Minimizing classical resources in variational measurement-based quantum computation for generative modeling

Arunava Majumder,* Hendrik Poulsen Nautrup, and Hans J. Briegel
Department of Theoretical Physics, University of Innsbruck, Technikerstraße 21a, A-6020 Innsbruck, Austria

(Received 19 April 2026; revised 3 August 2026; accepted 20 October 2026; published 9 November 2026)

Measurement-based quantum computation (MBQC) is a framework for quantum information processing in which a computational task is carried out through one-qubit measurements on a highly entangled resource state. Due to the indeterminacy of the outcomes of a quantum measurement, the random outcomes of these operations, if not corrected, yield a variational quantum channel family. Traditionally, this randomness is corrected through classical processing in order to ensure deterministic unitary computations. Recently, variational measurement-based quantum computation (VMBQC) has been introduced to exploit this measurement-induced randomness to gain an advantage in generative modeling. A limitation of this approach is that the corresponding channel model has twice as many parameters compared with the unitary model, scaling as $N \times D$, where N is the number of logical qubits (width) and D is the depth of the VMBQC model. This can often make optimization more difficult and may lead to poorly trainable models. In this paper, we present a restricted VMBQC model that extends the unitary setting to a channel-based one using only a single additional trainable parameter. We show, both numerically and algebraically, that this minimal extension is sufficient to generate probability distributions that cannot be represented by the corresponding unitary model.

DOI: 10.1103/5hzz-v77n

I. INTRODUCTION

Quantum computers are expected to outperform classical computers for certain classes of problems [1–5]. In the standard circuit model, quantum computation proceeds deterministically through a sequence of unitary gates, followed by measurements performed only at the final stage to extract the output. Measurement-based (or one-way) quantum computation (MBQC) [6–8] offers a fundamentally different approach to perform computation. In this paradigm, one first prepares a highly entangled resource state, e.g., a cluster state [9], and then computation is carried out entirely via a sequence of one-qubit measurements on this state.

Since measurement outcomes in MBQC are, in general, intrinsically random, the computation naturally gives rise to probabilistic evolution corresponding to a mixed quantum channel. However, by employing fully adaptive measurement schemes that correct unintended outcomes at every step, this randomness can be compensated, effectively reducing the computation to a deterministic unitary evolution. Recent works have shown that the view of MBQC as a quantum channel can be exploited to achieve advantages over both classical approaches [10] and purely unitary quantum models [11]. In a related direction, Ref. [12] studies density quantum neural networks, randomized mixtures of parametrized unitaries that effectively implement a quantum channel, and demonstrates benefits compared with purely unitary models. Another recent work [13] shows that parametrized instantaneous quantum polynomial (IQP) circuits with hidden (traced-out) qubits, effectively forming a quantum channel on the visible qubits, are universal for generating arbitrary probability distributions on bit strings. While IQP circuits on N qubits alone are not universal, it was shown that IQP circuits on $2N + 1$ qubits with $N + 1$ qubits traced out become universal on the probability simplex of the remaining N output bits. This illustrates the effective advantage of channel models over purely unitary models, as a channel implemented on N qubits can realize distributions that a unitary model acting only on N qubits cannot. In particular, Ref. [11] introduces a quantum learning model based on MBQC embedded in the quantum circuit Born machines (QCBMs) [14,15] framework, called variational MBQC (VMBQC), which allows for controllable partial correction of measurement outcomes by introducing additional learning parameters that tune the likelihood of correcting the unintended measurement outcomes. This controlled partial adaptation of measurement outcomes gives rise to a trainable quantum channel model that can be shown to outperform its unitary counterpart (where all measurements are corrected), given the same (quantum) resources, both analytically and numerically. The authors demonstrate this advantage in a generative learning setting [16], where the goal is to learn and generate new samples from a target probability distribution given a limited number of training samples.

While the VMBQC-based channel model does not require additional quantum resources, it does require additional classical parameters, namely, the correction probabilities, which

*Contact author: arunava.majumder@uibk.ac.at

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.

control the partial adaptation of measurement outcomes. Given a unitary MBQC implemented on a cluster state of width N and depth D, the corresponding channel model introduces $N \times D$ additional trainable classical parameters, which gives twice as many as in the unitary model. While increasing the number of parameters may enhance the expressivity of a quantum learning model, it can significantly complicate the classical optimization process [17]. In addition, the randomness of measurement outcomes can be interpreted as a form of noise, which may add to a flat loss landscape [18]. This motivates the following question:

How many additional classical resources are sufficient to generate distributions with the VMBQC-based channel models that cannot be generated by the unitary models?

Here, classical resources refer to the number of additional trainable parameters introduced to the VMBQC model along with their gradient estimation overhead. In this paper, we show both analytically and numerically that introducing just a single additional trainable parameter is sufficient for the variational quantum channel model to outperform its purely unitary counterpart in generative learning tasks.

This paper is structured as follows: In Sec. II A, we introduce quantum generative modeling and the QCBM framework, while Sec. II B describes the MBQC framework and the VMBQC model. Section II C presents detailed descriptions of all training models used in this work. The results are presented in Sec. III. We first show algebraically that there exists a VMBQC channel model with a single additional trainable parameter whose output distribution cannot be represented by the corresponding unitary model. Motivated by this result, we then numerically compare the performance of several channel models, each with a single additional parameter, against the corresponding unitary model.

II. METHODS

This section reviews classical and quantum generative models (Sec. II A). Section II B summarizes variational measurement-based quantum computation from Ref. [11]. Section II C introduces the learning models used in our analysis, and Sec. II D defines the loss and corresponding gradients.

A. Quantum generative modeling

Generative learning refers to a class of unsupervised learning methods that aim to model the probability distribution underlying a given dataset (assumed to be the output of some probabilistic process). Generative models can be broadly categorized as either explicit, which provide a tractable mathematical expression for the distribution they generate, denoted by $q_{(\boldsymbol{\theta},\boldsymbol{p})}$, or implicit, in which case they describe only the stochastic process of sampling from the distribution, i.e., $x \sim q_{(\boldsymbol{\theta},\boldsymbol{p})}$ [19], with $(\boldsymbol{\theta}, \boldsymbol{p})$ being parameters of the learning model. In this work, we focus on implicit generative models. The training dataset for a generative model consists of a collection of M independent and identically distributed samples $D = \{x_1, x_2, \ldots, x_M\}$, where each $x_k \in \{0,1\}^N$ is a bit string of length N, with $k \in [M] := \{1, \ldots, M\}$, sampled from some unknown target distribution $Y(x)$. The goal is to design and train an implicit model such that its optimal empirical distribution $q_{(\boldsymbol{\theta}^*,\boldsymbol{p}^*)}(x)$ (defined by its optimal set of parameters $\{\boldsymbol{\theta}^*, \boldsymbol{p}^*\}$) approximates $Y(x)$ within some desired accuracy (see Fig. 1). Note that, in the generative machine-learning setting considered here, the samples in D are not supplied individually as inputs to the learning model. Rather, they are used to characterize the target distribution and evaluate the training loss. Therefore, no sample-wise classical data encoding is required. The learning model starts from a fixed quantum state.

Classical generative models typically rely on neural networks with trainable weights to learn a target probability distribution. In quantum generative modeling, for example, in quantum circuit Born machines, the neural network is replaced by a parametrized quantum circuit (PQC). Accordingly, the QCBM framework consists of several components: (i) a sufficiently expressive PQC composed of parametrized single-qubit gates and entangling two-qubit gates, (ii) a classical postprocessing stage in which a cost function and its corresponding gradients are evaluated using samples generated by the PQC, and (iii) a feedback loop that iteratively updates the PQC parameters until convergence is reached. The QCBM can thus be summarized as follows: First, a parametrized quantum state $|\psi(\boldsymbol{\theta}, \boldsymbol{p})\rangle$ is prepared by applying the PQC, with trainable parameters $\{\boldsymbol{\theta}, \boldsymbol{p}\}$, to a fixed initial state. Measurements performed at the end of the circuit, in a chosen basis, then generate samples according to the Born rule $q_{(\boldsymbol{\theta},\boldsymbol{p})}(x) = |\langle x|\psi(\boldsymbol{\theta}, \boldsymbol{p})\rangle|^2$. Each execution of the PQC yields a single sample (bit string x) drawn from the model distribution $q_{(\boldsymbol{\theta},\boldsymbol{p})}(x)$. By collecting sufficiently many such samples, one can perform loss optimization using an appropriate loss function (see Fig. 1). A key advantage of QCBMs is their ability to efficiently generate unbiased samples from the learned distribution, which is a fundamental requirement for generative learning. The VMBQC framework introduced in Ref. [11] provides a natural realization of PQCs by leveraging a circuit-model representation of MBQC in terms of so-called Clifford quantum cellular automata (CQCA); see Ref. [20]. The relevant construction is reviewed in Sec. II B.

B. MBQC with probabilistic byproduct correction

The discussion below provides a concise summary of the framework developed in Refs. [11,20], which serve as the foundation for the extension introduced in the present work.

In MBQC, one can perform universal quantum computation using single-qubit measurements on a highly entangled resource state [7,9]. The most common resource state is the so-called cluster state, which can be prepared by placing each qubit in the $|+\rangle$ state on a rectangular lattice and applying entangling CZ gates between neighbors [Fig. 2 (top)]. We always assume periodic boundary conditions on the cluster state, connecting the qubit $(1, j)$ and qubit (N, j) for all $j \in [1, D]$. Computation is carried out by measuring each qubit at site (i, j); $i \in [N]$, $j \in [1, D]$ in the cluster state in the XY plane [21], i.e., measuring the qubit in the basis $|\pm_{\theta_i^j}\rangle := 1/\sqrt{2}(|0\rangle \pm e^{-i\theta_i^j}|1\rangle)$. Here, N is the width of the cluster and D is its depth. The measurement angles θ_i^j then determine the effective unitary gate that is applied. The lattice has a total size of $N \times (D+1)$, where the final column (layer

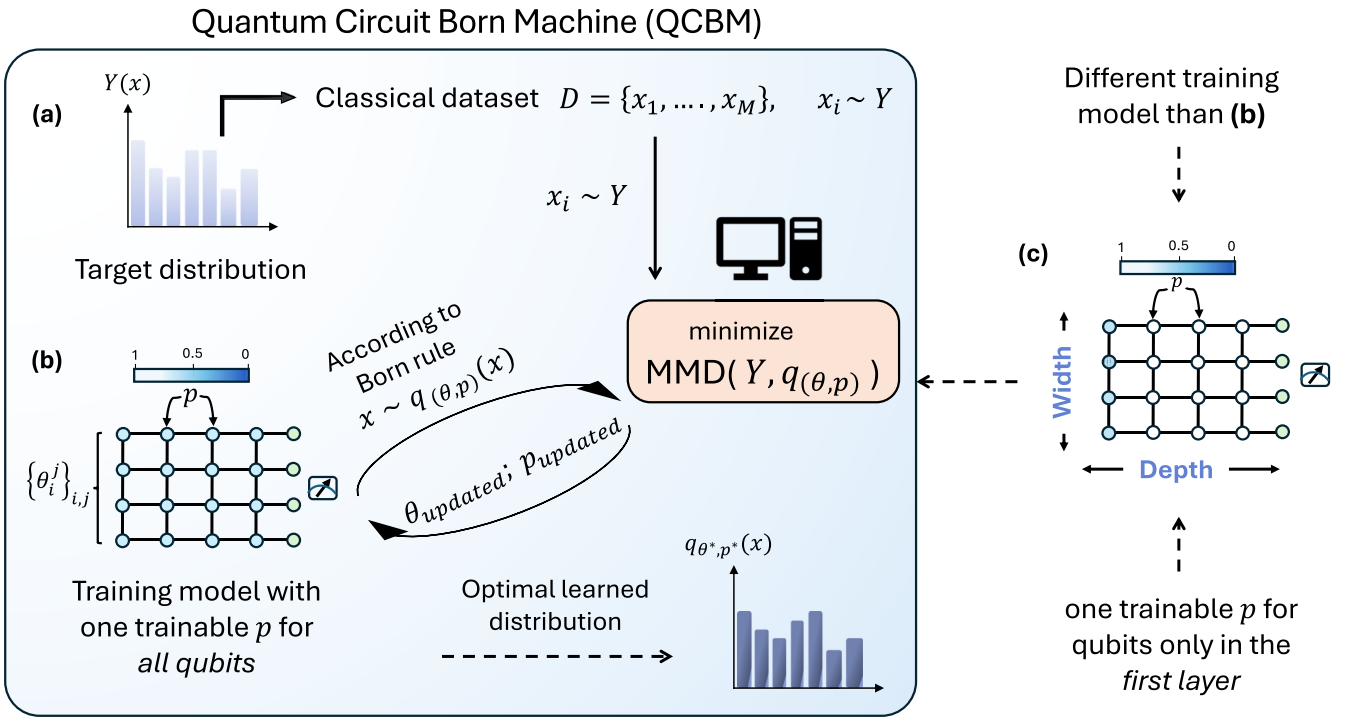


FIG. 1. Quantum circuit Born machines with VMBQC. (a) Target distribution $Y(x)$ (usually unknown), from which samples are given, $x \sim Y$. In this paper, the target distribution is generated from the VMBQC model itself with a randomly chosen set of parameters $\{\boldsymbol{\theta}, \boldsymbol{p}\}$ on the cluster state with some given width N and depth D. (b) The learning model with width N and depth D is parametrized by measurement angles $\boldsymbol{\theta} := \{\theta_i^j\}_{i,j}$ and a single correction probability p, that is, $\boldsymbol{p} := \{p_i^j = p\}_{i,j}$, for all partially adapted qubits (blue), by which we mean qubits, on which the measurement outcomes are adapted with probability p. When measured on the last layer, the model produces bit strings from the underlying distribution $[x \sim q_{(\boldsymbol{\theta},\boldsymbol{p})}]$ according to the Born rule. The maximum mean discrepancy (MMD) loss takes samples from both the learning model and the target distribution and measures the distance between the two distributions Y and $q_{(\boldsymbol{\theta},\boldsymbol{p})}$ and updates the parameters using the parameter shift rule on a classical computer. The model receives the updated parameters, and this loop continues until the MMD converges to some minimum value. Once converged, the optimal parameters $\boldsymbol{\theta}^*$ and $\boldsymbol{p}^*$ generate the optimal learned distribution. The entire learning process can also be implemented using a different learning model (c) where we allow partially uncorrected qubits ($p_i^j < 1$) only on the first layer of the cluster state (blue), with the same $p_i^j = p < 1$ for all of them, and the training procedure remains identical. All other qubits, in white, have $p_i^j = 1$, i.e., measurements on those qubits are always corrected. All the models in this study have periodic boundary conditions, i.e., the first and the last qubits in the cluster state are connected. The qubits in green at the end of the cluster are the output qubits measured in the computational basis. The samples $D = \{x_1, \ldots, x_M\}$ are used only to evaluate the loss, while the VMBQC model starts from a fixed quantum state.

with qubits in green in Fig. 2) serves as the output layer and is typically measured in the computational Z basis.

When measurements are performed in the aforementioned basis on the cluster state, the postmeasurement state is projected, with equal probability, onto either $|+_\theta\rangle\langle+_\theta|$ or $|-_\theta\rangle\langle-_\theta|$. The $+1$ measurement outcome directly implements the intended computation, whereas the -1 outcome realizes the same computation up to an additional Pauli-Z operator, since $|-_\theta\rangle\langle-_\theta| = Z\,|+_\theta\rangle\langle+_\theta|\,Z$. These Z operators are referred to as byproduct operators and can be compensated through classical processing and adaptive measurement updates [11].

MBQC on a cluster state with XY plane measurements is universal for quantum computation (Ref. [21]) and admits a well-defined circuit model representation, as established in Refs. [20,22–24]. As illustrated in Fig. 2 (bottom), each XY plane measurement on qubit i in layer j corresponds to a single qubit Z-rotation, $R_z(\theta_i^j) = e^{-iZ_i\theta_i^j/2}$, in the circuit model. A measurement outcome $+1$ (white), equivalently $s_i^j = 0$, implements the intended Z rotation without any additional byproduct. A measurement outcome -1 (blue), equivalently $s_i^j = 1$, introduces a Pauli-Z_i byproduct immediately after the rotation. Both cases are compactly represented by $Z_i^{s_i^j}$ in Fig. 2, where $s_i^j \in \{0, 1\}$ denotes the binary measurement outcome. These operations are followed by a fixed Clifford layer T_c, consisting of nearest-neighbor CZ gates (with periodic boundary conditions) and Hadamard gates applied to all qubits (see Fig. 2). Further details can be found in Refs. [11,20].

The fixed Clifford layer associated with one column of the periodic cluster state is

$$T_c := H^{\otimes N}\left(\prod_{i=1}^{N-1} \mathrm{CZ}_{i,i+1}\right)\mathrm{CZ}_{N,1}, \tag{1}$$

along with the rotation convention

$$R_z\left(\theta_i^j\right) := \exp\left(-i\theta_i^j Z_i/2\right). \tag{2}$$

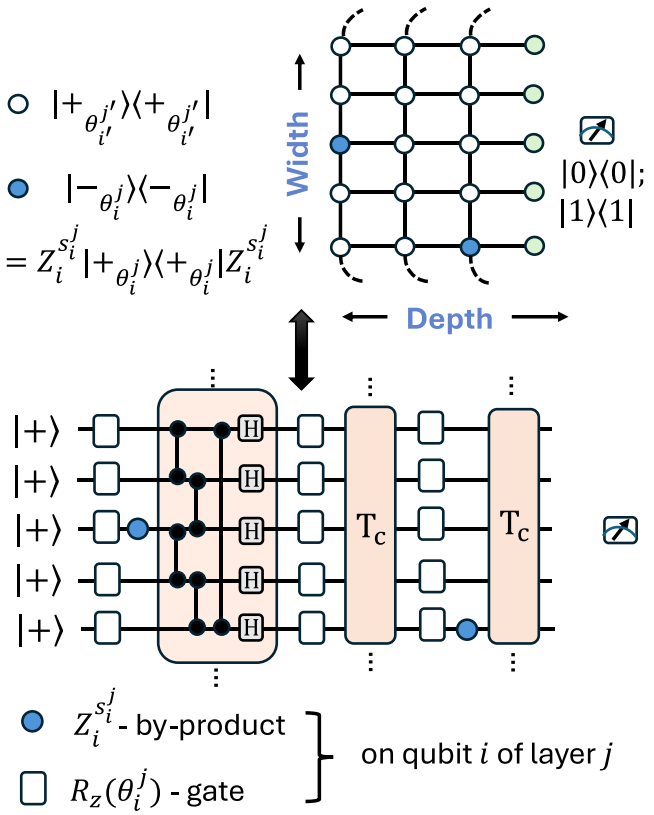


FIG. 2. C-QCA-based circuit translation of an MBQC on a cluster state. An MBQC on the cluster state (top) with width $N = 5$ and depth $D = 3$ can be understood in the circuit picture (bottom) where XY plane measurements become Z rotations in the circuit, interleaved with Clifford blocks T_c. The fourth column (top), with output qubits (green), is measured in the computational Z basis. An unintended measurement outcome, e.g., when the $(i, j) = (3, 1)$ qubit (blue) in the cluster state is measured in the $e^{-i\theta_3^1 Z/2}|-\rangle\langle -|e^{i\theta_3^1 Z/2}$ basis, yields the outcome -1, resulting in an equivalent byproduct operator $Z_i^{s_i^j}$ appearing in the circuit. The qubits in white yield a $+1$ outcome. All models have periodic boundary conditions, i.e., the first (top) and last (bottom) qubits are connected with a CZ gate.

As detailed in Ref. [20], the (universal) family of unitaries, parametrized by $\boldsymbol{\theta} = \{\theta_i^j\}_{i,j}$, that is implementable by the circuit in Fig. 2 (bottom, assuming no byproducts) is given by

$$U_c(\boldsymbol{\theta}) = \prod_{j=D,\ldots,1} \left(T_c \prod_{i=N,\ldots,1} \exp\left(-i\theta_i^j Z_i/2\right) \right), \tag{3}$$

which is referred to as the Clifford quantum cellular automata (CQCA) in Ref. [20]. Equation (3) can be directly put into correspondence with the circuit in Fig. 2 (without any Z byproduct).

In this way, MBQC on a cluster state naturally gives rise to a PQC ansatz for variational quantum algorithms (VQAs) [25], as formalized in Eq. (3) (see also Ref. [20]).

An expression similar to Eq. (3), but including byproducts [see Fig. 2 (bottom-blue)] is (see Ref. [11])

$$U_c(\boldsymbol{\theta}, \boldsymbol{s}) = \prod_{j=D,\ldots,1} \left(T_c \prod_{i=N,\ldots,1} \left(Z_i^{s_i^j}\right) \exp\left(-i\theta_i^j Z_i/2\right) \right), \tag{4}$$

where $\boldsymbol{s} \in \{0, 1\}^{N\times D}$ is an $N \times D$ matrix with binary elements $s_i^j \in \{0, 1\}$, representing the measurement outcomes at different sites ($i \in [N]$, $j \in [D]$). It controls the presence ($s_i^j = 1$) or absence ($s_i^j = 0$) of a byproduct operator on qubit i at layer j as $Z_i^{s_i^j}$ (see Fig. 2).

Besides using CQCA as the parametrized quantum circuit ansatz for the QCBM (Fig. 1), another key element in the model is the controlled use of these byproducts of U_c in Eq. (4). In the context of generative modeling, the authors of Ref. [11] showed that introducing controlled randomness in the VMBQC model can aid distribution learning. In this approach, instead of correcting byproducts every time during the computation, they assign trainable correction probabilities p_i^j for every qubit i at layer j to partially correct those byproducts. To describe the model with these additional trainable correction probabilities, they introduce additional implicit parameters $c_i^j \in \{0, 1\}$ for all qubits $(i, j) \in [N] \times [D]$ that decide whether to correct ($c_i^j = 1$) a specific byproduct at position (i, j) or not ($c_i^j = 0$). The c_i^j can then be sampled with probability p_i^j as

$$c_i^j = \begin{cases} 0 & \text{with probability } 1 - p_i^j \\ 1 & \text{with probability } p_i^j \end{cases} \tag{5}$$

where p_i^j is a trainable correction probability.

To incorporate the trainable correction probabilities into the model, the authors in Ref. [11] replace the measurement outcomes s_i^j with an adaptive correction scheme $\tilde{s}_i^j = (1 - c_i^j)s_i^j$ that depends on the controllable parameters c_i^j. When we do not correct a byproduct ($c_i^j = 0$, with probability $1 - p_i^j$), the probability of a -1 outcome ($|-\rangle$ state) is $p(s_i^j = 1) = \frac{1}{2}$. Therefore, from the law of total probability, we can write

$$\begin{aligned} p(\tilde{s}_i^j = 1) &= \tfrac{1}{2}(1 - p_i^j), \\ p(\tilde{s}_i^j = 0) &= \tfrac{1}{2}(1 + p_i^j). \end{aligned} \tag{6}$$

Given the set of trainable correction probabilities $\boldsymbol{p} = \{p_i^j\}_{i,j}$, and Eq. (4), variational MBQC gives rise to a family of quantum channels as follows:

$$\mathcal{E}_c(\boldsymbol{\theta}, \boldsymbol{p})[\rho] = \sum_{\tilde{\boldsymbol{s}}\in\{0,1\}^{N\times D}} p(\tilde{\boldsymbol{s}}) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}), \tag{7}$$

where $p(\tilde{\boldsymbol{s}}) = \prod_{i,j} p(\tilde{s}_i^j) = \prod_{i,j} \left(\frac{1+(-1)^{\tilde{s}_i^j} p_i^j}{2}\right)$, with $\tilde{\boldsymbol{s}}$ being an $N \times D$ binary matrix with entries $\tilde{s}_i^j \in \{0, 1\}$, and p_i^j is the trainable correction probability for the qubit at the (i, j) location.

In MBQC, measurement-induced byproducts can be propagated through the cluster state by repeatedly applying the stabilizer relations (see Fig. 3, top, with one byproduct $Z_3^{s_3^1}$),

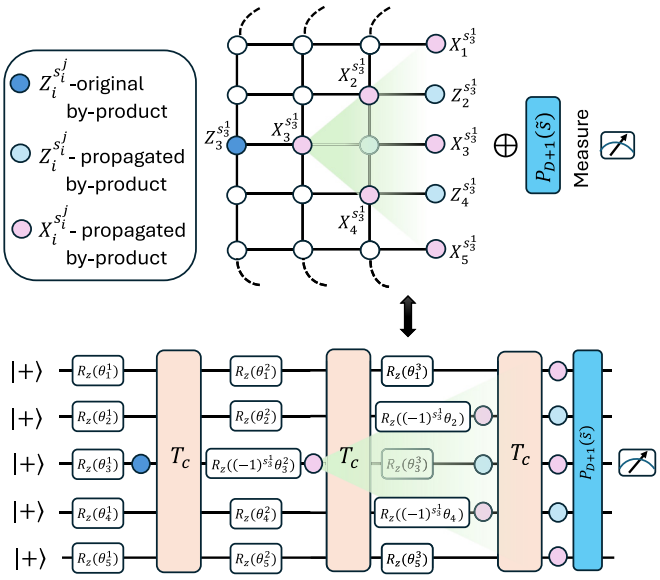


FIG. 3. Propagation of byproducts in $\tilde{\mathcal{E}}_c(\boldsymbol{\theta}, \boldsymbol{p})$ model. (Top) Consider a byproduct $Z_3^{s_3^1}$ on qubit 3 at position $(3, 1)$. Propagating this through the cluster state via repeated stabilizer relations generates $X_i^{s_3^1}$ byproducts on qubits i within the forward light-cone (green), which flips (if $s_3^1 = 1$) the measurement angles of the qubits. For instance, the angle θ_3^2 at position $(3, 2)$ is transformed to $(-1)^{s_3^1}\theta_3^2$, following $X_3^{s_3^1} R_z(\theta_3^2)\,|\pm\rangle\,\langle\pm|\,R_z^\dagger(\theta_3^2) X_3^{s_3^1} = R_z[(-1)^{s_3^1}\theta_3^2]\,|\pm\rangle\,\langle\pm|\,R_z^\dagger[(-1)^{s_3^1}\theta_3^2]$ up to a global phase. After propagating through all layers in the cluster, the byproducts accumulate at the final layer (before computational measurements), forming a Pauli string of X^s and Z^s byproducts that depend on outcomes $\boldsymbol{s}$. (Bottom) Equivalent circuit representation of byproduct propagation via repeated Clifford circuits T_c^{D-j+1} $(D = 3,\ j \in [D])$ for a byproduct $Z_i^{s_i^j}$, $i \in [N]$. In both representations, the accumulated byproducts at the end of the computation form a Pauli string and the additional $P_{D+1}(\tilde{\boldsymbol{s}})$ layer removes this Pauli string (Clifford part) from the computation and leaves the non-Clifford effect of the byproducts in the intermediate-angle flips.

which can be understood by Eq. (1) in Ref. [11]. In the equivalent circuit representation, this corresponds to commuting the Pauli-$Z_i^{s_i^j}$ byproducts to the end of the circuit through all the following Clifford circuits T_c^{D-j+1} (for a total depth of D), which induces sign flips of the intermediate rotation angles along the way (Fig. 3, bottom, with $D = 3$). Once the byproducts reach the end of the circuit, they combine into a single Pauli string (e.g., $\hat{P} = X_1^{s_3^1} \otimes Z_2^{s_3^1} \otimes X_3^{s_3^1} \otimes Z_4^{s_3^1} \otimes X_5^{s_3^1}$ in Fig. 3). The $P_{D+1}(\tilde{\boldsymbol{s}})$ layer in Fig. 3 (bottom) corresponds to this Pauli byproduct operator $\hat{P}$, as it depends on the measurement outcome $\tilde{\boldsymbol{s}}$, and is responsible for removing the accumulated Pauli string $\hat{P}$. In Ref. [11], this process is described in more detail, and it serves the purpose of removing the (trivial) Clifford part of the byproducts ($\hat{P}$) while retaining the (nontrivial) non-Clifford effect of the byproducts. This non-Clifford effect of the byproducts influences qubits within a specific forward light cone (green in Fig. 3) by inducing sign flips of the corresponding rotation angles [as $X^s R_z(\theta) = R_z[(-1)^s\theta]X$]. This results in unitary evolution:

$$\tilde{U}_c(\boldsymbol{\theta}, \boldsymbol{s}) = P_{D+1}(\tilde{\boldsymbol{s}}) \Bigg[\prod_{j=D,\ldots,1} \Bigg(T_c \prod_{i=N,\ldots,1} \left(Z_i^{s_i^j}\right) \times \exp\left(-i\theta_i^j Z_i/2\right) \Bigg) \Bigg]. \tag{8}$$

Similar to Eq. (7), combining the learnable probabilities $\boldsymbol{p} = \{p_i^j\}_{i,j}$ with the above unitary evolution defines a family of quantum channels:

$$\tilde{\mathcal{E}}_c(\boldsymbol{\theta}, \boldsymbol{p})[\rho] = \sum_{\tilde{\boldsymbol{s}} \in \{0,1\}^{N\times D}} p(\tilde{\boldsymbol{s}}) \tilde{U}_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho \tilde{U}_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}), \tag{9}$$

where $p(\tilde{\boldsymbol{s}}) = \prod_{i,j} p(\tilde{s}_i^j)$, and $p(\tilde{s}_i^j)$ is given by Eq. (6). For further details, see Refs. [11,20].

The discussion in this section is intended to summarize the MBQC-to-circuit correspondence and the role of measurement-induced byproducts needed for the present analysis. Detailed derivations of the circuit equivalence, byproduct propagation, and adaptive correction rules can be found in Refs. [11,20]. The results developed in the following sections build on this framework by restricting the correction structure and analyzing the resulting learning performance.

C. Learning models

In this section, we introduce several simpler variants of the VMBQC model [Eq. (7)]. Specifically, we consider models with only one correction probability $\boldsymbol{p} := \{p_i^j = p\}_{i,j}$ for different qubits (i, j) in the cluster state. In contrast, in the model from Ref. [11], each qubit is assigned a different individually adjustable correction probability. We present four different variants of this kind, depicted in Fig. 4.

For the model corresponding to Fig. 4(a), one can rewrite Eq. (7) as

$$\mathcal{E}_c^{(A)}(\boldsymbol{\theta}, \boldsymbol{p})[\rho] = \sum_{\tilde{\boldsymbol{s}} \in \{0,1\}^{N\times D}} \prod_{\substack{j\in[D]\\ i\in[N]}} \left(\frac{1 + (-1)^{\tilde{s}_i^j} p}{2} \right) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}). \tag{10}$$

In Eq. (10), all qubits at positions (i, j) for $i \in [N],\ j \in [D]$ have the same correction probability p. The quantity $\tilde{s}_i^j \in \{0, 1\}$, appearing in the exponent, is a binary-valued entry of the matrix $\tilde{\boldsymbol{s}} \in \{0, 1\}^{N\times D}$ for all $i \in [N],\ j \in [D]$, which specifies whether a byproduct is present ($\tilde{s}_i^j = 1$) at position (i, j) or absent ($\tilde{s}_i^j = 0$) at position (i, j).

To define the model in Fig. 4(b), we consider the matrix $\tilde{S}^{(B)} := \{(s_i^j)_{i=1,\ldots,N}^{j=1,\ldots,D} \in \{0, 1\}^{N\times D} \mid s_i^j = 0 \text{ if } j \neq 1\}$, which is a subset of $\{0, 1\}^{N\times D}$. Each element $\tilde{\boldsymbol{s}} \in \tilde{S}^{(B)}$ is an $N \times D$ matrix whose first N elements are binary numbers, referring only to the qubits (blue) in the first layer $j = 1$ of the model in Fig. 4(b). The rest of the elements of $\tilde{\boldsymbol{s}} \in \{0, 1\}^{N\times D}$ are zero, implying the absence of byproducts from other qubits outside the first layer, as those qubits $(i \in [1, N] := \{1, \ldots, N\},\ j \in [2, D] := \{2, \ldots, D\})$ are always corrected [white qubits in

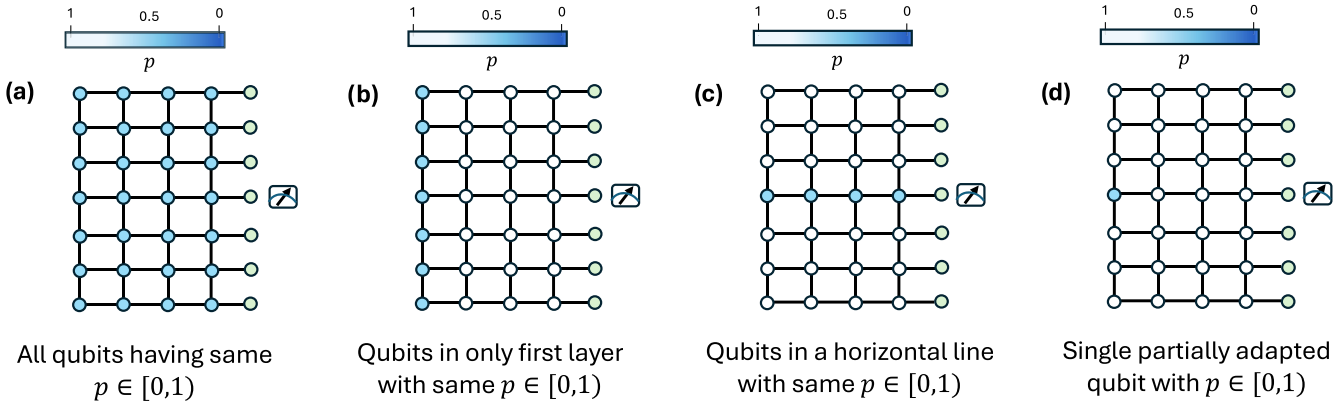


FIG. 4. VMBQC models with one correction probability. We consider four variants of the VMBQC model in a restricted setting, where the number of trainable correction probabilities is limited to a single parameter, i.e., $\boldsymbol{p} := \{p_i^j = p\}_{i,j}$. (a) Channel model $\mathcal{E}_c^{(A)}$ corresponding to Eq. (10) where measurement outcomes of all qubits (blue) are partially adapted and all are sharing the same correction probability $p < 1$. (b) Channel model $\mathcal{E}_c^{(B)}$ corresponding to Eq. (11) where partially corrected qubits are restricted only to the first layer (blue), each with the same correction probability $p < 1$, while all remaining qubits (white), when measured, are always fully corrected ($p = 1$). (c) Channel model $\mathcal{E}_c^{(C)}$ corresponding to Eq. (12) where exactly one partially adapted qubit appears in each layer (blue), repeated across all layers (chosen as the central qubit). (d) Channel model $\mathcal{E}_c^{(D)}$ corresponding to Eq. (13) has only a single qubit, depicted in blue, whose measurement outcome is partially adapted with $p < 1$. All the models considered above have periodic boundary conditions, and the qubits in green are the output qubits measured in the computational basis.

Fig. 4(b)] in this case:

$$\begin{aligned}
&\mathcal{E}_c^{(B)}(\boldsymbol{\theta}, \boldsymbol{p})[\rho] \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(B)}} \prod_{\substack{j\in[D]\\ i\in[N]}} p\big(\tilde{s}_i^j\big)\, U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(B)}} \prod_{i\in[N]} p\big(\tilde{s}_i^1\big) \prod_{\substack{j\in[D]\setminus\{1\}\\ i\in[N]}} \underbrace{p\big(\tilde{s}_i^j = 0\big)}_{=1} U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(B)}} \left(\prod_{i=1}^{N} p\big(\tilde{s}_i^1\big) \right) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}})\, \rho\, U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(B)}} \left(\prod_{i=1}^{N} \frac{1 + (-1)^{\tilde{s}_i^1} p}{2} \right) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}})\, \rho\, U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}), \qquad (11)
\end{aligned}$$

where we have used the rule that qubits located at (i, j) for $i \in [N]$, $j \in [D] \setminus \{1\} := \{2, \ldots, D\}$ are always corrected, i.e., $p(\tilde{s}_i^j = 0) = 1$ for these qubits [Fig. 4(b)].

For the model in Fig. 4(c), we similarly define $\tilde{S}^{(C)} := \{(s_i^j)_{i=1,\ldots,N}^{j=1,\ldots,D} \in \{0,1\}^{N\times D} \mid s_i^j = 0 \text{ if } i \neq q \text{ for all } j \in [D]\} \subset \{0,1\}^{N\times D}$ where any $\tilde{\boldsymbol{s}} \in \tilde{S}^{(C)}$ is an $N \times D$ matrix with D binary elements. Each $s_q^j \in \{0,1\}$ corresponds to the qubit at location $i = q$ in each layer $j \in [D]$ of the model, and the rest of the $(N-1) \times D$ elements are zero, corresponding to the (white) qubits that are always corrected. Thus, we can write

$$\begin{aligned}
&\mathcal{E}_c^{(C)}(\boldsymbol{\theta}, \boldsymbol{p})[\rho] \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(C)}} \prod_{\substack{j\in[D]\\ i\in[N]}} p\big(\tilde{s}_i^j\big)\, U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(C)}} \prod_{j=1}^{D} \left(\frac{1 + (-1)^{\tilde{s}_q^j} p}{2} \right) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}), \qquad (12)
\end{aligned}$$

where we have used $p(\tilde{s}_i^j = 0) = 1$ for $i \in [N] \setminus \{q\}$, $j \in [D]$, i.e., for qubits that are always corrected (Fig. 4(c)). As depicted in the figure, in this model, we choose the position $(i = q = 4)$ for a partially adapted qubit in each layer and repeat the same procedure for all layers $j \in [D]$.

And lastly, for the Fig. 4(d) model, we consider $\tilde{S}^{(D)} := \{(s_i^j)_{i=1,\ldots,N}^{j=1,\ldots,D} \in \{0,1\}^{N\times D} \mid \; s_i^j = 0 \quad \text{if} \quad (i,j) \neq (q,l)\} \subset \{0,1\}^{N\times D}$. Each $\tilde{\boldsymbol{s}} \in \tilde{S}^{(D)}$ is a matrix with only one binary element that corresponds to a qubit located at index $(i, j) = (q, l) = (4, 1)$, while all other elements are zero.

Thus, the model can be expressed as

$$\begin{aligned}
&\mathcal{E}_c^{(D)}(\boldsymbol{\theta}, \boldsymbol{p})[\rho] \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(D)}} \prod_{\substack{j\in[D]\\ i\in[N]}} p\big(\tilde{s}_i^j\big)\, U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \\
&= \sum_{\tilde{\boldsymbol{s}} \in \tilde{S}^{(D)}} \left(\frac{1 + (-1)^{\tilde{s}_q^l} p}{2} \right) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}) \\
&= \left(\frac{1-p}{2} \right) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}_{(q,l)}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}_{(q,l)}) \\
&\quad + \left(\frac{1+p}{2} \right) U_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}} = \boldsymbol{0}) \rho U_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}} = \boldsymbol{0}), \qquad (13)
\end{aligned}$$

where we used $p(\tilde{s}_i^j = 0) = 1 \quad \forall \quad (i,j) \in ([N] \times [D]) \setminus \{(q,l)\}$, as the rest of the qubits in the cluster except the one at the (q, l) position are always corrected [see Fig. 4(d)]. Here, $\tilde{\boldsymbol{s}}_{(q,l)} \in \{0,1\}^{N\times D}$ is a matrix whose (q, l)th element is 1, while all other elements are zero, and $\tilde{\boldsymbol{s}} = \boldsymbol{0}$ is an $N \times D$ matrix with all elements equal to zero.

Notice that in all the models shown in Fig. 4, there is only one trainable correction probability p that allows for

different channel models shown in Eqs. (10)–(13). In Sec. III, we perform our numerical analysis using these models.

Throughout this paper, we describe all the models using the cluster-state MBQC representation, as it provides a transparent picture of measurement adaptivity and byproduct propagation. We note that this representation is in one-to-one correspondence with the equivalent CQCA-based circuit description (see Fig. 2), and both viewpoints are used interchangeably in the analysis of this paper.

D. Loss and gradient

For training, we employ the maximum mean discrepancy (MMD) [26] as an implicit loss function, defined below:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{p}) = \mathop{\mathbb{E}}_{\substack{x\sim q_{(\boldsymbol{\theta},\boldsymbol{p})}\\ y\sim q_{(\boldsymbol{\theta},\boldsymbol{p})}}} [K(x,y)] - 2 \mathop{\mathbb{E}}_{\substack{x\sim q_{(\boldsymbol{\theta},\boldsymbol{p})}\\ y\sim Y}} [K(x,y)] + \mathop{\mathbb{E}}_{\substack{x\sim Y\\ y\sim Y}} [K(x,y)]. \tag{14}$$

Here, the $q_{(\boldsymbol{\theta},\boldsymbol{p})}$ represents the output distribution of the training model with the set of parameters $\{\boldsymbol{\theta}, \boldsymbol{p}\}$ (see Fig. 1), and $Y(x)$ is some target distribution. The $K(x, y)$ in Eq. (14) is referred to as the kernel function that quantifies the similarity between samples x and y.

The MMD loss function is particularly suitable for implicit quantum generative models like VMBQC, whereas Kullback–Leibler (KL) divergence requires access to explicit probability densities and can become ill-defined when the model assigns zero probability to target events. Moreover, Ref. [27] shows that MMD can avoid the trainability problems associated with explicit losses such as KL divergence, provided that an appropriate kernel is chosen.

To train the QCBM, the parameters of the VMBQC model are updated iteratively by computing the gradients of the loss function with respect to the variational angles θ_i^j and correction probabilities p_i^j, i.e., $\frac{\partial \mathcal{L}}{\partial \theta_i^j}$ and $\frac{\partial \mathcal{L}}{\partial p_i^j}$, respectively. References [11,14] provide more details on how to compute these gradients analytically. The corresponding expression is also provided in Appendix A. However, in our training models, we have only one variational parameter p shared across different qubits, and thus one can find the gradient of the loss $\frac{\partial \mathcal{L}}{\partial p}$ using the chain rule

$$\frac{\partial \mathcal{L}}{\partial p} = \sum_{i,j=1}^{N\times D} \frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{p})}{\partial p_i^j}\Big|_{(\boldsymbol{p}:=\{p,\ldots,p\})}. \tag{15}$$

From Eq. (15), we observe that the analytical gradient computation requires computing $\frac{\partial \mathcal{L}}{\partial p_i^j}|_{(\boldsymbol{p}:=\{p,\ldots,p\})}$ at all positions (i, j), which does not speed up the gradient estimation even though it is evaluated at a point determined by the single parameter p. However, one can use the finite difference method to estimate the gradient, which requires only one step,

$$\frac{\partial \mathcal{L}}{\partial p} = \frac{\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{p}+\boldsymbol{\epsilon}) - \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{p}-\boldsymbol{\epsilon})}{2\epsilon} + O(\epsilon^2), \tag{16}$$

where $\boldsymbol{p} \pm \boldsymbol{\epsilon} = \{p_i^j := (p \pm \epsilon)\}_{i,j}$, for $\epsilon > 0$ and $O(\epsilon^2)$ contains the terms of order ϵ^2 in the Taylor expansion of $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{p} \pm \boldsymbol{\epsilon})$.

While Eq. (15) provides the exact gradient of the loss function, it scales poorly with system size. For larger systems, Eq. (16) is therefore preferred, despite being only an approximation of Eq. (15). These gradients are estimated using samples generated by the training and target models. One could also use batch or minibatch updates that can reduce the statistical variance of the loss and gradient estimates by averaging over multiple data samples and circuit shots before updating the parameters. This may stabilize optimization, including the finite-difference gradient estimator in Eq. (16).

III. RESULTS

In this section, we analyze the expressive power of the single-parameter VMBQC models introduced in Sec. II C. We begin by revisiting Theorem 1 of Ref. [11] from the perspective of the present work. In particular, we highlight how this result implies that an advantage of the channel-based model over its unitary counterpart can already be achieved by introducing a single additional trainable parameter.

Observation 1. Let N be the number of qubits, D be the depth of the model, $\boldsymbol{\theta} := \{\theta_i^j | \theta_i^j \in [0, 2\pi)\}_{i\in[N], j\in[D]}$ be the set of trainable rotation angles, and $\boldsymbol{p} := \{p_i^j = p \mid p \in [0, 1)\}_{i\in[N], j=[D]}$ be the set of trainable correction probabilities. Then there exists a tuple (N,D,$\boldsymbol{\theta}$, $\boldsymbol{p}$) for which the set of probability distributions $\mathcal{Q}_{\mathcal{E}_c}(N, D, \boldsymbol{\theta}, \boldsymbol{p})$ generated by the $\mathcal{E}_c$ model [in Eq. (11) and Fig. 4(a)] is strictly larger than the set of distributions $\mathcal{Q}_{U_c}(N, D, \boldsymbol{\theta})$ generated by the pure unitary model.

This observation is a special case of the result presented in Theorem 1 of Ref. [11], specialized to the model depicted in Fig. 4(b). The proof follows the same reasoning. In Ref. [11], the authors validate the Observation 1 for the VMBQC model with width $N = 3$, depth $D = 1$, and for a specific choice of parameters $\boldsymbol{\theta} = \{\theta_i^1\}_{i\in[1,3]}$ and $\boldsymbol{p} = \{p_i^1 = \frac{1}{2}\}_{i\in[1,3]}$. This algebraic result motivates the following numerical investigation into the advantages of the restricted VMBQC model.

In Sec. III A, we numerically investigate the performance of the variational quantum channel models in Eqs. (10)–(13) with only a single trainable correction probability by comparing with a fully deterministic (unitary) model given by Eq. (3).

In Sec. III C, we further examine the effect of a single partially corrected qubit [Fig. 4(d)] on the resulting output distribution of the model. We then extend this analysis to a model with two partially corrected qubits sharing a single correction probability. For this purpose, we focus on the channel model $\tilde{\mathcal{E}}_c(\boldsymbol{\theta}, \boldsymbol{p})$ defined in Eq. (9).

A. Learning quantum channel distributions

1. Training process

First, we generate a dataset with samples drawn from the output distribution of the target model (described in Sec. III A 2), with randomly initialized parameters, and then train different learning models (described in Sec. III A 3) to learn this target distribution via samples using the MMD loss in Eq. (14). All learning models are trained for a fixed number of epochs with identical hyperparameters (width and depth of the cluster state), and results are averaged over multiple

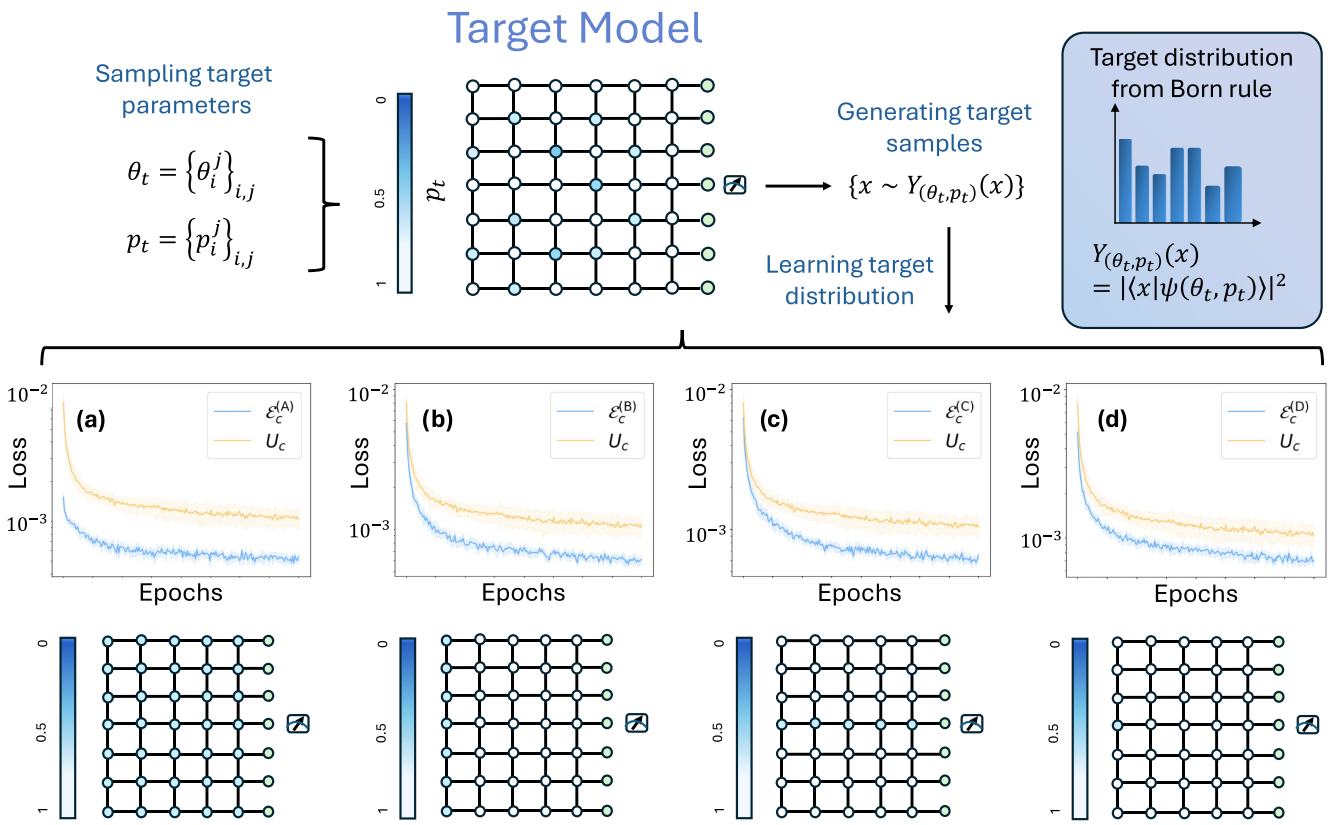


FIG. 5. Learning performances of VMBQC models. Four families of channel models, defined in Eqs. (10)–(13), together with the unitary model in Eq. (3), are trained to approximate a target distribution generated by the channel $\mathcal{E}_c(\boldsymbol{\theta}_t, \boldsymbol{p}_t)$ in Eq. (7). (Top row) The target model has width $N = 7$, depth $D = 6$, and generates target samples $x \sim Y_{(\boldsymbol{\theta}_t,\boldsymbol{p}_t)}(x)$ according to the Born rule (in blue box). The parameters $\boldsymbol{\theta}_t = \{\theta_i^j \in [0, 2\pi)\}$ and $\boldsymbol{p}_t = \{p_i^j \in [0.9, 1)\}$ are drawn independently and uniformly from their specified ranges. (a)–(d) Learning performance of the unitary model U_c (yellow) compared with the channel models $\mathcal{E}_c^{(A)}$, $\mathcal{E}_c^{(B)}$, $\mathcal{E}_c^{(C)}$, and $\mathcal{E}_c^{(D)}$ (blue), all with $N = 7$ and $D = 5$. The qubits (green) in the last layer $D = 6$ (Bottom row) correspond to the output qubits. Curves show the mean MMD loss over 200 epochs, averaged across 10 random initializations; shaded regions indicate standard deviations. (Bottom row) Corresponding channel-based training models, each with a single trainable correction probability p, as specified in Eqs. (10)–(13). The depicted values of p in the figure are not the optimal values found in our experiments but randomly chosen for illustration. The input to all the learning models is a fixed quantum state $|+\rangle^{\otimes N}$.

random initializations, as shown in Figs. 5(a)–5(d). More training details are provided in Appendix B 1.

2. Target model

We randomly select an instantiation of the channel model $\mathcal{E}_c(\boldsymbol{\theta}_t, \boldsymbol{p}_t)$, defined in Eq. (7) with $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ and $\boldsymbol{p} = \boldsymbol{p}_t$, as the target. In particular, for Fig. 5, we choose $N = 7$ and $D = 6$ for the target model. The target is initialized with measurement angles $\boldsymbol{\theta}_t := \{\theta_i^j \in [0, 2\pi)\}_{i,j}$ and correction probabilities $\boldsymbol{p}_t := \{p_i^j \in [0.9, 1)\}_{i,j}$, both drawn uniformly at random. Note that, in this model, each qubit is assigned its own independent correction probability. The resulting output distribution generated by the target model $\mathcal{E}_c(\boldsymbol{\theta}_t, \boldsymbol{p}_t)$ serves as the target distribution for the various learning models defined in Eqs. (3) and (10)–(13).

3. Learning models

The VMBQC learning models considered in this work include a purely unitary model [Eq. (3)] and four families of channel models defined in Eqs. (10)–(13) [Figs. 4(a)–4(d)], which differ only in the placement of the partially uncorrected qubits on the cluster. We numerically compare the learning performances of these channel models with that of the unitary model $U_c(\boldsymbol{\theta})$. All channel and unitary models share the same width $N = 7$ and depth $D = 5$.

4. Analysis

In Figs. 5(a)–5(d), we present the averaged learning curves comparing four families of channel models with the purely unitary model. In all cases, the models are trained to learn the same target distribution generated by another channel model defined in Eq. (7). Across all panels, a significant separation is observed between the final losses achieved by the unitary model U_c (in yellow) and those achieved by the corresponding channel models $\mathcal{E}_c^{(A)}$, $\mathcal{E}_c^{(B)}$, $\mathcal{E}_c^{(C)}$, and $\mathcal{E}_c^{(D)}$ (all in blue). This consistent performance gap shows that channel-based VMBQC models achieve better learning performance on channel-based target distributions than the unitary VMBQC model, even with only a single additional trainable parameter.

To assess the performance retained by the restricted parametrization, we additionally compare the

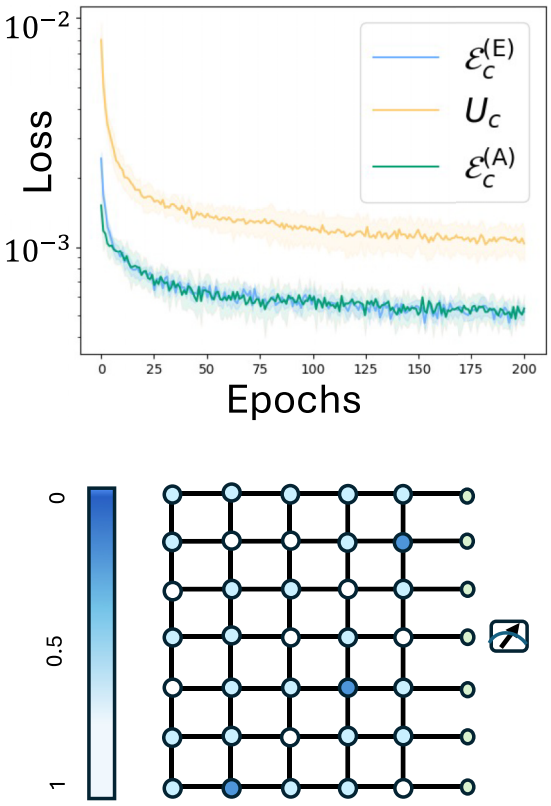


FIG. 6. Comparison with the full-VMBQC model. The unitary model U_c, the single-parameter VMBQC model $\mathcal{E}_c^{(A)}$, and the full-VMBQC model $\mathcal{E}_c^{(E)}$ with $N \times D$ independently trainable correction probabilities are trained on the same target distribution using the same training protocol as in Fig. 5. The upper panel shows the mean MMD loss over 200 epochs, averaged over 10 random initializations, while the shaded regions indicate one standard deviation. The lower panel illustrates the independently trainable correction probabilities p_i^j of the full model across the measured cluster-state qubits. The output qubits are shown in green. The depicted values of p_i^j in the lower panel of the figure are not the optimal values found in our experiments but randomly chosen for illustration.

single-parameter model $\mathcal{E}_c^{(A)}$ with the full-VMBQC model containing $(N \times D)$ independently trainable correction probabilities from Ref. [11]. The results are presented in Sec. III B, where all models are trained on the same target distribution and under the same training protocol.

B. Comparison with the full-VMBQC model

We compare the unitary model U_c, the single-parameter model $\mathcal{E}_c^{(A)}$, and the full-VMBQC channel model, denoted by $\mathcal{E}_c^{(E)}$, with independent correction probabilities p_i^j for every measured qubit i in layer j as defined in Eq. (10) of Ref. [11]. All models use the same circuit depth, target distribution, optimizer, and sampling budget as in Fig. 5. Note that the full-VMBQC channel model $\mathcal{E}_c^{(E)}$, from Ref. [11], belongs to the same family of models as the higher-depth target channel model in Fig. 5 that generates the target samples. As shown in Fig. 6, the single-parameter model achieves a loss comparable to that of the full model for the target considered, indicating that much of the observed learning performance can be retained despite the substantial reduction in correction-probability parameters. This result suggests that, for certain target distributions, the single-parameter channel model is as expressive as the full-VMBQC model in Ref. [11]. This does not imply, however, that the single-parameter model can always perform as well as the full-VMBQC model.

Furthermore, the more spatially restricted single-parameter models converge more slowly and attain higher losses while learning the same target distribution, as shown in Figs. 5(b)–5(d). Thus, parameter sharing can also reduce learning performance depending on the correction pattern. This comparison is numerical and does not imply equivalence between the two model families.

We note that this comparison also focuses on target correction probabilities $\boldsymbol{p}_t := \{p_i^j \in [0.9, 1)\}_{i,j}$, close to the unitary limit. The behavior of the single-parameter model further away from this regime is not systematically studied in this work.

C. Effect of byproducts on the learning performance

So far, we have numerically demonstrated the advantage of using channel-based models with a single correction probability p over unitary models. However, it remains unclear in what way this advantage is connected to the position of the byproduct in the cluster. To further analyze and understand the demonstrated advantage, we consider the model in Fig. 4(d) and compare the advantage between different positions of the byproduct in the two-dimensional (2D) grid. As the model in Fig. 4(b) contains byproducts in the same time slice, and which are therefore commuting, we can also expect the analysis to carry over to this case. To highlight the role of a single byproduct, we therefore isolate its effect by considering VMBQC channel models with exactly one partially adapted qubit [see Fig. 4(d)] and analyze how the resulting localized measurement-induced byproduct influences the output distribution depending on its position in the cluster state.

Here we choose the learning model to be the unitary model U_c [Eq. (3)], and assess its ability to learn different target models that are based on the channel model in Eq. (17) with one byproduct at a specific position in the cluster. For each target, the unitary learning model is trained with 10 random instantiations using the MMD loss, and performance is summarized by recording the minimum MMD loss attained by each model instantiation over all training epochs. The resulting distributions of these minimum losses (shown on the y axis) across different targets (shown on the x axis) are presented in Figs. 7 and 8 using box plots.

For the following analysis of the results, we mainly focus on the lower whisker endpoint (LWE) (bottom horizontal line) of each distribution, which represents the minimum nonoutlier MMD loss within that distribution and thus the best observed performance across 10 learning models, for each target. Furthermore, we refer to each distribution using the color of the box. Further training details are provided in Appendix B 2.

1. Model with one byproduct

We now demonstrate the effect of a single byproduct, arising from one partially adapted qubit, on both the output distribution of the VMBQC channel model and the learning

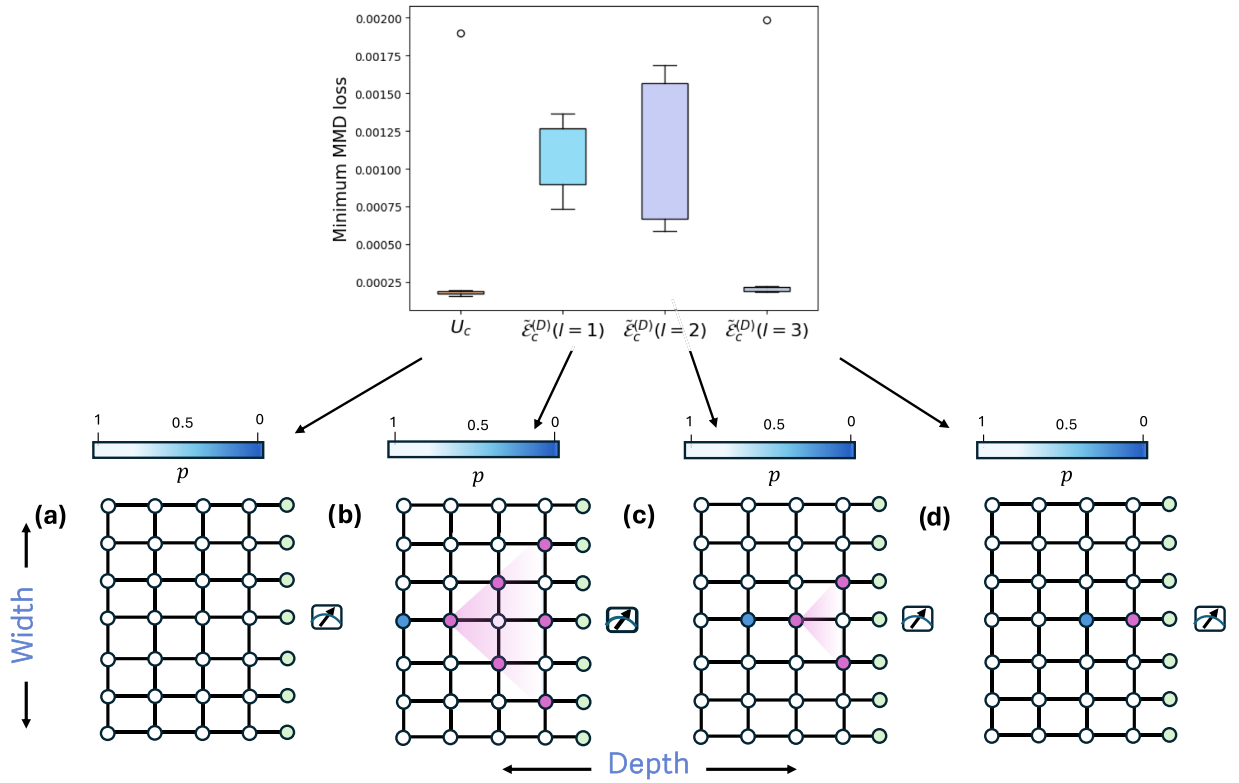


FIG. 7. Effect of a single byproduct. Here we examine how well the unitary model U_c can learn the channel model $\tilde{\mathcal{E}}_c^{(D)}(\boldsymbol{\theta}, \boldsymbol{p})$ when only the measurement outcome of a single qubit is probabilistically adapted [see Eq. (17)]. The x axis labels the different target models, while the y axis reports the distribution of minimum MMD losses achieved by 10 independently initialized unitary learning models U_c over 200 epochs. (a) The target distribution is generated by a unitary model with measurement angles $\boldsymbol{\theta} = \{\theta_i^j \in [0, 1)\}_{i,j}$ drawn uniformly at random. A unitary model is then trained to approximate this distribution. (Top, leftmost) The distribution of minimum MMD losses is shown by the box labeled by U_c on the x axis, with a single outlier indicated in white. (b) A single partially uncorrected qubit (in blue with $p_q^l = 0.135$) is introduced while keeping $\boldsymbol{\theta}$ unchanged to generate a target distribution. (Top, second from left) The corresponding MMD distribution is shown with the $\tilde{\mathcal{E}}_c^{(D)}(l = 1)$ labeled box. (c) The same rotation angles are used, while the partially uncorrected qubit is moved to $(l, q) = (2, 4)$ with the same p_q^l as in panel (b), yielding the MMD distribution shown by the $\tilde{\mathcal{E}}_c^{(D)}(l = 2)$ labeled box. (d) Moving the partially uncorrected qubit to $(l, q) = (3, 4)$ while keeping the rest the same as in panels (a)–(c). (Top, rightmost) The resulting MMD distribution is shown by the $\tilde{\mathcal{E}}_c^{(D)}(l = 3)$ labeled box, with an outlier in white.

performance of a purely unitary model. Specifically, we use the channel model $\tilde{\mathcal{E}}_c(\boldsymbol{\theta}, \boldsymbol{p})$ defined in Eq. (9) as the target. This model distills the non-Clifford contributions of the byproduct by adding a byproduct-dependent Pauli operator $P_{D+1}(\tilde{\boldsymbol{s}})$ that adjusts the final readout basis at the end of the circuit (see Fig. 3). Selecting a specific qubit, characterized by a layer index l and a qubit index q, allows Eq. (9) to be decomposed like Eq. (13) as follows:

$$\begin{aligned}\tilde{\mathcal{E}}_c^{(D)}(\boldsymbol{\theta}, \boldsymbol{p})[\rho] = &\left(\frac{1-p}{2}\right)\tilde{U}_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}_{(q,l)})\rho\tilde{U}_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}}_{(q,l)}) \\ &+ \left(\frac{1+p}{2}\right)\tilde{U}_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}} \\ &= \boldsymbol{0})\rho\tilde{U}_c^\dagger(\boldsymbol{\theta}, \tilde{\boldsymbol{s}} = \boldsymbol{0}),\end{aligned} \tag{17}$$

where $\tilde{\boldsymbol{s}}_{(q,l)} \in \{0, 1\}^{N\times D}$ is a matrix whose (q, l)th element is 1, while all other elements are zero. Note that, in Eq. (17), $\tilde{U}_c(\boldsymbol{\theta}, \tilde{\boldsymbol{s}} = \boldsymbol{0}) = U_c(\boldsymbol{\theta})$, i.e., the pure unitary evolution without any byproduct [Eq. (3)]. With this setup, we perform a series of simulations in which the unitary model U_c, in Eq. (3), is trained to approximate different target distributions generated by various target models, as shown along the x axis in Fig. 7. Throughout these simulations, all models, both target and learning, have the same width $N = 7$ and depth $D = 4$.

We begin with a unitary model U_c [Fig. 7(a) with $p = 1$] that generates a target distribution for some $\boldsymbol{\theta} = \{\theta_i^j \in [0, 1]\}_{i,j}$ chosen uniformly at random for all qubits. Since the learning model is also unitary, it is, in principle, capable of learning the target distribution accurately, as reflected by the MMD distribution in the leftmost (U_c labeled on the x axis) box (Fig. 7). However, one training instance converges to a suboptimal local minimum due to the inherent variability in random initialization. Despite this outlier (depicted by a white dot), the key observation from the plot is the minimum loss achieved by the unitary model, demonstrating that, with a suitable parameter initialization, U_c can reach the optimal loss and faithfully reproduce the target distribution.

In Fig. 7(b), for the $\tilde{\mathcal{E}}_c^{(D)}(\boldsymbol{\theta}, \boldsymbol{p})$ model, we fix the measurement angles $\boldsymbol{\theta}$ to those of the U_c target in Fig. 7(a), and introduce a single partially corrected qubit (in blue) with a

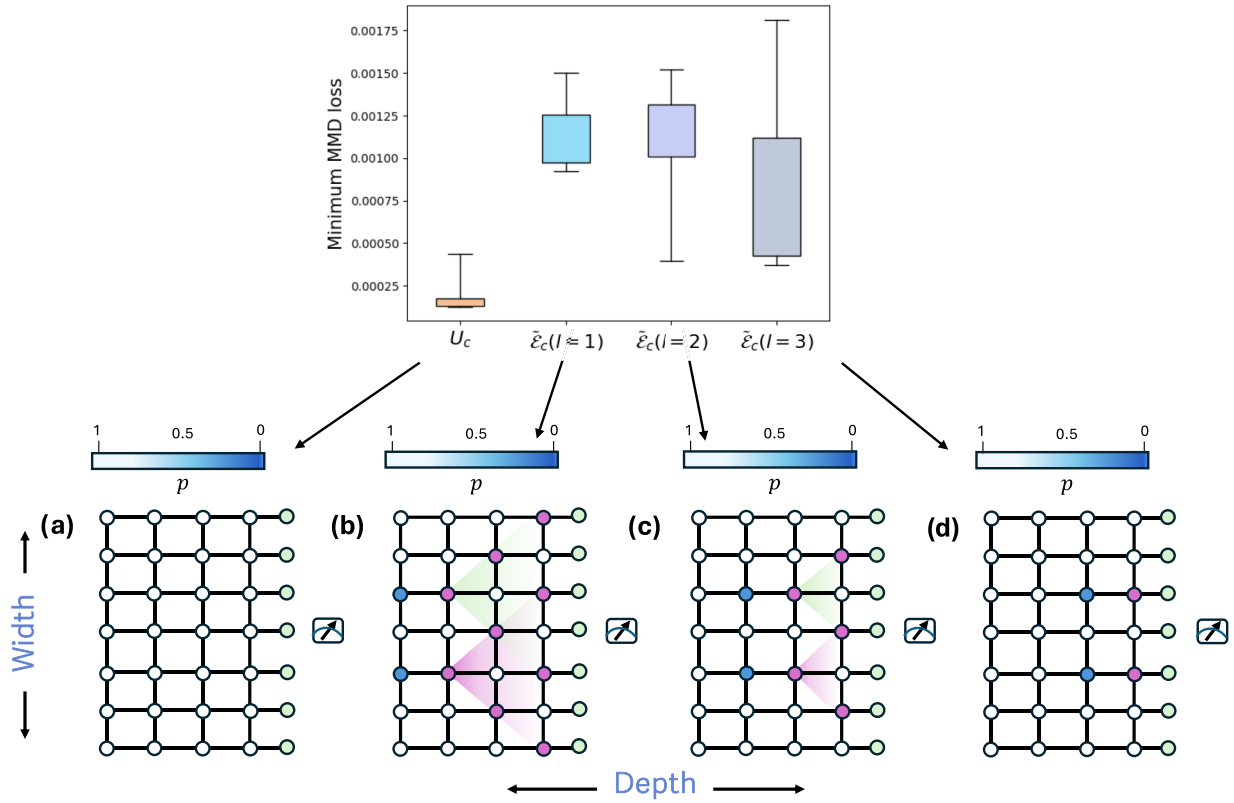


FIG. 8. Effect of two byproducts. Here we examine how well the unitary model U_c can learn the channel model $\tilde{\mathcal{E}}_c(\boldsymbol{\theta}, \boldsymbol{p})$ with two partially uncorrected qubits whose measurement outcomes are probabilistically adapted. The x axis labels the different target models, while the y axis reports the distribution of minimum MMD losses achieved by 10 independent randomly initialized unitary models U_c over 200 epochs. (a) The target distribution is generated from the unitary model itself with $\boldsymbol{\theta} = \{\theta_i^j \in [0, 1)\}_{i,j}$ drawn uniformly randomly. (Top, leftmost) The resulting minimum MMD distribution is shown with the box labeled by U_c on the x axis. (b) We choose two uncorrected qubits (in blue) for some fixed $p_3^1 = p_5^1 = 0.1$ for both of them, keeping the θ the same as earlier. (Top, second from left) The minimum MMD losses, for the corresponding output distribution, are shown using the $\tilde{\mathcal{E}}_c(l = 1)$ labeled box. (c) The uncorrected qubits are moved to $l = 2$, $q = 3, 5$, with the same p_q^l and $\boldsymbol{\theta}$ as before. (Top, third from left) The corresponding MMD distribution, for this target model, is shown by the $\tilde{\mathcal{E}}_c(l = 2)$ labeled box. (d) The uncorrected qubits are placed in layer $l = 3$, and the resulting MMD distribution is shown with the corresponding box labeled $\tilde{\mathcal{E}}_c(l = 3)$.

fixed correction probability $p_q^l = 0.135$ (here, $l = 1$, $q = 4$). This induces a probabilistic byproduct whose forward propagation leads to intermediate angle flips (dark purple) within its light cone (shown in purple gradient, see also Fig. 3). Learning the resulting distribution using unitary models $U_c(\boldsymbol{\theta})$, each initialized with different random parameters, yields the MMD loss distribution shown in Fig. 7 [second from left, $\tilde{\mathcal{E}}_c^{(D)}(l = 1)$ labeled box on x axis], where the LWE is higher compared with the fully unitary target [Fig. 7(a)]. It indicates that even a single probabilistic byproduct makes the channel distribution harder to learn for the unitary model. The effect is most pronounced for low correction probabilities, typically for $p_q^l \in [0.1, 0.2]$.

In Fig. 7(c), we retain the same measurement angles from Fig. 7(a) and move the partially adapted qubit to the next layer, $l = 2$, $q = 4$. As a result, the LWE of the corresponding channel distribution [Fig. 7, third from left, $\tilde{\mathcal{E}}_c^{(D)}(l = 2)$ labeled box] decreases relative to Fig. 7(b) but remains higher than the fully unitary case [Fig. 7(a)]. Shifting the uncorrected qubit reduces the byproduct light cone size (purple gradient), leading to a smaller non-Clifford effect of the byproducts (dark purple) and a smaller impact on the output distribution.

In Fig. 7(d), we maintain the same setup and move the partially uncorrected qubit to $l = 3$, $q = 4$. In this configuration, all the unitary learning models reach a very low MMD loss [Fig. 7, rightmost, $\tilde{\mathcal{E}}_c^{(D)}(l = 3)$ labeled box], with a single outlier due to poor initialization. This illustrates that, while a single probabilistic byproduct can make the output distribution of the channel model harder to learn for the unitary model, its impact depends strongly on its position. Notably, placing the uncorrected qubit in the final layer ($l = 4$) has no effect, as no further angle flips occur and the output remains effectively unitary.

2. Model with two byproducts

In this section, we extend the previous analysis with byproducts at two different positions but with the same correction probability and their impact on the output distribution of the $\tilde{\mathcal{E}}_c(\boldsymbol{\theta}, \boldsymbol{p})$ model. Similar to Sec. III C 1, we conduct a series of simulations with four different target models [Figs. 8(a)–8(d)], and the results follow a trend similar to that in Fig. 7. All target models and the unitary learning model have the same width $N = 7$ and depth $D = 4$, with the last layer being the measurement layer.

We start again with a unitary target model U_c [Fig. 8(a)], which defines a target distribution using measurement angles $\boldsymbol{\theta} = \{\theta_i^j \in [0, 1]\}_{i,j}$ drawn uniformly at random (distinct from those in Sec. III C 1). Since the learning model is also unitary, it successfully reproduces the target distribution, as indicated by the low MMD losses (leftmost, U_c labeled box on x axis).

In Fig. 8(b), we keep the same measurement angles $\boldsymbol{\theta}$ used in Fig. 8(a) and introduce two partially uncorrected qubits (highlighted in blue) that share a single fixed correction probability $p_3^1 = p_5^1 = 0.1$. The corresponding byproduct light cones (green and purple gradients) overlap, leading to angle flips (shown in dark purple) within their combined region. Training the unitary models, with random initializations, on the resulting target distribution yields a distribution of minimum MMD losses, shown in Fig. 8 (second from left, $\tilde{\mathcal{E}}_c(l = 1)$ labeled box on x axis). As expected, the LWE of the MMD loss distribution increases compared with the purely unitary target, with the effect being most pronounced for small values of p_q^l, typically within $[0.1, 0.2]$.

In Fig. 8(c), both partially adapted qubits are shifted to the next layer ($l = 2$ with $q = 3, 5$), while keeping the same measurement angles $\boldsymbol{\theta}$ as earlier. The distribution of MMD [Fig. 8, third from left, $\tilde{\mathcal{E}}_c(l = 2)$ labeled box] shows that the LWE decreases relative to Fig. 8(b) but remains higher than the pure unitary case [Fig. 8(a)]. The placement of the byproducts shrinks the combined light cone (green and purple gradients) compared with Fig. 8(b), resulting in a smaller non-Clifford effect of the byproducts (shown in dark purple) and lower loss (LWE) than Fig. 8(b).

Lastly, in Fig. 8(d), the partially uncorrected qubits are placed at layer $l = 3$ while the rest remain the same as earlier. The corresponding distribution of the MMD losses is shown using the rightmost [$\tilde{\mathcal{E}}_c(l = 3)$ labeled] box in Fig. 7. However, notice that the LWE remains higher than that of the fully unitary target [Fig. 8(a)].

In this section, we have been able to partially attribute the learning advantage demonstrated in the previous section (Sec. III A) to the position of the byproducts in the cluster. The results in Secs. III C 1 and III C 2 highlight that, while randomness in the form of probabilistically adapted qubits can influence the output distribution, its impact depends strongly on their position in the cluster state.

Since the models considered in Secs. III C 1 and III C 2, as well as the other models used in the numerical analysis, are translationally invariant and use periodic boundary conditions, the absolute position of a measurement-induced byproduct within a given layer has the same light-cone effect on the rest of the qubits in the cluster state as shown in Fig. 7. Therefore, the position of the byproduct does not affect the output distribution significantly. The relevant quantity is instead its distance from the final readout layer. In Figs. 7 and 8, we observe that larger distances generally influence the output distributions more significantly.

IV. CONCLUSION AND OUTLOOK

In this work, we investigated the variational measurement-based quantum computation (VMBQC) framework for generative modeling introduced in Ref. [11]. VMBQC leverages the intrinsic indeterminacy of measurement outcomes in MBQC to realize quantum channels as learning models. Specifically, Ref. [11] introduced independent trainable parameters for each qubit in the MBQC that govern whether random measurement outcomes in MBQC are corrected. In contrast, the present work reduces these additional classical resources, namely, the trainable correction probabilities, to a single shared parameter while preserving an expressive advantage over the corresponding purely unitary model. This setting substantially reduces the sampling overhead associated with gradient estimation, since only one additional parameter must be differentiated, compared with the full-VMBQC channel model of Ref. [11], which requires estimating gradients with respect to $N \times D$ additional parameters, where N is the height of the cluster state and D is its depth.

We demonstrate the effectiveness of our simplified channel VMBQC approach, both algebraically and numerically, through various learning tasks, benchmarking it against purely unitary models. Our results show that channel-based models, even when endowed with only a single trainable correction probability, can generate output distributions that cannot be learned, up to a desired accuracy, by the corresponding unitary models. Importantly, these numerical results are robust: the qualitative behavior and conclusions remain unchanged under different random choices of (target) model parameters within the same range, indicating that the observed advantage is not an artifact of specific instances but reflects an intrinsic property of the model class. For the benchmark considered, the single-parameter model retains much of the task-relevant expressive capability of the full-VMBQC model and achieves comparable learning performance. However, we do not establish a general quantitative characterization of how much expressive power is retained by the restricted VMBQC channel model family compared with the model family with independent correction probabilities as introduced in Ref. [11], across arbitrary target distributions.

We also point out that the success of these learning models might depend on the underlying target distribution. Furthermore, properly chosen hyperparameters, particularly the choice and initialization of the shared correction probability, can affect the performance of the model. Most importantly, the benefit of introducing such measurement-induced classical randomness is expected to be most pronounced in the shallow-circuit regime, since sufficiently deep unitary models may already access a much larger portion of the relevant state space.

Moreover, we show that the placement of the random byproducts plays a critical role in determining their impact on the output distribution. By strategically selecting which qubits retain byproducts, one can utilize the channel characteristics of the VMBQC model while keeping the circuit width and depth fixed and the classical control resources minimal. Together, these results provide concrete design principles for quantum generative models that exploit measurement-induced randomness with very few classical control resources within the MBQC framework.

ACKNOWLEDGMENTS

The authors thank Marius Krumm for useful discussions. This research was funded in part by the Austrian Science Fund

(FWF) [SFB BeyondC F7102, DOI: 10.55776/F71; WIT9503323, DOI: 10.55776/WIT9503323]. This work was also supported by the European Union (ERC Advanced Grant, QuantAI, No. 101055129).

The views and opinions expressed in this article are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council—neither the European Union nor the granting authority can be held responsible for them.

DATA AVAILABILITY

The data that support the findings of this article are openly available [28].

APPENDIX A: GRADIENTS OF LOSS FUNCTION

For training purposes, we require calculating the gradient of the loss function in Eq. (14). The resulting gradient with respect to a variational measurement angle θ_i^j (see Ref. [14]) of the VMBQC model is

$$\frac{\partial \mathcal{L}}{\partial \theta_i^j} = \mathop{\mathbb{E}}_{\substack{x\sim q_{(\boldsymbol{\theta}^+,\boldsymbol{p})}\\ y\sim q_\phi}}[K(x,y)] - \mathop{\mathbb{E}}_{\substack{x\sim q_{(\boldsymbol{\theta}^-,\boldsymbol{p})}\\ y\sim q_\phi}}[K(x,y)] - \mathop{\mathbb{E}}_{\substack{x\sim q_{(\boldsymbol{\theta}^+,\boldsymbol{p})}\\ y\sim Y}}[K(x,y)] + \mathop{\mathbb{E}}_{\substack{x\sim q_{(\boldsymbol{\theta}^-,\boldsymbol{p})}\\ y\sim Y}}[K(x,y)]. \tag{A1}$$

Here, $q_{(\boldsymbol{\theta}^+,\boldsymbol{p})}$ and $q_{(\boldsymbol{\theta}^-,\boldsymbol{p})}$ are the output distributions of the VMBQC channel model with parameters $\boldsymbol{\theta}^\pm = \boldsymbol{\theta} \pm \frac{\pi}{2}\boldsymbol{e}_i^j$ and $\boldsymbol{e}_i^j$ is a vector with a 1 at position (i, j) and zeros elsewhere. The parameter shifting is done for each angle individually, while the other angles remain unchanged. Y is the target distribution generated from the target model. In the channel models considered for training in this study, there is a single trainable correction probability $p_i^j = p$ shared across different qubits (i, j). Thus, we need to compute the gradient only with respect to one p_i^j, for some (i, j). In Ref. [11], the general form of the gradient with respect to any p_i^j is provided as

$$\frac{\partial \mathcal{L}}{\partial p_i^j} = 2\{\mathop{\mathbb{E}}_{\substack{x\sim q_{\boldsymbol{\phi}^1}\\ y\sim q_{\boldsymbol{\phi}}}}[K(x,y)] - \mathop{\mathbb{E}}_{\substack{x\sim q_{\boldsymbol{\phi}^0}\\ y\sim q_{\boldsymbol{\phi}}}}[K(x,y)]\} - 2\{\mathop{\mathbb{E}}_{\substack{x\sim q_{\boldsymbol{\phi}^1}\\ y\sim Y}}[K(x,y)] - \mathop{\mathbb{E}}_{\substack{x\sim q_{\boldsymbol{\phi}^0}\\ y\sim Y}}[K(x,y)]\}. \tag{A2}$$

Here, $\boldsymbol{\phi}$ is the set of unchanged parameters, i.e., $\boldsymbol{\phi} = (\boldsymbol{\theta}, \boldsymbol{p})$, and $\boldsymbol{\phi}^1 = (\boldsymbol{\theta}, \boldsymbol{p}^1)$ and $\boldsymbol{\phi}^0 = (\boldsymbol{\theta}, \boldsymbol{p}^0)$, where

$$(p^b)_{i'}^{j'} := \begin{cases} p_{i'}^{j'} & \text{for } i', j' \neq i, j \\ b & \text{for } i', j' = i, j \end{cases} \tag{A3}$$

for the i'th qubit and j'th layer, and $b \in \{0, 1\}$. While calculating the gradient of the loss $\mathcal{L}$ with respect to p_i^j, the other correction probabilities $p_{i'}^{j'}$, $(i', j') \neq (i, j)$, remain unchanged, whereas the only modification occurs at the (i, j)th location, where the correction probability p_i^j is set to zero for $b = 0$ and 1 for $b = 1$. The detailed derivation of this gradient is provided in Ref. [11]. To find the gradient with respect to any other $p_{i'}^{j'} = p$, $(i'j') \neq (i, j)$, we use the chain rule

$$\frac{\partial \mathcal{L}}{\partial p} = \sum_{i,j=1}^{N\times D} \frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{p})}{\partial p_i^j}\Big|_{(\boldsymbol{p}:=\{p,\ldots,p\})}. \tag{A4}$$

APPENDIX B: TRAINING DETAILS

All the training in this paper is performed with the Adagrad [29] optimizer, using the gradients in Eqs. (A1) and (A2).

1. Learning quantum channel distribution

In this appendix, we provide the training details corresponding to the results presented in Sec. III A. We employ gradient-based optimization to train the learning models, using the gradients derived in Eqs. (A1) and (A2). Specifically, both the variational measurement angles $\boldsymbol{\theta}$ and the correction probabilities $\boldsymbol{p}$ are updated during training. Distinct learning rates are used for these parameters: a fixed learning rate of 0.1 for the measurement angles $\boldsymbol{\theta}$, and 0.2 for the correction probability $p_i^j = p$ shared across different qubits (i, j). A systematic hyperparameter optimization is beyond the scope of this paper. We additionally employ the Adagrad optimizer, which adaptively rescales the effective learning rates during training based on the accumulated gradient history.

For each experiment, we first generate a dataset of 8000 samples from a target distribution defined by a VMBQC channel model. The target model parameters, measurement angles $\boldsymbol{\theta}_t = \{\theta_i^j \in [0, 2\pi)\}_{i,j}$ and correction probabilities $\boldsymbol{p}_t = \{p_i^j = p \in [0.9, 1)\}_{i,j}$, are drawn independently and uniformly at random. Both unitary and channel-based VMBQC models are then trained to learn this target distribution.

All learning models initialize their variational measurement angles $\boldsymbol{\theta}$ uniformly at random from $[0, 2\pi)$. For the channel models, the single trainable correction probability p is initialized uniformly from the interval [0.85,1).

We choose the correction probabilities in the target model of Fig. 5 to be $\boldsymbol{p}_t = \{p_i^j \in [0.9, 1)\}$, i.e., close to the unitary regime so that the target distribution remains sufficiently nonuniform [11] and the comparison probes a nontrivial learning task. Otherwise, both models could approximate an almost uniform target distribution without revealing a meaningful separation.

Initializing the correction probability p for the learning model close to unity [between [0.85,1)] is empirically favorable, as smaller values tend to produce overly uniform output distributions [11], from which optimization is empirically less effective.

During training, each model generates 8000 samples per epoch, matching the size of the target dataset. These samples, together with the target samples, are used to evaluate the MMD loss function [Eq. (14)] on a classical computer (see Fig. 1). Model parameters are then updated after each epoch using the gradients given in Eqs. (A1) and (A2). Each model is trained for a total of 200 epochs.

To ensure robustness and statistical reliability, all experiments are repeated with 10 independent random initializations of the model parameters. The reported results correspond to the average performance across these runs, as shown in

Figs. 5(a)–5(d). All learning models in this section are trained using identical hyperparameters, including the cluster-state width N, depth D, and dataset size.

2. Effect of byproducts on the learning performance

This appendix provides the training details corresponding to the results presented in Secs. III C 1 and III C 2. In contrast with Sec. III A, here we fix the learning model to be the unitary VMBQC U_c in Eq. (3) and evaluate its ability to learn a variety of target models.

For each target model, the parameters, measurement angles $\boldsymbol{\theta} = \{\theta_i^j \in [0, 1]\}_{i,j}$, and correction probabilities $\boldsymbol{p} = \{p_i^j = p \in [0.1, 0.2]\}_{i,j}$ are sampled randomly, and a dataset of 8000 samples is generated from the corresponding output distribution. To learn each target distribution, we train 10 independent instances of the unitary model U_c, each initialized with different random parameters. Training is performed using the MMD loss function [Eq. (14)], which is evaluated at every epoch and used to update the model parameters.

Each unitary learning model is trained for 200 epochs. While the loss is monitored throughout training, performance is summarized by recording the minimum MMD loss achieved by each model across all 200 epochs. The distributions of these minimum losses are shown as box plots in Figs. 7 and 8, where each box (and the vertical line) corresponds to the distribution of 10 minimum MMD losses obtained by the U_c model, for a given target model. Across all experiments in this section, we use a fixed learning rate of 0.1 for the unitary learning model.

The figures in these sections employ standard box plots to summarize the distributions of MMD losses across different target models. The x axis labels the different target models under consideration, while the y axis reports the corresponding distribution of minimum MMD loss values. Each box spans the interquartile range (IQR), representing the central 50% of the data. The whiskers (vertical lines), containing the rest of the data, extend to the most extreme data points within $1.5 \times$ IQR from the quartiles, with values beyond this range treated as outliers. Throughout our analysis, we focus on the lower whisker endpoint (bottom horizontal line), which corresponds to the minimum nonoutlier MMD loss achieved for a given target model and learning method, and thus reflects the best performance without reliance on statistical outliers.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).

[2] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[3] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, Practical quantum advantage in quantum simulation, Nature (London) **607**, 667 (2022).

[4] Google Quantum AI and Collaborators, Observation of constructive interference at the edge of quantum ergodicity, Nature (London) **646**, 825 (2025).

[5] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. **41**, 303 (1999).

[6] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, Measurement-based quantum computation, Nat. Phys. **5**, 19 (2009).

[7] R. Raussendorf and H. J. Briegel, A one-way quantum computer, Phys. Rev. Lett. **86**, 5188 (2001).

[8] R. Raussendorf, D. E. Browne, and H. J. Briegel, Measurement-based quantum computation on cluster states, Phys. Rev. A **68**, 022312 (2003).

[9] H. J. Briegel and R. Raussendorf, Persistent entanglement in arrays of interacting particles, Phys. Rev. Lett. **86**, 910 (2001).

[10] H.-Y. Huang, M. Broughton, N. Eassa, H. Neven, R. Babbush, and J. R. McClean, Generative quantum advantage for classical and quantum problems, arXiv:2509.09033.

[11] A. Majumder, M. Krumm, T. Radkohl, L. J. Fiderer, H. P. Nautrup, S. Jerbi, and H. J. Briegel, Variational measurement-based quantum computation for generative modeling, Phys. Rev. A **110**, 062616 (2024).

[12] B. Coyle, S. Raj, N. Mathur, E. A. Cherrat, N. Jain, S. Kazdaghli, and I. Kerenidis, Training-efficient density quantum machine learning, npj Quantum Inf. **11**, 172 (2025).

[13] 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.

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

[15] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, A generative modeling approach for benchmarking and training shallow quantum circuits, npj Quantum Inf. **5**, 45 (2019).

[16] J. M. Tomczak, *Deep Generative Modeling* (Springer International Publishing, 2022).

[17] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).

[18] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, Nat. Commun. **12**, 6961 (2021).

[19] S. Mohamed and B. Lakshminarayanan, Learning in implicit generative models, arXiv:1610.03483.

[20] H. P. Nautrup and H. J. Briegel, Measurement-based quantum computation from Clifford quantum cellular automata, Phys. Rev. A **110**, 062617 (2024).

[21] A. Mantri, T. F. Demarie, and J. F. Fitzsimons, Universality of quantum computation with cluster states and (x, y)-plane measurements, Sci. Rep. **7**, 42861 (2017).

[22] D. T. Stephen, H. P. Nautrup, J. Bermejo-Vega, J. Eisert, and R. Raussendorf, Subsystem symmetries, quantum cellular automata, and computational phases of quantum matter, Quantum **3**, 142 (2019).

[23] R. Raussendorf, C. Okay, D.-S. Wang, D. T. Stephen, and H. P. Nautrup, Computationally universal phase of quantum matter, Phys. Rev. Lett. **122**, 090501 (2019).

[24] R. Raussendorf, Quantum computation via translation-invariant operations on a chain of qubits, Phys. Rev. A **72**, 052301 (2005).

[25] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[26] 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).

[27] 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).

[28] A. Majumder, VMBQC_1p, 2026, https://github.com/ArunM10/Restricted_VMBQC.

[29] J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. **12** (2011).