Symmetry Breaking in Geometric Quantum Machine Learning in the Presence of Noise

Cenk Tüysüz,[1,2,*] Su Yeon Chang,[3,4] Maria Demidik,[1,5] Karl Jansen,[1,5] Sofia Vallecorsa,[3] and Michele Grossi[3,†]

[1] *Deutsches Elektronen-Synchrotron DESY, Zeuthen 15738, Germany*
[2] *Institut für Physik, Humboldt-Universität zu Berlin, Berlin 12489, Germany*
[3] *European Organization for Nuclear Research (CERN), Geneva 1211, Switzerland*
[4] *Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland*
[5] *Computation-Based Science and Technology Research Center, The Cyprus Institute, Nicosia 2121, Cyprus*

(Received 5 February 2024; revised 8 May 2024; accepted 20 June 2024; published 23 July 2024)

Geometric quantum machine learning based on equivariant quantum neural networks (EQNNs) recently appeared as a promising direction in quantum machine learning. Despite encouraging progress, studies are still limited to theory, and the role of hardware noise in EQNN training has never been explored. This work studies the behavior of EQNN models in the presence of noise. We show that certain EQNN models can preserve equivariance under Pauli channels, while this is not possible under the amplitude damping channel. We claim that the symmetry breaks linearly in the number of layers and noise strength. We support our claims with numerical data from simulations as well as hardware up to 64 qubits. Furthermore, we provide strategies to enhance the symmetry protection of EQNN models in the presence of noise.

DOI: 10.1103/PRXQuantum.5.030314

I. INTRODUCTION

Variational quantum algorithms (VQAs) appear to be one of the promising algorithms of the noisy intermediate-scale quantum (NISQ) era [1] in the literature [2]. Furthermore, recent results showed noise resilience of VQAs, which further increased hope [3]. However, there exist many roadblocks to making this promise a reality. Some problems that are common to most VQAs are barren plateaus (BPs), i.e., the number of shots needed to estimate the sufficiently precise values of the cost function grows exponentially [4,5], many local minima [6–8], and the lack of efficient gradient computation (e.g., parameter shift rules require circuit executions that scale linearly in the number of parameters) [9]. While certain issues can be partially alleviated through a range of methods [10–14], faithfully running these algorithms on NISQ hardware, beyond what is classically simulable [e.g., $n > 40$ qubits and at least $\log(n)$ depth], is still a practical challenge.

Proposals of geometric quantum machine learning (GQML) opened new avenues, which in theory bring VQAs closer to practicality [15]. This has attracted the attention of the community and many applications have appeared [16–18]. The GQML framework leverages inductive biases on problems and uses this to construct algorithms with improved trainability and generalization [19]. This requires the circuit to have a certain structure from the initial state until the final measurements. However, this is where NISQ hardware fails due to the presence of coherent and incoherent errors [1,20]. Issues such as noise-induced BPs [21] and their implications for statistical learning of VQAs [22] are among the studied complications. General behavior of VQAs in the presence of noise has also been a topic of study in the literature [23,24], including state preparation and time evolution of quantum systems, in which many physical symmetries arise [25,26]. However, these results do not directly translate to the setting of GQML. For this reason, we study the behavior of these algorithms, specifically equivariant quantum neural networks (EQNNs), under hardware noise in this work.

In this paper, we study the behavior of EQNN models in the presence of noise. Our theoretical and numerical results indicate that, for the models considered, equivariance can be protected under realistic Pauli channels. We further show that the symmetry is broken under the nonunital amplitude damping channel. We characterize

*Contact author: cenk.tueysuez@desy.de
†Contact author: michele.grossi@cern.ch

this with metrics that we introduce and show that symmetry breaks approximately linearly in the number of layers and the noise strength. Moreover, we provide strategies such as choice of representation and adaptive thresholding to improve performance.

We structure the paper as follows. In Sec. II, we introduce the necessary preliminary definitions from how to construct EQNNs, to how the hardware noise is modeled. Then, in Sec. III, we start by constructing a toy model and use it to show how hardware noise can break the equivariance. After establishing the theoretical intuition, we define data-driven metrics to quantify the symmetry breaking. Section IV consists of numerical experiments performed with classical simulators as well as NISQ hardware. In Sec. V, we share our point of view on what error mitigation means for the results that we establish, and we conclude by giving suggestions for deploying EQNN models on hardware and discussing future directions and some open questions.

II. FRAMEWORK

A. Equivariant quantum neural networks

This paper focuses on the supervised learning task over a classical data space $\mathcal{R}$, where the data point $\mathbf{x}_i \in \mathcal{R}$ is associated with a label $y_i \in \mathcal{Y}$ following the hidden distribution $f : \mathcal{R} \to \mathcal{Y}$. In the most general framework of quantum machine learning (QML) manipulating the classical data, we embed each data point $\mathbf{x}$ into a quantum state $\rho_{\mathbf{x}} \in \mathcal{M}$ with a certain quantum feature map $\Psi : \mathcal{R} \to \mathcal{M}$, where $\mathcal{M}$ is the space of density matrices [27].

The input state is transformed via a quantum map $\mathcal{U}_{\boldsymbol{\theta}}(\rho)$, which is the adjoint action of $U_{\boldsymbol{\theta}}$ on state ρ,

$$\mathcal{U}_{\boldsymbol{\theta}}(\rho_{\mathbf{x}}) = U_{\boldsymbol{\theta}} \rho_{\mathbf{x}} U_{\boldsymbol{\theta}}^{\dagger} \tag{1}$$

with $U_{\boldsymbol{\theta}}$ the quantum neural network (QNN) parameterized by a set of trainable parameters $\boldsymbol{\theta}$. Without losing generality, we consider the most general setup where the final prediction of the QNN is the expectation value of an observable O:

$$\hat{y}(\mathbf{x}) = \hat{f}_{\boldsymbol{\theta}}(\rho_{\mathbf{x}}) = \mathrm{Tr}[\mathcal{U}_{\theta}(\rho_{\mathbf{x}}) O]. \tag{2}$$

Throughout the training process, the model learns the hidden data distribution within the training set, aiming for $\hat{f}_{\boldsymbol{\theta}}$ to closely approximate the target function f. At the end of the training, we expect that $\mathcal{U}_{\boldsymbol{\theta}}$ can also predict the labels of the unseen test set.

The key idea behind geometric quantum machine learning (GQML) is to design models that capture meaningful relations in the dataset by incorporating the architecture with geometric priors. In the case of geometric supervised learning, we consider the *label symmetry* of the dataset given as the following definition.

Definition 1 (Invariance). Let us consider a symmetry group $\mathcal{S}$ with representation $R\colon \mathcal{S} \to \mathrm{Aut}(\mathcal{R})$, where $\mathrm{Aut}(\mathcal{R})$ is the automorphism group acting on the classical data space $\mathcal{R}$. We say that a function h has a label symmetry if and only if h remains *invariant* under the action of the elements in $\mathcal{S}$, i.e.,

$$h(\rho_{R(g)\cdot\mathbf{x}}) = h(\rho_{\mathbf{x}}) \quad \text{for all } g \in \mathcal{S}, \tag{3}$$

with $R(g)$ the representation of a group element g.

GQML aims to construct a QNN ansatz that guarantees this label symmetry so that the final prediction $\hat{y}(\mathbf{x})$ is invariant under the action of any symmetry group element $g \in \mathcal{S}$. Recent papers suggest approaching the GQML with an $\mathcal{S}$*-equivariant quantum model* [15,19].

Definition 2 (Equivariant embedding). We call an embedding $\Psi : \mathcal{R} \to \mathcal{M}$ with $\Psi(\mathbf{x}) = \rho_{\mathbf{x}}$ *equivariant* with respect to a symmetry element g if and only if there exists a unitary representation $R_q(g)$ such that

$$\rho_{R(g)\cdot\mathbf{x}} = R_q(g) \rho_{\mathbf{x}} R_q^{\dagger}(g). \tag{4}$$

We call $R_q(g)$ the unitary representation of g *induced* by embedding Ψ [19]. The group symmetry emerges naturally in the QNN architecture via the equivariant embedding and can be captured by the equivariant quantum gates. For simplicity, let us focus on a set of quantum gates of the form

$$U_G(\theta) = \exp(-i\theta G), \qquad G \in \mathcal{G}, \tag{5}$$

where G is a Hermitian generator and $\mathcal{G}$ is the generator set of U.

Definition 3 (Equivariant gate). A quantum gate $U_G(\theta) = \exp(-i\theta G)$ with $\theta \in \mathbb{R}$ is called *equivariant* with respect to $\mathcal{S}$ if and only if it commutes with $R_q(g)$ for all $g \in \mathcal{S}$, i.e.,

$$[U_G(\theta), R_q(g)] = 0 \quad \text{for all } \theta \in \mathbb{R} \text{ and all } g \in \mathcal{S}, \tag{6}$$

or, equivalently,

$$[G, R_q(g)] = 0 \quad \text{for all } g \in \mathcal{S}. \tag{7}$$

Different methods have been proposed to construct the equivariant gateset [28], such as the *twirling* method, which is the most common and practical method for a finite symmetry group.

Similarly, a QNN ansatz is said to be equivariant if and only if it consists of equivariant quantum gates. By

combining the equivariant embedding and the equivariant QNN ansatz with an equivariant observable O, i.e.,

$$R_q(g)OR_q^\dagger(g) = O \quad \text{for all } g \in \mathcal{S}, \tag{8}$$

we construct an *invariant quantum classifier model* that guarantees this label symmetry.

Lemma 1 (Invariance from equivariance). A quantum learning model that consists of equivariant embedding layer, an equivariant quantum circuit ansatz and an equivariant observable with respect to a symmetry group $\mathcal{S}$, is invariant with respect to $\mathcal{S}$:

$$\begin{aligned}\hat{y}(R(g)\cdot\mathbf{x}) &= \mathrm{Tr}[U_{\boldsymbol{\theta}}\rho_{R(g)\cdot\mathbf{x}}U_{\boldsymbol{\theta}}^\dagger O]\\ &= \mathrm{Tr}[U_{\boldsymbol{\theta}}R_q(g)\rho_{\mathbf{x}}R_q^\dagger(g)U_{\boldsymbol{\theta}}^\dagger O]\\ &= \mathrm{Tr}[R_q(g)U_{\boldsymbol{\theta}}\rho_{\mathbf{x}}U_{\boldsymbol{\theta}}^\dagger OR_q^\dagger(g)]\\ &= \mathrm{Tr}[R_q^\dagger(g)R_q(g)\mathcal{U}_\theta(\rho_{\mathbf{x}})O]\\ &= \mathrm{Tr}[\mathcal{U}_\theta(\rho_{\mathbf{x}})O] = \hat{y}(\mathbf{x}) \quad \text{for all } g \in \mathcal{S}.\end{aligned} \tag{9}$$

The equivariant QNN leads to the trade-off between the gain of expressibility and the loss of expressibility by constraining the search space that the model can explore. In the previous studies, GQML has shown promising results in various problem setups leveraging the advantage in terms of complexity, trainability, and generalization [15,18,28–32]. However, all the tests have been undertaken in the absence of hardware noise and the impact of noise on the EQNN has never been studied before.

B. Noise models

In this work, we only consider quantum channels acting locally on qubits. Some examples of these channels are the *bit flip* (BF) channel, *depolarizing* (DP) channel, and *amplitude damping* (AD) channel. One way to define the action of a noise channel $\mathcal{N}$ on the quantum state ρ is through the Kraus operators K [33]. Then this can be written as

$$\mathcal{N}(\rho) = \sum_i K_i\rho K_i^\dagger. \tag{10}$$

Bit flip channel. The BF channel with probability p can be described using two Kraus operators $K_0 = \sqrt{1-p}I$ and $K_1 = \sqrt{p}X$. The action of the BF channel on the single-qubit state simply becomes

$$\mathcal{N}(\rho) = (1-p)\rho + pX\rho X. \tag{11}$$

This can be extended to multiqubit systems. In the two-qubit case, the action of the noise channel can be written as

$$\begin{aligned}\mathcal{N}(\rho) = {} & (1-p_0)(1-p_1)\rho\\ & + p_0(1-p_1)(X\otimes I)\rho(X\otimes I)\\ & + (1-p_0)p_1(I\otimes X)\rho(I\otimes X)\\ & + p_0p_1(X\otimes X)\rho(X\otimes X),\end{aligned} \tag{12}$$

where p_0 and p_1 are the probabilities of acting on qubit 0 and qubit 1, respectively. Following this logic, all local noise channels can be generalized to multiqubit systems.

Depolarizing channel. The Kraus operators of the DP channel are $K_0 = \sqrt{1-p}I$, $K_1 = \sqrt{p/3}X$, $K_2 = \sqrt{p/3}Y$, $K_3 = \sqrt{p/3}Z$. The single-qubit DP channel shrinks the Bloch sphere from all directions symmetrically. Hence, any quantum state moves towards the maximally mixed state under the action DP channel.

Pauli channel. Both the BF and DP channels are special cases of Pauli channels. The Kraus operators of the Pauli channel are $K_0 = \sqrt{1-p_x-p_y-p_z}I$, $K_1 = \sqrt{p_x/3}X$, $K_2 = \sqrt{p_y/3}Y$, $K_3 = \sqrt{p_z/3}Z$. One can recover the BF channel by setting $p_y = p_z = 0$ and the DP channel by setting $p_x = p_y = p_z = p$.

Amplitude damping channel. The picture changes significantly under the AD channel. The Kraus operators of the AD channel can be written as

$$K_0 = \begin{bmatrix}1 & 0\\ 0 & \sqrt{1-\gamma}\end{bmatrix}, \qquad K_1 = \begin{bmatrix}0 & \sqrt{\gamma}\\ 0 & 0\end{bmatrix}, \tag{13}$$

with γ the amplitude damping probability.

The action of single-qubit AD channel shrinks the Bloch sphere towards the ground state ($|0\rangle$), creating an asymmetry on the Hilbert space along the z direction. Another common way of describing the noise channels is through the *Pauli transfer matrix* (PTM) formalism [34]. This simplifies some computations and is used in this work. We refer the reader to Appendix A 2 for more details on the PTM formalism.

With these definitions, we can now describe the action of noise on the quantum circuit. Let us consider a quantum system with initial state ρ_0 and at every layer the circuit acts with unitary U_i, such that $\rho_i = \mathcal{U}_i(\rho_{i-1}) = U_i\rho_{i-1}U_i^\dagger$. Then, the quantum state, after layer d, becomes

$$\rho_d = \mathcal{N}\circ\mathcal{U}_d\circ\cdots\circ\mathcal{N}\circ\mathcal{U}_2\circ\mathcal{N}\circ\mathcal{U}_1(\rho_0). \tag{14}$$

This can be visualized as the circuit picture in Fig. 1, where Λ is the local action of the noise channel $\mathcal{N}$. On the real hardware, the action of Λ is different for all qubits, but, for simplicity, we assume that they are the same for simulations.

An extended description of the noise channels can be found in Appendix A.

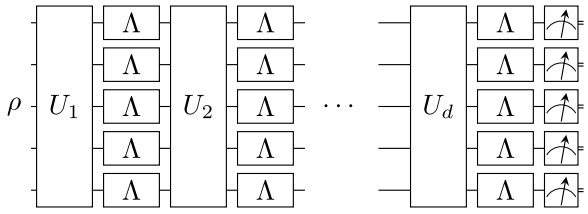


FIG. 1. Schematic of the local noise model. A circuit with input ρ and layers U_i, where the local Λ representing the action of noise are applied after each layer.

III. EQUIVARIANCE UNDER NOISE

Writing down analytical expressions for noisy quantum circuits is a difficult task in general. The expressions are unique to each circuit and noise model, resulting in complicated equations with just a few layers of gates. Nonetheless, this offers a good understanding of the behavior of the model in simple settings. To be able to do this, we construct a toy model. This allows us to build a theoretical understanding and give us an intuition of what to expect from numerical results.

A. Toy model

Let us consider the following circuit, where the one-dimensional input data $x \in \mathbb{R}$ are encoded using the R_Y rotation gate. Then, we apply an identity gate that we decompose into the form $UU^\dagger$, d times. This formulation will allow us to incorporate the effects of gate decompositions on the behavior of the circuit. When designing algorithms in the NISQ era, we should keep in mind that we only have access to a limited set of native gates on hardware. The noise channel Λ will be applied between each U and $U^\dagger$ gates, as described in Fig. 2.

We consider a dataset with the $\mathcal{Z}_2 = \{e, \sigma\}$ symmetry, such that $R(e) \cdot x = x$ and $R(\sigma) \cdot x = -x$. Then, one can use any rotation gate R_G, such that the twirl with representation $R_q(\sigma)$ is $R_q(\sigma) G R_q^\dagger(\sigma) = -G$. Then, one can use the R_Y rotation gate to encode this symmetry simply due to the fact that $XYX = -Y$ and, similarly, $ZYZ = -Y$. This means that we have the freedom to choose either X or Z as representation $R_q(\sigma)$. The choice of representation is also going to put a constraint on the input state. For this walk-through, let us choose the input state $|\psi\rangle = |+\rangle$, $R_q(\sigma) = X$, and $U = R_Y(\theta)$, and the choice of representation requires us to have the observable $O = X$. Here, we refer the reader to Refs. [15,19,28] for more details on constructing EQNNs.

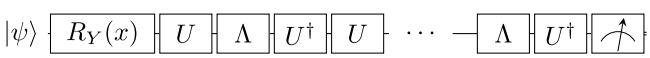


FIG. 2. One qubit toy model under noise with identity gates decomposed into unitaries U and $U^\dagger$, d times, i.e., $I = (UU^\dagger)^d$.

Having defined our complete model, we can now choose a noise model and express the model outputs analytically. We refer the reader to Appendix B for step-by-step calculations in this section. First, we consider the Pauli channel. Then, the output of the model can be written as

$$\hat{y}(x) = \tfrac{1}{2}[({f_x}^d + {f_z}^d)\cos(x) + ({f_x}^d - {f_y}^d)\cos(x + 2\theta)], \tag{15}$$

where f_i is the Pauli fidelity of the Pauli-σ_i channel [e.g., $f_x = 1 - 2(p_y + p_z)$ according to the definition in Sec. II B; see Appendix A for more details]. The first term of the equation gives us the noiseless outcome that is suppressed exponentially in the number of layers around zero (i.e., $\{|f_x|, |f_y|, |f_z|\} \leq 1$). This result is also known as noise-induced barren plateaus [21]. The second term of the equation constitutes the motivation of this work. We see that this term breaks the equivariance for some values. On real hardware, it is often observed that the Pauli fidelities have similar values to one another for a chosen qubit [35]. Therefore, on a realistic setting it is expected to have $f_x \simeq f_y$ and this would make the impact of this term minimal. Consequently, for Pauli channels, which are *depolarizinglike* ($f_x \simeq f_y \simeq f_z$), the symmetry breaking is minimal.

Last but not least, the value of θ also plays a role in the amount of symmetry breaking. It may, in fact, make the symmetry breaking zero regardless of the values of f_x and f_y. This is a natural result, as there will be decompositions that improve robustness against noise. However, such decompositions of gates may not be available on hardware, and one should keep this in mind during the transpilation process.

Now, let us consider the nonunital AD channel with probability γ. Then, the outcome of the circuit can be written as

$$\begin{aligned}\hat{y}(x) = {} & \tfrac{1}{2}[(1-\gamma)^{d/2} + (1-\gamma)^d]\cos(x) \\ & + \tfrac{1}{2}[(1-\gamma)^{d/2} - (1-\gamma)^d]\cos(x + 2\theta) \\ & + [1 - (1-\gamma)^d]\sin(\theta).\end{aligned} \tag{16}$$

We observe that the AD channel results in a more complicated form. The first and second terms jointly result in the exponential concentration induced by the AD channel. This can be easily seen by setting $\theta = 0$. Furthermore, we observe a new term $[\sin(\theta)]$ that shifts the mean of the distribution.

The amplitude of the shift behaves approximately linear ($\propto \gamma d$) for practically relevant depths and noise levels and is upper bounded, i.e., $\mathcal{O}(\gamma d)$. Current superconducting hardware has $\gamma \simeq 10^{-2}$ and a controlled-NOT (CNOT) gate depth of 10–20. Note that the values are approximate and vary from device to device. We provide a brief discussion on how to obtain these values in Appendix H 1. This

behavior can be observed better if we rewrite it using the binomial theorem as

$$1-(1-\gamma)^d=\sum_{k=1}^{d}\binom{d}{k}\gamma^k(-1)^{k+1}. \tag{17}$$

The contribution of the higher-order terms is negligible since γ in general is less than the inverse of d [36]. This means that the term $[1-(1-\gamma)^d]$ will behave linearly in the low-noise regime, where it is expected to get a reasonable signal from the hardware.

We observe that the term responsible for the shift of the mean also corresponds to the only off-diagonal entry in the PTM of the AD channel. We denote this term as $\Lambda^{(d)}_{\mathrm{AD}_{(4,1)}}$. The upper index (d) denotes the dth power of this matrix. We refer the reader to Appendices A 2 and B for details.

Last but not least, we consider the term responsible for symmetry breaking. The term $[(1-\gamma)^{d/2}-(1-\gamma)^d]$ behaves asymptotically similar to the $\Lambda^{(d)}_{\mathrm{AD}_{(4,1)}}$ term, e.g., is approximately linear for the parameters considered above. Furthermore, it is upper bounded by $\mathcal{O}(\gamma d)$, and, thus, the symmetry breaks approximately linearly in the number of layers d or noise strength γ under the AD channel. This can be seen by considering the approximate scaling of $[1-(1-\gamma)^d]\approx\gamma d$. Then, using this, we can see that $[(1-\gamma)^{d/2}-(1-\gamma)^d]\approx\gamma d/2$. We provide further details regarding the scaling in Appendix C.

One final important setting to consider is the combination of the Pauli channel with the AD channel. It is straightforward to compose this effective channel using the PTM picture. We obtain the noisy prediction as

$$\begin{aligned}\hat{y}(x)=&\tfrac{1}{2}[{f_x}^d(1-\gamma)^{d/2}+{f_z}^d(1-\gamma)^d]\cos(x)\\&+\tfrac{1}{2}[{f_x}^d(1-\gamma)^{d/2}-{f_z}^d(1-\gamma)^d]\cos(x+2\theta)\\&+\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}}\sin(\theta),\end{aligned} \tag{18}$$

and the term $\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}}$ reads

$$\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}}\simeq\left(\sum_{k=1}^{d}{f_z}^k\right)\gamma-\left(\sum_{k=1}^{d}(k-1){f_z}^k\right)\gamma^2. \tag{19}$$

This term determines the shift of the mean. We see that it behaves the same except that this time it is modulated with the Pauli fidelity f_z at every layer. Similarly, the amplitude of symmetry breaking depends on the second term as

$$\begin{aligned}&\hat{y}(x)-\hat{y}(-x)\\&\quad=-[{f_x}^d(1-\gamma)^{d/2}-{f_z}^d(1-\gamma)^d]\sin(\theta)\sin(x).\end{aligned} \tag{20}$$

This means that the symmetry breaking is also modulated with the Pauli fidelities f_x and f_z in each layer. Note that we can recover the term for the pure AD channel if we set $f_x=f_z=1$. Overall, the behavior of the term does not change, and it grows approximately linear in the AD channel noise strength γ with minor contributions from the Pauli channel.

The linearity argument can be generalized to multiqubit systems under local noise channel assumptions. In this setting, the contribution of adding new qubits is negligible. This can be seen in Eq. (12), where the action of a local BF channel is given for a two-qubit system. One can observe that the bit flip probability of only the first qubit is $p_0-p_0p_1$ and the bit flip probability of both qubits is p_0p_1, which together sums to p_0. The probability of the bit flip term that acts on all qubits at the same time has a vanishing amplitude in the number of qubits (e.g., $p=\prod_i p_i$ for all $p_i\ll 1$).

Combining all the intuition we have developed so far, we conjecture that a generic EQNN model experiences symmetry breaking dominantly under nonunital channels, and scales linearly in the noise strength γ and depth d. In Sec. IV below, we perform numerical experiments to confirm the implications of the toy model and present evidence directly from hardware experiments. For this purpose, we continue by introducing metrics that can be computed using the simulation and hardware data such that we can decouple the symmetry-breaking terms from the rest of the terms in the model outputs.

B. Quantifying symmetry breaking

Preserving symmetries and quantifying the amount of symmetry are paramount for the success of tasks such as state preparation and time evolution of quantum systems in the presence of hardware noise. In fact, there is a growing literature that studies these aspects [25,26]. Although this may look like a very similar problem in GQML, there is a fundamental difference. In the former, the state belongs to a subspace that is governed by the symmetry of the corresponding system, while in the latter, what matters is the relative positions of the symmetric inputs in the subspace that is governed by the label symmetry. Furthermore, in tasks such as binary classification, the continuous output of a model is mapped to a binary decision based on a threshold. This means that small deviations in the expectation value may not change the binary decision. Overall, these points relax the conditions to preserve the symmetry in the context of GQML. Ragone *et al.* [5] recently introduced $\mathfrak{g}$ *purity*, which can be used to measure the symmetry breaking in GQML, but $\mathfrak{g}$ purity is expensive to compute and does not account for binary decisions. Thus, there is a need to define metrics that can capture all of these aspects.

We start by defining a metric that can use the continuous outputs of a model (i.e., $\hat{y}_i$ for input $\mathbf{x}_i$ [37]). For this purpose, we have to choose the symmetry group. In this paper, we focus on the discrete $\mathcal{Z}_2=\{e,\sigma\}$ symmetry,

such that $R(e)\cdot(\mathbf{x}_i)=(\mathbf{x}_i)$ and $R(\sigma)\cdot(\mathbf{x}_i)=(\mathbf{x}_j)$, where R is the representation of the symmetry group element in the data space $\mathcal{R}$. Then, the equivariance implies that $\hat{y}_i=\hat{y}_j$. We accordingly define the $\mathcal{Z}_2$-symmetry generalized McNemar-Bowker (MB) test [38] as follows.

Definition 4 ($\mathcal{Z}_2$ generalized MB test). Consider the $\mathcal{Z}_2=\{e,\sigma\}$ symmetry, such that $R(e)\cdot(\mathbf{x}_i)=(\mathbf{x}_i)$ and $R(\sigma)\cdot(\mathbf{x}_i)=(\mathbf{x}_j)$. Then, the normalized MB test [38] of a model with predictions $\hat{y}_i$ for input $\mathbf{x}_i$ over M samples can be defined as

$$\chi^2=\frac{1}{M}\sum_{i=1}^{M}\frac{(\hat{y}_i-\hat{y}_j)^2}{\hat{y}_i+\hat{y}_j}. \tag{21}$$

This definition can be further extended to binary predictions. For this purpose, we define the *threshold function* τ, which is a step function that has the transition point t. A naïve choice for the value of t is the center point of the two binary class predictions (e.g., $t=0.5$ if the classes are defined as 0 and 1, $t=0$ if the classes are defined as -1 and 1). However, as we illustrated earlier, the predictions of a model may shift towards a value under hardware noise, and, thus, the central and fixed t value becomes a bad choice. Furthermore, this value is often optimized by following the area under the curve of the receiver operation characteristics of a model [39]. Unsuitably, this makes the choice data dependent. With these points in mind, we choose threshold t such that it is the median of the continuous outputs of a model for the inputs from the training set. This allows us to update the value and account for the shift in the center of the expectation values. Then, we can use the binary predictions $\tau(\hat{y}_i)$ to compute χ^2. We refer to this value as *label misassignment*, as it counts the amount of the predictions that have a different prediction than their $\mathcal{Z}_2$ counterparts.

Definition 5 (Label misassignment). Consider the $\mathcal{Z}_2=\{e,\sigma\}$ symmetry, such that $R(e)\cdot(\mathbf{x}_i)=(\mathbf{x}_i)$ and $R(\sigma)\cdot(\mathbf{x}_i)=(\mathbf{x}_j)$. Let us take a model returning binary predictions $\tau(\hat{y}_i)$, where the $\hat{y}_i$ are the continuous predictions of the model for input $\mathbf{x}_i$ and τ is a step function that has a transition point at the median of all $\hat{y}_i$. Then, label misassignment (LM) of a model over M samples can be defined as

$$\mathrm{LM}=\frac{1}{M}\sum_{i=1}^{M}\frac{[\tau(\hat{y}_i)-\tau(\hat{y}_j)]^2}{\tau(\hat{y}_i)+\tau(\hat{y}_j)}. \tag{22}$$

Note that each term in the sum is either 0 [40] (if the model prediction is the same for $\mathbf{x}_i$ and $\mathbf{x}_j$) or 1 (if the predictions are different). This allows LM to count the amount of misassigned predictions. For example, a model that has perfectly symmetric outputs will be 0% of LM, while a model that produces random outputs 50% of LM. A model that predicts the opposite label for all symmetric inputs will have 100% of LM.

Furthermore, $1-\mathrm{LM}/2$ can be used to upper bound the accuracy of a model. Consider the model that predicts the opposite label each time (i.e., LM = 1.0); this model can have, at best, 50% accuracy. Similarly, a model with random outputs (i.e., LM = 0.5) cannot have an accuracy larger than 75%. Note that $1-\mathrm{LM}/2$ does not predict the accuracy of a model, but only upper bounds it; otherwise, one would expect the completely random model to have 50% accuracy.

Definition 5 assumes a binary classifier, which only admits $\mathcal{Z}_2$ symmetries. This definition can easily be modified to account for multiple classes and other discrete symmetry groups. For instance, the equation can be modified to be a sum over the Dirac delta function over a collection of symmetric data samples. Consider the continuous prediction $\hat{y}_i$ for the data sample x_i that has K many symmetric samples $x_i^{(j)}$, and let $\tau:\mathcal{R}\to\mathcal{Z}$ be the prediction function that takes the continuous prediction as input and returns the integer for a dataset that has L many labels such that $\tau(\hat{y}_i)\in\{1,2,\ldots,L\}$. Then, for a dataset over M total data samples, LM can be computed as

$$\mathrm{LM}=\frac{1}{M}\sum_{i=1}^{M/K}\sum_{j=1}^{K}\delta_{\tau(\hat{y}_i)\tau(\hat{y}_i^{(j)})}. \tag{23}$$

Last but not least, we compute χ^2 and LM values for the toy model presented in Sec. III A. For this purpose, we consider the model under the AD channel [see Eq. (16)] and choose the parameter $\theta=\pi/4$. We consider the input $x\in[-\pi/2,\pi/2]$ and choose 1000 linearly separated values such that we have 500 data points as well as their $\mathcal{Z}_2$ symmetric counterparts. Then, the continuous predictions $[\hat{y}(x)]$ are linearly mapped to be in the range $[0,1]$ [i.e., $(\hat{y}+1)/2$].

Figure 3 presents χ^2 and LM values for various noise strengths (γ) and circuit depths (d). The value of χ^2 increases following the increase in the noise strength or the circuit depth. We observe that the value tends to converge in the high-noise regime. This is due to the concentration of the expectation values. Contrary to χ^2, the values of LM continue to increase even in the high-noise regime. Furthermore, they show a linear scaling with respect to the noise strength and the circuit depth, matching our theoretical insight.

Here, we see that χ^2 fails to capture the symmetry breaking in the high-noise regime. This is because it is not solely measuring the symmetry breaking and is affected by the concentration. LM can successfully do this as it is not biased by the exact value of the continuous predictions. Another thing to note is that χ^2 is biased towards the *0-label*. This can be seen by comparing prediction pairs of

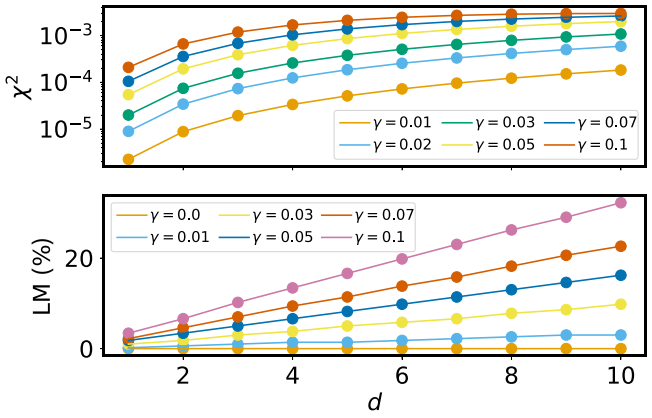


FIG. 3. Symmetry breaking of the toy model considered in Sec. III A with respect to various noise strengths (γ) and circuit depths (d). The output of the model is obtained using Eq. (16) with $\theta = \pi/4$ in the presence of amplitude damping noise.

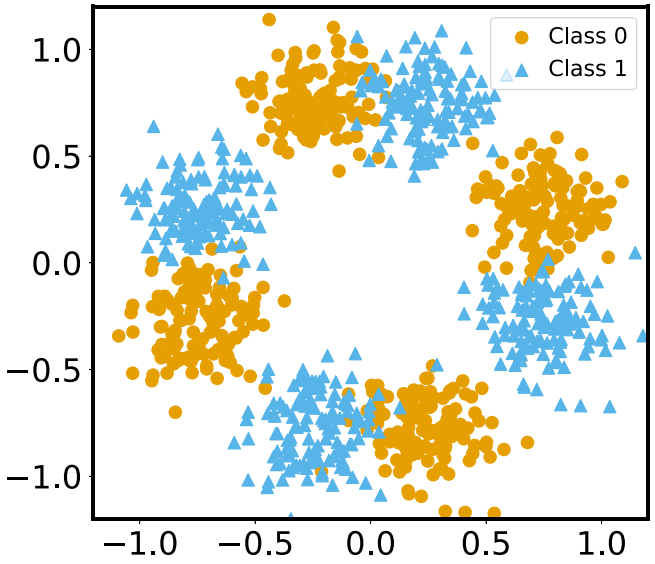


FIG. 4. An ad-hoc dataset with $\mathcal{Z}_2$ label symmetry such that $R(\sigma) \cdot (\mathbf{x}_i) = -\mathbf{x}_i$. The dataset has been generated by sampling 1000 points with equal class split. Although the sampled data points do not have explicit symmetric counterparts explicitly, sampling from the $\mathcal{Z}_2$ symmetric distribution ensures label symmetry. Note that applying $R(\sigma)$ to any of the clusters does not change its class.

{0, 0.2} and {1.0, 0.8}, which have contributions of 0.2 and 0.022, respectively. This makes LM a more suitable measure for symmetry breaking. Nevertheless, χ^2 still informs us about the relationship between symmetry breaking and concentration. Therefore, we provide the measurements for χ^2 in the next sections for completeness.

To complete the analysis of the toy model, we provide χ^2 and LM measurements for the Pauli channel case for realistic scenarios. As expected, the depolarizing channel does not induce symmetry breaking and *depolarizinglike* Pauli channels induce negligible amounts of symmetry breaking. We refer the reader to Appendix D for more details.

IV. EXPERIMENTS

In this section, we provide numerical experiments to validate our findings. To this end, we perform binary classification experiments, and compute χ^2 and LM values that we previously defined in Sec. III B, utilizing both simulated and hardware results.

For the experiments, we consider datasets with $\mathcal{Z}_2$ symmetry as described before. Accordingly, we choose the symmetry transformation such that $R(\sigma) \cdot (\mathbf{x}_i) = -\mathbf{x}_i$. We generate a dataset, as depicted in Fig. 4, that carries this symmetry for the classification experiments. The dataset comprises 1000 samples, divided into training and testing sets with a ratio of 8/2.

As illustrated earlier, the choice of an equivariant data embedding induces a specific unitary representation of the symmetry group element, which will restrict the choices of the parametrized gates and the observable. We define two different two-qubit EQNN models, *EQNN-Z* and *EQNN-XY*, as shown in Figs. 5(a) and 5(b). In both models the data encoding is performed with the Pauli rotation gates R_Y and R_X, inducing the representation $R_q(\sigma) = Z_0 Z_1$. EQNN-*XY* data encoding uses the same gate at each layer, while in the EQNN-*Z* case, the order of R_X and R_Y gates are alternated.

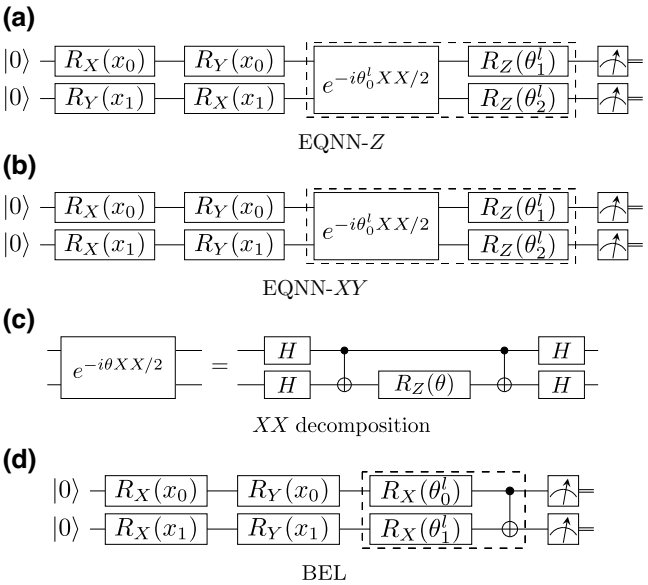


FIG. 5. Two-qubit circuits used in the experiments. The parts in the dashed boxes depict the parametrized layers, which are repeated d times, each having independent parameters. (a) EQNN-Z model, (b) EQNN-XY model, (c) XX decomposition, (d) BEL.

The parametrized gates in both cases are the same. We select three generators $G \in \{X_0X_1, Z_0I_1, I_0Z_1\}$ from the set of commutators of representation Z_0Z_1. These generators are used to obtain parametrized gates of the form $U_G = \exp(-i\theta G/2)$. It is sufficient to use only these three generators, because the nested set of commutators of these three generators is equivalent to the set of commutators of representation Z_0Z_1. The three gates form a parametrized layer and each layer is repeated d times, having independent parameters. Lastly, we choose the equivariant observable $O = (Z_0 + Z_1)/2$ for the EQNN-Z ansatz and $O = X_0Y_1$ for the EQNN-XY ansatz. Expectation values of these observables are linearly mapped to be $\hat{y}(x) \in [0, 1]$ and called the continuous prediction. The same mapping is performed for all models.

Spurious symmetries may arise when building EQNN models. This appears as an unwanted SWAP symmetry (i.e., $x_i^0 \to x_i^1$, $x_i^1 \to x_i^0$) in our example. We handle this in different ways in two models. The EQNN-XY model breaks the unwanted symmetry by employing the observable X_0Y_1, which does not commute with the SWAP gate. In the case of the EQNN-Z model, this is broken at the data encoding level, since the order of the R_X and R_Y gates is alternated. The choice of breaking it at the data encoding level or the measurement level will impact the performance of the model, as we will see later.

Additionally, we define a nonequivariant model, which consists of, namely, *basic entangler layers* (BELs), which does not encompass any symmetrical property from the dataset, as shown in Fig. 5(d). This model is compared to the equivariant one using the same observables. Similar to EQNN, the models with Z and XY observables are defined as *BEL-Z* and *BEL-XY*, respectively.

Last but not least, to model the effect of noise for EQNN circuits, we decompose the $\exp(-\theta XX/2)$ gate using a CNOT-based decomposition, as depicted in Fig. 5(c), and apply noisy gates after each layer, as was shown in Fig. 1. Furthermore, we simulate the EQNN-Z circuit without any decomposition to discern the noise effect, and we refer to this experiment as *EQNN-Z native*.

A. Binary classification

Numerical experiments for classification are conducted using two-qubit circuits, described previously. To compare the accuracy of the model under different noise channels, we run all circuits up to ten layers for a given noise strength and plot the value of the best-performing layer averaged over ten runs. This is to find the best-case scenario for each model as each model will have different effective depth and experience noise differently for a given number of layers d. The binary cross-entropy loss function was minimized using the Adam optimizer [41] with the hyperparameters $\{lr = 0.02, \beta_1 = 0.7, \beta_2 = 0.99\}$. The hyperparameters have not been optimized for each case. The results may be improved in all cases by performing a dedicated hyperparameter optimization. The simulations are performed in the absence of shot noise using the PYTHON library Pennylane [42].

We showcase the results of models trained on the train set for 100 epochs under varying strengths of the DP and AD channels in Figs. 6(a) and 6(b), respectively, illustrating the accuracy assessed on the separate test set. We have not observed overfitting (see Appendix G). In the absence of noise, all models can show more than 90% accuracy. We see a discrepancy between the EQNN-XY and EQNN-Z models. This is due to the location of the spurious symmetry breaking we mentioned earlier. Since the EQNN-Z model breaks this spurious symmetry at the data encoding level, it is more expressive and, hence, can perform better.

In the case of the DP channel, all models experience a similar performance drop. This is a natural outcome of the gradients getting smaller as the noise strength increases due to the emergence of noise-induced barren plateaus.

When considering the AD channel, the performance drops more significantly, characterized by a sharper decline in accuracy. In particular, the BEL-Z, EQNN-Z, and EQNN-XY models demonstrate more pronounced effects compared to other models, while the BEL-XY model performs the best among the four models. There are two reasons for this. The first reason is the symmetry breaking, which impacts both EQNN models. This effect

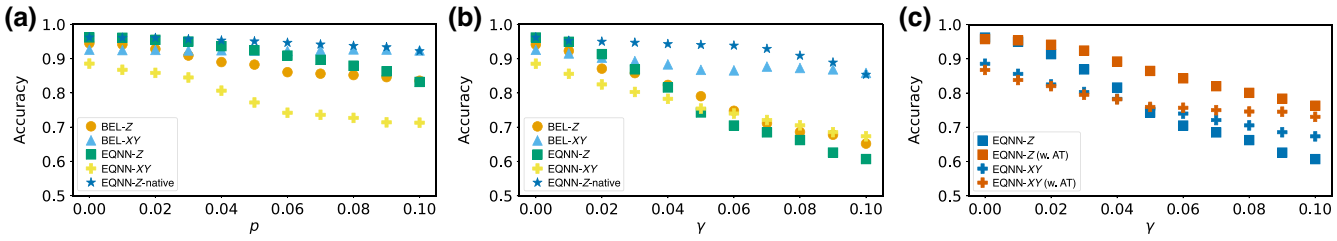


FIG. 6. Binary classification results under noise channels. All models are trained with ten different initializations and layers varied from 1–10. The test accuracy, averaged over the runs, is plotted for the best-performing layer of the corresponding model. Noise strength p in the case of the DP channel and γ in the case of the AD channel is varied from 0.0 to 0.1 with 0.01 increments. (a) Results under the DP channel. (b) Results under the AD channel. (c) Results under the AD channel with and without using adaptive thresholding (AT) during training.

can be observed better when we compare the EQNN-Z-native and EQNN-Z models. Our intuition from the toy model was that the symmetry breaking should be observed in the case of the AD channel and not in the DP channel. We observe that, under the DP channel, these models perform more similarly than they do under the AD channel. Since the EQNN-Z-native model results in a shorter depth, it is also expected to perform better under the DP channel.

The second reason is the shift of the mean for the Z observable under the AD channel. This results in the model having a bias towards one label, when a fixed threshold function is used. To alleviate the effects of the shift of the mean, a simple practical trick called adaptive thresholding is employed. Using prior knowledge on the dataset labels (e.g., a balanced dataset has equal amounts of both classes), one can adaptively change the prediction threshold throughout training. The threshold value can be computed as the median over the predictions of the training set at every iteration. Our results depicted in Fig. 6(c) indicate significant improvement in model performance, particularly in the case when measurements are affected asymmetrically in the z direction. Consequently, this improvement would not be limited to equivariant models. This result shows that adaptive thresholding is a useful and cheap technique to improve model performance for binary classification under hardware noise.

One final point worth noting is the exceptional performance of the EQNN-Z-native model. It consistently outperforms all other models under both the DP and AD channels. The impact of the DP channel on both equivariant and nonequivariant models is expected to be similar. However, what stands out is that the EQNN-Z-native model shows no significant performance drop under the AD channel. This resilience is attributed to the specific choice of the Z_0Z_1 representation, which commutes with the AD channel (cf. Appendix A for details). Despite its impressive performance, it is important to note that the model faces implementation challenges on current quantum hardware due to limitations in the native gate set.

B. Symmetry breaking

In this section, we quantify the level of symmetry breaking across the entire parameter space in the presence of noise. To do so, the model parameters are randomly drawn from the uniform distribution between $[-\pi/2, \pi/2]$ at each run. Similarly, we sample $M = 200$ [see Eq. (21)] unique inputs from the same uniform distribution. Using these samples, we obtain 400 predictions for each setting such that each unique data point has its $\mathcal{Z}_2$ counterpart included. Here, we emphasize that one does not need a trained model to measure the symmetry breaking. Since the amount of symmetry breaking is parameter dependent, we integrate over the input and parameter space to give an average-case estimate of the symmetry breaking.

1. Two-qubit case

In order to explain the discrepancy of performance in training, we measure the proposed metrics χ^2 and LM using simulated data as well as data collected from superconducting quantum computers.

We start by considering the two-qubit EQNN-XY model and collect predictions with ten random initializations for 400 input data samples under different strengths of the AD channel. We plot the variance of the output predictions, χ^2, and LM averaged over the ten runs for a varying number of layers in Fig. 7. The exponential decay of the variance numerically confirms the existence of noise-induced BPs. The value of χ^2 first increases and then decreases for small values of γ and completely decreases for larger values. This is a joint result of symmetry breaking and noise-induced BPs. The χ^2 metric can measure the symmetry breaking until the exponential concentration dominates the landscape and brings all predictions closer to the same value. In fact, it is upper bounded by the variance. One can use the LM metric to decouple these two effects. LM can measure the symmetry breaking separately since it uses the adaptive threshold t. LM grows linearly in the noise strength γ and the number of layers d. This perfectly matches the analytical expression we obtained in Eq. (16) and gives numerical evidence for our linear symmetry-breaking conjecture.

Using this result, we can also comment on the binary classification performance. From the bottom right panel of Fig. 7, we see that LM reaches 20% in the shortest depth scenario. We can use this line to compare the performance of the EQNN-XY model. As mentioned earlier, LM upper bounds the accuracy with $1 - \mathrm{LM}/2$. Based on this, we can say that at $\gamma = 0.1$ the EQNN-XY model should experience a 10% drop in accuracy, only caused by symmetry

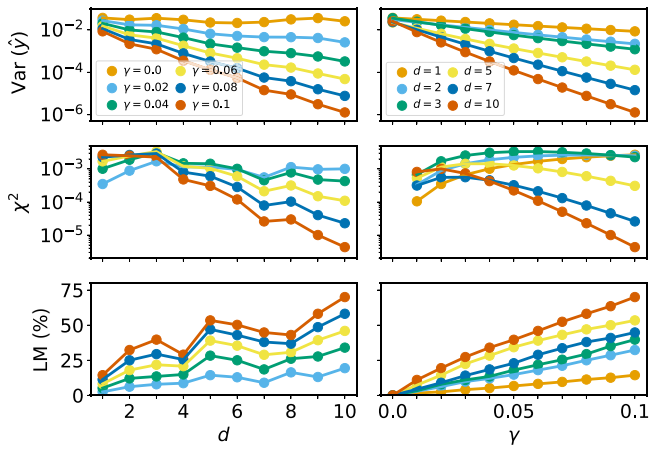


FIG. 7. Simulated two-qubit symmetry breaking for the EQNN-XY model under the AD channel. Both columns show the same data points: on the left metrics are plotted against the number of layers d; on the right metrics are plotted against the noise strength γ.

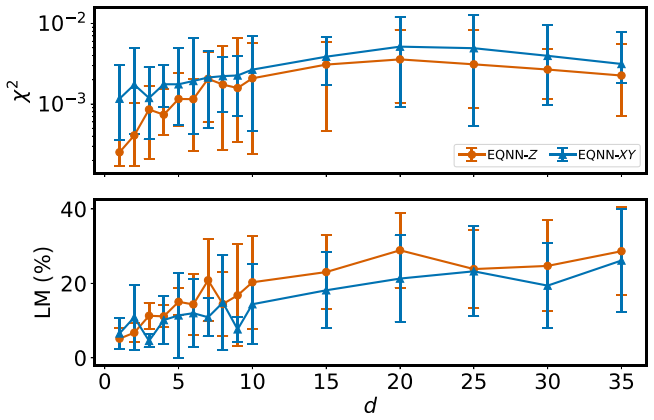


FIG. 8. Two-qubit symmetry breaking for the EQNN-*XY* and EQNN-*Z* models measured on the *ibmq_cairo* superconducting quantum computer. Plots of χ^2 (top) and LM (bottom) against the number of layers d.

breaking. It is difficult to comment on the impact of a single factor, as there are many factors contributing to the drop in performance in the presence of noise. Nonetheless, looking at Fig. 6, this value appears reasonable. We provide an extended discussion of symmetry breaking before and after training in Appendix G.

Next, we repeat this experiment on the *ibmq_cairo* superconducting quantum computer using the EQNN-*Z* and EQNN-*XY* models with 4000 shots. For this purpose, we use the same dataset and the same parameters for the circuits. We report χ^2 and LM values for the number of layers up to 35 in Fig. 8. These results show that both models behave similarly, matching the numerical simulations that were conducted only using the AD channel. This confirms our prediction of the fact that the AD channel dominantly contributes to the symmetry breaking for this setting.

There is a discrepancy between the χ^2 and LM values of the two models. In the case of χ^2, both models observe the increase and then later the decrease due to concentration. However, it is not enough to look at the value of χ^2 to comment on the amount of symmetry breaking. This is because the scale of this metric is controlled by the variance of the observable, and one should keep this in mind when comparing observables with different variances. Next, looking at the LM plot, we see that the EQNN-*Z* model, in general, suffers more symmetry breaking compared to the EQNN-*XY* model. This is mainly due to the fact that the z direction is asymmetric in the AD channel. This result also agrees with Fig. 8, in which EQNN-*Z* performance deteriorates faster. Furthermore, we observe that LM behaves linearly in the number of layers while approaching 50%. The LM values this time converge to 50% since we have shot noise, and the output becomes completely random at a large depth. All of these results combined align with the predictions of the AD channel dominating the symmetry breaking.

2. Multiqubit case

So far, we have considered only the two-qubit case in our experiments, yet our primary interest revolves around the behavior of symmetry breaking at a large scale. Performing simulations on a larger scale imposes significant challenges, becoming computationally expensive. In this section, we focus on obtaining empirical results from the 127-qubit *ibmq_cusco* superconducting chip. For this purpose, we use the nearest-neighbor qubits, as shown in Appendix H 2.

In order to run experiments on hardware, we define a hardware-efficient multiqubit circuit, *EQNN HWE*, illustrated in Fig. 9. Data encoding is performed using R_X and R_Y gates. This results in representation $Z^{\otimes n}$, similar to all other ansatzes we studied so far. A hardware-efficient brick-work layer, constructed from $\exp(-\theta XX/2)$ gates, followed by R_Z gates, is repeated d times. Notably, observables are measured on central qubits to maximize the amount of gates captured by the light cone. Our experiments include probing observables with varying bodyness: $\{Z, XY, XYZ, XYZZ\}$. Note that all the observables commute with representation $Z^{\otimes n}$ to ensure equivariance. Figure 10 presents the χ^2 and LM values obtained for log-depth ($d = \log_2 n$) circuits with varying observables.

Results obtained for χ^2 highlight disparities in the bodyness of the observables. As the locality of the observable increases, the measured expectation values demonstrate a significantly accelerated concentration, leading to a decrease in χ^2. This is expected as the locality of an observable is directly related to the variance of an observable [5,43] in general. We note that there are exceptions to this statement in the literature [44]. Furthermore, the trend for χ^2 with respect to the number of qubits aligns with the two-qubit models that were simulated using only the AD channel. The behavior of LM is consistent with prior findings, showcasing that a log-depth equivariant circuit approaches almost 50% in LM starting from $n = 8$ qubits, corresponding to random outcomes.

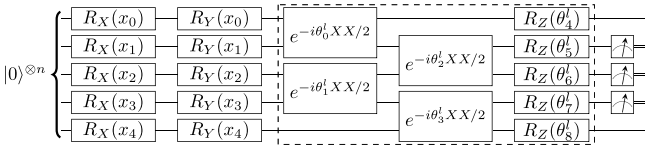


FIG. 9. Hardware-efficient circuit used in the experiments to measure symmetry breaking. The part inside the dashed box is repeated d times with different parameters. Here, a five-qubit circuit is plotted for reference. The measurements are always performed on the central qubits and the amount of qubits measured depends on the observable choice (e.g., two qubits are measured for *XY*). This model is denoted EQNN HWE.

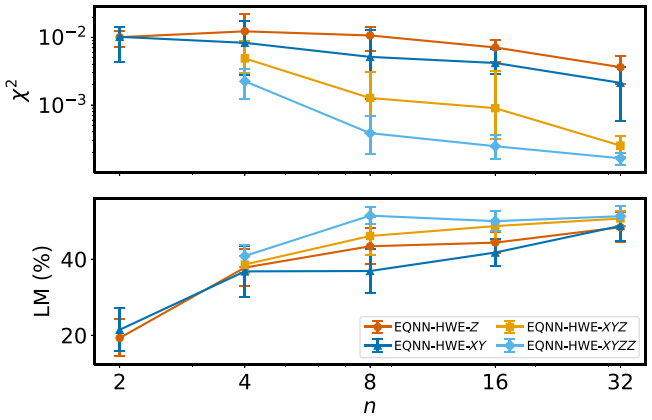


FIG. 10. Log-depth EQNN-HWE results from *ibmq_cusco*. Hardware-efficient circuits defined in Fig. 9 with the number of layers $d = \log_2 n$, where n is the number of qubits. Each model uses a different observable denoted in the legend. The x axis is plotted on a log scale, such that it is linear in the number of layers.

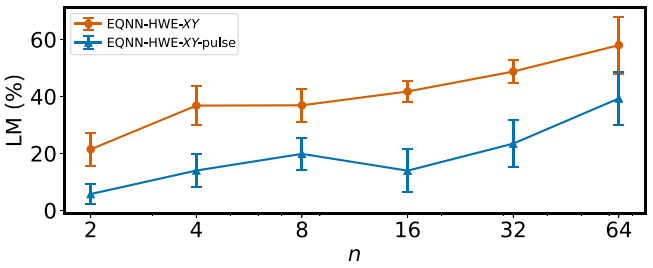


FIG. 11. Label misassignment of the EQNN-HWE-*XY* model using different transpilation methods. The label EQNN-HWE-*XY* refers to the standard transpilation used throughout this work. The EQNN-HWE-*XY* pulse refers to the pulse-efficient transpilation [46].

These results indicate that log-depth EQNN models are not scalable on this hardware due to the combination of concentration and symmetry breaking. This should not be surprising since there is always a cutoff depth for reasonable output on noisy devices. Although this cutoff depth does not look very promising, it can be further improved with various methods.

Pulse-efficient implementation is one of the possible methods to improve the results at the hardware level. The default IBM Qiskit [45] transpilation only exposes fixed pulse gates, such as the calibrated CNOT gate, or echoed cross-resonance (*ECR*) gate, which is equivalent to the CNOT gate up to single-qubit prerotations [46,47]. Thus, any two-qubit gates are decomposed into a decomposition of CNOT and ECR gates and single-qubit gates. Although not ideal, this way of automated transpilation is less time consuming and is a favorable application-agnostic approach. However, these fixed pulse gates have relatively long gate times for low entangling angles, and thus lead to large errors. Thus, in order to improve the hardware result, it is possible to create $R_{ZX}(\theta)$ gates by controlling pulses in a continuous way, instead of using the fixed pulse gates.

Following Earnest *et al.* [46], we use the pulse-efficient implementation where the two-qubit quantum gates are decomposed into the hardware-native *RZX* gates. This allows us to implement the same circuit almost twice as fast using arbitrary parameterization of the pulse control. To show the effectiveness of this approach, we repeat the same experiment with the EQNN-HWE-*XY* model using this scheme and report the results in Fig. 11. As expected from our linearity argument, the symmetry breaking reduces to half of the previous experiment since twice faster execution can be thought of as half the AD channel strength. We refer the reader to Appendix H 3 for more details on the pulse-efficient execution.

V. CONCLUSION

In this work, we studied the behavior of EQNN models in the presence of noise. We highlight that these models experience symmetry breaking in the presence of realistic hardware noise. This adds another noise-induced complication to EQNN models, while the major one is noise-induced BPs [21] and the hardness of learning caused by it [22]. Notably, we demonstrated that the impact of Pauli channels on symmetry breaking could be negligible, while the AD channel induces a symmetry breaking that is linear in the number of layers and noise strength. This further enables predicting the performance of an EQNN model on hardware prior to execution.

Our numerical results highlight the fact that a simple model with no geometric priors can outperform the EQNN models in the presence of noise. This brings doubts to the practicality of GQML and their applicability on near-term devices.

To address these challenges, we proposed effective strategies for mitigating performance drops caused by hardware noise. The first of these was adaptive thresholding that can cope with the shift of the mean. Furthermore, we showed that choosing the $Z^{\otimes n}$ representation is beneficial since it commutes with the AD channel. While our focus was on the $\mathcal{Z}_2$ symmetry for simplicity, our conclusions can be extended to other discrete symmetry groups. However, the implications for continuous groups remain uncertain and this makes it an interesting future research direction. Moreover, we demonstrated that more efficient hardware implementation can contribute to reducing symmetry breaking.

The symmetry protection under the Pauli channel result raises the question of employing Pauli twirling to convert nonunital noise channels to Pauli channels [35]. However, the scalability of the amount of twirls to preserve equivariance remains unclear, posing an open question for future exploration.

In our experiments, we have not considered error-mitigation methods. This was an intentional choice. Our target in this manuscript was to investigate the scalability of GQML on hardware, rather than just being able to execute circuits. This means that error-mitigation methods such as *probabilistic error cancelation* (PEC) are not suitable for this study due to their exponential overhead [35]. Furthermore, a naïve implementation of PEC may result in further loss of equivariance. This opens up new avenues to explore whether we can perform PEC by preserving given group symmetries. Additionally, we briefly explore the potential of *zero noise extrapolation* (ZNE) in Appendix F, revealing its effectiveness when provided with analytical expectation values, but highlighting challenges with a limited number of shots.

We would also like to point out the fact that artificially added symmetry breaking can be beneficial when it is used in moderation. Le *et al.* [17] recently showed that the performance can be improved by injecting a controlled amount of symmetry breaking by adding an extra nonequivariant gate to the equivariant ansatz. However, the study was performed in the absence of noise and, thus, may not be practical in the presence of noise on real quantum hardware.

In conclusion, our study not only advances our understanding of the intricate interplay between hardware noise and GQML models, but also lays the groundwork for informed strategies to enhance their resilience. As we navigate the challenges posed by noise in QML, our findings open new avenues for further exploration and optimization, offering a promising trajectory for the future development of robust and scalable GQML on quantum hardware.

ACKNOWLEDGMENTS

C.T. is supported in part by the Helmholtz Association -"Innopool Project Variational Quantum Computer Simulations (VQCS)". S.C. is supported by the quantum computing for earth observation (QC4EO) initiative of ESA Φ-lab, partially funded under Grant No. 4000135723/21/I-DT-lr, in the FutureEO program. S.C. and M.G. are supported by CERN through the CERN Quantum Technology Initiative. This work is supported with funds from the Ministry of Science, Research and Culture of the State of Brandenburg within the Centre for Quantum Technologies and Applications (CQTA). This work is funded within the framework of QUEST by the European Union's Horizon Europe Framework Programme (HORIZON) under the ERA Chair scheme with Grant Agreement No. 101087126. Access to the IBM Quantum Services was obtained through the IBM Quantum Innovation Centers at CERN and at DESY CQTA.

The authors would like to thank Stefan Kühn, Tobias Hartung, and Marco Cerezo for fruitful discussions. C.T. thanks Daniel J. Egger for his help with the pulse-efficient experiments. The views expressed here are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

APPENDIX A: NOISE MODELS

1. Amplitude damping channel

In Sec. II B, we introduced the AD channel with the Kraus operators

$$K_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \qquad K_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}. \tag{A1}$$

These were given as matrices. Here, we also give them in the Pauli basis:

$$K_0 = \frac{1+\sqrt{1-\gamma}}{2} I + \frac{1-\sqrt{1-\gamma}}{2} Z,$$
$$K_1 = \frac{\sqrt{\gamma}}{2} X - i\frac{\sqrt{\gamma}}{2} Y. \tag{A2}$$

This allows us to see the commutation of the AD channel with the Z gate.

2. Pauli transfer matrix formalism

Working with the Kraus operators can become messy very quickly. The PTM formalism allows us to simplify this process [34]. In this formalism, we start by choosing the normalized Pauli basis $\hat{\mathbb{P}} = \{I, X, Y, Z\}/\sqrt{2}$. Then, the n-qubit operator $\hat{P} \in \hat{\mathbb{P}}^{\otimes n}$ can be represented as a basis vector $|P\rangle\rangle \in \mathbb{R}^{4^n}$.

We can also write the density matrix of a quantum state using this formalism. Consider the state $|\psi\rangle = |0\rangle$, which has the density matrix $\rho = |\psi\rangle\langle\psi| = |0\rangle\langle 0|$. The density matrix ρ can be simply written as $[1/\sqrt{2}, 0, 0, 1/\sqrt{2}]$. This can easily be seen when $|0\rangle\langle 0|$ is explicitly written as $(I + Z)/2$.

Following this, a quantum channel $\mathcal{E} \in \mathbb{R}^{4^n \times 4^n}$ becomes a matrix. Finally, the expectation value of the operator on the density matrix is simply $\mathrm{tr}(\rho\hat{P})$. Then, using the PTM formalism, we can compute the adjoint action of the unitaries as well as the noise channels as simple matrix multiplications.

Now, let us recall that the Kraus operators of the Pauli channel $\mathcal{N}_{\mathrm{P}}$ are given as $K_0 = \sqrt{1-p_x-p_y-p_z}I$, $K_1 = \sqrt{p_x}X$, $K_2 = \sqrt{p_y}Y$, $K_3 = \sqrt{p_z}Z$. To obtain the PTM matrix of the Pauli channel, we can write the action of the channel on all Pauli operators and perform state tomography. This will be fairly simple in this case:

$$\mathcal{N}_{\mathrm{P}}(I) = I, \tag{A3}$$

$$\mathcal{N}_{\mathrm{P}}(X) = [1-2(p_y+p_z)]X, \tag{A4}$$

$$\mathcal{N}_{\mathrm{P}}(Y) = [1-2(p_x+p_y)]Y, \tag{A5}$$

$$\mathcal{N}_{\mathrm{P}}(Z) = [1-2(p_x+p_y)]Z. \tag{A6}$$

We define the Pauli fidelity f_{P} of a Pauli operator P as the coefficient we observe in front [e.g., $f_x = 1-2(p_y+p_z)$]. Then, the PTM of the Pauli channel becomes

$$\Lambda_{\mathrm{P}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & f_x & 0 & 0 \\ 0 & 0 & f_y & 0 \\ 0 & 0 & 0 & f_z \end{bmatrix}. \tag{A7}$$

Following this, we can recover the BF, *phase flip* (PF), DP channels' Kraus operators and the corresponding PTMs.

The BF channel with probability p becomes $K_0 = \sqrt{1-p}I$, $K_1 = \sqrt{p}X$. Then its PTM reads

$$\Lambda_{\mathrm{BF}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1-2p & 0 \\ 0 & 0 & 0 & 1-2p \end{bmatrix}. \tag{A8}$$

The PF channel with probability p becomes $K_0 = \sqrt{1-p}I$, $K_1 = \sqrt{p}Z$. Then its PTM reads

$$\Lambda_{\mathrm{PF}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1-2p & 0 & 0 \\ 0 & 0 & 1-2p & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A9}$$

The DP channel with probability p becomes $K_0 = \sqrt{1-p}I$, $K_1 = \sqrt{p/3}X$, $K_2 = \sqrt{p/3}Y$, $K_3 = \sqrt{p/3}Z$. Then its PTM reads

$$\Lambda_{\mathrm{DP}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1-2p/3 & 0 & 0 \\ 0 & 0 & 1-2p/3 & 0 \\ 0 & 0 & 0 & 1-2p/3 \end{bmatrix}. \tag{A10}$$

The PTM of the AD channel can also be obtained following the same procedure. Here we skip this step and directly give the matrix:

$$\Lambda_{\mathrm{AD}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-\gamma} & 0 & 0 \\ 0 & 0 & \sqrt{1-\gamma} & 0 \\ \gamma & 0 & 0 & 1-\gamma \end{bmatrix}. \tag{A11}$$

Finally, we can use the PTM formalism to show the commutation of the Pauli-Z operator with the AD channel. Recall that we need to satisfy the following for the commutation:

$$\mathcal{N}_{\mathrm{AD}} \circ \mathrm{Ad}_Z(\cdot) = \mathrm{Ad}_Z \circ \mathcal{N}_{\mathrm{AD}}(\cdot). \tag{A12}$$

Then, it is easy to show this using the PTM formalism:

$$\begin{aligned} &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-\gamma} & 0 & 0 \\ 0 & 0 & \sqrt{1-\gamma} & 0 \\ \gamma & 0 & 0 & 1-\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-\gamma} & 0 & 0 \\ 0 & 0 & \sqrt{1-\gamma} & 0 \\ \gamma & 0 & 0 & 1-\gamma \end{bmatrix}. \end{aligned} \tag{A13}$$

Since we are considering local noise models, the PTM of the n-qubit AD channel can be obtained by taking the nth Kronecker power of the single qubit Λ_{AD}, i.e., it is $\Lambda_{\mathrm{AD}}^{\otimes n}$. Similarly, this also applies to $\mathrm{Ad}_Z(\cdot)$, and as a result, we can conclude that $Z^{\otimes n}$ commutes with the n-qubit AD channel.

APPENDIX B: CALCULATIONS FOR THE TOY MODEL

In this appendix, we give the details for the calculations in Sec. III A. Let us start by recalling the definition of the toy model, which was described in Fig. 2. The data are encoded using the R_Y gate and the redundant computation of $UU^\dagger$ is repeated d times. The input state is chosen to be $|+\rangle$. The noise is modeled by applying the noisy operation between each U and $U^\dagger$ gate. For simplicity, U is chosen to be $R_Y(\theta)$, and the output of the model is considered to be the expectation value of the Pauli-X operator. Then the final state of the model, before measurement, for input data x is given as

$$\begin{aligned} \rho = &\mathrm{Ad}_{R_Y(-\theta)} \circ \mathcal{N} \circ \mathrm{Ad}_{R_Y(\theta)} \circ \mathrm{Ad}_{R_Y(-\theta)} \circ \cdots \circ \mathrm{Ad}_{R_Y(\theta)} \\ &\circ \mathrm{Ad}_{R_Y(-\theta)} \circ \mathcal{N} \circ \mathrm{Ad}_{R_Y(\theta)} \circ \mathrm{Ad}_{R_Y(x)}(|+\rangle\langle+|). \end{aligned} \tag{B1}$$

The terms $\mathrm{Ad}_{R_Y(\theta)}$ and $\mathrm{Ad}_{R_Y(-\theta)}$ that appear next to each other will be the identity. Then, this reduces to

$$\rho = \mathrm{Ad}_{R_Y(-\theta)} \circ \underbrace{\mathcal{N} \circ \cdots \mathcal{N}}_{\text{d times}} \circ \mathrm{Ad}_{R_Y(\theta)} \circ \mathrm{Ad}_{R_Y(x)}(|+\rangle\langle+|). \tag{B2}$$

We can compute this using the PTM of these terms. We already defined the PTM of the noise channels in Appendix A 2. Then, we give the definitions for the remaining terms here. The density matrix of $|+\rangle\langle+|$ can be written as $(I+X)/2$. Then, it can be expressed with the vector $[1/\sqrt{2}, 1/\sqrt{2}, 0, 0]$. The PTM that represents the adjoint action of the $R_Y(\theta)$ gate can be expressed as

$$\mathrm{Ad}_{R_Y(\theta)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}. \tag{B3}$$

Furthermore, we need to point to the fact that repetitive application of the noise channel will appear as the dth power of the PTM matrix of the corresponding noise channel. Finally, the expectation value of X in the PTM picture will correspond to a dot product of the vector $[0, \sqrt{2}, 0, 0]$ with the final state. Then, let us write the full expression to obtain the expectation value under the Pauli channel, as it was given in Eq. (15):

$$\begin{aligned} \hat{y}(x) = &\begin{bmatrix} 0 \\ \sqrt{2} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & 0 & \sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & {f_x}^d & 0 & 0 \\ 0 & 0 & {f_y}^d & 0 \\ 0 & 0 & 0 & {f_z}^d \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \\ &\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(x) & 0 & -\sin(x) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(x) & 0 & \cos(x) \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{B4}$$

After matrix multiplication, one obtains

$$\hat{y}(x) = [({f_x}^d + {f_z}^d)\cos(x) + ({f_x}^d - {f_y}^d)\cos(x + 2\theta)]/2. \tag{B5}$$

Next, we would like to compute the output of the model under the AD channel. The PTM of the dth power of the AD channel results in a different structure, since it is not a diagonal matrix. This matrix can be given as

$$\Lambda_{\mathrm{AD}}^{(d)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (1-\gamma)^{d/2} & 0 & 0 \\ 0 & 0 & (1-\gamma)^{d/2} & 0 \\ [1-(1-\gamma)^d] & 0 & 0 & (1-\gamma)^d \end{bmatrix}. \tag{B6}$$

Then, this can be used to compute the expectation value under the AD channel. Using this, we can obtain the noisy prediction under the AD channel as

$$\begin{aligned} \hat{y}(x) = &\tfrac{1}{2}[(1-\gamma)^d + (1-\gamma)^{d/2}]\cos(x) \\ &+ \tfrac{1}{2}[(1-\gamma)^d - (1-\gamma)^{d/2}]\cos(x+2\theta) \\ &+ [1-(1-\gamma)^d]\sin(\theta). \end{aligned} \tag{B7}$$

Next, we consider the combination of the Pauli channel with the AD channel. In the PTM formalism, their joint action can be represented as a matrix multiplication, such that $\Lambda_{\mathrm{P+AD}} = \Lambda_{\mathrm{P}} \cdot \Lambda_{\mathrm{AD}}$ and it can be written as

$$\Lambda_{\mathrm{P+AD}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & f_x\sqrt{1-\gamma} & 0 & 0 \\ 0 & 0 & f_y\sqrt{1-\gamma} & 0 \\ f_z\gamma & 0 & 0 & f_z(1-\gamma) \end{bmatrix}. \tag{B8}$$

Then, $\Lambda_{\mathrm{P+AD}}$ can be used to calculate the noisy predictions under the joint action of the Pauli and AD channels. This

becomes

$$\hat{y}(x) = \tfrac{1}{2}[{f_x}^d(1-\gamma)^{d/2} + {f_z}^d(1-\gamma)^d]\cos(x) + \tfrac{1}{2}[{f_x}^d(1-\gamma)^{d/2} - {f_z}^d(1-\gamma)^d]\cos(x+2\theta) + \Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}}\sin(\theta), \tag{B9}$$

where the term $\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}}$ corresponds to the matrix element of index (4, 1) and the first two leading terms can be written as

$$\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}} \simeq \left(\sum_{k=1}^{d} {f_z}^k\right)\gamma - \left(\sum_{k=1}^{d}(k-1){f_z}^k\right)\gamma^2. \tag{B10}$$

APPENDIX C: ANALYTICAL SCALING OF SYMMETRY BREAKING

In Sec. III A, we argued that the symmetry breaking term $(1-\gamma)^{d/2} - (1-\gamma)^d$ in the case of the AD channel scales as $\gamma d/2$ for the parameters we considered, e.g., $\gamma \approx 10^{-2}$ and $d \approx 10$–20. Here, we first show the exact analytical expression for this term:

$$\begin{cases} \gamma - \gamma^2 & (d=2), \\ 2\gamma - 5\gamma^2 + 4\gamma^3 - \gamma^4 & (d=4), \\ 3\gamma - 12\gamma^2 + 19\gamma^3 - 15\gamma^4 + 6\gamma^5 - \gamma^6 & (d=6). \end{cases} \tag{C1}$$

Here, it can be seen that the contribution of the higher-order terms are negligible. We visualize this in Fig. 12 by plotting the $(1-\gamma)^{d/2} - (1-\gamma)^d$ along with $\gamma d/2$. It can be seen that this term follows $\gamma d/2$ in the low-noise regime and the behavior does not deviate much from linear even when the noise level is increased.

APPENDIX D: SYMMETRY BREAKING UNDER PAULI CHANNELS

Here, we compute χ^2 and LM values for the toy model presented in Sec. III A. For this purpose, we consider

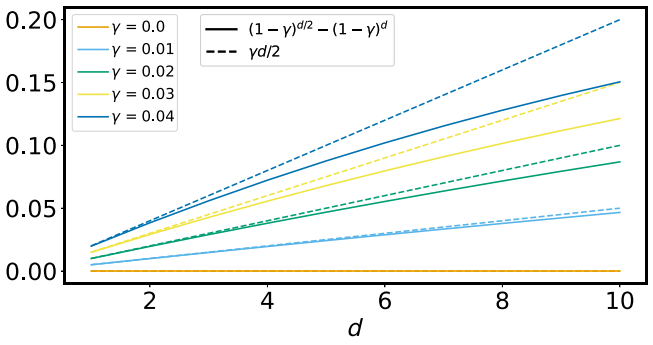


FIG. 12. The scaling of the values of the term responsible for the symmetry breaking in the toy model considered in Sec. III A for the amplitude damping channel.

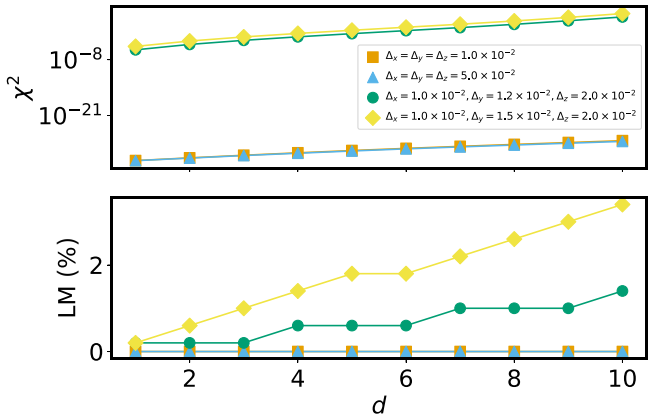


FIG. 13. Symmetry breaking of the toy model considered in Sec. III A with respect to four Pauli channel scenarios. The output of the model is obtained using Eq. (15) with $\theta = \pi/4$ in the presence of Pauli noise. The noise strengths are reported as Pauli infidelities (e.g., $\Delta_x = 1 - f_x$).

the model under the Pauli channel [see Eq. (15)] and choose the parameter $\theta = \pi/4$. We consider the input $x \in [-\pi/2, \pi/2]$ and choose 1000 linearly separated values such that we have 500 data points as well as their $\mathcal{Z}_2$ symmetric counterparts. Then, the continuous predictions $[\hat{y}(x)]$ are linearly mapped to be in the range $[0, 1]$ [i.e., $(\hat{y}+1)/2$].

We consider four cases to depict different scenarios. The first two channels are depolarizing channels with Pauli fidelities ($f_x = f_y = f_z$) 0.99 and 0.95, respectively. The other two cases are deviations from the depolarizing channel. The values here are chosen according to van den Berg *et al.* [35], who *learn* the Pauli fidelities of a superconducting quantum computer, which is a similar device to what has been considered in this work. They reported Pauli infidelities ($1 - f_i = \Delta_i$) to be in the range (1–2) $\times 10^{-2}$.

Figure 13 presents χ^2 and LM values for the four scenarios with respect to increasing circuit depth (d). There is no symmetry breaking in the two depolarizing channel scenarios. We measure nonzero symmetry breaking in the other two cases, although the values appear to be small compared to the AD channel cases (see Fig. 3). This matches our theoretical insight and shows that the contribution of realistic Pauli channels to symmetry breaking is negligible.

Here, we emphasize that there exists Pauli channels that can induce symmetry breaking much larger than what we considered for the AD channel. Our observations and claims following them are based on the noise characteristics of the currently available hardware. According to the specifications of such devices, the AD channel appears as the main contributor to symmetry breaking compared to the *depolarizinglike* Pauli channels.

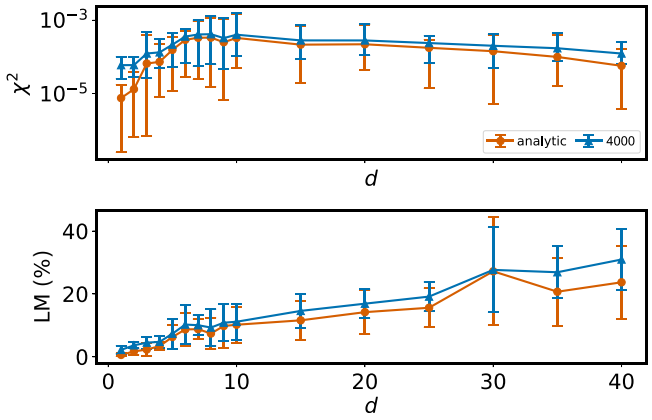


FIG. 14. Comparison of symmetry breaking measurements with (4000 shots) and without shot noise. The EQNN-Z model is simulated under the AD channel with $\gamma = 0.01$.

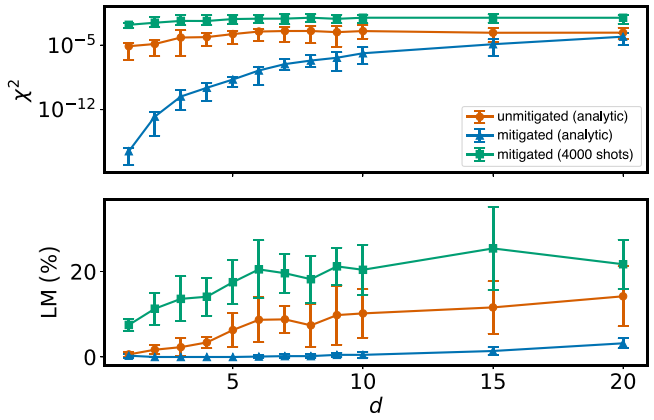


FIG. 15. Symmetry breaking experiments with ZNE in the presence of AD channel noise ($\gamma = 0.01$). The EQNN-Z model used in Sec. IV B 1 is employed with (4000 shots) and without (analytical) shot noise.

APPENDIX E: IMPACT OF SHOT NOISE

All simulations in the main text of the manuscript are performed with analytic expectation values omitting shot noise. All of the hardware runs are performed with 4000 shots. Here in Fig. 14, we present the simulation of the EQNN-Z model simulated with the AD channel using noise strength $\gamma = 0.01$ to show that the number of shots chosen is enough to match analytic results with high confidence.

APPENDIX F: ZERO NOISE EXTRAPOLATION

Zero noise extrapolation is an error-mitigation method that uses the expectation values measured at different noise strengths [48]. These values can be extrapolated to the zero noise level using Richardson's extrapolation method to obtain *noiseless* expectation values.

We perform two separate numerical experiments to compare the effectiveness of ZNE. In the first one, the expectation values are computed analytically, while in the other one, the expectation values are computed using 4000 shots. In both experiments, the base noise level ($\lambda = 1$) is chosen to be $\gamma = 0.01$. Then, the experiments are repeated using increasing levels of $\gamma \in \{0.015, 0.020, 0.025, 0.030\}$. These five expectation values for all noise scale factors are then extrapolated using Richardson's extrapolation to obtain the *noiseless* expectation values. Results for a varying number of layers in the presence of AD channel noise are presented in Fig. 15.

An important side note here is that ZNE may sometimes cause the values to go beyond their allowed values [e.g., $\hat{y}(x) \in [0, 1]$], especially in the cases it fails. We have observed that this is happening often in the case with 4000 shots. To prevent this, we have assigned predictions below zero to be exactly 0.0 and predictions above one to be exactly 1.0 in all cases.

It is clear that ZNE can improve the accuracy of the results and bring LM values down significantly in the analytical case. However, when the number of shots is limited, ZNE fails and even worsens the results. This highlights the fact that ZNE requires many shots to work properly and the number of shots required will inevitably grow exponentially in the number of layers due to noise-induced BPs.

APPENDIX G: DETAILS ON BINARY CLASSIFICATION EXPERIMENTS

In this appendix, we give more details on the binary classification results discussed in Sec. IV A. We start by providing the training curves for one of the settings. Results for the EQNN-Z model with $d = 3$ layers in the presence of various values of AD channel noise are plotted in Fig. 16. There is no clear impact to generalization in

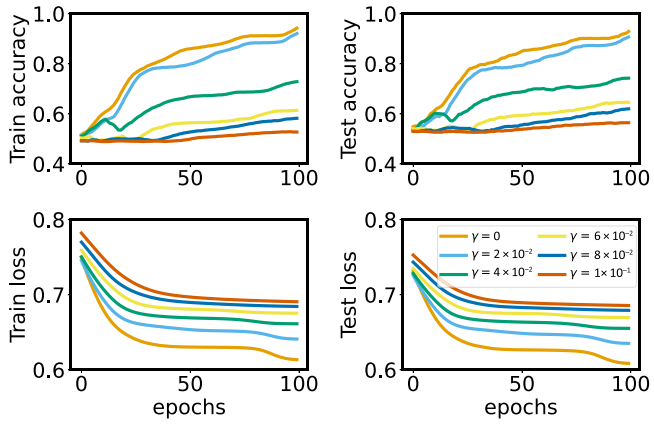


FIG. 16. Training curves of the EQNN-Z model with $d = 3$ layers in the presence of the AD channel.

TABLE I. LM values of the two-qubit EQNN-*XY* model for $d = 2$ before and after training. The values under the column "bef. train." correspond to LM values computed using the ten random parameter sets used to initialize the model. The values under "after training" correspond to values computed for models trained under different values of the AD channel noise strength and re-evaluated at various noise strengths. The LM values are presented as averaged over the ten runs and the variance is reported as the error.

	LM (bef. train.)	LM (after training with γ_t)		
γ	N/A	$\gamma_t = 0.0$	$\gamma_t = 0.01$	$\gamma_t = 0.02$
0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
0.01	2.9 ± 0.2	9.9 ± 0.1	10.9 ± 0.0	10.7 ± 0.1
0.02	6.2 ± 0.4	21.2 ± 1.0	22.7 ± 0.6	22.7 ± 0.5
0.03	9.8 ± 0.6	31.6 ± 2.2	33.5 ± 1.6	33.1 ± 1.5
0.04	12.1 ± 0.9	40.3 ± 3.1	42.4 ± 2.1	41.8 ± 1.6
0.05	14.8 ± 1.2	46.1 ± 2.9	47.9 ± 1.7	49.5 ± 2.2

the presence of noise. The overall performance steadily worsens as the noise level is increased.

Next, we compare the LM measured on the EQNN-*XY* model before and after training, both on noiseless and noisy devices. For this purpose, we consider the EQNN-*XY* model with $d = 2$ layers. Then, we consider ten runs as in previous classification experiments. We train the model in the presence of the AD channel with $\gamma_t \in \{0.0, 0.01, 0.02\}$. After training, we use the trained parameters to evaluate the LM values in the absence and presence of noise with $\gamma \in \{0.0, 0.01, 0.02, 0.03, 0.04, 0.05\}$. We also consider the same measurements on the initial *random* parameters, which are randomly drawn from the uniform distribution $\mathcal{U} \sim [-\pi/2, \pi/2]$. The measured values are presented in Table I.

These results show that the value of LM is larger after training both with and without noise. This points to the fact that the symmetry breaking is independent of the training procedure. The model parameters that result in *good* performance may result in high symmetry breaking. In general, the training procedure is expected to find a reasonable trade-off between the two to maximize the model performance. Furthermore, we observe that the model trained without noise exhibits similar symmetry breaking compared to the models trained in the presence of noise, when LM is measured afterwards at the same level of noise. This indicates that the training procedure is not able to *learn* the noise to be able to account for the symmetry breaking. Noise-aware training procedures may be an interesting avenue to explore in future work.

APPENDIX H: HARDWARE EXPERIMENTS

In this appendix, we give details of the hardware experiments. All experiments are performed with the same settings using 4000 shots and no error-mitigation method is used. The *light optimization* is used to transpile the circuits, which includes the *SABRE* method [49], single-qubit gate optimization, and dynamical decoupling [50]. The list of devices, along with some of their properties, is presented in Table II.

TABLE II. Properties of the physical quantum hardware used in this work. All values are reported as the median across all qubits on the chip. The values may change daily with each calibration.

Name	$T1(\mu s)$	$T2(\mu s)$	Gate time (ns)	Readout length (ns)
ibmq_cairo	91.99	92.4	321.778	732.444
ibmq_cusco	126.78	78.77	460	4000

1. Connecting noise model parameters to hardware specifications

The amplitude damping channel strength can be obtained as [33]

$$\gamma = 1 - e^{(-t/T1)}, \tag{H1}$$

where t is the duration of the circuit. Since we model the noise after each gate, we can assume that the circuit will have total time = 1 gate time + readout time. Then we can use Table II to compute γ for the *ibmq_cairo* as

$$\gamma = 1 - e^{-1.054\,222/91.990} \simeq 0.011. \tag{H2}$$

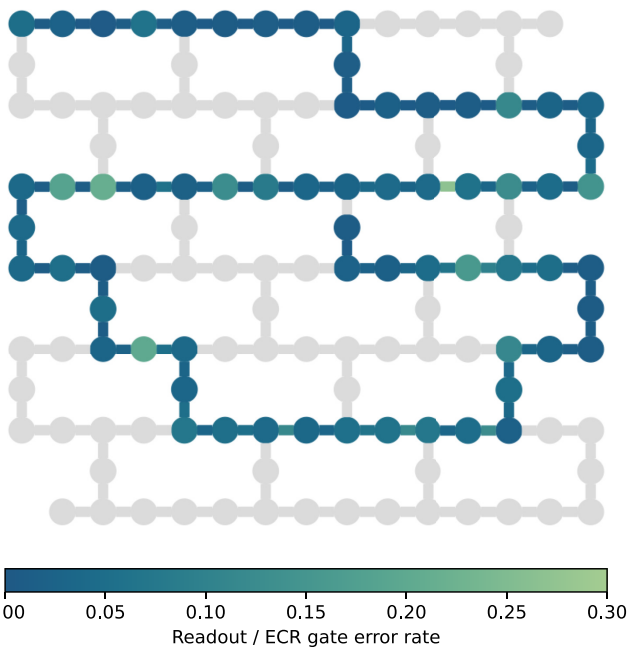


FIG. 17. Coupling map of *ibmq_cusco* and the qubit configuration chosen to run the quantum circuit of 64 qubits. The colors represent the readout error for each qubit and the two-qubit ECR gate for each qubit connection.

```
rzx_basis = ['rzx', 'rz', 'x', 'sx']

pm = PassManager([
    # Consolidate consecutive two-qubit operations.
    Collect2qBlocks(),
    ConsolidateBlocks(basis_gates=['rz', 'sx', 'x', 'rxx']),

    # Rewrite circuit in terms of Weyl-decomposed echoed RZX gates.
    EchoRZXWeylDecomposition(backend),

    # Attach scaled CR pulse schedules to the RZX gates.
    RZXCalibrationBuilderNoEcho(backend),

    # Simplify single-qubit gates.
    UnrollCustomDefinitions(std_eqlib, rzx_basis),
    BasisTranslator(std_eqlib, rzx_basis),
    Optimize1qGatesDecomposition(rzx_basis),
])
```

FIG. 18. PYTHON code to construct the "PassManager" used for *RZX* transpilation in the Qiskit implementation. The code is derived from Ref. [54]. The code is licensed under the Apache License 2.0.

Similarly, if we consider two gates, then we get $\gamma \simeq 0.015$. We can repeat this for *ibmq_cusco*. Then, we obtain $\gamma \simeq 0.035$ and $\gamma \simeq 0.038$ for one and two gates, respectively.

We should still pay attention to the fact that there is a model mismatch in this computation. Amplitude damping acts continuously on a system, while we model it to act discretely after each gate. Furthermore, the noise model assumes that the noise strength is constant and independent of the circuit depth. Nevertheless, we can approximate the value of γ with this method and conclude that it will be of the order of 10^{-2}.

2. Hardware topology

The coupling map of *ibmq_cusco* used for the 64-qubit experiments is presented in Fig. 17. We choose a suitable nearest-neighbor set of qubits to have one-dimensional connectivity.

3. Pulse-efficient transpilation

In order to run a generic quantum circuit on the real IBM Quantum hardware, the circuit should first be transpiled into the set of basis gates, which are precalibrated on the corresponding hardware. The automatic IBM quantum transpilation only exposes precalibrated fixed-frequency cross-resonance gates [51] such as CNOT or ECR gates. Instead of the fixed-frequency gates, we can use the continuous gate native to the quantum hardware. For low rotation angles, the circuit duration becomes shorter using continuous gates, leading to less decoherence noise on the overall circuit and, thus, more accurate results.

The calibrated CNOT gates are built with a Gaussian-Square pulse, which is a flat-top pulse with area

$$\alpha* = \|A^*\|\left[w^* + \sqrt{2\pi}\sigma \cdot \mathrm{erf}\left(\frac{rf}{\sqrt{2}\sigma}\right)\right], \tag{H3}$$

where A^* is the amplitude, w^* is the width, rf is the rise-fall, and σ is the standard deviation of the corresponding Gaussian flanks [46]. The pulse of the $R_{ZX}(\theta)$ gate is created by rescaling the area α as [52]

$$\alpha(\theta) = \frac{2\theta\alpha^*}{\pi}. \tag{H4}$$

In the Qiskit implementation, the *RZX*-based transpilation works as shown in Fig. 18. First of all, we collect all the consecutive two-qubit operations and consolidate them

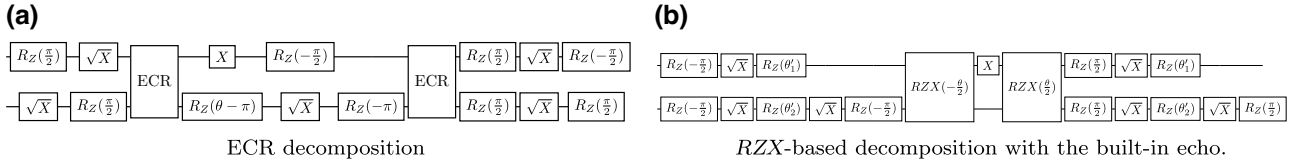
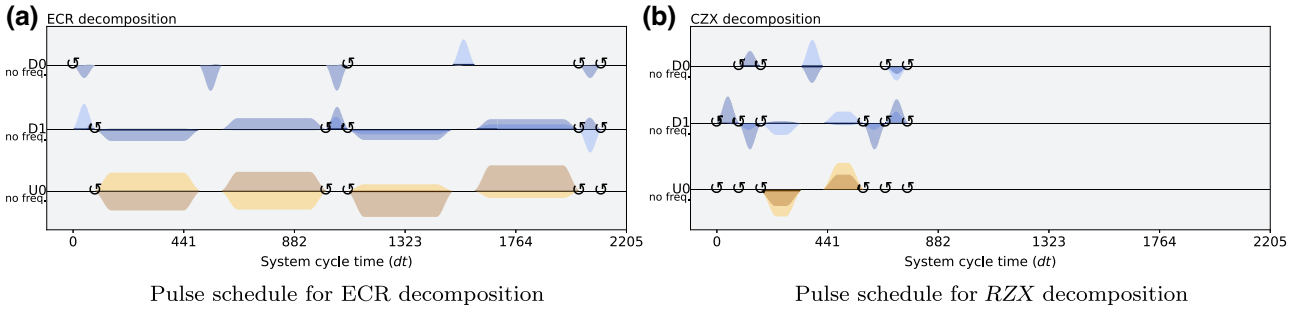


FIG. 19. Circuit decomposition of the $RXX(\theta)$ gate on *ibmq_cusco*. In (b) θ_1' and θ_2' are the single-qubit rotation angles computed by Cartan's decomposition. (a) ECR decomposition, (b) *RZX*-based decomposition with the built-in echo.



Pulse schedule for ECR decomposition

Pulse schedule for *RZX* decomposition

FIG. 20. Pulse schedule for the ECR-based decomposition and pulse-efficient *RZX*-based decompositions (cf. Fig. 19). The symbol ↺ indicates the virtual *Z* gates [55]. (a) Pulse schedule for ECR decomposition. (b) Pulse schedule for *RZX* decomposition.

into a general two-qubit SU(4) operation. Then, the corresponding two-qubit gate is decomposed in terms of the echoed *RZX* gates thanks to Cartan's decomposition [53]. Those gates are calibrated by scaling the Gaussian square pulses of the fixed-frequency CNOT or ECR gates following Eq. (H4). Finally, all the single-qubit gates are simplified and optimized to reduce the total circuit depth.

Figures 19 and 20 display the decomposed circuits of the *RXX* gate for ECR-based and *RZX*-based decomposition using the basis gates on *ibmq_cusco* and the corresponding pulse schedule, respectively. As shown in Fig. 20, the pulse schedule with *RZX* decomposition is much shorter compared to that with ECR decomposition, resulting in less decoherence and better results, as mentioned previously.

[1] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 1 (2018).

[2] 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).

[3] E. Fontana, N. Fitzpatrick, D. M. Ramo, R. Duncan, and I. Rungger, Evaluating the noise resilience of variational quantum algorithms, Phys. Rev. A **104**, 022403 (2021).

[4] 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).

[5] M. Ragone, B. N. Bakalov, F. Sauvage, A. F. Kemper, C. O. Marrero, M. Larocca, and M. Cerezo, A unified theory of barren plateaus for deep parametrized quantum circuits, arXiv:2309.09342.

[6] E. R. Anschuetz and B. T. Kiani, Quantum variational algorithms are swamped with traps, Nat. Commun. **13**, 7760 (2022).

[7] X. You and X. Wu, in *Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research*, Vol. 139, edited by M. Meila and T. Zhang (PMLR, 2021), p. 12144.

[8] J. Rivera-Dean, P. Huembeli, A. Acín, and J. Bowles, Avoiding local minima in variational quantum algorithms with neural networks, arXiv:2104.02955.

[9] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, Quantum **6**, 677 (2022).

[10] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, Quantum **3**, 214 (2019).

[11] T. Volkoff and P. J. Coles, Large gradients via correlation in random parameterized quantum circuits, Quantum Sci. Technol. **6**, 025008 (2021).

[12] S. H. Sack, R. A. Medina, A. A. Michailidis, R. Kueng, and M. Serbyn, Avoiding barren plateaus using classical shadows, PRX Quantum **3**, 020365 (2022).

[13] T. L. Patti, K. Najafi, X. Gao, and S. F. Yelin, Entanglement devised barren plateau mitigation, Phys. Rev. Res. **3**, 033090 (2021).

[14] C. Tüysüz, G. Clemente, A. Crippa, T. Hartung, S. Kühn, and K. Jansen, Classical splitting of parametrized quantum circuits, Quantum Mach. Intell. **5**, 34 (2023).

[15] M. Larocca, F. Sauvage, F. M. Sbahi, G. Verdon, P. J. Coles, and M. Cerezo, Group-invariant quantum machine learning, PRX Quantum **3**, 030341 (2022).

[16] X. Wang, Y. Chai, M. Demidik, X. Feng, K. Jansen, and C. Tüysüz, Symmetry enhanced variational quantum imaginary time evolution, arXiv:2307.13598.

[17] I. N. M. Le, O. Kiss, J. Schuhmacher, I. Tavernelli, and F. Tacchino, Symmetry-invariant quantum machine learning force fields, arXiv:2311.11362.

[18] S. Y. Chang, M. Grossi, B. Le Saux, and S. Vallecorsa, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2023), Vol. 1, p. 229.

[19] J. J. Meyer, M. Mularski, E. Gil-Fuster, A. A. Mele, F. Arzani, A. Wilms, and J. Eisert, Exploiting symmetry in variational quantum machine learning, PRX Quantum **4**, 010328 (2023).

[20] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[21] 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).

[22] A. Nietner, Unifying (quantum) statistical and parametrized (quantum) algorithms, arXiv:2310.17716.

[23] V. Kungurtsev, G. Korpas, J. Marecek, and E. Y. Zhu, Iteration complexity of variational quantum algorithms, arXiv:2209.10615.

[24] A. Abbas *et al.*, Quantum optimization: Potential, challenges, and the path forward, arXiv:2312.02279.

[25] M. C. Tran, Y. Su, D. Carney, and J. M. Taylor, Faster digital quantum simulation by symmetry protection, PRX Quantum **2**, 010323 (2021).

[26] N. H. Nguyen, M. C. Tran, Y. Zhu, A. M. Green, C. H. Alderete, Z. Davoudi, and N. M. Linke, Digital quantum simulation of the Schwinger model and symmetry protection with trapped ions, PRX Quantum **3**, 020324 (2022).

[27] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[28] Q. T. Nguyen, L. Schatzki, P. Braccia, M. Ragone, P. J. Coles, F. Sauvage, M. Larocca, and M. Cerezo, Theory for equivariant quantum neural networks, PRX Quantum **5**, 020328 (2024).

[29] S. Kazi, M. Larocca, and M. Cerezo, On the universality of s_n-equivariant k-body gates, New J. Phys. **26**, 053030 (2024).

[30] M. Ragone, P. Braccia, Q. T. Nguyen, L. Schatzki, P. J. Coles, F. Sauvage, M. Larocca, and M. Cerezo, Representation theory for geometric quantum machine learning, arXiv:2210.07980.

[31] L. Schatzki, M. Larocca, Q. T. Nguyen, F. Sauvage, and M. Cerezo, Theoretical guarantees for permutation-equivariant quantum neural networks, npj Quantum Info. **10**, 1 (2024).

[32] H. Zheng, Z. Li, J. Liu, S. Strelchuk, and R. Kondor, Speeding up learning quantum states through group equivariant convolutional quantum ansätze, PRX Quantum **4**, 020327 (2023).

[33] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).

[34] J. M. Chow, J. M. Gambetta, A. D. Córcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. R. Rozen, M. B. Ketchen, and M. Steffen, Universal quantum gate set approaching fault-tolerant thresholds with superconducting qubits, Phys. Rev. Lett. **109**, 060501 (2012).

[35] E. van den Berg, Z. K. Minev, A. Kandala, and K. Temme, Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors, Nat. Phys. **19**, 1116 (2023).

[36] D. Stilck França and R. García-Patrón, Limitations of optimization algorithms on noisy quantum devices, Nat. Phys. **17**, 1221 (2021).

[37] Bold symbols are used to represent vectors. Here $\mathbf{x}_i$ denotes the ith data sample with arbitrary size.

[38] A. Krampe and S. Kuhnt, Bowker's test for symmetry and modifications within the algebraic framework, Comput. Stat. Data Anal. **51**, 4124 (2007).

[39] A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognit. **30**, 1145 (1997).

[40] We avoid division by zero in the case of zero predictions, by adding a small epsilon to the denominator for the numerical experiments.

[41] D. P. Kingma and J. Ba, in *3rd International Conference on Learning Representations, ICLR 2015* (San Diego, CA, USA, 2015).

[42] V. Bergholm *et al.*, Pennylane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968.

[43] 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).

[44] N. L. Diaz, D. García-Martín, S. Kazi, M. Larocca, and M. Cerezo, Showcasing a barren plateau theory beyond the dynamical Lie algebra, arXiv:2310.11505.

[45] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, Quantum computing with Qiskit, arXiv:2405.08810.

[46] N. Earnest, C. Tornow, and D. J. Egger, Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware, Phys. Rev. Res. **3**, 043088 (2021).

[47] D. J. Egger, C. Capecci, B. Pokharel, P. K. Barkoutsos, L. E. Fischer, L. Guidoni, and I. Tavernelli, Pulse variational quantum eigensolver on cross-resonance-based hardware, Phys. Rev. Res. **5**, 033159 (2023).

[48] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Error mitigation extends the computational reach of a noisy quantum processor, Nature **567**, 491 (2019).

[49] G. Li, Y. Ding, and Y. Xie, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '19* (Association for Computing Machinery, New York, 2019), p. 1001.

[50] L. Viola and S. Lloyd, Dynamical suppression of decoherence in two-state quantum systems, Phys. Rev. A **58**, 2733 (1998).

[51] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and M. Steffen, Simple all-microwave entangling gate for fixed-frequency superconducting qubits, Phys. Rev. Lett. **107**, 080502 (2011).

[52] J. P. T. Stenger, N. T. Bronn, D. J. Egger, and D. Pekker, Simulating the dynamics of braiding of Majorana zero modes using an IBM quantum computer, Phys. Rev. Res. **3**, 033171 (2021).

[53] N. Khaneja and S. J. Glaser, Cartan decomposition of SU(2n) and control of spin systems, Chem. Phys. **267**, 11 (2001).

[54] Qiskit 0.33 release notes, https://docs.quantum.ibm.com/api/qiskit/release-notes/0.33, accessed: 2024-07-05.

[55] 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).