

Learning quantum symmetries with interactive quantum-classical variational algorithms

Jonathan Z. Lu,^{*} Rodrigo A. Bravo,[†] Kaiying Hou,[†]
Gebremedhin A. Dagneu,[‡] Susanne F. Yelin,[§] and Khadijeh Najafi[¶]

Department of Physics, Harvard University, Cambridge, MA

(Dated: June 27, 2022)

A symmetry of a state $|\psi\rangle$ is a unitary operator of which $|\psi\rangle$ is an eigenvector. When $|\psi\rangle$ is an unknown state supplied by a black-box oracle, the state's symmetries serve to characterize it, and often relegate much of the desired information about $|\psi\rangle$. In this paper, we develop a variational hybrid quantum-classical learning scheme to systematically probe for symmetries of $|\psi\rangle$ with no a priori assumptions about the state. This procedure can be used to learn various symmetries at the same time. In order to avoid re-learning already known symmetries, we introduce an interactive protocol with a classical deep neural net. The classical net thereby regularizes against repetitive findings and allows our algorithm to terminate empirically with all possible symmetries found. Our scheme can be implemented efficiently on average with non-local SWAP gates; we also give a less efficient algorithm with only local operations, which may be more appropriate for current noisy quantum devices. We demonstrate our algorithm on representative families of states.

I. INTRODUCTION

Symmetries are of ubiquitous importance across physics, relating intimately to conservation laws and guiding the formulation of physical theories. The search for symmetries in a given physical system is a central problem in the field, and until recently, has remained a purely analytical and model-driven task. However, developments in machine learning have now opened the door for data-driven approaches. Classically, recent studies have used learning algorithms to discover conserved quantities by using the system's equation of motion [1–5] or by clustering methods [6]. In the quantum setting, classical neural networks have been successfully trained to classify states of matter [7–9], including symmetry-protected topological ones [10–12]. Symmetry-abiding neural networks can also be used as a variational ansatz for states of lattice models [13, 14]. Quite recently, variational quantum algorithms that utilize the symmetries of a particular problem have appeared in the literature [15, 16]. Along these lines, it has been shown that knowledge of symmetries can bootstrap algorithms such as the quantum approximate optimization algorithm (QAOA) [17].

Beyond their intrinsic importance, quantum symmetries also encode key information about the underlying quantum state. Specifically, when a state $|\psi\rangle$ is unknown, efficient extraction of information

about $|\psi\rangle$ from limited data is of great interest across both physics and theoretical computer science. Full information extraction, however, is exponentially costly with the system size, requiring complete state tomography [18]. While novel approaches circumventing this issue have appeared—including basis-enhanced Born machines, classical machine learning, and shadow tomography—comparatively little attention has been paid to the direct and systematic learning of quantum symmetries, which also serve to characterize a state [19–27]. In this article, we take a first step in that direction.

Define a quantum symmetry of $|\psi\rangle$ to be a unitary U such that

$$U|\psi\rangle = e^{i\theta}|\psi\rangle; \quad (1)$$

that is, an operator U for which $|\psi\rangle$ is an eigenvector. For maximal generality, we shall assume that a black-box oracle machine prepares copies of an unknown state $|\psi\rangle$; our goal is to efficiently express the set of symmetries of $|\psi\rangle$. To find the symmetries of $|\psi\rangle$ [28], we propose a hybrid quantum-classical algorithm consisting of a variational quantum circuit interacting with a classical neural network. Under the oracle model, we assume no prior information about $|\psi\rangle$, so that our algorithm is agnostic to the input state whose symmetries it learns. The quantum circuit generates symmetries of $|\psi\rangle$ while simultaneously training the classical net to “remember” the history of symmetries already found. In turn, the classical net alerts the quantum circuit when it parameterizes an operator close to a previously-found symmetry, allowing the quantum circuit to redirect its search and thereby avoid repeatedly generating similar symmetries. By both generating symmetries and keeping track of known symmetries, our algorithm continuously finds new symmetries.

^{*} jlu@college.harvard.edu

[†] This symbol denotes equal author contribution.

[‡] Presently at 1QB Information Technologies, Inc.

[§] syelin@g.harvard.edu

[¶] knajafi@ibm.com; IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA

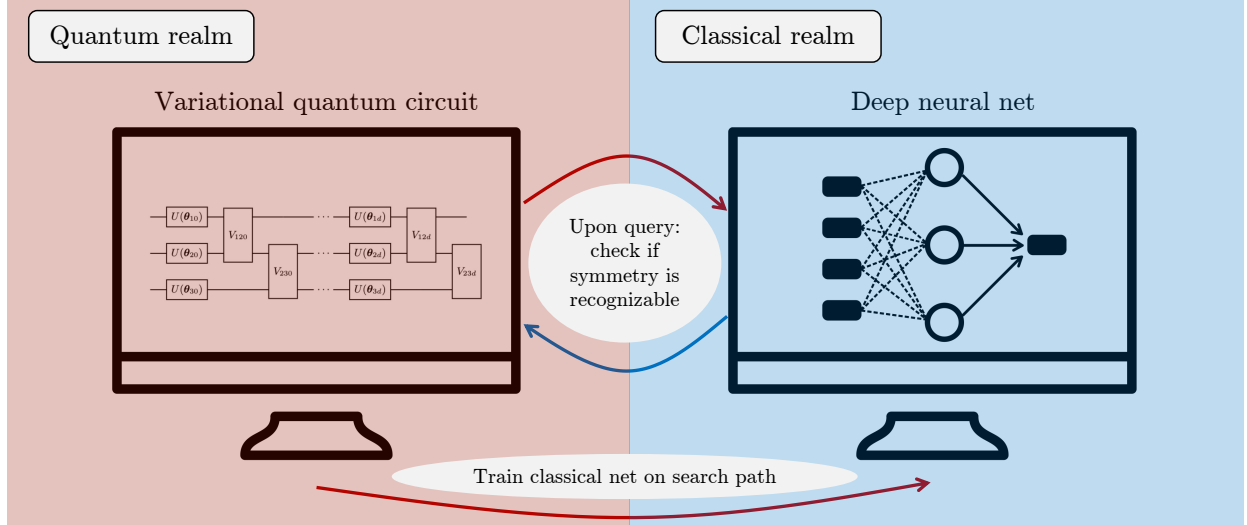


FIG. 1. Schematic overview of symmetry learning scheme. A variational quantum circuit minimizes a loss function to generate symmetries, at the same time training a classical neural network to recognize the path of potential symmetry operators it searches. The classical net then alerts the quantum circuit when it is searching for symmetries similar to those already found, so that the quantum circuit can redirect its search accordingly.

Thus, we observe empirically that the learning procedure eventually terminates. The overall structure of quantum-classical interaction is shown in Fig. 1.

In Section II, we discuss the full details of the symmetry learning scheme and place it within the context of noisy near-term quantum hardware. In Section III, we demonstrate the algorithm on three representative families of quantum states. We also benchmark the scalability of the algorithm and its robustness to noise. Finally, in Section IV we summarize our findings and discuss some applications of symmetry learning.

II. HYBRID LEARNING SCHEME

By analogy to an experimental setting, we assume access to copies of an unknown state $|\psi\rangle$ and the task is to discover the set of its symmetries. Since the closeness of an operator to a symmetry can be quantified in a manner similar to metrics of state overlap, symmetry learning has a natural interpretation as an optimization problem, given as

$$S[|\psi\rangle] = \{V \in U(2^L) : |\langle\psi|V|\psi\rangle|^2 = 1\}, \quad (2)$$

where $U(2^L)$ is the unitary matrix Lie group of dimension 2^L (for L qubits). Symmetries can be filtered into a collection of sets by fixing a universal parameterized quantum circuit (PQC) family $C_{L,D}(\theta)$ of depth D on L qubits and parameters θ . Define $S_D[|\psi\rangle]$ as the collection of symmetries of $|\psi\rangle$ representable by $C_{L,D}(\theta)$. By universality, every symme-

try is contained in $S_{O(\exp(L))}[|\psi\rangle]$ and

$$S_0[|\psi\rangle] \subseteq S_1[|\psi\rangle] \subseteq \dots S_{O(\exp(L))}[|\psi\rangle]. \quad (3)$$

We formally define the symmetry learning problem as the discovery and classification of all symmetries of $|\psi\rangle$ for a fixed and constant depth D . In practice, this may neglect symmetries that require exponentially long circuit depths to be implemented in the chosen PQC architecture.

We develop the symmetry learning scheme in three stages. First, we devise a method to verify whether a given operator U is a symmetry. We then show that the verification procedure can be upgraded into a learning procedure, by introducing a variational quantum algorithm (VQA) built upon a PQC family $C_{L,D}$ of depth D acting on L qubits. Finally, we boost the learning scheme by introducing a regularization technique based on classical deep learning that prevents the VQA from repetitively proposing similar symmetries. The full scheme is illustrated in Fig. 2, which we refer to throughout this section.

A. Symmetry Verification

To begin, suppose that we are given an operator U represented by a PQC family $C_{L,D}$. We restrict ourselves only to symmetries expressible in such a decomposition, noting that a universal quantum gate family can generally express any operator up to small error. Our first step will be to verify

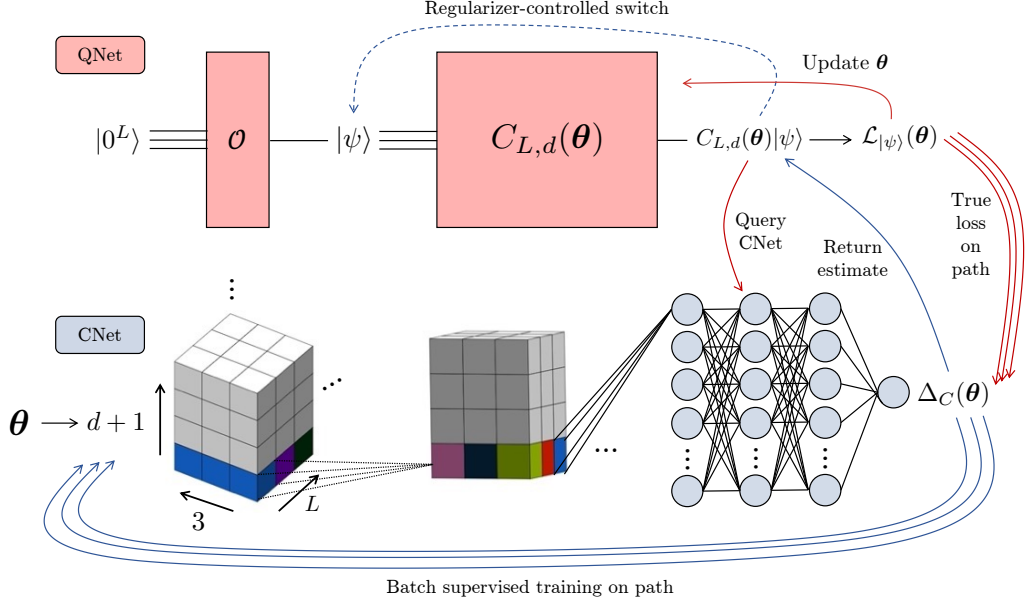


FIG. 2. Depiction of the symmetry learning algorithm. The quantum net (red) involves a variational quantum circuit $C_{L,d}(\theta)$ on L qubits and with block-depth d measured by a loss function $\mathcal{L}_{|\psi\rangle}(\theta)$. θ is varied until $C_{L,d}(\theta)$ represents a symmetry. At the end of each epoch, a classical 3-dimensional convolutional deep net (CNet; blue) learns the loss function along the path just explored by the QNet. During future epochs, the CNet informs the QNet as to whether it has already explored its current path, and hence whether it needs to leave its current path to an as-yet explored region. The CNet structure matches the parameter structure of the QNet. The first layer convolves over each set of 3 parameters per qubit, and the second layer convolves over every parameter; the result is inputted into a 3-layer fully connected network.

that U is indeed a symmetry of $|\psi\rangle$. Such a problem can be decided in polynomial time by a method adapted from quantum fingerprinting, known as the SWAP_ϵ test [29]. The heart of the procedure is the controlled-SWAP gate, which takes $|0\rangle \leftrightarrow |1\rangle$ unitarily if a control qubit is $|1\rangle$. By utilization of the circuit in Fig. 3, measurement of the ancillary qubit yields $|0\rangle$ with probability

$$\Pr[0] = \frac{1}{2} + \frac{1}{2} |\langle \psi | U | \psi \rangle|^2. \quad (4)$$

Thus the overlap $|\langle \psi | U | \psi \rangle|^2$ is a function of the bias (probability distance from $1/2$) of a coin flip, which can be determined to error ϵ in $O(1/\epsilon^2)$ trials [29]. Since the number of swaps scales linearly with qubit size, the time complexity of the SWAP_ϵ test is polynomial: $O(L/\epsilon^2)$.

Although the SWAP_ϵ is efficient, it uses long-range SWAP gates spanning $O(L)$ qubits, which are not practical for general near-term quantum devices. In Section II E, we give an alternative verification procedure free of long-range couplings at the expense of computational efficiency.

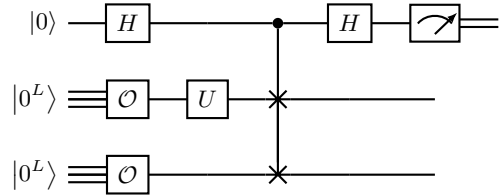


FIG. 3. Verification circuit V implementing a SWAP_ϵ test. The central operation is a swap of the two L -qubit registers (bottom registers), controlled by the ancillary qubit (top register). By running V $O(1/\epsilon^2)$ times, we can determine the overlap $|\langle \psi | U | \psi \rangle|^2$ up to error ϵ . We estimate U to be a symmetry of $|\psi\rangle$ if the overlap is at least $1 - O(\epsilon)$.

B. Variational Quantum Generative Algorithm

In the process of symmetry verification we utilized the matrix element $|\langle \psi | U | \psi \rangle|^2$ as a measure of the closeness U is to a symmetry. The overlap may be interpreted as a loss metric, i.e.

$$\mathcal{L}_{|\psi\rangle}(U) = (1 - |\langle \psi | U | \psi \rangle|^2)^2. \quad (5)$$

This overlap can be estimated statistically by the SWAP_ϵ test. The formulation of verification as a loss

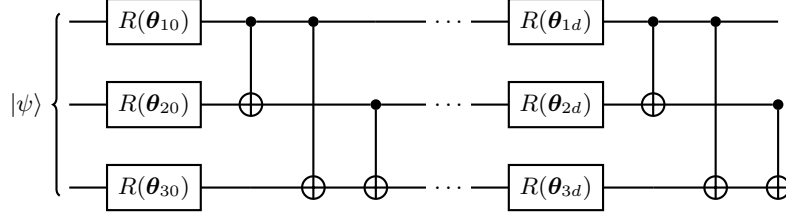


FIG. 4. Example of a universal parameterized quantum circuit on $L = 3$ qubits. The cross-hairs are controlled-NOT gates and the R gates are single-qubit rotations, parameterized by θ . There are d layers of $\binom{L}{2}$ controlled-NOT gates between each pair of qubits, sandwiched by the rotation gates.

allows for a transformation from verification—a discriminative procedure—to proposition (search)—a generative model. Consider again a universal PQC; one such family includes layers of $\binom{L}{2}$ CNOT gates that are sandwiched in between single-qubit rotation gates, as depicted in Fig. 4 for $L = 3$ qubits. This PQC ansatz is well-established and is known to be universal as the depth reaches exponential values [30]. In place of circuit depth, we use the *block-depth* $d \geq 0$, the number of layers of $\binom{L}{2}$ CNOT gates. Hence, $d = 0$ corresponds to a single layer of rotations, $d = 1$ gives 2 layers of rotations sandwiching 1 layer of CNOT gates, etc. The circuit on L qubits is specified uniquely by either the circuit depth D or the block-depth d ; we shall use the latter, writing the circuit as $C_{L,d}$ and filtering the symmetries by block-depth as

$$S_0[|\psi\rangle] \subseteq \cdots \subseteq S_d[|\psi\rangle] \subseteq \cdots S_{O(\exp(L))}[|\psi\rangle]. \quad (6)$$

Under a parameterization θ , let

$$\mathcal{L}_{|\psi\rangle}(\theta) = \mathcal{L}_{|\psi\rangle}(C_{L,d}(\theta)). \quad (7)$$

Our VQA generates symmetries as follows. Initialize the PQC randomly by drawing $\theta_0 \sim \text{Unif}(0, 2\pi)^{3L(d+1)}$. Next, run $C_{L,d}$ on $|\psi\rangle$ and compute $\mathcal{L}_{|\psi\rangle}(\theta_0)$. The loss will likely be large, but θ may be improved by an updating algorithm of classical machine learning. Our algorithm employs Nelder-Mead simplex search, which optimizes θ locally by updating the parameter to its simplex neighbor with the minimum loss value [31]. The use of a classical updating scheme on a PQC has been adopted to considerable success in many hybrid quantum-classical algorithms such as variational quantum eigensolver (VQE) and QAOA. Here, θ is iteratively updated until $\mathcal{L}_{|\psi\rangle}(\theta)$ falls below an error threshold δ , at which point the symmetry is proposed. See Appendix B for details on our choices of hyperparameters.

Since the PQC consists of layers of parameterized gates updated by a training algorithm, it has a natural interpretation as a quantum analog of a neural

network, which we refer to as the QNet. The QNet VQA comprises the top (red) half of Fig. 2.

C. Symmetries as Manifolds

The VQA proposed in the previous section above successfully generates symmetries; however, with no further assistance the PQC may propose the same symmetry repetitively. In the extreme case, the PQC could simply choose to always output the identity matrix. The prevention of repeat findings motivates the introduction of a regularization procedure. A naïve method might enumerate the set of symmetries found up to the k th epoch of search, $\theta_1, \dots, \theta_k$, and then adding a regularization term to the loss function of the form

$$\mathcal{L}_{|\psi\rangle}^{(\text{reg})}(\theta) = \mathcal{L}_{|\psi\rangle}(\theta) + \lambda \sum_{k'=0}^k \|\theta - \theta_{k'}\|^{-2}. \quad (8)$$

Yet such regularizers are doomed to fail due to the algebraic structure of $S[|\psi\rangle]$. Not only are the products of symmetries also symmetries, generating $O(\exp(k))$ symmetries from a given set of size $O(k)$, but the symmetries are generally continuous families forming a Lie subgroup $S_d[|\psi\rangle] \leq \text{U}(2^L)$. Consider, for example, the L -qubit GHZ state

$$|\text{GHZ}_L\rangle = \frac{|0^L\rangle + |1^L\rangle}{\sqrt{2}}. \quad (9)$$

For simplicity, consider $d = 0$ (single-qubit rotation) symmetries $S_0[|\text{GHZ}_L\rangle]$. Under a manifold picture of the Lie subgroup, the rotation symmetries can be partitioned into two submanifolds

$$\left\{ \bigotimes_{k=1}^L \begin{pmatrix} e^{i\alpha_k} & 0 \\ 0 & e^{i\beta_k} \end{pmatrix} \middle| \sum_{k=1}^L \alpha_k = \sum_{k=1}^L \beta_k = 1 \right\}, \quad (10)$$

$$\left\{ \bigotimes_{k=1}^L \begin{pmatrix} 0 & e^{i\alpha_k} \\ e^{i\beta_k} & 0 \end{pmatrix} \middle| \sum_{k=1}^L \alpha_k = \sum_{k=1}^L \beta_k = 1 \right\}.$$

The first submanifold is composed of diagonal matrices with phases as their entries, and includes the identity matrix. The second submanifold is composed of off-diagonal matrices with phases as their entries, and includes the bit flip symmetry X where X is the Pauli spin operator σ_x . Each submanifold has four degrees of freedom, forming a four-torus T^4 . These may be equivalently expressed in circuit decomposition as

$$\begin{aligned} M_1 &= \left\{ \bigotimes_{k=1}^L XP(\alpha_k)XP(\beta_k) \right\}, \\ M_2 &= \left\{ \bigotimes_{k=1}^L P(\beta_k)XP(\alpha_k) \right\}, \end{aligned} \quad (11)$$

where $P(\varphi)$ is the phase gate

$$P(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}. \quad (12)$$

Each submanifold M_i is a smoothly parameterized set of symmetries. Thus, in principle we need only inspect a few points in the submanifold to deduce a closed-form expression for M_i .

D. Interactive Classical Learning

To learn efficiently, we cannot explore the same submanifold repeatedly while other manifolds remain unexplored. Rather, we must develop a procedure to push the PQC away from symmetry submanifolds already explored sufficiently. While such a task is not obvious, it is not impossible from a measure-theoretic standpoint. That is, there will never be a case in which $|\psi\rangle$ has two symmetry submanifolds M_1, M_2 such that $\dim(M_1) < \dim(M_2)$, so that M_1 would never be found. For as the manifolds are also groups, there exists automorphisms mapping M_1 to M_2 and vice versa, and hence the topological dimension of every symmetry submanifold must be identical. Nonetheless, no pointwise regularization procedure, such as that of Eq. (8), can recognize two symmetries being of the same submanifold and thereby push the PQC away from proposing both of them.

We therefore introduce a regularization method based on classical deep learning: rather than keeping track of specific symmetries, we train the classical net (CNet) to approximate $\mathcal{L}_{|\psi\rangle}(\theta)$ by its output $\hat{\mathcal{L}}_{|\psi\rangle}(\theta)$. The direct statistical estimation of $\mathcal{L}_{|\psi\rangle}(\theta)$ requires quantum operations and measurement handled by the QNet, so we supervise the CNet using the QNet itself. Specifically, for every batch of loss-functional evaluations the QNet performs as

it searches for symmetries, we train the CNet in parallel using those evaluations as the supervisory dataset. Each batch is a transcript of the path in $U(2^L)$ (henceforth abbreviated to U) walked by the QNet that terminates at a symmetry. Ultimately, the aim of the CNet is to estimate with low error the loss for θ in a neighborhood of each θ explored by the QNet. That is, we train the CNet by minimizing the squared error

$$\Delta_C(\theta) = (\mathcal{L}_{|\psi\rangle}(\theta) - \hat{\mathcal{L}}_{|\psi\rangle}(\theta))^2 \quad (13)$$

where $\hat{\mathcal{L}}_{|\psi\rangle}$ is the CNet estimate of the true loss.

Since the landscape of the loss function can be highly complex, we cannot hope to obtain a good classical estimate of the loss globally with only a few local training points. Instead, we design the CNet to estimate local patches around points (neighborhoods of the point in U) explored by the QNet with very low error, but estimate patches far from points explored by the QNet with high error. Equivalently, the CNet should overfit purposefully. Thus, by analyzing the CNet estimation error on a point θ , the QNet can determine whether $\mathcal{L}_{L,d}(\theta)$ resembles a symmetry previously explored (i.e. the absolute difference of their overlaps are small). If so, the QNet can adjust accordingly by leaving the local region of parameter space to hopefully find another region of U that has not yet been explored. We found numerically (see Appendix C) that the average difference in Δ_C evaluated on a training point versus a random point is about two orders of magnitude, making for easy distinction.

Classical deep neural nets are well-suited to the task because a CNet can extrapolate relatively well locally. Consequently, as we will also show empirically, the CNet accomplishes what no pointwise regularization scheme can: it regularizes against entire continuous families of symmetries. Since we train the CNet on the entire search path, including points in U not corresponding to symmetries, the CNet regularizes against any subsets of U that the QNet has already explored, even if they are not symmetry submanifolds. Thus, the CNet minimizes the time the QNet wastes exploring parts of U already well-understood. Eventually, as the QNet explores more and more symmetries, all of the symmetry submanifolds of fixed d will be learned. The CNet will then find low estimation error everywhere, at which point the learning algorithm will terminate. Thus when classical learning is added to the QNet VQA, we find empirically that the symmetry learning process converges.

As shown in the bottom half of Figure 2, we develop a CNet with two three-dimensional convolutional layers. We design the convolutions to respect the structure of the parameterization of the PQC.

In particular, the first layer scans over each set of 3 rotation parameters per qubit per block-depth, and the second layer scans over each individual parameter. The convolutional net layer is connected to a simple three-layer fully-connected neural net with leaky rectified linear unit (ReLU) activation functions [32].

We train the CNet in path-batches; during each symmetry-learning epoch, the CNet reports Δ_C while the QNet walks Nelder-Mead paths in parameter space, corresponding to a path in U . At the end of each epoch, when a symmetry is proposed, we use batch stochastic gradient descent (SGD) on the transcript of points walked in the path. Note that the CNet cannot undergo training during an epoch of QNet symmetry search. Otherwise, the CNet will extrapolate the local path well and mislead the QNet into thinking that it has already traversed its current path. We elect to use SGD instead of an algorithm similar to Nelder-Mead for its superior empirical performance (see Appendix C).

At Every N steps on the QNet search path, the QNet pauses the search and queries the CNet. If the CNet estimation error is low, then the QNet must

adjust its search path accordingly. We propose two methods by which the QNet can redirect its search. In the first, which we call the global method, the QNet simply randomly restarts. Since the parameter dimension is

$$\dim \theta = O(Ld), \quad (14)$$

the dimensionality of search space is sufficiently large for even moderate L, d that random restarts give a substantial likelihood of landing in an as-yet traversed region of U . N is chosen to be much smaller than the number of steps to find a symmetry, but still reasonably large (~ 100), in consideration of a tradeoff between wasting too much time querying the CNet and wasting too much time searching a familiar region.

The second method, which we call the local method, alternates between finding symmetries (optimization) and maximizing the classical estimation error Δ_C (exploration). That is, instead of randomly jumping to a new region, we directly walk in the direction that appears most unfamiliar. In the exploratory phase, we use finite difference gradient descent implemented by the shift rule [33],

$$\theta \leftarrow \theta - \eta \left(\frac{\mathcal{L}_{|\psi\rangle}(\theta_{101} + \delta, \theta_{102}, \dots, \theta_{3d3}) - \mathcal{L}_{|\psi\rangle}(\theta_{101} - \delta, \theta_{102}, \dots, \theta_{3d3})}{2\delta}, \dots \right), \quad (15)$$

where η is the learning rate and δ is the finite difference parameter. We favor the shift rule over Nelder-Mead because the latter fails to update θ if all points in the neighboring simplex have similar Δ_C ; hence, no exploration occurs. By contrast, gradient descent will still update θ , allowing progressive movements away from known regions. The local method is more efficient for states that are highly symmetric (i.e. many symmetry submanifolds close to each other, for a reasonable metric of submanifolds on U). The global regularizer has an advantage on states with high-dimensional parameter spaces, wherein random restarts are generally faster than explicitly carving paths to unexplored regions.

The full learning scheme is given in Figure 2. As Nelder-Mead has been shown to be average-case efficient, under the assumption that the regularizer prevents the same manifold from being explored more than polynomially many times, our algorithm solves the symmetry learning problem efficiently. In the absence of regularization, symmetry learning may take superexponential time or even fail to terminate at all.

E. Adjustments for Near-term Practical Implementation

Due to the limitations of near-term quantum circuits, the symmetry learning algorithm as formulated above will be difficult to implement with current hardware. Specifically, both the SWAP_ϵ test and the PQC given in Fig. 4 are spatially nonlocal computations—they contain gates that span $O(L)$ qubits. In this section, we adjust the algorithm to use only local operations, thereby becoming realizable with near-term hardware, at the cost of some efficiency and universality.

A SWAP operation between two L -qubit registers is implemented via L single-qubit SWAP gates spanning $O(L)$ qubits. While recent advancement in Rydberg atom quantum computers have made long-range SWAPs a reality [34], in more general quantum hardware the SWAP gate must be decomposed into controlled-NOT gates as

$$\text{SWAP}_{ij} = \text{CNOT}_{i \rightarrow j} \text{CNOT}_{j \rightarrow i} \text{CNOT}_{i \rightarrow j}. \quad (16)$$

Hence a controlled-SWAP can be carried out in $O(L)$ Toffoli gates. However, such gates span $O(L)$ qubits.

An alternative verification procedure to the SWAP_ϵ test is via statistical comparison of the projective measurements (i.e. their distribution in a particular measurement basis) of $|\psi\rangle$ with that of $U|\psi\rangle$. For any fixed basis b , we may estimate the classical Kullback-Leibler (KL) divergence $\text{KL}_b(|\psi\rangle, U|\psi\rangle)$ between the measurement distributions of $|\psi\rangle$ and $U|\psi\rangle$ by constructing a sampling-based empirical cumulative distribution function (ECDF) for each state in the basis expansion of b . Choosing multiple bases to capture the structure of the phases, we define the quantum KL divergence as a sum over the KL divergence of each projective measurement:

$$\text{QKL}_{|\psi\rangle}(U) = \sum_b \text{KL}_b(|\psi\rangle, U|\psi\rangle). \quad (17)$$

In practice, we find that only two bases, which need not be random, are required to capture the contributions of the phases (see Appendix A). Generally, estimation of $\text{QKL}_{|\psi\rangle}(U)$ requires $O(\exp(L))$ evaluations of U just to sample every basis element at least once on average. However, if only a small (polynomial) subset of the basis elements have non-negligible amplitude, the QKL loss may be evaluated efficiently. Thus, although the exponential search is generally unavoidable, we can first search for a polynomially sparse basis by checking the z and x bases and choosing the sparser one. Further optimizations can be done via a method of binary search to find a sparse basis. A second basis can then be found by rotating the sparsest basis by a small angle (we chose $\pi/10$). Such basis optimizations should be done before beginning the learning algorithm, since checking a basis requires one sampling round, whereas the learning algorithm will use hundreds of thousands or more sampling rounds, one for each step in the search path. With preliminary basis finding, the QKL verification remains relatively efficient, and we define the near-term loss as

$$\mathcal{L}_{|\psi\rangle}(\boldsymbol{\theta}) \xrightarrow{\text{near-term}} \text{QKL}_{|\psi\rangle}(C_{L,d}(\boldsymbol{\theta})). \quad (18)$$

The second modification is of the PQC ansatz. The form given in Fig. 4 is universal, but has $\binom{L}{2}$ CNOT gates in each layer, most of which span $O(L)$ qubits. Ideally, a hardware-efficient circuit family has only gates spanning $O(1)$ qubits and scales in depth at most linearly with L . Thus, we restrict the CNOT gates to only nearest neighbors, giving a hardware efficient circuit shown in Fig. 5. Although the family may no longer be universal, or at least require much longer depths to represent a general unitary, it can still represent most symmetries of practical interest, including any expressible in terms of local couplings. For present applicability, we adopt both adjustments in this paper. Note that our algorithm

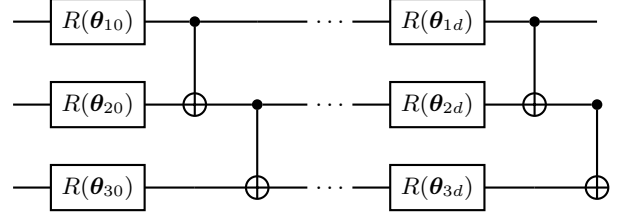


FIG. 5. Restricted quantum circuit family, of d layers, appropriate for current noisy devices capable of local operations. Cross-hairs represent controlled-NOT gates, while R gates are single-qubit rotations, parameterized by $\boldsymbol{\theta}$.

is agnostic to the choice of PQC family and loss function, so these hardware efficient modifications can be readily removed when the appropriate experimental hardware is realized.

III. RESULTS AND DISCUSSION

We demonstrate our results on four quantum states: a Bell state, the GHZ state from Eq. (9) and the planar cluster state [35], which both arise from a broader family known as stabilizer states, and the transverse Field Ising Model (TFIM) which is known as one of the simplest quantum spin chains. Much of our focus is on the GHZ state, for due to their sensitivity to decoherence, they are often utilized for characterization of noisy quantum hardware [36–38] and other quantum technologies, such as error correction, quantum metrology, and quantum communication [39, 40]. We will first visualize some symmetries of the states. We then benchmark the scaling of the algorithm with respect to L and d as well as its robustness against noise. We collect data by backend classical simulation in Qiskit [41], although our results on simulated noise indicate comparable results on quantum hardware.

A. Symmetry Visualization

The exponential size of the matrix representation of $\boldsymbol{\theta}$ implies that there is no simple method to visualize a general symmetry aside from its circuit decomposition. However, for simple cases, we can explicitly examine the geometric or analytical structure of the symmetry, which we discuss for the L -qubit GHZ state, measured in the z basis for maximal sparseness. When $d = 0$, each symmetry factors as a tensor product of 2×2 matrices, which belong to one of the submanifolds in Eq. (11). Since they are either di-

agonal or off-diagonal, we use principal component analysis (PCA) in 2 dimensions to visualize their geometric structure, illustrated in Fig 6 for $L = 3$. The diagonal and off-diagonal symmetries form distinct clusters, which is consistent with the orthogonality of their analytical representations.

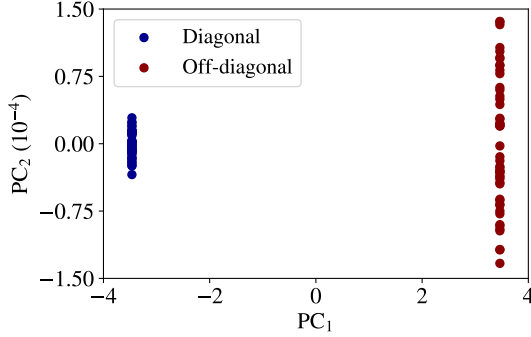


FIG. 6. PCA projection of 100 symmetries of the 3-GHZ state, found by the QNet, into 2 dimensions. The separation between them implies that the parameterizations of the manifolds are easily distinguishable, which we know analytically to be true.

Beyond $d = 0$, no such tensor product decomposition exists, so the symmetries become more difficult to visualize. (One can do so, though, by using PCA on θ itself and then clustering the resultant dataset.) However, for $L = 2$, the unitary implemented by $C_{2,d}$ may be explicitly written. A key symmetry in many entangled states is the swap symmetry, which require at least $d = 3$ to be represented, by Eq. (16). We test the algorithm on a Bell basis state

$$|\Phi\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \quad (19)$$

to determine how quickly it finds a swap symmetry. On average, after ~ 10 trials, we find many symmetries in the family

$$U = e^{i\varphi} \begin{pmatrix} \cdot & c_1 & -c_1 & \cdot \\ \cdot & \alpha & 1 - \alpha & \cdot \\ \cdot & 1 - \beta & \beta & \cdot \\ \cdot & c_2 & -c_2 & \cdot \end{pmatrix}, \quad (20)$$

where $\alpha, \beta, c_j \in \mathbb{C}$ and the dots representing arbitrary numbers to make U unitary. In the limit of $c_j, \alpha, \beta \rightarrow 0$, U reduces to the familiar SWAP operation, but the algorithm finds a more general family that may be interpreted as a partial swap operation on each of the $|01\rangle$ and $|10\rangle$ basis states. More generally, we find a similar family for the L -GHZ state, with a copy of the 2×2 submatrix at the center of Eq. (20) present in each two-qubit subspace.

As shown in Eq. (6), more symmetries are representable by $C_{L,d}$ with increasing d . Inversely, the minimum d required for the learning algorithm to find symmetries easily is itself a measure of state complexity. For example, although every $S_0[|\psi\rangle]$ is nonempty for any $|\psi\rangle$ (due to the identity matrix), the rotation symmetries among all possible single-qubit rotation operations may be sparse and thus difficult to find. Thus, relatively few epochs will successfully find a symmetry. For a given d , the average loss over many (say, 100) epochs on a given state measures the difficulty of learning symmetries at that block-depth. We can examine the average loss on various states to compare the complexity of their symmetries—that is, how large d must be before most epochs can find symmetries. Importantly, this simple measure can be done with no knowledge of the actual symmetries themselves; that is, we do not even need to look at the symmetries themselves to understand the relation between the loss and the block-depth.

We demonstrate the above procedure on the GHZ state, the planar cluster state, and the ground state of the TFIM Hamiltonian. The planar cluster state is represented as a graph state for which nodes are qubits and edges are controlled- Z gates, where Z is the Pauli σ_z operator. They are defined only for $L = \ell^2$ qubits, and are shown in Fig. 7. The ground

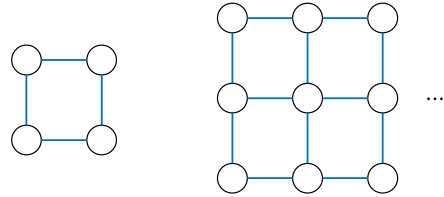


FIG. 7. Graph representation of planar cluster states for $L = \ell^2$ qubits. The nodes are qubits and the edges are controlled- Z gates. The cluster state is thus defined as $\Pi_{(a,b) \in \text{Edges}} C_{a,b} |+\rangle^{\otimes L}$ where $C_{a,b}$ is a control- Z gate between qubits a,b and $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

state of the transverse field Ising model (TFIM) Hamiltonian. The TFIM Hamiltonian is given by

$$\mathcal{H} = 0.5 \sum_i S_i^z - J \sum_i S_i^x S_{i+1}^x, \quad (21)$$

where $J = 1$ is the interaction strength and the σ_i are the Pauli spin operators on the i th qubit. While such variational methods as VQE and QAOA can be used to prepare the TFIM ground state, we did so numerically via exact diagonalization.

Our preliminary analysis shows that the TFIM state is maximally sparse in the x basis and cluster/GHZ states in the z basis. Finding 100 symmetries for each depth with $L = 4$ yields an aver-

age loss given in Fig. 8. As we might expect, the

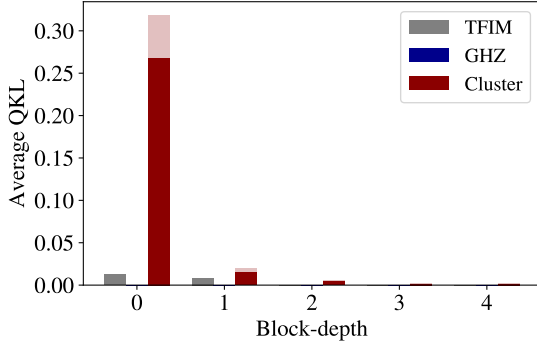


FIG. 8. Average loss for 4-qubit GHZ, TFIM, and cluster states as a function of d . The GHZ losses are all negligibly small and thus invisible. The transparent bar indicates one standard deviation.

cluster state has the most complex symmetries, requiring $d \approx 2$ to easily find symmetries with every epoch. The GHZ state is the simplest due to its well-studied $d = 0$ symmetries from Fig. 6. In practice, we can resolve the issue of hard-to-find symmetries for small d by postselecting on epochs that do find symmetries, using regularization to prevent traversing known paths. However, Fig. 8 shows that for relatively small d even complicated states like the cluster state permit most epochs to learn symmetries, evidencing that our hardware-efficient PQC ansatz remains sufficiently rich to efficiently parameterize non-trivial symmetries.

B. Query complexity

We benchmark the numerical scalability of the learning algorithm both with respect to d and L , using the L -GHZ state as a concrete example. We measure scalability in terms of the search query complexity; that is, the average number of Nelder-Mead iterations per epoch of symmetry learning. The advantage of query complexity is that it is agnostic to the verification procedure, so our results hold even without the near-term related modifications. With current devices, short-depth implementations are of the most significance. Demonstrated on the GHZ state, Fig. 9 shows that the learning algorithm scales reasonably with small d . The query complexity also scales well with respect to L on average. Such findings give numerical credence to our previous claims about the general efficiency of the learning algorithm, at least for moderately sized systems.

C. Regularization Performance

The heart of the CNet is a classical estimation of $\mathcal{L}_{|\psi\rangle}(\boldsymbol{\theta})$. The more rapidly the local landscape of a symmetry submanifold can be fully learned (i.e. estimated to low error) by the CNet, the more efficiently our scheme finds new symmetries. Measurement of classical learning efficiency manifests differently in the global and local methods. For each, we define a characteristic *learning time*; for the global scheme,

$$\tau_G = \frac{1}{N} \sum_{n=1}^N \text{argmin}_{t \geq 1} [M(\boldsymbol{\theta}_0^{(n)}) \neq M(\boldsymbol{\theta}_t^{(n)})] \quad (22)$$

where $\boldsymbol{\theta}_t^{(n)}$ is the symmetry proposed on the t th epoch of the n th fully-reset search (i.e. the CNet is re-initialized to random weights) starting from a random $\boldsymbol{\theta}$, and $M(\boldsymbol{\theta})$ is the symmetry submanifold containing $\boldsymbol{\theta}$. For large N , τ_G represents the expected number of epochs of training from Nelder-Mead paths the CNet requires to learn a randomly chosen symmetry manifold. For an ideal regularizer, $\tau_G = 1$; for a useless regularizer, $\tau_G \rightarrow \infty$.

On the other hand, for the local scheme. As the QNet alternates between optimization and exploration with no random restarts, the key metric of its performance is thus the frequency with which the QNet crosses to a different manifold and back. Hence the learning time may be defined implicitly by

$$\tau_L(T)^{-1} = \frac{1}{T} \sum_{t=1}^T \mathbb{1}[M(\boldsymbol{\theta}_{t-1}) \neq M(\boldsymbol{\theta}_t)], \quad (23)$$

where $\boldsymbol{\theta}_t$ is the symmetry on the t th optimization epoch. For large T , τ_L has the same interpretation as τ_G .

Explicit measurement of the learning time requires analytical knowledge of the set of all symmetry submanifolds. Since we have written that set for $S_0[|\text{GHZ}\rangle_L]$, we will measure the learning times for both methods on the 3-GHZ state. In the global scheme, over 100 runs we found $\tau \approx 1.2$ regardless of the steps N per CNet query, in a range of $N = 100$ to $N = 5000$. The learning time of the local method depends on the distance walked in the exploration phase, which can be measured either by the step size in the gradient descent procedure or in the number of iterations; we fix the latter and use the former, and plot the learning time (which can be interpreted as a scoring function since bigger is better) in Fig. 10 for $T = 400$, fitted by standard methods in the dashed line to guide the eye. The shift rule also requires a finite difference parameter h , but we found that the learning time does not depend significantly on it. Figure 10 illustrates the

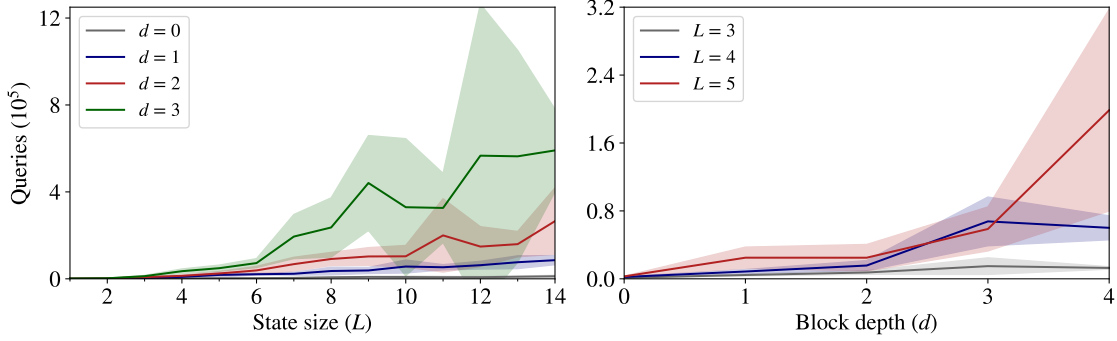


FIG. 9. Average query complexity (number of search iterations to find a symmetry) of the GHZ state as a function of state size L (left) and block-depth d (right), estimated with 10 trials per L per d . Shading represents standard deviation.

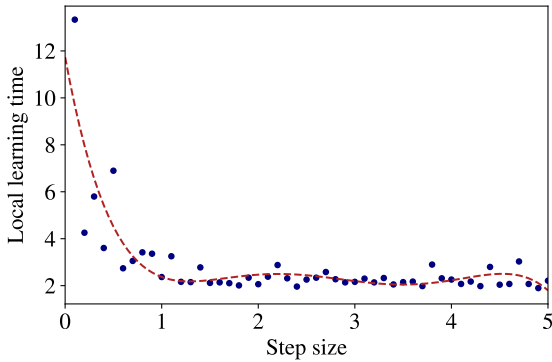


FIG. 10. Local learning time $\tau_L(400)$ (blue dots) fitted with a trend line (red, dashed) for ease of visualization. τ_L represents the average number of points on a symmetry submanifold needed to be learned before the CNet recognizes the entire submanifold—the smaller τ_L is, the more efficiently the algorithm regularizes. For larger gradient descent step sizes, the learning time saturates.

intuition that, for small step sizes/time spent exploring, the algorithm requires much more time to learn each manifold because it cannot walk sufficiently far away from a symmetry submanifold before the exploratory phase ends. For sufficiently large step size, however, (≈ 5 for 3-GHZ), the exploration has taken the QNet sufficiently far away from the known symmetry submanifold that more exploration is unnecessary. Although Fig. 10 benchmarks a simple state, the lessons taught by the results hold more generally in that a sufficiently large choice of step size is crucial to build an effective local regularizer.

D. Learning in the Presence of Noise

A final consideration for implementation in near-term devices is the robustness of the learning algorithm in the presence of noise. For simplicity, we consider a noise model, simulated classically, for which each gate (both in the state preparation and $C_{L,d}$) can incur a bit flip (erroneous X error) or reset (qubit resets to $|0\rangle$), each independently with probability p_{error} . The measurement of each qubit may also be erroneous with the same probability. The presence of such noise may be physically interpreted as thermal fluctuations that jiggle the path traversed by the QNet in \mathcal{U} . Because our QNet requires many iterations to propose a symmetry— 10^5 or more for some systems, as shown in Fig. 9—we expect that such small thermal fluctuations average out in the process, at least up to a certain threshold of p_{error} .

We examine the loss metric $\mathcal{L}_{|\psi\rangle}(\theta)$ as a function of p_{error} on the 3-GHZ state for $d = 1$ to demonstrate this effect numerically. The black curve in Fig. 11 represents the true average loss of the symmetry; that is, the QKL value between the shadows of $U|\text{GHZ}_3\rangle$ and $|\text{GHZ}_3\rangle$, where U is the symmetry proposed by the noisy PQC. Until $p_{\text{error}} \sim 10^{-2}$, the loss remains relatively low and constant. However, in practice we cannot calculate the true loss due to noisiness of state preparation. When $|\text{GHZ}_3\rangle$ is itself prepared with noise, the resultant average QKL value is given in red. As expected, they are much higher, but the difference between the two, plotted with standard smoothing methods in grey, also remains relatively constant until $p_{\text{error}} \sim 10^2$. Thus, when the error is below a reasonably high threshold, the learning algorithm continues to propose true symmetries, and the amount by which the loss increases due to the noisiness of state preparation varies little. By subtracting out this constant

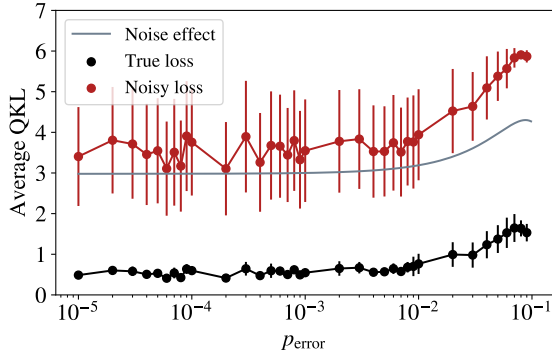


FIG. 11. Average loss over 20 symmetries proposed by a noisy circuit with respect to a noisily prepared GHZ state (red) and a noiseless GHZ state (black), which represents the true loss of the noisy symmetry. Error bars show log-standard deviation. The grey line is a fitted curve of the difference in losses evaluated with respect to a noisy state and a noiseless state.

gap, we can account for the effect of the noise and thereby continue to learn symmetries in the presence of noise. Above the threshold, the loss increases roughly linearly and slowly, and symmetries may still be classified, albeit slightly noisily.

IV. CONCLUSION & OUTLOOK

We developed an interactive hybrid quantum-classical learning algorithm that efficiently solves the symmetry learning problem—to discover and classify every symmetry of an unknown state $|\psi\rangle$ that can be represented by a certain quantum circuit family with a fixed block-depth d . We first showed that known symmetries could be verified efficiently via a quantum algorithm, and then devised a method to upgrade the verification procedure into a variational quantum learning algorithm that generated symmetries. To regularize against repetitive propositions of similar symmetries, we introduced an interactive protocol with a classical deep neural network that guided the variational quantum algorithm away from areas of unitary space $U(2^L)$ explored in previous iterations of learning. Thus, for each state we demonstrated it upon, the algorithm converged. For purposes of near-term implementation, we showed that aspects of the algorithm difficult to implement in current hardware could be replaced by presently realizable methods at the cost of some efficiency. We benchmarked our algorithm in simple cases and showed that it scales well with respect to d and L . We further showed that the learning scheme is robust against noise.

Although the parameterized quantum circuit families we discussed are universal or near-universal, a general symmetry may require a very deep circuit in the family to be represented. In practice, under a restriction of depth, we can detect more symmetries by choosing various families of circuits and running the learning algorithm on each. Since some symmetries are more easily represented by certain circuit architectures than others, such an approach maximizes the number of symmetries found empirically on near-term devices.

An important next step in the utilization of our algorithm is more robust benchmarking on various quantum hardware, such as IBM digital quantum devices or Rydberg machines. More analysis on a broader set of states is also of interest. As for regularization, a more complex protocol may allow for even more efficiency. For example, a procedure based on the Metropolis-Hastings algorithm, which stochastically combines the global and local approaches proposed in this paper, is a natural next step of study. Indeed, recent work has shown that application of similar approaches based on Markov chain Monte Carlo methods significantly improve the efficiency of many variational quantum algorithms [42].

In broader context, symmetry learning can serve as a data-driven guide to learn about the physical phenomena of quantum systems and their relation to symmetries. For example, symmetry breaking is closely related to phase transitions and other emergent properties. Symmetry learning also aligns with the goals of partial tomography in providing a characterization of a state without completely reconstructing it classically. While recent methods such as shadow tomography [43] offer an efficient and convenient way to understand properties of a state from a computational perspective, symmetry learning may be of more relevance from a physics perspective, guiding researchers in quantum machine learning and shedding light on the physics of complex, empirically created quantum systems such as spin liquids [44].

ACKNOWLEDGMENTS

The classical simulations in this paper were performed on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University. RAB acknowledges support from NSF Graduate Research Fellowship under Grant No. DGE1745303, as well as funding from Harvard University's Graduate Prize Fellowship. SFY acknowledges funding from NSF and AFOSR.

Appendix A: QKL Basis Requirement

The definition of the QKL divergence requires a choice of a number of bases. For every family of states we considered, only 2 constant bases were necessary. We show this by defining a QKL cross validation loss and showing that for 2 constant bases, the cross validation is sufficiently small.

The methodology is as follows. Choose 2 bases to train on. These may be chosen randomly. However, if one knows that $|\psi\rangle$ has most of its amplitude concentrated in the small proportion of the basis elements in some basis, that basis will be optimal to train on. The second basis may be obtained by perturbing the optimal basis by a small rotation angle. We find that a rotation about an arbitrary axis of $\pi/10$ suffices to capture all of the phase information of the state.

To verify that 2 bases suffices, choose n random bases after training and evaluate the QKL value over the random set. If it is under the desired error threshold, the two bases suffice. In our case, we found that the QKL over 3 random bases was of order $\sim 10^{-8}$ consistently.

Appendix B: Hyperparameters

We use a error tolerance of 10^{-12} for Nelder-Mead search, with a maximum allowed iterations of 10^4 for $L = 3$ qubits and up to 10^6 for $L = 15$ qubits. The prediction error threshold for the global regularization scheme is $\delta = 10^{-2}$. The CNet uses 100 nodes per layer in the fully connected neural network component. We chose these parameters for their good numerical results and rapid convergence in the systems studied in this paper, but different states or those with sizes larger than the sizes considered in our study may require different choices.

Appendix C: CNet Demonstration

It is instructive to examine the classical deep net independently to verify its ability to learn the loss metric landscape. As an example, we consider the CNet on the 3-qubit GHZ state with a block-depth of $d = 2$. We generate a simple dataset of 3000 random parameters on two bases, one the computational (z) basis and the other a perturbation by a rotation of $\pi/10$ radians about a fixed axis. For each basis, the respective CNet learns the projected classical KL value on the training set, then attempt to estimate it on a testing set of 500 values. The resulting learning curve is given in Fig. 12.

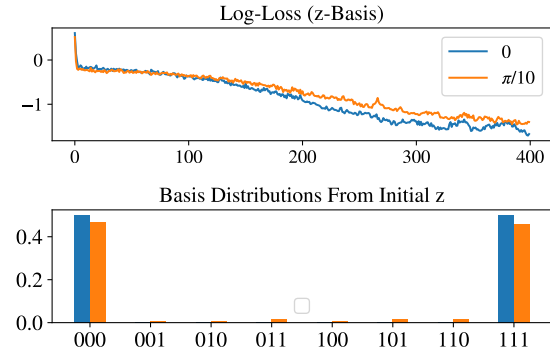


FIG. 12. Top: $\log \Delta_C$ of C Nets under each basis (0 [blue] corresponds to the z -basis). Bottom: visualization of the shadows of the 3-GHZ state.

The computational basis learns to a training error of 0.024 and incurs a cross-validation (test) error of 1.372, while the perturbed basis has a training/validation error of 0.038/1.224. The gap between the training and cross-validation error spans two orders of magnitude. Such effects, already visible in this toy example, are dramatically enhanced in practice wherein the training set is a local path rather than a random global sample, for the correlation between points is much higher, leading to a lower training loss and a higher cross-validation error. This is precisely the desired property of the CNet.

-
- [1] Y.-i. Mototake, Interpretable conservation law estimation by deriving the symmetries of dynamics from trained deep neural networks, *Phys. Rev. E* **103**, 033303 (2021).
 - [2] S. J. Wetzel, R. G. Melko, J. Scott, M. Panju, and V. Ganesh, Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks, *Phys. Rev. Research* **2**, 033499 (2020).
 - [3] S. Ha and H. Jeong,.
 - [4] Z. Liu, V. Madhavan, and M. Tegmark, Ai poincaré 2.0: Machine learning conservation laws from differential equations, arXiv preprint arXiv:2203.12610 (2022).
 - [5] Z. Liu, V. Madhavan, and M. Tegmark, Ai poincaré 2.0: Machine learning conservation laws from differential equations (2022).

- [6] S. Krippendorff and M. Syvaeri, Detecting symmetries with neural networks, *Machine Learning: Science and Technology* **2**, 015010 (2020).
- [7] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, *Scientific reports* **7**, 1 (2017).
- [8] X.-Y. Dong, F. Pollmann, X.-F. Zhang, *et al.*, Machine learning of quantum phase transitions, *Physical Review B* **99**, 121104 (2019).
- [9] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [10] Y. Zhang, P. Ginsparg, and E.-A. Kim, Interpreting machine learning of topological quantum phase transitions, *Physical Review Research* **2**, 023283 (2020).
- [11] D.-L. Deng, X. Li, and S. D. Sarma, Machine learning topological states, *Physical Review B* **96**, 195145 (2017).
- [12] N. K  ming, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg, Unsupervised machine learning of topological phase transitions from experimental data, *Machine Learning: Science and Technology* **2**, 035037 (2021).
- [13] D. Luo, Z. Chen, K. Hu, Z. Zhao, V. M. Hur, and B. K. Clark, Gauge invariant autoregressive neural networks for quantum lattice models, *arXiv preprint arXiv:2101.07243* (2021).
- [14] D. Luo, G. Carleo, B. K. Clark, and J. Stokes, Gauge equivariant neural networks for quantum lattice gauge theories, *Physical Review Letters* **127**, 276402 (2021).
- [15] 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, *arXiv preprint arXiv:2205.06217* (2022).
- [16] M. Larocca, F. Sauvage, F. M. Sbahi, G. Verdon, P. J. Coles, and M. Cerezo, Group-invariant quantum machine learning, *arXiv preprint arXiv:2205.02261* (2022).
- [17] Z. Zhou, Y. Du, X. Tian, and D. Tao, Qaoa-in-qaoa: solving large-scale maxcut problems on small quantum machines, *arXiv preprint arXiv:2205.11762* (2022).
- [18] M. Paris and J. Rehacek, *Quantum state estimation*, Vol. 649 (Springer Science & Business Media, 2004).
- [19] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [20] G. Torlai, B. Timar, E. P. L. van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuleti  , M. D. Lukin, R. G. Melko, and M. Endres, Integrating neural networks with a quantum simulator for state reconstruction, *Physical Review Letters* **123**, 230504 (2019), *arXiv:1904.08441*.
- [21] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nature Physics* **14**, 447 (2018).
- [22] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Reconstructing quantum states with generative models, *Nature Machine Intelligence* **1**, 155 (2019).
- [23] J. Wang, Z.-Y. Han, S.-B. Wang, Z. Li, L.-Z. Mu, H. Fan, and L. Wang, Scalable quantum tomography with fidelity estimation, *Physical Review A* **101**, 032321 (2020).
- [24] J. Carrasquilla and G. Torlai, How to use neural networks to investigate quantum many-body physics, *PRX Quantum* **2**, 040201 (2021).
- [25] P. Cha, P. Ginsparg, F. Wu, J. Carrasquilla, P. L. McMahon, and E.-A. Kim, Attention-based quantum tomography, *Machine Learning: Science and Technology* **3**, 01LT01 (2022).
- [26] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert, and J. Preskill, Provably efficient machine learning for quantum many-body problems, *arXiv:2106.12627 [quant-ph]* (2021).
- [27] A. M. Gomez, S. F. Yelin, and K. Najafi, *Reconstructing quantum states using basis-enhanced born machines* (2022).
- [28] We restrict ourselves to pure states $|\psi\rangle$ and system-isolated operators (i.e. acting only on $|\psi\rangle$) for simplicity. However, The algorithm generalizes readily to a density matrix and Krauss operator-sum formalism.
- [29] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, Quantum fingerprinting, *Physical Review Letters* **87**, 167902 (2001).
- [30] V. V. Shende, I. L. Markov, and S. S. Bullock, Minimal universal two-qubit cnot-based circuits, *arXiv preprint quant-ph/0308033* (2003).
- [31] J. A. Nelder and R. Mead, A simplex method for function minimization, *The computer journal* **7**, 308 (1965).
- [32] A. F. Agarap, Deep learning using rectified linear units (relu), *arXiv preprint arXiv:1803.08375* (2018).
- [33] C. Grossmann, H.-G. Roos, and M. Stynes, *Numerical treatment of partial differential equations*, Vol. 154 (Springer, 2007).
- [34] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, *et al.*, A quantum processor based on coherent transport of entangled atom arrays, *arXiv preprint arXiv:2112.03923* (2021).
- [35] D. Gottesman and I. L. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, *Nature* **402**, 390 (1999).
- [36] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, *et al.*, Generation and manipulation of schr  dinger cat states in rydberg atom arrays, *Science* **365**, 570 (2019).
- [37] C. Song, K. Xu, H. Li, Y.-R. Zhang, X. Zhang, W. Liu, Q. Guo, Z. Wang, W. Ren, J. Hao, and *et al.*, Generation of multicomponent atomic schr  dinger cat states of up to 20 qubits, *Science* **365**, 574-577 (2019).

- [38] K. X. Wei, I. Lauer, S. Srinivasan, N. Sundaresan, D. T. McClure, D. Toyli, D. C. McKay, J. M. Gambetta, and S. Sheldon, Verifying multipartite entangled greenberger-horne-zeilinger states via multiple quantum coherences, *Phys. Rev. A* **101**, 032343 (2020).
- [39] Z. Zhao, Y.-A. Chen, A.-N. Zhang, T. Yang, H. J. Briegel, and J.-W. Pan, *Experimental demonstration of five-photon entanglement and open-destination teleportation* (2004).
- [40] L. Pezzè, A. Smerzi, M. K. Oberthaler, R. Schmied, and P. Treutlein, Quantum metrology with nonclassical states of atomic ensembles, *Rev. Mod. Phys.* **90**, 035005 (2018).
- [41] M. S. A. et. al., *Qiskit: An open-source framework for quantum computing* (2021).
- [42] T. L. Patti, O. Shehab, K. Najafi, and S. F. Yelin, Markov chain monte-carlo enhanced variational quantum algorithms, arXiv preprint arXiv:2112.02190 (2021).
- [43] S. Aaronson, Shadow tomography of quantum states, *SIAM Journal on Computing* **49**, STOC18 (2019).
- [44] G. Semeghini, H. Levine, A. Keesling, S. Ebadi, T. T. Wang, D. Bluvstein, R. Verresen, H. Pichler, M. Kalinowski, R. Samajdar, A. Omran, S. Sachdev, A. Vishwanath, M. Greiner, V. Vuletić, and M. D. Lukin, Probing topological spin liquids on a programmable quantum simulator, *Science* **374**, 1242 (2021).