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

Geometric quantum machine learning based on equivariant quantum neural networks (EQNN) recently appeared as a promising direction in quantum machine learning. Despite the encouraging progress, the 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 breaking grows 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.

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.* number of shots needed to estimate the sufficiently precise values of the cost function grows exponentially [4, 5], many local minima [6–8] and lack of efficient gradient computation (*e.g.* parameter shift rules require circuit executions that scale linearly in 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]. The GQML framework leverages inductive biases on problems and uses this to construct algorithms with improved trainability and generalization [16]. This requires the circuit to have a certain structure from the initial state until the final measurements. On the other hand, this is where the NISQ hardware fails to provide due to coherent and incoherent errors present [1, 17]. In the literature, this topic has been explored in the context of state preparation and time evolution of quantum systems, in which many physical symmetries arise [18, 19]. However, these results don't 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 non-unital amplitude damping channel. We characterize this with metrics that we introduce and show that symmetry breaking grows 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 Section II, we introduce the necessary preliminary definitions from how to construct EQNNs, to how the hardware noise is modeled. Then, in Section 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 Section V, we share our point of view on what error mitigation means for the results that we establish, and we conclude by giving suggestions on deploying EQNN models on hardware and talk about 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 $\mathbf{x}_i$ into a quantum state $\rho_{\mathbf{x}_i} \in \mathcal{M}$ with a certain quantum feature map $\Psi : \mathcal{R} \to \mathcal{M}$ where $\mathcal{M}$ is the space of semidefinite positive density matrices [20]. The input state is transformed via a quantum

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

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

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

with $U_{\boldsymbol{\theta}}$ the quantum neural network (QNN) is 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 :

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

During the training, the model learns the hidden data distribution from the training set in such way that $\hat{f}_{\boldsymbol{\theta}}$ approaches as close as possible to the target function f. At the end of the training, we expect that $\mathcal{U}_{\boldsymbol{\theta}}$ can also predict the label of the unseen test set.

The key idea behind geometric quantum machine learning (GQML) is to design models that capture the meaningful relations in the dataset by incorporating the architecture with the 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 a representation $R : \mathcal{S} \to \mathrm{Aut}(\mathcal{R})$ acting on the classical data space $\mathcal{R}$. We call that a function h has a label symmetry if and only if h is invariant under $\mathcal{S}$, i.e.,*

$$h(\rho_{R(g)\cdot\mathbf{x}}) = h(\rho_{\mathbf{x}}), \ \ \forall g \in \mathcal{S}. \tag{3}$$

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 $\mathcal{S}$-*equivariant quantum model* [15, 16].

Definition 2 (Equivariant Embedding) *We call an embedding $\Psi : \mathcal{R} \to \mathcal{M}$ with $\Psi(\mathbf{x}) = \rho_{\mathbf{x}}$ to be* ***equivariant*** *with respect to a symmetry element g, if and only if there exist 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 the embedding Ψ [16]. 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 :

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

where G is a Hermitian generator and $\mathcal{G}$ 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 to be* ***equivariant*** *with respect to $\mathcal{S}$ if and only if it commutes with $R_q(g)$ for all $g \in \mathcal{S}$, i.e.,*

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

or equivalently,

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

There exist different methods proposed to construct the equivariant gateset [21], such as *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 :

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

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

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

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

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, 21–26]. However, all the tests have been undertaken in the absence of hardware noise and the impact of noise on EQNN has never been studied before.

B. Noise models

The description of noise effects during quantum gates operation is based on the open quantum system theory [27, 28]. The Markovian evolution of the density matrix $\hat{\rho}_t$ of the qubits system in a given environment is described by the following Lindblad equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\rho}_t = -\frac{i}{\hbar}[\hat{\mathrm{H}}_t, \hat{\rho}_t] + \mathcal{L}\hat{\rho}_t, \tag{10}$$

where

$$\mathcal{L}\hat{\rho}_t = \epsilon^2 \sum_k \left[\hat{\mathrm{L}}_k\hat{\rho}_t\hat{\mathrm{L}}^{\dagger}{}_k - \frac{1}{2}\{\hat{\mathrm{L}}^{\dagger}{}_k\hat{\mathrm{L}}_k, \hat{\rho}_t\}\right], \tag{11}$$

$\hat{H}_t$ is the time-dependent Hamiltonian realizing a given gate, and $\hat{L}_k$ are the Lindblad operators capturing the action of the environment.

In this work, we only consider quantum channels acting locally on qubits. Some examples of these channels are *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 [28]. Then this can be written as,

$$\mathcal{N}(\rho) = \sum_i K_i \rho K_i^\dagger. \tag{12}$$

Bit Flip Channel: 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,

$$\mathcal{N}(\rho) = (1-p)\rho + pX\rho X. \tag{13}$$

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

$$\begin{aligned}\mathcal{N}(\rho) = &(1-p_0)(1-p_1)\rho \\ &+ p_0(1-p_1)(X\otimes I)\rho(X\otimes I) \\ &+ (1-p_0)p_1(I\otimes X)\rho(I\otimes X) \\ &+ p_0p_1(X\otimes X)\rho(X\otimes X),\end{aligned} \tag{14}$$

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

Depolarizing Channel: Kraus operators of the DP channel are $K_0 = \sqrt{1-p}\ I$, $K_1 = \sqrt{p/3}\ X$, $K_2 = \sqrt{p/3}\ Y$, $K_3 = \sqrt{p/3}\ Z$. 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 BF and DP channel are special cases of Pauli channels. 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 AD channel. Kraus operators of the AD channel can be written as,

$$K_0 = \begin{bmatrix}1 & 0\\ 0 & \sqrt{1-\gamma}\end{bmatrix},\ K_1 = \begin{bmatrix}0 & \sqrt{\gamma}\\ 0 & 0\end{bmatrix}, \tag{15}$$

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 [29]. This simplifies some computations and is used in this work. Please refer to Appendix A 2 for more details on the PTM formalism.

Having 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 the 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,

$$\rho_d = \mathcal{N}\circ\mathcal{U}_d\circ\cdots\circ\mathcal{N}\circ\mathcal{U}_2\circ\mathcal{N}\circ\mathcal{U}_1(\rho_0). \tag{16}$$

This can be visualized in the circuit picture as in Fig. 1, where Λ is the local action of 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.

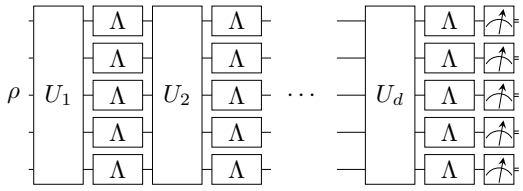


Fig. 1. Drawing of the local noise model. A circuit with input ρ and layers U_i, where the local Λ representing the action of noise are applied after each layer.

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

C. Concentration of measure

Variational algorithms may experience an exponential concentration of measure mainly due to the fact that the quantum state living in the 2^n-dimensional Hilbert space. The term *concentration of measure* refers to the observation that in many high-dimensional spaces, continuous functions are almost everywhere close to their mean [30]. As defined in Definition 4, the exponential concentration is commonly referred to as barren plateaus (BPs) in the QML literature. It was shown on different occasions that BPs may exist due to excessive expressivity of the circuit [31], highly entangled input state (*e.g.* volume law entanglement) [32], global observables [33] and hardware noise [34]. We refer the readers to recent work by Ragone et al. [5], which offers a unified picture of these causes.

Definition 4 (Exponential concentration) *Consider the random variable X. X is said to be deterministically exponentially concentrated in the number of qubits n around a certain fixed value α for some $b > 1$ if*

$$|X - \alpha| \leq \beta \in \mathcal{O}(1/b^n). \tag{17}$$

EQNNs can avoid BPs[1] by incorporating inductive biases into the ansatz design. This follows the fact that the existence of BPs in the case where the ansatz admits a Lie algebra $\mathfrak{g}$, such that $\dim(\mathfrak{g}) \in \mathcal{O}(\exp(n))$. Consequently, certain EQNNs can be constructed in a polynomial subspace of the Hilbert space such that they admit $\dim(\mathfrak{g}) \in \mathcal{O}(\text{poly}(n))$ and allow BP-free parametrized circuit designs [5]. This framework can ensure BP-free models unless there is no hardware noise present. Inevitably, EQNNs will experience noise-induced BPs [34]. This will be an important point when discussing the performance of EQNNs in the presence of noise.

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 $x \in \mathbb{R}$ is encoded using the R_Y rotation gate. Then, we will 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.

[1] Here, the term avoiding barren plateaus, is used in the context of barren plateaus, where the concentration of measure is caused by the *expressivity* of the ansatz. This doesn't hold for all cases, *e.g.* global observables, noise-induced BPs etc.

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

We assume 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 the 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 we have the freedom of choosing either X or Z as the representation $R_q(\sigma)$. The choice of representation is going to put a constraint also on the input state. For this walk-through, let's choose the input state $|\psi\rangle = |+\rangle$, $R_q(\sigma) = X$, $U = R_Y(\theta)$ and the choice of representation requires us to have the observable $O = X$. Here, we refer the reader to Refs. [15, 16, 21] for more details on constructing EQNNs.

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,

$$\hat{y}(x) = \frac{1}{4}\Big((f_x^d + f_z^d)\cos(x) + (f_x^d - f_y^d)\cos(x + 2\theta)\Big), \tag{18}$$

where f_i is the Pauli fidelity of the Pauli σ_i (*e.g.* $f_x = 1 - 2(p_y + p_z)$ according to the definition in Section II B, refer to 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 [34]. The second term of the equation constitutes the motivation of this work. We see that this term breaks the equivariance for some values. First and foremost, we see that the symmetry breaking term has an exponentially vanishing amplitude. Then, this term becomes even smaller when $f_x \simeq f_y$, which is, in fact, the case on hardware. These two results combined indicate that the noise-induced symmetry breaking should not hinder the equivariance under Pauli channels. 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 which 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's consider the non-unital AD channel with the probability γ. Then, the outcome of the circuit can be written as,

$$\hat{y}(x) = \frac{1}{4}((1-\gamma)^{d/2} + (1-\gamma)^{d})\cos(x) + \frac{1}{4}((1-\gamma)^{d/2} - (1-\gamma)^{d})\cos(x+2\theta) + \frac{1}{2}\Lambda^{(d)}_{\mathrm{AD}_{(4,1)}}\sin(\theta). \quad (19)$$

The term $\Lambda_{\mathrm{AD}_{(4,1)}}$ refers to the only off-diagonal entry in the Pauli Transfer Matrix (PTM) of the AD channel. The upper index (d) denotes the d^{th} power of this matrix. We refer the reader to Appendix A 2 and B for the details. We can write this term explicitly as,

$$\Lambda^{(d)}_{\mathrm{AD}_{(4,1)}} \simeq d\gamma - \frac{d(d-1)}{2}\gamma^2. \quad (20)$$

Here, we skip writing the remaining terms as their contribution will be negligible as long as we consider shallow circuits. Going back to the full expression for $\hat{y}(x)$, we immediately see that the AD channel results in a more complicated form. Nonetheless, it is easy to see the implications of each term one by one and this will give us the necessary intuition for the remaining part of this work.

The first and second terms jointly result in the exponential concentration induced by the AD channel. This can be easily seen by setting $\theta = 0$. The concentration happens around the third term, which shifts with the addition of each layer. This shift behaves approximately linear for practically relevant depths and noise levels[2], *e.g.* $\mathcal{O}(\gamma d)$. The second term is the one responsible for symmetry breaking. The term $((1-\gamma)^{d/2} - (1-\gamma)^{d})$ behaves similar to the $\Lambda^{(d)}_{\mathrm{AD}_{(4,1)}}$ term, *e.g.* is approximately linear for relevant values of the parameters. 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.

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,

$$\hat{y}(x) = \frac{1}{4}(f_x^d(1-\gamma)^{d/2} + f_z^d(1-\gamma)^{d})\cos(x) + \frac{1}{4}(f_x^d(1-\gamma)^{d/2} - f_z^d(1-\gamma)^{d})\cos(x+2\theta) + \frac{1}{2}\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}}\sin(\theta), \quad (21)$$

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

$$\Lambda^{(d)}_{\mathrm{P+AD}} \simeq \left(\sum_{k=1}^{d} f_z^k\right)\gamma - \left(\sum_{k=1}^{d}(k-1)\times f_z^k\right)\gamma^2. \quad (22)$$

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

$$\hat{y}(x) - \hat{y}(-x) = -(f_x^d(1-\gamma)^{d/2} - f_z^d(1-\gamma)^{d})\sin(\theta)\sin(x)/2 \quad (23)$$

This means the symmetry breaking is also modulated with the Pauli fidelity f_x and f_z in each layer. Notice that we can recover the term for pure AD channel if we set $f_x = f_z = 1$. Overall, the behavior of the term doesn't change, and it grows approximately linear in AD channel noise strength γ with minor contributions from the Pauli channel. This statement can also be generalized to multi-qubit systems. Following the structure of Eq. 14, we see that the addition of local noise channels on other qubits has negligible effects as these terms appear as multiplicative terms. Hence, we conjecture that a generic EQNN model experiences symmetry breaking dominantly under the AD channel, and the amount grows linearly in noise strength γ and depth d.

In Section IV, we perform numerical experiments to confirm the implications of the toy model and present evidence directly from hardware runs. 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 [18, 19]. 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*,

[2] Current superconducting hardware has $\gamma \simeq 10^{-2}$ and CNOT depth of 10 – 20. The values are approximate and vary from device to device.

which can be used to measure the symmetry breaking in GQML, but $\mathfrak{g}$-purity is expensive to compute and doesn't account for the 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 $\mathbf{x}_i$[3]). For this purpose, we have to make a choice of 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 $\hat{y}_i = \hat{y}_j$. We define accordingly $\mathcal{Z}_2$ symmetry generalized McNemar-Bowker (MB) test [35] as follows,

Definition 5 ($\mathcal{Z}_2$ 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 McNemar-Bowker (MB) test [35] of a model with predictions $\hat{y}_i$ for input $\mathbf{x}_i$ over M samples can be defined as,*

$$\chi^2 = \frac{1}{M} \sum_{i=1}^{M} \frac{(\hat{y}_i - \hat{y}_j)^2}{\hat{y}_i + \hat{y}_j} \tag{24}$$

This definition can be further extended to the 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 [36]. Unsuitably, this makes the choice data-dependent. With these points in mind, we choose the 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 will refer to this value as *label misassignment* (LM), as it counts the amount of the predictions that have a different prediction than their $\mathcal{Z}_2$ counterparts.

Definition 6 (Label Misassignment (LM)) *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 $\hat{y}_i$ are the continuous predictions of the model for input $\mathbf{x}_i$ and τ a step function that has the transition point at the median of all $\hat{y}_i$. Then, label misassignment (LM) of a model over M samples can be defined as,*

$$LM = \frac{1}{M} \sum_{i=1}^{M} \frac{(\tau(\hat{y}_i) - \tau(\hat{y}_j))^2}{\tau(\hat{y}_i) + \tau(\hat{y}_j)} \tag{25}$$

Notice that each term in the sum is either 0[4] (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−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) can't have an accuracy larger than 75%. Notice that 1−LM/2 doesn't predict the accuracy of a model but only upper bounds it, otherwise one would expect the completely random model to have 50% accuracy.

IV. EXPERIMENTS

In this section, we provide numerical experiments to validate our findings. To achieve this goal, we perform binary classification experiments, compute χ^2 and label misassignment (LM) that we previously defined in Section 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. 3 that carries this symmetry for the classification experiments.

As we 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 Fig. 4a, 4b. 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.

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 the representation Z_0Z_1. These generators are used to obtain the parametrized gates of

[3] Bold symbols are used to represent vectors. Here $\mathbf{x}_i$ denotes i^{th} data sample with arbitrary size.

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

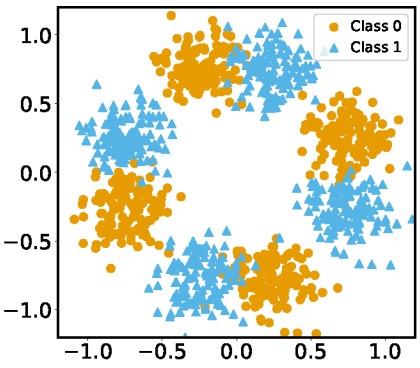


Fig. 3. An ad-hoc dataset with $\mathcal{Z}_2$ label symmetry such that $R(\sigma)\cdot(\mathbf{x}_i) = -\mathbf{x}_i$.

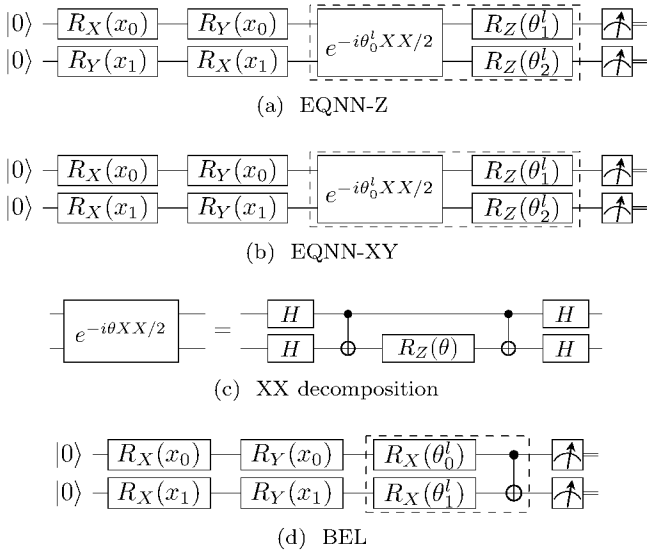


Fig. 4. Two qubit circuits used in the experiments.

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 the 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 EQNN-Z ansatz and $O = X_0Y_1$ for EQNN-XY.

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. EQNN-XY breaks the unwanted symmetry by employing the observable X_0Y_1, which doesn't commute with the $SWAP$ gate. In the case of EQNN-Z, this is broken at the data encoding level, since the order of R_X and R_Y gates are 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 non-equivariant model that doesn't use any geometric priors from the dataset as shown in Fig. 4d. We denote this model with *BEL* and compare it to the EQNN models using the same observables.

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. 4c and apply noisy gates after each layer as it was shown in Fig. 1. Furthermore, we simulate the EQNN-Z circuit without any decomposition to discern the noise effect and 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 [37]. The simulations are performed in the absence of shot noise using the Python library Pennylane [38].

We showcase the results of trained models under varying strength of DP and AD channels respectively in Fig. 5a, 5b. 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 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 demonstrate more pronounced effects compared to other models, while BEL-XY 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 can be observed better when we compare EQNN-Z-native and EQNN-Z. 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 much similarly than they do under the AD channel. Since EQNN-Z-native results in shorter depth, it is expected to perform better also under DP channel.

The second reason is the shift of mean for the Z ob-

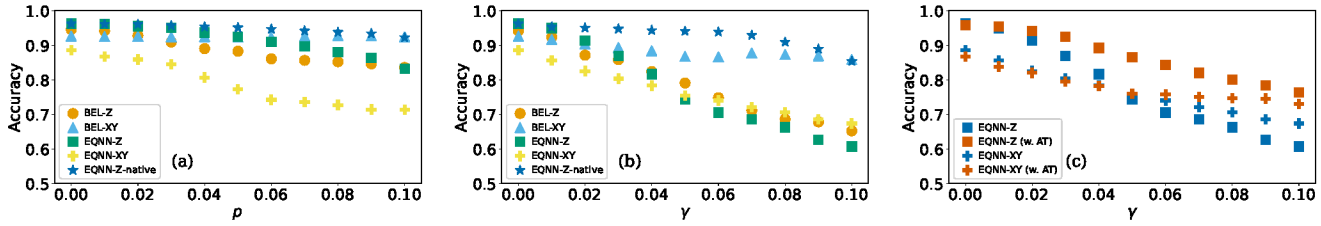


Fig. 5. Binary classification results under noise channels. All models are trained with ten different initializations and layers varied from 1-10. The test accuracy, averaged over the runs, is plotted for the best-performing layer of the corresponding model. Noise strength p in the case of the DP channel and γ in the case of the AD channel is varied from 0.0 to 0.1 with 0.01 increments. a) Results under DP channel. b) Results under AD channel. c) Results under AD channel with and without using adaptive thresholding (AT) during training.

servable 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 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. 5c, indicate significant improvement in the model performance, particularly in the case when measurements are affected asymmetrically in 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 non-equivariant 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 (please see Appendix A for details). Despite its impressive performance, it's important to note that this model faces implementation challenges on current quantum hardware due to limitations in the native gate set.

B. 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.

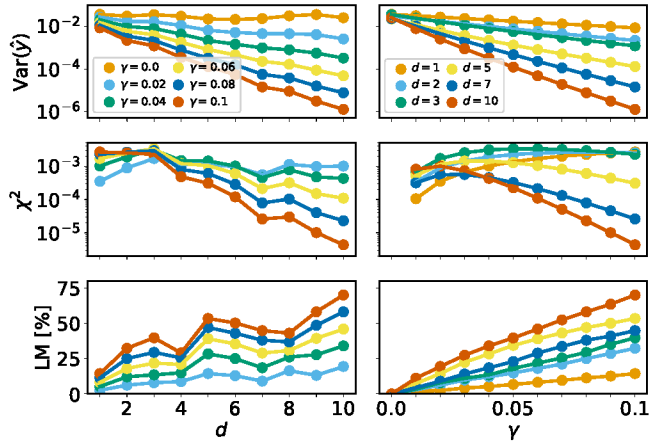


Fig. 6. Simulated two-qubit symmetry breaking for the EQNN-XY model under AD channel. Both columns show the same data points, on the left metrics are plotted against number of layers d, on the right metrics are plotted against noise strength γ.

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 varying number of layers in Fig. 6. The exponential decay of the variance numerically confirms the existence of the 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. χ^2 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

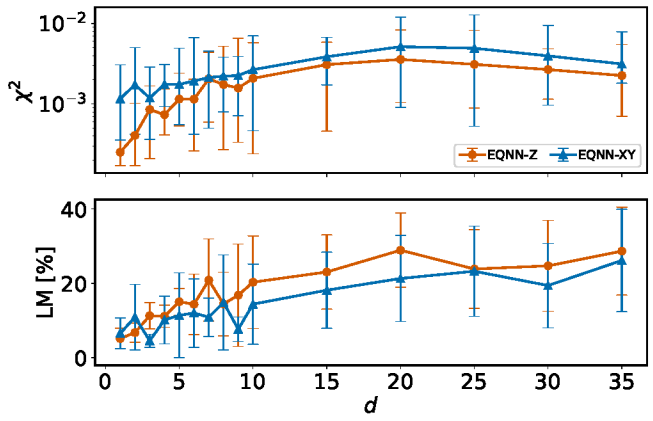


Fig. 7. Two qubit symmetry breaking for the EQNN-XY and EQNN-Z models measured on the *ibmq_cairo* superconducting quantum computer. (top) χ^2 and (bottom) LM are plotted against the number of layers d.

we have obtained in Eq. 19 and gives numerical evidence for our linear symmetry breaking conjecture.

Using this result, we can also comment on the binary classification performance. Following the bottom right panel of Fig. 6, 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−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 breaking. It is difficult to comment on the impact of a single factor, as there are many factors that contribute to the drop in performance in the presence of noise. Nonetheless, looking at Fig. 5, this value appears reasonable.

Next, we repeat this experiment on the *ibmq_cairo* superconducting quantum computer using the models EQNN-Z and EQNN-XY 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. 7. 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's not enough to look at the value of χ^2 to make comments 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 being asymmetric in the AD channel. This result also agrees with Fig. 7, in which EQNN-Z performance detoriates 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 well with the predictions of the AD channel dominating the symmetry breaking.

2. *Multi-qubit 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 E 1.

In order to run experiments on hardware, we define a hardware efficient multi-qubit circuit, *EQNN-HWE*, illustrated in Fig. 8. Data encoding is performed using R_X and R_Y gates. This results in the representation $Z^{\otimes n}$, similar to all other ansätze 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: $\{Z, XY, XYZ, XYZZ\}$. Notice that all the observables commute with the representation $Z^{\otimes n}$ to ensure equivariance. Fig. 9 presents the χ^2 and LM values obtained for log-depth ($d = \log_2 n$) circuits with varying observables.

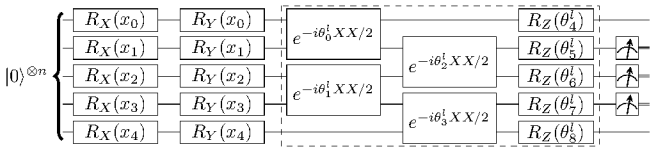


Fig. 8. Hardware efficient circuit used in the experiments to measure symmetry breaking. The part inside the dashed box is repeated d times with different parameters. Here, a five qubit circuit is plotted for reference. This model is denoted with EQNN-HWE.

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, 33] in general. We note that there are exceptions to this statement in the literature [39]. Furthermore, the trend for χ^2 with respect to the number of qubits aligns

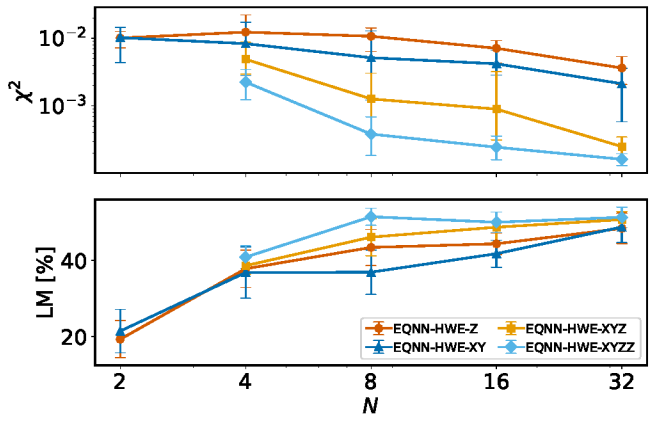


Fig. 9. Log-depth EQNN-HWE results from *ibmq_cusco*. Hardware efficient circuits defined in Fig. 8 with the number of layers $d = \log_2 n$. Each model uses a different observable denoted in the legend. The x-axis is plotted in log-scale, such that it is linear in number of layers.

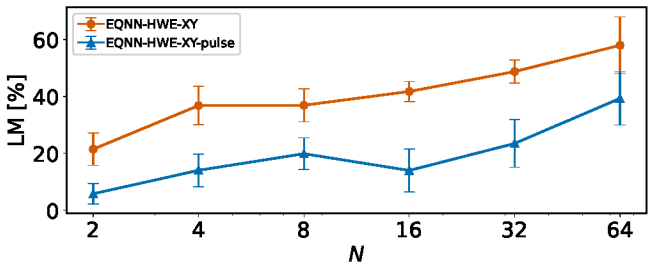


Fig. 10. Label misassignment of the EQNN-HWE-XY model using different transpilation methods. The label EQNN-HWE-XY refers to the standard transpilation used throughout this work. The EQNN-HWE-XY-pulse refers to the pulse efficient transpilation [41].

with the two-qubit models that were simulated only using 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.

These results indicate that log-depth EQNN models are not scalable on this hardware due to the combination of concentration and symmetry breaking. This shouldn't 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 [40] transpilation only exposes fixed pulse gates, such as the calibrated CNOT gate, or *ECR* gate, which is equivalent to CNOT gate up to single-qubit pre-rotations [41, 42]. 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 time for low entangling angles, and, thus leading 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. [41], 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 EQNN-HWE-XY using this scheme and report the results in Fig. 10. As expected from our linearity argument, the symmetry breaking reduces to half of the previous experiment since twice faster execution can be thought as half the AD channel strength. We refer the reader to Appendix E 2 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 being noise-induced BPs [34]. 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.

To address these challenges, we proposed effective strategies for mitigating performance drops caused by hardware noise. First of these is the adaptive thresholding that can cope with the concentration as well as the shift of 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 non-unital noise channels to Pauli channels [43]. However, the scalability of the amount of twirls to preserve equivariance remains unclear, posing an open question for future exploration.

In our experiments, we haven't considered error mitigation methods. This was an intentional choice. Our

target in this manuscript is to investigate the scalability of GQML on hardware, rather than just being able to execute circuits. This means error mitigation methods such as *probabilistic error cancellation* (PEC) are not suitable for this study due to their exponential overhead [43]. 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 D, revealing its effectiveness when provided with analytical expectation values but highlighting challenges with a limited number of shots.

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

CT is supported in part by the Helmholtz Association -"Innopool Project Variational Quantum Computer Simulations (VQCS)". SC is supported by the quantum computing for earth observation (QC4EO) initiative of ESA Φ-lab, partially funded under contract 4000135723/21/I-DT-lr, in the FutureEO program. SC, SV and MG 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. Authors would like to thank Stefan Kühn, Tobias Hartung and Marco Cerezo for fruitful discussions. 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.

[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, Nature Reviews Physics , 625 (2021).
[3] E. Fontana, N. Fitzpatrick, D. M. Ramo, R. Duncan, and I. Rungger, Evaluating the noise resilience of variational quantum algorithms, Physical Review 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, Nature Communications **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 (2023), arXiv:2309.09342 [quant-ph].
[6] E. R. Anschuetz and B. T. Kiani, Quantum variational algorithms are swamped with traps, Nature Communications **13**, 7760 (2022).
[7] X. You and X. Wu, Exponentially many local minima in quantum neural networks, 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) pp. 12144–12155.
[8] J. Rivera-Dean, P. Huembeli, A. Acín, and J. Bowles, Avoiding local minima in variational quantum algorithms with neural networks (2021), arXiv:2104.02955 [quant-ph].
[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 Science and Technology **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, Physical Review Research **3**, 033090 (2021).
[14] C. Tüysüz, G. Clemente, A. Crippa, T. Hartung, S. Kühn, and K. Jansen, Classical splitting of parametrized quantum circuits, Quantum Machine Intelligence **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] 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).
[17] 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).

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

[19] 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).

[20] 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).

[21] 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 (2022), arXiv:2210.08566 [quant-ph].

[22] S. Kazi, M. Larocca, and M. Cerezo, On the universality of s_n-equivariant k-body gates (2023), arXiv:2303.00728 [quant-ph].

[23] 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 (2023), arXiv:2210.07980 [quant-ph].

[24] L. Schatzki, M. Larocca, Q. T. Nguyen, F. Sauvage, and M. Cerezo, Theoretical guarantees for permutation-equivariant quantum neural networks (2022), arXiv:2210.09974 [quant-ph].

[25] 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).

[26] S. Y. Chang, M. Grossi, B. Le Saux, and S. Vallecorsa, Approximately equivariant quantum neural network for p4m group symmetries in images, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 01 (2023) pp. 229–235.

[27] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, 2007).

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

[29] 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).

[30] M. P. Müller, D. Gross, and J. Eisert, Concentration of measure for quantum states with a fixed expectation value, Communications in Mathematical Physics **303**, 785 (2011).

[31] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus, PRX Quantum **3**, 010313 (2022).

[32] C. Ortiz Marrero, M. Kieferová, and N. Wiebe, Entanglement-Induced Barren Plateaus, PRX Quantum **2**, 040316 (2021).

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

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

[35] A. Krampe and S. Kuhnt, Bowker's test for symmetry and modifications within the algebraic framework, Computational Statistics & Data Analysis **51**, 4124 (2007).

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

[37] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, CoRR **abs/1412.6980** (2014).

[38] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. Akash-Narayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. D. Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez-Barrera, R. Moyard, Z. Niu, L. J. O'Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, and N. Killoran, Pennylane: Automatic differentiation of hybrid quantum-classical computations (2018), arXiv:1811.04968 [quant-ph].

[39] N. L. Diaz, D. García-Martín, S. Kazi, M. Larocca, and M. Cerezo, Showcasing a barren plateau theory beyond the dynamical lie algebra (2023), arXiv:2310.11505 [quant-ph].

[40] Qiskit contributors, Qiskit: An open-source framework for quantum computing (2023).

[41] 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).

[42] 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).

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

[44] 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).

[45] G. Li, Y. Ding, and Y. Xie, Tackling the qubit mapping problem for nisq-era quantum devices, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19 (Association for Computing Machinery, 2019) p. 1001–1014.

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

[47] 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).
[48] Qiskit 0.33 release notes, https://docs.quantum.ibm.com/api/qiskit/release-notes/0.33, accessed: 2024-01-12.
[49] 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).
[50] N. Khaneja and S. J. Glaser, Cartan decomposition of SU(2n) and control of spin systems, Chemical Physics **267**, 11 (2001).
[51] 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).

Appendix A: Noise models

1. Amplitude Damping Channel

In Section II B, we introduced the *amplitude damping* (AD) channel with the following Kraus operators,

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

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

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

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. *Pauli transfer matrix* (PTM) formalism allows us to simplify this process [29]. In this formalism, we start by choosing the normalized Pauli basis $\hat{\mathbb{P}} = \frac{1}{\sqrt{2}}\{I, X, Y, Z\}$. 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 $[1/2, 0, 0, 1/2]$. 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 $tr(\rho\hat{P})$. 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's recall 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 will 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 *bit flip* (BF), *phase flip* (PF), *depolarizing* (DP) channels' Kraus operators and the corresponding PTMs.

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

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

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

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

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,

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

PTM of the AD channel can also be obtained following the same procedure. Here we will 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 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's easy to show this using the PTM formalism,

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

Since we are considering local noise models, the PTM of the n-qubit AD channel can be obtained by taking n^{th} 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 section, we will give the details for the calculations in Section III A. Let's start by recalling the definition of the toy model, which was described in Fig. 2. The data is 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 $U^\dagger$ gates. 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. Then the final state of the model, before measurement, for input data x is given as,

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

The terms $\mathrm{Ad}_{R_Y(\theta)}$ and $\mathrm{Ad}_{R_Y(-\theta)}$ that appear next to each other will be 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 $[1/2, 1/2, 0, 0]$. 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 the repetitive application of the noise channel will appear as the d^{th} 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 $[0, 1, 0, 0]$ 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. 18,

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

After the matrix multiplication, one obtains,

$$\hat{y}(x) = ((f_x^d + f_z^d)\cos(x) + (f_x^d - f_y^d)\cos(x + 2\theta))/4. \tag{B5}$$

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

$$\Lambda_{\mathrm{AD}}^{(d)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (1-\gamma)^{d/2} & 0 & 0 \\ 0 & 0 & (1-\gamma)^{d/2} & 0 \\ \Lambda_{\mathrm{AD}_{(4,1)}}^{(d)} & 0 & 0 & (1-\gamma)^{d} \end{bmatrix}, \tag{B6}$$

where the term $\Lambda_{\mathrm{AD}_{(4,1)}}^{(d)}$ corresponds to the matrix element of the index $(4,1)$. This term can be explicitly written as,

$$\Lambda_{\mathrm{AD}_{(4,1)}}^{(d)} \simeq d\gamma - \frac{d(d-1)}{2}\gamma^2. \tag{B7}$$

As also described in the main text, we skip writing the remaining terms as their contribution becomes negligible when realistic values are considered for the variables. For example, $\gamma \simeq 10^{-2}$ and $d < 20$. 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,

$$\hat{y}(x) = \frac{1}{4}((1-\gamma)^d + (1-\gamma)^{d/2})\cos(x) + \frac{1}{4}((1-\gamma)^d - (1-\gamma)^{d/2})\cos(x+2\theta) + \frac{1}{2}\Lambda^{(d)}_{\mathrm{AD}_{(4,1)}}\sin(\theta). \tag{B8}$$

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,

$$\Lambda_{\mathrm{P+AD}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & f_x\sqrt{1-\gamma} & 0 & 0 \\ 0 & 0 & f_y\sqrt{1-\gamma} & 0 \\ f_z\gamma & 0 & 0 & f_z(1-\gamma) \end{bmatrix}. \tag{B9}$$

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,

$$\hat{y}(x) = \frac{1}{4}(f_x^d(1-\gamma)^{d/2} + f_z^d(1-\gamma)^d)\cos(x) + \frac{1}{4}(f_x^d(1-\gamma)^{d/2} - f_z^d(1-\gamma)^d)\cos(x+2\theta) + \frac{1}{2}\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}}\sin(\theta). \tag{B10}$$

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

$$\Lambda^{(d)}_{\mathrm{P+AD}_{(4,1)}} \simeq \left(\sum_{k=1}^{d} f_z^k\right)\gamma - \left(\sum_{k=1}^{d}(k-1)\times f_z^k\right)\gamma^2. \tag{B11}$$

Appendix C: 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. 11, we present the simulation of the EQNN-Z model simulated with 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.

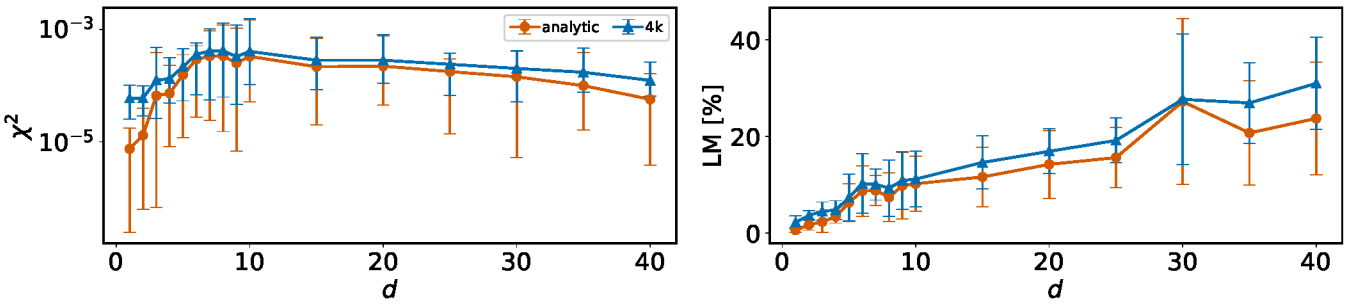


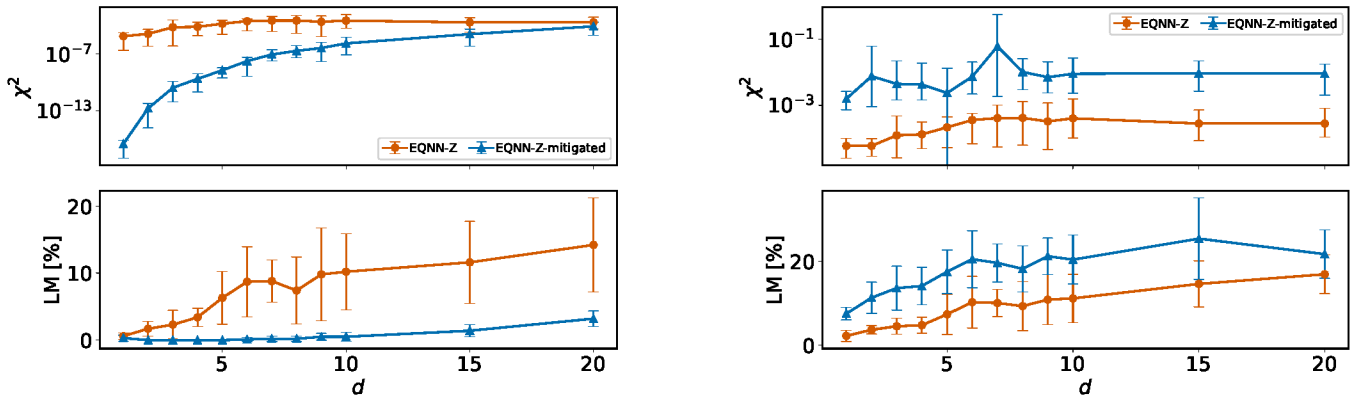
Fig. 11. Comparison of symmetry breaking measurements with (4000 shots) and without shot noise. EQNN-Z model is simulated under AD channel with $\gamma = 0.01$.

Appendix D: Zero noise extrapolation

Zero noise extrapolation (ZNE) is an error mitigation method that uses the expectation values measured at different noise strengths [44]. These values can be extrapolated to 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 varying number of layers in the presence of AD channel noise are presented in Fig. 12.

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.



(a) Analytical expectation values

(b) Expectation values with 4k shots

Fig. 12. Symmetry breaking experiments with ZNE. EQNN-Z model used in Section IV B 1 is employed with and without shot noise.

Appendix E: Hardware experiments

In this section, 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 [45], 1Q gate optimization, and dynamical decoupling [46]. The list of the devices, along with some of their properties, is presented in Table I.

Name	T1 [us]	T2 [us]	Gate time [ns]	Readout length [ns]
ibmq_cairo	91.99	92.4	321.778	732.444
ibmq_cusco	126.78	78.77	460	4000

Table I. 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.

1. Hardware topology

The coupling map of *ibmq_cusco* used for the 64 qubit experiments is presented in Fig. 13. We choose a suitable nearest neighbor set of qubits to have 1D connectivity.

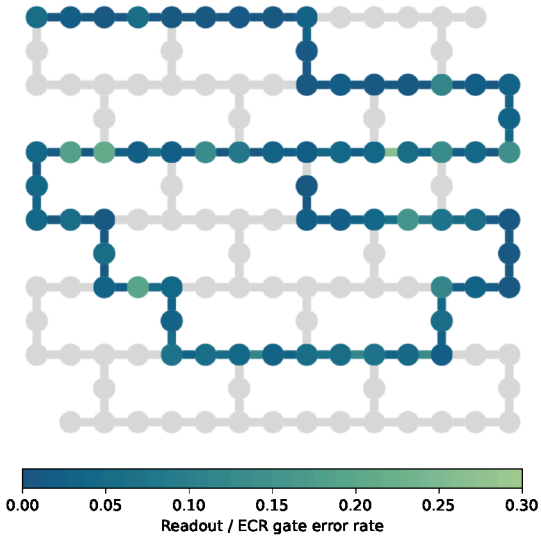


Fig. 13. Coupling map of *ibmq_cusco* and the qubit configuration chosen to run the quantum circuit of 64 qubits. The colors represent the readout error for each qubit and the two-qubit ECR gate for each qubit connection.

2. Pulse efficient transpilation

In order to run a generic quantum circuit on the real IBM Quantum hardware, the circuit should first transpired into the set of basis gates, which are pre-calibrated on the corresponding hardware. The automatic IBM quantum transpilation only exposes fixed-frequency cross-resonance gates [47]. Instead of the fixed frequency gates, we can use the continuous gate native to the quantum hardware. For the low rotation angles, the circuit duration becomes shorter, leading to less decoherence noise and more accurate results.

```
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. 14. Python code for RZX transpilation in Qiskit implementation taken from Ref. [48].

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

$$\alpha* = \|A*\|[w* + \sqrt{2\pi}\sigma \cdot erf(\frac{rf}{\sqrt{2}\sigma})], \tag{E1}$$

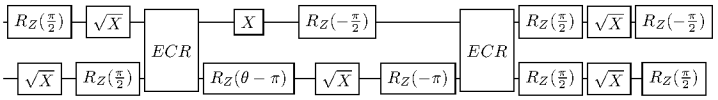
with $A*$ the amplitude, $w*$ the width, rf the risefall and σ the standard deviation of the corresponding Gaussian

flanks [41]. The pulse of $RZX(\theta)$ gate is created by rescaling the area α as [49]

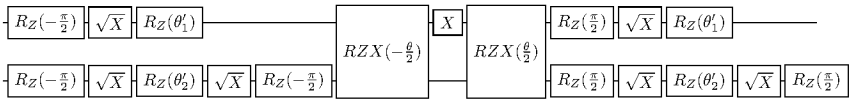
$$\alpha(\theta) = \frac{2\theta\alpha*}{\pi}. \tag{E2}$$

In the Qiskit implementation, the RZX-based transpilation works as shown in Fig. 14. First of all, we collect all the consecutive two-qubit operations and consolidate them into a general two-qubit $SU(4)$ operation. Then, the corresponding two-qubit gate is decomposed in terms of echoed RZX gates by leveraging Cartan's decomposition [50]. Those gates are calibrated by scaling the Gaussian square pulses of the fixed-frequency CNOT or ECR gates. Finally, the single-qubit gates are simplified and optimized.

Fig. 15 and Fig. 16 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. 16, the pulse schedule with RZX decomposition is much shorter compared to the one with ECR decomposition, resulting in less decoherence and better results, as mentioned previously.

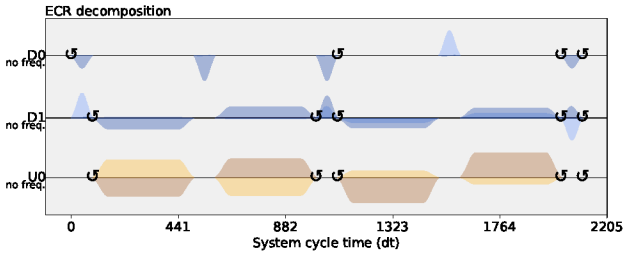


(a) ECR decomposition

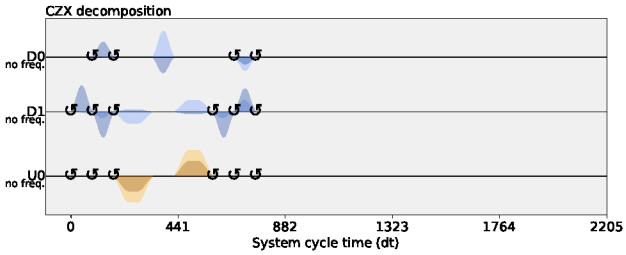


(b) RZX-based decomposition with the built-in echo.

Fig. 15. Circuit decomposition of $RXX(\theta)$ gate on *ibmq_cusco*. θ_1' and θ_2' in (b) are the single-qubit rotation angles computed by Cartan's decomposition



(a) Pulse schedule for ECR decomposition



(b) Pulse schedule for RZX decomposition

Fig. 16. Pulse schedule for ECR-based decomposition and pulse-efficient RZX-based decompositions (c.f. Fig. 15). The symbol ↺ indicates the virtual Z gates [51].