Probabilistic simulation of quantum circuits using a deep-learning architecture

Juan Carrasquilla,[1] Di Luo,[2] Felipe Pérez,[3] Ashley Milsted,[4] Bryan K. Clark,[2] Maksims Volkovs,[3] and Leandro Aolita[5]

[1] *Vector Institute, MaRS Centre, Toronto, Ontario, Canada M5G 1M1,*
[2] *Institute for Condensed Matter Theory and IQUIST and Department of Physics,*
University of Illinois at Urbana-Champaign, Illinois 61801, USA
[3] *Layer6 AI, MaRS Centre, Toronto, Ontario, Canada M5G 1M1*
[4] *Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, Ontario, Canada N2L 2Y5*
[5] *Instituto de Física, Universidade Federal do Rio de Janeiro, Caixa Postal 68528, Rio de Janeiro, RJ 21941-972, Brazil*

(Received 23 April 2021; revised 19 August 2021; accepted 19 August 2021; published 20 September 2021)

The fundamental question of how to best simulate quantum systems using conventional computational resources lies at the forefront of condensed matter and quantum computation. It impacts both our understanding of quantum materials and our ability to emulate quantum circuits. Here we present an exact formulation of quantum dynamics via factorized generalized measurements which maps quantum states to probability distributions with the advantage that local unitary dynamics and quantum channels map to local quasistochastic matrices. This representation provides a general framework for using state-of-the-art probabilistic models in machine learning for the simulation of quantum many-body dynamics. Using this framework, we have developed a practical algorithm to simulate quantum circuits using an attention network based on a powerful neural network ansatz responsible for the most recent breakthroughs in natural language processing. We demonstrate our approach by simulating circuits that build Greenberger-Horne-Zeilinger and linear graph states of up to 60 qubits, as well as a variational quantum eigensolver circuit for preparing the ground state of the transverse field Ising model on several system sizes. Our methodology constitutes a modern machine learning approach to the simulation of quantum physics with applicability both to quantum circuits as well as other quantum many-body systems.

DOI: 10.1103/PhysRevA.104.032610

I. INTRODUCTION

In his celebrated keynote address at the California Institute of Technology in May 1981, Feynman introduced the idea of a computer that could act as a quantum mechanical simulator [1], which has inspired the field of quantum computing since its inception. In his keynote, Feynman also intriguingly asked “can quantum systems be probabilistically simulated by classical computer?,” which he answered negatively observing that a probabilistic simulation is unfeasible since the description of both the quantum state and its evolution necessarily involves nonpositive quasiprobabilities. In fact, quantum computers will display potential speedups over their classical counterparts at the onset of negative values in the quasiprobabilities associated with the description and evolution of their quantum states. This observation about negative probabilities eventually stimulated the field of quantum computation.

Given the difficulty of simulating quantum computers probabilistically, it is interesting to instead ask what alternatives exist for classical simulations of quantum circuits. One promising approach is to compress the quantum state into a compact representation and then update this compact representation upon the application of each quantum gate. The nonpositive quasiprobabilities contribute to making even this approach difficult as the signs induce rapid oscillations that are naively more difficult to compress.

One area where there has been significant work in compressing large vectors is in machine learning where exponentially large probability distributions are commonly compressed into generative models. The most mature of these is in the area of language modeling and translation where neural probabilistic models such as transformers [2] encode the probability that a given string of characters results in a sensible conversation. Recently, such models have been used in the context of quantum state reconstruction [3]. Such a strategy resulted in an accurate quantum state representation of families of prototypical states in quantum information as well as complex ground states of one- and two-dimensional local Hamiltonians describing large many-body systems relevant to condensed matter, cold atomic systems, and quantum simulators [3].

To use this technology, it is important to be able to map a quantum state to a probability distribution. One might naively expect to simply consider the state’s amplitude but this loses critical phase information. Although the presence of negative quasiprobabilities is often linked to intrinsically quantum phenomena with no classical counterpart such as entanglement and quantum interference, a purely probabilistic representation of the quantum state is possible [3–7]. While in the standard formulation of quantum mechanics a quantum state is represented by a density operator, a quantum state can also be completely specified by the outcome probability of a physical measurement, provided that the measurement probes enough information about the quantum state. This notion is made precise through two fundamental concepts in quantum theory: The so-called Born rule, which is the theoretical principle of quantum physics linking quantum theory and experiment, and the concept of informationally complete (IC) measurements, which are described by positive-operator valued measures (POVMs). Whereas POVMs describe the most general type of measurements allowed by quantum theory going beyond the notion of projective measurements [8], informational completeness means that the outcome statistics of such a measurement specifies the quantum state unambiguously.

To compactly represent these probability distributions, we will use an autoregressive model to store the instantaneous state in its probabilistic representation. We then develop a powerful stochastic algorithm to update the probabilistic model representing the quantum state under the application of unitary dynamics. We note that other approaches [9–15] to compactly represent and update states of a quantum circuit exist.

The choice of autoregressive models is motivated in various ways. To begin with, such models are known to be able to capture long-range correlations and volume law states [16,17]. This would in principle allow them to capture states efficiently beyond the capabilities of matrix product states. In addition, our algorithms to update the compressed state after the application of a quantum gate require the use of Monte Carlo approaches. Typically, this would be done through a Markov chain Monte Carlo (MCMC) technique, but we emphasize that such MCMC methods are potentially affected by issues such as long autocorrelation times and lack of ergodicity, which effectively decrease in speed the simulations as well as affect the quality of the estimators used to update the models. Autoregressive models, and in particular a deep-learning architecture called "the Transformer," avoid all these problems by allowing for exact sampling, making the entire algorithm significantly more efficient.

We test our ideas by considering quantum circuits which prepare prototypical states in quantum information. In particular we consider the Greenberger-Horne-Zeilinger (GHZ) state, linear graph state, and the variational ground state of the transverse field Ising model (TFIM). Through numerical experiments, we show that our strategy produces accurate results for the target states of up to 60 qubits, which opens up a probabilistic avenue for simulation of quantum circuits, as well as quantum channels and quantum dynamics more broadly.

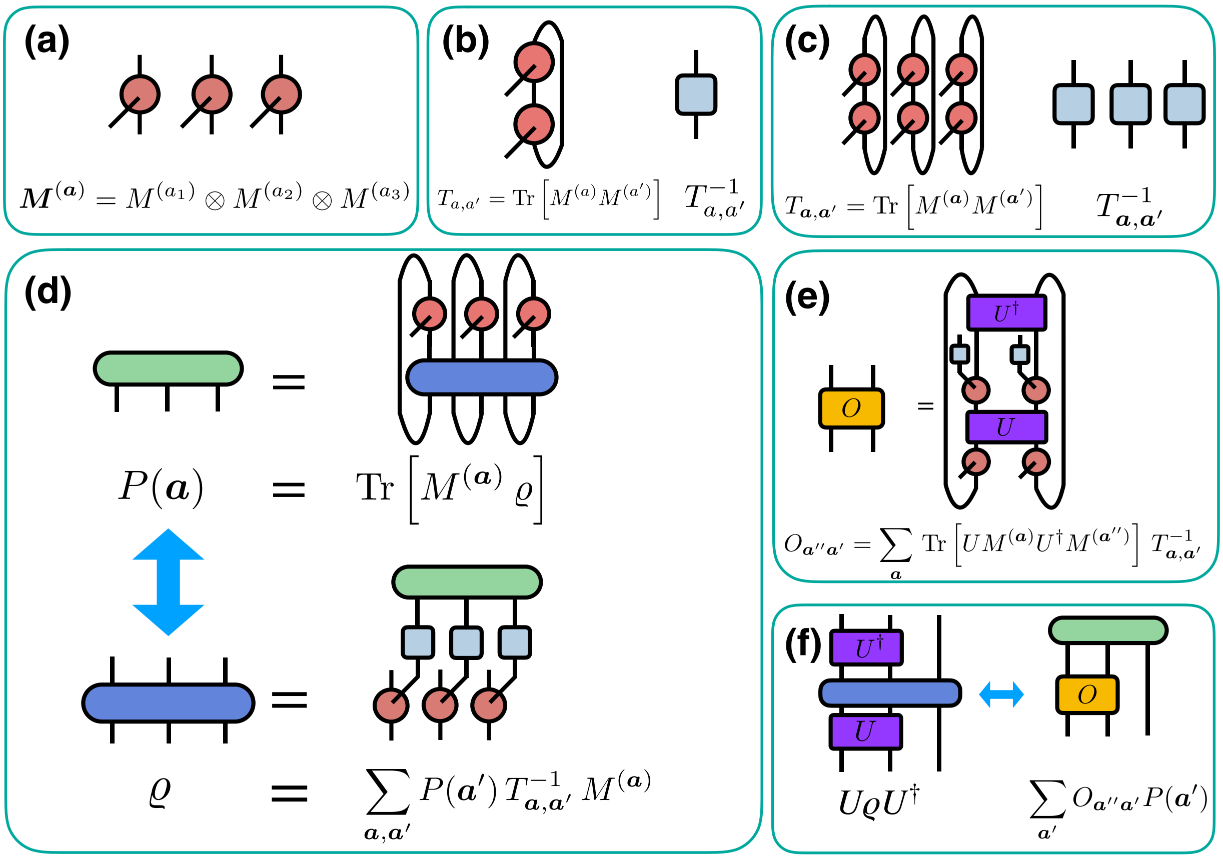
(a)
$\boldsymbol{M}^{(\boldsymbol{a})} = M^{(a_1)} \otimes M^{(a_2)} \otimes M^{(a_3)}$
(b)
$T_{a,a'} = \mathrm{Tr}\left[M^{(a)}M^{(a')}\right]$
$T^{-1}_{a,a'}$
(c)
$T_{\boldsymbol{a},\boldsymbol{a}'} = \mathrm{Tr}\left[M^{(\boldsymbol{a})}M^{(\boldsymbol{a}')}\right]$
$T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}$
(d)
$=$
$P(\boldsymbol{a}) = \mathrm{Tr}\left[M^{(\boldsymbol{a})}\,\varrho\right]$
$=$
$\varrho = \sum_{\boldsymbol{a},\boldsymbol{a}'} P(\boldsymbol{a}')\,T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\,M^{(\boldsymbol{a})}$
(e)
$U^\dagger$
O
$=$
U
$O_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} \mathrm{Tr}\left[UM^{(\boldsymbol{a})}U^\dagger M^{(\boldsymbol{a}'')}\right] T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}$
(f)
$U^\dagger$
U
O
$U\varrho U^\dagger$
$\sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'}P(\boldsymbol{a}')$

FIG. 1. Tensor-network representation of the mapping between quantum states (gates) and probability distributions (quasistochastic matrices) used in this work. (a) N-qubit measurement $\boldsymbol{M} = \{M^{(a_1)} \otimes M^{(a_2)} \otimes \cdots M^{(a_N)}\}_{a_1,\ldots,a_N}$ made from ($N = 3$) single-qubit measurements $\{M^{(a)}\}_a$ (red). Vertical indices in the red tensors act on the physical degrees of freedom (qubits) while the horizontal index labels the measurement outcome a. (b) The overlap matrix T and its inverse T^{-1} (light blue) (c) multiqubit version of (b). (d) The Born rule relates the probability $P(\boldsymbol{a})$ (green; indices encode the different measurement outcomes on each qubit) to the quantum state ϱ (blue). (e) Unitary gates (purple) map to quasistochastic matrices (yellow). (f) Application of a unitary matrix to a density matrix corresponds to the contraction of a quasistochastic matrix with P.

II. FORMALISM

We focus on physical systems composed of N qubits whose quantum state, traditionally represented by a density matrix ϱ, will be uniquely specified by the measurement statistics of an informationally complete POVM (IC-POVM). To build an IC-POVM for N qubits, we first consider an m-outcome single-qubit IC-POVM defined by a collection $\{M^{(a)}\}_{a\in\{1\ldots m\}}$, of positive semidefinite operators $M^{(a)} \geqslant 0$, each one labeled by a measurement outcome $a = 0, 1, \ldots, m-1$ [see Fig. 1(a) where we describe our representation through the lens of tensor networks and its graphical notation [18]]. Following Ref. [3], we construct N-qubit measurements as tensor products of the single-qubit IC-POVM elements $\boldsymbol{M} = \{M^{(a_1)} \otimes M^{(a_2)} \otimes \cdots M^{(a_N)}\}_{a_1,\ldots,a_N\in\{1\ldots m\}^N}$, as graphically depicted in Fig. 1(a). We choose for our numerical simulations the four-Pauli IC-POVM measurement described in Ref. [3], $\{M^{(0)} = \frac{1}{3}|0\rangle\langle 0|,\ M^{(1)} = \frac{1}{3}|+\rangle\rangle+|,\ M^{(2)} = \frac{1}{3}|r\rangle\langle r|,\ M^{(3)} = \mathbb{1} - M^{(0)} - M^{(1)} - M^{(2)}\}$. Here $|0\rangle, |+\rangle, |r\rangle$ are the eigenvectors with eigenvalue $+1$ of the Pauli matrices $\sigma^x, \sigma^y, \sigma^z$, respectively. Note that this is a natural choice for quantum circuits since the probability distribution over these operators can easily be measured on currently available gate-based quantum computers. Born's rule predicts that the probability distribution $\boldsymbol{P} = \{P(\boldsymbol{a})\}_{\boldsymbol{a}=(a_1,a_2,\ldots,a_N)}$ over measurement outcomes $\boldsymbol{a}$ on a quantum state ϱ is given by the following:

$$P(\boldsymbol{a}) = \mathrm{Tr}[M^{(\boldsymbol{a})}\varrho], \tag{1}$$

which is graphically explained in Fig. 1(d). Note that a quantum state is specified by m^N probabilities. Due to the factorized nature of the IC-POVM, a product state $\bigotimes_i |\Psi_i\rangle$ takes the form of a product distribution over statistically independent sets of variables $P(\boldsymbol{a}) = P(a_1)P(a_2)\cdots P(a_N)$ where $P(a_i) = \mathrm{Tr}[M^{(a_i)}|\Psi_i\rangle\langle\Psi_i|]$. Provided that the measurement is informationally complete, the density matrix can be inferred from the statistics of the measurement outcome as

$$\varrho = \sum_{\boldsymbol{a},\boldsymbol{a}'} P(\boldsymbol{a}')\, T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} M^{(\boldsymbol{a})}, \tag{2}$$

where T represents the overlap matrix given by $T_{\boldsymbol{a},\boldsymbol{a}'} = \mathrm{Tr}[M^{(\boldsymbol{a})}M^{(\boldsymbol{a}')}]$. See Fig. 1(d) for a graphical representation of these elements in Eq. (2).

To study quantum circuits, we first have to translate the action of a quantum gate on the density matrix in the IC-POVM representation. The former corresponds to a unitary transformation, i.e., $\varrho_U = U\varrho U^\dagger$. If the initial quantum state is prescribed in terms of the outcome statistics of an IC-POVM $\boldsymbol{P}$, we can track its evolution directly in the probabilistic representation:

$$P_U(\boldsymbol{a}'') = \mathrm{Tr}[U\varrho U^\dagger M^{(\boldsymbol{a}'')}] = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}'), \tag{3}$$

where

$$O_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} \mathrm{Tr}[UM^{(\boldsymbol{a})}U^\dagger M^{(\boldsymbol{a}'')}]T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} \tag{4}$$

is a *somewhat* stochastic matrix since the values in each column add up to 1 but its entries can be positive or negative [5,6,19,20]. Somewhat stochastic matrices are also known as pseudostochastic or quasistochastic matrices [5,6]. We note that the evolution described in Eq. (3) leads to a formulation of quantum mechanics equivalent to, e.g., Heisenberg's matrix mechanics, including the description of open quantum systems, quantum channels, and measurements of other POVMs (see Appendices A, B, C, and D).

Here we emphasize that Eq. (3) resembles the standard rule for stochastic evolution commonly used to describe the transitions in a Markov chain, where the traditional stochastic (or Markov) matrix has been replaced with a quasistochastic matrix. Despite the resemblance, a generic classical MCMC simulation of quantum evolution in the probabilistic factorized POVM language remains unfeasible due to the numerical sign problem arising from the negative entries of the quasistochastic matrix describing the process.

Due to the factorized nature of the IC-POVM, if a unitary matrix or a quantum channel acts nontrivially on only k qubits of the quantum system, the quasistochastic matrix $O_{\boldsymbol{a}''\boldsymbol{a}'}$ acts only on the measurement outcomes of those k qubits too. For example, a two-qubit unitary gate acting on qubits i and j is represented by a $m^2 \times m^2$ quasistochastic matrix acting on outcomes a_i and a_j. The relation between the local quasistochastic matrices and the local unitary gates, as well as their action on a quantum state, are graphically depicted in Figs. 1(e) and 1(f) using tensor diagrams. Furthermore, the locality of $O_{\boldsymbol{a}''\boldsymbol{a}'}$ implies that traditional quantum circuit diagrams [8] translate into probabilistic circuits that look exactly the same as their traditional counterparts.

A quantum circuit is a generalization of the circuit model of classical computation where a product state is evolved through a series of unitary gates, $U^{(1)}, U^{(2)}, \ldots, U^{(r)}$, each of which acts nontrivially on a constant number k qubits. Note that for each gate $U^{(i)}$ there is a corresponding somewhat stochastic matrix $O^{(i)}$ as in Eq. (4). In the IC-POVM representation, an initial probability distribution $P_0(\boldsymbol{a}) = P(a_1)P(a_2)\cdots P(a_N)$ of statistically independent sets of variables $\boldsymbol{a}$ is evolved through a series of local quasistochastic matrices of the form depicted in Fig. 1(e). The measurement statistics after unitary evolution through the first gate $U^{(1)}$ is given by $\boldsymbol{P}_1 = O^{(1)}\boldsymbol{P}_0$, and the application of each subsequent gate $U^{(i)}$ defines a series of intermediate probability distributions $\boldsymbol{P}_i = O^{(i)}O^{(i-1)}\cdots O^{(2)}O^{(1)}\boldsymbol{P}_0$ with $i = 1, \ldots, r$. The main goal of our approach is to accurately represent the distribution $\boldsymbol{P}_r$ since it contains all the information of the final quantum state which specifies the outcome of the quantum computation.

III. OPTIMIZATION ALGORITHMS

The strategy to approximate the output distribution $\boldsymbol{P}_r$ consists in constructing models $\boldsymbol{P}_{\theta_i} = \{P(\boldsymbol{a};\theta_i)\}_{\boldsymbol{a}=(a_1,a_2,\ldots,a_N)}$ based on a rich family of probability distributions $\boldsymbol{P}(\boldsymbol{a};\theta)$. These are expressed in terms of a neural network with parameters θ so that $\boldsymbol{P}_{\theta_i} \approx \boldsymbol{P}_i$. At each time step i, we assume that an accurate neural approximation has been reached $\boldsymbol{P}_{\theta_i} \approx \boldsymbol{P}_i$, and consider the exactly evolved distribution $\boldsymbol{P}^{(e)}_{i+1} \equiv O^{(i+1)}\boldsymbol{P}_{\theta_i}$. While the representation of the quantum state at step i is not exact, if $\boldsymbol{P}_{\theta_i}$ is sufficiently accurate the expectation is that the distribution $\boldsymbol{P}_{\theta_{i+1}} \approx \boldsymbol{P}_{i+1}$. See Fig. 2 for a depiction of the distributions involved during the simulation.

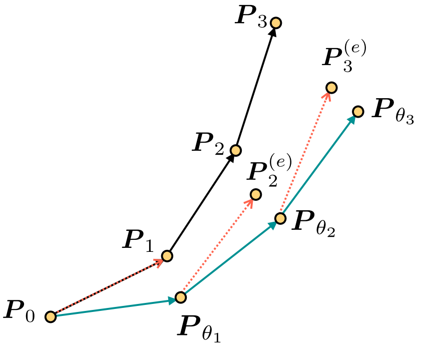

FIG. 2. Schematics of the different distributions involved in the training procedure for a circuit with three gates. Black arrows indicate the exact trajectory while green arrows represent the trajectory of the optimized models $\boldsymbol{P}_{\theta_i}$. The red dotted arrows point to the exactly evolved distributions $\boldsymbol{P}^{(e)}_{i+1}$ after the application of each gate $O^{(i+1)}$ on the trained model $\boldsymbol{P}_{\theta_i}$. Note that if the resulting KL divergence after training $\mathrm{KL}(\boldsymbol{P}^{(e)}_{i+1}||\boldsymbol{P}_{\theta_{i+1}}) = 0$ for every gate in the circuit, then the black trajectory coincides with the green one and the procedure becomes exact.

To train the model $\boldsymbol{P}_{\theta_i}$ given a gate $i+1$, we adopt a variational approach and select the parameters θ_{i+1} such that the Kullback-Liebler (KL) divergence between $\boldsymbol{P}^{(e)}_{i+1}$ and $\boldsymbol{P}_{\theta_{i+1}}$,

$$\mathrm{KL}\big(\boldsymbol{P}^{(e)}_{i+1}\big|\big|\boldsymbol{P}_{\theta_{i+1}}\big) = -\sum_{\boldsymbol{a}} P^{(e)}_{i+1}(\boldsymbol{a}) \ln\left(\frac{P_{\theta_{i+1}}(\boldsymbol{a})}{P^{(e)}_{i+1}(\boldsymbol{a})}\right) \tag{5}$$

is minimized. Recall $\mathrm{KL}(\boldsymbol{P}^{(e)}_{i+1}||\boldsymbol{P}_{\theta_{i+1}}) \geqslant 0$, with the equality being satisfied only when $\boldsymbol{P}_{\theta_{i+1}} = \boldsymbol{P}^{(e)}_{i+1}$. To minimize the KL divergence, we will apply a variant of gradient descent (i.e., Adam [21]) where we repeatedly update the parameters of θ_{i+1} by taking steps in the direction of the gradient of Eq. (5). This gradient, assuming that the model is normalized, can be written as

$$\begin{aligned}
&\nabla_{\theta_{i+1}}\mathrm{KL}\big(\boldsymbol{P}^{(e)}_{i+1}||\boldsymbol{P}_{\theta_{i+1}}\big) && \text{(6a)}\\
&= -\sum_{\boldsymbol{a}} P^{(e)}_{i+1}(\boldsymbol{a})\nabla_{\theta_{i+1}} \ln\,[P_{\theta_{i+1}}(\boldsymbol{a})] && \text{(6b)}\\
&= -\mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})}\left[\frac{P^{(e)}_{i+1}(\boldsymbol{a})}{P_{\theta_{i+1}}(\boldsymbol{a})}\nabla_{\theta_{i+1}} \ln\,[P_{\theta_{i+1}}(\boldsymbol{a})]\right] && \text{(6c)}\\
&= -\mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})}\left[\left(\frac{P^{(e)}_{i+1}(\boldsymbol{a})}{P_{\theta_{i+1}}(\boldsymbol{a})} - k\right)\nabla_{\theta_{i+1}} \ln\,[P_{\theta_{i+1}}(\boldsymbol{a})]\right]. && \text{(6d)}
\end{aligned}$$

Here, $k = \mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})}[\frac{P^{(e)}_{i+1}(\boldsymbol{a})}{P_{\theta_{i+1}}(\boldsymbol{a})}]$. Note that $\mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})}\nabla_{\theta_{i+1}} \ln[P_{\theta_{i+1}}(\boldsymbol{a})] = 0$, which justifies the equality in Eq. (6d). To estimate the gradient in Eq. (6a), we generate a minibatch of N_s samples of $\boldsymbol{a}$ sampled from $P_{\theta_{i+1}}(\boldsymbol{a})$ and average over these samples both to compute the value of k as well as Eq. (6d). While computing Eq. (6d) or (6c) (both evaluate the gradient), Eq. (5d) has a significantly lower variance. In the limit where $P_{\theta_{i+1}}(\mathbf{a})$ approaches $P^{(e)}_{i+1}(\mathbf{a})$ (i.e., ideally towards the end of the optimization of step $i+1$), the variance of the gradient estimator Eq. (6d) goes to zero. This is known as the zero-variance principle [22]. For a fixed $\boldsymbol{a}$ we evaluate $P^{(e)}_{i+1}(\boldsymbol{a}) = \sum_{\boldsymbol{a}'} O^{(i+1)}_{\boldsymbol{a}\boldsymbol{a}'} P_{\theta_i}(\boldsymbol{a}')$ by explicitly summing over the substring of outcomes in $\boldsymbol{a}'$ on which $O^{(i+1)}$ acts (with the other outcomes in the string fixed).

By construction, the unitary matrices and their corresponding quasistochastic matrices considered here are k local, which means that the calculation of the gradient estimator in Eq. (6a) is efficient. Barring the cost of evaluating the probabilities $P_{\theta_{i+1}}(\boldsymbol{a})$ and $P_{\theta_i}(\boldsymbol{a})$, using N_s samples the gradient can be computed in $O(N_s 4^k)$, which is a significant improvement over the full multiplication $\boldsymbol{P}_{i+1} = O^{(i+1)}\boldsymbol{P}_i$ which takes $O(2^{2N})$.

Another algorithm we adopt in this paper is the forward-backward gate algorithm. Consider a unitary gate U and decompose it as $U = U_1 U_1$. Under the POVM transformation, U_1 is transformed into O_1 and in the exact evolution $O_1\boldsymbol{P}_{\theta_i}$ should match $O_1^T\boldsymbol{P}_{\theta_{i+1}}$. U_1 (O_1 under POVM transformation) and $U_1^\dagger$ (O_1^T under POVM transformation) can be considered as the forward and backward evolution gate separately. We define a cost function by optimizing the following:

$$\begin{aligned}
C &= \left\|O_1\boldsymbol{P}_{\theta_i} - O_1^T\boldsymbol{P}_{\theta_{i+1}}\right\|_1 && \text{(7a)}\\
&= \sum_{\boldsymbol{a}}\left|\sum_{\boldsymbol{a}'} O_{1,\boldsymbol{a}\boldsymbol{a}'}\boldsymbol{P}_{\theta_i}(\boldsymbol{a}') - O^T_{1,\boldsymbol{a}\boldsymbol{a}'}\boldsymbol{P}_{\theta_{i+1}}(\boldsymbol{a}')\right|. && \text{(7b)}
\end{aligned}$$

The gradient can be computed as follows:

$$\nabla_{\theta_{i+1}}C = \mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}}\frac{1}{P_{\theta_{i+1}}(\boldsymbol{a})}\sum_{\boldsymbol{a}'} O^T_{1,\boldsymbol{a}\boldsymbol{a}'}\nabla_{\theta_{i+1}}P_{\theta_{i+1}}(\boldsymbol{a}') \tag{8}$$

$$\times\,\mathrm{sgn}\left\{\sum_{\boldsymbol{a}'} O_{1,\boldsymbol{a}\boldsymbol{a}'}\boldsymbol{P}_{\theta_i}(\boldsymbol{a}') - O^T_{1,\boldsymbol{a}\boldsymbol{a}'}\boldsymbol{P}_{\theta_{i+1}}(\boldsymbol{a}')\right\}. \tag{9}$$

The details for optimization can be found in Appendix G.

IV. TRANSFORMER ARCHITECTURE

For simplicity, in order to model $\boldsymbol{P}_i$, we restrict to models $P_{\theta_i}(\boldsymbol{a})$ with a tractable density and exact sampling. While other models such as the variational autoencoder [25] can represent the quantum state probabilistically, having both a tractable density and exact sampling significantly simplifies the calculation of the quantities involved in the gradient estimation. The exact sampling avoids expensive MCMC simulation which would otherwise be required to obtain the samples for the gradient estimator. Specifically, we consider prototypical autoregressive models commonly used in neural machine translation and language modeling based on Transformer encoder blocks [2]. This neural architecture models a probability distribution $P_\theta(\boldsymbol{a})$ through its conditionals $P_\theta(a_{k+1}|a_1,\ldots,a_k)$. Note that we can recover $\boldsymbol{P}$ via the chain rule

$$P_\theta(a_1,\ldots,a_N) = \prod_{k=1}^{N} P_\theta(a_k|a_{<k}),$$

which we heavily use in our simulations.

The Transformer architecture is constructed using the elements depicted in Fig. 3. The first and most important element is the self-attention mechanism. Self-attention takes an em-

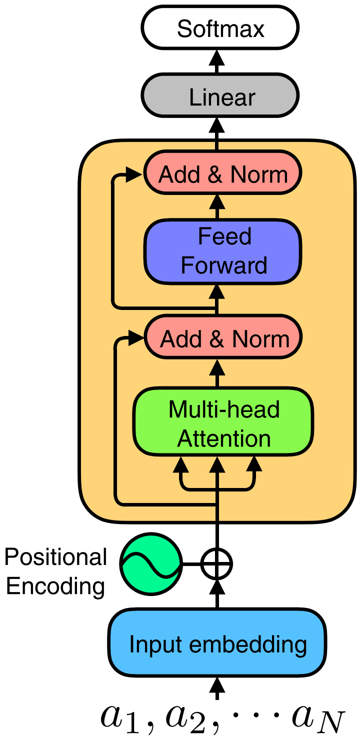

FIG. 3. Schematic representation of the Transformer model. Lines with arrowheads denote incoming arrays from the output of one node to the inputs of others. The Transformer architecture starts with an input measurement $\boldsymbol{a}$. First, a high-dimensional linear embedding $\hat{A}$ of the input measurement is computed. This is followed by the addition of positional encoding vectors to the input embeddings $\hat{A}$. The multihead attention mechanism is applied to the modified embedding, followed by a residual connection [23] and layer normalization [24]. A positionwise feedforward is then applied to the outcome of the previous layer again followed by a residual connection and layer normalization. The output of the last layer is processed through a linear layer followed by a softmax which returns the conditional probabilities $P_\theta(a_{k+1}|a_1, \dots, a_k)$.

bedding of the measurement outcome $\boldsymbol{a}$, and computes an autocorrelation matrix where the different measurement outcomes across the different qubits form the columns and rows. The embedding is a linear transformation on the original input $\boldsymbol{a}$, i.e., a trainable matrix multiplying a one-hot encoding of the input. The self-attention and its correlation matrix are useful to introduce correlations between qubits separated at any distance in the quantum system. This is analogous to a two-body Jastrow factor [26] which induces pairwise long-distance correlations between the bare degrees of freedom (i.e., spins, qubits, electrons) in a wave function. In contrast to traditional sequence models based on recurrent neural networks, which tend to suppress correlations beyond a certain length ξ, the self-attention networks are suitable to model systems exhibiting power-law correlations present in natural sequences as well as physical systems exhibiting (classical or quantum) critical behavior [16].

More precisely, the attention mechanism can be described as a map between a "query" array Q, a "key" array K, and "value" V, to an output vector. The query, keys, and values are linear transformations of the input vectors, e.g., $K = \hat{A}W^{(K)}$, where $\hat{A} \in \mathbb{R}^{N\times d_{\text{model}}}$ is a d_{model}-dimensional embedding of the measurements outcome at the different qubits $j = 1, \dots, N$, and $W^{(K)} \in \mathbb{R}^{d_{\text{model}}\times d_k}$ is a parameter of the model. Analogously, the values and queries are calculated as a parametrized linear transformation on the embedding $\hat{A}$.

The specific type of attention mechanism the Transformer uses is the so-called scaled dot-product attention. The input consists of queries and keys of dimension d_k, and values of dimension d_v, and the output is computed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{d_k}}\right)V, \tag{10}$$

where the softmax function acting on a vector results in $\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$. The argument of the softmax is $\hat{A}W^{(Q)}W^{(K)^{\text{T}}}\hat{A}^{\text{T}}$, which induces pairwise, all-to-all correlations between the qubits in the system, thus resembling a Jastrow factor with parameters $W^{(Q)}W^{(K)^{\text{T}}}$.

As in Ref. [2], we use a multihead attention mechanism where instead of computing a single attention function, we linearly project the queries, keys, and values h times with different, learned linear projections to d_k, d_k, and d_v dimensions. Each of these projections are then followed by the attention function in parallel, producing d_v-dimensional output values. These are concatenated and projected. The output of the multihead attention is

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^{(0)}, \tag{11}$$

where $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$, $K_i = \hat{A}W_i^{(K)}$, $Q_i = \hat{A}W_i^{(Q)}$, and $V_i = \hat{A}W_i^{(V)}$. Here, $W_i^{(K)} \in \mathbb{R}^{d_{\text{model}}\times d_k}$, $W_i^{(Q)} \in \mathbb{R}^{d_{\text{model}}\times d_k}$, and $W_i^{(V)} \in \mathbb{R}^{d_{\text{model}}\times d_v}$. In our work we use $h = 8$ attention heads, and $d_k = d_v = d_{\text{model}}/h$ with $d_{\text{model}} = 16$ or 32. Since the conditional probability requires that the later input information cannot be known to the prior input, a mask is added in the multihead attention.

Additionally, the Transformer features a positionwise feedforward network, which is a fully connected feedforward network applied to each position separately and identically. This layer consists of two linear transformations with a ReLU [27] activation in between.

Each sublayer (i.e., the self-attention and the positionwise feedforward network) has a residual connection around it, and is followed by a layer-normalization step. That is, the output of each sublayer is $\text{LayerNorm}[x + \text{Sublayer}(x)]$, where $\text{Sublayer}(x)$ is the function implemented by either the self-attention or the positionwise feedforward network. The residual connection [23] makes it simple for the architecture to perform the identity operation on the input x since $\text{Sublayer}(x)$ can easily be trained to output zeros. The layer normalization [24] is a technique to normalize the intermediate outcome of the sublayers to have zero mean and unit variance, which enables a more stable calculation of the gradients in Eq. (6a) and faster training. An encoder block is defined as the composition of one self-attention layer and one positionwise feedforward layer with residual connection and layer normalization as the orange part in Fig. 3 shows. A number of encoder blocks can be further composed to enhance the expressiveness of the model and the number of encoder blocks is denoted as n_{ed}.

The embedding mentioned earlier converts the values of measurements $\boldsymbol{a}$ to vectors of dimension d_{model} through a parametrized linear transformation. Since the Transformer model contains no recurrence or convolutions, the model cannot naturally use the information of the spatial ordering of the qubits. To fix this, we include information about the relative or absolute position of the measurements in the system by adding positional encodings to the input embeddings. The positional encodings have the same dimension d_{model} as the embeddings and are added to the original embedding [2]. The last element of the Transformer is a linear layer followed by a softmax activation that outputs the conditional distribution.

V. APPLICATIONS

A. GHZ state and linear graph state preparation

We first demonstrate our approach on quantum circuits that produce a GHZ state and one-dimensional graph states. We use a variety of quality metrics to quantify the efficacy of our method: The KL divergence of Eq. (5), the classical fidelity $F_c(\boldsymbol{P},\boldsymbol{Q}) = \sum_{\boldsymbol{a}} \sqrt{P(\boldsymbol{a})Q(\boldsymbol{a})}$, and the L_1 norm of the probability distributions are all designed to measure the difference between the probability distribution of the neural probabilistic model and the exact probability distribution (either $\boldsymbol{P}_{i+1}$ or $\boldsymbol{P}^{(e)}_{i+1}$) of the POVM. Note that these measures directly bound how far off the POVM measurement statistics of the actual quantum state differ from our simulation. These measures depend on the POVM basis; we can also directly compare basis-independent quantities such as the quantum fidelity of the state,

$$F(\varrho_1, \varrho_2) = \mathrm{Tr}\left[\sqrt{\sqrt{\varrho_1}\,\varrho_2\,\sqrt{\varrho_1}}\right] \tag{12}$$

and

$$F_2(\varrho_1, \varrho_2) = \sqrt{1 - \|\varrho_1 - \varrho_2\|_F^2/2}\,. \tag{13}$$

Here we would like to make a few remarks on the different measures that are used in the paper. F and F_2 are equal to the overlap $|\langle\Psi_1|\Psi_2\rangle|$ when ϱ_1 and ϱ_2 represent pure states. The quantum fidelity F in Eq. (12) is standard for comparing density matrices in quantum information science [28]. In addition, $1 - F$ and $\sqrt{1 - F^2}$ provide a lower bound and an upper bound for the trace distance between quantum states. For F_2, it is a general norm for matrices and it is well defined even if the "density matrices" generated from the POVM probability do not correspond to physical density matrices. In terms of measures on POVM representation of the quantum states, the KL divergence used in Eq. (5) is the objective function used in optimization and hence indicates its importance. The classical fidelity F_c is also widely used in the literature [3,29–32] and provides an upper bound of the quantum fidelity F. In addition, since the POVMs are physical observables, F_c also encodes the quality of the measurements statistics with respect to the measuring of the four-Pauli POVMs. Besides the KL divergence and F_c, we will also compute the L_1 distance between two states in the POVM representation. It is worth noticing that the L_1 distance of the classical distribution is twice that of the total variance distance, which has a quantum generalization as the trace distance. Note that of these observables, the ones that are computable on large systems (in polynomial time) are KL divergence, F_c, and the L_1 distance making them suitable choices for comparisons on a larger number of qubits.

Figures 4(a)–4(d) show these measures for two quantum gates [see Fig. 4(a)(inset)] which generate the GHZ state with $N = 2$ qubits, namely, the Bell state $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

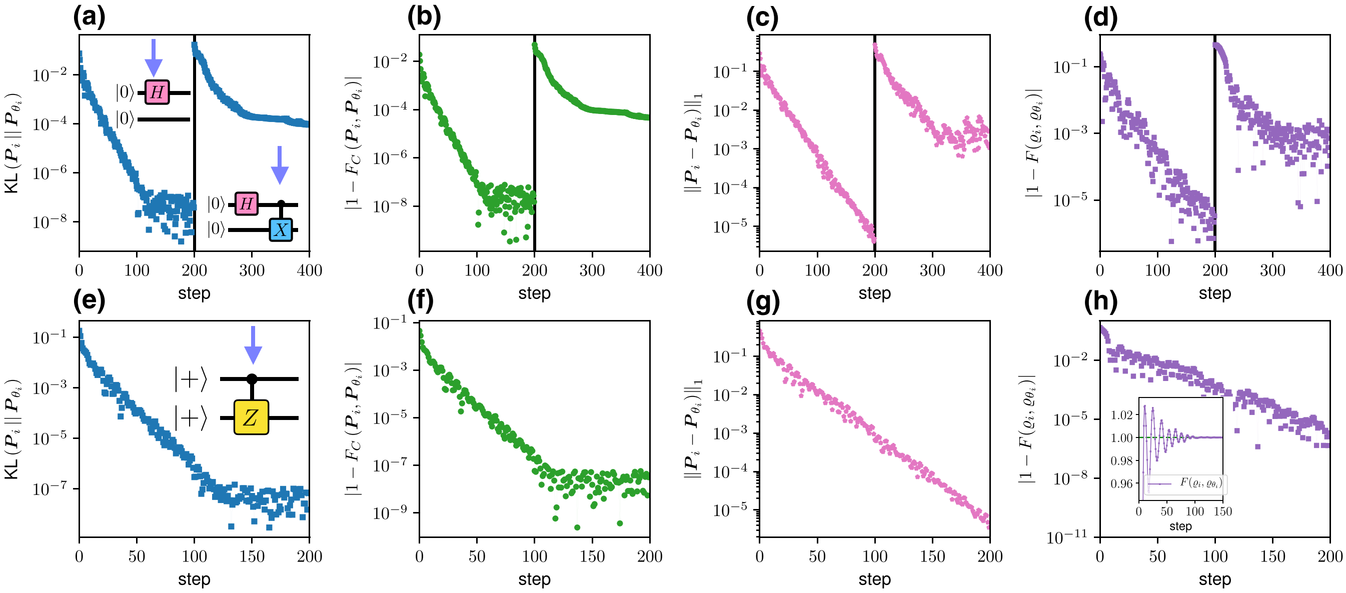

FIG. 4. Measures of training of two qubit circuits [shown in the insets of (a) and (e)] between the exact state and the quantum state represented by a Transformer ($d_{\text{model}} = 16$, $n_{ed} = 1$) after each circuit element. KL divergence (a) and (e), classical fidelity (b) and (f), L_1 norm (c) and (g), and the quantum fidelity (d) and (h). The main panels use a log-linear scale whereas the inset in (h) displays the fidelity in linear scale.

For the application of each of the two gates, the KL divergence, the L_1 classical error, the classical fidelity error, and the quantum fidelity error all initially approach zero exponentially in the number of steps. The quantum fidelity oscillates around 1 until finally settling at 1 by the end of the optimization. The KL divergence and the classical fidelity error both eventually saturate, but, interestingly, the L_1 classical error and the quantum fidelity error both continue to improve for the application of the first gate in the circuit. This suggests that further improvements due to better training of the ansatz and a better choice of objective function are possible. The observed saturation at $\sim 10^{-8}$ also suggests some quantities are limited by the 32-bit floating-point precision used in computations. Increasing the precision could also lead to improved convergence. In Figs. 4(e)–4(h), we present analogous results for a circuit generating a two-qubit graph state, i.e., $|\Psi\rangle = \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle - |11\rangle)$, where we observe similar behavior. Note that for these examples, we are primarily probing the quality of our optimization given that the Transformer with hidden dimension 16 should be powerful enough to exactly represent the exact probability distribution.

The small oscillations of the fidelity above 1.0 evident in the inset of Fig. 4(h) exists because the Transformer model can represent probability distributions without a corresponding physical density matrix. This is because only a subset of the probability simplex, which is the space where the distributions expressed by the Transformer live, corresponds to physical density matrices upon inversion in Eq. (2). The subset of probability distributions with a valid quantum state in our setting forms a convex set similar to the so-called *Qplex* in quantum Bayesian theory [33]. Here we emphasize that the fidelity values in Figs. 4(d) and 4(h) eventually converge to 1 and the oscillations above and below 1 are suppressed exponentially with the training steps, suggesting that the model converges to the target quantum state. We provide more details about the Qplex and the presence of unphysical states in our representation in Appendix E.

We now turn our attention to the quality of the circuit simulation as a function of the number of qubits in the circuit, letting both the depth and gate number grow linearly with the number of qubits. We find in constructing GHZ and linear graph states [see Fig. 5 (inset)] in the range from 10 to 60 qubits that the classical fidelity falls approximately linearly with number of qubits (see Fig. 5) reaching a classical fidelity of approximately 0.9 at 60 qubits. In addition, we have considered two different hidden dimensions, 16 and 32, and find that there is an improvement of the classical fidelity over all qubit sizes as we increase the hidden dimension. We attribute this to an improved representability power of the larger transformers suggesting that one of the bottlenecks of our simulation is the ability of our neural probabilistic model to represent the probability distribution that corresponds to the output of the quantum circuit.

B. Simulation of VQE circuits

We further apply our method to simulate variational circuits for the ground state of the transverse Ising field model at the critical point [34]. We start with state preparation of a six-qubit TFIM ground state using a variational quantum eigensolver circuit [34] (see Appendix F). The variational quantum eigensolver [35] (VQE) is a quantum-classical hybrid algorithm that can be used to approximate the lowest energy eigenvalues and eigenvectors of a qubit Hamiltonian H on a quantum processor. Rather than performing an optimization of the VQE ansatz, we focus on the probabilistic preparation of an already optimized VQE circuit for the ground state of the TFIM, as demonstrated below. The simulation is performed by optimizing the KL divergence in Eq. (5). We note that the particular circuit we consider has more gates per qubit than our previous examples. However, we limit our simulation to a small number of qubits so that the estimation of quantum fidelity, whose computational cost is exponential in the number of qubits in our approach [3], remains possible. Thus we evaluate both classical and quantum infidelity between the Transformer model and the exact state at each step after the application of each quasistochastic gate in the circuit (see Appendix F for a precise specification of the quantum circuit and details of its probabilistic preparation). Both the classical and quantum infidelities shown in Fig. 6 increase with the number of gates in the circuit; in fact, as demonstrated in Appendix F in Fig. 12, there is a correlation between the classical and quantum fidelity as classical fidelity approaches 1. It is natural to expect that the increase of infidelity observed in our simulation is brought on by an accumulation of errors building up after successive gates in the circuit. We can give further evidence of this by looking at the error made after a single step, $1 - F_2(\varrho_i, \varrho_i^{(e)})$, which directly compares $\boldsymbol{P}_i^{(e)}$ and $\boldsymbol{P}_{\theta_i}$ (see Fig. 6). We find that the single-step error is roughly constant and small throughout the circuit suggesting each step of the simulation is fairly accurate. This is consistent with the observations in Fig. 4.

We further extend the simulation to the VQE circuits for system size $L = 8, 10, 12, 14, 16, 18$ in Ref. [34]. The simulation is performed by using the forward-backward gate algorithm through optimizing Eq. (7a). The details of architectures can be found in Appendix F. It can be seen that the

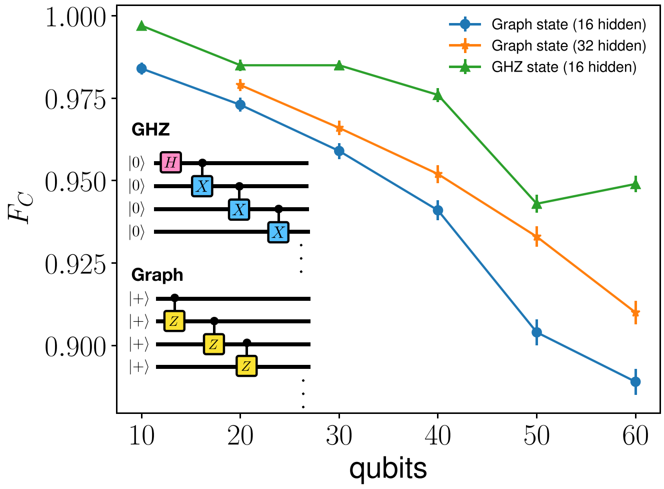

FIG. 5. Classical fidelity F_c between the final probability distribution of the Transformer versus the exact POVM measurements of the postcircuit quantum states as a function of the total number of qubits for circuits (see insets) generating the GHZ state and linear graph states. The Transformers have $d_{\text{model}} = 16$ and 32 and $n_{ed} = 1$.

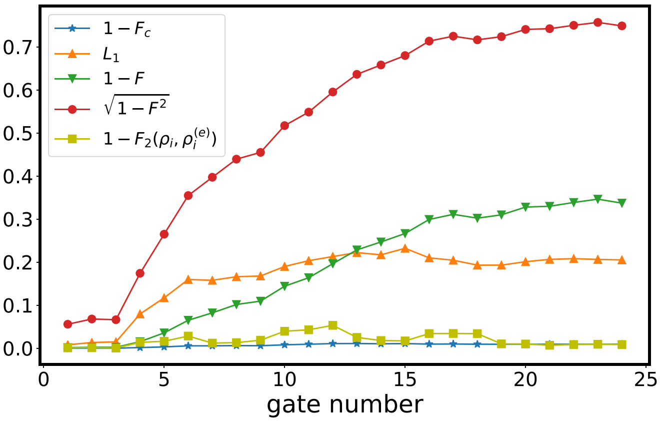

FIG. 6. Different infidelity measurements of a VQE circuit for the preparation of a six-qubit ground state of the TFIM as a function of the gate number.

classical fidelity drops roughly linearly and the L_1 difference increases roughly linearly as the gate number increases (see Fig. 7).

VI. CONCLUSIONS

We have introduced an approach for the classical simulation of quantum circuits using probabilistic models and validate it on a number of different circuits. This is done by using a POVM formalism which maps states to probability distributions and gates to quasistochastic matrices. To represent the probability distribution over this POVM basis, we use a Transformer, a powerful machine learning architecture heavily used in natural language processing. We develop an efficient sampling scheme which updates the Transformer after each gate application within the quantum circuit. This sampling scheme works well out to a large number of qubits; in this work we demonstrate simulations up to 60 qubits and empirically see that the accuracy of the simulation drops roughly linearly with the number of qubits at a fixed hidden dimension of the Transformer architecture. We observe that increasing the hidden dimension of the model improves our results for the circuits we considered. Although this observation suggests that our approach is scalable, a detailed study of the representational power of the Transformer and its relation to optimization over a wider class of circuits and Transformer hyperparameters is required to properly establish the scalability and applicability of our approach.

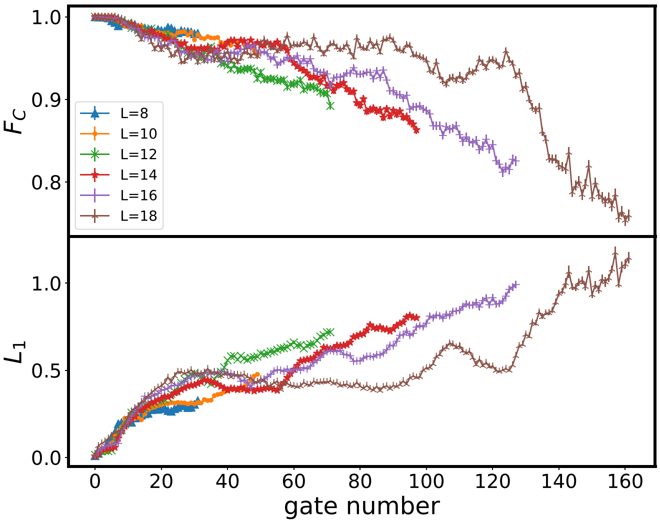

FIG. 7. Classical fidelity and L_1 difference of VQE circuits for the preparation of ground states of the TFIM as a function of the gate number for system size $L = 8, 10, 12, 14, 16, 18$.

Optimizing the Transformer after each gate is a critical step in our algorithm. While already reasonably efficient, there are various ways this optimization might be further improved. For example, our current simulations allow probabilistic representations which do not map to physical quantum states (i.e., outside of the Qplex). We anticipate that constraining the optimization to the physically relevant subspace would improve the quality of the simulations and their broad applicability. Additionally, further strategies from machine learning may be applicable; a common training strategy in natural language processing has been to simultaneously train multiple models selecting the best one at each step and this technique may improve the accuracy in our quantum circuit simulations.

The choice of the POVM basis directly affects the structure of the underlying probability distributions describing the quantum states as well as the efficiency of their simulation. Here we chose a simple IC-POVM basis which is single-qubit factorable, which means that all of the entanglement and complexity associated with the quantum state can be traced back to $\boldsymbol{P}$ and not the POVM basis. Practically, the factorized IC-POVM representation ensures local unitaries and quantum channels map to local quasistochastic matrices allowing for the design of practical algorithms. A common alternative POVM basis, the SIC-POVM [4–7,36] has an elegant formalism but is more difficult to work with algorithmically since SIC-POVM bases are not known to exist for large systems [37] and do not map local unitaries to local matrices. It is nonetheless an interesting research question whether these more complicated bases can be useful in the context of numerical simulations. Indeed, POVM is related to the Wigner-function quasiprobability representation and it will also be worth further investigating their relation.

While in this work we have stored the probability distribution using a Transformer, there are other options for storing this probability distribution including other machine-learning architectures and tensor networks. In fact, it is not even necessary to explicitly store the representation at all; instead, in the spirit of quantum Monte Carlo, it could be sampled stochastically. While such a simulation will generically have a sign problem, there may be preferred basis choices for the POVM which minimize that effect for a particular set of quantum circuits.

In general, the classical simulation of quantum circuits is known to be difficult [38]. Nonetheless, in the era of noisy intermediate-scale quantum technology it is important to be able to benchmark machines which have qubit sizes that are outside the limits of what can be simulated exactly on classical computers to validate and test quantum computers and algorithms. Moreover, the ability to simulate ever larger and more difficult circuits helps better delineate the boundary between classical and quantum computation. The number of approaches for simulating quantum circuits is small and our approach introduces an alternative to the standard approach of simulating the quantum state either explicitly [9–15,39], or stochastically [40,41]. We anticipate advantages with respect to established algorithms enabled by the ability of Transformers to model long-range correlations [16], the autoregressive nature of the model, as well as the nature of the self-attention mechanism, which allows a high degree of parallelization of most of the computations required in our approach. Additionally, extensions of the model which encode information about the spatial structure of the problem (e.g., two-dimensional Transformers [42]) can be easily defined while retaining all the computational and modeling advantages of the Transformers used in this work.

Beyond the simulation of circuits, our POVM approach can be naturally extended to various problems in quantum many-body systems, such as the simulation of real-time dynamics of closed and open quantum systems (see Appendices A and B). Thus our work opens up possibilities for combining the POVM formalism with different numerical methods, ranging from quantum Monte Carlo to machine learning to tensor networks, in an effort to better classically simulate quantum many-body systems.

ACKNOWLEDGMENTS

We would like to thank Martin Ganahl, Jimmy Ba, Amirmassoud Farahmand, Andrew Tan, Jianqiao Zhao, Emily Tyhurst, G. Torlai, R. Melko, and Lei Wang for useful discussions. We also thank the Kavli Institute for Theoretical Physics (KITP) in Santa Barbara and the program “Machine Learning for Quantum Many-Body Physics.” This research was supported in part by the National Science Foundation under Grant No. NSF PHY-1748958. This work utilizes resources supported by the National Science Foundation’s Major Research Instrumentation program, Grant No. 1725729, as well as the University of Illinois at Urbana-Champaign. J.C. acknowledges support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Shared Hierarchical Academic Research Computing Network (SHARCNET), Compute Canada, and the Canada CIFAR AI chair program. B.K.C. acknowledges support from the Department of Energy Grant No. DOE de-sc0020165. L.A. acknowledges financial support from the Brazilian agencies CNPq (PQ Grant No. 311416/2015-2 and INCT-IQ), FAPERJ (JCN E-26/202.701/2018), CAPES (PROCAD2013), and the Serrapilheira Institute (Grant No. Serra-1709-17173). Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Research, Innovation and Science.

APPENDIX A: QUANTUM CHANNELS

A complete description of the evolution of closed and open quantum systems is fundamental to the understanding and manipulation of quantum information devices. In contrast to closed quantum systems, the evolution of open quantum systems is not unitary. Instead, the evolution of the density operator of an open quantum system is described by the action of a quantum operation or quantum channel, which is specified by a completely positive trace-preserving (CPTP) map between spaces of operators [8]. A commonly used representation of CPTP maps is the Kraus or operator-sum representation [43] where a CPTP map $\mathcal{E}$ acts on a quantum state ϱ as

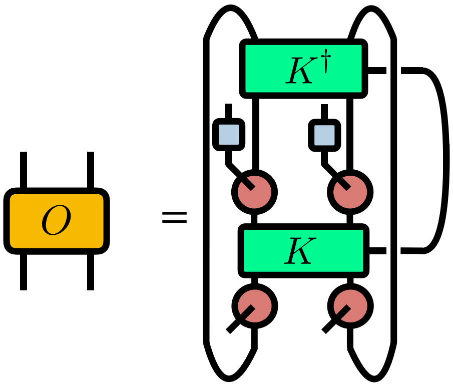

FIG. 8. Tensor network representation of $O_{\boldsymbol{a}''\boldsymbol{a}'}$ corresponding to a quantum channel that acts on two qubits. The green tensors represent the Kraus operators, which specify the quantum channel and are understood as a rank-3 array.

$$\mathcal{E}(\varrho) = \sum_{\alpha}^{D} K^{(\alpha)} \varrho K^{(\alpha)\dagger}, \tag{A1}$$

where $\sum_{\alpha}^{D} K^{(\alpha)} K^{(\alpha)\dagger} = \mathbb{1}$. The set of matrices $\{K^{(\alpha)}, \alpha = 1, \dots, D\}$ act on the Hilbert space of the qubits and can be thought of as an array with three indices. The maximum value of $D = 4^N$, and the minimum is $D = 1$, which corresponds to a unitary transformation.

Similar to the unitary evolution, if the initial quantum state ϱ is prescribed in terms of the outcome statistics of an IC-POVM $\boldsymbol{P}$, we can track its evolution under a CPTP map directly in its probabilistic representation:

$$P_{\mathcal{E}}(\boldsymbol{a}'') = \sum_{\alpha}^{D} \mathrm{Tr}[K^{(\alpha)} \varrho K^{(\alpha)\dagger} M^{(\boldsymbol{a}'')}] = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}'), \tag{A2}$$

where

$$O_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a},\alpha} \mathrm{Tr}[K^{(\alpha)} M^{(\boldsymbol{a})} K^{(\alpha)\dagger} M^{(\boldsymbol{a}'')}]\, T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} \tag{A3}$$

is a quasistochastic matrix since, as in the unitary case, the values in each column add up to 1 but its entries can be positive or negative [5,6,19,20]. In Ref. [7], a similar formulation specific to SIC-POVM has also been proposed to describe quantum channel evolution and master equations in open systems.

If a quantum channel acts nontrivially on only k qubits, it implies that the quasistochastic matrix $O_{\boldsymbol{a}''\boldsymbol{a}'}$ acts also only on k qubits. The relation between the quasistochastic gates and the local quantum channel in Eq. (A3) is graphically depicted in Fig. 8 using tensor diagrams.

APPENDIX B: LIOUVILLE–VON NEUMANN EQUATION IN POVM FORMULATION

In Eq. (3) we have discussed how unitary evolution on a quantum state in the traditional density matrix formulation translates into the factorized IC-POVM formulation used in our study. Accordingly, the unitary dynamics induced by Hamiltonian $\mathcal{H}$ acting on the system during an infinitesimal time Δt, i.e., $U(t) = e^{-i\Delta t\mathcal{H}}$, implies an equation of motion for the measurement statistics

$$i\frac{\partial P(\boldsymbol{a}'', t)}{\partial t} = \sum_{\boldsymbol{a},\boldsymbol{a}'} \mathrm{Tr}([\mathcal{H}, M^{(\boldsymbol{a})}]M^{(\boldsymbol{a}'')})T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}P(\boldsymbol{a}', t). \tag{B1}$$

This is equivalent to the Liouville–von Neumann equation $i\frac{\partial\rho}{\partial t} = [\mathcal{H}, \rho]$.

A solution to Eq. (B1) for a time-independent Hamiltonian is given by $\boldsymbol{P}(t) = e^{-iAt}\boldsymbol{P}(0)$, where the matrix elements $A_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}[\mathrm{Tr}([\mathcal{H}, M^{(\boldsymbol{a})}]M^{(\boldsymbol{a}'')})]$.

APPENDIX C: LINDBLAD EQUATION IN POVM FORMULATION

Applicable to open quantum systems, an infinitesimal Markovian but nonunitary evolution leads to the equivalent of the Linblad equation

$$i\frac{\partial P(\boldsymbol{a}, t)}{\partial t} = \sum_{\boldsymbol{a}} A_{\boldsymbol{a}'',\boldsymbol{a}'}P(\boldsymbol{a}, t), \tag{C1}$$

where the matrix elements are augmented to

$$\begin{aligned} A_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\Bigg\{ &\mathrm{Tr}([\mathcal{H}, M^{(\boldsymbol{a})}]M^{(\boldsymbol{a}'')}) \\ &+ \sum_k \Bigg[-\frac{i}{2}\mathrm{Tr}(\{L_k^\dagger L_k, M^{(\boldsymbol{a})}\}M^{(\boldsymbol{a}'')}) \\ &+ i\,\mathrm{Tr}(L_k M^{(\boldsymbol{a})} L_k^\dagger M^{(\boldsymbol{a}'')})\Bigg]\Bigg\}. \end{aligned} \tag{C2}$$

Here, the operators L_k are called Lindblad operators or quantum jump operators. Like the Liouville–von Neumann equation, Eq.(C2) has a solution for a time-independent Hamiltonian given by $\boldsymbol{P}(t) = e^{-iAt}\boldsymbol{P}(0)$.

APPENDIX D: MEASUREMENTS

Although the probabilistic representation of the quantum state in Eq. (2) already provides the measurement statistics of the factorized POVM $M^{(\boldsymbol{a})}$, the statistics of other POVMs $\Pi^{(\boldsymbol{b})}$, e.g., a POVM describing standard measurements in the computational basis and other experimentally relevant operators, are related to $M^{(\boldsymbol{a})}$ via the Born rule:

$$\begin{aligned} P_\Pi(\boldsymbol{b}) &= \sum_{\boldsymbol{a},\boldsymbol{a}'} P(\boldsymbol{a}')\, T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\, \mathrm{Tr}[M^{(\boldsymbol{a})}\Pi^{(\boldsymbol{b})}] \\ &= \sum_{\boldsymbol{a}'} q(\boldsymbol{b}|\boldsymbol{a}')P(\boldsymbol{a}'), \end{aligned}$$

where $q(\boldsymbol{b}|\boldsymbol{a}') = \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\, \mathrm{Tr}[M^{(\boldsymbol{a})}\Pi^{(\boldsymbol{b})}]$ can be characterized as a quasiconditional probability distribution since its entries can either be positive or negative but its trace over $\boldsymbol{b}$ is the identity $\mathbb{1}_{\boldsymbol{a}'}$. Due to its evocative resemblance with the law of total probability, the relation between measurement statistics $P_\Pi(\boldsymbol{b})$ and $P(\boldsymbol{a})$ is often called the quantum law of total probability in quantum Bayesianism [44].

APPENDIX E: QPLEX AND POSITIVITY OF QUANTUM STATES

The traditional quantum theory can be viewed as a noncommutative generalization of probability theory where quantum states are specified by Hermitian, positive semidefinite trace one matrices. However, quantum states can also be specified through probability distributions corresponding to the statistics of the outcome of an informationally complete physical measurement. From this viewpoint, quantum theory is not necessarily a generalization of probability theory; instead, it can be seen as augmenting probability theory with further rules for dynamics and measurements on quantum systems [33]. When we represent the probabilities of a IC-POVM as points in the corresponding probability simplex Δ_{4^N}, these probabilities are not arbitrary, since not any point of the simplex Δ can represent a quantum state, only a subset of the simplex. For a symmetric IC-POVM [45], this subset is referred to as the Qplex [33]. Even though the IC-POVM used in our work is not symmetric, we will still refer to the subset of distributions with a corresponding quantum state in Eq. (2) as a Qplex. The space of all possible states of a given quantum system and the corresponding Qplex are schematically represented in Figs. 9(a) and 9(b).

A small quantum computation is also depicted schematically in Figs. 9(a) and 9(b). This computation starts with a simple pure product state ϱ_0 followed by the application of three unitary matrices which take the state from ϱ_0 to ϱ_3. These computations occur at the boundary separating valid quantum states from other operators; such boundary includes all the pure states. Correspondingly, since the relation between the space of quantum states and the Qplex is linear, quantum computations in the probabilistic language take place at the interior of the Qplex, as illustrated in Fig. 9(b).

While these observations do not have any major conceptual implication for the physical realization of quantum

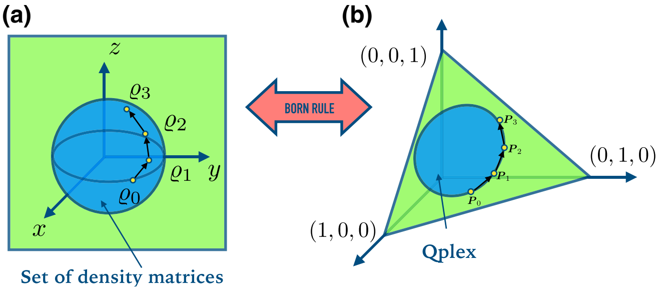

FIG. 9. Geometry of quantum states. (a) Schematic representation of the subset of density matrices (blue sphere). For one qubit, this set corresponds to the Bloch sphere. (b) Schematic representation of the probability simplex Δ_{m^N}, which represents the set of all possible categorical probability distributions with m^N outcomes. A subset of these distributions termed Qplex (blue oval) is isomorphic to the usual space of quantum states in (a).

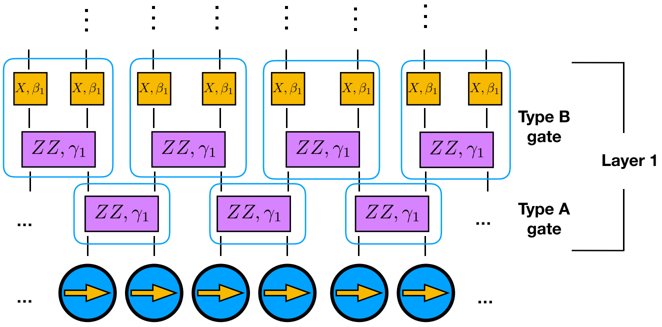

FIG. 10. Variational circuit preparation for TFIM. Only the first of the four layers in our calculation is shown. The gates encircled in rounded blue squares are combined and subsequently transformed into quasistochastic gates for the probabilistic simulation of the quantum circuit.

computations, this geometric interpretation can help us clarify some aspects we observe in the results from our simulation strategy. The most important aspect is the fact that the probabilistic model in our study, in general, lives in a standard simplex Δ_{m^N} and is not constrained to the subset of valid "quantum" distributions. Even though the update rule in Eq. (3) should produce distributions that live on the Qplex, since we use an approximate update, it is possible that the model may temporarily leave the Qplex. This is observed in Figs. 4(d) and 4(h) where we observe values of quantum fidelity higher than 1, which means that during the training process, the Transformer induces matrices in Eq. (2) that are not valid quantum states. Note that the fidelity in Figs. 4(d) and 4(h) eventually converges and stays at values very close to 1 and that the oscillations above 1 disappear, suggesting that the state is getting closer and closer to the target, valid quantum state.

APPENDIX F: VARIATIONAL CIRCUIT FOR TFIM

We use the variational circuit depicted in Fig. 10 for the six-qubit TFIM preparation [34]. The parameters for gamma and beta are taken alternatively from the following sequence describing a circuit with four layers, (0.2496, 0.6845, 0.4808, 0.6559, 0.5260, 0.6048, 0.4503, 0.3180). For $L = 8, 10, 12, 14, 16, 18$, the circuit parameters are the same as Ref. [34]. Note that in our simulations, we do not directly transform the original gates into quasistochastic gates. To save computational resources, we combine the quantum gates encircled in rounded blue squares, after which we transform them into quasistochastic matrices.

For Transformers used in the VQE circuits simulation, the encoder block does not include LayerNorm. For $L = 6, 8, 10, 12$, $d_{\text{model}} = 16$ and $n_{ed} = 1$. For $L = 14, 16, 18$, $d_{\text{model}} = 32$ and $n_{ed} = 1$.

Using this circuit, we have also computed the $\sigma_i^z \sigma_j^z$ correlation of the exact variational circuit in Fig. 10 and the POVM trained circuit, which are compared in Fig. 11. Even though there is a nice polynomial bound between quantum fidelity and classical fidelity under the SIC-POVM formulation [7], it is not known in general the bound between a non SIC-POVM (like we use) and the classical fidelity. We therefore numerically plot the relation between quantum fidelity and classical fidelity of the $L = 6$ VQE simulation in Fig. 12.

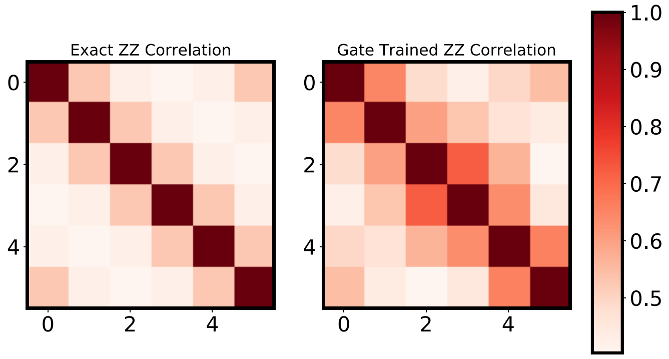

FIG. 11. Comparison between $\sigma_i^z \sigma_j^z$ correlation from exact quantum circuit state and the gate training state.

APPENDIX G: OPTIMIZATION DETAILS

The models are optimized using Adam optimizer [21] in PyTorch [46] with an initial learning rate of 0.01. The weights and biases are initialized using PyTorch's [46] default initialization except for the last layer. We use single-precision (32-bit) floating-point representation for real numbers. The batch size of each training is around 10^4. Most models converge in less than 200 steps. VQE circuit simulations can be completed within a few hours for small system size L and up to 1 day for $L = 18$ with one V100 GPU. GHZ circuits and graph state circuits simulation up to 60 qubits can be completed within 1 or 2 days with four V100 GPUs in parallel.

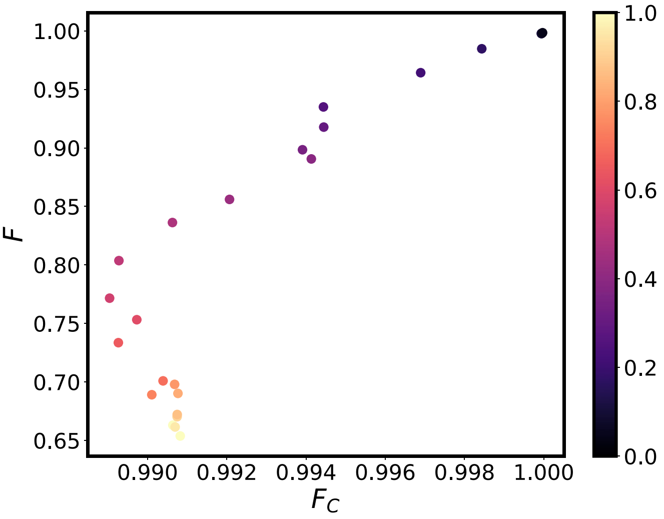

FIG. 12. Correlation between classical fidelity and quantum fidelity. The darker color corresponds to gate that is applied earlier.

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

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., Red Hook, 2017), pp. 5998–6008.

[3] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Nat. Mach. Intell. **1**, 155 (2019).

[4] C. A. Fuchs and R. Schack, Found. Phys. **41**, 345 (2011).

[5] D. Chruściński, V. I. Man'ko, G. Marmo, and F. Ventriglia, Phys. Scr. **90**, 115202 (2015).

[6] J. van de Wetering, Electron. Proc. Theor. Comput. Sci. **266**, 179 (2018)

[7] E. O. Kiktenko, A. O. Malyshev, A. S. Mastiukova, V. I. Man'ko, A. K. Fedorov, and D. Chruściński, Phys. Rev. A **101**, 052320 (2020).

[8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th anniversary ed. (Cambridge University Press, New York, 2011).

[9] K. De Raedt, K. Michielsen, H. De Raedt, B. Trieu, G. Arnold, M. Richter, T. Lippert, H. Watanabe, and N. Ito, Comput. Phys. Commun. **176**, 121 (2007).

[10] I. Markov and Y. Shi, SIAM J. Comput. **38**, 963 (2008).

[11] M. Smelyanskiy, N. P. D. Sawaya, and A. Aspuru-Guzik, arXiv:1601.07195.

[12] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. Solomonik, E. W. Draeger, E. T. Holland, and R. Wisnieff, arXiv:1710.05867.

[13] J. Chen, F. Zhang, C. Huang, M. Newman, and Y. Shi, arXiv:1805.01450.

[14] R. Li, B. Wu, M. Ying, X. Sun, and G. Yang, arXiv:1804.04797.

[15] B. Jónsson, B. Bauer, and G. Carleo, arXiv:1808.05232.

[16] H. Shen, arXiv:1905.04271.

[17] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, Phys. Rev. Lett. **122**, 065301 (2019).

[18] R. Penrose, Applications of negative dimensional tensors, in *Combinatorial Mathematics and its Applications*, edited by D. Welsh (Academic Press, New York, 1971), pp. 221–244.

[19] B. Ćurgus and R. I. Jewett, arXiv:0709.0309.

[20] B. Ćurgus and R. I. Jewett, Am. Math. Mon. **122**, 36 (2015).

[21] D. P. Kingma and J. Ba, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (ICLR, San Diego, CA, USA, 2015).

[22] R. Assaraf and M. Caffarel, Phys. Rev. Lett. **83**, 4682 (1999).

[23] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 770–778.

[24] J. L. Ba, J. R. Kiros, and G. E. Hinton, arXiv:1607.06450.

[25] I. A. Luchnikov, A. Ryzhov, P.-J. Stas, S. N. Filippov, and H. Ouerdane, Entropy **21**, 1091 (2019).

[26] F. Becca and S. Sorella, *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press, Cambridge, UK, 2017).

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016), http://www.deeplearningbook.org.

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

[29] S. Cree and J. Sikora, arXiv:2006.06918.

[30] R. Adamczak, J. Phys. A: Math. Theor. **50**, 105302 (2017).

[31] S. Deffner, Heliyon **3**, e00444 (2017).

[32] H. Huang, H. Situ, and S. Zheng, Chin. Phys. Lett. **38**, 040303 (2021).

[33] M. Appleby, C. A. Fuchs, B. C. Stacey, and H. Zhu, Eur. Phys. J. D **71**, 197 (2017).

[34] W. W. Ho and T. H. Hsieh, SciPost Phys. **6**, 29 (2019).

[35] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, Nat. Commun. **5**, 4213 (2014).

[36] C. Ferrie, Rep. Prog. Phys. **74**, 116001 (2011).

[37] C. A. Fuchs, M. C. Hoang, and B. C. Stacey, Axioms **6**, 21 (2017).

[38] A. Bouland, B. Fefferman, C. Nirkhe, and U. Vazirani, Nat. Phys. **15**, 159 (2019).

[39] I. L. Markov, A. Fatima, S. V. Isakov, and S. Boixo, arXiv:1807.10749.

[40] N. J. Cerf and S. E. Koonin, Math. Comput. Simul. **47**, 143 (1998).

[41] A. Mari and J. Eisert, Phys. Rev. Lett. **109**, 230503 (2012).

[42] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, A. Ku, and D. Tran, in *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 80, edited by J. Dy and A. Krause (PMLR, 2018), pp. 4055–4064.

[43] K. Kraus, in *States, Effects, and Operations: Fundamental Notions of Quantum Theory*, edited by A. Böhm, J. D. Dollard, and W. H. Wootters, Lecture Notes in Physics (Springer-Verlag, Berlin/Heidelberg, 1983).

[44] C. A. Fuchs and R. Schack, Rev. Mod. Phys. **85**, 1693 (2013).

[45] J. M. Renes, R. Blume-Kohout, A. J. Scott, and C. M. Caves, J. Math. Phys. **45**, 2171 (2004).

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates Inc., Red Hook, 2019), pp. 8024–8035.