Shallow instantaneous quantum polynomial-time circuits for generative modeling on noisy intermediate-scale quantum hardware

Oriol Balló-Gimbernat [1,2,3,*] Marcos Arroyo-Sánchez [1,2,3] Paula García-Molina [2]
Adan Garriga [1] and Fernando Vilariño [2,3]
[1] *Eurecat, Centre Tecnològic de Catalunya, Barcelona, Spain*
[2] *Centre de Visió per Computador (CVC), Barcelona, Spain*
[3] *Universitat Autònoma de Barcelona (UAB), Barcelona, Spain*

(Received 13 November 2025; accepted 12 March 2026; published 15 April 2026)

Generative modeling is one of the most promising applications of quantum machine learning, yet training and deploying quantum generative models (QGMs) on near-term hardware remains effectively intractable due to prohibitive gradient estimation and implementation costs. We propose a resource-efficient approach based on shallow instantaneous quantum polynomial-time (IQP) circuits that circumvents these bottlenecks by leveraging efficient classical training while retaining the guarantee of sampling hardness. To validate this approach, we formalize graph generation as a hierarchy of physical correlations, allowing us to map abstract data features—such as edge density and bipartiteness—directly to the quantum observables required to learn them. We validate our protocol through demonstrations both on real hardware (from 28 to 153 qubits) and simulations (28 qubits). Results show that while global structural features exhibit significant degradation beyond 91 qubits, our models achieve high-precision reproduction of local correlations, even up to 153 qubits. These findings establish shallow IQP circuits as a robust, scalable candidate for generative tasks on current noisy intermediate-scale quantum (NISQ) devices.

DOI: 10.1103/jcdk-q3xc

I. INTRODUCTION

Generative modeling stands as a cornerstone of unsupervised machine learning, enabling the synthesis of data that captures the underlying statistics of complex distributions. Within this domain, quantum generative models (QGMs) have emerged as a promising frontier, leveraging quantum computation to represent probability distributions that are potentially intractable for classical systems [1–3]. A particularly compelling architecture in this landscape is the quantum circuit born machine (QCBM) [4–7]. However, training QCBMs on near-term hardware presents significant hurdles, primarily due to the poor scaling of gradient estimation costs and the prevalence of barren plateaus, which currently restricts training to small-scale models. Furthermore, even in instances where large-scale models can be trained without barren plateaus, efficient implementation and robust guarantees of classical hardness remain elusive [8].

In this work, we propose a generative modeling approach that overcomes these bottlenecks by leveraging the unique properties of instantaneous quantum polynomial-time (IQP) circuits. IQP circuits occupy a prime position for near-term applications: they are conjectured to be hard to sample from classically [9,10], yet their specific algebraic structure permits the efficient classical calculation of certain output statistics that can be used for training [11,12]. We introduce a hybrid workflow based on shallow IQP circuits that exploits this duality—training the model on classical hardware and deploying it on quantum processors for sampling (see Fig. 1) [11,13]—while minimizing the resources required for implementation and retaining classical hardness. This approach effectively circumvents the scalability and implementation challenges inherent in standard QCBMs.

To rigorously validate this approach and characterize the expressivity of shallow IQP circuits, we use graph generation as a testbed. Graphs represent relationships between entities and provide a unified framework to describe complex systems via their components and interactions [14]. Their versatility has driven the development of graph generation methods for applications ranging from drug discovery to scheduling [15–19]. Learning a graph distribution remains challenging because node interactions span both local and global dependencies [20–25]. As the number of nodes increases, the combinatorial complexity of pairwise and higher-order relationships grows rapidly: adding a single node to an M-node graph introduces M new potential edges, so that the number of possible configurations increases from $2^{\binom{M}{2}}$ to $2^{\binom{M}{2}} \cdot 2^M$. This exponential scaling implies that even modest increases in graph size can dramatically expand the hypothesis space. Although practical problems typically restrict attention to structured graph families, models must still possess sufficient expressivity to navigate this high-dimensional space [23,26].

Rather than treating graph generation as a downstream application, we utilize it to map abstract data features to physical

*Contact author: oriol.ballo@eurecat.org

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

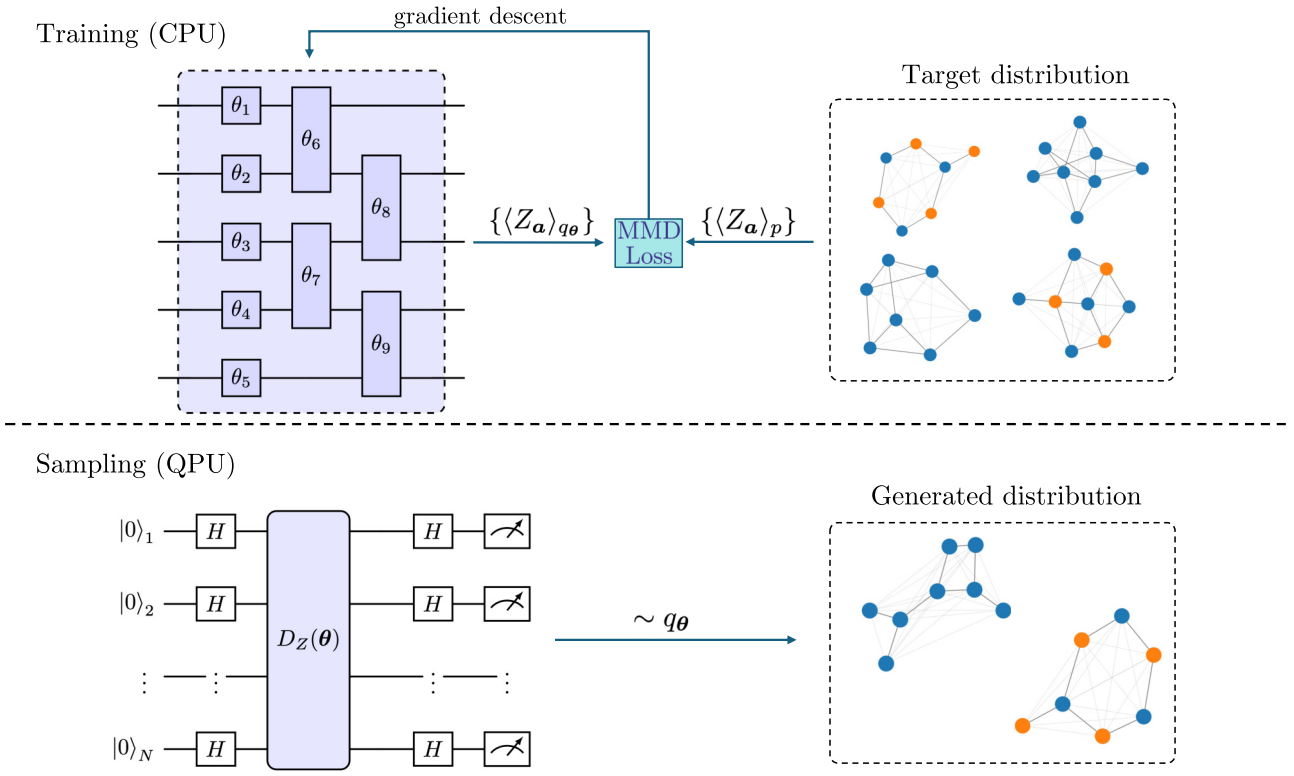


FIG. 1. Overview of the proposed workflow. Top: Model training is performed entirely on classical hardware via a classical estimation of the MMD loss. Bottom: Sampling is carried out on IBM's Aachen quantum processor.

correlations. Simple properties like edge density correspond to local (one-body) correlations, while structural constraints like bipartiteness correspond to global ones. This formalization allows us to disentangle the model's capacity to learn features of different complexities.

We demonstrate the efficacy of this approach by training shallow IQP models to generate random bipartite and Erdős-Rényi graph distributions. Our demonstration results, obtained on an IBM Heron r2 156 qubit processor, show that the proposed models are highly effective at reproducing low-bodied correlations, even up to 153 qubits, while global structural features exhibit higher performance degradation beyond 91 qubits due to hardware noise. These findings establish shallow IQP circuits as a resource-efficient candidate for near-term generative tasks.

The paper is organized as follows. Section II defines the target graph distributions, their representations, and our features of interest. Section III describes the proposed shallow IQP model and its optimization. Section IV outlines the demonstration setup, including the datasets, training, hyperparameter optimization, and sampling. Section V reports the results. Finally, Sec. VI concludes and discusses future directions.

II. GRAPHS AND RANDOM GRAPH DISTRIBUTIONS

A graph distribution defines a probability distribution over the space of graphs satisfying a given criterion. In this work, we restrict attention to undirected, unweighted graphs to isolate structural dependencies while maintaining a compact binary representation. Such graphs connect nodes by undirected edges: if node i is linked to node j, then node j is equally linked to node i. Consequently, the adjacency matrix $\mathbf{A}$ is symmetric, $A_{ij} = A_{ji}$, and self-loops are excluded, i.e., $A_{ii} = 0\ \forall\ i$.

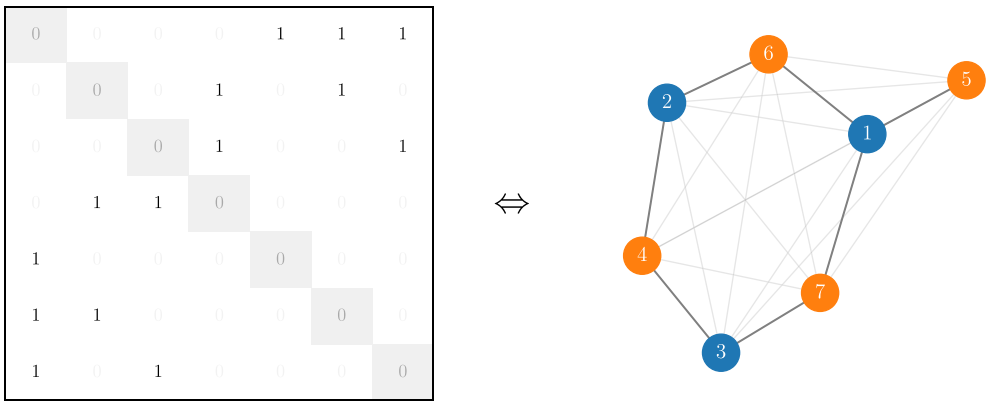
Because of this symmetry, specifying one triangular portion of $\mathbf{A}$ suffices. An undirected graph with M nodes is therefore fully characterized by

$$N = \frac{M(M-1)}{2}, \tag{1}$$

binary variables that encode the presence or absence of edges between distinct nodes (see Fig. 2). This encoding defines the effective dimensionality of the generative problem.

We focus on two families of random graph distributions. The first is the Erdős-Rényi (ER) model [27], which defines graphs in which each possible edge is included independently with probability $\rho \in [0, 1]$, known as the edge density. The second family comprises bipartite (BP) graphs, whose vertices can be partitioned into two disjoint sets such that edges exist only between and never within—equivalently, two-colorable graphs. BP distributions impose structural constraints that contrast with the edge-independent probabilistic structure of ER graphs, making them a useful testbed.

These two classes are not mutually exclusive: an ER sample may be bipartite, particularly at low edge densities, while any BP graph can be viewed as a constrained instance of the ER model. Throughout this work, BP distributions refer to ensembles of random bipartite graphs, whereas ER distributions denote ensembles of random graphs that may or may not exhibit bipartite structure.



$$000111 \otimes 01010 \otimes 1001 \otimes 000 \otimes 00 \otimes 0$$

FIG. 2. Top: Adjacency matrix representation of a seven-node undirected graph. Bottom: Corresponding bit string encoding constructed from the upper-triangular part of the adjacency matrix.

A. Bit string graph representation

To map the circuit output to a graph, we flatten the upper-triangular part of the adjacency matrix row by row into a binary string $\mathbf{z} = (z_1, \ldots, z_N)$, where $N = M(M-1)/2$ and M is the number of nodes, as illustrated in Fig. 2. In this representation, each qubit in the quantum circuit corresponds to a potential edge, and a measurement in the computational basis yields a binary string that uniquely defines a specific graph.

This encoding scales quadratically with the number of nodes. While more compact latent encodings or reparametrizations could reduce the required qubit count, we adopt this direct mapping as an interpretable baseline. It ensures an unambiguous correspondence between quantum measurement outcomes and graph structures, avoiding potential confounding factors introduced by latent representations.

B. Features of interest

Graph features represent different degrees of node interactions: some depend only on local edge probabilities, while others require knowledge of global relationships.

We formalize this notion as follows. Let $P(\mathbf{z})$ denote the target distribution over adjacency vectors, $\mathbf{z} = (z_1, \ldots, z_N) \in \{0,1\}^N$, and let $f : \{0,1\}^N \to \mathbb{R}$ be a feature of interest.

Definition 1 (k-bodied feature). A feature f is k-bodied if its expected value under P, $\mathbb{E}_P[f(\mathbf{z})]$, depends only on correlations among at most k edges.

Formally, for any subset of edge indices $S \subseteq \{1, \ldots, N\}$ with $|S| \leqslant k$, let $P_S(\mathbf{z}_S)$ denote the marginal distribution of the subvector $\mathbf{z}_S = (z_i : i \in S)$. Then f is k-bodied if the collection of all such marginals,

$$\{P_S(\mathbf{z}_S) : S \subseteq \{1, \ldots, N\},\ |S| \leqslant k\}, \tag{2}$$

is sufficient to uniquely determine $\mathbb{E}_P[f(\mathbf{z})]$.

Definition 2 (Global feature). A feature f is said to be global if it is not k-bodied for any fixed $k < N$.

These definitions measure the correlations required to learn to reproduce any given property in an idealized setting. We now describe the features used in this work, their theoretical locality, and practical considerations.

The simplest feature is density ρ, which measures the fraction of present edges relative to the maximum number of edges,

$$\rho(\mathbf{z}) = \frac{2}{M(M-1)} \sum_{i=1}^{N} z_i. \tag{3}$$

Density provides a coarse summary of the overall connectivity, showing whether a graph is sparse ($\rho \approx 0$) or dense ($\rho \approx 1$), but it does not reveal how edges are arranged among nodes. Since the expected density depends only on single-edge probabilities, it is a one-bodied feature. Consequently, it should be easy to learn and reproduce.

The next feature of interest is the degree distribution, which captures the number of edges incident on each node and provides a more detailed view of connectivity than density alone. For a node v_j, the degree is defined as

$$\deg(v_j) = \sum_{i \in \mathcal{E}_j} z_i, \tag{4}$$

where $\mathcal{E}_j$ is the set of edges incident on v_j. The degree distribution of a graph is then the histogram of $\deg(v_j)$ over all nodes $j \in \{1, \ldots, M\}$. For random graphs with independent edges, the degree of a node is the sum of independent Bernoulli variables, yielding a binomial distribution. Therefore, the degree distribution is also one-bodied. Although it has the same bodyness as the density, it should be harder to reproduce in practice, as it encodes additional information beyond the mean edge probability.

Another feature we consider is the bipartite property, which indicates whether a graph is two-colorable—i.e., has no odd-length cycles. The degree of correlations required to determine this property increases with graph size; for an M-node graph, one must capture all odd-cycle correlations up to length M, or $M-1$ if M is even. Consequently, bipartiteness is a global feature, as it depends on correlations that span the entire graph.

Among the considered features, bipartiteness is the most difficult to reproduce. It is a binary property—a graph is either bipartite or not—plus it is highly sensitive to perturbations: even a single bit flip can introduce an odd-length cycle, converting a bipartite graph into a nonbipartite one.

III. IQP CIRCUIT BORN MACHINES

A QCBM is a quantum generative model that represents a probability distribution in the measurement statistics of a pure quantum state [6]. Sampling from this distribution is achieved through projective measurements in the computational basis, producing single-shot samples.

In this work, we employ QCBMs constructed with IQP circuits. These represent a class of quantum computations that are restricted to commuting operations [9,28]. Despite this apparent limitation, they have emerged as a powerful tool for generative modeling. Sampling from most instances of IQP circuits is conjectured to be classically hard [29], as simulating them efficiently would imply a collapse of the polynomial hierarchy to its second level [10,30]. Yet, classical computers can efficiently estimate the expectation values of their commuting operators [31]. This feature enables their classical optimization when the cost function depends solely on those values [13]. Thus, it allows the implementation of a hybrid framework where classical resources are used for training and quantum resources for sampling.

Our objective is to train shallow IQP models to generate graphs according to a ground-truth distribution p. Specifically, we aim to learn a parametrized distribution $q_{\boldsymbol{\theta}}$ such that, for any feature of interest $f(\mathbf{z})$,

$$|\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\theta}}}[f(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim p}[f(\mathbf{z})]| < \varepsilon, \tag{5}$$

where ε defines the target accuracy.

A. Shallow IQP circuits

The foundation of our models is an N-qubit shallow IQP circuit, building on the general definition from Nakata *et al.* [28].

Definition 3. A shallow IQP circuit on N qubits is defined by the following sequential operations:

(1) Basis Transformation: An initial layer of Hadamard gates $H^{\otimes N}$ applied to all qubits.

(2) Diagonal Evolution: A parametrized unitary block $D_Z(\boldsymbol{\theta})$ diagonal in the computational basis. This block consists of a sequence of commuting rotation gates generated by Pauli-Z strings acting on subsets of qubits S:

$$D_Z(\boldsymbol{\theta}) = \prod_{S\in\mathcal{I}} \exp\left(-i\theta_S \prod_{j\in S} Z_j\right), \tag{6}$$

where $\mathcal{I}$ is a collection of subsets of $\{1,\ldots,N\}$ specifying the interactions. The collection $\mathcal{I}$ is chosen such that the total circuit depth is $\mathcal{O}(1)$, independent of the number of qubits N.

(3) Inverse Transformation: A final layer of Hadamard gates $H^{\otimes N}$ applied to all qubits.

(4) Measurement: A projective measurement in the computational basis.

The circuit diagram form is illustrated in Fig. 3.

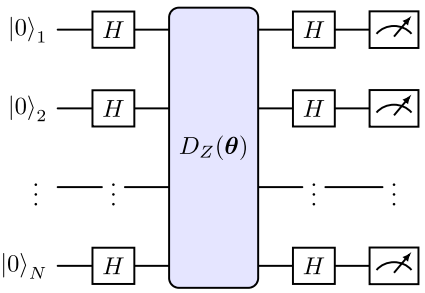


FIG. 3. Circuit diagram of a parametrized IQP circuit. The unitary $D_Z(\boldsymbol{\theta})$ contains all diagonal parametrized gates. For constant-depth $D_Z(\boldsymbol{\theta})$, the circuit realizes a shallow instance of the IQP model.

This sequence of operations implements a unitary $U(\boldsymbol{\theta}) = H^{\otimes N} D_Z(\boldsymbol{\theta}) H^{\otimes N}$, where $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^n$, that prepares the following quantum state:

$$|\Psi(\boldsymbol{\theta})\rangle = H^{\otimes N} D_Z(\boldsymbol{\theta}) H^{\otimes N} |0\rangle^{\otimes N}, \tag{7}$$

which encodes a probability distribution with respect to measurements in the computational basis.

Crucially, for any observable O_z that is a product of Pauli-Z operators, its expectation value,

$$\langle\Psi(\boldsymbol{\theta})|O_z|\Psi(\boldsymbol{\theta})\rangle, \tag{8}$$

can be computed efficiently on classical hardware. The precise complexity of this process is detailed in Proposition 2 of Recio-Armengol *et al.* [13].

All parametrized operations are contained within the diagonal gate block. Therefore, the ansatz is fully specified by this block alone. Crucially, it determines both the expressivity and the hardware efficiency of a parametrized circuit. A simple ansatz may be easy to implement but fail to capture high-bodied correlations, whereas a more complex one may be able to represent such correlations but at the cost of increased resource requirements—and an increased noise sensitivity. To balance this trade-off, we employ a shallow ansatz that uses few resources while still being conjectured to be classically intractable.

The classical simulability of IQP circuits depends strictly on the connectivity and type of gates employed, rather than circuit depth, as the commuting nature of the gates renders the temporal order of operations irrelevant. Circuits composed solely of single-qubit rotations are trivially simulable, as they generate no entanglement. Furthermore, circuits restricted to nearest-neighbor two-qubit gates on a planar grid can also be efficiently simulated, regardless of the number of layers. This is because their output probability distribution maps to the partition function of a two-dimensional (2D) Ising model with zero magnetic field, a problem exactly solvable via the Pfaffian method (free fermions) [32].

Notably, combining these two simulable designs in a planar 2D layout pushes the model out of the simulable regime [32]. Introducing single-qubit rotations corresponds to adding magnetic fields to the underlying Ising model. This modification breaks the free-fermion mapping and renders the calculation of the partition function #P-hard [32]. This combination—nearest-neighbor interactions plus single-qubit rotations—constitutes the design chosen for this work,

ensuring the ansatz remains shallow yet conjectured to be classically intractable.

This ansatz maps efficiently to the target quantum processor (QPU), IBM's Aachen, since both parametrized gates belong to the device's native gate set. In particular, RZ gates are implemented as virtual rotations with negligible latency and error [33]. Although Hadamard gates are not native, the compiler decomposes them into $\mathrm{RZ}(\pi/2)$ and SX gates, introducing only a minimal depth overhead.

Notably, we omit ancilla qubits, even though they are necessary to achieve universality [34]. While this choice limits expressive power, it may also enhance trainability—as expressivity and barren plateaus are closely linked [8,35].

B. Optimization via the maximum mean discrepancy

To train an IQP model on classical hardware, one must formulate the optimization task in terms of expectation values—both from the target data and the circuit itself.

The maximum mean discrepancy (MMD) can be formulated in such terms. It is an integral probability metric that compares two probability distributions, p and q, based on samples, $\boldsymbol{x}$ and $\boldsymbol{y}$, drawn from each in a reproducing kernel Hilbert space (RKHS) [36]. The squared MMD is defined as [37]

$$\begin{aligned}\mathrm{MMD}^2(p, q_\theta) = \mathbb{E}_{\boldsymbol{x}\sim p,\boldsymbol{y}\sim p}[k(\boldsymbol{x},\boldsymbol{y})] \\ - 2\mathbb{E}_{\boldsymbol{x}\sim p,\boldsymbol{y}\sim q_\theta}[k(\boldsymbol{x},\boldsymbol{y})] \\ + \mathbb{E}_{\boldsymbol{x}\sim q_\theta,\boldsymbol{y}\sim q_\theta}[k(\boldsymbol{x},\boldsymbol{y})],\end{aligned} \tag{9}$$

where $k(\boldsymbol{x},\boldsymbol{y})$ is a positive definite kernel. If k is characteristic, then $\mathrm{MMD}^2 = 0$ if and only if $p = q_\theta$ [36]. The three terms respectively measure intradata similarity, model-data cross-similarity, and intramodel similarity.

Central to this approach, we use the Gaussian kernel,

$$k_\sigma(\boldsymbol{x},\boldsymbol{y}) = \exp -\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2\sigma^2}, \tag{10}$$

for vectors $\boldsymbol{x},\boldsymbol{y} \in \{0,1\}^n$. Since each bit contributes either 0 or 1 to the sum, the squared Euclidean distance $\|\boldsymbol{x}-\boldsymbol{y}\|^2$ coincides with the Hamming distance, i.e., the number of positions at which the entries of the vectors differ. This equivalence allows the MMD to be expressed in terms of Pauli-Z observables [38],

$$\mathrm{MMD}^2(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{a}\sim\mathcal{P}_\sigma(\boldsymbol{a})}[(\langle Z_{\boldsymbol{a}}\rangle_p - \langle Z_{\boldsymbol{a}}\rangle_{q_\theta})^2], \tag{11}$$

where $\boldsymbol{a} \in \{0,1\}^n$ encodes a "mask" that corresponds to a Pauli-Z observable,

$$Z_{\boldsymbol{a}} = \prod_{i=1}^{n} Z_i^{a_i}. \tag{12}$$

The expectation value measures parity correlations among the selected bits,

$$\langle Z_{\boldsymbol{a}}\rangle_p = \mathbb{E}_{\boldsymbol{x}\sim p}[(-1)^{\boldsymbol{a}\cdot\boldsymbol{x}}], \tag{13}$$

where $(-1)^{\boldsymbol{a}\cdot\boldsymbol{x}} = +1$ for even parity and -1 for odd parity.

Crucially, the weighting distribution $\mathcal{P}_\sigma$ determines which correlations the loss emphasizes,

$$\mathcal{P}_\sigma(\boldsymbol{a}) = (1-p_\sigma)^{n-|\boldsymbol{a}|}(p_\sigma)^{|\boldsymbol{a}|}, \quad p_\sigma = \frac{1-e^{-1/2\sigma^2}}{2}. \tag{14}$$

Each bit appears independently with probability p_σ, so the active bit count $|\boldsymbol{a}|$ follows a binomial distribution with mean np_σ [38]. A derivation of the MMD as a quantum observable is shown in Appendix A.

This behavior has a profound impact on training. When σ is constant, $\sigma \in O(1)$, the MMD emphasizes high-order correlations, effectively making the loss global [38]. Such cost function is known to yield barren plateaus in unstructured circuits, potentially rendering the model untrainable [39]. Conversely, if σ scales with the number of qubits, $\sigma \in O(n)$, the MMD primarily captures low-order correlations, ensuring nonvanishing gradients and preserving trainability. However, it blinds the loss to the higher-order correlations.

This observation directly connects to the learnability of the target properties. Features such as density are expected to be learnable in a regime free of barren plateaus, whereas bipartiteness may require a constant σ and thus lie in an untrainable regime.

IV. DEMONSTRATION SETUP

This section describes the demonstration framework used to evaluate the proposed workflow (Fig. 1). We begin by outlining the generation of synthetic graph datasets employed for training and validation. The following subsections detail the model training procedure, hyperparameter optimization (HPO), and sampling strategy. We then introduce the quantitative metrics used to compare the generated and target graph distributions. Together, these components form a reproducible pipeline for assessing the model's performance.

All the code and data supporting this work are openly available in ZENODO [40].

A. Datasets

We generated synthetic datasets for the two graph families introduced in Sec. II: ER graphs, where each edge is included independently with probability ρ, and bipartite (BP) graphs, where a two-colorability constraint is imposed. We use the `NetworkX` library [41] to build them, ensuring that each sample represents a unique, nonisomorphic graph using the procedure described in Sec. II A.

For each graph family, we make sparse, medium, and dense instances with node counts $M \in \{8, 10, 14, 18\}$, corresponding to vectors of 28, 45, 91, and 153 bits, respectively. These categories refer to relative densities within a given node count and family; that is, for a fixed M and graph type, sparse instances have lower average density than medium ones, and so on, though absolute values vary across families and sizes. Lastly, the size of each dataset depends on the number of nodes, as smaller graphs admit fewer unique configurations. Table I summarizes the properties of each dataset, including graph type, average density, percentage of bipartite graphs, average bipartivity, and number of samples. A detailed description of these metrics is provided in Sec. IV E.

TABLE I. Summary of the datasets. $\bar{\rho}$ denotes the average density of the samples, BP (%) the fraction of bipartite graphs, $\bar{\beta}$ the average spectral bipartivity, and N the total number of samples.

		Dense				Medium				Sparse			
Nodes	Type	$\bar{\rho}$	BP.%	$\bar{\beta}$	N	$\bar{\rho}$	BP.%	$\bar{\beta}$	N	$\bar{\rho}$	BP.%	$\bar{\beta}$	N
8	BP	0.3110	100	1.0	261	0.2887	100	1.0	271	0.2256	100	1.0	133
	ER	0.7618	0.0	0.55	200	0.4420	0.5	0.79	200	0.2207	51.5	0.95	200
10	BP	0.3470	100	1.0	498	0.2307	100	1.0	500	0.1771	100	1.0	473
	ER	0.7884	0.0	0.51	500	0.4380	0.2	0.69	500	0.1919	33.8	0.94	500
14	BP	0.3727	100	1.0	995	0.2084	100	1.0	999	0.1042	100	1.0	995
	ER	0.8145	0.0	0.50	1000	0.4487	0.0	0.57	1000	0.1575	27.9	0.93	1000
18	BP	0.3697	100	1.0	995	0.1967	100	1.0	998	0.0747	100	1.0	992
	ER	0.8255	0.0	0.5	1000	0.4428	0.0	0.53	1000	0.1497	17.7	0.89	1000

B. Training

The training objective is to minimize the MMD between the target and generated distributions. We train all models on a single laptop using the `IQPopt` library [13], an open-source tool for classically optimizing parametrized IQP circuits in `JAX`. We configure it with the ADAM optimizer [42], the median heuristic for kernel bandwidth selection, and the data-driven initialization method of Recio-Armengol *et al.* [11], which sets two-qubit gate parameters proportional to the covariance of the corresponding bits in the training data. To refine these initial settings, we introduce scaling hyperparameters for both the kernel bandwidth and the initialization.

C. Hyperparameter optimization

To ensure a consistent basis for comparison across models, we perform HPO using the `Optuna` library [43] employing a tree-structured Parzen estimator (TPE) sampler [44]. We allocate a budget of 60 trials per dataset. This allocation is justifiable given that our search space has only three hyperparameters. Even under the conservative theoretical bounds of random search, 60 iterations provide a 95% probability ($1 - 0.95^{60} \approx 0.954$) of sampling a configuration within the top 5% of the search space volume [45]. Since TPE leverages Bayesian optimization to prioritize promising regions, it is expected to locate optimal configurations more efficiently than this baseline. Furthermore, each configuration is evaluated via repeated k-fold cross-validation, with three repetitions and five folds. Once the best configuration is found, it is trained three separate times, with small perturbations on the initialization, and the best-performing model is saved for sampling.

We optimize three key parameters: the learning rate, which controls the optimizer step size; the bandwidth multiplier, which scales the median-heuristic kernel width; and the initialization multiplier, which adjusts the amplitude of the initial weights. Other settings—such as the optimizer choice—showed negligible influence on performance in preliminary tests and were therefore kept fixed.

D. Sampling

After training, we generate samples by preparing the N-qubit state with the optimized parameters on the IBM Aachen quantum computer and measuring in the computational basis. Each measurement produces an N-bit string $\mathbf{z}$, which we map to a graph following the procedure in Sec. II A.

Importantly, we do not employ any error mitigation technique or classical postprocessing, which allows us to directly assess the raw performance of the quantum hardware.

E. Evaluation metrics

We evaluate model performance by comparing the generated and target graph distributions across the features of interest.

For the density, we measure the deviation in the expected value between the generated and target distributions,

$$\Delta\mathbb{E}[\rho] = \mathbb{E}_{\mathbf{z}\sim D_{\boldsymbol{\theta}}}[D(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim D}[D(\mathbf{z})]. \tag{15}$$

This test verifies whether the model reproduces the correct average edge density. Since this reflects the overall edge occupancy, a model that fails here will likely struggle to capture more complex features.

For the degree distribution, we employ a more comprehensive comparison. In an Erdős-Rényi graph, where edges are independent, the degree distribution follows a binomial law. Accordingly, we compute the expected degree distribution from the generated samples and compare it to the corresponding binomial, both qualitatively—via histograms—and quantitatively using the total variation distance (TVD)

$$\mathrm{TVD}(p, q_{\boldsymbol{\theta}}) = \frac{1}{2}\sum_{k} |p(k) - q_{\boldsymbol{\theta}}(k)|. \tag{16}$$

Here, $p(k)$ denotes the probability that a randomly selected node has degree k, and $q_{\boldsymbol{\theta}}(k)$ the corresponding probability under the model with parameters $\boldsymbol{\theta}$. This metric captures discrepancies in the full degree distribution, providing a sensitive test of how well local edge correlations are reproduced.

Lastly, we quantify the bipartite structure of the generated distributions using two complementary measures. First, bipartite accuracy, defined as the percentage of bipartite graphs generated, is determined via an exact two-coloring check. Given that this property is binary and can be very sensitive to noise—a single misplaced edge can introduce an odd-length cycle—we complement it with an expected bipartivity measure based on the spectral bipartivity index β introduced by

TABLE II. Performance validation for the eight-node datasets (28-qubit models) conducted via simulations using pennylane's `lightning.qubit` simulator.

		Dense		Medium		Sparse	
Nodes	Type	$\mathbb{E}[\rho]$ (Difference)	BP.% (Target %)	$\mathbb{E}[\rho]$ (Difference)	BP.% (Target %)	$\mathbb{E}[\rho]$ (Difference)	BP.% (Target %)
8	BP	0.312 (−0.001)	49.21 (100)	0.230 (0.058)	63.28 (100)	0.228 (−0.003)	77.19 (100)
	ER	0.759 (0.002)	0.0 (0.0)	0.437 (0.004)	3.9 (0.5)	0.221 (0.0005)	56.64 (51.5)

Estrada *et al.* [46],

$$\beta = \frac{\sum_i \cosh \lambda_i}{\sum_i e^{\lambda_i}}, \tag{17}$$

where λ_i are the eigenvalues of the adjacency matrix $\mathbf{A}$. This index compares the weighted contributions of even- and odd-length cycles: bipartite graphs yield $\beta = 1$, while complete graphs yield $\beta = 0$.

We compute these metrics on 512 samples per trained model. Simulations use PENNYLANE's `lightning.qubit` backend [47], while hardware demonstrations are executed on IBM's 156-qubit Aachen QPU, with each batch completing in 5.0 ± 0.5 s.

V. RESULTS

We organize the analysis in three stages. First, we validate the models at the 28-qubit scale through noiseless simulations. Next, we assess their noise robustness by comparing the results of identical models from simulations and quantum hardware. Finally, we investigate scaling behavior by deploying models ranging from 28 to 153 qubits.

For each dataset, we conduct a single HPO run and save the top three best-fitting models in terms of the cross-validation loss. Next, we sample to obtain the generation metrics. From these, we select the best-performing model for each dataset according to the following criteria: for BP datasets, the model that generates the highest proportion of bipartite graphs; and for ER datasets, the model that minimizes the total variation distance of the degree distribution.

A. Model validation

In this section, we validate our approach by simulating the sampling of the 28-qubit models in a noiseless regime. We evaluate performance based on the expected density—which corresponds to single-bodied correlations—and bipartite structure, which depends on all odd-order correlations. Together, these metrics probe the two ends of the expressivity spectrum. Successful modeling of the expected density implies the capacity to learn the lowest-order correlations. While the accuracy in capturing the bipartite structure examines whether the method is sufficiently expressive to represent complex, high-order dependencies.

All results are shown in Table II. For ER datasets, we observe negligible deviations in expected density across all connectivity profiles, with a maximum error of 4×10^{-3}. In the context of eight-node graphs (28 potential edges), this corresponds to an average overestimate of just $\sim$0.1 edges, indicating that the mean edge probability is captured with high precision. To further confirm this finding, we calculate the TVD of the degree distributions, which provides a broader assessment of the accuracy. We find that the models maintain high precision across all densities, yielding TVD values of approximately 0.0075 (dense), 0.0138 (medium), and 0.0085 (sparse). Collectively, these results confirm that shallow IQP models are capable of reproducing first-order correlations with high precision in a noiseless environment.

For the BP datasets, we first distinguish learned structure from random occurrence. We employ the Erdős-Rényi model as a null-hypothesis baseline: for each dataset, we generate 10^6 random graphs matching the target density and measure the bipartite fraction. Performance exceeding this baseline indicates that the model can reproduce the bipartite structure beyond random chance.

As shown in Table III, models across all densities significantly outperform the null hypothesis. The BP-Sparse model attains the best combined performance, reaching a bipartite accuracy of 75.19%—an improvement of nearly 20 percentage points over the baseline—while maintaining an empirical density error of -3×10^{-3} (Table II). The BP-Medium and BP-Dense models demonstrate stronger performance on bipartite generation, achieving accuracies of 63.281% and 49.218%, respectively, but fall shorter on the empirical density error.

Collectively, these results show that in a noiseless 28-qubit regime, shallow IQP circuits can effectively capture both local and global correlations. The models reproduce the expected density with high precision and bipartite structure beyond the null-hypothesis baseline. For broader context, we provide a comparison with classical generative models in Appendix B.

B. Model noise resilience

Before scaling to regimes beyond classical simulation, we assess the noise resilience of our models by benchmarking 28-qubit performance on IBM's Aachen QPU against noiseless simulations. Consistency between backends would signal robustness to hardware noise, whereas significant deviations highlight sensitivity. The quantum device characteristics and

TABLE III. Comparison of bipartite accuracies (%) from BP models with those from an Erdős-Rényi graph model of equal density. BP accuracies are computed from 512 generated samples, while the baseline is obtained from 10^6 random realizations.

Density	Baseline	Generated	Improvement
Dense	25.159	49.218	+24.059
Medium	32.163	63.281	+31.118
Sparse	55.822	75.195	+19.373

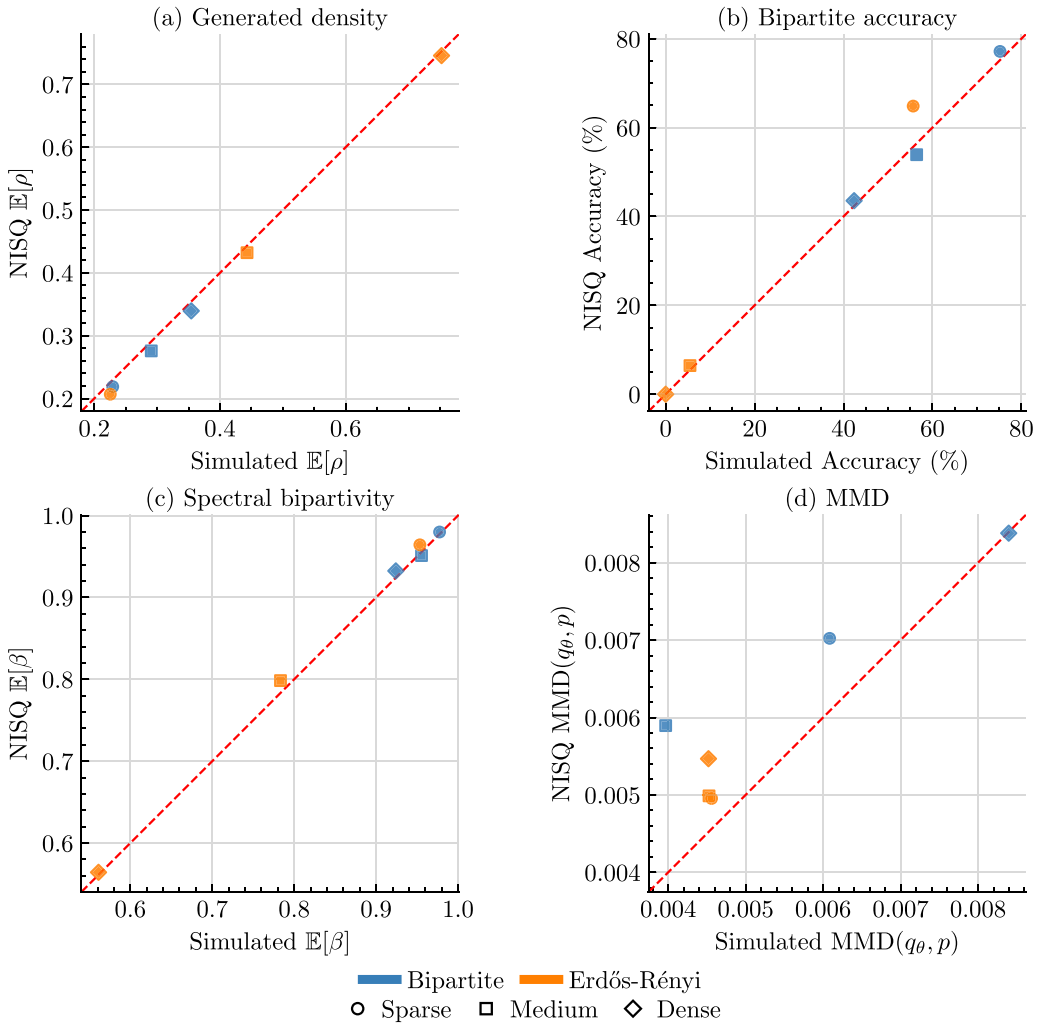


FIG. 4. Comparison of graph generation metrics between NISQ hardware and classical simulations for identical models. Panels show (a) generated density $\mathbb{E}[\rho]$, (b) bipartite accuracy, (c) spectral bipartivity $\mathbb{E}[\beta]$, and (d) MMD(q_θ, p). The red dashed line denotes perfect agreement. Deviations highlight discrepancies due to hardware noise and finite sampling.

parameters at the time of data acquisition are outlined in Appendix C.

Figure 4(a) illustrates a near-perfect alignment between NISQ-measured densities and simulated values, confirming that single-bodied statistics are minimally affected by hardware noise at this scale. To further assess the impact of noise on such features, we compare the TVD of the degree distributions. While noiseless simulations yielded an average TVD of $\sim$0.01, the quantum hardware achieves an average of $\sim$0.0307 (see Fig. 5). This increase is expected due to hardware noise, yet the value remains low enough to confirm that local structures are robustly reproduced.

Remarkably, global features also exhibit notable resilience. Despite noise-induced errors, the bipartite accuracy on NISQ hardware [Fig. 4(b)] closely tracks the noiseless ideal, with data points clustering tightly along the diagonal. This suggests that the learned bipartite structure is sufficiently robust for the noise profile of the quantum hardware at the 28-qubit scale. Similarly, the expected spectral bipartivity [Fig. 4(c)] remains in close agreement across backends. Finally, the MMD comparison [Fig. 4(d)] corroborates this high precision; deviations between the results observed on simulations and quantum hardware are constrained to the order of 10^{-3}.

Together, these observations suggest that while hardware noise impacts performance, it is not decisive at this scale. Local, single-body statistics exhibit the highest robustness, whereas global properties are comparatively more susceptible. However, contrary to the expectation that fragile global features would rapidly decay, we find that even strict bipartiteness is preserved with remarkable consistency at this scale.

C. Model scaling

Building on the validation and noise-resilience analysis, we now investigate the scaling behavior of shallow IQP models as they approach the limits of available quantum hardware. While the inherent imperfections of the NISQ regime constrain definitive conclusions regarding expressivity, these results establish an empirical baseline for generative performance on state-of-the-art devices.

We analyze the local connectivity of the ER models via the degree distribution and its TVD from the theoretical binomial (Fig. 5). The 28- and 45-qubit configurations reproduce the

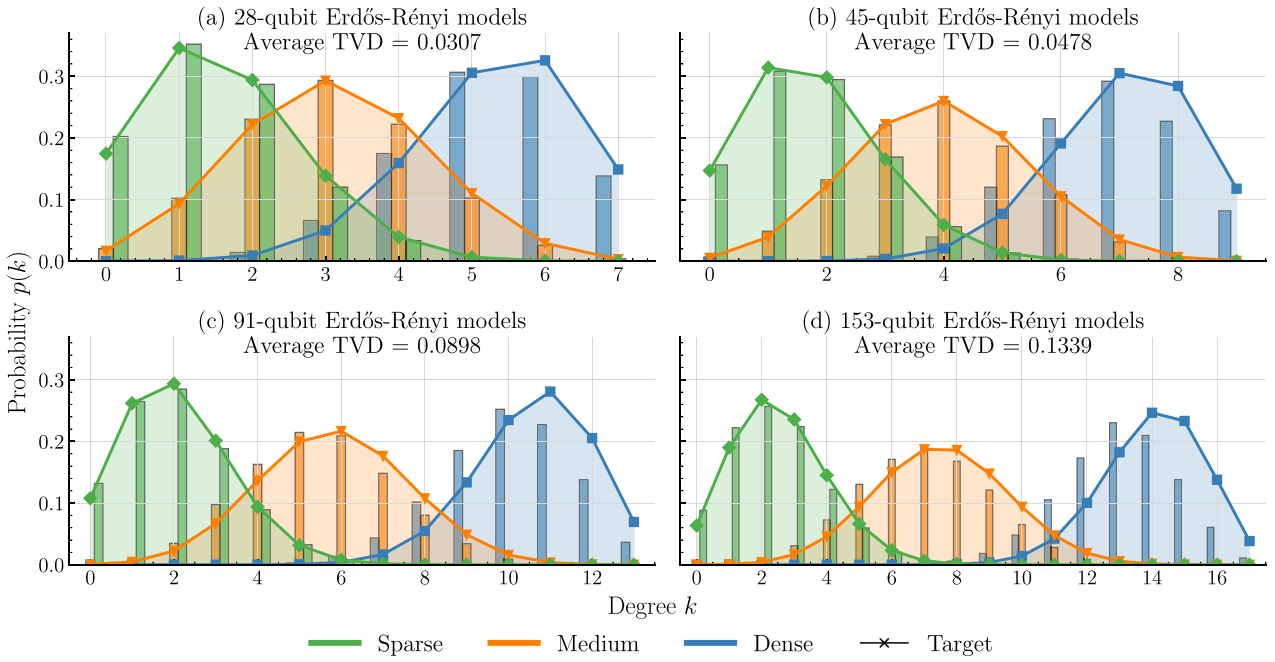


FIG. 5. Degree distributions obtained from models from 28 to 153 qubits trained on Erdős-Rényi datasets and executed on NISQ hardware. Bars indicate empirical node-degree frequencies, while solid lines denote theoretical binomial targets. The TVD measures the deviation between generated and target distributions.

target statistics with high fidelity, yielding average TVDs of 0.0307 and 0.0478, respectively. However, deviation increases with system size: average TVDs rise to 0.0898 at 91 qubits and 0.1339 at 153 qubits, representing a $\sim 3\times$ increase relative to the smallest instances. Given that the complexity of reproducing the degree distribution is independent of system size, we attribute the degradation primarily to noise. Despite this quantitative decline, qualitative performance remains consistent; as the histograms demonstrate, the generated distributions retain the characteristic binomial shape across all densities.

We employ the MMD as a global accuracy metric to track the divergence between learned and target distributions as system size increases (Fig. 6). The results reveal a clear hierarchy of learnability dictated by structural complexity. The Erdős-Rényi families (orange) prove to be the easiest to learn, maintaining consistently low MMD values (below 10^{-2}) across all scales. In contrast, the bipartite distributions (blue) exhibit higher divergence, particularly in the dense regime.

Notably, the MMD does not exhibit a monotonic increase with system size for BP models, as we would expect from the increased complexity at larger scales. For instance, the BP-Sparse configuration shows improving MMD values at larger scales, even as global features like strict bipartiteness degrade. This counterintuitive behavior underscores a critical limitation of using the MMD as a standalone metric in this context: the statistical correlations that dominate the loss landscape do not necessarily align with the defining structural features of the target distribution. Consequently, while a low MMD is a necessary condition for convergence, it is insufficient for validation; a rigorous assessment requires inspecting specific structural features of the generated data to fully characterize the model's fidelity.

For the BP models, we evaluate the learning of high-order correlations via the bipartite generation accuracy and spectral bipartivity (Fig. 7). Both metrics exhibit a similar trend: performance degrades significantly as system size and density increase, reflecting the difficulty of maintaining global constraints at larger scales where noise is more pronounced. Strict bipartite accuracy drops to zero for the 91- and 153-qubit models shortly after the baseline, whereas smaller models consistently maintain higher performance. Notably, spectral bipartivity follows a similar decay but maintains a score exceeding the random baseline even when strict bipar-

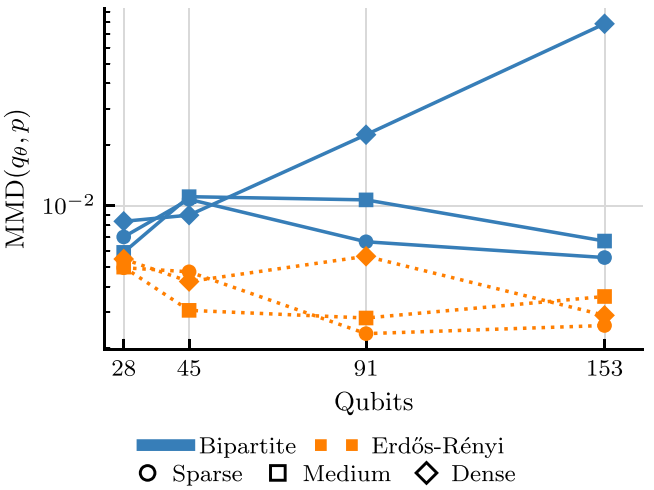


FIG. 6. Scaling of the MMD across all models. MMD values are computed between samples produced on NISQ hardware and the target dataset.

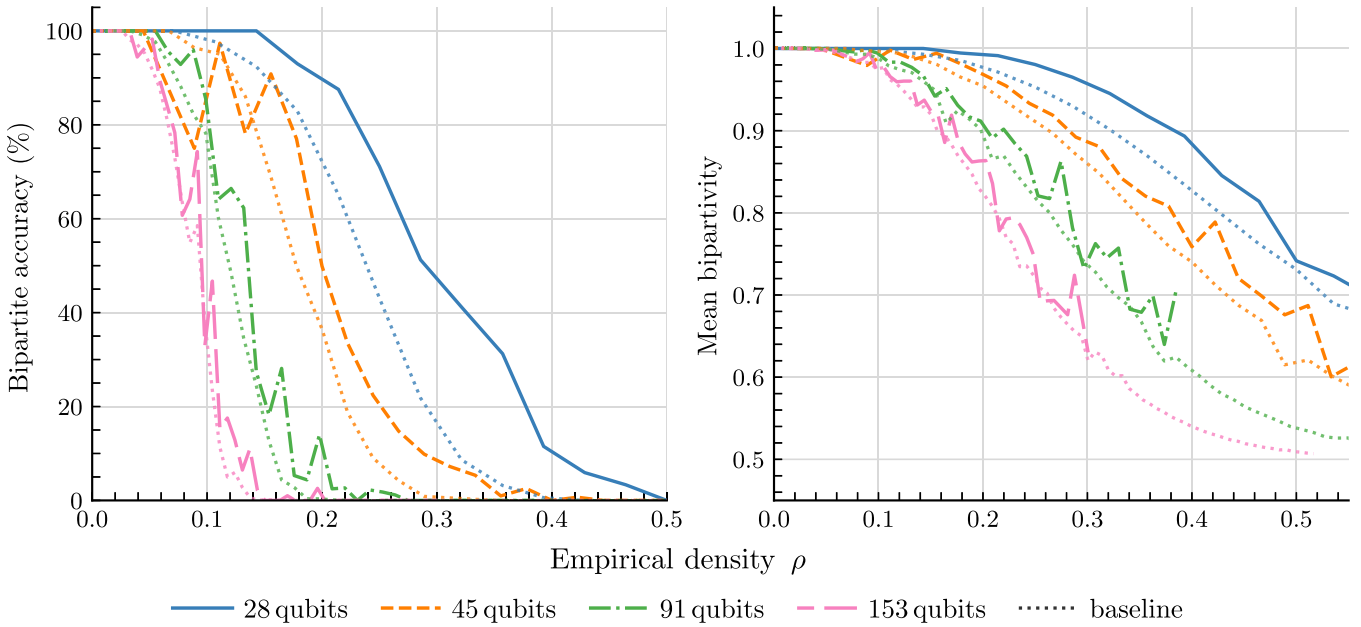


FIG. 7. Scaling behavior of bipartite graph generation as a function of empirical density ρ for models trained on bipartite datasets. Curves correspond to different qubit counts, with the dotted line showing the null-hypothesis baseline at each density.

tite accuracy is zero (e.g., $\rho \approx 0.2$–0.4 at 91 qubits). This indicates that partial bipartite features persist even if they are insufficient to satisfy the exact global constraint. Furthermore, the 91- and 153-qubit models fail to generate graphs beyond a specific connectivity threshold. This behavior suggests an optimization trade-off at large scales, unable to simultaneously reproduce the target density and enforce bipartiteness; the models appear to prioritize lower-density configurations where the constraint is less restrictive, effectively failing to populate the denser regions of the target distribution.

Collectively, these results demonstrate that our approach maintains high fidelity for local features, such as degree distributions, even at the 153-qubit scale. While global properties dependent on high-order correlations degrade with system size, the analysis of spectral bipartivity reveals that structural learning persists beyond the limits of strict bipartite accuracy. Ultimately, the models exhibit a clear hierarchy of learnability: unconstrained and local structures remain robust, whereas the combination of global features with strict constraints represents the current frontier for our approach on NISQ hardware.

VI. CONCLUSION

In this work, we have introduced a method for quantum generative modeling on NISQ devices that combines the theoretical advantages of IQP circuits with a practical implementation strategy. By exploiting the classical trainability of the IQP ansatz, our approach bypasses the sampling overhead and noise constraints during training, enabling the optimization of large-scale models. Furthermore, we minimize the resources required for implementation while retaining classical hardness through an ansatz that maps to a 2D Ising model with a magnetic field.

We validated this approach through a formalized graph generation setup, which allowed us to probe the relationship between feature locality and model accuracy on simulations and real quantum hardware, ranging from 28 to 153 qubits. The proposed method demonstrated exceptional robustness in capturing local features, accurately reproducing the degree distribution of target graphs up to the full 153-qubit scale without error mitigation. In contrast, global structural constraints such as bipartiteness were successfully learned and reproduced up to 45 qubits—outperforming random baselines—but degraded at larger scales due to hardware noise.

These results position shallow IQP circuits as a viable, scalable solution for NISQ generative modeling. The ability to learn and generate distributions defined by local and intermediate-range correlations at over 100 qubits suggests practical significance in domains where such features dominate, as our method is not limited to graph generation.

ACKNOWLEDGMENTS

O.B. and M.A. are fellows of Eurecat's "Vicente López" Ph.D. grant program. This piece of research was carried out with the partial support of the following granted projects: SGR Grant No. 2021 SGR 01559 from the Generalitat de Catalunya, Departament de Recerca i Universitats; QML-CV Grant No. SDC007/25/000001 from the Generalitat de Catalunya, Departament d'Empresa i Treball; GRAIL Grant No. PID2021-126808OB-I00 and SUKIDI Grant No. PID2024-157778OB-I00 from the Spanish Ministry of Science and Innovation, the European Union, and the Agencia Estatal de Investigación; and Cátedra ENIA UAB-Cruïlla Grant No. TSI-100929-2023-2 from the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union NextGenerationEU.

DATA AVAILABILITY

The data that support the findings of this article are openly available [40].

APPENDIX A: DERIVATION OF THE MAXIMUM MEAN DISCREPANCY AS A QUANTUM OBSERVABLE

This Appendix details the derivation of the squared MMD as a mixture of Pauli-Z expectation values. This formulation, first introduced in Rudolph *et al.* [38], allows the classical training of instantaneous quantum polynomial (IQP) circuits as implicit generative models.

The squared MMD is an integral probability metric that quantifies the distance between a target distribution p and a model distribution q_θ. It is defined via a kernel $k(\boldsymbol{x}, \boldsymbol{y})$ as

$$\begin{aligned}\mathrm{MMD}^2(p, q_\theta) = \mathbb{E}_{\boldsymbol{x}\sim p,\boldsymbol{y}\sim p}[k(\boldsymbol{x}, \boldsymbol{y})] \\ - 2\mathbb{E}_{\boldsymbol{x}\sim p,\boldsymbol{y}\sim q_\theta}[k(\boldsymbol{x}, \boldsymbol{y})] \\ + \mathbb{E}_{\boldsymbol{x}\sim q_\theta,\boldsymbol{y}\sim q_\theta}[k(\boldsymbol{x}, \boldsymbol{y})].\end{aligned} \tag{A1}$$

Crucial to this derivation, we employ the Gaussian kernel,

$$k_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \exp -\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{2\sigma^2}. \tag{A2}$$

Our goal is to rewrite the Euclidean distance $||\boldsymbol{x} - \boldsymbol{y}||^2$ with quantum operators. First, since $\boldsymbol{x}$ and $\boldsymbol{y}$ are binary strings, the expression becomes the Hamming distance,

$$||\boldsymbol{x} - \boldsymbol{y}||^2 = \sum_{i=1}^{n}(x_i - y_i)^2. \tag{A3}$$

Next, through the properties of the exponential, the kernel becomes

$$k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{n} \exp -\frac{(x_i - y_i)^2}{2\sigma^2}, \tag{A4}$$

since $e^{a+b} = e^a e^b$. Then we can write the application of the kernel on a single bit as

$$k_i(x_i, y_i) = \exp -\frac{(x_i - y_i)^2}{2\sigma^2}, \tag{A5}$$

which is 1 whenever $x_i = y_i$ and $e^{-1/2\sigma^2}$ otherwise. We seek an operator that satisfies those same conditions. In Rudolph *et al.* [38], they show that

$$k_i(x_i, y_i) = \exp -\frac{(x_i - y_i)^2}{2\sigma^2} \equiv (1 - p_\sigma) + p_\sigma(-1)^{x_i}(-1)^{y_i}, \tag{A6}$$

where p_σ is a scalar parameter that we can obtain by enforcing the previous conditions. For $x_i = y_i$ we trivially obtain $(1 - p_\sigma) + p_\sigma = 1$. For $x_i \neq y_i$ we obtain $(1 - p_\sigma) - p_\sigma = e^{-1/2\sigma^2}$, which yields

$$p_\sigma = \frac{1 - e^{-\frac{1}{2\sigma^2}}}{2}. \tag{A7}$$

We observe that the term $(-1)^{x_i}(-1)^{y_i}$ corresponds to the eigenvalue of the Pauli-Z tensor product $Z_i \otimes Z_i$ acting on the computational basis states $|x_i\rangle \otimes |y_i\rangle$. Consequently, the single-bit kernel is equivalent to the expectation value of the quantum operator

$$O_i = (1 - p_\sigma)\mathbb{I} \otimes \mathbb{I} + p_\sigma(Z_i \otimes Z_i). \tag{A8}$$

The full kernel operator is then the tensor product of these single-qubit operators across all n qubits,

$$\hat{K} = \bigotimes_{i=1}^{n}[(1 - p_\sigma)\mathbb{I} \otimes \mathbb{I} + p_\sigma(Z_i \otimes Z_i)]. \tag{A9}$$

Expanding this tensor product yields a sum over all possible Pauli-Z strings. We define $Z_{\boldsymbol{a}} = \bigotimes_{i=1}^{n} Z_i^{a_i}$ for a bit string $\boldsymbol{a} \in \{0, 1\}^n$, where the operator Z_i is applied if and only if $a_i = 1$. The weight of each term in the expansion is determined by the number of Z operators chosen (the Hamming weight $|\boldsymbol{a}|$). This results in the operator expansion

$$\hat{K} = \sum_{\boldsymbol{a}\in\{0,1\}^n} \mathcal{P}_\sigma(\boldsymbol{a})(Z_{\boldsymbol{a}} \otimes Z_{\boldsymbol{a}}), \tag{A10}$$

where $\mathcal{P}_\sigma(\boldsymbol{a})$ corresponds to the probability of sampling the string $\boldsymbol{a}$ from a product of Bernoulli distributions,

$$\mathcal{P}_\sigma(\boldsymbol{a}) = (1 - p_\sigma)^{n-|\boldsymbol{a}|} p_\sigma^{|\boldsymbol{a}|}. \tag{A11}$$

Returning to the classical domain, the kernel evaluation is given by the expectation of this operator with respect to the basis states $|\boldsymbol{x}\rangle$ and $|\boldsymbol{y}\rangle$, $\langle \boldsymbol{x}, \boldsymbol{y}|\hat{K}|\boldsymbol{x}, \boldsymbol{y}\rangle$. We note that the eigenvalue of the operator $Z_{\boldsymbol{a}}$ on a computational basis state is the product of the individual bitwise eigenvalues,

$$Z_{\boldsymbol{a}}|\boldsymbol{x}\rangle = (-1)^{\boldsymbol{x}\cdot\boldsymbol{a}}|\boldsymbol{x}\rangle. \tag{A12}$$

Substituting this eigenvalue into the operator expansion, the kernel function becomes

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{a}\in\{0,1\}^n} \mathcal{P}_\sigma(\boldsymbol{a})(-1)^{\boldsymbol{x}\cdot\boldsymbol{a}}(-1)^{\boldsymbol{y}\cdot\boldsymbol{a}}. \tag{A13}$$

Finally, we substitute this expansion into the definition of the MMD. By linearity of expectation, we can interchange the sum and the expectations. For the cross-term, this yields

$$\mathbb{E}_{\boldsymbol{x}\sim p,\boldsymbol{y}\sim q_\theta}[k(\boldsymbol{x}, \boldsymbol{y})] = \sum_{\boldsymbol{a}} \mathcal{P}_\sigma(\boldsymbol{a})\mathbb{E}_{\boldsymbol{x}\sim p}[(-1)^{\boldsymbol{x}\cdot\boldsymbol{a}}]\mathbb{E}_{\boldsymbol{y}\sim q_\theta}[(-1)^{\boldsymbol{y}\cdot\boldsymbol{a}}]. \tag{A14}$$

Identifying the inner expectations as the moments $\langle Z_{\boldsymbol{a}}\rangle_p$ and $\langle Z_{\boldsymbol{a}}\rangle_{q_\theta}$, respectively, and applying the same logic to the self-similarity terms, the MMD simplifies to a weighted sum of squared differences between expectations,

$$\mathrm{MMD}^2(p, q_\theta) = \mathbb{E}_{\boldsymbol{a}\sim\mathcal{P}_\sigma}[(\langle Z_{\boldsymbol{a}}\rangle_p - \langle Z_{\boldsymbol{a}}\rangle_{q_\theta})^2]. \tag{A15}$$

APPENDIX B: COMPARISON WITH CLASSICAL MODELS

The classical training of IQP circuits naturally raises questions regarding whether this hybrid approach compromises potential quantum advantages or if it simply reproduces results achievable by standard classical generative models. To address this, we compare the performance of the proposed IQP models against a GraphVAE [23] baseline for both bipartite and ER graph generation.

TABLE IV. Comparison of bipartite generation accuracy between shallow IQP circuits and GraphVAE models trained on eight-node bipartite datasets. Bold values denote the best performance.

Density	Model	Bip (%)
Sparse	GraphVAE	**86.5**
	IQP	75.19
Medium	GraphVAE	**86.1**
	IQP	63.3
Dense	GraphVAE	**80.9**
	IQP	49.2

TABLE V. Comparison of generated average edge probability between shallow IQP circuits and GraphVAE models trained on eight-node ER datasets. Bold values denote the best performance.

Density	Model	ρ	Target ρ
Sparse	GraphVAE	0.2242	0.2207
	IQP	**0.2204**	
Medium	GraphVAE	0.4308	0.4420
	IQP	**0.4419**	
Dense	GraphVAE	0.7543	0.7618
	IQP	**0.7617**	

1. GraphVAE architecture

The GraphVAE [23] is a deep generative model specifically designed to generate small graphs (between 9 and 38 nodes). It formulates the generation process within a variational autoencoder (VAE) [48] that outputs a probabilistic, fully connected graph of a predefined maximum size in a single shot, similar to the sampling of quantum circuits.

The model consists of a graph encoder $q_\phi(\mathbf{z}|G)$ and a deterministic decoder $p_\theta(G|\mathbf{z})$. While the original architecture supports graphs with variable node counts and features H, we simplify the model for our fixed-size, unweighted, undirected datasets. We omit node feature vectors and, since the node count is fixed to N, we remove the node-existence predictions (originally encoded on the diagonal) by enforcing a zero diagonal ($\tilde{A}_{ii} = 0$) in the output adjacency matrix $A \in \{0, 1\}^{N\times N}$.

The encoder utilizes a feed-forward network employing edge-conditioned graph convolutions (ECC) [23] to embed the input graph into a continuous latent representation. This process propagates node information through the graph layers, which is then aggregated into a fixed-size graph representation using a gated global pooling mechanism. This aggregated representation parametrizes the variational posterior $q_\phi(\mathbf{z}|G)$ (assumed to be a multivariate Gaussian), from which the latent vector $\mathbf{z} \in \mathbb{R}^{d_z}$ is sampled via the reparametrization trick.

The decoder is structured as a multilayer perceptron (MLP) that maps the latent vector $\mathbf{z}$ directly to a probabilistic adjacency matrix $\tilde{A} \in [0, 1]^{N\times N}$. The entries $\tilde{A}_{i,j}$ represent the probability of an edge existing between nodes i and j. The model assumes that these edge probabilities are Bernoulli variables that are independent conditioned on $\mathbf{z}$. Crucially, this independence assumption does not limit the model's ability to capture global structure (such as bipartiteness). The complex correlations required for such global features are encoded within the latent variable $\mathbf{z}$ itself; the decoder simply acts as a mapping from this globally correlated latent state to the individual edge probabilities.

2. Training objective and graph matching

A fundamental challenge in graph generation tackled by this model is the lack of a canonical node ordering; a generated adjacency matrix $\hat{A}$ may be isomorphic to the ground truth A under a node permutation P, yet appear distinct in a standard element-wise comparison. To evaluate the reconstruction likelihood $p_\theta(G|\mathbf{z})$, the model must align the generated probabilistic graph with the ground truth.

The GraphVAE optimizes an evidence lower bound (ELBO) objective, comprising a reconstruction loss and a Kullback-Leibler (KL) divergence term. While the original architecture employs approximate graph matching (specifically max-pooling matching) to scale to larger graphs, the small system size considered in this comparison ($N = 8$) allows for tractable exact matching. We define the reconstruction loss by explicitly iterating over the set of all possible permutation matrices $\mathcal{P}_N$ to find the optimal alignment,

$$\mathcal{L}_{\text{recon}} = \min_{P\in\mathcal{P}_N} \mathcal{L}_{\text{BCE}}(\hat{A}, PAP^T), \tag{B1}$$

where $\mathcal{L}_{\text{BCE}}$ denotes the binary cross-entropy loss. This ensures that the training objective is invariant to the arbitrary node ordering of the dataset.

3. Results

The GraphVAE models were subjected to HPO with a computational budget comparable to that of the shallow IQP circuits. We define this budget as the total number of optimization steps across all HPO trials (approximately 10^4 total epochs for both model classes). The GraphVAE search space swept over the latent space dimension $d_z \in [16, 64]$, learning rate $\eta \in [10^{-4}, 10^{-2}]$, and the KL annealing rate $\beta \in [10^{-3}, 1.0]$. Table IV presents the performance comparison for bipartite generation accuracy. While the GraphVAE

TABLE VI. Comparison of parameter count and TVD of the generated degree distributions across models trained on eight-node datasets. Bold values denote the best performance.

Graph	Density	Model	Params	TVD
Bipartite	Sparse	GraphVAE	205.2k	0.0811
		IQP	**56**	**0.0189**
	Medium	GraphVAE	200.1k	**0.0629**
		IQP	**56**	0.1428
	Dense	GraphVAE	199.4k	0.0789
		IQP	**56**	**0.0357**
ER	Sparse	GraphVAE	200.1k	0.0242
		IQP	**56**	**0.0085**
	Medium	GraphVAE	202.8k	0.0597
		IQP	**56**	**0.0137**
	Dense	GraphVAE	206.7k	0.0382
		IQP	**56**	**0.0074**

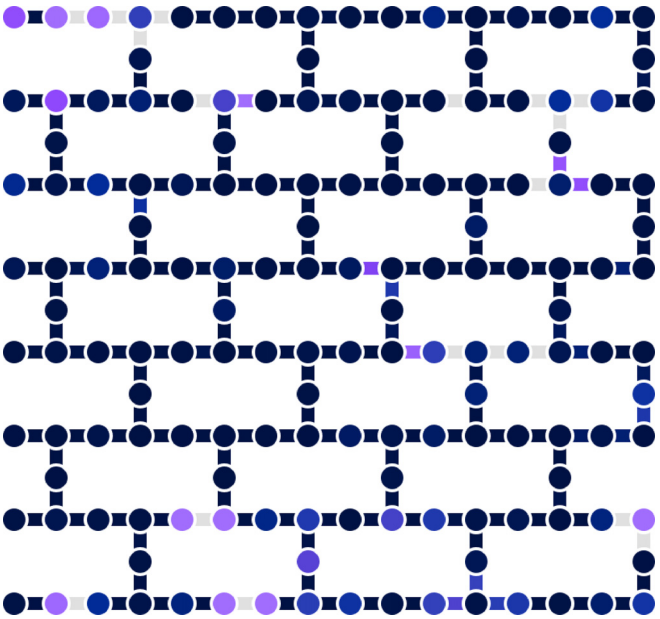


FIG. 8. Topology and connectivity graph of the `ibm_aachen` QPU at the time of the demonstration. Nodes represent physical qubits, with the color map indicating readout assignment error. Edges denote the RZZ error. In both cases, darker color signifies smaller errors.

models generally achieve higher accuracy in capturing the global bipartite structure—benefiting from their high parameter count and deep inductive biases—shallow IQP circuits remain competitive, particularly given the orders-of-magnitude difference in model complexity.

Interestingly, for ER datasets, the performance gap narrows significantly. As shown in Table V, shallow IQP circuits outperform the GraphVAE baseline in reproducing the target average edge probability ρ in all three cases.

To contextualize these results, Table VI compares the parameter counts across models against the performance in terms of the TVD of the generated degree distributions. The shallow IQP circuits achieve comparable or superior performance on these local features while utilizing approximately 3500× fewer parameters (56 vs ≈200 000).

We emphasize that this comparison is not intended to demonstrate "quantum advantage" in the strict computational sense. Rather, it serves as an empirical benchmark to contextualize our approach against a classical model designed for a similar task. A key takeaway from these results is the parameter efficiency. The quantum model achieves competitive performance with orders-of-magnitude fewer parameters. We do acknowledge that with larger datasets or an increased computational budget, the performance landscape could shift; however, for the tasks considered here, the shallow IQP approach offers a remarkably efficient alternative.

APPENDIX C: QUANTUM DEVICE CHARACTERISTICS AND DEMONSTRATION PARAMETERS

The quantum demonstrations reported in this work were executed on the `ibm_aachen` QPU, a superconducting quantum computing platform accessed via the IBM Quantum cloud service. In accordance with the guidelines for cloud quantum computing demonstrations, we provide the hardware characteristics at the time of execution.

1. Device layout and connectivity

The `ibm_aachen` device features a heavy-hexagonal lattice topology. The specific physical layout is illustrated in Fig. 8.

2. Coherence and gate fidelities

Device parameters fluctuate over time; the values presented here represent the state of the QPU during our data acquisition. Table VII summarizes the key performance metrics across the active qubits, extracted from the device calibration data provided by the backend. Specific qubit frequencies were not available in the retrieved calibration payload and are thus omitted.

3. Data analysis and mitigation

For this demonstration, the raw bit strings obtained from the `ibm_aachen` QPU were used directly to evaluate the circuits. No algorithmic error mitigation, readout error mitigation, or other postprocessing techniques were applied to the results. The data presented reflects the bare hardware performance under the calibrated conditions specified above.

TABLE VII. Summary of `ibm_aachen` device parameters during the demonstration. Values represent the range, median, and mean extracted from the device calibration dataset.

Parameter	Range	Median	Mean
Relaxation time, T_1 (μs)	2.46–499.57	219.10	220.12
Dephasing time, T_2 (μs)	3.35–485.63	182.58	189.25
Single-qubit gate error	1.0×10^{-4}–3.18×10^{-2}	2.0×10^{-4}	7.0×10^{-4}
Two-qubit (CZ) gate error	9.0×10^{-4}–1.91×10^{-1}	2.0×10^{-3}	1.05×10^{-2}
Readout assignment error	0.0017–0.5000	0.0108	0.0681
Prob meas0 prep1	0.0000–1.0000	0.0132	0.0836
Prob meas1 prep0	0.0000–1.0000	0.0045	0.0534
Single-qubit gate length (ns)	32	32	32
Readout length (ns)	2600	2600	2600

[1] C. Zoufal, Generative quantum machine learning, arXiv:2111.12738.

[2] R. Sweke, J.-P. Seifert, D. Hangleiter, and J. Eisert, On the quantum versus classical learnability of discrete distributions, Quantum **5**, 417 (2021).

[3] J. Zeng, Y. Wu, J.-G. Liu, L. Wang, and J. Hu, Learning and inference on generative adversarial quantum circuits, Phys. Rev. A **99**, 052306 (2019).

[4] B. Coyle, D. Mills, V. Danos, and E. Kashefi, The Born supremacy: Quantum advantage and training of an Ising Born machine, npj Quantum Inf. **6**, 60 (2020).

[5] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[6] J.-G. Liu and L. Wang, Differentiable learning of quantum circuit Born machines, Phys. Rev. A **98**, 062324 (2018).

[7] O. Kiss, M. Grossi, E. Kajomovitz, and S. Vallecorsa, Conditional Born machine for Monte Carlo event generation, Phys. Rev. A **106**, 022612 (2022).

[8] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp, E. R. Anschuetz, and Z. Holmes, Does provable absence of barren plateaus imply classical simulability? Nat. Commun. **16**, 7907 (2025).

[9] D. Shepherd and M. J. Bremner, Instantaneous quantum computation, Proc. R. Soc. A **465**, 1413 (2009).

[10] S. C. Marshall, S. Aaronson, and V. Dunjko, Improved separation between quantum and classical computers for sampling and functional tasks, arXiv:2410.20935.

[11] E. Recio-Armengol, S. Ahmed, and J. Bowles, Train on classical, deploy on quantum: Scaling generative quantum machine learning to a thousand qubits, arXiv:2503.02934.

[12] S. Kasture, O. Kyriienko, and V. E. Elfving, Protocols for classically training quantum generative models on probability distributions, Phys. Rev. A **108**, 042406 (2023).

[13] E. Recio-Armengol and J. Bowles, IQPopt: Fast optimization of instantaneous quantum polynomial circuits in JAX, arXiv:2501.04776.

[14] M. Newman, *Networks* (Oxford University, New York, 2018), Vol. 1.

[15] M. Drobyshevskiy and D. Turdakov, Random graph modeling: A survey of the concepts, ACM Comput. Surv. **52**, 1 (2020).

[16] A. Bonifati, I. Holubová, A. Prat-Pérez, and S. Sakr, Graph generators: State of the art and open challenges, ACM Comput. Surv. **53**, 1 (2021).

[17] P. Bongini, M. Bianchini, and F. Scarselli, Molecular generative graph neural networks for drug discovery, Neurocomputing **450**, 242 (2021).

[18] N. Yang, H. Wu, K. Zeng, Y. Li, S. Bao, and J. Yan, Molecule generation for drug design: A graph learning perspective, Fundam. Res. **6**, 40 (2026).

[19] D. Cordeiro, G. Mounié, S. Perarnau, D. Trystram, J.-M. Vincent, and F. Wagner, Random graph generation for scheduling simulations, in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques* (ICST, Malaga, Spain, 2010).

[20] A. Grover, A. Zweig, and S. Ermon, Graphite: Iterative generative modeling of graphs, arXiv:1803.10459.

[21] C. Tran, W.-Y. Shin, A. Spitz, and M. Gertz, DeepNC: Deep generative network completion, arXiv:1907.07381.

[22] D. Bacciu, A. Micheli, and M. Podda, Edge-based sequential graph generation with recurrent neural networks, Neurocomputing **416**, 177 (2020).

[23] M. Simonovsky and N. Komodakis, GraphVAE: Towards generation of small graphs using variational autoencoders, arXiv:1802.03480.

[24] D. Flam-Shepherd, T. Wu, and A. Aspuru-Guzik, Graph deconvolutional generation, arXiv:2002.07087.

[25] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, and S. Ermon, Permutation invariant graph generation via score-based generative modeling, arXiv:2003.00638.

[26] X. Guo and L. Zhao, A systematic survey on deep generative models for graph generation, arXiv:2007.06686.

[27] P. Erdős and A. Rényi, On random graphs. I., Publ. Math. Debrecen **6**, 290 (2022).

[28] Y. Nakata and M. Murao, Diagonal quantum circuits: Their computational power and applications, Eur. Phys. J. Plus **129**, 152 (2014).

[29] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Average-case complexity versus approximate simulation of commuting quantum computations, Phys. Rev. Lett. **117**, 080501 (2016).

[30] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy, Proc. R. Soc. A **467**, 459 (2011).

[31] M. V. den Nest, Simulating quantum computers with probabilistic methods, Quantum Info. Comput. **11**, 784 (2011).

[32] K. Fujii and T. Morimae, Quantum commuting circuits and complexity of Ising partition functions, New J. Phys. **19**, 033003 (2017).

[33] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient *Z*-gates for quantum computing, Phys. Rev. A **96**, 022330 (2017).

[34] A. Kurkin, K. Shen, S. Pielawa, H. Wang, and V. Dunjko, Note on the universality of parameterized IQP circuits with hidden units for generating probability distributions, arXiv:2504.05997.

[35] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, Barren plateaus in variational quantum computing, Nat. Rev. Phys. **7**, 174 (2025).

[36] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, A kernel two-sample test, J. Mach. Learn. Res. **13**, 723 (2012).

[37] D. J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton, Generative models and model criticism via optimized maximum mean discrepancy, arXiv:1611.04488.

[38] M. S. Rudolph, S. Lerch, S. Thanasilp, O. Kiss, O. Shaya, S. Vallecorsa, M. Grossi, and Z. Holmes, Trainability barriers and opportunities in quantum generative modeling, npj Quantum Inf. **10**, 116 (2024).

[39] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nat. Commun. **12**, 1791 (2021).

[40] O. Balló-Gimbernat, M. Arroyo-Sánchez, P. García Molina, A. Garriga, and F. Vilariño, Code for "Shallow instantaneous quantum polynomial-time circuits for generative modeling on noisy intermediate-scale quantum hardware", Zenodo (2026), https://doi.org/10.5281/zenodo.18983816.

[41] A. A. Hagberg, D. A. Schult, and P. J. Swart, Exploring network structure, dynamics, and function using networkx, in *Proceedings of the 7th Python in Science Conference*, edited by G. Varoquaux, T. Vaught, and J. Millman (Pasadena, CA, 2008), pp. 11–15.

[42] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980.

[43] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2019), pp. 2623–2631.

[44] S. Watanabe, Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance, arXiv:2304.11127.

[45] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. **13**, 281 (2012).

[46] E. Estrada and J. A. Rodríguez-Velázquez, Spectral measures of bipartivity in complex networks, Phys. Rev. E **72**, 046105 (2005).

[47] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. D. Matteo, A. Dusko, *et al.*, PennyLane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968.

[48] D. P. Kingma and M. Welling, An introduction to variational autoencoders, Found. Trends Mach. Learn. **12**, 307 (2019).