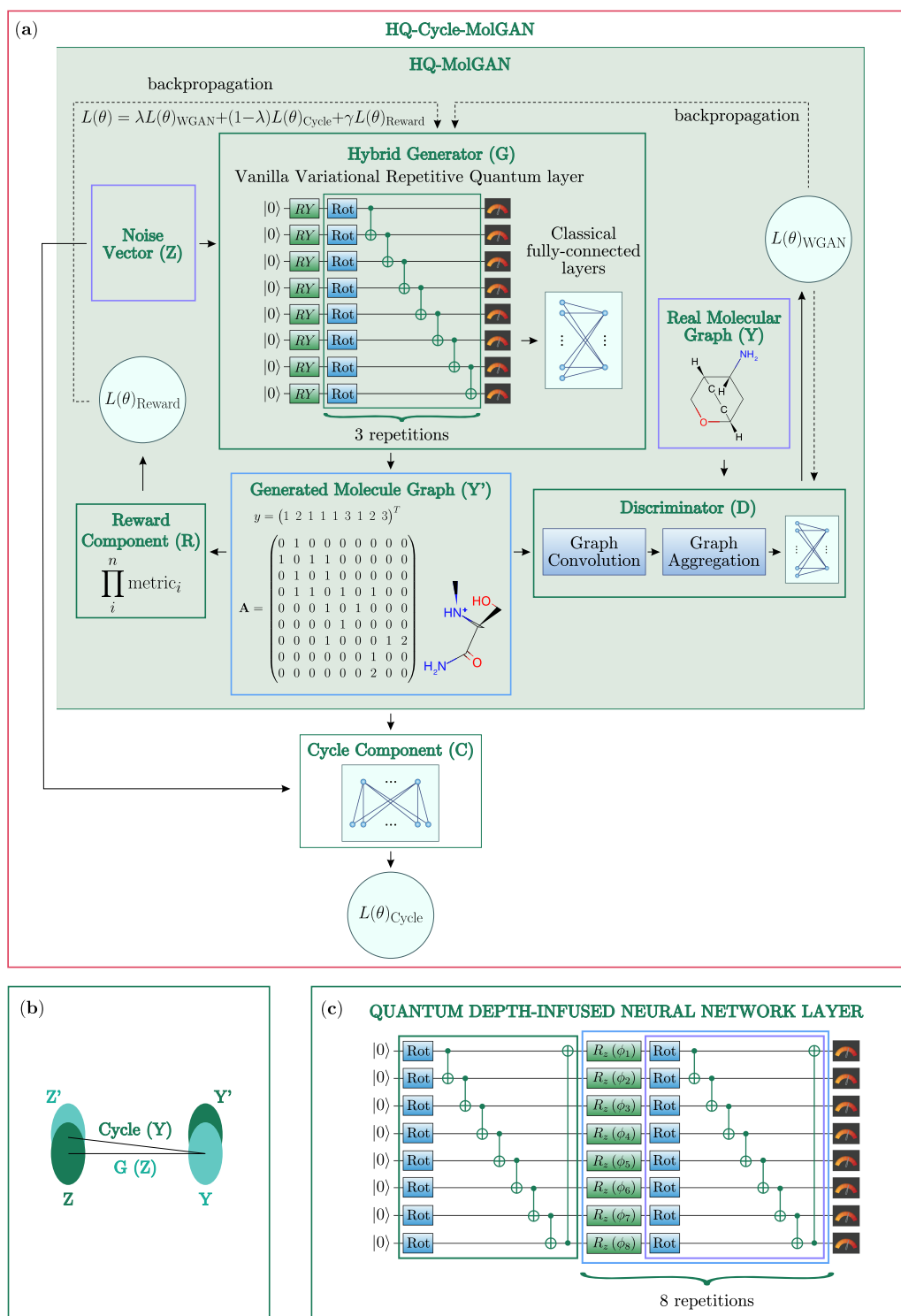
FIGURE 2. (a) Structure of HQ-Cycle-MolGAN: generator (G), discriminator (D), cycle component (C). The part highlighted in green is the same as HQ-MolGAN. (b) Illustration of the work of the cycle components. Suppose that Z is a space of normally distributed noise vectors, and Y is a chemical space of datasets. The generator maps Z to some chemical space Y', and after the cycle component restores vector $G(Z)$ back to noise. The accuracy of these restorations is then optimized. (c) Quantum depth-infused neural network layer used as the HQ-Cycle component.

data are encoded onto the qubits using angle embedding. This is followed by several variational layers and a second encoding phase, in which the amplitude of the rotational gates is systematically increased to double its initial value.

This dual-encoding scheme, particularly with the amplified rotational amplitude in the second phase, is designed to enhance the expressive power of the quantum circuit [43]. By manipulating the amplitude of rotations in this manner,

the EFQ layer could potentially induce a more diverse and complex quantum state space.

C. HYBRID QUANTUM CYCLE MOLGAN

The Cycle-MolGAN model introduces an innovative "cycle component" (C) [see Fig. 2(b)] to the established MolGAN architecture. This component is ingeniously designed to reverse the molecule generation process. Specifically, it converts the generated molecular samples back into their originating noise vectors and assesses the accuracy of this reverse conversion. This approach, as proposed in [32] and [44], has been identified as particularly advantageous in the realm of molecular optimization tasks. It contributes significantly to the stability of the training performance and is instrumental in suppressing the training of the nonisomorphic generator compounds within the hybrid-MolGAN framework, especially pertinent in generating small molecules.

Practically implemented, the cycle component takes the form of a multilayer perceptron (MLP) model. This model effectively combines the adjacency matrix and the feature matrix of each generated molecular sample into a singular unified tensor. Following this integration, the cycle component proceeds to "mirror" the layers of the generator, albeit in reverse order. This mirroring process is a critical step as it compresses the expanded dimensions of the combined tensor, specifically **batch_size** × 405 from the adjacency matrix and **batch_size** × 45 from the feature matrix, down to a more manageable size of **batch_size** × 8. This reduction is pivotal for effectively re-encoding the complex molecular information back into the concise form of noise vectors.

In the development of the hybrid cycle component within the Cycle-MolGAN framework, we adhere to the classical design but with a crucial modification in the final layer. This layer is replaced by a quantum depth-infused neural network layer, as described in [45]. This quantum depth-infused layer undertakes the task of encoding a vector of size **batch_size** × 64 into eight qubits through a series of eight repetitive encoding layers [blue rectangles in Fig. 2(c)].

To optimize the performance of the generator within this architecture, we employ a combined loss function, articulated as follows:

$$L(\theta) = \lambda \cdot L(\theta)_{\text{WGAN}} + (1 - \lambda) \cdot L(\theta)_{\text{Cycle}} + \gamma L(\theta)_{\text{Reward}}$$

$\gamma, \ \lambda \in [0, 1]$.

This loss function integrates the Wasserstein GAN loss ($L(\theta)_{\text{WGAN}}$), the cycle loss ($L(\theta)_{\text{Cycle}}$), and the reward loss ($L(\theta)_{\text{Reward}}$). The coefficients γ and λ regulate the relative influence of each component in the overall loss calculation. In our experiments, we set λ to 0.5, thereby assigning equal importance to both cycles of transformation (from noise vector Z to generated sample Y', and back from Y' to Z), as illustrated in Fig. 2(b).

III. RESULTS

The models in this study were developed in Python3, utilizing the PyTorch framework [46] and PennyLane [47] for the quantum computation. The simulations described further

were performed on classical simulators emulating quantum hardware. To evaluate the chemical properties of the synthesized compounds, we utilized the RDKit library. Our computational experiments leveraged GPU hardware, specifically the Tesla V100 and RTX 3060 GPUs, to facilitate efficient processing.

For testing the potential performance on quantum hardware, a Qiskit [48] implementation of the VQC was used on the simulator of the IBM Brisbane device [49], [50].

For the classical MolGAN models, the generator's architecture was scaled down by reducing the number of parameters in each layer by half, resulting in a total of 157 570 parameters in the generator. The classical models, including both the standard MolGAN and Cycle-MolGAN, underwent a training regime of 200 000 iterations with a batch size of ten samples. In contrast, the hybrid quantum models were subjected to a shorter training duration of 50 000 iterations. The validation set size was limited, containing either 100 samples when training on a single dataset or 250 samples in cases where multiple datasets were employed. In both the EFQ and VVRQ models, the number of ancilla qubits is equal to 2.

The experimental investigation was conducted in four distinct stages.

1) *First stage (see Section III-A):* This phase focused on evaluating the performance differences between the VVRQ and EFQ layers when integrated into the hybrid generator in HQ-MolGAN.
2) *Second stage (see Section III-B):* The objective was to assess the impact of the classical cycle component on the performance of MolGAN and HQ-MolGAN.
3) *Third stage (see Section III-C):* This stage involved an analysis of the hybrid-cycle component, including a comparative study against the classical cycle component.
4) *Fourth stage (see Section III-D):* This phase included a setup and analysis of HQ-MolGAN's performance test after forward pass on the classical simulator of the IBM Brisbane quantum device.

These four stages are described in detail in Sections III-A–III-C.

A. HYBRID QUANTUM MOLGAN

We start the investigation into the efficacy of the hybrid generator within the MolGAN framework with a comparative analysis focusing on the chemical properties of the generated molecular samples. Specifically, this analysis evaluates the logP and QED scores. By contrasting the logP and QED scores yielded by the molecules generated from each model, we aim to quantify and elucidate the impact of the hybrid generator's integration on the model's performance in generating chemically viable and optimally structured molecules.

Fig. 3 illustrates a notable distinction in the behavior of the classical MolGAN and the HQ-MolGAN in terms of their generated molecular score distributions. The classical
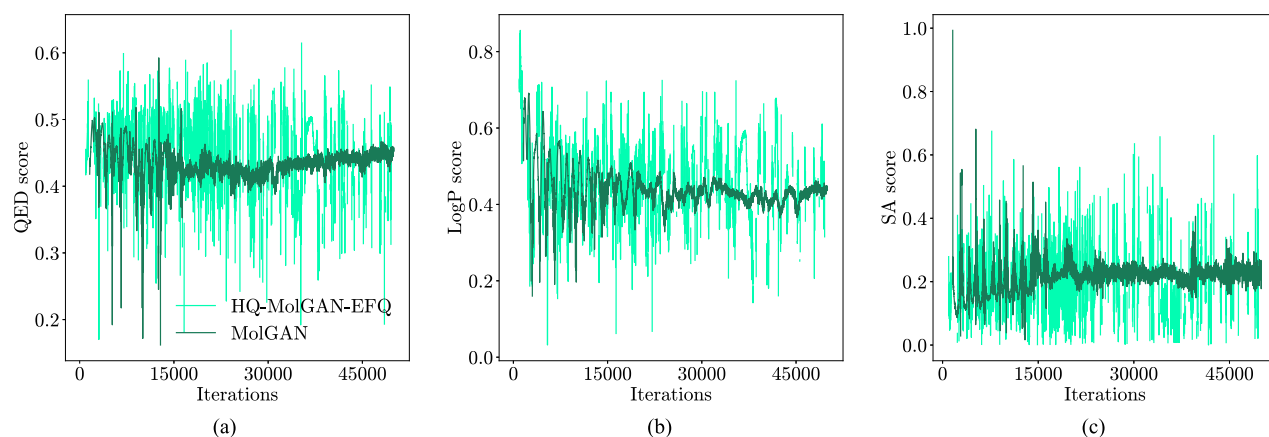
FIGURE 3. Chart of (a) QED, (b) LogP, and (c) SA scores during the training of classical MolGAN and hybrid MolGAN. It can be seen that while MolGAN limits its scores to a narrow beam of values even after 50 000 iterations, hybrid MolGAN presents a wider range of compounds, covering greater scores of key metrics.

TABLE 1. MolGAN and HQ-MolGAN

Model	Unique (%)	Valid (%)	Diversity	Druglikeliness	Synthesizability	Solubility
MolGAN (QM9)	63.0	1.1	0.98	0.47	0.64	0.52
MolGAN (PC9)	39.9	14.6	0.96	0.51	0.38	0.80
MolGAN (Both Datasets)	46.2	3.7	0.99	0.53	0.42	0.68
HQ-MolGAN-VVRQ (QM9)	**71.1**	14.3	0.97	0.53	**0.84**	0.61
HQ-MolGAN-VVRQ (PC9)	65.7	3.2	0.99	0.51	0.40	0.66
HQ-MolGAN-VVRQ (Both Datasets)	68.8	11.5	0.98	0.52	0.63	0.75
HQ-MolGAN-EFQ (QM9)	45.8	5.4	0.99	0.50	0.37	0.79
HQ-MolGAN-EFQ (PC9)	53.8	3.9	0.97	**0.62**	0.39	0.75
HQ-MolGAN-EFQ (Both Datasets)	39.4	12.9	0.97	0.53	0.49	**0.84**
QGAN-HG MR [9]	54.0	44.0	**1.00**	0.51	0.11	0.49
P2-QGAN-HG MR [9]	41.0	**52.0**	**1.00**	0.49	0.12	0.62
QuMolGAN [33]	5.4	42.94	**1.00**	0.57	0.76	0.44

The bold values are the maximum values.

MolGAN model demonstrates a tendency to produce scores that converge toward a relatively narrow range. In contrast, the molecular samples generated by HQ-MolGAN exhibit an oscillatory behavior in their score values. This variability in the HQ-MolGAN's scores can significantly influence the model's training dynamics, particularly due to the reward component, which calculates the product of these metric values.

A potential explanation for the discontinuous score trends observed in the HQ-MolGAN could be attributed to its limited validation set size. Nevertheless, the HQ-MolGAN can generate molecular samples with competitive scores in terms of druglikeness, synthesizability, and solubility, as depicted in Fig. 4(a). As shown in Table 1, the HQ-MolGAN-VVRQ model trained on the QM9 dataset achieves a LogP score of 0.84, the HQ-MolGAN-EFQ model trained on the PC9 dataset results in a QED score of 0.62, and the HQ-MolGAN-EFQ model trained on both datasets attains an SA score of 0.84.

The HQ-MolGAN-VVRQ model exhibits a propensity for generating samples with higher SA scores. In contrast, the HQ-MolGAN-EFQ model demonstrates superior performance in achieving greater QED and LogP scores.

Furthermore, a dataset-dependent variance in performance is observed. Models trained on the PC9 dataset consistently reach higher LogP scores compared with those trained on the

QM9 dataset. This trend aligns with the inherent distribution of scores within these datasets, as illustrated in Fig. 1. However, such a correlation does not extend to the QED scores, where no discernible pattern is evident based on the choice of training dataset.

These findings underscore the nuanced impact of model configuration and training dataset on the performance of HQ-MolGAN in generating molecular samples with desired chemical properties. They highlight the need for careful consideration of both the model architecture and the dataset characteristics in optimizing the performance of molecule generation models.

During our experiments, we observed a notable limitation of the hybrid models, characterized by the generator's tendency to gravitate toward a "high entropy state." This phenomenon is illustrated in Fig. 4(b). In this state, the generator predominantly produces molecular structures that are either bare unbound atoms or a collection of disconnected small molecules. Intriguingly, despite their simplistic and fragmented nature, these structures are often assigned high scores in terms of LogP and SA by the RDKit library within the reward component. This paradoxical scoring poses a challenge to the model's reliability in generating chemically meaningful and complex molecules.

This observation indicates a critical issue in the generator's mapping process. Essentially, various noise samples Z
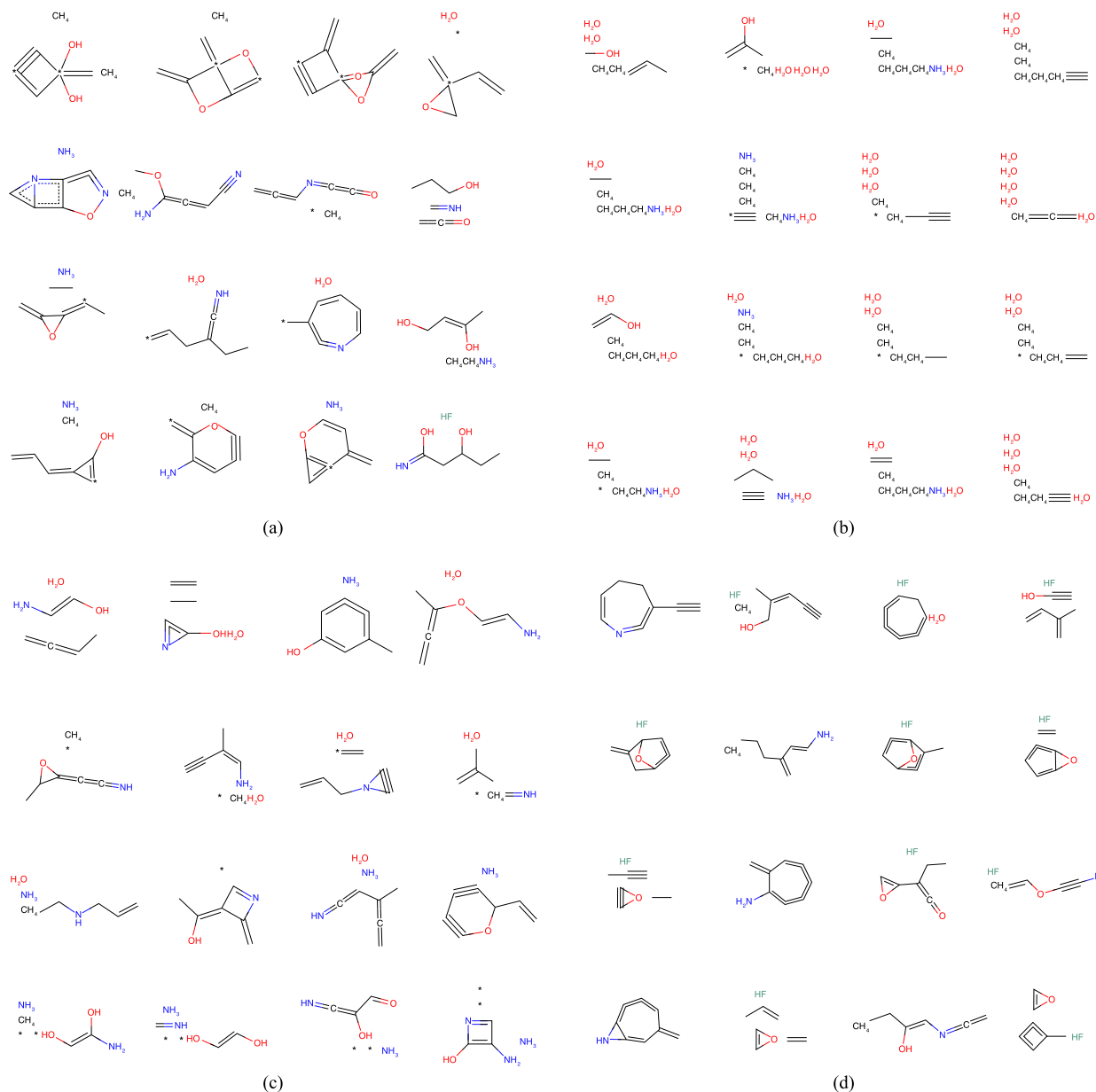
FIGURE 4. (a) Samples generated by HQ-MolGAN-VVRQ trained on QM9. (b) "High entropy state": HQ-MolGAN-VVRQ generated inappropriate samples and RDKit rewarded them with an average metric of LogP $\propto$ 0.9. (c) Samples generated by HQ-Cycle-MolGAN-VVRQ trained on both datasets. (d) Samples generated by MolGAN with HQ-Cycle trained on both datasets.

are mapped to a limited and similar region in the chemical space Y', resulting in repetitive and high-entropy molecular samples. Such a mapping significantly diminishes the generator's expressivity, constraining its ability to generate a diverse range of molecular structures.

To ensure that the models generate unique and varied molecular samples, it is imperative to establish a one-to-one correspondence between different noise vectors and distinct molecular structures. In other words, the model must possess isomorphic properties to map distinct noise vectors to chemically diverse molecular samples. To achieve this objective, the integration of a cycle component is proposed. The cycle component is designed to reinforce the isomorphic nature

of the model by facilitating a more diverse and accurate mapping from noise vectors to molecular samples and vice versa, thereby enhancing the model's capability to generate a wider array of unique molecular structures.

B. HYBRID QUANTUM CYCLE MOLGAN

Prior to assessing the effect of the cycle component on the training of HQ-MolGAN, it is essential to first examine its impact on the conventional MolGAN framework. As indicated in Fig. 5(a), the incorporation of the cycle component into MolGAN (termed Cycle-MolGAN) results in a more stable training process compared to the ordinary MolGAN model. This stability significantly enhances the quality of the
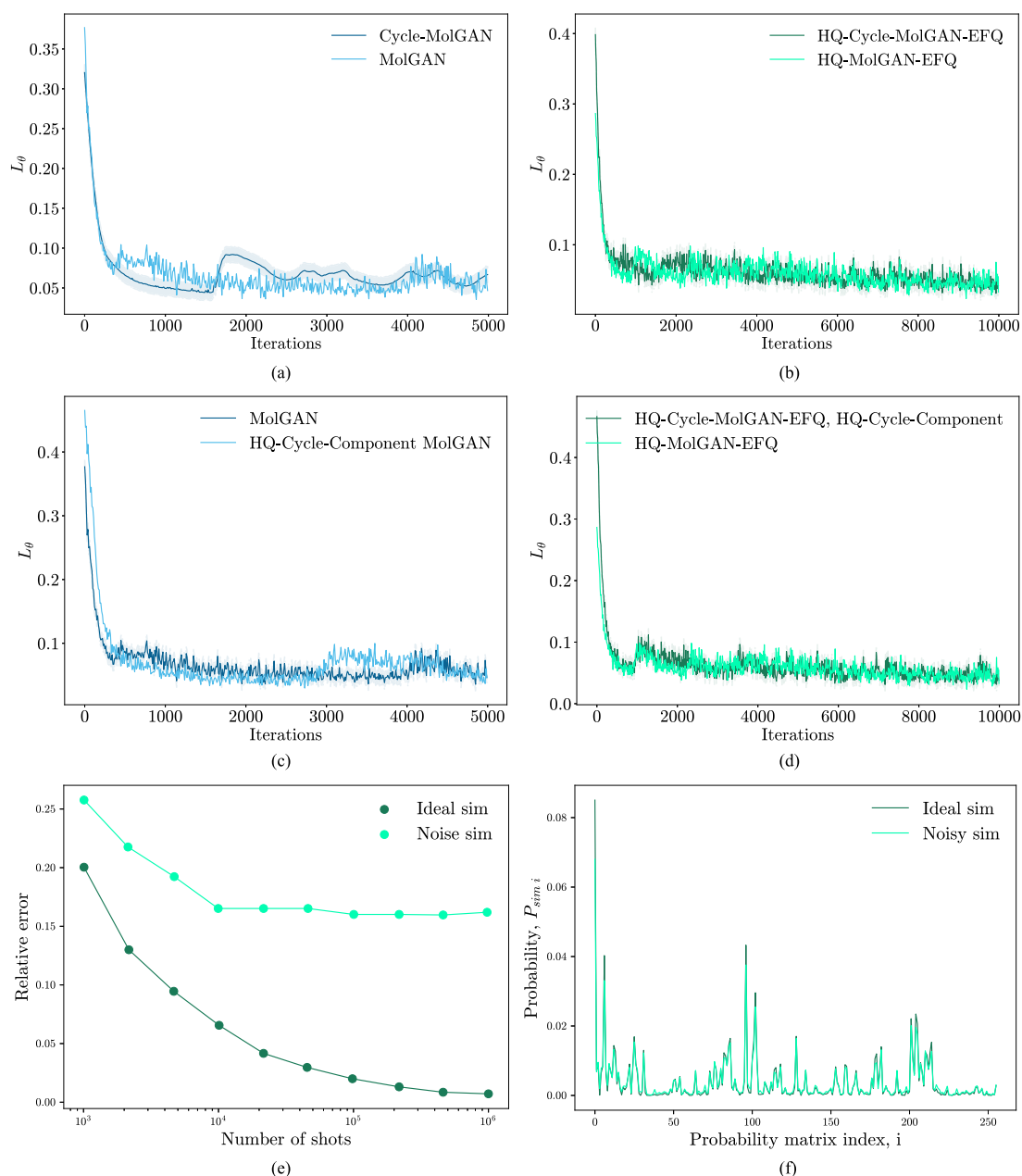
FIGURE 5. (a)–(d) present a comparison of the combined losses during training on the QM9 dataset. (a) MolGAN and Cycle-MolGAN. Cycle-MolGAN has a more stable training process compared to the MolGAN. (b) HQ-Cycle-MolGAN-EFQ and HQ-MolGAN-VVRQ. (c) MolGAN an HQ-Cycle-Component MolGAN. (d) HQ-Cycle-MolGAN-EFQ versus Hybrid-MolGAN-EFQ. No significant impact of the Cycle component on the loss curve is observed. (e) and (f) present charts for the IBM Brisbane execution. (e) Graph of the relative errors of the simulators' probability matrices with respect to the number of shots. (f) Comparison of the probabilities generated by the noisy and ideal simulators using 2×10^5 shots.

generated molecular samples, as reflected in their improved uniqueness scores (see Tables 1 and 2). Furthermore, the cycle component contributes to the generation of more complex and "bounded" molecular structures, indicating a higher degree of chemical realism [see Fig. 4(c) and (d)].

In the context of HQ-MolGAN, while the integration of the cycle component does not markedly alter the loss curve, as depicted in Fig. 5(b), its influence is evident in the improved key metric scores of the final HQ-Cycle-MolGAN models. Whether trained on the PC9 dataset or a combination of datasets, the HQ-Cycle-MolGAN demonstrates superior

performance in key metrics, as shown in Table 2. The stabilizing properties of the cycle component aid the model in consistently generating "bounded" molecular samples. Notably, both the VVRQ and EFQ variants of HQ-MolGAN achieve significant scores in terms of LogP (**0.93** and **0.94**) and QED.

In addition, an analysis of Tables 1 and 2 reveals that models equipped with the cycle component are capable of producing a greater number of unique samples. This finding aligns with the intended objective of the cycle component, which is to navigate through the "high entropy state" and

TABLE 2. MolGAN and HQ-MolGAN With Classic Cycle Component

Model	Unique (%)	Valid (%)	Diversity	Druglikeliness	Synthesizability	Solubility
Cycle-MolGAN (QM9)	86.3	0.7	**1.00**	0.47	0.37	0.46
Cycle-MolGAN (PC9)	67.8	3.2	0.98	0.48	0.27	0.52
Cycle-MolGAN (Both Datasets)	68.4	4.2	0.95	0.52	0.48	0.69
HQ-Cycle-MolGAN-VVRQ (QM9)	64.2	4.3	0.97	0.54	0.38	0.92
HQ-Cycle-MolGAN-VVRQ (PC9)	73.8	14.5	0.99	0.51	**0.50**	0.93
HQ-Cycle-MolGAN-VVRQ (Both Datasets)	**86.7**	6.8	0.98	**0.58**	0.48	0.75
HQ-Cycle-MolGAN-EFQ (QM9)	66.9	5.8	0.98	0.55	0.33	0.69
HQ-Cycle-MolGAN-EFQ (PC9)	81.2	**22.1**	0.96	0.54	0.40	**0.94**
HQ-Cycle-MolGAN-EFQ (Both Datasets)	64.1	7.5	0.98	0.53	0.35	0.66

The bold values are the maximum values.

TABLE 3. MolGAN and HQ-MolGAN With HQ-Cycle Component

Model	Unique (%)	Valid (%)	Diversity	Druglikeliness	Synthesizability	Solubility
Cycle-MolGAN (QM9)	92.4	2.7	**0.99**	0.47	0.35	0.64
Cycle-MolGAN (PC9)	93.2	5.1	0.97	0.46	0.28	0.65
Cycle-MolGAN (Both Datasets)	**93.9**	6.52	**0.99**	0.49	0.25	0.78
HQ-Cycle-MolGAN-VVRQ (QM9)	60.4	8.7	0.94	**0.53**	0.38	0.61
HQ-Cycle-MolGAN-VVRQ (PC9)	67.8	9.1	0.94	**0.53**	**0.50**	0.93
HQ-Cycle-MolGAN-VVRQ (Both Datasets)	65.5	**15.0**	0.98	0.51	0.35	**0.95**
HQ-Cycle-MolGAN-EFQ (QM9)	76.8	4.1	0.98	0.51	0.42	0.63
HQ-Cycle-MolGAN-EFQ (PC9)	88.8	11.0	0.98	0.50	0.35	0.69
HQ-Cycle-MolGAN-EFQ (Both Datasets)	74.7	9.3	0.96	0.52	0.49	0.73

The bold values are the maximum values.

enhance the diversity and uniqueness of the molecular samples generated by the model. The addition of cycle components provides a more stable (even smoother) training process for classical MolGAN. In terms of HQ-MolGAN, it rapidly increases its isomorphic properties that significantly help to get through a "high entropy state" during training. This increase in objectivity can be seen in the increase in uniqueness scores of cycle models.

C. HYBRID QUANTUM CYCLE MOLGAN

In the third stage of our experimental series, we focused on evaluating the impact of the HQ-Cycle component on both the classical MolGAN and the HQ-MolGAN architectures. In our simulations, the VVRQ generator uses a quantum circuit with $3 \times 8 \times 3$ parameters. In the case of the quasi-diffusion magnetic resonance imaging (QDI) layer in the HQ-Cycle component, the quantum circuit has 8×8 parameters.

Fig. 5(c) and (d) presents a comparative analysis of the generator losses between MolGAN and HQ-MolGAN models with and without the HQ-Cycle component. According to the data presented in Table 3, the incorporation of the HQ-Cycle component does not result in significant improvements in most of the desired metrics, except for a notable LogP score of 0.95 achieved by the HQ-Cycle-MolGAN-VVRQ model trained on both the datasets.

This absence of a marked enhancement in performance metrics for models incorporating the HQ-Cycle component, as compared to their counterparts without it, could potentially be attributed to an insufficient number of training iterations. This hypothesis is supported by the observations from Fig. 5(c), where the training losses of HQ-MolGAN and HQ-MolGAN with the HQ-Cycle component exhibit minimal divergence, suggesting that extended training might be necessary for realizing the full potential of the HQ-Cycle component.

Interestingly, the introduction of the HQ-Cycle component appears to significantly elevate the "unique" score of the models, surpassing even that achieved with the standard cycle component. This outcome validates our initial hypothesis that a more precise generation of unique molecular samples is feasible, even with complex models, such as HQ-Cycle-MolGAN-EFQ or HQ-Cycle-MolGAN-VVRQ. This finding underscores the effectiveness of the HQ-Cycle component in enhancing the diversity and uniqueness of the generated molecular structures.

D. EXECUTION ON SIMULATORS OF NOISY QUANTUM DEVICES

In the last section of our numerical experiments, we explore the potential of executing the HQ-MolGAN-VVRQ model on quantum devices. For that exploration, we generated molecular samples using two IBM simulators: the "noisy simulator" of the IBM Brisbane quantum computer and the "ideal simulator" of the noiseless IBM Brisbane quantum computer [49], [50].

To perform the numerical experiments, we took the generator of the HQ-MolGAN-VVRQ model and separated it into two parts: VQC and MLP. In the experiments, we executed the VQC part on the simulator of the noisy and noiseless quantum hardware and fed the results to MLP executed on classical hardware. In the noisy simulation, we performed quantum operations on eight noisy qubits with the best fidelities out of 127 available on the IBM Brisbane quantum computer. In the ideal simulation, we performed the same VQC with the same initial Gaussian-distributed vector on an ideal simulator.

The comparison between noisy and ideal simulations is shown in Fig. 5(e), where relative losses of noisy and ideal simulators with respect to the number of given shots are

TABLE 4. HQ-MolGAN-VVRQ Forward Pass on the Noisy (IBM Brisbane) and Ideal (Qiskit) Simulators Using 2×10^5 Shots Budget

Model	Unique (%)	Valid (%)	Diversity	Druglikeliness	Synthesizability	Solubility
Noisy HQ-MolGAN-VVRQ (QM9)	**80.0**	**6.52**	**0.97**	**0.44**	**0.23**	**0.76**
Ideal HQ-MolGAN-VVRQ (QM9)	**80.0**	6.51	**0.97**	**0.44**	**0.23**	0.75

The bold values are the maximum values.

shown. The relative error is estimated as

$$\text{Err}(N) = \frac{| \sum_{n=1}^{256} [P_{\text{sim } i}(N) - P_{\text{ideal } i}(+\infty)] |}{\sum_{n=1}^{256} P_{\text{ideal } i}(+\infty)}$$

where $P_{\text{ideal}}(+\infty)$ is the matrix of probabilities generated on the ideal simulator after a large number of iterations ($N \rightarrow +\infty$) and $P_{\text{sim}}(N)$ is the matrix of probabilities generated on the specific simulator (either ideal or noisy) after N shots.

The relative error of the ideal simulator approaches zero as the number of shots grows, as shot noise is the only source of error. For the noisy simulator, the relative error hits a limit specific to the noise model of the quantum device. This systematic error of the device has an impact on the probability values given by the circuit, leading to slightly different initial states of the vector [see Fig. 5(f)], which is given to the MLP layer in the course of generation.

For the generation of molecular samples, we created 1000 vectors $[x_1, .., x_8]$, $x_i \sim \mathcal{N}(0, 1)$, forward passed them on both simulators, postprocessed them, and then used them in the MLP layer. After that, the generated molecular graph properties were evaluated. As seen in Table 4, molecular graphs generated on the noisy simulator have slightly greater validity and solubility. This may be because, on the one hand, the probability vectors obtained on the noisy simulator do not differ too much from the ideal one. On the other hand, the MLP layer can play a role as an error correction algorithm.

IV. DISCUSSION

In this article, we propose a novel approach leveraging QML for small molecule generation. Our chosen task of small molecule generation serves as a benchmark for the performance of hybrid QML models.

To enhance the classical MolGAN, we introduced two solutions: the incorporation of VQCs as the initial layer of the generator and the utilization of a cycle component to restore the original data from the graph representation of the generated molecule.

Our empirical results substantiate the merit of diversifying training datasets, not limiting to the QM9 dataset alone but also incorporating the PC9 dataset. Notably, the HQ-MolGAN model, with their generator's layers scaled down by half and trained for four times fewer iterations, has outperformed the classical MolGAN model [8] and its hybrid quantum analogues [9], [33] across key chemical metrics: QED, logP, SA, and uniqueness. The HQ-MolGAN model was also tested for potential execution on quantum computers. By using the noisy and ideal simulators of an IBM quantum computer, we observed that the HQ-MolGAN model is resilient to noise: the achieved scores of the noisy simulation are shown to be similar to the ideal noise-free simulation.

The introduction of the cycle component to HQ-MolGAN, and especially its hybrid quantum variant, marks an advance in the training of hybrid quantum models. These models not only enhance the desired uniqueness score but also effectively mitigate the occurrence of the "high entropy state," a notable challenge in molecular generation tasks. Consequently, these models hold substantial promise for applications in the domain of small drug compound design, both for commercial and scientific purposes within pharmacology.

This work contributes insights into the potential of QML for small molecule generation, emphasizing the benefits of hybrid quantum–classical approaches in drug design. The results underscore the significance of employing quantum-enhanced models to achieve improved performance across essential molecular optimization metrics.

Looking ahead, we see a potential for hybrid QML models to further advance the field of molecule generation using hybrid quantum models. We plan to delve deeper into refining the model architecture, particularly focusing on optimizing the balance between the quantum and classical components. This involves experimenting with different configurations and parameters to enhance the overall efficiency and accuracy of the models.

Another critical avenue we intend to pursue is the expansion of our training datasets. By incorporating a broader range of chemical compounds and molecular structures, we aim to increase the diversity and representativeness of our models. This expansion is expected to improve the models' generalization capabilities and their applicability. Through these focused research efforts, we aspire to contribute significantly to the advancement of hybrid quantum computing in drug discovery, ultimately aiding in the development of more effective and innovative therapeutic solutions.

APPENDIX A
QUANTUM CIRCUIT ANALYSIS

In this appendix, we analyze the quantum circuits employed in the HQ-Cycle-MolGAN framework, specifically the VVRQ and the QDI layers. We assess these circuits using several metrics:

1) ZX-calculus circuit reducibility;
2) Fisher information degeneracy;

A. ZX-CALCULUS

ZX-calculus serves as a graphical language capable of depicting a quantum circuit through diagrams consisting of "spider"—nodes interconnected by edges. These ZX diagrams can be simplified [51] and minimized using the language's graphical rewriting rules [52], which are grounded in the fundamentals of quantum operations. By simplifying
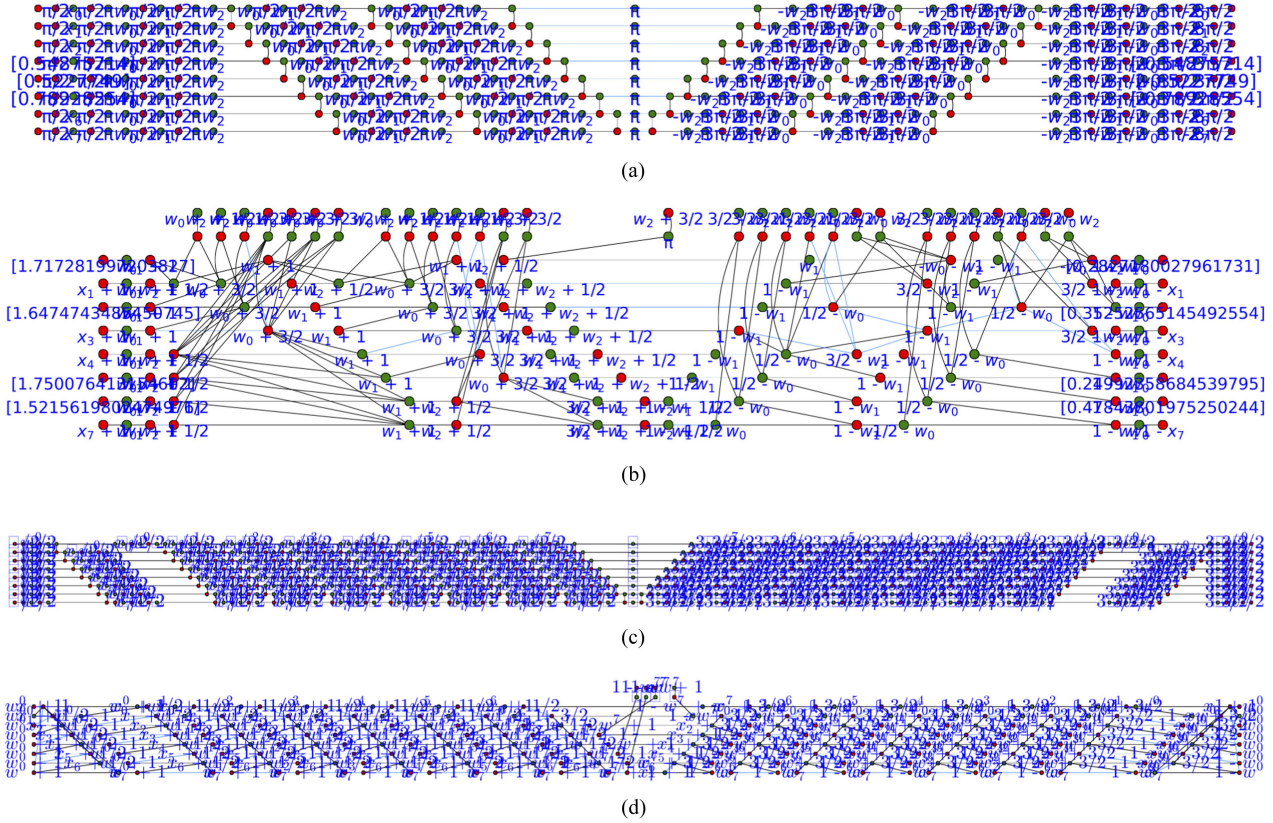
FIGURE 6. (a) VVRQ layer with original parameters. (b) VVRQ layer without redundant parameters. (c) QDI layer with original parameters. (d) QDI layer without redundant parameters.

these diagrams, we can derive a more efficient circuit configuration. Moreover, ZX-calculus offers a metric for evaluating circuit efficiency by comparing the number of parameters in the simplified diagram against the initial number of parameters. A reduction in redundant parameters indicates enhanced circuit performance. A circuit deemed unable to be optimized in this manner is classified as ZX-irreducible.

The key adjustments to the circuit shown in Fig. 6(a) and (b) consist of the rearrangement of certain weights after their reduction. During the optimization phase, 139 out of 150 parameters (approximately 93%) were preserved, illustrating the circuit's significant degree of optimization. As illustrated in Fig. 6(c) and (d), for QDI, the ZX-calculus algorithm merely adjusted some of the weights following their reduction. Throughout the optimization process, 269 out of 272 parameters (about 99%) were maintained, indicating that the circuit is highly optimized and yields almost perfect outcomes.

By using the ZX-calculus algorithm, it was revealed that both VVRQ and QDI perform very well and have close to no redundant parameters. However, other metrics should be applied to obtain a more precise analysis.

B. FISHER INFORMATION

A supervised machine learning task can be described as the creation of a hypothesis model $h_\theta(\hat{x})$ based on a labeled

dataset $(x, y) \in X \times Y$ to provide an approximation of the distribution, $f(x)$, of the data in nature. Using a subset of S labeled data points from this distribution, we optimize our hypothesis model to provide high-accuracy modeling of $f(\hat{x})$. For this, we maximize the probability of acquiring the associated label y from the model with parameters θ and data points x. The needed conditional probability can be written as $P(y|x, \theta)$. Taking into account the uniform distribution over X, the joint probability $P(y, x|\theta)$ is used for better accuracy, and its distribution can be calculated for any value of θ for a data points x_i. Thus, we represent the joint probability as an N-dimensional manifold with N as the number of trainable parameters $N = |\theta|$. The Fisher information matrix (FIM) $F(\theta)$ [53], [54] is a metric over this manifold

$$F(\theta) = \mathbb{E}_{\{x_i, y_i\}} \left[\nabla_\theta \log(P) \nabla_\theta \log(P)^T \right]. \quad (1)$$

The next step is to diagonalize this metric to get a locally Euclidean tangential basis, where the diagonal values are the square gradient of our joint probability in this basis. These are the eigenvalues of the Fisher matrix. This is very important to detect and prevent the barren plateau problem, which involves vanishing gradients with a high number of qubits in quantum neural networks (QNNs).

As shown in [55], their expectation values become zero, and their variance decreases exponentially with a growing number of qubits. This can be seen if the gradients mostly
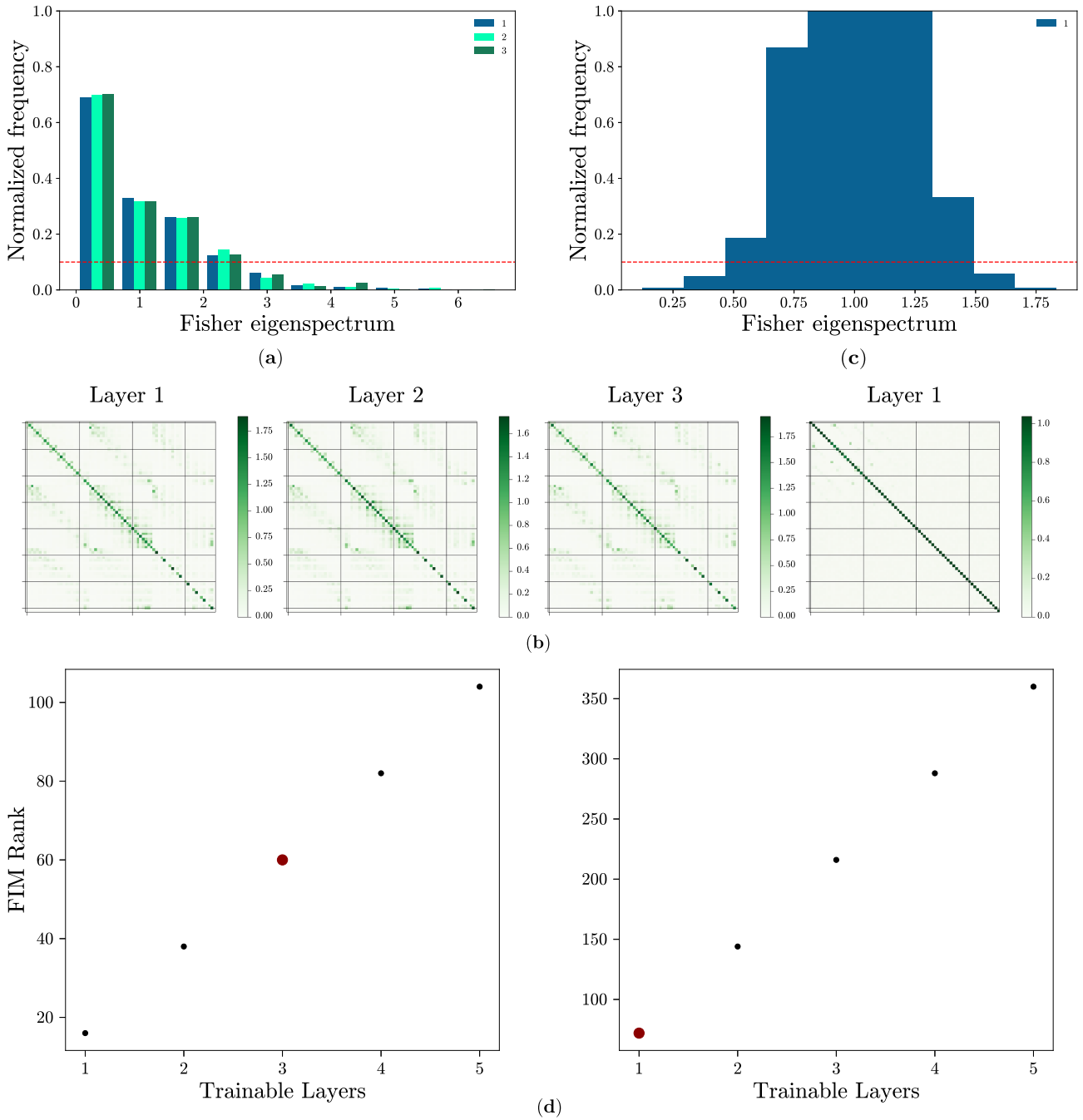
FIGURE 7. (a) Normalized histogram of (left) the VVRQ and (right) QDI Fisher eigenspectrum. For VVRQ, all three layers have the first four parameters contributing the most impact (with the third layer adding extra frequency to the first parameter). This circuit is moderately expressive. For QDI, the majority of parameters achieved very high frequencies, which indicates excellent trainability. (b) and (c) Average Fisher matrices for (left) VVRQ and (right) QDI. For VVRQ, the diagonal elements show that the circuit distributes the gradients to all trainable parameters with no evident single-parameter dominance (only a slight gradient shift toward the latter parameters for the deeper layers). For QDI, the diagonal elements show that the circuit distributes the gradients to all trainable parameters with no single-parameter dominance and almost no nondiagonal element, indicating high trainability. (d) FIM rank for VVRQ (left, 3/5 layers used) and QDI (right, 1/5 layers used) illustrates the circuit's limit of overparameterization. For VVRQ, the circuit is not overparameterized, and the addition of the second layer more than doubled the rank (from 16 to 38). The third layer adds the same amount, so it is not necessary to increase further. For QDI, taking into account excellent eigenspectrum performance and the usual increase from the first to the second layer (from 72 to 144, doubled, unlike the VVRQ), it is unnecessary to add new layers.

degenerate near zero, which means that many parameters do not participate in training at all. Therefore, calculating the eigenvalue spectrum of Fisher matrices for many realizations of θ helps investigate the trainability and robustness of the QNN against barren plateaus. A more highly trainable neural network would have less eigenvalue degeneracy.

The FIM can be calculated for the specific hyperparameters of our circuit. Using a method from [53], we create a Gaussian dataset $x \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$. Then, the joint probability can be found by overlapping the computed state and the state of our quantum layer

$$P(y, x|\theta) = \langle y|\psi(\theta, x)\rangle \qquad (2)$$

where y is the output state. By averaging over all x and y, we can calculate the Fisher information for any given θ.

As a result, we can see in Fig. 7(a) that both circuits have at least half of their parameters significantly impacting the result. The QDI shows especially good results with four highly impactful parameters and two moderately impactful parameters. Fig. 7(b) shows the average Fisher matrices with no redundant elements on the diagonal. This shows that all the parameters are used in training, which leads to high trainability.

Larocca et al. [56] stated that some QNNs may show lowered parameter efficiency due to overparameterization. This was calculated by finding that, at some point, parameter addition leaves the rank of the FIM rank unchanged. This happens when the circuit starts to become saturated. After that, there is no increase in expressivity, and there can be a risk of overparameterization. As one can see in Fig. 7(d), the addition of new layers does not show any overparameterization on this scale. Hence, the increase in rank with additional layers (and other metrics) can determine the necessity of structure change. As the previous analysis showed, VVRQ performs quite well, which may indicate that a more complex circuit is not needed. At least two layers are required since the addition of the second one changed the rank from 16 to 38, while the expected scenario would be an increase of the same amount as the first one (to 32). It shows the underdevelopment of the first layer that is fixed by the addition of a new one. For QDI, the circuit is already optimized enough and does not show any underdevelopment for the first layer. Thus, the need for the addition of new layers can be determined by how much the rank increases, complemented by other methods (such as the Fisher eigenspectrum analysis).

APPENDIX B
HYBRID QUANTUM GENERATOR OSCILLATORY BEHAVIOR

HQ-MolGAN models have shown oscillatory behavior in terms of chemical metrics during training (see Fig. 3). The reason for this phenomenon may lie in the breakdown of the interplay between the hybrid quantum generator and the classical discriminator. To investigate this possibility, we provide
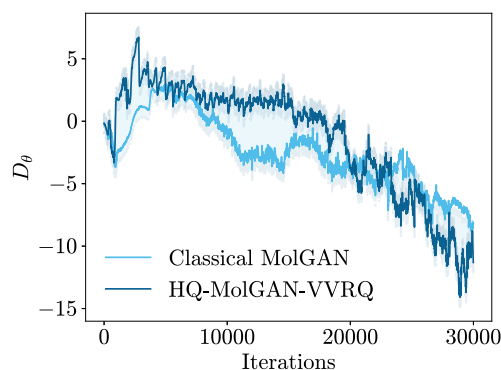

FIGURE 8. Loss curve of the discriminator for the classical MolGAN and the HQ-MolGAN-VVRQ.

a loss chart of the discriminator component during the training of the classical MolGAN and the HQ-MolGAN-VVRQ (PC9).

Fig. 8 shows that the discriminator competing with the hybrid quantum generator does not show any significant change compared to the classical MolGAN generator. Also, neither chart shows any significant increase after 15 000 iterations, which means that MolGAN's generative expressiveness converges to a narrow beam of values due to the generator's properties and not the discriminator's.

REFERENCES

[1] J. P. Hughes, S. Rees, S. B. Kalindjian, and K. L. Philpott, "Principles of early drug discovery," *Brit. J. Pharmacol.*, vol. 162, no. 6, pp. 1239–1249, 2011, doi: 10.1111/j.1476-5381.2010.01127.x.

[2] F. Wang, Y.-T. Chen, J.-M. Yang, and T. Akutsu, "A novel graph convolutional neural network for predicting interaction sites on protein kinase inhibitors in phosphorylation," *Sci. Rep.*, vol. 12, no. 1, 2022, Art. no. 229, doi: 10.1038/s41598-021-04230-7.

[3] I. J. Goodfellow et al., "Generative adversarial networks," 2014, *arXiv:1406.2661*, doi: 10.48550/arXiv.1406.2661.

[4] R. M. Schmidt, "Recurrent neural networks (RNNs): A gentle introduction and overview," 2019, *arXiv:1912.05911*, doi: 10.48550/arXiv.1912.05911.

[5] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations Trends in Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019, doi: 10.1561/2200000056.

[6] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, 1988, doi: 10.1021/ci00057a005.

[7] A. I. Gircha, A. S. Boev, K. Avchaciov, P. O. Fedichev, and A. K. Fedorov, "Hybrid quantum-classical machine learning for generative chemistry and drug design," *Sci. Rep.*, vol. 13, no. 1, 2023, Art. no. 8250, doi: 10.1038/s41598-023-32703-4.

[8] N. De Cao and T. Kipf, "MolGAN: An implicit generative model for small molecular graphs," 2022, *arXiv:1805.11973*, doi: 10.48550/arXiv.1805.11973.

[9] J. Li, R. O. Topaloglu, and S. Ghosh, "Quantum generative models for small molecule drug discovery," *IEEE Trans. Quantum Eng.*, vol. 2, 2021, Art. no. 3103308, doi: 10.1109/TQE.2021.3104804.

[10] A. Melnikov, M. Kordzanganeh, A. Alodjants, and R.-K. Lee, "Quantum machine learning: From physics to software engineering," *Adv. Phys.: X*, vol. 8, no. 1, 2023, Art. no. 2165452, doi: 10.1080/23746149.2023.2165452.

[11] S. Jerbi, L. J. Fiderer, H. P. Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, "Quantum machine learning beyond kernel methods," *Nature Commun.*, vol. 14, no. 1, 2023, Art. no. 517, doi: 10.1038/s41467-023-36159-y.

[12] A. Pérez-Salinas, R. Draškić, J. Tura, and V. Dunjko, "Shallow quantum circuits for deeper problems," *Phys. Rev. A*, vol. 108, Dec. 2023, Art. no. 062423, doi: 10.1103/PhysRevA.108.062423.

[13] C. Simon, C. M. Gyurik, and V. Dunjko, "High dimensional quantum machine learning with small quantum computers," *Quantum*, vol. 7, 2023, Art. no. 1078, doi: 10.22331/q-2023-08-09-1078.

[14] M. Kordzanganeh, D. Kosichkina, and A. Melnikov, "Parallel hybrid networks: An interplay between quantum and classical neural networks," *Intell. Comput.*, vol. 2, 2023, Art. no. 0028, doi: 10.34133/icomputing.0028.

[15] A. Senokosov, A. Sedykh, A. Sagingalieva, B. Kyriacou, and A. Melnikov, "Quantum machine learning for image classification," *Mach. Learn.: Sci. Technol.*, vol. 5, no. 1, 2024, Art. no. 015040, doi: 10.1088/2632-2153/ad2aef.

[16] Y. Li, R.-G. Zhou, R. Xu, J. Luo, and W. Hu, "A quantum deep convolutional neural network for image recognition," *Quantum Sci. Technol.*, vol. 5, no. 4, 2020, Art no. 044003, doi: 10.1088/2058-9565/ab9f93.

[17] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A*, vol. 98, no. 3, 2018, Art. no. 032309, doi: 10.1103/PhysRevA.98.032309.

[18] E. H Houssein, Z. Abohashima, M. Elhoseny, and W. M. Mohamed, "Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images," *J. Comput. Des. Eng.*, vol. 9, no. 2, pp. 343–363, 2022, doi: 10.1093/jcde/qwac003.

[19] L. Lusnig et al., "Hybrid quantum image classification and federated learning for hepatic steatosis diagnosis," *Diagnostics*, vol. 14, no. 5, 2024, Art. no. 558, doi: 10.3390/diagnostics14050558.

[20] P. Jain and S. Ganguly, "Hybrid quantum generative adversarial networks for molecular simulation and drug discovery," 2022, *arXiv:2212.07826*, doi: 10.48550/arXiv.2212.07826.

[21] A. Sedykh, M. Podapaka, A. Sagingalieva, K. Pinto, M. Pflitsch, and A. Melnikov, "Hybrid quantum physics-informed neural networks for simulating computational fluid dynamics in complex shapes," *Mach. Learn.: Sci. Technol.*, vol. 5, no. 2, 2024, Art. no. 025045, doi: 10.1088/2632-2153/ad43b2.

[22] A. Kurkin, J. Hegemann, M. Kordzanganeh, and A. Melnikov, "Forecasting the steam mass flow in a powerplant using the parallel hybrid network," 2023, *arXiv:2307.09483*, doi: 10.48550/arXiv.2307.09483.

[23] A. Sagingalieva et al., "Photovoltaic power forecasting using quantum machine learning," 2023, *arXiv:2312.16379*, doi: 10.48550/arXiv.2312.16379.

[24] N. Haboury et al., "A supervised hybrid quantum machine learning solution to the emergency escape routing problem," 2023, *arXiv:2307.15682*, doi: 10.48550/arXiv.2307.15682.

[25] S. Rainjonneau et al., "Quantum algorithms applied to satellite mission planning for earth observation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 7062–7075, 2023, doi: 10.1109/JSTARS.2023.3287154.

[26] A. Sagingalieva et al., "Hybrid quantum ResNet for car classification and its hyperparameter optimization," *Quantum Mach. Intell.*, vol. 5, no. 2, 2023, Art. no. 38, doi: 10.1007/s42484-023-00123-2.

[27] J. Landman et al., "Quantum methods for neural networks and application to medical image classification," *Quantum*, vol. 6, 2022, Art. no. 881, doi: 10.22331/q-2022-12-22-881.

[28] M. Perelshtein et al., "Practical application-specific advantage through hybrid quantum computing," 2022, *arXiv:2205.04858*, doi: 10.48550/arXiv.2205.04858.

[29] G G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins, "Quantifying the chemical beauty of drugs," *Nature Chem.*, vol. 4, no. 2, pp. 90–98, 2012, doi: 10.1038/nchem.1243.

[30] P. Ertl and A. Schuffenhauer, "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions," *J. Cheminform.*, vol. 1, 2009, Art. no. 8, doi: 10.1186/1758-2946-1-8.

[31] A. Leo, C. Hansch, and D. Elkins, "Partition coefficients and their uses," *Chem. Rev.*, vol. 71, no. 6, pp. 525–616, 1971, doi: 10.1021/cr60274a001.

[32] Ł. Maziarka, A. Pocha, J. Kaczmarczyk, K. Rataj, T. Danel, and M. Warchoł, "Mol-CycleGAN: A generative model for molecular optimization," *J. Cheminform.*, vol. 12, 2020, Art. no. 1, doi: 10.1186/s13321-019-0404-1.

[33] P.-Y. Kao et al., "Exploring the advantages of quantum generative adversarial networks in generative chemistry," *J. Chem. Inf. Model.*, vol. 63, no. 11, pp. 3307–3318, 2023, doi: 10.1021/acs.jcim.3c00562.

[34] R. Ramakrishnan, O. Pavlo, M. D. Rupp, and O. A. Von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Sci. Data*, vol. 1, no. 1, 2014, Art. no. 140022, doi: 10.1038/sdata.2014.22.

[35] L. Ruddigkeit, R. van Deursen, C. L. Blum, and J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17," *J. Chem. Inf. Model.*, vol. 52, no. 11, pp. 2864–2875, 2012, doi: 10.1021/ci300415d.

[36] M. Glavatskikh, J. Leguy, G. Hunault, T. Cauchy, and B. Da Mota, "Dataset's chemical diversity limits the generalizability of machine learning predictions," *J. Cheminform.*, vol. 11, no. 1, 2019, Art. no. 69, doi: 10.1186/s13321-019-0391-2.

[37] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*, doi: 10.48550/arXiv.1701.07875.

[38] P. Ertl, S. Roggo, and A. Schuffenhauer, "Natural product-likeness score and its application for prioritization of compound libraries," *J. Chem. Inf. Model.*, vol. 48, no. 1, pp. 68–74, 2007, doi: 10.1021/ci700286x.

[39] M. Kordzanganeh et al., "Benchmarking simulated and physical quantum processing units using quantum and hybrid algorithms," *Adv. Quantum Technol.*, vol. 6, no. 8, 2023, Art. no. 2300043, doi: 10.1002/qute.202300043.

[40] M. Kordzanganeh, P. Sekatski, L. Fedichkin, and A. Melnikov, "An exponentially-growing family of universal quantum circuits," *Mach. Learn.: Sci. Technol.*, vol. 4, no. 3, 2023, Art. no. 035036, doi: 10.1088/2632-2153/ace757.

[41] F. Bloch, "Nuclear induction," *Phys. Rev.*, vol. 70, no. 7/8, 460–474, 1946, doi: 10.1103/PhysRev.70.460.

[42] S. L. Tsang, M. T. West, S. M. Erfani, and M. Usman, "Hybrid quantum–classical generative adversarial network for high-resolution image generation," *IEEE Trans. Quantum Eng.*, vol. 4, 2023, Art. no. 3102419, doi: 10.1109/TQE.2023.3319319.

[43] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Phys. Rev. A*, vol. 103, 2021, Art. no. 032430, doi: 10.1103/PhysRevA.103.032430.

[44] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251, doi: 10.1109/ICCV.2017.244.

[45] A. Sagingalieva, M. Kordzanganeh, N. Kenbayev, D. Kosichkina, T. Tomashuk, and A. Melnikov, "Hybrid quantum neural network for drug response prediction," *Cancers*, vol. 15, no. 10, 2023, Art. no. 2705, doi: 10.3390/cancers15102705.

[46] PyTorch—Machine Learning framework, 2022. [Online]. Available: https://pytorch.org/

[47] V. Bergholm et al., "PennyLane: Automatic differentiation of hybrid quantum-classical computations," 2018 *arXiv:1811.04968*, doi: 10.48550/arXiv.1811.04968.

[48] A. Javadi-Abhari et al., "Quantum computing with Qiskit," 2024, *arXiv:2405.08810*.

[49] "IBM: Using IBM quantum cloud-based simulators," 2024. [Online]. Available: https://docs.quantum.ibm.com/verify/using-ibm-quantum-simulators

[50] "IBM: Build noise models," 2024. [Online]. Available: https://docs.quantum.ibm.com/verify/building_noise_models

[51] B. Coecke and R. Duncan, "Interacting quantum observables," in *Automata, Languages and Programming*, vol. 13. Berlin, Germany: Springer, 2008, pp. 298–310, doi: 10.1007/978-3-540-70583-3_25.

[52] J. van de Wetering, "ZX-calculus for the working quantum computer scientist," 2020, *arXiv:2012.13966*, doi: 10.48550/arXiv.2012.13966.

[53] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nature Comput. Sci.*, vol. 1, pp. 403–409, 2020, doi: 10.1038/s43588-021-00084-1.

[54] S. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998, doi: 10.1162/089976698300017746.

[55] J. R. McClean, S. Boixo, and V. N. Smelyanskiy, "Barren plateaus in quantum neural network training landscapes," *Nature Commun.*, vol. 9, 2018, Art. no. 4812, doi: 10.1038/s41467-018-07090-4.

[56] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, "Theory of overparametrization in quantum neural networks," *Nature Comput. Sci.*, vol. 3, no. 6, pp. 542–551, 2023, doi: 10.1038/s43588-023-00467-6.