© 2021 by Di Luo. All rights reserved.

QUANTUM MANY-BODY PHYSICS WITH CLASSICAL AND QUANTUM ALGORITHMS

BY

DI LUO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Physics
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Aida X El-Khadra, Chair
Professor Bryan K. Clark, Director of Research
Professor Brian L. DeMarco
Assistant Professor Zhizhen Zhao

Abstract

The thesis is composed of three main topics involving research in condensed matter physics, high energy physics and quantum information science. The first component is the development of neural network quantum states for studying ground states and real time dynamics for condensed matter and high energy physics. Relevant work is presented in Chapter 2, 3, 4, 5. The second component is the development of a probabilistic formulation of quantum mechanics for the simulation of quantum circuit and open quantum system. Relevant work is presented in Chapter 6, 7. The third component is the development of quantum information science, which includes quantum dynamics, quantum simulation, quantum control and quantum measurement. Relevant work is presented in Chapter 8, 9, 10, 11.

To My Family.

Acknowledgments

I am deeply grateful to my advisor Prof. Bryan Clark. I appreciate Bryan wholeheartedly for the freedom to choose research topics and his sincere career support over the years. Bryan's enthusiasm for physics and computation, curiosity of knowledge and professional working attitude have greatly inspired my pursuit of science.

Meanwhile, I would like to thank Prof. Juan Carrasquilla for his support and encouragement of exploring new knowledge. I would like to thank Prof. Giuseppe Carleo and James Stokes for their support as well as fruitful discussions on physics and algorithms. I would like to thank Nicholas Rubin for his support and valuable advice in quantum information science research. I am also in debt to and learn a lot from Prof. Vera Mikyoung Hur, Prof. Zhizhen Zhao, Prof. Aida El-Khadra, Prof. Bryce Gadway, Prof. Brian DeMarco, Prof. Patrick Draper, Prof. Pinshane Huang, Prof. Graham Kells, Prof. Yang Zhang, Miles Stoudenmire, Prof. Lei Wang, Prof. Anthony Leggett, Prof. Lin Lin, Prof. Garnet Chan and Prof. Nigel Goldenfeld.

It is my fortune to make many good friends during my graduate school. I am grateful for research collaborations with Xiongjie Yu, Jiayu Shen, Zhuo Chen, Ryan Levy, Abid Khan, Chiahao Lee, Michael Highman, Chuqiao Shi, Kaiwen Hu, Jiangran Wang and Luuk Coopmans. I would like to thank many friends in the Bryan's group, the physics department, the University as well as all my friends outside the University for their kindness and help. It is also appreciated to have the support from the U.S. Department of Energy under Award No. DE-SC0019213, as well as the care from the physics department with Wendy Wimmer and Prof. Lance Cooper.

Finally, I would like to express my gratitude to my parents, my grandparents and other family members for their encouragement, support and love.

Table of Contents

Chapter 1 Overview **1**
1.1 Machine Learning and Physics 1
1.1.1 Neural Network Representations 1
1.1.2 Variational Principle 3
1.2 Quantum Information Science 4
1.2.1 Positive Operator-valued Measurement 4
1.2.2 Shadow Quantum State Tomography 5
1.2.3 Quantum Circuit 6
1.3 Gauge Theories 9
1.3.1 $\mathbb{Z}_2$ Toric Code 9
1.3.2 Quantum Link Model 10

Chapter 2 Backflow Transformations via Neural Networks for Quantum Many-Body Wave-Functions **12**
2.1 Introduction 12
2.2 Background 14
2.3 Neural Network Backflow 15
2.4 Explicit Construction of Standard Backflow 17
2.5 Results 18
2.6 Conclusion 20
2.7 Supplementary Materials 21
2.7.1 Numerical Results on Neural Network Backflow of Standard Form 21
2.7.2 Optimization Scheme 23
2.7.3 Backward Propagation of Neural Network Backflow 24
2.7.4 Neural Network Backflow on Various Systems 27

Chapter 3 Gauge equivariant neural networks for quantum lattice gauge theories **31**
3.1 Introduction 31
3.2 $\mathbb{Z}_2$ Gauge Theory 33
3.3 Gauge Equivariant Neural Networks 34
3.4 Gauge Equivariant Layer 35
3.5 Numerical Experiments 36
3.6 Discussion and Future Directions 37
3.7 Supplemental Materials 37
3.7.1 Detailed Description of the $\mathbb{Z}_2$ Gauge Theory Hilbert Space 37
3.7.2 Exact Representation of $\mathbb{Z}_2$ Toric Code Ground States 37
3.7.3 Gauge Equivariant Construction for $\mathbb{Z}_d$ Toric Code 38
3.7.4 Gauge Equivariant Construction for 3D Toric Code and X-cube Fracton Model 40
3.7.5 Gauge Equivariant Construction for Kitaev $D(G)$ Models 41
3.7.6 Neural Network Architecture and Numerical Details 42
3.7.7 Observables and Quantities across Phase Transition 43

Chapter 4 Gauge Invariant Autoregressive Neural Networks for Quantum Lattice Models 53
4.1 Introduction . . . 53
4.2 Gauge Invariant Construction . . . 55
4.3 Optimization Algorithms . . . 57
4.4 Applications in Quantum Lattice Models . . . 58
4.4.1 U(1) Quantum Link Model . . . 58
4.4.2 2D $\mathbb{Z}_N$ Gauge Theory . . . 60
4.4.3 3D Toric Code and Fracton Model . . . 63
4.4.4 $SU(2)_k$ Anyonic Chain and SU(2) Symmetry . . . 64
4.5 Conclusion . . . 65
4.6 Supplementary Materials . . . 67
4.6.1 Kitaev's $D(G)$ Model and Exact Representation of Ground State . . . 67
4.6.2 (1+1)D $\mathbb{Z}_N$ Lattice QED Model . . . 67
4.6.3 SU(2) Invariant Autoregressive Neural Network for Heisenberg Model . . . 68
4.6.4 Neural Network Architecture . . . 68
4.6.5 Derivation of Stochastic Gradients for Variational and Dynamics Optimization . . . 71

Chapter 5 Spacetime Neural Network for High Dimensional Quantum Dynamics . . . 88
5.1 Introduction . . . 88
5.2 First Order Optimization and Time-dependent Variational Principle . . . 90
5.3 Second Order Optimization with Spacetime Formulation . . . 90
5.4 Architecture of Spacetime Neural Network . . . 92
5.5 Numerical Experiments . . . 93
5.6 Discussions and Conclusion . . . 95

Chapter 6 Probabilistic Simulation of Quantum Circuits with the Transformer . . . 96
6.1 Introduction . . . 96
6.2 Formalism . . . 98
6.3 Optimization Algorithms . . . 101
6.4 Transformer Architecture . . . 104
6.5 Applications . . . 107
6.5.1 GHZ State and Linear Graph State Preparation . . . 107
6.5.2 Simulation of VQE Circuits . . . 110
6.6 Conclusions . . . 112
6.7 Supplementary Materials . . . 114
6.7.1 Quantum Channels . . . 114
6.7.2 Liouville-Von Neumann Equation in POVM Formulation . . . 115
6.7.3 Lindblad Equation in POVM Formulation . . . 115
6.7.4 Measurements . . . 116
6.7.5 Qplex and Positivity of Quantum States . . . 116
6.7.6 Variational Circuit for TFIM . . . 118
6.7.7 Optimization Details . . . 120

Chapter 7 Autoregressive Neural Network for Simulating Open Quantum Systems via a Probabilistic Formulation . . . 121
7.1 Introduction . . . 121
7.2 Lindblad Equation as a Probability Equation . . . 123
7.3 Autoregressive Models and String States . . . 124
7.4 Optimization and Results . . . 125
7.5 Conclusion . . . 128
7.6 Supplementary Materials . . . 128
7.6.1 Lindblad Equation in POVM Formalism . . . 128
7.6.2 Tetrahedral and 4-Pauli POVM . . . 129
7.6.3 Convergence of Loss . . . 130

7.6.4 Results for Heisenberg Model with Larger Energy Scale . . . 130
7.6.5 Improve Performance by Combining Probabilities . . . 131
7.6.6 Neural Network Initialization . . . 131
7.6.7 Explanation of the Dynamics and Variational Cost Functions in Detail . . . 131
7.6.8 Exact Sampling from Conditional Probability Distributions . . . 133
7.6.9 Analysis of the Gradients of Dynamics and Variational Cost Functions . . . 134
7.6.10 Additional Benchmarks with Classical and Quantum Algorithms . . . 135

Chapter 8 Beyond many-body localized states in a spin-disordered Hubbard model with pseudo-spin symmetry . . . 144
8.1 Introduction . . . 144
8.1.1 Overview of Results . . . 145
8.2 Introduction to Pseudo-spin Algebra . . . 147
8.2.1 Pseudo-spin $SU(2)$ symmetry . . . 147
8.2.2 Particle-hole Transformation . . . 148
8.3 Entanglement Entropy of Eta-pairing States . . . 148
8.3.1 Maximally Polarized Sector . . . 149
8.3.2 Reference States with Only Singlons . . . 152
8.4 Numerical Evaluation of Entanglement . . . 153
8.5 Time Evolution after Quantum Quench . . . 157
8.6 Conclusion . . . 161
8.7 Supplementary Materials . . . 162
8.7.1 Reference States . . . 162
8.7.2 Normalization Factor of Eta-pairing States Built from Many Particle Reference State 163
8.7.3 Additional Numerical Evidence on Eta-pairing State Entanglement Entropy . . . 164
8.7.4 Bimodal Distribution of Reference States Entanglement Entropy . . . 166
8.7.5 Staggered Magnetization . . . 167
8.7.6 Disorder Dependence of Entanglement . . . 168
8.7.7 Disorder Dependence of Dynamics . . . 169

Chapter 9 Framework for simulating gauge theories with dipolar spin systems . . . 172
9.1 Introduction . . . 172
9.2 Quantum Link models . . . 175
9.3 Dipolar Molecules . . . 175
9.4 Effective Hamiltonian . . . 176
9.5 Realization and Tests of $S = 1/2$ QLM . . . 177
9.6 Realization and Tests of $S = 1$ QLM . . . 179
9.7 Conclusions and Outlook . . . 181

Chapter 10 Protocol Discovery for the Quantum Control of Majoranas by Differentiable Programming and Natural Evolution Strategies . . . 186
10.1 Summary . . . 186

Chapter 11 Shadow Quantum Process Tomography on Near-term Quantum Computers 189
11.1 Summary . . . 189

Chapter 12 References . . . 191

Chapter 1

Overview

This chapter provides an overview of the underlying concepts and foundations for the later chapters. Sec. 1.1 discusses the recent progress in machine learning and physics. Sec. 1.2 reviews basic concepts in quantum information science. Sec. 1.3 introduces gauge theories.

1.1 Machine Learning and Physics

Historically, machine learning has a deep connection to statistical physics, which both study high dimensional probability distribution. Inspired by recent success of machine learning in computer vision, natural language processing and reinforcement learning [1, 2, 3], physicists have started to explore applications of machine learning in physics in the past few years.

One attempt is in the line of phase classification and big data processing using machine learning [4, 5]. This is related to the supervised learning approach, where one can produce observable data in different phases numerically or experimentally which is then used for training. After the training, the machine can be employed to predict phases or properties for new data. Another attempt is to use machine learning for scientific calculation and simulations. For example, this direction includes research in chemical physics, condensed matter physics and high energy physics simulations [6, 7, 8, 9].

1.1.1 Neural Network Representations

The artificial neural network is historically inspired by the study of the biological neural network with motivation to model the complex biological phenomena. A simple diagram of a neural network is given in Fig. 1.1. The circle denotes neurons while the lines are the connections between the neurons. The first layer of neurons are called the input layer while the last layer of neurons are called the output layer. The rest of the neuron layers are referred as hidden layers. The number of neuron in a layer is called the width of the layer and the number of the total layers is called the depth of the layer.

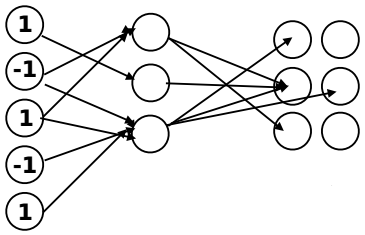


Figure 1.1: A simple diagram of an artificial neural network. The leftmost layer is the input layer and the rightmost layer is the output layer. The layers between the input and the output layer are hidden layers.

Mathematically, an artificial neural network in general consists of affine transformation and non-linearity. There are different architectures for artificial neural networks. Common examples include the fully connected neural network (FCN) [10], the convolution neural network (CNN) [10] and the restricted Boltzmann machine (RBM) [11].

A fully connected neural network has a dense affine transformation between hidden layers such that

$$\vec{h}^{j+1} = W\vec{h}^{j} + \vec{b} \tag{1.1}$$

where h^{j+1} and h^{j} are the $j+1$-th and j-th hidden layers and W is a dense matrix.

A convolution neural network has encoded translation invariance through the convolution matrix. The key is to replace the dense affine transformation in FCN with the convolution operation as follows

$$(W * x)[n] = \sum_{m=-K}^{K} W[n-m]x[m] \tag{1.2}$$

where K is the kernel size of the convolution.

A restricted Boltzmann machine is given by the following parametrization.

$$f(\vec{\sigma}) = e^{\sum_j a_j \sigma_j} \prod 2\cosh[b_i + \sum_j W_{ij}\sigma_j] \tag{1.3}$$

where a_j, b_i, W_{ij} are parameters for the RBM.

We further review other commonly used activation functions as follow

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.4}$$

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{otherwise} \end{cases} \tag{1.5}$$

for some parameter α.

$$Leaky\ RELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases} \tag{1.6}$$

for some parameter α.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{1.7}$$

$$Softmax(x) = \frac{e^{x_i}}{\sum e^{x_i}} \tag{1.8}$$

where $x = (x_1, \cdots, x_n)$

One important property of artificial neural network is that it behaves as a universal function approximator. An approximation theorem of neural network is given in the following [12].

Theorem 1.1 For any given function $f \in \mathcal{C}(I_n)$ (continuous in the n-dimensional unit cube $[0,1]^n$) and $\epsilon > 0$, there exists a function $g(x)$ as a one layer neural network such that

$$|g(x) - f(x)| < \epsilon \tag{1.9}$$

for all $x \in I_n$. In particular, $g(x)$ can be constructed in the following form

$$g(x) = \sum_j a_j \sigma(w_j x + b_j) \tag{1.10}$$

where $\sigma(x)$ can be the sigmoid activation function.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1.11}$$

1.1.2 Variational Principle

The variational principle for ground state calculation can be formulated as follows

$$E = min_{\psi_\theta} \frac{\langle \psi_\theta | H \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \tag{1.12}$$

where ψ_θ is a variational ansatz for the ground state with parameters θ. Notice that the Hilbert space grows exponentially with respect to the number of particles so that $\langle \psi_\theta | H \psi_\theta \rangle$ and $\langle \psi_\theta | \psi_\theta \rangle$ could not be computed exactly in general. Instead, one could perform important sampling to evaluate the quantities. The variational principle allows for updating the variational ansatz through gradient descent.

$$\theta \to \theta - \gamma \nabla_\theta E \tag{1.13}$$

where γ is the learning rate and $\nabla_\theta E$ the energy derivative.

A sampling version of the ground state energy derivative can be formulated as follows

$$\nabla_\theta E = \frac{2}{N} \sum_{x \sim |\psi_\theta|^2}^{N} \Re \left\{ \left[E_{\text{loc}}(x) - E_{\text{avg}} \right] \frac{\partial}{\partial \theta} \log \psi_\theta^*(x) \right\}, \tag{1.14}$$

where $E_{\text{loc}} \equiv H\psi_\theta(x)/\psi_\theta(x)$, $E_{\text{avg}} \equiv \sum_{x \sim |\psi_\theta|^2}^{N} E_{\text{loc}}(x)/N$ over the batch of samples. There are different ways to sample from $|\psi_\theta|^2$, such as Markov Chain Monte Carlo, Hamiltonian Monte Carlo and exact sampling.

1.2 Quantum Information Science

1.2.1 Positive Operator-valued Measurement

Positive Operator-valued Measure (POVM) is a measure which values are positive semi-definite operators on Hilbert space. It generalizes the standard projective valued measurement, which measures in certain computational basis $|c\rangle$ and attains $|\langle c|\psi\rangle|^2$ for a given state ψ. Mathematically, for finite dimensional Hilbert space $\mathcal{H}$, POVM is a set of positive semi-definite matrices M_i such that

$$\sum_i M_i = \mathbb{1} \tag{1.15}$$

The probability corrsponding to measuring M_i on a quantum state ρ is given as

$$P(i) = \text{tr}(\rho M_i) \tag{1.16}$$

If a POVM is said to be an informationally complete POVM (IC-POVM) if it is able to span the space of self-adjoint operators $\mathcal{L}(\mathcal{H})$. The followings are some common examples of one qubit IC-POVM.

Tetra POVM

$$M_{(a)} = \frac{1}{4}\left(\mathbb{1} + \boldsymbol{v}_{(a)} \cdot \boldsymbol{\sigma}\right), \tag{1.17}$$

where $\mathbb{I}$ is the identity matrix, $\boldsymbol{\sigma}$ are the Pauli matrices, and $\boldsymbol{v}_{(a)}$ are four unit vectors which form a tetrahedral. In particular, the following is a valid choice.

$$\boldsymbol{v}_{(1)} = \langle 0, 0, 1\rangle, \tag{1.18}$$

$$\boldsymbol{v}_{(2)} = \left\langle \frac{2\sqrt{2}}{3}, 0, -\frac{1}{3}\right\rangle, \tag{1.19}$$

$$\boldsymbol{v}_{(3)} = \left\langle -\frac{\sqrt{2}}{3}, \frac{\sqrt{6}}{3}, -\frac{1}{3}\right\rangle, \tag{1.20}$$

$$\boldsymbol{v}_{(4)} = \left\langle -\frac{\sqrt{2}}{3}, -\frac{\sqrt{6}}{3}, -\frac{1}{3}\right\rangle. \tag{1.21}$$

4-Pauli POVM

$$M_{(1)} = \frac{1}{3}|0\rangle\langle 0|, \tag{1.22}$$

$$M_{(2)} = \frac{1}{3}|+\rangle\langle +|, \tag{1.23}$$

$$M_{(3)} = \frac{1}{3}|r\rangle\langle r|, \tag{1.24}$$

$$M_{(4)} = \mathbb{I} - M_{(1)} - M_{(2)} - M_{(3)}, \tag{1.25}$$

where $|0\rangle$, $|+\rangle$, and $|r\rangle$ are the positive eigenstates of σ_z, σ_x, and σ_y respectively.

The 4-Pauli POVM can be implemented directly in current experimental setup. In practice, one can randomly draw a measurement from Pauli X, Y and Z. If the measurement is Pauli Z and the measurement outcome state is $|0\rangle$, then the we denote the result to be 1. If the measurement is Pauli X and the measurement outcome state is $|+\rangle$, then the we denote the result to be 2. If the measurement is Pauli Y and the measurement outcome state is $|r\rangle$, then the we denote the result to be 3. For all the other measurement outcome states, we denote the result to be 4. There are different ways to construct general many-body POVM. One simple way is to use the tensor product of one-qubit POVM.

1.2.2 Shadow Quantum State Tomography

Shadow quantum state tomography is developed for state reconstruction and attaining information of observables of a quantum state with fewer measurements than standard quantum tomography. In general, one has a set of measurement primitive U, which could be unitary. The measurement primitive U is performed

on a quantum circuit representation of the quantum state followed by computational basis measurement $|b\rangle$. The set of $\{U|b\rangle\}$ are called classical shadow and can be used to retrieve information on the quantum state. An important result for the shadow quantum state tomography has been developed in Ref. [13], which allows for efficient measurement complexity for reconstruction local observables.

Theorem 1.2.1 *Fix a measurement primitive U, a collection $\{O_i\}_{i=1,\dots,M}$ of $2^n \times 2^n$ Hermitian matrices and accuracy parameters $\epsilon, \delta \in [0,1]$. Set $K = 2\log(2M/\delta)$ and $N = \frac{34}{\epsilon^2}\max_i ||O_i - \frac{\mathrm{tr}(O_i)}{2}\mathbb{1}||^2_{shadow}$, where $||.||_{shadow}$ is the shadow norm defined with the inverse channel $\mathcal{M}^{-1}$ induced by the measurement primitive as follows,*

$$||O||_{shadow} = max_\sigma(\mathbb{E}_{U\sim\mathcal{U}} \sum_{b\in\{0,1\}^n} \langle b|U\sigma U^\dagger|b\rangle \langle b|U\mathcal{M}^{-1}(O)U^\dagger|b\rangle^2)^{1/2} \tag{1.26}$$

Then, a collection of NK independent classical shadows allow for accurate predicting all features via median of means prediction $\hat{o_i}$:

$$|\hat{o_i}(N,K) - \mathrm{tr}(O_i\rho)| \leq \epsilon \tag{1.27}$$

for all $1 \leq i \leq M$ *with probability at least* $1-\delta$.

In particular, for random global Clifford measurement primitive, $||O - \frac{\mathrm{tr}(O)}{2}\mathbb{1}||^2_{shadow} \leq 3\mathrm{tr}(O^2)$. It implies that the global Clifford measurement allows for $log(M)$ measurements for M local observables with additive errors. For random Pauli measurement primitive, $||O - \frac{\mathrm{tr}(O)}{2}\mathbb{1}||^2_{shadow} \leq 4^k||O||^2_\infty$ where k is the locality of the operator O and $||.||_\infty$ is the spectral norm. Furthermore, if O is a tensor product of k-qubit observable, $||O - \frac{\mathrm{tr}(O)}{2}\mathbb{1}||^2_{shadow} \leq 3^k||O||^2_\infty$.

1.2.3 Quantum Circuit

Quantum circuit is a basic building block for digital quantum computation. In general, a quantum circuit consist of an initial state and the circuit unitary gate operator U. The initial state can be a product state, such as $|0\rangle^{\otimes n}$, where n is the number of qubits. The circuit unitary gate operator U is usually composed of multiple one-qubit and two-qubit gates that act on different qubit locations. An example of a quantum circuit is shown in Fig. 6.6. We further review several important quantum logic gates that build up general quantum circuits, which include one-qubit, two-qubit and three-qubit quantum gates.

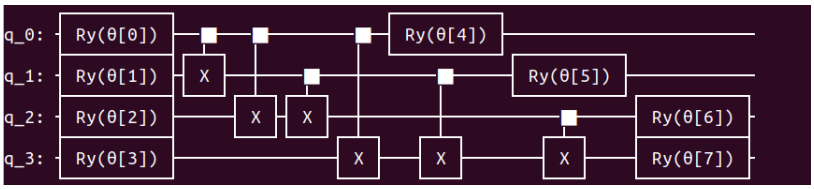


Figure 1.2: A diagram of a quantum circuit.

One-qubit quantum gates:

$$Pauli \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{1.28}$$

$$Pauli \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{1.29}$$

$$Pauli \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{1.30}$$

$$Rotation\ X \quad R_x(\theta) = \begin{bmatrix} cos(\theta/2) & -isin(\theta/2) \\ -isin(\theta/2) & cos(\theta/2) \end{bmatrix} \tag{1.31}$$

$$Rotation\ Y \quad R_y(\theta) = \begin{bmatrix} cos(\theta/2) & -sin(\theta/2) \\ -sin(\theta/2) & cos(\theta/2) \end{bmatrix} \tag{1.32}$$

$$Rotation\ Z \quad R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{1.33}$$

$$Hadamard \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1.34}$$

$$Phase \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{1.35}$$

$$\pi/8 \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \tag{1.36}$$

$$\textit{Phase shift} \quad P(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \tag{1.37}$$

Two-qubit quantum gates:

$$\textit{Controlled Not} \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.38}$$

$$\textit{Controlled Z} \quad CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{1.39}$$

$$\textit{Controlled U} \quad CU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_1 & u_2 \\ 0 & 0 & u_3 & u_4 \end{bmatrix} \tag{1.40}$$

where u_1, u_2, u_3, u_4 are parameters of the U gate.

$$\textit{SWAP gate} \quad SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{1.41}$$

Three-qubit quantum gates:

$$\textit{Toffoli gate} \quad \textit{Toffoli} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.42}$$

Despite the existence of multiple quantum logic gates, it has been shown that $\{CNOT, H, T\}$ is a universal gate set [14].

1.3 Gauge Theories

Gauge theories appear naturally in condensed matter physics, high energy physics and quantum information science. The general idea of a gauge theory is to expose extra constraints on the Hilbert space, which is called 'gauge constraints' or 'Gauss' law'. To start with we first consider the $\mathbb{Z}_2$ Toric code in condensed matter and then review the Quantum Link Model (QLM) Hamiltonian in high energy physics.

1.3.1 $\mathbb{Z}_2$ Toric Code

We start with the lattice gauge theory for the simplest non-trivial gauge group $\mathbb{Z}_2 = \{-1, 1\}$. We consider the Hamiltonian with $\mathbb{Z}_2$ fields on the edges $e \in E$ of a periodic square lattice,

$$H = -J \sum_{f \in F} B_f - \sum_{v \in V} A_v \ , \tag{1.43}$$

with $B_f := \prod_{e \in f} Z_e$, where F is the set of the smallest 1×1 plaquettes $\square$ on the lattice, $e \in f$ are the edges around plaquette f, V be the set of vertices on the lattice. We further define a local operator for each vertex $v \in V$ consisting of the product of Pauli-X operators incident on the given vertex, $A_v := \prod_{e \ni v} X_e$, where $e \ni v$ indicates the edges containing v. Z_e, X_e are the Pauli Z and Pauli X matrices. These vertex operators commute amongst themselves, commute with the Hamiltonian H and have eigenvalues ± 1. In this case, A_v is gauge constraint or the Gauss' law operator which generates the physical Hilbert space.

$$\mathcal{H}_{\text{phys}} := \{|\psi\rangle \in \mathcal{H} : A_v|\psi\rangle = |\psi\rangle \ \ \forall v \in V\} \ . \tag{1.44}$$

One important research topic in the gauge theory is to study the confinement/deconfinement phases. For $\mathbb{Z}_2$ toric code, we can define the Wilson loop operator any closed curve C on the lattice as follows

$$\hat{W}_C := \prod_{e \in C} Z_e \ . \tag{1.45}$$

In general, the Wilson loop decays exponentially with area law in the confinement phase, while decays exponentially with perimeter law in the deconfinement phase. It is known that for $\mathbb{Z}_2$ toric code with an additional transverse field term $-h \sum_{e \in E} X_e$ exhibits a phase transition from deconfinement (small h) to confinement (large h) [15].

1.3.2 Quantum Link Model

The Quantum Link Model is an attempt to formulate gauge theory with fermions on a lattice.

$$\begin{aligned} H_{\text{QLM}} = & - w \sum_x \left[\psi_x^\dagger U_{x,x+1} \psi_{x+1} + \psi_{x+1}^\dagger U_{x,x+1}^\dagger \psi_x \right] \\ & + m \sum_x (-1)^x \psi_x^\dagger \psi_x + \frac{g^2}{2} \sum_x E_{x,x+1}^2, \end{aligned} \tag{1.46}$$

The Gauss' law or the gauge constraint operator of this theory is $\widetilde{G}_x = \psi_x^\dagger \psi_x - E_{x,x+1} + E_{x-1,x} + \frac{1}{2}[(-1)^x - 1]$.

The 1+1-d $U(1)$ QLM shares similar physics for 1+1-d QED. One can choose different spin representation for $U_{x,x+1}$ in QLM. Consider the electric flux between a pair of particle and anti-particle as a string. A string forms if the electric flux density is large while a string breaks if the electric flux density is weak. For $S = 1/2$ representation, there exists a string inversion phenomena in real time dynamics for small mass m shown in Fig. 1.3. For $S = 1$ representation, there exists a string breaking phenomena in real time dynamics for small m shown in Fig. 1.4.

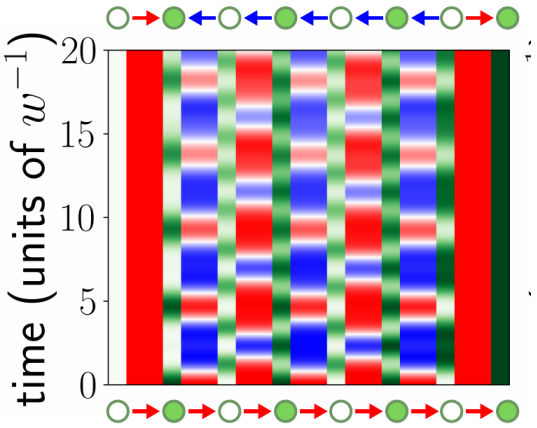


Figure 1.3: String inversion real time dynamics for $S = 1/2$(reproduced from Ref. [16]).

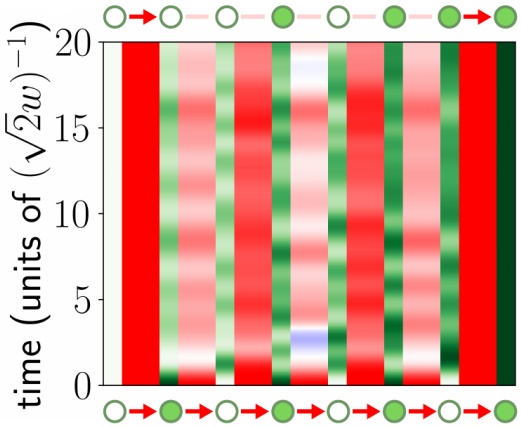


Figure 1.4: String inversion real time dynamics for $S = 1$(reproduced from Ref. [16]).

Chapter 2

Backflow Transformations via Neural Networks for Quantum Many-Body Wave-Functions

This chapter is written based on the research in Ref. [17]. Obtaining an accurate ground state wave function is one of the great challenges in the quantum many-body problem. In this paper, we propose a new class of wave functions, neural network backflow (NNB). The backflow approach, pioneered originally by Feynman [18], adds correlation to a mean-field ground state by transforming the single-particle orbitals in a configuration-dependent way. NNB uses a feed-forward neural network to learn the optimal transformation via variational Monte Carlo. NNB directly dresses a mean-field state, can be systematically improved and directly alters the sign structure of the wave-function. It generalizes the standard backflow[19] which we show how to explicitly represent as a NNB. We benchmark the NNB on Hubbard models at intermediate doping finding that it significantly decreases the relative error, restores the symmetry of both observables and single-particle orbitals, and decreases the double-occupancy density. Finally, we illustrate interesting patterns in the weights and bias of the optimized neural network.

2.1 Introduction

A key question in strongly correlated quantum systems is to obtain an approximation for the ground state wave function. This is especially important for Fermion systems in two or more dimensions where only approximate or exponentially costly methods for evaluating observables of quantum systems exist. Early attempts for writing down variational Fermion wave-functions, such as Slater determinants [20] and BCS wave-functions[21], focused on finding the ground state of a mean field Hamiltonian which best matched the interacting ground state. Since these early attempts more sophisticated wave-functions have been developed which dress these mean-field starting points including Slater-Jastrow [22, 23], Slater-Jastrow-Backflow [18, 24] and iterative backflow [25] which has recently been described as a non-linear network [26]. These wave-functions have the advantage that the mean-field starting point can directly incorporate the basic physics of the problem.

Instead of starting from a dressed mean-field, many other classes of wave-functions are parameterized

by a tuning parameter D which interpolates from a trivial state at small D to a universal wave-function spanning the entire Hilbert space at exponential D. Examples of such wave-functions include matrix-product states[27, 28], other forms of tensor networks[29, 30, 28], Huse-Elser states[31, 32, 33], and string-bond states [34]. Recently, wave-functions based on neural network primitives, such as restricted Boltzmann machines (RBM) and feed forward neural network(FNN), have been introduced with similar universal properties[35, 36, 37, 38, 39, 35, 40, 41, 42, 43, 44, 45, 46, 47, 48, 47, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59]: as the number of hidden neurons increases, the neural network state can represent all probability distribution although may require complex weights to represent the sign structure of Fermion wave-functions. A recent attempt to incorporate RBM into Fermion states by using the RBM as a more general Jastrow [40] shows promise but was still restricted to the sign-structure of the underlying mean-field ansatz. Even though general neural networks could alter the sign-structure, it may struggle with capturing the underlying mean-field physics both in terms of the number of neurons required as well as optimization.

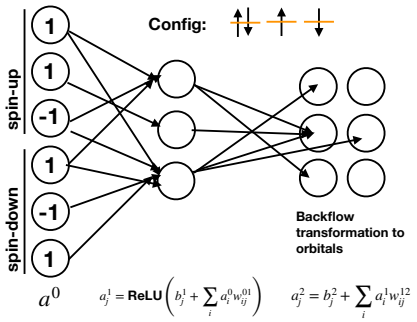


Figure 2.1: Cartoon of (spin-up) neural network being used in this work for a^{NN} (other transformations are similar) with input shown for the configuration displayed of 4 electrons on 3 sites. Every layer is fully connected with arrows but only a fraction of them are shown for image clarity. The input layer is set by the configuration. Parameters b_j and w_{ij} are bias and weights to be optimized. The output layer is a $n_{\text{up-electrons}} \times n_{\text{sites}}$ (i.e 2×3) matrix which will be the backflow transformation added to the single particle orbitals.

In this work, we propose a new class of wave-functions, the Neural Network Backflow (NNB), which dresses a mean-field wave-function, can make changes to the sign structure directly, and can be systematically improved by increasing the number of hidden neurons and is able to be made theoretically exact in the limit of enough neurons. To accomplish this we use a feed-forward neural network (FNN), not in the standard approach of returning a wave-function amplitude, but instead to transform the single particle orbitals in a configuration dependent way; these orbitals are then used in the mean-field wave-function. Wave-functions

with configuration-dependent orbitals are known as a backflow wave-function[18, 60, 61, 62, 63, 64, 19, 65, 24, 25, 26, 66, 67, 68, 69, 70, 71, 72].

2.2 Background

Mean field theory - approximating the ground state of a quartic Hamiltonian by the ground state, ψ_{MF}, of a quadratic Hamiltonian - is a powerful first step to understanding correlated quantum systems. Various mean fields lead to different types of ground states including Slater determinants,

$$\psi_{SD}(\boldsymbol{r}) = \det\left[M^{SD,\uparrow}\right] \det\left[M^{SD,\downarrow}\right]; \tag{2.1}$$

$$M^{SD,\sigma}_{ik} = \phi_{k\sigma}(r_{i\sigma}) \tag{2.2}$$

and Bogoliubov de Gennes wave-functions,

$$\psi_{BDG}(\boldsymbol{r}) = \det[\Phi] \tag{2.3}$$

$$\Phi_{ij} = \sum_{k,l=1}^{N} \phi_{k\uparrow}(r_{i,\uparrow})\, S_{kl}\, \phi_{l\downarrow}(r_{j,\downarrow}) \tag{2.4}$$

where $\phi_{k\sigma}$ is the k'th single particle orbital and $r_{i,\sigma}$ is the position of the i'th particle of spin σ. Eq. 2.1 only takes the occupied orbitals while Eq. 2.3 is summed over both occupied and unoccupied orbitals.

Mean field states are uncorrelated by construction. The simplest way to capture correlation physics is through the introduction of a Jastrow giving $\psi_{\text{Jastrow}}(\boldsymbol{r}) = \exp[-U(\boldsymbol{r})]\psi_{MF}(\boldsymbol{r})$, where $U(\boldsymbol{r})$ is an arbitrary function. In this work, we always use a charge Jastrow $U(\boldsymbol{r}) = \frac{1}{2}\sum_{i,j} v_{ij} n_i n_j$, where n_i is the charge density, v_{ij} is the variational parameters. While Jastrow factors can introduce many-body correlations, they can't modify the mean-field's sign structure. One approach to add additional sign-structure modifying correlation is through a backflow correction [60, 61, 62, 63, 64, 19, 65, 24, 25, 26, 66, 67, 68, 69, 70, 71, 72] which introduces correlations by having the single-particle orbitals act on a configuration-dependent quasi-particle position. On the lattice, the backflow approach instead uses a configuration-dependent mean-field [19, 64, 62, 61] - i.e. the quadratic Hamiltonian or single-particle orbitals $\phi^b_{k,\sigma}(r_i; \mathbf{r})$ depend not only on the position r_i but on all other electron positions $\mathbf{r}$; our NNB builds on top of this formulation of backflow.

2.3 Neural Network Backflow

The NNB uses a FNN to modify the single particle orbitals for a spin σ,

$$\phi^b_{k\sigma}(r_{i,\sigma};\boldsymbol{r}) = \phi_{k\sigma}(r_{i,\sigma}) + a^{NN}_{ki,\sigma}(\boldsymbol{r}) \tag{2.5}$$

where each value of $a^{NN}_{ij,\sigma}$ is represented by an output neuron of the FNN. We use one neural net for each of $\sigma \in \{\uparrow,\downarrow\}$. This is to be contrasted with the standard backflow[61] parameterization,

$$\begin{aligned} \phi^b_{k\sigma}(r_{i,\sigma};\boldsymbol{r}) &= \phi_{k\sigma} + \sum_j \eta_{ij,\sigma}\phi_{k\sigma}(r_{j,\sigma}) \\ \eta_{ij,\sigma} &= tD_iH_j\theta_{|i-j|,\sigma} \end{aligned} \tag{2.6}$$

with $D_i = n_{i,\uparrow}n_{i,\downarrow}$, $H_i = (1-n_{i,\uparrow})(1-n_{i,\downarrow})$. $\theta_{1,\sigma}$ and $\theta_{2,\sigma}$ are the only non-zero variational parameters.

Interestingly, the backflow transformation of Eq. (2.3), can be represented as a neural network for $a^{NN}_{ij,\sigma}(\boldsymbol{r})$ with three hidden layers and a linear number of neurons; an explicit construction will be given in the next section. This ensures that there exists a three layer neural network which is at least as good as the standard backflow transformation.

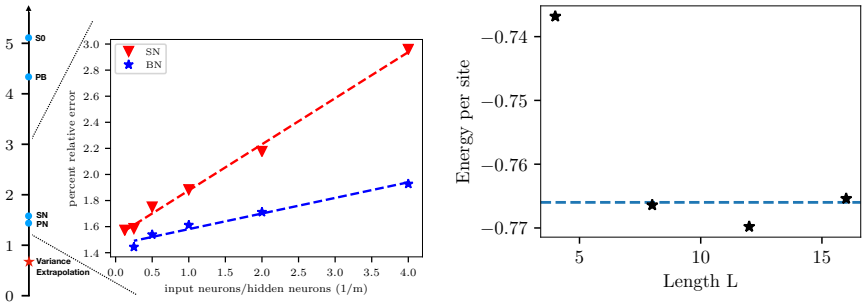


Figure 2.2: **Top Left:** Percentage relative error from the exact ground-state energy of Eq. 2.7 ($E = -11.868$[73]) on 4×4 Hubbard model at U/t=8, n=0.875, for various classes of wave-functions. The star is the variance extrapolation result of $\Psi_{\rm PN}$ (see Supplementary Material [74] Sec. I). **Top Right:** Percentage relative energy error ,$(E_{exact}$-$E_{NNB})/E_{exact}$*100 %, as a function of $1/m$ for NNB. Statistical error bars are shown but smaller than the marker size. **Bottom:** Variance extrapolated energy per site for Hubbard model with U/t=8, n=0.875 with system size $L\times4$ for $L = 4, 8, 12, 16$. The dash line is the DMRG energy per site ($-0.7659 \pm 4\times10^{-5}$) for system size $\infty\times4$ (PBC, open) [75].

We consider two NNB wave-functions, Ψ_{SN} and Ψ_{PN}, implemented on top of a Slater Determinant and

method	backflow transformation	mean field	variational functions
ψ_{S0}	NA.	Eq. (2.1)	$\phi_{k\uparrow}(r_{i,\uparrow})$,$\phi_{k\downarrow}(r_{i,\downarrow})$,$v_{ij}$
Ψ_{SN}	$\phi^b_{k\sigma}(r_{i,\sigma};\boldsymbol{r}) = \phi_{k\sigma}(r_{i,\sigma}) + a^{NN}_{ki,\sigma}(\boldsymbol{r})$	Eq. (2.1)	$a^{NN}_{ki,\sigma}(\boldsymbol{r})$,$v_{ij}$
Ψ_{PB}	$\phi^b_{k\sigma}(r_{i,\sigma};\boldsymbol{r}) = \phi_{k\sigma}(r_{i,\sigma}) + \theta_{1\sigma}\sum_j tD_iH_j\phi_{k\sigma}(r_{j,\sigma}) + \theta_{2\sigma}\sum_j tD_iH_j\phi_{k\sigma}(r_{j,\sigma})$	Eq. (2.3)	$\theta_{1\sigma}$,$\theta_{2\sigma}$,S_{kl},v_{ij}
Ψ_{PN}	$\phi^b_{k\sigma}(r_{i,\sigma};\boldsymbol{r}) = \phi_{k\sigma}(r_{i,\sigma}) + a^{NN}_{ki,\sigma}(\boldsymbol{r})$; $S_{kl}(\boldsymbol{r}) = S_{kl} + d^{NN}_{kl}(\boldsymbol{r})$	Eq. (2.3)	$a^{NN}_{ki,\sigma}(\boldsymbol{r})$,$d^{NN}_{kl}(\boldsymbol{r})$,$v_{ij}$

Table 2.1: Wave Function Ansatzs

BCS pairing wave-functions respectively. The neural nets used in these wave-functions are similar although Ψ_{SN} has outputs which only correspond to the occupied orbitals, while the outputs of Ψ_{PN} correspond to all the orbitals. In addition, for Ψ_{SN} there are only two neural nets (one for each of the spin-up and spin-down orbitals) while for Ψ_{PN} there is an additional neural net used to generate a system dependent $S_{kl}(\boldsymbol{r})$. This is implemented by letting $S_{kl}(\boldsymbol{r}) = S_{kl} + d^{NN}_{kl}(\boldsymbol{r})$, where $d^{NN}_{kl}(\boldsymbol{r})$ is represented by an FNN (in this work always fixed to 16 hidden neurons) that inputs the system configuration $\boldsymbol{r}$ and outputs the symmetric matrix correction d^{NN}_{kl}. Notice that Ψ_{PN} is trivially a superset of Ψ_{SN}.

Although various architectures can be used, we adopt a three-layer fully-connected FNN for each of the functions $a^{NN}_{ki,\sigma}$ and d^{NN}_{kl} (see Fig. 2.1). The input layer has $2N$ neurons with neuron i (neuron $i+N$) outputting 1 if there is spin up (spin down) on site i and -1 otherwise, where N is the total system size. The hidden layer contains mN hidden neurons for constant m with Rectifier Linear Units (ReLU) [76] activation functions. The output layer then contains $O(N^2)$ neurons specifying the values of the respective functions. Gradients are computed in the standard way using variational Monte Carlo (see Supplementary Material [74] Sec. II) which requires evaluating the derivative of the wave-function with respect to the weights and bias in the neural network. Derivatives for FNN are typically taken using back-propagation. Because the wave-function is a determinant of a matrix generated by the neural-network output, we evaluate this full derivative by envisioning this determinant as an additional final layer of the neural network and then performing back-propagation including this layer. This ensures the cost of computing all the derivatives is of the same order as the evaluation of the wave-function (see Supplementary Material [74] Sec. III). Optimization is performed by stepping each parameter in the direction of the gradient with a random magnitude, which helps us avoid shallow local minima [77], or by the RMSPROP method [78].

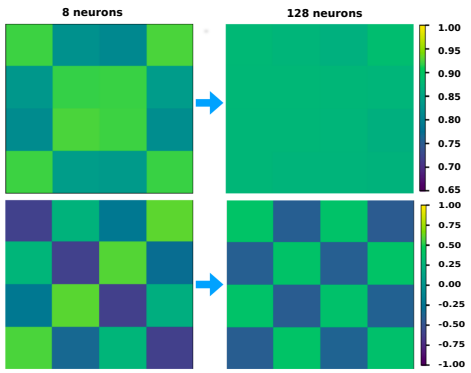


Figure 2.3: Charge Density (Top) and Spin Density (Bottom) from Ψ_{PN} with 8 hidden neurons (left) and 128 hidden neurons (right) on 4×4 Hubbard model at U/t=8, n=0.875.

The computational complexity of the NNB implemented with a single layer of $O(mN)$ hidden neurons scales as $O(mN^4)$ per sweep (i.e. after N electrons move) for forward and backward propagation and $O(N^4)$ per sweep for the evaluation of the mean-field determinant. This is similar to the scaling of standard backflow.

2.4 Explicit Construction of Standard Backflow

In this section, we provide an explicit construction which represents the standard backflow transformation in the form of Eq. (2.3) written as a NNB.

In Eq. (2.3), $\eta_{ij,\sigma} = tD_iH_j\theta_{|i-j|,\sigma} = tn_{i,\uparrow}n_{i,\downarrow}h_{j,\uparrow}h_{j,\downarrow}\theta_{|i-j|,\sigma}$, where $h_{j,\sigma} = 1 - n_{j,\sigma}$, $\theta_{1,\sigma}$ and $\theta_{2,\sigma}$ are the only non-zero variational parameters. We first demonstrate that $\eta_{ij,\sigma}$ can be presented by a two layer neural network with input layer as $(\sigma_1, ..., \sigma_N, \sigma_{N+1}, ..., \sigma_{2N})$ where $\sigma_i = 2n_{i,\uparrow}-1$ and $\sigma_{i+N} = 2n_{i,\downarrow}-1$. By construction, $n_{i,\sigma}$ and $h_{j,\sigma}$ take value of 0 or 1 so that $n_{i,\uparrow}n_{i,\downarrow}h_{j,\uparrow}h_{j,\downarrow}$ is 1 if and only if $n_{i,\uparrow} = n_{i,\downarrow} = h_{j,\uparrow} = h_{j,\downarrow} = 1$. Therefore, $t\theta_{|i-j|,\sigma}D_iH_j = t\theta_{|i-j|,\sigma}n_{i,\uparrow}n_{i,\downarrow}h_{j,\uparrow}h_{j,\downarrow}$ is equivalent to $\mathrm{ReLU}[t\theta_{|i-j|,\sigma}(n_{i,\uparrow} + n_{i,\downarrow} + h_{j,\uparrow} + h_{j,\downarrow} - 3)]$, which is the same as $\mathrm{ReLU}[t\theta_{|i-j|,\sigma}(\sigma_i/2 + \sigma_{i+N}/2 - \sigma_j/2 - \sigma_{j+N}/2 - 1)]$. As a result, for each $\eta_{ij,\sigma}$, we associate it with a hidden neuron, such that the weights connecting it to $\sigma_i, \sigma_{i+N}, \sigma_j, \sigma_{j+N}$ are $t\theta_{|i-j|,\sigma}/2, t\theta_{|i-j|,\sigma}/2, -t\theta_{|i-j|,\sigma}/2, -t\theta_{|i-j|,\sigma}/2$ respectively, the bias is $-t\theta_{|i-j|,\sigma}$ and the activation function is ReLU. In general, for more complicated backflow [19, 64, 62] with terms $n_{i,\sigma}h_{i,-\sigma}n_{j,-\sigma}h_{j,\sigma}$, $n_{i,\sigma}n_{i,-\sigma}n_{j,-\sigma}h_{j,\sigma}$ and $n_{i,\sigma}h_{i,-\sigma}h_{j,\sigma}h_{j,-\sigma}$, where σ is the spin index, we can use more hidden neurons and represent it in the same way.

After we have the neural network construction for the standard $\eta_{ij,\sigma}$, the term $a_{ki}^{NN} = \sum_j \eta_{ij,\sigma}\phi_{k\sigma}(r_{j,\sigma})$ in Eq. (2.5) can be realized through an extra layer taking the outputs $\eta_{ij,\sigma}$ to a neuron representing a_{ki} where

the weight is given by the single particle orbital values $\phi_{k\sigma}(r_{j,\sigma})$, there is no bias and the activation function is the identity. This construction shows that the standard backflow parameterization is thus a subset of our three-layer NNB.

2.5 Results

We have benchmarked the quality of our NNB on a number of systems including Hubbard models at various sizes and doping (all at $U/t = 8$) as well as a frustrated magnet, the Heisenberg model on the Kagome lattice. In the main text we focus primarily on the Hubbard model at $n = 0.875$ (primarily on the 4×4 lattice) leaving the additional benchmarks as Supplementary Material [74] Sec. IV., [79, 80, 81]) The Hubbard Hamiltonian is

$$H = -t \sum_{i\sigma} (c^\dagger_{i\sigma} c_{i+1\sigma} + h.c.) + \sum_i U n_{i\uparrow} n_{i\downarrow} \tag{2.7}$$

where we use $U/t = 8$. We compare the results to an optimized unrestricted (i.e. different single particle-orbitals for spin-up and spin-down) Slater Determinant (Ψ_{S0}) as well as a backflow BDG wave function (Ψ_{PB}) which transforms single particle orbitals of each spin by Eq. (2.3). The formulation and the variational parameters of each wave function ansatz are summarized in Table. 2.1. The parameters which aren't optimized, such as the initial set of orbitals $\{\phi_{k\sigma}(r_{i,\sigma})\}$ are obtained for Ψ_{PB} and Ψ_{SN} by optimizing a restricted Slater-Jastrow wave-function while Ψ_{PN} uses orbitals taken from the free hopping Hamiltonian (in practice the nature of the neural net allows for a direct change to the orbitals by altering the bias' on the final layer).

The relative error of the energy of NNB is 1.4% (and 0.66% after variance extrapolation (see Supplementary Material [74] Sec. I) which is significantly better then the standard wave-functions (see Fig. 2.2(left)). We examine the effect of the number of hidden neurons mN (see Fig. 2.2(right)). We find that at small hidden neuron number, Ψ_{PN} is much better than Ψ_{SN} but this advantage eventually largely disappears at large neuron number suggesting that a backflow parameterized with a small neural networks can compensate for the missing pairing in a Slater-determinant. Surprisingly in the regime we've probed both NNB have energies linear with respect to $1/m$ in spite of the fact that in the $m \to \infty$ limit, they both must become exact as the FNN could simply put the exact amplitude $\Psi(R)$ on one element of the diagonal [82], one on the rest of the diagonal and zero everywhere else.

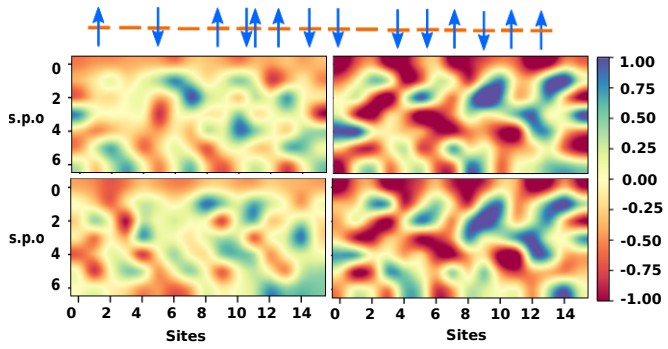


Figure 2.4: Spin-up (top) and spin-down (bottom) single particle orbital(s.p.o) for Ψ_{S0} (left) and Ψ_{SN} (right) with 256 hidden neurons on 4×4 Hubbard model at U/t=8, n=0.875, where the s.p.o are evaluated at the (1d-reshaped) spin-configuration shown. For the s.p.o, row is orbital index and the column is position index.

In addition, we have studied the NNB at $4 \times L$ for $L = \{4, 8, 12, 16\}$ comparing against the DMRG energy (PBC,open) for the $4 \times \infty$ system of $-0.7659 \pm 4 \times 10^{-5}$ per site [75]. We find our result very comparable (see Fig. 2.2 (bottom)) to the DMRG result, especially for system sizes which are commensurate with wave-length 8 stripes[75].

We also investigate how the neural network backflow wave function affects the observables of our system. We see the expectation of double occupancy decreases as $1/m$ (see Supplementary Material [74] Sec. I) and the spin and charge densities become significantly more symmetric as the number of hidden neurons increases (see Fig. 2.3).

To understand the role of neural network in backflow transformation, we investigate how the neural network backflow transformation modify the orbitals. In Fig. 2.4 (right), we notice although the backflow transformation on spin up orbitals and spin down orbitals are performed by two different neural networks, they produce similar backflow transformed orbitals and roughly preserves the symmetry of spin up and spin down for a given configuration. This is different from the optimized unrestricted Slater Determinant Ψ_{S0}, which breaks the spin up and spin down symmetry significantly (see Fig. 2.4 (Left)).

One feature of using a NNB is the ability to alter the sign-structure of the wave-function. Here we consider the amount the sign changes between Ψ_{SN} with 16 hidden neurons and Ψ_{S0} by evaluating the integral

$$\frac{\int |\Psi_{\mathrm{S0}}(x)|^2 \mathrm{sgn}(\Psi_{\mathrm{SN}}(x)) \mathrm{sgn}(\Psi_{\mathrm{S0}}(x)) dx}{\int |\Psi_{\mathrm{S0}}(x)|^2 dx} \tag{2.8}$$

which is approximately 0.815 giving a 9% difference between the signs.

Furthermore, we open up the Ψ_{SN} neural network for $m = 8$ and analyze the weight between the input

layer and the hidden layer, which represents the features that the neural network learns from input. In Fig. 2.5, we plot these weights for both the spin-up and spin-down neural networks. Interestingly the spin-up neural network primarily has large weights connected to the spin-down configurations while the spin-down neural network primarily has large weights connected to the spin-up configurations. This allows the neural network to introduce correlation between spin-up and spin-down configurations. Another observation is that more neurons tend to take large weight in negative bias, and small weight in positive bias.

2.6 Conclusion

In this paper, we utilize the generality of artificial neural networks and the physical insight from backflow to develop a new class of wave function ansatz, the neural network backflow wave function, for strongly correlated Fermion systems on lattice. It achieves good performance for Hubbard model at nontrivial filling. We also show improvement on a kagome Heisenberg model in the supplement. While this work has focused on Fermion system on the lattice, the NNB is straightforward to generalize to frustrated spin systems as well as the continuum. In the latter case, the input could be represented as a lexicographically ordered set of particle locations. Our work provides a new approach toward combining machine learning methodology with dressed mean-field variational wave-functions which allows us to take simultaneous advantage of their respective strengths.

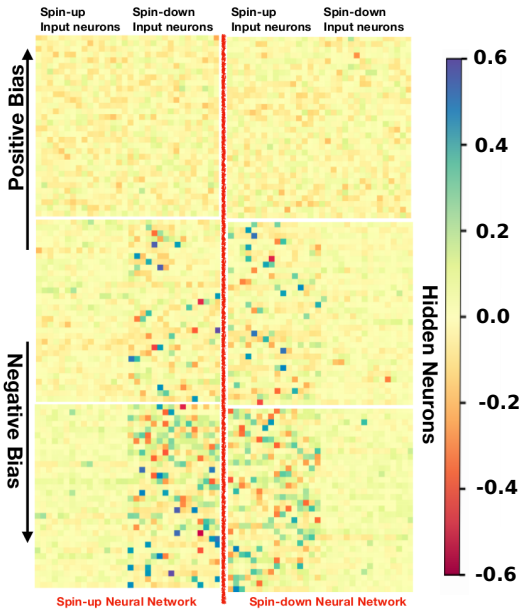


Figure 2.5: Weights between the input layer and hidden layer for Ψ_{SN} with 256 hidden neurons for the spin-up (left) and spin-down (right) neural networks on 4×4 Hubbard model at U/t=8, n=0.875. Hidden neurons are ordered by their bias and shown are neurons 1-32 (top), 96-128 (middle) and 224-256 (bottom).

2.7 Supplementary Materials

2.7.1 Numerical Results on Neural Network Backflow of Standard Form

We have shown that the standard backflow can be represented by a three layer neural network. In practice, we have also investigated the neural network in the following form, which is a direct representation of the standard backflow [61]

$$\phi^{b}_{k\sigma}(r_{i,\sigma};\boldsymbol{r}) = \phi_{k\sigma} + \sum_{j} \eta^{NN}_{ij,\sigma}\phi_{k\sigma}(r_{j,\sigma}) \tag{2.9}$$

where $\eta^{NN}_{ij,\sigma}$ is represented by a three layer neural network similar to the one used for $a^{NN}_{ij,\sigma}$.

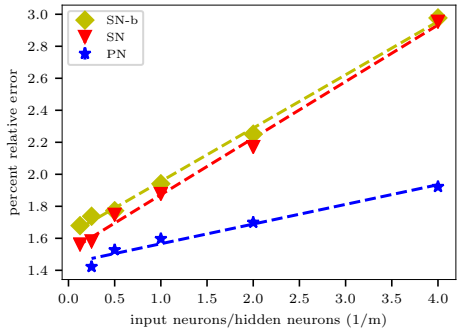


Figure 2.6: Percentage relative energy error as a function of $1/m$ for Ψ_{SN}, $\Psi_{\text{SN-b}}$ and Ψ_{PN}. Statistical error bars for energy are shown.

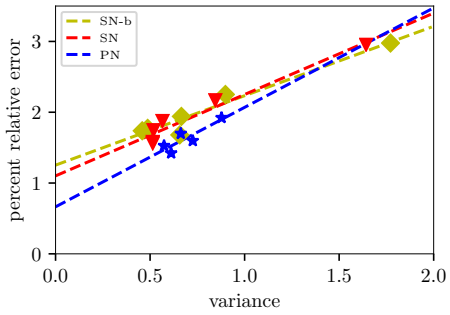


Figure 2.7: Extrapolation of percentage relative energy error vs. variance of Ψ_{SN}, $\Psi_{\text{SN-b}}$ and Ψ_{PN}. Statistical error bars for energy and variance are shown.

We implement this backflow transformation in the Slater Determinant mean field, which is named as $\Psi_{\text{SN-b}}$. Notice that $\Psi_{\text{SN-b}}$ can also be represented by Ψ_{SN} since we can define $a^{NN}_{ij,\sigma} = \sum_j \eta^{NN}_{ij,\sigma}\phi_{k\sigma}(r_{j,\sigma})$. In practice, Ψ_{SN} and $\Psi_{\text{SN-b}}$ have similar performance in computing energy and doublon density (see Fig. 2.6, Fig. 2.7 and Fig. 2.8).

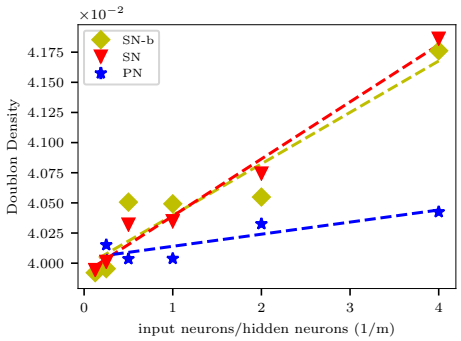


Figure 2.8: Expectation value of doublon density as a function of $1/m$ for Ψ_{SN}, $\Psi_{\text{SN-b}}$ and Ψ_{PN}. Statistical error bars for doublon density are shown.

2.7.2 Optimization Scheme

Given a wave function ψ and Hamiltonian H, the energy $E = \frac{\langle\psi|H\psi\rangle}{\langle\psi|\psi\rangle}$. For any variational parameter t, to compute the derivative $\frac{\partial E}{\partial t}$ in quantum Monte Carlo, we define the local energy $E_L = \frac{\langle \boldsymbol{r}|H\psi\rangle}{\psi(\boldsymbol{r})}$, where $\boldsymbol{r}$ is a system configuration. Then the derivative can be computed through Monte Carlo sampling.

$$\frac{\partial E}{\partial t} = 2\left(\langle E_L(\frac{1}{\psi}\frac{\partial \psi}{\partial t})\rangle - E\langle\frac{1}{\psi}\frac{\partial \psi}{\partial t}\rangle\right) \tag{2.10}$$

where $\langle\ldots\rangle$ is averaged over the Monte Carlo samples.

To update the parameter t in the $k+1$ -th iteration, we use the sign of the gradient with a random magnitude as follows.

$$t_{k+1} = t_k - \alpha\gamma\frac{|\partial E/\partial t_k|}{\partial E/\partial t_k} \tag{2.11}$$

where α is a random number in the range $(0, 1)$, γ is the step size, t_k and t_{k+1} are the values for variational parameter t at the k-th and $k+1$-th iteration. The scheme that alpha is random is inspired by the method described in [77]. This method can help to get the optimization out of shallow local minima and is less sensitive to noise then standard gradient descent approaches. As a practical matter, we have optimized using this approach, gradient descent and stochastic reconfiguration and find that this approach is the best tradeoff between speed and quality of optimization. In practice, we choose $\gamma = 0.001$, weights and bias' of the neural network are initialized from uniform distribution in the range of $(-0.005, 0.005)$.

Notice that to represent Slater Determinant with NNB, one can set the bias of the output layer in NNB to be the orbitals while the other parameters to be zero. This observation can reduce the difficulty of

optimization by initializing NNB with the optimized Slater Determinant, where the bias of the output layer is the optimized orbitals and the other parameters are small random numbers around zero. This trick is helpful for optimization in large systems.

2.7.3 Backward Propagation of Neural Network Backflow

From the optimization scheme, it is clear that we need to compute $\frac{1}{\psi}\frac{\partial \psi}{\partial t}$. In this section, we give the details on how to compute this quantity by backward propagation.

In neural network algorithm, forward propagation refers to the evaluation of the network while backward propagation is essentially performing chain rule to compute derivatives. For forward propagation, the values in the $i+1$-th layer are computed from the values in the i-th layer under matrix operation and activation function, so that the output layer's values can be obtained from the input layer's values. For backward propagation, intuitively one can understand it as backward propagation of 'errors', where the 'errors' in the i-th layer are computed from the 'errors' in the $i+1$-th layer. More accurately, the 'errors', such as ${}^i\delta_p^l$ in Eq. (2.16), are related to the derivatives of the parameters. It indicates that the derivatives of parameters in the i-th layer level can be computed from the derivatives of in the $i+1$-th layer level, which is essentially the chain rule of computing derivatives. A detailed description of applying backward propagation in NNB is given as follows.

Consider backward propagation in a neural network with N layers. Denote the value in each layer by a set of vectors $a^0, a^1, ..., a^{N-1}$. Denote the value before activation in each layer (counting from the 2nd layer) by a set of vectors $z^1, z^2, ..., z^{N-1}$. Denote the activation function in each layer (counting from the 2nd layer) by a set of functions $f_1, f_2, ..., f_{N-1}$. Denote the weights by a set of weights (counting from the 2nd layer) by a set of matrix $w^1, w^2, ..., w^{N-1}$ and the bias by a set of vectors by a set of vector $b^1, b^2, ..., b^{N-1}$. Then a^0 is the input configuration, for $1 < l \leq N-1$, forward propagation reads

$$z^l = w^l a^{l-1} + b^l \tag{2.12}$$

$$a^l = f_l(z^l) \tag{2.13}$$

To calculate derivative, we use backward propagation. For the i-th output in the final layer, denote the error of each layer (counting from the 2nd layer) with respect to this output by a set of vectors ${}^i\delta^1, {}^i\delta^2, ..., {}^i\delta^{N-1}$. Set ${}^i\delta^{N-1} = f'_{N-1}(z_i^{N-1})$. For $1 \leq l \leq N-1$, backward propagation reads

$$
\begin{aligned}
{}^{i}\delta^{l-1} &= ((w^{l})^{T}\,{}^{i}\delta^{l}) * f'_{l-1}(z^{l-1}) && (2.14)\\
\frac{\partial a_i^{N-1}}{\partial b_p^l} &= {}^{i}\delta_p^l && (2.15)\\
\frac{\partial a_i^{N-1}}{\partial w_{pq}^l} &= a_q^{l-1} \otimes {}^{i}\delta_p^l && (2.16)
\end{aligned}
$$

where $*$ is the element-wise multiplication and the last equation uses outer product $\otimes$.

To transfer backward propagation to $\frac{1}{\psi}\frac{\partial \psi}{\partial t}$, we should view it as a special mapping on top of a^{N-1}. Therefore, we only need to choose a different ${}^{i}\delta^{N-1}$ and the rest will be the same as above. First, we consider the Slater-Determinant type neural network backflow.

$$
\begin{aligned}
\frac{1}{\psi_{SD}}\frac{\partial \psi_{SD}(\boldsymbol{r})}{\partial t} = &\frac{1}{det[M^{SD,\uparrow}]}\frac{\partial \det\left[M^{SD,\uparrow}\right]}{\partial t}\\
&+ \frac{1}{det[M^{SD,\downarrow}]}\frac{\partial \det\left[M^{SD,\downarrow}\right]}{\partial t};
\end{aligned}
\tag{2.17}
$$

where $M_{ik}^{SD,\sigma} = \phi_{k\sigma}^{b}(r_{i\sigma})$, t is any variational parameter $\{w_{pq}^{l}, b_{p}^{l}\}$. Under the Einstein notation, the chain rule gives rise to,

$$
\begin{aligned}
&\frac{1}{det[M^{SD,\sigma}]}\frac{\partial \det\left[M^{SD,\sigma}\right]}{\partial t} && (2.18)\\
&= \frac{1}{det[M^{SD,\sigma}]}\frac{\partial \det\left[M^{SD,\sigma}\right]}{\partial \phi_{k\sigma}^{b}(r_{i\sigma})}\frac{\partial \phi_{k\sigma}^{b}(r_{i\sigma})}{\partial t} && (2.19)\\
&= tr((M^{SD,\sigma})^{-1}\frac{\partial \det\left[M^{SD,\sigma}\right]}{\partial \phi_{k\sigma}^{b}(r_{i\sigma})})\frac{\partial \phi_{k\sigma}^{b}(r_{i\sigma})}{\partial t} && (2.20)\\
&= (M^{SD,\sigma})_{ki}^{-1}\frac{\partial \phi_{k\sigma}^{b}(r_{i\sigma})}{\partial t} && (2.21)
\end{aligned}
$$

For Ψ_{SN}, we have

$$
\begin{aligned}
&\frac{1}{det[M^{SD,\sigma}]}\frac{\partial \det\left[M^{SD,\sigma}\right]}{\partial t} && (2.22)\\
&= (M^{SD,\sigma})_{ki}^{-1}\frac{\partial \phi_{k\sigma}^{b}(r_{i\sigma})}{\partial a_{ki,\sigma}^{NN}(\boldsymbol{r})}\frac{a_{ki,\sigma}^{NN}(\boldsymbol{r})}{\partial t} && (2.23)\\
&= (M^{SD,\sigma})_{ki}^{-1}\frac{a_{ki,\sigma}^{NN}(\boldsymbol{r})}{\partial t} && (2.24)
\end{aligned}
$$

Therefore, we set ${}^i\delta_{ki}^{N-1} = (M^{SD,\sigma})_{ki}^{-1} f'_{N-1}(z_{ki}^{N-1})$ and perform the backward propagation.Notice that we use a matrix index ki for simplicity here. In practice, a matrix will be reshaped into an array for computation.

For $\Psi_{\text{SN-b}}$, we have

$$\frac{1}{det[M^{SD,\sigma}]} \frac{\partial \det\left[M^{SD,\sigma}\right]}{\partial t} \tag{2.25}$$

$$= (M^{SD,\sigma})_{ki}^{-1} \frac{\partial \phi_{k\sigma}^b(r_{i\sigma})}{\partial \eta_{ij,\sigma}^{NN}(\boldsymbol{r})} \frac{\eta_{ij,\sigma}^{NN}(\boldsymbol{r})}{\partial t} \tag{2.26}$$

We then set ${}^i\delta_{ij}^{N-1} = (M^{SD,\sigma})_{ki}^{-1} \frac{\partial \phi_{k\sigma}^b(r_{i\sigma})}{\partial \eta_{ij,\sigma}^{NN}(\boldsymbol{r})} f'_{N-1}(z_{ij}^{N-1})$ and perform the backward propagation. Notice that we use a matrix index ij for simplicity here. In practice, a matrix will be reshaped into an array for computation.

Next, we consider Bogoliubov de Gennes type neural network backflow wave-functions Ψ_{PN},

$$\frac{1}{\psi_{BDG}} \frac{\partial \psi_{BDG}(\boldsymbol{r})}{\partial t} = \frac{1}{det[\Phi]} \frac{\partial \det[\Phi]}{\partial t} \tag{2.27}$$

where $\Phi_{ij} = \sum_{k,l=1}^N \phi_{k\uparrow}^b(r_{i,\uparrow}) S_{kl} \phi_{l\downarrow}^b(r_{j,\downarrow})$, t is any variational parameter $\{w_{pq}^l, b_p^l\}$ of $a_{ij,\sigma}^{NN}$ and d_{kl}^{NN}.

For the case that t is a parameter of $a_{ki,\uparrow}^{NN}$, we rewrite $\Phi_{ij} = \sum_{k,l=1}^N \phi_{k\uparrow}^b(r_{i,\uparrow}) R_{kj}$, where $R_{kj} = \sum_l^N S_{kl}^{NN} \phi_{l\downarrow}^b(r_{j,\downarrow})$, $\phi_{k\uparrow}^b(r_{i,\uparrow})$ is given by Eq. (5) in the main body of the paper and $S_{kl}^{NN} = S_{kl} + d_{kl}^{NN}$. The chain rule gives rise to

$$\frac{1}{det[\Phi]} \frac{\partial \det[\Phi]}{\partial t} \tag{2.28}$$

$$= \frac{1}{det[\Phi]} \frac{\partial \det[\Phi]}{\partial \phi_{k\uparrow}^b(r_{i,\uparrow})} \frac{\partial \phi_{k\uparrow}^b(r_{i,\uparrow})}{\partial a_{kl}} \frac{\partial a_{kl}}{\partial t} \tag{2.29}$$

$$= \frac{1}{det[\Phi]} \frac{\partial det[\Phi]}{\partial \phi_{k\uparrow}^b(r_{i,\uparrow})} \frac{\partial a_{kl}}{\partial t} \tag{2.30}$$

$$= \Phi_{ji}^{-1} R_{kj} \frac{\partial a_{kl}}{\partial t} \tag{2.31}$$

We then set ${}^i\delta_{ki}^{N-1} = \Phi_{ji}^{-1} R_{kj} f'_{N-1}(z_{ki}^{N-1})$ and perform the backward propagation. For t is a parameter of $a_{ki,\downarrow}^{NN}$, similarly we define $Q_{il} = \sum_k^N (\phi_{k\downarrow}^b)^T(r_{i,\downarrow}) S_{kl}^{NN}$ and then set ${}^i\delta_{ki}^{N-1} = \Phi_{ij}^{-1} Q_{jk} f'_{N-1}(z_{ki}^{N-1})$.

For t is a parameter of d_{kl}^{NN},

$$\frac{1}{det[\Phi]}\frac{\partial \det[\Phi]}{\partial t} \tag{2.32}$$

$$= \frac{1}{det[\Phi]}\frac{\partial \det[\Phi]}{\partial S^{NN}_{kl}}\frac{\partial S^{NN}_{kl}}{\partial d_{kl}}\frac{\partial d_{kl}}{\partial t} \tag{2.33}$$

$$= \frac{1}{det[\Phi]}\frac{\partial \det[\Phi]}{\partial S^{NN}_{kl}}\frac{\partial d_{kl}}{\partial t} \tag{2.34}$$

$$= \phi^b_{k\uparrow}(r_{i,\uparrow})(\Phi^{-1})^T_{ij}\phi^b_{l\downarrow}(r_{j,\downarrow})\frac{\partial d_{kl}}{\partial t} \tag{2.35}$$

We set ${}^i\delta^{N-1}_{ki} = \phi^b_{k\uparrow}(r_{i,\uparrow})(\Phi^{-1})^T_{ij}\phi^b_{l\downarrow}(r_{j,\downarrow})f'_{N-1}(z^{N-1}_{ki})$ and perform the backward propagation.

2.7.4 Neural Network Backflow on Various Systems

To further test the ability of NNB, we have optimized $\Psi_{\rm SN}$ in various systems: (i) 4×4 Hubbard model at $U/t = 8$ with $n = 0.75$ and $n = 1.0$ (See Fig. 2.9, 2.10, 2.11). (ii) 8×4, 12×4, 16×4 Hubbard model with $n = 0.875$ and $U/t = 8$ (See Fig. 2.15, 2.16, 2.12). (iii) 12×8 Hubbard model at $U/t = 8$ with $n = 0.875$ (See Fig. 2.13). (iv) $4\times 4\times 3$ Kagome Heisenberg model with $J = 1$ ((See Fig. 2.14)). The optimization has been implemented with the RMSPROP [78] method.

For the 4×4 Hubbard model, we find that the smallest doping ($n = 0.75$) does best with an error (variance extrapolated error) of 0.631% (0.233%) with the largest doping ($n = 1.0$) doing the worst giving a 2.714% (1.745%) error (See Fig. 2.9, 2.10, 2.11 for details). The results are compared with the exact diagonalization data [73].

We consider both the 16×4 and 12×8 lattice at $U = 8$ and $n = 0.875$. Because there are no exact answers at these system sizes, we compare against approximate (not necessarily variationally upper-bounded) AFQMC. For our 16×4 system, using a number of hidden neurons $n_h = \{8, 16, 32, 64\}$, we find the energy decreases from -46.211 (for the optimized Slater determinant) to -47.745 ($n_h = 64$). The variance extrapolated energy is approximately -49 which is reasonably close to the twist-averaged boundary condition AFQMC result of -49.088[81] (See Fig. 2.12). In the 12×8 system we use $n_h = \{8, 16, 32\}$ with relative errors against the same system in AFQMC [75] of 6.3% for the optimized Slater determinant, 3.94% for the $n_h = 32$ NNB and 0.655% after variance extrapolation (See Fig. 2.13). NNB significantly improves on the standard Slater-Jastrow methodology and the variance extrapolation result is competitive with other techniques.

Finding wave-functions for the ground state of the Heisenberg model on the Kagome lattice is challenging. The best per site energy of a projected BDG result on a $4\times 4\times 3$ lattice is approximately -0.4305 [80] compared to the exact answer of -0.4387 [79]. We have implemented the neural network backflow on top of the Slater

Relative energy error	Slater-Jastrow	NNB	NNB neuron extrapolation	NNB variance extrapolation
4×4 Hubbard, n=0.75	$(3.6 \pm 6 \times 10^{-4})\%$	$(0.631 \pm 8 \times 10^{-4})\%$	0.607%	0.233%
4×4 Hubbard, n=0.875	$(5.4 \pm 2 \times 10^{-3})\%$	$(1.156 \pm 2 \times 10^{-3})\%$	1.152%	1.101%
4×4 Hubbard, n=1.0	$(5.9 \pm 2 \times 10^{-3})\%$	$(2.714 \pm 5 \times 10^{-3})\%$	2.700%	1.745%
16×4 Hubbard, n=0.875	$(5.9 \pm 2 \times 10^{-3})\%$	$(2.734 \pm 8 \times 10^{-3})\%$	2.592%	0.209%
12×8 Hubbard, n=0.875	$(6.3 \pm 3 \times 10^{-3})\%$	$(3.94 \pm 10^{-2})\%$	3.727%	0.655%
$4 \times 4 \times 3$ Kagome	$(1.8 \pm 10^{-5})\%$	$(1.093 \pm 4 \times 10^{-3})\%$	1.055%	0.286%

Table 2.2: Neural Network Backflow performance on various systems

Determinant with $n_h = \{8, 16, 32, 64, 128, 256\}$. The associated energies ranges from -0.4311 ($n_h = 8$) to -0.4339 ($n_h = 256$), giving an improvement of the relative error of approximately 50% (See Fig. 2.14).

The results are summarized in Table 2.2 and 2.3.

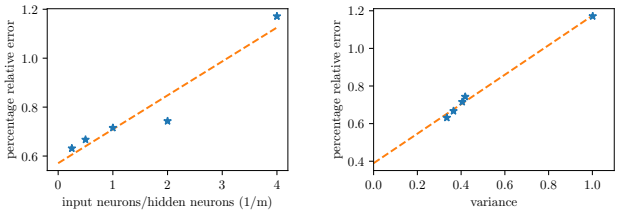


Figure 2.9: Ψ_{SN} on 4×4 Hubbard model with filling $n = 0.75$. **Left**: Percentage relative energy error as a function of $1/m$. **Right**: Percentage relative energy error as a function of variance. Statistical error bars for energy are shown.

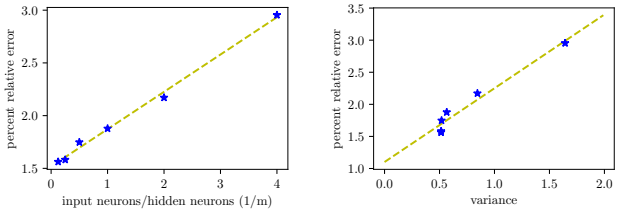


Figure 2.10: Ψ_{SN} on 4×4 Hubbard model with filling $n = 0.875$. **Left**: Percentage relative energy error as a function of $1/m$. **Right**: Percentage relative energy error as a function of variance. Statistical error bars for energy are shown.

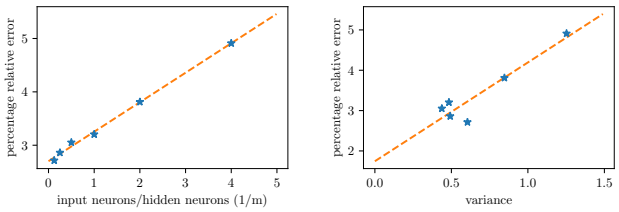


Figure 2.11: Ψ_{SN} on 4×4 Hubbard model with filling $n = 1.0$. **Left**: Percentage relative energy error as a function of $1/m$. **Right**: Percentage relative energy error as a function of variance. Statistical error bars for energy are shown.

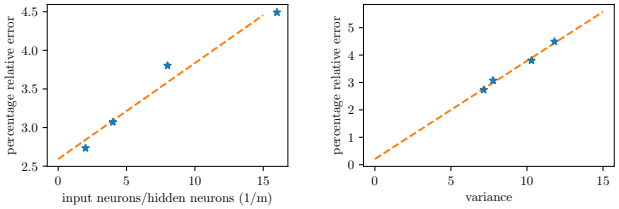


Figure 2.12: Ψ_{SN} on 16×4 Hubbard model with filling $n = 0.875$. **Left**: Percentage relative energy error as a function of $1/m$. **Right**: Percentage relative energy error as a function of variance. Statistical error bars for energy are shown.

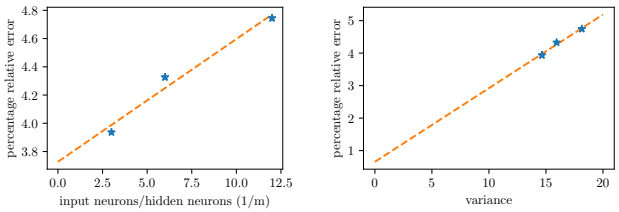


Figure 2.13: Ψ_{SN} on 12×8 Hubbard model with filling $n = 0.875$. **Left**: Percentage relative energy error as a function of $1/m$. **Right**: Percentage relative energy error as a function of variance. Statistical error bars for energy are shown.

Hubbard model energy per site	Slater-Jastrow	NNB	NNB variance extrapolation
4×4, U/t=8, n=0.875	-0.702 ± 10^{-5}	$-0.730 \pm 8 \times 10^{-6}$	-0.734
8×4, U/t=8, n=0.875	-0.719 ± 10^{-5}	$-0.755 \pm 4 \times 10^{-5}$	-0.767
12×4, U/t=8, n=0.875	$-0.722 \pm 2 \times 10^{-5}$	$-0.746 \pm 5 \times 10^{-5}$	-0.770
16×4, U/t=8, n=0.875	-0.722 ± 10^{-5}	$-0.746 \pm 6 \times 10^{-5}$	-0.765

Table 2.3: Finite size effect study of Neural Network Backflow

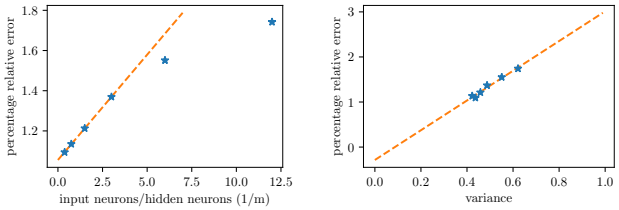


Figure 2.14: Ψ_{SN} on $4 \times 4 \times 3$ Kagome model. **Left**: Percentage relative energy error as a function of $1/m$. **Right**: Percentage relative energy error as a function of variance. Statistical error bars for energy are shown.

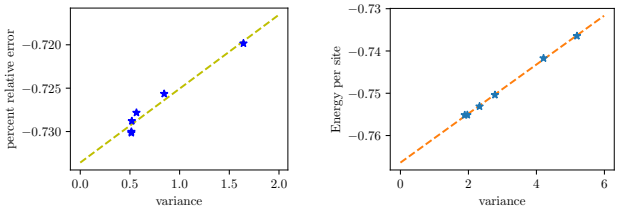


Figure 2.15: Ψ_{SN} energy per site vs. variance on Hubbard model at U/t=8, n=0.875. **Left**: 4×4 Hubbard. **Right**: $8 \times 4s$ Hubbard. Statistical error bars for energy are shown.

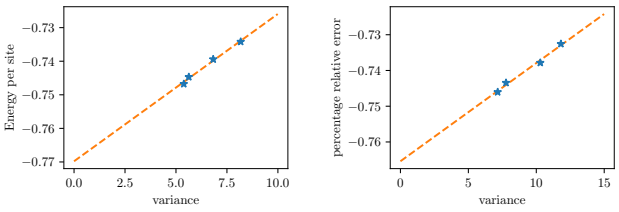


Figure 2.16: Ψ_{SN} energy per site vs. variance on Hubbard model at U/t=8, n=0.875. **Left**: 12×4 Hubbard. **Right**: 16×4 Hubbard. Statistical error bars for energy are shown.

Chapter 3

Gauge equivariant neural networks for quantum lattice gauge theories

This chapter is written based on the research in Ref. [83]. Gauge symmetries play a key role in physics appearing in areas such as quantum field theories of the fundamental particles and emergent degrees of freedom in quantum materials. Motivated by the desire to efficiently simulate many-body quantum systems with exact local gauge invariance, gauge equivariant neural-network quantum states are introduced, which exactly satisfy the local Hilbert space constraints necessary for the description of quantum lattice gauge theory with $\mathbb{Z}_d$ gauge group and non-abelian Kitaev $D(G)$ models on different geometries. Focusing on the special case of $\mathbb{Z}_2$ gauge group on a periodically identified square lattice, the equivariant architecture is analytically shown to contain the loop-gas solution as a special case. Gauge equivariant neural-network quantum states are used in combination with variational quantum Monte Carlo to obtain compact descriptions of the ground state wavefunction for the $\mathbb{Z}_2$ theory away from the exactly solvable limit, and to demonstrate the confining/deconfining phase transition of the Wilson loop order parameter.

3.1 Introduction

Quantum many-body systems defined on a lattice with local gauge invariance occur ubiquitously in the description of physics at diverse energy scales — they appear in the effective theories of many-electron systems [84] and topological phases [85], in bosonization of two-dimensional lattice fermions [86, 87], and in the microscopically regulated description of interacting elementary particles in the Standard Model [88]. Quantum lattice gauge theories are characterized by the fact that the physical states span a subspace of the many-body Hilbert space which is defined by satisfying a set of local operator constraints. These operator constraints gives rise to group invariance of the associated wavefunction, called gauge invariance.

Because of the analytic intractability of gauge theories, it is important to develop techniques for simulating them. Methods such as lattice-gauge theory [89] accomplish this via a quantum-classical mapping which rewrites the quantum problems as a statistical mechanics problem in one higher dimension. This mapping can only be done for sign-free problems without introducing a 'negative' weight in the statistical mechanics model

which induces either an exponential cost in the simulations or the need for additional approximations to mitigate the sign-problem. Alternatively, quantum gauge theories can be simulated using variational methods based on compact parameterizations of many-body wavefunctions. DMRG [90], a variational approach based on matrix-product states, has been extensively used to analyze gauge theories [91]. While DMRG is efficacious for one-dimensional systems, the bond-dimension of the matrix product state needed to accurately represent the ground state grows exponentially with width in higher dimensions.

Machine-learning techniques based on a neural-network variational representation of quantum states [92], have extended the scope of variational methods by accurately representing low-energy states of strongly correlated systems in two or more spatial dimensions [93, 94, 6, 95, 96, 97, 98, 99, 100]. Imposing physical symmetries in neural-network quantum states is a very active research topic [101, 102] that reflects the broader need to impose symmetries in machine learning applications to physics [103]. Our work further extends the scope of neural-network quantum states by developing gauge equivariant neural networks which are special-purpose variational families of wave-functions that are explicitly gauge invariant.

Our work is inspired by recent developments of group-invariant networks with group equivariant layers. These have found applications both in data-science and physics. For data-science, the data-generating process is often assumed to be invariant under a symmetry group G [104, 105]. For example, ref. [106] derived equivariant layers starting from the assumption that the input data transforms covariantly as a tensor field on a Riemannian manifold. Very recently, flow-based gauge equivariant networks have been constructed and variationally optimized using the reverse-relative entropy to approximate the Gibbs distribution associated with the Euclidean action functional for lattice Yang-Mills in the case of abelian $U(1)$ [107] and subsequently non-abelian $SU(N)$ gauge group [108]. This approach, which is closely related to variational inference [109], relies on the existence of an analytic continuation between the quantum theory and a positive-definite Gibbs measure, which presents a challenge in the presence of fermions, however, due to the well-known sign problem.

The paper is organized as follows. The $\mathbb{Z}_2$ gauge theory Hamiltonian is introduced using a notation that generalizes to the $\mathbb{Z}_d$ gauge group and higher dimensional models such as 3D toric code and X-cube model (explicitly described in SM III and IV). The general construction of the gauge equivariant neural network is then described. Gauge equivariant neural wavefunctions are variationally optimized using variational Monte Carlo on square lattices up to system size 12×12, demonstrating the transition of Wilson loop order parameters from perimeter to area law.

3.2 $\mathbb{Z}_2$ Gauge Theory

We start by briefly reviewing the formulation of the lattice gauge theory for the simplest non-trivial gauge group $\mathbb{Z}_2 = \{-1, 1\}$. The generalization to $\mathbb{Z}_d$ and higher dimensional models, such as 3D toric code and X-cube model, are discussed in Sec. III and IV of the Supplemental Material. We consider the Hamiltonian with $\mathbb{Z}_2$ fields on the edges $e \in E$ of a periodic square lattice,

$$H = -J \sum_{f \in F} B_f - h \sum_{e \in E} X_e \ , \tag{3.1}$$

with $B_f := \prod_{e \in f} Z_e$, where F is the set of the smallest 1×1 plaquettes $\square$ on the lattice, $e \in f$ are the edges around plaquette f and Z_e, X_e are the usual Pauli matrices. Let V be the set of vertices on the lattice. In order to define the Hilbert space of physical states, define a local operator for each vertex $v \in V$ consisting of the product of Pauli-X operators incident on the given vertex, $A_v := \prod_{e \ni v} X_e$, where $e \ni v$ indicates the edges containing v. These vertex operators commute amongst themselves, commute with the Hamiltonian H and have eigenvalues ± 1. In this work we focus on the so-called even gauge theory in which the physical Hilbert space $\mathcal{H}_{\text{phys}}$ is chosen to be the $+1$ eigenspace of all vertex operators,

$$\mathcal{H}_{\text{phys}} := \{|\psi\rangle \in \mathcal{H} : A_v|\psi\rangle = |\psi\rangle \ \ \forall v \in V\} \ . \tag{3.2}$$

Since the vertex operators satisfy the global operator identity $\prod_{v \in V} A_v = \mathbb{1}$, it follows that only $|V| - 1$ of the constraints defining $\mathcal{H}_{\text{phys}}$ are independent. The dimension of physical state space is therefore found to be $\dim \mathcal{H}_{\text{phys}} = \frac{2^{|E|}}{2^{|V|-1}} = 2^{L^2+1}$. Further details on the description of the Hilbert space are provided in Sec. I of the Supplemental Material.

The $\mathbb{Z}_2$ gauge theory is exactly solvable in both the weak coupling ($h \to 0$) and strong coupling ($h \to \infty$) limits. For infinite transverse field $h = \infty$ the non-degenerate ground state is simply the uniform superposition state $|+\rangle^{\otimes E}$ which is manifestly gauge invariant, where $|+\rangle$ satisfies $X|+\rangle = |+\rangle$. In the opposite extreme of $h = 0$, the ground states are also eigenstates of all B_f and are four-fold degenerate. As shown originally by Wegner using duality arguments [110], the uniform superposition $|+\rangle^{\otimes E}$ and the ground states at $h = 0$ correspond to different phases of matter, which are distinguished by the expectation value of a non-local operator called the Wilson loop which is defined for any closed curve $C \subseteq E$ on the lattice as follows

$$\hat{W}_C := \prod_{e \in C} Z_e \ . \tag{3.3}$$

Wegner found a critical value of the transverse field h_c separating a deconfined phase for $h < h_c$ in which

$\langle \hat{W}_C \rangle$ decays exponentially with the perimeter of C, from a confined phase where $\langle \hat{W}_C \rangle$ decays exponentially with the area enclosed by C.

3.3 Gauge Equivariant Neural Networks

We present now a neural network which explicitly preserves the gauge invariance of the wave-function. The classical configuration space $\mathbb{Z}_2^E$ can be regarded as a subset of the continuous vector space $\mathbb{C}^E := \mathbb{C}^{L \times L \times 2}$ consisting of tensors with shape $(L, L, 2)$, where the edge e is specified by the vertex $v \in V$ indexed by the first two axes and the direction $\mu \in \{\hat{x}, \hat{y}\}$ indexed by the third axis. The components of an arbitrary tensor $\phi \in \mathbb{C}^E$ will then be indexed as $\phi_\mu(v)$ where $v \in V$ and $\mu \in \{\hat{x}, \hat{y}\}$. The action of the gauge group on the space of $(L, L, 2)$ tensors is described as follows. Given a square matrix $\Omega \in \{-1, 1\}^V := \{-1, 1\}^{L \times L}$, we define a gauge transformation $g_\Omega : \mathbb{C}^E \to \mathbb{C}^E$ by the following rule,

$$(g_\Omega \cdot \phi)_\mu(v) := \Omega(v)\phi_\mu(v)\Omega(v + \mu) \ , \tag{3.4}$$

where $\Omega(v)$ and $\Omega(v + \mu)$ denote the entries of the matrix Ω at the lattice location $v \in V$ and the shifted lattice location $v + \mu \in V$, assuming boundaries are periodically identified. It is straightforward to show that the gauge transformation associated with Hilbert space operator A_v is given by g_{Ω_v} where Ω_v is defined for each $v' \in V$ by,

$$\Omega_v(v') = \begin{cases} -1, & v' = v \\ +1, & v' \neq v \end{cases} . \tag{3.5}$$

A wave-function which obeys the Gauss law constraint is then one in which

$$\Psi(g_{\Omega_v} \cdot \phi) = \Psi(\phi), \tag{3.6}$$

for the case where $\phi \in \mathbb{Z}_2^E$. Let us call a function $h : \mathbb{C}^E \to \mathbb{C}$ gauge invariant if it satisfies $h(g_\Omega \cdot \phi) = h(\phi)$ for all g_Ω and a function $f : \mathbb{C}^E \to \mathbb{C}$ gauge equivariant if it satisfies $f(g_\Omega \cdot \phi) = g_\Omega \cdot f(\phi)$ for all g_Ω. A wave-function which consists of multiple layers of gauge equivariant functions interwoven with pointwise nonlinearities followed by a final gauge invariant layer is guaranteed to obey Eq. 3.6 since group equivariance is preserved by composure and pointwise nonlinearities. In the following, we construct neural network blocks which are equivariant and invariant respectively.

3.4 Gauge Equivariant Layer

The construction of the gauge equivariant layer parallels the construction found in [107], applied to discrete abelian groups. Given vertices $v, v' \in V$ and a path $\gamma \subseteq E$ from v to v', consisting of a sequence of adjacent edges, we define the Wilson path as the function $W_\gamma : \mathbb{C}^E \to \mathbb{C}$ given by the formula,

$$W_\gamma(\phi) := \prod_{e \in \gamma} \phi_e \ . \tag{3.7}$$

It is easy to show that for any Ω and any path γ from v to v' we have

$$W_\gamma(g_\Omega \cdot \phi) = \Omega(v) W_\gamma(\phi) \Omega(v') \ . \tag{3.8}$$

The above Wilson path is the fundamental primitive from which the gauge-equivariant layers will be constructed. Note that if γ is a closed curve C then the function W_C is gauge invariant.

Consider a fixed equivariant layer described by the equivariant function $f : \mathbb{C}^E \to \mathbb{C}^E$. Each such layer is specified by decorating the edges $e = (v, v + \mu) \in E$ of the lattice with the following data

1. A path $\gamma_e \subseteq E$ from v to $v + \mu$
2. A collection of $n_e \geq 1$ closed curves $C_1^e, \dots C_{n_e}^e$
3. A parametrized neural network $h_e : \mathbb{C}^{n_e} \to \mathbb{C}$

From the above data we construct a gauge-equivariant function $f : \mathbb{C}^E \to \mathbb{C}^E$ defined for all $\phi \in \mathbb{C}^E$ by the rule $f : \phi_e \mapsto f_e(\phi)$, where

$$f_e(\phi) := W_{\gamma_e}(\phi) h_e\big(W_{C_1^e}(\phi), \dots, W_{C_{n_e}^e}(\phi)\big) \ . \tag{3.9}$$

The equivariance follows directly from Eq. (3.8). The explicit calculation is outlined below for convenience of the reader,

$$\begin{aligned} f_e(g_\Omega \cdot \phi) &= h_e\big(W_{C_1^e}(g_\Omega \cdot \phi), \dots, W_{C_n^e}(g_\Omega \cdot \phi)\big) W_{\gamma_e}(g_\Omega \cdot \phi) \\ &= h_e\big(W_{C_1^e}(\phi), \dots, W_{C_n^e}(\phi)\big) \Omega(v) W_{\gamma_e}(\phi) \Omega(v + \mu) \\ &= \Omega(v) f_e(\phi) \Omega(v + \mu) \qquad (3.10) \\ &= \big(g_\Omega \cdot f(\phi)\big)_e \ . \qquad (3.11) \end{aligned}$$

3.5 Numerical Experiments

Here we determine the phase diagram of Eq. 3.1 for different values of the transverse field h. When $h = 0$, we can analytically write a network which exactly represents the ground state with a single gauge invariant block (and no equivariant blocks), which is detailed in Sec. II of the Supplemental Material. For all h, we use variational Monte Carlo to approximately determine the ground states on square lattices.The family of variational wavefunctions is summarized graphically in Fig. 3.1. It consists of multiple gauge equivariant layers, in which the gauge invariant Wilson loop features within each layer were chosen to consist of all elementary plaquettes of the form $\gamma_e = \square$. The gauge equivariant Wilson path associated with each edge e is chosen to be the curves of the form $\sqcap$ and $\sqsubset$ ending on e. Real-valued weights and biases are used in all neural networks. The neural networks h_e are chosen to be convolutional neural network with periodic padding to capture symmetry and facilitate transfer learning. Residual layers or equivalently skip connections, which are manifestly gauge equivariant, are also employed. The neural network parameters are optimized using the Stochastic Reconfiguration algorithm [111]. Further details about the architecture and optimization are provided in Sec. V of the Supplemental Material.

We start with a benchmark of the hamiltonian in Eq. 3.1 on a 3×3 square lattice by comparing gauge equivariant neural network, gauge invariant network, restricted Boltzmann machine (RBM) and exact diagonalization, where Fig. 3.2 presents the energy and variance. In addition, we benchmark a hamiltonian with a sign problem $H = -J \sum_{f \in F} \prod_{e \in f} Z_e - h \sum_{e \in E} X_e \ - J_y \sum_{f \in F} \prod_{e \in f} Y_e$ for $J = h = 1$, which results are shown in Fig. 3.3. The number of variational parameters for the above three neural networks are 66, 24 and 1044 respectively. It can be seen that even with small number of parameters, the gauge equivariant neural network achieves better performance than the RBM and attains accurate results close to the exact. We further apply our method to larger square lattices of size $L \times L$ with $L \in \{8, 10, 12\}$. It is known that the Wilson loop expectation value $\langle \hat{W}_C \rangle$ decays exponentially with area law for $h > h_c$ and with perimeter law for $h < h_c$ [112]. In Fig. 3.4, the variational wave function is shown to capture the area law and the perimeter law behaviors of the Wilson loop in the corresponding regimes and attains the related decay factors. In Fig. 3.5, we compute the energy derivatives and perform a simultaneous fitting of area and perimeter law for $\log\langle \hat{W}_C \rangle$ with different h. The relatively sharp changes of the data at around $h = 0.3$ suggest a deconfinement/confinement phase transition, which is consistent with [113, 15].

3.6 Discussion and Future Directions

In this work, we have showed how to use gauge equivariant networks to represent variational states which exactly obey local gauge constraints. This significantly expands the space of models whose phase diagrams can now be numerically established. For example one could variationally explore the phase diagrams of the $\mathbb{Z}_d$ lattice gauge theory for $d > 2$ (see SM III); the 3D Toric-Code (SM IV); X-cube fracton model (SM IV) or Kitave $D(G)$ models (SM V) with external field; or models with disorder. Another interesting application would be to relax the restriction to the even sector of the gauge theory and explore the physics of different gauge sectors. Beyond the explicit constructions given here, it will be interesting to further extend the reach of such networks by generalizing the approach described in this work to models with different gauge symmetries or constraints from gauging subsystem symmetries. Finally, the ability to exactly employ gauge constraints variationally has potential to have impact beyond ground state calculations for example in overcoming obstacles in optimizing combinatorial structures [114] or in successful quantum-state tomography [115].

3.7 Supplemental Materials

3.7.1 Detailed Description of the $\mathbb{Z}_2$ Gauge Theory Hilbert Space

The lattice underlying the gauge theory is chosen to be the $L \times L$ square lattice with periodic boundary conditions and topology of the torus. Let $G = (V, E)$ denote the undirected interaction graph with $|V| = L^2$ vertices and $|E| = 2L^2$ edges and $|F| = L^2$ faces, in accordance with Euler's formula $|V|-|E|+|F| = 2-2g$ for a torus, which has genus $g = 1$. Vertices, edges and faces are indexed by $v \in V$, $e \in E$, and $f \in F$ respectively. Each edge hosts a qubit $\mathbb{C}^2 = \text{span}_{\mathbb{C}}\{|-1\rangle, |1\rangle\}$ with basis vectors labeled by the eigenvalues[1] of the Pauli-Z operator $Z|\pm 1\rangle = \pm|\pm 1\rangle$, so the joint tensor product Hilbert space for all qubits is $\mathcal{H} = \bigotimes_{e\in E} \mathbb{C}^2$ with orthonormal basis elements $|x\rangle := \bigotimes_{e\in E} |x_e\rangle$ where $x_e \in \mathbb{Z}_2$. It will be convenient to define the superposition state $|+\rangle = \frac{1}{\sqrt{2}}(|-1\rangle + |1\rangle) \in \mathbb{C}^2$. Cardinalities of sets are omitted when they appear in superscripts so, for example, $x \in \mathbb{Z}_2^E$ denotes a ± 1 string of length $|E|$, indexed by the edges of the graph G.

3.7.2 Exact Representation of $\mathbb{Z}_2$ Toric Code Ground States

In this section, we provide exact constructions of gauge equivariant neural networks for the ground states and the excited state of the $\mathbb{Z}_2$ toric code model. This can be done with a neural network with no equivariant

[1] Please note that this convention differs from the quantum information literature.

blocks and a single invariant block. It differs from our construction in the main text in two additional ways: (1) we use a fully connected network instead of a convolution network and (2) in addition to all the primitive plaquettes we include two topologically non-trivial loops around the torus.

There are four degenerate ground states of the $\mathbb{Z}_2$ model. One exact ground state can be obtained by starting with the uniform superposition state $|+\rangle^{\otimes E}$ and applying the following projection operator which commutes with all vertex operators,

$$P := \prod_{f \in F} \frac{\mathbb{1} + B_f}{2} \quad . \tag{3.12}$$

which gives rise to the loop-gas ground state $|\psi_{\text{loop-gas}}\rangle = P|+\rangle^{\otimes E}$. Excited states are obtained from the ground state by relaxing eigenvalues of various plaquette operators B_f to -1, in a manner consistent with the global operator constraint $\prod_{f \in F} B_f = \mathbb{1}$. Because the toric code is frustration free, to be a ground state of Eq. 3.1, it suffices to be a ground state of B_f and X_e separately. As the gauge equivariant construction already ensures the system is a ground state of X_e we simply choose a function of the elementary Wilson loops $W_f(\phi)$ for each plaquette $f \in F$

$$f(\phi) = h_e\big(W_{C_1}(\phi), \ldots, W_{C_{L^2}}(\phi), W_{C_x}(\phi), W_{C_y}(\phi)\big) \tag{3.13}$$

for our network where each C_i is the elementary plaquette $\square$ for $i = 1, ..., L^2$, C_x is a loop across the whole torus from x-direction and C_y is a loop across the whole torus from y-direction.

We must choose a h_e such that Eq. 3.13 is a ground state for each B_f, i.e. $\langle x|B_f|\psi\rangle = \langle x|\psi\rangle$ for each B_f and x. It implies that $\langle x|\psi\rangle = 0$ for any $|x\rangle$ such that $B_f|x\rangle = -|x\rangle$. Consider two operators $\tau_z^x = \prod_{i \in C_x^e} Z_i$ and $\tau_z^y = \prod_{i \in C_y^e} Z_i$. τ_z^x and τ_z^y further distinguishes the four degenerate ground state, where the expectation values a, b of the two operators are from one of the choices in $\{\pm 1, \pm 1\}$. Hence, we define h_e as follows

$$h_e\big(W_{C_1}(\phi), \ldots, W_{C_n}(\phi), W_{C_x}(\phi), W_{C_y}(\phi)\big) = \prod_i h_1(W_{C_i}(\phi)) h_x(W_{C_x}(\phi)) h_y(W_{C_y}(\phi)) \tag{3.14}$$

where $h_1(W_{C_i}(\phi)) = 1$ for $W_{C_i}(\phi) = 1$ and zero otherwise, $h_x(W_{C_x}(\phi)) = 1$ for $W_{C_x}(\phi) = a$ and zero otherwise, $h_y(W_{C_y}(\phi)) = 1$ for $W_{C_y}(\phi) = b$ and zero otherwise. The above construction provides an exact representation of a one-layer gauge invariant network for each of the ground state of $\mathbb{Z}_2$ toric code.

3.7.3 Gauge Equivariant Construction for $\mathbb{Z}_d$ Toric Code

In this section we briefly review the generalization to the $\mathbb{Z}_d$ gauge group [116]. Consider the usual $L \times L$ square periodic lattice where each edge hosts a d-dimensional qudit. The Hamiltonian defined on $\mathcal{H}(\mathbb{C}^d)^{\otimes E}$

is

$$H = -\sum_{v\in V}\sum_{h\in\mathbb{Z}_d}(A_v)^h - \sum_{f\in F}\sum_{h\in\mathbb{Z}_d}(B_f)^h \tag{3.15}$$

where A_v and B_f are shown in Fig. 3.6 with

$$X = \sum_{h\in\mathbb{Z}_d}|h+1\rangle\langle h| \ , \qquad Z = \sum_{h\in\mathbb{Z}_d}\omega^h|h\rangle\langle h| \ , \qquad \omega = e^{i2\pi/d} \tag{3.16}$$

To discuss the gauge equivariant and invariant construction for the $\mathbb{Z}_d$, we represent the basis for qudit on each edge as $\{e^{i2k\pi/d}|k = 0,1,...,d-1\}$. The action of the gauge group on the space of $(L,L,2)$ tensors is described as follows. Given a square matrix $\Omega \in \{e^{i2k\pi/d}|k = 0,1,...,d-1\}^V := \{e^{i2k\pi/d}|k = 0,1,...,d-1\}^{L\times L}$, we define a gauge transformation $g_\Omega : \mathbb{C}^E \to \mathbb{C}^E$ by the following rule,

$$(g_\Omega\cdot\phi)_\mu(v) := \Omega(v)\phi_\mu(v)\Omega(v+\mu)^* \ , \tag{3.17}$$

where $\Omega(v)$ and $\Omega(v+\mu)$ denote the entries of the matrix Ω at the lattice location $v \in V$ and the shifted lattice location $v+\mu \in V$, assuming boundaries are periodically identified.

It is straightforward to show that the gauge transformation associated with Hilbert space operator $(A_v)^h$ is given by $g_{\Omega_v^h}$ where Ω_v^h is defined for each $v' \in V$ by,

$$\Omega_v^h(v') = \begin{cases} e^{i2\pi h/d}, & v' = v \\ 1, & v' \neq v \end{cases} . \tag{3.18}$$

Given vertices $v, v' \in V$ and a directional path $\gamma \subseteq E$ from v to v', consisting of a sequence of adjacent edges with arrow direction, we define the Wilson path as the function $W_\gamma : \mathbb{C}^E \to \mathbb{C}$ given by the formula,

$$W_\gamma(\phi) := \overrightarrow{\prod_{e\in\gamma}}\phi_e \ . \tag{3.19}$$

where the product is directional such that it takes original value ϕ when the path γ direction on the edge goes upward and right and conjugate value ϕ^* when the path γ direction on the edge goes down and left.

An exact ground state can be found starting with the uniform superposition state $\left(\frac{1}{\sqrt{d}}\sum_{h\in\mathbb{Z}_d}|h\rangle\right)^{\otimes E}$ and applying the projection operator $P = \prod_{f\in F}\frac{1}{d}\sum_{h\in\mathbb{Z}_d}(B_f)^h$. It can be shown that the amplitudes in the

standard basis are of this state are computed by the following single-layer gauge-invariant neural network,

$$f(\phi) = \prod_{f \in F} \sum_{h \in \mathbb{Z}_d} W_f(\phi)^h \ . \tag{3.20}$$

The invariant and equivariant features consist of Wilson paths corresponding to closed loops and open paths ending on a given edge. The equivariance follows from the identity $W_\gamma(g_\Omega \cdot \phi) = \Omega(v) W_\gamma(\phi) \Omega(v')^*$.

3.7.4 Gauge Equivariant Construction for 3D Toric Code and X-cube Fracton Model

In this section, we generalize the gauge equivariant neural network construction to the 3D toric code and X-cube fracton model. The Hamiltonian of 3D toric code is given by,

$$H = -\sum_{v \in V} A_v - \sum_{f \in F} B_f \tag{3.21}$$

where $A_v := \prod_{e \ni v} X_e$ is the product of Pauli-X over the six edges incident on a vertex v and $B_f := \prod_{e \in f} Z_e$ is the product of Pauli-Z over each square face $\square$ of the lattice. The Gauss law constraint can be imposed using a gauge-equivariant wavefunction in which the invariant and equivariant features are chosen to be Wilson paths corresponding to closed loops and open paths ending on a given edge. The Wilson path function is the same form given in Eq. 3.7 in the main text.

The X-cube fracton model is defined on the same cubic lattice and the Hamiltonian is given by

$$H = -\sum_{v \in V, i} A_v^i - \sum_{c \in C} B_c \tag{3.22}$$

where each of the terms are defined in Fig. 3.7. The Gauss law constraint can again be imposed using a gauge-equivariant wavefunction. Invariant features can be constructed from Wilson paths evaluated on the star-shaped curves shown in Fig. 3.7. Equivariant features on an edge $v'v$ can be constructed from Wilson paths evaluated on the curves shown in Fig. 3.8.

The ground state wavefunction amplitudes of 3D toric code can be represented exactly by a gauge invariant neural network of the form (3.13) where the closed Wilson loops correspond to each square face $\square$ of the cubes. The ground state wavefunction amplitudes of the X-cube model can likewise be represented as a product of Heaviside step functions of Wilson paths, where now the Wilson paths correspond to each star-shape curves in x-y, y-z and x-z direction of the cube as shown in Fig. 3.7.

3.7.5 Gauge Equivariant Construction for Kitaev $D(G)$ Models

In this section we provide gauge equivariant neural network construction for general Kitaev Models over $\mathbb{Z}_d$ group and non-abelian groups. Consider the usual $L \times L$ square periodic lattice where each edge has a basis $\{|g\rangle, g \in G\}$ for certain group G. We focus on finite group here and group $G = \mathbb{Z}_d$ for $\mathbb{Z}_d$ theory. Without loss of generality, we attach an upward arrow for each edge in y-direction a right arrow for each edge in x-direction. We introduce operators A_v^g and B_f as follows

The Hamiltonian defined on $\mathcal{H}(G)^{\otimes E}$ is

$$H = -\sum_{v \in V} A_v - \sum_{f \in F} B_f \tag{3.23}$$

where $A_v = \frac{1}{|G|} \sum_{g \in G} A_v^g$ is the Gauss' law and the gauge constraint.

Define $|\circlearrowleft\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle$, then the ground state $|\psi\rangle = \prod_{f \in F} B_f |\circlearrowleft\rangle^{\otimes E}$. This is because $|\psi\rangle$ is both ground state for each A_v and B_f. It is easy to verify that $B_f|\psi\rangle = |\psi\rangle$. To see $A_v|\psi\rangle = |\psi\rangle$, notice that A_v and B_f commute with each other and $A_v|\circlearrowleft\rangle^{\otimes E} = |\circlearrowleft\rangle^{\otimes E}$.

Given a configuration x on the lattice, vertices $v_1, v_2 \in V$ and a directional path $\gamma \subseteq E$ from v_1 to v_2, consisting of a sequence of adjacent edges with direction, we define the generalized Wilson path as the function $W_\gamma : G^E \to G^E$ given by the formula,

$$W_\gamma(\phi) := \overrightarrow{\prod_{e \in \gamma}} \phi_e \ . \tag{3.24}$$

where the product is directional such that it takes original value g on the edge when the path γ direction agrees with the edge direction and inverse value g^{-1} when the path γ direction is opposite to the edge direction.

We claim that for any open path γ, W_γ is gauge equivariant, i.e. W_γ commutes with A_v^g for each v and g. To see this, we consider three cases. The first case is that A_v^g does not act on any edge in the path γ and it is clear that they commute. The second case is that A_v^g acts on a vertex along the path γ but not v_1, v_2. Notice that A_v^g must touch two adjacent edges at the same time and due to the arrow convention, the effect of A_v^g will also cancel out in the product. The last case is that A_v^g is on one of the vertex v_1, v_2 of the path γ, it is straightforward to verify directly A_v^g commutes with W_γ. Without loss of generality, we consider the a Wilson path function on the bottom edge of a plaquette with path γ going clockwise. One can check the following

For a closed loop C, we also introduce a Wilson loop function $W_C : G^E \to \mathbb{C}$

$$W_C(\phi) := \mathrm{tr}\overrightarrow{\prod_{e\in\gamma}}\phi_e \ . \tag{3.25}$$

Notice that W_C is gauge invariant, i.e. $W_C \circ A_v^g(\phi) = W_C(\phi)$ for any v and g. This is because any A_v^g touches either no edge or two adjacent edges along C. The invariance property holds clearly if no edge is touched and still holds if two edges are touched since the effect of A_v^g always cancel out due to the arrow convention. Without loss of generality, this can be verified on each plaquette as follows

With W_γ and W_C, we can construct gauge equivariant and invariant layer for the gauge equivariant neural network with Eq. 3.26 and Eq. 3.27 for general Kitave model with both $\mathbb{Z}_d$ and non-abelian group G. A gauge-equivariant function $f : \mathbb{C}^E \to \mathbb{C}^E$ is defined for all $\phi \in \mathbb{C}^E$ by the rule $f : \phi_e \mapsto f_e(\phi)$, where

$$f_e(\phi) := W_{\gamma_e}(\phi) h_e\big(W_{C_1^e}(\phi), \ldots, W_{C_{n_e}^e}(\phi)\big) \ . \tag{3.26}$$

and a gauge-invariant function $h : \mathbb{C}^E \to \mathbb{C}$ is defined for all $\phi \in \mathbb{C}^E$ by the rule $h : \phi_e \mapsto h_e(\phi)$, where

$$h_e(\phi) := h_e\big(W_{C_1^e}(\phi), \ldots, W_{C_{n_e}^e}(\phi)\big) \ . \tag{3.27}$$

In particular, we can construct a one-layer gauge invariant network for the ground state as follows

$$f(\phi) = \prod_i h_1(W_{C_i}(\phi)) \tag{3.28}$$

where each C_i is the elementary plaquette $\square$ for $i = 1, ..., L^2$, $h_1(W_{C_i}(\phi)) = 1$ for $W_{C_i}(\phi) = \mathrm{tr}(\mathbb{1})$ for the identity element $\mathbb{1} \in G$ and zero otherwise. This is because $A_v^g|\psi\rangle = |\psi\rangle$ for each v, g from the above discussion and so is $A_v|\psi\rangle = |\psi\rangle$. And $B_f|\psi\rangle = |\psi\rangle$ holds due to the fact that for any finite group $\mathrm{tr}(g) = \mathrm{tr}(\mathbb{1})$ implies that $g = \mathbb{1}$.

3.7.6 Neural Network Architecture and Numerical Details

In this section we provide the details for the architecture of the gauge equivariant neural network. There are two basic components for the network, which are the equivariant layer and the invariant layer. For the equivariant layer, we choose γ_e to be $\sqcap$ for horizontal edge and $\sqsubset$ for vertical edge and all C_i^e to be $\square$ in Eq. 3.9. The function h_e is taken to be convolutional neural network (CNN)and leaky relu activation function. One can further compose different equivariant layers in series or in parallel. For the invariant layer, it is expressed

as $f_e(\phi) = h_e\big(W_{C_1^e}(\phi), \ldots, W_{C_{n_e}^e}(\phi)\big)$, with all C_i^e to be $\square$ and h_e to be convolutional neural network. There are two outputs of h_e in the invariant layer, which parameterizes the log amplitude and the phase of the wave function separately. The elu activation is used for the log amplitude output while the softsign activation is used for the phase output. The full network is made of composition of equivariant layers (Fig. 3.1(b)) followed by an invariant layer (Fig. 3.1(c)) in the end. For all the $8 \times 8, 10 \times 10, 12 \times 12$ experiments, we use neural network architecture as Fig. 3.1 with kernel size of the CNN to be 4, 5, 5 respectively. For experiments in Fig. 3.2, the gauge equivariant neural network uses architecture in Fig. 3.1(a) with one channel of gauge equivariant blocks and the gauge invariant neural network uses architecture in Fig. 3.1(c), where the kernel size in all CNNs is 3. The RBM has hidden neurons three times as large as input neurons and outputs log amplitude and phase for the wave function.

Stochastic reconfiguration was performed with batch size 1000 and fixed learning rate 0.05. The number of iteration is 120 in general except that the experiments in 10×10 and 12×12 have iteration 150. The experiment in 12×12 starts with transfer learning of ansatzs optimized in 10×10. The convolution neural network parameters are initialized with orthogonal initialization [117]. For sampling, we adopt the standard Monte Carlo sampling with single spin flip.

3.7.7 Observables and Quantities across Phase Transition

We provide further figures (see Fig. 3.12, 3.13, 3.14, 3.15, 3.16) on observables and quantities across the phase transition.

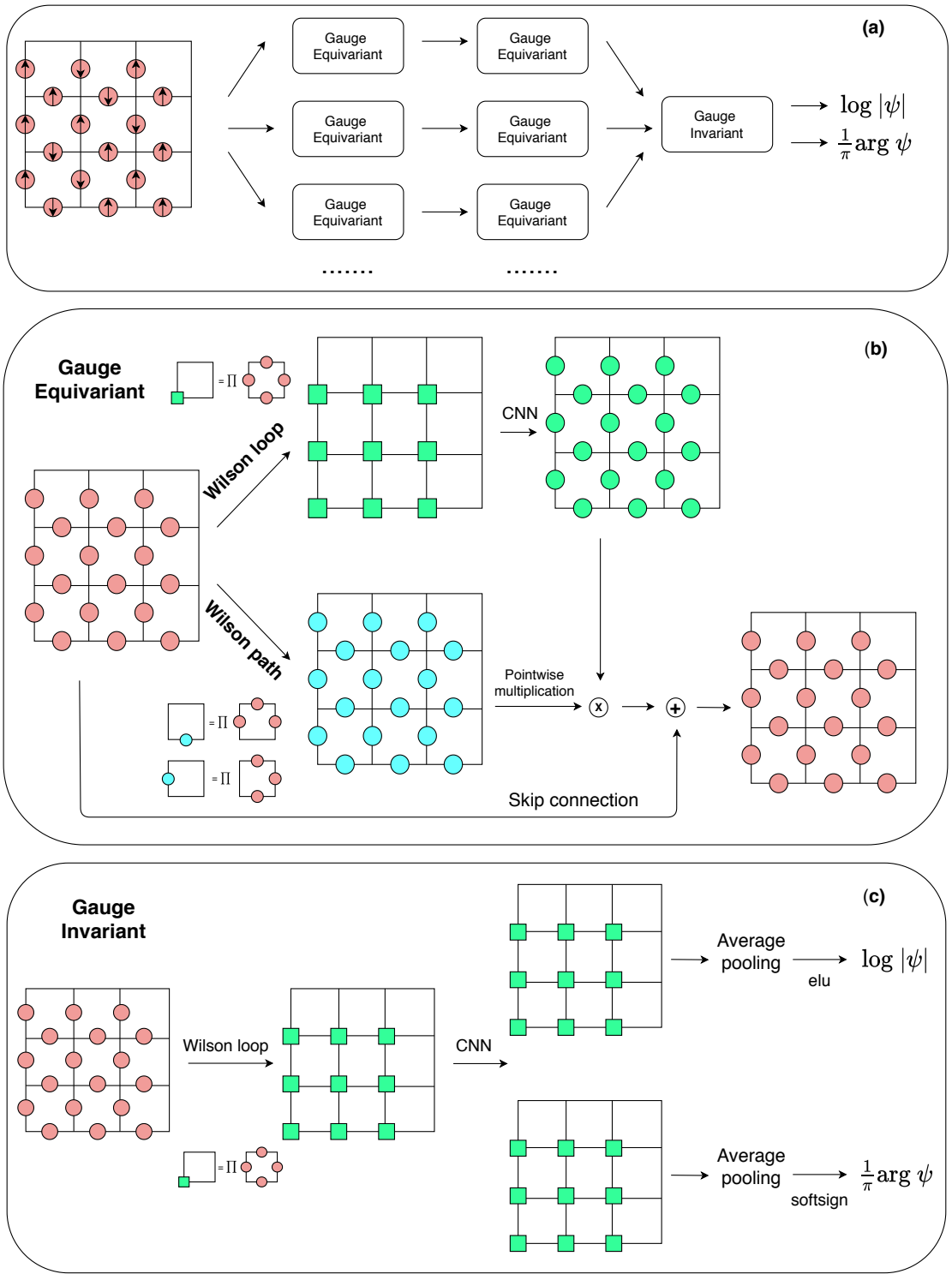


Figure 3.1: (a) Gauge equivariant neural network architecture. (b) Gauge equivariant block. (c) Gauge invariant block. For (b) and (c), the convolution neural network (CNN) component uses channel=2, stride=1, activation function=leaky relu and periodic boundary padding. The kernel size in CNN is different for different systems sizes.

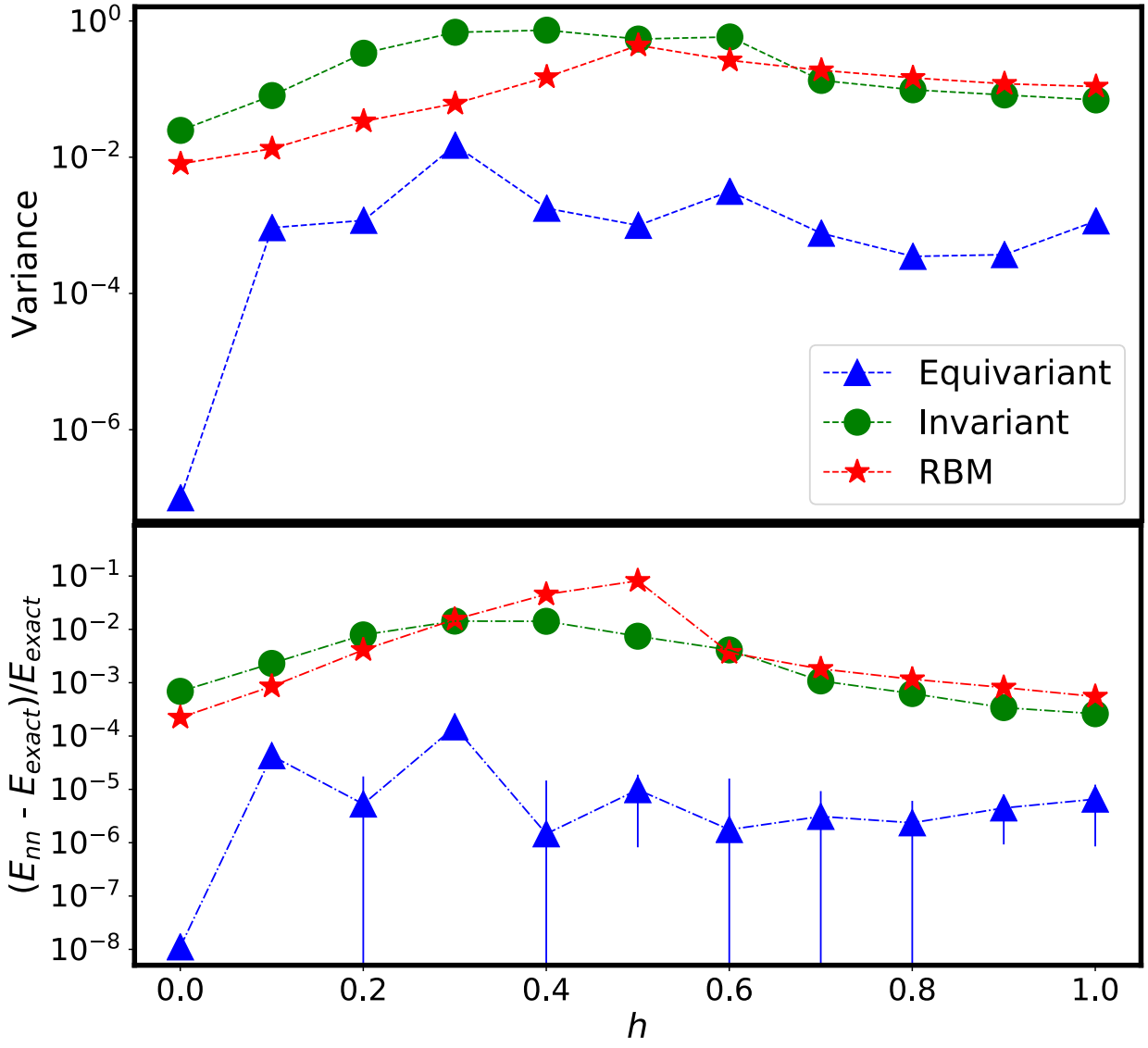


Figure 3.2: The variance **(top)** and the energy difference between the exact ground state **(bottom)** of the gauge equivariant network, gauge invariant network (no equivariant layers) and RBM on a 3×3 lattice of Eq. 3.1. The details of the above networks are provided in Sec. VI of the Supplemental Material.

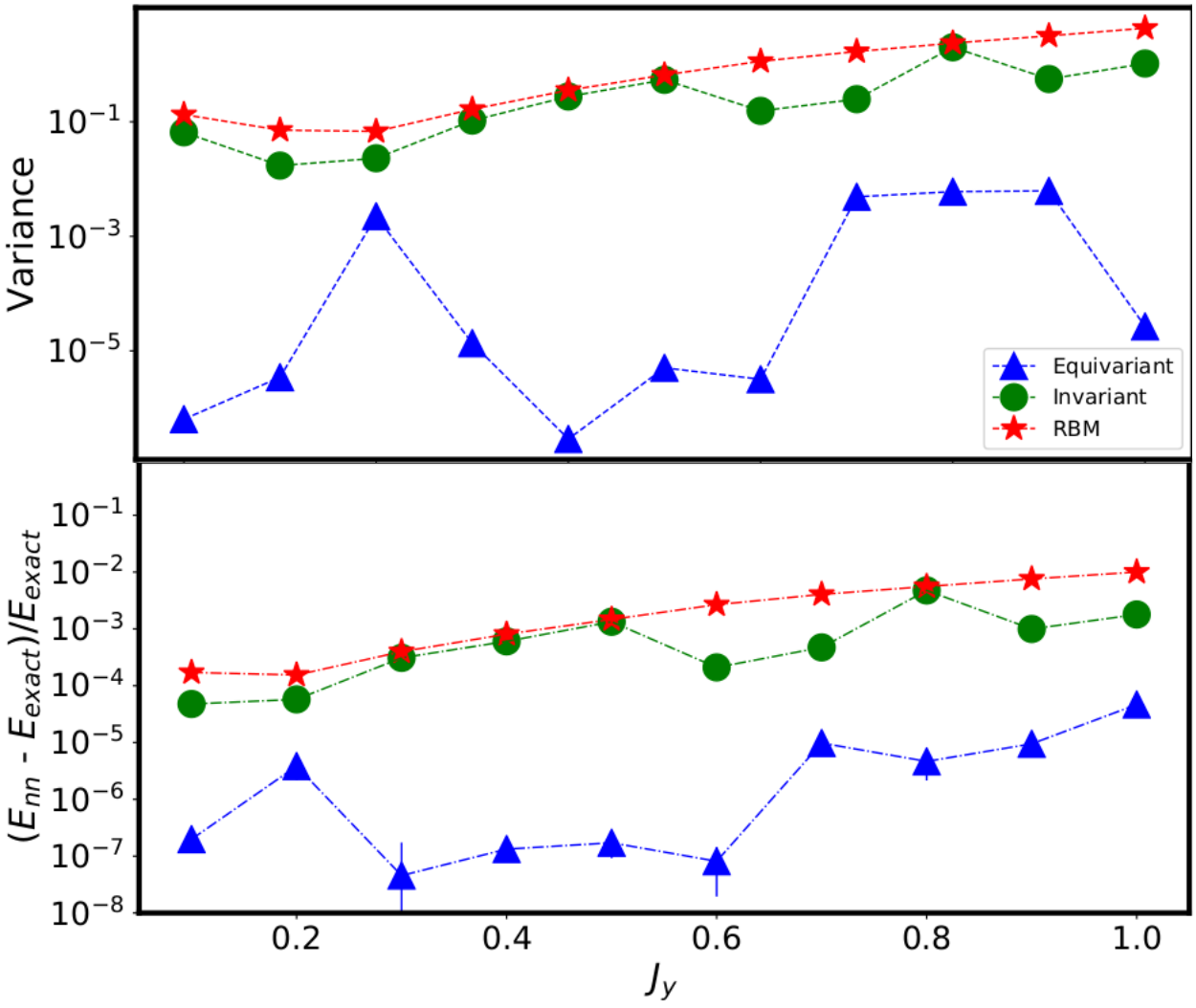


Figure 3.3: The variance **(top)** and the energy difference between the exact ground state **(bottom)** of the gauge equivariant network, gauge invariant network (no equivariant layers) and RBM on a 3×3 lattice of $H = -\sum_{f \in F} \prod_{e \in f} Z_e - \sum_{e \in E} X_e \; - J_y \sum_{f \in F} \prod_{e \in f} Y_e$. The details of the above networks are the same as Fig. 3.2.

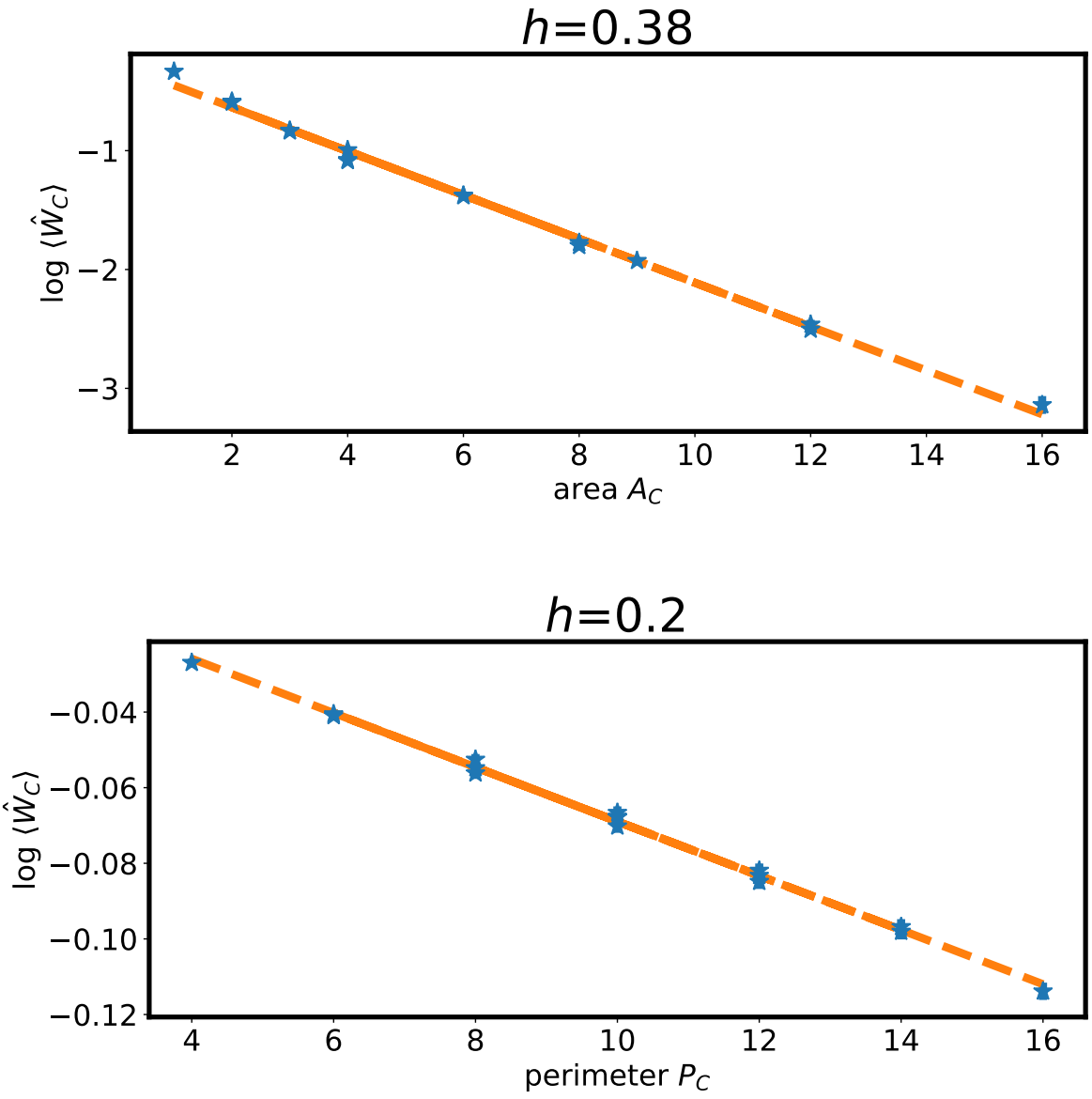


Figure 3.4: The ground state expectation value of rectangle Wilson loops of size $l_1 \times l_2$ ($l_1, l_2 \leq 4$) as a function of the enclosed area A_C in the confining phase for $h > h_c$ (**Top**) and of the enclosed perimeter P_C in the deconfined phase for $h < h_c$ (**Bottom**) on a 12×12 lattice. The linear fit to the log-linear plot is consistent with area law scaling $\langle \hat{W}_C \rangle \sim e^{-\alpha A_C}$ with best fit parameter $\alpha = 0.185$ and a perimeter law scaling $\langle \hat{W}_C \rangle \sim e^{-\alpha' P_C}$ with best fit parameter $\alpha' = 0.00718$.

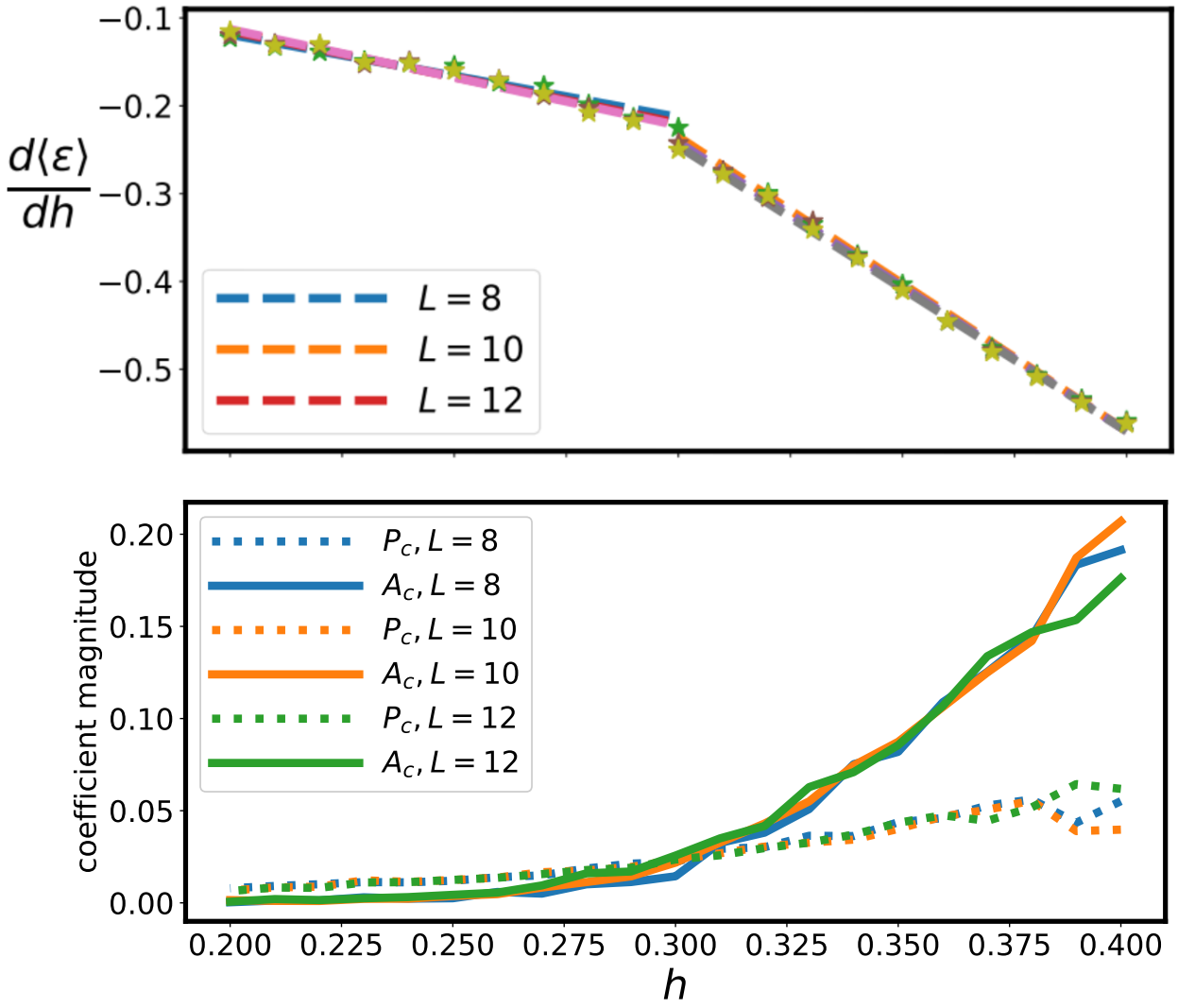


Figure 3.5: Based on the optimized gauge equivariant states on $L \times L$ lattices as a function of h, it shows (**Top**): derivative of the energy per site calculated by the Hellmann-Feynman theorem. The dotted lines are linear regression before and after $h = 0.30$. (**Bottom**): Magnitude of k_1 (solid) and k_2 (dotted) for fitting $\log\langle\hat{W}_C\rangle = k_1 A_C + k_2 P_C + b$ with area A_c and perimeter P_c. The discontinuity in the top figure and the relatively sharp increase in A_c in the bottom figure both at $h \approx 0.3$ suggests the confined/deconfined phase transition.

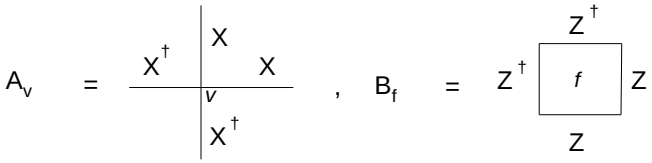


Figure 3.6: A_v and B_f in the $\mathbb{Z}_d$ model.

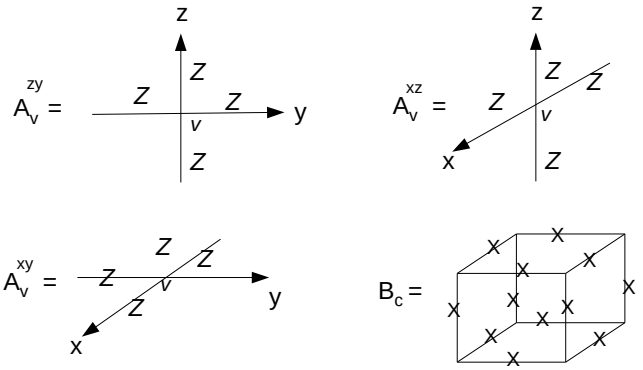


Figure 3.7: A_v^i and B_c in X-cube model.

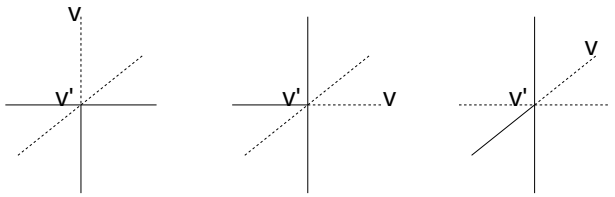


Figure 3.8: Equivariant features on an edge $v'v$ in the X-cube model can be constructed from curves described by solid lines shown.

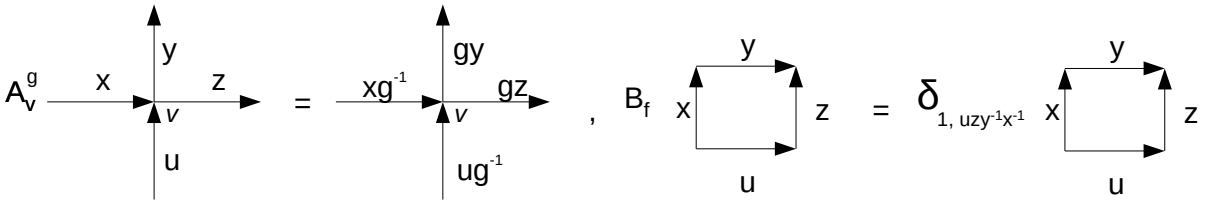


Figure 3.9: A_v^g and B_f operators.

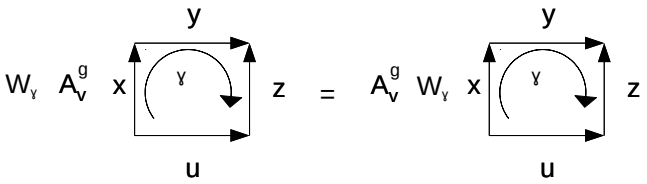


Figure 3.10: W_γ commutes with A_v^g.

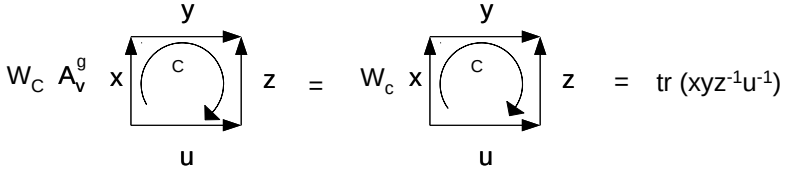


Figure 3.11: W_C is gauge invariant.

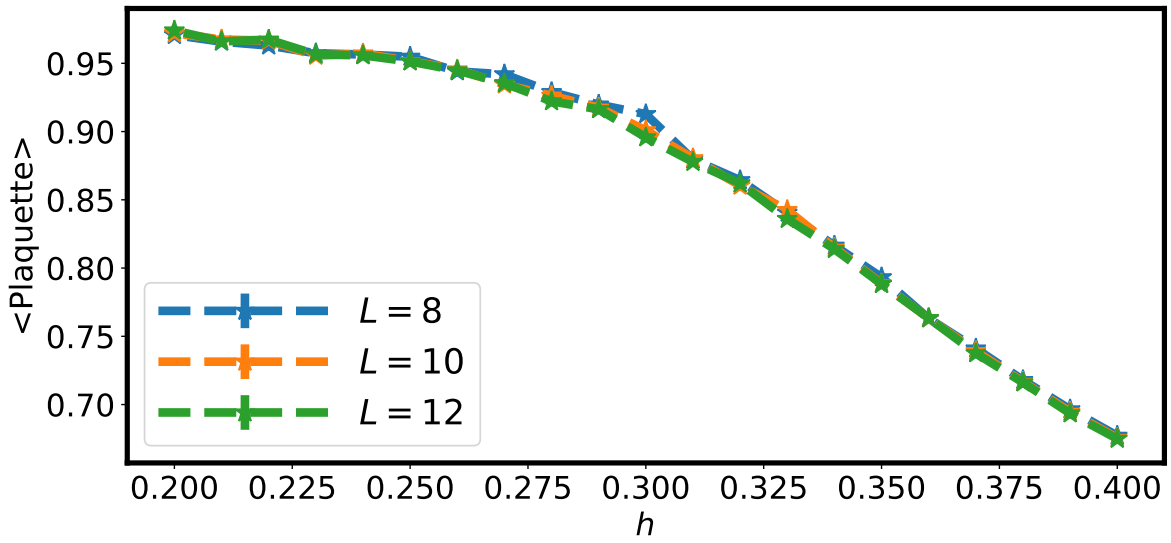


Figure 3.12: Average of expectation of plaquette $\langle \square \rangle$ over all plaquettes for different external field h on $L \times L$ lattices. The relatively sharp change of the slope at around $h = 0.3$ suggests a phase transition.

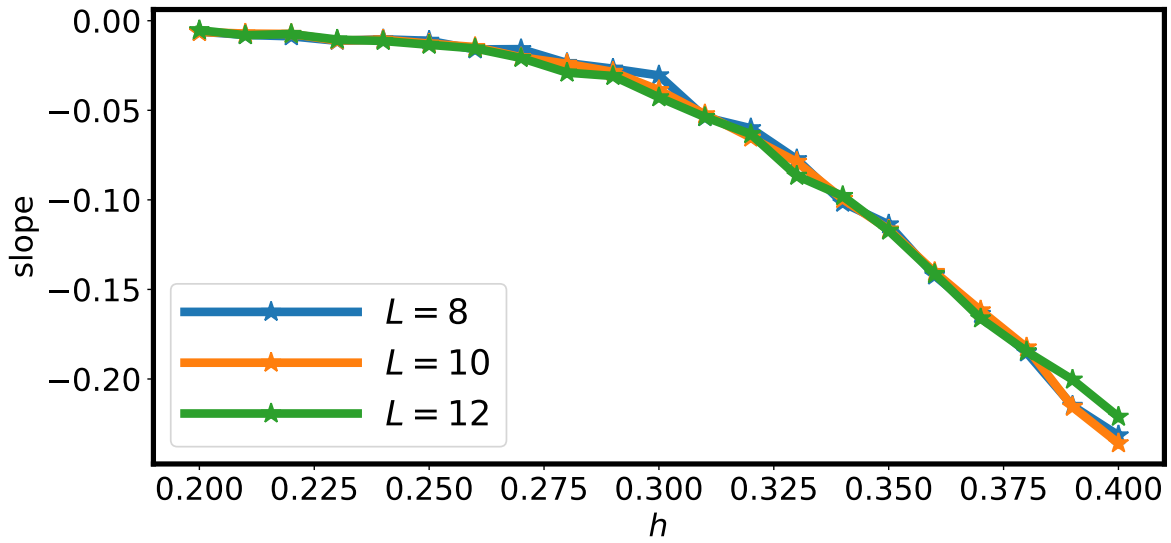


Figure 3.13: Slope k_1 of the area (A_C) law fitting in $\log\langle \hat{W}_C \rangle = k_1 A_C + b$ for different external field h on $L \times L$ lattices. The relatively sharp changes of the slopes at around $h = 0.3$ suggests a phase transition.

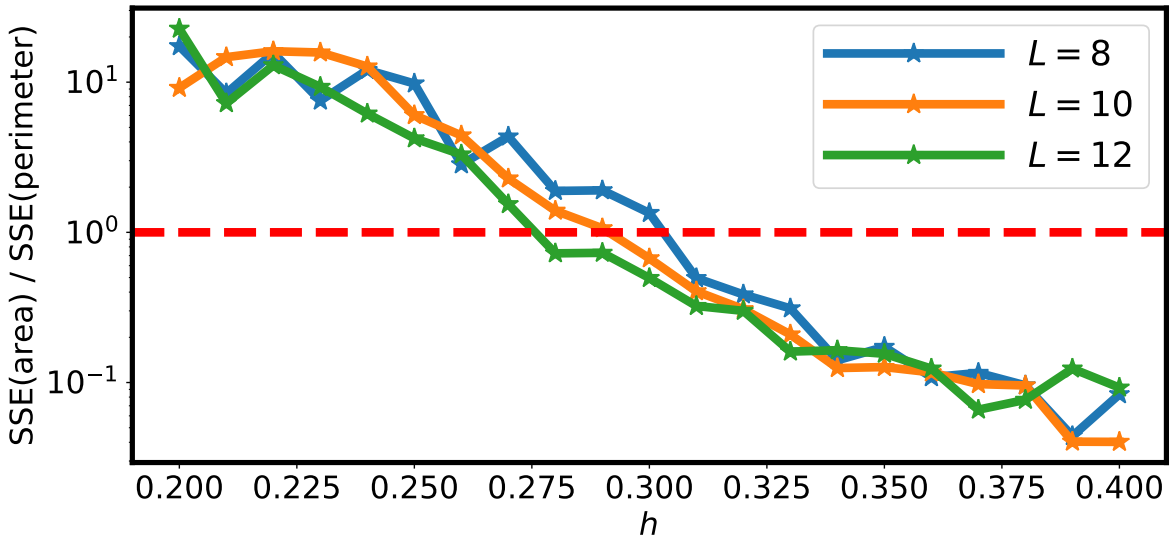


Figure 3.14: Ratio between the sum of squared estimate of errors (SSE) of the area law fitting of $\log\langle\hat{W}_C\rangle$ and the SSE of the perimeter fitting of $\log\langle\hat{W}_C\rangle$ for different external field h on $L \times L$ lattices.

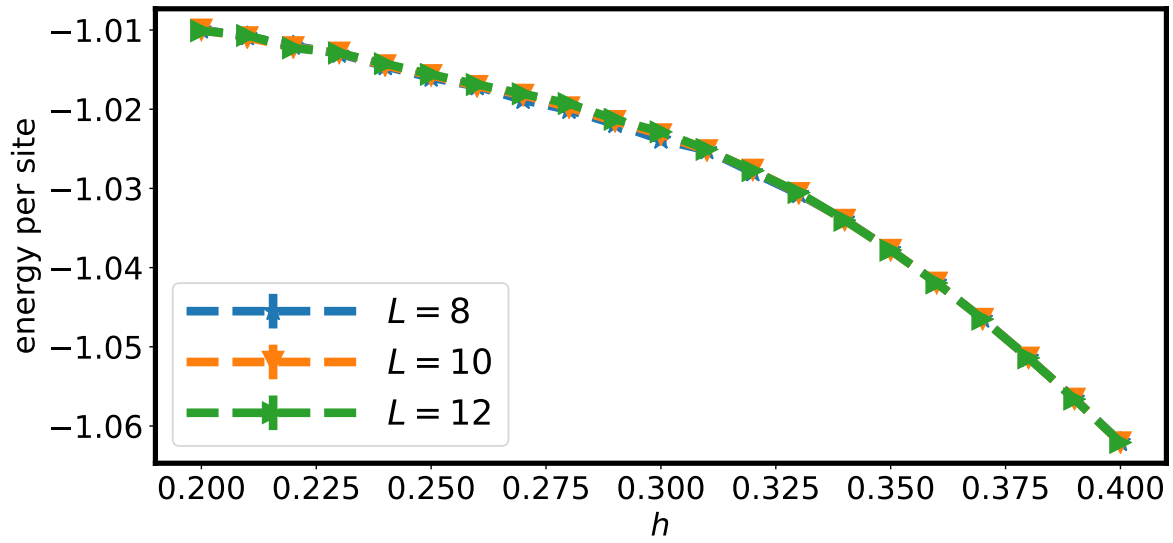


Figure 3.15: Energy per site for different external field h on $L \times L$ lattices.

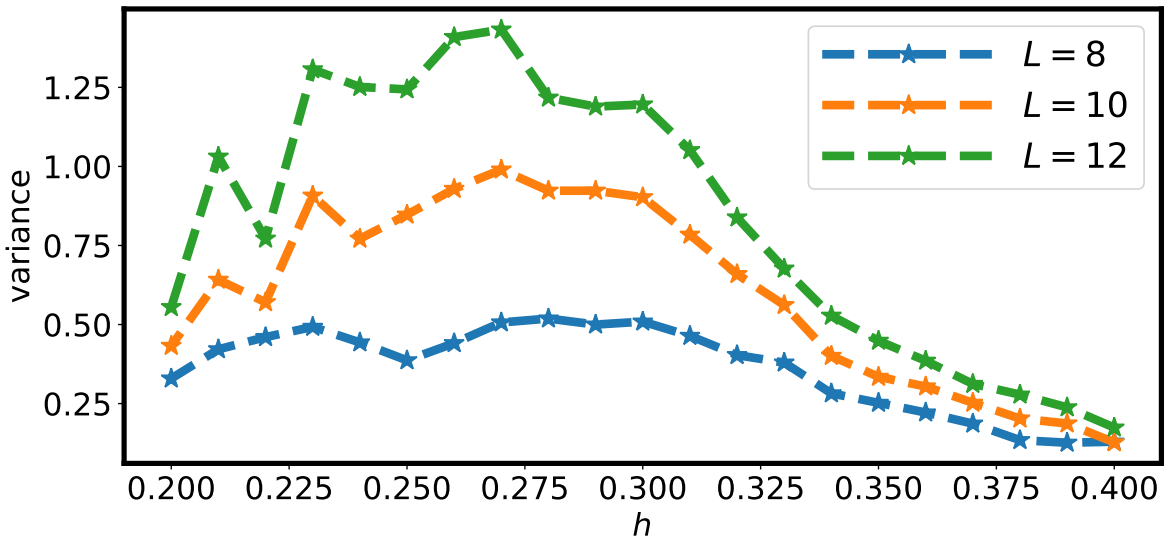


Figure 3.16: Variance for different external field h on $L \times L$ lattices.

Chapter 4

Gauge Invariant Autoregressive Neural Networks for Quantum Lattice Models

This chapter is written based on the research in Ref. [118]. Gauge invariance plays a crucial role in quantum mechanics from condensed matter physics to high energy physics. We develop an approach to constructing gauge invariant autoregressive neural networks for quantum lattice models. These networks can be efficiently sampled and explicitly obey gauge symmetries. We variationally optimize our gauge invariant autoregressive neural networks for ground states as well as real-time dynamics for a variety of models. We exactly represent the ground and excited states of the 2D and 3D toric codes, and the X-cube fracton model. We simulate the dynamics of the quantum link model of U(1) lattice gauge theory, obtain the phase diagram for the 2D $\mathbb{Z}_2$ gauge theory, determine the phase transition and the central charge of the $\mathrm{SU}(2)_3$ anyonic chain, and also compute the ground state energy of the SU(2) invariant Heisenberg spin chain. Our approach provides powerful tools for exploring condensed matter physics, high energy physics and quantum information science.

4.1 Introduction

In recent years, there has been a growing interest in machine learning approaches to simulating quantum many-body systems [119, 120, 121, 122, 17, 6, 94, 123, 124, 98, 99, 100, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135]. An important step in this direction is the use of neural networks, e.g. restricted Boltzmann machines, to represent variational wave functions. However, many neural networks do not automatically enforce the symmetries of physical models. A considerable amount of work has been devoted to remedy the deficiency for several classes of global symmetries, such as translational symmetry [121], discrete rotational symmetry [121], global U(1) symmetry [122], and anti-symmetry [17, 6, 94].

In addition to global symmetries, local symmetries can be encoded through gauge invariance. The notion of gauge invariance is crucial in quantum mechanics. In high energy physics, theory is required to be invariant under the action of gauge symmetry groups [88]. Gauge invariance appears naturally in various condensed matter physics models. For example, topological states of toric code and double semion models arise as the ground states of their gauge-invariant Hamiltonians [136, 116]. Also, novel quantum matter such as

fracton is the ground state of a Hamiltonian where the subsystem symmetry is gauged [137]. In quantum information, various quantum error correction codes can be viewed as eigenstates in a certain gauge-invariant code space [138]. The study of quantum lattice models with gauge symmetries is significant to enhance our understanding of high energy physics, condensed matter physics, and quantum information science.

Simulating quantum many-body gauge theory is exponentially costly. There has been much effort to efficiently simulate quantum lattice gauge theory with both digital and analog quantum computers [139], but more effort is required experimentally to achieve good fidelity. Two standard approaches to simulating gauge theory classically are stochastic, integrating an effective Lagrangian by sampling, and variational. When simulating gauge theory, the stochastic approach naturally obeys gauge invariance but is plagued with exponential costs associated with the sign problem in models with finite density of fermions or involving quantum dynamics [140]. The variational approach overcomes the difficulty by being constrained to an approximate variational space. Imposing gauge symmetries in the variational approach is particularly important and challenging as, otherwise, lower energy states can exist in the gauge-violating part of a Hilbert space. Therefore gauge symmetries must be explicitly constrained. While the stochastic approach has been well studied, there have been limited attempts at using the variational approach for gauge theory. Tensor networks can be readily applied to gauge theory in one dimension and ongoing efforts are required to work with challenges in higher dimensions [139, 141]. A variational approach based on gauge equivariant networks has been introduced very recently [83, 9, 108].

We develop for the first time, a general approach to constructing gauge invariant autoregressive neural networks (AR-NN) for quantum lattice models with gauge symmetries. We also extend our AR-NN to algebraic structures beyond gauge invariance, such as the fusion rule for anyons. Autoregressive neural networks, such as recurrent neural networks (RNN) [142, 143], pixel convolutional neural networks (pixelCNN) [144], and Transformers [145], have revolutionized the fields of computer vision and language translation and generation, among many others. Autoregressive neural networks quantum states have recently been introduced in quantum many-body physics [146, 122, 147] and shown to be capable of representing volume law states (as one generically needs in dynamics) with a number of parameters that scale sub-linearly [148]. A central feature of AR-NN is their capability of exactly sampling configurations from them. This is to be contrasted with the standard approach of sampling configurations by doing a random walk over a Markov chain, which is often plagued with long equilibration times and non-ergodic behaviors. We construct gauge invariant AR-NN for the quantum link model of U(1) lattice gauge theory [149], $\mathbb{Z}_N$ gauge theory, and $\mathrm{SU}(2)_k$ anyons. We demonstrate the exact representation of gauge invariant AR-NN for the ground and excited states of the 2D [116] and 3D [150] toric codes, and the X-cube fracton model [151]. We optimize our gauge invariant

AR-NN for the quantum link model, the 2D toric code in a transverse field, the 1D Heisenberg chain with SU(2) symmetry, and the $SU(2)_3$ anyonic chain [152, 153, 154], to obtain ground states accurately and extract phase diagrams and various dynamic properties.

4.2 Gauge Invariant Construction

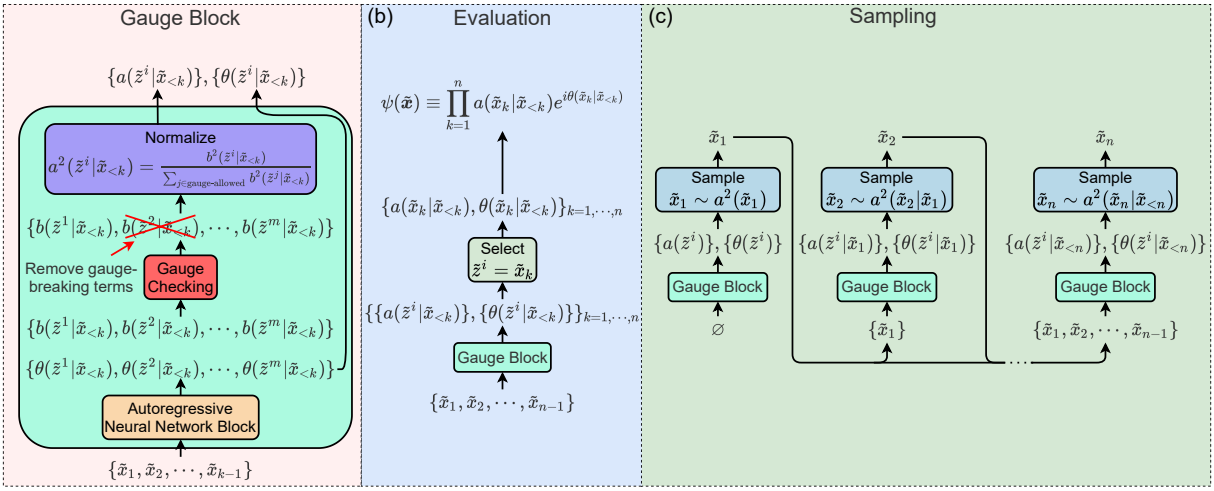


Figure 4.1: Autoregressive parameterization of wave function with n composite particles. (a) Gauge block. The input $\{\widetilde{x}_1, \widetilde{x}_1, \ldots, \widetilde{x}_{k-1}\}$ is processed through the autoregressive neural network block (see Supplementary Materials 4.6.4 for details), to output amplitude and phase parts. The amplitude part goes through gauge checking, which removes the gauge breaking terms. Afterwards, the square of the amplitude is normalized. (b) Evaluation process. The evaluation process can be performed in parallel for all the input sites. Given the input $\{\widetilde{x}_k\}$, the gauge block simultaneously generates amplitudes and phases for all sites. We then select the correct amplitudes and phases based on the input configuration for each site and construct the wave function from the selected amplitudes and phases. (c) Sampling process. The sampling is done sequentially for each site. We begin with no input and generate the amplitude and phase for the first site. The configuration of the first site is then sampled from the square of the amplitude. Afterwards, we feed the first sample into the gauge block to obtain the second sample. This process continues until we obtain the whole configuration.

Our goal in this work is to generate autoregressive neural networks (AR-NN) which variationally represent wave functions of quantum lattice models and explicitly obey their gauge symmetries—i.e. given a set of gauge symmetry operators $\{G_i\}$ with local support, we would like to construct a wave function $|\psi\rangle$ such that $G_i |\psi\rangle = |\psi\rangle$ for each i. To do this, we will work within the 'gauge basis' $\{|x\rangle\}$ which is diagonal in the gauge, $\langle x|\psi\rangle = \langle x|G_i|\psi\rangle$. A sufficient condition of gauge invariance of the wave function is to ensure that the gauge-violating basis elements $|x\rangle$ have zero amplitude in $|\psi\rangle$. Throughout this work, we will primarily work with gauges G_i which are local—i.e. $G_i |x\rangle$ only affects a compact range of sites within the vicinity of site i.

While we would typically want our AR-NN to take as input the configuration $\{x_1, x_2, \ldots, x_n\}$ and evaluate $\psi(x_1, x_2, \ldots, x_n)$, we will find it useful to instead evaluate $\psi(\widetilde{\boldsymbol{x}})$ where $\widetilde{\boldsymbol{x}} \equiv \{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_n\}$, $\widetilde{x}_i \equiv (x_{i_1}, x_{i_2}, \ldots, x_{i_v})$, is a composite particle specifying the configuration of not only site i but also some number of nearby sites. The motivation for working with composite particles is that a particular local gauge constraint G_i might only depend on composite particle $\widetilde{x}_i$ (and potentially $\widetilde{x}_{i+1}$), making it easier to apply the gauge constraints. Different composite particles can naturally overlap in physical sites and we will simply augment our gauge constraints to require that the configurations of the composite particles agree on the state of a physical site—i.e. basis states of composite particles which map to disagreeing physical states should also have zero amplitude.

AR-NN perform two functions: sampling and evaluation. AR-NN can sample configurations $\widetilde{\boldsymbol{x}}$ from $|\psi(\widetilde{\boldsymbol{x}})|^2$. This is done sequentially (in some pre-determined order) one composite particle $\widetilde{x}_i$ at a time; the probability to sample $\widetilde{x}_i$ is equal to $a^2(\widetilde{x}_i|\widetilde{x}_{<i})$ where $a(\widetilde{x}_i|\widetilde{x}_{<i})$ is a function which returns the conditional amplitude. Evaluation of the AR-NN gives a value $\psi(\widetilde{\boldsymbol{x}}) = \prod_{i=1}^{n} a(\widetilde{x}_i|\widetilde{x}_{<i}) e^{i\theta(\widetilde{x}_i|\widetilde{x}_{<i})}$ where $\theta(\widetilde{x}_i|\widetilde{x}_{<i})$ is a function which returns the conditional phase. Both evaluation and sampling rely on the existence of a gauge block which takes $\widetilde{x}_1, \ldots, \widetilde{x}_{k-1}$ and outputs the possible values $\{\widetilde{z}^i\}$ of $\widetilde{x}_k$ along with their respective amplitudes $a(\widetilde{z}^i|\widetilde{x}_{<k})$ and phases $\theta(\widetilde{z}^i|\widetilde{x}_{<k})$, ensuring that the amplitude of any configuration which is going to violate the gauge constraint is set to zero. To build this gauge block, we start with an autoregressive neural network block which returns a list of amplitudes which do not constrain the gauge (such blocks are standard in autoregressive models such as Transformers and RNN); we then zero out those partial configurations which break the gauge (on the already established composite particles) and renormalize the probabilities in this list (see Fig. 4.1(a)). Given the gauge block it is then straightforward to both sample and evaluate (see Fig. 4.1(b, c)). Note the probability induced by our AR-NN is different from the probability induced by the AR-NN with only the autoregressive neural network block even if one projects out the gauge-breaking configurations from the latter network.

It is worth noticing that the gauge invariant autoregressive neural networks are not limited to gauge theory, but can be generalized to wave functions with either local or global constraints which are checked in the same way as gauge constraints are checked. This will be helpful for describing constraints from certain global symmetries or special algebraic structure, such as the SU(2) symmetry for the Heisenberg model and the $\mathrm{SU}(2)_k$ fusion rules for non-abelian anyons.

4.3 Optimization Algorithms

We use AR-NN to calculate both ground states and real-time dynamic properties. In both cases, we need to optimize our AR-NN. For ground states, an AR-NN is optimized with respect to energy and for real-time dynamics, we optimize an AR-NN at time-step $t + 2\tau$ given a network at time t. We describe the details of these optimizations. As these optimization approaches are general, we use x to denote a configuration, but for the context of the paper, x should be viewed as a composite particle configuration.

For the ground state optimization, we stochastically minimize the expectation of energy for a Hamiltonian H and a wave function $|\psi_\theta\rangle$ as

$$\langle\psi_\theta|H|\psi_\theta\rangle \approx \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\frac{H\psi_\theta(x)}{\psi_\theta(x)} \equiv \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^{N} E_{\mathrm{loc}}(x), \tag{4.1}$$

where N is the batch size and the gradient is given by

$$\frac{\partial}{\partial\theta}\langle\psi_\theta|H|\psi_\theta\rangle \approx \frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\Re\left\{E_{\mathrm{loc}}(x)\frac{\partial}{\partial\theta}\log\psi_\theta^*(x)\right\}. \tag{4.2}$$

We further control the sampling variance [155] by subtracting from $E_{\mathrm{loc}}(x)$ the average over the batch, $E_{\mathrm{avg}} \equiv 1/N\sum_{x\in\mathrm{batch}} E_{\mathrm{loc}}(x)$, and define the stochastic variance reduced loss function as

$$\mathcal{L}_g = \frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\Re\Big\{\big[E_{\mathrm{loc}}(x) - E_{\mathrm{avg}}\big]\log\psi_\theta^*(x)\Big\}, \tag{4.3}$$

where the gradient is taken on $\log\psi_\theta^*$ using PyTorch's [156] automatic differentiation.

With this loss function, we also use transfer learning techniques [157, 158]. We train our neural networks in smaller systems and use these parameters as the initial starting points for optimizing for larger systems (see Supplementary Materials 4.6.4 for details).

For the dynamics optimization, we use a stochastic version of the logarithmic forward-backward trapezoid method [159], which can be viewed as a higher order generalization of IT-SWO [160] and the logarithmic version of the loss functions in Refs. [146, 124]. We initialize two copies of the neural network $\psi_{\theta(t)}$ and $\psi_{\theta(t+2\tau)}$. At each time step, we train $\psi_{\theta(t+2\tau)}$ to match $(1 + iH\tau)\left|\psi_{\theta(t+2\tau)}\right\rangle \equiv |\Psi_\theta\rangle$ and $(1 - iH\tau)\left|\psi_{\theta(t)}\right\rangle \equiv |\Phi\rangle$ by minimizing the negative logarithm of the overlap, $-\log\left(\langle\Psi_\theta|\Phi\rangle\langle\Phi|\Psi_\theta\rangle\right)/\left(\langle\Psi_\theta|\Psi_\theta\rangle\langle\Phi|\Phi\rangle\right)$. (Since we only take the gradient on $\theta(t + 2\tau)$, for simplicity, we write θ for $\theta(t + 2\tau)$ and neglect $\theta(t)$.) The inner

products related to θ can be evaluated stochastically as

$$\langle\Psi_\theta|\Phi\rangle \approx \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\frac{\Psi_\theta^*(x)\Phi(x)}{|\psi_\theta(x)|^2} \equiv \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\alpha(x), \tag{4.4}$$

$$\langle\Psi_\theta|\Psi_\theta\rangle \approx \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\frac{|\Psi_\theta(x)|^2}{|\psi_\theta(x)|^2} \equiv \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\beta(x). \tag{4.5}$$

The gradient of the negative logarithm of the overlap can be evaluated stochastically as

$$\begin{aligned}&\frac{\partial}{\partial\theta}\left(-\log\frac{\langle\Psi_\theta|\Phi\rangle\langle\Phi|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle\langle\Phi|\Phi\rangle}\right)\\ &\approx\frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\Re\left\{\left[\frac{\beta(x)}{\beta_{\text{avg}}}-\frac{\alpha(x)}{\alpha_{\text{avg}}}\right]\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x)\right\},\end{aligned} \tag{4.6}$$

where α_{avg} and β_{avg} are respectively the average values of $\alpha(x)$ and $\beta(x)$ over the batch of samples. We can then define the loss function as

$$\mathcal{L}_d \approx \frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\Re\left\{\left[\frac{\beta(x)}{\beta_{\text{avg}}}-\frac{\alpha(x)}{\alpha_{\text{avg}}}\right]\log\Psi_\theta^*(x)\right\}, \tag{4.7}$$

where the gradient is taken on $\log\Psi_\theta^*$ using PyTorch's [156] automatic differentiation.

For both optimizations, $\psi_\theta(x)$ is evaluated as described in Fig. 4.1(a) and x is sampled from $|\psi_\theta|^2$ as described in Fig. 4.1(b). The full derivations of the stochastic gradients for both optimizations are in Supplementary Materials 4.6.5.

4.4 Applications in Quantum Lattice Models

4.4.1 U(1) Quantum Link Model

The quantum link model (QLM) of U(1) lattice gauge theory in $1+1$ dimensions in the Hamiltonian formulation with staggered fermions [149] is defined as

$$\begin{aligned}H_{\text{QLM}} =& -\sum_i\left[\psi_i^\dagger U_{i,i+1}\psi_{i+1}+\psi_{i+1}^\dagger U_{i,i+1}^\dagger\psi_i\right]\\ &+m\sum_i(-1)^i\psi_i^\dagger\psi_i+\frac{g^2}{2}\sum_i E_{i,i+1}^2,\end{aligned} \tag{4.8}$$

where m is the staggered fermion mass, g is the gauge coupling, $i=1,2,\ldots$ labels the lattice site, ψ_i is the fermion operator, $U_{i,i+1}$ is the link variable and $E_{i,i+1}$ the electric flux for the U(1) gauge field on link

$(i, i+1)$ [149]. We denote by $|q_i\rangle$ the basis state at site i, and by $|e_{i,i+1}\rangle$ the basis at link $(i, i+1)$. Each unit cell is defined to include two sites and two links. The operators $E_{i,i+1}$ and $U_{i,i+1}$ satisfy the following commutation relations: $[E_{i,i+1}, U_{i,i+1}] = U_{i,i+1}$, $[E_{i,i+1}, U^\dagger_{i,i+1}] = -U^\dagger_{i,i+1}$ and $[U_{i,i+1}, U^\dagger_{i,i+1}] = 2E_{i,i+1}$. The gauge constraint is given by the Gauss's law operator $\widetilde{G}_i = \psi^\dagger_i \psi_i - E_{i,i+1} + E_{i-1,i} + \frac{1}{2}[(-1)^i - 1]$ such that the ground state $|\psi\rangle$ satisfies $\widetilde{G}_i |\psi\rangle = 0$ for each i. The QLM has gained growing interests and been studied in different settings in recent years [139, 141, 161, 162, 163]. We focus on the (1+1)D QLM with the $S = 1/2$ representation for the link operators $U_{i,i+1}$ and $E_{i,i+1}$. Under the Jordan-Wigner transformation, Eq. 9.1 becomes [16]

$$\begin{aligned} H = &- \sum_i \left[S^+_i S^+_{i,i+1} S^-_{i+1} + \text{H.c.} \right] \\ &+ m \sum_i (-1)^i (S^3_i + \frac{1}{2}) + \frac{g^2}{2} \sum_i \frac{1}{4}, \end{aligned} \tag{4.9}$$

where $S^\pm \equiv S^1 \pm iS^2$, S^1, S^2, S^3 are the Heisenberg matrices, and the Gauss's law operator becomes $G_i = S^3_i - S^3_{i,i+1} + S^3_{i-1,i} + \frac{1}{2}(-1)^i$. For the $S = 1/2$ representation, the last term on the right side of Eq. 4.9 is constant and, hence, can be discarded.

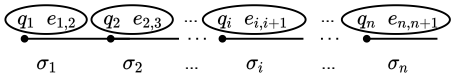


Figure 4.2: Composite particles for the quantum link model. Each composite particle is defined as $|\sigma_i\rangle \equiv |q_i, e_{i,i+1}\rangle$. We check Gauss's law between $|\sigma_i\rangle$ and $|\sigma_{i+1}\rangle$.

We define the composite particles of our gauge invariant AR-NN as in Fig. 4.2 and choose an order from left to right. Each composite particle $|\sigma_i\rangle$ consists of a fermion $|q_i\rangle$, which can be either $|\bullet\rangle$ or $|\circ\rangle$, and a gauge field in the link $|e_{i,i+1}\rangle$, which can be either $|\rightarrow\rangle$ or $|\leftarrow\rangle$. Note that in this case the composite particles do not overlap. The Gauss's law operator G_i acts on $|\sigma_i\rangle$ and $|\sigma_{i+1}\rangle$ to determine allowed configurations and so can be checked in the gauge block which generates the composite particle at site $i+1$. For example, given $|\sigma_i\rangle = |\bullet \rightarrow\rangle$, $|\sigma_{i+1}\rangle$ can only be $|\bullet \rightarrow\rangle$ or $|\circ \leftarrow\rangle$ if i is even, and $|\circ \rightarrow\rangle$ if i is odd.

We implement and variationally optimize this AR-NN for the ground state of Eq. 4.9. Fig. 4.3 shows the results for 6 unit cells (i.e. 12 particles) which closely match the energy of the exact solution. More importantly, the gauge invariant construction guarantees that the solution is in the physical space, while the neural network without gauge constraint (i.e. removing the gauge-checking from the AR-NN) finds a lower energy but non-physical state.

We also consider the real-time dynamics for $m = 0.1$ and $m = 2.0$ for 12 unit cells. Fig. 4.4(a) shows the conservation of energy for different ansatzes. The total energy is -1.2 for $m = 0.1$, and -24 for $m = 2.0$. Although the gauge invariant AR-NN exhibits jumps in energy at certain points due to optimization

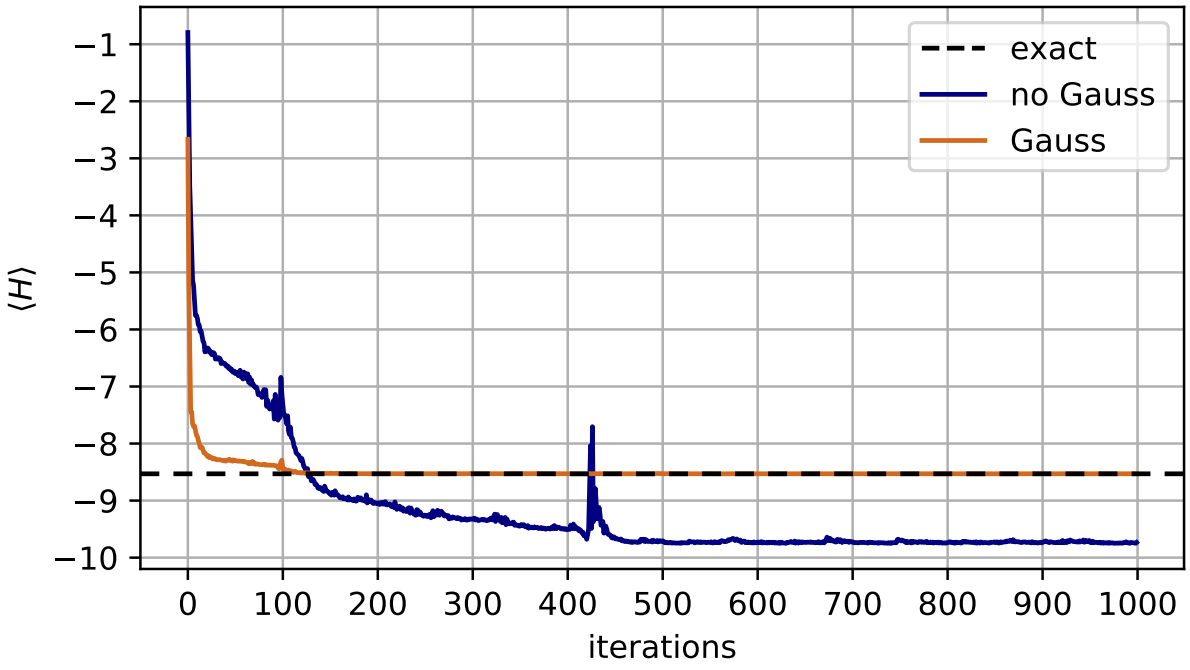


Figure 4.3: Variational ground state optimization for the 6-unit-cell (12 sites and 12 links) open-boundary QLM for $m = 0$ with and without gauge invariant construction. The gauge invariant autoregressive neural network reaches an accurate ground state while the ansatz without gauge constraints arrives at a non-physical state in the optimization. We use the Transformer neural network with 1 layer, 32 hidden dimensions and the real-imaginary parameterization (see Fig. 4.17). The neural network is randomly initialized and is trained for 1000 iterations with 12000 samples in each iteration. The neural network architecture and optimization details are discussed in Supplementary Materials 4.6.4.

instability, it preserves energy more smoothly in most steps compared with the non-gauge ansatz. We find that our gauge invariant AR-NN captures the correct electric field oscillation compared with the non-gauge ansatz (see Fig 4.4(b)) and, additionally, the anticipated string inversion of the electric flux for small mass (and respectively the static electric flux for large mass) (see Fig 4.5).

While the current work focuses on the $S = 1/2$ representation, our construction can be generalized to an arbitrary S representation. For a higher spin S, composite particles can be defined similarly (see Fig. 4.2) except that the degree of freedom for each $e_{i,i+1}$ increases to $2S + 1$ as S increases.

4.4.2 2D $\mathbb{Z}_N$ Gauge Theory

For the 2D toric code [116], consider an $L \times L$ periodic square lattice, where each edge has the basis $\{|0\rangle, |1\rangle\}$. Let V, P, E denote the sets of vertices, plaquettes and edges of the lattice, respectively, such that

$|V| = L^2, |P| = L^2, |E| = 2L^2$. Here we consider the toric code with a transverse magnetic field

$$H = H_{TC} - h \sum_{e \in E} \sigma_e^z, \tag{4.10}$$

where H_{TC} is the toric code Hamiltonian

$$H_{TC} = -\sum_{v \in V} A_v - \sum_{p \in P} B_p, \tag{4.11}$$

$A_v \equiv \prod_{e \ni v} \sigma_e^z$ (the star operator) , $B_p \equiv \prod_{e \ni p} \sigma_e^x$, and h is the strength of the transverse field. Note that A_v is the gauge constraint such that the ground state $|\psi\rangle$ of Eq. 4.10 and Eq. 4.11 satisfies $A_v |\psi\rangle = |\psi\rangle$ for each v.

The composite particle construction is illustrated in Fig. 4.6. We order our consecutive particles by an "S" shape going up one row and down the next (see Fig. 4.22(b) in Supplementary Materials 4.6.4). Two constraints must be checked in the gauge checking process of a gauge block.

When working on the gauge block associated with composite particle $\widetilde{x}_v$, we check that $A_v |\widetilde{x}_v\rangle = |\widetilde{x}_v\rangle$ (despite A_v acting on an entire state, this can be checked locally on a single composite particle). In addition, composite particles overlap with their four immediately adjacent composite particles. The gauge block for the composite particle at v therefore checks consistency of the physical sites with the first $v-1$ composite particles. For example, given the configuration $\left|{}^{0}_{0}{}^{1}_{1}\right\rangle$ for a composite particle, the composite particle to the right can only be $\left|{}^{0}_{0}{}^{1}_{1}\right\rangle$, $\left|{}^{1}_{0}{}^{0}_{1}\right\rangle$, $\left|{}^{0}_{1}{}^{0}_{1}\right\rangle$ or $\left|{}^{1}_{1}{}^{1}_{1}\right\rangle$, as the $|1\rangle$ on the right of the left particle must also be on the left of the right particle. In the "S" ordering, there always exists valid choices for each composite particle. For a composite particle that is not the last one, there is an unchosen site which provides freedom of choices to be valid. For the last composite particle, though all sites are fixed, the fixed configuration must be valid because the product of all the Gauss's law constraints is 1 and all previous Gauss's law constraints have been satisfied to be 1.

We begin by showing that we can analytically generate an AR-NN for the ground state of H_{TC}. One ground state of Eq. 4.11 is $|\psi\rangle = \prod_{v \in V} (\mathbb{1} + A_v) |+\rangle^{\otimes n}$ where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$—i.e. an equal superposition of all configurations in the gauge basis which do not violate the gauge constraint. In our construction, if we use an autoregressive neural network block which gives equal weight to all the configurations (this is straightforward to arrange by setting the last linear layer's weight matrices to zero and bias vectors to equal amplitudes), we exactly achieve this state. Checking the A_v does not affect the relative probabilities because it is not conditional involving only one composite particle. On the other hand, the 'gauge constraints' which verify consistency of the underlying state of the sites leave equal probability between all consistent

states. To see this, we examine the effect of the gauge constraint on $|a(\widetilde{x}_k|\widetilde{x}_{<k})|^2$ for any given k, which is the conditional probability of the composite particle $\widetilde{x}_k$. Due to the conditioning from previous composite particles $\{\widetilde{x}_{<k}\}$, some sites of the composite particle $\widetilde{x}_k$ are fixed. For the Gauss's law gauge constraints to be 1, the product of all the unchosen site configurations in $\widetilde{x}_k$ must be either 1 or -1, depending on the chosen site configurations. Let $S_1 = \{b_1, \ldots, b_j | \prod_{r=1}^{j} b_r = 1\}$ and $S_{-1} = \{c_1, \ldots, c_j | \prod_{r=1}^{j} c_r = -1\}$ be the two possible sets of unchosen site configurations, where b_r and c_r are the configurations of the unchosen sites in $\widetilde{x}_k$. Consider a function $f : S_1 \to S_{-1}$ such that $f(b_1) = -c_1$ and $f(b_r) = c_r$ otherwise. Notice that f is bijective and thus S_1 and S_{-1} have the same cardinality, implying that after normalization $|a(\widetilde{x}_k|\widetilde{x}_{<k})|^2$ will have the same amplitude for any $\{\widetilde{x}_{\leq k}\}$. We can also generate excited states by changing the A_v for a fixed (even) number of vertices to constrain this local eigenvalue to be -1 instead of 1. We provide a numerical verification of this by computing the energy for an exactly represented tower of ground and excited states in Fig. 4.7.

With a nonzero value of the external field h, the ground state of Eq. 4.10 is no longer exactly representable, and we variationally optimize our AR-NN to compute the ground state energy. Fig. 4.8 shows the minimum energy for Eq. 4.10 for different h and the energy derivative, computed using the Hellman-Feynman theorem [164]. The toric code is expected to exhibit a quantum phase transition between the topological and trivial phases at an intermediate value of h, and the sharp change of the energy derivative around $h = 0.34$ is an indicator of this phase transition, which is consistent with the theoretical prediction of $h \approx 0.32$ in the thermodynamic limit [15, 113]. We can additionally identify the transition by considering the Wilson loop operator $W_C = \prod_{e \in C} \sigma_e^x$ for a closed loop C. It is predicted that the topological order phase follows an area law decay, $\langle W_C \rangle \sim \exp(-\alpha A_C)$, and the trivial phase follows a perimeter law decay, $\langle W_C \rangle \sim \exp(-\beta P_C)$, where A_C, P_C are the enclosed area and perimeter of the loop C [112]. Fig. 4.9 shows the values of $\langle W_c \rangle$ using our variationally optimized AR-NN. By comparing the respective fits to the area and perimeter laws we again see the transition at $h = 0.34$. Finally, we compare the non-local string correlation operators $S_\gamma = \prod_{e \in \gamma} \sigma_e^z$ of our variational states which could be viewed as a measure of the correlation of a pair of excited particle and anti-particle along a path γ. In the topological order phase the non-local string operators will decay to zero while they will remain constant at the trivial phase [165]. In Fig. 4.10, this is seen clearly on both sides of the transition.

Our approach can be naturally generalized to 2D $\mathbb{Z}_N$ gauge theory, which can be described in the language of Kitaev's $D(G)$ model with group $G = \mathbb{Z}_N$ (see Supplementary Materials 4.6.1). In this case, the basis at each edge becomes a group element in $\mathbb{Z}_N$. Similarly to Fig. 4.6, one can define a composite particle over four edges from a vertex and impose gauge invariance. We can also extend our approach to the (1+1)D $\mathbb{Z}_N$

lattice quantum electrodynamics (QED) model, which is discussed in Supplementary Materials 4.6.2.

4.4.3 3D Toric Code and Fracton Model

We turn to gauge invariant AR-NN for the ground and excited states of the 3D toric code [150] and the fracton model [166, 151]. The 3D toric code generalizes the 2D toric code to an $L \times L \times L$ periodic cube where each edge has the basis $\{|0\rangle, |1\rangle\}$. The Hamiltonian takes the same form as the 2D model (see Eq. 4.11) except that for each $A_v \equiv \prod_{e \ni v} \sigma_e^z$ there are six edges e associated with each vertex v. A ground state of the 3D toric cube similarly satisfies $A_v |\psi\rangle = B_p |\psi\rangle = |\psi\rangle$ for each v, p. One of the degenerate ground states can also be expressed as $|\psi\rangle = \prod_{v \in V} (\mathbb{1} + A_v) |+\rangle^{\otimes n}$. The excited states can be generated by breaking certain constraints from A_v and B_p as in the 2D case.

The X-cube fracton model [151] is also defined on an $L \times L \times L$ periodic cube where each edge has the basis $\{|0\rangle, |1\rangle\}$. The Hamiltonian takes the following form

$$H_{\text{fracton}} = -\sum_{v \in V, i} A_v^i - \sum_{c \in C} B_c, \tag{4.12}$$

where $B_c \equiv \prod_{e \in c} \sigma_e^z$ over the edges in a small cube. The gauge constraint, i.e. Gauss's law, is $B_c|\psi\rangle = |\psi\rangle$. There are three $A_v^i \equiv \prod_{e^i \ni v} \sigma_{e^i}^x$ for three choices of $i = zy, xy, xz$, depending on which 2D plane A_v^i acts on. The operators are illustrated in Fig. 4.11. A ground state of the X-cube fracton model satisfies $A_v^i |\psi\rangle = B_c |\psi\rangle = |\psi\rangle$ for each i, v, c. One of the ground states can be expressed as $|\psi\rangle = \prod_c (\mathbb{1} + B_c)|+\rangle^{\otimes n}$. The excited states break some constraints such that $A_v^i |\psi\rangle = -|\psi\rangle$ or $B_c |\psi\rangle = -|\psi\rangle$ for certain A_v^i, B_c.

The composite particles for the 3D toric code and the X-cube fracton model are illustrated in Fig. 4.12. For the 3D toric code, a composite particle is made up of six particles associated with a vertex. The ground state can be constructed by initializing the bias of the autoregressive neural network to be all the $|+\rangle$ state and imposing gauge checking on each composite particle to be 1. The excited states can be constructed by forcing even numbers of composite particles to have gauge checking value -1. For the X-cube fracton model, a composite particle consists of twelve particles on each mini cube. The ground state comes from initializing all biases of the autoregressive neural network to be $|+\rangle$ and requiring all composite particles to have the gauge checking value 1. The excited states break the gauge checking value on sets of four nearby composite particles to be -1. We numerically verify the exact representations of ground and excited states of the 3D toric code and the X-cube fracton model in Fig. 4.7, where the energy is shown to be exactly the same as the theoretical predictions.

Our approach can be naturally generalized to the Haah's code fracton [166] and checkerboard fracton [151]

models. Similarly to 2D $\mathbb{Z}_N$ gauge theory (see Section 4.4.2), one can consider applying gauge invariant AR-NN to study the 3D $\mathbb{Z}_N$ gauge theory in the context of the 3D toric code or the X-cube fracton model [137] with an external field.

4.4.4 $\mathrm{SU}(2)_k$ Anyonic Chain and SU(2) Symmetry

Non-abelian anyons play a crucial role in universal topological quantum computation. Here we consider a chain of Fibonacci anyons, which can be regarded as an $\mathrm{SU}(2)_{k=3}$ deformation of the ordinary quantum spin-1/2 chain [154]. In this model, there is one type of anyon τ and a trivial vacuum state $\mathbb{1}$ for each site. The constraint from symmetry requires that τ and $\mathbb{1}$ satisfy the following fusion rule: $\tau \otimes \tau = \tau \oplus \mathbb{1}$, $\tau \otimes \mathbb{1} = \mathbb{1} \otimes \tau = \tau$. We work directly in this basis where each site is either $\mathbb{1}$ or τ, generating a gauge invariant AR-NN. We then proceed to work out the entire phase diagram of the Fibonacci anyons. This can be done particularly efficiently compared with standard Monte Carlo sampling [125] thanks to the exact sampling of the AR-NN.

Our gauge invariant AR-NN is constructed so that it obeys the anyon fusion rule directly by checking two adjacent input configurations and imposing zero amplitude when both are τ. Each anyon is a composite particle and the gauge checking implements the constraint from the anyon fusion rule.

We consider the Hamiltonian [153]

$$H(\theta) = -\cos\theta \sum_i H_i^{(2)} - \sin\theta \sum_i H_i^{(3)}, \tag{4.13}$$

where the two-anyon interactions can be described by the golden chain Hamiltonian [152, 153]

$$\begin{aligned} H_i^{(2)} = &|\mathbb{1}\tau\mathbb{1}\rangle\langle\mathbb{1}\tau\mathbb{1}| + \phi^{-2}\,|\tau\mathbb{1}\tau\rangle\langle\tau\mathbb{1}\tau| \\ &+ \phi^{-1}\,|\tau\tau\tau\rangle\langle\tau\tau\tau| \\ &+ \phi^{-3/2}\left(|\tau\mathbb{1}\tau\rangle\langle\tau\tau\tau| + \text{H.c.}\right), \end{aligned} \tag{4.14}$$

and the three-anyon interactions can be described by the Majumdar-Gosh chain Hamiltonian [153]

$$\begin{aligned} H_i^{(3)} = &|\mathbb{1}\tau\tau\mathbb{1}\rangle\langle\mathbb{1}\tau\tau\mathbb{1}| + \left(1-\phi^{-2}\right)|\tau\tau\tau\tau\rangle\langle\tau\tau\tau\tau| \\ &+ \left(1-\phi^{-1}\right)\left(|\tau\tau\mathbb{1}\tau\rangle\langle\tau\tau\mathbb{1}\tau| + |\tau\mathbb{1}\tau\tau\rangle\langle\tau\mathbb{1}\tau\tau|\right) \\ &- \phi^{-5/2}\left(|\tau\mathbb{1}\tau\tau\rangle\langle\tau\tau\tau\tau| + |\tau\tau\mathbb{1}\tau\rangle\langle\tau\tau\tau\tau| + \text{H.c.}\right) \\ &+ \phi^{-2}\left(|\tau\tau\mathbb{1}\tau\rangle\langle\tau\mathbb{1}\tau\tau| + \text{H.c.}\right), \end{aligned} \tag{4.15}$$

$\phi = \left(\sqrt{5}+1\right)/2$ is the golden ratio.

This model is predicted to exhibit five phases with respect to different θ [153]. Fig. 4.13 shows the optimized energies of the Hamiltonian in Eq. 4.13 for different θ and the energy derivative computed using the Hellman-Feynman theorem [164]. The non-differentiable points of the energy derivative indicates the phase transition points, which agree with the conformal field theory prediction. In the special case of $\theta = 0$, the model reduces to the Fibonacci anyons in a golden chain, which has a gapless phase [153]. Using our optimized AR-NN, we compute the second Renyi entropy S_2 [122]. Since the second Renyi entropy S_2 is related to the central charge c under the periodic boundary condition as $S_2 \sim \frac{c}{4}\log(L)$ with system size L [167], we then extract the central charge finding a value $c = 0.703 \pm 0.005$ very close to the exact result of 0.7 (see Fig. 4.14).

This can be generalized to the $\mathrm{SU}(2)_k$ formulation of anyon theory, for which there are $k+1$ species of anyons labeled by $j = 0, 1/2, 1, \ldots, k/2$ with the fusion rule of $\mathrm{SU}(2)_k$ [152]. The Hamiltonian can be expressed with operators from the representation of the Temperley-Lieb algebra [152]. To construct a gauge invariant autoregressive neural network for the general $\mathrm{SU}(2)_k$ anyonic chain, one works in the angular momentum basis $\{|\ldots, j_{i-1}, j_i, j_{i+1}, \ldots\rangle\}$ where $j_i \in \{0, 1/2, 1, \ldots, k/2\}$. Since each j_i is included as the $\mathrm{SU}(2)_k$ fusion rule outcome of j_{i-1} and an extra $1/2$ angular momentum, one can view j_i as a composite particle and gauge checking is the fusion rule.

The Fibonacci anyon is a special case of the $\mathrm{SU}(2)_{k=3}$ formulation, considering the mapping $\tau \mapsto j = 1$ and $\mathbb{1} \mapsto j = 0$ and applying $3/2 \times j = 3/2 - j$ from the $\mathrm{SU}(2)_3$ fusion rule to the even-number sites [152]. Note that this gives a slightly different AR-NN from what is described above. Besides the Fibonacci anyon, one can consider the Yang-Lee anyon, which follows the $\mathrm{SU}(2)_3$ fusion rule [168].

Using this framework, one can also consider the Heisenberg spin chain with SU(2) symmetry since it can be considered as the $\mathrm{SU}(2)_k$ deformation of the ordinary quantum spin-1/2 chain [154] as $k \to \infty$. In Supplementary Materials 4.6.3, we provide the detailed construction of an SU(2) invariant autoregressive neural network for the Heisenberg model, which can be viewed as the case of $\mathrm{SU}(2)_{k=\infty}$, and obtain accurate results for the 1D Heisenberg model.

4.5 Conclusion

We have provided a general approach to constructing gauge invariant autoregressive neural network wave functions for various quantum lattice models with gauge constraints or special algebraic structures. These wave functions explicitly satisfy the gauge or algebraic constraints, allow for perfect sampling of configura-

tions, and are capable of explicitly returning the amplitude of a configuration including normalization. To accomplish this, we have upgraded standard AR-NN in such a way that the constraints can be autoregressively satisfied.

We have given explicit constructions of AR-NN which exactly represent the ground and excited states of several models, including the 2D and 3D toric codes as well as the X-cube fracton model. For those models for which exact representations are unknown, we variationally optimize our gauge invariant AR-NN to obtain either high-quality ground states or time-dependent wave functions. This has been done for the U(1) quantum link model, $\mathbb{Z}_N$ gauge theory, the $\mathrm{SU}(2)_3$ anyonic chain, and the SU(2) quantum spin-1/2 chain. For these systems we are able to measure dynamical properties, produce phase diagrams, and compute observables accurately.

Our gauge invariant autoregressive neural networks open up the possibility of probing a larger variety of models and the physics associated with them. For example, the higher spin representation $S > 1/2$ in the (1+1)D QLM models would allow one to probe the quantum chromodynamics related physics of confinement and string breaking [169]. For the (3+1)D QLM models, there is the Coulomb phase which manifests in pyrochlore spin liquids [170]. For $\mathbb{Z}_N$ gauge theory, it will be interesting to consider the general $\mathbb{Z}_N$ toric code with transverse field or disorder, with a goal of understanding its phase diagram. Recently, there have been proposals to understand the $\mathbb{Z}_N$ X-cube fracton model with non-trivial statistical phases [137]. For non-abelian anyons, the general $\mathrm{SU}(2)_k$ formulation exhibits rich physics for different k and one can study the corresponding topological liquids and edge states [154]. Our approach can also be extended to study the phase diagram for the 2D Heisenberg models with SU(2) symmetry.

Besides exploring various models in condensed matter physics and high energy physics, our approach can also be further applied to quantum information and quantum computation. Fibonacci anyons are known to support universal topological quantum computation, which is robust to local perturbations [171]. It will be interesting to see how well one can approximately simulate topological quantum computation or different braiding operations with gauge invariant autoregressive neural networks. As toric codes are an important example of quantum error correction code, our approach can be used to approximately study the performance of a toric code under different noise conditions. With respect to the recent efforts on simulating lattice gauge theories with quantum computation, our approach also provides an alternative method to compare to and benchmark quantum computers. In summary, the approach we have developed is versatile and powerful for investigating condensed matter physics, high energy physics and quantum information science.

4.6 Supplementary Materials

4.6.1 Kitaev's $D(G)$ Model and Exact Representation of Ground State

We generalize our gauge invariant autoregressive construction for the 2D $\mathbb{Z}_2$ toric code. Kitaev's $D(G)$ model [116] is defined on an $L \times L$ periodic square lattice where each edge has a basis $\{|g\rangle, g \in G\}$ for some group G. Here we focus on finite groups, in particular $G = \mathbb{Z}_N$ for $\mathbb{Z}_N$ theory. Without loss of generality, we attach an upward arrow for each edge in the y-direction and a right arrow for each edge in the x-direction. We employ the notation of Sec. 4.4.2 and introduce operators A_v^g and $B_p^{h_u,h_d,h_l,h_r}$ as in Fig. 4.15.

The Hamiltonian defined on $\mathcal{H}(G)^{\otimes E}$ is

$$H = -\sum_{v\in V} A_v - \sum_{p\in P} B_p, \tag{4.16}$$

where $A_v = \frac{1}{|G|}\sum_{g\in G} A_v^g$ is Gauss's law and the gauge constraint, and $B_p = \sum_{h_u h_r h_d h_l = \mathbb{1}_G} B_p^{h_u,h_r,h_d,h_l}$.

Let $|+\rangle = \frac{1}{\sqrt{|G|}}\sum_{g\in G}|g\rangle$, and $|\psi\rangle = \prod_{p\in P} B_p |+\rangle^{\otimes E}$ is the ground state. This is because $|\psi\rangle$ is a ground state for each A_v and B_p. It is easy to verify that $B_p|\psi\rangle = |\psi\rangle$. To see $A_v|\psi\rangle = |\psi\rangle$, notice that A_v and B_p commute and $A_v|+\rangle^{\otimes E} = |+\rangle^{\otimes E}$. Similarly to the $\mathbb{Z}_2$ toric code, the ground state can be constructed using gauge invariant autoregressive neural networks by defining each star as a composite particle and checking Gauss's law and bond consistency.

4.6.2 (1+1)D $\mathbb{Z}_N$ Lattice QED Model

Our approach in Sec. 4.4.1 can be applied to the (1+1)D $\mathbb{Z}_N$ lattice quantum electrodynamics (QED) model, which is a discretization of the Schwinger model for the continuous-space QED in 1+1 dimensions [172]. The (1+1)D $\mathbb{Z}_N$ model takes a similar form as the (1+1)D QLM, which has fermions on sites and electric fields on links between two sites. Let $\{|e_{i,i+1}\rangle\}$ for $1 \le e_{i,i+1} \le N$ denote the orthonormal basis on each link $(i, i+1)$. The (1+1)D $\mathbb{Z}_N$ gauge theory can take the following form [172]

$$\begin{aligned} H = &-\sum_i \left[\psi_i^\dagger U_{i,i+1}\psi_{i+1} + \psi_{i+1}^\dagger U_{i,i+1}^\dagger \psi_i\right] \\ &+ m\sum_i (-1)^i \psi_i^\dagger \psi_i + \frac{g^2}{8}\sum_i (V_{i,i+1} - \mathbb{1})(V_{i,i+1}^\dagger - \mathbb{1}), \end{aligned} \tag{4.17}$$

where $U_{i,i+1}\,|e_{i,i+1}\rangle = |(e_{i,i+1}+1)\text{mod } N\rangle$, and $V_{i,i+1}|e_{i,i+1}\rangle = e^{-i2\pi m/N}|e_{i,i+1}\rangle$ for $m = e_{i,i+1}$. The Gauss's law operator G_i of the model can be written as

$$G_i = e^{\frac{i2\pi}{N}(\psi_i^\dagger\psi_i + \frac{1}{2}(-1)^i - \frac{1}{2})} V_{i,i+1} V_{i-1,i}^\dagger \tag{4.18}$$

such that $G_i\,|\psi\rangle = |\psi\rangle$ for each i [172].

Similarly to the (1+1)D QLM, one can construct the gauge invariant autoregressive neural network as Fig. 4.2 and perform gauge checking with G_i in Eq. 4.18.

4.6.3 SU(2) Invariant Autoregressive Neural Network for Heisenberg Model

The 1D Heisenberg Model is described as

$$H = \sum_i \sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z. \tag{4.19}$$

We work in the angular momentum basis $\{|j_1, j_2, j_3, \ldots, j_n\rangle\}$ similarly to [154], instead of the spin basis, to construct an SU(2) invariant autoregressive wave function. Each j_i is the total angular momentum quantum number for spins from 1 to i and $j_n \equiv J$ is the total angular momentum quantum number for all spins. For the ground state of the Heisenberg model, the total angular momentum is zero, so $j_n = 0$. We define the first composite particle as j_1 and the i'th composite particle as the difference $j_i - j_{i-1}$. Note that this uniquely defines a physical state. We then autoregressively enforce $j_{i<n} \geq 0$ and $j_n = 0$ as gauge checking, to achieve the SU(2) invariant property.

Fig. 4.16 demonstrates the performance of our SU(2) invariant autoregressive neural network on the Heisenberg model with SU(2) symmetry.

4.6.4 Neural Network Architecture

Complex Parameterization

Wave functions are complex in general but both the Transformer network and 1D/2D RNN are real. We use two approaches (Fig. 4.17)—(a) amplitude-phase and (b) real-imaginary—to parametrize complex wave functions from real neural networks. In both parameterizations, the input configuration $\widetilde{\boldsymbol{x}}$, together with a default configuration $\widetilde{x}_0$, is embedded (i.e. each state of a composite particle is mapped to a unique vector) before fed into the Transformer or 1D/2D RNN. Certain gauge blocks in an AR-NN take a default state $\widetilde{x}_0$ as opposed to any state of the composite particles; the embedded vector of this default state has arbitrary

parameters which are trained during optimization.

Transformer

The Transformer used in this work (Fig. 6.3) is the same as the Transformer used in Ref. [123], which can be viewed as the standard Transformer encoder with masked multi-head attention from Ref. [145] but without an additional add & norm layer. The Transformer consists of a standard positional encoding layer, which uses sinusoidal functions to encode the positional information of the embedded input. After positional encoding, the input is fed into the standard masked multi-head attention mechanism. The mask here is crucial for autoregressiveness, as it only allows each site to depend on the previous sites. The output of the attention layer is then passed through a standard feed forward layer. The detailed explanation of the Transformer can be found in Refs. [145, 123]. This transformer is essentially equivalent to the standard PyTorch implementation [156], but was implemented independently because that implementation did not exist at the start of our work.

RNN Cells

For all RNNs in this work, we used the gated recurrent unit (GRU) cell [142] (Fig. 4.19) in PyTorch [156], which takes one input vector x_k, the hidden input h_{k-1}, and computes

$$\begin{aligned} r &= \sigma(W_{xr}x_k + b_{xr} + W_{hr}h_{k-1} + b_{hr}), \\ z &= \sigma(W_{xz}x_k + b_{xz} + W_{hz}h_{k-1} + b_{hz}), \\ n &= \tanh(W_{xn}x_k + b_{xn} + r \odot (W_{hn}h_{k-1} + b_{hn})), \\ h_k &= (1 - z) \odot n + z \odot h_{k-1}, \\ y_k &= h_k, \end{aligned} \tag{4.20}$$

where σ is the sigmoid function and $\odot$ means element-wise product.

We then build 1D and periodic 2D RNN cells (Fig. 4.20) based on the GRU cell. The 1D RNN cell computes

$$\begin{aligned} (y_{\text{raw}}, h_{\text{new}}) &= \text{GRUcell}(x_k, h_{\text{old}}), \\ y_k &= y_{\text{raw}} + x_k, \end{aligned} \tag{4.21}$$

whereas the periodic 2D RNN cell computes

$$\begin{aligned}
&(y_{\text{raw}}, h_{\text{raw}}) = \text{GRUcell}(x_k, [h_{\text{old1}}, h_{\text{old2}}, h_{\text{old3}}, h_{\text{old4}}]),\\
&[h_{\text{new1}}, h_{\text{new2}}, h_{\text{new3}}, h_{\text{new4}}] = h_{\text{raw}} + x_k,\\
&h_{\text{new}} = \frac{1}{4}\left(h_{\text{new1}} + h_{\text{new2}} + h_{\text{new3}} + h_{\text{new4}}\right),\\
&y_k = y_{\text{raw}} + x_k.
\end{aligned} \tag{4.22}$$

RNNs

With the RNN cells, we can build 1D and periodic 2D RNNs.

The 1D RNN (Fig. 4.21) has a multi-layer design and shares the same structure as the Pytorch [156] GRU [142]. The embedded input configuration is fed into the cells one at a time through multiple layers and produces a raw output. In our work, the cells at different layers share the weight matrices and bias vectors.

The periodic 2D RNN has a more complicated design to capture the most correlations and can be viewed as a periodic extension of the 2D RNN in Ref. [122]. In each layer, the hidden vector h is passed around according to Fig. 4.22(a), where each cell receives a maximum number of four hidden vectors and concatenates them according to Fig. 4.20(b). When the number of hidden vectors received is less than four, zero vectors are used to pad the concatenated hidden vector to the correct length. The configuration is evaluated and sampled in a zigzag S path (Fig. 4.22(b)) to ensure autoregressiveness.

Before the first layer of the periodic 2D RNN, a special concatenation of the embedded input needs to be performed. At each location, the concatenation layer takes the four surrounding inputs (periodically) and concatenates them into a single vector. If any (or all) of the surrounding inputs lies later in the conditioning order in Fig. 4.22(b), the corresponding input is replaced with a default input x_0. For a 4×4 2D input array shown in Fig. 4.23(a), some concatenation examples are shown in Fig. 4.23(b). After the first layer, the output of a previous layer can be directly fed into the next layer similar to a regular RNN without any further process.

A multi-layer periodic 2D RNN consists of one input concatenation layer and several 2D RNN layers as shown in Fig. 4.24.

Initialization and Optimization

We use different initialization techniques for different models. For the QLM model, the initialization is done through tomography, minimizing $-\log|\psi(x)|^2$ for a desired configuration x ($|\bullet \to \circ \to\rangle$ for each unit cell in this case). For the 2D toric code model, we set the weight matrix in the last linear layer to be 0 and the bias

such that the wave function for each composite particle is $\sqrt{0.23}\left|0^{0}_{0}0\right\rangle+\sqrt{0.11}\left|1^{1}_{0}0\right\rangle+\sqrt{0.11}\left|1^{0}_{1}0\right\rangle+\sqrt{0.11}\left|0^{0}_{1}1\right\rangle+\sqrt{0.11}\left|0^{1}_{0}1\right\rangle+\sqrt{0.11}\left|1^{0}_{0}1\right\rangle+\sqrt{0.11}\left|0^{1}_{1}0\right\rangle+\sqrt{0.11}\left|1^{1}_{1}1\right\rangle$, which empirically produces a very low initial energy. We used a transfer learning technique, where we first train our neural network on a 6×6 model before training it on the 10×10 model. When transferred to the larger system, the weight and bias in the last linear layer is dropped and replaced with the initialization scheme described above. In Fig. 4.25, we show the effect of transfer learning on a 10×10 toric code model. With transfer learning, the energy is clearly lower than the energy without transfer learning.

For the anyon model, we set the weight matrix in the last linear layer to be 0 and the bias to be the state of $1/\sqrt{2}\,|\mathbb{1}\rangle+1/\sqrt{2}\,|\tau\rangle$ for each particle. When producing the phase diagram, we used a transfer learning technique, where we first train the neural network on 32 anyons with $\theta = 0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4$, and 2π for 3000 iterations, and then transfer the model with θ that is closest to the desired value of θ for 40 anyons for another 3000 iterations. Similar to the toric code case, when transferred to the larger system, the weight and bias in the last linear layer is dropped and replaced by the initialization described above. In all models, except the last layer, the weights and biases are initialized using PyTorch's [156] default initialization. For optimization, we used the Adam [174] optimizer with an initial learning rate of 0.01. For dynamics, the learning rate is halved at iterations 100, 200, 270, 350, and 420, whereas for the ground state optimization, the learning rate is halved at iterations 100, 500, 1000, 1800, 2500, 4000, and 6000.

4.6.5 Derivation of Stochastic Gradients for Variational and Dynamics Optimization

In Sec. 4.3, we presented stochastic gradients of the variational and dynamics optimizations. This section includes their derivations.

The variational optimization has been widely used and derived many times in other works [119, 122].

Here we present the derivation for the sake of completeness:

$$
\begin{aligned}
&\frac{\partial \langle\psi_\theta|H|\psi_\theta\rangle}{\partial\theta} \\
&= \left\langle \frac{\partial\psi_\theta}{\partial\theta}\middle| H \middle|\psi_\theta\right\rangle + \left\langle \psi_\theta\middle| H \middle|\frac{\partial\psi_\theta}{\partial\theta}\right\rangle \\
&= 2\sum_x \Re\left\{\frac{\partial\psi_\theta^*(x)}{\partial\theta} H\psi_\theta(x)\right\} \\
&= 2\sum_x \Re\left\{\frac{1}{\psi_\theta^*(x)}\frac{\partial\psi_\theta^*(x)}{\partial\theta}\psi_\theta^*(x)H\psi_\theta(x)\right\} \\
&= 2\sum_x \Re\left\{\psi_\theta^*(x)H\psi_\theta(x)\frac{\partial}{\partial\theta}\log\psi_\theta^*(x)\right\} \\
&\approx \frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N} \Re\left\{\frac{H\psi_\theta(x)}{\psi_\theta(x)}\frac{\partial}{\partial\theta}\log\psi_\theta^*(x)\right\} \\
&\equiv \frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N} \Re\left\{E_{\mathrm{loc}}(x)\frac{\partial}{\partial\theta}\log\psi_\theta^*(x)\right\},
\end{aligned}
\tag{4.23}
$$

where the local energy is $E_{\mathrm{loc}}(x) \equiv H\psi_\theta(x)/\psi_\theta(x)$. We can further control the variance by subtracting from the $E_{\mathrm{loc}}(x)$ the average energy $E_{\mathrm{avg}} \equiv \sum_{x\sim|\psi_\theta|^2}^{N} E_{\mathrm{loc}}(x)/N$ over the batch of samples [155] as we did in Sec. 4.3 and use the stochastic variance reduced gradient as

$$
\frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N} \Re\left\{\left[E_{\mathrm{loc}}(x) - E_{\mathrm{avg}}\right]\frac{\partial}{\partial\theta}\log\psi_\theta^*(x)\right\}, \tag{4.24}
$$

which has the same expectation as Eq. 4.23.

For the dynamics optimization gradient, as in Sec. 4.3, we define $|\Psi_\theta\rangle = (1 + iH\tau)\left|\psi_{\theta(t+2\tau)}\right\rangle$ and $|\Phi\rangle = (1 - iH\tau)\left|\psi_{\theta(t)}\right\rangle$, and we drop $\theta(t)$ and name $\theta \equiv \theta(t + 2\tau)$. We start by splitting the negative log overlap:

$$
\begin{aligned}
&-\log\frac{\langle\Psi_\theta|\Phi\rangle\langle\Phi|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle\langle\Phi|\Phi\rangle} \\
&= -\log\langle\Psi_\theta|\Phi\rangle - \log\langle\Phi|\Psi_\theta\rangle + \log\langle\Psi_\theta|\Psi_\theta\rangle + \log\langle\Phi|\Phi\rangle\,.
\end{aligned}
\tag{4.25}
$$

We then compute the gradient term by term. The first term on the right side of Eq. 4.25 becomes

$$\begin{aligned}
&\frac{\partial}{\partial\theta}\log\langle\Psi_\theta|\Phi\rangle \\
&= \frac{1}{\langle\Psi_\theta|\Phi\rangle}\left\langle\frac{\partial\Psi_\theta}{\partial\theta}\middle|\Phi\right\rangle \\
&= \frac{1}{\sum_x \Psi_\theta^*(x)\Phi(x)}\sum_x \frac{\partial\Psi_\theta^*(x)}{\partial\theta}\Phi(x) \\
&= \frac{1}{\sum_x \Psi_\theta^*(x)\Phi(x)}\sum_x \Psi_\theta^*(x)\Phi(x)\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x) \\
&\approx \frac{1}{\sum_{x\sim|\psi_\theta|^2}^N \frac{\Psi_\theta^*(x)\Phi(x)}{|\psi_\theta(x)|^2}}\sum_{x\sim|\psi_\theta|^2}^N \frac{\Psi_\theta^*(x)\Phi(x)}{|\psi_\theta(x)|^2}\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x) \\
&\equiv \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^N \frac{\alpha(x)}{\alpha_{\text{avg}}}\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x),
\end{aligned} \tag{4.26}$$

where $\alpha(x) = \Psi_\theta^*(x)\Phi(x)/|\psi_\theta(x)|^2$ and $\alpha_{\text{avg}} = \sum_{x\sim|\psi_\theta|^2}^N \alpha(x)/N$ as in Sec. 4.3. The second term on the right side of Eq. 4.25 is just the complex conjugate of the first term, whereby

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log\langle\Phi|\Psi_\theta\rangle &= \left[\frac{\partial}{\partial\theta}\log\langle\Psi_\theta|\Phi\rangle\right]^* \\
&\approx \frac{1}{N}\sum_{x\sim|\psi_\theta|^2}^N \left[\frac{\alpha(x)}{\alpha_{\text{avg}}}\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x)\right]^*.
\end{aligned} \tag{4.27}$$

The third term on the right side of Eq. 4.25 becomes

$$\begin{aligned}
&\frac{\partial}{\partial\theta}\log\langle\Psi_\theta|\Psi_\theta\rangle \\
&= \frac{1}{\langle\Psi_\theta|\Psi_\theta\rangle}\left(\left\langle\frac{\partial\Psi_\theta}{\partial\theta}\middle|\Psi_\theta\right\rangle + \left\langle\Psi_\theta\middle|\frac{\partial\Psi_\theta}{\partial\theta}\right\rangle\right) \\
&= \frac{2}{\sum_x |\Psi_\theta(x)|^2}\sum_x \Re\left\{\frac{\partial\Psi_\theta^*(x)}{\partial\theta}\Psi_\theta(x)\right\} \\
&= \frac{2}{\sum_x |\Psi_\theta(x)|^2}\sum_x \Re\left\{|\Psi_\theta(x)|^2\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x)\right\} \\
&\approx \frac{2}{\sum_{x\sim|\psi_\theta|^2}^N \frac{|\Psi_\theta(x)|^2}{|\psi_\theta(x)|^2}}\sum_{x\sim|\psi_\theta|^2}^N \Re\left\{\frac{|\Psi_\theta(x)|^2}{|\psi_\theta(x)|^2}\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x)\right\} \\
&\equiv \frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^N \Re\left\{\frac{\beta(x)}{\beta_{\text{avg}}}\frac{\partial}{\partial\theta}\log\Psi_\theta^*(x)\right\},
\end{aligned} \tag{4.28}$$

where $\beta(x) = |\Psi_\theta(x)|^2/|\psi_\theta(x)|^2$ and $\beta_{\text{avg}} = \sum_{x\sim|\psi_\theta|^2}^N \beta(x)/N$ as in Sec. 4.3. The last term on the right side

of Eq. 4.25 is θ independent such that

$$\frac{\partial}{\partial\theta}\log\langle\Phi|\Phi\rangle = 0. \tag{4.29}$$

Combining all the terms together,

$$\begin{aligned}&\frac{\partial}{\partial\theta}\left(-\log\frac{\langle\Psi_\theta|\Phi\rangle\langle\Phi|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle\langle\Phi|\Phi\rangle}\right)\\ &\approx\frac{2}{N}\sum_{x\sim|\psi_\theta|^2}^{N}\Re\left\{\left[\frac{\beta(x)}{\beta_{\text{avg}}}-\frac{\alpha(x)}{\alpha_{\text{avg}}}\right]\frac{\partial}{\partial\theta}\log\Psi^*_\theta(x)\right\}.\end{aligned} \tag{4.30}$$

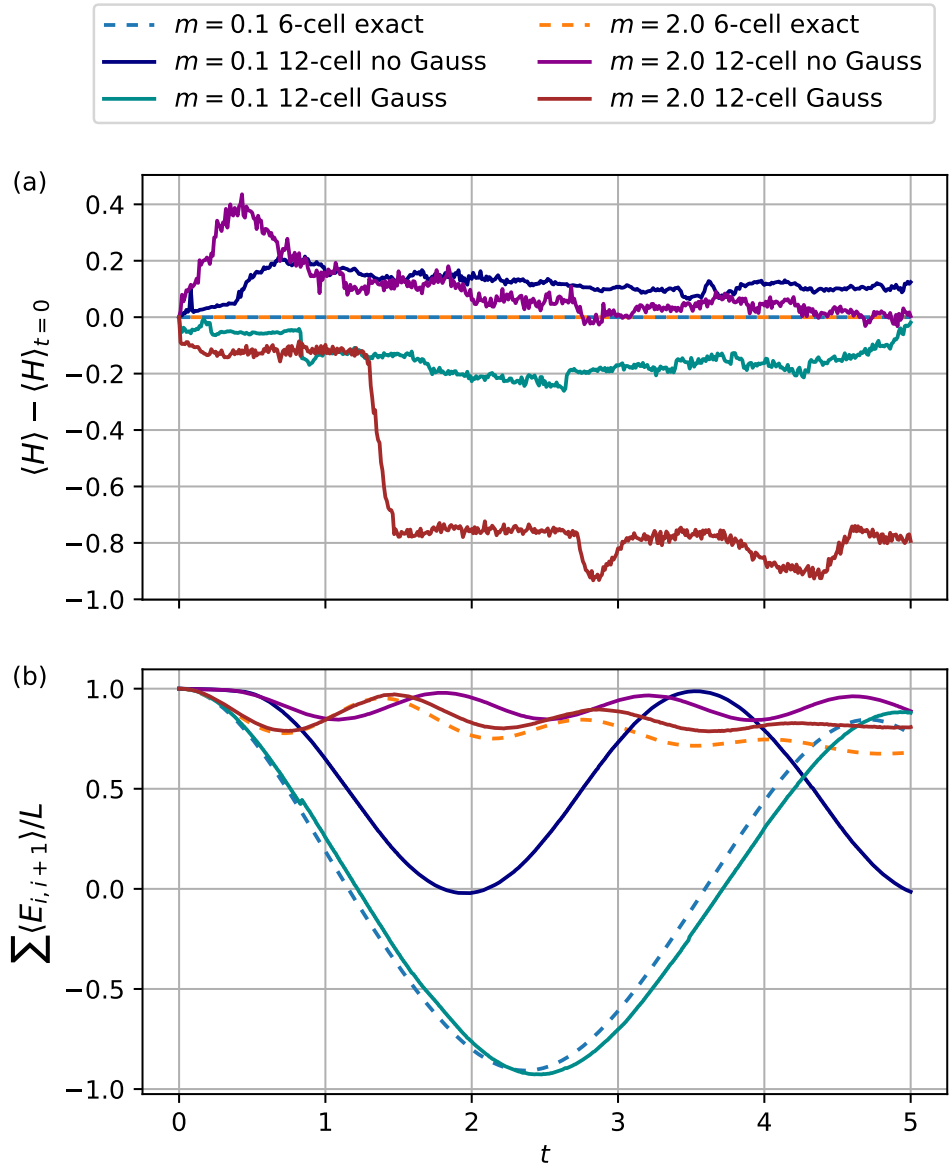


Figure 4.4: Dynamics for the 12-unit-cell (24 sites and 24 links) open-boundary QLM for $m = 0.1$ and $m = 2.0$ with and without gauge invariant construction. The dashed curves are the exact results from the exact diagonalization for 6 unit cells, which can be used as a reference for the 12-unit-cell neural network results. (a) The change in the energy during the dynamics. (b) The expectation value of the electric field averaged over all links. We use the Transformer neural network with 1 layer, 32 hidden dimensions and the real-imaginary parameterization. The initial state is $|\bullet \rightarrow \circ \rightarrow\rangle$ for each unit cell and we train the neural network using the forward-backward trapezoid method with the time step $\tau = 0.005$, 600 iterations in each time step, and 12000 samples in each iteration. The neural network architecture, initialization and optimization details are discussed in Supplementary Materials 4.6.4.

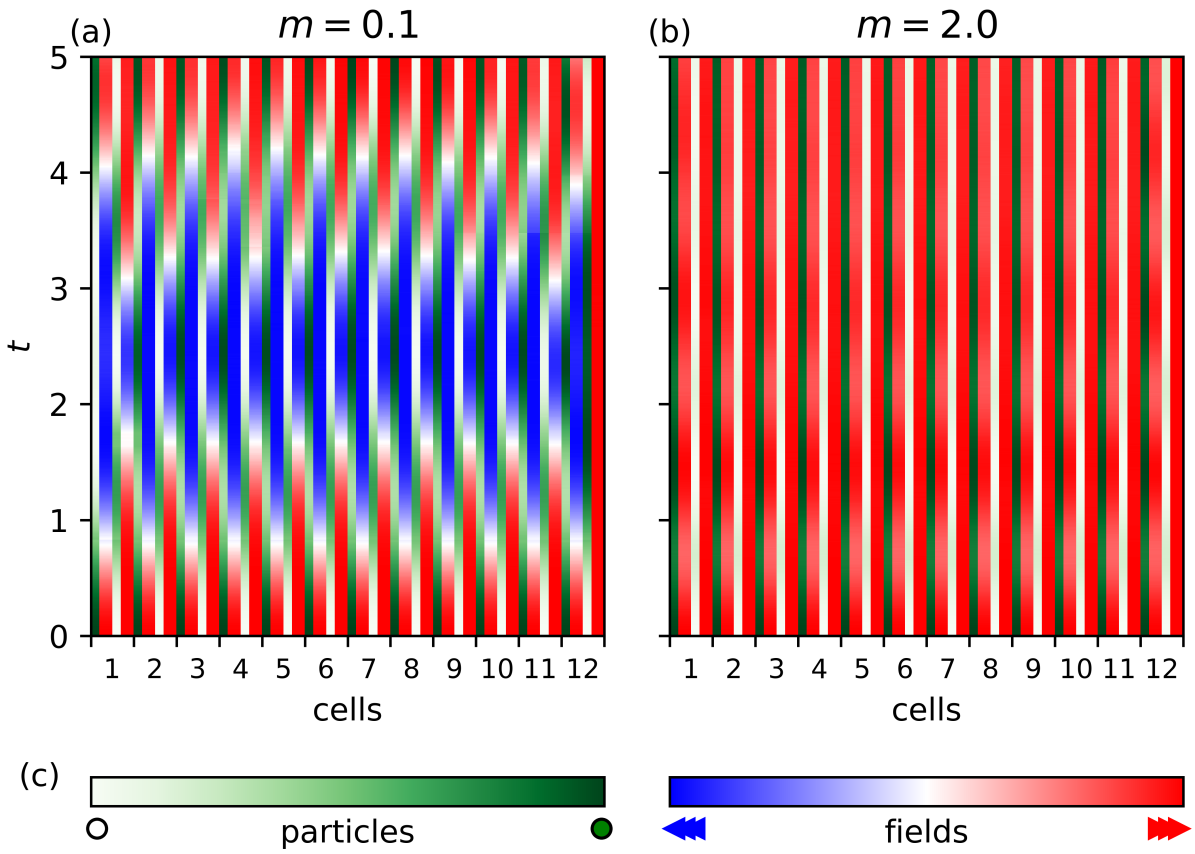


Figure 4.5: Dynamics of the gauge invariant AR-NN for the 12-unit-cell QLM with (a) $m = 0.1$ and (b) $m = 2.0$. The ansatz, initialization and optimization are the same as in Fig. 4.4 and are discussed in Supplementary Materials 4.6.4.

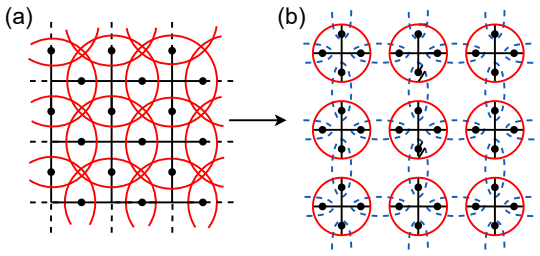


Figure 4.6: Composite particles for the 2D toric code. (a) Physical structure of 2D toric code with red circles specifying composite particles. Note multiple composite particles share the same physical sites. (b) Composite particles. We define each star as a composite particle (red circle) and check bond consistency for physical sites shared by adjacent composite particles (blue dashed ovals).

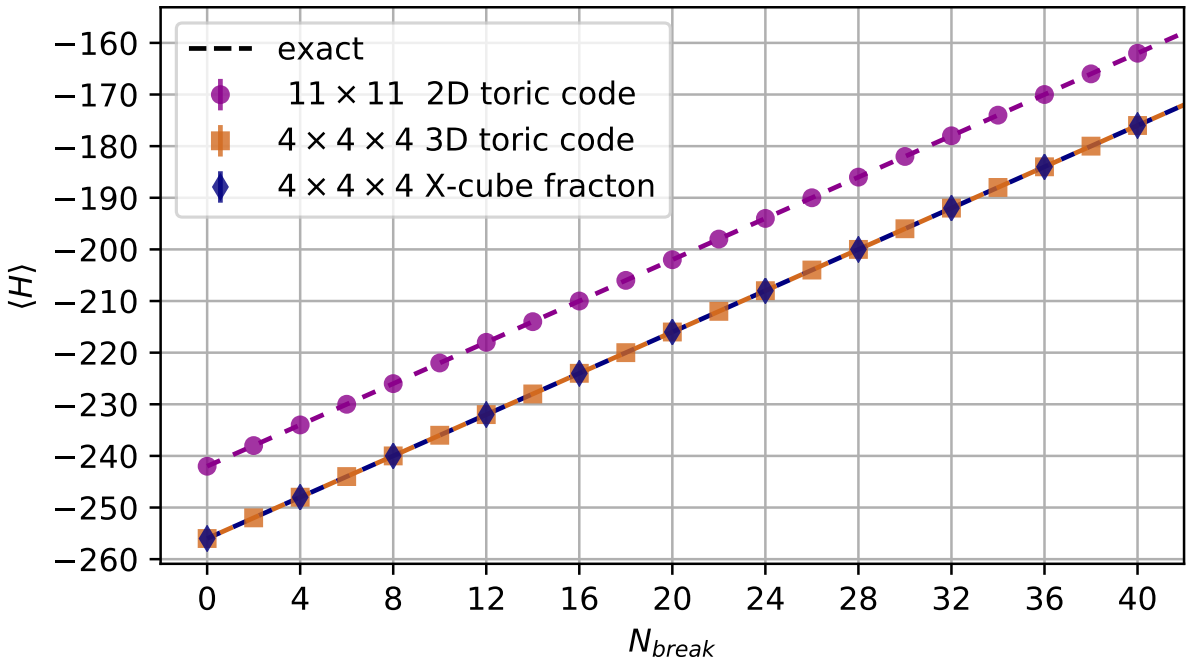


Figure 4.7: Energies of the analytical constructions of the ground and excited states of the 11×11 2D and $4 \times 4 \times 4$ 3D toric code, and the $4 \times 4 \times 4$ X-cube fracton model. The dashed lines are the exact values for each model. The analytical construction generates the same values as the exact up to stochastic errors from sampling.

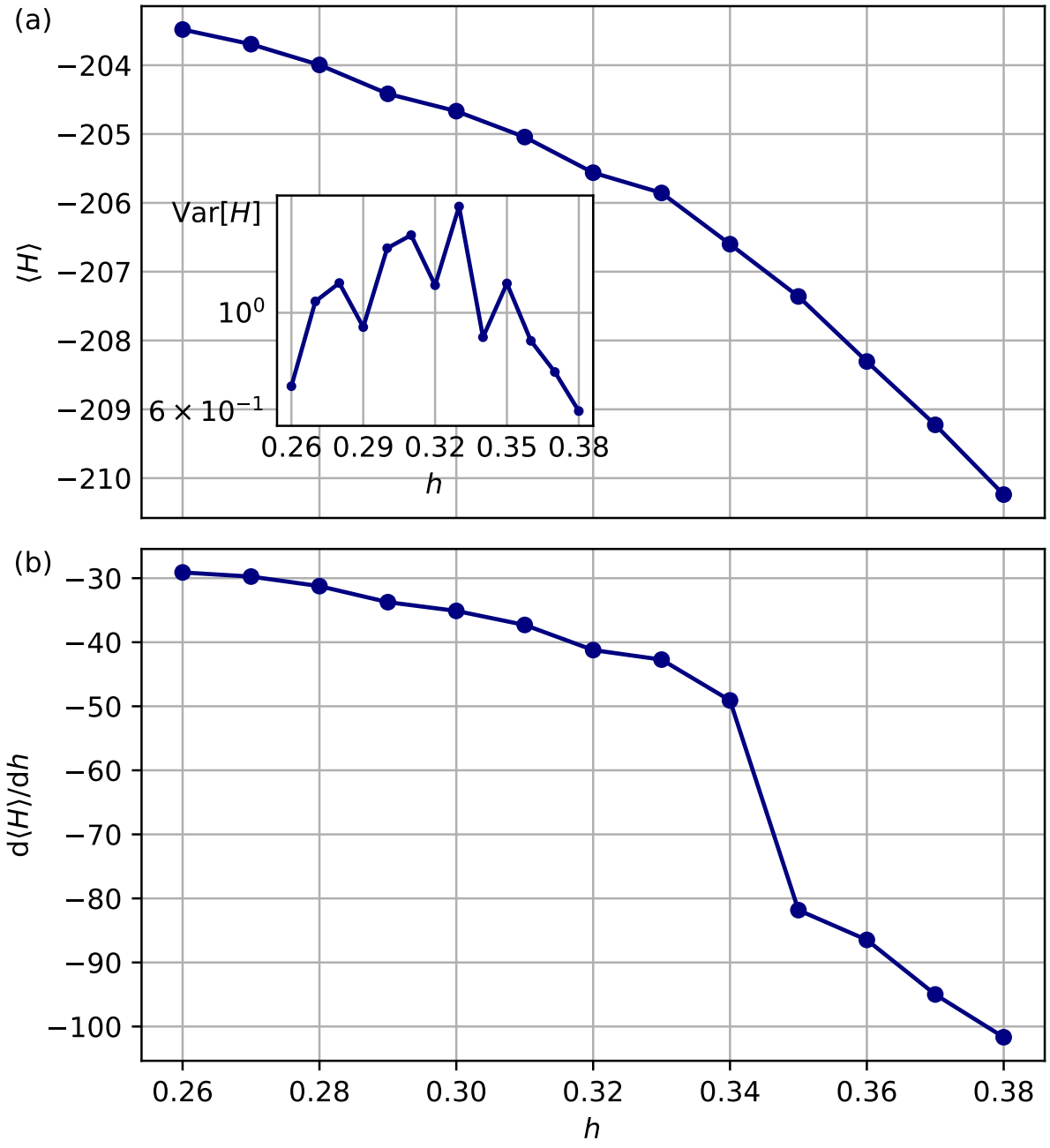


Figure 4.8: (a) Energy, (inset) energy variance and (b) energy derivative (computed by the Hellman-Feynman theorem[164] as $\mathrm{d}\langle H\rangle/\mathrm{d}h = \langle \mathrm{d}H/\mathrm{d}h\rangle = -\sum_{e\in E}\langle\sigma_e^z\rangle$) versus h. We use the 2D RNN with 3 layers, 32 hidden dimensions and the amplitude-phase parameterization (see Fig. 4.17). We use the transfer learning technique where we first train the neural network on a 6×6 model for 8000 iterations and then we transfer the neural network to the 10×10 model for another 1000 iterations. In each iteration, we use 12000 samples. The neural network architecture, initialization and optimization details are discussed in Supplementary Materials 4.6.4.

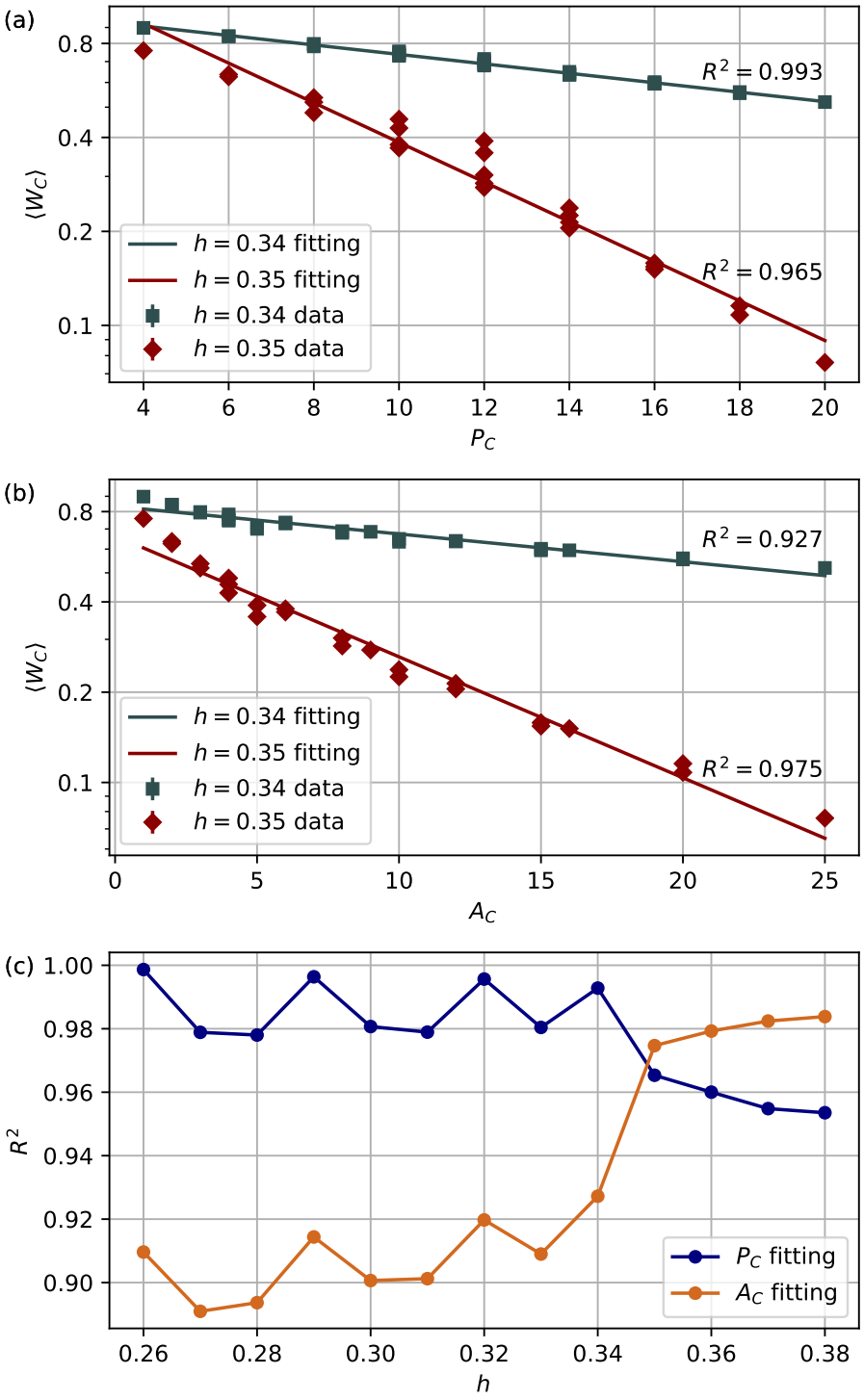


Figure 4.9: Perimeter and area laws for the 10×10 2D toric code. The expectation value of the Wilson loop operator with respect to the (a) perimeter and (b) area of the loop in a log-y scale for $h = 0.34$ and 0.35. (c) The fitting of the correlation coefficient R^2 for the perimeter and area laws for different h. The ansatz, initialization, and optimization are the same as in Fig. 4.8 and discussed in Supplementary Materials 4.6.4.

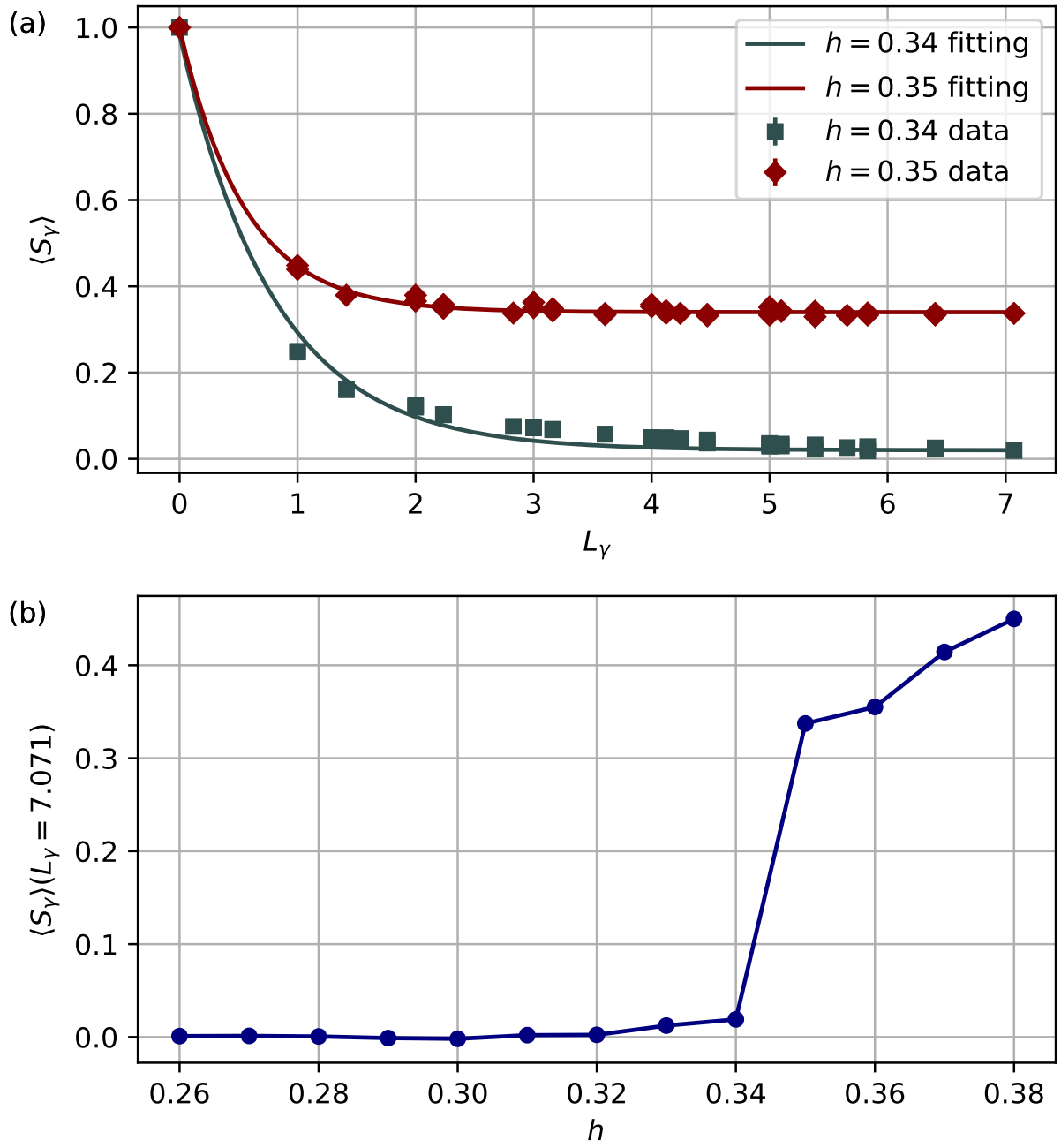


Figure 4.10: (a) Non-local string correlation function for the 10×10 2D toric code between a pair of particle and anti-particle with a distance of L_y apart. (b) The correlation of a pair of particle and anti-particle at a distance of $L_y = 5\sqrt{2}$ for different h. The ansatz, initialization, and optimization are the same as in Fig. 4.8 and discussed in Supplementary Materials 4.6.4.

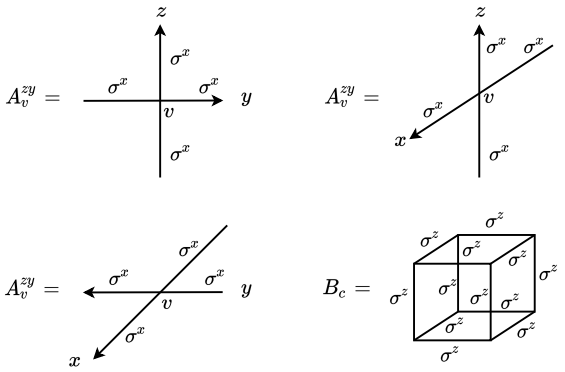


Figure 4.11: A_v^i and B_c for the X-cube fracton model.

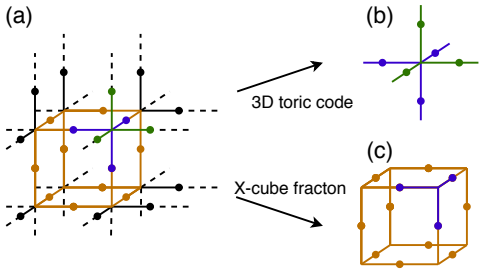


Figure 4.12: Composite particles of the 3D toric code and X-cube fracton model. The colors are to help identify how composite particles are defined (i.e. which parts of (a) map to (b) and (c)). (a) Physical structure of the 3D toric code and X-cube fracton model. (b) Composite particles of the 3D toric code. We define each star as a composite particle and check bond consistency for adjacent particles similarly to the 2D toric code. (c) Composite particles of the X-cube fracton model. We define each cube as a composite particle and check bond consistency on faces of adjacent particles.

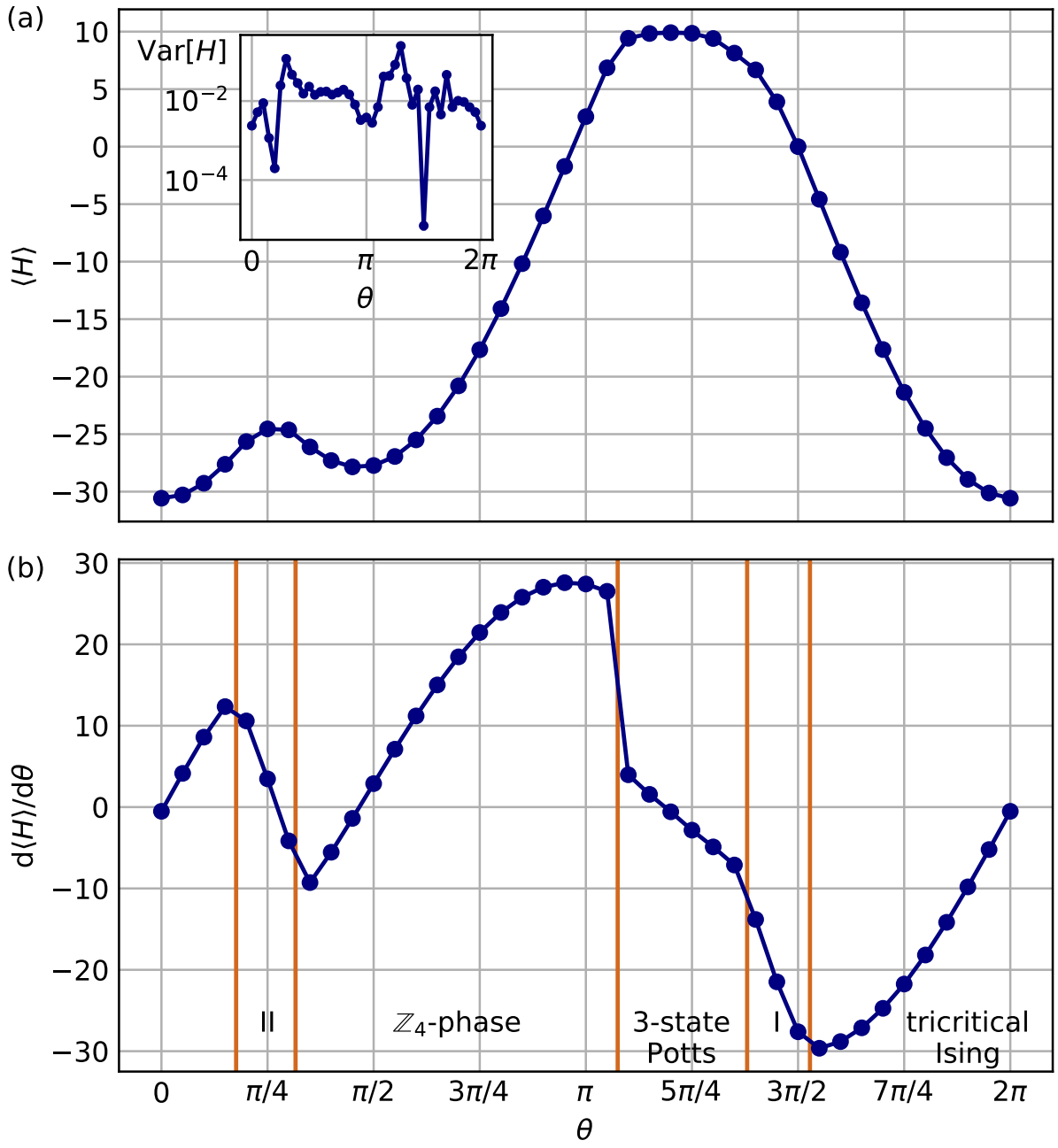


Figure 4.13: Phase diagram for 40 anyons with the periodic boundary condition. (a) Energy, (inset) energy variance and (b) energy derivative, computed using the Hellmann–Feynman theorem [164] as $\mathrm{d}\langle H\rangle/\mathrm{d}\theta = \langle \mathrm{d}H/\mathrm{d}\theta\rangle = \sin\theta \sum_i \langle H_i^{(2)}\rangle - \cos\theta \sum_i \langle H_i^{(3)}\rangle$, versus θ. Phase transitions occurs when the energy function is not differentiable. Vertical lines in orange are located at the exact phase transition points [153]. We use the 1D RNN with 3 layers, 36 hidden dimensions and the real-imaginary parameterization. We use the transfer learning technique where we first train the neural network on a 32-anyon model for 3000 iterations and then we transfer the neural network to the 40-anyon model for another 3000 iterations. In each iteration, we use 12000 samples. The neural network architecture, initialization, and optimization details are discussed in Supplementary Materials 4.6.4.

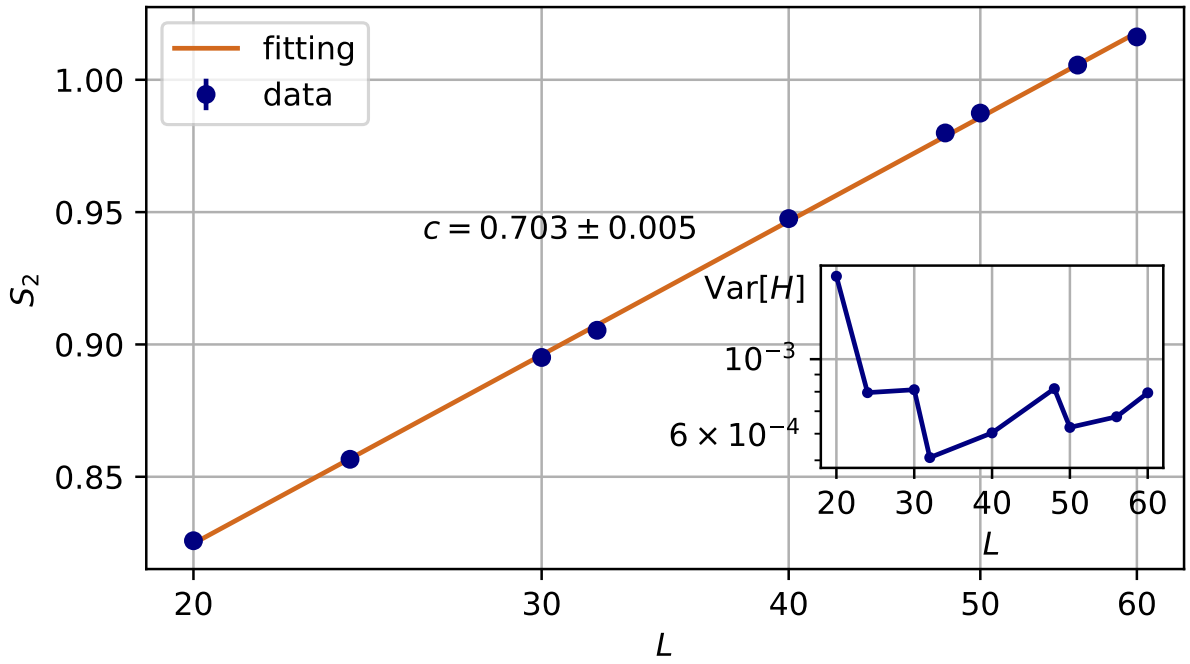


Figure 4.14: The second Renyi entropy S_2 versus the system size L for the optimized AR-NN for the Fibonacci anyons in a golden chain ($\theta = 0$) with the periodic boundary condition. The inset shows the variance of energy of the AR-NN. The slope of the fitted line is the central charge c. The AR-NN is the 1D RNN with 3 layers, L hidden dimensions, and the amplitude-phase parameterization. The neural network is trained for 8000 iterations with a sample size of 12000. The neural network architecture, initialization, and optimization details are discussed in Supplementary Materials 4.6.4.

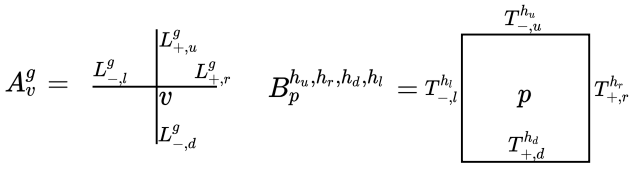


Figure 4.15: A_v^g and $B_p^{h_u,h_r,h_d,h_l}$ operators. $A_v^g = L^g_{+,u}L^g_{+,r}L^g_{-,d}L^g_{-,l}$ and $B_p^{h_u,h_r,h_d,h_l} = T^{h_u}_{-,u}T^{h_r}_{+,r}T^{h_d}_{+,d}T^{h_l}_{-,l}$, where $L^g_+ |g\rangle = |gz\rangle$, $L^g_- |g\rangle = \left|zg^{-1}\right\rangle$, $T^h_+ |z\rangle = \delta_{h,z} |z\rangle$ and $T^h_- |z\rangle = \delta_{h^{-1},z} |z\rangle$.

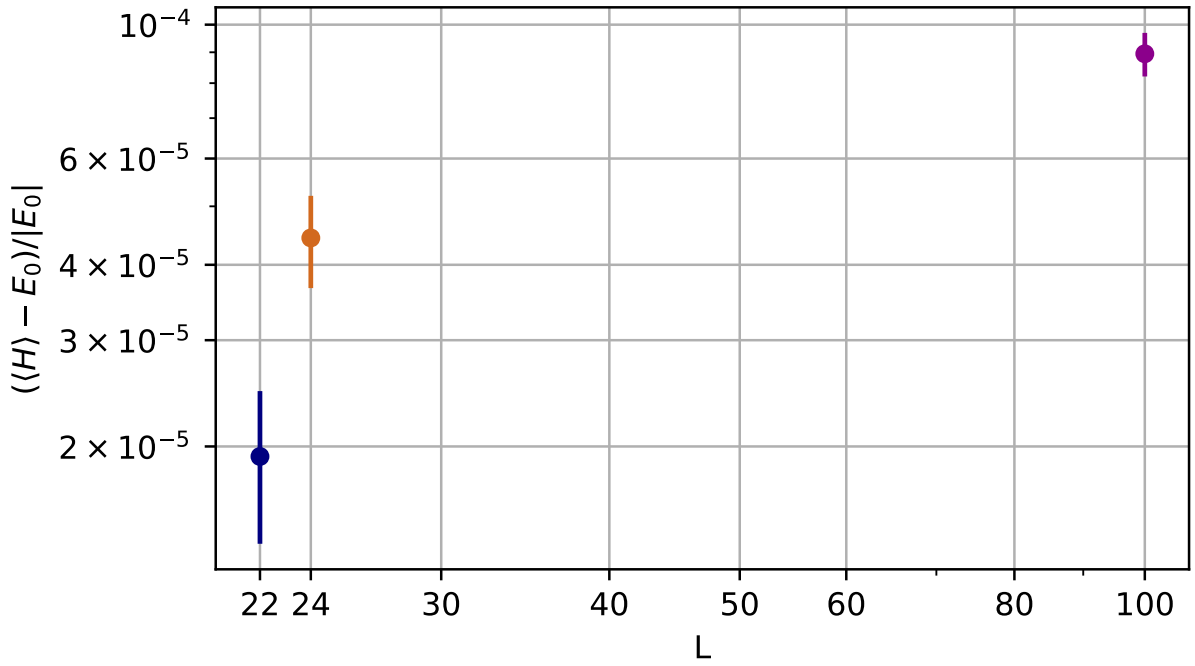


Figure 4.16: Relative error of variational ground state energy for the 1D Heisenberg model with SU(2) symmetry for 22, 24 and 100 spins. We use the 1D RNN with 3 layers, L hidden dimensions and the real-imaginary parameterization. We train the neural network for 5000 iterations with 12000 samples in each iteration. The exact solutions for 22 spins and 24 spins are computed with the exact diagonalization, and the exact solution for 100 spins is the DMRG result in Ref. [122].

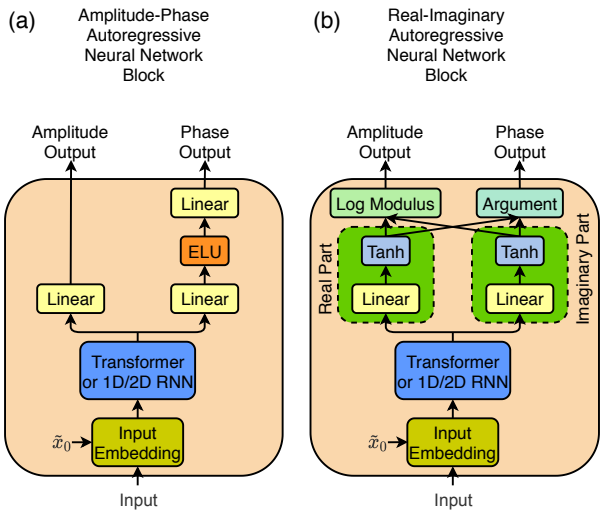


Figure 4.17: Two parameterizations of complex wave functions from autoregressive neural networks. (a) The amplitude-phase parameterization. The raw output is used as the input of both the amplitude branch and the phase branch. (b) The real-imag parameterization. The raw output is used as the input of both the real branch and the imaginary branch, which later is converted to the amplitude branch and the phase branch.

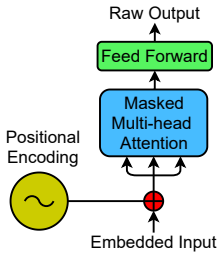


Figure 4.18: A single layer Transformer network. The embedded input is fed into the Transformer and the positional encoding is added. After a masked multi-head self-attention is applied, a feed forward layer produces the raw output.

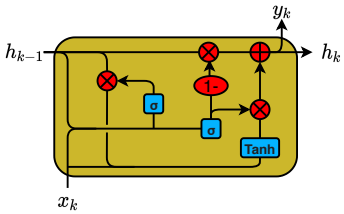


Figure 4.19: The GRU cell [142] on which different RNNs are constructed. This is the same GRU cell as the PyTorch [156] implementation.

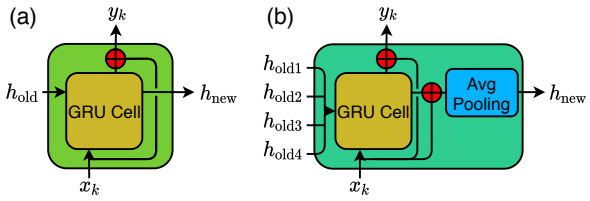


Figure 4.20: (a) 1D RNN cell. A ResNet (skip connection) [173] is added between the input x_k and output y_k. To be noted that x_k, y_k, h_{old} and h_{new} has the same dimension. (b) 2D RNN cell with periodic boundary condition. This cell requires four hidden inputs $h_{\text{old}i,i=1,2,3,4}$ and generates one hidden output h_{new}. Skip connections are added for both output y_k and hidden output h_{new}. The average pooling reduces the hidden output to have the same dimension as each hidden input. To be noted that the dimension of x_k and y_k is four times the dimension of each $h_{\text{old}i}$ and h_{new}.

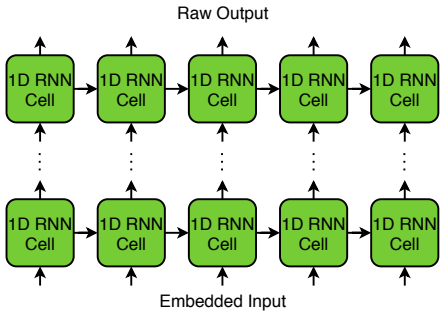


Figure 4.21: 1D RNN built from 1D RNN Cells. The neural network has a multi-layer design similar to Pytorch GRU implementation [156, 142]. The weight matrices and biases are shared between different layers.

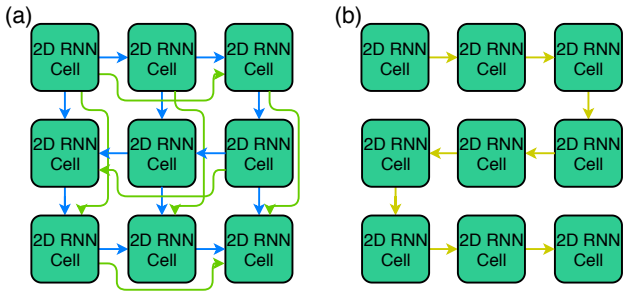


Figure 4.22: (a) The hidden information path for 2D RNN with periodic boundary condition for a 3×3 system. Blue arrows show the non-boundary information path whereas the green arrows show the periodic boundary information path. When there are less than four hidden inputs, zero vectors are used for padding. (b) The conditioning and sampling order of the 2D RNN.

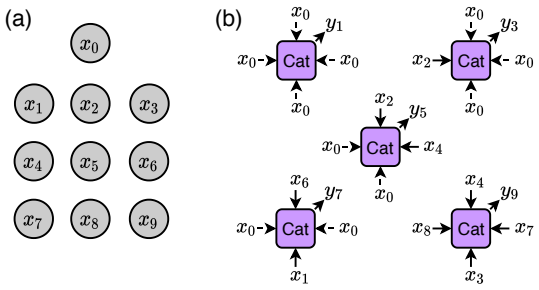


Figure 4.23: 2D RNN input concatenation layer. For (a) a 3×3 input array with a default input x_0, (b) the concatenation layer takes the four input vectors surrounding each site with periodic boundary condition and outputs the concatenated vector of the four surrounding vectors. x_0 is used when the surrounding inputs appear later in the conditioning order.

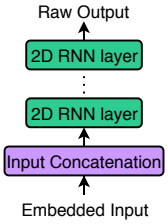


Figure 4.24: The 2D RNN is built from one input concatenation layer and multiple 2D RNN layers. The weight matrices and biases are shared between different layers.

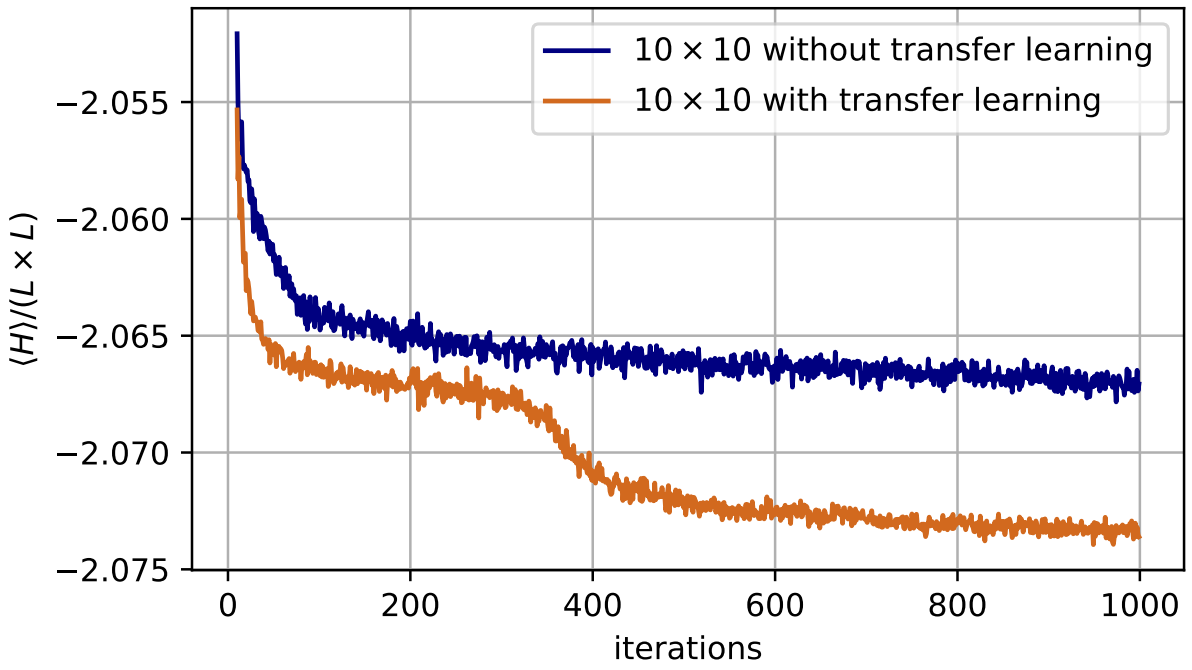


Figure 4.25: Per site energy with and without transfer learning during the training process for the 10×10 toric code model. The first 10 iterations are not included as they are too large and outside the range of the figure. The energy with transfer learning is clearly lower than the energy without transfer learning.

Chapter 5

Spacetime Neural Network for High Dimensional Quantum Dynamics

This chapter is written based on the research in Ref. [175]. We develop a spacetime neural network method with second order optimization for solving quantum dynamics from the high dimensional Schrödinger equation. In contrast to the standard iterative first order optimization and the time-dependent variational principle, our approach utilizes the implicit mid-point method and generates the solution for all spatial and temporal values simultaneously after optimization. We demonstrate the method in the Schrödinger equation with a self-normalized autoregressive spacetime neural network construction. Future explorations for solving different high dimensional differential equations are discussed.

5.1 Introduction

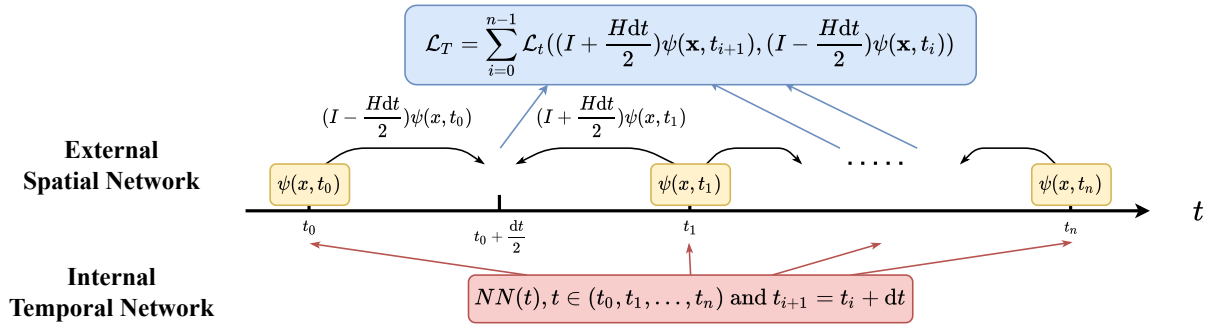


Figure 5.1: Second order optimization with spacetime neural network. The spacetime neural network consists of an internal temporal network and an external spatial network. The internal temporal network provides time dependent weights for the external spatial network at each time step. The loss function is computed over all time steps and is used to train the spacetime neural network.

Differential equation plays a fundamental role in science and engineering. Among different differential equations, the Schrödinger equation is a famous example of high dimensional differential equations that describes quantum dynamics of a physical system. The real time Schrödinger equation is given by

$$i\frac{\partial \psi(\mathbf{x}, t)}{\partial t} = H\psi(\mathbf{x}, t) \tag{5.1}$$

and the imaginary time version is

$$\frac{\partial \psi(\mathbf{x}, t)}{\partial t} = -H\psi(\mathbf{x}, t) \tag{5.2}$$

where $\psi(\mathbf{x}, t)$ is a complex-valued function whose dimensionality grows exponentially with respect to the number of particles ($\mathbf{x}$ is the configuration of the particles) and H is the Hamiltonian of the physical system.

With the recent advancement of quantum science and engineering, the research of quantum dynamics is becoming an important topic. Quantum dynamics takes place in various areas, such as photosynthesis, chemical reactions, cold atom experiments, and quantum computation [176, 177, 178, 179]. New approaches for solving the Schrödinger equation will provide powerful tools to understand and explore quantum dynamics with various applications.

In recent years, the advancement of machine learning opens up new possibilities for solving the high dimensional Schrödinger equation [180]. The standard approach is to compactly represent the high-dimensional wave function $\psi(\mathbf{x}, t)$ at a single moment in time t using a neural network. The differential equation then indicates how to update in time the full high-dimensional wave function. One would typically optimize a new neural network for time $t + \mathrm{d}t$ which compactly represents this propagated state. This is accomplished by optimizing the neural network representation at time $t + \mathrm{d}t$ to maximally match the non-compact state propagated at first order.

In this work, we develop an alternative approach. Instead of building the wave function snapshot by snapshot, we develop an approach which simultaneously compresses the entire space-time wave function. This is accomplished by having an internal temporal network which takes a time t and outputs the parameters for a second external spatial network which compactly represents the full wave function at that time t as an autoregressive Transformer [118]. The whole spacetime neural network is then optimized using a second order formulation with an implicit mid-point method. Our work makes a direct connection with the path-integral formulation of quantum mechanics that is used to describe quantum field theories [181].

In Sec. 5.2, we describe the standard first-order single time-slice approach. We then (Sec. 5.3 and 5.4) go on to describe our novel space-time formulation. Our approach (Sec. 5.5) was tested on a 12-spin Heisenberg model with imaginary time evolution and achieved good performance. Even though the current work is demonstrated with imaginary time quantum dynamics, our approach is general and applies to both real time dynamics as well as more general classes of differential equations.

5.2 First Order Optimization and Time-dependent Variational Principle

One way to solve the Schrödinger equations in quantum dynamics is to introduce a loss function based on the first order Euler method and minimize the loss function with respect to the neural network representation [160]. This can be used as a projective method which iteratively updates the neural network at each time step. For example, in the case of the imaginary time evolution of the Schrödinger equation,

$$\mathcal{L}_1^{(t)} = \sum_{\mathbf{x}} |\psi(\mathbf{x}, t + \mathrm{d}t) - (I - H\mathrm{d}t)\psi(\mathbf{x}, t)|^2 \tag{5.3}$$

where I is the identity operator. For each t, $\mathcal{L}_1^{(t)}$ is optimized with respect to $\psi(\mathbf{x}, t + \mathrm{d}t)$ using stochastic gradient descent while $\psi(\mathbf{x}, t)$ is fixed without taking gradient. This procedure is then iterated where time $t + \mathrm{d}t$ becomes the new time t.

Minimizing Eq. 5.3 is equivalent to introducing a nonlinear differential equation in the parameter space [182]. For a fixed time t, by denoting the parameters for the neural network representation of $\psi(\mathbf{x}, t)$ as θ_t, the time-dependent variational principle [182] claims $\frac{d\theta_t}{\mathrm{d}t} = -F^{-1}\nabla_{\theta_t} E$, where $E = \sum_{\mathbf{x}} \psi^*(\mathbf{x}, t) H \psi(\mathbf{x}, t)$ and $F = \sum_{\mathbf{x}} \nabla_{\theta_t} \psi^*(\mathbf{x}, t) \nabla_{\theta_t} \psi(\mathbf{x}, t)$ is the quantum Fisher information matrix. Furthermore, a first order discretization with Euler method gives rise to the gradient update formula with a learning rate of $\mathrm{d}t$,

$$\theta_{t+\mathrm{d}t} = \theta_t - \mathrm{d}t F^{-1} \nabla_{\theta_t} E \tag{5.4}$$

If one views E as a loss function, the above formula is actually the natural gradient method [183] in the context of quantum mechanics. Since the natural gradient method has information beyond first order optimization, it implies that a first order optimization in terms of time step with respect to the function in the wave function space (Eq. 5.3) gives rises to an optimization beyond first order in the parameter space (Eq. 5.4).

5.3 Second Order Optimization with Spacetime Formulation

An implicit mid-point method which generalizes Eq. 5.3 to second order (Eq. 5.5) has been recently used for real time evolution of the Schrödinger equation [184]. Notice that the mid-point method is generic and not limited to the real time Schrödinger equation. It is known that the second order implicit mid-point method has the advantage of preserving the sympletic form of the Hamiltonian dynamics as well as be more stable

with larger time step $\mathrm{d}t$ compared to the first order Euler method [185]. For each t, $\mathcal{L}_2^{(t)}$ is optimized with respect to $\psi(\mathbf{x}, t+\mathrm{d}t)$ using stochastic gradient descent while $\psi(\mathbf{x}, t)$ is fixed without taking gradient. This procedure is iterative as in the case of the first order optimization with Eq. 5.3.

$$\mathcal{L}_2^{(t)} = \sum_{\mathbf{x}} \left| (I + \frac{iH\mathrm{d}t}{2})\psi(\mathbf{x}, t+\mathrm{d}t) - (I - \frac{iH\mathrm{d}t}{2})\psi(\mathbf{x}, t) \right|^2 \tag{5.5}$$

It is a common practice to represent $\psi(\mathbf{x}, t)$ with a neural network for each t and perform iterative optimization with approaches in Eq. 5.3, 5.4, 5.5 in the community of neural network quantum states [180, 160, 184]. The neural network in the above approaches essentially only represents the spatial part of the wave function because different neural networks are required for different t. Although the above approaches are flexible and useful, the number of copies of neural network increases linearly with the number of time step.

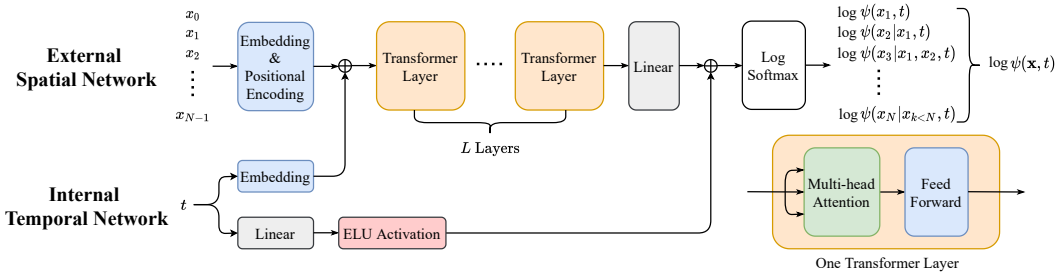


Figure 5.2: Architecture of autoregressive spacetime neural network. The neural network has an external spatial network and an internal temporal network. The internal temporal network takes the time as the input and generates the relevant weights for the external spatial network. The external spatial network is a Transformer which utilizes these weights to generate the log wave function.

In this work, we propose a second order optimization method with spacetime formulation, which makes use of the implicit mid-point method and the full spatial and temporal neural network representation. Instead of parameterizing the wave function $\psi(\mathbf{x}, t)$ with a neural network for each discrete time point t, we construct a spacetime neural network that represents $\psi(\mathbf{x}, t)$ for all $\mathbf{x}$ and t altogether. We further utilize the second order implicit mid-point formulation and define a new loss. For example, in the context of imaginary time evolution of the Schrödinger equation over a total time T, we have

$$\mathcal{L}_T = \sum_t^T \mathcal{L}_t((I + \frac{H\mathrm{d}t}{2})\psi(\mathbf{x}, t+\mathrm{d}t), (I - \frac{H\mathrm{d}t}{2})\psi(\mathbf{x}, t)) \tag{5.6}$$

where $\mathcal{L}_t(\psi_1(\mathbf{x}, t), \psi_2(\mathbf{x}, t))$ is a loss function between two wave functions $\psi_1(\mathbf{x}, t)$ and $\psi_2(\mathbf{x}, t)$ for a fixed t.

In this work, we choose $\mathcal{L}_t$ to be the log overlap function such that

$$\mathcal{L}_t(\psi_1(\mathbf{x},t),\psi_2(\mathbf{x},t)) = -\log \frac{|\langle\psi_1,\psi_2\rangle|^2}{\langle\psi_1,\psi_1\rangle\langle\psi_2,\psi_2\rangle} \tag{5.7}$$

where $\langle\psi_1,\psi_2\rangle = \sum_{\mathbf{x}} \psi_1^*(\mathbf{x},t)\psi_2(\mathbf{x},t)$. Notice that the key difference between Eq. 5.6 and Eq. 5.5 is that Eq. 5.6 is defined over all time t and $\psi(\mathbf{x},t)$ is represented by only one spacetime neural network. For $T = \mathrm{d}t$ as one time step, Eq. 5.6 will reduce to the imaginary time version of Eq. 5.5 by choosing $\mathcal{L}_t$ as the L_2 norm. Fig. 5.1 demonstrates the high level picture of the second order optimization spacetime neural network. Our spacetime neural network consists of internal temporal network and external spatial network and the details will be given in the following section. The internal temporal network generates a series of external spatial networks to represent a set of $\{\psi(\mathbf{x},t_i)\}$, which are used in the loss function Eq. 5.6. To optimize Eq. 5.6 with high dimensional $\psi(\mathbf{x},t)$, we use exact sampling for configuration $\mathbf{x}$ and take a uniform discretization in $[0,T]$ with step $\mathrm{d}t$ and sum over t.

The formulation of Eq. 5.6 provides a second order method for optimizing the spacetime neural network $\psi(\mathbf{x},t)$, which is able to predict values of $\psi(\mathbf{x},t)$ for any $\mathbf{x}$ and t after the optimization. Even though the main discussion of our work is in the context of high dimensional Schrödinger equation, the approach could be extended to other high dimensional differential equations.

5.4 Architecture of Spacetime Neural Network

In this section, we propose a self-normalized spacetime neural network architecture for solving the high dimensional Schrödinger equation. Consider a wave function $\psi(\mathbf{x},t)$ of N spin particles for a fixed t; its dimensionality grows exponentially as 2^N. In order to resolve the high-dimensionality issue presented by quantum mechanics, we need a compact representation for the wave function. In addition, the wave function is a complex-valued function that is L_2 normalized, which requires that $\sum_{\mathbf{x}} |\psi(\mathbf{x},t)|^2 = 1\ \forall t$. For a wave function $\psi(\mathbf{x})$ with only spatial dependence, recent progress [186] in autoregressive models provides a method for representing $\psi(\mathbf{x})$ with polynomial number of parameters and maintaining the normalization condition $\sum_{\mathbf{x}} |\psi(\mathbf{x})|^2 = 1$. The key idea behind the autoregressive model is to factorize the wave function into conditional wave functions on previous sites, such that $\psi(\mathbf{x}) = \psi(x_1,...,x_N) = \prod_{k=1}^{N} \psi(x_k|x_{k-1},...,x_1)$. One can normalize the high dimensional wave function $\psi(\mathbf{x})$ by normalizing all the conditional wave function distributions $\sum_{x_1,...,x_k} |\psi(x_k|x_{k-1},...,x_1)|^2 = 1$ for all $k \leq N$. In this work, we generalize the autoregressive

model to a spacetime neural network wave function such that

$$\psi(\mathbf{x},t)=\prod_{k=1}^{N}\psi(x_k|x_{k-1},...,x_1,t) \tag{5.8}$$

This construction ensures $\psi(\mathbf{x},t)$ is normalized for each t. Therefore, for each t, $\mathbf{x}$ can be sampled exactly according to the conditional probability distribution of the wave function, making it more efficient compared to Markov chain Monte Carlo sampling for high dimensional wave functions.

To realize Eq. 5.8, we construct the spacetime neural network with an internal temporal network and an external spatial network as Fig. 5.2 shows. For the external spatial network, we use a Transformer since it has the desired autoregressive property and can compactly represent high dimensional distributions. Transformers [145] were introduced in 2017 for natural language processing and later were found applicable to many different areas. The external spatial network consists of an embedding layer, a positional encoding layer, L transformer layers ($L=1$ in our experiment), a linear layer and a log softmax layer. Each transformer layer is composed of a multi-head attention layer and a feed-forward layer. For our implementation, we choose 16 as the hidden dimension for all transformer layers and linear layers, and we set the number of attention heads to 8. The internal temporal network takes the current time t as input and adds time dependencies to the spatial network after the embedding layer and before the log softmax layer. The neural network takes the N-particle configuration $\mathbf{x}=(x_0,x_1,...,x_{N-1})$ and the current time t as inputs, and outputs the log of the wave function $\psi(\mathbf{x},t)$ at time t.

5.5 Numerical Experiments

We benchmarked our spacetime neural network on a Heisenberg model with 12 spins for imaginary time evolution. The Hamiltonian H of the Heisenberg model is given by:

$$H=-J\left(\sum_i\sigma_i^x\sigma_{i+1}^x+\sum_i\sigma_i^y\sigma_{i+1}^y\right)+\sum_i\sigma_i^z\sigma_{i+1}^z \tag{5.9}$$

where J is the coupling constant and $\sigma_i^x,\sigma_i^y,\sigma_i^z$ are Pauli matrices acting on site i. We set $J=1$, $\mathrm{d}t=0.01$ and $T=2$ for solving $\psi(\mathbf{x},t)$, where $\mathbf{x}$ is the spin configuration of the 12 sites. The optimization is performed using Adam optimizer for 150 steps.

We use the absolute overlap value between the spacetime neural network and the exact solution, i.e. $|\sum_{\mathbf{x}}\psi^*(\mathbf{x},t)\psi_{\text{exact}}(\mathbf{x},t)|$, to measure the accuracy of the simulation of imaginary time evolution. The exact solution is obtained from exact diagonalization. We also compute the exact and the predicted energy for the

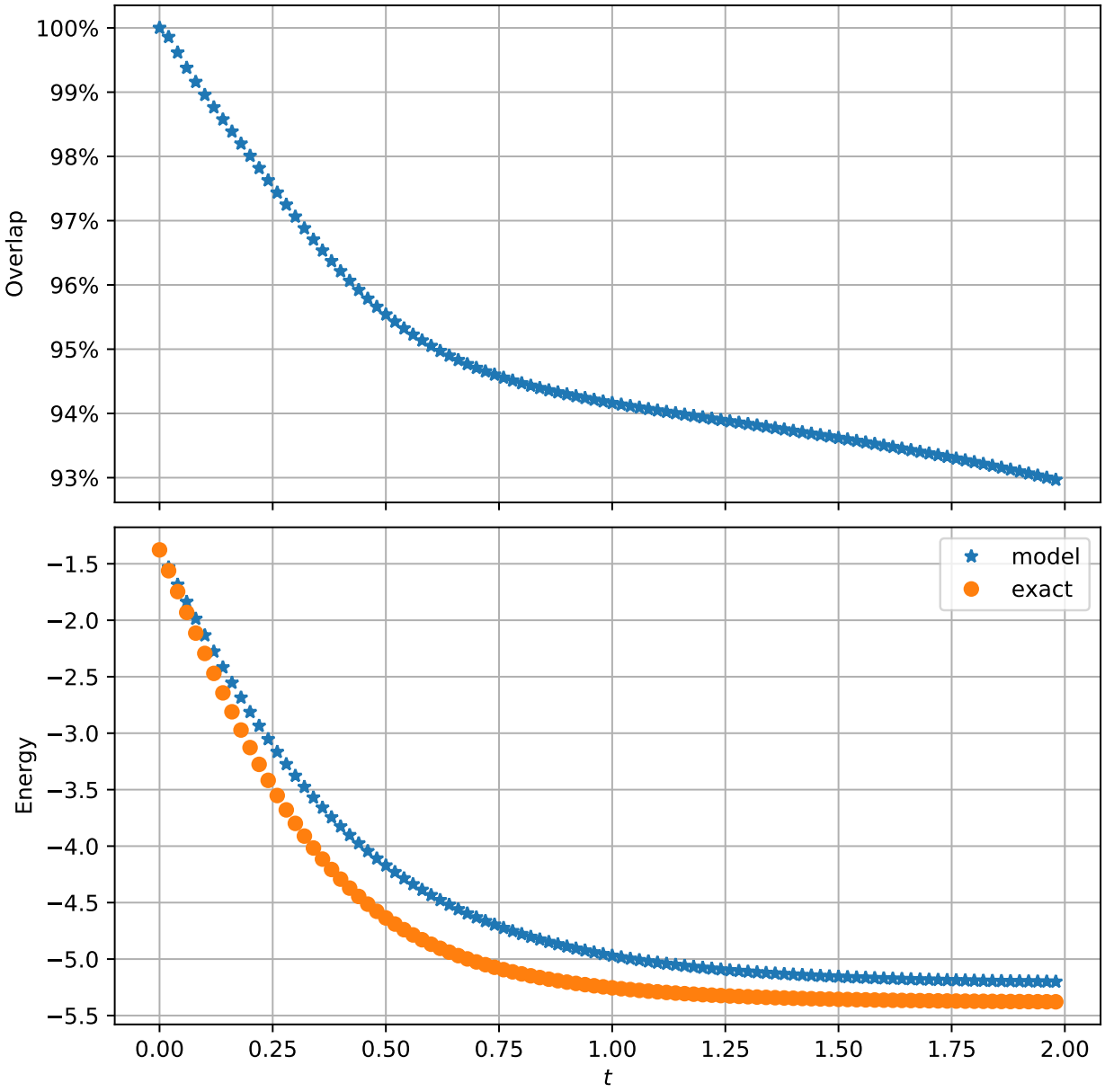


Figure 5.3: Overlap and energy comparison between spacetime neural network and the exact solution on a 12 spins Heisenberg model.

wave functions at different time steps. The energy is calculated as $E(t) = \sum_{\mathbf{x}} \psi^*(\mathbf{x}, t) H \psi(\mathbf{x}, t)$, where H is the Hamiltonian of the system. Our experiment results are shown in Fig. 5.3 where the top plot shows the overlap and the bottom plot shows the energy. It is found that our method provides good overlap and close match in the energy to the exact solution.

5.6 Discussions and Conclusion

We have introduced a spacetime neural network with a second order optimization for solving high dimensional Schrödinger equations. The advantages of the approach come from optimizing $\psi(\mathbf{x}, t)$ for all $\mathbf{x}$ and t with a second order optimization formulation. Our spacetime neural network is autoregressive and self-normalized for each t, which enables efficient exact sampling in high dimension. Contrasted to the standard method of obtaining the wave function at discrete times with iterative projective method, our spacetime neural network $\psi(\mathbf{x}, t)$ stores all spatial and temporal information, which is indeed equivalent to obtaining the path integral in quantum mechanics. It further allows to compute observables correlated in time, such as the Green function.

Even though the current experiments are performed for quantum dynamics with the imaginary time Schrödinger equation, our approach is general and it could be applied to differential equations in both quantum and classical physics contexts. One next step is to explore the formulation in the situation of real time quantum dynamics, where there could be more oscillation in the dynamics. Our work is also connected to recent work [187, 188, 189] in the applied mathematics community which aims to solve differential equations $\frac{\partial u(\mathbf{x},t)}{\partial t} = Lu(\mathbf{x}, t)$ (L is an operator) by parameterizing $u(\mathbf{x}, t)$ with neural networks. It will be interesting to apply our approach to different scenarios, such as the Hamilton-Jacobi equation, the high-dimensional master equation and the Black-Scholes model. It is also anticipated that improvements on the architecture of the spacetime neural network will be helpful for solving differential equations in different setups.

Another interesting direction is to explore the connection between our spacetime neural network and the neural ODEs [190]. One can use neural ODEs to generate the spacetime neural network for different t and consider the optimization using adjoint method. We believe that our work opens up new opportunities for research in machine learning, applied mathematics, and physics.

Chapter 6

Probabilistic Simulation of Quantum Circuits with the Transformer

This chapter is written based on the research in Ref. [191]. 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 quasi-stochastic 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 with the Transformer, a powerful ansatz responsible for the most recent breakthroughs in natural language processing. We demonstrate our approach by simulating circuits which build GHZ 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.

6.1 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 [192], 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 non-positive quasi-probabilities. In fact, quantum computers will display potential speed-ups over their classical counterparts at the onset of negative values in the quasi-probabilities associated with the description and evolution of their quantum states. This observation about non-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 non-positive quasi-probabilities 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 strings of characters results in a sensible conversation. Recently, such models have been used in the context of quantum state reconstruction [193]. 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 [193].

In order 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 take a states amplitude (or square) but this loses critical phase information. Although the presence of negative quasi-probabilities is often linked to intrinsically quantum phenomena with no classical counterpart like entanglement and quantum interference, a purely probabilistic representation of the quantum state is possible [194, 195, 196, 193, 197]. 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 [198], informational completeness means that the outcome statistics of such a measurement specifies the quantum state unambiguously.

To compactly represent these probability distribution, we will use an autoregressive model to store the instantaneous state in its probabilistic representation. We then develop powerful stochastic algorithm to update the probabilistic model representing the quantum state under the application of unitary dynamics. It is worth noting that other approaches [199, 200, 201, 202, 203, 200, 204, 205] 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 [206, 207]. 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 of the simulations as well as affects the quality of the estimators used to update the models. Transformers 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 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 new probabilistic avenue for simulation of quantum circuits, as well as quantum channels and quantum dynamics more broadly.

6.2 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 $\{M^{(a)}\}_{a\in\{1..m\}}$, of positive semi-definite operators $M^{(a)} \geq 0$, each one labeled by a measurement outcome $a = 0, 1, .., m-1$ [see Fig.6.1(a) where we describe our representation through the lens of tensor networks and its graphical notation [208]]. Following Ref. [193], we construct N-qubit measurements as tensor products of the single-qubit IC-POVM elements $\boldsymbol{M} = \left\{M^{(a_1)} \otimes M^{(a_2)} \otimes \ldots M^{(a_N)}\right\}_{a_1,\ldots a_N\in\{1..m\}^N}$, as graphically depicted in Fig.6.1(a). We choose for our numerical simulations the 4-Pauli IC-POVM measurement described in Ref.[193], $\{M^{(0)} = \frac{1}{3}|0\rangle\langle 0|, M^{(1)} = \frac{1}{3}|+\rangle\langle +|, M^{(2)} = \frac{1}{3}|r\rangle\langle r|, M^{(3)} = \mathbb{1} - M^{(0)} - M^{(1)} - M^{(2)}\}$. Here $|0\rangle, |+\rangle, |r\rangle$ are the $+1$ eigenvector with respect to $\sigma^x, \sigma^y, \sigma^z$ 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

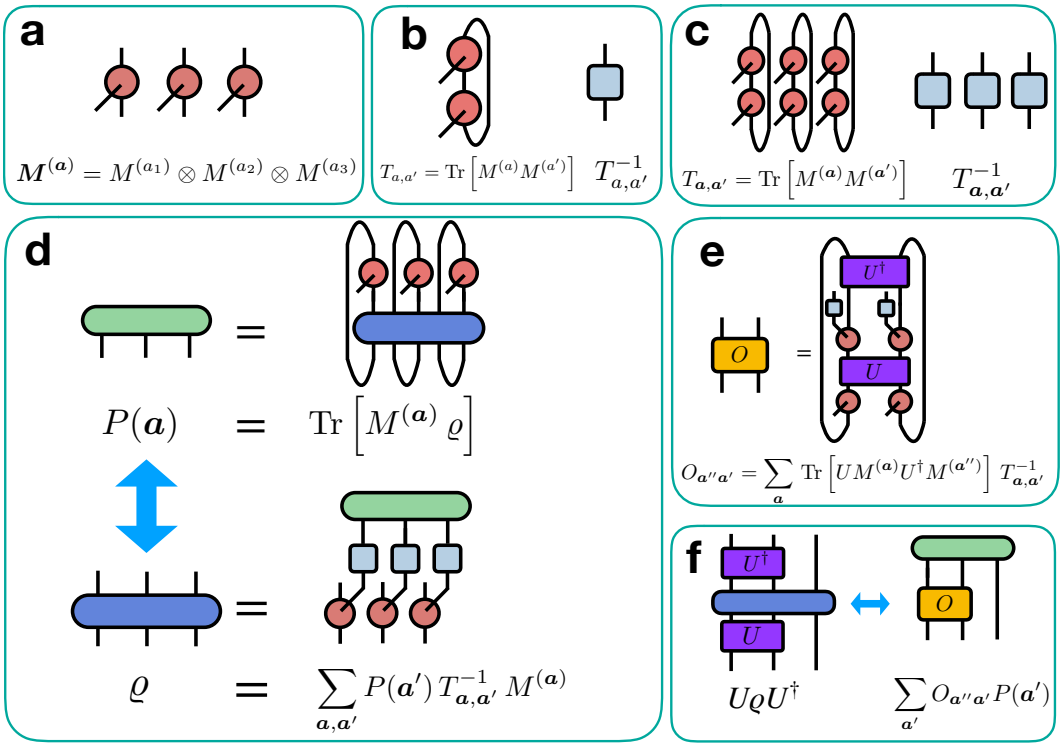


Figure 6.1: Tensor-network representation of the mapping between quantum states (gates) and probability distributions (quasi-stochastic matrices) used in this work. (a) N-qubit measurement $\boldsymbol{M} = \left\{M^{(a_1)} \otimes M^{(a_2)} \otimes \ldots M^{(a_N)}\right\}_{a_1,\ldots a_N}$ made from ($N = 3$) 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 $T_{\boldsymbol{a_1},\boldsymbol{a}'_1}$ and its inverse T^{-1}(light blue) (c) multi-qubit 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 quasi-stochastic matrices (yellow). (f) Application of a unitary matrix to a density matrix corresponds to applying a quasi-stochastic matrix to P.

quantum state ϱ is given by the following

$$P(\boldsymbol{a}) = \mathrm{Tr}\left[M^{(\boldsymbol{a})}\,\varrho\right] \tag{6.1}$$

which is graphically explained in Fig.6.1(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

$$\varrho = \sum_{\boldsymbol{a},\boldsymbol{a}'} P(\boldsymbol{a}')\, T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\, M^{(\boldsymbol{a})}, \tag{6.2}$$

where T represents the overlap matrix given by $T_{\boldsymbol{a},\boldsymbol{a}'} = \mathrm{Tr}\left[M^{(\boldsymbol{a})}M^{(\boldsymbol{a}')}\right]$. See Fig.6.1(d) for a graphical representation of elements in Eq.6.2.

To study quantum circuits, we first have to translate the action of a quantum gate on the density matrix to 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:

$$P_U\left(\boldsymbol{a}''\right) = \mathrm{Tr}\left[U\varrho U^\dagger M^{(\boldsymbol{a}'')}\right] = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}') \tag{6.3}$$

where

$$O_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} \mathrm{Tr}\left[UM^{(\boldsymbol{a})}U^\dagger M^{(\boldsymbol{a}'')}\right] T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} \tag{6.4}$$

is a *somewhat* stochastic matrix since the values in each column add up to 1 but its entries can be positive or negative [209, 210, 195, 196]. Somewhat stochastic matrices are also known as pseudo-stochastic or quasi-stochastic matrices [195, 196]. We note that the evolution described in Eq. 6.3 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 Supplementary Materials A,B,C and D).

Here we emphasize that Eq. 6.3 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 quasi-stochastic 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 quasi-stochastic 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 quasi-stochastic 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 $m^2 \times m^2$ quasi-stochastic matrix acting on outcomes a_i and a_j. The relation between the local quasi-stochastic matrices and the local unitary gates, as well as their action on a quantum state are graphically depicted in Fig.6.1(e-f) using tensor diagrams. Furthermore, the locality of $O_{\boldsymbol{a}''\boldsymbol{a}'}$ implies that traditional quantum circuit diagrams [198] 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, $U^{(1)}, U^{(2)}, \cdots, U^{(r)}$, 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. 6.4. Thus, the quantum circuit in the IC-POVM representation becomes a computation where an initial probability distribution $P(\boldsymbol{a}) = P(a_1)P(a_2)\cdots P(a_N)$ of statistically independent sets of variables $\boldsymbol{a}$ which is evolved through a series of local quasi-stochastic matrices of the form depicted in Fig.6.1(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 $i = 1, \cdots, r$. The main goal of our approach is to accurately represent the distribution $\boldsymbol{P}_r$ since it contains all the information about the quantum state after the gates have been applied and specifies the outcome of the quantum computation.

6.3 Optimization Algorithms

The strategy to learn 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 highly expressive neural network with parameters θ so that $\boldsymbol{P}_{\theta_i} \approx \boldsymbol{P}_{\boldsymbol{i}}$. 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 isn't 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. 6.2 for a depiction of the distributions involved during the simulation.

To train the model $\boldsymbol{P}_{\theta_i}$ given a gate $i+1$, we adopt a variational approach and select the parameters

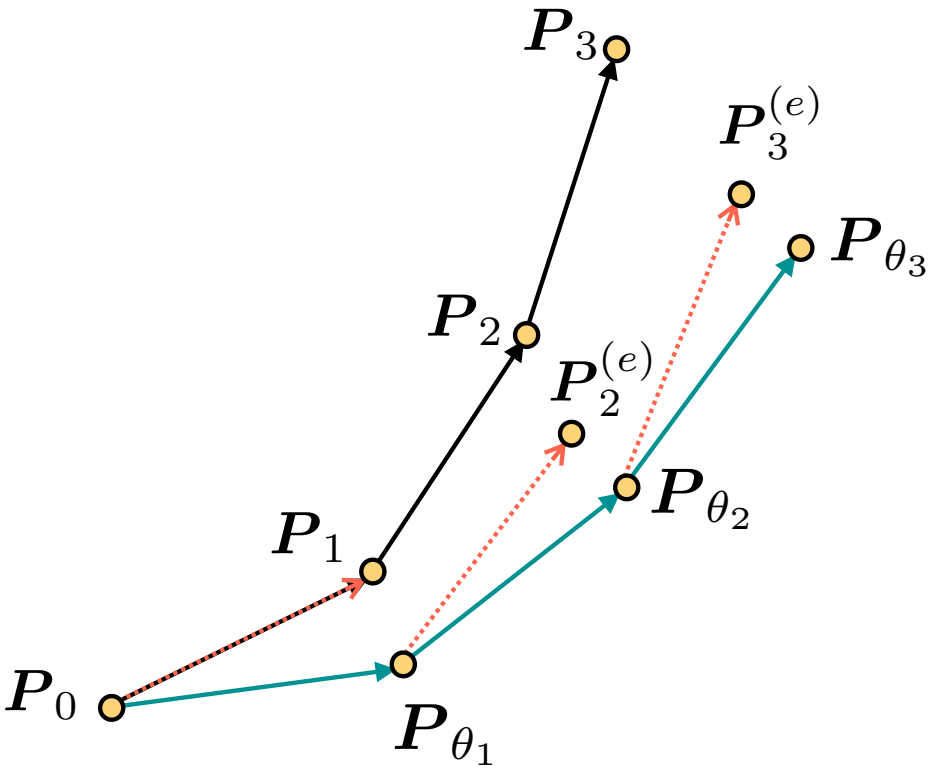


Figure 6.2: Schematics of the different distributions involved in the training procedure for a circuit with 3 gates. Black arrows indicate the exact trajectory while green arrows represent the trajectory of the optimized models $\boldsymbol{P}_{\theta_i}$. The red dotted arrows point to the exactly evolved distributions $\boldsymbol{P}_{i+1}^{(e)}$ after the application of each gate $O^{(i)}$ on the trained model $\boldsymbol{P}_{\theta_i}$. Note that if the resulting KL divergence after training $\mathrm{KL}\left(\boldsymbol{P}_{i+1}^{(e)}||\boldsymbol{P}_{\theta_{i+1}}\right) = 0$ for every gate in the circuit, then the black trajectory coincides with the green one and the procedure becomes exact.

θ_{i+1} such that the Kullback-Liebler (KL) divergence between $\boldsymbol{P}_{i+1}^{(e)}$ and $\boldsymbol{P}_{\theta_{i+1}}$,

$$\mathrm{KL}\left(\boldsymbol{P}_{i+1}^{(e)}||\boldsymbol{P}_{\theta_{i+1}}\right) = -\sum_{\boldsymbol{a}} P_{i+1}^{(e)}(\boldsymbol{a}) \log\left(\frac{P_{\theta_{i+1}}(\boldsymbol{a})}{P_{i+1}^{(e)}(\boldsymbol{a})}\right) \tag{6.5}$$

is minimized. Recall $\mathrm{KL}\left(\boldsymbol{P}_{i+1}^{(e)}||\boldsymbol{P}_{\theta_{i+1}}\right) \geq 0$, with the equality being satisfied only when $\boldsymbol{P}_{\theta_{i+1}} = \boldsymbol{P}_{i+1}^{(e)}$. To minimize the KL divergence, we will apply a variant of gradient descent (i.e. Adam [211]) where we repeatedly update the parameters of θ_{i+1} by taking steps in the direction of the gradient of Eq. 6.3. This gradient, assuming that the model is normalized, can be written as

$$\nabla_{\theta_{i+1}} \mathrm{KL}(\boldsymbol{P}_{i+1}^{(e)}||\boldsymbol{P}_{\theta_{i+1}}) \tag{6.6a}$$

$$= -\sum_{\boldsymbol{a}} P_{i+1}^{(e)}(\boldsymbol{a}) \nabla_{\theta_{i+1}} \log\left(P_{\theta_{i+1}}(\boldsymbol{a})\right) \tag{6.6b}$$

$$= -\mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})}\left[\frac{P_{i+1}^{(e)}(\boldsymbol{a})}{P_{\theta_{i+1}}(\boldsymbol{a})} \nabla_{\theta_{i+1}} \log\left(P_{\theta_{i+1}}(\boldsymbol{a})\right)\right] \tag{6.6c}$$

$$= -\mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})}\left[\left(\frac{P_{i+1}^{(e)}(\boldsymbol{a})}{P_{\theta_{i+1}}(\boldsymbol{a})} - k\right) \nabla_{\theta_{i+1}} \log\left(P_{\theta_{i+1}}(\boldsymbol{a})\right)\right]. \tag{6.6d}$$

Here, $k = \mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})}\left[\frac{P_{i+1}^{(e)}(\boldsymbol{a})}{P_{\theta_{i+1}}(\boldsymbol{a})}\right]$. Note that $\mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}(\boldsymbol{a})} \nabla_{\theta_{i+1}} \log\left(P_{\theta_{i+1}}(\boldsymbol{a})\right) = 0$, which justifies the equality in Eq. 5d. To estimate the gradient in Eq. 6.6, we generate a mini-batch 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. 5d. While computing Eq. 5d or Eq. 5c both evaluate the gradient, Eq. 5d has significantly lower variance. In the limit where $P_{\theta_{i+1}}(\mathbf{a})$ approaches $P_{i+1}^{(e)}(\mathbf{a})$ (i.e. ideally towards the end of the optimization of gate $i+1$), the variance of this gradient goes to zero. This is known as the zero variance principle [212]. $P_{i+1}^{(e)}(\boldsymbol{a}) = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}\boldsymbol{a}'}^{(i+1)} P_{\theta_i}(\boldsymbol{a}')$ is evaluated by explicitly summing over the sub-string 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 quasi-stochastic matrices considered here are k-local, which means that the calculation of the gradient estimator in Eq. 6.6 is efficient. More precisely, using N_s samples the gradient can be computed in $O(N_s k^2)$ 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

$$C = ||O_1 \boldsymbol{P}_{\theta_i} - O_1^T \boldsymbol{P}_{\theta_{i+1}}||_1 \tag{6.7a}$$

$$= \sum_{\boldsymbol{a}} |\sum_{\boldsymbol{a'}} O_{1,\boldsymbol{a}\boldsymbol{a'}} \boldsymbol{P}_{\theta_i}(\boldsymbol{a'}) - O^T_{1,\boldsymbol{a}\boldsymbol{a'}} \boldsymbol{P}_{\theta_{i+1}}(\boldsymbol{a'})| \tag{6.7b}$$

The gradient can be computed as follows

$$\nabla_{\theta_{i+1}} C = \mathbb{E}_{\boldsymbol{a}\sim P_{\theta_{i+1}}} \frac{1}{P_{\theta_{i+1}}(\boldsymbol{a})} \sum_{\boldsymbol{a'}} O^T_{1,\boldsymbol{a}\boldsymbol{a'}} \nabla_{\theta_{i+1}} P_{\theta_{i+1}}(\boldsymbol{a'}) \tag{6.8}$$

$$\text{sign}\{\sum_{\boldsymbol{a'}} O_{1,\boldsymbol{a}\boldsymbol{a'}} \boldsymbol{P}_{\theta_i}(\boldsymbol{a'}) - O^T_{1,\boldsymbol{a}\boldsymbol{a'}} \boldsymbol{P}_{\theta_{i+1}}(\boldsymbol{a'})\} \tag{6.9}$$

The details for optimization can be found in the Supplementary Materials G.

6.4 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 that allow for exact sampling. While other models such as the variational autoencoder [215] 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 Markov Chain Monte Carlo 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 modelling based on Transformer encoder blocks [2]. This neural architecture models a probability distribution $P_{\theta_{i+1}}(\boldsymbol{a})$ through the conditional $P_\theta(a_{k+1}|a_1, \ldots, a_k)$. 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. 6.3. The first and most important element is the self-attention mechanism. Self-attention takes an embedding of the measurement outcome $\boldsymbol{a}$, and computes an auto-correlation matrix where the different measurement outcomes across the different qubits form the columns and rows. The embedding is simply a linear transformation on the original

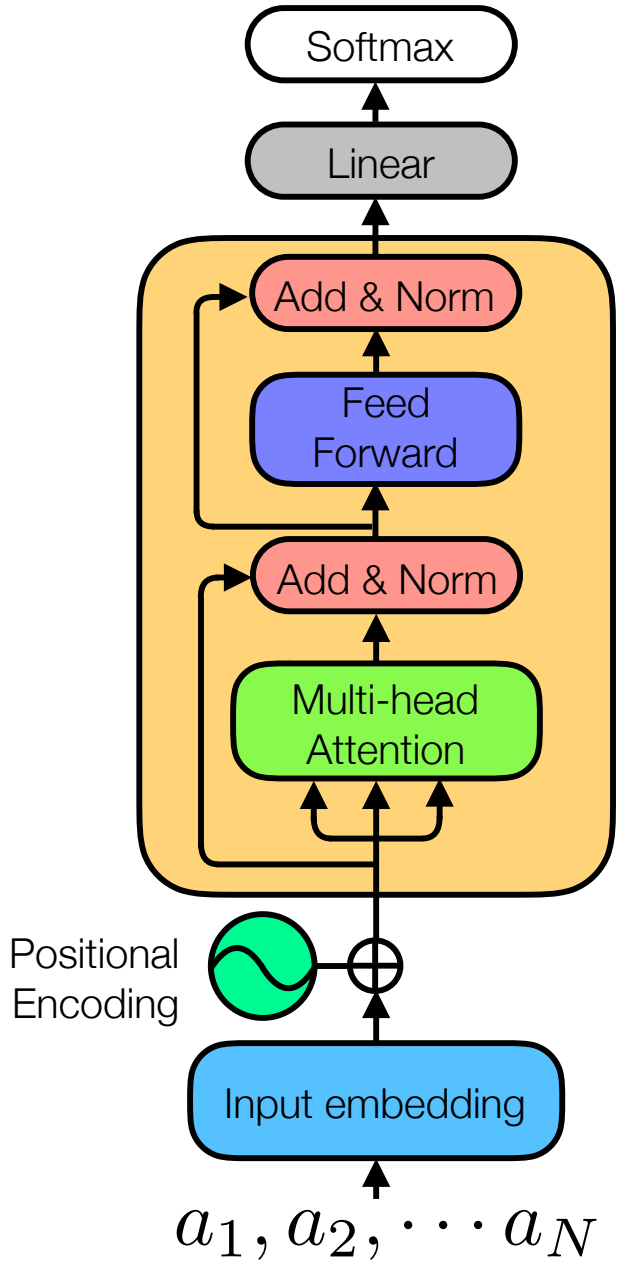


Figure 6.3: Schematic representation of the Transformer model. Lines with arrowheads denote incoming arrays from the output of one node to the inputs of others. The Transformer architecture starts with an input measurement $\boldsymbol{a}$. First, a high-dimensional linear embedding $\hat{A}$ of the input measurement is computed. This is followed by addition of positional encoding vectors to the input embeddings $\hat{A}$. The multi-head attention mechanism is applied to the modified embedding, followed by a residual connection [213] and layer normalization [214]. A position-wise feed-forward is then applied 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 $P_\theta(a_{k+1}|a_1, \ldots, a_k)$.

input $\boldsymbol{a}$, i.e. a trainable matrix multiplying the 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 [216] which induces pairwise long-distance correlations between the bare degrees of freedom (i.e. spins, qubits, electrons) in a wavefunction. 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 behaviour [206].

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 the query, keys, 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 $j = 1, \ldots, N$, 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

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V, \tag{6.10}$$

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 $\hat{A}W^{(Q)}{W^{(K)}}^{\mathrm{T}}\hat{A}^{\mathrm{T}}$, which induces pairwise, all-to-all correlations between the qubits in the system, thus resembling a Jastrow factor with parameters $W^{(Q)}{W^{(K)}}^{\mathrm{T}}$.

As in Ref. [2], we use a multi-head 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 multi-head attention is

$$\begin{aligned} &\text{Multi-Head}\left(Q, K, V\right) = \\ &\text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right) W^{(0)}, \end{aligned} \tag{6.11}$$

where $\text{head}_i = \text{Attention}\left(Q_i, K_i, V_i\right)$, $K_i = \hat{A}W_i^{(K)}$, $Q_i = \hat{A}W_i^{(Q)}$, 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 can not be known to the prior input, a mask is added in the multi-head attention.

Additionally, the Transformer features a position-wise feed-forward network, which is a fully connected feed-forward network applied to each position separately and identically. This layer consists of two linear transformations with a ReLU [217] activation in between.

Each sub-layer (i.e. the self-attention and the position-wise feed-forward network) has a residual connection around it, and is followed by a layer-normalization step. That is, the output of each sub-layer is $\text{LayerNorm}\,(x + \text{Sublayer}(x))$, where $\text{Sublayer}(x)$ is the function implemented by either the self-attention or the position-wise feed-forward network. The residual connection [213] 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 [214] is a technique to normalize the of intermediate outcome of the sub-layers to have zero mean and unit variance, which enables a more stable calculation of the gradients in Eq.6.6 and faster training. An encoder block is defined as the composition of one self-attention layer and one position wise feed-forward layer with residual connection and layer normalization as the orange part in Fig. 6.3 shows. A number of encoder blocks can be further composed to the enhanced the expressiveness of the model and the number of encoder blocks is denoted as n_{ed}.

The embedding mentioned earlier convert 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 can't naturally use the information of the 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.

6.5 Applications

6.5.1 GHZ State and Linear Graph State Preparation

We first demonstrate our approach on quantum circuits that produce 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. 6.3, 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,

$$F(\varrho_1, \varrho_2) = \mathrm{Tr}\left[\sqrt{\sqrt{\varrho_1}\,\varrho_2\,\sqrt{\varrho_1}}\right] \tag{6.12}$$

and

$$F_2(\varrho_1, \varrho_2) = \sqrt{1 - ||\varrho_1 - \varrho_2||_F^2/2}\,. \tag{6.13}$$

Here we would like to make a few remarks on the different measures 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. 6.12 is standard for comparing density matrices in quantum information science [218]. 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 don't correspond to physical density matrices. In terms of measures on POVM representation of the quantum states, the KL divergence used in Eq. 6.3 is the objective function used in optimization and hence indicates its importance. The classical fidelity F_c is also widely used in the literature [219, 220, 221, 222, 223] 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 4-Pauli POVMs. Besides the KL divergence and F_c, we will also compute the L_1 distance between two POVM representation. It is worth noticing that the L_1 distance of the classical distribution is twice 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 larger number of qubits.

Fig.6.4(a-d) shows these measures for two quantum gates (see Fig.6.4(a)[inset]) which generate the GHZ state with $N = 2$ qubits, namely the Bell state $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. 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 one 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

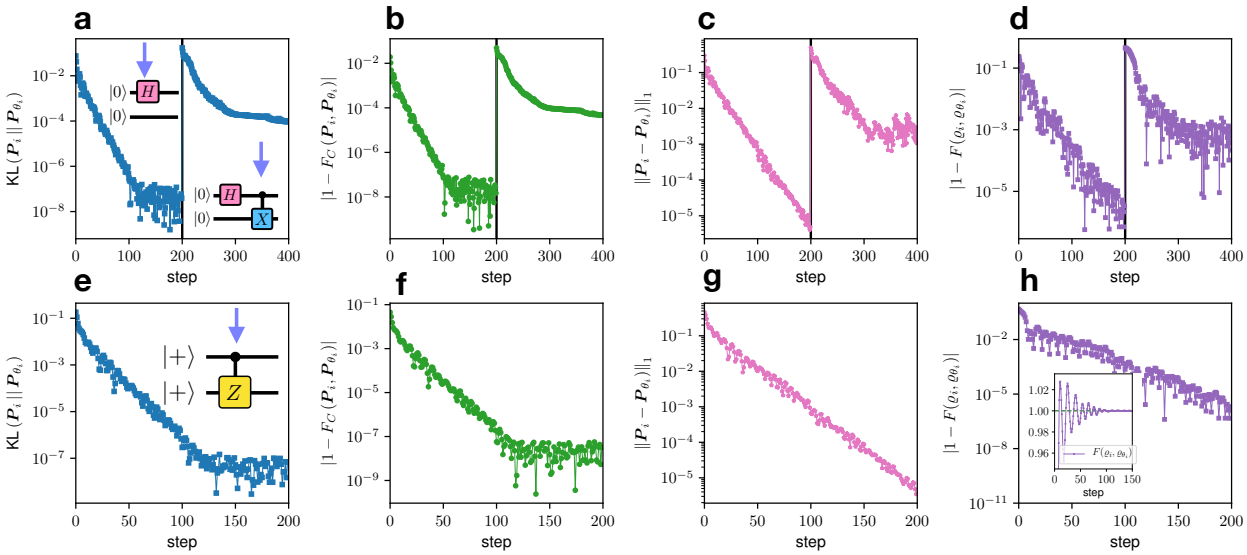


Figure 6.4: Measures of training of two qubit circuits (shown in insets of (a) and (e)) between the exact state and the quantum state represented by a Transformer ($d_{\text{model}} = 16$, $n_{ed} = 1$) after each circuit element. KL divergence (a and e), classical fidelity (b and f), L_1 norm (c and 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.

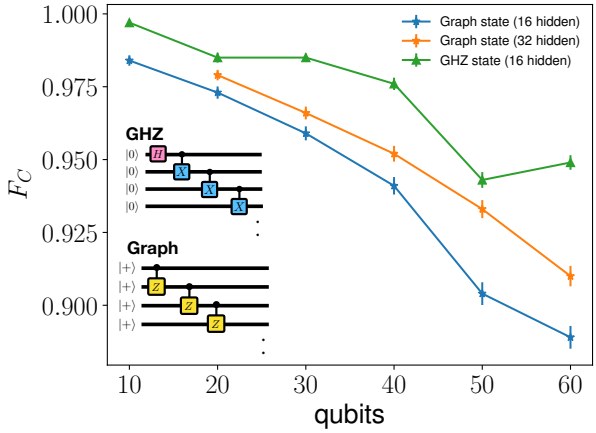


Figure 6.5: Classical fidelity F_c between the final probability distribution of the Transformer versus the exact POVM measurements of the post-circuit quantum states as a function of the total number of qubits for circuits (see insets) generating the GHZ state and linear graph states. The Transformers have $d_{\text{model}} = 16$ and 32 and $n_{ed} = 1$.

computations. Increasing the precision could also lead to improved convergence. In Fig. 6.4(e-h), we present analogous results for a circuit generating a 2-qubit graph state, i.e. $|\Psi\rangle = \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle - |11\rangle)$, where we observe similar behaviour. 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. 6.4(h) exists because the Transformer 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 we are interested in live, corresponds to physical density matrices upon inversion in Eq.6.2. The subset of probability distributions with a valid quantum states in our setting forms a convex set similar to the so-called *Qplex* in quantum Bayesian theory [224]. Here we emphasize that the fidelity values in Fig.6.4(d) and (h) eventually converge to one 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 the Supplementary Materials 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. 6.5(inset)) in the range from 10-60 qubits that the classical fidelity falls approximately linearly with number of qubits (see Fig. 6.5) 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.

6.5.2 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 [225]. We start with state preparation of a 6-qubit TFIM ground state using a variational quantum eigensolver circuit [225] (see Supplementary Materials F). The variational quantum eigensolver [226] (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

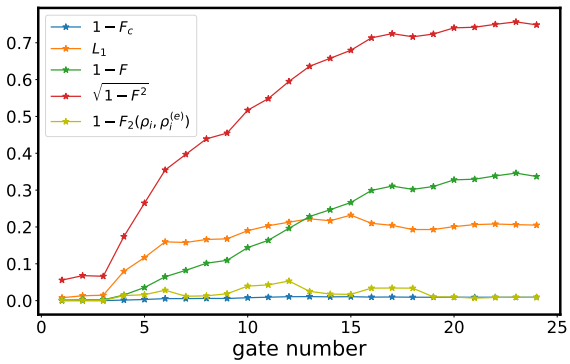


Figure 6.6: Different infidelity measurements of a VQE circuit for the preparation of a 6-qubit ground state of the TFIM as a function of the gate number.

by optimizing the KL divergence in Eq. 6.3. 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 [193], 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 quasi-stochastic gate in the circuit (see Supplementary Materials F for a precise specification of the quantum circuit and details of its probabilistic preparation). Both the classical and quantum infidelities shown in Fig. 6.6 increase with the number of gates in the circuit; in fact, as demonstrated in the Supplementary Materials F in Fig. 6.12, there is a correlation between the classical and quantum fidelity as classical fidelity approaches to 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. 6.6). 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. 6.4.

We further extend the simulation to the VQE circuits for system size $L = 8, 10, 12, 14, 16, 18$ in Ref. [225]. The simulation is performed by using the forward-backward gate algorithm through optimizing Eq. 6.7a. The details of architectures can be found in the Supplementary Materials F. It can be seen that the classical fidelity drops roughly linearly and the L_1 difference increases roughly linearly as the gate number increases.

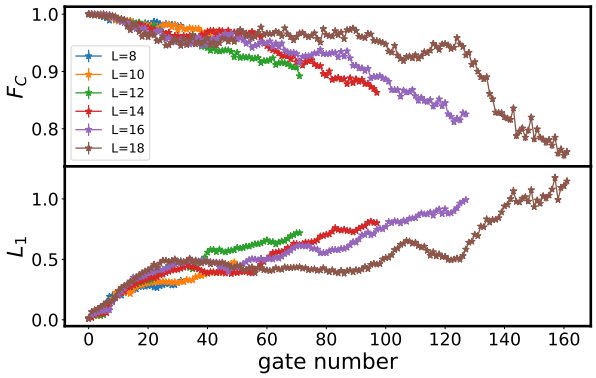


Figure 6.7: Classical fidelity and L_1 difference of VQE circuits for the preparation of ground states of the TFIM as a function of the gate number for system size $L = 8, 10, 12, 14, 16, 18$.

6.6 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 quasi-stochastic 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.

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 don't 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 quasi-stochastic matrices allowing for the design of practical algorithms. A common alternative POVM basis, the SIC-POVM [227, 194, 195, 196, 197] has an elegant formalism but is more difficult to work with algorithmically since SIC-POVM basis are not known to exist for large systems [228] and don't map local unitaries to local matrices. It is nonetheless an interesting research question whether these more complicated basis can be useful in the context of numerical simulations. Indeed, POVM is related to the Wigner-function quasi-probability 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 [229]. 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 [199, 200, 201, 202, 203, 230, 204, 205], or stochastically [231, 232]. We anticipate advantages with respect to established algorithms enabled by the ability of Transformers to model long-range correlations [206], 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 [233]) can be easily defined while retaining all the computational and modelling 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 Supplementary Materials A and B). Thus our work opens up new 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.

6.7 Supplementary Materials

6.7.1 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) maps between spaces of operators [198]. A commonly used representations of CPTP-maps is the Kraus or operator-sum representation [234] where a CPTP-map $\mathcal{E}$ acts on a quantum state ϱ as

$$\mathcal{E}(\varrho) = \sum_{\alpha}^{D} K^{(\alpha)} \varrho K^{(\alpha)\dagger}, \tag{6.14}$$

where $\sum_{\alpha}^{D} K^{(\alpha)} K^{(\alpha)\dagger} = \mathbb{1}$. The set of matrices $\{K^{(\alpha)},\ \alpha = 1, \ldots, D\}$ act on the Hilbert space of the qubits and can be thought of as an array with 3 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:

$$P_{\mathcal{E}}(\boldsymbol{a}'') = \sum_{\alpha}^{D} \mathrm{Tr}\left[K^{(\alpha)} \varrho K^{(\alpha)\dagger} M^{(\boldsymbol{a}'')}\right] = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}') \tag{6.15}$$

where

$$O_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a},\alpha} \mathrm{Tr}\left[K^{(\alpha)} M^{(\boldsymbol{a})} K^{(\alpha)\dagger} M^{(\boldsymbol{a}'')}\right] T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} \tag{6.16}$$

is a quasi-stochastic matrix since, as in the unitary case, the values in each column add up to 1 but its entries can be positive or negative [209, 210, 195, 196]. In Ref. [197], a similar formulation specific to SIC-POVM has also been proposed to describe quantum channel evolution and master equations in open system.

If a quantum channel acts nontrivially on only k qubits, it implies that the quasi-stochastic matrix $O_{\boldsymbol{a}''\boldsymbol{a}'}$ acts also only on k qubits. The relation between the quasi-stochastic gates and the local quantum channel

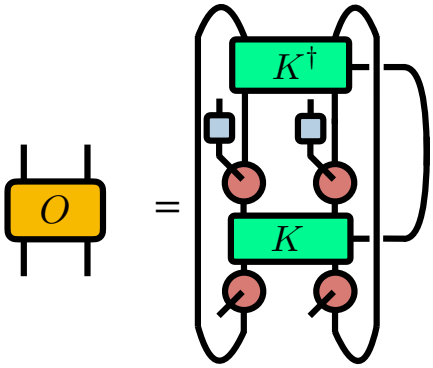


Figure 6.8: Tensor network representation of $O_{\boldsymbol{a}''\boldsymbol{a}'}$ corresponding to a quantum channel that acts on two qubits. The green tensors represent the Kraus operators, which specify the quantum channel and are understood as a rank-3 array.

in Eq.6.16 is graphically depicted in Fig.6.8 using tensor diagrams.

6.7.2 Liouville-Von Neumann Equation in POVM Formulation

In Eq.6.3 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

$$i\frac{\partial P\left(\boldsymbol{a}'',t\right)}{\partial t}=\sum_{\boldsymbol{a},\boldsymbol{a}'}\mathrm{Tr}\left(\left[\mathcal{H},M^{(\boldsymbol{a})}\right]M^{(\boldsymbol{a}'')}\right)T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}P\left(\boldsymbol{a}',t\right). \tag{6.17}$$

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

A solution to Eqs.(6.17) for a time-independent Hamiltonian is given by $\boldsymbol{P}(t)=e^{-iAt}\boldsymbol{P}(0)$, where the matrix elements $A_{\boldsymbol{a}''\boldsymbol{a}'}=\sum_{\boldsymbol{a}}T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\left[\mathrm{Tr}\left(\left[\mathcal{H},M^{(\boldsymbol{a})}\right]M^{(\boldsymbol{a}'')}\right)\right]$.

6.7.3 Lindblad Equation in POVM Formulation

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

$$i\frac{\partial P\left(\boldsymbol{a},t\right)}{\partial t}=\sum_{\boldsymbol{a}}A_{\boldsymbol{a}'',\boldsymbol{a}'}P\left(\boldsymbol{a},t\right), \tag{6.18}$$

where the matrix elements are augmented to

$$
\begin{aligned}
A_{\boldsymbol{a}''\boldsymbol{a}'} = & \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} \left(\mathrm{Tr} \left(\left[\mathcal{H}, M^{(\boldsymbol{a})} \right] M^{(\boldsymbol{a}'')} \right) \right. \\
& + \sum_{k} \left[-\frac{i}{2} \mathrm{Tr} \left(\{ L_k^\dagger L_k, M^{(\boldsymbol{a})} \} M^{(\boldsymbol{a}'')} \right) \right. \\
& \left. \left. + \, i \, \mathrm{Tr} \left(L_k M^{(\boldsymbol{a})} L_k^\dagger M^{(\boldsymbol{a}'')} \right) \right] \right)
\end{aligned}
\tag{6.19}
$$

Here, the operators L_k are called Lindblad operators or quantum jump operators. Like the Liouville-Von Neumann equation, Eq.(6.19) has a solution for a time-independet Hamiltonians given by $\boldsymbol{P}(t) = e^{-iAt}\boldsymbol{P}(0)$.

6.7.4 Measurements

Although the probabilistic representation of the quantum state in Eq.6.2 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:

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

where $q(\boldsymbol{b}|\boldsymbol{a}') = \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} \, \mathrm{Tr} \left[M^{(\boldsymbol{a})} \Pi^{(\boldsymbol{b})} \right]$ can be characterized as a quasi-conditional 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 $P(\boldsymbol{b})$ and $P(\boldsymbol{a})$ is often called the quantum law of total probability in quantum Bayesianism [235].

6.7.5 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 semi-definite 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 [224]. 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 [236], this subset is referred to as the Qplex. [224] 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.6.2 as a Qplex. The space of all possible states of a given quantum system and the corresponding Qplex are schematically represented in Fig.6.9(a)-(b).

A small quantum computation in also depicted schematically in Fig.6.9(a) and (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.6.9(b).

While these observations do not have any major conceptual implication for the physical realization of quantum 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 an standard simplex Δ_{m^N} and is not constrained to the subset of valid "quantum" distributions. Even though the update rule in Eq.6.3 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 Fig.6.4(d) and (h) where we observe values of quantum fidelity higher than 1, which means that during the training process, the ransformer induces matrices in Eq.6.2 that are not valid quantum states. Note that the fidelity in Fig.6.4(d) and (h) eventually converges and stays at values very close to one and that the oscillations above 1 disappear, suggesting that the state is getting closer and closer to the target, valid quantum state.

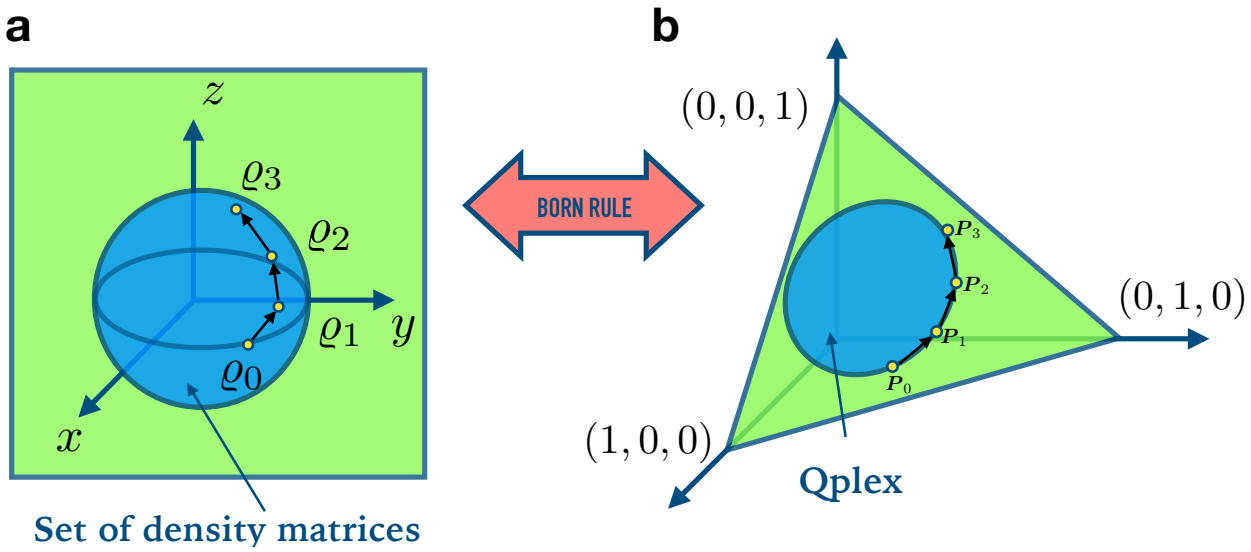


Figure 6.9: Geometry of quantum states. (a) Schematic representation of the subset of density matrices (blue sphere). For one qubit, this set corresponds to the Bloch sphere. (b) Schematic representation of the probability simplex Δ_{m^N}, which represents the set of all possible categorical probability 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).

6.7.6 Variational Circuit for TFIM

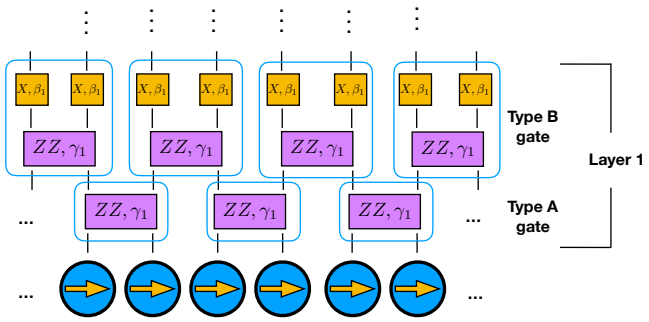


Figure 6.10: Variational circuit preparation for TFIM. Only the first of the 4 layers in our calculation is shown. The gates encircled in rounded blue squares are combined and subsequently transformed into quasi-stochastic gates for the probabilistic simulation of the quantum circuit.

We use the variational circuit depicted in Fig.6.10 for the 6-qubit TFIM preparation [225]. The parameters for gamma and beta are taken alternatively from the following sequence describing a circuit with 4 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. [225]. Note that in our simulations, we do not directly transform the original gates into quasi-stochastic gates. To save computational

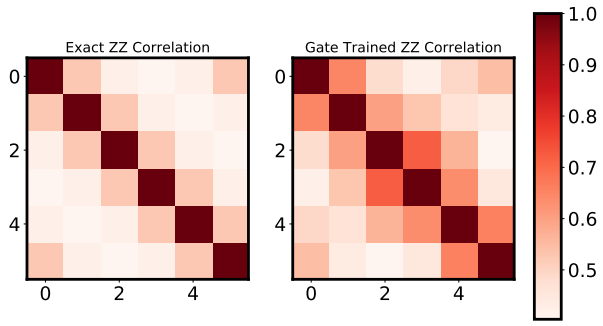


Figure 6.11: Comparison between $\sigma_i^z \sigma_j^z$ correlation from exact quantum circuit state and the gate training state.

resources, we combine the quantum gates encircled in rounded blue squares, after which we transform them into quasi-stochastic matrices.

For Transformers used in the VQE circuits simulation, the encoder block does not include LayerNorm. For $L = 6, 8, 10, 12$, $d_{model} = 16$ and $n_{ed} = 1$. For $L = 14, 16, 18$, $d_{model} = 32$ 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. 6.10 and the POVM trained circuit, which are compared in Fig. 6.11. Even though there is a nice polynomial bound between quantum fidelity and classical fidelity under the SIC-POVM formulation [197], 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. 6.12.

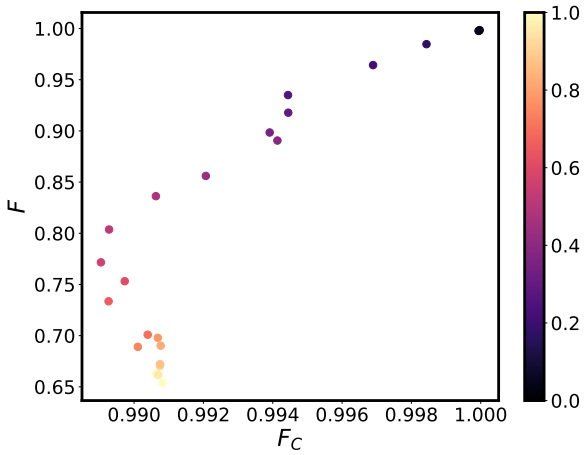


Figure 6.12: Correlation between classical fidelity and quantum fidelity. The darker color corresponds to gate that is applied earlier.

6.7.7 Optimization Details

The models are optimized using Adam Optimizer [211] in Pytorch [237] with an initial learning rate of 0.01. The weights and biases are initialized using PyTorch's [237] 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 with can be completed within a few hours for small system size L and up to one day for $L = 18$ with one V100 GPU. GHZ circuits and Graph state circuits simulation up to 60 qubits can be completed between one or two days with 4 V100 GPUs in parallel.

Chapter 7

Autoregressive Neural Network for Simulating Open Quantum Systems via a Probabilistic Formulation

This chapter is written based on the research in Ref. [238]. The theory of open quantum systems lays the foundations for a substantial part of modern research in quantum science and engineering. Rooted in the dimensionality of their extended Hilbert spaces, the high computational complexity of simulating open quantum systems calls for the development of strategies to approximate their dynamics. In this paper, we present an approach for tackling open quantum system dynamics. Using an exact probabilistic formulation of quantum physics based on positive operator-valued measure (POVM), we compactly represent quantum states with autoregressive neural networks; such networks bring significant algorithmic flexibility due to efficient exact sampling and tractable density. We further introduce the concept of String States to partially restore the symmetry of the autoregressive neural network and improve the description of local correlations. Efficient algorithms have been developed to simulate the dynamics of the Liouvillian superoperator using a forward-backward trapezoid method and find the steady state via a variational formulation. Our approach is benchmarked on prototypical one and two-dimensional systems, finding results which closely track the exact solution and achieve higher accuracy than alternative approaches based on using Markov chain Monte Carlo to sample restricted Boltzmann machines. Our work provides general methods for understanding quantum dynamics in various contexts, as well as techniques for solving high-dimensional probabilistic differential equations in classical setups.

7.1 Introduction

While the universe itself is a closed quantum system, all other systems within the universe are open quantum systems coupled to the environment around them. Open quantum systems (OQS) play a crucial role in fundamental quantum science, quantum control and quantum engineering [239, 240]. In recent years, there has been a significant interest both theoretically and experimentally in better understanding open quantum systems [241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 248, 275, 276, 250, 252]. In the

field of quantum engineering, coupling to the environment generates decoherence driving the destruction of entanglement within quantum devices. Quantum computers rely on the qubit-environment coupling to apply quantum gates as well as try to minimize unwanted coupling to mitigate errors on the qubits [277].

Unlike closed quantum states which can be represented by a wavefunction, the density matrix ρ becomes the core object of study in open quantum systems. A typical model of an OQS evolves the density matrix under both the Hamiltonian H as well as a series of dissipative operators which transfer energy and information out to a featureless bath leading to the Lindblad equation,

$$\dot{\rho} = \mathcal{L}\rho \equiv -i[H, \rho] + \sum_k \frac{\gamma_k}{2} \left(2\Gamma_k \rho \Gamma_k^\dagger - \{\rho, \Gamma_k^\dagger \Gamma_k\} \right), \tag{7.1}$$

where γ_k are the dissipation rates associated with jump operators Γ_k. Although there is hope that quantum algorithms [278, 279, 280, 281, 282] may eventually overcome the simulation bottlenecks in OQS, a direct solution to the Lindblad equation is difficult because the Hilbert space grows exponentially with the number of particles, making classical simulations largely intractable. To deal with this curse of dimensionality, OQS have historically been studied with renormalization group approaches [249, 250, 251], mean field methods [247, 248, 283]; or simulated with tensor networks [243, 244, 245, 246, 256] which compress the density matrix. Unfortunately, while tensor networks have proved fruitful in one dimension, their use for OQS in higher dimensions has been severely limited. Recently, inspired by the advances in the description of many-body systems in terms of neural network wavefunctions [284, 180, 93, 7, 6, 95, 97, 98, 99, 100], ideas from machine learning have been applied to OQS studying real-time dynamics in one dimension (1-D), steady states in one and two dimensions (2-D) [285, 133, 134, 135] and determining the Liouvillian gap [286] by representing the density matrix as a restricted Boltzmann machine (RBM) [287].

Here, we outline an alternative approach to using machine learning ideas to simulate the Lindblad equation. Many machine learning architectures and generative models (such as the RBM) have fundamentally been designed to represent probability distributions (e.g. probability distributions over images on the internet) making them inadequate to store quantum states, which are complex valued in general. To overcome this, novel approaches have been devised such as using complex weights within RBM; despite these innovative ideas, effectively representing states with signs have been a key bottleneck in this field [288, 289, 290].

This motivation has inspired us to utilize the recent developments in the probabilistic formulation of quantum mechanics [291, 292, 123, 293] to simulate the Lindblad equation. In this formulation the state is mapped to a probability distribution which we represent compactly using the Transformer [145]—a machine learning architecture from which one can efficiently sample the probability distribution exactly. Using this,

we develop efficient algorithms to both update the state of the Transformer under dynamic evolution as well as find the Transformer which represents the steady state of the Lindblad equation. To perform the dynamic evolution, we combine the second-order forward-backward trapezoid method [159] with stochastic optimization on the Transformer. Since the Transformer does not naively preserve the symmetry of the true dynamic (or fixed-point) state, we further improve upon our results by developing an additional ansatz—string states—which explicitly restores some of these symmetries. We proceed to benchmark this work on a series of one- and two-dimensional systems.

7.2 Lindblad Equation as a Probability Equation

The general objective of this paper is to develop an approach to solve for the dynamics and fixed point of the density matrix ρ in the Lindblad equation (Eq. 7.1). We test this approach on two models—the transverse-field Ising model (TFIM), where $H = \frac{V}{4}\sum_{\langle i,j\rangle}\sigma_i^{(z)}\sigma_j^{(z)} + \frac{g}{2}\sum_k \sigma_k^{(x)}$, and the Heisenberg model, where $H = \sum_{\langle i,j\rangle}\sum_{w=x,y,z} J_w\sigma_i^{(w)}\sigma_j^{(w)} + B\sum_k \sigma_k^{(z)}$. In both cases, $\Gamma_k = \sigma_k^{(-)} = \frac{1}{2}(\sigma_k^{(x)} - i\sigma_k^{(y)})$. We are interested in the expectation values of local observables given by the Pauli matrices averaged over all qubits, i.e. for a system with n qubits, we consider $\langle\sigma_w\rangle = \frac{1}{n}\sum_i\langle\sigma_i^{(w)}\rangle$ for $w = x, y, z$. Typically, the density matrix ρ is represented (explicitly or implicitly) in an orthogonal basis. In this work, we instead represent ρ in the POVM formalism. Given an informationally complete POVM (IC-POVM), a density matrix ρ of a spin-1/2 system can be uniquely mapped to a probability distribution $p(\boldsymbol{a})$, where $\boldsymbol{a}$ spans over all 4^n measurement outcomes in the POVM basis. An IC-POVM is defined by a collection of positive semi-definite operators $\{M_{(\boldsymbol{a})}\}$ called the frame, which specifies the probability distribution $p(\boldsymbol{a}) = \mathrm{Tr}\big(\rho M_{(\boldsymbol{a})}\big)$. The inverse transformation is given by $\rho = \sum_{\boldsymbol{b}} p(\boldsymbol{b})N^{(\boldsymbol{b})}$, where the dual-frame $\{N^{(\boldsymbol{b})}\}$ can be computed from the frame as $N^{(\boldsymbol{b})} = \sum_a M_{(\boldsymbol{a})}T^{-1}_{\boldsymbol{ab}}$. The elements of the overlap matrix T are given by $T_{\boldsymbol{ab}} = \mathrm{Tr}\big(M_{(\boldsymbol{a})}M_{(\boldsymbol{b})}\big)$, and $T^{-1}_{\boldsymbol{ab}}$ represent the elements of the inverse overlap matrix T^{-1}. Thus, we can re-express the Lindblad equation as

$$\dot{p}(\boldsymbol{a}) = \sum_{\boldsymbol{b}} p(\boldsymbol{b})L_{\boldsymbol{a}}^{\boldsymbol{b}} = \sum_{\boldsymbol{b}} p(\boldsymbol{b})\left(A_{\boldsymbol{a}}^{\boldsymbol{b}} + B_{\boldsymbol{a}}^{\boldsymbol{b}}\right), \tag{7.2}$$

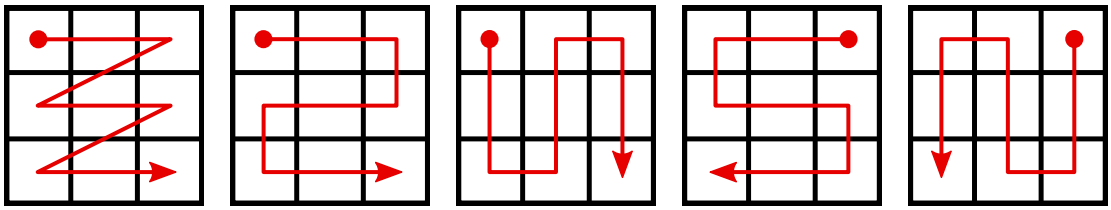
with

$$\begin{aligned} A_{\boldsymbol{a}}^{\boldsymbol{b}} &= -i\,\mathrm{Tr}\left(H[N^{(\boldsymbol{b})}, M_{(\boldsymbol{a})}]\right); \\ B_{\boldsymbol{a}}^{\boldsymbol{b}} &= \sum_k \frac{\gamma_k}{2}\,\mathrm{Tr}\left(2\Gamma_k N^{(\boldsymbol{b})}\Gamma_k^\dagger M_{(\boldsymbol{a})} - \Gamma_k^\dagger\Gamma_k\{N^{(\boldsymbol{b})}, M_{(\boldsymbol{a})}\}\right). \end{aligned} \tag{7.3}$$

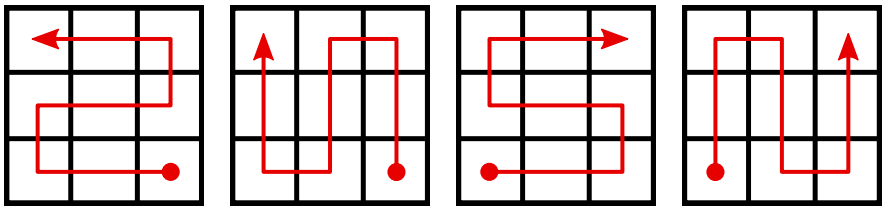
We work with an IC-POVM where the frame and dual-frame are constructed from local frames acting on single spins as $\{M_{(\boldsymbol{a})}\} = \{M_{(a_1)} \otimes M_{(a_2)} \otimes M_{(a_3)} \otimes \cdots\}$ and $\{N^{(\boldsymbol{b})}\} = \{N^{(b_1)} \otimes N^{(b_2)} \otimes N^{(b_3)} \otimes \cdots\}$ with four outcomes per spin a_i. This allows us to write $p(\boldsymbol{a}) = p(a_1, a_2, a_3, \cdots)$. The expectation value of observables are given by $\langle O \rangle = \sum_{\boldsymbol{b}} p(\boldsymbol{b}) \operatorname{Tr}\left(ON^{\boldsymbol{b}}\right) \approx \frac{1}{N_s} \sum_{\boldsymbol{b}\sim p}^{N_s} \operatorname{Tr}\left(ON^{\boldsymbol{b}}\right)$, where N_s is the number of samples $\boldsymbol{b}$ drawn from the distribution $p(\boldsymbol{b})$ used to estimate $\langle O \rangle$. We emphasize that a complete specification of the probability distribution $p(\boldsymbol{b})$ requires 4^n probability values for an n-site system.

7.3 Autoregressive Models and String States

We have chosen to model the probability distribution in a compact way with an autoregressive neural network where the probability of a given configuration $\boldsymbol{a}$ is expressed through its conditional probabilities $p_\theta(\boldsymbol{a}) = \prod_k p_\theta(a_k | a_1, a_2, \cdots, a_{k-1})$. This representation allows for exact sampling of a configuration from the space of probability distributions without invoking Markov chain Monte Carlo techniques. Modern incarnations of autoregressive models include, among others, recurrent neural networks (RNN) [143, 142], pixel convolutional neural networks (PixelCNN) [144], Transformers [145]. Recent work has effectively applied these models to quantum systems [292, 95, 97, 123, 294]. Here, we use an autoregressive Transformer, which follows the same architecture as the model in [123]. The Transformer consists of two hyper-parameters: the number of transformer layers stacked on each other n_l and the hidden dimension n_d, which we adjusted for different tests.



(a) String 0. (b) String 1. (c) String 2. (d) String 3. (e) String 4.



(f) String 5. (g) String 6. (h) String 7. (i) String 8.

Figure 7.1: Strings used for mapping 1-D Transformer to 2-D quantum systems. String 0 is the default mapping (which we refer to as no strings). We always refer to the first (in order) n strings (excluding string 0) when we say we used n strings.

Since our Transformer gives 'ordered' measurement outcomes, when we simulate two-dimensional systems we need to choose a linear ordering of our two-dimensional sites (i.e. a string of sites). We consider two different single-string orderings (string 0 and string 1 from Fig. 7.1(a)). These strings explicitly break a symmetry of our system which then would need to be restored (to the degree to which the model has the variational freedom to do so) by the Transformer itself. We can partially (or completely) restore this symmetry explicitly by choosing our ansatz to be a mixture of distributions defined over multiple different symmetry-related strings - i.e. $p_\theta(\boldsymbol{a}) = \sum_{\mathcal{S}} p_\theta(\boldsymbol{a}|\mathcal{S})p(\mathcal{S})$, where $p(\mathcal{S}) = 1/N_{\text{string}}$ for a total number of N_{string} strings; we call this refined ansatz a String state. This linear combination of the Transformer probabilities can be interpreted as a mixture model [295] and bears some resemblance to string bond states [296]. Restoring symmetries explicitly has proved useful in variational calculations of quantum states [297, 298, 299, 95, 288]. Given a set of strings and a configuration $\boldsymbol{a}$ we can compute $p(\boldsymbol{a})$ explicitly. Sampling an $\boldsymbol{a}$ from $p_\theta(\boldsymbol{a})$ is also straightforward because of linearity and the fact that each term in our average is positive. To do so, we first sample an ordered $\{a_1, a_2, ...a_{k-1}\}$ from the Transformer and then randomly choose a string to map these ordered values to get the final configuration. Here, we test a subset of strings 1-k for different k (see Fig. 7.1(b)-7.1(i)).

7.4 Optimization and Results

Eq. 7.2 gives a prescription for applying time evolution to the density matrix by time-evolving the POVM probability distribution. To solve for the time-evolved distribution, we discretize time and use a second-order forward-backward trapezoid method [159]. We designed the following objective function

$$\mathcal{C} = \frac{1}{N_s} \sum_{\boldsymbol{a}\sim p_{\theta(t+2\tau)}}^{N_s} \frac{1}{p_{\theta(t+2\tau)}(\boldsymbol{a})} \left| \sum_{\boldsymbol{b}} \left[p_{\theta(t+2\tau)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) - p_{\theta(t)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} + \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) \right] \right|, \tag{7.4}$$

where N_s is the number of samples, $\delta_{\boldsymbol{a}}^{\boldsymbol{b}}$ is the Kronecker delta function, the sum is over $\boldsymbol{a}$ sampled stochastically from $p_{\theta(t+2\tau)}$, and the gradient of the objective function $\mathcal{C}$ with respect to the parameters in $p_{\theta(t+2\tau)}(\boldsymbol{b})$ is computed using PyTorch's [156] automatic differentiation. To optimize the objective function we use Adam [211]. In the limit where $\mathcal{C}$ is zero, we get exact time evolution up to the discretization error induced by the trapezoid rule. More typically, it will be impossible for the Transformer to exactly represent the time-evolved state; instead by minimizing $\mathcal{C}$ the optimization continuously projects onto a nearby state in the manifold of distributions represented by our Transformer. This can be viewed as a higher order generalization of IT-SWO

[160] and the method in Ref. [300] but here applied instead to a probability distribution. The dominant source of error in performing our dynamics comes from the limited set of states that the Transformer can represent. Additionally, it is possible that even within this manifold of states, one may not reach the optimal value if there are optimization issues such as local minima. Over multiple time steps, errors will naturally accumulate due to the unitary dynamics of the system and be suppressed by the dissipative operators which should drive all dynamics to a fixed point. We test this dynamic evolution on the 1-D and a 2-D Heisenberg

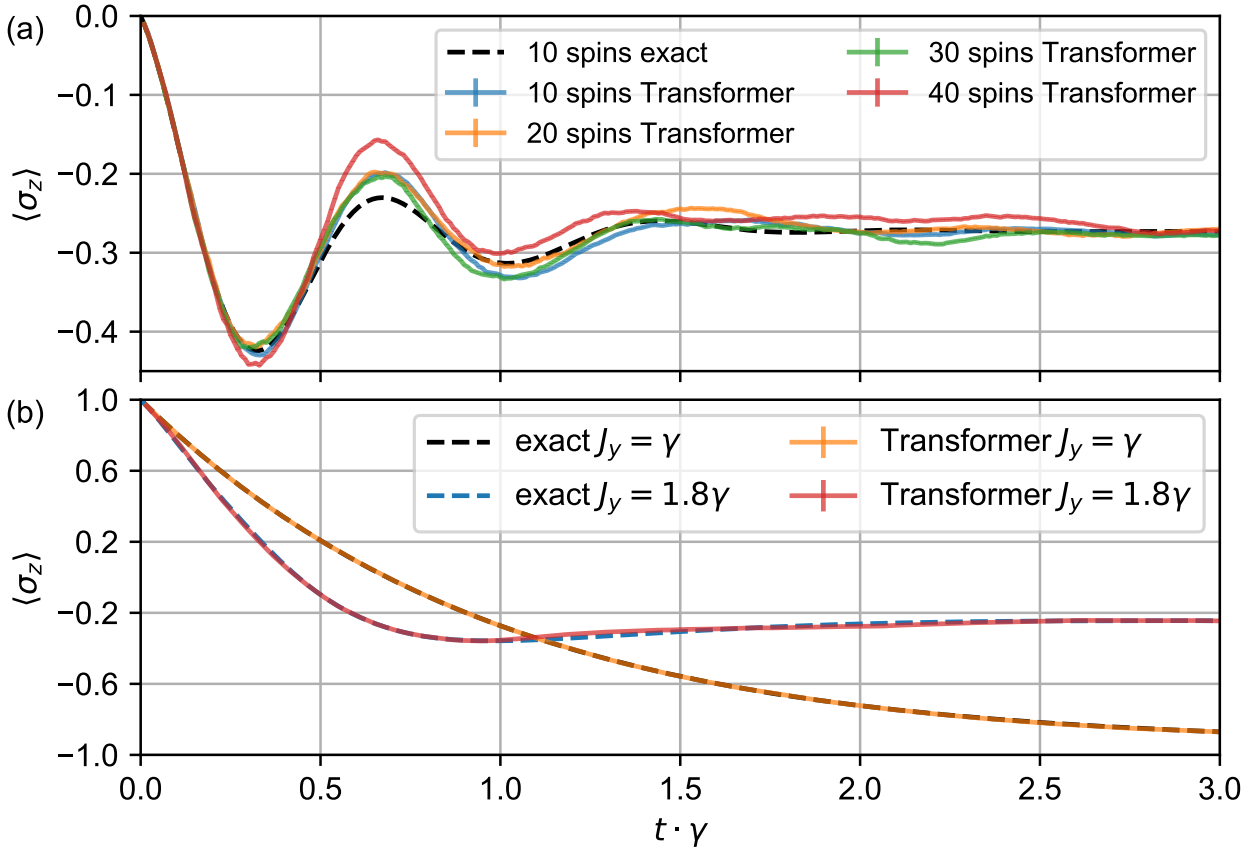


Figure 7.2: The expectation value $\langle\sigma_z\rangle$ as a function of time (a) for the 1-D Heisenberg model with $B = \gamma$, $J_x = 2\gamma$, $J_y = 0$, and $J_z = \gamma$ using a time step $\tau = 0.005\gamma^{-1}$. The initial state is the product state $\prod_{i=1}^{N} |\leftarrow\rangle$ ($\langle\sigma_y\rangle = -1$). (b) for the 3×3 Heisenberg model with $B = 0$, $J_x = 0.9\gamma$, $J_y = 1.0\gamma, 1.8\gamma$, and $J_z = \gamma$ using a time step $\tau = 0.008\gamma^{-1}$. The initial state is the product state $\prod_{i=1}^{N} |\uparrow\rangle$ ($\langle\sigma_z\rangle = 1$). Both models use periodic boundary conditions. Exact curves are produced using QuTip [301, 302]. The Transformer has one encoder layer and 32 hidden dimensions, and is trained using a forward-backward trapezoid method with a sample size $N_s = 12000$.

model (see Fig. 7.2) using the tetrahedral POVM basis (see Supplementary Material [303] Sec. II) where we find that the dynamics matches closely to the exact result. We capture both the qualitative behavior (i.e. the peaks and oscillation of the observables) as well as their quantitative values. The values are especially accurate in both the limit of small and large time. In our results, we have simulated one-dimensional chains

up to $N = 40$ and two-dimensional chains for 3×3 lattices.

One approach to finding the fixed point of the Liouvillian superoperator $\mathcal{L}$ is through a sufficiently long time-evolution (for an example see the large time limit of Fig. 7.2). Interestingly, our approximate time evolution fluctuates around a fixed value of the observable, though it may not reach a true fixed point (i.e. $p_\theta(t + 2\tau) = p_\theta(t)$) even in the limit of small τ. (see Supplementary Material [303] Sec. VII)

Alternatively, we can search for the fixed point by direct minimization of the L_1-norm of $\dot{p_\theta}$ giving

$$\|\dot{p_\theta}\|_1 = \sum_{\boldsymbol{a}} \left| \sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right| \approx \frac{1}{N_s} \sum_{\boldsymbol{a} \sim p_\theta}^{N_s} \frac{\left| \sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right|}{p_\theta(\boldsymbol{a})}, \tag{7.5}$$

where the second line offers a stochastic approach to evaluate the $\|\dot{p_\theta}\|_1$ by sampling $\boldsymbol{a}$ from $p_\theta(\boldsymbol{a})$. The gradient in Eq. 7.5 is taken with respect to the parameters in $p_\theta(\boldsymbol{b})$ using PyTorch's [156] automatic differentiation. Notice that because the gradients of Eq. 7.28 and Eq. 7.5 (see Supplementary Material [303] Sec. VII) are different (except in the limit where the manifold of states representable by the Transformer span the full space), they will converge to different answers.

In Fig. 7.3, we consider the one-dimensional TFIM, with the 4-Pauli POVM basis (see Supplementary Material [303] Sec. II), and compute the expectation value of all three Pauli matrices at various values of γ. We find strong agreement with the exact method. In addition, we find that this approach performs particularly well in the regime of $1 < g/\gamma < 2.5$ which have proven particularly challenging for the RBM method [285]. We can further improve the performance by averaging over multiple simulations (see Supplementary Material [303] Sec. V). In Fig. 7.4, we consider optimizing a 3×3 Heisenberg model using Eq. 7.5 with various different variational ansatz (here we use the tetrahedral POVM basis (see Supplementary Material [303] Sec. II)). In looking at the quality of $\langle \sigma_z \rangle$ we find that increasing the size of the Transformer both in depth and hidden dimension improves the result although this improvement is marginal until we reach two transformer layers and a hidden dimension of 64. Interestingly, we find that the use of strings has a significant effect on our results (see Fig. 7.1(i)). To begin with, the use of string 1 is marginally superior to string 0. We expect this is because string 1 better addresses local correlations. More importantly, we find that there is a significant improvement (for any Transformer) by including more symmetry related strings out to the maximum of eight strings we considered. In fact, eight strings with one hidden layer and a hidden dimension of 32 provides a similar accuracy to 1 string with 2 hidden layers and a hidden dimension of 64. Additionally, we compared the results obtained through time evolution at long time to the fixed point method and found that the steady-state approached by the time-evolved state provides significantly more accurate results. While the evaluation of the dynamics is computationally slower, we find that supplement-

ing the fixed-point method with further dynamical evolution achieves the same steady-state solution as the dynamical approach at an overall reduced computational time.

7.5 Conclusion

We have demonstrated an efficient and scalable approach to simulate the real-time dynamics of open quantum systems via an exact probabilistic formulation. By parameterizing the quantum state using an autoregressive Transformer, we accurately track the dynamics and steady state in 1-D and 2-D transverse field Ising and Heisenberg models. For 2-D systems, we introduce String States which partially restore the symmetry of the Transformer.

Our methods constitute an important step in the machine learning approach for quantum many-body dynamics simulation. It provides the first exact sampling method for neural networks in OQS, which is a crucial improvement over the standard Markov chain Monte Carlo techniques with RBM, as well as an efficient stochastic optimization method for high dimensional differential equations. Our approach is versatile and applicable to general quantum dynamics in various contexts, including closed systems quantum dynamics, finite temperature dynamics of the density matrix, as well as challenging fermionic transport problems [304, 305] with interactions to the environment [306]. Due to the probabilistic formulation as a quantum-classical mapping, our work has applications beyond quantum mechanics and demonstrates how to efficiently solve high-dimensional probabilistic differential equations with autoregressive neural networks. Such probabilistic equations appear in a wide variety of classical contexts and our work represents an important step forward in the direction.

7.6 Supplementary Materials

7.6.1 Lindblad Equation in POVM Formalism

Start from the Lindblad equation for density matrices

$$\dot{\rho} = \mathcal{L}\rho = -i[H, \rho] + \sum_k \frac{\gamma_k}{2} \left(2\Gamma_k \rho \Gamma_k^\dagger - \{\rho, \Gamma_k^\dagger \Gamma_k\} \right), \tag{7.6}$$

the frame and dual-frame satisfies

$$p(\boldsymbol{a}) = \mathrm{Tr}\big(\rho M_{(\boldsymbol{a})}\big), \tag{7.7}$$

and

$$\rho = \sum_{\boldsymbol{b}} p(\boldsymbol{b}) N^{(\boldsymbol{b})}. \tag{7.8}$$

Plugging Eq. 7.8 into Eq. 7.6, we have

$$\sum_{\boldsymbol{b}} \dot{p}(\boldsymbol{b}) N^{(\boldsymbol{b})} = \sum_{\boldsymbol{b}} p(\boldsymbol{b}) \Big[-i[H, N^{(\boldsymbol{b})}] + \sum_k \frac{\gamma_k}{2} \Big(2\Gamma_k N^{(\boldsymbol{b})} \Gamma_k^\dagger - \{N^{(\boldsymbol{b})}, \Gamma_k^\dagger \Gamma_k\} \Big) \Big]. \tag{7.9}$$

Plugging Eq. 7.8 into Eq. 7.7, we have

$$p(\boldsymbol{a}) = \sum_{\boldsymbol{b}} p(\boldsymbol{b}) \operatorname{Tr}\Big(N^{(\boldsymbol{b})} M_{(\boldsymbol{a})} \Big). \tag{7.10}$$

Therefore, let's add $M_{(\boldsymbol{a})}$ and take trace on both sides of Eq. 7.9,

$$\sum_{\boldsymbol{b}} \dot{p}(\boldsymbol{b}) \operatorname{Tr}\Big(N^{(\boldsymbol{b})} M_{(\boldsymbol{a})} \Big) = \sum_{\boldsymbol{b}} p(\boldsymbol{b}) \left[-i \operatorname{Tr}\Big([H, N^{(\boldsymbol{b})}] M_{(\boldsymbol{a})} \Big) + \sum_k \frac{\gamma_k}{2} \operatorname{Tr}\Big(2\Gamma_k N^{(\boldsymbol{b})} \Gamma_k^\dagger M_{(\boldsymbol{a})} - \{N^{(\boldsymbol{b})}, \Gamma_k^\dagger \Gamma_k\} M_{(\boldsymbol{a})} \Big) \right]. \tag{7.11}$$

Replacing the left side with Eq. 7.10 and rearranging the right side, we arrive at

$$\dot{p}(\boldsymbol{a}) = \sum_{\boldsymbol{b}} p(\boldsymbol{b}) \left[-i \operatorname{Tr}\Big(H[N^{(\boldsymbol{b})}, M_{(\boldsymbol{a})}] \Big) + \sum_k \frac{\gamma_k}{2} \operatorname{Tr}\Big(2\Gamma_k N^{(\boldsymbol{b})} \Gamma_k^\dagger M_{(\boldsymbol{a})} - \Gamma_k^\dagger \Gamma_k \{N^{(\boldsymbol{b})}, M_{(\boldsymbol{a})}\} \Big) \right] \equiv \sum_{\boldsymbol{b}} p(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \tag{7.12}$$

7.6.2 Tetrahedral and 4-Pauli POVM

In the main paper, we used two POVMs, the tetrahedral POVM and the 4-Pauli POVM. The tetrahedral POVM forms a tetrahedral in the Bloch sphere. In particular, it takes the form of

$$M_{(a)} = \frac{1}{4} \left(\mathbb{1} + \boldsymbol{v}_{(a)} \cdot \boldsymbol{\sigma} \right), \tag{7.13}$$

where $\mathbb{1}$ is the identity matrix, $\boldsymbol{\sigma}$ are the Pauli matrices, and $\boldsymbol{v}_{(a)}$ are four unit vectors which form a tetrahedral. In the main paper, we choose the four vectors to be

$$\boldsymbol{v}_{(1)} = \langle 0, 0, 1 \rangle , \tag{7.14}$$

$$\boldsymbol{v}_{(2)} = \left\langle \frac{2\sqrt{2}}{3}, 0, -\frac{1}{3} \right\rangle , \tag{7.15}$$

$$\boldsymbol{v}_{(3)} = \left\langle -\frac{\sqrt{2}}{3}, \frac{\sqrt{6}}{3}, -\frac{1}{3} \right\rangle , \tag{7.16}$$

$$\boldsymbol{v}_{(4)} = \left\langle -\frac{\sqrt{2}}{3}, -\frac{\sqrt{6}}{3}, -\frac{1}{3} \right\rangle . \tag{7.17}$$

The 4-Pauli POVM, on the other hand, takes the form of

$$M_{(1)} = \frac{1}{3} |0\rangle \langle 0| , \tag{7.18}$$

$$M_{(2)} = \frac{1}{3} |+\rangle \langle +| , \tag{7.19}$$

$$M_{(3)} = \frac{1}{3} |r\rangle \langle r| , \tag{7.20}$$

$$M_{(4)} = \mathbb{1} - M_{(1)} - M_{(2)} - M_{(3)}, \tag{7.21}$$

where $|0\rangle$, $|+\rangle$, and $|r\rangle$ are the positive eigenstates of σ_z, σ_x, and σ_y respectively. The multi-site POVM is constructed as the tensor product of single-site POVMs.

7.6.3 Convergence of Loss

Here we show that the observable converges for dynamics process (Fig. 7.5) and the training loss converges for variational method (Fig. 7.6).

7.6.4 Results for Heisenberg Model with Larger Energy Scale

The dynamics algorithm is also tested on a different Heisenberg model where $B = 10\gamma$, $J_x = 20\gamma$, $J_y = 0$, and $J_z = 10\gamma$. This model has a higher energy compared with the model in the main paper. Since the dissipation operator is relatively small compared with the Hamiltonian, this model reveals closed system properties as well as open system properties. In Fig. 7.7, we show the short time dynamics for different number of spins of this model. It can be seen that for short time dynamics, the neural network predicts the observables to a great precision. In Fig. 7.8, we show the long time dynamics behavior for different number of spins. Even though the performances of different number of spins are different, it starts to converge to

the right steady state as the system size goes larger.

7.6.5 Improve Performance by Combining Probabilities

Because of the stochastic nature of the initialization and training process, each training could yield a slightly different result. In principle, we can average over multiple results to achieve better performance by defining the overall POVM probability

$$p(\boldsymbol{a}) = \frac{1}{N}\sum_i p_i(\boldsymbol{a}). \tag{7.22}$$

Then, the observable is computed as

$$\langle O\rangle = \frac{1}{N}\sum_{i,\boldsymbol{b}} p_i(\boldsymbol{b})\,\mathrm{Tr}\big(ON^{\boldsymbol{b}}\big) = \frac{1}{N}\sum_i \langle O\rangle_i, \tag{7.23}$$

which turns out to be the average of observables. In Fig. 7.9, we show the results for 1-D transverse Ising model for multiple training processes. It can be noted that for $\langle\sigma_x\rangle$ and $\langle\sigma_y\rangle$, the average result is indeed better.

7.6.6 Neural Network Initialization

All the neural networks used in the main paper have the weights and biases (except in the last layer) initialized using PyTorch [156] default linear layer normalization. All training process starts with a product state of either $|\uparrow\rangle$ or $|\leftarrow\rangle$ ($\langle\sigma_z\rangle = 1$ or $\langle\sigma_y\rangle = -1$ respectively, see figure captions for the exact initial state). To initialize the neural networks in such a product state, in the last fully connected layer, the weight is set to zero and the bias is set to $\log\left(\langle\psi| M_{(a)} |\psi\rangle\right)$ where $M_{(a)}$ is the single spin POVM basis. Thus, after softmax, the output of the neural network would be the corresponding product state in POVM basis.

7.6.7 Explanation of the Dynamics and Variational Cost Functions in Detail

In this section, we explain the dynamics and variational cost functions (Eq. 4 and Eq. 5) in the main paper in detail. We start with the dynamics cost function. We would like to produce the probability distribution at $t + 2\tau$ from the probability distribution at t using forward-backward trapezoid method [159] as

$$p_{\theta(t+2\tau)} - \tau L p_{\theta(t+2\tau)} = p_{\theta(t)} + \tau L p_{\theta(t)}. \tag{7.24}$$

To make the notation compatible with the main paper, we could write Eq. 7.24 as

$$\sum_{\boldsymbol{b}} p_{\theta(t+2\tau)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) = \sum_{\boldsymbol{b}} p_{\theta(t)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} + \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right), \tag{7.25}$$

where $\delta_{\boldsymbol{a}}^{\boldsymbol{b}}$ is the Kronecker delta function. Then, we could design the cost function as the L_1-distance between the left hand side and right hand side as

$$\mathcal{C} = \sum_{\boldsymbol{a}} \left| \sum_{\boldsymbol{b}} \left[p_{\theta(t+2\tau)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) - p_{\theta(t)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} + \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) \right] \right|. \tag{7.26}$$

Minimizing this cost function with respect to $p_{\theta(t+2\tau)}$ would be equivalent to solve the original equation if the cost function can be minimized to zero. However, since the neural network can only approximate the probability distribution, we are approximately solving the original equation. In addition, the probability distribution has a dimension that exponentially increases as the number of spins increase, so it is not feasible to minimize Eq. 7.26 exactly. Therefore, we seek for a stochastic version of the cost function. This can be achieved by applying the trick of multiplying the cost function by $1 = p_{\theta(t+2\tau)}(\boldsymbol{a})/p_{\theta(t+2\tau)}(\boldsymbol{a})$ as

$$\mathcal{C} = \sum_{\boldsymbol{a}} p_{\theta(t+2\tau)}(\boldsymbol{a}) \frac{1}{p_{\theta(t+2\tau)}(\boldsymbol{a})} \left| \sum_{\boldsymbol{b}} \left[p_{\theta(t+2\tau)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) - p_{\theta(t)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} + \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) \right] \right|. \tag{7.27}$$

Notice that we should only take gradient on $p_{\theta(t+2\tau)}(\boldsymbol{b})$ but not on $p_{\theta(t+2\tau)}(\boldsymbol{a})$. Then, we could turn the first $p_{\theta(t+2\tau)}(\boldsymbol{a})$ into sampling $\boldsymbol{a} \sim p_{\theta(t+2\tau)}(\boldsymbol{a})$ and the resulting equation is

$$\mathcal{C} = \frac{1}{N_s} \sum_{\boldsymbol{a} \sim p_{\theta(t+2\tau)}}^{N_s} \frac{1}{p_{\theta(t+2\tau)}(\boldsymbol{a})} \left| \sum_{\boldsymbol{b}} \left[p_{\theta(t+2\tau)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) - p_{\theta(t)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} + \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) \right] \right|, \tag{7.28}$$

which is exactly Eq. 4 in the main paper. Sampling $\boldsymbol{a} \sim p_{\theta(t+2\tau)}(\boldsymbol{a})$ is efficient, as the autoregressive neural network is designed to sample from the probability distribution efficiently and exactly. (See Sec. VIII for sampling details.) We should additionally notice that we don't need to sample over $\boldsymbol{b}$. Since the Hamiltonian and jumping operators are local, $L_{\boldsymbol{a}}^{\boldsymbol{b}}$ is sparse. For a given $\boldsymbol{a}$, only a small number of (16 to be exact) $\boldsymbol{b}$'s are involved in the computation. Therefore, we could evaluate the sum over $\boldsymbol{b}$ exactly. Notice that autoregressive neural network allows exact inference so that the probability of each configuration can be exactly evaluated. The final cost function C is then the expectation of the summand over a sampled from $p_\theta(t + 2\tau)$.

We can deal with the variational cost function similarly. Since we are searching for the steady state, we would like to solve for p_θ such that

$$0 = \dot{p_\theta} = L p_\theta, \tag{7.29}$$

or, in the notation of the main paper,

$$0 = \dot{p_\theta}(\boldsymbol{a}) = \sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}}. \tag{7.30}$$

We can just define the cost function as

$$\|\dot{p_\theta}\|_1 = \sum_{\boldsymbol{a}} \left| \sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right|. \tag{7.31}$$

Similar to the dynamics cost function, this can be turned into a stochastic cost function using the same method, and the result is

$$\|\dot{p_\theta}\|_1 = \frac{1}{N_s} \sum_{\boldsymbol{a}\sim p_\theta}^{N_s} \frac{\left|\sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}}\right|}{p_\theta(\boldsymbol{a})}, \tag{7.32}$$

the same as Eq. 5 in the main paper. Similar to the dynamics cost function, the gradient should be taken on $p_\theta(\boldsymbol{b})$ only.

Notice that these two cost functions serve two different purposes. The dynamics cost function is designed to train a neural network for each time step in an evolution, while the variational cost function seeks the steady state directly. In other words, the dynamics cost function produces many neural network states, where each of them represents the quantum states at a particular time, but the variational cost function only produces one neural network states, which is the final steady state. Since the evolution is dissipative, the neural network states generated from time evolution for large time should match the neural network state generated from the variational cost function. In practice, however, we noticed that time evolution generally produces better steady states. We believe the reason lies in the fact that the gradient of the two cost functions are different. (Please refer to the Sec. IX for details.)

7.6.8 Exact Sampling from Conditional Probability Distributions

In this section, we explain how the probability distribution is sampled exactly. In the main paper, we explained that the Transformer neural network parameterizes the probability distribution over all spins as a product of conditional probabilities on each spin as

$$p(\boldsymbol{a}) = p(a_1, a_2, a_3, \cdots) = \prod_k p_\theta(a_k | a_1, a_2, \cdots, a_{k-1}). \tag{7.33}$$

The sampling procedure is as follows:

1. Sample $a_1 \sim p_\theta(a_1)$;

2. Sample $a_2 \sim p_\theta(a_2|a_1)$;

3. Sample $a_3 \sim p_\theta(a_3|a_1, a_2)$;

$\vdots$

Due to the autoregressive structure of the neural network, each sample can be drawn efficiently [145]. This procedure allows for sampling without Markov chain Monte Carlo (MCMC), so it does not need to "warm up" before generating usable samples. In addition it allows for an arbitrary number of samples to be sampled parallelly and independently, avoiding the correlation between samples in MCMC.

7.6.9 Analysis of the Gradients of Dynamics and Variational Cost Functions

The gradient of the dynamics cost function

$$\mathcal{C} = \frac{1}{N} \sum_{\boldsymbol{a} \sim p_{\theta(t+2\tau)}}^{N} \frac{1}{p_{\theta(t+2\tau)}(\boldsymbol{a})} \left| \sum_{\boldsymbol{b}} \left[p_{\theta(t+2\tau)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}} \right) - p_{\theta(t)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} + \tau L_{\boldsymbol{a}}^{\boldsymbol{b}} \right) \right] \right|. \tag{7.34}$$

is

$$\frac{\partial \mathcal{C}}{\partial \theta} = \sum_{\boldsymbol{a}} \left[\sum_{\boldsymbol{b}} \frac{\partial p_{\theta(t+2\tau)}(\boldsymbol{b})}{\partial \theta} \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}} \right) \right] \text{sign} \left\{ \sum_{\boldsymbol{b}} \left[p_{\theta(t+2\tau)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}} \right) - p_{\theta(t)}(\boldsymbol{b}) \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} + \tau L_{\boldsymbol{a}}^{\boldsymbol{b}} \right) \right] \right\}. \tag{7.35}$$

We find in our simulations that there is no numerical fixed point (i.e. $p_{\theta(t)} = p_{\theta(t+2\tau)}$) for our Transformer dynamics. This is plausible as $\sum_{\boldsymbol{b}} p_{\theta(t+2\tau)}(\boldsymbol{b}) L_a^b$ is not generically going to be zero except in the case when the Transformer reaches the exact steady state solution (which generically a moderate size transformer won't be able to represent). This means that neither Eq. 7.34 nor Eq. 7.35 is going to be zero.

The gradient of the variational cost function

$$\|\dot{p_\theta}\|_1 = \sum_{\boldsymbol{a}} \left| \sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right| = \frac{1}{N} \sum_{\boldsymbol{a} \sim p_\theta}^{N} \frac{\left| \sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right|}{p_\theta(\boldsymbol{a})}, \tag{7.36}$$

is

$$\frac{\partial \|\dot{p_\theta}\|_1}{\partial \theta} = \sum_{\boldsymbol{a}} \left[\sum_{\boldsymbol{b}} \frac{\partial p_\theta(\boldsymbol{b})}{\partial \theta} L_{\boldsymbol{a}}^{\boldsymbol{b}} \right] \text{sign} \left[\sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right]. \tag{7.37}$$

It is worth contrasting how these two approaches produce different results. It can be easily observed that the gradient is very different for these two approaches. In addition, the dynamics algorithm only locally

matches the Transformer at two different time steps, while the variational algorithm globally searches for the steady state.

Since empirically the variational method is fast but not so accurate while the dynamics method is accurate but not fast, we might combine the gradient to take the advantages of both approaches. One can consider the interpolated dynamics cost function as follows

$$\mathcal{C}_1 = \lambda \mathcal{C} + (1-\lambda)\|\dot{p_\theta}\|_1. \tag{7.38}$$

During the dynamics process, one could slowly increase λ from 0 to 1, switching from variational algorithm to dynamics algorithm. This cost function would produce inaccurate intermediate dynamics process, but should produce accurate steady state result. In the main paper, we have performed dynamics after the variational results, attaining accurate observables while reducing the training cost, which could be viewed as a special case of the above.

Even though the numerical fixed point may not be a local minimum of Eq. 7.34 as discussed previously, it might still be useful to consider how gradient of the dynamics loss at the numerical fixed point looks like

$$\begin{aligned}\frac{\partial \mathcal{C}}{\partial \theta} &= \sum_{\boldsymbol{a}} \left[\sum_{\boldsymbol{b}} \frac{\partial p_\theta(\boldsymbol{b})}{\partial \theta} \left(\delta_{\boldsymbol{a}}^{\boldsymbol{b}} - \tau L_{\boldsymbol{a}}^{\boldsymbol{b}}\right) \right] \text{sign} \left[-\sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right] \\ &= \tau \frac{\partial \|\dot{p_\theta}\|_1}{\partial \theta} - \sum_{\boldsymbol{a}} \frac{\partial p_\theta(\boldsymbol{a})}{\partial \theta} \text{sign} \left[\sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right].\end{aligned} \tag{7.39}$$

The first term is the same as the variational gradient, up to a scaling factor. Although a direct optimization using this gradient would not work since it only works at the numerical fixed point, one could be inspired by this gradient and formulate a new variational cost function as

$$\mathcal{C}_2 = \lambda \|\dot{p_\theta}\|_1 - (1-\lambda) \sum_{\boldsymbol{a}} p_\theta(\boldsymbol{a}) \text{sign} \left[\sum_{\boldsymbol{b}} p_\theta(\boldsymbol{b}) L_{\boldsymbol{a}}^{\boldsymbol{b}} \right]. \tag{7.40}$$

Then, one could choose different λ to adjust the effect of the second term. In practice, we observed some improvements using this cost function, but the performances are unstable. It may be related to property that the sign function is sensitive to small changes.

7.6.10 Additional Benchmarks with Classical and Quantum Algorithms

In the main paper, we compared the results from Ref. [285] and showed that we achieved better results. Here, we additionally benchmark with Fig. 3 in Ref. [135] (results shown in Fig. 7.10), Fig. 9 in Ref. [278]

(results shown in Fig. 7.11), and Fig. 3 in Ref [282] (results shown in Fig. 7.12). Specifically, Ref. [135] is another stochastic machine learning algorithm with Restricted Boltzmann machine (RBM) in the standard density matrix formulation, while Ref. [278] and Ref. [282] are variational quantum algorithms proposed recently. It is shown in the figures that our results are better in every comparison. We have also included the new benchmark results in the Supplmentary materials.

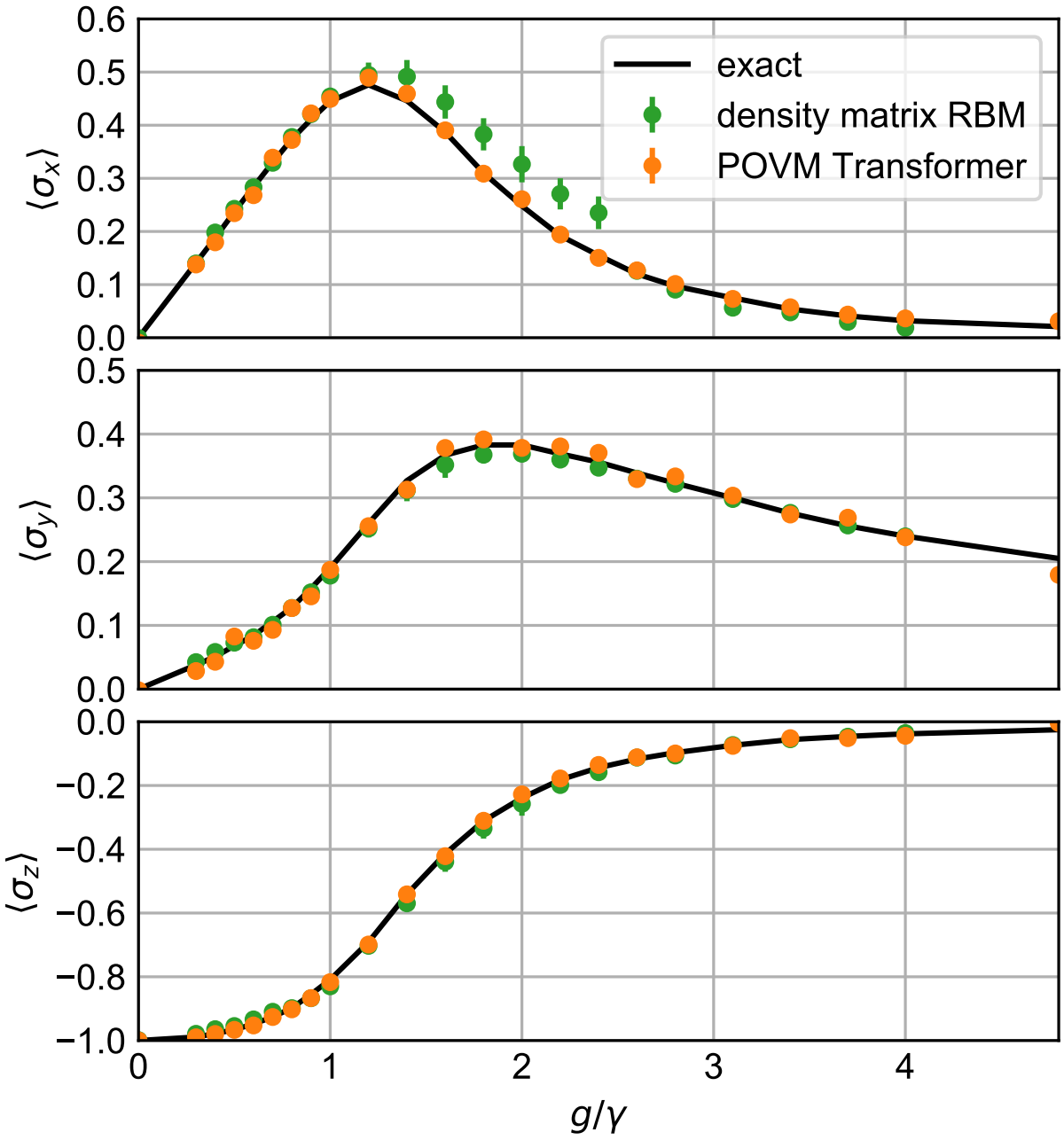


Figure 7.3: Variational steady-state solution for a 16-site TFIM chain with periodic boundary condition and $V = 2\gamma$ (orange dots). The initial state is the product state $\prod_{i=1}^{N} |\uparrow\rangle$ ($\langle\sigma_z\rangle = 1$). The Transformer has one encoder layer and 32 hidden dimensions, and is trained using Adam [211] in 500 iterations with $N_s = 12000$. Green points are the fixed point solution representing the density matrix as an RBM; both the exact curve (black line) and density matrix results are digitized from Ref. [285].

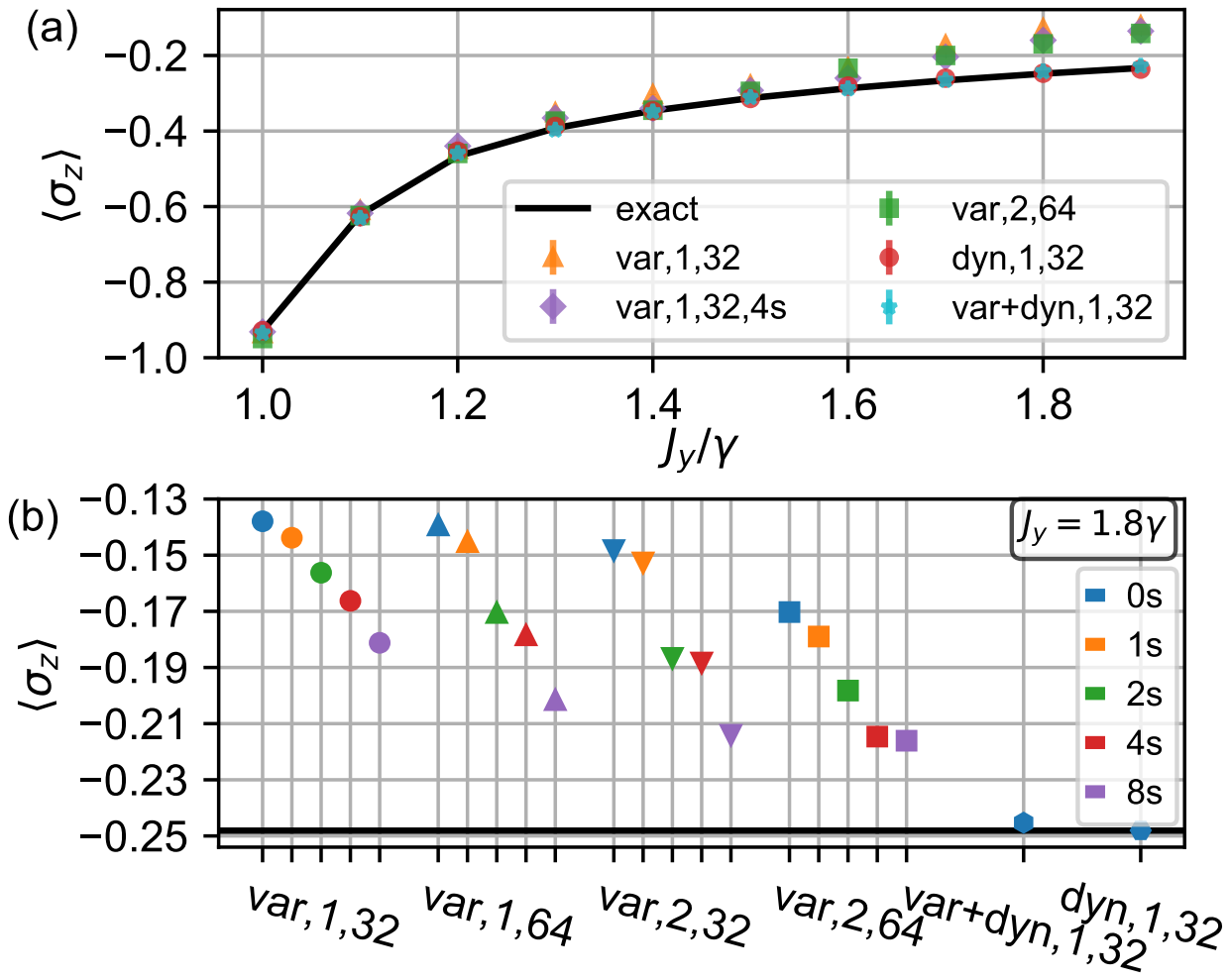


Figure 7.4: Steady-state solutions for 3×3 Heisenberg model with periodic boundary condition with $B = 0$, $J_x = 0.9\gamma$, and $J_z = \gamma$. The exact curves (black lines) are produced using QuTiP [301, 302]. (a) $\langle \sigma_z \rangle$ for different values of J_y for POVM variational results (var), POVM dynamics (dyn) and POVM dynamics starting from the variational results (var+dyn). The two integers in the legend label are the number of transformer layers and hidden dimensions. (b) Steady-state solution $J_y = 1.8\gamma$ comparing different variational ansatz. "0s" and "1s" use one string (String 0 and String 1); "2s", "4s", and "8s" use Strings 1-2, 1-4, and 1-8 respectively (see Fig. 7.1(i)) All initial states are $\prod_{i=1}^{N} |\uparrow\rangle$ ($\langle \sigma_z \rangle = 1$). The dynamics and variational plus dynamics approaches use the time step $\tau = 0.008\gamma^{-1}$. The results of two transformer layers are computed exactly under all POVM frame elements.

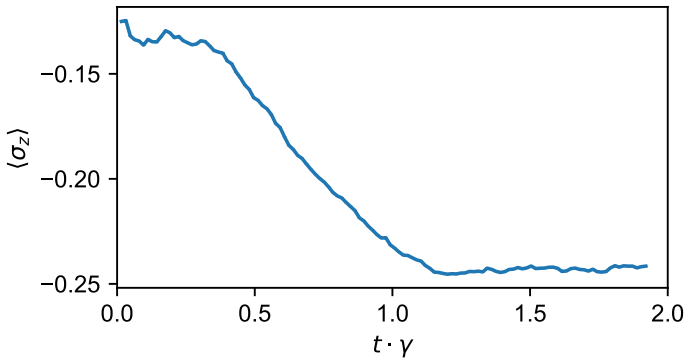


Figure 7.5: $\langle \sigma_z \rangle$ for variational plus dynamics result for 3×3 Heisenberg model with $B = 0$, $J_x = 0.9\gamma$, $J_y = 1.8\gamma$, and $J_z = \gamma$ using one transformer layer, 32 hidden dimensions and no strings. The Transformer is trained using a forward-backward trapezoid method with 12000 samples and a time step $\tau = 0.008\gamma^{-1}$.

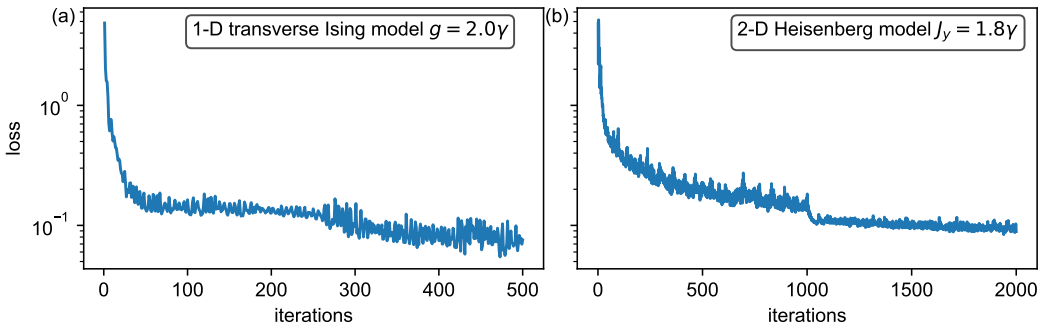


Figure 7.6: Examples of variational loss values. (a) loss values for 16 spins 1D TFIM with $V = g = 2\gamma$ using one transformer layer and 32 hidden dimensions. (b) loss values for 3×3 Heisenberg model with $B = 0$, $J_x = 0.9\gamma$, $J_y = 1.8\gamma$, and $J_z = \gamma$ using two layers, 64 hidden dimensions and no strings. The Transformer is trained using Adam [211] with a sample size of 12000.

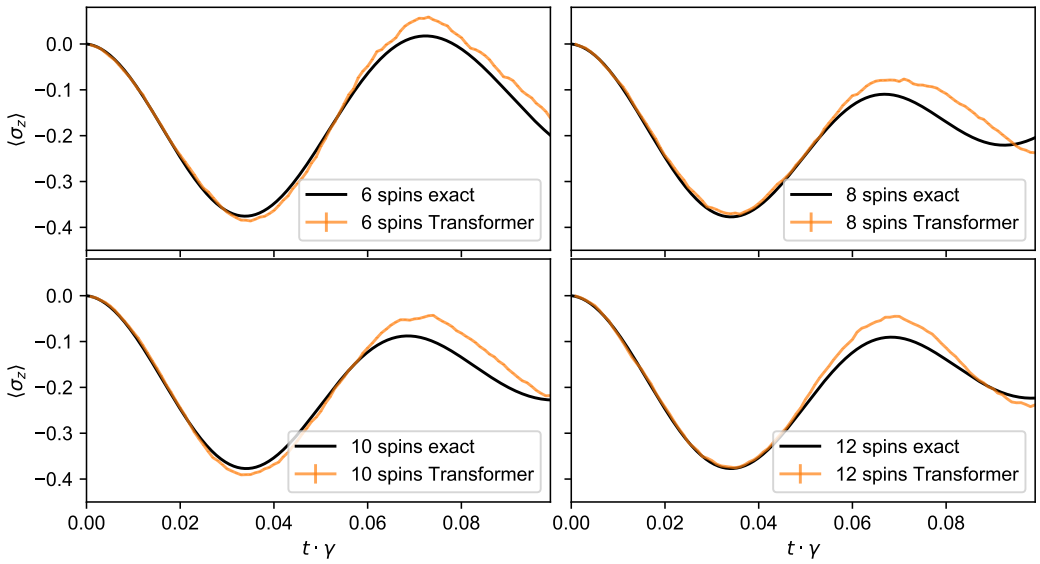


Figure 7.7: $\langle \sigma_z \rangle$ as a function of time computed with POVM dynamics. Various different system sizes of short time dynamics results for Heisenberg model in 1-D configuration with periodic boundary condition where $B = 10\gamma$, $J_x = 20\gamma$, $J_y = 0$, and $J_z = 10\gamma$. The initial state is a product state of $\prod_{i=1}^{N} |\leftarrow\rangle$ ($\langle \sigma_y \rangle = -1$). The exact curve is produced using QuTip [301, 302]. The Transformer is trained using a forward-backward trapezoid method with a sample size of 12000 and a time step of $0.0005\gamma^{-1}$. The neural network has one encoder layer and 32 hidden dimensions.

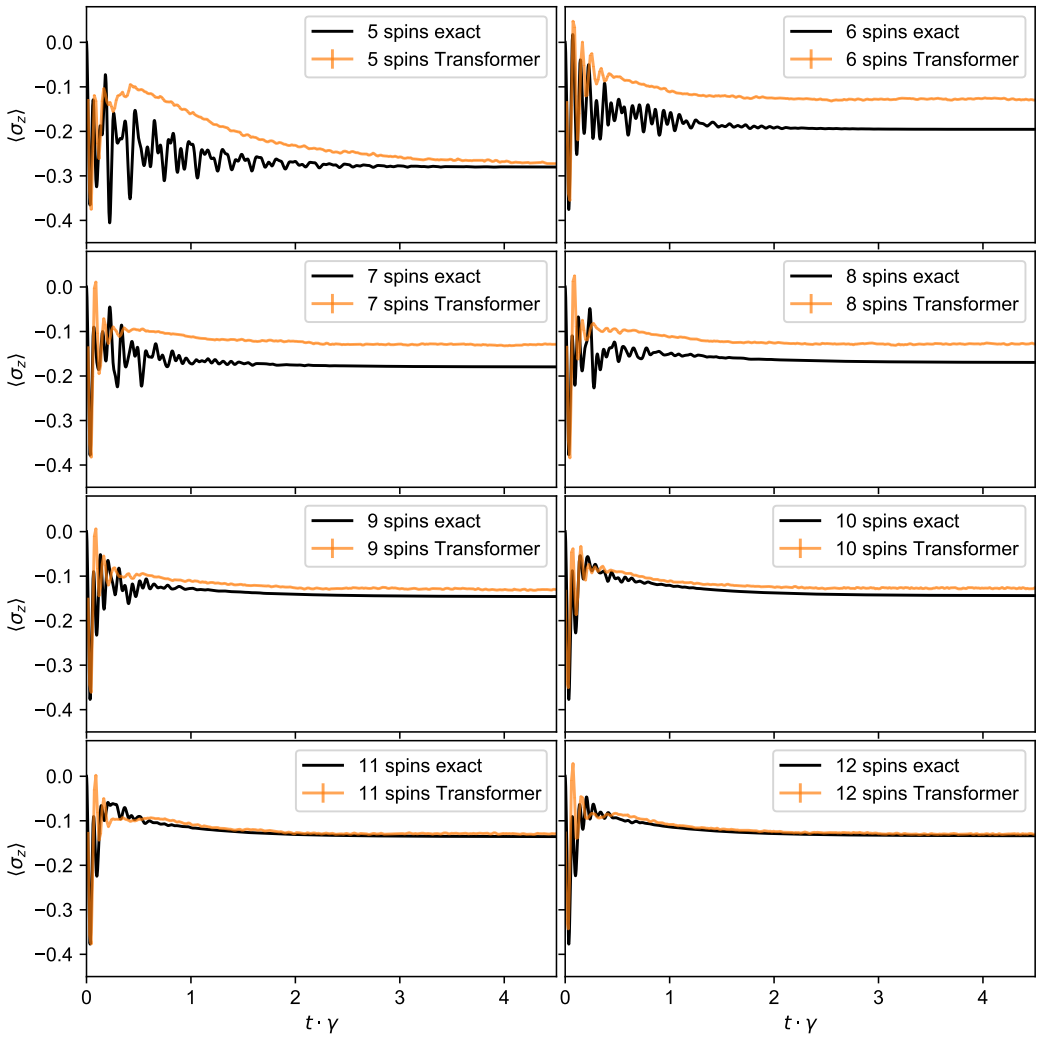


Figure 7.8: $\langle\sigma_z\rangle$ as a function of time computed with POVM dynamics. Various different system sizes of long time dynamics results for Heisenberg model in 1-D configuration with periodic boundary condition where $B = 10\gamma$, $J_x = 20\gamma$, $J_y = 0$, and $J_z = 10\gamma$. The initial state is a product state of $\prod_{i=1}^{N} |\leftarrow\rangle$ ($\langle\sigma_y\rangle = -1$). The exact curve is produced using QuTip [301, 302]. The neural network is trained using a forward-backward trapezoid method with a sample size of 12000 and a time step of $0.0075\gamma^{-1}$. The neural network has one encoder layer and 32 hidden dimensions.

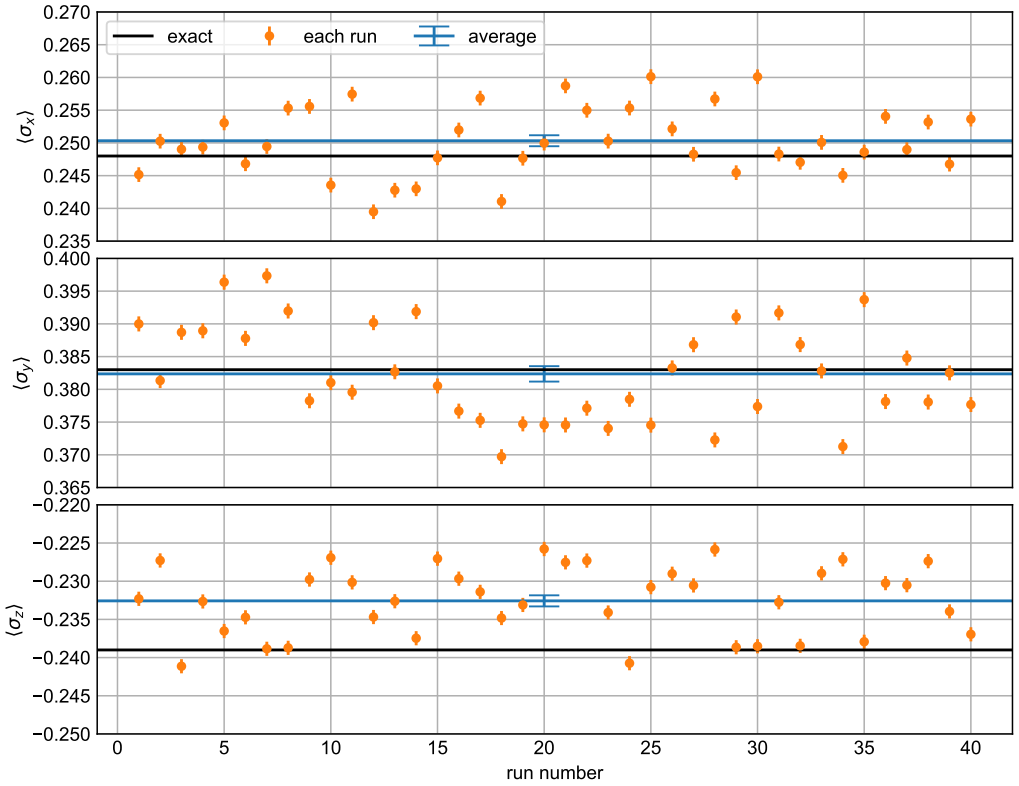


Figure 7.9: Data of multiple runs for variational steady state solution for one dimensional 16-site spin chain for the TFIM with periodic boundary condition where $V = g = 2\gamma$. The initial state is a product state of $\prod_{i=1}^{N} |\uparrow\rangle$ ($\langle\sigma_z\rangle = 1$). The neural network has one encoder layer and 32 hidden dimensions, and is trained using Adam [211] in 500 iterations with a sample size of 12000. The exact curve (black line) is digitized from Ref. [285].

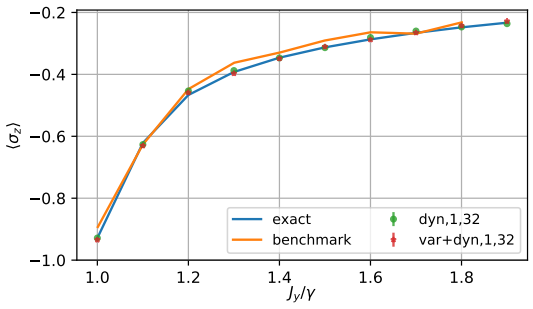


Figure 7.10: 3×3 Heisenberg model benchmarked with Ref. [135]. The exact curve (blue) is generated using QuTiP [301, 302]. The benchmark curve (orange) is from Ref. [135] with RBM. Our results (green and red) are the same as in the main paper, and the two numbers in the legend are layers, hidden dimensions.

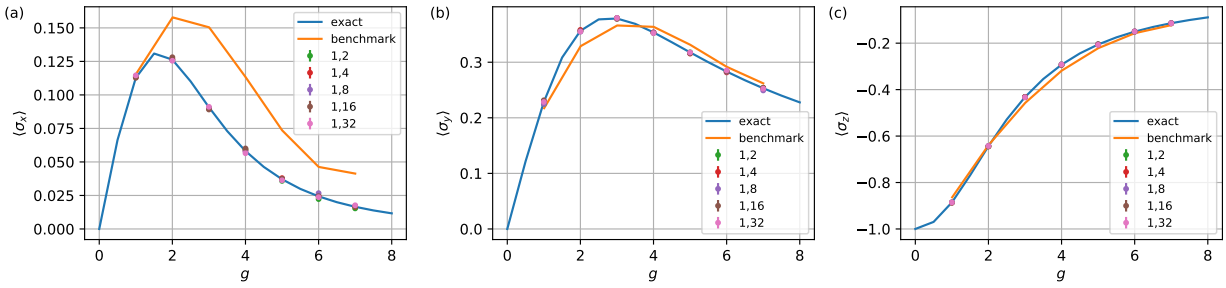


Figure 7.11: 8 qubit transverse-field Ising model benchmarked with Ref. [278]. The exact curve (blue) is generated using QuTiP [301, 302]. The benchmark curve (orange) is from Ref. [278]. The two numbers in the legend mean number of layers and hidden dimensions respectively. To be noted that the definition of g here is different from Ref. [278]. Here the convention is chosen as in our main paper, while the other authors choose a different convention. It is worth noting that the neural network is not designed to work for a hidden dimension less than 8, but the result here is still significantly better than Ref. [278].

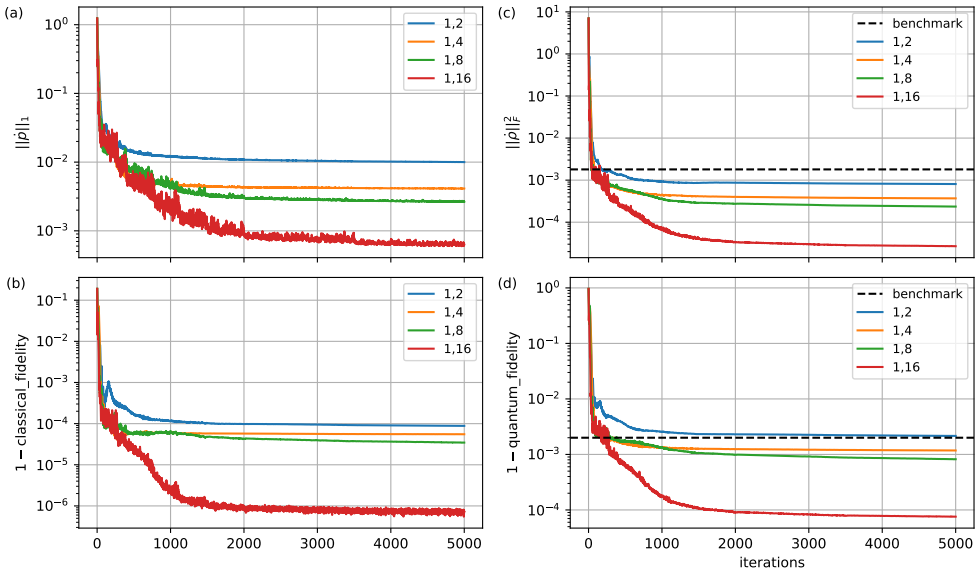


Figure 7.12: 4 qubit transverse-field Ising model benchmarked with Ref. [282]. The exact results are generated using QuTiP [301, 302]. The benchmark line (dashed black) is from Ref. [282]. The two numbers in the legend mean number of layers and hidden dimensions respectively. It is worth noting that the neural network is not designed to work for a hidden dimension less than 8, but the result here is still better than Ref. [282].

Chapter 8

Beyond many-body localized states in a spin-disordered Hubbard model with pseudo-spin symmetry

This chapter is written based on the research in Ref. [307]. It is also related to the preliminary results in Chapter 6 of Xiongjie Yu's thesis [308]. A prime characterization of many-body localized (MBL) systems is the entanglement of their eigenstates; in contrast to the typical ergodic phase whose eigenstates are volume law, MBL eigenstates obey an area law. In this work, we show that a spin-disordered Hubbard model has both a large number of area-law eigenstates as well as a large number of eigenstates whose entanglement scales logarithmically with system size (log-law). This model provides a microscopic Hamiltonian which is neither ergodic nor many-body localized. We establish these results through a combination of analytic arguments based on the eta-pairing operators combined with a numerical analysis of eigenstates. In addition, we describe and simulate a dynamic time evolution approach starting from product states through which one can separately probe the area law and log-law eigenstates in this system.

8.1 Introduction

Pioneered by the seminal works of Basko [309] and Gornyi [310], the many-body localization (MBL) transition is defined as a dynamical phase transition which happens at finite energy density for a disordered and isolated many-body interacting system. Conceptually, MBL is when Anderson localization [311, 312] survives inter-particle interactions. In MBL systems, under unitary time evolution, local observables fail to thermalize to their ergodic values.

Typical MBL models are disordered spin chains with short-ranged interactions in one dimension [313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340]. More recently, systems with itinerant degrees of freedom have been explored including disordered Hubbard or t-J models [341, 342, 343, 344]. This focus has been partially motivated by cold-atom experiments [345, 346, 347, 348].

Phenomenologically, the full MBL (FMBL) phase is characterized by a complete set of local integrals of motion (LIOM) [349, 350, 351, 352, 353, 354, 355, 356, 357, 319, 358, 359, 360], or the existence of a small

bond-dimension unitary tensor network (UTN) which diagonalizes the MBL Hamiltonian [319, 361]. A key application of the LIOMs or UTN is to explain the entanglement behavior of the MBL system. They imply that the entanglement of eigenstates are area law and that entanglement grows logarithmically under time evolution after a quench [362].

In this work, we report on a microscopic Hamiltonian which goes beyond the FMBL or ergodic phases. We show that this microscopic Hamiltonian has both constant (area law) as well as logarithmically entangled (log law) eigenstates. These eigenstates are interspersed throughout the spectrum (i.e. they don't make up a mobility edge). We then show how to probe separately the area-law and log-law eigenstates through time-evolution from simple product states giving potential access to these different types of states through cold-atom experiments.

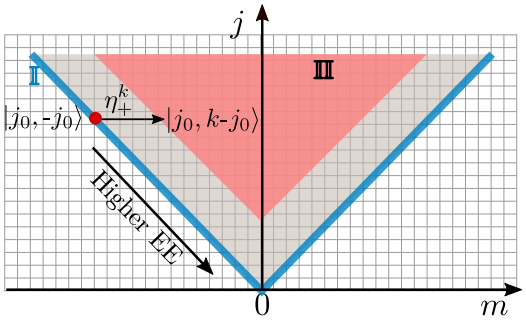


Figure 8.1: Schematic phase diagram of the spin-disordered Hubbard chain at large disorder in the 2D plane labeled by quantum numbers j and m. The eta-pairing raising operator η_+ acts as $\eta_+|E, S^z_{\text{total}}, j, m\rangle = |E+U, S^z_{\text{total}}, j, m+1\rangle$ moving eigenstates horizontally in this figure. The particle-hole transformation maps $|E, S^z_{\text{total}}, j, m\rangle$ to $|E-2mU, S^z_{\text{total}}, j, -m\rangle$, equivalent to a mirror symmetry about $m=0$. In the $S^z_{\text{total}}=0$ sector, the top left corner of the triangle is the vacuum state. **Region I** (blue) are reference states which are destroyed by η_- (left edge) or η_+ (right edge) and contain a mixture of area-law and log-law states. **Region II** (pink) is the region where all eigenstates' entanglement entropies have logarithmic correction due to repeated application of η_+ (left edge) or η_- (right edge).

8.1.1 Overview of Results

We consider a one-dimensional Hubbard model with spin disorder

$$H = -t\sum_{i\sigma}(c^\dagger_{i\sigma}c_{i+1\sigma} + h.c.) + \sum_i U n_{i\uparrow}n_{i\downarrow} + \sum_i h_i S^z_i, \tag{8.1}$$

where $S_i^z = (n_{i\uparrow} - n_{i\downarrow})/2$, and the disordered magnetic field $h_i \in [-W, W]$ is sampled uniformly. We focus on the case of $U = 1, t = 1$.

To scaffold this discussion, we first note that Eq. (8.1) has a pseudo-spin SU(2) symmetry [363, 364] (see Sec. 8.2), which allows us to label our eigenstates by four quantum numbers $|E, S^z_{\text{total}}, j, m\rangle$ including the energy E, total S_z and two quantum numbers j and m associated with the pseudo-spin symmetry. Because this symmetry is a continuous non-abelian symmetry, we don't expect to have a fully-MBL phase [365, 366]. Unless otherwise noted, we work with $S^z_{\text{total}} = 0$ (although our results generically apply to all S^z_{total}) and separately consider the entanglement of eigenstates in different quantum number sectors. Fig. 8.1 is a diagram of available quantum numbers and this figure will set the framework in which we discuss our results.

From the pseudo-spin algebra, one can analytically build towers of excited states of increasing m using the eta-pairing raising and lower operators η_+ and η_- [363, 364]; every application of η_+ moves states horizontally in Fig. 8.1. The blue line (region I) are the eigenstates at the bottom of the towers which we call reference states. We then consider eigenstates in region II which are generated from a reference state by the application of η_+^N for a given constant N. A common feature of this group of excited states is the large number of double occupancies. We will show that all such eigenstates have, at least, an additive logarithmic correction to their von Neumann entanglement entropy with respect to the reference state; this violates the area-law entanglement for a typical MBL phase. We accomplish this by identifying a particular sector of the reduced density matrix for these states that leads to the logarithmic correction (see Sec. 8.3). This extends results of Ref. [367] which recently showed such corrections in the case of the vacuum state (top left of Fig. 8.1) with applications to a non-disordered Hubbard model. Additionally, we show that any eigenstate which is made of only singlons has an exact logarithmic correction.

We then numerically consider a number of disorder realizations of the Hamiltonian in Eq. (8.1) for $L = 8$ at large W using the slope of the cut average entanglement (SCAEE), introduced in Ref. [325]. We find that the reference eigenstates contain a mixture of area-law and log-law states and that the full spectrum of eigenstates in region II do indeed exhibit a logarithmic increase in entanglement (see Sec. 8.4).

Having characterized the eigenstates, we then discuss how to separately probe the localization physics both from area-law states as well as from log-law states dynamically using time evolution (see Sec. 8.5). This would allow cold-atom experiments to directly probe this physics. We identify two extreme cases of product states – all single occupancies at quarter filling which occupy primarily area law states, and all double occupancies at half filling which occupy log-law states. We find that in former case the entanglement entropy grows logarithmically and the charge imbalance does not relax as is typical in a MBL system, while in

the latter the entanglement entropy grows as a power law (but not linear) fashion and the charge imbalance tends to fully relax, which is delocalized but not ergodic.

8.2 Introduction to Pseudo-spin Algebra

8.2.1 Pseudo-spin $SU(2)$ symmetry

It is easy to see that the spin disorder breaks the spin rotation symmetry of H. To prove that the pseudo-spin $SU(2)$ symmetry is intact, one can introduce the eta-pairing operators [363, 364], with notations for 1D specifically.

$$\eta_- = \sum_i (-1)^i c_{i\uparrow} c_{i\downarrow},\ \eta_+ = \eta_-^\dagger,\ \eta_0 = \frac{1}{2}(\hat{N} - L), \tag{8.2}$$

where L is the number of sites and has to be even, and $\hat{N}$ is the operator for total number of electrons in the system. The eta-pairing operators generate a $SU(2)$ algebra because

$$[\eta_0, \eta_\pm] = \pm\eta_\pm, \quad [\eta_+, \eta_-] = 2\eta_0. \tag{8.3}$$

To prove that pseudo-spin symmetry is preserved, one can straightforwardly check that

$$[H, \eta_\pm] = \pm U\eta_\pm,\ [H, \eta_0] = 0,\ [H, \vec{\eta}^2] = 0, \tag{8.4}$$

where the total pseudo-spin operator $\vec{\eta}^2$ is

$$\vec{\eta}^2 = \frac{1}{2}(\eta_+\eta_- + \eta_-\eta_+) + \eta_0^2. \tag{8.5}$$

Therefore, $\{H, S^z_{\text{total}}, \vec{\eta}^2, \eta_0\}$ is a complete set of commuting observables. For simplicity, we will denote the eigenstate of $\{\vec{\eta}^2, \eta_0\}$ as $|j, m\rangle$, with

$$\vec{\eta}^2|j, m\rangle = j(j+1)|j, m\rangle, \quad \eta_0|j, m\rangle = m|j, m\rangle, \tag{8.6}$$

where $|m| \leq j$.

Because of Eq. (8.3) and (8.4), $\eta_\pm$ are a pair of ladder operators for η_0 and the Hamiltonian H. Consider a simultaneous eigenstate $|E, S^z_{\text{total}}, j, m\rangle$. Applying η_+ to this eigenstate will lead to $|E+U, S^z_{\text{total}}, j, m+1\rangle$, and vice versa for η_-.

8.2.2 Particle-hole Transformation

The particle-hole (PH) transformation

$$c_{i\uparrow} \to (-1)^i c_{i\downarrow}^\dagger, \quad c_{i\downarrow} \to (-1)^i c_{i\uparrow}^\dagger \tag{8.7}$$

has some consequences for the eigenstates.

Firstly, S_i^z is invariant under the PH transformation. Secondly, we have

$$\eta_\pm \to \eta_\mp, \ \eta_0 \to -\eta_0, \ \vec{\eta}^2 \to \vec{\eta}^2. \tag{8.8}$$

Thirdly, the Hamiltonian H transforms as

$$H \to H - 2U\eta_0. \tag{8.9}$$

Therefore, from the wave function's perspective, an eigenstate $|E, S^z_{\text{total}}, j, m\rangle$ under the PH transformation becomes $|E - 2mU, S^z_{\text{total}}, j, -m\rangle$. Both these eigenstates will have the same entanglement; therefore, both the right and left edge of Fig. 8.1 can be considered reference states.

8.3 Entanglement Entropy of Eta-pairing States

The pseudo-spin $SU(2)$ algebra has direct influence on the eigenstates' energy and entanglement entropy. Given a reference eigenstate $|\psi_{\text{ref}}\rangle$ of H, one can build a tower of highly excited states with η_+,

$$|\psi^N\rangle = \mathcal{A}_N \eta_+^N |\psi_{\text{ref}}\rangle, \quad N \in \mathbb{N}^+, \tag{8.10}$$

where $\mathcal{A}_N$ is the normalization factor. With increasing N, $|\psi^N\rangle$ has increasing energy and number of doublons, until annihilated. We will call these excited states the eta-pairing states, and only consider the reference state $|\psi_{\text{ref}}\rangle$ which can be annihilated by η_- and has relatively small number of electrons (see Supplementary Materials 8.7.1 for details).

In this section, we prove two things: (A) eta-pairing states (with large enough N) have, at least, a logarithmically increasing entanglement with respect to its reference state and (B) eta-pairing states (with large enough N) whose reference state consist of only singlons have exactly a logarithmically increasing entanglement.

To accomplish this, we decompose $|\psi_{ref}\rangle = \sum_t |\psi_t\rangle$ into a linear superposition of terms labeled by property t which is preserved under the application of η_+^N. Additionally, the reduced density matrix is block diagonal in blocks labeled by t, i.e. $\rho_t = tr_B|\psi_t\rangle\langle\psi_t|$. It then follows that the entanglement entropy is a sum of these individual blocks. To determine the change of entanglement, we need consider only how the entanglement of each term $|\psi_t\rangle$ changes.

To prove (A), the property t is the spin polarization in the subsystem A. We consider only the term where $S_{A,z}$ is maximally polarized (i.e. $S_{A,z} = K/2$ for a system of K electrons) and show that this term has a logarithmically increasing entanglement. To prove (B), the property t is the singlon number in subsystem A and we can show that every term has a logarithmically increasing entanglement. Proving the logarithmic increase in entanglement uses a similar approach to Ref. [367]. In the subsections below we detail these claims.

8.3.1 Maximally Polarized Sector

In this subsection, we show that, for any eta-pairing eigenstate in region II of Fig. 8.1, the entanglement entropy grows at least logarithmically, whose contribution comes from the maximally polarized sector in the reduced density matrix.

Consider an eta-pairing state built on a many-body reference state with K electrons. Without loss of generality, let $S_z = 0$. Decompose $|\psi_{ref}\rangle$ into terms of fixed $S_{A,z}$. Notice that the operation of η_+ only adds doublons to a basis vector and therefore, can't change the value of $S_{A,z}$ except by destroying the state. Since S_z is fixed, $S_{B,z}$ will not change either when η_+ is applied. When we trace out B to calculate the reduced density matrix of A, the terms $|\psi_{t'}\rangle\langle\psi_t|$ where $t \neq t'$ will vanish because $|\psi_{t'}\rangle$ and $|\psi_t\rangle$ have different values $S_{B,z}$. As a result, the reduced density matrix will be block diagonal according to $S_{A,z}$.

Take a reference state with K electrons for the disordered Hubbard model.

$$|\psi_{\text{ref}}\rangle = \sum_{(i_1,\sigma_1),\cdots,(i_K,\sigma_K)} \alpha_{(i_1,\sigma_1),\cdots,(i_K,\sigma_K)} c^\dagger_{i_1,\sigma_1} \cdots c^\dagger_{i_K,\sigma_K} |0\rangle, \tag{8.11}$$

which satisfies $K \ll L/2$ and $\eta_-|\psi_{\text{ref}}\rangle = 0$. We consider the block in the reduced density matrix with maximum $S_{A,z}$ for which there is $K/2$ spin-up electrons in A and $K/2$ spin-down electrons in B.

$$|\psi_{K/2}\rangle = \sum_{(i_1,\uparrow),\cdots,(i_K,\downarrow)} \alpha_{(i_1,\uparrow),\cdots,(i_K,\downarrow)} c^\dagger_{i_1,\uparrow},\cdots,c^\dagger_{i_K,\downarrow}|0\rangle \tag{8.12}$$

$$= \sum_{i\in I, j\in J} \alpha_{i,j}\{c^\dagger_{i\uparrow}\}\{c^\dagger_{j\downarrow}\}|0\rangle. \tag{8.13}$$

I is the set of site sequences with only spin-up electrons. $\{c^\dagger_{i\uparrow}\}$ is the product of $c^\dagger_\uparrow$ from a particular site sequence i. Similar notation is used for J and $\{c^\dagger_{j\downarrow}\}$ for the case of spin-down electrons. We then perform a Schmidt decomposition on $|\psi_{K/2}\rangle$.

$$|\psi_{K/2}\rangle = \sum_{i\in I, j\in J} \alpha_{i,j}\{c^\dagger_{i\uparrow}\}\{c^\dagger_{j\downarrow}\}|0\rangle \tag{8.14}$$

$$= \sum_{i\in I, j\in J}\sum_k u_{ik}d_{kk}v_{kj}\{c^\dagger_{i\uparrow}\}\{c^\dagger_{j\downarrow}\}|0\rangle \tag{8.15}$$

$$= \sum_k d_{kk}(\sum_{i\in I} u_{ik}\{c^\dagger_{i\uparrow}\})(\sum_{j\in J} v_{kj}\{c^\dagger_{j\downarrow}\})|0\rangle \tag{8.16}$$

$$= \sum_k \alpha_k |k_{\uparrow,A}\rangle|k_{\downarrow,B}\rangle \tag{8.17}$$

Notice that we construct Schmidt vectors with $K/2$ spin-up singlons in A and $K/2$ spin-down singlons in B. By using the same contour integral technique as Ref. [367], the reduced density matrix has the following form.

$$\begin{aligned}\rho_A^{\text{single}} &= tr_B[\mathcal{A}_N\eta_+^N|\psi_{K/2}\rangle\langle\psi_{K/2}|\mathcal{A}_N\eta_-^N] \\ &= (\mathcal{A}_N N!)^2 \oint_o\oint_o \frac{dz_1dz_2^*}{(2\pi)^2}\sum_k \alpha_k^2\langle k_{\downarrow,B}| \\ &\frac{e^{z_2^*\eta_{-,B}}}{(z_2^*)^{N+1}}\frac{e^{z_1\eta_{+,B}}}{z_1^{N+1}}|k_{\downarrow,B}\rangle e^{z_1\eta_{+,A}}|k_{\uparrow,A}\rangle\langle k_{\uparrow,A}|e^{z_2^*\eta_{-,A}}\end{aligned} \tag{8.18}$$

$$\begin{aligned}\rho_A^{\text{single}} &= (\mathcal{A}_N N!)^2 \oint_o\oint_o \frac{dz_1dz_2^*}{(2\pi)^2}\sum_k \alpha_k^2\frac{(1+z_1z_2^*)^{L_B-K/2}}{(z_1z_2^*)^{N+1}} \\ &e^{z_1\eta_{+,A}}|k_{\uparrow,A}\rangle\langle k_{\uparrow,A}|e^{z_2^*\eta_{-,A}}\end{aligned} \tag{8.19}$$

where

$$\eta_{-,A/B} = \sum_{i\in A/B} (-1)^i c_{i\uparrow} c_{i\downarrow}, \quad \eta_{+,A/B} = \eta^\dagger_{-,A/B}. \tag{8.20}$$

After carrying out the contour integrals, we get

$$\rho_A^{\text{single}} = \sum_k \alpha_k^2 \sum_n \lambda_n |N-n, k_{\uparrow,A}\rangle\langle N-n, k_{\uparrow,A}| \tag{8.21}$$

$$\lambda_n = \frac{C^{N-n}_{L_A-K/2} C^n_{L_B-K/2}}{C^N_{L-K}}, \tag{8.22}$$

$$|N-n, k_{\uparrow,A}\rangle = \mathcal{A}^{L_A}_{N-n} \eta^{N-n}_{+,A} |k_{\uparrow,A}\rangle. \tag{8.23}$$

where $C^k_n = n!/[k!(n-k)!]$ is the combinatorial number, $\mathcal{A}^{L_A}_{N-n}$ is a normalization constant given by Eq. (8.46) by replacing L with the subsystem size L_A. It is crucial to choose the maximum polarized sector so that $|k_{\uparrow,A}\rangle$ is a reference state of $\eta_{+,A}$.

The entanglement entropy can be calculated as follows

$$S = -\sum_{k,n} \alpha_k^2 \lambda_n \ln(\alpha_k^2 \lambda_n) \tag{8.24}$$

$$= -\sum_{k,n} \alpha_k^2 \lambda_n \ln(\alpha_k^2) - \sum_{k,n} \alpha_k^2 \lambda_n \ln(\lambda_n) \tag{8.25}$$

$$= -\sum_{k} \alpha_k^2 \ln(\alpha_k^2) - \beta_A \sum_{n} \lambda_n \ln(\lambda_n) \tag{8.26}$$

$$= S_{ref} - \beta_A \sum_n \lambda_n \ln(\lambda_n) \tag{8.27}$$

where $S_{ref} = -\sum_k \alpha_k^2 \ln(\alpha_k^2)$, $\beta_A = \sum \alpha_k^2 = \sum_k d_{kk}^2$. Since the sum of the square of the singular values is equal to the Frobenius norm of the matrix, it follows that $\beta_A = \sum_{i\in I, j\in J} \alpha_{i,j}^2$. Eq. (8.27) is checked numerically with system size $L = 8$ and reference electron number $K = 2$ in Fig. 8.12.

Naturally, we are interested in the limit of highly excited states, large system size and large heat bath size. Therefore we take the limit of $L_B \gg L_A$, $N \gg L_A - K/2$, and $L_A \gg K/2$. In this case, one can simplify $S - S_{\text{ref}}$ using the Stirling approximation, replace the summation by an integral, and finally apply a saddle point approximation, which leads to

$$S - S_{\text{ref}} \approx \frac{\beta_A}{2}(1 + \ln[2\pi\nu(1-\nu)(L_A - K/2)]) \tag{8.28}$$

where $\nu = N/(L-K)$ indicates the portion of available sites taken by double occupancies. Notice that the symmetric appearance of $\nu(1-\nu)$ with respect to $\frac{1}{2}$ is a consequence of the particle-hole symmetry (see Fig. 8.15). We can clearly see the logarithmic contribution of $\ln(L_A - K/2)$ to the total von Neumann entanglement entropy.

8.3.2 Reference States with Only Singlons

In this subsection, we show that, for any eta-pairing eigenstate in region II of Fig. 8.1 whose reference state contains only singlons, the entanglement entropy grows exactly logarithmically in the thermodynamic limit.

Given a reference $|\psi_{ref}\rangle$ with only singlons, we decompose it into terms of fixed singlon number i in subsystem A. Notice that the operation of η_+ only adds doublons to a basis vector and therefore, can't change the value of i except by destroying the state. Since the total singlon number is fixed in the reference state, the singlon number in subsystem B will not change either when η_+ is applied. When we trace out B to calculate the reduced density matrix of A, the terms $|\psi_{t'}\rangle\langle\psi_t|$ where $t \neq t'$ will vanish because $|\psi_{t'}\rangle$ and $|\psi_t\rangle$ have different values in terms of singlon number in subsystem B. As a result, the reduced density matrix will be block diagonal according to the singlon number i in subsystem A.

Take a many particle reference state in the form of Eq. (8.40) with all K electrons to be singlons. We consider the component $|\psi_i\rangle$ with i singlons in subsystem A. Following the same calculation as in the previous subsection, we perform a Schmidt decomposition and have

$$|\psi_i\rangle = \sum_k \alpha_{i,k} |k_i\rangle |k_{K-i}\rangle \tag{8.29}$$

where we construct Schmidt vectors of i singlons in A and $K-i$ singlons in B.

Using the same contour integral technique, one can show that the reduced density matrix from this reference state has the following form.

$$\rho_{i,A}^{\text{single}} = \sum_{k,n} \alpha_{i,k}^2 \lambda_{i,n} |\phi_{i,n}^A\rangle\langle\phi_{i,n}^A| \tag{8.30}$$

where $\lambda_{i,n} = \frac{C^n_{L_B-(K-i)} C^{N-n}_{L_A-i}}{C^N_{L-K}}$ and $|\phi_{i,n}^A\rangle$ is a set of orthonormal basis.

It follows that the entanglement entropy has the form.

$$S_{i,A}^{\text{single}} = S_{i,ref} - \beta_{i,A} \sum_n \lambda_{i,n} \ln \lambda_{i,n} \tag{8.31}$$

where $S_{i,ref} = -\sum_{i,k} \alpha_{i,k}^2 \ln \alpha_{i,k}^2$, $\beta_{i,A} = \sum \alpha_{i,n}^2 = \sum_{(i_1,\sigma_1),\cdots,(i_K,\sigma_K)} \alpha^2_{(i_1,\sigma_1),\cdots,(i_K,\sigma_K)}$ with the constraint

that there are i singlons in A and $K-i$ singlons in B.

Summing up the entanglement entropy contribution from different singlon number i sectors, the total entanglement entropy is

$$S_{total} = \sum_i S_{i,ref} - \sum_{i,n} \beta_{i,A} \lambda_{i,n} \ln \lambda_{i,n} \tag{8.32}$$

Eq. (8.32) is consistent with numerical result with system size $L = 8$ and reference electron number $K = 2$ in Fig. 8.13.

In the thermodynamic limit of $L_B \gg L_A$, $N \gg L_A - K$, $L_A \gg K$, and $N + K < L$, we can apply Stirling approximation and saddle point approximation to each $S_{i,A}^{\text{single}}$, which implies an additive scaling of $\ln(L_A - i)$ to the entanglement entropy. The result is true for all $i \leq K$ sectors as long as the reference state has only singlon occupancies. Therefore, the total entanglement entropy of the eta-pairing states built on reference states with only singlon occupancies will have logarithmically more entanglement entropy than the reference states.

8.4 Numerical Evaluation of Entanglement

In the previous section, we proved that there is, at least, an additive logarithmic increase in entanglement from the reference state in various contexts. While this rules out that eta-pairing states are area law, these proofs are not sufficient to determine the actual entanglement entropy because we don't know the entanglement of the reference state nor whether the increase in entanglement is greater then logarithmic. In this section we take steps to answer these questions numerically.

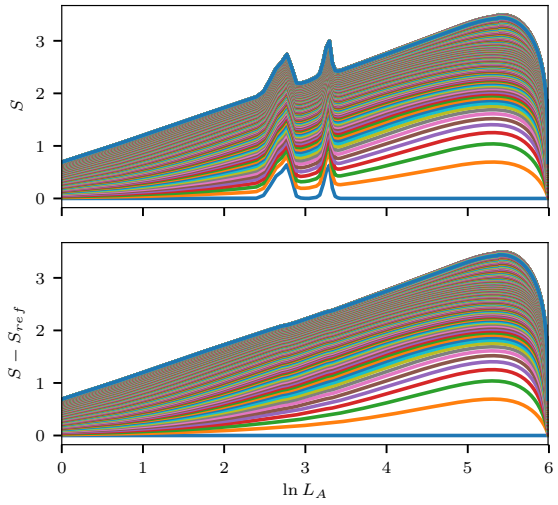


Figure 8.2: **Top:** Entanglement entropy $S(L_A)$ of $\mathcal{A}_N \eta_+^N |\psi_{ref}\rangle$ for various N (starting at $N = 0$ for the bottom curve) with $|\psi_{ref}\rangle$ as a single disordered realization of the open-boundary condition two-particle ground state of Eq. (8.1) with $W = 4, t = 1$ and $U = 1$. **Bottom:** Entanglement entropy difference between $\mathcal{A}_N \eta_+^N |\psi_{ref}\rangle$ and $|\psi_{ref}\rangle$ for various N.

The simplest reference states we can consider are ground states. We consider the ground state of Hamiltonian in Eq. (8.1) for $W = 4$ in the sector where total $S_z = 0$ and $K = 2$ for $M = 400$ total sites. This state is close to the top of region I in Fig. 8.1. We apply η_+ many times and see a clear logarithmic increase in entanglement (see Fig. 8.2). Strictly speaking, under open boundary conditions, pseudo-spin symmetry is no longer exact, but this does not seem to be a problem at large system sizes. While our proof in Sec. 8.3.1 does not forbid a faster growth of entanglement, we do not see it in this case.

We then consider the entanglement entropy of eigenstates in the middle of the spectrum. We consider the entanglement of these states for a system of size $L = 8$. For various disorder strengths, we compute the cut averaged entanglement entropy (CAEE) and the slope of the cut-averaged entanglement entropy (SCAEE) [325]. Note that the SCAEE equals to 1 at all L_A for an infinite-temperature volume law state and zero for large enough L_A for an area law state. See Fig. 8.3 for a histogram of these results. We find the SCAEE consistent with a volume law at small disorder strengths. At larger disorder strength, though, there is a broad distribution of the SCAEE with some states exhibiting area-law behavior and some states exhibiting sub-volume non-area law behavior. Together with the unusual behavior of CAEE in Fig. 8.19, it suggests a possible non-ergodic, non-MBL phase in the model. More details on the disorder dependence of entanglement can be found in Supplementary Materials. 8.7.6.

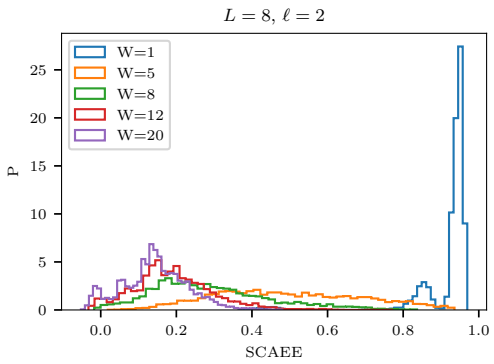


Figure 8.3: Probability density of SCAEE at $W = 1, 5, 8, 12, 20$, for $L = 8$ and a subsystem size of $L_A = 2$.

To understand this better, we start by considering the probability density of the SCAEE in reference states of various quantum-number sectors at subsystem size $L_A = 2$ for system size $L = 8$ (see Fig. 8.4). We find a bimodal distribution with one of the peaks centered at zero and the other peak at a non-zero value much less than 1 again suggesting a mix of area-law and sub-volume law states (see Fig. 8.4(bottom), Fig. 8.16, Fig. 8.17). We check this by considering the disordered average entanglement for both peaks. The peak centered at zero is clearly area-law while the other peak is consistent with logarithmically growing entanglement (see Fig. 8.4).

We then proceed to consider the difference in entanglement between the reference states and the eta-pairing states. The numerical results (see Fig. 8.5) are consistent with all states have a logarithmic increase in entanglement. This is interesting given that this is in a regime where the proof is not applicable and much of the entanglement comes from sectors other then the maximally polarized. See Fig. 8.14 for the non-disordered average version of this curve.

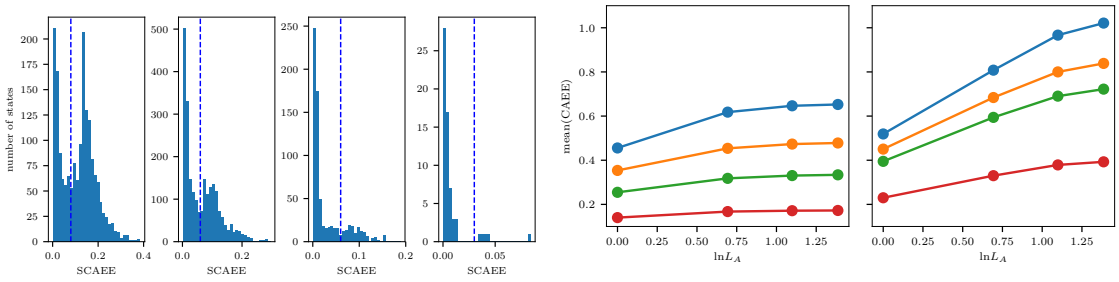


Figure 8.4: **Top:** The SCAEE histograms at reference states corresponding to (from left to right) $j = 0, 1, 2, 3$ with $L = 8$, subsystem size $l = 2$, $S_z = 0$ and $W = 14$. **Bottom:** Mean cut-averaged entanglement entropy (CAEE) vs. $\ln L_A$ for different reference states [$j = 0$ (blue curve), $j = 1$ (orange curve), $j = 2$ (green curve), $j = 3$ (red curve)] for the eigenstates with SCAEE value on the left (right) side of the dashed line corresponding to the left (right) figure.

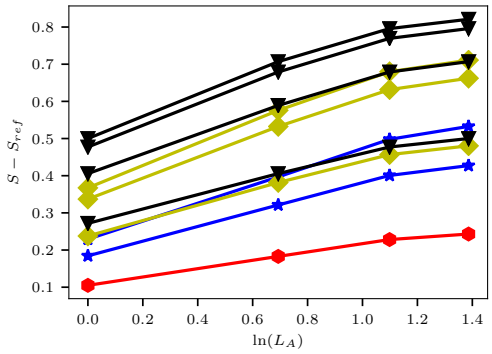


Figure 8.5: The entanglement entropy difference vs. $\ln(L_A)$ for $L = 8$, $S_z = 0$, and $W = 14$ at different quantum number sectors[$j = 4$ (black curve), $j = 3$ (yellow curve), $j = 2$ (blue curve), $j = 1$ (red curve)]. The entanglement entropy for each quantum number sector is averaged over the entropy of all the eigenstates in the sector obtained from exact diagonalization.

Finally, we note that for a reference state of only singlons, Eq. (8.32) exactly implies the (logarithmic) increase in entanglement. While our reference states don't typically have only singlons, we can check the efficacy of this formula as a function of the number of non-singlons in our system. We find (see Fig. 8.6) that the formula still is applicable with small deviations when the average number of non-singlons is close to zero.

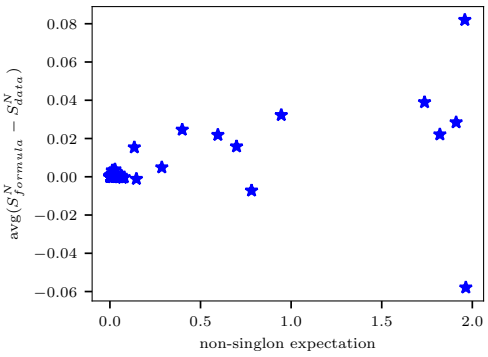


Figure 8.6: Entanglement entropy difference for $L = 8$, $S_z = 0$ and $W = 14$ between Eq. (8.32) and the exact entanglement entropy vs. average non-singlon number in the quantum number sector $j = 3, m = 0$. The entanglement entropy difference is averaged over all L_A. When the non-singlon number is close to zero, Eq. (8.32) provides a good description for the total entanglement entropy.

8.5 Time Evolution after Quantum Quench

We have given evidence for a Hamiltonian which has both area-law and log-law eigenstates. Here we show how these different eigenstates can be probed using time-evolution. In the process this will give further evidence for the two types of states as well as supply a physical picture for why we might expect this difference.

To accomplish this, our goal will be to find states that are simple to prepare, such as product states, that have overlap with primarily area-law or log-law eigenstates and then consider the effect of time-evolution on these states. We will consider two product states: a quarter filled singlon state, shown in Fig. 8.8(top), and a half filled doublon state, shown in Fig. 8.8(bottom).

For a product state $|p\rangle$ (in the occupation basis) the number of single occupancies n_1 and double occupancies n_2 fix the expected value of the quantum numbers (j, m),

$$\langle p|\eta_0|p\rangle = -\frac{(L - n_1 - 2n_2)}{2} \tag{8.33}$$

and

$$\langle p|\vec{\eta}^2|p\rangle = \frac{L - n_1 - 2n_2}{2}\left(\frac{L - n_1 - 2n_2}{2} + 1\right) + n_2. \tag{8.34}$$

Quarter filled singlon state: To find area-law states we must focus away from states with overlap

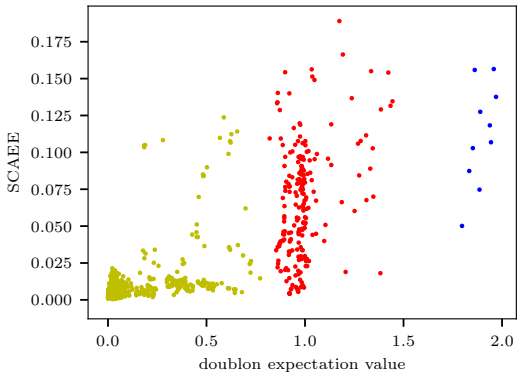


Figure 8.7: SCAEE vs. doublon expectation value for $L = 8$, $l = 2$, $S_z = 0$ and $W = 14$ in quantum number sector $j = 2, m = -2$. For doublon expectation value close to zero, we see area-law entanglement.

in region II (which can't be area-law) and instead on those with high overlap with the reference states. A product state $|p\rangle$ with only single occupancy, is always an eigenstate of $\{\vec{\eta}^2, \eta_0\}$ of eigenvalues $(\frac{L-n_1}{2}, -\frac{L-n_1}{2})$ and therefore is a linear superposition of only reference states in one quantum number sector. Moreover, of those states, we find (see Fig. 8.7) that states with low doublon number are area law states. An area law state should be 'many-body localized' and so we generically expect that time evolution starting in such states should not equilibrate.

Half filled doublon state: On the other hand, to find log-law states, we can look for product states which have high overlap in region II. While the average single and double occupancy doesn't fix the quantum number sector it localizes it around a given quantum number sector. For $n_2 < \frac{L}{2}$, as n_2 increases, $\langle \eta_0 \rangle$ grows towards 0, while $\langle \vec{\eta}^2 \rangle$ decreases towards $\frac{L}{2}$. For $n_2 > \frac{L}{2}$, as n_2 increases, $\langle \eta_0 \rangle$ approaches $\frac{L}{2}$, while $\langle \vec{\eta}^2 \rangle$ increases towards $\frac{L}{2}(\frac{L}{2}+1)$. For either case, one can see that the half-filled doublon state is composed of eta-pairing states high up on long pseudo-spin ladders, which have logarithmic corrections to their entanglement entropy. As a log-law state we expect less localization than a MBL state.

Both of the product states start with zero entanglement entropy, and highly imbalanced charge distributions between even and odd sites. By considering the time evolution of doublon number in both quarter filling and half filling settings, we can verify that the doublon number is largely localized (see Fig. 8.9).

With these two initial product states, we investigate the time evolution of von Neumann entanglement entropy, and charge imbalance

$$I = \frac{\sum_j (-1)^j (n_{j\uparrow} + n_{j\downarrow})}{\sum_j (n_{j\uparrow} + n_{j\downarrow})}. \tag{8.35}$$

We also look at staggered magnetization in Supplementary Materials 8.7.5. The main goal is to see the rate of entanglement entropy growth and whether the charge imbalance relaxes.

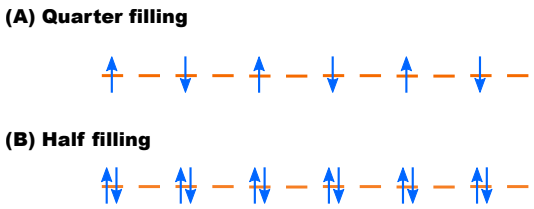


Figure 8.8: **A:** A schematic diagram for the quarter filled singlon state used in unitary time evolution. **B:** A schematic diagram for half filled doublon state used in unitary time evolution. Both states have a charge imbalance per electron of 1.

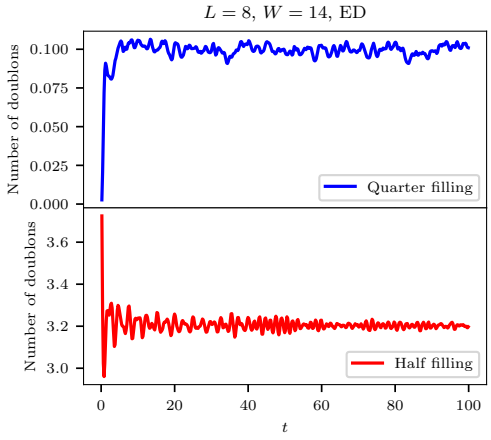


Figure 8.9: Ensemble averaged number of doublon for $W = 14, L = 8$ with quarter filled singlon state and half filled doublon states. Same samples are used as in the first column of Fig. 8.10. The initial doublon number is zero for the quarter filled state and four for the half filled state.

The real time evolution simulations are carried out separately for $L = 8$, which uses the exact diagonalization (ED) method, and for $L = 12$ and $L = 24$, which uses the time-evolving block decimation (TEBD) method based on the open source ITensor library [368]. We consider disorder strengths $W = 14$. Under each simulation, the entanglement entropy, charge imbalance and staggered magnetization are averaged over disorder realizations.

We find (see Fig. 8.10, Fig. 8.11) that the quarter filled singlon case exhibits logarithmic growth in entanglement entropy and a charge imbalance that, after an initial decay, never relaxes stabilizing around a non-zero value. This is as expected for a many-body localized state. On the other hand, the half filled doublon case exhibits a power-law growth of entanglement as well as a charge-imbalance which decays quickly to zero. Although this is suggestive of thermalization, the slope of the entanglement is significantly below the expected linear growth of an ergodic phase. We attribute this difference to the logarithmic as opposed to volume-law entanglement of the eigenstates. Further evidences from the disorder dependence of dynamics

are presented in Supplementary Materials. 8.7.7.

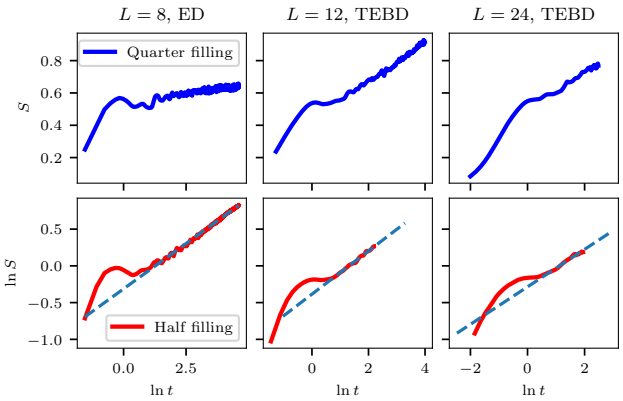


Figure 8.10: Ensemble averaged von Neumann entropy S for $W = 14$ with quarter filled singlon and half filled doublon initial product states. **First column:** $L = 8$, exact diagonalization and periodic boundary conditions. Results are averaged over 400 (top) and 150 (bottom) samples respectively. **Second column:** $L = 12$, TEBD, and open boundary conditions. Results were averaged over 210 samples. **Third column:** $L = 24$, TEBD, and open boundary conditions. Results were averaged over 200 (top) samples and 210 (bottom) samples respectively. For the quarter filling case (blue curves), the entropy grows logarithmically with respect to time. For the half filling case (red curves), the entropy grows as a power law with t, with the power law exponent equal to 0.25 for $L = 8$, 0.29 for $L = 12$ and 0.26 for $L = 24$.

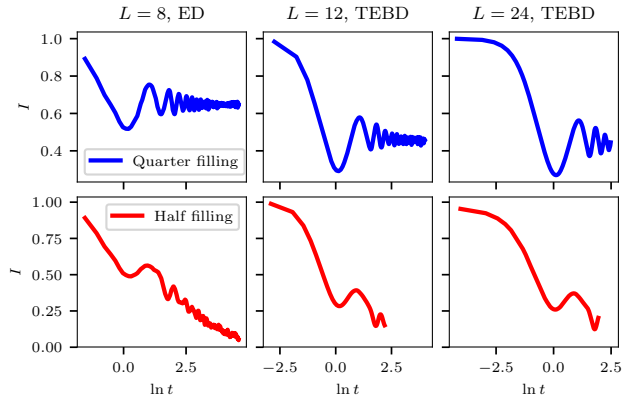


Figure 8.11: Ensemble averaged charge imbalance I using the same samples and parameters as Fig. 8.10.

There is a simple physical picture consistent with these results. Since double occupancy has $S_z = 0$, spin-up and spin-down electrons can hop together through a second order process, which leads to full charge delocalization in the half filled setting. However, single occupancies can not hop freely due to spin disorder,

which prevents full charge delocalization in the quarter filling case. Under the spin disorder potential, the double occupancy tends to hop together and creates charge relaxation.

From the above analysis, it is clear that the quarter filled singlon product state acts many-body localized while the half filled doublon product state is neither fully ergodic nor MBL. Moreover, via time-evolution we see that we can directly probe the area-law and log-law parts of the spectrum opening up the possibility that this effect can be seen experimentally.

8.6 Conclusion

Overall, then we have two pieces of evidence - entanglement and dynamics - which suggests the spin-disordered Hubbard model is neither in the MBL phase nor in the ergodic phase. Instead, at large disorder, it has both area and log-law eigenstates throughout the spectrum.

From Fig. 8.3 and Fig. 8.19, we see that there appear to be three qualitatively different types of eigenstates in this model as we tune disorder. At $L = 8$ and small disorder (i.e. $W \leq 4$), we find a slope of the cut-averaged entanglement entropy (SCAEE) at $L_A = 2$ which is peaked near one, the expected value for an infinite temperature volume law in the thermodynamic limit. At $W \gtrsim 6$ there is a qualitative change in the distribution of eigenstates. Instead, the distribution is peaked at a low SCAEE eventually approaching a value near 0.1 at larger disorder (i.e. $W \approx 15$). This suggests a plurality of eigenstates which are neither volume law nor area law, which would (for large L_A and L) corresponds to a SCAEE of zero. Further validation of a phase transition out of the ergodic phase is suggested by a maxima in the standard deviation of the CAEE (see Fig. 8.19). At disorder $W \geq 9$, an additional peak at near-zero SCAEE suggests a fraction of the states are also area-law (we anticipate such a peak would extend to lower W for larger L_A, L).

We better probe the nature of these eigenstates at large disorder by considering the eigenstates in regions I and II from Fig. 8.1. In region I, we separately consider the SCAEE of the eigenstates in various quantum number sectors. We again (see Fig. 8.4) find a bimodal distribution for the SCAEE of eigenstates. By considering the eigenstates in the peak away from zero, we see their entanglement is best fit by a logarithmically growing entanglement (Fig. 8.4(right)). The area law states in region I can be probed dynamically by choosing an all singlon product state. Such a state only has overlap in one quantum number sector in region I and is biased toward the area-law states (see Fig. 8.7). We consider the time evolution of a quarter-filled and half-filled singlon state. We find that both states have entanglement entropy which scales logarithmically in time (as anticipated for a MBL state) and whose charge imbalance and staggered magnetization respectively do not equilibrate (see Fig. 8.10, Fig. 8.11, Fig. 8.21, Fig. 8.22). This behavior

is consistent with the area-law nature of the states seen in entanglement.

Instead in region II, area law states are analytically forbidden (see Sec. 8.3). Here, we considered the difference in entanglement between a given quantum number sector in region II and its respective reference state in region I. We find for a two-electron ground state (an area law state) and $N = 400$ that the states in region II differ logarithmically (see Fig. 8.2). For smaller $N = 8$ but for all eigenstates, we also see a difference which is best fit by a logarithmic entanglement (see Fig. 8.5). Given the log-law and area-law nature of region I this means that region II contain log-law states. By dynamically time evolving with a product state of half-filled doublons, we probe eigenstates in quantum number sectors with $m = 0$ which have overlap in region II. This state's entanglement scales polynomially but not linearly under time evolution (see Fig. 8.10) and has a charge imbalance I which appears to eventually equilibrate (see Fig. 8.11).

Our work provides a microscopic Hamiltonian that demonstrates the possible existence of a non-ergodic, non-MBL phase in a one-dimensional system. Such phases will not have local integrals of motion nor small unitary tensor networks. This work opens up the possibility of different entanglement structures beyond the area-law of many body localized state in disordered systems.

8.7 Supplementary Materials

8.7.1 Reference States

Note that a reference state has the property that

$$\eta_-|\psi_{\text{ref}}\rangle = 0. \tag{8.36}$$

This implies that any eigenstate which has overlap with any product state of only singlons is a reference state. This follows because we know that $\eta_+\eta_-$ acted on a non-reference eigenstate gives back the eigenstate and since $\eta_\pm$ only affects doublons, any single-occupancy-only configuration will be annihilated by η_-.

For a K-electron eigenstate $|\psi_{\text{ref}}\rangle$ which can be annihilated by η_-, we have

$$\eta_0|\psi_{\text{ref}}\rangle = -\frac{L-K}{2}|\psi_{\text{ref}}\rangle, \tag{8.37}$$

Because of Eq. (8.3), it then follows that

$$\vec{\eta}^2|\psi_{\text{ref}}\rangle = (\eta_+\eta_- - \eta_0 + \eta_0^2)|\psi_{\text{ref}}\rangle, \tag{8.38}$$

which can be reduced to

$$\vec{\eta}^2|\psi_{\text{ref}}\rangle = \frac{L-K}{2}\left(\frac{L-K}{2}+1\right)|\psi_{\text{ref}}\rangle, \tag{8.39}$$

meaning that $|\psi_{\text{ref}}\rangle$ has pseudo-spin quantum number of $(\frac{L-K}{2}, -\frac{L-K}{2})$ for $(\vec{\eta}^2, \eta_0)$, which then can be raised by η_+ for at most $(L-K)$ times.

8.7.2 Normalization Factor of Eta-pairing States Built from Many Particle Reference State

Assume that we have a K-particle eigenstate of the spin-disordered Hubbard model as

$$|\psi_{\text{ref}}\rangle = \sum_{(i_1,\sigma_1),\cdots,(i_K,\sigma_K)} \alpha_{(i_1,\sigma_1),\cdots,(i_K,\sigma_K)} c^\dagger_{i_1,\sigma_1}\cdots c^\dagger_{i_K,\sigma_K}|0\rangle, \tag{8.40}$$

which satisfies $K \ll L/2$ and $\eta_-|\psi_{\text{ref}}\rangle = 0$

On top of this eigenstate, one can also build a tower of eta-pairing states as

$$|\psi^N\rangle = \mathcal{A}_N \eta_+^N \sum_{(i_1,\sigma_1),\cdots,(i_K,\sigma_K)} \alpha_{(i_1,\sigma_1)\cdots(i_K,\sigma_K)} c^\dagger_{i_1,\sigma_1}\cdots c^\dagger_{i_K,\sigma_K}|0\rangle. \tag{8.41}$$

But now it becomes too complicated to calculate the normalization factor directly.

To move forward, one should consider Eq. (8.3), (8.37), and (8.39). It is clear that

$$\eta_-\eta_+|\psi_{\text{ref}}\rangle = (L-K)|\psi_{\text{ref}}\rangle, \tag{8.42}$$

and

$$\eta_-^2\eta_+^2|\psi_{\text{ref}}\rangle = 2(L-K)(L-K-1)|\psi_{\text{ref}}\rangle, \tag{8.43}$$

Using mathematical induction, is quite easy to show that

$$\eta_-^N\eta_+^N|\psi_{\text{ref}}\rangle = C^N_{L-K}(N!)^2|\psi_{\text{ref}}\rangle, \tag{8.44}$$

which means that

$$\frac{1}{\mathcal{A}_N^2} = \langle\psi_{\text{ref}}|\eta_-^N\eta_+^N|\psi_{\text{ref}}\rangle = C^N_{L-K}(N!)^2. \tag{8.45}$$

So we finally arrive at

$$\mathcal{A}_N = \sqrt{\frac{(L-N-K)!}{(L-K)!N!}}. \tag{8.46}$$

8.7.3 Additional Numerical Evidence on Eta-pairing State Entanglement Entropy

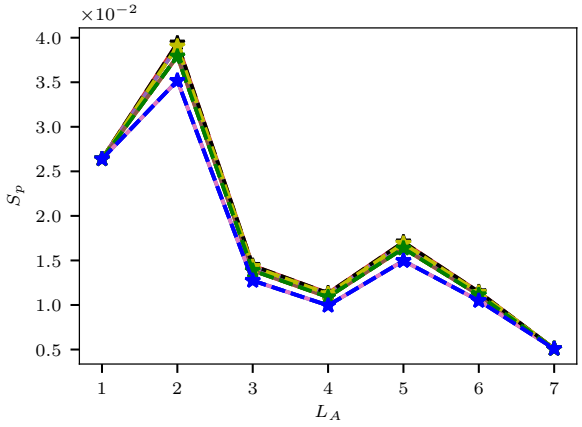


Figure 8.12: Entanglement entropy from maximum polarized sector vs. subsystem size L_A for eta-pairing state with η_+^N [$N = 0$ (blue curve), $N = 1$ (green curve),$N = 2$ (orange curve), $N = 3$ (black curve)]. The numerical data is indicated by $*$ and is consistent with the lines generated by Eq. (8.27). The sample is selected from reference state with average singlon number equal to 0.036, which is far away from the reference state electron number $K = 2$. System size $L = 8$, polarization $S_z = 0$ and disorder strength $W = 14$.

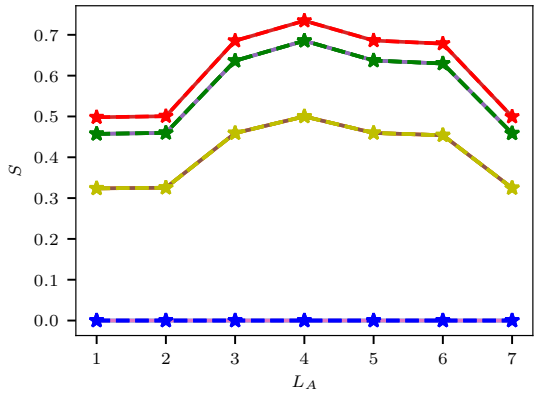


Figure 8.13: Entanglement entropy vs. subsystem size L_A for eta-pairing with η_+^N [$N = 0$ (blue curve), $N = 1$ (orange curve), $N = 2$ (green curve), $N = 3$ (red curve)]. The numerical data is indicated by $*$ and it is consistent with the lines generated by Eq. (8.32). The sample is selected from reference state with average singlon number almost equal to the reference state electron number $K = 2$. System size $L = 8$, polarization $S_z = 0$ and disorder strength $W = 14$.

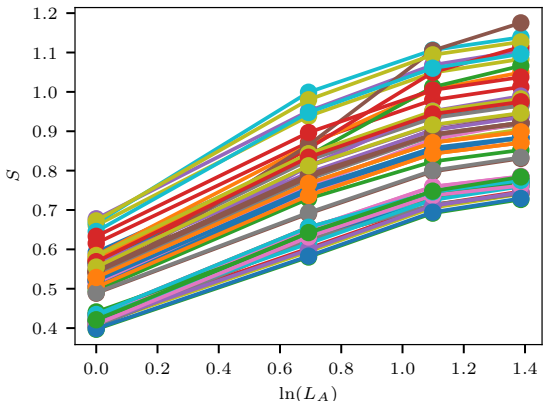


Figure 8.14: The entanglement entropy difference vs. $\ln(L_A)$ for all states in the quantum number sector $j = 3, m = 0$ for $L = 8$, $S_z = 0$ and $W = 14$. The entanglement entropy for each quantum number sector is averaged over the entropy of all the eigenstates in that sector.

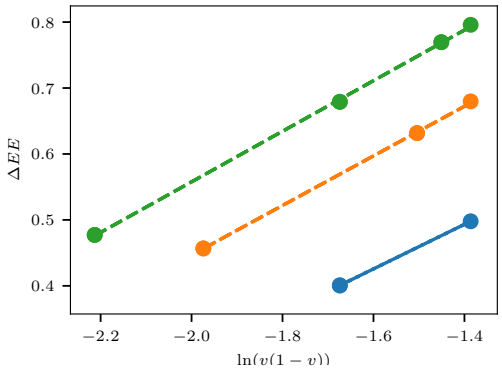


Figure 8.15: Entanglement entropy difference between different quantum number sectors and their reference states [$j = 4$ (blue curve), $j = 3$ (green curve), $j = 2$ (red curve)] vs. $\ln[\nu(1-\nu)]$. ν is defined as $N/(L-K)$, where N is the number of times of applying the η_+ operators on the reference state and K is the number of electrons in the reference state. The entanglement entropy is linear with respect to $\ln[\nu(1-\nu)]$, which agrees with Eq. (8.28).

Eq. (8.28) indicates that for fixed L_A in the limit $N \gg L_A - K \gg 0$, the entanglement in the single-occupancy reference state sector should increase logarithmically with $\nu(\nu-1)$. In spite of not being in this regime, for the $L = 8$ case we interestingly find that the entanglement entropy of the states in the same quantum number sector j increases logarithmically in $\nu(1-\nu)$.

8.7.4 Bimodal Distribution of Reference States Entanglement Entropy

Here provides further details on the probability density of the SCAEE in reference states of various quantum-number sectors at subsystem size $L_A = 2$ for system size $L = 8$ (see Fig. 8.4).

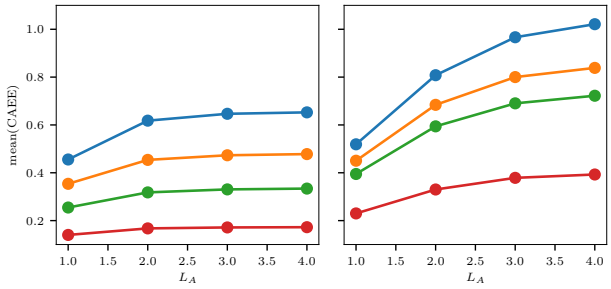


Figure 8.16: Mean cut-averaged entanglement entropy (CAEE) vs. L_A for different reference states [$j = 0$ (blue curve), $j = 1$ (orange curve), $j = 2$ (green curve), $j = 3$ (red curve)]. L_A is the subsystem size. System size $L = 8$, polarization $S_z = 0$ and disorder strength $W = 14$. **Left:** The CAEE in each reference state sector is averaged over the eigenstates with SCAEE value on the left hand side of the dashed line in Fig. 8.4. The mean CAEE in this case indicates a area law. **Right:** The CAEE in each reference state sector is averaged over the eigenstates with SCAEE value on the right hand side of the dashed line in Fig. 8.4. The mean CAEE in this case indicates an sub-volume law.

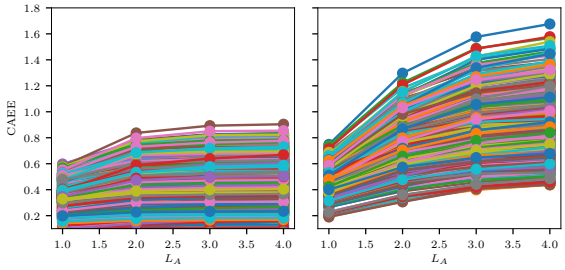


Figure 8.17: Cut-averaged entanglement entropy (CAEE) vs. L_A for reference state sector at $j = 1$. L_A is the subsystem size. System size $L = 8$, polarization $S_z = 0$ and disorder strength $W = 14$. **Left:** The CAEE in each reference state sector is from the eigenstates with SCAEE value on the left hand side of the dashed line in Fig. 8.4. The CAEE in this case indicates a area law. **Right:** The CAEE in each reference state sector is from the eigenstates with SCAEE value on the right hand side of the dashed line in Fig. 8.4. The CAEE in this case indicates an sub-volume law.

8.7.5 Staggered Magnetization

We also look at the time evolution of the staggered magnetization.

$$M_s = \frac{\sum_j (-1)^j (n_{j\uparrow} - n_{j\downarrow})}{\sum_j (n_{j\uparrow} + n_{j\downarrow})}. \tag{8.47}$$

We have chosen the initial state to be the half filled singlon state, whose staggered magnetization starts from 1 (see Fig. 8.18(Top)). Since the initial state has no doublon, it is an area-law state (see Fig. 8.18(Middle)). Due to the large spin disorder, the spin degree of freedom is localized and the final staggered magnetization stabilizes at a value far away from zero (see Fig. 8.18(Bottom)).

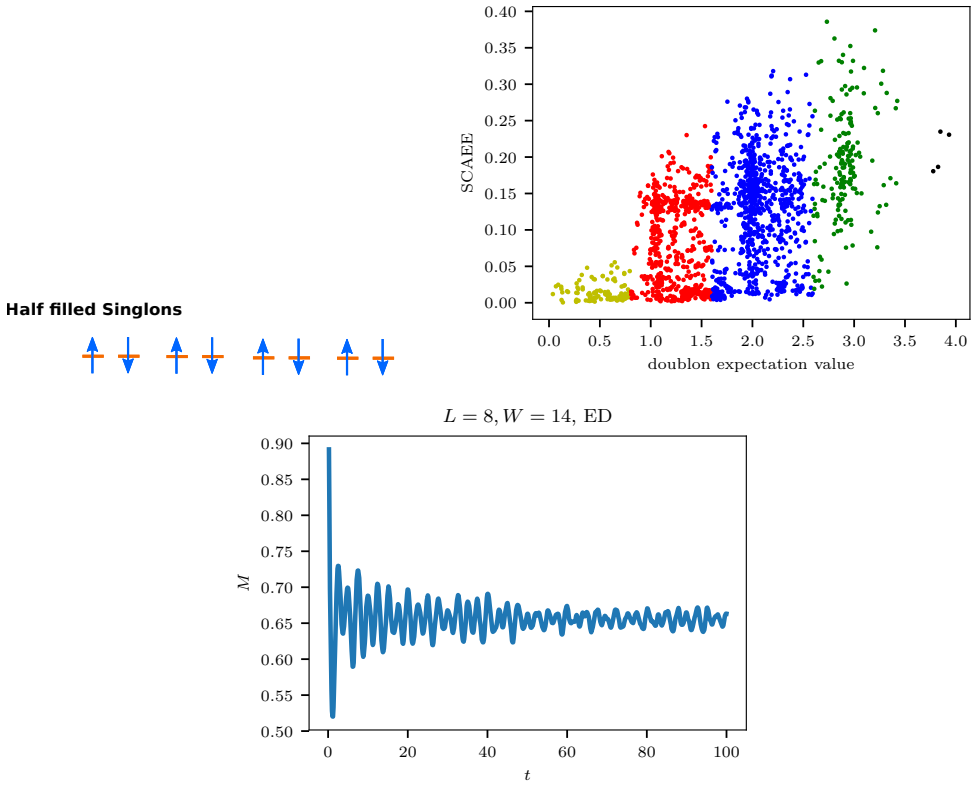


Figure 8.18: **Top:** Schematic diagram for the half filled singlon state. **Middle:** SCAEE vs. doublon expectation value for $L = 8$, $l = 2$, $S_z = 0$ and $W = 14$ in quantum number sector $j = 0, m = 0$. **Bottom:** Disorder averaged (over 200 samples) time evolution of staggered magnetization M_s of the half filled singlon state for $L = 8, W = 14$. The initial staggered magnetization is 1.

8.7.6 Disorder Dependence of Entanglement

In this section, we consider the disorder dependence of entanglement. To do this, we start by considering the CAEE as a function of W. We first note that the standard deviation of the CAEE has a peak around $W \approx 5$ for $L = 8$ suggesting a transition between two different types of entanglement. By looking at the histogram of the slope of the cut averaged entanglement entropy (SCAEE) we can better understand the distribution of entanglement. At $W \leq 4$ the distribution of the SCAEE is peaked at slope close to 1.0 as would signify an infinite temperature volume law. At $W \geq 5$ the slope is instead peaked at a much smaller non-zero value. This suggests that the plurality of states are neither volume law nor area law (which should be peaked at zero for large enough L_A and L). It is reasonable to anticipate that these are then primarily log-law states and this scaling is consistent with the measured scaling at $W = 14$. Notice, in addition for all $W > 8$ there is non-trivial number of eigenstates with zero slope and in particular a non-trivial peak at zero for $W > 11$. This suggests the existence of a non-trivial number of area law states. The fact that these states become visible only at W somewhat larger than the transition is not surprising. For W closer to the transition, the subsystem size L_A at which the CAEE should saturate to zero slope is expected to increase. Since $L_A = 2$ is chosen at Fig. 8.20, the zero slope for W closer to the transition may not appear to be obvious.

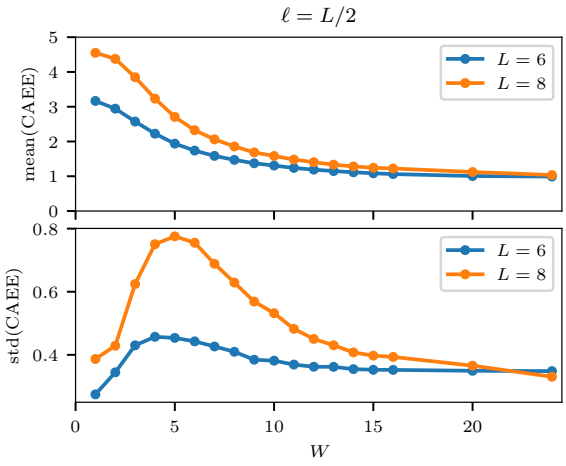


Figure 8.19: Mean (**top**) and standard deviation (**bottom**) of the half-cut CAEE vs. the disorder strength, for $L = 6, 8$, with 100 disorder samples for each cases, and 30 to 50 eigenstates per sample at the middle of the spectrum in the half filling, $S^z_{\text{total}} = 0$ sector. The standard deviation of CAEE is more peaked at $L = 8$. Two noticeable differences from the normal MBL transition can be observed. (1) At very large W, CAEE stays around 1, instead of zero. (2) For both the mean and the standard deviation of CAEE curves, the crossing point between different system sizes does not show up until very large W. These could be signs of a non-ergodic, non-MBL phase.

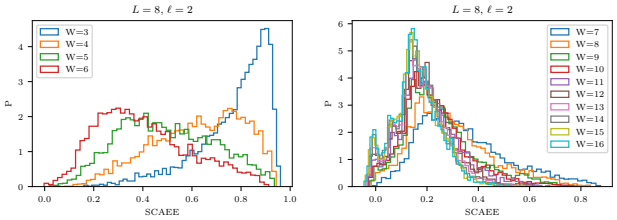


Figure 8.20: **Left:** Probability density of SCAEE at $W = 3 \sim 6$, for $L = 8$ and a subsystem size of $L_A = 2$. **Right:** Probability density of SCAEE at $W = 7 \sim 16$, for $L = 8$ and a subsystem size of $L_A = 2$. The results are averaged over 100 disorder samples for each disorder strength W.

8.7.7 Disorder Dependence of Dynamics

In the main body of the text, we have focused on $W = 14$ for the dynamics. As larger disorder should have a smaller correlation length, this should minimize finite-size effects while still not being at such a large disorder that Anderson localization effects dominate; one way to validate the latter is to see the logarithmic increase in entanglement which distinguishes MBL from Anderson localization. In this section, we focus on

the effect of disorder dependence on the dynamics.

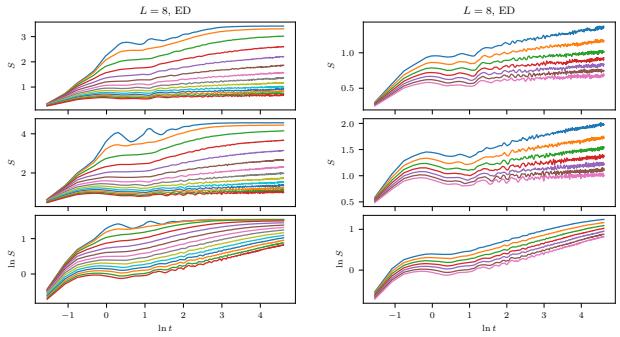


Figure 8.21: Time evolution of entanglement entropy for $L = 8$ under different disorder strength W. Disorder goes in increments of 1 from bottom to top. **Left:** $1 \leq W \leq 14$ **Right:** $8 \leq W \leq 14$; Results are averaged over 200 disorder samples except for half filled doublon $W = 13$ (173 samples) and $W = 14$ (150 samples). **First row:** Quarter filled singlon initial state (Fig. 8.8). **Second row:** Half filled singlon initial state (Fig. 8.18). **Third row:** Half filled doublon initial state (Fig. 8.8).

As in the main body of the text, we look at the dynamics of three product states: the quarter-filled singlon state, the half-filled singlon state and the half-filled doublon state (always shown in panels top, middle, and bottom respectively). The quarter filled singlon state has a mid-bond entanglement which grows logarithmically in time for $W \gtrsim 5$ and a change-imbalance which doesn't appear to equilibrate for $W \gtrsim 3$. The half-filled singlon state has a mid-bond entanglement which also grows logarithmically in time out to $t = 100$ for $W \gtrsim 5$ and whose staggered magnetization doesn't equilibrate for $W \gtrsim 3$. This is consistent with a transition in the range $3 \leq W \leq 5$ between area-law states, which are probed by the singlon product states, and ergodic states, which are volume law. For the half-filled state, instead we see that at large $W \gtrsim 5$ the entanglement grows polynomially but not linearly. We find that the charge imbalance equilibrates for all W.

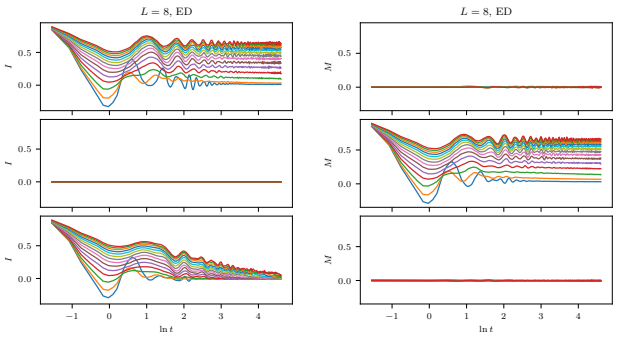


Figure 8.22: Time evolution of charge imbalance **(left)** and staggered magnetization **(right)** for $L = 8$ under different disorder strengths $1 \leq W \leq 14$ in increments of 1 starting from the bottom. Results are averaged over 200 disorder samples except for the half filled doublon $W = 13$ (173 samples) and $W = 14$ (150 samples). **First row:** Quarter filled singlon initial state (Fig. 8.8). **Second row:** Half filled singlon initial state (Fig. 8.18). **Third row:** Half filled doublon initial state (Fig. 8.8).

Chapter 9

Framework for simulating gauge theories with dipolar spin systems

This chapter is written based on the research in Ref. [16]. Gauge theories appear broadly in physics, ranging from the Standard Model of particle physics to long-wavelength descriptions of topological systems in condensed matter. However, systems with sign problems are largely inaccessible to classical computations and also beyond the current limitations of digital quantum hardware. In this work, we develop an analog approach to simulating gauge theories with an experimental setup that employs dipolar spins (molecules or Rydberg atoms). We consider molecules fixed in space and interacting through dipole-dipole interactions, avoiding the need for itinerant degrees of freedom. Each molecule represents either a site or gauge degree of freedom, and Gauss's law is preserved by a direct and programmatic tuning of positions and internal state energies. This approach can be regarded as a form of analog systems programming and charts a path forward for near-term quantum simulation. As a first step, we numerically validate this scheme in a small system study of U(1) quantum link models in (1+1) dimensions with link spin $S = 1/2$ and $S = 1$ and illustrate how dynamical phenomena such as string inversion and string breaking could be observed in near-term experiments. Our work brings together methods from atomic and molecular physics, condensed matter physics, high energy physics, and quantum information science for the study of nonperturbative processes in gauge theories.

9.1 Introduction

Gauge theories are fundamental to descriptions of a wide range of phenomena in many areas of physics. Euclidean lattice field theory [369] has been developed into a general quantitative tool for the study of nonperturbative phenomena of gauge theories using Monte Carlo simulations on classical computers. In particular, the use of lattice gauge theory (LGT) to study quantum chromodynamics (QCD) has yielded profound insights into its nonperturbative dynamics and hence our understanding of particle and nuclear physics [370, 371, 372, 373, 374, 375, 376]. However, problems involving nonequilibrium dynamics or systems described by complex actions suffer from the sign problem and are not easily amenable to Euclidean LGT

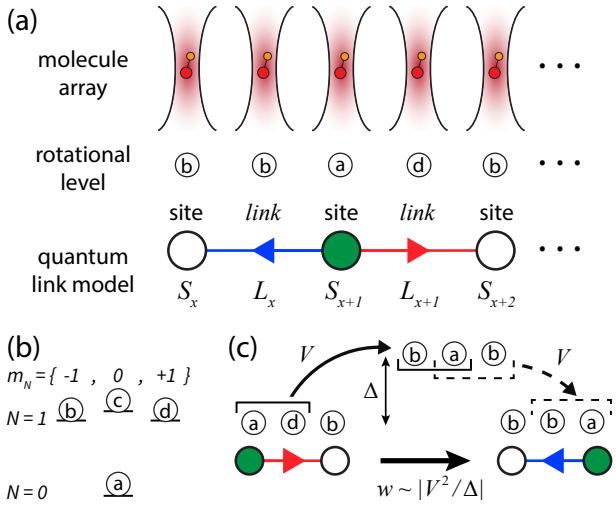


Figure 9.1: **Emulating quantum link models (QLMs) with arrays of dipolar molecules.** (a) Mapping between the rotational levels of molecules in an array and the sites and links of the QLM for spin $S = 1/2$. The designation of particular molecules as sites or links (S_x, L_x, S_{x+1}, L_{x+1} for a given unit cell) is enforced through local laser control of level-dependent light shifts. (b) Low-lying molecular rotational levels $|N, m_N\rangle$ and their redefinition in terms of states $|a\rangle$, $|b\rangle$, $|c\rangle$, and $|d\rangle$. (c) The hopping of "fermions" between sites and the associated spin operations on the links are realized by a second-order dipolar exchange of rotational excitations.

simulations. Recent approaches based on tensor networks have been useful for investigating low-dimensional LGTs [377], but their extension to higher dimensions may not be straightforward, due to the entanglement constraint.

The possibility of using quantum computations or simulations for these problems has led to renewed interest in Hamiltonian formulations [378]. Quantum link models (QLMs) [379, 380, 381], an alternative formulation of LGTs, are well suited to studies of real-time dynamics on both classical and quantum devices. They use quantum spins in finite integer and half-integer representations of S to replace the infinite gauge degrees of freedom.

Digital quantum computers, typically based on qubits and quantum circuits, are still in the noisy intermediate-scale 50 quantum (NISQ) era and are limited by both qubit number and gate depth due to noise and decoherence [382]. Still, for small systems [383] and with the aid of variational techniques [384], digital approaches to the quantum simulation of gauge theory dynamics have shown recent success. Alternatively, analog quantum simulators — special purpose quantum systems designed to implement the real-time evolution of model Hamiltonians — may be more suitable for near-term investigations of LGT dynamics on moderate to large systems [385, 386, 377, 387, 388, 389, 390, 391, 392, 393]. While some problems are emulated naturally on physical platforms, such as realizing the Hubbard model with cold atoms in optical lattices [394, 395] or realizing Heisenberg spin models with arrays of polar molecules or Rydberg

atoms [396, 397, 398, 399, 400], matter-gauge dynamics is not. Here, we show that the physics of matter coupled to dynamical gauge fields can arise naturally in dipolar spin systems [401, 402] if the elementary dipolar processes are restricted in a way that effectively imposes gauge invariance. Specifically, our approach relies on encoding the LGT Hamiltonian into a set of physically realizable degrees of freedom, for example in the internal states of polar molecules. This type of *analog systems programming* or hardware-specific encoding is an intermediate approach between pure analog emulators, whose microscopic degrees of freedom closely match the emulated model, and fully digital simulations.

The encoding of matter-gauge dynamics into a pure spin model with nonitinerant particles addresses one of the main challenges facing the simulation of gauge theories based on atomic Hubbard models [403, 404]: the challenge of removing motional entropy and mitigating sources of heating [405]. Such issues that plague itinerant systems are further compounded in schemes based upon spin- or species-dependent optical lattices [403], due to off-resonant light scattering [406, 407]. These issues are avoided in pure spin systems, as motional entropy can be divorced from the dynamics of internal degrees of freedom [408] initialized with near-zero entropy.

Recent theory work has shown that in experiments with Rydberg atom arrays [409], LGT dynamics arises from Ising spin models by integrating out fermion fields [410, 411]. The phenomenology of field theories, such as confinement, has also been demonstrated [412, 413] in other Ising spin systems. The reduction of the required degrees of freedom by integration over fermion fields is a powerful approach, but it typically introduces projectors onto certain gauge field states that impose Gauss's law. For general link spin S, this can require unphysical forms of long-range interactions, and thus would be difficult to implement in analog simulations. In our proposed approach we explicitly represent both the matter and gauge degrees of freedom with the matter-gauge dynamics arising due to dipole-mediated hopping of spin excitations in an array of nonitinerant dipolar spins. Gauge invariance, namely, Gauss' law, is imposed in this construction by the application of local, state-dependent energy shifts that serve to constrain the dynamics of the spin excitations. As a first step, we specifically consider how a platform of trapped dipolar molecules with control of internal state energies can realize the analog simulation of a U(1) quantum link model in (1+1) dimensions with spin $S = 1/2$ and $S = 1$. This approach relies on the dipolar nature of the spin-spin interactions, and could alternatively be realized in arrays of Rydberg atoms [400, 414] or other dipolar systems [415]. We describe the detailed mapping between the states and parameters of the base molecular spin Hamiltonian and the target QLMs. Our numerical simulations show that this scheme can allow for high fidelity analog simulations of dynamical phenomena fundamental to LGTs under realistic experimental conditions.

9.2 Quantum Link models

The local interaction terms of a gauge theory between matter and gauge-boson fields are imposed by the underlying gauge symmetry. In lattice gauge theory, the interaction terms involve matter fields at neighboring lattice sites. In order to preserve gauge invariance even at finite lattice spacing, the gauge bosons are usually represented by so-called link fields [369], which take continuous values being elements of a continuous gauge group. In the quantum link model version of LGTs, the link variables are instead represented by noncommuting, finite-dimensional operators that are analogous to quantum spin operators, a feature that makes QLMs more directly accessible to quantum simulation. Here, we consider QLMs of a U(1) LGT in $1+1$ dimensions in the Hamiltonian formulation with staggered fermions [378], which take the form

$$\begin{aligned} H_{\mathrm{QLM}} = & - w \sum_{x} \left[\psi_x^{\dagger} U_{x,x+1} \psi_{x+1} + \psi_{x+1}^{\dagger} U_{x,x+1}^{\dagger} \psi_x \right] \\ & + m \sum_{x} (-1)^x \psi_x^{\dagger} \psi_x + \frac{g^2}{2} \sum_{x} E_{x,x+1}^2, \end{aligned} \tag{9.1}$$

where x labels the spatial lattice sites, ψ_x is the fermion operator with the staggered mass $m(-1)^x$, $w > 0$ is the hopping parameter, $U_{x,x+1}$ is the link variable, $E_{x,x+1}$ is the electric flux for the U(1) gauge field on the link between x and $x+1$, and g is the gauge coupling [378, 381]. In this paper we focus on QLMs with $S = 1/2$ and $S = 1$ representations for the link variables, but this may be generalized to larger S. Note that the continuous gauge symmetry is recovered in the $S \to \infty$ limit [403]. The physical Hilbert space of the QLM is constrained by the gauge symmetry through Gauss's law and the gauge transformations are generated by the the Gauss-law operator $\widetilde{G}_x = \psi_x^{\dagger}\psi_x - E_{x,x+1} + E_{x-1,x} + \frac{1}{2}[(-1)^x - 1]$, where the last term stems from using staggered fermions.

9.3 Dipolar Molecules

A dipolar molecule is effectively a quantum rotor with angular momentum and projection eigenstates $|N_\alpha, m_{N_\alpha}\rangle$. We restrict our consideration to electronic, vibrational, and hyperfine ground states in the absence of large dc electric fields. Furthermore, we only consider rotational angular momentum states with $N_\alpha \in \{0, 1\}$, hereafter using the notation $|a\rangle \equiv |0,0\rangle$, $|b\rangle \equiv |1,-1\rangle$, $|c\rangle \equiv |1,0\rangle$ and $|d\rangle \equiv |1,1\rangle$ for the states

$S = 1/2$		$S = 1$	
Molecular Levels	QLM States	Molecular Levels	QLM States
$\|a\rangle_S$	Occupied	$\|a\rangle_S$	Occupied
$\|b\rangle_S$	Unoccupied	$\|c\rangle_S$	Unoccupied
$\|b\rangle_L$	$S^3 = -1/2$	$\|d\rangle_L$	$S^3 = -1$
$\|d\rangle_L$	$S^3 = 1/2$	$\|b\rangle_L$	$S^3 = 0$
		$\|c\rangle_L$	$S^3 = 1$

Table 9.1: Mapping from molecule levels to QLM states.

we consider (see Fig. 9.1). The Hamiltonian for a system of fixed dipolar molecules (DMH) is

$$H_{\mathrm{DMH}} = \sum_{i,\alpha} \left(2hB_{\mathrm{rot}} N_\alpha (N_\alpha + 1) + \epsilon_{i,\alpha}\right) b^\dagger_{i,\alpha} b_{i,\alpha} + \frac{1}{2} \sum_{i,j} \sum_{\alpha,\beta,\gamma,\eta} V_{i,j}^{\alpha,\beta;\gamma,\eta} b^\dagger_{i,\gamma} b^\dagger_{j,\eta} b_{j,\beta} b_{i,\alpha}. \tag{9.2}$$

where $b^\dagger$ and b are hard-core bosonic creation and annihilation operators, B_{rot} is the molecule's rotational constant, i, j label molecular positions, and $\alpha, \beta, \gamma, \eta$ label angular momentum states. The $\epsilon_{i,\alpha}$ are the additional position- and/or state-dependent contributions to the single-molecule energies, which include both spatially uniform but state-dependent terms arising from weak electric fields or nuclear-rotational coupling [398], as well as position-dependent and state-dependent energy shifts, which can be engineered through local differential ac Stark shifts owing to the anisotropic polarizability of molecules [416]. In particular, local control over molecular light shifts could be naturally incorporated in molecule arrays with individual laser addressing [417]. The second line of Eq. (9.2) represents the dipole-dipole interaction $V_{i,j}^{\alpha,\beta;\gamma,\eta}$, where $\alpha, \beta; \gamma, \eta$ label pairs of initial and final angular momentum states. The forms of the interaction terms are naturally restricted by the dipole selection rules $\Delta N = \pm 1$ and $\Delta m_N = 0, \pm 1$, but importantly their dipolar nature allows for the populations of the various molecular levels to be nonconserved (i.e., dipolar interactions allow for the interconversion of rotational and orbital angular momentum [418]). Experimentally, one can use laser power and polarization to tune the relative $\epsilon_{i,\alpha}$ terms, and we consider control of the $V_{i,j}^{\alpha,\beta;\gamma,\delta}$ magnitudes through control of intermolecular distances.

9.4 Effective Hamiltonian

We now generate a map between the parameters of the QLM and the physical parameters of the DMH. In our mapping, every site and link in the QLM maps to a different individual molecule; the Hilbert spaces

are mapped as in Table 9.1. In this construction, the rotational levels on the "site" molecules are used to represent the "fermions" (which are hard core bosons in one dimension) while rotational levels on the "link" molecules represent the link gauge fields. For a QLM with staggered fermions, each unit cell corresponds to two sites (odd x and even x) and two links, with the link between sites S_x and S_{x+1} labeled as L_x.

Letting H_0 be the one-body terms in the first line of Eq. (9.2), we tune the energies $\epsilon_{i,\alpha}$ so that Gauss's law is satisfied in the QLM and so that all of the dipolar configurations satisfying Gauss's law are nearly degenerate in H_0 while all of the other configurations are separated from the Gauss-law configurations by an energy scale $\Delta \gg V$. While we keep $m, g^2 \ll V \ll \Delta$ to preserve this condition, this still allows for the tuning of m and $g^2/2$ on scales comparable to the hopping w. If the molecular system is prepared s in an initial state that satisfies Gauss's law, energy constraints will ensure that the time-evolved state will remain in the physical Hilbert space.

The hopping term $-w\sum_x[\psi_x^\dagger U_{x,x+1}\psi_{x+1} + \text{H.c.}]$ of the QLM involves two sites and one link, and implies changes in the states of three molecules, S_x, L_x, and S_{x+1}. Because the DMH [Eq. (9.2)] contains only two-body interactions, matching to the QLM hopping term proceeds by constructing the quasidegenerate effective Hamiltonian to second order [419] and includes a combination of exchange terms $V_{i,j}$ and state-dependent energies $\epsilon_{i,\alpha}$. In the following two sections we describe the details of this mapping for $S = 1/2$ and $S = 1$.

9.5 Realization and Tests of $S = 1/2$ QLM

Here we describe details specific to the mapping for the case $S = 1/2$ and we numerically validate this mapping. In this mapping, the molecular state $|c\rangle$ is kept energetically decoupled from gauge-invariant initial states, which can be achieved by means of a small dc electric field. Furthermore, local light shifts can be used to decouple the molecular state $|d\rangle$ from dynamics at the site positions. On the links, while the molecular state $|a\rangle$ does not map directly to anything in the QLM Hilbert space, it is utilized to help mediate the second-order process needed to describe the fermion hopping interaction in the QLM.

A three-body hopping process is illustrated in Fig. 9.1 for the example: $|a\rangle_{S_x}|d\rangle_{L_x}|b\rangle_{S_{x+1}} \to |b\rangle_{S_x}|b\rangle_{L_x}|a\rangle_{S_{x+1}}$. In the dipole system this is a second-order process, which can proceed via

$$|a\rangle_{S_x}|d\rangle_{L_x}|b\rangle_{S_{x+1}} \xrightarrow{\text{virtual}} |b\rangle_{S_x}|a\rangle_{L_x}|b\rangle_{S_{x+1}} \xrightarrow{\text{virtual}} |b\rangle_{S_x}|b\rangle_{L_x}|a\rangle_{S_{x+1}}\,, \tag{9.3}$$

where $|b\rangle_{S_x}|a\rangle_{L_x}|b\rangle_{S_{x+1}}$ is an intermediate state outside the physical Hilbert space with an energy difference $\Delta \gg V$.

The second-order hopping process has a term given by

$$-w = \frac{1}{2} V^{b,a;a,d}_{S_x,L_x} V^{b,a;a,b}_{L_x,S_{x+1}} \left[\frac{1}{\Delta\epsilon_{1,x}} + \frac{1}{\Delta\epsilon_{2,x}} \right] , \tag{9.4}$$

with $\Delta\epsilon_{1,x} = \epsilon_{S_x,a} + \epsilon_{L_x,d} - \epsilon_{S_x,b} - \epsilon_{L_x,a}$ and $\Delta\epsilon_{2,x} = \epsilon_{L_x,b} + \epsilon_{S_{x+1},a} - \epsilon_{L_x,a} - \epsilon_{S_{x+1},b}$. This yields one equation for every x. While the QLM has a two-site unit cell that repeats, the microscopic parameters of the underlying DMH can be varied slightly between x and $x+2$, e.g., to allow for the mitigation of undesired processes resulting from the long-ranged dipolar interactions. Given a target w value, we can then simultaneously solve for all these equations up to corrections of order greater than $O(V^2/\Delta)$, which generates relationships between the various V and ϵ values. These parameters are chosen so as to preserve Gauss's law, and furthermore so that they are physically reasonable. Finally, in addition to the kinetic terms that appear in the second-order effective model, second-order self-interaction terms appear as well. These small diagonal terms can be fully compensated by a slight renormalization of the DMH energy terms $\epsilon_{i,\alpha}$ (see Tables IV and V in Supplementary Materials of Ref. [16]).

We numerically confirm the mapping for $S = 1/2$ by simulating the QLM on three unit cells with open boundary conditions, and comparing to the full simulation of the DMH. Figures 9.2(a)-9.2(c) show the DMH dynamics of the sites and links, for an initial product state configuration with staggered site occupations and polarized electric fields. Note that the electric flux energy is always a constant in the $S = 1/2$ QLM, and therefore g^2 is irrelevant. For a small mass, $m = 0.1w$, the dynamics of the DMH reveal a string inversion of the electric fluxes [Figs. 9.2(a) and 9.2(c)]. For a large mass, $m = 2.0w$ (more than three times the reported critical mass $m_c = 0.655w$ [420]), the system shows little dynamics, as there is almost a static flux string with small fluctuations [Figs. 9.2(b) and 9.2(c)].

We additionally compute the wave-function fidelity and violation of Gauss's law, shown in Fig. 9.2(d). The fidelity is defined as $|\langle\psi_{\mathrm{QLM}}|\psi_{\mathrm{DMH}}\rangle|^2$, where ψ_{DMH} is the wave-function from the DMH and ψ_{QLM} is the wave-function mapped from the QLM into the DMH space. The fidelity of our scheme is greater than 0.9 out to $t = 20w^{-1}$, and Gauss's law is preserved over this time range at the 10^{-6} level. For the molecule NaRb (body-frame dipole moment of 3.3 D) and a minimum separation of 0.5 μm, these calculations relate to a hopping rate of $w/h = 41.3$ Hz (with h Planck's constant). This robust energy scale should be compatible with long molecule trapping times [421] and coherence times [401, 402]. In addition, larger values of the hopping can be achieved by reducing the scale of imposed energy penalties Δ, albeit with fidelities lower than those shown in Fig. 9.2(d).

9.6 Realization and Tests of $S = 1$ QLM

The $S = 1$ case is realized similarly to $S = 1/2$; however more internal levels are used to represent the larger number of link spin values. The link states are represented by the various $N = 1$ rotational sublevels as in Table 9.1. For the sites, the $|b\rangle$ and $|d\rangle$ levels are decoupled from the dynamics by large local light shifts. Unlike for $S = 1/2$, in the $S = 1$ case second-order self-interaction processes, such as $|c\rangle_{S_x} |b\rangle_{L_x} |a\rangle_{S_{x+1}} \xrightarrow{\text{virtual}} |c\rangle_{S_x} |a\rangle_{L_x} |c\rangle_{S_{x+1}} \xrightarrow{\text{virtual}} |c\rangle_{S_x} |b\rangle_{L_x} |a\rangle_{S_{x+1}}$, cannot be entirely removed through coordination of the $\epsilon_{i,\alpha}$ terms. This comes from the fact that they are not simply renormalized one-body terms. For example, the above process generates an $O(V^2/\Delta)$ term of the form $b^\dagger_{L_x,b} b_{L_x,b} b^\dagger_{S_{x+1},a} b_{S_{x+1},a}$ that is not in the QLM. Nevertheless, these additional terms are diagonal in the molecular $\{|N, m_N\rangle\}$ basis and still preserve Gauss' law, but cannot be removed because the simultaneous constraints for w and setting these "extra" terms to zero results in an overdetermined system of equations. To overcome this, we introduce a new length scale.

For every x, we set the distances between molecules representing S_x and L_x to be small and denote the characteristic scale of the dipole-dipole interaction between them as V_{short}. Meanwhile, the distances between molecules representing L_x and S_{x+1} are chosen to be larger with a characteristic scale for the dipole-dipole interaction between them of V_{long}, and $V_{\text{long}} \ll V_{\text{short}}$. The hopping parameter w is $O(V_{\text{short}} V_{\text{long}}/\Delta)$. The second-order self-interaction term for S_x and L_x is $O(V^2_{\text{short}}/\Delta)$ and that for L_x and S_{x+1} is $O(V^2_{\text{long}}/\Delta)$, which can be safely neglected. Therefore, we only need to consider the equations between S_x and L_x, which decrease the total number of equations and leave them underdetermined. A nonunique solution to these equations can then be found, and we can obtain experimental parameters such that these second-order terms are made small or can be removed from the effective Hamiltonian. To note, for larger $V_{\text{short}}/V_{\text{long}}$ ratios, higher-order terms eventually limit this minimization of the extra terms. We set the intermolecular distances as $r_{S_{2n+1},L_{2n+1}} = r_{S_1,L_1}$, $r_{L_{2n+1},S_{2n+2}} = \gamma r_{S_1,L_1}$, $r_{S_{2n+2},L_{2n+2}} = \beta r_{S_1,L_1}$, and $r_{L_{2n+2};S_{2n+3}} = \beta\gamma r_{S_1,L_1}$ for every n, where n labels the unit cell. γ is defined as a variable long-short distance ratio greater than or equal to one. β is fixed by energy conditions and independent of γ.

To benchmark our scheme for $S = 1$, we perform exact diagonalization on three unit cells with open boundary conditions. The initial configuration is chosen as shown in Fig. 9.3, relating to a flux string connecting static charges. String breaking is a key dynamical phenomenon in high-energy physics, found in QCD [423] as well as the simpler Schwinger model [424]. Depending on the parameters of the $S = 1$ QLM, the initial configuration as a string may break into approximate vacuum in the middle region, resulting in the production of two mesons on the edge. To verify that our scheme reflects the physical dynamics properly, we investigate both a small-mass scenario [Fig. 9.3(a), $\{m = 0.25 \times \sqrt{2}w, g^2 = \sqrt{2}w\}$] and a large-mass scenario

[Fig. 9.3(b), $\{m = 2.0 \times \sqrt{2}w, g^2 = \sqrt{2}w\}$], which should result in string breaking and a stabilized string, respectively. Figure 9.3 shows that the dynamics of the dipolar molecular system (solid lines) reflects this behavior. We find good agreement with the expected dynamics of the target QLM (dashed line). Specifically, for large mass, the string stays approximately in its initial configuration up to small fluctuations, while the string of the small mass case breaks on the hopping timescale. These results are consistent with the estimated critical length $L_c = 4m/g^2 + 3$ [403].

Gauss's law is preserved to high accuracy [see Fig. 9.3(d)]; however, the fidelity of the DMH "simulator" drops from 1.0 to roughly 0.5 over 10–20 time units [in terms of $(\sqrt{2}w)^{-1}$]. This infidelity comes almost entirely from the additional gauge-invariant self-interaction terms that do not appear in the QLM. These terms slightly modify the frequencies of oscillations in the DMH and QLM dynamics, and thus have a large influence on the fidelity at long times, but otherwise do not alter the expected QLM phenomenology [see Fig. 9.3(c)].

While one approach to improving this fidelity is simply adding the additional gauge-invariant terms to the target QLM (as is done in Ref. [403]), we can suppress the additional gauge invariant terms and improve the fidelity by adjusting the "long-short" distance ratios. In this work we use the value of $V_{\text{short}}/V_{\text{long}} = 1.5^3 \approx 3.4$ (see Fig. 9.3), which represents a nonoptimal but more experimentally realistic compromise; the parameters used in Fig. 9.3 already relate to hopping energies of $\sqrt{2}w/h = 3.2$ Hz for NaRb (3.3 D) and an assumed minimum spacing of 0.5 μm.

We have investigated the effect of the long-short distance ratio γ on fidelity. Figures 9.4 and 9.5 demonstrate the average fidelity up to time $t = 20(\sqrt{2}w)^{-1}$ at various long-short distance ratios for $m = 0.25 \times \sqrt{2}w$ and $m = 2.0 \times \sqrt{2}w$. It is shown that the fidelity is low for both small and large γ, while the optimal value is achieved at an intermediate ratio between 2 and 3. For small long-short distance ratios, certain second-order self-interactions at the order $O(V_{\text{long}}^2/\Delta)$ are not sufficiently suppressed, which results in a low fidelity. For large long-short distance ratios, higher-order self-interaction terms [such as fourth-order self-interactions, i.e., at the order $O(V_{\text{short}}^4/\Delta^3)$, for the pair of molecules S_x and L_x separated by a short distance] may become comparable to the hopping parameter $w \sim O(V_{\text{short}}V_{\text{long}}/\Delta)$ and are thus not negligible. Therefore, the optimal choice of γ for fidelity should balance the trade-off between suppressing second-order self-interaction terms at the order $O(V_{\text{long}}^2/\Delta)$ and avoiding high-order self-interactions, which is attained in the intermediate ratio. The effect of long-short distance ratios is further presented in Figs. 9.5(a) and 9.5(b) for real-time fidelity changes.

The long-short distance ratios that provide the highest fidelity (as compared to the target QLM) may be less ideal from a practical perspective, as they result in lower hopping parameters w for a fixed minimum

separation of the molecules. However, the reduction of the fidelity due to the extra gauge-invariant terms in the $S = 1$ case does not necessarily preclude the DMH dynamics from displaying the physical processes of interest. In our case, at a suboptimal long-short distance ratio, even though the fidelity may not be very high, the observed phenomenology is not significantly altered by the extra gauge invariant terms. In Figs. 9.5(c) and 9.5(d), we compare the sum of electric fluxes, an indicator of string breaking, over various long-short distance ratios. Although different long-short distance ratios have very different fidelities, they result in similar string breaking phenomena except for $\gamma = 1.0$. Qualitatively, at all of the ratios except for $\gamma = 1.0$, our results at $m = 0.25 \times \sqrt{2}w$ reveal string breaking while the results at $m = 2.0 \times \sqrt{2}w$ do not. This provides some evidence that we may be able to choose suboptimal (in terms of fidelity with respect to the ideal QLM) but experimentally favorable parameters to explore the physics of interest through analog simulation.

9.7 Conclusions and Outlook

In this paper, we propose an approach for simulating quantum link models based on the restricted internal-state dynamics of fixed dipolar spins. Our numerical tests of simple U(1) LGTs in $1 + 1$ dimensions show that this approach enables the experimental exploration of important dynamical phenomena such as string inversion and string breaking. Further directions are suggested by the present work, including extensions to realizations of higher S values, non-Abelian LGTs, quantum mechanical θ angles [425], and QLM dynamics in higher dimensions [426]. Moreover, while our scheme based on dipolar spin interactions is of relevance to a range of systems, such as cold molecules and Rydberg atom arrays, the potential reach could be greatly broadened by generalizing this framework to generic spin systems, as realized by most physical quantum information platforms.

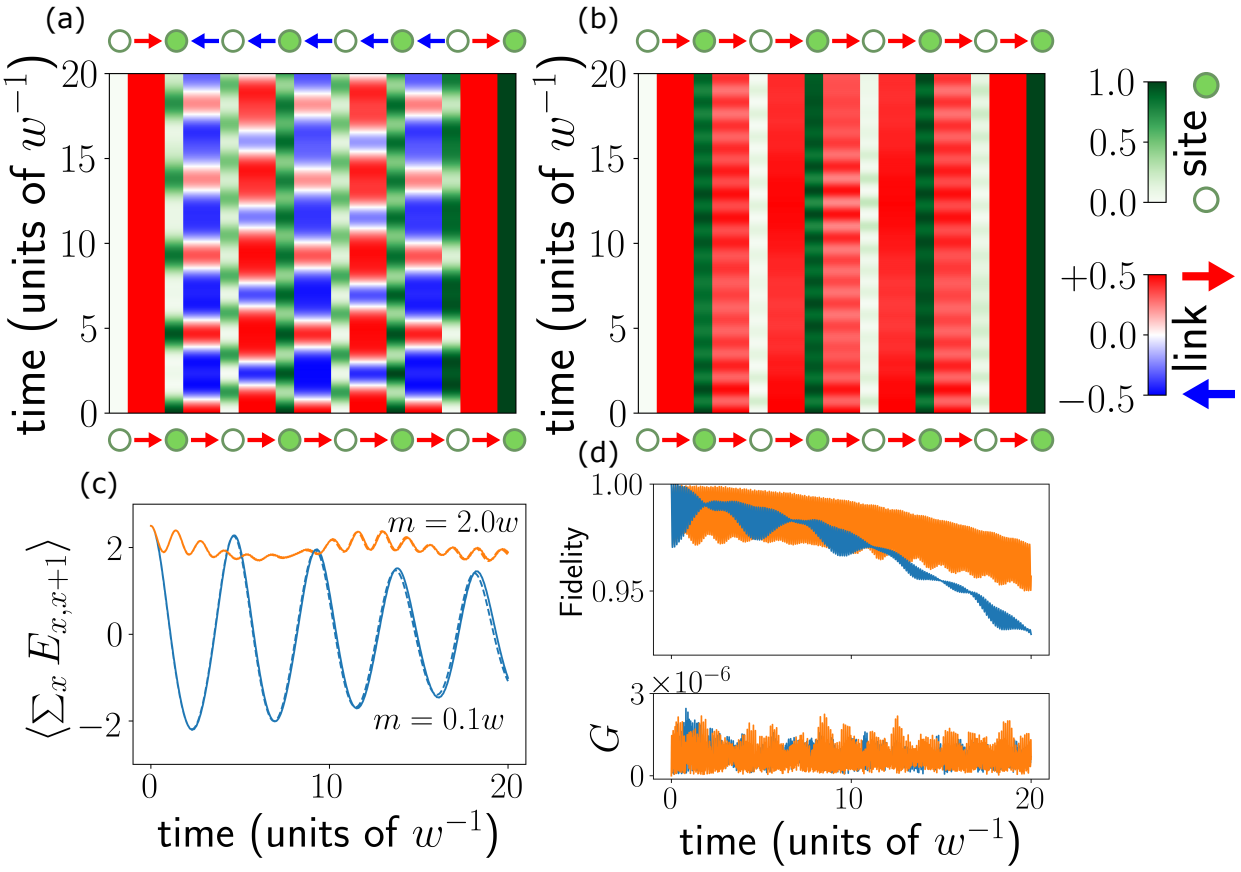


Figure 9.2: Real-time evolution of densities of sites and links in the dipolar molecular system to simulate the $S = 1/2$ QLM on three unit cells. The dynamics for initialized strings of right-pointing electric fields are shown for the cases of small mass, (a) $m = 0.1w$, and large mass, (b) $m = 2.0w$. Time is in units of the inverse hopping, w^{-1}. For small mass (a), the electric field of the string undergoes large-scale oscillations. For large mass (b), the string stays roughly fixed, with only small fluctuations of the charge densities and link spins. To note for both (a) and (b), the outermost sites and links are fixed because of the open boundary conditions. (c) Electric fluxes summed over all dynamical links for $m = 0.1w$ (blue) and $m = 2.0w$ (orange). Solid and dashed lines relate to the DMH and QLM dynamics, respectively. (d) Top: Fidelity of the dipolar molecular wavefunction versus the QLM wavefunction. Bottom: The effective gauge invariance parameter $G \equiv \sum_x |\langle \tilde{G}_x \rangle|/L$ [403] at the two mass values shown in (a) and (b).

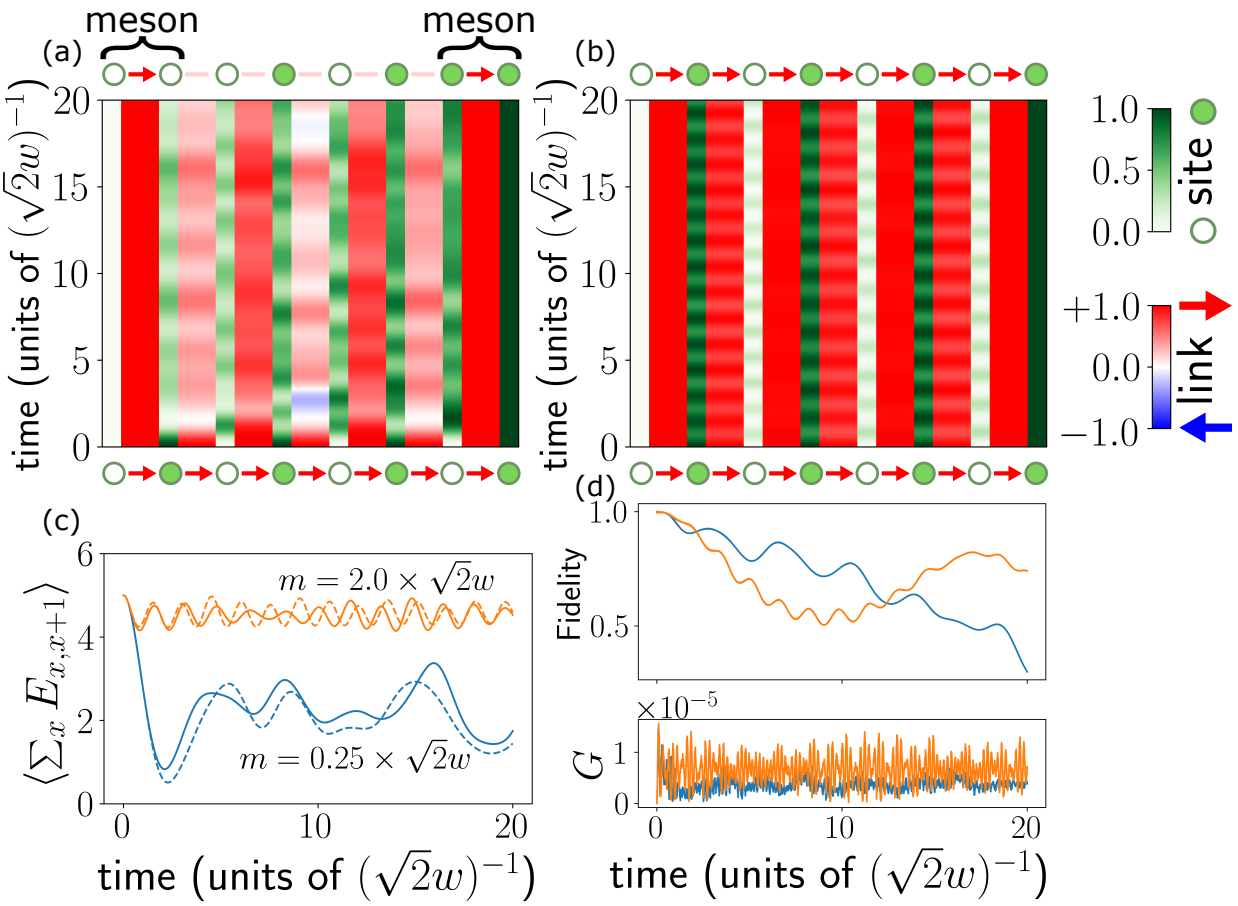


Figure 9.3: Real-time evolution of fermions and links in the dipolar molecular system to simulate the $S = 1$ QLM on three unit cells with $g^2 = \sqrt{2}w$ starting from a string of right-pointing electric fields at (a) $m = 0.25 \times \sqrt{2}w$ and (b) $m = 2.0 \times \sqrt{2}w$. Time is in units of $(\sqrt{2}w)^{-1}$. With a small mass (a), the string breaks (modulo finite size effects [422]) reaching values close to zero on the hopping timescale, resulting in two approximate mesons on the edges and approximate vacuum in between. With a large mass (b), the string approximately remains, with small fluctuations in densities. In (a) and (b), the densities of the two sites and two links on the edges are fixed due to the open boundary condition. (c) Electric fluxes summed over all dynamical links for both $m = 0.25 \times \sqrt{2}w$ (blue) and $m = 2.0 \times \sqrt{2}w$ (orange). Solid and dashed lines relate to the DMH and QLM dynamics, respectively. (d) Top: Fidelity of the dipolar molecular wavefunction versus the QLM wavefunction. Bottom: The effective gauge invariance parameter $G \equiv \sum_x |\langle \tilde{G}_x \rangle|/L$ [403] at two values of masses in (a) and (b).

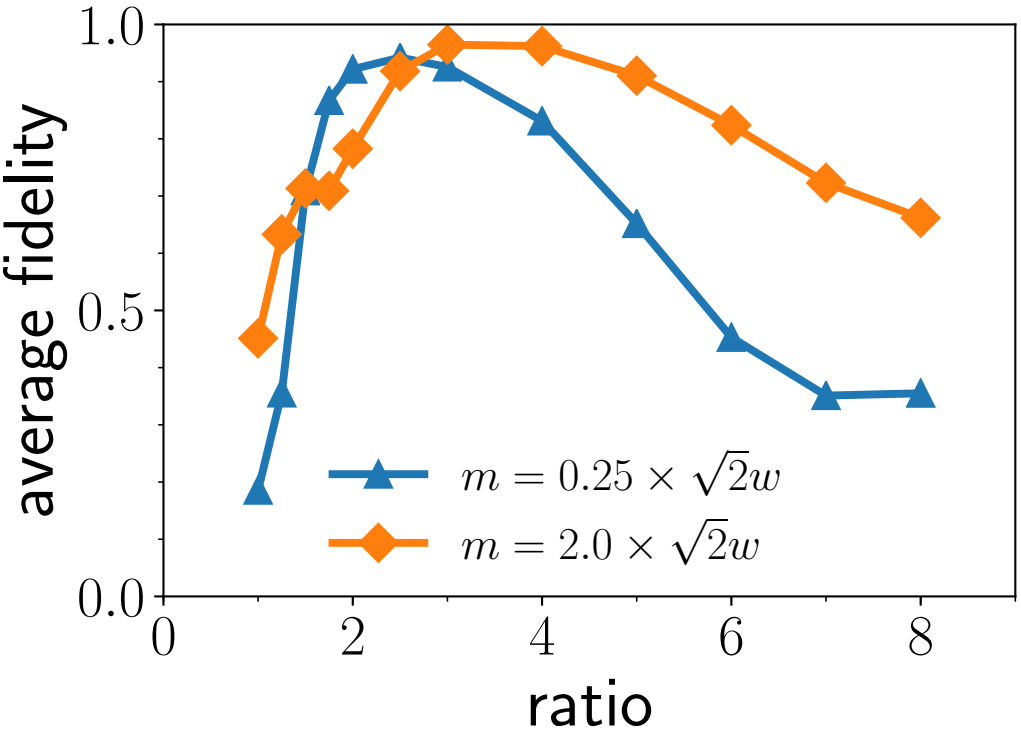


Figure 9.4: Average fidelity of the calculated DMH dynamics based on the mapping to the $S = 1$ QLM, plotted versus the "long-short" distance ratio γ used to mitigate the influence of "extra" gauge-invariant terms. The plotted curves are for mass values of $m = 0.25 \times \sqrt{2}w$ and $m = 2.0 \times \sqrt{2}w$. These represent the fidelity when starting from the initial product state as in Fig. 9.3, comparing the DMH-evolved state to the evolution under the ideal QLM, with averaging over the time period $0 \leq t \leq 20$ [with time in units of $(\sqrt{2}w)^{-1}$]. An initial rise in fidelity is seen for moderate ratios as the "extra" gauge invariant terms are suppressed, but it decreases for larger ratios as higher-order terms become important.

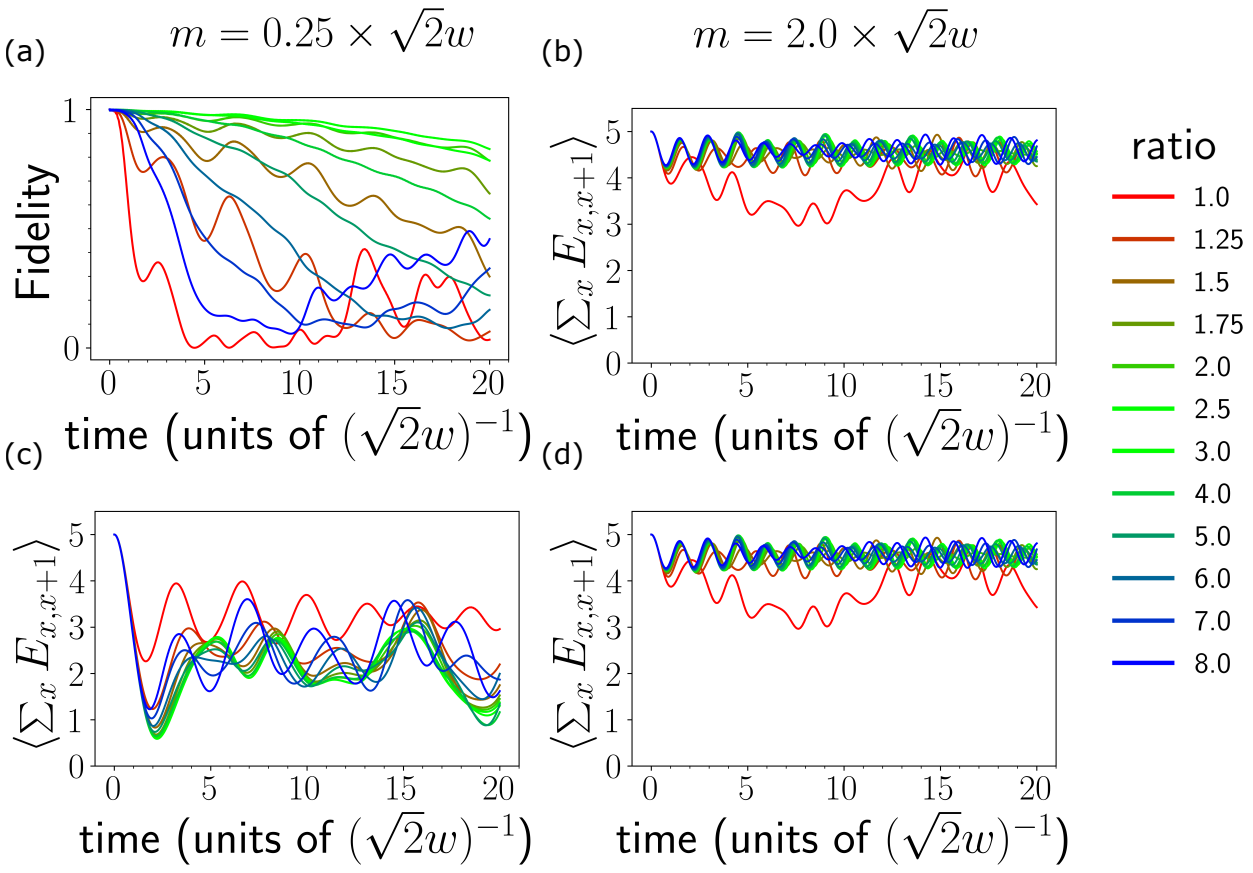


Figure 9.5: Effect of additional gauge-invariant terms generated by the dipolar molecular system for simulation of the $S = 1$ QLM. This figure considers the influence of these terms on the fidelity (when comparing the DMH simulations to the ideal QLM) and the dynamics of key physical observables. Fidelity versus time for (a) $m = 0.25 \times \sqrt{2}w$ and (b) $m = 2.0 \times \sqrt{2}w$ with various long-short distance ratios. The sum of electric fields for (c) $m = 0.25 \times \sqrt{2}w$ and (d) $m = 2.0 \times \sqrt{2}w$ with various long-short distance ratios. For all panels, time is in units of $(\sqrt{2}w)^{-1}$. While the fidelity (as compared to the ideal QLM) is strongly dependent on the ratio value, the physical observations of string breaking in (c) and a fixed string with small fluctuations in (d) are robust over a very large range of ratio values.

Chapter 10

Protocol Discovery for the Quantum Control of Majoranas by Differentiable Programming and Natural Evolution Strategies

10.1 Summary

This chapter is a summary based on the research in Ref. [427]. We study the quantum control protocol for transporting Majorana zero modes in the 1D Kitaev Chain [428] as follows

$$\begin{aligned}\mathcal{H}(t) = &-\sum_{x=1}^{N}[\mu(x) - V(x,t)](c_x^\dagger c_x - 1/2) \\ &- t\sum_{x=1}^{N-1}(c_x^\dagger c_{x+1} + h.c) + \Delta\sum_{x=1}^{N-1}(c_x^\dagger c_{x+1}^\dagger + h.c.),\end{aligned} \tag{10.1}$$

where $c_x^{(\dagger)}$ are fermionic annihilation (creation) operators, Δ is the p-wave superconducting gap, $\mu(x)$ is the onsite chemical potential and t is the hopping amplitude. In particular, the time-dependent external potential $V(x,t)$ is parameterized by

$$V(x,t) = V_{\text{height}}[f(x - x_L(t)) + f(x_R - x)] \tag{10.2}$$

where V_{height} is the maximum height of the outer potential walls, $x_{L(R)}$ are the left(right) wall positions and $f(x) = 1/(1+\exp(x/\sigma))$ is a sigmoid function. Our goal is to search for protocols to move $x_L(t)$ such that after time T the infidelity $\mathcal{I}(T)$ is minimized

$$\mathcal{I}(T) = 1 - |\langle\psi_B|\mathcal{T}e^{-i\int_0^T \mathcal{H}(t)dt}|\psi_A\rangle|^2 \tag{10.3}$$

where ψ_A and ψ_B are the ground state for the initial place and the final place.

To search for the quantum control protocol, we develop both the differentiable programming approach and the natural evolutionary strategy approach. The high level idea of the differentiable programming

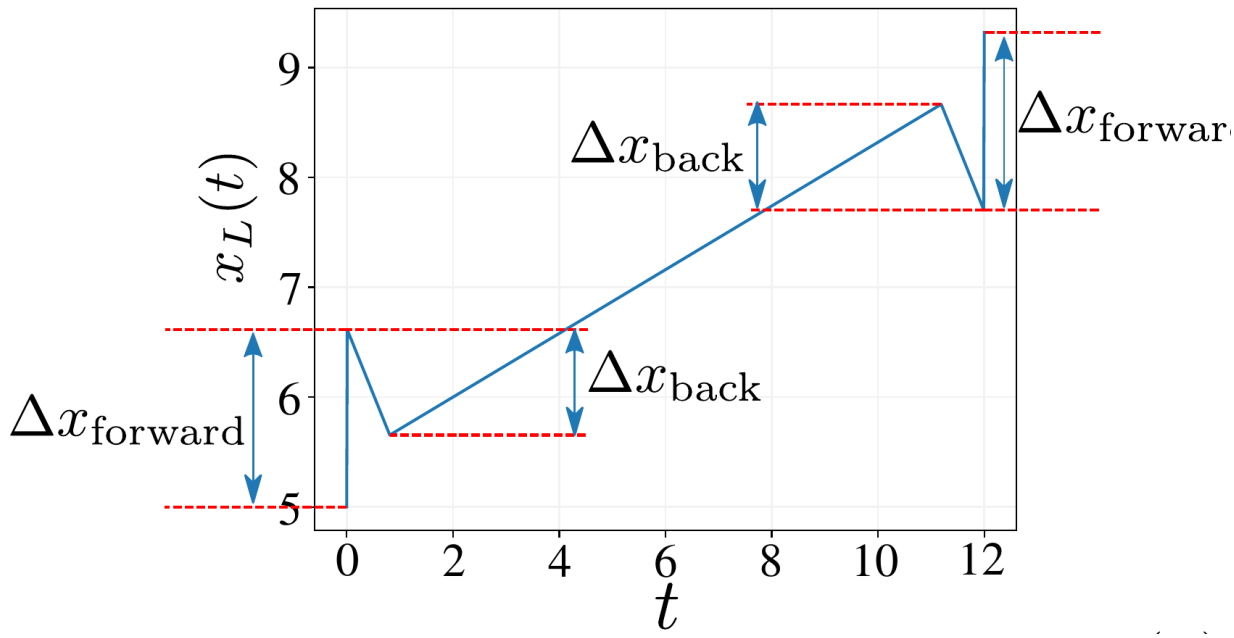


Figure 10.1: Jump move jump protocol (reproduced from Ref. [427]).

approach is to compute $\frac{d\mathcal{I}(T)}{dx_L(t)}$ through automatic differentiation with the Onishi Formula and update $x_L(t) \to x_L(t) - \nabla_{x_L(t)} I(T)$ by gradient descent. For the natural evolutionary strategy approach, we choose p_θ to be a Gaussian distribution as $\mathcal{N}(\theta, \sigma^2 I)$ over an ensemble of protocols $x_L(t)$. We compute the gradient as follows and update p_θ by gradient descent.

$$\nabla_\theta \mathbb{E}_{\phi \sim \mathcal{N}(\theta, \sigma^2 I)} \left[\mathcal{I}(T, \phi)\right] = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} \left[\mathcal{I}(T, \theta + \sigma\epsilon)\epsilon/\sigma\right]. \tag{10.4}$$

Both the differentiable programming and the natural evolutionary strategies approaches have discovered a new efficient jump-move-jump (JMJ) protocol shown in Fig. 10.1. We have compared the differentiable programming (DP), the natural evolutionary strategies (NES) and the simulated annealing (SA) in four different Majorana motion regimes, which is shown in Fig. 10.2. More details and theoretical analysis of the new protocol can be found in Ref. [427].

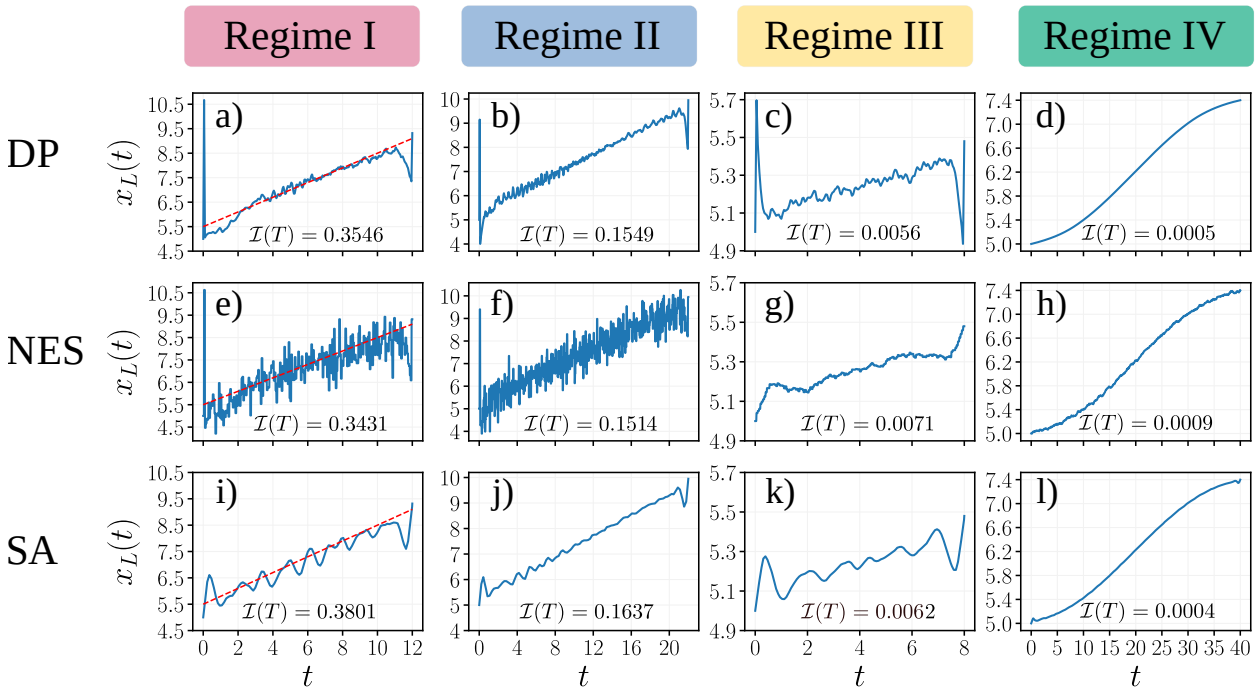


Figure 10.2: Overview of the optimal controls in the four different Majorana motion regimes (reproduced from Ref. [427]).

Chapter 11

Shadow Quantum Process Tomography on Near-term Quantum Computers

11.1 Summary

This chapter is a summary based on ongoing research with Ryan Levy and Bryan K. Clark. In this work, we have developed shadow tomography algorithms for quantum process tomography. We have further implemented the algorithms on IonQ quantum computer for both unitary and non-unitary quantum process up to 4 qubits.

The key idea of the approach is based on the recent development of shadow tomography on quantum state [13] and the the Jamiołkowski process-state isomorphism [429]. An important result in classical shadow tomography states that [13]: Classical shadows of size N suffice to predict M linear functions $\{tr(O_i\rho)\}_{i=1,\ldots,M}$ up to additive error ϵ given that $N \geq$ (order) $\log(M)\max_i ||O_i||^2_{shadow}/\epsilon^2$.

To connect quantum process tomography with the quantum state tomography, we utilize the Jamiołkowski process-state isomorphism [429]. It provides a positive semi-definite operator representation Λ_ε for a n qubits process ε, which is the Choi matrix representation [430]

$$\Lambda_\varepsilon = (\mathbb{1} \otimes \varepsilon)(\left|\phi^+\right\rangle \left\langle\phi^+\right|^{\otimes n}) \tag{11.1}$$

where $|\phi^+\rangle = |00\rangle + |11\rangle$.

Based on the classical shadow tomography property and the Choi matrix representation, we state the following theorem for classical shadow tomography for quantum process.

Theorem 11.1.1 *For a n-qubit quantum process Choi matrix Λ and $\epsilon, \delta \in [0,1]$, the number of measurements N that suffices to predict any reduced k-qubit process Choi matrix $\Lambda^{(k)}$ such that each Pauli basis error up to ϵ and probability $1-\delta$ is*

$$N = \frac{68}{\epsilon^2}\log(2d^4/\delta) max_i ||O_i^{(2k)} - \frac{\mathrm{tr}(O_i^{(2k)})}{2}\mathbb{1}||^2_{shadow} \tag{11.2}$$

where $O_i^{(2k)}$ is any $2k$-qubit Pauli observable and $d = 2^k$.

In particular, for random global Clifford measurement, $N = \frac{205}{\epsilon^2} d^2 \log(2d^4/\delta)$ *and for random Pauli measurement,* $N = \frac{68}{\epsilon^2} 9^k \log(2d^4/\delta) max_i ||O_i^{(2k)}||_\infty^2$.

Proof According to the Jamiołkowski process-state isomorphism [429], a quantum process on a n-qubit system can be represented by a density matrix ρ on a $2n$-qubit system. Here we consider normalized Choi matrix with $\mathrm{tr}\rho = 1$. Any reduced k-qubit process matrix ρ_k could be expressed as

$$\rho_k = \frac{1}{4^k} \sum^{16^k} \alpha_i O_i^{(2k)} \tag{11.3}$$

where $\alpha_i = \mathrm{tr}(\rho_k O_i^{(2k)}) = \mathrm{tr}(\rho O_i^{(2k)} \otimes \mathbb{1}_{2n-2k})$.

Notice that there are $M = d^4 = 16^k$ number of Pauli basis $O_i^{(2k)}$ as observables to measure, which leads to the contribution of $\log(2M/\delta) = \log(2d^4/\delta)$. For random global Clifford measurement, $||O - \frac{\mathrm{tr}(O)}{2}\mathbb{1}||_{shadow}^2 \leq 3\mathrm{tr}(O^2) = 3d^2$ for $O = O_i^{(2k)}$. For random Pauli measurement, $||O - \frac{\mathrm{tr}(O)}{2}\mathbb{1}||_{shadow}^2 \leq 9^k ||O||_\infty^2$ with $O = O_i^{(2k)}$ in this case. ∎

Chapter 12

References

[1] A. Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[3] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[4] Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, May 2017.

[5] Yi Zhang, A. Mesaros, K. Fujita, S. D. Edkins, M. H. Hamidian, K. Ch'ng, H. Eisaki, S. Uchida, J. C. Séamus Davis, Ehsan Khatami, and Eun-Ah Kim. Machine learning in electronic-quantum-matter imaging experiments. *Nature*, page 1, June 2019.

[6] Jan Hermann, Zeno Schätzle, and Frank Noé. Deep neural network solution of the electronic schrödinger equation, 2019.

[7] David Pfau, James S. Spencer, Alexander G. de G. Matthews, and W. M. C. Foulkes. Ab-initio solution of the many-electron schrödinger equation with deep neural networks, 2019.

[8] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355:602–606, 2017.

[9] Gurtej Kanwar, Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Equivariant flow-based sampling for lattice gauge theory. *Physical Review Letters*, 125(12), Sep 2020.

[10] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, Jan 2015.

[11] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines*. *Cognitive Science*, 9(1):147–169, 1985.

[12] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[13] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, Oct 2020.

[14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[15] Fengcheng Wu, Youjin Deng, and Nikolay Prokof'ev. Phase diagram of the toric code model in a parallel magnetic field. *Physical Review B*, 85(19):195104, 2012.

[16] Di Luo, Jiayu Shen, Michael Highman, Bryan K. Clark, Brian DeMarco, Aida X. El-Khadra, and Bryce Gadway. Framework for simulating gauge theories with dipolar spin systems. *Physical Review A*, 102(3), Sep 2020.

[17] Di Luo and Bryan K. Clark. Backflow transformations via neural networks for quantum many-body wave functions. *Physical Review Letters*, 122(22), Jun 2019.

[18] R. P. Feynman and Michael Cohen. Energy spectrum of the excitations in liquid helium. *Phys. Rev.*, 102:1189–1204, Jun 1956.

[19] Luca F. Tocchio, Federico Becca, Alberto Parola, and Sandro Sorella. Role of backflow correlations for the nonmagnetic phase of the t–$t^{'}$ hubbard model. *Phys. Rev. B*, 78:041101, Jul 2008.

[20] J. C. Slater. The theory of complex spectra. *Phys. Rev.*, 34:1293–1322, Nov 1929.

[21] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. Theory of superconductivity. *Phys. Rev.*, 108:1175–1204, Dec 1957.

[22] Robert Jastrow. Many-body problem with strong forces. *Phys. Rev.*, 98:1479–1484, Jun 1955.

[23] Lubo š Mitáš and Richard M. Martin. Quantum monte carlo of nitrogen: Atom, dimer, atomic, and molecular solids. *Phys. Rev. Lett.*, 72:2438–2441, Apr 1994.

[24] Yongkyung Kwon, D. M. Ceperley, and Richard M. Martin. Effects of three-body and backflow correlations in the two-dimensional electron gas. *Phys. Rev. B*, 48:12037–12046, Oct 1993.

[25] Michele Taddei, Michele Ruggeri, Saverio Moroni, and Markus Holzmann. Iterative backflow renormalization procedure for many-body ground-state wave functions of strongly interacting normal fermi liquids. *Phys. Rev. B*, 91:115106, Mar 2015.

[26] Michele Ruggeri, Saverio Moroni, and Markus Holzmann. Nonlinear network description for many-body quantum systems in continuous space. *Phys. Rev. Lett.*, 120:205302, May 2018.

[27] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992.

[28] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91:147902, Oct 2003.

[29] G. Vidal. Entanglement renormalization. *Phys. Rev. Lett.*, 99:220405, Nov 2007.

[30] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac. Criticality, the area law, and the computational power of projected entangled pair states. *Phys. Rev. Lett.*, 96:220601, Jun 2006.

[31] David A. Huse and Veit Elser. Simple variational wave functions for two-dimensional heisenberg spin-$\frac{1}{2}$ antiferromagnets. *Phys. Rev. Lett.*, 60:2531–2534, Jun 1988.

[32] F Mezzacapo, N Schuch, M Boninsegni, and J I Cirac. Ground-state properties of quantum many-body systems: entangled-plaquette states and variational monte carlo. *New Journal of Physics*, 11(8):083026, 2009.

[33] Konrad H Marti, Bela Bauer, Markus Reiher, Matthias Troyer, and Frank Verstraete. Complete-graph tensor network states: a new fermionic wave function ansatz for molecules. *New Journal of Physics*, 12(10):103008, 2010.

[34] Norbert Schuch, Michael M. Wolf, Frank Verstraete, and J. Ignacio Cirac. Simulation of quantum many-body systems with strings of operators and monte carlo tensor contractions. *Phys. Rev. Lett.*, 100:040501, Jan 2008.

[35] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[36] Xun Gao and Lu-Ming Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature Communications*, 8(1):662, 2017.

[37] Dong-Ling Deng, Xiaopeng Li, and S. Das Sarma. Quantum entanglement in neural network states. *Phys. Rev. X*, 7:021021, May 2017.

[38] Ivan Glasser, Nicola Pancotti, Moritz August, Ivan D. Rodriguez, and J. Ignacio Cirac. Neural-network quantum states, string-bond states, and chiral topological states. *Phys. Rev. X*, 8:011006, Jan 2018.

[39] Jing Chen, Song Cheng, Haidong Xie, Lei Wang, and Tao Xiang. Equivalence of restricted boltzmann machines and tensor network states. *Phys. Rev. B*, 97:085104, Feb 2018.

[40] Yusuke Nomura, Andrew S. Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Phys. Rev. B*, 96:205152, Nov 2017.

[41] G. Carleo, Y. Nomura, and M. Imada. Constructing exact representations of quantum many-body systems with deep neural networks. *ArXiv e-prints*, February 2018.

[42] N. Freitas, G. Morigi, and V. Dunjko. Neural Network Operations and Susuki-Trotter evolution of Neural Network States. *ArXiv e-prints*, March 2018.

[43] Zi Cai and Jinguo Liu. Approximating quantum many-body wave functions using artificial neural networks. *Phys. Rev. B*, 97:035116, Jan 2018.

[44] K. Choo, G. Carleo, N. Regnault, and T. Neupert. Symmetries and many-body excited states with neural-network quantum states. *ArXiv e-prints*, July 2018.

[45] J. Han, L. Zhang, and W. E. Solving Many-Electron Schr\"odinger Equation Using Deep Neural Networks. *ArXiv e-prints*, July 2018.

[46] Andrea Rocchetto, Edward Grant, Sergii Strelchuk, Giuseppe Carleo, and Simone Severini. Learning hard quantum distributions with variational autoencoders. *npj Quantum Information*, 4(1):28, 2018.

[47] H. Saito. Method to Solve Quantum Few-Body Problems with Artificial Neural Networks. *Journal of the Physical Society of Japan*, 87(7):074002, July 2018.

[48] Hiroki Saito and Masaya Kato. Machine learning technique to find quantum many-body ground states of bosons on a lattice. *Journal of the Physical Society of Japan*, 87(1):014001, 2018.

[49] Y. Huang and J. E. Moore. Neural network representation of tensor network and chiral states. *ArXiv e-prints*, January 2017.

[50] Dong-Ling Deng, Xiaopeng Li, and S. Das Sarma. Machine learning topological states. *Phys. Rev. B*, 96:195145, Nov 2017.

[51] Jing Chen, Song Cheng, Haidong Xie, Lei Wang, and Tao Xiang. Equivalence of restricted boltzmann machines and tensor network states. *Phys. Rev. B*, 97:085104, Feb 2018.

[52] Stephen R Clark. Unifying neural-network quantum states and correlator product states via tensor networks. *Journal of Physics A: Mathematical and Theoretical*, 51(13):135301, 2018.

[53] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, 2018.

[54] Raphael Kaubruegger, Lorenzo Pastori, and Jan Carl Budich. Chiral topological phases from artificial neural networks. *Phys. Rev. B*, 97:195136, May 2018.

[55] X. Liang, W.-Y. Liu, P.-Z. Lin, G.-C. Guo, Y.-S. Zhang, and L. He. Solving frustrated quantum many-particle models with convolutional neural networks. *ArXiv e-prints*, July 2018.

[56] B. Jónsson, B. Bauer, and G. Carleo. Neural-network states for the classical simulation of quantum computing. *ArXiv e-prints*, August 2018.

[57] L. Pastori, R. Kaubruegger, and J. C. Budich. Generalized Transfer Matrix States from Artificial Neural Networks. *ArXiv e-prints*, August 2018.

[58] Sirui Lu, Xun Gao, and L. M. Duan. Efficient Representation of Topologically Ordered States with Restricted Boltzmann Machines. *arXiv e-prints*, page arXiv:1810.02352, October 2018.

[59] Dmitrii Kochkov and Bryan K. Clark. Variational optimization in the AI era: Computational Graph States and Supervised Wave-function Optimization. *arXiv e-prints*, page arXiv:1811.12423, November 2018.

[60] M. Holzmann, D. M. Ceperley, C. Pierleoni, and K. Esler. Backflow correlations for the electron gas and metallic hydrogen. *Phys. Rev. E*, 68:046707, Oct 2003.

[61] Luca F. Tocchio, Federico Becca, and Claudius Gros. Backflow correlations in the hubbard model: An efficient tool for the study of the metal-insulator transition and the large-u limit. *Phys. Rev. B*, 83:195138, May 2011.

[62] Federico Becca, Luca F Tocchio, and Sandro Sorella. Metal-insulator transition and strong-coupling spin liquid in the t–t ' hubbard model. *Journal of Physics: Conference Series*, 145(1):012016, 2009.

[63] Giuseppe Carleo, Saverio Moroni, Federico Becca, and Stefano Baroni. Itinerant ferromagnetic phase of the hubbard model. *Phys. Rev. B*, 83:060411, Feb 2011.

[64] Luca F. Tocchio, Alberto Parola, Claudius Gros, and Federico Becca. Spin-liquid and magnetic phases in the anisotropic triangular lattice: The case of $\kappa-(\mathrm{ET})_2\mathrm{X}$. *Phys. Rev. B*, 80:064419, Aug 2009.

[65] N. D. Drummond, P. López Ríos, A. Ma, J. R. Trail, G. G. Spink, M. D. Towler, and R. J. Needs. Quantum monte carlo study of the ne atom and the ne+ ion. *The Journal of Chemical Physics*, 124(22):224104, 2006.

[66] Luca F. Tocchio, Federico Becca, and Claudius Gros. Interaction-induced fermi-surface renormalization in the t_1-t_2 hubbard model close to the mott-hubbard transition. *Phys. Rev. B*, 81:205109, May 2010.

[67] Yongkyung Kwon, D. M. Ceperley, and Richard M. Martin. Effects of backflow correlation in the three-dimensional electron gas: Quantum monte carlo study. *Phys. Rev. B*, 58:6800–6806, Sep 1998.

[68] P. López Ríos, A. Ma, N. D. Drummond, M. D. Towler, and R. J. Needs. Inhomogeneous backflow transformations in quantum monte carlo calculations. *Phys. Rev. E*, 74:066701, Dec 2006.

[69] Michael A. Lee, K. E. Schmidt, M. H. Kalos, and G. V. Chester. Green's function monte carlo method for liquid ^{3}He. *Phys. Rev. Lett.*, 46:728–731, Mar 1981.

[70] K. E. Schmidt, Michael A. Lee, M. H. Kalos, and G. V. Chester. Structure of the ground state of a fermion fluid. *Phys. Rev. Lett.*, 47:807–810, Sep 1981.

[71] R. M. Panoff and J. Carlson. Fermion monte carlo algorithms and liquid ^{3}He. *Phys. Rev. Lett.*, 62:1130–1133, Mar 1989.

[72] Markus Holzmann, Bernard Bernu, and D. M. Ceperley. Many-body wavefunctions for normal liquid ^{3}He. *Phys. Rev. B*, 74:104510, Sep 2006.

[73] E. Dagotto, A. Moreo, F. Ortolani, D. Poilblanc, and J. Riera. Static and dynamical properties of doped hubbard clusters. *Phys. Rev. B*, 45:10741–10760, May 1992.

[74] Bryan K. Clark Di Luo. See supplemental material at.

[75] Bo-Xiao Zheng, Chia-Min Chung, Philippe Corboz, Georg Ehlers, Ming-Pu Qin, Reinhard M. Noack, Hao Shi, Steven R. White, Shiwei Zhang, and Garnet Kin-Lic Chan. Stripe order in the underdoped region of the two-dimensional hubbard model. *Science*, 358(6367):1155–1160, 2017.

[76] B. Xu, N. Wang, T. Chen, and M. Li. Empirical Evaluation of Rectified Activations in Convolutional Network. *ArXiv e-prints*, May 2015.

[77] Jie Lou and Anders W. Sandvik. Variational ground states of two-dimensional antiferromagnets in the valence bond basis. *Phys. Rev. B*, 76:104432, Sep 2007.

[78] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv e-prints*, page arXiv:1609.04747, Sep 2016.

[79] Andreas M. Läuchli, Julien Sudan, and Roderich Moessner. The $S=1/2$ Kagome Heisenberg Antiferromagnet Revisited. *arXiv e-prints*, page arXiv:1611.06990, November 2016.

[80] Bryan K. Clark, Jesse M. Kinder, Eric Neuscamman, Garnet Kin-Lic Chan, and Michael J. Lawler. Striped spin liquid crystal ground state instability of kagome antiferromagnets. *Phys. Rev. Lett.*, 111:187205, Nov 2013.

[81] J. P. F. LeBlanc, Andrey E. Antipov, Federico Becca, Ireneusz W. Bulik, Garnet Kin-Lic Chan, Chia-Min Chung, Youjin Deng, Michel Ferrero, Thomas M. Henderson, Carlos A. Jiménez-Hoyos, E. Kozik, Xuan-Wen Liu, Andrew J. Millis, N. V. Prokof'ev, Mingpu Qin, Gustavo E. Scuseria, Hao Shi, B. V. Svistunov, Luca F. Tocchio, I. S. Tupitsyn, Steven R. White, Shiwei Zhang, Bo-Xiao Zheng, Zhenyue Zhu, and Emanuel Gull. Solutions of the two-dimensional hubbard model: Benchmarks and results from a wide range of numerical algorithms. *Phys. Rev. X*, 5:041041, Dec 2015.

[82] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[83] Di Luo, Giuseppe Carleo, Bryan K. Clark, and James Stokes. Gauge equivariant neural networks for quantum lattice gauge theories, 2020.

[84] Patrick A Lee, Naoto Nagaosa, and Xiao-Gang Wen. Doping a mott insulator: Physics of high-temperature superconductivity. *Reviews of modern physics*, 78(1):17, 2006.

[85] A Kitaev and C Laumann. Topological phases and quantum computation. *arXiv preprint arXiv:0904.2771*, 2009.

[86] Yu-An Chen, Anton Kapustin, and Djordje Radicevic. Exact bosonization in two spatial dimensions and a new class of lattice gauge theories. *Annals of Physics*, 393:234–253, 2018.

[87] Sergey B Bravyi and Alexei Yu Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002.

[88] John Kogut and Leonard Susskind. Hamiltonian formulation of wilson's lattice gauge theories. *Physical Review D*, 11(2):395, 1975.

[89] John B Kogut. An introduction to lattice gauge theory and spin systems. *Reviews of Modern Physics*, 51(4):659, 1979.

[90] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992.

[91] Mari Carmen Bañuls and Krzysztof Cichy. Review on novel methods for lattice gauge theories. *Reports on Progress in Physics*, 83(2):024401, January 2020. Publisher: IOP Publishing.

[92] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[93] Di Luo and Bryan K. Clark. Backflow transformations via neural networks for quantum many-body wave functions. *Phys. Rev. Lett.*, 122:226401, Jun 2019.

[94] David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Phys. Rev. Research*, 2:033429, Sep 2020.

[95] Mohamed Hibat-Allah, Martin Ganahl, Lauren E. Hayward, Roger G. Melko, and Juan Carrasquilla. Recurrent neural network wave functions. *Phys. Rev. Research*, 2:023358, Jun 2020.

[96] James Stokes, Javier Robledo Moreno, Eftychios A. Pnevmatikakis, and Giuseppe Carleo. Phases of two-dimensional spinless lattice fermions with first-quantized deep neural-network quantum states. *Phys. Rev. B*, 102:205122, Nov 2020.

[97] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Phys. Rev. Lett.*, 124:020503, Jan 2020.

[98] Sirui Lu, Xun Gao, and L.-M. Duan. Efficient representation of topologically ordered states with restricted boltzmann machines. *Phys. Rev. B*, 99:155136, Apr 2019.

[99] Xun Gao and Lu-Ming Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature Communications*, 8(1):662, Sep 2017.

[100] Ivan Glasser, Nicola Pancotti, Moritz August, Ivan D. Rodriguez, and J. Ignacio Cirac. Neural-network quantum states, string-bond states, and chiral topological states. *Physical Review X*, 8(1), Jan 2018.

[101] Kenny Choo, Giuseppe Carleo, Nicolas Regnault, and Titus Neupert. Symmetries and many-body excitations with neural-network quantum states. *Phys. Rev. Lett.*, 121:167204, Oct 2018.

[102] Tom Vieijra, Corneel Casert, Jannes Nys, Wesley De Neve, Jutho Haegeman, Jan Ryckebusch, and Frank Verstraete. Restricted boltzmann machines for quantum states with non-abelian or anyonic symmetries. *Phys. Rev. Lett.*, 124:097201, Mar 2020.

[103] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.

[104] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016.

[105] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755, 2018.

[106] Taco S Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. *arXiv preprint arXiv:1902.04615*, 2019.

[107] Gurtej Kanwar, Michael S Albergo, Denis Boyda, Kyle Cranmer, Daniel C Hackett, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E Shanahan. Equivariant flow-based sampling for lattice gauge theory. *arXiv preprint arXiv:2003.06413*, 2020.

[108] Denis Boyda, Gurtej Kanwar, Sébastien Racanière, Danilo Jimenez Rezende, Michael S Albergo, Kyle Cranmer, Daniel C Hackett, and Phiala E Shanahan. Sampling using $su(n)$ gauge equivariant flows. *arXiv preprint arXiv:2008.05456*, 2020.

[109] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[110] Franz J Wegner. Duality in generalized ising models and phase transitions without local order parameters. *Journal of Mathematical Physics*, 12(10):2259–2272, 1971.

[111] Federico Becca and Sandro Sorella. *Quantum Monte Carlo Approaches for Correlated Systems*. Cambridge University Press, 2017.

[112] K Gregor, David A Huse, R Moessner, and S L Sondhi. Diagnosing deconfinement and topological order. *New Journal of Physics*, 13(2):025009, Feb 2011.

[113] Julien Vidal, Sébastien Dusuel, and Kai Phillip Schmidt. Low-energy effective theory of the toric code model in a parallel magnetic field. *Physical Review B*, 79(3):033109, 2009.

[114] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. *arXiv preprint arXiv:1703.07950*, 2017.

[115] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, February 2018.

[116] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, Jan 2003.

[117] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014.

[118] Di Luo, Zhuo Chen, Kaiwen Hu, Zhizhen Zhao, Vera Mikyoung Hur, and Bryan K. Clark. Gauge invariant autoregressive neural networks for quantum lattice models, 2021.

[119] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[120] Xizhi Han and Sean A. Hartnoll. Deep quantum geometry of matrices. *Physical Review X*, 10(1), Mar 2020.

[121] Kenny Choo, Titus Neupert, and Giuseppe Carleo. Two-dimensional frustrated j1j2 model studied with neural network quantum states. *Physical Review B*, 100(12), Sep 2019.

[122] Mohamed Hibat-Allah, Martin Ganahl, Lauren E. Hayward, Roger G. Melko, and Juan Carrasquilla. Recurrent neural network wave functions. *Phys. Rev. Research*, 2:023358, Jun 2020.

[123] Juan Carrasquilla, Di Luo, Felipe Pérez, Ashley Milsted, Bryan K. Clark, Maksims Volkovs, and Leandro Aolita. Probabilistic simulation of quantum circuits with the transformer, 2019.

[124] Irene López Gutiérrez and Christian B. Mendl. Real time evolution with neural-network quantum states, 2020.

[125] Tom Vieijra, Corneel Casert, Jannes Nys, Wesley De Neve, Jutho Haegeman, Jan Ryckebusch, and Frank Verstraete. Restricted boltzmann machines for quantum states with non-abelian or anyonic symmetries. *Physical Review Letters*, 124(9), Mar 2020.

[126] Yusuke Nomura, Andrew S. Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Physical Review B*, 96(20), Nov 2017.

[127] Markus Schmitt and Markus Heyl. Quantum many-body dynamics in two dimensions with artificial neural networks. *Physical Review Letters*, 125(10), Sep 2020.

[128] James Stokes, Javier Robledo Moreno, Eftychios A. Pnevmatikakis, and Giuseppe Carleo. Phases of two-dimensional spinless lattice fermions with first-quantized deep neural-network quantum states. *Physical Review B*, 102(20), Nov 2020.

[129] Filippo Vicentini, Alberto Biella, Nicolas Regnault, and Cristiano Ciuti. Variational neural-network ansatz for steady states in open quantum systems. *Physical Review Letters*, 122(25), Jun 2019.

[130] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, May 2018.

[131] Kim A. Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Phys. Rev. E*, 101:023304, Feb 2020.

[132] Kim A. Nicoli, Christopher J. Anders, Lena Funcke, Tobias Hartung, Karl Jansen, Pan Kessel, Shinichi Nakajima, and Paolo Stornati. Estimation of thermodynamic observables in lattice field theories with deep generative models. *Phys. Rev. Lett.*, 126:032001, Jan 2021.

[133] Nobuyuki Yoshioka and Ryusuke Hamazaki. Constructing neural stationary states for open quantum many-body systems. *Phys. Rev. B*, 99:214306, Jun 2019.

[134] Michael J. Hartmann and Giuseppe Carleo. Neural-network approach to dissipative quantum many-body dynamics. *Phys. Rev. Lett.*, 122:250502, Jun 2019.

[135] Alexandra Nagy and Vincenzo Savona. Variational quantum monte carlo method with a neural-network ansatz for open quantum systems. *Phys. Rev. Lett.*, 122:250501, Jun 2019.

[136] Michael A. Levin and Xiao-Gang Wen. String-net condensation:a physical mechanism for topological phases. *Physical Review B*, 71(4), Jan 2005.

[137] Wilbur Shirley, Kevin Slagle, and Xie Chen. Foliated fracton order from gauging subsystem symmetries. *SciPost Physics*, 6(4), Apr 2019.

[138] Shawn X. Cui, Dawei Ding, Xizhi Han, Geoffrey Penington, Daniel Ranard, Brandon C. Rayhaun, and Zhou Shangnan. Kitaev's quantum double model as an error correcting code. *Quantum*, 4:331, Sep 2020.

[139] Mari Carmen Bañuls, Rainer Blatt, Jacopo Catani, Alessio Celi, Juan Ignacio Cirac, Marcello Dalmonte, Leonardo Fallani, Karl Jansen, Maciej Lewenstein, Simone Montangero, and et al. Simulating lattice gauge theories within quantum technologies. *The European Physical Journal D*, 74(8), Aug 2020.

[140] Claudio Rebbi. *Lattice gauge theories and Monte Carlo simulations*. World Scientific, 1983.

[141] Giuseppe Magnifico, Timo Felser, Pietro Silvi, and Simone Montangero. Lattice quantum electrodynamics in (3+1)-dimensions at finite density with tensor networks, 2020.

[142] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[143] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[144] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[145] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

[146] Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. Autoregressive neural network for simulating open quantum systems via a probabilistic formulation, 2020.

[147] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Phys. Rev. Lett.*, 124:020503, Jan 2020.

[148] Yoav Levine, Or Sharir, Nadav Cohen, and Amnon Shashua. Quantum entanglement in deep learning architectures. *Physical Review Letters*, 122(6), Feb 2019.

[149] John Kogut and Leonard Susskind. Hamiltonian formulation of wilson's lattice gauge theories. *Phys. Rev. D*, 11:395–408, Jan 1975.

[150] Alioscia Hamma, Paolo Zanardi, and Xiao-Gang Wen. String and membrane condensation on three-dimensional lattices. *Physical Review B*, 72(3), Jul 2005.

[151] Sagar Vijay, Jeongwan Haah, and Liang Fu. Fracton topological order, generalized lattice gauge theory, and duality. *Physical Review B*, 94(23), Dec 2016.

[152] Adrian Feiguin, Simon Trebst, Andreas W. W. Ludwig, Matthias Troyer, Alexei Kitaev, Zhenghan Wang, and Michael H. Freedman. Interacting anyons in topological quantum liquids: The golden chain. *Physical Review Letters*, 98(16), Apr 2007.

[153] Simon Trebst, Eddy Ardonne, Adrian Feiguin, David A. Huse, Andreas W. W. Ludwig, and Matthias Troyer. Collective states of interacting fibonacci anyons. *Physical Review Letters*, 101(5), Jul 2008.

[154] Charlotte Gils, Eddy Ardonne, Simon Trebst, Andreas W. W. Ludwig, Matthias Troyer, and Zhenghan Wang. Collective states of interacting anyons, edge states, and the nucleation of topological liquids. *Physical Review Letters*, 103(7), Aug 2009.

[155] Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *J. Mach. Learn. Res.*, 5:1471–1530, December 2004.

[156] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[157] Christopher Roth. Iterative retraining of quantum spin models using recurrent neural networks, 2020.

[158] Kyle Sprague, Juan Felipe Carrasquilla, Stephen Whitelam, and Isaac Tamblyn. Watch and learn – a generalized approach for transferrable learning in deep neural networks via physical principles. *Machine Learning: Science and Technology*, Nov 2020.

[159] Arieh Iserles. *Euler's method and beyond*, page 8–13. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2 edition, 2008.

[160] Dmitrii Kochkov and Bryan K. Clark. Variational optimization in the AI era: Computational graph states and supervised wave-function optimization, 2018.

[161] Yi-Ping Huang, Debasish Banerjee, and Markus Heyl. Dynamical quantum phase transitions in u(1) quantum link models. *Physical Review Letters*, 122(25), Jun 2019.

[162] P. Karpov, R. Verdel, Y. P. Huang, M. Schmitt, and M. Heyl. Disorder-free localization in an interacting two-dimensional lattice gauge theory, 2020.

[163] Roberto Verdel, Markus Schmitt, Yi-Ping Huang, Petr Karpov, and Markus Heyl. Variational classical networks for dynamics in interacting quantum matter, 2020.

[164] R. P. Feynman. Forces in molecules. *Phys. Rev.*, 56:340–343, Aug 1939.

[165] Mohammad Hossein Zarei. Ising order parameter and topological phase transitions: Toric code in a uniform magnetic field. *Physical Review B*, 100(12), Sep 2019.

[166] Jeongwan Haah. Local stabilizer codes in three dimensions without string logical operators. *Physical Review A*, 83(4), Apr 2011.

[167] A. Bazavov, Y. Meurice, S.-W. Tsai, J. Unmuth-Yockey, Li-Ping Yang, and Jin Zhang. Estimating the central charge from the rényi entanglement entropy. *Physical Review D*, 96(3), Aug 2017.

[168] E Ardonne, J Gukelberger, A W W Ludwig, S Trebst, and M Troyer. Microscopic models of interacting yang–lee anyons. *New Journal of Physics*, 13(4):045006, Apr 2011.

[169] D. Banerjee, M. Dalmonte, M. Müller, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller. Atomic quantum simulation of dynamical gauge fields coupled to fermionic matter: From string breaking to evolution after a quench. *Phys. Rev. Lett.*, 109:175302, Oct 2012.

[170] U.-J. Wiese. Ultracold quantum gases and lattice systems: quantum simulation of lattice gauge theories. *Annalen der Physik*, 525(10-11):777–796, Jul 2013.

[171] Chetan Nayak, Steven H. Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Reviews of Modern Physics*, 80(3):1083–1159, Sep 2008.

[172] Simone Notarnicola, Elisa Ercolessi, Paolo Facchi, Giuseppe Marmo, Saverio Pascazio, and Francesco V Pepe. Discrete abelian gauge theories for quantum simulations of qed. *Journal of Physics A: Mathematical and Theoretical*, 48(30):30FT01, Jul 2015.

[173] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[174] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[175] Jiangran Wang, Zhuo Chen, Di Luo, Zhizhen Zhao, Vera Mikyoung Hur, and Bryan K. Clark. Spacetime neural network for high dimensional quantum dynamics, 2021.

[176] Elisabet Romero, Ramunas Augulis, Vladimir I. Novoderezhkin, Marco Ferretti, Jos Thieme, Donatas Zigmantas, and Rienk van Grondelle. Quantum coherence in photosynthesis for efficient solar-energy conversion. *Nature Physics*, 10(9):676–682, Sep 2014.

[177] David C. Clary. Quantum theory of chemical reaction dynamics. *Science*, 279(5358):1879–1882, 1998.

[178] J. Ignacio Cirac and Peter Zoller. Goals and opportunities in quantum simulation. *Nature Physics*, 8(4):264–266, Apr 2012.

[179] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov,

Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, Oct 2019.

[180] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, Feb 2017.

[181] Michael E. Peskin and Daniel V. Schroeder. *An Introduction to quantum field theory*. Addison-Wesley, Reading, USA, 1995.

[182] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, Oct 2019.

[183] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, Feb 1998.

[184] Irene López Gutiérrez and Christian B. Mendl. Real time evolution with neural-network quantum states, 2021.

[185] Ernst Haier, Christian Lubich, and Gerhard Wanner. *Geometric Numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer, 2006.

[186] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Physical Review Letters*, 124(2), Jan 2020.

[187] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[188] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, Dec 2018.

[189] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[190] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

[191] Juan Carrasquilla, Di Luo, Felipe Pérez, Ashley Milsted, Bryan K. Clark, Maksims Volkovs, and Leandro Aolita. Probabilistic simulation of quantum circuits with the transformer, 2019.

[192] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, June 1982.

[193] Juan Carrasquilla, Giacomo Torlai, Roger G. Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155, March 2019.

[194] Christopher A. Fuchs and Rüdiger Schack. A Quantum-Bayesian Route to Quantum-State Space. *Foundations of Physics*, 41(3):345–356, March 2011.

[195] D. Chruściński, V. I. Man'ko, G. Marmo, and F. Ventriglia. On pseudo-stochastic matrices and pseudo-positive maps. *Physica Scripta*, 90(11):115202, October 2015.

[196] John van de Wetering. Quantum Theory is a Quasi-stochastic Process Theory. *Electronic Proceedings in Theoretical Computer Science*, 266:179–196, February 2018.

[197] E. O. Kiktenko, A. O. Malyshev, A. S. Mastiukova, V. I. Man'ko, A. K. Fedorov, and D. Chruściński. Probability representation of quantum dynamics using pseudo-stochastic maps. *arXiv:1908.03404 [quant-ph]*, August 2019.

[198] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[199] K. De Raedt, K. Michielsen, H. De Raedt, B. Trieu, G. Arnold, M. Richter, Th. Lippert, H. Watanabe, and N. Ito. Massively parallel quantum computer simulator. *Computer Physics Communications*, 176(2):121–136, January 2007.

[200] I. Markov and Y. Shi. Simulating Quantum Computation by Contracting Tensor Networks. *SIAM Journal on Computing*, 38(3):963–981, January 2008.

[201] Mikhail Smelyanskiy, Nicolas P. D. Sawaya, and Alán Aspuru-Guzik. qHiPSTER: The Quantum High Performance Software Testing Environment. *arXiv:1601.07195 [quant-ph]*, January 2016.

[202] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, Erik W. Draeger, Eric T. Holland, and Robert Wisnieff. Breaking the 49-Qubit Barrier in the Simulation of Quantum Circuits. *arXiv:1710.05867 [quant-ph]*, October 2017.

[203] Jianxin Chen, Fang Zhang, Cupjin Huang, Michael Newman, and Yaoyun Shi. Classical Simulation of Intermediate-Size Quantum Circuits. *arXiv:1805.01450 [quant-ph]*, May 2018.

[204] Riling Li, Bujiao Wu, Mingsheng Ying, Xiaoming Sun, and Guangwen Yang. Quantum Supremacy Circuit Simulation on Sunway TaihuLight. *arXiv:1804.04797 [quant-ph]*, April 2018.

[205] Bjarni Jónsson, Bela Bauer, and Giuseppe Carleo. Neural-network states for the classical simulation of quantum computing. *arXiv:1808.05232 [cond-mat, physics:physics, physics:quant-ph]*, August 2018.

[206] Huitao Shen. Mutual Information Scaling and Expressive Power of Sequence Models. *arXiv:1905.04271 [cond-mat, stat]*, May 2019.

[207] Yoav Levine, Or Sharir, Nadav Cohen, and Amnon Shashua. Quantum entanglement in deep learning architectures. *Phys. Rev. Lett.*, 122:065301, Feb 2019.

[208] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1:221–244, 1971.

[209] Branko Ćurgus and Robert I. Jewett. Somewhat stochastic matrices. *arXiv:0709.0309 [math]*, September 2007.

[210] Branko Ćurgus and Robert I. Jewett. Somewhat Stochastic Matrices. *The American Mathematical Monthly*, 122(1):36–42, January 2015.

[211] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[212] Roland Assaraf and Michel Caffarel. Zero-variance principle for monte carlo algorithms. *Phys. Rev. Lett.*, 83:4682–4685, Dec 1999.

[213] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015.

[214] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv:1607.06450 [cs, stat]*, July 2016.

[215] Ilia A. Luchnikov, Alexander Ryzhov, Pieter-Jan Stas, Sergey N. Filippov, and Henni Ouerdane. Variational Autoencoder Reconstruction of Complex Many-Body Physics. *Entropy*, 21(11):1091, November 2019.

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

[217] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[218] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[219] Juan Carrasquilla, Giacomo Torlai, Roger G. Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, Mar 2019.

[220] Sam Cree and Jamie Sikora. A fidelity measure for quantum states based on the matrix geometric mean, 2021.

[221] Radosław Adamczak. Metric and classical fidelity uncertainty relations for random unitary matrices. *Journal of Physics A: Mathematical and Theoretical*, 50(10):105302, Feb 2017.

[222] Sebastian Deffner. Demonstration of entanglement assisted invariance on ibm's quantum experience. *Heliyon*, 3(11):e00444, Nov 2017.

[223] Huikang Huang, Haozhen Situ, and Shenggen Zheng. Bidirectional information flow quantum state tomography. *Chinese Physics Letters*, 38(4):040303, 2021.

[224] Marcus Appleby, Christopher A. Fuchs, Blake C. Stacey, and Huangjun Zhu. Introducing the Qplex: A novel arena for quantum theory. *The European Physical Journal D*, 71(7):197, July 2017.

[225] Wen Wei Ho and Timothy H. Hsieh. Efficient variational simulation of non-trivial quantum states. *SciPost Phys.*, 6:29, 2019.

[226] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):1–7, July 2014.

[227] Christopher Ferrie. Quasi-probability representations of quantum theory with applications to quantum information science. *Reports on Progress in Physics*, 74(11):116001, October 2011.

[228] Christopher A. Fuchs, Michael C. Hoang, and Blake C. Stacey. The SIC Question: History and State of Play. *Axioms*, 6(4):21, July 2017.

[229] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. Quantum supremacy and the complexity of random circuit sampling. 2018.

[230] Igor L. Markov, Aneeqa Fatima, Sergei V. Isakov, and Sergio Boixo. Quantum Supremacy Is Both Closer and Farther than It Appears. *arXiv:1807.10749 [quant-ph]*, July 2018.

[231] N. J. Cerf and S. E. Koonin. Monte Carlo simulation of quantum computation1Presented at the IMACS Seminar on Monte Carlo Methods, Brussels, 1–3 April 1997.1. *Mathematics and Computers in Simulation*, 47(2):143–152, August 1998.

[232] A. Mari and J. Eisert. Positive wigner functions render classical simulation of quantum computation efficient. *Phys. Rev. Lett.*, 109:230503, Dec 2012.

[233] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image Transformer. *arXiv:1802.05751 [cs]*, June 2018.

[234] K. Kraus. *States, Effects, and Operations: Fundamental Notions of Quantum Theory*. Lecture Notes in Physics. Springer-Verlag, Berlin Heidelberg, 1983.

[235] Christopher A. Fuchs and Rüdiger Schack. Quantum-Bayesian coherence. *Reviews of Modern Physics*, 85(4):1693–1715, December 2013.

[236] Joseph M. Renes, Robin Blume-Kohout, A. J. Scott, and Carlton M. Caves. Symmetric informationally complete quantum measurements. *Journal of Mathematical Physics*, 45(6):2171–2180, May 2004.

[237] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[238] Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. Autoregressive neural network for simulating open quantum systems via a probabilistic formulation, 2021.

[239] Frank Verstraete, Michael M. Wolf, and J. Ignacio Cirac. Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics*, 5(9):633–636, Sep 2009.

[240] Julio T. Barreiro, Markus Müller, Philipp Schindler, Daniel Nigg, Thomas Monz, Michael Chwalla, Markus Hennrich, Christian F. Roos, Peter Zoller, and Rainer Blatt. An open-system quantum simulator with trapped ions. *Nature*, 470(7335):486–491, Feb 2011.

[241] L M Sieberer, M Buchhold, and S Diehl. Keldysh field theory for driven open quantum systems. *Reports on Progress in Physics*, 79(9):096001, aug 2016.

[242] Mohammad F. Maghrebi and Alexey V. Gorshkov. Nonequilibrium many-body steady states via keldysh formalism. *Phys. Rev. B*, 93:014307, Jan 2016.

[243] Eduardo Mascarenhas, Hugo Flayac, and Vincenzo Savona. Matrix-product-operator approach to the nonequilibrium steady state of driven-dissipative quantum arrays. *Phys. Rev. A*, 92:022116, Aug 2015.

[244] Jian Cui, J. Ignacio Cirac, and Mari Carmen Bañuls. Variational matrix product operators for the steady state of dissipative quantum systems. *Phys. Rev. Lett.*, 114:220601, Jun 2015.

[245] Daniel Jaschke, Simone Montangero, and Lincoln D Carr. One-dimensional many-body entangled open quantum systems with tensor network methods. *Quantum Science and Technology*, 4(1):013001, nov 2018.

[246] A. H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero. Positive tensor network approach for simulating open quantum many-body systems. *Phys. Rev. Lett.*, 116:237201, Jun 2016.

[247] Alberto Biella, Jiasen Jin, Oscar Viyuela, Cristiano Ciuti, Rosario Fazio, and Davide Rossini. Linked cluster expansions for open quantum systems on a lattice. *Phys. Rev. B*, 97:035103, Jan 2018.

[248] Jiasen Jin, Alberto Biella, Oscar Viyuela, Leonardo Mazza, Jonathan Keeling, Rosario Fazio, and Davide Rossini. Cluster mean-field approach to the steady-state phase diagram of dissipative spin systems. *Phys. Rev. X*, 6:031011, Jul 2016.

[249] S. Finazzi, A. Le Boité, F. Storme, A. Baksic, and C. Ciuti. Corner-space renormalization method for driven-dissipative two-dimensional correlated systems. *Phys. Rev. Lett.*, 115:080604, Aug 2015.

[250] R. Rota, F. Storme, N. Bartolo, R. Fazio, and C. Ciuti. Critical behavior of dissipative two-dimensional spin lattices. *Phys. Rev. B*, 95:134431, Apr 2017.

[251] Riccardo Rota, Fabrizio Minganti, Cristiano Ciuti, and Vincenzo Savona. Quantum critical regime in a quadratically driven nonlinear photonic lattice. *Phys. Rev. Lett.*, 122:110405, Mar 2019.

[252] Wim Casteels, Ryan M. Wilson, and Michiel Wouters. Gutzwiller monte carlo approach for a critical dissipative spin model. *Phys. Rev. A*, 97:062107, Jun 2018.

[253] Alexandra Nagy and Vincenzo Savona. Driven-dissipative quantum monte carlo method for open quantum systems. *Phys. Rev. A*, 97:052129, May 2018.

[254] Nathan Shammah, Shahnawaz Ahmed, Neill Lambert, Simone De Liberato, and Franco Nori. Open quantum systems with local and collective incoherent processes: Efficient numerical simulations using permutational invariance. *Phys. Rev. A*, 98:063815, Dec 2018.

[255] Filippo Vicentini, Fabrizio Minganti, Alberto Biella, Giuliano Orso, and Cristiano Ciuti. Optimal stochastic unraveling of disordered open quantum systems: Application to driven-dissipative photonic lattices. *Phys. Rev. A*, 99:032115, Mar 2019.

[256] Augustine Kshetrimayum, Hendrik Weimer, and Román Orús. A simple tensor network algorithm for two-dimensional steady states. *Nature Communications*, 8(1):1291, Nov 2017.

[257] Iacopo Carusotto and Cristiano Ciuti. Quantum fluids of light. *Rev. Mod. Phys.*, 85:299–366, Feb 2013.

[258] Michael J Hartmann. Quantum simulation with interacting photons. *Journal of Optics*, 18(10):104005, sep 2016.

[259] Changsuk Noh and Dimitris G Angelakis. Quantum simulations and many-body physics with light. *Reports on Progress in Physics*, 80(1):016401, nov 2016.

[260] Nicola Bartolo, Fabrizio Minganti, Wim Casteels, and Cristiano Ciuti. Exact steady state of a kerr resonator with one- and two-photon driving and dissipation: Controllable wigner-function multimodality and dissipative phase transitions. *Phys. Rev. A*, 94:033841, Sep 2016.

[261] Alberto Biella, Florent Storme, José Lebreuilly, Davide Rossini, Rosario Fazio, Iacopo Carusotto, and Cristiano Ciuti. Phase diagram of incoherently driven strongly correlated photonic lattices. *Phys. Rev. A*, 96:023839, Aug 2017.

[262] Matteo Biondi, Gianni Blatter, Hakan E. Türeci, and Sebastian Schmidt. Nonequilibrium gas-liquid transition in the driven-dissipative photonic lattice. *Phys. Rev. A*, 96:043809, Oct 2017.

[263] H. J. Carmichael. Breakdown of photon blockade: A dissipative quantum phase transition in zero dimensions. *Phys. Rev. X*, 5:031028, Sep 2015.

[264] W. Casteels, R. Fazio, and C. Ciuti. Critical dynamical properties of a first-order dissipative phase transition. *Phys. Rev. A*, 95:012128, Jan 2017.

[265] W. Casteels, F. Storme, A. Le Boité, and C. Ciuti. Power laws in the dynamic hysteresis of quantum nonlinear photonic resonators. *Phys. Rev. A*, 93:033824, Mar 2016.

[266] J. M. Fink, A. Dombi, A. Vukics, A. Wallraff, and P. Domokos. Observation of the photon-blockade breakdown phase transition. *Phys. Rev. X*, 7:011012, Jan 2017.

[267] Thomas Fink, Anne Schade, Sven Höfling, Christian Schneider, and Ataç Imamoglu. Signatures of a dissipative phase transition in photon correlation measurements. *Nature Physics*, 14(4):365–369, Apr 2018.

[268] Mattias Fitzpatrick, Neereja M. Sundaresan, Andy C. Y. Li, Jens Koch, and Andrew A. Houck. Observation of a dissipative phase transition in a one-dimensional circuit qed lattice. *Phys. Rev. X*, 7:011016, Feb 2017.

[269] M. Foss-Feig, P. Niroula, J. T. Young, M. Hafezi, A. V. Gorshkov, R. M. Wilson, and M. F. Maghrebi. Emergent equilibrium in many-body optical bistability. *Phys. Rev. A*, 95:043826, Apr 2017.

[270] E. M. Kessler, G. Giedke, A. Imamoglu, S. F. Yelin, M. D. Lukin, and J. I. Cirac. Dissipative phase transition in a central spin system. *Phys. Rev. A*, 86:012116, Jul 2012.

[271] Jamir Marino and Sebastian Diehl. Driven markovian quantum criticality. *Phys. Rev. Lett.*, 116:070407, Feb 2016.

[272] Vincenzo Savona. Spontaneous symmetry breaking in a quadratically driven nonlinear photonic lattice. *Phys. Rev. A*, 96:033826, Sep 2017.

[273] L. M. Sieberer, S. D. Huber, E. Altman, and S. Diehl. Dynamical critical phenomena in driven-dissipative systems. *Phys. Rev. Lett.*, 110:195301, May 2013.

[274] Filippo Vicentini, Fabrizio Minganti, Riccardo Rota, Giuliano Orso, and Cristiano Ciuti. Critical slowing down in driven-dissipative bose-hubbard lattices. *Phys. Rev. A*, 97:013853, Jan 2018.

[275] Tony E. Lee, Sarang Gopalakrishnan, and Mikhail D. Lukin. Unconventional magnetism via optical pumping of interacting spin systems. *Phys. Rev. Lett.*, 110:257204, Jun 2013.

[276] R Rota, F Minganti, A Biella, and C Ciuti. Dynamical properties of dissipative XYZ heisenberg lattices. *New Journal of Physics*, 20(4):045003, apr 2018.

[277] Alexandre Blais, Arne L. Grimsmo, S. M. Girvin, and Andreas Wallraff. Circuit quantum electrodynamics, 2020.

[278] Nobuyuki Yoshioka, Yuya O. Nakagawa, Kosuke Mitarai, and Keisuke Fujii. Variational quantum algorithm for non-equilibrium steady states, 2019.

[279] Zidu Liu, L. M. Duan, and Dong-Ling Deng. Solving quantum master equations with deep quantum neural networks, 2020.

[280] Chee-Kong Lee, Pranay Patil, Shengyu Zhang, and Chang-Yu Hsieh. A neural-network variational quantum algorithm for many-body dynamics, 2020.

[281] Nathan Ramusat and Vincenzo Savona. A quantum algorithm for the direct estimation of the steady state of open quantum systems, 2020.

[282] Huan-Yu Liu, Tai-Ping Sun, Yu-Chun Wu, and Guo-Ping Guo. Variational quantum algorithms for steady states of open quantum systems, 2021.

[283] Orazio Scarlatella, Aashish A. Clerk, Rosario Fazio, and Marco Schiró. Dynamical mean-field theory for open markovian quantum many body systems, 2020.

[284] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural network methods in quantum mechanics. *Computer Physics Communications*, 104(1):1–14, August 1997.

[285] Filippo Vicentini, Alberto Biella, Nicolas Regnault, and Cristiano Ciuti. Variational neural-network ansatz for steady states in open quantum systems. *Phys. Rev. Lett.*, 122:250503, Jun 2019.

[286] Dong Yuan, Heran Wang, Zhong Wang, and Dong-Ling Deng. Solving the liouvillian gap with artificial neural networks, 2020.

[287] Giacomo Torlai and Roger G. Melko. Latent space purification via neural density operators. *Phys. Rev. Lett.*, 120:240503, Jun 2018.

[288] Francesco Ferrari, Federico Becca, and Juan Carrasquilla. Neural gutzwiller-projected variational wave functions. *Phys. Rev. B*, 100:125131, Sep 2019.

[289] Tom Westerhout, Nikita Astrakhantsev, Konstantin S. Tikhonov, Mikhail I. Katsnelson, and Andrey A. Bagrov. Generalization properties of neural network approximations to frustrated magnet ground states. *Nature Communications*, 11(1):1593, March 2020.

[290] Attila Szabó and Claudio Castelnovo. Neural network wave functions and the sign problem. *Phys. Rev. Research*, 2:033075, Jul 2020.

[291] J. S. Lundeen, A. Feito, H. Coldenstrodt-Ronge, K. L. Pregnell, Ch. Silberhorn, T. C. Ralph, J. Eisert, M. B. Plenio, and I. A. Walmsley. Tomography of quantum detectors. *Nature Physics*, 5(1):27–30, Jan 2009.

[292] Juan Carrasquilla, Giacomo Torlai, Roger G Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 2019.

[293] E. O. Kiktenko, A. O. Malyshev, A. S. Mastiukova, V. I. Man'ko, A. K. Fedorov, and D. Chruściński. Probability representation of quantum dynamics using pseudostochastic maps. *Physical Review A*, 101(5), May 2020.

[294] Peter Cha, Paul Ginsparg, Felix Wu, Juan Carrasquilla, Peter L. McMahon, and Eun-Ah Kim. Attention-based Quantum Tomography. *arXiv:2006.12469 [quant-ph]*, June 2020.

[295] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[296] Norbert Schuch, Michael M. Wolf, Frank Verstraete, and J. Ignacio Cirac. Simulation of quantum many-body systems with strings of operators and monte carlo tensor contractions. *Phys. Rev. Lett.*, 100:040501, Jan 2008.

[297] Ankit Mahajan and Sandeep Sharma. Symmetry-projected jastrow mean-field wave function in variational monte carlo. *The Journal of Physical Chemistry A*, 123(17):3911–3921, 2019.

[298] Daisuke Tahara and Masatoshi Imada. Variational monte carlo method combined with quantum-number projection and multi-variable optimization. *Journal of the Physical Society of Japan*, 77(11):114701, 2008.

[299] Yiheng Qiu, Thomas M Henderson, Jinmo Zhao, and Gustavo E Scuseria. Projected coupled cluster theory. *The Journal of Chemical Physics*, 147(6):064111, 2017.

[300] Irene López Gutiérrez and Christian B. Mendl. Real time evolution with neural-network quantum states, 2019.

[301] J.R. Johansson, P.D. Nation, and Franco Nori. Qutip 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234 – 1240, 2013.

[302] J.R. Johansson, P.D. Nation, and Franco Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760 – 1772, 2012.

[303] Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. Supplementary material, 2020.

[304] Fabrício M. Souza and L. Sanz. Lindblad formalism based on fermion-to-qubit mapping for nonequilibrium open quantum systems. *Phys. Rev. A*, 96:052110, Nov 2017.

[305] YiJing Yan. Theory of open quantum systems with bath of electrons and phonons and spins: Many-dissipaton density matrixes approach. *The Journal of Chemical Physics*, 140(5):054105, February 2014.

[306] Timothy C. Berkelbach and Michael Thoss. Special topic on dynamics of open quantum systems. *The Journal of Chemical Physics*, 152(2):020401, January 2020.

[307] Xiongjie Yu, Di Luo, and Bryan K. Clark. Beyond many-body localized states in a spin-disordered hubbard model. *Physical Review B*, 98(11), Sep 2018.

[308] Xiongjie Yu. Many-body localization and tensor networks. 2018.

[309] D. M. Basko, I. L. Aleiner, and B. L. Altshuler. Metal–insulator transition in a weakly interacting many-electron system with localized single-particle states. *Annals of Physics*, 321(5):1126–1205, May 2006.

[310] I. V. Gornyi, A. D. Mirlin, and D. G. Polyakov. Interacting Electrons in Disordered Wires: Anderson Localization and Low-T Transport. *Phys. Rev. Lett.*, 95(20):206603, November 2005.

[311] P. W. Anderson. Absence of Diffusion in Certain Random Lattices. *Phys. Rev.*, 109(5):1492–1505, March 1958.

[312] L. Fleishman and P. W. Anderson. Interactions and the anderson transition. *Phys. Rev. B*, 21:2366–2377, Mar 1980.

[313] Vadim Oganesyan and David A. Huse. Localization of interacting fermions at high temperature. *Phys. Rev. B*, 75(15):155111, April 2007.

[314] Marko Žnidarič, Tomaž Prosen, and Peter Prelovšek. Many-body localization in the Heisenberg XXZ magnet in a random field. *Phys. Rev. B*, 77(6):064426, February 2008.

[315] Timothy C. Berkelbach and David R. Reichman. Conductivity of disordered quantum lattice models at infinite temperature: Many-body localization. *Phys. Rev. B*, 81:224429, Jun 2010.

[316] Arijeet Pal and David A. Huse. Many-body localization phase transition. *Phys. Rev. B*, 82(17):174411, November 2010.

[317] Jonas A. Kjäll, Jens H. Bardarson, and Frank Pollmann. Many-Body Localization in a Disordered Quantum Ising Chain. *Phys. Rev. Lett.*, 113(10):107204, September 2014.

[318] A. De Luca and A. Scardicchio. Ergodicity breaking in a model showing many-body localization. *EPL*, 101(3):37003, 2013.

[319] David Pekker and Bryan K. Clark. Encoding the structure of many-body localization with matrix product operators. *Phys. Rev. B*, 95:035116, Jan 2017.

[320] Yevgeny Bar Lev and David R. Reichman. Dynamics of many-body localization. *Phys. Rev. B*, 89:220201, Jun 2014.

[321] Yevgeny Bar Lev, Guy Cohen, and David R. Reichman. Absence of diffusion in an interacting system of spinless fermions on a one-dimensional disordered lattice. *Phys. Rev. Lett.*, 114:100601, Mar 2015.

[322] Kartiek Agarwal, Sarang Gopalakrishnan, Michael Knap, Markus Müller, and Eugene Demler. Anomalous diffusion and griffiths effects near the many-body localization transition. *Phys. Rev. Lett.*, 114:160401, Apr 2015.

[323] David J. Luitz, Nicolas Laflorencie, and Fabien Alet. Many-body localization edge in the random-field Heisenberg chain. *Phys. Rev. B*, 91(8):081103, February 2015.

[324] Xiongjie Yu, David Pekker, and Bryan K. Clark. Finding matrix product state representations of highly excited eigenstates of many-body localized hamiltonians. *Phys. Rev. Lett.*, 118:017201, Jan 2017.

[325] Xiongjie Yu, David J. Luitz, and Bryan K. Clark. Bimodal entanglement entropy distribution in the many-body localization transition. *Phys. Rev. B*, 94:184202, Nov 2016.

[326] Maksym Serbyn, Z. Papić, and Dmitry A. Abanin. Criterion for Many-Body Localization-Delocalization Phase Transition. *Phys. Rev. X*, 5(4):041047, December 2015.

[327] Soumya Bera, Henning Schomerus, Fabian Heidrich-Meisner, and Jens H. Bardarson. Many-Body Localization Characterized from a One-Particle Perspective. *Phys. Rev. Lett.*, 115(4):046603, July 2015.

[328] David J. Luitz, Nicolas Laflorencie, and Fabien Alet. Extended slow dynamical regime close to the many-body localization transition. *Phys. Rev. B*, 93(6):060201, February 2016.

[329] Rajeev Singh, Jens H. Bardarson, and Frank Pollmann. Signatures of the many-body localization transition in the dynamics of entanglement and bipartite fluctuations. *New J. Phys.*, 18(2):023046, 2016.

[330] Frank Pollmann, Vedika Khemani, J. Ignacio Cirac, and S. L. Sondhi. Efficient variational diagonalization of fully many-body localized hamiltonians. *Phys. Rev. B*, 94:041116, Jul 2016.

[331] Vedika Khemani, Frank Pollmann, and S. L. Sondhi. Obtaining highly excited eigenstates of many-body localized hamiltonians by the density matrix renormalization group approach. *Phys. Rev. Lett.*, 116:247204, Jun 2016.

[332] S. P. Lim and D. N. Sheng. Many-body localization and transition by density matrix renormalization group and exact diagonalization studies. *Phys. Rev. B*, 94:045111, Jul 2016.

[333] David J. Luitz. Long tail distributions near the many-body localization transition. *Phys. Rev. B*, 93(13):134201, April 2016.

[334] David J. Luitz and Yevgeny Bar Lev. Anomalous thermalization in ergodic systems. *Phys. Rev. Lett.*, 117:170404, Oct 2016.

[335] Soumya Bera and Arul Lakshminarayan. Local entanglement structure across a many-body localization transition. *Phys. Rev. B*, 93(13):134204, April 2016.

[336] Maksym Serbyn, Alexios A. Michailidis, Dmitry A. Abanin, and Z. Papić. Power-law entanglement spectrum in many-body localized phases. *Phys. Rev. Lett.*, 117:160601, Oct 2016.

[337] S. Stagraczyński, L. Chotorlishvili, M. Schüler, M. Mierzejewski, and J. Berakdar. Many-body localization phase in a spin-driven chiral multiferroic chain. *Phys. Rev. B*, 96:054440, Aug 2017.

[338] Vedika Khemani, S. P. Lim, D. N. Sheng, and David A. Huse. Critical properties of the many-body localization transition. *Phys. Rev. X*, 7:021013, Apr 2017.

[339] David J. Luitz and Yevgeny Bar Lev. Information propagation in isolated quantum systems. *Phys. Rev. B*, 96:020406, Jul 2017.

[340] John Z. Imbrie. Diagonalization and many-body localization for a disordered quantum spin chain. *Phys. Rev. Lett.*, 117:027201, Jul 2016.

[341] P. Prelovšek, O. S. Barišić, and M. Žnidarič. Absence of full many-body localization in the disordered hubbard chain. *Phys. Rev. B*, 94:241104, Dec 2016.

[342] J. Zakrzewski and D. Delande. Spin-charge separation and many-body localization. *ArXiv e-prints*, February 2018.

[343] Yevgeny Bar Lev, David R. Reichman, and Yoav Sagi. Many-body localization in system with a completely delocalized single-particle spectrum. *Phys. Rev. B*, 94:201116, Nov 2016.

[344] G. Lemut, M. Mierzejewski, and J. Bonca. Complete many-body localization in the t-J model caused by random magnetic field. *ArXiv e-prints*, November 2017.

[345] Michael Schreiber, Sean S. Hodgman, Pranjal Bordia, Henrik P. Lüschen, Mark H. Fischer, Ronen Vosk, Ehud Altman, Ulrich Schneider, and Immanuel Bloch. Observation of many-body localization of interacting fermions in a quasirandom optical lattice. *Science*, 349(6250):842–845, 2015.

[346] S. S. Kondov, W. R. McGehee, W. Xu, and B. DeMarco. Disorder-induced localization in a strongly correlated atomic hubbard gas. *Phys. Rev. Lett.*, 114:083002, Feb 2015.

[347] Pranjal Bordia, Henrik P. Lüschen, Sean S. Hodgman, Michael Schreiber, Immanuel Bloch, and Ulrich Schneider. Coupling identical one-dimensional many-body localized systems. *Phys. Rev. Lett.*, 116:140401, Apr 2016.

[348] H. P. Lüschen, P. Bordia, S. Scherg, F. Alet, E. Altman, U. Schneider, and I. Bloch. Evidence for Griffiths-Type Dynamics near the Many-Body Localization Transition in Quasi-Periodic Systems. *ArXiv e-prints*, December 2016.

[349] Jens H. Bardarson, Frank Pollmann, and Joel E. Moore. Unbounded growth of entanglement in models of many-body localization. *Phys. Rev. Lett.*, 109:017202, Jul 2012.

[350] Maksym Serbyn, Z. Papić, and Dmitry A. Abanin. Local Conservation Laws and the Structure of the Many-Body Localized States. *Phys. Rev. Lett.*, 111(12):127201, September 2013.

[351] John Z. Imbrie. On many-body localization for quantum spin chains. *Journal of Statistical Physics*, 163(5):998–1048, Jun 2016.

[352] David A. Huse, Rahul Nandkishore, and Vadim Oganesyan. Phenomenology of fully many-body-localized systems. *Phys. Rev. B*, 90(17):174202, November 2014.

[353] Anushya Chandran, Isaac H. Kim, Guifre Vidal, and Dmitry A. Abanin. Constructing local integrals of motion in the many-body localized phase. *Phys. Rev. B*, 91(8):085425, February 2015.

[354] A. Chandran, J. Carrasquilla, I. H. Kim, D. A. Abanin, and G. Vidal. Spectral tensor networks for many-body localization. *Phys. Rev. B*, 92:024201, Jul 2015.

[355] V. Ros, M. Müller, and A. Scardicchio. Integrals of motion in the many-body localized phase. *Nuclear Physics B*, 891:420–465, February 2015.

[356] David Pekker, Bryan K. Clark, Vadim Oganesyan, and Gil Refael. Fixed points of wegner-wilson flows and many-body localization. *Phys. Rev. Lett.*, 119:075701, Aug 2017.

[357] Stephen Inglis and Lode Pollet. Accessing Many-Body Localized States through the Generalized Gibbs Ensemble. *Physical Review Letters*, 117(12):120402, September 2016.

[358] Thorsten B. Wahl, Arijeet Pal, and Steven H. Simon. Efficient representation of fully many-body localized systems using tensor networks. *Phys. Rev. X*, 7:021018, May 2017.

[359] John Z. Imbrie, Valentina Ros, and Antonello Scardicchio. Local integrals of motion in many-body localized systems. *Annalen der Physik*, 529(7):1600278–n/a, 2017. 1600278.

[360] Cecile Monthus. Many-Body Localization : construction of the emergent local conserved operators via block real-space renormalization. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(3):033101, March 2016. arXiv: 1509.06258.

[361] A. Chandran, J. Carrasquilla, I. H. Kim, D. A. Abanin, and G. Vidal. Spectral tensor networks for many-body localization. *Phys. Rev. B*, 92:024201, Jul 2015.

[362] Jens H. Bardarson, Frank Pollmann, and Joel E. Moore. Unbounded growth of entanglement in models of many-body localization. *Phys. Rev. Lett.*, 109:017202, Jul 2012.

[363] Chen Ning Yang. η. *Phys. Rev. Lett.*, 63:2144–2147, Nov 1989.

[364] Shoucheng Zhang. Pseudospin symmetry and new collective modes of the hubbard model. *Phys. Rev. Lett.*, 65:120–122, Jul 1990.

[365] Andrew C. Potter and Romain Vasseur. Symmetry constraints on many-body localization. *Phys. Rev. B*, 94:224206, Dec 2016.

[366] Ivan V. Protopopov, Wen Wei Ho, and Dmitry A. Abanin. Effect of su(2) symmetry on many-body localization and thermalization. *Phys. Rev. B*, 96:041122, Jul 2017.

[367] O. Vafek, N. Regnault, and B. A. Bernevig. Entanglement of Exact Excited Eigenstates of the Hubbard Model in Arbitrary Dimension. *ArXiv e-prints*, Aug 2016.

[368] Edwin Miles Stoudenmire and Steven White. ITensor - Intelligent Tensor Library. *ITensor*.

[369] Kenneth G. Wilson. Confinement of quarks. *Phys. Rev. D*, 10:2445–2459, 1974.

[370] Carleton DeTar and Steven Gottlieb. Lattice quantum chromodynamics comes of age. *Physics Today*, 57(2):45–51, 2004.

[371] Andreas S. Kronfeld. Twenty-first Century Lattice Gauge Theory: Results from the QCD Lagrangian. *Ann. Rev. Nucl. Part. Sci.*, 62:265–284, 2012.

[372] Aida X. El-Khadra. QCD on the Lattice: The Central Role of Effective Field Theory. In *Proceedings, Theoretical Advanced Study Institute in Elementary Particle Physics: Journeys Through the Precision Frontier: Amplitudes for Colliders (TASI 2014): Boulder, Colorado, June 2-27, 2014*, pages 183–233. WSP, WSP, 2015.

[373] Heng-Tong Ding, Frithjof Karsch, and Swagato Mukherjee. Thermodynamics of strong-interaction matter from Lattice QCD. *Int. J. Mod. Phys. E*, 24(10):1530007, 2015.

[374] Claudia Ratti. Lattice QCD and heavy ion collisions: a review of recent progress. *Reports on Progress in Physics*, 81(8):084301, jul 2018.

[375] Christoph Lehner et al. Opportunities for Lattice QCD in Quark and Lepton Flavor Physics. *Eur. Phys. J. A*, 55(11):195, 2019.

[376] S. Aoki et al. FLAG Review 2019. 2019.

[377] M. C. Bañuls, R. Blatt, J. Catani, A. Celi, J.I. Cirac, M. Dalmonte, L. Fallani, K. Jansen, M. Lewenstein, S. Montangero, C. A. Muschik, B. Reznik, E. Rico, L. Tagliacozzo, K. Van Acoleyen, F. Verstraete, U.-J. Wiese, M. Wingate, J. Zakrzewski, and P. Zoller. Simulating lattice gauge theories within quantum technologies. *arXiv:1911.00003*, 2019.

[378] John Kogut and Leonard Susskind. Hamiltonian formulation of wilson's lattice gauge theories. *Phys. Rev. D*, 11:395–408, 1975.

[379] D. Horn. Finite matrix models with continuous local gauge invariance. *Phys. Lett. B*, 100:149, 1981.

[380] P. Orland and D. Rohrlich. Lattice gauge magnets: Local isospin from spin. *Nucl. Phys. B*, 338:647, 1990.

[381] S. Chandrasekharan and U.-J. Wiese. Quantum link models: A discrete approach to gauge theories. *Nucl. Phys. B*, 492:455, 1997.

[382] J. Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[383] Esteban A. Martinez, Christine A. Muschik, Philipp Schindler, Alexander Nigg, Daniel AU Erhard, Markus Heyl, Philipp Hauke, Marcello Dalmonte, Thomas Monz, Peter Zoller, and Rainer Blatt. Real-time dynamics of lattice gauge theories with a few-qubit quantum computer. *Nature*, 534:516, 2016.

[384] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller. Self-verifying variational quantum simulation of lattice models. *Nature*, 569:355, 2019.

[385] U.-J. Wiese. Ultracold quantum gases and lattice systems: quantum simulation of lattice gauge theories. *Ann. der Phys.*, 525:777–796, 2013.

[386] Erez Zohar, J Ignacio Cirac, and Benni Reznik. Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices. *Reports on Progress in Physics*, 79(1):014401, 2015.

[387] A. Bermudez, L. Mazza, M. Rizzi, N. Goldman, M. Lewenstein, and M. A. Martin-Delgado. Wilson fermions and axion electrodynamics in optical lattices. *Phys. Rev. Lett.*, 105:190404, Nov 2010.

[388] Yoshihito Kuno, Kenichi Kasamatsu, Yoshiro Takahashi, Ikuo Ichinose, and Tetsuo Matsui. Real-time dynamics and proposal for feasible experiments of lattice gauge–higgs model simulated by cold atoms. *New Journal of Physics*, 17(6):063005, jun 2015.

[389] Yoshihito Kuno, Shinya Sakane, Kenichi Kasamatsu, Ikuo Ichinose, and Tetsuo Matsui. Quantum simulation of $(1 + 1)$-dimensional u(1) gauge-higgs model on a lattice by cold bose gases. *Phys. Rev. D*, 95:094507, May 2017.

[390] Luca Barbiero, Christian Schweizer, Monika Aidelsburger, Eugene Demler, Nathan Goldman, and Fabian Grusdt. Coupling ultracold matter to dynamical gauge fields in optical lattices: From flux attachment to 2 lattice gauge theories. *Science Advances*, 5(10), 2019.

[391] Christian Schweizer, Fabian Grusdt, Moritz Berngruber, Luca Barbiero, Eugene Demler, Nathan Goldman, Immanuel Bloch, and Monika Aidelsburger. Floquet approach to F_2 lattice gauge theories with ultracold atoms in optical lattices. *Nat. Phys.*, 15:1168–1173, 2019.

[392] Frederik Görg, Kilian Sandholzer, Joaquín Minguzzi, Rémi Desbuquois, Michael Messer, and Tilman Esslinger. Realization of density-dependent peierls phases to engineer quantized gauge fields coupled to ultracold matter. *Nat. Phys.*, 15:1161–1167, 2019.

[393] Alexander Mil, Torsten V. Zache, Apoorva Hegde, Andy Xia, Rohit P. Bhatt, Markus K. Oberthaler, Philipp Hauke, Jürgen Berges, and Fred Jendrzejewski. Realizing a scalable building block of a u(1) gauge theory with cold atomic mixtures. *arXiv:1909.07641*, 2019.

[394] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller. Cold bosonic atoms in optical lattices. *Phys. Rev. Lett.*, 81:3108–3111, Oct 1998.

[395] W. Hofstetter, J. I. Cirac, P. Zoller, E. Demler, and M. D. Lukin. High-temperature superfluidity of fermionic atoms in optical lattices. *Phys. Rev. Lett.*, 89:220407, Nov 2002.

[396] Ryan Barnett, Dmitry Petrov, Mikhail Lukin, and Eugene Demler. Quantum magnetism with multicomponent dipolar molecules in an optical lattice. *Phys. Rev. Lett.*, 96:190401, 2006.

[397] A. Micheli, G. K. Brennen, and P. Zoller. A toolbox for lattice-spin models with polar molecules. *Nature Physics*, 2:341–347, 2006.

[398] Alexey V. Gorshkov et al. Quantum magnetism with polar alkali-metal dimers. *Phys. Rev. A*, 84:033619, 2011.

[399] M. L. Wall, K. R. A. Hazzard, and A. M. Rey. *Quantum Magnetism with Ultracold Molecules*, pages 3–37. World Scientific Publishing Co, June 2015.

[400] Antoine Browaeys, Daniel Barredo, and Thierry Lahaye. Experimental investigations of dipole–dipole interactions between a few rydberg atoms. *J. Phys. B: At. Mol. Opt. Phys.*, 49:152001, 2016.

[401] Bo Yan, Steven A. Moses, Bryce Gadway, Jacob P. Covey, Kaden R. A. Hazzard, Ana Maria Rey, Deborah S. Jin, and Jun Ye. Observation of dipolar spin-exchange interactions with lattice-confined polar molecules. *Nature*, 501:521–525, 2013.

[402] Kaden R. A. Hazzard, Bryce Gadway, Michael Foss-Feig, Bo Yan, Steven A. Moses, Jacob P. Covey, Norman Y. Yao, Mikhail D. Lukin, Jun Ye, Deborah S. Jin, and Ana Maria Rey. Many-body dynamics of dipolar molecules in an optical lattice. *Phys. Rev. Lett.*, 113:195302, 2014.

[403] D. Banerjee, M. Dalmonte, M. Müller, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller. Atomic quantum simulation of dynamical gauge fields coupled to fermionic matter: From string breaking to evolution after a quench. *Phys. Rev. Lett.*, 109:175302, 2012.

[404] D. Banerjee, M. Bögli, M. Dalmonte, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller. Atomic quantum simulation of $\mathbf{U}(n)$ and $\mathrm{SU}(n)$ non-abelian lattice gauge theories. *Phys. Rev. Lett.*, 110:125303, Mar 2013.

[405] D C McKay and B DeMarco. Cooling in strongly correlated optical lattices: prospects and challenges. *Reports on Progress in Physics*, 74(5):054401, apr 2011.

[406] L. J. LeBlanc and J. H. Thywissen. Species-specific optical lattices. *Phys. Rev. A*, 75:053612, May 2007.

[407] D McKay and B DeMarco. Thermometry with spin-dependent lattices. *New Journal of Physics*, 12(5):055013, may 2010.

[408] Kaden R. A. Hazzard, Salvatore R. Manmana, Michael Foss-Feig, and Ana Maria Rey. Far-from-equilibrium quantum magnetism with ultracold polar molecules. *Phys. Rev. Lett.*, 110:075301, Feb 2013.

[409] Sylvain Schwartz, Alexander Keesling, Harry Levine, Ahmed Omran, Hannes Pichler, Soonwon Choi, Alexander S. Zibrov, Manuel Endres, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Probing many-body dynamics on a 51-atom quantum simulator. *Nature*, 551:579–584, 2017.

[410] Federica M. Surace, Paolo P. Mazza, Giuliano Giudici, Alessio Lerose, Andrea Gambassi, and Marcello Dalmonte. Lattice gauge theories and string dynamics in rydberg atom quantum simulators. *arXiv:1902.09551*, 2019.

[411] Simone Notarnicola, Mario Collura, and Simone Montangero. Real-time-dynamics quantum simulation of $(1+1)$-dimensional lattice qed with rydberg atoms. *Phys. Rev. Research*, 2:013288, Mar 2020.

[412] Roberto Verdel, Fangli Liu, Seth Whitsitt, Alexey V. Gorshkov, and Markus Heyl. Real-time dynamics of string breaking in quantum spin chains. *arXiv:1911.11382*, 2019.

[413] W. L. Tan, P. Becker, F. Liu, G. Pagano, K. S. Collins, A. De, L. Feng, H. B. Kaplan, A. Kyprianidis, R. Lundgren, W. Morong, S. Whitsitt, A. V. Gorshkov, and C. Monroe. Observation of domain wall confinement and dynamics in a quantum simulator. *arXiv:1912.11117*, 2019.

[414] Sylvain de Léséleuc, Daniel Barredo, Vincent Lienhard, Antoine Browaeys, and Thierry Lahaye. Optical control of the resonant dipole-dipole interaction between rydberg atoms. *Phys. Rev. Lett.*, 119:053202, Aug 2017.

[415] Jianming Cai, Alex Retzker, Fedor Jelezko, and Martin B. Plenio. A large-scale quantum simulator on a diamond surface at room temperature. *Nat. Phys.*, 9:168–173, 2013.

[416] B. Neyenhuis, B. Yan, S. A. Moses, J. P. Covey, A. Chotia, A. Petrov, S. Kotochigova, J. Ye, and D. S. Jin. Anisotropic polarizability of ultracold polar $^{40}\mathrm{K}^{87}\mathrm{Rb}$ molecules. *Phys. Rev. Lett.*, 109:230403, 2012.

[417] Loïc Anderegg, Lawrence W. Cheuk, Yicheng Bao, Sean Burchesky, Wolfgang Ketterle, Kang-Kuen Ni, and John M. Doyle. An optical tweezer array of ultracold molecules. *Science*, 365(6458):1156–1158, 2019.

[418] A. Einstein and W. J. de Haas. Experimenteller nachweis der ampéreschen molekularströme. *Verhandl. Deut. Phys. Ges.*, 17:152–170, 1915.

[419] R Winkler, S Papadakis, E De Poortere, and M Shayegan. *Spin-Orbit Coupling in Two-Dimensional Electron and Hole Systems*, volume 41. Springer, 2003.

[420] E. Rico, T. Pichler, M. Dalmonte, P. Zoller, and S. Montangero. Tensor networks for lattice gauge theories and atomic quantum simulation. *Phys. Rev. Lett.*, 112:201601, May 2014.

[421] Amodsen Chotia, Brian Neyenhuis, Steven A. Moses, Bo Yan, Jacob P. Covey, Michael Foss-Feig, Ana Maria Rey, Deborah S. Jin, and Jun Ye. Long-lived dipolar molecules and feshbach molecules in a 3d optical lattice. *Phys. Rev. Lett.*, 108:080405, Feb 2012.

[422] T. Pichler, M. Dalmonte, E. Rico, P. Zoller, and S. Montangero. Real-time dynamics in u(1) lattice gauge theories with tensor networks. *Phys. Rev. X*, 6:011023, Mar 2016.

[423] Gunnar S. Bali, Hartmut Neff, Thomas Düssel, Thomas Lippert, and Klaus Schilling. Observation of string breaking in qcd. *Phys. Rev. D*, 71:114513, Jun 2005.

[424] Boye Buyens, Jutho Haegeman, Henri Verschelde, Frank Verstraete, and Karel Van Acoleyen. Confinement and string breaking for qed_2 in the hamiltonian picture. *Phys. Rev. X*, 6:041040, Nov 2016.

[425] G. Magnifico, D. Vodola, E. Ercolessi, S. P. Kumar, M. Müller, and A. Bermudez. $\mathbb{Z}_N$ gauge theories coupled to topological fermions: qed_2 with a quantum mechanical θ angle. *Phys. Rev. B*, 100:115152, Sep 2019.

[426] Lorenzo Cardarelli, Sebastian Greschner, and Luis Santos. Deconfining disordered phase in two-dimensional quantum link models. *arXiv:1910.12829*, 2019.

[427] Luuk Coopmans, Di Luo, Graham Kells, Bryan K. Clark, and Juan Carrasquilla. Protocol discovery for the quantum control of majoranas by differentiable programming and natural evolution strategies. *PRX Quantum*, 2(2), Jun 2021.

[428] A Yu Kitaev. Unpaired majorana fermions in quantum wires. *Physics-Uspekhi*, 44(10S):131–136, oct 2001.

[429] A. Jamiołkowski. Linear transformations which preserve trace and positive semidefiniteness of operators. *Reports on Mathematical Physics*, 3(4):275–278, 1972.

[430] Man-Duen Choi. Completely positive linear maps on complex matrices. *Linear Algebra and its Applications*, 10(3):285–290, 1975.