Dynamical simulation via quantum machine learning with provable generalization

Joe Gibbs,[1,2,*] Zoë Holmes,[3,4,*] Matthias C. Caro,[5,6,7,8] Nicholas Ezzell,[3,9] Hsin-Yuan Huang,[8,10] Lukasz Cincio,[11]
Andrew T. Sornborger,[3] and Patrick J. Coles[11]

[1]*Department of Physics, University of Surrey, Guildford GU2 7XH, United Kingdom*
[2]*AWE, Aldermaston, Reading RG7 4PR, United Kingdom*
[3]*Information Sciences, Los Alamos National Laboratory, Los Alamos, New Mexico, USA*
[4]*Institute of Physics, Ecole Polytechnique Fédéderale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland*
[5]*Department of Mathematics, Technical University of Munich, Garching, Germany*
[6]*Munich Center for Quantum Science and Technology (MCQST), Munich, Germany*
[7]*Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, Berlin, Germany*
[8]*Institute for Quantum Information and Matter, Caltech, Pasadena, California, USA*
[9]*Department of Physics & Astronomy, University of Southern California, Los Angeles, California, USA*
[10]*Department of Computing and Mathematical Sciences, Caltech, Pasadena, California, USA*
[11]*Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico, USA*

(Received 4 October 2022; revised 8 September 2023; accepted 11 January 2024; published 5 March 2024)

Much attention has been paid to dynamical simulation and quantum machine learning (QML) independently as applications for quantum advantage, while the possibility of using QML to enhance dynamical simulations has not been thoroughly investigated. Here we develop a framework for using QML methods to simulate quantum dynamics on near-term quantum hardware. We use generalization bounds, which bound the error a machine learning model makes on unseen data, to rigorously analyze the training data requirements of an algorithm within this framework. Our algorithm is thus resource efficient in terms of qubit and data requirements. Furthermore, our preliminary numerics for the XY model exhibit efficient scaling with problem size, and we simulate 20 times longer than Trotterization on IBMQ-Bogota.

DOI: 10.1103/PhysRevResearch.6.013241

I. INTRODUCTION

The exponential speedup of dynamical quantum simulation provided the original motivation for quantum computers [1,2]. In the long term, large-scale quantum simulations are expected to transform fields such as materials science, chemistry, and high-energy physics. Nearer term, since efficient classical dynamical simulation methods are lacking (in contrast to those for computing static quantum properties like electronic structure), dynamical simulation may plausibly be one of the first applications to see quantum advantage.

Achieving near-term quantum advantage for dynamics will require long-time simulations on noisy intermediate-scale quantum (NISQ) hardware [3]. Standard methods like Trotterization grow the circuit depth in proportion to the simulation time, ultimately running into the decoherence time of the NISQ device [4,5]. Fast-forwarding methods for long-time simulations on NISQ devices have recently been introduced [6–9], but are limited by various inefficiencies (e.g., qubit and data requirements). Here, we address these inefficiencies, potentially opening the door for near-term quantum advantage.

In parallel to these developments, quantum machine learning (QML) [10,11] has emerged as another potential application for quantum advantage [12]. At its core, QML involves using classical or quantum data to train a parameterized quantum circuit. A number of promising paradigms for training are being pursued, including variational quantum algorithms using training data [13], quantum generative adversarial networks [14,15], and quantum kernel methods [11] (to name just a few). Here we seek to combine the potential of QML and dynamical simulation, by leveraging recent advances in QML to reduce resource requirements for dynamical simulation.

To assess the scalability of QML methods, as well as their applicability to real world problems, it is critical to understand their training data requirements, quantified by so-called generalization bounds [16–29]. These provide bounds on the error a machine learning model makes on unseen data, as a function of the amount of data the model is trained on and of the training performance. In this paper, we assess the training data requirements of QML approaches to dynamical simulation.

Our analysis provides the groundwork for a QML-inspired algorithm for dynamical simulation that we call the “resource-efficient fast-forwarding” algorithm (REFF). This algorithm uses training data to learn a circuit that allows for fast forwarding, where long-time simulations can be performed using

[*] These authors contributed equally to this work.

a fixed-depth circuit. The REFF algorithm is efficient in the amount of training data required. It is also qubit efficient in the sense that simulating an n-qubit system only requires n qubits, in contrast to earlier paper [6], which required $2n$ qubits. We use generalization bounds to rigorously lower bound the final simulation fidelity as a function of the amount of training data used, the optimization quality (i.e., final cost obtained), and the simulation time. This analysis is complemented by numerical implementations, as well as a demonstration of our algorithm on IBMQ-Bogota.

II. GENERAL FRAMEWORK

Given a set of initial states $\mathcal{S}_\rho$ and an n-qubit Hamiltonian H, the goal of dynamical simulation is to predict the evolution of a set of observables $\mathcal{S}_O$ up to time T. A promising approach in the NISQ era is to use QML to fit a time-dependent quantum model that can be extrapolated to long-simulation times using a short-depth quantum circuit [6–8,30]. This is valuable in the NISQ era since high noise levels constrain the depth of circuits that may be simulated. More concretely, the aim is to find some time-dependent quantum neural network (QNN) $V_t(\boldsymbol{\alpha})$, and optimized parameters $\boldsymbol{\alpha}_{\text{opt}}$, such that for any time $t < T$,

$$\mathrm{Tr}[V_t(\boldsymbol{\alpha}_{\text{opt}})\rho V_t(\boldsymbol{\alpha}_{\text{opt}})^\dagger O] \approx \mathrm{Tr}[e^{-iHt}\rho e^{iHt} O] \tag{1}$$

for any $\rho \in \mathcal{S}_\rho$ and $O \in \mathcal{S}_O$. Possible time-dependent QNNs include those formed from the Newton-Cartan decomposition of H [30,31], or a diagonalization of H [7,32,33], or of the propagator for short-time evolution [6,8]. Figure 1 depicts this framework.

In training $V_t(\boldsymbol{\alpha})$, the most appropriate choice in training data will depend on the set of initial states $\mathcal{S}_\rho$ and observables $\mathcal{S}_O$, but typically will be generated by the properties of the system at short times. For example, if one is interested in only knowing the evolution of a single observable, the training data could consist of the evolution of this observable for some subset of the target states up to some short time Δt. Alternatively, if one is interested in simulating the evolution of any possible n qubit observable, it would be natural to use training data consisting of N pairs of input-output states $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N$ where the output states are generated by evolving the input states for some short time, i.e., $|\Phi^{(j)}\rangle = U_{\Delta t}|\Psi^{(j)}\rangle$ where $U_{\Delta t} \approx e^{-iH\Delta t}$ is a gate sequence (such as a Trotterization) that approximates the true time evolution.

The training data is initially used to train the QNN to reproduce the properties of the system at short times, i.e., to ensure that Eq. (1) holds for times $t \leqslant \Delta t$. Crucially, while trained on a subset of the target states and/or observables, the hope is that the learned QNN *generalizes* to the unseen target data, i.e., it well reproduces Eq. (1) for short times ($t \lesssim \Delta t$) for any $\rho \in \mathcal{S}_\rho$ and $O \in \mathcal{S}_O$. The properties of the system at some longer time T can then be extrapolated via $V_T(\boldsymbol{\alpha}_{\text{opt}})$.

Generalization bounds quantify the performance of a QNN on unseen data, after optimization on a limited training set. In particular, the generalization error measures how much the performance on new data differs from the performance on the training data. Reference [24] has shown that if the parameterized quantum circuit has K trainable local gates, the generalization error scales at worst as $\tilde{\mathcal{O}}(\sqrt{K/N})$. Crucially, as established in Ref. [26], these training states may be product states. Below we use these quantum generalization bounds to quantify the performance of dynamical simulation via QML.

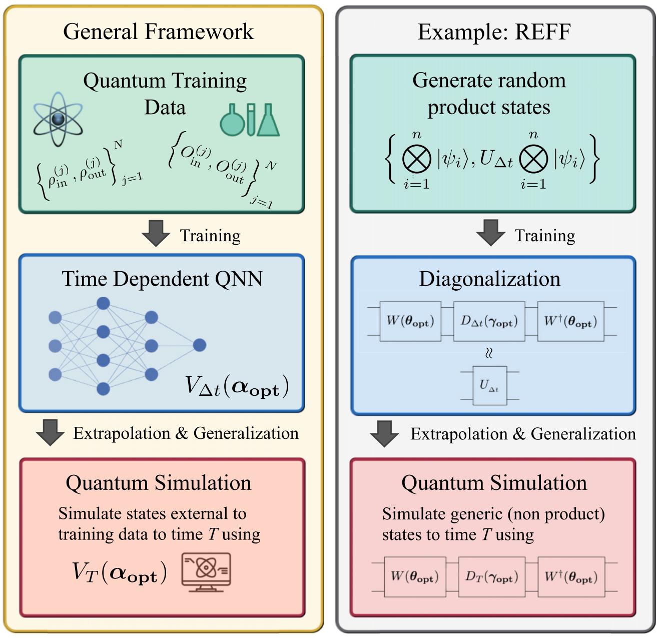
General Framework
Quantum Training Data
$\left\{\rho_{\text{in}}^{(j)}, \rho_{\text{out}}^{(j)}\right\}_{j=1}^N$
$\left\{O_{\text{in}}^{(j)}, O_{\text{out}}^{(j)}\right\}_{j=1}^N$
Training
Time Dependent QNN
$V_{\Delta t}(\boldsymbol{\alpha}_{\text{opt}})$
Extrapolation & Generalization
Quantum Simulation
Simulate states external to training data to time T using
$V_T(\boldsymbol{\alpha}_{\text{opt}})$
Example: REFF
Generate random product states
$\left\{\bigotimes_{i=1}^n |\psi_i\rangle, U_{\Delta t}\bigotimes_{i=1}^n |\psi_i\rangle\right\}$
Training
Diagonalization
$W(\boldsymbol{\theta}_{\text{opt}})$
$D_{\Delta t}(\gamma_{\text{opt}})$
$W^\dagger(\boldsymbol{\theta}_{\text{opt}})$
$\approx$
$U_{\Delta t}$
Extrapolation & Generalization
Quantum Simulation
Simulate generic (non product) states to time T using
$W(\boldsymbol{\theta}_{\text{opt}})$
$D_T(\gamma_{\text{opt}})$
$W^\dagger(\boldsymbol{\theta}_{\text{opt}})$

FIG. 1. QML framework for dynamical simulation. Our general framework (left panel) consists of using quantum training data, e.g., composed of input-output state pairs and/or input-output observable pairs, to train a time-dependent QNN $V_t(\boldsymbol{\alpha})$. Typically the training occurs at a short time Δt, resulting in the trained QNN, $V_{\Delta t}(\boldsymbol{\alpha}_{\text{opt}})$. The evolution of the system at some longer time T is extrapolated via $V_T(\boldsymbol{\alpha}_{\text{opt}})$. The resource efficient fast-forwarding (REFF) algorithm (right panel) is an illustrative example of this framework. The training data consists of Haar-random product states as inputs and then the time-evolved states as outputs, i.e., evolved under $U_{\Delta t} \approx \exp(-iH\Delta t)$. The time-dependent QNN is a parameterized quantum circuit in a diagonal form, $W(\boldsymbol{\theta})D_t(\boldsymbol{\gamma})W^\dagger(\boldsymbol{\theta})$. Hence training the QNN amounts to approximately diagonalizing the short-time evolution $U_{\Delta t}$, resulting in the trained QNN, $W(\boldsymbol{\theta}_{\text{opt}})D_{\Delta t}(\boldsymbol{\gamma}_{\text{opt}})W^\dagger(\boldsymbol{\theta}_{\text{opt}})$. This model can simulate the evolution of arbitrary input states up to time T via $W(\boldsymbol{\theta}_{\text{opt}})D_T(\boldsymbol{\gamma}_{\text{opt}})W^\dagger(\boldsymbol{\theta}_{\text{opt}})$.

REFF algorithm. For the remainder of this letter, we focus on a time-dependent QNN of the following form:

$$V_t(\boldsymbol{\alpha}) = W(\boldsymbol{\theta})D_t(\boldsymbol{\gamma})W^\dagger(\boldsymbol{\theta}), \tag{2}$$

where $\boldsymbol{\alpha} = (\boldsymbol{\theta}, \boldsymbol{\gamma})$ and $W(\boldsymbol{\theta})$ is a time-independent unitary. $D_t(\boldsymbol{\gamma})$ is a time-dependent unitary that is diagonal in the standard basis. Since D is diagonal, M applications of D_t are equivalent to one application of D_{Mt}, i.e., $D_t(\boldsymbol{\gamma})^M = D_{Mt}(\boldsymbol{\gamma})$ for any positive integer M. We will train $V_t(\boldsymbol{\alpha})$ at time $t = \Delta t$ to obtain the trained QNN, $V_{\Delta t}(\boldsymbol{\alpha}_{\text{opt}})$. If this well approximates the target unitary, i.e., $V_{\Delta t}(\boldsymbol{\alpha}_{\text{opt}}) \approx U_{\Delta t}$, then we have learned an approximate diagonalization of $U_{\Delta t}$ and the long-time simulation e^{-iHT} can be approximated using the fixed depth circuit $V_T(\boldsymbol{\alpha}_{\text{opt}})$ [6,8]. Below we formally bound the fidelity between the simulated evolution and the exact evolution.

In what follows we consider the (most difficult) task of learning the dynamics for all input states and all observables, i.e., where S_ρ is the whole Hilbert space and S_O is the set of all positive operator valued measure (POVM) elements. (We remark, however, that our analysis can be extended to dynamics within subspaces, such as subspaces that preserve certain symmetries.) For this task, as discussed above, a natural choice in training data would be N pairs of input-output states $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, U_{\Delta t}|\Psi^{(j)}\rangle)\}_{j=1}^{N}$, where each input state is evolved by a Trotter operator of fixed Δt. A simple choice for the input training states $|\Psi^{(j)}\rangle$ would be to use (Haar) random n-qubit states. However, such random states will typically be highly entangled, requiring deep circuits to prepare, and thus are unsuitable for NISQ hardware. A more promising approach is to use tensor products of (Haar) random single-qubit states, which are preparable using only single-qubit gates, and so induce less noise. Thus we suppose that $U_{\Delta t}$ is learned using a set of *product* input states, i.e., that the training data has the form

$$\mathcal{D}_{\rm P}(N) := \left\{\left(\left|\Psi_{\rm P}^{(j)}\right\rangle, \left|\Phi_{\rm P}^{(j)}\right\rangle\right)\right\}_{j=1}^{N} \quad \text{with} \quad \left|\Psi_{\rm P}^{(j)}\right\rangle = \bigotimes_{i=1}^{n} \left|\psi_i^{(j)}\right\rangle, \tag{3}$$

where the states $\{|\psi_i^{(j)}\rangle\}_{i=1}^{N}$ are independently drawn from the single-qubit Haar distribution [34].

There is freedom in how exactly this training data is used to learn $U_{\Delta t}$ but a natural approach would be to minimize the distance between the target output state $|\Phi_{\rm P}^{(j)}\rangle = U_{\Delta t}|\Psi_{\rm P}^{(j)}\rangle$ and the hypothesized output state $V_{\Delta t}(\boldsymbol{\alpha})|\Psi_{\rm P}^{(j)}\rangle$, averaged over the N states in the training set. That is, we minimize the following cost:

$$C^{\rm G}_{\mathcal{D}_{\rm P}(N)}(\boldsymbol{\alpha}) = \frac{1}{4N}\sum_{j=1}^{N} \left\| \left|\Phi_{\rm P}^{(j)}\right\rangle\!\left\langle\Phi_{\rm P}^{(j)}\right| - V\left|\Psi_{\rm P}^{(j)}\right\rangle\!\left\langle\Psi_{\rm P}^{(j)}\right|V^{\dagger} \right\|_1^2, \tag{4}$$

where for compactness we write $V_{\Delta t}(\boldsymbol{\alpha}) \equiv V$ and $\|\cdot\|_1$ denotes the trace norm. We can rewrite this cost in terms of the fidelity as

$$C^{\rm G}_{\mathcal{D}_P(N)}(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{j=1}^{N}\left(1 - \left|\left\langle\Phi_{\rm P}^{(j)}\right|V\left|\Psi_{\rm P}^{(j)}\right\rangle\right|^2\right), \tag{5}$$

which can be measured on a quantum computer using the Loschmidt echo circuit [28] shown in Appendix C.

While natural and intuitive, Eq. (5) is a global cost [35] since it is measured via the global measurements $\mathbb{1} - |\Psi_{\rm P}^{(j)}\rangle\langle\Psi_{\rm P}^{(j)}|$ on all n qubits of the states $V^{\dagger}|\Phi_{\rm P}^{(j)}\rangle\langle\Phi_{\rm P}^{(j)}|V$ for $j = 1, \ldots, N$. Hence, it encounters exponentially vanishing gradients, known as barren plateaus [35–46]. To mitigate such trainability issues, we advocate instead training using a local version of the cost of the form

$$C^{\rm L}_{\mathcal{D}_{\rm P}(N)}(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{j=1}^{N} {\rm Tr}\left[V^{\dagger}\left|\Phi_{\rm P}^{(j)}\right\rangle\!\left\langle\Phi_{\rm P}^{(j)}\right|V O_L^{(j)}\right], \tag{6}$$

with $O_L^{(j)} := \mathbb{1} - \frac{1}{n}\sum_{i=1}^{n}|\psi_i^{(j)}\rangle\langle\psi_i^{(j)}| \otimes \mathbb{1}_{\bar{i}}$, where $\bar{i}$ denotes the set of all qubits except for i. This cost is faithful [47], i.e., vanishing if and only if $U_{\Delta t} = V$, but crucially is also trainable as long as the ansatz is not too deep [35].

We call this algorithm, which uses the local product state cost, Eq. (6), to learn a diagonalization QNN of the short-time evolution unitary of a system and thereby fast forward its evolution, the resource-efficient fast-forwarding (REFF) algorithm. The algorithm is both efficient in terms of qubit usage (requiring only n qubits to simulate an n-qubit system) and, as shown below, in terms of quantum data usage.

III. SIMULATION ERROR BOUNDS

An operationally meaningful measure of the quality of the simulation via $V_T(\boldsymbol{\alpha}_{\rm opt})$ is given by the average simulation fidelity [48]

$$\overline{F}(\boldsymbol{\alpha}_{\rm opt}, T) = \int_{\psi} |\langle\psi|V_T(\boldsymbol{\alpha}_{\rm opt})^{\dagger}e^{-iHT}|\psi\rangle|^2\, d\psi, \tag{7}$$

where the integral is over states $|\psi\rangle$ chosen according to the n-qubit Haar measure. In this section, we lower bound the final simulation fidelity $\overline{F}(\boldsymbol{\alpha}_{\rm opt}, T)$ for the REFF algorithm, allowing for an arbitrary optimization procedure. Our bound depends on the time simulated T, the amount of training data N, the learning error over the training data [i.e., the minimum cost achieved $C^{\rm G}_{\mathcal{D}_{\rm P}(N)}(\boldsymbol{\alpha}_{\rm opt})$ or $C^{\rm L}_{\mathcal{D}_{\rm P}(N)}(\boldsymbol{\alpha}_{\rm opt})$], and the error incurred from approximating the short-time evolution $e^{-iH\Delta t}$ with the gate sequence $U_{\Delta t}$, that is $\epsilon = \|U_{\Delta t} - e^{-iH\Delta t}\|_2$. The Trotter error can be reduced by using a higher-order approximation of the Trotter unitary.

Theorem 1 (Simulation error for product-state training—Informal). Consider a QNN $V_t(\boldsymbol{\alpha})$ given by Eq. (2) and composed of K parameterized local gates. When trained with the global cost $C^{\rm G}_{\mathcal{D}_{\rm P}(N)}$ using training data $\mathcal{D}_{\rm P}(N)$, the simulation fidelity after time $T = M\Delta t$, for a positive integer M, satisfies

$$\overline{F}(\boldsymbol{\alpha}_{\rm opt}, T) \geqslant 1 - 8M^2\left[\tilde{\epsilon}^2 + C^{\rm G}_{\mathcal{D}_{\rm P}(N)}(\boldsymbol{\alpha}_{\rm opt})\right] - \mathcal{O}\big(M^2 f(K,N)\big), \tag{8}$$

with high probability over the choice of random product state data. Here $f(K,N) := \sqrt{\frac{K\log(K)}{N}}$ and $\tilde{\epsilon} := \frac{\epsilon}{2\sqrt{(2^n+1)}}$ with $\epsilon = \|U_{\Delta t} - e^{-iH\Delta t}\|_2$ the Trotter error.

Alternatively, if the local cost $C^{\rm L}_{\mathcal{D}_{\rm P}(N)}$ is used for training, Eq. (8) holds with $C^{\rm G}_{\mathcal{D}_{\rm P}(N)}(\boldsymbol{\alpha}_{\rm opt}) \rightarrow nC^{L}_{\mathcal{D}_{\rm P}(N)}(\boldsymbol{\alpha}_{\rm opt})$ and $\mathcal{O}(M^2 f(K,N)) \rightarrow \mathcal{O}(nM^2 f(K,N))$.

Theorem 1 implies that the fast-forwarded simulation fidelity deviates from 1 at worst quadratically in the number of fast-forwarding steps M. That is, if we can predict the short-time dynamics, we should also be able to approximately predict the longer-time dynamics. We highlight that Theorem 1 requires $C^{\rm G}_{\mathcal{D}_P(N)}(\boldsymbol{\alpha}_{\rm opt})$ to be smaller than $\mathcal{O}(\frac{1}{M^2})$; this condition is that a high-fidelity simulation for a longer time requires the compilation to be correspondingly more accurate. Moreover, inverting Eq. (8) provides a means of bounding the number of product state training pairs and the minimal cost function sufficient to guarantee a given desired simulation fidelity and total simulation time.

In particular, Theorem 1 implies that a high fidelity may be achieved whenever the number of training pairs N is effectively of the same order as M^4Kn^2, i.e., scales polynomially in the product of the number of fast-forwarding steps, the number of parameters used for the diagonalization, and the system size. Thus, the success of this QML-inspired approach to simulation depends critically on the number of parameters required to approximately diagonalize the short-time evolution of a system.

For example, Ref. [49] analytically constructs the circuits required to exactly diagonalize the Ising chain. These circuits, which can simulate the XY model and Kitaev's honey-comb lattice, require n gates for the diagonal matrix D and $\mathcal{O}(n^2)$ gates for W. Thus, assuming the correct discrete structure is known (or can be approximately found variationally), such models are captured by ansätze with only a polynomial number of parameters. This provides hope that other systems may similarly be diagonalized with favorable parameter number scalings.

IV. IMPLEMENTATIONS

In practice, even less training data than that suggested by our theoretical bounds may suffice. Here we numerically investigate the minimal training data required for two oft-studied Hamiltonians. We train on random product input states, and to simplify notation we write $C^{\mathrm{L/G}}_{\mathrm{REFF}} \equiv C^{\mathrm{L/G}}_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha})$. To evaluate the quality of the learned diagonalization and the resulting simulation, we use the average fidelity $\overline{\mathcal{F}}_M$ between the learned diagonalization and a second-order Trotterized unitary $U^M_{\Delta t}$. [This is similar to Eq. (7), but with $U^M_{\Delta t}$ replacing e^{-iHT}.]

Let us consider the 4-qubit Heisenberg Hamiltonian $H = \sum_{i=1}^{4} \boldsymbol{S}_i \cdot \boldsymbol{S}_{i+1}$ with periodic boundary conditions. For a set of input states composed of 3, 4, or 5 Haar-random product states, we minimized the $C^{\mathrm{G}}_{\mathrm{REFF}}$ cost and simultaneously tracked the average gate fidelity between the ansatz [50] and the Trotter unitary (with $\Delta t = 0.1$), as shown in Fig. 2(a). The optimization was repeated for 10 different sets of training data, from which we computed the geometric mean and standard deviation (arithmetic mean and standard deviation in logspace). We found that five input states were sufficient to perform a full Hilbert space diagonalization, with all 10 runs of five training states achieving $1 - \overline{\mathcal{F}}_1 < 10^{-6}$ once $C^{\mathrm{G}}_{\mathrm{REFF}}$ had reached 10^{-8}. As shown in the inset, the QNN trained with $k = 5$ states generalized well, producing a long-time high-fidelity simulation of the Hamiltonian evolution. To give evidence for the wider applicability of the algorithm, we present further numerical simulations in Appendix C of REFF applied to two systems exhibiting frustrated magnetism.

Next, we study the scaling with system size of the number of training states required to simulate the XY model. Figure 2(b) shows the results of using REFF to diagonalize and fast forward the XY Hamiltonian, $H = \sum_{i=1}^{n-1} X_i X_{i+1} + Y_i Y_{i+1}$ with open boundary conditions, for $n \in \{4, 6, \ldots, 12\}$. Here we use the local training cost, Eq. (6), to help mitigate the exponential suppression of gradients as the system size grows due to barren plateaus. In all system sizes tested, as shown by the overlap of the lines and markers, only a single training state was required to diagonalize over the entire Hilbert space. The inset of Fig. 2(b) indicates that the final simulation error, as measured by $1 - \overline{\mathcal{F}}_M$, scales subquadratically with time, as predicted. In addition, we observe efficient (i.e., polynomial) scaling with n for the gate count required for diagonalization. For further elaboration on this point, and as well as additional implementation details, see Appendix C. We note that our simulation fidelities are computed with respect to the Trotterized unitary, to quantify the quality of the learning of the target unitary. The Trotterization has an associated Trotter error that will cause a deviation from the true dynamics of the Hamiltonian; however, increasing the order of the Trotter approximation can arbitrarily decrease this error. This effect on the simulation fidelity is numerically explored further in Appendix C.

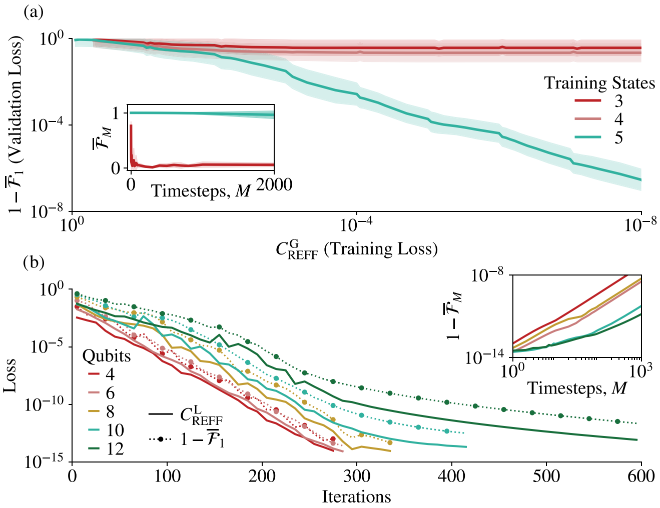

FIG. 2. Numerical simulations. (a) REFF is used to diagonalize the 4-qubit Heisenberg Hamiltonian with periodic boundary conditions. Only five Haar-random product training states are required to generalize over the whole Hilbert space, measured by the decrease in simulation infidelity $1 - \overline{\mathcal{F}}_1$. The inset shows that the average fidelity $\overline{\mathcal{F}}_M \equiv \overline{\mathcal{F}}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\Delta t)$ in this case remains over 0.95 for 2000 time steps. (b) REFF is applied to increasing sizes of the XY model. For all sizes tested, a single Haar-random product training state was sufficient to achieve a machine precision simulation infidelity. For each system size, the ansatz is saved upon reaching $C^{\mathrm{L}}_{\mathrm{REFF}} = 10^{-14}$ and used to fast forward the evolution, as shown in the inset.

To demonstrate the suitability of REFF for near-term hardware, we implemented REFF to diagonalize and fast forward a 2-qubit spin chain, described by the XY Hamiltonian. The decrease in the noise-free cost as the optimization progresses shows the algorithm successfully learning a diagonalization for the Hamiltonian, and the separation between the noisy cost measured on the quantum computer and the noise-free cost computed classically demonstrates the noise resilience of the algorithm. The clear alignment between $C^{\mathrm{G}}_{\mathrm{REFF}}$ and $1 - \overline{\mathcal{F}}_1$ in Fig. 3(a) shows that, whilst only using a single unentangled training state, a full approximate diagonalization has been successfully learned. This then enables the high-fidelity fast forwarding shown in Fig. 3(b). Namely, for random input states we achieved, on average, a fidelity of 0.8 for 94 time steps [51]. This is a factor of 23.8 improvement on the standard Trotter method, which has a fidelity of less than 0.8 after 4 time steps.

V. DISCUSSION

In this paper, we introduced a framework for leveraging the power of QML for dynamical quantum simulations. The core idea is that quantum training data may be used to train a

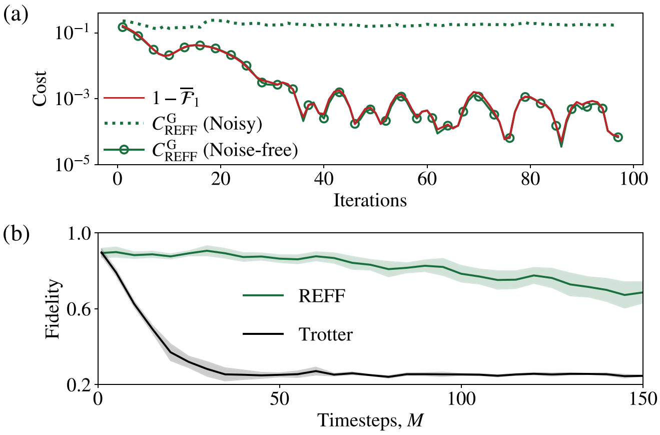

FIG. 3. Quantum hardware implementation. (a) The 2-qubit XY Hamiltonian is diagonalized via REFF on ibmq_bogota using 2^{16} measurement shots per circuit. The noisy $C^{\rm G}_{\rm REFF}$ cost is measured on ibmq_bogota, whereas the noise-free $C^{\rm G}_{\rm REFF}$ cost and $1-\overline{\mathcal{F}}_1$ are computed classically. (b) After training, the fast-forwarded performance is compared to the iterated Trotter method for five Haar-random product states and we plot the mean and standard deviations.

time-dependent QNN, which can then predict the evolution of the properties of the target system at long times using a short-depth circuit. By way of example, we introduced the REFF algorithm, which uses training data composed of product-state inputs and corresponding time-evolved outputs to learn an approximate diagonalization of the short-time evolution of the system. We showed that generalization bounds provide a tool to rigorously ground this QML-driven approach to quantum simulation. Specifically, for REFF we proved that a high-fidelity simulation may be achieved with a number of training pairs N that scales polynomially in the product of the number of fast-forwarding steps, the number of parameters of the QNN, and the system size.

We stress that we do not claim to prove that it will always be possible to successfully train. Indeed, no fast-forwarding theorems [52] imply that this will not always be possible. Nonetheless, we are optimistic that it will be possible in certain cases of interest (for a discussion see Appendix Note 5 of Ref. [6]). To support this, our approach is successfully employed in a quantum hardware demonstration, numerical simulations of frustrated models and large simulations for the XY model. Further work will investigate methods for how to construct diagonalizing circuits for more interesting interacting Hamiltonians.

While our error analysis and implementations focus on REFF, the framework is much more general. In particular, the product state compilation method advocated here is highly transferable. Important future steps include investigating alternative ansätze for time-dependent QNNs and alternative forms of training data. The most appropriate choice of training data is dictated partially by the states and observables one wants to simulate and partially by what is available. For example, to simulate only within a particular subspace, one may use only training data from that subspace or attempt to constrain the ansatz to exploring a subspace using ideas from geometric QML [53,54]. How to do so most efficiently remains an open question.

ACKNOWLEDGMENTS

We thank Sujay Kazi, Elliott Ball, and Robert M. Parrish for helpful conversations. Z.H. acknowledges support from the LANL Mark Kac Fellowship and from the Sandoz Family Foundation-Monique de Meuron program for Academic Promotion. M.C.C. was supported by the TopMath Graduate Center of the TUM Graduate School at the Technical University of Munich, Germany, the TopMath Program at the Elite Network of Bavaria, by a doctoral scholarship of the German Academic Scholarship Foundation (Studienstiftung des deutschen Volkes), by the BMWK (PlanQK), and by a DAAD PRIME Fellowship. NE was supported by the U.S. DOE, Department of Energy Computational Science Graduate Fellowship under Award No. DE-SC0020347. H.H. is supported by a Google PhD Fellowship. PJC and ATS acknowledge initial support from the Los Alamos National Laboratory (LANL) ASC Beyond Moore's Law project. ATS was also supported by the Laboratory Directed Research and Development (LDRD) program of Los Alamos National Laboratory under Project No. 20210116DR. L.C. acknowledges support from LDRD program of LANL under Project No. 20200022DR. L.C. and P.J.C. were also supported by the U.S. DOE, Office of Science, Office of Advanced Scientific Computing Research, under the Quantum Computing Application Teams (QCAT) program.

APPENDIX A: GENERALIZATION BOUNDS FOR VARIATIONAL QUANTUM MACHINE LEARNING AND UNITARY COMPILING

Here, we recall the theoretical results of Ref. [24] and discuss how they apply to our setting with quantum training data. Mathematically, a *quantum neural network (QNN)* (or *variational quantum machine learning model*) is a parametrized completely positive and trace-preserving (CPTP) map $\mathcal{E}^{\rm QNN}_{\boldsymbol{\alpha}}(\cdot)$, where the parameter vector $\boldsymbol{\alpha}=(\boldsymbol{k},\boldsymbol{\theta})$ consists of continuous parameters $\boldsymbol{\theta}$ and potentially also of discrete parameters $\boldsymbol{k}$. Importantly, the parametrization is determined by the structure of the QNN, e.g., by the layout of the underlying quantum circuit and the chosen parametrization of the gates therein. The performance of such a QNN on a data point $(|\Psi\rangle,|\Phi\rangle)$ is evaluated by a cost/loss function

$$\ell(\boldsymbol{\alpha};|\Psi\rangle,|\Phi\rangle)=\mathrm{Tr}\big[O^{\rm loss}_{|\Psi\rangle,|\Phi\rangle}\mathcal{E}^{\rm QNN}_{\boldsymbol{\alpha}}(|\Psi\rangle\langle\Psi|)\big], \tag{A1}$$

with a Hermitian loss observable $O^{\rm loss}_{|\Psi\rangle,|\Phi\rangle}$, which may depend on the data point.

When using a QNN for machine learning purposes, given a data set $\mathcal{D}=\mathcal{D}(N)=\{(|\Psi^{(j)}\rangle,|\Phi^{(j)}\rangle)\}_{j=1}^{N}$ consisting of N input-output pairs, we train the parameters $\boldsymbol{\alpha}$ in such a way as to achieve small average cost on the training data. The latter is given as

$$C_{\mathcal{D}(N)}(\boldsymbol{\alpha})=\frac{1}{N}\sum_{j=1}^{N}\ell(\boldsymbol{\alpha};|\Psi^{(j)}\rangle,|\Phi^{(j)}\rangle) \tag{A2}$$

and is also often called the *training error* of the QNN with parameter setting $\boldsymbol{\alpha}$. To analyze the performance of the QNN beyond the training data, we consider the *prediction error*, given by the expected cost

$$C_P(\boldsymbol{\alpha}) = \mathbb{E}_{(|\Psi\rangle,|\Phi\rangle)\sim P}[\ell(\boldsymbol{\alpha};|\Psi\rangle,|\Phi\rangle)], \tag{A3}$$

where P is the data-generating probability distribution. In using the QNN, our goal is to identify a parameter setting that achieves small prediction error. However, as the data-generating distribution is typically unknown, we cannot directly evaluate this quantity. Instead, a common strategy in machine learning is to attempt to achieve small training error instead. Justifying this strategy requires bounding the *generalization error* $\mathrm{gen}(\boldsymbol{\alpha}) = C_P(\boldsymbol{\alpha}) - C_{\mathcal{D}(N)}(\boldsymbol{\alpha})$.

In this paper, we make use of the following two generalization guarantees for QNNs. The first applies to QNNs that have only continuous parameters, but no discrete parameters allowing for training the structure of the QNN:

Theorem 2 ([24], Theorem 11). Let $\mathcal{E}^{\mathrm{QNN}}_{\boldsymbol{\theta}}(\cdot)$ be a QNN with a fixed architecture consisting of K parameterized local CPTP maps and an arbitrary number of nontrainable, global CPTP maps. Let P be a probability distribution over input-output pairs. Suppose that, given training data $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N$ of size N, with the $(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)$ drawn i.i.d. according to P, our optimization yields the parameter setting $\boldsymbol{\theta}^* = \boldsymbol{\theta}^*(\mathcal{D}(N))$.

Then, with probability at least $1-\delta$ over the choice of i.i.d. training data $\mathcal{D}(N)$ of size N according to P,

$$C_P(\boldsymbol{\theta}^*) - C_{\mathcal{D}(N)}(\boldsymbol{\theta}^*) \leqslant \mathcal{O}\left(C_{\mathrm{loss}}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{A4}$$

with $C_{\mathrm{loss}} = \sup_{|\Psi\rangle,|\Phi\rangle} \|O^{\mathrm{loss}}_{|\Psi\rangle,|\Phi\rangle}\|$.

To also allow for optimizing over the structure of the quantum circuit used in the QNN, we use the following:

Theorem 3 ([24], Corollary 3). Let $\mathcal{E}^{\mathrm{QNN}}_{\boldsymbol{\alpha}}(\cdot)$ be a QNN with a variable structure. Suppose that, for every $\kappa \in \mathbb{N}$, there are at most $G_\kappa \in \mathbb{N}$ allowed structures with exactly κ parameterized local CPTP maps and an arbitrary number of nontrainable, global CPTP maps. Let P be a probability distribution over input-output pairs. Suppose that, given training data $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N$ of size N, with the $(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)$ drawn i.i.d. according to P, our optimization yields a (data-dependent) structure with $K = K(S)$ parameterized 2-qubit CPTP maps and the parameter setting $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^*(\mathcal{D}(N))$.

Then, with probability at least $1-\delta$ over the choice of i.i.d. training data $\mathcal{D}(N)$ of size N according to P,

$$\begin{aligned} &C_P(\boldsymbol{\alpha}^*) - C_{\mathcal{D}(N)}(\boldsymbol{\alpha}^*) \\ &\leqslant \mathcal{O}\left(C_{\mathrm{loss}}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log G_K}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \end{aligned} \tag{A5}$$

with $C_{\mathrm{loss}} = \sup_{|\Psi\rangle,|\Phi\rangle} \|O^{\mathrm{loss}}_{|\Psi\rangle,|\Phi\rangle}\|$.

As observed in Ref. [24], Theorems 2 and 3 can be applied to obtain rigorous guarantees for variational unitary compilation. In this setting, training examples are of the form $(|\Psi\rangle, |\Phi\rangle)$, where $|\Phi\rangle = U|\Psi\rangle$ with U the (unknown) target unitary to be learned. The loss of a unitary QNN $\mathcal{V}^{\mathrm{QNN}}_{\boldsymbol{\alpha}}(\cdot) = V_{\boldsymbol{\alpha}}(\cdot)V_{\boldsymbol{\alpha}}^\dagger$ on the example $(|\Psi\rangle, |\Phi\rangle)$ is

$$\begin{aligned} \ell(\boldsymbol{\alpha};|\Psi\rangle,|\Phi\rangle) &= \tfrac{1}{4}\big\| |\Phi\rangle\langle\Phi| - \mathcal{V}^{\mathrm{QNN}}_{\boldsymbol{\alpha}}(|\Psi\rangle\langle\Psi|)\big\|_1^2 && \text{(A6)}\\ &= 1 - \mathrm{Tr}\big[|\Phi\rangle\langle\Phi| \cdot \mathcal{V}^{\mathrm{QNN}}_{\boldsymbol{\alpha}}(|\Psi\rangle\langle\Psi|)\big] && \text{(A7)}\\ &= 1 - \mathrm{Tr}[V_{\boldsymbol{\alpha}}^\dagger|\Phi\rangle\langle\Phi|V_{\boldsymbol{\alpha}} \cdot |\Psi\rangle\langle\Psi|]. && \text{(A8)} \end{aligned}$$

For convenience, we change the perspective to the Heisenberg picture. That is, given a training data point $(|\Psi\rangle, |\Phi\rangle)$, we think of $|\Phi\rangle$ as the input state to the unknown unitary $U^\dagger$ and of $|\Psi\rangle$ as the output state. Then, we see that the loss above is of the form of Eq. (A1), with loss observable

$$O^{\mathrm{loss}}_{|\Psi\rangle,|\Phi\rangle} = \mathbb{1} - |\Psi\rangle\langle\Psi|, \tag{A9}$$

which gives

$$\ell(\boldsymbol{\alpha};|\Psi\rangle,|\Phi\rangle) = \mathrm{Tr}\big[V_{\boldsymbol{\alpha}}^\dagger|\Phi\rangle\langle\Phi|V_{\boldsymbol{\alpha}} \cdot O^{\mathrm{loss}}_{|\Psi\rangle,|\Phi\rangle}\big]. \tag{A10}$$

Correspondingly, given a training data set $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N$ of size N, the training loss takes the form

$$\begin{aligned} C_{\mathcal{D}(N)}(\boldsymbol{\alpha}) &= \frac{1}{N}\sum_{j=1}^N \mathrm{Tr}\big[V_{\boldsymbol{\alpha}}^\dagger|\Phi^{(j)}\rangle\langle\Phi^{(j)}|V_{\boldsymbol{\alpha}} \cdot O^{\mathrm{loss}}_{|\Psi^{(j)}\rangle,|\Phi^{(j)}\rangle}\big] && \text{(A11)}\\ &= \frac{1}{N}\sum_{j=1}^N \big(1 - \big|\langle\Phi^{(j)}|V_{\boldsymbol{\alpha}}|\Psi^{(j)}\rangle\big|^2\big). && \text{(A12)} \end{aligned}$$

The expected cost is given by

$$\begin{aligned} C_P(\boldsymbol{\alpha}) &= \mathbb{E}_{|\Psi\rangle\sim P}[\mathrm{Tr}[V_{\boldsymbol{\alpha}}^\dagger|\Phi\rangle\langle\Phi|V_{\boldsymbol{\alpha}} \cdot O^{\mathrm{loss}}_{|\Psi\rangle,|\Phi\rangle}]] && \text{(A13)}\\ &= 1 - \mathbb{E}_{|\Psi\rangle\sim P}[|\langle\Phi|V_{\boldsymbol{\alpha}}|\Psi\rangle|^2], && \text{(A14)} \end{aligned}$$

where $|\Phi\rangle = U|\Psi\rangle$ and P is the data-generating measure. In this setting, the guarantees of Theorems 2 and 3 become:

Corollary 1. Let $\mathcal{V}^{\mathrm{QNN}}_{\boldsymbol{\theta}}(\cdot)$ be a unitary QNN with a fixed architecture consisting of K parameterized local unitaries and an arbitrary number of nontrainable, global unitaries. Let P be a probability distribution over input-output pairs. Suppose that, given training data $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N$ of size N, with the $|\Psi^{(j)}\rangle$ drawn i.i.d. according to P and $|\Phi^{(j)}\rangle = U|\Psi^{(j)}\rangle$, our optimization yields the parameter setting $\boldsymbol{\theta}^* = \boldsymbol{\theta}^*(\mathcal{D}(N))$.

Then, with probability at least $1-\delta$ over the choice of i.i.d. training data $\mathcal{D}(N)$ of size N according to P and U,

$$C_P(\boldsymbol{\theta}^*) - C_{\mathcal{D}(N)}(\boldsymbol{\theta}^*) \leqslant \mathcal{O}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{A15}$$

Again, this result has a variant for variable structure QNNs:

Corollary 2. Let $\mathcal{V}^{\mathrm{QNN}}_{\boldsymbol{\alpha}}(\cdot)$ be a unitary QNN with a variable structure. Suppose that, for every $\kappa \in \mathbb{N}$, there are at most $G_\kappa \in \mathbb{N}$ allowed structures with exactly κ parameterized local unitaries and an arbitrary number of nontrainable, global unitaries. Let P be a probability distribution over input-output pairs. Suppose that, given training data $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N$ of size N, with the $|\Psi^{(j)}\rangle$ drawn i.i.d. according to P and $|\Phi^{(j)}\rangle = U|\Psi^{(j)}\rangle$, our optimization yields a (data-dependent) structure with $K = K(S)$ parameterized local unitaries maps and the parameter setting $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^*(\mathcal{D}(N))$.

Then, with probability at least $1 - \delta$ over the choice of i.i.d. training data $\mathcal{D}(N)$ of size N according to P and U,

$$\begin{aligned} &C_P(\boldsymbol{\alpha}^*) - C_{\mathcal{D}(N)}(\boldsymbol{\alpha}^*) \\ &\quad\leqslant \mathcal{O}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log G_K}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right). \end{aligned} \tag{A16}$$

Note that we can define the loss function, and thus the training and expected cost, also using different loss observables $O^{\text{loss}}_{|\Psi^{(j)}\rangle,|\Phi^{(j)}\rangle}$ in Eqs. (A10), (A11), and (A13). As Theorems 2 and 3 allow for such general loss observables, a change of this kind does not alter the statements of Corollaries 1 and 2. We will make use of this observation in Appendix B 2 to define a suitable local cost.

APPENDIX B: PROOFS OF RESULTS PRESENTED IN THE MAIN TEXT

In this Appendix, we give complete proofs for all analytical results presented in the main text. We start in Appendix B 1 by providing the error analysis for training using fully Haar random n-qubit training states. These training states will typically be highly entangled and so hard to prepare on near term hardware. While we do not advocate this training strategy in practice, the error analysis for this case is simpler and thus acts as preparation for our subsequent analysis. In Appendix B 2, we provide the error analysis for the approach proposed in the main text whereby the training data is composed of tensor products of locally random states.

1. Training with Haar-random *n*-qubit input states

Haar-random n-qubit states provide a natural ensemble for generating data on which to train or test our ansatz $V_{\Delta t}(\boldsymbol{\alpha})$ for the short-time evolution. When working in the setting of unitary compilation described in Appendix A, choosing the n-qubit Haar measure as data-generating distribution in Eq. (A11) leads to training data of the form

$$\mathcal{D}_{\mathrm{E}}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N = \{(|\Psi^{(j)}\rangle, U|\Psi^{(j)}\rangle)\}_{j=1}^N, \tag{B1}$$

where the $|\Psi^{(j)}\rangle$ are i.i.d. Haar-random n-qubit states, typically highly entangled, and the $|\Phi^{(j)}\rangle = U|\Psi^{(j)}\rangle$ are the corresponding output states under the (approximate short-time) evolution $U = U_{\Delta t}$ that is to be learned. Plugging such a data set $\mathcal{D}_{\mathrm{E}}(N)$ into Eq. (A11), we obtain the corresponding training cost

$$C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{j=1}^N \mathrm{Tr}\left[V_{\boldsymbol{\alpha}}^\dagger |\Phi^{(j)}\rangle\langle\Phi^{(j)}| V_{\boldsymbol{\alpha}} \cdot O^{\text{loss}}_{|\Psi^{(j)}\rangle,|\Phi^{(j)}\rangle}\right] \tag{B2}$$

$$= 1 - \frac{1}{N}\sum_{j=1}^N |\langle\Psi^{(j)}|U^\dagger V_{\Delta t}(\boldsymbol{\alpha})|\Psi^{(j)}\rangle|^2. \tag{B3}$$

According to Eq. (A13), the corresponding expected cost is

$$C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}) = \mathbb{E}_{|\Psi\rangle\sim\mathrm{Haar}_n}\left[\mathrm{Tr}\left[V_{\Delta t}(\boldsymbol{\alpha})^\dagger|\Phi\rangle\langle\Phi|V_{\Delta t}(\boldsymbol{\alpha}) \cdot O^G_{|\Psi\rangle,|\Phi\rangle}\right]\right] \tag{B4}$$

$$= 1 - \mathbb{E}_{|\Psi\rangle\sim\mathrm{Haar}_n}[\mathrm{Tr}[U|\Psi\rangle\langle\Psi|U^\dagger V_{\Delta t}(\boldsymbol{\alpha})|\Psi\rangle\langle\Psi|V_{\Delta t}(\boldsymbol{\alpha})^\dagger]], \tag{B5}$$

where the expectation is with respect to states drawn from the n-qubit Haar measure, and $|\Phi\rangle = U|\Psi\rangle$, for U the approximate short-time evolution unitary to be learned.

There is another natural candidate for measuring the distance between two unitaries, and thus for evaluating the cost in our compilation task. Namely, we could also consider the *Hilbert-Schmidt test (HST) cost*, arising from the Hilbert-Schmidt inner product, given by

$$C_{HST}(U, V_{\Delta t}(\boldsymbol{\alpha})) = 1 - \frac{1}{d^2}|\mathrm{Tr}[U^\dagger V_{\Delta t}(\boldsymbol{\alpha})]|^2, \tag{B6}$$

where we write $d = 2^n$. This cost was proposed for the variational compilation of unitaries in Ref. [47]. In fact, the HST cost and the cost based on an expected fidelity over the n-qubit Haar measure are closely related:

Lemma 1. When taking a Haar-random n-qubit state as test state, the true cost for the time step Δt takes the form

$$C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}) = 1 - \overline{F}(U, V_{\Delta t}(\boldsymbol{\alpha})) = \frac{d}{d+1}C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha})). \tag{B7}$$

Here, we denote by

$$\overline{F}(U, V_{\Delta t}(\boldsymbol{\alpha})) := \mathbb{E}_{|\Psi\rangle\sim\mathrm{Haar}_n}[|\langle\Psi|U^\dagger V_{\Delta t}(\boldsymbol{\alpha})|\Psi\rangle|^2] \tag{B8}$$

the average fidelity over Haar-random inputs.

Proof. By definition of the cost function $C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha})$ and the average fidelity $\overline{F}(U, V_{\Delta t}(\boldsymbol{\alpha}))$, we directly have $C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}) = 1 - \overline{F}(U, V_{\Delta t}(\boldsymbol{\alpha}))$. The relation

$$C_{\mathrm{HST}}(U, V) = 1 - \frac{1}{d^2}|\mathrm{Tr}[U^\dagger V]|^2 = \frac{d+1}{d}(1 - \overline{F}(U, V)) \tag{B9}$$

can, e.g., be found in Refs. [47,56,57]. ■

Lemma 1 effectively allows us to freely choose in our analysis whether we work with $C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha})$, $\overline{F}(U, V_{\Delta t}(\boldsymbol{\alpha}))$, or $C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}))$. With this freedom, we can now establish a guarantee on the overall simulation fidelity for an evolution along a Hamiltonian H for time T, given by

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, T) = \overline{F}(e^{-iTH}, V_T(\boldsymbol{\alpha}_{\mathrm{opt}})), \tag{B10}$$

when using a QNN with K trainable local gates to learn the approximate short-time evolution. We denote by M the number of time steps of size Δt needed to achieve the overall time T, i.e., $T = M \cdot \Delta t$. Moreover, as in the main text, we denote by

$$\epsilon = \|U_{\Delta t} - e^{-iH\Delta t}\|_2, \tag{B11}$$

the short-time approximation error (measured according to the Schatten 2-norm) incurred by approximating $e^{-iH\Delta t}$ with $U_{\Delta t}$. With this notation in place, we state and prove our first guarantee:

Theorem 4 [Simulation error (entangled training, global cost)]. For a QNN $V_{\Delta t}(\boldsymbol{\alpha})$ with K parameterized local gates, with probability $\geqslant 1 - \delta$ over the choice of training data $\mathcal{D}_{\mathrm{E}}(N) = \{(|\Psi_i\rangle, U|\Psi_i\rangle)\}_{i=1}^N$, with the $|\Psi_i\rangle$ independent Haar-random n-qubit states and $U = U_{\Delta t}$ the unknown approximate short-time evolution unitary, the total simulation fidelity satisfies

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - 2M^2\left(\frac{\epsilon^2}{d+1} + 2C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})\right) - \mathcal{O}\left(M^2\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B12}$$

where $\boldsymbol{\alpha}_{\mathrm{opt}}$ denotes the final parameter setting after training.

Proof. By Theorem 2, with probability $\geqslant 1 - \delta$ over the choice of training data $\mathcal{D}_{\mathrm{E}}(N) = \{(|\Psi_i\rangle, U|\Psi_i\rangle)\}_{i=1}^N$, with the $|\Psi_i\rangle$ independent Haar-random n-qubit states, the final parameter setting $\boldsymbol{\alpha}_{\mathrm{opt}}$ satisfies

$$C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}_{\mathrm{opt}}) \leqslant C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{B13}$$

Once we recall from Lemma 1 that $C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}_{\mathrm{opt}}) = \frac{d}{d+1}C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}}))$, we see that the above gives a bound on the HST cost of the learned short-time evolution in terms of the training cost on N Haar-random n-qubit states, which holds with high probability. Namely, with probability $\geqslant 1 - \delta$,

$$C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}})) \leqslant \frac{d+1}{d}C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(\frac{d+1}{d}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right). \tag{B14}$$

Next, we make use of the behavior of the HST cost under iterated applications of unitaries, which it inherits from the Schatten 2-norm. Namely, according to Eq. (S20) of Ref. [6], we have

$$\sqrt{1 - \sqrt{1 - \frac{d+1}{d}(1 - \overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t))}} \leqslant \frac{M}{\sqrt{2d}}\epsilon + M\sqrt{1 - \sqrt{1 - C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}})}}. \tag{B15}$$

Rearranging this inequality, and applying both $\sqrt{1-x} \geqslant 1 - x$ for $x \in [0, 1]$ and $(a+b)^2 \leqslant 2(a^2 + b^2)$ for $a, b \in \mathbb{R}$, we can conclude

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - \frac{2M^2}{d+1}[\epsilon^2 + 2d\cdot C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}}))], \tag{B16}$$

where ϵ denotes the short-time approximation error as defined above. Now, we can plug in the bound from Eq. (B14) and obtain

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - \frac{2M^2}{d+1}\left[\epsilon^2 + 2d\cdot\left(\frac{d+1}{d}C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(\frac{d+1}{d}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right)\right)\right] \tag{B17}$$

$$= 1 - 2M^2\left(\frac{\epsilon^2}{d+1} + 2C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})\right) - \mathcal{O}\left(M^2\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B18}$$

which is the claimed bound. ∎

Theorem 4 tells us: When using a QNN to learn the approximate short-time evolution by training on a data set with Haar-random input states, a training data size effectively scaling as $N \sim M^4 \cdot K\log(K)$ will, with high probability, lead to good generalization also on the level of the long-time evolution.

Remark 1. From a more practical perspective, we can interpret Theorem 4 as justifying a purely training-cost-based termination condition for training if a large enough training data set is used. We can see this as follows: Suppose we want to achieve a simulation fidelity $\geqslant 1 - \varepsilon$ for up to M_0 forwarding steps, with some fixed success probability, say $\geqslant 0.95$. Then, it suffices to choose training data of size $N \sim M_0^4 \cdot \frac{K\log(K)}{(\varepsilon/2)^2}$ and to ensure that the training cost and the short-time approximation error satisfy $\frac{\epsilon^2}{d+1} + 2C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) \lesssim \frac{\varepsilon}{4M_0^2}$. In other words, assuming a suitable training data size is chosen, we can terminate training as soon as the training cost satisfies $C^G_{\mathcal{D}_{\mathrm{E}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) \lesssim \frac{\varepsilon}{8M_0^2} - \frac{\epsilon^2}{2(d+1)}$. (Naturally, this presupposes a small enough short-time approximation error, namely $\frac{\epsilon^2}{d+1} < \frac{\varepsilon}{4M_0^2}$.)

Remark 2. Throughout our proofs in this subsection, the only property of the n-qubit Haar measure Haar_n that entered our reasoning was the connection of the corresponding expected cost to the average fidelity and the HST cost, see

Lemma 1. We can replace the data-generating measure Haar_n by an n-qubit 2-design. This will lead to the same expected cost, still satisfying Lemma 1, and thus to the same simulation error guarantee.

2. Training with product of Haar-random single-qubit input states

The costs presented in Appendix B 1 are based on Haar-random n-qubit states. However, these are costly to prepare in practice as they require deep quantum circuits. In particular, assuming access to a training data set consisting of multiple Haar-random n-qubit states and their output states under (approximate) short-time evolution might be too optimistic for many practical applications as n grows. Therefore, we instead use a different notion of cost, based on tensor products of easy-to-prepare Haar-random single-qubit states. That is, we work with training data of the form

$$\begin{aligned}\mathcal{D}_{\mathrm{P}}(N) &= \left\{\left(\left|\Psi_{\mathrm{P}}^{(j)}\right\rangle,\ \left|\Phi_{\mathrm{P}}^{(j)}\right\rangle\right)\right\}_{j=1}^{N}\\ &= \left\{\left(\bigotimes_{i=1}^{n}\left|\psi_i^{(j)}\right\rangle, U\left(\bigotimes_{i=1}^{n}\left|\psi_i^{(j)}\right\rangle\right)\right)\right\}_{j=1}^{N},\end{aligned} \tag{B19}$$

where the $|\psi_i^{(j)}\rangle$ are drawn i.i.d. according to the single qubit Haar measure Haar_1 and $U = U_{\Delta t}$ is the unknown short-time evolution that we are trying to learn. Note that, if we have access to a black box implementing the evolution according to U, we can create such a training data set efficiently because only local Haar random states are required as inputs.

a. Global cost function

If we plug this training data for product state inputs into Eq. (A11), we see that the training cost is given by

$$C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{j=1}^{N}\mathrm{Tr}\left[V_{\boldsymbol{\alpha}}^{\dagger}\left|\Phi_{\mathrm{P}}^{(j)}\right\rangle\!\left\langle\Phi_{\mathrm{P}}^{(j)}\right|V_{\boldsymbol{\alpha}}\cdot O^{G}_{|\Psi_{\mathrm{P}}\rangle,|\Phi_{\mathrm{P}}\rangle}\right] \tag{B20}$$

$$= 1-\frac{1}{N}\sum_{j=1}^{N}\mathrm{Tr}\left[U\left|\Psi_{\mathrm{P}}^{(j)}\right\rangle\!\left\langle\Psi_{\mathrm{P}}^{(j)}\right|U^{\dagger}V_{\Delta t}(\boldsymbol{\alpha})\left|\Psi_{\mathrm{P}}^{(j)}\right\rangle\!\left\langle\Psi_{\mathrm{P}}^{(j)}\right|V_{\Delta t}(\boldsymbol{\alpha})^{\dagger}\right]. \tag{B21}$$

This cost is a global cost because the loss observables $O^{\mathrm{loss}}_{|\Psi_{\mathrm{P}}^{(j)}\rangle,|\Phi_{\mathrm{P}}^{(j)}\rangle} = O^{G}_{|\Psi_{\mathrm{P}}^{(j)}\rangle,|\Phi_{\mathrm{P}}^{(j)}\rangle} = \mathbb{1} - |\Psi_{\mathrm{P}}^{(j)}\rangle\langle\Psi_{\mathrm{P}}^{(j)}|$ are global measurements. That is, they require performing measurements on all n qubits. We will begin by providing rigorous results for the global cost. Later, we will discuss generalization to the local version that enjoys a better trainability.

Following Eq. (A13), for the global cost using product training states, Eq. (B20), the expected cost is

$$C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) = \mathbb{E}_{|\Psi_{\mathrm{P}}\rangle\sim\mathrm{Haar}_1^{\otimes n}}\left[\mathrm{Tr}\left[V_{\Delta t}(\boldsymbol{\alpha})^{\dagger}|\Phi_{\mathrm{P}}\rangle\langle\Phi_{\mathrm{P}}|V_{\Delta t}(\boldsymbol{\alpha})\cdot O^{G}_{|\Psi_{\mathrm{P}}\rangle,|\Phi_{\mathrm{P}}\rangle}\right]\right] \tag{B22}$$

$$= 1-\mathbb{E}_{|\Psi_{\mathrm{P}}\rangle\sim\mathrm{Haar}_1^{\otimes n}}[\mathrm{Tr}[U|\Psi_{\mathrm{P}}\rangle\langle\Psi_{\mathrm{P}}|U^{\dagger}V_{\Delta t}(\boldsymbol{\alpha})|\Psi_{\mathrm{P}}\rangle\langle\Psi_{\mathrm{P}}|V_{\Delta t}(\boldsymbol{\alpha})^{\dagger}]]. \tag{B23}$$

Here, we write $\Psi_{\mathrm{P}} = \bigotimes_{i=1}^{n}|\psi_i\rangle \sim \mathrm{Haar}_1^{\otimes n}$ to express that the expectation is over tensor products of states, each drawn from Haar_1, and $|\Phi_{\mathrm{P}}\rangle = U|\Psi_{\mathrm{P}}\rangle$.

As a special case of [26, Lemma 1], we obtain that the product cost $C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha})$ is closely related to the entangled cost $C^G_{\mathrm{Haar}}(\boldsymbol{\alpha})$:

Lemma 2 [Product versus entangled cost (global)]. For any parameter setting $\boldsymbol{\alpha}$, we have the following relation between the global versions of the product and the entangled cost on n qubits,

$$C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \leqslant \frac{d+1}{d}C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}) \leqslant 2C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}). \tag{B24}$$

With this connection between $C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha})$ and $C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha})$ in place, we can now prove the first part of Theorem 1, which we state formally here:

Theorem 5 (Simulation error (product training, global cost)—First half of Theorem 1). For a QNN $V_{\Delta t}(\boldsymbol{\alpha})$ with K parameterized local gates, with probability $\geqslant 1-\delta$ over the choice of training data $\mathcal{D}_{\mathrm{P}}(N) = \{\bigotimes_{j=1}^{n}|\psi_i^{(j)}\rangle, U\bigotimes_{j=1}^{n}|\psi_i^{(j)}\rangle\}_{i=1}^{N}$, with the $|\psi_i^{(j)}\rangle$ independent Haar-random single-qubit states, the total simulation fidelity satisfies

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1-2M^2\left[\frac{\epsilon^2}{d+1}+4C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})\right]-\mathcal{O}\left(M^2\left(\sqrt{\frac{K\log(K)}{N}}+\sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B25}$$

where $\boldsymbol{\alpha}_{\mathrm{opt}} = \boldsymbol{\alpha}_{\mathrm{opt}}(\mathcal{D}_{\mathrm{P}}(N))$ denotes the final parameter setting after training.

Proof. By Theorem 2 and Corollary 1, we know: With probability $\geqslant 1-\delta$ over the choice of training data $\mathcal{D}_{\mathrm{P}}(N) = \{\bigotimes_{j=1}^{n}|\psi_i^{(j)}\rangle, U\bigotimes_{j=1}^{n}|\psi_i^{(j)}\rangle\}_{i=1}^{N}$, with the $|\psi_i^{(j)}\rangle$ independent Haar-random single-qubit states, the final parameter setting $\boldsymbol{\alpha}_{\mathrm{opt}} = \boldsymbol{\alpha}_{\mathrm{opt}}(\mathcal{D}_{\mathrm{P}}(N))$ satisfies

$$C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \leqslant C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})+\mathcal{O}\left(\sqrt{\frac{K\log(K)}{N}}+\sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{B26}$$

By Lemmas 1 and Lemma 2, this implies that, with probability $\geqslant 1-\delta$,

$$C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}})) = \frac{d+1}{d} C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}) \leqslant 2C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \leqslant 2C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{B27}$$

We now recall from Eq. (B16) that

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - \frac{2M^2}{d+1}[\epsilon^2 + 2d\cdot C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}}))], \tag{B28}$$

with ϵ the short-time approximation error

$$\epsilon = \|U_{\Delta t} - e^{-iH\Delta t}\|_2. \tag{B29}$$

If we plug our above upper bound on the HST cost $C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}}))$ into this lower bound for the overall simulation fidelity $\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t)$, we obtain: With probability $\geqslant 1-\delta$,

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - \frac{2M^2}{d+1}\left[\epsilon^2 + 2d\cdot\left(2C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right)\right] \tag{B30}$$

$$= 1 - 2M^2\left[\frac{\epsilon^2}{d+1} + \frac{4d}{d+1}\cdot C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})\right] - \mathcal{O}\left(M^2\cdot\frac{d}{d+1}\cdot\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right) \tag{B31}$$

$$\geqslant 1 - 2M^2\left[\frac{\epsilon^2}{d+1} + 4C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})\right] - \mathcal{O}\left(M^2\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B32}$$

as claimed. ■

In the case of a variable structure QNN for learning the approximate short-time evolution, we have the following guarantee:

Corollary 3. For a QNN $V_{\Delta t}(\boldsymbol{\alpha})$ with a variable structure such that, for every $\kappa \in \mathbb{N}$, there are at most $G_\kappa \in \mathbb{N}$ allowed structures with exactly κ parameterized local gates, with probability $\geqslant 1-\delta$ over the choice of training data $\mathcal{D}_{\mathrm{P}}(N) = \{\bigotimes_{j=1}^n |\psi_i^{(j)}\rangle, U\bigotimes_{j=1}^n |\psi_i^{(j)}\rangle\}_{i=1}^N$, with the $|\psi_i^{(j)}\rangle$ independent Haar-random single-qubit states, the total simulation fidelity satisfies

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - 2M^2\left[\frac{\epsilon^2}{d+1} + 4C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})\right] - \mathcal{O}\left(M^2\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log G_K}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B33}$$

where $K = K(\mathcal{D}_{\mathrm{P}}(N))$ denotes the number of parameterized local gates used in the final structure after training and $\boldsymbol{\alpha}_{\mathrm{opt}} = \boldsymbol{\alpha}_{\mathrm{opt}}(\mathcal{D}_{\mathrm{P}}(N))$ denotes the final parameter setting after training.

Proof. The proof of this Corollary is analogous to the proof of Theorem 5, we only have to use the generalization guarantees of Theorem 3 and Corollary 2 instead of Theorem 2 and Corollary 1. ■

Analogously to Remark 1, one can use Theorem 5 and Corollary 3 to define a termination condition based on the achieved training cost, if a sufficiently large training data set is used.

b. Local cost function

While tensor products of Haar-random single-qubit states are relatively cheap to prepare, a training cost of the form used in Theorem 5 and Corollary 3 also can be challenging to deal with. Namely, it is based on performing measurements on all n qubits at the output of a parametrized quantum circuits, which leads to poor trainability due to barren plateaus even for QNNs of moderate depth [35]. This occurs because the loss observables $O^{\mathrm{loss}}_{|\Psi_{\mathrm{P}}^{(j)}\rangle,|\Phi_{\mathrm{P}}^{(j)}\rangle} = O^{G}_{|\Psi_{\mathrm{P}}^{(j)}\rangle,|\Phi_{\mathrm{P}}^{(j)}\rangle} = \mathbb{1} - |\Psi_{\mathrm{P}}^{(j)}\rangle\langle\Psi_{\mathrm{P}}^{(j)}|$ are global. In contrast, using a cost function that depends only on measurements of single qubits at the circuit output can lead to improved trainability for shallow QNNs [35].

Therefore, we consider a local version of our cost based on Haar-random single-qubit states. We still employ the product state training data

$$\mathcal{D}_{\mathrm{P}}(N) = \{(|\Psi_{\mathrm{P}}^{(j)}\rangle, |\Phi_{\mathrm{P}}^{(j)}\rangle)\}_{j=1}^N = \left\{\left(\bigotimes_{i=1}^n |\psi_i^{(j)}\rangle, U\left(\bigotimes_{i=1}^n |\psi_i^{(j)}\rangle\right)\right)\right\}_{j=1}^N, \tag{B34}$$

with the $|\psi_i^{(j)}\rangle$ drawn i.i.d. according to Haar_1. Now, we define the local loss observables

$$O^{\mathrm{loss}}_{|\Psi_{\mathrm{P}}^{(j)}\rangle,|\Phi_{\mathrm{P}}^{(j)}\rangle} = O^{L}_{|\Psi_{\mathrm{P}}^{(j)}\rangle,|\Phi_{\mathrm{P}}^{(j)}\rangle} = \mathbb{1} - \frac{1}{n}\sum_{i=1}^n |\psi_i^{(j)}\rangle\langle\psi_i^{(j)}| \otimes \mathbb{1}_{\bar{i}}, \tag{B35}$$

which again have operator norm bounded by 1. Here $\mathbb{1}_{\bar{i}}$ denotes the identity acting on all but the ith qubit. With this choice of training data and loss observables, Eq. (A11) gives the training cost

$$C^L_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{j=1}^{N}\mathrm{Tr}\big[V_{\boldsymbol{\alpha}}^{\dagger}\big|\Phi_{\mathrm{P}}^{(j)}\big\rangle\big\langle\Phi_{\mathrm{P}}^{(j)}\big|V_{\boldsymbol{\alpha}}\cdot O^L_{|\Psi_{\mathrm{P}}\rangle,|\Phi_{\mathrm{P}}\rangle}\big] \tag{B36}$$

$$= 1 - \frac{1}{nN}\sum_{j=1}^{N}\sum_{i=1}^{n}\mathrm{Tr}\big[U\big|\Psi_{\mathrm{P}}^{(j)}\big\rangle\big\langle\Psi_{\mathrm{P}}^{(j)}\big|U^{\dagger}V_{\Delta t}(\boldsymbol{\alpha})\big(\big|\psi_i^{(j)}\big\rangle\big\langle\psi_i^{(j)}\big|\otimes\mathbb{1}_{\bar{i}}\big)V_{\Delta t}(\boldsymbol{\alpha})^{\dagger}\big]. \tag{B37}$$

Comparing this expression to $C^G_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha})$, we see that $C^L_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha})$ still uses tensor products of random single-qubit states as inputs, but instead of measuring all qubits at the output, we average over simple single-qubit measurements. And the expected cost arising from Eq. (A13) is

$$C^L_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) = \mathbb{E}_{|\Psi_{\mathrm{P}}\rangle\sim\mathrm{Haar}_1^{\otimes n}}\big[\mathrm{Tr}\big[V_{\Delta t}(\boldsymbol{\alpha})^{\dagger}|\Phi_{\mathrm{P}}\rangle\langle\Phi_{\mathrm{P}}|V_{\Delta t}(\boldsymbol{\alpha})\cdot O^L_{|\Psi_{\mathrm{P}}\rangle,|\Phi_{\mathrm{P}}\rangle}\big]\big] \tag{B38}$$

$$= 1 - \mathbb{E}_{|\Psi_{\mathrm{P}}\rangle\sim\mathrm{Haar}_1^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathrm{Tr}\big[U|\Psi_{\mathrm{P}}\rangle\langle\Psi_{\mathrm{P}}|U^{\dagger}V_{\Delta t}(\boldsymbol{\alpha})\big(\big|\psi_i^{(j)}\big\rangle\big\langle\psi_i^{(j)}\big|\otimes\mathbb{1}_{\bar{i}}\big)V_{\Delta t}(\boldsymbol{\alpha})^{\dagger}\big]\right]. \tag{B39}$$

The first observation underlying our analysis of the overall simulation fidelity in terms of $C^L_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha})$ is that the global and local versions of the cost defined in terms of tensor products of Haar-random single-qubit states are tightly related:

Lemma 3 (Product cost: Global versus local). For any parameter setting $\boldsymbol{\alpha}$, we have the following relation between the global and local versions of the product cost on n qubits,

$$C^L_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \leqslant C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \leqslant n\cdot C^L_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}). \tag{B40}$$

Proof. The proof of this Lemma is a direct application of the reasoning from Appendix C of [47]. ■

This relation between the two costs allows us to extend our previous analysis to $C^L_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha})$:

Corollary 4 (Simulation error (product training, local cost) – Second half of Theorem 1). For a QNN $V_{\Delta t}(\boldsymbol{\alpha})$ with K parameterized local gates, with probability $\geqslant 1-\delta$ over the choice of training data $\mathcal{D}_{\mathrm{P}}(N) = \{\bigotimes_{j=1}^{n}|\psi_i^{(j)}\rangle, U\bigotimes_{j=1}^{n}|\psi_i^{(j)}\rangle\}_{j=1}^{N}$, with the $|\psi_i^{(j)}\rangle$ independent Haar-random single-qubit states, the total simulation fidelity satisfies

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - 2M^2\left[\frac{\epsilon^2}{d+1} + 4nC^L_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}})\right] - \mathcal{O}\left(M^2 n\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B41}$$

where $\boldsymbol{\alpha}_{\mathrm{opt}} = \boldsymbol{\alpha}_{\mathrm{opt}}(\mathcal{D}_{\mathrm{P}}(N))$ denotes the final parameter setting after training.

Proof. We start by stating the generalization guarantee obtained from Theorem 2 and Corollary 1 for our scenario: With probability $\geqslant 1-\delta$ over the choice of training data $\mathcal{D}_{\mathrm{P}}(N) = \{\bigotimes_{i=1}^{n}|\psi_i^{(j)}\rangle, U\bigotimes_{j=1}^{n}|\psi_i^{(j)}\rangle\}_{i=1}^{N}$, with the $|\psi_i^{(j)}\rangle$ independent Haar-random single-qubit states, the final parameter setting $\boldsymbol{\alpha}_{\mathrm{opt}} = \boldsymbol{\alpha}_{\mathrm{opt}}(\mathcal{D}_{\mathrm{P}}(N))$ satisfies

$$C^L_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \leqslant C^L_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{B42}$$

Combining Lemma 1, Lemma 2, and Lemma 3, this implies that, with probability $\geqslant 1-\delta$,

$$C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}})) = \frac{d+1}{d}C^G_{\mathrm{Haar}_n}(\boldsymbol{\alpha}) \tag{B43}$$

$$\leqslant 2C^G_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \tag{B44}$$

$$\leqslant 2nC^L_{\mathrm{Haar}_1^{\otimes n}}(\boldsymbol{\alpha}) \tag{B45}$$

$$\leqslant 2nC^L_{\mathcal{D}_{\mathrm{P}}(N)}(\boldsymbol{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(n\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right). \tag{B46}$$

We again recall that, by Eq. (B16),

$$\overline{F}(\boldsymbol{\alpha}_{\mathrm{opt}}, M\cdot\Delta t) \geqslant 1 - \frac{2M^2}{d+1}[\epsilon^2 + 2d\cdot C_{\mathrm{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\mathrm{opt}}))]. \tag{B47}$$

Algorithm: 1 Resource Efficient Fast Forwarding

```
Input: $n$-qubit Hamiltonian, $H$; Target simulation fidelity at $M_0$ fast-forwarding steps, $1-\varepsilon$; Diagonal ansatz,
    $V_t(\boldsymbol{\theta},\boldsymbol{\gamma}) = W(\boldsymbol{\theta})D(t\boldsymbol{\gamma})W(\boldsymbol{\theta})^\dagger$ with $K$ parameterized gates; set of $N \sim M_0^4\frac{K\log(K)}{(\varepsilon/2)^2}$ input-output training
    states, $\mathcal{D}(N) = \{(|\Psi^{(j)}\rangle, |\Phi^{(j)}\rangle)\}_{j=1}^N$ where the input states are random product states $\{\bigotimes_{i=1}^n |\psi_i^{(j)}\rangle\}_{j=1}^N$,
    with $|\psi_i^{(j)}\rangle$ drawn from the 1-qubit Haar distribution, and the output states are generated by evolving the
    input states by a short-time Trotterized evolution operator $|\Phi^{(j)}\rangle = U_{\Delta t}|\Psi^{(j)}\rangle$ where $U_{\Delta t} \approx e^{-iH\Delta t}$.
Output: Time-dependent QNN $V_t(\theta_{\text{opt}}, \gamma_{\text{opt}})$
Randomly initialize the parameters of $V_t(\boldsymbol{\theta},\boldsymbol{\gamma})$
while $C^G_{\mathcal{D}_{\text{P}}(N)}(\boldsymbol{\theta},\boldsymbol{\gamma}) > \frac{\varepsilon}{16M_0^2} - \frac{\epsilon^2}{4(2^n+1)}$ do
    Calculate gradient vector using Eqs. (C1) & (C3) via the circuit shown in Fig. 4.
    Update parameters of $V_{\Delta t}(\boldsymbol{\theta},\boldsymbol{\gamma})$ with classical optimizer
return $(\theta_{\text{opt}}, \gamma_{\text{opt}})$
```

Plugging our above upper bound on the HST cost $C_{\text{HST}}(U, V_{\Delta t}(\boldsymbol{\alpha}_{\text{opt}}))$ into this lower bound for the overall simulation fidelity $\overline{F}(\boldsymbol{\alpha}_{\text{opt}}, M\cdot\Delta t)$, we obtain: With probability $\geqslant 1-\delta$,

$$\overline{F}(\boldsymbol{\alpha}_{\text{opt}}, M\cdot\Delta t) \geqslant 1 - \frac{2M^2}{d+1}\left[\epsilon^2 + 2d\cdot\left(2nC^L_{\mathcal{D}_{\text{P}}(N)}(\boldsymbol{\alpha}_{\text{opt}}) + \mathcal{O}\left(n\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right)\right)\right] \tag{B48}$$

$$= 1 - 2M^2\left[\frac{\epsilon^2}{d+1} + \frac{4d}{d+1}\cdot nC^G_{\mathcal{D}_{\text{P}}(N)}(\boldsymbol{\alpha}_{\text{opt}})\right] - \mathcal{O}\left(M^2 n\cdot\frac{d}{d+1}\cdot\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right) \tag{B49}$$

$$\geqslant 1 - 2M^2\left[\frac{\epsilon^2}{d+1} + 4nC^G_{\mathcal{D}_{\text{P}}(N)}(\boldsymbol{\alpha}_{\text{opt}})\right] - \mathcal{O}\left(M^2 n\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B50}$$

which is the claimed bound. ■

Again, we can easily state the variable structure version of this Theorem.

Corollary 5. For a QNN $V_{\Delta t}(\boldsymbol{\alpha})$ with a variable structure such that, for every $\kappa \in \mathbb{N}$, there are at most $G_\kappa \in \mathbb{N}$ allowed structures with exactly κ parameterized local gates, with probability $\geqslant 1-\delta$ over the choice of training data $\mathcal{D}_{\text{P}}(N) = \{\bigotimes_{j=1}^n |\psi_i^{(j)}\rangle, U\bigotimes_{j=1}^n |\psi_i^{(j)}\rangle\}_{i=1}^N$, with the $|\psi_i^{(j)}\rangle$ independent Haar-random single-qubit states, the total simulation fidelity satisfies

$$\overline{F}(\boldsymbol{\alpha}_{\text{opt}}, M\cdot\Delta t) \geqslant 1 - 2M^2\left[\frac{\epsilon^2}{d+1} + 4nC^L_{\mathcal{D}_{\text{P}}(N)}(\boldsymbol{\alpha}_{\text{opt}})\right] - \mathcal{O}\left(M^2 n\left(\sqrt{\frac{K\log(K)}{N}} + \sqrt{\frac{\log G_K}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right), \tag{B51}$$

where $K = K(\mathcal{D}_{\text{P}}(N))$ denotes the number of parameterized local gates used in the final structure after training and $\boldsymbol{\alpha}_{\text{opt}} = \boldsymbol{\alpha}_{\text{opt}}(\mathcal{D}_{\text{P}}(N))$ denotes the final parameter setting after training.

As before, from Corollaries 4 and 5 we can derive termination conditions based on the training cost achieved in the optimization and the size of the training data sets.

Remark 3. As the results of Ref. [26] also apply when replacing the data-generating measure Haar_1^n by a tensor product of 1-qubit 2-designs, we can do the same replacement here and still obtain the same bounds. As a concrete example, the same guarantees hold when training on tensor products of uniformly random single-qubit stabilizer states.

APPENDIX C: FURTHER DETAILS ON NUMERICAL IMPLEMENTATIONS

1. Statement of Algorithm

The REFF algorithm is outlined with pseudocode in Algorithm 1. By inverting Theorem 1, our results provide an upper bound on the number of training states required to achieve a target simulation fidelity, $1-\varepsilon$, after M_0 fast-forwarding steps at $N \sim M_0^4\frac{K\log(K)}{(\varepsilon/2)^2}$. Our numerical results, however, have found that a full generalization can be achieved with far fewer states than this bound would suggest. To take advantage of this and seek a minimally sized training dataset, an algorithm variant is motivated, starting with a small training dataset that is grown over time until generalization is observed. As we are working below the upper bound, the guarantees on generalization no longer apply, so an extra validation error is required to quantify the QNN's generalization. This could be implemented by creating a validation dataset of Haar-random product states that the C_{REFF} cost is periodically tested against as the optimization progresses. If the training C_{REFF} is decreasing during the optimization, but the validation C_{REFF} is observed to plateau, this indicates the training dataset is too small and its size should be increased.

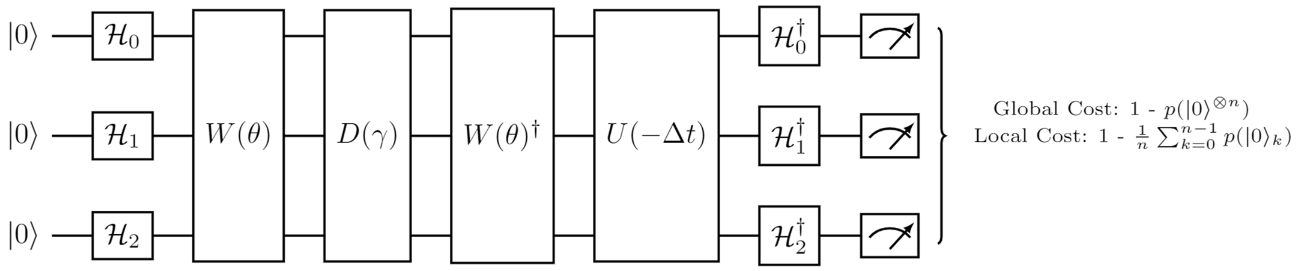

FIG. 4. REFF circuit. A generalized 3-qubit representation of the circuit used to compute C_{REFF} is displayed. $\mathcal{H}_k$ is a Haar-random single-qubit unitary. For a training dataset of size N, the output of this circuit is averaged over the N unique training states to compute the cost function.

2. Gradient Formula

Following the method of [6,8], the partial derivative of $C_{\mathrm{REFF}}(U, V(\boldsymbol{\theta}, \boldsymbol{\gamma}))$ with respect to θ_l, is

$$\begin{aligned}\frac{\partial C_{\mathrm{REFF}}(U,V)}{\partial \theta_l} &= \frac{1}{2}(C_{\mathrm{REFF}}(U, W_{l+}DW^\dagger) - C_{\mathrm{REFF}}(U, W_{l-}DW^\dagger) \\ &\quad + C_{\mathrm{REFF}}(U, WD(W_{l+})^\dagger) - C_{\mathrm{REFF}}(U, WD(W_{l-})^\dagger)). \end{aligned} \tag{C1}$$

The unitary W_{l+} (W_{l-}) is generated from the original unitary $W(\boldsymbol{\theta})$ by the addition of an extra $\frac{\pi}{2}$ ($-\frac{\pi}{2}$) rotation about a given parameter's rotation axis,

$$W_{l\pm} := W(\boldsymbol{\theta}_{l\pm}) \text{ with } (\theta_{l\pm})_i := \theta_l \pm \frac{\pi}{2}\delta_{i,l}. \tag{C2}$$

The analogous formula for the partial derivative with respect to γ_l, is

$$\begin{aligned}\frac{\partial C_{\mathrm{REFF}}(U,V)}{\partial \gamma_l} \\ = \frac{1}{2}(C_{\mathrm{REFF}}(U, WD_{l+}W^\dagger) - C_{\mathrm{REFF}}(U, WD_{l-}W^\dagger)). \end{aligned} \tag{C3}$$

3. Application to Alternate Topologies

Here we demonstrate the resource efficient fast-forwarding algorithm applied to systems with topologies and interactions causing frustrated magnetism. Figure 5(a) shows the learning of a diagonalization of a bond-frustrated 6-qubit section of the Kitaev Honeycomb model [58], the topology is shown in the inset. We use the basin-hopping global optimizer [59], causing the sharp decreases in loss function when a new local minimum is found. The training loss, C_{REFF}, was computed using two product training states, which was sufficient to achieve

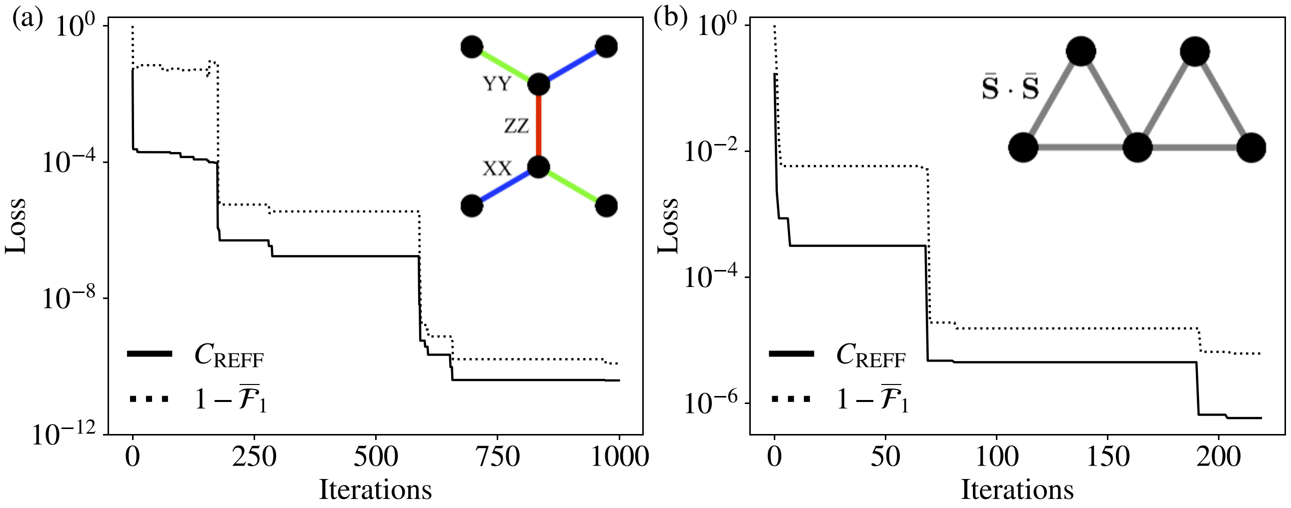

FIG. 5. Frustrated magnetism diagonalizations. Here we apply the REFF algorithm to two systems exhibiting frustrated magnetism, the respective topologies are shown in the insets. C_{REFF} corresponds to the training loss function using product state overlaps; $1-\overline{\mathcal{F}}_1$ corresponds to the testing loss signifying the global compilation over the Hilbert space. (a) Bond-frustrated 6-qubit section of the Kitaev Honeycomb model: two product states are required to result in a compilation infidelity, $1-\overline{\mathcal{F}}_1$, equal to 2.09×10^{-10} after convergence of the global optimizer. (b) Geometrically-frustrated 5-qubit "bowtie" with Heisenberg interactions: 10 product states are required to result in a compilation infidelity equal to 6.46×10^{-6} after convergence of the global optimizer.

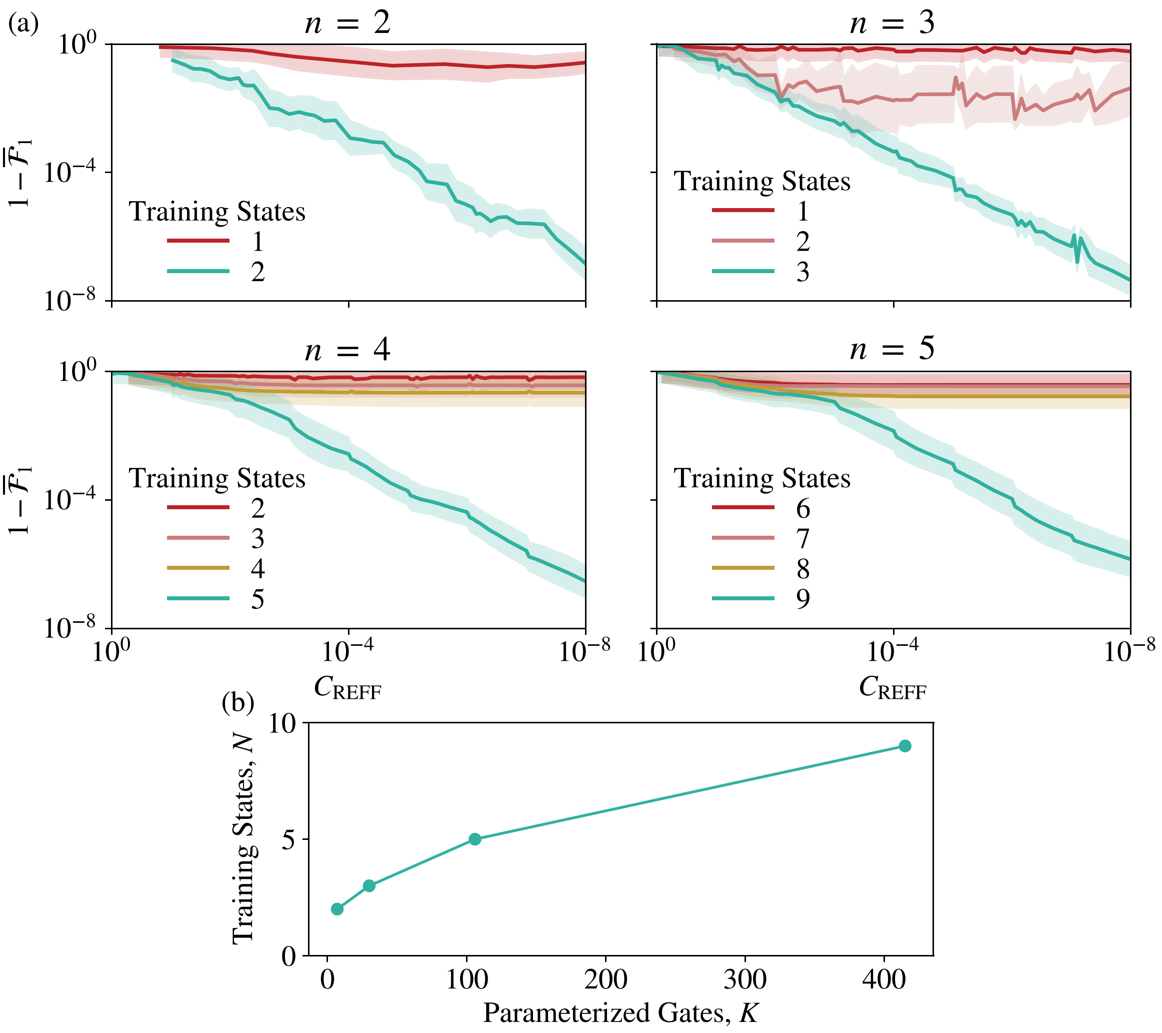

FIG. 6. Scaling. (a) REFF is applied to increasing sizes of the n-qubit Heisenberg Hamiltonian, $H = \sum_{i=1}^{n} \boldsymbol{S}_i \cdot \boldsymbol{S}_{i+1}$ with periodic boundary conditions. An increasing size dataset of Haar-random product states is used for training, to determine the minimum number required for generalization. This is seen when minimizing the training loss ($C^{\text{G}}_{\text{REFF}}$) results in simultaneously minimizing the testing loss ($1 - \overline{\mathcal{F}}_1$) without premature plateauing. (b) The minimum number of training states required for generalization in the above numerics are plotted against the number of parameterized gates in its VFF ansatz, showing an apparent sublinear scaling.

generalization, signified by the simultaneous minimization of the testing loss $1 - \overline{\mathcal{F}}_1$. The ansatz in total had 98 trainable parameters. Figure 5(b) shows the learning of a diagonalization of a geometrically-frustrated five qubit "bowtie" with Heisenberg interactions. Here, 10 product training states were used to achieve sufficient generalization. The ansatz had in total 399 trainable parameters.

4. Choice of Ansatz

XY Hamiltonian. For our diagonalizing unitary, we exploit the particle-number conservation property of the Heisenberg Hamiltonians, and for W use only 2-qubit Givens rotation gates that also respect this symmetry. This ensures the state remains in the symmetry subspace, simplifying the optimization. For the numerics shown in Fig. 2(b), a brickwork-style ansatz was used for W. Each layer was composed of a Givens rotation gate between odd pairs of qubits, then a Givens rotation gate between even pairs of qubits, resulting in $n - 1$ gates per layer. All the examples used $1.5n$ layers, resulting in a total gate count for W of $1.5n(n-1)$ and a depth of $3n$; this can be considered an improvement to the $\mathcal{O}(n^2)$ gate count and $\mathcal{O}(n\log(n))$ circuit depth presented in [49]. The ansatz for D was composed of a single R_z gate on each qubit.

Heisenberg Hamiltonian. It is known [60] that a 2-local nearest-neighbor gate fabric of Givens rotations is not universal for the Hamming-weight preserving subgroup $\mathcal{H}(2^n)$, so the ansatz we used for the XY Hamiltonian does not diagonalize the more general Heisenberg Hamiltonian. In these numerics, for the W layered ansatz, we instead use the most general 2-qubit gate that conserves the particle number, which takes four parameters and is displayed below:

$$\text{Sym}(\theta_1, \theta_2, \theta_3, \theta_4) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_1) & -e^{i\theta_2}\sin(\theta_1) & 0 \\ 0 & e^{i\theta_3}\sin(\theta_1) & e^{i(\theta_2+\theta_3)}\cos(\theta_1) & 0 \\ 0 & 0 & 0 & e^{i\theta_4} \end{pmatrix} \tag{C4}$$

D is composed of an R_z rotation gate on each qubit, and an R_{zz} gate between all pairs of qubits.

Kitaev honeycomb model. For the diagonalizing unitary, W, we use a hardware efficient ansatz, applying noncommuting layers of CNOTs, dressed with layers of Ry and Rz single qubit rotation gates. The diagonal unitary D was composed of single qubit Rz gates.

5. Data Requirements for Generalization

Theorem 1 upperbounds the number of training states required to guarantee a high-simulation fidelity for a QNN with K parameterized gates by the scaling $\mathcal{O}(K\log(K))$. Here we numerically investigate this bound by performing REFF on increasing n-qubit Heisenberg Hamiltonians and observe the minimum number of training states required for generalization. For each system size, we perform REFF with a range of training set sizes, training until $C^{\rm G}_{\rm REFF} = 10^{-8}$ and simultaneously compute the simulation fidelity $(1-\overline{\mathcal{F}}_1)$ as our metric for generalization. Each instance of REFF performed on a particular training set size was repeated five times to generate a mean and standard deviation. We observe, as shown in Fig. 6, that training sets of insufficient size do not produce generalization when trained on, but when the training set reaches some critical threshold, minimizing $C^{\rm G}_{\rm REFF}$ simultaneously minimizes the simulation fidelity. The scaling of this threshold is plotted in Fig. 6(b) as a function of the number of parameterized gates,K, the required number of training states N, appears to scale sub-linearly, within our analytic upperbound.

6. Simulation fidelity against the true Hamiltonian evolution

The numerical results in Fig. 2 demonstrate that with the correct ansatz design, the Hilbert-Schmidt test between the Trotterized unitary and the diagonalized ansatz can reduced arbitrarily small for the systems tested. As shown in the inset of Fig. 2(b), this allows the action of the Trotterized unitary to be fast forwarded for long-time simulations. The unitary we are learning is a Trotterization, with an associated Trotter error with respect to the true Hamiltonian evolution, which will also be learned during the REFF training. REFF can at best perfectly learn the evolution with this Trotter error included; however, increasing the order of the Trotter approximation decreases this error to reproduce the true Schrödinger evolution. Figure 7 shows the fast forwarding of a diagonalized ansatz after REFF training against the 10 qubit Hamiltonian $H=\sum_{i=1}^{9}X_iX_{i+1}+Y_iY_{i+1}$ with open boundary conditions. To compare the simulation fidelity against the Trotter unitary $U(\Delta t)$ for noninteger multiples of Δt, for $t=M\Delta t+c$ (with $c<\Delta t$) the unitary applied is $U_{\rm Trotter}(t)=U(\Delta t)^M U(\frac{c}{\Delta t})$. Here the Trotterization is a second-order Suzuki-Trotter approximation, with Trotter number 10, and hence has a very low Trotter error. The VFF ansatz had in total 360 CNOT gates, compared to 380 in the Trotter unitary.

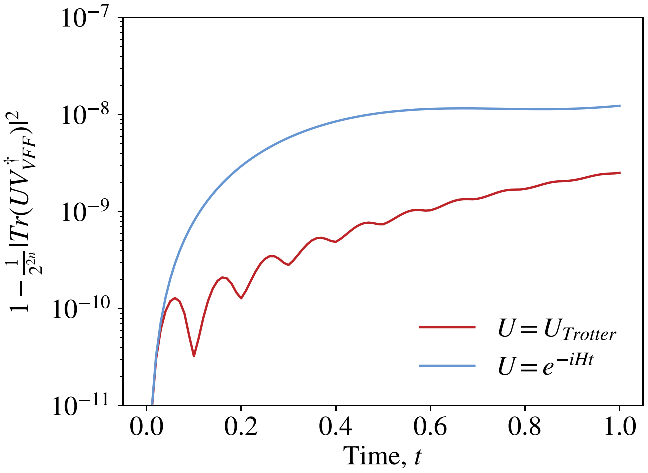

FIG. 7. Trotter error. The REFF algorithm is applied to the 10 qubit Hamiltonian $H=\sum_{i=1}^{9}X_iX_{i+1}+Y_iY_{i+1}$ with open boundary conditions, and the resulting diagonalized unitary performs a fast forwarding of the Hamiltonian. The Trotterization has a timestep of $\Delta t = 0.1$. The fidelity of the fast forwarding is computed with respect to the Trotterized unitary and the true Hamiltonian evolution.

7. Quantum state tomography of hardware implementation

Figure 3(b) shows the evaluation of the fast-forwarding performance of the QNN trained on ibmq_bogota. We use quantum state tomography to reconstruct the output density matrix of both the REFF-evolved state and the Trotter-evolved state at timestep N, $\rho(N)$, and compute the fidelity with respect to the noise-free Trotter evolved state $|\psi(N)\rangle$. The method for performing this follows [8]; an n-dimensional density matrix ρ can be decomposed into the Pauli product basis as $\rho=\boldsymbol{\eta}\cdot\boldsymbol{\sigma}^{(n)}$. Here $\boldsymbol{\sigma}^{(n)}$ is a 4^n dimensional vector composed of the elements of the n-qubit Pauli group $P_n=\{\sigma_I,\sigma_X,\sigma_Y,\sigma_Z\}^{\otimes n}$ and $\boldsymbol{\eta}$ is the corresponding vector of Pauli weights, i.e., $\eta_k=\frac{1}{2^n}{\rm Tr}(\sigma_k^{(n)}\rho)$. The values η_k were computed on the quantum device using 8192 shots, then used to classically calculate the fidelity $F(N)=\langle\psi(N)|\rho(N)|\psi(N)\rangle$.

[1] R. P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. **21**, 467 (1982).

[2] S. Lloyd, Universal quantum simulators, Science, **273** 1073 (1996).

[3] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[4] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, Phys. Rev. Lett. **114**, 090502 (2015).

[5] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).

[6] C. Cirstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, Variational fast forwarding for quantum simulation beyond the coherence time, npj Quantum Inf. **6**, 82 (2020).

[7] B. Commeau, M. Cerezo, Z. Holmes, L. Cincio, P. J. Coles, and A. Sornborger, Variational Hamiltonian diagonalization for dynamical quantum simulation, arXiv:2009.02559.

[8] J. Gibbs, K. Gili, Z. Holmes, B. Commeau, A. Arrasmith, L. Cincio, P. J. Coles, and A. Sornborger, Long-time simulations for fixed input states on quantum hardware, npj Quantum Inf. **8**, 135 (2022).

[9] M. R. Geller, Z. Holmes, P. J. Coles, and A. Sornborger, Experimental quantum learning of a spectral decomposition, Phys. Rev. Res. **3**, 033200 (2021).

[10] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[11] M. Schuld, Quantum machine learning models are kernel methods, arXiv:2101.11020.

[12] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, Quantum advantage in learning from experiments, Science **376**, 1182 (2022).

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

[14] S. Lloyd, Quantum approximate optimization is computationally universal, arXiv:1812.11075.

[15] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, npj Quantum Inf. **5**, 103 (2019).

[16] M. C. Caro and I. Datta, Pseudo-dimension of quantum circuits, Quantum Mach. Intell. **2**, 14 (2020).

[17] K. Bu, D. E. Koh, L. Li, Q. Luo, and Y. Zhang, Statistical complexity of quantum circuits, Phys. Rev. A **105**, 062431 (2022).

[18] C. Gyurik, D. Vreumingen, van, and V. Dunjko, Structural risk minimization for quantum linear classifiers, Quantum **7**, 893 (2023).

[19] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nat. Comput. Sci. **1**, 403 (2021).

[20] Y. Du, Z. Tu, X. Yuan, and D. Tao, Efficient measure for the expressivity of variational quantum algorithms, Phys. Rev. Lett. **128**, 080506 (2022).

[21] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert, and R. Sweke, Encoding-dependent generalization bounds for parametrized quantum circuits, Quantum **5**, 582 (2021).

[22] C.-C. Chen, M. Watabe, K. Shiba, M. Sogabe, K. Sakamoto, and T. Sogabe, On the expressibility and overfitting of quantum circuit learning, ACM Trans. Quantum Comput. **2**, 1 (2021).

[23] C. M. Popescu, Learning bounds for quantum circuits in the agnostic setting, Quant. Info. Proc. **20**, 286 (2021).

[24] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, Generalization in quantum machine learning from few training data, Nat. Commun. **13**, 4919 (2022).

[25] H. Cai, Q. Ye, and D.-L. Deng, Sample complexity of learning parametric quantum circuits, Quantum Sci. Technol. **7**, 025014 (2022).

[26] M. C. Caro, H.-Y. Huang, N. Ezzell, J. Gibbs, A. T. Sornborger, L. Cincio, P. J. Coles, and Z. Holmes, Out-of-distribution generalization for learning quantum dynamics, Nat. Commun. **14**, 3751 (2023).

[27] K. Poland, K. Beer, and T. J. Osborne, No free lunch for quantum machine learning, arXiv:2003.14103.

[28] K. Sharma, M. Cerezo, Z. Holmes, L. Cincio, A. Sornborger, and P. J. Coles, Reformulation of the no-free-lunch theorem for entangled datasets, Phys. Rev. Lett. **128**, 070501 (2022).

[29] T. Volkoff, Z. Holmes, and A. Sornborger, Universal compiling and (no-)free-lunch theorems for continuous-variable quantum learning, PRX Quantum **2**, 040327 (2021).

[30] E. Kökcü, T. Steckmann, J. Freericks, E. F. Dumitrescu, and A. F. Kemper, Fixed depth Hamiltonian simulation via Cartan decomposition, Phys. Rev. Lett. **129**, 070501 (2022).

[31] T. Steckmann, T. Keen, A. F. Kemper, E. F. Dumitrescu, and Y. Wang, Simulating the mott transition on a noisy digital quantum computer via Cartan-based fast-forwarding circuits, arXiv:2112.05688.

[32] S. K. Radha, Quantum constraint learning for quantum approximate optimization algorithm, arXiv:2105.06770.

[33] H. Horowitz, P. Rao, and S. K. Radha, A quantum generative model for multi-dimensional time series using Hamiltonian learning, arXiv:2204.06150.

[34] We can replace the assumption of independent Haar-random single-qubit states by an assumption of independent states drawn at random according to a 1-qubit 2-design and our error analysis will still hold.

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

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

[37] M. Cerezo and P. J. Coles, Higher order derivatives of quantum neural networks with barren plateaus, Quantum Sci. Technol. **6**, 035006 (2021).

[38] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, Effect of barren plateaus on gradient-free optimization, Quantum **5**, 558 (2021).

[39] Z. Holmes, A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, Barren plateaus preclude learning scramblers, Phys. Rev. Lett. **126**, 190501 (2021).

[40] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, PRX Quantum **3**, 010313 (2022).

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

[42] K. Sharma, M. Cerezo, L. Cincio, and P. J. Coles, Trainability of dissipative perceptron-based quantum neural networks, Phys. Rev. Lett. **128**, 180505 (2022).

[43] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Absence of barren plateaus in quantum convolutional neural networks, Phys. Rev. X **11**, 041011 (2021).

[44] A. Uvarov and J. D. Biamonte, On barren plateaus and cost function locality in variational quantum algorithms, J. Phys. A: Math. Theor. **54**, 245301 (2021).

[45] C. O. Marrero, M. Kieferová, and N. Wiebe, Entanglement-induced barren plateaus, PRX Quantum **2**, 040316 (2021).

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

[47] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum-assisted quantum compiling, Quantum **3**, 140 (2019).

[48] Note that by relating fidelity to trace distance and then invoking the operational meaning of the latter in terms of observable differences [55], the simulation fidelity gives rise to an upper bound on the difference between the left- and right-hand sides in Eq. (1).

[49] F. Verstraete, J. I. Cirac, and J. I. Latorre, Quantum circuits for strongly correlated quantum systems, Phys. Rev. A **79**, 032316 (2009).

[50] For more details on the ansatz we used see Appendix C.

[51] Quantum state tomography is used to reconstruct the output density matrix, ρ, of the noisy fast forwarding, which is used to compute the fast-forwarding fidelity, $\langle\psi|\rho|\psi\rangle$, where $|\psi\rangle$ is the target noise-free state.

[52] Y. Atia and D. Aharonov, Fast-forwarding of Hamiltonians and exponentially precise measurements, Nat. Commun. **8**, 1572 (2017).

[53] 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).

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

[55] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[56] M. Horodecki, P. Horodecki, and R. Horodecki, General teleportation channel, singlet fraction, and quasidistillation, Phys. Rev. A **60**, 1888 (1999).

[57] M. A. Nielsen, A simple formula for the average gate fidelity of a quantum dynamical operation, Phys. Lett. A **303**, 249 (2002).

[58] A. Kitaev, Anyons in an exactly solved model and beyond, Ann. Phys. **321**, 2 (2006).

[59] D. J. Wales and J. P. Doye, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms, J. Phys. Chem. A **101**, 5111 (1997).

[60] G.-L. R. Anselmetti, D. Wierichs, C. Gogolin, and R. M. Parrish, Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems, New J. Phys. **23**, 113010 (2021).