

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM SCHOOL OF NATURAL SCIENCES

**Efficient simulations of quantum many-body dynamics  
with classical and quantum computers**

Sheng-Hsuan Lin

Vollständiger Abdruck der von der TUM School of Natural Sciences der Technischen Universität München zur Erlangung des akademischen Grades eines **Doktors der Naturwissenschaften (Dr. rer. nat.)** genehmigten Dissertation.

Vorsitz: Prof. Dr. Christian Pfeiderer

Prüfer\*innen der Dissertation:

1. Prof. Dr. Frank Pollmann
2. Prof. Dr. Barbara Kraus

Die Dissertation wurde am 28.06.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Natural Sciences am 26.10.2023 angenommen







# Abstract

Simulating quantum many-body dynamics is essential for understanding the emergent phenomena that arise due to the interplay of interactions and quantum fluctuations. Except for rare analytically tractable cases, classical numerical methods or quantum simulations are indispensable. This thesis explores variational approaches to study quantum many-body dynamics. We consider three classes of variational wavefunctions: tensor network states, neural-network quantum states, and sequential quantum circuit wavefunctions. First, we show that classical algorithms based on isometric tensor network states can efficiently simulate the dynamics of two-dimensional systems on finite and infinite strip geometries. Second, we study the complexity scaling of classically simulating quantum many-body dynamics with neural-network quantum states. Third, we confirm the complexity scaling of using sequential quantum circuits to simulate dynamics on a quantum computer and propose hybrid quantum-classical algorithms for simulating the dynamics of one-dimensional finite and infinite systems.



# Kurzfassung

Die Simulation der Quanten-Vielteilchendynamik ist für das Verständnis der emergenten Phänomene, die durch das Zusammenspiel von Wechselwirkungen und Quantenfluktuationen entstehen, unerlässlich. Abgesehen von wenigen analytisch zugänglichen Fällen sind klassische numerische Methoden oder Quantensimulationen unverzichtbar. In dieser Arbeit werden variationelle Ansätze zur Beschreibung der Quanten-Vielteilchendynamik untersucht. Wir betrachten drei Klassen von variationellen Wellenfunktionen: Tensornetzwerkzustände, neuronale Netzwerkquantenzustände und Zustände gegeben durch sequentielle Quantenschaltungen. Zunächst zeigen wir, dass klassische Algorithmen, die auf isometrischen Tensornetzwerkzuständen basieren, die Dynamik zweidimensionaler Systeme auf endlichen und unendlichen Streifengeometrien effizient simulieren können. Sodann untersuchen wir das Skalenverhalten der Komplexität von Dynamiksimulationen mit Quantenzuständen repräsentiert durch neuronale Netzwerke. Schließlich bestätigen wir die Komplexitätsskalierung der Verwendung sequentieller Quantenschaltungen zur Simulation der Dynamik auf einem Quantencomputer und schlagen hybride quantenklassische Algorithmen zur Simulation der Dynamik eindimensionaler endlicher und unendlicher Systeme vor.





# Publications

This thesis contains extended versions of published works in peer-reviewed journals and works uploaded to the open-access repository arXiv.

## Works contained in this thesis:

1. **Sheng-Hsuan Lin**, Michael Zaletel, and Frank Pollmann: “Efficient simulation of the Dynamics in Two-Dimensional Quantum Spin Systems with Isometric Tensor Networks”, [Physical Review B](#), **106(24)** (2022) [1].  
SHL performed the calculations and developed the DMRG<sup>2</sup> algorithm. This work has substantial overlap with and is built on top of MZ and FP’s previous work.
2. Yantao Wu, Sajant Anand, **Sheng-Hsuan Lin**, Frank Pollmann, Michael P. Zaletel: “Two Dimensional Isometric Tensor Networks on an Infinite Strip”, [Physical Review B](#) **107(24)** (2023) [2].  
YW initialized the project. YW and SA performed most of the calculations. SHL developed one iMM variant, proposed the bMPO evaluation, and performed the QMC calculation. YW, SA, and SHL wrote the manuscript. FP and MZ supervised the project.
3. **Sheng-Hsuan Lin** and Frank Pollmann: “Scaling of neural-network quantum states for time evolution”, [Physica Status Solidi B](#) **259(5)** (2021) [3].
4. **Sheng-Hsuan Lin**, Rohit Dilip, Andrew G. Green, Adam Smith, and Frank Pollmann: “Real and imaginary-time evolution with compressed quantum circuits”, [Physical Review X Quantum](#) **2(1)** (2021) [4].  
SHL and RD performed the calculation. AS, RD, and SHL wrote the manuscript. AG, AS, and FP conceived the project.
5. Nikita Astrakhantsev, **Sheng-Hsuan Lin**, Frank Pollmann, Adam Smith: “Time Evolution of Uniform Sequential Circuits”, [arXiv:2210.03751](#) (2022) [5].  
NA performed the calculations. SHL proposed the algorithm. NA and SHL wrote the manuscript. FP and AS initialized and supervised the project.

## Works not contained in this thesis:

1. Giuseppe Carleo, Kenny Choo, Damian Hofmann, James E.T. Smith, Tom Westerhout, Fabien Alet, Emily J. Davis, Stavros Efthymiou, Ivan Glasser, **Sheng-Hsuan Lin**, Marta Mauri, Guglielmo Mazzola, Christian B. Mendl, Evert van Nieuwenburg, Ossian O’Reilly, Hugo Théveniaut, Giacomo Torlai, Filippo Vicentini, and Alexander Wietek: “NetKet: a machine learning toolkit for many-body quantum systems” [SoftwareX](#) **10**, 100311, 2019 [6],
2. **Sheng-Hsuan Lin**, Olivier Kuijpers, Sebastian Peterhansl, and Frank Pollmann: “Distributive pre-training of generative modeling using matrix product states” [arXiv:2306.14787](#) Position paper of [QTNML Workshop, NeurIPS 2021](#) [7]

# Acknowledgements

I am deeply indebted to my supervisor, Frank Pollmann. His kindness, positivity, and delightful disposition made my academic journey a pleasurable experience. He is always approachable and willing to explain complex concepts in the simplest way possible. He is always curious about new research topics, giving me complete freedom to explore and, at the same time, providing his guidance to help me focus on important research topics. It is a pleasure to work with him, and I hope I can continue to learn from him.

I am grateful to my previous supervisors, Fabian Heidrich-Meisner, Lode Pollet, and Alexander Wolf, who introduced me to this fascinating field. Special thanks are due to Jheng-Wei Li and Chia-Min Chung for providing invaluable exchanges of ideas about tensor networks.

It has been an absolute honor and privilege to work with my collaborators. I extend my gratitude to Adam Smith for teaching me the importance of clear writing and presentation. Laurens Vanderstraeten, through his work on numerical methods, has been a constant source of inspiration. I am also grateful for the productive collaboration with Mike Zaletel's group. I particularly appreciate the dedication of Yantao Wu and Sajant Anand, with whom I spent countless nights perfecting our methodology and exploring new research directions. Finally, I thank Zhehao for his creative and insightful thoughts and patience.

I am glad to be a part of the TUM condensed matter theory group that provides a stimulating and friendly environment. Firstly, I would like to thank Raúl Morral Yepes for brightening our office with his incredible singing. Pablo Sala de Torres-Solanot, your dedication to teaching me physics has been invaluable. Markus Drescher, I greatly appreciate your assistance in helping me call the emergency hospital in German. And Julian Bibo, even though we never had the opportunity to publish together, you are the best office neighbor. Special thanks to Claudine Voelcker for all your administrative support and to Johannes Hauschild and Johannes Feldmeier for your invaluable IT assistance. To the other members of the group: Kévin, Elisabeth, Clemens, Bernhard, Joe, Leo, Yahui, Wonjune, Wilhelm, GiBaik, Nico, Melissa, and many others, the time spent with you in the group has always been enjoyable and refreshing. To my former group members whom I had the honor of sharing time with—Ruben Verresen, Tibor Rakovszky, Alex Schuckert, and Ananda Roy—I want to express my gratitude. I could have learned so much more from all of you. Thank you, Bhilahari, Umberto, Claudio, and Sergej, for organizing the Fly by Night Physics book club during the challenging first year of COVID.

My experience at AWS Center for Quantum Computing has greatly enriched my perspective and knowledge. I am immensely thankful to Kyungjoo Noh for affording me the opportunity to join this enriching environment. My thanks also go to Simon Lieu, Joe Iverson, Nikola Pancotti, Asmae Benhemou, Jonathan Conrad, Rachel, Joaquín, Ash, and Glen for their support.

I would like to express my appreciation to Pablo, Raúl, and Adam for their helpful feedback on parts of this thesis and to Pablo and Frank for their advice on thesis organization. Special thanks go to Sajant for dedicating time to read the full thesis and to Markus and Jakob for their help in translating the abstract into German.

On a more personal note, I would like to express my heartfelt thanks to Miguel, Jerome, David, Lamino, Stephanie, Maira, Julian, Anna, Paul, Vasily, Patricia, Franzy, Jiaqi, Ana, and many others for introducing me to the exhilarating world of dance and sharing their

rich cultures with me. The moments we shared, filled with music and laughter, have been indispensable and a true solace amidst the rigors of the Ph.D.

I owe all these years of wonderful experience to my parents and friends. Mom, your love and support have been invaluable. Dad, thank you for teaching me Math. Cheng-Hsin, my best friend, thank you for the love. This thesis is dedicated to you.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Kurzfassung</b>	<b>iii</b>
<b>Publications</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Contents</b>	<b>viii</b>
<b>1. Introduction</b>	<b>1</b>
<b>I. Algorithms for classical simulations</b>	<b>5</b>
<b>Overview of Part I</b>	<b>7</b>
<b>2. Review of tensor network methods</b>	<b>9</b>
2.1. TNSs and isoTNSs in 1D	10
2.2. TNSs in 2D	14
2.2.1. Boundary contraction	14
2.2.2. Algorithms for time evolution	16
2.2.3. Algorithms for energy minimization	17
2.3. IsoTNSs in 2D	18
2.3.1. Finite 2D isoTNSs	18
2.3.2. Variational Moses move	20
2.3.3. Sequential Moses move	23
2.3.4. TEBD <sup>2</sup> algorithm	26
2.3.5. Discussion	28
<b>3. Efficient simulation of dynamics with two-dimensional isometric tensor networks</b>	<b>31</b>
3.1. Extended study on Moses move	31
3.1.1. Sequential Moses move with alternative cost functions	31
3.1.2. Comparison of Moses moves	34
3.1.3. Implementation details	35
3.2. DMRG <sup>2</sup> algorithm	37
3.3. Spectral functions	40
3.3.1. Transverse field Ising model	41
3.3.2. Kitaev model on honeycomb lattice	42
3.4. Conclusion	43
<b>4. Isometric tensor network states on an infinite strip</b>	<b>45</b>
4.1. Isometric TNS on an infinite strip	46
4.2. Infinite Moses move	47
4.2.1. iMM-Local	48

4.2.2.	iMM-Polar	49
4.2.3.	iMM-MPO	50
4.2.4.	iMM-CG	51
4.2.5.	Error measures	52
4.3.	iMM benchmarks	53
4.3.1.	Benchmark for a single run of iMM	53
4.3.2.	Repeated application of iMM to an isoTNS	55
4.4.	Evaluation of local observables: Energy as an example	56
4.4.1.	Energy evaluated by iMM	56
4.4.2.	Energy evaluated by boundary MPO	57
4.4.3.	Energy benchmarks	57
4.5.	iTEBD <sup>2</sup>	58
4.5.1.	iTEBD <sup>2</sup> algorithm	58
4.5.2.	Ground state search	60
4.6.	Conclusions	60
<b>5.</b>	<b>Scaling of neural network quantum states for time evolution</b>	<b>63</b>
5.1.	Methods	63
5.1.1.	Neural-network quantum states	64
5.1.2.	Neural autoregressive quantum states	64
5.1.3.	Supervised learning for quantum state approximation	65
5.2.	Numerical results	67
5.2.1.	Approximation with MPSs	67
5.2.2.	Approximation with shallow NAQs	68
5.2.3.	Approximation with deep NAQs	71
5.2.4.	Approximation with RBMQs	72
5.3.	Conclusion	74
<b>II.</b>	<b>Algorithms for quantum simulations</b>	<b>75</b>
	<b>Overview of Part II</b>	<b>77</b>
<b>6.</b>	<b>Review of sequential quantum circuits</b>	<b>79</b>
6.1.	Gate-based quantum computation	79
6.2.	IsoTNSs as sequential circuits	80
6.3.	Quantum complexity	82
<b>7.</b>	<b>Real- and imaginary-time evolution with compressed quantum circuits</b>	<b>85</b>
7.1.	Compressed circuits	85
7.1.1.	Efficient representation of quantum states	87
7.2.	Variational time evolution algorithm	89
7.2.1.	Real time evolution	89
7.2.2.	Imaginary time evolution	91
7.2.3.	Simulation on QPU	92
7.3.	Discussion	94
<b>8.</b>	<b>Time evolution of uniform sequential circuits</b>	<b>97</b>
8.1.	Methodology	97
8.1.1.	The layered uniform sequential circuit ansatz	98
8.1.2.	Translationally invariant Trotterization	100
8.1.3.	The time evolution algorithm	101

## Contents

8.1.4. Required measurements . . . . .	102
8.2. Results . . . . .	104
8.2.1. Layered state unitary, exact environment . . . . .	104
8.2.2. Optimization with a layered environment . . . . .	106
8.2.3. Simulation on QPU . . . . .	107
8.3. Discussion . . . . .	108
<b>9. Conclusion and outlook</b>	<b>111</b>
9.1. Part I: Classical simulation with isometric tensor network states . . . . .	111
9.2. Part I: Classical simulation with neural-network quantum states . . . . .	113
9.3. Part II: Quantum simulation with sequential quantum circuits . . . . .	115
<b>A. Appendix IsoTNS</b>	<b>119</b>
A.1. Optimization of isometries . . . . .	119
A.1.1. Orthogonal Procrustes problem . . . . .	119
A.1.2. Evenly-Vidal algorithm . . . . .	120
A.1.3. Gradient descent algorithm . . . . .	120
A.2. Weighted trace gauge, internal correlation, and the corner double line tensors	121
A.3. Extra Data . . . . .	124
<b>B. Appendix inf IsoTNS</b>	<b>127</b>
B.1. Proof of Eq. (4.18) . . . . .	127
B.2. Isometric filling of $A$ . . . . .	128
B.3. Numerical results for $g = 3.04438$ . . . . .	129
<b>C. Appendix Neural Network</b>	<b>131</b>
C.1. Network architectures . . . . .	131
C.2. Review of neural autoregressive quantum states . . . . .	133
C.3. Cost functions and stochastic gradient descent . . . . .	134
C.4. Setup of optimizations . . . . .	136
C.5. Additional data . . . . .	137
<b>D. Appendix Circuit</b>	<b>145</b>
D.1. Types of ansätze . . . . .	145
D.2. Classical simulation algorithm for quantum circuit . . . . .	145
D.3. Non-unitary gates . . . . .	148
D.4. Cost of the imaginary-time evolution . . . . .	149
D.5. Detailed data for parameter counting . . . . .	150
D.6. Randomized circuits for QPU measurement . . . . .	151
<b>E. Appendix Inf Circuit</b>	<b>153</b>
E.1. Transfer matrix . . . . .	153
E.1.1. Transfer matrix of USC . . . . .	153
E.1.2. Evaluating local observables . . . . .	153
E.1.3. Derivative of the transfer matrix . . . . .	154
E.1.4. Post-selection probability . . . . .	155
E.2. Details of classical optimization of l-USC . . . . .	155
E.2.1. Unitary parametrization and reunitarization . . . . .	155
E.2.2. Derivative with respect to a gate . . . . .	156
E.2.3. Environment optimization . . . . .	156
E.2.4. Details of gradient descent method and stopping criteria . . . . .	156
E.3. Sufficiency of the $M_U = 1$ state-unitary decomposition . . . . .	156

E.4. The accumulated error $\mathcal{M}$ measure . . . . .	157
E.5. Environment representation complexity . . . . .	158
E.6. Randomized circuits for QPU measurement . . . . .	159
E.7. The equivalence of fixed points . . . . .	159
<b>Acronyms</b>	<b>162</b>
<b>Bibliography</b>	<b>165</b>





# 1. Introduction

Quantum many-body systems are well-known for their rich emergent behaviors arising from the interactions between a large number of degrees of freedom. Prominent examples are quantum spin liquids that give rise to fractionalized excitations [8, 9] and high- $T_c$  superconductivity [10]. Driven by the rapid experimental progress, e.g., cold atoms and trapped ions, with unprecedented levels of control and long coherence times [11–16], even more unexpected novel phenomena can emerge out of equilibrium. These include many-body localization [17–21], quantum many-body scars [22–24]. The underlying theme in all these discoveries is the study of quantum thermalization and its breakdown [25], which is at the intersection of condensed matter, quantum information, and statistical physics and has application in engineering quantum devices [26, 27].

*How do we study quantum many-body systems?* On the one hand, the emergent nature of these fascinating behaviors means that the systems can be qualitatively different with a larger number of degrees of freedom. As a result, we would like to study  $N$ -body systems where  $N$  is as large as possible. On the other hand, interactions play a central role. Therefore, we are interested in methods that could solve all coupling regimes, even when perturbation theories break down. There are few exactly solvable models that give us valuable insights. More often, we turn to the help of numerical simulations. Below we briefly give an overview of wavefunction-based numerical approaches<sup>1</sup> for studying many-body ground states and dynamics, and we comment on their connection to excited states and finite temperature.

Consider a pure state describing a quantum many-body system

$$|\psi\rangle = \sum_{\sigma_1, \sigma_2, \dots, \sigma_N} \Psi^{\sigma_1 \sigma_2 \dots \sigma_N} |\sigma_1\rangle \otimes |\sigma_2\rangle \otimes \dots \otimes |\sigma_N\rangle, \quad (1.1)$$

where  $\{|\sigma_i\rangle \in \mathcal{H}_i\}$  are the local basis states. The full Hilbert space is the tensor product of a set of local Hilbert spaces  $\mathcal{H} = \otimes_i^N \mathcal{H}_i$ . Each is of dimension  $d_i$ . The order- $N$  coefficient tensor  $\Psi^{\sigma_1 \sigma_2 \dots \sigma_N}$  contains all the information of the state, but the number of parameters  $\mathcal{O}(\prod_{i=1}^N d_i)$  scales exponentially with the system size  $N$ . Nevertheless, for a small system, we can represent the quantum state exactly and use standard linear algebra algorithms to solve quantum many-body problems. Widely used are exact diagonalization (ED) and Krylov-subspace methods, which simulate ground states and dynamics exactly. These serve as an important benchmark for developing new numerical methods and are a reliable method for studying systems with unknown properties. However, the system size we can address is strongly limited due to the exponential growth of the many-body Hilbert space.

## Wavefunction-based methods

Variational wavefunction-based methods are a family of methods overcoming the exponential scaling by parameterizing the wavefunction with fewer parameters. The choice

---

<sup>1</sup>There are various numerical methods not based on wavefunctions. One broad class is the variational Monte Carlo (VMC) methods [28]. The stochastic series expansion is an example that computes the path integral in imaginary time by Monte Carlo sampling. In general, QMC methods provide efficient simulations for finite temperatures and ground states properties of one-dimensional (1D) and higher dimensional quantum systems. However, QMC methods are restricted by the infamous sign problem [29] when the systems involve fermionic degrees of freedom or geometric frustration, and from these methods, it is hard to obtain dynamical properties as the numerical analytic continuation is often ill-conditioned.

and design of variational wavefunctions depend on prior knowledge about the physical states. Below we consider three classes of variational wavefunctions: (i) tensor network states (TNSs), (ii) neural-network quantum states (NQSs), (iii) sequential quantum circuit wavefunctions.

Tensor-network methods [30–35] approximate wavefunctions with networks of tensors, where contracting all tensors together recovers the wavefunction. The decomposition of a global wavefunction into local tensors utilizes the locality of the problem. Tensor network methods are not limited by the system size but by the entanglement [30, 36]. Therefore, ground states of local gapped Hamiltonians that satisfy the entanglement area-law [37–39]<sup>2</sup> can be efficiently simulated<sup>3</sup>. For 1D systems, tensor networks provide arguably the most powerful numerical method in finding ground states and simulating dynamics using matrix-product state (MPS) [45, 46] methods, such as the density matrix renormalization group (DMRG) algorithm [31, 47] and the time-evolving block decimation (TEBD) algorithm [48, 49]. Nevertheless, the application of MPSs to study non-equilibrium dynamics [48–54] is strongly limited by the fast growth of entanglement [55, 56]. In certain cases, non-equilibrium phenomena are accessible even at long times using additional approximations [57–59]. For 2D systems, TNSs [60–63] carry similar successes and limitations as MPSs in 1D. Ground states and dynamics problems can be solved using different methods, including full update [64, 65], gradient update [66–68], variational Monte Carlo (VMC) [69–71], and more. The main difference between 1D TNSs and 2D TNSs are the lack of canonical form and the non-trivial boundary contraction [60, 64, 72–74] in 2D TNSs. Such difference motivates the proposal of isometric tensor network states (isoTNSs) [75], which are TNSs consisting of only isometric tensors. The isometric structure in isoTNSs gives rise to an efficient time evolution algorithm and a principled variational energy optimization algorithm, which are the main topics discussed in Part I of the thesis.

A promising new approach to simulate the ground states and dynamics of many-body systems is based on a representation of quantum states using artificial neural networks (ANNs) [76]. Neural networks are parameterized functions that take strings representing the computational basis  $\{\sigma_1, \dots, \sigma_N\}$  as input and output a complex number  $f(\sigma_1, \dots, \sigma_N)$  corresponding to the quantum amplitude. Recent works have shown not only promising results for simulating non-equilibrium dynamics [77] but also in describing 2D critical ground states [78] and states with chiral topological order [79]. Importantly, states based on ANNs can efficiently encode volume law entanglement [80, 81] and are thus per se not limited by the entanglement growth in non-equilibrium systems. While the mathematical foundation and limitations of neural networks are active fields of research [82], less is known in the context of representing quantum many-body states. In Chapter 5, we study the open question on the scaling of representing states under quench dynamics.

In recent years, the rapid development of universal quantum computers [83, 84] has led to the era of noisy intermediate scale quantum (NISQ) devices [85]. NISQ devices contain order 50 qubits and give access to hundreds of quantum gate operations [86]. The noise currently precludes implementing many quantum algorithms [87, 88] and motivates the consideration of quantum circuits directly as a variational ansatz for studying ground states [89] and dynamics [90–96]. The premise is that quantum circuits can be efficiently executed on a quantum computer and are hard to simulate on a classical computer, so it could potentially give us an advantage by treating a quantum computer as an accelerated device to execute the quantum circuit. Whether or not we would see a quantum advantage

---

<sup>2</sup>While it is proven that the ground states of 1D local gapped Hamiltonian have area-law entanglement [37] and MPSs are faithful representations [40], it remains an open question in two-dimensional (2D) [38, 41] and is only proven with additional conditions [38, 42, 43], e.g., non-interacting or frustration-free Hamiltonian.

<sup>3</sup>This is rigorously proven [44] in 1D. In practice, tensor network methods work well for 2D problems. We discuss this in more detail in Chapter 2.

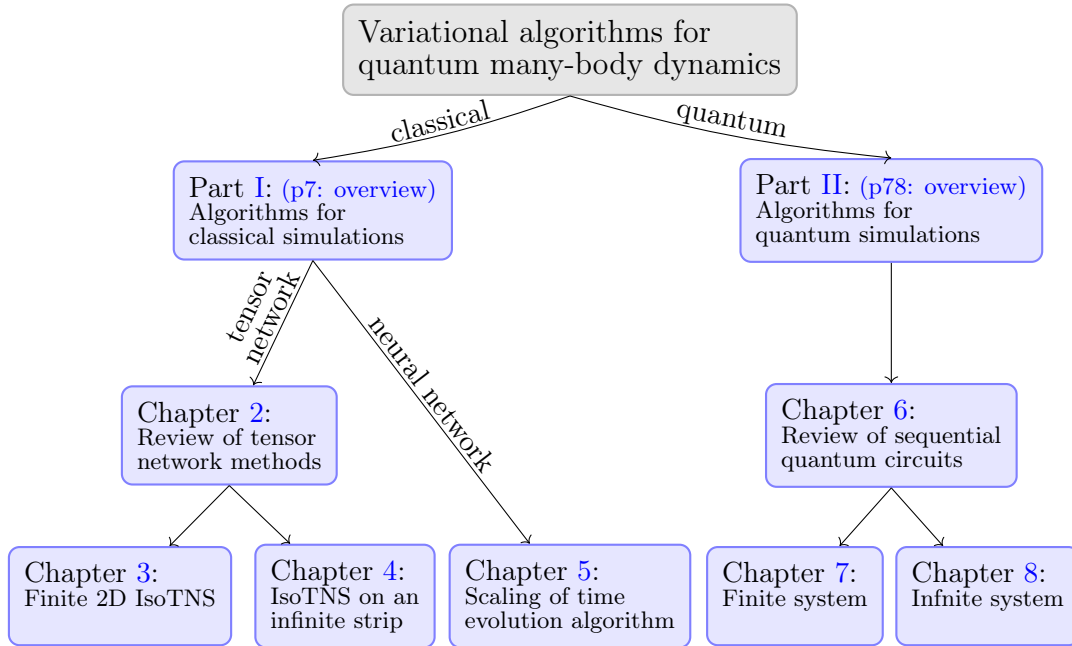


Figure 1.1.: A roadmap through the chapters of the thesis.

for ground state problems is under debate [97]. A more promising application is the simulation of dynamics [98]. We expand the discussion in Chapter 6 and review the equivalence of isoTNSs and sequential quantum circuits. We propose variational algorithms for time evolution and study the properties of sparse sequential quantum circuits as variational ansätze to represent states under quantum quenches in Part II of the thesis.

The main goal of the thesis is to understand the limitation of current numerical methods and develop new numerical methods for quantum many-body problems. While we focus on algorithms for the simulation of ground states and quantum dynamics, these algorithms are also the building blocks of algorithms to study excited states or finite temperature physics. We can use imaginary-time evolution with purified states [99–101] or the minimally entangled typical thermal states [102] to study finite temperature physics. The real-time evolution algorithms could be used to obtain spectral functions, i.e., dynamical structure factors [103–105], and compute physical observables at finite energy [106–108].

### The organization of this thesis:

An interconnection of the chapters of this thesis is shown in Fig. 1.1. We separate the thesis into two parts with overviews on pages 7 and 73, respectively. Part I studies the classical algorithm for simulating quantum many-body problems. Chapter 2 gives a review of TNSs and isoTNSs algorithms in the literature. Chapter 3 generalizes the DMRG algorithm for isoTNSs in 2D and perform the real-time evolution algorithm to compute the spectral functions. Chapter 4 generalizes isoTNSs to an infinite strip geometry. Part II studies the quantum algorithm for simulating quantum many-body problems. We give a brief review of quantum computing and the connection of isoTNSs and sequential circuits in Chapter 6. We study the representation power of finite sparse sequential circuits and infinite (uniform) sparse sequential circuits in Chapter 7 and Chapter 8 respectively.



Part I.

# Algorithms for classical simulations



# Overview of Part I

As mentioned in the introduction, Part I of this thesis focuses on the classical algorithms for simulating quantum many-body dynamics.

## Chapter 2: Review of tensor network methods

Chapter 2 provides a review of tensor network methods from the literature, focusing on one-dimensional (1D) and two-dimensional (2D) tensor network states (TNSs) and isometric tensor network states (isoTNSs). We begin with a quick refresh of the notation and the basics of tensor networks and set up the central idea for the following chapters by identifying the matrix-product states (MPSs) in isometric form as an example of 1D isoTNSs. We describe the generic properties of the isoTNSs and show that the isometric condition is crucial for efficient numerical algorithms.

We then briefly review 2D TNSs, which differ from 1D TNSs due to the nontrivial contraction. We review boundary contraction algorithms, which are used in the next chapters. Utilizing the contraction algorithms, we see that the 1D algorithms for time evolution and energy minimization can be generalized to 2D with slightly higher complexity. In the last part of the chapter, we review the pioneering work [75] on 2D isoTNSs, showing a different route to address 2D tensor network problem by imposing isometric form. This avoids the need for tensor network contraction and instead utilizes the algorithm for shifting isometric form, i.e., the Moses move (MM) algorithm. As an application, the time-evolving block decimation (TEBD) algorithm is generalized to 2D, termed TEBD<sup>2</sup> algorithm.

## Chapter 3: Efficient simulation of dynamics with two-dimensional isometric tensor networks

Chapter 3 explores the efficient simulation of dynamics with 2D isometric tensor networks. We extend the study of the MM algorithm, studying the effect of alternative cost functions on the tripartite decomposition in MM and proposing a two-column variational MM algorithm. We further generalize the 1D variational energy minimization algorithm, i.e., density matrix renormalization group (DMRG) algorithm to 2D, dubbed DMRG<sup>2</sup> algorithm, and benchmark its performance by comparing it to imaginary time evolution using TEBD<sup>2</sup>. We then revisit the TEBD<sup>2</sup> and explore its power for the real-time evolution of 2D lattice systems. Using TEBD<sup>2</sup> and DMRG<sup>2</sup>, we demonstrate that we can simulate the dynamics of locally perturbed 2D quantum many-body ground state, which give us access to the spectral functions, i.e., dynamical spin structure factors. We focus on two paradigmatic models: First, we compare our results for the transverse field Ising model on a square lattice with the prediction of the spin-wave theory. Second, we consider the Kitaev model on the honeycomb lattice and compare it to the result from the exact solution.

## Chapter 4: Isometric tensor network states on an infinite strip

Tensor networks have been shown to provide a powerful way to study quantum many-body systems in the thermodynamic limit since their introduction. Chapter 4 attempts to further complete the study of 2D isoTNSs by generalizing finite 2D isoTNSs to states on a finite by infinite geometry, i.e., an infinite strip. The essential step is the generalized infinite Moses

move (iMM) algorithm. The chapter explores various variations of the algorithm, including iMM-Local, iMM-Polar, iMM-MPO, and iMM-CG. It also discusses error measures and presents benchmarks to evaluate the performance of these different variant algorithms. As the application of the iMM algorithms, we demonstrate that the local observables can be evaluated efficiently. Finally, we introduce an infinite time-evolving block decimation algorithm (iTEBD<sup>2</sup>) and use it to approximate the ground state of the 2D transverse field Ising model on lattices of infinite strip geometry.

## Chapter 5: [Scaling of neural network quantum states for time evolution](#)

Chapter 5 delves into neural-network quantum states (NQSs). As a new emerging numerical technique for simulating quantum many-body systems, their limitation on simulating quantum dynamics remains unknown. In this work, we provide insight into the scaling of the number of parameters required based on numerical results of simulating global quench dynamics of a non-integrable 1D quantum Ising chain.

After briefly reviewing the NQSs and neural autoregressive quantum states (NAQSs), we explore the supervised learning framework for quantum state approximation with stochastic sampling and stochastic gradient descent. We propose cost functions for joint learning magnitudes and phases with NAQSs, ensuring an unbiased stochastic gradient. We confirm the correctness of the framework by comparing numerical results of state approximation by supervised learning with MPSs and by the standard MPS variational algorithm.

The bulk part of the chapter focuses on approximating states undergoing global quench dynamics of a non-integrable quantum Ising chain. We present numerical results for approximating such time-evolved states by shallow NAQSs, deep NAQSs, and restricted Boltzmann machine quantum states. We find that the number of parameters required to represent the quantum state at a given accuracy increases exponentially in time. The growth rate is only slightly affected by the network architecture over a wide range of different design choices: shallow and deep networks, small and large filter sizes, and dilated and normal convolutions, with and without shortcut connections.



## 2. Review of tensor network methods

In this chapter, we review the properties and algorithms for one-dimensional (1D) and two-dimensional (2D) tensor network states (TNSs) and isometric tensor network states (isoTNSs) from previous works in the literature.

As discussed in the introduction, tensor networks are efficient variational wavefunctions for studying quantum many-body systems. Consider a quantum many-body system described by the pure state

$$|\psi\rangle = \sum_{\sigma_1, \sigma_2, \dots, \sigma_N} \Psi^{\sigma_1 \sigma_2 \dots \sigma_N} |\sigma_1\rangle \otimes |\sigma_2\rangle \otimes \dots \otimes |\sigma_N\rangle, \quad (2.1)$$

where  $\{|\sigma_i\rangle \in \mathcal{H}_i\}$  are the local basis states. Tensor network methods approximate the order- $N$  coefficient tensor  $\Psi^{\sigma_1 \sigma_2 \dots \sigma_N}$  by low-rank tensor decompositions [34]. Usually tensor networks, for example, matrix-product states (MPSs) in 1D and TNSs in 2D, have a connectivity resembling the underlying lattices. The success of tensor network methods is due to the locality of the physical system. Roughly speaking, tensor networks are efficient representations for area-law entangled states with  $\text{poly}(N)$  number of parameters. In 1D, all states with area-law entanglement can be expressed as MPSs with a system size independent bond dimension [40, 109]. In contrast for  $D \geq 2$ , TNSs capture only part of the area-law states [110] but are believed to capture ground states of local gapped Hamiltonian [41]<sup>1</sup>.

### Tensor network diagram

Tensor networks are defined through diagrams consisting of nodes and edges representing tensors. An order- $k$  tensor is represented by a node with  $k$  edges. The example for  $k = 0, 1, 2, 3$  are given below,

$$\boxed{s} = s, \quad \begin{array}{c} i \\ | \\ \boxed{v} \end{array} = v_i, \quad i - \boxed{M} - j = M_{ij}, \quad \begin{array}{c} j \\ | \\ i - \boxed{T} - k \end{array} = T_{ijk}, \quad (2.2)$$

where from left to right we have a scalar, a vector, a matrix, and an order-3 tensor.

Since the spaces of tensor,  $\mathbb{C}^{\prod_{i=1}^N d_i}$  and  $\mathbb{C}^{\prod_{i=1}^M \tilde{d}_i}$ , are isomorphic as long as the overall dimensions match, i.e.,  $\prod_{i=1}^N d_i = \prod_{i=1}^M \tilde{d}_i$ , this means we can group the indices or split the indices and change the order of the tensor. It is common to first transform a tensor into a matrix, i.e., matricization, by grouping the indices of the tensor and then apply linear algebra operation on the matrix. The following discussion regarding matrices can also be applied to higher-order tensors by grouping the indices.

If two nodes share one edge, the index over the corresponding edge is summed over and we can group the two nodes into one. For example, the matrix-matrix multiplication can be represented as,

$$i - \boxed{M} - \begin{array}{c} j \\ | \\ \boxed{N} \end{array} - k = \sum_j M_{ij} N_{jk} = (MN)_{ik} = i - \boxed{MN} - k, \quad (2.3)$$

<sup>1</sup>There were doubts on whether TNSs can represent non-critical chiral topological states due to the no-go theorem proven in the case of free fermions [111–115]. Recent numerical study shows positive supporting evidence for the use of TNSs to represent chiral topological states [116].

where the intermediate index  $j$  is summed over. This operation is known as tensor contraction.

Conversely, we can split one node into two nodes sharing one edge, for example through matrix decomposition. An example of QR decomposition is given below.

$$M_{ik} = i \text{---} \boxed{M} \text{---} k = i \text{---} \boxed{Q} \text{---}^j \text{---} \boxed{R} \text{---} k = \sum_j Q_{ij} R_{jk}. \quad (2.4)$$

Special care should be taken into account to distinguish the difference between the original Hilbert space, the ket space, and the dual Hilbert space, the bra space. This restricts the operations we can perform on the tensors. For example, we should not contract two ket vectors directly or group the ket index with the bra index directly. Some people use the convention of drawing incoming or outgoing arrows on the edges to distinguish the ket and the bra. However, here we use a different convention where the arrow directions are used for denoting the isometric direction as discussed below.

To sum up, tensor networks take advantage of the inherent locality structure within the quantum wavefunction. Instead of dealing with a single, large coefficient tensor, the tensor network decomposes it into a series of smaller tensors linked by contractions. This approach results in a network of interconnected tensors, hence giving the name – “tensor network”. It is the systematic exploitation of these local structures that facilitates the development of more efficient algorithms. In this thesis, we explore a specific subclass of tensor networks composed of isometric tensors. By constraining our focus on isometric tensors, we introduce additional mathematical structure into the network that leads to efficient algorithms in higher dimensions.

### Notation for isometries and isometric tensors

An isometry is a linear map  $W : V_s \rightarrow V_l$  from a smaller vector space  $V_s$  to a larger vector space  $V_l$ , such that  $W^\dagger W = \mathbb{1}$ ,  $W W^\dagger = \mathcal{P}_{V_s}$ , where  $\mathcal{P}_{V_s}$  is the projection operator to the vector space  $V_s$ . Isometric tensors are tensors that by grouping the indices, i.e. matricization, become isometries. We consider a convention as shown below. In particular, for isometric tensors, we draw the indices belonging to the larger dimensions as incoming arrows and the indices for smaller dimensions as outgoing arrows, for example,

$$\begin{array}{c}
 \begin{array}{c} \curvearrowright \\ \curvearrowright \\ \curvearrowright \end{array} \text{---} \boxed{A} \text{---} k \\
 \uparrow j \\
 \boxed{A^*} \text{---} k' \\
 \begin{array}{c} \curvearrowleft \\ \curvearrowleft \\ \curvearrowleft \end{array}
 \end{array}
 = \sum_{ij} A_{ijk} A_{ijk'}^* = \delta_{kk'} = \begin{array}{c} \curvearrowright^k \\ \cdot \\ \curvearrowleft^{k'} \end{array}. \quad (2.5)$$

For general (non-isometric) tensors, we draw the indices without arrows. Throughout the thesis, we refer to isometric tensor networks as tensor networks consisting of isometric tensors whose edges can consistently be assigned arrows.

## 2.1. TNSs and isoTNSs in 1D

In this section, we review the basic properties of the simplest tensor network in 1D, i.e., MPSs. We identify the MPSs with left/right-normalized tensors as 1D isoTNSs. For more detailed reviews for MPSs and general TNSs, we refer the readers to [31–33, 54].

An MPS is defined as

$$|\psi\rangle = \sum_{\sigma} \sum_{m_1, m_2, \dots} T_{m_0, m_1}^{\sigma_1} T_{m_1, m_2}^{\sigma_2} \cdots T_{m_{N-1}, m_N}^{\sigma_N} |\sigma\rangle, \quad (2.6)$$

where the  $T$  are general tensors of order three and the  $m_i$  are indices in virtual space between site  $i$  and  $i + 1$ . The bond dimension  $D$  is the dimension of the virtual space, such that  $m_i = 1, \dots, D$ . Carrying out the summation  $\sum_{m_1, m_2, \dots}$ , i.e. contraction, over the virtual indices explicitly gives the coefficient  $\Psi^{\sigma_1 \sigma_2 \dots \sigma_N}$ . The number of parameters of MPSs scales as  $\mathcal{O}(ND^2)$ , which avoids the exponential scaling in the system size.

One can always transform the MPS tensors into the left/right-normalized form without loss of generality. Importantly, the left/right normalization of the tensors simplifies and stabilizes MPS-based algorithms. A tensor  $A_{m_{i-1}, m_i}^{\sigma_i}$  is left-normalized if it satisfies the isometric condition,

$$\sum_{\sigma_i m_{i-1}} A_{m_{i-1}, m_i}^{\sigma_i} \left( A_{m_{i-1}, m_i}^{\sigma_i} \right)^* = \mathbb{1}_{m_i, m_i}. \quad (2.7)$$

Similarly a tensor  $B_{m_{i-1}, m_i}^{\sigma_i}$  is right-normalized if it satisfies the isometric condition,

$$\sum_{\sigma_i m_i} B_{m_{i-1}, m_i}^{\sigma_i} \left( B_{m_{i-1}, m_i}^{\sigma_i} \right)^* = \mathbb{1}_{m_{i-1}, m_{i-1}}. \quad (2.8)$$

The tools to bring MPSs tensors to the left/right-normalized form are orthogonal matrix decompositions, i.e., QR decomposition and singular value decomposition (SVD). Given an MPS with general tensors, we can bring all tensors successively into the left/right-normalized form by successive SVDs or QR decompositions. For example, one can start from the left with QR decomposition  $T_{m_0, m_1}^{\sigma_1} = \sum_{m_{1'}} A_{m_0, m_{1'}}^{\sigma_1} R_{m_{1'}, m_1}$ , and absorb the  $R$  tensor into the nearby  $T$  tensor,  $\sum_{m_1} R_{m_{1'}, m_1} T_{m_1, m_2}^{\sigma_2} = \tilde{T}_{m_{1'}, m_2}^{\sigma_2}$ . Now the original  $T_{m_0, m_1}^{\sigma_1}$  tensor becomes a left-normalized tensor  $A_{m_0, m_{1'}}^{\sigma_1}$ . Iteratively, one can *exactly* bring all tensors into the left-normalized form. Similarly, one can start from the right and move left and end up with all tensors being right-normalized.

### Gauge degrees of freedom

The fact that we can always bring tensors in an MPS into left/right-normalized form is related to the *gauge degrees of freedom* of an MPS. The *gauge degrees of freedom* of an MPS, and more generally for a tensor network, is the freedom to insert a pair of invertible matrix and its inverse  $XX^{-1}$  to the non-external edge (virtual bond) of the tensor network. By separately absorbing the  $X$  and  $X^{-1}$  to the nearby two tensors, we modify the two tensors in the MPS while leaving the overall physical state unchanged. Therefore, given a physical state, its tensor network representation is not unique. This is known as the gauge degree of freedom in the tensor networks. In the case of MPSs, this gauge freedom is sufficient to bring the tensors into left/right-normalized form. However, this does not apply to TNSs in higher dimensions.

The left/right-normalized form does not fully fix the gauge freedom for the MPSs. There is the freedom to insert pairs of unitaries and their adjoint to the virtual bond, which preserve the isometric form. We will introduce the canonical form below, which fully fixes the gauge of degrees of freedom.

### Orthogonality center

The orthogonality center of an MPS is a single bond or a region of sites such that to the left of the center all tensors are left-normalized and to the right, all are right-normalized (note that this does not have to be the geometric center of the chain). By the combination of both moves mentioned above, one can *exactly* obtain an MPS with normalized tensors and orthogonality center at any desired bond or region.

For example, following the normalization procedure by QR decomposition mentioned above from both ends of the MPS inwards, we have the following decomposition:

$$\Psi^{\sigma_1 \cdots \sigma_N} = \sum_{\{m_i\}} A_{m_1}^{\sigma_1} \cdots A_{m_{l-1}, m_l}^{\sigma_{l-1}} R_{m_l, m_{l'}} B_{m_{l'}, m_{l+1}}^{\sigma_l} \cdots B_{m_{N-1}}^{\sigma_N}. \quad (2.9)$$

By definition,  $R_{m_l, m_{l'}}$  is the orthogonality center on bond- $l$ . Note that, the orthogonality center on bond- $l$  can be a general matrix without the restriction of upper-triangular form. From this point on, we denote such general matrix by  $\Psi_{m_l, m_{l'}}$ , which is also known as the 0-site wavefunction.

We can obtain the MPS with orthogonality center on a single site by merging the tensors  $\sum_{m_{l'}} \Psi_{m_l, m_{l'}} B_{m_{l'}, m_{l+1}}^{\sigma_l} = \Psi_{m_l, m_{l+1}}^{\sigma_l}$ , leading to

$$\Psi^{\sigma_1 \cdots \sigma_N} = \sum_{\{m_i\}} A_{m_1}^{\sigma_1} \cdots A_{m_{l-1}, m_l}^{\sigma_{l-1}} \Psi_{m_l, m_{l+1}}^{\sigma_l} B_{m_{l+1}, m_{l+2}}^{\sigma_{l+1}} \cdots B_{m_{N-1}}^{\sigma_N}. \quad (2.10)$$

$\Psi_{m_l, m_{l+1}}^{\sigma_l}$  is the orthogonality center on site- $l$  and a single-site wavefunction. We can move the orthogonality center forward keeping constant bond dimension by repeating orthogonal matrix decompositions and merging tensors.

### Isometric boundary map

The tensors excluding the orthogonality center, for example  $\Psi_{m_l, m_{l+1}}^{\sigma_l}$  in Eq. (2.10), are a collection of isometries

$$\{A_{m_1}^{\sigma_1}, \cdots, A_{m_{l-1}, m_l}^{\sigma_{l-1}}, B_{m_{l+1}, m_{l+2}}^{\sigma_{l+1}}, \cdots, B_{m_{N-1}}^{\sigma_N}\}.$$

Contracting all the internal virtual indices, they form a single isometry  $T^{V \leftarrow \partial V}$ . In other words, the boundary map  $T^{V \leftarrow \partial V}$  of the orthogonality center  $\Psi_{m_l, m_{l+1}}^{\sigma_l}$  is an isometry mapping from the virtual space  $\partial V$  to physical Hilbert space  $V$ . Similarly, the boundary map of the orthogonality center on bond- $l$  is also isometric. Because the boundary map is isometric, we can interpret the orthogonality center itself as the wavefunction in the lower dimensional space. For example, the  $\Psi_{m_l, m_{l'}}$  can be interpreted as a zero-dimensional (0D) representation of the state in terms of orthogonal states  $|m_l\rangle$  and  $|m_{l'}\rangle$ , i.e.,

$$\begin{aligned} |\psi\rangle &= \sum_{\sigma} \sum_{\{m_i\}} (A_{m_1}^{\sigma_1} \cdots A_{m_{l-1}, m_l}^{\sigma_{l-1}} |\sigma_1, \dots, \sigma_{l-1}\rangle) \Psi_{m_l, m_{l'}} (B_{m_{l'}, m_{l+1}}^{\sigma_l} \cdots B_{m_{N-1}}^{\sigma_N} |\sigma_l, \dots, \sigma_N\rangle) \\ &= \sum_{m_l, m_{l'}} \Psi_{m_l, m_{l'}} |m_l\rangle |m_{l'}\rangle. \end{aligned} \quad (2.11)$$

Since the isometric map is norm-preserving, the truncation on  $\Psi_{m_l, m_{l'}}$  based on an SVD is not only optimal for  $\Psi_{m_l, m_{l'}}$  in  $L^2$  norm but also for  $\Psi^{\sigma_1 \cdots \sigma_N}$  in  $L^2$  norm.

Moreover, we define the norm tensor  $N_l$  with respect to site- $l$  as the contraction of the norm  $\langle \Psi | \Psi \rangle$  but leaving out tensors on site- $l$ , e.g.,  $\Psi_{m_l, m_{l+1}}^{\sigma_l}$  and  $(\Psi_{m_l, m_{l+1}}^{\sigma_l})^*$ . In other words, it is the contraction of the boundary map with its complex conjugation,  $(T^{V \leftarrow \partial V})^\dagger T^{V \leftarrow \partial V}$ . Because of the isometric condition of the boundary map, the norm tensor  $N_l = (T^{V \leftarrow \partial V})^\dagger T^{V \leftarrow \partial V} = \mathbb{1}_{\partial V}$  with respect to the orthogonality center is an identity operator.

From the above we see that an isoTNS in 1D has the following properties: It is a tensor network composed of isometries, as from Eq. (2.7) and (2.8), and all the tensors excluding the orthogonality center form an isometric boundary maps in both Eq. (2.9) and (2.10).

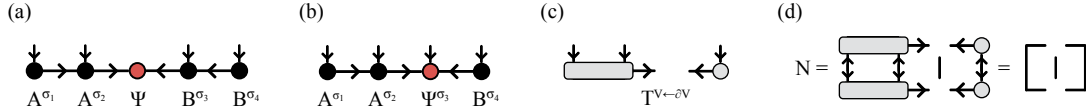


Figure 2.1.: (a) An MPS with orthogonality center on a bond and (b) on a single site. (c) The boundary map  $T^{V \leftarrow \partial V}$  of the MPS (b). (d) The norm tensor  $N$  of the orthogonality center  $\Psi$  of the MPS (b).

### Efficient tensor network algorithms in 1D

Coincidentally, the invention of the density matrix renormalization group (DMRG) algorithm [47] and MPSs [45, 117] happen in the same year 1992. The connection between the DMRG algorithm and the underlying tensor network structure [118], which gives DMRG algorithm the interpretation as a variational algorithm over MPS wavefunction [119], is not well appreciated by the community back then. The understanding of the underlying tensor network structure naturally leads to the further development of powerful algorithms, for example, the time-evolving block decimation (TEBD) algorithm [48, 49] for solving time evolution and its generalization to higher dimension [60].

The finite DMRG algorithm is an iterative algorithm updating tensors  $\Psi$  such that  $\Psi$  satisfies the stationary condition for an extremum.

$$\begin{aligned} \partial_{\Psi^*} \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} &= 0 \\ \langle \partial_{\Psi^*} \psi | H | \psi \rangle - \langle \partial_{\Psi^*} \psi | \psi \rangle &= \langle \partial_{\Psi^*} \psi | H | \partial_{\Psi} \psi \rangle \Psi - \langle \partial_{\Psi^*} \psi | \partial_{\Psi} \psi \rangle \Psi \\ &= H_{\text{eff}} \Psi - N \Psi = 0 \end{aligned} \quad (2.12)$$

That is we update  $\Psi$  by solving the lowest eigenvector of the eigenvalue problem  $H_{\text{eff}} \Psi = N \Psi$ . The TEBD algorithm performs time evolution by applying Trotterized 2-site gate  $e^{-idtH_i}$  on the two-site orthogonality center  $\Psi$  and then truncates the bond dimension by SVD. For a detailed review of the algorithms, we refer the reader to the review article [31].

### Isometric form and canonical form

The canonical form of MPSs imposes an additional condition aside from the isometric condition [46],

$$\sum_{\sigma_i m_i} A_{m_{i-1}, m_i}^{\sigma_i} \rho_{m_i, m_i}^{(i)} \left( A_{m'_{i-1}, m_i}^{\sigma_i} \right)^* = \rho_{m_{i-1}, m'_{i-1}}^{(i-1)} \quad (2.13)$$

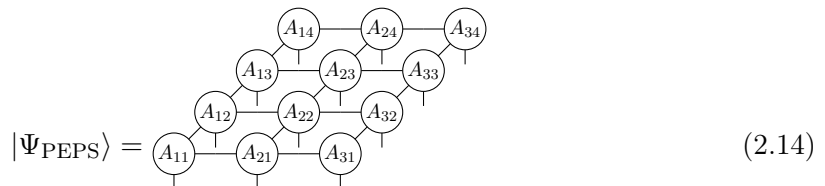
where the  $\rho^{(i)}$  and  $\rho^{(i-1)}$  are positive diagonal matrices and  $\text{Tr}[\rho^{(i)}] = \text{Tr}[\rho^{(i-1)}] = 1$ . An MPS with all left/right-normalized tensors can be brought into canonical form by the unitary transform determined by the condition in Eq. (2.13). It is often stated in the literature that the canonical form of MPSs is crucial to the success of the DMRG and TEBD algorithms. Here, we would like to point out that most of the advantages in 1D numerical algorithms come from merely the isometric condition. In terms of numerical algorithms, the linear geometry of MPSs itself ensures the  $\mathcal{O}(D^3)$  complexity for the exact shifting of the orthogonality center and the exact contraction between MPSs. Apart from that, the success of efficient 1D algorithms including DMRG and TEBD with MPSs are because of the two properties at the orthogonality center: (i) the identity norm tensor  $N$  and (ii) the optimal bond truncation. Firstly, because the norm matrix at the orthogonality center is the identity, the optimization problem in DMRG is a standard eigenvalue problem instead of a generalized eigenvalue problem. Secondly, at each step in the algorithm, the truncation based on the SVD over the two-site orthogonality center is a local update, which

is optimal for the global state since the basis is orthonormal. Both properties above come from the isometric condition of the normalized MPS tensors as discussed above. And it does not require the additional Schmidt-state gauge condition from the canonical form. To generalize the success of 1D algorithms to higher dimensions, one can focus on generalizing the isometric condition to TNSs in higher dimensions. We review the previous work in Sec. 2.3.

## 2.2. TNSs in 2D

In this section, we review the properties and algorithms of general TNSs in 2D, focusing on rectangle lattice geometry.

The projected entangled pair states (PEPSs) [60] are the generalization of MPSs to 2D and higher-dimensional systems. The concept has been introduced earlier [61–63] as TNSs in higher dimensions. In the following, we use the two terms “PEPS” and “TNS” interchangeably. An example of PEPS on a finite 2D lattice is given below.



$$|\Psi_{\text{PEPS}}\rangle = \text{Diagram of a 3x4 grid of tensors } A_{ij} \text{ with free legs} \quad (2.14)$$

There are some significant differences between 1D TNSs and 2D TNSs. First of all, not all area-law states can be expressed as constant bond dimension TNSs for  $D > 1$  [110]. Still, 2D TNSs are conjectured to be efficient representations for ground states of gapped local Hamiltonians [41] and numerically [66–68, 120] have shown great success in simulating ground states of strongly correlated models, even for gapped chiral topological ordered state [116].

Secondly, even if a finite bond dimension PEPS efficiently approximates the desired ground state in terms of the number of parameters, the computation can still be nontrivial. Precisely, the exact evaluation of the norm of the PEPS or any physical observables is complete for the complexity class  $\#\mathcal{P}$  [72, 121, 122]. Nevertheless, in practice, approximate contraction methods still work reasonably well for physical states with polynomial cost in bond dimensions. We briefly review the approximate contraction scheme and the numerical algorithms based on them in the following sections.

Lastly, we would like to point out that there is no rigorous definition of canonical form for TNSs in dimension  $D > 1$ <sup>2</sup>. Moreover, general TNSs cannot be brought into the isometric form directly by gauge freedom. This is, however, not surprising as the isometric form of TNSs ensures efficient computation of the norm of the TNSs. IsoTNSs form a subclass of general TNSs and we give a more detailed introduction in Sec. 2.3

### 2.2.1. Boundary contraction

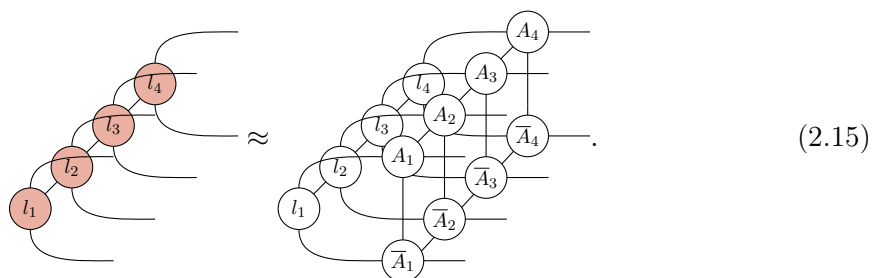
As mentioned above, the evaluation of tensor network contraction is average-case hard [122]. Approximate contraction methods [124] have been developed since the start of the field and can be roughly separated into two categories; namely, coarse-graining methods and boundary contraction methods. Coarse graining methods include the tensor renormalization group (TRG) [125] and its generalization [126–130]. These methods resemble the

<sup>2</sup>A recent work proposes the minimal canonical form which differs from the original definition of canonical form but can rigorously generalize to higher dimensions. See Ref [123].

real-space renormalization group scheme from Kadanoff [131] where the algorithm iteratively zoom out from, i.e., coarse grain, the underlying systems. Hence, these algorithms can contract exponentially large system sizes in terms of the number of iterations.

In the following and in this thesis, we mainly focus on the boundary contraction approaches, where the boundary is contracted iteratively and the number of degrees of freedom summed over grows linearly with the number of iterations. There are two main common boundary approaches to contract the double layer (bra-ket) structure of PEPS; namely the boundary matrix-product operator (bMPO) contraction and the corner transfer matrix renormalization group (CTMRG) approach.

The bMPO approach relies on compressing the boundary environment information with matrix-product operators (MPOs) iteratively. At each iteration, one contracts a new column of the PEPS and its conjugation with the bMPO and solve the approximation problem to find the new bMPO,



We approximate the right-hand side of the equation, including the previous bMPO and the column to contract, by a new bMPO with at most bond dimension  $D'$ . The approximation can be formalized as a minimization problem of the  $L^2$  difference<sup>3</sup> between the left and right-hand sides of the equation.

For finite systems without translational invariance, the problem of finding optimal bMPO resembles the structure of MPO-MPS compression. The direct SVD approach first merges the bra column, ket column, and bMPO tensor and then performs SVD truncation, which has more expensive scaling  $\mathcal{O}(D^8 D'^2 + D^6 D'^3)$ . Instead, the variational update from solving the alternating least square problem can utilize the sparse tensor contraction structure and has the scaling  $\mathcal{O}(D^6 D'^2 + D^4 D'^3)$ .

For infinite systems with translational invariance, the problem of finding the bMPOs is reduced to finding one fixed point bMPO and is first solved by infinite time evolving block decimation (iTEBD) algorithm [120] and recently by the variational uniform matrix-product state (VUMPS) method [133].

The other common boundary approach applied to infinite translationally invariant systems is the CTMRG method [73, 134], which combines the idea of corner transfer matrix from Baxter [135–137] and White’s DMRG algorithm [47, 138]. The CTMRG methods have successfully been applied to the contraction of 2D and three-dimensional (3D) classical partition functions and 2D quantum systems.

All the approaches above are algorithms designed to contract the tensor network on a regular lattice in some given orders based on prior knowledge about the physical systems. It is worth noting the recent development in automatic contraction schemes for both exact [139, 140] and approximate contraction [141]. For approximate contraction, the contraction order and compression strategy are optimized according to the score function, yielding results matching the state of the art of tensor network contraction algorithms<sup>4</sup>.

<sup>3</sup>It may be advantageous to consider other distance measures. One potential direction is to incorporate the recently proposed MPO canonical form [132].

<sup>4</sup>For exact contraction, the algorithm can find contraction paths yielding many orders of magnitude speed-up across various use cases. For approximation contraction, contractions of “non-hard” tensor net-



Another exciting direction is the connection between the tensor network contraction and the belief propagation [142], motivated by the duality of tensor networks and graphical models [143]. It is shown that the approximate contraction using a product state is equivalent to the loopy belief propagation [142]. The improved version using the block belief propagation shows similar results to the approximate contraction using bMPO [144]. It is an active field of study that potentially could lead to more powerful algorithms [145].

### 2.2.2. Algorithms for time evolution

Time evolution of 2D PEPS can be achieved using various algorithms. In this section, we will review different approaches for performing time evolution on PEPS.

#### Local update approaches

Local update approaches are methods that perform local tensor updates iteratively, for example, the TEBD algorithm with MPS. These algorithms typically involve applying a Trotterized time evolution operator locally and finding the update tensors that approximate the time-evolved state locally. However, there is a fundamental difference between the time evolution algorithm for PEPS and MPS. While general MPS can be brought into the canonical/isometric form, allowing the update tensors to be determined through a SVD, general PEPS does not have this isometric condition. As a result, updating the tensors in PEPS requires solving an alternative least squares (ALS) problem, which is defined by the double-layer contraction of the overlap and the norm matrix  $N$ . The contraction is usually computed with the previously introduced approximate contraction algorithms. Below, we provide a list of different methods categorized based on the accuracy of their environment approximations:

- **Simple update (SU):** This method involves approximating the environment by a product state [146]. The entanglement structure of the PEPS is neglected, resulting in a simplified representation of the environment. The complexity of the update is  $\mathcal{O}(D^5)$  when using reduced tensors.
- **Cluster update (CU) / Neighborhood tensor update (NTU):** In this approach, a cluster of tensors, i.e., the neighborhood, around the update tensor is taken into account as the environment [147]. The cluster is either contracted exactly to preserve hermiticity [148] or approximately using boundary contraction approaches [149]. The computational cost is reduced compared to a full update.
- **Full update (FU):** The FU method aims to approximate the full environment surrounding the update tensor by approximate contraction [64, 65]. This approach considers the entanglement structure explicitly and provides a more accurate representation of the environment, but it can be computationally demanding.

#### Global update approaches

Global update approaches are methods that update all the parameters of the PEPS simultaneously at each iteration. Two common methods are:

- **Gradient update:** This approach involves approximating a time-evolved state  $U(dt)|\psi\rangle$  with a variational state  $|\phi\rangle$  of smaller bond dimension, where  $U(dt)$  is typically an unitary acting globally. The variational state is determined by maximizing

---

works, e.g., the partition function of the Ising model, achieve similar or slightly better performance than standard boundary contraction methods on regular lattices.



the overlap between the variational state and the time-evolved state with gradient methods over all variational parameters. The gradient is computed by approximate boundary contraction methods.

- **Variational Monte Carlo (VMC):** VMC is a stochastic method to evaluate the expectation values from the given trial wave function with Monte Carlo sampling. Moreover, one can evaluate the energy gradient and the quantum Fisher matrix. This information, together with the log gradient, defines the time-dependent variational principle (TDVP) equation, also often known under the name of the stochastic re-configuration (SR) method in VMC context [150]. As a result, we can perform time evolution by solving the TDVP equation.

These algorithms for time evolution provide different trade-offs between accuracy and computational cost. Depending on the specific problem and available resources, different approaches may be preferred. In the section 2.3, we will see that imposing the isometric condition on PEPS leads to a simple local update method for time evolution, i.e. TEBD<sup>2</sup>, which is a 2D generalization of TEBD algorithm.

### 2.2.3. Algorithms for energy minimization

Algorithms for energy minimization are crucial for studying the ground states of quantum many-body systems. In this section, we discuss various algorithms that are commonly used.

- **Imaginary time evolution:** Imaginary time evolution is a widely used technique for finding the ground state of a quantum system. By applying the imaginary time evolution operator  $e^{-\tau H}$  to an initial state  $|\psi_0\rangle$  that has a finite overlap to the ground state and iteratively projecting onto the low-energy states, the system gradually evolves towards the ground state, i.e.,

$$|\text{ground state}\rangle = \lim_{\tau \rightarrow \infty} \frac{e^{-\tau H} |\psi_0\rangle}{|e^{-\tau H} |\psi_0\rangle|}. \quad (2.16)$$

This method is versatile as it can be implemented using one of the time evolution methods introduced above by changing the evolution time  $t$  to imaginary time  $\tau = -it$ .

- **DMRG<sup>2</sup>:** The DMRG algorithm is a powerful method originally developed for studying one-dimensional quantum systems using MPS. The algorithm iteratively updates the tensors locally to minimize energy by solving the standard eigenvalue problem. It can also be generalized to PEPS [60]. The algorithm iteratively updates the tensors locally to minimize energy by solving the *generalized* eigenvalue problem, due to the lack of isometric condition. We denote this 2D generalization of the algorithm as DMRG<sup>2</sup>.
- **Gradient descent:** Gradient descent algorithms can be broadly categorized into two types: stochastic gradient descent (SGD) algorithms and approximate gradient descent algorithms. The SGD algorithm is commonly implemented within the VMC framework. A simple example of a SGD algorithm is the plain VMC method. By incorporating techniques such as SR, the SGD algorithm can be interpreted as an imaginary time evolution or a second-order SGD algorithm. Approximate gradient descent algorithms attempt to approximate the exact gradient using approximate boundary contractions. They have been shown to yield state-of-the-art results for infinite projected entangled pair states (iPEPS). While these algorithms can also be applied to finite PEPS, special care must be taken to avoid instability.

These algorithms for energy minimization provide valuable tools for exploring the ground state properties of quantum many-body systems. In the following chapters and sections, we will further explore and discuss the applications of these algorithms in the context of isoTNS on finite 2D and infinite strip geometries.

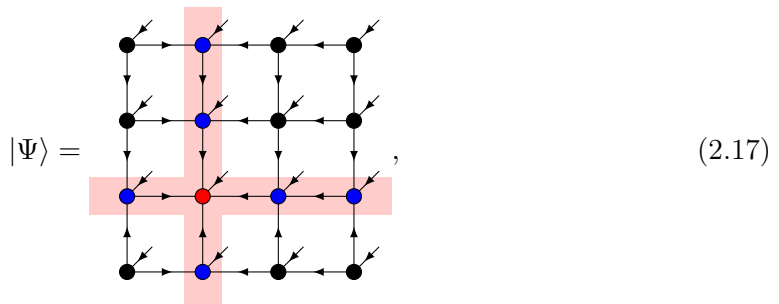
## 2.3. IsoTNSs in 2D

IsoTNSs, which are tensor networks constructed using only isometric tensors, have their origins in the study of 2D quantum cellular automata, as first described by Richter in 1995 [151]. However, it was not until recently, that isoTNSs were formulated within the modern tensor network language as variational ansatz by Zaletel and Pollmann [75]. This enables the application of isoTNSs to the study of 2D quantum many-body systems.

Following the work, it has been proven by Soejima et al. [152] that all the fixed-point ground state wavefunction of string-net liquid model [153, 154] admit exact isoTNS representation with finite bond dimension and that the orthogonality hypersurface may be placed anywhere. Moreover, states formed by finite depth local unitary circuits with the fixed point wavefunctions also admit exact isoTNS representations of finite bond dimension. This covers a wide range of 2D quantum states, including all bosonic abelian topological orders with gappable edges, indicating the ability of isoTNSs to encode long-range entanglement as in the topological ordered state.

### 2.3.1. Finite 2D isoTNSs

Using the introduced isometric tensor notation, an isoTNS on a 2D square lattice can be represented diagrammatically as



$$|\Psi\rangle = \text{Diagram}, \quad (2.17)$$

where each tensor has five indices, one physical index, and four virtual indices, with the local Hilbert space dimension denoted as  $d$  and the virtual bond dimension as  $D$ . The specific structural arrangement of the 2D isoTNS is motivated by the following considerations.

### Generalization of isometric structure to higher dimensions

To generalize the isometric structure of 1D isoTNS to higher dimensions, we consider a structural arrangement of isometries for the isoTNSs described as follows. For a  $k$ -dimensional isoTNS, we assume that we can find a  $(k - 1)$ -dimensional hyperplane in the isoTNS, that separates the  $k$ -dimensional space into two parts with the condition that the isometries from both parts are pointing towards the hyperplane. This means we can define a  $(k - 1)$ -dimensional state in terms of the boundary states of the two isometric maps representing the full  $k$ -dimensional state. We can then successive continue this reduction of the dimension until 0D. In other words, we represent the  $k$ -dimensional state as the  $(k - 1)$ -dimensional state in an orthonormal basis formed by the  $k$ -dimensional boundary map.

For clarity, let us first consider some examples: (i) A 1D MPS with an orthogonality center, as discussed in the previous section, has a 0D orthogonality center and 1D boundary maps as shown in Fig. 2.1. The truncation on the orthogonality center is optimal for the 1D quantum state as shown in Eq. (2.11). (ii) A 2D isoTNS on a rectangular lattice as shown in Fig. 2.2a. The column colored in red is an effective 1D wavefunction with 2D isometric boundary maps as shown in Fig. 2.2c. We can view the 1D wavefunction as a 1D MPS by grouping and reinterpreting the virtual indices.

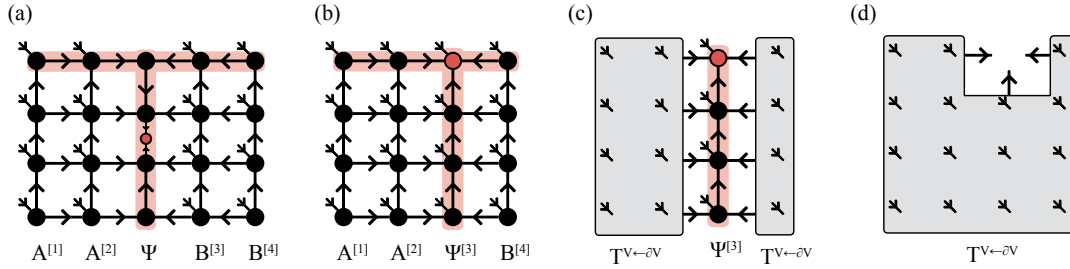


Figure 2.2.: Analogously, an isoTNS in 2D having orthogonality hypersurface colored in red and with a column (a) without physical indices  $\Psi$  and (b) with physical indices  $\Psi^{[l]}$ . (c) The decomposition of an isoTNS as the left and right isometric boundary maps with effective 1D wavefunction  $\Psi^{[3]}$ . (d) The boundary map  $T^{V\leftarrow\partial V}$  of the orthogonality center of the isoTNS (b).

### General properties of isoTNSs

With the structure of isoTNSs in mind, we now discuss their general properties and give concrete examples using 2D isoTNSs. In terms of the numerical algorithm, isoTNSs have two ideal properties: (i) *optimal bond truncation* on the orthogonality center and (ii) *identity norm tensor* on the orthogonality center. These two properties are the direct consequence of the isometric boundary map. The nested isometric boundary maps of dimension  $k, k-1, \dots, 1$  would form a single isometric boundary map to the orthogonality center. Therefore, any optimal truncation on the orthogonality center in the  $L^2$  norm, e.g. SVD, is an optimal truncation of the  $k$ -dimensional wavefunction because the isometric boundary map is norm-preserving. Similarly, because the boundary map of the tensor on the orthogonality center is isometric as shown in Fig. 2.2d, the norm tensor  $N_l = (T^{V\leftarrow\partial V})^\dagger T^{V\leftarrow\partial V} = \mathbb{1}_{\partial V}$  is an identity operator. Formally, the arrangement of isometries described above defines a causal structure of the tensor network flowing in the reverse direction of the arrows since the isometries would form Kraus operators. With our setup, the arrows of isometries don't form a loop. It means the isoTNSs considered are physical states that could be prepared by sequential quantum circuits [155, 156]. Here, we only consider the isoTNSs with a single orthogonality center <sup>5</sup>.

### Notation

In this work, we focus on 2D isoTNSs and use a notation inherited from MPSs. We can view the 2D isoTNS on the rectangular lattice as a generalization of the MPS where each isometry of the MPS is extended to a column of isometries. Therefore, we use  $A^{[l]}$  and  $B^{[l]}$  to denote columns of isometries that are left-normalized and right-normalized. We use  $\Psi$  and  $\Psi^{[l]}$  to denote the columns containing orthogonality center that are without and with physical indices as shown in Fig. 2.2a and Fig. 2.2b. We see in Fig. 2.2, that the direction

<sup>5</sup>It is possible to have more than one orthogonality center in the isoTNS. In that case, the reduced density matrices of the orthogonality centers will be separable.

of the isometries is chosen consistently pointing toward the orthogonality center (red circle) and each column can be contracted to an isometry, recovering the MPS structure.

Following the discussion above, we see the red-colored region in Fig. 2.2a and Fig. 2.2b define sub-regions of isoTNSs that have special properties. We call these effectively 1D regions the “orthogonality hypersurfaces” of the isoTNSs. Inside this region, we can move the orthogonality center from site to site exactly by orthogonal matrix decomposition. This 1D region has only incoming arrows and hence an isometric boundary map. Any variationally optimal algorithm inside the orthogonality hypersurface is variationally optimal for the global state. Utilizing these properties, we can run similar 1D algorithms on the orthogonality hypersurface.

### Evaluation expectation values

General 2D TNSs contractions have the complexity scaling exponentially in the system size and require approximate contractions [124]. For a 2D isoTNS, the expectation values for any operator acting only on column  $\Psi^{[l]}$  or within the orthogonality hypersurface permit an exact evaluation with polynomial scaling in the system size. The expectation values can be evaluated using standard MPS contraction on this column, i.e.,  $\langle \psi | \hat{O} | \psi \rangle = \langle \Psi^{[l]} | \hat{O} | \Psi^{[l]} \rangle$  – no contraction of 2D networks is involved.

For observables outside the column  $\Psi^{[l]}$ , we could move the orthogonality column or hypersurface over the sites by approximate methods introduced in Sec. 2.3.2 and evaluate the expectation as described above. Alternatively, one can perform the general 2D TNSs contraction. Note that both approaches are approximate algorithms.

### Shifting the orthogonality column

To implement practical algorithms using isoTNSs, we need a method to move the column  $\Psi^{[l]}$  around efficiently. For an MPS, we can exactly move the orthogonality center to any position by the QR algorithm, enabling efficient evaluation of local operators. However, it is not possible to change the isometric direction of the entire column for 2D isoTNS directly going from  $\Psi^{[l]}$  to  $A^{[l]}\Psi$ . The direct approach, which collapses the entire column  $\Psi^{[l]}$  into a single tensor and performs a QR decomposition, is doomed because of the exponential scaling. Therefore, new algorithms are required to shift the isometry direction of the whole column  $\Psi^{[l]}$  to  $A^{[l]}\Psi$ . The pioneering work [75] proposed two different algorithms to do this; namely, the variational Moses move (MM) and the sequential MM<sup>6</sup>. We give a brief review of the methods in Sec. 2.3.2 and Sec. 2.3.3 and extend the study of the algorithms in the next chapter.

#### 2.3.2. Variational Moses move

Recall in 1D the orthogonal matrix decompositions, e.g. QR decomposition, move the orthogonality center (thus shift the directions of the isometries) by  $\Psi^{\sigma_{l-1}} B^{\sigma_l} = A^{\sigma_{l-1}} \Psi B^{\sigma_l} = A^{\sigma_{l-1}} \Psi^{\sigma_l}$ , where each of the  $\Psi, A, B$  is a tensor. In 2D,  $\Psi, A, B$  each refers to a column of tensors as in Fig. 2.2. There is no orthogonal decomposition of the whole column of tensors while keeping the locality (tensor decomposition) structure. QR decomposition or SVD on each individual tensor would destroy the matching of isometry directions and thus destroy the identity boundary map of isoTNS.

While an exact solution does not exist, we consider the following variational problem: Given the column  $\Psi^{[l]}$ , we find columns  $A^{[l]}, \Psi$  such that the distance between two repre-

<sup>6</sup>The name comes from the biblical story of Moses splitting the Red Sea. The name suggested by Roger Mong is given for that the algorithms approximately split the sea of entanglement.

sented states is minimized, i.e.,

$$\arg \min_{A^{[l], \Psi} \left\| \Psi^{[l]} - A^{[l]} \Psi \right\|^2 \quad (2.18)$$

with the constraint that  $A^{[l]}$  is a column of isometries pointing toward  $\Psi$  column (see Fig. 2.3). Notice that the  $\Psi$  column does not have physical indices. One can think of this variational problem as the analogy of the QR decomposition for a column of tensors except that  $\Psi$  is not restricted to be upper triangular. While the QR decomposition is a deterministic algorithm providing a numerical exact decomposition, the variational problem here generically only provides an approximate decomposition. If the problem could be solved, the next step is to contract the “zero-site” column  $\Psi$  to the next right-normalized column  $B^{[l+1]}$  and form the  $\Psi^{[l+1]}$  column as in Fig. 2.3b and Fig. 2.3c. And one can continue forward with the move in analogy to the 1D case. We call this procedure of solving Eq. (2.18) iteratively and obtaining  $A^{[l], \Psi}$  columns the *variational Moses move*.

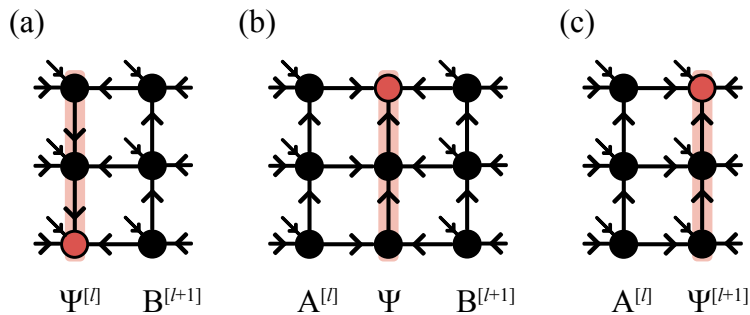


Figure 2.3.: The column  $\Psi^{[l]}$  is shifted by starting with a Moses move from (a)  $\Psi^{[l]}B^{[l+1]}$  to (b)  $A^{[l]}\Psi B^{[l+1]}$ . The combination of  $\Psi B^{[l+1]} = \Psi^{[l+1]}$  would lead to higher bond dimension which requires further truncation. We implement this with the same idea as the standard MPO-MPS compression method. This gives the two columns  $A^{[l]}\Psi^{[l+1]}$  in (c). An optional but more expensive step includes the maximizing of the two columns overlap  $\langle \psi | \psi' \rangle$  of the wavefunction before in (a) and after in (c). We describe the detail of the steps above in Appendix 3.1.3.

We separate the variational MM into two parts which correspond to two common types of variational problems for tensor networks. The first part is the variational optimization over tensors in column  $\Psi$ . This is an unconstrained optimization problem with general tensors. The second part is the variational optimization over the isometries in column  $A^{[l]}$ . This is a constrained optimization problem with isometry tensors. Both types of problems could be solved in an ALS fashion as described below.

### Unconstrained optimization for tensors in column $\Psi$

The unconstrained optimization problem over  $\Psi$  is similar to the problem of variational approximation that occurs in the iterative compression of an MPS [31, 64]. The general problem has the following setup: Given a target state  $|\psi\rangle$ , we want to find the optimal isoTNS  $|\phi\rangle$  representation in  $L^2$  norm by varying a single tensor  $x$  at site- $l$  at a time <sup>7</sup>.

The solution of the minimization problem

$$\arg \min_x \left\| |\psi\rangle - |\phi(x)\rangle \right\|^2 \quad (2.19)$$

<sup>7</sup>Notice that when the problem is imposed on normalized tensor networks, minimizing the difference in  $L^2$  norm is equivalent to maximizing the fidelity  $\mathcal{F}(\phi, \psi) = |\langle \phi | \psi \rangle|^2$ . This is because of the global phase is not fixed and one can always set it such that  $|\langle \phi | \psi \rangle| = \text{Re}[\langle \phi | \psi \rangle]$

must satisfy the extremum condition  $\partial_{x^*} \langle \phi | \phi \rangle - \partial_{x^*} \langle \phi | \psi \rangle = 0$ . Therefore, the optimal tensor  $x$  is found by solving the system of linear equations,

$$N_l x = b, \quad \text{i.e. } x = N_l^{-1} b \quad (2.20)$$

where  $N_l$  is the norm matrix for site- $l$  obtained from contraction  $\langle \phi | \phi \rangle$  and leaving out tensor  $x$  in ket and tensor  $x^*$  in bra, and  $b$  results from leaving out tensor  $x^*$  from  $\langle \phi | \psi \rangle$ .

Similar to the variational compression of MPS, at each update, we keep the update tensor  $x$  at the orthogonality center such that the norm matrix for site- $l$  is the identity operator,  $N_l = \mathbb{1}$ . As a result, the optimal update is given by the contraction  $b$  without solving a system of linear equations. In variational MM, each local update  $b$  is formed by the contraction of three column  $\Psi^{[l]}, A^{[l]}, \Psi$ . After each update, we move the orthogonality center to the next site in the column by orthogonal matrix decomposition.

### Constrained optimization for isometries in column $A^{[l]}$

The constrained optimization over tensors  $x$  within column  $A^{[l]}$  requires  $x$  to be an isometry. The constrained optimization could not be solved by the previous approach as in Eq. (2.20), since the previous solution is a general tensor. The isometric constraint cannot be consistently restored by orthogonal matrix decomposition. First, we rewrite the problem from minimizing  $L^2$  distance between states to maximizing the real part of the overlap,

$$\begin{aligned} & \arg \min_{x \in \text{isometry}} 2 - 2\text{Re} [\langle \psi | \phi \rangle] \\ &= \arg \max_{x \in \text{isometry}} \text{Re} [\langle \psi | \phi \rangle] \\ &= \arg \max_{x \in \text{isometry}} \text{Re} [Tr[b^\dagger x]]. \end{aligned}$$

In the first line, we use the condition that the isoTNSs have an identity norm. In the last line, we reshape the isometric tensor  $x$  and the tensor  $b$  to matrices. The resulting constrained optimization problem is known as the orthogonal Procrustes [157]<sup>8</sup> problem and admits closed-form solutions. The optimal update for  $x$  is given by  $VU^\dagger$  from the SVD of the  $b^\dagger = U\Sigma V^\dagger$  matrix. A similar problem also appears in optimizing multi-scale entanglement renormalization ansatz (MERA) with linearization [158, 159]. The derivation and detailed discussion are given in Appendix A.1.

One could start with randomly initialized  $A^{[l]}, \Psi$  and iteratively sweep through and update all the tensors in both columns. The algorithm stops when desired accuracy or convergence criterion is reached. In practice, we observe that random initialization with local updates may lead to slow convergence toward a sub-optimal minimum. Therefore, we introduce in the next section a complementary approach for shifting the column, which could serve as a good initialization for the variational MM.

After each variational MM, one contracts the  $\Psi$  column to the next column. The two columns contraction is similar to the application of an MPO to an MPS. Therefore, the most efficient way is to consider the MPO-MPS contraction variationally [54, 102]. Note that there are a few subtleties: The variational MM  $\Psi^{[l-1]} B^{[l]} = A^{[l]} \Psi^{[l]}$  is generically not exact and inherits errors. Furthermore, after moving and tagging the column, the bond dimension on column  $l$  grows. To keep the bond dimension fixed, a truncation occurs.

<sup>8</sup>The problem is named after Procrustes, a bandit from the Greek mythology, who forced passersby to fit his bed by stretching or cutting off their bodies.



### 2.3.3. Sequential Moses move

In this section, we review a sequential solution for moving the orthogonality hypersurface [75]. The sequential MM is a greedy algorithm that sequentially splits one column into two satisfying the isometric constraints by a single unzipping sweep. We observe in practice that the sequential MM has an error very close to the optimal variational result while being much faster. In addition, this approximate solution can serve as a good initialization for variational MM.

The idea of the sequential MM is to perform a sequence of tripartite decompositions at the orthogonality center. We illustrate the idea in Fig. 2.4. At each step, we split the single tensor into three tensors as shown in Fig. 2.4a to Fig. 2.4d. Iteratively, we split the full column into two by repeating such decompositions as illustrated in Fig. 2.4e. To simplify the notation, we always merge the indices of the tensor at the orthogonality center as an order-3 tensor and denote it as  $\Psi_{a,b,c}$ .

The tripartite decomposition is composed of two consecutive SVDs and a gauge fixing procedure. We use Einstein summation convention and describe the decomposition step by step as follows:

- (i) Perform an SVD on  $\Psi_{(a),(b,c)}$

$$\Psi_{a,(b,c)} = A_{a,s} \Lambda_{s,s} V_{s,(b,c)} = A_{a,s} \Theta_{s,b,c}.$$

The  $A$  is an isometry. In principle, we could also directly arrive at the final form by reduced QR decomposition  $\Psi_{(a),(b,c)} = A_{a,s} \Theta_{s,(b,c)}$ . However, in certain cases, we would truncate the bond dimension which requires the SVD instead.

- (ii) Split the index  $s$ . To get to Fig. 2.4b, we split the index  $s$  into  $s_l, s_r$ . In practice, we choose their dimensions  $|s_l| \sim |s_r| \sim \lfloor \sqrt{|s|} \rfloor$  in order to distribute the bonds evenly (though in anisotropic models, other prescriptions may be appropriate). Also the bond dimension  $|s_l|$  and  $|s_r|$  should be smaller than the maximum vertical bond dimension  $D_V$  and horizontal bond dimension  $D_H$  set for the simulation. Notice that if  $|s| > |s_l||s_r|$ , then the SVD in step (i) is a truncated SVD, i.e.,

$$\Psi_{a,b,c} \approx A_{a,(s_l,s_r)} \Theta_{(s_l,s_r),b,c}. \quad (2.21)$$

For a given pairs of  $(s_l, s_r)$ , we first truncate and keep only the leading  $|s_l||s_r|$  singular values in  $\Lambda_{s,s}$ . We then form the  $\Theta_{s,b,c}$  and reshape it into  $\Theta_{s_l,s_r,b,c}$ . Note the ordering of the reshaping, column-major or row-major, has only a minor effect for the following reason. The decomposition  $\Psi = A\Theta$  has a gauge degree of freedom  $A\Theta = (AU^\dagger)(U\Theta) = A'\Theta'$ , where  $U$  is an arbitrary unitary matrix. Absorbing  $U^\dagger$  into orthonormal basis  $A$  results in a new orthonormal basis  $A'$ . We defer the discussion of finding the unitary  $U$ , i.e., fixing the gauge, for the decomposition until after we describe the full picture of the sequential MM. At this point, we assume we fix the gauge by the unitary determined by some procedures.

- (iii) Perform an SVD on  $\Theta_{(s_l,c),(s_r,b)}$ . Given  $\Theta_{s_l,s_r,b,c}$ , we rearrange and group the indices  $(s_l, c)$  and  $(s_r, b)$  and perform the (truncated) SVD as shown in Fig. 2.4b and Fig. 2.4d. That is

$$\Theta_{(s_l,c),(s_r,b)} \simeq V_{(s_l,c),t} \Lambda_{t,t} Q_{t,(s_r,b)} = \Psi_{s_l,c,t} Q_{t,(s_r,b)}. \quad (2.22)$$

Finally, combining Eq. (2.21) and Eq. (2.22), we have the decomposition

$$\Psi_{a,b,c} \approx A_{a,s_l,s_r} \Psi_{s_l,c,t} Q_{t,s_r,b} \quad (2.23)$$

as in Fig. 2.4d.

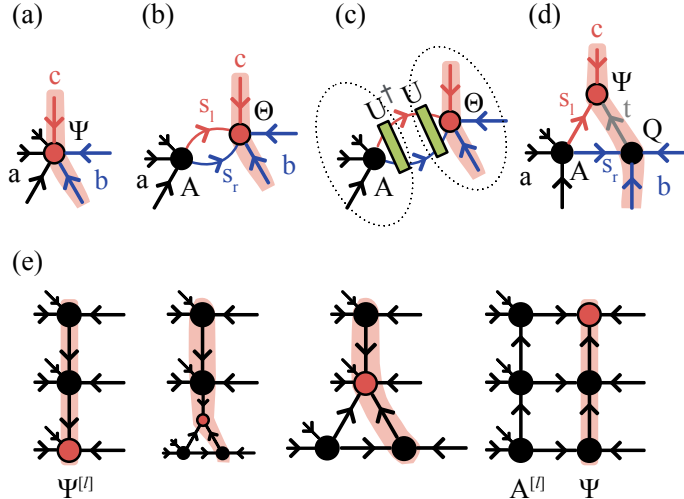


Figure 2.4.: (a-d) The tripartite decomposition (e) Moses move  $\Psi^{[l]} = A^{[l]}\Psi$  by iterative tripartite decomposition. (a) We group the indices of color black, blue, and red together respectively as  $a, b, c$  and denote the orthogonality center as  $\Psi_{a,b,c}$ . (b) After the SVD and merging the singular values to the orthogonality center, we again group the indices of the same color together and denote it as  $\Theta_{(s_l,c),(s_r,b)}$ . The index  $s_r$  is colored in blue and  $s_l$  in red. (c) The insertion of identity operator  $I = U^\dagger U$ , where  $U$  is chosen to minimize the entanglement in  $\Theta$ . We absorb the  $U^\dagger$  into isometry  $A$  and  $U$  into  $\Theta$ . (d) SVD on the final  $\Theta$  complete the tripartite decomposition. (e) Utilize the decomposition, we unzip the columns with the orthogonality center moving to the top.

The entire sequential MM is shown in Fig. 2.4e, where we start from the orthogonality center at the bottom and move to the top, unzipping the entire  $\Psi^{[l]}$  column by the tripartite decomposition. Before the MM, the  $\Psi^{[l]}$  column is composed of tensors  $[\Psi^{\sigma_1}, P^{\sigma_2}, \dots, P^{\sigma_{L_y-1}}, P^{\sigma_{L_y}}]$ . At each step, we

- (i) Reshape the orthogonality center  $\Psi^{\sigma_l}$  to an order-3 tensor  $\Psi_{a,b,c}$
- (ii) Perform a tripartite decomposition on  $\Psi_{a,b,c}$  as in Fig. 2.4a-d
- (iii) Merge the orthogonality center after the decomposition with the tensor above to form the new center, i.e.,  $\Psi P^{\sigma_{l+1}} = \Psi^{\sigma_{l+1}}$ .

After step (iii) (merging the tensors), we recover the form as in Fig. 2.4a. Therefore, we can repeat and continue from step (i) until we reach the top of the column. We end the splitting at the top of the column by a single SVD. Collecting the  $A$  tensors and the  $Q$  tensors along the way, we obtain the  $A^{[l]}$  column consisting of tensors  $[A^{\sigma_1}, A^{\sigma_2}, \dots, A^{\sigma_{L_y-1}}, A^{\sigma_{L_y}}]$  and the  $\Psi$  column consisting of tensors  $[Q, Q, \dots, Q, \Psi]$ . The full procedure is illustrated in Fig. 2.4e. To continue, we see we could combine the  $\Psi$  column easily with the  $B^{[l+1]}$  column as in Fig. 2.3b and result in the  $\Psi^{[l+1]}$  column consisting of tensors  $[Q^{\sigma_1}, Q^{\sigma_2}, \dots, Q^{\sigma_{L_y-1}}, \Psi^{\sigma_{L_y}}]$  as in Fig. 2.3c.

In practice, we would impose a maximum bond dimension for column  $\Psi$  and also for column  $\Psi^{[l+1]} = \Psi B^{[l]}$  as in Fig. 2.3c. Truncations may take place on column  $\Psi$  during the second SVD and on column  $\Psi^{[l+1]}$  after the combination of columns.

Now we discuss the criterion and how to find the optimal  $U$  in the tripartite decomposition. The crucial insight from [75] is that the truncation error occurs in MM can be made smaller by utilizing the gauge degree of freedom between tensors (see Fig. 2.4c). An insertion of a pair of unitary and its conjugate, i.e., identity operator, before the second



SVD leave the overall tensors invariant. However, this changes the distribution of the singular values of the tensor  $\Theta$  when we absorb the unitary  $U$  into it and hence changes the truncation error. Therefore, we include the insertion of the pair of unitary and its conjugate as in Fig. 2.4c between the first and second SVD.

---

**Algorithm 1:** The sequential Moses move algorithm

---

**Input** : central column  $\Psi^{[l]}$ , cost function  $\mathcal{L}$ , bond dimension  $D$ , central bond dimension  $\eta$

**Output:** left-normalized column  $A^{[l]}$  and central column  $\Psi$  minimizing  $\|\Psi^{[l]} - A^{[l]}\Psi\|^2$  with error  $\varepsilon$

idx = 0,  $\varepsilon_0 = \text{inf}$ ;

**for** ( idx <  $N_{\text{row}} - 1$  ) {

    idx = idx + 1 ;

    Group the indices of  $\Psi^{[l]}[\text{idx}]$  to form  $\Psi_{a,b,c}$ ;

**Tripartite Split** (  $\Psi_{a,b,c}$  ) {

        (i) SVD on  $\Psi_{(a),(b,c)}$

$$C_{a,b,c} \simeq A_{a,s} \Lambda_{s,s} V_{s,b,c} = A_{a,s} \Theta_{s,b,c}$$

        (ii) Split the index  $s$  to  $s_l, s_r$ , s.t.

$$|s_l| < D, |s_r| < D$$

$$C_{a,b,c} \simeq A_{a,(s_l,s_r)} \Theta_{(s_l,s_r),b,c}$$

        (iii) Find unitary  $U$  from

$$\arg \min_{U \in \text{unitary}} \mathcal{L}(U\Theta)$$

        (iv) Insert the identity  $I = U^\dagger U$

$$A \leftarrow AU^\dagger \text{ and } \Theta \leftarrow U\Theta$$

        (v) SVD on  $\Theta_{(s_l,c),(s_r,b)}$ , s.t.

$$\Theta_{s_l,s_r,b,c} \simeq V_{s_l,c,t} \Lambda_{t,t} Q_{t,s_r,b} = \Psi_{s_l,c,t} Q_{t,s_r,b}$$

$$|t| < \eta$$

        (vi) Collect the tensors and error

$$\Psi_{a,b,c} \approx A_{a,s_l,s_r} \Psi_{s_l,c,t} Q_{t,s_r,b}$$

$$A^{[l]}[\text{idx}] \leftarrow A, \Psi[\text{idx}] \leftarrow \Psi$$

$$\varepsilon_{\text{split}} = \varepsilon_{\text{SVD-1}} + \varepsilon_{\text{SVD-2}}$$

    }

    Absorb  $\Psi$  to  $\Psi^{[l]}[\text{idx} + 1]$  ;

$\varepsilon_+ = \varepsilon_{\text{split}}$

}

SVD on  $\Psi^{[l]}[N] \simeq A^{[l]}[N]\Psi[N]$

$\varepsilon_+ = \varepsilon_{\text{SVD}}$

---

The optimal  $U$  is the  $U$  that leads to the smallest truncation error. Therefore, we solve the variational problem

$$\arg \min_{U \in \text{unitary}} \mathcal{L}(U\Theta) \quad (2.24)$$

where the cost function  $\mathcal{L}(U\Theta)$  is chosen such that minimizing it reduces truncation error. Originally, Rényi- $\alpha$  entanglement entropies, which act as a proxy to the truncation error, is considered as the cost function [75]. Once the cost function is given, the variational problem is solved by gradient descent methods. We give an overview of these procedures in Appendix A.1.3. The overall MM algorithm has a complexity of  $\mathcal{O}(D^7)$ , when all virtual bond dimensions in the isoTNS are  $D$  and physical dimension  $d \ll D$  is ignored.

In Chapter 3 Sec. 3.1, we expand our study by considering truncation error as the cost function. We compare the results obtained by using the Rényi- $\alpha$  entanglement entropies

and truncation errors as cost functions and provide insights on the optimization problem. Furthermore, we thoroughly examine the complexity of the algorithms, particularly when the bond dimension on the orthogonality column is increased to  $\eta$ .

To conclude, we show the overall sequential MM algorithm as in Alg. 1. We choose the convention that the algorithm takes central column  $\Psi^{[l]}$  with isometries pointing downward as input and returns left-normalized column  $A^{[l]}$  and central column  $\Psi$  with isometries pointing upward.

### 2.3.4. TEBD<sup>2</sup> algorithm

Given the ability to move the orthogonality center and column around the isoTNSs, standard tensor network algorithms can naturally be formulated. In the original work [75], the imaginary time evolution algorithm TEBD<sup>2</sup> is proposed and used to study the ground state problem. Here we give a brief review of the algorithm. In Chapter 3, we use the algorithm to perform real-time evolution.

The TEBD algorithm with MPSs is an algorithm utilizing local updates to perform time evolution. The TEBD-like algorithms for time evolution consist of three parts: (i) Suzuki-Trotter decomposition of the time evolution operator  $\hat{U}(dt) = \prod_i e^{-idtH_i}$  of local Hamiltonian  $H = \sum_i H_i$  into a set of two-site local operators (ii) local updates by applying the time evolution operator following optimal bond truncation at the orthogonality center (iii) shifting of orthogonality center. Combining these with 1D isoTNSs, i.e., MPSs, the resulting TEBD-like algorithms are similar algorithms slightly varying in the implementation details [48–51].

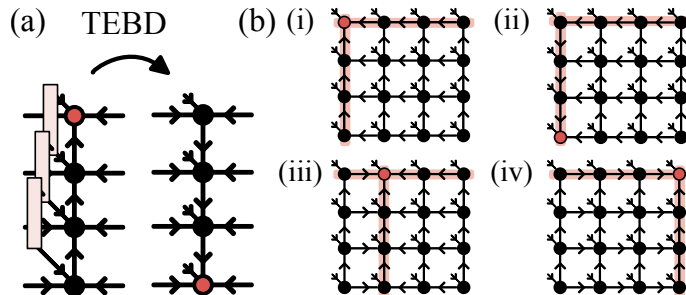


Figure 2.5.: The TEBD<sup>2</sup> algorithm: (a) The 1D TEBD (b)(i) Beginning with isometries pointing toward the upper left, one TEBD step as in (a) on the column would bring the orthogonality center down as in (ii). The MM, as in Fig. 2.3, brings the arrow back up and shifts the orthogonality hypersurface as in (iii). Repeating this through the system, we arrive at the form in (iv) in the end. With an anti-clockwise 90° rotation of (iv) we are back to (i) with column and row reverse. We then repeat the same steps and evolve the rows.

The TEBD<sup>2</sup> algorithm is the 2D generalization of the TEBD algorithm with 2D isoTNSs. The three building blocks work similarly in two dimension. For part (i), we consider the Suzuki-Trotter decomposition of the time evolution operator  $\hat{U}(dt) = \prod_{r,i} e^{-idtH_i^r} \prod_{c,j} e^{-idtH_j^c}$ , where  $H_i^r$  and  $H_j^c$  are the terms of the local Hamiltonian acting on columns and rows. For part (ii), we perform similar local updates at the orthogonality center. Since optimal bond truncation is guaranteed by the isometric form, there is no difference to the 1D algorithm except for the additional indices. However, the computational cost can be drastically reduce from  $\mathcal{O}(D^9)$  to  $\mathcal{O}(D^5)$  by applying the two-site gate update on the reduced tensors [64]. Moreover, the reduced tensor update is optimal for isoTNSs, which is different from the general TNSs [64, 160]. For part (iii), we utilize the SVD and MM to move around the orthogonality center and orthogonality hypersurface.

We sketch the outline of TEBD<sup>2</sup> algorithm here.

- (i) Start with an isoTNS with all isometries pointing toward the top left as in Fig. 2.5b(i).
- (ii) Perform 1D TEBD with a *reduced tensor* update on the column of the orthogonality center. After the sweep, the isometries all point down as in Fig. 2.5b(ii).
- (iii) Perform MM to bring the orthogonality center forward to the next column Fig. 2.5b(iii), then repeat the 1D TEBD as in step (ii). Continue and repeat these steps over all columns.
- (iv) The isometries now point toward the top right as in Fig. 2.5b(iv). The orientation of the isometries has effectively been rotated by 90° counterclockwise from the starting point, Fig. 2.5b(i). We may thus go back to step (i) by rotating the network by 90°, exchanging the role of rows and columns, and repeat.

Each TEBD<sup>2</sup> step evolves the system  $dt$  with two cycles of operation from step (i) to (iv). This is because on the first round, we finish the application of all the terms on columns  $\hat{U}^{\text{col}}(dt) = \prod_{c,j} e^{-idtH_j^c}$  to the state. With the rotation at step (iv), we interchange the columns and rows. On the second round, we thus evolve the “rows” of the original lattice. After two rounds, we arrive at a 180° rotated lattice with the system evolved by  $\hat{U}(dt)$ . By repeating this, one can perform real- or imaginary-time evolution with isoTNSs. We note that while each 1D TEBD step is individually a 1st-order Trotterization, after four rounds we obtain a 2nd-order Trotterization within columns and rows, as the effective reversal by 180° during rounds 3-4 cancels out errors via symmetrization. By using half time steps in the first and last time steps of the column updates, we could make the overall algorithm 2nd-order [75]. The method is termed TEBD<sup>2</sup> since it is a nested loop of the 1D TEBD algorithm [75].

The TEBD<sup>2</sup> algorithm differs from the time evolution algorithm of general 2D TNSs [64, 161] in the step of tensor updates. In general, applying the time evolution operator increases the bond dimensions of the TNSs. For isoTNSs, we can simply truncate the bond dimension by a local SVD, which is the *globally* optimal truncation because of the isometry conditions. For generic TNSs, local truncation is not optimal because it does not take into account the information of the rest of the tensor network. Instead, one has to solve the minimization problem approximating the time-evolved state  $|\psi(t + dt)\rangle$ ,

$$\arg \min_x \left\| |\phi\rangle - \hat{U}(dt)|\psi(t)\rangle \right\|^2. \quad (2.25)$$

The  $|\psi(t)\rangle$  denotes the original state at time  $t$ . One updates a single local tensor  $x$  in  $|\phi\rangle$ , where  $|\phi\rangle$  is a 2D TNS of the same fixed bond dimensions as the original TNS<sup>9</sup>. The optimal update is given by solving the systems of linear equations with the norm matrix  $N$  and vector  $b$ , similar to the problem described in variational MM. For TNSs, the evaluation of the norm matrix  $N$  and vector  $b$  involves the approximate TNSs contraction for the environment, which gives rise to the difference in SU and FU scheme [64]<sup>10</sup>. The isometric condition provides the optimal truncation and avoids the need of solving the systems of equation and environment approximation. This advantage comes with the cost of the truncation error in MM. Nevertheless, the computational complexity for the time evolution algorithm decreases from  $D^{10}$  for general TNSs to  $D^7$  for isoTNSs, where the  $D^7$  complexity comes from the MM.

<sup>9</sup>In practice, one updates two tensors at the a time using the reduced tensors update to lower the computational cost [64].

<sup>10</sup>See discussion in Sec. 2.2.2.

The 1D TEBD algorithm with MPSs has two sources of error: the Trotterization error  $\varepsilon_{\text{Trotter}}$  and truncation error  $\varepsilon_{\text{trunc}}$  due to the restricted bond dimension. Suppose we want to evolve the system to time  $T$  with a controllable targeted error of order  $\varepsilon_T$ , we could keep the error  $\varepsilon_{\text{Trotter}}$  and  $\varepsilon_{\text{trunc}}$  both around the order  $\sim \varepsilon_T$ . The truncation error  $\varepsilon_{\text{trunc}}$  could be made smaller than  $\varepsilon_T$  by increasing the bond dimension. The Trotterization error can be made as small as one wishes by decreasing step size  $\delta t$  or increasing the order  $p$  of Trotterization [54]. More precisely, the Trotterization error of each time step  $\delta t$  is  $\sim \delta t^{(p+1)}$  in  $L^2$  norm, and the accumulated Trotterization error is estimated by the sum of error at each time step and is about  $\sim T\delta t^p$ <sup>11</sup>.

The TEBD<sup>2</sup> algorithm has one additional source of error: the error in the MM  $\varepsilon_{\text{MM}}$ . This additional error affects the optimal choice of step size and order of Trotterization, depending on the use case. Crucially, unlike in one dimension, where the truncation per step  $\varepsilon_{\text{trunc}} \rightarrow 0$  as  $\delta t \rightarrow 0$  (because the state does not change), in two dimension  $\varepsilon_{\text{MM}}$  remains finite even as  $\delta t \rightarrow 0$ . This is because the MM is generally not exact (though  $\varepsilon_{\text{MM}} \rightarrow 0$  as  $D \rightarrow \infty$ ). To assess the consequence for real-time evolution, suppose we are interested in evolving a system to a specific total time  $T$ , and assume the MM error  $\varepsilon_{\text{MM}}$  is fixed, there exists an optimal step size  $\delta t$  and order  $p$ , which minimizes the sum of Trotterization error  $\varepsilon_{\text{Trotter}}$  and MM error  $\varepsilon_{\text{MM}}$ , i.e.  $\varepsilon_{\text{total}} = (T\delta t^p)^2 + \frac{T}{\delta t}\varepsilon_{\text{MM}}$ . By iterating through  $p = 1, 2, \dots$ , we can solve the minimization problem given the values of  $T$  and  $\varepsilon_{\text{MM}}$  and obtain the error  $\varepsilon_{\text{total}}^*$  and step size  $\delta t^*$  for the corresponding  $p$ . Comparing the total error obtained for different  $p$ , we can pick the optimal  $p^*$  and the corresponding optimal step size  $\delta t^*$ . When considering imaginary-time evolution, this error  $\varepsilon_{\text{total}}$  acts against the deceasing of energy  $\sim \varepsilon_E e^{-\Delta E_{\text{gap}} d\tau}$  per time step  $d\tau$ . The converging results thus have an error in energy  $\varepsilon_E \sim a\delta\tau^{2p} + b\frac{\varepsilon_{\text{MM}}}{\delta\tau}$  [75].

### 2.3.5. Discussion

To conclude the section on isoTNSs, we discuss and give some general remarks on the properties of isoTNSs and the related work.

Similar 2D isoTNS ansätze have been proposed recently with different approaches for shifting the orthogonality hypersurface [162, 163]. The isoTNSs have been generalized to represent 2D thermal states [101] and can be generalized to higher dimensions or different lattice geometries, for example, the recent work on 3D cubic lattice [164].

We note that it is possible for 2D isoTNSs to have a different arrangement in the direction of isometries that does not look like Fig. 2.2a. For example, we can have columns  $A$  and  $B$  not contracted to isometries pointing only toward the orthogonality center. As also pointed out in [162], the identity maps formed by the left and the right boundary are a sufficient but not necessary condition for an isoTNS to have an orthogonality center. For our work, we choose the “natural” arrangement in the sense that it is the direct generalization of MPS tensors to columns. In [162], columns of TNSs are turned into layers of unitaries as in quantum circuits, which corresponds roughly to the pattern as in Fig. 2.2a. All the work so far [75, 162, 163, 165] considers a similar arrangement of isometries for numerical convenience.

Aside from the previous attempts to generalize the canonical form of MPSs to PEPSs [166], recently a different generalization of canonical form for cyclic tensor networks by gauge fixing is proposed by Evenbly [167]. The proposed weighted trace gauge (WTG) condition requires the left and right boundary matrices to be *proportional* to identity  $\mathbb{1}$  for all the

<sup>11</sup>For an error per unit time  $\varepsilon = \varepsilon_T/T$  to be satisfied, first order method needs  $\frac{T}{\delta t} = \frac{T}{\varepsilon}$  time steps, while second order method  $\frac{T}{\delta t} = \frac{T}{\varepsilon^{1/2}}$  and so and so on. The total evaluation for first order TEBD is then  $\frac{T}{\varepsilon} \times 2$ , and for second order  $\frac{T}{\varepsilon^{1/2}} \times 5$ . Higher order is computationally preferred if one aims at higher accuracy  $\varepsilon \ll 1$ .

virtual bonds. We briefly review the definition of WTG in Appendix A.2. For acyclic networks, the WTG condition is equivalent to the canonical form condition. For general cyclic networks, the WTG condition does not restrict the state compared to the isometric condition since the WTG condition is only a gauge fixing, which does not change the overall states. TNSs in the canonical form defined with WTG are the same manifold as general TNSs. In contrast, isoTNSs restricted the states and are sub-manifolds of the general TNSs. Applying WTG to isoTNSs will lead to the  $\Gamma - \Lambda$  form [48, 49]. The isometric condition is still satisfied by combining the bond and site tensors.

One distinguishing property of isoTNSs from TNSs is the absence of internal correlation. The direct renormalization and truncation for general cyclic TNSs are not optimal due to internal correlations. One class of prominent examples are cyclic tensor networks with corner double line tensors [128, 168]. The cycle entropy  $S_{\text{cycle}}$  on a bond is a measure defined in [167] to quantify this physically redundant information contained in cyclic networks, i.e., internal correlations. We give a brief review of the definition of the cycle entropy  $S_{\text{cycle}}$  in Appendix A.2. When  $S_{\text{cycle}} = 0$ , the bond does not carry physically redundant information. However, TNSs usually have a non-zero internal correlation, i.e.,  $S_{\text{cycle}} \neq 0$ . One advantage of isoTNSs representation is that the cycle entropy  $S_{\text{cycle}}$  is always zero for all the bonds by construction. This is a property following the definition of the cycle entropy  $S_{\text{cycle}}$ . While we discuss this in more detail in Appendix A.2, we provide an intuitive argument here. IsoTNSs are states generated under sequential unitaries from the product states [169, 170]. The bonds in isoTNSs correspond to the actual physical degree of freedom on which the unitaries act, and hence,  $S_{\text{cycle}} = 0$ .



# 3. Efficient simulation of dynamics with two-dimensional isometric tensor networks

This chapter is largely based on the work published in Ref. [1], which builds on the previous work on isometric tensor network states (isoTNSs) reviewed in Chapter 2. The chapter is organized as follows. In Sec. 3.1, we revisit and study the effect of different cost functions for tripartite decomposition in the sequential Moses move (MM). We discuss the implementation detail and extend the variational MM to a two-column variational MM. In Sec. 3.2, we introduce a generalized density matrix renormalization group (DMRG) algorithm with two-dimensional (2D) isoTNSs (DMRG<sup>2</sup>) for variational energy minimization and implement the time-evolving block decimation (TEBD) algorithm with 2D isoTNSs (TEBD<sup>2</sup>) for real-time evolution. We show that one can study ground state properties and simulate the dynamics of the 2D systems with isoTNSs. In Sec. 3.3, we demonstrate both methods introduced by computing the dynamical spin structure factor of 2D quantum spin systems. First, we compare our results for the transverse field Ising model on a square lattice with the prediction by spin-wave theory. Second, we consider the Kitaev model on the honeycomb lattice and compare it to the result from the exact solution. We conclude by discussing these results in Sec. 3.4.

## 3.1. Extended study on Moses move

We have reviewed two complementary approaches shifting the orthogonality hypersurface  $\Psi^{[l]} = A^{[l]}\Psi$  in Sec. 2.3. The sequential MM focuses on solving the optimal tripartite decompositions locally and building up the solution from these decompositions. In Sec. 3.1.1 and Sec. 3.1.2, we study the effect of the cost functions on the tripartite decompositions and the MM. The variational MM treats the whole shifting as a variational problem and solves it with the alternative least square updates. The original MM [75] transform the isoTNSs in the form  $\Psi^{[l]}B^{[l+1]}$  to  $A^{[l]}\Psi B^{[l+1]}$ . Going from  $A^{[l]}\Psi B^{[l+1]}$  to  $A^{[l]}\Psi^{[l+1]}$ , we apply standard variational matrix-product operator (MPO)-matrix-product state (MPS) compression. Similar to the numerical algorithms in one-dimensional (1D), the move from  $\Psi^{[l]}B^{[l+1]}$  to  $A^{[l]}\Psi^{[l+1]}$  is the fundamental step. In Sec. 3.1.3, we describe the implementation detail of MM and extend the variational MM between the two-columns wavefunction before  $\Psi^{[l]}B^{[l+1]}$  and the two-columns wavefunctions after  $A^{[l]}\Psi^{[l+1]}$ .

### 3.1.1. Sequential Moses move with alternative cost functions

In the following, we consider two classes of cost functions. Regardless of the cost function in consideration, the gradient of the unitary can be computed by either analytical derivation or auto-differentiation scheme [68]. One can solve the optimization problem using Riemannian gradient descent or Newton-based methods over the Stiefel manifold [171–175]. We give an overview of these procedures in Appendix A.1.3.

### Entanglement entropies as cost functions

The first class of cost functions we consider are entanglement entropies. In [75], Rényi- $\alpha$  entanglement entropies  $\frac{\alpha}{1-\alpha} \log [\text{Tr}(\rho^\alpha)]$  are chosen as the cost function, where  $\rho$  is the reduced density matrix from the bipartition of the tensor  $\Theta_{(c,s_l),(b,s_r)}$  in Fig. 2.4 and  $\alpha = \frac{1}{2}$  or 2<sup>1</sup>. Choosing Rényi- $\frac{1}{2}$  entropy as a cost function is justified by the fact that Rényi- $\alpha$  entropy with  $\alpha < 1$  upper bounds the truncation error for a fixed bond dimension [40]. With  $\alpha > 1$ , the optimization, however, would not give a certified bound on the truncation error. Such a cost function is still often considered in the literature since the optimization is more straightforward. For example, the optimization converges to a minimum quickly with the Evenly-Vidal algorithm [159] due to the cost function landscape [175]. See Appendix A.1.2. Therefore, it is common to consider Rényi- $\alpha$  entropy with  $\alpha = 2$  as an alternative cost function or initialization for optimization with  $\alpha < 1$ .

### Truncation error as cost functions

The second class of cost functions is simply the truncation error set by the maximum bond dimension  $D$ ,

$$\varepsilon_D = \sum_{i=D+1}^{\infty} \Lambda_{i,i}^2. \quad (3.1)$$

Direct minimization of the truncation error is possible and may be more effective than minimizing an upper bound or surrogate cost function in some cases.

As an illustrative example, we consider the case when the random tensor  $\Theta_{(c,s_l),(s_r,b)}$  of size  $(6, 4, 4, 6)$  has a completely flat spectrum, i.e., constant singular values in the singular value decomposition (SVD) between the indices pair  $(c, s_l), (s_r, b)$ . We evaluate different cost functions with the final unitaries  $U$  found by minimizing Rényi- $\alpha$  entropy and the truncation error  $\varepsilon_D$ . The overall result is shown in Table 3.1, and the singular values after the optimization are plotted in Fig. 3.1.

Comparison of disentangling approach and direct minimization of truncation error							
Result Minimize	Rényi-2	Rényi-1	Rényi- $\frac{1}{2}$	$\varepsilon_{D=4}$	$\varepsilon_{D=8}$	$\varepsilon_{D=12}$	$\varepsilon_{D=16}$
–	3.18	3.18	3.18	0.833	0.667	0.5	0.333
Rényi-2	<b>1.56</b>	2.09	2.53	0.28	0.116	0.0432	0.0114
Rényi-1.5	1.58	2.06	2.5	0.26	0.1	0.0364	0.00945
Rényi-1	1.72	<b>2.04</b>	2.41	0.236	0.0645	0.0181	0.00344
Rényi-0.75	1.86	2.09	<b>2.35</b>	0.266	0.0608	0.00727	0.000507
Rényi-0.5	1.94	2.16	2.36	0.32	0.0727	0.00656	7.59e-06
$\varepsilon_{D=4}$	1.8	2.12	2.5	<b>0.208</b>	0.0964	0.0373	0.0108
$\varepsilon_{D=8}$	2.02	2.17	2.43	0.374	<b>0.0336</b>	0.0133	0.00407
$\varepsilon_{D=12}$	2.26	2.37	2.47	0.465	0.165	<b>0.000714</b>	0.000184
$\varepsilon_{D=16}$	2.52	2.62	2.69	0.561	0.273	0.087	<b>9.95e-17</b>

Table 3.1.: The resulting values for Rényi- $\alpha$  entropy and truncation error  $\varepsilon_D$  of  $U\Theta$  after minimizing Rényi- $\alpha$  or  $\varepsilon_D$  as the cost function. The first row gives values for the original tensor  $\Theta$  before optimization. The minimal value of each column is highlighted in boldface.

<sup>1</sup>Similar variational problems of “disentangling” also show up in various contexts. In [176], one finds the minimal entanglement representation of MPSs for purified states by utilizing the gauge degree of freedom of the ancilla space in the purification, where Rényi-2 entropy is considered. In a different context of interacting fermionic system, the local mode transformation looks for the optimal unitary leading to the smallest truncation in MPS representation by minimizing the Rényi- $\frac{1}{2}$  entropy [177, 178].



We observe that all the entanglement entropies and the truncation errors  $\varepsilon_D$  become smaller after minimization for different cost functions considered, compared to the original values before optimization. This suggests that exploiting the gauge degree of freedom and inserting the unitary is beneficial regardless of the cost functions chosen. Additionally, we generally observe a trade-off: minimizing entanglement entropies results in lower final entanglement entropies but slightly higher final truncation errors. Conversely, minimizing truncation errors results in lower final truncation errors but slightly higher final entanglement entropies. Overall, the distribution of the resulting singular values depends on the cost function chosen and could have orders of magnitude differences in truncation error.

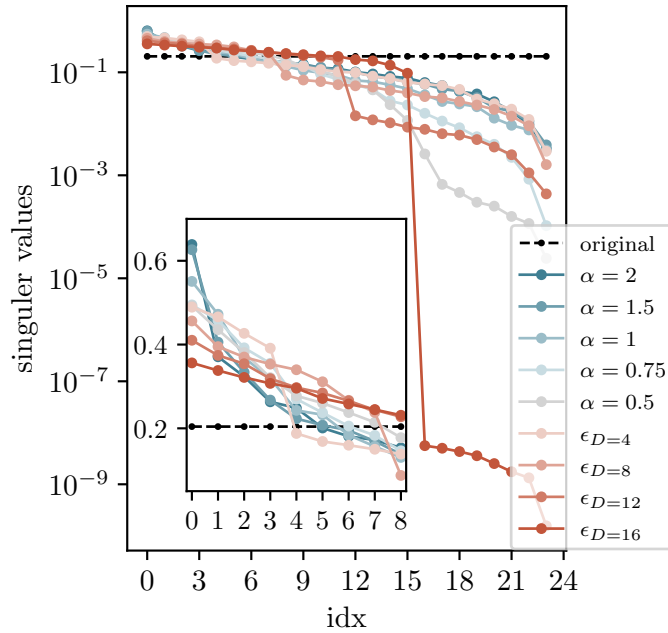


Figure 3.1.: Singular values of  $U\Theta$  after optimization with respect to  $U$  using Rényi- $\alpha$  and  $\varepsilon_D$  as cost functions, where  $\Theta$  is a random tensor with flat singular values spectrum. The inset shows a zoom-in view.

Choosing Rényi- $\alpha$  as a cost function, we observe in Fig. 3.1 that the singular values modified vary “smoothly”. For smaller  $\alpha$ , we see smaller singular values in the tail, while for larger  $\alpha$ , we see the first few singular values have larger values. This means minimizing Rényi- $\alpha$  entropy for a smaller  $\alpha$  leads to better results when the truncated bond dimension  $D$  is large, while minimizing a larger  $\alpha$  leads to better results when the truncated bond dimension  $D$  is small. This is consistent with the fact that the Rényi- $\alpha$  entanglement entropy for  $\alpha \rightarrow \infty$  is equivalent to  $\varepsilon_{D=1}$ . Overall, the optimal  $\alpha$  depends on the truncated bond dimension. It is worth noting that, in all cases, minimizing the Rényi- $\alpha$  entropy leads to the smallest Rényi- $\alpha$  entropy but could have a larger truncation error compared to the result of minimizing  $\varepsilon_D$ .

In contrast to the “smooth” change in singular values when minimizing the entanglement entropy, the singular values obtained from minimizing  $\varepsilon_D$  show a sharp drop at the corresponding bond dimension  $D$ . The optimization takes into account the information of the specific bond dimension  $D$  and pushes down all singular values afterward to minimize the truncation error. For a given bond dimension  $D$ , we see that minimizing  $\varepsilon_D$  always leads to the smallest truncation error  $\varepsilon_D$  even while it may have larger entanglement entropy.

The moral we learned from this illustrative example is that the conventional way of “disentangling” a tensor  $\Theta$  by minimizing entropy indeed brings down the truncation error,

but (at least locally) in a sub-optimal way. Disentangling modifies the overall spectrum but does not utilize the information about the anticipated truncated bond dimension  $D$ . The direct minimization of truncation error utilizes such information to avoid the ambiguity in choosing  $\alpha$  and thus could potentially lead to smaller truncation errors.

In practice, the spectrum for a physical system would not be a constant but decay exponentially. We observe similar results that direct minimization of truncation error gives a slightly smaller truncation error than the disentangling approach (See Sec. 3.1.2). However, the optimization problems are prone to get stuck at local minima when minimizing truncation error, while minimizing entanglement entropy is generally more robust. As a result, we only report the results based on minimizing Rényi- $\frac{1}{2}$  entanglement entropy in this work.

### 3.1.2. Comparison of Moses moves

We consider the similar test on truncation error for a 2-column physical wavefunction <sup>2</sup> as in [75] for MM  $\Psi^{[1,2]} \rightarrow A^{[1]}\Psi^{[2]}$  minimizing two different cost functions, Rényi- $\frac{1}{2}$  and  $\epsilon_\eta$ . The wavefunction considered is the ground state of the transverse field Ising model on a two-column ladder of size  $2 \times 20$  obtained from DMRG. The coupling strength is set to be different between the horizontal and vertical bonds, i.e.,  $H = \sum_i g \sigma_i^x - \sum_{\langle i,j \rangle_h} J_h \sigma_i^z \sigma_j^z - \sum_{\langle i,j \rangle_v} J_v \sigma_i^z \sigma_j^z$ , where  $g = 2.5, J_h = 0.5, J_v = 1.5$ .

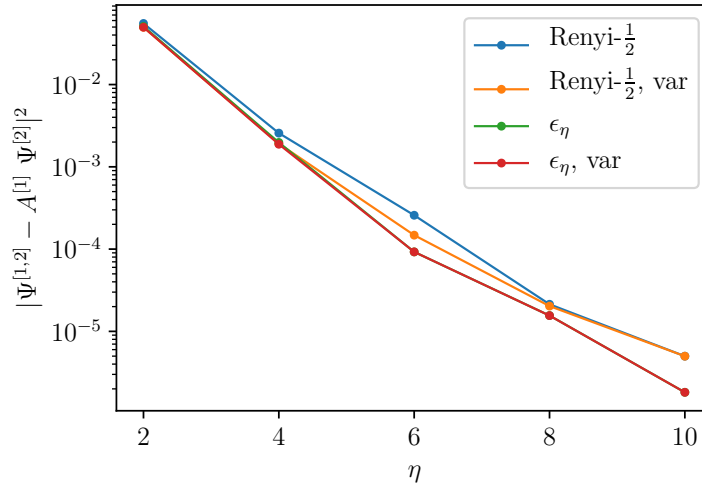


Figure 3.2.: Comparison of the error in the decomposition  $\|\Psi^{[1,2]} - A^{[1]}\Psi^{[2]}\|^2$  for the MM with minimization based on Rényi- $\frac{1}{2}$  entropy and truncation error  $\epsilon_\eta$ . Furthermore, the variationally optimized solutions are also included.  $\eta$  is the vertical bond dimension of  $\Psi^{[2]}$ . The test state  $\Psi^{[1,2]}$  is the ground state of the TFI model on a two-columns ladder.

The MM factorizes the 2-column wavefunction  $\Psi^{[1,2]}$  into  $A^{[1]}\Psi^{[2]}$ , where [1] denotes all physical indices for the first column and [2] for the second column. We consider the  $A^{[1]}$  column to have fixed horizontal bond dimension  $D_H$  and vertical bond dimension  $D_V$  with  $D_H = D_V = 2$ , while for  $\Psi^{[2]}$  column we have  $D_H = 2$  and  $D_V = \eta$ . We compare the result of two different cost functions in the tripartite decomposition as described in Sec. 2.3.3 and

<sup>2</sup>We represent the 2-column wavefunction as an MPS where there are two physical indices at each site. We can place one index to the left and one to the right of the column, resulting in the standard form of the input for MM.

Comparison of disentangling approach and direct minimization of truncation error							
Result Minimize	Rényi-2	Rényi-1	Rényi- $\frac{1}{2}$	$\varepsilon_{\eta=2}$	$\varepsilon_{\eta=4}$	$\varepsilon_{\eta=6}$	$\varepsilon_{\eta=8}$
–	0.386	0.575	0.914	0.0198	0.00263	0.000178	1.28e-05
Rényi-2	<b>0.266</b>	0.429	0.749	0.00751	0.00103	9.06e-05	9.59e-06
Rényi-1.5	<b>0.266</b>	0.426	0.74	0.00673	0.000973	9.23e-05	9.44e-06
Rényi-1	0.268	<b>0.425</b>	0.727	0.00558	0.000943	0.000105	9.19e-06
Rényi-0.75	0.272	0.426	0.72	0.00494	0.000885	0.000109	9.22e-06
Rényi-0.5	0.278	0.431	<b>0.718</b>	0.00461	0.000767	7.49e-05	9.88e-06
$\varepsilon_{\eta=2}$	0.283	0.435	0.719	<b>0.00437</b>	0.000623	6.43e-05	9.35e-06
$\varepsilon_{\eta=4}$	0.289	0.441	<b>0.717</b>	0.00446	<b>0.000579</b>	4.44e-05	7.34e-06
$\varepsilon_{\eta=6}$	0.296	0.449	<b>0.718</b>	0.00481	0.000665	<b>1.99e-05</b>	1.45e-06
$\varepsilon_{\eta=8}$	0.298	0.452	0.725	0.00511	0.000842	2.28e-05	<b>1.27e-06</b>

Table 3.2.: The result for tripartite decomposition from a tensor taken in the middle of MM for the two-column wavefunction described in Sec. 3.1.2. We compare the resulting values for Rényi- $\alpha$  entropy and truncation error  $\varepsilon_D$  of  $U\Theta$  after minimizing Rényi- $\alpha$  or  $\varepsilon_D$  as the cost function. We use a bond dimension  $D_V = \eta$  for the column  $\Psi^{[2]}$ , so we use the notation  $\varepsilon_\eta$  instead. The values in the first row are from the original tensor  $\Theta$ . The best values of each column are highlighted in boldface.

measure the performance of MMs by  $\|\Psi^{[1,2]} - A^{[1]}\Psi^{[2]}\|^2$  which depends on  $\eta$  as shown in Fig. 3.2. Choosing the truncation error  $\varepsilon_\eta$  as a cost function gives slightly better results, and the variational Moses move based on such initialization barely improve.

To understand the effect of cost functions in the tripartite decomposition for physical states, we take one tensor from the middle of the column during the MM and repeat the benchmark of the tripartite decomposition on this tensor based on all different cost functions introduced in Sec. 2.3.3. We show the resulting truncation error and entanglement entropy in Table 3.2. Similar to the observation in Sec. 2.3.3, disentangling and direct minimizing truncation error both lead to smaller truncation errors, while direct minimizing truncation error gives a slightly better result. Unlike the test in Table 3.1, we see exponential decay in the truncation error  $\varepsilon_\eta$  for the original tensor in the first row. This indicates that the singular values of the original tensor decay exponentially. Utilizing the gauge degree of freedom to perform disentangling or direct minimizing truncation error could still substantially improve the truncation error.

### 3.1.3. Implementation details

#### Optimization

In practice, the results for MMs depend both on the cost function and the optimization, and the optimization is generically not guaranteed to converge to the global minimum or stopped early due to efficiency considerations. We consider two different types of first-order Riemannian optimization methods: Riemannian Adam [179] and Riemannian non-linear conjugate gradient method [180] with line search. Setting Rényi- $\frac{1}{2}$  entanglement entropy as the cost function, both optimization methods give similar convergent results. When the truncation error is set as the cost function, the Riemannian Adam yields a slightly lower truncation error compared to using Rényi- $\frac{1}{2}$  entanglement entropy as the cost function. However, setting the truncation error as the cost function, the Riemannian conjugate gradient method with line search often gives worse results, i.e., higher truncation error, than the result of having Rényi- $\frac{1}{2}$  entanglement entropy as cost function. Although the Riemannian Adam optimization with truncation error as a cost function gives slightly

better results, it has a slower convergence rate. Moreover, the Riemannian Adam method does not include a line search procedure but is sensitive to set step size and requires problem-specific step size tuning. As a result, we consider the Riemannian conjugate gradient methods with Rényi- $\frac{1}{2}$  entanglement entropy as a cost function in this work for efficiency reasons.

In practice, we find the tensors from the consecutive SVDs, update the isometry with unitary found by linearizing the Rényi-2 entanglement entropy until convergence, and finally, we perform conjugate gradient method with Rényi- $\frac{1}{2}$  entanglement entropy. The above scheme is not guaranteed optimal but is stable.

### Bond dimensions and computational complexity

In practice, it is observed that increasing the bond dimensions on the orthogonality hypersurface could increase the representation power with less cost compared to increasing the bond dimensions uniformly. As a result, we consider a maximal bond dimension  $D$  throughout the tensor network and a maximal bond dimension  $\eta$  on the orthogonality hypersurface. With this setup, MM would decompose the column  $\Psi^{[l]}$  with bond dimension  $\eta$  into two new columns  $A^{[l]}$  and  $\Psi$  with bond dimension  $D$  and  $\eta$  respectively<sup>3</sup>. The computation complexity of MM is  $\mathcal{O}(\eta^3 D^4 + \eta^2 D^5)$ , including the variational Moses move.

After the MM, we combine  $\Psi$  column and  $B^{[l+1]}$  column to form the new  $\Psi^{[l+1]}$  column. This step is similar to the standard MPO-MPS contraction. The direct contraction and truncation by randomized SVD have complexity  $\mathcal{O}(\eta^2 D^5)$  and  $\mathcal{O}(\eta^3 D^4)$  respectively. Similar to MPSs compression, the one-sided truncation may lead to a sub-optimal result, and one could consider it as initialization and further improve by variationally optimizing the truncated column with  $\mathcal{O}(\eta^3 D^4)$ . Another possible way would be to consider combining the column variationally like variational MPO-MPS contraction, which gives the same structure in contraction as in variational MM but with now the single  $\Psi^{[l+1]}$  column varying. Thus, it would also have the same cost. Note that direct contraction of  $\Psi$  and  $B^{[l]}$  following standard SVD truncation would cost  $\mathcal{O}(\eta^3 D^5)$ , which should be avoided. With the above counting, we show that shifting the columns from  $\Psi^{[l]} B^{[l+1]}$  to  $A^{[l]} \Psi^{[l+1]}$  has complexity  $\mathcal{O}(\eta^3 D^4 + \eta^2 D^5)$  in general.

After MM and combining the columns, i.e.,  $\Psi^{[l]} B^{[l+1]} \rightarrow A^{[l]} \Psi^{[l+1]}$ , an optional step to improve overlap can be considered by variationally maximizing

$$\mathcal{F} = |\langle \Psi_{\text{before}} | \Psi_{\text{after}} \rangle|^2 = |\langle \Psi^{[l]} B^{[l+1]} | A^{[l]} \Psi^{[l+1]} \rangle|^2$$

again over the two new column  $A^{[l]}$  and  $\Psi^{[l+1]}$ <sup>4</sup>. This variational optimization has a contraction structure of four columns and the computation complexity  $\mathcal{O}(\eta^3 D^4 + \eta^2 D^6)$ . In this work, we adapted all three variational procedures as in Fig. 2.3 when the numerical cost is acceptable. Notice that even with the optional variational step, the overall computational complexity for the time evolution algorithm would still be cheaper than full update [64] if the bond dimension in central column  $\eta$  does not grow as  $\mathcal{O}(D^2)$ .

One can reduce the computational complexity further by decomposing the order-4 isometry into two trivalent tensors (omitting the physical indices). A similar strategy is considered in the so-called triad network in the context of tensor network renormalization

<sup>3</sup>For the schematic drawing, see Fig. 2.3.

<sup>4</sup>The variational optimization over two new column  $A^{[l]}$  and  $\Psi^{[l+1]}$  is different and more precise than the original variational Moses Move [75]. In more recent works [2, 101], another different approach from the original MM is used. The new approach first groups two physical columns  $\Psi^{[l]}$  and  $B^{[l+1]}$  to form  $\Psi^{[l,l+1]}$  and then perform MM on the two-column wavefunction to split it into  $A^{[l]}$  and  $\Psi^{[l+1]}$ . This two-column approach is related to the two-column variational approach here and gives better numerical results.

group [181]<sup>5</sup>. The scaling could then be reduced to  $\mathcal{O}(\eta^3 D^3)$ . The trade-off for less representation power is compensated by the potential advantage of working with larger bond dimensions. It has shown to be advantageous in the context of tensor renormalizations.

### 3.2. DMRG<sup>2</sup> algorithm

With the isometric form generalized to two dimensions, we can easily adapt the 1D tensor network states (TNSs) algorithms. As seen in Chapter 2 Sec. 2.3.4, the time evolution algorithm TEBD<sup>2</sup> mimics the TEBD algorithm in 1D for MPSs. This section describes the ground state search algorithm by variational energy minimization – DMRG<sup>2</sup>. TEBD<sup>2</sup> and DMRG<sup>2</sup> algorithms can be formulated as the minimization problems of certain cost functions for local tensors. In this fashion, both algorithms are iterative algorithms performing local updates over each tensor. Here we consider the case where the tensor updated is always at the orthogonality center by shifting the orthogonality hypersurface.

Extending the DMRG algorithm for 2D TNSs is considered in [60, 182] with the main drawbacks of high complexity  $\mathcal{O}(D^{12})$  and numerical instability<sup>6</sup>. Here, we first review the general energy minimization approach with TNSs and then discuss the difference when isoTNSs were used. The energy minimization algorithm with 2D isoTNSs is dubbed DMRG<sup>2</sup> as it resembles the 1D DMRG algorithm.

The energy minimization problem is solved by an iterative local update on each tensor  $x$  with

$$x^{\text{update}} \leftarrow \arg \min_x \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle}. \quad (3.2)$$

By introducing the Lagrangian multiplier  $\lambda_E$ , the problem can be rewritten as,

$$\partial_{x^*} \langle \psi | \hat{H} | \psi \rangle - \partial_{x^*} \lambda_E \langle \psi | \psi \rangle = 0, \quad (3.3)$$

and the solution of the optimization problem on a single tensor  $x$  is given by the generalized eigenvalue problem

$$H_{\text{eff}} x = \lambda_E N x, \quad (3.4)$$

where  $H_{\text{eff}}$  is the contraction of energy expectation value  $\langle \psi | \hat{H} | \psi \rangle$  with leaving  $x$  and  $x^*$  tensors out. The  $N$  is the norm matrix, as defined in Sec. 2.3, which is the contraction of the norm  $\langle \psi | \psi \rangle$  leaving the  $x$  and  $x^*$  tensors out.

The crucial difference between considering TNSs and isoTNSs as variational ansätze is that the generalized eigenvalue problem reduces to a standard eigenvalue problem with isoTNSs. This is because the norm matrix of the orthogonality center is an identity operator  $N = \mathbb{1}$  by the isometric condition. See Fig. 3.3. This simplifies the computation and stabilizes the algorithm since the ill-conditioned generalized eigenvalue problem may return infinite or ill-disposed eigenvalues [182].

In practice, we implement the Hamiltonian as a sum of 1D MPOs over the rows and columns. The expected energy is a sum of the contraction over the double layer isoTNSs and MPOs. Similarly, the  $H_{\text{eff}}$  is a sum of the contraction over the isoTNS and MPOs while leaving the tensor  $x$  and  $x^*$  (See Fig. 3.3). We contract each term approximately using the boundary matrix-product operators (bMPOs) approach [64]. The accuracy of the approximation is controlled by the bond dimension  $D_{\text{bMPO}}$  of the bMPO, and the overall cost of contracting the bMPO is  $\mathcal{O}(D^6 D_{\text{bMPO}}^2 + D^4 D_{\text{bMPO}}^3)$ . In the following, we

<sup>5</sup>This is similar to approximated gate decomposition in the language of the quantum circuit. The consideration of sparse structure within the tensor network is an active research direction, which we discuss further in the second part of the thesis, particularly in Chapter 6.

<sup>6</sup>A recent proposal on fixing the issue of instability for general TNSs [97] shows potential to be applied to large system sizes.

take  $D_{\text{bMPO}} = 2D^2$  and explicitly construct the matrix  $H_{\text{eff}}$  with the cost  $\mathcal{O}(D^{12})$ . Note that, in principle, only the matrix-vector multiplication  $H_{\text{eff}}x$  operation is required for solving the eigenvalue problem. The complexity can be reduced to  $\mathcal{O}(D^{10})$  if  $H_{\text{eff}}$  were not constructed explicitly. It is possible that the approach above may be improved by the advanced optimal MPO compression scheme in [183].

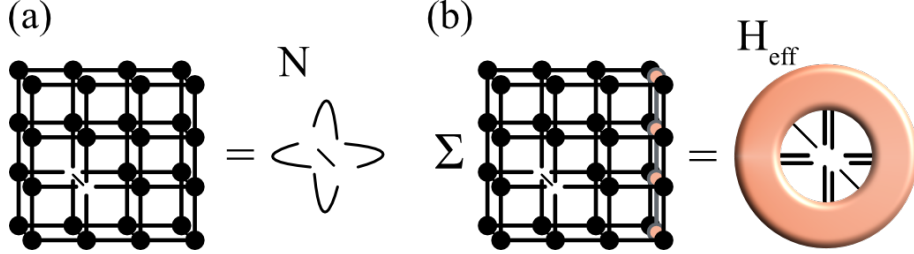


Figure 3.3.: The generalization of DMRG algorithm to 2D isoTNSs: (a) The norm matrix  $N$  of isoTNSs is an identity operator  $N = \mathbb{1}$  when the update site is the orthogonality center. (b)  $H_{\text{eff}}$  is constructed by summing up the contraction of MPO representation of the Hamiltonian in the rows and columns.

We sketch the outline of the DMRG<sup>2</sup> algorithm here. It is similar to the TEBD<sup>2</sup> algorithm but replaces the local update with solving an eigenvalue problem.

- (i) Start with an isoTNS with the orthogonality center at the left-most column.
- (ii) Perform the 1D DMRG over the column. That is, we update each tensor  $x$  in the column by solving the standard eigenvalue problem  $H_{\text{eff}}x = \lambda x$  on the orthogonality center to obtain the lowest eigenvector  $x^{\text{update}}$ . We move the orthogonality center from site to site by SVD.
- (iii) Perform MM to bring the orthogonality center forward to the next column, then repeat the 1D DMRG as in step (ii). Continue and repeat these steps over all columns.
- (iv) At the end of the sweep, all tensors are updated. We perform a similar trick of rotation or a horizontal reflection to bring the direction of isometries back to the starting arrangement as in step (i).

The above steps give one DMRG<sup>2</sup> sweep updating over all tensors. The algorithm continues until the energy converges.

We would like to point out that the DMRG<sup>2</sup> proposed here is not a standard variational algorithm, which optimizes over the parameters of a single variational wavefunction. Instead, we optimize over a set of quantum states approximately connected by MM. While we can variationally update the tensors in the column with an orthogonality center to improve the result, we always introduce an approximation (truncation) error when we move on with MM to optimize the tensors in the next column. As a result, the variational energy does not monotonically decrease as in the standard DMRG algorithm, which is observed later in our numerical experiment and similarly in Ref. [163]. The approximation of the ground state energy found by DMRG<sup>2</sup> is, thus, similar to TEBD<sup>2</sup>, bound by the MM error.

Another possible way to carry out DMRG<sup>2</sup> with isoTNSs is to fix the isometric structure and not perform MM when one sweeps through the lattice. In that case, one has to compute the  $N$  and solve for generalized eigenvalue problems. One expects better condition numbers compared to the case without gauge fixing. With this approach, one can study the representation power of the isoTNSs since there is no truncation involved. Our observation is that the isometric condition itself, without gauge fixing, is still not stable. Therefore, for the application in this work, we consider the former approach only.



### Benchmark on transverse field Ising model

To benchmark both TEBD<sup>2</sup> and DMRG<sup>2</sup> algorithms discussed, we consider the transverse field Ising (TFI) model on the square lattice defined as

$$H_{TFI} = -J \left( \sum_{\langle i,j \rangle} \hat{\sigma}_i^x \hat{\sigma}_j^x - g \sum_i \hat{\sigma}_i^z \right), \quad (3.5)$$

where  $\langle i, j \rangle$  denotes the nearest neighbors for site  $i, j$ . We set  $J = 1$  as the unit. In the thermodynamic limit, the TFI model exhibits a quantum phase transition from a symmetry-broken phase to a disordered phase at  $g_c \approx 3.044$ . We consider a square lattice of size  $L_x = L_y = 11$  with open boundary conditions (OBC) and  $g = 3.0$ , close to the critical point. We compare the ground state energy estimate obtained from the imaginary-time evolution using second-order TEBD<sup>2</sup> and DMRG<sup>2</sup> with the numerically exact results from 1D-DMRG simulation with bond dimension  $D = 1024$ . The result is plotted in Fig. 3.4. The bond dimension of the overall isoTNS is denoted by  $D$ , and we allow the bond dimension in the orthogonality hypersurface to be  $\eta$ . We consider the setup where  $\eta = 2D$  for TEBD<sup>2</sup> and  $\eta = D$  for DMRG<sup>2</sup>. In Fig. 3.4a, we show that for TEBD<sup>2</sup> the energy estimates do not go down monotonically with the decrease of step  $d\tau$  because of the MM error  $\varepsilon_{\text{MM}}$ . As described in the previous section, the energy error could be fitted with  $\Delta E_{\text{TEBD}^2} = a\varepsilon_{\text{MM}}/d\tau + bd\tau^{2p}$  where  $p$  is the order of Trotterization [75]. The extrapolated optimal energy estimate is given by the minimum of the fit. On the other hand,  $\Delta E_{\text{DMRG}^2} \propto \varepsilon_{\text{MM}}$ . We see, in general, DMRG<sup>2</sup> has smaller error estimates than the imaginary TEBD<sup>2</sup> even when we use the same bond dimension  $\eta = D$  over the orthogonality hypersurface. We plot the computational runtime in Fig. 3.4b. Despite the difference in the scaling of computational complexity with respect to bond dimension, we observe that TEBD<sup>2</sup> and DMRG<sup>2</sup> reach similar accuracy at a given time for isoTNSs of different bond dimensions.

Let us finally comment on the error made by repeated MMs. Note that we introduce an error  $\varepsilon_{\text{MM}}$  for each MM. Consider a state  $|\Psi_0\rangle$  with the orthogonality center in the 0<sup>th</sup>-column. By repeating the MM, we can move the orthogonality center to  $n^{\text{th}}$ -column—we denote the corresponding state as  $|\Psi_n\rangle$ . The accumulated error of repeating the MM results in a deviation from the original state, which can be measured by the fidelity between the state  $|\Psi_n\rangle$  and the original state  $|\Psi_0\rangle$ , i.e.,  $\mathcal{F} = |\langle \Psi_0 | \Psi_n \rangle|^2 \approx (1 - \varepsilon_{\text{MM}})^n$ .

We take the  $D = \eta = 2$  isoTNS obtained from the DMRG<sup>2</sup> for the TFI model with  $g = 3.0$  on an  $11 \times 11$  square lattice and repeatedly perform MM using bond dimension  $D', \eta'$  sweeping from left to right and then from right to left. The result is shown in Fig. 3.5. We indeed observe a decrease of  $\mathcal{F}$  to  $n$  in Fig. 3.5a as we sweep from left to right. However, we find that sweeping from the right back to the original 0<sup>th</sup>-column does not cause a further decrease in fidelity. In fact, the fidelity  $\mathcal{F}$  even increases in some cases. We measure the corresponding energy for state  $|\Psi_n\rangle$  and plot the error in Fig. 3.5b. The error density  $\varepsilon_{\text{MM}}$  introduces a bound on the accuracy that can reach by the algorithms. Continuing the left-right sweep of MM more than once leads to further degradation in fidelity. But we also find an approximate fixed point in some cases, where the fidelity almost converges with the continuing left-right sweep of MM. We show the data in Appendix A.3.

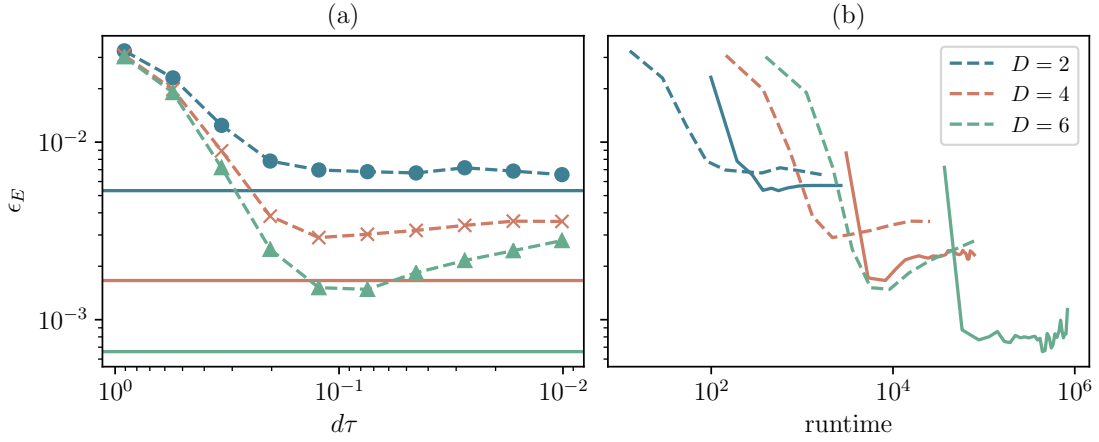


Figure 3.4.: The result from the TEBD<sup>2</sup> and DMRG<sup>2</sup> algorithms for the TFI model with  $g = 3.0$  on an  $11 \times 11$  square lattice. (a) The relative energy error  $\epsilon_E = (E - E_{\text{exact}})/|E_{\text{exact}}|$  is plotted as function of the Trotter step size  $d\tau$ . We show data of different bond dimensions  $D$  in different colors. The solid lines in the background represent the error of DMRG<sup>2</sup> with the bond dimension  $D$  of the same color. (b) We plot the relative energy error  $\epsilon_E$  against the runtime with TEBD<sup>2</sup> in dotted lines and DMRG<sup>2</sup> in solid lines. The bond dimension on the orthogonality hypersurface is all  $\eta = 2D$  for TEBD<sup>2</sup> and  $\eta = D$  for DMRG<sup>2</sup>.

### 3.3. Spectral functions

As an application of the algorithms introduced above, we consider numerical evaluation of the dynamical spin structure factor (DSF), a.k.a. the spectral function. The DSF is defined as

$$S^{\alpha\alpha}(\mathbf{k}, \omega) = \frac{1}{2\pi} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \int_0^\infty 2\text{Re} \left[ e^{i\omega t} \mathcal{C}^{\alpha\alpha}(\mathbf{R}, t) \right] dt, \quad (3.6)$$

where the correlation functions  $\mathcal{C}^{\alpha\alpha}(\mathbf{R}, t) = \langle \hat{\sigma}_{\mathbf{R}}^{\alpha\dagger}(t) \hat{\sigma}_{\mathbf{0}}^{\alpha}(0) \rangle$  is evaluated with respect to the ground state  $|\psi_0\rangle$ . It is of special importance since it gives us direct insight into the physical properties of the quasi-particles and the spectral properties of the Hamiltonian. In addition, DSF can be measured by inelastic neutron scattering in experiments and can be computed using various methods theoretically.

Here, we compute  $S^{\alpha\alpha}$  numerically following the definition Eq. (3.6). We first obtain the ground state  $|\psi_0\rangle$  by DMRG<sup>2</sup> with isoTNSs. Then the locally perturbed state  $\hat{\sigma}^{\alpha}|\psi_0\rangle$  is evolved using the TEBD<sup>2</sup> algorithm. Once we have the ground state  $|\psi_0\rangle$  and the time evolved state  $e^{-i\hat{H}t}\hat{\sigma}^{\alpha}|\psi_0\rangle$ , the time dependent real space correlation function  $\langle \hat{\sigma}_{\mathbf{R}}^{\alpha\dagger}(t) \hat{\sigma}_{\mathbf{0}}^{\alpha}(0) \rangle$  is obtained by the approximate contraction of TNSs. For the data shown in the next section, we apply linear prediction to double the time simulated from  $T$  to  $2T$ . In all cases, we multiply the data with a Gaussian ( $\sigma_t \approx 0.44T$ ), which corresponds to a decay of factor 10 at time  $T$ , and effectively smooth out and broaden the data in the frequency space. Finally, the double Fourier transform of the correlation function gives us the spectral function. We plot the spectral function in a logarithmic scale with a cutoff chosen to avoid showing the noise.

For the isoTNS with bond dimension chosen here, we observed MM error around  $\epsilon_{\text{MM}} = \|\psi - \psi'\|^2 \sim 10^{-2}$  for sweeping the central column from left to right. The corresponding



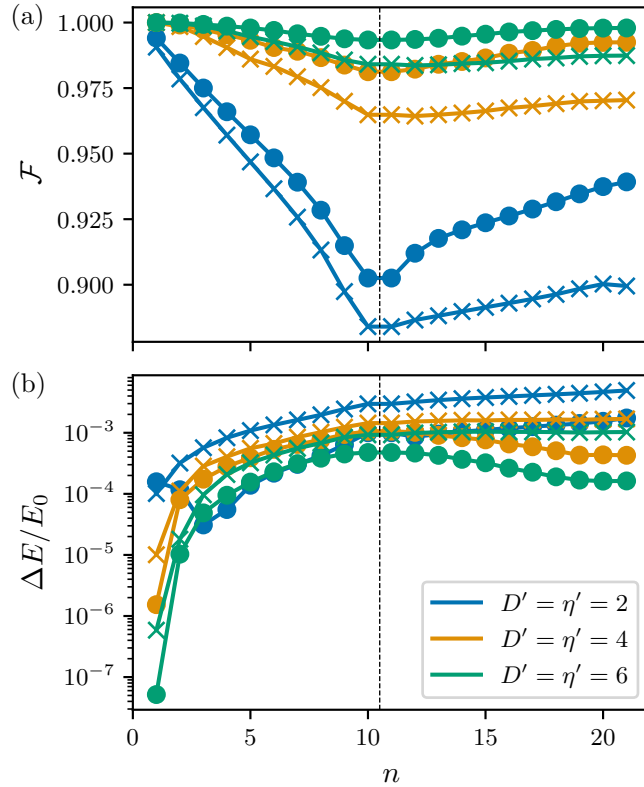


Figure 3.5.: Benchmark on repeating the MM over a  $D = \eta = 2$  isoTNS representing the ground state of the TFI model with  $g = 3.0$  on an  $11 \times 11$  square lattice. (a) The fidelity  $\mathcal{F}$  between the original state  $|\Psi_0\rangle$  and the state  $|\Psi_n\rangle$  after  $n$  MM. (b) The relative energy difference of the state  $|\Psi_n\rangle$  compared to the original state. We perform in total  $2L - 1 = 21$  MM, sweeping from left to right and right to left. The MM is carried out in different bond dimensions  $D', \eta'$  plotted in different colors. The solid circle and the cross mark the data obtained with variational MM and without variational MM. The fidelity and the energy are measured by bMPO method with  $D_{\text{bMPO}} = 4\eta'^2$ .

two errors made for first-order and second-order TEBD<sup>2</sup> are similar. For easier evaluation of the time-dependent correlation function, we choose the first-order TEBD<sup>2</sup> method here <sup>7</sup>. We consider two different models to demonstrate the methods introduced could give qualitative insight into the physical Hamiltonian.

### 3.3.1. Transverse field Ising model

The TFI model, defined in Eq. (3.5), is a paradigmatic model for studying quantum many-body systems. The ground state is ferromagnetically ordered for  $g$  smaller than  $g_c \approx 3.044$  [184, 185], and a disordered phase for  $g > g_c$ . The excitation spectrum is known perturbatively in the large and small- $g$  limit by the effective Hamiltonian method. We compute the DSF  $S^{yy}$  on a square lattice of size  $L_x = L_y = 11$  for  $g = 1$  to  $g = 5$ . We plot the DSF result from the simulation and the perturbative calculation in Fig. 3.6.

In the limit  $g \ll 1$ , the single-particle (magnon) excitation consists of a single spin flip costing energy  $\sim 8J$ . To lowest non-vanishing order in  $g$ , we find a nearest-neighbor

<sup>7</sup>With  $\varepsilon_{\text{MM}} \sim 0.01$  and  $T \sim 1$ , for first-order method, we have optimal around  $\delta t \sim 0.21$  and total error  $\sim 0.09$ . And for the second-order method, total error  $\sim 0.05$ ,  $\delta t = 0.4$ .

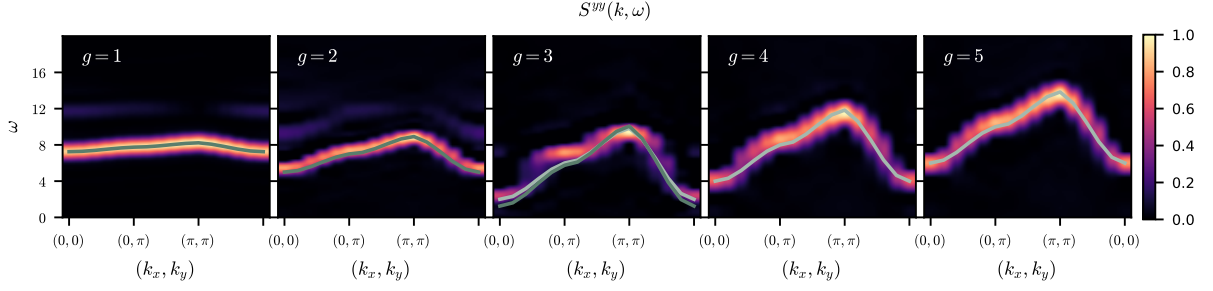


Figure 3.6.: The dynamical structure factor  $S^{yy}(\mathbf{k}, \omega)$  for  $g = 1$  to  $g = 5$  in logarithmic color scale. The dark (light) green curve indicates the dispersion calculated perturbatively from the limit  $g \ll 1$  ( $g \gg 1$ ).

hopping model with energy,

$$\varepsilon_{g \ll 1} = 8 - \frac{g^2}{4}(1 + \cos(k_x) + \cos(k_y)) + \mathcal{O}(g^3). \quad (3.7)$$

The two-magnon excitations in the ferromagnetic phase form a bound state. By simple counting, these bound states, which consist of two flipped spins on nearest neighbor sites, have energy  $12J$  (lower than the two-particle continuum  $\varepsilon \sim 16J$ ). For the  $g = 1$  and  $g = 2$  plots in Fig. 3.6 the dispersion obtained by the simulation matches the result from the perturbative calculation. Moreover, we can see a slight signal of the bound states. However, throughout the full range of coupling, the two-magnon continuum is not observed.

In the limit  $g \gg 1$ , by carrying out a similar calculation, the energy of single-particle excitation is given as,

$$\varepsilon_{g \gg 1} = g \left[ 2 - \frac{2}{g}(\cos(k_x) + \cos(k_y)) + \mathcal{O}(1/g^2) \right]. \quad (3.8)$$

We again find that the DSF matches well with the perturbative calculation.

Near the critical point  $g = 3$ , we observe a small gap. The gap size is slightly larger than the result in [186] due to the finite size effect.

### 3.3.2. Kitaev model on honeycomb lattice

As a second example, we consider the Kitaev model on the honeycomb lattice [187], consisting of three alternating spin couplings between bonds,

$$H_{\text{Kitaev}} = -J_x \sum_{\langle ij \rangle_x} \hat{\sigma}_i^x \hat{\sigma}_j^x - J_y \sum_{\langle ij \rangle_y} \hat{\sigma}_i^y \hat{\sigma}_j^y - J_z \sum_{\langle ij \rangle_z} \hat{\sigma}_i^z \hat{\sigma}_j^z \quad (3.9)$$

The Kitaev model is an exactly solvable model describing two types of quantum spin liquids depending on the couplings. The system is either a gapped  $Z_2$  spin liquid with abelian excitations or a spin liquid with gapless Majorana and gapped flux excitations. Here, we consider the isotropic coupling  $J_x = J_y = J_z$ , which belongs to the latter category.

The Kitaev model is of special importance because there are few examples of excitations of topological states that can be solved analytically. Utilizing the Majorana fermions representation, one can obtain the ground state properties and the excitations' properties. The exact solutions for the Kitaev model provided by [188, 189] for infinite systems and [190] for finite system serve as a challenging benchmark for numerical simulation of the DSF for 2D systems.

We obtain the DSF by a similar procedure with isoTNSs as before and plot the result in Fig. 3.7<sup>8</sup>. We compare the data with the exact solution [188]. The DSF at the isotropic gapless point is gapped due to the flux excitations and has a broad excitation continuum, as seen in Fig. 3.7a. In Fig. 3.7b, the simulation reproduces the gapped excitation and broad dispersiveless signal due to fractionalization, similar to the analytic result. In Fig. 3.7c, we examine the  $S^{xx}(k=0, \omega)$  more closely, and confirm the excitation to be gapped. While it is promising to see that we can qualitatively reproduce the result similar to the analytic solution, we would like to point out that the result is still severely limited by the accuracy from both DMRG<sup>2</sup> and TEBD<sup>2</sup> method. See Appendix A.3 for more details.

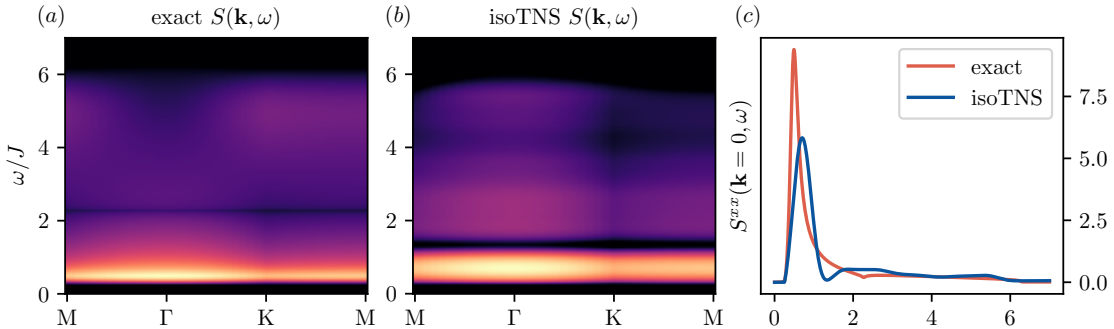


Figure 3.7.: The total dynamical structure factor  $S(\mathbf{k}, \omega) = \sum_{\alpha} S^{\alpha}(\mathbf{k}, \omega)$  for Kitaev honeycomb model at the isotropic point in logarithmic color scale. (a) The exact result obtained for infinite system size following [188]. (b) Our numerical result obtained from the simulation of system size  $11 \times 11$ . For better comparison, we perform smoothing by linear interpolation between discrete k-points. (c) The dynamical structure factor at  $\Gamma$  point, i.e.  $S_{\mathbf{k}=0, \omega}^{xx}$ .

### 3.4. Conclusion

IsoTNSs are the natural generalization of MPSs in the isometric form to higher dimensions. The isoTNSs in higher dimensions embed an effective 1D sub-region resembling the MPSs, where the known 1D algorithms can run efficiently within it. The MM is the method to shift this 1D orthogonality region introduced in [75] and reviewed in Sec. 2.3. In this work, we study the effect of different cost functions on the tripartite decomposition in the MM. Despite noting a reduced truncation error when directly minimizing it within the tripartite decomposition, the convergence outcome remains sensitive to the chosen optimization method. Therefore, further refinement is necessary before we can use the direct minimization of truncation error in practical application. We propose the DMRG<sup>2</sup> algorithm for variational energy minimization with 2D isoTNSs. Combining with the previously proposed TEBD<sup>2</sup> algorithm, we demonstrate that one can efficiently simulate the real-time evolution for 2D systems and compute the dynamical structure factors of the TFI model on a square lattice and the Kitaev model on a honeycomb lattice.

The study of isoTNSs is related to quantum computation [155, 156]. Essentially, isoTNSs are sequential and geometrically local circuit ansatz. IsoTNSs, in this perspective, are states that could be directly prepared on quantum computers. The study of the properties, e.g., variational power, of isoTNSs tells us the properties of the constant depth sequential quantum circuit [169, 170] constructed by local gates of the size of bond dimensions.

<sup>8</sup>We perform DMRG<sup>2</sup> to obtain the ground state and evolve the states by TEBD<sup>2</sup> with  $dt = 0.2$  to total time  $t = 20$ .

Algorithms for isoTNSs can be viewed as classical simulation algorithms for quantum circuits. The insight from the study of isoTNSs could potentially lead to new quantum algorithms. As we see, MMs are approximate algorithms for changing the isometric pattern in isoTNSs. In other words, they are approximate algorithms for re-ordering the quantum gates in the circuit. They have potential applications in quantum state preparation and quantum circuit compilation. One example is that applying MM iteratively on MPSs yields 1D quantum circuits. Additionally, it is shown that noisy quantum computers could be simulated efficiently classically by MPS [87]. As a generalization of MPS, 2D isoTNSs could possibly yield a better classical simulation algorithm for 2D quantum circuits.

The accuracy achieved by the isoTNSs algorithm could be improved by improving the algorithms for shifting orthogonality hypersurface. A bottom-up approach would be to develop better tripartite decomposition to directly target the truncation error and overcome the difficulties in the optimization. Alternatively, one may consider the top-down approach which optimizes the variational Moses move using different gradient-based optimization. This global update approach may result in a better minimum than the current local update approach in solving the alternating least square problem.

One promising application of isoTNSs would be to combine them with Monte Carlo methods for studying ground states and time evolution. IsoTNSs fit in Monte Carlo methods because isoTNSs allow ancestral sampling along the direction of causality, i.e. reverse direction of the arrows in the isometries. This requires only single layer TNSs contraction which is shown can be contracted at a cheaper cost  $\mathcal{O}(D^6)$  [69, 70]. Samples from ancestral sampling are independent and do not have the problem with the auto-correlation time as in Markov chain Monte Carlo (MCMC) sampling. Therefore, it may be more efficient in terms of the number of Monte Carlo sweeps and the number of samples  $N_{MC}$  compared with the approach using general TNSs [69, 70]. In this approach, there is no truncation error since the orthogonality hypersurface is held fixed and it would serve as a good test for the variational power of isoTNSs. Recently, a similar idea has been applied to general TNSs with the combination of importance sampling [71].

## 4. Isometric tensor network states on an infinite strip

Quantum phase transitions, strictly speaking, take place only in the thermodynamic limit. While some analytic solutions provide insight directly at such limits, most numerical methods provide solutions for finite systems and are typically combined with finite-size scaling analysis. Tensor network methods can provide another angle to approach the problem by directly working at the thermodynamic limit with infinite translationally invariant tensor networks [47, 66, 67, 138, 191]<sup>1</sup>. For the tensor networks considered in this thesis, i.e., matrix-product state (MPS) and projected entangled pair states (PEPS)<sup>2</sup>, the tensor network representation of the infinite translationally invariant critical states is restricted by the finite bond dimension, thus the finite entanglement, instead of the finite size. The finite entanglement induces a length scale and, similar to finite-size scaling, the theory of finite-entanglement scaling for MPS has been established [194–197]. Recently, efforts have been put into developing a similar scaling theory for PEPSs [198–200]. It is a natural question to ask whether we can also extend the study of 2D isoTNSs to the thermodynamic limit. The first step to generalize the finite two-dimensional (2D) isometric tensor network states (isoTNSs) to infinite translationally invariant systems is to consider translational invariance in one direction. Thus, we are motivated to consider the tensor network on a finite by infinite geometry with the open boundary condition, i.e., an infinite strip.

On a separate note, MPSs are often used to study quasi-2D systems with infinite lengths and finite widths, such as an infinite strip (cylinder) when open (periodic) boundary conditions are used for the finite dimension. Since an MPS is inherently a one-dimensional (1D) ansatz targeting 1D area-law states, the complexity of such quasi-2D MPS simulation grows exponentially with the width [201, 202], necessitating the extrapolation of results from strips of modest width. To avoid this exponential scaling yet retain the benefits of the isometric form to study quasi-2D systems, we are motivated to generalize the finite isoTNS and the MM algorithm on a square lattice to infinite strip geometries.

This chapter is structured as follows. We consider the generalization of the finite isoTNS on a square lattice to infinite strip geometries in Sec. 4.1 and extend the Moses move (MM) algorithm to the thermodynamic limit in Sec. 4.2. This infinite Moses move (iMM) algorithm splits a two-sided infinite matrix product state (iMPS) into an infinite isometric tensor network operator (isoTNO) and a normalized iMPS, which are all translationally invariant. In Sec. 4.2.1–Sec. 4.2.4, we propose and compare four different methods for solving the splitting problem: (i) repeated application of the finite, local MM algorithm; (ii–iii) two iterative update methods optimizing over different objective functions; and (iv) a conjugate gradient optimization maximizing the overlap. We then benchmark the various iMM algorithms in Sec. 4.3. In Sec. 4.4, we show that one can evaluate the expectation value of physical observables, e.g., energy, efficiently by utilizing the iMM algorithm. We compare the iMM approach to different methods for evaluating expectation value, including boundary matrix-product operator (bMPO) contractions methods. As a highlight of applications using iMM, we develop a generalized time-evolving block decimation (TEBD) algorithm,

---

<sup>1</sup>In the following, we refer to infinite translationally invariant tensor networks either as infinite tensor networks or uniform tensor networks.

<sup>2</sup>Here we do not include the multi-scale entanglement renormalization ansatz (MERA) [192, 193] into the discussion, which can extract the critical exponents directly due to its scale invariance structure.

dubbed iTEBD<sup>2</sup> algorithm, and investigate its performance via imaginary time evolution for finding the ground state (GS) of the 2D transverse field Ising model in Sec. 4.5. We end with the discussion and outlook in Sec. 4.6.

## 4.1. Isometric TNS on an infinite strip

In this section, we generalize the 2D finite isoTNS to infinite strip geometries. An isoTNS on an infinite strip that is uniform along the infinite direction can be represented diagrammatically as

The diagram shows a 2D tensor network representing a quantum state  $|\Psi\rangle$ . It consists of a grid of tensors. Each tensor is a square with four legs: two horizontal legs pointing left and right, and two vertical legs pointing up and down. The tensors are arranged in a regular grid. A vertical red shaded region, representing the 1D orthogonality hypersurface, passes through the center of the grid. The tensors within this region are highlighted in blue, while the others are black. The grid is bounded by vertical ellipses on the left and right sides, and horizontal ellipses at the top and bottom, indicating it extends infinitely in those directions. The equation is labeled (4.1) on the right.

The example is a  $4 \times \infty$  isoTNS, where the tensors within a column are the same but could be different for each column. Within the 1D orthogonality hypersurface (colored light red), the orthogonality center (OC) can be placed anywhere within the column using the standard uniform MPS methods [203]. In the networks we consider, we place the OC at either  $y = \pm\infty$  (given the upward pointing arrows, it is placed at positive  $\infty$  in Eq. (4.1)) so that the network has uniform vertical isometry arrows for all columns. Different from the finite case, here we work with an orthogonality “column”, instead of the +-shape orthogonality hypersurface. We note that while we only consider a single-site unit cell in this work, all algorithms presented can be extended to multi-site unit cells, allowing for periodic inhomogeneity along the infinite direction.

While boundary indices for finite isoTNS are typically trivial, as in Eq. (2.17), in general, the indices on the boundary can be non-trivial with dimension  $D_b$ , as shown in the infinite strip network Eq. (4.1). The boundary legs always carry incoming arrows as the OC is contained within the network. When  $D_b = 1$ ,  $|\Psi\rangle$  is a pure quantum state. When  $D_b > 1$ , indicating non-trivial boundary indices,  $|\Psi\rangle$  can be viewed as a purification of the density matrix of the physical sites. Typically for a bond-dimension  $\chi$  strip isoTNS, we take  $D_b = \chi$ <sup>3</sup>.

<sup>3</sup>It can be shown [2] that having non-trivial boundaries makes optimization easier and ensures that the orthogonality column, viewed as a two-sided MPS, is injective [35].

## 4.2. Infinite Moses move

Recall the finite MM algorithm performs the following decomposition of the orthogonality column as a two-sided MPS:

$$(i) \quad (ii) \quad (iii) \quad (iv) \quad (v), \quad (4.2)$$

where in the 2D isoTNS, the physical sites are grouped either with the left or the right virtual index. The MM algorithm effectively “unzips” the original MPS into an isoTNO and a new MPS. The starting point of the MM algorithm is an orthogonality column with all vertical arrows pointing down, all horizontal arrows pointing in, and the bottom-most site being the OC.

The core of the algorithm is the approximate local tripartite decomposition at each site of the column:

$$(4.3)$$

Due to the isometric form, this local decomposition is done in an orthonormal basis and is a constrained optimization problem. For details of the tri-splitting algorithm in Eq. (4.3), see [1, 75].

Similar to the case of finite isoTNS, we desire the ability to change the isometric structure and move the orthogonality column of the infinite strip isoTNS network. This would allow for the evaluation of correlation functions within the orthogonality column by 1D iMPS methods and also ensures that optimization algorithms are done in an orthonormal basis. The iMM algorithm solves the problem of splitting a normalized infinite two-sided MPS  $|\Psi\rangle$  into an infinite isoTNO  $A$  and a normalized infinite two-sided MPS  $|\Phi\rangle$ :

$$|\Psi\rangle \quad A \quad |\Phi\rangle, \quad (4.4)$$

Note that we group the physical leg with the right or left legs of the  $\psi$  tensor depending on whether we want the physical leg to be on the  $a$  tensor or the  $\phi$  tensor after splitting. In cases where  $|\psi\rangle$  represents the horizontal contraction of two columns, we distribute one physical leg to each of the left and right legs of  $\psi$  so that both  $a$  and  $\phi$  will have a physical



leg. For all iMM algorithms we present, we require the initial iMPS to have all vertical arrows pointing down, all horizontal arrows incoming, and the OC to be at  $-\infty$ . With this, the iMM can again be viewed as unzipping the two-column iMPS into an isoTNO  $A$  and a new iMPS  $|\Phi\rangle$ .

Formulating this as an optimization problem, one finds the optimal isometric tensor  $a'$  and normalized MPS tensor  $\phi'$  by maximizing the overlap density between  $|\Psi\rangle$  and  $A|\Phi\rangle$ :

$$a', \phi' = \arg \max_{a, \phi \in \text{isometry}} \text{Re } \lambda_1(T_{\psi:a\phi}) \quad (4.5)$$

where  $\text{Re } \lambda_1$  denotes the real part of the largest eigenvalue of the mixed transfer matrix  $T_{\psi:a\phi}$  between  $|\Psi\rangle$  and  $A|\Phi\rangle$ :

$$T \equiv T_{\psi:a\phi} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} \psi \\ | \\ a \\ | \\ \phi \end{array} \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} \quad (4.6)$$

As any iMPS can be written exactly in isometric form with the same bond dimension, we directly search for isometric  $\phi$  tensors. In the following, when no confusion arises, we simply use  $T$  to denote this mixed transfer matrix. Note that maximizing  $\text{Re } \lambda_1(T)$  or  $|\lambda_1(T)|$  is equivalent for our purpose because of the unitary freedom of the isometries.

We now propose and evaluate four algorithms for solving the splitting problem posed in Eq. (4.4): **iMM-Local**, **iMM-Polar**, **iMM-MPO**, and **iMM-CG**. We first give a brief overview of these methods. **iMM-Local** is the direct generalization of the finite MM, in which the local tripartite decomposition in Eq. (4.3) is iterated until convergence. The method is not guaranteed to converge, but we find it quickly provides an approximate solution. Thus, we use it as an initialization for the following optimization methods. The **iMM-Polar** method minimizes the error of the fixed-point tripartite decomposition using repeated polar decompositions. It is slower than **iMM-Local** and does not directly maximize the overlap density  $\text{Re } \lambda_1$ . However, we find it to be nearly optimal in practice. Finally, **iMM-MPO** and **iMM-CG** maximize the overlap density  $\text{Re } \lambda_1$  through two different methods. The **iMM-MPO** method finds the  $\phi$  tensor by the variational MPO-MPS compression algorithm and the  $a$  tensor by the polar decomposition over the linearized overlap. The method is slightly slower than **iMM-Polar** and yields comparable results. The **iMM-CG** method is based on the conjugate gradient ascent of the overlap density  $\text{Re } \lambda_1$ . This method has difficulty reaching a satisfying (local) minimum on its own and is best used to improve the results of other methods. It is found that the most efficient strategy is to use **iMM-Local** to obtain a stable initial guess and then use **iMM-Polar** to finish the infinite splitting optimization. Using **iMM-MPO** or **iMM-CG** in the end is optional, as the improvements they provide after **iMM-Polar** are very few. We include details of the four methods for completeness but do not use **iMM-MPO** and **iMM-CG** beyond benchmarking.

#### 4.2.1. iMM-Local

Given the input iMPS  $|\Psi\rangle$  made of tensors  $\psi$  as shown on the left in Eq. (4.4), one performs the finite MM in Eq. (4.2) assuming that all the blue tensors above the OC equal  $\psi$ . The first, normalized  $s_1$  is randomly generated such that the dimensions of the lower two legs match that of the desired  $a$  and  $\phi$  tensors;  $s_1$  plays the role of the initial zero-site wavefunction. One then uses the tripartite decomposition in Eq. (4.3) to solve the following problem for site  $n$  iteratively along the infinite direction:

$$\begin{array}{c} \downarrow \\ \psi \\ \leftarrow \rightarrow \\ \downarrow \\ s_n \\ \leftarrow \rightarrow \end{array} \approx \begin{array}{c} \downarrow l \\ s_{n+1} \\ \leftarrow \rightarrow \\ \downarrow \\ a_n \quad \phi_n \\ \leftarrow \rightarrow \\ \downarrow \\ j \end{array} \quad (4.7)$$



This iteration terminates if  $\|s_n - s_{n+1}\|$  is smaller than a certain threshold or  $n$  exceeds a certain iteration limit. One then takes  $A$  and  $|\Phi\rangle$  to be composed, respectively, of the last  $a_n$  and  $\phi_n$  in the iteration.

For faster runtime and better convergence, we found it important to minimize Rényi-2 entropy in the tripartite splitting here. We suggest the interested reader check out the details in previous works [1, 75] and describe the additional modification in the following. The tripartite splitting has unitary gauge redundancies on the internal bonds  $i, j$ , and  $k$ . To increase the chance that the MM iterations converge along the infinite direction, we need to fix the gauge redundancies on bond  $k$  and  $i$ . We do not need to fix the redundancy on bond  $j$ , because only  $S$  is used for the convergence criteria. To fix  $k$  and  $i$ , we do an singular value decomposition (SVD) on matrix  $S_{k:li} = UsV^\dagger$ , which is formed by grouping bond  $i$  and  $l$  as the column index, and replace  $S$  with  $sV^\dagger$ . We then do an SVD on matrix  $S_{lk:i} = UsV^\dagger$ , formed by grouping bond  $k$  and  $l$ , and replace  $S$  with  $Us$ . The two SVDs almost fix the gauge redundancy except for the gauge freedom in the SVD itself: every column of  $U$  from an SVD can be multiplied by a phase, as long as the corresponding row of the  $V^\dagger$  is multiplied by the inverse of that phase. To fix this phase freedom, we view  $S_{l=0}$  as a matrix and demand its first column and first row to be all positive numbers. This can be achieved by a unitary diagonal matrix on bond  $i$  and  $k$  independently. When the singular values of  $S_{k:li}$  and  $S_{lk:i}$  are not degenerate, these operations fix the gauge freedom on bond  $i$  and  $k$  completely. Despite the gauge-fixing, `iMM-Local` still often fails to converge. Thus, it is best used to provide an initial guess for the `iMM-Polar` algorithm which is guaranteed to converge along the infinite direction.

#### 4.2.2. iMM-Polar

To overcome the convergence issue of `iMM-Local`, we consider `iMM-Polar` to directly optimize the approximate fixed point equation of the tripartite decomposition in Eq. (4.7). In other words, we enforce translational invariance of the iMM by requiring  $s_n = s_{n+1} = s$  in Eq. (4.7) and optimize over  $s$ ,  $a$ , and  $\phi$  to make the equality close to exact. More precisely, we maximize the real part of the overlap between the left and right-hand side of the Eq. (4.7) under the constraints that  $\|s\| = 1$ ,  $a$  is an isometry, and  $\phi$  is a normalized MPS tensor in isometric form <sup>4</sup>:

$$F = \text{Re} \left[ \text{Tr} \left( \begin{array}{c} \psi \\ \bar{s} \end{array} \begin{array}{c} s \\ a \end{array} \begin{array}{c} \phi \end{array} \right) \right] = \text{Re} s^\dagger T s. \quad (4.8)$$

We note here that the fitness function above is not the same as the one in Eq. (4.5), in which the  $s$  tensor plays no role. `iMM-Polar` is thus not variationally optimal. This way of solving Eq. (4.5) approximately is entirely motivated by the finite MM, and as shown later, is very close to being variationally optimal.

Here we describe the steps of `iMM-Polar` which consists of maximizing Eq. (4.8) alternately:

1. When  $s$  and  $a$  are fixed, one forms the environment  $E_\phi$  of  $\phi$  in  $F$  such that  $F = \text{Tr}(E_\phi^\dagger \phi)$ . Here  $\phi$  is viewed as a matrix with its incoming and outgoing indices grouped as the row and column index respectively.  $E_\phi$  is grouped as a matrix such that its row

<sup>4</sup>The reason to take the real part in Eq. (4.8) is that for any normalized states  $|x\rangle$  and  $|y\rangle$ ,  $\| |x\rangle - |y\rangle \|^2 = 2(1 - \text{Re}\langle x|y\rangle)$ .

and column index contract with the row and column index of  $\phi$  respectively. Then the optimal isometry  $\phi'$  maximizing  $F$  is given by the polar decomposition of  $E_\phi$ :  $\phi' = \arg \max_\phi F = U_\phi$  where  $U_\phi P_\phi = E_\phi$  is the polar decomposition of  $E_\phi$  [158].

2. When  $s$  and  $\phi$  are fixed, one analogously find the optimal update  $a' = \arg \max_a F$  through the polar decomposition of  $E_a$ .
3. When  $a$  and  $\phi$  are fixed,  $F = \text{Re } s^\dagger T s = s^\dagger ((T + T^\dagger)/2) s$ , and optimal  $s' = \arg \max_s F$  is given by the leading eigenvector of the Hermitian matrix  $(T + T^\dagger)/2$ .

We repeat steps 1 to 3 to update  $\phi, a, s$  many times until convergence is reached within a threshold. Note that  $F$  is strictly increasing at each step. Like DMRG, this style of alternate optimization is not convex, and a good initial guess, given by `iMM-Local`, can speed up the optimization greatly. Typically such an optimization results in local extremum.

### 4.2.3. iMM-MPO

The `iMM-MPO` method maximizes the overlap density  $\text{Re } \lambda_1$  through alternatively updating  $a$  and  $\phi$  tensor until convergence. We describe the two alternative steps of `iMM-MPO` as follows:

1. Given  $a$ , which determines a matrix product operator (MPO)  $A^\dagger$  acting on the state  $|\Psi\rangle$ , the optimal update of  $\phi$  is determined by the variational MPO-MPS compression algorithm developed for uniform (infinite) MPS [204]. The update for  $\phi$  is optimal since the compression algorithm maximizes the overlap density. We perform only one update step in the MPO-MPS compression algorithm [204] instead of finding the converged solution.
2. Given  $\phi$ , we linearize the overlap  $\text{Re } \langle \Psi | A \Phi \rangle$ , i.e., viewing each  $a$  tensor in  $|A\rangle$  as an independent tensor, and find the update for  $a$  by the polar decomposition over the environment of  $a$  tensor,

$$E_a = \text{Tr} \left[ \begin{array}{c} \phi \\ \phi \\ L \\ R \end{array} \right] \quad (4.9)$$

where  $L^T$  (row vector) and  $R$  (column vector) are the left and right leading eigenvectors of mixed transfer matrix  $T$ .

We repeat steps 1 to 2 to update  $\phi, a$  until convergence is reached within a threshold or the overlap density  $\text{Re } \lambda_1$  starts increasing in the update for the  $a$  tensor. Note that the update in step 2 usually, but is not guaranteed to, increase the overlap density. In fact, it is related to a gradient ascent update [174, 205] on tensor  $a$ . We observe this update is efficient in increasing the overlap at the initial stage but is slower in the final stage of the convergence compared to a non-linear conjugate gradient update. Overall, this method tends to give slightly more accurate results at the cost of slightly longer runtimes compared to `iMM-Polar`.

#### 4.2.4. iMM-CG

An alternative way to maximize the overlap density  $\text{Re } \lambda_1$  is to perform non-linear conjugate gradient ascent on isometries  $a$  and  $\phi$ . To respect the isometric constraint on  $A$  and  $\Phi$ , we parametrize the tensors as  $a = U_a a_0 = \exp(X_a) a_0$  and  $\phi = U_\phi \phi_0 = \exp(X_\phi) \phi_0$ , where  $a_0$  and  $\phi_0$  are any fixed isometries,  $U_a$  and  $U_\phi$  are unitary matrices acting on the incoming legs of  $a$  and  $\phi$ , respectively, and  $X_a$  and  $X_\phi$  are anti-Hermitian matrices. Assuming all bonds to be dimension  $\chi$ ,  $X_a$  is a  $\chi^2 \times \chi^2$  matrix, while  $X_\phi$  is  $\chi^3 \times \chi^3$ . The variational space is now the vector space of the anti-Hermitian matrices  $X_a$  and  $X_\phi$ , and conjugate gradient ascent can readily be applied.

Denote the fitness function as

$$O[X_a, X_\phi] \equiv \text{Re } \lambda_1(T) \equiv \text{Re } \lambda_1(T_{\psi:a[X_a]\phi[X_\phi]}). \quad (4.10)$$

The change in the objective due to  $dX_a$  can be computed as

$$\begin{aligned} dO &= \text{Re } L^T \cdot dT \cdot R = \text{Re } \text{tTr}(E_{X_a}, dX_a) \\ &= \text{tTr}(\text{Re}E_{X_a}, d\text{Re}X_a) - \text{tTr}(\text{Im}E_{X_a}, d\text{Im}X_a) \end{aligned} \quad (4.11)$$

where  $L^T$  (row vector) and  $R$  (column vector) are the left and right leading eigenvectors of  $T$ . We assume they are normalized so that  $L^T R = 1$ . Above,  $\text{tTr}$  denotes tensor contraction, and  $E_{X_a}$  is the environment of  $X_a$  in the tensor contraction:

$$E_{X_a} = \quad (4.12)$$

Note the implicit dependency of  $L$  and  $R$  on  $X_a$  in the current form give zero contribution to the change in the objective [206].

The ascent direction for the maximization is thus given by the derivative:

$$\frac{\partial O}{\partial \text{Re}X_a} = \text{Re}E_{X_a}, \quad \frac{\partial O}{\partial \text{Im}X_a} = -\text{Im}E_{X_a} \quad (4.13)$$

Thus, the ascent direction of  $X_a$  is  $\overline{E_{X_a}}$ <sup>5</sup>. In fact,  $dO$  is manifestly positive if  $dX_a = \overline{E_{X_a}}$ , where the overline denotes complex conjugation. Note that the  $E_{X_a}$  computed in Eq. (4.12) and the ascent direction  $\overline{E_{X_a}}$  are generally not anti-Hermitian. Therefore, one needs to anti-Hermitian-ize  $dX_a$  so that the updated  $X_a$  is still anti-Hermitian. One can analogously compute the ascent direction for  $X_\phi$ .

With these ingredients, the conjugate gradient ascent is done as follows.

1. At CG step  $k$ , compute  $E_{X_a}$  and  $E_{X_\phi}$  and anti-Hermitian-ize them. We overload notation and use  $E_X$  to refer to the anti-Hermitian environments below.
2. Set the ascent direction,  $H$ , using the gradient and the ascent direction from the previous step:

$$\begin{aligned} H_a(k) &= \overline{E_{X_a}} - \beta H_a(k-1) \\ H_\phi(k) &= \overline{E_{X_\phi}} - \beta H_\phi(k-1) \end{aligned} \quad (4.14)$$

where  $\beta$  is determined by a non-linear CG  $\beta$ -mixer, e.g. Polak-Ribière.

<sup>5</sup>Here we treat the real and imaginary part of  $X_a$  as independent variables. An alternative way to arrive at the same result is to treat  $X_a$  as complex-valued variables and the ascent direction is given by  $dO/d\overline{X_a} = \frac{1}{2} R^\dagger (dT^\dagger/d\overline{X_a}) L = \frac{1}{2} \overline{E_{X_a}}$ .

3. Parametrize  $a(t)$  and  $\phi(t)$  along the ascent direction:  $a(t) = \exp(tH_a)a$  and  $\phi(t) = \exp(tH_\phi)\phi$ . Via the linesearch algorithm, look for  $t_{\max}$  at which  $O(t_{\max})$  is maximized along the  $t$ -curve. This needs the computation of  $dO(t)/dt$ :

$$\begin{aligned} \frac{dO(t)}{dt} = & \operatorname{Re} \operatorname{tTr}(E_{X_{a(t)}}, H_a) \\ & + \operatorname{Re} \operatorname{tTr}(E_{X_{\phi(t)}}, H_\phi) \end{aligned} \quad (4.15)$$

4. Update  $a \rightarrow a(t_{\max})$  and  $\phi \rightarrow \phi(t_{\max})$  and  $k \rightarrow k + 1$ .

This process is iterated until  $a$  and  $\phi$  converge to within the desired threshold.

#### 4.2.5. Error measures

Before presenting benchmarks and applications of the iMM, we first discuss error measures for the splitting problem in Eq. (4.4). As before, let  $|\Psi\rangle$  be the input to the iMM, and  $A$  and  $|\Phi\rangle$  be the output.  $|\Psi\rangle$  and  $|\Phi\rangle$  are always normalized. We consider the error of the fidelity density,

$$\varepsilon \equiv 1 - (\operatorname{Re} \lambda_1(T))^2 = 1 - \lambda_1(T)^2, \quad (4.16)$$

where we assume the iMM algorithm finds  $A$  and  $|\Phi\rangle$  such that the dominant eigenvalue of  $T$  is real<sup>6</sup>.

It can be shown that  $\varepsilon$  is a sum of two errors (see Appendix B.1 for the derivation):

$$\varepsilon = \varepsilon_p + \varepsilon_t + O(\varepsilon_p^2, \varepsilon_t^2, \varepsilon_p \varepsilon_t) \quad (4.18)$$

where

$$\begin{aligned} \varepsilon_p &\equiv 1 - \lambda_1(T_{A^\dagger \Psi: A^\dagger \Psi}), \\ \varepsilon_t &\equiv 1 - \left( \lambda_1(\widetilde{T_{A^\dagger \Psi: \Phi}}) \right)^2. \end{aligned} \quad (4.19)$$

Here  $\widetilde{A^\dagger |\Psi\rangle} = A^\dagger |\Psi\rangle / \|A^\dagger |\Psi\rangle\|$  is normalized.  $\varepsilon_p$  measures the norm that  $A^\dagger |\Psi\rangle$  loses due to the projection and  $\varepsilon_t$  measures the truncation error due to approximating  $\widetilde{A^\dagger |\Psi\rangle}$  with the MPS  $|\Phi\rangle$ .

In practice, these errors guide the choice of internal bond dimensions of the iMM. Let us denote the bond dimensions of the iMM as the following:

$$\begin{array}{ccc} \begin{array}{c} \chi_0 \\ \downarrow \\ \chi_\ell \rightarrow \psi \leftarrow \chi_r \\ \downarrow \\ \chi_0 \\ |\Psi\rangle \end{array} & \approx & \begin{array}{cc} \begin{array}{c} \chi_v \\ \uparrow \\ \chi_\ell \rightarrow a \leftarrow \chi_h \\ \uparrow \\ \chi_v \\ A \end{array} & \begin{array}{c} \eta \\ \uparrow \\ \phi \leftarrow \chi_r \\ \uparrow \\ \eta \\ |\Phi\rangle \end{array} \end{array} \quad (4.20)$$

Assuming the non-convex optimization in the iMM is successful, increasing  $\eta$  decreases  $\varepsilon_t$ , and increasing  $\chi_v$  and  $\chi_h$  decreases  $\varepsilon_p$ . Perhaps less obvious is that increasing  $\chi_v$

<sup>6</sup>To motivate this definition, let us consider the example of splitting a finite and uniform system of size  $L$ , as in Eq. (4.2). The error of the splitting is given by

$$\begin{aligned} \|\Psi\rangle - A|\Phi\rangle\|^2 &= 2 - 2\operatorname{Re}\langle\Psi|A|\Phi\rangle \\ &\approx 2 - 2(\sqrt{1-\varepsilon})^L \approx \varepsilon L. \end{aligned} \quad (4.17)$$

Thus,  $\varepsilon$  is the intensive error density.

and  $\chi_h$  can also decrease  $\varepsilon_t$ , because  $A$ , in addition to being a projector, also serves as a disentangler of  $\Psi$ <sup>7</sup>. That is, if  $A$  is well-chosen,  $\widetilde{A^\dagger|\Psi\rangle}$  will have less entanglement entropy than  $\Psi$ , and, when truncated to an MPS with smaller bond dimension, will have less truncation error compared to directly truncating  $|\Psi\rangle$ . Implicitly in iMM, when minimizing  $\varepsilon$ , the optimization reaches a balance between the projecting ( $\varepsilon_p$ ) and the disentangling ( $\varepsilon_t$ ) role of  $A$  so that their collective effect minimizes  $\varepsilon$ . The disentangling effect of  $A$  will be reflected in  $|\Phi\rangle$  having less entanglement entropy than  $|\Psi\rangle$ .

The above discussion would appear to indicate that, as long as the computational cost is affordable, the internal bond dimensions should be as large as possible to reduce  $\varepsilon$ . In regular iMPS compression algorithms, one can set the bond dimension of the trial state to be as large as desired, and the only side-effect is that the code will run for longer. This is indeed the case for  $\eta$ , as it is just the bond dimension of the new MPS  $|\Phi\rangle$ . For  $\chi_h$ , we have to choose  $\chi_h \leq \chi_\ell$  in order for  $A$  to be an isometry (see Eq. (4.20)). However, special care must be taken in choosing the bond dimension  $\chi_v$  to ensure the injectivity of the state. One way to ensure this is by choosing  $\chi_\ell > \chi_h$ <sup>8</sup> – this motivates the choice of the boundary bond dimension  $D_b$  to be greater than 1.

### 4.3. iMM benchmarks

Having introduced algorithms for performing the iMM splitting procedure depicted in Eq. (4.4) and decomposed the resulting error as the sum of truncation and projection error terms, we now perform several experiments. The first is to compare the four different iMM algorithms introduced in Sec. 4.2 and various combinations of these algorithms. From this, we conclude that the combination of `iMM-Local` and `iMM-Polar` produces accurate results without the significant computational cost, and we use this combination for all future experiments. Then, we repeatedly apply iMM to a strip isoTNS, sweeping back and forth, and find that the accumulated error saturates after about 20 iterations.

In this section and those that follow, we consider the 2D transverse field Ising (TFI) model,

$$H = - \sum_{\langle i,j \rangle} Z_i Z_j - g \sum_i X_i, \quad (4.21)$$

to benchmark the performance of the iMM algorithms and the iTEBD<sup>2</sup> algorithm. Unless otherwise noted, we choose  $g = 3.5$  to be in the paramagnetic phase. The critical coupling for this model, obtained via cluster Monte Carlo simulations, in the thermodynamic limit in both directions is  $g_C^{2D} \approx 3.04438$  [185], while the critical coupling with infinite density matrix renormalization group (iDMRG) calculation on infinite cylinders increases from  $g_C^{1D} = 1$  towards  $g_C^{2D}$  with width [207].

#### 4.3.1. Benchmark for a single run of iMM

We begin by investigating the performance of the iMM algorithms introduced earlier. The input state is the ground state of the 2D TFI model on an infinite strip of width  $L_x = 4$  obtained from iDMRG with bond dimension  $\chi = 128$ . The input tensor for the algorithm is obtained by contracting the four iMPS tensors in the unit cell (one row) to one iMPS tensor with 4 physical legs, each of dimension  $d = 2$ . We then group the two physical legs corresponding to the left two sites in the row into the left leg and the remaining two legs

<sup>7</sup>The hint is given in the tripartite decomposition, where tensors in column  $A$  act as a disentangler to the physical wavefunction  $\Psi$ .

<sup>8</sup>This is due to the structure theorem derived in [2] and the counting of the bond dimensions.

into the right leg of the two-sided iMPS. The splitting problem is defined diagrammatically below.

$$\begin{array}{ccc}
 \begin{array}{c} 128 \\ \downarrow \\ \psi \\ \downarrow \\ 128 \\ |\Psi\rangle \end{array} & \approx & \begin{array}{ccc} & \begin{array}{c} 2 \\ \uparrow \\ a \\ \uparrow \\ 2 \\ A \end{array} & \begin{array}{c} 10 \\ \uparrow \\ \phi \\ \uparrow \\ 10 \\ |\Phi\rangle \end{array} \\
 4 \rightarrow & & 4 \rightarrow & & 4 \leftarrow \\
 \end{array} \quad (4.22)
 \end{array}$$

This splitting problem involves both projection and truncation, as  $\chi_h < \chi_\ell$ .

We compare all four iMM algorithms discussed above and various combinations of them. We report errors and the runtimes on a standard workstation, averaged over five runs, in Table 4.1. In this discussion, we use the shorthand notation **L** to represent **iMM-Local**, **P** for **iMM-Polar**, **M** for **iMM-MPO**, and **C** for **iMM-CG** for convenience.

iMM method	$\varepsilon_p$	$\varepsilon_t$	time (s)
L	$1.8\text{e-}03 \pm 1.0\text{e-}06$	$3.4\text{e-}05 \pm 8.2\text{e-}06$	$0.7 \pm 0.01$
P	$1.3\text{e-}03 \pm 1.4\text{e-}03$	$2.3\text{e-}05 \pm 4.4\text{e-}06$	$38 \pm 15$
M	$1.0\text{e-}02 \pm 6.7\text{e-}03$	$1.4\text{e-}03 \pm 1.7\text{e-}03$	$12 \pm 21$
C	$2.2\text{e-}01 \pm 2.6\text{e-}01$	$3.7\text{e-}01 \pm 3.6\text{e-}01$	$21 \pm 10$
LP	$6.2\text{e-}04 \pm 1.9\text{e-}08$	$2.6\text{e-}05 \pm 1.8\text{e-}08$	$11 \pm 0.1$
LM	$5.0\text{e-}04 \pm 2.0\text{e-}10$	$9.4\text{e-}06 \pm 5.7\text{e-}10$	$62 \pm 0.6$
LC	$5.0\text{e-}04 \pm 1.2\text{e-}07$	$4.2\text{e-}05 \pm 1.1\text{e-}06$	$29 \pm 1.9$
LPM	$4.6\text{e-}04 \pm 2.0\text{e-}08$	$5.8\text{e-}07 \pm 8.8\text{e-}11$	$12 \pm 0.05$
LPC	$4.6\text{e-}04 \pm 4.2\text{e-}06$	$5.2\text{e-}06 \pm 2.6\text{e-}07$	$19 \pm 1.4$
LMC	$5.0\text{e-}04 \pm 2.1\text{e-}11$	$9.3\text{e-}06 \pm 2.1\text{e-}11$	$72 \pm 0.4$
LPMC	$4.5\text{e-}04 \pm 3.7\text{e-}10$	$2.4\text{e-}07 \pm 1.0\text{e-}10$	$20 \pm 0.07$

Table 4.1.: Projection and truncation errors as defined in Eq. (4.19) in Sec. 4.3.1 and runtimes for different iMM methods applied to the splitting problem in Eq. (4.22). Results from five different runs are average to give the standard deviations. We conclude that the combination of **iMM-Local** and **iMM-Polar** provides an accurate result without incurring a large computational cost.

We find that **L** is indeed the fastest but does not achieve the accuracy of other standalone methods. However, even though this method is not guaranteed to converge, the gauge fixing procedure reduces variance, indicating that this method can be used to provide a stable starting point for other methods. Seeding methods **P** and **M** with the  $a$  and  $\phi$  tensors provided by **L**, yielding methods **LP** and **LM**, decreases the error and standard deviation, while also for **LP** significantly reducing the runtime, compared to **P** and **M** alone. Further improving the results with **C** is possible, but the improvements are not significant yet have added computational cost. Thus for all future experiments, we use **LP**.

Additionally, empirically we find that when  $\chi_v$  is large, it is best to use **L** to produce starting solutions with a smaller vertical bond dimension, say  $\chi'_v = 4$ , and then isometrically expand the vertical dimensions of the  $a$  tensor gradually. At each intermediate bond dimension between  $\chi'_v$  and  $\chi_v$ , we use **P** to improve the result. We then expand the isometric tensor  $a$  by viewing it as a matrix by grouping the incoming legs into a row index and the outgoing legs into a column index. The incoming row index can be increased by zero padding, while the outgoing leg must be increased by adding orthogonal columns. Such a gradual iMM procedure improves the stability and performance of the splitting as  $\chi_v$  grows.

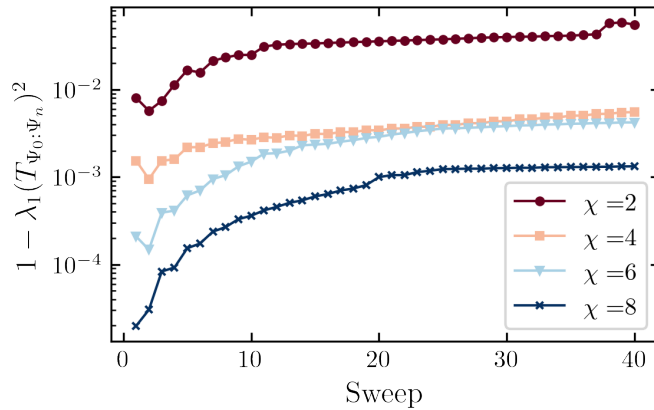


Figure 4.1.: Error between original isoTNS  $|\Psi^{(0)}\rangle$  and isoTNS  $|\Psi^{(n)}\rangle$  produced by repeated  $n$  iMM sweeps.  $|\Psi^{(0)}\rangle$  is an isoTNS with  $L_x = 6$ ,  $\chi_{\text{init}} = 2$ ,  $D_b = 1$  describing the ground state of the 2D TFI model.

### 4.3.2. Repeated application of iMM to an isoTNS

The iMM algorithm performs the splitting in Eq. (4.4) approximately, and so, unlike moving the OC by QR decomposition in 1D MPS, iMM will not exactly preserve the state. Thus it is important to quantify how repeated applications of iMM affect an isoTNS and perturb it away from the starting state. Here we consider specifically the case where the starting state is an approximate ground state of the 2D TFI model on an  $L_x = 6$  strip represented by an isoTNS. We measure the deviation of the state after consecutive iMM from the original state.

Suppose that the current (after  $n - 1$  iterations) isoTNS  $|\Psi^{(n-1)}\rangle$  has its orthogonality column  $|\Psi_1^{(n-1)}\rangle$  as the leftmost column of the strip; we then write the tensor network as  $|\Psi^{(n-1)}\rangle = |\Psi_1^{(n-1)}\rangle B_2^{(n-1)} \dots B_{L_x}^{(n-1)}$ , where  $B$  denotes an isoTNO with horizontal isometry arrows pointing left. We fuse the columns  $|\Psi_1^{(n-1)}\rangle$  and  $B_2^{(n-1)}$  to form a doubled column  $|\Psi_{1,2}^{(n)}\rangle$  with two physical sites per tensor; we split this doubled orthogonality column using iMM into the isoTNO  $A_1^{(n)}$  with rightward pointing isometry arrows and the new orthogonality column  $|\Psi_2^{(n)}\rangle$ , with each column having a physical index. We perform this procedure of merging and splitting two columns a total of  $L_x - 1$  times, moving the orthogonality column entirely to the right of the strip. This completes one sweep and produces a new isoTNS  $|\Psi^{(n)}\rangle = A_1^{(n)} \dots A_{L_x-1}^{(n)} |\Psi_{L_x}^{(n)}\rangle$ . We can now repeat the process moving to the left (or in practice horizontally mirroring the isoTNS and again moving to the right) to produce  $|\Psi^{(n+1)}\rangle$ . We measure the error of fidelity density as in Eq. (4.16), where the transfer matrix  $T$  is formed by one row of the original state  $|\Psi^{(0)}\rangle$  and the state at  $n$ -iteration  $|\Psi^{(n)}\rangle$ .

The results of this procedure for an  $L_x = 6$ ,  $\chi_{\text{init}} = 2$  isoTNS is shown in Fig. 4.1. This isoTNS represents the ground state of the 2D TFI model. We choose bond dimensions in the iMM splitting to be  $\chi = \chi_v = \chi_h$  and  $\eta = 24$ . After each sweep, the maximum bond dimension in  $|\Psi_n\rangle$  is  $\chi$ . Some bonds will have values smaller than  $\chi$  so that the isometric conditions on each tensor are satisfied. We find that for each of the bond dimensions used in the iMM and thus the bond dimensions of the resulting isoTNS, the accumulated error saturates after 20 sweeps. Additionally, we find that increasing  $\chi$  decreases the error as expected, as the iMMs in the sweep can be done more accurately and thus have a less corrupting effect on the state.



## 4.4. Evaluation of local observables: Energy as an example

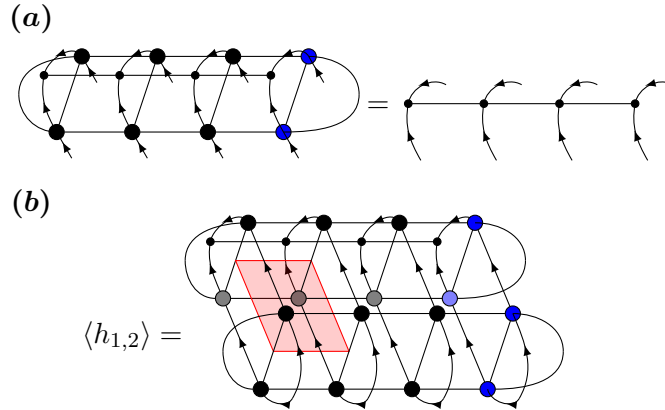


Figure 4.2.: bMPO energy evaluation. **(a)** boundary MPO representing the fixed point of the isoTNS transfer matrix.  $L_x \times \infty$  strip is oriented so that the infinite direction is vertical. Horizontal arrows are omitted for clarity. The fixed point only needs to be found against the direction of vertical arrows, as the fixed point in the direction of arrows is the identity. **(b)** Evaluation of  $\langle h_{1,2} \rangle$  using bMPO fixed point and trivial fixed point. The energy of a row is found by  $\sum_c \langle h_{c,c+1} \rangle$ .

The evaluation of expectation values of local observables is crucial and non-trivial for 2D finite and infinite TNS. As the second application of the iMM algorithm, we show that, up to some small error density, we can efficiently evaluate the expectation of local observables of the given strip isoTNS using iMM. We benchmark the result by comparing it to results obtained by approximate PEPS contractions using boundary MPO (bMPO) approach [124, 149] and the exact contraction by collapsing the state into iMPS. Here, we consider the evaluation of energy as an example, but the method can be applied to the evaluation of other local observables.

### 4.4.1. Energy evaluated by iMM

Suppose we wish to evaluate the energy of a state given a Hamiltonian  $H = \sum_c H_{c,c+1}$  where  $H_{c,c+1}$  acting on columns  $c$  and  $c+1$  is composed of a translationally invariant local operator  $h_{c,c+1}$ . The expectation value of  $H_{c,c+1}$  can be found by evaluating the energy of each local operator  $h_{c,c+1}$  with an isoTNS with doubled column  $c, c+1$  as the orthogonality column. This is because calculating expectation values of operators contained entirely in the orthogonality column reduces to an efficient 1D iMPS problem. If we could move the orthogonality column freely around the strip in an exact fashion, this would give us an exact method for evaluating the energy. However, as moving the orthogonality column requires the iMM, this method inherently incurs an approximation error on the order of the iMM errors. As demonstrated in Sec. 4.3.2, the state represented by an isoTNS is not significantly affected by iMM applications, where the individual errors of each iMM application were presented in Table 4.1. Thus we can perform a full sweep of iMM iterations over the entire strip and use the two-site orthogonality center of each doubled column to evaluate the expectation value of  $h_{c,c+1}$ . Doing this for each two-column term gives us the energy of one row,  $\langle E_{\text{row}} \rangle = \sum_c \langle h_{c,c+1} \rangle$ , which can be converted to per-site energy by dividing by strip width.

Note that the maximum bond dimension  $\chi' = \chi'_h = \chi'_v$ , and  $\eta'$  of the iMM used to calculate observables do not need to be the same as  $\chi$  and  $\eta$ , that of the original isoTNS.



The accuracy of calculated observables increases as we increase  $\chi'$ , as this improves the accuracy of iMM. This method for calculating observables scales linearly with the strip width and requires  $L_x - 1$  applications of iMM, which as stated previously has a computational cost of  $\mathcal{O}(\chi'^4 \eta'^3)$ .

#### 4.4.2. Energy evaluated by boundary MPO

To check the energy evaluation from the iMM independently, we compute the energy of an isoTNS strip without using the iMM. To do this, we consider the isoTNS row transfer matrix and find its fixed point. We can compute local observables efficiently given the fixed points of the row transfer matrix. However, as the dimension of the transfer matrix grows exponentially in the width  $L_x$ , we approximate the fixed point “vector” by the boundary MPO (bMPO) [60] as depicted in Fig. 4.2(a). Note that the strip is oriented such that the infinite direction is vertical. Combining the bMPO contraction methods developed for finite TNS [60, 64, 149] and the power method, we find the dominant eigenvector of the transfer matrix represented by a bMPO with bond dimension  $D_{\text{bMPO}}$ .

We only have to converge the fixed point *against* the direction of the vertical isometric arrows (from the top down in Fig. 4.2(b)). This is because the fixed point along the direction of the arrows is, by definition, an identity operator over each column. The non-trivial fixed point *against* the isometric arrow direction admits a spectral decomposition  $U\rho U^\dagger$ , where  $\rho$  is diagonal and positive definite, encoding the square of the Schmidt values. However, we do not utilize this property but use a bMPO directly to parameterize the fixed point vector. With this bMPO representing the fixed point of the isoTNS row transfer matrix, we calculate the energy of a row by evaluating  $\langle \sum_c h_{c,c+1} \rangle = \sum_c \langle h_{c,c+1} \rangle$ . This is done by contracting the network shown in Fig. 4.2(b) for each two-column local operator  $h_{c,c+1}$ , shown here to act on two neighboring rows.

This method can be made arbitrarily accurate by increasing the bond dimension  $D_{\text{bMPO}}$ , but we note that this method is very costly as strip width  $L_x$  and the bond dimension  $\chi$  of the isoTNS grows. Calculating the fixed point scales as  $\mathcal{O}(N_{\text{iter}} L_x (\chi^4 D_{\text{bMPO}}^3 + d \chi^6 D_{\text{bMPO}}^2))$ , where  $d$  is the local Hilbert space dimension and  $\chi$  is the dimension of all virtual legs in the isoTNS. Typically  $D_{\text{bMPO}} \sim \chi^2$ , so this method scales as  $\mathcal{O}(\chi^{10})$ . The  $N_{\text{iter}}$  is the number of the transfer matrix-vector multiplications required for convergence which is related to the gap in the transfer matrix.

#### 4.4.3. Energy benchmarks

The exact but most computationally intensive  $\mathcal{O}(\exp(L_x))$  method is to find the exact fixed point and perform an exact contraction. To this end, we collapse each row of an isoTNS to form an iMPS with physical dimension  $d^{L_x}$ , representing  $L_x$  physical sites per tensor. Then, standard MPS methods can be used to find the exact fixed point and evaluate the energy. We use this essentially exact method only to benchmark the previous two methods.

The benchmark result is shown in Fig. 4.3. We apply both methods to a  $L_x = 8$ ,  $\chi = 4$  isoTNS representing 2D TFI ground state. We find that the bMPO method is essentially exact for large enough  $D_{\text{bMPO}}$ , while the accuracy of the iMM energy increases with both  $\chi'$  and  $\eta'$ , as the splitting can be done more accurately with the larger space of available tensors. From this, we note that the iMM energy tends to underestimate the true energy; yet for large strip widths and large  $\chi$ , other methods are infeasible due to computational costs.

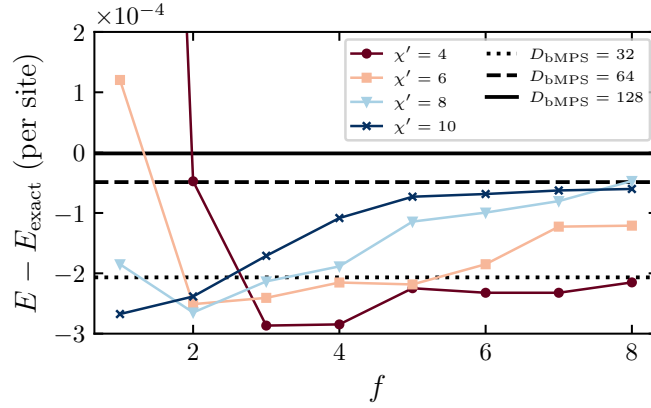


Figure 4.3.: The energy of the  $L_x = 8$ ,  $\chi = 4$  isoTNS produced by peeling evaluated with the iMM and bMPO methods. We compare the energies to the exact energy evaluated by exact iMPS contraction. The energy from the bMPO method is close to exact for large  $D_{\text{bMPO}}$  but is very costly as  $D_{\text{bMPO}}$  grows due to the complexity of finding the fixed point. The accuracy of iMM energy increases with both  $\eta'$  and  $\chi'$  used in iMM. In the figure, we use  $f \equiv \eta'/\chi'$ .

## 4.5. iTEBD<sup>2</sup>

We now introduce the TEBD-based time evolution algorithm for infinite strip isoTNS, dubbed iTEBD<sup>2</sup>. We then demonstrate the algorithm by performing imaginary time evolution to find the ground states of the two-dimensional transverse field Ising model.

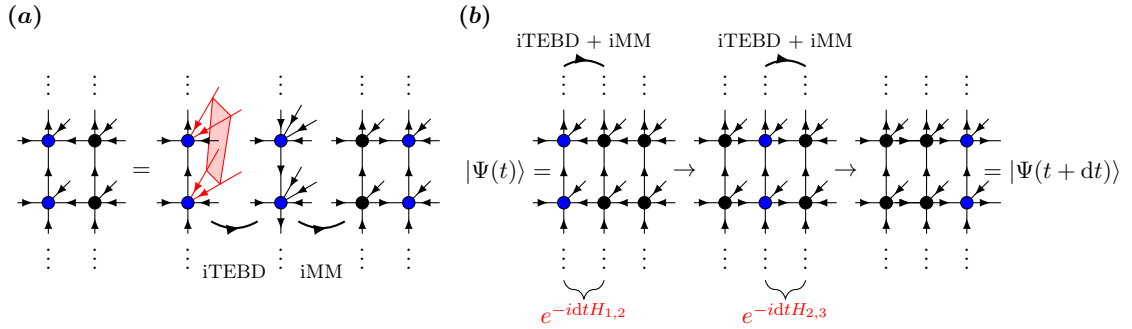


Figure 4.4.: iTEBD<sup>2</sup> Algorithm. **(a)** Subroutine acting on two columns, the left of which is the orthogonality column. First, the two columns are fused. 1D iTEBD is applied to the physical legs (colored red) of the double column using gates  $e^{-idtH_{c,c+1}}$ , acting on a plaquette of four physical sites; note that the gate does not act on any virtual legs. iMM is then applied to split the doubled column and move the orthogonality center to the right. **(b)** The iTEBD<sup>2</sup> algorithm involves  $L_x - 1$  iterations of the subroutines described in (a) to implement a total Trotterized time evolution step of  $dt$ . Subsequent iTEBD<sup>2</sup> sweeps alternate sweep directions, left-to-right and vice-versa.

### 4.5.1. iTEBD<sup>2</sup> algorithm

TEBD-like algorithms perform time evolution by approximating it as successive local time evolutions via the Suzuki-Trotter decomposition of a Hamiltonian, i.e., the sum of local

terms.

$$|\Psi(t + dt)\rangle = e^{-iHdt}|\Psi(t)\rangle \approx \prod_j e^{-ih_j dt}|\Psi(t)\rangle. \quad (4.23)$$

After each local time evolution operation  $e^{-ih_j dt}$ , we find the closest state within the ansatz manifold to represent the time-evolved state. This general approach leads to tensor network implementations as the TEBD algorithm for finite MPS [48, 49], the iTEBD algorithm for iMPS [191, 208], the TEBD<sup>2</sup> algorithm for 2D isoTNS [75], and the simple and full update for TNS time evolution [65, 120, 209]. Here we focus on the application of iTEBD<sup>2</sup> to strip isoTNS.

For a width  $L_x$  strip, we write the Hamiltonian as

$$H = \sum_{c=1}^{L_x-1} H_{c,c+1}, \quad (4.24)$$

where each  $H_{c,c+1}$  is the infinite collection of local operators acting on columns  $c$  and  $c+1$ . Each infinite two-column operator is the sum of local terms, and we will assume the form

$$H_{c,c+1} = \sum_i h_{c,c+1}^{i,i+1}, \quad (4.25)$$

which acts on a plaquette of four spins on columns  $c$  and  $c+1$  and rows  $i$  and  $i+1$ . We work with models where the local term  $h_{c,c+1}^{i,i+1}$  is translationally invariant in the vertical direction, so we will drop the row superscripts.

To perform time evolution, we Trotterize the full Hamiltonian according to

$$e^{-idtH} = e^{-iH_{1,2}dt} e^{-iH_{2,3}dt} \dots e^{-iH_{L_x-1,L_x}dt} \quad (4.26)$$

which is a first-order splitting of the column operators  $H_{c,c+1}$ . We then perform a first-order splitting of the individual plaquette terms  $h_{c,c+1}$  within the column operators. The iTEBD<sup>2</sup> algorithm on an infinite strip using a Hamiltonian of this form is depicted graphically in Fig. 4.4. Our initial isoTNS has the orthogonality column with arrows pointing up as the left-most column. As shown in Fig. 4.4(a), we first merge columns 1 and 2 to form a doubled column with two physical sites per tensor. We then apply the time evolution operator  $e^{-idtH_{1,2}/2} \approx \prod e^{-idth_{1,2}/2}$  to only the physical legs of the doubled column, which flips the isometric arrows to point down. We note that we do not use the standard iTEBD algorithm that enforces at least a two-site unit cell along the column [191], as we do not wish to have a non-trivial unit cell. Instead, we simply apply the two-site gates and do SVD truncation with gauge-fixing to sweep downward until convergence. Following this 1D iTEBD on a doubled column, we apply the chosen iMM algorithm to split this doubled column into an isoTNO  $A_1$  and a new orthogonality column  $|\Phi_2\rangle$ , both of which have isometry arrows pointing up and a single physical leg. We can now repeat this process with columns 2 and 3. Proceeding in this way from left (right) to right (left) on odd (even) iterations of iTEBD<sup>2</sup>, we perform one-time evolution step  $dt$  using  $L_x - 1$  applications of the iMM and 1D iTEBD; this procedure is summarized in Fig. 4.4(b).

Both of these subroutines utilize deterministic SVDs and have complexity  $\mathcal{O}(\chi^4\eta^3)$ <sup>9</sup>, where we assume that  $\chi_h = \chi_v = \chi$ , and allow the bond dimension  $\eta$  along the orthogonality column to be different from  $\chi$ . Hence we see that increasing the strip width incurs a linear increase in the cost of algorithms for systems obeying area law, compared to an exponential increase in costs for 1D algorithms applied to 2D strips.

<sup>9</sup>After merging two columns, the merged column would have bond dimension  $\eta\chi$ . Optionally, compression can be performed after merging two columns and before the iTEBD. In this work, we compress the bond dimension from  $\eta\chi$  down to  $\eta$  before performing the iTEBD. If a compression to bond dimension  $\eta$  is performed before the iTEBD, the complexity is  $\mathcal{O}(\chi^4\eta^3)$ ; otherwise, the complexity is slightly higher,  $\mathcal{O}(\chi^5\eta^3)$ . In practice, we observe a slight deterioration in the result with compression. But we can obtain better results overall by using larger bond dimensions within the same run-time.

## 4.5.2. Ground state search

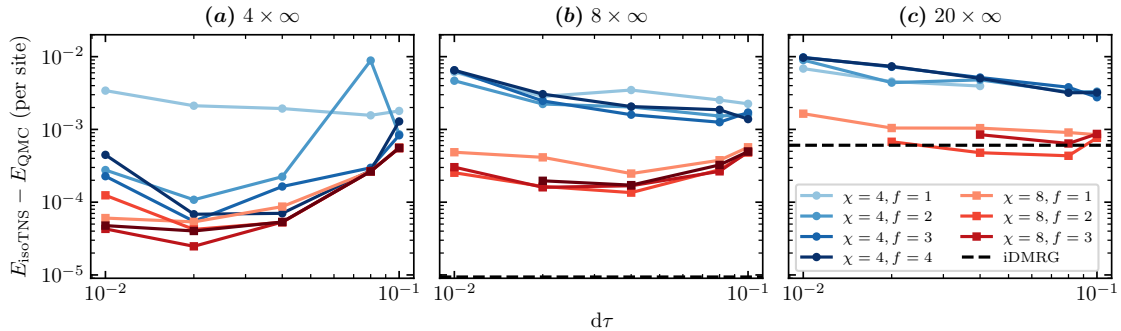


Figure 4.5.: Ground state energies achieved with  $\text{iTEBD}^2$  for paramagnetic 2D TFI with  $g = 3.50$  on an  $L_x = 4, 8$ , and  $20$  strip. We compare the isoTNS energies against essentially exact energies from QMC extrapolated from strips of finite length. As a comparison, the dashed line is the result of an iDMRG calculation with bond dimension  $\chi = 512$ . For  $L_x = 4$ , the iDMRG result is below the bottom axis of the plot.

With the  $\text{iTEBD}^2$  algorithm, we can perform imaginary time evolution to find the ground state of Hamiltonian starting from an initial state  $|\Psi_0\rangle$ :

$$|\Psi_{\text{GS}}\rangle = \lim_{\tau \rightarrow \infty} \frac{e^{-\tau H} |\Psi_0\rangle}{\|e^{-\tau H} |\Psi_0\rangle\|}. \quad (4.27)$$

We benchmark the algorithm with 2D TFI model on strips of width  $L_x = 4, 8, 20$  using isoTNS of bond dimensions  $\chi = 4, 8$  with  $D_b = \chi$ . We investigate a range of  $f \equiv \eta/\chi$  values, where again  $f$  controls the bond dimension of the intermediate  $|\Phi\rangle$  columns produced by IMM during the  $\text{iTEBD}^2$  sweeps. As an essentially exact benchmark, we compute the energy via QMC with the ALPS library [210] on strips of both finite width and length; we find the energy of strips of increasing length and extrapolate to infinite strips. The results are presented in Fig. 4.5. Here, the energy of each isoTNS is calculated by the IMM method discussed in Sec. 4.4.1, using  $\chi = 10$  and  $f = 4$  to give an accurate energy estimate.

If IMM were exact, then the energy would decrease monotonically with time step  $d\tau$ . Yet we clearly see that there exists an energy minimum at intermediate  $d\tau$ . There are competing effects between less Trotter error from a smaller time step but then more error accumulated from an increased number of  $\text{iTEBD}$  and IMM iterations [1, 75]. Additionally, the largest  $f$  does not provide the lowest energy, as one would naively suspect. Larger  $f$  leads to increased vertical bond dimensions during the sweep, leading to larger truncation errors during the  $\text{iTEBD}$  on the doubled column.

## 4.6. Conclusions

In this work, we have extended isometric tensor networks on a square lattice to infinite strip geometries and introduced algorithms to both manipulate and time-evolve the ansatz. We introduced four different IMM algorithms and found the combination of `iMM-Local` and `iMM-Polar` to be efficient and stable. The isoTNOs produced by IMM have a significant disentangling effect on a two-sided MPS. We demonstrated two different applications based on IMM algorithms: (i) the evaluation of local observables and (ii) the  $\text{iTEBD}^2$

algorithm, which enables ground state optimization via imaginary time evolution. These results demonstrate that the isoTNS is a 2D network ansatz permitting both efficient optimization and calculations, and we expect the benefits of this method to become apparent as strip width increases beyond the reach of 1D methods.

We conclude by commenting on interesting applications of our work. Of foremost interest is the extension of our methods to the simulation of strongly correlated many-body systems that are infinite in both directions. Having now dealt with infinite columns through the methods we've introduced, a remaining challenge is to find a fixed point solution to the splitting problem  $A\Psi = \Psi B$ , where  $A$  ( $B$ ) is an infinite isometric column with horizontal arrows pointing right (left) and  $\Psi$  is an infinite orthogonality column. While this is solved in 1D by iterative gauge-fixed QR decompositions, the iMM algorithms, as currently formulated, do not fix the gauge on the horizontal legs, so there is no a priori reason that repeatedly applying the algorithm will converge. Additionally, many interesting physical systems display spontaneous translational symmetry breaking. While in our current prescription, the tensors in a row can differ from one another, each row is repeated along the vertical direction. To allow for a non-trivial unit cell in the infinite direction, we must generalize the iMM methods discussed in Sec. 4.2 to multi-site unit cells. Such modifications are simple extensions of the iMM algorithms introduced earlier. We leave the generalization of iMM to infinite width and exploration of non-trivial unit cells as future works.

A second application is motivated by using tensor networks as sequential circuits on quantum computers [169, 170, 211]. This relies on the isometric condition so that tensors can be interpreted as unitaries acting on qubits and that there is a unidirectional flow of time opposite to the isometry directions. The isometric tensor of bond dimension  $\chi$  would translate into a gate acting across  $\log \chi$  qubits. To construct the circuits that could run on quantum computers, we have to further decompose such ‘‘dense’’ unitaries into quantum circuits consisting of two-site gate [212], resulting in the so-called quantum circuit tensor network. Quantum circuits of finite 2D isoTNS [155, 156] and infinite 1D isoTNS [5, 213, 214] have been numerically and analytically explored for this purpose<sup>10</sup>. With the abovementioned changes, the infinite strip networks developed here can be prepared on a quantum computer, allowing for the calculation of expectation values by directly measuring the state without expensive and approximate boundary contraction methods. Additionally, the finite MM algorithm was recently used to prepare isometric circuits encoding entanglement renormalization principles to accurately measure long-range correlations in critical quantum chains [215]. The iMM algorithm developed here can be used to extend this work to the thermodynamic limit.

---

<sup>10</sup>We explore the quantum algorithm inspired by infinite 1D isoTNSs in Chapter 8.



# 5. Scaling of neural network quantum states for time evolution

*Hope is not a strategy.*

---

Anonymous

In the past few chapters, we have discussed tensor network methods for simulating ground states and dynamics of quantum many-body systems. The typical limit of exponential growth of the many-body Hilbert space is circumvented [30–34], and instead, the simulation is limited by the entanglement. For example, ground states of one-dimensional (1D) local gapped Hamiltonians [47] satisfy the entanglement area-law [37, 39] and can thus be efficiently simulated. However, the application of matrix-product state (MPS) to study 1D non-equilibrium dynamics [48–53] is strongly limited by the fast growth of entanglement. In particular, the entanglement generically grows linearly and hence leads to an exponential growth of the required parameters.

A promising new numerical approach to simulate the dynamics of many-body systems is based on a representation of quantum states using artificial neural networks (ANNs) [76]. Recent works have shown not only promising results for simulating non-equilibrium dynamics [77] but also in describing two-dimensional (2D) critical ground states [78] and states with chiral topological order [79]. Notably, states based on ANNs can efficiently encode volume law entanglement [80, 81] and are thus, per se, not limited by the entanglement growth in non-equilibrium systems. It remains an open question of what limits the representation based on ANNs and how the number of parameters generically scales.

In this Chapter, we benchmark the variational power of a number of different ANNs approaches to simulate quench dynamics in 1D and compare them to MPS-based approximations. We find numerical evidence that states based on ANNs can describe quantum many-body states under quenched time evolution of local Hamiltonian with a number of parameters that grow exponentially in time – which, in fact, resembles the scaling of MPS-based simulations.

The outline of this chapter is as follows: We first describe the variational wavefunctions based on ANNs and the variational method to approximate quantum states using supervised learning in Sec. 5.1. To study the time scales accessible by different variational wavefunctions, we variationally optimize wavefunctions to approximate the exact time evolution in Section 5.2. In Section 5.3, we conclude by discussing the implication of numerical algorithms and possible future directions.

## 5.1. Methods

In this section, we review variational wavefunctions based on general neural networks and autoregressive neural networks. Moreover, we introduce a novel cost function to approximate complex-valued quantum states with autoregressive neural networks in the supervised learning framework [82, 216].

### 5.1.1. Neural-network quantum states

A pure quantum state  $|\Psi\rangle$  describing a quantum many-body system in the Hilbert space  $\mathcal{H} = \otimes_i^N \mathcal{H}_i$  can be represented as

$$|\Psi\rangle = \sum_{\sigma_1, \sigma_2, \dots, \sigma_N} \Psi_{\sigma_1 \sigma_2 \dots \sigma_N} |\sigma_1\rangle \otimes |\sigma_2\rangle \otimes \dots \otimes |\sigma_N\rangle,$$

where  $\{\mathcal{H}_i\}$  and  $|\sigma_i\rangle$  are the local Hilbert spaces and the corresponding local basis states, respectively. The exact representation  $\Psi_{\sigma_1 \sigma_2 \dots \sigma_N}$  requires a number of parameters that grow exponentially with the system size  $N$ . Neural-network quantum states (NQSs) [76] are variational wavefunctions that approximate the amplitudes by neural networks  $f$  with a polynomial number of parameters  $\text{poly}(N)$ ,

$$\Psi_{\sigma_1 \sigma_2 \dots \sigma_N} \approx f(\sigma_1, \sigma_2, \dots, \sigma_N; \mathbf{w}) := \Psi_{\sigma_1 \sigma_2 \dots \sigma_N}^{\text{NN}}, \quad (5.1)$$

where  $\mathbf{w}$  denotes the parameters. One of the motivations to consider neural networks is the universal approximation theorem [217–221], which guarantees an arbitrarily close approximation of the given function provided a sufficiently large number of hidden neurons is included (a review of neural networks can be found in Appendix C.1). Various different types of neural networks have been considered for approximating quantum states, including restricted Boltzmann machines (RBMs) [222, 223] in Ref. 76, fully-connected feed-forward neural networks (NNs) in Ref. 224, convolutional neural networks (CNNs) in Ref. 225, and recurrent neural networks (RNNs) [226, 227] in Refs. 228, 229. NQSs found applications in studying ground states and low-energy states [230–234], quantum state tomography [235], simulating open quantum systems [229, 236–240], and real-time evolution [76, 77, 241–244].

In general, NQSs have the following properties: The unnormalized probability amplitude and its gradient  $\nabla_{\mathbf{w}} \Psi^{\text{NN}}$  can be efficiently evaluated at a cost of  $\mathcal{O}(N^2)$ . This property allows one to combine NQSs with conventional Monte Carlo methods. For example, the expectation value of few-body observables  $\hat{O}$  can be computed as

$$\frac{\langle \Psi^{\text{NN}} | \hat{O} | \Psi^{\text{NN}} \rangle}{\langle \Psi^{\text{NN}} | \Psi^{\text{NN}} \rangle} = \sum_{\sigma} \frac{|\Psi^{\text{NN}}(\sigma)|^2}{\langle \Psi^{\text{NN}} | \Psi^{\text{NN}} \rangle} O_{\text{local}}(\sigma), \quad (5.2)$$

where  $O_{\text{local}}(\sigma) = \sum_{\sigma'} \langle \sigma | \hat{O} | \sigma' \rangle \Psi^{\text{NN}}(\sigma') / \Psi^{\text{NN}}(\sigma)$ . This expectation value can be estimated using Markov chain Monte Carlo (MCMC) sampling from the normalized probability  $P_{\Psi^{\text{NN}}}(\sigma) = \frac{|\Psi^{\text{NN}}(\sigma)|^2}{\langle \Psi^{\text{NN}} | \Psi^{\text{NN}} \rangle}$ . One can then minimize the energy expectation value by stochastic gradient descent (SGD) method. This method is also known as variational Monte Carlo (VMC) [245–248] and is often combined with the second-order optimization method, stochastic reconfiguration (SR) [248–250], a.k.a. imaginary time-dependent variational principle or natural gradient descent [251]. While being a versatile method, the NQS-based method has potential drawbacks, including the intractable normalization and the long auto-correlation time in MCMC, which can lead to difficulties in training large deep neural networks.

### 5.1.2. Neural autoregressive quantum states

A proposed solution to resolve the potential drawbacks mentioned above is the neural autoregressive quantum states (NAQSs) [78, 228, 229, 252]. NAQSs are NQSs with causal restrictions in the connectivity of the networks. Concretely, NAQSs approximate the conditional probability amplitudes of a given ordering of sites  $\{1, \dots, N\}$  by a neural network,

$$\Psi_{\sigma_1 \sigma_2 \dots \sigma_N} = \Psi_{\sigma_1} \Psi_{\sigma_2 | \sigma_1} \dots \Psi_{\sigma_N | \sigma_1 \dots \sigma_{N-1}} \quad (5.3)$$

$$\approx f_{\text{NN}}^{[1]}(\sigma_1) f_{\text{NN}}^{[2]}(\sigma_2 | \sigma_1) \dots f_{\text{NN}}^{[N]}(\sigma_N | \sigma_1 \dots \sigma_{N-1}) \quad (5.4)$$



where the network has  $N$  outputs. Each output represents a conditional probability amplitude on each site  $\Psi_j := \Psi_{\sigma_j|\sigma_{i<j}}$ . The fact that  $\Psi_j$  can only depend on  $\sigma_{i<j}$  leads to a causal structure. Since every single site conditional probability amplitude  $\Psi_j$  can be easily normalized, the multiplication of them gives a normalized variational quantum state (see Fig. 5.1).

As a subclass of NQSs, NAQSs share the same properties of NQSs where one can efficiently evaluate the probability amplitudes and the gradients. Furthermore, a NAQS represents an autoregressive model. An autoregressive model can be sampled by the ancestral sampling algorithm which allows sampling directly from the probability  $P_{\Psi^{NN}}(\boldsymbol{\sigma}) = |\Psi^{NN}(\boldsymbol{\sigma})|^2$  (see Appendix C.2 for further details). As a result, quantum states represented by a deep neural network with millions of parameters can be trained efficiently using the SGD algorithm for studying ground states [78].

### Neural Autoregressive Quantum States

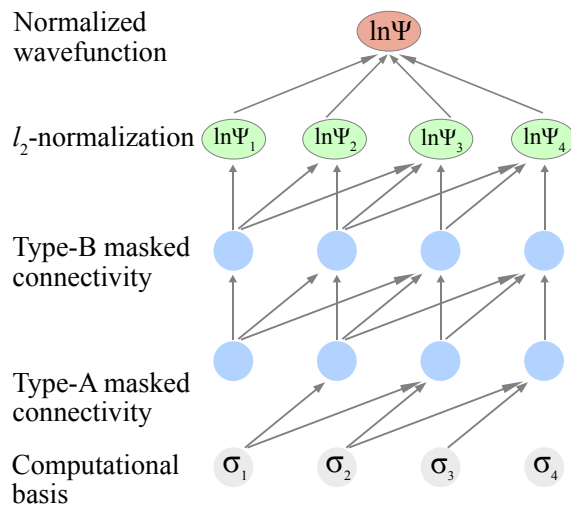


Figure 5.1.: Illustration of a neural autoregressive quantum state. The neural network approximates the conditional probability amplitudes  $\Psi_1, \Psi_2, \dots$  in the log space. Each of the conditional probability amplitudes is normalized to give a normalized wavefunction. The network connectivity is restricted (masked) for causality, such that  $\Psi_j$  only depends on  $\sigma_1 \dots \sigma_{j-1}$ . The network itself could be either a fully-connected neural network or a convolutional neural network.

#### 5.1.3. Supervised learning for quantum state approximation

We now describe the variational problem of approximating complex-valued wavefunctions in the supervised learning setting and propose a new cost function for normalized variational wavefunctions, e.g., NAQSs, to jointly learn the probabilities and phases.

Given a target wave function  $|\Phi\rangle$ , we would like to find the optimal NAQS  $|\Psi^{NN}\rangle$  approximating the target wave function by minimizing a cost function  $\mathcal{R}(\Phi, \Psi^{NN})$ . In the following, we consider the setup that we are given the samples  $\{\boldsymbol{\sigma}, \Phi(\boldsymbol{\sigma})\}$  according to the probability distribution  $P_\Phi(\boldsymbol{\sigma}) = |\Phi(\boldsymbol{\sigma})|^2$  instead of the full wave function. It is a practical setup for the exponentially large Hilbert space. In the supervised learning setup, a loss function  $\mathcal{L}$  measures the difference between the data points  $\langle \boldsymbol{\sigma} | \Phi \rangle, \langle \boldsymbol{\sigma} | \Psi^{NN} \rangle$ . A cost function  $\mathcal{R}$  is the expectation value of the loss function  $\mathcal{L}$  with respect to the probability  $P_\Phi(\boldsymbol{\sigma})$  and is, therefore, a measure of the difference between the quantum states  $|\Phi\rangle, |\Psi^{NN}\rangle$ . As a result, the optimization problem can be solved using the SGD method, where the

stochastic gradient is computed from the sampled cost function  $\hat{R}$  (see Appendix C.3 for a detailed discussion).

Here we present a joint learning scheme for a complex-valued wave function where the cost function  $\mathcal{R}_{\text{joint}}$  is the sum of the so-called Kullback–Leibler (KL) divergence of the probabilities  $\mathcal{R}_{\text{KL}}$  and the weighted  $L^2$  distance of the phases  $\mathcal{R}_\theta$ ,

$$\mathcal{R}_{\text{joint}} = \mathcal{R}_{\text{KL}} + \mathcal{R}_\theta. \quad (5.5)$$

In order to learn the magnitude of the probability amplitude  $\text{Re} [\log \Psi^{\text{NN}}(\boldsymbol{\sigma})]$ , we consider the forward KL-divergence  $\mathcal{R}_{\text{KL}}$ , which is defined as

$$\begin{aligned} \mathcal{R}_{\text{KL}} &= \sum_i^{D_{\mathcal{H}}} P_\Phi(\boldsymbol{\sigma}_i) \log \left( \frac{P_\Phi(\boldsymbol{\sigma}_i)}{P_{\Psi^{\text{NN}}}(\boldsymbol{\sigma}_i)} \right) \\ &\approx \frac{\sum_{\boldsymbol{\sigma}_i \sim |\Phi(\boldsymbol{\sigma}_i)|^2} 2 \text{Re} [\log \Phi(\boldsymbol{\sigma}_i) - \log \Psi^{\text{NN}}(\boldsymbol{\sigma}_i)]}{N_{\text{samples}}}. \end{aligned} \quad (5.6)$$

It is a common cost function for learning a probability distribution when one can sample from the exact distribution, which is equivalent to maximal likelihood learning.

For learning the phase of the probability amplitude  $\text{Im} [\log \Psi^{\text{NN}}(\boldsymbol{\sigma})]$ , we consider the weighted  $L^2$  distance of the phases embedded on a unit circle,

$$\begin{aligned} \mathcal{R}_\theta &= \sum_i^N |\Phi(\boldsymbol{\sigma}_i)|^2 \text{dist}(\text{Im} [\log \Phi(\boldsymbol{\sigma}_i)], \text{Im} [\log \Psi^{\text{NN}}(\boldsymbol{\sigma}_i)]) \\ &= \sum_i^N |\Phi(\boldsymbol{\sigma}_i)|^2 \text{dist}(\theta^\Phi(\boldsymbol{\sigma}_i), \theta^{\Psi^{\text{NN}}}(\boldsymbol{\sigma}_i)) \\ &\approx \frac{1}{N_{\text{samples}}} \sum_{\boldsymbol{\sigma}_i \sim |\Phi(\boldsymbol{\sigma}_i)|^2} \text{dist}(\theta^\Phi(\boldsymbol{\sigma}_i), \theta^{\Psi^{\text{NN}}}(\boldsymbol{\sigma}_i)) \end{aligned}, \quad (5.7)$$

with

$$\text{dist}(\theta_1, \theta_2) = (\cos(\theta_1) - \cos(\theta_2))^2 + (\sin(\theta_1) - \sin(\theta_2))^2.$$

The stochastic gradient can be computed from the sampling estimate of the cost functions in Eq. (5.6) and Eq. (5.7). During the optimization procedure, the magnitudes and phases are learned at the same time. This cost function is only valid for normalized wavefunction because the KL-divergence is unbounded from below if  $P_{\Psi^{\text{NN}}}(\boldsymbol{\sigma}_i)$  is an unnormalized probability distribution. As a result, we consider NAQSS which ensure the normalization without additional cost.

The advantage of choosing the cost function  $\mathcal{R}_{\text{joint}}$  is that the resulting stochastic gradient is unbiased and with low variance. The optimization problem can be solved by SGD with a small mini-batch size or even a single data input at a time. It is important to have this property for efficiently solving the variational state approximation problem with a large neural network and when the Hilbert space is large.

Another cost function, which also can be rewritten as an expectation value and gives a physical interpretation, is the real part of the negative overlap,

$$\begin{aligned} \mathcal{R}_{\text{neg. overlap}} &= -\text{Re} \left[ \sum_i^{D_{\mathcal{H}}} \Phi^*(\boldsymbol{\sigma}_i) \Psi^{\text{NN}}(\boldsymbol{\sigma}_i) \right] \\ &= -\sum_i^{D_{\mathcal{H}}} P_\Phi(\boldsymbol{\sigma}_i) \text{Re} \left[ \frac{\Psi^{\text{NN}}(\boldsymbol{\sigma}_i)}{\Phi(\boldsymbol{\sigma}_i)} \right] \end{aligned} \quad (5.8)$$

We observed that it has a similar performance as the joint cost function. In Appendix C.3, we show the two cost functions are indeed equivalent to first order in error. In principle, one should include a constant term  $c_{\text{ratio}}$  in the equation  $\mathcal{R}_{\text{joint}} = \mathcal{R}_{\text{KL}} + c_{\text{ratio}}\mathcal{R}_{\theta}$ , where the  $c_{\text{ratio}}$  is a hyper-parameter to be tuned controlling the ratio between the two cost functions. We consider  $c_{\text{ratio}} = 1$  because the equivalence to  $\mathcal{R}_{\text{neg. overlap}}$  in the small error limit.

In passing, we mention that several different proposals for the cost functions have been discussed in the literature. This includes the  $L^2$  distance of the probability amplitudes [224, 253], the weighted  $L^2$  distance of log of the probability amplitudes [254], and negative log fidelity [255]. We discuss the detail of the difference between different cost functions in Appendix C.3

Note that the variational algorithm to approximate quantum states using neural networks has found applications in imaginary-time evolution [253], real-time evolution [241], simulating open quantum system [229], classical simulation of quantum computation [255] and the study of the expressivity and generalization properties [6, 224, 254, 256].

## 5.2. Numerical results

To test the variational power of different networks, we consider far-from-equilibrium dynamics resulting from global quenches for a paradigmatic 1D model. The fast (ballistic) growth of entanglement prohibits the classical simulation based on MPS at long times. Recent work with NQSs suggests that NQSs could reach time scales comparable with or exceeding those of the state-of-the-art tensor network methods [77]. Here, we consider a similar setup with a  $x$ -polarized initial product state  $|\psi_0\rangle = \prod_i |\rightarrow\rangle_i$ . We then evolve the state using the Hamiltonian of the quantum Ising model with transverse ( $g$ ), longitudinal ( $h$ ) fields, and an interaction term ( $k$ ):

$$\hat{H} = -J \left[ \sum_{j=1}^{N-1} (\hat{\sigma}_j^z \hat{\sigma}_{j+1}^z + k \hat{\sigma}_j^x \hat{\sigma}_{j+1}^x) + \sum_{j=1}^N (g \hat{\sigma}_j^x + h \hat{\sigma}_j^z) \right]. \quad (5.9)$$

We consider two cases that exhibit different growth of entanglement: a weak quench to  $g = 3$ ,  $h = 0.1$ ,  $k = 0$  with slow growth of entanglement and a strong quench near the critical point  $g = 1$ ,  $h = 0$ ,  $k = 0.25$  with fast growth. Note that the model is non-integrable for both parameter sets. In the following, we consider open boundary conditions (OBC) and provide additional data for periodic boundary conditions (PBC) in Appendix C.5.

### 5.2.1. Approximation with MPSs

We first variationally approximate the exact time-evolved state using the MPS ansatz with small bond dimension  $\chi$  and reproduce the well-known result that the number of parameters scales exponentially in time. In particular, we perform the exact simulation of time evolution following the quench to obtain the target states  $|\Phi_{\text{exact}}(t)\rangle$  at certain discrete times using exact diagonalization (ED). We then find the MPS with the given bond dimension  $\chi$  approximating the target state at each time  $t$  by SGD over the cost function  $\mathcal{R}_{\text{joint}}$ . Note that the MPS found by supervised learning with the SGD method is close to the optimal result obtained by iteratively minimizing the  $L^2$  distance of the quantum states [31]. We include this additional result in Appendix C.5 for comparison. The details of the parameters used in the optimization are shown in Appendix C.4.

In Fig. 5.2 we show the error in fidelity after the quantum state approximation for the two parameter sets of the Ising model. We also show the expectation value  $\langle \hat{\sigma}_x \rangle$  in the middle of the chain and the half-chain von Neumann entanglement entropy  $S_{vN}$ . For a fixed bond dimension  $\chi$ , an MPS can accurately approximate the state under time evolution to a

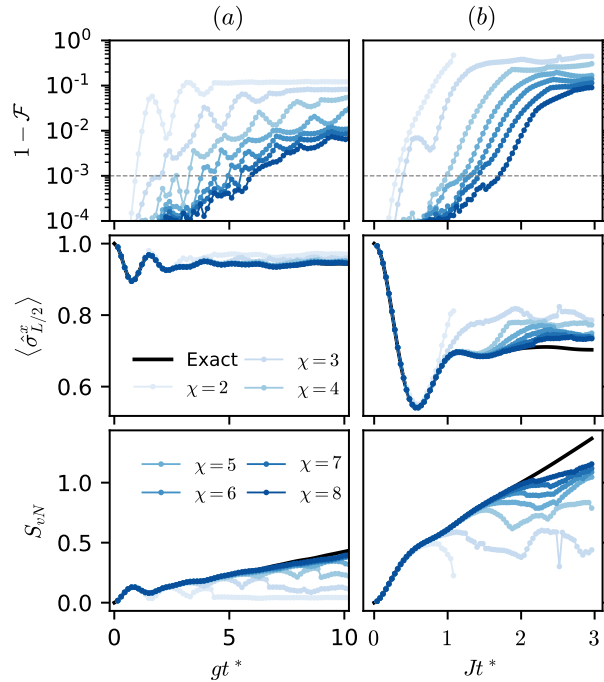


Figure 5.2.: MPSs with bond dimension  $\chi$  approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The MPSs are obtained by supervised learning with SGD. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in the paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ).

certain timescale after which the error in fidelity grows exponentially to a saturation value. This timescale coincides with the deviation in the local observable  $\langle \hat{\sigma}_x \rangle$  and the saturation of entanglement.

To quantitatively access this time scale, we determine the reachable time  $t^*$  by the time when the error in fidelity exceeds a threshold value  $1 - \mathcal{F} = 10^{-3}$ . This is indicated by the grey dashed lines in Fig. 5.2. We then plot the number of parameters of the MPS as a function of the time  $t^*$  that can be reached in Fig. 5.4. The data shows a clear exponential growth of the number of parameters as a function of  $t^*$ . This is directly related to the linear growth of the entanglement  $S$  entropy [56], which yields an exponentially growing bond dimension  $\chi \sim e^S$  to correctly capture the underlying states [257, 258].

Note that the exponential scaling of the number of parameters with  $t^*$  is not affected by the threshold value chosen as long as the threshold value for the error in fidelity  $1 - \mathcal{F}$  is chosen below the value where the curves flatten due to finite size effects.

### 5.2.2. Approximation with shallow NAQs

We now investigate the variational power of NAQs in a similar setup as before. Namely, given the exact time-evolved state  $|\Phi(t)\rangle$  at the selected time  $t$ , we perform supervised learning to approximate the state variationally with NAQs. As a first example, we consider a one-hidden layer convolutional (CNN-1) NAQs. The ratio of the number of hidden neurons to the system size  $\alpha = N_{\text{hidden}}/N$  controls the width of the network. For CNNs,  $\alpha$  is the number of channels. Increasing the width of the network increases the number

of parameters and the variational power of the network. We optimize the cost function described in Section 5.1.3 using SGD with the Adam [259] optimizer. We begin with a batch size of 512 samples and a learning rate of  $10^{-3}$  which is then decreased to  $10^{-4}$  once the sampled cost converges. Technical details are described in Appendix C.1.

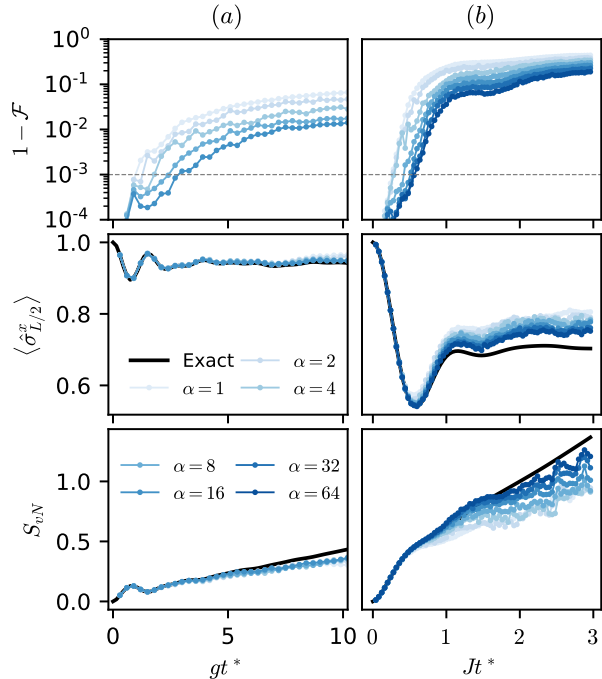


Figure 5.3.: One-hidden layer convolutional NAQS of different width  $\alpha$  approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in the paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ).

In Fig. 5.3, we show the results obtained for the quantum Ising model described in Eq. (5.9). We observe for a fixed network that the accuracy of the approximation decreases with time. The error increases exponentially and saturates at a size-dependent final value as in the case of MPSs. Moreover, we find that the accuracy improves when increasing the width  $\alpha$  of the networks. Identically to the procedure for MPSs, we determine the reachable time  $t^*$  by the time when the error in fidelity exceeds the threshold value  $1 - \mathcal{F} = 10^{-3}$  indicated by the grey dashed lines. We notice that the entanglement entropy also deviates from the exact value but does not saturate even at long times after the accessible time  $t^*$ . This result is plausible since NAQSs can indeed represent volume law states.

In Fig. 5.4, we plot the number of parameters against this reachable time  $t^*$  with various widths  $\alpha$ . The result suggests an exponential scaling of the number of parameters with the reachable time  $t^*$ . This is identical to the scaling behavior of MPSs but the growth constant is in fact larger.

A big advantage of NAQSs is the flexibility in the design of network architectures. Essentially, NAQSs are broad classes of wavefunctions based on different network architectures. With a fixed network architecture, two common factors to control the model complexity are the network width and network depth [260–262]. An important question is how these choices affect the network expressivity, hence the quality of the approximation.

To understand the effect of the network width on the expressivity, we focus on shallow

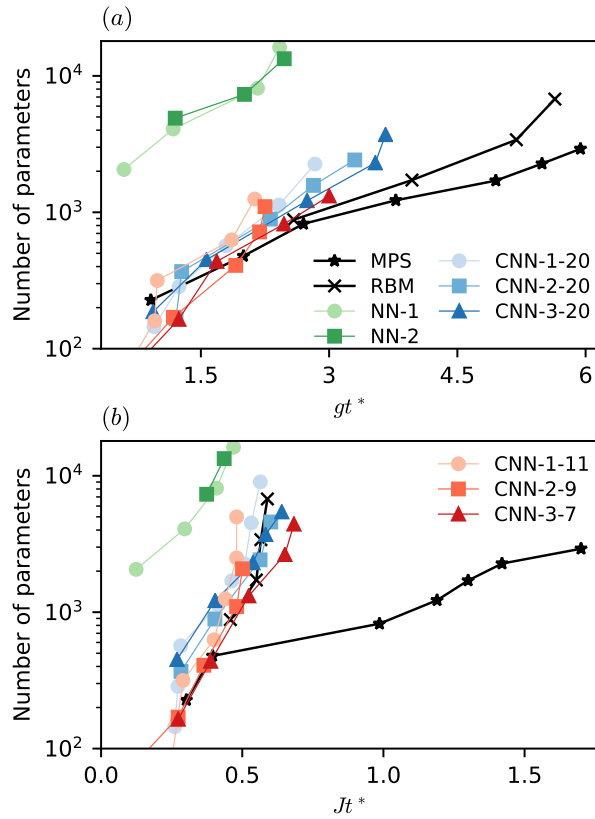


Figure 5.4.: Effect of increasing network width: The number of parameters required to reach time  $t^*$  for following a quantum quench for different approximation schemes. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ). The black line with star (cross) markerS shows the results for MPS (RBM). For NAQS, we denote each combination as (network type)-(number of hidden layers)-(filter size). The circles, squares, and triangles represent 1,2,3-layer networks respectively. The gradient colors describe the same networks of different numbers of hidden layers. NNs (Blue); CNNs with global connectivity (Green-Blue); CNNs with local connectivity (Red).

NNs with at most 3-hidden layers and consider three types of network architectures: (i) fully-connected neural networks (NNs), (ii) CNNs with a large filter size equaling to the system size, and (iii) CNNs with small filter sizes but have receptive fields (causal cone) covering the full system. For simplicity, we fix the same network width  $\alpha$  over the hidden layers. For a given combination of the network type and the number of hidden layers, we repeat the procedure of state approximation described above with an increasing network width  $\alpha$ <sup>1</sup>. We plot the number of parameters of the network against the obtained accessible times  $t^*$  in Fig. 5.4. We denote each combination as (network type)-(number of hidden layers) and with (filter size) if a CNN is considered.

From the data, we observe consistent exponential scaling of the number of parameters with the reachable time  $t^*$  by increasing the width  $\alpha$  for all network architectures and depths considered. Changing the network architecture results in a constant shift and a

<sup>1</sup>In the context of CNN, increasing the width means increasing the number of channels.

slight change in the growth rate in the scaling. There is a significant difference between NNs and CNNs, where we see an order of magnitude improvement likely due to the sharing of parameters<sup>2</sup>. The difference between CNNs with small and large filter sizes is, however, not obvious. While here we focus on shallow neural networks, the data still shows a slight improvement in the growth rate when increasing the network depth. Notably, the scaling of NNs deviates more from that of MPS for parameter set (b). It is different from the naive expectation that MPS would perform less efficiently when the entanglement grows rapidly.

We would like to point out that the scaling does not depend on boundary conditions—the same exponential scaling is observed for systems with PBC (see Appendix C.5). Due to the causality requirement of the NAQs, we have zero padding instead of periodic padding for CNNs. This implies that the wavefunction is not translational invariant; hence, it can be applied to both OBC and PBC systems. While we found a significant constant shift in scaling between CNNs and NNs due to parameter sharing, we expect an even more significant shift for other types of NQs where one enforces exact translational symmetry.

### 5.2.3. Approximation with deep NAQs

To study the effect of network depth on expressivity, we consider two types of deep network architectures: (i) CNNs with gated activation unit and residual connection (Gated-CNNs) (ii) WaveNets [263]. Both networks consist of modular building blocks. One can systematically increase the network depth by stacking more blocks. The building block of Gated-CNNs includes causal convolution following a gated activation unit and a residual connection [260]. The building block of WaveNets replaces the standard causal convolution with dilated causal convolution, which allows the size of the receptive field (causal cone) to grow exponentially with the network depth, and it includes an additional parameterized skip connection. We show the building block and the network architecture in Fig C.1 and describe the detail of the network architectures in Appendix C.1. For a fixed network architecture and width  $\alpha$ , we increase the network depth to increase the number of parameters and repeat the procedure of state approximation. We vary the number of blocks between 6 to 14 for WaveNets and 10 to 20 for Gated-CNNs.

In Fig. 5.5, we show the results for Gated-CNNs of width  $\alpha = 12$ . We find a similar result as in Fig. 5.3. With a larger network, it is now obvious that the entanglement entropy deviates but does not saturate long after the accessible time. We even observe slightly larger entanglement entropy at the deviation (See also data for WaveNets in Appendix C.5).

In Fig. 5.6, we plot the number of parameters of networks with different depths against the accessible times  $t^*$  and denote each combination as (network type)-(network width  $\alpha$ ). We observe again consistent exponential scaling of the number of parameters with the reachable time  $t^*$  by increasing the network depth for the two networks and widths considered. Similarly, we see a small constant shift for different network widths favoring “narrower” networks. More importantly, data for both network architectures roughly fall on the same line despite the drastically different design choices of using dilated or normal convolution. Compared with the data from shallow CNNs where we vary the network width, all data again fall roughly on the same line. This suggests that increasing the number of parameters by increasing network width has the same effect on expressivity as increasing the network depth for the cases considered. Even more surprisingly, such results are independent of the network architectures of whether one considers global or local, dilated or normal convolution, with or without skip and residual connections. Note

---

<sup>2</sup>CNNs utilize translational invariance such that the convolution kernels are applied across the system size, instead of independent parameterized weight acting on each site. Therefore, the parameters are “shared” compared to typical NNs.



that, however, the optimization problem may be strongly affected by the choice of different network architectures. For example, a deep neural network without shortcut connections may be hard to optimize.

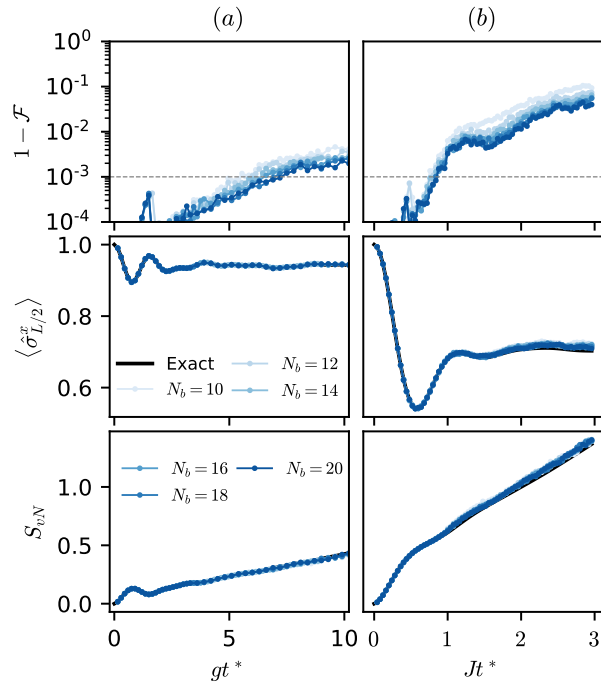


Figure 5.5.: Gated-CNNs NAQS of different depth, i.e. number of blocks  $N_b$ , approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The network is of width  $\alpha = 12$ . The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in the paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ).

We further note that there are two main differences in the network structure considered here and in previous works [76, 77, 241, 244]. The first difference is the autoregressive property. It is unknown how the autoregressive constraint affects the expressivity of the neural networks. The second difference is the activation function. In Ref. 77, 241, the choice of polynomial activation with complex-valued weight is crucial to the result. Notice that such networks [221] do not have the universal approximation property. In contrast, we consider real-valued weights with the rectified linear unit (ReLU) [264], tanh, and sigmoid activation functions. For shallow NNs, we observe that changing the activation function does not affect the result significantly. The difference in the setup calls into question whether the good result before [77] is not a general property of neural networks.

#### 5.2.4. Approximation with RBMQSs

As the last example, we study the expressivity of the restricted Boltzmann machine quantum states (RBMQSs) [76] with increasing width. One prominent difference between RBMQSs and NAQSs is that RBMQSs belong to general (non-autoregressive) neural networks, and hence is not normalized. The normalization constant is in general intractable. Nevertheless, it is possible to estimate the fidelity to another wavefunction by stochastic sampling. The gradient of such stochastic estimates gives a biased stochastic gradient,



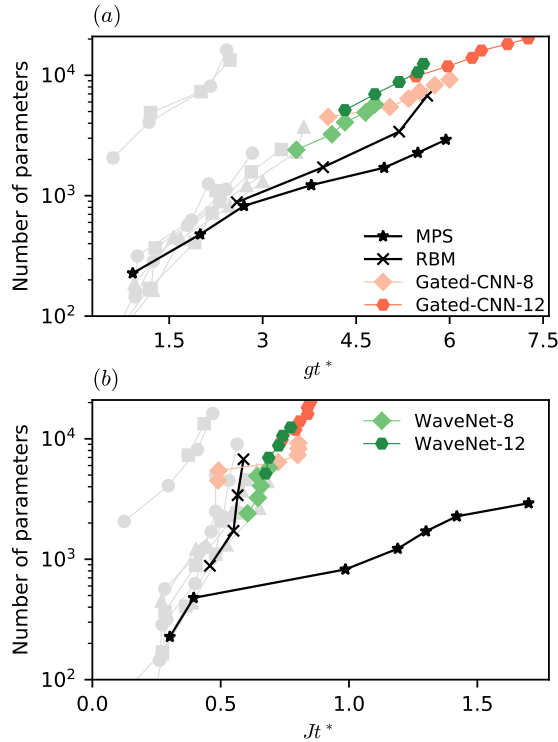


Figure 5.6.: Effect of increasing network depth: The number of parameters required to reach time  $t^*$  for following a quantum quench for different approximation schemes. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in paramagnetic phase with weak longitudinal field ( $g = 3, h = 0.1, k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1, h = 0, k = 0.25$ ). The black line with star (cross) markers shows the results for MPS (RBM). For NAQS, we denote each combination as (network type)-(network width  $\alpha$ ). The diamonds and hexagons represent a network of width  $\alpha = 8, 12$  respectively. The gradient colors describe the same networks of different network widths. Gated-CNNs (Red); WaveNets (Green); we keep the shallow NAQSSs data (light grey) from Fig. 5.4 for comparison.

which works well in practice for supervised learning with stochastic gradient methods.

We repeat the similar benchmark of state approximation on networks of increasing width  $\alpha = N_{\text{hidden}}/N = \{1, 2, 4, 8\}$ . We minimize the negative fidelity between the RBMQS and the target state by SGD. We confirm our implementation yields similar results to the NetKet [6] implementation using the SR method. For more detail, see the discussion of the cost function in Appendix C.3 and the additional data for consistency check in Appendix C.5.

The result for state approximation using RBMQSSs is shown in Fig. C.6 in Appendix C.5, which resembles the results of NAQSSs shown in Fig. 5.3 and Fig. 5.5. We plot the reachable time  $t^*$  in Fig. 5.4 and Fig. 5.6. We observe again consistent exponential scaling of the number of parameters with the reachable time  $t^*$  by increasing the network width.

We would like to point out that our observation in the reachable time is consistent with previous works [76, 244] which show similar difficulties around  $Jt = 0.5$  for representing states quenched by Hamiltonian near the critical point using RBM-based NQS. It is also observed in [241] that a lack of expressibility from the RBM-based NQS leads to a rapid

increase in the error in fidelity in a short time scale for a quench across critical points.

Lastly, similar scaling plots as in Fig. 5.4 and Fig. 5.6 could also be obtained in a more realistic setup when exact data is not known. For MPSs, one could perform the time-evolving block decimation (TEBD) algorithm [48, 49], and only increase the bond dimension  $\chi$  when the approximation (truncation) error reaches the threshold value  $10^{-3}$ . For NQSs, one could perform a time evolution algorithm based on supervised learning [229, 241] and increase the size of the neural network when the approximation error reaches the threshold value  $10^{-3}$ . If we plot the number of parameters of the variational wavefunctions to the time steps, we could obtain a similar scaling plot.

### 5.3. Conclusion

We provide numerical evidence that the required number of parameters to represent quantum states following a global quench by neural networks grows exponentially in time. Thus, despite the ability to represent highly-entangled states, the NQSs, including NAQSs and RBMQSs, considered in this work cannot represent time-evolved states efficiently. An important remaining question is to understand this scaling. The scaling is agnostic to the design of network architectures we considered but is only affected by the difference in architecture related to symmetry, i.e. convolution. The scaling of NQSs resembles that of MPSs. However, for the 1D models considered, NQSs show a larger growth rate of parameters in time while its entanglement entropy saturates slower. Explaining this difference and understanding the limitation of NQSs may require a new measure for the complexity of states different from the entanglement entropy.

The proposed cost function for joint learning magnitudes and phases with NAQSs ensures an unbiased and low-variance stochastic gradient. It could be applied to the real-time evolution for pure states and mixed states [229, 241] and the imaginary-time evolution [253]. We expect it to speed up the state approximation at each time step and to be suitable for learning with deep neural networks. Monitoring the training and convergence of the two costs  $\mathcal{R}_{\text{KL}}$  and  $\mathcal{R}_\theta$  gives information about the hardness of learning probability distributions and the phases. We observe similar difficulty in learning both the magnitudes and the phases for time-evolved states. This may provide additional insight for learning frustrated ground states, which previous works [254, 265] show the main difficulty comes from learning the phases.

Finally, our findings do not invalidate the simulation for time evolution with NQSs in general but rather provide insight into the choice of networks to reach the desired time scale. It also suggests the importance to consider dynamically increasing the network size for the time evolution algorithm based on supervised learning. It is still important to search for different network architectures [260, 263, 266–269], which might be more efficient in optimization or better in the scaling for different Hamiltonians. Network architectures incorporating the symmetry of the states have the potential to give better scaling. Moreover, the approach based on neural networks is less affected by the spatial dimension of the systems than that based on tensor networks. This suggests the potential advantage of NQSs over tensor network states (TNSs) in two and higher dimensions.

While understanding the expressivity and scaling of NQSs is the essential step for designing practical algorithms, the recent work [270] studies the numerical instability in obtaining the NQS using time-dependent VMC when the time-evolved states can be expressed as NQSs accurately. The article [270] and this work complement each other and provide insights for simulating time evolution using NQSs. The code implementation of this work is based on TensorFlow [271] and is available at [272].

Part II.

# Algorithms for quantum simulations



# Overview of Part II

As mentioned in the introduction, Part II of this thesis focuses on the quantum algorithms for simulating quantum many-body dynamics.

## Chapter 6: Review of sequential quantum circuits

Chapter 6 provides a comprehensive review of sequential quantum circuits, the variational ansätze considered in both Chapter 7 and Chapter 8. We start with the overview of gate-based quantum computation and focus on the connection between isometric tensor network states (isoTNSs) and sequential circuits. We first show the formal equivalence of isoTNSs and the sequential circuits consisting of  $n$ -site unitary. Then, we consider the approximate decomposition of the  $n$ -site unitaries into local circuits of 2-site gates. We refer to the resulting ansätze the sparse sequential circuits, alternatively referred to as the quantum circuit tensor networks (QCTNs) in the literature.

We end the chapter by discussing the number of 2-site gates required to approximate a given  $n$ -site unitary, which relates to the concept of quantum complexity that measures the computational complexity of implementing the given unitary on the quantum computer. By comparing the classical computational complexity of classical algorithms with the quantum complexity of quantum algorithms, we discuss the tasks in which potential quantum advantages are to be expected.

## Chapter 7: Real- and imaginary-time evolution with compressed quantum circuits

Chapter 7 investigates the properties of the sparse sequential circuits on one-dimensional (1D) finite systems introduced in Chapter 6. We study the representation power of the sparse sequential circuits and confirm that exponentially fewer parameters compared to matrix-product states (MPSs) are required to represent the time-evolved state generated under non-equilibrium quantum dynamics. Based on the efficient representation, we propose a hybrid quantum-classical algorithm for time evolution. We benchmark the algorithms classically by finding the ground state with imaginary time evolution and simulating a global quench of the transverse field Ising model with a longitudinal field. Finally, we run the classically optimized gate sets on the IBM quantum processing unit (QPU) and show qualitatively consistent results with the classical simulation.

## Chapter 8: Time evolution of uniform sequential circuits

Chapter 8 continues the exploration of Chapter 7 and extends it to the infinite case, where the uniform MPS motivated sparse sequential circuits on infinite 1D systems are introduced. We name the ansätze the layered uniform sequential circuits (l-USCs).

Notably, quantum algorithms for infinite translationally invariant systems are still rarely considered. Inspired by the tensor network algorithm, we introduced the fundamental concept of the left and right environment of the infinite circuit, which are the fixed points of the (mixed) transfer matrix. We introduced algorithms to find these environments and measure physical observables for the infinite quantum circuit on a quantum device with a finite number of qubits.

We then turn to the investigation of the representation power of the ansätze. By tuning the two complexity parameters, the width and the depth of the local circuit, the observations reveal a similar overall linear complexity growth in the state complexity for simulating dynamics, similar to the findings from the finite case. Interestingly, a quadratic growth of complexity with time is found for circuits representing the environments.

A hybrid quantum-classical algorithm is proposed to time evolve 1D uniform systems in the thermodynamic limit. We represent both the infinite translationally invariant quantum states and the environments using l-USCs. We show numerically that this ansatz requires a number of parameters scaling polynomially in the simulation time for a given accuracy. After benchmarking the evolution algorithm on a classical computer, we demonstrate the measurement of observables of this uniform state using a finite number of qubits on a QPU.

## 6. Review of sequential quantum circuits

In the first part of the thesis, we see that it is hard for classical computers to simulate quantum dynamics. Fundamentally, this is due to the fact that classical computers, such as laptops, personal computers, cellphones, etc., are based on classical physics. In the early 1980s, people started to consider the possibility of utilizing quantum mechanics for simulating quantum systems [98, 273, 274]. The proposals are now understood as quantum simulators or analog quantum computation, where the “quantum computer” contains some qubits and can be controlled to realize certain dynamics resembling the dynamics of a model system of interest. In 1985, David Deutsch defined the universal quantum Turing machine [275], which leads to the digital quantum computation or gate-based quantum computation.

While analog quantum simulation has been an active area of research over the past two decades [276, 277], recently digital quantum simulation has started to enter the game. Analog quantum simulation has shown the ability to study dynamics that may be hard to simulate classically [22, 84, 278–282]. In the long run, we expect digital quantum computers to be the better quantum simulators because of the general purpose framework and the ability to implement quantum error correction [283]. In this chapter, we review the basics of gate-based quantum computation in Sec. 6.1, show the connection between isometric tensor network states (isoTNSs) and sequential circuits in Sec. 6.2, and discuss the classical and quantum complexity of simulating the problem in Sec. 6.3.

### 6.1. Gate-based quantum computation

Digital quantum computers, which perform gate-based quantum computation, are computers with a finite number of qubits and a universal set of quantum gates. Furthermore, they need to have the ability to prepare an initial state and measure the final state in the computational basis [284]. An algorithm run on a digital quantum computer prepares an initial state, transforms the state by a sequence of unitary gate operations, and performs measurements. Each of these parts poses the challenge for realizing a large-scale quantum computer: (i) scalable number of qubits, (ii) accurate quantum gates, (iii) accurate state preparation, (iv) long qubit coherence time, and (v) accurate measurement. Experimental realizations of gate-based quantum computers have been achieved using a variety of physical systems, including superconducting qubits, trapped ions, and photonics.

Current noisy intermediate scale quantum (NISQ) [85, 283] devices contain order 50 qubits and give access to hundreds of quantum gate operations [86]. The term NISQ device indicates that we are at the transition point where the number of qubits reaches intermediate scale  $> 50$ , which is beyond brute force simulation but these devices are noisy and hence have restricted ability. The term NISQ era is also used to contrast the longer-term goal of reaching fault-tolerant quantum computing (FTQC). By quantum error correction (QEC), which trades a large number of noisy physics qubits with a few logical qubits, we can have FTQC with the cost of the overhead given by the QEC [285, 286].

The idea of a universal gate set for quantum computation leads to the distinction between different computation complexity classes. For different universal gate sets, we can simulate one from each other with an error  $\varepsilon$  using  $\text{POLYLOG}(1/\varepsilon)$  number of gates according to the Solovay-Kitaev theorem [287]. This means an easy quantum circuit for one universal gate

set is easy for all universal gate sets. Therefore, we can assign the state or the unitary transformation a “complexity” based on the number of gates required. We discuss this in further detail in Sec. 6.3.

## 6.2. IsoTNSs as sequential circuits

Remarkably, there exists a connection between isoTNSs and quantum circuits, which has been recognized since the early stage of development of tensor network theory [169, 170, 288]. By viewing isometries as unitaries acting on an ancilla qubit, an matrix-product state (MPS) in isometric form is equivalent to a sequential quantum circuit. The equivalence can be easily generalized to isoTNSs in higher dimensions and was recently applied in several works [92, 213, 289–291]. The resulting sequential circuits consist of multi-qubit gates. By approximately decomposing the multi-qubit unitaries into a sequence of 2-site gates, we obtain the “sparse” sequential circuit ansätze, which is also known as quantum circuit tensor networks (QCTNs) [212]. We study the one-dimensional (1D) finite version of the ansatz in Chapter 7 and the 1D infinite version of the ansatz in Chapter 8. Below, we review the equivalence of isoTNSs and sequential circuits using MPSs as an example and discuss the approximate decomposition of  $n$ -site gates.

### IsoTNSs as “dense” sequential circuits

An MPS of bond dimension  $\chi$  is equivalent to a sequential quantum circuit with  $(n+1)$ -site unitaries, where  $n = \log_2 \chi$ <sup>1</sup>. An MPS in right isometric (normalized) form is given as,

$$|\psi\rangle = \sum_{\{i_k\}} \sum_{\{\alpha_l\}} B_{\alpha_0 \alpha_1}^{[1]i_1} B_{\alpha_1 \alpha_2}^{[2]i_2} \cdots B_{\alpha_{N-1} \alpha_N}^{[N]i_N} |i_1 i_2 i_3 \cdots i_N\rangle \quad (6.1)$$

where  $\{i\}$  are indices representing physical degrees of freedom and  $\{\alpha\}$  are virtual indices, which encode entanglement. The rank of the virtual indices is proportional to the size of the gates in the quantum circuit representation, as we show below. The right orthogonality condition

$$\sum_{i_k, \alpha_k} B_{\alpha_{k-1} \alpha_k}^{[k]i_k} (B_{\alpha'_{k-1} \alpha_k}^{[k]i_k})^* = \delta_{\alpha_{k-1}, \alpha'_{k-1}} \quad (6.2)$$

indicates that each individual tensor  $B^{[k]}$  is an isometry mapping between the vector spaces  $V_{|\alpha_{k-1}\rangle} \rightarrow V_{|\alpha_k, i_k\rangle}$ . Any isometry can always be rewritten as slices of unitary. Therefore, it is equivalent to unitary acting on a normalized state  $|0_k\rangle$ <sup>2</sup>, i.e.,

$$B^{[k]} = U^{[k]}|0_k\rangle \quad (6.3)$$

$$B_{\alpha_{k-1} \alpha_k}^{[k]i_k} = \langle \alpha_k, i_k | U^{[k]} |0_k, \alpha_{k-1}\rangle \quad (6.4)$$

One can easily verify the equivalence by substituting Eq. 6.4 into Eq. 6.2. Note that the equivalence holds for generic isometric tensors and hence the discussion also holds for higher dimensional isoTNSs.

Once the connection between the isometries  $B^{[k]}$  and unitaries  $U^{[k]}$  acting on a state  $|0_k\rangle$  is established, we can rewrite the right normalized MPS as a quantum circuit with a set of corresponding gates  $\{U^{[k]}\}$  acting on the initial state  $|0\rangle^{\otimes N}$  (see Fig. 6.1). As expected, the dimension of the final state is the same as the initial state, because the virtual indices  $\{\alpha\}$  are internally contracted. This is the formal equivalence of an isometric MPS and a sequential quantum circuit.

<sup>1</sup>We note that the mapping applies to isoTNSs [75] in general.

<sup>2</sup>The state  $|0_k\rangle$  lives in the vector space of dimension  $\dim(|0_k\rangle) = \chi_k \times \dim(|i_k\rangle) / \chi_{k-1}$ . We assume  $\dim(|0_k\rangle)$  to be an integer without loss of generality since we can always enlarge the bond dimension to match this condition.



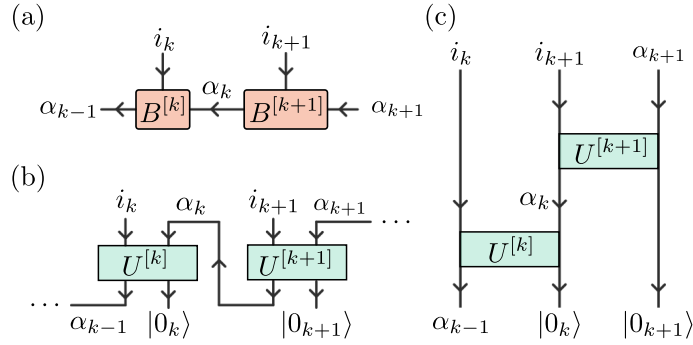


Figure 6.1.: (a) Two tensors in right orthogonal form. (b) A right orthogonal MPS can be directly mapped to a quantum circuit. (c) The corresponding quantum circuit, where the gates act sequentially from the first to the last qubit.

### From “dense” sequential circuits to “sparse” sequential circuits

For spin-1/2 systems, the physical dimension is  $d = 2$  and the  $|0_k\rangle$  gives the standard qubit. If the MPS consists of tensors  $B^{[k]}$  with bond dimension  $\chi = \dim(\alpha_{k-1}) = \dim(\alpha_k) = 2^n$ , where  $n \in \mathbb{N}$ , then the corresponding unitaries act on  $(n + 1)$  qubits. As a result, an MPS with maximum bond dimension  $\chi$  is equivalent to a quantum circuit defined by unitaries acting on maximally  $\log_2 \chi + 1$  sites sequentially.

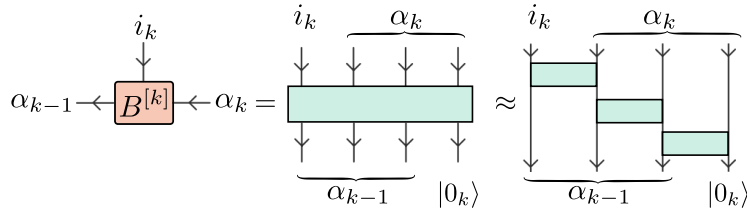


Figure 6.2.: An MPS tensor can be exactly represented as a unitary over some number of qubits, which can then be approximated as a series of 2-qubit gates.

The  $n$ -site unitary contains  $\sim \chi^2$  number of parameters, which must be decomposed into a series of sequential 2-site unitaries to make the quantum circuit implementable on a real device. In the worst case, the number of required 2-site unitaries scales at most exponential in  $n$  [292]. The number of 2-site unitaries required is problem dependent and gives us insight into the potential quantum advantage (See Sec. 6.3). Here, we consider approximating the  $n$ -site unitary by a local circuit of 2-qubit gates and treat the resulting “sparse” sequential circuits as variational ansätze. The structure of the local circuit can be of arbitrary types, for example, brickwall or staircase (ladder)<sup>3</sup>, and the depth of the local circuit is a tuning parameter of the complexity of the variational ansätze. In the following chapters, we investigate the power of such ansätze for simulating dynamics. The central question we want to investigate is: *In order to reach similar accuracy in representing the target states, how does the depth of the local circuit scale?* We will show that for the simulation of quantum dynamics we can reach similar accuracy by using shallow local circuits, which have parameters scaling polynomial in  $n$ , to parameterize the  $n$ -site unitaries. This means we use exponentially fewer parameters compared to the generic case and hence have the potential of exponential quantum advantage (see discussion below).

<sup>3</sup>We note that there are different possible circuit structures. Numerical efforts have been carried out to study the effect of different circuit structures and the scaling for ground state problems [212]

In chapter 7, we consider the approximation of the  $n$ -qubit unitary by a single layer of sequential unitaries as shown in Fig. 6.2. We call this finite ansatz of order- $M$  where  $M + 1 = n$  is the number of qubits the unitary acts upon. For example, an MPS of bond dimension  $\chi = 2$  maps exactly to our circuit ansatz of order-1. Note that this is particular to our ansatz; the commonly studied brickwall circuit structure with two layers, which has the same number of two-site gates, can be mapped to an MPS of bond dimension  $\chi = 2$  but cannot represent states having correlations of arbitrary length scales, for example, the GHZ state. Our order- $M$  circuit ansatz permits a sparse representation of MPS of bond dimension  $2^M$ . The sparsity of the representation comes from replacing the  $(M + 1)$ -site unitary with a sequence of 2-site unitaries, see Fig. 6.2. Repeating such replacement, one arrives at a circuit with the pattern as in Fig. 7.1 (a).

In chapter 8, we further generalize the ansatz by considering approximating the  $n$ -qubit unitary by  $M_U$  layers of sequential unitaries and furthermore imposing the translational invariance over the quantum circuit. We observe a similar exponential reduction in the number of parameters required to represent the  $n$ -qubit unitaries.

## Entanglement and classical complexity

Before delving into the quantum complexity of tensor-network-inspired quantum circuits, it is important to first review the computational complexity of classical tensor network simulations. Brute force classical simulation of quantum systems has computational complexity scaling exponentially in the number of qubits or system size. As we have seen in earlier chapters of this thesis, tensor network algorithms provide powerful simulation tools, which avoid this exponential scaling in the system size. Instead, the computational complexity of tensor network algorithms is restricted by and scales exponentially with the entanglement entropy. Roughly speaking, the bond dimensions of tensor networks scale  $\sim e^S$  with the entanglement entropy  $S$ , and the computational complexity is polynomial in the bond dimensions. Therefore, the computational complexity scales exponentially  $\sim e^S$  in the entanglement entropy. This means that  $\sim e^S$  elementary operations on a classical computer are required to evaluate the answer, for example, the expectation value of some physical observables. The value of the entanglement entropy  $S$  depends on the problem at hand. Prominent examples include an area-law entanglement for the ground states of gapped local Hamiltonians, which is  $\sim 1$  for 1D systems. For simulating the dynamics of many-body localized systems, the entanglement growth is logarithmic [293], i.e.,  $\sim \log t$ , while for simulating the dynamics of chaotic systems, the entanglement growth is linear [56], i.e.,  $\sim t$ .

## 6.3. Quantum complexity

Suppose that we have a quantum computer, a natural and important question to consider is: how difficult is it to solve a particular problem using this computer? As discussed in the previous section, the concept of *quantum complexity* can be defined for a quantum computer equipped with a universal gate set. It is a measure of the number of computation steps required to carry out to solve the problem. In this context, it is useful to recall two important concepts from quantum information theory: the *state complexity* and the *unitary complexity*.

Broadly speaking, the **state complexity** of a state  $|\Psi\rangle$  is the minimum size of a quantum circuit over some universal gate set required to generate a state  $|\tilde{\Psi}\rangle$  from  $|0\rangle^{\otimes m}$ , where the error  $\| |\tilde{\Psi}\rangle - |\Psi\rangle \|^2 < \varepsilon$  is within some tolerance  $\varepsilon$ . Similarly, the **unitary complexity** of a unitary  $U$  is the minimum size of a quantum circuit over some universal gate set required

to generate a unitary  $\tilde{U}$  such that  $\|U - \tilde{U}\|_\diamond < \varepsilon$  for some error  $\varepsilon$ <sup>4</sup>.

### Potential quantum advantage

With the concepts of quantum complexity and entanglement in mind, we can estimate the classical and quantum complexity to simulate 1D quantum many-body systems specifically for MPSs and the corresponding sequential circuits. We consider the setup where the problem is defined with the system size  $L$ , the evolved time  $t$ , and an error tolerance  $\varepsilon$ . The classical MPSs simulation with bond dimension  $\chi$  has a number of parameters<sup>5</sup>  $\sim L\chi^2$  and the complexity<sup>6</sup>  $\sim L\chi^3$ . For the sparse sequential circuits introduced above, the number of parameters is given as  $\sim LC$ , where  $C$  is the number of two-site gates within the local circuit parameterizing a  $n$ -site unitary<sup>7</sup>. Since  $C$  is the quantum complexity of the unitary as shown in Fig. 6.2, the overall quantum complexity of the sparse sequential circuits scales the same as the number of parameters  $\sim LC$ . For sequential circuits, the number of parameters is proportional to the computational complexity on a quantum device, we can utilize this fact and understand the complexity by studying the required number of parameters.

For simulating quantum dynamics, we consider the following setup. We fix a large enough system size  $L$  and a target accuracy set by the error tolerance  $\varepsilon$ . We vary the evolved time  $t$  and study the classical and quantum complexity of the variational ansätze introduced. Precisely, we study how the required bond dimension  $\chi$  to accurately (within error  $\varepsilon$ ) represent the time-evolved states grows with  $t$ . Similarly, we study how the required  $C$  of the sequential quantum circuits grows with  $t$ . By this setup, the observation would give us a hint on the number of parameters required and the computational complexity of the classical and quantum algorithms. If we observe  $C \sim \log(\chi)$  asymptotically with increasing  $t$ , this indicates a potential *exponential quantum advantage*. On the other hand, if we observe  $C \sim \chi^p$  asymptotically with increasing  $t$ , this indicates the potential of *no quantum advantage* or *polynomial quantum advantage*. Note that we provide neither exclusive search on all possible variational ansätze nor on all possible algorithms. Therefore, in both cases, the observed result will merely indicate the (non)existence of a *potential* quantum advantage.

As mentioned earlier, a linear entanglement growth is expected for simulating the global quench dynamics of chaotic systems [55, 56], which translates into an exponential growth of the bond dimension  $\chi \sim e^t$ . On the other hand, nature simulates its own evolution and intuitively a complexity scaling  $C \sim t$  for the quantum simulation is expected, which is indeed proven analytically using product formula [295]. As a result, the quantum simulation for dynamics with the Lie-Trotter product formula is expected to provide an *exponential quantum advantage*, if no classical algorithm can be more efficient than  $e^t$ . In Chapter 7 and Chapter 8, we confirm the linear scaling in  $C$  in the context of variational optimization and treating the sparse sequential circuits as variational ansätze.

It is noteworthy that the situation differs for the ground state problems. Recent works have been carried out for similar analyses for studying ground state problems. It is shown that if we fix the system size  $L$  and study the accuracy of the ground state representation in terms of the energy error  $\varepsilon$  by variational optimization, the  $\varepsilon \sim \text{POLY}(1/D)$  and  $\varepsilon \sim \text{POLY}(1/C)$  for MPSs and sparse sequential circuits [212]. This suggests potentially a polynomial quantum advantage. A separate study [97] compares classical variational algo-

<sup>4</sup>See [294], for more rigorous definition.

<sup>5</sup>We assume a fixed and small physical dimension  $d \ll \chi$ .

<sup>6</sup>Both the density matrix renormalization group (DMRG) and time-evolving block decimation (TEBD) algorithm has this scaling. In general, variational algorithms for MPSs have similar scaling.

<sup>7</sup>We ignore the constant prefactor from the number of parameters required for parameterizing a two-site gate.

rithms in evaluating the ground state energy  $E$  with the quantum phase estimate (QPE) algorithm, which has a complexity  $\text{POLY}(L)\text{POLY}(1/\varepsilon)\text{POLY}(1/S)$  in determining  $E$  on a fault-tolerant device, where the  $S$  is the overlap between the initial state used for QPE and the actual ground state. Numerically, it is observed that, as one of the classical variational algorithms, tensor network methods have the empirical cost  $\text{POLY}(L)\text{POLY}(1/\varepsilon)$ <sup>8</sup> for 1D, two-dimensional (2D), and three-dimensional (3D) paradigmatic models. Thus, this raises questions about the existence of generic exponential quantum advantage for ground-state problems but does not rule out the possibility of polynomial speed-up.

---

<sup>8</sup>This comes from the empirical observation that the error density goes down as  $\varepsilon/L \sim 1/D$  and a constant energy density  $\varepsilon/L$  holds for fixed  $D$  in 3D Heisenberg model for  $L$  up to 1000 sites. Since the tensor network algorithms have the complexity  $\text{POLY}(L)\text{POLY}(D)$ , equivalently we have the complexity  $\text{POLY}(L)\text{POLY}(1/\varepsilon)$

# 7. Real- and imaginary-time evolution with compressed quantum circuits

Experimental advances in quantum computation technology have raised several fundamental questions about the relationship between the complexity and entanglement of physically relevant quantum states. In classical algorithms, especially tensor network methods, the entanglement is a good proxy for the difficulty of representing a state. For a quantum circuit, however, these measures are relatively independent; one can have states with high entanglement but low complexity. This distinction between complexity and entanglement means we can roughly divide the states in the Hilbert space into three categories (i) The low-entangled and low-complexity states can be efficiently simulated on both classical and quantum devices, for example, the ground states of local gapped Hamiltonians. (ii) The highly-entangled and high-complexity states occupy the majority of the full Hilbert space. Such states are not physical because they can only be produced after an exponentially long time [296]. (iii) The remaining highly-entangled and low-complexity states can be simulated efficiently on quantum but not classical devices. The identification of these classes of states is a major problem in quantum complexity and is of importance for understanding quantum advantage [86].

In this work, we focus on the class of states generated under time evolution (which is known to fall in the third category) and study a class of quantum circuits motivated by the representation of matrix-product states. For a given amount of entanglement, we see the quantum circuits require exponentially fewer parameters than the matrix-product states, which agrees with the picture we described. While these states mark the limit of current classical numerical methods, quantum simulators and quantum computers may allow us to study these physically interesting states. Moreover, we propose a variational quantum algorithm for time evolution.

means we can roughly divide the states in the Hilbert space into three categories (i) The

The structure of this chapter is as follows. We demonstrate that the ansatz states are a good approximation for the states obtained during time evolution in Section 7.1 by comparing them with classical numerics. In Section 7.2, we consider a variational time evolution algorithm for general quantum circuits for both real and imaginary time evolution. In Section 7.2.3, we classically optimize the gates, then implement the compression directly on a quantum processing unit (QPU). We conclude, in Section 7.3, by noting several future avenues of exploration using the techniques developed in this work.

## 7.1. Compressed circuits

Although entanglement is a good proxy for the difficulty of representing a state using an matrix-product state (MPS) ansatz, the light cone determined by a time evolution under a local Hamiltonian enforces a particularly simple entanglement pattern that can in principle be captured with fewer parameters. We use an ansatz where sequential quantum circuits represent our states. These circuits consist of a set of two-qubit gates  $\{U_i\}$  that are applied sequentially as shown in Fig. 7.1(a). The circuit is said to be of order  $M$  when there are  $M$  “layers” of gates. The total depth of this circuit is  $2(M - 1) + N - 1$ , which scales linearly in both the system size  $N$  and with the order  $M$ . We emphasize that the algorithms

we present in this work are independent of the choice of ansatz, though different ansätze may describe different classes of states. For instance, in contrast to the more commonly studied brickwall/checkerboard circuit structure [289, 297], the sequential ansatz can have correlations over arbitrary length scales even at the lowest order  $M$ . This means that an order  $M = 1$  sequential ansatz can represent states like a GHZ state, which is not possible for a brickwall/checkerboard type of circuit with the same number of gates. For this reason, we studied the sequential ansatz. For the definition and comparison with the brickwall circuit ansatz, see Appendix D.1.

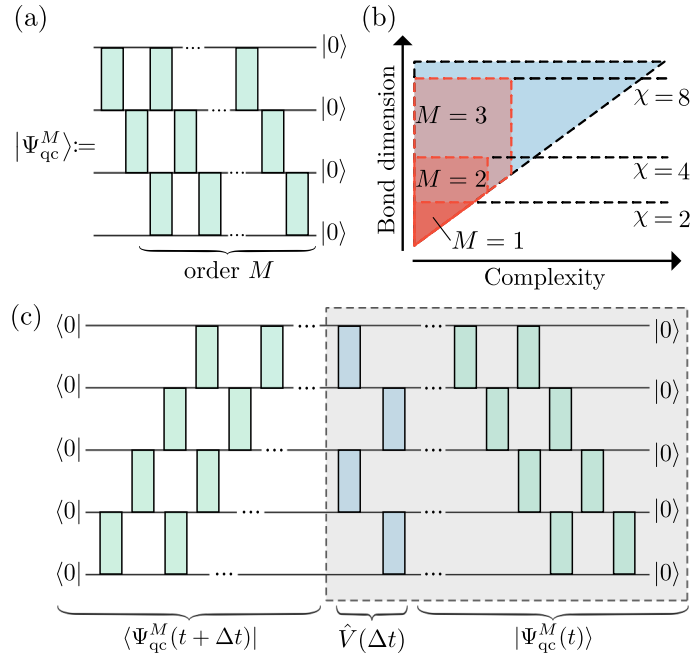


Figure 7.1.: (a) We parameterize an order  $M$  variational ansatz with  $M$  layers of gates. (b) We show the region of the Hilbert space accessible by our ansatz, which corresponds to states with high entanglement but (by definition) low complexity. Each circuit order spans a sub-manifold of a larger MPS manifold with bond dimension  $\chi$ . (c) To perform time evolution, we prepare a state  $|\Psi_{\text{qc}}^M(t)\rangle$ , then apply a Trotterized time evolution to obtain  $|\Psi_{\text{qc}}^M(t + \Delta t)\rangle$ . By variationally optimizing each of the gates, we find an optimal representation of the time-evolved state within the sub-manifold defined by our variational ansatz.

We note that the states defined by these quantum circuits form a sub-manifold of MPSs with bond dimension  $\chi = 2^M$ . In the case of  $M = 1$ , the quantum circuit is exactly equivalent to an MPS of bond dimension  $\chi = 2$  (see Chapter 6, Sec. 6.2). However, for  $M > 1$ , these quantum circuits have exponentially fewer parameters than a generic matrix-product state in canonical form with bond dimension  $2^M$ . In other words, these quantum circuits describe states with high entanglement but low complexity, which—as we demonstrate below—encompass time-evolved states. Note that this reduction of parameters does not necessarily translate into a sparse representation when converted into MPS form. Furthermore, although we still store the parameters of our circuit classically, if we were to process the state classically (such as in the computation of an observable), this would still scale exponentially with the entanglement in the state.

To test this class of quantum circuit ansatz, we first consider the far-from-equilibrium dynamics of a global quantum quench. Crucially, such dynamics are typically accompanied by fast ballistic growth of entanglement, which puts mid-to-long time dynamics out of

reach for numerics beyond small systems. Concretely, we consider dynamics under the Hamiltonian

$$\hat{H} = -J \left[ \sum_{j=1}^{N-1} \hat{\sigma}_j^x \hat{\sigma}_{j+1}^x + \sum_{j=1}^N g \hat{\sigma}_j^z + \sum_{j=1}^N h \hat{\sigma}_j^x \right], \quad (7.1)$$

which is a quantum Ising spin chain on  $N$  sites with both transverse ( $g$ ) and longitudinal ( $h$ ) fields. For the special case  $h = 0$ , the model is integrable. We consider a global quantum quench protocol with polarized initial state  $|\Psi\rangle = |\cdots \uparrow \uparrow \uparrow \cdots\rangle$  at time  $t = 0$ . Our goal is then to accurately approximate the state  $|\Psi(t)\rangle = e^{-i\hat{H}t}|\Psi\rangle$ , at (real or imaginary) time  $t$  after the quantum quench.

### 7.1.1. Efficient representation of quantum states

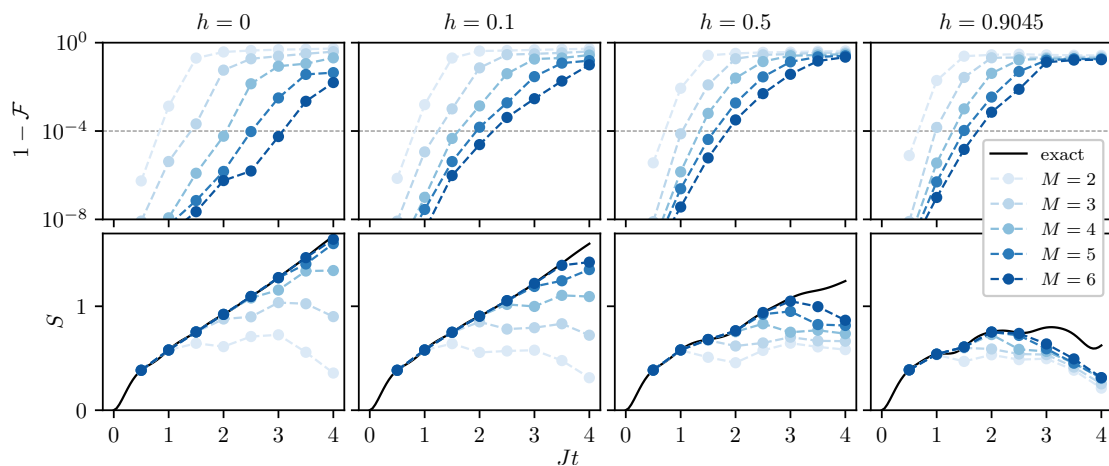


Figure 7.2.: Quantum circuit representation of order  $M$  of quantum states generated under non-equilibrium dynamics. The Hamiltonian is given in Eq. (7.1) for a chain of length  $N = 31$  with transverse field  $g = 1.4$  and data shown for  $h = 0, 0.1, 0.5, 0.9045$ . The top row shows the fidelity  $\mathcal{F}$  defined in Eq. (7.2), compared with MPS with bond dimension  $\chi = 1024$ . The bottom row shows the half-chain von Neumann entanglement entropy  $S$  for the quantum circuit.

We now demonstrate the representation power of the quantum circuit ansatz by comparing it with classical numerics using MPS. We first perform the time evolution using 4<sup>th</sup> order Trotterized time-evolving block decimation (TEBD) [31] for  $N = 31$  with maximum bond dimension  $\chi = 1024$  and step size  $\tau = 0.01$  to obtain a quasi-exact approximation of the state  $|\Psi(t)\rangle$ . This bond dimension ensures that our results are close to exact for all considered timescales. We then take the MPS at a selection of times, which we denote  $|\Psi_{\text{mps}}(t)\rangle$ , and find the optimal quantum circuit of order  $M$ , which we denote  $|\Psi_{\text{qc}}^M(t)\rangle$ . The state represented by the quantum circuit is implicitly parameterized by a set of two-qubit unitaries  $\{U_i(t)\}$ . We perform an optimization over the unitaries in our quantum circuit to find the state with maximum fidelity

$$\mathcal{F} = |\langle \Psi_{\text{qc}}^M(t) | \Psi_{\text{mps}}(t) \rangle|^2. \quad (7.2)$$

This is done iteratively by updating each  $U_i(t)$  using a polar decomposition [158] (see Appendix D.2 for more details).



In Fig. 7.2 we show the fidelity of the quantum state obtained from the quantum circuit ansatz as well as the half-chain von Neumann entanglement entropy,  $S$ . Data are shown for a range of values of the longitudinal field  $h = 0, 0.1, 0.5, 0.9045$ . The parameters  $g = 1.4, h = 0.9045$  are chosen such that the dynamics of the system are expected to be chaotic and hard to simulate due to fast scrambling [56, 298]. The accuracy of the approximation decreases with time as correlations build throughout the system, but improves as the order  $M$  is increased. For a given order  $M$ , this data also shows that the circuit more accurately captures the state for weaker  $h$ , indicating an increase in the complexity of the simulation for larger  $h$ .

Figure 7.2 also shows the growth of entanglement. We find that the ansatz easily captures the rapid ballistic growth of entanglement for small  $h$ . As we increase  $h$ , we find that the growth of entanglement slows down. This indicates that the practical complexity of the quantum states increases with  $h$  whereas the growth of entanglement decreases, thus providing a diminished but still exponential advantage.

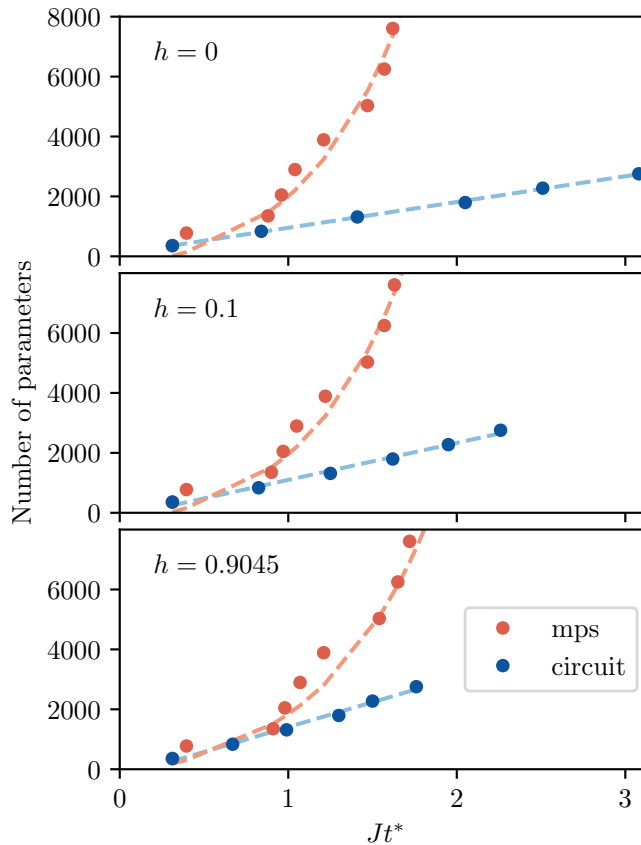


Figure 7.3.: Comparison of the number of parameters and the accessible time  $t^*$ . This time  $t^*$  corresponds to the time at which the fidelity  $\mathcal{F}$  drops below  $1 - 10^{-4}$ . Data are shown for MPS and our quantum circuit ansatz for three values of the longitudinal field  $h$ . The dashed lines correspond to exponential (MPS) and linear (circuit) fits, respectively. See Appendix D.5 for more details.

From this data, we can compare the number of parameters required to achieve a given accuracy using our quantum circuits with those needed for an MPS. For a given order  $M$  we find the time  $t^*$  up to which the fidelity is greater than  $\mathcal{F} = 1 - 10^{-4}$ , indicated by the grey dashed line in Fig. 7.2. In Fig. 7.3, we plot the number of parameters in the quantum circuits and the MPSs as a function of the reachable time  $t^*$ . This figure



shows that the number of parameters in our quantum circuit ansatz scales linearly with the reachable time  $t^*$ , in stark contrast to the exponential growth in parameters for the MPS. Note that the circuit depth of a fully Trotterized time-evolution also scales linearly with time [295, 299] and has for sufficiently small time steps an error for local observables that is independent of system size as well as simulation time [300]. However, we find that the compressed circuit generically performs better while the quantitative improvement over the fully Trotterized time-evolution depends on the model parameters—this reduction of circuit depth is particularly valuable for current noisy intermediate scale quantum (NISQ) devices on which Trotterized time evolution is very challenging [92].

We stress that the linear scaling of the number of parameters persists across the different values of  $h$ . The additional complexity for large values of  $h$  appears as a change in the gradient of the linear scaling. These results demonstrate that the complexity of the quantum state grows linearly in time, while the MPS ansatz requires a number of parameters that grow exponentially in time due to the linear growth of entanglement. For all values of  $h$  we can see that the quantum circuit has an exponential advantage over MPS in terms of the number of parameters required. Even for short times of  $\mathcal{O}(1)$  in the coupling  $J$ , we require fewer parameters to accurately represent the state with a quantum circuit than with an MPS.

## 7.2. Variational time evolution algorithm

Having confirmed the representation power of our ansatz, we now demonstrate how to implement time evolution restricted to the states defined by our ansatz. This, in turn, demonstrates that the optimization of the quantum circuit can be performed on a quantum device using hybrid quantum optimization algorithms. This potentially enables the simulation of dynamics beyond the reach of classical numerical methods, which are limited by the cost of storing the quantum state.

Our algorithm for time evolution is shown schematically in Fig. 7.1(c). Given the quantum circuit at time  $t$ , we apply a second-order Trotterized approximation  $\hat{V}(\Delta t) = e^{-i\hat{H}_{\text{even}}\Delta t/2}e^{-i\hat{H}_{\text{odd}}\Delta t}e^{-i\hat{H}_{\text{even}}\Delta t/2}$  to the time evolution operator  $e^{-i\hat{H}\Delta t}$ . We then find the state  $|\Psi_{\text{qc}}^M(t + \Delta t)\rangle$  that maximizes fidelity

$$\mathcal{F} = |\langle \Psi_{\text{qc}}^M(t + \Delta t) | \hat{V}(\Delta t) | \Psi_{\text{qc}}^M(t) \rangle|^2, \quad (7.3)$$

That is, we iteratively optimize over the set of 2-site unitary gates  $\{U_i(t + \Delta t)\}$  that define the state  $|\Psi_{\text{qc}}^M(t + \Delta t)\rangle$ . To carry out the optimization as a hybrid quantum-classical algorithm, one measures the fidelity on quantum devices and performs the optimization classically. The fidelity/overlap can be measured by the setup as in Fig. 7.1(c) or by the swap test [301, 302]. The optimizations for fidelity/overlap find direct application in targeting excited states and have been realized on quantum devices in recent works [303, 304]. Moreover, we expect the recent advance in new approaches for overlap measurement [305–307] and in optimization algorithms [308–313] could be applied effectively for our algorithm. We leave this for future work. In this work, we simulate the algorithm classically and perform the optimization similarly to that in the previous section, where we update each  $U_i$  iteratively using a polar decomposition.

### 7.2.1. Real time evolution

In Fig. 7.4 we show the local magnetization and the half-chain entanglement entropy simulated using our quantum time evolution algorithm. We consider the same quantum quench protocol as above with  $h = 0.1$ . Importantly, this case is non-integrable and has a fast linear growth of entanglement under the non-equilibrium dynamics.

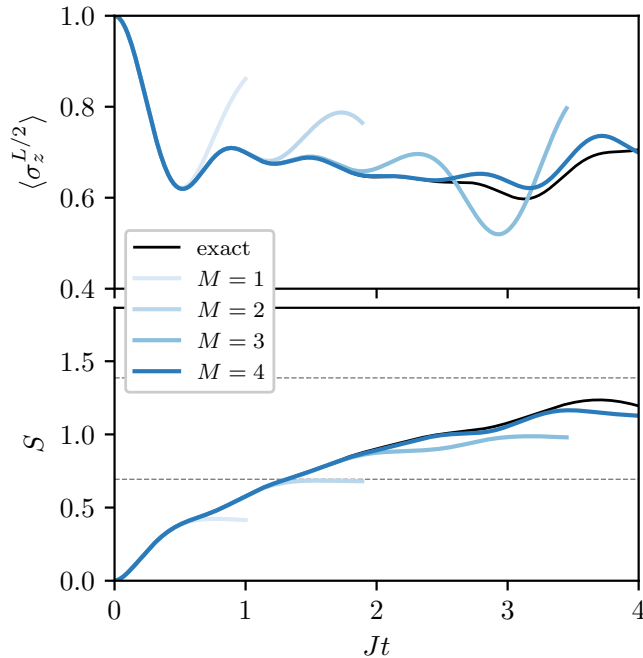


Figure 7.4.: Time evolution algorithm restricted to the quantum circuit ansatz for different orders  $M$ . We use  $g = 1.4$  and  $h = 0.1$ , with  $N = 11$  sites. In the top panel, we show the magnetization on the central sites, and in the bottom panel, we show the half-chain von Neumann entanglement entropy  $S$ . See Appendix D.2 for more details.

Our results show that we are able to accurately capture the magnetization for times that scale linearly with the order  $M$ . Here it is important to note that the time evolution is performed entirely within the sub-manifold of circuits defined by our ansatz with a fixed order. We additionally find that we are able to capture the linear growth of entanglement using these quantum circuits and that the saturation of the entanglement depends linearly on the order  $M$ . In contrast, the corresponding MPS representation has an exponentially large bond dimension requiring  $\mathcal{O}(2^M)$  parameters.

We emphasize that this time evolution algorithm is different from the time-dependent variational principle (TDVP) algorithm simulating time evolution with a quantum circuit proposed in [93, 94]. In those approaches, one solves the TDVP equations approximately by stochastic sampling, i.e., measurement, and performs finite time stepping by numerical integration<sup>1</sup>. In the present algorithm, we first perform finite time stepping by Trotterization and then try to find the optimal states within the sub-manifold defined by our ansatz. This is much closer to the time-dependent density matrix renormalization group (tDMRG) [50, 51] or TEBD [48, 49] algorithms, but also has similarities with the infinite TDVP-inspired algorithm proposed in [213]. This optimization algorithm for the time evolution is similar to one used for the multi-scale entanglement renormalization ansatz (MERA) [314] and in the context of symmetry-preserving ansätze [315].

The problem of efficiently optimizing a variational ansatz is one that is common to many current hybrid quantum-classical algorithms [89, 316]. The primary requirement to implement our algorithm is the capability to evaluate fidelities, at which point one can classically optimize the unitaries. This general scheme has been applied in various other cases – for example, in variational quantum eigensolvers [89], where the cost function being

<sup>1</sup>A similar argument applies to equations resulting from the Dirac-Frenkel variational principle and the McLachlan variational principle.

optimized is the expectation value of a Hamiltonian [89]. In this paper, we performed the optimization classically to avoid problems such as barren plateaux [317]. However, as previous studies have demonstrated the capability to compute fidelities [301, 302], this algorithm can be directly implemented as a hybrid quantum-classical algorithm.

### 7.2.2. Imaginary time evolution

We can also apply this time evolution algorithm to find ground states using imaginary-time evolution. In this section, we first explicitly show how one can embed the required non-unitary operators in unitary gates using an ancilla qubit and post-selection [72]. Second, we demonstrate that our ansatz can effectively converge to the ground state under imaginary-time evolution.

One can formally write down the exact imaginary-time evolution procedure  $|\text{GS}\rangle = \lim_{\tau \rightarrow \infty} e^{-\hat{H}\tau} |\psi_0\rangle$ , where  $\tau$  is real. This is equivalent to evolving in imaginary time ( $t \rightarrow -i\tau$ ) and corresponds to acting on the state with a non-unitary operator, which becomes a projector onto the ground state in the limit  $\tau \rightarrow \infty$ . We perform imaginary-time evolution analogously to our real-time evolution algorithm, where we sequentially compress the state back onto our ansatz as in Eq. (7.3) but now with  $\hat{V}(\Delta\tau) = e^{-\hat{H}\Delta\tau}$ . Similarly to real-time evolution,  $\hat{V}(\Delta\tau)$  can be approximated by a product of 2-qubit non-unitary gates using Trotterization.

To perform imaginary-time evolution, we are therefore required to implement non-unitary gates on the quantum computer. We achieve this by embedding the non-unitary gate in a unitary gate acting on one extra ancilla qubit. For a generic non-unitary operator  $A$  acting on  $N$  qubits, we define a unitary ( $N + 1$ )-unitary  $V_A$  by

$$V_A = \begin{pmatrix} sA & B \\ C & D \end{pmatrix}, \quad (7.4)$$

The strategy we employ is to find a block  $C$  and a scaling factor  $s$  that ensures the first  $2^N$  columns of  $V_A$  are mutually orthonormal, which guarantees unitarity. The remaining columns can be fixed using a QR decomposition. We explicitly show the full embedding procedure in Appendix D.3. Note that if we were to implement a full Trotter step for each optimization step, as we did previously for real-time evolution, we would require a linear number of ancilla qubits resulting in an exponential cost due to post-selection. Instead, one should apply and optimize the state for each 2-qubit gate in the Trotterized time evolution separately. In this case, the total number of measurements required across all Trotterized gates in a single time step scales only linearly with system size. Furthermore, the successful rate of post-selection is controlled by the step size  $\Delta\tau$ . In the small  $\Delta\tau$  limit, the failure rate is linear in  $\Delta\tau$ . This suggests the imaginary time evolution has complexity scaling linearly in both the system size and also the imaginary time evolved. See Appendix D.4.

To benchmark how well our ansatz can approximate the true ground state, we directly minimize the energy

$$E = \langle \Psi_{\text{qc}}^M | \hat{H} | \Psi_{\text{qc}}^M \rangle. \quad (7.5)$$

This procedure is similar to the variational quantum eigensolver (VQE) [89], where the parameters encoding the quantum state are iteratively adjusted to minimize the energy. We perform the procedure on a classical computer where it is intended to benchmark our imaginary-time evolution algorithm, where we consider the energy in Eq. (7.5) to be the best achievable by our chosen ansatz, shown as dashed lines in Fig. 7.5. Instead of using gradient descent methods, we iteratively replace the unitaries using polar decomposition, see appendix D.2.

In Fig 7.5, we show the results of our imaginary time evolution. These show that we can successfully converge to the optimal energy attainable with this ansatz. As expected, the

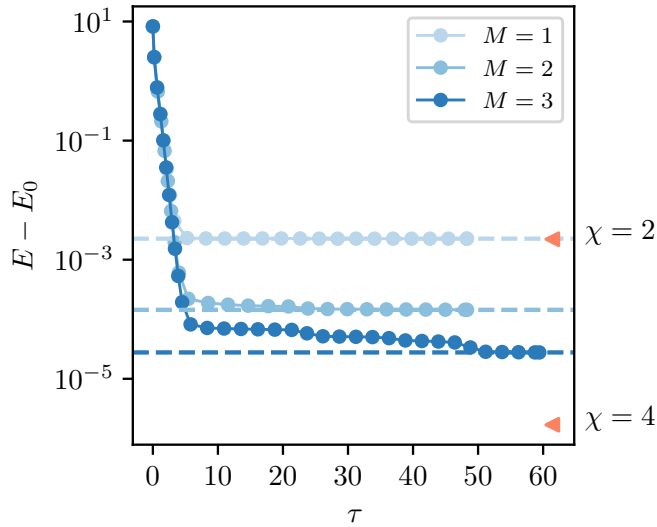


Figure 7.5.: We perform imaginary-time evolution for circuits of order  $M = 1, 2, 3$  for the quantum Ising model with  $N = 31$ ,  $g = 1.2$ ,  $h = 0.1$ . To find the optimal performance of our ansatz, we perform a procedure similar to VQE, where we iteratively optimize the expectation value of the Hamiltonian in Eq. (7.1) for our ansatz. For these depths, our imaginary-time evolution algorithm successfully converges to the optimal point, depicted by the dashed lines. The  $\chi = 4$  line indicates the ground state energy of an MPS with bond dimension 4 found using DMRG. To achieve a better accuracy, we successively decrease the time step  $\Delta\tau$  after achieving convergence for the previous step size.

results for  $M = 1$  match those from density matrix renormalization group (DMRG) with bond dimension  $\chi = 2$  due to the equivalence between the circuit and MPS representation. Note that while a modest MPS bond dimension  $\chi = 4$  performs better than our ansatz for  $M = 2, 3$ , we still achieve errors well below the threshold of current NISQ hardware, which validates this approach as a method for finding ground states on a quantum device.

We note that an alternative approach to imaginary-time evolution was taken in the QITE algorithm [318–320]. There it was noted that if enough information about the initial state is known, a non-unitary gate can be replaced by a unitary one without the use of ancillas. However, getting closer to the ground state requires state-dependent unitary operators with increasingly large support. In contrast, our algorithm requires a fixed set of local gates that can be repeatedly applied to reach later times, much like the TEBD algorithm for MPS [48]. While the approximation step is stochastic on a quantum computer, the overall procedure deterministically converges to the ground state. The choice of ansatz is also completely flexible. Viewing the procedure as a sequential compression in this way raises an interesting comparison with the compression of a tensor network to form a MERA and the emerging view of learning with tensor networks as a procedure of compression [128, 321].

### 7.2.3. Simulation on QPU

While our algorithms are designed for near-term quantum computers, the noise and coherence times of currently available devices place strong limits on what can be achieved. However, we are able to demonstrate parts of the algorithm on a quantum computer by delegating more of the algorithm to the classical computer. Here we classically optimize the time evolved states  $|\Psi_{\text{qc}}^M(t)\rangle$ , then construct and measure the corresponding state on

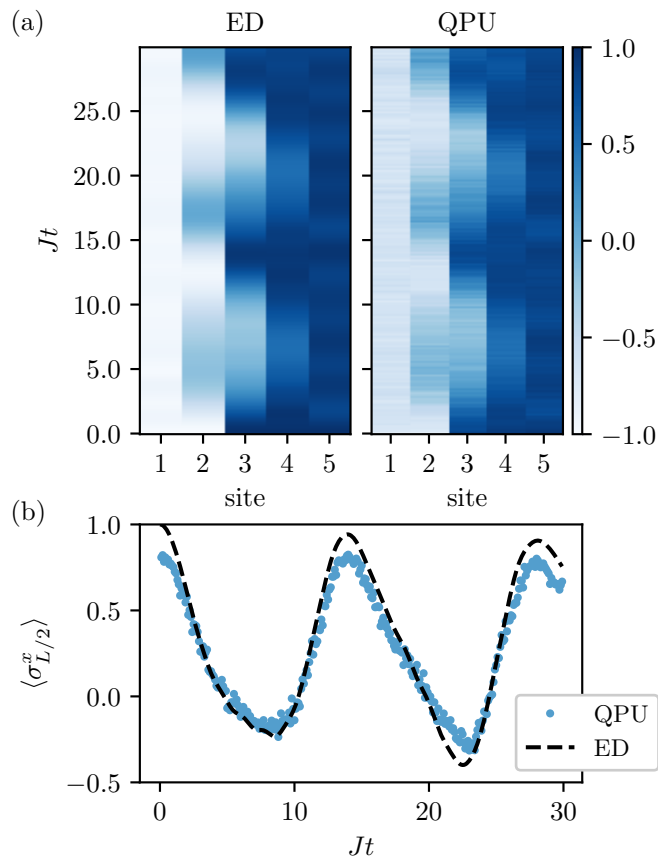


Figure 7.6.: We show the benchmark results for  $L = 5$ . Quenched dynamics from a product state with a single domain wall and Hamiltonian parameters  $g = 0.25, h = 0.2$ . (a) The  $\langle \sigma^x \rangle$  expectation values over the full system from ED simulation and measurements on the time evolved states prepared on QPU. (b) The  $\langle \sigma_{L/2}^x \rangle$  expectation value on the central qubit measured on the QPU and compared with ED. The data displayed is averaged over ten different circuit realizations (see Appendix D.6).

a QPU, namely the 5 qubit IBM-Q device codenamed Bogota [322]. This process allows us to access times on the QPU that are inaccessible using standard Trotterized evolution techniques.

Concretely, we consider the following quantum quench setup on  $N = 5$  qubits. We initialize the system in the product state  $| - - + + + \rangle$ , i.e., a domain wall in the  $x$ -basis, and evolve with the Hamiltonian (7.1) with  $g = 0.25, h = 0.2$ . For this range of parameters and initial state the dynamics are dominated by the motion of a single mobile domain wall and so can be well approximated by an order  $M = 1$  circuit. The longitudinal field,  $h$ , leads to a linearly confining potential between domain walls, and in the case of a single domain wall corresponds to a linear background potential leading to Wannier-Stark localization [323]. In Fig. 7.6(a), our ED results show the characteristic periodic melting and revival of the domain wall.

In Fig. 7.6 we show the results of constructing and measuring our compressed quantum state on the IBM QPU compared with ED results. Here we optimize the set of gates  $\{U_i(t)\}$  on a classical computer, which is then fed to the QPU to create the quantum state. The measurement of the magnetization in the  $x$ -basis closely matches the exact results. In particular, the spatial distribution of the magnetization (Fig. 7.6(a)) shows the periodic

spreading and reconstitution of the domain wall. Furthermore, the magnetization on the central spin, shown in Fig. 7.6(b) accurately and quantitatively matches the ED simulation for long times, which are not limited to the range we have considered. These timescales are currently inaccessible using a naive Trotterized evolution on this quantum device, which would require a circuit depth of  $\mathcal{O}(t)$ .

### 7.3. Discussion

In this paper, we have shown that physically relevant quantum states, namely ground states and those arising under non-equilibrium dynamics, can be efficiently represented using a sequential quantum circuit ansatz. This ansatz describes a “sparse” representation spanning a corner of the larger MPS manifold. For time evolution, the time scales that we can reach scale linearly with the number of parameters in the circuit, representing an exponential advantage over existing classical methods. This suggests that even within the class of MPS defined by a fixed bond dimension, there exists a range of physical states for which our ansatz is a more efficient representation than an MPS. To exploit the representation power, we used a time evolution algorithm for a general quantum circuit ansatz that can be implemented natively on existing quantum computers. Importantly, the quantum circuit ansatz is flexible and is not restricted to the one used in this paper. Using near-term devices may provide access to non-equilibrium dynamics beyond the reach of current classical algorithms. Finally, we have shown that this time evolution algorithm can also be applied in imaginary time to obtain ground states on a quantum computer.

The optimization procedure that we used [158]—fidelity maximization using a polar decomposition—may have other potential applications. For instance, instead of considering the compression of states, one can consider the compression of unitaries. This technique can be applied to approximate a multi-qubit unitary by a series of 2-qubit unitaries or to compress a deep quantum circuit. Both of these are particularly important for current NISQ devices.

Our procedure is also a potentially practical tool for studying quantum complexity. Quantum state complexity is an intriguing research field but is difficult to study numerically. Previous results primarily focus on non-interacting systems [324–326]. By using states acquired from procedures such as TEBD and DMRG, and approximating them using a chosen ansatz and polar decomposition methods, one can concretely probe the complexity of generic classes of states (such as quantum scar states and many-body localized states) that were previously difficult to analyze. Additionally, the separation between complexity and entanglement is of significant interest. In particular, Ref. [294] uses a random unitary circuit model for time evolution to demonstrate that even when the growth of entanglement saturates in a finite system, the complexity of the quantum states continues to grow linearly in time over far longer time-scales. This highlights a drastic distinction between maximally entangled states and maximally complex states. Our work shows that this separation appears to be smaller for non-integrable systems (see Fig 7.2). The techniques developed in this paper open the opportunity to directly study the separation between complexity and entanglement in concrete systems.

The algorithms studied in this work open up several intriguing generalizations. First, one could apply the algorithm to study short-time dynamics for higher dimensional systems, which are generally difficult problems for classical numerics. Applied directly on a quantum computer, this algorithm offers a tractable way to study higher dimensional systems at large system sizes and to probe physics that only manifests at higher dimensions. Moreover, the algorithms considered are agnostic to the specific ansatz used. It is an interesting question to compare how an ansatz with a different entanglement pattern performs. For instance, in Ref. [327] quantum circuits containing entangling gates acting

over the full system are considered and optimized to represent time-evolved states by a reinforcement learning approach, which is complementary to our approach. Additionally, quantum circuits inspired by matrix product states have shown promise for solving non-linear Schrödinger equations [328]. Similar analyses for various ansatz structures could shed light on the deeper relationship between entanglement and complexity.





## 8. Time evolution of uniform sequential circuits

In this Chapter, we extend the study of finite systems in Chapter 7 to infinite systems. Simulation of dynamics of infinite translationally invariant systems facilitates our understanding of the physics in the thermodynamic limit. However, the quantum advantage and the complexity scaling are subtler since direct applying the algorithms for finite system case [295, 329] is not possible, due to the linear scaling in the system size and the requirement of an infinite large quantum computer. Recent works have proposed and shown that it is possible to perform the simulation with a finite number of qubits [213, 214] for models with small entanglement growth. It is a crucial question now on how to improve the scalability and stability of such an algorithm.

In this context, we present a modified hybrid quantum-classical algorithm for time evolving translation invariant infinite systems in one-dimensional (1D). We propose layered uniform sequential circuits (l-USC) ansatz, which is a generalization to the single-layer uniform sequential circuits (USC) ansatz introduced in Ref. [213]. The ansatz l-USC forms a subclass of dense uniform sequential circuits (d-USC), which are equivalent to uniform matrix-product states (MPSs) [169, 288]. Moreover, we propose a gradient-based algorithm for time evolving quantum states within the manifold spanned by l-USC. This includes a routine for computing the transfer matrix and environments of the uniform states that does not require tomography or post-selection that would lead to exponential scaling.

To benchmark the proposed algorithm, we simulate it on a classical computer. We show that the number of variational parameters required to accurately time-evolve a quantum state for the time  $t$  with the l-USC ansatz, scales only polynomially in  $t$ . Lastly, having obtained the time-evolved l-USC state representation on a classical computer, we compute physical observables on a cloud-based quantum processing unit (QPU) and demonstrate agreement with quasi-exact results obtained with the infinite time evolving block decimation (iTEBD) algorithm at a large bond dimension [191].

This chapter is organized as follows. In Section 8.1, we introduce the layered uniform sequential circuit ansatz and the gradient-based variational time-evolution algorithm. In Section 8.2 we present the simulation results and analyze the effect of the layered decomposition on the accuracy of the time-evolved quantum state representation and fixed points of the transfer matrix. In addition, we show physical observables obtained from the classically optimized circuits, measured on real quantum hardware. In Section 8.3 we discuss the obtained data and outline the prospects for future work.

### 8.1. Methodology

In this section, we introduce the l-USC ansatz, which is a subclass of d-USC where the dense unitary is replaced by the layered decomposition. We present the necessary entities, e.g., transfer matrices and the environments, to measure the physical observables with l-USC. We then turn to the algorithm for time evolving the l-USC and the routines to perform the variational time evolution.

### 8.1.1. The layered uniform sequential circuit ansatz

Motivated by the sequential quantum circuit representation of the MPS [169, 288], we consider the d-USC on an infinite chain, as shown in Fig. 8.1 (a). The resulting wave function encoded by the d-USC reads

$$|\psi_R\rangle = \prod_{i=-\infty}^{+\infty} \hat{U}_R^i(\boldsymbol{\theta})|0\rangle, \quad |\psi_L\rangle = \prod_{i=+\infty}^{-\infty} \hat{U}_L^i(\boldsymbol{\theta}')|0\rangle, \quad (8.1)$$

where  $\hat{U}_R^i(\boldsymbol{\theta})$  and  $\hat{U}_L^i(\boldsymbol{\theta}')$  are  $i$ -independent unitaries acting on  $N_q$  consecutive qubits  $i, i+1, \dots, i+N_q-1$ . The ‘R’ index denotes the *right representation* and similarly the *left representation* ‘L’ is defined by a different unitary  $\hat{U}_L(\boldsymbol{\theta}')$  acting in the opposite order. The d-USC ansatz in left and right representations over  $N_q$  qubits are MPS in left and right isometric forms with the bond dimension  $\chi = 2^{N_q-1}$  respectively.

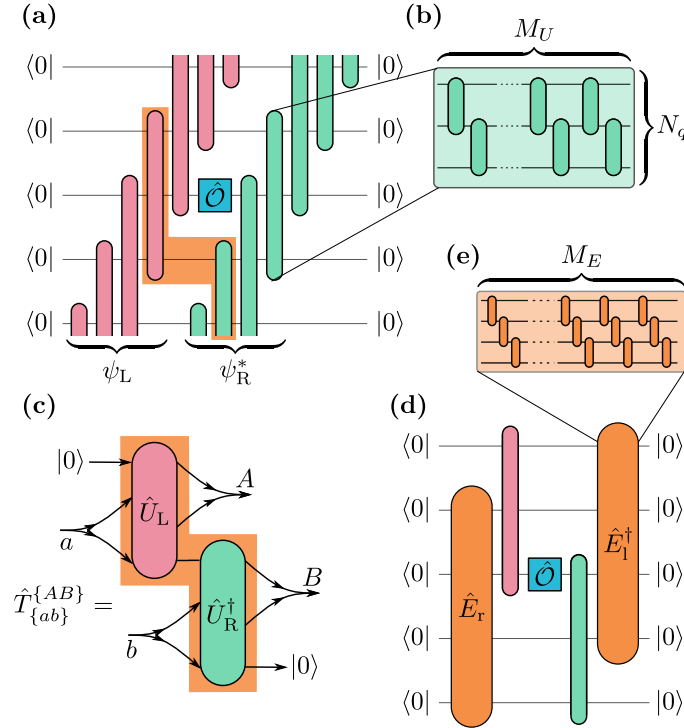


Figure 8.1.: (a) Circuit representing  $\langle\psi_R(\boldsymbol{\theta})|\hat{O}|\psi_L(\boldsymbol{\theta}')\rangle$  with  $|\psi_L(\boldsymbol{\theta}')\rangle$  and  $|\psi_R(\boldsymbol{\theta})\rangle$  being the same state in left and right representations. The shaded region singles out the repeated circuit element, i.e., transfer matrix. (b) Decomposition of a state-unitary into  $M_U$  layers of sequential 2-qubit gates. (c) Transfer matrix  $\hat{T}_{\{ab\}}^{\{AB\}}(\boldsymbol{\theta}, \boldsymbol{\theta}')$  between the left and right representations  $|\psi_L(\boldsymbol{\theta}')\rangle$  and  $|\psi_R(\boldsymbol{\theta})\rangle$ . (d) Circuit representation of  $\langle l, 0|\hat{U}_R^\dagger\hat{O}\hat{U}_L|0, r\rangle$  on finite number of qubits (e) Decomposition of an environment-unitary into  $M_E$  layers of sequential 2-qubit gates.

The l-USC ansatz is defined as a specific form of Eq. (8.1), where each unitary  $\hat{U}_{R/L}$  is parameterized by a sequential circuit of  $M_U$  layers, as shown in Fig. 8.1 (b). Each layer consists of a consecutive application of 2-qubit gates between neighboring qubits in the direction shown in Fig. 8.1 (a-b). While for any d-USC state in the right representation, there exists an exact d-USC of the same size in the left representation, this does not always hold for l-USC with the same  $N_q$  and  $M_U$  unless the state is inversion symmetric.

We note that l-USC ansatz belongs to the broad class of quantum circuit tensor network ansätze [212]<sup>1</sup>, where the dense unitaries in the isometric tensor networks are replaced by various kinds of local circuits, e. g., brick-wall circuits or sequential circuits (see discussion in Chapter 6, Sec. 6.2). The d-USC and l-USC wave functions are both universal: if one allows arbitrary  $N_q, M_U$ , all translationally-invariant quantum many-body states can be represented either d-USC or l-USC form. The required  $N_q, M_U$  indicates the complexity of the quantum many-body state. For example, the Greenberger–Horne–Zeinger (GHZ) state [330] can be represented with  $N_q = 2, M_U = 1$ . Rigorous studies of the scaling properties of the quantum circuit ansatz could give us more insight into the ansatz, for example, the recent work for ground states [331]. In this work, we focus on studying the expressivity of the ansatz applied to the time evolution with a chaotic Hamiltonian.

The l-USC with local circuits acting on  $N_q$  qubits defines a subclass of states within the manifold of d-USC, or equivalently uniform MPS of bond dimension  $\chi = 2^{N_q-1}$ . As a generic 2-qubit gate, up to a global phase, requires 15 parameters [332], the l-USC ansatz is parametrized by at most  $15(N_q - 1)M_U$  optimization parameters<sup>2</sup>, as compared to  $2^{2N_q+1}$  parameters necessary for the dense parametrization in the d-USC ansatz. In the previous works, it has been shown that similar ansätze on a finite system are polynomially more efficient in representing ground states [212] and exponentially more efficient in representing time-evolved states [4]. Previous works studying the dynamics of infinite systems have been focused on the specific case  $N_q = 2, M_U = 1$ . The question remains on whether the same conclusion, i.e., exponential advantage, applied to the generic l-USC, hold in the thermodynamic limit. The question remains on whether the same conclusions, applied to the l-USC, hold in the thermodynamic limit. Later in Section 8.2, we will demonstrate that l-USC forms a physically relevant subset of the d-USC states with the corresponding bond dimension, and allows for efficient time-evolution of quantum states. We begin here with the description of the tools to acquire physical observables from the l-USC state representation.

### Transfer matrix

Computation of physical observables and other operations of an infinite system can be performed on a finite number of qubits using the transfer matrix and its dominant eigenvectors, known as *environments* in the context of tensor networks. Utilizing the left and right representation of l-USC, we always consider the (mixed) transfer matrix defined between the states in the left representation,  $|\psi_L\rangle$ , and in the right representation,  $|\psi_R\rangle$ , as shown as the shaded area in Fig. 8.1 (a). In Fig. 8.1 (c), we explicitly write down the transfer matrix  $\hat{T}_{\{ab\}}^{\{AB\}}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ , with  $\{AB\}$  forming a united out-index and  $\{ab\}$  forming a united in-index. The arrow directions indicate the flow of time of the quantum circuit execution.

With this construction, the transfer matrix is a linear operator  $T : V^{ab} \rightarrow V^{AB}$  mapping a pure state in Hilbert space  $V^{ab}$  to a pure state in Hilbert space  $V^{AB}$ . The linear map is realized by a combination of unitary operators with the post-selection on one qubit, as shown in Fig. 8.1 (c). The transfer matrix is therefore generally non-Hermitian and non-unitary. In Appendix E.1.4, we show that the post-selection probability is close to unity for cases considered in this work. This formalism comes from the construction of the transfer matrix using simultaneously left and right representations. This is different from Ref. [213, 214], where the transfer matrix is defined with the inner product of states in the same representation, and the transfer matrix is a quantum channel mapping between

<sup>1</sup>Following the naming scheme in [212], our ansatz is the uniform qMPS-L. The direction of the application of layered 2-site gates is, however, the opposite.

<sup>2</sup>We note that the number of parameters can be reduced considering the redundancy of the consecutive single-qubit gates.

density matrices.

The left and right environments  $|l\rangle$  and  $|r\rangle$  are the dominant eigenvectors of the transfer matrix  $\hat{T}$  satisfying the fixed point equations  $\hat{T}|r\rangle = \lambda|r\rangle$ ,  $\hat{T}^\dagger|l\rangle = \lambda^*|l\rangle$ , where  $\lambda$  is the eigenvalue of  $\hat{T}$  with the maximum absolute magnitude. The absolute value of the eigenvalue  $|\lambda| \leq 1$  defines the overlap density between the two states, and  $|\lambda| = 1$  if and only if the states are identical. In such case, the left and right environments are identical up to complex conjugation, as we prove in Appendix E.7.

From the construction of the transfer matrix, these environments are of dimension  $2^{2N_q-2}$ . To translate the environments into variational quantum circuits, we introduce two  $2^{2N_q-2} \times 2^{2N_q-2}$  parametrized environment unitaries  $\hat{E}_r$  and  $\hat{E}_l$ , such that  $|r\rangle = \hat{E}_r(\varphi_r)|0\rangle$  and  $|l\rangle = \hat{E}_l(\varphi_l)|0\rangle$ , as shown in Fig. 8.1 (d). Ultimately, we also consider the decomposition of environment unitaries in the form of the sequential circuits decomposition with  $M_E$  layers, as shown in Fig. 8.1 (e). We discuss the method of obtaining the environments in the next section.

### Evaluating local observables

We evaluate the expectation value of an local observables utilizing the mixed representation,

$$\langle \hat{O} \rangle = \frac{\langle \psi | \hat{O} | \psi \rangle}{\langle \psi | \psi \rangle} = \frac{\langle \psi_R(\boldsymbol{\theta}) | \hat{O} | \psi_L(\boldsymbol{\theta}') \rangle}{\langle \psi_R(\boldsymbol{\theta}) | \psi_L(\boldsymbol{\theta}') \rangle}. \quad (8.2)$$

In Fig. 8.1 (a), we show the circuit representation of the numerator  $\langle \psi_R(\boldsymbol{\theta}) | \hat{O} | \psi_L(\boldsymbol{\theta}') \rangle$ , where  $\hat{O}$  is a local observable that is Hermitian and unitary. Using the definition of the environments, the expectation reduces to

$$\langle \hat{O} \rangle = \frac{\langle l, 0 | \hat{U}_R^\dagger \hat{O} \hat{U}_L | 0, r \rangle}{\langle l, 0 | \hat{U}_R^\dagger \hat{U}_L | 0, r \rangle} = \frac{\langle l, 0 | \hat{U}_R^\dagger \hat{O} \hat{U}_L | 0, r \rangle}{\lambda \langle l | r \rangle}. \quad (8.3)$$

Therefore, the expectation value of local observables can be evaluated by measuring finite circuits, which can be implemented on a quantum computer. The projective measurement on  $|00\dots 0\rangle$  at the end of the circuit in Fig. 8.1 (d) has the probability equal to the squared magnitude of the expectation value  $|\langle l, 0 | \hat{U}_R^\dagger \hat{O} \hat{U}_L | 0, r \rangle|^2$ . The same applies for the denominator. Combining this together, one can measure the squared magnitude of the expectation value  $|\langle \hat{O} \rangle|^2$ . In Appendix E.1, we provide the derivation of the above equations. In the next section, we will describe the procedure to measure the expectation  $\langle \hat{O} \rangle$ , including both real and imaginary parts.

We note that the outlined procedure can be generalized to evaluating correlation functions of the form  $\langle \psi | \hat{A}_i \hat{B}_{i+\delta} | \psi \rangle$ , where the operators  $\hat{A}_i$ ,  $\hat{B}_{i+\delta}$  act on single qubits and are separated by  $\delta$  sites.

### 8.1.2. Translationally invariant Trotterization

The time evolution of an initial wave function  $|\psi_0\rangle$  under the action of a Hamiltonian  $\hat{H}$  is given by application of the evolution operator to the initial state  $|\psi(t)\rangle = \hat{U}_t |\psi_0\rangle = \exp(-it\hat{H})|\psi_0\rangle$ . Here, we consider a Hamiltonian acting on a one-dimensional infinite spin-1/2 chain. When  $\hat{H}$  is local, i. e., can be written as  $\hat{H} = \sum_i \hat{h}_i$  with all terms  $\hat{h}_i$  having a finite support, we can approximate the evolution operator  $\hat{U}_T$  using a sequential Trotter decomposition. A first-order sequential Trotterization can be written as

$$\hat{U}_T = (\hat{u}(\delta t))^k + \mathcal{O}(k\delta t^2), \quad (8.4)$$

where  $\hat{u}(\delta t) = \prod_j \hat{u}_j(\delta t)$ ,  $\hat{u}_j(\delta t) = \exp(i\delta t \hat{h}_j)$  and  $\delta t = T/k$ . A single sequential evolution operator  $\hat{u}(\delta t)$  is shown in Fig. 8.2 (a). Due to the sequential decomposition,  $\hat{u}(\delta t)$  and

hence  $\hat{U}_T$  are translationally invariant with a single site unit cell. Starting with a translationally invariant state, we always need only a single unitary  $\hat{U}(\boldsymbol{\theta})$  parameterizing the state as in Eq. (8.1)<sup>3</sup>. All these considerations can be generalized to cases with a larger unit cell.

### 8.1.3. The time evolution algorithm

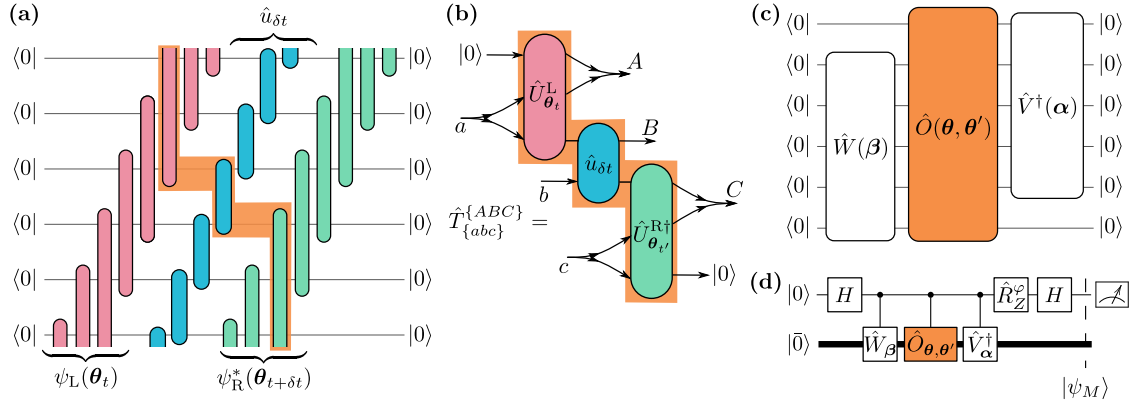


Figure 8.2.: (a) Quantum circuit for the overlap between the time-evolved left-represented state  $\hat{u}(\delta t)|\psi_L(\boldsymbol{\theta}_t)\rangle$  at time  $t$  and right-represented state  $|\psi_R(\boldsymbol{\theta}_{t+\delta t})\rangle$  at time  $t + \delta t$ . For illustration, here  $N_q = 3$  is chosen. The orange shaded area singles out the transfer matrix. (b) The explicit form of the transfer matrix  $\hat{T}_{\{abc\}}^{\{ABC\}}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+\delta t})$ . The capital indices  $\{ABC\}$  form a composite out-index, similarly  $\{abc\}$  form a composite in-index. The arrow direction indicates the operation order (time flow). (c) The circuit of the generalized functional  $\mathcal{L}$ . The unitary  $\hat{W}(\boldsymbol{\beta})$  acts on the first  $2N_q - 1$  qubits of the  $|00\dots 0\rangle$  state prepared on  $2N_q$  qubits, then unitary  $\hat{O}(\boldsymbol{\theta}, \boldsymbol{\theta}')$  acts all qubits and  $\hat{V}^\dagger(\boldsymbol{\alpha})$  acts on the last  $2N_q - 1$  qubits. (d) General Hadamard test scheme for measurement of the algebraic value of  $\mathcal{L}$ .

We now introduce a hybrid quantum-classical algorithm to perform time evolution of the l-USC representation. At the time  $t$ , we parametrize the state-unitary  $\hat{U}_{R/L}(t)$  by a set of variational parameters  $\boldsymbol{\theta}_t$ . The gradients of the parameters are measured on a quantum computer and the update is performed on a classical computer. Here, for the sake of concrete notation, we present the *even* time steps of the algorithm. In these steps, representation of the wave function flips from left to right. The *odd* steps are done similarly, but with flipping from right representation to left.

To perform the time evolution at an even step, one is required to find the closest state  $|\psi_R(\boldsymbol{\theta}_{t+\delta t})\rangle$  in right representation approximating the time-evolved state  $\hat{u}(\delta t)|\psi_L(\boldsymbol{\theta}_t)\rangle$ . The direct measure of the closeness is the fidelity, i.e., squared overlap, between the two states,

$$|\xi(t, \delta t)|^2 = |\langle \psi_R(\boldsymbol{\theta}_{t+\delta t}) | \hat{u}(\delta t) | \psi_L(\boldsymbol{\theta}_t) \rangle|^2.$$

It is the probability of measuring the state  $|\dots 000\dots\rangle$  at the end of the circuit shown in Fig. 8.2 (a). This quantity is either 1 or 0 in the thermodynamic limit and cannot be used for posing the optimization problem. Instead, we define the mixed transfer matrix between

<sup>3</sup>This is in contrast to the classical iTEBD algorithm, where the evolution unitaries at even and odd chain sites are applied consecutively, which results in two-site unit cell [191, 208].

the two states over indices  $\{abc\}$ ,  $\{ABC\}$  as shown in Fig. 8.2 (a-b) with one additional index coming from the trotterized unitary. To find the closest state, we maximize the absolute value of the overlap density  $|\lambda|$  with respect to the parameters at the next time step  $\boldsymbol{\theta}_{t+\delta t}$ . The squared magnitude of the overlap density  $|\lambda|^2$  is the probability of measuring the state  $|00\dots 0\rangle$  at the end of the circuit shown in Fig. 8.2 (c).

We solve the maximization problem with a gradient ascent algorithm which requires the knowledge of environments  $|l\rangle$  and  $|r\rangle$  and the leading eigenvalue from the mixed transfer matrix. To obtain the environments  $|l\rangle$  and  $|r\rangle$ , we employ the *modified power method*. We describe the procedure of obtaining the right environment  $|r\rangle$ , while the procedure for the left environment is similar, apart from the replacement  $T \rightarrow T^\dagger$ . The idea of the power method is to take an initial state  $|\psi_0\rangle$  and project it onto the leading eigenvector of  $\hat{T}$  by repeated application of  $\hat{T}$ , because  $\lim_{p \rightarrow \infty} (T/\lambda)^p \rightarrow |r\rangle\langle l|$ . Here, we consider an iterative algorithm, which is a slight modification of the power method: at each step, we find the new vector  $|r(\boldsymbol{\varphi}'_r)\rangle$  by performing only a single gradient descent step maximizing the overlap magnitude  $|\lambda|^2 = |\langle r(\boldsymbol{\varphi}'_r)|\hat{T}|r(\boldsymbol{\varphi}_r)\rangle|^2$  with respect to  $\boldsymbol{\varphi}'_r$ . Namely,  $\boldsymbol{\varphi}'_r \leftarrow \boldsymbol{\varphi}_r + \eta \nabla_{\boldsymbol{\varphi}'_r} |\lambda|^2$ , where  $\eta$  is the learning rate. Alternatively, a gradient-free method, such as Rotosolve [310–312, 333], could be used. The environment vector is then updated  $|r(\boldsymbol{\varphi}_r)\rangle \leftarrow |r(\boldsymbol{\varphi}'_r)\rangle$  and is used in the next iteration. At each step,  $|r(\boldsymbol{\varphi}_r)\rangle$  has a strictly increasing overlap with the leading eigenvector of  $\hat{T}$  provided a small enough step size  $\eta$ . The method is presented in Algorithm 2.

---

**Algorithm 2:** The power method for an environment

---

**Input** :  $\hat{T}, \boldsymbol{\varphi}_r^p$   
**Output:** Environment  $|r(\boldsymbol{\varphi}_r)\rangle$   
 $\boldsymbol{\varphi}_r, \boldsymbol{\varphi}'_r \leftarrow \boldsymbol{\varphi}_r^p$  ▷ from the previous step ;  
**while** *not converge* **do**  
     Measure  $\lambda = \langle 0|\hat{E}_r^\dagger(\boldsymbol{\varphi}'_r)\hat{T}\hat{E}_r(\boldsymbol{\varphi}_r)|0\rangle$  ;  
     Measure  $\nabla_{\boldsymbol{\varphi}'_r} \lambda$  ▷ See Eq. (8.5) ;  
      $\nabla_{\boldsymbol{\varphi}'_r} |\lambda|^2 = 2\text{Re} [\lambda^* \nabla_{\boldsymbol{\varphi}'_r} \lambda]$  ;  
      $\boldsymbol{\varphi}'_r \leftarrow \boldsymbol{\varphi}'_r + \eta \nabla_{\boldsymbol{\varphi}'_r} |\lambda|^2$  ;  
      $\boldsymbol{\varphi}_r \leftarrow \boldsymbol{\varphi}'_r$  ;  
**end**  
 Return  $|r(\boldsymbol{\varphi}_r)\rangle = \hat{E}_r(\boldsymbol{\varphi}_r)|\mathbf{0}\rangle$  ;

---

Next, we show in Algorithm 3 how to perform a time evolution step using gradient ascent methods with the environments we obtained. The algorithm uses a nested variational approach, in which left- and right-environments are variationally optimized between the consecutive gradient descent steps. Both algorithms are run until the change of  $|\lambda|^2$  between two consecutive iterations becomes smaller than  $10^{-12}$ .

Note that we can use the algorithm to find the opposite representation of the same wave function if the time evolution operator is taken to be the identity. The algorithm proposed here resembles the time evolution algorithm for a finite size system [4, 315] and for an infinite system [213]. As the main difference, in this work the (mixed) transfer matrix is constructed as the mixed representation  $\langle \psi_R(\boldsymbol{\theta})|\psi_L(\boldsymbol{\theta}')\rangle$ . The environments of the transfer matrix are represented as quantum states parametrized with layered sequential circuits. We study the effect of such approximation in Sec. 8.2 (B).

#### 8.1.4. Required measurements

To implement the time-evolution Algorithm 3 in practice, one has to measure the algebraic value of the overlap  $\lambda = \langle 0|\hat{E}_r^\dagger(\boldsymbol{\varphi}'_r)\hat{T}\hat{E}_r(\boldsymbol{\varphi}_r)|0\rangle$ , of its derivative with respect to the pa-

**Algorithm 3:** Time evolution algorithm for l-USC

---

**Input** :  $\boldsymbol{\theta}_t, \boldsymbol{\varphi}_r^t, \boldsymbol{\varphi}_l^t$   
**Output:** New parameters  $\boldsymbol{\theta}_{t+\delta t}$  after one evolution step  
 $\boldsymbol{\theta}_{t+\delta t} \leftarrow \boldsymbol{\theta}_t$ ;  
 $\boldsymbol{\varphi}_r, \boldsymbol{\varphi}_l \leftarrow \boldsymbol{\varphi}_r^t, \boldsymbol{\varphi}_l^t$ ;  
**while** *not converged* **do**  
     $\boldsymbol{\varphi}_r \leftarrow \text{Environment}(\hat{T}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+\delta t}), \boldsymbol{\varphi}_r)$ ;  
     $\boldsymbol{\varphi}_l \leftarrow \text{Environment}(\hat{T}^\dagger(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+\delta t}), \boldsymbol{\varphi}_l)$ ;  
     $|r\rangle \leftarrow \hat{E}_r(\boldsymbol{\varphi}_r), |l\rangle \leftarrow \hat{E}_l(\boldsymbol{\varphi}_l)$ ;  
     $\lambda = \langle l | \hat{T}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+\delta t}) | r \rangle / \langle l | r \rangle$ ;  
     $\nabla_{\boldsymbol{\theta}_{t+\delta t}} |\lambda|^2 = 2 \text{Re} [\lambda^* \nabla_{\boldsymbol{\theta}_{t+\delta t}} \lambda]$  ; ▷ See Eq. (8.5) ;  
     $\boldsymbol{\theta}_{t+\delta t} \leftarrow \boldsymbol{\theta}_{t+\delta t} + \eta \nabla_{\boldsymbol{\theta}_{t+\delta t}} |\lambda|^2$ ;  
**end**  
**return**  $\boldsymbol{\theta}_{t+\delta t}$ ;

---

parameters of the environment  $\nabla_{\boldsymbol{\varphi}_r'} \lambda = \langle 0 | \nabla_{\boldsymbol{\varphi}_r'} \hat{E}_r^\dagger(\boldsymbol{\varphi}_r') \hat{T} \hat{E}_r(\boldsymbol{\varphi}_r) | 0 \rangle$ , and of its derivative with respect to the state unitary

$$\nabla_{\boldsymbol{\theta}_{t+\delta t}} \lambda = \frac{\langle l | \nabla_{\boldsymbol{\theta}_{t+\delta t}} \hat{T}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+\delta t}) | r \rangle}{\langle l | r \rangle}. \quad (8.5)$$

We prove the latter formula in Appendix E.1.3. Notably, the implicit dependency of the right- and left-environments on  $\boldsymbol{\theta}_{t+\delta t}$  gives no contribution to the gradient. These expectation values can all be expressed in terms of a general overlap functional  $\mathcal{L}[\hat{V}(\boldsymbol{\alpha}), \hat{O}(\boldsymbol{\theta}, \boldsymbol{\theta}'), \hat{W}(\boldsymbol{\beta})] = \langle 0 | \hat{V}^\dagger(\boldsymbol{\alpha}) \hat{O}(\boldsymbol{\theta}, \boldsymbol{\theta}') \hat{W}(\boldsymbol{\beta}) | 0 \rangle$ . In our case,  $\hat{V}(\boldsymbol{\alpha})$  and  $\hat{W}(\boldsymbol{\beta})$  are the environment unitaries or their derivatives, while  $\hat{O}(\boldsymbol{\theta}, \boldsymbol{\theta}')$  is the transfer matrix or its derivatives. We note that all mentioned unitaries' derivatives are also unitary due to the specific parametrization of the two-qubit gates (for details, see Appendix E.2).

Absolute and algebraic values of this functional can be measured on a quantum computer. First, the square of the magnitude  $|\mathcal{L}|^2$  is given by the probability of projection onto the  $|00\dots 0\rangle$  state in the circuit in Fig. 8.2 (c) [334]. The algebraic value of the expectation  $\mathcal{L}$  and its derivative  $\nabla_{\boldsymbol{\varphi}_r'} \mathcal{L}$  can be measured within the Hadamard test procedure [335] shown in Fig. 8.2 (d). We denote  $|\bar{0}\rangle = |00\dots 0\rangle$  and the quantum state before the ancilla qubit measurement reads

$$|\psi_M\rangle = \frac{1}{2} |0\rangle \otimes \left( |\bar{0}\rangle + e^{i\varphi} \hat{V}^\dagger(\boldsymbol{\alpha}) \hat{O}(\boldsymbol{\theta}, \boldsymbol{\theta}') \hat{W}(\boldsymbol{\beta}) |\bar{0}\rangle \right) + \frac{1}{2} |1\rangle \otimes \left( |\bar{0}\rangle - e^{i\varphi} \hat{V}^\dagger(\boldsymbol{\alpha}) \hat{O}(\boldsymbol{\theta}, \boldsymbol{\theta}') \hat{W}(\boldsymbol{\beta}) |\bar{0}\rangle \right). \quad (8.6)$$

The probability difference in measurement over the ancilla qubit yields

$$p(|0\rangle) - p(|1\rangle) = \text{Re} \left[ e^{i\varphi} \langle \bar{0} | \hat{V}^\dagger(\boldsymbol{\alpha}) \hat{O}(\boldsymbol{\theta}, \boldsymbol{\theta}') \hat{W}(\boldsymbol{\beta}) | \bar{0} \rangle \right]. \quad (8.7)$$

This scheme can also be used for obtaining algebraic values of the observable expectation  $\langle \mathcal{O} \rangle$  introduced in Section II. A.



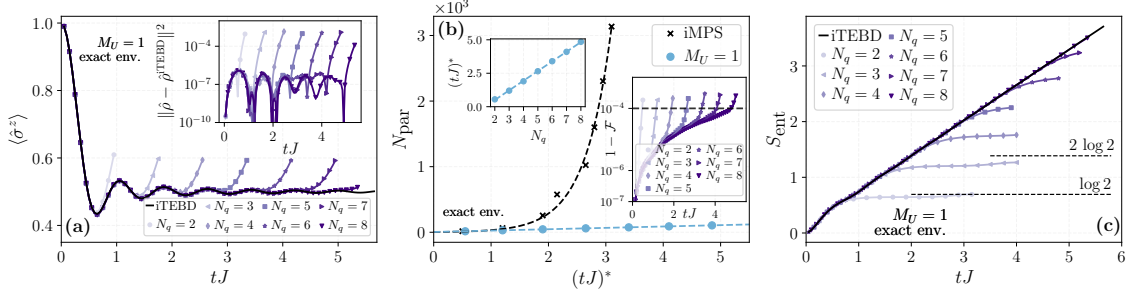


Figure 8.3.: Simulations of time evolution using l-USC with  $M_U = 1$  and exact environment for the Hamiltonian in Eq. (8.8) with  $g/J = 1.0$  and  $h/J = 0$ . (a) Expectation value of  $\langle \hat{\sigma}^z(t) \rangle$  using l-USC with different values of  $N_q$ . Inset: the difference in the single-site density matrices between the quasi-exact iTEBD simulation and optimization of l-USC. (b) The number of parameters required to reach time  $t^*$  with the fidelity density at least  $\mathcal{F} = 1 - 10^{-4}$ . The black crosses represent the standard iTEBD approach with the black dashed line showing an exponential fit. The blue markers show the results for l-USC, while the line shows the linear fit. Left inset: the reachable time  $t^*$  under the condition  $\mathcal{F} \geq 1 - 10^{-4}$  as a function of  $N_q$ . The dashed line shows a linear fit. Right inset: error in fidelity density  $1 - \mathcal{F}$  for different  $N_q$ . (c) Entanglement entropy as a function of evolution time  $tJ$  compared with the quasi-exact result. The horizontal dashed lines mark the theoretical maximum entanglement entropy levels  $(N_q - 1) \log 2$ .

## 8.2. Results

To benchmark the proposed algorithm, we simulate the quenched dynamics of the transverse-field Ising model with the longitudinal field

$$\hat{H} = J \sum_i \sigma_i^x \sigma_{i+1}^x + g \sum_i \sigma_i^z + h \sum_i \sigma_i^x \quad (8.8)$$

over an infinite spin-1/2 chain. The initial wave function is taken as a fully-magnetized state  $|\psi_0\rangle = |\dots 000 \dots\rangle$  in the  $\sigma^z$  basis. We use a fourth-order Trotterized iTEBD simulation with  $\chi = 1024$  uniform MPS and  $\delta t J = 10^{-2}$  as a quasi-exact reference labeled *iTEBD* in all figures. In the following, we simulate the algorithm on a classical computer to study the properties of the l-USC ansatz. The complexity of the l-USC ansatz is controlled by  $M_U$  and  $N_q$ . The complexity of measuring local observables and running the time-evolution algorithm depends additionally on  $M_E$ , i. e., the accuracy of approximating the environments. In Sec. 8.2.1, we study the effect of varying  $N_q$  and  $M_U$  in simulating the time evolution with exact environment obtained by exact diagonalization of the transfer matrix. In Sec. 8.2.2, we study the accuracy of approximating an exact environment with a layered circuit with finite  $M_E$ , derive the relation between the required  $M_E$  and  $N_q$  and perform the full realistic simulation with both state and environment presented in the sequential form. Finally, in Sec. 8.2.3, we demonstrate measuring the evolution of physical observables on a QPU with a classically optimized l-USC circuit.

### 8.2.1. Layered state unitary, exact environment

As the first step, we study the performance of the l-USC ansatz using exact environments obtained through direct diagonalization of the transfer matrix, and perform the algorithm



outlined in Algorithm 3. In Fig. 8.3, we show the simulation results obtained with  $M_U = 1$  and various  $N_q$ .

In Fig. 8.3 (a), we plot the evolution of the local magnetization  $\langle \sigma^z(t) \rangle$  and observe that the time of deviation from the quasi-exact solution increases with  $N_q$ . In the inset, we plot the Frobenius norm squared of the difference in the single-site density matrices between the quasi-exact state and the l-USC, i. e.  $\|\rho - \rho^{\text{exact}}\|^2$ . The difference shows fluctuating behavior as a function of  $tJ$ , but at some point shows rapid growth. This fast growth coincides in evolution time  $t$  with the noticeable discrepancy in  $\langle \sigma^z(t) \rangle$ .

To quantify the representation capacity of l-USC, we define the reachable time  $t^*$  of the given ansatz as the time when the error in fidelity density with the quasi-exact (iTEBD) state crosses the threshold value  $1 - \mathcal{F} = 10^{-4}$ . Here,  $\mathcal{F}$  is the fidelity density, i. e., squared overlap per unit cell, between the l-USC state and the iTEBD wave function. In Fig. 8.3 (b), we plot the number of parameters in a given circuit against the reachable dimensionless time  $t^*J$ . We see that within l-USC, the required number of parameters grows linearly with the reachable time. Note that, in contrast, the number of parameters required to store a d-USC state grows exponentially in the reachable time  $t^*J$ . Therefore, the l-USC ansatz defines a sub-manifold of uniform MPS that is relevant for representing states under time evolution. Namely, the l-USC is “sparse” as compared to d-USC (uniform MPS) and requires exponentially fewer parameters.

In practice, one does not have access to the exact state and, therefore, no access to the error in fidelity. Instead, one can utilize the leading transfer matrix eigenvalues  $\{\lambda_i\}$  obtained at all steps of the time evolution and define the accumulated error measure  $\mathcal{M}(t) = 1 - \prod_{i < t} |\lambda_i|^2$  to monitor the error and understand whether the simulation result is reliable. In Appendix E.4, we demonstrate that this measure follows closely the true infidelity  $1 - \mathcal{F}$  and thus can be used for assessment of the optimization quality.

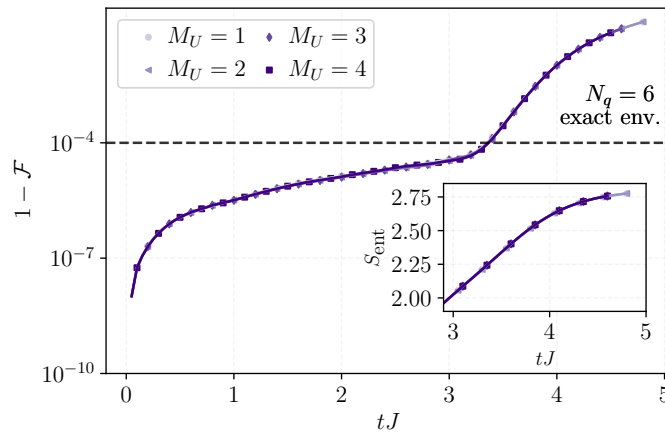


Figure 8.4.: Fidelity density between the quasi-exact simulation and the time-evolved l-USC state with  $N_q = 6$  and varied number of layers  $M_U$ . Inset: the entanglement entropies obtained within optimization of the l-USC circuit at  $N_q = 6$ .

Lastly, in Fig. 8.3 (c) we show entanglement entropy as a function of  $tJ$ . The deterioration of wave function quality, as shown in Fig. 8.3 (a-b), is clearly connected to saturation of capability of a circuit with given  $N_q$  to encode the linearly-growing entanglement entropy of the system. An l-USC of given  $N_q$  could encode at most  $S_{\text{ent}} \leq (N_q - 1) \log 2$  entanglement entropy. We show in Fig. 8.3 (c) that the entanglement grows linearly with time up to saturation. We observe that for  $N_q = 2$  and 3 the saturated entanglement entropy is close to the theoretical bound and for  $N_q \geq 4$  the entanglement entropy does not reach the theoretical bound.

To study the effect of increasing the  $M_U$  in the layered sequential unitary decomposition, in Fig. 8.4 we compare the quality of time-evolved l-USC ansätze with  $1 \leq M_U \leq 4$ . Strikingly, we observe that increasing  $M_U$  leads to negligible improvement in the reachable time  $t^*J$ , as compared with the effect of  $N_q$ . Similarly, increasing  $M_U$  does not lead to a significant change in the entanglement entropy of the time-evolved l-USC ansatz. In Appendix E.3, we demonstrate that time-evolution of the l-USC ansatz at  $M_U = 1$  leads to the same wave function accuracy, as optimization of the full dense d-USC ansatz at  $\chi = 2^{N_q-1}$ .

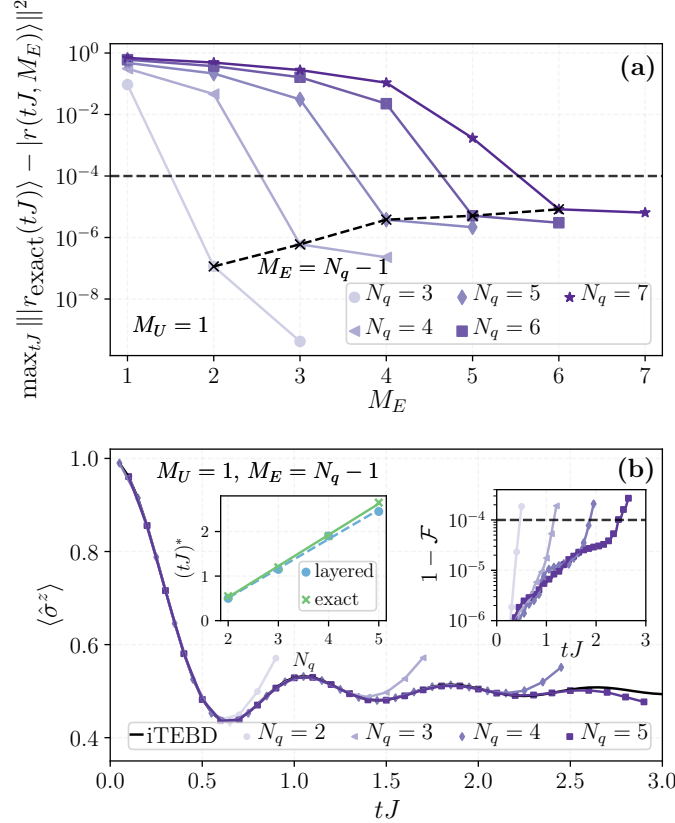


Figure 8.5.: **(a)** Maximum (over evolution time) error of approximating the exact environment by the  $M_E$ -layered environment throughout the full simulation shown in Fig. 8.3 (b). The black dashed line shows the environment approximation error at  $M_E = N_q - 1$ . **(b)** Expectation value  $\langle \hat{\sigma}^z \rangle$  obtained within the full algorithm of time-evolution of l-USC with  $M_U = 1$  and layered environment with  $M_E = N_q - 1$ . Left inset: the reachable time  $t^*$  of simulations with the exact environment and the layered environment with  $M_E = N_q - 1$ . The lines show linear fits. Right inset: error in fidelity density  $1 - \mathcal{F}$  for different  $N_q$ .

### 8.2.2. Optimization with a layered environment

In the previous section, we have shown that the state-unitary  $U$  can be approximated by the layered quantum circuit using exponentially fewer parameters than the d-USC ansatz. In a real simulation, however, the environments should also be approximated. We now investigate if the environments  $|l\rangle, |r\rangle$  can also be represented by layered quantum circuits using fewer parameters. To address the question, we take the exact environments obtained during the  $M_U = 1$  simulation shown previously in Fig. 8.3 (b) and approximate them with the  $M_E$ -layer sequential circuits. The approximation is based on an alternative update

with polar decomposition outlined in Appendix E.2. We plot the approximation error in Fig. 8.5 (a).

We observe that, for a fixed  $M_U = 1$ , the environment approximation error increases upon increasing  $N_q$ , while the error decreases with increasing  $M_E$ . We set a threshold  $10^{-4}$  in the approximation errors for the environment, which is motivated by the respective error threshold in the fidelity density. From Fig. 8.5 (a), we see that the error in the environment approximation remains strictly below  $10^{-4}$  during the whole time evolution, if it is approximated using  $M_E = N_q - 1$  layers. It remains an open question on whether the approximation holds for larger  $N_q$ .

When our observation holds, this allows one to determine the number of variational parameters to approximate the environment. Since  $M_E \propto N_q$ ,  $N_q \propto t^* J$  (see Fig. 8.3 (b)), and each layer of the sequential circuit for the environment contains  $2N_q - 2$  two-qubit gates, representing the environment requires  $\mathcal{O}(t^2)$  or  $\mathcal{O}(N_q^2)$  two-qubit gates. By contrast, representing the environment exactly using dense unitary requires  $\mathcal{O}(e^{2N_q})$ , or, equivalently,  $\mathcal{O}(e^{2tJ})$  parameters. The fact that we can approximate the environment with quantum circuits efficiently makes the overall algorithm scaling polynomially in time  $tJ$  instead of exponentially. We note that although *a priori* the complexity of the environment approximation for the l-USC ansatz is not known, there exist exact solutions representing the environments for an infinite brickwall circuits [289, 331]. These exact solutions are formed by contracting  $\mathcal{O}(N_{\text{depth}}^2)$  number of gates, where  $N_{\text{depth}}$  is the depth of the brickwall circuit, which is consistent with our finding here.

Using the condition  $M_E = N_q - 1$ , we simulate classically the time evolution algorithm using l-USC with  $M_U = 1$ , different  $N_q$ , and layered sequential circuits for the environment. We plot the  $\langle \sigma^z \rangle$  expectation value, the reachable time  $(tJ)^*$  as a function of  $N_q$ , and the error in the fidelity density  $1 - \mathcal{F}$  in Fig. 8.5 (b). The reachable time is again determined by the threshold value  $\mathcal{F} = 1 - 10^{-4}$ . As shown in the left inset, we observe slightly smaller reachable times, compared to the simulation with the exact environment, due to the accumulation of the approximation errors and approximated environment. Nevertheless, the reachable time  $(tJ)^*(N_q)$  retains the linear scaling with  $N_q$ .

### 8.2.3. Simulation on QPU

Given the available cloud-based QPU from IBM-Q, we implement the circuit shown in Fig. 8.1 (d) for  $N_q = 2$  and measure  $|\langle \sigma^z(t) \rangle|^2$  as the probability of projecting onto the  $|00 \dots\rangle$  state. The parameters of the environments and states are optimized on a classical computer. Unfortunately, the available hardware does not allow to use controlled two-qubit gates in large amounts, since they require decomposition into several non-controlled two-qubit gates. Due to error and noise levels, this is out of reach for the available device. This prevents us from measuring the algebraic value of  $\langle \sigma^z(tJ) \rangle$  using the Hadamard test. Nevertheless, the numbers of qubits and gates required to run the algorithm until the time  $t^*$  scale linearly and quadratically, respectively with  $t^*$ . Depending on the device connectivity, an additional constant to linear factor in  $t^*$  overhead may occur in implementing the controlled-unitary operation. Therefore, with improved read-out and gate noise level, we expect this algorithm to be usable on the NISQ devices. For the measurement of the squared magnitude, we consider two parameter sets:  $g = 0.2 J$ ,  $h = 0$  and  $g = 0.8 J$ ,  $h = 0.05 J$  and  $N_q = 2$ ,  $M_U = M_E = 1$ . To mitigate the device noise, we employ the *randomized circuits averaging* introduced in Ref. [4] (for details, see Appendix E.6), and readout error mitigation.

In Fig. 8.6, we include data obtained from various sources. This includes the quasi-exact simulation (iTEBD), the classical simulation of the algorithm (l-USC), a simulation of the magnetization measurement on a fault-tolerant device using a finite number of circuit

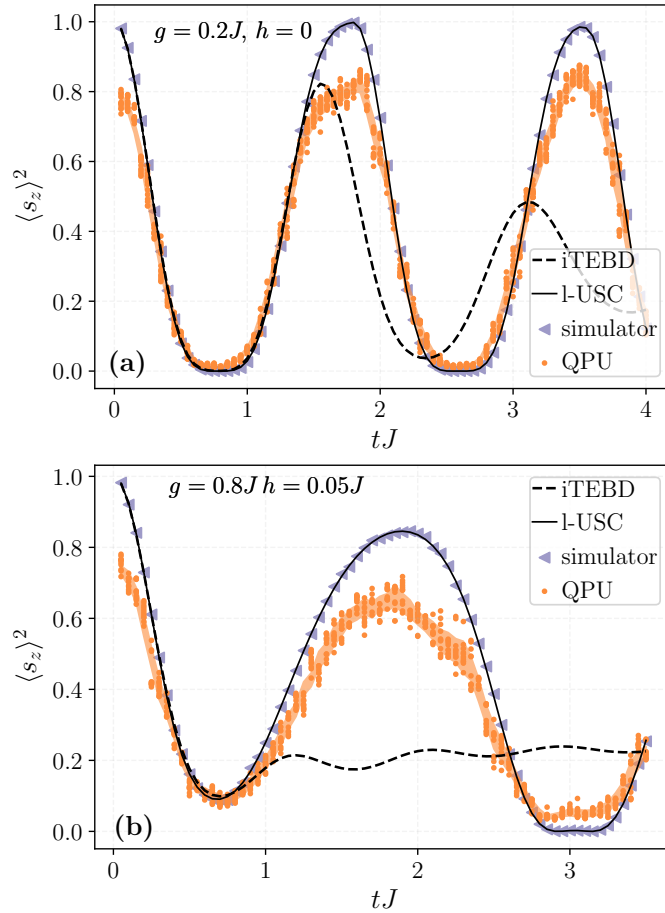


Figure 8.6.: **(a)**: Expectation value of  $|\langle \hat{\sigma}^z(t) \rangle|^2$  obtained by various methods: quasi-exact (iTEBD), the presented algorithm with  $N_q = 2, M_U = M_E = 1$  (l-USC), simulation of an ideal quantum device (simulator), and on the real hardware `ibmq-jakarta` (QPU) at  $h/J = 0, g/J = 0.2$ . **(b)**:  $h/J = 0.05, g/J = 0.8$ .

‘shots’ (simulator), and the direct measurement on a real hardware device `ibmq-jakarta` (QPU). Due to the small expressive power of the quantum circuit at  $N_q = 2, M_U = M_E = 1$ , the exact and the simulated time evolution algorithm results agree only up to  $tJ = 1.4$  in the former and 0.8 in the latter cases. However, the quantum hardware measurement shows a good degree of agreement with the classical simulation of the l-USC ansatz and the simulation of QPU on a classical computer.

### 8.3. Discussion

In this work, we introduced and studied a hybrid quantum-classical algorithm for time evolution based on the l-USC ansatz, which is a generalization of the single-layer sequential quantum circuit motivated by uniform MPS. Unlike previous works [213, 214, 289], we construct the transfer matrix in the mixed representation. In this formalism, the environments are pure states instead of density matrices. We represent the environments by quantum circuits and determine the variational parameters of these circuits using gradient descent. Based on the result from classical simulation, we observe that the number of parameters required to accurately represent the state at a given time  $t$  scales linearly with  $t$ , which gives an exponential advantage compared to classical algorithms based on MPS. While such scaling is anticipated based on the theoretical prediction [295], more interestingly,

we observe numerically that the number of parameters required to represent the environment scales quadratically in the evolution time. This suggests that while the ansatz has a linearly scaling number of parameters with the evolution time, the overall algorithm for simulating the time evolution of an infinite system has complexity scaling quadratically in the evolution time using quantum computers with a finite number of qubits. Importantly, by working directly in the thermodynamic limit, complexity does not scale with the system size  $L$  which is in contrast with the  $\mathcal{O}(Lt)$  complexity scaling required for a finite system simulation [4]. We emphasize that unlike Ref. [4, 213], we consider the multi-layered decomposition of the state-unitary with  $M_U \geq 1$ . However, as we have seen from Fig. 8.3 (b), Fig. 8.4, and Appendix E.3, considering  $M_U > 1$  does not lead to any sizable improvement in the ansatz performance for time evolution.

We note that we can also perform imaginary time evolution with the proposed algorithm with the price of one additional ancilla qubit [336] to realize the non-unitary gates in the transfer matrix. One straightforward application is the study of ground states for infinite systems. Important questions remain on whether one would observe similar polynomial advantages [212] in representing the ground state using l-USC for an infinite system, and on whether the environments of the ground states can be efficiently represented as quantum circuits. Another future direction is to consider the generalization for quantum systems and circuits in two dimensions. For instance, recently the formal generalization of sequential quantum circuits to finite two-dimensional (2D) systems is proposed [156]. The study of ground states of finite 2D systems is performed using quantum circuits of isometric tensor network states [155]. We believe it will be therefore beneficial to generalize the l-USC ansatz and extend the algorithm to infinite two-dimensional systems.



## 9. Conclusion and outlook

In this thesis, we studied classical and quantum algorithms for simulating quantum many-body dynamics. In part I of the thesis, we focused on classical algorithms and showed the efficient simulation with isometric tensor network states (isoTNSs) on a finite-by-finite and finite-by-infinite two-dimensional (2D) geometry. While the tensor-network-based methods are restricted by entanglement, we showed that the neural-network-based methods are not restricted by entanglement but still have exponential scaling in the number of parameters in simulating global quench for chaotic Hamiltonian. In part II of the thesis, we focused on quantum algorithms and showed the potential exponential quantum advantage in simulating both finite and infinite one-dimensional (1D) systems.

In this concluding chapter, we remark on the key results of each part of this thesis. Following the remark, we expand the discussion at the end of each chapter and provide an overview of the remaining open questions and how we might move forward.

### 9.1. Part I: Classical simulation with isometric tensor network states

#### Summary:

In the first three chapters of Part I, we focused on the classical simulation using tensor network methods. We have briefly reviewed the 1D and 2D tensor network states (TNSs) and the isoTNSs. IsoTNSs generalize the matrix-product state (MPS) isometric form to higher dimensions, and the Moses move (MM) [75] serves as an approximate numerical method for moving the orthogonality center. The well-known algorithms for 1D MPSs, i.e., time-evolving block decimation (TEBD) and density matrix renormalization group (DMRG), can be generalized to 2D isoTNSs. In Chapter 3, we first revisited the MM algorithms and showed better results in sequential MM when directly minimizing the truncation error in the tripartite decomposition. We then introduced the first main result of this part, namely, a 2D generalization of the DMRG algorithm (DMRG<sup>2</sup>) for ground state search. Similar to the 2D generalization of the TEBD algorithm (TEBD<sup>2</sup>), the DMRG<sup>2</sup> utilizes MM to move the orthogonality center around and perform tensor updates on the orthogonality center. We observed similarly the DMRG<sup>2</sup> algorithm is also affected by the MM error.

We obtained the second main result of this part with the TEBD<sup>2</sup> algorithm, demonstrating the ability to simulate 2D real-time evolution. The accessible time scale and accuracy of the TEBD<sup>2</sup> algorithm are restricted by the entanglement entropy and the MM error. We simulated the dynamics of the locally perturbed ground state of the transverse field Ising model on a square lattice and the Kitaev model on a honeycomb lattice and computed the time-dependent correlation functions. By Fourier transforming the time-dependent correlation functions, we obtained the dynamical spin structure factor, which shows the low-energy spectral properties of the underlying systems.

In Chapter 4, we gave the third important result of this part, generalizing the MM on finite systems to infinite systems. The infinite Moses move (iMM) splits an infinite 1D wavefunction into the infinite projector column and the new infinite 1D wavefunction. The iMM first iteratively solves the tripartite decomposition and then minimizes the error of the approximate tripartite decomposition at the fixed point. With the iMM, we can

generalize the study of 2D isoTNSs on finite systems to finite by infinite systems. Since we consider only the open boundary condition, the systems are defined on infinite strips. We showed that we could compute physical observables efficiently and generalize the TEBD<sup>2</sup> algorithm to an infinite TEBD<sup>2</sup> algorithm.

### Outlook:

- Accessing excited states, excitation spectrum, and the entanglement entropy:** Perhaps the most interesting outlook is to utilize the isometric structure further and explore the possibility of accessing excited states and entanglement entropy. For a finite strongly disordered system, we can modify the DMRG algorithm [337] to access the highly excited state that has a large overlap with the given initial state. This so-called DMRG-X algorithm relies on the isometric structure of the MPS. We can similarly modify the DMRG<sup>2</sup>, and potentially study the highly excited state in 2D. For finite critical systems, it has been shown that the excitation spectrum of the effective Hamiltonian during the DMRG sweep accurately captures the exact low-energy spectrum [338]. It is an interesting application to explore the generalization in 2D with DMRG<sup>2</sup>. Preliminary results of both directions above suggest it is indeed possible to access the excited state or excitation spectrum, limited again by the finite MM error. Finally, an additional application given the isometric structure is the possibility of reading out the bipartite entanglement entropy of the system. Consider the orthogonality column without physical indices, which corresponds to a column encoding the Schmidt spectrum information. It would be interesting to see if we can compute the entanglement entropy using the recently developed tensor network algorithm [339].
- Infinite boundary contraction:** In Chapter 4, we showed that one way to evaluate the expectation value of local observables of isoTNSs on an infinite strip is to find the fixed point of the transfer matrix by the finite boundary matrix-product operator (bMPO) method. On the other hand, it is possible to perform boundary contraction in the other direction using an infinite bMPO [132]. With such contraction, we can generalize the variational uniform matrix-product state (VUMPS) [340], algorithm for finding ground states with 2D isoTNSs.
- Incorporating triad tensors or anisotropic tensors:** Some recent development in higher-dimensional tensor network algorithms, namely, the triad tensor networks [181] and anisotropic tensor renormalization [341], present modified algorithms with lower memory footprint and computation cost. These advancements are primarily based on low-rank tensor decomposition, for example, representing a rank-4 tensor by the approximate contraction of two rank-3 tensors. It has been shown that trading off the representation power with lower computation cost and increasing the overall bond dimension could lead to more accurate results [342–344]. The idea can be naturally incorporated into the isoTNS framework by decomposing the rank-5 tensor into contraction of lower rank tensors. The modification can lead to a MM with complexity  $D^6$ , which may offer potential advantages over the original MM.
- Pushing the numerical limits:** Despite the better result in tripartite decomposition by minimizing truncation error, we refrained from it and instead minimized the Rènyi-1/2 entanglement entropy in all the works due to stability and convergence issues. Potentially, finding a stable and efficient way to directly minimize the truncation error in tripartite decomposition could lower the MM error, which is the current limitation of all isoTNSs algorithms. While we can continue pushing the



limit of isoTNSs algorithms by increasing the bond dimension, perhaps another important aspect is to put a comparison with generic TNSs algorithms. Assuming the bond dimension on the orthogonality hypersurface  $\eta$  scales linear with the overall bond dimension  $D$ , the MM and TEBD<sup>2</sup> have the complexity  $D^7$ , which is between the complexity of the simple update (SU)  $D^6$  and full update (FU)  $D^{10}$  methods of general TNSs. In this respect, TEBD<sup>2</sup> algorithm is an efficient algorithm that provides insight into the dynamics and ground state properties, while having different restrictions compared to the SU method. While the SU method is systematically biased (even with increasing bond dimension) by the trivial boundary approximation, isoTNSs-based methods are unbiased as we observed vanishing MM error with increasing bond dimension. A yet-to-be-answered question is to understand whether isoTNSs can reach the same accuracy as TNSs under the same amount of computation costs.

- **Open questions:** The above discussion leads to the important open question: *Can isoTNSs efficiently represent local gapped Hamiltonian in 2D?* The question remains open even for general 2D TNSs [41]. So perhaps, we can focus on the question: *What class of states can isoTNSs efficiently represent?* It has been shown [152] that 2D string-net liquids states [153] admit exact isoTNSs representation. Further study in this direction would facilitate our understanding of isoTNSs. Another direction is to study the finite entanglement scaling of infinite isoTNSs on an infinite strip. In this regard, new algorithms are required to access the bipartite entanglement entropies for a cut along the finite direction. We leave this as future work.

## 9.2. Part I: Classical simulation with neural-network quantum states

### Summary:

In the last chapter of Part I, we considered simulating quantum dynamics with neural-network quantum states (NQSs). As a newly developed numerical method, NQSs have shown comparable results for simulating quantum many-body ground states and dynamics. At the same time, less is known about the limitation of NQSs. Based on numerical simulation, we studied the number of parameters required to approximate the quantum state undergoing global quench dynamics to a given accuracy. For a fixed system size, we found that the required number of parameter scales exponentially with the evolution time. Moreover, we found the coefficient of the scaling is problem dependent but is not affected by the network architecture over a wide range of different design choices: shallow and deep networks, small and large filter sizes, dilated and normal convolutions, with and without shortcut connections.

### Outlook:

With the remarkable progress and surprising performance in large neural networks [267, 345, 346], we believe we are just at the beginning page of numerical simulations based on NQSs, and we anticipate more exciting development is yet to come. While incorporating the autoregressive property [78] provided us with stable training with large models, recent developments, for example, the quantization [347] or improving optimizer [348], may push this even further. One exciting direction of having large models is the possibility of encoding an ensemble of quantum states [349]. Perhaps we are close to our chatGPT moment. While the physics community can benefit from learning new developments in the computer science community, hopefully, the reverse also holds true that insight from physics can also help

train larger neural networks more efficiently [346, 350]. In the following, we list the outlook and open questions of NQSs.

- **Simulating dynamics of 2D systems:** Although restricted by entanglement growth, tensor network algorithms give access to short-time dynamics efficiently and accurately and are so far the best simulation methods for 1D quantum systems. In 2D, the tensor network algorithms for simulating real-time dynamics are less explored [64, 148, 351, 352], while algorithms based on NQSs have shown competitive results recently, such as simulating quench dynamics [77, 353] and resolving spectral functions [354]. It is argued that the result is not limited by the expressivity of the NQSs but instead by the instability of the variational Monte Carlo (VMC) method [77, 270]. The study of expressivity and instability could benefit from each other. We think an important step is to extend our study to 2D and understand the restriction and scaling of the NQSs in representing time-evolved states. Resolving the instability and expressivity issue may bring a great step forward in simulating 2D dynamics.
- **Combining neural autoregressive quantum states (NAQSs) and isoTNSs:** As both NAQSs and isoTNSs admit direct sampling [355], it is natural to ask whether we can combine the two ansätze together. On the one hand, NAQSs parameterize the conditional probability amplitudes and describe autoregressive quantum states. On the other hand, the isometric tensors in isoTNSs can be interpreted as conditional probability amplitudes over physical and virtual degrees of freedom,

$$B_{m_{i-1}, m_i}^{\sigma_i} = \psi(\sigma_i, m_i | m_{i-1}) \quad (9.1)$$

The probability amplitude of an isoTNS is then interpreted as the marginalized probability amplitude from the joint probability amplitude, including both physical and virtual degrees of freedom,

$$\psi_{\sigma_1 \sigma_2 \dots \sigma_N} = \sum_{m_1, m_2, \dots} \psi(\sigma_1, \sigma_2, \dots, \sigma_N, m_1, m_2, \dots, m_{N-1}). \quad (9.2)$$

Taking this subtle difference into account, we can consider the neural autoregressive tensor network states (NATNSs)

$$\psi_{\sigma_1 \sigma_2 \dots \sigma_N} = \sum_{m_1, m_2, \dots} \psi(\sigma_1, m_1) \psi(\sigma_2, m_2 | m_1, \sigma_1) \cdots \psi(\sigma_N | m_{N-1}, \sigma_1, \sigma_2, \dots), \quad (9.3)$$

where each conditional probability amplitude over the physical and virtual degrees of freedom is parameterized and represented as part of the neural networks. The normalization and direct sampling can be ensured by imposing the condition that

$$\sum_{\sigma_i, m_i} \psi(\sigma_i, m_i | m_{i-1}, \sigma_1, \dots, \sigma_{i-1}) \psi^*(\sigma_i, m_i | m'_{i-1}, \sigma_1, \dots, \sigma_{i-1}) = \delta_{m_{i-1}, m'_{i-1}}. \quad (9.4)$$

A slightly different idea to combine NAQSs and TNSs has been proposed very recently [356], where a sum of TNS and NATNS is considered. Moreover, the normalization is not enforced by the isometric condition but by explicit normalization. The work [356] gives state-of-the-art result for computing 2D J1-J2 ground state energies. It would be interesting for future work to compare these two different approaches.

- **Infinite systems:** One unexplored direction is to study infinite translationally-invariant systems with NQS. Related ideas have been explored, including iterative

retraining [357] and studying the connection of the MPS and the recurrent neural network (RNN) [358]. But the last missing piece is the formalism of the fixed point of the RNN functions [359], which may enable the study of infinite systems with NQS. If such formalism worked out to be valid, an interesting question would be to study the critical states in the thermodynamic limit by increasing the widths or depths of the networks, similar to the finite entanglement scaling of MPSs [194–196].

- **Open questions:** Finally, it is fair to say the question remains open, *What determines the complexity of neural networks in representing quantum many-body states?* Is it more challenging to represent the ground state at a critical point?? Concerning global quench dynamics, what controls the growth rate of the required parameters? Perhaps a systematic study of potential factors including, finite energy density, entanglement growth, proximity to integrable models, and others may provide us with clues. Lastly, more broadly speaking, a challenging question is *could we prove the generic classical hardness of simulating dynamics of chaotic quantum Hamiltonian?*

### 9.3. Part II: Quantum simulation with sequential quantum circuits

#### Summary:

In Part II, we turned our attention to quantum algorithms. It turns out that isoTNSs discussed in the Part I of the thesis are equivalent to sequential quantum circuits [169, 170, 288]. In Chapter 6, we first discussed the equivalence between isoTNSs and sequential circuits with multi-site unitaries and introduced the sparse sequential circuits ansätze, consisting of two-site unitaries approximating multi-site unitaries. The sparse sequential circuits ansätze is equivalent to the known quantum MPS (qMPS) ansatz in 1D, and more broadly, it belongs to the class of quantum circuit tensor network (QCTN) ansätze [212]. We showed that the sparse sequential circuits ansätze compared to the sequential quantum circuits (or isoTNSs) require exponentially fewer parameters to represent the time-evolved states. We observed this scaling for finite systems in Chapter 7 and for infinite systems in Chapter 8. The framework proposed in Chapter 8 draws inspiration from tensor network algorithms, offering essential building blocks for implementing quantum algorithms for infinite systems on a quantum device with a finite number of qubits. Throughout this part, we also discussed hybrid quantum-classical algorithms for simulating the time evolution of finite and infinite 1D systems.

#### Outlook:

While our work presents a first step towards understanding the scaling of uniform sequential circuits representing time-evolved states, more careful scaling analysis [331] on uniform sequential circuits is still yet to be explored. Moreover, representing environments, i.e., the fixed point of the transfer matrix, requires a number of parameters scaling quadratically with the evolved time. The nature of this scaling also requires further study. Given the equivalence of isoTNSs and sequential circuits and our discussion over 2D isoTNSs, it is natural to consider generalizing the (sparse) sequential circuits to 2D. Indeed such an idea has been considered recently [155, 156]. It would be challenging but interesting to study the scaling of such ansätze for representing ground state and time-evolved states. Lastly, as the gate and measurement fidelity of quantum devices continues to advance, the long-term goal would be to show that we could simulate longer-time dynamics beyond the reach of classical devices. In the short term, it may be possible that we can explore the time evolution algorithm with noisy intermediate scale quantum (NISQ) devices by combining

the classical shadow method [360], as have been demonstrated for ground state search and variational circuit recompilation [361].

Some open questions remain. Numerous quantum algorithms simulating the dynamics or the ground states of many-body quantum systems have been proposed, yet fewer quantum algorithms have been suggested for preparing excited states or thermal states. The recent development of using isoTNSs to represent thermal states [101] indicates that we can also prepare thermal states with sequential quantum circuits. This leads to a series of compelling questions: *What is the quantum complexity of thermal states? Do we expect a potential exponential quantum advantage in preparing thermal states?* It is possible that we can modify the hybrid quantum-classical algorithm for imaginary time evolution proposed in Chapter 7 for thermal states. If so, *How would such an algorithm compare to other algorithms preparing thermal states* [106–108, 362]?

In conclusion, this ongoing exploration of quantum algorithms continues to shape the future of quantum computing.

# Appendices



# A. Appendix IsoTNS

## A.1. Optimization of isometries

Here, we provide an overview of the optimization problems with isometries. In the most general form, the problem we are concerned with is the following. Given  $W \in C^{m \times n}$ ,  $m \geq n$  and  $f : W \rightarrow f(W) \in \mathbb{R}$ , we want to find the optimal  $W^{\text{opt}}$  leading to extreme of  $f(W^{\text{opt}})$  under the isometry constraint  $W^\dagger W = \mathbb{1}$ . These problems show up commonly in algorithms for isometric tensor networks and quantum circuits.

The simplest case for this type of problem is when  $f$  is a linear function and with  $W$  restricted to be real-valued, i.e. orthonormal matrix. The problem is known as the orthogonal Procrustes problem and has close-formed solutions. We review the solution and the proof of this type of problem in Appendix A.1.1. For general cases where  $f$  is a non-linear function, one could consider to linearize the function and update  $W$  in a similar fashion as in the linear case. This is also known as the Evenbly-Vidal algorithm, which we review in Appendix A.1.2. However, such an algorithm does not converge to the extrema in general. As a result, we review the standard gradient descent methods over isometries [171] used in our previous work [75] and this work in Appendix A.1.3.

### A.1.1. Orthogonal Procrustes problem

The orthogonal Procrustes problem [157] is an optimization problem of finding the orthonormal matrix  $W \in \mathbb{R}^{m \times m}$  which best transforms matrix  $A \in \mathbb{R}^{l \times m}$  to matrix  $B \in \mathbb{R}^{l \times m}$ , that is

$$\arg \min_W \|AW - B\|_F$$

Expanding out the expression, the problem is equivalent to

$$\arg \max_W \text{Tr}[WM]$$

where  $M = B^\dagger A$  and  $M \in \mathbb{R}^{m \times m}$ .

More generally speaking, the optimization problem could be stated as finding the maxima of the function  $f^{\text{linear}} : W \rightarrow f^{\text{linear}}(W) \in \mathbb{R}$ , where  $f^{\text{linear}}$  is a function linear in  $W$ . Such a problem permits an exact solution. We first derive the maximum value of the function  $f$ , and show the solution which gives the maximum value.

We first find out the upper bound for the quantity  $\text{Tr}[WM]$ . Suppose the SVD of  $M$  gives  $M = USV^\dagger$ ,

$$\begin{aligned} \text{Tr}[WM] &= \text{Tr}[WU\sqrt{S}\sqrt{S}V^\dagger] = \text{Tr}[(\sqrt{S}U^\dagger W^\dagger)^\dagger(\sqrt{S}V^\dagger)] \\ &= \langle \sqrt{S}U^\dagger W^\dagger | \sqrt{S}V^\dagger \rangle \end{aligned}$$

Since the matrix inner product induces the Frobenius norm. By the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \text{Tr}[WM] &\leq \|\sqrt{S}U^\dagger W^\dagger\|_F \|\sqrt{S}V^\dagger\|_F = \|\sqrt{S}\|_F \|\sqrt{S}\|_F \\ &= \text{Tr}[S]. \end{aligned}$$

We use the invariance of the Frobenius norm under orthonormal transformation in the first equality. The result suggests that the quantity  $\text{Tr}[WM]$  is upper bounded by  $\text{Tr}[S]$ . At the same time, we see that by choosing orthonormal matrix  $W = VU^\dagger$ , we could have the maximum value  $\text{Tr}[S]$ ,

$$\text{Tr}[WM] = \text{Tr}[VU^\dagger USV^\dagger] = \text{Tr}[S] \quad (\text{A.1})$$

Therefore, the solution to the optimization problem is given by  $W^{\text{opt}} = VU^\dagger$ .

The generalized version of the problem consists of a matrix  $M$  of dimension  $(n, m)$ ,  $m \geq n$ , which could be complex-valued,  $M \in \mathbb{C}^{n \times m}$ . Instead of optimizing over an orthonormal matrix, we are now looking for an isometry  $W \in \mathbb{C}^{m \times n}$  which maximizes the absolute value  $|\text{Tr}[WM]|$ . Note that this is equivalent to maximizing  $\text{Re}[\text{Tr}[WM]]$  since one can always absorb the phase factor inside the isometry. A similar derivation from the above holds. The solution is then given as  $\tilde{W}^{\text{opt}} = \tilde{V}U^\dagger$  from the reduced SVD  $M = US\tilde{V}^\dagger$ , where  $U, S, \tilde{V}^\dagger$  are of dimension  $(n, n)$ ,  $(n, n)$ ,  $(n, m)$ .  $\tilde{V}$  is now isometry. That is  $\tilde{V}^\dagger\tilde{V} = I$ , and  $\tilde{V}\tilde{V}^\dagger = P_n$ .

Note that with a matrix  $M$  of dimension  $(n, m)$ ,  $m > n$ , the solution is an isometry of dimension  $(m, n)$ . That is, there is a fixed direction for the isometry tensor. In some cases in the isoTNSs algorithms, we require the isometry tensor to be in a different direction. To satisfy the isometric condition needed in the algorithm, one must first truncate the surrounding tensors to having dimensions  $n = m$  and then solve for the unitary matrix  $W$ .

### A.1.2. Evenbly-Vidal algorithm

In general, the optimization problems with isometries are nonlinear, such as the disentangling problem in the tripartite decomposition or finding the ground state with MERA. The optimization problem then is to find the isometry matrix  $W \in \mathbb{C}^{m \times n}$ ,  $m \geq n$ , which minimizes the function  $f : W \rightarrow f(W) \in \mathbb{R}$ . There is no exact solution in general. It was proposed by Evenbly and Vidal [158, 159] to linearize the function  $f(W)$  and apply the exact solution from the previous section as an iterative update.

The idea of linearizing the function is to keep all the tensors fixed except the one being optimized. One could rewrite the function as,

$$f(W) = \text{Tr}[WE_W] + \text{constant} \quad (\text{A.2})$$

where  $E_W$  is the environment tensor of  $W$  which in general may also depend on  $W$ . Evenbly and Vidal proposed to update the isometry  $W \leftarrow W' = VU^\dagger$  by treating  $E_W$  as if it were independent of  $W$  and  $E_W = U\Sigma V^\dagger$ . The algorithm continues iteratively until convergence.

The algorithm has been generalized to cases where the environment tensor cannot be written easily as a tensor network [175]. Instead of obtaining the environment tensor  $E_W$  by tensor-network contraction, one can compute the derivative with respect to  $W$  i.e.,  $\frac{\partial f}{\partial W}$ . The algorithm consists of iterative update of the isometry  $W \leftarrow W' = VU^\dagger$  until convergence, where  $\frac{\partial f}{\partial W} = U\Sigma V^\dagger$ .

This algorithm could be viewed as a first-order optimization algorithm with the connection given in [174, 175]. The algorithm converges to the optimal point only for restricted cases. The algorithm converges for a negative (positive) definite quadratic form, which includes examples such as the entanglement renormalization [175] and minimizing Rényi- $\alpha$  entropy with  $\alpha = 2$ . It is observed that it also converges for cases with  $\alpha > 1$ .

### A.1.3. Gradient descent algorithm

Gradient descent algorithms are iterative optimization algorithms finding the local minimum given a differentiable function  $f$ . Assuming a Euclidean geometry, a simple version



## A.2. Weighted trace gauge, internal correlation, and the corner double line tensors

of the gradient descent algorithm updates the parameters  $W$  with,

$$W \leftarrow W' = W - \gamma \times \frac{\partial f}{\partial W^*}$$

where  $\gamma$  is known as the step size and is determined by the line-search procedure or other prescribed procedures.

To apply the gradient descent algorithm to problems with isometry constraints, one can consider modifying the update with projection, i.e.,

$$\begin{aligned} U\Sigma V^\dagger &= W - \gamma \times \frac{\partial f}{\partial W^*} \\ W &\leftarrow UV^\dagger. \end{aligned}$$

A better way to adapt to the isometry constraint is to consider the Riemannian optimization approach [171] on the Stiefel manifold with Euclidean metric [363]. Such an approach has recently been reintroduced for isometric tensor networks and quantum circuits [174, 175]. The gradient is defined as the projection of partial derivative  $\Gamma_W = \frac{\partial f}{\partial W^*}$  onto the tangent space  $T_W$  and is given by

$$\nabla f = \Gamma_W - \frac{1}{2}W(W^\dagger\Gamma_W + \Gamma_W^\dagger W) \quad (\text{A.3})$$

Then the update is given by moving in the gradient direction along the geodesics with step size  $\gamma$ ,

$$W \leftarrow e^{-\gamma\nabla f}W. \quad (\text{A.4})$$

One may also consider generalized approaches not following the geodesics but retraction and updating the isometry  $W$  by The Cayley transform. These approaches are equivalent to second order [172].

In addition, one can combine the Riemannian gradient descent with various different first-order gradient-based optimization methods [175, 179, 364]. In this paper and our previous work on isoTNSs [75], we have used the Riemannian non-linear conjugate gradient algorithm [180].

## A.2. Weighted trace gauge, internal correlation, and the corner double line tensors

Here, we review the definition of weighted trace gauge (WTG) condition and cycle entropy  $S_{\text{cycle}}$  quantifying the internal correlation introduced in [167]. We consider a tensor network state  $|\psi\rangle$ , which in general includes bond matrices  $\sigma$  on the virtual leg between two tensors. The bond environment  $\gamma_{i'j'}^{ij}$  is defined through the contraction of  $\langle\psi|\psi\rangle$  leaving out the corresponding bond matrix  $\sigma$  and its complex conjugation, where the indices  $ij, i'j'$  are the corresponding bond indices. The left and right boundary matrices are defined as  $(\rho_L)_{i'}^i = \sum_{k,j,j'} \sigma_{kj} \sigma_{kj'} \gamma_{i'j'}^{ij}$ , and  $(\rho_R)_{j'}^j = \sum_{k,i,i'} \sigma_{ik} \sigma_{i'k} \gamma_{i'j'}^{ij}$ . The WTG is the gauge choice over the bond such that the resulting left and right boundary matrices  $\rho_L$  and  $\rho_R$  are proportional to the identity operator and the bond matrix is diagonal and positive and has elements in descending magnitude. An algorithm to find the WTG is proposed in [167]. For an acyclic tensor network, the WTG is equivalent to the standard canonical form.

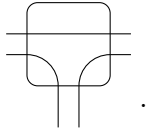
For an acyclic tensor network, a bond is a ‘‘bridge’’ if by cutting the bond the tensor network becomes bipartite. As a result, the bond environment factorizes,  $\gamma_{i'j'}^{ij} = (\gamma_R)_{i'}^i (\gamma_L)_{j'}^j$ , when the bond is a bridge. To quantify the amount of internal correlation over a bond, the cycle entropy  $S_{\text{cycle}}$  is defined as follows,

$$S_{\text{cycle}} = - \sum_i \tilde{\lambda}_i \log \tilde{\lambda}_i \quad (\text{A.5})$$

## Appendix A. Appendix IsoTNS

where  $\tilde{\lambda}_i = |\lambda_i|/(\sum_i |\lambda_i|)$  is the normalized eigenvalue of the transfer operator  $(\sigma \otimes \sigma)\gamma$  formed by contracting the tensor product of the bond matrices to the bond environment. The definition of cycle entropy is chosen such that it is gauge-invariant and is zero if the underlying bond is a bridge. It is also invariant under the unitary transformation acting on the physical degree of freedom as this does not change the bond matrix  $\sigma$  and the bond environment  $\gamma$ . For an isoTNS, we see for any chosen bond, the corresponding bond environment always factorizes due to isometric conditions. Therefore, isoTNSs have no internal correlation inside the tensor network. The alternative way to see that isoTNSs have zero internal correlation is based on the property that the bond environment is invariant under unitary transformation acting on the physical degree of freedom. The isoTNSs have zero cycle entropy because the product states have zero cycle entropy, and isoTNSs are local unitary transformations from product states.

One example of tensor network having internal correlation is TNS consisting of corner double line (CDL) tensors, which has the form



Each line in the tensor is a Kronecker-delta  $\delta_{ij}$  of dimension  $d$ . Taking this CDL tensor as an example, we will gain intuition on why isoTNS has no internal correlation.

The CDL tensor can be viewed as the left or right isometric form up to a normalization factor as shown below.

$$\begin{aligned} \text{CDL with left loop} &= d \times \left( \text{left arc} \right), \\ \text{CDL with right loop} &= d \times \left( \text{right arc} \right). \end{aligned}$$

Consider a state defined by the CDL tensors in a loop over four sites as in [167].

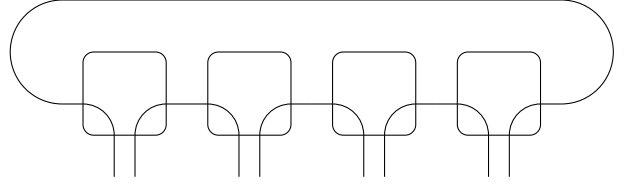
$$\text{Loop of 4 CDL tensors} \propto \text{Chain of 4 square tensors} \quad (\text{A.6})$$

The representation can be viewed as a periodic MPS or a generic 2D TNS. We can also view it as an isoTNS that has an isometric direction forming a loop and has no orthogonality center. This is, in fact, an example of invalid isoTNS representation, which violates our assumption that the isometric direction of isoTNSs does not form a loop and must have exactly one orthogonality center. We will show in the following valid isoTNS representations of the same state, satisfying our assumption.

Firstly, we notice that the state defined in Eq. (A.6) has redundancy in the tensor

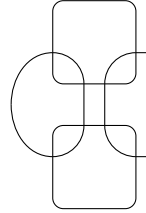
## A.2. Weighted trace gauge, internal correlation, and the corner double line tensors

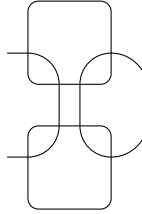
network representation and can be rewritten as

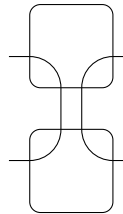

(A.7)

The tensor network states in Eq. (A.6) and in Eq. (A.7) are equivalent, but the tensor network in Eq. (A.6) has bond dimension  $d^2$  instead of  $d$  due to the internal correlation. When using generic TNS describing the state, this is often an issue and is hard to diagnose in tensor renormalization group [125] algorithms for generic TNS. Recently, there are proposals for removing this redundancy locally [167, 365].

The tensor in Eq. (A.7) can be normalized and identified as a tensor in four different isometric forms:

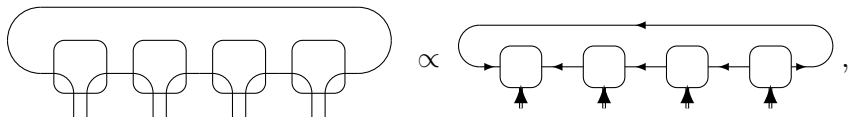

 $= d \times \left( \right),$ 
(A.8)


 $= d \times \left. \right),$ 
(A.9)


 $= \left. \right) \left( \right),$ 
(A.10)


 $= d^2.$ 
(A.11)

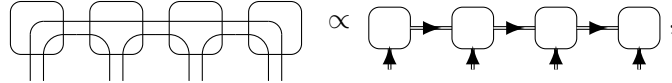
Using the identities shown above, we can rewrite the state in Eq. (A.7) as a valid isoTNS,


(A.12)

where the orthogonality center is on the first site as in Eq. (A.11) and the tensor on the last site is as in Eq. (A.10). Note that this is a valid isoTNS representation, i.e., the isoTNS

has no loop in the isometric direction and has exactly one orthogonality center. In the valid isoTNS representation of the state, there is no internal correlation.

The valid isoTNS representation of the state, however, is not unique. In fact, we can also rewrite the tensor network in Eq. (A.7) in the following form,



$$\text{Diagram} \propto \text{Diagram}, \quad (\text{A.13})$$

which permits another valid isoTNS representation. We notice that this isoTNS representation has bond dimension  $d^2$ , which comes from the price of encoding the correlation from the first site to the last site.

Although the isoTNS representation of the state is not unique and might be subject to the growth of bond dimension due to encoding long-range correlation, under this construction, it is not possible to add additional redundancy, i.e., internal correlation, to the valid isoTNS representation without violating the assumption or the isometric condition. This is different from the generic tensor network as in Eq. (A.6), which in principle, can have arbitrary bond dimension growth due to the internal correlation.

The state consisting of CDL tensors shows an illustrative example of why isoTNS representation has zero cycle entropy. It might be important to develop an algorithm to understand and distinguish between the representation in Eq. (A.12) and that in Eq. (A.13).

### A.3. Extra Data

In Fig. A.1, we show the benchmark results on repeating the MM over a  $D = \eta = 2$  isoTNS representing the ground state of the TFI model with  $g = 3.0$  on an  $11 \times 11$  square lattice. We observe that for MM using  $D' = \eta' = 2$ , we can reach an approximate fixed point where the fidelity almost remains the same. Interestingly, for MM with  $D' = \eta' = 4$ , we find better results for the initial sweeps but do not find an approximate fixed point.

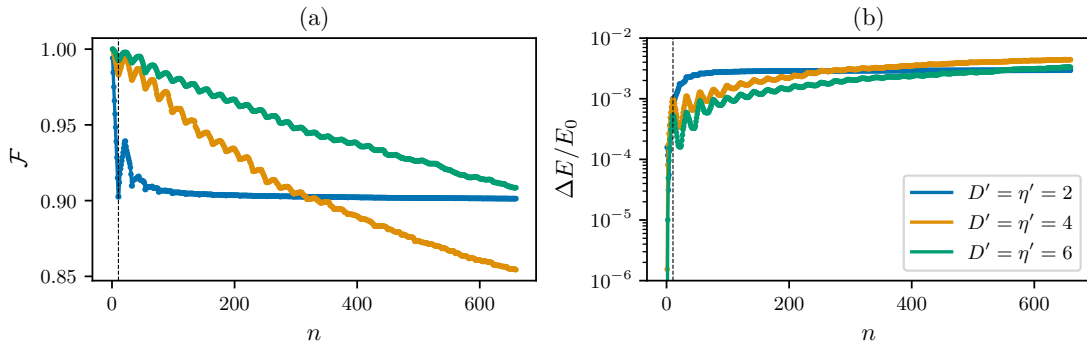


Figure A.1.: Benchmark on repeating the MM over a  $D = \eta = 2$  isoTNS representing the ground state of the TFI model with  $g = 3.0$  on an  $11 \times 11$  square lattice. (a) The fidelity  $\mathcal{F}$  between the original state  $|\Psi_0\rangle$  and the state  $|\Psi_n\rangle$  after  $n$  MM. (b) The relative energy difference of the state  $|\Psi_n\rangle$  comparing to the original state. We perform 30 left-right sweeps, which is in total  $30 \times (2L - 1) = 630$  MM. The MMs are carried out in different bond dimensions  $D', \eta'$  plotted in different colors. The solid circle and the cross mark the data obtained with variational MM and without variational MM. The fidelity and the energy are measured by bMPO method with  $D_{\text{bMPS}} = 4\eta'^2$ .

We take the approximate fixed point wavefunction of  $D' = \eta' = 2$  in Fig. A.1 after 30 left-right sweeps and compute the connected correlation functions  $\langle \sigma_{i,5}^z \sigma_{5,5}^z \rangle_c$  and  $\langle \sigma_{i,5}^x \sigma_{5,5}^x \rangle_c$

along the horizontal line across the center of the lattice. We plot the comparison with the result obtained with the ground state wavefunction in Fig. A.2. The result suggests the correlation functions remain unchanged under multiple MMs.

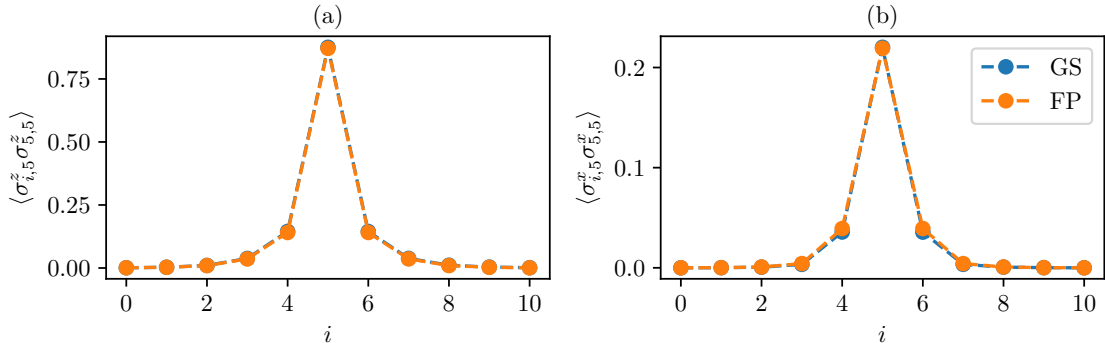


Figure A.2.: The comparison of the connected correlation functions (a)  $\langle \sigma_{i,5}^z \sigma_{5,5}^z \rangle_c$  and (b)  $\langle \sigma_{i,5}^x \sigma_{5,5}^x \rangle_c$ . The approximate fixed point wavefunction (FP) of  $D' = \eta' = 2$  is obtained from the 30 left-right sweeps as shown in Fig. A.1. The ground state wavefunction (GS) is the isoTNS obtained from DMRG<sup>2</sup> and is also the initial state for the repeating MM sweep.

We list the ground state energy for Kitaev honeycomb mode obtained through DMRG<sup>2</sup> with 2D isoTNSs in Table A.1.

Table A.1.: Table for energy per site  $E_0/N$  of Kitaev honeycomb model at isotropic point  $J_x = J_y = J_z = 1$ .

System Size <sup>1</sup> ( $L_x, L_y$ )	Exact	DMRG $\chi = 512$	DMRG $\chi = 1024$	isoTNS $D = 4, \eta = 8$	isoTNS $D = 6, \eta = 12$
(5, 5)	-0.71402401	-0.71362916	-0.71401082	-0.70689939	
(7, 7)	-0.73416737	-0.72593548	-0.73039993	-0.72696637	-0.72987357
(11, 11)	-0.75303346	-0.72405548	-0.72808531	-0.74448479	-0.74682331



## B. Appendix inf IsoTNS

### B.1. Proof of Eq. (4.18)

Recall  $A$  is an isometry such that  $A^\dagger A = \mathbb{1}$  and  $AA^\dagger = \mathcal{P}$ . As a result, we have the following identity:

$$\begin{aligned} \|\Psi\rangle - A|\Phi\rangle\|^2 &= \|\Psi\rangle - AA^\dagger|\Psi\rangle\|^2 \\ &\quad + \|A^\dagger|\Psi\rangle - |\Phi\rangle\|^2. \end{aligned} \quad (\text{B.1})$$

This identity introduces the unnormalized intermediate states  $A^\dagger|\Psi\rangle$  and invites an interpretation of the MM error as the sum of a projection error and an MPS truncation error, which are respectively the first term and the second term in Eq. (B.1).

As we prove below, for a uniform finite system of size  $L$ , to the first order in the errors, both errors are proportional to  $L$ :

$$\begin{aligned} \|\Psi\rangle - AA^\dagger|\Psi\rangle\|^2 &\approx [1 - \lambda_1(T_{A^\dagger\Psi:A^\dagger\Psi})]L \\ \|A^\dagger|\Psi\rangle - |\Phi\rangle\|^2 &\approx [1 - (\lambda_1(T_{\widetilde{A^\dagger\Psi:\Phi}}))]^2 L, \end{aligned} \quad (\text{B.2})$$

where  $\widetilde{A^\dagger|\Psi\rangle} \propto A^\dagger|\Psi\rangle$  is the normalized state<sup>1</sup>. Recall the definitions as in Eq. (4.19):

$$\begin{aligned} \varepsilon_p &\equiv 1 - \lambda_1(T_{A^\dagger\Psi:A^\dagger\Psi}), \\ \varepsilon_t &\equiv 1 - (\lambda_1(T_{\widetilde{A^\dagger\Psi:\Phi}}))^2. \end{aligned}$$

Evidently, the total error density in Eq. (4.16) can be decomposed as

$$\varepsilon = \varepsilon_p + \varepsilon_t + O(\varepsilon_p^2, \varepsilon_t^2, \varepsilon_p \varepsilon_t), \quad (\text{B.3})$$

which is Eq. (4.18).

To complete this appendix, we now give the derivation of Eq. (B.2). The first term in Eq. (B.1) is due to the projection  $AA^\dagger$ :

$$\begin{aligned} \|\Psi\rangle - AA^\dagger|\Psi\rangle\|^2 &= 1 - \|A^\dagger|\Psi\rangle\text{rVert}^2 \\ &\equiv 1 - (1 - \varepsilon_p)^L \approx \varepsilon_p L. \end{aligned} \quad (\text{B.4})$$

The second source of  $\varepsilon$  comes from the truncation error of representing the unnormalized state  $A^\dagger|\Psi\rangle$  with the MPS  $|\Phi\rangle$ :

$$\begin{aligned} \|A^\dagger|\Psi\rangle - |\Phi\rangle\text{rVert}^2 &= 1 + \|A^\dagger|\Psi\rangle\|^2 - 2\text{Re}\langle\Phi|A^\dagger|\Psi\rangle \\ &= 2 - \varepsilon_p L - 2\text{Re}\langle\Phi|\frac{A^\dagger|\Psi\rangle}{(\sqrt{1 - \varepsilon_p})^L}(\sqrt{1 - \varepsilon_p})^L \\ &\equiv 2 - \varepsilon_p L - 2\text{Re}\langle\Phi|\widetilde{A^\dagger|\Psi\rangle}(\sqrt{1 - \varepsilon_p})^L, \end{aligned} \quad (\text{B.5})$$

<sup>1</sup>Note that it is wrong to use  $T_{A^\dagger\Psi:\Lambda}$  in Eq. (B.2) as the states making up the transfer matrix are not properly normalized

## Appendix B. Appendix inf IsoTNS

where we defined  $\widetilde{A^\dagger|\Psi\rangle} = A^\dagger|\Psi\rangle/(\sqrt{1-\varepsilon_p})^L$  as the normalized state. Analogously to Eq. (4.17),

$$\|\widetilde{|\Phi\rangle} - \widetilde{A^\dagger|\Psi\rangle}\|^2 = 2 - 2\text{Re}\langle\widetilde{|\Phi\rangle}|\widetilde{A^\dagger|\Psi\rangle}\rangle \approx \varepsilon_t L. \quad (\text{B.6})$$

Thus, by putting everything together, we have

$$\begin{aligned} \|A^\dagger|\Psi\rangle - |\Phi\rangle\|^2 &\approx 2 - \varepsilon_p L - (2 - \varepsilon_t L) \left(1 - \frac{\varepsilon_p L}{2}\right) \\ &= \varepsilon_t L. \end{aligned} \quad (\text{B.7})$$

## B.2. Isometric filling of $A$

Let  $a$  denote the tensor making up  $A$ . Label its indices as below:

$$\begin{array}{c} i_3 \\ \uparrow \\ (a) \\ \leftarrow i_0 \quad \rightarrow i_1 \quad \cdot \\ \downarrow \\ i_2 \end{array} \quad (\text{B.8})$$

To enlarge the bond dimension  $\chi_v = \dim(i_2) = \dim(i_3)$  to  $\chi'_v$  while keeping the operator that  $A$  represents invariant and the isometric condition intact, one groups  $i_2 i_0$  and  $i_3 i_1$  respectively as the row and column index of the isometric matrix  $a$ .  $i_2$  and  $i_3$  are the “slow” index of their respective combined indices. Here we assume  $\dim(i_0) = \dim(i_1)$ , and thus  $a$  is square. To enlarge the bond dimensions, one first zero-pads on the index  $i_2$  and then adds orthogonal columns on index  $i_3$ :

$$[a] \rightarrow \begin{bmatrix} a \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} a & 0 \\ 0 & a^\perp \end{bmatrix}, \quad (\text{B.9})$$

where  $a^\perp$  is an arbitrary unitary matrix with  $(\chi'_v - \chi_v) \dim(i_0)$  number of rows and columns. Thus, when  $\dim(i_0) = \dim(i_1)$ , the result of isometric filling is  $a \rightarrow a' = a \oplus a^\perp$ .

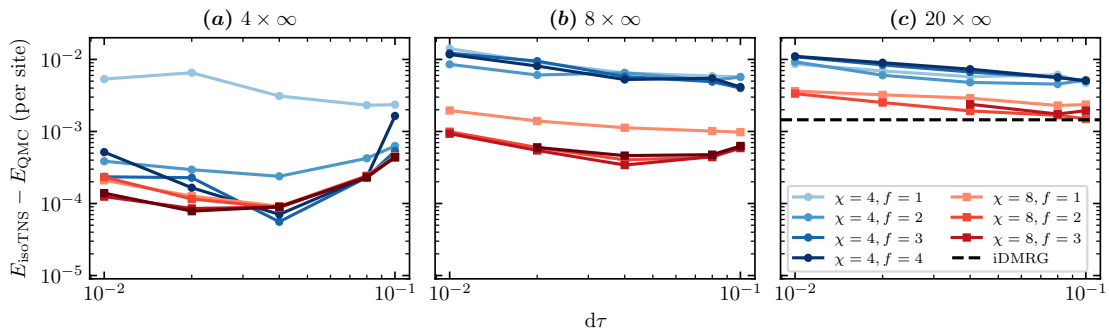


Figure B.1.: Ground state energies achieved with iTEBD<sup>2</sup> for critical  $g = 3.04438$  2D TFI on  $L_x = 4, 8, 20$  strip. An intermediate  $d\tau$  and  $f$  yield the best energy, while  $\chi = 8$  outperforms  $\chi = 4$ . We compare the isoTNS energies against essentially exact energies from quantum Monte Carlo (QMC) extrapolated from strips of finite length. As a comparison, the dashed line is the result of an iDMRG calculation with bond dimension  $\chi = 512$ . For  $L_x = 4, 8$ , the iDMRG result is below the bottom axis of the plot.



### B.3. Numerical results for $g = 3.04438$

Here we present numerics at  $g = g_C^{2D} = 3.04438$ , the critical transverse field for the two-dimensional TFI in the thermodynamic limit. Close to the critical point, we expect this model on finite-width strips to be more difficult to capture by an iMPS due to increased entanglement.

We use the iTEBD<sup>2</sup> algorithm to search for the ground state of the  $g = 3.04438$  2D TFI. Again we compare isoTNS energies evaluated by IMM against iDMRG energies for  $\chi = 512$ . Results for  $L_x = 4$ ,  $L_x = 8$ , and  $L_x = 20$  infinite strips are shown in Fig. B.1, where again we find an intermediate  $dt$ , which balances iTEBD and IMM errors, leads to the optimal energies.



# C. Appendix Neural Network

## C.1. Network architectures

Feed-forward neural networks are functions consisting of alternating affine maps and elementwise non-linear functions. The input of the network is the one-hot encoding of the computational basis. That is

$$X : \boldsymbol{\sigma} \in \{1, 2, \dots, M\}^N \longrightarrow X(\boldsymbol{\sigma}) \in \mathbb{R}^{M \times N} \quad (\text{C.1})$$

$$X(\boldsymbol{\sigma})_{i,j} = \begin{cases} 1, & \text{if } i = \sigma_j \\ 0, & \text{else} \end{cases} \quad (\text{C.2})$$

Fully-connected neural networks (NNs) are defined by a recursion relation at each layer

$$\mathbf{z}^{(l)} = g(W^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}), \quad (\text{C.3})$$

where the weight matrix  $W^{(l)}$  and bias vector  $\mathbf{b}^{(l)}$  at each layer ( $l$ ) together are the variational parameters. The  $g$  is the elementwise non-linear function, a.k.a activation function. At the first layer, the input is first flattened as a vector, i.e.  $z^{(0)} = \text{Vec}(X(\boldsymbol{\sigma}))$ .

We consider NNs with one- and two-hidden layers and denote them as NN-1 and NN-2. The number of neurons are  $(N, \alpha N, 4N)$  for NN-1 and  $(N, \alpha N, \alpha N, 4N)$  for NN-2 respectively. The real-valued output is of dimension  $4N$  to encode the  $M = 2$  different complex-valued conditional probability amplitude on  $N$  sites.

Similarly, convolutional neural networks (CNNs) are defined by a recursion relation at each layer

$$z^{(l)} = g(W^{(l)} * z^{(l-1)} + b^{(l)}), \quad (\text{C.4})$$

where now we replace the matrix multiplication with the convolution operation  $*$ . To be explicit, we can write it as

$$z_{c,j}^{(l)} = g\left(\sum_{c'=1}^{\alpha N} \sum_{k=1}^{N_F} W_{c,c',k}^{(l)} z_{c',(j+k-N_F)}^{(l-1)} + b_{c,j}^{(l)}\right), \quad (\text{C.5})$$

where  $N_F$  denotes the filter size. We consider two different setups of global connectivity  $N_F = N$  and local connectivity  $N_F < N$ . We pad the input with zeros to keep the width of the network fixed for simplicity and only change the number of channels at each layer. For shallow networks, we consider CNNs up to three hidden layers and denote them as CNN-[depth]. The number of channels are  $(2, \alpha, 4)$  for CNN-1,  $(2, \alpha, \alpha, 4)$  for CNN-2, and  $(2, \alpha, \alpha, \alpha, 4)$  for CNN-3. The output layer is with 4 channels to represent the  $M = 2$  different complex-valued conditional probability amplitudes. All shallow NNs and CNNs have ReLU as an activation function, i.e.  $g(z) = \max(0, z)$ .

To turn the networks into networks satisfying autoregressive properties, the connectivity is restricted by masking the weights. For example, the type-A masking makes sure the  $z_i^{(l)}$  does not depend on  $z_i^{(l-1)}$  and component afterwards by restricting the sum to  $z_i^{(l)} = g(\sum_{j < i} W_{i,j}^{(l)} z_j^{(l-1)} + b_i^{(l)})$ . The type-B masking makes sure the  $z_i^{(l)}$  does not depend on  $z_{i+1}^{(l-1)}$  and component afterwards by restricting the sum to  $z_i^{(l)} = g(\sum_{j \leq i} W_{i,j}^{(l)} z_j^{(l-1)} +$

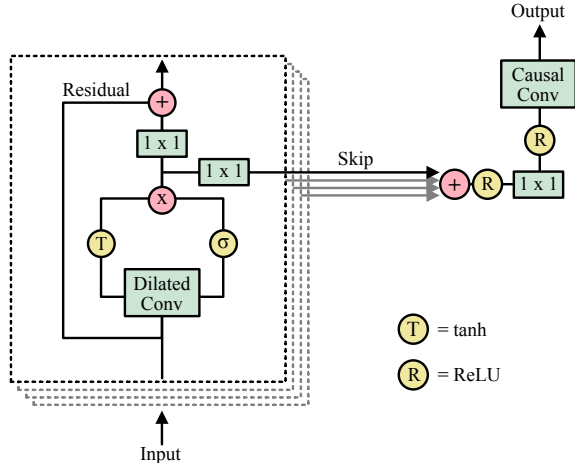


Figure C.1.: The building block and network architecture of WaveNet-like network adapted from [263]. The building block is indicated by the dotted box. We repeat the building block to increase the depth. Except for the first and the last block, the output of the residual connection is the input of the next following block. The Gated-CNN is a simplification of the WaveNet without skip-connections and the following layers. Additionally, it uses convolution instead of dilated convolution.

$b_i^{(l)}$ ). Combining the type-A masking at the first layer and type-B masking at all layers afterward, we ensure the autoregressive properties of the network. For more detail about autoregressive models, see Appendix C.2 and see Fig. 5.1 for a graphical illustration.

We consider two different deep neural networks. In Fig. C.1, we show the network adapted from the WaveNet [263]. The WaveNet uses dilated convolution, which is the convolution applied over a larger area by skipping input values. The stacked dilated convolutions with exponential growing dilation factors enable an exponential growing receptive field (causal cone). Similar to the WaveNet, we double the dilation factors at each layer and repeat again once it reaches the system size, e.g.  $1, 2, 4, 8, 16, 1, 2, 4, \dots$  for  $N = 20$ .

Following the dilated convolution is the gated activation unit [268],

$$z = \tanh(W^f * x) \odot \sigma(W^g * x) \quad (\text{C.6})$$

where  $\odot$  is the element-wise multiplication. The  $\sigma(\cdot)$  is the sigmoid activation function and should not be confused with the local computation basis. The  $W^f$  and  $W^g$  are the weight for the filter and gate, respectively.

After the gated activation unit, there are two special connections. The residual connections [260] are implemented by the  $1 \times 1$  convolution and the summation with the shortcut from the input of the dilated convolution. The output of the residual connection is the input of the next block. The parameterized skip connections are implemented by the  $1 \times 1$  convolution to form shortcuts from each block directly to the end of repeating blocks, which are then summed together with the final output. These “shortcut” connections are key to training deep neural networks.

In addition to the WaveNet, we consider a simple deep CNN, Gated-CNN, where we remove the skip connections and use only convolution instead of the dilated convolution. Without the dilated convolution, the receptive field of a Gated-CNN grows linear with depth. For the receptive field to cover the system size  $N = 20$ , we consider Gated-CNNs with 10 to 20 blocks. As for WaveNets, we consider having 6 to 14 blocks. While WaveNets includes one block of  $1 \times 1$  convolution before the final causal convolution, the output of the Gated-CNN blocks is followed directly by the causal convolution. The design choice is

arbitrary and turns out not to affect the scaling result. Similar to shallow CNNs, the final output of both networks has 4 channels to represent the  $M = 2$  different complex-valued conditional probability amplitudes.

The code implementation based on TensorFlow [271] is available [272] and the hyperparameters and general setup of the optimization can be found in Appendix C.4.

## C.2. Review of neural autoregressive quantum states

A joint probability distribution  $P$  over  $N$  discrete random variables  $x_i$  can be factorized as the multiplication of conditional probabilities.

$$\begin{aligned} P(x_1, x_2, \dots, x_N) &= P(x_1) \times P(x_2|x_1) \times P(x_3|x_1, x_2) \dots \\ &= \prod_{i=1}^N p(x_i|x_1, x_2, \dots, x_{i-1}) \\ &= \prod_{i=1}^N p(x_i|\mathbf{x}_{<i}) \end{aligned}$$

The decomposition imposes an ordering over the random variables and allows for exact sampling. However, the exact representation of conditional probabilities also scales exponentially with the number of random variables. Autoregressive models are parameterized functions approximating conditional probabilities.

One important class is neural autoregressive models [366, 367]. By utilizing neural networks to approximate conditional probabilities, it allows exact and efficient sampling and inference. It has been applied to studying statistical physics in the pioneering work [368] and in [369].

In principle, one can approximate each conditional probability with a different neural network,

$$P_{\text{AR}} = f(x_1) \times g(x_2|x_1) \times h(x_3|x_1, x_2) \times \dots$$

A simpler and common approach is to approximate all  $N$  conditional probability by one neural network with  $N$  inputs and  $N$  outputs,

$$P_{\text{AR}} = f^{[1]}(x_1) \times f^{[2]}(x_2|x_1) \times f^{[3]}(x_3|x_1, x_2) \times \dots$$

The network can no longer be fully connected, but with connectivity satisfying the autoregressive property. That is  $f^{[2]}$  can only depend on input  $x_1$  and  $f^{[3]}$  only on inputs  $x_1, x_2$  and so on. It is common to enforce this by zero-masking the original weights  $w$ . Equivalently, we say we have masked weight multiplication or masked convolution (see Fig. 5.1).

Following the same principle, a probability amplitude can be rewritten as a product of conditional probability amplitudes [370].

$$\Psi_{\sigma_1\sigma_2\dots\sigma_N} = \Psi_{\sigma_1} \Psi_{\sigma_2|\sigma_1} \dots \Psi_{\sigma_N|\sigma_1\dots\sigma_{N-1}} \quad (\text{C.7})$$

Neural autoregressive quantum states (NAQS) [78] approximate the conditional probability amplitudes by neural networks.

$$\Psi_{\sigma_1\sigma_2\dots\sigma_N}^{\text{NAQS}} = f_{\text{NN}}^{[1]}(\sigma_1) f_{\text{NN}}^{[2]}(\sigma_2|\sigma_1) \dots f_{\text{NN}}^{[N]}(\sigma_N|\sigma_1\dots\sigma_{N-1}) \quad (\text{C.8})$$

In the following, we denote  $f_{\text{NN}}^{[i]}$  as  $\psi_i$ .

It is shown that if each conditional probability amplitude is normalized  $\sum_{\sigma_i} |\Psi_{\sigma_i|\sigma_{j<i}}|^2 = 1$ , then the full wavefunction is normalized  $|\Psi_{\sigma_1\sigma_2\dots\sigma_N}|^2 = 1$ . Therefore, we can normalize

the full wavefunction by normalizing each conditional probability amplitude. Suppose the local basis is of dimension  $M$ , each of the  $N$  outputs is actually a complex vector  $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,M})$ , which represents the logarithm of the conditional probability amplitude at site  $i$ .

$$\psi_i(\sigma_i | \sigma_1, \dots, \sigma_{i-1}) = \frac{e^{v_{i,\sigma_i}}}{\sum_{\sigma'_i} |e^{v_{i,\sigma'_i}}|^2} \quad (\text{C.9})$$

gives the normalized conditional probability amplitudes. Furthermore, each  $|\psi_i|^2$  gives a valid conditional probability induced by Born's rule. This means a NAQS contains a neural autoregressive model, which permits efficient and exact sampling.

### C.3. Cost functions and stochastic gradient descent

In this section, we give a brief review of the setup of supervised learning problems and discuss the difference between it and the general optimization problem. We further give an overview of different cost functions for variational states approximation and the resulting properties. The discussion is general and applies to general variational wavefunctions.

Supervised learning is a minimization problem to learn a function mapping from  $X$  to  $Y$ . Given the data  $\{x, y\}^N$ , where the data pair  $\{x, y\} \sim P(x, y)$  are sampled from an unknown probability distribution  $P(x, y)$ , and the parameterized function  $f : x \in X \rightarrow f(x) \in Y$  with parameters  $w$ , we would like to minimize the cost function  $\mathcal{R}(f)$  with respect to parameters  $w$ .  $\mathcal{R}(f)$  is defined as

$$\mathcal{R}(f) = \int \mathcal{L}(f(x; w), y) dP(x, y) \quad (\text{C.10})$$

where  $\mathcal{L}(f(x; w), y)$  is the loss function measuring the difference between  $f(x; w)$  and  $y$ . In general, the expression cannot be evaluated. Instead, the supervised learning framework, a.k.a. empirical risk minimization, relies on minimizing the cost from the sampling (empirical risk)  $\hat{\mathcal{R}}(f)$ ,

$$\hat{\mathcal{R}}(f) = \sum_{\{x, y\} \sim P(x, y)} \mathcal{L}(f(x; w), y). \quad (\text{C.11})$$

One can compute the full batch gradient based on the empirical risk and perform a gradient descent algorithm. In modern applications, often even the empirical risk itself is too expensive to evaluate at all steps given the large size of the dataset. One turns to compute the mini-batch stochastic gradient  $\widehat{\nabla \mathcal{R}} = \sum_i \nabla_w \mathcal{L}(f(x_i; w), y_i)$  where the sum is taken over the mini-batch samples. In terms of deep learning, the supervised learning framework fits nicely with the stochastic gradient descent algorithm.

Another type of optimization problem is to minimize the cost function  $\mathcal{R}$ , which is not in the form as in Eq. (C.10), i.e. an integral or a sum of the loss function  $\mathcal{L}$ . The cost function  $\mathcal{R}$  itself is the thing we want to optimize. In this case, gradient-based algorithms would require computing the gradient  $\nabla_w \mathcal{R}$ . In certain cases, the gradient  $\nabla_w \mathcal{R}$  itself cannot be computed exactly and approximation is required. Moreover, the gradient  $\nabla_w \mathcal{R}$  might contain intractable terms as integral or summation, which can be approximated by sampling. In this case, the sampled "stochastic gradient" has quite different properties compared to the previous one in the setup of supervised learning. More precisely, *the stochastic gradient is unbiased when the expectation value over the sampling of this stochastic gradient is equal to the true gradient*, i.e.

$$\nabla \mathcal{R} = \mathbb{E}[\widehat{\nabla \mathcal{R}}]$$

The stochastic gradient is *biased* if the expectation value of the sampled stochastic gradient does not equal the true gradient. Stochastic gradients in supervised learning setup are

### C.3. Cost functions and stochastic gradient descent

unbiased, while in the general optimization problem they are not. With this in mind, we can distinguish the cost function for variational state approximation by whether it has the form of Eq. (C.10).

In terms of the variational state approximation, given the target wavefunction  $|\Phi\rangle$ , the variational wavefunction  $|\Psi\rangle$  and the dimension of the many-body Hilbert space  $D_{\mathcal{H}}$ . here we review different possible cost functions that work for both unnormalized wavefunctions and normalized wavefunctions introduced in previous works.

In [224, 253], the  $\mathcal{L}_2$  distance of the probability amplitudes  $\mathcal{R}_{\text{MSE}}$  are considered,

$$\begin{aligned}\mathcal{R}_{\text{MSE}} &= \frac{1}{D_{\mathcal{H}}} \sum_i^{D_{\mathcal{H}}} |\Psi(X_i) - \Phi(X_i)|^2 \\ &= \frac{1}{D_{\mathcal{H}}} \sum_i^{D_{\mathcal{H}}} P_{\Phi}(X_i) \frac{|\Psi(X_i) - \Phi(X_i)|^2}{|\Phi(X_i)|^2}.\end{aligned}$$

Similarly, in Ref. 254, the weighted  $\mathcal{L}_2$  distance of log of the probability amplitudes is considered,

$$\mathcal{R}_{\text{weighted MSE log}} = \frac{1}{D_{\mathcal{H}}} \sum_i^{D_{\mathcal{H}}} P_{\Phi}(X_i) \left| \log \Psi(X_i) - \log \Phi(X_i) \right|^2$$

while in [224] a uniform sampling scheme is considered, and in Ref. 253, 254 the samples are taken with respect to  $P_{\Phi}$ . These cost functions are unbiased but with high variance. Furthermore, the problem is posed as a regression problem instead of taking into account the probability nature of the problem.

A different setup is considered in Ref. 255 with the negative log fidelity as the cost function.

$$\begin{aligned}\mathcal{R}_{\text{neg. log fidelity}} &= -\log \left[ \frac{\langle \Psi | \Phi \rangle \langle \Phi | \Psi \rangle}{\langle \Psi | \Psi \rangle \langle \Phi | \Phi \rangle} \right] \\ &= -\log \left[ \frac{\left( \sum_i^{D_{\mathcal{H}}} \Psi^*(X_i) \Phi(X_i) \right) \left( \sum_i^{D_{\mathcal{H}}} \Phi^*(X_i) \Psi(X_i) \right)}{\left( \sum_i^{D_{\mathcal{H}}} \Psi^*(X_i) \Psi(X_i) \right) \left( \sum_i^{D_{\mathcal{H}}} \Phi^*(X_i) \Phi(X_i) \right)} \right]\end{aligned}$$

This cost function and the corresponding gradient are in the form that could be estimated by stochastic sampling. However, the resulting stochastic gradient is biased. In practice, this cost function works well as long as the batch size is taken large enough. In this work, we use directly the negative fidelity as the cost function when optimizing the RBMQS.

$$\begin{aligned}\mathcal{R}_{\text{neg. fidelity}} &= -\frac{\left( \sum_i \Psi_i^* \Phi_i \right) \left( \sum_i \Psi_i \Phi_i^* \right)}{\left( \sum_i \Psi_i^* \Psi_i \right) \left( \sum_i \Phi_i^* \Phi_i \right)} \\ &= -\frac{\mathbb{E}_{i \sim P_{\Phi}} [\Psi_i^* / \Phi_i^*] \mathbb{E}_{i \sim P_{\Phi}} [\Psi_i / \Phi_i]}{\mathbb{E}_{i \sim P_{\Phi}} [|\Psi_i|^2 / |\Phi_i|^2]}\end{aligned}$$

With the stochastic estimate of the fidelity, the biased stochastic gradient is then obtained from the gradient evaluated by the automatic differentiation framework.

In this work, we consider two cost functions that work only for normalized wavefunction, which give an unbiased stochastic gradient with low variance. In Section 5.1.3, we mainly focus on the cost function

$$\mathcal{R}_{\text{joint}} = \mathcal{R}_{\text{KL}} + \mathcal{R}_{\theta}. \quad (\text{C.12})$$

An alternative cost function is the real part of the negative overlap,

$$\begin{aligned}\mathcal{R}_{\text{neg. overlap}} &= - \sum_i^{D_{\mathcal{H}}} \text{Re} [\langle \Phi | X_i \rangle \langle X_i | \Psi \rangle] \\ &= - \sum_i P_{\Phi}(X_i) \text{Re} \left[ \frac{\langle X_i | \Psi \rangle}{\langle X_i | \Phi \rangle} \right]\end{aligned}$$

Minimizing the real part  $\text{Re} [\langle \Phi | \Psi \rangle]$  is equivalent to minimizing  $|\langle \Phi | \Psi \rangle|$  because we can always absorb a phase factor into our complex-valued wavefunction. Minimizing only the real part gives the form of expectation value and again we can apply the stochastic gradient descent method. This, however, cannot apply to unnormalized wavefunction because of the intractable normalization constant.

The fidelity  $\mathcal{F}$  is the square of the overlap,  $\mathcal{F} = |\langle \Phi | \Psi \rangle|^2$ . It may be tempting to maximize the fidelity  $\mathcal{F}$  instead of the sampling estimate of the overlap. However, again we encounter the problem of a biased gradient. Although, the fidelity estimate cannot serve as a cost function. We can still evaluate it as a check for the validity of the final result.

Finally, we show that these two cost functions are equivalent to the first order.

$$\begin{aligned}\mathcal{R}_{\text{neg. overlap}} &= - \sum_i P_{\Phi}(X_i) \text{Re} \left[ \frac{\langle X_i | \Psi \rangle}{\langle X_i | \Phi \rangle} \right] \\ &= - \sum_i P_{\Phi}(X_i) \text{Re} \left[ \frac{e^{r_i^{\Psi} + i\theta_i^{\Psi}}}{e^{r_i^{\Phi} + i\theta_i^{\Phi}}} \right] \\ &= - \sum_i P_{\Phi}(X_i) \text{Re} \left[ e^{(r_i^{\Psi} - r_i^{\Phi}) + i(\theta_i^{\Psi} - \theta_i^{\Phi})} \right] \\ &\approx - \sum_i P_{\Phi}(X_i) \left[ (1 + (r_i^{\Psi} - r_i^{\Phi})) \left( 1 + \frac{(\Delta\theta_i)^2}{2} \right) \right] \\ &\approx \sum_i P_{\Phi}(X_i) \left[ -1 + (r_i^{\Phi} - r_i^{\Psi}) - \frac{(\Delta\theta_i)^2}{2} \right] \\ &\approx -1 + \frac{1}{2}(\mathcal{R}_{\text{KL}} + \mathcal{R}_{\theta})\end{aligned}$$

In practice, we observe slightly different dynamics at the beginning of the learning process.

## C.4. Setup of optimizations

Below, we first discuss the common setup of optimizations for both NAQS and MPS.

- *Optimizer and batch size:* We consider Adam optimizer and a batch size of 512.
- *Convergence criterion:* The convergence criterion is based on monitoring the error in fidelity. We set the criterion to be either if the error were below  $10^{-4}$  or if an average error of the latest 500 batches were higher than the average error of the latest 5000 to 2500 batches. In addition, there is a break condition set by the maximal number of steps.
- *Maximal number of steps:* We set the maximal number of steps to be  $3 \times 10^5$ , which is almost never reached. The typical number of steps is below  $10^5$ .



- *Number of runs*: For each data point, we run the optimization with random initialization 4 times and report the best result.

*Matrix Product States*: The MPS considered are complex-valued. We start out with a fixed learning rate  $10^{-2}$  until convergence. Then, we repeat the procedure with the learning rate decreased to  $10^{-3}$  and  $10^{-4}$ .

Notice that the canonical (isometric) form or normalization of MPS is not enforced during the optimization. In fact, we observed that the result is better than the result enforcing the isometric form using Riemannian optimization [174, 205].

*Neural Network*: All the networks considered are parameterized by real-valued weights and biases. We start out with a fixed learning rate  $10^{-3}$  until the convergence criterion is met. Then, we decrease the learning rate to  $10^{-4}$  and continue the optimization again until the convergence.

Here, we provide the setup of using NetKet to perform state approximation with RBMQS.

*NetKet*:

- *Version*: 2.1
- *Optimizer and batch size*: stochastic reconfiguration and stochastic gradient descent. The batch size is 1000. We take the best result from different step sizes:  $1 \times 10^{-1}$ ,  $3 \times 10^{-2}$ ,  $1 \times 10^{-2}$ .
- *Convergence criterion*: The convergence criterion is based on monitoring the average error in fidelity. We set the criterion to be either if the average error were below  $10^{-4}$  or if average error of the latest 50 batches were close to the average error of the latest 100 to 50 batches with a relative error  $3 \times 10^{-6}$  and absolute error  $3 \times 10^{-7}$ . In addition, there is a break condition set by the maximal number of steps.
- *Maximal number of steps*: We start checking for convergence after 5000 iterations and set a maximum iteration of 10000, which is almost never reached.
- *Number of runs*: For each data point, we run the optimization with random initialization 4 times and report the best result.

## C.5. Additional data

In Fig. C.2, we include the result of MPS approximating the target state by the alternating least square algorithm, which iteratively minimizes the  $\mathcal{L}_2$  distance of the quantum states [31] and yield a slightly better result than SGD optimization with supervised learning as shown in Fig. 5.2.

In Fig. C.3, we include the results with shallow NAQS for periodic boundary condition (PBC) and plot the number of parameters of the network against the obtained accessible time  $t^*$ .

In Fig. C.4, we show the results for WaveNets of width  $\alpha = 8$ .

In Fig. C.5, we plot the errors of the results for Gated-CNNs of width  $\alpha = 12$  and different depths. We see the errors in  $\langle \hat{\sigma}_{L/2}^x \rangle$  expectation value and half-chain entanglement entropy  $S_{\text{vN}}$  follows similar tendency as of the error in fidelity. Interestingly, we observe a peak in both errors at a relatively short time before the network reaches its accessible time  $t^*$ .

In Fig. C.6, we show the results for RBMQS.

In Fig. C.7, we compare the results for state approximation by RBMQS using SGD and exact gradient optimization and the SGD with stochastic reconfiguration from the NetKet [6] implementation.

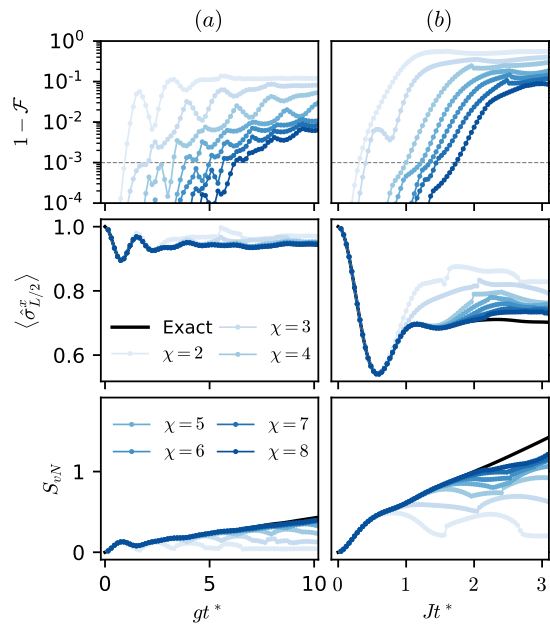


Figure C.2.: MPS with bond dimension  $\chi$  approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The MPS are obtained by iteratively minimizing the  $\mathcal{L}_2$  distance. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in the paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ).

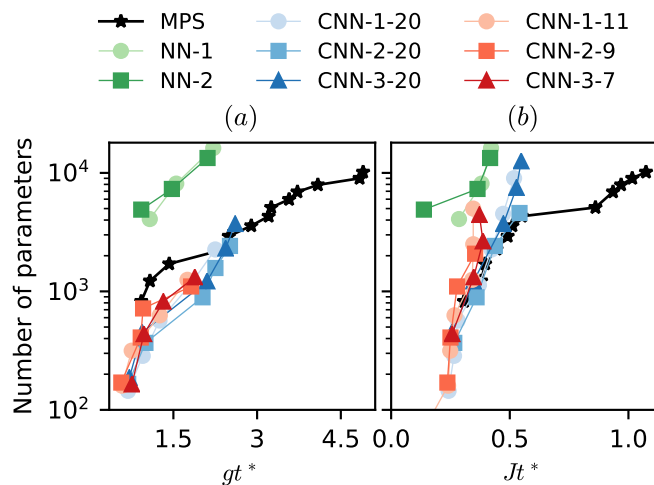


Figure C.3.: The number of parameters required to reach time  $t^*$  for following a quantum quench for different approximation schemes. The quenched Hamiltonian is similar to Eq. (5.9) but with PBC on a chain of length  $N = 20$  with (a) quantum Ising model in paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ). The black line shows the results for MPS. For NAQS, we denote each combination as (network type)-(number of hidden layers)-(filter size). The circles, squares, and triangles represent 1,2,3-layer networks, respectively. The gradient colors describe the same networks of different numbers of layers. NNs (Blue); CNNs with global connectivity (Green-Blue); CNNs with local connectivity (Red).

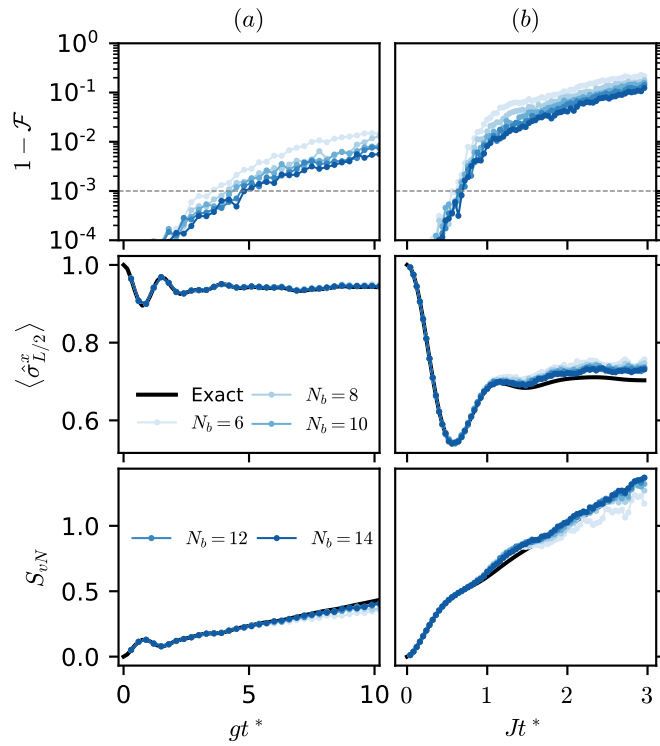


Figure C.4.: WaveNet NAQS of different depths, i.e. number of blocks  $N_b$ , approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The network is of width  $\alpha = 8$ . The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in the paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ).

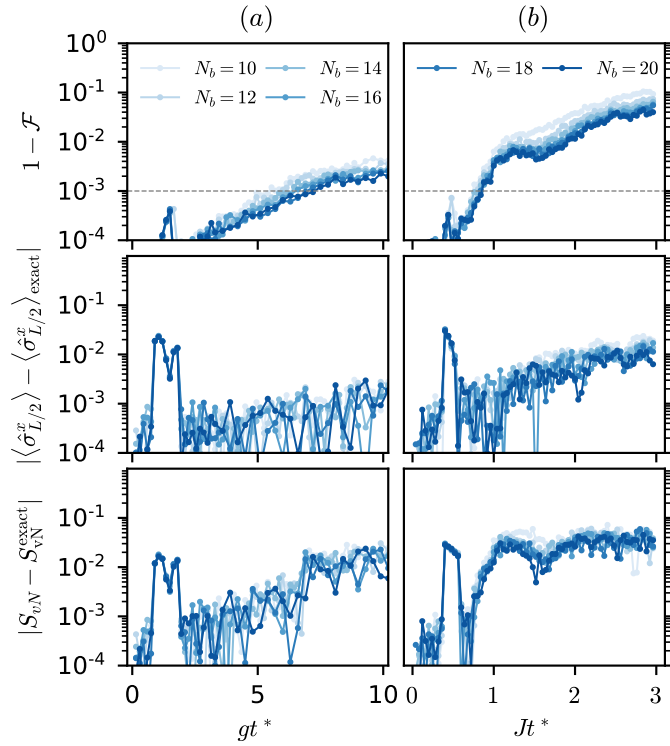


Figure C.5.: The errors of Gated-CNNs NAQS of different depths, i.e. number of blocks  $N_b$ , approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The network is of width  $\alpha = 12$ . The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in the paramagnetic phase with weak longitudinal field ( $g = 3$ ,  $h = 0.1$ ,  $k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1$ ,  $h = 0$ ,  $k = 0.25$ ).

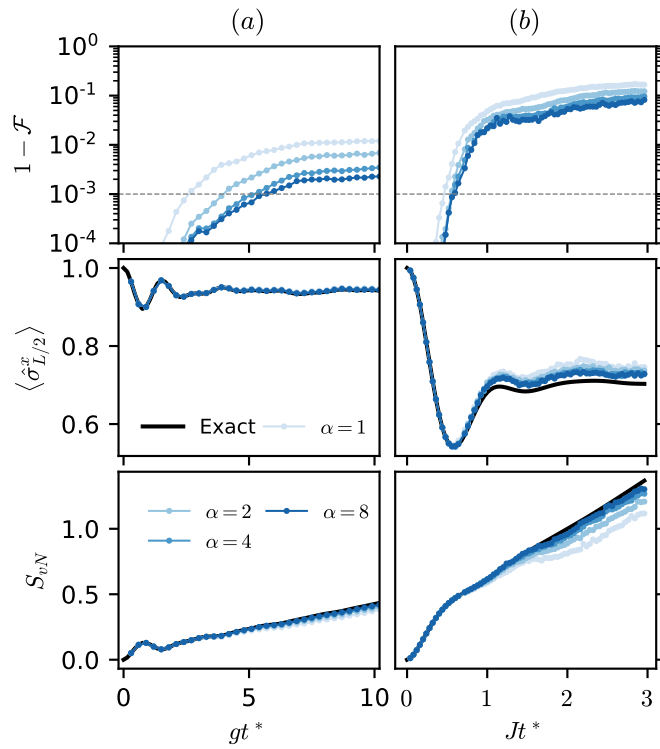


Figure C.6.: RBMQS of different widths, i.e.  $\alpha = N_{\text{hidden}}/N$ , approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 20$  with (a) quantum Ising model in the paramagnetic phase with weak longitudinal field ( $g = 3, h = 0.1, k = 0$ ) and (b) close to a critical point with interacting term ( $g = 1, h = 0, k = 0.25$ ).

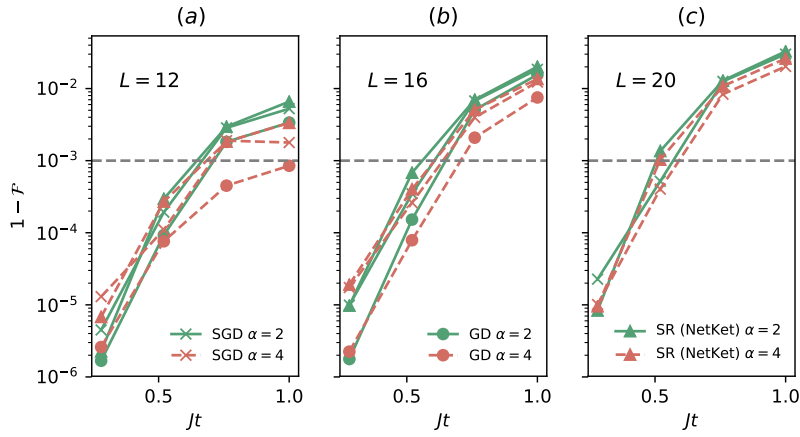


Figure C.7.: Comparison of results obtained from different optimizers and implementations for RBMQS of two different widths  $\alpha = 2, 4$  approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 12, 16, 20$  with quantum Ising model close to a critical point with interacting term ( $g = 1, h = 0, k = 0.25$ ).

In Fig. C.8, we compare the results for state approximation by one-hidden layer NAQS using SGD and GD over negative fidelity and using SGD over joint cost function.

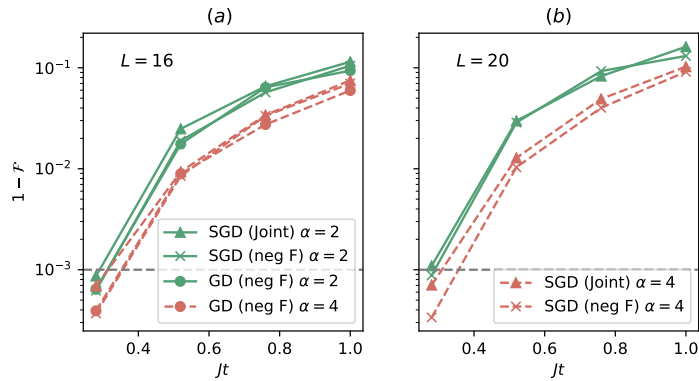


Figure C.8.: Comparison of results obtained by different optimizers and cost functions for one-hidden layer NAQS of two different widths  $\alpha = 2, 4$  approximating the exact time-evolved states following a quantum quench from an initial paramagnetic product state. The quenched Hamiltonian is given in Eq. (5.9) for a chain of length  $N = 12, 16, 20$  with quantum Ising model close to a critical point with interacting term ( $g = 1, h = 0, k = 0.25$ ).





# D. Appendix Circuit

## D.1. Types of ansätze

Quantum circuits with low depth can have different structures and will, by definition, represent low-complexity quantum states. In Figure D.1, we show two possible ansätze – a sequential circuit and a brickwall/checkerboard circuit. The brickwall circuit strictly limits the possible correlations because two qubits are only correlated if their reversed light cones coincide. If the light cones do not coincide, the correlations should be zero, i.e.  $\langle \hat{O}_i \hat{O}_j \rangle = \langle \hat{O}_i \rangle \langle \hat{O}_j \rangle$ . More precisely, each layer of the brickwall circuit is a locality-preserving unitary operator that only creates correlations locally. This contrasts with the sequential circuit, which allows correlations between qubits over arbitrary length scales. For this reason, we expect that different ansätze will capture slightly different states.

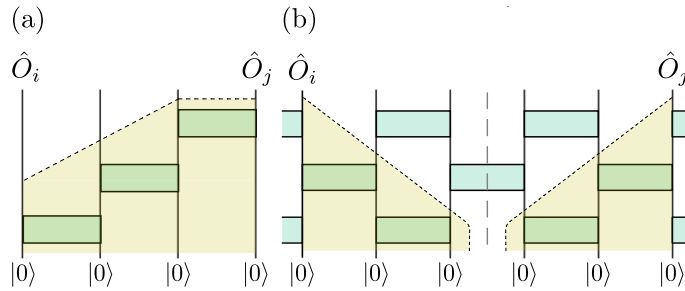


Figure D.1.: Consider the correlation for  $\langle \hat{O}_i \hat{O}_j \rangle - \langle \hat{O}_i \rangle \langle \hat{O}_j \rangle$ . (a) The sequential circuit can have a nonzero correlation throughout the entirety of the system. (b) The correlation in the brickwall/checkerboard circuit is 0 if the backward lightcones do not cross an input qubit.

Although sequential circuits could represent states having a correlation of arbitrary length scale, e.g. GHZ state, sequential circuits cannot support the long-range decay of generic correlators, but only a special subset.

## D.2. Classical simulation algorithm for quantum circuit

In this section, we describe two algorithms. The first algorithm maximizes the fidelity between two states defined by a set of unitaries, similar to the known Evenly-Vidal algorithm [158]. The second algorithm uses the first algorithm to perform time evolution restricted to the space defined by the ansatz under consideration.

To maximize the fidelity  $\mathcal{F} = |\langle \Psi_{\text{target}} | \Psi_{\text{qc}}^M \rangle|^2$ , we iteratively optimize the fidelity with respect to each gate  $U_{i,j}$ , while keeping the remaining gates fixed. Note that the double indices  $(i, j)$  refer to order and site respectively, whereas in the main text we group the indices into a single index.

We first rewrite the overlap between the target state  $|\Psi_{\text{target}}\rangle$  and the order-M circuit

## Appendix D. Appendix Circuit

$|\Psi_{\text{qc}}^M\rangle$  in the following form,

$$\langle \Psi_{\text{target}} | \Psi_{\text{qc}}^M \rangle \quad (\text{D.1})$$

$$= \langle \Psi_{\text{target}} | \prod_{i=1}^M \prod_{j=1}^{N-1} U_{i,j} | \Psi_{\text{product}} \rangle \quad (\text{D.2})$$

$$= \langle \Psi_{\text{target}} | \underbrace{U_{M,N-1} U_{M,N-2} \dots U_{i,j}}_{\langle \phi |} \dots U_{1,2} U_{1,1} | \Psi_{\text{product}} \rangle^{\psi} \quad (\text{D.3})$$

$$= \langle \phi | U_{i,j} | \psi \rangle \quad (\text{D.4})$$

$$= \text{Tr} [ | \psi \rangle \langle \phi | U_{i,j} ] \quad (\text{D.5})$$

$$= \text{Tr} [ E U_{i,j} ] \quad (\text{D.6})$$

$$(\text{D.7})$$

where  $U_{i,j}$  is the unitary to optimize and  $E$  is the environment matrix as shown in Fig. D.2(a).

The fidelity  $\mathcal{F} = |\langle \Psi_{\text{target}} | \Psi_{\text{qc}}^M \rangle|^2 = \text{Re} [\langle \phi | U_{i,j} | \psi \rangle]^2$  is equal to the square of the real part of the overlap. This is because any global phase offset can always be compensated by absorbing a single site rotation into the 2-site unitary. The solution to the unitary maximizing  $\text{Re} [\langle \phi | U_{i,j} | \psi \rangle]$  is known; for  $E = XSY^\dagger$ , the optimal  $U_{i,j}$  is given by  $YX^\dagger$  as in Fig. D.2(b).

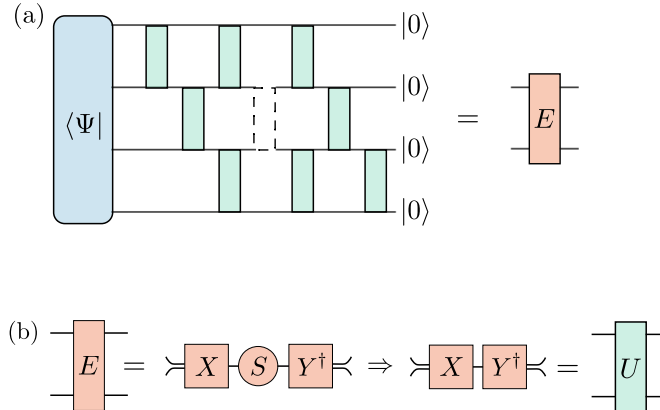


Figure D.2.: (a) The environment tensor is constructed by excluding the pertinent unitary from the overall contraction and viewing the resulting tensor network as a four-index tensor. (b) To update  $U_{i,j}$ , we perform a polar decomposition of the environment tensor.

To obtain the optimal circuit, we iterate through all of the gates and update each gate with the exact solution of the local optimization problem. Given a maximal iteration number  $N_{\text{iter}}$ , absolute convergence error  $\varepsilon_a$ , and relative convergence error  $\varepsilon_r$ , the algorithm is described in Alg. 4.

We used standard tensor network techniques to construct the environment tensor and truncated singular values less than  $10^{-14}$ . The algorithm was made significantly less expensive by caching and updating the environments to avoid recomputing the entire environment from scratch during each new iteration. For our computations,  $N_{\text{iter}} = 10^5$ ,  $\varepsilon_a = 10^{-12}$ ,  $\varepsilon_r = 10^{-4}$ .

We now introduce our second algorithm, which performs time evolution directly on the manifold defined by our ansatz. To time evolve a state  $|\Psi(t)\rangle$ , we maximize the fidelity

---

**Algorithm 4:** Maximizing overlap
 

---

**Input** :  $|\Psi_{\text{target}}\rangle, |\Psi_{\text{qc}}^M\rangle, N_{\text{iter}}, \varepsilon_a, \varepsilon_r$   
**Output:** A set of  $\{U_{i,j}\}$  maximizing  $\langle \Psi_{\text{target}} | \Psi_{\text{qc}}^M(\{U_{i,j}\}) \rangle, \varepsilon$   
 $\text{idx} = 0, \varepsilon_0 = \text{inf};$   
**while**  $\text{idx} < N_{\text{iter}}$  and  $\varepsilon_{\text{idx}} > \varepsilon_a$  and  $\Delta\varepsilon > \varepsilon_r$  **do**  
      $\text{idx} = \text{idx} + 1;$   
     **for**  $(i = 1; i < M; i = i + 1)$  {  
         **for**  $(j = 1; j < N - 1; j = j + 1)$  {  
             Construct environment matrix  $E;$   
              $E = X\Sigma Y^\dagger;$   
             Update  $U_{i,j} \leftarrow YX^\dagger;$   
         }  
     }  
      $\varepsilon_{\text{idx}} = 1 - \langle \Psi_{\text{target}} | \Psi_{\text{qc}}^M \rangle^2;$   
      $\Delta\varepsilon = |\varepsilon_{\text{idx}} - \varepsilon_{\text{idx}-1}| / |\varepsilon_{\text{idx}-1}|$   
**end**

---

$\mathcal{F} = |\langle \Psi(t + \Delta t) | \hat{V}(\Delta t) | \Psi(t) \rangle|^2$ , where our unitaries parameterize  $|\Psi(t + \Delta t)\rangle$  and  $\hat{V}(\Delta t)$  is a single Trotterized time step. In this way, we can iteratively evolve forward in time from an initial state. The overall algorithm for time evolution is given as in Alg. 5

---

**Algorithm 5:** Algorithm for Time Evolution
 

---

**Input** :  $H, |\Psi_{\text{qc}}^M(0)\rangle, t_{\text{end}}, \Delta t$   
**Output:** The set of gates  $\{U_{i,j}(t_{\text{end}})\}$  for the state  $|\Psi_{\text{qc}}^M(t_{\text{end}})\rangle$ , Overall error  $1 - \mathcal{E}$   
 $\mathcal{E} = 1;$   
**for**  $(t = 0; t < t_{\text{end}}; t = t + \Delta t)$  {  
     (1) Prepare the state  $|\Psi_{\text{qc}}^M(t)\rangle$  from the set of gates  $\{U_{i,j}(t)\}$  ;  
     (2) Apply time evolution gates and obtain  $|\Psi_{\text{qc}}^M(t + \Delta t)\rangle;$   
     (3) Find the new set of gates  $\{U_{i,j}(t + \Delta t)\}$  best representing the state  $|\Psi_{\text{qc}}^M(t + \Delta t)\rangle$  by Alg.1 ;  
     (4)  $\mathcal{E} = \mathcal{E} \times \mathcal{F}$  ;  
**}**

---

There are two primary sources of error in our algorithm: the Trotterization error and the projection error. The Trotterization error arises from approximating the true time evolution operator by a series of 2-site gates. This can be made arbitrarily small by decreasing  $\Delta t$  or by taking higher-order Trotter decompositions. The projection error arises from projecting the time evolved state back onto the manifold of circuits of order  $M$ . This error is affected by the chosen ansatz and limits the time to which one can simulate within a given error threshold.

We can estimate the total error by monitoring the fidelity at the end of each optimization  $\prod_i \mathcal{F}_i$ . This total error estimate is accurate as long as the Trotterization error remains small and if  $\mathcal{F}_i$  is close to 1 at each step. As an example, in Fig. D.3 we show the error estimates for the simulation performed in Fig. 7.4. We see that the time when the error crosses the threshold matches the time when  $\langle \sigma_z \rangle$  starts to deviate.

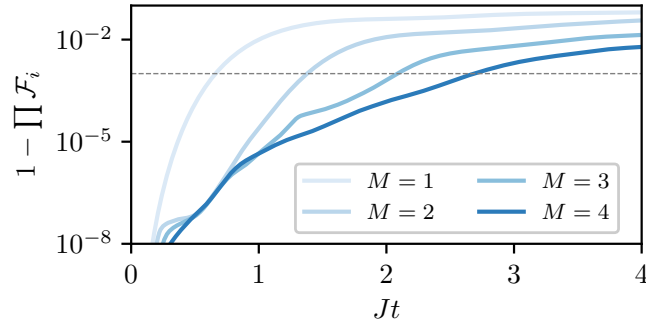


Figure D.3.: Approximation error of time evolution algorithm restricted to the quantum circuit ansatz for different  $M$ .

### D.3. Non-unitary gates

In this section, we describe a procedure to embed an arbitrary non-unitary  $N$ -qubit operator  $A$  in an  $(N + 1)$ -qubit unitary gate. We consider the first of these  $(N + 1)$ -qubits to be an ancilla qubit that we initialize in the  $|0\rangle$  state and project into the state  $|0\rangle$  by post-selection. Our claim is that there exists a unitary of the form

$$U_A = \begin{pmatrix} sA & B \\ C & D \end{pmatrix}, \quad (\text{D.8})$$

where  $s^{-2}$  is the maximum eigenvalue of  $A^\dagger A$  (or equivalently  $AA^\dagger$ ). We note that the matrices  $A^\dagger A$  and  $AA^\dagger$  have real and non-negative spectra. This follows from a singular value decomposition, i.e.  $A = U\Sigma V^\dagger$  with  $U, V$  unitary and  $\Sigma$  non-negative real and diagonal, and so  $A^\dagger A = V(\Sigma^2)V^\dagger$  and  $AA^\dagger = U(\Sigma^2)U^\dagger$ . Our goal is to show that for any  $A$  ( $\neq 0$ , although this case can also be included) we can find the  $2^N \times 2^N$  matrices,  $B, C$  and  $D$  such that  $U_A$  is unitary.

Our approach is the following. We first note that  $U_A$  being unitary is equivalent to the statement that the columns of  $U_A$  form an orthonormal basis of  $\mathbb{C}^{2^{N+1}}$ . We will then use this to find a block  $C$  and a scaling factor  $s$  consistent with this, i.e. such that the first  $2^N$  columns of  $U_A$  are orthonormal. Given  $A, C$ , and  $s$  we can then use a QR-decomposition to easily find  $B$  and  $D$ , as explained below.

Let us denote the columns of  $A$  and  $C$  by  $a_j$  and  $c_j$  respectively, e.g.,  $[C]_{ij} = [c_j]_i$ . For  $U_A$  to be unitary  $C$  and  $s$  must satisfy

$$C^\dagger C = \mathbf{1} - s^2 A^\dagger A, \quad CC^\dagger = \mathbf{1} - s^2 AA^\dagger. \quad (\text{D.9})$$

In terms of the column vectors, these can be written as

$$c_i \cdot c_j + s^2 a_i \cdot a_j = \delta_{ij}. \quad (\text{D.10})$$

For  $i = j$  this is a statement that the first  $2^N$  columns of  $U_A$  are normalized, and for  $i \neq j$  it is the statement that these columns are mutually orthogonal.

Next, we note that if  $C$  satisfies Eq. (D.9), then we have the singular value decomposition  $C = U\tilde{\Sigma}V^\dagger$ , where  $U$  and  $V$  are the same unitaries as in the SVD of  $A = U\Sigma V^\dagger$ . This implies that

$$C^\dagger C = V\tilde{\Sigma}^2V^\dagger, \quad CC^\dagger = U\tilde{\Sigma}^2U^\dagger. \quad (\text{D.11})$$

Since both  $\Sigma^2$  and  $\tilde{\Sigma}^2$  must be non-negative, we only have a solution to Eq. (D.9) if  $s^{-2}$  is greater than the largest eigenvalue of  $A^\dagger A$  (all of which are non-negative), and so we set

#### D.4. Cost of the imaginary-time evolution

$s^{-2}$  equal to the largest eigenvalue. We therefore have that  $\tilde{\Sigma}^2 = \mathbb{1} - s^2\Sigma^2$ , with our choice of  $s$  ensuring that  $\tilde{\Sigma}$  is real and non-negative.

Finally, given  $A$  and  $C$ , we can find the blocks  $B$  and  $D$  using QR-decomposition. Namely, let us construct the matrix

$$\tilde{U}_A = \begin{pmatrix} sA & \tilde{B} \\ C & \tilde{D} \end{pmatrix}, \quad (\text{D.12})$$

where  $\tilde{B}$  and  $\tilde{D}$  are random matrices, then by QR-decomposition

$$\tilde{U}_A = U_A R, \quad (\text{D.13})$$

where  $U_A$  is the unitary in Eq. (D.9) and  $R$  is an upper triangular matrix. Since the first  $2^N$  columns of  $\tilde{U}_A$  are orthonormal, they will be untouched by the QR-decomposition algorithm.

#### D.4. Cost of the imaginary-time evolution

In our imaginary-time evolution algorithm, we make use of an ancilla qubit to implement the non-unitary gates. This approach is therefore probabilistic in the sense that there is a probability that the ancilla qubit is measured in the state  $|1\rangle$  and the corresponding measurement of the fidelity must be disregarded. It is therefore important to analyze the probability of the failure of the algorithm and how the resources required scale with system size and simulation time.

We are interested in using the imaginary-time evolution algorithm to simulate the evolution with respect to a Hamiltonian  $H = \sum_i h_i$ , where  $h_i$  have a finite non-trivial support. Note that we typically are interested in the case when  $h_i$  are also local since most quantum computer architectures have a local connectivity structure. If we have long-range connectivity in the quantum computer, or are able to accommodate the linear cost of using swap gates, then the locality can be relaxed so long as the terms act on a finite number of qubits. Note that the circuit depth generically scales exponentially with the non-trivial support of  $h_i$ .

In the trotterized evolution, the non-unitary gates we wish to implement are of the form  $A = e^{-\Delta\tau h_i}$ . Taking into account the scaling factor  $s$ , we wish to embed the matrix  $sA = e^{-\Delta\tau(h_i - E_{i,\min})}$ , where  $E_{i,\min}$  is the smallest eigenvalue of  $h_i$ . The probability  $P(0)$  that we measure the ancilla in the state  $|0\rangle$  is dependent on the state of the physical qubits, i.e.

$$P(0) = \text{Tr} \left[ \rho s^2 A^\dagger A \right] = \text{Tr} \left[ \rho e^{-2\Delta\tau(h_i - E_{i,\min})} \right] \quad (\text{D.14})$$

$$P(1) = \text{Tr} \left[ \rho (\mathbb{1} - s^2 A^\dagger A) \right] = \text{Tr} \left[ \rho (\mathbb{1} - e^{-2\Delta\tau(h_i - E_{i,\min})}) \right] \quad (\text{D.15})$$

where  $\rho = |\psi\rangle\langle\psi|$  is the density matrix of the physical state. However, in the worst case scenario, the probability of success is  $e^{-2\Delta\tau(E_{i,\max} - E_{i,\min})}$ . We can choose to scale the Hamiltonian such that the maximum bandwidth of the terms  $h_i$  is 1, that is we scale the total Hamiltonian such that

$$\max_i (E_{i,\max} - E_{i,\min}) = 1. \quad (\text{D.16})$$

This scaling of the Hamiltonian can be incorporated into  $\Delta\tau$ . Indeed,  $\Delta\tau$  should be specified relative to the maximum energy density of the Hamiltonian. Given this natural scaling of the Hamiltonian, the worst-case probability of success for any of the gates in the

Trotterization is  $e^{-2\Delta\tau}$ , which depends only of  $\Delta\tau$  and not on the system size and can be made arbitrarily close to one.

To measure the fidelity with a certain statistical accuracy, we thus need to measure a number of shots proportional to  $e^{2\Delta\tau}$ . Furthermore, at any step, we can determine the statistical accuracy of our observables by counting the number of times the ancilla qubit is in the  $|0\rangle$  state and perform more measurements if necessary without the need to restart the algorithm. We are able to do this because we optimize and store the state ansatz after applying each of the gates in the Trotter evolution. Therefore, despite the probabilistic nature of the algorithm, the total cost of the simulation scales linearly with system size and with the total simulation time.

## D.5. Detailed data for parameter counting

In this section, we include the data corresponding to parameter counts required to achieve a fixed fidelity as a function of time for matrix-product states (Fig. D.4(a)) and quantum circuits (Fig. D.4(b)).

We observe that a complex isometric matrix  $W \in \mathbb{C}^{n \times p}$ ,  $n \geq p$ , satisfying the isometric condition  $W^\dagger W = \mathbb{1}$  has  $2np - p^2$  real independent parameters since the isometric condition imposes  $p^2$  independent real-valued constraints. To count the number of parameters for an MPS, we first put the MPS into the canonical form and then sum up the number of parameters in each isometric tensor.

When counting the number of parameters of an order- $M$  ansatz, because the circuit starts from a fixed initial state ( $|000\dots 00\rangle$ ), there are redundant degrees of freedom. If we consider a gate acting on a fixed qubit in matrix form, the columns that do not correspond to the fixed qubit are irrelevant. The very first gate in the first layer, which acts on two fixed qubits, will have  $2d^2 - 1 = 7$  parameters. All the other gates in the first layer act only on one fixed qubit, and thus have  $2d^3 - d^2 = 12$  parameters. The gates in all other layers have  $2^4 = 16$  parameters.

Proceeding with this counting, the number of parameters of a bond dimension  $\chi = 2$  MPS matches our order  $M = 1$  ansatz, while a two-layer brickwall quantum circuit has fewer parameters. This confirms the result in Appendix 6.2.

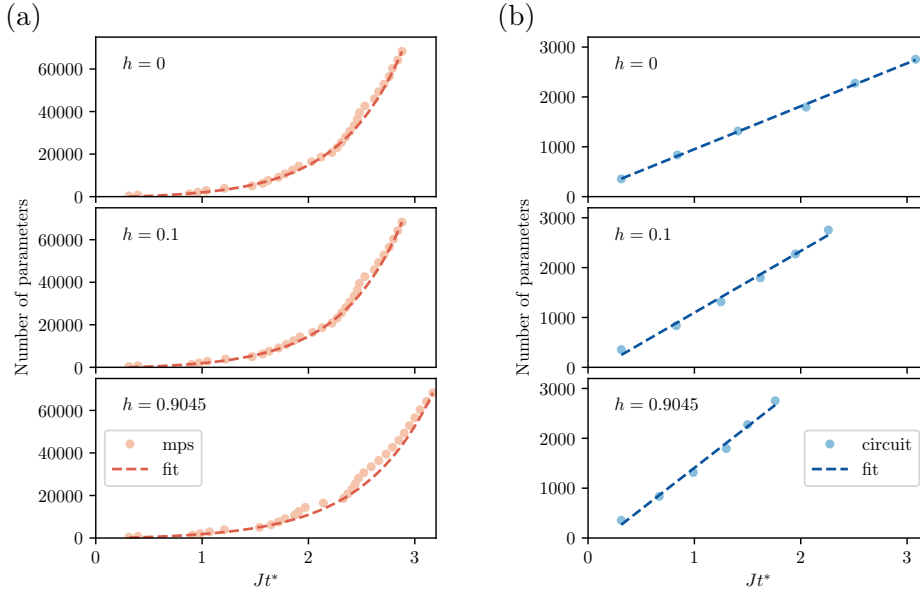


Figure D.4.: (a) We fit the MPS data with  $f(Jt^*) = ae^{bJt^*} + c$  and obtain the  $(a, b, c)$  parameters for  $h = 0$ ,  $(540, 1.69, -910)$ ,  $h = 0.1$ ,  $(506, 1.71, -836)$ , and  $h = 0.9045$ ,  $(559, 1.52, -711)$  respectively. Note that we only fit the data points with bond dimension being power of two, i.e.  $\chi = 2^n$ ,  $n \in \mathbb{Z}^+$ . (b) We fit the quantum circuit data with  $f(Jt^*) = a \times (Jt^*) + b$  and obtain  $(a, b)$  parameters for each case,  $h = 0$ ,  $(861, 91)$ ,  $h = 0.1$ ,  $(1236, -138)$ , and  $h = 0.9045$ ,  $(1659, -251)$ , respectively.

## D.6. Randomized circuits for QPU measurement

The quantum circuit considered in this paper is described by a series of two-site gates  $\{U_i\}$ . When the quantum circuit is implemented on a QPU, the two-site gates are decomposed into a series of finitely-many gates selected from some universal gate set. A small perturbation of a two-site gate may lead to a large perturbation in the decomposition. These differences translate into large fluctuations in the measured observables due to imperfections in the QPU.

To compensate for this problem, we average over the gauge freedom in a quantum circuit. Given the two-site gates  $\{U_i\}$  describing the quantum states, there are gauge degrees of freedom to insert identities described by random unitaries and their complex conjugates. For example, if  $U_i U_{i+1}$  act consecutively on the same qubit, we can insert the random single-site unitary  $V$  and its complex conjugate as

$$U_{i+1}U_i = U_{i+1}V^\dagger V U_i = W_{i+1}W_i \quad (\text{D.17})$$

and obtain the two-site gates  $W_{i+1}W_i$  describing the same operation. To average over the gauge degrees of freedom, we average measurement outcomes corresponding to circuits differing by the insertion of random unitaries and their conjugates. This procedure mitigates the previously mentioned error to a certain extent.





# E. Appendix Inf Circuit

## E.1. Transfer matrix

Transfer matrices are utilized for computing the physical observables and other operations of an infinite system. In the context of tensor networks, the transfer matrix  $\hat{T}$  of the quantum state  $|\psi\rangle$  is defined as the repeating block in the computation of the inner product,  $\text{Tr}[\hat{T}^N] = \langle\psi|\psi\rangle$ , where  $N$  is the system size. Under the mild assumption that the state considered is injective, the expression is well-defined in the thermodynamic limit  $N \rightarrow \infty$  regardless of the boundary conditions. In the following, we extend the same formalism to USC.

### E.1.1. Transfer matrix of USC

Utilizing the left and right representations of the l-USC ansatz, we have the freedom to write the inner product as  $\langle\psi_{\text{R}}(\boldsymbol{\theta})|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle = \lim_{N \rightarrow \infty} \text{Tr}[\hat{T}^N]$ , of which the repeated block is defined as the transfer matrix shown as the shaded area in Fig. 8.1 (a). Such mixed representation allows us to express the transfer matrix as a linear operator acting on pure states instead of density matrices.

In general, the states  $|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle$  and  $|\psi_{\text{R}}(\boldsymbol{\theta})\rangle$  can be of different  $N_q$  and  $M_U$ , and they represent similar but not exactly identical states. We can define the mixed transfer matrix  $\hat{T}$  between two different quantum states  $|\psi_{\text{L}}\rangle$  and  $|\phi_{\text{R}}\rangle$  as the repeating block in the computation of their inner product,  $\xi = \langle\phi_{\text{R}}|\psi_{\text{L}}\rangle = \lim_{N \rightarrow \infty} \text{Tr}[\hat{T}^N]$ . The left and right environments  $|l\rangle$  and  $|r\rangle$  and the leading eigenvalue  $\lambda$  are defined similarly to the case of the (not mixed) transfer matrix. In the thermodynamic limit, the absolute value of the inner product  $|\xi|$  is given by

$$\lim_{N \rightarrow \infty} |\text{Tr}[\hat{T}^N]| = \lim_{N \rightarrow \infty} |\lambda^N| = \begin{cases} 1 & |\lambda| = 1 \\ 0 & |\lambda| < 1 \end{cases}, \quad (\text{E.1})$$

which is either an identity or zero, depending on whether the states are identical. Therefore, a better quantity to consider is instead the overlap density, which is equal to the absolute value of the leading eigenvalue of the transfer matrix  $|\lambda|$ , satisfying the relation

$$\log|\lambda| = \lim_{N \rightarrow \infty} \frac{1}{N} \log|\xi|. \quad (\text{E.2})$$

### E.1.2. Evaluating local observables

One can evaluate the expectation value of a local observable with respect to the state  $|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle$  following the equation

$$\langle\hat{\mathcal{O}}\rangle = \frac{\langle\psi|\hat{\mathcal{O}}|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\langle\psi_{\text{L}}(\boldsymbol{\theta}')|\hat{\mathcal{O}}|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle}{\langle\psi_{\text{L}}(\boldsymbol{\theta}')|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle}. \quad (\text{E.3})$$

The last expression can be evaluated with the right environment in the density matrix form, which satisfies the fixed-point equations. This is the approach taken by [213, 214].

## Appendix E. Appendix Inf Circuit

Here, assuming we have the (approximately) identical state in left and right representation, we take an alternative approach and approximate the expression by

$$\langle \hat{\mathcal{O}} \rangle \approx \frac{\langle \psi_{\text{R}}(\boldsymbol{\theta}) | \hat{\mathcal{O}} | \psi_{\text{L}}(\boldsymbol{\theta}') \rangle}{\langle \psi_{\text{R}}(\boldsymbol{\theta}) | \psi_{\text{L}}(\boldsymbol{\theta}') \rangle} + O\left(\left|\frac{\Delta}{\lambda_1 - \lambda_2}\right|\right). \quad (\text{E.4})$$

The expectation value is now evaluated utilizing the mixed representation of the transfer matrix. Here,  $\Delta$  is the norm difference between the tensors per site describing the states when collapsing the circuit to uniform MPS form, and  $\lambda_1, \lambda_2$  are the leading and second leading eigenvalues of the transfer matrix of the state  $|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle$ . Therefore, the expression is exact when  $|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle$  and  $|\psi_{\text{R}}(\boldsymbol{\theta})\rangle$  are exactly the same state. The expression is a valid approximation, when the norm difference is smaller than the size of the gap in the transfer matrix of the physical state  $|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle$ .

In the simpler case where  $|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle$  and  $|\psi_{\text{R}}(\boldsymbol{\theta})\rangle$  represent exactly the same physical state, we can always absorb the phase factor into one of the state unitaries such that  $\lambda = 1$ . The numerator and the denominator are then reduced to

$$\langle \psi_{\text{R}}(\boldsymbol{\theta}) | \hat{\mathcal{O}} | \psi_{\text{L}}(\boldsymbol{\theta}') \rangle = \langle l, 0 | \hat{U}_{\text{R}}^\dagger \hat{\mathcal{O}} \hat{U}_{\text{L}} | 0, r \rangle \quad (\text{E.5})$$

$$\langle \psi_{\text{R}}(\boldsymbol{\theta}) | \psi_{\text{L}}(\boldsymbol{\theta}') \rangle = \langle l | r \rangle \quad (\text{E.6})$$

following the definition of the environments. Therefore, we can transform the infinite circuit  $\langle \psi_{\text{R}}(\boldsymbol{\theta}) | \hat{\mathcal{O}} | \psi_{\text{L}}(\boldsymbol{\theta}') \rangle$  shown in Fig. 8.1 (a) into the finite circuit  $\langle l, 0 | \hat{U}_{\text{R}}^\dagger \hat{\mathcal{O}} \hat{U}_{\text{L}} | 0, r \rangle$  given in Fig. 8.1 (d), which can be implemented on a quantum computer.

If  $|\psi_{\text{L}}(\boldsymbol{\theta}')\rangle$  and  $|\psi_{\text{R}}(\boldsymbol{\theta})\rangle$  are not identical, the numerator is suppressed by the additional factor  $\lambda^{N-1}$  which cancels out mostly with the additional factor in the denominator  $\lambda^N$ , leading to the expression

$$\langle \hat{\mathcal{O}} \rangle \approx \lim_{N \rightarrow \infty} \frac{\lambda^{N-1} \langle l, 0 | \hat{U}_{\text{R}}^\dagger \hat{\mathcal{O}} \hat{U}_{\text{L}} | r, 0 \rangle}{\lambda^N \langle l | r \rangle} = \frac{\langle l, 0 | \hat{U}_{\text{R}}^\dagger \hat{\mathcal{O}} \hat{U}_{\text{L}} | r, 0 \rangle}{\lambda \langle l | r \rangle}. \quad (\text{E.7})$$

The expression suggests that for generic cases, one shall also take into account the contribution from  $\lambda \neq 1$ .

### E.1.3. Derivative of the transfer matrix

In this appendix, we derive the expression for the derivative of the leading eigenvalue of the transfer matrix with respect to the state unitary. Consider the transfer matrix  $\hat{T}(\boldsymbol{\theta})$  depending on variational parameters  $\boldsymbol{\theta}$  and its left and right environments  $|l(\boldsymbol{\theta})\rangle, |r(\boldsymbol{\theta})\rangle$ , such that  $\hat{T}(\boldsymbol{\theta})|r(\boldsymbol{\theta})\rangle = \lambda(\boldsymbol{\theta})|r(\boldsymbol{\theta})\rangle, \hat{T}^\dagger(\boldsymbol{\theta})|l(\boldsymbol{\theta})\rangle = \lambda^*(\boldsymbol{\theta})|l(\boldsymbol{\theta})\rangle$ . Therefore, the environments and the leading eigenvalue depend on  $\boldsymbol{\theta}$ . We express the leading eigenvalue of the transfer matrix as

$$\lambda = \frac{\langle l | \hat{T} | r \rangle}{\langle l | r \rangle}, \quad (\text{E.8})$$

where the  $\boldsymbol{\theta}$ -dependence is omitted for the sake of notation. Taking derivative with respect to  $\boldsymbol{\theta}$ , we obtain

$$\begin{aligned} \nabla \lambda &= \frac{\langle \nabla l | \hat{T} | r \rangle + \langle l | \nabla \hat{T} | r \rangle + \langle l | \hat{T} | \nabla r \rangle}{\langle l | r \rangle} \\ &\quad - \lambda \frac{\langle \nabla l | r \rangle + \langle l | \nabla r \rangle}{\langle l | r \rangle}. \end{aligned} \quad (\text{E.9})$$

Using  $\hat{T}|r\rangle = \lambda|r\rangle$  and  $\langle l|\hat{T} = \lambda\langle l|$ , we note that the first and third terms in the first fraction cancel out with the second fraction, leaving

$$\nabla \lambda = \frac{\langle l | \nabla \hat{T} | r \rangle}{\langle l | r \rangle}. \quad (\text{E.10})$$

### E.1.4. Post-selection probability

Consider an arbitrary  $N$ -qubit vector  $|v\rangle$  parameterized by a unitary  $\hat{E}_v$  such that  $|v\rangle = \hat{E}_v|0\dots 0\rangle$ , and the (mixed) transfer matrix given by  $\hat{T} = \langle 0_{\text{last}}|U_R^\dagger U_L|0_{\text{first}}\rangle$ , acting on  $N$  qubits as shown in Fig. 8.1 (c). The action of the transfer matrix on the vector reads

$$\begin{aligned}\hat{T}|v\rangle &= \hat{T}\hat{E}_v|0\dots 0_{\text{last}}\rangle \\ &= \langle 0_{\text{last}}|U_R^\dagger U_L(\mathbb{1} \otimes E_v)|0_{\text{first}}, 0\dots 0_{\text{last}}\rangle.\end{aligned}\quad (\text{E.11})$$

Therefore, the probability of measuring  $|0_{\text{last}}\rangle$ , i. e.,  $|0\rangle$  on the last qubit, is given by

$$P(0) = \|\hat{T}|v\rangle\|^2. \quad (\text{E.12})$$

Note that by definition, the leading eigenvalue of the transfer matrix is unity and the absolute value of the leading eigenvalue of any mixed transfer matrix is equal or less than unity.

We see that the probability is unity if the input vector is the environment  $|r\rangle$ , i. e., the leading eigenvector of the transfer matrix  $\hat{T}$ , since in such case  $P(0) = \|\hat{T}|r\rangle\|^2 = |\lambda|^2 = 1$ . For an arbitrary input state  $|v\rangle$ , the probability can be expressed as

$$P(0) = \|\hat{T}|v\rangle\|^2 = \sum_i |c_i|^2 |\lambda_i|^2, \quad (\text{E.13})$$

where  $c_i$  is the coefficient of the eigenbasis of  $\hat{T}$ . As a result, the probability  $P(0)$  is lower bounded by the square of the coefficient  $|c_1|^2$  corresponding to the leading eigenvector. We note that  $|c_1|^2$  is close to unity in the case of Algorithm 2 if we initialize the vector using the environments from the previous time step.

Furthermore, the above property motivates an alternative algorithm for obtaining the environments by maximizing probability  $P(0)$  using gradient ascent methods. This algorithm is potentially more efficient as it only requires the measurement of the last qubit, with gradients measured in the absence of ancilla qubits using only the well-known parameter shift rule [332].

## E.2. Details of classical optimization of l-USC

In this appendix, we provide details on the optimization of l-USC that we perform in the course of classical simulation of the time evolution algorithm.

### E.2.1. Unitary parametrization and reunitarization

Optimization of the l-USC ansatz and environments, is performed with the gradient descent method. To incorporate the gradient descent method with the quantum circuits running on a quantum computer, one can employ the decomposition of a general two-qubit gate into 15 gates of the form

$$\hat{u}(\alpha) = \exp\left(i\alpha\sigma^a \otimes \sigma^b\right), \quad a \in \{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}, \quad (\text{E.14})$$

introduced in Ref. [332]. These gates have an important property,  $\partial_\alpha \hat{u}(\alpha) = \hat{u}(\alpha + \pi/2)$ , i. e. the derivative of these unitary gates is also unitary. This results in all derivatives of  $\hat{U}_{R/L}(\boldsymbol{\theta})$  or environments being unitary.

In classical optimization, we store  $N \times N$  unitaries directly using  $2N^2$  parameters, which is redundant but significantly speeds up the optimization. To ensure the correctness of the algorithm, after each finite gradient descent step, we *reunitarize* a gate  $\hat{U}$  by (i) performing the singular value decomposition  $\hat{U} = \hat{V}^\dagger \hat{D} \hat{W}$  with  $\hat{D}$  being a diagonal matrix with singular values and (2) replacing  $\hat{D}$  with a unity matrix:  $\hat{U} \rightarrow \hat{V}^\dagger \hat{W}$ .

### E.2.2. Derivative with respect to a gate

The main building block of the optimization is obtaining derivatives of the expectation values. The outlined recipe is applicable not only to two-qubit gates, used in the case of layered state or environment, but also to larger unitaries, used in the optimization of the dense (exact) environment or d-USC. Consider an expectation value (generally, a scalar complex-valued function) that depends on a set of unitary gates  $\lambda(\hat{U}_1, \hat{U}_2, \dots, \hat{U}_N)$ . To compute the derivative with respect to  $\hat{U}_k$ , any such scalar expectation can be written as  $\lambda = \text{Tr} \hat{U}_k \hat{W}_k^\dagger$ , with some  $\hat{W}_k$  depending on the remaining unitaries. Therefore, the derivative reads:

$$\frac{\partial \lambda}{\partial \hat{U}_k} = \hat{W}_k. \quad (\text{E.15})$$

### E.2.3. Environment optimization

If the proposed time-evolution algorithm is performed using a dense representation of environments, the environment is obtained by finding the eigenvector of the transfer matrix with the largest magnitude of eigenvalue using the Arnoldi iteration method.

If the environments are in the layered representation, we first obtain the exact dense environment  $|E_{\text{exact}}\rangle$  using the Arnoldi iteration, and then obtain the two-qubit gates of the approximating environment by maximizing the overlap  $|\langle E_{\text{layered}} | E_{\text{exact}} \rangle|^2$ . In such case, instead of performing a gradient descent with the gradient computed using Eq. (E.15), we employ the *polar decomposition rule* [158]. The polar decomposition rule utilizes the fact that the expectation  $\lambda = \text{Tr} \hat{U}_k \hat{W}_k^\dagger$  is maximized over all possible unitaries by taking  $\hat{U}_k$  as the reunitarization of  $\hat{W}_k$ . Thus, to optimize a layered environment, we sweep sequentially over all two-qubit gates in  $|E_{\text{layered}}\rangle$  and change them using the polar decomposition rule. We stop when the overlap between the environments obtained on two consecutive sweeps exceeds  $1 - 10^{-10}$ .

### E.2.4. Details of gradient descent method and stopping criteria

In the classical simulation of the proposed algorithm, we employ the redundant parametrization of the unitaries. First, having computed the gradient of  $|\lambda|^2$  with respect to a unitary  $\hat{U}_k$ ,  $\hat{D}_k = \partial |\lambda|^2 / \partial \hat{U}_k = 2 \text{Re} \left[ \lambda^* \partial \lambda / \partial \hat{U}_k \right]$ , we project this gradient onto the tangent space of the manifold of  $N \times N$  unitary matrices:

$$\hat{D}_k \rightarrow \hat{D}_k - \frac{1}{2} \hat{U}_k \hat{U}_k^\dagger \hat{D}_k + \frac{1}{2} \hat{U}_k \hat{D}_k^\dagger \hat{U}_k. \quad (\text{E.16})$$

The resulting unitary is reunitarized.

We employ the ADAM optimizer [259] with the learning rate  $\eta = 3 \times 10^{-3}$ . These two modifications improve the convergence of the algorithm. The optimization finished when the improvement of the leading eigenvalue of the transfer matrix between the two consecutive iterations was less than  $10^{-10}$ .

Lastly, to speed up the Arnoldi iteration method, we employed the graphical processing units (GPU) Nvidia V100.

## E.3. Sufficiency of the $M_U = 1$ state-unitary decomposition

In this appendix, we demonstrate that the accuracy of the time-evolved  $M_U = 1$  l-USC ansatz corresponds to the accuracy of the full dense d-USC time-evolved wave function. To obtain the d-USC ansatz wave function at  $N_q$ , we optimize the uniform MPS at  $\chi =$

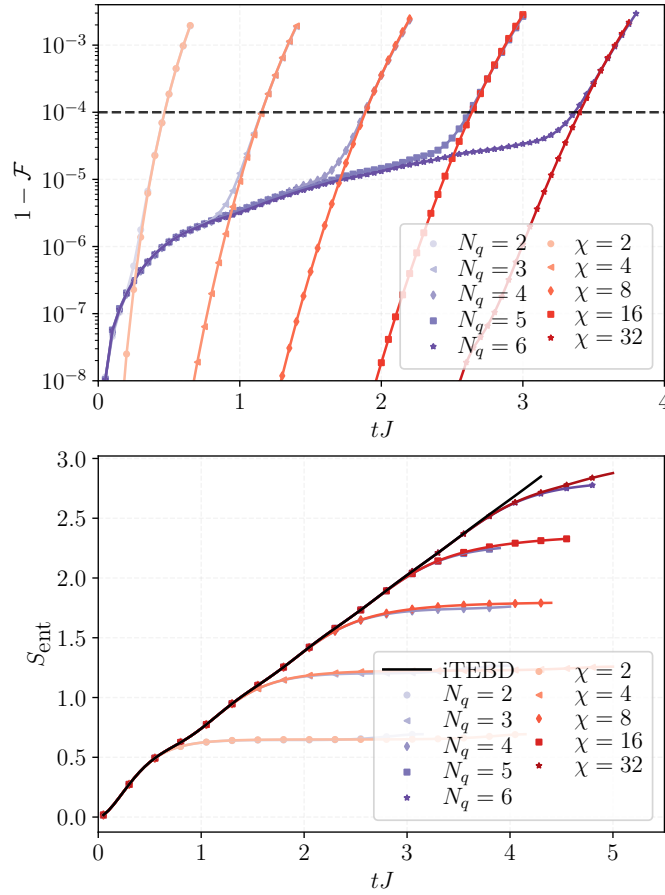


Figure E.1.: (a) Infidelity density between the quasi-exact solution and the l-USC ansatz at  $M_U = 1$  or uniform MPS at  $\chi = 2^{N_q-1}$  as a function of  $tJ$  at  $g/J = 1.0$ . (b) Entanglement entropy as a function of  $tJ$ . The solid line represents the quasi-exact solution.

$2^{N_q-1}$  using the classical iTEBD algorithm. In Fig. E.1 we show the fidelity densities and entanglement entropies in both layered and full dense cases.

In Fig. E.1 (a), the infidelity of the uniform MPS differs in the small  $tJ$  region, where the infidelity is vanishing, due to different optimization protocols: in the case of the l-USC ansatz, the gradient descent is used, while uniform MPS is optimized using the singular-value decomposition and provides best possible approximation at each time step. Nevertheless, the locations of crossing of the  $10^{-4}$  infidelity threshold coincide within our resolution. Therefore, increasing  $M_U$  would not improve the l-USC ansatz performance, as it is bounded from above by the uniform MPS performance at  $\chi = 2^{N_q-1}$ .

Lastly, in Fig. E.1 (b), we show the entanglement entropy obtained within both approaches. Similarly, the curves are almost identical in the whole course of time evolution.

## E.4. The accumulated error $\mathcal{M}$ measure

During a realistic optimization on quantum hardware, one has no access to the quasi-exact time-evolved state. Thus, in order to estimate the current error, one can instead define an accumulated error measure

$$\mathcal{M}(t) = 1 - \prod_{i < t} |\lambda_i|^2, \quad (\text{E.17})$$

which is the deviation of the product of leading eigenvalues of the transfer matrices from unity. Such measure, in case of the absence of Trotter errors, should serve as an upper bound for the infidelity  $1 - \mathcal{F} \leq \mathcal{M}$ .

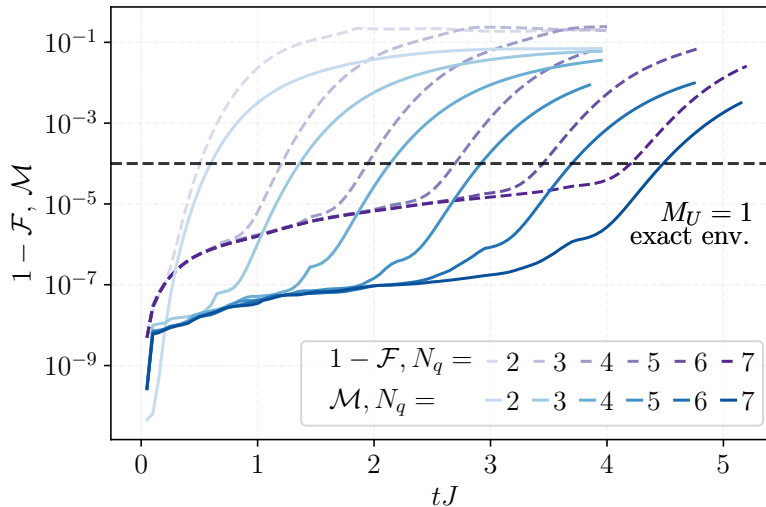


Figure E.2.: Comparison between the true infidelity density  $1 - \mathcal{F}$  computed using the exact solution (crosses) and using the cumulative estimation metric  $\mathcal{M}(t) = 1 - \prod_{i < t} |\lambda_i|^2$ . The simulations were performed at  $g/J = 1.0$ ,  $h/J = 0$  with an exact environment and state unitary with  $M_U = 1$ .

However, in this work, we obtain the exact wave function by running a classical iTEBD algorithm at high bond dimension, which breaks translation symmetry to two-site emergent unit cell, while in proposed algorithm we use a second-order translationally-invariant Trotterization. This discrepancy breaks the inequality, however, the two measures are still strongly correlated. To see this, in Fig. E.2 we show the dependence of the true time-evolution infidelity  $1 - \mathcal{F}(t)$  and the accumulated error measure. We observe that the two measures cross the  $10^{-4}$  threshold at close moments of time.

## E.5. Environment representation complexity

In this appendix, we present additional data illustrating the capacity of the layered representation of the environment. In Fig. E.3, we show the norm of the discrepancy between the exact environment and the layered environment at  $M_E$  layers as a function of  $tJ$ . The exact environments were obtained within the time-evolution of the l-USC ansatz with  $N_q = 6$  at  $g/J = 1.0$ . The maximum evolution time  $tJ$  is such that the overlap density between the ansatz and the exact state always exceeds  $1 - 10^{-4}$ .

We observe that, as the complexity of the state grows with time evolution, the approximations  $M_E < N_q - 1$  are clearly incapable of accurately representing  $|r_{\text{exact}}\rangle$  in the course of time evolution.

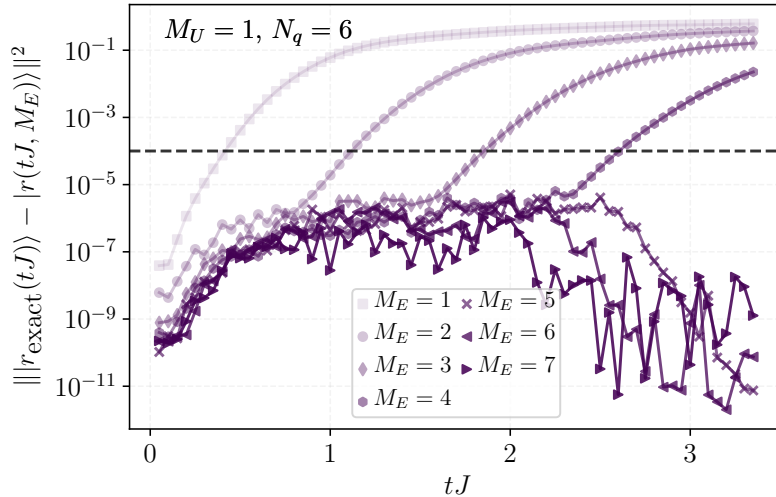


Figure E.3.: The norm difference between the exact environment  $|r_{\text{exact}}\rangle$  and the approximated  $|r(M_E)\rangle$  environment at  $M_E$  layers as a function of  $tJ$ . The  $10^{-4}$  boundary is chosen in the main text as the threshold of *accurate representation*. The data is obtained within fitting of the exact environment emerging during the  $N_q = 6$  evolution of the l-USC ansatz at  $g/J = 1.0$ .

## E.6. Randomized circuits for QPU measurement

The quantum circuit considered in this paper is described by a series of two-site gates  $\{\hat{u}_i\}$ . When this circuit is implemented on a QPU, the two-qubit gates within `qiskit` are decomposed into a series of gates selected from a universal gate set. A small perturbation of a two-site gate may lead to a large change of the decomposition. These differences lead to large fluctuations in the measured observables due to the QPU noise.

To mitigate these errors, we consider the following procedure. If the two consequent gates  $\hat{u}_i$  and  $\hat{u}_{i+1}$  act on the same qubit  $q$ , we sample a random  $SU(2)$  matrix  $\hat{v}$  acting only on the qubit  $q$  and modify  $\hat{u}_i \rightarrow \hat{u}_i \hat{v}$ ,  $\hat{u}_{i+1} \rightarrow \hat{v}^\dagger \hat{u}_{i+1}$ . We repeat the measurement scheme in several runs, each time sampling new single-qubit matrices  $\hat{v}$ .

## E.7. The equivalence of fixed points

In this appendix, we show that the left and right environments (fixed points), of the l-USC transfer matrix in mixed representation are identical up to complex conjugation. Using the formal equivalence between uniform MPS and d-USC shown in Section 6.2, we first show such property held in the case of uniform MPS. Consider a uniform MPS in the  $\Lambda$ - $\Gamma$  canonical form [31, 208]

$$|\Psi\rangle = \dots \text{---} \Lambda \text{---} \Gamma \text{---} \Lambda \text{---} \Gamma \text{---} \Lambda \text{---} \Gamma \text{---} \Lambda \text{---} \dots, \quad (\text{E.18})$$

where  $\Lambda$  is a positive-valued diagonal matrix, encoding the Schmidt values. The combinations of  $\Lambda$  and  $\Gamma$  give the left (normalized) isometric tensor  $A = \Lambda\Gamma$  and the right (normalized) isometric tensor  $B = \Gamma\Lambda$ . The overlap between the same physical state and itself is given as the following equation:

$$\langle\Psi|\Psi\rangle = \dots \begin{array}{c} \text{---} \Lambda^* \text{---} \Gamma^* \text{---} \Lambda^* \text{---} \Gamma^* \text{---} \Lambda^* \text{---} \Gamma^* \text{---} \Lambda^* \text{---} \\ \text{---} \Lambda \text{---} \Gamma \text{---} \Lambda \text{---} \Gamma \text{---} \Lambda \text{---} \Gamma \text{---} \Lambda \text{---} \end{array} \dots \quad (\text{E.19})$$

## Appendix E. Appendix Inf Circuit

The transfer matrix in the mixed representation is constructed with the left isometric tensor  $A = \Lambda\Gamma$  colored in light red and the right isometric tensor  $B^* = \Gamma^*\Lambda^*$  colored in light green. From the left and right isometric conditions, we see that the left and right environments are simply  $\Lambda^*$  and  $\Lambda$ , respectively. In this specific case, since the diagonal matrix  $\Lambda$  is real and positive, the left and right environments are identical.

In the above case, we consider the isometries with gauge fixing, leading to the  $\Lambda$ - $\Gamma$  canonical form. In general, the d-USC is equivalent to uniform MPS in isometric form without gauge fixing. That is, we are allowed to insert identity operators  $U^\dagger U = \mathbb{1}$  and  $V^\dagger V = \mathbb{1}$  to the left and the right of the  $\Lambda$  tensor, respectively, where the  $U$  and  $V$  are arbitrary unitaries. Similarly, the transfer matrix is constructed by the left and right isometric tensors describing the same physical state, but now, in arbitrary gauge. The overlap is then given by the equation

$$\langle \Psi | \Psi \rangle = \dots \left( \begin{array}{c} \text{---} \textcircled{U_2^\dagger} \textcircled{U_2^*} \textcircled{\Lambda^*} \textcircled{V_2^\dagger} \textcircled{V_2^*} \textcircled{\Gamma^*} \textcircled{U_2^\dagger} \textcircled{U_2^*} \textcircled{\Lambda^*} \textcircled{V_2^\dagger} \textcircled{V_2^*} \text{---} \\ | \\ \text{---} \textcircled{U_1^\dagger} \textcircled{U_1} \textcircled{\Lambda} \textcircled{V_1^\dagger} \textcircled{V_1} \textcircled{\Gamma} \textcircled{U_1^\dagger} \textcircled{U_1} \textcircled{\Lambda} \textcircled{V_1^\dagger} \textcircled{V_1} \text{---} \end{array} \right) \dots \quad (\text{E.20})$$

As a result, the left isometric tensor is now given by  $A = U_1\Lambda\Gamma U_1^\dagger$  colored in light red and the right isometric tensor is given by  $B^* = V_2^*\Gamma^*\Lambda^*V_2^T$  colored in light green. Similarly, by isometric conditions, the left environment is  $U_1^*\Lambda^*V_2^T$  while the right environment is  $U_1\Lambda V_2^\dagger$ . Therefore, the left environment and the right environment are identical up to complex conjugation.

Since any l-USC state can be viewed as a d-USC state, the statement also applies to the l-USC ansatz: when the l-USC transfer matrix is constructed in the mixed representation, the left and right environments are identical up to complex conjugation.





# Acronyms

0D	zero-dimensional 12, 18, 19
1D	one-dimensional 1, 2, 7, 8, 9, 12, 13, 14, 18, 19, 20, 21, 26, 27, 28, 31, 37, 38, 43, 45, 46, 61, 63, 67, 74, 77, 78, 80, 82, 83, 84, 97, 111, 114, 115
2D	two-dimensional 2, 3, 7, 8, 9, 14, 15, 17, 18, 19, 20, 26, 27, 28, 31, 37, 38, 42, 43, 44, 45, 46, 47, 61, 63, 84, 109, 111, 112, 113, 114, 115
3D	three-dimensional 15, 28, 84
ALS	alternative least squares 16, 21
ANN	artificial neural network 2, 63
bMPO	boundary matrix-product operator 15, 16, 37, 41, 45, 112, 124
CNN	convolutional neural network 64
CTMRG	corner transfer matrix renormalization group 15
CU	cluster update 16
d-USC	dense uniform sequential circuits 97, 98, 99, 105, 106
DMRG	density matrix renormalization group 2, 3, 7, 13, 15, 17, 31, 34, 37, 38, 83, 92, 111, 112
DSF	dynamical spin structure factor 40, 41, 42, 43
ED	exact diagonalization 1, 67, 93, 94
FTQC	fault-tolerant quantum computing 79
FU	full update 16, 27, 113
iDMRG	infinite density matrix renormalization group 53, 60
iMM	infinite Moses move 7, 8, 45, 60, 61, 111
iMPS	infinite matrix product state 45
iPEPS	infinite projected entangled pair states 17
isoTNO	isometric tensor network operator 45, 47, 48, 55, 59, 60
isoTNS	isometric tensor network state 2, 3, 7, 9, 10, 12, 14, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29, 31, 37, 38, 39, 40, 41, 43, 44, 45, 46, 61, 77, 79, 80, 111, 112, 113, 114, 115, 116

iTEBD	infinite time evolving block decimation 15, 97, 104, 105
KL	Kullback–Leibler 66
l-USC	layered uniform sequential circuits 77, 78, 97, 98, 99, 101, 103, 104, 105, 106, 107, 108, 109
MCMC	Markov chain Monte Carlo 44, 64
MERA	multi-scale entanglement renormalization ansatz 22, 90
MM	Moses move 7, 20, 21, 22, 23, 24, 25, 26, 27, 28, 31, 34, 35, 36, 38, 39, 40, 41, 43, 44, 45, 47, 61, 111, 112, 113
MPO	matrix-product operator 15, 21, 22, 31, 36, 37, 38
MPS	matrix-product state 2, 7, 8, 9, 10, 11, 12, 13, 14, 16, 19, 20, 21, 22, 26, 28, 31, 32, 36, 37, 43, 44, 45, 46, 63, 67, 68, 69, 74, 77, 80, 81, 82, 83, 85, 86, 87, 88, 89, 90, 92, 94, 97, 98, 111, 112, 115
NAQS	neural autoregressive quantum state 8, 64, 65, 66, 68, 69, 71, 72, 73, 74, 114
NISQ	noisy intermediate scale quantum 2, 79, 89, 92, 94, 115
NN	fully-connected feed-forward neural network 64
NQS	neural-network quantum state 2, 8, 64, 65, 67, 71, 73, 74, 113, 114, 115
NTU	neighborhood tensor update 16
OBC	open boundary conditions 39, 67
OC	orthogonality center 46, 48, 55
PBC	periodic boundary conditions 67
PEPS	projected entangled pair states 14, 15, 16, 17, 28, 45
QCTN	quantum circuit tensor network 77, 80, 115
QEC	quantum error correction 79
QMC	variational Monte Carlo 1, 60
QPE	quantum phase estimate 84
QPU	quantum processing unit 77, 78, 85, 93, 97
RBM	restricted Boltzmann machine 64
RBMQS	restricted Boltzmann machine quantum state 72, 73, 74
RNN	recurrent neural network 64, 115
SGD	stochastic gradient descent 17, 64, 65, 66, 67, 69, 73
SR	stochastic reconfiguration 17, 64, 73
SU	simple update 16, 27, 113
SVD	singular value decomposition 11, 12, 13, 15, 16, 19, 20, 22, 23, 24, 25, 26, 27, 32, 36, 38, 49, 59

## Acronyms

tDMRG	time-dependent density matrix renormalization group 90
TDVP	time-dependent variational principle 17, 90
TEBD	time-evolving block decimation 2, 7, 13, 16, 17, 26, 27, 28, 31, 37, 45, 58, 74, 83, 87, 90, 111
TNS	tensor network state 2, 3, 7, 9, 10, 11, 14, 20, 27, 28, 29, 37, 40, 44, 74, 111, 113, 114
TRG	tensor renormalization group 14
USC	uniform sequential circuits 97
VMC	variational Monte Carlo 2, 17, 64, 74, 114
VQE	variational quantum eigensolver 91, 92
VUMPS	variational uniform matrix-product state 15, 112
WTG	weighted trace gauge 28, 29

# Bibliography

- [1] Sheng-Hsuan Lin, Michael P. Zaletel, and Frank Pollmann. Efficient simulation of dynamics in two-dimensional quantum spin systems with isometric tensor networks. *Phys. Rev. B*, 106:245102, Dec 2022. doi: 10.1103/PhysRevB.106.245102. URL <https://link.aps.org/doi/10.1103/PhysRevB.106.245102>.
- [2] Yantao Wu, Sajant Anand, Sheng-Hsuan Lin, Frank Pollmann, and Michael P. Zaletel. Two-dimensional isometric tensor networks on an infinite strip. *Phys. Rev. B*, 107:245118, Jun 2023. doi: 10.1103/PhysRevB.107.245118. URL <https://link.aps.org/doi/10.1103/PhysRevB.107.245118>.
- [3] Sheng-Hsuan Lin and Frank Pollmann. Scaling of neural-network quantum states for time evolution. *physica status solidi (b)*, 259(5):2100172, 2022.
- [4] Sheng-Hsuan Lin, Rohit Dilip, Andrew G. Green, Adam Smith, and Frank Pollmann. Real- and imaginary-time evolution with compressed quantum circuits. *PRX Quantum*, 2:010342, Mar 2021. doi: 10.1103/PRXQuantum.2.010342. URL <https://link.aps.org/doi/10.1103/PRXQuantum.2.010342>.
- [5] Nikita Astrakhantsev, Sheng-Hsuan Lin, Frank Pollmann, and Adam Smith. Time evolution of uniform sequential circuits. *arXiv preprint arXiv:2210.03751*, 2022.
- [6] Giuseppe Carleo, Kenny Choo, Damian Hofmann, James ET Smith, Tom Westerhout, Fabien Alet, Emily J Davis, Stavros Efthymiou, Ivan Glasser, Sheng-Hsuan Lin, et al. Netket: A machine learning toolkit for many-body quantum systems. *SoftwareX*, 10:100311, 2019.
- [7] Sheng-Hsuan Lin, Olivier Kuijpers, Sebastian Peterhansl, and Frank Pollmann. Distributive pre-training of generative modeling using matrix-product states. *arXiv preprint arXiv:2306.14787*, 2023.
- [8] Horst L Stormer, Daniel C Tsui, and Arthur C Gossard. The fractional quantum hall effect. *Reviews of Modern Physics*, 71(2):S298, 1999.
- [9] Robert B Laughlin. Anomalous quantum hall effect: an incompressible quantum fluid with fractionally charged excitations. *Physical Review Letters*, 50(18):1395, 1983.
- [10] J George Bednorz and K Alex Müller. Possible high  $c$  superconductivity in the  $\text{Ba-Cu-O}$  system. *Zeitschrift für Physik B Condensed Matter*, 64(2):189–193, 1986.
- [11] Markus Greiner, Olaf Mandel, Theodor W Hänsch, and Immanuel Bloch. Collapse and revival of the matter wave field of a bose–einstein condensate. *Nature*, 419(6902):51–54, 2002.
- [12] Immanuel Bloch, Jean Dalibard, and Wilhelm Zwerger. Many-body physics with ultracold gases. *Reviews of modern physics*, 80(3):885, 2008.
- [13] Anatoli Polkovnikov, Krishnendu Sengupta, Alessandro Silva, and Mukund Vengalattore. Colloquium: Nonequilibrium dynamics of closed interacting quantum systems. *Rev. Mod. Phys.*, 83:863–883, Aug 2011. doi: 10.1103/RevModPhys.83.863. URL <https://link.aps.org/doi/10.1103/RevModPhys.83.863>.

## Bibliography

- [14] Martin Gärttner, Justin G Bohnet, Arghavan Safavi-Naini, Michael L Wall, John J Bollinger, and Ana Maria Rey. Measuring out-of-time-order correlations and multiple quantum spectra in a trapped-ion quantum magnet. *Nature Physics*, 13(8):781–786, 2017.
- [15] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, 2019.
- [16] Antoine Browaeys and Thierry Lahaye. Many-body physics with individually controlled rydberg atoms. *Nature Physics*, 16(2):132–142, 2020.
- [17] Denis M Basko, Igor L Aleiner, and Boris L Altshuler. Metal–insulator transition in a weakly interacting many-electron system with localized single-particle states. *Annals of physics*, 321(5):1126–1205, 2006.
- [18] Arijeet Pal and David A Huse. Many-body localization phase transition. *Physical review b*, 82(17):174411, 2010.
- [19] Michael Schreiber, Sean S Hodgman, Pranjali 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.
- [20] Jae-yoon Choi, Sebastian Hild, Johannes Zeiher, Peter Schauß, Antonio Rubio-Abadal, Tarik Yefsah, Vedika Khemani, David A Huse, Immanuel Bloch, and Christian Gross. Exploring the many-body localization transition in two dimensions. *Science*, 352(6293):1547–1552, 2016.
- [21] Jacob Smith, Aaron Lee, Philip Richerme, Brian Neyenhuis, Paul W Hess, Philipp Hauke, Markus Heyl, David A Huse, and Christopher Monroe. Many-body localization in a quantum simulator with programmable random disorder. *Nature Physics*, 12(10):907–911, 2016.
- [22] Hannes Bernien, Sylvain Schwartz, Alexander Keesling, Harry Levine, Ahmed Omran, Hannes Pichler, Soonwon Choi, Alexander S Zibrov, Manuel Endres, Markus Greiner, et al. Probing many-body dynamics on a 51-atom quantum simulator. *Nature*, 551(7682):579–584, 2017.
- [23] Christopher J Turner, Alexios A Michailidis, Dmitry A Abanin, Maksym Serbyn, and Zlatko Papić. Weak ergodicity breaking from quantum many-body scars. *Nature Physics*, 14(7):745–749, 2018.
- [24] Soonwon Choi, Christopher J Turner, Hannes Pichler, Wen Wei Ho, Alexios A Michailidis, Zlatko Papić, Maksym Serbyn, Mikhail D Lukin, and Dmitry A Abanin. Emergent su(2) dynamics and perfect quantum many-body scars. *Physical review letters*, 122(22):220603, 2019.
- [25] Luca D’Alessio, Yariv Kafri, Anatoli Polkovnikov, and Marcos Rigol. From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics. *Advances in Physics*, 65(3):239–362, 2016.
- [26] Christoph Berke, Evangelos Varvelis, Simon Trebst, Alexander Altland, and David P DiVincenzo. Transmon platform for quantum computing challenged by chaotic fluctuations. *Nature communications*, 13(1):2495, 2022.

- [27] Evangelos Varvelis and David P DiVincenzo. Perturbative analysis of quasi-periodic patterning of transmon quantum computers: Enhancement of many-body localization. *arXiv preprint arXiv:2212.03805*, 2022.
- [28] Anders W Sandvik. Computational studies of quantum spin systems. In *AIP Conference Proceedings*, volume 1297, pages 135–338. American Institute of Physics, 2010.
- [29] Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Physical review letters*, 94(17):170201, 2005.
- [30] Frank Verstraete, Valentin Murg, and J Ignacio Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008.
- [31] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.
- [32] Jacob C Bridgeman and Christopher T Chubb. Hand-waving and interpretive dance: an introductory course on tensor networks. *Journal of Physics A: Mathematical and Theoretical*, 50(22):223001, 2017.
- [33] Johannes Hauschild and Frank Pollmann. Efficient numerical simulations with tensor networks: Tensor network python (tenpy). *SciPost Physics Lecture Notes*, 2018.
- [34] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42. Springer, 2012.
- [35] J Ignacio Cirac, David Perez-Garcia, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Reviews of Modern Physics*, 93(4):045003, 2021.
- [36] Norbert Schuch, Michael M Wolf, Frank Verstraete, and J Ignacio Cirac. Entropy scaling and simulability by matrix product states. *Physical review letters*, 100(3):030504, 2008.
- [37] Matthew B Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024, 2007.
- [38] Jens Eisert, Marcus Cramer, and Martin B Plenio. Area laws for the entanglement entropy—a review. *arXiv preprint arXiv:0808.3773*, 2008.
- [39] Jens Eisert, Marcus Cramer, and Martin B Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277, 2010.
- [40] Frank Verstraete and J Ignacio Cirac. Matrix product states represent ground states faithfully. *Physical review b*, 73(9):094423, 2006.
- [41] Juan Ignacio Cirac, José Garre-Rubio, and David Pérez-García. Mathematical open problems in projected entangled pair states. *Revista Matemática Complutense*, 32:579–599, 2019.
- [42] Fernando GSL Brandao and Marcus Cramer. Entanglement area law from specific heat capacity. *Physical Review B*, 92(11):115134, 2015.
- [43] Anurag Anshu, Itai Arad, and David Gosset. An area law for 2d frustration-free spin systems. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 12–18, 2022.

## Bibliography

- [44] Zeph Landau, Umesh Vazirani, and Thomas Vidick. A polynomial time algorithm for the ground state of one-dimensional gapped local hamiltonians. *Nature Physics*, 11(7):566–569, 2015.
- [45] Mark Fannes, B Nachtergaele, and RF Werner. Abundance of translation invariant pure states on quantum spin chains. *letters in mathematical physics*, 25(3):249–258, 1992.
- [46] D Perez-Garcia, F Verstraete, MM Wolf, and JI Cirac. Matrix product state representations. *Quantum Information & Computation*, 7(5):401–430, 2007.
- [47] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.
- [48] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14):147902, 2003.
- [49] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Physical review letters*, 93(4):040502, 2004.
- [50] Steven R White and Adrian E Feiguin. Real-time evolution using the density matrix renormalization group. *Physical review letters*, 93(7):076401, 2004.
- [51] Andrew John Daley, Corinna Kollath, Ulrich Schollwöck, and Guifré Vidal. Time-dependent density-matrix renormalization-group using adaptive effective hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, 2004.
- [52] C. Hubig, A. Bohrdt, M. Knap, F. Grusdt, and J. I. Cirac. Evaluation of time-dependent correlators after a local quench in iPEPS: hole motion in the t-J model. *SciPost Phys.*, 8:21, 2020. doi: 10.21468/SciPostPhys.8.2.021. URL <https://scipost.org/10.21468/SciPostPhys.8.2.021>.
- [53] Piotr Czarnik and Jacek Dziarmaga. Time evolution of an infinite projected entangled pair state: An algorithm from first principles. *Phys. Rev. B*, 98:045110, Jul 2018. doi: 10.1103/PhysRevB.98.045110. URL <https://link.aps.org/doi/10.1103/PhysRevB.98.045110>.
- [54] Sebastian Paeckel, Thomas Köhler, Andreas Swoboda, Salvatore R Manmana, Ulrich Schollwöck, and Claudius Hubig. Time-evolution methods for matrix-product states. *Annals of Physics*, 411:167998, 2019.
- [55] Pasquale Calabrese and John Cardy. Evolution of entanglement entropy in one-dimensional systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(04):P04010, 2005.
- [56] Hyungwon Kim and David A. Huse. Ballistic spreading of entanglement in a diffusive nonintegrable system. *Phys. Rev. Lett.*, 111:127205, Sep 2013. doi: 10.1103/PhysRevLett.111.127205. URL <https://link.aps.org/doi/10.1103/PhysRevLett.111.127205>.
- [57] Tomaž Prosen and Marko Žnidarič. Matrix product simulations of non-equilibrium steady states of quantum spin chains. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(02):P02035, 2009.



- [58] Jian Cui, J Ignacio Cirac, and Mari Carmen Bañuls. Variational matrix product operators for the steady state of dissipative quantum systems. *Physical review letters*, 114(22):220601, 2015.
- [59] Tibor Rakovszky, CW von Keyserlingk, and Frank Pollmann. Dissipation-assisted operator evolution method for capturing hydrodynamic transport. *Physical Review B*, 105(7):075131, 2022.
- [60] Frank Verstraete and J Ignacio Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*, 2004.
- [61] Harald Niggemann, Andreas Klümper, and Johannes Zittartz. Quantum phase transition in spin-3/2 systems on the hexagonal lattice—optimum ground state approach. *Zeitschrift für Physik B Condensed Matter*, 104:103–110, 1997.
- [62] Tomotoshi Nishino and Kouichi Okunishi. A density matrix algorithm for 3d classical models. *Journal of the Physical Society of Japan*, 67(9):3066–3072, 1998.
- [63] G Sierra. The density matrix renormalization group, quantum groups and conformal field theory. In *Proceedings of the Workshop on the Exact Renormalization Group*. World Scientific, 1998.
- [64] Michael Lubasch, J Ignacio Cirac, and Mari-Carmen Banuls. Algorithms for finite projected entangled pair states. *Physical Review B*, 90(6):064425, 2014.
- [65] Ho N Phien, Johann A Bengua, Hoang D Tuan, Philippe Corboz, and Román Orús. Infinite projected entangled pair states algorithm improved: Fast full update and gauge fixing. *Physical Review B*, 92(3):035142, 2015.
- [66] Laurens Vanderstraeten, Jutho Haegeman, Philippe Corboz, and Frank Verstraete. Gradient methods for variational optimization of projected entangled-pair states. *Physical Review B*, 94(15), Oct 2016. ISSN 2469-9969. doi: 10.1103/physrevb.94.155123. URL <http://dx.doi.org/10.1103/PhysRevB.94.155123>.
- [67] Philippe Corboz. Variational optimization with infinite projected entangled-pair states. *Physical Review B*, 94(3), Jul 2016. ISSN 2469-9969. doi: 10.1103/physrevb.94.035133. URL <http://dx.doi.org/10.1103/PhysRevB.94.035133>.
- [68] Hai-Jun Liao, Jin-Guo Liu, Lei Wang, and Tao Xiang. Differentiable programming tensor networks. *Physical Review X*, 9(3):031041, 2019.
- [69] Wen-Yuan Liu, Shao-Jun Dong, Yong-Jian Han, Guang-Can Guo, and Lixin He. Gradient optimization of finite projected entangled pair states. *Physical Review B*, 95(19):195154, 2017.
- [70] Wen-Yuan Liu, Yi-Zhen Huang, Shou-Shu Gong, and Zheng-Cheng Gu. Accurate simulation for finite projected entangled pair states in two dimensions. *Physical Review B*, 103(23), Jun 2021. ISSN 2469-9969. doi: 10.1103/physrevb.103.235155. URL <http://dx.doi.org/10.1103/PhysRevB.103.235155>.
- [71] Tom Vieijra, Jutho Haegeman, Frank Verstraete, and Laurens Vanderstraeten. Direct sampling of projected entangled-pair states. *arXiv preprint arXiv:2109.07356*, 2021.
- [72] Norbert Schuch, Michael M Wolf, Frank Verstraete, and J Ignacio Cirac. Computational complexity of projected entangled pair states. *Physical review letters*, 98(14):140506, 2007.

## Bibliography

- [73] Tomotoshi Nishino and Kouichi Okunishi. Corner transfer matrix renormalization group method. *Journal of the Physical Society of Japan*, 65(4):891–894, 1996.
- [74] Román Orús and Guifré Vidal. Simulation of two-dimensional quantum systems on an infinite lattice revisited: Corner transfer matrix for tensor contraction. *Physical Review B*, 80(9):094403, 2009.
- [75] Michael P Zaletel and Frank Pollmann. Isometric tensor network states in two dimensions. *Physical review letters*, 124(3):037201, 2020.
- [76] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.
- [77] Markus Schmitt and Markus Heyl. Quantum many-body dynamics in two dimensions with artificial neural networks. *Physical Review Letters*, 125(10):100503, 2020.
- [78] 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):020503, 2020.
- [79] 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. doi: 10.1103/PhysRevX.8.011006. URL <https://link.aps.org/doi/10.1103/PhysRevX.8.011006>.
- [80] Dong-Ling Deng, Xiaopeng Li, and S Das Sarma. Quantum entanglement in neural network states. *Physical Review X*, 7(2):021021, 2017.
- [81] Yoav Levine, Or Sharir, Nadav Cohen, and Amnon Shashua. Quantum entanglement in deep learning architectures. *Physical review letters*, 122(6):065301, 2019.
- [82] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [83] Sepehr Ebadi, Tout T Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, et al. Quantum phases of matter on a 256-atom programmable quantum simulator. *Nature*, 595(7866):227–232, 2021.
- [84] Andrew J Daley, Immanuel Bloch, Christian Kokail, Stuart Flannigan, Natalie Pearson, Matthias Troyer, and Peter Zoller. Practical quantum advantage in quantum simulation. *Nature*, 607(7920):667–676, 2022.
- [85] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [86] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.
- [87] Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What limits the simulation of quantum computers? *Phys. Rev. X*, 10:041038, Nov 2020. doi: 10.1103/PhysRevX.10.041038. URL <https://link.aps.org/doi/10.1103/PhysRevX.10.041038>.
- [88] Daniel Stilck Franca and Raul Garcia-Patron. Limitations of optimization algorithms on noisy quantum devices. *arXiv preprint arXiv:2009.05532*, 2020.

- [89] 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):4213, 2014.
- [90] Esteban A Martinez, Christine A Muschik, Philipp Schindler, Daniel Nigg, Alexander Erhard, Markus Heyl, Philipp Hauke, Marcello Dalmonte, Thomas Monz, Peter Zoller, et al. Real-time dynamics of lattice gauge theories with a few-qubit quantum computer. *Nature*, 534(7608):516–519, 2016.
- [91] Henry Lamm and Scott Lawrence. Simulation of nonequilibrium dynamics on a quantum computer. *Physical review letters*, 121(17):170501, 2018.
- [92] Adam Smith, Bernhard Jobst, Andrew G Green, and Frank Pollmann. Crossing a topological phase transition with a quantum computer. *arXiv preprint arXiv:1910.05351*, 2019.
- [93] Ying Li and Simon C Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7(2):021050, 2017.
- [94] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5(1):1–6, 2019.
- [95] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational algorithms for linear algebra. *arXiv preprint arXiv:1909.03898*, 2019.
- [96] Marcello Benedetti, Mattia Fiorentini, and Michael Lubasch. Hardware-efficient variational quantum algorithms for time evolution. *Physical Review Research*, 3(3):033083, 2021.
- [97] Seunghoon Lee, Joonho Lee, Huanchen Zhai, Yu Tong, Alexander M Dalzell, Ashutosh Kumar, Phillip Helms, Johnnie Gray, Zhi-Hao Cui, Wenyuan Liu, et al. Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry. *Nature Communications*, 14(1):1952, 2023.
- [98] Richard P Feynman. Simulating physics with computers. *Int. J. Theor. Phys*, 21(6/7), 1982.
- [99] Frank Verstraete, Juan J Garcia-Ripoll, and Juan Ignacio Cirac. Matrix product density operators: Simulation of finite-temperature and dissipative systems. *Physical review letters*, 93(20):207204, 2004.
- [100] Adrian E Feiguin and Steven R White. Finite-temperature density matrix renormalization using an enlarged hilbert space. *Physical Review B*, 72(22):220401, 2005.
- [101] Wilhelm Kadow, Frank Pollmann, and Michael Knap. Isometric tensor network representations of two-dimensional thermal states. *Physical Review B*, 107(20):205106, 2023.
- [102] EM Stoudenmire and Steven R White. Minimally entangled typical thermal state algorithms. *New Journal of Physics*, 12(5):055026, 2010.
- [103] Matthias Gohlke, Ruben Verresen, Roderich Moessner, and Frank Pollmann. Dynamics of the kitaev-heisenberg model. *Physical review letters*, 119(15):157203, 2017.
- [104] Ruben Verresen, Frank Pollmann, and Roderich Moessner. Quantum dynamics of the square-lattice heisenberg model. *Physical Review B*, 98(15):155102, 2018.

## Bibliography

- [105] Radu Coldea, DA Tennant, EM Wheeler, E Wawrzynska, D Prabhakaran, M Telling, K Habicht, P Smeibidl, and K Kiefer. Quantum criticality in an ising chain: experimental evidence for emergent e8 symmetry. *Science*, 327(5962):177–180, 2010.
- [106] Sirui Lu, Mari Carmen Banuls, and J Ignacio Cirac. Algorithms for quantum simulation at finite energies. *PRX Quantum*, 2(2):020321, 2021.
- [107] Yilun Yang, J Ignacio Cirac, and Mari Carmen Bañuls. Classical algorithms for many-body quantum systems at finite energies. *Physical Review B*, 106(2):024307, 2022.
- [108] Alexander Schuckert, Annabelle Bohrdt, Eleanor Crane, and Michael Knap. Probing finite-temperature observables in quantum simulators of spin systems with short-time dynamics. *Physical Review B*, 107(14):L140410, 2023.
- [109] Norbert Schuch, Michael M. Wolf, Frank Verstraete, and J. Ignacio Cirac. Entropy scaling and simulability by matrix product states. *Physical Review Letters*, 100(3), Jan 2008. ISSN 1079-7114. doi: 10.1103/physrevlett.100.030504. URL <http://dx.doi.org/10.1103/PhysRevLett.100.030504>.
- [110] Yimin Ge and Jens Eisert. Area laws and efficient descriptions of quantum many-body states. *New Journal of Physics*, 18(8):083026, 2016.
- [111] J. Dubail and N. Read. Tensor network trial states for chiral topological phases in two dimensions and a no-go theorem in any dimension. *Phys. Rev. B*, 92:205307, Nov 2015. doi: 10.1103/PhysRevB.92.205307. URL [<https://link.aps.org/doi/10.1103/PhysRevB.92.205307>] (<https://link.aps.org/doi/10.1103/PhysRevB.92.205307>).
- [112] N. Read. Compactly supported wannier functions and algebraic  $k$ -theory. *Phys. Rev. B*, 95:115309, Mar 2017. doi: 10.1103/PhysRevB.95.115309. URL [<https://link.aps.org/doi/10.1103/PhysRevB.95.115309>] (<https://link.aps.org/doi/10.1103/PhysRevB.95.115309>).
- [113] T. B. Wahl, H.-H. Tu, N. Schuch, and J. I. Cirac. Projected entangled-pair states can describe chiral topological states. *Phys. Rev. Lett.*, 111:236805, Dec 2013. doi: 10.1103/PhysRevLett.111.236805. URL [<https://link.aps.org/doi/10.1103/PhysRevLett.111.236805>] (<https://link.aps.org/doi/10.1103/PhysRevLett.111.236805>).
- [114] Shuo Yang, Thorsten B. Wahl, Hong-Hao Tu, Norbert Schuch, and J. Ignacio Cirac. Chiral projected entangled-pair state with topological order. *Physical Review Letters*, 114(10), Mar 2015. ISSN 1079-7114. doi: 10.1103/physrevlett.114.106803. URL <http://dx.doi.org/10.1103/PhysRevLett.114.106803>.
- [115] Didier Poilblanc, J. Ignacio Cirac, and Norbert Schuch. Chiral topological spin liquids with projected entangled pair states. *Phys. Rev. B*, 91:224431, Jun 2015. doi: 10.1103/PhysRevB.91.224431. URL [<https://link.aps.org/doi/10.1103/PhysRevB.91.224431>] (<https://link.aps.org/doi/10.1103/PhysRevB.91.224431>).
- [116] Juraj Hasik, Maarten Van Damme, Didier Poilblanc, and Laurens Vanderstraeten. Simulating chiral spin liquids with projected entangled-pair states. *Physical Review Letters*, 129(17):177201, 2022.

- [117] Mark Fannes, Bruno Nachtergaele, and Reinhard F Werner. Finitely correlated states on quantum spin chains. *Communications in mathematical physics*, 144:443–490, 1992.
- [118] Stellan Östlund and Stefan Rommer. Thermodynamic limit of density matrix renormalization. *Physical review letters*, 75(19):3537, 1995.
- [119] Jorge Dukelsky, Miguel A Martín-Delgado, Tomotoshi Nishino, and Germán Sierra. Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains. *Europhysics letters*, 43(4):457, 1998.
- [120] Jacob Jordan, Roman Orús, Guifre Vidal, Frank Verstraete, and J Ignacio Cirac. Classical simulation of infinite-size quantum lattice systems in two spatial dimensions. *Physical review letters*, 101(25):250602, 2008.
- [121] Itai Arad and Zeph Landau. Quantum computation and the evaluation of tensor networks. *SIAM Journal on Computing*, 39(7):3089–3121, 2010.
- [122] Jonas Haferkamp, Dominik Hangleiter, Jens Eisert, and Marek Gluza. Contracting projected entangled pair states is average-case hard. *Physical Review Research*, 2(1):013010, 2020.
- [123] Arturo Acuaviva, Visu Makam, Harold Nieuwboer, David Pérez-García, Friedrich Sittner, Michael Walter, and Freek Witteveen. The minimal canonical form of a tensor network. *arXiv preprint arXiv:2209.14358*, 2022.
- [124] Shi-Ju Ran, Emanuele Tirrito, Cheng Peng, Xi Chen, Luca Tagliacozzo, Gang Su, and Maciej Lewenstein. *Tensor network contractions: methods and applications to quantum many-body systems*. Springer Nature, 2020.
- [125] Michael Levin and Cody P Nave. Tensor renormalization group approach to two-dimensional classical lattice models. *Physical review letters*, 99(12):120601, 2007.
- [126] Zhi-Yuan Xie, Hong-Chen Jiang, Qinjun N Chen, Zheng-Yu Weng, and Tao Xiang. Second renormalization of tensor-network states. *Physical review letters*, 103(16):160601, 2009.
- [127] Zhi-Yuan Xie, Jing Chen, Ming-Pu Qin, Jing W Zhu, Li-Ping Yang, and Tao Xiang. Coarse-graining renormalization by higher-order singular value decomposition. *Physical Review B*, 86(4):045139, 2012.
- [128] Glen Evenbly and Guifre Vidal. Tensor network renormalization yields the multiscale entanglement renormalization ansatz. *Physical review letters*, 115(20):200401, 2015.
- [129] Shuo Yang, Zheng-Cheng Gu, and Xiao-Gang Wen. Loop optimization for tensor network renormalization. *Physical review letters*, 118(11):110504, 2017.
- [130] Matthias Bal, Michaël Mariën, Jutho Haegeman, and Frank Verstraete. Renormalization group flows of hamiltonians using tensor networks. *Physical Review Letters*, 118(25):250602, 2017.
- [131] Leo P Kadanoff and Anthony Houghton. Numerical evaluations of the critical properties of the two-dimensional ising model. *Physical Review B*, 11(1):377, 1975.
- [132] Daniel E Parker, Xiangyu Cao, and Michael P Zaletel. Local matrix product operators: Canonical form, compression, and control theory. *Physical Review B*, 102(3):035147, 2020.

## Bibliography

- [133] MT Fishman, Laurens Vanderstraeten, V Zauner-Stauber, Jutho Haegeman, and Frank Verstraete. Faster methods for contracting infinite two-dimensional tensor networks. *Physical Review B*, 98(23):235148, 2018.
- [134] Tomotoshi Nishino and Kouichi Okunishi. Corner transfer matrix algorithm for classical renormalization group. *Journal of the Physical Society of Japan*, 66(10):3040–3047, 1997.
- [135] Rodney J Baxter. Dimers on a rectangular lattice. *Journal of Mathematical Physics*, 9(4):650–654, 1968.
- [136] Rodney J Baxter. Variational approximations for square lattice models in statistical mechanics. *Journal of Statistical Physics*, 19:461–478, 1978.
- [137] Rodney J Baxter. *Exactly solved models in statistical mechanics*. Elsevier, 2016.
- [138] Steven R White. Density-matrix algorithms for quantum renormalization groups. *Physical review b*, 48(14):10345, 1993.
- [139] Johnnie Gray and Stefanos Kourtis. Hyper-optimized tensor network contraction. *Quantum*, 5:410, 2021.
- [140] Feng Pan and Pan Zhang. Simulating the sycamore quantum supremacy circuits. *arXiv preprint arXiv:2103.03074*, 2021.
- [141] Johnnie Gray and Garnet Kin Chan. Hyper-optimized compressed contraction of tensor networks with arbitrary geometry. *arXiv preprint arXiv:2206.07044*, 2022.
- [142] Roy Alkabetz and Itai Arad. Tensor networks contraction and the belief propagation algorithm. *Physical Review Research*, 3(2):023073, 2021.
- [143] Elina Robeva and Anna Seigal. Duality of graphical models and tensor networks. *Information and Inference: A Journal of the IMA*, 8(2):273–288, 2019.
- [144] Chu Guo, Dario Poletti, and Itai Arad. Block belief propagation algorithm for 2d tensor networks. *arXiv preprint arXiv:2301.05844*, 2023.
- [145] Yijia Wang, Yuwen Ebony Zhang, Feng Pan, and Pan Zhang. Tensor network message passing. *arXiv preprint arXiv:2305.01874*, 2023.
- [146] Hong-Chen Jiang, Zheng-Yu Weng, and Tao Xiang. Accurate determination of tensor network state of quantum lattice models in two dimensions. *Physical review letters*, 101(9):090603, 2008.
- [147] Ling Wang and Frank Verstraete. Cluster update for tensor network states. *arXiv preprint arXiv:1110.4362*, 2011.
- [148] Jacek Dziarmaga. Time evolution of an infinite projected entangled pair state: Neighborhood tensor update. *Physical Review B*, 104(9):094411, 2021.
- [149] Michael Lubasch, J Ignacio Cirac, and Mari-Carmen Banuls. Unifying projected entangled pair state contractions. *New Journal of Physics*, 16(3):033014, 2014.
- [150] Federico Becca and Sandro Sorella. *Quantum Monte Carlo approaches for correlated systems*. Cambridge University Press, 2017.
- [151] Susanne Richter. *Construction of states on two-dimensional lattices and quantum cellular automata*. Shaker, 1995.

- [152] Tomohiro Soejima, Karthik Siva, Nick Bultinck, Shubhayu Chatterjee, Frank Pollmann, Michael P Zaletel, et al. Isometric tensor network representation of string-net liquids. *Physical Review B*, 101(8):085117, 2020.
- [153] Michael A Levin and Xiao-Gang Wen. String-net condensation: A physical mechanism for topological phases. *Physical Review B*, 71(4):045110, 2005.
- [154] Chien-Hung Lin and Michael Levin. Generalizations and limitations of string-net models. *Physical Review B*, 89(19):195130, 2014.
- [155] Lucas Slattery and Bryan K. Clark. Quantum circuits for two-dimensional isometric tensor networks, 2021.
- [156] Zhi-Yuan Wei, Daniel Malz, and J. Ignacio Cirac. Sequential generation of projected entangled-pair states. *Phys. Rev. Lett.*, 128:010607, Jan 2022. doi: 10.1103/PhysRevLett.128.010607. URL <https://link.aps.org/doi/10.1103/PhysRevLett.128.010607>.
- [157] John C Gower and Garnt B Dijkstrahuis. *Procrustes problems*, volume 30. OUP Oxford, 2004.
- [158] Glen Evenbly and Guifré Vidal. Algorithms for entanglement renormalization. *Physical Review B*, 79(14):144108, 2009.
- [159] Glen Evenbly and Guifré Vidal. Algorithms for entanglement renormalization: boundaries, impurities and interfaces. *Journal of Statistical Physics*, 157(4-5):931–978, 2014.
- [160] Philippe Corboz, Román Orús, Bela Bauer, and Guifré Vidal. Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states. *Phys. Rev. B*, 81:165104, Apr 2010. doi: 10.1103/PhysRevB.81.165104. URL <https://link.aps.org/doi/10.1103/PhysRevB.81.165104>.
- [161] Valentin Murg, Frank Verstraete, and J Ignacio Cirac. Variational study of hard-core bosons in a two-dimensional optical lattice using projected entangled pair states. *Physical Review A*, 75(3):033605, 2007.
- [162] Reza Haghshenas, Matthew J O’Rourke, and Garnet Kin-Lic Chan. Conversion of projected entangled pair states into a canonical form. *Physical Review B*, 100(5):054404, 2019.
- [163] Katharine Hyatt and E Miles Stoudenmire. Dmrg approach to optimizing two-dimensional tensor networks. *arXiv preprint arXiv:1908.08833*, 2019.
- [164] Maurits SJ Tepaske and David J Luitz. Three-dimensional isometric tensor networks. *Physical Review Research*, 3(2):023236, 2021.
- [165] Yifan Zhang and Edgar Solomonik. On stability of tensor networks and canonical forms. *arXiv preprint arXiv:2001.01191*, 2020.
- [166] D Pérez-García, M Sanz, CE Gonzalez-Guillen, Michael M Wolf, and J Ignacio Cirac. A canonical form for projected entangled pair states and applications. *arXiv preprint arXiv:0908.1674*, 2009.
- [167] Glen Evenbly. Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops. *Physical Review B*, 98(8):085155, 2018.



## Bibliography

- [168] Zheng-Cheng Gu and Xiao-Gang Wen. Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order. *Physical Review B*, 80(15):155131, 2009.
- [169] Christian Schön, Enrique Solano, Frank Verstraete, J Ignacio Cirac, and Michael M Wolf. Sequential generation of entangled multiqubit states. *Physical review letters*, 95(11):110503, 2005.
- [170] Mari-Carmen Banuls, David Pérez-García, Michael M Wolf, Frank Verstraete, and J Ignacio Cirac. Sequentially generated states for the study of two-dimensional systems. *Physical Review A*, 77(5):052306, 2008.
- [171] Steven T Smith. Optimization techniques on riemannian manifolds. *Fields institute communications*, 3(3):113–135, 1994.
- [172] Traian E Abrudan, Jan Eriksson, and Visa Koivunen. Steepest descent algorithms for optimization under unitary matrix constraint. *IEEE Transactions on Signal Processing*, 56(3):1134–1147, 2008.
- [173] Andrew J Ferris and Guifre Vidal. Variational monte carlo with the multiscale entanglement renormalization ansatz. *Physical Review B*, 85(16):165147, 2012.
- [174] Markus Hauru, Maarten Van Damme, and Jutho Haegeman. Riemannian optimization of isometric tensor networks. *SciPost Phys*, 10:040, 2021.
- [175] I Luchnikov, M Krechetov, and S Filippov. Riemannian optimization and automatic differentiation for complex quantum architectures. *arXiv preprint arXiv:2007.01287*, 2020.
- [176] Johannes Hauschild, Eyal Leviatan, Jens H Bardarson, Ehud Altman, Michael P Zaletel, and Frank Pollmann. Finding purifications with minimal entanglement. *Physical Review B*, 98(23):235163, 2018.
- [177] Christian Krumnow, Libor Veis, Örs Legeza, and Jens Eisert. Fermionic orbital optimization in tensor network states. *Physical review letters*, 117(21):210402, 2016.
- [178] C Krumnow, J Eisert, and Ö Legeza. Towards overcoming the entanglement barrier when simulating long-time evolution. *arXiv preprint arXiv:1904.11999*, 2019.
- [179] Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *arXiv preprint arXiv:2002.01113*, 2020.
- [180] Elijah Polak and Gerard Ribiere. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 3(R1):35–43, 1969.
- [181] Daisuke Kadoh and Katsumasa Nakayama. Renormalization group on a triad network. *arXiv preprint arXiv:1912.02414*, 2019.
- [182] Iztok Pižorn and Frank Verstraete. Fermionic implementation of projected entangled pair states algorithm. *Physical Review B*, 81(24):245110, 2010.
- [183] Daniel E. Parker, Xiangyu Cao, and Michael P. Zaletel. Local matrix product operators: Canonical form, compression, and control theory. *Phys. Rev. B*, 102:035147, Jul 2020. doi: 10.1103/PhysRevB.102.035147. URL <https://link.aps.org/doi/10.1103/PhysRevB.102.035147>.



- [184] Heiko Rieger and Naoki Kawashima. Application of a continuous time cluster algorithm to the two-dimensional random quantum ising ferromagnet. *The European Physical Journal B-Condensed Matter and Complex Systems*, 9(2):233–236, 1999.
- [185] Henk W. J. Blöte and Youjin Deng. Cluster monte carlo simulation of the transverse ising model. *Phys. Rev. E*, 66:066110, Dec 2002. doi: 10.1103/PhysRevE.66.066110. URL <https://link.aps.org/doi/10.1103/PhysRevE.66.066110>.
- [186] Laurens Vanderstraeten, Jutho Haegeman, and Frank Verstraete. Simulating excitation spectra with projected entangled-pair states. *Physical Review B*, 99(16):165121, 2019.
- [187] Alexei Kitaev. Anyons in an exactly solved model and beyond. *Annals of Physics*, 321(1):2–111, 2006.
- [188] J. Knolle, D. L. Kovrizhin, J. T. Chalker, and R. Moessner. Dynamics of a two-dimensional quantum spin liquid: Signatures of emergent majorana fermions and fluxes. *Phys. Rev. Lett.*, 112:207203, May 2014. doi: 10.1103/PhysRevLett.112.207203. URL <https://link.aps.org/doi/10.1103/PhysRevLett.112.207203>.
- [189] J. Knolle, D. L. Kovrizhin, J. T. Chalker, and R. Moessner. Dynamics of fractionalization in quantum spin liquids. *Phys. Rev. B*, 92:115127, Sep 2015. doi: 10.1103/PhysRevB.92.115127. URL <https://link.aps.org/doi/10.1103/PhysRevB.92.115127>.
- [190] Fabian Zschocke and Matthias Vojta. Physical states and finite-size effects in kitaev’s honeycomb model: Bond disorder, spin excitations, and nmr line shape. *Physical Review B*, 92(1):014403, 2015.
- [191] G. Vidal. Classical simulation of infinite-size quantum lattice systems in one spatial dimension. *Phys. Rev. Lett.*, 98:070201, Feb 2007. doi: 10.1103/PhysRevLett.98.070201. URL <https://link.aps.org/doi/10.1103/PhysRevLett.98.070201>.
- [192] Guifre Vidal. Entanglement renormalization. *Physical review letters*, 99(22):220405, 2007.
- [193] Guifré Vidal. Class of quantum many-body states that can be efficiently simulated. *Physical review letters*, 101(11):110501, 2008.
- [194] Luca Tagliacozzo, Thiago R de Oliveira, S Iblisdir, and JI Latorre. Scaling of entanglement support for matrix product states. *Physical review b*, 78(2):024410, 2008.
- [195] Frank Pollmann, Subroto Mukerjee, Ari M Turner, and Joel E Moore. Theory of finite-entanglement scaling at one-dimensional quantum critical points. *Physical review letters*, 102(25):255701, 2009.
- [196] Bogdan Pirvu, Guifre Vidal, Frank Verstraete, and Luca Tagliacozzo. Matrix product states for critical spin chains: Finite-size versus finite-entanglement scaling. *Physical review b*, 86(7):075117, 2012.
- [197] Bram Vanhecke, Jutho Haegeman, Karel Van Acoleyen, Laurens Vanderstraeten, and Frank Verstraete. Scaling hypothesis for matrix product states. *Physical Review Letters*, 123(25):250604, 2019.
- [198] Michael Rader and Andreas M Läuchli. Finite correlation length scaling in lorentz-invariant gapless ipeps wave functions. *Physical Review X*, 8(3):031030, 2018.

## Bibliography

- [199] Philippe Corboz, Piotr Czarnik, Geert Kapteijns, and Luca Tagliacozzo. Finite correlation length scaling with infinite projected entangled-pair states. *Physical Review X*, 8(3):031031, 2018.
- [200] Bram Vanhecke, Juraj Hasik, Frank Verstraete, and Laurens Vanderstraeten. Scaling hypothesis for projected entangled-pair states. *Physical Review Letters*, 129(20):200601, 2022.
- [201] Juan Osorio Iregui, Matthias Troyer, and Philippe Corboz. Infinite matrix product states versus infinite projected entangled-pair states on the cylinder: A comparative study. *Physical Review B*, 96(11), Sep 2017. ISSN 2469-9969. doi: 10.1103/physrevb.96.115113. URL <http://dx.doi.org/10.1103/PhysRevB.96.115113>.
- [202] Johannes Motruk, Michael P. Zaletel, Roger S. K. Mong, and Frank Pollmann. Density matrix renormalization group on a cylinder in mixed real and momentum space. *Physical Review B*, 93(15), Apr 2016. ISSN 2469-9969. doi: 10.1103/physrevb.93.155139. URL <http://dx.doi.org/10.1103/PhysRevB.93.155139>.
- [203] Laurens Vanderstraeten, Jutho Haegeman, and Frank Verstraete. Tangent-space methods for uniform matrix product states. *SciPost Phys. Lect. Notes*, page 7, 2019. doi: 10.21468/SciPostPhysLectNotes.7. URL <https://scipost.org/10.21468/SciPostPhysLectNotes.7>.
- [204] Bram Vanhecke, Maarten Van Damme, Jutho Haegeman, Laurens Vanderstraeten, and Frank Verstraete. Tangent-space methods for truncating uniform mps. *SciPost Physics Core*, 4(1):004, 2021.
- [205] Iliia A Luchnikov, Alexander Ryzhov, Sergey N Filippov, and Henni Ouerdane. Qgopt: Riemannian optimization for quantum technologies. *arXiv preprint arXiv:2011.01894*, 2020.
- [206] Hao Xie, Jin-Guo Liu, and Lei Wang. Automatic differentiation of dominant eigensolver and its applications in quantum physics. *Phys. Rev. B*, 101:245139, Jun 2020. doi: 10.1103/PhysRevB.101.245139. URL <https://link.aps.org/doi/10.1103/PhysRevB.101.245139>.
- [207] Tomohiro Hashizume, Ian P. McCulloch, and Jad C. Halimeh. Dynamical phase transitions in the two-dimensional transverse-field ising model. *Phys. Rev. Research*, 4:013250, Mar 2022. doi: 10.1103/PhysRevResearch.4.013250. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.4.013250>.
- [208] R. Orús and G. Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Phys. Rev. B*, 78:155117, Oct 2008. doi: 10.1103/PhysRevB.78.155117. URL <https://link.aps.org/doi/10.1103/PhysRevB.78.155117>.
- [209] H. C. Jiang, Z. Y. Weng, and T. Xiang. Accurate determination of tensor network state of quantum lattice models in two dimensions. *Physical Review Letters*, 101(9), Aug 2008. ISSN 1079-7114. doi: 10.1103/physrevlett.101.090603. URL <http://dx.doi.org/10.1103/PhysRevLett.101.090603>.
- [210] Bela Bauer, LD Carr, Hans Gerd Evertz, Adrian Feiguin, J Freire, S Fuchs, Lukas Gamper, Jan Gukelberger, E Gull, Siegfried Guertler, et al. The alps project release 2.0: open source software for strongly correlated systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(05):P05001, 2011.

- [211] Michael Foss-Feig, David Hayes, Joan M. Dreiling, Caroline Figgatt, John P. Gaebler, Steven A. Moses, Juan M. Pino, and Andrew C. Potter. Holographic quantum algorithms for simulating correlated spin systems. *Physical Review Research*, 3(3), Jul 2021. ISSN 2643-1564. doi: 10.1103/physrevresearch.3.033002. URL <http://dx.doi.org/10.1103/PhysRevResearch.3.033002>.
- [212] Reza Haghshenas, Johnnie Gray, Andrew C Potter, and Garnet Kin-Lic Chan. Variational power of quantum circuit tensor networks. *Physical Review X*, 12(1):011047, 2022.
- [213] Fergus Barratt, James Dborin, Matthias Bal, Vid Stojevic, Frank Pollmann, and Andrew G Green. Parallel quantum simulation of large systems on small nisq computers. *npj Quantum Information*, 7(1):79, 2021.
- [214] James Dborin, Vinul Wimalaweera, Fergus Barratt, Eric Ostby, Thomas E O’Brien, and Andrew G Green. Simulating groundstate and dynamical quantum phase transitions on a superconducting quantum computer. *Nature Communications*, 13(1):5977, 2022.
- [215] Sajant Anand, Johannes Hauschild, Yuxuan Zhang, Andrew C Potter, and Michael P Zaletel. Holographic quantum simulation of entanglement renormalization circuits. *arXiv preprint arXiv:2203.00886*, 2022.
- [216] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [217] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [218] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [219] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.
- [220] Taehwan Kim and Tülay Adalı. Universal approximation of fully complex feedforward neural networks. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–973. IEEE, 2002.
- [221] Felix Voigtlaender. The universal approximation theorem for complex-valued neural networks. *arXiv preprint arXiv:2012.03351*, 2020.
- [222] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
- [223] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798, 2007.
- [224] Zi Cai and Jinguo Liu. Approximating quantum many-body wave functions using artificial neural networks. *Physical Review B*, 97(3):035116, 2018.
- [225] Kenny Choo, Titus Neupert, and Giuseppe Carleo. Two-dimensional frustrated  $J_1 - J_2$  model studied with neural network quantum states. *Phys. Rev. B*, 100:125124, Sep 2019. doi: 10.1103/PhysRevB.100.125124. URL <https://link.aps.org/doi/10.1103/PhysRevB.100.125124>.

## Bibliography

- [226] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [227] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [228] Mohamed Hibat-Allah, Martin Ganahl, Lauren E Hayward, Roger G Melko, and Juan Carrasquilla. Recurrent neural network wave functions. *Physical Review Research*, 2(2):023358, 2020.
- [229] Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K Clark. Probabilistic formulation of open quantum many-body systems dynamics with autoregressive models. *arXiv preprint arXiv:2009.05580*, 2020.
- [230] Kenny Choo, Giuseppe Carleo, Nicolas Regnault, and Titus Neupert. Symmetries and many-body excitations with neural-network quantum states. *Physical review letters*, 121(16):167204, 2018.
- [231] Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. *Nature communications*, 11(1):1–7, 2020.
- [232] Yusuke Nomura. Helping restricted boltzmann machine with quantum-state representation by restoring symmetry. *Journal of Physics: Condensed Matter*, 2021.
- [233] 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. doi: 10.1103/PhysRevLett.124.097201. URL <https://link.aps.org/doi/10.1103/PhysRevLett.124.097201>.
- [234] Tom Vieijra and Jannes Nys. Many-body quantum states with exact conservation of non-abelian and lattice symmetries through variational monte carlo. *arXiv e-prints*, pages arXiv–2104, 2021.
- [235] Juan Carrasquilla, Giacomo Torlai, Roger G Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 2019.
- [236] Michael J. Hartmann and Giuseppe Carleo. Neural-network approach to dissipative quantum many-body dynamics. *Phys. Rev. Lett.*, 122:250502, Jun 2019. doi: 10.1103/PhysRevLett.122.250502. URL <https://link.aps.org/doi/10.1103/PhysRevLett.122.250502>.
- [237] 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. doi: 10.1103/PhysRevLett.122.250503. URL <https://link.aps.org/doi/10.1103/PhysRevLett.122.250503>.
- [238] Moritz Reh, Markus Schmitt, and Martin Gärttner. Time-dependent variational principle for open quantum systems with artificial neural networks. *arXiv preprint arXiv:2104.00013*, 2021.
- [239] 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. doi: 10.1103/PhysRevLett.122.250501. URL <https://link.aps.org/doi/10.1103/PhysRevLett.122.250501>.

- [240] Nobuyuki Yoshioka and Ryusuke Hamazaki. Constructing neural stationary states for open quantum many-body systems. *Phys. Rev. B*, 99:214306, Jun 2019. doi: 10.1103/PhysRevB.99.214306. URL <https://link.aps.org/doi/10.1103/PhysRevB.99.214306>.
- [241] Irene López Gutiérrez and Christian B Mendl. Real time evolution with neural-network quantum states. *arXiv preprint arXiv:1912.08831*, 2019.
- [242] G. Fabiani and J. H. Mentink. Investigating ultrafast quantum magnetism with machine learning. *SciPost Phys.*, 7:4, 2019. doi: 10.21468/SciPostPhys.7.1.004. URL <https://scipost.org/10.21468/SciPostPhys.7.1.004>.
- [243] G Fabiani, MD Bouman, and JH Mentink. Supermagnonic propagation in two-dimensional antiferromagnets. *arXiv preprint arXiv:2101.10945*, 2021.
- [244] Stefanie Czischek, Martin Gärttner, and Thomas Gasenzer. Quenches near ising quantum criticality as a challenge for artificial neural networks. *Phys. Rev. B*, 98:024311, Jul 2018. doi: 10.1103/PhysRevB.98.024311. URL <https://link.aps.org/doi/10.1103/PhysRevB.98.024311>.
- [245] William Lauchlin McMillan. Ground state of liquid he 4. *Physical Review*, 138(2A):A442, 1965.
- [246] WMC Foulkes, Lubos Mitas, RJ Needs, and Guna Rajagopal. Quantum monte carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33, 2001.
- [247] Hisatoshi Yokoyama and Hiroyuki Shiba. Variational monte-carlo studies of hubbard model. i. *Journal of the Physical Society of Japan*, 56(4):1490–1506, 1987.
- [248] Sandro Sorella. Wave function optimization in the variational monte carlo method. *Physical Review B*, 71(24):241103, 2005.
- [249] Sandro Sorella. Generalized lanczos algorithm for variational quantum monte carlo. *Physical Review B*, 64(2):024512, 2001.
- [250] Eric Neuscamman, CJ Umrigar, and Garnet Kin-Lic Chan. Optimizing large parameter sets in variational quantum monte carlo. *Physical Review B*, 85(4):045103, 2012.
- [251] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.
- [252] Di Luo, Zhuo Chen, Kaiwen Hu, Zhizhen Zhao, Vera Mikyoung Hur, and Bryan K Clark. Gauge invariant autoregressive neural networks for quantum lattice models. *arXiv preprint arXiv:2101.07243*, 2021.
- [253] Dmitrii Kochkov and Bryan K Clark. Variational optimization in the ai era: Computational graph states and supervised wave-function optimization. *arXiv preprint arXiv:1811.12423*, 2018.
- [254] 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):1–8, 2020.
- [255] Bjarni Jónsson, Bela Bauer, and Giuseppe Carleo. Neural-network states for the classical simulation of quantum computing. *arXiv preprint arXiv:1808.05232*, 2018.

## Bibliography

- [256] Artem Borin and Dmitry A Abanin. Approximating power of machine-learning ansatz for quantum many-body states. *Physical Review B*, 101(19):195141, 2020.
- [257] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73:094423, Mar 2006. doi: 10.1103/PhysRevB.73.094423. URL <https://link.aps.org/doi/10.1103/PhysRevB.73.094423>.
- [258] Jens Eisert. Entanglement and tensor network states. *arXiv preprint arXiv:1308.3318*, 2013.
- [259] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [260] 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.
- [261] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- [262] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.
- [263] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [264] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.
- [265] Attila Szabó and Claudio Castelnovo. Neural network wave functions and the sign problem. *Physical Review Research*, 2(3):033075, 2020.
- [266] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [267] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
- [268] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016.
- [269] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.
- [270] Damian Hofmann, Giammarco Fabiani, Johan H Mentink, Giuseppe Carleo, and Michael A Sentef. Role of stochastic noise and generalization error in the time propagation of neural-network quantum states. *arXiv preprint arXiv:2105.01054*, 2021.
- [271] TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <http://tensorflow.org/>. Software available from tensorflow.org.



- [272] Sheng-Hsuan Lin. Neural-network-quantum-state. <https://github.com/ShHsLin/Neural-Network-Quantum-State>, 2021.
- [273] Yu I Manin. Vychislimoe i nevychislimoe (computable and noncomputable), moscow: Sov, 1980.
- [274] Paul Benioff. Quantum mechanical hamiltonian models of turing machines. *Journal of Statistical Physics*, 29:515–546, 1982.
- [275] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- [276] Dieter Jaksch, Christoph Bruder, Juan Ignacio Cirac, Crispin W Gardiner, and Peter Zoller. Cold bosonic atoms in optical lattices. *Physical Review Letters*, 81(15):3108, 1998.
- [277] Markus Greiner, Olaf Mandel, Tilman Esslinger, Theodor W Hänsch, and Immanuel Bloch. Quantum phase transition from a superfluid to a mott insulator in a gas of ultracold atoms. *nature*, 415(6867):39–44, 2002.
- [278] Jiehang Zhang, Guido Pagano, Paul W Hess, Antonis Kyprianidis, Patrick Becker, Harvey Kaplan, Alexey V Gorshkov, Z-X Gong, and Christopher Monroe. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature*, 551(7682):601–604, 2017.
- [279] Christie S Chiu, Geoffrey Ji, Annabelle Bohrdt, Muqing Xu, Michael Knap, Eugene Demler, Fabian Grusdt, Markus Greiner, and Daniel Greif. String patterns in the doped hubbard model. *Science*, 365(6450):251–256, 2019.
- [280] Biswaroop Mukherjee, Parth B Patel, Zhenjie Yan, Richard J Fletcher, Julian Struck, and Martin W Zwierlein. Spectral response and contact of the unitary fermi gas. *Physical review letters*, 122(20):203402, 2019.
- [281] Giulia Semeghini, Harry Levine, Alexander Keesling, Sepehr Ebadi, Tout T Wang, Dolev Bluvstein, Ruben Verresen, Hannes Pichler, Marcin Kalinowski, Rhine Samajdar, et al. Probing topological spin liquids on a programmable quantum simulator. *Science*, 374(6572):1242–1247, 2021.
- [282] Dolev Bluvstein, Ahmed Omran, Harry Levine, Alexander Keesling, Giulia Semeghini, Sepehr Ebadi, Tout T Wang, Alexios A Michailidis, Nishad Maskara, W Wei Ho, et al. Controlling quantum many-body dynamics in driven rydberg atom arrays. *Science*, 371(6536):1355–1359, 2021.
- [283] John Preskill. Quantum computing 40 years later. *arXiv preprint arXiv:2106.10522*, 2021.
- [284] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- [285] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. In *Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics*, volume 68, pages 13–58, 2010.
- [286] Earl T Campbell, Barbara M Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, 2017.

## Bibliography

- [287] A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, 1997.
- [288] C Schön, K Hammerer, Michael M Wolf, J Ignacio Cirac, and E Solano. Sequential generation of matrix-product states in cavity qed. *Physical Review A*, 75(3):032311, 2007.
- [289] Sarang Gopalakrishnan and Austen Lamacraft. Unitary circuits of finite depth and infinite width from quantum channels. *Physical Review B*, 100(6):064309, 2019.
- [290] Shi-Ju Ran. Efficient encoding of matrix product states into quantum circuits of one-and two-qubit gates. *arXiv preprint arXiv:1908.07958*, 2019.
- [291] Zhi-Yuan Wei, Daniel Malz, Alejandro González-Tudela, and J Ignacio Cirac. Generation of photonic matrix product states with a rydberg-blockaded atomic array. *arXiv preprint arXiv:2011.03919*, 2020.
- [292] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.
- [293] Jens H Bardarson, Frank Pollmann, and Joel E Moore. Unbounded growth of entanglement in models of many-body localization. *Physical review letters*, 109(1):017202, 2012.
- [294] Fernando GSL Brandão, Wissam Chemissany, Nicholas Hunter-Jones, Richard Kueng, and John Preskill. Models of quantum complexity growth. *arXiv preprint arXiv:1912.04297*, 2019.
- [295] Andrew M Childs and Yuan Su. Nearly optimal lattice simulation by product formulas. *Physical review letters*, 123(5):050503, 2019.
- [296] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum simulation of time-dependent hamiltonians and the convenient illusion of hilbert space. *Physical review letters*, 106(17):170501, 2011.
- [297] A. V. Uvarov, A. S. Kardashin, and J. D. Biamonte. Machine learning phase transitions with a quantum processor. *Phys. Rev. A*, 102:012415, Jul 2020. doi: 10.1103/PhysRevA.102.012415. URL <https://link.aps.org/doi/10.1103/PhysRevA.102.012415>.
- [298] J Karthik, Auditya Sharma, and Arul Lakshminarayan. Entanglement, avoided crossings, and quantum chaos in an ising model with a tilted magnetic field. *Physical Review A*, 75(2):022304, 2007.
- [299] Jeongwan Haah, Matthew Hastings, Robin Kothari, and Guang Hao Low. Quantum algorithm for simulating real time evolution of lattice hamiltonians. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 350–360. IEEE, 2018.
- [300] Markus Heyl, Philipp Hauke, and Peter Zoller. Quantum localization bounds trotter errors in digital quantum simulation. *Science advances*, 5(4):eaau8342, 2019.
- [301] Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. Quantum fingerprinting. *Phys. Rev. Lett.*, 87:167902, Sep 2001. doi: 10.1103/PhysRevLett.87.167902. URL <https://link.aps.org/doi/10.1103/PhysRevLett.87.167902>.



- [302] Daniel Gottesman and Isaac Chuang. Quantum digital signatures. *arXiv preprint quant-ph/0105032*, 2001.
- [303] Oscar Higgott, Daochen Wang, and Stephen Brierley. Variational quantum computation of excited states. *Quantum*, 3:156, 2019.
- [304] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C Benjamin. Variational quantum algorithms for discovering hamiltonian spectra. *Physical Review A*, 99(6):062304, 2019.
- [305] Lukasz Cincio, Yiğit Subaşı, Andrew T Sornborger, and Patrick J Coles. Learning the quantum algorithm for state overlap. *New Journal of Physics*, 20(11):113022, 2018.
- [306] Marco Cerezo, Alexander Poremba, Lukasz Cincio, and Patrick J Coles. Variational quantum fidelity estimation. *Quantum*, 4:248, 2020.
- [307] Marco Fanizza, Matteo Rosati, Michalis Skotiniotis, John Calsamiglia, and Vittorio Giovannetti. Beyond the swap test: optimal estimation of quantum state overlap. *Physical Review Letters*, 124(6):060503, 2020.
- [308] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- [309] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K Fährmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *arXiv preprint arXiv:1910.01155*, 2019.
- [310] Javier Gil Vidal and Dirk Oliver Theis. Calculus on parameterized quantum circuits. *arXiv preprint arXiv:1812.06323*, 2018.
- [311] Ken M Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2(4):043158, 2020.
- [312] Robert M Parrish, Joseph T Iosue, Asier Ozaeta, and Peter L McMahon. A jacobi diagonalization and anderson acceleration algorithm for variational quantum algorithm parameter optimization. *arXiv preprint arXiv:1904.03206*, 2019.
- [313] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Quantum circuit structure learning. *arXiv preprint arXiv:1905.09692*, 2019.
- [314] Matteo Rizzi, Simone Montangero, and Guifre Vidal. Simulation of time evolution with multiscale entanglement renormalization ansatz. *Phys. Rev. A*, 77:052328, May 2008. doi: 10.1103/PhysRevA.77.052328. URL <https://link.aps.org/doi/10.1103/PhysRevA.77.052328>.
- [315] Matthew Otten, Cristian L. Cortes, and Stephen K. Gray. Noise-resilient quantum dynamics using symmetry-preserving ansatzes, 2019.
- [316] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.
- [317] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.*, 9(1):1–6, 2018. ISSN 20411723. doi: 10.1038/s41467-018-07090-4. URL <http://dx.doi.org/10.1038/s41467-018-07090-4>.

## Bibliography

- [318] Mario Motta, Chong Sun, Adrian TK Tan, Matthew J O'Rourke, Erika Ye, Austin J Minnich, Fernando GSL Brandão, and Garnet Kin-Lic Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, 16(2):205–210, 2020.
- [319] Kübra Yeter-Aydeniz, Raphael C Pooser, and George Siopsis. Practical quantum computation of chemical and nuclear energy levels using quantum imaginary time evolution and lanczos algorithms. *npj Quantum Information*, 6(1):1–8, 2020.
- [320] Kok Chuan Tan. Fast quantum imaginary time evolution. *arXiv preprint arXiv:2009.12239*, 2020.
- [321] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.
- [322] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, F Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, et al. Qiskit: An open-source framework for quantum computing. *Accessed on: Mar, 16, 2019*.
- [323] Gregory H Wannier. Dynamics of band electrons in electric and magnetic fields. *Reviews of Modern Physics*, 34(4):645, 1962.
- [324] Katharine Hyatt, James R Garrison, and Bela Bauer. Extracting entanglement geometry from quantum states. *Physical review letters*, 119(14):140502, 2017.
- [325] Fangli Liu, Seth Whitsitt, Jonathan B Curtis, Rex Lundgren, Paraj Titum, Zhi-Cheng Yang, James R Garrison, and Alexey V Gorshkov. Circuit complexity across a topological phase transition. *Physical Review Research*, 2(1):013323, 2020.
- [326] Zijian Xiong, Dao-Xin Yao, and Zhongbo Yan. Nonanalyticity of circuit complexity across topological phase transitions. *Physical Review B*, 101(17):174305, 2020.
- [327] Adrien Bolens and Markus Heyl. Reinforcement learning for digital quantum simulation. *arXiv preprint arXiv:2006.16269*, 2020.
- [328] Michael Lubasch, Jaewoo Joo, Pierre Moinier, Martin Kiffner, and Dieter Jaksch. Variational quantum algorithms for nonlinear problems. *Phys. Rev. A*, 101:010301, Jan 2020. doi: 10.1103/PhysRevA.101.010301. URL <https://link.aps.org/doi/10.1103/PhysRevA.101.010301>.
- [329] Nathan Wiebe, Dominic W Berry, Peter Høyer, and Barry C Sanders. Simulating quantum dynamics on a quantum computer. *Journal of Physics A: Mathematical and Theoretical*, 44(44):445308, 2011.
- [330] Daniel M Greenberger, Michael A Horne, and Anton Zeilinger. Going beyond bell's theorem. *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72, 1989.
- [331] Bernhard Jobst, Adam Smith, and Frank Pollmann. Finite-depth scaling of infinite quantum circuits for quantum critical points. *arXiv preprint arXiv:2203.11975*, 2022.
- [332] Gavin E. Crooks. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition, 2019. URL <https://arxiv.org/abs/1905.13311>.
- [333] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Structure optimization for parameterized quantum circuits. *Quantum*, 5:391, 2021.

- [334] Yuan Liang Lim, Sean D Barrett, Almut Beige, Pieter Kok, and Leong Chuan Kwek. Repeat-until-success quantum computing using stationary and flying qubits. *Physical Review A*, 73(1):012304, 2006.
- [335] Kosuke Mitarai and Keisuke Fujii. Methodology for replacing indirect measurements with direct measurements. *Physical Review Research*, 1(1):013006, 2019.
- [336] Taichi Kosugi, Yusuke Nishiya, and Yu-ichiro Matsushita. Probabilistic imaginary-time evolution by using forward and backward real-time evolution with a single ancilla: first-quantized eigensolver of quantum chemistry for ground states, 2021. URL <https://arxiv.org/abs/2111.12471>.
- [337] Vedika Khemani, Frank Pollmann, and Shivaji Lal Sondhi. Obtaining highly excited eigenstates of many-body localized hamiltonians by the density matrix renormalization group approach. *Physical review letters*, 116(24):247204, 2016.
- [338] Natalia Chepiga and Frédéric Mila. Excitation spectrum and density matrix renormalization group iterations. *Physical Review B*, 96(5):054425, 2017.
- [339] Moritz August and Mari Carmen Bañuls. Efficient approximation for global functions of matrix product operators. *Physical Review B*, 98(7):075128, 2018.
- [340] Valentin Zauner-Stauber, Laurens Vanderstraeten, Matthew T Fishman, Frank Verstraete, and Jutho Haegeman. Variational optimization algorithms for uniform matrix product states. *Physical Review B*, 97(4):045145, 2018.
- [341] Daiki Adachi, Tsuyoshi Okubo, and Synge Todo. Anisotropic tensor renormalization group. *Physical Review B*, 102(5):054432, 2020.
- [342] Daisuke Kadoh, Hideaki Oba, and Shinji Takeda. Triad second renormalization group. *Journal of High Energy Physics*, 2022(4):1–26, 2022.
- [343] Shinichiro Akiyama, Daisuke Kadoh, Yoshinobu Kuramashi, Takumi Yamashita, and Yusuke Yoshimura. Tensor renormalization group approach to four-dimensional complex  $\phi^4$  theory at finite density. *Journal of High Energy Physics*, 2020(9):1–11, 2020.
- [344] Shinichiro Akiyama, Yoshinobu Kuramashi, and Yusuke Yoshimura. Phase transition of four-dimensional lattice  $\phi^4$  theory with tensor renormalization group. *Physical Review D*, 104(3):034507, 2021.
- [345] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [346] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [347] Or Sharir, Garnet Kin Chan, and Anima Anandkumar. Towards neural variational monte carlo that scales linearly with system size. *arXiv preprint arXiv:2212.11296*, 2022.
- [348] Ao Chen and Markus Heyl. Efficient optimization of deep neural quantum states toward machine precision. *arXiv preprint arXiv:2302.01941*, 2023.
- [349] Yuan-Hang Zhang and Massimiliano Di Ventra. Transformer quantum state: A multipurpose model for quantum many-body problems. *Physical Review B*, 107(7):075147, 2023.

## Bibliography

- [350] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.
- [351] Piotr Czarnik, Jacek Dziarmaga, and Philippe Corboz. Time evolution of an infinite projected entangled pair state: An efficient algorithm. *Physical Review B*, 99(3):035115, 2019.
- [352] Jacek Dziarmaga. Time evolution of an infinite projected entangled pair state: A gradient tensor update in the tangent space. *Physical Review B*, 106(1):014304, 2022.
- [353] Markus Schmitt, Marek M Rams, Jacek Dziarmaga, Markus Heyl, and Wojciech H Zurek. Quantum phase transition dynamics in the two-dimensional transverse-field ising model. *Science Advances*, 8(37):eabl6850, 2022.
- [354] Tiago Mendes-Santos, Markus Schmitt, and Markus Heyl. Highly resolved spectral functions of two-dimensional systems with neural quantum states. *arXiv preprint arXiv:2303.08184*, 2023.
- [355] Andrew J Ferris and Guifre Vidal. Perfect sampling with unitary tensor networks. *Physical Review B*, 85(16):165146, 2012.
- [356] Zhuo Chen, Laker Newhouse, Eddie Chen, Di Luo, and Marin Soljačić. Autoregressive neural tensor net: Bridging neural networks and tensor networks for quantum many-body simulation. *arXiv preprint arXiv:2304.01996*, 2023.
- [357] Christopher Roth. Iterative retraining of quantum spin models using recurrent neural networks. *arXiv preprint arXiv:2003.06228*, 2020.
- [358] Dian Wu, Riccardo Rossi, Filippo Vicentini, and Giuseppe Carleo. From tensor network quantum states to tensorial recurrent neural networks. *arXiv preprint arXiv:2206.12363*, 2022.
- [359] Zhengxiong Wang and Anton Ragni. Approximate fixed-points in recurrent neural networks. *arXiv preprint arXiv:2106.02417*, 2021.
- [360] 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, 2020.
- [361] Gregory Boyd and Bálint Koczor. Training variational quantum circuits with covar: covariance root finding with classical shadows. *Physical Review X*, 12(4):041022, 2022.
- [362] Anirban Narayan Chowdhury and Rolando D Somma. Quantum algorithms for gibbs sampling and hitting-time estimation. *arXiv preprint arXiv:1603.02940*, 2016.
- [363] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [364] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.
- [365] Markus Hauru, Clement Delcamp, and Sebastian Mizera. Renormalization of tensor networks using graph-independent local truncations. *Physical Review B*, 97(4):045111, 2018.